# Assignment

Foo <- 12

Foo

# Creating a numeric vector using concatenation operator c()

precip <- c(43, 20, 28, 28, 28, 15, 21, 10, 10, 19, 22, 58, 34, 21, 42, 50)

#Percip is called a Vector. How many elements does this vector have?

length(precip)

# Assign Names to these objects within the vector

COLONY <- c("SS", "SB", "WSB", "JRC", "JRH", "SJ", "CR", "UO", "LO", "DP", "PZ", "MC", "IF", "AF", "GH", "GL")

#Trick is to make sure both vectors are of same length

names(precip) <- COLONY

text(precip,label=names(precip))

# Indexing

precip[13]

# Creating a Matrix

RICHNESS <- c(2, 2, 3, 2, 2, 1, 5, 1)

RICHNESS

RICHMAT <- matrix(RICHNESS, nrow = 4)

RICHMAT

# To be more specific, the number of columns can be declared

RICHMAT <- matrix(RICHNESS, nrow = 4, ncol=2)

RICHMAT

# Indexing in Matrix

RICHMAT[3, 2]

RICHMAT[2,]

RICHMAT[,1]

# Alternative way to create a matrix: defining separate vectors of the same length and binding them by column with cbind

richy1 <- c(2, 2, 3, 2)

richy2 <- c(2, 1, 5, 1)

RICHMAT1 <- cbind(richy1, richy2)

RICHMAT1

RICHMAT1[,1]

RICHMAT1[,"richy1"]

# Asking for names of columns and rows in a matrix

colnames(RICHMAT1)

rownames(RICHMAT1)

# Defining names of rows

rownames(RICHMAT1) <- c("S1","S2","S3","S4")

rownames(RICHMAT1)

RICHMAT1

# Data Frame

# Data frame creation (importing data from a tab delimited file)

Euroemp <- read.delim("~/Desktop/MVA/R Exercise/K25945 - R code and data sets/Chapter 1/Euroemp.txt", row.names=1)

euro.emp <- Euroemp

class(euro.emp)

# Displaying the contents the data frame containing the imported data

euro.emp

# Displaying the first six rows of the data frame euro.emp

head(euro.emp)

# Showing the structure of the data frame euro.emp

str(euro.emp)

# Displaying the contents of the numeric vector AGR, one of the variables

# in the data frame euro.emp

euro.emp$AGR

# The following line would produce an error message (the name of the data frame was not specified)

AGR

# The class of the variable Group within euro.emp is "factor"

class(euro.emp$Group)

# The mode of the (factor) variable Group within euro.emp is "numeric" mode(euro.emp$Group)

# Importing data from a csv (comma separated values) file

# The file has a header (the first row has variable names) and

# the first column carries the names of the sampling units

Euroemp <- read.csv("~/Desktop/MVA/R Exercise/K25945 - R code and data sets/Chapter 1/Euroemp.csv", row.names=1)

euro.empcsv <- Euroemp

euro.empcsv

# Indexing elements in a data frame

euro.emp$AGR[3:8]

euro.emp$AGR[c(1, 3, 5)]

euro.emp[27:30,]

euro.emp[c("UK","Romania"),]

euro.emp[,c(2,5)]

euro.emp[,c("Group","MIN")]

euro.emp[euro.emp$Group == "Other",]

# Create a Matrix

# Create a 2x3 Matrix called A columnwise

A <- matrix(c(2,-1,3,0,1,4))

A

A <- matrix(c(2,-1,3,0,1,4),nrow = 2)

A

A <- matrix(c(2,-1,3,0,1,4),nrow = 2, byrow = TRUE)

A

#Dimension of matrix by dim() is a 2 demensional vector displaying number of rows and columns

dim(A)

nrow(A)

ncol(A)

NROW(A)

NCOL(A)

# what happens when we supply incorrect number of elements?

A1 <- matrix(c( 2, 3, 1,-1, 4), nrow=2, byrow=TRUE)

A1

A1 <- matrix(c( 2, 3, 1,-1, 4), nrow=3, byrow=TRUE)

A1

#Vector in R is different from Matrix. Vector has length but no dimension

r <- c(3,4,6,8)

length(r)

dim(r)

#Convert a vector to a matrix

c.vec <- matrix(c(3,4,6,8))

#Will it be a row or column vector?

c.vec

A <- matrix(c( 2, 3, 1,-1, 0, 4), nrow=2, byrow=TRUE)

# Transpose: function t()

A

dim(A)

A.t <- t(A)

dim(A.t)

A.t

r.vec <- t(c.vec)

r.vec

dim(r.vec)

dim(c.vec)

# Creating a 3x2 Zero Matrix

matrix(0,3,2)

#creating a diagonal matrix

diag(c(1,4,0,5))

diag(c(sin(pi/2),cos(0))) == diag(2)

diag(2)

diag(c(sin(pi/2),cos(0)))

#Are two matrix equal?

diag(c(sin(pi/2),cos(0))) == diag(2)

m\_equal <- diag(c(sin(pi/2),cos(0))) == diag(2)

class(m\_equal)

m\_equal

# Trace of a matrix

C <- matrix(c(2,-1,-0,-1,4,6,0,6,1),nrow=3,byrow=TRUE)

C

trace.C <-sum(diag(C))

trace.C

library(psych)

tr(C)

# Addition and subtraction of matrices; multiplication of a matrix by a scalar

A <- matrix(c( 3, 1, 4,-1, 0, 2), 2, byrow=TRUE)

B <- matrix(c(1,1,-2,0,3,-2),2,byrow=TRUE)

A

B

dim(A)

dim(B)

A+B

A-B

20\*A

#Matrix Multiplication

A <- matrix(c(3,1,4,-1,0,2),2,byrow=TRUE)

B <- matrix(c(1,-1,2,0,3,1),3,byrow = TRUE)

A%\*%B

B%\*%A

# \* is element wise multiplication. Only works on equal dimension matrix

A\*B

C <- matrix(c(1,-1,2,1),2,byrow=TRUE)

# Will we get a error?

A%\*%C

#Matrix Inversion by using Solve

M <- matrix(c(2,1,1,2),2,byrow=TRUE)

solve(M)

#Since M has an inversion, determinant of M should not be 0

det(M)

#Singular Matrix

N <- matrix(c(2,1,4,2),2,byrow = TRUE)

N

det(N)

solve(N)

# Orthonogal Matrix

P <- matrix(c(0,-1,0,1,0,0,0,0,-1),3,byrow=TRUE)

P

P.t <- t(P)

P.t

P.tinv <- solve(P)

P.tinv

P.tinv == P.t

# Quadratic forms

x <- c(1,2)

attributes(x)

class(x)

type(x)

A <- matrix(c(1,3,3,-1),2,byrow = TRUE)

A

Q <- t(x)%\*%A%\*%x

Q

class(Q)

t(x)

#x is a vector but was coerced into a matrix during transpose

class(t(x))

# Vector of Means, Coveraiance and correlation Matrix

Bumpus\_sparrows <- read.delim("~/Desktop/MVA/R Exercise/K25945 - R code and data sets/Chapter 2/Bumpus\_sparrows.txt")

sparr <- Bumpus\_sparrows

sparr

sparr[,-1]

#We will drop column 1 as it does not contain a number

sparr.num <- sparr[,-1]

# Computing the means of each variable in data frame sparr.num

colMeans(sparr.num)

# Covariance matrix

cov(sparr.num)

# Correlation matrix

cor(sparr.num)

# Lets do some MVA stuff

install.packages("MVA")

library(MVA)

install.packages("HSAUR2")

library(HSAUR2)

# Check out the datasets available in these packages

data(package='MVA')

data(package='HSAUR2')

# Lets create a dataset called Measure

measure <-

structure(list(V1 = 1:20, V2 = c(34L, 37L, 38L, 36L, 38L, 43L,

40L, 38L, 40L, 41L, 36L, 36L, 34L, 33L, 36L, 37L, 34L, 36L, 38L,

35L), V3 = c(30L, 32L, 30L, 33L, 29L, 32L, 33L, 30L, 30L, 32L,

24L, 25L, 24L, 22L, 26L, 26L, 25L, 26L, 28L, 23L), V4 = c(32L,

37L, 36L, 39L, 33L, 38L, 42L, 40L, 37L, 39L, 35L, 37L, 37L, 34L,

38L, 37L, 38L, 37L, 40L, 35L)), .Names = c("V1", "V2", "V3",

"V4"), class = "data.frame", row.names = c(NA, -20L))

measure <- measure[,-1]

names(measure) <- c("chest", "waist", "hips")

measure$gender <- gl(2, 10)

levels(measure$gender) <- c("male", "female")

measure

#Variance

var(measure)

# Since it forces NA, lets take the last column out

cov(measure[, c("chest", "waist", "hips")])

#Take a subset by gender

cov(subset(measure, gender == "female")[,c("chest", "waist", "hips")])

cov(subset(measure, gender == "male")[,c("chest", "waist", "hips")])

# Take the correlation

cor(measure[, c("chest", "waist", "hips")])

# Lets understand Distance. use Scale function when units are different

dist(scale(measure[, c("chest", "waist", "hips")],center = FALSE))

x <- measure[, c("chest", "waist", "hips")]

cm <- colMeans(x)

S <- cov(x)

d <- apply(x, MARGIN = 1, function(x)t(x - cm) %\*% solve(S) %\*% (x - cm))

# Normal Plot. Are they in a straight line.

qqnorm(measure[,"chest"], main = "chest"); qqline(measure[,"chest"])

qqnorm(measure[,"waist"], main = "waist"); qqline(measure[,"waist"])

qqnorm(measure[,"hips"], main = "hips"); qqline(measure[,"hips"])

# Density Plot. What are the outliers

plot(qc <- qchisq((1:nrow(x) - 1/2) / nrow(x), df = 3), sd <- sort(d),xlab = expression(paste(chi[3]^2, " Quantile")),ylab = "Ordered distances")

oups <- which(rank(abs(qc - sd), ties = "random") > nrow(x) - 3)

text(qc[oups], sd[oups] - 1.5,oups)

abline(a = 0, b = 1)