# Printout

Sunday, December 6, 2020 8:42 AM

```
#---------------------------------------------------------------------
#Chapter 3.4 ---------------------------------------------------------
#Exercise 2 ----------------------------------------------------------
#---------------------------------------------------------------------
#Script Header - This section provides basic information about the student.
#Created by Mowalola Onigbanjo on 12/03/2020
#Program: Masters Business Analytics
#Email: monigbanjo2020@student.hult.edu
#---------------------------------------------------------------------



#---------------------------------------------------------------------
#Chapter 3.4 ---------------------------------------------------------
#Exercise 3 ----------------------------------------------------------
#Importing a library to build a neural network (type: perceptron)
install.packages("neuralnet")
install.packages("keras")


#Load package and get help function
library(neuralnet)
library(keras)
library(tensorflow)

#Getting help on the functions
?tensorflow
?train



#Chapter 3.4 ---------------------------------------------------------
#Exercise 8 ----------------------------------------------------------
#Creating a transformmatrix function ---------------------------------

transformmatrix <- function(x){ #Beginning of user defined function -
transformmatrix

  diagonal_x <- diag(x) #This displays the diagonal of the matrix

  diag_mean <- mean(diagonal_x, na.rm =TRUE) #This calculates the diagonal
mean of the matrix
  diag_median <- median(diagonal_x, na.rm =TRUE) #This calculates the diagonal
median of the matrix
  return (c(diag_mean, diag_median)) #This returns the values of the diagonal
mean and median of the matrix

  } #Closing the transformmatrix function

#Exercise 3.4 ---------------------------------------------------------
#Exercise 8a ----------------------------------------------------------
#Matrix from Chapter 2 Exercise 7 ------------------------------------
```

```
#Creating vectors with variables that can be transformed into a matrix
m1 <- c(10, 11, 9, 15, 19)
m2 <- c(52, 19, 7, 10, 22)
m3 <- c(28, 40, 6, 99, 33)
m4 <- c(35, 26, 5, 87, 91)
m5 <- c(0, 12, 16, 81, 200)

#Calling the transformmatrix function

transformmatrix (x = matrix(x <- c(m1,m2,m3,m4,m5), nrow = 5, ncol=5, byrow =
T))



#Exercise 3.4 ------------------------------------------------------------
#Question 8b  ------------------------------------------------------------
#Matrix from Example in Chapter 2 ----------------------------------

#Creating vectors with variables that can be transformed into a matrix

p1 <- c(1, 2, 3)
p2 <- c(4, 5, 6)
p3 <- c(7, 8, 9)

#Calling the transformmatrix function

transformmatrix (x = matrix(x <- c(p1,p2,p3), nrow = 3, ncol=3, byrow = T))
#-----------------------------------------------------------------------
#Chapter 4 ---------------------------------------------------------------
#Exercise 4.4 ------------------------------------------------------------
#Question 10  ------------------------------------------------------------

#Importing data set from library
library(MASS)
airquality
my_df <- airquality #saving data set in a new variable

clean_df <- function (x,col_idx){ #opening the clean_df function
  new_df <- x
  for (i in col_idx) { #beginning a for loop to iterate over missing values
    m <- c()
    missing_val <- length(which(is.na(new_df[,i])))
    m[1:missing_val] <- which(is.na(new_df[,i]))
    if(is.null(m)){ #Beginning of if/else loop
      next
    }
    else{
      #new_df <- x[-which(is.na(x[,col_idx[i]])),]
      new_df <- new_df[-m, ]
    } #end of if/else loop

  } #Closing the for loop
  return(new_df)
} #Closing clean_df function
```

```
#testing clean_df function
clean_df( x = my_df, col_idx = c(1:6))
```