LAB 2

2048015 - MANOJ KUMAR

25/01/2021

### 1. Create a vector different data types(Logical, Numeric, Integer, Complex, Character) and display their class and typeof each datatype.

# Logical Datatype  
  
Vector\_Logical <- c(TRUE, FALSE , TRUE, FALSE)  
sprintf('The class of Vector\_Logical is: %s', class(Vector\_Logical))

## [1] "The class of Vector\_Logical is: logical"

sprintf('The type of Vector\_Logical is: %s', typeof(Vector\_Logical))

## [1] "The type of Vector\_Logical is: logical"

# Numeric Datatype  
  
Vector\_Numeric <- c(2048015, 1948015, 1848015)  
sprintf('The class of Vector\_Numeric is: %s', class(Vector\_Numeric))

## [1] "The class of Vector\_Numeric is: numeric"

sprintf('The type of Vector\_Numeric is: %s', typeof(Vector\_Numeric))

## [1] "The type of Vector\_Numeric is: double"

# Integer Datatype  
  
Vector\_Integer <- c(75L, 85L)  
sprintf('The class of Vector\_Integer is: %s', class(Vector\_Integer))

## [1] "The class of Vector\_Integer is: integer"

sprintf('The type of Vector\_Integer is: %s', typeof(Vector\_Integer))

## [1] "The type of Vector\_Integer is: integer"

# Complex Datatype  
  
Vector\_Complex <- c(2 + 4i, 2 + 3i)  
sprintf('The class of Vector\_Complex is: %s', class(Vector\_Complex))

## [1] "The class of Vector\_Complex is: complex"

sprintf('The type of Vector\_Complex is: %s', typeof(Vector\_Complex))

## [1] "The type of Vector\_Complex is: complex"

#Character Datatype  
  
Vector\_Char <- c('Manoj', "Kumar", "MDS")  
sprintf('The class of Vector\_Char is: %s', class(Vector\_Char))

## [1] "The class of Vector\_Char is: character"

sprintf('The type of Vector\_Char is: %s', typeof(Vector\_Char))

## [1] "The type of Vector\_Char is: character"

### 2. Get and print the current working directory.

sprintf('The current directory is: %s', getwd())

## [1] "The current directory is: /Users/manojkumarr/Downloads/Other/Christ University/SEM 2/R/Lab"

# To set the directory.  
  
setwd("/Users/manojkumarr/Downloads/Other/Christ University/SEM 2/R/Lab")  
sprintf('The new directory is: %s', getwd())

## [1] "The new directory is: /Users/manojkumarr/Downloads/Other/Christ University/SEM 2/R/Lab"

### 3. Create this file using windows notepad by copying and pasting this data. Save the file as student.csv

S.No ,Sname, Degree, Total.marks, Grade

1,Andrew,UG,435,B

2,Babita,UG,210,D

3,Cathy,UG,459,A

4,Dominic,UG,542,A

5,Elsa,PG,520,B

6,Franko,PG,320,C

7,Gorang,UG,205,D

8,Harsha,PG,325,C

## student.csv file created.

### 

### 4.Save this file in the current working directory.

## File Saved in the current working directory

### 

### 5.Read the csv file in your current working directory.

student\_details<-read.csv("student.csv")  
student\_details

## S.No Sname Degree Total.marks Grade  
## 1 1 Andrew UG 435 B  
## 2 2 Babita UG 210 D  
## 3 3 Cathy UG 459 A  
## 4 4 Dominic UG 542 A  
## 5 5 Elsa PG 520 B  
## 6 6 Franko PG 320 C  
## 7 7 Gorang UG 205 D  
## 8 8 Harsha PG 325 C

### 

### 6.Check whether your CSV file is a dataframe and also check the number of rows and columns.

sprintf('The class of df is: %s', class(student\_details))

## [1] "The class of df is: data.frame"

# nrow(df) gives the number of rows in df.  
  
sprintf('Number of rows in df is: %s', nrow(student\_details))

## [1] "Number of rows in df is: 8"

# ncol(df) gives the number of columns in df.  
  
sprintf('Number of columns in df is: %s', ncol(student\_details))

## [1] "Number of columns in df is: 5"

### 

### 7. Apply all the functions sum(),mean(),sqrt()related to dataframe.

str(student\_details)

## 'data.frame': 8 obs. of 5 variables:  
## $ S.No : int 1 2 3 4 5 6 7 8  
## $ Sname : chr "Andrew" "Babita" "Cathy" "Dominic" ...  
## $ Degree : chr "UG" "UG" "UG" "UG" ...  
## $ Total.marks: int 435 210 459 542 520 320 205 325  
## $ Grade : chr "B" "D" "A" "A" ...

**Total marks is the only column that is numerical. So we will be performing the functions in that particular column only**

Total\_Marks <- student\_details$Total.marks  
Total\_Marks

## [1] 435 210 459 542 520 320 205 325

# SUM  
  
sprintf('Sum of Total marks is: %s', sum(Total\_Marks))

## [1] "Sum of Total marks is: 3016"

# MEAN  
  
sprintf('Mean of Total marks is: %s', mean(Total\_Marks))

## [1] "Mean of Total marks is: 377"

# Square root of Total Marks for each student is:-  
  
sprintf('Square root of Total marks is: %s',sqrt(Total\_Marks))

## [1] "Square root of Total marks is: 20.8566536146142"  
## [2] "Square root of Total marks is: 14.4913767461894"  
## [3] "Square root of Total marks is: 21.4242852856285"  
## [4] "Square root of Total marks is: 23.2808934536456"  
## [5] "Square root of Total marks is: 22.8035085019828"  
## [6] "Square root of Total marks is: 17.8885438199983"  
## [7] "Square root of Total marks is: 14.3178210632764"  
## [8] "Square root of Total marks is: 18.0277563773199"

### 

### 8.Get the highest marks from the data frame.

HighestMark<-max(student\_details$Total.marks)  
sprintf('The Highest Marks is: %s', HighestMark)

## [1] "The Highest Marks is: 542"

### 

### 9. Get the details of the person with highest marks.

# Case 1  
  
index=which.max(student\_details$Total.marks)  
student\_details[index,]

## S.No Sname Degree Total.marks Grade  
## 4 4 Dominic UG 542 A

# Case 2  
  
HighestScorer = subset(student\_details, Total.marks==max(student\_details$Total.marks))  
HighestScorer

## S.No Sname Degree Total.marks Grade  
## 4 4 Dominic UG 542 A

# Case 3  
  
student\_details[student\_details$Total.marks == max(student\_details$Total.marks),]

## S.No Sname Degree Total.marks Grade  
## 4 4 Dominic UG 542 A

### 

### 10. Get all the students in UG degree whose marks is greater than 300.

Students\_list1 <- student\_details[(student\_details$Degree == 'UG') & (student\_details$Total.marks > 300), ]  
Students\_list1

## S.No Sname Degree Total.marks Grade  
## 1 1 Andrew UG 435 B  
## 3 3 Cathy UG 459 A  
## 4 4 Dominic UG 542 A

### 

### 11. Add one more vector Date\_ of\_ Joining(DOJ) to the already existing dataframe.

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

doj <- c("20160505","20160507","20170509","20180517","20290621","20200625","20200627", "20200611")

student\_details$Date\_of\_Joining<-as.Date(doj, format = "%Y%m%d")  
student\_details

## S.No Sname Degree Total.marks Grade Date\_of\_Joining  
## 1 1 Andrew UG 435 B 2016-05-05  
## 2 2 Babita UG 210 D 2016-05-07  
## 3 3 Cathy UG 459 A 2017-05-09  
## 4 4 Dominic UG 542 A 2018-05-17  
## 5 5 Elsa PG 520 B 2029-06-21  
## 6 6 Franko PG 320 C 2020-06-25  
## 7 7 Gorang UG 205 D 2020-06-27  
## 8 8 Harsha PG 325 C 2020-06-11

### 

### 12. Get the details of the students who have joined after 2017.

Student\_list2 <- student\_details[year(student\_details$Date\_of\_Joining) > 2017, ]  
Student\_list2

## S.No Sname Degree Total.marks Grade Date\_of\_Joining  
## 4 4 Dominic UG 542 A 2018-05-17  
## 5 5 Elsa PG 520 B 2029-06-21  
## 6 6 Franko PG 320 C 2020-06-25  
## 7 7 Gorang UG 205 D 2020-06-27  
## 8 8 Harsha PG 325 C 2020-06-11

### 

### 13. Write the filtered data into a new file.

write.csv(Student\_list2, "/Users/manojkumarr/Downloads/Other/Christ University/SEM 2/R/Lab/result.csv")

### 