Multivariate Data Analaysis using R

Setia Pramana, PhD

February 17, 2015

Chapter 1

Introduction to R

1.1 Matrix Operation in R

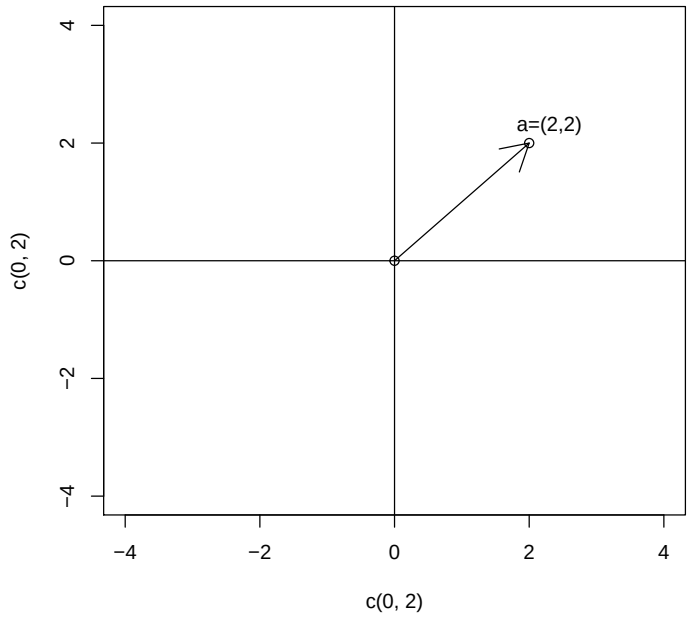
Row and column vectors are drawn the same way.

The vector in R printed row format" but can really be regarded as a column vector, cfr the convention above.

```
> a <- c(2,2)
> a
```

```
[1] 2 2
```

```
> #draw the vector
>
> plot(c(0,2),c(0,2), xlim=c(-4,4), ylim=c(-4,4))
> arrows(0,0,2,2)
> abline(h=0,v=0)
> text(2.3,2.3,"a=(2,2)")
>
>
>
```




```
> a <- c(1:3)
> a

[1] 1 2 3

> # transpose
>
> t(a)

     [,1] [,2] [,3]
[1,]    1    2    3

> # Multiplication by a number
>
> 8*a

[1]  8 16 24

> #Vector Addition
>
>  a <- c(1,3,2)
>  b <- c(6,8,1)
>  a+b
```

```
[1]  7 11  3

> ## length of a Vector (Norm)
>
> sqrt(sum(a*a))

[1] 3.741657

> x <- c(-1,5,2,-2)
> y <- c(4,-3,0,1)
> lx <- sqrt(sum(x^2))
> ly <- sqrt(sum(y^2))
> cosxy <- 1/lx * 1/ly * sum (x*y)
>   acos (cosxy)

[1] 2.355063

>
>
>

> A <- matrix(c(1,3,2,4,8,7),ncol=3)
> A

     [,1] [,2] [,3]
[1,]    1    2    8
[2,]    3    4    7

> 8*A

     [,1] [,2] [,3]
[1,]    8   16   64
[2,]   24   32   56

> t(A)

     [,1] [,2]
[1,]    1    3
[2,]    2    4
[3,]    8    7

> ## Addition of matrices
>
> B <- matrix(c(5,8,3,4,2,7),ncol=3,byrow=T)
> A+B

     [,1] [,2] [,3]
[1,]    6   10   11
[2,]    7    6   14
```

```
> ## Multiplication ##
>
>  A%*%a

     [,1]
[1,]   23
[2,]   29

> # Different with
>  A*a

     [,1] [,2] [,3]
[1,]    1    4   24
[2,]    9    4   14

> A <- matrix(c(1,3,2,2,8,9),ncol=2)
> B <- matrix(c(5,8,4,2), ncol=2)
> A

     [,1] [,2]
[1,]    1    2
[2,]    3    8
[3,]    2    9

> B

     [,1] [,2]
[1,]    5    4
[2,]    8    2

> A%*%B

     [,1] [,2]
[1,]   21    8
[2,]   79   28
[3,]   82   26

> # determinant ##
>
> #det(A)
> # Error in determinant.matrix(x, logarithm = TRUE, ...) :
> #  'x' must be a square matrix
>
> det(B)

[1] -22

> D <- matrix(c(5,8,4,2,5,6,7,8,9), ncol=3)
> det(D)
```

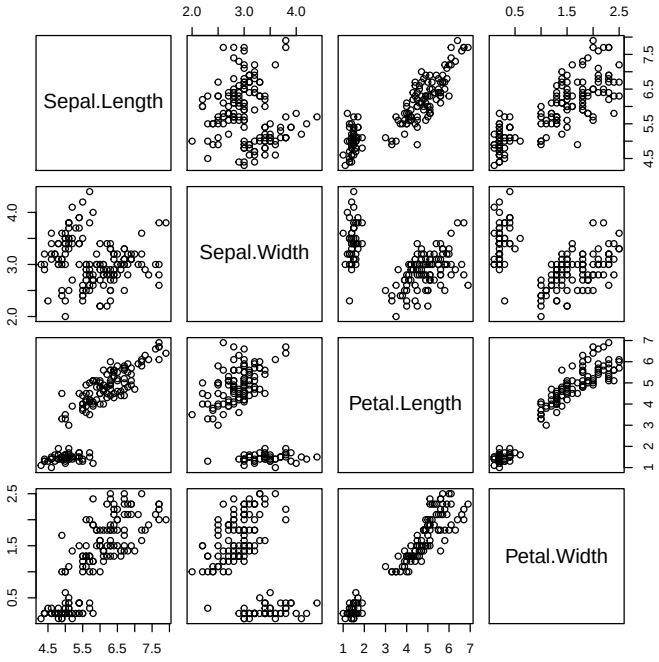
```
[1] 101

> ## DIagonal Matrix ##
> C <- diag(c(1,2,3,5))
> C

     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    2    0    0
[3,]    0    0    3    0
[4,]    0    0    0    5

> det(C)

[1] 30

> diag(1,3)

     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1

> diag(A)

[1] 1 8

> ## Inverse Matrix ###
>
> A <- matrix(c(1,3,2,4),ncol=2,byrow=T)
> A

     [,1] [,2]
[1,]    1    3
[2,]    2    4

> #invers#
> B <- solve(A)
> B

     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5

> A%*%B

     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

```
> A <- matrix(c(1,3,2,6),ncol=2)
> A

     [,1] [,2]
[1,]    1    2
[2,]    3    6

> #invers#
> #solve(A)
> #Error in solve.default(A) :
> #  Lapack routine dgesv: system is exactly singular: U[2,2] = 0
>
>
>
>
> ## Solving systems of linear equations ##
> A <- matrix(c(1,2,3,4),ncol=2)
> A

     [,1] [,2]
[1,]    1    3
[2,]    2    4

>  b <- c(7,10)
>  x <- solve(A)%*%b
>  x

     [,1]
[1,]    1
[2,]    2

>
>
>

> A <- cbind(x1=c(42,52,48,58),x2=c(4,5,4,3))
> meanA <- colMeans(A)
> meanA

x1 x2
50  4

> var(A[,1])

[1] 45.33333

> var(A[,2])

[1] 0.6666667
```

```
> var(A)

          x1         x2
x1 45.33333 -2.0000000
x2 -2.00000  0.6666667

> cor(A)

           x1         x2
x1  1.0000000 -0.3638034
x2 -0.3638034  1.0000000

> t(A)-meanA

   [,1] [,2] [,3] [,4]
x1   -8    2   -2    8
x2    0    1    0   -1

> a <- (t(A)-meanA)
> s1 <- sum(a[1,]^2)/4
> s1

[1] 34

> s2 <- sum(a[2,]^2)/4
> s2

[1] 0.5

> b <- t(a)
> b[,1]*b[,2]

[1]  0  2  0 -8

> s12 <- sum( b[,1]*b[,2] )/4
> s12

[1] -1.5

> r12 <- s12/(sqrt(s1)*sqrt(s2))
> r12

[1] -0.3638034

> cov(A)

          x1         x2
x1 45.33333 -2.0000000
x2 -2.00000  0.6666667
```

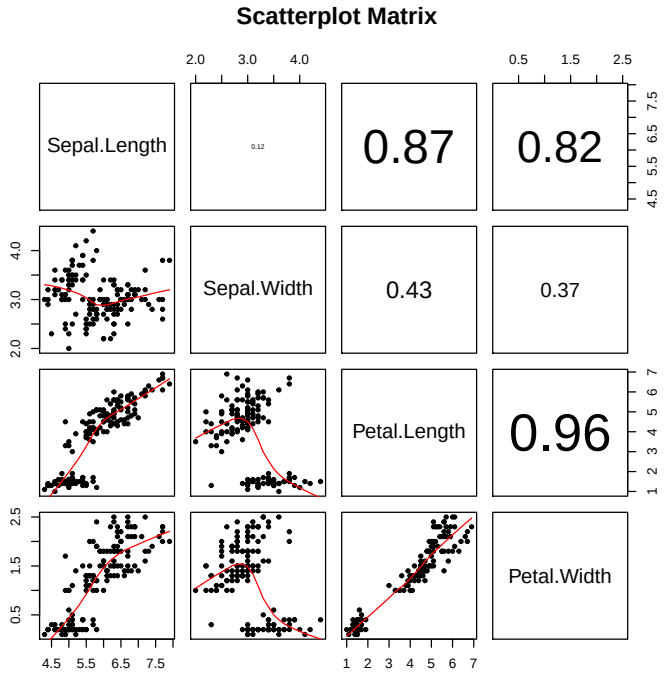
1.2 Visualisation

```
> ## Visualization
>
> data(iris)
> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> # Plot #1: Basic scatterplot matrix of the four measurements
> pairs(~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width, data=iris)
>
>
```

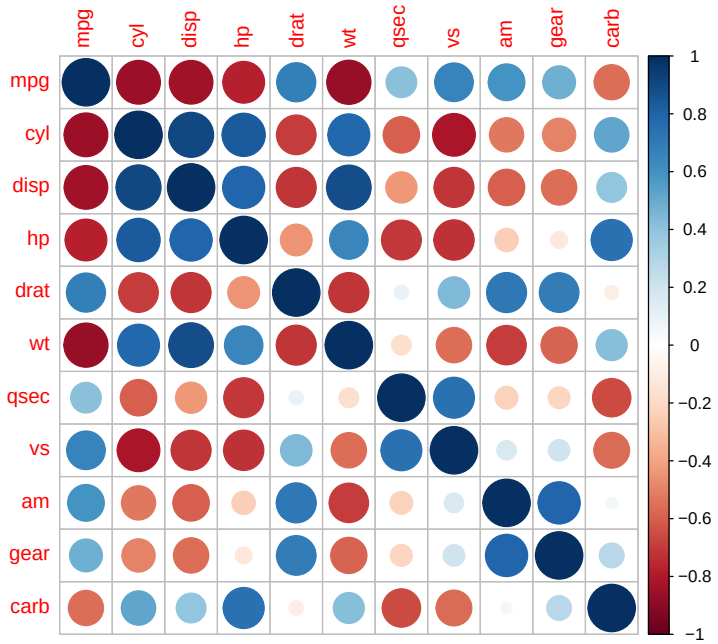



```
> panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)
+ {
+     usr <- par("usr"); on.exit(par(usr))
```

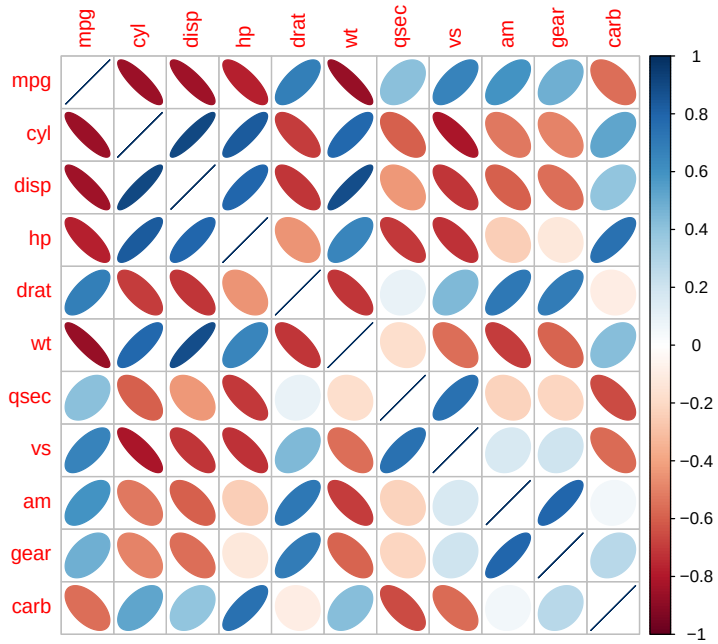
```
+     par(usr = c(0, 1, 0, 1))
+     r <- abs(cor(x, y))
+     txt <- format(c(r, 0.123456789), digits=digits)[1]
+     txt <- paste(prefix, txt, sep="")
+     if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
+     text(0.5, 0.5, txt, cex = cex.cor * r)
+ }
> pairs(~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width, data=iris,
+       lower.panel=panel.smooth, upper.panel=panel.cor,
+       pch=20, main=" Scatterplot Matrix")
>
>
```



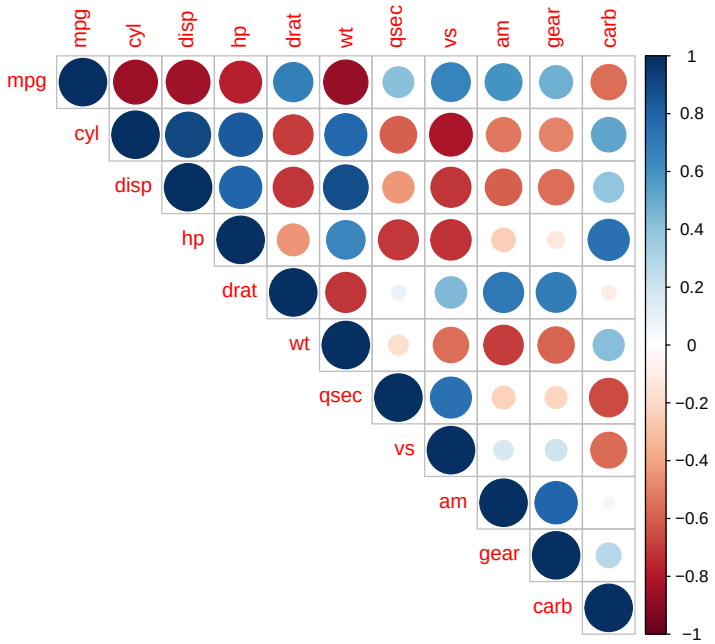

```
> #install.packages("corrplot")
> library(corrplot)
> M <- cor(mtcars)
> corrplot(M, method = "circle")
```



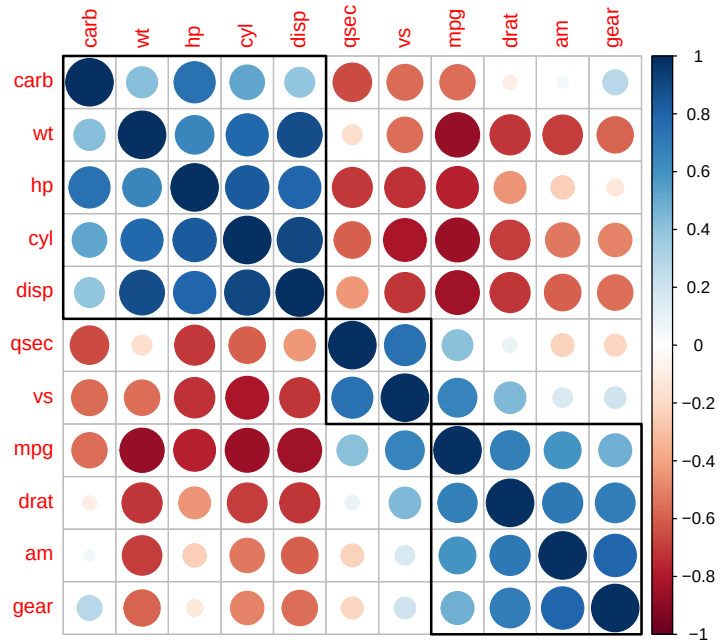

```
> corrplot(M, method = "ellipse")
```




```
> corrplot(M, type = "upper")
```

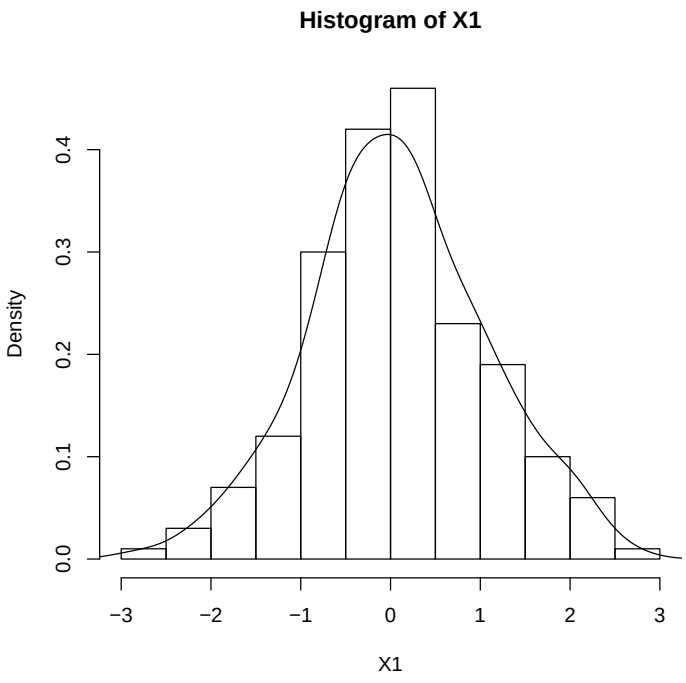



```
> corrplot(M, order = "hclust", addrect = 3)
```




```
> ### Larger Data
>
> X1 <- rnorm(n=200,mean=0,sd=1)
> X2 <- rnorm(n=200,mean=0,sd=1)
> X3 <- rnorm(n=200,mean=0,sd=1)
> hist(X1, prob=TRUE)
> lines(density(X1))
> dt <- cbind(X1,X2,X3)
> colMeans(dt)

         X1          X2          X3
 0.08229199 -0.02556872  0.01206121

> #rowMeans(dt)
> cov(dt)

            X1          X2         X3
X1  0.96989673 -0.03731477 0.08036119
X2 -0.03731477  1.10626715 0.07787986
X3  0.08036119  0.07787986 0.97415020

> cor(dt)
```

```
            X1          X2         X3
X1  1.00000000 -0.03602368 0.08267432
X2 -0.03602368  1.00000000 0.07502089
X3  0.08267432  0.07502089 1.00000000

> dt2 <- cbind(X1,X2=-X1,X3=2*X1)
> cov(dt2)

           X1         X2        X3
X1  0.9698967 -0.9698967  1.939793
X2 -0.9698967  0.9698967 -1.939793
X3  1.9397935 -1.9397935  3.879587

> cor(dt2)

   X1 X2 X3
X1  1 -1  1
X2 -1  1 -1
X3  1 -1  1

>
>
>
```



1.3 Regresion Example

```
> #setwd("~/MUltivariate_Data_Analysis/MVA using R/")
>
> setwd("C:/Users/Administrator/Documents/MUltivariate_Data_Analysis")
> hsb2 <- read.table("hsb.txt")
> y <- matrix(hsb2$write, ncol = 1)
> x <- as.matrix(cbind(1, hsb2$math, hsb2$science, hsb2$socst, hsb2$female))
> n <- nrow(x)
> p <- ncol(x)
> #parameter estimates
> beta.hat <- solve(t(x) %*% x) %*% t(x) %*% y
> beta.hat

          [,1]
[1,] 6.5689235
[2,] 0.2801611
[3,] 0.2786543
[4,] 0.2681117
[5,] 5.4282152

> y.hat <- x %*% beta.hat
> y.hat[1:5, 1]

[1] 46.43465 60.75571 46.17103 49.51943 53.66160

> #the variance, residual standard error and df's
> sigma2 <- sum((y - y.hat)^2)/(n - p)
> #residual standard error
> sqrt(sigma2)

[1] 6.101191

> #degrees of freedom
> n - p

[1] 195

> #the standard errors, t-values and p-values for estimates
> #variance/covariance matrix
> v <- solve(t(x) %*% x) * sigma2
> #standard errors of the parameter estimates
> sqrt(diag(v))

[1] 2.81907949 0.06393076 0.05804522 0.04919499 0.88088532

> #t-values for the t-tests of the parameter estimates
> t.values <- beta.hat/sqrt(diag(v))
> t.values
```

```
         [,1]
[1,] 2.330166
[2,] 4.382257
[3,] 4.800642
[4,] 5.449980
[5,] 6.162227

> #p-values for the t-tests of the parameter estimates
> 2 * (1 - pt(abs(t.values), n - p))

             [,1]
[1,] 2.082029e-02
[2,] 1.917191e-05
[3,] 3.142297e-06
[4,] 1.510015e-07
[5,] 4.033511e-09

> #checking that we got the correct results
> ex1 <- lm(write ~ math + science + socst + female, hsb2)
> summary(ex1)

Call:
lm(formula = write ~ math + science + socst + female, data = hsb2)

Residuals:
     Min       1Q   Median       3Q      Max
-18.3086  -3.8149   0.1035   3.8394  15.5882

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.56892    2.81908   2.330   0.0208 *
math         0.28016    0.06393   4.382 1.92e-05 ***
science      0.27865    0.05805   4.801 3.14e-06 ***
socst        0.26811    0.04919   5.450 1.51e-07 ***
female       5.42822    0.88089   6.162 4.03e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.101 on 195 degrees of freedom
Multiple R-squared:  0.594,        Adjusted R-squared:  0.5857
F-statistic: 71.32 on 4 and 195 DF,  p-value: < 2.2e-16

> ## Multicol situation
>
>
> x <- as.matrix(cbind(1, hsb2$math, hsb2$math, hsb2$socst, 2*hsb2$socst))
> n <- nrow(x)
```

```
> p <- ncol(x)
> # beta.hat <- solve(t(x) %*% x) %*% t(x) %*% y
> # Error in solve.default(t(x) %*% x) :
> #  Lapack routine dgesv: system is exactly singular: U[3,3] = 0
>
> hsb3 <- data.frame(x1=hsb2$math, x2=hsb2$math+0.4, x3= hsb2$socst+.5, x4=2*hsb2$socst,y=hsb2$wr
> ex1 <- lm(y ~ . , hsb3)
> summary(ex1)

Call:
lm(formula = y ~ ., data = hsb3)

Residuals:
     Min       1Q   Median       3Q      Max
-23.1061  -4.0256   0.2952   4.0969  21.2657

Coefficients: (2 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.12635    2.97258   4.416 1.66e-05 ***
x1           0.41439    0.06175   6.711 2.01e-10 ***
x2                NA         NA      NA       NA
x3           0.33708    0.05388   6.256 2.41e-09 ***
x4                NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.845 on 197 degrees of freedom
Multiple R-squared:  0.4838,        Adjusted R-squared:  0.4786
F-statistic: 92.32 on 2 and 197 DF,  p-value: < 2.2e-16

> ex1 <- lm(y ~ x1+x3 , hsb3)
> summary(ex1)

Call:
lm(formula = y ~ x1 + x3, data = hsb3)

Residuals:
     Min       1Q   Median       3Q      Max
-23.1061  -4.0256   0.2952   4.0969  21.2657

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.12635    2.97258   4.416 1.66e-05 ***
x1           0.41439    0.06175   6.711 2.01e-10 ***
x3           0.33708    0.05388   6.256 2.41e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.845 on 197 degrees of freedom
Multiple R-squared:  0.4838,        Adjusted R-squared:  0.4786
F-statistic: 92.32 on 2 and 197 DF,  p-value: < 2.2e-16

>
>
>

> ## Eigen Values and Rank
> A <- cbind(c(3,1),c(1,3))
> A

     [,1] [,2]
[1,]    3    1
[2,]    1    3

> solve(A)

       [,1]   [,2]
[1,]  0.375 -0.125
[2,] -0.125  0.375

> det(A)

[1] 8

> eigen(A)

$values
[1] 4 2

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068

> B <- matrix(c(4,8,8,4),ncol=2)
> B

     [,1] [,2]
[1,]    4    8
[2,]    8    4

> qr(B)

$qr
           [,1]      [,2]
[1,] -8.9442719 -7.155418
```

```
[2,]  0.8944272 -5.366563

$rank
[1] 2

$qraux
[1] 1.447214 5.366563

$pivot
[1] 1 2

attr(,"class")
[1] "qr"
```

```
> eigen (B)
```

```
$values
[1] 12 -4

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

```
> B <- matrix(c(4,8,2,4),ncol=2)
> B
```

```
     [,1] [,2]
[1,]    4    2
[2,]    8    4
```

```
> qr(B)
```

```
$qr
           [,1]      [,2]
[1,] -8.9442719 -4.472136
[2,]  0.8944272  0.000000

$rank
[1] 1

$qraux
[1] 1.447214 0.000000

$pivot
[1] 1 2

attr(,"class")
[1] "qr"
```

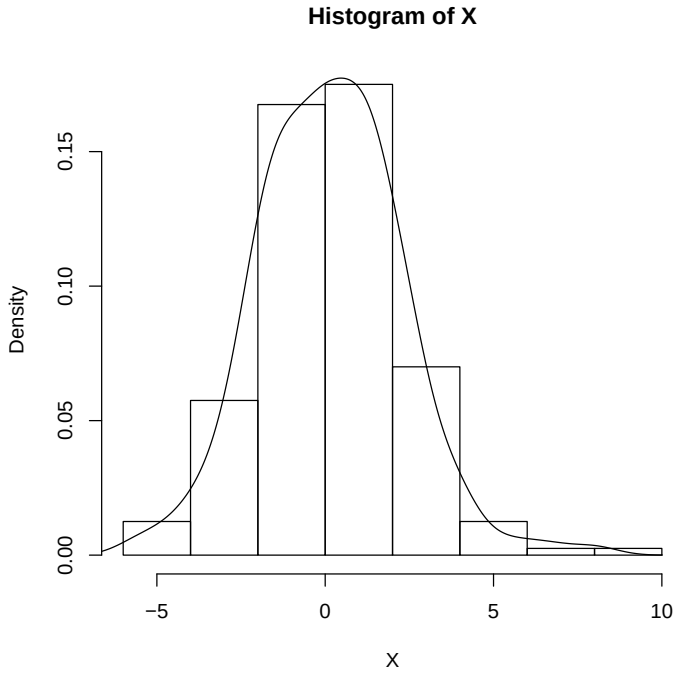
```
> eigen(B)

$values
[1] 8.000000e+00 8.881784e-16

$vectors
          [,1]       [,2]
[1,] 0.4472136 -0.4472136
[2,] 0.8944272  0.8944272

> #solve(B)
> #Error in solve.default(B) :
> #  Lapack routine dgesv: system is exactly singular: U[2,2] = 0
>

> ## Orthogonal MAtrix  #
> k <- 5
> set.seed(25)
> tstMat <- array(runif(k), dim=c(k,k))
> tstOrth <- qr.Q(qr(tstMat))
> t(tstOrth)%*%tstOrth

              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.000000e+00 2.515349e-17 -2.168404e-18 6.938894e-18 -1.387779e-17
[2,]  2.515349e-17 1.000000e+00  3.252607e-18 7.806256e-18  0.000000e+00
[3,] -2.168404e-18 3.252607e-18  1.000000e+00 1.344411e-17  1.734723e-18
[4,]  6.938894e-18 7.806256e-18  1.344411e-17 1.000000e+00  1.387779e-17
[5,] -1.387779e-17 0.000000e+00  1.734723e-18 1.387779e-17  1.000000e+00

> ## Check ortogonal
>
> t(tstOrth[,2]) %*% tstOrth[,3]

             [,1]
[1,] 3.252607e-18

> t(tstOrth[1,]) %*% tstOrth[2,]

             [,1]
[1,] 8.066464e-17

> ## Spectral Decomposition ##
>
> A <- matrix(c(2.2, 0.4, .4, 2.8),2)
> eigen(A)

$values
[1] 3 2
```

```
$vectors
           [,1]       [,2]
[1,] 0.4472136 -0.8944272
[2,] 0.8944272  0.4472136

>
>
>
```

Chapter 2

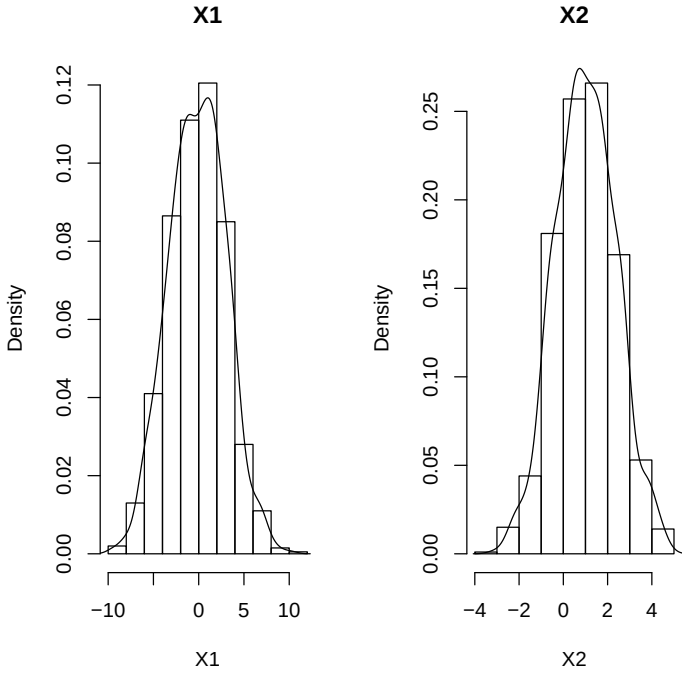
Multivariate Normal Distribution

```
> ## Generate univariate normal distribution ##
>
> X <- rnorm(n=200,mean=0,sd=2)
> hist(X, prob=TRUE)
> lines(density(X))
> X2 <- rnorm(n=200,mean=10,sd=2)
> hist(c(X,X2), prob=TRUE)
> lines(density(c(X,X2)))
>
>
```

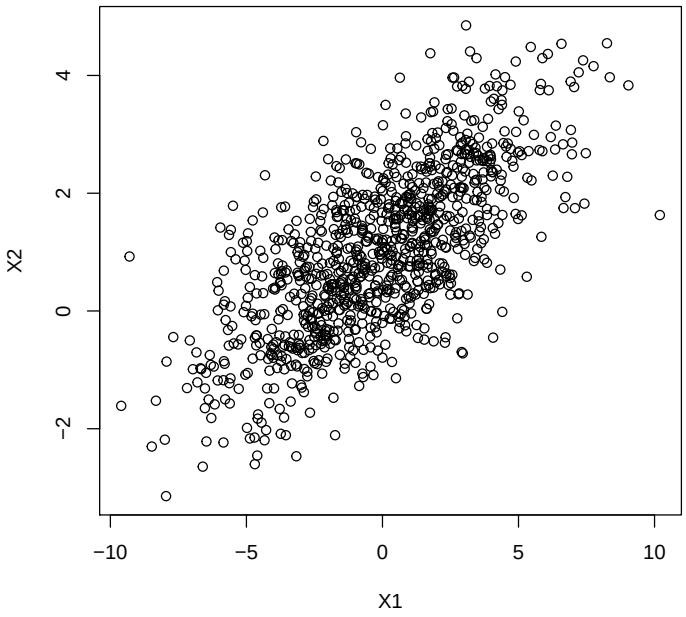



```
> ## Generate Bivariate normal dist #
> library(MASS)
> ## Var COv Matrix
> Sigma <- matrix(c(10,3,3,2),2,2)
> Sigma

     [,1] [,2]
[1,]   10    3
[2,]    3    2

> ## what is the  Correlation ?
>
>
> dt <- mvrnorm(n=1000, c(0,1), Sigma)
> head(dt)

           [,1]       [,2]
[1,] -1.1964133 -0.2059783
[2,] -0.4390306 -0.6178706
[3,] -6.6868434 -0.9730950
[4,] -1.4305064  0.7393564
[5,] -2.5192069  1.8593600
[6,]  1.7857931  2.8835975
```

```
> colMeans(dt)

[1] -0.1421655  1.0114161

> var(dt)

          [,1]     [,2]
[1,] 9.802383 2.996384
[2,] 2.996384 1.866566

> par(mfrow=c(1,2))
> for (i in  1:2) {
+   hist(dt[,i], prob=TRUE, main=paste("X",i, sep=""), xlab=paste("X",i, sep=""))
+   lines(density(dt[,i]))
+   }
```



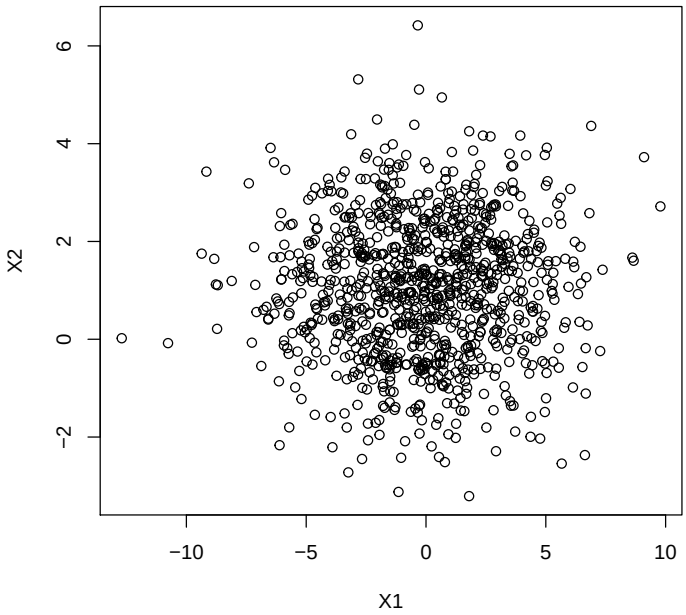

```
> plot(dt, xlab="X1", ylab="X2")
>
```




```
> ## no Correlation
> Sigma <- matrix(c(10,0,0,2),2,2)
> dt2 <- mvrnorm(n=1000, c(0,1), Sigma)
> head(dt2)
```

```
           [,1]     [,2]
[1,]  1.6926359 1.918075
[2,] -3.3944088 1.771776
[3,] -3.2946322 2.188686
[4,]  0.6640126 2.233692
[5,] -3.9805889 1.211079
[6,]  0.8270292 3.270756
```

```
> plot(dt2, xlab="X1", ylab="X2")
>
```

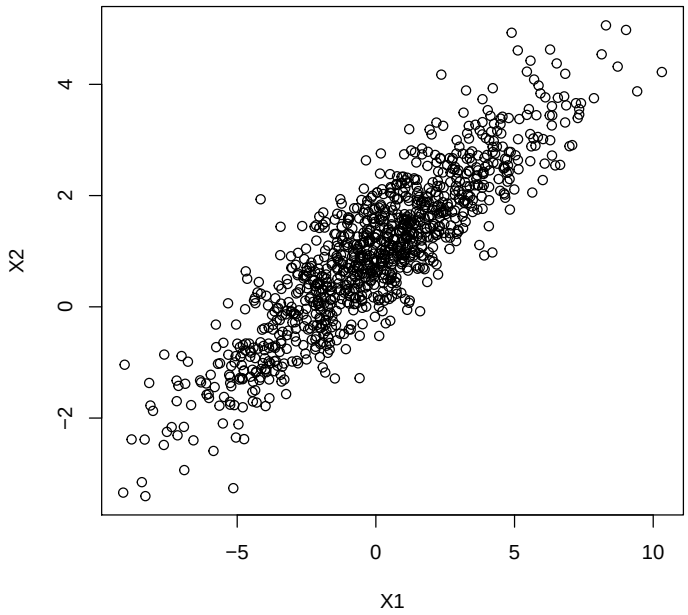



```
> ## high  Correlation
> Sigma <- matrix(c(10,4,4,2),2,2)
> dt3 <- mvrnorm(n=1000, c(0,1), Sigma)
> head(dt3)
```

```
          [,1]       [,2]
[1,]  2.303957  1.6091003
[2,] -6.029965 -1.5665926
[3,] -3.503138  0.3817391
[4,] -1.391998  0.3297537
[5,] -1.947719 -0.3138427
[6,] -2.882276 -0.3959555
```

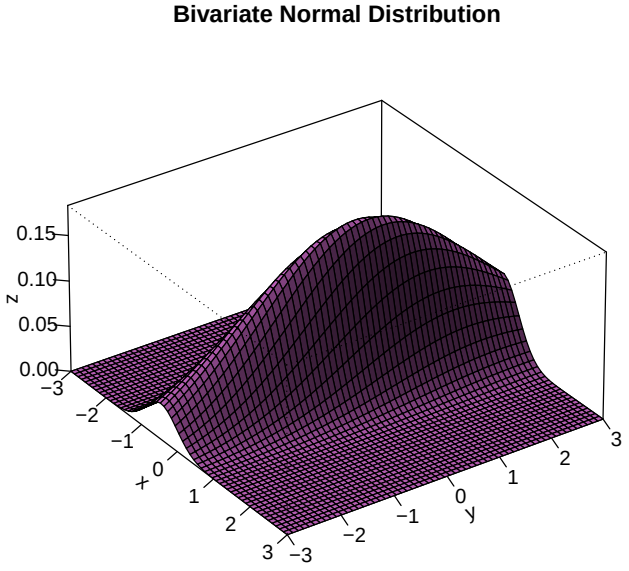
```
> plot(dt3, xlab="X1", ylab="X2")
> cor(dt3)
```

```
          [,1]      [,2]
[1,] 1.0000000 0.8836261
[2,] 0.8836261 1.0000000
```




```
> # Édouard Tallent @ TaGoMa.Tech
> # September 2012
> # This code plots simulated bivariate normal distributions
>
> # Some variable definitions
> mu1 <- 0  # expected value of x
> mu2 <- 0.5        # expected value of y
> sig1 <- 0.5        # variance of x
> sig2 <- 2        # variance of y
> rho <- 0.5        # corr(x, y)
> # Some additional variables for x-axis and y-axis
> xm <- -3
> xp <- 3
> ym <- -3
> yp <- 3
> x <- seq(xm, xp, length= as.integer((xp + abs(xm)) * 10))  # vector series x
> y <- seq(ym, yp, length= as.integer((yp + abs(ym)) * 10))  # vector series y
> # Core function
> bivariate <- function(x,y){
+         term1 <- 1 / (2 * pi * sig1 * sig2 * sqrt(1 - rho^2))
+         term2 <- (x - mu1)^2 / sig1^2
+         term3 <- -(2 * rho * (x - mu1)*(y - mu2))/(sig1 * sig2)
```

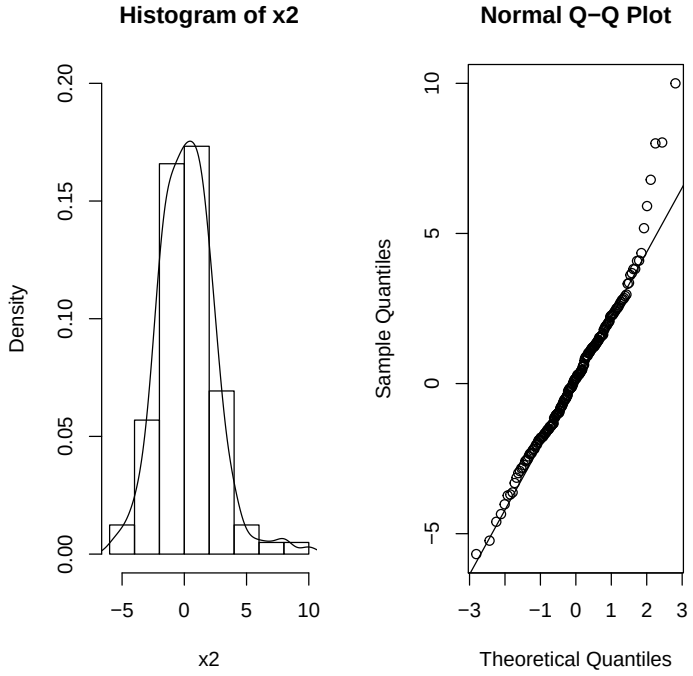
```
+         term4 <- (y - mu2)^2 / sig2^2
+         z <- term2 + term3 + term4
+         term5 <- term1 * exp((-z / (2 *(1 - rho^2))))
+         return (term5)
+ }
> # Computes the density values
> z <- outer(x,y,bivariate)
> # Plot
>
> persp(x, y, z, main = "Bivariate Normal Distribution",
+                 sub = bquote(bold(mu[1])==.(mu1)~", "~sigma[1]==.(sig1)~", "
+     ~mu[2]==.(mu2)~", "~sigma[2]==.(sig2)~", "~rho==.(rho)),
+                 col="orchid2", theta = 55, phi = 30, r = 40, d = 0.1,
+       expand = 0.5,ltheta = 90, lphi = 180, shade = 0.4,
+       ticktype = "detailed", nticks=5)
>
```



$\mu_1 = 0$, $\sigma_1 = 0.5$, $\mu_2 = 0.5$, $\sigma_2 = 2$, $\rho = 0.5$

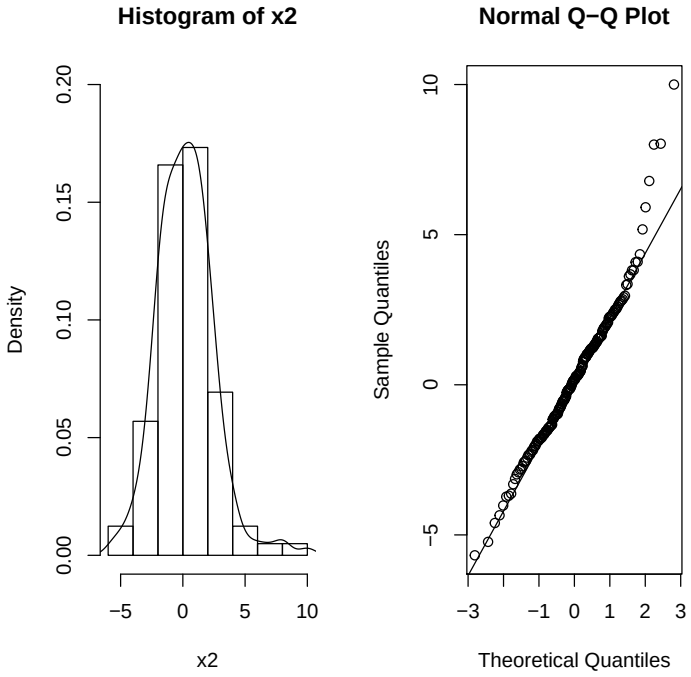
```
> qqnorm(X);qqline(X)
> ## Any Normal Dist
> shapiro.test(X)

        Shapiro-Wilk normality test
```

```
data:  X
W = 0.9898, p-value = 0.1655

> # Specific Normal Dist
> ks.test(X, "pnorm", mean=0, sd=2, alternative="two.sided")

        One-sample Kolmogorov-Smirnov test

data:  X
D = 0.0566, p-value = 0.5438
alternative hypothesis: two-sided
```




```
> x2 <- c(10,8,X)
> par(mfrow=c(1,2))
> hist(x2, prob=TRUE, ylim=c(0,0.2),);lines(density(x2))
> qqnorm(x2);qqline(x2)
> shapiro.test(x2)

        Shapiro-Wilk normality test

data:  x2
W = 0.9663, p-value = 9.26e-05
```

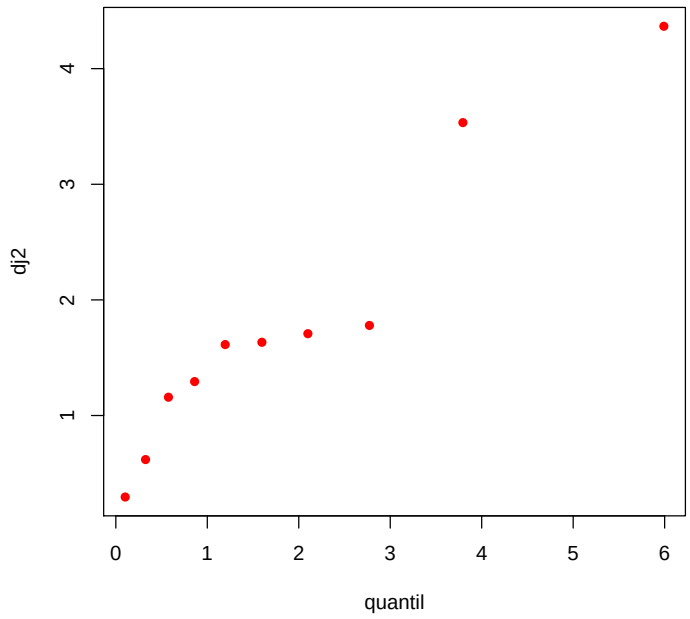
```
> ks.test(x2, "pnorm", mean=0, sd=1, alternative="two.sided")

        One-sample Kolmogorov-Smirnov test

data:  x2
D = 0.2113, p-value = 2.925e-08
alternative hypothesis: two-sided

>
>
```



2.1 Checking Outlier

```
> # Code by a KS student #
> ## Input Data ##
> v1<-matrix(c(108.28,152.36,95.04,65.45,62.97,263.99,265.19,285.06,92.01,165.68), ncol=1)
> v2<-matrix(c(17.05,16.59,10.91,14.14,9.52,25.33,18.54,15.73,8.10,11.13),ncol=1)
> Y<-cbind(v1,v2)
> ## Chi sq Plot ##
>
> ChiSqPlot <- function(X){
+
```

```
+   xbar<-matrix(c(colMeans(X)),ncol=1)
+   sinvers<-solve(cov(X))
+
+   dj2<-c()
+   quantil<-c()
+   for (i in 1:nrow(X) ) {
+     dj2[i]<-t(X[i,]-xbar)%*%sinvers%*%(X[i,]-xbar)
+     quantil[i]<-qchisq((nrow(X)-i+0.5)/nrow(X), df=2)
+   }
+   dj2<-sort(dj2,decreasing=FALSE)
+   quantil<-sort(quantil,decreasing=FALSE)
+   plot(quantil,dj2, pch=16, col=2)
+ }
> ChiSqPlot(Y)
>
>
>
```

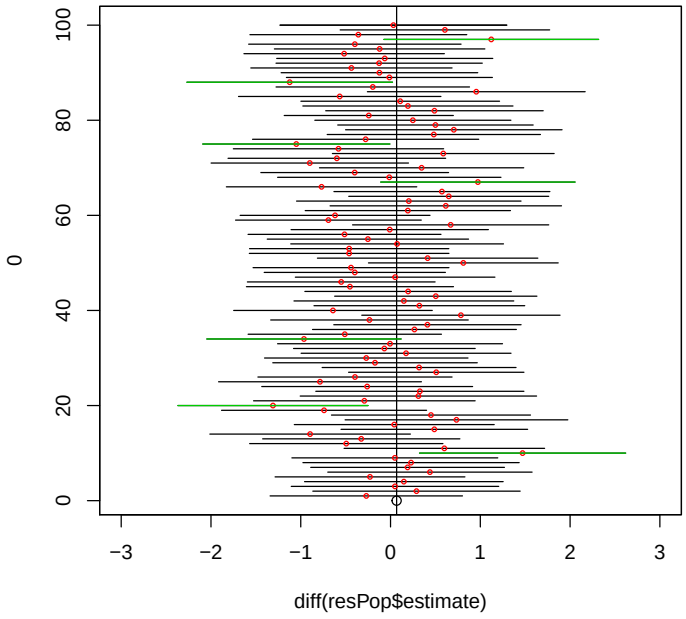


2.2 Sampling Distribution

```
> Pop <- cbind( y=rnorm(1000,0,3)  ,x= rep(c(0,1),500) )
> by(Pop[,1],Pop[,2],mean)

Pop[, 2]: 0
[1] -0.06959212
------------------------------------------------------------
Pop[, 2]: 1
[1] -0.001647941

> resPop <- t.test(y~x, data=Pop)
> mean(Pop[,1])

[1] -0.03562003

> str(resPop)

List of 9
 $ statistic  : Named num -0.375
  ..- attr(*, "names")= chr "t"
 $ parameter  : Named num 998
  ..- attr(*, "names")= chr "df"
 $ p.value    : num 0.708
 $ conf.int   : atomic [1:2] -0.423 0.287
  ..- attr(*, "conf.level")= num 0.95
 $ estimate   : Named num [1:2] -0.06959 -0.00165
  ..- attr(*, "names")= chr [1:2] "mean in group 0" "mean in group 1"
 $ null.value : Named num 0
  ..- attr(*, "names")= chr "difference in means"
 $ alternative: chr "two.sided"
 $ method     : chr "Welch Two Sample t-test"
 $ data.name  : chr "y by x"
 - attr(*, "class")= chr "htest"

> diff(resPop$estimate)

mean in group 1
     0.06794417

> res <- NULL
> for (i in 1:100) {
+   set.seed(i)
+         idx <- sample(1:1000,100)
+         Sampi <- Pop[idx,]
+         tst <- t.test(y~x, data=Sampi)
+         res <- rbind(res,c(t=tst$statistic,pval=tst$p.value,meandif= -as.numeric(diff(tst $esti
+ , confint=tst $ conf.int))
+ }
> head(res)
```

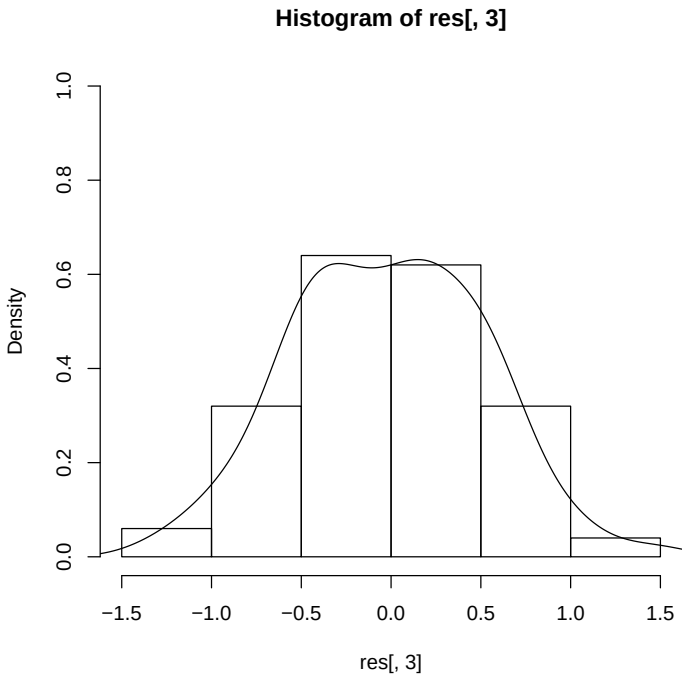
```
             t.t      pval     meandif   confint1  confint2
[1,] -0.50072368 0.6177908 -0.27008246 -1.3416974 0.8015325
[2,]  0.49697250 0.6203492  0.28912122 -0.8657322 1.4439746
[3,]  0.08521118 0.9322775  0.04958162 -1.1059566 1.2051198
[4,]  0.26628918 0.7905761  0.14841962 -0.9576510 1.2544903
[5,] -0.42970345 0.6684284 -0.22873080 -1.2860954 0.8286338
[6,]  0.76630895 0.4455468  0.43885015 -0.6992592 1.5769595
```

```
> plot (diff(resPop$estimate),0,xlim=c(-3,3),ylim=c(1,100))
> abline(v=diff(resPop$estimate))
> box()
> segments(res [i,4], i,res [i,5], i, col=1)
> for (i in 1:100) {
+         segments(res [i,4], i,res [i,5], i, col=1)
+         points (res [i,3], i, pch=1, cex=0.5, col=2)
+ }
> sigres <- which(res[,2] <= 0.1 )
> for (i in 1:length(sigres )) {segments(res [sigres [i],4], sigres [i],res [sigres [i]
>
>
```




```
> #Distribusi beda rata-rata.
> hist(res[,3], prob=T,ylim=c(0,1))
```

```
> lines(density(res[,3]))
>
```



Chapter 3

Mean Vectors Inferences

3.1 One-sample Hotelling's T2-test

```
> #install.packages( c("DescTools", "mvtnorm"))
>
> ## One-sample Hotelling's T2-test
>
> set.seed(1234)
> library(mvtnorm)
> Nj    <- c(15, 25)
> Sigma <- matrix(c(16,-2, -2,9), byrow=TRUE, ncol=2)
> mu1   <- c(-4, 4)
> Y1    <- rmvnorm(Nj[1], mean=mu1, sigma=Sigma)
> head(Y1)
           [,1]      [,2]
[1,] -8.8953835  5.174542
[2,]  0.9991152 -3.315792
[3,] -2.4330012  5.388190
[4,] -6.1363288  2.532387
[5,] -5.9968290  1.503935
[6,] -5.6176262  1.155359
> muH0 <- c(-1, 2)
> library(DescTools)
> HotellingsT2Test(Y1, mu=muH0)

        Hotelling's one sample T2-test

data:  Y1
T.2 = 12.8335, df1 = 2, df2 = 13, p-value = 0.0008374
alternative hypothesis: true location is not equal to c(-1,2)

>
```

3.2 Two-sample Hotelling's T2-test

```
> #Hotelling's T2-test for two independent samples #
> mu2 <- c(3, 3)
> Y2  <- round(rmvnorm(Nj[2], mean=mu2, sigma=Sigma))
> Y12 <- rbind(Y1, Y2)
> IV  <- factor(rep(1:2, Nj))
> HotellingsT2Test(Y12 ~ IV)

        Hotelling's two sample T2-test

data:  Y12 by IV
T.2 = 13.6321, df1 = 2, df2 = 37, p-value = 3.667e-05
alternative hypothesis: true location difference is not equal to c(0,0)

> ##  Hotelling's T2-test for two dependent samples
> N    <- 20
> P    <- 2
> muJK <- c(90, 100, 85, 105)
> Sig  <- 15
> Y1t0 <- rnorm(N, mean=muJK[1], sd=Sig)
> Y1t1 <- rnorm(N, mean=muJK[2], sd=Sig)
> Y2t0 <- rnorm(N, mean=muJK[3], sd=Sig)
> Y2t1 <- rnorm(N, mean=muJK[4], sd=Sig)
> Ydf  <- data.frame(id=factor(rep(1:N, times=P)),
+                    Y1=c(Y1t0, Y1t1),
+                    Y2=c(Y2t0, Y2t1),
+                    IV=factor(rep(1:P, each=N), labels=c("t0", "t1")))
> dfDiff <- aggregate(cbind(Y1, Y2) ~ id, data=Ydf, FUN=diff)
> DVdiff <- data.matrix(dfDiff[ , -1])
> muH0   <- c(0, 0)
> HotellingsT2Test(DVdiff, mu=muH0)

        Hotelling's one sample T2-test

data:  DVdiff
T.2 = 15.9576, df1 = 2, df2 = 18, p-value = 0.0001031
alternative hypothesis: true location is not equal to c(0,0)

>
```

Chapter 4

Mid Tem Exam

1. Di dalam R terdapat data yang bernama state.x77 (hint: gunakan perintah data(state)). Dalam data tersebut tedapat 8 variable yang penjelasan masing-masing variable dapat dilihat di help R. Lakukan: a. Explorasi data tersebut, tuliskan kesimpulan anda mengenai data ini.

b. Uji apakah Illiteracy dan Life Exp mengikuti distribusi bivariate normal. Gambarkan plot yang diperlukan untuk uji tsb.

c. Uji apakah rata-rata illeterasi dan life.exp = [1.5, 70]. Jelaskan hasilnya dan berikan kesimpulannya.

d. Uji beda rata-rata Illiteracy, Life Exp dan Murder dari high and low income state (Negara bagian). Gunakan median income sbg cut-off (hint: buat group dengan gunakan fungsi ifelse). Jelaskan hasilnya dan berikan kesimpulannya.

e. Lakukan soal c dengan beda antara Negara bagian (state) yang jumlah penduduknya sangat sedikit (< =Q1) dan jumlah penduduknya yang sangat banyak (> =Q3) (gunakan fungsi quantile)

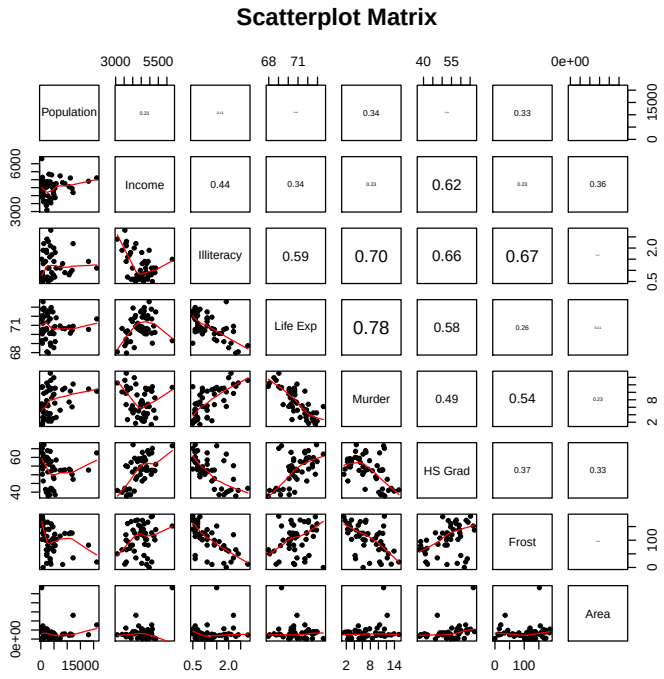
```
> library(DescTools)
> ## Load The Dataset ##
>
> data(state)
> head(state.x77)
```

```
           Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
California      21198   5114        1.1    71.71   10.3    62.6    20 156361
Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

```
> summary(state.x77)
```

```
   Population        Income       Illiteracy       Life Exp
 Min.   :  365   Min.   :3098   Min.   :0.500   Min.   :67.96
 1st Qu.: 1080   1st Qu.:3993   1st Qu.:0.625   1st Qu.:70.12
 Median : 2838   Median :4519   Median :0.950   Median :70.67
 Mean   : 4246   Mean   :4436   Mean   :1.170   Mean   :70.88
 3rd Qu.: 4968   3rd Qu.:4814   3rd Qu.:1.575   3rd Qu.:71.89
 Max.   :21198   Max.   :6315   Max.   :2.800   Max.   :73.60
     Murder          HS Grad          Frost             Area
 Min.   : 1.400   Min.   :37.80   Min.   :  0.00   Min.   :  1049
 1st Qu.: 4.350   1st Qu.:48.05   1st Qu.: 66.25   1st Qu.: 36985
 Median : 6.850   Median :53.25   Median :114.50   Median : 54277
 Mean   : 7.378   Mean   :53.11   Mean   :104.46   Mean   : 70736
 3rd Qu.:10.675   3rd Qu.:59.15   3rd Qu.:139.75   3rd Qu.: 81163
 Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.   :566432
```

```
> panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)
+ {
+     usr <- par("usr"); on.exit(par(usr))
+     par(usr = c(0, 1, 0, 1))
+     r <- abs(cor(x, y))
+     txt <- format(c(r, 0.123456789), digits=digits)[1]
+     txt <- paste(prefix, txt, sep="")
+     if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
+     text(0.5, 0.5, txt, cex = cex.cor * r)
+ }
> pairs(~., data=state.x77,
+       lower.panel=panel.smooth, upper.panel=panel.cor,
+       pch=20, main=" Scatterplot Matrix")
>
```




```
> ### One sample multivariate t-test ###
>
>
>
> muH0 <- c(1.5, 70)
> X1 <- state.x77[,c(3,4)]
> summary(X1)

   Illiteracy       Life Exp
 Min.   :0.500   Min.   :67.96
 1st Qu.:0.625   1st Qu.:70.12
 Median :0.950   Median :70.67
 Mean   :1.170   Mean   :70.88
 3rd Qu.:1.575   3rd Qu.:71.89
 Max.   :2.800   Max.   :73.60

> HotellingsT2Test(X1, mu=muH0)

        Hotelling's one sample T2-test

data:  X1
T.2 = 11.4053, df1 = 2, df2 = 48, p-value = 8.86e-05
alternative hypothesis: true location is not equal to c(1.5,70)
```

```
> HotellingsT2Test(X1, mu=muH0, test="chi")

        Hotelling's one sample T2-test

data:  X1
T.2 = 23.2858, df = 2, p-value = 8.781e-06
alternative hypothesis: true location is not equal to c(1.5,70)

>

> ###  Two sample t test ###
>
> ## make groups ##
>
>
> grp <- ifelse (state.x77[,2] <= median(state.x77[,2]) ,0,1)
> X2 <- state.x77[,c(3:5)]
> HotellingsT2Test(X2 ~ grp)

        Hotelling's two sample T2-test

data:  X2 by grp
T.2 = 2.3878, df1 = 3, df2 = 46, p-value = 0.08108
alternative hypothesis: true location difference is not equal to c(0,0,0)

> HotellingsT2Test(X2 ~ grp, test="chi")

        Hotelling's two sample T2-test

data:  X2 by grp
T.2 = 7.4749, df = 3, p-value = 0.05821
alternative hypothesis: true location difference is not equal to c(0,0,0)

> ## Case of lower population state vs. higher population state ##
>
>
> ## make groups ##
> grp2 <- ifelse (state.x77[,1] <= quantile(state.x77[,1],0.25) ,0, ifelse(state.x77[,1
> X2 <- state.x77[,c(3:5)]
> HotellingsT2Test(X2 ~ grp2)

        Hotelling's two sample T2-test

data:  X2 by grp2
T.2 = 3.1595, df1 = 3, df2 = 22, p-value = 0.04494
alternative hypothesis: true location difference is not equal to c(0,0,0)

> HotellingsT2Test(X2 ~ grp2, test="chi")
```

```
        Hotelling's two sample T2-test

data:  X2 by grp2
T.2 = 10.3402, df = 3, p-value = 0.01589
alternative hypothesis: true location difference is not equal to c(0,0,0)

> ## SPlit into two groups ###
>
> lowstate <-   state.x77[state.x77[,1] <= quantile(state.x77[,1],0.25) ,c(3:5)]
> histate <-   state.x77[state.x77[,1] > quantile(state.x77[,1],0.75) ,c(3:5)]
>
>
>
```

2. Suatu studi yang dirancang untuk mengetahui efektifitas program penanganan stres. Program tersebut bertujuan mengurangi tingkat dan ciri kecemasan pada mahasiswa. Untuk tujuan studi ini, dipilih secara acak 24 mahasiswa yang mempunyai kondisi stres relatif mirip dan dibagi menjadi 2 kelompok yaitu kelompok kontrol (KK) dan kelompok dalam program (KP). Setelah beberapa waktu tertentu, diketahui nilai tingkat dan ciri kecemasan sampel mahasiswa disajikan pada Tabel I. Pada tingkat signifikansi 5% dan dengan menggunakan data pada Tabel I:

a. Uji pula efektifitas program penanganan stres tersebut.

b. Susunlah selang kepercayaan simultan Hotelling atau Bonferroni, untuk mengetahui variabel yang mendukung hasil pada butir a. Berikan ulasan mengenai nilai selang kepercayaan tersebut!

```
> X1 <- c(41,48,34,31,26,37,44,53,46,34,33,50)
> X2 <- c(38,41,33,40,23,31,32,47,41,38,39,45)
> KK  <- cbind(X1,X2)
> head(KK)

     X1 X2
[1,] 41 38
[2,] 48 41
[3,] 34 33
[4,] 31 40
[5,] 26 23
[6,] 37 31

> X1 <- c(46,47,39,28,35,40,46,58,47,39,36,54)
> X2 <- c(35,50,36,38,19,30,45,53,48,39,41,40)
> KP  <- cbind(X1,X2)
> head(KP)

     X1 X2
[1,] 46 35
```

```
[2,] 47 50
[3,] 39 36
[4,] 28 38
[5,] 35 19
[6,] 40 30

> ### Using T Hotteling function in R ###
> diffKP <- KK-KP
> muH0   <- c(0, 0)
> ## F test ##
> HotellingsT2Test(diffKP , mu=muH0)

        Hotelling's one sample T2-test

data:  diffKP
T.2 = 8.9804, df1 = 2, df2 = 10, p-value = 0.005851
alternative hypothesis: true location is not equal to c(0,0)

> ## Chi-squared test#
> HotellingsT2Test(diffKP , mu=muH0, test="chi")

        Hotelling's one sample T2-test

data:  diffKP
T.2 = 19.7569, df = 2, p-value = 5.127e-05
alternative hypothesis: true location is not equal to c(0,0)

> n <- nrow(diffKP)
> p <- ncol(diffKP)
> covDiff <- cov(diffKP )
> dBar <- colMeans(diffKP )
> T2 <- n*t(dBar )%*% solve(covDiff )%*% dBar
> T2

         [,1]
[1,] 19.75695

> ## Confidence Interval ##
>
> ## Hotelling ##
> for (i in 1:2) {
+   se <- sqrt((p*(n-1)/(n-p)) * qf(1-0.05,p,n-p)) * sqrt(covDiff [i,i]/n)
+         print(paste(dBar[i]-se, "< D",i, "< ", dBar[i]+se))
+ }

[1] "-5.90510510447297 < D 1 <  -0.428228228860367"
[1] "-7.04248131015674 < D 2 <  2.70914797682341"
```

```
> ## Bonferroni ##
> for (i in 1:2) {
+         se <- qt( 1- (.05/(2*p) ), (n-1)) * sqrt(covDiff [i,i]/n)
+         print(paste(dBar[i]-se, "< D",i, "< ", dBar[i]+se))
+ }

[1] "-5.53023626753058 < D 1 <  -0.803097065802749"
[1] "-6.37502392301315 < D 2 <  2.04169058967982"

>
>
>
```

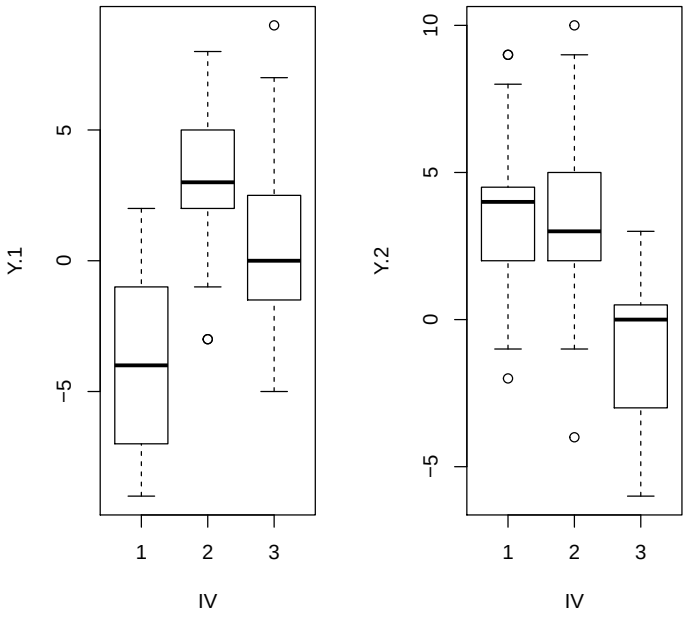
Chapter 5

MANOVA

5.1 One-way MANOVA

```
> library(car)
> library(mvtnorm)
> set.seed(123)
> P     <- 3
> Nj    <- c(15, 25, 20)
> Sigma <- matrix(c(16,-2, -2,9), byrow=TRUE, ncol=2)
> mu11  <- c(-4,  4)
> mu21  <- c( 3,  3)
> mu31  <- c( 1, -1)
> ## Generate Multvariate Norm
> Y11 <- round(rmvnorm(Nj[1], mean=mu11, sigma=Sigma))
> Y21 <- round(rmvnorm(Nj[2], mean=mu21, sigma=Sigma))
> Y31 <- round(rmvnorm(Nj[3], mean=mu31, sigma=Sigma))
> dataMan1 <- data.frame(Y =rbind(Y11, Y21, Y31),
+                        IV=factor(rep(1:P, Nj)))
> head(dataMan1)

  Y.1 Y.2 IV
1  -6   3  1
2   2   4  1
3  -4   9  1
4  -2   0  1
5  -7   3  1
6   1   5  1

> par(mfrow=c(1,2))
> plot(Y.1 ~ IV, data=dataMan1 )
> plot(Y.2 ~ IV, data=dataMan1 )
> #dev.off()
```

```
> ## One Way ANOVA
>
> manRes1 <- manova(cbind(Y.1, Y.2) ~ IV, data=dataMan1)
> summary(manRes1, test="Wilks")

          Df   Wilks approx F num Df den Df    Pr(>F)
IV         2 0.38675   17.024      4    112 6.222e-11 ***
Residuals 57
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(manRes1, test="Pillai")

          Df Pillai approx F num Df den Df    Pr(>F)
IV         2 0.7519   17.169      4    114 4.767e-11 ***
Residuals 57
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(manRes1, test="Roy")

          Df    Roy approx F num Df den Df    Pr(>F)
IV         2 0.7476   21.307      2     57 1.231e-07 ***
Residuals 57
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary.aov(manRes1)

 Response Y.1 :
            Df Sum Sq Mean Sq F value    Pr(>F)
IV           2 462.76 231.380  21.204 1.306e-07 ***
Residuals   57 621.97  10.912
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Response Y.2 :
            Df Sum Sq Mean Sq F value    Pr(>F)
IV           2 244.41  122.20  13.669 1.415e-05 ***
Residuals   57 509.59    8.94
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

>
```




```
> ## MAnova for example STATE Data ##
> ## Divide into 4 groups ##
> quat  <- quantile(state.x77[,1],c(.25, .5, .75))
> grp4 <- cut(state.x77[,1], breaks= c(-Inf,quat ,Inf), labels= 1:4)
> X2 <- state.x77[,c(3:5)]
> X2 <- data.frame(X2 ,grp4)
> head(X2)

           Illiteracy Life.Exp Murder grp4
Alabama           2.1    69.05   15.1    3
Alaska            1.5    69.31   11.3    1
Arizona           1.8    70.55    7.8    2
Arkansas          1.9    70.66   10.1    2
California        1.1    71.71   10.3    4
Colorado          0.7    72.06    6.8    2

> par(mfrow=c(1,2))
> plot(Illiteracy ~ grp4, data=X2 )
> plot(Murder ~ grp4, data=X2 )
> ManRes <- manova(cbind(Illiteracy, Murder) ~ grp4, data=X2 )
> summary(ManRes)

          Df  Pillai approx F num Df den Df Pr(>F)
```

```
grp4       3 0.21257   1.8235      6     92  0.103
Residuals 46
```

```
> summary(ManRes, test="Wilks")
```

```
          Df   Wilks approx F num Df den Df Pr(>F)
grp4       3 0.79802   1.7913      6     90 0.1097
Residuals 46
```

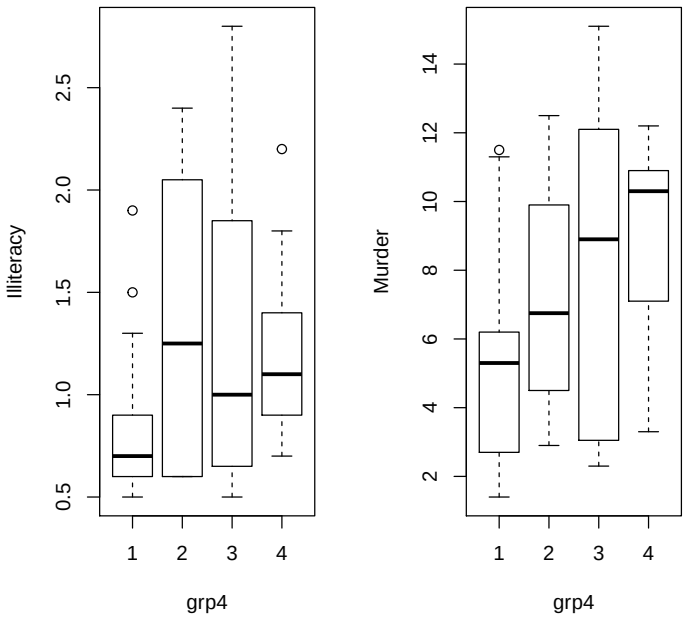
```
> summary(ManRes, test="Roy")
```

```
          Df     Roy approx F num Df den Df  Pr(>F)
grp4       3 0.15343   2.3526      3     46 0.08445 .
Residuals 46
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> summary(ManRes, test="Hotelling-Lawley")
```

```
          Df Hotelling-Lawley approx F num Df den Df Pr(>F)
grp4       3          0.23985   1.7589      6     88 0.1169
Residuals 46
```

```
>
>
>
```

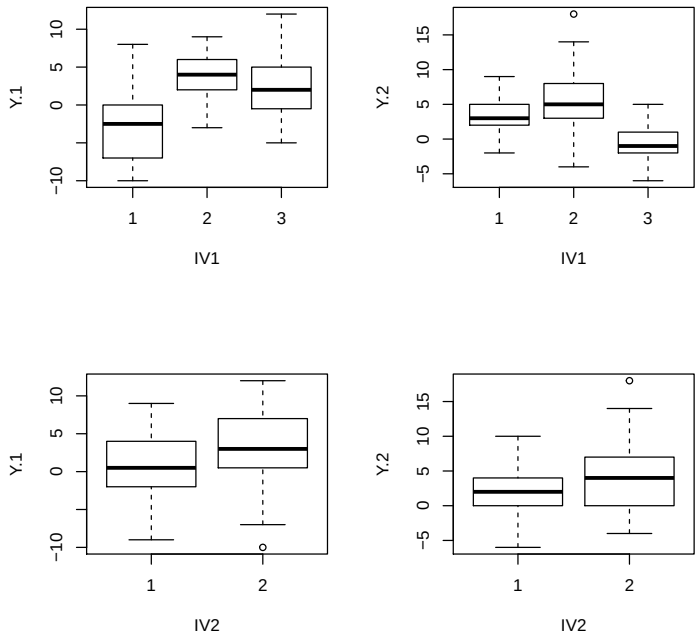


5.2 Two-way MANOVA

```
> Q    <- 2
> mu12 <- c(-1,  4)
> mu22 <- c( 4,  8)
> mu32 <- c( 4,  0)
> library(mvtnorm)
> Y12  <- round(rmvnorm(Nj[1], mean=mu12, sigma=Sigma))
> Y22  <- round(rmvnorm(Nj[2], mean=mu22, sigma=Sigma))
> Y32  <- round(rmvnorm(Nj[3], mean=mu32, sigma=Sigma))
> dataMan2 <- data.frame(Y  =rbind(Y11, Y21, Y31, Y12, Y22, Y32),
+                        IV1=factor(rep(rep(1:P, Nj), Q)),
+                        IV2=factor(rep(1:Q, each=sum(Nj))))
> head(dataMan2)

  Y.1 Y.2 IV1 IV2
1  -6   3   1   1
2   2   4   1   1
3  -4   9   1   1
4  -2   0   1   1
5  -7   3   1   1
6   1   5   1   1

> par(mfrow=c(2,2))
> plot(Y.1 ~ IV1, data=dataMan2 )
> plot(Y.2 ~ IV1, data=dataMan2 )
> plot(Y.1 ~ IV2, data=dataMan2 )
> plot(Y.2 ~ IV2, data=dataMan2 )
> #dev.off()
>
> manRes2 <- manova(cbind(Y.1, Y.2) ~ IV1*IV2, data=dataMan2)
> summary(manRes2, test="Pillai")

           Df  Pillai approx F num Df den Df    Pr(>F)
IV1         2 0.81891   39.521      4    228 < 2.2e-16 ***
IV2         1 0.24055   17.896      2    113 1.771e-07 ***
IV1:IV2     2 0.14550    4.472      4    228  0.001693 **
Residuals 114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(manRes2, test="Roy")

           Df     Roy approx F num Df den Df    Pr(>F)
IV1         2 1.10198   62.813      2    114 < 2.2e-16 ***
IV2         1 0.31675   17.896      2    113 1.771e-07 ***
IV1:IV2     2 0.16205    9.237      2    114 0.0001915 ***
```

```
Residuals 114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> summary.aov(manRes2)

 Response Y.1 :
             Df  Sum Sq Mean Sq F value    Pr(>F)
IV1           2  782.98  391.49 26.9184 2.661e-10 ***
IV2           1  182.53  182.53 12.5508  0.000575 ***
IV1:IV2       2   39.73   19.86  1.3657  0.259338
Residuals   114 1657.96   14.54
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Response Y.2 :
             Df Sum Sq Mean Sq F value    Pr(>F)
IV1           2 891.47  445.74 51.6958 < 2.2e-16 ***
IV2           1 118.01  118.01 13.6864 0.0003340 ***
IV1:IV2       2 158.57   79.29  9.1955 0.0001985 ***
Residuals   114 982.94    8.62
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
> ## One-way MAnova ##
> ## Divide into 4 groups ##
>
> grp4 <- cut(state.x77[,1], breaks= c(-Inf,quat ,Inf), labels= 1:4)
> X2 <- state.x77[,c(3:5)]
> X2 <- data.frame(X2 ,grp4)
> head(X2)

           Illiteracy Life.Exp Murder grp4
Alabama           2.1    69.05   15.1    3
Alaska            1.5    69.31   11.3    1
Arizona           1.8    70.55    7.8    2
Arkansas          1.9    70.66   10.1    2
California        1.1    71.71   10.3    4
Colorado          0.7    72.06    6.8    2

> ManRes <- manova(cbind(Illiteracy, Murder) ~ grp4, data=X2 )
> summary(ManRes)

          Df  Pillai approx F num Df den Df Pr(>F)
grp4       3 0.21257   1.8235      6     92  0.103
Residuals 46

> summary(ManRes, test="Wilks")

          Df   Wilks approx F num Df den Df Pr(>F)
grp4       3 0.79802   1.7913      6     90 0.1097
Residuals 46

> summary(ManRes, test="Roy")

          Df     Roy approx F num Df den Df  Pr(>F)
grp4       3 0.15343   2.3526      3     46 0.08445 .
Residuals 46
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(ManRes, test="Hotelling-Lawley")

          Df Hotelling-Lawley approx F num Df den Df Pr(>F)
grp4       3          0.23985   1.7589      6     88 0.1169
Residuals 46

> ## Two-way MAnova ##
>
> X3 <- cbind(X2 ,grp)
> head(X3)
```

```
           Illiteracy Life.Exp Murder grp4 grp
Alabama           2.1    69.05   15.1    3   0
Alaska            1.5    69.31   11.3    1   1
Arizona           1.8    70.55    7.8    2   1
Arkansas          1.9    70.66   10.1    2   0
California        1.1    71.71   10.3    4   1
Colorado          0.7    72.06    6.8    2   1
```

```
> ManRes2 <- manova(cbind(Illiteracy, Murder) ~ grp*grp4, data=X3 )
> summary(ManRes2)

          Df  Pillai approx F num Df den Df  Pr(>F)
grp        1 0.11343   2.6228      2     41 0.08475 .
grp4       3 0.24977   1.9979      6     84 0.07496 .
grp:grp4   3 0.31727   2.6396      6     84 0.02148 *
Residuals 42
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> summary(ManRes2, test="Wilks")

          Df   Wilks approx F num Df den Df  Pr(>F)
grp        1 0.88657   2.6228      2     41 0.08475 .
grp4       3 0.76155   1.9941      6     82 0.07581 .
grp:grp4   3 0.69380   2.7410      6     82 0.01773 *
Residuals 42
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> summary(ManRes2, test="Roy")

          Df     Roy approx F num Df den Df   Pr(>F)
grp        1 0.12794   2.6228      2     41 0.084747 .
grp4       3 0.23496   3.2895      3     42 0.029742 *
grp:grp4   3 0.38386   5.3740      3     42 0.003189 **
Residuals 42
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> summary(ManRes2, test="Hotelling-Lawley")

          Df Hotelling-Lawley approx F num Df den Df  Pr(>F)
grp        1          0.12794   2.6228      2     41 0.08475 .
grp4       3          0.29824   1.9883      6     80 0.07698 .
grp:grp4   3          0.42540   2.8360      6     80 0.01484 *
Residuals 42
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
>
>
```

Chapter 6

Principal Component Analysis

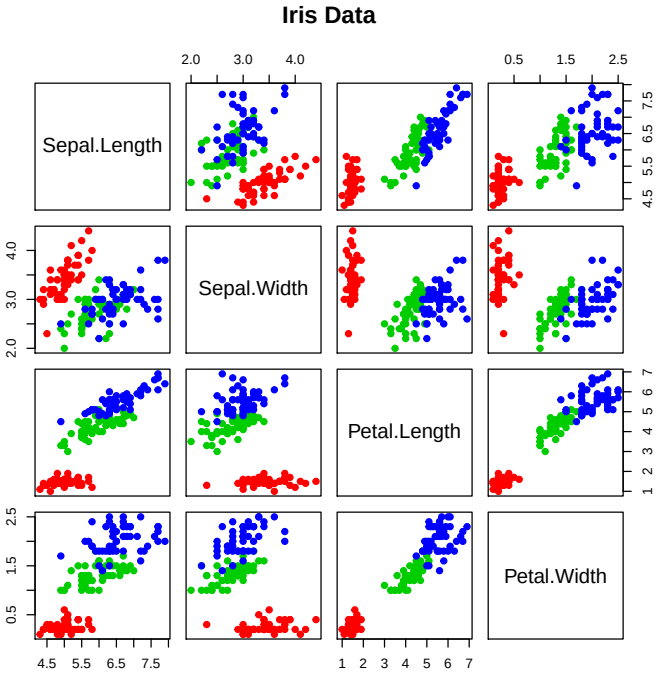
6.1 Example 1: Iris Data

The Iris data, collected over several years by Edgar Anderson was used to show that these measurements could be used to differentiate between species of irises. That data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

```
> # Load data
> data(iris)
> dim(iris)

[1] 150   5

> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> table(iris[,5])

    setosa versicolor  virginica
        50         50         50
```

```
> pairs(iris[1:4],main="Iris Data", pch=19, col=as.numeric(iris$Species)+1)
> #To examine variability of all numeric variables
> sapply(iris[1:4],var)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.6856935    0.1899794    3.1162779    0.5810063

> # maybe this range of variability is big in this context.
> #Thus, we will use the correlation matrix
>
>
>
> ## Use Correlation matrix for PCA
> pca <- prcomp(iris[,1:4],scale=T)
> pca

Standard deviations:
[1] 1.7083611 0.9560494 0.3830886 0.1439265

Rotation:
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971

> summary(pca)

Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000

> #pcaNo <- prcomp(iris[,1:4],scale=F)
> #summary(pcaNo)
> #pcaNo$rotation
>
>
>
> ## The following code would give the same result
> pca2 <- princomp(iris[,1:4], cor=T)
> summary(pca2,loadings=T)

Importance of components:
                          Comp.1    Comp.2     Comp.3      Comp.4
Standard deviation     1.7083611 0.9560494 0.38308860 0.143926497
```

```
Proportion of Variance 0.7296245 0.2285076 0.03668922 0.005178709
Cumulative Proportion  0.7296245 0.9581321 0.99482129 1.000000000

Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length  0.521 -0.377  0.720  0.261
Sepal.Width  -0.269 -0.923 -0.244 -0.124
Petal.Length  0.580        -0.142 -0.801
Petal.Width   0.565        -0.634  0.524



> #plot of variance of each PCA/ ScreePlot
>
> screeplot(pca, type="lines",col=3)
> pca$rotation



                   PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971



> pca2$loadings



Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length  0.521 -0.377  0.720  0.261
Sepal.Width  -0.269 -0.923 -0.244 -0.124
Petal.Length  0.580        -0.142 -0.801
Petal.Width   0.565        -0.634  0.524

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00



>
```

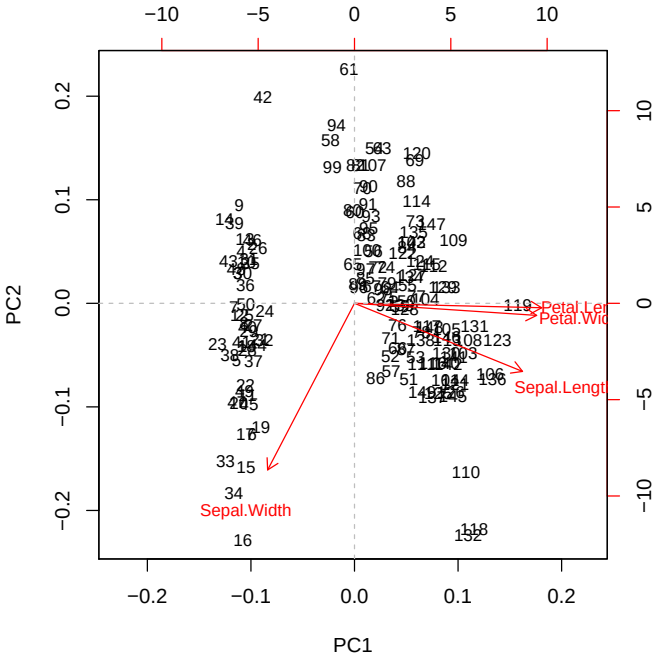


The weights of the PC1 are similar except the associate to Sepal.Width variable that is negative. This component discriminate on one side the Sepal.Width and on the other side the rest of variables (see biplot). This one principal component accounts for over 72% of the variability in the data.

All weights on the second principal component are negative. Thus the PC2 might seem considered as an overall size measurement. When the iris has larger sepal and petal values than average, the PC2 will be smaller than average. This component explain the 23% of the variability.

The following figure show the first two components and the observations on the same diagram, which helps to interpret the factorial axes while looking at observations location.

```
> #biplot of first two principal components
> biplot(pca,cex=0.8)
> abline(h = 0, v = 0, lty = 2, col = 8)
```



6.2 Example 2: Head Size Data

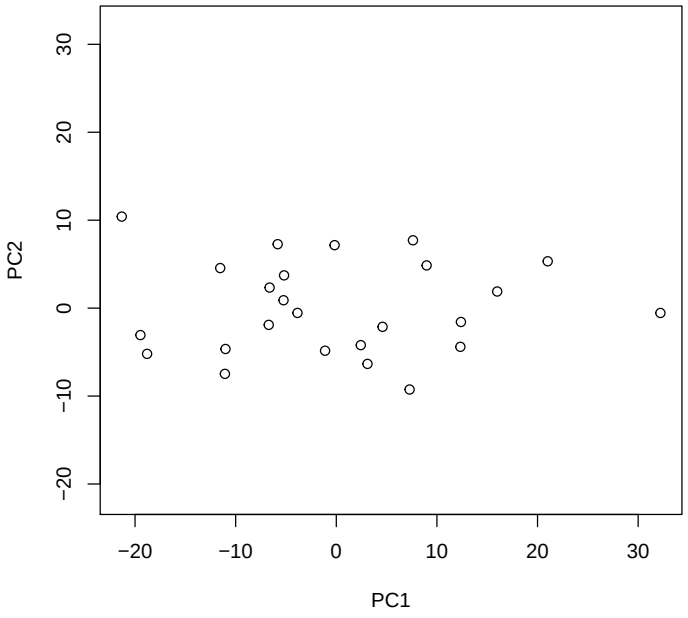
The data contains head lengths and head breadths (in millimetres) for each of the first two adult sons in 25 families.

```
> library(boot)
> head(frets)

   l1  b1  l2  b2
1 191 155 179 145
2 195 149 201 152
3 181 148 185 149
4 183 153 188 149
5 176 144 171 142
6 208 157 192 152

> head_dat <- frets[, c("l1", "l2")]
> colMeans(head_dat)

    l1     l2
185.72 183.84

> cov(head_dat)
```

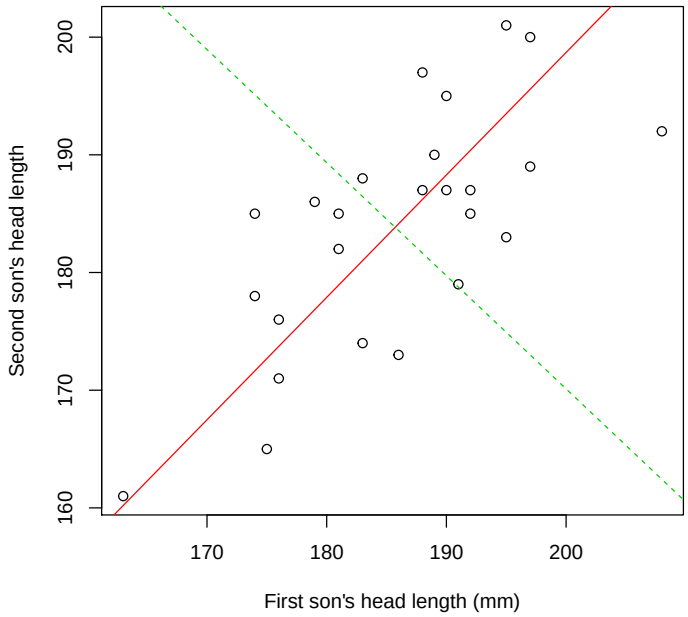
```
         l1        l2
l1 95.29333  69.66167
l2 69.66167 100.80667

> ## Eigen Vectors & Eigen Values ##
>
> eigen(cov(head_dat))

$values
[1] 167.76619  28.33381

$vectors
          [,1]       [,2]
[1,] 0.6929858 -0.7209512
[2,] 0.7209512  0.6929858

> head_pca <- prcomp(x = head_dat)
> str(head_pca)

List of 5
 $ sdev    : num [1:2] 12.95 5.32
 $ rotation: num [1:2, 1:2] -0.693 -0.721 0.721 -0.693
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "l1" "l2"
  .. ..$ : chr [1:2] "PC1" "PC2"
 $ center  : Named num [1:2] 186 184
  ..- attr(*, "names")= chr [1:2] "l1" "l2"
 $ scale   : logi FALSE
 $ x       : num [1:25, 1:2] -0.17 -18.8 2.43 -1.11 15.99 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:25] "1" "2" "3" "4" ...
  .. ..$ : chr [1:2] "PC1" "PC2"
 - attr(*, "class")= chr "prcomp"

> print(summary(head_pca),loadings=TRUE)

Importance of components:
                          PC1    PC2
Standard deviation     12.9525 5.3230
Proportion of Variance  0.8555 0.1445
Cumulative Proportion   0.8555 1.0000

> head(head_pca$x)

          PC1       PC2
1  -0.1695614  7.160674
2 -18.8024312 -5.201210
```

```
3   2.4345897 -4.206753
4  -1.1142355 -4.843808
5  15.9928357  1.890292
6 -21.3226862 10.408028
```

```
> lim <- range(head_pca$x[,1])
> plot(head_pca$x, xlim=lim, ylim=lim)
>
```




```
> a1 <- 183.84-0.721*185.72/0.693
> b1 <- 0.721/0.693
> a2 <- 183.84-(-0.693*185.72/0.721)
> b2 <- -0.693/0.721
> plot(head_dat, xlab = "First son's head length (mm)", ylab = "Second son's head length")
> abline(a1, b1, col=2)
> abline(a2, b2, lty = 2, col=3)
>
```



6.3 Example 3: Heptathlon Data

```
> library(HSAUR2)
> head(heptathlon)
```

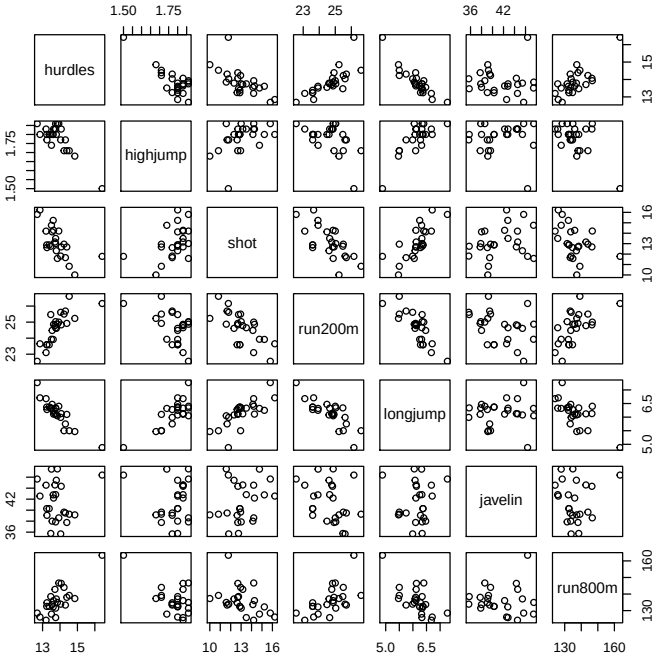
```
                   hurdles highjump  shot run200m longjump javelin run800m
Joyner-Kersee (USA)   12.69     1.86 15.80   22.56     7.27   45.66  128.51
John (GDR)            12.85     1.80 16.23   23.65     6.71   42.56  126.12
Behmer (GDR)          13.20     1.83 14.20   23.10     6.68   44.54  124.20
Sablovskaite (URS)    13.61     1.80 15.23   23.92     6.25   42.78  132.24
Choubenkova (URS)     13.51     1.74 14.76   23.93     6.32   47.46  127.90
Schulz (GDR)          13.75     1.83 13.50   24.65     6.33   42.82  125.79
                   score
Joyner-Kersee (USA)  7291
John (GDR)           6897
Behmer (GDR)         6858
Sablovskaite (URS)   6540
Choubenkova (URS)    6540
Schulz (GDR)         6411
```

```
>  score <- which(colnames(heptathlon) == "score")
>  heptathlon2 <- heptathlon[,-score]
```

```
>  round(cor(heptathlon2), 2)
```

```
         hurdles highjump  shot run200m longjump javelin run800m
hurdles     1.00    -0.81 -0.65    0.77    -0.91   -0.01    0.78
highjump   -0.81     1.00  0.44   -0.49     0.78    0.00   -0.59
shot       -0.65     0.44  1.00   -0.68     0.74    0.27   -0.42
run200m     0.77    -0.49 -0.68    1.00    -0.82   -0.33    0.62
longjump   -0.91     0.78  0.74   -0.82     1.00    0.07   -0.70
javelin    -0.01     0.00  0.27   -0.33     0.07    1.00    0.02
run800m     0.78    -0.59 -0.42    0.62    -0.70    0.02    1.00
```

```
>  plot(heptathlon2)
>
>
```

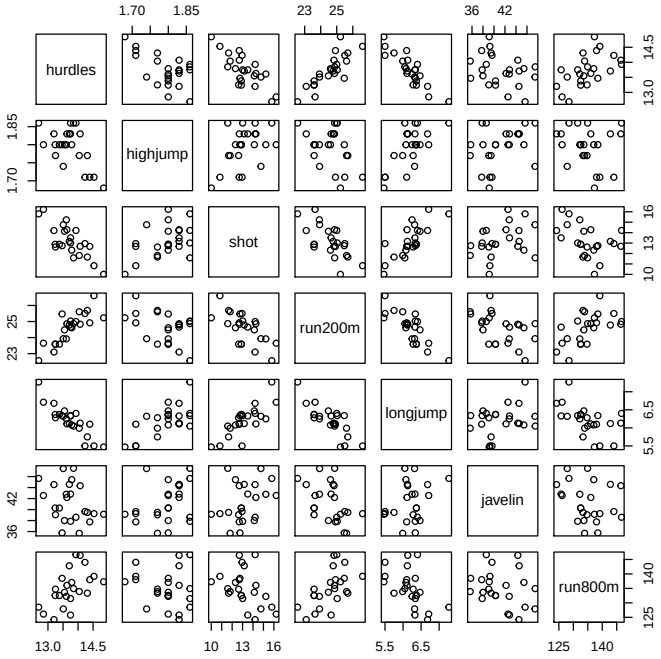



```
> heptathlon2<- heptathlon2[-grep("PNG", rownames(heptathlon2)),]
> round(cor(heptathlon2), 2)
```

```
         hurdles highjump  shot run200m longjump javelin run800m
hurdles     1.00    -0.58 -0.77    0.83    -0.89   -0.33    0.56
highjump   -0.58     1.00  0.46   -0.39     0.66    0.35   -0.15
shot       -0.77     0.46  1.00   -0.67     0.78    0.34   -0.41
run200m     0.83    -0.39 -0.67    1.00    -0.81   -0.47    0.57
```

```
longjump   -0.89      0.66  0.78   -0.81      1.00     0.29   -0.52
javelin    -0.33      0.35  0.34   -0.47      0.29     1.00   -0.26
run800m     0.56     -0.15 -0.41    0.57     -0.52    -0.26    1.00
```

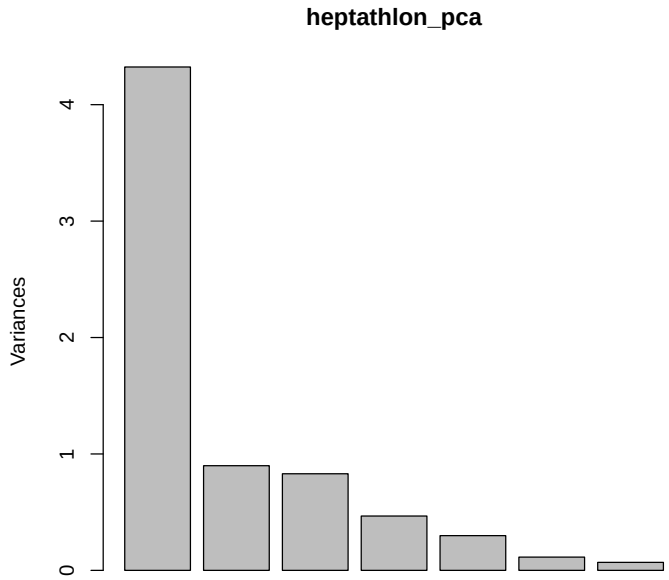
```
>  plot(heptathlon2)
>
```



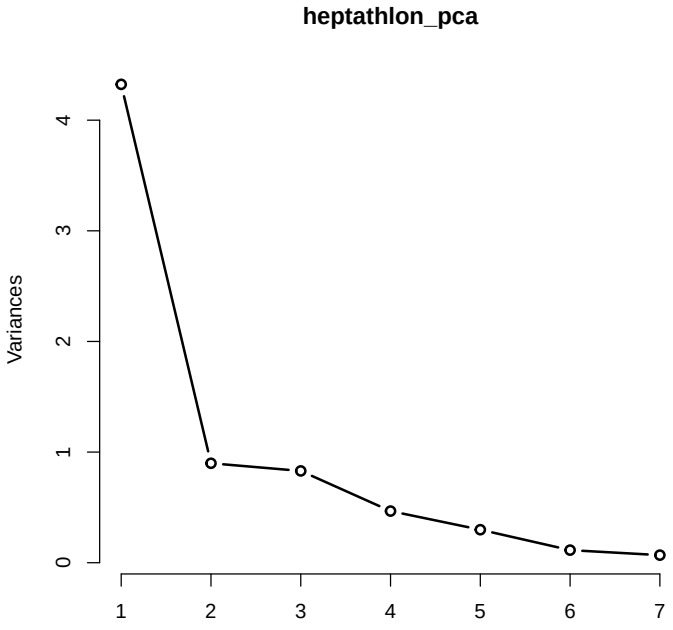

```
> heptathlon_pca <- prcomp(heptathlon2, scale = TRUE)
> heptathlon_pca

Standard deviations:
[1] 2.0793370 0.9481532 0.9109016 0.6831967 0.5461888 0.3374549 0.2620420

Rotation:
                PC1         PC2        PC3         PC4         PC5         PC6
hurdles   0.4503876 -0.05772161 -0.1739345  0.04840598  0.19889364 -0.84665086
highjump -0.3145115 -0.65133162  0.2088272  0.55694554  0.07076358 -0.09007544
shot     -0.4024884 -0.02202088  0.1534709 -0.54826705  0.67166466 -0.09886359
run200m   0.4270860 -0.18502783  0.1301287  0.23095946  0.61781764  0.33279359
longjump -0.4509639 -0.02492486  0.2697589  0.01468275 -0.12151793 -0.38294411
javelin  -0.2423079 -0.32572229 -0.8806995 -0.06024757  0.07874396  0.07193437
run800m   0.3029068 -0.65650503  0.1930020 -0.57418128 -0.31880178  0.05217664
                PC7
```

```
hurdles   0.06961672
highjump  0.33155910
shot      0.22904298
run200m  -0.46971934
longjump -0.74940781
javelin  -0.21108138
run800m  -0.07718616
```

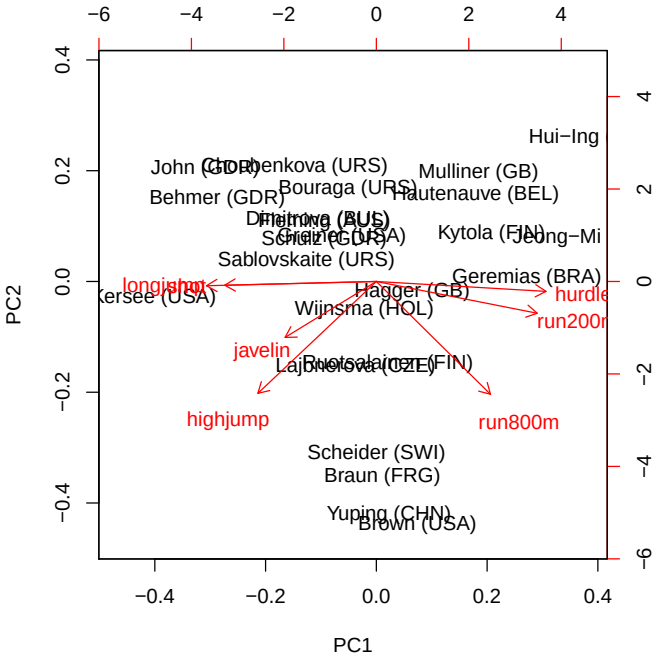
```
> summary(heptathlon_pca)

Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     2.0793 0.9482 0.9109 0.68320 0.54619 0.33745 0.26204
Proportion of Variance 0.6177 0.1284 0.1185 0.06668 0.04262 0.01627 0.00981
Cumulative Proportion  0.6177 0.7461 0.8646 0.93131 0.97392 0.99019 1.00000

> plot(heptathlon_pca)
>
```




```
> plot(heptathlon_pca,type="l", lwd=2)
>
```




```
> biplot(heptathlon_pca)
>
```


−6
−4
−2
0
2
4
0.4
0.2
0.0
−0.2
−0.4
PC2
PC1
−0.4
−0.2
0.0
0.2
0.4
4
2
0
−2
−4
−6
Hui−Ing
John (GDR)
Choubenkova (URS)
Mulliner (GB)
Bouraga (URS)
Behmer (GDR)
Hautenauve (BEL)
Kytola (FIN)
Jeong−Mi
Sablovskaite (URS)
Geremias (BRA)
Hagger (GB)
hurdle
Wijnsma (HOL)
run200r
javelin
highjump
run800m
Scheider (SWI)
Braun (FRG)
Yuping (CHN)
Brown (USA)

Chapter 7

Factor Analysis

7.1 Example 1

This dataset contains a hypothetical sample of 300 responses on 6 items from a survey of college students' favorite subject matter. The items range in value from 1 to 5, which represent a scale from Strongly Dislike to Strongly Like. Our 6 items asked students to rate their liking of different college subject matter areas, including biology (BIO), geology (GEO), chemistry (CHEM), algebra (ALG), calculus (CALC), and statistics (STAT).

```
> #dataFA1=read.csv("C:/Users/Administrator/Documents/MUltivariate_Data_Analysis/dataset_#explora
>
> dataFA1=read.csv("dataset_exploratoryFactorAnalysis.csv")
> dim(dataFA1)

[1] 300   6

> head(dataFA1)

  BIO GEO CHEM ALG CALC STAT
1   1   1    1   1    1    1
2   4   4    3   4    4    4
3   2   1    3   4    1    1
4   2   3    2   4    4    3
5   3   1    2   2    3    4
6   1   1    1   4    4    4

> cordata=cor(dataFA1)
> cordata

           BIO       GEO      CHEM       ALG      CALC      STAT
BIO  1.0000000 0.6822208 0.7470278 0.1153204 0.2134271 0.2028315
GEO  0.6822208 1.0000000 0.6814857 0.1353557 0.2045215 0.2316288
```

```
CHEM 0.7470278 0.6814857 1.0000000 0.0838225 0.1364251 0.1659747
ALG  0.1153204 0.1353557 0.0838225 1.0000000 0.7709303 0.4094324
CALC 0.2134271 0.2045215 0.1364251 0.7709303 1.0000000 0.5073147
STAT 0.2028315 0.2316288 0.1659747 0.4094324 0.5073147 1.0000000
```

```
> covdata=cov(dataFA1)
> covdata
```

```
           BIO       GEO      CHEM       ALG      CALC      STAT
BIO  1.5068450 1.0300334 1.1669342 0.1662207 0.2952731 0.3134225
GEO  1.0300334 1.5128094 1.0666555 0.1954849 0.2835117 0.3586288
CHEM 1.1669342 1.0666555 1.6193868 0.1252508 0.1956633 0.2658751
ALG  0.1662207 0.1954849 0.1252508 1.3787625 1.0202341 0.6051839
CALC 0.2952731 0.2835117 0.1956633 1.0202341 1.2702230 0.7197436
STAT 0.3134225 0.3586288 0.2658751 0.6051839 0.7197436 1.5846042
```

```
> fa_1 <- factanal(dataFA1,2,rotation="varimax")
> fa_1
```

```
Call:
factanal(x = dataFA1, factors = 2, rotation = "varimax")

Uniquenesses:
  BIO   GEO  CHEM   ALG  CALC  STAT
0.252 0.375 0.249 0.374 0.048 0.715

Loadings:
     Factor1 Factor2
BIO  0.855   0.133
GEO  0.779   0.135
CHEM 0.865
ALG          0.791
CALC         0.971
STAT 0.170   0.506

               Factor1 Factor2
SS loadings      2.124   1.863
Proportion Var   0.354   0.311
Cumulative Var   0.354   0.665

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 2.94 on 4 degrees of freedom.
The p-value is 0.568
```

```
> sapply(1:2, function(f) factanal(dataFA1, factors = f, method ="mle")$PVAL)
```

```
   objective    objective
6.109624e-69 5.676271e-01
```

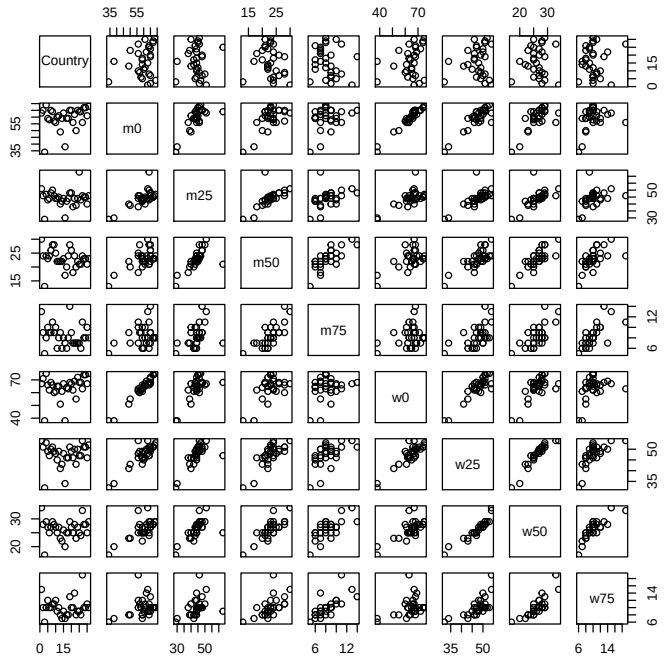
```
> #install.packages(c("psych","GPArotation"))
>
> library(psych)
> KMO (cordata)

Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = cordata)
Overall MSA =  0.7
MSA for each item =
 BIO  GEO CHEM  ALG CALC STAT
0.73 0.81 0.72 0.60 0.60 0.84

> cortest.bartlett (cordata, n=300)

$chisq
[1] 849.2133

$p.value
[1] 2.58314e-171

$df
[1] 15

> require(GPArotation)
> fares <- fa(dataFA1, nfactors=2,rotate="varimax", fm="ml")
> fares

Factor Analysis using method =  ml
Call: fa(r = dataFA1, nfactors = 2, rotate = "varimax", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
      ML2  ML1   h2    u2 com
BIO  0.85 0.13 0.75 0.252 1.0
GEO  0.78 0.13 0.63 0.375 1.1
CHEM 0.86 0.06 0.75 0.249 1.0
ALG  0.03 0.79 0.63 0.374 1.0
CALC 0.10 0.97 0.95 0.048 1.0
STAT 0.17 0.51 0.29 0.715 1.2

                       ML2  ML1
SS loadings           2.12 1.86
Proportion Var        0.35 0.31
Cumulative Var        0.35 0.66
Proportion Explained  0.53 0.47
Cumulative Proportion 0.53 1.00

Mean item complexity =  1.1
Test of the hypothesis that 2 factors are sufficient.
```

```
The degrees of freedom for the null model are  15  and the objective function was  2.87
The degrees of freedom for the model are 4  and the objective function was  0.01

The root mean square of the residuals (RMSR) is  0.01
The df corrected root mean square of the residuals is  0.02

The harmonic number of observations is  300 with the empirical chi square  0.97  with p
The total number of observations was  300  with MLE Chi Square =  2.94  with prob <  0.

Tucker Lewis Index of factoring reliability =  1.005
RMSEA index =  0  and the 90 % confidence intervals are  NA 0.076
BIC =  -19.87
Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                                 ML2  ML1
Correlation of scores with factors              0.94 0.98
Multiple R square of scores with factors        0.88 0.95
Minimum correlation of possible factor scores   0.76 0.91

>
```

7.2 example 2: life data

The life data contains the life expectancy in years by country, age, and sex.

```
> #### Life Data #
>
> #life <- read.table("C:/Users/Administrator/Documents/MUltivariate_Data_Analysis/life
>
> life <- read.table("life.dat",header=TRUE)
> head(life)

     Country m0 m25 m50 m75 w0 w25 w50 w75
1    Algeria 63  51  30  13 67  54  34  15
2   Cameroon 34  29  13   5 38  32  17   6
3 Madagascar 38  30  17   7 38  34  20   7
4  Mauritius 59  42  20   6 64  46  25   8
5    Reunion 56  38  18   7 62  46  25  10
6 Seychelles 62  44  24   7 69  50  28  14

> pairs(life)
>
```



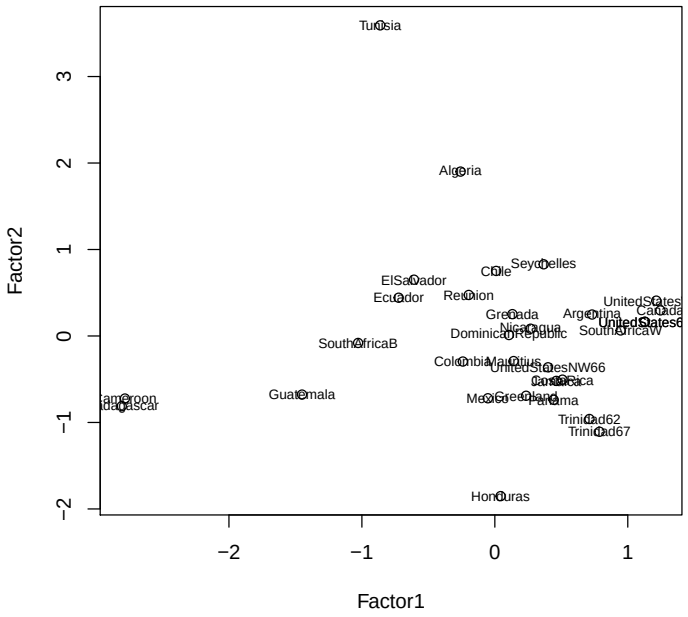

```
> sapply(1:3, function(f) factanal(life[,-1], factors = f, method ="mle")$PVAL)

   objective    objective    objective
1.879555e-24 1.911514e-05 4.578204e-01

> factanal(life[,-1], factors = 3, method ="mle")

Call:
factanal(x = life[, -1], factors = 3, method = "mle")

Uniquenesses:
   m0   m25   m50   m75    w0   w25   w50   w75
0.005 0.362 0.066 0.288 0.005 0.011 0.020 0.146

Loadings:
    Factor1 Factor2 Factor3
m0  0.964   0.122   0.226
m25 0.646   0.169   0.438
m50 0.430   0.354   0.790
m75         0.525   0.656
w0  0.970   0.217
w25 0.764   0.556   0.310
w50 0.536   0.729   0.401
```

```
w75 0.156   0.867   0.280

               Factor1 Factor2 Factor3
SS loadings      3.375   2.082   1.640
Proportion Var   0.422   0.260   0.205
Cumulative Var   0.422   0.682   0.887

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 6.73 on 7 degrees of freedom.
The p-value is 0.458
```

```
> scores <- factanal(life[,-1], factors = 3, method = "mle", scores = "regression")$sco
> head(scores)
```

```
        Factor1    Factor2     Factor3
[1,] -0.2580626  1.9009577  1.91581631
[2,] -2.7824958 -0.7234001 -1.84772224
[3,] -2.8064282 -0.8115882 -0.01210318
[4,]  0.1410049 -0.2902845 -0.85862443
[5,] -0.1963521  0.4742992 -1.55046466
[6,]  0.3673713  0.8290238 -0.55214085
```

```
> plot(scores[,1:2])
> text(scores[,1:2],labels=life[,1], cex=.7)
>
```


Tunisia
Algeria
Seychelles
Chile
ElSalvador
Ecuador
Reunion
Greenada
Argentina
Dominican Republic
SouthAfricaB
Colombia
UnitedStatesNW66
Guatemala
Mexico
Panama
Trinidad62
Trinidad67
Honduras
Factor2
Factor1
−2
−1
0
1
2
3

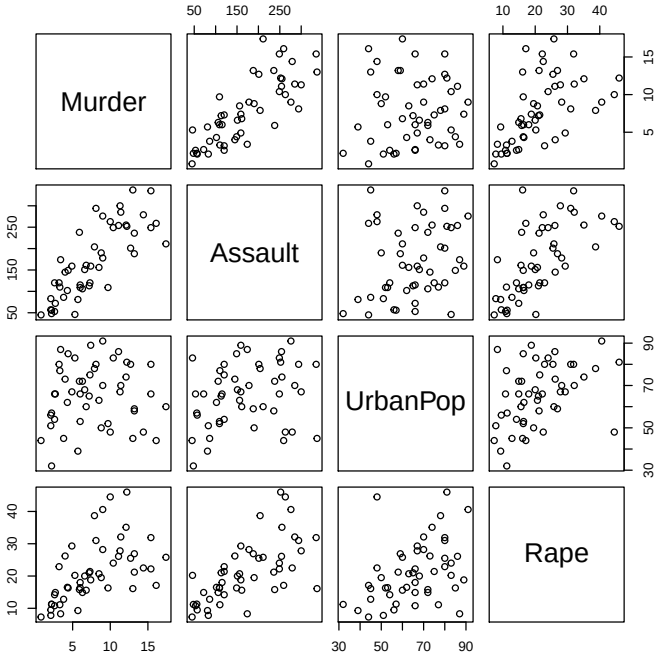
Chapter 8

Cluster Analysis

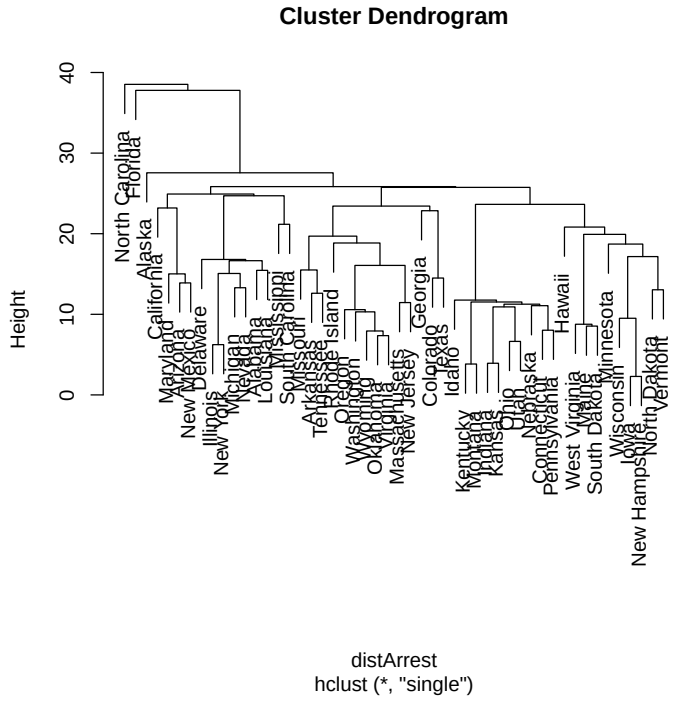
```
> library(corrplot)
> data(USArrests)
> head(USArrests)
```

```
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

```
> pairs(USArrests)
```

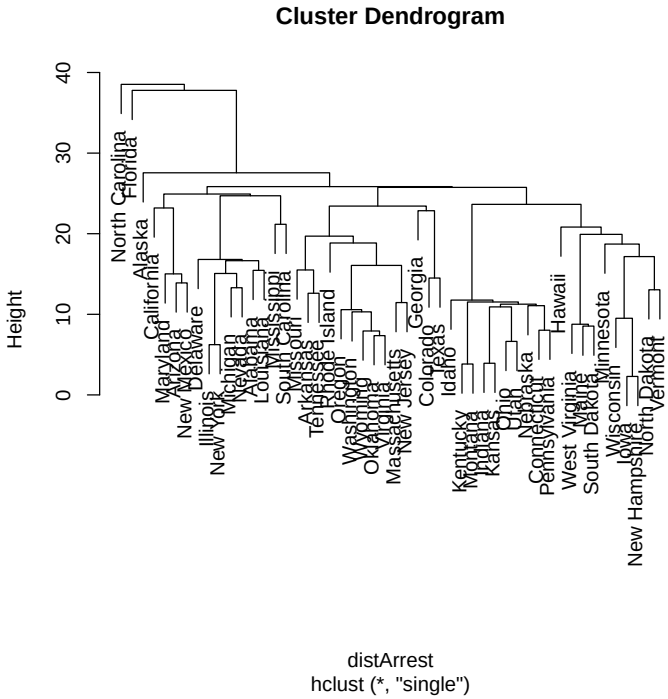



```
> corrplot(cor(USArrests), method = "ellipse", type = "upper")
```

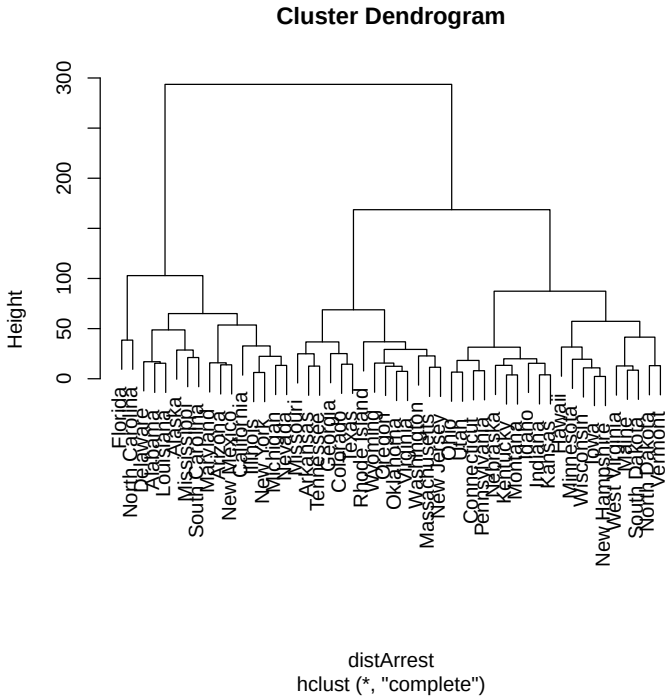


8.1 Hierachical CLustering

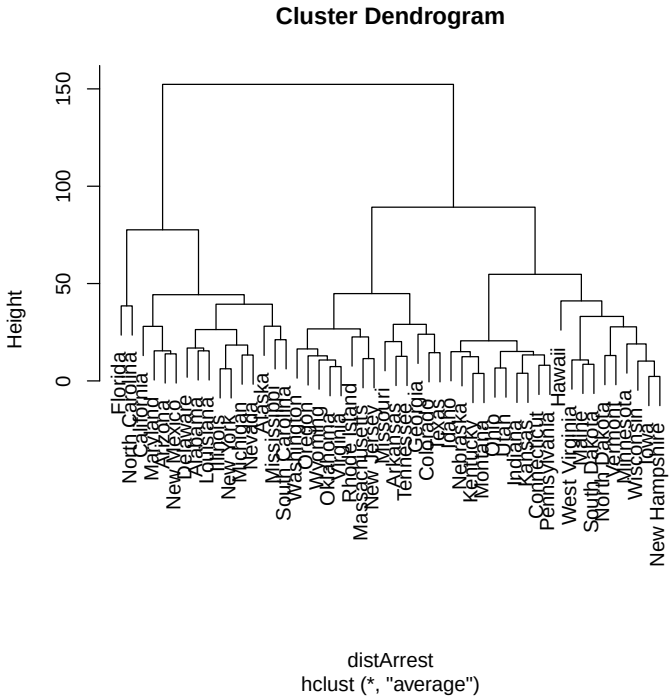
```
> ################################
> ### Hierachical CLustering  ##
> ################################
>
>
>
> distArrest <- dist (USArrests)
> ### Distance Methods Could be:  ##
> ### "euclidean", "maximum", "manhattan", "canberra" ##
> ### "binary" or "minkowski" ##
>
>
> # Clustering
>
> plot(res1 <- hclust(distArrest , method="single")  )
```



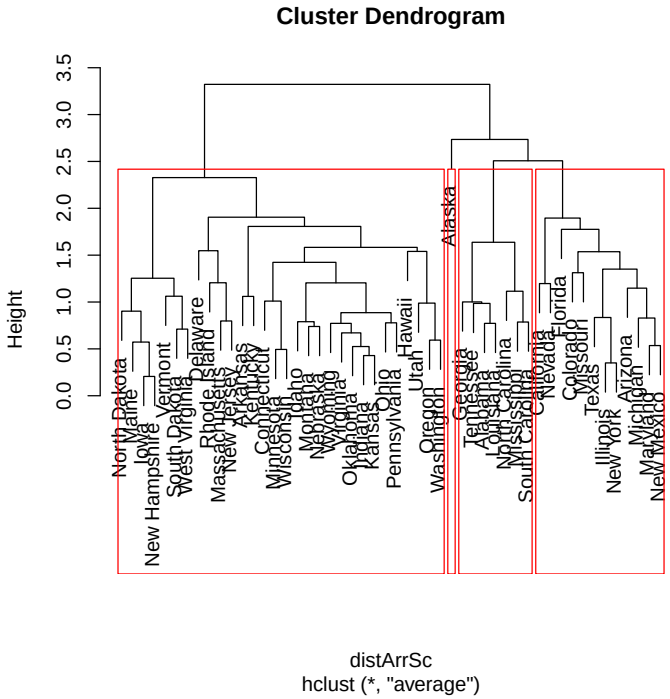

```
> ### max DIstance
> plot(res2 <- hclust(distArrest , method="complete")  )
>
```




```
> plot(res3 <- hclust(distArrest , method="average")  )
```




```
> ## Rescaling the data ###
>
> USArrScale <- scale (USArrests, scale=T)
> distArrSc <- dist (USArrScale )
> plot(res4 <- hclust(distArrSc , method="average")  )
> rect.hclust(res4, k = 4)
> head(cutree(res4, h = 2))

   Alabama     Alaska    Arizona   Arkansas California   Colorado
         1          2          3          4          3          3

> head( cutree(res4, k = 3))

   Alabama     Alaska    Arizona   Arkansas California   Colorado
         1          2          1          3          1          1

>
>
```



8.2 K-Means CLustering

```
> ###############
> ### K Means ###
> ###############
>
>
> kMres <- kmeans(USArrScale , centers=3)
> kMres $cluster
```

```
       Alabama         Alaska        Arizona       Arkansas     California
             2              2              2              1              2
      Colorado    Connecticut       Delaware        Florida        Georgia
             2              1              1              2              2
        Hawaii          Idaho       Illinois        Indiana           Iowa
             1              3              2              1              3
        Kansas       Kentucky      Louisiana          Maine       Maryland
             1              3              2              3              2
 Massachusetts       Michigan      Minnesota    Mississippi       Missouri
             1              2              3              2              2
       Montana       Nebraska         Nevada  New Hampshire     New Jersey
```

```
           3            3            2            3            1
  New Mexico     New York North Carolina North Dakota         Ohio
           2            2            2            3            1
    Oklahoma       Oregon Pennsylvania Rhode Island South Carolina
           1            1            1            1            2
South Dakota    Tennessee        Texas         Utah      Vermont
           3            2            2            1            3
    Virginia   Washington West Virginia    Wisconsin      Wyoming
           1            1            3            3            1
```

```
> #install.packages("fpc")
> library(fpc)
> plotcluster(USArrScale , kMres $cluster)
> pairs(USArrScale , col=c(1:3)[kMres $cluster])
> kMres1 <- kmeans(USArrScale , centers=3)
> head(kMres1 $cluster)
```

```
   Alabama     Alaska    Arizona   Arkansas California   Colorado
         2          2          2          1          2          2
```

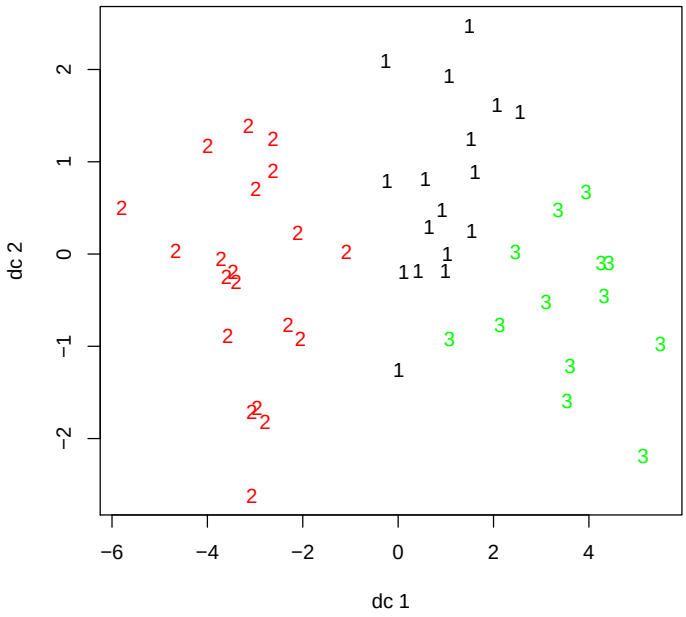
```
> kMres2 <- kmeans(USArrScale , centers=3)
> head(kMres2 $cluster)
```

```
   Alabama     Alaska    Arizona   Arkansas California   Colorado
         2          2          2          3          2          2
```

```
> table(kMres1 $cluster,kMres2 $cluster)
```

```
   1  2  3
1  0  0 17
2  0 20  0
3 13  0  0
```

```
>
>
>
>
```


dc 2
dc 1
−6
−4
−2
0
2
4
2
1
0
−1
−2

Chapter 9

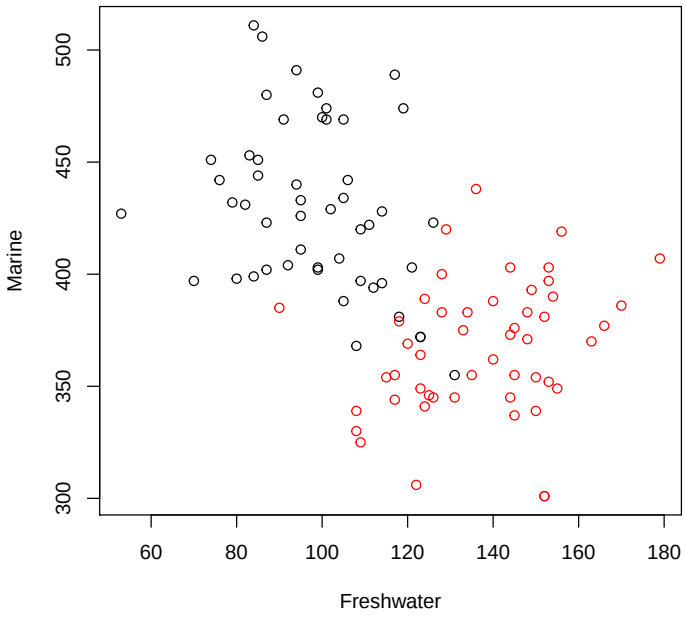
Classification

9.1 Linear Discriminant Analysis

```
> library(MASS)
> #install.packages("klaR")
> library(klaR)
> #salmon<- read.table("C:/Users/Administrator/Documents/MUltivariate_Data_Analysis/salmon.txt")
>
> salmon<- read.table("salmon.txt")
> head(salmon)
```

```
  SalmonOrigin Freshwater Marine
1       Alaska        108    368
2       Alaska        131    355
3       Alaska        105    469
4       Alaska         86    506
5       Alaska         99    402
6       Alaska         87    423
```

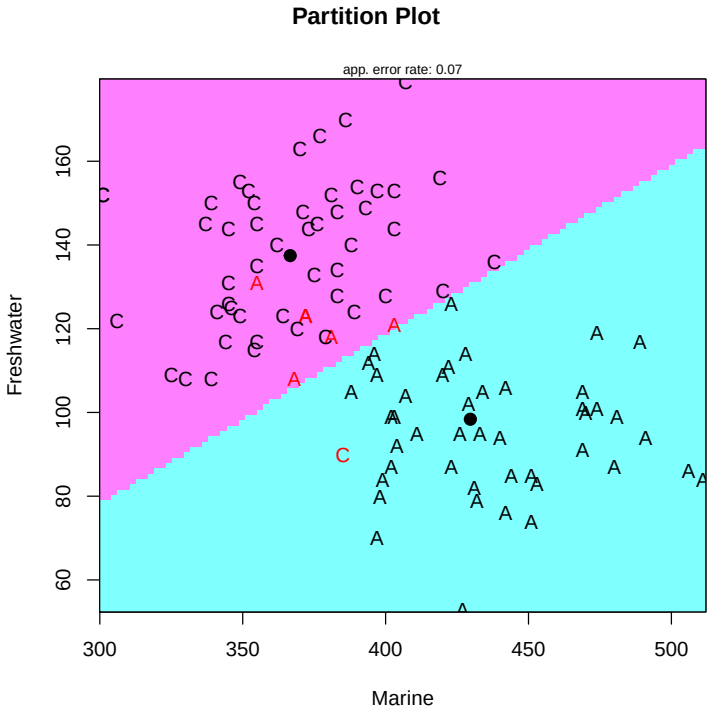
```
>  plot(salmon[,-1],col=as.factor(salmon[,1]))
```



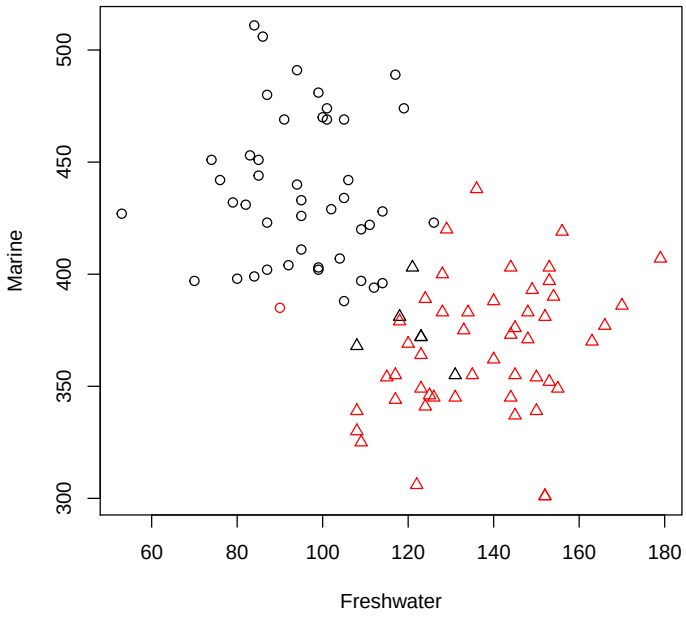

```
> partimat(SalmonOrigin ~ Freshwater + Marine,data=salmon,method="lda")
>  ## Split the data
> strain <- salmon[c(1:40,51:90),]
> stest <- salmon[c(41:50,91:100),]
> ldaRes  <- lda(strain[,c(2,3)],grouping=strain[,1])
> ldaRes

Call:
lda(strain[, c(2, 3)], grouping = strain[, 1])

Prior probabilities of groups:
Alaska Canada
   0.5    0.5

Group means:
       Freshwater  Marine
Alaska    100.550 422.275
Canada    138.625 368.650

Coefficients of linear discriminants:
                   LD1
Freshwater  0.04390178
```

```
Marine     -0.01806237

> plot(ldaRes)
> predlda <- predict(ldaRes  )
> ## How good is the classification ##
>
> ct <- table (prediction=predlda $class, real=strain[,1])
> ct

          real
prediction Alaska Canada
    Alaska     36      3
    Canada      4     37

> prop.table(ct, 2)

          real
prediction Alaska Canada
    Alaska  0.900  0.075
    Canada  0.100  0.925

> diag(prop.table(ct, 2))

Alaska Canada
 0.900  0.925

> # total percent correct overall
> sum(diag(prop.table(ct)))

[1] 0.9125

> predres <- cbind(predlda$class, predlda$x)
> head(predres)

            LD1
1 1 -0.01267408
2 2  1.23187773
3 1 -1.96867877
4 1 -3.47112033
5 1 -1.02191070
6 1 -1.92804185

> ## See how well to classify the Test data set
> table ( predict(ldaRes,stest[,c(2,3)])$class, stest[,1])

         Alaska Canada
  Alaska     10      0
  Canada      0     10
```

```
> ldaRescv=lda(salmon[,c(2,3)],grouping=salmon[,1],CV=TRUE)
> summary(ldaRescv)

          Length Class  Mode
class     100    factor numeric
posterior 200    -none- numeric
call        4    -none- call

> table ( ldaRescv$class, salmon[,1])

         Alaska Canada
  Alaska     44      1
  Canada      6     49
```




```
> plot(salmon[,c(2,3)],col=as.factor(salmon[,1]),pch=as.numeric(ldaRescv$class))
>
```



9.2 K-Nearest Neighbour

```
> ### K Nearest Neighbour ##
> train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
> test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
> cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
> library(class)
> #LOOCV ##
> KnnRes <- knn.cv(strain[,-1], strain[,1], k = 3, prob = TRUE)
> table(KnnRes, strain[,1])

KnnRes   Alaska Canada
  Alaska     35      7
  Canada      5     33

> KnnRes2 <- knn(strain[,-1] , stest[,-1], strain[,1], k = 3, prob=TRUE)
> table(KnnRes2, stest[,1])

KnnRes2  Alaska Canada
  Alaska     10      1
  Canada      0      9
```

```
>
```

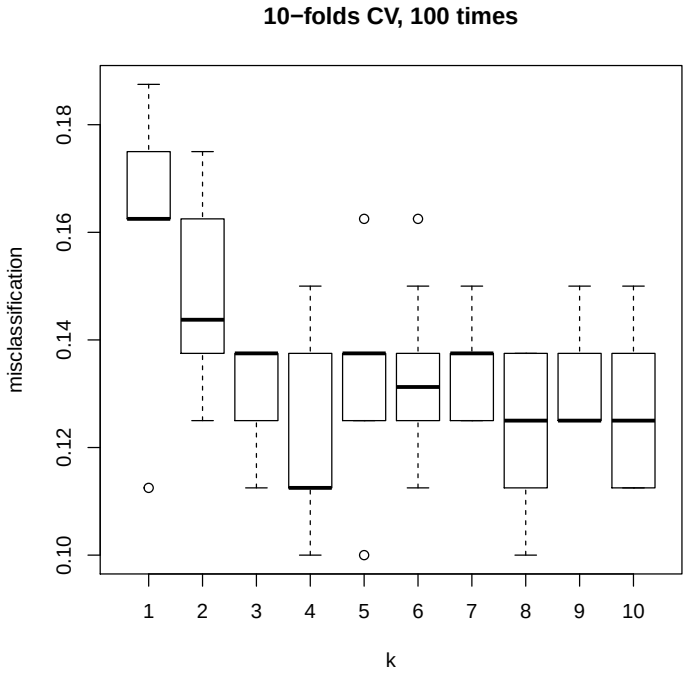
```
>     require(e1071)
> ### 10-folds CV is performed 10 times ###
>
> rep = 10
> kfoldcv <- matrix(NA,rep ,10)
> set.seed(123)
> for (i in 1:rep ) {
+ kfold  <- tune.knn(strain[,-1] ,  strain[,1], k = 1:10,
+           tunecontrol = tune.control(sampling = "cross",
+                                      cross=10))
+ kfoldcv[i,] <-summary(kfold )$performances[,2]
+ cat(i)
+ }
```

```
12345678910
```

```
> colnames(kfoldcv) <-  1:10
> boxplot(kfoldcv, ylab="misclassification", xlab="k", main="10-folds CV, 100 times")
> ## note: result show k =4 ##
>
>
> KnnRes3 <- knn(strain[,-1] , stest[,-1], strain[,1], k = 4, prob=TRUE)
> table(KnnRes3, stest[,1])
```

```
KnnRes3  Alaska Canada
  Alaska     10      0
  Canada      0     10
```

```
>
```



9.3 Logistics Regression

```
> fitLog <- glm( SalmonOrigin ~ Freshwater + Marine,
+                data=salmon, family="binomial")
> fitted <- fitted(fitLog )
> OC = function(grp, fit, np=100){
+   prob = seq(0.01,0.99, len=np)
+   pred = outer(fit, prob,'>')
+   spec = apply((grp==0 & !pred), 2, sum)/sum(grp==0)
+   sens = apply((grp==1 & pred),2,sum)/sum(grp==1)
+   a = approx(spec,sens, xout=seq(0.001,0.999,len=1000), rule=2)$y
+   auc = sum(a)/1000
+   return(list(sens=sens,spec=spec,auc=auc))
+ }
> grp <- ifelse( salmon$SalmonOrigin=="Alaska",0,1)
> roc <- OC(grp ,fitted  )
> plot(1-roc$spec, roc$sens, type="l")
> res <- cbind(prob=fitted,  Specificity = roc$spec, Sensitivity = roc$sens)
> head(res)
```

```
          prob Specificity Sensitivity
1 4.192345e-01        0.54        1.00
2 9.609966e-01        0.66        1.00
3 3.658558e-03        0.66        1.00
4 5.555317e-05        0.66        0.98
5 4.266105e-02        0.70        0.98
6 3.530058e-03        0.70        0.98
```

```
> fitted <- fitted(fitLog )
> ### Assuming 0.5 as the best threshold ##
>
> ctLog <- table(prediction=fitted> 0.5, realgroup= grp)
> ctLog
```

```
          realgroup
prediction  0  1
     FALSE 46  3
     TRUE   4 47
```

```
> prop.table(ctLog, 2)
```

```
          realgroup
prediction    0    1
     FALSE 0.92 0.06
     TRUE  0.08 0.94
```

```
> diag(prop.table(ctLog, 2))
```

```
[1] 0.92 0.94
```

```
> # total percent correct overall
>
> sum(diag(prop.table(ctLog)))
```

```
[1] 0.93
```

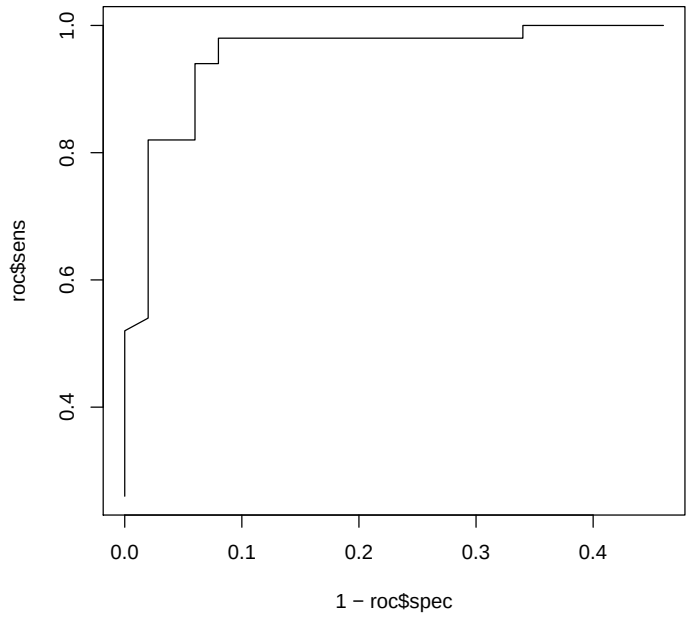
```
> ### If we use 0.7 as the  threshold ##
>
> ctLog <- table(fitted> 0.7, grp)
> prop.table(ctLog, 2)
```

```
       grp
          0    1
  FALSE 0.94 0.12
  TRUE  0.06 0.88
```

```
> diag(prop.table(ctLog, 1))
```

```
[1] 0.8867925 0.9361702
```

```
> # total percent correct
> sum(diag(prop.table(ctLog)))

[1] 0.91

>
>
> ## SO threshold 0.5 is better than 0.7 ###
>
>
```



Chapter 10

Canonical Regression

All data files can be downloaded from: "https://sites.google.com/site/biostatinfocore/introduction-to-r"

```
> mm <- read.csv("mmreg.csv")
> colnames(mm) <- c("Control", "Concept", "Motivation", "Read",
+                   "Write", "Math", "Science", "Sex")
> summary(mm)
```

```
    Control            Concept            Motivation          Read
 Min.   :-2.23000   Min.   :-2.620000   Min.   :0.0000   Min.   :28.3
 1st Qu.:-0.37250   1st Qu.:-0.300000   1st Qu.:0.3300   1st Qu.:44.2
 Median : 0.21000   Median : 0.030000   Median :0.6700   Median :52.1
 Mean   : 0.09653   Mean   : 0.004917   Mean   :0.6608   Mean   :51.9
 3rd Qu.: 0.51000   3rd Qu.: 0.440000   3rd Qu.:1.0000   3rd Qu.:60.1
 Max.   : 1.36000   Max.   : 1.190000   Max.   :1.0000   Max.   :76.0
     Write            Math          Science           Sex
 Min.   :25.50   Min.   :31.80   Min.   :26.00   Min.   :0.000
 1st Qu.:44.30   1st Qu.:44.50   1st Qu.:44.40   1st Qu.:0.000
 Median :54.10   Median :51.30   Median :52.60   Median :1.000
 Mean   :52.38   Mean   :51.85   Mean   :51.76   Mean   :0.545
 3rd Qu.:59.90   3rd Qu.:58.38   3rd Qu.:58.65   3rd Qu.:1.000
 Max.   :67.10   Max.   :75.50   Max.   :74.20   Max.   :1.000
```

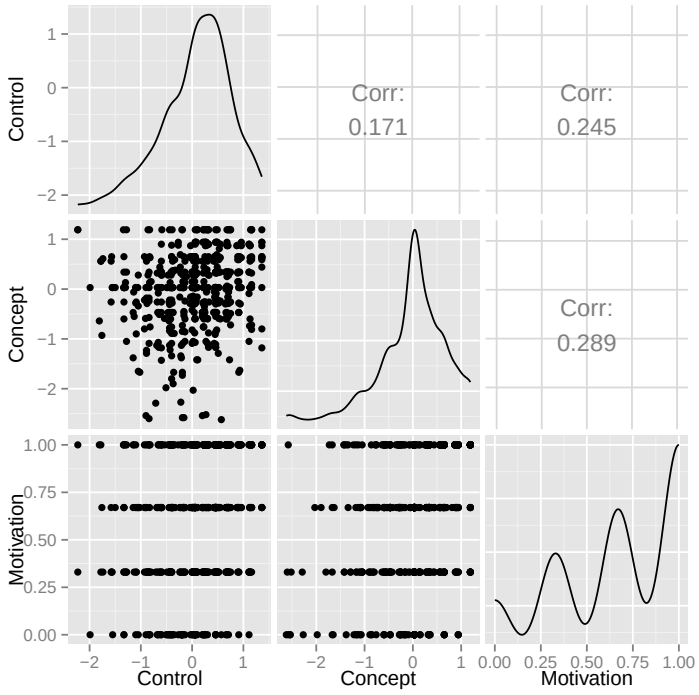
```
> #install.packages(c("ggplot2", "GGally", "CCA", "CCP"))
>
> require(ggplot2)
> require(GGally)
> require(CCA)
> require(CCP)
> psych <- mm[, 1:3]
> acad <- mm[, 4:7]
> head(psych)
```

```
  Control Concept Motivation
1   -0.84   -0.24       1.00
2   -0.38   -0.47       0.67
3    0.89    0.59       0.67
4    0.71    0.28       0.67
5   -0.64    0.03       1.00
6    1.11    0.90       0.33
```

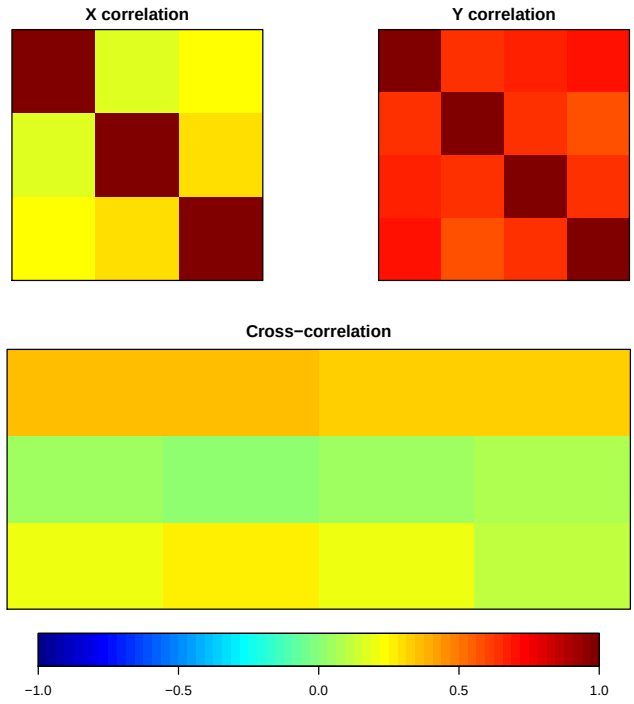
```
> head(acad)
```

```
  Read Write Math Science
1 54.8  64.5 44.5    52.6
2 62.7  43.7 44.7    52.6
3 60.6  56.7 70.5    58.0
4 62.7  56.7 54.7    58.0
5 41.6  46.3 38.4    36.3
6 62.7  64.5 61.4    58.0
```

```
> ggpairs(psych)
>
```



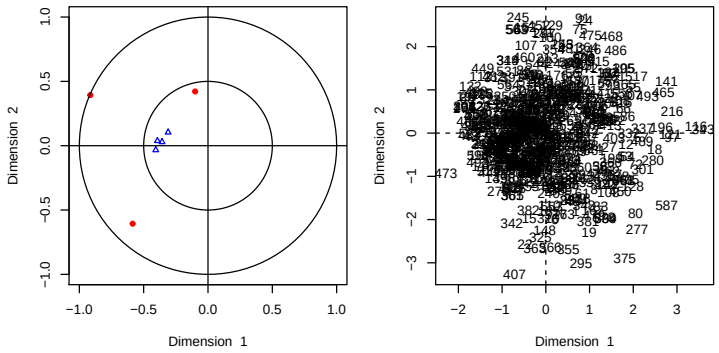

```
> corMat <- matcor(psych, acad)
> img.matcor(corMat , type = 2)
```




```
> ## use package CCA ##
> cc1 <- cc(psych, acad)
> cc1$cor
```

```
[1] 0.44643648 0.15335902 0.02250348
```

```
> plt.cc(cc1)
```




```
> ## CC from Base ##
> cc2 <- cancor(psych, acad)
> cc2$cor

[1] 0.44643648 0.15335902 0.02250348

> ### Testing CCA, using CCP package##
> N = nrow(psych)
> p = ncol(psych)
> q = ncol(acad)
> rho = cc2$cor
> ## Calculate p-values using the F-approximations of different test statistics:
> p.asym(rho, N, p, q, tstat = "Wilks")

Wilks' Lambda, using F-approximation (Rao's F):
              stat     approx df1      df2   p.value
1 to 3:  0.7814670 12.7735403  12 1569.222 0.0000000
2 to 3:  0.9759865  2.4210265   6 1188.000 0.0248771
3 to 3:  0.9994936  0.1507323   2  595.000 0.8601108

> p.asym(rho, N, p, q, tstat = "Hotelling")

 Hotelling-Lawley Trace, using F-approximation:
                stat     approx df1  df2    p.value
```

```
1 to 3:  0.2735079556 13.4854617  12 1775 0.00000000
2 to 3:  0.0245921193  2.4332536   6 1781 0.02400296
3 to 3:  0.0005066631  0.1509012   2 1787 0.85994364
```

```
> p.asym(rho, N, p, q, tstat = "Pillai")
```

```
 Pillai-Bartlett Trace, using F-approximation:
                stat     approx df1  df2    p.value
1 to 3:  0.2233309300 11.9641466  12 1785 0.00000000
2 to 3:  0.0240253971  2.4098260   6 1791 0.02530983
3 to 3:  0.0005064066  0.1516944   2 1797 0.85926186
```

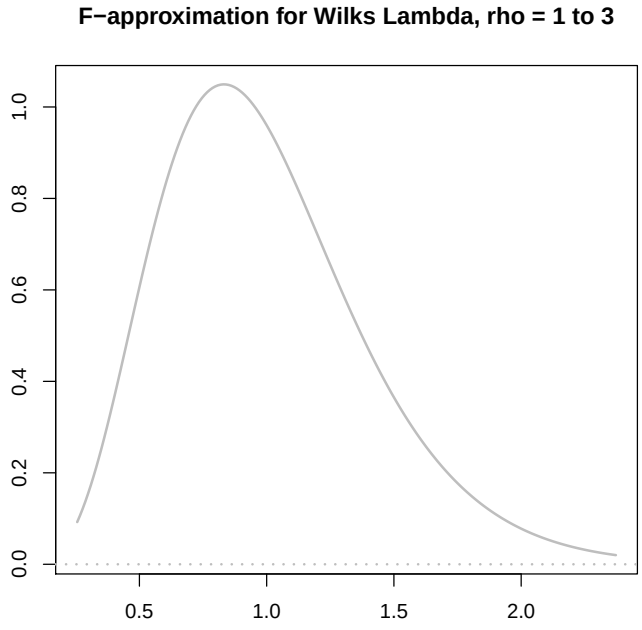
```
> p.asym(rho, N, p, q, tstat = "Roy")
```

```
 Roy's Largest Root, using F-approximation:
             stat   approx df1 df2 p.value
1 to 1:  0.1993055 37.02623   4 595       0

 F statistic for Roy's Greatest Root is an upper bound.
```

```
> ## Plot the F-approximation for Wilks' Lambda, considering 3, 2, or 1 canonical correlation(s):
> res1 <- p.asym(rho, N, p, q, tstat = "Wilks")
```

```
Wilks' Lambda, using F-approximation (Rao's F):
             stat     approx df1      df2   p.value
1 to 3:  0.7814670 12.7735403  12 1569.222 0.0000000
2 to 3:  0.9759865  2.4210265   6 1188.000 0.0248771
3 to 3:  0.9994936  0.1507323   2  595.000 0.8601108
```

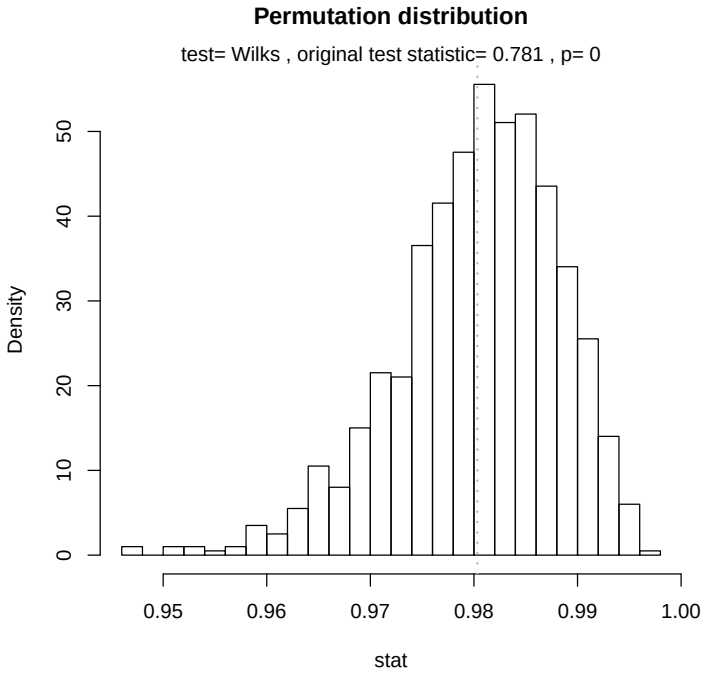
```
> plt.asym(res1,rhostart=1)
>
```




```
> resPerm <- p.perm(psych, acad, nboot = 999, rhostart = 1, type = "Wilks")
```

```
 Permutation resampling using Wilks 's statistic:
    stat0     mstat nboot nexcess p
 0.781467 0.9803274   999       0 0
```

```
> plt.perm(resPerm )
>
>
```


Permutation distribution
test= Wilks , original test statistic= 0.781 , p= 0
Density
0
10
20
30
40
50
0.95
0.96
0.97
0.98
0.99
1.00
stat