R- Basic Exercise

1.

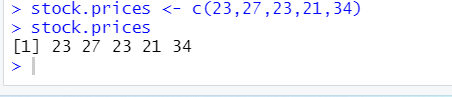
2^5



2.

stock.prices <- c(23,27,23,21,34)

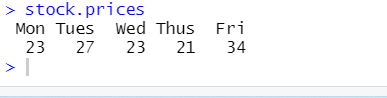
stock.prices



3.

names(stock.prices) <- c('Mon','Tues','Wed','Thus','Fri')

stock.prices



4.

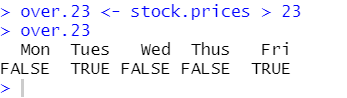
mean(stock.prices)



5.

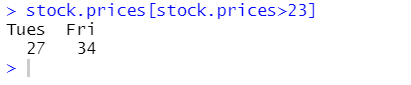
over.23 <- stock.prices > 23

over.23



6.

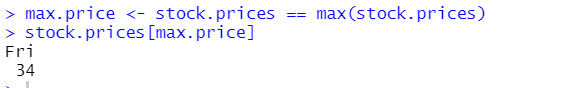
stock.prices[stock.prices>23]



7.

max.price <- stock.prices == max(stock.prices)

stock.prices[max.price]



R Matrix Exercises

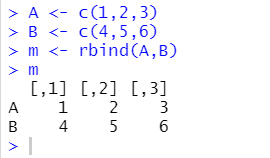
1.

A <- c(1,2,3)

B <- c(4,5,6)

m <- rbind(A,B)

m



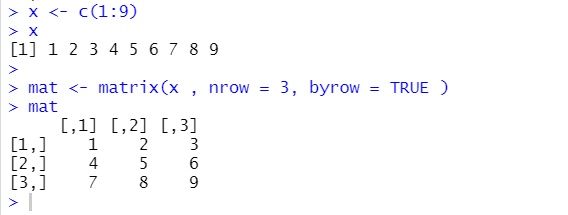
2.

x <- c(1:9)

x

mat <- matrix(x , nrow = 3, byrow = TRUE )

mat



3.

is.matrix(mat)

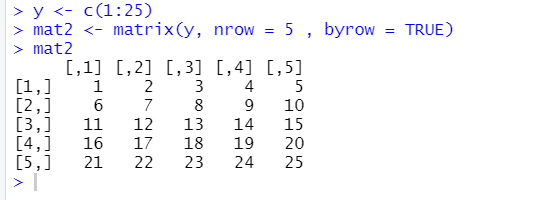


4.

y <- c(1:25)

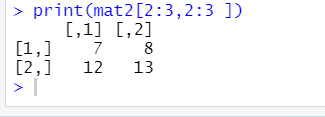
mat2 <- matrix(y, nrow = 5 , byrow = TRUE)

mat2



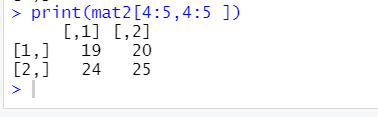
5.

print(mat2[2:3,2:3 ])



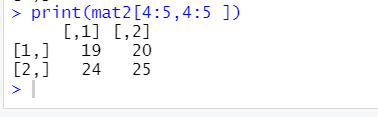
6.

print(mat2[4:5,4:5 ])



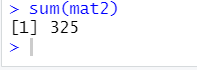
7.

print(mat2[4:5,4:5 ])



8.

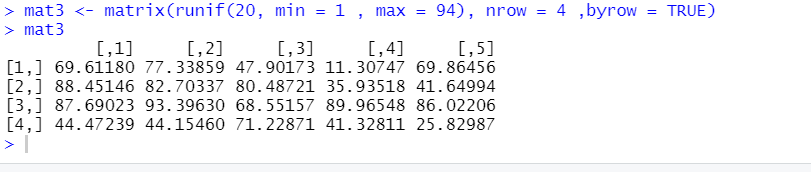
sum(mat2)



9.

mat3 <- matrix(runif(20, min = 1 , max = 94), nrow = 4 ,byrow = TRUE)

mat3



Conditional Statements Exercises

1.

x <- 1

if (x == 1){

print('Hello')

}



2.

x <- 3

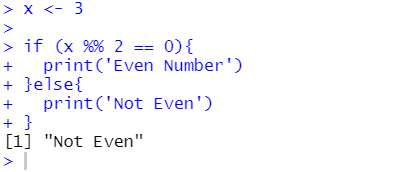
if (x %% 2 == 0){

print('Even Number')

}else{

print('Not Even')

}



3.

x <- matrix()

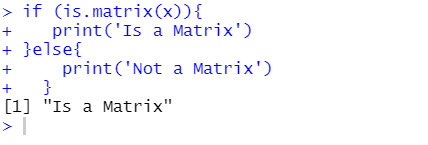
if (is.matrix(x)){

print('Is a Matrix')

}else{

print('Not a Matrix')

}



4.

x <- c(3,7,1)

if (x[1] > x[2]){

if(x[1] > x[3]){

print(x[1])

if(x[2] > x[3]){

print(x[2])

print(x[3])

}else{

print(x[3])

print(x[2])

}

}

}else if (x[2] > x[1]){

if(x[2] > x[3]){

print(x[2])

if(x[1] > x[3]){

print(x[1])

print(x[3])

}else{

print(x[3])

print(x[1])

}

}

}else if (x[3] > x[1]){

if(x[3] > x[2]){

print(x[3])

if(x[1] > x[2]){

print(x[1])

print(x[2])

}else{

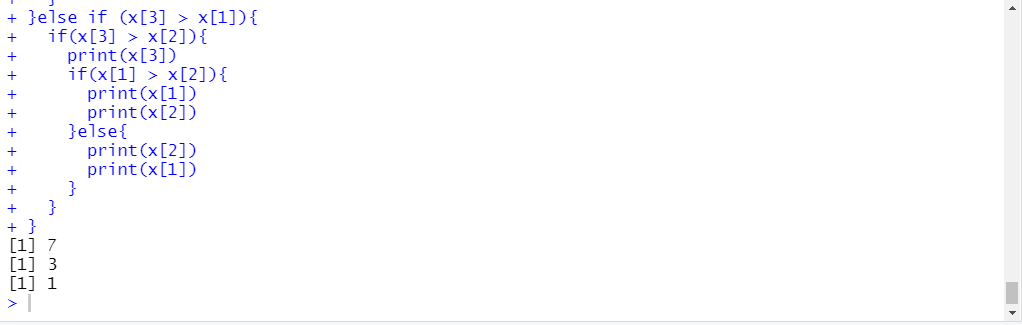
print(x[2])

print(x[1])

}

}

}



5.

x <- c(20, 10, 1)

if (x[1] > x[2] && x[1] > x[3]){

print(x[1])

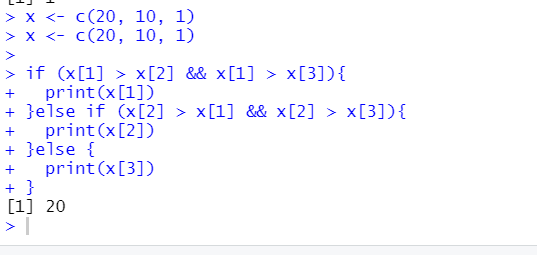
}else if (x[2] > x[1] && x[2] > x[3]){

print(x[2])

}else {

print(x[3])

}



R Functions Exercises

1.

hello\_you <- function(name) { # create a function with the name my\_function

print(paste("Hello" , name))

}

hello\_you('Sam')



2.

hello\_you2 <- function(name) { # create a function with the name my\_function

return(paste("Hello" , name))

}

hello\_you2('Sam')



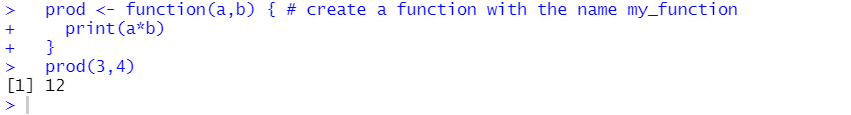
3.

prod <- function(a,b) { # create a function with the name my\_function

print(a\*b)

}

prod(3,4)



4.

num\_check <- function(a,b) { # create a function with the name my\_function

if (a %in% b){

print(TRUE)

}else {FALSE}

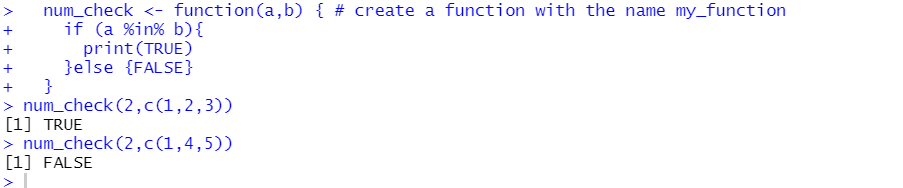
}

Input: num\_check(2,c(1,2,3))

Output: TRUE

Input: num\_check(2,c(1,4,5))

Output: FALSE



5.

num\_count <- function(a,b) { # create a function with the name my\_function

count2 <- 0

for ( p in (1:length(b))) {

if (a == b[p]){

count2 <- count2 + 1

}

}

return(count2)

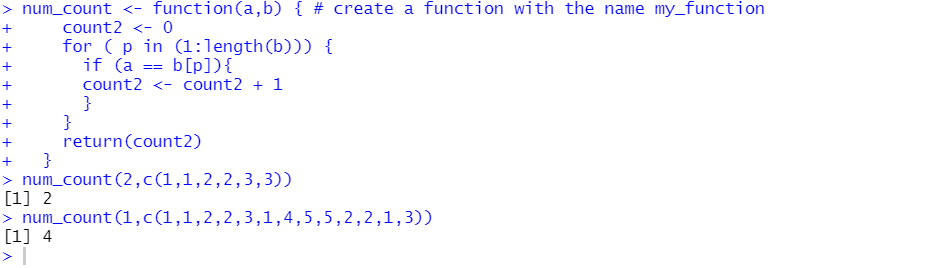
}

Input: num\_count(2,c(1,1,2,2,3,3))

Output:2

Input: num\_count(1,c(1,1,2,2,3,1,4,5,5,2,2,1,3))

Output:4



6.

bar\_count <- function(a) { # create a function with the name my\_function

t <- a %% b2

p <- a %/% b2

count3 <- t + p

return(count3)

}

Input: bar\_count(6)

Output: 2

Input: bar\_count(17)

Output:5



7.

summer <- function(a,b,c){

count1 <- 0

if (a %% 3 != 0){

count1 <- count1 + a

}

if (b %% 3 != 0){

count1 <- count1 + b

}

if (c %% 3 != 0){

count1 <- count1 + c

}

return(count1)

}

Input: summer(7,2,3)

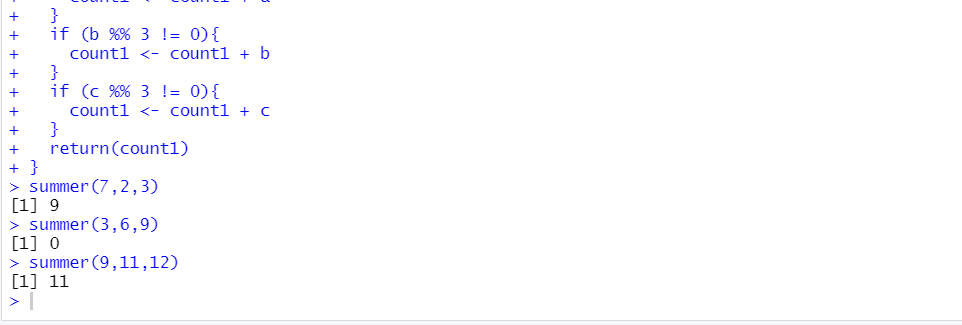
Output:9

Input: summer(3,6,9)

Output:0

Input: summer(9,11,12)

Output:11



8.

prime\_check <- function(a){

c1 <- vector()

for (x in (1:1000)){

if (a %% x == 0){

c1 <- c(c1,TRUE)

}

}

if (length(c1) > 2){

return(FALSE)

}else{

return(TRUE)

}

}

Input: prime\_check(2)

Output:TRUE

Input: prime\_check(5)

Output: TRUE

Input: prime\_check(4)

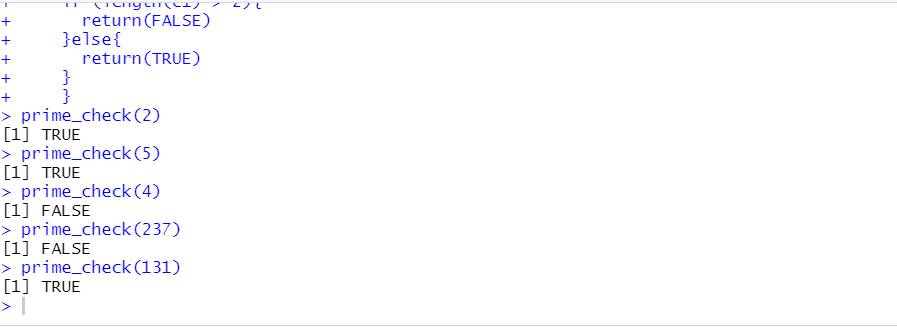
Output: FALSE

Input: prime\_check(237)

Output: FALSE

Input: prime\_check(131)

Output: TRUE



R Data frames Exercises

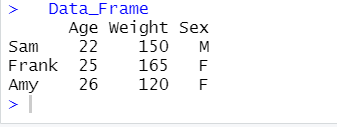
1.

Data\_Frame <- data.frame(

Age , Weight , Sex , row.names = paste0(c('Sam','Frank','Amy'))

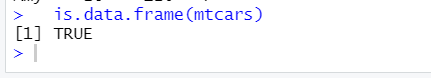
)

Data\_Frame



2.

is.data.frame(mtcars)

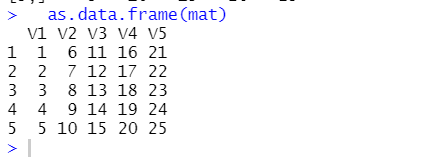


3.

mat <- matrix(1:25 , nrow = 5 )

mat

as.data.frame(mat)



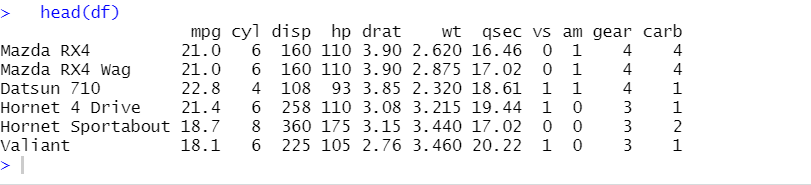
4.

df <- mtcars

df

5.

head(df)



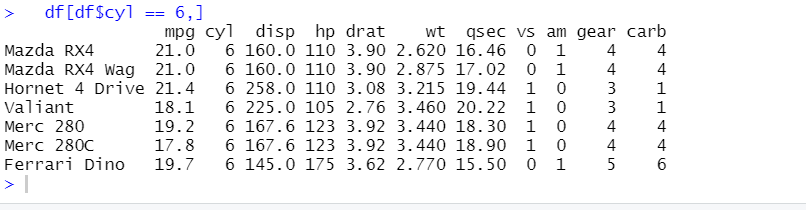
6.

mean(df$mpg)



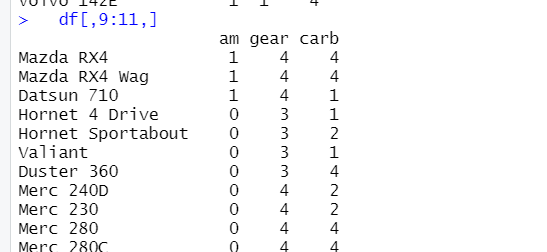
7.

df[df$cyl == 6,]



8.

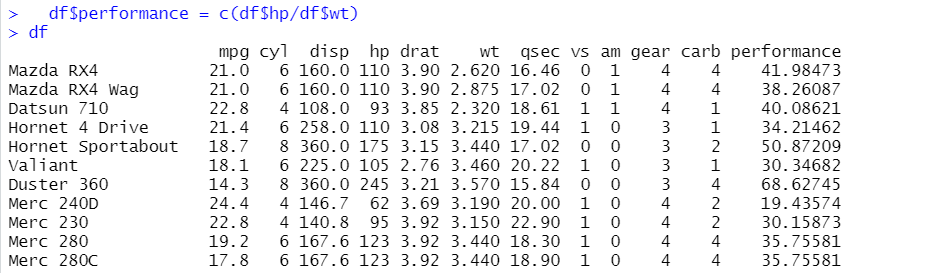
df[,9:11,]



9.

df$performance = c(df$hp/df$wt)

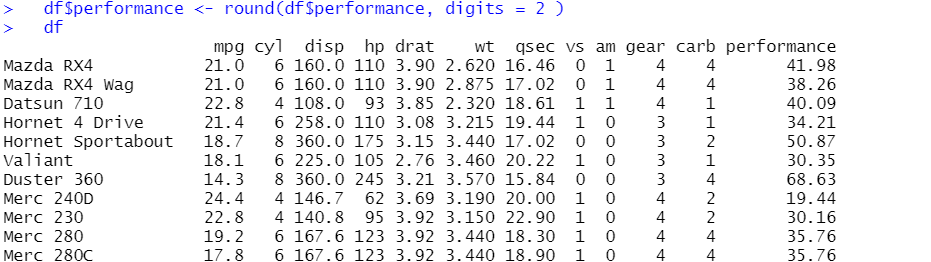
df



10.

df$performance <- round(df$performance, digits = 2 )

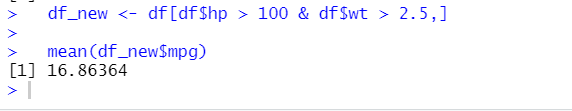
df



11.

df\_new <- df[df$hp > 100 & df$wt > 2.5,]

mean(df\_new$mpg)



12

df[5,1,]

