Assignment - 1

1. What is the basic difference and similarity between a vector and a matrix?

   Differences:

Vector	Matrix
It is one dimensional.	It is two dimensional.
Present in either column or row.	Has rows and columns.
It represents magnitude and direction	It represents coefficient of linear equations.

   Similarities:

   In R both matrix and vector can have data types as logical, integer, double, character.

2. What is the basic difference and similarity between a data frame and a matrix?

Matrix	Data Frame
Contains only data with same data types.	Can contain data with multiple data types.
Cannot have no data.	Can have no data (N/A) in data frame.

   Similarities:

   Both are m*n arrays.

   The list of vectors consisting must be of equal length.

3. Create a vector using (15, TRUE, “World”). What happened to your result?

   The result will be a vector with 3 string values due to explicit coercion “15”, “TRUE” and “World”

4. John's scores in the final semester for the three subjects are 95, 91, and 88. The subjects are Statistics, Linear Algebra, and Calculus. Using these create a vector and give names to all elements of the vector based on their subjects.

```
subjects_vector <- c("Statistics", "Linear Algebra", "Calculus")
score_vector <- c(95, 91, 88)

names(score_vector) <- subjects_vector
score_vector

#Statistics Linear Algebra      Calculus
#95               91               88
```

5. Please check the types (character or numeric) of the vector you created.

```
typeof(subjects_vector)
#Result
#[1] "character"
 typeof(score_vector)
 #Result
 # [1] "double"
```

6. You have three students in your class (choose any name you want). You must create a matrix using their score in the above mentioned subjects (question 4) Student 1 (95, 91, and 88), Student 2(96, 94, and 97), Student 3(88, 98, and 85). Create a matrix and label column and row names.

```
 student_names <- c("Alex", "Nia", "Robin")
 alex_scores <- c(95, 91,88)
 nia_scores <- c(96, 94,97)
 robin_scores <- c(88, 98, 85)
 student_scores <- c(alex_scores, nia_scores, robin_scores)

student_matrix <- matrix(student_scores, nrow=3, byrow=TRUE, dimnames = list(
student_names, subjects_vector))
student_matrix
```

```
#Result
#Statistics Linear Algebra Calculus
#Alex          95              91      88
#Nia           96              94      97
#Robin         88              98      85
```

7. Convert the created matrix into a data frame.

```
student_dataframe <- data.frame(Alex = alex_scores, Nia = nia_scores, Robin= robin_scores)
rownames(student_dataframe) <- subjects_vector
student_dataframe

#Alex Nia Robin
#Statistics       95  96  88
#Linear Algebra   91  94      98
#Calculus         88  97  85
```

8. Create three vectors using five countries (your choice) from the following website. The first vector should be country names, the second vector should be the total number of cases, and the third vector should contain the total number of deaths. Create a data frame using these vectors.
   https://www.worldometers.info/coronavirus/

```
 covid_country_name <- c("USA","Brazil", "India", "Russia", "France")
 covid_total_cases <- c(30639264, 12136615, 11734058, 4483471, 4313073)
 covid_total_death <- c(556891, 298843, 160477, 96219, 92908)

covid_df <- data.frame(covid_total_cases, covid_total_death)
colnames(covid_df) <- c( "Total Cases", "Total Death")
rownames(covid_df) <- covid_country_name
covid_df

#Total Cases Total Death
#USA            30639264        556891
#Brazil 12136615        298843
#India  11734058        160477
#Russia         4483471         96219
#France         4313073         92908
```

9. Please read the mtcars data set from R. It is an built-in data set. Please check the structure of the data set. If required, please convert the data into their appropriate data types (character, logical, factor, etc). Save your results as a new data frame using a new name.

```
mtcars
str(mtcars)
#first 7 columns
first_7_col<- mtcars[,1:7]
factor(mtcars$vs)
#vs has two levels so we can convert it to logical data type
vs <- as.logical(as.integer(mtcars$vs))
factor(mtcars$am)
#am has two levels so we can convert it to logical data type
am<- as.logical(as.integer(mtcars$am))
#last two columns
last_2_col<- mtcars[,10:11]

new_mtcars_df <-data.frame(first_7_col, vs, am, last_2_col)
new_mtcars_df
```