**Basic Operations in R Assignment**

**1.** Execute the following lines which create two vectors of random integers which are chosen with

replacement from the integers 0, 1, : : : , 999. Both vectors have length 250.

set.seed(100)

x <- sample (0:999, 250, replace=T)

y <- sample (0:999, 250, replace=T)

1. Identify out the values in y which are > 500.

**y[y>500]**

1. Identify the index positions in y of the values which are > 700?

**which(y>700)**

(c) What are the values in x which are in same index position to the values in y which are >

400?

**x[y>400]**

(d) **How many values in y are within 200** of the maximum value of the terms in y?

**count<-length(which(y <= 200))**

(e) How many numbers in x are divisible by 2?

**sum(x%%2==0)**

(f) Sort the numbers in the vector x in the order of increasing values in y.

**x[order(y)]**

**order**() returns the indices of the vector **in a sorted order**

**2.** Use the function paste to create the following character vectors of length 30:

(a) ("Label 1", "Label 2", ....., "Label 30").

\*Note that there is a single space between label and the number following.

**paste ("Label",1:30)**

(b) ("FN1", "FN2", ..., "FN30").

\*\*In this case, there is no space between fn and the number following.

**paste ("FN",1:30,sep="")**

**3.** Compound interest can be computed using the formula

A = P × (1 + R/100)n, where P is the original money lent, A is what it amounts to in n years at R

percent per year interest.

Write R code to calculate the amount of money owed after n years, where n changes from 1 to 15 in yearly increments, if the money lent originally is 10000 Rupees and the interest rate remains constant throughout the period at 11.5%.

**P<-10000**

**R<-11.5**

**n<-1**

**for(i in 1:15)**

**{**

**A<-P\*(1+R/100)\*n**

**P<-A**

**print(A)**

**}**

**4.** Generate the following matrices.

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

[1,] 1 101 201 301

[2,] 2 102 202 302

[3,] 3 103 203 303

[4,] 4 104 204 304

[5,] 5 105 205 305

**m<-matrix(c(1:5,101:105,201:205,301:305),nrow=5, ncol=4)**

**5.** Create a 6 by 10 matrix of random integers chosen from 1 to 10 by executing the following two lines of code:

set.seed(100)

GMAT <- matrix( sample(10, size=60, replace=T), nr=6)]

(a) Find the number of entries in each row which are greater than 4.

**apply(GMAT,1,function(x){sum(x>4)})**

(b) Which rows contain exactly two occurrences of the number seven?

**which(apply(GMAT,1,function(x){sum(x==7)==2}))**

(c) Find those pairs of columns whose total (over both columns) is >= 50. The answer should be a matrix with two columns.

**GMatColSums <- colSums(GMAT)**

**which( outer(GMatColSums,GMatColSums,"+")>50, arr.ind=T )**

colSums() function in R Language is used to compute the sums of matrix or **array** columns.

The **which() function in R** returns the position or the index of the value which satisfies the given condition. The which() function in R gives you the position of the value in a logical vector.

The R outer function **applies a function to two arrays**.

outer(x, y, "\*")

Outer Product of Two Vectors

x2 <- 1:5

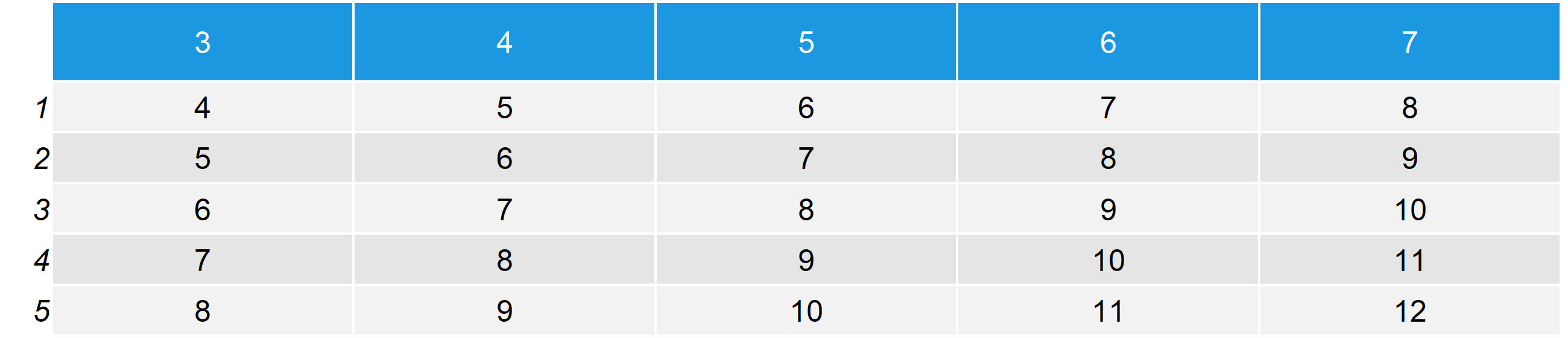
y2 <- 3:7

names(x2) <- x2 # Prepare row names of output matrix

names(y2) <- y2 # Prepare column names of output matrix

output <- outer(x2, y2, "+")

the outer command returns a matrix in which all combinations of the two vectors are stored.



sales<-c(100,50,75,150,200,25)

rank<-rank(sales)

sorted<-sort(sales)

ordered<-order(sales)

view <-Data.frame(Sales,Ranks,Sorted,Ordered)