PFA Assignment 1: Data Types and Data Structures in R

Sayantan Paul; Student ID: 2021H1540807P

28/09/2021

R Assignment 1 - Data Types and Data Structures

Assignment Author

- Sayantan Paul | Student ID: 2021H1540807P

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Exercise 1

Find the class and type of 'x' in the following cases:-

1. x=22 // *Modify the code to create or declare an integer value.*

2. y= 2
   z=3
   x=y>z

3. x = 2i

4. x="20-09-2021"

```
x<- 22
class(x)
```

```
## [1] "numeric"
```

```
typeof(x)
```

```
## [1] "double"
```

```
y<-2
z<-3
x<- y>z
class(x)
```

```
## [1] "logical"
```

```
typeof(x)
```

```
## [1] "logical"
```

```
x<- 2i
class(x)
```

```
## [1] "complex"
```

```
typeof(x)
```

```
## [1] "complex"
```

```
x="20-09-2021"
class(x)
```

```
## [1] "character"
```

```
typeof(x)
```

```
## [1] "character"
```

Exercise 2

1. Find the output when 1+2i is converted to character type

2. Find output when "ProgrammingForAnalytics" is converted to numeric type
3. Given: x<-0:5, write code to output:
   [1] FALSE TRUE TRUE TRUE TRUE TRUE (and)
   [1] "0" "1" "2" "3" "4" "5"
4. Given: x<-c("a","b","c")
   Do all possible coercions to output [1] NA NA NA

```
#Find the output when 1+2i is converted to character type
x<- as.character(1+2i)
print(x)
```

```
## [1] "1+2i"
```

```
#Find output when "ProgrammingForAnalytics" is converted to numeric type
x<- as.numeric("ProgrammingForAnalytics")
```

```
## Warning: NAs introduced by coercion
```

```
print(x)
```

```
## [1] NA
```

```
#Given: x<-0:5, write code to output:
#[1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  (and)
#[1] "0" "1" "2" "3" "4" "5"

x<-0:5

print(as.logical(x))
```

```
## [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
print(as.character(x))
```

```
## [1] "0" "1" "2" "3" "4" "5"
```

```
#Given: x<-c("a","b","c"); Do all possible coercions to output [1] NA NA NA

x<-c("a","b","c")
print(as.complex(x))
```

```
## Warning in print(as.complex(x)): NAs introduced by coercion

## [1] NA NA NA
```

```
print(as.logical(x))
```

```
## [1] NA NA NA
```

```
print(as.double(x))
```

```
## Warning in print(as.double(x)): NAs introduced by coercion

## [1] NA NA NA
```

Exercise 3

Fill the table with your understanding of Data Structures (Atomic vector, List, Dataframe, Array, Matrix)

	Linear	2 Dimensional	N Dimensional
Homogenous			
Heterogenous			

Answer:

	Linear	2 Dimensional	N Dimensional
Homogenous	Atomic vector	Matrix	Array
Heterogenous	List	Data frame	

Exercise 4

Create a vector with a sequence of descending numbers from 20 to 0 in steps of 2.

(i) Write code to access all except the 2nd to 5th elements.
(ii) Write code to access all numbers greater than 10 excluding the one at 2nd index.
(iii) Write code to change values of all elements less than 10 to 0.

```
#a vector with a sequence of descending numbers from 20 to 0 in steps of 2
n <- seq(0,20, by = 2)
numbers<- sort(n, decreasing = TRUE)
print(numbers)
```

```
##  [1] 20 18 16 14 12 10  8  6  4  2  0
```

```
#code to access all except the 2nd to 5th elements
i <- numbers[-c(2,5)]
print(i)
```

```
## [1] 20 16 14 10  8  6  4  2  0
```

```
#Write code to access all numbers greater than 10 excluding the one at 2nd index
x<-numbers[-2]
y <- which(x>10)
print(x[y])
```

```
## [1] 20 16 14 12
```

```
#Write code to change values of all elements less than 10 to 0
z<- which(numbers<10)
numbers = replace(numbers,z,0)
print(numbers)
```

```
##  [1] 20 18 16 14 12 10  0  0  0  0  0
```

Exercise 5

Create a matrix with 2 columns and 4 rows by passing a vector having 4 repetitions of 1 and 2 (i.e., 1,2,1,2,. . . use rep() command). Arrange these elements in a row-wise manner.

(i) Write code to access the 2nd column of this matrix.

(ii) Name the columns: "c1", "c2". Name the rows: "r1", "r2", "r3", "r4".

(iii) Write code to access the 2nd row using its row name.

(iv) Delete the first row

```
#Create a matrix with 2 columns and 4 rows by passing a vector having 4 repetitions of 1 and 2 (i.e., 1
y <- matrix(rep(1:2), nrow =4, ncol=2, byrow = TRUE)
print(y)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    2
## [3,]    1    2
## [4,]    1    2
```

```
#code to access the 2nd column of this matrix
print(y[,2])
```

```
## [1] 2 2 2 2
```

```
#Name the columns: "c1", "c2". Name the rows: "r1", "r2", "r3", "r4"
rownames = c("r1", "r2", "r3","r4")
colnames = c("c1", "c2")
y <- matrix(rep(1:2), nrow =4, ncol=2, byrow = TRUE,dimnames = list(rownames, colnames))
print(y)
```

```
##    c1 c2
## r1  1  2
## r2  1  2
## r3  1  2
## r4  1  2
```

```
#code to access the 2nd row using its row name
print(y["r2",])
```

```
## c1 c2
##  1  2
```

```
#Delete the first row
y <- y[-1,]
print(y)
```

```
##    c1 c2
## r2  1  2
## r3  1  2
## r4  1  2
```

Exercise 6

(i) Create a vector “V” which contains 10 random integer values between -100 and +100. (hint:)
(ii) Create a two-dimensional 5×5 array “A” comprised of sequence of even integers greater than 25.
(iii) Create a list “S” containing sequence of 20 capital letters, starting with ‘C’.

Create a list named “l” containing all the previously created objects. Name them “my_vector”, “my_array” and “my_list” respectively.

Without running any R command, answer the following questions pertaining to the exercise :-

1. How many elements are there in the list?
2. what is the result of l[[3]]?
3. How would you access random-th letter in the list element “my_list”?
4. If you convert list l to a vector, what will be the type of it’s elements?
5. Can this list be converted to an array? What will be the data type of elements in array?
6. How would you add a new element to this list?

```
#a vector "V" which contains 10 random integer values between -100 and +100
v <- sample(-100:100, 10, replace=TRUE)

#a two-dimensional 5x5 array "A" comprised of sequence of even integers greater than 25
a <- array(seq(from = 26, length.out = 25, by = 2), c(5, 5))

#a list "S" containing sequence of 20 capital letters, starting with 'C'
s <- LETTERS[match("C", LETTERS):(match("C", LETTERS)+19)]

#a list named "l" containing all the previously created objects. Name them "my_vector","my_array" and "
l <- list(my_vector = v, my_array = a, my_list = s)
```

1. How many elements are there in the list? Ans. 3

2. what is the result of l[[3]]? Ans: It will start from C and take next 19 elements till V

3. How would you access random-th letter in the list element “my_list”? Ans: l[[3]][sample(1:length(l[[3]]), 1)]

4. If you convert list l to a vector, what will be the type of it’s elements? Ans: Character

5. Can this list be converted to an array? What will be the data type of elements in array? Ans: Yes. Data type will be List

6. How would you add a new element to this list? Ans: l$my_{l}ist < -append(”W”, l$my_list)

Exercise 7

Write a program to create a Data Frame by passing vectors for name (character), age (integer) and vaccinated (logical).

1. Print the number of rows using dim().
2. Write code to change the age column into complex data type.
3. Use “as” function to check if dataframes can be coerced into other data types or data structures.

```r
#a program to create a Data Frame by passing vectors for name (character), age (integer) and vaccinated
name <- c("a","b","c","d","e")
age <- as.integer(c(25,26,27,28,29))
vaccinated <- as.logical(c("F","T","T","F","T"))

x <- data.frame(name,
                age,
                vaccinated
                )
x
```

```
##   name age vaccinated
## 1    a  25      FALSE
## 2    b  26       TRUE
## 3    c  27       TRUE
## 4    d  28      FALSE
## 5    e  29       TRUE
```

```r
View(x)
#Print the number of rows using dim()
dim(x)
```

```
## [1] 5 3
```

```r
nrow(x)
```

```
## [1] 5
```

```r
#code to change the age column into complex data type
class(x)
```

```
## [1] "data.frame"
```

```r
class(x$age) = "Complex"
class(x$age)
```

```
## [1] "Complex"
```

```r
#Use of "as" function to check if dataframes can be coerced into other data types or data structure
x<- lapply(x, as.vector)
print(x)
```

```
## $name
## [1] "a" "b" "c" "d" "e"
##
## $age
## [1] 25 26 27 28 29
##
## $vaccinated
## [1] FALSE  TRUE  TRUE FALSE  TRUE
```

Exercise 8

Debug the following and run the correct code
1. num = c(1, 2, 3, 4, 5)
name = c("one", "two", "four", "five")
df = data.frame(num,name)
print(df)
2. x = c(1, "BITS", 5, 7.2, True, 1+i)
print(X)

```
num = c(1, 2, 3, 4, 5)
name = c("one", "two", "four", "five",NA)
df = data.frame(num,name,check.rows= TRUE)
print(df)
```

```
##   num name
## 1   1  one
## 2   2  two
## 3   3 four
## 4   4 five
## 5   5 <NA>
```

```
x = c(1, "BITS", 5, 7.2, TRUE, "1+i")
print(x)
```

```
## [1] "1"    "BITS" "5"    "7.2"  "TRUE" "1+i"
```