2. Algebraic review (focus on real, symmetric, positive)

* Transpose of a matrix
  $(A^T)^T = A$
  $(A+B)^T = A^T + B^T$
  $(AB)^T = B^T A^T$
* A matrix A is symmetric $\Leftrightarrow A = A^T$

* Inverse of A: $AA^{-1} = A^{-1}A = I$
  - A is invert $\Leftrightarrow$ { rank A = size A
    rows / columns of A are linearly indep
    the only sol of $A\underset{\sim}{x} = 0$ is $x = 0$

* A and B are invertible $\Leftrightarrow$ $(AB)^{-1} = B^{-1}A^{-1}$
  $(A^T)^{-1} = (A^{-1})^T$

* Def 2.5

Supp - real valued r.vs $X_1, \dots, X_p$ are mutually independent $F_{\underset{\sim}{X}}(\underset{\sim}{x}) = \prod_{i=1}^{p} F_{X_i}(x_i)$, $\forall \underset{\sim}{x} \in \mathbb{R}^p$

---

* Def

The two random vectors $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ are independent $\iff F_{\underset{\sim}{X},\underset{\sim}{Y}}(\underset{\sim}{x}, \underset{\sim}{y}) = F_{\underset{\sim}{X}}(\underset{\sim}{x})\, F_{\underset{\sim}{Y}}(\underset{\sim}{y})$

* $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ are independent $\iff \Sigma \begin{bmatrix} \underset{\sim}{X} \\ \underset{\sim}{Y} \end{bmatrix} = \begin{bmatrix} \Sigma_X & 0 \\ 0 & \Sigma_Y \end{bmatrix}$

$(X_1, \dots, X_p)$ don't need to be independent

---

* Def (Population correlation matrix)

Let covariance - covariance matrix $\Sigma_{\underset{\sim}{X}} = V(\underset{\sim}{X}) = C(\underset{\sim}{X}, \underset{\sim}{X}) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix}$ where $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_p \end{bmatrix}$

- Then consider (population) correlation matrix

$\rho_{\underset{\sim}{X}} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & 1 & \cdots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \cdots & 1 \end{bmatrix}_{p \times p}$ a (symmetric) matrix $\quad \rho_{ik} = \operatorname{cor}(X_i, X_k) = \dfrac{\sigma_{ik}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{kk}}}$

- Let $V^{1/2}$: standard deviation matrix

$V^{1/2} = \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_{pp}} \end{bmatrix}_{p \times p} = [\operatorname{diag}(\sqrt{\sigma_{ii}})]$

Then

$\Sigma_X = V^{1/2} \rho_X V^{1/2}$

$\rho_X = V^{-1/2} \Sigma_X V^{-1/2}$

---

* The mean vector + covariance matrix for linear combinations of random vectors

* Theorem 2.3

Let $\underset{\sim}{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix}$ (number vector) then $\underset{\sim}{c}^T \underset{\sim}{X} = [c_1 \; c_2 \; \cdots \; c_p] \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \sum_{i=1}^{n} c_i X_i$ is a linear combination of p random variables $X_1, \dots, X_p$.

$\underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$ (random vector)

Then we have a) $E(c^T X) = c^T E(X)$

b) $V(c^T X) = \underset{\sim}{c}^T V(X) \underset{\sim}{c}$

---

* Theorem 2.3

Let $\underset{\sim}{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ c_{p1} & c_{p2} & \cdots & c_{pp} \end{bmatrix}_{p \times p}$ $\underset{\sim}{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix}$ then $\underset{\sim}{Z} = \underset{\sim}{C}\underset{\sim}{X} + \underset{\sim}{d} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & \cdots & c_{pp} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix} = \begin{bmatrix} Z_{11} \\ Z_{21} \\ \vdots \\ Z_{p1} \end{bmatrix}$

$Z_i = (c_{i1} X_1 + c_{i2} X_2 + \cdots + c_{ip} X_p) + d_i = \sum_{j=1}^{p} c_{ij} X_j + d_i$

Then we have

$E(\underset{\sim}{C}\underset{\sim}{X} + \underset{\sim}{d}) = \underset{\sim}{C} E(\underset{\sim}{X}) + \underset{\sim}{d}$

$V(CX) = C\, V(X)\, C^T$

$V(CX + d) = C\, V(X)\, C^T$

* Theorem:

$A_{p \times p}$: symmetric matrix, $X_{p \times 1} \sim (\mu_{x1}, \Sigma_x)$ $\Rightarrow E(X^T A X) = \operatorname{tr}(A\Sigma_x) + \mu_x^T A \mu_x$

($X^T A X$: quadratic form of X)

* Chapter 3: Sample geometry and Random sampling.

* 3.1 Introduction

* 3.2 The geometry of the sample

- A single multivariate observation is a collection of measurements on p different variables taken on a trial.
- If we have n observations for those p variables, the entire data set is

$$\underset{n\times p}{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}_{n\times p} = \begin{bmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_n^T \end{bmatrix} = [Y_1 \,|\, Y_2 \,|\, \cdots \,|\, Y_p] \qquad \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix}$$

(person ↑, variable →)

where $X_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{bmatrix}_{p\times 1}$, $i = \overline{1,n}$ (↑ people)

observe/need from the person i

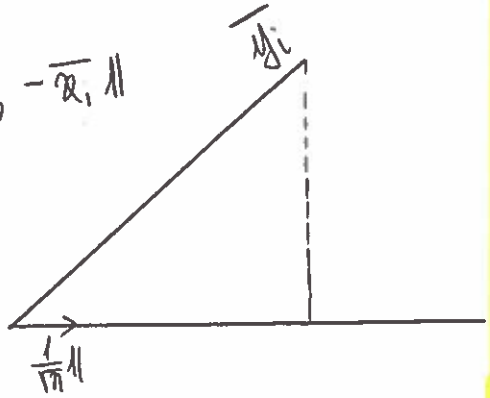
where $Y_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix}_{n\times 1}$, $i = \overline{1,p}$

observe for variable i by n people

* Def 3.1:

For each $Y_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix}$, we define deviation vector $d_i = y_i - \bar{x}_i \mathbb{1}$

- Projection of y_i to $\frac{1}{\sqrt{n}}\mathbb{1}$ is $\bar{x}_i \mathbb{1}$, where $\mathbb{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$.



4. Multivariate normal distribution.

* Recall univariate normal dist $X \sim N(\mu, \sigma^2)$ $f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$

$\frac{(x-\mu)^2}{\sigma^2} = (x-\mu)(\sigma^2)^{-1}(x-\mu)^T$

* Multivariate normal distribution

$\underset{\sim}{X}_{p\times 1} \sim N_p(\mu_{p\times 1}, \Sigma_{p\times p})$ where Σ p.d (Σ p.s.d (in general case))

Then pdf

$f_{\underset{\sim}{X}}(\underset{\sim}{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\underset{\sim}{x}-\mu)^T\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu})\right\}$, where $\underset{\sim}{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}$

is a function in $\mathbb{R}$, depends on $(x_1, \dots, x_p)$

* Example 4.1 The bivariate normal ($p=2$)

$\underset{\sim}{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}\right)$ X_1 and X_2 are independent $\Leftrightarrow \rho = 0$

$f_{\underset{\sim}{X}}(x_1, x_2) = \frac{1}{2\pi\sigma_1^2\sigma_2^2(1-\rho^2)} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right)\right]\right.$

* Theorem 4.1.

Let $\underset{\sim}{X} = \begin{bmatrix} \underset{\sim}{X_1} \\ \underset{\sim}{X_2} \end{bmatrix} \sim N_p(\mu, \Sigma)$

then X_1 and X_2 are independent $\Leftrightarrow \Sigma = \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right]$

* Theorem 4.2

Constant probability density contour = {all $\underset{\sim}{x}$ such that $(\underset{\sim}{x}-\underset{\sim}{\mu})^T\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}) = c^2$}

= surface of an ellipsoid {centered at $\underset{\sim}{\mu}$ / having axes $\pm c\sqrt{\lambda_i}\, e_i$

(where (λ_i, e_i) are (eigenvalues, eigenvectors) of Σ

* Result 4.1/53.

Σ positive definite $\Leftrightarrow$ Σ^{-1} positive definite

(λ_i, e_i) are (eigenvalue, eigenvector of Σ) $\Rightarrow$ $(\frac{1}{\lambda_i}, e_i)$ are (eigenvalue, eigenvector) of Σ^{-1}

* Theorem 4.3?

$\underset{\sim}{X} = \begin{bmatrix} \underset{\sim}{X_1} \\ \underset{\sim}{X_2} \end{bmatrix} \begin{matrix} p_1 \\ p_2 \end{matrix} \sim N_{p_1+p_2}\left[\mu, \Sigma = \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right]\right]$

then $X_1 \sim N_{p_1}(\mu_1, \Sigma_{11})$

* Additional properties of multivariate normal distribution

Let $\underset{p\times 1}{X} \sim N_p(\mu, \Sigma)$, $\Sigma > 0$, then

- $X + c \sim N_p(\mu + c, \Sigma)$
- $\underset{q\times 1}{AX} \sim N_q(\underset{q\times 1}{A\mu}, A\Sigma A^T)$ $\qquad a^T X \sim N(a^T\mu, a^T\Sigma a)$ ($1\times p$, $p\times 1$)
- If $\Sigma > 0$ then $Y = \Sigma^{-1/2}(X - \mu) \sim N_p(\underset{p\times 1}{0}, \underset{p\times p}{I})$
- $\Sigma = \sigma^2 I$ (diagonal matrix with elements σ^2) and G is orthogonal ($G^T G = I$), G is $q\times p$ ($q < p$) $\Rightarrow \underset{q\times 1}{GX} \sim N(\underset{q\times 1}{G\mu}, \underset{q\times q}{\sigma^2 I})$

* Alternative multivariate normal distribution (MVN)

Let $\underset{\sim}{Z} \sim N(0, I)$ $\qquad Z = \begin{pmatrix} Z_1 \\ \vdots \\ Z_p \end{pmatrix}$ where $Z_i \sim N(0, 1)$.

then $X = \mu + LZ \sim N(\mu, LL^T)$

* Def (for Chi Square):

Let $\underset{p\times 1}{Z} \sim N(0, I)$ $\qquad Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} \sim N(0, I)$ where $Z_i \sim N(0, 1)$.

then $= Z^T Z = \sum_{i=1}^{p} Z_i^2 \sim \chi^2_p$.

* Theorem

$X \sim N_p(\mu, \underset{p\times p}{\Sigma})$ Σ p.d.

then $(X-\mu)^T \Sigma^{-1} (X-\mu) \sim \chi^2_p$

When $X_1, \dots, X_n \sim N_p(\mu, \Sigma)$

$\Rightarrow n(\bar{X}-\mu)\Sigma^{-1}(\bar{X}-\mu) \sim \chi^2_p$

* Theorem

$X_p \sim N_p(\mu, \underset{p\times p}{\Sigma})$ $\Sigma > 0$ $\Rightarrow P\left((X-\mu)^T \Sigma^{-1}(X-\mu) \le \chi^2_{p,1-\alpha}\right) = 1-\alpha$.

* Theorem 4.6.7 (Conditional distribution)

$X \sim N_p(\mu, \Sigma)$ $\Sigma > 0$

$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \begin{matrix} p_1 \\ p_2 \end{matrix}$ $\qquad p_1 + p_2 = p$

$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ $\qquad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$ $\qquad \Sigma_{21} = \Sigma_{12}^T$

Then:

$$X_1 \big|_{X_2 = x_2} \sim N_{p_1}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2),\ \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

* Maximum likelihood estimation of Σ, μ

* Def

Let $X_1, \dots, X_n \overset{iid}{\sim} N_p(\mu, \Sigma)$, $\Sigma > 0$

The likelihood is

$$L(\mu, \Sigma \mid X_1, \dots, X_n) = \prod_{j=1}^{n} \left(\frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_j - \mu)^T \Sigma^{-1} (X_j - \mu)} \right)$$

$$= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\sum_{j=1}^{n} \frac{1}{2}(X_j - \mu)^T \Sigma^{-1} (X_j - \mu)}$$

* Remark 4.9/p168

$A_{p \times p}$ symmetric

$x_{p \times 1}$ vector

then $x^T A x = tr(x^T A x) = tr(A x x^T)$

* Theorem

$$\sum_{j=1}^{n} \left((X_j - \mu)^T \Sigma^{-1} (X_j - \mu) \right) = tr\left[\Sigma^{-1} \left(\underbrace{\sum_{j=1}^{n} (X_j - \bar{X})(X_j - \bar{X})^T}_{(n-1)S} \right) \right] + n \left[(\bar{X} - \mu)^T \Sigma^{-1} (\bar{X} - \mu) \right]$$

* Def 4.2 (Sufficiency)

$T(X)$ is sufficient for θ if $P(X = x \mid T(X) = t, \theta) = P(X = x \mid T(x) = t)$, free of θ

* Theorem 4.8 (Factorization theorem for sufficiency)

T is sufficient for $\theta \iff f_\theta(x) = h(x)\, g_\theta(T(x))$

* Theorem 4.9

$(S, \bar{X})$ are jointly sufficient for (μ, Σ) $\quad S = \frac{1}{(n-1)} \sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})^T$

* Theorem 4.10

Let $B > 0$ $(p \times p)$

$b > 0$ be a scalar

$$\frac{1}{|\Sigma|^b} e^{-\frac{1}{2} tr(\Sigma^{-1} B)} \leq \frac{1}{|B|^b} (2b)^{bp} e^{-bp} \quad \text{for all } \Sigma > 0$$

equality $\iff \Sigma = \frac{1}{2b} B$

* Theorem 4.11

Let $X_1, \dots, X_n \overset{iid}{\sim} N_p(\mu, \Sigma)$

The MLE of (μ, Σ) is

$\hat{\mu}_{MLE} = \bar{X}$

$\hat{\Sigma}_{MLE} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})^T$

$\bar{X}$ and $\hat{\Sigma}_{MLE}$ independent

* Theorem 4.12

Let $A_{p \times p}$ symmetric

$B_{q \times p}$

$X \sim N_p(\mu, \Sigma)$

Then BX and $X^T A X$ are independent $\iff B \Sigma A = 0_{q \times p}$

(BX: $X \to x$; $X^T A X$: x^2)

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 2 & 4 & 2 & 6 \\ 1 & 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & C & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 4 & 4 & 2 & 6 \\ 2 & 2 & 3 & 1 \end{bmatrix} \quad B_1 = B_0 C$$

B_1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 3 & 4 \\ 3 & 6 & 7 & 8 \\ 4 & 4 & 2 & 6 \\ 2 & 2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 3 & 4 \\ 3 & 6 & 7 & 8 \\ 2 & 2 & 1 & 3 \\ 2 & 2 & 1 & 3 \end{bmatrix} \quad B_2 = D B_1 = D B_0 C$$

D. add B_1 add B_2

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 3 & 4 \\ 3 & 6 & 7 & 8 \\ 2 & 2 & 1 & 3 \\ 2 & 2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 & 7 \\ 6 & 7 & 8 \\ 2 & 2 & 1 & 3 \\ 2 & 2 & 1 & 3 \end{bmatrix}$$

E add B_2 B_3

$$\begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 \\ 8 & 6 & 7 & 3 \\ 3 & 2 & 1 & 2 \\ 3 & 2 & 1 & 2 \end{bmatrix} \quad B_4 = B_3 D$$

B_3 D B_4

subtract row 2 from each of the row.

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 & 6 & 7 & 3 \end{bmatrix} = \begin{bmatrix} 8 & 6 & 7 & 3 \end{bmatrix}$$

B_4 B_5

row 1 (B_5) = 1 row 1 (B_4) + (-1) row 2 (B_4)
row 2 (B_5) = row 2 (B_4)
row 3 (B_5) = row 3 (B_4) - (1) row 2 (B_4)

AP
work with column : after A
works with rows : P A
then Pa row / col Pormain diag

Time	Monday Jan 15	Tuesday Jan 16	Wednesday Jan 17	Thursday Jan 18	Friday Jan 19	Saturday Jan 20	Sunday Jan 21
8:00AM		MAT 682 - M001 Section 8:00AM - 9:20AM Carnegie 120		MAT 682 - M001 Section 8:00AM - 9:20AM Carnegie 120			
9:00AM		MAT 682 - M001 Section 8:00AM - 9:20AM Carnegie 120 MAT 652 - M001 Section 9:30AM - 10:50AM Carnegie 122		MAT 682 - M001 Section 8:00AM - 9:20AM Carnegie 120 MAT 652 - M001 Section 9:30AM - 10:50AM Carnegie 122	8:25 - 9:20 Carn 124 9:30 - 10:25 Carn 124		
10:00AM		MAT 652 - M001 Section 9:30AM - 10:50AM Carnegie 122		MAT 652 - M001 Section 9:30AM - 10:50AM Carnegie 122	ENL 640 - M002 Section 10:35AM - 11:30AM MARSHALL SQUARE MALL 208B		
11:00AM			11am - 12pm Math clinic				
12:00PM		MAT 695 - M001 Section 12:30PM - 1:50PM Carnegie 100	Office hr	MAT 695 - M001 Section 12:30PM - 1:50PM Carnegie 100			
1:00PM					12:45 - 1:40 Carn 124		
2:00PM					2:15 - 3:10 Carn 124		
3:00PM		MAT 526 - M001 Section 3:30PM - 4:50PM Carnegie 115		MAT 526 - M001 Section 3:30PM - 4:50PM Carnegie 115	Office hr		
4:00PM							
5:00PM							
6:00PM							
7:00PM							
8:00PM							
9:00PM							

Eigenvalues, eigenvectors and eigendecomposition

Eigendecomposition
Why's useful ?
M inverse
power
Properties
How to compute ?
Power iteration
QR algorithm

* Eigenvalue, eigenvector

$v \neq 0$, $Av = \lambda v \Rightarrow$ λ eigenvalue, v eigenvector

- $Av = \lambda v \Leftrightarrow (A - \lambda I)v = 0 \Rightarrow v \neq 0$ is a solution of $[A - \lambda I]v = 0$

** Why is eigendecomposition is useful ?

1. Assume A has k eigenvalues $\lambda_1, \dots, \lambda_k$ with k associated eigenvector $v_1, v_2, \dots, v_k$

Then if $V = [v_1\ v_2 \dots v_k]$ then $AV = V\Sigma$ — Special decomposition theorem

$$\Sigma = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \\ & & & \lambda_k \end{bmatrix}$$

$A = V\Sigma V^{-1}$.

$AV = V\Sigma$

- Σ: diagonal matrix contains all eigenvalue of A.
- V contains all eigen vectors of A
- We say V and A are similar

$\Rightarrow$ This can only be done if a matrix is diagonalizable.

$\Rightarrow$ Def of a diagonalizable matrix

A is a diagonalizable matrix if we can eigendecompose it into n eigenvectors.

$\in \mathbb{C}^{n \times n}$

+ Remark: If a matrix A is a ~~diagonal~~ triangular matrix
→ all eigenvalue are on the diagonal.

- Matrix inverse with eigendecomposition

$A = V\Sigma V^{-1} \Rightarrow A^{-1} = V\Sigma^{-1}V^{-1}$

- Power of a matrix with eigendecomposition

$A^n = V\Sigma^n V^{-1}$ for all integers n

* Properties of eigendecomposition

- $\det A = \prod_{i=1}^{n} \lambda_i$
- $\text{trace } A = \sum_{i=1}^{n} \lambda_i$
- eigenvalues of A^{-1} are λ_i^{-1}

  eigenvalues of A^n are λ_i^n

Theo 2.3
$A \in \mathbb{C}^{m \times m}$ Hermitian (symmetric if $A \in \mathbb{R}$)
→ a) all eigenvalues are real
b) all eigenvectors corresponding to "distinct" are orthogonal.

- The eigenvectors of A^{-1} = The eigenvectors of A.
- A is Hermitian, A is full rank (all row/columns are linearly independent) $\Rightarrow$ { eigenvectors are mutually orthogonal; eigenvalues are real }
- A is invertible $\Leftrightarrow \lambda_i \neq 0, \forall i$
- If the eigenvalues of A are distinct $\Rightarrow$ A can be eigendecomposed

§1.7 Aspects of multivariate analysis. 1.3 The organization of Data.

$p \geq 1$: # of variables/characters to record. (The values of those variables are all recorded for each distinct item/individual/unit.

x_{jk} ← k^{th} variable; j^{th} item/trial

Item \ variable	Variable 1	Variable 2	...	Variable (p)
Item 1				
⋮				
Item (n)				

{ n items (people); p variables.

⇒ $[X]_{n \times p}$ n: # observations (people/items); p: # of variables

* Descriptive Statistics

+ The arithmetic average (sample mean) $\bar{x}_1 = \frac{1}{n}\sum_{j=1}^{n} x_{j1}$ ← sample mean for the first variable.

$\bar{x}_k = \frac{1}{n}\sum_{j=1}^{n} x_{jk}$ (works with k^{th} column ↔ k^{th} variable).

+ sample variance $s_{kk} = s_k^2 = \frac{1}{n}\sum_{j=1}^{n}(x_{jk} - \bar{x}_k)^2$

each variable may have different units.

+ sample standard deviation $\sqrt{s_{kk}}$

+ sample covariance $s_{ik} = \frac{1}{n}\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)$

$s_{ik} > 0$, large ↔ large

$s_{ik} > 0$ large ↔ small

$s_{ik} \approx 0$, there is no particular association.

$s_{ik} = s_{ki}$, i^{th} variable, k^{th} variable

s_{ik}: A measure of linear association between the measurement of variable i^{th} and variable k.

+ Sample correlation coefficient $r_{ik} = \frac{s_{ik}}{\sqrt{s_{ii}}\sqrt{s_{kk}}} = \frac{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)}{\sqrt{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)^2}\sqrt{\sum_{j=1}^{n}(x_{jk} - \bar{x}_k)^2}}$

- is a standardized version of sample covariance
- is the same whether n or (n−1) is chosen as the common divisor for s_{ii}, s_{kk}, s_{ik}.
- can be considered sample covariance, when we replace x_{jk} by $\frac{x_{ji} - \bar{x}_i}{\sqrt{s_{ii}}}$ of standardized observations x_{jk} by $\frac{x_{jk} - \bar{x}_k}{\sqrt{s_{kk}}}$

① $-1 \leq r \leq 1$.

② r measures the strength of the linear association.

$r < 0$ ⇒ tendency for { one value in the pair < its average; other value is smaller than its average.

$r > 0$ ⇒ both value in one pair bigger or smaller than its average

③ r_{ik} is unchanged when x_{ji} (i^{th} variables) changed to $y_{ji} = a x_{ji} + b$, $j = 1, n$

x_{jk} (k^{th} variables) changed to $y_{jk} = c x_{jk} + d$, $j = 1, n$

a, c have the same sign

* The sum of squares of the deviations from the mean

$$w_{kk} = \sum_{j=1}^{n} (x_{jk} - \bar{x}_k)^2$$

* The sum of cross product deviation

$$w_{ik} = \sum_{j=1}^{n} (x_{ji} - \bar{x}_i)^{\#} (x_{jk} - \bar{x}_k)^{\#}$$

* Arrays of Basic Descriptive Statistics.

- Sample means $\underset{\sim}{\bar{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix}_{p \times 1}$

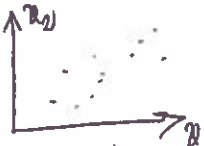
- Sample variances and covariances $\underset{\sim}{S_n} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & \cdots & s_{pp} \end{bmatrix}_{p \times p}$ ← symmetric matrix

use n to remind that n items are samples for elements.

- Sample correlation $\underset{\sim}{R} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & & \\ \vdots & & \ddots & \\ r_{p1} & r_{p2} & & 1 \end{bmatrix}_{p \times p}$ ← symmetric matrix

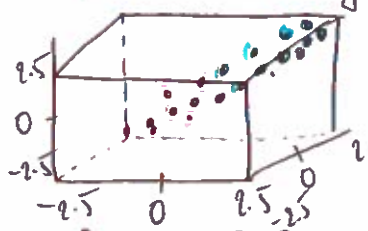
* Graphical Techniques.

- Plot pairs of variables is quite informative.



can help calculate $\bar{x}_1$, $\bar{x}_2$, s_{11}, s_{12}, s_{ik}

- The effect of unusual observation on sample correlations
- Graphic of (n points) in (p dimensions). $(x_{j1}, x_{j2}, x_{j3}, \cdots, x_{jp})$ for $j = \overline{1, n}$ ← or p points in n-dimensional space.
- Example 1.6: (looking for lower dimensional structure)



- some variables may have a much larger variance than the others.
- we want to use standardized variables $z_{jk} = \frac{x_{jk} - \bar{x}_k}{\sqrt{s_{kk}}}$

- Example 1.7 (looking for group structure in three dimensions) (p18)
  - female
  - male
  - groups of female and male.

1.4 Data displays and pictorial representations

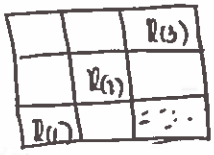
- 2-dimensional scatter plot
- graphs of growth curve
- stars
- chernoff face.

Powerful computers → help have good { data analysis / graphics.

C8 (principal component analysis) ⟩ represent p-dimensional observations in few (2) dimensions s.t. the original distances (or similarities) between pair of observations are (nearly) preserved.
C8' (canonical correlation analysis) ⟩

* Linking multiple 2-dimensional scatter plots.

* Example 1.8 (Linked scatter plots and brushing)



1, selecting
2, deleting outliers
3,

→ brushing: operation of highlighting points corresponding to a selected range of one of the variables

* Example 1.9 (Rotate plots in three dimensions).

Rotating and turning the 3-dimensional coordinate axes can help getting a better ~~idea~~ understanding of the three-dimensional aspects of the data. → to confirm what specimens are outliers

* Graphs of growth curves

Growth curve: the points are plotted and then connected by lines to produce a graph.
(height of a young child, for ex) is measure at each birth day)

* Example 1.10 (Arrays of growth curve)

for example a growth curve for weight of some female bears by year
(each ≠ line for each ≠ bear)

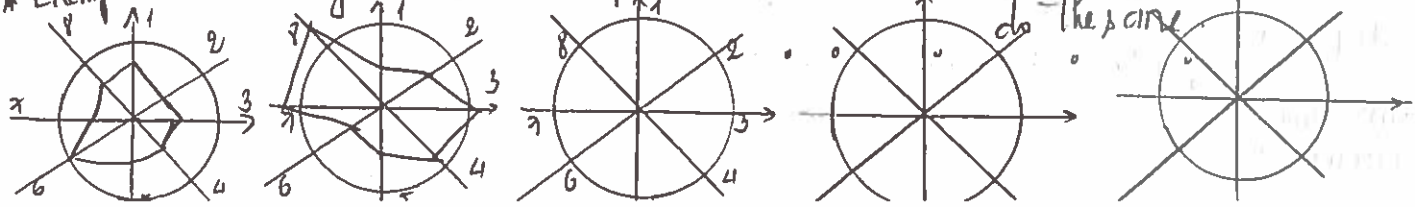
we can also have 7 individual growth curves for weight for 7 ≠ bears 

* Stars: each data unit consists of nonnegative observation on p ≥ 2 variables

→ In two dimensions, → construct circles with { a fixed radius.
{ has p equally spaced rays
{ the length of the rays represent the values of the variable

• The observations on all variables were standardized.

* Example 1.11 (utility data as stars) p27.



* Chernoff Faces.

* 1.5 Distance.

* Euclidean distance.

$P(x_1, \dots, x_p)$
$Q(y_1, \dots, y_p)$
$\rightarrow d(P,Q) := \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$

* Statistical distance

* $P(x_1, x_2)$
$Q(y_1, y_2)$

$$d(P,Q) = \sqrt{\frac{(x_1 - y_1)^2}{s_1^2} + \frac{(x_2 - y_2)^2}{s_2^2}} \qquad (1.13)$$

* $P(x_1, \dots, x_p)$
$Q(y_1, \dots, y_p)$

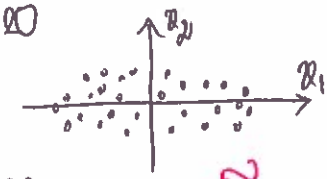
$$d(P,Q) = \sqrt{\sum_{i=1}^{p} \frac{(x_i - y_i)}{s_i^2}} = \sqrt{\frac{(x_1 - y_1)^2}{s_1^2} + \frac{(x_2 - y_2)^2}{s_2^2} + \dots + \frac{(x_p - y_p)^2}{s_p^2}}$$

(statistical distance)

If $s_1^2 = \dots = s_p^2$ ⇒ the Euclidean distance formula is appropriate

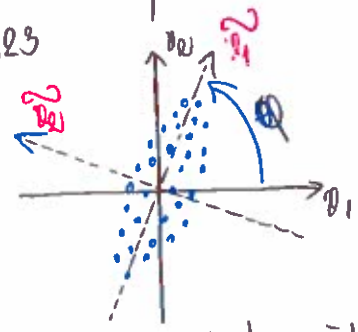
The above distance does not include the most important cases we shall encounter,
⟶ (because: assumption of independent coordinates.

Fig 2.0



when x_1 measurements do not vary independently of the x_2 measurements.
(x_1, x_2) ← has tendency to be large or small together.

Fig 2.3



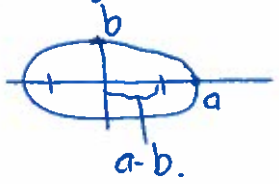
* So now we want to find a distance (statistical distance that is meaningful) when
{ variability in the x_1 direction is different from the variability in the x_2 direction.
the variables x_1 and x_2 are correlated
(If we look at things in the right way)

If we axes $(\tilde{x}_1, \tilde{x}_2)$ then with $P(\tilde{x}_1, \tilde{x}_2)$ $O(0,0)$ $d(P, O) = \sqrt{\frac{\tilde{x}_1^2}{s_1^2} + \frac{\tilde{x}_2^2}{s_2^2}}$ (1.17)
coordinate system in $(\tilde{x}_1, \tilde{x}_2)$ coordinate

where $\begin{cases} \tilde{x}_1 = x_1 \cos\varphi + x_2 \sin\varphi \\ \tilde{x}_2 = -x_1 \sin\varphi + x_2 \cos\varphi \end{cases}$ (1.18)

Then $d(P, O) = \sqrt{a_{11} x_1^2 + 2 a_{12} x_1 x_2 + a_{22} x_2^2}$
in original coordinate

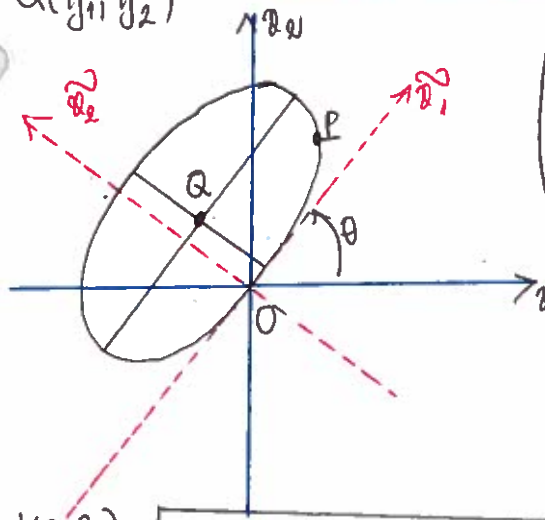
* Remind:
ellipse equation:
$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$



* In general

$P(x_1, x_2)$
$Q(y_1, y_2)$

$$d(P,Q) = \sqrt{a_{11}(x_1-y_1)^2 + 2a_{12}(x_1-y_1)(x_2-y_2) + a_{22}(x_2-y_2)^2} \quad (1.20)$$



- The coordinates of all points $P(x_1, x_2)$, that are a constant square c^2 from Q satisfy

$$a_{11}(x_1-y_1)^2 + 2a_{12}(x_1-y_1)(x_2-y_2) + a_2(x_2-y_2)^2 = c^2 \quad (1.21$$

(ellipse center Q)

- Let $(Q) = (0, 0, \dots, 0)$

$Q = (y_1, \dots, y_p)$ ← fixed point.

$$d(O,P) = \sqrt{a_{11}x_1^2 + a_{22}x_2^2 + \dots + a_{pp}x_p^2 + 2a_{12}x_1x_2 + 2a_{13}x_1x_3 + \dots + 2a_{p-1,p}x_{p-1}x_p}$$

$$d(P,Q) = \sqrt{a_{11}(x_1-y_1)^2 + a_{22}(x_2-y_2)^2 + \dots + a_{pp}(x_p-y_p)^2 + 2a_{12}(x_1-y_1)(x_2-y_2) + 2a_{13}(x_1-y_1)(x_3-y_2) + \dots + 2a_{p-1,p}(x_{p-1}-y_{p-1})(x_p-y_p)}$$

where a's are the numbers such that the distance are always nonnegative.

⇒ The distances are completely determined by coefficients (weights) a_{ik} $i = 1, \dots, p$, $k = 1, \dots, p$.

2.1 * Structure and organization

n: # observations (here we assume $n \gg p$)

p: # variables

Data is represented in the form

obs \ var	variable 1	variable 2	...	variable p
obs 1	x_{11}	x_{12}	...	x_{1p}
obs 2	x_{21}	x_{22}	...	x_{2p}
⋮	⋮	⋮	⋱	⋮
observation n	x_{n1}	x_{n2}	...	x_{np}

x_{jk}: observation j, variable k

$j = \overline{1,n}$, $p = \overline{1,p}$

* Wide form (repeated observations) (mostly data is presented in this way)

	Time 1			Time T		
	var 1	...	var p	var 1	...	var p
obs 1	x_{111}	...	x_{1p1}	x_{11T}		x_{1pT}
obs n	x_{n11}		x_{np1}	x_{n1T}		x_{npT}

$n, p \times T$

* Long form

		var 1	var 2	...	var p	ID
person 1	obs 1-1	x_{111}			x_{1p1}	1
	obs 1-2	x_{112}			x_{1p2}	1
	obs 1-T	x_{11T}			x_{1pT}	1
person 2	obs 2-1	x_{211}			x_{2p1}	2
	obs 2-2	x_{212}			x_{2p2}	2
	obs 2-T	x_{21T}			x_{2pT}	2
⋮						
person n						

nxT, p

↑ observation + time.

x_{ijk}: person, variable, time

* Review descriptive statistics.

← before analysing data:
- 1st step: check if data is correct ex 0 1 2 gender is not correct
- 2nd step: makes sense? → find out when do we [illegible]

- The sample mean of the k^{th} variable

$$\bar{x}_k = \frac{1}{n}\sum_{j=1}^{n} x_{jk} \qquad \underset{p\times 1}{\bar{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix}$$

- Sample variance. covariance.

$$s_k^2 = s_{kk} = \frac{1}{(n-1)}\sum_{j=1}^{n}(x_{jk}-\bar{x}_k)^2$$

$$s_{ik} = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ji}-\bar{x}_i)(x_{jk}-\bar{x}_k)$$

$$S_{p\times p} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & \cdots & s_{1p} \\ s_{21} & s_{22} & s_{23} & \cdots & s_{2p} \\ & & & & \\ s_{p1} & s_{p2} & & & s_{pp} \end{bmatrix}_{p\times p}$$

is a $p \times p$ symmetric matrix

correlation between variables.

- Sample correlation coefficient.

$$r_{ik} = \frac{s_{ik}}{\sqrt{s_{ii}\, s_{kk}}} \qquad -1 \le r_{ik} \le 1$$

$$R_{p\times p} = \begin{bmatrix} 1 & r_{12} & & r_{1p} \\ r_{21} & 1 & & r_{2p} \\ & & \ddots & \\ r_{p1} & r_{p2} & & 1 \end{bmatrix}_{p\times p}$$

* Distance and metric

Let $\mathcal{S}$ be a sample space.

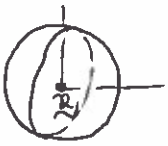
Distance function $d: \mathcal{S} \times \mathcal{S} \longrightarrow \mathbb{R}$ satisfies
$$\begin{cases} \forall x, y, z \in \mathcal{S} \\ d(x,y) \geq 0, \forall x,y \quad d(x,y)=0 \Leftrightarrow x=y \\ d(x,y) = d(y,x) \\ d(x,y) \leq d(x,z) + d(z,y) \end{cases}$$
non-negative, coincide, symmetry, Δ inequality

* Common distance in $\mathbb{R}$

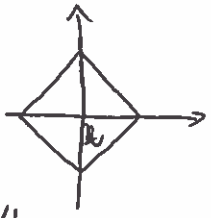
$\underset{\sim}{x}(x_1, \dots, x_p)$
$\underset{\sim}{y}(y_1, \dots, y_p)$

Euclidean
$$d(\underset{\sim}{x}, \underset{\sim}{y}) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

all points on this curve equidistant from $\underset{\sim}{x}$

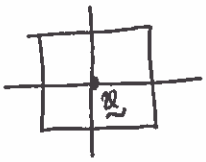


- City block
$$d(\underset{\sim}{x}, \underset{\sim}{y}) = \sum_{i=1}^{p} |x_i - y_i|$$



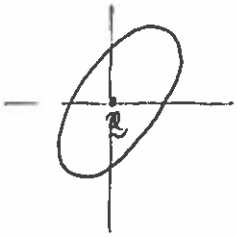
- Minkovski
$$d_\lambda(\underset{\sim}{x}, \underset{\sim}{y}) = \left[\sum_{i=1}^{p} |x_i - y_i|^\lambda \right]^{1/\lambda}, \quad \lambda \geq 1$$

- Manhatten:
$$d(\underset{\sim}{x}, \underset{\sim}{y}) = \lim_{\lambda \to \infty} d_\lambda(x,y) = \max_i |x_i - y_i|$$



- Mahalanobis distance
$$d(\underset{\sim}{x}, \underset{\sim}{y}) = \sqrt{(\underset{\sim}{x} - \underset{\sim}{y})^T_{1\times p} \; S^{-1}_{p\times p} \; (\underset{\sim}{x} - \underset{\sim}{y})_{p\times 1}}$$

S: variance - covariance metric



← includes the structure of data

2.7 Algebraic review (focus on Real, Symmetric, positive)

* Definition: Let $A = (A_{ij})_p$

Let $(A)_{ij} = (a_{ij})$ then transpose matrix of A: $A^T = (a_{ji})$

$(A^T)^T = A$

$(A+B)^T = A^T + B^T$

$(AB)^T = B^T A^T$

* Def 2.2

A matrix A is symmetric if $A^T = A$ all var, cov, cor matrices are symmetric.

* Def 2.3

- Let A be a matrix and suppose $\exists B$ s.t. $AB = BA = I$ then $A^{-1} = B$ is the inverse of A
- A matrix is invertible (non singular) $\Leftrightarrow$ When $A_{p\times p}$
  - (1) rank $A = p$ = the size of matrix.
  - (2) rows / columns of A linearly independent
  - (3) the only sol for $A\underset{\sim}{x} = 0$ is $\underset{\sim}{x} = \underset{\sim}{0}$

Note: $\underset{\sim}{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}$ $\underset{\sim}{x}^T = (x_1 \dots x_p)$

- A and B are invertible $(AB)^{-1} = B^{-1}A^{-1}$

$(A^T)^{-1} = (A^{-1})^T$

* Def 2.4

The determinant of a matrix $A_{p\times p}$ satisfies

$|AB| = |A||B|$

$|A^T| = |A|$

$|cA| = c^p|A|$

$|I| = 1$

$|A^{-1}| = |A|^{-1}$

$\det(\underset{\sim}{a}_1 \; \underset{\sim}{a}_{j+1} \; \underset{\sim}{a}_j \dots \underset{\sim}{a}_p) = -\det A$

$\det(\underset{\sim}{a}_j \; k\underset{\sim}{a}_j + \underset{\sim}{a}_i \; \dots) = k \det A$

($\underset{\sim}{a}_j$: column vector; $k\underset{\sim}{a}_j + \underset{\sim}{a}_i$: j column)

* Def 2.4

$A_{p\times p}$ is nonsingular iff $|A| \neq 0$

singular iff $|A| = 0$

* Def 2.5

Let A be a nonsingular matrix satisfying $A^{-1} = A^T$ def $\Leftrightarrow$ A is orthogonal

$AA^T = I$

$\Leftrightarrow \underset{\sim}{x}^T\underset{\sim}{y} = 0$

vectors $\underset{\sim}{x} \perp \underset{\sim}{y}$ $\hookrightarrow$ perpendicular

* Def 2.6

The trace of a matrix $A_{p\times p}$ $tr(A) = \sum_{i=1}^{k} a_{ii}$ — only applied for square matrices

We have $tr(A+cB) = tr(A) + c\,tr(B)$

$tr(AB) = tr(BA)$

$tr(AA^T) = tr(A^TA) = \sum_{i,j}^{p} a_{ij}^2$

$tr(A \pm B) = tr(A) \pm tr(B)$

$tr(A_1 A_2 \dots A_n) = tr(A_{2+1} \dots A_n A_1 \dots A_2)$

$A = [\underset{\sim}{a}_1 | \underset{\sim}{a}_2 | \dots | \underset{\sim}{a}_p]$ then $\underset{\sim}{a}_i^T \underset{\sim}{a}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$

* Def 2.7

Let A be a square matrix | If we have $A\underset{\sim}{x} = \lambda \underset{\sim}{x}$, $\lambda \in \mathbb{C}$ for some $\underset{\sim}{x} \neq \underset{\sim}{0}$ | then $\begin{cases} \underset{\sim}{x}: \text{is an eigenvector} \\ \lambda: \text{associated eigenvalue} \end{cases}$

* We have $\Leftrightarrow (A - \lambda I)\underset{\sim}{x} = \underset{\sim}{0}$

i) A is singular $\Leftrightarrow \lambda = 0$ is an eigenvalue

ii) $\text{rank}(A) = \#$ nonzero eigenvalues.

iii) $\#$ of independent eigenvector $\leq$ size A

iv) $\sum_{i=1}^{p} \lambda_i = \text{trace}(A)$ $\quad$ $\prod_{i=1}^{p} \lambda_i = \det A$

A matrix has size p has p eigenvalues.

* If A symmetric $\begin{cases} \text{i) } \lambda \in \mathbb{R}, \ i = \overline{1,p} \\ \text{ii) } \#\text{ linear independent eigenvectors} = p \\ \text{iii) If } e_i = \frac{x_i}{\|x_i\|} \text{ then } \forall \lambda_i \neq \lambda_j, \ \vec{e}_i \text{ and } \vec{e}_j \text{ are orthogonal} \\ \text{and } \exists \text{ a set of } p \text{ linear independent eigenvectors.} \end{cases}$

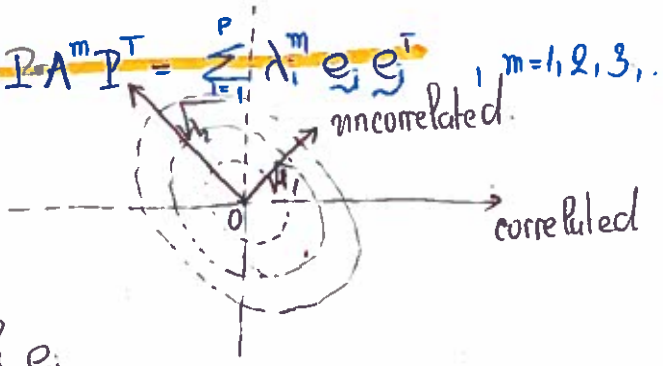
in some cases, we may have repeated eigenvalues.

* Theorem 2.1 (Spectral decomposition theorem for matrices) eigendecomposition

Let A be symmetric then $\exists$ an orthogonal matrix $P = [e_1, \dots, e_p]$ $(i.e\ P^T P = I)$

such that $A = \sum_{i=1}^{p} \lambda_i e_i e_i^T = P \Lambda P^T = P \Lambda P^{-1} \Rightarrow A^m = P \Lambda^m P^T = \sum_{i=1}^{p} \lambda_i^m e_i e_i^T$, $m = 1, 2, 3, \dots$

where $\begin{cases} \Lambda = \text{diag}(\lambda_i) = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{pmatrix} \\ \lambda_i: \text{eigenvalue of } A \text{ associated with } e_i \end{cases}$



positive definite $\Rightarrow$ symetric non singular

* Def 2.8

Let A be symmetric, then $\underset{\sim}{x}^T A \underset{\sim}{x}$ is a quadratic form.

* Def 2.9

- Suppose you call $\begin{cases} \underset{\sim}{x} \neq 0 \\ A \text{ symmetric} \end{cases}$ the QF $\underset{\sim}{x}^T A \underset{\sim}{x} > 0$ then A is positive definite (pd) $\xleftarrow{\text{def.}}$
- If $\underset{\sim}{x}^T A \underset{\sim}{x} \geq 0$ then A positive semi definite (psd)

* Theorem 2.2

positive semi definite $\Leftrightarrow \lambda_i \geq 0, \forall i$

Let A be a nonsingular + symmetric, then A is positive definite $\Leftrightarrow \lambda_i > 0, \forall i = \overline{1,p}$

positive definite $\Leftrightarrow$ { nonsingular, symmetric, $\lambda_i > 0, \forall i$

$\begin{cases} A\underset{\sim}{x} = 0 \Rightarrow \underset{\sim}{x}^T A \underset{\sim}{x} = 0 \Rightarrow \text{positive semi def} \\ \text{Actually when } A \text{ is singular} \Rightarrow A \text{ can not be positive definite} \end{cases}$

- Proof

$$\underset{\sim}{x}^T A \underset{\sim}{x} = x^T \Gamma \Lambda \Gamma^T x = (\Gamma^T x)^T \Lambda (\Gamma^T x) = y^T \Lambda y = \sum_{i=1}^{p} \lambda_i y_i^2 \quad \left(\text{can choose } y \text{ arbitrary } x = \Gamma y\right)$$

$\Rightarrow$ $y_i^2 > 0$ choose $y = \begin{pmatrix} 0 \\ 1 \leftarrow i \\ 0 \end{pmatrix}$ $\quad x_i^T A x_i > 0$ for all $x \neq 0 \quad \Rightarrow \lambda_i > 0$

$\Leftarrow$ Trivially if $\lambda_i > 0 \quad i = 1, 2, \ldots, p$ for $y \neq 0$

$$0 < \sum_{i=1}^{p} \lambda_i y_i^2 = x^T A x \Rightarrow A \text{ is positive definite}$$

* Theorem 2.3

Let A be pd/psd we define $A^{1/2} := \Gamma \Lambda^{1/2} \Gamma^T = \sum_{i=1}^{p} \sqrt{\lambda_i}\, e_i e_i^T$

(Γ: orthogonal matrix created by eigenvectors of A; Λ: diagonal matrix, contains all eigenvalues of A)

well defined since positive def $\Leftrightarrow \lambda_i > 0, \forall i$

then $A^{1/2}$ is the unique p.d matrix B satisfying $(A^{1/2})^2 = A$

* Theorem 2.4 (Sylvesters Criterion)

Let $A = \begin{bmatrix} A_m & \vdots \\ \cdots & \cdots \\ & p \end{bmatrix}_p$

A is p.s. $\Leftrightarrow$ { A is symmetric. all leading $m \times m$ ($m = \overline{1,p}$) matrices $A_{m\times m}$ have $\det A_{m\times m} > 0$

* Example 2.1

Let $R = \begin{bmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{bmatrix}$

$A_{(1)} = 1 \quad \det A_{(1)} = 1 > 0$

$A_{(2)} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \quad \det A_{(2)} = 1 - \rho^2 > 0$ when $|\rho| < 1$.

$A_{(3)} = A \quad \det A_{(3)} = (1-\rho^2) - \rho(\rho - \rho^3) + \rho^2(\rho^2 - \rho^2)$

$= 1 - 2\rho^2 + \rho^4$

$= (1-\rho^2)^2 > 0 \quad , |\rho| \neq 1$.

So by S.C Γ is p.d $\Leftrightarrow |\rho| < 1$.

$$X_t = \rho X_{t-1} + Z_t$$

* Example:

Consider any random vector $\underset{\sim}{X} \in \mathbb{R}^n = \begin{bmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{bmatrix}$ $\mu = E(\underset{\sim}{X}) = \begin{bmatrix} E(X_{11}) \\ E(X_{21}) \\ \vdots \\ E(X_{n1}) \end{bmatrix}$ $S = E\left((X-\mu)(X-\mu)^T\right)$

Then we have S is positive (semi) definite

• Proof

$\forall \underset{\sim}{x} \in \mathbb{R}^n$, $x^T S x = x^T E\left((X-\mu)(X-\mu)^T\right) x = E\left(x^T (X-\mu)(X-\mu)^T x\right) =$

$= E\left(\left[x^T (X-\mu)\right]\left[x^T (X-\mu)\right]^T\right) = Cov\left(\right.$

* 25 Random vectors and matrices

* **Def:**
  - A random vector $\underset{\sim}{X}_{1\times p} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$ where $X_i, i=\overline{1,p}$ are random variables
  - A random matrix $\underset{\sim}{X}_{n\times p} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix}$ where X_{ij} $i=\overline{1,n}$, $j=\overline{1,p}$ are random variables.

* The expected of a random matrix $E(\underset{\sim}{X}) = \begin{bmatrix} E(X_{11}) & E(X_{12}) & \dots & E(X_{1p}) \\ E(X_{21}) & E(X_{22}) & \dots & E(X_{2p}) \\ \vdots & \vdots & & \vdots \\ E(X_{n1}) & E(X_{n2}) & \dots & E(X_{np}) \end{bmatrix}$

  - For f is a function $E(f(\underset{\sim}{X})) = \left[E(f(X_{ij})) \right]_{ij}$
  - Let $\underset{\sim}{X}, \underset{\sim}{Y}$ be random matrices, $\underset{\sim}{A}, \underset{\sim}{B}$ be constant matrices. Then $E(\underset{\sim}{X}+\underset{\sim}{Y}) = E(\underset{\sim}{X}) + E(\underset{\sim}{Y})$
    $E(\underset{\sim}{A}\underset{\sim}{X}\underset{\sim}{B}) = \underset{\sim}{A}\, E(\underset{\sim}{X})\, \underset{\sim}{B}$

26 Mean vectors and covariance matrices

* **Def**
  - $\underset{\sim}{X}_{p\times 1}$ is a random vector $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$ where $X_i, i=\overline{1,p}$ are random variables.
  - $\underset{\sim}{\mu}_X = E(\underset{\sim}{X}) = \begin{bmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_p) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}$ ($\underset{\sim}{\mu}_X$: mean vector) $\quad \mu_i = E(X_i) = \forall i=\overline{1,p}$
  - $\Sigma_X = V(\underset{\sim}{X}) = \mathrm{cov}(\underset{\sim}{X}) = E\left((\underset{\sim}{X}-\underset{\sim}{\mu}_X)(\underset{\sim}{X}-\underset{\sim}{\mu}_X)^T\right) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \dots & \sigma_{2p} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \dots & \sigma_{3p} \\ \vdots & \vdots & \vdots & & \vdots \\ \sigma_{p1} & \sigma_{p2} & \sigma_{p3} & \dots & \sigma_{pp} \end{bmatrix}_{p\times p} = V^{1/2} \rho_X V^{1/2}$
    (Σ_X: covariance matrix, variance matrix)
    $\sigma_{ii} = V(X_i) = E\left((X_i-\mu_i)^2\right)$
    $\sigma_{ij} = \mathrm{cov}(X_i, X_j) = E\left[(X_i-\mu_i)(X_j-\mu_j)\right]$

* **Def 2.4**
  Two (real valued) rvs X, Y are independent $\overset{\text{def}}{\iff} F_{X,Y}(x,y) = F_X(x)\, F_Y(y) \quad \forall x,y \in \mathbb{R}$

* **Theorem 2.2**
  If X, Y are independent $\Rightarrow \mathrm{cov}(X,Y) = 0$
  - $\mathrm{cov}(X,Y) = 0 \not\Rightarrow$ independent

Ex:

X \ Y	0	1	2
0	0	1/4	0
1	1/4	0	1/4
2	0	1/4	0

X and Y are not independent

* Def 2.5
p- real valued rvs $X_1, \dots, X_p$ are multually independent $F_{\underset{\sim}{X}}(\underset{\sim}{x}) = \prod F_{X_i}(x_i) \quad \forall \underset{\sim}{x} \in \mathbb{R}^p$

* Def
The two random (vectors) $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ are independent $\overset{\text{def}}{\iff} F_{\underset{\sim}{X},\underset{\sim}{Y}}(\underset{\sim}{x},\underset{\sim}{y}) = \underbrace{F_{\underset{\sim}{X}}(\underset{\sim}{x})}_{X_1, \dots, X_p \text{ don't need to be multually independent}} \cdot F_{\underset{\sim}{Y}}(\underset{\sim}{y})$

- If $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ are independent then

$$\Rightarrow \quad \Sigma \begin{bmatrix} \underset{\sim}{X} \\ \underset{\sim}{Y} \end{bmatrix} = \left[\begin{array}{c|c} \Sigma_{\underset{\sim}{X}} & 0 \\ \hline 0 & \Sigma_{\underset{\sim}{Y}} \end{array}\right]$$

$\Leftarrow$ (*)

- Def of population correlation matrix

When $\Sigma_X = V(X) = C(X,X) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix}$

Then we consider (population) correlation coefficient $\rho_{ik} = \frac{\sigma_{ik}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{kk}}}$.

Then the (population) correlation matrix $\rho_X = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & 1 & \cdots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \cdots & 1 \end{bmatrix}$ where

$\rho_{ik} = \text{correlation}(X_i, X_k) = \frac{\sigma_{ik}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{kk}}} = (V^{1/2})^{-1} \Sigma (V^{1/2})^{-1}$

- Let $V_{p\times p}$ standard deviation matrix $V^{1/2} = \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & 0 \\ 0 & \sqrt{\sigma_{22}} & 0 \\ 0 & 0 & \sqrt{\sigma_{pp}} \end{bmatrix}$ $\quad V = [\text{diag}(\sigma_{ii})]$

$\Sigma_X = V^{1/2} \rho_X V^{1/2} \qquad \rho_X = V^{-1/2} \Sigma V^{-1/2}$

(§7.5) The mean vector and covariance matrix for linear combinations of r.v.s.

*** Theorem 2.3 (Linear combination)**

Let $\underset{\sim}{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix}$ ← number vector, $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$ random vector

then $\underset{\sim}{c}^T \underset{\sim}{X} = [c_1\ c_2 \dots c_p] \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \sum_{i=1}^{p} c_i X_i$ is a linear combination of p random variables $X_1, \dots, X_p$

Then we have a) $E(\underset{\sim}{c}^T \underset{\sim}{X}) = \underset{\sim}{c}^T E(\underset{\sim}{X})$

b) $V(\underset{\sim}{c}^T \underset{\sim}{X}) = \underset{\sim}{c}^T \Sigma_x \underset{\sim}{c}$

*** Example 2.17**

Let $X_1, \dots, X_p \overset{iid}{\sim} (\mu, \sigma^2)$

Then we have $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n} \mathbf{1}^T \underset{\sim}{X}$ where $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$ $\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$.

Note that $\underset{\sim}{\mu} = \mu_x = \begin{bmatrix} \mu \\ \mu \\ \vdots \\ \mu \end{bmatrix}$ $\Sigma_x = \begin{bmatrix} \sigma^2 & & 0 \\ & \sigma^2 & \\ 0 & & \ddots \\ & & & \sigma^2 \end{bmatrix}$ since $X_1, \dots, X_p$ iid (μ, σ^2)

Then we have

- $E(\bar{X}) = E\left(\frac{1}{n}\mathbf{1}^T \underset{\sim}{X}\right) = \frac{1}{n}\mathbf{1}^T E(\underset{\sim}{X}) = \frac{1}{n}\mathbf{1}^T \mu_x = \frac{1}{n}\mathbf{1}^T \mu \mathbf{1} = \frac{1}{n}\mu\, \mathbf{1}^T \mathbf{1} = \frac{1}{n}\mu n = \mu$.
- $V(\bar{X}) = V\left(\frac{1}{n}\mathbf{1}^T \underset{\sim}{X}\right) = \left(\frac{1}{n}\mathbf{1}^T\right) V(\underset{\sim}{X}) \left(\frac{1}{n}\mathbf{1}\right) = \frac{1}{n^2}\mathbf{1}^T V(\underset{\sim}{X}) \mathbf{1} = \frac{1}{n^2} V(\mathbf{1}^T \underset{\sim}{X})$

$= \frac{1}{n^2}\mathbf{1}^T (\sigma^2 I)\mathbf{1} = \frac{\sigma^2}{n^2}\mathbf{1}^T I \mathbf{1} = \frac{\sigma^2}{n^2}\mathbf{1}^T \mathbf{1} = \frac{\sigma^2}{n^2} n = \frac{\sigma^2}{n}$ □ example.

*** Explain the idea of theorem 2.3 in 2×2 case.**

- $E(a_1X_1 + a_2X_2) = a_1 EX_1 + a_2 E(X_2) \Leftrightarrow E\left([a_1\ a_2]\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}\right) = [a_1\ a_2]\begin{bmatrix} EX_1 \\ EX_2 \end{bmatrix}$
- $V(a_1X_1 + a_2X_2) = a_1^2 V(X_1) + a_2^2 V(X_2) + 2a_1a_2\, cov(X_1, X_2) = a_1^2\sigma_{11} + a_2^2\sigma_{12} + 2a_1a_2\sigma_{12}$

$\Leftrightarrow V\left([a_1\ a_2]\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}\right) = [a_1\ a_2]\begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$.

*** Theorem 2.3 (General case)**

Now consider $C_{k\times p} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1p} \\ c_{21} & c_{22} & \dots & c_{2p} \\ \vdots & \vdots & & \vdots \\ c_{k1} & c_{k2} & \dots & c_{kp} \end{bmatrix}_{p\times p}$ ← constant

$d_{k\times 1} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix}$

Then Let $\underset{\sim}{Z} = \underset{\sim}{C}\underset{\sim}{X} + \underset{\sim}{d}$

$= \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1p} \\ c_{21} & c_{22} & \dots & c_{2p} \\ \vdots & & & \vdots \\ c_{k1} & c_{k2} & \dots & c_{kp} \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix}$

$Z_1 = c_{11}X_1 + c_{12}X_2 + \dots + c_{1p}X_p = \sum_{i=1}^{p} c_{1i}X_i$

Then $\begin{cases} \mu_Z = E(\underset{\sim}{C}\underset{\sim}{X} + \underset{\sim}{d}) = \underset{\sim}{C}E(\underset{\sim}{X}) + \underset{\sim}{d} \\ V_Z = V(\underset{\sim}{C}\underset{\sim}{X} + \underset{\sim}{d}) = \underset{\sim}{C}V(\underset{\sim}{X})\underset{\sim}{C}^T \end{cases}$

* Example 2.27

Let $\underset{\sim}{X}_{p\times 1} \sim (\mu_X, \Sigma_X)$

$$\underset{\sim}{Z}_{p\times 1} = \Sigma_X^{-1/2}(\underset{\sim}{X} - \underset{\sim}{\mu}_X) = \frac{X - \mu_X}{\sqrt{\Sigma_X}}$$

then $E(Z_p) = \underset{p\times p}{\Sigma_X^{-1/2}} E(\underset{\sim}{X} - \mu_X) = \Sigma_X^{-1/2}\underbrace{[E(X) - \mu_X]}_{=\underset{\sim}{0}_{p\times 1}} = \underset{\sim}{0}_{p\times 1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

$$V(Z_p) = V\left(\Sigma_X^{-1/2}(X - \mu_X)\right) = \Sigma_X^{-1/2} V(X - \mu_X)\left(\Sigma_X^{-1/2}\right)^T =$$

$$= \Sigma_X^{-1/2}\,\Sigma_X\left(\Sigma_X^{-1/2}\right)^T = I$$

(Σ_X: symmetric)

---

* Theorem 2.4

Let $A_{p\times p}$ be (symmetric) constant matrix

$X_{p\times 1} \sim (\mu_X, \Sigma_X)$

Then $E(X^T A X) = tr(A\Sigma_X) + \mu_X^T A \mu_X$

$$E\left(\begin{bmatrix} X_1 & \cdots & X_p \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ & & & \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}\right) =$$

* Proof

• Assume $\mu_X = 0$

$$E(X^T A X) \ominus E\left(tr(X^T A X)\right) = E\left(tr(AXX^T)\right) \ominus tr\left(E(AXX^T)\right) = tr\left(A\,E(XX^T)\right) \ominus$$

$$= E(XX^T) = tr(A\Sigma_X)$$

2.7 p78 Matrix inequalities and maximization.

* **Cauchy Schwarz inequality**

Let $b_{p\times 1}$ & $d_{p\times 1}$ vectors { Then $(b'd)^2 \le (b'b)(d'd)$
equality $\Leftrightarrow$ $b = cd$ for some constant $c \neq 0$.

* **Extended Cauchy Schwarz inequality.**

Let $b_{p\times 1}$ & $d_{p\times 1}$
$B_{p\times p}$ positive definite matrix { Then $(b'd)^2 \le (b'Bb)(d'B^{-1}d)$
equality $\Leftrightarrow$ $b = cB^{-1}d$ for some constant $c \neq 0$
or $d = cBb$

* **Maximization lemma**

$B_{p\times p}$ positive definite
$d_{p\times 1}$: given vector
Then for any arbitrary nonzero vector $x_{p\times 1}$ { $\max_{x\neq 0} \frac{(x'd)^2}{x'Bx} = d'B^{-1}d$
with maximum attained when $x = cB^{-1}d$ $c\neq 0$ ($p\times p$, $p\times 1$)

* **Theorem 2.5 (Extended Cauchy Schwarz) (p79).**

Let $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ be $p\times 1$ vectors { Then $(X^TY)^2 \le (X^TAX)(Y^TA^{-1}Y)$
$\underset{p\times p}{A}$ positive definite

* **Theorem 2.6 Maximization lemma: (p80)**

Let A: positive definite { then $\max_{x\neq 0} \frac{(X^Td)^2}{X^TAX} = d^TA^{-1}d$
d: constant vector
$x \neq 0$ vector
c is any constant
some kind of normalization
the maximum attained when $x = cA^{-1}d$, for any constant $c\neq 0$

* **Theorem 2.7 (Maximization of Quadratic Forms for Points on the unit sphere.**

Let A be a positive definite, with eigenvalues $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_p \ge 0$
associated eigenvectors $e_1, e_2, \ldots, e_p$

Then
$\max_{x\neq 0} \frac{x^TBx}{x^Tx} = \lambda_1$
$\min_{x\neq 0} \frac{x^TAx}{x^Tx} = \lambda_p$
$\max_{x\perp e_1,\ldots,e_k} \frac{x^TAx}{x^Tx} = \lambda_{k+1}$

* Remark:

$X^TAX = (A^{1/2}X)^T(A^{1/2}X)$

C3. Sample geometry and random sampling

3.1 Introduction

3.2 The geometry of the sample

There are n vector X_i, X_i is a $(p\times 1)$ vector

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$$

* A single multivariate observation (group of n people, p variables) is a collection

If obs. The entire data set is

$$X_{n\times p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}_{n\times p} = \begin{bmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_n^T \end{bmatrix} = [Y_1 \,|\, Y_2 \,|\, \cdots \,|\, Y_p]$$

$\Rightarrow Y_1$, $\Rightarrow Y_2$

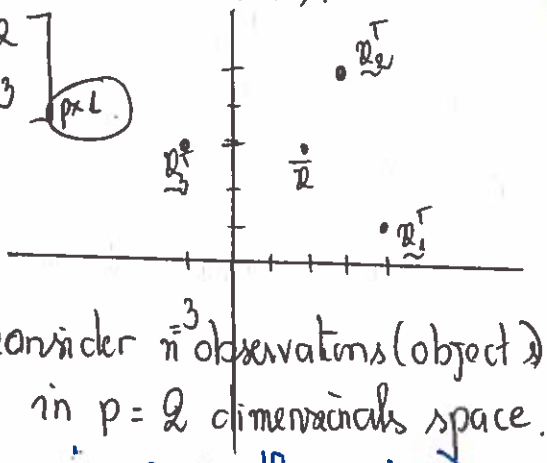
X_1^T: p variables observation of the 1st person.

Y_1: observation of the 1st variable (n people)

$$X_1 = \begin{bmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{1p} \end{bmatrix}_{p\times 1} \qquad Y_1 = \begin{bmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{bmatrix}_{n\times 1}$$

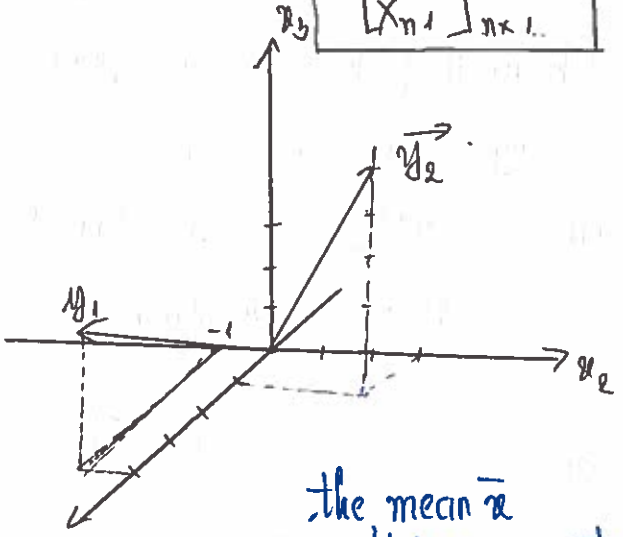
* Example 3.1 (Compute the mean vector).

$$X = \begin{bmatrix} 4 & 1 \\ -1 & 3 \\ 3 & 5 \end{bmatrix}_{n\times p} \qquad \bar{X} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}_{p\times 1}$$



if we consider $n=3$ observations (object) in $p=2$ dimensionals space.

(then the mean is in the center)



* From the second kind, we can find a geometrical interpretation of the process of finding a sample mean $y_1 = \begin{bmatrix} 4 \\ -1 \\ 3 \end{bmatrix}$ $y_2 = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$

the mean $\bar{x}$ can't be represented in this case

Then consider $1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}$ then $\left(\frac{1}{\sqrt{n}}\right)1$ has length 1.

consider the projection of y_i on the unit vector $\left(\frac{1}{\sqrt{n}}1\right)$ is $\bar{x}_i 1$

Remind: the projection of x on y is $\frac{x'y}{y'y}\vec{y}$

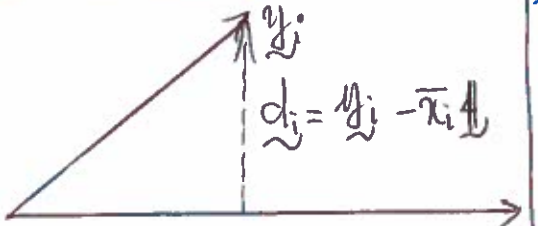
$$\left(\frac{y_i^T\left(\frac{1}{\sqrt{n}}1\right)}{\text{length}\left(\frac{1}{\sqrt{n}}1\right)}\frac{1}{\sqrt{n}}1 = y_i^T\frac{1}{\sqrt{n}}1\,\frac{1}{\sqrt{n}}1 = \frac{1}{n}\left(y_i^T 1\right)1 = \frac{1}{n}(x_{1i}+\cdots x_{ni}) = \bar{x}_i 1\right)$$

$$y_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix}$$

* Def 3.17

For each $\underset{\sim}{y}_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix}$, we define deviation vector $\underset{\sim}{d}_i = \underset{\sim}{y}_i - \bar{x}_i \underset{\sim}{1}$, $i = \overline{1,p}$

$\bar{\underset{\sim}{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix}$



- Note: $\underset{\sim}{d}_i$ and $\underset{\sim}{1}$ are orthogonal
- Note $\bar{x}_i = \frac{1}{n} \underset{\sim}{1}^T \underset{\sim}{y}_i \Rightarrow n\bar{x}_i = \underset{\sim}{1}^T \underset{\sim}{y}_i$ (*)
- Note $\bar{x}_i \underset{\sim}{1}$ is the projection of $\underset{\sim}{y}_i$ onto $\frac{1}{n}\underset{\sim}{1}$ $\quad \text{Proj}_{\frac{1}{n}\underset{\sim}{1}} \underset{\sim}{y}_i = \bar{x}_i \underset{\sim}{1}$.

* Prove that $\underset{\sim}{d}_i$ and $\underset{\sim}{1}$ are orthogonal

$$\underset{\sim}{d}_i^T \underset{\sim}{1} = (\underset{\sim}{y}_i - \bar{x}_i \underset{\sim}{1})^T \underset{\sim}{1} = \underset{\sim}{y}_i^T \underset{\sim}{1} - \bar{x}_i \underset{\sim}{1}^T \underset{\sim}{1} = \underset{\sim}{1}^T \underset{\sim}{y}_i - \bar{x}_i n \overset{(*)}{=} 0$$

* The length of the deviation square $\quad \perp_{d_i}^2 = \underset{\sim}{d}_i^T \underset{\sim}{d}_i = \sum_{j=1}^{n} (x_{ji} - \bar{x}_i)^2$

- For any i and k, $\underset{\sim}{d}_i^T \underset{\sim}{d}_k = \sum_{j=1}^{n} (x_{ij} - \bar{x}_i)(x_{kj} - \bar{x}_k) = \perp_{d_i} \perp_{d_k} \cos\theta_{ik}$

Then $r_{ik} = \dfrac{\underset{\sim}{d}_i^T \underset{\sim}{d}_k}{\sqrt{\underset{\sim}{d}_i^T \underset{\sim}{d}_i}\sqrt{\underset{\sim}{d}_k^T \underset{\sim}{d}_k}} = \dfrac{\frac{1}{n-1}\sum (x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)}{\sqrt{\frac{1}{(n-1)}\sum (x_{ij} - \bar{x}_i)^2 \frac{1}{(n-1)}\sum (x_{kj} - \bar{x}_k)^2}} = \dfrac{\perp_{d_i} \perp_{d_k} \cos(\theta_{ik})}{\sqrt{\perp_{d_i}^2 \perp_{d_k}^2}} =$

$= \cos(\theta_{ik})$

Then $r_{ik} = \cos\theta_{ik} = \cos(\widehat{d_i, d_k})$

* $[S_n]_{ik} = \frac{1}{(n-1)} \underset{\sim}{d}_i^T \underset{\sim}{d}_k$.

$[S_n] = \frac{1}{(n-1)} D^T D$ where $D = [d_1 | d_2 | \dots | d_p]$.

3.3 Random Samples and the expected values of the sample mean and covariance matrix

Def 3.2

A sequence including n $p \times 1$ random vectors $\underset{p\times 1}{X_1}, \ldots, \underset{p\times 1}{X_n}$ is a random sample if { they are independent; they have common distribution } (i.e dist of each r.v. vector is the same)

Theorem 3.1

Let $X_1, X_2 \ldots, X_n$ be a random sample from (μ, Σ). Then

a) $E(\bar{X}) = \mu$

b) $V(\bar{X}) = \frac{1}{n}\Sigma$

c) $E(S_n) = \Sigma$

Note that $s_{ik} = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)$ and $[S_n] = [s_{ik}]$

$$S_n = \frac{1}{n-1}\sum_{j=1}^{n}(X_j - \bar{X})(X_j - \bar{X})^T$$

3.4/p123. Generalized Variance.

Idea: $S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{12} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1p} & s_{2p} & \cdots & s_{pp} \end{bmatrix} = [s_{ik}]_{p\times p}$ $\quad s_{ik} = \frac{1}{(n-1)} \sum_{j=1}^{n} (x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)$

square symmetric

there are $p + \frac{1}{2}p(p-1)$ (s_{ik})

where $\underset{\sim}{x}_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix}$ n^{th} persons, j^{th} variable

$S \leftarrow \frac{1}{n-1}$

$S_n \leftarrow \frac{1}{n}$

* Def 3.3

The generalized sample variance is Generalized var $= \det(S_n) = |S_n|$

* Note:

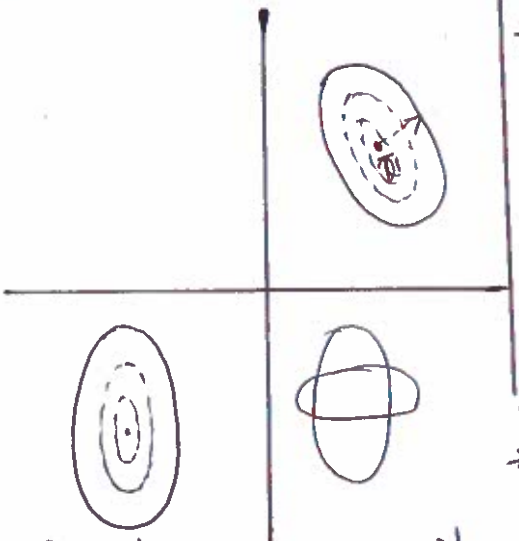
Let $V(d_1, .., d_p)$ be the volume spanned by $d_1, .., d_p$

Then $|S_n| = \frac{1}{(n-1)^p} |D|^2 = \frac{1}{(n-1)^p} V^2_{(d_1, .., d_p)}$

* Recall: the quantity $(\underset{\sim}{x} - \bar{x})^T S^{-1} (\underset{\sim}{x} - \bar{x})$ has ellipse level curves

→ Volume$\{\underset{\sim}{x} \mid (\underset{\sim}{x} - \bar{x})^T S^{-1} (\underset{\sim}{x} - \bar{x}) \le c\} = k_p |S|^{1/2} c^p$ $\quad \propto |S|^{1/2}$

$k_p = \frac{2\pi^{p/2}}{p\,\Gamma(p/2)}$



allow to compare even one does not strictly contain another.

* When is the generalized sample variance equal 0

(If some subset of v's is a linear combination of the other.

Ex: $X_1 = 0.5X_2 + 1.3X_3$

or $X_1 = 0X_2 + 0X_3$

or X_1 is a degenerated random variable

* Theorem 3.2

$|S| = \left(\prod_{i=1}^{p} s_{ii}\right) |R| = \prod_{i=1}^{p} \lambda_i$ where λ_i are eigenvalues of S_n (make sure units are the same)

* Def 3.2

The total (sample) variance is $tr(S_n) = \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} s_{ii} = \frac{1}{n-1} \sum_{i=1}^{p} \underset{\sim}{d}_i^T \underset{\sim}{d}_i$

$tr(R) = \sum \lambda_{ii} = \sum_{i=1}^{p} 1 = p$ (always)

does not depend on correlation structure

4. Multivariate normal distribution

* Recall univariate normal pdf $X \sim N(\mu, \sigma^2)$ $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

- $\frac{(x-\mu)^2}{\sigma^2} = (x-\mu)(\sigma^2)^{-1}(x-\mu)$ quadratic form
- Replace with $(\underset{\sim}{x}-\underset{\sim}{\mu})^T \Sigma^{-1} (\underset{\sim}{x}-\underset{\sim}{\mu})$

* Then we have multivariate normal distribution

$\underset{\sim}{X}_{p\times 1} \sim N(\underset{\sim}{\mu}_{p\times 1}, \Sigma_{p\times p})$, Σ positive definite (then density)
positive semi definite (generally)

has pdf $f_{\underset{\sim}{X}}(\underset{\sim}{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu})^T \Sigma^{-1} (\underset{\sim}{x}-\underset{\sim}{\mu})}$, $\underset{\sim}{\mu} \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d\times d}$ (pd.), $\underset{\sim}{x} \in \mathbb{R}^d$

(*) that (*) has expression is the square of the generalized distance from $\underset{\sim}{x}$ to $\underset{\sim}{\mu}$.

* Example 4.1 The bivariate normal ($p=2$) $\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$ X_1, X_2 ind $\Leftrightarrow \rho = 0$

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{\sigma_1^2\sigma_2^2(1-\rho^2)}} \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) \right] \right\}$$

- Note: X_1, X_2 independent $\Leftrightarrow \rho = 0$

* Theorem 4.1

Let $\underset{\sim}{X} = \begin{bmatrix} \underset{\sim}{X}_1 \\ \underset{\sim}{X}_2 \end{bmatrix} \sim N_p(\underset{\sim}{\mu}, \Sigma)$

then X_1 and X_2 are independent $\Leftrightarrow \Sigma = \begin{bmatrix} \Sigma_{X_1} & 0 \\ 0 & \Sigma_{X_2} \end{bmatrix}$

* Theorem 4.2:

The contours of constant density for p-variate normal are the ellipsoids such that $(\underset{\sim}{x}-\underset{\sim}{\mu})^T \Sigma^{-1} (\underset{\sim}{x}-\underset{\sim}{\mu}) = c^2$
- center at μ
- having axes $\pm c\sqrt{\lambda_i}\, e_i$ where $\Sigma \underset{\sim}{e}_i = \lambda_i \underset{\sim}{e}_i$

The axes of the ellipsoid are in the directions of the eigenvectors of Σ
the length of the j^{th} longest axis $\propto \sqrt{\lambda_j}$

* Result 4.4/153

If Σ positive definite $\Rightarrow \Sigma^{-1}$ exists and is

(Σ has eigenvalue $\lambda \Leftrightarrow \Sigma^{-1}$ has eigenvalue $\left(\frac{1}{\lambda}, \underset{\sim}{e}\right)$) $= (\lambda, \underset{\sim}{e})$

* Example 4.2.7

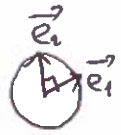
Let $X_2 \sim N_2\left(\underset{\sim}{\mu}, \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{11} \end{bmatrix}\right)$ ← equal

- Compute the eigenvalues of Σ.

$$\det(\Sigma - \lambda I) = \det\begin{pmatrix} \sigma_{11}-\lambda & \sigma_{12} \\ \sigma_{12} & \sigma_{11}-\lambda \end{pmatrix} = (\sigma_{11}-\lambda)^2 - \sigma_{12}^2 \Rightarrow \lambda = \sigma_{11} \pm \sigma_{12}$$

- If $\sigma_{12} = 0 \Rightarrow \lambda_1 = \lambda_2$

e_1 and e_2 can be any orthogonal unit vectors



- If $\sigma_{12} > 0 \Rightarrow \lambda_1 > \lambda_2$ $\quad \sigma_{12} < 0 \Rightarrow \lambda_1 < \lambda_2$

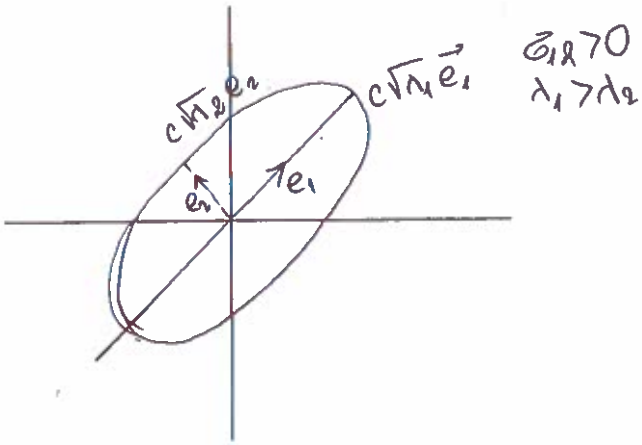
- Compute the eigenvector when $\sigma_{12} \neq 0$

$$(\lambda_1 I - \Sigma)\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = 0 \Leftrightarrow \begin{pmatrix} \sigma_{12} & -\sigma_{12} \\ -\sigma_{12} & \sigma_{12} \end{pmatrix}$$

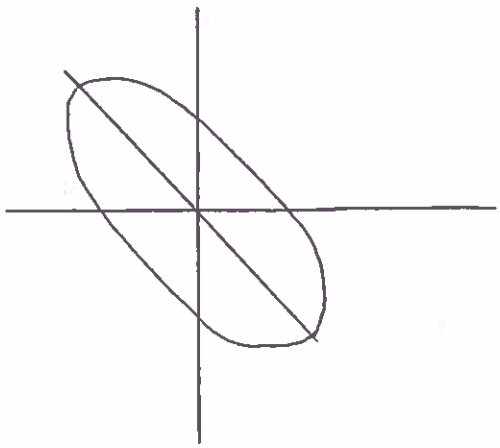
$$\Rightarrow e_1 \propto \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow e_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

$$(\lambda_2 I - \Sigma)\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = 0 \Leftrightarrow \begin{pmatrix} -\sigma_{12} & -\sigma_{12} \\ -\sigma_{12} & -\sigma_{12} \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow e_2 \propto \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow e_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$$



$\sigma_{12} > 0$
$\lambda_1 > \lambda_2$



* Properties of

Let $\underset{p\times 1}{X} \sim N_p(\mu, \Sigma)$ Σ p.s.d ($\Sigma \geq 0$) then

a) $\underset{p\times 1}{X} + \underset{p\times 1}{c} \sim N_p(\mu + c, \Sigma)$

b) $\underset{q\times 1}{AX} \sim N_q(\underset{q\times 1}{A\mu}, \underset{q\times q}{A\Sigma A^T})$ (A: $q\times p$, X: $p\times 1$)

c) If $\Sigma > 0$ (Σ is p.d), then $Y = \Sigma^{-1/2}(X-\mu) \sim N_p(\underset{p\times 1}{0}, \underset{p\times p}{I})$

d) If $\Sigma = \sigma^2 I$ (diagonal matrix with element σ^2) $\Rightarrow GX \sim N(G\mu, \sigma^2 I)$ (G: $q\times p$, X: $p\times 1$, $G\mu$: $q\times 1$, $\sigma^2 I$: $q\times q$)

G is orthogonal ($G^TG = I$) $[I \mid 0]$ (I: $q\times q$)

in the meaning that

$q \leq p$

* Alternative definition of multivariate normal distribution (MVN)

Let $Z_1, Z_2, \dots, Z_p \overset{iid}{\sim} N(0,1)$ Then Let $Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} \sim N(0, I)$

Then $\underset{p\times 1}{X} = \underset{p\times 1}{\mu} + \underset{p\times l}{L}\,\underset{l\times 1}{Z} \sim N_p(\underset{p\times 1}{\mu}, \underset{p\times p}{\Sigma = LL^T})$

* Definition 4.1 (for ChiSquare)

Let $Z_1, \dots Z_p \overset{iid}{\sim} N(0,1)$ Let $Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} \sim N(0, I)$

Let $X = \sum_{i=1}^{p} Z_i^2 = Z^TZ \sim \chi^2_p$

* Theorem 4.4

Let $\underset{p\times 1}{X} \sim N_p(\underset{p\times 1}{\mu}, \underset{p\times p}{\Sigma})$ Σ p.d.

Then $(X-\mu)^T(\Sigma)^{-1}(X-\mu) \sim \chi^2_p$

* Proof:

$(X-\mu)^T\Sigma^{-1}(X-\mu) = [\Sigma^{-1/2}(X-\mu)]^T[\Sigma^{-1/2}(X-\mu)] = Y^TY$ where $Y \sim N_p(0, I)$ theorem 4.3c

$\sim \chi^2_p$

* Theorem 4.5.7

Let $\Sigma > 0$ (Σ p.d.), $X \sim N_p(\mu, \Sigma)$ } Then $P\left\{(X-\mu)^T \Sigma^{-1}(X-\mu) \le \chi^2_{p,1-\alpha}\right\} = 1-\alpha$.

where $P\left(\chi^2_p \le \chi^2_p(c)\right) = c$

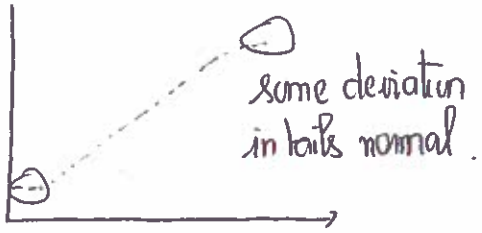
---

* Assessing Normality:

- Recall QQ plots: plot pairs $(X_{(j)}, q_{(j)})$ ← quantile
  - $X_{(j)}$: empirical, j^{th} order statistic
  - $q_{(j)}$: theoretical, can be anything. $q_{(j)} = F^{-1}(p_{(j)})$
  - $0 < p_{(j)} < 1$
  - $p_{(j)} < p_{(j+1)}$
  - $p_{(j)} / j/n \to 1$.

In R for normal $q(j) = \Phi^{-1}\left(\frac{j-0.375}{n+0.4}\right)$

If the iid data are normal, then the line should be straight



- For bivariate ps2 data, apply theorem 4.4

plot d_j^2 = j^{th} largest $(x_i - \bar{x}_i)^T S^{-1} (x_i - \bar{x}_i)$

$\left(d_j^2, \chi^2_p(p_{(j)})\right)$ or $\left(d_j^2, \sqrt{\chi^2_p(p_{(j)})}\right)$

both should be line through origin with slope 1. (include (0,0) in plot)

{ roystons test in MVN uses $p_{(j)} = \frac{j-.05}{n}$ }

---

* Theorem 4.6

Let $\underset{\sim}{X} \sim N_p(\underset{\sim}{\mu}, \Sigma)$ Σ p.d.

* Let $\underset{p\times 1}{\underset{\sim}{X}} = \begin{bmatrix} \underset{\sim}{X_1} \\ \hline \underset{\sim}{X_2} \end{bmatrix} \begin{matrix} p_1 \times 1 \\ p_2 \times 1 \end{matrix}$ $p_1 + p_2 = p$.

$\underset{\sim}{\mu} = \begin{bmatrix} \underset{\sim}{\mu_1} \\ \hline \underset{\sim}{\mu_2} \end{bmatrix}$

$\Sigma = \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right]$ $\Sigma_{21} = \Sigma_{12}^T$

Then $\underset{\sim}{X_1}\big|_{X_2 = x_2} \sim N_{p_1}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2),\ \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$

* Example

Let $\underset{\sim}{\mu} = \mu \underset{\sim}{1} = \begin{bmatrix} \mu \\ \mu \\ \mu \end{bmatrix}$ $\quad \Sigma = \begin{bmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{bmatrix}$

Let $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ \hline X_2 \\ X_3 \end{bmatrix} \sim N_3\left(\underset{\sim}{\mu}, \Sigma\right)$ Observe $X_1 = a$.

Find dist of $\left.\begin{bmatrix} X_2 \\ X_3 \end{bmatrix}\right| X_1 = a.$

- We have $\left.\underset{\sim}{\mu}_{\begin{bmatrix} X_2 \\ X_3 \end{bmatrix}}\right|_{X_1 = a} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1) = \mu\begin{bmatrix} 1-\rho \\ 1-\rho^2 \end{bmatrix} + a\begin{bmatrix} \rho \\ \rho^2 \end{bmatrix}$

$$\left.\Sigma_{\begin{bmatrix} X_2 \\ X_3 \end{bmatrix}}\right|_{X_1 = a} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} - \begin{bmatrix} \rho \\ \rho^2 \end{bmatrix}[1]^{-1}[\rho\ \rho^2] =$$

$$= \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} - \begin{bmatrix} \rho^2 & \rho^3 \\ \rho^3 & \rho^4 \end{bmatrix} = \begin{bmatrix} 1-\rho^2 & \rho-\rho^3 \\ \rho-\rho^3 & 1-\rho^2 \end{bmatrix} = (1-\rho^2)\begin{bmatrix} 1 & \rho \\ \rho & 1+\rho^2 \end{bmatrix}$$

Then $\left.\begin{bmatrix} X_2 \\ X_3 \end{bmatrix}\right|_{X_1 = a} \sim N_2\left(\mu\begin{bmatrix} 1-\rho \\ 1-\rho^2 \end{bmatrix} + a\begin{bmatrix} \rho \\ \rho^2 \end{bmatrix}, (1-\rho^2)\begin{bmatrix} 1 & \rho \\ \rho & 1+\rho^2 \end{bmatrix}\right)$

* Find $\left.X_3\right|_{\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}}$ and $\left.X_2\right|_{\begin{bmatrix} X_1 \\ X_3 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix}}$

* Theorem 4.3.

2) $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ \hline X_2 \end{bmatrix} \begin{matrix} p_1 \\ \\ p_2 \end{matrix} \sim N_{p_1+p_2}\left(\begin{bmatrix} \mu_1 \\ \hline \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$

Then $X_1 \sim N_{p_1}(\mu_1, \Sigma_{11})$

* Example 4.3

$X_2 | X_1 = a \sim N(\mu(1-\rho) + a\rho, (1-\rho^2))$

$X_3 | X_1 = a \sim N(\mu(1-\rho) + a\rho^2, (1-\rho^2)(1+\rho^2))$

* 4.3 Maximum likelihood estimation of Σ and μ.

* Let $\underset{\sim}{X}_1, \dots, \underset{\sim}{X}_n \sim N_p(\mu, \Sigma)$ ($\Sigma > 0$)

The likelihood is

$$L(\mu, \Sigma \mid \underset{\sim}{x}_1, \dots, \underset{\sim}{x}_n) = \prod_{j=1}^{n} \left(\frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\underset{\sim}{x}_j - \mu)^T \Sigma^{-1} (\underset{\sim}{x}_j - \mu)} \right)$$

$$= \frac{1}{(2\pi)^{np/2} |\Sigma|^{n/2}} e^{-\frac{1}{2} \sum_{j=1}^{n} (\underset{\sim}{x}_j - \mu)^T \Sigma^{-1} (\underset{\sim}{x}_j - \mu)}$$

* Theorem 4.7

$$\sum_{j=1}^{n} (\underset{\sim}{X}_j - \mu)^T \Sigma^{-1} (\underset{\sim}{X}_j - \mu) = tr\left[\Sigma^{-1} \left(\sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})(\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})^T \right) \right] + n \left[(\bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \mu) \right]$$

$$= tr\left[\Sigma^{-1} (n-1) S \right] + n (\bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \mu)$$

* Result 4.9/168 A symmetric, $\underset{\sim}{x}$ is a vector $\Rightarrow \underset{\sim}{x}^T A \underset{\sim}{x} = tr(\underset{\sim}{x}^T A \underset{\sim}{x}) = tr(A \underset{\sim}{x}^T \underset{\sim}{x})$

* Proof

$$\sum_{j=1}^{n} (\underset{\sim}{X}_j - \mu)^T \Sigma^{-1} (\underset{\sim}{X}_j - \mu) \overset{\text{insert } \bar{\underset{\sim}{X}}}{=\!=} \sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}} + \bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}} + \bar{\underset{\sim}{X}} - \mu)$$

$$= \sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})^T \Sigma^{-1} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}}) + \underbrace{\sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \mu) + \sum_{j=1}^{n} (\bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})}_{= 0 \text{ because } \sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})^T = \underset{\sim}{0}^T \text{ and } \sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}}) = \underset{\sim}{0}} +$$

$$+ \sum_{j=1}^{n} (\bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \mu)$$

$$= tr\left(\sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})^T \Sigma^{-1} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}}) \right) + n (\bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \mu)$$

$\overset{\Sigma^{-1} \text{ symmetric}}{=\!=}$

$$tr\left(\Sigma^{-1} \left(\sum_{j=1}^{n} (\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})(\underset{\sim}{X}_j - \bar{\underset{\sim}{X}})^T \right) + n \left((\bar{\underset{\sim}{X}} - \mu)^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \mu) \right) \right)$$

* Def 4.2 (Sufficiency)

$T(\underset{\sim}{X})$ is sufficient for θ if $P(\underset{\sim}{X} = \underset{\sim}{x} \mid T(\underset{\sim}{x}) = t, \theta) = P(\underset{\sim}{X} = \underset{\sim}{x} \mid T(\underset{\sim}{x}) = t)$

* Theorem 4.8 (Factorization theorem)

T is sufficient for $\theta \iff f_\theta(\underset{\sim}{x}) = h(\underset{\sim}{x})\, g_\theta(T(\underset{\sim}{x}))$

* Theorem 4.9

$(S, \bar{\underset{\sim}{X}})$ are jointly sufficient for μ, Σ

* Theorem 4.10 (this result will eventually allow us to obtain the maximum likelihood estimators of $\underset{\sim}{\mu}$ and $\underset{\sim}{\Sigma}$)

Let $\underset{p\times p}{B} > 0$, $b > 0$, a scalar $\Big\}$ $\frac{1}{|\Sigma|^b} e^{-\frac{1}{2}tr(\Sigma^{-1}B)} \leq \frac{1}{|B|^b}(2b)^{pb} e^{-bp}$ for all $\Sigma > 0$

equality holds only for $\Sigma = \frac{1}{2b} B$.

* Proof:

• Note that $tr(\Sigma^{-1}B) = tr(\Sigma^{-1}B^{1/2}B^{1/2}) = tr(B^{1/2}\Sigma^{-1}B^{1/2}) = \sum_{i=1}^{p} n_i$, eigenvalues of $B^{1/2}\Sigma^{-1}B^{1/2}$

• $|B^{-1/2}\Sigma^{-1}B^{-1/2}| = |B||\Sigma^{-1}|$

$\Rightarrow |\Sigma^{-1}| = \frac{|B^{-1/2}\Sigma^{-1}B^{-1/2}|}{|B|} = \frac{\prod_{i=1}^{p} n_i}{|B|}$

$\Rightarrow$ LHS $= \frac{1}{|\Sigma|^b} e^{-\frac{1}{2}tr(\Sigma^{-1}B)} = \frac{(\prod n_i)^b}{|B|^b} e^{-\frac{1}{2}\sum_{i=1}^{p} n_i} = \frac{1}{|B|^b}\prod n_i^b \prod e^{-\frac{1}{2}n_i} = \frac{1}{|B|^b}\prod (n_i^b e^{-\frac{1}{2}n_i})$

• Let $f(n) = n^b e^{-\frac{1}{2}n}$ (Gamma density) is maximized at $n = 2b$.

$\Rightarrow$ LHS $\leq \frac{1}{|B|^b}\prod_{i=1}^{p}((2b)^b e^{-b}) = \frac{1}{|B|^b}(2b)^{bp} e^{-bp}$

* Equality happens when $n_i = 2b$, $i = 1, \dots, p$

i.e the characteristic polynomial $B^{1/2}\Sigma^{-1}B^{1/2} \ominus (\lambda - 2b)^p = 2bI$

$B = 2b\Sigma \Leftrightarrow \Sigma = \frac{1}{2b} B$

* Theorem 4.11.

Let $\underset{\sim}{X}_1, \dots, \underset{\sim}{X}_n \sim N_p(\mu, \Sigma)$

The MLE of μ is $\hat{\mu} = \bar{X} = \frac{1}{n}\sum_{i=1}^{p}(X_i$

Σ is $\frac{n-1}{n}S = \frac{1}{n}\sum_{i=1}^{n}(\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})(\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})^T$

* Proof: We have the likelihood is

$L(\mu, \Sigma | \underset{\sim}{X}_1, \dots, \underset{\sim}{X}_n) = \frac{1}{(2\pi)^{np/2}}\Big(\underbrace{\frac{1}{|\Sigma|^{n/2}} e^{-\frac{1}{2}tr(\Sigma^{-1}(\sum_{j=1}^{n}(X_j - \bar{X})(X_j - \bar{X})^T))}}_{(*)}\Big) e^{\frac{-n}{2}\underbrace{(\bar{X}-\mu)^T\Sigma^{-1}(\bar{X}-\mu)}_{\geq 0 \text{ unless } \mu = \bar{X}}}$

(since $\Sigma > 0 \Rightarrow \Sigma^{-1} > 0$)

$\Rightarrow \hat{\mu} = \bar{X}$

• Now we want to maximize $(*)$

Put $B = \sum_{j=1}^{n}(X_j - \bar{X})(X_j - \bar{X})^T = (n-1)S$

then $(*) = \frac{1}{|\Sigma|^{n/2}} e^{-\frac{1}{2}(\Sigma^{-1}\underbrace{(n-1)S}_{B})} \leq \frac{1}{|B|^b}(2b)^{pb} e^{-bp}$ $\quad b = \frac{n}{2}$

equality holds when $\Sigma = \frac{1}{2b} B = \frac{1}{2\frac{n}{2}}(n-1)S = \frac{n-1}{n} S$ $\square$.

max at $\hat{\mu} = \bar{X}$ since $\Sigma \geq 0 \Rightarrow \Sigma^{-1} \geq 0$

$-(\bar{X}-\mu)^T \Sigma^{-1} (\bar{X}-\mu) \leq 0$ then equal 0 only when $\bar{X} = \mu$ i.e $\bar{X}_{MLE} = \mu$

- Follows from Theorem 4.10 with $b = \frac{n}{2}$ $B = \sum_{j=1}^{n} (X_j - \bar{X})(X_j - \bar{X})^T = (n-1)S$

$$\hat{\Sigma} = \frac{1}{2b} B = \frac{1}{2(\frac{n}{2})}(n-1)S = \frac{n-1}{n} S$$

* Theorem 4.12

Let A be symmetric $p \times p$

B $q \times p$

Suppose $X \sim N_p(\mu, \Sigma)$

Then ($\bar{X}$, fill in for X^2)

BX and $X^T A X$ are independent if

$$B\Sigma A = \underset{q \times p}{0}$$

* Theorem 4.13

Let $X_1, \ldots, X_n \overset{iid}{\sim} N_p(\mu, \Sigma)$

then $\hat{\mu} = \bar{X}$ and $\hat{\Sigma} = \frac{n-1}{n} S$ are independent.

Σ positive $\Rightarrow \exists \Sigma^{-1/2} \Rightarrow$ Var = Q

C5: Inferences about a mean vector

5.1 Introduction : p correlated variables must be analyzed jointly.

5.2 The plausibility of μ_0 as a value for a Normal population mean

* Recall for (univariate) variables.

Want to test $\begin{cases} H_0: \mu = \mu_0 \\ H_1: \mu \neq \mu_0 \end{cases}$

- If $X_1, \dots, X_n \sim$ Normal, $\bar{X} := \frac{1}{n}\sum_{i=1}^{n} X_i$ $\quad S^2 = \frac{1}{n-1}\sum_{j=1}^{n}(X_j - \bar{X})^2$

Then the test statistic

$t_{obs} = \frac{\bar{X} - \mu_0}{\frac{S}{\sqrt{n}}} \sim t_{n-1}$ We reject H_0 when $|t_{observe}| > t_{\frac{\alpha}{2}, n-1}$.

$t^2 = n(\bar{X} - \mu_0)(S^2)^{-1}(\bar{X} - \mu_0) \Rightarrow$ Reject H_0 when $n(\bar{X} - \mu_0)(S^2)^{-1}(\bar{X} - \mu_0) > t^2_{\frac{\alpha}{2}, n-1}$.

* The confidence interval $CI = \left\{ \mu_0, \left| \frac{\bar{X} - \mu_0}{\frac{S}{\sqrt{n}}} \right| < t_{\frac{\alpha}{2}, n-1} \right\} = \bar{X} \mp t_{\frac{\alpha}{2}, n-1}\frac{S}{\sqrt{n}}$

---

* Now, consider a problem with $(p \times 1)$ ~~vector~~ random vectors.

$T^2 = n(\underset{\sim}{\bar{X}} - \underset{\sim}{\mu_0})^T S^{-1} (\underset{\sim}{\bar{X}} - \underset{\sim}{\mu_0})$

* Chapter 2: Inference for the mean (C5+C6 in the book)

* Wishart distribution.

Suppose that $\underset{\sim}{X}_1, \dots, \underset{\sim}{X}_n \overset{iid}{\sim} N_p(\underset{\sim}{0}, \Sigma)$

then $W_n = \sum_{i=1}^{n} \underset{\sim}{X}_i \underset{\sim}{X}_i^T \sim \text{Wishart}(\Sigma, n)$

* Properties.

$W_1 \sim \text{Wishart}(\Sigma, m_1)$

$W_2 \sim \text{Wishart}(\Sigma, m_2)$

W_1 and W_2 are independent

Let $C_{q\times p}$: constant matrix

$\underset{\sim}{a}_{p\times 1}$ constant vector $\underset{\sim}{a}^T \Sigma \underset{\sim}{a} \neq 0$

Then

a) $W_1 + W_2 \sim \text{Wishart}(\Sigma, m_1 + m_2)$

b) $C W_1 C^T \sim W_p(C\Sigma C^T, m_1)$

c) $\dfrac{a^T W_1 a}{a^T \Sigma a} \sim \chi^2_{m_1}$

* In univariate case $X_1, X_2, \dots, X_n \sim N(0, \sigma^2)$

then the sample mean $\bar{X} \sim N\left(\mu, \dfrac{\sigma^2}{n}\right)$

$\dfrac{\sqrt{n}(\bar{X}-\mu)}{\sigma} \sim N(0,1)$ $\quad \left(\dfrac{\sqrt{n}(\bar{X}-\mu)}{\sigma}\right)^2 \sim \chi^2_1$

* In multivariate case $\underset{\sim}{X}_1, \underset{\sim}{X}_2, \dots, \underset{\sim}{X}_n \overset{iid}{\sim} N_p(\underset{\sim}{\mu}, \Sigma)$

Then $\bar{\underset{\sim}{X}} \sim N_p\left(\underset{\sim}{\mu}, \dfrac{1}{n}\Sigma\right)$ $\quad \Rightarrow \dfrac{\sqrt{n}(\bar{\underset{\sim}{X}} - \underset{\sim}{\mu})}{} \sim N_p(\underset{\sim}{0}, \Sigma)$

$(n-1)S \sim W_p(\Sigma, n-1)$

$\sqrt{n}\,\Sigma^{-1/2}(\bar{\underset{\sim}{X}} - \underset{\sim}{\mu}) \sim N_p(\underset{\sim}{0}, I)$

$n(\bar{\underset{\sim}{X}} - \underset{\sim}{\mu})^T \Sigma^{-1} (\bar{\underset{\sim}{X}} - \underset{\sim}{\mu}) \sim \chi^2_{(p)}$

(Σ: unknown)

Replace Σ^{-1} by S^{-1}

$n(\bar{X} - \mu)^T S^{-1} (\bar{X} - \mu) \overset{?}{\sim} \chi^2_p$

no.

> we can't use $\bar{X}$ to infer about μ
> ⇒ introduce T^2 Hotelling dist

* Page 117 4

When $\underset{\sim}{X}_1, \underset{\sim}{X}_2, \dots, \underset{\sim}{X}_n \sim N_p(\underset{\sim}{\mu}, \Sigma)$. Then

a) $\bar{X} \sim N_p\left(\mu, \dfrac{1}{n}\Sigma\right)$

$(n-1)S \sim W(\Sigma, n-1)$

2. Hotelling's T^2

* Suppose that (univariate) $X_1, \dots, X_n \overset{iid}{\sim} N(\mu, \sigma^2)$

We use $t = \frac{\bar{X} - \mu_0}{\frac{S}{\sqrt{n}}}$ as a test statistic (Remind test $\begin{cases} \mu = \mu_0 \\ \mu \neq \mu_0 \end{cases}$

C.I. $\bar{X} \pm t_{1-\frac{\alpha}{2}, n-1} \frac{S}{\sqrt{n}}$ equal tail

* For multivariate distribution

Let $\underset{\sim}{X} \sim N_p(\underset{\sim}{0}, \Sigma)$ } independent.
$W \sim W_p(\Sigma, n-1)$

Then $T^2 = \underset{\sim}{X}^T \left(\frac{1}{n-1} W\right)^{-1} \underset{\sim}{X} \sim T^2_{p, n-1}$: Hotelling T^2 distribution. ← not very user friendly

* Mahalanobis distance

$d(\underset{\sim}{x}, \underset{\sim}{y}) = \sqrt{(\underset{\sim}{x} - \underset{\sim}{y})^T S^{-1} (\underset{\sim}{x} - \underset{\sim}{y})}$

then $T^2 = n (\bar{\underset{\sim}{X}} - \underset{\sim}{\mu_0})^T S^{-1} (\bar{\underset{\sim}{X}} - \underset{\sim}{\mu_0})$

Reject H_0 when T^2 is large

compute from data numbers.

n : sample size

$\bar{X}, S$: sample mean, sample covariance matrix.

* Try to make connection to F

$$F = \frac{n}{n-1} \frac{n-p}{p} (\bar{\underset{\sim}{X}} - \underset{\sim}{\mu_0})^T S^{-1} (\bar{\underset{\sim}{X}} - \underset{\sim}{\mu_0}) \sim F_{p, n-p}$$

$$= \frac{(n-p)}{(n-1)p} T^2 \sim F_{p, n-p}$$

$T^2 \sim \frac{(n-1)p}{(n-p)} F_{p, n-p}$

Note: 1. $n > p$

2. S is non-singular

* Remind: Sample

$$\underset{\sim}{X} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ X_{21} & X_{22} & \dots & X_{2p} \\ & & & \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{X}_1^T \\ \underset{\sim}{X}_2^T \\ \vdots \\ \underset{\sim}{X}_n^T \end{bmatrix}_{n \times p} = [Y_1, Y_2, \dots, Y_p]_{n \times p}$$

There are n $[X_i]_{p \times 1}$ represent for n people with p variables.

$$\bar{\underset{\sim}{X}}_{p \times 1} = \begin{bmatrix} \bar{Y}_1 \\ \bar{Y}_2 \\ \vdots \\ \bar{Y}_p \end{bmatrix} = \frac{1}{n} \sum_{i=1}^{n} \underset{\sim}{X}_i \qquad S = \frac{1}{(n-1)} \sum_{i=1}^{n} (\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})(\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})^T$$

Two sample Hotelling's T^2 test.

Assume that the two independent groups of sample have the same sample variance.

Want to test.

$$\begin{cases} H_0: \mu_1 = \mu_2 \\ H_1: \mu_1 \neq \mu_2 \end{cases}$$

Test statistic $T^2 = \frac{n_1 n_2}{n_1 + n_2} (\bar{X}_1 - \bar{X}_2)^T S^{-1} (\bar{X}_1 - \bar{X}_2)$

where S: pooled correlation matrix $S = \frac{(n_1 - 1) S_1 + (n_2 - 1) S_2}{(n_1 + n_2 - 2)}$

---

* Make connection between Hotelling T dist to F

$$F = \underbrace{\frac{(n_1 + n_2 - p - 1)}{(n_1 + n_2 - 2) p} T^2}_{\text{observed}} \sim F_{p,\, n_1 + n_2 - p - 1}$$

We require $\begin{cases} n_1 + n_2 > p \\ S \text{ non singular} \end{cases}$

---

* One sample T^2 test.

We want to test $\begin{cases} H_0: \mu = \mu_0 \\ H_1: \mu \neq \mu_0 \end{cases}$

At level α of significance, we reject H_0 when $T^2 = n (\bar{x} - \mu_0)^T S^{-1} (\bar{x} - \mu_0) > \frac{(n-1)p}{(n-p)} F_{n-p}(\alpha$

4) Constant density Ellipses (page 220-221) (to construct confidence interval)

Flury (1997) gives an interpretation of constant density ellipses in terms of Mahalanobis distance. Reject if $T^2 > c^2$ / p285

We wish to find region of squared Mahalanobis distance s.t.

$$\Pr\left(\left(\bar{\underset{\sim}{X}} - \underset{\sim}{\mu}\right)^T \underset{\sim}{S}^{-1} \left(\bar{\underset{\sim}{X}} - \underset{\sim}{\mu}\right) \leq c^2\right) = (1-\alpha) \qquad c^2 = \frac{n-1}{n}\frac{p}{n-p} F_{(1-\alpha),p,n-p}$$

c^2 — $\underset{\sim}{S}^{-1}$: observe

$F_{1-\alpha,p,n-p}$ is the $(1-\alpha)$ quantile of the F distribution with df $p, n-p$; sample size n

* we want to compare this constant density ellipse with normal CI

Opt 2 — 12/06 Present — 12/10 Final

√ * The Simultaneous confidence intervals (p223) (care about within variable correlation)

$$\left(\underset{\sim}{a}^T \bar{\underset{\sim}{X}} \pm \sqrt{\frac{p(n-1)}{n(n-p)} F_{1-\alpha,p,n-p}\ \underbrace{\underset{\sim}{a}^T \underset{\sim}{S} \underset{\sim}{a}}_{\text{diag}}}\right)$$

to do the test, we construct this interval and then compare.

where $\underset{\sim}{X}_1, \ldots, \underset{\sim}{X}_n \sim N_p(\mu, \Sigma)$.

Given that Σ positive definite

Let $\underset{\sim}{a}_i = (0, \ldots 0, \underset{\uparrow i^{th}}{1}, \ldots 0)$

The the CI (Constant density ellipses).

$$\left(\bar{X}_i \pm \sqrt{\frac{p(n-1)}{n(n-p)} F_{1-\alpha,p,n-p}}\ S_i\right)$$

* Want to compare the constant density ellipses with t interval

- On at arithmetic intervals (regular t interval) ignore the correlation)

Let $i = 1, \ldots, p$

for μ_i, CI is $\left(\bar{X}_i \pm t_{n-1,\frac{\alpha}{2}} \sqrt{\frac{S_{ii}}{n}}\right)$ $S_{ii} = \text{diag}\,\Sigma$

- When the number m of specified component mean μ_i or $a^T\mu$ is small Bonferroni corrected CI's (χ^2 interval → when we have very large sample)

$$\underset{\sim}{a}^T \bar{\underset{\sim}{X}} \pm \sqrt{\chi^2_p(\alpha)} \sqrt{\frac{\underset{\sim}{a}^T \underset{\sim}{S}\, \underset{\sim}{a}}{n}}$$

5* Review of Anova

* Assume we have g random variables, $X_l \sim N(\mu_l, \sigma^2)$, $l = \overline{1,g}$ (same σ^2)

For each group $X_l \sim N(\mu_l, \sigma^2)$, we take a sample $X_{l1}, \dots, X_{l,n_l}$, $l = \overline{1,g}$

(The random samples are independent.)

* For $l = \overline{1,g}$, $X_l \sim N(\mu_l, \sigma^2)$ then $\mu_l = \overline{X_l} = \mu + \tau_l$

$X_l \sim N(\mu + \tau_l, \sigma^2)$

μ_l: l^{th} group mean; μ: overall mean; τ_l: l^{th} group effect (treatment effect).

---

* The response j in group l, $X_{lj} \sim N(\mu + \tau_l, \sigma^2)$.

$X_{lj} \sim N(\mu + \tau_l, \sigma^2)$

$X_{lj} = \mu + \tau_l + \varepsilon_{lj} = \mu + \tau_l + N(0, \sigma^2)$, where $\varepsilon_{lj} \sim N(0, \sigma^2)$.

* $\sum_{l=1}^{g} n_l \tau_l = 0$

---

* We decompose X_{lj} as

$$X_{lj} = \overline{X} + (\overline{X}_l - \overline{X}) + (X_{lj} - \overline{X}_l)$$

X_{lj}: observation; $\overline{X}$: overall sample mean (an estimator of μ); $\overline{X}_l - \overline{X}$: estimated treatment effect (an estimator of τ_l); $X_{lj} - \overline{X}_l$: residual (an estimate of error ε_{lj})

* $SST = SS_t + SSR$

Sum of square total = SS treatment + SS residual

$$\sum_{l=1}^{g}\sum_{j=1}^{n_l} (X_{lj} - \overline{X})^2 = \sum_{l=1}^{g} n_l (\overline{X}_l - \overline{X})^2 + \sum_{l=1}^{g}\sum_{j=1}^{n_l} (X_{lj} - \overline{X}_l)^2$$

* One way ANOVA table (use to compare 2 means from two independent, unrelated groups using F distribution. $\begin{cases} H_0: \mu_1 = \mu_2 \\ H_1: \mu_1 \neq \mu_2 \end{cases}$)

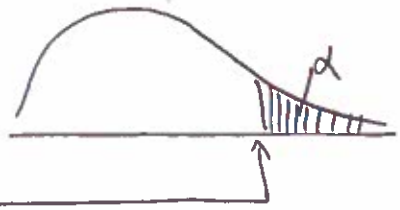
Source	SS (sum of square)	df (degree of freedom)	MS (mean square)
treatment	$SS_t = \sum_{l=1}^{g} n_l (\overline{X}_l - \overline{X})^2$	$g-1$	$SS_t/(g-1)$
residual	$SSR = \sum_{l=1}^{g}\sum_{j=1}^{n_l} (X_{lj} - \overline{X}_l)^2$	$\sum_{l=1}^{g} n_l - g$	$\frac{SSR}{\sum_{l=1}^{g} n_l - g}$
total	$\sum_{l=1}^{g}\sum_{j=1}^{n_l} (X_{lj} - \overline{X})^2$	$\sum_{l=1}^{g} n_l - 1$	

* When we want to test there is treatment effect or no treatmen effect.

$$\begin{cases} H_0 : T_\ell = 0 \\ H_1 : T_\ell = 1 \end{cases}$$

$$F\text{-test statistic} = \frac{\frac{SSt_r}{g-1}}{\frac{SSR}{\sum_{\ell=1}^{g} n_\ell - g}} = \frac{MSt_r}{MSR}$$

At level α, $F > F_{g-1, \sum_{\ell=1}^{g} n_\ell - g}(\alpha)$



6) MANOVA (Use for multiple comparison)

* The response $\underset{\sim}{X}_{lj} \sim N(\underset{\sim}{\mu} + \underset{\sim}{\tau}_l, \Sigma)$

$\underset{\sim}{X}_{lj} = \underset{\sim}{\mu} + \underset{\sim}{\tau}_l + \underset{\sim}{\varepsilon}_{lj}$; where $\varepsilon_{lj} \sim N_p(0, \Sigma)$, $\sum_{l=1}^{n_l} n_l \underset{\sim}{\tau}_l = \underset{\sim}{0}$

* One way MANOVA (for comparing population mean vectors)

Source	SS	df
Treatment	$B = \sum_{l=1}^{g} n_l (\bar{X}_l - \bar{X})(\bar{X}_l - \bar{X})^T$	$g-1$
Residure	$W = \sum_{l=1}^{g} \sum_{j=1}^{n_l} (X_{lj} - \bar{X}_l)(X_{lj} - \bar{X}_l)^T$	$\sum_{l=1}^{g} n_l - g$
Total	$B+W = \sum_{l=1}^{g} \sum_{j=1}^{n_l} (X_{lj} - \bar{X})(X_{lj} - \bar{X})^T$	$\sum_{l=1}^{g} n_l - 1$

* We consider the test

$H_0 : \underset{\sim}{\tau}_l = \underset{\sim}{0}$, $l = \overline{1,g}$

$\Lambda^* = \frac{|W|}{|B+W|}$ ← Wilk's lambda.

Reject H_0 when Λ^* is small

* Let $\lambda_1, \ldots, \lambda_p$ be the eigenvalues of BW^{-1}

then $\Lambda^* = \frac{|I|}{|BW^{-1} + I|} = \prod_{i=1}^{p} \frac{1}{\lambda_i + 1}$

* Assumption and limitation.

1) The response variables are continuous.
2) The residuals follow the $N_p(0, \Sigma)$
3) The individuals are independent
4) The variance-covariance matrices of each group of residuals are equal

* Box's M test (to test that the variance-covariance

$H_0: \Sigma_1 = \Sigma_2 = \cdots = \Sigma_g$

$\Lambda_{box} = \prod_{l=1}^{g} \left(\frac{|S_l|}{|S_{pooled}|} \right)^{\frac{n_l - 1}{2}}$, $S_{pooled} = \frac{1}{\sum_{l=1}^{g} (n_l - 1)} \left[(n_1 - 1)S_1 + (n_2 - 1)S_2 + \cdots + (n_g - 1)S_g \right]$

pooled covariance matrix.

Let $M = -2 \ln \Lambda = \sum_{l=1} (n_l - 1) \ln |S_{pooled}| - \sum_{l} (n_l - 1) \ln |S_i|$

* Extend to 2 way MANOVA $[\underset{\sim}{X_1}, \underset{\sim}{X_2}, \underset{\sim}{X_3}]$

rate	additive
low	low
high	high

rate × additive (LL, HH, LH, HL)

	Source
B matrix	① Treatment 1
	② Treatment 2
	③ Treatment 1×2
W matrix	Residual
b+w	Total

* Example 2-way ANOVA

treatments

rate	additive	rate × additive
low	low	l l
high	high	h h
		l l
		h h

[X_1, X_2, X_3] — variable

$H_0: \mu_{1,tr_1} = X_{1,tr_2}$

by function.

not significance.

* Multiple comparison in MANOVA

inference < CI for $\underset{\sim}{\mu}$ for $\underset{\sim}{\mu_1}, \underset{\sim}{\mu_2}, \dots, \underset{\sim}{\mu_g}$
hypotheses test

$\underset{\sim}{\mu_l} = \underset{\sim}{\mu_{l1}} - \underset{\sim}{\mu_{l2}} \Rightarrow \underset{\sim}{\mu_{kj}} = \underset{\sim}{\mu_{ij}} - \underset{\sim}{\mu_{ik}}$

$(t_{1-\alpha})$ (BB adjusted interval) ( ) W residual, diagonal of the residual matrix.

↑ approximate of normal Z

* Example MANOVA 4 variables — Species: 3 species

table(Irish $ Species) × matax

[first observe that 4 variable are ≠
→ then we want to compare them in pairs

variable 1 → Sped 1 & 2
1 & 3
2 & 3.

2 →

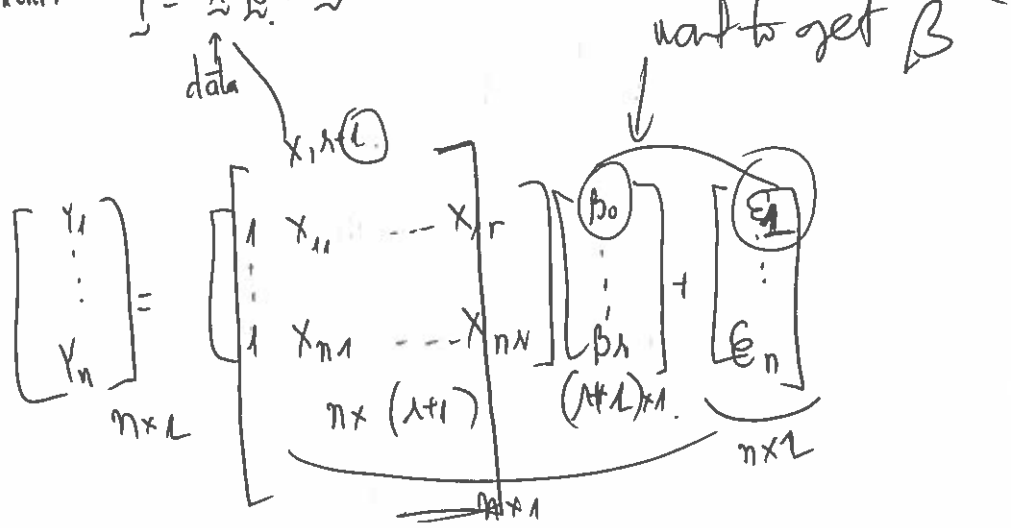
3 →

4 →

} 3×4 CI

C3 →

17 β: unknown $Y = X\beta + \varepsilon$



$= \beta^T$

$Y = \beta^T X^* + \varepsilon^*$ ← different way ← keep consistence then OK

$1 \times n \quad r \times (r+1) \quad (r+1) \times n \quad 1 \times n$

Theorem 1: (Least square)

* Least square ← try to minimize the square of the residual
  Maximum like

$\hat{\beta}_{LS} = \underset{\beta}{\operatorname{argmin}} (Y - X\beta)^T (Y - X\beta)$

$\hat{\beta}_{LS} = (X^T X)^{-1} X^T Y$

$[X]$

H: Hat matrix .

Y: Yobserved .

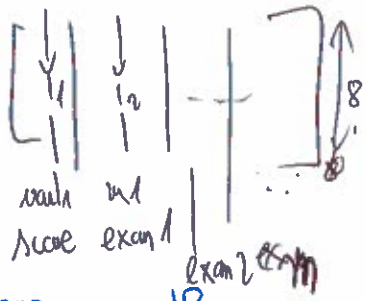
T2 : L

T3 β: true mean .

T4 : asymptotic → χ^2
normality t

2. Multivariate Regression

Y : matrix
$n \times m$



response variable

(n individuals / n observations.)

$$\begin{bmatrix} Y_{11} & \cdots & Y_{1m} \\ \vdots & & \vdots \\ Y_{n1} & \cdots & Y_{nm} \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1r} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & \cdots & X_{nr} \end{bmatrix} \underbrace{\begin{bmatrix} \beta_{01} & \cdots & \beta_{0m} \\ \vdots & & \vdots \\ \beta_r & & \beta_{rm} \end{bmatrix}}_{\beta_{(r)}} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$\beta_{(r)}$: $(r+1) \times m$ matrix

2 way
- least square m respond variable
  - book Kim's course
- Read the prove (ex: Max likelihood esting)

want to get $\hat{\beta}$.

Y matrix

$$\begin{bmatrix} | & | & & | \\ Y_1 & Y_2 & \cdots & Y_m \\ | & | & & | \end{bmatrix}$$

m response variable

$n \times m \quad n \times (\lambda+1)(\lambda+1) m \quad n \times m$

$$\underset{\sim}{Y} = \underset{\sim}{X}\,\underset{\sim}{\beta} + \underset{\sim}{\varepsilon}$$

λ: # of predictor

m: # of response

$$= \begin{bmatrix} \varepsilon_1' \\ \varepsilon_2^T \\ \vdots \\ \varepsilon_n^T \end{bmatrix} = \{\varepsilon_{(1)}, \dots, \varepsilon_{(m)}$$

$$\begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1m} \\ \vdots & Y_{22} & \cdots & Y_{2m} \\ Y_{n1} & Y_{n2} & & Y_{nm} \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X \\ 1 & & \\ \vdots & & \\ 1 & X_{n1} & X_{n\lambda} \end{bmatrix} \begin{bmatrix} \beta_{01} & \beta_{02} & \beta_{0m} \\ & & \\ \beta_{\lambda 1} & \beta_{\lambda 2} & \beta_{\lambda m} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} & \varepsilon_{1m} \\ & \\ \varepsilon_{n1} & \varepsilon_{nm} \end{bmatrix}$$

person ↑ response variable

$\varepsilon_{(j)} \sim N(\underset{\sim}{0}_n, \sigma_{jj}^2 I_n)$ $j = 1, \dots, m$, j^{th} column

$\varepsilon_{(i)} \overset{id}{\sim} N(0_m, \Sigma)$ i^{th} row

$$vec(\varepsilon) = \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{n1} \\ \hline \varepsilon_{12} \\ \vdots \\ \varepsilon_{n2} \\ \hline \vdots \\ \hline \varepsilon_{1m} \\ \vdots \\ \varepsilon_{nm} \end{bmatrix} \sim N(\underset{\sim}{0}_{nm}, \Sigma \otimes I_n)$$

? full model
reduce model
linear regression

* $\beta_2(0)$ → we want to test if { only the upper part has effect on the data
the lower part does not have effect on the data

use $\Lambda \longrightarrow \chi^2$ (↑ not friendly)

$H_0: \beta_2 = 0$
Reject H_0

$\lambda = 2$ $m = 2$.

lm command | resid $\leftrightarrow \varepsilon$

$\hat{Y} = x\hat{\beta}$ ↓

$\hat{\varepsilon} = Y - \hat{Y}$

* Want to test $\beta_2 = 0$

$\hat{\Sigma}_{MLE} = \frac{1}{n}(Y-\hat{Y})^T(Y-\hat{Y}) \frac{1}{n}\hat{\varepsilon}^T\hat{\varepsilon}$

sigma full

: all $\beta = 0$?

* Quicker way

directly compute residual without computing $\hat{y}_i$, $\hat{y}_i - \bar{y}$

$Y = \mu + \tau_\ell + \varepsilon_\ell$ (relation between linear regression & MANOVA)

MANOVA helps test $\begin{cases} \bar{Y}_{full} = X_2 \hat{\beta} \\ Y_{reduce} = \end{cases}$

when MANOVA still be valid $\Rightarrow$ X_2 is not important.

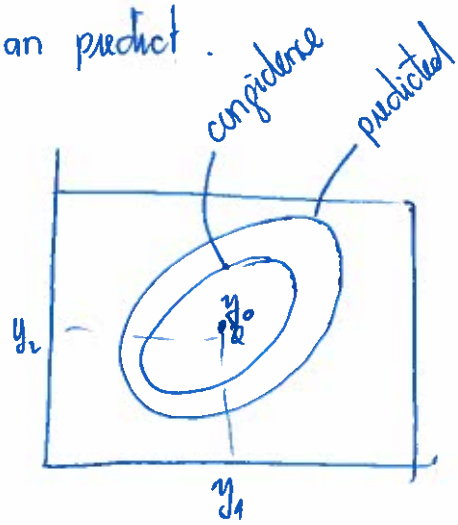
* $\begin{cases} \text{Confidence interval} \\ \text{Predicted} \quad " \end{cases}$ $\leftarrow$ new data comes in $\Rightarrow$ we can predict.

$\hat{Y}_0 = \hat{X}_0 \hat{\beta}$

↑ ↑ the $\hat{\beta}$ that we got from previous data

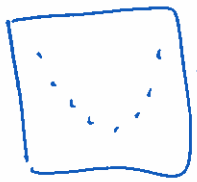
↑ new data

predicted data → predicted interval



- pair data $X_0 = \begin{bmatrix} x_{011} & x_{012} \\ x_{021} & x_{022} \end{bmatrix}$ new

- Residual plot



→ need to check.

box cox, transformation.

* $\begin{cases} \text{fail to reject the last 3} \end{cases}$ type III F-test

With MANOVA
we assume that they are
real factors
affect the variable.

solve(S) = S^{-1}
t(S) = S^T

Chapter 2. Inference for the mean

Jianxuan Liu

Fall 2018

1. Wishart Distribution

Let $X_1, \cdots, X_n \sim^{iid} N_p(\mathbf{0}, \Sigma)$, then

$$W_n = \sum_{i=1}^{n} X_i X_i^T \sim Wishart(\Sigma, n).$$

Let $X_1 \sim W_p(\Sigma, m_1)$ and $X_2 \sim W_p(\Sigma, m_2)$ be independent. Let C be a $q \times p$ constant matrix and a be a p constant vector s.t. $a^T \Sigma a \neq 0$. then

- $X_1 + X_2 \sim W_p(\Sigma, m_1 + m_2)$
- $CX_1C^T \sim W_p(C\Sigma C^T, m_1)$
- $\frac{a^T X a}{a^T \Sigma a} \sim \chi^2_{m_1}$

Let $X_1, \cdots, X_n \sim^{iid} N_p(\mu, \Sigma)$. then

- $\bar{X} \sim N_p(\mu, \Sigma/n)$.
- $(n-1)S \sim W_p(\Sigma, n-1)$.

Since $\bar{X} \sim N_p(\mu, \Sigma/n)$, then $\sqrt{n}(\bar{X} - \mu) \sim N_p(0, \Sigma)$ and $\sqrt{n}\Sigma^{-1/2}(\bar{X} - \mu) \sim N_p(\mathbf{0}, \mathbf{I})$. We also have $n(\bar{X} - \mu)^T \Sigma^{-1}(\bar{X} - \mu) \sim \chi^2_p$. $n(\bar{X} - \mu)^T S^{-1}(\bar{X} - \mu)$ does not follow χ^2_p.

2. Hotelling's T^2

Suppose that (in univariate case) $x_1, \cdots, x_n \sim^{iid} N(\mu, \sigma^2)$. We use $t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$ as a test statistic. The confidence interval for μ is

$$\bar{x} \pm \frac{s}{\sqrt{n}} t_{1-\alpha/2, n-1}$$

Let $\mathbf{x} \sim N_p(\mathbf{0}, \Sigma)$ and $W \sim W_p(\Sigma, n-1)$ be independent, then

$$T^2 = \mathbf{x}^T (\frac{1}{n-1} W)^{-1} \mathbf{x} \sim T^2_{p, n-1}$$

Consideration of the squared Mahalabnobis distance leads us to consider the so called T^2 statistic (the nomenclature reflects that this relates to a t-statistic for one variable). This can be found as:

$$T^2 = n(\bar{x} - \mu_0)^T \mathbf{S}^{-1} (\bar{x} - \mu_0)$$

where n is the sample size, μ_0 is the hypothesized mean, $\bar{x}$ and $\mathbf{S}$ are the sample mean and covariance matrices respectively. It turns out that this statistic follows a T^2 distribution, however, given that there is a simple relationship between the T^2 and F distribution it is often easier to work with the latter. If $x_i, i = 1, \cdots n$ represent a sample from a p variate normal distribution with mean μ_0 and covariance Σ, provided Σ is positive definite and $n > p$, given sample estimators for mean and covariance $\bar{x}$ and $\mathbf{S}$ respectively, then:

$$F = \frac{n}{n-1} \frac{n-p}{p} (\bar{x} - \mu_0)^T \mathbf{S}^{-1} (\bar{x} - \mu_0)$$

follows an F-distribution with p and $(n-p)$ degrees of freedom. Note the requirement that $n > p$, i.e. that $\mathbf{S}$ is non-singular.

3. Two sample Hotelling's T^2 test

Analagous to the univariate context, we wish to determine whether the mean vectors are comparable, more formally:

$$H_0 : \mu_1 = \mu_2$$

The T^2 statistic proposed by Hotelling (1931), will be based this time on the distance between two mean vectors. It can be calculated as:

$$T^2 = \frac{n_1 n_2}{n_1 + n_2}(\bar{x}_1 - \bar{x}_2)^T \mathbf{S}^{-1}(\bar{x}_1 - \bar{x}_2)$$

where $\mathbf{S}^{-1}$ is the inverse of the pooled correlation matrix given by:

$$\mathbf{S}_{pool} = \frac{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2}{n_1 + n_2 - 2}$$

given the sample estimates for covariance, $\mathbf{S}_1$ and $\mathbf{S}_2$ in the two samples. As before, there is a simple relationship between the test statistic, T^2, and the F distribution.

If $x_{1i}, i = 1, \cdots n_1$ and $x_{2i}, i = 1, \cdots n_2$ represent independent samples from two p variate normal distribution with mean vectors μ_1 and μ_2 but with common covariance matrix Σ, provided Σ is positive definite and $n > p$, given sample estimators for mean and covariance $\bar{x}$ and $\mathbf{S}$ respectively, then:

$$F_{observed} \leftarrow F = \frac{(n_1 + n_2 - p - 1)T^2}{(n_1 + n_2 - 2)p} \sim F_{p,\, n_1+n_2-p-1}.$$

has an F distribution on p and $(n_1 + n_2 - p - 1)$. Essentially, we compute the test statistic, and see whether it falls within the $(1-\alpha)$ quantile of the F distribution on those degrees of freedom. Note again that to ensure non-singularity of $\mathbf{S}$, we require that $n_1 + n_2 > p$.

We are going to consider an example using data from Flea Beetles reported by Lubischew (1962) and used in (Flury, 1997, page 307). It should be noted that in terms of practical computation, methods are based on the QR decomposition will be used, details are given in Seber (1984). However, for the purposes of understanding the principles behind the test, we follow the formula directly.

```
library(Flury)
?flea.beetles
data(flea.beetles)
```

Species TG Elytra Second.Antenna Third Antenna
1 oleracea
2 carduorum.

It can be seen that there is a factor "Species" denoting whether the beetles are from 'oleracea' or 'carduorum'. There are four numeric variables as follows:

1. 'TG': Distange of the Transverse Groove to the posterior border of the prothorax (microns)
2. 'Elytra': Length of the Elytra (in units of 0.01mm)
3. 'Second.Antenna': Length of the second antennal joint (microns)
4. 'Third.Antenna': Length of the third antennal joint (microns).

We need to estimate the mean for each sample, and calculate the difference between the two vectors:

```
mu <- by(flea.beetles[,-1], flea.beetles$Species, colMeans)
mudiff <- mu[[1]] - mu[[2]]
p <- dim(flea.beetles)[2] - 1 ## how many variables are we using
```

The next step is to extract the two covariance matrices:

Variables: TG, Elytra
Second. Antenna
Third. Antenna.

covmats = [S1, S2] covmats[[1]] = S1 covmats[[2]] = S2.

```
covmats <- by(flea.beetles[,-1], flea.beetles$Species, cov)
covmats
```

```
## flea.beetles$Species: oleracea
##                      TG    Elytra Second.Antenna Third.Antenna
## TG             187.59649 176.86257       48.37135     113.58187
## Elytra         176.86257 345.38596       75.97953     118.78070
## Second.Antenna  48.37135  75.97953       66.35673      16.24269
## Third.Antenna  113.58187 118.78070       16.24269     239.94152
## ------------------------------------------------------------
## flea.beetles$Species: carduorum
##                      TG    Elytra Second.Antenna Third.Antenna
## TG             101.83947 128.06316       36.98947      32.59211
## Elytra         128.06316 389.01053      165.35789      94.36842
## Second.Antenna  36.98947 165.35789      167.53684      66.52632
## Third.Antenna   32.59211  94.36842       66.52632     177.88158
```

Spooled

and then to estimate the pooled covariance matrix S for the flea beetle data (where N[1] gives n1, N[2] gives n2), can be calculated as:

```
N <- xtabs(~flea.beetles[,1])  # count of the level of categorical variables
pooledS <- ((N[1]-1) * covmats[[1]] + (N[2]-1) * covmats[[2]]) / (N[1] + N[2] -2)
pooledS
```

```
##                      TG    Elytra Second.Antenna Third.Antenna
## TG             143.55910 151.8034       42.52660      71.99253
## Elytra         151.80341 367.7878      121.87653     106.24467
## Second.Antenna  42.52660 121.8765      118.31408      42.06401
## Third.Antenna   71.99253 106.2447       42.06401     208.07290
```

```
Sinv <- solve(pooledS)  # calculate the inverse of Spooled
Sinv
```

```
##                      TG        Elytra Second.Antenna Third.Antenna
## TG             0.013257964 -0.0053492256   0.0015134494 -0.0021617878
## Elytra        -0.005349226  0.0066679441  -0.0047337699 -0.0005969439
## Second.Antenna 0.001513449 -0.0047337699   0.0130490933 -0.0007445297
## Third.Antenna -0.002161788 -0.0005969439  -0.0007445297  0.0060093005
```

Having calculated the inverse of the pooled correlation matrix we also need the scaling factor $\frac{n_1 n_2}{n_1 + n_2}$. Hotellings T^2 is then quite straightforward to calculate:

```
scaleFact <- (N[1]*N[2]) / (N[1]+N[2])
Hotellings <- t(mudiff) %*% Sinv %*% mudiff * scaleFact   # T^2
Hotellings
```

```
##          [,1]
## [1,] 133.4873
```

which is the value of the T^2 statistic. We could work with this value directly, but it is more convenient to transform it into something we can compare with the F distribution.

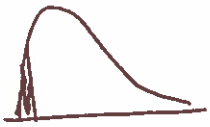
```
test <- ((N[1] + N[2] - p - 1) * Hotellings )/ ((N[1] + N[2] - 2) * p)
test
```

test = $\frac{(n_1+n_2-p-1)T^2}{(n_1+n_2-2)p}$

```
##        [,1]
## [1,] 30.666
```

Reject H_0 when $F_{observed} = \frac{(n_1+n_2-p-1)}{(n_1+n_2-2)p} T^2 > F_{p, n_1+n_2-p-1}$.

When we want to test $\begin{cases} H_0: \mu_1 = \mu_2 \\ H_1: \mu_1 \neq \mu_2 \end{cases}$



Stat505 note
We would reject Ho at level α if
$F_{observe} > F_{p, n_1+n_2-p-1, \alpha}$

*Equal covariance case:
Reject when
$T^2_{observed} > T^2_{1-\alpha, p, n_1+n_2-2}$

and we compare th ... an check this as follows: degree

```
pf(test, p, N[1]+N[2]-p-1,lower.tail = FALSE )
```

pf(q, df1, df2, lower.tail = T ⇒ P(X ≤ q), lower.tail = F ⇒ P(X > q))

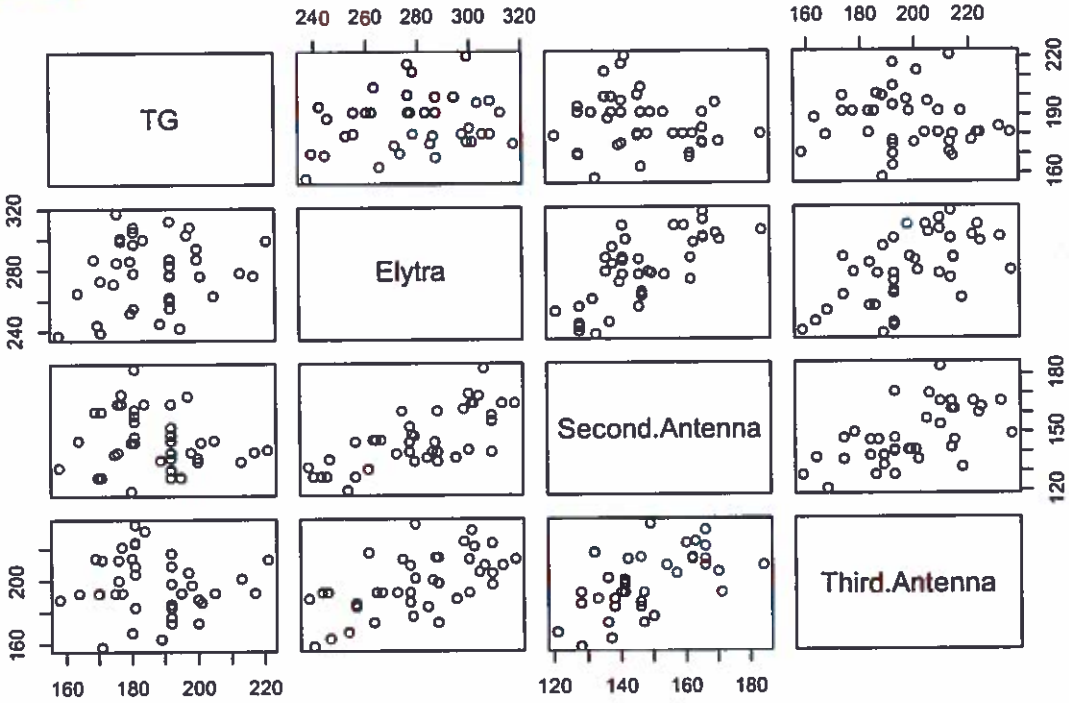
```
##              [,1]
## [1,] 7.521799e-11
```

⇒ P(F > F_test) ⇒ really small ⇒ F_test < F

which gives us the area under the curve from our test statistic (30.666) to 0. Clearly in this case, we have reject H_0, i.e. there is evidence that the mean vectors, $\bar{x}_{oleracea} = (194.4737, 267.0526, 137.3684, 185.9474)$, $\bar{x}_{carduorum} = (179.55, 290.80, 157.20, 209.25)$, for the two species differ.

⇒ $\bar{x}_1 \neq \bar{x}_2$

```
pairs(flea.beetles[,-1],col=flea.beetles[,1])
```



4. Constant Density Ellipses

Flury (1997) gives an interpretation of constant density ellipses in terms of the Mahalanobis distance which is worth reading. Essentially, we wish to find a region of squared Mahlanobis distance such that:

$$Pr((\bar{\mathbf{x}} - \mu)^T \mathbf{S}^{-1} (\bar{\mathbf{x}} - \mu) \leq c^2) = 1 - \alpha$$

and we can find c^2 as follows:

$$c^2 = \frac{n-1}{n} \frac{p}{n-p} F_{(1-\alpha),p,(n-p)}$$

where $F_{(1-\alpha),p,(n-p)}$ is the $(1-\alpha)$ quantile of the F distribution with p and $n-p$ represents the number of variables and n the sample size. Illustration: Firstly, we need a function to draw ellipses:

Ch2_Constant Density Ellipse

```
Ellipse <- function(covmat, centroid, csquare, resolution, plot = TRUE) {
angles <- seq(0, by = (2 * pi)/resolution, length = resolution)
sd <- covmat[1,2] / sqrt(covmat[1,1] * covmat[2,2])
projmat <- matrix(0,2,2)
```

length

```
projmat[1,1] <- sqrt(covmat[1,1] %*% (1+sd)/2)
projmat[1,2] <- -sqrt(covmat[1,1] %*% (1-sd)/2)
projmat[2,1] <- sqrt(covmat[2,2] %*% (1+sd)/2)
projmat[2,2] <- sqrt(covmat[2,2] %*% (1-sd)/2)
circle <- cbind(cos(angles), sin(angles))
Ellipse <- t(centroid + sqrt(csquare) * projmat %*% t(circle))
if (plot == TRUE) {lines(Ellipse)}
return(Ellipse)
}
```

compute c^2

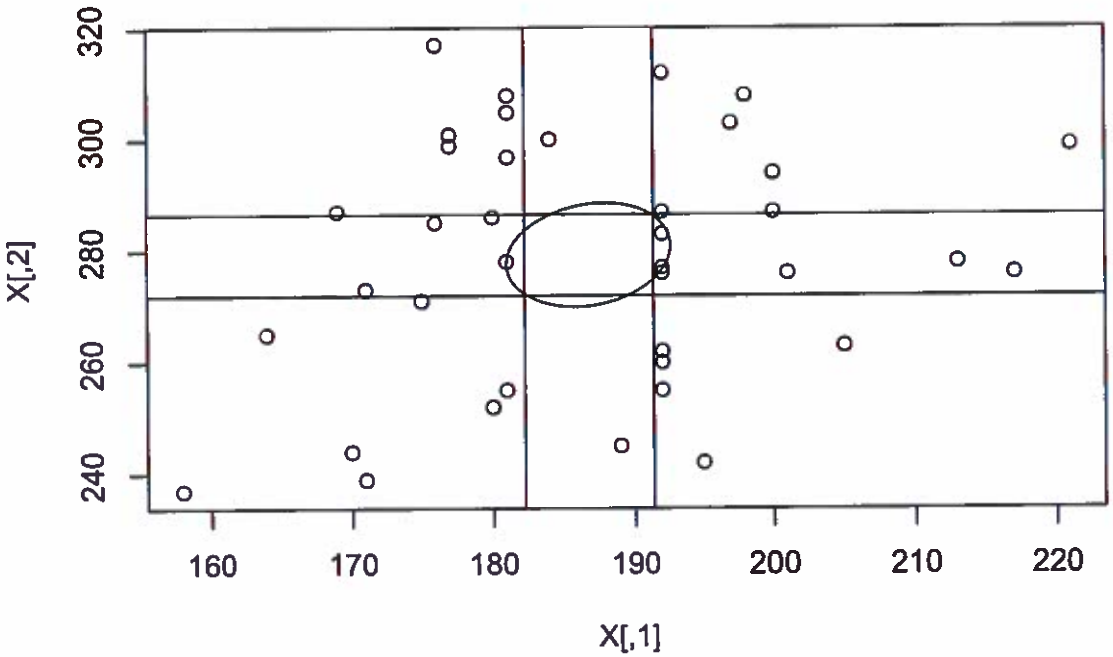
It is possible to define a function which calculates c^2 and calls the Ellipse routine

```
cdellipse <- function (data, alpha=0.05, resolution=500)
{
xbar <- colMeans(data)
n <- dim(data)[1]
p <- dim(data)[2]
f <- qf(1-alpha, p, n-p)
csquare <- ((n-1)/n) * (p / (n-p)) * f
cat(csquare)
Ellipse <- Ellipse(cov(data), xbar, csquare, resolution)
}
```

For illustrative purposes, we'll create a $n \times 2$ data object from our flea beetles, and plot the confidence ellipse for these. $\alpha = 0.05$

```
X <- cbind(flea.beetles[,2], flea.beetles[,3])
plot(X)
cdellipse(X, alpha = 0.05)  # the cdellipse helps draw an ellipse covers (2) variables

## 0.1712725

#These can be contrasted with the univariate confidence intervals:
abline(v = confint(lm(X[,1]~1)))
abline(h = confint(lm(X[,2]~1)))
```

} # univariate comp for the two above variable



It can be seen that the univariate confidence intervals and the constant density ellipse support different areas of parameter space. Ignoring the correlation structure in these data could lead to flaws in inference when assessesing parameter uncertainty.

Results and comparison

Let $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \cdots, \bar{\mathbf{x}}_n$ be a random sample from an $N_p(\mu, \Sigma)$ population with Σ positive definite.

- Simultaneouly for all $\mathbf{a}$, the intervals

$$\left(\mathbf{a}^T\bar{\mathbf{x}} \pm \sqrt{\frac{p(n-1)}{n(n-p)}F_{1-\alpha,p,n-p}\mathbf{a}^T\mathbf{S}\mathbf{a}}\right)$$

  are exact at $1-\alpha$ for all $\mathbf{a}^T\mu$.

- Let $(\mathbf{a}_i = (0, \cdots, 0, 1, 0, \cdots, 0)$, then $\mathbf{a}_i^T\mu = \mu_i$ and

$$\left(\mathbf{a}_i^T\bar{\mathbf{x}} \pm \sqrt{\frac{p(n-1)}{n(n-p)}F_{1-\alpha,p,n-p}\mathbf{a}_i^T\mathbf{S}\mathbf{a}_i}\right)$$

  that is

$$\left(\bar{\mathbf{x}}_i \pm \sqrt{\frac{p(n-1)}{n(n-p)}F_{1-\alpha,p,n-p}\mathbf{S}_i}\right)$$

  is simultaneouly exact for all μ_i at $1-\alpha$ level.

- One-at-a-Time intervals ignore the covariance structure of the p variables. This approach construct confidence intervals for μ_j one at a time. For $j = 1, \cdots, p$, the One-at-a-Time intervals are

$$\left(\bar{x}_j \pm t_{n-1,\alpha/2}\sqrt{s_{jj}/n}\right)$$

- When the number m of specified component mean μ_i, or linear combinations $\mathbf{a}^T\mu$ is small, then a better approach is to calculate the Bonferroni corrected confidence intervals as given in the expression below:

$$\left(\bar{x}_j \pm t_{n-1,\alpha/2p}\sqrt{s_{jj}/n}\right)$$

- Asymptotic simultaneous intervals use the χ^2 approximation for large sample. If $n-p$ is large, then

$$\left(\mathbf{a}^T\bar{\mathbf{x}} \pm \sqrt{\chi^2_p(\alpha)}\sqrt{\frac{\mathbf{a}^T\mathbf{S}\mathbf{a}}{n}}\right)$$

contain $\mathbf{a}^T\mu$ for every $\mathbf{a}$, with probability approximately $1-\alpha$.

Lizard data example:

CP2_Lizard

```
library(ICSNP) # For one and two sample Hotelling's T2 test (HotellingsT2)
```

```
## Warning: package 'ICSNP' was built under R version 3.4.4
```

```
## Loading required package: mvtnorm
```

```
## Warning: package 'mvtnorm' was built under R version 3.4.4
```

```
## Loading required package: ICS
```

```
## Warning: package 'ICS' was built under R version 3.4.3
```

```
library(ellipse) # For ellipse CI
```

```
## Warning: package 'ellipse' was built under R version 3.4.3
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
# ------------------------------------------------------------------
# Import lizard data set
# ------------------------------------------------------------------

lizard <- read.table("Wichern_data/T1-3.dat")
names(lizard) <- c("Mass", "SVL", "HLS")

n <- dim(lizard)[1] # number of observations
p <- dim(lizard)[2] # number of variables
n
```

```
## [1] 25
```

```
p
```

```
## [1] 3
```

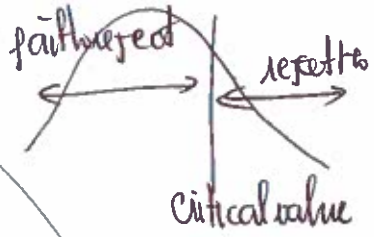
```
# ------------------------------------------------------------------
#  Do the one sample hotelling T2-test manually
# ------------------------------------------------------------------
```

One sample Hotelling T2 test

$X = \begin{bmatrix} \text{mass} \\ \text{SVL} \\ \text{HLS} \end{bmatrix}$ we want to testing $\mu_X = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} \overset{?}{=} \mu_0 = \begin{bmatrix} 0 \\ 68 \\ 129 \end{bmatrix}$

One sample T^2 test manually

```
#-----------
# Do the one sample Hotelling T^2-test manually #
#-----------
alpha <- 0.05

liz.xbar <- sapply(lizard, mean)   x̄
liz.cov <- cov(lizard)             S

liz.xbar  // x̄

##     Mass      SVL      HLS
##   8.6866  68.4000 129.3200
liz.cov  S

##           Mass       SVL       HLS
## Mass  7.186551  20.69952  33.53684
## SVL  20.699521  63.77083 102.08542
## HLS  33.536842 102.08542 185.83083
solve(liz.cov) # S^{-1}

##             Mass          SVL          HLS
## Mass  2.17163898 -0.64273969 -0.03882867
## SVL  -0.64273969  0.32026006 -0.05993851
## HLS  -0.03882867 -0.05993851  0.04531561
u0 <- c(9, 68, 129)

T2 <- n * mahalanobis(x = liz.xbar, center = u0, cov = liz.cov)   T^2 = n (x̄ - μ0)^T S^{-1} (x̄ - μ0)

T2

## [1] 10.56927
crit.val <- p * (n-1) / (n-p) * qf(1-alpha, p, n-p)   (n-1)p/(n-p) F_{p,n-p,1-α} ?

crit.val # T2 > crit.val  => reject Ho.

## [1] 9.978955
# ------------------------------------------------------------
#  Do the one sample hotelling T2-test using HotellingsT2
# ------------------------------------------------------------

liz.T2 <- HotellingsT2(X = lizard, mu = u0)   liz.T2 = (n-p)/(p(n-1)) T2 = F ~ F_{p,n-p}

liz.T2

##
##  Hotelling's one sample T2-test
##
## data:  lizard
## T.2 = 3.2295, df1 = 3, df2 = 22, p-value = 0.04202
## alternative hypothesis: true location is not equal to c(9,68,129)
# Note that this is the rescale version of the above T2 statistic.
# So that liz.T2 = (n-p) / p / (n-1) * T2,
# where T2 is the above one covered in class

# ------------------------------------------------------------
#
```



in class

pvalue < alpha → reject Ho.

One sample T^2 test using R command

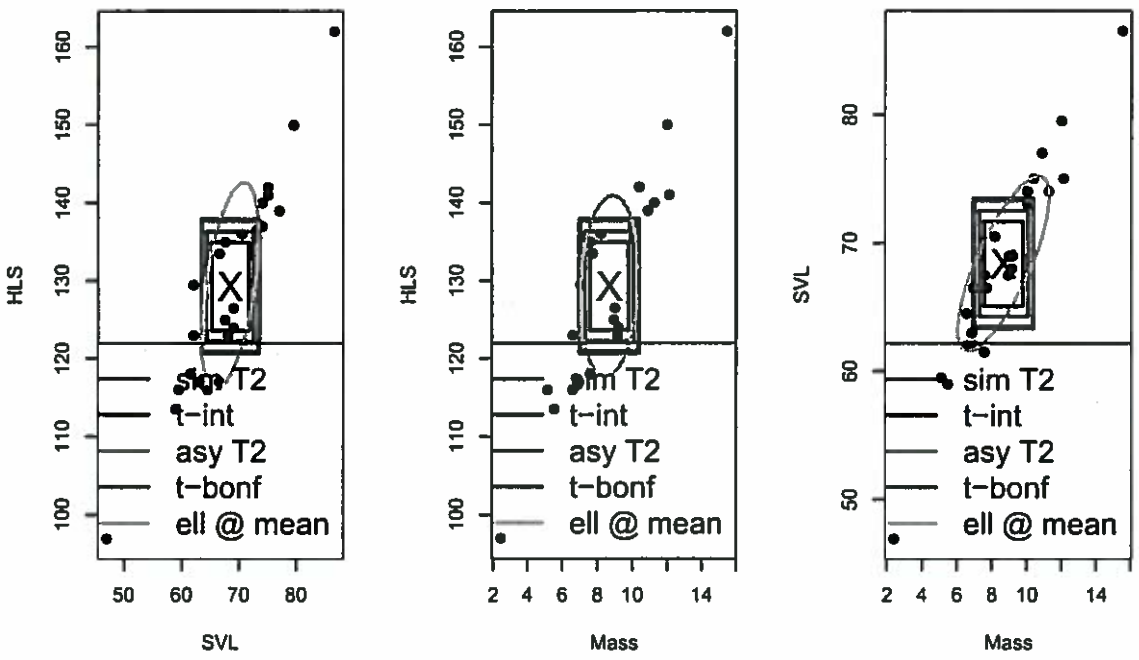
$a^T \bar{X} \mp \sqrt{\frac{p(n-1)}{n(n-p)} F_{1-\alpha, p, n-p} \frac{a^T S a}{\text{diag of } S}}$ where $a_i' = (0, .., \underset{\uparrow i}{1}, .., 0)$

```
# Simultaneous T2 intervals (Using the Hotelling T2 intervals)
#
# ------------------------------------------------------------------

ci.T2 <- rbind(liz.xbar - sqrt(p*(n-1)/(n-p)/n*qf(1-alpha, p, n-p)*diag(liz.cov)),
               liz.xbar + sqrt(p*(n-1)/(n-p)/n*qf(1-alpha, p, n-p)*diag(liz.cov)))
row.names(ci.T2)<-c("T2_L", "T2_U")

# ------------------------------------------------------------------
#
# One at a time t-intervals (the usual t-interval at level 1 - alpha
# with no correction for p = 3 variables). See page 229 - 231
#
# ------------------------------------------------------------------

ci.t <- rbind(liz.xbar - sqrt(diag(liz.cov)/n)*qt(1-alpha/2, n-1),
              liz.xbar + sqrt(diag(liz.cov)/n)*qt(1-alpha/2, n-1))
row.names(ci.t)<-c("t_L", "t_U")

# ------------------------------------------------------------------
#
# Asmptotic Simultaneous intervals (Using the chisq approximation -
# i.e., the Hotelling T2 for large n) See page 224 - 235
#
# ------------------------------------------------------------------

ci.Asy <- rbind(liz.xbar - sqrt(qchisq(1-alpha/2,p)/n*diag(liz.cov)),
                liz.xbar + sqrt(qchisq(1-alpha/2,p)/n*diag(liz.cov)))
row.names(ci.Asy)<-c("Asy_L", "Asy_U")

# ------------------------------------------------------------------
#
# Bonferonni corrected (the usual t-interval with bonferroni correction
#  i.e., level 1 - alpha / p) See page 232 - 234
#
# ------------------------------------------------------------------

ci.Bon <- rbind(liz.xbar - sqrt(diag(liz.cov)/n)*qt(1-alpha/2/p, n-1),
                liz.xbar + sqrt(diag(liz.cov)/n)*qt(1-alpha/2/p, n-1))
row.names(ci.Bon)<-c("Bon_L", "Bon_U")

# ------------------------------------------------------------------
#
# All together (_L indicates the lower endpoint and _U the upper endpoint)
#
# ------------------------------------------------------------------

all.ci <- round(rbind(ci.T2, ci.t, ci.Asy, ci.Bon),digits=2)
all.ci
```

```
##         Mass   SVL    HLS
## T2_L    6.99 63.35 120.71
## T2_U   10.38 73.45 137.93
## t_L     7.58 65.10 123.69
```

```
## t_U    9.79 71.70 134.95
## Asy_L  7.05 63.52 120.98
## Asy_U 10.33 73.28 137.66
## Bon_L  7.31 64.29 122.30
## Bon_U 10.07 72.51 136.34
# ------------------------------------------------------------------
#
# Plot over scatter plots
#
# ------------------------------------------------------------------

old.par <- par(no.readonly = TRUE)

#dev.new(width = 15)
  par(mfrow = c(1, p), oma = c(0, 0, 2, 0))
  for(i in 1:p){
  plot(lizard[,-i], pch = 19)
    text(liz.xbar[-i][1], liz.xbar[-i][2], labels = "X", col = "black", cex = 2)
    lines(all.ci[,-i][c(1, 1, 2, 2, 1), 1], all.ci[,-i][c(1, 2, 2, 1, 1), 2],
          col = "red", lwd = 2)
    lines(all.ci[,-i][2+c(1, 1, 2, 2, 1), 1], all.ci[,-i][2+c(1, 2, 2, 1, 1), 2],
          col = "blue", lwd = 2)
    lines(all.ci[,-i][4+c(1, 1, 2, 2, 1), 1], all.ci[,-i][4+c(1, 2, 2, 1, 1), 2],
          col = "green", lwd = 2)
    lines(all.ci[,-i][6+c(1, 1, 2, 2, 1), 1], all.ci[,-i][6+c(1, 2, 2, 1, 1), 2],
          col = "purple", lwd = 2)
    lines(ellipse(x = solve(solve(liz.cov)[-i,-i]), centre = liz.xbar[-i],
                  level = 1-alpha, npoints = 1000), lwd = 2, col = "orange")
    legend("bottomright", legend = c("sim T2", "t-int", "asy T2", "t-bonf", "ell @ mean"),
           col = c("red", "blue", "green", "purple", "orange"), lty = 1, lwd = 2, cex = 1.5)
  }
  title("2-dimensional view of CI's for means", outer = TRUE)
```

2-dimensional view of CI's for means



#par(old.par)

ell & mean is the ellipsoid evaluated at the mean of the third omitted variable. These ellipses may look small and misleading because they are not drawn in the direction of the principal components, but it gives an idea of the relative size and shape.

5. Review of ANOVA

One factor ⇒ one way ANOVA ‖ variable is a random vector
two factors ⇒ two-way ANOVA ‖ (not vector variables).

Analysis of variance (ANOVA) is a collection of statistical methodologies for comparing several means. When there is only one way (one factor) to classify the populations of interest, we use one-way ANOVA to analyze the data. When analyzing the effect of two factors, we use two-way ANOVA.

Suppose that $X_{l1}, X_{l2}, \cdots, X_{ln_l}$ is a random sample from an $N(\mu_l, \sigma^2)$ population, $l = 1, 2, \cdots, g$, and that the random samples are independent. We write

$$X_l = \mu + \tau_l$$

where

- X_l is the l^{th} group mean
- μ is the overall mean
- τ_l is the l^{th} group effect or treatment effect

The response $X_{lj} \sim N(\mu + \tau_l, \sigma^2)$ can be expressed as

$$X_{lj} = \mu + \tau_l + \epsilon_{lj}$$

where ϵ_{lj} are independent $N(0, \sigma^2)$ random variables and $\sum_{l=1}^{g} n_l \tau_l = 0$.

We decompose X_{lj} as

$$X_{lj} = \bar{X} + (\bar{X}_l - \bar{X}) + (X_{lj} - \bar{X}_l)$$

where

- X_{lj}: observation
- $\bar{X}$: overall sample mean, an estimate of μ
- $(\bar{X}_l - \bar{X})$: estimated treatment effect, an estimate of τ_l
- $(X_{lj} - \bar{X}_l)$: residual, an estimate of the error ϵ_{lj}

Recall decomposing SST = SSTr+ SSR

$$\underbrace{\sum_{l=1}^{g}\sum_{j=1}^{n_l}(X_{lj} - \bar{X})^2}_{\text{SST}} = \underbrace{\sum_{l=1}^{g} n_l(\bar{X}_l - \bar{X})^2}_{\text{SSTr, within sample.}} + \underbrace{\sum_{l=1}^{g}\sum_{j=1}^{n_l}(X_{lj} - \bar{X}_l)^2}_{\text{SSR (within samples)}}$$

One-Way ANOVA table

Table 1: ANOVA Table for Comparing Univariate Population Means

Sourse	Sum of squares (SS)	df	MS	F
Treatments	SStr= $\sum_{l=1}^{g} n_l(X_l - X)^2$	$g-1$	SStr/$(g-1)$	MStr/MSR
Residuals	SSR = $\sum_{l=1}^{g}\sum_{j=1}^{n_l}(X_{lj} - \bar{X}_l)^2$	$\sum_{l=1}^{g} n_l - g$	SSR/$\sum_{l=1}^{g} n_l - 1$	
Total	SST = $\sum_{l=1}^{g}\sum_{j=1}^{n_l}(X_{lj} - \bar{X})^2$	$\sum_{l=1}^{g} n_l - 1$		

The usual F-test rejects $H_0 : \tau_l = 0, l = 1, \cdots, g$ at level α if

$$F = \frac{SStr/(g-1)}{SSR/(\sum_{l=1}^{g} n_l - g)} > F_{g-1,\sum_{l=1}^{g} n_l - g}(\alpha)$$

where $F_{g-1,\sum_{l=1}^{g} n_l - g}(\alpha)$ is the upper $\alpha \times 100\%$ of the F-distribution with df $g-1$ and $\sum_{l=1}^{g} n_l - g$.

6. Multivariate Analysis of variance (MANOVA)

As with the univariate situation, t-tests are fine for comparing the means of two groups, but we would have "multiple comparison" problems if we tried to compare more than two. In an analagous way, in the multivariate context we have MANOVA.

The response $\mathbf{X}_{lj} \sim N_p(\mu + \tau_l, \Sigma)$ can be expressed as

$$\mathbf{X}_{lj} = \mu + \tau_l + \epsilon_{lj}$$

where ϵ_{lj} are independent $N_p(\mathbf{0}, \Sigma)$ random variables and $\sum_{l=1}^{g} n_l\tau_l = \mathbf{0}$.

MANOVA table

Table 2: MANOVA Table for Comparing Population Mean Vectors

Sourse	Sum of squares (SS)	df
Treatments	$\mathbf{B} = \sum_{l=1}^{g} n_l(\bar{\mathbf{X}}_l - \bar{\mathbf{X}})(\bar{\mathbf{X}}_l - \bar{\mathbf{X}})^T$	$g-1$
Residuals	$\mathbf{W} = \sum_{l=1}^{g}\sum_{j=1}^{n_l}(\mathbf{X}_{lj} - \bar{\mathbf{X}}_l)(\mathbf{X}_{lj} - \bar{\mathbf{X}}_l)^T$	$\sum_{l=1}^{g} n_l - g$
Total	$\mathbf{B}+\mathbf{W} = \sum_{l=1}^{g}\sum_{j=1}^{n_l}(\mathbf{X}_{lj} - \bar{\mathbf{X}})(\mathbf{X}_{lj} - \bar{\mathbf{X}})^T$	$\sum_{l=1}^{g} n_l - 1$

matrix

$W = \Sigma^T\Sigma$ where $S \approx$

test > F ⇒

Test of $H_0 : \tau_l = 0, l = 1, \cdots, g$ involves generalized varian
variances

$$\Lambda^* = \frac{|\mathbf{W}|}{|\mathbf{B}+\mathbf{W}}$$

is too small. Λ^* is referred ad Wilk's lambda. Let $\lambda_1, \cdots, \lambda_l$

$$\Lambda^* = \frac{|\mathbf{I}|}{|\mathbf{B}\mathbf{W}^{-1}+\mathbf{I}|} =$$

Assumptions and Limitations

The following assumptions are made when using a MANOVA.

1. The response variables are continuous.
2. The residuals follow the multivariate-normal probability distribution with means equal to zero.
3. The individuals are independent.
4. The variance-covariance matrices of each group of residuals are equal. MANOVA makes the assumption that the within-cell (group) covariance matrices are equal. If the design is balanced so that there is an equal number of observations in each cell, the robustness of the MANOVA tests is guaranteed. If the design is unbalanced, you should test the equality of covariance matrices using Box's M test.

$$H_0 : \Sigma_1 = \Sigma_2 = \cdots = \Sigma_g$$

Let

$$\Lambda_{Box} = \prod_{l=1}^{g} \left(\frac{|\mathbf{S}_l|}{|\mathbf{S}_{pooled}|} \right)^{(n_l-1)/2}$$

where

$$\mathbf{S}_{pooled} = \frac{1}{\sum_l (n_l - 1)} \{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2 + \cdots + (n_g - 1)\mathbf{S}_g\}.$$

Let

$$M = -2 \ln \Lambda = \left[\sum_l (n_l - 1) \right] \ln |\mathbf{S}_{pooled}| - \sum_l [(n_l - 1) \ln |\mathbf{S}_l|]$$

Set

$$u = \left[\sum_l \frac{1}{n_l - 1} - \frac{1}{\sum_l (n_l - 1)} \right] \left[\frac{2p^2 + 3p - 1}{6(p+1)(g-1)} \right]$$

where p is the number of variables and g is the number of groups. Then

$$C = (1-u)M = (1-u) \left\{ \left[\sum_l (n_l - 1) \right] \ln |\mathbf{S}_{pooled}| - \sum_l [(n_l - 1) \ln |\mathbf{S}_l|] \right\}$$

has an approximate χ^2 distribution with degreess of freedom

$$\nu = \frac{1}{2}gp(p+1) - \frac{1}{2}p(p+1) = \frac{1}{2}p(p+1)(g-1).$$

We reject H_0 if $C > \chi^2_{p(p+1)(g-1)/2}(\alpha)$. Box's χ^2 approximation workd well if $n_l \leq 20$ and $p, g \leq 5$. Box's M test is sensitive to some forms of non-normality.

* Example Film quality. rate

X_1 X_2 X_3 low

tear gloss opacity high

Example

First of all we'll enter some data relating to the production of plastic film reported in Krzanowski (2000). Tear, gloss and opacity are measures of the manufactured films.

```
tear <- c(6.5, 6.2, 5.8, 6.5, 6.5, 6.9, 7.2, 6.9, 6.1, 6.3,
6.7, 6.6, 7.2, 7.1, 6.8, 7.1, 7.0, 7.2, 7.5, 7.6)
gloss <- c(9.5, 9.9, 9.6, 9.6, 9.2, 9.1, 10.0, 9.9, 9.5, 9.4,
9.1, 9.3, 8.3, 8.4, 8.5, 9.2, 8.8, 9.7, 10.1, 9.2)
opacity <- c(4.4, 6.4, 3.0, 4.1, 0.8, 5.7, 2.0, 3.9, 1.9, 5.7,
2.8, 4.1, 3.8, 1.6, 3.4, 8.4, 5.2, 6.9, 2.7, 1.9)
Y <- cbind(tear, gloss, opacity)
```

We now need to put in information on the rate of extrusion, and the amount of additive used (`gl()` is a command which specifically creates these kind of experimental factors).

```
rate <- factor(gl(2,10), labels=c("Low", "High"))
additive <- factor(gl(2, 5, len=20), labels=c("Low", "High"))
```

There are three conventional ANOVA that could be considered here, but to consider the three responses together we may wish to conduct a MANOVA. However, we can use manova() to fit the multivariate ANOVA, and use `summary.aov()` to extract the results of the unvariate analyses. There are three matrices of interest in MANOVA:

- Between-group **B**
- Within-group **W**
- Total **B + W**

Two way

$X_{ijk} = \mu + \tau_i + \beta_j + \text{interaction} + \varepsilon_{ijk}$

Wilk's Lambda is the ratio $\Lambda^* = \frac{|\mathbf{W}|}{|\mathbf{B}+\mathbf{W}|}$ → Fdist

* Care about interaction

factors.

```
fit <- manova(Y ~ rate * additive)
summary.aov(fit) # univariate ANOVA tables
```

```
##  Response tear :
##                Df Sum Sq Mean Sq F value   Pr(>F)
## rate            1 1.7405 1.74050 15.7868 0.001092 **
## additive        1 0.7605 0.76050  6.8980 0.018330 *
## rate:additive   1 0.0005 0.00050  0.0045 0.947143
## Residuals      16 1.7640 0.11025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response gloss :
##                Df Sum Sq Mean Sq F value  Pr(>F)
## rate            1 1.3005 1.30050  7.9178 0.01248 *
## additive        1 0.6125 0.61250  3.7291 0.07139 .
## rate:additive   1 0.5445 0.54450  3.3151 0.08740 .
## Residuals      16 2.6280 0.16425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response opacity :
##                Df Sum Sq Mean Sq F value Pr(>F)
## rate            1  0.421  0.4205  0.1036 0.7517
## additive        1  4.901  4.9005  1.2077 0.2881
## rate:additive   1  3.960  3.9605  0.9760 0.3379
```

In this example, we have two cases for factors

[rate & additive have interaction

rate & additive don't have interaction

* With problem about Egyptian skulls we have 4 variable

x_1 breath of skull

x_2 basbremath g/d.

x_3 bumPaverlar

x_4 nasal

with factors are time periods

4000 BC

3300 BC

1850 BC.

⇒ they don't have interaction.

pvalue < α the action is significant → affect.

pvalue > α → the interaction is not significant

→ we

```
## Residuals     16 64.924  4.0578
```

A call to summary() will give the MANOVA table.

```
summary(fit, test="Wilks") # ANOVA table of Wilks' lambda

##                Df   Wilks approx F num Df den Df   Pr(>F)
## rate            1 0.38186   7.5543      3     14 0.003034 **
## additive        1 0.52303   4.2556      3     14 0.024745 *
## rate:additive   1 0.77711   1.3385      3     14 0.301782
## Residuals      16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As with Hotellings T^2, Wilk's Lambda has to be converted into an F statistic, a calculation that has been done by the software.

In any case, the interaction term is not significant. We can fit the model without interactions, which as anticipated suggests that both additive and extrusion rate have an effect on the outcome measures. We will use by() to examine the various group means.

{ Don't care about interaction.

```
fit <- manova(Y ~ rate + additive)
summary(fit, test = "Wilks")
```

(care about rate and additive without caring about the interaction)

```
##           Df   Wilks approx F num Df den Df  Pr(>F)
## rate       1 0.38684   7.9253      3     15 0.00212 **
## additive   1 0.55384   4.0279      3     15 0.02753 *
## Residuals 17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
by(Y, rate, summary) ## group means according to extrusion rate

## INDICES: Low
##       tear           gloss           opacity
##  Min.   :5.800   Min.   : 9.100   Min.   :0.800
##  1st Qu.:6.225   1st Qu.: 9.425   1st Qu.:2.250
##  Median :6.500   Median : 9.550   Median :4.000
##  Mean   :6.490   Mean   : 9.570   Mean   :3.790
##  3rd Qu.:6.800   3rd Qu.: 9.825   3rd Qu.:5.375
##  Max.   :7.200   Max.   :10.000   Max.   :6.400
## ------------------------------------------------------------
## INDICES: High
##       tear           gloss           opacity
##  Min.   :6.60   Min.   : 8.300   Min.   :1.600
##  1st Qu.:6.85   1st Qu.: 8.575   1st Qu.:2.725
##  Median :7.10   Median : 9.150   Median :3.600
##  Mean   :7.08   Mean   : 9.060   Mean   :4.080
##  3rd Qu.:7.20   3rd Qu.: 9.275   3rd Qu.:4.925
##  Max.   :7.60   Max.   :10.100   Max.   :8.400
```

```
by(Y, additive, summary) ## group means according to additive

## INDICES: Low
##       tear           gloss           opacity
##  Min.   :5.800   Min.   :8.300   Min.   :0.80
##  1st Qu.:6.500   1st Qu.:8.650   1st Qu.:2.85
##  Median :6.550   Median :9.250   Median :3.60
##  Mean   :6.590   Mean   :9.140   Mean   :3.44
```

```
##  3rd Qu.:6.775   3rd Qu.:9.575   3rd Qu.:4.10
##  Max.   :7.200   Max.   :9.900   Max.   :6.40
## ------------------------------------------------------
## INDICES: High
##       tear          gloss          opacity
##  Min.   :6.10   Min.   : 8.80   Min.   :1.900
##  1st Qu.:6.90   1st Qu.: 9.20   1st Qu.:2.175
##  Median :7.05   Median : 9.45   Median :4.550
##  Mean   :6.98   Mean   : 9.49   Mean   :4.430
##  3rd Qu.:7.20   3rd Qu.: 9.85   3rd Qu.:5.700
##  Max.   :7.60   Max.   :10.10   Max.   :8.400
by(Y, list(rate,additive), summary) ## group means by both.

## : Low
## : Low
##       tear          gloss          opacity
##  Min.   :5.8   Min.   :9.20   Min.   :0.80
##  1st Qu.:6.2   1st Qu.:9.50   1st Qu.:3.00
##  Median :6.5   Median :9.60   Median :4.10
##  Mean   :6.3   Mean   :9.56   Mean   :3.74
##  3rd Qu.:6.5   3rd Qu.:9.60   3rd Qu.:4.40
##  Max.   :6.5   Max.   :9.90   Max.   :6.40
## ------------------------------------------------------
## : High
## : Low
##       tear          gloss          opacity
##  Min.   :6.60   Min.   :8.30   Min.   :1.60
##  1st Qu.:6.70   1st Qu.:8.40   1st Qu.:2.80
##  Median :6.80   Median :8.50   Median :3.40
##  Mean   :6.88   Mean   :8.72   Mean   :3.14
##  3rd Qu.:7.10   3rd Qu.:9.10   3rd Qu.:3.80
##  Max.   :7.20   Max.   :9.30   Max.   :4.10
## ------------------------------------------------------
## : Low
## : High
##       tear          gloss          opacity
##  Min.   :6.10   Min.   : 9.10   Min.   :1.90
##  1st Qu.:6.30   1st Qu.: 9.40   1st Qu.:2.00
##  Median :6.90   Median : 9.50   Median :3.90
##  Mean   :6.68   Mean   : 9.58   Mean   :3.84
##  3rd Qu.:6.90   3rd Qu.: 9.90   3rd Qu.:5.70
##  Max.   :7.20   Max.   :10.00   Max.   :5.70
## ------------------------------------------------------
## : High
## : High
##       tear          gloss          opacity
##  Min.   :7.00   Min.   : 8.8   Min.   :1.90
##  1st Qu.:7.10   1st Qu.: 9.2   1st Qu.:2.70
##  Median :7.20   Median : 9.2   Median :5.20
##  Mean   :7.28   Mean   : 9.4   Mean   :5.02
##  3rd Qu.:7.50   3rd Qu.: 9.7   3rd Qu.:6.90
##  Max.   :7.60   Max.   :10.1   Max.   :8.40
```

High levels of extrusion rate lead to higher levels of tear and opacity but lower levels of gloss. High levels of

addtive lead to higher levels of tear, gloss and opacity.

Multiple Comparisons in MANOVA

If the F-test rejects the null that all g groups have equal mean vectors, we can use multiple comparisons to determine which specific pairs of groups have differing mean vectors (and which components of those mean vectors differ). This is most simply done via Bonferroni confidence intervals for all the differences $\tau_{ki} - \tau_{li}$ for all components $i = 1, \cdots, p$ and all difference $l < k = 1, \cdots, g$. With family confidence level $1 - \alpha$, the set of confidence intervals defined by

$$\bar{x}_{ki} - \bar{x}_{li} \pm t_{n-g}\left(\frac{\alpha}{pg(g-1)}\right)\sqrt{\frac{w_{ii}}{n-g}\left(\frac{1}{n_k} + \frac{1}{n_l}\right)}$$

Here w_{ii} is the i^{th} diagonal element of w. Any such interval that does not contain zero would indicate a significant difference in that component of the mean vector between that pair of groups.

Example: Manova on three groups, using Fisher's classic Iris data

This dataset consists of 50 cases of each of 3 species, namely Iris setosa, Iris virginica, and Iris versicolor. Each case has 4 measurements on the length and width of its petals and sepals.

```
attach(iris)
iris.manova <- manova(cbind(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) ~ Species)
```

Doing the MANOVA test using Wilks' Lambda:

```
summary(iris.manova, test="Wilks")
```

```
##            Df    Wilks approx F num Df den Df    Pr(>F)
## Species     2 0.023439   199.15      8    288 < 2.2e-16 ***
## Residuals 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using the other test statistics:

```
summary(iris.manova, test="Roy")
```

```
##            Df    Roy approx F num Df den Df    Pr(>F)
## Species     2 32.192     1167      4    145 < 2.2e-16 ***
## Residuals 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(iris.manova, test="Hotelling-Lawley")
```

```
##            Df Hotelling-Lawley approx F num Df den Df    Pr(>F)
## Species     2           32.477   580.53      8    286 < 2.2e-16 ***
## Residuals 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(iris.manova, test="Pillai")
```

```
##            Df Pillai approx F num Df den Df    Pr(>F)
## Species     2 1.1919   53.466      8    290 < 2.2e-16 ***
```

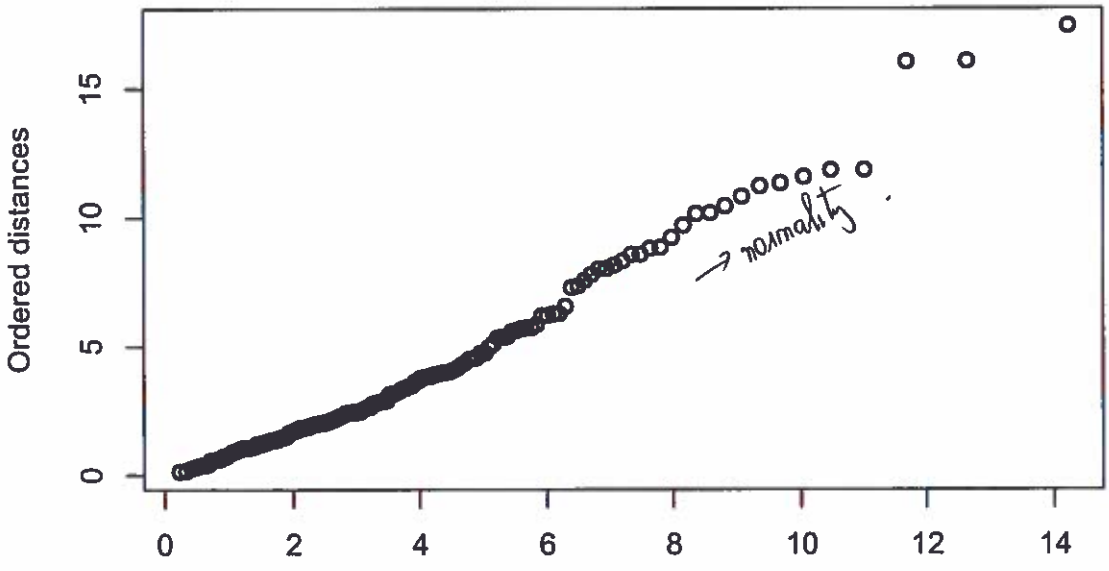
```
## Residuals 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have strong evidence that the mean vectors differ across the 3 species.

Checking model assumption of normality:

```
chisplot <- function(x) {
    if (!is.matrix(x)) stop("x is not a matrix")
    ### determine dimensions
    n <- nrow(x)
    p <- ncol(x)
    xbar <- apply(x, 2, mean)
    S <- var(x)
    S <- solve(S)
    index <- (1:n)/(n+1)
    xcent <- t(t(x) - xbar)
    di <- apply(xcent, 1, function(x,S) x %*% S %*% x,S)
    quant <- qchisq(index,p)
    plot(quant, sort(di), ylab = "Ordered distances",
         xlab = "Chi-square quantile", lwd=2,pch=1)
}
```

```
chisplot(residuals(iris.manova))
```



There is no strong evidence against normality – we are safe. Now we examine the sample covariance matrices for each group:

```
by(iris[,-5], Species, var)
```

```
## Species: setosa
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length   0.12424898 0.099216327  0.016355102 0.010330612
```

```
## Sepal.Width    0.09921633 0.143689796  0.011697959 0.009297959
## Petal.Length   0.01635510 0.011697959  0.030159184 0.006069388
## Petal.Width    0.01033061 0.009297959  0.006069388 0.011106122
## ------------------------------------------------------------
## Species: versicolor
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length   0.26643265  0.08518367   0.18289796  0.05577959
## Sepal.Width    0.08518367  0.09846939   0.08265306  0.04120408
## Petal.Length   0.18289796  0.08265306   0.22081633  0.07310204
## Petal.Width    0.05577959  0.04120408   0.07310204  0.03910612
## ------------------------------------------------------------
## Species: virginica
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length   0.40434286  0.09376327   0.30328980  0.04909388
## Sepal.Width    0.09376327  0.10400408   0.07137959  0.04762857
## Petal.Length   0.30328980  0.07137959   0.30458776  0.04882449
## Petal.Width    0.04909388  0.04762857   0.04882449  0.07543265
```

An R function to test the equality of several covariance matrices:

Need to install the 'asbio' package from the Internet:

```
library(asbio)  # load the 'asbio' package once it is installed
```

```
## Warning: package 'asbio' was built under R version 3.4.4
```

```
## Loading required package: tcltk
```

```
Kullback(Y=iris[,-5],X=Species)
```

```
##
## Kullback test for equal covariance matrices
##            Chi* df  P(Chi>Chi*)
## setosa 73.33162 20 5.158075e-08
```

Another R function to test the equality of several covariance matrices: Need to install the 'biotools' package from the Internet:

```
library(biotools)  # load the 'biotools' package once it is installed
```

```
## Loading required package: rpanel
```

```
## Package `rpanel', version 1.1-3: type help(rpanel) for summary information
```

```
## Loading required package: tkrplot
```

```
## Warning: package 'tkrplot' was built under R version 3.4.4
```

```
## Loading required package: MASS
```

```
## Loading required package: lattice
```

```
## Loading required package: SpatialEpi
```

```
## Warning: package 'SpatialEpi' was built under R version 3.4.4
```

```
## Loading required package: sp
```

```
## ---
## biotools version 3.1
```

```
##
```

X quant 2

Use the box M test to check the equality cov matrices

```
boxM(iris[,-5],Species)

##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  iris[, -5]
## Chi-Sq (approx.) = 140.94, df = 20, p-value < 2.2e-16
```

The covariance matrices may not be equal across groups.

Differences in Means for doing Bonferroni multiple comparisons:

$\Sigma_{150\times 4}$

```
means.by.grps <- cbind(tapply(Sepal.Length,Species,mean),
                       tapply(Sepal.Width,Species,mean),
                       tapply(Petal.Length,Species,mean),
                       tapply(Petal.Width,Species,mean) )
```

The matrix W:

need W

$Var(\text{residuals}) = \frac{\Sigma^T\Sigma}{n-1}$

We want $W = \Sigma^T\Sigma$

```
my.n <- nrow(iris[,-5])
W <- (my.n - 1)*var(residuals(iris.manova))
W
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length      38.9562     13.6300      24.6246      5.6450
## Sepal.Width       13.6300     16.9620       8.1208      4.8084
## Petal.Length      24.6246      8.1208      27.2226      6.2718
## Petal.Width        5.6450      4.8084       6.2718      6.1566
```

Calculation of the Bonferroni CIs:

```
my.alpha <- 0.10 # family confidence level for CIs is 90% here

my.m <- 3  # number of groups
my.q <- 4  # number of variables
group.sample.sizes <- c(50,50,50)

for (k in 1:my.q) {

pair.mean.diffs <- cbind( t(combn(my.m,2)),
                          combn(tapply(iris[,k],Species,mean),2,FUN=diff) )

t.val <- qt(my.alpha/(my.q*my.m*(my.m-1)), df=my.n-my.m, lower=F)

#print(paste("For i=",i,"i.prime=",j,"and k=",k,"pt est is"

CI.L <- pair.mean.diffs[,3] -
  t.val*sqrt((diag(W)[k]/(my.n-my.m))*
 (1/group.sample.sizes[pair.mean.diffs[,1]] +
    1/group.sample.sizes[pair.mean.diffs[,2]]) )
CI.U <- pair.mean.diffs[,3] +
  t.val*sqrt((diag(W)[k]/(my.n-my.m))*
 (1/group.sample.sizes[pair.mean.diffs[,1]] +
    1/group.sample.sizes[pair.mean.diffs[,2]]) )

my.table.mat<-cbind(pair.mean.diffs,round(CI.L,3),round(CI.U,3),
                    rep(k,times=nrow(pair.mean.diffs)) )
```

```
my.table<-as.data.frame(my.table.mat)
names(my.table)=c('grp1','grp2','diff.samp.means',
                  'lower.CI','upper.CI','variable');
print(my.table)
}
```

Compare groups

not c — not contain 0 → reject, there is no diference ⇒ diference between $g1 \neq g2$

X_1 sepal length

	grp1	grp2	diff.samp.means	lower.CI	upper.CI	variable
1	1	2	0.930	0.655	1.205	1
2	1	3	1.582	1.307	1.857	1
3	2	3	0.652	0.377	0.927	1

$\mu_{X_1,F1}$ $\mu_{X_1,F2}$ $\neq 0$

$\mu_{X_1,F1}$ $\mu_{X_1,F3}$

$\mu_{X_1,F2}$ $\mu_{X_1,F3}$.

X_2

	grp1	grp2	diff.samp.means	lower.CI	upper.CI	variable
1	1	2	-0.658	-0.840	-0.476	2
2	1	3	-0.454	-0.636	-0.272	2
3	2	3	0.204	0.022	0.386	2

X_3

	grp1	grp2	diff.samp.means	lower.CI	upper.CI	variable
1	1	2	2.798	2.568	3.028	3
2	1	3	4.090	3.860	4.320	3
3	2	3	1.292	1.062	1.522	3

X_4

	grp1	grp2	diff.samp.means	lower.CI	upper.CI	variable
1	1	2	1.08	0.971	1.189	4
2	1	3	1.78	1.671	1.889	4
3	2	3	0.70	0.591	0.809	4

lower CI and upper CI of the $\hat{\mu}_i - \hat{\mu}_j$

X_1 Sepal Length — X_2 Sepal Width — X_3 petal length — X_4 Petal Width

1 Factor

Iris setosa	1
Iris virginica	2
Iris versicolor	3 .

we want to find β.

Chapter 3. Multivariate Regression

Jianxuan Liu

Fall 2018

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1r} \\ 1 & X_{21} & & X_{2r} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & & X_{nr} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_r \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}_{n\times 1}$$

$n\times 1$ $\quad$ $n\times(r+1)$ $\quad$ $(r+1)\times 1$

1. Classical Linear Regression Model

$$\underset{\sim}{Y} = X\underset{\sim}{\beta} + \underset{\sim}{\epsilon}$$

where $\mathbf{X}$ is a $n \times (r+1)$ data matrix, $\beta = [\beta_0, \beta_1, \cdots, \beta_r]$ unknown parameters, $\mathbf{Y}$ is a $n \times 1$ vector and the error terms have the following properties:

- $E(\epsilon) = \mathbf{0}$
- $Cov(\epsilon) = E(\epsilon\epsilon^T) = \sigma^2\mathbf{I}$.

The least square estimate

Theorem 1. Let $\mathbf{X}$ have full rank $r+1 \le n$. The least squares estimate of β is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

$$\hat{\beta}_{LS} = \arg\min_{\beta} (Y - X\beta)^T (Y - X\beta)$$

Let $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = H\mathbf{Y}$ denote the fitted values of $\mathbf{Y}$, where $H = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. Then the residuals

$$\hat{\epsilon} = \mathbf{Y} - \hat{\mathbf{Y}} = [I - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\mathbf{Y} = [I - H]\mathbf{Y}$$

satisfy $\mathbf{X}^T\hat{\epsilon} = \mathbf{0}$ and $\hat{\mathbf{Y}}^T\hat{\epsilon} = 0$. real ↑ approximated ↖

Theorem 2. The least squares estimator $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ is unbiased, i.e.

$$E(\hat{\beta}) = \beta$$

and

$$Cov(\hat{\beta}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}.$$

The residuals $\hat{\epsilon}$ have the properties

- $E(\hat{\epsilon}) = \mathbf{0}$
- $Cov(\hat{\epsilon}) = \sigma^2[I - H]$
- $E(\hat{\epsilon}^T\hat{\epsilon}) = (n - r - 1)\sigma^2$
- $E(s^2) = \sigma^2$ where $s^2 = \dfrac{\hat{\epsilon}^T\hat{\epsilon}}{n-r-1} = \dfrac{\mathbf{Y}^T[I-H]\mathbf{Y}}{n-r-1}$.
- $\hat{\beta}$ and $\hat{\epsilon}$ are uncorrelated.

Theorem 3. Given that $\mathbf{X}$ has full rank $r+1$ and $\epsilon \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$, the maximum likelihood estimator of β is the same as the least squares estimator, $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. Moreover, $\hat{\beta} \sim N_{r+1}(\beta, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$ and $\hat{\beta}$ is independent of the residuals $\hat{\epsilon} = \mathbf{Y} - \hat{\mathbf{Y}}$. Further,

$$n\hat{\sigma}^2 = \hat{\epsilon}^T\hat{\epsilon} \sim \sigma^2\chi^2_{n-r-1}$$

where $\hat{\sigma}^2$ is the maximum likelihood estimator of σ^2.

Theorem 4. The simultaneous $100(1-\alpha)\%$ confidence region for β_i are

$$\hat{\beta}_i \pm \sqrt{(r+1)F_{r+1,n-r-1}(\alpha)\hat{Var}(\hat{\beta}_i)}, i = 0, 1, \cdots, r.$$

For a new observation $\mathbf{x}_0^T = [1, x_{01}, \cdots, x_{0r}]$, the estimate of the expected values of Y_0 is $\hat{Y}_0 = \mathbf{x}_0^T\hat{\beta}$. If the error terms are normally distributed, then a $100(1-\alpha)\%$ confidence interval of $E(Y_0 \mid \mathbf{x}_0)$ is

$$\mathbf{x}_0^T\hat{\beta} \pm t_{n-r-1,\alpha/2}\sqrt{(\mathbf{x}_0^T(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}_0)s^2}.$$

A $100(1-\alpha)\%$ prediction interval of Y_0 is

$$\mathbf{x}_0^T\hat{\beta} \pm t_{n-r-1,\alpha/2}\sqrt{(1+\mathbf{x}_0^T(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}_0)s^2}.$$

2. Multivariate Regression

In multivariate regression we wish to predict or explain a set of m response (or dependent) variables $\mathbf{Y}_1, \cdots, \mathbf{Y}_m$ via a set of r predictor (or independent) variables $\mathbf{x}_1, ..., \mathbf{x}_r$. The classical linear regression model can be written in matrix-vector form as

m response variables, each is a $(n\times 1)$ random vector

$$\underset{n\times m}{\mathbf{Y}} = \underset{n\times(r+1)}{\mathbf{x}}\underset{(r+1)\times m}{\beta} + \underset{n\times m}{\epsilon}$$

where

- $\mathbf{x}$ is a $n \times (r+1)$ data matrix
- $\beta = [\beta_{(1)} \vdots \beta_{(2)} \vdots \cdots \vdots \beta_{(m)}]$
- $\mathbf{Y} = [\mathbf{Y}_{(1)} \vdots \mathbf{Y}_{(2)} \vdots \cdots \vdots \mathbf{Y}_{(m)}]$ is a $n \times m$ matrix
- $\epsilon = [\epsilon_{(1)} \vdots \epsilon_{(2)} \vdots \cdots \vdots \epsilon_{(m)}]$
- $E(\epsilon_{(i)}) = \mathbf{0}$
- $Cov(\epsilon_{(i)}, \epsilon_{(j)}) = E(\epsilon_{(i)}\epsilon_{(j)}^T) = \sigma_{ij}^2\mathbf{I}$.

$$\begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1m} \\ \vdots & \vdots & & \vdots \\ Y_{n1} & Y_{n2} & & Y_{nm} \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1r} \\ 1 & x_{21} & & x_{2r} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & & x_{nr} \end{bmatrix} \begin{bmatrix} \beta_{01} & \beta_{02} & \cdots & \beta_{0m} \\ \beta_{11} & \beta_{12} & & \beta_{1m} \\ \vdots & \vdots & & \vdots \\ \beta_{r1} & \beta_{r2} & & \beta_{rm} \end{bmatrix} + \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \cdots & \epsilon_{1m} \\ \vdots & \vdots & & \vdots \\ \epsilon_{n1} & \epsilon_{n2} & & \epsilon_{nm} \end{bmatrix}$$

person, response, person, predictor

Further Explanation of the Multivariate Regression Model

- The n rows of $\mathbf{Y}$ correspond to the n different individuals.
- The m columns of $\mathbf{Y}$ correspond to the m different response variables.
- Note that the first row of β is a row of intercept terms corresponding to the m response variables.
- Then the $(i+1, j)$ entry of β measures the marginal effect of the i-th predictor variable on the j-th response variable.

Theorem 5.: The least squares estimate of β is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

In particular, for the i^{th} response, $\hat{\beta}_{(i)} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_{(i)}$.

- $E(\hat{\beta}_{(i)}) = \beta_{(i)}$, or $E(\hat{\beta}) = \beta$
- $Cov(\hat{\beta}_{(i)}, \hat{\beta}_{(k)}) = \sigma_{ik}(\mathbf{X}^T\mathbf{X})^{-1}, i, k = 1, 2, \cdots, m$
- $E(\hat{\epsilon}) = \mathbf{0}$ and $E(\hat{\epsilon}^T\hat{\epsilon}) = (n-r-1)\Sigma$

Theorem 6.: Let $\mathbf{P} = \boldsymbol{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \boldsymbol{I}_n - \boldsymbol{H}$, then the maximum likelihood estimators are $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ and $\hat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{Y}^T\mathbf{P}\mathbf{Y}$.

Proof:

Assume $rank(\mathbf{X}) = r$ and $n > m + r$.

$$L(\mathbf{Y}; \beta, \mathbf{\Sigma}) = \prod_{i=1}^{n} \mid 2\pi\mathbf{\Sigma} \mid^{-1/2} \exp\{-\frac{1}{2}(\mathbf{y}_i - \mathbf{x}_i\beta)\mathbf{\Sigma}^{-1}(\mathbf{y}_i - \mathbf{x}_i\beta)^T\}$$

implies

$$l(\mathbf{Y}; \beta, \mathbf{\Sigma}) = -\frac{n}{2}\log \mid 2\pi\mathbf{\Sigma} \mid -\frac{1}{2}tr\{(\mathbf{Y} - \mathbf{X}\beta)\mathbf{\Sigma}^{-1}(\mathbf{Y} - \mathbf{X}\beta)^T\}.$$

Write

$$l(\beta, \mathbf{\Sigma}) = -\frac{n}{2}\log \mid 2\pi\mathbf{\Sigma} \mid -\frac{1}{2}tr\{\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}\} - \frac{1}{2}tr\{\mathbf{\Sigma}^{-1}(\hat{\beta} - \beta)^T\mathbf{X}^T\mathbf{X}(\hat{\beta} - \beta)\}.$$

Here, $(\hat{\beta} - \beta)^T\mathbf{X}^T\mathbf{X}(\hat{\beta} - \beta)$ is independent of $\mathbf{\Sigma}$. $l(\beta, \mathbf{\Sigma})$ is maximized when $(\hat{\beta} - \beta)^T\mathbf{X}^T\mathbf{X}(\hat{\beta} - \beta)$ is minimized which is at $\beta = \hat{\beta}$. Plugging this into the log-likelihood we get

$$l(\hat{\beta}, \mathbf{\Sigma}) = -\frac{np}{2}\log \mid 2\pi \mid -\frac{n}{2}(\log \mid \mathbf{\Sigma} \mid + tr\{\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}\})$$

which is maximized at $\mathbf{\Sigma} = \hat{\mathbf{\Sigma}}$. Note that $(\hat{\beta}, \hat{\mathbf{\Sigma}})$ is sufficient for $(\beta, \mathbf{\Sigma})$.

Theorem 7.: $\hat{\beta}$ and $\hat{\mathbf{\Sigma}}$ are independent.

Proof*:

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{Y}^T\mathbf{P}\mathbf{Y} = \frac{1}{n}(\mathbf{Y} - \beta\mathbf{X})^T P(\mathbf{Y} - \beta\mathbf{X})$$

$$\hat{\beta} - \beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}(\mathbf{Y} - \beta\mathbf{X})$$

where $\mathbf{Y} - \beta\mathbf{X} \sim N_m(\mathbf{0}, \mathbf{\Sigma})$. Note that $\mathbf{P}\mathbf{Y}$ and $\hat{\beta} - \beta$ are independent because $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{P} = \mathbf{0}$.

The Multivariate Regression Model Assumptions

- We assume that all of the nm elements of ϵ have mean 0.
- Any single row of ϵ has covariance matrix $\mathbf{\Sigma}$ (generally non-diagonal). This implies that the response variables **within** an individual multivariate observation may be correlated.
- However, we also assume that response values from different individuals are uncorrelated.
- For doing inference about the multivariate regression model, we further assume that each column of ϵ has a multivariate normal distribution.

Likelihood Ratio Tests for Regression Parameters

We may we may wish to test whether a set of several predictor variables is not related to the set of response variables.

$$H_0 : \beta_{(2)} = \mathbf{0}$$

where $\beta = [\beta_{(1)} \vdots \beta_{(2)}]^T$. $\beta_{(1)}$ is the first $q+1$ of the predictors are related to the set of response variables, and the last $r - q$ are useless in predicting the set of response variables. The likelihood ratio Λ can be expressed in terms of generalized variances

$$\Lambda = \frac{\max_{\beta_{(1)}, \mathbf{\Sigma}} L(\beta_{(1)}, \mathbf{\Sigma})}{\max_{\beta, \mathbf{\Sigma}} L(\beta, \mathbf{\Sigma})} = \frac{L(\hat{\beta}_{(1)}, \hat{\mathbf{\Sigma}})}{L(\hat{\beta}, \hat{\mathbf{\Sigma}})} = \left(\frac{\mid \hat{\mathbf{\Sigma}} \mid}{\mid \hat{\mathbf{\Sigma}}_1 \mid}\right)^{n/2}$$

Equivalently, Wilk's lambda statistic is

$$\Lambda^{2/n} = \frac{|\hat{\Sigma}|}{|\hat{\Sigma}_1|}.$$

Under H_0, $\mathbf{Y} = \mathbf{X}_1\beta_{(1)} + \epsilon$ and the likelihood ratio test of H_0 is equivalent to rejecting H_0 for large values of

$$-2\ln\Lambda = -n\ln\frac{|\hat{\Sigma}|}{|\hat{\Sigma}_1|} = -n\ln\frac{|n\hat{\Sigma}|}{|n\hat{\Sigma} + n(\hat{\Sigma} - \hat{\Sigma}_1)|}.$$

For n larger, the modified statistic

$$-\left[n - r - 1 - \frac{1}{2}(m - r + q + 1)\right]\ln\frac{|\hat{\Sigma}|}{|\hat{\Sigma}_1|}$$

can be approximated as a $\chi^2_{m(r-q)}$.

Example 1: The Computer Data

```
comput <- read.table("computerdata.txt", header=T)
#comput <- read.table(file.choose(), header=T)

attach(comput)

# y1 = CPU time (in hours)
# y2 = disk input/output capacity
# x1 = customer orders (in thousands)
# x2 = add-delete items (in thousands)

# Fitting the multivariate linear regression model:

comp.mod.y1 <- lm(y1 ~ x1 + x2)
comp.mod.y2 <- lm(y2 ~ x1 + x2)

Beta.hat <- cbind(coef(comp.mod.y1), coef(comp.mod.y2) )
Beta.hat
```

```
##                  [,1]      [,2]
## (Intercept) 8.4236890 14.141491
## x1          1.0789825  2.253854
## x2          0.4198885  5.665367
```

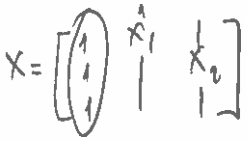
```
# A quicker way to get a summary:
Y <- as.matrix(comput[,c("y1","y2")])
comp.mod <- lm(Y ~ x1 + x2)
summary(comp.mod)
```

```
## Response y1 :
##
## Call:
## lm(formula = y1 ~ x1 + x2)
##
## Residuals:
##       1       2       3       4       5       6       7
## -1.0632 -1.0315  1.1007 -0.9151  0.4650  1.1066  0.3375
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.42369    3.44328   2.446   0.0707 .
## x1           1.07898    0.02749  39.249 2.52e-06 ***
## x2           0.41989    0.14447   2.906   0.0438 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.204 on 4 degrees of freedom
## Multiple R-squared:  0.9979, Adjusted R-squared:  0.9969
## F-statistic: 966.5 on 2 and 4 DF,  p-value: 4.265e-06
##
##
## Response y2 :
##
## Call:
## lm(formula = y2 ~ x1 + x2)
##
## Residuals:
##       1       2       3       4       5       6       7
## -2.6350  0.4754  1.4750 -0.9790  0.7887 -0.3555  1.2304
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.14149    5.06912    2.79   0.0493 *
## x1           2.25385    0.04047   55.69 6.22e-07 ***
## x2           5.66537    0.21269   26.64 1.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.772 on 4 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.999
## F-statistic:  2932 on 2 and 4 DF,  p-value: 4.646e-07
```

 # Getting the matrix of fitted values:

```
X.mat <- cbind(rep(1,times=nrow(comput)),x1,x2)
```



```
Y.hat <- X.mat %*% Beta.hat
Y.hat
```

```
##          [,1]     [,2]
## [1,] 142.5632 304.4350
## [2,] 169.9315 395.6246
## [3,] 153.6993 326.7250
## [4,] 147.4151 308.3790
## [5,] 172.3350 361.6113
## [6,] 158.9934 369.8555
## [7,] 108.1625 227.8696
```

 # Getting the matrix of residuals:

```
resid.mat <- cbind (y1,y2) - Y.hat
resid.mat
```

```
##             y1         y2
```

```
## [1,] -1.0631528 -2.6350279
## [2,] -1.0314649  0.4754450
## [3,]  1.1006666  1.4749621
## [4,] -0.9151495 -0.9789776
## [5,]  0.4649604  0.7887049
## [6,]  1.1065938 -0.3555287
## [7,]  0.3375464  1.2304222
```

```
# Note we could get these matrices from the individual regression models as well:

cbind(fitted(comp.mod.y1), fitted(comp.mod.y2))
```

```
##       [,1]     [,2]
## 1 142.5632 304.4350
## 2 169.9315 395.6246
## 3 153.6993 326.7250
## 4 147.4151 308.3790
## 5 172.3350 361.6113
## 6 158.9934 369.8555
## 7 108.1625 227.8696
```

```
cbind(resid(comp.mod.y1), resid(comp.mod.y2))
```

```
##         [,1]       [,2]
## 1 -1.0631528 -2.6350279
## 2 -1.0314649  0.4754450
## 3  1.1006666  1.4749621
## 4 -0.9151495 -0.9789776
## 5  0.4649604  0.7887049
## 6  1.1065938 -0.3555287
## 7  0.3375464  1.2304222
```

```
#### Testing about x2 in the model:

## Full model:

my.n <- length(y1)  # number of individuals
my.r <- ncol(X.mat) - 1  # total number of predictors
my.m <- ncol(resid.mat)  # total number of responses

Sigma..full <- (my.n-1)*var(resid.mat)/my.n
# or
Sigma..full <- t(resid.mat)%*%resid.mat/my.n
# or
Sigma..full <- crossprod(resid.mat)/my.n


## Reduced model (without x2)
red.mod <- lm(Y ~ x1)
Beta.hat.redu <- cbind(coef(red.mod))
```

Note: If we were testing the null hypothesis that the *entire set* of predictors was useless in the model, our reduced model would contain ONLY an intercept term. We could fit such a reduced model using:

```
Beta.hat.redu <- cbind(coef(lm(y1 ~ 1) ), coef(lm(y2 ~ 1) ) )
```

```
X.mat.redu <- cbind(rep(1,times=nrow(comput)),x1)
my.q <- ncol(X.mat.redu) - 1  # number of predictors in the reduced model

Y.hat.redu <- X.mat.redu %*% Beta.hat.redu

resid.mat.redu <- cbind (y1,y2) - Y.hat.redu

Sigma..redu <- (my.n-1)*var(resid.mat.redu)/my.n

Wilks <- det(Sigma..full)/det(Sigma..redu)
Wilks
```

```
## [1] 0.004002217
```

```
my.test.stat <-  -(my.n - my.r - 1 - 0.5*(my.m - my.r + my.q + 1)) * log(Wilks )
my.test.stat
```

```
## [1] 16.56272
```

```
p.value <- pchisq(my.test.stat, df = my.r*(my.r - my.q), lower.tail=F )
p.value
```

```
## [1] 0.0002531925
```

```
## A quicker way to test H_0
# Estimated Error Covariance Matrix
hat_epsilon2 <- crossprod(Y - comp.mod$fitted.values)
#SigmaTilde <- hat_epsilon2 / (my.n - my.r - 1) # Sums-of-Squares and Crossproducts
SigmaHat <- hat_epsilon2 / my.n #MLE of \Sigma
SigmaHat
```

```
##           y1        y2
## y1 0.8282650 0.7455714
## y2 0.7455714 1.7951036
```

```
SigmaHat1 <- crossprod(Y - red.mod$fitted.values) / my.n #MLE of \Sigma for redu model
Wilks <- det(SigmaHat)/det(SigmaHat1)
Wilks
```

```
## [1] 0.004002217
```

```
# modified statistic for large n
-(my.n - my.r - 1 - 0.5*(my.m - my.r + my.q + 1)) *log(Wilks)
```

```
## [1] 16.56272
```

```
## Even quicker.
anova(comp.mod, red.mod, test="Wilks")
```

```
## Analysis of Variance Table
##
## Model 1: Y - x1 + x2
## Model 2: Y - x1
##   Res.Df Df Gen.var.     Wilks approx F num Df den Df    Pr(>F)
## 1      4      1.6885
## 2      5  1  21.3521 0.0040022   373.29      2      3 0.0002532 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Confidence Ellipsoids and Prediction Ellipsoids in Multivariate Regression

Suppose we have a new individual whose values of the predictor variables are known but whose values for the response variables are not available. A point prediction of $[Y_1, Y_2, \cdots, Y_m]$ for this individual is simply $\mathbf{x}_0^T\hat{\boldsymbol{\beta}}$ where $\mathbf{x}_0 = [1, \mathbf{x}_{10}, \cdots, \mathbf{x}_{r0}]$ contains the known values of the predictor variables for that individual.

An m-dimensional $100(1-\alpha)\%$ **Confidence ellipsoid** for the individual responses $\mathbf{Y}_{0i}$ are

$$\mathbf{X}_0^T\hat{\boldsymbol{\beta}}_{(i)} \pm \sqrt{\left(\frac{m(n-r-1)}{n-r-m}F_{m,n-r-m}(\alpha)\right)}\sqrt{\mathbf{x}_0^T(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}_0\left(\frac{n}{n-r-1}\hat{\sigma}_{ii}\right)}, i = 1, 2, \cdots, m$$

An m-dimensional $100(1-\alpha)\%$ **prediction ellipsoid** for the individual responses $\mathbf{Y}_{0i}$ are

$$\mathbf{X}_0^T\hat{\boldsymbol{\beta}}_{(i)} \pm \sqrt{\left(\frac{m(n-r-1)}{n-r-m}F_{m,n-r-m}(\alpha)\right)}\sqrt{(1+\mathbf{x}_0^T(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}_0)\left(\frac{n}{n-r-1}\hat{\sigma}_{ii}\right)}, i = 1, 2, \cdots, m$$

Suppose we have a new site with 130 thousand orders and 7.5 thousand add-delete items. Let's get a 95% prediction ellipse for that site's (CPU time, disk input/output capacity). The predictor values of interest (including a "1" for the intercept term):

```
pred.mlm <- function(object, newdata, level=0.95,
                     interval = c("confidence", "prediction")){
  form <- as.formula(paste("~",as.character(formula(object))[3]))
  xnew <- model.matrix(form, newdata)
  fit <- predict(object, newdata)
  Y <- model.frame(object)[,1]
  X <- model.matrix(object)
  n <- nrow(Y)
  m <- ncol(Y)
  p <- ncol(X) - 1
  sigmas <- colSums((Y - object$fitted.values)^2) / (n - p - 1)
  fit.var <- diag(xnew %*% tcrossprod(solve(crossprod(X)), xnew))
  if(interval[1]=="prediction") fit.var <- fit.var + 1
  const <- qf(level, df1=m, df2=n-p-m) * m * (n - p - 1) / (n - p - m)
  vmat <- (n/(n-p-1)) * outer(fit.var, sigmas)
  lwr <- fit - sqrt(const) * sqrt(vmat)
  upr <- fit + sqrt(const) * sqrt(vmat)
  if(nrow(xnew)==1L){
    ci <- rbind(fit, lwr, upr)
    rownames(ci) <- c("fit", "lwr", "upr")
  } else {
    ci <- array(0, dim=c(nrow(xnew), m, 3))
    dimnames(ci) <- list(1:nrow(xnew), colnames(Y), c("fit", "lwr", "upr") )
    ci[,,1] <- fit
    ci[,,2] <- lwr
    ci[,,3] <- upr
  }
  ci
}


newdata <- data.frame(x1=130, x2=7.5)
# confidence interval
pred.mlm(comp.mod, newdata)
```

```
##           y1       y2
## fit 151.8406 349.6327
## lwr 146.9515 342.4351
## upr 156.7297 356.8303
```

```
# prediction interval
pred.mlm(comp.mod, newdata, interval="prediction")
```

```
##           y1       y2
## fit 151.8406 349.6327
## lwr 142.4323 335.7821
## upr 161.2488 363.4833
```

```
# confidence interval (multiple new observations)
newdata <- data.frame(x1=c(130,140),x2=c(7.5,8.5))
pred.mlm(comp.mod, newdata)
```

```
## , , fit
##
##         y1       y2
## 1 151.8406 349.6327
## 2 163.0503 377.8366
##
## , , lwr
##
##         y1       y2
## 1 146.9515 342.4351
## 2 157.7044 369.9666
##
## , , upr
##
##         y1       y2
## 1 156.7297 356.8303
## 2 168.3962 385.7067
```

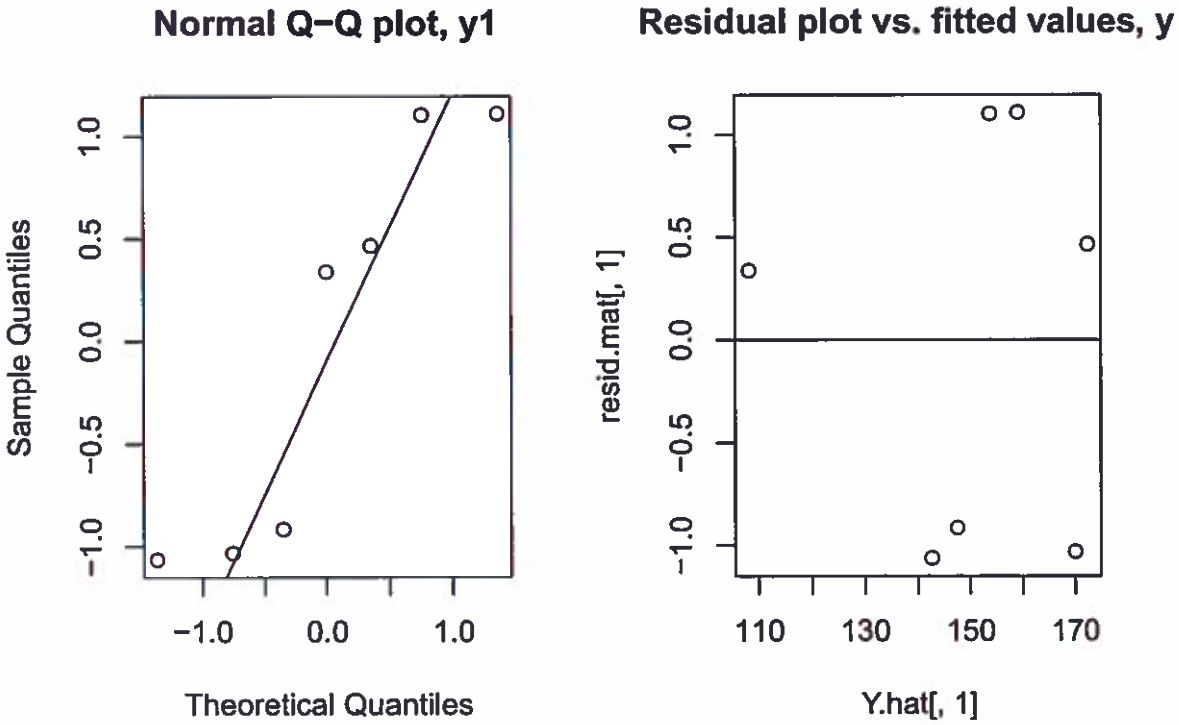
```
# prediction interval (multiple new observations)
pred.mlm(comp.mod, newdata, interval="prediction")
```

```
## , , fit
##
##         y1       y2
## 1 151.8406 349.6327
## 2 163.0503 377.8366
##
## , , lwr
##
##         y1       y2
## 1 142.4323 335.7821
## 2 153.3968 363.6250
##
## , , upr
##
##         y1       y2
## 1 161.2488 363.4833
## 2 172.7038 392.0483
```

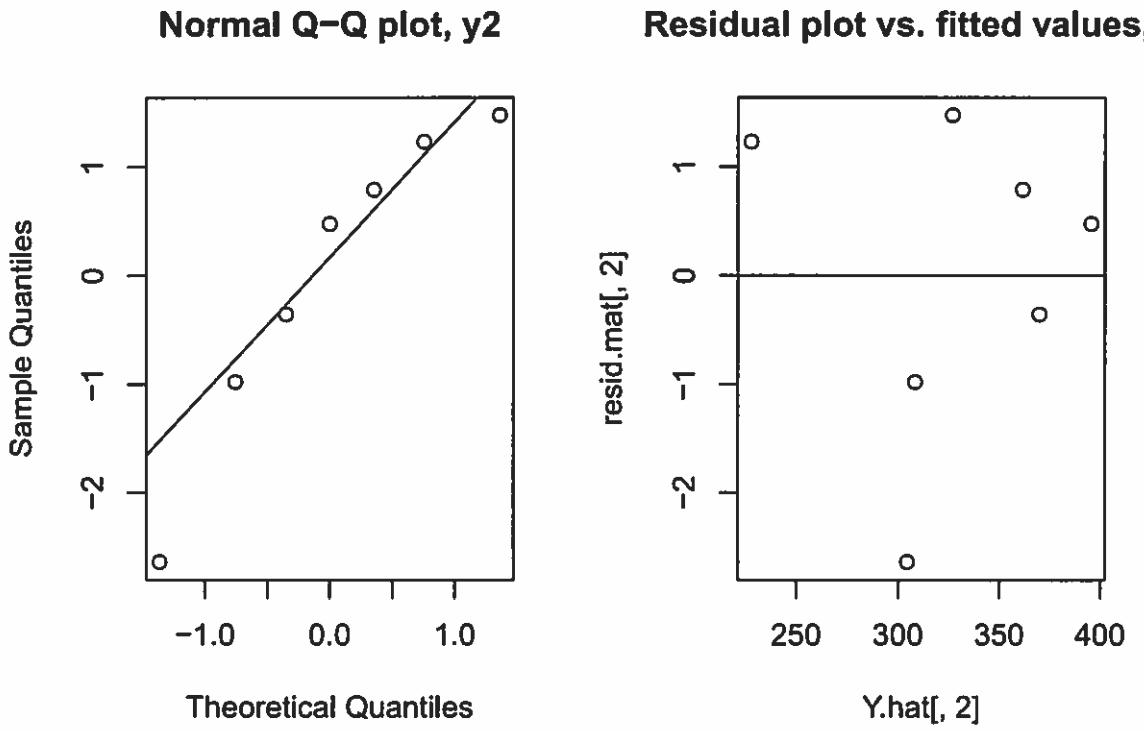
Checking Model Assumptions in Multivariate Regression

- The model assumptions should be checked in multivariate regression using techniques similar to those used in simple linear regression or multiple linear regression.
- To check the normality of the error terms, a normal QQ plot of the residual vectors $\epsilon_{(1)}, \epsilon_{(2)}, \cdots, \epsilon_{(m)}$ for each response variable can be examined.
- For each response variable, the residual vector can be plotted against the vector of fitted values to look for outliers or unusual patterns.
- Transformations of one or more response variables may be tried if violations of the model assumptions are apparent.

```
### Checking model assumptions:
par(mfrow=c(1,2))
qqnorm(resid.mat[,1], main = "Normal Q-Q plot, y1")
qqline(resid.mat[,1])
plot(Y.hat[,1],resid.mat[,1], main = "Residual plot vs. fitted values, y1"); abline(h=0)
```




```
qqnorm(resid.mat[,2], main = "Normal Q-Q plot, y2")
qqline(resid.mat[,2])
plot(Y.hat[,2],resid.mat[,2], main = "Residual plot vs. fitted values, y2"); abline(h=0)
```



Example 2: The sales performance data:

```
salesdata <- read.table("salesmat.txt", header=T)

attach(salesdata)

## The following objects are masked from comput:
##
##     x1, x2
# Recall the variables:
# X1 = Sales growth
# X2 = Sales profitability
# X3 = New account sales
# X4 = Creativity Test
# X5 = Mechanical Reasoning Test
# X6 = Abstract Reasoning Test
# X7 = Mathematics Test
# X8 = Historical Facts Test
# X9 = Sports Trivia Test
# X10 = Music Trivia Test

# We will try to predict the "performance" variables (x1, x2, x3) using the "test score" variables (x4,

# Fitting the multivariate linear regression model:

sales.mod.1 <- lm(x1 ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
sales.mod.2 <- lm(x2 ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
sales.mod.3 <- lm(x3 ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
```

```
# A quick summary:
sales.mod <- lm(cbind(x1,x2,x3) ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
summary(sales.mod)

## Response x1 :
##
## Call:
## lm(formula = x1 ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9553 -0.9130  0.4151  1.2096  2.0329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.27323    1.67778  41.289  < 2e-16 ***
## x4           0.38292    0.07790   4.915 1.40e-05 ***
## x5           0.29869    0.10143   2.945  0.00525 **
## x6           0.69855    0.13863   5.039 9.38e-06 ***
## x7           0.45469    0.03288  13.830  < 2e-16 ***
## x8          -0.01418    0.02128  -0.666  0.50877
## x9           0.02917    0.02515   1.160  0.25273
## x10         -0.01121    0.02521  -0.445  0.65886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.71 on 42 degrees of freedom
## Multiple R-squared:  0.9536, Adjusted R-squared:  0.9459
## F-statistic: 123.3 on 7 and 42 DF,  p-value: < 2.2e-16
##
##
## Response x2 :
##
## Call:
## lm(formula = x2 ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7704 -1.5003 -0.1821  1.1856  3.6953
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 76.48366    1.87834  40.719  < 2e-16 ***
## x4           0.17506    0.08722   2.007   0.0512 .
## x5           0.83095    0.11356   7.317 5.13e-09 ***
## x6          -0.48542    0.15520  -3.128   0.0032 **
## x7           0.78419    0.03681  21.306  < 2e-16 ***
## x8          -0.02167    0.02382  -0.910   0.3681
## x9          -0.03491    0.02816  -1.240   0.2220
## x10         -0.02057    0.02822  -0.729   0.4701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

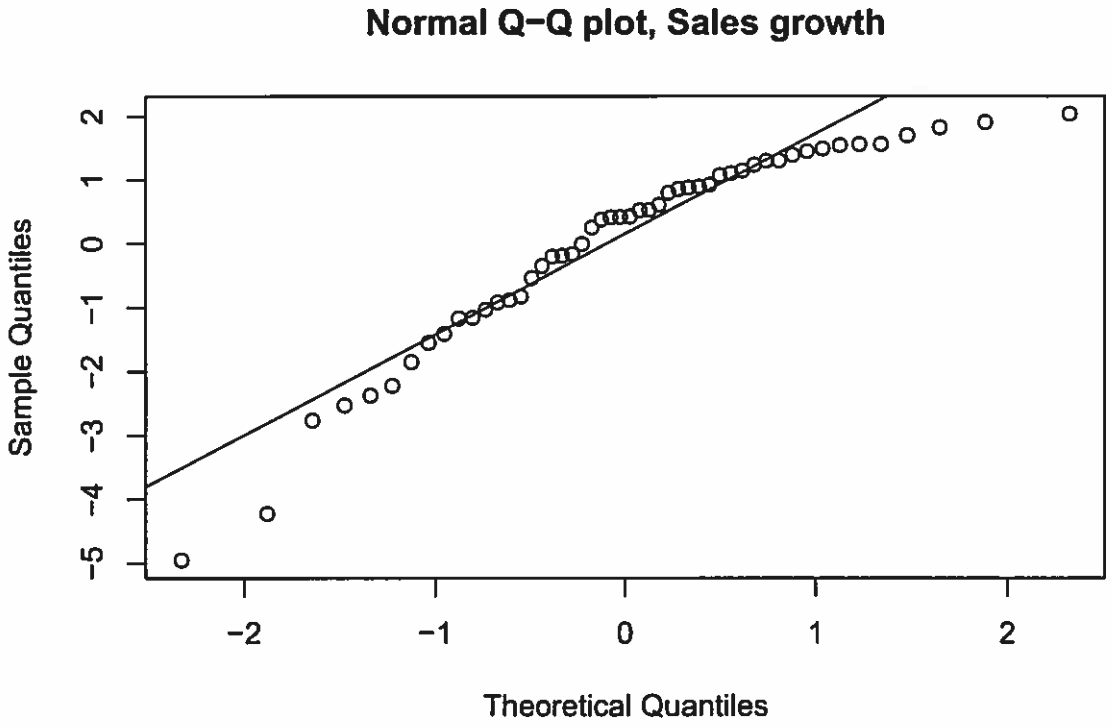
```
## Residual standard error: 1.914 on 42 degrees of freedom
## Multiple R-squared:  0.9693, Adjusted R-squared:  0.9642
## F-statistic: 189.4 on 7 and 42 DF,  p-value: < 2.2e-16
##
##
## Response x3 :
##
## Call:
## lm(formula = x3 ~ x4 + x5 + x6 + x7 + x8 + x9 + x10)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4800 -0.6304 -0.0103  0.6852  2.7140
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 84.057314   1.327509  63.320  < 2e-16 ***
## x4           0.531052   0.061640   8.615 7.88e-11 ***
## x5          -0.059393   0.080257  -0.740    0.463
## x6           0.672190   0.109689   6.128 2.60e-07 ***
## x7           0.229394   0.026012   8.819 4.16e-11 ***
## x8          -0.010272   0.016834  -0.610    0.545
## x9          -0.009681   0.019902  -0.486    0.629
## x10          0.009646   0.019947   0.484    0.631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.353 on 42 degrees of freedom
## Multiple R-squared:  0.9298, Adjusted R-squared:  0.9181
## F-statistic: 79.46 on 7 and 42 DF,  p-value: < 2.2e-16
```

```
Beta.hat <- cbind(coef(sales.mod.1), coef(sales.mod.2), coef(sales.mod.3) )
Beta.hat
```

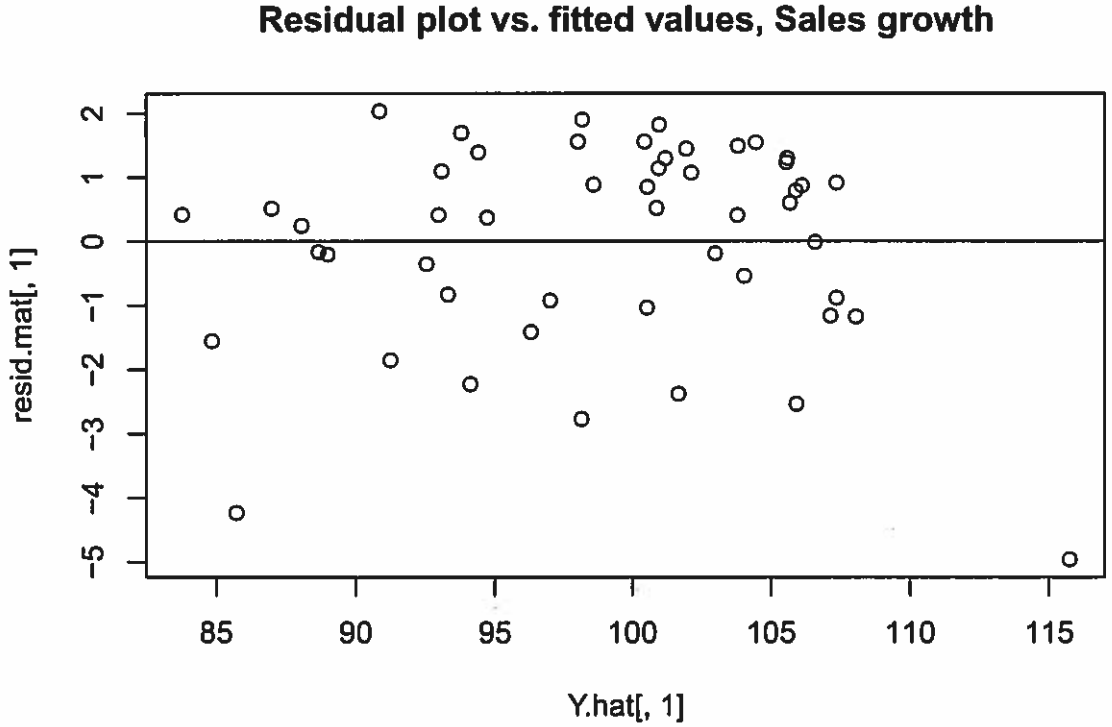
```
##                    [,1]        [,2]         [,3]
## (Intercept) 69.27322626 76.48366386 84.057313608
## x4           0.38292240  0.17505706  0.531051725
## x5           0.29869074  0.83095235 -0.059393242
## x6           0.69854730 -0.48542385  0.672190486
## x7           0.45468653  0.78419335  0.229393734
## x8          -0.01417885 -0.02166938 -0.010271937
## x9           0.02917020 -0.03491088 -0.009680862
## x10         -0.01120951 -0.02057208  0.009646095
```

```
X.mat <- cbind(rep(1,times=nrow(salesdata)), x4, x5, x6, x7, x8, x9, x10)
Y.hat <- cbind(fitted(sales.mod.1), fitted(sales.mod.2), fitted(sales.mod.3))

resid.mat <- cbind(resid(sales.mod.1), resid(sales.mod.2), resid(sales.mod.3))

#### Testing whether the set (x8, x9, x10) is useless,
#### given the presence of x4, x5, x6, x7 as predictors in the model:

## Full model:

my.n <- length(x1)  # number of individuals
```

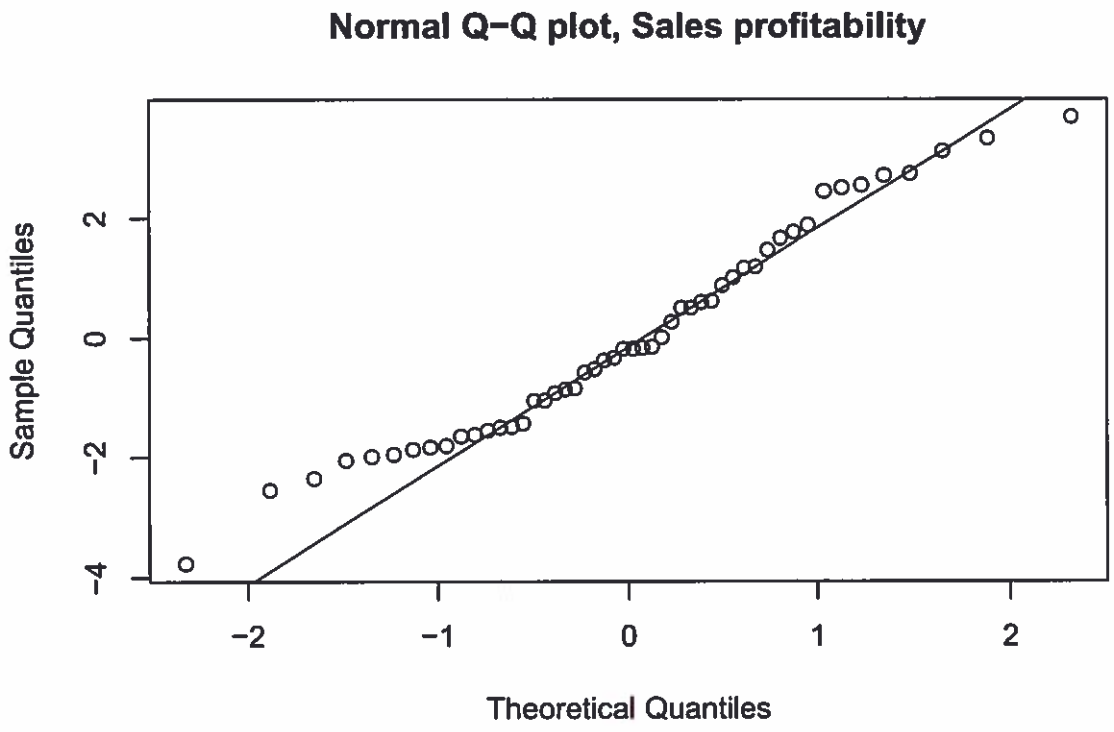
```
my.p <- ncol(X.mat) - 1  # total number of predictors
my.r <- ncol(resid.mat)  # total number of responses

Sigma.full <- (my.n-1)*var(resid.mat)/my.n

## Reduced model (without x8, x9, x10)

Beta.hat.redu <- cbind(coef(lm(x1 ~ x4 + x5 + x6 + x7) ),
                       coef(lm(x2 ~ x4 + x5 + x6 + x7) ), coef(lm(x3 ~ x4 + x5 + x6 + x7)) )

X.mat.redu <- cbind(rep(1,times=nrow(salesdata)), x4, x5, x6, x7)
my.p.redu <- ncol(X.mat.redu) - 1  # number of predictors in the reduced model

Y.hat.redu <- X.mat.redu %*% Beta.hat.redu


resid.mat.redu <- cbind (x1,x2,x3) - Y.hat.redu

Sigma.redu <- (my.n-1)*var(resid.mat.redu)/my.n

my.test.stat <-  -(my.n - my.p - 1 - 0.5*(my.r - my.p + my.p.redu + 1))*
            log( det(Sigma.full)/det(Sigma.redu) )
my.test.stat

## [1] 6.440941

p.value <- pchisq(my.test.stat, df = my.r*(my.p - my.p.redu), lower.tail=F )
p.value

## [1] 0.6951036

## new data
newdata <- data.frame(x4=10, x5=12,x6=9, x7=25, x8=15,x9=12,x10=40 )
# confidence interval
pred.mlm(sales.mod, newdata)

##           x1        x2        x3
## fit 94.02981 101.87483 100.55526
## lwr 91.75158  99.32427  98.75266
## upr 96.30804 104.42539 102.35786

# prediction interval
pred.mlm(sales.mod, newdata, interval="prediction")

##           x1        x2        x3
## fit  94.02981 101.87483 100.55526
## lwr  88.00355  95.12822  95.78712
## upr 100.05606 108.62143 105.32341

### Checking model assumptions:

qqnorm(resid.mat[,1], main = "Normal Q-Q plot, Sales growth")
qqline(resid.mat[,1])
```



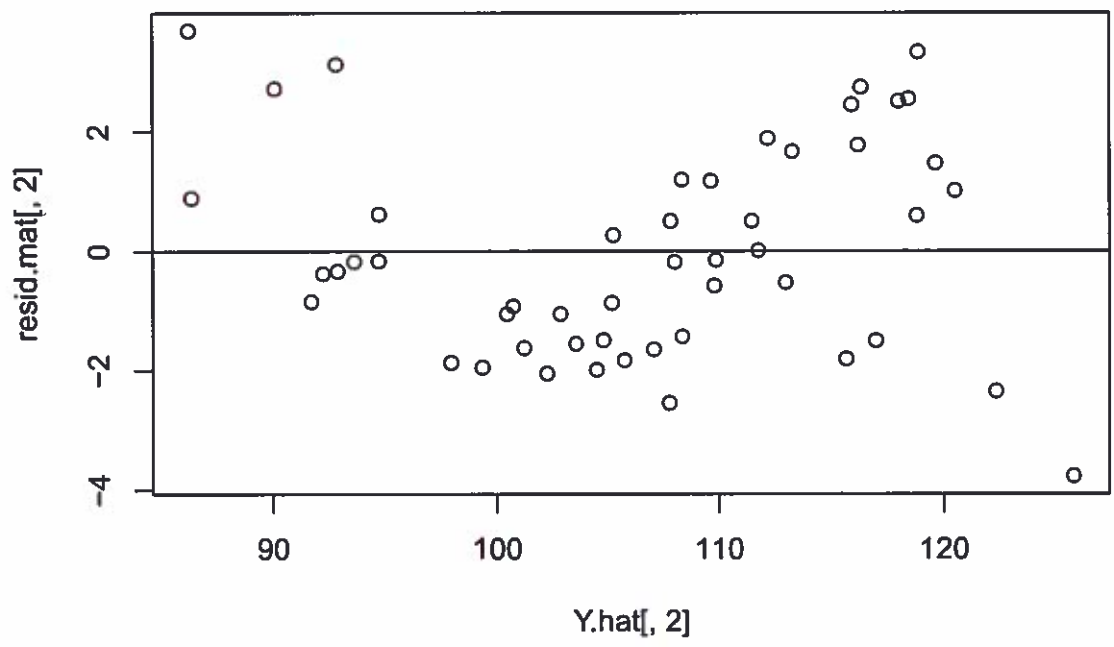

```
plot(Y.hat[,1],resid.mat[,1], main = "Residual plot vs. fitted values, Sales growth");
abline(h=0)
```



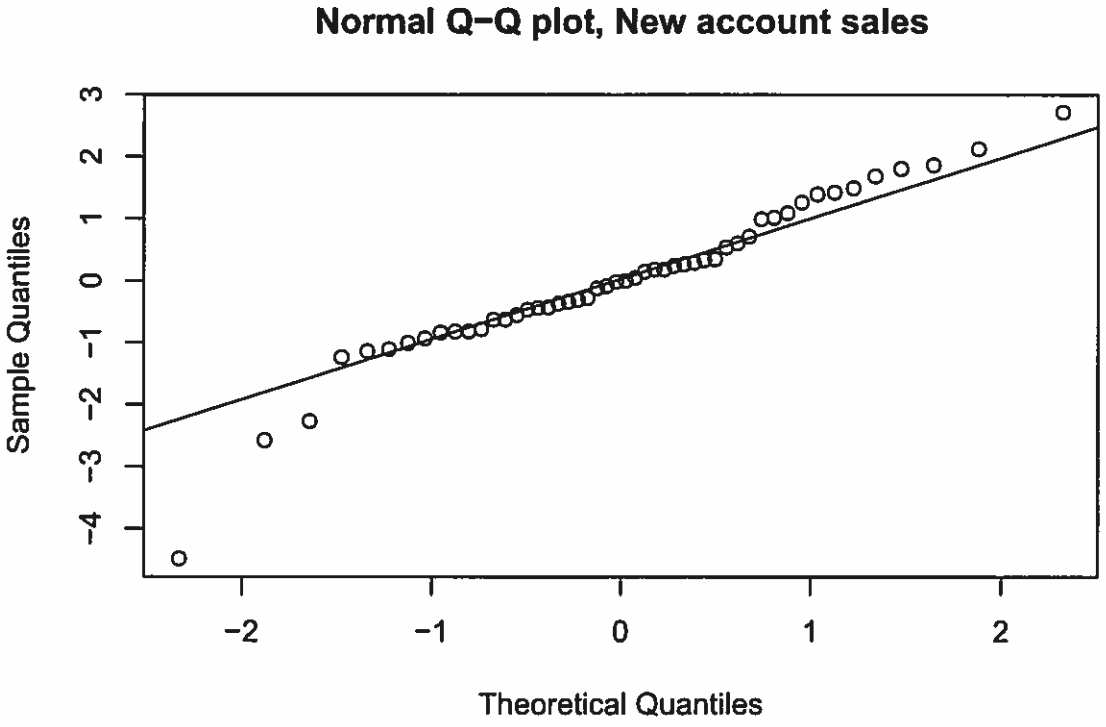

```
qqnorm(resid.mat[,2], main = "Normal Q-Q plot, Sales profitability")
qqline(resid.mat[,2])
```



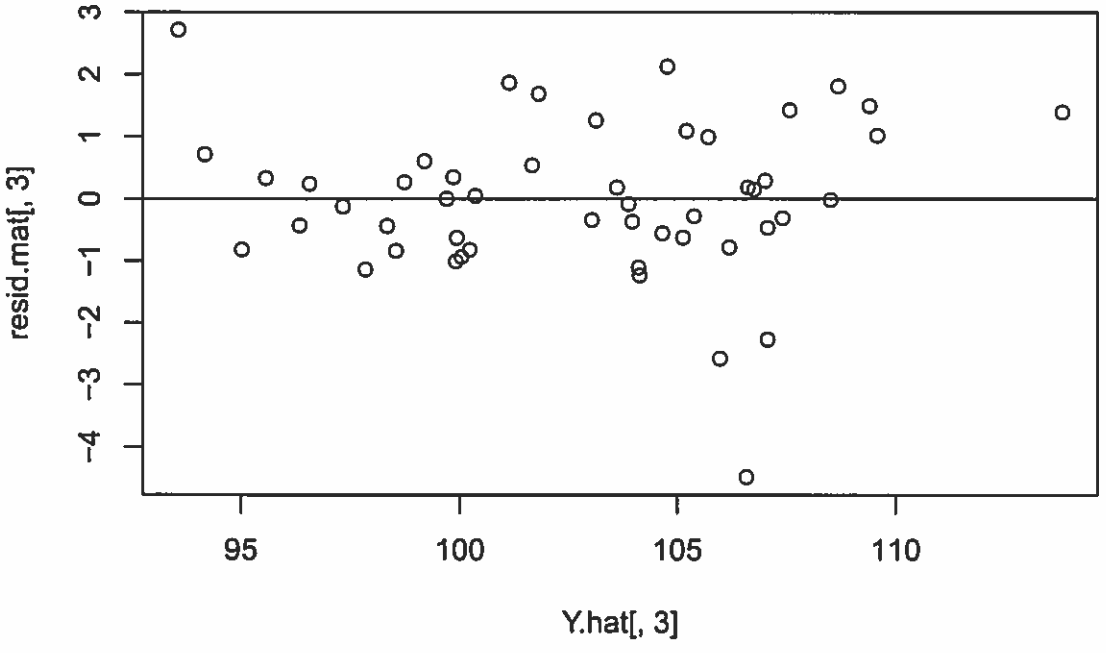

```
plot(Y.hat[,2],resid.mat[,2], main = "Residual plot vs. fitted values, Sales profitability");
abline(h=0)
```




```
qqnorm(resid.mat[,3], main = "Normal Q-Q plot, New account sales")
qqline(resid.mat[,3])
```




```
plot(Y.hat[,3],resid.mat[,3], main = "Residual plot vs. fitted values, New account sales");
abline(h=0)
```



Note: At first we have larger amount of random vectors.
$X_1, X_2 \cdots, X_q, \cdots X_p$ big $p > q$.

Chapter 4. Principal Component Analysis

Jianxuan Liu

Fall 2018

One of the aims in multivariate data analysis is to summarize the data in fewer than the original number of dimensions without losing essential information. More than a century ago, Pearson (1901) considered this problem, and Hotelling (1933) proposed a solution to it: instead of treating each variable separately, he considered combinations of the variables. Clearly, the average of all variables is such a combination, but many others exist. Two fundamental questions arise: 1. How should one choose these combinations? 2. How many such combinations should one choose?

There is no single strategy that always gives the right answer. This chapter will describe many ways of tackling at least the first problem.

$$\underset{\sim}{X} = \begin{bmatrix} X_{11} & X_{p1} \\ X_{1n} & X_{qn} \end{bmatrix}_{q \times n} \Rightarrow \begin{bmatrix} \\ \end{bmatrix} \Rightarrow Y = a^T X$$

1. Principal Components Analysis

The goal of principal components analysis (PCA) is to describe most of the variation in the multivariate data set $X_1, X_2, \cdots, X_q$ using a smaller or more concise set of variables. PCA can be thought of as a form of data reduction.

random variables

Data reduction is to use a lower-dimensional summarization of the data that still contains the relevant information that is in the full data. The hope is to be able to summarize what makes observations similar and what makes them different using relatively few indices. Also, we'd like each of these indices to say something distinct about how the observations vary.

In PCA, we create a new set of "variables" $\overline{Y_1, Y_2, \cdots, Y_q}$, each of which is a linear combination of $X_1, X_2, \cdots, X_q$,

choose the first m variables Y_i, $i = \overline{1,m}$ so that we can capture

$$\begin{cases} Y_1 = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1q}X_q \\ Y_2 = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2q}X_q \\ \vdots \\ Y_q = a_{q1}X_1 + a_{q2}X_2 + \cdots + a_{qq}X_q. \end{cases} = \begin{bmatrix} a_1^T \\ a_2^T \\ \\ a_q^T \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \\ X_q \end{bmatrix}$$

The principal components are those uncorrelated linear combinations $Y_1, Y_2, \cdots, Y_q$ whose variance are as large as possible.

We further require these new "variables" (or indices) Y's to be uncorrelated. This assures us that the information in Y_2, say, does not overlap with the information in Y_1. However, having q of these indices does not give us any data reduction. We would like to choose only the first m of these (where $m < q$) to focus on. Thus we choose the first linear combination, Y_1, so that it is the linear combination of $X_1, X_2, \cdots, X_q$ that accounts for as much of the variation in the original data as possible. Then the second linear combination, Y_2, is the linear combination that accounts for as much of the variation in the original data as possible (provided that it is uncorrelated with Y_1). And the third linear combination, Y_3, is the linear combination that accounts for as much of the variation in the original data as possible (provided that it is uncorrelated with Y_1 and Y_2), and so on. Eventually, it becomes a practical issue of how many of these indices (which are called principal components) we really need.

Idea: We first have big amount of variables (vectors).
- Then we choose the first q important vectors.
- Through those important q important vectors, $X_1, \cdots, X_q$, we construct $Y_1, \cdots, Y_q$ and then .

2. Mathematics Behind PCA

Singular value decomposition (SVD)

Suppose $\mathbf{X}$ is a $p \times n$ matrix with rank r. Then there exists an $p \times n$ matrix

$$\Gamma = \begin{bmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{bmatrix}$$

rank(A) = r

for which the diagonal enteries in $\mathbf{D}$ are the first r singular values of $\mathbf{X}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, and there exist an $p \times p$ orthogonal matrix U and an $n \times n$ orthogonal matrix V such that

$$\mathbf{X} = U\Gamma V^T. \qquad U^TU = I \quad V^TV = I$$

orthogonal orthogonal

The singular values of $\mathbf{X}$ are the square roots of the eigenvalues of $\mathbf{X}^T\mathbf{X}$, denoted by $\sigma_i, i = 1, \cdots p$ and they are arranged in decreasing order. That is, $\sigma_i = \sqrt{\lambda_i}$ for $1 \leq i \leq n$, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p \geq 0$.

- the columns of U are called **left singular vectors** of $\mathbf{X}$
- the columns of V are called **right singular vectors** $\mathbf{X}$

Example: Find a singular value decomposition of $\mathbf{X} = \begin{bmatrix} 1 & -1 \\ -2 & 2 \\ 2 & -2 \end{bmatrix}$.

Solution:

* $A = X^TX = \begin{bmatrix} 1 & -2 & 2 \\ -1 & 2 & -2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -2 & 2 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} 9 & -9 \\ -9 & 9 \end{bmatrix}$

* Find eigenvalue of A:

$\det(\lambda I - A) = \det\begin{pmatrix} 9-\lambda & -9 \\ -9 & 9-\lambda \end{pmatrix} = (9-\lambda)^2 - 8 = 0$

$\Rightarrow \lambda_1 = 18 \qquad \lambda_2 = 0$

$\Rightarrow$ corresponding unit eigenvectors

$\vec{v_1} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \qquad \vec{v_2} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

$\Rightarrow V = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

* Singular values are $\sigma_1 = \sqrt{18} \quad \sigma_2 = 0$

$\rightarrow \Gamma = \begin{bmatrix} \sqrt{18} & 0 \\ 0 & 0 \\ \hline 0 & 0 \end{bmatrix} \begin{matrix} D \\ 0 \end{matrix}$

* To construct U: $XV_1 = \begin{bmatrix} 2/\sqrt{2} \\ -4/\sqrt{2} \\ 4/\sqrt{2} \end{bmatrix} \qquad XV_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

orthogonal unit vector u_2, u_3

$u_1^TX = 0 \Rightarrow x_1 - 2x_2 + 2x_3 = 0 \quad (1)$

A basis for the solution of (1)

is $w_1 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \qquad w_2 = \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix}$

$\rightarrow u_1 = \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \\ 0 \end{bmatrix} \qquad u_2 = \begin{bmatrix} -2/3\sqrt{5} \\ 4/3\sqrt{5} \\ 5/3\sqrt{5} \end{bmatrix}$

Then

$X = \begin{bmatrix} 1 & -2 \\ -2 & 2 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} 1/3 & 2/\sqrt{5} & -2/3\sqrt{5} \\ -2/3 & 1/\sqrt{5} & 4/3\sqrt{5} \\ 2/3 & 0 & 5/3\sqrt{5} \end{bmatrix} \begin{bmatrix} \sqrt{18} & 0 \\ 0 & 0 \\ \hline 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

$Y = a^T X$

$L(e_1, \lambda_1) = e_1^T \Sigma e_1 + \lambda(1 - e_1^T e_1)$ | Choose e_1 to maximize $e_1^T \Sigma e_1$ s.t. $e_1^T e_1 = 1$.

Principal component analysis can capture the maximum sample variance

- First step: Choose coefficients $a_{11}, a_{12}, \cdots, a_{1q}$ such that the sample variance of

$$Y_1 = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1q}X_q$$

$a_1 = \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1q} \end{bmatrix}$

is as large as possible.

Of course, we could arbitrarily inflate this sample variance by making $a_{11}, a_{12}, \cdots, a_{1q}$ arbitrarily large. However, to make this maximization problem meaningful, we include the constraint that

$$\mathbf{a_1}^T \mathbf{a_1} = \sum_{i=1}^{q} a_{1i}^2 = 1$$

$\vec{a_{1i}}$: new dimension

- Second step: Choose coefficients $a_{21}, a_{22}, \cdots, a_{2q}$ such that the sample variance of

$$Y_2 = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2q}X_q$$

is as large as possible, subject to a $\mathbf{a_2}^T \mathbf{a_2} = 1$. We also want Y_2 to be uncorrelated with Y_1, so we add the further constraint that

$$\mathbf{a_1}^T \mathbf{a_2} = \sum_{i=1}^{q} a_{1i} a_{2i} = 0$$

- j^{th}step: Choose coefficients $a_{j1}, a_{j2}, \cdots, a_{jq}$ such that the sample variance of

$$Y_j = a_{j1}X_1 + a_{j2}X_2 + \cdots + a_{jq}X_q$$

is as large as possible, subject to $\mathbf{a_j}^T \mathbf{a_j} = 1$ and $\mathbf{a_j}^T \mathbf{a_{j'}} = 0$ for all $j' < j$

Mathematically, we can obtain q linear combinations $Y_1, Y_2, \cdots, Y_q$ in this way. This optimization problem is solved using the method of Lagrange multipliers. In practice, we use a software package like R or SAS to find the coefficients for us.

Formally, consider $\mathbf{X} \sim (\mu, \Sigma)$ (could be any continuous distribution), with $\Sigma = \mathbf{P}\Lambda\mathbf{P}^T$. Let r be the rank of Λ and $k = 1, \cdots, r$

positive definite.

$\vec{x} - \vec{\lambda}$: center of the instance

1. The k^{th} principal component score is the scalar

$\vec{a_k}$: new k^{th} dimension

$$W_k = \mathbf{a}_k^T(\mathbf{X} - \mu).$$

2. The k-dimensional principal component vector is the scalar

$$\mathbf{W}^{(k)} = [W_1, \cdots, W_k]^T = \mathbf{P}_k^T(\mathbf{X} - \mu).$$

$= \left[a_k^T(X-\lambda)\right]^T \left[a_k(X-\lambda)\right]$

$\vec{a_k}$ load weight

3. The k^{th} principal component projection (vector) is the scalar

$$\mathbf{P}_k = \mathbf{a}_k \mathbf{a}_k^T(\mathbf{X} - \mu).$$

We use the letter W for the principal component score, indicating that they are weighted random variables. The k^{th} principal component score, W_k represents the contribution of $\mathbf{X}$ in the direction $\mathbf{a}_k$. The vector $\mathbf{a}_k$ are sometimes called the loadings as they represent the "load" or "weight" each variable is accorded in the projection.

Mathematically, W_k is obtained by projecting the centered $\mathbf{X}$ onto $\mathbf{a}_k$. Collecting the contributions of the first k scores into one object leads to the principal component vector $\mathbf{W}^{(k)}$, a k-dimensional principal component vector which summarizes the contributions of $\mathbf{X}$ along the first k eigen directions, the eigenvectors of Σ.

The principal component projection (vector) $\mathbf{P}_k$ is a q-dimensional random vector which points in the same or opposite direction as the k^{th} eigenvector $\mathbf{a}_k$, and the length or Euclidean norm of $\mathbf{P}_k$ is the absolute value of W_k.

Example

Let $\mathbf{X} = [X_1, X_2]^T$ be a two-dimensional random vector with mean μ and covariance matrix Σ given by

$$\mu = [0, -1]^T, \text{and } \Sigma = \begin{pmatrix} 2.4 & -0.5 \\ -0.5 & 1 \end{pmatrix}.$$

The eigenvalues and eigenvectors of Σ are

$$(\lambda_1, \mathbf{a}_1) = \left(2.5602, \begin{bmatrix} -0.9523 \\ 0.3052 \end{bmatrix}\right) \text{and } (\lambda_2, \mathbf{a}_2) = \left(0.8398, \begin{bmatrix} -0.3052 \\ -0.9523 \end{bmatrix}\right)$$

```
Sigma=matrix(c(2.4 ,-0.5 ,-0.5 , 1),nrow=2)
Sigma
##      [,1] [,2]
## [1,]  2.4 -0.5
## [2,] -0.5  1.0
eigen(Sigma,symmetric=T)

## eigen() decomposition
## $values
## [1] 2.5602325 0.8397675
##
## $vectors
##            [,1]       [,2]
## [1,] -0.9522955 -0.3051774
## [2,]  0.3051774 -0.9522955
```

The proportion of total variance accounted by the first PC, W1 is

$$\frac{\lambda_1}{\lambda_1+\lambda_2} = \frac{2.560}{2.560 + 0.839} =$$

$\lambda_1 + \lambda_2 = 3.4$
$\text{Var}(X_1) + \text{Var}(X_2) = 3.4$ → equal

$W_1 = a_1^T(\underline{X} - \mu) = [-0.952 \quad 0.3052] \begin{bmatrix} X_1 - 0 \\ X_2 + 1 \end{bmatrix}$

The eigenvectors show the axes along which the data vary most, with the first vector $\sqrt{\lambda_1}\mathbf{a}_1$ pointing along the direction in which the data have the largest variance. The first eigenvalue is considerably bigger than the second, so much more variability exists along this direction. The two principal component scores are

$$W_1 = -0.9523X_1 + 0.3052(X_2 + 1) \text{and } W_2 = -0.3052X_1 - 0.9523(X_2 + 1).$$

$\text{Var}(W_1) > \text{Var}(W_2)$ →
The first PC score is heavily weighted in the direction of the first variable, implying that the first variable contributes more strongly than the second to the variance of **x**.

$\text{Var}(W_1) = 0.9523^2 \underbrace{V(X_1)}_{2.4} + 0.3052^2 \underbrace{V(X_2)}_{1}$

$= 1.13$

$\text{Var}(W_2) =$

$= 1.00$

PCA and SVD

PCA is an application of the SVD. The principal components are equal to the right singular vectors if we first scale the data by subtracting the column mean and dividing each column by its standard deviation (`scale()` in R).

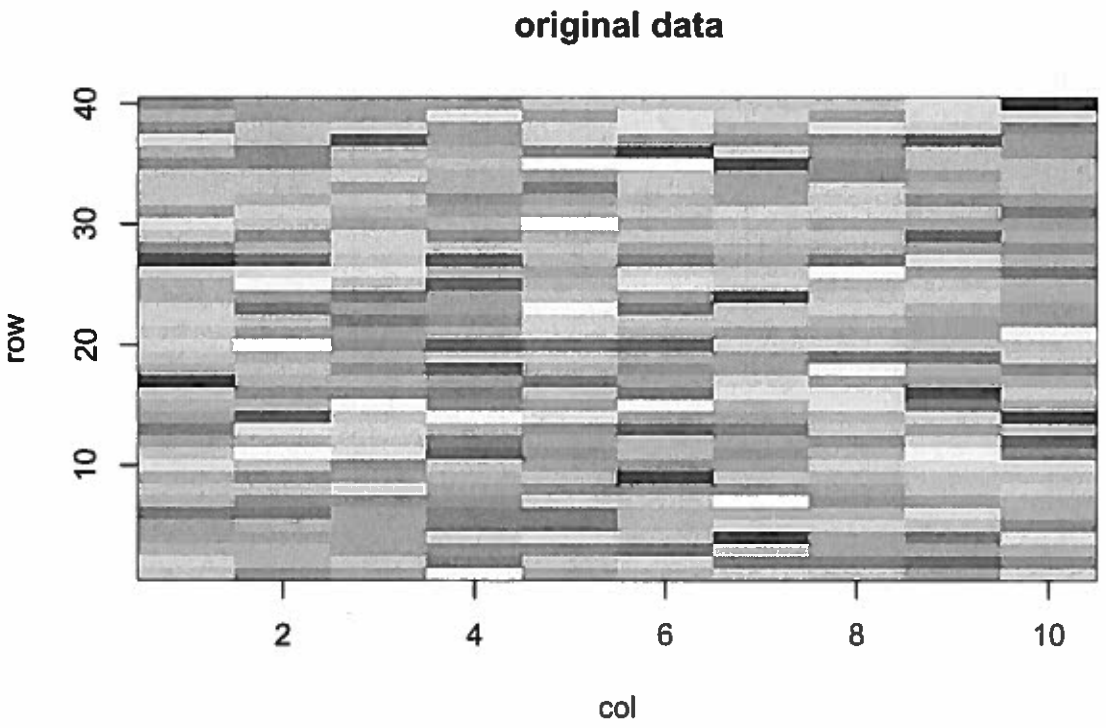
Below we plot the first left and right singular vectors along with the original data.

```
library(dplyr)
```
← working on data frame

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

40×10

```
set.seed(1)
DX=matrix(rnorm(400), nrow = 40)
image(1:10, 1:40, t(DX)[, nrow(DX):1],ylab="row",xlab="col",main="original data")
```




```
hh=dist(DX) %>% hclust
DXOrdered=DX[hh$order, ]
par(mfrow = c(1, 3))
## Complete data
image(t(DXOrdered)[, nrow(DXOrdered):1],main="Original Data")
## Show the row means
plot(rowMeans(DXOrdered), 40:1, , xlab = "Row Mean", ylab = "Row", pch = 19)
## Show the column means
plot(colMeans(DXOrdered), xlab = "Column", ylab = "Column Mean", pch = 19)
```

* Connection between SVD and PCA

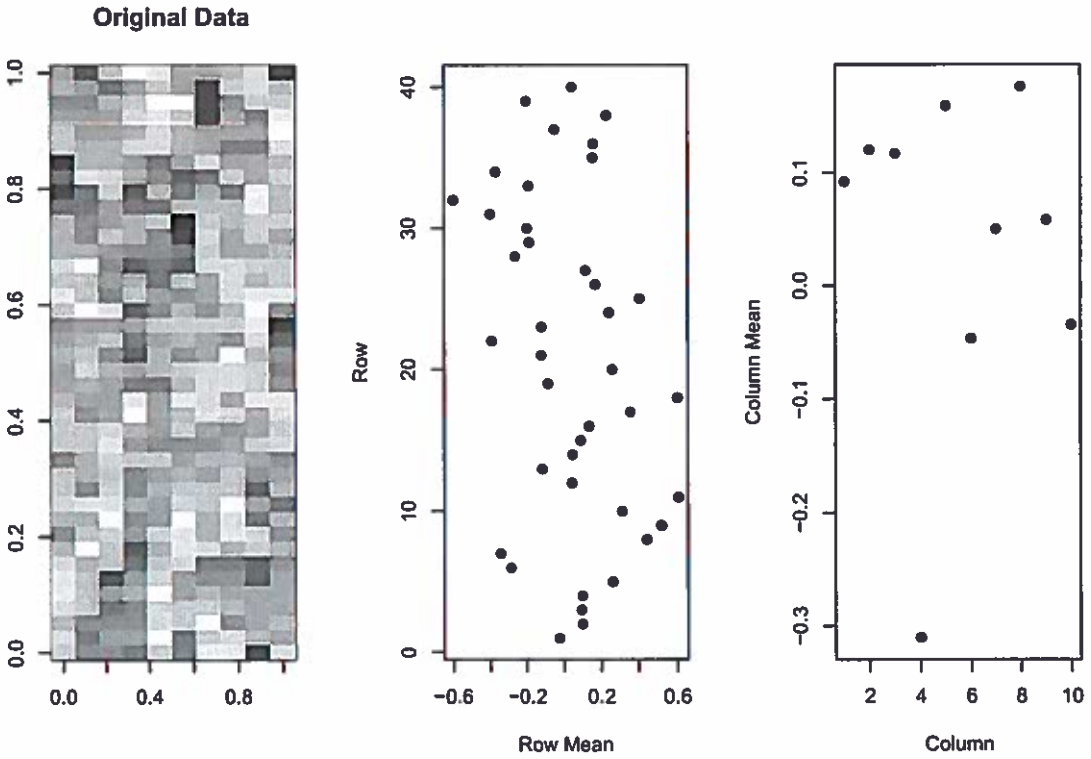
- In SVD $X = U \Sigma V^T$ Γ: contains singular value of X
- In PCA $X \sim (\underset{\sim}{\mu}, \Sigma)$ $\Sigma = P \Lambda P^T$

$$Var(X) = X^T X = (V \Gamma U^T)(U^T \Gamma V^T) = V \Gamma^2 V^T \propto \Sigma \propto P \Lambda P^T$$

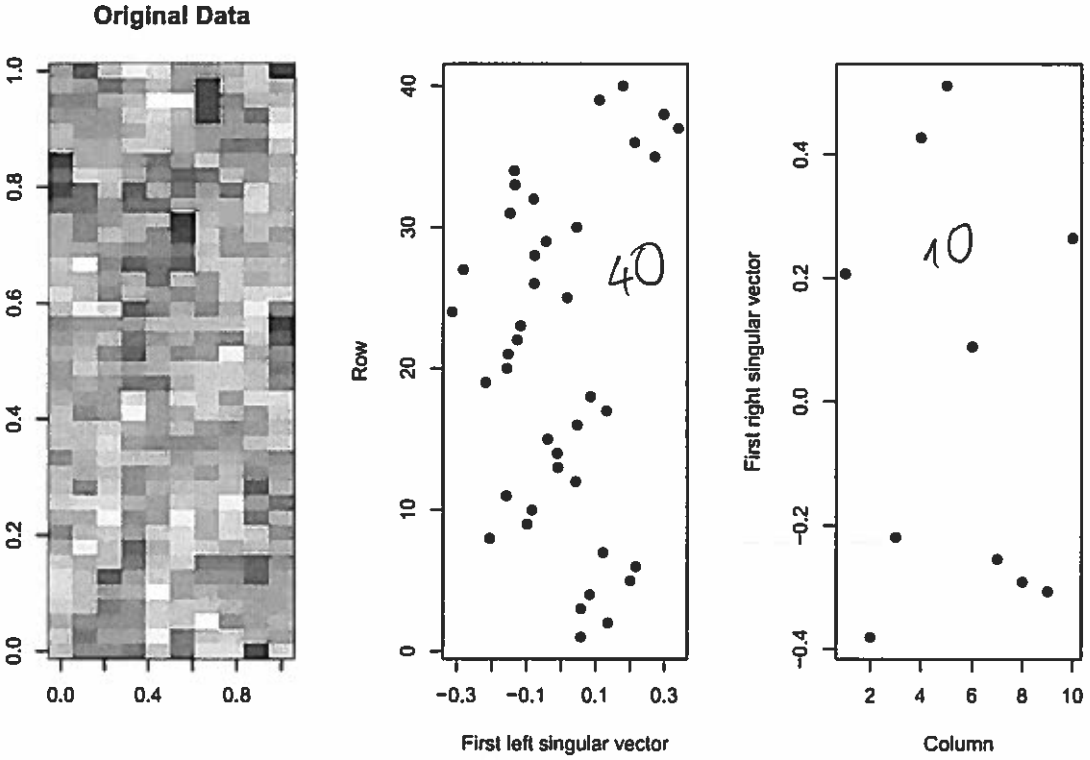
Γ^2: eigenvalue of $X^T X$

$\Rightarrow \Gamma = \Lambda^{1/2}$

$\Rightarrow P = a\, a^T (X - \underset{\sim}{\mu}) = V$




```
svd1 = svd(scale(DXOrdered))
par(mfrow = c(1, 3))
image(t(DXOrdered)[, nrow(DXOrdered):1], main = "Original Data")
plot(svd1$u[, 1], 40:1, , ylab = "Row", xlab = "First left singular vector",
     pch = 19)
plot(svd1$v[, 1], xlab = "Column", ylab = "First right singular vector", pch = 19)
```



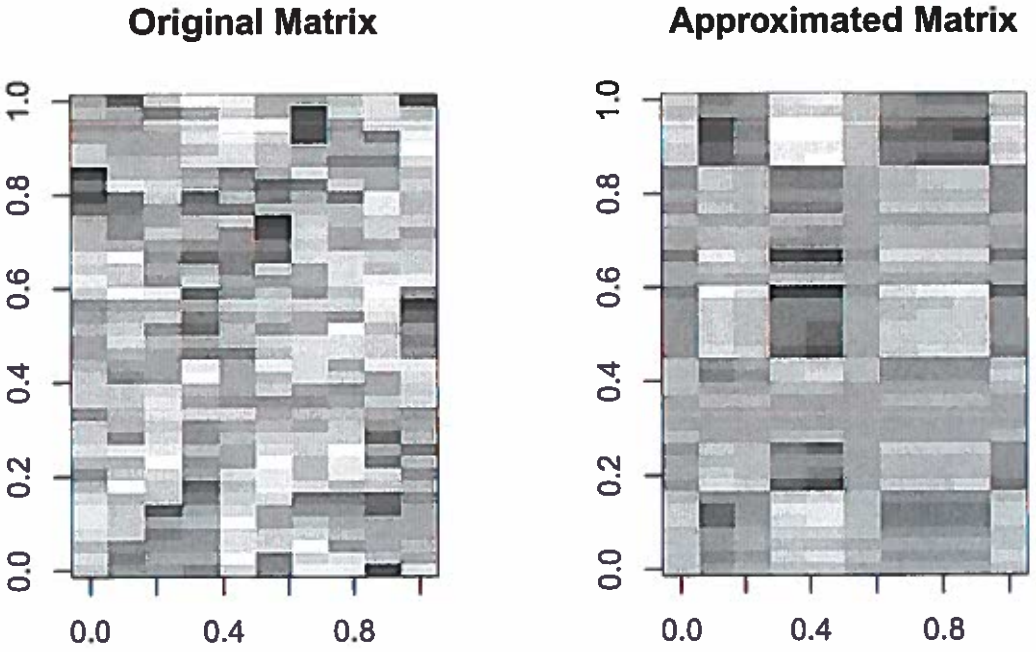
We can see how the first left and right singular vectors pick up the mean shift in both the rows and columns of the matrix.

If we believed that the first left and right singular vectors, call them u_1 and v_1, captured all of the variation in the data, then we could approximate the original data matrix with

$$X \approx u_1 v_1^T.$$

Thus, we would reduce 400 numbers in the original matrix to $40 + 10 = 50$ numbers in the compressed matrix, a nearly 90% reduction in information. Here's what the original data and the approximation would look like.

```
## Approximate original data with outer product of first singular vectors
approx =with(svd1, outer(u[, 1], v[, 1]))
## Plot original data and approximated data
par(mfrow = c(1, 2))
image(t(DXOrdered)[, nrow(DXOrdered):1], main = "Original Matrix")
image(t(approx)[, nrow(approx):1], main = "Approximated Matrix")
```



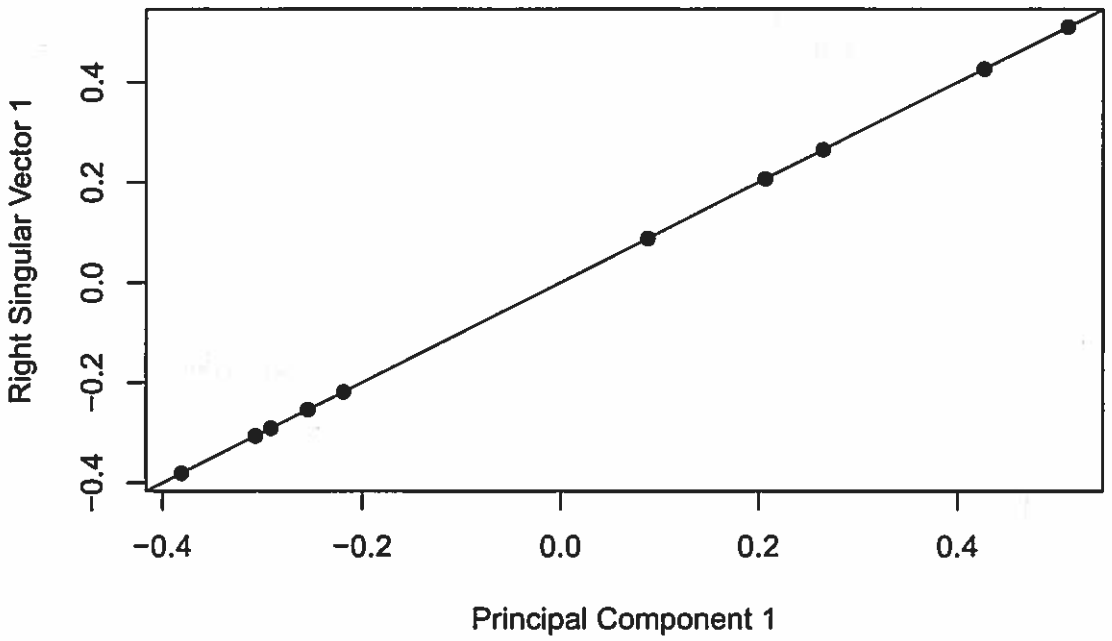
Obviously, the two matrices are not identical, but the approximation seems reasonable in this case. This is not surprising given that there was only one real feature in the original data.

As we mentioned above, the SVD has a close connection to principal components analysis (PCA). Here, we show that the first right singular vector from the SVD is equal to the first principal component vector returned by PCA.

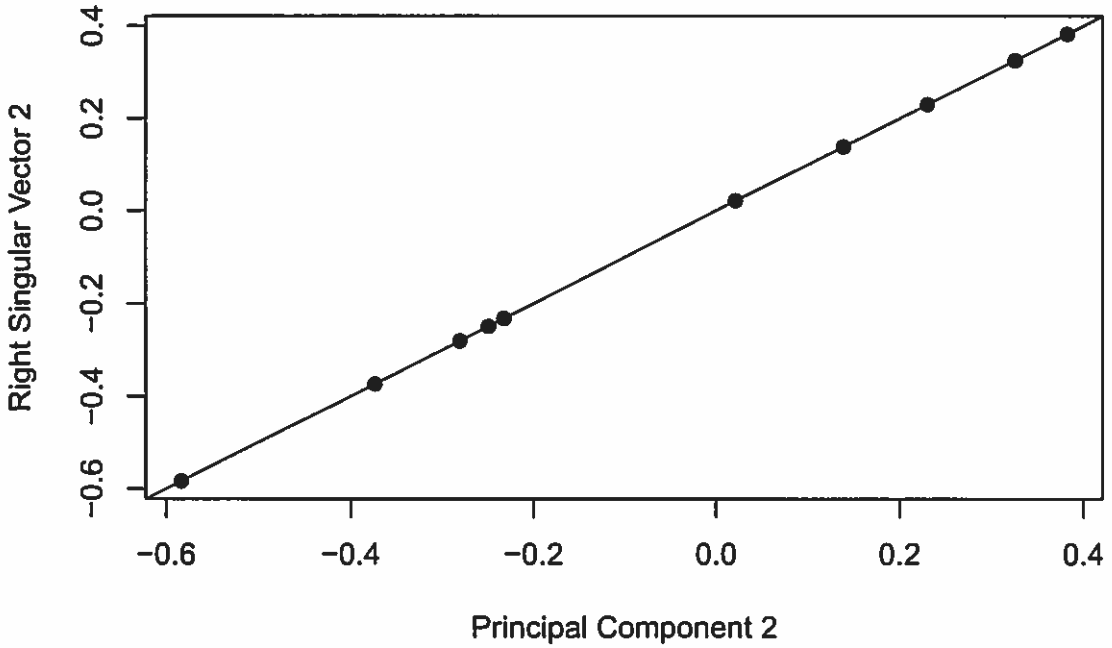
have to scale

```
svd1=svd(scale(DXOrdered))
pca1=prcomp(DXOrdered, scale = TRUE)
plot(pca1$rotation[, 1], svd1$v[, 1], pch = 19,
     xlab = "Principal Component 1", ylab = "Right Singular Vector 1")
abline(c(0, 1))
```

+ summary(pca1) → help us know the proportion of variance capture by PCA 1.




```
plot(pca1$rotation[, 2], svd1$v[, 2], pch = 19,
     xlab = "Principal Component 2",  ylab = "Right Singular Vector 2")
abline(c(0, 1))
```



$a_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \\ \vdots \\ a_{iq} \end{bmatrix}$ $Y_1, Y_2, \cdots, Y_p$ are uncorrelated

Results

1. Let Σ be the covariace matrix associated with the random vector $\mathbf{X}^T = [X_1, X_2, \cdots, X_q]$. Let Σ have the engenvalue-eigenvector paris $(\lambda_1, \mathbf{a}_1), (\lambda_2, \mathbf{a}_2), \cdots, (\lambda_q, \mathbf{a}_q)$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \geq 0$. Then the i^{th} principal component is given by

$$Y_i = a_{i1}X_1 + a_{i2}X_2 + \cdots + a_{iq}X_q, i = 1, 2, \cdots, q.$$

With these choices,

$$E(Y_i) = \mathbf{a}_i^T \mu = \sum_{j=1}^{q} a_{ij}\mu_j, i = 1, 2, \cdots, q$$

$$Var(Y_i) = \mathbf{a}_i{}^T \Sigma \mathbf{a}_i = \lambda_i, i = 1, 2, \cdots, q \quad \text{same indices}$$

$$Cov(Y_i, Y_k) = \mathbf{a}_i{}^T \Sigma \mathbf{a}_k = \lambda_i, i \neq k. \quad \neq \text{indices}.$$

If some λ_i are equal, then choces of the corresponding coefficient vectors, $\mathbf{a}_i$, and hence Y_i are not unique.

2. Let Σ be the covariace matrix associated with the random vector $\mathbf{X}^T = [X_1, X_2, \cdots, X_q]$. Let Σ have the engenvalue-eigenvector paris $(\lambda_1, \mathbf{a}_1), (\lambda_2, \mathbf{a}_2), \cdots, (\lambda_q, \mathbf{a}_q)$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \geq 0$. Let $Y_1 = \mathbf{a}_1^T\mathbf{X}, Y_2 = \mathbf{a}_2^T\mathbf{X}, \cdots, Y_q = \mathbf{a}_q^T\mathbf{X}$ be the principal components. Then

$$\sigma_{11} + \sigma_{22} + \cdots + \sigma_{qq} = \sum_{i=1}^{q} Var(X_i) = \lambda_1 + \lambda_2 + \cdots + \lambda_q = \sum_{i=1}^{q} Var(Y_i)$$

The trace of a covariance matrix is equal to the sum of its eigenvalues

$$trace(\Sigma) = \sum_{i=1}^{q} \lambda_i$$

3. If $Y_1 = \mathbf{a}_1^T\mathbf{X}, Y_2 = \mathbf{a}_2^T\mathbf{X}, \cdots, Y_q = \mathbf{a}_q^T\mathbf{X}$ are the principal components obtained from the covariance matrix Σ, then

$$\rho_{Y_i, X_k} = \frac{a_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}, i, k = 1, 2, \cdots, q$$

are the correlation coefficients between the components Y_i and the variables X_k. Here $(\lambda_1, \mathbf{a}_1), (\lambda_2, \mathbf{a}_2), \cdots, (\lambda_q, \mathbf{a}_q)$ are the engenvalue-eigenvector paris for Σ.

Choosing the Number of Components

To explain all the variation in the original data, we would need (in general) all q principal components. But it is practically sufficient to explain most of the variation in the original data. This can usually be done using merely the "first few" principal components. If we will retain the first $m < q$ components, how can we choose m?

Possible Rules

1. Retain the first m components sufficient to explain a specified percentage (70%? 80%? 90%?) of the total variance of the original variables.
2. Keep only the components whose eigenvalues are at least $\sum_{i=1}^{q} \frac{\lambda_i}{q}$, which is the average eigenvalue and also the average sample variance of the original variables.
3. When PCA is done on the correlation matrix, this average is 1, so Kaiser (1958) suggested keeping components with eigenvalues at least 1. Jolliffe (1972) preferred using 0.7 as the threshold.
4. Cattell (1965) introduced the scree diagram, which plots λ_i against i, for $i = 1, \cdots, q$. Look for the "elbow" in the curve and choose the corresponding number of components.
5. Jolliffe (1986) suggested a modified scree plot that plotted $\log(\lambda_i)$ against i, for $i = 1, \cdots, q$.

3 An illustration with the Sydney heptathlon data

Before doing anything else with these data, it needs to be noted that in the three running events, better performance is indicated by a lower measure (time), whereas in the jumping and throwing events good performance is indicated by a higher measure (distance). It seems sensible to readjust data so that large values are good and small values bad (Timed racing events are obviously the opposite).

```
library(HSAUR)
```

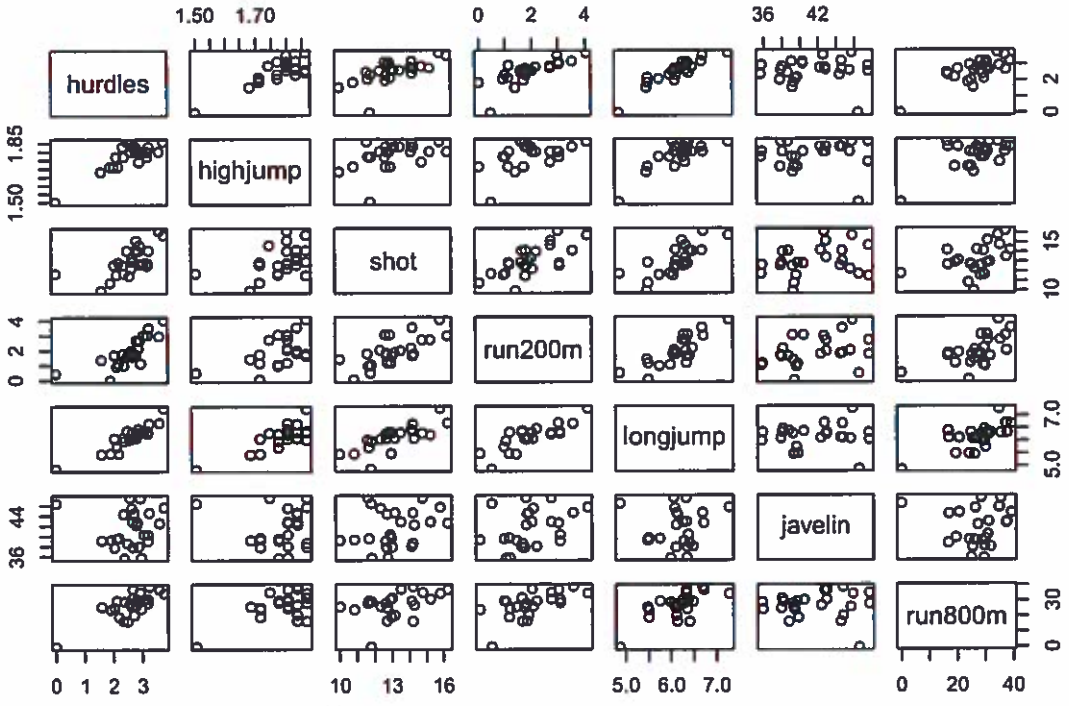
```
## Loading required package: tools
```

```
data("heptathlon", package = "HSAUR")
head(heptathlon)
```

```
##                     hurdles highjump  shot run200m longjump javelin
## Joyner-Kersee (USA)   12.69     1.86 15.80   22.56     7.27   45.66
## John (GDR)            12.85     1.80 16.23   23.65     6.71   42.56
## Behmer (GDR)          13.20     1.83 14.20   23.10     6.68   44.54
## Sablovskaite (URS)    13.61     1.80 15.23   23.92     6.25   42.78
## Choubenkova (URS)     13.51     1.74 14.76   23.93     6.32   47.46
## Schulz (GDR)          13.75     1.83 13.50   24.65     6.33   42.82
##                     run800m score
## Joyner-Kersee (USA)  128.51  7291
## John (GDR)           126.12  6897
## Behmer (GDR)         124.20  6858
## Sablovskaite (URS)   132.24  6540
## Choubenkova (URS)    127.90  6540
## Schulz (GDR)         125.79  6411
```

This example has a score column

```
# Readjust data so that large values are good and small values bad
# (Timed racing events are obviously the opposite)
heptathlon$hurdles <- max(heptathlon$hurdles) - heptathlon$hurdles
heptathlon$run200m <- max(heptathlon$run200m) - heptathlon$run200m
heptathlon$run800m <- max(heptathlon$run800m) - heptathlon$run800m

plot(heptathlon[, -which(names(heptathlon) %in% "score")])
```




```
# Check out the correlation
round(cor(heptathlon[, -which(names(heptathlon) %in% "score")]), digits = 3)
```

```
##          hurdles highjump  shot run200m longjump javelin run800m
## hurdles    1.000    0.811 0.651   0.774    0.912   0.008   0.779
```

```
## highjump   0.811    1.000 0.441   0.488    0.782    0.002    0.591
## shot       0.651    0.441 1.000   0.683    0.743    0.269    0.420
## run200m    0.774    0.488 0.683   1.000    0.817    0.333    0.617
## longjump   0.912    0.782 0.743   0.817    1.000    0.067    0.700
## javelin    0.008    0.002 0.269   0.333    0.067    1.000   -0.020
## run800m    0.779    0.591 0.420   0.617    0.700   -0.020    1.000
```

```
summary(heptathlon[, -which(names(heptathlon) %in% "score")])
```

```
##     hurdles         highjump          shot          run200m
##  Min.   :0.00   Min.   :1.500   Min.   :10.00   Min.   :0.000
##  1st Qu.:2.35   1st Qu.:1.770   1st Qu.:12.32   1st Qu.:1.380
##  Median :2.67   Median :1.800   Median :12.88   Median :1.780
##  Mean   :2.58   Mean   :1.782   Mean   :13.12   Mean   :1.961
##  3rd Qu.:2.95   3rd Qu.:1.830   3rd Qu.:14.20   3rd Qu.:2.690
##  Max.   :3.73   Max.   :1.860   Max.   :16.23   Max.   :4.050
##     longjump        javelin         run800m
##  Min.   :4.880   Min.   :35.68   Min.   : 0.00
##  1st Qu.:6.050   1st Qu.:39.06   1st Qu.:24.95
##  Median :6.250   Median :40.28   Median :28.69
##  Mean   :6.152   Mean   :41.48   Mean   :27.38
##  3rd Qu.:6.370   3rd Qu.:44.54   3rd Qu.:31.19
##  Max.   :7.270   Max.   :47.50   Max.   :39.23
```

```
# Scaling (using cor) is done with scale.
hep.pr <- prcomp(heptathlon[, -which(names(heptathlon) %in% "score")], scale. = TRUE)
```

```
attributes(hep.pr)
```

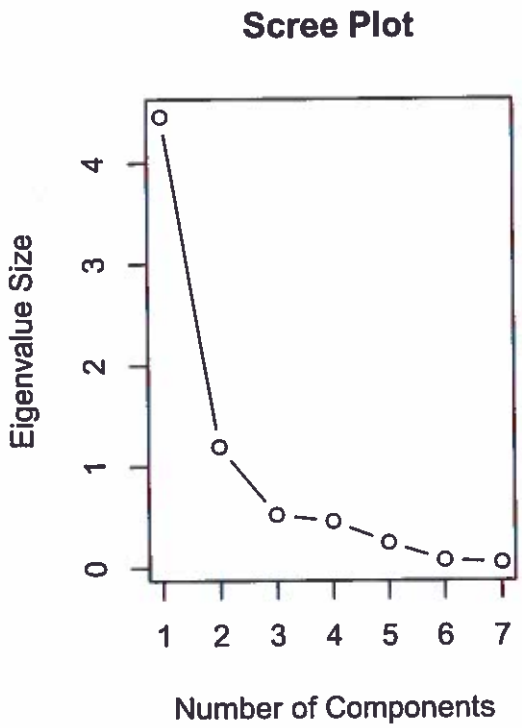
```
## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
##
## $class
## [1] "prcomp"
```

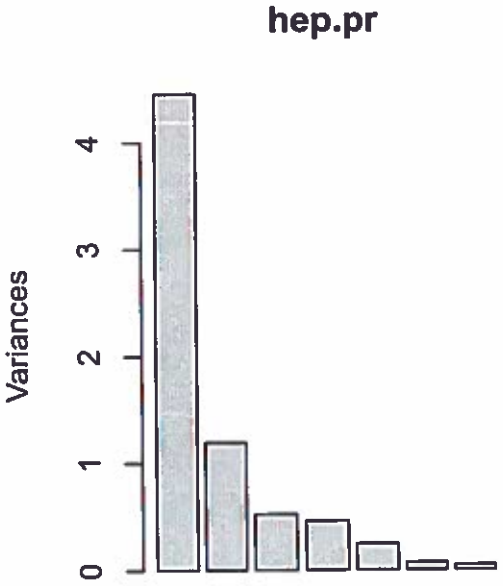
```
summary(hep.pr)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.1119 1.0928 0.72181 0.67614 0.49524 0.27010
## Proportion of Variance 0.6372 0.1706 0.07443 0.06531 0.03504 0.01042
## Cumulative Proportion  0.6372 0.8078 0.88223 0.94754 0.98258 0.99300
##                           PC7
## Standard deviation     0.2214
## Proportion of Variance 0.0070
## Cumulative Proportion  1.0000
```

Note that 2 PC's have $\lambda > 1$ (or 0.7) first 4 PC's have total variance greater than 90%. Let us look at the scree plot.

```
par(mfrow = c(1, 2))
plot(1:length(hep.pr$sdev), (hep.pr$sdev)^2, type='b',
main="Scree Plot", xlab="Number of Components", ylab="Eigenvalue Size")
# OR use
screeplot(hep.pr)
```






```
# Mean and sd dev
hep.pr$center
```

```
##  hurdles highjump     shot  run200m longjump  javelin  run800m
##   2.5800   1.7820  13.1176   1.9608   6.1524  41.4824  27.3760
```

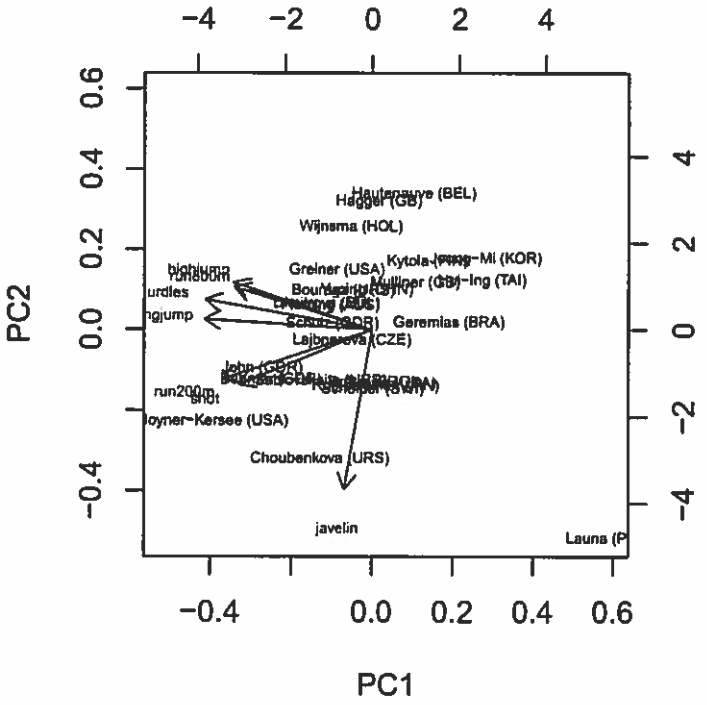
```
hep.pr$scale
```

```
##    hurdles   highjump       shot    run200m   longjump    javelin
## 0.73664781 0.07794229 1.49188438 0.96955712 0.47421233 3.54565612
##    run800m
## 8.29108809
```

```
# Factor loadings
hep.pr$rotation
```

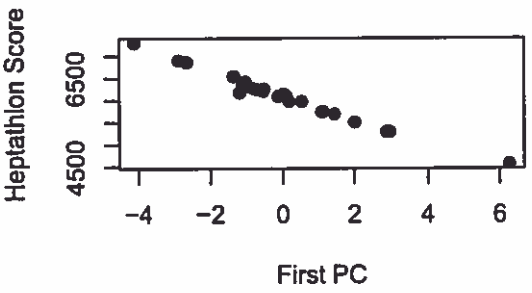
```
##                 PC1         PC2         PC3         PC4         PC5
## hurdles  -0.4528710  0.15792058 -0.04514996  0.02653873 -0.09494792
## highjump -0.3771992  0.24807386 -0.36777902  0.67999172  0.01879888
## shot     -0.3630725 -0.28940743  0.67618919  0.12431725  0.51165201
## run200m  -0.4078950 -0.26038545  0.08359211 -0.36106580 -0.64983404
## longjump -0.4562318  0.05587394  0.13931653  0.11129249 -0.18429810
## javelin  -0.0754090 -0.84169212 -0.47156016  0.12079924  0.13510669
## run800m  -0.3749594  0.22448984 -0.39585671 -0.60341130  0.50432116
##                 PC6         PC7
## hurdles  -0.78334101  0.38024707
## highjump  0.09939981 -0.43393114
## shot     -0.05085983 -0.21762491
## run200m   0.02495639 -0.45338483
## longjump  0.59020972  0.61206388
## javelin  -0.02724076  0.17294667
```

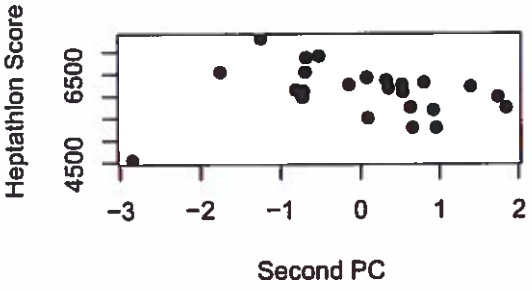
```
## run800m   0.15555520 -0.09830963
# Note that the scores with prcomp are 'x', so hep.pr$x
# Correlation of pc's to data
round(cor(heptathlon[, -which(names(heptathlon) %in% "score")], hep.pr$x), digits = 4)

##              PC1     PC2     PC3     PC4     PC5     PC6     PC7
## hurdles  -0.9564  0.1726 -0.0326  0.0179 -0.0470 -0.2116  0.0842
## highjump -0.7966  0.2711 -0.2655  0.4598  0.0093  0.0268 -0.0961
## shot     -0.7668 -0.3163  0.4881  0.0841  0.2534 -0.0137 -0.0482
## run200m  -0.8614 -0.2846  0.0603 -0.2441 -0.3218  0.0067 -0.1004
## longjump -0.9635  0.0611  0.1006  0.0752 -0.0913  0.1594  0.1355
## javelin  -0.1593 -0.9198 -0.3404  0.0817  0.0669 -0.0074  0.0383
## run800m  -0.7919  0.2453 -0.2857 -0.4080  0.2498  0.0420 -0.0218
par(mfrow = c(1, 1))
biplot(hep.pr, cex = 0.5)
```

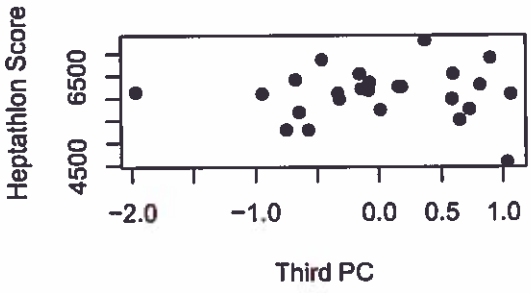


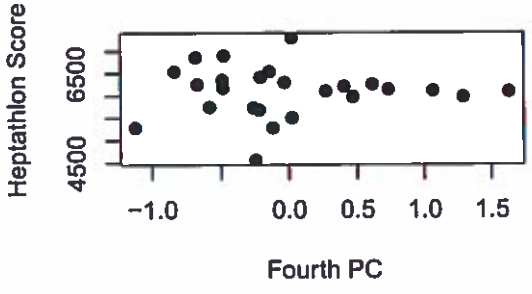
Compare the first PC to the overall score. Note that the first is highly correlated with the score with rapidly declining association

```
par(mfrow = c(2, 2))
plot(hep.pr$x[,1], heptathlon[, "score"], pch = 19, xlab = "First PC", ylab = "Heptathlon Score")
plot(hep.pr$x[,2], heptathlon[, "score"], pch = 19, xlab = "Second PC", ylab = "Heptathlon Score")
plot(hep.pr$x[,3], heptathlon[, "score"], pch = 19, xlab = "Third PC", ylab = "Heptathlon Score")
plot(hep.pr$x[,4], heptathlon[, "score"], pch = 19, xlab = "Fourth PC", ylab = "Heptathlon Score")
```










```
cor(hep.pr$x, heptathlon$score)
```

```
##              [,1]
## PC1 -0.991097775
## PC2 -0.097885776
## PC3 -0.005162860
## PC4  0.005161233
## PC5  0.045780044
## PC6  0.030648671
## PC7  0.006510970
```

Example: Bumpus bird data

The data contains the body measurements of female sparrows:

- X1=total length
- X2=alar length
- X3=length of beak and head
- X4=length of humerus
- X5=length of keel and sternum; all in mm)

Birds 1 to 21 survived a severe storm near Brown University in Rhode Island while the remainder died. (Original source Bumpus 1898.)

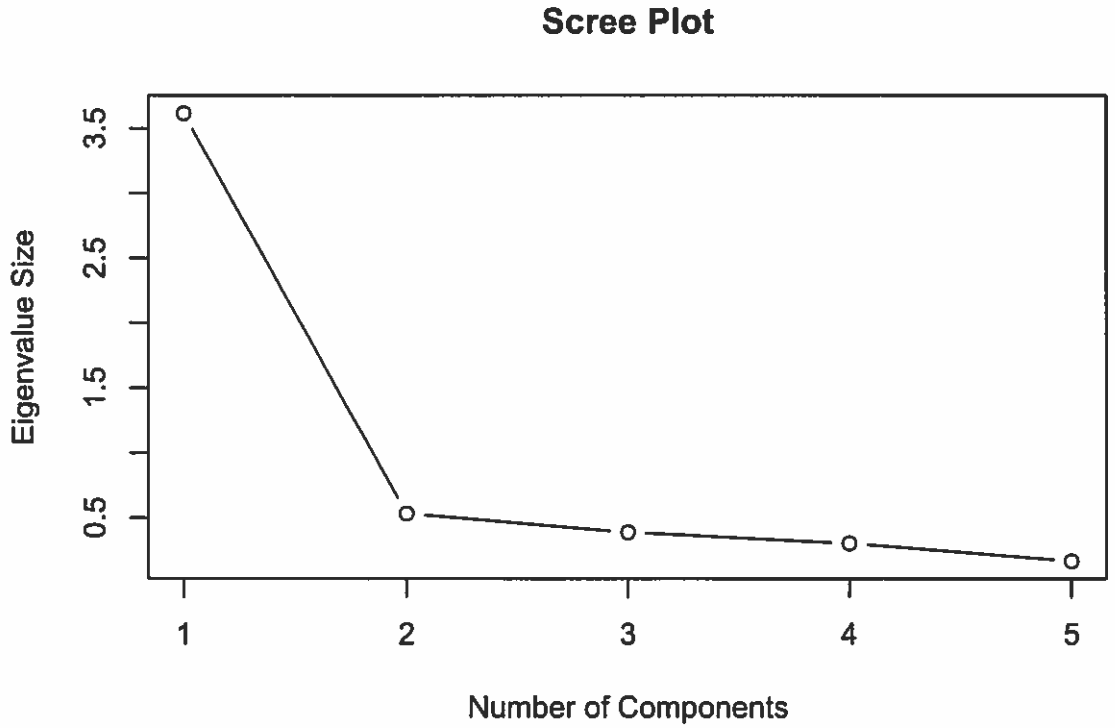
```
bumpbird=read.table("bumpusbird.txt", header=T) #"bumpusbird.txt"
names(bumpbird)
```

```
## [1] "ID" "X1" "X2" "X3" "X4" "X5"
```

```
names(bumpbird) = c("ID", "tot.length", "alar.length",
                    "beak.head.length", "humerus.length",
                    "keel.stern.length")
attach(bumpbird)
bird.pc=princomp(bumpbird[,-1], cor=T)
summary(bird.pc,loadings=T)

## Importance of components:
##                           Comp.1    Comp.2     Comp.3    Comp.4    Comp.5
## Standard deviation     1.9015726 0.7290433 0.62163056 0.5491498 0.4056199
## Proportion of Variance 0.7231957 0.1063008 0.07728491 0.0603131 0.0329055
## Cumulative Proportion  0.7231957 0.8294965 0.90678139 0.9670945 1.0000000
##
## Loadings:
##                   Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## tot.length        -0.452         0.690  0.420 -0.374
## alar.length       -0.462 -0.300  0.341 -0.548  0.530
## beak.head.length  -0.451 -0.325 -0.454  0.606  0.343
## humerus.length    -0.471 -0.185 -0.411 -0.388 -0.652
## keel.stern.length -0.398  0.876 -0.178         0.192
```

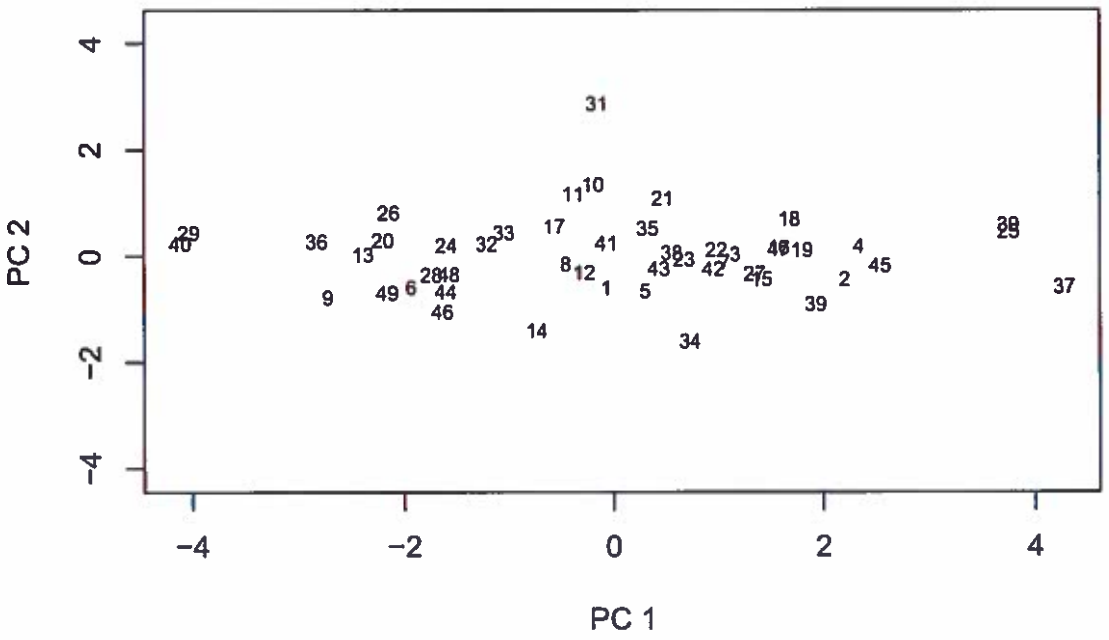
The first component seems to be a very general size component, but the second component really picks up the variability in keel and sternum length. Let's make a scree plot

```
plot(1:(length(bird.pc$sdev)),  (bird.pc$sdev)^2, type='b',
main="Scree Plot", xlab="Number of Components", ylab="Eigenvalue Size")
```



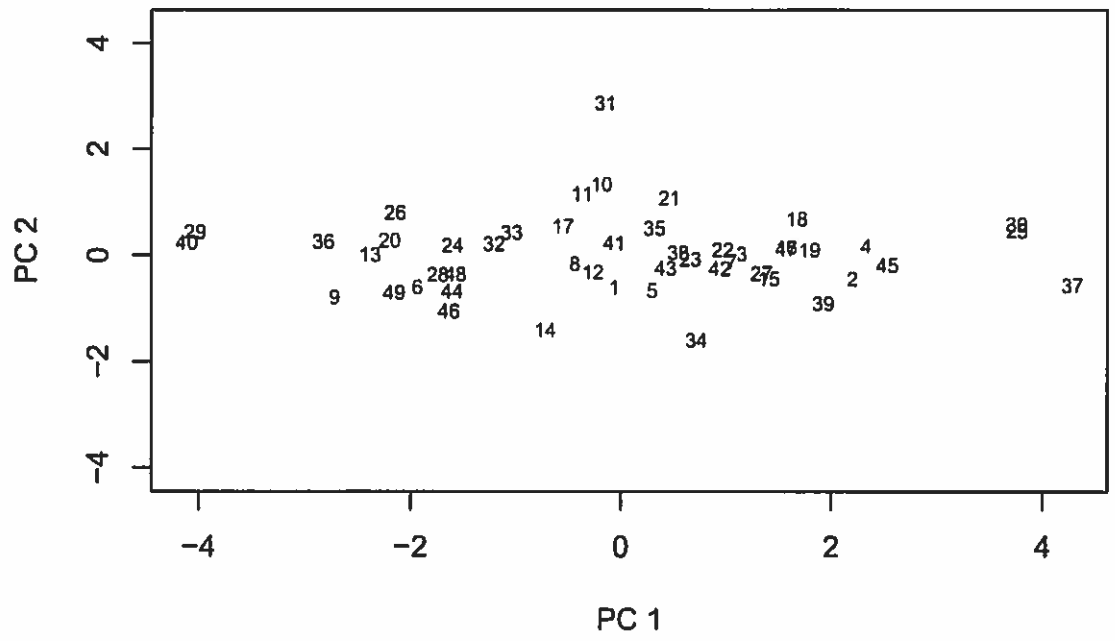
Where does the "elbow" occur? What seems to be a reasonable number of PCs to use? We plot the PC scores for the sample data in the space of the first two principal components:

```
plot(bird.pc$scores[,1], bird.pc$scores[,2], ylim=range(bird.pc$scores[,1]),
xlab="PC 1", ylab="PC 2", type ='n', lwd=2)
# labeling points with IDs for birds:
text(bird.pc$scores[,1], bird.pc$scores[,2], labels=ID, cex=0.7, lwd=2)
```



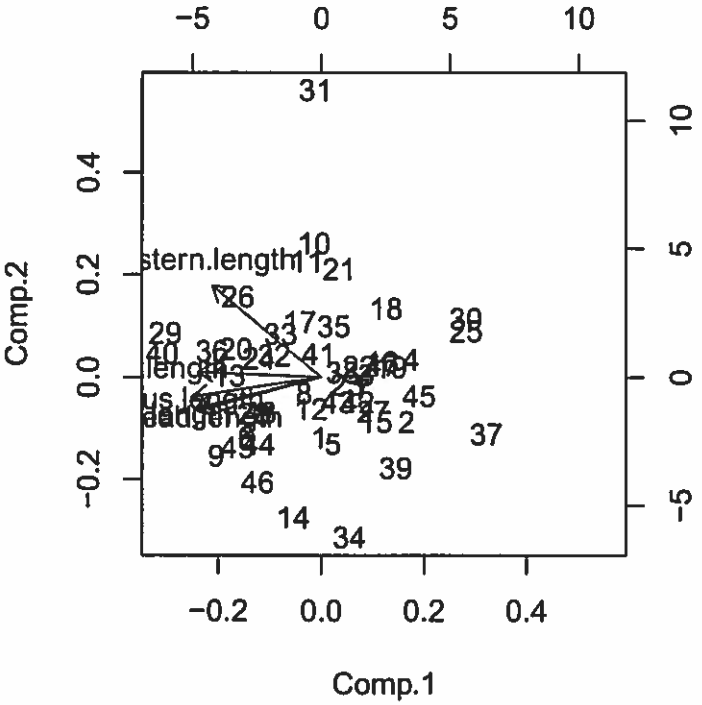
Interestingly, birds 1 to 21 survived a storm in RI while birds 22 to 49 died. Let's see the survivors (red) and the deceased (blue) separated by color.

```
plot(bird.pc$scores[,1], bird.pc$scores[,2], ylim=range(bird.pc$scores[,1]),
xlab="PC 1", ylab="PC 2", type ='n', lwd=2)
# labeling points with IDs for birds:
text(bird.pc$scores[,1], bird.pc$scores[,2], labels=ID, cex=0.7, lwd=2,
col=c(rep("red", times = 21), rep("blue", times=28)) )
```

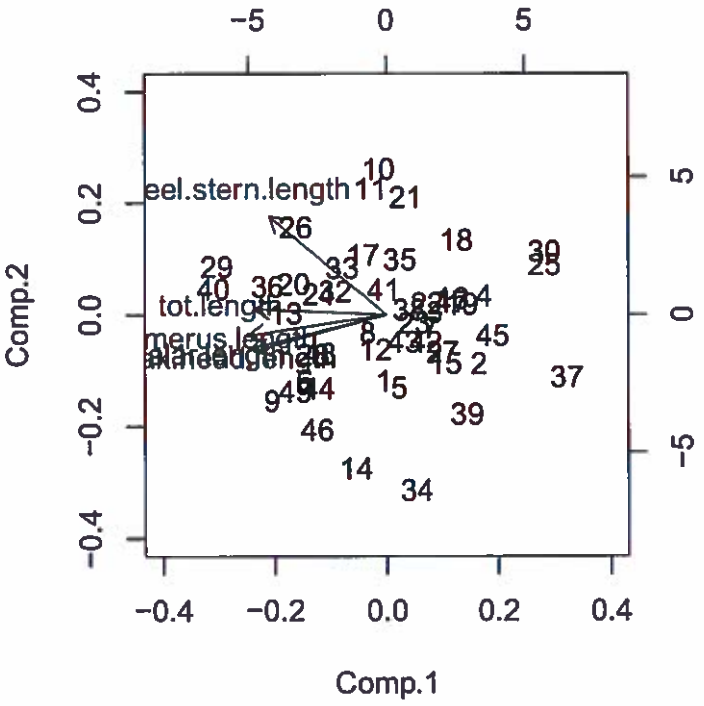


The medium-sized birds seem to have had the best survival chances. The biplot can add information about the variables to the plot of the first two PC scores:

```
biplot(bird.pc,xlabs=ID)
```




```
# Expanding the plotting window a bit:
biplot(bird.pc,xlabs=ID, xlim=c(-0.4,0.4), ylim=c(-0.4,0.4))
```



4. Statistical Inference with Principal Components

The principal components solutions are based on the eigenvalues and eigenvectors of the sample covariance matrix S. The eigenvalues and eigenvectors of the true covariance matrix Σ are unknown. We can (for large samples) perform inference about these unknown eigenvalues and eigenvectors. These results assume the data vectors $X_1, X_2, \cdots, X_q$ are a random sample from a multivariate normal distribution. This assumption could be checked with plots such as the chi-squared plot. If this multivariate normal assumption is close to correct, the inferences will be approximately valid.

Let the variance of the i^{th} population principal component (which is the i^{th} eigenvalue of the true covariance matrix Σ) be denoted by ξ_i. Under multivariate normality, the large-sample distribution of the variance of the i^{th} (sample) principal component λ_i is $N(\xi_i, 2\xi_i^2/n)$. Furthermore, the λ_i's are independent. Thus a large-sample $100(1-\alpha)\%$ confidence interval for ξ_i is

$$\left(\frac{\lambda_i}{1 + z_{\alpha/2}\sqrt{2/n}}, \frac{\lambda_i}{1 - z_{\alpha/2}\sqrt{2/n}} \right).$$

If we want simultaneous intervals for m such eigenvalues, replace $z_{\alpha/2}$ with $z_{\alpha/2m}$ in the formula to get Bonferroni confidence intervals.

We can write a R function to calculate the confidence intervals for eigenvalue(s) of Sigma:

```
evalue.CI=function(datamatrix, labelvec, m=length(labelvec), conf.level=0.95)
  {
alpha=1-conf.level
n=nrow(datamatrix)
z=qnorm(alpha/(2*m),lower=F)
lambdas<- princomp(datamatrix)$sdev^2
print(lambdas)
LCL=lambdas[labelvec]/(1+z*sqrt(2/n))
```

```
UCL=lambdas[labelvec]/(1-z*sqrt(2/n))
CIs=cbind(LCL,UCL)
return(CIs)
}
```

In the bird example, the 95% CI for variance of the first population principle component is

```
evalue.CI(bumpbird[,-1],1)
```

```
##      Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
## 34.60482313  4.52812344  0.61804191  0.30640029  0.07593901
```

```
##             LCL      UCL
## Comp.1 24.78904 57.29015
```

The joint 95% CIs for variances of the first two population principle components are

```
evalue.CI(bumpbird[,-1],c(1,2))
```

```
##      Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
## 34.60482313  4.52812344  0.61804191  0.30640029  0.07593901
```

```
##              LCL       UCL
## Comp.1 23.818879 63.243476
## Comp.2  3.116757  8.275559
```

Note that the first interval is wider because the FAMILYWISE confidence coefficient is 95% here.

We can also calculate the 99% CI for variance of the first population PC:

```
evalue.CI(bumpbird[,-1],1, conf.level=0.99)
```

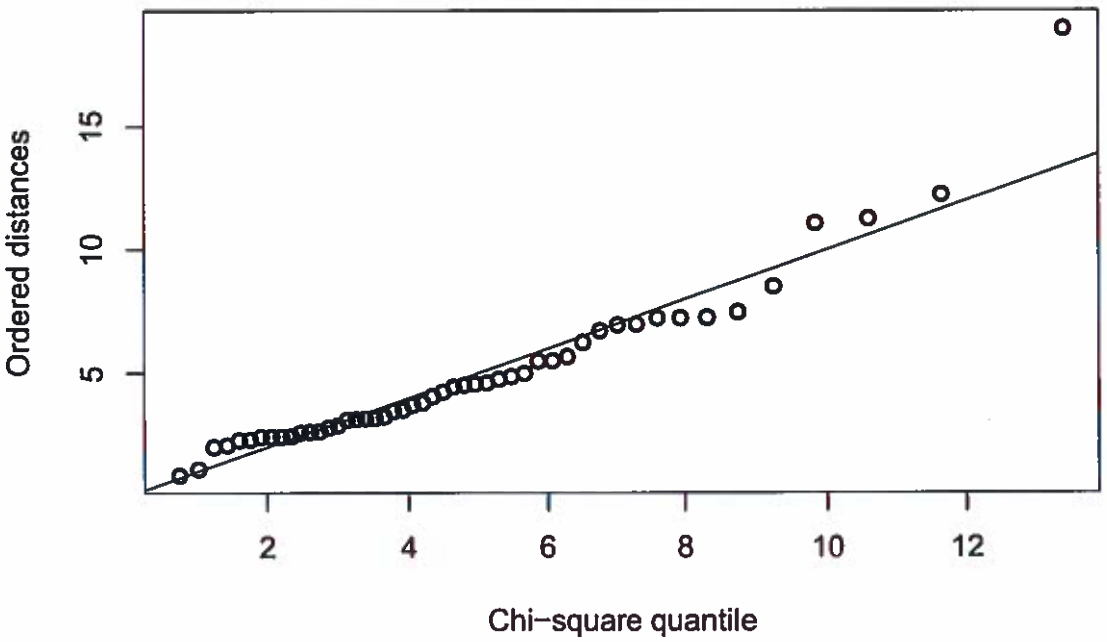
```
##      Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
## 34.60482313  4.52812344  0.61804191  0.30640029  0.07593901
```

```
##            LCL      UCL
## Comp.1 22.7604 72.15292
```

Check multivariate normality

```
#Function will produce a  chi-square plot to test multivariate normality
chisplot <- function(x) {
    if (!is.matrix(x)) stop("x is not a matrix")

    ### determine dimensions
    n <- nrow(x)
    p <- ncol(x)
    #
    xbar <- apply(x, 2, mean)
    S <- var(x)
    S <- solve(S)
    index <- (1:n)/(n+1)
    #
    xcent <- t(t(x) - xbar)
    di <- apply(xcent, 1, function(x,S) x %*% S %*% x,S)
    #
    quant <- qchisq(index,p)
    plot(quant, sort(di), ylab = "Ordered distances",
         xlab = "Chi-square quantile", lwd=2,pch=1)
    abline(a=0,b=1)
```

```
}
chisplot(as.matrix(bumpbird[,-1]))
```



C5 Factor Analysis

Jianxuan Liu

Fall 2018

It is not always possible to measure the quantities of interest directly (e.g. intelligence quotient). Behavioral scientist Charles Spearman is credited with being the originator and pioneer of the classical theory of mental tests, the theory of intelligence and what is now called Factor Analysis. The appeal of Factor Analysis lies in the ease of use and the recognition that there is an association between the hidden quantities and the measured quantities. The aim of Factor Analysis is:

1. to exhibit the relationship between the measured and the underlying variables;
2. to estimate the underlying variables, called the hidden or latent variables.

FA and PCA have similar themes, i.e., to explain covariation between variables via linear combinations of other variables. However, there are distinctions between the two approaches:

- FA assumes a statistical model that describes covariation in observed variables via linear combinations of latent variables
- PCA finds uncorrelated linear combinations of observed variables that explain maximal variance (no latent variables here)
- FA refers to a statistical model, whereas PCA refers to the eigenvalue decomposition of a covariance (or correlation) matrix.

1. The Orthogonal Factor Model

Let the observable random vector $\mathbf{X}$, with p components, has mean μ and covariance matrix Σ. The factor model postulates that $\mathbf{X}$ is linearly dependent upon a few unobservable random variables $F_1, F_2, \cdots, F_m$, called **common factors** (or **factor scores** or **vector of latent or hidden variables**). In particular, the factor analysis model is

$$\mathbf{X} - \mu = \mathbf{L}\mathbf{F} + \epsilon \qquad \mathbf{X} = \mathbf{L}\mathbf{F} + \mu + \epsilon. \qquad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{pmatrix}$$

- $\mathbf{L}$ is a $p \times m$ linear transformation matrix, called the **factor loadings**. l_{jk} is the loading of the j^{th} variable on the k^{th} common factor
- $\mathbf{F} = (F_1, \cdots, F_m)^T$ where F_k is the score on the k^{th} common factor (Assumption 1)

$$\mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{pmatrix}$$

$$E(\mathbf{F}) = \mathbf{0}$$
$$Cov(\mathbf{F}) = E(\mathbf{F}\mathbf{F}^T) = \mathbf{I}$$

$$\begin{aligned} X_1 - \mu_1 &= l_{11}F_1 + l_{12}F_2 + \cdots + l_{1m}F_m + \epsilon \\ X_2 - \mu_2 &= l_{21}F_1 + l_{22}F_2 + \cdots + l_{2m}F_m + \\ &\vdots \\ X_p - \mu_p &= l_{p1}F_1 + l_{p2}F_2 + \cdots + l_{pm}F_m + \epsilon \end{aligned}$$

- $\epsilon = (\epsilon_1, \cdots, \epsilon_p)^T$ is a p-dimensional random vector of latent error terms such that (Assumption 2)

$$E(\epsilon) = \mathbf{0}$$
$$Cov(\epsilon) = E(\epsilon\epsilon^T) = \Psi = \begin{bmatrix} \psi_1 & 0 & \cdots & 0 \\ 0 & \psi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_p \end{bmatrix}$$

- If $\mathbf{F}$ and ϵ are uncorrelated, (Assumption 3)

$$Cov(\epsilon, \mathbf{F}) = E(\epsilon\mathbf{F}^T) = \mathbf{0}$$

$$X = LF + \mu + \varepsilon$$

The covariance structure

$$\Sigma = Cov(\mathbf{X}) = \mathbf{L}\mathbf{L}^T + \Psi$$

$$\begin{aligned} cov(X) &= E\left((X-\mu)(X-\mu)^T\right) \\ &= E\left((LF+\varepsilon)(LF+\varepsilon)^T\right) \\ &= E\left(LFF^TL^T + LF\varepsilon^T + \varepsilon LF + \varepsilon\varepsilon^T\right) \\ &= L\underbrace{E(FF^T)}_{I_{m\times m}}L^T + \underbrace{LE(F\varepsilon^T)}_{=0} + \underbrace{E(\varepsilon F^T)L^T}_{=0} + \underbrace{E(\varepsilon\varepsilon^T)}_{\Psi} \\ &= LL^T + 0 + \Psi \end{aligned}$$

non orthogonal. $F^T = \Phi$

If $Cov(\mathbf{F}) = \Phi$, then $\Sigma = \mathbf{L}\Phi\mathbf{L}^T + \Psi$.

This implies that the covariance between $\mathbf{X}$ and $\mathbf{F}$ has the form

$$Cov(\mathbf{X}, \mathbf{F}) = E[(\mathbf{X}-\mu)\mathbf{F}^T] = E[(\mathbf{LF}+\epsilon)\mathbf{F}^T] = \mathbf{L}$$

The portion of variance of the i^{th} variable that is explained by the m common factors is called the **communality** of the i^{th} variable:

is the proportion of variance of X_i explained by the m common factor

If X_i is informative $\Rightarrow$ communality is high.

$$\underbrace{\sigma_{ii}}_{Var(X_i)} = \underbrace{h_i^2}_{\text{communality}} + \underbrace{\psi_i}_{\text{uniqueness}}$$

specific variance.

where

- σ_{ii} is the **variance** of X_i (i.e., the i^{th} diagonal of Σ)
- $h_i^2 = (\mathbf{L}\mathbf{L}^T)_{ii} = (l_{i1}^2 + l_{i2}^2 + \cdots + l_{im}^2)$ is the **communality** of X_i. Note that the communality h_i^2 is the sum of squared loadings for X_i
- ψ_i is the **specific variance** (or uniqueness) of X_i

Example (Idea of this example: Given $X \sim (\mu, \Sigma)$, we want to find L)

We consider a one-factor model. The factor might be physical fitness, which we could obtain from the performance of two different sports, or the talent for language, which might arise from oral and written comunication skills. We assume that the two-dimensional random vector $\mathbf{X}$ has mean zero and covariance matrix

$$\Sigma = \begin{pmatrix} 1.25 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}.$$

Calculate the communality and uniqueness for a one-factor model.

* Minimum variance is the prefered choice

observed

$$\underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} F + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \qquad F \text{ is a scalar}$$

$$\Sigma = LL^T + \Phi$$

$$\sigma_{11} = l_1^2 + \phi_{11}$$

$$\sigma_{22} = l_2^2 + \phi_{22}$$

$$\sigma_{12} = l_1 l_2$$

$$cov(X_i, X_k) = l_{i1}l_{k1} + \cdots + l_{im}l_{km}$$

$$\left.\begin{aligned} 1.25 &= \sigma_{11} > l_1^2 \\ 0.5 &= \sigma_{22} > l_2^2 \\ \sigma_{12} &= l_1 l_2 = 0.5 \end{aligned}\right\}_2 \Rightarrow \begin{cases} l_1 = 1 \\ l_2 = 0.5 \end{cases} \text{ or } \begin{cases} l_1^* = \frac{3}{4} \\ l_2^* = \frac{2}{3} \end{cases}$$

$$\bullet \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} =$$

Consider $\begin{cases} l_1 = 1 \\ l_2 = 0.5 \end{cases}$

Sol. • $\begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \Rightarrow LL^T = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}\begin{bmatrix} 1 & 0.5 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.25 \end{bmatrix}$

$\Sigma = LL^T + \Psi \qquad \Psi = E(\varepsilon)$ $\qquad tr(\Psi) = 0.25 + 0.25 = 0.5$

$\begin{bmatrix} 1.25 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.25 \end{bmatrix} + \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$

• $\begin{bmatrix} l_1^* \\ l_2^* \end{bmatrix} = \begin{bmatrix} 3/4 \\ 2/3 \end{bmatrix} \quad LL^T = \begin{bmatrix} 9/16 & 1/2 \\ 1/2 & 4/9 \end{bmatrix} \Rightarrow \underbrace{\begin{bmatrix} 1.25 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}}_{\Sigma} = \underbrace{\begin{bmatrix} 9/16 & 1/2 \\ 1/2 & 4/9 \end{bmatrix}}_{LL^T} + \underbrace{\begin{bmatrix} 11/16 & 0 \\ 0 & 1/18 \end{bmatrix}}_{\Psi} \Rightarrow tr(\phi) = \frac{11}{16} + \frac{1}{18} = 0.74$

For $l_1^* = \frac{3}{4}$ $l_2^* = \frac{2}{3}$

$1.25 = \left(\frac{3}{4}\right)^2 + \frac{1}{16}$ ← var (X_1) communality u

$0.5 = \left(\frac{2}{3}\right)^2 + \frac{1}{18}$ ← var (X_2) comm uniq

Example: Chicken-bone data (from Example 9.14 in Johnson & Wichern)

The correlation matrix for chicken-bone measurements.

```
var.names <- c("SkullL", "SkullB", "FemurL", "TibiaL", "HumerusL", "UlnaL")
CorMat <- read.table("Wichern_data/E9-14.dat", fill = TRUE, col.names = var.names)
row.names(CorMat) <- var.names
n <- dim(CorMat)
# fill the upper triangle
#for(j in 1:(n-1)){CorMat[j, (j+1):n] <- CorMat[(j+1):n, j]}
CorMat <- as.matrix(CorMat)
```

The estimated factor loadings were extracted by the maximum likelihood procedure

```
# Enter the estimated factor loadings   2 factors models
L <- matrix(c(0.602, 0.200, 0.467, 0.154, 0.926, 0.143,
              1.000, 0.000, 0.874, 0.476, 0.894, 0.327), ncol= 2,byrow=T)
as.data.frame(cbind(var.names,L))
```

```
##   var.names    V2    V3
## 1    SkullL 0.602   0.2
## 2    SkullB 0.467 0.154
## 3    FemurL 0.926 0.143
## 4    TibiaL     1     0
## 5  HumerusL 0.874 0.476
## 6     UlnaL 0.894 0.327
```

```
# the specific variances
psi <- diag(diag(CorMat - L %*% t(L)))
round(diag(psi), digits = 3)
```

```
## [1] 0.598 0.758 0.122 0.000 0.010 0.094
```

← uniquenesses.

```
#  the communalities
hi2 <- apply(L, 1, function(x) sum(x^2))
round(hi2, digits =.3)
```

```
## [1] 0.402 0.242 0.878 1.000 0.990 0.906
```

~~uniquenesses~~ ~~for the last~~

2. Factor Loadings

The main two approaches to estimating the factor loadings divide into nonparametric methods and methods which rely on the normality of the data. For normal data, we expect the latter methods to be better; in practice, methods based on assumptions of normality still work well if the distribution of the data does not deviate too much from the normal distribution.

A tool for finding the factors is the covariance matrix. In a parametric framework, there are two main methods for determining the factors:

- Principal Component Factor Analysis
- Maximum likelihood method

1 Principal Component Factor Analysis

Note that the parameters of interest are the factor loadings $\mathbf{L}$ and specific variances on the diagonal of Ψ. Let $\mathbf{X} \sim N_p(\mu, \Sigma)$. For $m < p$ common factors, the PCA solution estimates $\mathbf{L}$ and Ψ as

$$\hat{\mathbf{L}} = [\lambda_1^{1/2}\mathbf{v}_1, \lambda_2^{1/2}\mathbf{v}_2, \cdots, \lambda_m^{1/2}\mathbf{v}_m]$$

$$\hat{\Psi}_i = \sigma_{ii} - \hat{h}_i^2$$

where $\Sigma = \mathbf{V}\Lambda\mathbf{V}^T$ is the spectral decomposition of Σ. The communality of the i^{th} variable is estimated as

$$\hat{h}_i^2 = \sum_{k=1}^{m} \hat{l}_{ik}^2.$$

Proportion of total sample variance explained by the k-th factor is

$$R_k^2 = \frac{\sum_{i=1}^{p} \hat{l}_{ik}^2}{\sum_{i=1}^{p} \sigma_{ii}} = \frac{(\lambda_k^{1/2}\mathbf{v}_k)^T(\lambda_k^{1/2}\mathbf{v}_k)}{\sum_{i=1}^{p} \sigma_{ii}} = \frac{\lambda_k}{\sum_{i=1}^{p} \sigma_{ii}}$$

```
# the proportion of variance explained by each factor
VxF <- apply(L, 2, function(x) sum(x^2))
totVar <- sum(diag(CorMat))
propVxF <- VxF / totVar
round(propVxF, digits = 3)
```

```
## [1] 0.667 0.070
```

```
# residual matrix R-LL^T-Psi
Rmat <- CorMat - L %*% t(L) - psi
round(Rmat, digits = 4)
```

```
##           SkullL  SkullB  FemurL TibiaL HumerusL UlnaL
## SkullL    0.0000      NA      NA     NA       NA    NA
## SkullB    0.1931  0.0000      NA     NA       NA    NA
## FemurL   -0.0171 -0.0325  0.0000     NA       NA    NA
## TibiaL    0.0000  0.0000  0.0000      0       NA    NA
## HumerusL -0.0003  0.0005 -0.0004      0        0    NA
## UlnaL    -0.0006 -0.0179  0.0034      0        0     0
```

iterative method

(2) **Principal Factor Analysis or Principal Axis Factoring** (modified method of (1))

Principal Factor Analysis or Principal Axis Factoring is a modified approach. In Principal Component Factor Analysis, the underlying covariance matrix is Σ. In contrast, Principal Factor Analysis is based on the scaled covariance matrix of the common factors. Assume we are applying Factor Analysis to a sample correlation matrix $\boldsymbol{R}$

$$\boldsymbol{R} - \boldsymbol{\Psi} = \boldsymbol{L}\boldsymbol{L}^T$$

and we have some initial estimate of the specific variance $\hat{\psi}_i$. We can use $\hat{\psi}_i = 1/r^{ii}$ where r^{ii} is the i^{th} diagonal of $\boldsymbol{R}^{-1}$. The iterated principal axis factoring algorithm:

1. Form $\tilde{\boldsymbol{R}} = \boldsymbol{R} - \hat{\boldsymbol{\Psi}}$ given current $\hat{\psi}_i$ estimates
2. Updata $\tilde{\boldsymbol{L}} = [\tilde{\lambda}_1^{1/2}\tilde{\mathbf{v}}_1, \tilde{\lambda}_2^{1/2}\tilde{\mathbf{v}}_2, \cdots, \tilde{\lambda}_m^{1/2}\tilde{\mathbf{v}}_m]$ where $\tilde{\boldsymbol{R}} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{V}}^T$ is the eigenvalue decomposition of $\tilde{\boldsymbol{R}}$
3. Update $\hat{\psi}_i = 1 - \sum_{k=1}^{m} \tilde{l}_{ik}^2$

(3) **Maximum Likelihood Method for Factor Analysis**

(MLE of multivariate normal distribution) Let $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n$ be a random sample from a normal population with mean μ and covariance Σ. Then

$$\hat{\mu} = \bar{\mathbf{x}}, \text{and } \hat{\Sigma} = \frac{1}{n}\sum_{j=1}^{n}(\mathbf{X}_j - \bar{\mathbf{X}})(\mathbf{X}_j - \bar{\mathbf{X}})^T = \frac{n-1}{n}\mathbf{S}$$

are the maximum likelihood estimators of μ and Σ, respectively. Proof.

• $X \sim N(\mu, \Sigma)$

* The likelihood

$$L(\mu, \Sigma | \underset{\sim}{X}) = \frac{1}{(2\pi)^{\frac{np}{2}} |\Sigma|^{\frac{n}{2}}} \exp\left\{-\frac{1}{2} tr\left[\Sigma^{-1}\left(\sum_{j=1}^{n}(X_j - \bar{X})(X_j - \bar{X})^T\right)\right] + n(\bar{X} - \mu)(\bar{X} - \mu)^T\right\}$$

• Σ^{-1} is positive definite, so $(X_j - \bar{X})\Sigma^{-1}(X_j - \bar{X})^T > 0$ unless $\underset{\sim}{\mu} = \bar{X}$

Thus the likelihood $L(\mu, \Sigma)$ is maximized wrt μ at $\hat{\mu} = \bar{x}$

Let $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n$ be a random sample from $N_p(\mu, \boldsymbol{L}\boldsymbol{L}^T + \boldsymbol{\Psi})$, where $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T + \boldsymbol{\Psi}$. The maximum likelihood estimators $\hat{\mathbf{L}}, \hat{\boldsymbol{\Psi}}$ and $\hat{\mu}$ maximize the log-likelihood function for a sample of n observations

$$\mathcal{L}(\mu, L, \Psi) = \frac{np\log(2\pi)}{2} + \frac{n\log(|\ \Sigma^{-1}\ |)}{2} - \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \mu)^T\Sigma^{-1}(\mathbf{x}_i - \mu))}{2}$$

subject to $\hat{\mathbf{L}}^T\hat{\boldsymbol{\Psi}}^{-1}\hat{\mathbf{L}}$ being diagonal. Use an iterative algorithm to maximize $\mathcal{L}$. The maximum likelihood estimates of the communalities are

$$\hat{h}_i^2 = \hat{l}_{i1}^2 + \hat{l}_{i2}^2 + \cdots + \hat{l}_{im}^2, i = 1, 2, \cdots, p.$$

The proportion of total sample variance due to j^{th} factor is

$$\frac{\hat{l}_{1j}^2 + \hat{l}_{2j}^2 + \cdots + \hat{l}_{pj}^2}{s_{11} + s_{22} + \cdots + s_{pp}}.$$

Example: Stock Price Data $p=5$ $m=2$

Data are weekly gains in stock prices for 100 consecutive weeks for five companies: Allied Chemical, Du Pont, Union Carbide, Exxon and Texaco. Note that the first three are chemical companies and the last two are oil companies. Fit an $m = 2$ orthogonal factor model

```
stocks <- read.table("stocks.txt", col.names = paste("x", 1:5, sep = ""))
names(stocks)=c("Allied", "Dupont", "Carbide", "Exxon", "Texaco")
head(stocks);dim(stocks)
```

chemical company oil companies

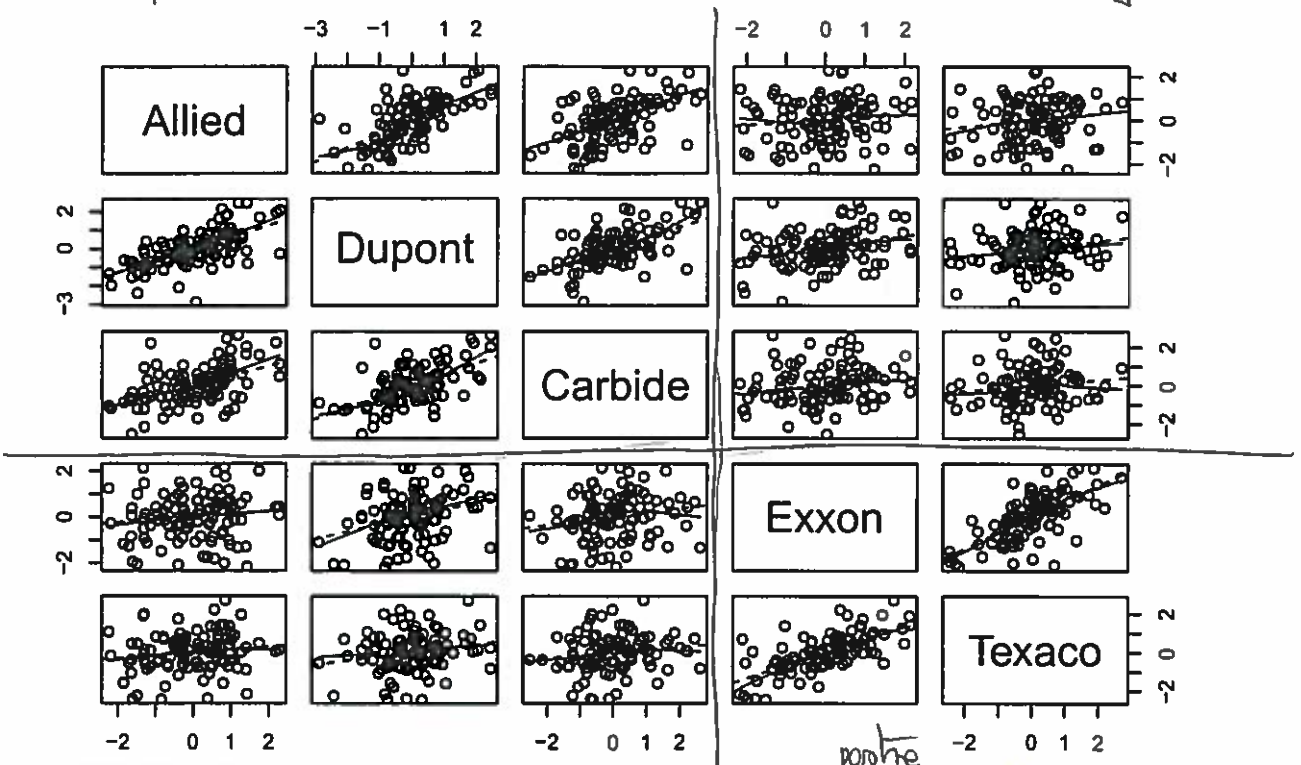
```
##       Allied     Dupont     Carbide      Exxon     Texaco
## 1  0.0130338 -0.0078431 -0.0031889 -0.0447693  0.0052151
## 2  0.0084862  0.0166886 -0.0062100  0.0119560  0.0134890
## 3 -0.0179153 -0.0086393  0.0100360  0.0000000 -0.0061428
## 4  0.0215589 -0.0034858  0.0174353 -0.0285917 -0.0069534
## 5  0.0108225  0.0037167 -0.0101345  0.0291900  0.0409751
## 6  0.0101713 -0.0121978 -0.0083768  0.0137083  0.0029895
```

```
## [1] 103   5
```

```
# plot standardized data on pairwise scatter plots
stockss <- scale(stocks, center = T, scale = T)
pairs(stockss, labels = c("Allied", "Dupont", "Carbide", "Exxon", "Texaco"),
      panel = function(x, y) { panel.smooth(x, y)
        abline(lsfit(x, y), lty = 2)})
```

positive .

hard to say the relationship



positive

```
# perform factor analysis with (two) factor
stocks.fac=factanal(x = stocks, factors = 2,
                    scores = "regression",
                    rotation = "varimax",
                    method = "mle")
stocks.fac
```

Another function may be fa

```
##
## Call:
## factanal(x = stocks, factors = 2, scores = "regression", rotation = "varimax",     method = "mle
##
## Uniquenesses:
##  Allied  Dupont Carbide   Exxon  Texaco
##   0.417   0.275   0.542   0.005   0.530
##
## Loadings: Factor loading
##         Factor1 Factor2
## Allied   0.763
## Dupont   0.819   0.232
## Carbide  0.668   0.108
## Exxon    0.113   0.991
## Texaco   0.108   0.677
##
##                Factor1 Factor2
## SS loadings      1.725   1.507
## Proportion Var   0.345   0.301
## Cumulative Var   0.345   0.646
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 1.97 on 1 degree of freedom.
## The p-value is 0.16
```

the first factor is affected by the first 3 companies .

← sum of square of the loading (don't care)

← R_i in page 4 just cumulative

This only have us decide # of factors

pvalue = 0.167 → big ⇒ → we see that since 6 factors is ~~too~~ enough .

The proportions of total variance accounted by the first and second factors are 0.345 and 0.301.

The first factor is dominated by chemical companies while the second factor by oil companies

Now we calculate the residuals:

```
stocks.pred <- stocks.fac$loadings%*% t(stocks.fac$loadings) + diag(stocks.fac$uniquenesses)
residuals <- var(stocks) - stocks.pred
```

Now list factor scores

```
stocks.fac$scores
```

Example: Air pollution data

Use air pollution variables X_1, X_2, X_5 ant X_6. Compare the factorization obtained by the principal component and the maximum likelihood methods for a factor model with $m = 1$ and $m = 2$.

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.4.4
```

```
air <- read.table("Wichern_data/T1-5.dat")
names(air) <- c("Wind", "Solar", "CO", "NO", "NO2", "O3", "HC")
air.4 <- air[,c(1,2,5,6)]

# principal component solution to a factor model with m =1, no rotation
air.pc1 <- principal(air.4, nfactors = 1, covar = TRUE, rotate = "none")
# MLE of L and \Psi for m =1 no rotation
air.fa1 <- fa(air.4, nfactors = 1, covar = TRUE, fm = "ml", rotate = "none")

# principal component solution to a factor model with m =1, varimax
air.pc1.v <- principal(air.4, nfactors = 1, covar = TRUE, rotate = "varimax")
# MLE of L and \Psi for m =1,varimax
air.fa1.v <- fa(air.4, nfactors = 1, covar = TRUE, fm = "ml", rotate = "varimax")

# principal component solution to a factor model with m =2, no rotation
air.pc2 <- principal(air.4, nfactors = 2, covar = TRUE, rotate = "none")
# MLE of L and \Psi for m =2 no rotation
air.fa2 <- fa(air.4, nfactors = 2, covar = TRUE, fm = "ml", rotate = "varimax")

# principal component solution to a factor model with m =1, varimax
air.pc2.v <- principal(air.4, nfactors = 2, covar = TRUE, rotate = "varimax")
# MLE of L and \Psi for m =1,varimax
air.fa2.v <- fa(air.4, nfactors = 2, covar = TRUE, fm = "ml", rotate = "varimax")
```

	PC			ML		
	PC1	ψ_i	h_i^2	ML1	ψ_i	h_i^2
Wind	-0.1750	2.469	0.031	-0.1747	2.469	0.031
Solar	17.3247	0.371	300.145	17.2962	1.358	299.158
NO2	0.4214	11.186	0.178	0.4207	11.187	0.177
O3	1.9587	27.142	3.837	1.9555	27.154	3.824

(a) Principal component and ML solutions for 1 factor with no rotation

	PC				ML			
	PC1	PC2	ψ_i	h_i^2	ML1	ML2	ψ_i	h_i^2
Wind	-0.1750	-0.4048	2.306	0.194	-0.1747	-0.3975	2.311	0.189
Solar	17.3247	-0.6086	0.001	300.515	17.2962	-0.5976	1.001	299.515
NO2	0.4214	0.7422	10.635	0.728	0.4207	0.7288	10.655	0.708
O3	1.9587	5.1867	0.239	30.739	1.9555	5.0932	1.213	29.765

(b) Principal component and ML solutions for 2 factors with no rotation

	PC			ML		
	PC1	ψ_i	h_i^2	ML1	ψ_i	h_i^2
Wind	-0.1750	2.469	0.031	-0.1747	2.469	0.031
Solar	17.3247	0.371	300.145	17.2962	1.358	299.158
NO2	0.4214	11.186	0.178	0.4207	11.187	0.177
O3	1.9587	27.142	3.837	1.9555	27.154	3.824

(c) Principal component and ML solutions for 1 factor with varimax rotation

	PC				ML			
	RC1	RC2	ψ_i	h_i^2	ML1	ML2	ψ_i	h_i^2
Wind	-0.0758	-0.4344	2.306	0.194	-0.0829	-0.4262	2.311	0.189
Solar	16.9895	3.4457	0.001	300.515	17.0038	3.2225	1.001	299.515
NO2	0.2368	0.8200	10.635	0.728	0.2500	0.8035	10.655	0.708
O3	0.6960	5.5004	0.239	30.739	0.7870	5.3987	1.213	29.765

(d) Principal component and ML solutions for 2 factors with varimax rotation

Table 1: Principal component and maximum likelihood solutions with and without varimax with 1 and 2 factors (factor loadings, uniquenesses, and communalities)

	PC	ML
	Factor 1	Factor 1
SS loadings	304.1895	303.1895
Proportion Var	0.8808	0.8779
Cumulative Var	0.8808	0.8779

(a) Principal component and ML solutions for 1 factor with no rotation

	PC		ML	
	Factor 1	Factor 2	Factor 1	Factor 2
SS loadings	304.1895	27.9874	303.1895	26.9874
Proportion Var	0.8808	0.0810	0.8779	0.0781
Cumulative Var	0.8808	0.9618	0.8779	0.9560

(b) Principal component and ML solutions for 2 factors with no rotation

	Factor 1	Factor 1
SS loadings	304.1895	303.1895
Proportion Var	0.8808	0.8779
Cumulative Var	0.8808	0.8779

(c) Principal component and ML solutions for 1 factor with varimax rotation

	PC		ML	
	Factor 1	Factor 2	Factor 1	Factor 2
SS loadings	289.1885	42.9884	289.8194	40.3575
Proportion Var	0.8374	0.1245	0.8392	0.1169
Cumulative Var	0.8374	0.9618	0.8392	0.9560

(d) Principal component and ML solutions for 2 factors with varimax rotation

Table 2: Principal component and maximum likelihood solutions with and without varimax with 1 and 2 factors (variance explained)

In both the 1 and 2 factor models, the principal component and maximum likelihood solutions are extremely similar. The only apparent differences come in the size of the uniquenesses (and thus to some degree communalities), having largest differences between the principal component and maximum likelihood solutions for Solar and O3 which have the largest variances

```
# variances
round(diag(cov(air.4)), digits = 2)
```

```
##    Wind  Solar    NO2     O3
##    2.50 300.52  11.36  30.98
```

To decide the adequate number of factors that we should use.

4. Asymptotic Results and the Number of Factors

p: data size m: # of factors.

We consider normal models and formulate hypothesis tests for the number of factors. In PCA, very small eigenvalues are evidence that the corresponding PC score is negligible. In FA, the likelihood of the data drives the hypothesis testing for the adequacy of a m-factor model.

$$H_0 : \Sigma_{p\times p} = L_{p\times m} L^T_{m\times p} + \Psi_{p\times p}$$

against $H_1 : \Sigma$ any other positive definite matrix. When Σ does not have any special form, the maximum of the likelihood function with $\hat{\Sigma} = (n-1)\mathbf{S}/n$ is proportional to $|\mathbf{S}_n|^{-n/2} e^{-np/2}$, where $\mathbf{S}_n = (n-1)\mathbf{S}/n$.

Under H_0, the MLE for $\hat{\mu} = \bar{x}$ and $\hat{\Sigma} = \hat{L}\hat{L}^T + \hat{\Psi}$ where $\hat{L}$ and $\hat{\Psi}$ are the MLE of L and Ψ.

The likelihood ratio statistic for testing H_0 is

$$-2\ln\Lambda = -2\ln\left(\frac{\text{maximized likelihood under} H_0}{\text{maximized likelihood}}\right) = -2\ln\left(\frac{|\hat{\Sigma}|}{|\mathbf{S}_n|}\right)^{-n/2} + n\left[tr(\hat{\Sigma}^{-1}\mathbf{S}_n) - p\right]$$

with degrees of freedom $\nu = \frac{1}{2}p(p-1) - [p(m+1) - \frac{1}{2}m(m-1)] = \frac{1}{2}[(p-m)^2 - p - m]$. $tr(\hat{\Sigma}^{-1}\mathbf{S}_n) = p$ provided that $\hat{\Sigma} = \hat{\mathbf{L}}\hat{\mathbf{L}}^T + \hat{\Psi}$ (See Johnson Wichern Page 527). Thus we have the likelihood ratio statistic

$$-2\ln\Lambda = n\ln\left(\frac{|\hat{\Sigma}|}{|\mathbf{S}_n|}\right), \quad n\to\infty \sim \chi^2_\nu$$

which is approximately χ^2_ν as $n \to \infty$.

In the stocks data, we test the adequacy of the m-factor model with the tests

$$H_{0,m} : \Sigma = LL + \Psi, H_{1,m} : \Sigma \neq LL + \Psi$$

Starting with $m = 1$,

```
stocks.fac=factanal(x = stocks, factors = 1,
                    scores = "regression",
                    rotation = "varimax",
                    method = "mle")
stocks.fac
```

when have cov matrix need to add one option n.obs = (when we don't have the raw data)

```
##
## Call:
## factanal(x = stocks, factors = 1, scores = "regression", rotation = "varimax",     method = "mle
##
## Uniquenesses:
##  Allied  Dupont Carbide   Exxon  Texaco
```

```
##   0.488   0.234   0.560   0.880   0.922
##
## Loadings:
##         Factor1
## Allied  0.715
## Dupont  0.875
## Carbide 0.663
## Exxon   0.346
## Texaco  0.279
##
##                Factor1
## SS loadings      1.915
## Proportion Var   0.383
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 62.22 on 5 degrees of freedom.
## The p-value is 4.22e-12
```

Since p-value < 0.0001, we reject the null hypothesis and conclude that a one factor model is not sufficient to describe the correlations among the weekly returns for the five stocks.

Two factors, $m = 2$

```
stocks.fac=factanal(x = stocks, factors = 2,
                    scores = "regression",
                    rotation = "varimax",
                    method = "mle")
stocks.fac
```

```
##
## Call:
## factanal(x = stocks, factors = 2, scores = "regression", rotation = "varimax",     method = "mle"
##
## Uniquenesses:
##  Allied  Dupont Carbide   Exxon  Texaco
##   0.417   0.275   0.542   0.005   0.530
##
## Loadings:
##         Factor1 Factor2
## Allied  0.763
## Dupont  0.819   0.232
## Carbide 0.668   0.108
## Exxon   0.113   0.991
## Texaco  0.108   0.677
##
##                Factor1 Factor2
## SS loadings      1.725   1.507
## Proportion Var   0.345   0.301
## Cumulative Var   0.345   0.646
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 1.97 on 1 degree of freedom.
## The p-value is 0.16
```

Since p-value $= 0.16$, we do not reject the null hypothesis that a two-factor model is sufficient to describe the correlations among the weekly returns for the five stocks.

4. Factor Rotation

In factor analysis, the concept of **factor rotation** is commonly used. Unlike a rotation matrix, which is an orthogonal transformation, a factor rotation is a linear transformation which need not be orthogonal but which satisfies certain criteria, such as goodness-of-fit.

Let $\hat{\mathbf{L}}$ is the $p \times m$ matrix of estimated factor loadings obtained by any method(principal compoonent, maximum likelihood, etc), then $\hat{\mathbf{L}}^* = \hat{\mathbf{L}}\mathbf{R}$ where $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$ is a $p \times m$ matrix of "rotated" loadings. That is, multiplying a matrix of factor loadings by any orthogonal matrix leads to the same approximation to the covariance (or correlation) matrix.

- The estimated residual matrix also remains unchanged

$$\mathbf{S}_n - \hat{\mathbf{L}}\hat{\mathbf{L}}^T - \hat{\mathbf{\Psi}} = \mathbf{S}_n - \hat{\mathbf{L}}\mathbf{R}\mathbf{R}^T\hat{\mathbf{L}}^T - \hat{\mathbf{\Psi}} = \mathbf{S}_n - \hat{\mathbf{L}}^*\hat{\mathbf{L}}^{*T} - \hat{\mathbf{\Psi}}$$

- The specific variances $\hat{\Psi}_i$ and therefore the communalities also remain unchanged

Since only the loadings change by rotation, we rotate factors to see if we can better interpret results. There are many choices for a rotation matrix $\mathbf{R}$, and to choose, we first establish a mathematical criterion and then see which $\mathbf{R}$ can best satisfy the criterion. One possible objective is to have each one of the p variables load highly on only one factor and have moderate to negligible loads on all other factors.

Many popular orthogonal factor rotation methods try to maximize

$$V(\mathbf{L}, \mathbf{R} \mid \gamma) = \frac{1}{p}\sum_{k=1}^{m}\left[\sum_{j=1}^{p}(\hat{l}_{jk}^*/\hat{h}_j)^4 - \frac{\gamma}{p}\left(\sum_{j=1}^{p}(\hat{l}_{jk}^*/\hat{h}_j)^2\right)\right]$$

where $\hat{l}_{jk}^*$ is the rotated loading of the j^{th} variable on the k^{th} factor. $\hat{h}_j$ is the square-root of the communality for X_j

Changing the γ parameter corresponds to different criertia

- $\gamma = 1$ corresponds to varimax criterion
- $\gamma = 0$ corresponds to quartimax criterion
- $\gamma = m/2$ corresponds to equamax criterion
- $\gamma = p(m-1)/(p+m-2)$ corresponds to parsimax criterion

The procedure finds the orthogonal transformation of the loading matrix that maximizes the sum of those variances, summing across all m rotated factors.

The varimax rotation will destroy an "overall" factor. After rotation each of the p variables should load highly on at most one of the rotated factors.

The quartimax rotation try to preserve an overall factor such that each of the p variables has a high loading on that factor and create other factors such that each of the p variables has a high loading on at most one factor.

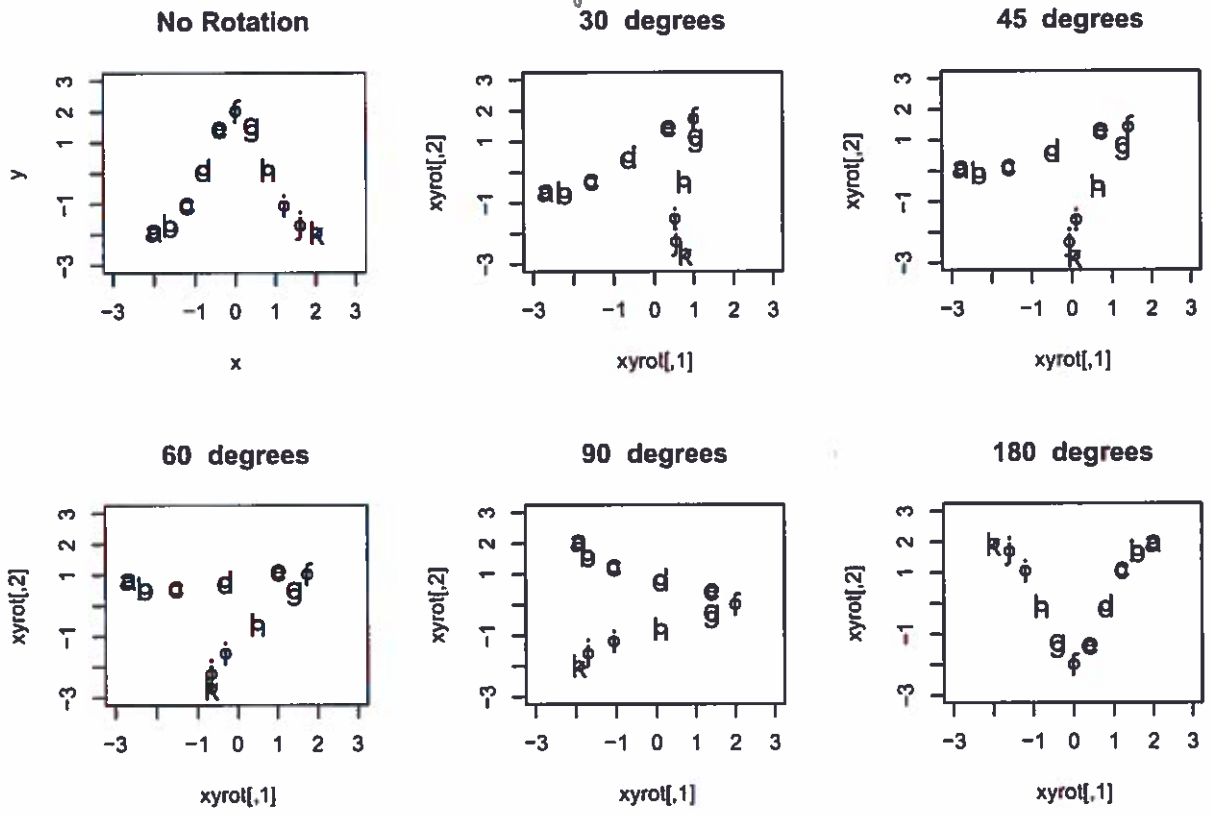
The above four rotations produce uncorrelated factors.

Example: Visualization of 2D Clockwise Rotation

```
rotmat2d <- function(theta){
matrix(c(cos(theta),sin(theta),-sin(theta),cos(theta)),2,2)
}
x <- seq(-2,2,length=11)
```

```
y <- 4*exp(-x^2) - 2
xy <- cbind(x,y)
rang <- c(30,45,60,90,180)
par(mfrow=c(2,3))
plot(x,y,xlim=c(-3,3),ylim=c(-3,3),main="No Rotation")
text(x,y,labels=letters[1:11],cex=1.5)
for(j in 1:5){
rmat <- rotmat2d(rang[j]*2*pi/360)
xyrot <- xy%*%rmat
plot(xyrot,xlim=c(-3,3),ylim=c(-3,3))
text(xyrot,labels=letters[1:11],cex=1.5)
title(paste(rang[j]," degrees"))
}
```

orthogonal rotation



Oblique transformation

Orthogonal rotations produce uncorrelated factors. However, PROMAX is a non-orthogonal (oblique) transformation that is not a rotation and can produce correlated factors

In PROMAX transformation,

1. we first perform a varimax rotate to obtain loadings $\mathbf{L}^*$.
2. Construct another $p \times m$ matrix $\mathbf{Q}$ such that

$$q_{ij} = \begin{cases} |\, l_{il}^* \,|^{m-1} \, l_{il}^* & \text{for } l_{il}^* \neq 0 \\ 0 & \text{for } l_{il}^* = 0 \end{cases}$$

where $m > 1$ is selected by trial and error, ususally $m < 4$.

3. Find a matrix $\mathbf{U}$ such that each column of $\mathbf{L}^*\mathbf{U}$ is close to the corresponding column of $\mathbf{Q}$. Choose the j-th column of $\mathbf{U}$ to minimize

$$(\mathbf{q}_j - \mathbf{L}^*\mathbf{u}_j)(\mathbf{q}_j - \mathbf{L}^*\mathbf{u}_j)^T$$

which yields

$$\mathbf{U} = (\mathbf{L}^{*T}\mathbf{L}^*)^{-1}\mathbf{L}^{*T}\mathbf{Q}$$

4. Rescale $\mathbf{U}$ so that the transformed factors have unit variance. Compute $\mathbf{D}^2 = diag[(\mathbf{U}^T\mathbf{U})^{-1}]$ and $\mathbf{M} = \mathbf{UD}$.

5. The PROMAX factors are obtained from $\mathbf{L}^*\mathbf{F} = \mathbf{L}^*\mathbf{M}\mathbf{M}^{-1}\mathbf{F} = \mathbf{L}_{promax}\mathbf{F}_{promax}$. The PROMAX transformation yields factors with loadings

$$\mathbf{L}_{promax} = \mathbf{L}^*\mathbf{M}$$

The correlation martix for the transformed factors is $(\mathbf{M}^T\mathbf{M})^{-1}$.

$$Var(\mathbf{F}_{promax}) = Var(\mathbf{M}^{-1}\mathbf{F}) = \mathbf{M}^{-1}Var(\mathbf{F})(\mathbf{M}^{-1})^T = \mathbf{M}^{-1}\mathbf{I}(\mathbf{M}^{-1})^T = (\mathbf{M}^T\mathbf{M})^{-1}$$

In the stocks example, now we change the rotation to promax from varimax

```
promax(stocks.fac$loadings, m = 2)
```

```
## $loadings
##
## Loadings:
##         Factor1 Factor2
## Allied   0.778
## Dupont   0.814   0.123
## Carbide  0.672
## Exxon            0.995
## Texaco           0.676
##
##                Factor1 Factor2
## SS loadings      1.722   1.468
## Proportion Var   0.344   0.294
## Cumulative Var   0.344   0.638
##
## $rotmat
##            [,1]       [,2]
## [1,]  1.0234906 -0.1377833
## [2,] -0.1066074  1.0197617
```

Of course, we could also have done it directly, if we had started using "promax"

```
stocks.facp <- factanal(x = stockss, factors = 2,
                        scores = "regression",
                        rotation = "promax",
                        method = "mle")
stocks.facp
```

```
mod1=factanal(x = stockss, factors = 2, scores = "regression",
        rotation = "varimax",     method = "mle")
mod1$loadings
```

```
##
## Loadings:
```

```
##          Factor1 Factor2
## Allied   0.763
## Dupont   0.819   0.232
## Carbide  0.668   0.108
## Exxon    0.113   0.991
## Texaco   0.108   0.677
##
##                Factor1 Factor2
## SS loadings      1.725   1.507
## Proportion Var   0.345   0.301
## Cumulative Var   0.345   0.646
```

```
tcrossprod(solve(mod1$rotmat)) # correlation between rotated factor scores
```

```
##              [,1]         [,2]
## [1,] 1.000000e+00 4.163336e-17
## [2,] 4.163336e-17 1.000000e+00
```

when we use varimax
LL^T ← F_1 and F_2 are uncorrelated

```
mod2=factanal(x = stockss, factors = 2, scores = "regression",
         rotation = "promax",method = "mle")
mod2$loadings
```

when we use promax
then F_1 and F_2 are correlated.

```
##
## Loadings:
##          Factor1 Factor2
## Allied     0.786  -0.104
## Dupont     0.821
## Carbide    0.679
## Exxon              0.997
## Texaco             0.676
##
##                Factor1 Factor2
## SS loadings      1.753   1.471
## Proportion Var   0.351   0.294
## Cumulative Var   0.351   0.645
```

(can use (*) in page 19.

```
tcrossprod(solve(mod2$rotmat)) # correlation between rotated factor scores
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.2787533
## [2,] 0.2787533 1.0000000
```

factor 1 and 2 are correlate
off diagonal ~~elements~~ are not zero.

5. Factor Scores

In Factor Analysis, interest is usually centered on the parameters in the factor model. However, the estimated values of the common factors, called **factor scores**, may also be required. Factor scores are estimates of values for the unobserved random factor vectors $\mathbf{F}_j, j = 1, \cdots, n$.

Estimating Factor Scores via the Weighted Least Squares Method

Suppose that the mean vector μ, the factor loading $\mathbf{L}$ and the specific variance Ψ are known for the factor model

$$\mathbf{X} = \mathbf{LF} + \mu + \epsilon$$

where $\epsilon^T = [\epsilon_1, \cdots, \epsilon_p]$ are the errors, $Var(\epsilon_i) = \psi_i, i = 1, \cdots, p$ need not be equal. The sum of the squares of the errors, weighted by the reciprocal of their variances, is

sum of square $\longrightarrow$ $$\sum_{i=1}^{p} \left(\frac{\epsilon_i^2}{\psi_i}\right) = \epsilon^T \Psi^{-1} \epsilon = (\mathbf{x} - \mu - \mathbf{L}\mathbf{f})^T \Psi^{-1} (\mathbf{x} - \mu - \mathbf{L}\mathbf{f})$$

The estimate $\hat{\mathbf{f}}$ that minimize the weighted sum of the squares of the errors is

weighted least square $\longrightarrow$ $$\hat{\mathbf{f}}_i^{(WLS)} = (\hat{\mathbf{L}}^T \hat{\Psi}^{-1} \hat{\mathbf{L}})^{-1} \hat{\mathbf{L}}^T \hat{\Psi}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})$$

where $\mathbf{x}_i$ is the i^{th} subject's vector of data, $\bar{\mathbf{x}} = \sum_{i=1}^{n} \mathbf{x}_i / n$ is the sample mean.

- Note that if the factor loadings are estimated by the principal component method, then it is typical to use

$$\hat{\mathbf{f}}_i = (\hat{\mathbf{L}}^T \hat{\mathbf{L}})^{-1} \hat{\mathbf{L}}^T (\mathbf{x}_i - \bar{\mathbf{x}})$$

which is the unweighted least squares estimate.

center data.

Estimating Factor Scores via the Regression Method

Using the maximum likelihood method, the joint distribution of $(\mathbf{X} - \mu, \mathbf{F})$ is multivariate normal with mean vector $\mathbf{0}_{p+m}$ and covariance matrix

var(X-μ) var(F)

$$\begin{pmatrix} \mathbf{X} - \mu \\ \mathbf{F} \end{pmatrix} \sim N_{p+m}\left(\mathbf{0}, \Sigma^*_{(m+p)\times(m+p)} = \begin{bmatrix} (\mathbf{L}\mathbf{L}^T + \Psi)_{p\times p} & \mathbf{L}_{p\times m} \\ \mathbf{L}^T_{m\times p} & \mathbf{I}_{m\times m} \end{bmatrix}\right)$$

which implies that the conditional distribution of $\mathbf{F}$ given $\mathbf{X}$ has

orthogal factor model

- $E(\mathbf{F} \mid \mathbf{X}) = \mathbf{L}^T(\mathbf{L}\mathbf{L}^T + \Psi)^{-1}(\mathbf{X} - \mu)$
- $Var(\mathbf{F} \mid \mathbf{X}) = \mathbf{I}_{m\times m} - \mathbf{L}^T(\mathbf{L}\mathbf{L}^T + \Psi)^{-1}\mathbf{L}$

$X = LF + \mu + \epsilon$

Proof $E(F|X) = L^T \Sigma^{-1}(X-\mu) = L^T(LL^T + \Psi)^{-1}(X-\mu)$

$Var(F|X) = I - L^T \Sigma^{-1} L = I - L^T [LL^T + \Psi]^{-1} L$

Assume that $\boldsymbol{\Sigma}, \Psi$ and $(\mathbf{I}_{m\times m} + \mathbf{L}^T\Psi^{-1}\mathbf{L})$ are invertible matrices. The underlying Factor model is $\boldsymbol{\Sigma} = \mathbf{L}^T\mathbf{L} + \Psi$, then the regression estimate of the factor scores have the form

$$\hat{\mathbf{f}}_i^{(REG)} = \hat{\mathbf{L}}^T(\hat{\mathbf{L}}\hat{\mathbf{L}}^T + \hat{\Psi})^{-1}(\mathbf{x}_i - \mu) = (\mathbf{I}_{m\times m} + \hat{\mathbf{L}}^T\hat{\Psi}^{-1}\hat{\mathbf{L}})^{-1}\hat{\mathbf{L}}^T\hat{\Psi}^{-1}(\mathbf{x}_i - \mu)$$

We want to show

$$L^T(LL^T + \psi)^{-1} = (I_{m\times m} + L^T\psi^{-1}L)^{-1} L^T \psi^{-1} \quad (3)$$

By assumption, $\Sigma^{-1} = (LL^T + \psi)^{-1}$

and $(I_{m\times m} + \underbrace{L^T\psi^{-1}L}_{B})^{-1}$ exist

Because $(I_{m\times m} + B)^{-1}(I_{m\times m} + B)(I_{m\times m} + B)^{-1}$

$= (I_{m\times m} + B)^{-1}(I_{m\times m} + B)[I_{m\times m} - (I_{m\times m} + B)^{-1}]$

it follows that

$(LL^T + \psi)^{-1}(LL^T + \psi) - [\psi^{-1} - \psi^{-1}L(B + I_m)^{-1}L^T\psi^{-1}](LL^T + \psi)$

Let $B = L^T\psi^{-1}L$

$(I_{m\times m} + B)(I_{m\times m} + B)^{-1}B - (I_{m\times m} + B)[I_{m\times m} - (I_{m\times m} + B)^{-1}] = B[(I_{m\times m} + B) - I_{m\times m}] = 0$

which implies that $(I_{m\times m} + B)^{-1}B = I_{m\times m} - (I_{m\times m} + B)^{-1}$ (2)

cause $(LL^T+\Psi)$ is invertible we conclude that $(LL^T+\Psi)^{-1} = \Psi^{-1} - \Psi^{-1}L(I_{m\times m}+B)^{-1}L^T\Psi^{-1}$
+(2) $\Rightarrow (LL^T+\Psi)^{-1}L = \Psi^{-1}L(I_{m\times m} + L^T\Psi^{-1}L)^{-1}$
ie desired expression for $\hat{f}_{REG}$ follows from (3) $\Rightarrow$ the equality.

Note that there is a simple relationship between the weighted least squares estimate and the regression estimate

$$\hat{\mathbf{f}}_i^{(WLS)} = (\hat{\mathbf{L}}^T\hat{\Psi}^{-1}\hat{\mathbf{L}})^{-1}(\mathbf{I}_{m\times m} + \hat{\mathbf{L}}^T\hat{\Psi}^{-1}\hat{\mathbf{L}})\hat{\mathbf{f}}_i^{(REG)} = [\mathbf{I}_{m\times m} + (\hat{\mathbf{L}}^T\hat{\Psi}^{-1}\hat{\mathbf{L}})^{-1}]\hat{\mathbf{f}}_i^{(REG)}$$

The relationship between the weighted least squares estimate and the regression estimate implies that

$$\| \hat{\mathbf{f}}_i^{(WLS)} \|^2 > \| \hat{\mathbf{f}}_i^{(REG)} \|^2$$

where $\| \cdot \|$ denotes the Euclidean norm.

Example: Decathlon data

```
library(ade4)
```

```
## Warning: package 'ade4' was built under R version 3.4.4
```

```
data(olympic)
decathlon <- cbind(olympic$tab,olympic$score)
colnames(decathlon) <- c("run100","long.jump","shot","high.jump","run400","hurdle",    "discus","pol
head(decathlon)
```

```
##   run100 long.jump  shot high.jump run400 hurdle discus pole.vaule javelin
## 1  11.25      7.43 15.48      2.27  48.90  15.13  49.28        4.7   61.32
## 2  10.87      7.45 14.97      1.97  47.71  14.46  44.36        5.1   61.76
## 3  11.18      7.44 14.20      1.97  48.29  14.81  43.66        5.2   64.16
## 4  10.62      7.38 15.02      2.03  49.06  14.72  44.80        4.9   64.04
## 5  11.02      7.43 12.92      1.97  47.44  14.40  41.20        5.2   57.46
## 6  10.83      7.72 13.58      2.12  48.34  14.18  43.06        4.9   52.18
##   run1500 score
## 1  268.95  8488
## 2  273.02  8399
## 3  263.20  8328
## 4  285.11  8306
## 5  256.64  8286
## 6  274.07  8272
```

```
# resign running events (so higher score means better performance)
decathlon[,c(1,5,6,10)] <- (-1)*decathlon[,c(1,5,6,10)]
head(decathlon)
```

```
##   run100 long.jump  shot high.jump run400 hurdle discus pole.vaule javelin
## 1 -11.25      7.43 15.48      2.27 -48.90 -15.13  49.28        4.7   61.32
## 2 -10.87      7.45 14.97      1.97 -47.71 -14.46  44.36        5.1   61.76
## 3 -11.18      7.44 14.20      1.97 -48.29 -14.81  43.66        5.2   64.16
## 4 -10.62      7.38 15.02      2.03 -49.06 -14.72  44.80        4.9   64.04
## 5 -11.02      7.43 12.92      1.97 -47.44 -14.40  41.20        5.2   57.46
## 6 -10.83      7.72 13.58      2.12 -48.34 -14.18  43.06        4.9   52.18
##   run1500 score
## 1 -268.95  8488
## 2 -273.02  8399
## 3 -263.20  8328
## 4 -285.11  8306
## 5 -256.64  8286
## 6 -274.07  8272
```

Another way to solve is taking the maximum − the minimum.

```
# check variable scales
apply(decathlon,2,mean)
```

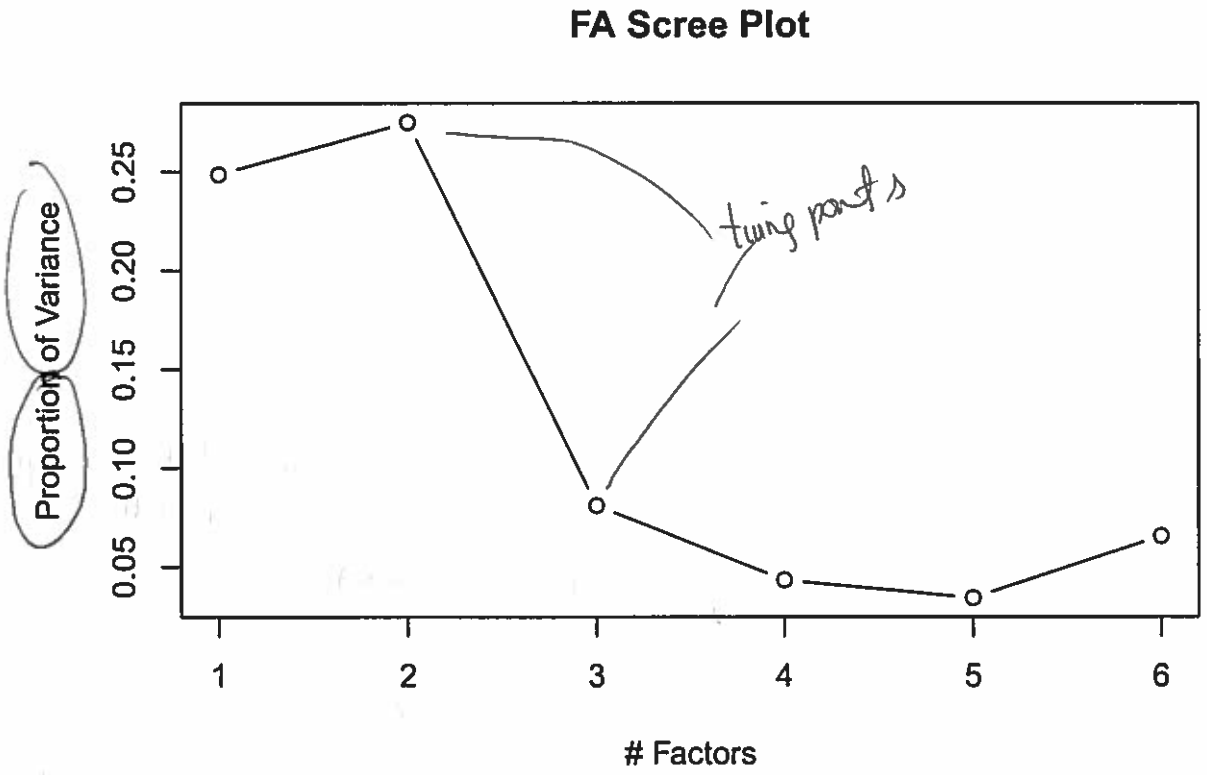
```
##      run100   long.jump        shot   high.jump      run400      hurdle
##  -11.196364    7.133333   13.976364    1.982727  -49.276667  -15.048788
##      discus  pole.vaule     javelin      run1500       score
##   42.353939    4.739394   59.438788 -276.038485 7856.909091
```

```
apply(decathlon,2,sd)
```

```
##      run100   long.jump        shot   high.jump      run400      hurdle
##   0.2433210   0.3043401   1.3319906   0.0939838   1.0696602   0.5067652
##      discus  pole.vaule     javelin      run1500       score
##   3.7191312   0.3344206   5.4959984  13.6570975 415.0694493
```

run long distance has bigger sd.

```
# scree plot for factor analysis results
famods <- vector("list", 6)
for(k in 1:6){
  famods[[k]] <- factanal(x=decathlon[,1:10], factors=k)
}
vafs <- sapply(famods, function(x) sum(x$loadings^2)) / nrow(famods[[1]]$loadings)
vaf.scree <- vafs - c(0, vafs[1:5])
#dev.new(width=10, height=5, noRStudioGD=TRUE)
plot(1:6, vaf.scree, type="b", xlab="# Factors", ylab="Proportion of Variance",
     main="FA Scree Plot")
```

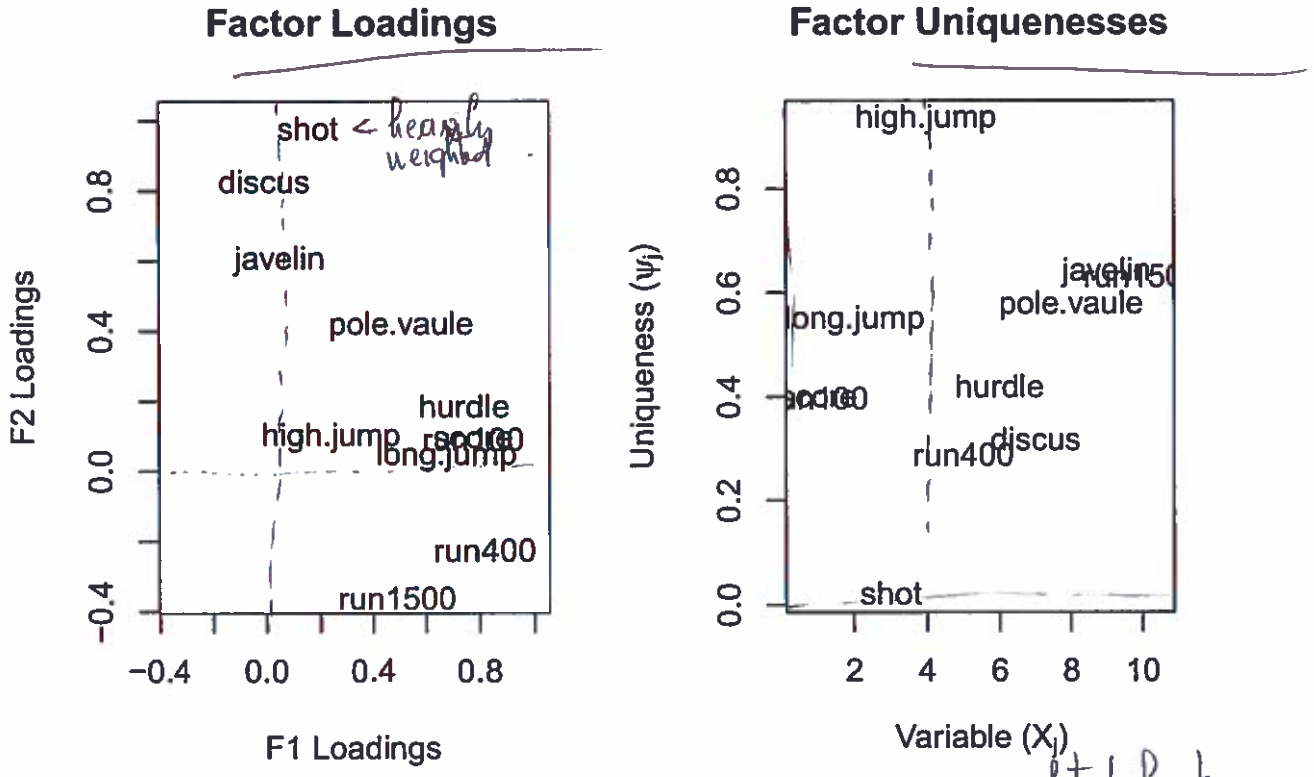



```
# FA on correlation matrix (w/ varimax rotation)
famod <- factanal(x=decathlon[,1:10], factors=2)
names(famod)
```

```
## [1] "converged"    "loadings"     "uniquenesses" "correlation"
```

```
##  [5] "criteria"     "factors"      "dof"          "method"
##  [9] "rotmat"       "STATISTIC"    "PVAL"         "n.obs"
## [13] "call"
```

```
# plot factor loadings and uniquenesses
#dev.new(width=10, height=5, noRStudioGD=TRUE)
par(mfrow=c(1,2))
plot(famod$loadings, xlab="F1 Loadings", ylab="F2 Loadings",
     type="n", main="Factor Loadings", xlim=c(-0.35, 1), ylim=c(-0.35, 1))
text(famod$loadings, labels=colnames(decathlon))
plot(famod$uniquenesses, xlab=expression("Variable ("*X[j]*")"),
     ylab=expression("Uniqueness ("*psi[j]*")"),
     type="n", main="Factor Uniquenesses", xlim=c(0.5,10.5))
text(famod$uniquenesses, labels=colnames(decathlon))
```




```
# refit model and get FA scores (NOT GOOD IDEA!!)
famodW <- factanal(x=decathlon[,1:10], factors=2, scores="Bartlett")
famodR <- factanal(x=decathlon[,1:10], factors=2, scores="regression")
round(cor(famodW$scores, famodR$scores), 4)
```

weighted least square approach.

cor $(\hat{f}_{WLS}, \hat{f}_{REG})$

```
##         Factor1 Factor2
## Factor1  0.9998  0.0000
## Factor2  0.0000  0.9998
```

```
# check correlation of FA scores w/ overall decathlon score
round(cor(decathlon$score, famodR$scores), 4)
```

overall score

correlation of Y and FL.

$(X_1, \ldots, X_{10}, Y)$ ① $Y \approx X_1 + \cdots + X_{10}$

② $X = LF + \mu + \epsilon$

→ 2 factor model

F_1

F_2

$Y \approx F_1 + F_2$

reduce from 10th 2

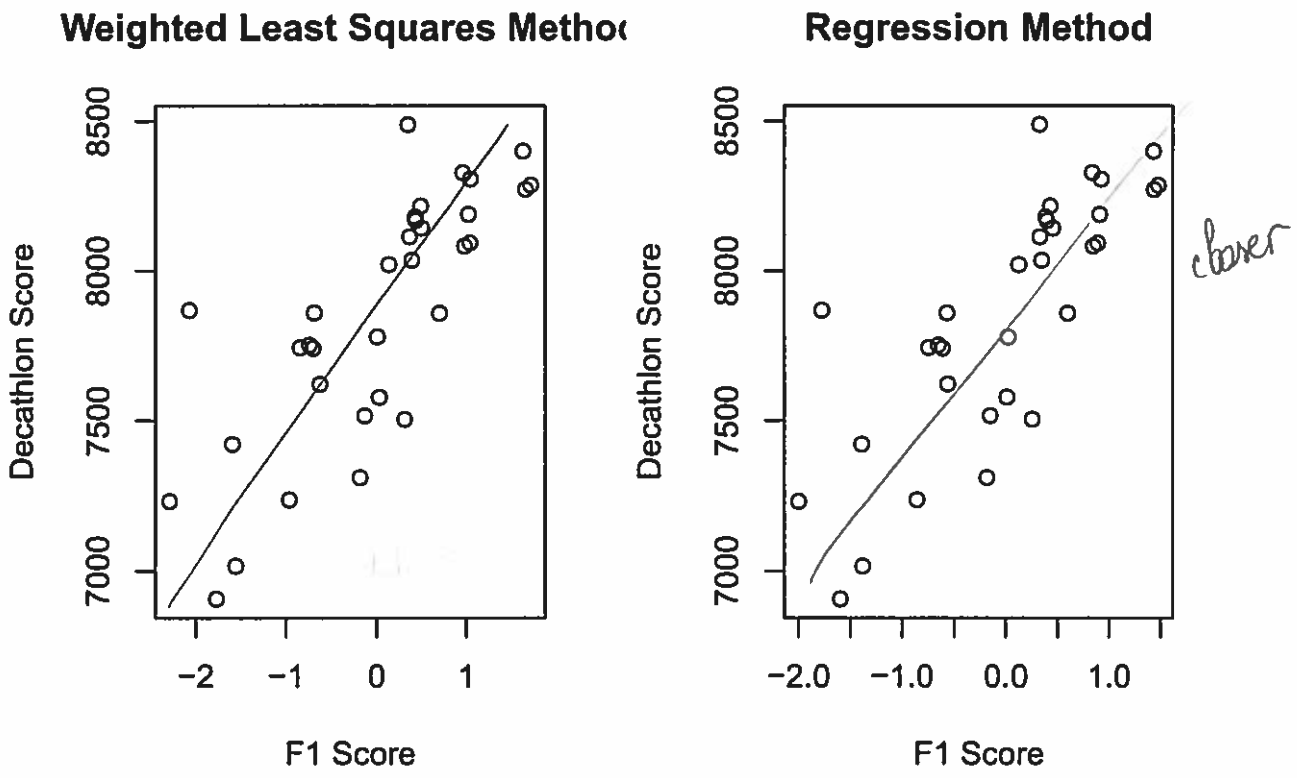
```
##      Factor1 Factor2
## [1,]  0.7952  0.5022
```

```
round(cor(decathlon$score, famodW$scores), 4)
```

$r^2 = \frac{SSR}{SST}$

```
##      Factor1 Factor2
## [1,]  0.7854  0.4865
```

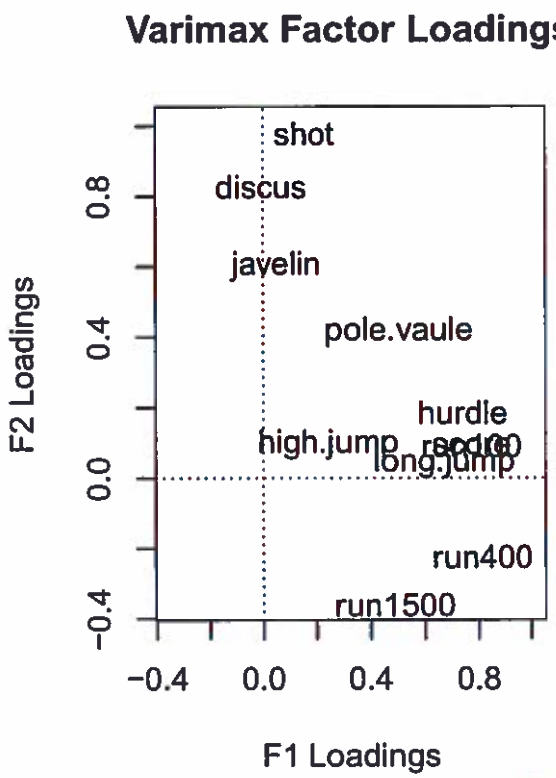
```
#dev.new(width=10, height=5, noRStudioGD=TRUE)
par(mfrow=c(1,2))
plot(famodW$scores[,1], decathlon$score, xlab="F1 Score",
     ylab="Decathlon Score", main="Weighted Least Squares Method")
plot(famodR$scores[,1], decathlon$score, xlab="F1 Score",
     ylab="Decathlon Score", main="Regression Method")
```

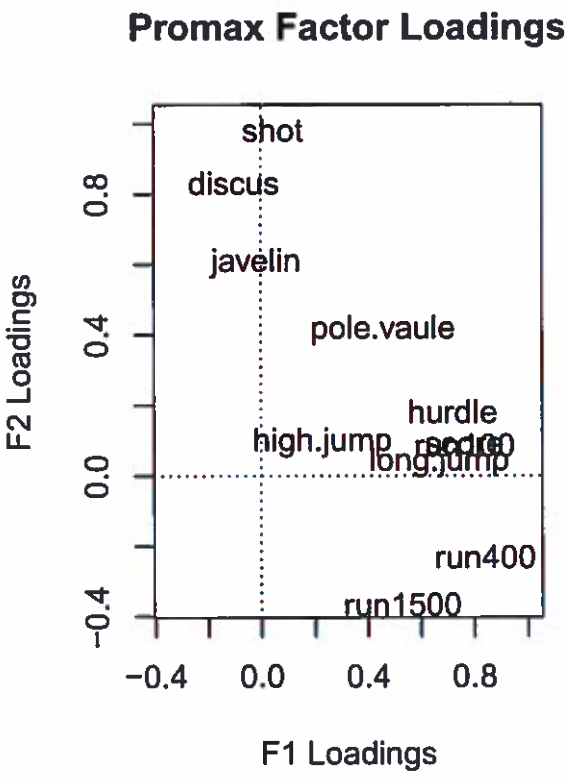


clover

```
# try promax rotation
famod.promax <- promax(famod$loadings)
#dev.new(width=10, height=5, noRStudioGD=TRUE)
par(mfrow=c(1,2))
plot(famod$loadings, xlab="F1 Loadings", ylab="F2 Loadings",
     type="n", main="Varimax Factor Loadings", xlim=c(-0.35, 1), ylim=c(-0.35, 1))
text(famod$loadings, labels=colnames(decathlon))
abline(0,0,lty=3)
abline(v=0,lty=3)
plot(famod.promax$loadings, xlab="F1 Loadings", ylab="F2 Loadings",
     type="n", main="Promax Factor Loadings", xlim=c(-0.35, 1), ylim=c(-0.35, 1))
text(famod.promax$loadings, labels=colnames(decathlon))
abline(0,0,lty=3)
abline(v=0,lty=3)
```

prefer uncorrated version






```
# compare loadings
oldFAloadings <- famod$loadings
newFAloadings <- famod$loadings %*% famod.promax$rotmat
sum((newFAloadings - famod.promax$loadings)^2)
```

```
## [1] 1.55904e-31
```

```
# compare reproduced data before and after rotation
oldFAscores <- famodR$scores
newFAscores <- oldFAscores %*% t(solve(famod.promax$rotmat))
Xold <- tcrossprod(oldFAscores, oldFAloadings)
Xnew <- tcrossprod(newFAscores, newFAloadings)
sum((Xold - Xnew)^2)
```

```
## [1] 4.431144e-30
```

```
# population and sample factor score covariance matrix (after rotation)
tcrossprod(solve(famod.promax$rotmat))    # population
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.1182015
## [2,] 0.1182015 1.0000000
```

```
cor(newFAscores)                          # sample
```

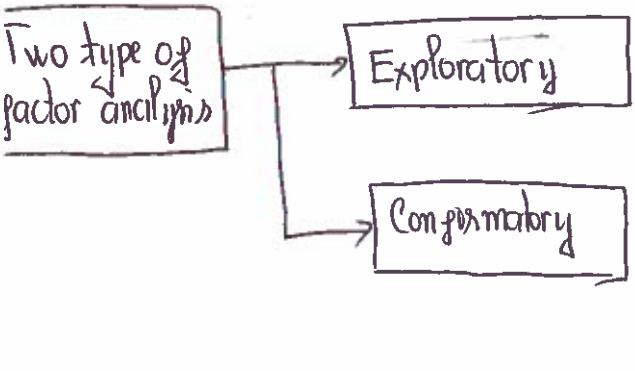
```
##           [,1]      [,2]
## [1,] 1.0000000 0.1320303
## [2,] 0.1320303 1.0000000
```

6. Comparison of PCA and FA (double check)

There are many parallels between Principal Component Analysis and Factor Analysis, but there are also important differences. The common goal is to represent the data in a smaller number of simpler components, but the two methods are based on different strategies and models to achieve these goals. We list some of the differences between the two approaches.

Table 1: Comparison of PCA and FA

	PCA	FA
Aim	Fewer and simpler components	Fewer factor loadings
Model	Not explicit	Strick k-factor model $\mathbf{X} = \mu + \mathbf{LF} + \epsilon$
Covariance matrix	$\mathbf{\Sigma} = \boldsymbol{P\Lambda P}^T$	factor $\mathbf{\Sigma} = \boldsymbol{LL}^T + \mathbf{\Psi}$
Solution method	Singular value decomposition $\mathbf{X} = U\Gamma V^T$; Spectral decomposition $\mathbf{\Sigma} = \mathbf{P\Lambda P}^T$?	PC-based loadings or ML-based loadings with orthogonal or oblique rotations
Scores	Projection onto eigenvectors ranked by variance; uniquely defined uncorrelated component scores	PC-based scores or ML regression scores; not unique, include rotations; rotated scores correlated
Data description	Approximate $\mathbf{X}$ $Y_1 = \mathbf{a}_1^T\mathbf{X}, Y_2 = \mathbf{a}_2^T\mathbf{X}, \cdots, Y_q = \mathbf{a}_q^T\mathbf{X}$	Complete, includes specific factor $\mathbf{X} - \mu = \mathbf{LF} + \epsilon$



← It is exploratory when you do not have a pre-defined idea of the structure or how many dimensions are in the set of variable

← helps confirm when we want to test about the structure of the # of dimension underlying a set of variables.

Chapter 6. Canonical Correlation

Jianxuan Liu

Fall 2018

Canonical Correlation Analysis (CCA) seeks to identify and quantify the associations between two sets of variables. CCA connects two sets of variables by finding linear combinations of variables that maximally correlate. The idea is first to determine the pair of linear combinations having the largest correlation. Next, we determine the pair of linear combinations having the largest correlation among all pairs uncorrelated with the initially selected pair, and so on. The pairs of linear combinations are called **canonical variables** and their correlations are called **canonical correlations**

There are two typical purposes of CCA:

- Data reduction: explain covariation between two sets of variables using small number of linear combinations
- Data interpretation: find features (i.e., canonical variates) that are important for explaining covariation between sets of variables

1. Canonical variables and canonical correlations

The first set, of p variable, is represented by the $p \times 1$ random vectors $\mathbf{X} = (X_1, X_2, \cdots, X_p)^T$. The second group, of q variables, is represented by the $q \times 1$ random vector $\mathbf{Y} = (Y_1, Y_2, \cdots, Y_q)^T$. For the random vectors $\mathbf{X}$ and $\mathbf{Y}$, let

$$E(\mathbf{X}) = \boldsymbol{\mu}_X, E(\mathbf{Y}) = \boldsymbol{\mu}_Y$$

$$Var(\mathbf{X}) = \boldsymbol{\Sigma}_X, Var(\mathbf{Y}) = \boldsymbol{\Sigma}_Y$$

$$Cov(\mathbf{X}, \mathbf{Y}) = E[(\mathbf{X} - \boldsymbol{\mu}_X)(\mathbf{Y} - \boldsymbol{\mu}_Y)^T] = \boldsymbol{\Sigma}_{XY}$$

- Define new variables U and V via linear combinations of $\mathbf{X}$ and $\mathbf{Y}$

canonical variable $\rightarrow$ $U = \mathbf{a}^T\mathbf{X} = a_1X_1 + a_2X_2 \cdots$ $V = \mathbf{b}^T\mathbf{Y} =$

for some pair of coefficient vectors $\mathbf{a}$ and $\mathbf{b}$. Then U and V have the following properties

$$Var(U) = \mathbf{a}^T\boldsymbol{\Sigma}_X\mathbf{a}, Var(V) = \mathbf{b}^T\boldsymbol{\Sigma}_Y\mathbf{b}, Cov(U, V) = \mathbf{a}^T\boldsymbol{\Sigma}_{XY}\mathbf{b}$$

CCA seek coefficient vectors $\mathbf{a}$ and $\mathbf{b}$ such that

$$Cor(U, V) = \frac{Cov(U, V)}{\sqrt{Var(U)Var(V)}}.$$

is as large as possible, subject to U and V having unit variance.

- The first pair of canonical variables is the pair of linear combinations of U_1 and V_1 having unit variance, which maximize the correlation $Cor(U, V)$.
- The second pair of canonical variables is the pair of linear combinations of U_2 and V_2 having unit variance, which maximize the correlation $Cor(U, V)$ among all choices that are uncorrelated with the first pair of canonical variables (U_1, V_1)
- The k^{th} pair of canonical variables is the pair of linear combinations of U_k and V_k having unit variance, which maximize the correlation $Cor(U, V)$ among all choices that are uncorrelated with the previous $k - 1$ pairs of canonical variable pairs.

The correlation between the k^{th} pair of canonical variables is called the k^{th} canonical correlation.

The k^{th} pair of canonical variates is given by

$$U_k = \mathbf{u}_k^T \Sigma_X^{-1/2} \mathbf{X}, \quad V_k = \mathbf{v}_k \Sigma_Y^{-1/2} \mathbf{Y}$$
$$= \mathbf{a}_k^T \mathbf{X} \qquad = \mathbf{b}_k^T \mathbf{Y}$$

where $\mathbf{a}_k^T = \mathbf{u}_k^T \Sigma_X^{-1/2}, \mathbf{b}_k^T = \mathbf{v}_k \Sigma_Y^{-1/2}$.

- $\mathbf{u}_k$ is the k^{th} eigenvector of $\Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \Sigma_X^{-1/2}$
- $\mathbf{v}_k$ is the k^{th} eigenvector of $\Sigma_Y^{-1/2} \Sigma_{YX} \Sigma_X^{-1} \Sigma_{XY} \Sigma_Y^{-1/2}$

The k^{th} **canonical correlation** is given by

square here →

$$Cor(U_k, V_k) = \rho_k$$

where ρ_k^2 is the k^{th} eigenvalue of $\Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \Sigma_X^{-1/2}$ or $\Sigma_Y^{-1/2} \Sigma_{YX} \Sigma_X^{-1} \Sigma_{XY} \Sigma_Y^{-1/2}$.

assume $X \sim \mathbb{R}^p$ $Y \sim \mathbb{R}^q$

$\sim (\mu_X, \sigma_X)$ $Y \sim (\mu_Y, \sigma_Y)$

$cov(X, Y) = \Sigma_{XY}$

et $u = a^T X$ $v = b^T Y$

$r(u, v) = \rho = \dfrac{a^T \Sigma_{XY} b}{\sqrt{a^T \Sigma_X a * b^T \Sigma_Y b}}$

efine $c = \Sigma_X^{1/2} a \Rightarrow a = \Sigma^{1/2} c$ $a^T = c^T \Sigma^{1/2}$

$d = \Sigma_Y^{1/2} b$

$= \dfrac{c^T \Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} d}{\|c\| \|d\|}$ $\|\cdot\|$: Euclidean norm

$c^T \Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1/2} d \| \leq \|c^T \Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1/2}\| \|d\|$

$^2 = \dfrac{c^T \Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \Sigma_X^{-1/2} c}{c^T c}$

ie maximum occurs when c is the eigen vector of corresponding to the largest eigenvalue of

$^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \Sigma_X^{-1/2}$

he maximized correlation is the largest eigenvalue of $\Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \Sigma_X^{-1/2}$

(ρ, a) are (eigenvalue, eigenvector) of $\Sigma_X^{-1}, \Sigma_{XY}, \Sigma$

Use this.

Alternatively, the canonical variables coeffcient vectors a and b and the corresponding correlations can be found by solving the eigenvalue equations

$$\Sigma_X^{-1}\Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}\mathbf{a} = \rho^2\mathbf{a}$$

$$\Sigma_Y^{-1}\Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY}\mathbf{b} = \rho^2\mathbf{b}$$

This is often more practical to use when computing the coeffcients and the correlations.

The canonical variables have the following properties:

- $Var(U_k) = Var(V_k) = 1$
- $Cov(U_k, U_l) = Cor(U_k, U_l) = 0, k \neq l$
- $Cov(V_k, V_l) = Cor(V_k, V_l) = 0, k \neq l$
- $Cov(U_k, V_l) = Cor(U_k, V_l) = 0, k \neq l$

Let $\mathcal{U} = \underset{\sim}{A}^T \underset{\sim}{X}$. Show that $cov(\mathcal{U}) = A^T \Sigma_x A = I$

$A = \underset{\sim}{e}^T \Sigma^{-1/2} = \underset{\sim}{e}^T P_x \Lambda_x P$ where e is an orthogonal mat

$\Sigma_x = P_x \Lambda_x P_x^T$

$P_x^T X$ is the set of all pc derived from $\underset{\sim}{X}$ alone

matrix $\Lambda_x^{-1/2} P_x^T \underset{\sim}{X}$ has the i^{th} row $\frac{1}{\sqrt{\lambda_i}} P_i^T \underset{\sim}{X}$ which is the i^{th} pc scaled to have unit variance

$cov(\Lambda_x^{-1/2} P_x^T \underset{\sim}{X}) = \Lambda_x^{-1/2} P_x^T \Sigma_x P_x \Lambda^{-1/2}$

$= \Lambda_x^{-1/2} P_x^T (P_x \Lambda_x P_x^T) P_x \Lambda^{-1/2}$

$= \Lambda_x^{-1/2} \Lambda_x \Lambda_x^{-1/2} = I$

Show $cov(V) = E(VV^T) = E($

$= E(f^T \Sigma_Y^{-1/2} (Y - \mu_Y)(Y - \mu_Y)^T \Sigma_Y^{-1/2} f)$

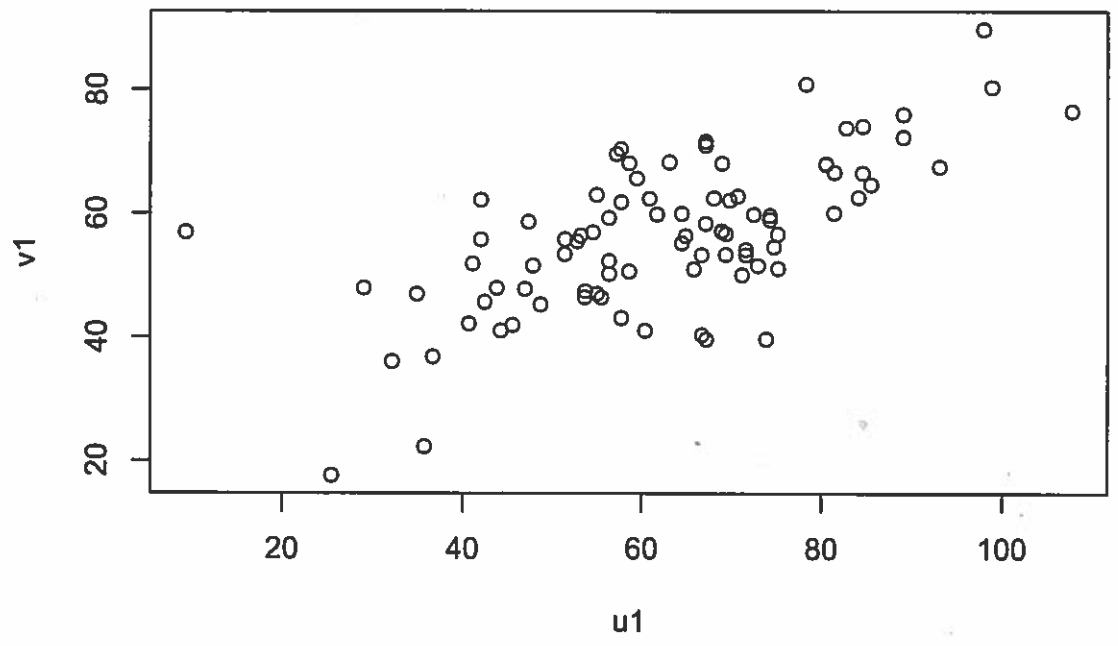
$= f^T \Sigma_Y^{-1/2} \Sigma_Y \Sigma_Y^{-1/2} f$

$= I$ since $cov(Y) = E(YY^T) - \mu\mu^T =$

Example: Open/closed book exams

Input: Σ or raw data

```
library(bootstrap)
data(scor) # first two are closed book, last three are open book
S=cov(scor)
S11=S[1:2,1:2]; S22=S[3:5,3:5]; S12=S[1:2,3:5]; S21=S[3:5,1:2]
e1=eigen(solve(S11)%*%S12%*%solve(S22)%*%S21)
e2=eigen(solve(S22)%*%S21%*%solve(S11)%*%S12)
# canonical vectors for 2 closed book exams
a=-e1$vectors[,1]
# canonical vectors for 3 open book exams
b=e2$vectors[,1]
# (a,b) is the first pair

u1=a%*%t(scor[,1:2])
v1=b%*%t(scor[,3:5])
plot(u1,v1)
```



```
cor(t(rbind(u1,v1)))
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.6630521
## [2,] 0.6630521 1.0000000
```

```
sqrt(e1$values[1])  #cor(u1, v1)
```

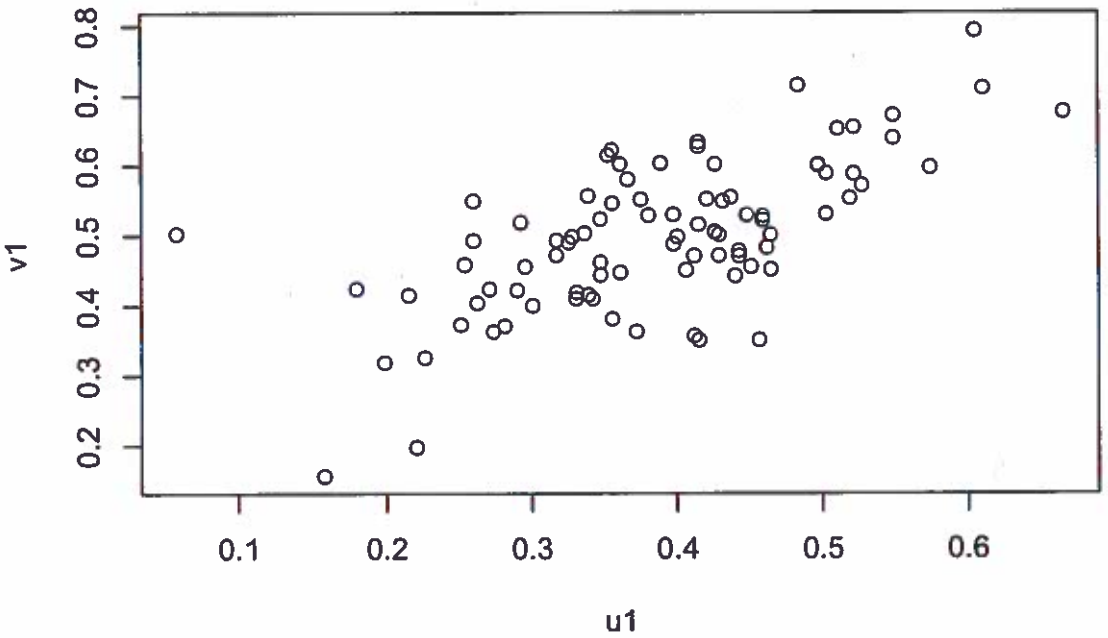
```
## [1] 0.6630521
```

```
sqrt(e2$values[1])
```

```
## [1] 0.6630521
```

use build in function

```
## use build-in function cancor
u1=cancor(scor[,1:2],scor[,3:5])$xcoef[,1]%*%t(scor[,1:2])
v1=cancor(scor[,1:2],scor[,3:5])$ycoef[,1]%*%t(scor[,3:5])
plot(u1,v1)
```




```
cor(t(rbind(u1,v1)))
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.6630521
## [2,] 0.6630521 1.0000000
```

```
sqrt(e2$values[1])
```

```
## [1] 0.6630521
```

```
## cor(u1,u2) , cor(v1,v2)
u2=-e1$vectors[,2]%*%t(scor[,1:2])
cor(t(rbind(u1,u2))) #cor(u1, u2)
```

```
##               [,1]          [,2]
## [1,]  1.000000e+00 -3.700511e-18
## [2,] -3.700511e-18  1.000000e+00
```

```
v2=e2$vectors[,2]%*%t(scor[,3:5])
cor(t(rbind(v1,v2))) #cor(v1, v2)

##              [,1]         [,2]
## [1,] 1.000000e+00 3.490312e-16
## [2,] 3.490312e-16 1.000000e+00
cor(t(rbind(u1,v2))) #cor(u1, v2)

##              [,1]         [,2]
## [1,] 1.000000e+00 2.512154e-16
## [2,] 2.512154e-16 1.000000e+00
```

Covariance of Original and Canonical Variables

Let $\mathbf{U} = \mathbf{A}^T\mathbf{X}$ and $\mathbf{V} = \mathbf{B}^T\mathbf{Y}$ where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_p]$ and $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_q]$. Then

- $\mathbf{U} = (U_1, \cdots, U_p)^T$ contains the p canonical variates from $\mathbf{X}$.
- $\mathbf{V} = (V_1, \cdots, V_q)^T$ contains the q canonical variates from Y.

The canonical variates and original variables have covariance matrices

$$Cov(\mathbf{U}, \mathbf{X}) = Cov(\mathbf{A}^T\mathbf{X}, \mathbf{X}) = \mathbf{A}^T\mathbf{\Sigma}_{\mathbf{X}}$$

$$Cov(\mathbf{U}, \mathbf{Y}) = Cov(\mathbf{A}^T\mathbf{X}, \mathbf{Y}) = \mathbf{A}^T\mathbf{\Sigma}_{\mathbf{XY}}$$

$$Cov(\mathbf{V}, \mathbf{X}) = Cov(\mathbf{B}^T\mathbf{Y}, \mathbf{X}) = \mathbf{B}^T\mathbf{\Sigma}_{\mathbf{YX}}$$

$$Cov(\mathbf{V}, \mathbf{Y}) = Cov(\mathbf{B}^T\mathbf{Y}, \mathbf{Y}) = \mathbf{B}^T\mathbf{\Sigma}_{\mathbf{Y}}$$

The canonical variates and original variables have correlation matrices

$$Cor(\mathbf{U}, \mathbf{X}) = Cov(\mathbf{A}^T\mathbf{X}, \tilde{\mathbf{\Sigma}}_{\mathbf{X}}^{-1/2}\mathbf{X}) = \mathbf{A}^T\mathbf{\Sigma}_{\mathbf{X}}\tilde{\mathbf{\Sigma}}_{\mathbf{X}}^{-1/2}$$

$$Cor(\mathbf{U}, \mathbf{Y}) = Cov(\mathbf{A}^T\mathbf{X}, \tilde{\mathbf{\Sigma}}_{\mathbf{Y}}^{-1/2}\mathbf{Y}) = \mathbf{A}^T\mathbf{\Sigma}_{\mathbf{XY}}\tilde{\mathbf{\Sigma}}_{\mathbf{Y}}^{-1/2}$$

$$Cor(\mathbf{V}, \mathbf{X}) = Cov(\mathbf{B}^T\mathbf{Y}, \tilde{\mathbf{\Sigma}}_{\mathbf{X}}^{-1/2}\mathbf{X}) = \mathbf{B}^T\mathbf{\Sigma}_{\mathbf{YX}}\tilde{\mathbf{\Sigma}}_{\mathbf{X}}^{-1/2}$$

$$Cor(\mathbf{V}, \mathbf{Y}) = Cov(\mathbf{B}^T\mathbf{Y}, \tilde{\mathbf{\Sigma}}_{\mathbf{Y}}^{-1/2}\mathbf{Y}) = \mathbf{B}^T\mathbf{\Sigma}_{\mathbf{Y}}\tilde{\mathbf{\Sigma}}_{\mathbf{Y}}^{-1/2}$$

given that $Var(U_k) = Var(V_l) = 1$ for all k, l. Here $\tilde{\mathbf{\Sigma}}_{\mathbf{X}}^{-1/2}$ is a diagonal matrix containing $\mathbf{X}$ variances and $\tilde{\mathbf{\Sigma}}_{\mathbf{Y}}^{-1/2}$ is a diagonal matrix containing $\mathbf{Y}$ variances.

3. Canonical Variables Interpretation

Canonical variables are, in general, artificial. They have no physical meaning. the canonical coefficients $\mathbf{a}$ and $\mathbf{b}$ have units proportional to the original variables $\mathbf{X}$ and $\mathbf{Y}$. It is common to standardize all the variables before performing the canonical correlation analysis. We may interpret the coefficients of the canonical variables similarly to how we interpret the coefficients of principal components.

For checking the condition $\Sigma_{XY} \neq 0$, since the canonical correlation can only be applied when $\Sigma_{XY} \neq 0$.

4. Large Sample Inferences

When $\Sigma_{XY} = 0$, $\mathbf{a}^T\mathbf{X}$ and $\mathbf{b}^T\mathbf{Y}$ have covariance $\mathbf{a}^T\Sigma_{XY}\mathbf{b} = 0$ for all vectors $\mathbf{a}$ and $\mathbf{b}$. Consequently, all the canonical correlations must be zero and there is no point in pursuing a canonical correlation analysis. To test

$$H_0 : \Sigma_{XY} = 0_{p\times q} \text{ (All canonical correlations are zero)}$$

$$H_1 : \Sigma_{XY} \neq 0_{p\times q} \text{ (At least one canonical correlation significantly differs from zero)}$$

Let

$$\mathbf{Z}_j = [\mathbf{X}_j, \mathbf{Y}_j]^T, j = 1, 2, \cdots, n$$

be a random sample from an $N_{p+q}(\mu, \Sigma)$ population with

$$\Sigma = \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix}.$$

Then the likelihood ratio test of $H_0 : \Sigma_{XY} = 0_{p\times q}$ versus $H_1 : \Sigma_{XY} \neq 0_{p\times q}$ rejects H_0 for large values of

$$-2\ln\Lambda = n\ln\left(\frac{|\mathbf{S}_X||\mathbf{S}_Y|}{|\mathbf{S}|}\right) = -n\ln\prod_{i=1}^{p}(1-\hat{\rho}_i^{*2})$$

where $\mathbf{S}$ is the unbiased estimator of Σ. For $n \to \infty$, the test statistic is approximately distributed as a χ^2_{pq}. Bartlett suggests replacing the multiplicative factor n in the likelihood ratio statistic with the factor $-\left[n-1-\frac{1}{2}(p+q+1)\ln\prod_{i=1}^{p}(1-\hat{\rho}_i^{*2})\right]$ to improve the χ^2 approximation to the sampling distribution of $-2\ln\Lambda$. Thus, reject H_0 at significance level α if

$$-[n-1-\frac{1}{2}(p+q+1)]\ln\prod_{i=1}^{p}(1-\hat{\rho}_i^{*2}) > \chi^2_{pq}(\alpha)$$

Not neccessary to have the raw data.

Example: The Los Angeles Depression Study Data We don't have the raw data

Canonical correlation analysis using a sample correlation matrix rather than the raw data matrix.

Data include n=294 individuals with variables

$\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$

X are "health variables":

- CESD: A numerical measure of depression
- Health: A measure of general perceived health status

Y are "personal (~demographic) variables":

- Gender: Low=Male, High=Female
- Age
- Income
- Education Level

use formulas in page 3 to compo

+ From covariance matrix

⇒ compute a, b, ρ

→ $\underset{\sim}{u} = a^T X$

$\underset{\sim}{v} = b^T Y$

Suppose the sample correlation matrix **R** is given

```
R22 <- matrix( c(
1,.044,-.106,-.18,
.044,1,-.208,-.192,
-.106,-.208,1,.492,
-.18,-.192,.492,1 ),
ncol=4,byrow=T)
```

```
R11 <- matrix( c(
1,.212,.212,1),
ncol=2,byrow=T)

R12 <- matrix( c(
.124,-.164,-.101,-.158,
.098,.308,-.27,-.183),
ncol=4, byrow=T)

R21 <- t(R12)
```

because from page 3

Finding the E1 and E2 matrices:

contain (ρ_1, a)

contain (ρ_1, b)

```
E1 <- solve(R11) %*% R12 %*% solve(R22) %*% R21
E2 <- solve(R22) %*% R21 %*% solve(R11) %*% R12
eigen(E1)

## eigen() decomposition
## $values
## [1] 0.16763669 0.06806171
##
## $vectors
##            [,1]       [,2]
## [1,]  0.4560253  0.9476307
## [2,] -0.8899668  0.3193681
eigen(E2)

## eigen() decomposition
## $values
## [1] 1.676367e-01+0.000000e+00i 6.806171e-02+0.000000e+00i
## [3] 3.483919e-19+6.105072e-18i 3.483919e-19-6.105072e-18i
##
## $vectors
##                [,1]                [,2]                [,3]
## [1,]  0.02473006+0i  0.4522578+0i 0.6926400+0.0000000i
## [2,]  0.90283535+0i -0.4598545+0i 0.2071287-0.1099990i
## [3,] -0.40965230+0i -0.4745706+0i 0.4675316-0.3579553i
## [4,]  0.12830336+0i -0.5989820+0i 0.0297314+0.3429957i
##                [,4]
## [1,] 0.6926400+0.0000000i
## [2,] 0.2071287+0.1099990i
## [3,] 0.4675316+0.3579553i
## [4,] 0.0297314-0.3429957i
```

$a_1 = [0.46 \quad -0.89]$

$\Sigma_X^{-1} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX}$

← complex

The canonical correlations are:

$\rho_1 = \gamma_1$

```
canon.corr <- sqrt(eigen(E1)$values)
canon.corr

## [1] 0.4094346 0.2608864
canon.corr <- sqrt(eigen(E2)$values)
canon.corr

## [1] 4.094346e-01+0.00000e+00i 2.608864e-01+0.00000e+00i
## [3] 1.797693e-09+1.69803e-09i 1.797693e-09-1.69803e-09i
```

$\mathrm{Cor}(U_k, V_l) = \rho_l$ which are the eigen value of E_1 the same.

First canonical variate:

$$u_1 = 0.46CESD - 0.89Health \quad = \vec{a}_1 X$$

$$v_1 = 0.02Gender + 0.90Age - 0.41Education + 0.13Income \quad = b_1 Y$$

Second canonical variate:

$$u_2 = -0.95CESD - 0.32Health \quad = a_2 X$$

$$v_2 = -0.45Gender + 0.46Age + 0.47Education + 0.60Income \quad b_2 Y$$ (* flip the order of b_2)

Test for the significance of the first canonical correlation: The null hypothesis is that the first (and smaller) canonical correlations are zero.

part 4 page 6

```
n <- 294;
p <- 2;
q <- 4
test.stat <- -( (n-1) - 0.5*(p+q+1) ) *sum(log(1-eigen(E1)$values))
test.stat
```

```
## [1] 73.52575
```

degree $= p \times q = 2 \times 4$

```
P.value <- pchisq(test.stat, df = p*q, lower.tail=F)
P.value
```

```
## [1] 9.728813e-13
```

Reject H_0

The first pair has non zero correlation.

Since the P-value is small, we conclude that there is at least one nonzero canonical correlation.

Test for the significance of the second canonical correlation: The null hypothesis is that the second (and smaller) canonical correlations are zero in general, but there's only two here.

```
test.stat <- -( (n-1) - 0.5*(p+q+1) ) * sum(log(1-eigen(E1)$values[-1]))
test.stat
```

exclude the first eigen value

```
## [1] 20.40647
```

```
P.value <- pchisq(test.stat, df = (p-1)*(q-1), lower.tail=F)
P.value
```

because exclude 1st pair.

```
## [1] 0.0001398029
```

The P-value is again very small, so we conclude there are at least two nonzero canonical correlations. In this case, that means exactly two nonzero canonical correlations.

both 2 are useful

Example: Decathlon Example This example: Input raw data (value of X & Y)

```
library(ade4)
```

```
## Warning: package 'ade4' was built under R version 3.4.4
```

```
data(olympic)
decathlon <- cbind(olympic$tab,olympic$score)
colnames(decathlon) <- c("run100","long.jump","shot","high.jump","run400","hurdle",
    "discus","pole.vault","javelin","run1500","score")
# resign running events (so higher score means better performance)
#head(decathlon)
decathlon[,c(1,5,6,10)] <- (-1)*decathlon[,c(1,5,6,10)]
#head(decathlon)
```

to standardize data so that larger → better performance

100 longjump shot highjump 400 hurdle discus pole.vault javelin 1500

run500 score.

```
## 1  268.95  8488
## 2  273.02  8399
## 3  263.20  8328
## 4  285.11  8306
## 5  256.64  8286
## 6  274.07  8272
```

```
decathlon[,c(1,5,6,10)] <- (-1)*decathlon[,c(1,5,6,10)]
head(decathlon)
```

```
##   run100 long.jump  shot high.jump run400 hurdle discus pole.vault javelin
## 1 -11.25      7.43 15.48      2.27 -48.90 -15.13  49.28        4.7   61.32
## 2 -10.87      7.45 14.97      1.97 -47.71 -14.46  44.36        5.1   61.76
## 3 -11.18      7.44 14.20      1.97 -48.29 -14.81  43.66        5.2   64.16
## 4 -10.62      7.38 15.02      2.03 -49.06 -14.72  44.80        4.9   64.04
## 5 -11.02      7.43 12.92      1.97 -47.44 -14.40  41.20        5.2   57.46
## 6 -10.83      7.72 13.58      2.12 -48.34 -14.18  43.06        4.9   52.18
##   run1500 score
## 1 -268.95  8488
## 2 -273.02  8399
## 3 -263.20  8328
## 4 -285.11  8306
## 5 -256.64  8286
## 6 -274.07  8272
```

```
# check variable scales
apply(decathlon,2,mean)
```

```
##      run100   long.jump        shot   high.jump      run400      hurdle
##  -11.223529    7.095000   13.850882    1.974412  -49.366176  -15.107647
##      discus  pole.vault     javelin      run1500       score
##   41.905294    4.676471   58.840588 -276.191471 7782.852941
```

check the center tendency

```
apply(decathlon,2,sd)
```

```
##      run100   long.jump        shot   high.jump      run400      hurdle
##   0.2872322   0.3738680   1.5019268   0.1044811   1.1755463   0.6056555
##      discus  pole.vault     javelin      run1500       score
##   4.5007105   0.4930172   6.4387360  13.4781327 594.5827227
```

check the spread out tendency

```
# separate into running/jumping vs throwing/arm events
X <- as.matrix(decathlon[,c("shot","discus","javelin","pole.vault")])
Y <- as.matrix(decathlon[,c("run100","run400","run1500",
                            "hurdle","long.jump","high.jump")])
n <- nrow(X)
p <- ncol(X)
q <- ncol(Y)
```

CCA via Covariance

Build in function

```
# canonical correlations of covariance (unstandardized data)
cca <- cancor(X, Y)
```

Manually

```
# cca (the normal way)
Sx <- cov(X)
```

```
Sy <- cov(Y)
Sxy <- cov(X,Y)
Sxeig <- eigen(Sx, symmetric=TRUE)
Sxisqrt <- Sxeig$vectors %*% diag(1/sqrt(Sxeig$values)) %*% t(Sxeig$vectors)
Syeig <- eigen(Sy, symmetric=TRUE)
Syisqrt <- Syeig$vectors %*% diag(1/sqrt(Syeig$values)) %*% t(Syeig$vectors)
Xmat <- Sxisqrt %*% Sxy %*% solve(Sy) %*% t(Sxy) %*% Sxisqrt
Ymat <- Syisqrt %*% t(Sxy) %*% solve(Sx) %*% Sxy %*% Syisqrt
Xeig <- eigen(Xmat, symmetric=TRUE)
Yeig <- eigen(Ymat, symmetric=TRUE)

# compare correlations (same)
cca$cor
```

$X_{mat} = \Sigma_X^{-1/2} \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \Sigma_X^{-1/2}$

```
## [1] 0.7702006 0.5033532 0.4184145 0.3052556
rho <- sqrt(Xeig$values)
rho
```

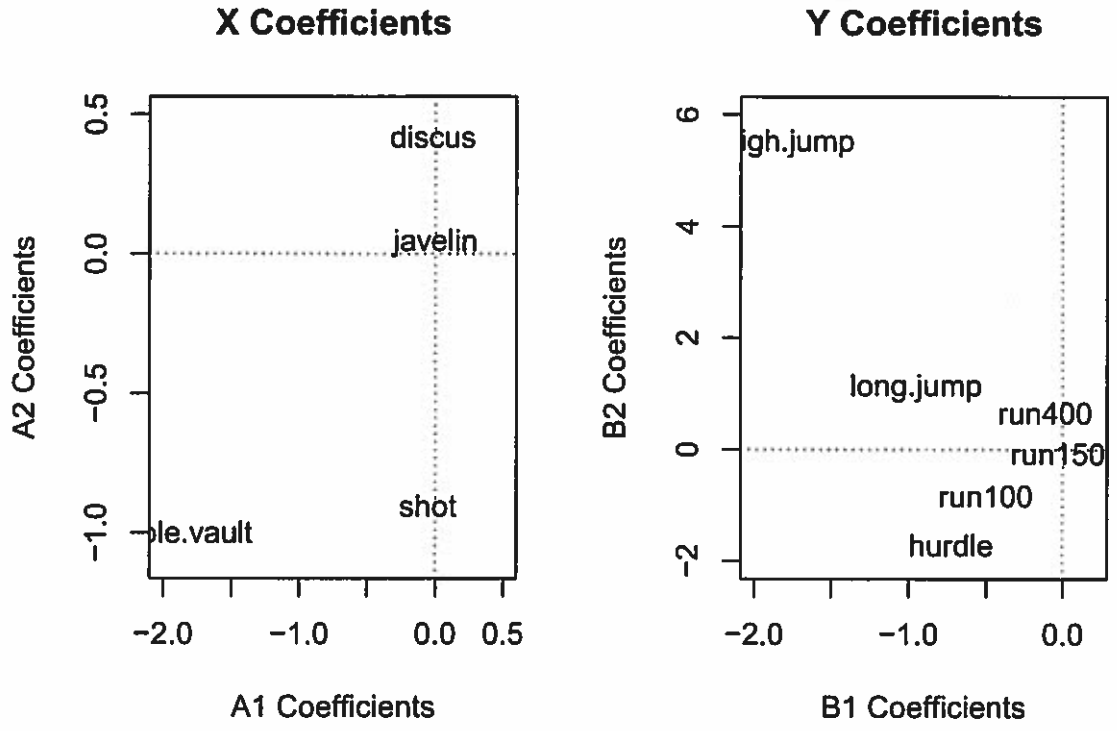
$\rho = \sqrt{\text{eigenvalue of } X} = \sqrt{\rho^2}$

```
## [1] 0.7702006 0.5033532 0.4184145 0.3052556
sqrt(Yeig$values[1:p])

## [1] 0.7702006 0.5033532 0.4184145 0.3052556
# compare linear combinations (different!)
Ahat <- Sxisqrt %*% Xeig$vectors
Bhat <- Syisqrt %*% Yeig$vectors
sum((cca$xcoef - Ahat)^2)

## [1] 6.710414
sum((cca$ycoef[,1:p] - Bhat[,1:p])^2)

## [1] 42.98483
```

NOTE*: we need to multiply R's xcoef and ycoef by $\sqrt{n-1}$ to obtain the results we are expecting.

```
# compare linear combinations (same!)
Ahat <- Sxisqrt %*% Xeig$vectors
Bhat <- Syisqrt %*% Yeig$vectors
sum((cca$xcoef * sqrt(n-1) - Ahat)^2)

## [1] 3.031301e-28
sum((cca$ycoef[,1:p] * sqrt(n-1) - Bhat[,1:p])^2)

## [1] 2.414499e-25
# plot coefficients
par(mfrow=c(1,2))
plot(Ahat[,1:2], xlab="A1 Coefficients", ylab="A2 Coefficients",
     type="n", main="X Coefficients", xlim=c(-2, 0.5), ylim=c(-1.1, 0.5))
text(Ahat[,1:2], labels=colnames(X))
abline(0,0,lty=3)
abline(v=0,lty=3)
plot(Bhat[,1:2], xlab="B1 Coefficients", ylab="B2 Coefficients",
     type="n", main="Y Coefficients", xlim=c(-2, 0.2), ylim=c(-2, 6))
text(Bhat[,1:2], labels=colnames(Y))
```

```
abline(0,0,lty=3)
abline(v=0,lty=3)
```



If $\mathbf{X} = \{x_{ij}\}_{n\times p}$ and $\mathbf{Y} = \{y_{ij}\}_{n\times q}$, then

- $\hat{\mathbf{U}} = \hat{\boldsymbol{A}}^T\mathbf{X} = \{\hat{u}_{ij}\}_{n\times p}$ where columns of $\hat{\mathbf{U}}$ contain the canonical variables for the $\mathbf{X}$ set.
- $\hat{\mathbf{V}} = \hat{\boldsymbol{B}}^T\mathbf{Y} = \{\hat{v}_{ij}\}_{n\times q}$ where columns of $\hat{\mathbf{V}}$ contain the canonical variables for the $\mathbf{Y}$ set.

```
# define canonical variates
U <- X %*% Ahat
V <- Y %*% Bhat

# canonical variable covariances
round(cov(U),4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
round(cov(V),4)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
## [4,]    0    0    0    1    0    0
## [5,]    0    0    0    0    1    0
## [6,]    0    0    0    0    0    1
```

```
round(cov(U,V),4)
```

```
##        [,1]   [,2]   [,3]   [,4] [,5] [,6]
## [1,] 0.7702 0.0000 0.0000 0.0000    0    0
## [2,] 0.0000 0.5034 0.0000 0.0000    0    0
## [3,] 0.0000 0.0000 0.4184 0.0000    0    0
## [4,] 0.0000 0.0000 0.0000 0.3053    0    0
```

```
# covariance of original and canonical variables (U and X)
Ainv <- solve(Ahat)
sum( ( cov(U, X) - crossprod(Ahat, Sx) )^2 )
```

```
## [1] 3.396329e-30
```

```
sum( ( Sx - crossprod(Ainv) )^2 )
```

```
## [1] 4.364327e-27
```

```
Sxhat <- matrix(0, p, p)
for(j in 1:p) Sxhat <- Sxhat + outer(Ainv[j,], Ainv[j,])
sum( (Sx - Sxhat)^2 )
```

```
## [1] 4.364327e-27
```

```
# covariance of original and canonical variables (V and Y)
Binv <- solve(Bhat)
sum( ( cov(V, Y) - crossprod(Bhat, Sy) )^2 )
```

```
## [1] 1.696269e-28
```

```
sum( ( Sy - crossprod(Binv) )^2 )
```

```
## [1] 3.027024e-26
```

```
Syhat <- matrix(0, q, q)
for(j in 1:q) Syhat <- Syhat + outer(Binv[j,], Binv[j,])
sum( (Sy - Syhat)^2 )
```

```
## [1] 3.027024e-26
```

```
# covariance of original and canonical variables (U and Y)
sum( (cov(U, Y) - crossprod(Ahat, Sxy))^2 )
```

```
## [1] 2.071712e-29
```

```
# covariance of original and canonical variables (V and X)
sum( (cov(V, X) - crossprod(Bhat, t(Sxy)))^2 )
```

```
## [1] 2.943246e-28
```

```
# covariance of canonical variables (U and V)
rhomat <- cbind(diag(rho), matrix(0, p, q-p))
sum( (cov(U, V) - rhomat)^2 )
```

```
## [1] 1.241068e-27
```

```
sum( (Sxy - crossprod(Ainv, rhomat) %*% Binv)^2 )
```

```
## [1] 1.355523e-25
```

```
Sxyhat <- matrix(0, p, q)
for(j in 1:p) Sxyhat <- Sxyhat + rho[j] * outer(Ainv[j,], Binv[j,])
sum( (Sxy - Sxyhat)^2 )
```

```
## [1] 1.37319e-25
```

Error of approximation matrices provide a descriptive measure of how well the first r pairs of canonical variables explain the covariation in the data. The error of approximation matrices are

$$\mathbf{S_X} - \sum_{j=1}^{r} [\mathbf{a}^{(j)}][\mathbf{a}^{(j)}]^T$$

$$\mathbf{S_Y} - \sum_{j=1}^{r} [\mathbf{b}^{(j)}][\mathbf{b}^{(j)}]^T$$

$$\mathbf{S_{XY}} - \sum_{j=1}^{r} \rho_j [\mathbf{a}^{(j)}][\mathbf{b}^{(j)}]^T$$

```
# error of approximation matrices (with r=2)
Ainv <- solve(Ahat)
Binv <- solve(Bhat)
r <- 2
Ex <- Sx - crossprod(Ainv[1:r,])
Ey <- Sy - crossprod(Binv[1:r,])
Exy <- Sxy - crossprod(diag(rho[1:r]) %*% Ainv[1:r,], Binv[1:r,])

# get norms of error matrices
sqrt(mean(Ex^2))
```

```
## [1] 6.561393
```

```
sqrt(mean(Ey^2))
```

```
## [1] 18.37339
```

```
sqrt(mean(Exy^2))
```

```
## [1] 1.725392
```

CCA via Correlation (standardized CCA)

```
# standardize data
Xs <- scale(X)
Ys <- scale(Y)

# canonical correlations of correlations (standardized data)
ccas <- cancor(Xs, Ys)

# cca (the normal way)
Sx <- cov(Xs)
Sy <- cov(Ys)
Sxy <- cov(Xs,Ys)
Sxeig <- eigen(Sx, symmetric=TRUE)
Sxisqrt <- Sxeig$vectors %*% diag(1/sqrt(Sxeig$values)) %*% t(Sxeig$vectors)
```

```
Syeig <- eigen(Sy, symmetric=TRUE)
Syisqrt <- Syeig$vectors %*% diag(1/sqrt(Syeig$values)) %*% t(Syeig$vectors)
Xmat <- Sxisqrt %*% Sxy %*% solve(Sy) %*% t(Sxy) %*% Sxisqrt
Ymat <- Syisqrt %*% t(Sxy) %*% solve(Sx) %*% Sxy %*% Syisqrt
Xeig <- eigen(Xmat, symmetric=TRUE)
Yeig <- eigen(Ymat, symmetric=TRUE)

# compare correlations (same)
cca$cor
```

```
## [1] 0.7702006 0.5033532 0.4184145 0.3052556
```

```
sqrt(Xeig$values)
```

```
## [1] 0.7702006 0.5033532 0.4184145 0.3052556
```

```
sqrt(Yeig$values[1:p])
```

```
## [1] 0.7702006 0.5033532 0.4184145 0.3052556
```

```
# compare linear combinations (different?)
Ahat <- Sxisqrt %*% Xeig$vectors
Bhat <- Syisqrt %*% Yeig$vectors
sum((ccas$xcoef * sqrt(n-1) - Ahat)^2)
```

```
## [1] 3.332536e-29
```

```
sum((ccas$ycoef[,1:p] * sqrt(n-1) - Bhat[,1:p])^2)
```

```
## [1] 11.59453
```

```
# note that the signing is arbitary!!
ccas$ycoef[,1:p] * sqrt(n-1)
```

```
##                [,1]       [,2]       [,3]        [,4]
## run100    -0.1439138 -0.2404940  0.5274876 -0.13754449
## run400    -0.1373435  0.7655659 -1.2826821  0.96359176
## run1500    0.3023537 -1.0519285 -0.1514027 -0.52923644
## hurdle    -0.4396044 -1.0374417  0.6303782  0.49905604
## long.jump -0.3564702  0.4110878 -0.0253127 -1.09325282
## high.jump -0.1855627  0.5731149 -0.2615838 -0.09007821
```

```
Bhat[,1:p]
```

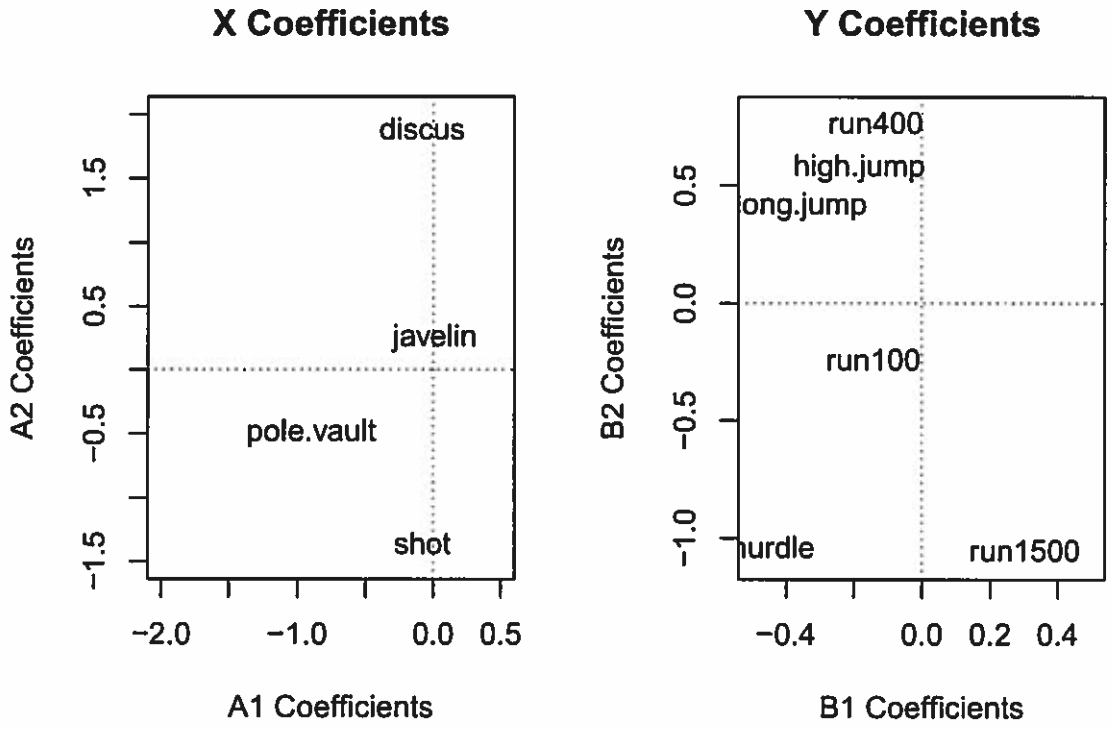
```
##           [,1]       [,2]       [,3]        [,4]
## [1,]  0.1439138 -0.2404940 -0.5274876 -0.13754449
## [2,]  0.1373435  0.7655659  1.2826821  0.96359176
## [3,] -0.3023537 -1.0519285  0.1514027 -0.52923644
## [4,]  0.4396044 -1.0374417 -0.6303782  0.49905604
## [5,]  0.3564702  0.4110878  0.0253127 -1.09325282
## [6,]  0.1855627  0.5731149  0.2615838 -0.09007821
```

```
Bhat[,1:p] <- Bhat[,1:p] %*% diag(c(-1,1,-1,1))
sum((ccas$ycoef[,1:p] * sqrt(n-1) - Bhat[,1:p])^2)
```

```
## [1] 1.132493e-28
```

```
# plot coefficients
par(mfrow=c(1,2))
```

```
plot(Ahat[,1:2], xlab="A1 Coefficients", ylab="A2 Coefficients",
     type="n", main="X Coefficients", xlim=c(-2, 0.5), ylim=c(-1.5, 2))
text(Ahat[,1:2], labels=colnames(X))
abline(0,0,lty=3)
abline(v=0,lty=3)
plot(Bhat[,1:2], xlab="B1 Coefficients", ylab="B2 Coefficients",
     type="n", main="Y Coefficients", xlim=c(-0.5, 0.5), ylim=c(-1.1, 0.8))
text(Bhat[,1:2], labels=colnames(Y))
abline(0,0,lty=3)
abline(v=0,lty=3)
```




```
# define canonical variates
U <- Xs %*% Ahat
V <- Ys %*% Bhat

# canonical variable covariances
round(cov(U),4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
round(cov(V),4)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
```

```
## [4,]    0    0    0    1    0    0
## [5,]    0    0    0    0    1    0
## [6,]    0    0    0    0    0    1
```

```
round(cov(U,V),4)
```

```
##        [,1]   [,2]   [,3]   [,4] [,5] [,6]
## [1,] 0.7702 0.0000 0.0000 0.0000    0    0
## [2,] 0.0000 0.5034 0.0000 0.0000    0    0
## [3,] 0.0000 0.0000 0.4184 0.0000    0    0
## [4,] 0.0000 0.0000 0.0000 0.3053    0    0
```

```
# covariance of original and canonical variables (U and Xs)
Ainv <- solve(Ahat)
sum( ( cov(U, Xs) - crossprod(Ahat, Sx) )^2 )
```

```
## [1] 2.759323e-31
```

```
sum( ( Sx - crossprod(Ainv) )^2 )
```

```
## [1] 6.569732e-30
```

```
Sxhat <- matrix(0, p, p)
for(j in 1:p) Sxhat <- Sxhat + outer(Ainv[j,], Ainv[j,])
sum( (Sx - Sxhat)^2 )
```

```
## [1] 6.569732e-30
```

```
# covariance of original and canonical variables (V and Ys)
Binv <- solve(Bhat)
sum( ( cov(V, Ys) - crossprod(Bhat, Sy) )^2 )
```

```
## [1] 2.406961e-31
```

```
sum( ( Sy - crossprod(Binv) )^2 )
```

```
## [1] 3.136492e-29
```

```
Syhat <- matrix(0, q, q)
for(j in 1:q) Syhat <- Syhat + outer(Binv[j,], Binv[j,])
sum( (Sy - Syhat)^2 )
```

```
## [1] 3.136492e-29
```

```
# covariance of original and canonical variables (U and Ys)
sum( (cov(U, Ys) - crossprod(Ahat, Sxy))^2 )
```

```
## [1] 5.477785e-32
```

```
# covariance of original and canonical variables (V and Xs)
sum( (cov(V, Xs) - crossprod(Bhat, t(Sxy)))^2 )
```

```
## [1] 1.336149e-31
```

```
# covariance of canonical variables (U and V)
rhomat <- cbind(diag(rho), matrix(0, p, q-p))
sum( (cov(U, V) - rhomat)^2 )
```

```
## [1] 1.272906e-29
```

```
sum( (Sxy - crossprod(Ainv, rhomat) %*% Binv)^2 )
```

```
## [1] 7.505349e-30
Sxyhat <- matrix(0, p, q)
for(j in 1:p) Sxyhat <- Sxyhat + rho[j] * outer(Ainv[j,], Binv[j,])
sum( (Sxy - Sxyhat)^2 )

## [1] 7.283289e-30
# error of approximation matrices (with r=2)
Ainv <- solve(Ahat)
Binv <- solve(Bhat)
r <- 2
Ex <- Sx - crossprod(Ainv[1:r,])
Ey <- Sy - crossprod(Binv[1:r,])
Exy <- Sxy - crossprod(diag(rho[1:r]) %*% Ainv[1:r,], Binv[1:r,])

# get norms of error matrices
sqrt(mean(Ex^2))

## [1] 0.2432351
sqrt(mean(Ey^2))

## [1] 0.2296716
sqrt(mean(Exy^2))

## [1] 0.07458264
```

Canonical correlation analysis.

- explore the relationship between two multivariate sets of variables (vector)
  - exercise : climbing rate, fast can run, amount weight lifted,
  - health : blood pressure, cholesterol measure, glucose level, body max ..
  - sale performance
  - aptitude variable

Chapter 7. Discrimination and Classification

Jianxuan Liu

Fall 2018

Main objectives of discrimination and classification are:

1. Separate distinct sets of objects: discrimination.
2. Classify new objects into well defined populations: classification.

Examples of discrimination:

- A tumor is benign or malignant, and the correct diagnosis needs to be obtained.
- In the finance and credit risk area, one wants to assess whether a company is likely to go bankrupt in the next few years or whether a client will default on mortgage repayments.

Example of classification:

- Using information on prisoners eligible for parole (good behavior, history of drug use, job skills, etc) can we successfully allocate a prisoners elegible for parole into two groups: those who will commit another crime or those who will not commit another crime?

* Supervise Learning
Discriminant classification
* Unsupervise clustering

Discriminant analysis and classification are termed "supervised learning" while clustering is called "unsupervised learning" in the machine learning literature. Discriminant analysis and classification are slightly different actions, but they are used interchangeably.

In practice, the two objectives often overlap: A function of the p variables that serves as a discriminant is also used for classifying a new object into one of the populations. An allocation or classification rule can often serve as a discriminant.

The setup is the usual one: p variables are measured on n sample units or subjects. We wish to find a function of the variables that will optimize the discrimination between units belonging to different populations (minimize classification errors).

1. Classifying Two Populations

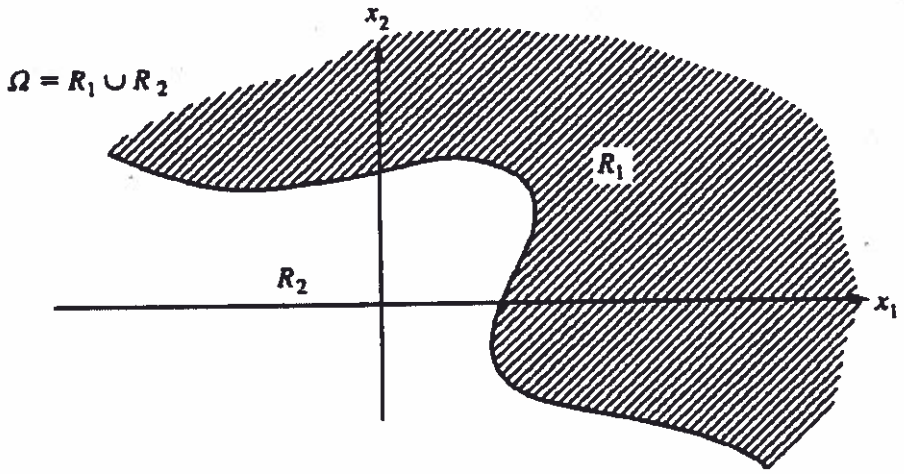
We wish to separate two populations and also allocate new units to one of the two populations. Let $\mathbf{X} = (X_1, X_2, \cdots, X_p)^T$ denote a p-dimensional random vector. Let $f_1(\mathbf{x})$ denote the probability density function (pdf) for population π_1 and $f_2(\mathbf{x})$ denote the probability density function (pdf) for population π_2. Given a realization $\mathbf{X} = \mathbf{x}$, we want to assign $\mathbf{x}$ to π_1 or π_2. We want to find some classification rule to determine whether a realization $\mathbf{X} = \mathbf{x}$ should be assigned to population π_1 or π_2.

Let Ω denote the sample space (collection of all possible values for $\mathbf{x}$).

- R_1 is the set of values of $\mathbf{x}$ for which we classify objects into π_1 and
- $R_2 = \Omega - R_1$ is the set of values of $\mathbf{x}$ for which we classify objects into π_2.

Since every object belongs into one of the two populations, we have that

$$\Omega = R_1 \cup R_2 \qquad R_1 \cap R_2 = \phi$$

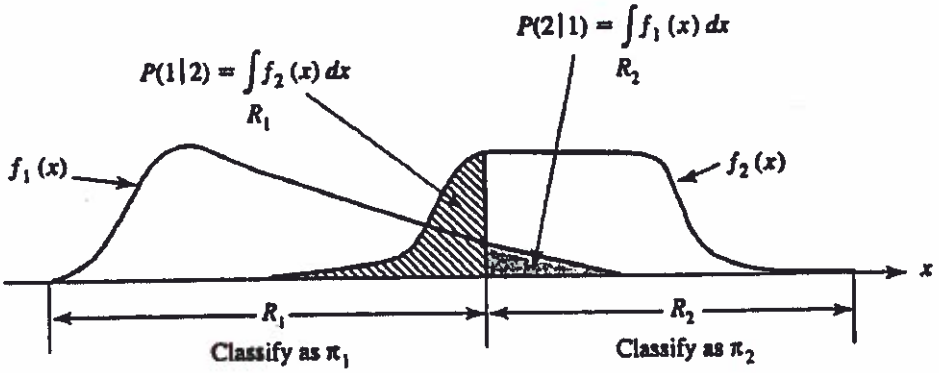


Ignore for now the prior probabilities of each population and the potentially different misclassification costs. The probability of misclassifying an object into π_2 when it belongs in π_1 is

$$P(2 \mid 1) = P(\mathbf{X} \in R_2 \mid \pi_1) = \int_{R_2} f_1(\mathbf{x})d\mathbf{x}$$

and probability of misclassifying an object into π_1 when it belongs in π_2 is

$$P(1 \mid 2) = P(\mathbf{X} \in R_1 \mid \pi_2) = \int_{R_1} f_2(\mathbf{x})d\mathbf{x}$$



In the above graph ($p = 1$ variable), the solid shaded region is $P(2 \mid 1)$ and the striped region is $P(1 \mid 2)$.

Probability of Misclassification

Let p_1, p_2 denote the **prior probabilities** of π_1, π_2, respectively, with the constraint that $p_1 + p_2 = 1$.

The **overall probabilities** of the four outcomes have the form

$P(\pi_1 \text{ and say } R_1)$ = $P(\text{correctly classify as } \pi_1) = P(\mathbf{X} \in R_1 \mid \pi_1)P(\pi_1) = P(1 \mid 1)p_1$

$P(\pi_2 \text{ and } R_2)$ = $P(\text{correctly classify as } \pi_2) = P(\mathbf{X} \in R_2 \mid \pi_2)P(\pi_2) = P(2 \mid 2)p_2$

$P(\pi_1 \text{ and } R_2) = P(R_2|\pi_1)\,P(\pi_1)$

$= P(\text{misclassify } \pi_1 \text{ as } \pi_2) = P(\mathbf{X} \in R_2 \mid \pi_1)P(\pi_1) = P(2 \mid 1)p_1$

$P(\text{misclassify } \pi_2 \text{ as } \pi_1) = P(\mathbf{X} \in R_1 \mid \pi_2)P(\pi_2) = P(1 \mid 2)p_2$

Further, let $c(1 \mid 2)$ and $c(2 \mid 1)$ be the costs of misclassifying an object into π_2 and π_1, respectively.

In many real world cases, **costs of misclassification** are not equal, for example:

(a). π_1 and π_2 are diseased and healthy

(b). π_1 and π_2 are guilty and not guilty

(c). π_1 and π_2 are buy and not buy stock

We can make a cost matrix to tabulate our misclassification costs:

		Classify as:	
		π_1	π_2
Truth	π_1	0	$c(2 \mid 1)$
	π_2	$c(1 \mid 2)$	0

(ECM)

Classification rules are often evaluated in terms of the **expected cost of misclassification** or ECM:

$$ECM = c(2 \mid 1)P(2 \mid 1)p_1 + c(1 \mid 2)P(1 \mid 2)p_2$$

There is no cost when units are correctly classified. We seek rules that minimize the ECM. This leads to an **optimal classification rule**: classify an object into π_1 if

$$\frac{f_1(\mathbf{x})c(2 \mid 1)p_1}{f_2(\mathbf{x})c(1 \mid 2)p_2} > 1.$$

when the "cost" that we assign R_2 while π_1 is bigger than another one.

Equivalently, the regions R_1, R_2 that minimize the ECM are defined by the values of $\mathbf{x}$ for which

$$R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \left(\frac{c(1 \mid 2)}{c(2 \mid 1)}\right)\left(\frac{p_2}{p_1}\right)$$

ratio of misclassification cost; ratio of prior prob

$$R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \left(\frac{c(1 \mid 2)}{c(2 \mid 1)}\right)\left(\frac{p_2}{p_1}\right)$$

just need to know the ratio of the cost

$$ECM = c(2|1)\int_{R_2} f_1(x)\,dx\; p_1 + c(1|2)\int_{R_1} f_2(x)\,dx\; p_2$$

We have $\Omega = R_1 + R_2 \Rightarrow 1 = \int_{R_1} f_1(x) + \int_{R_2} f_1(x)\,dx$

Then

$$ECM = c(2|1)p_1\left(1 - \int_{R_1} f_1(x)\,dx\right) + c(1|2)p_2\left(\int_{R_1} f_2(x)\,dx\right)$$

$$= \int_{R_1}\underbrace{\left[c(1|2)p_2 f_2(x) - c(2|1)p_1 f_1(x)\right]}_{(*)}dx + c(2|1)p_1$$

ECM is minimized when R_1 is chosen to be the region where $(*) \leq 0$

$$\Rightarrow c(1|2)p_2 f_2(x) \leq c(2|1)p_1 f_1(x)$$

$$\Rightarrow R_1 : \frac{f_1}{f_2} > \frac{c(1|2)}{c(2|1)}\,\frac{p_2}{p_1}$$

Remark: Implementation of the minimum ECM rule require for a new unit requires evaluation of f_1 and f_2 at the new vector of observations $\mathbf{x}_0$, but it does not require knowing the two costs or the two prior probabilities, just their ratio.

Special Cases of Minimum ECM

When the prior probabilities or the misclassification costs are equal, the classification rule above simplifies correspondingly.

- If $p_1 = p_2$, then $R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \frac{c(1 \mid 2)}{c(2 \mid 1)}$, and $R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{c(1 \mid 2)}{c(2 \mid 1)}$
- If $c(1 \mid 2) = c(2 \mid 1)$, then $R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \frac{p_2}{p_1}$, and $R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2}{p_1}$
- If $p_1 = p_2$ and $c(1 \mid 2) = c(2 \mid 1)$, then $R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > 1$, and $R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < 1$

If $\mathbf{x}$ is on the boundary between R_1 and R_2 then toss a coin, or randomly classify in some way.

Other Criteria for Choosing a Classification Rule

1. Highest posterior probability criteria

We might consider classifying a new unit with observation $\mathbf{x}_0$ into the population with the highest posterior probability $P(\pi_i \mid \mathbf{x}_0)$. By Bayes rule

$$P(\pi_1 \mid \mathbf{x}_0) = \frac{P(\mathbf{x}_0 \mid \pi_1)p_1}{P(\mathbf{x}_0 \mid \pi_1)p_1 + P(\mathbf{x}_0 \mid \pi_2)p_2} = \frac{p_1 f_1(\mathbf{x}_0)}{p_1 f_1(\mathbf{x}_0) + p_2 f_2(\mathbf{x}_0)}$$

$P(\pi_1|\mathbf{x}_0) > 0.5$ ⇒ assign to π_1
$P(\pi_1|\mathbf{x}_0) < 0.5$ ⇒ assign to π_2

Clearly, $P(\pi_2 \mid \mathbf{x}_0) = 1 - P(\pi_1 \mid \mathbf{x}_0)$. Using the posterior probability criterion, we classify a unit with measurements $\mathbf{x}_0$ into π_1 when $P(\pi_1 \mid \mathbf{x}_0) > P(\pi_2 \mid \mathbf{x}_0)$

2. Minimum total probability of misclassification (TPM) criteria The **total probability of misclassification (TPM)** ignores the cost of misclassification and is defined as the probability of either misclassifying a π_1 observation or misclassifying a π_2 observation:

$$TPM = p_1 \int_{R_2} f_1(\mathbf{x})d\mathbf{x} + p_2 \int_{R_1} f_2(\mathbf{x})d\mathbf{x}.$$

TPM is equivalent to ECM when costs are equal. An optimal rule in this sense would minimize TPM. The optimal TPM regions are the same as the special case of $c(1 \mid 2) = c(2 \mid 1)$. That is

$$R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \frac{p_2}{p_1} \quad \text{and} \quad R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2}{p_1}$$

2. Two Multivariate Normal Populations with $\Sigma_1 = \Sigma_2 = \Sigma$

$(n \times p)$ matrix n observations p variables

Let $\mathbf{X} = (X_1, \cdots, X_p)^T$ denote a random vector. We now assume that $f_1(\mathbf{x}) \sim N(\mu_1, \Sigma)$ and $f_2(\mathbf{x}) \sim N(\mu_2, \Sigma)$ denote the pdf for population π_1 and π_2, respectively. Given a realization $\mathbf{X} = \mathbf{x}$, we want to find some classification rule to determine whether a realization $\mathbf{X} = \mathbf{x}$ should be assigned to population π_1 or π_2. The multivariate normal densities have the form

$$f_k(\mathbf{x}) = (2\pi)^{-p/2} \mid \Sigma \mid^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k)\}, \quad k = 1, 2$$

which implies that

$$
\begin{aligned}
f^* &= \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \\
&= \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^T\Sigma^{-1}(\mathbf{x}-\mu_1) + \frac{1}{2}(\mathbf{x}-\mu_2)^T\Sigma^{-1}(\mathbf{x}-\mu_2)\right\} \\
&= \exp\left\{(\mu_1-\mu_2)^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1+\mu_2)\right\}.
\end{aligned}
$$

Then, R_1 is given by the set of $\mathbf{x}$ values for which:

$$
\begin{cases}
R_1 : \dfrac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \left(\dfrac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\dfrac{p_2}{p_1}\right) \\
\text{and} \\
R_2 : \dfrac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \left(\dfrac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\dfrac{p_2}{p_1}\right).
\end{cases}
$$

Given the definitions of R_1, R_2, an allocation that minimizes the ECM is the following: allocate $\mathbf{x}_0$ to π_1 if

$$
\ln\left(\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})}\right) = (\mu_1-\mu_2)^T\Sigma^{-1}\mathbf{x}_0 - \frac{1}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1+\mu_2) > \ln\left[\left(\frac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\frac{p_2}{p_1}\right)\right]
$$

allocate $\mathbf{x}_0$ to π_2 otherwise.

Since μ_1, μ_2, Σ are typically unknown, in practice we use $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2$ as estimators of the population means, and $\mathbf{S}_{pooled}$ as estimator of the common covariance matrix Σ. Given n_1 independent observations from π_1 and n_2 independent observations from π_2, we can estimate the needed parameters $\hat{\mu}_1 = \bar{\mathbf{x}}_1 = \frac{1}{n_1}\sum_{j=1}^{n_1}\mathbf{x}_{1j}$, $\hat{\mu}_2 = \bar{\mathbf{x}}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2}\mathbf{x}_{2j}$ and

$$
\hat{\Sigma} = \mathbf{S}_{pooled} = \frac{1}{n_1+n_2-2}\left[\sum_{j=1}^{n_1}(\mathbf{x}_{1j}-\bar{\mathbf{x}}_1)(\mathbf{x}_{1j}-\bar{\mathbf{x}}_1)^T + \sum_{j=1}^{n_2}(\mathbf{x}_{2j}-\bar{\mathbf{x}}_2)(\mathbf{x}_{2j}-\bar{\mathbf{x}}_2)^T\right].
$$

The estimated classification rule replaces f^* with its sample estimate:

$$
\hat{f}^* = \exp\left\{(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}\mathbf{x}_0 - \frac{1}{2}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}(\bar{\mathbf{x}}_1+\bar{\mathbf{x}}_2)\right\}.
$$

- If $\left(\dfrac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\dfrac{p_2}{p_1}\right) = 1$, then $\ln(1) = 0$ and the rule becomes

$$
R_1 : \hat{y} > \hat{m} \text{ and } R_2 : \hat{y} < \hat{m},
$$

where the scalar variables

$$
\hat{y} = (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}\mathbf{x}_0 = \mathbf{a}^T\mathbf{x}
$$

$$
\hat{m} = \frac{1}{2}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}(\bar{\mathbf{x}}_1+\bar{\mathbf{x}}_2) = \frac{1}{2}(\bar{y}_1+\bar{y}_2)
$$

with

- $\bar{y}_1 = (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}\mathbf{x}_1 = \hat{\mathbf{a}}^T\mathbf{x}_1$ and
- $\bar{y}_2 = (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}\mathbf{x}_2 = \hat{\mathbf{a}}^T\mathbf{x}_2$.

The coefficient vector $\hat{\mathbf{a}} = \mathbf{S}_{pooled}^{-1}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)$ is unique only up to a multiplicative constant. Therefore, $\hat{\mathbf{a}}$ is frequently "scaled" or "normalized" to ease the interpretation of its elements. Two most commonly methods for normalization are

1. $\hat{\mathbf{a}}^* = \frac{\hat{\mathbf{a}}}{\sqrt{\hat{\mathbf{a}}^T\hat{\mathbf{a}}}}$, $\hat{\mathbf{a}}^*$ has unit length

2. $\hat{\mathbf{a}}^* = \frac{\hat{\mathbf{a}}}{\hat{a}_1}$, the first element of the new coefficient vector $\hat{\mathbf{a}}^*$ is 1.

Fisher's Linear Discriminant Function

R. A. Fisher considered finding the linear combination $Y = \mathbf{a}^T\mathbf{x}$ that best separates the groups:

$$\text{separation} = \frac{|\bar{y}_1 - \bar{y}_2|}{s_y}$$

where $s_y^2 = \frac{\sum_{j=1}^{n_1}(\bar{y}_{1j} - \bar{y}_1)^2 + \sum_{j=1}^{n_2}(\bar{y}_{2j} - \bar{y}_2)^2}{n_1 + n_2 - 2}$ is the pooled estimate of the variance. $\bar{y}_1$ is the mean of the Y scores for the observations from π_1. $\bar{y}_2$ is the mean of the Y scores for the observations from π_2. The objective is to select the linear combination of the $\mathbf{x}$ to achieve maximum separation of the sample means $\bar{y}_1$ and $\bar{y}_2$.

Result: The linaer combination $\hat{y} = \hat{\mathbf{a}}^T\mathbf{x} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}\mathbf{x}$ maximize the ratio

$$\text{separation}^2 = \frac{(\bar{y}_1 - \bar{y}_2)^2}{s_y^2} = \frac{(\hat{\mathbf{a}}^T\bar{\mathbf{x}}_1 - \hat{\mathbf{a}}^T\bar{\mathbf{x}}_2)^2}{\hat{\mathbf{a}}^T\mathbf{S}_{pooled}\hat{\mathbf{a}}} = \frac{(\hat{\mathbf{a}}^T\mathbf{d})^2}{\hat{\mathbf{a}}^T\mathbf{S}_{pooled}\hat{\mathbf{a}}}$$

over all possible coefficient vectors $\hat{\mathbf{a}}$ where $\mathbf{d} = \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2$. The maximum of the ratio is

$$D^2 = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pooled}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2).$$

which is equal to the squared distance between the multivariate means.

$\max \frac{(X^Tc)^2}{X^TDX} = c^TD^{-1}c$ when $X = CD^{-1}c$, $C \neq 0$

A test of $H_0 : \mu_1 = \mu_2$ is also a test for the hypothesis of no separation. If Hotelling's T^2 fails to reject H_0, then the data will not provide a useful classification rule.

Example: Classifying gender using dental data

Time series

depends on time.

$\begin{bmatrix} = \\ = \end{bmatrix}$ } 1st subject.

```
library(reshape) # to create data frame from dental data
```

```
## Warning: package 'reshape' was built under R version 3.4.4
```

```
library(MASS) # has lda and qda functions
library(heavy) # has dental data
```

```
## Warning: package 'heavy' was built under R version 3.4.3
```

```
library(klaR) # visualize the LDA paritions
```

```
## Warning: package 'klaR' was built under R version 3.4.4
```

```
data(dental)
d2=cast(melt(dental,id=c("Subject","age","Sex")),Subject+Sex~age)
names(d2)[3:6]=c("d8","d10","d12","d14")
d2=d2[,-1]
g=2
p=ncol(d2) - 1
```

reshape2

Head d1 Sex d8 d10 d12 d14

use their measure to detect their gender

↳ Up 4 times.

```
# pooled covariances matrix
Sp=matrix(0, p, p)
nx=rep(0, g)
lev=levels(d2$Sex)
for(k in 1:g){
  x=d2[d2$Sex==lev[k],1:p+1]
  nx[k]=nrow(x)
  Sp=Sp + cov(x) * (nx[k] - 1)
}
Sp=Sp / (sum(nx) - g)
round(Sp,3)
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] 5.415 2.717 3.910 2.710
## [2,] 2.717 4.185 2.927 3.317
## [3,] 3.910 2.927 6.456 4.131
## [4,] 2.710 3.317 4.131 4.986
```

```
# fit lda model
# assume equal prior
ldamod=lda(d2$Sex ~ ., data=d2, prior=rep(1/2, 2))
names(ldamod)
```

fit gender by teeth measure

we have to know prior prob

prior probability

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```
# check the LDA coefficients/scalings
# a_1, ...  a_p are given as "Coefficients of linear discriminants".
ldamod$scaling
```

$Y = \vec{a}X$

X measure.

```
##             LD1
## d8  -0.05161940
## d10  0.22969447
## d12  0.05015623
## d14 -0.59205628
```

```
crossprod(ldamod$scaling, Sp) %*% ldamod$scaling
```

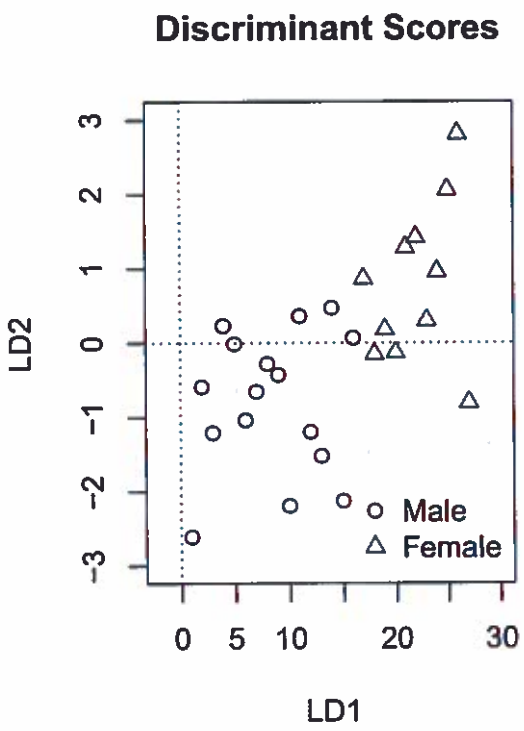
```
##     LD1
## LD1   1
```

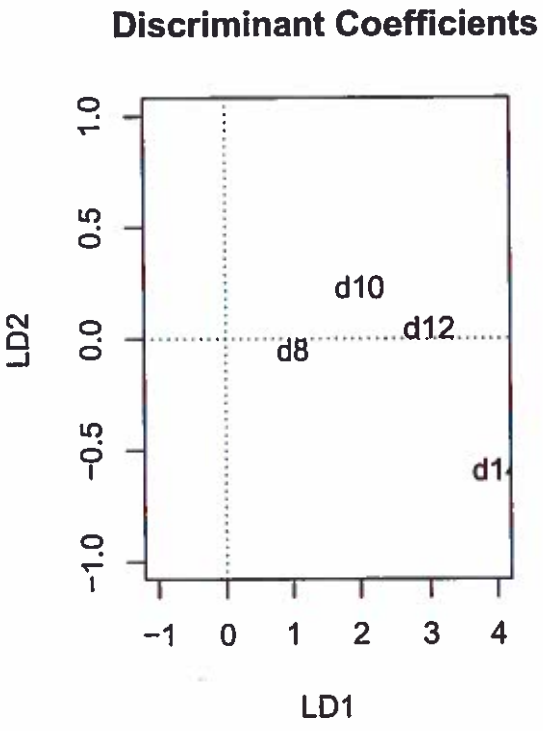
```
# create the (centered) discriminant scores
mu.k=ldamod$means
mu=colMeans(mu.k)
dscores=scale(d2[,-1], center=mu, scale=F) %*% ldamod$scaling
sum((dscores - predict(ldamod)$x)^2)
```

```
## [1] 0
```

```
# plot the scores and coefficients
spid=as.integer(d2$Sex)
par(mfrow=c(1,2))
plot(dscores, xlab="LD1", ylab="LD2", pch=spid, col=spid,
     main="Discriminant Scores", xlim=c(-2, 30), ylim=c(-3, 3))
abline(h=0, lty=3)
abline(v=0, lty=3)
legend("bottomright",lev,pch=1:3,col=1:3,bty="n")
```

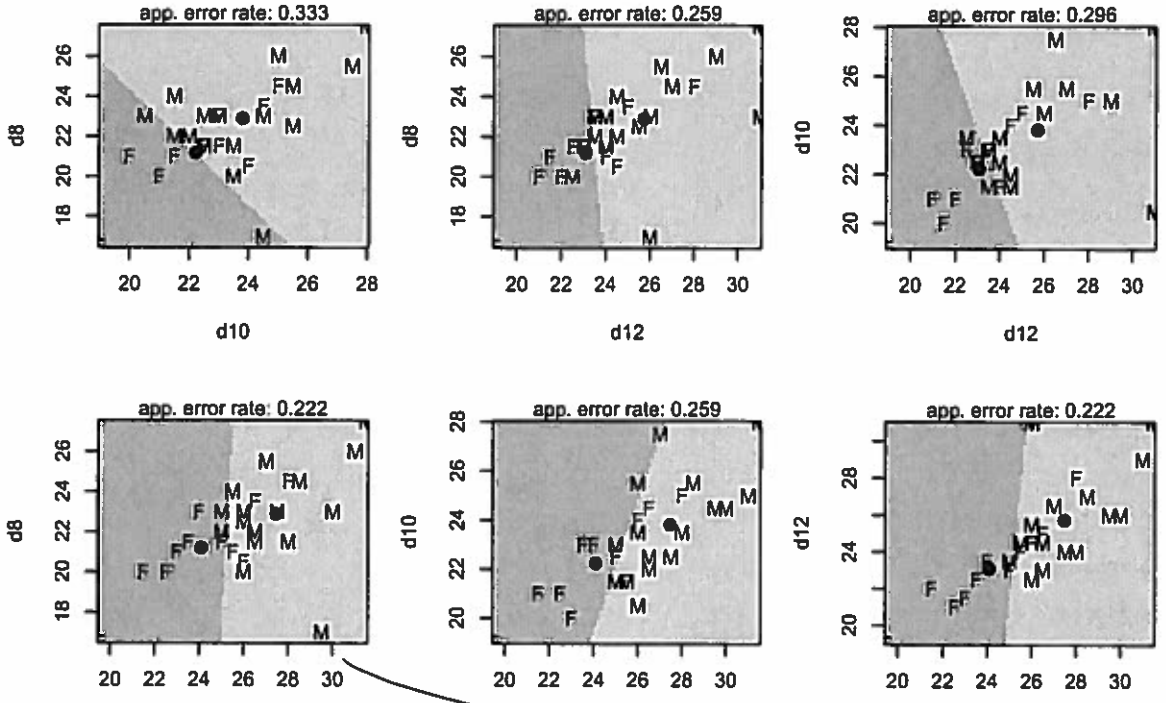
```
plot(ldamod$scaling, xlab="LD1", ylab="LD2", type="n",
     main="Discriminant Coefficients", xlim=c(-1, 4), ylim=c(-1, 1))
text(ldamod$scaling, labels=rownames(ldamod$scaling))
abline(h=0, lty=3)
abline(v=0, lty=3)
```






```
# visualize the LDA partitions
partimat(Sex ~ ., data=d2, method="lda")
```

Partition Plot d(8, 10, 12, 14)



First, have to check the assumption of $\Sigma_1 \neq \Sigma_2$ or $\Sigma_1 = \Sigma_2$ depends on misclassification then use only d8 & d14.

3. Two Multivariate Normal Populations with $\Sigma_1 \neq \Sigma_2$

When $\Sigma_1 \neq \Sigma_2$, we can no longer use a simple linear classification rule. Under normality, terms involving $|\Sigma_i|^{1/2}$ do not cancel and the exponential terms do not combine easily. Using the definition of R_1 given earlier and expressing the likelihood ratio in the log scale, we now have:

$R_1: \ln(y^*) > \ln\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right)$ $R_1 : (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})\mathbf{x} - k - \frac{1}{2}\mathbf{x}^T(\Sigma_1^{-1} - \Sigma_2^{-1})\mathbf{x} \geq \ln\left[\left(\frac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\frac{p_2}{p_1}\right)\right]$

$$R_2 : (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})\mathbf{x} - k - \frac{1}{2}\mathbf{x}^T(\Sigma_1^{-1} - \Sigma_2^{-1})\mathbf{x} < \ln\left[\left(\frac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\frac{p_2}{p_1}\right)\right]$$

where $k = \frac{1}{2}\ln\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \frac{1}{2}(\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2)$.

The classification regions are now quadratic functions of $\mathbf{x}$. The classification rule for a new observation $\mathbf{x}_0$ is now the following: allocate $\mathbf{x}_0$ to π_1 if

$$R_1 : (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})\mathbf{x}_0 - k - \frac{1}{2}\mathbf{x}_0^T(\Sigma_1^{-1} - \Sigma_2^{-1})\mathbf{x}_0 \geq \ln\left[\left(\frac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\frac{p_2}{p_1}\right)\right] \quad \checkmark$$

and allocate $\mathbf{x}_0$ to π_2 otherwise. In practice, we estimate the classification rule by substituting the unobservable population parameters $(\mu_1, \mu_2, \Sigma_1, \Sigma_2)$ by sample estimates $(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \mathbf{S}_1, \mathbf{S}_2)$.

Quadratic Classification Rule

Allocate $\mathbf{x}_0$ to π_1 if

$$(\bar{\mathbf{x}}_1^T\mathbf{S}_1^{-1} - \bar{\mathbf{x}}_2^T\mathbf{S}_2^{-1})\mathbf{x}_0 - k - \frac{1}{2}\mathbf{x}_0^T(\mathbf{S}^{-1} - \mathbf{S}_2^{-1})\mathbf{x}_0 \geq \ln\left[\left(\frac{c(1\mid 2)}{c(2\mid 1)}\right)\left(\frac{p_2}{p_1}\right)\right]$$

where $k = \frac{1}{2}\ln\left(\frac{|\mathbf{S}_1|}{|\mathbf{S}_2|}\right) + \frac{1}{2}(\bar{\mathbf{x}}_1^T\mathbf{S}_1^{-1}\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^T\mathbf{S}_2^{-1}\bar{\mathbf{x}}_2)$, and allocate $\mathbf{x}_0$ to π_2 otherwise.

Revisit the wine data.

Check equal variance assumption

```
library(biotools)
```

```
## Loading required package: rpanel
## Loading required package: tcltk
## Warning: running command ''/usr/bin/otool' -L '/Library/Frameworks/
## R.framework/Resources/library/tcltk/libs//tcltk.so'' had status 1
## Package `rpanel', version 1.1-3: type help(rpanel) for summary information
## Loading required package: tkrplot
## Warning: package 'tkrplot' was built under R version 3.4.4
## Loading required package: lattice
## Loading required package: SpatialEpi
## Warning: package 'SpatialEpi' was built under R version 3.4.4
## Loading required package: sp
## ---
## biotools version 3.1
##
```

```
boxM(d2[,-1],d2[,1])
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  d2[, -1]
## Chi-Sq (approx.) = 17.335, df = 10, p-value = 0.06727
```

At $\alpha = 10\%$, we reject ~~$H_0 : \Sigma_1 = \Sigma_2$~~. Now we fit a qda model

```
# fit qda model
qdamod=qda(Sex ~ ., data=d2, prior=rep(1/2, 2))
names(qdamod)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "ldet"    "lev"     "N"
##  [8] "call"    "terms"   "xlevels"
```

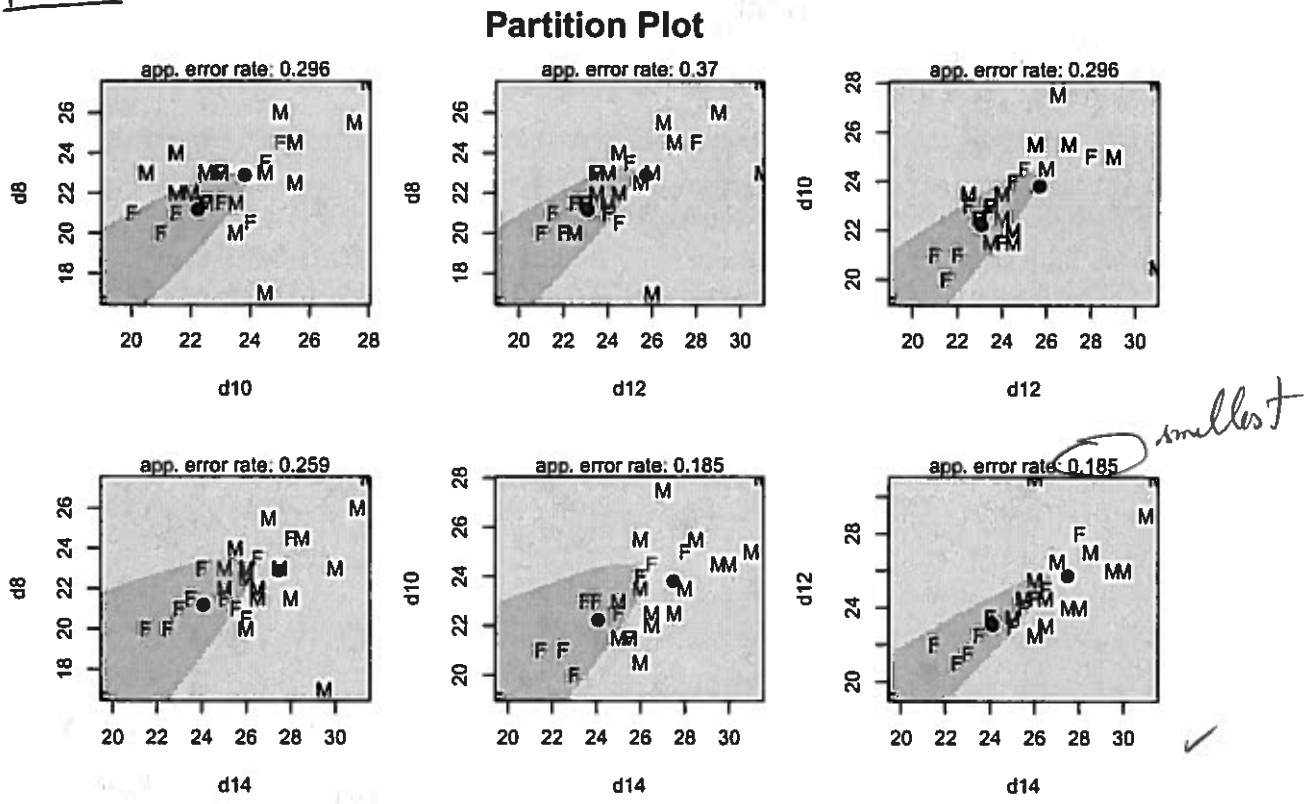
```
# check the QDA coefficients/scalings
dim(qdamod$scaling)
```

```
## [1] 4 4 2
```

```
dnames <- dimnames(qdamod$scaling)
dnames
```

```
## [[1]]
## [1] "d8"  "d10" "d12" "d14"
##
## [[2]]
## [1] "1" "2" "3" "4"
##
## [[3]]
## [1] "Male"   "Female"
```

```
# visualize the QDA partitions
partimat(Sex ~ ., data=d2, method="qda")
```



4. Evaluating Classification Functions

One important way of judging the performance of any classification procedure is to calculate its "error rate" or misclassification probabilities. In general, the population parameters are unknown in practice, so we focus on approaches that can estimate the error rates from the observed data.

The **Total Probability of Misclassification (TPM)** is defined as

$$TPM = p_1 \int_{R_2} f_1(\mathbf{x})d\mathbf{x} + p_2 \int_{R_1} f_2(\mathbf{x})d\mathbf{x}$$

for any classification rule (region) that partitions $\Omega = R_1 \cup R_2$. The **Optimum Error Rate (OER)** is the minimum possible value of TPM

$$OER = \min_{R_1, R_2} TMP(R_1, R_2) \quad \text{subject to} \quad \Omega = R_1 \cup R_2$$

which is obtained when

$$R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2}{p_1}$$

and

$$R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2}{p_1}.$$

Note that if $c(2 \mid 1) = c(1 \mid 2)$, then minimizing TPM is same as minimizing ECM.

The error rates require knowledge of the (typically unknown) parameters that define the densities $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. For example, in LDA, calculating OER requires μ_1, μ_2 and Σ. We replace these quantities in the allocation rules and evaluate the **actual error rate (AER)**,

practically impossible to find f_1 & f_2 apparent
theoretical possible

$$AER(\hat{R}_1, \hat{R}_2) = p_1 \int_{\hat{R}_2} f_1(\mathbf{x})d\mathbf{x} + p_2 \int_{\hat{R}_1} f_2(\mathbf{x})d\mathbf{x}$$

where $\hat{R}_1$ and $\hat{R}_2$ are the classification regions determined by samples of size n_1 and n_2 repectively.

The AER indicates how the sample classification function will perform in future samples. Like the OER, it cannot be calculate because it depends on unknown densities f_1 and f_2.

There is a measure of performance that does not depend on the form of populations and that can be calculated for any classification procedue. This measure is called the **apparent error rate (APER)** which is estimated using the observed (training) sample of data. The APER can be calculted from the **confussion matrix**.

Table 1: Confussion Matrix

		Classified as		
		π_1	π_2	
Truth	π_1	n_{C2} correct	n_{M1} mis	n_1 total
	π_2	n_{M2} mis	n_{C2} correct	n_2

where - n_{1C} is the number of π_1 items correctly classified as π_1 items

- n_{1M} is the number of π_1 items misclassified as π_2 items
- n_{2C} is the number of π_2 items correctly classified
- n_{2M} is the number of π_2 items misclassified

Then

$$APER = \frac{n_{1M} + n_{2M}}{n_1 + n_2}$$

Expected actual error rate

$$E(AER) = \frac{n_1 m^{(H)} + n_2 m^{(H)}}{n_1 + n_2}.$$

which is the **total proportion of misclassified** sample observations.

Leave-one-out Cross-Validation

Lachenbruch and Mickey proposed a better approach to estimate the AER:

1. Population 1 (for $i = 1, \cdots, n_1$)

(a). Hold out the i^{th} observation from π_1 and build classification rule

(b). Use classification rule from Step 1(a) to classify the i^{th} observation

2. Population 2 (for $i = 1, \cdots, n_2$)

(a). Hold out the i^{th} observation from π_2 and build classification rule

(b). Use classification rule from Step 2(a) to classify the i^{th} observation

Traing data ___ fit the model ___ error of training data
Testing data ___ ___ ___ testing data.

An (almost) unbiased estimate of the expected AER is given by

$$\hat{E}(AER) = \frac{n_{1M}^* + n_{2M}^*}{n_1 + n_2}$$

where n_{1M}^* and n_{2M}^* are the number of misclassified observations using the above "leave-one-out" procedure.

* library Lda

Revisit the wine data example: LDA

Use confusion to get AER
confusionCV to get E(AER)

```
# make confusion matrix (and APER)
confusion=table(d2$Sex, predict(ldamod)$class)
confusion

##
##          Male Female
##   Male     12      4
##   Female    3      8
```

APER

```
n=sum(confusion)
aper=(n - sum(diag(confusion))) / n
aper

## [1] 0.2592593
```

leave one out method

```
# use CV to get expected AER
ldamodCV=lda(Sex ~ ., data=d2, prior=rep(1/2, 2), CV=TRUE)
confusionCV=table(d2$Sex, ldamodCV$class)
confusionCV

##
##          Male Female
##   Male     10      6
##   Female    4      7
```

$\frac{n_{1M} + n_{2M}}{n_1 + n_2}$ E(AER)

```
eaer=(n - sum(diag(confusionCV))) / n
eaer

## [1] 0.3703704
```

when $c(2|1) = c(1|2)$

```
# split into separate matrices for each flower
X1=subset(d2, Sex=="Male")
X2=subset(d2, Sex=="Female")

# split into training and testing
set.seed(1)
id1=sample.int(n=16, size=12)
id2=sample.int(n=11, size=8)
Xtrain=rbind(X1[id1,], X2[id2,])
Xtest=rbind(X1[-id1,], X2[-id2,])
```

training data.

```
# fit lda to training and evaluate on testing
ldatrain=lda(Sex ~ ., data=Xtrain, prior=rep(1/2, 2))
confusionTest=table(Xtest$Sex, predict(ldatrain, newdata=Xtest)$class)
confusionTest

##
##          Male Female
##   Male      1      3
##   Female    1      2
```

```
n=sum(confusionTest)
aer=(n - sum(diag(confusionTest))) / n
aer
```

```
## [1] 0.5714286
```

high error

```
# split into training and testing (100 splits)
nrep=100
aer=rep(0, nrep)
set.seed(1)
for(k in 1:nrep){
  #cat("rep:",k,"\n")
  id1=sample.int(n=16, size=12)
  id2=sample.int(n=11, size=8)
  Xtrain=rbind(X1[id1,], X2[id2,])
  Xtest=rbind(X1[-id1,], X2[-id2,])
  ldatrain=lda(Sex ~ ., data=Xtrain, prior=rep(1/2, 2))
  confusionTest=table(Xtest$Sex, predict(ldatrain, newdata=Xtest)$class)
  confusionTest
  n=sum(confusionTest)
  aer[k]=(n - sum(diag(confusionTest))) / n
}
mean(aer)
```

using QDA

```
## [1] 0.3614286
```

Using QDA

```
# make confusion matrix (and APER)
confusion=table(d2$Sex, predict(qdamod)$class)
confusion
```

```
##
##          Male Female
##   Male     12      4
##   Female    1     10
```

```
n=sum(confusion)
aper=(n - sum(diag(confusion))) / n
aper
```

```
## [1] 0.1851852
```

E(AER)

```
# use CV to get expected AER
qdamodCV=qda(Sex ~ ., data=d2, prior=rep(1/2, 2), CV=TRUE)
confusionCV=table(d2$Sex, qdamodCV$class)
confusionCV
```

```
##
##          Male Female
##   Male      9      7
##   Female    3      8
```

```
eaer = (n - sum(diag(confusionCV))) / n
eaer
```

```
## [1] 0.3703704
```

```
# split into training and testing
set.seed(1)
id1=sample.int(n=16, size=12)
id2=sample.int(n=11, size=8)
Xtrain=rbind(X1[id1,], X2[id2,])
Xtest=rbind(X1[-id1,], X2[-id2,])

# fit qda to training and evaluate on testing
qdatrain=qda(Sex ~ ., data=Xtrain, prior=rep(1/2, 2))
confusionTest= table(Xtest$Sex, predict(qdatrain, newdata=Xtest)$class)
confusionTest

##
##          Male Female
##   Male      4      0
##   Female    2      1

n=sum(confusionTest)
aer=(n - sum(diag(confusionTest))) / n
aer

## [1] 0.2857143

# split into training and testing (100 splits)
nrep=100
aer=rep(0, nrep)
set.seed(1)
for(k in 1:nrep){
  #cat("rep:",k,"\n")
  id1=sample.int(n=16, size=12)
  id2=sample.int(n=11, size=8)
  Xtrain=rbind(X1[id1,], X2[id2,])
  Xtest=rbind(X1[-id1,], X2[-id2,])
  qdatrain=qda(Sex ~ ., data=Xtrain, prior=rep(1/2, 2))
  confusionTest=table(Xtest$Sex, predict(qdatrain, newdata=Xtest)$class)
  confusionTest
  n=sum(confusionTest)
  aer[k]=(n - sum(diag(confusionTest))) / n
}
mean(aer)

## [1] 0.3971429
```

Example: Classifying gender using dental data

```
library(reshape) # to create data frame from dental data
library(MASS) # has lda and qda functions
library(heavy) # has dental data
data(dental)
d2=cast(melt(dental,id=c("Subject","age","Sex")),Subject+Sex~age)
names(d2)[3:6]=c("d8","d10","d12","d14")
# these functions estimate pi_i using sample proportions
# you can provide other pi_i if needed
f1=lda(x=as.matrix(d2[,3:6]),grouping=d2[,2],CV=T)
```

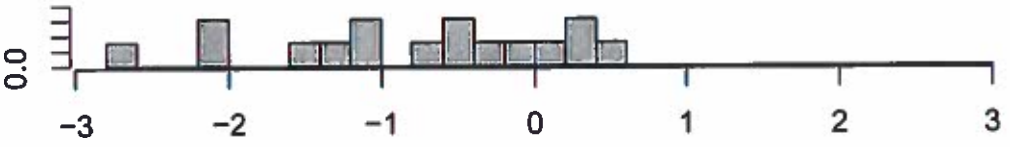
```
f1=lda(Sex~d8+d10+d12+d14,data=d2,CV=T)
f2=qda(Sex~d8+d10+d12+d14,data=d2,CV=T)
sum(f1$class!=d2$Sex)/length(d2$Sex) # CV error linear
```

```
## [1] 0.2962963
```

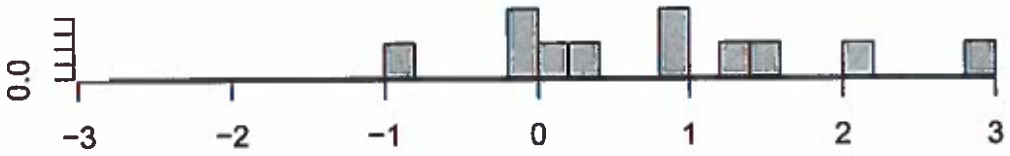
```
sum(f2$class!=d2$Sex)/length(d2$Sex) # CV error quadratic
```

```
## [1] 0.3333333
```

```
f1=lda(Sex~d8+d10+d12+d14,data=d2) # refit without CV
f2=qda(Sex~d8+d10+d12+d14,data=d2) # refit without CV
plot(f1) # uses disciminant functions
```

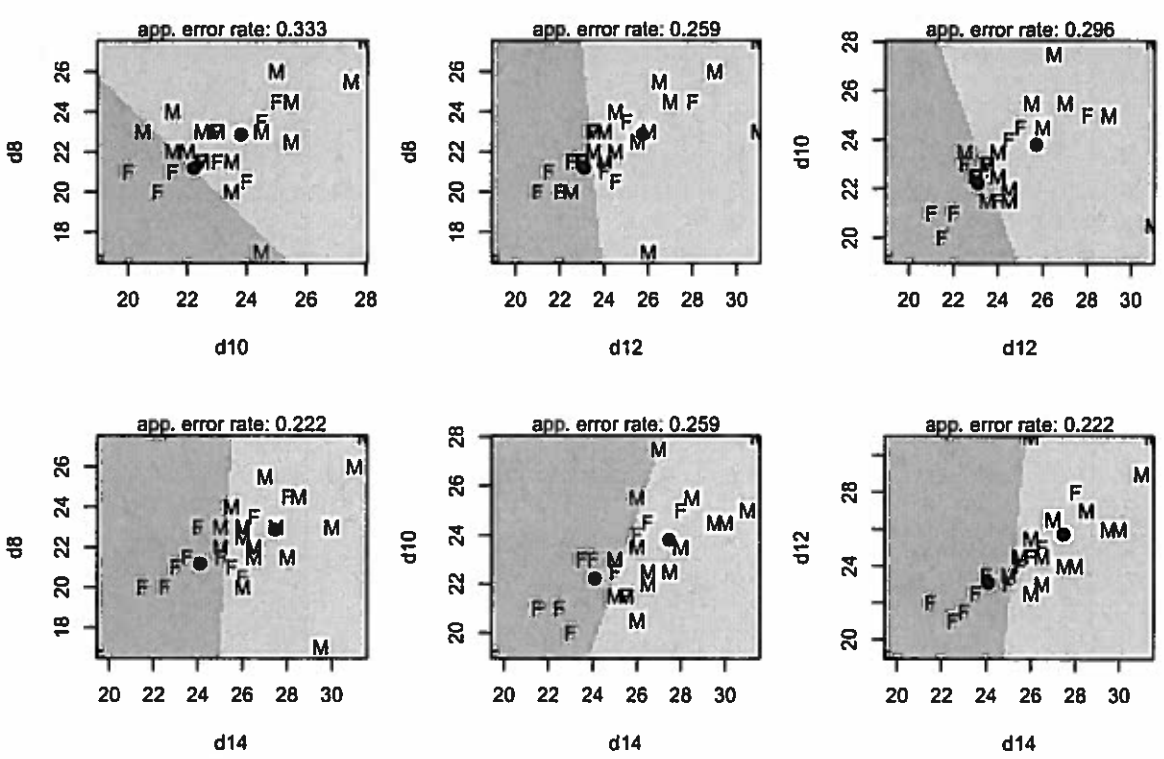


group Male

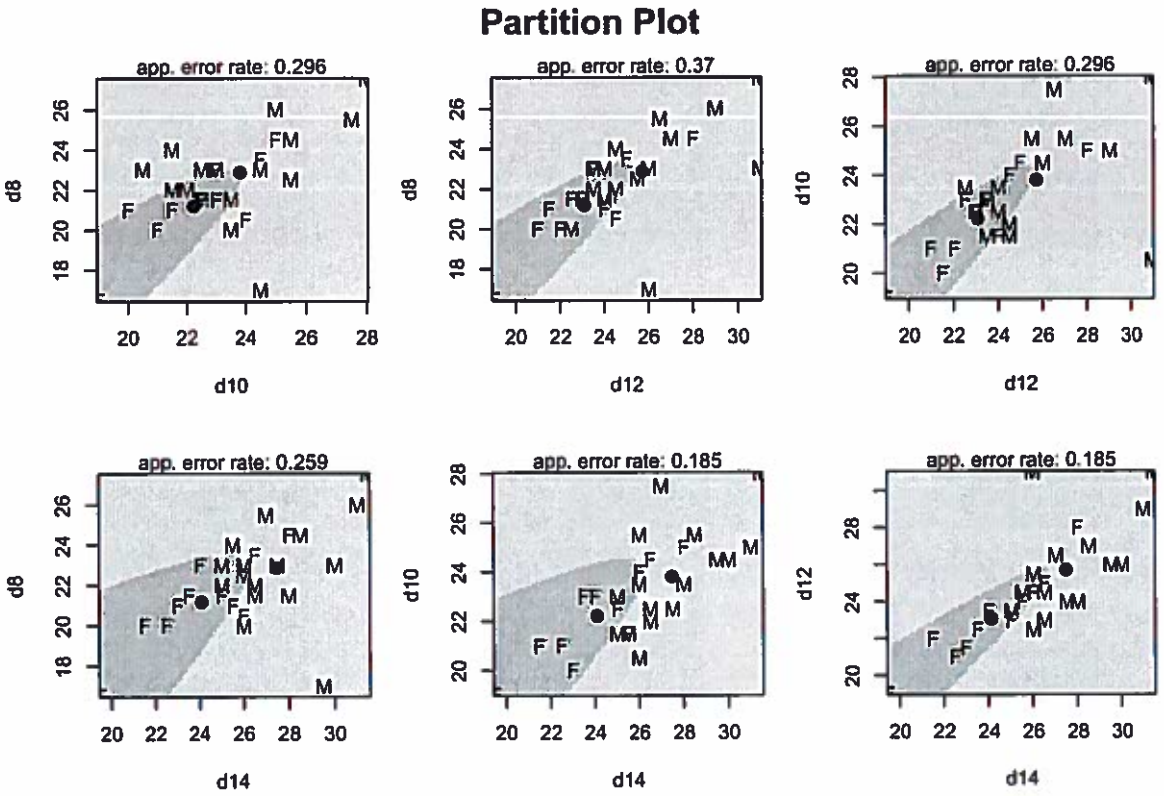


group Female

```
library(klaR) # provides panel of bivariate plots w/ regions
partimat(Sex~d8+d10+d12+d14,data=d2,method="lda")
```

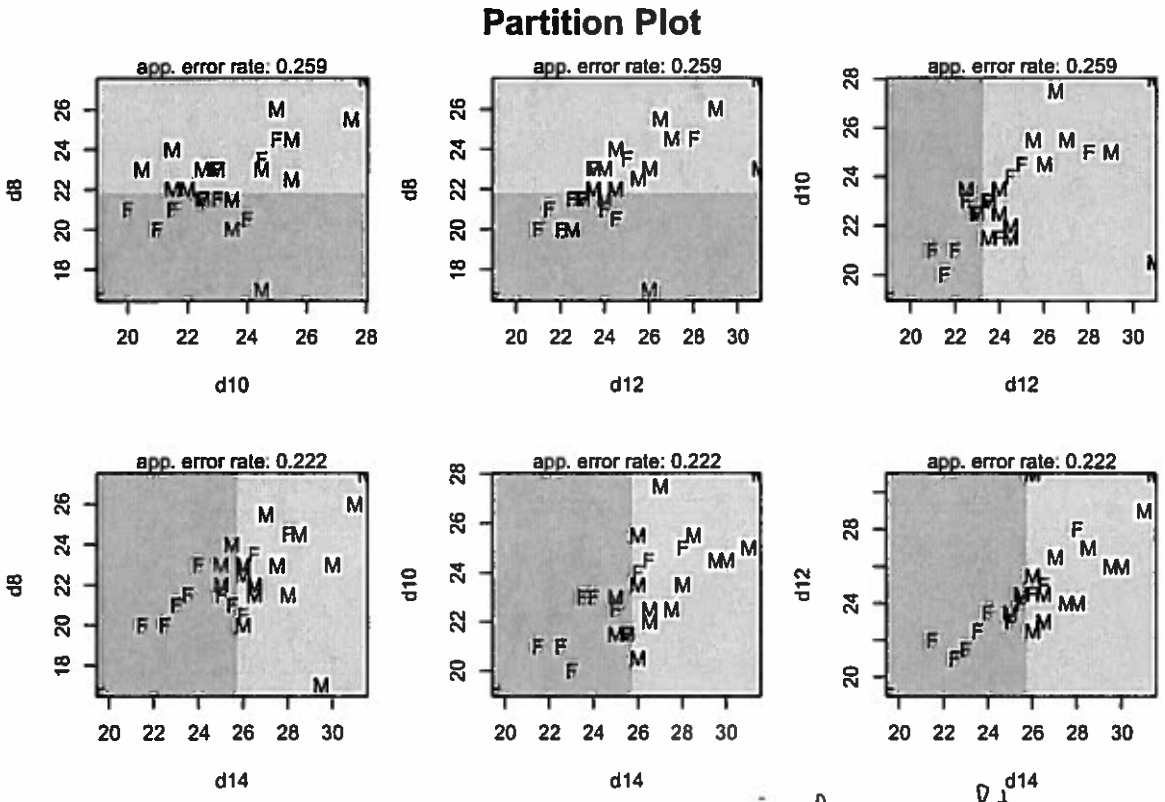



```
partimat(Sex-d8+d10+d12+d14,data=d2,method="qda")
```




```
partimat(Sex~d8+d10+d12+d14,data=d2,method="rpart") # classification tree
```

```
## Loading required namespace: rpart
```



5. Classification with Several Populations

Let $\mathbf{X} = (X_1, X_2, \cdots, X_p)^T$ denote a p-dimensional random vector. Let $f_k(\mathbf{x})$ denote the probability density function (pdf) for population π_k for $k = 1, 2, \cdots, g$. Given a realization $\mathbf{X} = \mathbf{x}$, we want to assign $\mathbf{x}$ to π_1, π_2 or π_g. We want to find some classification rule to determine whether a realization $\mathbf{X} = \mathbf{x}$ should be assigned to populationπ_1 or π_2, or π_g.

Let Ω denote the sample space (collection of all possible values for $\mathbf{x}$). $R_1 \subset \Omega$ is the set of values of $\mathbf{x}$ for which we classify objects into π_1, $R_2 \subset \Omega$ is the set of values of $\mathbf{x}$ for which we classify objects into π_2, $\cdots$,$R_g \subset \Omega$ is the set of values of $\mathbf{x}$ for which we classify objects into π_g. The classification rule partitions the sample space and the classification regions are mutually exclusive, i.e., $\Omega = R_1 \cup R_2 \cup \cdots R_g$ and $R_k \cap R_l = \emptyset$ for all $k \neq l$.

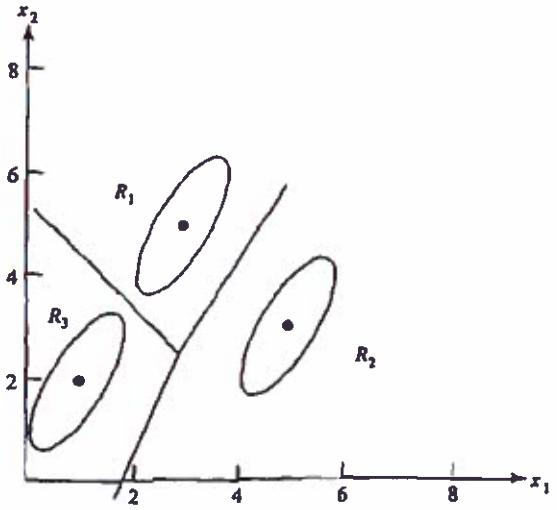


Figure 11.10 The classification regions R_1, R_2, and R_3 for the linear minimum TPM rule $(p_1 = \frac{1}{4}, p_2 = \frac{1}{2}, p_3 = \frac{1}{4})$.

The conditional probability $P(l \mid k)$ of classifying an object as π_l when the object really belongs to π_k is given by

$$P(l \mid k) = P(\mathbf{X} \in R_l \mid \pi_k) = \int_{R_l} f_k(\mathbf{x})d\mathbf{x}$$

for all $k \neq l$ with $k, l = 1, 2, \cdots, g$. Note that $P(k \mid k) = 1 - \sum_{l \neq k}(l \mid k)$ by definition. Let $c(l \mid k)$ denote the cost of allocating an object to π_l when the object really belongs to π_k , and let p_k denote the prior probability of π_k. Then the conditional expected cost of misclassifying an object from π_k is

$$ECM = \sum_{l \neq k} P(l \mid k)c(l \mid k)$$

Incorporating the prior probabilities, the overall ECM is given by

$$ECM = \sum_{k=1}^{g} p_k ECM(k) = \sum_{k=1}^{g} p_k \left[\sum_{l \neq k} P(l \mid k)c(l \mid k) \right]$$

The classification regions $\{R_1, R_2, \cdots, R_g\}$ that minimize the ECM arevdefined by allocating $\mathbf{X} = \mathbf{x}$ to the population π_k that minimizes

$$\sum_{l \neq k} p_l f_l(\mathbf{x})c(k \mid l)$$

Fisher's method for Discriminating among Several Populations

Fisher also developed his discriminant analysis for $g > 2$ populations. The motivation behind the Fisher discriminant analysis is the need to obtain a reasonable representation of the populations that involves only a few linear combination of the observations, such as $\mathbf{a}_1^T\mathbf{x}, \mathbf{a}_2^T\mathbf{x}, \mathbf{a}_3^T\mathbf{x}$. His approach offers a simple and useful procedure for classification, which also provides nice visualizations. We will be able to plot the linear combinations to visualize the discriminants.

Let $\mathbf{X} = (X_1, X_2, \cdots, X_p)^T$ denote a p-dimensional random vector and let $f_k(\mathbf{x}) \sim (\mu_k, \Sigma)$ denote the pdf for population π_k. Here we do not need the multivariate normality assumption but assume homogeneity of covariance matrix assumption. Let $\mu = \frac{1}{g}\mu_k$ denote the mean of the combined populations, and

$$\mathbf{B}_\mu = \sum_{k=1}^{g} (\mu_k - \bar{\mu})(\mu_k - \bar{\mu})^T$$

denote "Between" group sums of cross products matrix. We consider the linear combination $Y = \mathbf{a}^T\mathbf{X}$ which has expected value

$$E(Y \mid \pi_k) = \mathbf{a}^T E(\mathbf{X} \mid \pi_k) = \mathbf{a}^T \mu_k$$

and variance

$$Var(Y \mid \pi_k) = \mathbf{a}^T Cov(\mathbf{X} \mid \pi_k)\mathbf{a} = \mathbf{a}^T \mathbf{\Sigma} \mathbf{a}$$

We define the overall mean

$$\bar{\mu}_Y = \frac{1}{g}\sum_{k=1}^{g} \mu_{Y_k} = \frac{1}{g}\sum_{k=1}^{g} \mathbf{a}^T \mu_k = \mathbf{a}^T \bar{\mu}$$

and form the ratio of the between group separation over the variance of Y

$$\begin{aligned} F^* &= \frac{\sum_{k=1}^{g}(\mu_{Y_k} - \bar{\mu}_Y)^2}{\sigma_Y^2} \\ &= \frac{\sum_{k=1}^{g}(\mathbf{a}^T\mu_k - \mathbf{a}^T\bar{\mu})^2}{\mathbf{a}^T\mathbf{\Sigma}\mathbf{a}} \\ &= \frac{\mathbf{a}^T \sum_{k=1}^{g}(\mu_k - \bar{\mu})(\mu_k - \bar{\mu})^T \mathbf{a}}{\mathbf{a}^T\mathbf{\Sigma}\mathbf{a}} \\ &= \frac{\mathbf{a}^T \mathbf{B}_\mu \mathbf{a}}{\mathbf{a}^T\mathbf{\Sigma}\mathbf{a}} \end{aligned}$$

Note that higher F^* values relate to more separation between groups.

In general, $\mathbf{\Sigma}$ and the μ_i are unavailable. We use the sample data to estimated the "Between" and "Within" group sums of cross products matrices.

$$\mathbf{B} = \sum_{k=1}^{g}(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T$$

$$\mathbf{W} = \sum_{k=1}^{g}(n_k - 1)\mathbf{S}_k = \sum_{k=1}^{g}\sum_{i=1}^{n_k}(\mathbf{x}_{ki} - \bar{\mathbf{x}}_k)(\mathbf{x}_{ki} - \bar{\mathbf{x}}_k)^T$$

where the sample mean vectors $\bar{\mathbf{x}}_k = \frac{1}{n_k}\sum_{i=1}^{n_k}\mathbf{x}_{ki}$ and the "overall mean" verctor $\bar{\mathbf{x}} = \frac{1}{g}\sum_{k=1}^{g}\bar{\mathbf{x}}_k$.

Consequently, $\dfrac{\mathbf{W}}{n_1 + \cdots + n_g - g} = \mathbf{S}_{pooled}$ is the estimate of $\mathbf{\Sigma}$.

Then the sample k^{th} discriminant is the linear combination

$$\hat{Y}_k = \hat{\mathbf{a}}_k^T \mathbf{X}$$

where $\hat{\mathbf{a}}_k$ are scaled to make the $\hat{Y}_k$ have unit variance, i.e., $\hat{\mathbf{a}}_k^T \hat{\mathbf{\Sigma}} \hat{\mathbf{a}}_k = 1$ where $\hat{\mathbf{\Sigma}} = \mathbf{S}_p = \dfrac{1}{n-g}\mathbf{W}$ with $n = \sum_{k=1}^{g} n_k$.

Properties of Discriminants

Let $\mathbf{Y} = [Y_1, Y_2, \cdots, Y_s]^T = \mathbf{A}^T\mathbf{X}$, where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_s]$, then $\mathbf{Y}$ contains the s discriminants, that is, $\mathbf{Y} = [Y_1, Y_2, \cdots, Y_s]^T$. Columns of $\mathbf{A}$ contain the linear combination weights. $\mathbf{Y}$ has the following properties:

- mean of $\mathbf{Y}$: $E(\mathbf{Y} \mid \pi_k) = \mathbf{A}^T E(\mathbf{X} \mid \pi_k) = \mathbf{A}^T \mu_k = \mu_{kY}$
- covariance matrix of $\mathbf{Y}$: $Cov(\mathbf{Y}) = \mathbf{A}^T Cov(\mathbf{X} \mid \pi_k)\mathbf{A} = \mathbf{A}^T\mathbf{\Sigma}\mathbf{A} = \mathbf{I}_s$, the discriminants have unit variance and are uncorrelated.

6. Using Fisher's Discriminants to Classify Objects

Fisher's Discriminants were derived for the purpose of obtaining a low dimensional representation of the data that separates the populations as much as possible. The discriminants provide the basis for a classification rule. Let $Y_k = \mathbf{a}_k^T\mathbf{X}$ be the k^{th} discriminant, $k \leq s$, then the mean of $\mathbf{Y}$ is

$$\mu_{iY} = [\mu_{iY_1}, \mu_{iY_2}, \cdots, \mu_{iY_s}]^T = [\mathbf{a}_1^T\mu_i, \mathbf{a}_2^T\mu_i, \cdots, \mathbf{a}_s^T\mu_i]^T$$

under population π_i and the covariance matrix is $\mathbf{I}$ for all populations. Because the components of $\mathbf{Y}$ have unit variances and zero covariances, it is appropriate to measure the distance using

$$D_i = (\mathbf{y} - \mu_{iY})^T(\mathbf{y} - \mu_{iY}) = \sum_{j=1}^{s}(y_j - \mu_{iY_j})^2 = \sum_{j=1}^{s}[\mathbf{a}_j^T(\mathbf{x} - \mu_i)]^2$$

where $\mu_{iY} = \mathbf{A}^T\mu_i$, $y_j = \mathbf{a}_j^T\mathbf{x}$ and $\mu_{iY_j} = \mathbf{a}_j^T\mu_i$.

We alocate $\mathbf{x}$ to population π_k if

$$\sum_{j=1}^{r}(y_j - \mu_{kY_j})^2 = \sum_{j=1}^{r}[\mathbf{a}_j^T(\mathbf{x} - \mu_k)]^2 \leq \sum_{j=1}^{r}[\mathbf{a}_j^T(\mathbf{x} - \mu_i)]^2$$

for all $i \neq k$, $r \leq s$.

Example: Wine Recognition data

```
## Wine Recognition data in 'rattle'
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.4.4
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(MASS)
data(wine)
attach(wine)
head(wine)
```

```
##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39
##   Nonflavanoids Proanthocyanins Color  Hue Dilution Proline
## 1          0.28            2.29  5.64 1.04     3.92    1065
## 2          0.26            1.28  4.38 1.05     3.40    1050
## 3          0.30            2.81  5.68 1.03     3.17    1185
## 4          0.24            2.18  7.80 0.86     3.45    1480
## 5          0.39            1.82  4.32 1.04     2.93     735
## 6          0.34            1.97  6.75 1.05     2.85    1450
```

```
table(Type)
```

function table in IR Table(one column) ⇒ give # counts of elements.

```
## Type
##  1  2  3
## 59 71 48
```

```
g <- 3      # of groups
p <-ncol(wine) - 1      #dimension
# pooled covariances matrix
Sp <- matrix(0, p, p)
nx <- rep(0, g)
lev <- levels(wine$Type)
for(k in 1:g){
  x <- wine[wine$Type==lev[k],1:p+1]
  nx[k] <- nrow(x)
  Sp <- Sp + cov(x) * (nx[k] - 1)
}
Sp <- Sp / (sum(nx) - g)
round(Sp,3)
```

```
##                 Alcohol   Malic    Ash Alcalinity Magnesium Phenols
## Alcohol           0.262   0.008 -0.013     -0.097     0.018   0.027
## Malic             0.008   0.888  0.021      0.415    -0.889  -0.014
## Ash              -0.013   0.021  0.066      0.484     0.690   0.016
## Alcalinity       -0.097   0.415  0.484      8.007     3.238   0.109
## Magnesium         0.018  -0.889  0.690      3.238   180.658   0.577
## Phenols           0.027  -0.014  0.016      0.109     0.577   0.191
## Flavanoids        0.023  -0.009  0.030      0.246     0.667   0.161
## Nonflavanoids    -0.001   0.010  0.007      0.047    -0.287  -0.008
## Proanthocyanins   0.017   0.017  0.003      0.093     1.300   0.092
## Color             0.246  -0.259  0.010     -0.102     1.772   0.199
## Hue               0.001  -0.042  0.002     -0.010     0.126  -0.001
## Dilution         -0.005   0.047  0.011      0.226    -0.281   0.060
## Proline          12.237 -33.058 -0.501    -32.833   476.249   8.395
##                 Flavanoids Nonflavanoids Proanthocyanins  Color    Hue
## Alcohol              0.023        -0.001           0.017  0.246  0.001
## Malic               -0.009         0.010           0.017 -0.259 -0.042
## Ash                  0.030         0.007           0.003  0.010  0.002
## Alcalinity           0.246         0.047           0.093 -0.102 -0.010
## Magnesium            0.667        -0.287           1.300  1.772  0.126
## Phenols              0.161        -0.008           0.092  0.199 -0.001
## Flavanoids           0.275        -0.015           0.128  0.287 -0.004
## Nonflavanoids       -0.015         0.012          -0.009 -0.002  0.001
## Proanthocyanins      0.128        -0.009           0.246  0.229 -0.006
## Color                0.287        -0.002           0.229  2.285 -0.041
## Hue                 -0.004         0.001          -0.006 -0.041  0.024
## Dilution             0.068        -0.010           0.042 -0.066 -0.003
## Proline              3.426        -0.493          11.482 68.094  4.491
##                 Dilution   Proline
## Alcohol           -0.005    12.237
## Malic              0.047   -33.058
## Ash                0.011    -0.501
## Alcalinity         0.226   -32.833
## Magnesium         -0.281   476.249
```

```
## Phenols             0.060     8.395
## Flavanoids          0.068     3.426
## Nonflavanoids      -0.010    -0.493
## Proanthocyanins     0.042    11.482
## Color              -0.066    68.094
## Hue                -0.003     4.491
## Dilution            0.161   -10.943
## Proline           -10.943 29707.682
```

```
# fit lda model
# assume equal prior
ldamod <- lda(Type ~ ., data=wine, prior=rep(1/3, 3))
names(ldamod)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```
# check the LDA coefficients/scalings

# a_1, ...  a_p are given as "Coefficients of linear discriminants".
ldamod$scaling
```

```
##                          LD1          LD2
## Alcohol          -0.356369042  0.892051370
## Malic             0.181293364  0.296139458
## Ash              -0.243541326  2.362184415
## Alcalinity        0.146777356 -0.154421898
## Magnesium        -0.002185082 -0.000346807
## Phenols           0.615457266 -0.065102824
## Flavanoids       -1.685049247 -0.402774674
## Nonflavanoids    -1.580605983 -1.548924728
## Proanthocyanins   0.117537507 -0.313796985
## Color             0.368045785  0.233950076
## Hue              -0.897641355 -1.469888074
## Dilution         -1.153187021  0.112797172
## Proline          -0.002535348  0.002992344
```

```
crossprod(ldamod$scaling, Sp) %*% ldamod$scaling
```

```
##               LD1           LD2
## LD1  1.000000e+00 -9.436896e-16
## LD2 -1.082467e-15  1.000000e+00
```

unique solution property.

```
# create the (centered) discriminant scores
mu.k <- ldamod$means
mu <- colMeans(mu.k)
dscores <- scale(wine[,-1], center=mu, scale=F) %*% ldamod$scaling
sum((dscores - predict(ldamod)$x)^2)
```

discriminant score

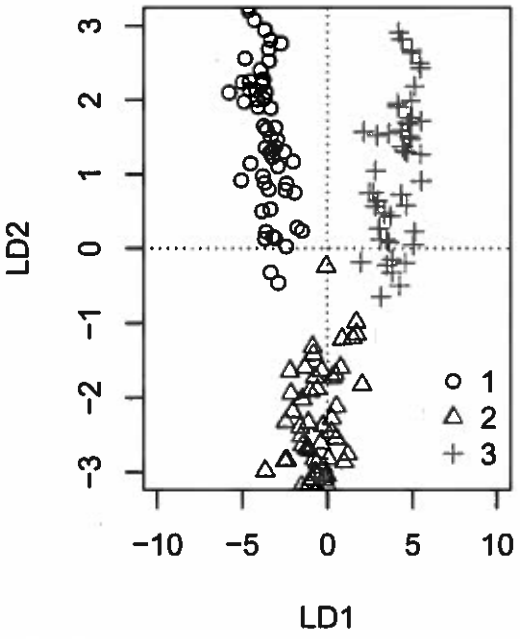
? Dē
↑ ndiy Fish.

```
## [1] 2.746106e-28
```

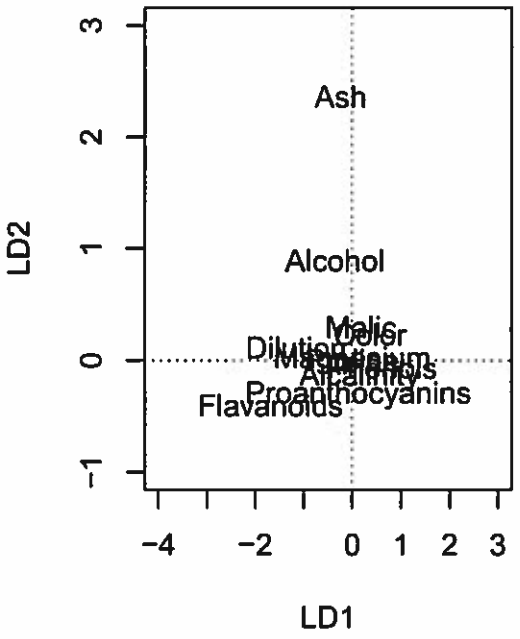
```
# plot the scores and coefficients
spid <- as.integer(wine$Type)
par(mfrow=c(1,2))
plot(dscores, xlab="LD1", ylab="LD2", pch=spid, col=spid,
     main="Discriminant Scores", xlim=c(-10, 10), ylim=c(-3, 3))
abline(h=0, lty=3)
abline(v=0, lty=3)
```

```
legend("bottomright",lev,pch=1:3,col=1:3,bty="n")
plot(ldamod$scaling, xlab="LD1", ylab="LD2", type="n",
     main="Discriminant Coefficients", xlim=c(-4, 3), ylim=c(-1, 3))
text(ldamod$scaling, labels=rownames(ldamod$scaling))
abline(h=0, lty=3)
abline(v=0, lty=3)
```

Discriminant Scores

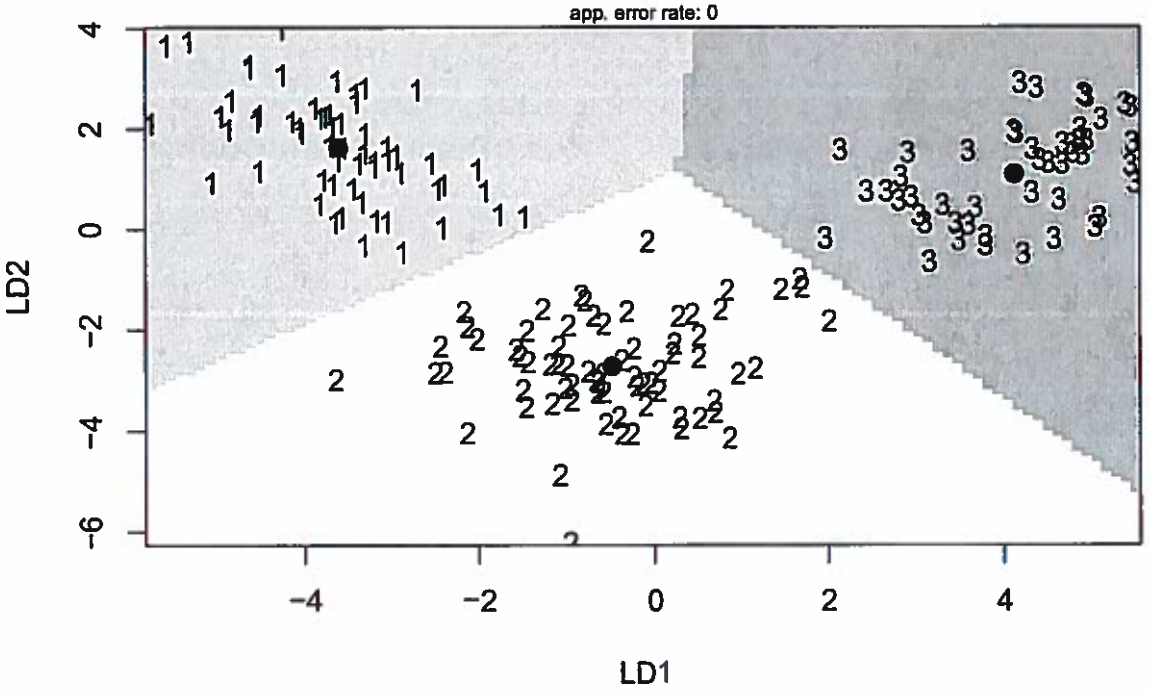


Discriminant Coefficients




```
# visualize the LDA paritions
type <- factor(Type)
library(klaR)
partimat(x=dscores[,2:1], grouping=type, method="lda")
```

Partition Plot




```
# visualize the LDA partitions (for all pairs)
#partimat(x=wine[,-1], grouping=type, method="lda")

################   QDA    ###################
# Check equal variance assumption
library(biotools)
boxM(wine[,-1],wine[,1])

##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  wine[, -1]
## Chi-Sq (approx.) = 684.2, df = 182, p-value < 2.2e-16
# fit qda model
qdamod <- qda(Type ~ ., data=wine, prior=rep(1/3, 3))
names(qdamod)

##  [1] "prior"   "counts"  "means"   "scaling" "ldet"    "lev"     "N"
##  [8] "call"    "terms"   "xlevels"
# check the QDA coefficients/scalings
dim(qdamod$scaling)

## [1] 13 13  3
dnames <- dimnames(qdamod$scaling)
dnames

## [[1]]
##  [1] "Alcohol"         "Malic"           "Ash"
```

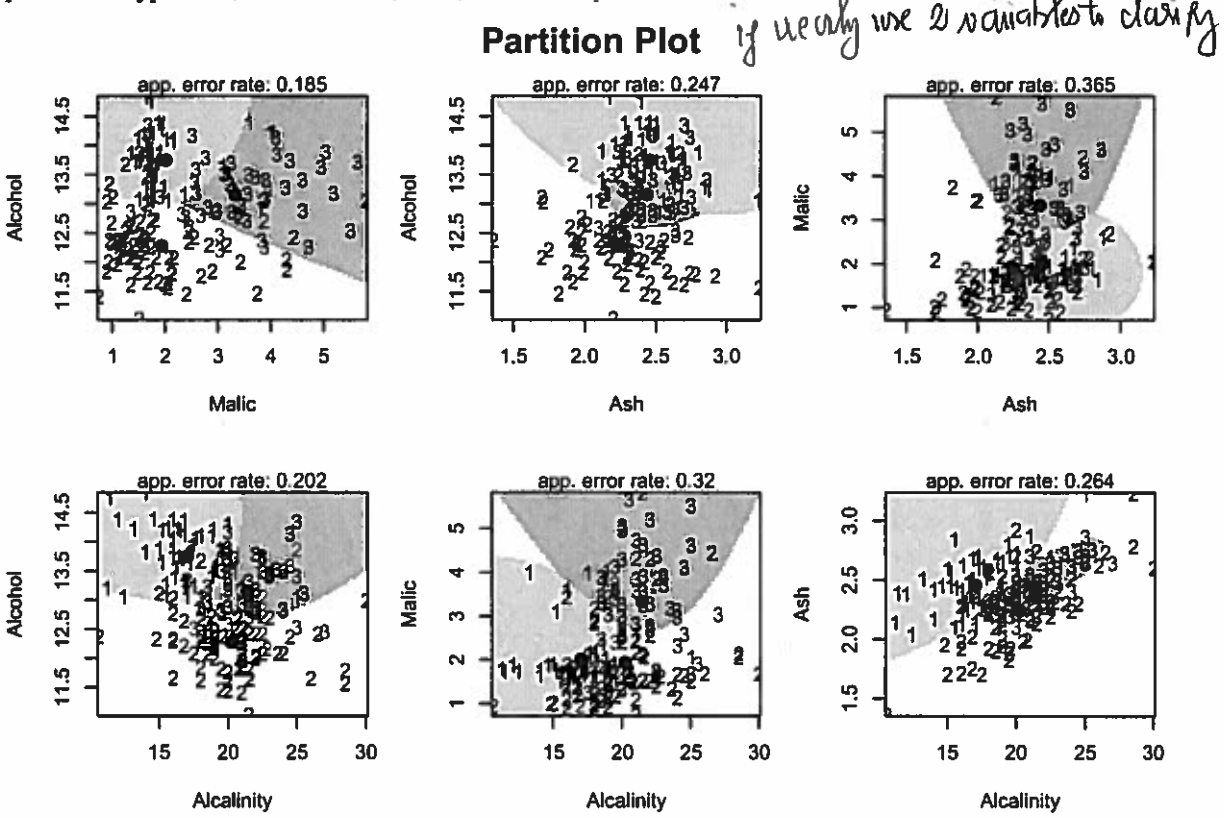
cd equal variance assumption.

→ the 3 type of wines ≠ not equal at least 2 of them have ≠ variance.

prior equally likely belong to 1/2/3.

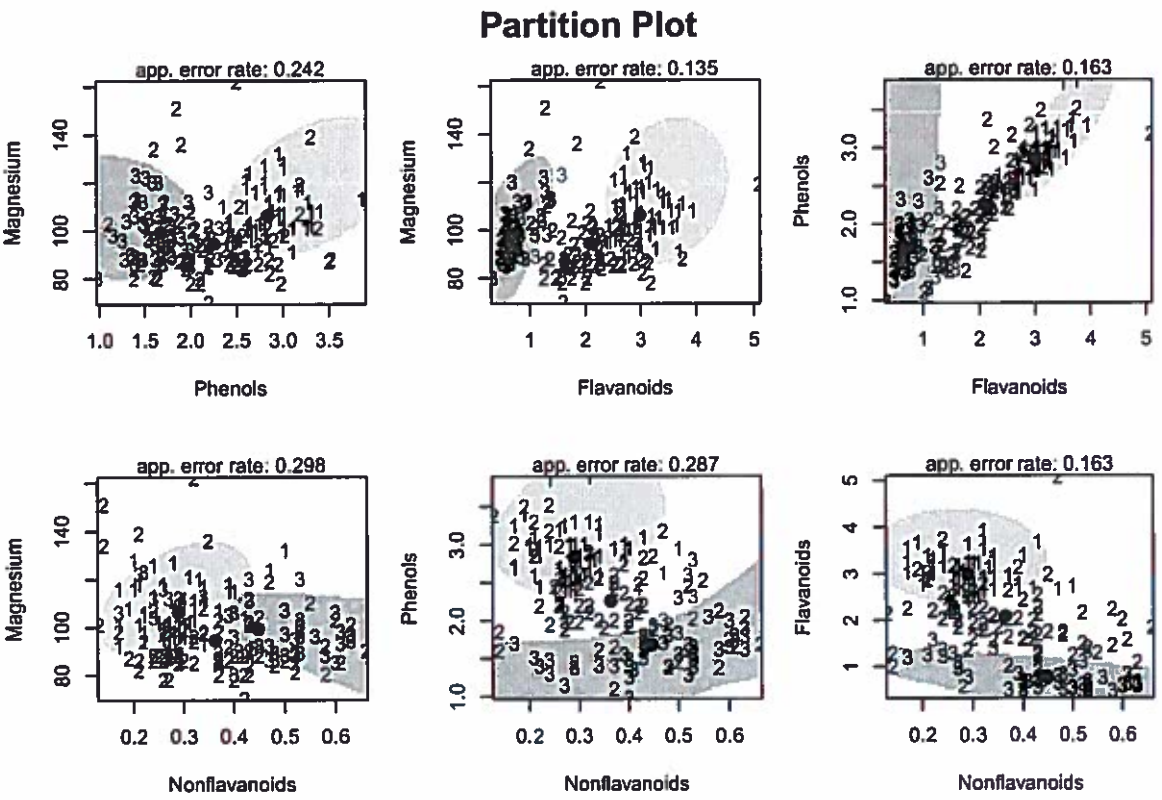
```
##  [4] "Alcalinity"      "Magnesium"       "Phenols"
##  [7] "Flavanoids"      "Nonflavanoids"   "Proanthocyanins"
## [10] "Color"           "Hue"             "Dilution"
## [13] "Proline"
##
## [[2]]
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"
##
## [[3]]
## [1] "1" "2" "3"
```

```
# visualize the QDA partitions
#partimat(Type ~ ., data=wine, method="qda") # all pairs
partimat(Type ~ ., data=wine[,2:5], method="qda")
```

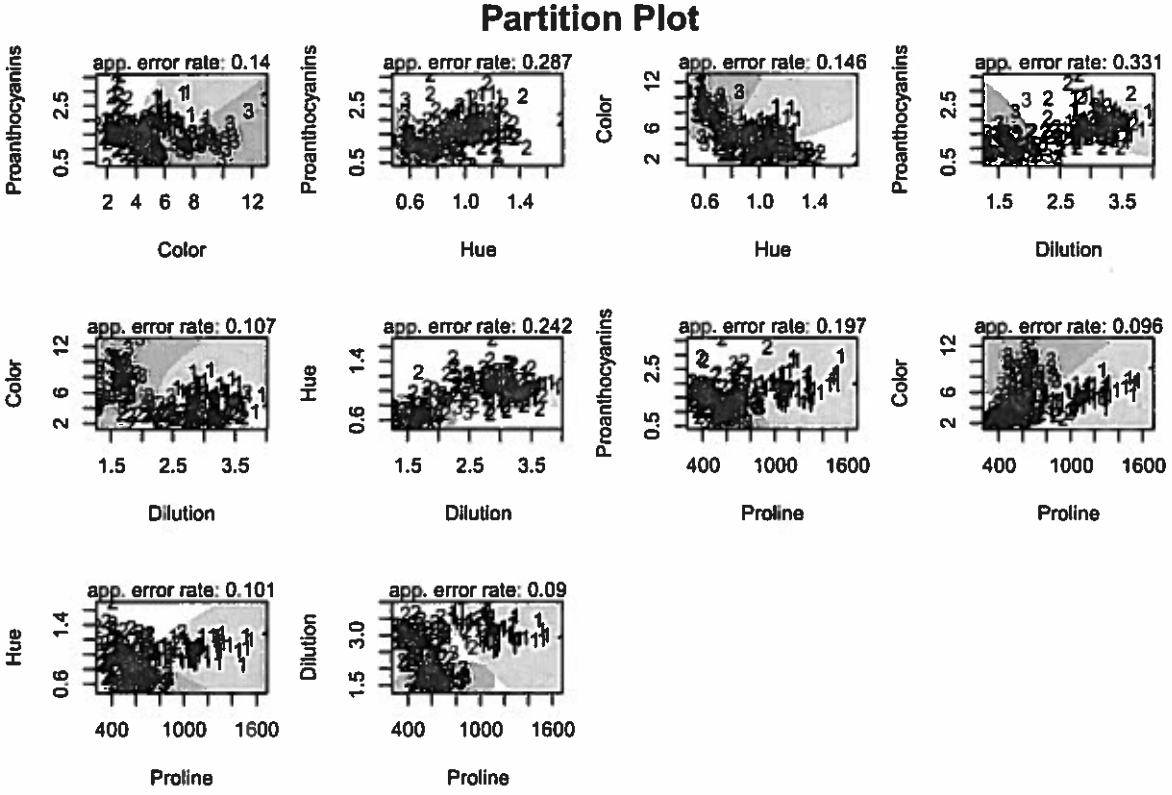


if we only use 2 variables to classify

```
partimat(Type ~ ., data=wine[,6:9], method="qda")
```




```
partimat(Type ~ ., data=wine[,10:14], method="qda")
```




```
################  CV LDA #################
# make confusion matrix (and APER)
confusion <- table(wine$Type, predict(ldamod)$class)
confusion
```

```
##
##      1  2  3
##   1 59  0  0
##   2  0 71  0
##   3  0  0 48
```

```
n <- sum(confusion)
aper <- (n - sum(diag(confusion))) / n
aper
```

```
## [1] 0
```

```
# use CV to get expected AER
ldamodCV <- lda(Type ~ ., data=wine, prior=rep(1/3, 3), CV=TRUE)
confusionCV <- table(wine$Type, ldamodCV$class)
confusionCV
```

```
##
##      1  2  3
##   1 59  0  0
##   2  1 69  1
##   3  0  0 48
```

```
eaer <- (n - sum(diag(confusionCV))) / n
eaer
```

```
## [1] 0.01123596
# OR
f1 <- lda(Type~.,data=wine,CV=T)
sum(f1$class!=wine$Type)/length(wine$Type) # CV error linear
```

Split during testing

```
## [1] 0.01123596
# split into separate matrices for each flower
X1 <- subset(wine, Type=="1")
X2 <- subset(wine, Type=="2")
X3 <- subset(wine, Type=="3")

# split into training and testing
set.seed(1)
id1 <- sample.int(n=59, size=41)
id2 <- sample.int(n=71, size=50)
id3 <- sample.int(n=48, size=34)
Xtrain <- rbind(X1[id1,], X2[id2,], X3[id3,])
Xtest <- rbind(X1[-id1,], X2[-id2,], X3[-id3,])

# fit lda to training and evaluate on testing
ldatrain <- lda(Type ~ ., data=Xtrain, prior=rep(1/3, 3))
confusionTest <- table(Xtest$Type, predict(ldatrain, newdata=Xtest)$class)
confusionTest

##
##      1  2  3
##   1 18  0  0
##   2  0 21  0
##   3  0  0 14
n <- sum(confusionTest)
aer <- (n - sum(diag(confusionTest))) / n
aer
## [1] 0 ← testing error rate ← good job! lda
# split into training and testing (100 splits)
nrep <- 100
aer <- rep(0, nrep)
set.seed(1)
for(k in 1:nrep){
  #cat("rep:",k,"\n")
  id1 <- sample.int(n=59, size=41)
  id2 <- sample.int(n=71, size=50)
  id3 <- sample.int(n=48, size=34)
  Xtrain <- rbind(X1[id1,], X2[id2,], X3[id3,])
  Xtest <- rbind(X1[-id1,], X2[-id2,], X3[-id3,])
  ldatrain <- lda(Type ~ ., data=Xtrain, prior=rep(1/3, 3))
  confusionTest <- table(Xtest$Type, predict(ldatrain, newdata=Xtest)$class)
  confusionTest
  n <- sum(confusionTest)
  aer[k] <- (n - sum(diag(confusionTest))) / n
}
mean(aer)
```

```
## [1] 0.01622642
################  CV QDA #################
# make confusion matrix (and APER)
confusion <- table(wine$Type, predict(qdamod)$class)
confusion

##
##      1  2  3
##   1 59  0  0
##   2  1 70  0
##   3  0  0 48
n <- sum(confusion)
aper <- (n - sum(diag(confusion))) / n
aper

## [1] 0.005617978
# use CV to get expected AER
qdamodCV <- qda(Type ~ ., data=wine, prior=rep(1/3, 3), CV=TRUE)
confusionCV <- table(wine$Type, qdamodCV$class)
confusionCV

##
##      1  2  3
##   1 59  0  0
##   2  1 70  0
##   3  0  0 48
eaer <- (n - sum(diag(confusionCV))) / n
eaer

## [1] 0.005617978
# OR
f2 <- qda(Type~.,data=wine,CV=T)
sum(f2$class!=wine$Type)/length(wine$Type) # CV error quadratic

## [1] 0.005617978
# split into training and testing
set.seed(1)
id1 <- sample.int(n=59, size=41)
id2 <- sample.int(n=71, size=50)
id3 <- sample.int(n=48, size=34)
Xtrain <- rbind(X1[id1,], X2[id2,], X3[id3,])
Xtest <- rbind(X1[-id1,], X2[-id2,], X3[-id3,])

# fit qda to training and evaluate on testing
qdatrain <- qda(Type ~ ., data=Xtrain, prior=rep(1/3, 3))
confusionTest <- table(Xtest$Type, predict(qdatrain, newdata=Xtest)$class)
confusionTest

##
##      1  2  3
##   1 17  1  0
##   2  0 21  0
##   3  0  0 14
```

```
n <- sum(confusionTest)
aer <- (n - sum(diag(confusionTest))) / n
aer
```

```
## [1] 0.01886792
```

```
# split into training and testing (100 splits)
nrep <- 100
aer <- rep(0, nrep)
set.seed(1)
for(k in 1:nrep){
  #cat("rep:",k,"\n")
  id1 <- sample.int(n=59, size=41)
  id2 <- sample.int(n=71, size=50)
  id3 <- sample.int(n=48, size=34)
  Xtrain <- rbind(X1[id1,], X2[id2,], X3[id3,])
  Xtest <- rbind(X1[-id1,], X2[-id2,], X3[-id3,])
  qdatrain <- qda(Type ~ ., data=Xtrain, prior=rep(1/3, 3))
  confusionTest <- table(Xtest$Type, predict(qdatrain, newdata=Xtest)$class)
  confusionTest
  n <- sum(confusionTest)
  aer[k] <- (n - sum(diag(confusionTest))) / n
}
mean(aer)
```

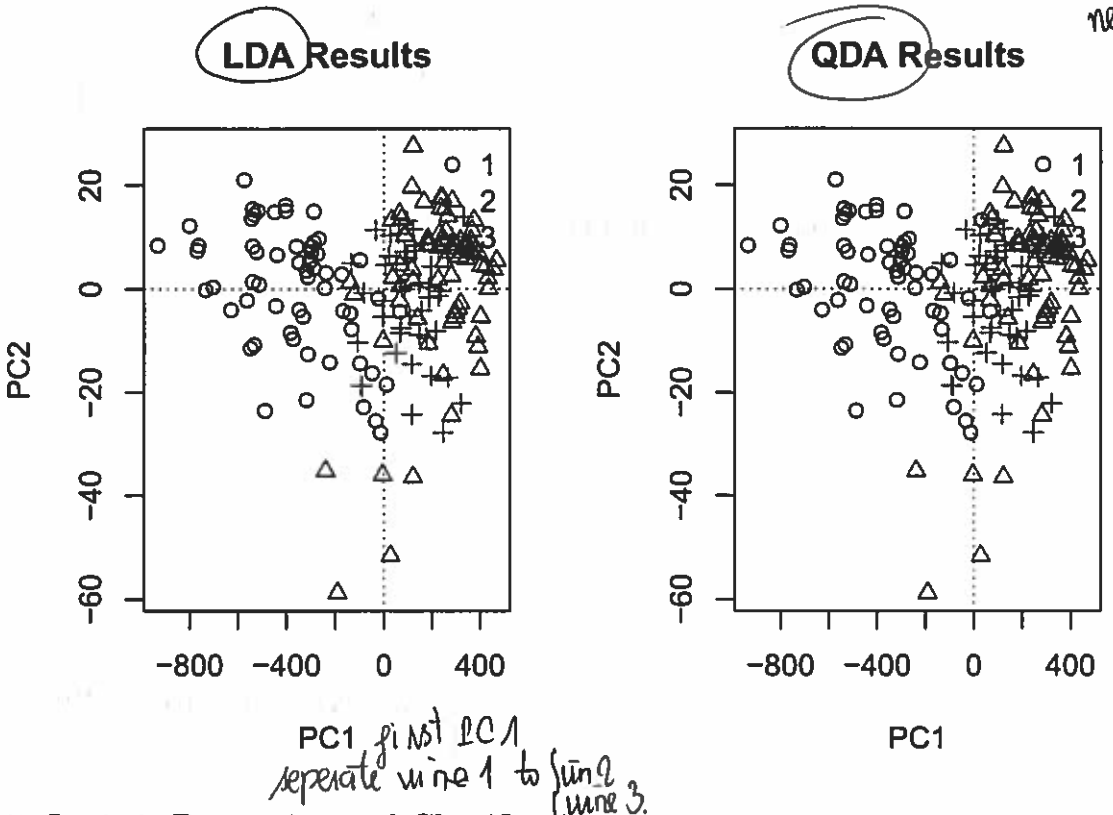
```
## [1] 0.01528302
```

using PCA

```
############### visualize LDA and QDA results via PCA ###############
ldaid <- as.integer(predict(ldamod)$class)
qdaid <- as.integer(predict(qdamod)$class)
pcamod <- princomp(wine[,-1])
par(mfrow=c(1,2))
plot(pcamod$scores[,1:2], xlab="PC1", ylab="PC2", pch=ldaid, col=ldaid,main="LDA Results")
legend("topright",lev,pch=1:3,col=1:3,bty="n")
abline(h=0,lty=3)
abline(v=0,lty=3)
plot(pcamod$scores[,1:2], xlab="PC1", ylab="PC2", pch=qdaid, col=qdaid,main="QDA Results")
legend("topright",lev,pch=1:3,col=1:3,bty="n")
abline(h=0,lty=3)
abline(v=0,lty=3)
```

why we PCs

principle comp
pc scores apply linear regression on the new scores



first LC1
seperate wine 1 to wine 2 wine 3.

7. Logistic Regression and Classification

When the number of groups $g = 2$, an alternative to model-based classification is logistic regression. Assume that $Y_i \in \{0, 1\}$, There are two classification probabilities $p(Y = 1)$ and $1 - p(Y = 0)$. The model is

$$\log \frac{p}{1-p} = \beta_0 + \beta^T \mathbf{x}$$

? model based vs model free in machine learning.

where $\beta^T = (\beta_1, \cdots, \beta_p)^T$. The **odds ratio**

$$odds = \frac{p}{1-p}$$ ← success prop ← failure prop

which is the ratio of the probability of 1 to be the probability of 0. The logistic model can be written as

$$p(\mathbf{x}) = \frac{\exp(\beta_0 + \beta^T \mathbf{x})}{1 + \exp(\beta_0 + \beta^T \mathbf{x})}$$

Note that $\mathbf{x}$ is allocated to the $Y = 1$ group when the odds ratio is greater than 1. Equivalently, we have simple linear discriminant rule

$$\hat{\beta}_0 + \hat{\beta}^T \mathbf{x} > 0$$

which is similar to Fisher's linear rule.

A more powerful version is additive logistic regression

$$\log \frac{p}{1-p} = \beta_0 + g_1(x_1) + \cdots + g_p(x_p)$$

This can provide nonlinear boundaries in $\mathcal{R}_p$. Additive logistic regression is available in gam. There is also locally-weighted logistic regression and kernel logistic regression for nonlinear classification. Logistic regression can be thought of as providing a "nonparametric" linear discriminant rule.

<- Can be used in final project .

Example:

math reading writing .

In the satgradu data, the first three columns are the SAT scores on the 3 tests, the last column is an indicator of whether the student successfully graduated (1 = graduated, 0 = did not graduate)

```
satgradu <- read.table("satgradu.txt", header=T)
attach(satgradu)
```

logister regression model

```
glm.fit.sat <- glm(gradu ~ math + reading + writing, data=satgradu, family=binomial)
summary(glm.fit.sat)

##
## Call:
## glm(formula = gradu ~ math + reading + writing, family = binomial,
##     data = satgradu)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.95784   0.04024   0.65174   0.78397   1.09748
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.4940606  2.5348993  -0.589    0.556
## math         0.0012308  0.0047303   0.260    0.795
## reading      0.0037698  0.0034071   1.106    0.269
## writing      0.0002277  0.0052656   0.043    0.966
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.987  on 39  degrees of freedom
## Residual deviance: 43.116  on 36  degrees of freedom
## AIC: 51.116
##
## Number of Fisher Scoring iterations: 4
```

not any are significant .

biggest far away from 0 ⇒ more significant in predicted graduation .

⇒ more important factor to predict graduation rate .

The z-tests tell us which explanatory variables are most important in predicting whether a student graduates. We see the reading score is the most important explanatory variable is predicting graduation.

Now let us predict whether a new applicant with SAT scores of: math = 550, reading = 610, writing = 480 will graduate:

```
newobs <- rbind( c(550,610,480) )
dimnames(newobs) <- list(NULL,c('math','reading', 'writing'))
newobs <- data.frame(newobs)
predict(glm.fit.sat,newdata=newobs, type='response')

##         1
## 0.830873
```

predict for the new data

← high probability of graduate .

Making predictions for several new individuals at once:

```
newobs <- rbind( c(300,420,280), c(510,480,470), c(780,760,710) )
dimnames(newobs) <- list(NULL,c('math','reading', 'writing'))
newobs <- data.frame(newobs)
predict(glm.fit.sat,newdata=newobs, type='response')

##         1         2         3
## 0.6276845 0.7408203 0.9236341
```

Check the misclassification rate of logistic regression rule:

1. Simple plug-in misclassification rate:

```
group.probs<-predict(glm.fit.sat, satgradu, type='response')
pred.group <- rep(0,times=nrow(satgradu))
pred.group[group.probs > 0.5] <- 1
table(pred.group,gradu)
```

```
##            gradu
## pred.group  0  1
##          1 10 30
```

The plug-in misclassification rate for logistic regression here is $10/40 = 0.25$, but all the students have been predicted to graduate! If we change the cutoff to 0.6, then two people are predicted to not graduate, and the misclassification rate is still $(1 + 9)/40 = 0.25$

```
group.probs<-predict(glm.fit.sat, satgradu, type='response')
pred.group <- rep(0,times=nrow(satgradu))
pred.group[group.probs > 0.6] <- 1
table(pred.group,gradu)
```

```
##            gradu
## pred.group  0  1
##          0  1  1
##          1  9 29
```

2. cross-validation rate of logistic regression rule:

```
correct<-rep(0,times=nrow(satgradu) )
for (j in 1:nrow(satgradu) ) {
  glm.fit.no.j<-glm(gradu ~ math + reading + writing, data=satgradu, family=binomial, subset=-j)
  mypred<-(predict(glm.fit.no.j,newdata=satgradu[j,1:3],typ='response') > 0.5)
  correct[j] <- (mypred==gradu[j])
}
cv.misclass <- 1-mean(correct)
cv.misclass
```

```
## [1] 0.275
```

The cross-validation misclassification rate for logistic regression here is 0.275.

Logistic regression with high dimensional covariates

reduce # of factors that affect the outcome

to shrink some parameter λ.

The LASSO (least absolute shrinkage and selection operator) and elastic net are two regularized regression approaches. "Regularization" broadly means to impose contraints to solve overparameterized problems. LASSO and elastic net both shrink logistic regression coefficients toward zero, and both are formulated as penalized regression. The LASSO maximizes

$$L_\lambda(\beta) = \frac{1}{2}\log\left[\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}\right] - \lambda\sum_{j=1}^{p} |\beta_j|$$

where $p_i = \dfrac{\exp(\beta_o + \beta^T \mathbf{x}_i)}{1+\exp(\beta_o + \beta^T \mathbf{x}_i)}$ and $\beta = (\beta_1, \cdots, \beta_p)^T$. Often λ is chosen through k-fold cross-validation, i.e. chosen to minimize prediction error. Typically covariates are standardized to have unit variance before using LASSO.

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:reshape':
##
##     expand
```

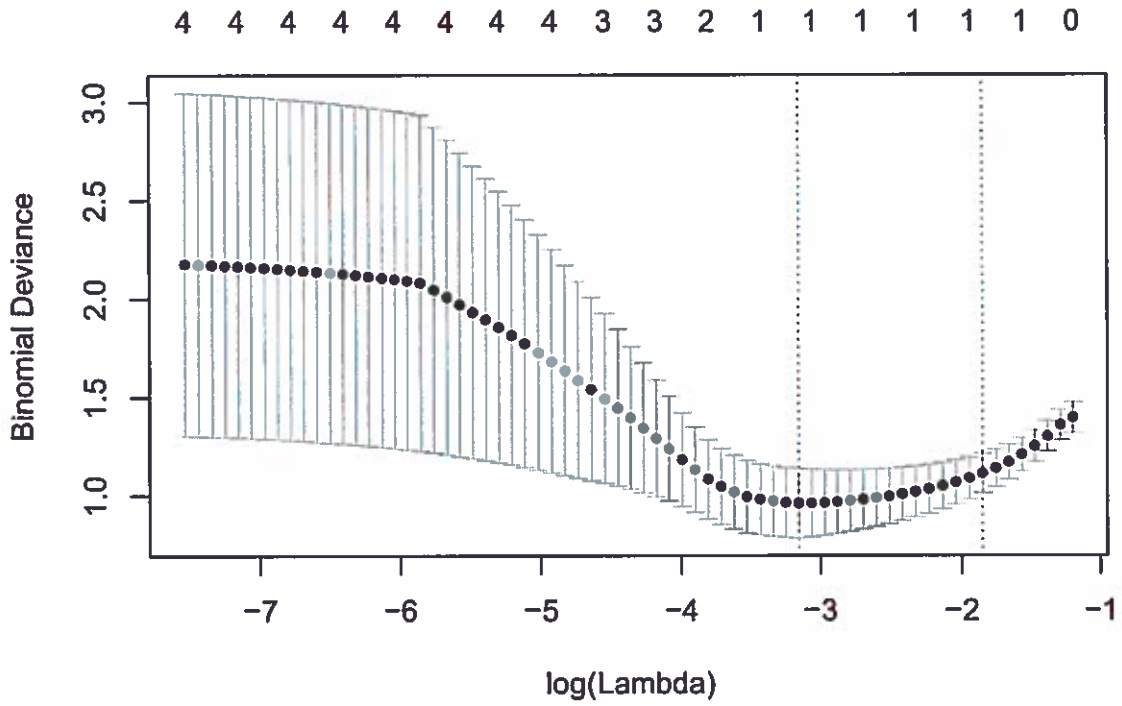
```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.4.3
```

```
## Loaded glmnet 2.0-16
```

select λ to classification

```
f <- cv.glmnet(as.matrix(d2[,3:6]),d2[,2],family="binomial",grouped=FALSE)
plot(f)
```



The plot shows that the log of the optimal value of lambda (i.e. the one that minimizes the root mean square error) is approximately -3. The exact value can be viewed by examining the variable 'lambda_min' in the code below.

ex dental data set
predict gender based on teeth measurement.

```
#min value of lambda
lambda_min <- f$lambda.min
lambda_min
```

```
## [1] 0.04257696
```

The cv.glmnet function finds the value of lambda that gives the simplest model but also lies within one standard error of the optimal value of lambda. This value of lambda (lambda.1se) is what we'll use in the rest of the computation.

```
#best value of lambda
lambda_1se <- f$lambda.1se
lambda_1se
```

the best lambda.

```
## [1] 0.1566142
#regression coefficients
coef(f,s=lambda_1se)
```

direct plugin.

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) -5.8427230
## V1           .
## V2           .
## V3           .
## V4          -0.2395536
f2 <- glmnet(as.matrix(d2[,3:6]),d2[,2],family="binomial")
pred.sex <-predict(f2,newx=as.matrix(d2[,3:6]),s=lambda_1se,type="class")
table(pred.sex,d2$Sex)
```

← only the last variable being used.

Know which factor is the most important factor to decide a person has or does not have diabetes.

```
##
## pred.sex Male Female
##   Female    0      6
##   Male     16      5
```

The misclassification error is $5/27 = 0.185$

leave 1 out

```
cverror=0
for(i in 1:dim(d2)[1]){
lambda=cv.glmnet(as.matrix(d2[-i,3:6]),d2[-i,2],family="binomial",grouped=FALSE)$lambda_1se
f=glmnet(as.matrix(d2[-i,3:6]),d2[-i,2],family="binomial")
pred.sex=predict(f,newx=as.matrix(d2[i,3:6]),s=lambda_1se,type="class")
if(d2[i,2]!=pred.sex){cverror=cverror+1}
}
```

```
## Warning in lognet(x, is.sparse, ix, jx, y, weights, offset, alpha, nobs, :
## one multinomial or binomial class has fewer than 8 observations; dangerous
## ground
```

```
## Warning in lognet(x, is.sparse, ix, jx, y, weights, offset, alpha, nobs, :
## one multinomial or binomial class has fewer than 8 observations; dangerous
## ground
```

```
cverror/dim(d2)[1]
```

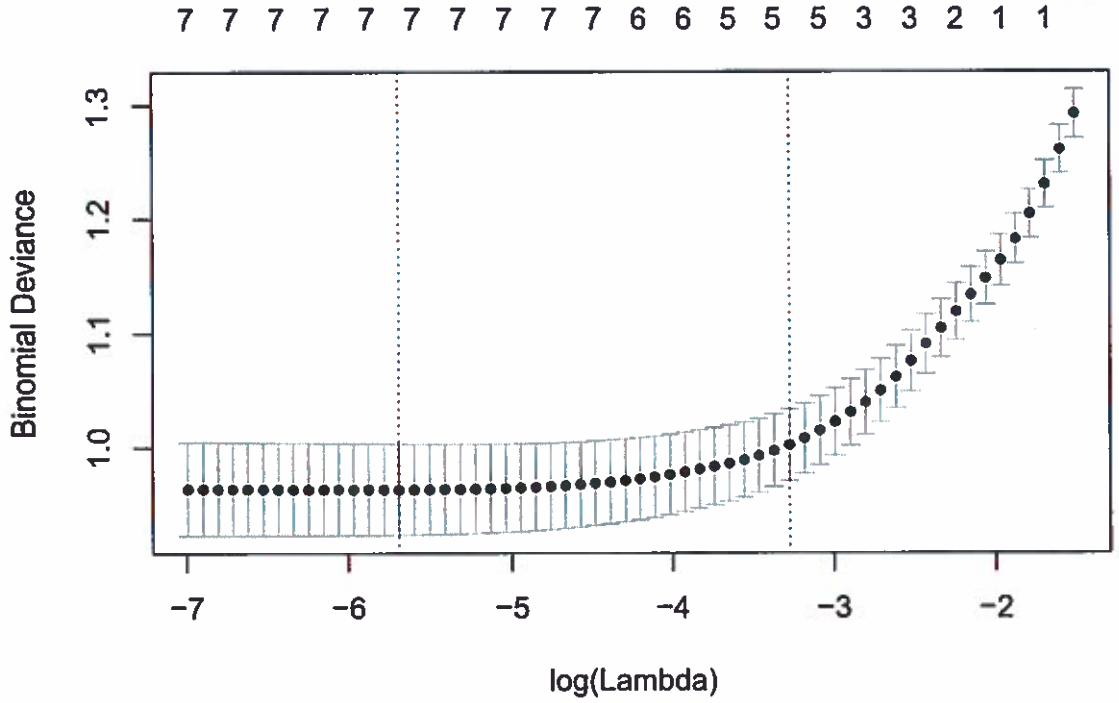
```
## [1] 0.2222222
```

We use cross-validation to find $\hat{\lambda}$. Note that LASSO can shrink coefficients to zero. That is why it is also useful for variable selection. It provides an "automatic" stepwise procedure. As $\lambda \to \infty$, the usual MLE estimates from logistic regression are obtained.

Example Pima Indian Diabetes dataset from the 'mlbench' package

```
library(mlbench)
data("PimaIndiansDiabetes")
attach(PimaIndiansDiabetes)
```

```
## The following object is masked from package:datasets:
##
##     pressure
head(PimaIndiansDiabetes)
##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1        6     148       72      35       0 33.6    0.627  50      pos
## 2        1      85       66      29       0 26.6    0.351  31      neg
## 3        8     183       64       0       0 23.3    0.672  32      pos
## 4        1      89       66      23      94 28.1    0.167  21      neg
## 5        0     137       40      35     168 43.1    2.288  33      pos
## 6        5     116       74       0       0 25.6    0.201  30      neg
library(glmnet)
f <- cv.glmnet(as.matrix(PimaIndiansDiabetes[,-9]),
               PimaIndiansDiabetes[,9],
               family="binomial",grouped=FALSE)
plot(f)
```




```
#min value of lambda
lambda_min <- f$lambda.min
lambda_min
## [1] 0.003380156
#best value of lambda
lambda_1se <- f$lambda.1se
lambda_1se
## [1] 0.03797011
#regression coefficients
coef(f,s=lambda_min)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept) -8.0790626793
## pregnant     0.1162781653
## glucose      0.0336913048
## pressure    -0.0111001234
## triceps      .
## insulin     -0.0008656466
## mass         0.0838838054
## pedigree     0.8561335099
## age          0.0136150356
```

← be removed from Logistic model

```
f2 <- glmnet(as.matrix(PimaIndiansDiabetes[,-9]),
             PimaIndiansDiabetes[,9],family="binomial")
pred.sex <-predict(f2,newx=as.matrix(PimaIndiansDiabetes[,-9]),
                   s=lambda_min,type="class")
table(pred.sex,PimaIndiansDiabetes$diabetes)
```

direct plug. method

```
##
## pred.sex neg pos
##      neg 445 115
##      pos  55 153
```

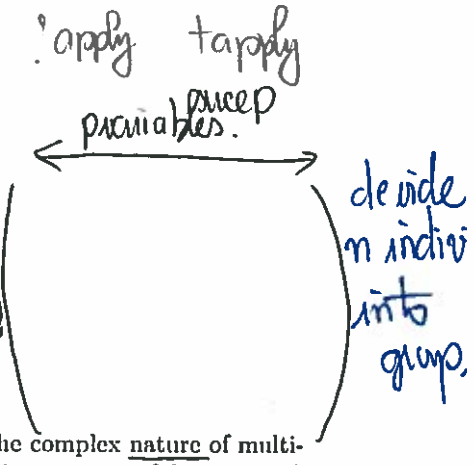
```
#The misclassification error is (55+115)/768
```

leave one out method -

```
cverror=0
for(i in 1:dim(PimaIndiansDiabetes)[1]){
  lambda=cv.glmnet(as.matrix(PimaIndiansDiabetes[-i,-9]),
                PimaIndiansDiabetes[-i,9],
                family="binomial",
                grouped=FALSE)$lambda_min
  f=glmnet(as.matrix(PimaIndiansDiabetes[-i,-9]),
           PimaIndiansDiabetes[-i,9],
           family="binomial")
  pred.diab=predict(f,newx=as.matrix(PimaIndiansDiabetes[i,-9]),
                    s=lambda_min,type="class")
  if(PimaIndiansDiabetes[i,9]!=pred.diab){cverror=cverror+1}
}
#The misclassification error
cverror/dim(PimaIndiansDiabetes)[1]
```

```
## [1] 0.2226562
```

? glmnet vs cv.glmnet



Chapter 8. Clusting

Jianxuan Liu

Fall 2018

Rudimentary, exploratory procedures are often quite helpful in understanding the complex nature of multivariate relationships. In this chapter, we will discuss some displays based on certain measures of distance and suggested step-by-step rules for grouping objects. Searching the data for a structure of "natural" groupings is an important exploratory techinique. Groupings can provide an informal means for assessing dimensionality, identifying outliers and suggesting interesting hypotheses concerning relationships.

The major goal of cluster analysis is to separate individual observations, or items, into groups, or clusters, on the basis of the values for the p variables measured on each individual. Often in clustering the items are called objects. We wish to create clusters such that the objects within each cluster are similar and objects in different clusters are dissimilar. The dissimilarity between any two objects is typically quantified using a distance measure (like Euclidean distance).

Applications of Cluster Analysis:

- In marketing, researchers attempt to find distinct clusters of the consumer population so that several distinct marketing strategies can be used for the clusters.
- In ecology, scientists classify plants or animals into various groups based on some measurable characteristics.
- Researchers in genetics may separate genes into several classes based on their expression ratios measured at different time points.

There are three major classes of clustering methods – from oldest to newest, they are:

1. **Hierarchical methods**: cluster the data in a series of n steps, typically joining observations together step by step to form clusters.
2. **Partitioning methods**: first determine k, and then typically attempt to find the partition into k clusters that optimizes some objective function of the data.
3. **Model-based clustering**: takes a statistical approach, formulating a model that categorizes the data into subpopulations and using maximum likelihood to estimate the model parameters.

To summarize, the basic objective in cluster analysis is to discover natural groupings of the items or variables. In turn, we must develop a quantitative scale on which to measure the association (similarity) between objects.

1. Similarity Measures

Most efforts to produce a rather simple group structure from a complex data set require a measure of "closeness", or "similarity". There is often a great deal of subjectivity involved in the choice of a similarity measure. Inportant consideratins inclde the nature of the variables, scales of measurement and subject matter knowledge.

When items are clustered, proximity is usually indicated by some sort of distance. By contrast, variables are usually groups on the basis of correlation coefficients or like measure of association.

Consider two p-dimensional observations (items) $\mathbf{x} = [x_1, x_2, \cdots, x_p]$ and $\mathbf{y} = [y_1, y_2, \cdots, y_p]$, then the distance of x, y is

1. Euclidean Distance

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_p-y_p)^2} = \sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}$$

2. Manhattan Distance:

$$d(\mathbf{x},\mathbf{y}) = \mid x_1-y_i \mid + \mid x_2-y_2 \mid + \cdots + \mid x_p-y_p \mid = \sum_{i=1}^{p} \mid x_i-y_i \mid$$

3. Statistical Distance

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T\mathbf{A}(\mathbf{x}-\mathbf{y})}$$

Ordinarily, $\mathbf{A} = \mathbf{S}^{-1}$ where $\mathbf{S}$ contain the sample variances and covariances. However, without prior knowledge of the distinct groups, these sample quantities cannot be computed. For this reason, Euclidean distance is often preferred for clustering.

- Minkowski Metric

$$d(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{p} \mid x_i-y_i \mid^m\right]^{1/m}$$

For $m = 1$, $d(\mathbf{x},\mathbf{y})$ measure the "city-block"distance between two points in p dimensions. For $m = 2$, $d(\mathbf{x},\mathbf{y})$ becomes the Euclidean distance. In general, varying m changes the weight given to larger and smaller differences.

- Canberra Metric

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{p} \frac{\mid x_i-y_i \mid}{(x_i+y_i)}$$

- Czekanowski Coefficient

$$d(\mathbf{x},\mathbf{y}) = 1 - \frac{2\sum_{i=1}^{p}\min(x_i,y_i)}{\sum_{i=1}^{p}(x_i+y_i)}$$

Note that Canberra Metric and Czekanowski coefficient are defined for nonnegative variables only.

Example: Characteristics matching

Characteristics Matching. If the characteristic is present when value is 1. If the characteristic is absent when value is 0.

Variables	1	2	3	4	5
Item 1	1	0	0	1	1
Item 2	1	1	0	1	0

Let x_{ij} be the score (1 or 0) of the j^{th} variable on the i^{th} item and x_{kj} be the score of the k^{th} item, $j = 1, 2, \cdots, p$. Consequently, the squared Euclidean distance $\sum_{i=1}^{p}(x_{ij}-x_{kj})^2$ provides a count of the number of mismatches. A large distance corresponds to many mismatches, that is, dissimilar items. From the preceding display, the square of the distance between item i and k would be

$$\sum_{i=1}^{p}(x_{ij}-x_{kj})^2 = (1-1)^2+(0-1)^2+(0-0)^2+(1-1)^2+(1-0)^2 = 2$$

Although a distance based on $\sum_{i=1}^{p}(x_{ij}-x_{kj})^2$ might be used to measure similarity, it suffers from weighting the $1-1$ and $0-0$ matches equally. In some cases, a $1-1$ match is a stronger indication of similarity than a

$0-0$ match. For instance, in grouping people, the evidence that two persons both read ancient Greek is stronger evidence of similarity than the absence of this ability. Thus, it might be reasonable to discount the $0-0$ matches or even disregard them completely. To allow for differential treatment of the $1-1$ matches and the $0-0$ matches, several schemes for defining similarity coefficients have been suggested. To introduce these schemes, let us arrange the frequencies of matches and mismatches for items i and k in the form of a contingency table:

		Item k		
		1	0	Totals
Item i	1	a	b	$a+b$
	0	c	d	$c+d$
Totals		$a+c$	$b+d$	$p=a+b+c+d$

In this table, a represents the frequency of $1-1$ matches, b is the frequency of $1-0$ matches, and so forth. Given the foregoing five pairs of binary outcomes, $a=2$ and $b=c=d=1$. Here is a list of common similarity coefficients defined in terms of the frequencies

- Euclidean distance $1-1$ and $0-0$ are euqal: $\frac{a+d}{p}$
- Double weight to $1-1$ and $0-0$: $\frac{2(a+d)}{2(a+d)+b+c}$
- Double weight for unmatched pairs: $\frac{a+d}{a+d+2(b+c)}$
- Only $1-1$: $\frac{a}{p}$
- Exclude $0-0$ entirely: $\frac{a}{a+b+c}$
- Ratio of matches to mismatches with 0-0 matches excluded: $\frac{a}{b+c}$

2. Hierarchical Clustering Methods

Hierarchical clustering, as is denoted by the name, involves organizing your data into a kind of hierarchy. The common approach is called an **agglomerative approach**. Agglomerative hierarchical methods start with the individual objects. Thus, there are initially as many clusters as objects. The most similar objects are first grouped, and these initial groups are merged according to their similarities. Eventually, as the similarity decreases, all subgroups are fused in to a single cluster.

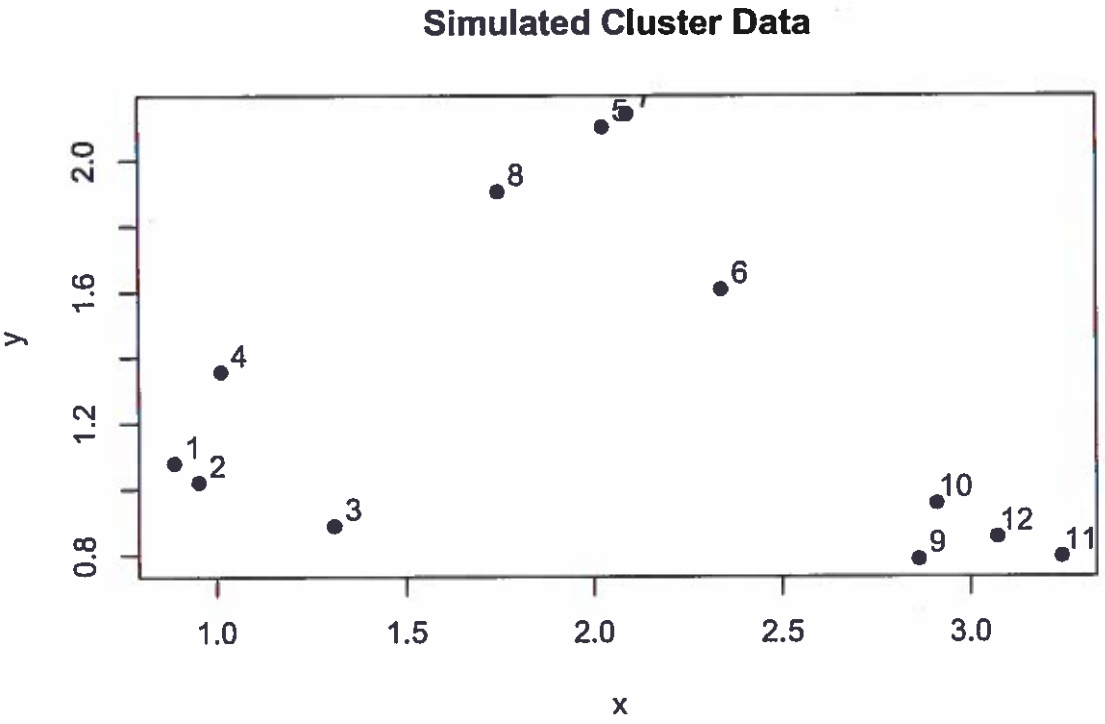
Imagine there is all these little particles floating around (your data points), and you start kind of grouping them together into little balls. And then the balls get grouped up into bigger balls, and the bigger balls get grouped together into one big massive cluster. That's the agglomerative approach to clustering. The algorithm is recursive and goes as follows:

1. Start with n clusters and on $N \times N$ distance matrix $D = \{d_{ik}\}$
2. Search for nearest/most similar pairs d_{uv} where U and V are most similar.
3. Merge clusters U and V to (UV). Update the distance matrix
4. Repeat steps 1-3 $N-1$ times

Example

Here is a simple example demonstrating how hierarchical clustering works. First we'll simulate some data in three separate clusters.

```
set.seed(123)
x = rnorm(12, rep(1:3, each = 4), 0.2)
y = rnorm(12, rep(c(1, 2, 1), each = 4), 0.2)
plot(x, y, col = 2, pch = 19, cex = 1,main='Simulated Cluster Data')
text(x + 0.05, y + 0.05, labels = as.character(1:12))
```



The first step in the basic clustering approach is to calculate the distance between every point with every other point. The result is a distance matrix, which can be computed with the dist() function in R.Here is just a piece of the distance matrix associated with the figure above.

```
df=data.frame(x=x, y=y)
round(dist(df),4)
```

```
##         1      2      3      4      5      6      7      8      9     10
## 2  0.0879
## 3  0.4650 0.3818
## 4  0.3046 0.3406 0.5551
## 5  1.5278 1.5198 1.4056 1.2548
## 6  1.5474 1.5070 1.2565 1.3521 0.5861
## 7  1.6044 1.5955 1.4749 1.3324 0.0778 0.5896
## 8  1.1913 1.1871 1.1059 0.9151 0.3398 0.6667 0.4175
## 9  1.9964 1.9232 1.5543 1.9347 1.5571 0.9710 1.5577 1.5801
## 10 2.0267 1.9580 1.6006 1.9387 1.4457 0.8633 1.4394 1.5018 0.1767
## 11 2.3741 2.3021 1.9354 2.3006 1.7856 1.2134 1.7717 1.8647 0.3823 0.3710
## 12 2.1957 2.1246 1.7606 2.1185 1.6264 1.0476 1.6168 1.6913 0.2200 0.1908
##        11
```

```
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12 0.1828
```

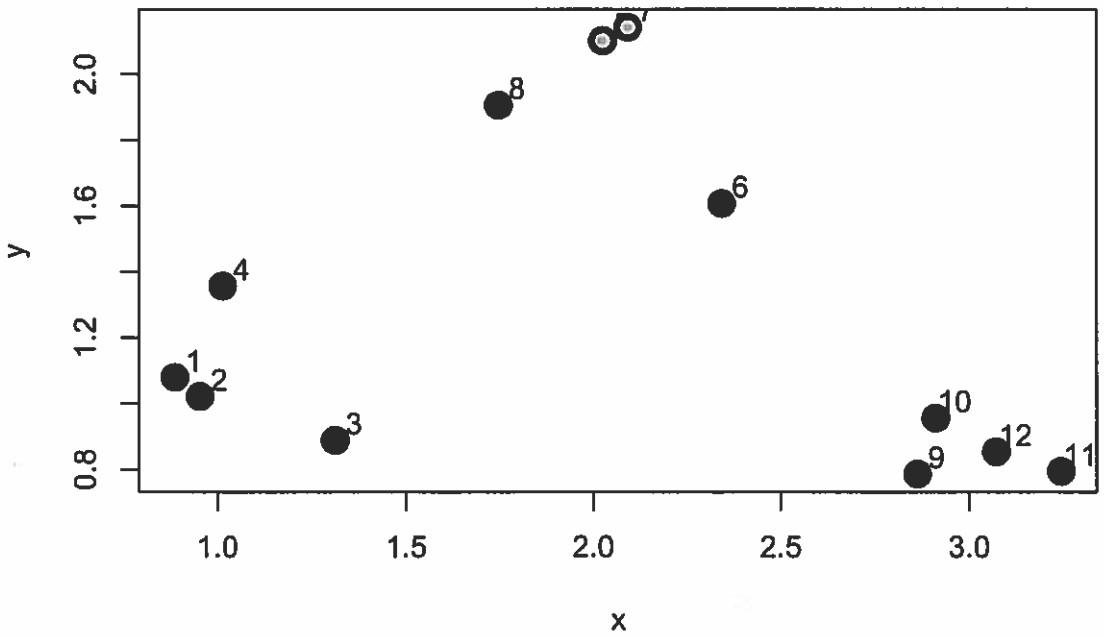
The default distance metric used by the 'dist()' function is Euclidean distance. First an agglomerative clustering approach attempts to find the two points that are closest together. In other words, we want to find the smallest non-zero entry in the distance matrix.

```
rdistxy = as.matrix(dist(df))
## Remove the diagonal from consideration
diag(rdistxy)=diag(rdistxy) + 100000
# Find the index of the points with minimum distance
ind=which(rdistxy == min(rdistxy), arr.ind = TRUE)
ind
```

```
##   row col
## 7   7   5
## 5   5   7
```

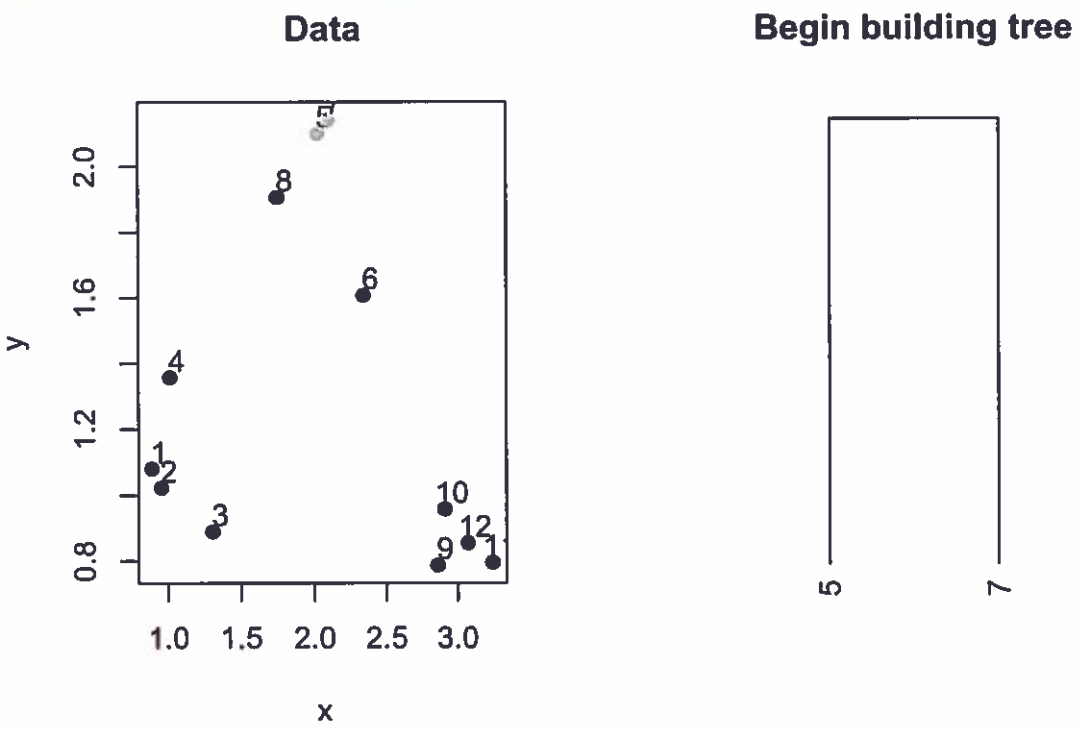
Now we can plot the points and show which two points are closest together according to our distance metric.

```
plot(x, y, col = "blue", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
points(x[ind[1, ]], y[ind[1, ]], col = "orange", pch = 19, cex = 1)
```



The next step for the algorithm is to start drawing the tree, the first step of which would be to "merge" these two points together.

```
par(mfrow = c(1, 2))
plot(x, y, col = "blue", pch = 19, cex = 1, main = "Data")
text(x + 0.05, y + 0.05, labels = as.character(1:12))
points(x[ind[1, ]], y[ind[1, ]], col = "orange", pch = 19, cex = 1)
# Make a cluster and cut it at the right height
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

hcluster=dist(df) %>% hclust
dendro=as.dendrogram(hcluster)
cutDendro=cut(dendro, h = (hcluster$height[1] + 0.00001))
plot(cutDendro$lower[[11]], yaxt = "n", main = "Begin building tree")
```



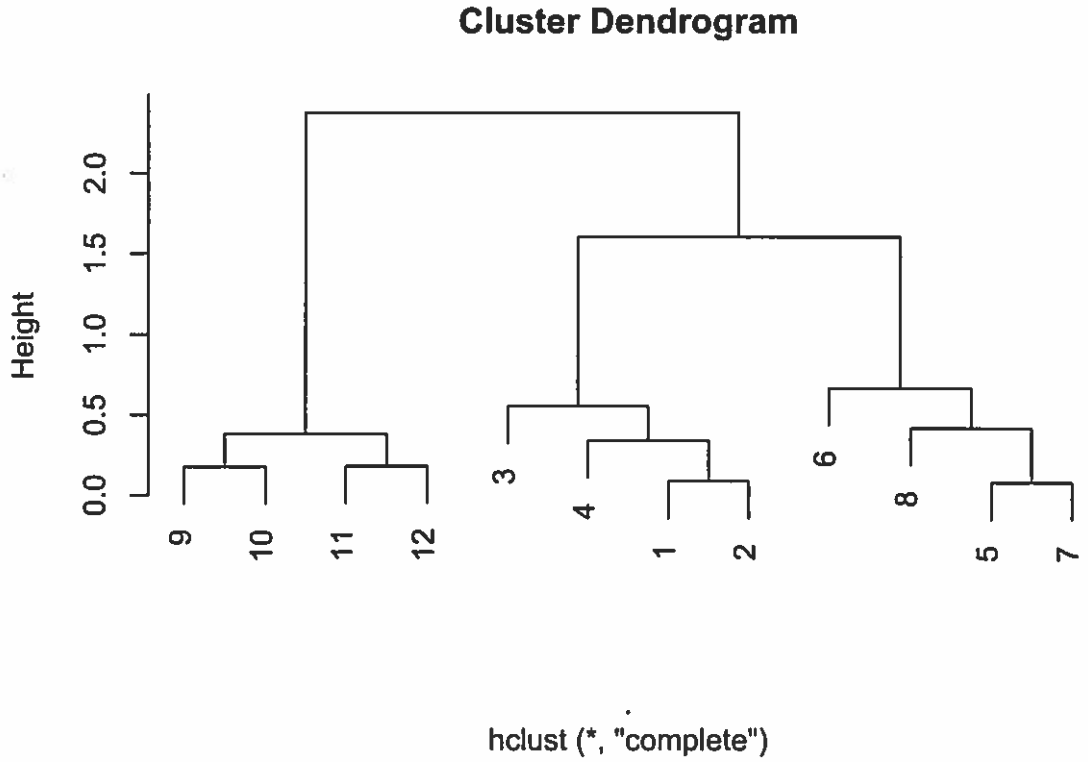
Now that we've merged the first two "leaves" of this tree, we can turn the algorithm crank and continue to build the tree. Now, the two points we identified in the previous iteration will get "merged" into a single point. We need to search the distance matrix for the next two closest points, ignoring the first two that we already merged.

```
nextmin=rdistxy[order(rdistxy)][3]
ind= which(rdistxy == nextmin,arr.ind=TRUE)
ind
```

```
##   row col
## 2   2   1
## 1   1   2
```

We continue with the next pair of points and the merged tree leaves. And on and on in this manner. If we were to continue in this fashion - identifying the two closest points and merging them, we'd end up with a dendrogram that looks like this one. Here, we call the hclust() do run the clustering algorithm.

```
hClustering =data.frame(x=x,y=y) %>% dist %>% hclust
plot(hClustering)
```



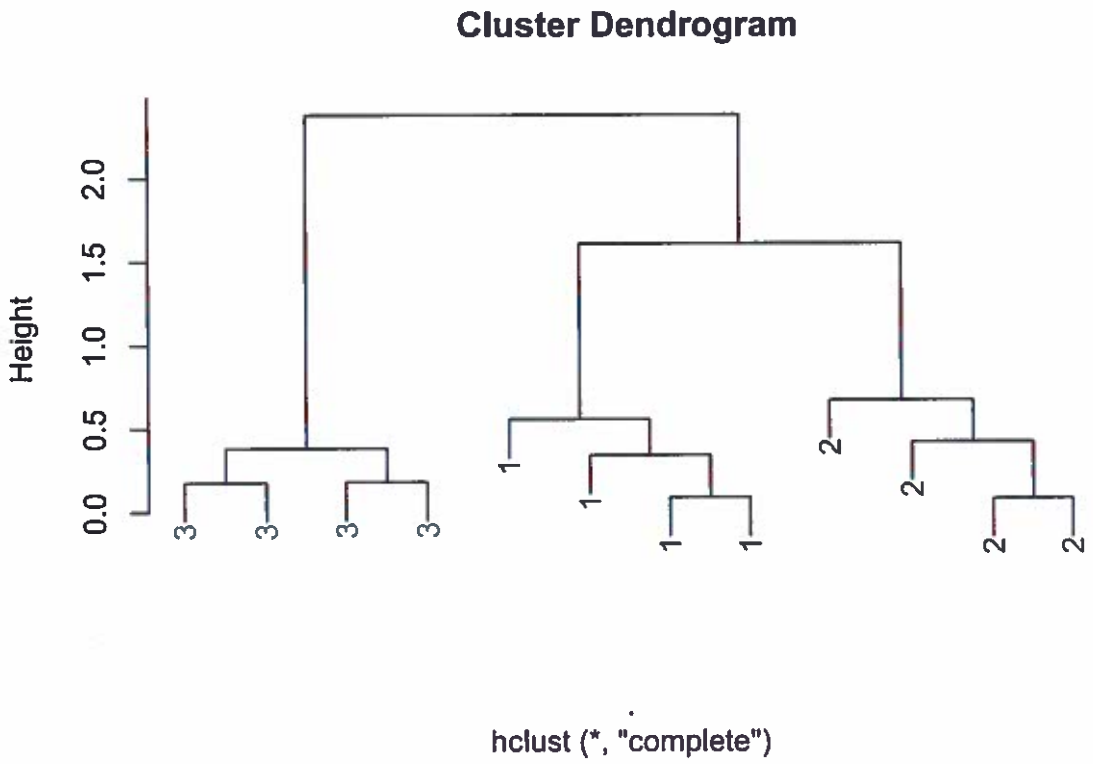
From the tree/dendrogram it's clear that there are three clusters each with four points.

Refined Dendrograms

It is desirable to make prettier dendrograms with color coded each of the cluster members by their cluster membership.

```
myplclust <- function(hclust, lab = hclust$labels,
                      lab.col = rep(1,length(hclust$labels)),
                      hang = 0.1, ...)
{
  y=rep(hclust$height, 2)
  x=as.numeric(hclust$merge)
  y=y[which(x < 0)]
  x=x[which(x < 0)]
  x=abs(x)
  y=y[order(x)]
  x=x[order(x)]
```

```
  plot(hclust, labels = FALSE, hang = hang, ...)
  text(x = x, y = y[hclust$order] - (max(hclust$height) * hang),
       labels = lab[hclust$order], col = lab.col[hclust$order],
       srt = 90, adj = c(1, 0.5), xpd = NA, ...) }

hClustering=data.frame(x = x, y = y) %>% dist %>% hclust
myplclust(hClustering, lab = rep(1:3, each = 4), lab.col = rep(1:3, each = 4))
```



3. Linkage Methods in Hierarchical Clustering

1. Single linkage clustering:

Single linkage clustering is sometimes called "nearest neighbor" clustering. At each step, joins the clusters whose minimum distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$d_{AB} = \min_{i \in A, j \in B}(d_{ij})$$

2. Complete linkage algorithm:

Complete linkage clustering is sometimes called "farthest neighbor" clustering. At each step, joins the clusters whose maximum distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$d_{AB} = \max_{i \in A, j \in B}(d_{ij})$$

3. Average linkage algorithm:

At each step, joins the clusters whose average distance between objects is smallest, i.e., joins the clusters A and B with the smallest

$$d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$$

where n_A and n_B are the number of objects in clusters A and B, respectively

Example

To illustrate the linkage algorithms, we consider the hypothetical distances between pairs of five objects as follows:

$$D = \{d_{ij}\} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & \mathbf{2} & 8 & 0 \end{pmatrix} \end{array}$$

Single linkage clustering:

Treating each object as a cluster, we commence clustering by merging the two closest items. Since $\min_{i,k} = d_{EC} = 2$, object E and C are merged to form the cluster (CE). To implement the next level of clustering, we need the distance between the cluster (CE) an the remaining objects A, B and D. The nearest neighbor distances are

$$d_{(CE)A} = \min\{d_{CA}, d_{EA}\} = \min\{3, 11\} = 3$$
$$d_{(CE)B} = \min\{d_{CB}, d_{EB}\} = \min\{7, 10\} = 7$$
$$d_{(CE)D} = \min\{d_{CD}, d_{ED}\} = \min\{9, 8\} = 8$$

Deleting the rows and columns of D corresponding to objects C and E, and addding a row and column for the cluster (CE), we obtain the new distance matrix

$$\begin{array}{c} \\ CE \\ A \\ B \\ D \end{array} \begin{array}{c} \begin{array}{cccc} CE & A & B & D \end{array} \\ \begin{pmatrix} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{pmatrix} \end{array}$$

The smallest distance between pairs of clusters in row $d_{(CE)A} = 3$, and we merge cluster A with cluster CE to get the next cluster, (ACE). Calculating

$$d_{(ACE)B} = \min\{d_{(CE)B}, d_{AB}\} = \min\{7, 9\} = 7$$
$$d_{(ACE)D} = \min\{d_{(CE)D}, d_{AD}\} = \min\{8, 6\} = 6$$

We find that the distance matrix for the next level of clustering is

$$\begin{array}{c} \\ A \\ B \\ D \end{array} \begin{array}{c} \begin{array}{ccc} ACE & B & D \end{array} \\ \begin{pmatrix} 0 & & \\ 7 & 0 & \\ 6 & 5 & 0 \end{pmatrix} \end{array}$$

The minimum nearest neighbor distance between pairs of clusters is $d_{DB} = 5$, and we merge objects D and B to get the cluster (BD). At this point we have two distinct clusters (ACE) and (BD). Their nearest neighbor distance is

$$d_{(ACE)(BD)} = \min\{d_{(ACE)B} d_{(ACE)D}\} = \min\{7, 6\} = 6$$

The final distance matrix becomes

$$\begin{array}{c} \\ ACE \\ BD \end{array} \begin{array}{c} \begin{array}{cc} ACE & BD \end{array} \\ \begin{pmatrix} 0 & \\ 6 & 0 \end{pmatrix} \end{array}$$

Complete linkage clustering:

At the first stage, object C and E are merged, since they are most similar. This gives the cluster (CE). At stage 2, we compute

$$\begin{aligned} d_{(CE)A} &= \max\{d_{CA}, d_{EA}\} = \max\{3, 11\} = 11 \\ d_{(CE)B} &= \max\{d_{CB}, d_{EB}\} = \max\{7, 10\} = 10 \\ d_{(CE)D} &= \max\{d_{CD}, d_{ED}\} = \max\{9, 8\} = 9 \end{aligned}$$

and the modified distance matrix becomes

$$\begin{array}{c} \\ CE \\ A \\ B \\ D \end{array} \begin{array}{c} \begin{array}{cccc} CE & A & B & D \end{array} \\ \begin{pmatrix} 0 & & & \\ 11 & 0 & & \\ 10 & 9 & 0 & \\ 9 & 6 & 5 & 0 \end{pmatrix} \end{array}$$

The next merger occurs between the most similar groups, B and D, to give the cluster (BD). At stage 3, we have

$$\begin{aligned} d_{(BD)(CE)} &= \max\{d_{B(CE)} d_{D(CE)}\} = \max\{10, 9\} = 10 \\ d_{(BD)A} &= \max\{d_{BA}, d_{DA}\} = \max\{9, 6\} = 6 \end{aligned}$$

and the distance matrix

$$\begin{array}{c} \\ CE \\ BD \\ A \end{array} \begin{array}{c} \begin{array}{ccc} CE & BD & A \end{array} \\ \begin{pmatrix} 0 & & \\ 10 & 0 & \\ 11 & 9 & 0 \end{pmatrix} \end{array}$$

The next merger produces the cluster (ABD). At the final stage, the groups (CE) and (ABD) are merges as the single cluster (ABCDE) at level

$$d_{(ABD)(CE)} = \max\{d_{A(CE)}, d_{(BD)(CE)}\} = \max\{11, 10\} = 11$$

The final distance matrix becomes

$$\begin{array}{c} \\ CE \\ ABD \end{array} \begin{array}{c} \begin{array}{cc} CE & ABD \end{array} \\ \begin{pmatrix} 0 & \\ 11 & 0 \end{pmatrix} \end{array}$$

It's important to consider standardizing observations.

Standardization of Observations ← explore this.

If the variables in the data set are of different types or are measured on very different scales, then some variables may play an inappropriately dominant role in the clustering process. In this case, it is recommended to standardize the variables in some way before clustering the objects. Possible standardization approaches:

- Divide each column by its sample standard deviation, so that all variables have standard deviation 1.
- Divide each variable by its sample range (max − min); Milligan and Cooper (1988) found that this approach best preserved the clustering structure.
- Convert data to z-scores by (for each variable) subtracting the sample mean and then dividing by the sample standard deviation – a common option in clustering software packages.

Example: Food data:

```
food=read.table("food.txt",head=T)  #food.txt
dim(food);head(food)
```

```
## [1] 27  6
```

```
##    Food Energy Protein Fat Calcium Iron
## 1    BB    340      20  28       9  2.6
## 2    HR    245      21  17       9  2.7
## 3    BR    420      15  39       7  2.0
## 4    BS    375      19  32       9  2.5
## 5    BC    180      22  10      17  3.7
## 6    CB    115      20   3       8  1.4
```

```
attach(food)
```

Let's first scale the data by dividing each variable by its standard deviation:

```
std=apply(food[,-1], 2, sd) # finding standard deviations of variables
food.std=sweep(food[,-1],2,std,FUN="/")
```

? see the head(food) after standardizing.

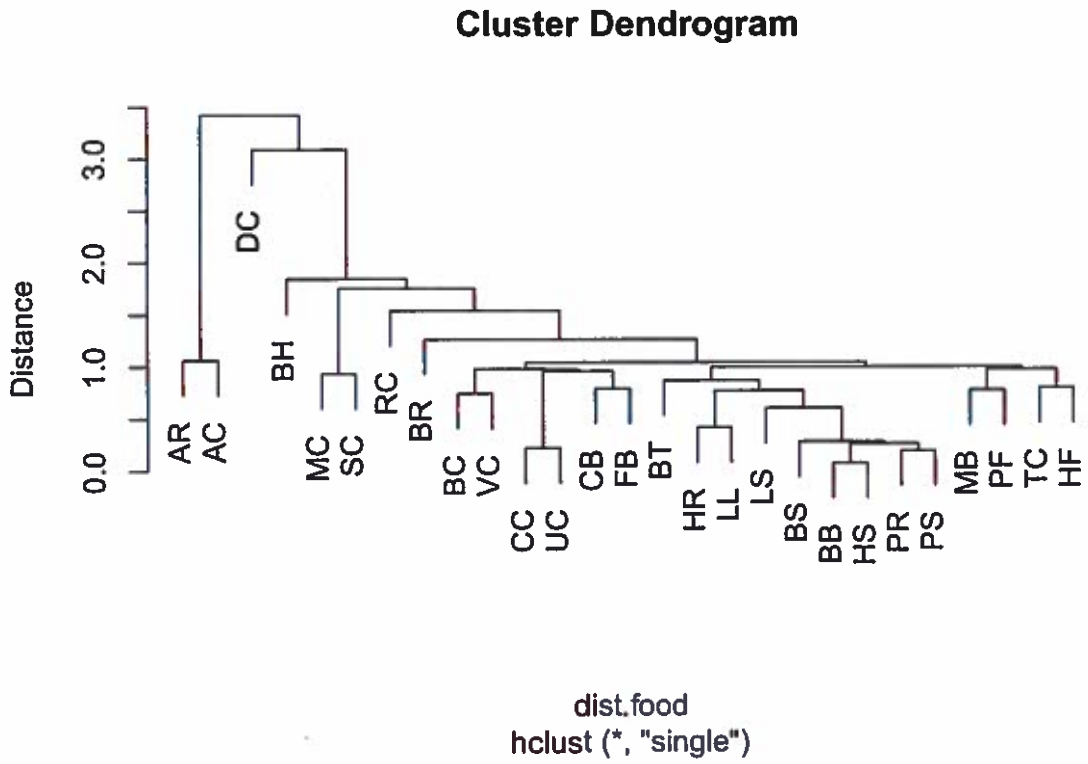
Calculating pairwise Euclidean distances between the (standardized) objects:

```
dist.food=dist(food.std)
```

! distinguish apply & sweep

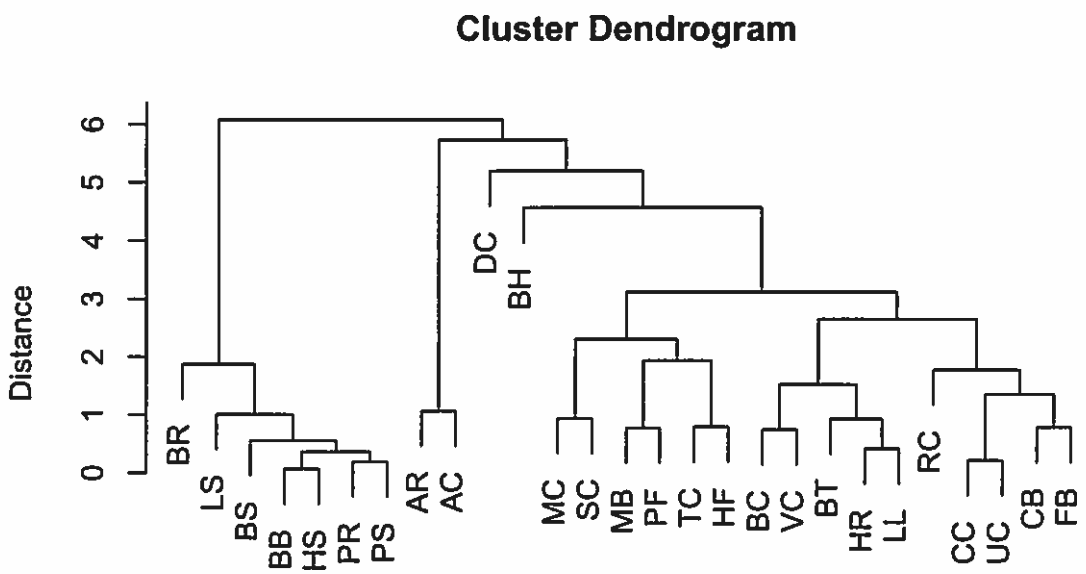
Single linkage:

```
food.single.link =hclust(dist.food, method='single') ←
# Plotting the single linkage dendrogram:
plot(food.single.link, labels=Food, ylab="Distance")
```



Complete linkage:

```
food.complete.link = hclust(dist.food, method='complete')
# Plotting the complete linkage dendrogram:
plot(food.complete.link, labels=Food, ylab="Distance")
```



Average linkage:

```
food.avg.link =hclust(dist.food, method='average')
# Plotting the average linkage dendrogram:
plot(food.avg.link, labels=Food, ylab="Distance")
```

? with diabetes data, which algorithm is the best one ?

? cut the cluster dendrogram.

then the function

cut ← cutree(food.avg.link, k = ...) (the # of cluster that we want to have) will return the result.

Pros and Cons of Hierarchical Clustering

- An advantage of hierarchical clustering methods is their computational speed for small data sets.
- Another advantage is that the dendrogram gives a picture of the clustering solution for a variety of choices of k (the number of clusters) at once.
- On the other hand, a major disadvantage is that once two clusters have been joined, they can never be split apart later in the algorithm, even if such a move would improve the clustering.
- The so-called partitioning methods of cluster analysis do not have this restriction.
- In addition, hierarchical methods can be less efficient than partitioning methods for large data sets, when n is much greater than k

2.7 Partioning methods

Partitioning methods fix the number of clusters k and seek the best possible partition for that k. The goal is to choose the partition which gives the optimal value for some clustering criterion, or objective function. In reality, we cannot search all possible partitions to try to optimize the clustering criterion, but the algorithms are designed to search intelligently among the partitions. For a fixed k, partitioning methods are able to investigate far more possible partitions than a hierarchical method is. In practice, it is recommended to run a partitioning method for several choices of k and examine the resulting clusterings.

4. K-means Cluster (we known the # of clusters that we want to use.

The goal of K-means, the most well-known partitioning method, is to find the partition of n objects into k clusters that minimizes a within-cluster sum of squares criterion. In the traditional K-means approach, "closeness" to the cluster centers is defined in terms of squared Euclidean distance, defined by:

$$d_E^2(\mathbf{x}, \bar{\mathbf{x}}) = (\mathbf{x} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) = \sum_m (x_{im} - \bar{x}_{cm})^2$$

where $\mathbf{x} = (x_1, \cdots, x_q)^T$ is any particular observation and $\bar{\mathbf{x}}$ is the centroid (multivariate mean vector) for cluster c.

The goal is to minimize the sum (over all objects within all clusters) of these squared Euclidean distances:

$$WSS = \sum_{c=1}^{k} \sum_{i \in c} d_E^2(\mathbf{x}_i, \bar{\mathbf{x}}_c)$$

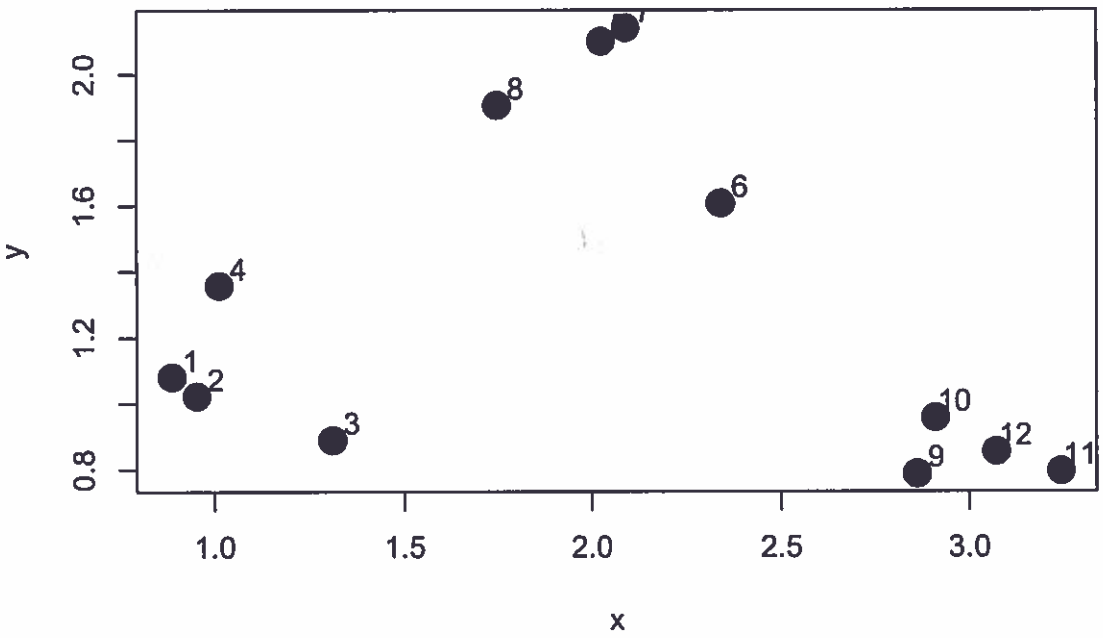
In practice, K-means will not generally achieve the global minimum of this criterion over the whole space of partitions. In fact, only under certain conditions will it achieve the local minimum (Selim and Ismail, 1984)

The K-means algorithm (MacQueen, 1967) begins by randomly allocating the n objects into k clusters (or randomly specifying k centroids). One at a time, the algorithm moves each object to the cluster whose centroid is closest to it, using the measure of closeness $d_E^2(\mathbf{x}, \bar{\mathbf{x}})$. When an object is moved, the centroids are immediately recalculated for the cluster gaining the object and the cluster losing it. The method repeatedly cycles though the objects until no reassignments of objects take place. The final clustering result will somewhat depend on the initial configuration of the objects. In practice, it is good to rerun the algorithm a few times (with different starting points) to make sure the result is stable. The R function kmeans() performs K-means clustering.

Example

We simulate some data from three clusters and plot the dataset below.

```
set.seed(123)
x=rnorm(12, mean = rep(1:3, each = 4), sd = 0.2)
y=rnorm(12, mean = rep(c(1, 2, 1), each = 4), sd = 0.2)
plot(x, y, col = "blue", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
```



Note: The 'kmeans()' function in R implements the K-means algorithm and can be found in the 'stats' package, which comes with R and is usually already loaded when you start R. Two key parameters that you have to specify are x, which is a matrix or data frame of data, and centers which is either an integer indicating the number of clusters or a matrix indicating the locations of the initial cluster centroids. The data should be organized so that each row is an observation and each column is a variable or feature of that observation.

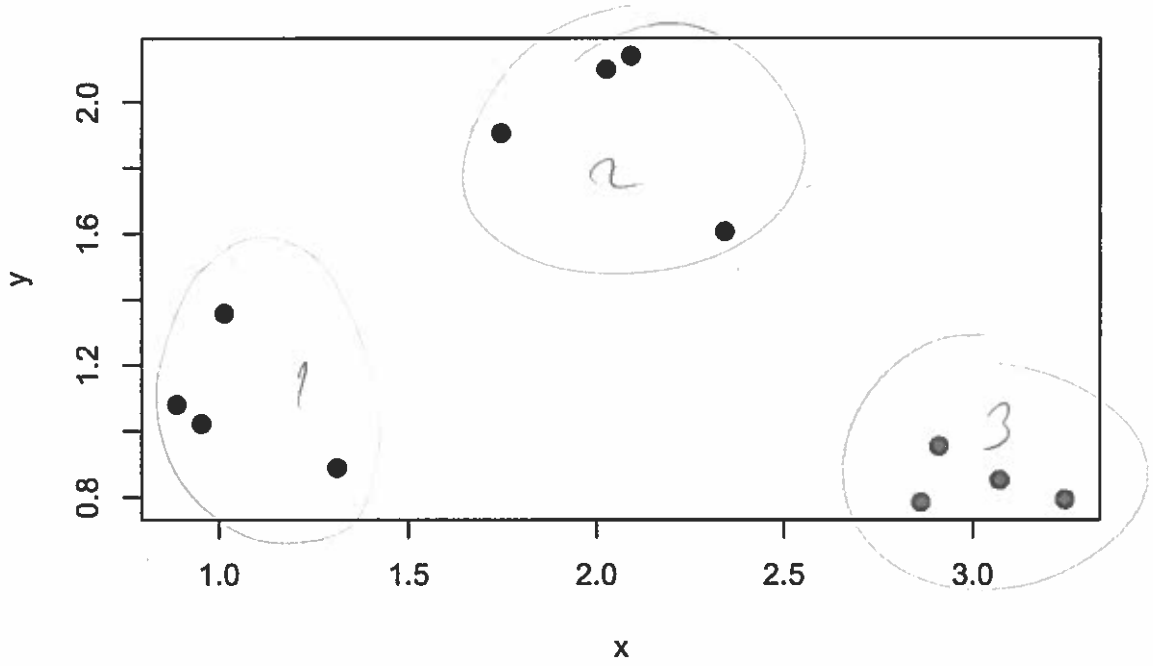
```
df= data.frame(x, y)
km =kmeans(df, centers = 3)
names(km)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
km$cluster
```

```
##  [1] 3 3 3 3 1 1 1 1 2 2 2 2
```

```
plot(df, col =(km$cluster +1) , main="K-Means result with 3 clusters", pch=20, cex=2)
```



Example: revisit food data

Consider the food.std data frame given above. We first consider a K-means cluster with $k = 5$

```
food.k5 <- kmeans(food.std, centers=5, iter.max=100, nstart=25)
food.k5
```

kmeans with food.std.

≠ last page kmeans with food data.

```
## K-means clustering with 5 clusters of sizes 8, 8, 2, 8, 1
##
## Cluster means:
##     Energy  Protein        Fat   Calcium      Iron
## 1 3.377951 4.410004 2.56506304 0.1121302 1.6599848
## 2 1.414170 4.116004 0.57741679 0.6743833 0.6930864
## 3 0.568138 2.116802 0.08883335 0.9995610 3.9018198
## 4 1.759993 5.380205 0.77729183 0.2771219 1.9509099
## 5 1.778519 5.174405 0.79950017 4.7030629 1.7113245
##
## Clustering vector:
##  [1] 1 4 1 1 4 2 4 4 1 1 1 1 1 4 4 2 3 3 2 2 2 2 2 2 5 4 4
##
## Within cluster sum of squares by cluster:
## [1]  4.3254549 10.2035285  0.5626614 13.0477424  0.0000000
##  (between_SS / total_SS =  78.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
# try k=4:
food.k4 <- kmeans(food.std, centers=4, iter.max=100, nstart=25)
food.k4

## K-means clustering with 4 clusters of sizes 14, 2, 3, 8
##
## Cluster means:
##     Energy  Protein        Fat   Calcium     Iron
## 1 1.619723 4.872005 0.68528586 0.2544670 1.388618
## 2 0.568138 2.116802 0.08883335 0.9995610 3.901820
## 3 1.498567 4.312004 0.68105570 2.9175222 1.140883
## 4 3.377951 4.410004 2.56506304 0.1121302 1.659985
##
## Clustering vector:
##  [1] 4 1 4 4 1 1 1 1 4 4 4 4 4 1 1 1 2 2 1 1 1 3 1 3 3 1 1
##
## Within cluster sum of squares by cluster:
## [1] 28.9804747  0.5626614  6.9589520  4.3254549
##  (between_SS / total_SS =  68.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

5. Determining the Optimal Clusters

At some point we need to choose a single value of k to get a clustering solution. A variety of criteria have been proposed to pick the best value of k.

1. Average silhouette width:

- The average silhouette width is based on the difference between the average dissimilarity of objects to other objects in their own cluster and the average dissimilarity of objects to the objects in a "neighbor cluster."
- The larger the average silhouette width, the better the clustering of the objects.
- We could calculate the average silhouette width for clusterings based on several values of k and choose the k with the largest average silhouette width.
- The silhouette function in the cluster package of R gives the average silhouette width for any clustering result and distance matrix

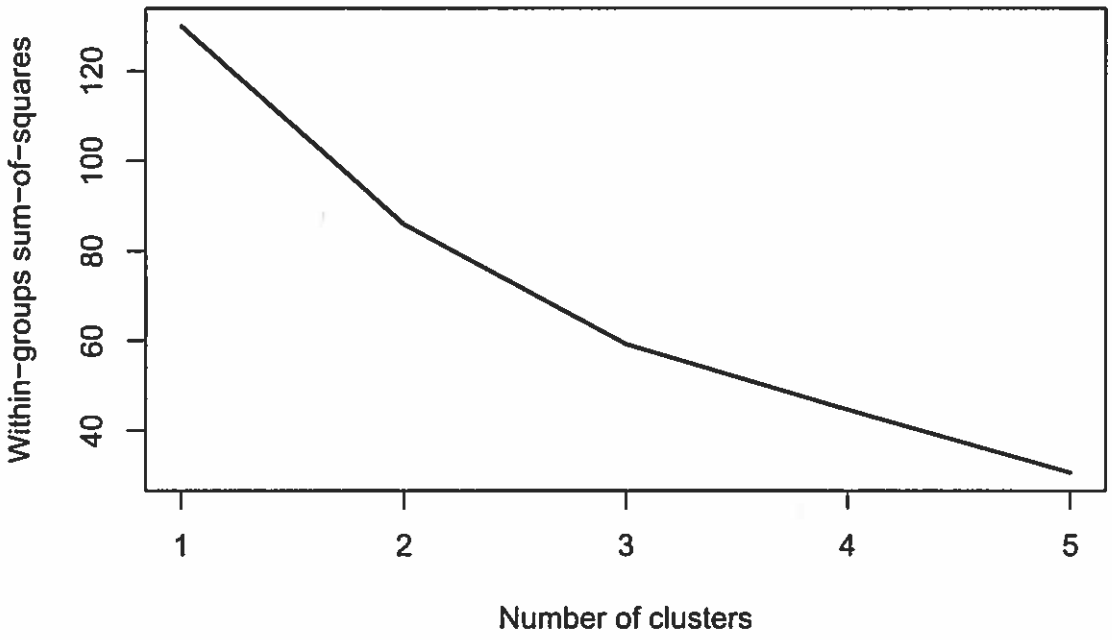
2. Dunn index and the Davies-Bouldin Index: These are implemented with the clv.Dunn and clv.Davies.Bouldin functions in the clv package.

3. Within cluster sum-of-squares (WSS):

- A common way to choose k in K-means clustering
- As k increases, the corresponding WSS will decrease, and at some point will level off.
- The "best" choice of k usually occurs near the "elbow" in this plot.

In the food.std data, we use the WSS method to determine the optimal number of k clusters.

```
my.data.matrix = food.std
my.k.choices = 2:5
n = length(my.data.matrix[,1])
wss1 = (n-1)*sum(apply(my.data.matrix,2,var))
wss = numeric(0)
for(i in my.k.choices) {
W = sum(kmeans(my.data.matrix,i)$withinss)
wss = c(wss,W)
}
wss = c(wss1,wss)
plot(c(1,my.k.choices),wss,type='l',xlab='Number of clusters',
ylab='Within-groups sum-of-squares', lwd=2)
```



From the plot, we see that the optimal k is 3. We then recluster the data with $k = 3$.

```
food.k3= kmeans(food.std, centers=3, iter.max=100, nstart=25)
food.k3
```

```
## K-means clustering with 3 clusters of sizes 16, 2, 9
##
## Cluster means:
##     Energy  Protein        Fat   Calcium     Iron
## 1 1.546941 4.762805 0.63293763 0.7624857 1.313442
## 2 0.568138 2.116802 0.08883335 0.9995610 3.901820
## 3 3.271597 4.468804 2.44785237 0.1124862 1.680901
##
## Clustering vector:
##  [1] 3 3 3 3 1 1 1 1 3 3 3 3 3 1 1 1 2 2 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 52.2876687  0.5626614  6.4094695
##  (between_SS / total_SS =  54.4 %)
##
```

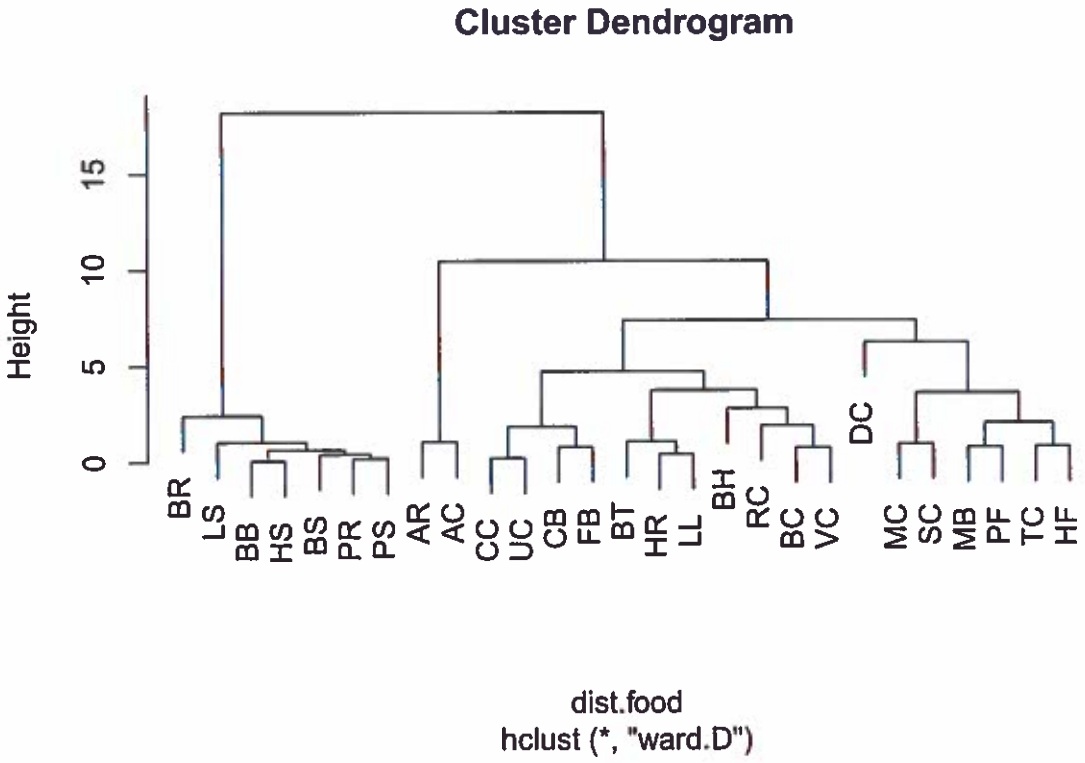
```
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

Ward's Method (hierarchical clustering + K-means)

The method of Ward (1963) is a hybrid of hierarchical clustering and K-means.

- It begins with n clusters and joins clusters together, one step at a time.
- At each step, the method searches over all possible ways to join a pair of clusters so that the K-means criterion WSS is minimized for that step.
- It begins with each object as its own cluster (so that WSS = 0) and concludes with all objects in one cluster.
- The R function hclust performs Ward's method if the option method = 'ward' is specified

```
food.ward =hclust(dist.food, method='ward.D')
plot(food.ward, labels=Food)
```



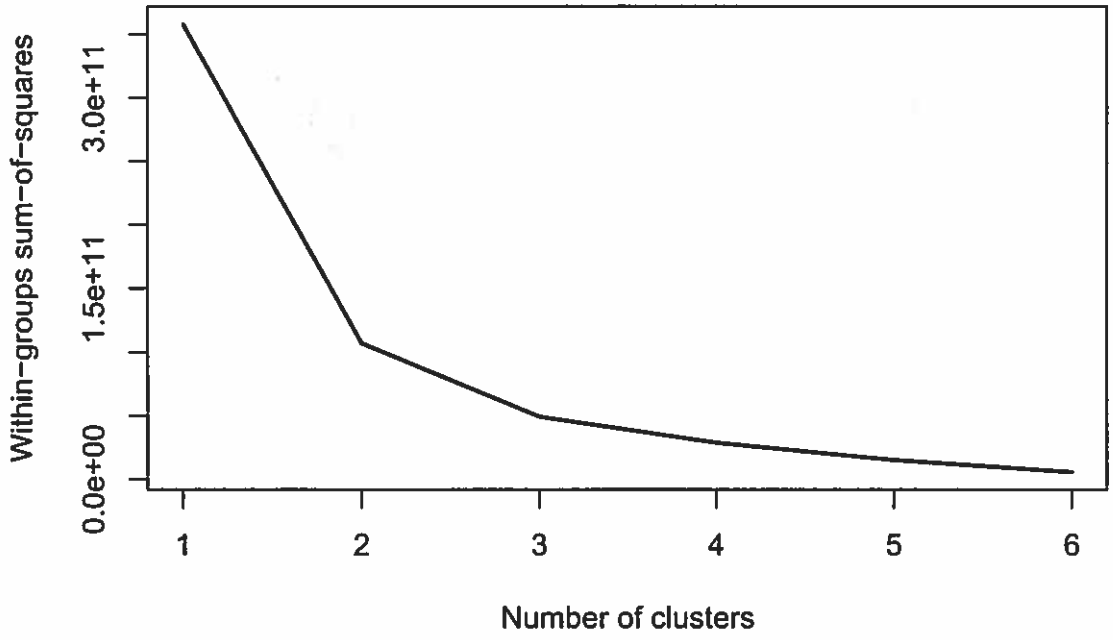
Example: States data

```
#####   NON-HIERARCHICAL (K MEANS) CLUSTERING   #####
library(maps)
library(RColorBrewer) #Creates nice looking color palettes especially for thematic maps
```

```
head(state.x77)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20
## Alaska            365   6315        1.5    69.31   11.3    66.7   152
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65
## California      21198   5114        1.1    71.71   10.3    62.6    20
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166
##               Area
## Alabama      50708
## Alaska      566432
## Arizona     113417
## Arkansas     51945
## California  156361
## Colorado    103766
```

```
my.k.choices = 2:6
n = length(state.x77[,1])
wss1 = (n-1)*sum(apply(state.x77,2,var))
wss = numeric(0)
for(i in my.k.choices) {
  W = sum(kmeans(state.x77,i)$withinss)
  wss = c(wss,W)
}
wss = c(wss1,wss)
plot(c(1,my.k.choices),wss,type='l',xlab='Number of clusters',
     ylab='Within-groups sum-of-squares', lwd=2)
```



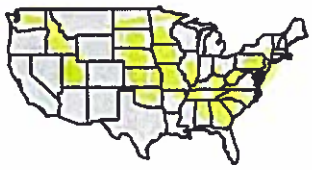

```
kmlist <- vector("list", 5)
for(k in 2:6){
  set.seed(1)
```

similarity.

```
  kmlist[[k-1]] <- kmeans(state.x77, k, nstart=5000)
}

par(mfrow=c(2,2))
for(k in 3:6){
  map(database = "state")
  title(paste0("K=",k," Clusters: State Data"))
  cols=brewer.pal(k, "Paired")
  for(j in 1:k){
    ix=names(which(kmlist[[k-1]]$cluster==j))
    if(length(ix) > 1) map(database = "state", regions = ix, col = cols[j], fill=T, add=TRUE)
  }
}
```

good by wsss.

K=3 Clusters: State Data



K=4 Clusters: State Data

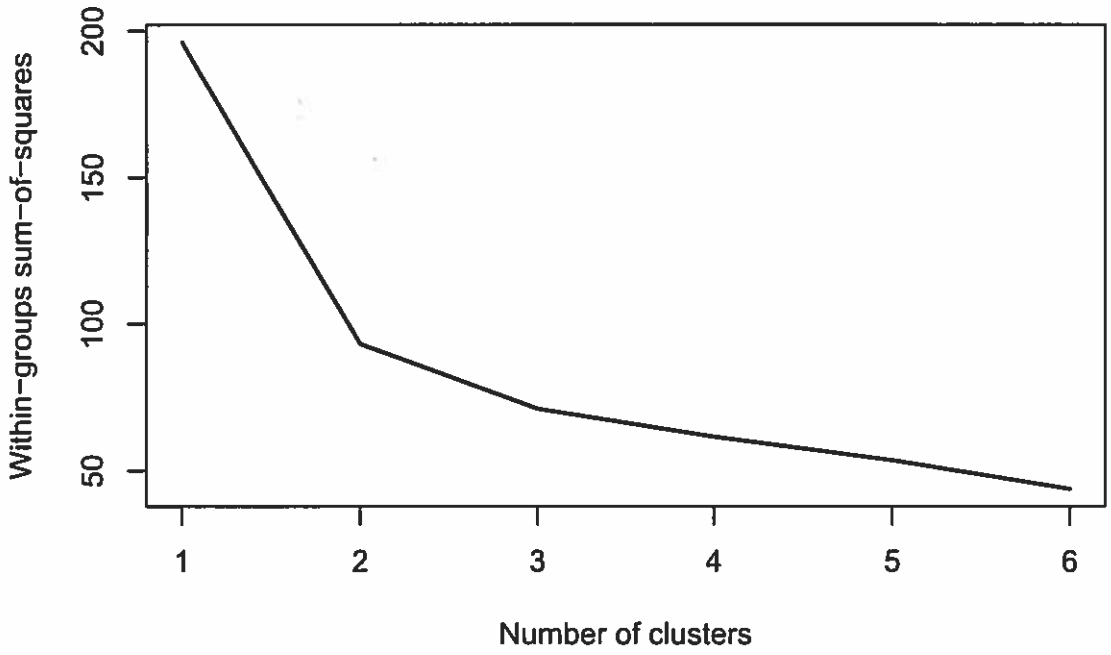
K=5 Clusters: State Data

K=6 Clusters: State Data

Look at positive factors.

Look at variables "Income","Illiteracy","Life Exp","HS Grad" only. We also standardized the data.

```
vars <- c("Income","Illiteracy","Life Exp","HS Grad")
state.x77sub=scale(state.x77[,vars])
my.k.choices = 2:6
n = length(state.x77sub[,1])
wss1 = (n-1)*sum(apply(state.x77sub,2,var))
wss = numeric(0)
for(i in my.k.choices) {
  W = sum(kmeans(state.x77sub,i)$withinss)
  wss = c(wss,W)
}
wss = c(wss1,wss)
plot(c(1,my.k.choices),wss,type='l',xlab='Number of clusters',
     ylab='Within-groups sum-of-squares', lwd=2)
```




```
kmlist <- vector("list", 5)
for(k in 2:6){
  set.seed(1)
  kmlist[[k-1]] <- kmeans(state.x77sub, k, nstart=5000)
}

par(mfrow=c(2,2))
for(k in 3:6){
  map(database = "state")
  title(paste0("K= ",k," Clusters: State standardized Data"))
  cols=brewer.pal(k, "Paired")
  for(j in 1:k){
    ix=names(which(kmlist[[k-1]]$cluster==j))
    if(length(ix) > 1) map(database = "state", regions = ix, col = cols[j], fill=T, add=TRUE)
  }
}
```

K= 3 Clusters: State standardized Data

K= 4 Clusters: State standardized Data

K= 5 Clusters: State standardized Data

K= 6 Clusters: State standardized Data

6. K-medoids Clustering

The K-medoids algorithm (Kaufman and Rousseeuw, 1987) is a robust alternative K-means.

- It attempts to minimize the criterion

$$\text{Crit}_{\text{Med}} = \sum_{c=1}^{k} \sum_{i \in c} d(\mathbf{x}_i, \mathbf{m}_c)$$

  where $\mathbf{m}_c$ is a medoid, or "most representative object" for cluster c.

- The algorithm begins (in the "build step") by selecting k such representative objects.
- It proceeds by assigning each object to the cluster with the closest medoid.
- Then (in the "swap step"), if swapping any non-medoid object with a medoid results in a decrease in the criterion Crit_{Med}, the swap is made.
- The algorithm stops when no swap can decrease Crit_{Med}.

Like K-means, the K-medoids algorithm does not globally minimize its criterion in general. The R function pam() in the cluster package performs K-medoids clustering. An advantage of K-medoids is that (unlike kmeans) the function can accept a dissimilarity matrix, as well as a raw data matrix. This is because the criterion to be minimized is a direct sum of pairwise dissimilarities between objects. The pam() function also produces tools called the silhouette plot and average silhouette width to guide the choice of k.

Example: cars data.

Let's cluster the cars into k groups using the K-medoids approach. Here we will focus on the variables that are continuous in nature rather than discrete:

```
# Loading the "cluster" package:
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.4.4
##
## Attaching package: 'cluster'
## The following object is masked from 'package:maps':
##
##     votes.repub
# Continuous variables
cars= mtcars[,c(1,3,4,5,6,7)]
# Standardizing by dividing through by the sample range of each variable
samp.range=function(x){
  myrange = diff(range(x))
  return(myrange)
}
my.ranges = apply(cars,2,samp.range)
cars.std = sweep(cars,2,my.ranges,FUN="/")
# K-medoids directly on the (standardized) data matrix:
cars.kmed.3=pam(cars.std, k=3, diss=F)
# Or you can do K-medoids by inputting the distance matrix:
#dist.cars=dist(cars,method = "euclidean", diag = FALSE)
#cars.kmed.3=pam(dist.cars, k=3, diss=T)
cars.kmed.3$clustering  # printing the "clustering vector"

##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##                   1                   1                   1
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##                   2                   2                   2
##          Duster 360           Merc 240D            Merc 230
##                   2                   1                   1
##           Merc 280           Merc 280C           Merc 450SE
##                   1                   1                   2
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##                   2                   2                   3
## Lincoln Continental   Chrysler Imperial            Fiat 128
##                   3                   3                   1
##         Honda Civic      Toyota Corolla       Toyota Corona
##                   1                   1                   1
##    Dodge Challenger         AMC Javelin          Camaro Z28
##                   2                   2                   2
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##                   2                   1                   1
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##                   1                   2                   1
##       Maserati Bora          Volvo 142E
##                   2                   1
cars.kmed.3$silinfo$avg.width  #printing the average silhouette width

## [1] 0.3980396
cars.3.clust=lapply(1:3, function(nc) row.names(cars)[cars.kmed.3$clustering==nc])

cars.3.clust   # printing the clusters in terms of the car names

## [[1]]
```

```
X4"       "Mazda RX4 Wag"  "Datsun 710"     "Merc 240D"
0"        "Merc 280"       "Merc 280C"      "Fiat 128"
ivic"     "Toyota Corolla" "Toyota Corona"  "Fiat X1-9"
 914-2"   "Lotus Europa"   "Ferrari Dino"   "Volvo 142E"
```

sportcars

```
##  [1] "Hornet 4 Drive"    "Hornet Sportabout" "Valiant"
##  [4] "Duster 360"        "Merc 450SE"        "Merc 450SL"
##  [7] "Merc 450SLC"       "Dodge Challenger"  "AMC Javelin"
## [10] "Camaro Z28"        "Pontiac Firebird"  "Ford Pantera L"
## [13] "Maserati Bora"
##
## [[3]]
## [1] "Cadillac Fleetwood"  "Lincoln Continental" "Chrysler Imperial"
```

highend cars.

From choosing $k = 3$, we see that Cluster 1 seems to be mostly compact cars, Cluster 2 is sports cars, Cluster 3 is large luxury sedans.

We write a function to calculate the average silhouette width and here is a variety of choices of k.

```
my.k.choices = 2:8
avg.sil.width=  rep(0, times=length(my.k.choices))
for (ii in (1:length(my.k.choices)) ){
  avg.sil.width[ii] <- pam(cars.std, k=my.k.choices[ii])$silinfo$avg.width
}
print( cbind(my.k.choices, avg.sil.width) )
```

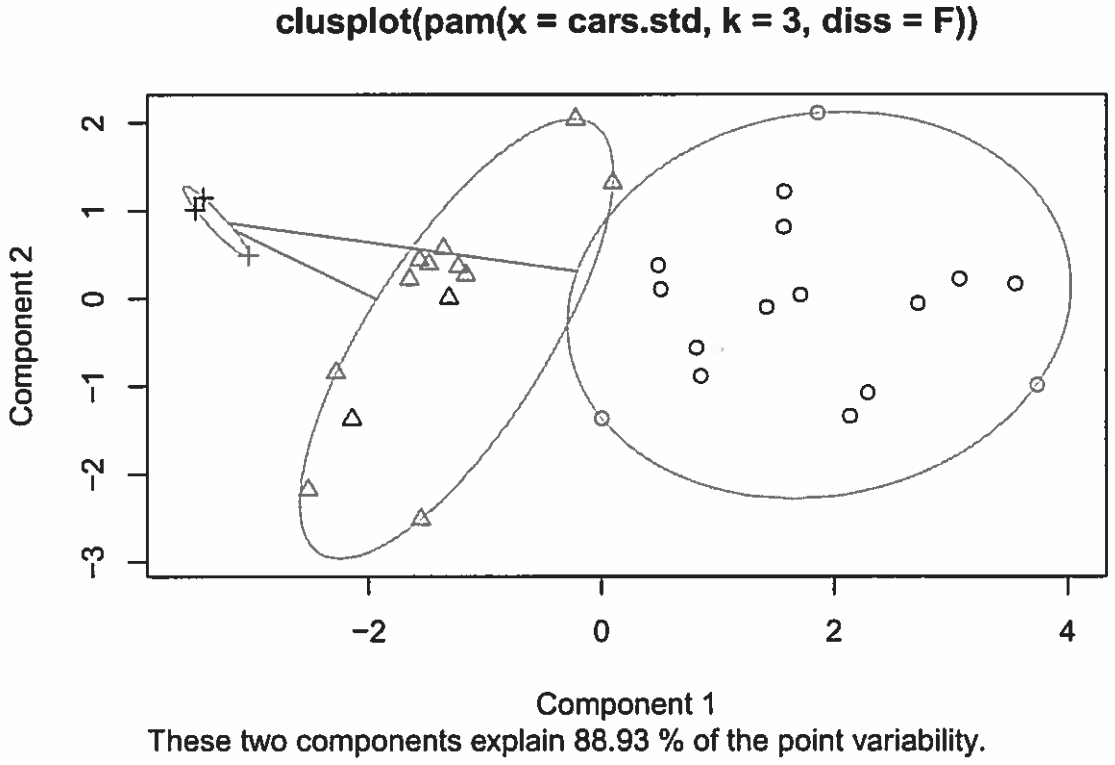
```
##      my.k.choices avg.sil.width
## [1,]            2     0.4569631
## [2,]            3     0.3980396
## [3,]            4     0.3389684
## [4,]            5     0.4023427
## [5,]            6     0.3745073
## [6,]            7     0.3779733
## [7,]            8     0.3600124
```

A LARGE average silhouette width indicates that the observations are properly clustered. So maybe $k = 2$ is the best choice of k here?

How do we visualize the Clusters? There is built-in plots available with the pam() function

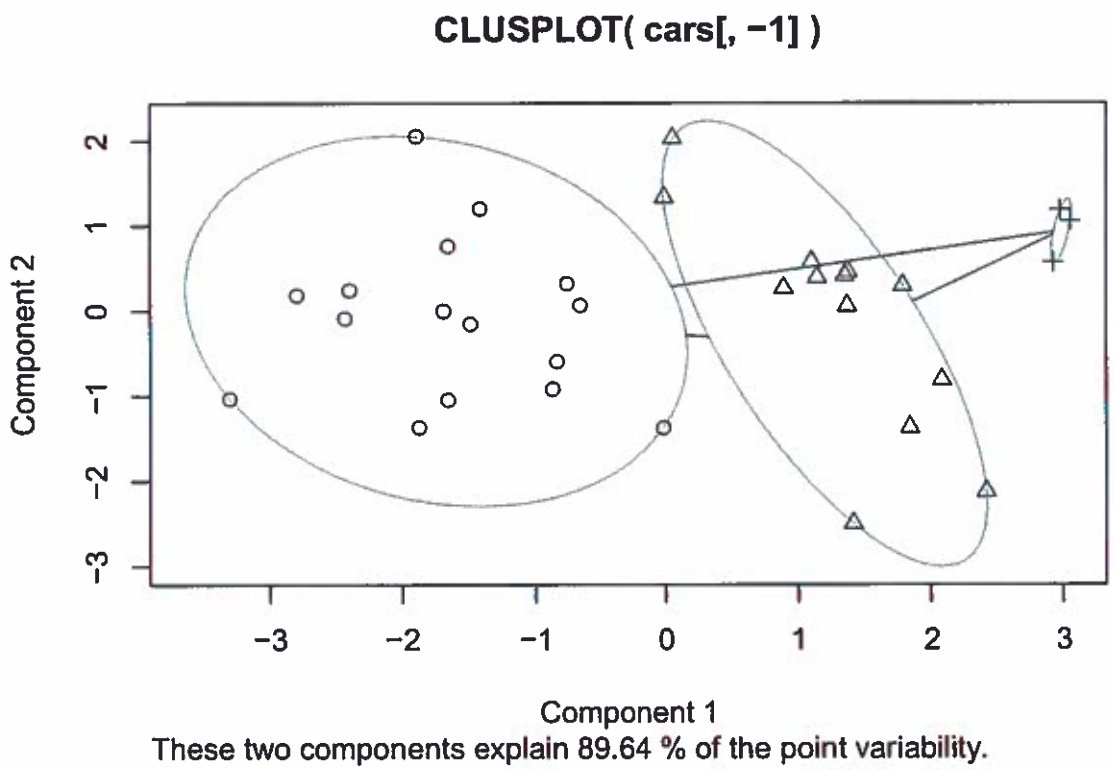
```
# The "clusplot":
plot(cars.kmed.3, which.plots=1)
```

put those people into clusters (# of clusters depends on # that we want)



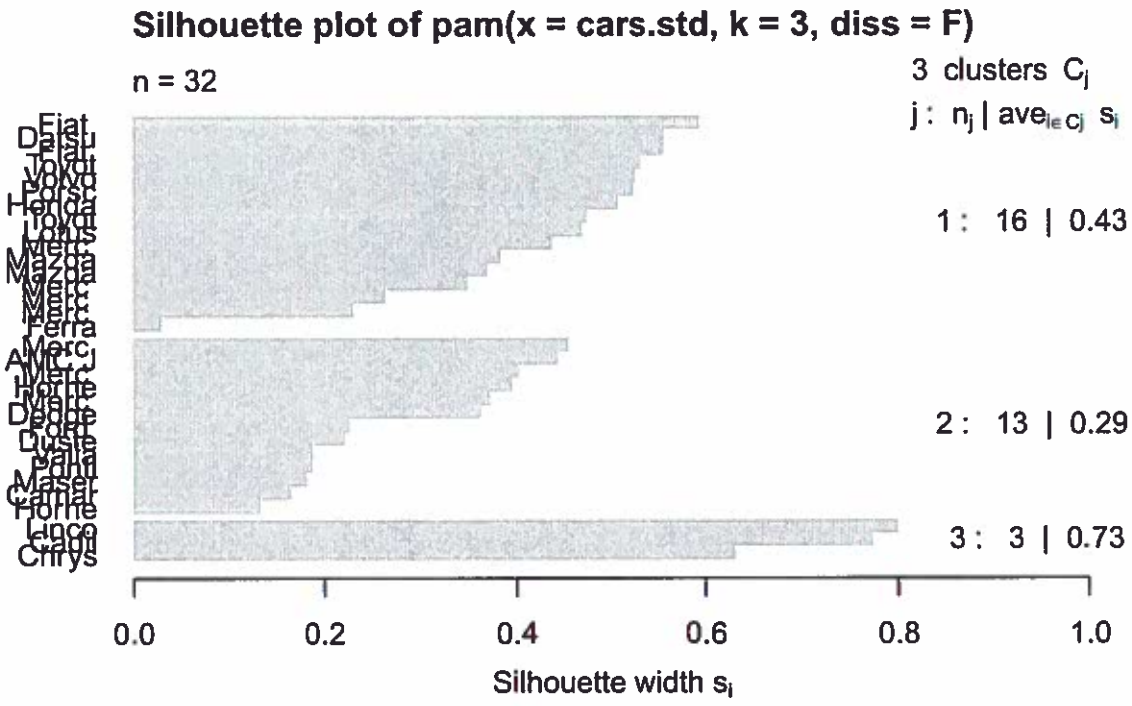
The clusplot (in the "cluster" library) can actually be used with any clustering partition by entering the data set and the clustering vector, e.g.:

```
clusplot(cars[,-1], cars.kmed.3$cluster)
```



The "silhouette plot":

```
plot(cars.kmed.3, which.plots=2)
```



3 Model-based clustering.

This shows which observations are "best clustered".

7. Model-based Clustering

EM algorithm.
Bayesian approach (it's useful to read this algorithm)

Neither hierarchical nor partitioning methods assume a specific statistical model for the data. They are strictly exploratory tools, and no formal inference about a wider population is possible. Model-based clustering assumes that the population generating the data consists of k subpopulations, which correspond to the k clusters. Therefore, the distribution for the data is assumed to be composed of k densities. This idea was originally proposed by Scott and Symons (1971) but fully developed in recent years by Banfield and Raftery (1993) and Fraley and Raftery (2002).

Clustering Model Setup: let $\gamma = [\gamma_1, \gamma_2, \cdots, \gamma_n]^T$ be a vector of cluster labels, such that $\gamma_i = j$ if observation $\mathbf{x}_i$ is from the j-th subpopulation. Suppose the subpopulation densities are denoted by $\mathbf{f}_j(\mathbf{x}; \theta)$, where θ_j contains the set of unknown parameters for the j-th density. Then the likelihood, given the observed data, is:

$$L(\theta_1, \theta_2, \cdots, \theta_k, \gamma \mid \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n) = \Pi_{i=1}^n f_{\gamma_i}(\mathbf{x}_i; \theta_{\gamma_i})$$

Fitting the model amounts to choosing $\theta_1, \theta_2, \cdots, \theta_k, , \gamma$ to maximize this likelihood. The estimated γ is the "clustering vector" that defines which cluster each object is assigned to.

The Multivariate Normality Assumption:

Assume that each subpopulation ($j = 1, \cdots, k$) follows a multivariate normal density having mean vectors μ_j and covariance matrices Σ_j , for $j = 1, \cdots, k$, as its parameters. Then the likelihood becomes

$$L(\theta_1, \theta_2, \cdots, \theta_k, \gamma) \propto \Pi_{j=1}^k \Pi_{i \in \mathbf{r}_j} \mid \Sigma_j \mid^{1/2} \exp[-\frac{1}{2}(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j)].$$

The MLE of μ_j is $\bar{\mathbf{x}}_j$, the sample mean vector for the observations in subpopulation j. Replacing μ_j with $\bar{\mathbf{x}}_j$, the log-likelihood function is a constant plus

$$-\frac{1}{2} \sum_{j=1}^k trace(\mathbf{W}_j \Sigma_j^{-1} + n \ln \mid \Sigma_j \mid),$$

where $\mathbf{W}_j$ is a matrix containing the sums of squares and cross products of variables for observations in subpopulation j. We can assume a certain structure for the covariance matrices Σ_j for $j = 1, \ldots, k$ and then determine computationally the value of γ that maximizes this (log) likelihood.

We consider a few **possible covariance structures:**

1. A simple (maybe unrealistic!) assumption is that each subpopulation has the same covariance structure and that all the $\Sigma_j = \sigma^2 \mathbf{I}$. In this case, γ is chosen so that the total within-group sum-of-squares $trace(\sum_{j=1}^k \mathbf{W}_j)$ is minimized. This tends to produce clusters that are spherical and roughly of equal size.
2. A slightly more complicated assumption is that each subpopulation has the same covariance structure, i.e., $\Sigma_j = \Sigma$ for all $j = 1, \ldots, k$. This tends to produce clusters that are elliptical with roughly the same directional slope.
3. An extremely unrestrictive assumption is that each subpopulation may have a completely different covariance structure, $\Sigma_j, j = 1, \ldots, k$. This may produce clusters that are different in size, shape, and orientation. We might consider assumptions that are less restrictive than the equal-covariances assumption yet more parsimonious than the unstructured-covariances assumption.

The covariance structure we assume leads to a clustering solution in which the sizes, shapes, and orientations of the clusters might be the same or different. In practice, the R function 'Mclust()' in the mclust package considers many such models, letting the covariance assumptions and the number of clusters k vary. Usually the Bayesian information Criterion (BIC) is used to choose the best of all these competing models and thus determine the model-based clustering result.

Example: USArrests data

In this example, we will perform a model-based clustering of the USArrests data. The data contains 50 states based on these 4 variables. Use

```
help(USArrests)
```

to see detail.
The R function 'Mclust()' performs model-based clustering for a range of models and a variety of values of k:

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 3.4.3
```

```
## Package 'mclust' version 5.4
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:maps':
##
##     map
```

```
arrest.clus <- Mclust(USArrests)
```

G = 2 ... ← or whatever number that we believe that it is the true # of clusters.

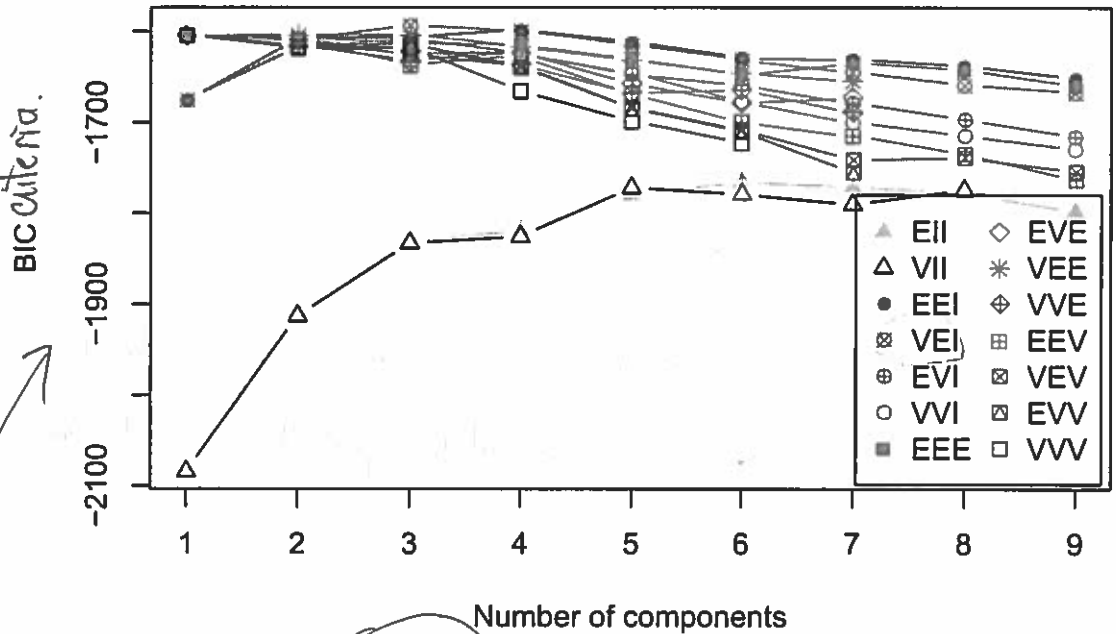
By default, the models considered are: - "EII": spherical, equal volume

- "VII": spherical, unequal volume
- "EEI": diagonal, equal volume and shape
- "VEI": diagonal, varying volume, equal shape
- "EVI": diagonal, equal volume, varying shape
- "VVI": diagonal, varying volume and shape
- "EEE": ellipsoidal, equal volume, shape, and orientation
- "EEV": ellipsoidal, equal volume and equal shape
- "VEV": ellipsoidal, equal shape
- "VVV": ellipsoidal, varying volume, shape, and orientation

We plot the BIC values: (Bayesian information criterion.

```
plot(arrest.clus, data=USArrests, what="BIC")
```

BIC : Bayesian information criterion.



The best solution is VEI with 3 clusters.

can't determine # of clusters..

```
summary(arrest.clus)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEI (diagonal, equal shape) model with 3 components:
##
##  log.likelihood  n df       BIC      ICL
##       -757.5594 50 20 -1593.359 -1597.95
##
## Clustering table:
##  1  2  3
## 20 10 20
```

with this

To see the clustering vector:

```
clus.vec.3 <- arrest.clus$classification
arrest.3.clust <- lapply(1:3, function(nc) row.names(USArrests)[clus.vec.3==nc])
arrest.3.clust
```

```
## [[1]]
##  [1] "Alabama"        "Alaska"         "Arizona"        "California"
##  [5] "Colorado"       "Florida"        "Georgia"        "Illinois"
##  [9] "Louisiana"      "Maryland"       "Michigan"       "Mississippi"
## [13] "Missouri"       "Nevada"         "New Mexico"     "New York"
## [17] "North Carolina" "South Carolina" "Tennessee"      "Texas"
##
## [[2]]
##  [1] "Idaho"         "Iowa"          "Maine"         "Minnesota"
##  [5] "New Hampshire" "North Dakota"  "South Dakota"  "Vermont"
##  [9] "West Virginia" "Wisconsin"
```

```
##
## [[3]]
##  [1] "Arkansas"      "Connecticut"   "Delaware"      "Hawaii"
##  [5] "Indiana"       "Kansas"        "Kentucky"      "Massachusetts"
##  [9] "Montana"       "Nebraska"      "New Jersey"    "Ohio"
## [13] "Oklahoma"      "Oregon"        "Pennsylvania"  "Rhode Island"
## [17] "Utah"          "Virginia"      "Washington"    "Wyoming"
```

This gives the probabilities of belonging to each cluster for every object:

```
round(arrest.clus$z,2)
```

```
##                [,1] [,2] [,3]
## Alabama        1.00 0.00 0.00
## Alaska         1.00 0.00 0.00
## Arizona        1.00 0.00 0.00
## Arkansas       0.41 0.00 0.59
## California     1.00 0.00 0.00
## Colorado       1.00 0.00 0.00
## Connecticut    0.00 0.02 0.98
## Delaware       0.37 0.00 0.63
## Florida        1.00 0.00 0.00
## Georgia        1.00 0.00 0.00
## Hawaii         0.00 0.00 1.00
## Idaho          0.00 0.64 0.36
## Illinois       1.00 0.00 0.00
## Indiana        0.00 0.00 1.00
## Iowa           0.00 1.00 0.00
## Kansas         0.00 0.00 1.00
## Kentucky       0.01 0.00 0.99
## Louisiana      1.00 0.00 0.00
## Maine          0.00 1.00 0.00
## Maryland       1.00 0.00 0.00
## Massachusetts  0.00 0.00 1.00
## Michigan       1.00 0.00 0.00
## Minnesota      0.00 0.85 0.15
## Mississippi    1.00 0.00 0.00
## Missouri       0.66 0.00 0.34
## Montana        0.00 0.02 0.98
## Nebraska       0.00 0.10 0.90
## Nevada         1.00 0.00 0.00
## New Hampshire  0.00 1.00 0.00
## New Jersey     0.01 0.00 0.99
## New Mexico     1.00 0.00 0.00
## New York       1.00 0.00 0.00
## North Carolina 1.00 0.00 0.00
## North Dakota   0.00 1.00 0.00
## Ohio           0.00 0.00 1.00
## Oklahoma       0.00 0.00 1.00
## Oregon         0.01 0.00 0.99
## Pennsylvania   0.00 0.00 1.00
## Rhode Island   0.00 0.00 1.00
## South Carolina 1.00 0.00 0.00
## South Dakota   0.00 0.99 0.01
## Tennessee      1.00 0.00 0.00
```

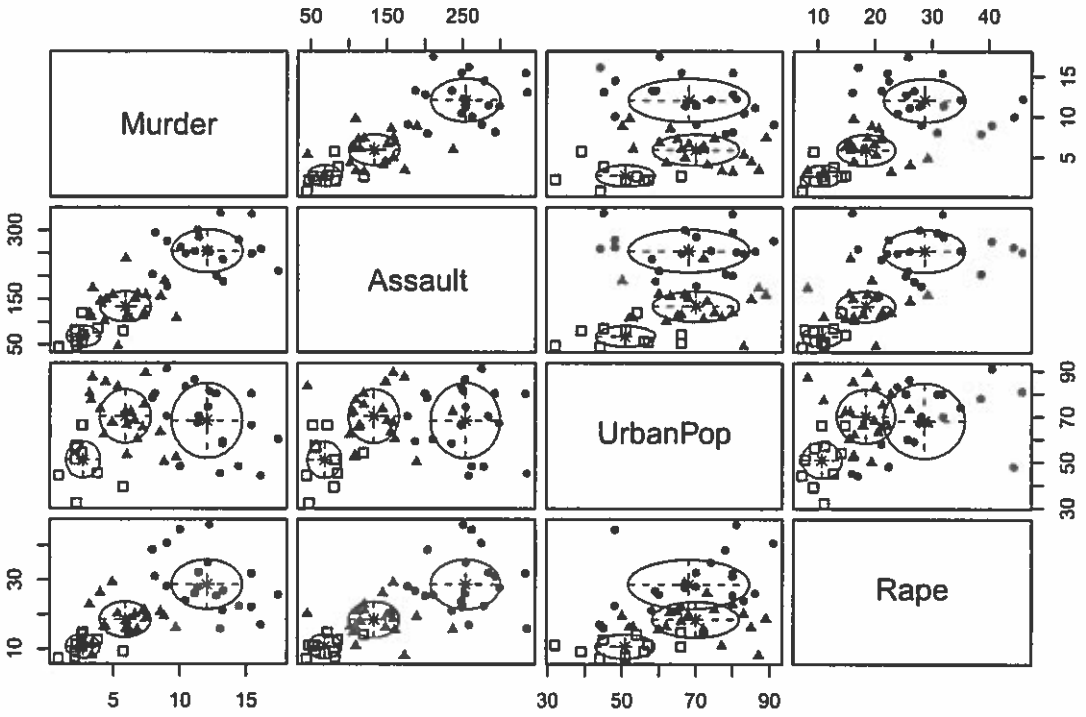
conclude that Alabama is in the 1st cluster.

It is hard to say which cluster Arkansas is in.

```
## Texas          1.00 0.00 0.00
## Utah           0.00 0.00 1.00
## Vermont        0.00 1.00 0.00
## Virginia       0.02 0.00 0.98
## Washington     0.00 0.00 1.00
## West Virginia  0.00 0.96 0.04
## Wisconsin      0.00 0.99 0.01
## Wyoming        0.00 0.00 1.00
```

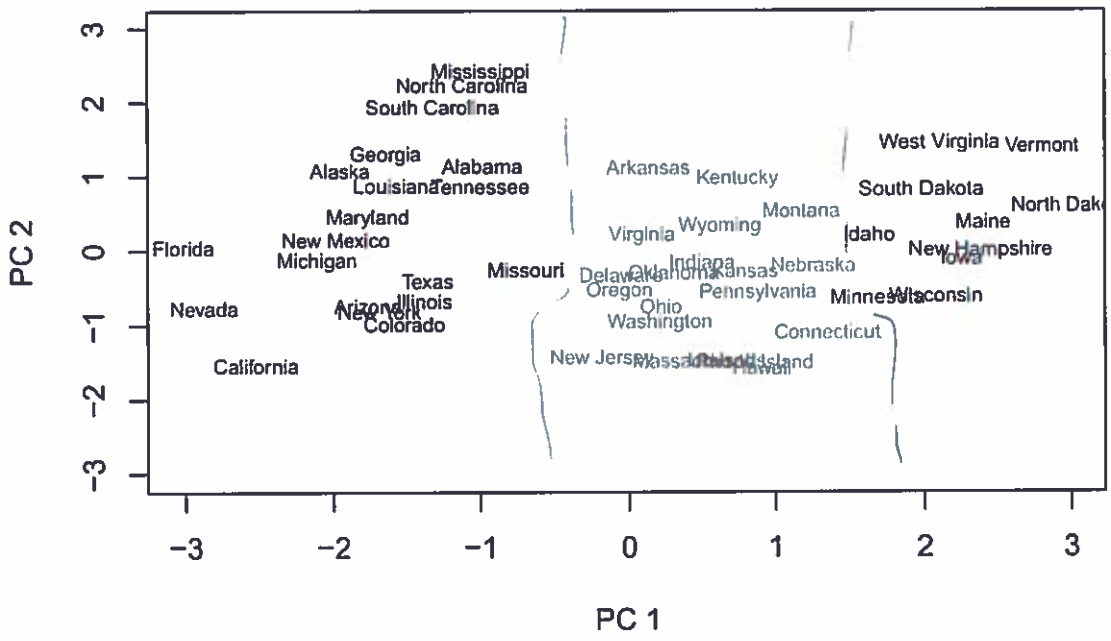
To visualize the clusters via a scatterplot matrix:

```
plot(arrest.clus, what="classification")
```



To visualize the clusters via a plot of the scores on the first 2 principal components, with the clusters separated by color:

```
arrests.pc <- princomp(USArrests,cor=T)
# Setting up the colors for the 5 clusters on the plot:
my.color.vector <- rep("blue", times=nrow(USArrests))
my.color.vector[arrest.clus$classification==2] <- "red"
my.color.vector[arrest.clus$classification==3] <- "green"

# Plotting the PC scores:
plot(arrests.pc$scores[,1], arrests.pc$scores[,2], ylim=range(arrests.pc$scores[,1]),
xlab="PC 1", ylab="PC 2", type ='n', lwd=2)
text(arrests.pc$scores[,1], arrests.pc$scores[,2], labels=row.names(USArrests),
cex=0.7, lwd=2, col=my.color.vector)
```



To review the PCA:

```
summary(arrests.pc,loadings=T)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4
## Standard deviation     1.5748783 0.9948694 0.5971291 0.41644938
## Proportion of Variance 0.6200604 0.2474413 0.0891408 0.04335752
## Cumulative Proportion  0.6200604 0.8675017 0.9566425 1.00000000
##
## Loadings:
##          Comp.1 Comp.2 Comp.3 Comp.4
## Murder   -0.536  0.418 -0.341  0.649
## Assault  -0.583  0.188 -0.268 -0.743
## UrbanPop -0.278 -0.873 -0.378  0.134
## Rape     -0.543 -0.167  0.818
```

Note PC1 is an overall "lack-of-crime" index and PC2 is a "rural" index.

lack of crime because we have negative sign here.

Chapter 9. Multidimensional Scaling

Jianxuan Liu

Fall 2018

This chapter discuss methods for displaying multivariate data in low-dimensional space. The objective is to "fit" the original data into a low-dimensional coordinate system such that any distortion caused by a reduction in dimensionality is minimized

Distortion generally refers to the similarities or dissimilarities (distances) among the original data points.

Specifically, multidimensional scaling techniques deal with the following problem: For a set of observed similarities or distances between every pair of n items, find a representation of the items in few dimensions such that the interitem proximities "nearly match" the original similarities or distances. Multidimensional Scaling can be viewed as a way of generating a geometric representation of some observed proximity matrix.

The proximity matrix could contain similarity values for pairs of observations or dissimilarity values, but we will typically work with dissimilarities (i.e., distances).

With multidimensional scaling, we begin with a distance matrix and produce a "possible data set" that could have yielded such a distance matrix. Note that MDS can work on either objects or variables.

Example: Sports rankings data

```
source("Sports.txt")
head(sportsranks)
```

```
##  Baseball Football Basketball Tennis Cycling Swimming Jogging
##         1        3          7      2       4        5       6
##         1        3          2      5       4        7       6
##         1        3          2      5       4        7       6
##         4        7          3      1       5        6       2
##         2        3          1      7       6        5       4
##         6        3          4      7       2        1       5
```

```
dim(sportsranks)
```

```
## [1] 130   7
```

```
cor(sportsranks)
```

```
##                Baseball   Football  Basketball      Tennis     Cycling
## Baseball     1.00000000  0.2816173  0.06225502 -0.25624133 -0.39268842
## Football     0.28161733  1.0000000  0.22412213 -0.27650958 -0.48593004
## Basketball   0.06225502  0.2241221  1.00000000 -0.18105420 -0.35726588
## Tennis      -0.25624133 -0.2765096 -0.18105420  1.00000000 -0.09455839
## Cycling     -0.39268842 -0.4859300 -0.35726588 -0.09455839  1.00000000
## Swimming    -0.49738160 -0.4740188 -0.35368023 -0.04047178  0.35106904
## Jogging     -0.28615216 -0.3892162 -0.33632916 -0.11179076  0.07558857
##                Swimming     Jogging
## Baseball    -0.49738160 -0.28615216
## Football    -0.47401883 -0.38921615
## Basketball  -0.35368023 -0.33632916
## Tennis      -0.04047178 -0.11179076
```

```
## Cycling      0.35106904  0.07558857
## Swimming     1.00000000  0.05342708
## Jogging      0.05342708  1.00000000
D=dist(t(sportsranks))
```

Here we have

$$d_{ij} = || \mathbf{z}_{(i)} - \mathbf{z}_{(j)} ||$$

for $1 \leq i, j \leq 7$ where $\mathbf{z}_{(1)}, \cdots, \mathbf{z}_{(7)} \in \mathcal{R}^{130}$. We want to find $\mathbf{x}_{(1)}, \cdots, \mathbf{x}_{(7)} \in \mathcal{R}^2$, that is, $k = 2$ such that

$$d_{ij} = || \mathbf{z}_{(i)} - \mathbf{z}_{(j)} || \approx || \mathbf{x}_i - \mathbf{x}_j ||$$

Here the traditional idea of objects and variables have been reversed. The $n = 7$ objects are different sport activities, and the $p = 130$ measurements taken on each sport are simply 130 individual rankings.

MDS is a family of different algorithms, each designed to arrive at some optimal low-dimensional configuration (usually $k = 2$ or $k = 3$)

With multidimensional scaling methods include

1. Classical MDS
2. Metric MDS
3. Non-metric MDS

1. Classical MDS

Suppose we have Euclidean distance matrix $\mathbf{D} = (d_{ij})$. The objective of classical multidimensional scaling is to find $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ such that

$$|| \mathbf{x}_i - \mathbf{x}_j || = d_{ij}$$

. Such a solution is not unique, because if $\mathbf{X}$ is the solution, then $\mathbf{x}_i^* = \mathbf{x} + c, c \in \mathcal{R}^k$ aksi satusfies

$$|| \mathbf{x}_i^* - \mathbf{x}_j^* || = || (\mathbf{x}_i + c) - (\mathbf{x}_j - c) || = d_{ij}$$

Any location c can be used, but the assumption of centered configuration, i.e.,

$$\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{0}$$

serves well for the purpose of dimension reduction. The classical MDS finds the centered configuration $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{R}^k$ for some $k \geq n - 1$ so that their pairwise distances are the same as those corresponding distances in $\mathbf{D}$. We may find the $n \times n$ Gram matrix $\mathbf{B} = \mathbf{X}^T\mathbf{X}$ rather than $\mathbf{X}$. The Gram matrix is the matrix of inner products. Denote the ij^{th} element of $\mathbf{B}$ as b_{ij} .

Theorem: Let $[\mathbf{x}_1, \cdots, \mathbf{x}_n]$ be centered data, $d_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)$, and $\mathbf{B} = \mathbf{X}^T\mathbf{X}$.

1. Then the $n \times n$ matrix $\mathbf{B}$ has entries

$$\begin{aligned} b_{ij} &= -\frac{1}{2}(d_{ij}^2 - \frac{2}{n}\sum_{i=1}^{n} d_{ij}^2 + \frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2) \\ &= -\frac{1}{2}(d_{ij}^2 - d_{\cdot j}^2 - d_{i\cdot}^2 + d_{\cdot\cdot}^2) \end{aligned}$$

where

- d_{j}^2 is the average of $\{d_{ij} : i = 1, \cdots, n\}$ for each j
- d_{i}^2 is the average of $\{d_{ij} : j = 1, \cdots, n\}$ for each i
- $d_{\cdot\cdot}^2$ is the average of $\{d_{ij} : i, j = 1, \cdots, n\}$

2. If $\mathbf{B} = V\Lambda V^T$ is the spectral decomposition of $\mathbf{B}$, then $\mathbf{X} = \Lambda^{1/2}V^T$ defines a configuration in k variables which minimize the STRESS.

$X = [x_1, x_2, \cdots, x_n]$ be centered data $d_{ij}^2 = (x_i - x_j)^T(x_i - x_j)$ $B = X^TX$

* Note that $d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j = d_{ji}^2$ (0)

So we have $\frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 = x_i^T x_j + \frac{1}{n}\sum_{j=1}^{n} x_j^T x_j \Rightarrow x_i^T x_i = \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 - \frac{1}{n}\sum_{j=1}^{n} x_i^T x_j$ (1)

$\Rightarrow \frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2 = \frac{2}{n}\sum_{i=1}^{n} x_i^T x_i$ (2)

* Consider $B = X^TX$

So we have $b_{ij} = x_i^T x_j$ (1)

So we have $b_{ij} \overset{(0)}{=} -\frac{1}{2}\left(d_{ij}^2 - \overbrace{x_i^T x_i}^{(1)} - x_j^T x_j\right) - \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2$

$= -\frac{1}{2}\left(d_{ij}^2 - \underbrace{\frac{1}{n}\sum_{j=1}^{n} d_{ij}^2}_{} + \underbrace{\frac{1}{n}\sum_{j=1}^{n} x_j^T x_j}_{} + \frac{1}{n}\sum_{i=1}^{n} x_i^T x_i\right)$

$\overset{(2)}{=} -\frac{1}{2}\left(d_{ij}^2 - \frac{2}{n}\sum_{j=1}^{n} d_{ij}^2 + \frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2\right)$

The space which $\mathbf{X}$ lies is the eigenspace where the first coordinate contains the largest variation, and is identified with $\mathcal{R}^k$. If we wish to reduce the dimension to $k \leq p$, then the first k rows of $\mathbf{X}_k$ best preserves the distances d_{ij} among all other linear dimension reduction of $\mathbf{X}$ (to k). Then

$$\mathbf{X}_{(k)} = \Lambda_k^{1/2} V_k^T$$

where Λ_k is the first $k \times k$ sub matrix of Λ, V_k is collected through the first k columns of V.

Classical MDS gives configurations $\mathbf{X}_{(k)}$ in $\mathcal{R}^k$ for any dimension $k = 1, \cdots, p$. The coordinates are given by the principal order of largest-to-smallest variances. Dimension reduction from $\mathbf{X} = \mathbf{X}_{(p)}$ to $\mathbf{X}_{(k)}, k < p$ is the same as PCA.

We can reduce the dimension of the solution by restricting attention to the k largest eigenvalues. If the distances are not Euclidean, $\mathbf{B}$ is not positive definite and some eigenvalues of $\mathbf{B}$ will be negative. In this case, we can still choose the dimension corresponding to the k largest positive eigenvalues.

Determining the Amount of Data Reduction

When using MDS to "reduce the dimensionality" from p to k, what is a proper choice of k?

If there are k "relatively large" eigenvalues of $\mathbf{B}$, this is evidence that a k-dimensional solution is appropriate. We could base the choice of k on the sizes of the first few eigenvalues $\lambda_1, \lambda_2, \cdots$ (listed in a decreasing order). We calculate (for each possible k)

$$P_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} |\lambda_i|}$$

Values of k that yield a P_k near 1 would give a good representation. The 'cmdscale' function in R prints this criterion when the "eig=T"' option is specified.

There is another option in choosing k. For each possible value of k, we minimize

$$\phi = \sum_{i,j} (d_{ij}^2 - \hat{d}_{ij}^2)$$

where d_{ij}^2 is the Euclidean distance between the i-th and j-th observations in the full p-dimensional space, and $\hat{d}_{ij}^2$ is the Euclidean distance between the i-th and j-th observations in the reduced k-dimensional space. As k increases, the minimum value of ϕ will decrease monotonically. We can plot this minimum against the various values of k and pick the k value at the "elbow" of the plot. Takane et al. (1977) suggested a scaled version of ϕ called SStress that always lies between 0 and 1:

$$SStress = \left[\frac{\sum_{i<j} (d_{ij}^2 - \hat{d}_{ij}^2)^2}{\sum_{i<j} d_{ij}^4} \right]^{1/2}$$

Values of SStress below 0.1 represent a good fit.

Examples: European distances

```
# Classical MDS using the cmdscale function:
# The default number of dimensions is k = 2:
class(eurodist)
```

```
## [1] "dist"
```

```
euro.mds.2 <- cmdscale(eurodist, eig=T)
euro.mds.2
```

```
## $points
##                         [,1]        [,2]
## Athens           2290.274680  1798.80293
## Barcelona        -825.382790   546.81148
## Brussels           59.183341  -367.08135
## Calais            -82.845973  -429.91466
## Cherbourg        -352.499435  -290.90843
## Cologne           293.689633  -405.31194
## Copenhagen        681.931545 -1108.64478
## Geneva             -9.423364   240.40600
## Gibraltar       -2048.449113   642.45854
## Hamburg           561.108970  -773.36929
## Hook of Holland   164.921799  -549.36704
## Lisbon          -1935.040811    49.12514
## Lyons            -226.423236   187.08779
## Madrid          -1423.353697   305.87513
## Marseilles       -299.498710   388.80726
## Milan             260.878046   416.67381
## Munich            587.675679    81.18224
```

```
## Paris            -156.836257  -211.13911
## Rome              709.413282  1109.36665
## Stockholm         839.445911 -1836.79055
## Vienna            911.230500   205.93020
##
## $eig
##  [1]  1.953838e+07  1.185656e+07  1.528844e+06  1.118742e+06  7.893472e+05
##  [6]  5.816552e+05  2.623192e+05  1.925976e+05  1.450845e+05  1.079673e+05
## [11]  5.139484e+04 -3.259629e-09 -9.496124e+03 -5.305820e+04 -1.322166e+05
## [16] -2.573360e+05 -3.326719e+05 -5.162523e+05 -9.191491e+05 -1.006504e+06
## [21] -2.251844e+06
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.7537543 0.8679134
```

The first number in the GOF section is what we called P_k Now change the number of dimensions k:

```
euro.mds.4 <- cmdscale(eurodist, k=4, eig=T)
euro.mds.4
```

```
## $points
##                        [,1]        [,2]       [,3]        [,4]
## Athens           2290.274680  1798.80293   53.79314 -103.826958
## Barcelona        -825.382790   546.81148 -113.85842   84.585831
## Brussels           59.183341  -367.08135  177.55291   38.797514
## Calais            -82.845973  -429.91466  300.19274  106.353695
## Cherbourg        -352.499435  -290.90843  457.35294  111.449150
## Cologne           293.689633  -405.31194  360.09323 -636.202379
## Copenhagen        681.931545 -1108.64478   26.09257  151.693056
## Geneva             -9.423364   240.40600 -344.20659  656.121110
## Gibraltar       -2048.449113   642.45854  167.86631   78.621423
## Hamburg           561.108970  -773.36929   80.91722   48.548472
## Hook of Holland   164.921799  -549.36704  270.82327  116.886334
## Lisbon          -1935.040811    49.12514 -483.02056 -315.241752
## Lyons            -226.423236   187.08779 -358.43234 -257.737009
## Madrid          -1423.353697   305.87513  253.26763    2.478812
## Marseilles       -299.498710   388.80726 -109.17417   12.651217
## Milan             260.878046   416.67381 -171.52428   20.926369
## Munich            587.675679    81.18224  -75.88485   13.080496
## Paris            -156.836257  -211.13911  131.30852   27.089432
## Rome              709.413282  1109.36665 -179.83052 -109.895049
## Stockholm         839.445911 -1836.79055 -541.35188 -108.755016
## Vienna            911.230500   205.93020   98.02313   62.375253
##
## $eig
##  [1]  1.953838e+07  1.185656e+07  1.528844e+06  1.118742e+06  7.893472e+05
##  [6]  5.816552e+05  2.623192e+05  1.925976e+05  1.450845e+05  1.079673e+05
## [11]  5.139484e+04 -3.259629e-09 -9.496124e+03 -5.305820e+04 -1.322166e+05
## [16] -2.573360e+05 -3.326719e+05 -5.162523e+05 -9.191491e+05 -1.006504e+06
```

```
## [21] -2.251844e+06
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.8173197 0.9411060
```

Using the P_k criterion for a variety of values of k to choose the appropriate amount of dimension reduction:

```
# consider values of k from 1 to 12
max.k <- 12
Pk <- rep(0,max.k)
SStress <- rep(0,max.k)
for (kk in 1:max.k){
my.mds.kk <- cmdscale(eurodist,k=kk,eig=T)
Pk[kk] <- my.mds.kk$GOF[1]
#SStress[kk] <- ( sum( (eurodist^2 - (dist(my.mds.kk$points))^2)^2 )/sum(eurodist^4) )^0.5
}
```

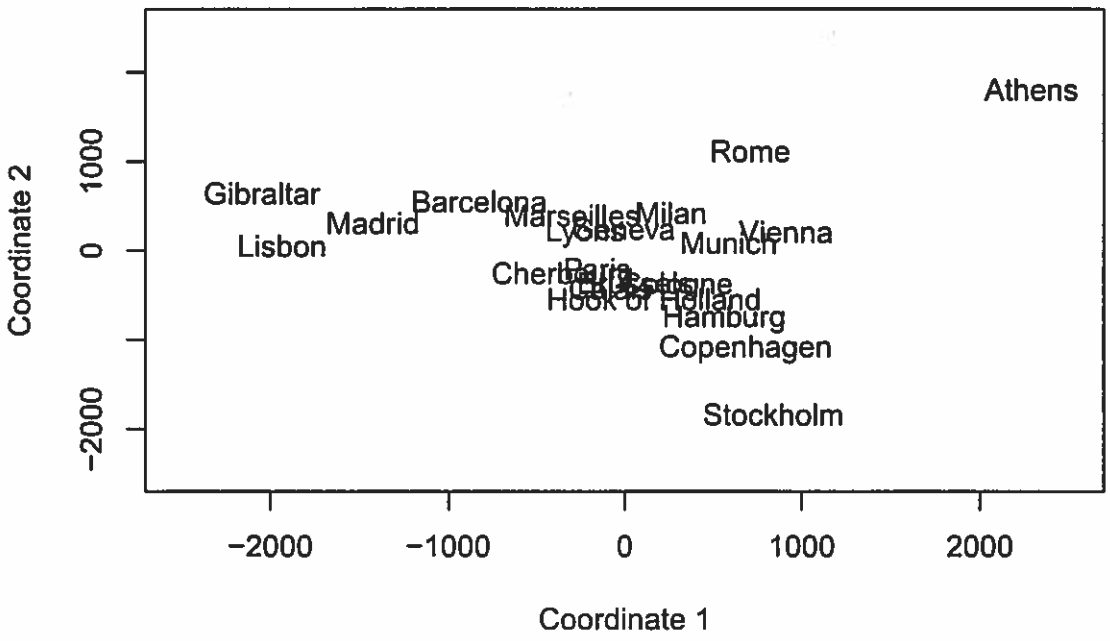
```
## Warning in cmdscale(eurodist, k = kk, eig = T): only 11 of the first 12
## eigenvalues are > 0
```

```
cbind(1:max.k,Pk)
```

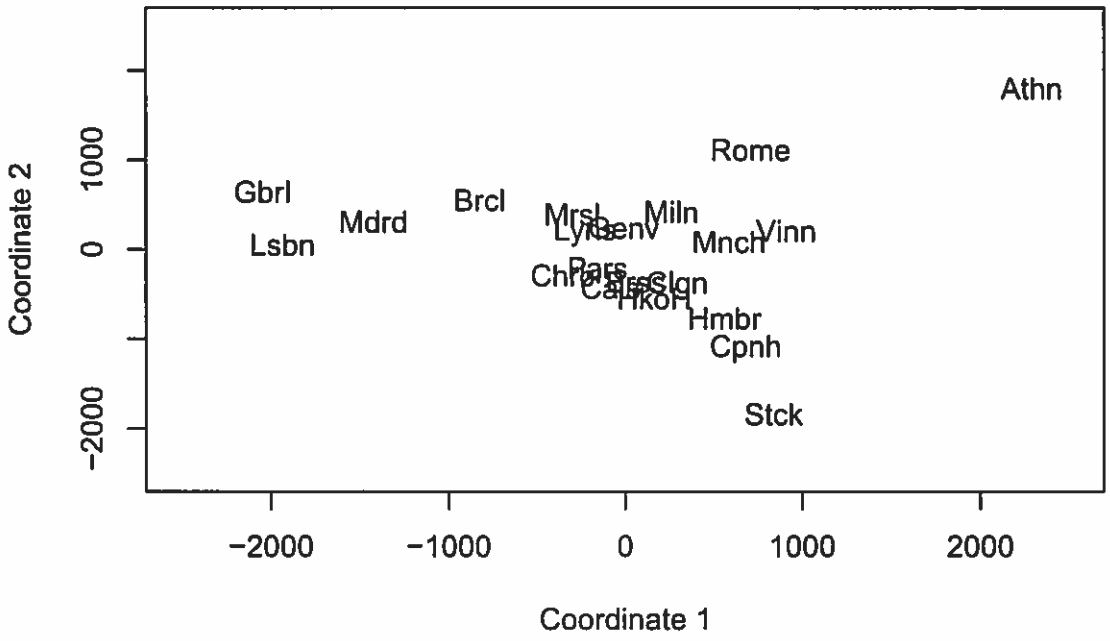
```
##                Pk
##  [1,]  1 0.4690928
##  [2,]  2 0.7537543
##  [3,]  3 0.7904600
##  [4,]  4 0.8173197
##  [5,]  5 0.8362709
##  [6,]  6 0.8502358
##  [7,]  7 0.8565337
##  [8,]  8 0.8611578
##  [9,]  9 0.8646411
## [10,] 10 0.8672332
## [11,] 11 0.8684672
## [12,] 12 0.8684672
```

It looks like 2 or 3 dimensions would be reasonable, and 4 gives quite a good fit. A 2-D representation of the solution for k=2:

```
plot(euro.mds.2$points[,1], euro.mds.2$points[,2], type='n',
     xlab="Coordinate 1", ylab="Coordinate 2", xlim=c(-2500,2500), ylim=c(-2500,2500) )
text(euro.mds.2$points[,1], euro.mds.2$points[,2], labels=labels(eurodist) )
```

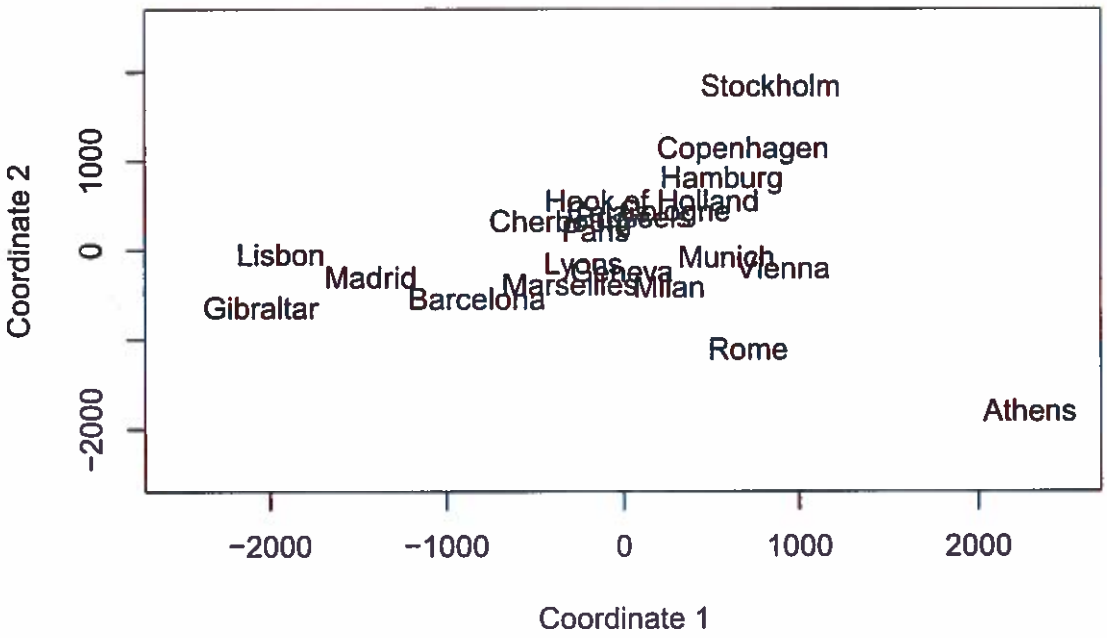



```
# use abbreviations:

euro.abb <- abbreviate(labels(eurodist))

plot(euro.mds.2$points[,1], euro.mds.2$points[,2], type='n',
     xlab="Coordinate 1", ylab="Coordinate 2", xlim=c(-2500,2500), ylim=c(-2500,2500) )
text(euro.mds.2$points[,1], euro.mds.2$points[,2], labels=euro.abb )
```




```
# Maybe rotating across the x-axis would produce a better reflection of reality

plot(euro.mds.2$points[,1], -euro.mds.2$points[,2], type='n',
```

```
    xlab="Coordinate 1", ylab="Coordinate 2", xlim=c(-2500,2500), ylim=c(-2500,2500) )
text(euro.mds.2$points[,1], -euro.mds.2$points[,2], labels=labels(eurodist) )
```



Summary:

- Classical MDS gives configurations $\mathbf{X}_{(k)}$ in $\mathcal{R}^k$ for any dimension $1 \leq k \leq p$
- Configuration is centered
- Coordinates are given by the principal scores, ordered from largest-to-smallest variation.
- Dimension reduction from $\mathbf{X} = \mathbf{X}_{(k)}$ to $\mathbf{X}_{(k)}$ $(k < p)$ is same as PCA (cutting some PC scores out).
- Leads to exact solution if the dissimilarity is based on Euclidean distances
- Can also be used for non-Euclidean distances, in fact, for any dissimilarities.

2. Metric MDS

In metric MDS, the dissimilarities $\{d_{ij}\}$ are quantitative measurements, usually Euclidean, but other distance metrics are possible. The function f is usually taken to be a parametric monotonic function, such as $f(d_{ij}) = \alpha + \beta d_{ij}$, where α and β are unknown positive coefficients. Since f is a parametric function, the distances $\{d_{ij}\}$ can be fitted to $\{f(d_{ij})\}$ by least squares method which results in metric LS scaling. If the dissimilarities are Euclidean distances and f is taken to be the identity function, then classical MDS can be viewed as an example of metric LS scaling.

Given a (low) dimension k and a monotone function f, metric MDS seeks to find an optimal configuration $\mathbf{X} \in \mathcal{R}^k$ that gives

$$f(d_{ij}) \approx \hat{d}_{ij} = \| \mathbf{x}_i - \mathbf{x}_j \|^2$$

as close as possible.

- The function f can be taken to be a parametric monotonic function, such as $f(d_{ij}) = \alpha + \beta d_{ij}$

- "As close as possible"' is now explicitly stated by square loss

$$stress = \mathcal{L}(\hat{d}_{ij}) = \left(\frac{\sum_{i<j} [\hat{d}_{ij} - f(\hat{d}_{ij})]^2}{\sum_{i<j} d_{ij}^2} \right)^{1/2}$$

and the metric MDS minimizes $\mathcal{L}(\hat{d}_{ij})$ over all $\hat{d}_{ij}$ and α and β.

Sammon Mapping

The Sammon nonlinear mapping, which has become a popular tool for pattern recognition, is generalization of the metric LS MDS. The Sammon mapping minimize the stress

$$\text{Sammon's stress}(\hat{d}_{ij}) = \frac{1}{\sum_{u<v} d_{uv}} \sum_{i<j} \frac{(\hat{d}_{ij} - d_{ij})^2}{d_{ij}}$$

This weighting system normalizes the squared-errors in pairwise distances by using the distance in the original space. As a result, Sammon mapping preserves the small d_{ij} , giving them a greater degree of importance in the fitting procedure than for larger values of d_{ij}. Optimal solution is found by numerical computation.

```
library(MASS)
```

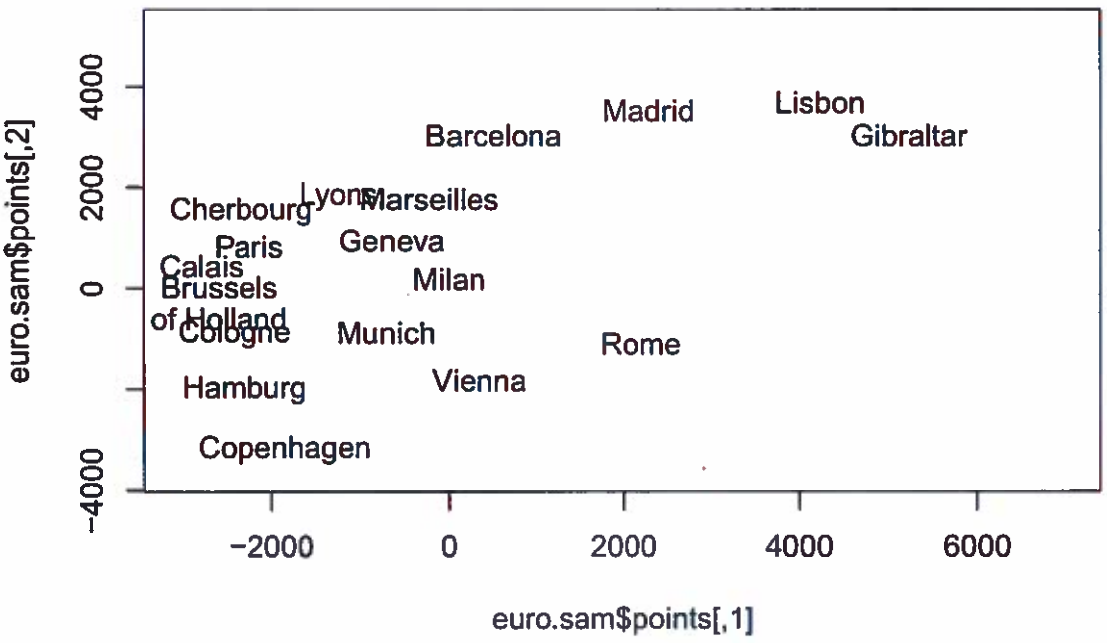
```
##
## Attaching package: 'MASS'
```

```
## The following objects are masked _by_ '.GlobalEnv':
##
##     lda, qda
```

```
euro.sam <- sammon(dist(eurodist))
```

```
## Initial stress        : 0.06361
## stress after  10 iters: 0.03351, magic = 0.092
## stress after  20 iters: 0.01925, magic = 0.500
## stress after  30 iters: 0.01864, magic = 0.500
## stress after  40 iters: 0.01843, magic = 0.500
## stress after  50 iters: 0.01835, magic = 0.500
```

```
plot(euro.sam$points, type = "n")
text(euro.sam$points[,1], -euro.sam$points[,2], labels=labels(eurodist) )
```



3. Non-metric MDS

In many applications of MDS, dissimularities are known only by their rank order, and the spacing between successively ranked dissimilarties if of no interest or is unavailable. This may happen because the data collected involve only ordinal information. In non-metric MDS, a.k.a. ordinal MDS, we assume that f is an arbitrary function that satisfies the monotonocity constraint $f(d_{ij}) \leq f(d_{kl})$ whenever $d_{ij} \leq d_{uv}$ for all $i, j, u, v = 1, \cdots, n$.

Kruskal (1964) considers construction of $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{R}^k$ from $\mathbf{D}$ based on ranks; any monotone transformation of the distances in $\mathbf{D}$ gives the same answer. A data matrix $[\mathbf{x}_1, \cdots, \mathbf{x}_n]$ with corresponding interpoint distances d_{ij} is constructed such that

$$d_{ij} \approx \| \mathbf{x}_i - \mathbf{x}_j \| .$$

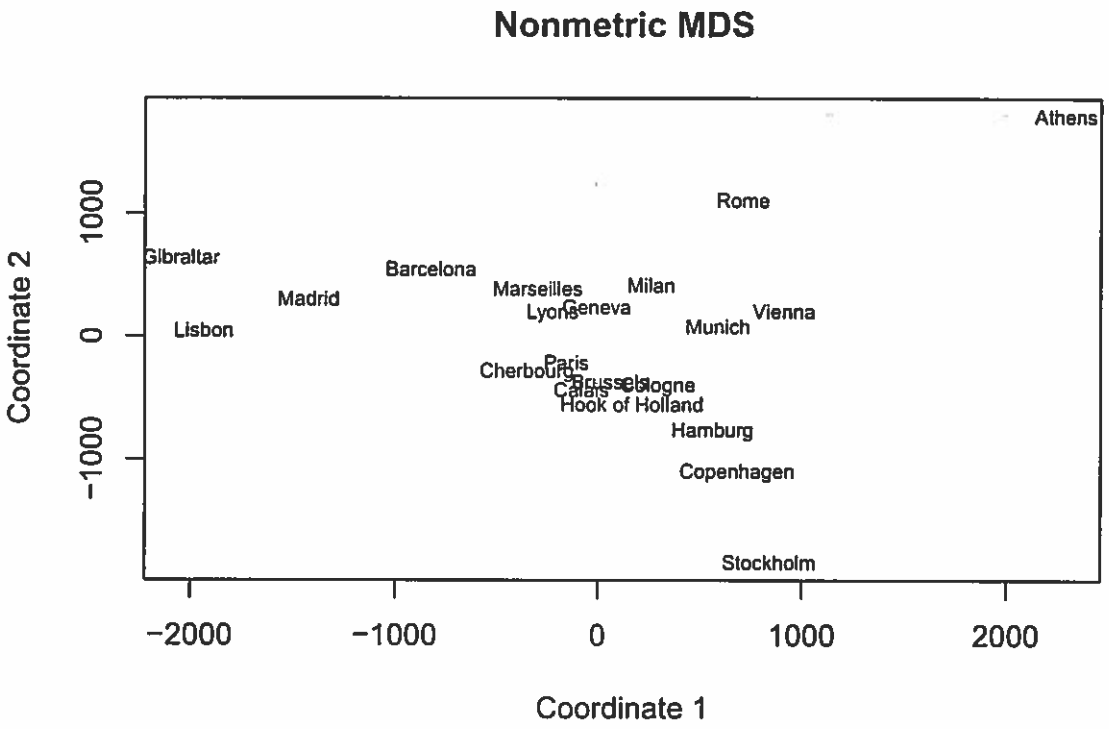
Kruskal's non-metric MDS minimizes the

$$stress(\tilde{d}_{ij}, d_{ij}) = \left(\sum_{i<j} \frac{(\tilde{d}_{ij} - \hat{d}_{ij})^2}{\sum \hat{d}_{ij}^2} \right)^{1/2}$$

```
f=isoMDS(eurodist) # Kruskal's non-metric multidimensional scaling
```

```
## initial  value 7.505733
## final  value 7.505688
## converged
```

```
x=f$points[,1]; y=f$points[,2]
plot(x,y,xlab="Coordinate 1", ylab="Coordinate 2",main="Nonmetric MDS", type="n")
text(x,y,labels=labels(eurodist),cex=0.7)
```


Nonmetric MDS
Athens
Rome
Gibraltar
Barcelona
Madrid
Marseilles
Milan
Lyons
Geneva
Vienna
Lisbon
Munich
Paris
Cherbourg
Brussels
Cologne
Calais
Hook of Holland
Hamburg
Copenhagen
Stockholm
Coordinate 2
1000
0
−1000
−2000
−1000
0
1000
2000
Coordinate 1

25/28 Tian Le

MATH 755 MULTIVARIATE STATISTICAL ANALYSIS

Assignment 1 — Due on in class Thursday September 6th

2 Q 1. Prove the cyclic property of the trace for arbitrary square matrices of common finite size ($n \geq 2$). I.e., prove

$$tr(\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_n) = tr(\mathbf{A}_{k+1}\cdots\mathbf{A}_n\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_k)$$

4 Q 2. Johnson and Wichern 2.4

3 Q 3. We have for a symmetric matrix A and positive integer m, $A^m = P\Lambda^m P^T$. Show that if A is non-singular, this holds for all integers m.

4 Q 4. Johnson and Wichern 2.8, 2.9 (Do not use a computer)

4 Q 5. Johnson and Wichern 2.12, 2.13

4 Q 6. Johnson and Wichern 2.15

4 Q 7. Johnson and Wichern 2.16 (Note that non-negative definite means positive semi definite)

Q17

Prove the cyclic property of the trace for arbitrary square matrices of common finite size ($n \geq 2$). Prove $tr(A_1 A_2 \dots A_k A_{k+1} \dots A_n) = tr(A_{k+1} \dots A_n A_1 A_2 \dots A_k)$

* Note that for all B and C are square matrices, we have $tr(BC) = tr(CB)$

We have

$$tr(\underbrace{A_1}_{B_1} \underbrace{A_2 \dots A_k A_{k+1} \dots A_n}_{C_1}) = tr(C_1 B_1) = tr(\underbrace{A_2}_{B_2} \underbrace{A_3 \dots A_k A_{k+1} \dots A_n A_1}_{C_2}) =$$

$$= tr(C_2 B_2) = tr(\underbrace{A_3}_{B_3} \underbrace{A_4 \dots A_n A_1 A_2}_{C_3}) = tr(C_3 B_3) = tr(\underbrace{A_4}_{B_4} \underbrace{A_5 \dots A_n A_1 A_2 A_3}_{C_4}) = \dots$$

Keep putting $\begin{cases} B_i = A_i \\ C_i = A_{i+1} \dots A_n A_1 \dots A_{i-1} \end{cases}$ $i = \overline{1, k}$

and apply $tr(B_i C_i) = tr(C_i B_i)$

then we have what we need to prove $\square 1$

prove

otherwise incomplete

Q2 I ∈ M 2 4

When A^{-1} and B^{-1} exist, prove that a) $(A')^{-1} = (A^{-1})'$

b) $(AB)^{-1} = B^{-1} A^{-1}$

a) We need to prove $(A')^{-1} = (A^{-1})'$ $\Rightarrow$ need to prove $\begin{cases} (A')(A^{-1})' = I \\ (A^{-1})' A' = I \end{cases}$

We have

$$\begin{cases} A^{-1}A = I = I' = (A^{-1}A)' = A'(A^{-1})' \\ A A^{-1} = I = I' = (A A^{-1})' = (A^{-1})' A' \end{cases} \Rightarrow \text{what we need to prove } \square 2a.$$

b) We need to prove $(AB)^{-1} = B^{-1}A^{-1}$ $\Rightarrow$ need to prove $\begin{cases} (AB)(B^{-1}A^{-1}) = I \\ (B^{-1}A^{-1})(AB) = I \end{cases}$

We have

$$\begin{cases} (AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AIA^{-1} = AA^{-1} = I \\ (B^{-1}A^{-1})(AB) = B^{-1}(A^{-1}A)B = B^{-1}IB = B^{-1}B = I \end{cases} \Rightarrow \text{what we need to prove } \square 2b.$$

Q37

We have for a symmetric matrix A and a positive integer m

Spectral decomposition ↓ Λ: contains eigen values of A

$A^m = P \Lambda^m P^T$ where P is an orthogonal matrix

Show that if A is non-singular then this holds for all integers m

* We have for $i = -1$, then

$$A^{-1} = (P \Lambda P^T)^{-1} \underset{P^T = P^{-1}}{\overset{P \text{ orthogonal}}{=}} (P \Lambda P^{-1})^{-1} = P \Lambda^{-1} P^{-1} \underset{P^{-1} = P^T}{=\!=} P \Lambda^{-1} P^T$$

* Assum the hypothesis is true for $i = -k$, which means we have $A^{-k} = P \Lambda^{-k} P^T$

(k is a positive integer)

* Then when $i = -(k+1)$, we have

$$A^{-(k+1)} = [A^{(k+1)}]^{-1} = (A^k A)^{-1} = A^{-1}(A^k)^{-1} = A^{-1} A^{-k} = (P \Lambda^{-1} P^T)(P \Lambda^{-k} P^T) =$$

$$= P \Lambda^{-1} \underbrace{(P^T P)}_{=I} \Lambda^{-k} P^T = P \Lambda^{-1} \Lambda^{-k} P^T = P \Lambda^{-(k+1)} P^T$$

This So by induction ⇒ the hypothesis is true for all negative integers.

→ all integers. □

what about $k = 0$?

3/4

Q4 7 J&W 2.8, 2.9 (Don't use computer)

2.8 Given $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$ a) Find eigenvalues λ_1 and λ_2, eigenvectors $\vec{e}_1$ and $\vec{e}_2$
b) Determine the spectral decomposition of A

a) • $\begin{cases} \lambda_1 + \lambda_2 = \text{trace } A = -1 \\ \lambda_1 \lambda_2 = \det A = -6 \end{cases} \Rightarrow \begin{cases} \lambda_1 = 2 \\ \lambda_2 = -3 \end{cases}$

• Solve $(A - \lambda_1 I)\underset{\sim}{x}_1 = 0 \Leftrightarrow \begin{pmatrix} -1 & 2 \\ 2 & -4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} -x_1 + 2x_2 = 0 \\ 2x_1 - 4x_2 = 0 \end{cases}$

$\Rightarrow \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ is an associative eigenvector of $\lambda_1 = 2 \Rightarrow \vec{e}_1 = \begin{pmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{pmatrix}$

• Solve $(A - \lambda_2 I)\underset{\sim}{x}_2 = 0 \Leftrightarrow \underbrace{\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}}_{\text{non-singular}}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} 4x_1 + 2x_2 = 0 \\ 2x_1 + 1x_2 = 0 \end{cases}$

$\Rightarrow \begin{pmatrix} 1 \\ -2 \end{pmatrix}$ is an associative eigenvector of $\lambda_2 = -3 \quad \vec{e}_2 = \begin{pmatrix} \frac{1}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} \end{pmatrix}$

b) Determine the spectral decomposition of A

$$A = P \Lambda P^T = \begin{pmatrix} 2/\sqrt{5} & 1/\sqrt{5} \\ 1/\sqrt{5} & -2/\sqrt{5} \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 0 & -3 \end{pmatrix}\begin{pmatrix} 2/\sqrt{5} & 1/\sqrt{5} \\ 1/\sqrt{5} & -2/\sqrt{5} \end{pmatrix}$$

$$\ominus\ 2\begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}\begin{bmatrix} 2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix} - 3\begin{bmatrix} 1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix}\begin{bmatrix} 1/\sqrt{5} & -2/\sqrt{5} \end{bmatrix}.$$

$$= 2\begin{bmatrix} 4/5 & 2/5 \\ 2/5 & 1/5 \end{bmatrix} - 3\begin{bmatrix} 1/5 & -2/5 \\ -2/5 & 4/5 \end{bmatrix} \qquad \square 2.8.$$

4/4

2.9

Let $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$ as in exercise 2.8

a) Find A^{-1}

b) Compute the eigenvalues and eigenvectors of A^{-1}

c) Write the spectral decomposition of A^{-1}. Compare it with that of A from ex 2.8.

a) $A^{-1} = \frac{1}{\det A}\begin{pmatrix} -2 & -2 \\ -2 & 1 \end{pmatrix} = -\frac{1}{6}\begin{pmatrix} -2 & -2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} 1/3 & 1/3 \\ 1/3 & -1/6 \end{pmatrix}$ symmetric → eigenvalues real; non-singular → eigenvalues $\in \mathbb{R}$.

b) eigenvalues $\begin{cases} \lambda_1 \lambda_2 = -\frac{1}{6} \\ \lambda_1 + \lambda_2 = \frac{1}{3} - \frac{1}{6} = \frac{1}{6} \end{cases} \rightarrow \begin{cases} \lambda_1 = \frac{1}{2} \\ \lambda_2 = -\frac{1}{3} \end{cases}$

- Eigenvectors associative with $\lambda_1 = \frac{1}{2}$

$$\left(A - \lambda_1 I\right)\underset{\sim}{x} = \underset{\sim}{0} \Leftrightarrow \begin{pmatrix} 1/3 - 1/2 & 1/3 \\ 1/3 & -1/6 - 1/2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} -1/6 & 1/3 \\ 1/3 & -2/3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{row } 2 = 2 \text{ row } 1.$$

$\Rightarrow \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ is an associative eigenvector of $\lambda_1 = 1/2$ choose $e_1 = \begin{pmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{pmatrix}$

- eigenvector associated with $\lambda_2 = -1/3$

$$(A - \lambda_2 I)x = 0 \Rightarrow \begin{pmatrix} 2/3 & 1/3 \\ 1/3 & 1/6 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{row } 1 = 2 \text{ row } 2$$

$\begin{pmatrix} 1 \\ -2 \end{pmatrix}$ is an associative eigenvector of $\lambda_2 = -1/3$ choose $e_2 = \begin{pmatrix} 2/\sqrt{5} \\ -2/\sqrt{5} \end{pmatrix}$

4/4

* Observe:

$$\text{eigenvalues of } A^{-1} = \frac{1}{\text{eigenvalues of } A}$$

eigenvectors of A^{-1} = eigenvector of A.

c) $A^{-1} = \sum_{i=1}^{2} \lambda_i e_i e_i^T = \frac{1}{2}\begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}\begin{bmatrix} 2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix}\begin{bmatrix} 1/\sqrt{5} & -2/\sqrt{5} \end{bmatrix}$

$$= \frac{1}{2}\begin{bmatrix} 4/5 & 2/5 \\ 2/5 & 1/5 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1/5 & -2/5 \\ -2/5 & 4/5 \end{bmatrix}$$

$\Rightarrow$ If we have $A = P \Lambda P^T$ then $A^{-1} = P \Lambda^{-1} P^T$

where Λ is a diagonal matrix. we have when $\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix}$ then $\Lambda^{-1} = \begin{bmatrix} 1/\lambda_1 & & \\ & 1/\lambda_2 & \\ & & \ddots \end{bmatrix}$

Q5 J & W 2.12, 2.13

2.12 Show that the determinant of a square symmetric matrix $p \times p$ A can be expressed as the product of its eigenvalue $\lambda_1, \lambda_2, \ldots, \lambda_p$ that is $\det(A) = \prod_{i=1}^{p} \lambda_i$ (the result is also true when A is not symmetric) ← prove by using eigenvalue decomposition (A diagonalizable)

2.13 Show that $\det Q = \pm 1$ if Q is a $p \times p$ orthogonal matrix

2.12 Since A non-singular + symmetric, by Spectral decomposition theorem,
$A = P \Lambda P^T$, where P orthogonal matrix
Λ: diagonal matrix contain p eigenvalues of A.

$\Rightarrow \det(A) = \det(P \Lambda P^T) = \det(P)\det(\Lambda P^T) = \det(P)\det(\Lambda)\det(P^T) =$
$= \det(P)\det(P^T)\det(\Lambda) = \det(P P^T)\det(\Lambda) = \det(I)\det(\Lambda)$
$= \det(\Lambda) = \prod_{i=1}^{p} \lambda_i$ since Λ is a diagonal matrix → $\det \Lambda$ is the product of all entries in the diagonal which are all eigenvalues of A. □

2.13. Show that if Q is an orthogonal matrix $\not\Rightarrow$ $\det Q = 1$ or $\det Q = -1$.

- Q is orthogonal $\Leftrightarrow Q Q^T = I$

$\Rightarrow 1 = \det(Q Q^T) = |Q Q^T| = |Q||Q^T| = |Q||Q| = |Q|^2 \Rightarrow |Q| = \pm 1$.
($|AB| = |A||B|$; $|A^T| = |A|$)

- Observe: the converge is not true.

ex $Q = \begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix}$ true.

4/4

Q6, 2.15

A quadratic form $x'Ax$ is said to be positive definite $\overset{\text{def}}{\iff}$ A is positive definite

Is the quadratic form $3x_1^2 + 3x_2^2 - 2x_1x_2$ positive definite ?

We have

$$3x_1^2 + 3x_2^2 - 2x_1x_2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underset{\sim}{x}' A \underset{\sim}{x} \quad \text{where } A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$$

Consider A, we have $A_{(1)} = 3$ $\det(A_{(1)}) > 0$

$A_{(2)} = A$ $\det(A_{(2)}) = 9 - 1 = 8 > 0$

$\Rightarrow$ A is positive definite $\iff$ the quadratic form is positive definite $\square$

$\Rightarrow$ Yes !

Q7 2.16

Consider an arbitrary $n \times p$ matrix A. Then $A'A$ is a symmetric $p \times p$ matrix

Show that $A'A$ is necessarily non negative definite.

Since $A'A$ is symmetric, $x'A'Ax$ is well defined and we want to prove that

$$\underset{\sim}{x}' A'A \underset{\sim}{x} \geq 0, \quad \forall \underset{\sim}{x} \neq \underset{\sim}{0}$$

Put $y := Ax$ then

$$\left.\begin{aligned} y'y = (Ax)'Ax &= x'A'Ax \\ \text{Since } y'y = \sum_{i=1}^{n} y_i^2 &\geq 0 \end{aligned}\right\} \Rightarrow x'A'Ax \geq 0.$$

equality happens when $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ good detail.

$\updownarrow$

when x is a solution of $Ax = 0$

$\Rightarrow$ $A'A$ is is positive semi definite.

4/4

MATH 755 MULTIVARIATE STATISTICAL ANALYSIS

Assignment 1 — Due on in class Thursday September 20th

Let $\mathbf{j} = \mathbf{1}$ denote the column vector of ones, and $\mathbf{J} = \mathbf{j}\mathbf{j}^T = \mathbf{1}\mathbf{1}^T$ be the square matrix of ones. All constants and random variables are taken to be real valued. The population variance σ^2 is taken to be positive ($\sigma^2 > 0$).

Q 1. Consider the sample variance S^2 for a random sample $X_1, \ldots, X_n \overset{\text{iid}}{\sim} (\mu, \sigma^2)$.

(a) Write the sample variance as a quadratic form

(b) Using the appropriate theorems show that it is unbiased for σ^2, namely show that $E(S^2) = \sigma^2$.

Q 2. Let $\mathbf{X}$ be a p-by-1 random vector with variance-covariance matrix $\mathbf{\Sigma}$.

(a) Explain why $\mathbf{\Sigma}$ must be positive-semi-definite. (What would it imply if $\mathbf{\Sigma}$ wasn't positive?)

(b) If $\mathbf{\Sigma}$ is positive-semi-definite, but not positive-definite, what does this say about $\mathbf{X}$? (Consider $\mathbf{c}^T\mathbf{X}$ for some constant column vector $\mathbf{c}$.)

Q 3. Let $\mathbf{A}$ be a p-by-p positive definite matrix.

(a) Show that there exists a p-by-1 random vector $\mathbf{X}$ such that $\mathbf{\Sigma_X} = \mathbf{A}$. (You may assume that there exists a random vector $\mathbf{Z}$ where $\mathbf{Z} = (Z_1, \ldots, Z_p)$ and $Z_1, \ldots, Z_p \overset{\text{iid}}{\sim} (0, 1)$.)

(b) Find a transformation of $\mathbf{X}$ such that the variance-covariance matrix of this transformed variable $\mathbf{Y} = \mathbf{C}\mathbf{X}$ is the same as the correlation matrix of $\mathbf{X}$ ($\mathbf{\Sigma_Y} = \rho_{\mathbf{X}}$). Comment on the relationship of $\mathbf{Y}$ to $\mathbf{X}$.

Q 4. Suppose that $\mathbf{Y} \sim (\mu, \mathbf{\Sigma})$ is a 3-by-1 random vector where $\mu = \mu\boldsymbol{j}$ and for some ρ,

$$\mathbf{\Sigma_Y} = \sigma^2 \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{bmatrix}$$

(a) Write $\mathbf{\Sigma}$ as a linear combination of $\mathbf{I}$ and $\mathbf{J}$.

(b) For what values of ρ is $\mathbf{\Sigma}$ a valid variance-covariance matrix..

(c) Compute the expectation $E\left[\sum_{i=1}^{3}(Y_i - \bar{Y})^2\right]$. Comment on this as a function of ρ.

Q 5. Let $\mathbf{X}$ be a 3-by-1 random vector with variance covariance matrix

$$\mathbf{\Sigma_X} = \begin{bmatrix} 5 & -4 & 1 \\ -4 & 13 & 9 \\ 1 & 9 & 10 \end{bmatrix}$$

(a) Find a unit vector $\mathbf{c} = (c_1, c_2, c_3)^T$ such that the variance of $\mathbf{c}^T\mathbf{X}$ is maximized/minimized. (You may use a computer for convenience)

(b) What is the largest/smallest possible variance for such linear combinations?

Q 6. Johnson and Wichern 2.41

Q 7. Let $\mathbf{X} \sim (\mu, \mathbf{\Sigma})$. Without using a computer compute

(a) $|\mathbf{\Sigma}|$

(b) $tr\boldsymbol{\Sigma}$

(c) $V(\mathbf{e}_i^T\mathbf{X}), i = 1, 2, 3$

(d) $\boldsymbol{\Sigma}$

(e) $\boldsymbol{\Sigma}^{1/2}$

where $\lambda_1 = 3$, $\lambda_2 = 7$, and $\lambda_3 = 1$ are the eigenvalues of $\boldsymbol{\Sigma}$, and $\mathbf{e}_1^T = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]$, $\mathbf{e}_2^T = [2/\sqrt{6}, -1/\sqrt{6}, -1/\sqrt{6}]$, and $\mathbf{e}_3^T = [0, 1/\sqrt{2}, -1/\sqrt{2}]$ are the associated eigenvectors.

Q 8. Let $\mathbf{X} \sim N_3(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}^T = [1, 0, -2]$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} 2 & -0.5 & 1 \\ -0.5 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

(a) Which components of $\mathbf{X}$ are independent?

(b) Find the conditional distribution of $X_2|(X_1, X_3)^T$

(c) Find the conditional distribution of $(X_1, X_3)^T|X_2$

(d) Comment on how your answer to part (a) affects parts (b) and (c).

Q 9. Using the data in Table 4.6 from the book (T4-6), check for univariate and multivariate normality for the females. Comment.

Q 10. Let $\mathbf{X}_1 \sim N_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $\mathbf{X}_2 \sim N_p(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$. Given independent random samples of sizes n_1 and n_2 respectively, obtain the MLE's of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{\Sigma}$.

Q 11. Suppose $\mathbf{X}_1, \ldots, \mathbf{X}_n \overset{iid}{\sim} N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

(a) Find the mle of k- a scalar- where $\boldsymbol{\Sigma} = k\boldsymbol{\Sigma}_0$ and both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_0$ are known.

(b) Find the mle of k- a scalar- where $\boldsymbol{\mu} = k\boldsymbol{\mu}_0$ and both $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}$ are known.

Q1)

Consider the sample variance S^2 for a random sample $X_1, \dots, X_n \overset{iid}{\sim} (\mu, \sigma^2)$

a) Write the sample variance as a quadratic form

b) Use the appropriate theorem to show that it is unbiased for σ^2; $E(S^2) = \sigma^2$.

a) We have

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 = \frac{1}{(n-1)}\begin{bmatrix} X_1 - \bar{X} & X_2 - \bar{X} & \dots & X_n - \bar{X} \end{bmatrix}\begin{bmatrix} X_1 - \bar{X} \\ X_2 - \bar{X} \\ \vdots \\ X_n - \bar{X} \end{bmatrix}$$

- Consider

$$\begin{bmatrix} X_1 - \bar{X} \\ X_2 - \bar{X} \\ \vdots \\ X_n - \bar{X} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} - \begin{bmatrix} \bar{X} \\ \bar{X} \\ \vdots \\ \bar{X} \end{bmatrix} = \underset{\sim}{X} - \frac{1}{n}\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \underset{\sim}{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\underset{\sim}{X} = I\underset{\sim}{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\underset{\sim}{X} =$$

$$= \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\underset{\sim}{X}, \text{ where } \underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

- Then $S^2 = \frac{1}{(n-1)}\left[\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\underset{\sim}{X}\right]^T\left[\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\underset{\sim}{X}\right]$

$$= \frac{1}{(n-1)}\left[\underset{\sim}{X}^T\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)^T\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\underset{\sim}{X}\right] =$$

- note that

$$\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)^T\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right) = I^2 + \frac{1}{n^2}(\mathbf{1}\mathbf{1}^T)^T(\mathbf{1}\mathbf{1}^T) - \frac{2}{n}I\,\mathbf{1}\mathbf{1}^T = I^2 + \frac{1}{n^2}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{1}}_{n}\mathbf{1}^T - \frac{2}{n}\mathbf{1}\mathbf{1}^T$$

$$= I^2 + \frac{1}{n}\mathbf{1}\mathbf{1}^T - \frac{2}{n}\mathbf{1}\mathbf{1}^T = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \overset{\text{put}}{=\!=} K$$

Then $S^2 = \frac{1}{(n-1)}\left(\underset{\sim}{X}^T K \underset{\sim}{X}\right)$

where $K^T = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)^T = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T = K \Rightarrow K$ symmetric

→ The quadratic form for S^2 is $S^2 = \frac{1}{(n-1)}\left(\underset{\sim}{X}^T K \underset{\sim}{X}\right)$

b) Use the appropriate theorem to show that $E(S^2) = \sigma^2$.

$$E(S^2) = \frac{1}{(n-1)}E\left(\underset{\sim}{X}^T K \underset{\sim}{X}\right) = \frac{1}{n-1}\left\{tr(K\Sigma_x) + \mu^T K \mu\right\} = \frac{1}{n-1}\left\{tr\left[\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\sigma I\right] + \mu^T\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\mu\right\}$$

$$= \frac{1}{n-1}\left\{\left[tr\left[\underset{\text{symmetric}}{I\sigma I}\right) - tr\left(\frac{1}{n}\mathbf{1}\mathbf{1}^T\sigma I\right)\right] + \mu^T I \mu - \frac{1}{n}\mu^T\underbrace{\mathbf{1}\mathbf{1}^T}_{n}\mu\right\}$$

$$= \frac{1}{n-1}\left\{\left[n\sigma - \frac{1}{n}tr\begin{bmatrix} \sigma & \sigma & \dots & \sigma \\ \sigma & \sigma & \dots & \sigma \\ \sigma & \sigma & \dots & \sigma \end{bmatrix}\right]\right\} + \underbrace{\mu^T\mu - \mu^T\mu}_{=0}$$

$$= \frac{1}{n-1}\left\{n\sigma - \frac{1}{n}n\sigma\right\} = \frac{1}{n-1}(n-1)\sigma = \sigma \quad \square$$

Q2 → Let $X_{p\times 1}$: random vector with covariance - covariance matrix Σ.

a) Explain why Σ must be positive semidefinite (what would imply if Σ wasn't positive?)

b) If Σ is positive-semidefinite, but not positive definite, what does this say about X?

(Hint: consider $c^T X$ for some constant column vector c)

a) First, we want to prove that Σ is positive semi definite.

We have $\Sigma_X = E((X-\mu_X)(X-\mu_X)^T)$. Then for any $u_{p\times 1} \neq 0$, we want to prove $u^T \Sigma_X u \geq 0$.

$u^T \Sigma_X u = u^T E((X-\mu_X)(X-\mu_X)^T) u = E([u^T(X-\mu_X)][(X-\mu_X)^T u]) = E(y^T y) = \sum_{i=1}^{p} y_i^2$

Then Σ_X is positive-semi-definite (y^T and y put = y ← real value random variable)

* Explain why Σ must be positive-semi-definite

If Σ wasn't positive $\iff \exists c_{p\times 1}$, a constant vector $c^T \Sigma c < 0$.

However, $c^T \Sigma c = c^T V(X) c = V(c^T X) = V(Y)$ where $Y = c^T X$

↑ can't be negative ⇒ (contradiction).

b) If Σ is positive-semi-definite, but not positive definite, then.

$= c^T \Sigma c = c^T (V(X)) c = V(c^T X) = V(Y)$, for $c_{p\times 1} \neq 0$

This means $V(c^T X) = V(\sum_{i=1}^{n} c_i X_i) = 0$

This happens when $\exists i$ so that $c_i \neq 0$ so that $\sum_{i=1}^{n} c_i X_i = 0$ ⇒ there are some X_i are not linearly independent.

• If all X_i are linearly independent, $V(\sum_{i=1}^{n} c_i X_i) = 0 \iff \sum_{i=1}^{n} c_i V(X_i) = 0$, this can be happended when $\exists i$ so that $V(X_i) = 0$

To sum up, $X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}$ can [have some $\{X_i\}$ are not linearly independent. have one / some X_i so that $V(X_i) = 0$ □.

Q3

Let $A_{p\times p}$ positive definite matrix.

a) Show that there exists $X_{p\times 1}$ random vector so that $\Sigma_X = A$

(You may assume that there exists a random vector $\underset{\sim}{Z} = \begin{bmatrix} z_1 \\ \vdots \\ z_p \end{bmatrix} \sim N(0, I)$)

b) Find a transformation of X such that the variance-covariance matrix of this transformed variable $Y = CX$ is the same as the correlation matrix of X ($\Sigma_Y = \rho_X$).

Comment on the relationship of Y to X

---

a) When A is a positive definite matrix then we have there exist a unique positive def matrix L such that $L^2 = A$, $L = A^{1/2} = P \cdot \Lambda^{-1/2} P^T$

(Remind: P: orthogonal matrix contains eigenvectors of A $\quad A = P\Lambda P$: spectral decomposition of A

Λ: diagonal matrix, contains eigenvalues of A

Then for $Z \sim N_p(0, I)$

Put $\underset{p\times 1}{X} = \underset{p\times 1}{\mu} + \underset{p\times p}{L}\,\underset{p\times 1}{Z} \overset{\text{theorem}}{\sim} N_p(\mu, L L^T) = N_p(\mu, L^2) = N_p(\mu, A)$ □a.

L is positive def $\Rightarrow$ symmetric.

b) Find $Y = CX$ so that $\Sigma_Y = \rho_X$

- Note that when $Y = CX$,
  then $\Sigma_Y = V(Y) = V(CX) = C\,V(X)\,C^T = C\,\Sigma_X\,C^T$
- Note that when $V^{1/2} = \begin{pmatrix} \sqrt{\sigma_{11}} & & \\ & \ddots & \\ & & \sqrt{\sigma_{pp}} \end{pmatrix}$ then $\Sigma_X = V^{1/2}\,\rho_X\,V^{1/2} \Leftrightarrow \rho_X = V^{-1/2}\,\Sigma\,V^{-1/2}$

($V^{1/2}$: standard deviation matrix of X)

Then let $C := V^{-1/2}$

then $\Sigma_Y = C\,\Sigma_X\,C^T = V^{-1/2}\Sigma_X(V^{-1/2})^T = V^{-1/2}\Sigma_X V^{-1/2} = \rho_X$

(V is diagonal $\Rightarrow$ symmetric.)

Then $Y = V^{-1/2}X$

Comment on the relationship of Y to X.

$Y = V^{-1/2}X$ then $\Sigma_Y = \rho_X$

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_p \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{\sigma_{11}}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\sigma_{pp}}} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_p \end{bmatrix} \Leftrightarrow Y_i = \frac{1}{\sqrt{\sigma_{ii}}} X_i, \quad \forall i = 1, p$$

Q4) Suppose $Y \sim \mathcal{N}(\mu, \Sigma)$ $\mu = \begin{bmatrix} \mu \\ \mu \\ \mu \end{bmatrix} = \mu \mathbb{1}$ $\Sigma_Y = \sigma^2 \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{bmatrix}$.

a) Write Σ_Y as a linear combination of I and J

b) For what value of ρ is Σ a valid variance-covariance matrix

c) Compute the expectation $E\left(\sum_{i=1}^{3}(Y_i - \bar{Y})^2\right)$. Comment on this as a function of ρ.

a) $$\Sigma_Y = \sigma^2 \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{bmatrix} = \sigma^2 \left\{ (1-\rho) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \rho \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right\} =$$

$$= \sigma^2 \left\{ (1-\rho) I + \rho \mathbb{1}\mathbb{1}^T \right\} = \sigma^2 \left\{ (1-\rho) I + \rho J \right\}$$

b) By Sylvester's criterion

Σ_Y is a variance-covariance matrix

$\Leftrightarrow \Sigma_Y$ is positive-semi-positive $\Leftrightarrow$ $\begin{cases} \Sigma_Y \text{ is symmetric} \\ \text{all leading } m \times m \ (m = \overline{1,p}) \ \Sigma_{n \times m} \text{ have } \det(\Sigma_{m \times m}) \geq 0 \end{cases}$

- $\det(\Sigma_{11}) = \det[\sigma^2] = \sigma^2 > 0$
- $\det(\Sigma_{22}) = \det\left[\sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right] = (\sigma^2)^2 \det \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} = \sigma^4 (1 - \rho^2)$
- $\det(\Sigma_{33}) = (\sigma^2)^3 \det \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix} = \sigma^6 \det \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ 1+2\rho & 1+2\rho & 1+2\rho \end{pmatrix} = \sigma^6 \det \begin{pmatrix} 1-\rho & 0 & \rho \\ 0 & 1-\rho & \rho \\ 0 & 0 & 1+2\rho \end{pmatrix} = \sigma^6 (1-\rho)^2 (1+2\rho)$

So we need $\begin{cases} 1 - \rho^2 \geq 0 \\ (1-\rho)(1+2\rho) \geq 0 \end{cases} \Leftrightarrow \begin{cases} -1 \leq \rho \leq 1 \\ -\frac{1}{2} \leq \rho \leq 1 \end{cases} \rightarrow -\frac{1}{2} \leq \rho \leq 1$

c) Compute $E\left(\sum_{i=1}^{3}(X_i-\bar{X})^2\right)$. Comment on this as a function of ρ

From problem 1) $\sum_{i=1}^{3}(Y_i-\bar{Y})^2 = \underset{\sim}{Y}^T K \underset{\sim}{Y}$, where $K = I - \frac{1}{n}\mathbb{1}\mathbb{1}^T$

Note that K is symmetric & $K^2 = K$, $\underset{\sim}{M}^T K \underset{\sim}{M} = 0$ when $\underset{\sim}{M} = \begin{pmatrix} M \\ \vdots \\ M \end{pmatrix} = M\mathbb{1}$

Then $E\left(\sum_{i=1}^{3}(Y_i-\bar{Y})^2\right) = E(Y^TKY) = tr(K\Sigma_Y) + \underset{\sim}{M}^T K \underset{\sim}{M}$

$= tr\left\{\left(I-\frac{1}{n}\mathbb{1}\mathbb{1}^T\right)\left(\sigma^2[(1-\rho)I + \rho\mathbb{1}\mathbb{1}^T]\right)\right\} + \underbrace{\underset{\sim}{M}^T\left(I-\frac{1}{n}\mathbb{1}\mathbb{1}^T\right)\underset{\sim}{M}}_{=0 \text{ (similar to problem 1)}}$

$= tr\left(I\sigma^2(1-\rho)I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\sigma^2(1-\rho)I + \rho I\mathbb{1}\mathbb{1}^T - \frac{\rho}{n}\mathbb{1}\mathbb{1}^T\mathbb{1}\mathbb{1}^T\right)$

$= n\sigma^2(1-\rho) - \frac{1}{n}\sigma^2(1-\rho)\,tr\left[\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}\begin{pmatrix}1&&\\&1&\\&&1\end{pmatrix}\right] + \rho\,tr\left[\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}\right] - \frac{\rho}{n}tr\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}$

$= n\sigma^2(1-\rho) - \sigma^2(1-\rho) + n\rho - n\rho$

$= (n-1)\sigma^2(1-\rho)$

$\underset{n=3}{=} 2\sigma^2(1-\rho)$

* Comment:

$E\left(\sum_{i=1}^{3}(Y_i-\bar{Y})^2\right) = 2\sigma^2(1-\rho) \geq 0$ (since $-\frac{1}{2} \leq \rho \leq 1$).

Q5 7

Let X be a 3×1 random vector, $\Sigma_X = \begin{bmatrix} 5 & -4 & 1 \\ -4 & 13 & 9 \\ 1 & 9 & 10 \end{bmatrix}$

a) Find a unit vector $\underset{\sim}{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$ s.t $V(c^T X)$ is maximized / minimized.
(May use a computer for convenience)

b) What is the largest / smallest possible variance for such linear combinations?

a) We have $V(\underset{\sim}{c}^T X) = \underset{\sim}{c}^T V(X) \underset{\sim}{c} = c^T \Sigma_X \underset{\sim}{c}$

we want $V(\underset{\sim}{c}^T X)$ to be maximized / minimized $\Rightarrow$

$\rightarrow$ we want to find $\underset{\sim}{c}$ associative with maximum eigenvalue and minimum eigenvalue

* The two commands

A = matrix(c(5,-4,1,-4,13,9,1,9,10), ncol=3, nrow=3, byrow=TRUE))

eigen(A)

gives us three eigenvalues: $2.1e+01$ ($=21$), $7e+00$, $2.13e-14$ ($=0$).

$\underset{\sim}{c}_{I} = \begin{bmatrix} 0.1543 \\ -0.7715 \\ -0.6172 \end{bmatrix}$ $\qquad$ $\underset{\sim}{c}_{II} = \begin{bmatrix} -0.5773 \\ 0.5773 \\ -0.5773 \end{bmatrix}$

are the two unit vectors when $V(c^T X)$ is maximized $\qquad$ $V(c^T X)$ is minimized.

b) The largest possible variance $\underset{\sim}{c}_{I}^T \Sigma_X \underset{\sim}{c}_{I} = 21$

The smallest possible variance $\underset{\sim}{c}_{II}^T \Sigma_X \underset{\sim}{c}_{II} = 0$

since $\underset{\sim}{c}^T \Sigma \underset{\sim}{c} = \begin{bmatrix} \lambda_1^{max} & & \\ & \lambda_2 & \\ & & \lambda_3^{min} \end{bmatrix}$.

Q6 Johnson and Wichern 2.4 L.

Let $\underset{\sim}{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$ $\underset{\sim}{\mu}_X = \begin{bmatrix} 3 \\ 2 \\ -2 \\ 0 \end{bmatrix}$ $\Sigma_X = \begin{bmatrix} 3 & & 0 & \\ & 3 & & \\ 0 & & 3 & \\ & & & 3 \end{bmatrix}$ Let $A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix}$.

a) Find $E(A\underset{\sim}{X})$ b) Find $Cov(AX)$

c) Which pairs of linear combinations have zero covariances.

a) $E(A\underset{\sim}{X}) = A\,E(X) = A\mu_X = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ -2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \\ 3 \end{bmatrix}$.

$3\times4 \quad 4\times1 \quad 3\times1$

b) $Cov(AX) = A\,cov(X)\,A^T = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & 36 \end{bmatrix}$

$3\times4 \quad 4\times4 \quad 4\times3 \quad 3\times3$.

c) Which pairs of linear combinations have zero covariances.

We have from the $Cov(AX)$ that all pairs (there of them) ~~are~~ have zero covariances.

Q7)

Let $\underset{\sim}{X} \sim (\underset{\sim}{\mu}, \Sigma)$. Without using computer, compute

(where $\lambda_1 = 3, \lambda_2 = 7, \lambda_3 = 1$ are eigenvalues of Σ

$\underset{\sim}{e}_1 = \begin{pmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix}$ $\underset{\sim}{e}_2 = \begin{pmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ -1/\sqrt{6} \end{pmatrix}$ $\underset{\sim}{e}_3 = \begin{pmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$ are associated eigenvectors.

compute a) $|\Sigma|$ b) $tr(\Sigma)$ c) $V(\underset{\sim}{e}_i^T X)$, $i = 1,2,3$ d) Σ e) $\Sigma^{1/2}$.

a) $|\Sigma| = \prod_{i=1}^{3} \lambda_i = 3 \cdot 7 \cdot 1 = 21$

b) $tr(\Sigma) = 3 + 7 + 1 = 11$

c) $\forall i = 1,2,3$

$V(\underset{\sim}{e}_i^T X) = \underset{\sim}{e}_i^T V(X) \underset{\sim}{e}_i = \underset{\sim}{e}_i^T \Sigma \underset{\sim}{e}_i = \underset{\sim}{e}_i^T P \Lambda P^T \underset{\sim}{e}_i = \underset{\sim}{e}_i^T [\underset{\sim}{e}_1 | \underset{\sim}{e}_2 | \underset{\sim}{e}_3]$

$= \underset{\sim}{e}_i^T [\underset{\sim}{e}_1 | \underset{\sim}{e}_2 | \underset{\sim}{e}_3] \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \begin{bmatrix} \underset{\sim}{e}_1^T \\ \underset{\sim}{e}_2^T \\ \underset{\sim}{e}_3^T \end{bmatrix} \underset{\sim}{e}_i$

Note that $\underset{\sim}{e}_i^T \underset{\sim}{e}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$

$\Rightarrow V(\underset{\sim}{e}_i^T X) = \lambda_i$, $\forall i = 1,3$.

- For example:

$V(\underset{\sim}{e}_1^T X) = [\underset{\sim}{e}_1^T [\underset{\sim}{e}_1 | \underset{\sim}{e}_2 | \underset{\sim}{e}_3] \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \begin{bmatrix} \underset{\sim}{e}_1^T \\ \underset{\sim}{e}_2^T \\ \underset{\sim}{e}_3^T \end{bmatrix} \underset{\sim}{e}_1 = [1 | 0 | 0] \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} =$

- Similarly for case $i = 2,3$. $\Rightarrow$ done.

d) Compute Σ

$\Sigma = P \Lambda P^T = \sum_{i=1}^{3} \lambda_i \underset{\sim}{e}_i \underset{\sim}{e}_i^T = 3 \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix} [1/\sqrt{3} \;\; 1/\sqrt{3} \;\; 1/\sqrt{3}] + 7 \begin{bmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ -1/\sqrt{6} \end{bmatrix} [2/\sqrt{6} \;\; -1/\sqrt{6} \;\; -1/\sqrt{6}] +$

$+ 1 \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix} [0 \;\; 1/\sqrt{2} \;\; -1/\sqrt{2}]$.

$= 3 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} + 7 \begin{bmatrix} 4/6 & -2/6 & -2/6 \\ -2/6 & 1/6 & 1/6 \\ -2/6 & 1/6 & 1/6 \end{bmatrix} + 1 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/2 & -1/2 \\ 0 & -1/2 & 1/2 \end{bmatrix} = \begin{bmatrix} 17/3 & -4/3 & -4/3 \\ -4/3 & 8/3 & 5/3 \\ -4/3 & 5/3 & 8/3 \end{bmatrix}$

$= \frac{1}{3} \begin{bmatrix} 17 & -4 & -4 \\ -4 & 8 & 5 \\ -4 & 5 & 8 \end{bmatrix}$

$$\Sigma^{1/2} = P \cdot \Lambda^{1/2} P^T = \sum_{i=1}^{3} \sqrt{\lambda_i}\, e_i e_i^T$$

$$= \sqrt{3}\begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}\begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} + \sqrt{7}\begin{bmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ -1/\sqrt{6} \end{bmatrix}\begin{bmatrix} 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \end{bmatrix} + 1\begin{bmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}\begin{bmatrix} 0 & 1/\sqrt{2} & \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} + \begin{bmatrix} \frac{4\sqrt{7}}{6} & \frac{-2\sqrt{7}}{6} & \frac{-2\sqrt{7}}{6} \\ \frac{-2\sqrt{7}}{6} & \frac{\sqrt{7}}{6} & \frac{\sqrt{7}}{6} \\ \frac{-2\sqrt{7}}{6} & \frac{\sqrt{7}}{6} & \frac{\sqrt{7}}{6} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$= \frac{1}{6}\begin{bmatrix} 2\sqrt{3}+4\sqrt{7} & 2\sqrt{3}-2\sqrt{7} & 2\sqrt{3}-2\sqrt{7} \\ 2\sqrt{3}-2\sqrt{7} & 2\sqrt{3}+\sqrt{7}+3 & 2\sqrt{3}+\sqrt{7}-3 \\ 2\sqrt{3}-2\sqrt{7} & 2\sqrt{3}+\sqrt{7}-3 & 2\sqrt{3}+\sqrt{7}+3 \end{bmatrix} \square.$$

Q8) Let $\underset{\sim}{X} \sim N_3(\mu, \Sigma)$ $\mu = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 2 & -0.5 & 1 \\ -0.5 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$

a) Which components of X are independent?

b) Find the conditional distribution of $X_2 \mid (X_1, X_3)^T$

c) Find the conditional distribution of $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix} \mid X_2$

d) Comment on how your answer to part a) affects parts b) and c)

a) Since $\Sigma = [\sigma_{ij}]_{ij}$ contains σ_{ij}, the population covariance of X_i and X_j

remember $\sigma_{23} = \sigma_{32} = 0 \Rightarrow X_2$ and X_3 are independent.

b) Find the conditional distribution $X_2 \mid \begin{bmatrix} X_1 \\ X_3 \end{bmatrix}^T$

We consider $\underset{\sim}{Y} = \begin{bmatrix} X_2 \\ \hline X_1 \\ X_3 \end{bmatrix} = \begin{bmatrix} Y_1 \\ \hline Y_2 \end{bmatrix}$ then we have $\mu_{\underset{\sim}{Y}} = \begin{bmatrix} \mu_2 \\ \hline \mu_1 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \hline 1 \\ -2 \end{bmatrix}$ $\Sigma_{\underset{\sim}{Y}} = \left[\begin{array}{c|cc} \sigma_{22} & \sigma_{21} & \sigma_{23} \\ \hline \sigma_{12} & \sigma_{11} & \sigma_{13} \\ \sigma_{32} & \sigma_{31} & \sigma_{33} \end{array}\right] = \left[\begin{array}{c|cc} 1 & -0.5 & \\ \hline -0.5 & 2 & \\ 0 & 1 & \end{array}\right]$ (Σ_{21}, Σ_{12}, Σ_{11}, Σ_{22})

Then $f_{X_2 \mid \begin{bmatrix} X_1 \\ X_3 \end{bmatrix}}$ and we also have $\mu_{\begin{bmatrix} X_1 \\ X_3 \end{bmatrix}} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $\Sigma_{\begin{bmatrix} X_1 \\ X_3 \end{bmatrix}} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

Then we have the conditional distribution of $X_2 \mid \begin{bmatrix} X_1 \\ X_3 \end{bmatrix}$ has normal distribution with

mean $\mu_{X_2 \mid \substack{X_1 \\ X_3}} = \mu_2 + \Sigma_{Y2} \Sigma_{ZZ}^{-1} \left[\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_3 \end{pmatrix} \right] =$

$= 0 + [-0.5 \quad 0] * \frac{1}{3} \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_3 + 2 \end{bmatrix} = \frac{1}{3} [-1 \quad 0.5] \begin{bmatrix} x_1 - 1 \\ x_3 + 2 \end{bmatrix} =$

$= -\frac{1}{3} x_1 + \frac{1}{6} x_3 + \frac{2}{3}$

variance $\Sigma_{X_2 \mid \begin{bmatrix} X_1 \\ X_3 \end{bmatrix}} = 1 - [-0.5 \quad 0] \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} = 1 - [-0.5 \quad 0] * \frac{1}{3} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}$

$= 1 - \frac{1}{3} [-1 \quad 0.5] \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} = 1 - \frac{1}{3}(0.5) = 1 - \frac{1}{6} = \frac{5}{6}$

So $X_2 \mid \begin{bmatrix} X_1 \\ X_3 \end{bmatrix} \sim N\left(-\frac{1}{3} x_1 + \frac{1}{6} x_3 + \frac{2}{3} \right) \frac{5}{6}$

c) Find the conditional distribution of $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}\Big|_{X_2}$.

Consider $\underline{X} = \begin{bmatrix} X_1 \\ X_3 \\ \hline X_2 \end{bmatrix}$ has $\mu_X = \begin{bmatrix} 1 \\ -2 \\ \hline 0 \end{bmatrix}$ $\Sigma_X = \left[\begin{array}{cc|c} \sigma_{11} & \sigma_{13} & \sigma_{12} \\ \sigma_{31} & \sigma_{33} & \sigma_{32} \\ \hline \sigma_{21} & \sigma_{23} & \sigma_{22} \end{array}\right] = \left[\begin{array}{cc|c} 2 & 1 & -0.5 \\ 1 & 2 & 0 \\ \hline -0.5 & 0 & 1 \end{array}\right]$

Then $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}\Big|_{X_2}$ has normal distribution with

$$\text{mean} = \begin{pmatrix} \mu_1 \\ \mu_3 \end{pmatrix} + \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} (1)^{-1} \left(x_2 - \mu_2 \right) = \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0 \end{pmatrix} (x_2 - 0) = \begin{pmatrix} 1 - 0.5x_2 \\ -2 \end{pmatrix}$$

$$\text{Variance} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} (1)^{-1} \begin{bmatrix} -0.5 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} 0.25 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.75 & 1 \\ 1 & 2 \end{bmatrix}$$

Then $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}\Big|_{X_2 = x_2} \sim \text{Normal}\left(\begin{pmatrix} 1 - 0.5x_2 \\ -2 \end{pmatrix}, \begin{bmatrix} 1.75 & 1 \\ 1 & 2 \end{bmatrix} \right)$

d) Comment on how the answer to part a ~~one~~ affects part b and c.

We have the result of part a is X_2 and X_3 are independent.

then from part b) the conditional distribution $X_2\Big|_{\begin{bmatrix} X_1 \\ X_3 \end{bmatrix}}$ is still depended on $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}$

from part c) $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}\Big|_{X_2 = x_2}$ is still depended on X_2

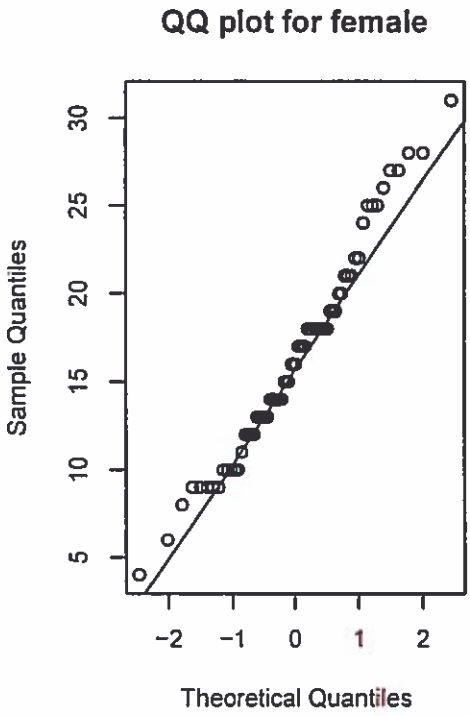
$\Rightarrow$ This means that even X_2 and X_3 are independent $X_2|_{X_3}$ is not depended on X_3

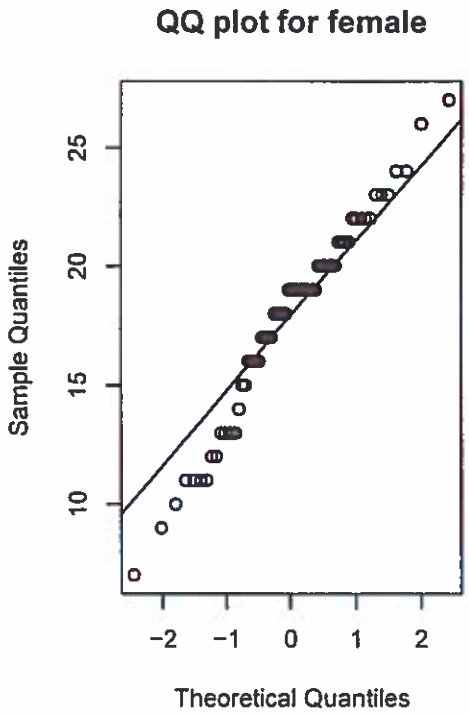
$X_3|_{X_2}$ is not dependeded on X_2.

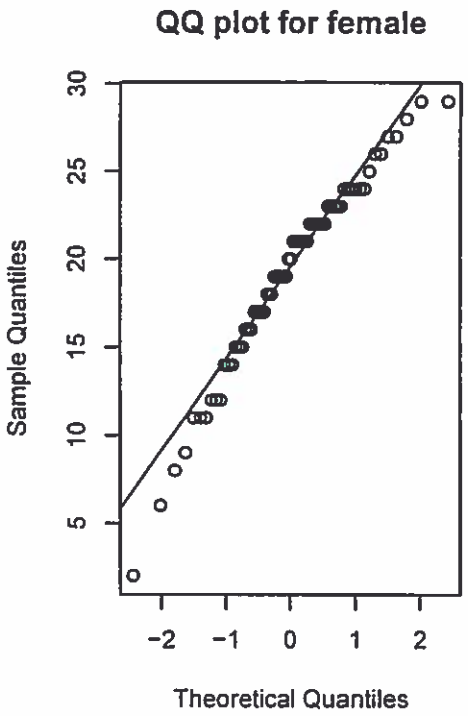
but when we consider $X_2\Big|_{\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}}$ ← vector created by X_1 and X_3 ← still depended on X_3.

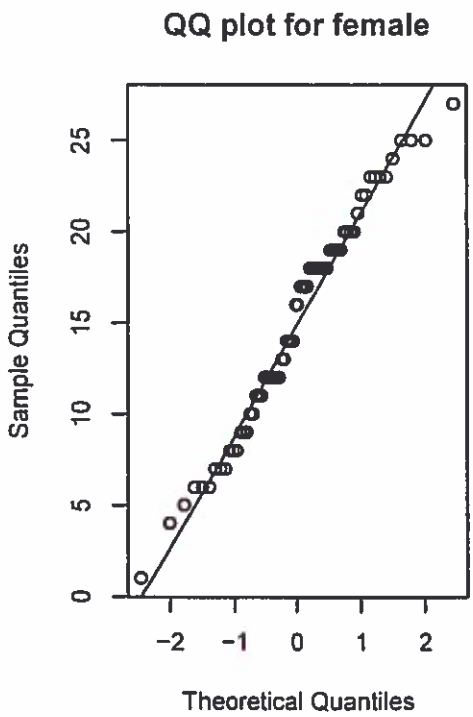
vector created by X_1 & X_3 $\longleftrightarrow$ $\begin{pmatrix} X_1 \\ X_3 \end{pmatrix}\Big|_{X_2}$ → is still depended on X_2.

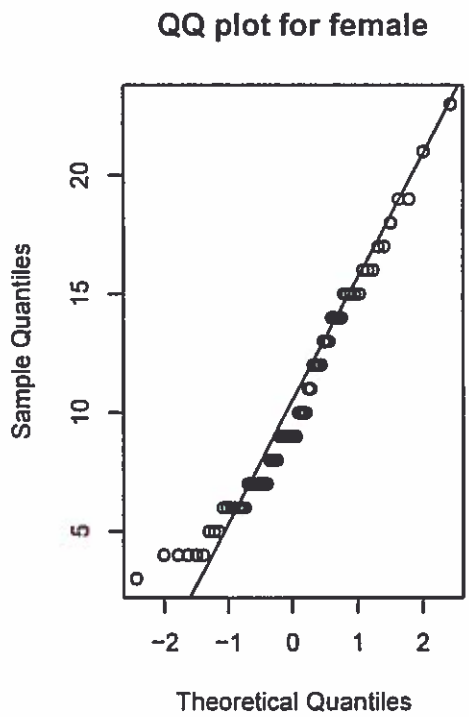
```
library(mvnTest)
setwd("C:/Users/tranl/Desktop/MAT755/Homework")
data<-read.table(file="C:/Users/tranl/Desktop/MAT755/Wichern_data_textbook/T4-6.DAT")
head(data)
#For each of teenagers, the gender(male=1, female=2)
#and socioeconomic status (low= 1, medium=2)
# The scores labeled independence (indep), support(supp), benevolence(benev)
#conformity(comform) and leadership (leader)

#Name the variables
names(data)<-c("Indep", "Supp", "Benev", "Conform", "Leader", "Gender", "Socio")
head(data)
#We want to check univariate and multivariate normalitry for female
is.female<-which(data$Gender==2)
data.female<-data[is.female, ]

dev.new(width=15, height=15)
par(mfrow=c(2,3))
for (i in c("Indep", "Supp", "Benev", "Conform", "Leader"))
{
 qqnorm(data.female[,i], main="QQ plot for female")
 qqline(data.female[,i])
}
```


QQ plot for female
Sample Quantiles
Theoretical Quantiles
5
10
15
20
25
30
−2
−1
0
1
2


QQ plot for female
Sample Quantiles
Theoretical Quantiles
10
15
20
25
−2
−1
0
1
2


QQ plot for female
Sample Quantiles
Theoretical Quantiles
5
10
15
20
25
30
−2
−1
0
1
2


QQ plot for female
Sample Quantiles
Theoretical Quantiles
0
5
10
15
20
25
−2
−1
0
1
2


QQ plot for female
Sample Quantiles
Theoretical Quantiles
5
10
15
20
−2
−1
0
1
2

Q10

Let $X_1 \sim N_p(\mu_1, \Sigma)$ Given independent random samples of size n_1 and n_2 respectively, $X_2 \sim N_p(\mu_2, \Sigma)$ obtain the MLE of μ_1, μ_2, Σ

* We have the likelihood is

$$L(\mu_1, \mu_2, \Sigma | X_{1,1}, \dots, X_{1,n_1}, X_{2,1}, \dots, X_{2,n_2}) = L(\mu_1, \Sigma | X_{1,1}, \dots, X_{1,n_1}) \; L(\mu_2, \Sigma | X_{2,1}, \dots, X_{2,n_2})$$

$$= \underbrace{\frac{1}{(2\pi)^{\frac{n_1 p}{2}} |\Sigma|^{\frac{n_1}{2}}} e^{-\frac{1}{2} \operatorname{tr}[\Sigma^{-1}(n_1-1)S_1] - \frac{n_1}{2}(\bar{X}_1 - \mu_1)^T \Sigma^{-1} (\bar{X}_1 - \mu_1)}}_{\text{constant}} \cdot \underbrace{\frac{1}{(2\pi)^{\frac{n_2 p}{2}} |\Sigma|^{\frac{n_2}{2}}} e^{-\frac{1}{2} \operatorname{tr}[\Sigma^{-1}(n_2-1)S_2] - \frac{n_2}{2}(\bar{X}_2 - \mu_2)^T \Sigma^{-1} (\bar{X}_2 - \mu_2)}}_{= (*)}$$

Since Σ is positive definite $\Rightarrow \Sigma^{-1}$ positive definite

$(\bar{X}_i - \mu_i)^T \Sigma^{-1} (\bar{X}_i - \mu_i) \geq 0$ minimum when $\mu_i = \bar{X}_i$

$\Rightarrow$ these terms are maximized when $\mu_1 = \bar{X}_1$ and $\mu_2 = \bar{X}_2$

So we have $\hat{\mu}_{1MLE} = \bar{X}_1 \quad \hat{\mu}_{2MLE} = \bar{X}_2$

* Now we want to maximize $(*)$

$$(*) = \frac{1}{|\Sigma|^{\frac{n_1+n_2}{2}}} e^{-\frac{1}{2} \operatorname{tr}\left\{\Sigma^{-1} \underbrace{[(n_1-1)S_1 + (n_2-1)S_2]}_{\text{put this equal } B}\right\}}$$

$$= \frac{1}{|\Sigma|^{\frac{n_1+n_2}{2}}} e^{-\frac{1}{2} \operatorname{tr}(\Sigma^{-1} B)} \leq \frac{1}{|B|^b} (2b)^{pb} e^{-bp} \quad \text{for all } \Sigma > 0$$

$b = \frac{n_1+n_2}{2}$

the equality happens when $\Sigma = \frac{1}{2b} B =$

$$= \frac{1}{2 \, \frac{n_1+n_2}{2}} [(n_1-1)S_1 + (n_2-1)S_2]$$

So we have $\hat{\Sigma}_{MLE} = \dfrac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2}$ □

Q11) Suppose $\underset{\sim}{X}_1, \ldots, \underset{\sim}{X}_n \overset{iid}{\sim} N_p(\underset{\sim}{\mu}, \Sigma)$ $\underbrace{\text{known}}$.

a) Find the MLE of k, a scalar, where $\Sigma = k\Sigma_0$, μ, Σ_0

*We want to find the likelihood of k.

$$L(k \mid \underset{\sim}{X}_1, \ldots, \underset{\sim}{X}_n, \mu, \Sigma_0) = \frac{1}{(2\pi)^{\frac{np}{2}} |\Sigma|^{\frac{n}{2}}} e^{-\frac{1}{2}\sum_{j=1}^{n}(\underset{\sim}{X}_j - \mu)^T \Sigma^{-1} (\underset{\sim}{X}_j - \mu)}$$

$$= \frac{1}{(2\pi)^{\frac{np}{2}}} \frac{1}{|k|^{\frac{np}{2}} |\Sigma_0|^{\frac{n}{2}}} e^{-\frac{1}{2}\sum_{j=1}^{n}\frac{1}{k}(X_j - \mu)^T \Sigma_0^{-1} (X_j - \mu)} \quad (*)$$

*The log likelihood

$$l(k \mid \underset{\sim}{X}_1, \ldots, \underset{\sim}{X}_n, \mu, \Sigma_0) \propto -\frac{n}{2}\log k - \frac{1}{2k}\sum_{j=1}^{n}(X_j - \mu)^T \Sigma_0^{-1} (X_j - \mu)$$

$$\frac{\partial l}{\partial k} = -\frac{n}{2k} + \frac{1}{2k^2}\sum_{j=1}^{n}(X_j - \mu)^T \Sigma_0^{-1} (X_j - \mu)$$

$$\frac{\partial^2 l}{\partial k^2} = \frac{+n}{2k^2} - \frac{1}{k^3}\sum_{j=1}^{n}(X_j - \mu)^T \Sigma_0^{-1} (X_j - \mu)$$

$$\frac{\partial l}{\partial k} = 0 \Rightarrow \hat{k} = \frac{1}{n}\sum_{j=1}^{n}(X_j - \mu)^T \Sigma_0^{-1} (X_j - \mu) \quad \text{is a maximum since } \left.\frac{\partial^2 l}{\partial k^2}\right|_{k=\hat{k}} < 0$$

b) Find the MLE of k, a scalar, where $\mu = k\mu_0$, μ_0, Σ are known.

$$L(k \mid X_1, \ldots, X_n, \mu_0, \Sigma) \propto \frac{1}{|\Sigma|^{\frac{n}{2}}} e^{-\frac{1}{2}\sum_{j=1}^{n}(\underset{\sim}{X}_j - k\underset{\sim}{\mu_0})^T \Sigma^{-1} (\underset{\sim}{X}_j - k\underset{\sim}{\mu_0})}$$

$$l(k \mid X_1, \ldots, \mu_0, \Sigma) \propto \frac{n}{2} \frac{1}{|\Sigma|^{\frac{n}{2}+1}}$$

MATH 755 MULTIVARIATE STATISTICAL ANALYSIS

Assignment 1 — Due on in class Thursday September 6th

Let $\mathbf{j} = \mathbf{1}$ denote the column vector of ones, and $\mathbf{J} = \mathbf{j}\mathbf{j}^T = \mathbf{1}\mathbf{1}^T$ be the square matrix of ones. All constants and random variables are taken to be real valued. The population variance σ^2 is taken to be positive ($\sigma^2 > 0$).

Q 1. Consider the sample variance S^2 for a random sample $X_1, \ldots, X_n \overset{iid}{\sim} (\mu, \sigma^2)$.

(a) Write the sample variance as a quadratic form

(b) Using the appropriate theorems show that it is unbiased for σ^2, namely show that $E(S^2) = \sigma^2$.

(a) Note that $\mathbf{I}\mathbf{X} = \mathbf{X}$ and $1/n\mathbf{J}\mathbf{X} = \bar{X}\mathbf{1}$. Thus the sum of squared deviations must be

$$\begin{aligned}\sum (X_i - \bar{X})^2 &= ((\mathbf{I} - 1/n\mathbf{J})\mathbf{X})^T((\mathbf{I} - 1/n\mathbf{J})\mathbf{X}) \\ &= ((\mathbf{I} - 1/n\mathbf{J})\mathbf{X})^T((\mathbf{I} - 1/n\mathbf{J})\mathbf{X})\end{aligned}$$

Note that $\mathbf{I} - 1/n\mathbf{J}$ is symmetric (since $\mathbf{I}$ and $\mathbf{J}$ are symmetric) and idempotent since

$$\begin{aligned}(\mathbf{I} - 1/n\mathbf{J})^2 &= \mathbf{I} - 2/n\mathbf{J} + 1/n^2\mathbf{J}^2 \\ &= \mathbf{I} - 2/n\mathbf{J} + 1/n^2(n\mathbf{J}) \\ &= \mathbf{I} - 2/n\mathbf{J} + 1/n\mathbf{J} \\ &= \mathbf{I} - 1/n\mathbf{J}\end{aligned}$$

so continuing along we have

$$\begin{aligned}\sum (X_i - \bar{X})^2 &= \mathbf{X}^T(\mathbf{I} - 1/n\mathbf{J})^T(\mathbf{I} - 1/n\mathbf{J})\mathbf{X} \\ &= \mathbf{X}^T(\mathbf{I} - 1/n\mathbf{J})^2\mathbf{X} \\ &= \mathbf{X}^T(\mathbf{I} - 1/n\mathbf{J})\mathbf{X}\end{aligned}$$

and thus we have that

$$S^2 = \frac{1}{n-1}\mathbf{X}^t(\mathbf{I} - 1/n\mathbf{J})\mathbf{X}$$

(b) Note that $\mu = \mu\mathbf{1}$ and $\Sigma = \sigma^2\mathbf{I}$, then by Theorem 2.4 we have

$$\begin{aligned}E\left[S^2\right] &= \frac{1}{n-1}E\left[\mathbf{X}^t(\mathbf{I} - 1/n\mathbf{J})\mathbf{X}\right] \\ &= \frac{1}{n-1}\left[tr\left((\mathbf{I} - 1/n\mathbf{J})\sigma^2\mathbf{I}\right) + \mu^T\mathbf{I}\mu - \frac{1}{n}\mu^T\mathbf{J}\mu\right] \\ &= \frac{1}{n-1}\left[\sigma^2 tr(\mathbf{I} - 1/n\mathbf{J}) + \mu^T\mathbf{I}\mu - \frac{1}{n}\mu^T(n\mu)\right] \\ &= \frac{1}{n-1}\left[\sigma^2(tr\mathbf{I} - 1/ntr\mathbf{J}) + \mu^T\mathbf{I}\mu - \mu^T\mathbf{I}\mu\right] \\ &= \frac{1}{n-1}\left[\sigma^2(n - 1/n(n))\right] \\ &= \frac{1}{n-1}\left[\sigma^2(n-1)\right] \\ &= \sigma^2\end{aligned}$$

Q 2. Let $\mathbf{X}$ be a p-by-1 random vector with variance-covariance matrix $\mathbf{\Sigma}$.

(a) Explain why $\mathbf{\Sigma}$ must be positive-semi-definite. (What would it imply if $\mathbf{\Sigma}$ wasn't positive?)

(b) If $\mathbf{\Sigma}$ is positive-semi-definite, but not positive-definite, what does this say about $\mathbf{X}$? (Consider $\mathbf{c}^T\mathbf{X}$ for some constant column vector $\mathbf{c}$.)

(a) Let $\mathbf{c}$ be a p-by-1 constant vector. Then the variance of the linear combination $Z = \mathbf{c}^T\mathbf{X}$ is

$$V(Z) = V(\mathbf{c}^T\mathbf{X}) = \mathbf{c}^T\mathbf{\Sigma}_\mathbf{X}\mathbf{c}$$

If $\mathbf{\Sigma}_\mathbf{X}$ is not positive-semidefinite, then there exists a $\mathbf{c}$ such that

$$0 > \mathbf{c}^T\mathbf{\Sigma}_\mathbf{X}\mathbf{c} = V(Z)$$

which is impossible since for all RV's Z

$$V(Z) = E[(Z - \mu_Z)^2] \geq 0$$

(b) If $\mathbf{\Sigma}$ is positive-semi-definite, but not positive-definite, then there exists a non-zero constant vector $\mathbf{c}$ such that the random variable Z has

$$V(Z) = V(\mathbf{c}^T\mathbf{X}) = \mathbf{c}^T\mathbf{\Sigma}_\mathbf{X}\mathbf{c} = 0$$

This can happen in two ways. Either

(i) One (or more) random variables are degenerate (i.e., $V(X_i) = 0$)
(ii) One (or more) subset of random variables are linearly dependent (almost surely)

(i) is technically a subcase of (ii) but is distinct in interpretation and ease of resolution. In either case we have redundant variables.

Q 3. Let $\mathbf{A}$ be a p-by-p positive definite matrix.

(a) Show that there exists a p-by-1 random vector $\mathbf{X}$ such that $\mathbf{\Sigma_X} = \mathbf{A}$. (You may assume that there exists a random vector $\mathbf{Z}$ where $\mathbf{Z} = (Z_1, \ldots, Z_p)$ and $Z_1, \ldots, Z_p \overset{\text{iid}}{\sim} (0, 1)$.)

(b) Find a transformation of $\mathbf{X}$ such that the variance-covariance matrix of this transformed variable $\mathbf{Y} = \mathbf{CX}$ is the same as the correlation matrix of $\mathbf{X}$ ($\mathbf{\Sigma_Y} = \rho_\mathbf{X}$). Comment on the relationship of $\mathbf{Y}$ to $\mathbf{X}$.

(a) Since $\mathbf{A}$ is positive definite it can be factored via its spectral value decomposition as $\mathbf{A} = \mathbf{P\Lambda P}^T$, and it has a symmetric square root, namely $\mathbf{A}^{1/2} = \mathbf{P\Lambda}^{1/2}\mathbf{P}^T$.

Let us consider first a random vector $\mathbf{Y}$ with mean vector $\mu = \mathbf{0}$ and variance-covariance matrix $\mathbf{\Sigma_Y} = \mathbf{I}$. We consider the random variable $\mathbf{X}$ via the following transformation $\mathbf{X} = \mathbf{A}^{1/2}\mathbf{Y}$. Then

$$\begin{aligned}\mathbf{\Sigma_X} &= V(\mathbf{X}) = V\left(\mathbf{A}^{1/2}\mathbf{Y}\mathbf{A}^{1/2}\right) \\ &= \mathbf{A}^{1/2}\mathbf{\Sigma_Y}\left[\mathbf{A}^{1/2}\right]^T \\ &= \mathbf{A}^{1/2}\mathbf{I}\mathbf{A}^{1/2} \\ &= \mathbf{A}\end{aligned}$$

as desired

(b) Let us consider transformations $\mathbf{Y} = \mathbf{CX}$ where $\mathbf{C}$ is a p-by-p constant matrix. Then

$$V(\mathbf{Y}) = V(\mathbf{CX}) = \mathbf{C\Sigma_X C}^T$$

but

$$\rho_\mathbf{X} = \mathbf{\Lambda_X^{-1/2}\Sigma_X\Lambda_X^{-1/2}}$$

So that if $\mathbf{C} = \mathbf{\Lambda}^{-1/2}$, so that $\mathbf{Y} = \mathbf{\Lambda_X^{-1/2}X}$ has the desired property.

Note that $\mathbf{\Lambda_X^{-1/2}}$ is the diagonal matrix of the reciprocal of the standard deviations. So $Y_i = \frac{1}{\sigma_i}X_i, i = 1, 2, \ldots, p$ are the un-centered normalized observations.

(Student answers could include other shifted versions)

Q 4. Suppose that $\mathbf{Y} \sim (\mu, \Sigma)$ is a 3-by-1 random vector where $\mu = \mu j$ and for some ρ,

$$\Sigma_{\mathbf{Y}} = \sigma^2 \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{bmatrix}$$

(a) Write Σ as a linear combination of $\mathbf{I}$ and $\mathbf{J}$.

(b) For what values of ρ is Σ a valid variance-covariance matrix..

(c) Compute the expectation $E\left[\sum_{i=1}^{3}(Y_i - \bar{Y})^2\right]$. Comment on this as a function of ρ.

(a) $\Sigma = \sigma^2\left[(1-\rho)\mathbf{I} + \rho\mathbf{J}\right]$

(b) Checking Sylvesters criterion we have

$\det(R_1) = \sigma^2 > 0$ for all ρ

$\det(R_2) = \sigma^4(1-\rho^2) > 0$ if $|\rho| < 1$

$\det(R_3) = \sigma^6(1-\rho^2)(1+2\rho) > 0$ if $-1/2 < \rho < 1$ or $\rho < -1$

So Σ is positive definite only if $-1/2 < \rho < 1$. If $\rho = -1/2, 1$ then Σ is positive semi-definite. Thus Σ is a valid variance covariance matrix if and only if $-1/2 \leq \rho \leq 1$

(c) We should first write out the sum as a quadratic form. Namely $\mathbf{Y}^T(\mathbf{I} - 1/n\mathbf{J})\mathbf{Y} = \sum_{i=1}^{3}(Y_i - \bar{Y})^2$
Then by Theorem 2.4 we have (n=3)

$$\begin{aligned}
E\left[\sum_{i=1}^{3}(Y_i - \bar{Y})^2\right] &= E\left[\mathbf{Y}^T(\mathbf{I} - 1/n\mathbf{J})\mathbf{Y}\right] \\
&= tr((\mathbf{I} - 1/n\mathbf{J})\sigma^2((1-\rho)\mathbf{I} + \rho\mathbf{J})) + \mu^T(\mathbf{I} - 1/n\mathbf{J})\mu \\
&= \sigma^2 tr\left[(1-\rho)\mathbf{I} - \frac{1}{n}(1-\rho)\mathbf{J} + \rho\mathbf{J} - \frac{1}{n}\rho\mathbf{J}^2\right] + \mu^T\mathbf{I}\mu - \frac{1}{n}\mu^T\mathbf{J}\mu \\
&= \sigma^2 tr\left[(1-\rho)\mathbf{I} - \frac{1}{n}(1-\rho)\mathbf{J} + \rho\mathbf{J} - \frac{1}{n}\rho n\mathbf{J}\right] + \mu^T\mu - \frac{1}{n}\mu^T(n\mu) \\
&= \sigma^2 tr\left[(1-\rho)\mathbf{I} - \frac{1}{n}(1-\rho)\mathbf{J} + \rho\mathbf{J} - \frac{1}{n}\rho n\mathbf{J}\right] + \mu^T\mu - \mu^T\mu \\
&= \sigma^2 tr\left[(1-\rho)\mathbf{I} - \frac{1}{n}(1-\rho)\mathbf{J} + \rho\mathbf{J} - \rho\mathbf{J}\right] \\
&= \sigma^2 tr\left[(1-\rho)\mathbf{I} - \frac{1}{n}(1-\rho)\mathbf{J}\right] \\
&= \sigma^2\left[(1-\rho)tr\mathbf{I} - \frac{1}{n}(1-\rho)tr\mathbf{J}\right] \\
&= \sigma^2\left[(1-\rho)n - \frac{1}{n}(n)(1-\rho)\right] \\
&= \sigma^2(1-\rho)(n-1) \\
&= 2\sigma^2(1-\rho)
\end{aligned}$$

This expectation and thus the mean of the sample variance is strictly decreasing to 0 as ρ increases to 1.

Q 5. Let $\mathbf{X}$ be a 3-by-1 random vector with variance covariance matrix

$$\mathbf{\Sigma_X} = \begin{bmatrix} 5 & -4 & 1 \\ -4 & 13 & 9 \\ 1 & 9 & 10 \end{bmatrix}$$

(a) Find a unit vector $\mathbf{c} = (c_1, c_2, c_3)^T$ such that the variance of $\mathbf{c}^T\mathbf{X}$ is maximized/minimized. (You may use a computer for convenience)

(b) What is the largest/smallest possible variance for such linear combinations?

We want the eigenvectors associated with the largest/smallest eigenvalue. From R, they eigen vectors/values are

e_1	e_2	e_3
0.1543033	0.8017837	0.5773503
-0.7715167	-0.2672612	0.5773503
-0.6172134	0.5345225	-0.5773503

or

e_1	e_2	e_3
$1/\sqrt{42}$	$3/\sqrt{14}$	$1/\sqrt{3}$
$-5/\sqrt{42}$	$-1/\sqrt{14}$	$1/\sqrt{3}$
$-4/\sqrt{42}$	$2/\sqrt{14}$	$-1/\sqrt{3}$

with associated eigenvalues $\lambda_1 = 21, \lambda_2 = 7, \lambda_3 = 0$ (R will return $2.131628e-14$ for the smallest, but this must be 0, since the determinant is 0)

Therefore, the constant vectors producing the maximum/minimum variances (21/0) of the linear combinations are $(1, -5, 4)/\sqrt{42}$ and $(1, 1, -1)/\sqrt{3}$ respectively.

(One can note then, that $X_3 = X_1 + X_2$ almost surely.)

Q 6. Johnson and Wichern 2.41

We have

(a) $E(\mathbf{AX}) = AE\mathbf{X}$

$$= \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ -2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 7 & 3 \end{bmatrix}$$

(b) $Cov(\mathbf{AX}) = \mathbf{A}Cov\mathbf{X}\mathbf{A}^T$

$$= \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix} \begin{bmatrix} 3 & 3 & 3 \\ -3 & 3 & 3 \\ 0 & -6 & 3 \\ 0 & 0 & -9 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & 36 \end{bmatrix}$$

(c) It is clear that the three pairs are all uncorrelated.

Q 7. Let $\mathbf{X} \sim (\mu, \mathbf{\Sigma})$. Without using a computer compute

(a) $|\mathbf{\Sigma}|$

(b) $tr\mathbf{\Sigma}$

(c) $V(\mathbf{e}_i^T\mathbf{X}), i = 1, 2, 3$

(d) $\mathbf{\Sigma}$

(e) $\mathbf{\Sigma}^{1/2}$

where $\lambda_1 = 3$, $\lambda_2 = 7$, and $\lambda_3 = 1$ are the eigenvalues of $\mathbf{\Sigma}$, and $\mathbf{e}_1^T = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]$, $\mathbf{e}_2^T = [2/\sqrt{6}, -1/\sqrt{6}, -1/\sqrt{6}]$, and $\mathbf{e}_3^T = [0, 1/\sqrt{2}, -1/\sqrt{2}]$ are the associated eigenvectors.

(a) $|\mathbf{\Sigma}| = \prod_{i=1}^{e} \lambda_i = (7)(3)(1) = 21$

(b) $tr\mathbf{\Sigma} = \sum_{i=1}^{3} \lambda_i = 7 + 3 + 1 = 11$

(c) We have

$$\begin{aligned} V(\mathbf{e}_i^T\mathbf{X}) &= \mathbf{e}_i^T\mathbf{\Sigma}_\mathbf{X}\mathbf{e}_i^T \\ &= \mathbf{e}_i^T\left[\sum_{i=j}^{3}\lambda_j\mathbf{e}_j\mathbf{e}_j^T\right]\mathbf{e}_i \\ &= \sum_{i=1}^{3}\lambda_j\mathbf{e}_i^T\mathbf{e}_j\mathbf{e}_j^T\mathbf{e}_i \\ &= \sum_{i=1}^{3}\lambda_j\begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \\ &= \lambda_i \end{aligned}$$

(d) We have

$$\begin{aligned} \mathbf{\Sigma} = \sum_{i=j}^{3}\lambda_j\mathbf{e}_j\mathbf{e}_j^T &= 3\begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}\begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} \\ &\quad + 7\begin{bmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ -1/\sqrt{6} \end{bmatrix}\begin{bmatrix} 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}\begin{bmatrix} 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \\ &= 3/3\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + 7/6\begin{bmatrix} 4 & -2 & -2 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{bmatrix} + 1/2\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \\ &= \frac{1}{3}\begin{bmatrix} 17 & -4 & -4 \\ -4 & 8 & 5 \\ -4 & 5 & 8 \end{bmatrix} \end{aligned}$$

(e) We have

$$\begin{aligned}
\boldsymbol{\Sigma}^{1/2} = \sum_{i=j}^{3} \sqrt{\lambda_j}\mathbf{e}_j\mathbf{e}_j^T &= \sqrt{3}\begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}\begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} \\
&\quad + \sqrt{7}\begin{bmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ -1/\sqrt{6} \end{bmatrix}\begin{bmatrix} 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}\begin{bmatrix} 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \\
&= \sqrt{3}/3\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \sqrt{7}/6\begin{bmatrix} 4 & -2 & -2 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{bmatrix} + 1/2\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \\
&= \frac{1}{6}\left\{\begin{bmatrix} 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \\ 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \\ 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \end{bmatrix} + \begin{bmatrix} 4\sqrt{7} & -2\sqrt{7} & -2\sqrt{7} \\ -2\sqrt{7} & \sqrt{7} & \sqrt{7} \\ -2\sqrt{7} & \sqrt{7} & \sqrt{7} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & -3 \\ 0 & -3 & 3 \end{bmatrix}\right\} \\
&= \frac{1}{6}\begin{bmatrix} 2\sqrt{3}+4\sqrt{7} & 2\sqrt{3}-2\sqrt{7} & 2\sqrt{3}-2\sqrt{7} \\ 2\sqrt{3}-2\sqrt{7} & 2\sqrt{3}+\sqrt{7}+3 & 2\sqrt{3}+\sqrt{7}-3 \\ 2\sqrt{3}-2\sqrt{7} & 2\sqrt{3}+\sqrt{7}-3 & 2\sqrt{3}+\sqrt{7}+3 \end{bmatrix}
\end{aligned}$$

Q 8. Let $\mathbf{X} \sim N_3(\mu, \Sigma)$ where $\mu^T = [1, 0, -2]$ and

$$\Sigma = \begin{bmatrix} 2 & -0.5 & 1 \\ -0.5 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

(a) Which components of $\mathbf{X}$ are independent?

(b) Find the conditional distribution of $X_2|(X_1, X_3)^T$

(c) Find the conditional distribution of $(X_1, X_3)^T|X_2$

(d) Comment on how your answer to part (a) affects parts (b) and (c).

(a) Checking the covariances clearly only X_2 and X_3 are independent.

Using the appropriate theorem for (b) and (c)

(b) We want $X_2|(X_1, X_3)^T = (a, c)^T$. Then

$$\begin{aligned} E\left(X_2 \middle| \begin{bmatrix} X_1 \\ X_3 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix}\right) &= 0 + \begin{bmatrix} -0.5 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} a \\ c \end{bmatrix} - \begin{bmatrix} 1 \\ -2 \end{bmatrix} \right) \\ &= \frac{1}{3} \begin{bmatrix} -0.5 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} a-1 \\ c+2 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} -1 & 0.5 \end{bmatrix} \begin{bmatrix} a-1 \\ c+2 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} -1 & 0.5 \end{bmatrix} \begin{bmatrix} a-1 \\ c+2 \end{bmatrix} \\ &= \frac{1}{3} [(1-a) + 1/2(c+2)] = \frac{1}{3} [2 + 1/2c - a] \\ &= 2/3 + 1/6c - 1/3a \end{aligned}$$

$$\begin{aligned} Cov\left(X_2 \middle| \begin{bmatrix} X_1 \\ X_3 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix}\right) &= 1 - \begin{bmatrix} -0.5 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \\ &= 1 - \frac{1}{3} \begin{bmatrix} -0.5 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \\ &= 1 - \frac{1}{3} \begin{bmatrix} -0.5 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0.5 \end{bmatrix} \\ &= 1 - \frac{1}{3} 0.5 = 5/6 \end{aligned}$$

So that $X_2|(X_1, X_3)^T = (a, c)^T \sim N(2/3 + 1/6c - 1/3a, 5/6)$

(c) We want $(X_1, X_3)^T | X_2 = b$. Then

$$
\begin{aligned}
E\left(\begin{bmatrix} X_1 \\ X_3 \end{bmatrix} | X_2 = b\right) &= \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix}^{-1} \begin{bmatrix} b - 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \begin{bmatrix} b \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} -0.5b \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 - 1/2b \\ -2 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
Cov\left(\begin{bmatrix} X_1 \\ X_3 \end{bmatrix} | X_2 = b\right) &= \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix}^{-1} \begin{bmatrix} -0.5 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \begin{bmatrix} -0.5 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} 1/4 & 0 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 7/4 & 1 \\ 1 & 2 \end{bmatrix}
\end{aligned}
$$

So that $(X_1, X_3)^T | X_2 = b \sim N_2\left(\begin{bmatrix} 1 - 1/2b \\ -2 \end{bmatrix}, \begin{bmatrix} 7/4 & 1 \\ 1 & 2 \end{bmatrix}\right)$

(d) In part (b) the conditional mean is affected by both values of X_1 and X_3 even though X_3 is independent of X_1 because X_1 and X_3 are correlated. In part (c) only the conditional mean of X_1 is changed as X_2 and X_3 are independent.

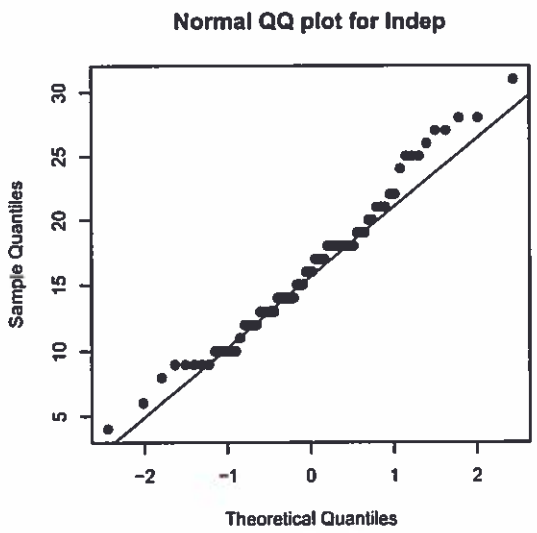
In part (b) the variance is reduced by a factor depending on the conditioning (but not the value) of both X_1 and X_3 since they are correlated. In part (c), only the variance of X_1 is affected and not the covariance or variance of X_2.

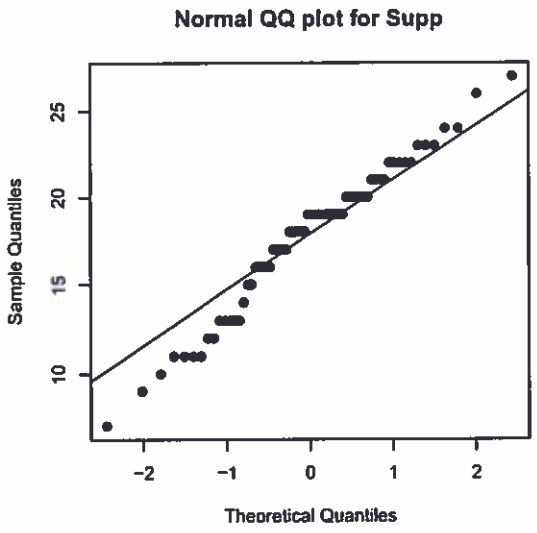
Q 9. Using the data in Table 4.6 from the book (T4-6), check for univariate and multivariate normality for the females. Comment.

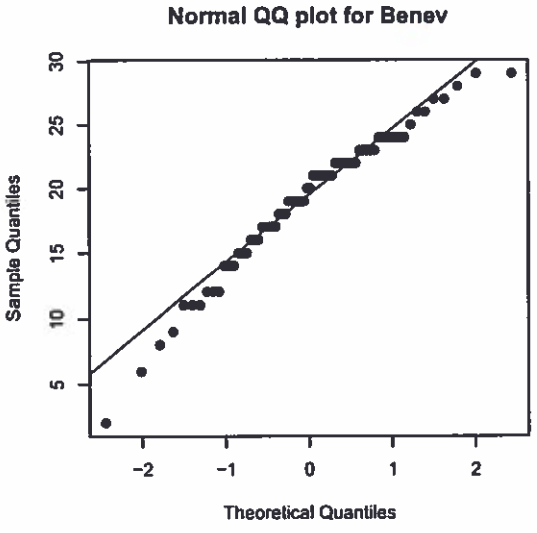
The normal QQ plots look reasonably straight with some mild curvature in Supp and Leader but nothing exaggerated. However, the multivariate chisquared qq plot definitely has some stronger curvature indicating multivariate normality might be violated.

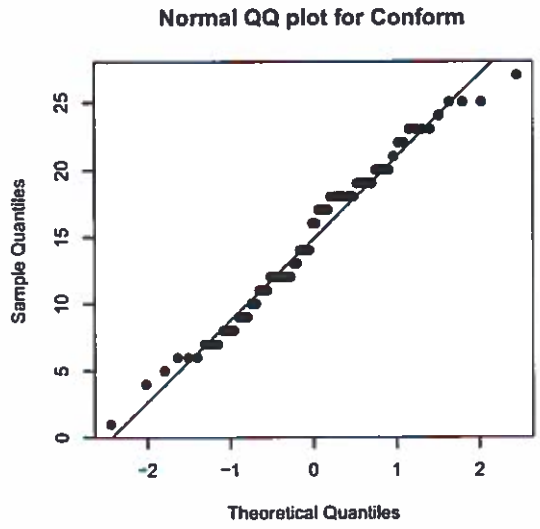
(As a side note, anderson-darling tests applied to the marginal distributions give a p-value of about 0.02 for Supp and 0.001 for Leader, and Royston's test for Multivariate Normality gives a p-value of about 0.01.

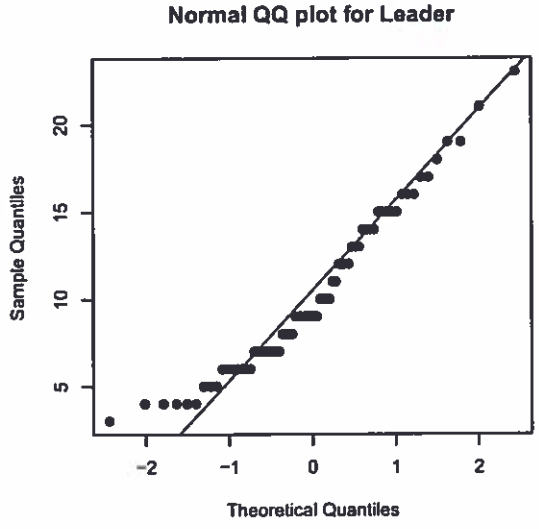
```
# ------------------------------------------------------------------
#
# Import functions
#
# ------------------------------------------------------------------

library(mvnTest) # For the Royston test & joint chisq plot

# ------------------------------------------------------------------
#
# Import data and draw plots
#
# ------------------------------------------------------------------

psych <- read.table(file = "datasets/Wichern_data/T4-6.dat")
names(psych) <- c("Indep", "Supp", "Benev", "Conform", "Leader", "Gender", "Socio")

is.female <- which(psych$Gender == 2)
psych.f <- psych[is.female, ]

dev.new(width = 15, height = 10)
par(mfrow = c(2, 3))

for(j in c("Indep", "Supp", "Benev", "Conform", "Leader")){
  qqnorm(psych.f[,j], main = paste("Normal QQ plot for", j), pch = 19)
  qqline(psych.f[,j])
}

R.test(psych.f[,c("Indep", "Supp", "Benev", "Conform", "Leader")], qqplot = TRUE)
```

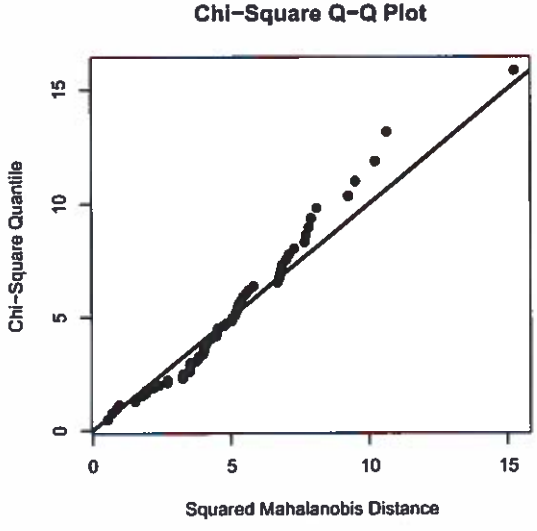

Normal QQ plot for Indep
Sample Quantiles
Theoretical Quantiles
5
10
15
20
25
30
−2
−1
0
1
2


Normal QQ plot for Supp
Sample Quantiles
Theoretical Quantiles
10
15
20
25
−2
−1
0
1
2


Normal QQ plot for Benev
Sample Quantiles
Theoretical Quantiles
5
10
15
20
25
30
−2
−1
0
1
2


Normal QQ plot for Conform
Sample Quantiles
Theoretical Quantiles
0
5
10
15
20
25
−2
−1
0
1
2


Normal QQ plot for Leader
Sample Quantiles
Theoretical Quantiles
5
10
15
20
−2
−1
0
1
2


Chi−Square Q−Q Plot
Chi−Square Quantile
Squared Mahalanobis Distance
0
5
10
15
0
5
10
15

Q 10. Let $\mathbf{X}_1 \sim N_p(\mu_1, \Sigma)$ and $\mathbf{X}_2 \sim N_p(\mu_2, \Sigma)$. Given independent random samples of sizes n_1 and n_2 respectively, obtain the MLE's of μ_1, μ_2, and Σ.

The likelihood is

$$\begin{aligned}
&\ell(\mu_1, \mu_2, \Sigma | \mathbf{x}_{11}, \ldots, \mathbf{x}_{1n_1}, \mathbf{x}_{21}, \ldots, \mathbf{x}_{2n_2}) \\
&\quad = \left[\prod_{j=1}^{n_1} \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_{1j}-\mu_1)^T\Sigma^{-1}(\mathbf{x}_{1j}-\mu_1)}\right]\left[\prod_{j=1}^{n_2} \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_{2j}-\mu_2)^T\Sigma^{-1}(\mathbf{x}_{2j}-\mu_2)}\right] \\
&\quad = \frac{1}{(2\pi)^{(n_1+n_2)p/2}|\Sigma|^{(n_1+n_2)/2}} e^{-\frac{1}{2}\left[\sum_{j=1}^{n_1}(\mathbf{x}_{1j}-\mu_1)^T\Sigma^{-1}(\mathbf{x}_{1j}-\mu_1)+\sum_{j=1}^{n_2}(\mathbf{x}_{2j}-\mu_2)^T\Sigma^{-1}(\mathbf{x}_{2j}-\mu_2)\right]}
\end{aligned}$$

By a theorem from class we have

$$\begin{aligned}
&= \frac{1}{(2\pi)^{(n_1+n_2)p/2}|\Sigma|^{(n_1+n_2)/2}} \times \\
&e^{-\frac{1}{2}\left[tr\left(\Sigma^{-1}\left\{\sum_{j=1}^{n_1}(\mathbf{x}_{1j}-\bar{\mathbf{x}}_1)(\mathbf{x}_{1j}-\bar{\mathbf{x}}_1)^T+\sum_{j=1}^{n_2}(\mathbf{x}_{2j}-\bar{\mathbf{x}}_2)(\mathbf{x}_{2j}-\bar{\mathbf{x}}_2)^T\right\}\right)+n_1(\bar{\mathbf{x}}_1-\mu_1)^T\Sigma^{-1}(\bar{\mathbf{x}}_1-\mu_1)+n_2(\bar{\mathbf{x}}_2-\mu_2)^T\Sigma^{-1}(\bar{\mathbf{x}}_2-\mu_2)\right]} \\
&= \left[\frac{1}{(2\pi)^{(n_1+n_2)p/2}}\right]\left[\frac{1}{|\Sigma|^{(n_1+n_2)/2}} e^{-\frac{1}{2}tr\left(\Sigma^{-1}\left\{\sum_{j=1}^{n_1}(\mathbf{x}_{1j}-\bar{\mathbf{x}}_1)(\mathbf{x}_{1j}-\bar{\mathbf{x}}_1)^T+\sum_{j=1}^{n_2}(\mathbf{x}_{2j}-\bar{\mathbf{x}}_2)(\mathbf{x}_{2j}-\bar{\mathbf{x}}_2)^T\right\}\right)}\right] \times \\
&\quad \left[e^{-\frac{1}{2}n_1(\bar{\mathbf{x}}_1-\mu_1)^T\Sigma^{-1}(\bar{\mathbf{x}}_1-\mu_1)}\right]\left[e^{-\frac{1}{2}n_2(\bar{\mathbf{x}}_2-\mu_2)^T\Sigma^{-1}(\bar{\mathbf{x}}_2-\mu_2)}\right]
\end{aligned}$$

All 4 products are strictly positive, the first being a constant and thus ignorable. The third and fourth are clearly maximized (uniquely since Σ is positive definite and thus Σ^{-1} is positive definite) when the quadratic form is 0, which occurs if $\hat{\mu}_1 = \bar{\mathbf{x}}_1$ and $\hat{\mu}_2 = \bar{\mathbf{x}}_2$. The second term is maximized by applying the theorem from class with

$$\begin{aligned}
b &= 1/2(n_1 + n_2) \\
B &= \sum_{j=1}^{n_1}(\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)(\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)^T + \sum_{j=1}^{n_2}(\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)(\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)^T
\end{aligned}$$

yielding

$$\begin{aligned}
\hat{\Sigma} &= \frac{1}{n_1 + n_2}\left[\sum_{j=1}^{n_1}(\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)(\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)^T + \sum_{j=1}^{n_2}(\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)(\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)^T\right] \\
&= \frac{1}{n_1 + n_2}\left[(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2\right]
\end{aligned}$$

Q 11. Suppose $\mathbf{X}_1, \ldots, \mathbf{X}_n \overset{iid}{\sim} N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

(a) Find the mle of k- a scalar- where $\boldsymbol{\Sigma} = k\boldsymbol{\Sigma}_0$ and both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_0$ are known.

(b) Find the mle of k- a scalar- where $\boldsymbol{\mu} = k\boldsymbol{\mu}_0$ and both $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}$ are known.

(a) The likelihood as a function of k is

$$L(k|\mathbf{x}_1, \ldots, x_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}_0) = \frac{1}{(2\pi)^{np/2}|k\boldsymbol{\Sigma}_0|^{n/2}} e^{-\frac{1}{2}\left[\sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu})^T [k\boldsymbol{\Sigma}_0]^{-1} (\mathbf{x}_j - \boldsymbol{\mu})\right]}$$

$$= \left[\frac{1}{(2\pi)^{np/2}|\boldsymbol{\Sigma}_0|^{n/2}}\right] \left[\frac{1}{k^{np/2}} e^{-\frac{1}{k}\left[\frac{1}{2}\sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x}_j - \boldsymbol{\mu})\right]}\right]$$

With log-likelihood

$$\ell(k|\mathbf{x}_1, \ldots, x_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}_0) = \ln L(k|\mathbf{x}_1, \ldots, x_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}_0)$$

$$= \text{constant} \; - \frac{np}{2} \ln k - \frac{1}{2k} \sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x}_j - \boldsymbol{\mu})$$

with derivative

$$\frac{d\ell}{dk} = -\frac{np}{2k} + \frac{1}{2k^2} \sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x}_j - \boldsymbol{\mu})$$

setting to 0 and solving yields

$$\widehat{k} = \frac{1}{np} \sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x}_j - \boldsymbol{\mu})$$

which is a (local) maximum because

$$\left.\frac{d^2\ell}{dk^2}\right|_{k=\widehat{k}} = \frac{np}{2\widehat{k}^2} - \frac{1}{\widehat{k}^3} \sum_{j=1}^n (\mathbf{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x}_j - \boldsymbol{\mu})$$

$$= -\frac{np^2}{2\widehat{k}} < 0$$

$(\hat{k}-2)^2 = 22$

(b)

$$L(k|\mathbf{x}_1, \ldots, x_n, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{np/2}|\boldsymbol{\Sigma}|^{n/2}} e^{-\frac{1}{2}\left[\sum_{j=1}^n (\mathbf{x}_j - k\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_j - k\boldsymbol{\mu}_0)\right]}$$

With log-likelihood

$$\ell(k|\mathbf{x}_1, \ldots, x_n, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}) = L(k|\mathbf{x}_1, \ldots, x_n, \boldsymbol{\mu}_0, \boldsymbol{\Sigma})$$

$$= \text{constant} \; - \frac{1}{2} \left[\sum_{j=1}^n (\mathbf{x}_j - k\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_j - k\boldsymbol{\mu}_0)\right]$$

$$= \text{constant} \; - \frac{1}{2} \left[\sum_{j=1}^n \left(\mathbf{x}_j^T \boldsymbol{\Sigma}^{-1} \mathbf{x}_j - 2k\boldsymbol{\mu}_0 \boldsymbol{\Sigma}^{-1} \mathbf{x}_j + k^2 \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0\right)\right]$$

$$= \text{constant} \; - \frac{1}{2} \left[\sum_{j=1}^n \left(-2k\boldsymbol{\mu}_0 \boldsymbol{\Sigma}^{-1} \mathbf{x}_j + k^2 \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0\right)\right]$$

$$= \text{constant} \; - \frac{1}{2} \left[nk^2 \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0 - 2 \sum_{j=1}^n k\boldsymbol{\mu}_0 \boldsymbol{\Sigma}^{-1} \mathbf{x}_j\right]$$

The derivative is

$$\frac{d\ell}{dk} = -nk\mu_0^T \Sigma^{-1} \mu_0 + \sum_{j=1}^{n} \mu_0 \Sigma^{-1} \mathbf{x}_j$$

setting to 0 and solving yields

$$\begin{aligned} \widehat{k} &= \frac{\sum_{j=1}^{n} \mu_0 \Sigma^{-1} \mathbf{x}_j}{n\mu_0^T \Sigma^{-1} \mu_0} \\ &= \frac{\mu_0 \Sigma^{-1} \bar{\mathbf{x}}}{\mu_0^T \Sigma^{-1} \mu_0} \end{aligned}$$

which is a global maximum because

$$\frac{d^2\ell}{dk^2} = -n\mu_0^T \Sigma^{-1} \mu_0 < 0$$

provided $\mu_0 \neq 0$. (Of course if $\mu_0 = 0$ then estimating the scale constant k is not possible.)

MATH 755 HW 3

Due 10/04/2018

Let $\underset{\sim}{X} = \begin{bmatrix} 2 & 12 \\ 8 & 9 \\ 6 & 9 \\ 8 & 10 \end{bmatrix}$ Test $H_0: \mu' = [7 \;\; 11]$
a) Find T^2 for testing $\mu' = [7 \;\; 11]$
b) Specify the dist of T^2 in a)
c) Use a) and b) at $\alpha = 0.05$. ? conclusion.

EX 5.17 / ~~1.~~ Johnson & Wichern 5.1

~~2.~~ Johnson & Wichern 5.18 (a)

```
college <- read.table(file = "Wichern_data/T5-2.dat")
```

a) Test: $H_0: \mu = \begin{bmatrix} 500 \\ 50 \\ 30 \end{bmatrix}$ and $H_1: \mu \neq \begin{bmatrix} 500 \\ 50 \\ 30 \end{bmatrix}$ at $\alpha = 0.05$ b) Length and direction of the axes of the 95% CI for μ.

3. Johnson & Wichern 5.19

c) QQ plot.

```
lumber <- read.table(file = "Wichern_data/T5-11.dat")
names(lumber) <- c("stiff", "bend")
```

a) Confidence ellipse for $\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ $\mu_1 = E(X_1)$ $\mu_2 = E(X_2)$.

not finished drawing ellipse.

4. Johnson & Wichern 6.24

```
egypt <- read.table(file = "Wichern_data/T6-13.dat")
names(egypt) <- c("MaxBreath", "BasHeight", "BasLength", "NasHeight", "TimePeriod")
```

5. Find a dataset online which contains (i) two factors and (ii) at least two continuous responses.

(a). Run a one-way MANOVA using only one factor. State your conclusions in plain language.

(b). Run a two-way MANOVA with interactions. State your conclusions in plain language. Refit a two-way MANOVA without interactions if the interactions are not significant. State your conclusions in plain language.

(c). One assumption we made in MANOVA is that the variance-covariance matrices of each group of residuals are equal. Does this assumption hold in your dataset?

MAT 755 Midterm Fall 2018

Due: October 16, 2018

Instructions: This is a take-home exam. It will weight 15% of your final grade. You should work on all the questions independently. If you submit your work electronically (Blackboard, email), please name your file as "lastname_midterm_F18".

1. Show that

$$\mathbf{Q} = \begin{pmatrix} \frac{5}{13} & \frac{12}{13} \\ -\frac{12}{13} & \frac{5}{13} \end{pmatrix}$$

is an orthogonal matrix.

2. Using the matrix

$$\mathbf{Q} = \begin{pmatrix} 4 & 8 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

(a). Calculate $\mathbf{Q}\mathbf{Q}^T$ and obtain its eigenvalues and eigenvectors.

(b). Calculate $\mathbf{Q}^T\mathbf{Q}$ and obtain its eigenvalues and eigenvectors. Check that the nonzero eigenvalues are the same as those in part (a).

(c). Obtain the singular value decomposition of $\mathbf{Q}$.

3. Consider an arbitrary $n \times p$ matrix $\mathbf{Q}$.

(a). Show that $\mathbf{Q}^T\mathbf{Q}$ is a symmetric $p \times p$ matrix.

(b). Show that $\mathbf{Q}^T\mathbf{Q}$ is necessarily nonnegative definite.

X_1 = lather
X_2 = mildness.

2 ways.

way 1: $\bar{x}_1 = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$ S_1

way 2 $\bar{x}_2 = \begin{bmatrix} 10 \\ 3 \end{bmatrix}$ S_2.

4. Bars of soap are manufactured in each of two ways. The characteristics $X_1 = \textit{lather}$ and $X_2 = \textit{mildness}$ are measured. The summary statistics for 61 and 61 bars produced by methods 1 and 2, respectively, are $\bar{\mathbf{x}}_1 = [8, 4]^T, \bar{\mathbf{x}}_2 = [10, 3]^T$, $\mathbf{S}_1 = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}$, $\mathbf{S}_2 = \begin{pmatrix} 7 & 2 \\ 2 & 7 \end{pmatrix}$.

(a). Test the hypothesis that the population means $\mu_1 = \mu_2$ at 10% significance level, using the pooled covariance matrix to estimate the population covariance matrix Σ and assuming that both populations are with multivariate normal distribution.

(b). Find the 90% Bonferroni simultaneous confidence interval for the individual mean difference.

5. Wichern and Johnson 7.24 on page 425

```
bull <- read.table(file="Wichern_data/T1-10.dat", header=F)
names(bull) <- c("Breed","SalePr","YrHgt","FtFrBody","PrctFFB","Frame","BkFat","SaleHt","SaleWt")
```

6. A researcher randomly assigns 33 subjects to one of three groups.

- Group 1 receives technical dietary information interactively from an on-line website.
- Group 2 receives the same information from a nurse practitioner.
- Group 3 receives the information from a video tape made by the same nurse practitioner.

Each subject then made three ratings: difficulty, usefulness, and importance of the information in the presentation. The researcher is interested in whether or not there is a difference in the modes of presentation. (Data: presentation.sav)

(a). Test the equality of the group means.

(b). What assumptions do you make in part (a)? Access the validity of the assumptions.

(c). Construct and comment the 95% Bonferroni confidence intervals for differences in mean components for the three responses for each pair of populations.

1/ Show that $Q = \begin{pmatrix} \frac{5}{13} & \frac{12}{13} \\ -\frac{12}{13} & \frac{5}{13} \end{pmatrix}$ is an orthogonal matrix

* We have Q is an orthogonal matrix iff $\begin{cases} ① Q \text{ is a square matrix (done)} \\ ② Q^TQ = QQ^T = I \end{cases}$

$$Q^TQ = \begin{pmatrix} \frac{5}{13} & -\frac{12}{13} \\ \frac{12}{13} & \frac{5}{13} \end{pmatrix}\begin{pmatrix} \frac{5}{13} & \frac{12}{13} \\ -\frac{12}{13} & \frac{5}{15} \end{pmatrix} = I$$

$$QQ^T = \begin{pmatrix} \frac{5}{13} & \frac{12}{13} \\ -\frac{12}{13} & \frac{5}{13} \end{pmatrix}\begin{pmatrix} \frac{5}{13} & -\frac{12}{13} \\ \frac{12}{13} & \frac{5}{13} \end{pmatrix} = I$$

$\Rightarrow$ ② has been done

① and ② $\Rightarrow$ Q is an orthogonal matrix

2) Using the matrix $Q = \begin{pmatrix} 4 & 8 & 8 \\ 3 & 6 & 9 \end{pmatrix}$

a) Calculate QQ^T and obtain its eigenvalues and eigenvectors

b) Calculate Q^TQ and obtain its eigenvalues and eigenvectors.
Check that the nonzero eigenvalues are the same as those in part a)

c) Obtain the singular value decomposition of Q

---

a)

* $QQ^T = \begin{pmatrix} 4 & 8 & 8 \\ 3 & 6 & 9 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 8 & 6 \\ 8 & 9 \end{pmatrix} = \begin{pmatrix} 144 & 132 \\ 132 & 126 \end{pmatrix} = A$

* Characteristic polynomial

$$\det(A - \lambda I) = \det \begin{pmatrix} 144-\lambda & 132 \\ 132 & 126-\lambda \end{pmatrix} = (144-\lambda)(126-\lambda) - 132^2 =$$

$$= \lambda^2 - 270\lambda + 720$$

$\Rightarrow$ Eigenvalues $\lambda_{1,2} = 135 \pm 3\sqrt{1945}$

7) Use the matrix

$$Q = \begin{pmatrix} 4 & 8 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

a) Calculate QQ^T and obtain its eigenvalues and eigenvectors.

* $QQ^T = \begin{pmatrix} 4 & 8 & 8 \\ 3 & 6 & 9 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 8 & 6 \\ 8 & 9 \end{pmatrix} = \begin{pmatrix} 144 & 132 \\ 132 & 126 \end{pmatrix}$

* Find eigenvalues:

• Characteristic polynomial

$$\det(QQ^T - \lambda I) = \det \begin{pmatrix} 144-\lambda & 132 \\ 132 & 126-\lambda \end{pmatrix} = (144-\lambda)(126-\lambda) - 132^2 = \lambda^2 - 270\lambda + 720$$

then $\lambda_{1,2} = 3(45 \pm \sqrt{1945}) \iff \begin{cases} \lambda_1 = 267.306 \\ \lambda_2 = 2.6995 \end{cases}$

* Find eigenvectors:

• Find eigenvectors associated with $\lambda_1 = 3(45+\sqrt{1945}) = 267.306$

the eigenvector $y = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$ is a solution of $QQ^T \underset{\sim}{u} = \lambda_1 \underset{\sim}{u} \iff (QQ^T - \lambda_1 I)\underset{\sim}{u} = \underset{\sim}{0}$

$$(QQ^T - \lambda_1 I) = \left(\begin{array}{cc|c} 144-3\cdot 45-3\cdot\sqrt{1945} & 132 & 0 \\ 132 & 126-3\cdot 45-3\cdot\sqrt{1945} & 0 \end{array}\right)$$

$$\begin{matrix} R_1 \to \frac{R_1}{3} \\ R_2 \to \frac{R_2}{3} \end{matrix} \iff \left(\begin{array}{cc|c} 48-45-\sqrt{1945} & 44 \ldots & 0 \\ 44 & 42-45-\sqrt{1945} & 0 \end{array}\right)$$

$$\iff \left(\begin{array}{cc|c} 3-\sqrt{1945} & 44 & 0 \\ 44 & -3-\sqrt{1945} & 0 \end{array}\right)$$

$$\Rightarrow \begin{cases} R_1 \Rightarrow \begin{cases} u_2 = \dfrac{-(3-\sqrt{1945})u_1}{44} \\ u_1^2 + u_2^2 = 1 \end{cases} \Rightarrow \begin{cases} u_1 = -0.7307 \\ u_2 = -0.6826 \end{cases} \end{cases}$$

• Find eigenvector of QQ^T associated with $\lambda_2 = 3\cdot 45 - 3\sqrt{1945}$

$$QQ^T - \lambda_2 I = 0 \iff \left(\begin{array}{cc|c} 144-3\cdot 45+3\sqrt{1945} & 132 & 0 \\ 132 & 126-3\cdot 45+3\sqrt{1945} & 0 \end{array}\right)$$

$$\begin{matrix} R_1 \to \frac{R_1}{3} \\ \Rightarrow \\ R_2 \to \frac{R_2}{3} \end{matrix} \left(\begin{array}{cc|c} 48-45+\sqrt{1945} & 44 & 0 \\ 44 & 42-45+\sqrt{1945} & 0 \end{array}\right)$$

$$\Rightarrow \left[\begin{array}{cc|c} 3+\sqrt{1945} & 44 & 0 \\ 44 & -3+\sqrt{1945} & 0 \end{array}\right] \Rightarrow \begin{cases} u_2 = \dfrac{-(3+\sqrt{1945})}{44} \\ u_1^2 + u_2^2 = 1 \end{cases} \Rightarrow \begin{cases} u_1 = 0.68263 \\ u_2 = -0.730761 \end{cases}$$

Calculate Q^TQ and obtain its eigenvalues and eigenvectors.
Check that the nonzero eigenvalues are the same as those in part a.

- $Q^TQ = \begin{pmatrix} 4 & 3 \\ 8 & 6 \\ 8 & 9 \end{pmatrix}\begin{pmatrix} 4 & 8 & 8 \\ 3 & 6 & 9 \end{pmatrix} = \begin{pmatrix} 25 & 50 & 59 \\ 50 & 100 & 118 \\ 59 & 118 & 145 \end{pmatrix}$

* Find eigenvalues of Q^TQ

- Chacktacteristic polynomial

$$\det(Q^TQ - \lambda I) = \det\begin{pmatrix} 25-\lambda & 50 & 59 \\ 50 & 100-\lambda & 118 \\ 59 & 118 & 145-\lambda \end{pmatrix} \xrightarrow[C_2-2C_1]{\det} \det\begin{pmatrix} 25-\lambda & 2\lambda & 59 \\ 50 & -\lambda & 118 \\ 59 & 118 & 145-\lambda \end{pmatrix}$$

$$\xrightarrow{R_2 \to 2R_2 + \lambda R_1} \det\begin{pmatrix} 25-\lambda & 2\lambda & 59 \\ 125-\lambda & 0 & 295 \\ 59 & 0 & 145-\lambda \end{pmatrix} = -2\lambda \det\begin{pmatrix} 125-\lambda & 295 \\ 59 & 145-\lambda \end{pmatrix}$$

$$= -2\lambda\underbrace{(\lambda^2 - 270\lambda + 720)}_{= \det(QQ^T - \lambda I) \text{ from a)}}$$

⇒ the nonzero eigenvalues are the same as those in part a

So, have eigenvalues of Q^TQ are $\begin{cases} \lambda_1 = 3(45+\sqrt{1945}) = 267.306 \\ \lambda_2 = 3(45-\sqrt{1945}) = 2.6935 \\ \lambda_3 = 0 \end{cases}$

* Find eigenvector of Q^TQ associated with $\lambda_1 = 3(45+\sqrt{1945})$.

the eigenvector $v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$ satisfies $Q^TQv = \lambda_1 v$

$$\Leftrightarrow (Q^TQ - \lambda_1 I)v = 0$$

$$\to \begin{pmatrix} -242.306 & 50 & 59 \\ 50 & -167.306 & 118 \\ 59 & 118 & -112.306 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \to \text{unit eigenvector } \begin{pmatrix} -0.304 \\ -0.608 \\ 0.733 \end{pmatrix}$$

* Find eigenvector of Q^TQ associated with $\lambda_2 = 3(45-\sqrt{1945})$

$$\begin{pmatrix} 22.3065 & 50 & 59 \\ 50 & 97.3065 & 118 \\ 59 & 118 & 142.3065 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \to \text{unit eigenvector } \begin{pmatrix} -0.3279 \\ -0.6559 \\ 0.6798 \end{pmatrix}$$

* Find eigenvector associated with $\lambda_3 = 0$

$$\begin{pmatrix} 25 & 50 & 59 \\ 50 & 100 & 118 \\ 59 & 118 & 145 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \longleftrightarrow \left(\begin{array}{ccc|c} 25 & 50 & 59 & 0 \\ 0 & 0 & 0 & 0 \\ 9 & 18 & 27 & 0 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 25 & 50 & 59 & 0 \\ 0 & 0 & 0 & 0 \\ 9 & 18 & 27 & 0 \end{array}\right)$$

$$\Rightarrow \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} -2v_2 \\ v_2 \\ 0 \end{pmatrix} \Rightarrow \text{corresponding unit vector } \begin{pmatrix} -2/\sqrt{5} \\ 1/\sqrt{5} \\ 0 \end{pmatrix}$$

c) Find SVD of Q

We have the singular values $\Rightarrow \sigma_1 = \sqrt{267.36} = 16.3511 \qquad \sigma_2 = \sqrt{2.6935} = 1.6411$

- We have the SVD of Q is $Q = U\Sigma V^T$ where U and V are orthogonal matrices
  Σ: diagonal matrix

So then:

① $QQ^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma \underbrace{V^T V}_{=I} \Sigma U^T$

$\Rightarrow u_i$ is eigen vector associative with eigenvalues of QQ^T

② $Q^TQ = (V\Sigma U^T)(U\Sigma V^T) = V\Sigma \underbrace{U^T U}_{I} \Sigma V^T = V\Sigma\Sigma V^T$

$\Rightarrow v_i$ is eigenvector associative with eigenvalue of Q^TQ.

$\Rightarrow$ We have the SVD of Q is:

$$Q = U\Sigma V^T \text{ where } U = \begin{pmatrix} -0.7307 & 0.6826 \\ -0.6826 & -0.7307 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 16.3511 & 0 \\ 0 & 1.6411 \end{pmatrix}$$

$$V = \begin{pmatrix} -0.304 & -0.3279 \\ -0.608 & -0.6559 \\ -0.733 & 0.6798 \end{pmatrix}$$

⇝ Consider arbitrary $n \times p$ matrix Q

a) Show that Q^TQ is a symmetric $p \times p$ matrix

b) Show that Q^TQ is necessarily nonnegative definite.

a)

* First, we have that $\underbrace{Q^T}_{p\times n}\underbrace{Q}_{n\times p}$ is a $p \times p$ matrix (square)

* Second, we have that:

Put $A = Q^TQ$

$\Rightarrow A^T = (Q^TQ)^T = Q^T Q^{TT} = Q^TQ = A$

$\Rightarrow A = Q^TQ$ is a square matrix and $A = A^T$

$\Rightarrow A$ is symmetric.

b) Since Q^TQ is symmetric, we have that $\underset{\sim}{x}^T Q^TQ \underset{\sim}{x}$ is well defined and we want to prove that $\underset{\sim}{x}^T Q^TQ \underset{\sim}{x} \geq 0$, $\forall \underset{\sim}{x} \neq \underset{\sim}{0}$, $\underset{p\times 1}{\underset{\sim}{x}}$

* Put $\underset{n\times 1}{\underset{\sim}{y}} = Q\underset{\sim}{x}$, then we have $\underset{\sim}{y}^T\underset{\sim}{y} = (Q\underset{\sim}{x})^T(Q\underset{\sim}{x}) = \underset{\sim}{x}^T Q Q \underset{\sim}{x}$ (1)

• Since $\underset{\sim}{y}^T\underset{\sim}{y} = \sum_{i=1}^{n} y_i^2 \overset{(2)}{\geq} 0$ when $\underset{\sim}{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$

$\updownarrow$

equality happens when $\underset{\sim}{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

$\updownarrow$

when $\underset{\sim}{x}$ is a solution of $Q\underset{\sim}{x} = \underset{\sim}{0}$

(1)+(2) $\Rightarrow \underset{\sim}{x}^T Q^TQ \underset{\sim}{x} \geq 0$, $\forall \underset{\sim}{x} \neq \underset{\sim}{0}$ $\Rightarrow Q^TQ$ is necessarily nonnegative definite.

$\updownarrow$

equality happens when $\underset{np}{Q}\underset{p}{\underset{\sim}{x}} = \underset{\sim}{0}_{np\times 1}$

MAT 755 HW4

Your names

Due: Tuesday, Oct. 23

1. Johnson & Wichern 8.10

```
stocks <- read.table("Wichern_data/T8-4.dat",
       col.names=c("JPM","Citibank","Wells","RoyalDutchShell","ExxonMobil"))
```

2. Johnson & Wichern 8.22

```
bulls<-read.table("Wichern_data/T1-10.dat")
names(bulls) <- c("Breed","SalePr","YrHgt","FtFrBody","PrctFFB","Frame","BkFat","SaleHt","SaleWt")
bulls<-bulls[,3:9]
```

3.

Let $\mathbf{X} \sim (\mu, \Sigma)$ where $\mu = [1, 2]^T$ and

$$\Sigma = \begin{bmatrix} 5/2 & 1 \\ 1 & 5/2 \end{bmatrix}$$

(a) Find the PCA factor loadings

(b) Find total variance and proportion of variance explained by each principal component

(c) Find the correlations between the principal components Y_i to the raw variables X_j

(d) Compute the principal components for the point $\mathbf{x} = [1 + \sqrt{2}, 2 - \sqrt{2}]^T$.

3) Let $X \sim (\mu, \Sigma)$ $\mu = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 5/2 & 1 \\ 1 & 5/2 \end{bmatrix}$

a) Find the PCA factor loadings

b) Find total variance and proportion of variance explained by each principal compone

c) Find the correlation between the principal component Y_i to the raw variables X_j

d) Compute the principal components for the point $x = \begin{bmatrix} 1+\sqrt{2} \\ 2-\sqrt{2} \end{bmatrix}$

a) * We first want to find eigenvalues λ_1, λ_2 and eigenvector $\vec{a_1}$ and $\vec{a_2}$ of Σ.

- Characteristic polynomial:

$$\det(\Sigma - \lambda I) = \det \begin{pmatrix} 5/2-\lambda & 1 \\ 1 & 5/2-\lambda \end{pmatrix} = (5/2-\lambda)^2 - 1 = \lambda^2 - 5\lambda + \frac{21}{4}$$

$$\det(\Sigma - \lambda I) = 0 \Rightarrow \lambda_1 = \frac{7}{2} \quad \lambda_2 = \frac{3}{2}$$

- Find eigenvector $\vec{a_1} = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix}$, $\vec{a_1}$ is a solution of $\Sigma a_1 = \lambda a_1 \Rightarrow (\Sigma - \lambda I)\vec{a_1} = \vec{0}$.

$$[\Sigma - \lambda_1 I \,;\, \underset{\sim}{0}] \Leftrightarrow \left[\begin{array}{cc|c} 5/2-7/2 & 1 & 0 \\ 1 & 5/2-7/2 & 0 \end{array}\right] \Longrightarrow \left[\begin{array}{cc|c} -1 & 1 & 0 \\ 1 & -1 & 0 \end{array}\right] \rightarrow \vec{a_1} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

- Find eigenvector $\vec{a_2}$ associative with $\lambda_2 = 3/2$

$$[\Sigma - \lambda_2 I \,;\, 0] = \left[\begin{array}{cc|c} 5/2-3/2 & 1 & 0 \\ 1 & 5/2-3/2 & 0 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 1 & 0 \\ 1 & 1 & 0 \end{array}\right] \Rightarrow \vec{a_2} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

⇒ The PCA factor loadings are $\vec{a_1} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$ and $\vec{a_2} = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$

b)

* The total variance $\lambda_1 + \lambda_2 = \frac{7}{2} + \frac{3}{2} = 5$

* The proportion of variance explained by the first principal component, W_1 is

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{\frac{7}{2}}{\frac{10}{2}} = \frac{7}{10}$$

* The proportion of variance explain by the second pc, W_2 is

$$\frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{\frac{3}{2}}{\frac{10}{2}} = \frac{3}{10}$$

c) Find the correlation between the principal component Y_i and raw variable X_j

We have $\begin{bmatrix} \vec{a_1}^T \\ \vec{a_2}^T \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$ $\lambda_1 = \frac{7}{2}$ $\lambda_2 = \frac{3}{2}$ $\Sigma = \begin{bmatrix} 5/2 & 1 \\ 1 & 5/2 \end{bmatrix}$.

Then

- $\rho_{Y_1,X_1} = \frac{a_{11}\sqrt{\lambda_1}}{\sqrt{\sigma_{11}}} = a_{11} = \frac{1/\sqrt{2}\sqrt{\frac{7}{2}}}{\sqrt{5/2}} = \frac{\sqrt{7}}{\sqrt{10}}$
- $\rho_{Y_1,X_2} = \frac{a_{12}\sqrt{\lambda_1}}{\sqrt{\sigma_{22}}} = \frac{1/\sqrt{2}\sqrt{\frac{7}{2}}}{\sqrt{\frac{5}{2}}} = \frac{\sqrt{7}}{\sqrt{10}}$
- $\rho_{Y_2,X_1} = \frac{a_{21}\sqrt{\lambda_2}}{\sqrt{\sigma_{11}}} = \frac{\frac{1}{\sqrt{2}}\sqrt{\frac{3}{2}}}{\sqrt{\frac{5}{2}}} = \frac{\sqrt{3}}{\sqrt{10}}$

$\rho_{Y_2,X_2} = \frac{a_{22}\sqrt{\lambda_2}}{\sqrt{\sigma_{22}}} = \frac{-\frac{1}{\sqrt{2}}\sqrt{\frac{3}{2}}}{\sqrt{\frac{5}{2}}} = -\frac{\sqrt{3}}{\sqrt{10}}$

d) Compute the principal component for the point $x = \begin{bmatrix} 1+\sqrt{2} \\ 2-\sqrt{2} \end{bmatrix}$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1+\sqrt{2} \\ 2-\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{2} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

5.1, 5.3, 5.4 Mat682
Matlab: Verify that 5.1, 5.3 a,c,d,e HW3
are correct. Tran Le

5.17 In example 3.1, we consider the matrix $A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$ and asserted, among other things, that its 2 norm is approximately 2.9208. Using the SVD, work out (on paper) the exact values of $\sigma_{min}(A)$ and $\sigma_{max}(A)$ for this matrix

* Let r = # of nonzero singular value of A
then since $\det A \neq 0 \Rightarrow r = \text{rank } A = 2$

* By a property of SVD, nonzero singular values of A = square root of nonzero eigenvalues of A^*A

• Consider $A^*A = \begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix}$

then characteristic function

$$\det\begin{bmatrix} 1-\lambda & 2 \\ 2 & 8-\lambda \end{bmatrix} = (1-\lambda)(8-\lambda) - 4 = \lambda^2 - 9\lambda + 4$$

$\Rightarrow$ Let $\lambda^2 - 9\lambda + 4 = 0$ then $\lambda_{1,2} = \dfrac{9 \pm \sqrt{81-16}}{2} = \dfrac{9 \pm \sqrt{65}}{2} > 0$

Then since $r = 2$, there 2 nonsingular values of A are

$$\sigma_{max}(A) = \sqrt{\frac{9+\sqrt{65}}{2}} = 2.9208$$

$$\sigma_{min}(A) = \sqrt{\frac{9-\sqrt{65}}{2}} = 0.6847$$

✓

5.37 Consider the matrix $A=\begin{bmatrix}-2 & 11\\ -10 & 5\end{bmatrix}$

a) Determine, on paper, a real SVD of A in the form $A=U\Sigma V^T$, find the one that has the minimal number of minus signs in U and V

* We will find U, V by $\begin{cases} AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma U^T & (1) \Rightarrow \lambda_1^2 u_1 + \lambda^2 u_2 \\ AV = U\Sigma \Rightarrow V = A^{-1}U\Sigma & (2) \end{cases}$

u_i is eigenvector associative with eigenvector of AA^T = (singular value of A)^2

★ Find U:

$$AA^T=\begin{bmatrix}-2 & 11\\ -10 & 5\end{bmatrix}\begin{bmatrix}-2 & -10\\ 11 & 5\end{bmatrix}=\begin{bmatrix}125 & 75\\ 75 & 125\end{bmatrix}$$

$$\det[AA^T-\lambda I]=\det\begin{bmatrix}125-\lambda & 75\\ 75 & 125-\lambda\end{bmatrix}=(125-\lambda)^2-75^2$$

Let $\det[AA^T-\lambda I]=0$, then $\lambda_1=200$ $\lambda_2=125-75=50$

By a theorem, the nonzero singular values of A = square root eigenvalues of AA^T

$\Rightarrow \sigma_1=\sqrt{\lambda_1}=\sqrt{200}=10\sqrt{2}$

$\sigma_2=\sqrt{\lambda_2}=\sqrt{50}=5\sqrt{2}$

$$\Rightarrow \boxed{\Sigma=\begin{bmatrix}10\sqrt{2} & 0\\ 0 & 5\sqrt{2}\end{bmatrix}}=\begin{bmatrix}14.142 & 0\\ 0 & 7.0710\end{bmatrix}$$

• Now find the eigenvector of AA^T associated with $\lambda_1=200$

$$[AA^T-\lambda_1 I]=\begin{bmatrix}-75 & 75\\ 75 & -75\end{bmatrix}\rightarrow\left[\begin{array}{cc|c}-1 & 1 & 0\\ 1 & -1 & 0\end{array}\right]\Rightarrow \vec{u}_1=\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix}$$

• Now find the eigenvector of AA^T associated with $\lambda_2=50$

$$[AA^T-\lambda_2 I]=\begin{bmatrix}75 & 75\\ 75 & 75\end{bmatrix}\Rightarrow \vec{u}_2=\begin{bmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{bmatrix}$$

$$\Rightarrow \boxed{U=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{bmatrix}}$$

★ Now we find V by (2)

$$\vec{v}_1=\sigma_1 A^{-1}\vec{u}_1=10\sqrt{2}\,\frac{1}{100}\begin{bmatrix}5 & -11\\ 10 & -2\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix}=\frac{10\sqrt{2}}{100}\begin{bmatrix}\frac{5-11}{\sqrt{2}}\\ \frac{10-2}{\sqrt{2}}\end{bmatrix}=\begin{bmatrix}\frac{-6}{10}\\ \frac{8}{10}\end{bmatrix}=\begin{bmatrix}-\frac{3}{5}\\ \frac{4}{5}\end{bmatrix}$$

$$\vec{v}_2=\sigma_2 A^{-1}\vec{u}_2=5\sqrt{2}\,\frac{1}{100}\begin{bmatrix}5 & -11\\ 10 & -2\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{bmatrix}=\frac{5\sqrt{2}}{100}\begin{bmatrix}\frac{5+11}{\sqrt{2}}\\ \frac{+10+2}{\sqrt{2}}\end{bmatrix}=\begin{bmatrix}\frac{+16}{20}\\ \frac{12}{20}\end{bmatrix}=\begin{bmatrix}+\frac{4}{5}\\ \frac{3}{5}\end{bmatrix}$$

$$\boxed{V=\begin{bmatrix}-\frac{3}{5} & \frac{4}{5}\\ \frac{4}{5} & \frac{3}{5}\end{bmatrix}}$$

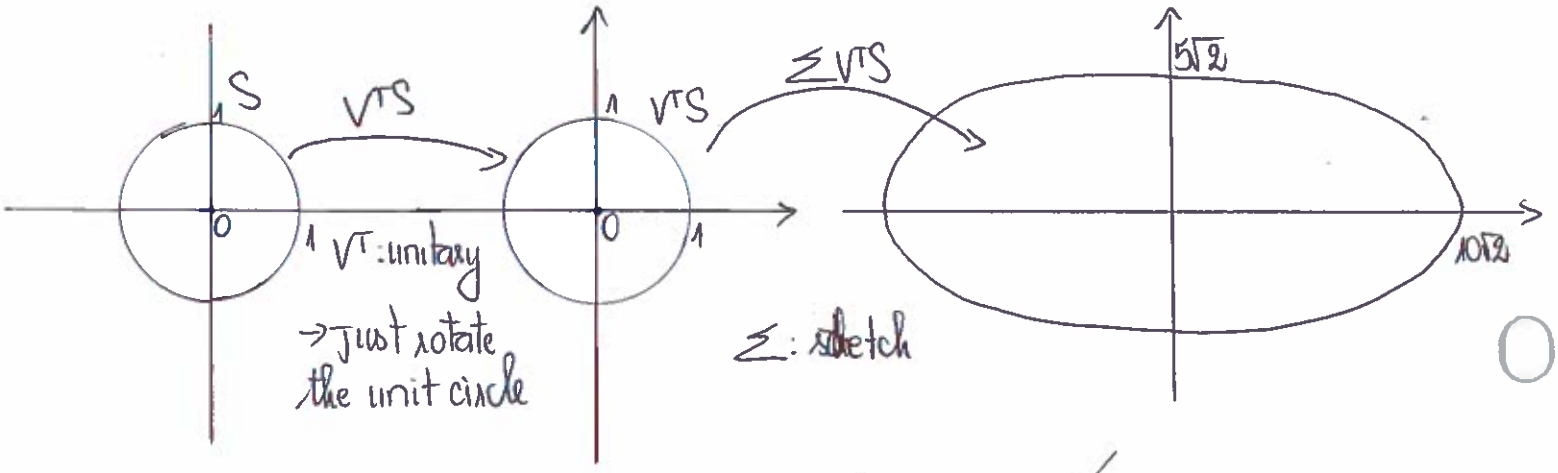
b) List the singular values, left singular vectors and right singular vectors of A
Draw a careful, labeled picture of the unit ball in $\mathbb{R}^2$ and its image under A, together with the singular vectors, with the coordinates of their vertices marked.

From a) singular values : $10\sqrt{2}$, $5\sqrt{2}$

left singular vectors : $\begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$ $\begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$ $\quad A = U\Sigma V^T$

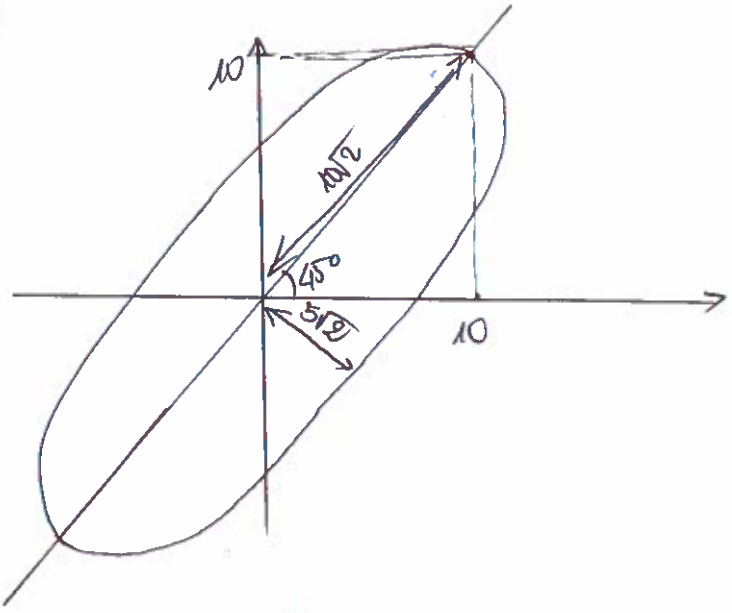
right singular vectors : $\begin{pmatrix} -3/5 \\ 4/5 \end{pmatrix}$ $\begin{pmatrix} 4/5 \\ 3/5 \end{pmatrix}$ $\quad \Sigma$

So when we put S = unit ball in $\mathbb{R}^2$. $\quad AS = U\Sigma V^T S$



$U = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$,

U : rotate $\frac{\pi}{4} = 45°$ counterclockwise



✓

c) What are the 1-, 2-, ∞-, Frobenius norms of A.

$\|A\|_1 = \max_{1\le j\le 2} \|a_j\|_1 = \max\{12, 16\} = 16$ ✓

$\|A\|_2 = \|\Sigma\|_2 = \sigma_1 = 10\sqrt{2} = 14.142$ ✓

$\|A\|_\infty = \max_{1\le j\le 2} \{\|a_j^*\|_1\} = \max\{13, 15\} = 15$ ✓

$\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{200+50} = \sqrt{250} = 5\sqrt{10} = 15.8114$ ✓

d) Find A^{-1} via the SVD

$A = U\Sigma V^T$

$A^{-1} = (U\Sigma V^T)^{-1} = (V^T)^{-1}\Sigma^{-1}U^{-1} = V\Sigma^{-1}U^T = \frac{1}{\sqrt{2}}\begin{bmatrix} -\frac{3}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{bmatrix}\begin{bmatrix} \frac{1}{10\sqrt{2}} & 0 \\ 0 & \frac{1}{5\sqrt{2}} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$= \begin{bmatrix} -\frac{3}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{bmatrix}\begin{bmatrix} \frac{1}{20} & 0 \\ 0 & \frac{1}{10} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} -\frac{3}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{bmatrix}\begin{bmatrix} \frac{1}{20} & \frac{1}{20} \\ \frac{1}{10} & -\frac{1}{10} \end{bmatrix}$

$= \begin{bmatrix} \frac{1}{20} & -\frac{11}{100} \\ \frac{1}{10} & -\frac{1}{50} \end{bmatrix} = \begin{bmatrix} 0.05 & -0.11 \\ 0.1 & -0.02 \end{bmatrix}$ ✓

e) Find the eigenvalues of A

$\det(A-\lambda I) = \det\begin{pmatrix} -2-\lambda & 11 \\ -10 & 5-\lambda \end{pmatrix} = -(2+\lambda)(5-\lambda) + 110$

$= \lambda^2 - 3\lambda + 100$

$\lambda_{1,2} = \frac{3 \pm \sqrt{9-400}}{2} = \frac{3 \pm \sqrt{391}\,i}{2}$ ✓

$\lambda_1 = 1.5 + 9.8869\,i$

$\lambda_2 = 1.5 - 9.8869\,i$

f) Verify that $\det A = \lambda_1\lambda_2$ $\quad |\det A| = \sigma_1\sigma_2$

$\det A = -10 + 110 = 100$ ✓

from quadratic equation $\lambda^2 - 3\lambda + 100 \Rightarrow \lambda_1\lambda_2 = 100$

$\sigma_1\sigma_2 = 10\sqrt{2}\cdot 5\sqrt{2} = 100$ ✓

g) Area of the ellipsoid $= \pi\sigma_1\sigma_2 = 100\pi$. ✓

*5.47 Suppose $A \in \mathbb{C}^{m\times m}$ has an SVD $A = U\Sigma V^*$

Find an eigenvalue decomposition $H = X\Lambda X^{-1}$ of the $2m \times 2m$ Hermitian matrix $H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}$

* Note that since $A = U\Sigma V^* \Rightarrow AV = U\Sigma$ ✓

$A^* = V\Sigma^* U^* \Rightarrow A^*U = V\Sigma^*$

So we have

$$\left(\begin{array}{c|c} 0 & A \\ \hline A^* & 0 \end{array}\right)\left(\begin{array}{c|c} -U & U \\ \hline V & V \end{array}\right) = \left(\begin{array}{c|c} AV & AV \\ \hline -A^*U & A^*U \end{array}\right) = \left(\begin{array}{c|c} U\Sigma & U\Sigma \\ \hline -V\Sigma^* & V\Sigma^* \end{array}\right) =$$

This is NOT H

$$= \left(\begin{array}{c|c} -U & U \\ \hline V & V \end{array}\right)\left(\begin{array}{c|c} -\Sigma & 0 \\ \hline 0 & \Sigma \end{array}\right)$$ ✓

This means we have an eigenvalue decomposition for H is $H = X\Lambda X^{-1}$

where $X = \left(\begin{array}{c|c} -U & U \\ \hline V & V \end{array}\right)$ $\Lambda = \left(\begin{array}{c|c} -\Sigma & 0 \\ \hline 0 & \Sigma \end{array}\right)$ ✓

* Note that (incase $A \in \mathbb{R}^{n\times m} \Rightarrow H$ is symmetric). ?

* We can also have an eigenvalue decomposition where X is orthonormal. $H = Y\Lambda Y^{-1}$, Y

$Y = \frac{1}{\sqrt{2}} X$

9.5/10

5.4 Suppose $A \in \mathbb{C}^{m \times m}$, has an SVD $A = U \Sigma V^*$

Find an eigenvalue decomposition $H = X \Lambda X^{-1}$ of the $2m \times 2m$ Hermitian matrix

$$H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}$$

* We want to find eigenvalues and associative eigenvectors of H, which means we want to find λ and X such that:

$$\underset{2m \times 2m}{H} \underset{2m \times 1}{X} = \lambda \underset{2m \times 1}{X} \quad (2m \times 1)$$

$$\Leftrightarrow \left[\begin{array}{c|c} 0_{m \times m} & A^*_{m \times m} \\ \hline A_{m \times m} & 0_{m \times m} \end{array}\right] \left[\begin{array}{c} \vec{x}_{1\, m \times 1} \\ \hline \vec{x}_{2\, m \times 1} \end{array}\right] = \lambda \left[\begin{array}{c} \vec{x}_1 \\ \hline \vec{x}_2 \end{array}\right]_{m \times 1}$$

$$X = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix}$$

$$\Leftrightarrow \left[\begin{array}{c} A^* \vec{x}_2 \\ \hline A \vec{x}_1 \end{array}\right] = \left[\begin{array}{c} \lambda \vec{x}_1 \\ \hline \lambda \vec{x}_2 \end{array}\right] \quad \begin{matrix} (1) \\ (2) \end{matrix}$$

- $(1) \Rightarrow A A^* \vec{x}_2 = \lambda A \vec{x}_1 \underset{(2)}{=\!=} \lambda^2 \vec{x}_2$

  $\Rightarrow \vec{x}_2$ is an left singular vector of A associative with singular value λ (since $\|\vec{x}_2\| = 1$)

- $(2) \Rightarrow A^* A \vec{x}_1 = \lambda A^* \vec{x}_2 \underset{(1)}{=\!=} \lambda^2 \vec{x}_1$

  $\Rightarrow \vec{x}_1$ is a left singular vector of A associative with singular value λ (since $\|\vec{x}_1\| = 1$).

Thus we see that:

```
%----------------------------------------
%Verify that my answer for 5.1 is correct
%----------------------------------------
A=[1 2;0 2 ]
[U,S,V]=svd(A)
sigma_max=S(1,1)
sigma_min=S(2,2)
%result
% A =
%      1     2
%      0     2
% U =
%     0.7497   -0.6618
%     0.6618    0.7497
%
% S =
%     2.9208         0
%          0    0.6847
% V =
%     0.2567   -0.9665
%     0.9665    0.2567
%
% sigma_max =
%     2.9208
% sigma_min =
%     0.6847
%-------------------------------------------------------
%Verify that my aswer for 5.3 a,c,d,and e are correct
%-------------------------------------------------------
A=[-2 11; -10 5]
[U,S,V]=svd(A)
%5.3a-------------------------------------
% A =
%     -2    11
%    -10     5
%
% U =
%    -0.7071   -0.7071
%    -0.7071    0.7071
%
% S =
%    14.1421         0
%          0    7.0711
% V =
%     0.6000   -0.8000
%    -0.8000   -0.6000

%5.3c-------------------------------------
norm(A,1)
norm(A,2)
norm(A,inf)
```

'Testdigit5.mat'

trainingset azip
testset testzip
dtest ← correct result.

* Compare:
mismatch_index25 = find(dtest ~= TestDigit 25)
(2007 - 122)/2007 = 0.9392 ← % success = 93%.

5subspace ⇒ % success = 90%.

* Consider unsuccessful cases.
• sampleId = randi([1 2007], 100, 1) —

plot(sampleId, dtest(sampleId), 0)
hold on
plot(sampleId, testdigit25(sampleId), 1x)

image(testzip(:,881))

```
norm(A,'fro')
% ans =
%     16
%
% ans =
%    14.1421
% ans =
%     15
% ans =
%    15.8114

%5.3d-----------------------------------------------------------
Ainverse=inv(A)
% Ainverse =
%     0.0500   -0.1100
%     0.1000   -0.0200
%5.3e-----------------------------------------------------------
[lambda]=eig(A)
lambda1=lambda(1)
lambda2=lambda(2)
% lambda =
%    1.5000 + 9.8869i
%    1.5000 - 9.8869i
%
% lambda1 =
%    1.5000 + 9.8869i
%
% lambda2 =
%    1.5000 - 9.8869i
```

MAT755_HW3

48/50

Tran Le

September 29, 2018

Johnson and Wichen 5.1

a. Evaluate T^2, for testing $\mu^T = [1 \quad 2]$, using the data

$$X = \begin{bmatrix} 2 & 12 \\ 8 & 9 \\ 6 & 9 \\ 8 & 10 \end{bmatrix}$$

```
rm(list = ls())
X=matrix(c(2,12,8,9,6,9,8,10), nrow=4, ncol=2, byrow=TRUE)
n <- dim(X)[1] # number of row of X (#observations)
p <- dim(X)[2] # number of columns (#variables)

Xbar<-apply(X,2,mean)
Xcov <- cov(X)

mu.o<-c(7, 11)
T2 <- n * mahalanobis(x = Xbar, center = mu.o, cov = Xcov)
T2

## [1] 13.63636
```

b. Specific the distribution of T^2 for the situation in (a).

We have that T^2 has $((n-1)*p)/(n-p)F_{p,n-p,\alpha}$

c. Using (a) and (b), test H_0 at the $\alpha = 0.05$. What conclusion do you reach?

From (a) and (b), we want to compare the value of $T_{observed}$ and the critical value

```
alpha <- 0.05
crit.val <- p * (n-1) / (n-p) * qf(1-alpha, p, n-p)
crit.val

## [1] 57
```

So we have that $T_{observed} <$ critical value, fail to reject H_0. In conclusion, at level $\alpha = 0.05$, we fail to reject that $\mu^T = [1 \quad 2]$.

2. Johnson & Wichern 5.18a

Use the college test data in Table 5.2 (see example 5.5). At the level $\alpha = 0.05$. Test the hypotheses

$$\begin{cases} H_0: \mu^T = [500,50,30] \\ H_1: \mu^T \neq [500,50,30] \end{cases}$$

```
getwd() #get working directory
```

```
## [1] "C:/Users/tranl/Desktop/MAT755/Homework/HW3"
```

```
college <- read.table("C:/Users/tranl/Desktop/MAT755/Homework/HW3/T5-2.dat")
head(college)
```

```
##    V1 V2 V3
## 1 468 41 26
## 2 428 39 26
## 3 514 53 21
## 4 547 67 33
## 5 614 61 27
## 6 501 67 29
```

```
n <- dim(college)[1] # number of observations
p <- dim(college)[2] # number of variables

alpha <- 0.05
col.xbar <- sapply(college, mean)
col.cov <- cov(college)


mu0 <- c(500, 50, 30)

T2 <- n * mahalanobis(x = col.xbar, center = mu0, cov =col.cov)
T2
```

```
## [1] 223.3102
```

```
crit.val <- p * (n-1) / (n-p) * qf(1-alpha, p, n-p)
crit.val
```

```
## [1] 8.333483
```

So we have that T^2>crit.val, we reject H_0, at level $\alpha = 0.05$, it is resonable to belive that the group of students represented by the scores in Table 5.2 is scoring differently than the average scores for thousands of collge students over the last 10 years.

I would like to minic your code to do the sample Hotelling T2-test using HotellingT2

```
library(ICSNP)
library(ellipse)
```

```
col.T2<-HotellingsT2(X=college, mu=mu0)
col.T2

##
##  Hotelling's one sample T2-test
##
## data:  college
## T.2 = 72.706, df1 = 3, df2 = 84, p-value < 2.2e-16
## alternative hypothesis: true location is not equal to c(500,50,30)
```

So we reject H_0

3. Johnson & Wichern 5.19 (lumber T5-11.dat)

Measurements of x_1=stiffness and x_2=beding strength for a sample of n=30 pieces of a particular grade of lumber are given in Table 5.11. The units are pounds/(inches)^2. Using the data in the table,

a. Construct and sketch a 95% confidence ellise for the pair $[\mu_1, \mu_2]^T$ where $\mu_1 = E(X_1)$ and $\mu_2 = E(X_2)$.

```
#import the data
lumber <- read.table(file="C:/Users/tranl/Desktop/MAT755/Homework/HW3/T5-
11.dat")
names(lumber) <- c("stiff", "bend")
head(lumber)

##   stiff  bend
## 1  1232  4175
## 2  1115  6652
## 3  2205  7612
## 4  1897 10914
## 5  1932 10850
## 6  1612  7627

lum.xbar<-sapply(lumber, mean)
lum.xbar

##    stiff     bend
## 1860.500 8354.133

lum.xbar[-1]

##     bend
## 8354.133

lum.cov<-cov(lumber)
lum.cov
```

```
##          stiff      bend
## stiff 124054.7  361620.4
## bend  361620.4 3486333.2
```

```
lum.cov[-1,-1]
```

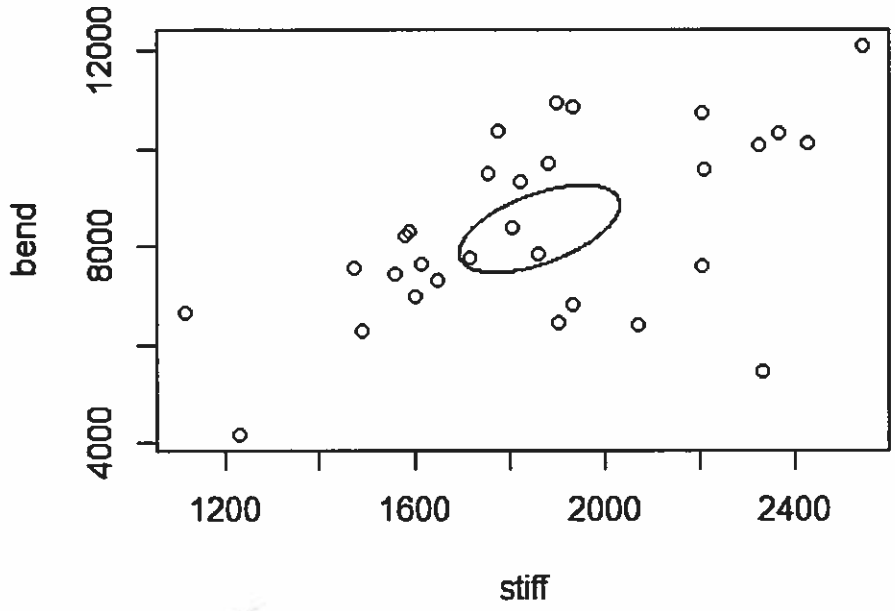
```
## [1] 3486333
```

```
alpha=0.05
ci.T2 <- rbind(lum.xbar - sqrt(p*(n-1)/(n-p)/n*qf(1-alpha, p, n-
p)*diag(lum.cov)),
               lum.xbar + sqrt(p*(n-1)/(n-p)/n*qf(1-alpha, p, n-
p)*diag(lum.cov)))
row.names(ci.T2)<-c("T2_L", "T2_U")
ci.T2
```

```
##         stiff     bend
## T2_L 1751.492 7776.253
## T2_U 1969.508 8932.014
```

Now we want to sketch the 95% confidence ellipse

```
Ellipse <- function(covmat, centroid, csquare, resolution, plot = TRUE) {
angles <- seq(0, by = (2 * pi)/resolution, length = resolution)
sd <- covmat[1,2] / sqrt(covmat[1,1] * covmat[2,2])
projmat <- matrix(0,2,2)
projmat[1,1] <- sqrt(covmat[1,1] %*% (1+sd)/2)
projmat[1,2] <- -sqrt(covmat[1,1] %*% (1-sd)/2)
projmat[2,1] <- sqrt(covmat[2,2] %*% (1+sd)/2)
projmat[2,2] <- sqrt(covmat[2,2] %*% (1-sd)/2)
circle <- cbind(cos(angles), sin(angles))
Ellipse <- t(centroid + sqrt(csquare) * projmat %*% t(circle))
if (plot == TRUE) {lines(Ellipse)}
return(Ellipse)
}

cdellipse <- function (data, alpha=0.05, resolution=500)
{
xbar <- colMeans(data)
n <- dim(data)[1]
p <- dim(data)[2]
f <- qf(1-alpha, p, n-p)
csquare <- ((n-1)/n) * (p / (n-p)) * f
cat(csquare)
Ellipse <- Ellipse(cov(data), xbar, csquare, resolution)
}
old.par <- par(no.readonly = TRUE)
plot(lumber)
cdellipse(lumber, alpha = 0.05)
```




```
## 0.2306457
```

b. Suppose $\mu_{10} = 2000$ and $\mu_{20} = 10000$ represent "typical values" for stiffness bending strengh, respectively. Given the data in table 5.11 consistent with these values? Explain.

We have the given data is not consistent with these values since these values are not belong to the above 95% confidence ellipse.

c. ?

4. J& W 6.24

Four measurements were made of male Egytian skulls for there different time periods. The measured variables are:

X_1=maximum breath of skull (mm)

X_2=basibregmatic heiht of skull (mm)

X_3=basilaveplar length of skull (mm)

X_4=nasal height of skull (mm)

a. Construct a one way MANOVA of the Egyptian skull data.

```
egypt <- read.table(file="C:/Users/tranl/Desktop/MAT755/Homework/HW3/T6-
13.dat")
names(egypt) <- c("MaxBreath", "BasHeight", "BasLength",
```

```
"NasHeight","TimePeriod")
head(egypt)

##   MaxBreath BasHeight BasLength NasHeight TimePeriod
## 1       131       138        89        49          1
## 2       125       131        92        48          1
## 3       131       132        99        50          1
## 4       119       132        96        44          1
## 5       136       143       100        54          1
## 6       138       137        89        56          1

attach(egypt)
dim(egypt)

## [1] 90  5

egypt$TimePeriod<-as.factor(egypt$TimePeriod)
egypt.manova<-
manova(cbind(MaxBreath,BasHeight,BasLength,NasHeight)~TimePeriod)
summary(egypt.manova, test="Wilks")

##            Df   Wilks approx F num Df den Df  Pr(>F)
## TimePeriod  1 0.86076   3.4374      4     85 0.01182 *
## Residuals  88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

by(egypt[,-5], egypt$TimePeriod, summary)

## egypt$TimePeriod: 1
##    MaxBreath       BasHeight       BasLength       NasHeight
##  Min.   :119.0   Min.   :121.0   Min.   : 89.00   Min.   :44.00
##  1st Qu.:128.0   1st Qu.:131.2   1st Qu.: 95.00   1st Qu.:49.00
##  Median :131.0   Median :134.0   Median :100.00   Median :50.00
##  Mean   :131.4   Mean   :133.6   Mean   : 99.17   Mean   :50.53
##  3rd Qu.:134.8   3rd Qu.:136.0   3rd Qu.:102.75   3rd Qu.:53.00
##  Max.   :141.0   Max.   :143.0   Max.   :114.00   Max.   :56.00
## ------------------------------------------------------------
## egypt$TimePeriod: 2
##    MaxBreath       BasHeight       BasLength       NasHeight
##  Min.   :123.0   Min.   :124.0   Min.   : 90.00   Min.   :45.00
##  1st Qu.:130.0   1st Qu.:129.2   1st Qu.: 97.00   1st Qu.:48.00
##  Median :132.0   Median :133.0   Median : 98.50   Median :50.50
##  Mean   :132.4   Mean   :132.7   Mean   : 99.07   Mean   :50.23
##  3rd Qu.:134.8   3rd Qu.:136.0   3rd Qu.:101.75   3rd Qu.:52.75
##  Max.   :148.0   Max.   :145.0   Max.   :107.00   Max.   :56.00
## ------------------------------------------------------------
## egypt$TimePeriod: 3
##    MaxBreath       BasHeight       BasLength       NasHeight
##  Min.   :126.0   Min.   :123.0   Min.   : 87.00   Min.   :45.00
##  1st Qu.:132.2   1st Qu.:131.0   1st Qu.: 92.25   1st Qu.:48.25
```

interpret ...

```
##  Median :136.0   Median :133.5   Median : 96.00   Median :50.00
##  Mean   :134.5   Mean   :133.8   Mean   : 96.03   Mean   :50.57
##  3rd Qu.:137.0   3rd Qu.:137.0   3rd Qu.: 99.75   3rd Qu.:52.75
##  Max.   :140.0   Max.   :145.0   Max.   :106.00   Max.   :60.00
```

b. Construct 95% simultaneous confidence intervals to determine which mean components differ among the populations represented by the three time periods.

```
my.n <- nrow(egypt[,-5])
W <- (my.n - 1)*var(residuals(egypt.manova))
W
```

```
##           MaxBreath BasHeight  BasLength NasHeight
## MaxBreath 1791.4500  183.5000  112.83333 293.11667
## BasHeight  183.5000 1944.3000  149.46667 178.23333
## BasLength  112.8333  149.4667 2196.02222 -10.98889
## NasHeight  293.1167  178.2333  -10.98889 842.20556
```

```
my.alpha<-0.05 # family confidence interval  here
my.m <- 3 # number of groups
my.q <- 4 # number of variables

group.sample.sizes <- c(30,30,30)
 for (k in 1:my.q) {
 pair.mean.diffs <- cbind( t(combn(my.m,2)),
 combn(tapply(egypt[,k],TimePeriod,mean),2,FUN=diff) )
 t.val <- qt(my.alpha/(my.q*my.m*(my.m-1)), df=my.n-my.m, lower=F)

 CI.L <- pair.mean.diffs[,3] -
 t.val*sqrt((diag(W)[k]/(my.n-my.m))*
 (1/group.sample.sizes[pair.mean.diffs[,1]] +
 1/group.sample.sizes[pair.mean.diffs[,2]]) )
 CI.U <- pair.mean.diffs[,3] +
 t.val*sqrt((diag(W)[k]/(my.n-my.m))*
 (1/group.sample.sizes[pair.mean.diffs[,1]] +
 1/group.sample.sizes[pair.mean.diffs[,2]]) )
 my.table.mat<-cbind(pair.mean.diffs,round(CI.L,3),round(CI.U,3),
 rep(k,times=nrow(pair.mean.diffs)) )

 my.table<-as.data.frame(my.table.mat)
 names(my.table)=c('grp1','grp2','diff.samp.means',
 'lower.CI','upper.CI','variable');
 print(my.table)
}
```

```
##   grp1 grp2 diff.samp.means lower.CI upper.CI variable
## 1    1    2             1.0   -2.448    4.448        1
## 2    1    3             3.1   -0.348    6.548        1
## 3    2    3             2.1   -1.348    5.548        1
##   grp1 grp2 diff.samp.means lower.CI upper.CI variable
## 1    1    2            -0.9   -4.492    2.692        2
```

```
## 2    1    3            0.2   -3.392    3.792        2
## 3    2    3            1.1   -2.492    4.692        2
##   grp1 grp2 diff.samp.means lower.CI upper.CI variable
## 1    1    2       -0.100000   -3.918    3.718        3
## 2    1    3       -3.133333   -6.951    0.684        3
## 3    2    3       -3.033333   -6.851    0.784        3
##   grp1 grp2 diff.samp.means lower.CI upper.CI variable
## 1    1    2     -0.30000000   -2.664    2.064        4
## 2    1    3      0.03333333   -2.331    2.398        4
## 3    2    3      0.33333333   -2.031    2.698        4
```

c. Are the usual MANOVA asumptions realistic for these data?

I would like to compare the results we get when using MANOVA from part (a) and using 95% Bonferroni Confidence interval.

First, comparing the mean size of Maxbearth changes thoughout periods of time, we see that the mean Maxbearth size was increasing. Then looking at the result when we use Bonferroni confidence interval, we see that the confidence interval for mean from period 1 to period 2 is [-2.448, 4.448], we see that there is a chance that 0 belongs to this interval, however, there is more change that the different is a positive number, which means there was an increasing of Maxbreath mean size from period 1 to period 2.

Similarly, we see that there was a slightly change in the skull size during these periods of time. Further investments are needed to be done to have futher and more presice conclusion.

Test MANOVA assumption...

5. Find a dataset online which contains (i) two factors and (ii) at least two continuous responses.

The data (see Micheal and Johnson problem 7.25)

Amitriptyline is prescribed by some physicians as an antidepressant. However, there are also conjectured side effects that seem to be related to the use of the drug: irregular heartbeat, abnormal blood pressures, and irregular waves on the electrocardiogram among other things. Data gathered on 17 patients who were admitted to the hospitai after an amitriptyline overdose are given in Table 7.6. The two response variables are

Y1 = Total TCAD plasma leyel (TOT)

Y2 = Amount of amitriptyline present in TCAD plasma level(A.MI)

The five predictor variables are

Z1 = Gender: 1 iffemale,Oifmale (GEN)

Z2 = Amount of antidepressants taken at time of overdose (AMT)

Z3 = PR wave measurement (PR)

Z4 = Diastolic blood pressure (DIAP)

Z5 = QRSwavemeasurement(QRS)

(a). Run a one-way MANOVA using only one factor. State your conclusions in plain language.

```
plasma <- read.table(file="C:/Users/tranl/Desktop/MAT755/Homework/HW3/T7-
6.dat")
names(plasma) <- c("TOT", "AMI", "GEN", "AMT","PR","DIAP","QRS")
attach(plasma)
head(plasma)
```

```
##    TOT  AMI GEN  AMT  PR DIAP QRS
## 1 3389 3149   1 7500 220    0 140
## 2 1101  653   1 1975 200    0 100
## 3 1131  810   0 3600 205   60 111
## 4  596  448   1  675 160   60 120
## 5  896  844   1  750 185   70  83
## 6 1767 1450   1 2500 180   60  80
```

```
dim(plasma)
```

```
## [1] 17  7
```

```
plasma.manova <- manova(cbind(TOT, AMI) ~ QRS)
summary(plasma.manova, test="Wilks")
```

```
##           Df   Wilks approx F num Df den Df  Pr(>F)
## QRS        1 0.62363   4.2247      2     14 0.03668 *
## Residuals 15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b). Run a two-way MANOVA with interactions. State your conclusions in plain language. Refit a two-way MANOVA without interactions if the interactions are not significant. State your conclusions in plain language.

```
fit.inter <- manova(cbind(TOT, AMI)~ PR * QRS)
#summary.aov(fit.inter) # univariate ANOVA tables
summary(fit.inter, test="Wilks")
```

```
##           Df   Wilks approx F num Df den Df   Pr(>F)
## PR         1 0.36468  10.4529      2     12 0.002352 **
## QRS        1 0.66291   3.0510      2     12 0.084863 .
## PR:QRS     1 0.57127   4.5028      2     12 0.034759 *
## Residuals 13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With level $\alpha = 0.05$, we see that the interaction is significant. The factor QSR alone is not siginicant, which means QRS wave measurement does not affect Total TCAD plasma leyel

(TOT) and amount of amitriptyline present in TCAD plasma level(A.MI). However, the interaction of QRS wave measurement and PR wave measurement does affect Total TCAD plasma leyel (TOT) and amount of amitriptyline present in TCAD plasma level(A.MI).

(c). One assumption we made in MANOVA is that the variance-covariance matrices of each group of residuals are equal. Does this assumption hold in your dataset?

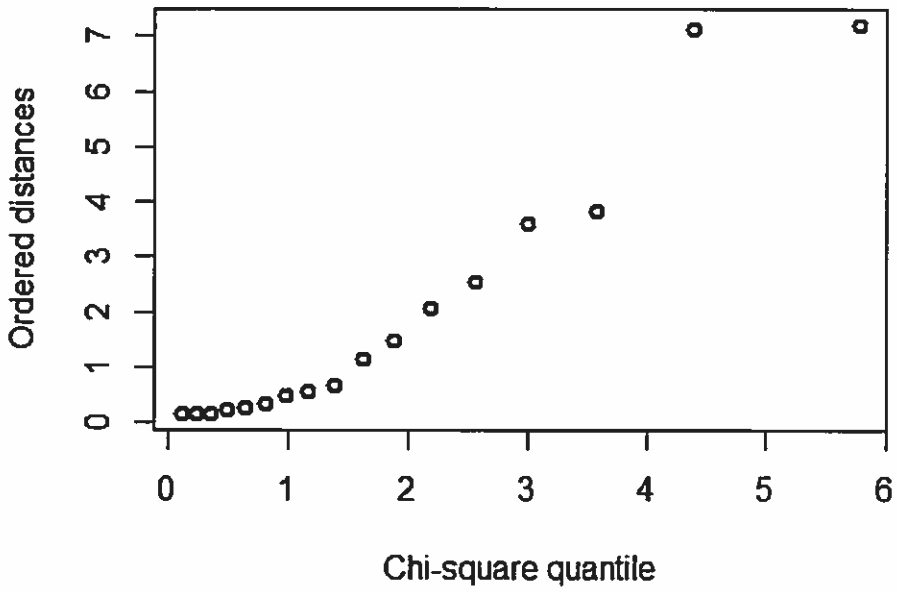
```
library(asbio)
Kullback(Y=plasma[,1:2], X=PR+QRS)
```

```
##
## Kullback test for equal covariance matrices
##     Chi* df P(Chi>Chi*)
## 214  NaN 42         NaN
```

Does box's m test work?

I could not explain why I have the NAN here, I have tried with all cases when X=PR, X=QRS, and X=PR+QRS, they all give the same result, NAN for P-value. So let's we try to check the condition whether or not the residuals follow multivariate normal probability with means equal zero.

```
chisplot <- function(x) {
if (!is.matrix(x)) stop("x is not a matrix")
### determine dimensions
n <- nrow(x)
p <- ncol(x)
xbar <- apply(x, 2, mean)
S <- var(x)
S <- solve(S)
index <- (1:n)/(n+1)
xcent <- t(t(x) - xbar)
di <- apply(xcent, 1, function(x,S) x %*% S %*% x,S)
quant <- qchisq(index,p)
plot(quant, sort(di), ylab = "Ordered distances",
xlab = "Chi-square quantile", lwd=2,pch=1)
}
chisplot(residuals(plasma.manova))
```



So from the QQ plot, we see that the residuals do not followthe multivariate normal probability distribution. So the results we got from Manova above may not be a good approach to work with this data set.

Mat 455, HW5, P1 — Tran Le

#7 Consider the covariance matrix

$$\Sigma = \begin{bmatrix} 0.95 & 0.25 & 0.25 \\ 0.25 & 0.45 & 0.25 \\ 0.25 & 0.25 & 1.35 \end{bmatrix}$$

a) Find a factor model with $m = 1$ factors based on Σ. Comment on the model.

b) Compute the communalities h_i^2, $i = 1, 2, 3$, and interpret these quantities.

c) Calculate $\text{Corr}(X_i, F_j)$ for $i = 1, 2, 3$ and $j = 1$. Which variable would carry the most weight in "naming" or "identifying" the common factor. Explain.

d) What proportion of the total variance is explained by the hidden factor.

---

a) We have the factor model is $X - \mu = LF + \varepsilon$ and the covariance structure is

$$\Sigma = LL' + \Psi = \begin{bmatrix} l_{11} \\ l_{21} \\ l_{31} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \end{bmatrix} + \begin{bmatrix} \psi_1 & 0 & \\ 0 & \psi_2 & \\ & & \psi_3 \end{bmatrix}$$

$$\Rightarrow \quad 0.95 \overset{①}{=} l_{11}^2 + \psi_1 \qquad 0.25 \overset{②}{=} l_{11} l_{21} \qquad 0.25 \overset{④}{=} l_{11} l_{31}$$

$$0.45 \overset{③}{=} l_{21}^2 + \psi_2 \qquad 0.25 \overset{⑤}{=} l_{21} l_{31}$$

$$1.35 \overset{⑥}{=} l_{31}^2 + \psi_3 .$$

- ④ and ⑤ $\Rightarrow l_{11} = l_{21}$

$l_{11} = l_{21}$ and ② $\Rightarrow l_{11} = l_{21} = \sqrt{0.25} = 0.5$

$l_{11} = 0.5$ and (4) $\Rightarrow l_{31} = 0.5$

$l_{11} = 0.5$ and (1) $\Rightarrow \psi_1 = 0.7$

$l_{21} = 0.5$ and (3) $\Rightarrow \psi_2 = 0.2$

$l_{31} = 0.5$ and $\Rightarrow \psi_3 = 1.1$

or

$l_{11} = l_{21} = -0.5$

$l_{31} = -0.5$

$\psi_1 = 0.7$

$\psi_2 = 0.2$

$\psi_3 = 1.1$

Since the variances are the same, we can use both of these models (this agrees with what we have known that the factor models are not unique)

In this case, variances are the same $\Rightarrow$ can use one of the two

$$X = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} F + \mu + \varepsilon .$$

b) Compute the communalities h_i^2, $i=1,2,3$ and interpret these quantities

$h_1^2 = l_{11}^2 = 0.25$
$h_2^2 = l_{21}^2 = 0.25$
$h_3^2 = l_{31}^2 = 0.25$

We have that the communalities h_i represents the proportion of variance of X_i is explained by $m=1$ factor, which means that the higher the communality, the more informative X_i is

since $h_1^2 = h_2^2 = h_3^2 \Rightarrow$ the there variables X_1, X_2, X_3 share the same amount information with the rest of the variables.

c) Cal $cor(X_i, F_j)$ $i=1,2,3$, $j=1$. Which variables would carry the most weight?

we have $Cov(X_i, F_j) = L \Rightarrow$ $cov(X_1, F) = 0.5$ $cor(X_1, F) = \frac{\sigma_{X_1,F}}{\sigma_{X_1}\sigma_F} = \frac{0.5}{\sqrt{0.95}\cdot 1}$

$cov(X_2, F) = 0.5$ $cor(X_2, F) = \frac{\sigma_{X_2,F}}{\sigma_{X_2}\cdot\sigma_F} = \frac{0.5}{\sqrt{0.45}\cdot 1}$

$cov(X_3, F) = 0.5$ $cor(X_3, F) = \frac{\sigma_{X_3,F}}{\sigma_{X_3}\sigma_F} = \frac{0.5}{\sqrt{1.35}\cdot 1}$.

So we have that $cor(X_2, F) > cor(X_1, F) > cor(X_3, F)$

$\Rightarrow$ The variable X_2 carries the most weight since it has the biggest correlation with F comparing with other variable

d) What proportion of the total variance is explained by the hidden factor.

* We first need to find the eigenvalues λ_i, $i=\overline{1,3}$ of Σ.

$$\det(\Sigma - \lambda I) = \det\begin{pmatrix} 0.95-\lambda & 0.25 & 0.25 \\ 0.25 & 0.45-\lambda & 0.25 \\ 0.25 & 0.25 & 1.35-\lambda \end{pmatrix} = -\lambda^3 + 2.75\lambda^2 - 2.13\lambda + 0.4365$$

$\Rightarrow \begin{cases} \lambda_1 = 1.5711 \\ \lambda_2 = 0.85331 \\ \lambda_3 = 0.32559 \end{cases}$ $\Rightarrow$ The total variance explained by the

by 1st factor is $\frac{\lambda_1}{\sum_{i=1}^{3}\sigma_{ii}} = \frac{1.5711}{\underbrace{0.95+0.45+1.35}_{2.75}} = 0.5713$

by 2nd factor is $\frac{0.85331}{2.75} = 0.3103$

by 3rd factor is $\frac{0.32559}{2.75} = 0.1184$

4. Wichern - Johnson _ 9.6

Verify the following matrix identities.

a) $(I + L'\Psi^{-1}L)^{-1}L'\Psi^{-1}L = I - (I + L'\Psi^{-1}L)^{-1}$

b) $(LL' + \Psi)^{-1} = \Psi^{-1} - \Psi^{-1}L(I + L'\Psi^{-1}L)^{-1}L'\Psi^{-1}$

c) $L'(LL' + \Psi)^{-1} = (I + L'\Psi^{-1}L)^{-1}L'\Psi^{-1}$

a) Put $B := L'\Psi^{-1}L$,

Problem a) $\Leftrightarrow$ we need to prove that $(I + B)^{-1}B = I - (I + B)^{-1}$ (*)

We multiply both sides of (*) by $(I+B)$, we have

$(I+B)(I+B)^{-1}B = (I+B)[I - (I+B)^{-1}]$

$\Leftrightarrow IB \qquad B = (I+B)I - (I+B)(I+B)^{-1} = (I+B) - I = B$. (correct equality)

so we have that (*) is correct □a).

b) Prove that $(LL' + \Psi)^{-1} = \Psi^{-1} - \Psi^{-1}L(I + L'\Psi^{-1}L)^{-1}L'\Psi^{-1}$

We multiply both sides of (b) by $(LL'+\Psi)$ we have:

- $(LHS)(LL'+\Psi) = (LL'+\Psi)^{-1}(LL'+\Psi) = I$
- $RHS(LL'+\Psi) = \left(\Psi^{-1} - \Psi^{-1}L(I + L'\Psi^{-1}L)^{-1}L'\Psi^{-1}\right)(LL'+\Psi)$

$= \Psi^{-1}LL' + \Psi^{-1}\Psi - \Psi^{-1}L\underbrace{(I + L'\Psi^{-1}L)^{-1}L'\Psi^{-1}L}_{\text{from (a)} = I - (I + L'\Psi^{-1}L)^{-1}}L' - \Psi^{-1}L(I + L'\Psi^{-1}L)^{-1}\underbrace{L'\Psi^{-1}}_{=I}$

$= \Psi^{-1}LL' + I - \Psi^{-1}L\left[I - (I + L'\Psi^{-1}L)^{-1} + (I + L'\Psi^{-1}L)^{-1}\right]L'$

$= \Psi^{-1}LL' + I - \Psi^{-1}LIL' = I$

So we have $(LHS)(LL'+\Psi) = RHS(LL'+\Psi)$

since $LL'+\Psi$ is invertible $\Rightarrow$ LHS = RHS □b

=> Prove that $L'(LL'+\Psi)^{-1} = (I+L'\Psi^{-1}L)^{-1}L'\Psi^{-1}$

- We first notice that Ψ, LL' and $I+L'\Psi^{-1}L$ are symmetric matrices.
- from b) we have that $(LL'+\Psi)^{-1} = \Psi^{-1} - \Psi^{-1}L(I+L'\Psi^{-1}L)^{-1}L'\Psi^{-1}$

then we have $(LL'+\Psi)^{-1}L = [\Psi^{-1} - \Psi^{-1}L(I+L'\Psi^{-1}L)^{-1}L'\Psi^{-1}]L$

$$= \Psi^{-1}L - \Psi^{-1}L\underbrace{(I+L'\Psi^{-1}L)^{-1}L'\Psi^{-1}L}_{\text{by(a)}\ I-(I+L'\Psi^{-1}L)^{-1}}$$

$$= \Psi^{-1}L(I+L'\Psi^{-1}L)^{-1}$$

take transpose both sides, note that Ψ, $L'L$ and $(I+L'\Psi^{-1}L)$ are symmetric, we have

$$L'((LL'+\Psi)^{-1})' = ((I+L'\Psi^{-1}L)^{-1})'L'(\Psi^{-1})'$$

$$\Rightarrow L'(LL'+\Psi)^{-1} = (I+L'\Psi^{-1}L)^{-1}L'\Psi^{-1} \quad \square c$$

Mat 755 Final exam

P1 → Classification rules are often evaluated in terms of the expected cost of misclassification $ECM = c(2|1)\,P(2|1)\,p_1 + c(1|2)\,P(1|2)\,p_2$

a) Show that the region R_1 and R_2 that minimize the ECM are

$$R_1: \frac{f_1(x)}{f_2(x)} \geq \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$$

$$R_2: \frac{f_1(x)}{f_2(x)} < \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$$

---

* We have $P(2|1) = P(R_2|\pi_1)$ = probability of misclassifying an object into π_2 while it belongs to π_1

$$= \int_{R_2} f_1(x)\,dx$$

Similarly $P(1|2) = \int_{R_1} f_2(x)\,dx$

*So we have

$$ECM = c(2|1)\,P(2|1)\,p_1 + c(1|2)\,P(1|2)\,p_2$$

$$= c(2|1)\int_{R_2} f_1(x)\,dx\; p_1 + c(1|2)\int_{R_1} f_2(x)\,dx\; p_2$$

- We note that since $\left.\begin{array}{l}\Omega = R_1 + R_2 \\ R_1 \cap R_2 = \phi\end{array}\right\} \Rightarrow 1 = \int_{R_1} f_1(x) + \int_{R_2} f_1(x)\,dx$

$$\Rightarrow \int_{R_2} f_1(x)\,dx = 1 - \int_{R_1} f_1(x)\,dx$$

- So then

$$ECM = c(2|1)\left[1 - \int_{R_1} f_1(x)\,dx\right] p_1 + c(1|2)\left[\int_{R_1} f_2(x)\,dx\right] p_2$$

$$= \int_{R_1} \underbrace{\left[c(1|2)\,f_2(x)\,p_2 - c(2|1)\,f_1(x)\,p_1\right]}_{(*)}\,dx + \underbrace{c(2|1)\,p_1}_{\text{constant}}$$

⇒ ECM is minimized when R_1 is chosen to be the region where $(*) \leq 0$

→ ECM is minized when R_1 is chosen to be the region where :

$$c(1|2)\, p_2\, f_2(x) \leq c(2|1)\, p_1\, f_1(x)$$

$$\Rightarrow R_1 : \frac{f_1}{f_2} \geq \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$$

And so since $\Omega = R_1 \cup R_2$, $R_1 \cap R_2 = \phi$

$$R_2 : \frac{f_1(x)}{f_2(x)} < \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$$

---

b) A researcher has enough data to estimate the density function $\begin{cases} f_1(x) \\ f_2(x) \end{cases}$ associated with population π_1 and π_2, respectively.

Let $c(2|1) = 50$, $c(1|2) = 100$

It is known that about 30% of all possible item belong to π_2.

Give the minimum ECM rule for assigning a new item to one of the two populatio

* $c(2|1) = 50 \quad c(1|2) = 100$

$p_1 = 0.7 \quad p_2 = 0.3$

Then $\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right) = \left(\frac{100}{50}\right)\left(\frac{0.3}{0.7}\right) = 2 * \frac{3}{7} = \frac{6}{7}$.

* So for each observation x, we need to compare $\frac{f_1(x)}{f_2(x)}$ and $\frac{6}{7}$.

assign the observation to group 1 (π_1), R1, when $\frac{f_1(x)}{f_2(x)} \geq \frac{6}{7}$

assign the observation to group 2 (π_2), R2, when $\frac{f_1(x)}{f_2(x)} < \frac{6}{7}$.

---

c) Measurements recorded on a new item yield the density values $f_1(x) = 0.3$ and $f_2(x) = 0.5$.

Given the preceding information, assign this item to π_1 or π_2 :

$$\frac{f_1(x)}{f_2(x)} = \frac{0.3}{0.5} = \frac{3}{5} < \frac{6}{7} = \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right).$$

so we assign the new item to π_2, group 2.

Problem 2:

Let $X = [x_1, x_2, \dots, x_n]$ be centered data

$d_{ij}^2 = (x_i - x_j)^T (x_i - x_j)$

$B = X^T X$

a) Show that B has entries $b_{ij} = -\frac{1}{2}\left[d_{ij}^2 - \frac{2}{n}\sum_{i=1}^{n} d_{ij}^2 + \frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2\right]$

- Note that $d_{ij}^2 = x_i^T x_i + x_j^T x_j - 2 x_i^T x_j \Rightarrow x_i^T x_j = -\frac{1}{2}\left(d_{ij}^2 - x_i^T x_i - x_j^T x_j\right)$ (1)

We also have

- $\frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 = \frac{1}{n}\sum_{j=1}^{n} x_i^T x_i + \frac{1}{n}\sum_{j=1}^{n} x_j^T x_j - \underbrace{\frac{2}{n}\sum_{j=1}^{n} x_i^T x_j}_{= -\frac{2}{n} x^T \sum_{j=1}^{n} x_j = 0}$ (2)

$= x_i^T x_i + \frac{1}{n}\sum_{j=1}^{n} x_j^T x_j$

- $\frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^T x_i + \underbrace{\frac{1}{n}\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} x_j^T x_j}_{= \frac{1}{n}\frac{1}{n} n \sum_{j=1}^{n} x_j^T x_j} = \frac{2}{n}\sum_{i=1}^{n} x_i^T x_i$ (3)

* So now consider $B = X^T X$, then we have B will have entries:

$b_{ij} = x_i^T x_j \overset{\text{by (1)}}{=\!=} -\frac{1}{2}\left(d_{ij}^2 - \underbrace{x_i^T x_i}_{} - x_j^T x_j\right)$

$\overset{\text{by (2)}}{=} -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 + \frac{1}{n}\sum_{j=1}^{n} x_i^T x_j - \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 + \frac{1}{n}\sum_{j=1}^{n} x_j^T x_j\right)$

$= -\frac{1}{2}\left(d_{ij}^2 - \frac{2}{n}\sum_{i=1}^{n} d_{ij}^2 + \underbrace{\frac{2}{n}\sum_{i=1}^{n} x_i^T x_i}_{\overset{\text{by (3)}}{=\!=} \frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2}\right)$

$= -\frac{1}{2}\left(d_{ij}^2 - \frac{2}{n}\sum_{i=1}^{n} d_{ij}^2 + \frac{1}{n^2}\sum_{i,j=1}^{n} d_{ij}^2\right)$ □a).

b) If $B = V \Lambda V^T$ is the spectral decomposition of B.
Show that $X = \Lambda^{1/2} V^T$

* Note that since B is symmetric and since $B = X^T X \Rightarrow$ positive definite.

$\Rightarrow B = V \Lambda V^T$ is well defined, where $\Lambda = \text{diag}(\lambda_i) = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$

and $B = V \Lambda V^T$ ($\lambda_i, i = \overline{1,n}$ are eigenvalues of B)

$= V \Lambda^{1/2} \Lambda^{1/2} V^T$ where $\Lambda^{1/2} = \text{diag}(\sqrt{\lambda_i})$

$= V (\Lambda^{1/2})^T \Lambda^{1/2} V^T$ (since $\Lambda^{1/2}$ is a diagonal matrix)

$B = (\Lambda^{1/2} V^T)^T (\Lambda^{1/2} V^T)$

Since $B = X^T X$ $\Rightarrow X = \Lambda^{1/2} V^T$ □ b