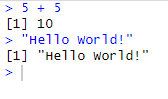
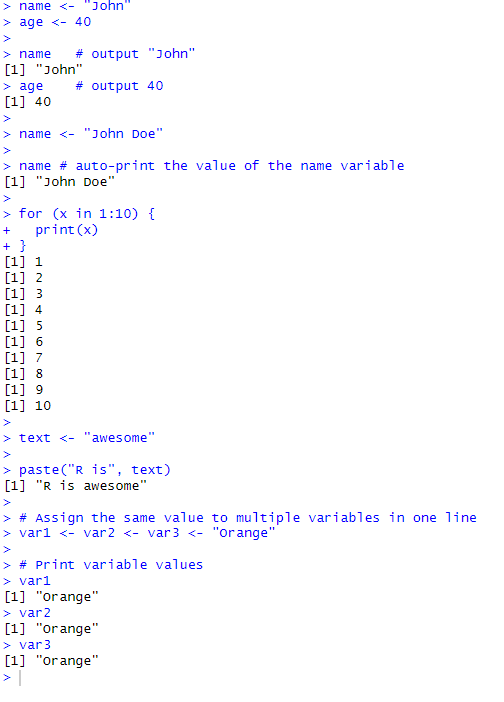
**Assignment No:** 1

**Assignment Topic:** Introduction to R Programming Language.

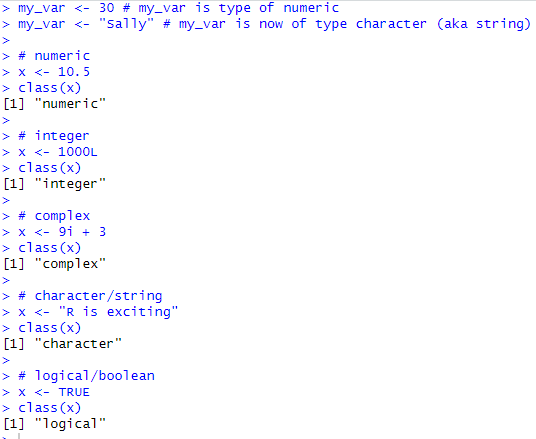
1. **R Syntax**



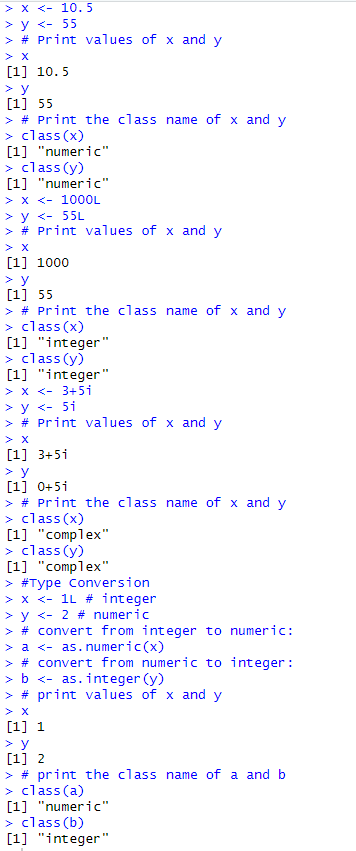
1. **R Variables**



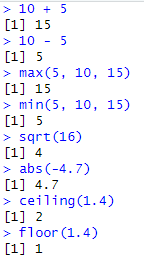
1. **R Data Types**



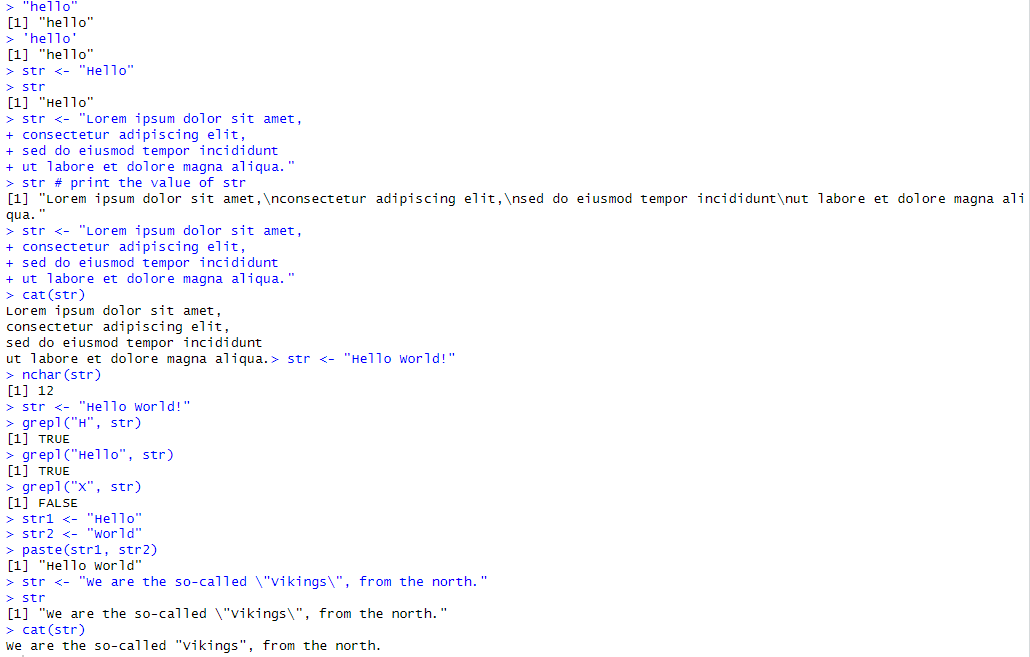
1. **R Numbers**



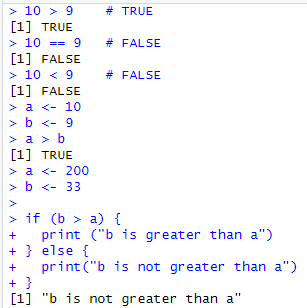
1. **R Math**

****

1. **R Strings**

****

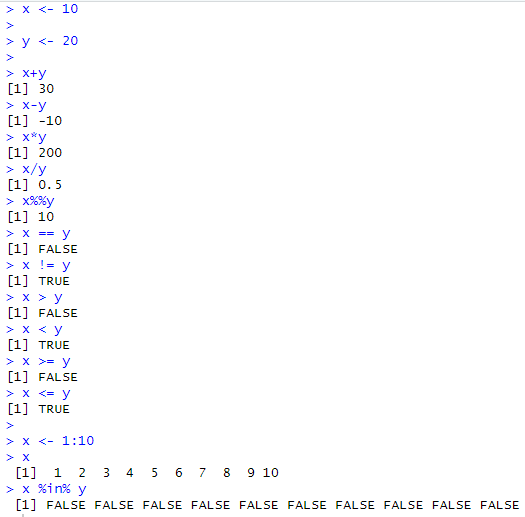
1. **R Booleans / Logical Values**

****

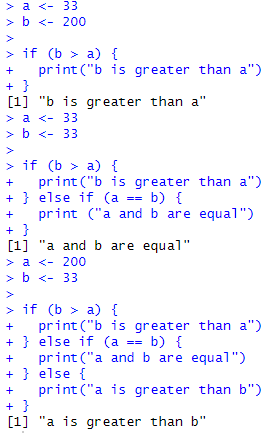
1. **R Operators**

R divides the operators in the following groups:

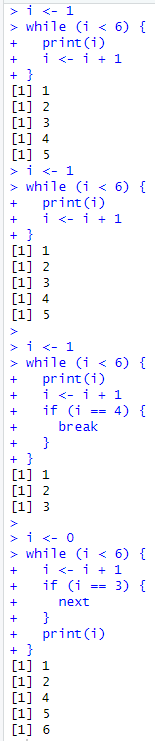
* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Miscellaneous operators

****

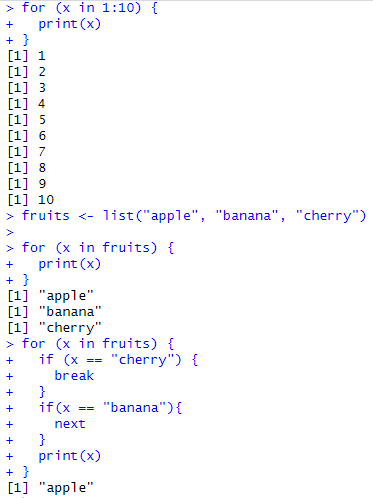
1. **R if…………….else**

****

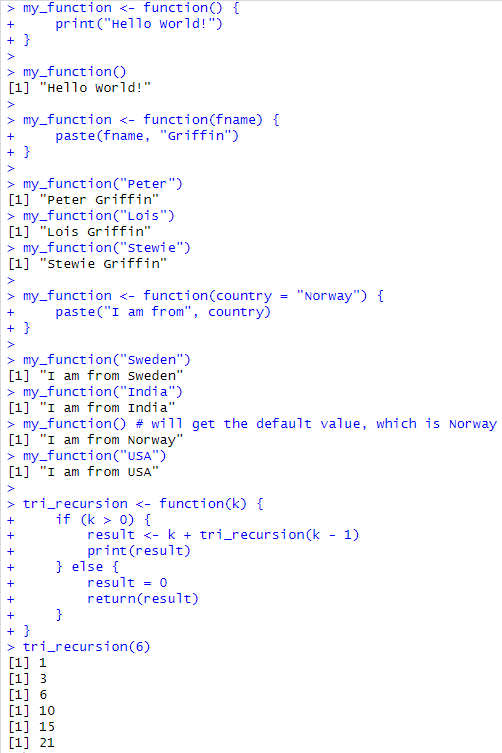
1. **R while loop**

****

1. **R for loops**

****

1. **R Functions**

****

**R Data Structures.**

* 1. **R Vectors**

**Source Code and Output:**

> # Vector of strings

> fruits <- c("banana", "apple", "orange")

>

> # Print fruits

> fruits

[1] "banana" "apple" "orange"

> # Vector of numerical values

> numbers <- c(1, 2, 3)

>

> # Print numbers

> numbers

[1] 1 2 3

> # Vector with numerical decimals in a sequence

> numbers1 <- 1.5:6.5

> numbers1

[1] 1.5 2.5 3.5 4.5 5.5 6.5

>

> # Vector with numerical decimals in a sequence where the last element is not used

> numbers2 <- 1.5:6.3

> numbers2

[1] 1.5 2.5 3.5 4.5 5.5

> fruits <- c("banana", "apple", "orange", "mango", "lemon")

> numbers <- c(13, 3, 5, 7, 20, 2)

>

> sort(fruits) # Sort a string

[1] "apple" "banana" "lemon" "mango" "orange"

> sort(numbers) # Sort numbers

[1] 2 3 5 7 13 20

> fruits <- c("banana", "apple", "orange")

>

> # Access the first item (banana)

> fruits[1]

[1] "banana"

> fruits <- c("banana", "apple", "orange", "mango", "lemon")

>

> # Change "banana" to "pear"

> fruits[1] <- "pear"

>

> # Print fruits

> fruits

[1] "pear" "apple" "orange" "mango" "lemon"

> numbers <- seq(from = 0, to = 100, by = 20)

>

> numbers

[1] 0 20 40 60 80 100

* 1. **R Lists.**

**Source Code and Output:**

> # List of strings

> thislist <- list("apple", "banana", "cherry")

>

> # Print the list

> thislist

[[1]]

[1] "apple"

[[2]]

[1] "banana"

[[3]]

[1] "cherry"

> thislist <- list("apple", "banana", "cherry")

>

> thislist[1]

[[1]]

[1] "apple"

> thislist <- list("apple", "banana", "cherry")

> thislist[1] <- "blackcurrant"

>

> # Print the updated list

> thislist

[[1]]

[1] "blackcurrant"

[[2]]

[1] "banana"

[[3]]

[1] "cherry"

> thislist <- list("apple", "banana", "cherry")

>

> length(thislist)

[1] 3

> thislist <- list("apple", "banana", "cherry")

>

> "apple" %in% thislist

[1] TRUE

> thislist <- list("apple", "banana", "cherry")

>

> append(thislist, "orange")

[[1]]

[1] "apple"

[[2]]

[1] "banana"

[[3]]

[1] "cherry"

[[4]]

[1] "orange"

> thislist <- list("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

>

> (thislist)[2:5]

[[1]]

[1] "banana"

[[2]]

[1] "cherry"

[[3]]

[1] "orange"

[[4]]

[1] "kiwi"

> thislist <- list("apple", "banana", "cherry")

>

> for (x in thislist) {

+ print(x)

+ }

[1] "apple"

[1] "banana"

[1] "cherry"

* 1. **R Matrices**

**Source Code and Output:**

> # Create a matrix

> thismatrix <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)

>

> # Print the matrix

> thismatrix

[,1] [,2]

[1,] 1 4

[2,] 2 5

[3,] 3 6

> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"), nrow = 2, ncol = 2)

>

> thismatrix[1, 2]

[1] "cherry"

> thismatrix <- matrix(c("apple", "banana", "cherry", "orange","grape", "pineapple", "pear", "melon", "fig"), nrow = 3, ncol = 3)

>

> thismatrix[c(1,2),]

[,1] [,2] [,3]

[1,] "apple" "orange" "pear"

[2,] "banana" "grape" "melon"

> thismatrix <- matrix(c("apple", "banana", "cherry", "orange","grape", "pineapple", "pear", "melon", "fig"), nrow = 3, ncol = 3)

>

> newmatrix <- cbind(thismatrix, c("strawberry", "blueberry", "raspberry"))

>

> # Print the new matrix

> newmatrix

[,1] [,2] [,3] [,4]

[1,] "apple" "orange" "pear" "strawberry"

[2,] "banana" "grape" "melon" "blueberry"

[3,] "cherry" "pineapple" "fig" "raspberry"

> thismatrix <- matrix(c("apple", "banana", "cherry", "orange", "mango", "pineapple"), nrow = 3, ncol =2)

>

> #Remove the first row and the first column

> thismatrix <- thismatrix[-c(1), -c(1)]

>

> thismatrix

[1] "mango" "pineapple"

> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"), nrow = 2, ncol = 2)

>

> for (rows in 1:nrow(thismatrix)) {

+ for (columns in 1:ncol(thismatrix)) {

+ print(thismatrix[rows, columns])

+ }

+ }

[1] "apple"

[1] "cherry"

[1] "banana"

[1] "orange"

* 1. **R Arrays**

**Source Code and Output:**

> # An array with one dimension with values ranging from 1 to 24

> thisarray <- c(1:24)

> thisarray

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

>

> # An array with more than one dimension

> multiarray <- array(thisarray, dim = c(4, 3, 2))

> multiarray

, , 1

[,1] [,2] [,3]

[1,] 1 5 9

[2,] 2 6 10

[3,] 3 7 11

[4,] 4 8 12

, , 2

[,1] [,2] [,3]

[1,] 13 17 21

[2,] 14 18 22

[3,] 15 19 23

[4,] 16 20 24

> thisarray <- c(1:24)

> multiarray <- array(thisarray, dim = c(4, 3, 2))

>

> multiarray[2, 3, 2]

[1] 22

> thisarray <- c(1:24)

>

> # Access all the items from the first row from matrix one

> multiarray <- array(thisarray, dim = c(4, 3, 2))

> multiarray[c(1),,1]

[1] 1 5 9

>

> # Access all the items from the first column from matrix one

> multiarray <- array(thisarray, dim = c(4, 3, 2))

> multiarray[,c(1),1]

[1] 1 2 3 4

> for(x in multiarray){

+ print(x)

+ }

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

[1] 6

[1] 7

[1] 8

[1] 9

[1] 10

[1] 11

[1] 12

[1] 13

[1] 14

[1] 15

[1] 16

[1] 17

[1] 18

[1] 19

[1] 20

[1] 21

[1] 22

[1] 23

[1] 24

* 1. **R Data Frames**

**Source Code and Output:**

> # Create a data frame

> Data\_Frame <- data.frame (

+ Training = c("Strength", "Stamina", "Other"),

+ Pulse = c(100, 150, 120),

+ Duration = c(60, 30, 45)

+ )

>

> # Print the data frame

> Data\_Frame

Training Pulse Duration

1 Strength 100 60

2 Stamina 150 30

3 Other 120 45

> Data\_Frame <- data.frame (

+ Training = c("Strength", "Stamina", "Other"),

+ Pulse = c(100, 150, 120),

+ Duration = c(60, 30, 45)

+ )

>

> Data\_Frame

Training Pulse Duration

1 Strength 100 60

2 Stamina 150 30

3 Other 120 45

>

> summary(Data\_Frame)

Training Pulse Duration

Length:3 Min. :100.0 Min. :30.0

Class :character 1st Qu.:110.0 1st Qu.:37.5

Mode :character Median :120.0 Median :45.0

Mean :123.3 Mean :45.0

3rd Qu.:135.0 3rd Qu.:52.5

Max. :150.0 Max. :60.0

> #We can use single brackets [ ], double brackets [[ ]] or $ to access columns from a data frame:

> Data\_Frame <- data.frame (

+ Training = c("Strength", "Stamina", "Other"),

+ Pulse = c(100, 150, 120),

+ Duration = c(60, 30, 45)

+ )

>

> Data\_Frame[1]

Training

1 Strength

2 Stamina

3 Other

>

> Data\_Frame[["Training"]]

[1] "Strength" "Stamina" "Other"

>

> Data\_Frame$Training

[1] "Strength" "Stamina" "Other"

* 1. **R Factors**

Factors are used to categorize data. Examples of factors are:

* Demography: Male/Female
* Music: Rock, Pop, Classic, Jazz
* Training: Strength, Stamina

To create a factor, use the factor() function and add a vector as argument:

> # Create a factor

> music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"))

>

> # Print the factor

> music\_genre

[1] Jazz Rock Classic Classic Pop Jazz Rock Jazz

Levels: Classic Jazz Pop Rock

> music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"), levels = c("Classic", "Jazz", "Pop", "Rock", "Other"))

>

> levels(music\_genre)

[1] "Classic" "Jazz" "Pop" "Rock" "Other"

> music\_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"), levels = c("Classic", "Jazz", "Pop", "Rock", "Opera"))

>

> music\_genre[3] <- "Opera"

>

> music\_genre[3]

[1] Opera

Levels: Classic Jazz Pop Rock Opera