**Batch: A-3 Roll No.: 16014022050**

**Experiment No. 1**

|  |
| --- |
| **Title:** Exploring R for Data Science. |

**Aim:**

To understand basics of R - Operators, built-in functions, Data types, Data manipulation in R, R packages for Data Science.

**Expected Outcome of Experiment:**

**CO: Implementation of basic statistical modelling and analysis using R Programming.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books / Journals / Websites referred:**

1. <https://cran.r-project.org/>
2. <https://www.codecademy.com/learn/learn-r>
3. <https://www.w3schools.com/r/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Theory:**

**What is R?**

* R is a scripting/programming language and environment for statistical computing, data science and graphics.
* R is a successor of the proprietary statistical computing programming language S.
* It is an important tool for computational statistics, visualization and data science.

**Why R?**

It provides techniques for various statistical analyses like classical tests and classification, time-series analysis, clustering, linear and non-linear modelling and graphical operations.

It has superior support for graphics.

Reasons for learning R:

* Free, Open source
* Great visualization
* Cross-platform compatibility
* Advanced statistics
* Integration with other programming languages
* Supportive open-source community
* Easy extensibility via packages

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Exploring the atomic datatypes supported by R-Logical, Numeric-integer, Character, Double, Complex, Raw.**

> # numeric

> x <- 10.5

> x

[1] 10.5

>

> # integer

> y <- 1000L

> y

[1] 1000

>

> # character / string

> z <- "this is fun!"

> z

[1] "this is fun!"

>

> # complex

> complex <- 9i + 3

> complex

[1] 3+9i

>

> # logical / boolean

> bool <- TRUE

> bool

[1] TRUE

>

> # raw

> a <- charToRaw("R CODER")

> a

[1] 52 20 43 4f 44 45 52

> typeof(a)

[1] "raw"

>

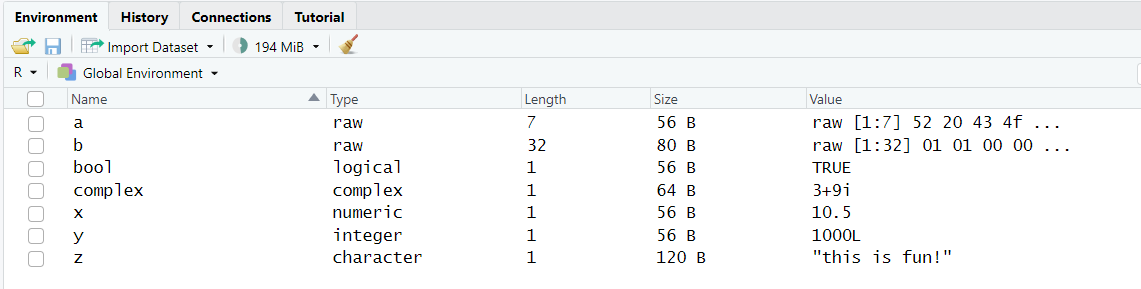
> b <- intToBits(3L)

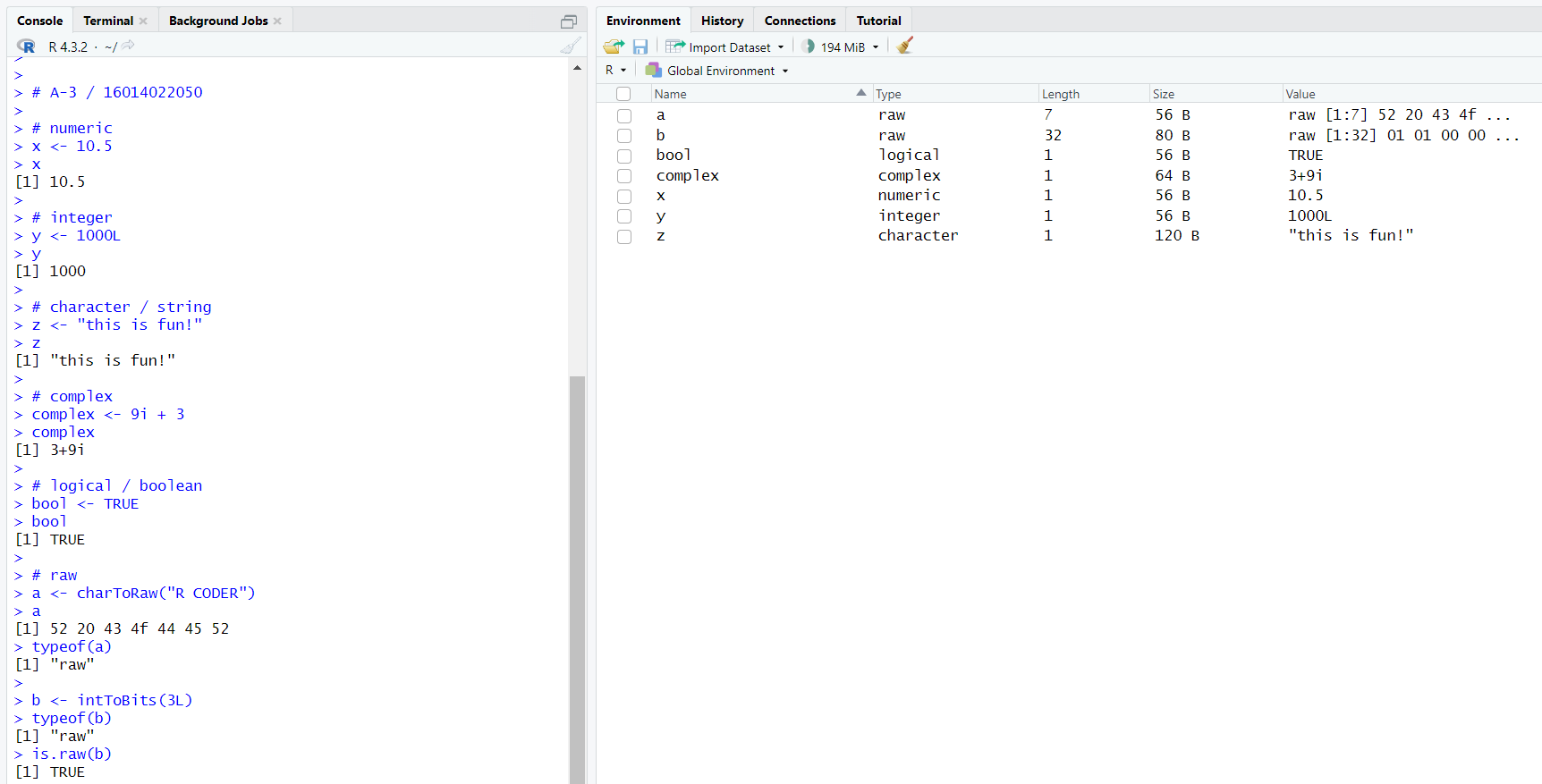
> typeof(b)

[1] "raw"

> is.raw(b)

[1] TRUE

****

****

1. **Exploring data manipulation of different data objects of R- Vectors- Matrices, Factors, List, Array, Data Frames.**

> # vectors

> numeric\_vector <- c(1, 2, 3, 4, 5)

> character\_vector <- c("apple", "cherry", "banana")

> logical\_vector <- c(TRUE, FALSE, TRUE)

>

> # matrices

> matrix\_data <- matrix(1:6, nrow = 2, ncol = 3, byrow = TRUE)

>

> # factors

> factor\_variable <- factor(c("low", "medium", "high", "low", "medium"))

>

> # lists

> list\_variable <- list(name = "John", age = 25, city = "New York")

>

> # arrays

> array\_data <- array(1:24, dim = c(2, 3, 4))

>

> # data drames

> data\_frame\_variable <- data.frame(

+ name = c("Alice", "Bob", "Charlie"),

+ age = c(30, 25, 35),

+ city = c("Chicago", "New York", "San Francisco")

+ )

>

> print("Numeric Vector:")

[1] "Numeric Vector:"

> print(numeric\_vector)

[1] 1 2 3 4 5

>

> print("\nCharacter Vector:")

[1] "\nCharacter Vector:"

> print(character\_vector)

[1] "apple" "cherry" "banana"

>

> print("\nLogical Vector:")

[1] "\nLogical Vector:"

> print(logical\_vector)

[1] TRUE FALSE TRUE

>

> print("\nMatrix:")

[1] "\nMatrix:"

> print(matrix\_data)

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

[1,] 1 2 3

[2,] 4 5 6

>

> print("\nFactor Variable:")

[1] "\nFactor Variable:"

> print(factor\_variable)

[1] low medium high low medium

Levels: high low medium

>

> print("\nList:")

[1] "\nList:"

> print(list\_variable)

$name

[1] "John"

$age

[1] 25

$city

[1] "New York"

>

> print("\nArray:")

[1] "\nArray:"

> print(array\_data)

, , 1

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

[1,] 1 3 5

[2,] 2 4 6

, , 2

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

[1,] 7 9 11

[2,] 8 10 12

, , 3

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

[1,] 13 15 17

[2,] 14 16 18

, , 4

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

[1,] 19 21 23

[2,] 20 22 24

>

> print("\nData Frame:")

[1] "\nData Frame:"

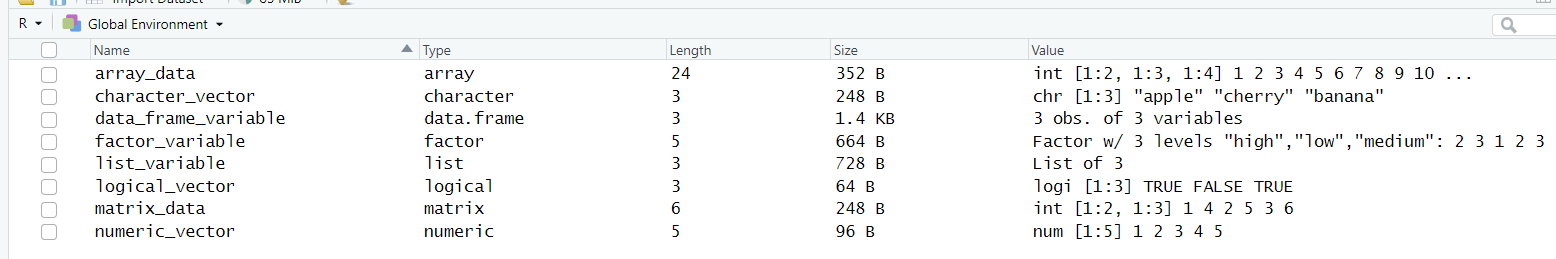
> print(data\_frame\_variable)

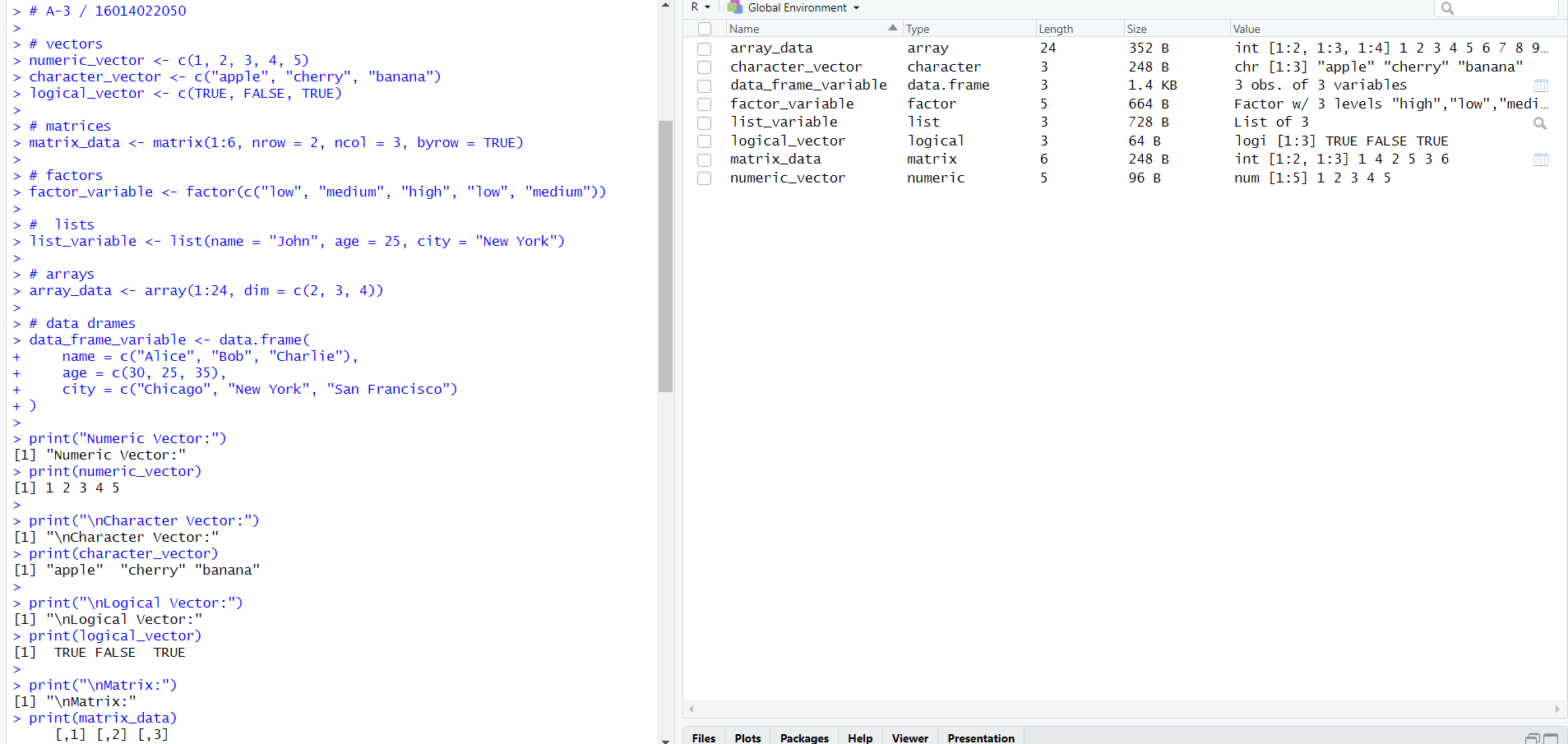
name age city

1 Alice 30 Chicago

2 Bob 25 New York

3 Charlie 35 San Francisco



****

1. **Exploring Operators and built-in functions and writing user-defined functions in R.**

> # arithmetics operator

>

> a <- 10

> b <- 5

>

> addition\_result <- a + b

> addition\_result

[1] 15

>

> subtraction\_result <- a - b

> subtraction\_result

[1] 5

>

> multiplication\_result <- a \* b

> multiplication\_result

[1] 50

>

> division\_result <- a / b

> division\_result

[1] 2

>

> modulus\_result <- a %% b

> modulus\_result

[1] 0

>

> exponentiation\_result <- a ^ b

> exponentiation\_result

[1] 1e+05

>

>

> # comparison operator

>

> greater\_than\_result <- a > b

> greater\_than\_result

[1] TRUE

>

> less\_than\_result <- a < b

> less\_than\_result

[1] FALSE

>

> equal\_to\_result <- a == b

> equal\_to\_result

[1] FALSE

>

> not\_equal\_to\_result <- a != b

> not\_equal\_to\_result

[1] TRUE

>

>

> # logical operators

>

> logical\_and\_result <- TRUE & FALSE

> logical\_and\_result

[1] FALSE

>

> logical\_or\_result <- TRUE | FALSE

> logical\_or\_result

[1] TRUE

>

> logical\_not\_result <- !TRUE

> logical\_not\_result

[1] FALSE

>

>

> # built-in functions

>

> vector\_example <- c(2, 4, 6, 8, 10)

> sum(vector\_example)

[1] 30

> mean(vector\_example)

[1] 6

> length(vector\_example)

[1] 5

> max(vector\_example)

[1] 10

> min(vector\_example)

[1] 2

>

>

> # user-defined function

> multiply\_by\_two <- function(x) {

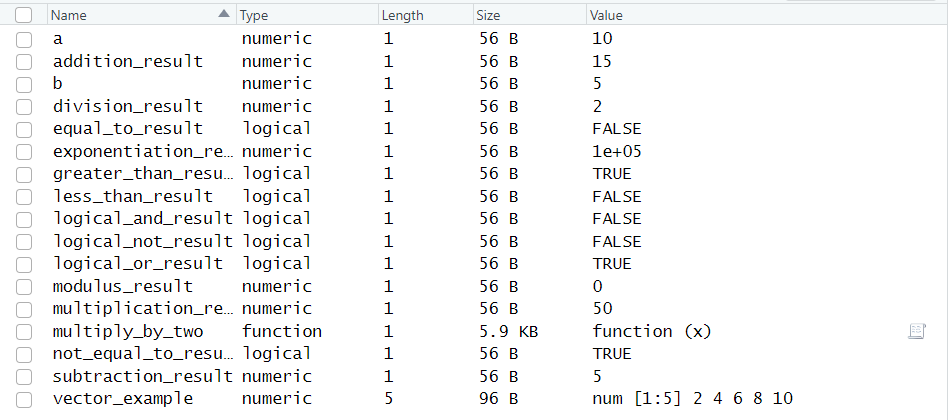
+ result <- x \* 2

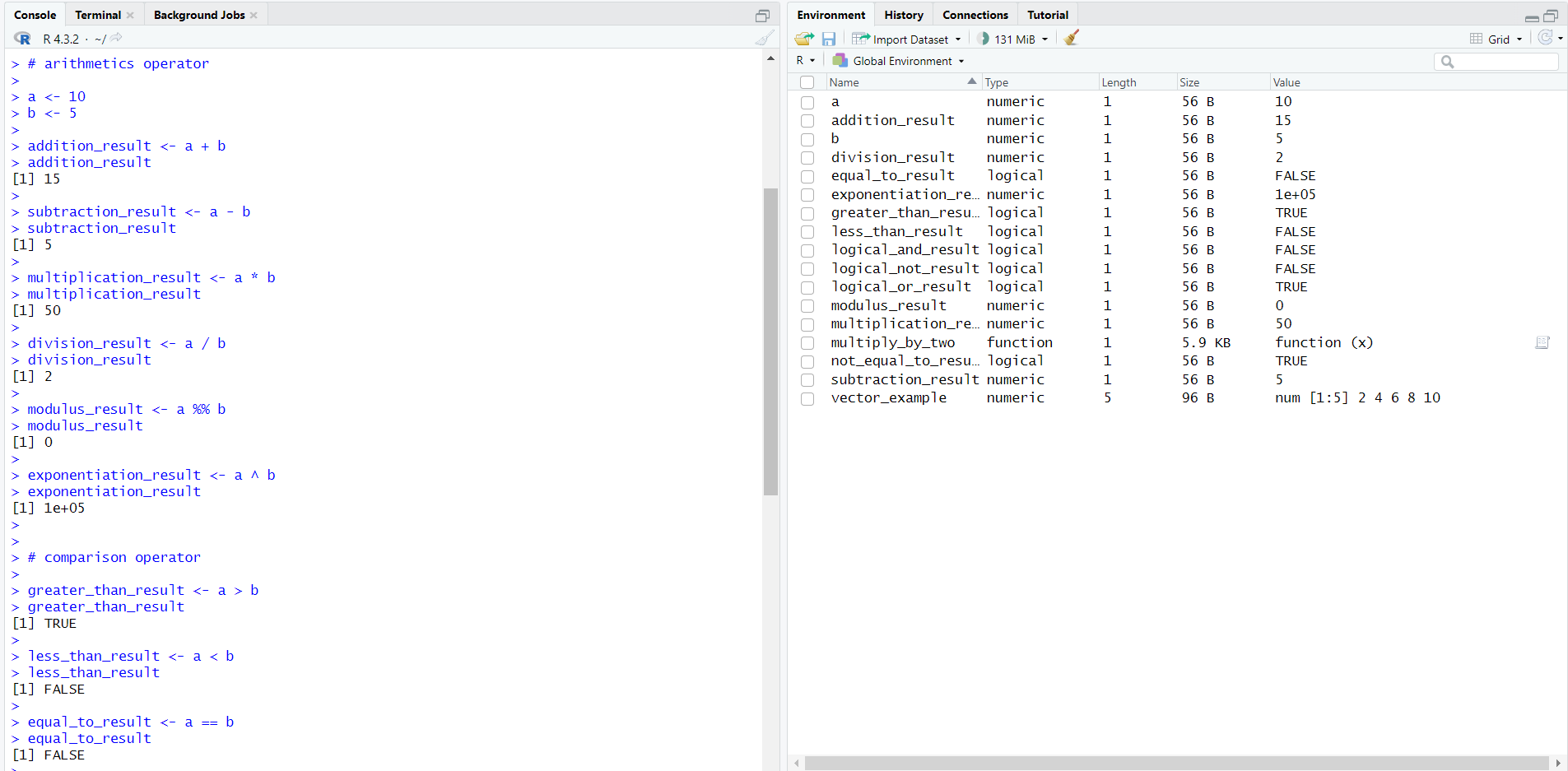
+ return(result)

+ }

> multiply\_by\_two(7)

[1] 14

****

****

1. **Using Looping constructs in R.**

> # for loop

>

> for (i in 1:5) {

+ print(i)

+ }

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

>

>

> # while loop

>

> counter <- 1

> while (counter <= 5) {

+ print(counter)

+ counter <- counter + 1

+ }

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

>

>

> # if loop

>

> for (i in 1:10) {

+ if (i %% 2 == 0) {

+ print("even")

+ } else {

+ print("odd")

+ }

+ }

[1] "odd"

[1] "even"

[1] "odd"

[1] "even"

[1] "odd"

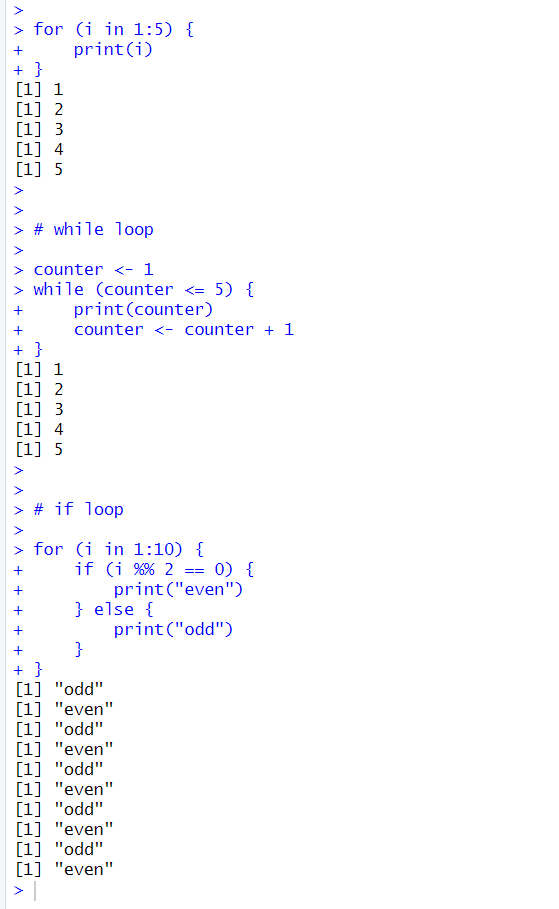
[1] "even"

[1] "odd"

[1] "even"

[1] "odd"

[1] "even"

****

1. **Exploring any Packages in R (any graphic package).**

ggplot2 is an R package for data visualization based on the Grammar of Graphics, which was introduced by Leland Wilkinson. The package is designed to make it easy to create complex and informative statistical graphics in R. The name "ggplot2" stands for "Grammar of Graphics plot 2.

> if (!requireNamespace("ggplot2", quietly = TRUE)) {

+ install.packages("ggplot2")

+ }

|  |
| --- |
| > library(ggplot2)  >  > # create sample data  > data <- data.frame(  + x = c(1, 2, 3, 4, 5),  + y = c(2, 4, 1, 7, 3)  + )  >  > # create scatter plot using ggplot2  > scatter\_plot <- ggplot(data, aes(x = x, y = y)) +  + geom\_point(color = "blue", size = 3) +  + labs(title = "Scatter Plot Example", x = "X-axis", y = "Y-axis")  >  > # displaying scatter plot  > print(scatter\_plot) |
|  |
| |  | | --- | |  | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Post-lab Questions:**

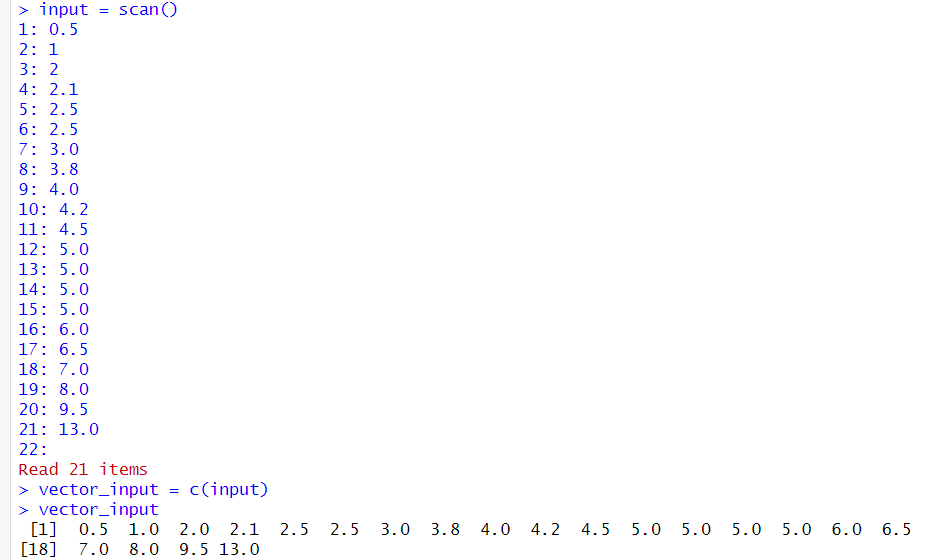
Write R commands for the following:

1. In an article in American Journal of Pathology, Pitts et al (2001) have taken the measurements on diameters in centimetres of the neoplasm removed from the breasts of 21 subjects with pure sarcoma.

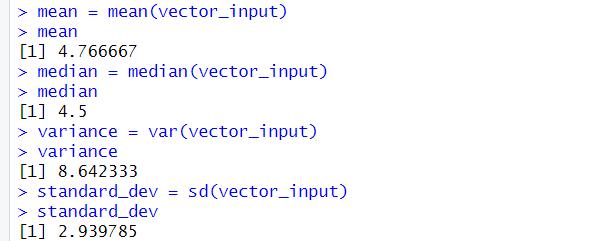
Following is the dataset:

0.5, 1.0, 2.0, 2.1, 2.5, 2.5, 3.0, 3.8, 4.0, 4.2, 4.5, 5.0, 5.0, 5.0, 5.0, 6.0, 6.5, 7.0, 8.0, 9.5, 13.0

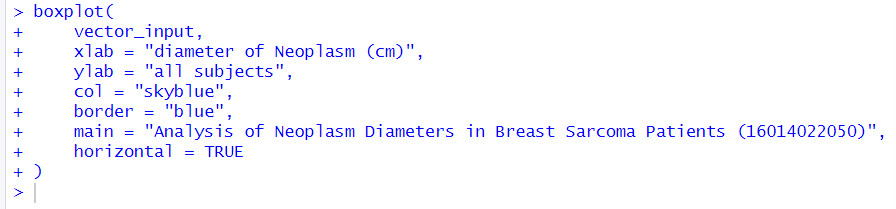
1. Enter the dataset using scan function and store in the variable X.

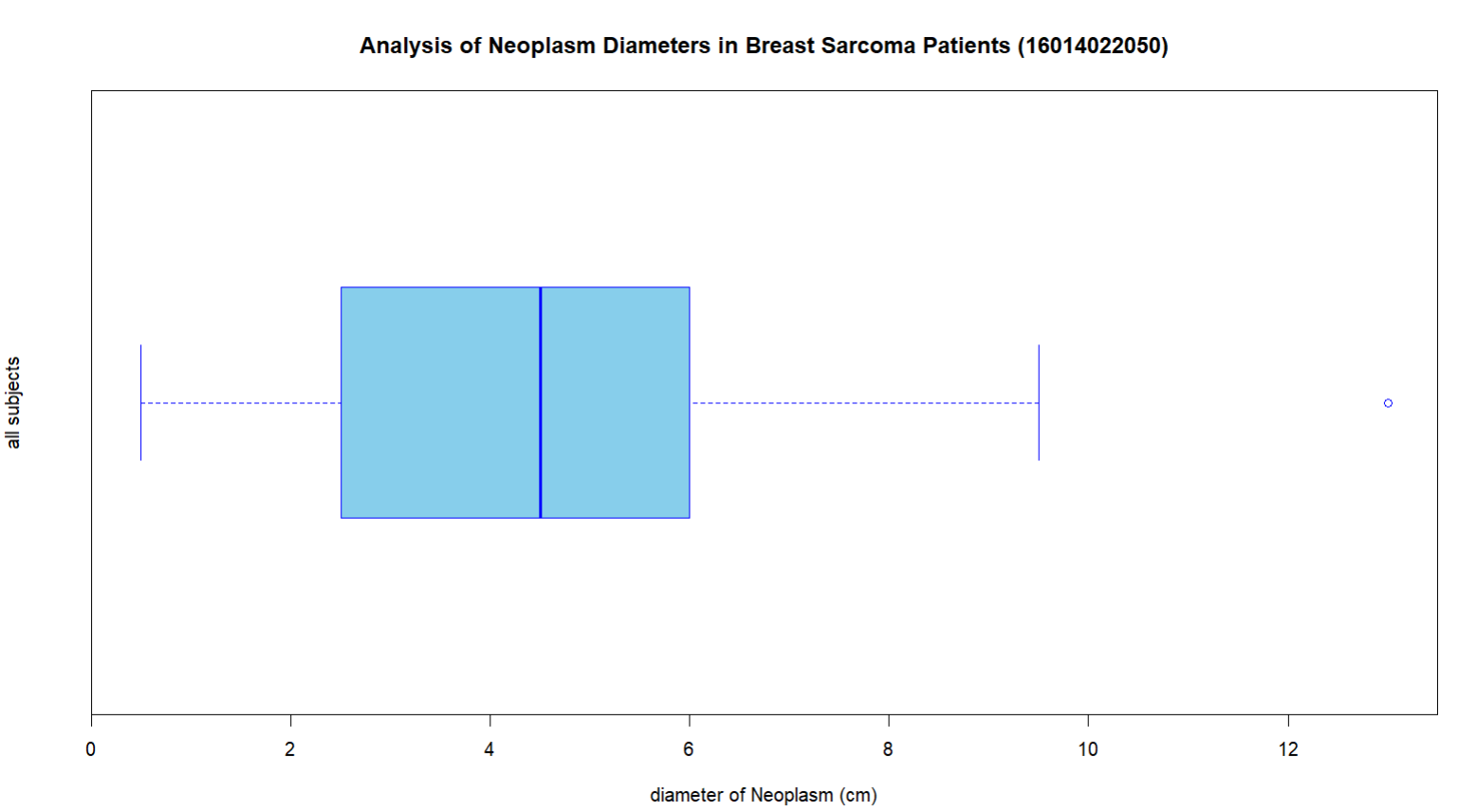


1. Find the mean, median, variance and standard deviation of x.



1. Create the boxplot.



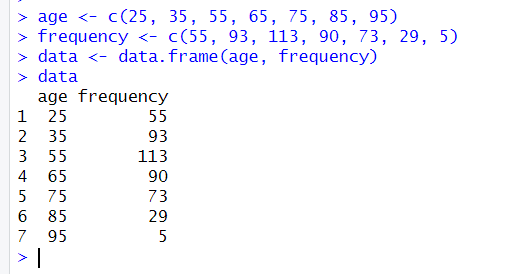


1. American Journal of psychiatry conducted a study of the presence of significant psychiatric illness in heterozygous carriers of the gene for the Wolfram syndrome. Among the subject studied were 543 blood relatives of patients of Wolfram syndrome.

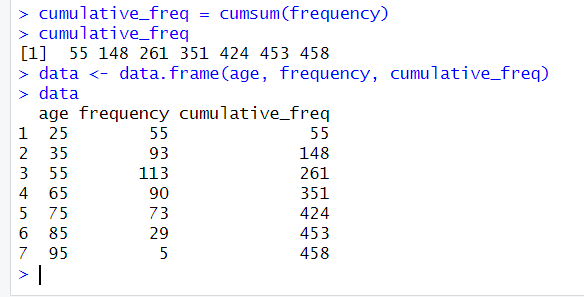
Following is the frequency distribution of ages of these blood relatives:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Age (Mid-Point) | 25 | 35 | 55 | 65 | 75 | 85 | 95 |
| Number (Frequency) | 55 | 93 | 113 | 90 | 73 | 29 | 5 |

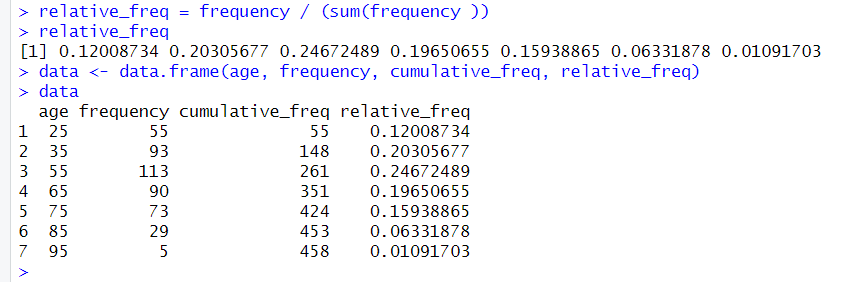
1. Enter the dataset using dataframe command.



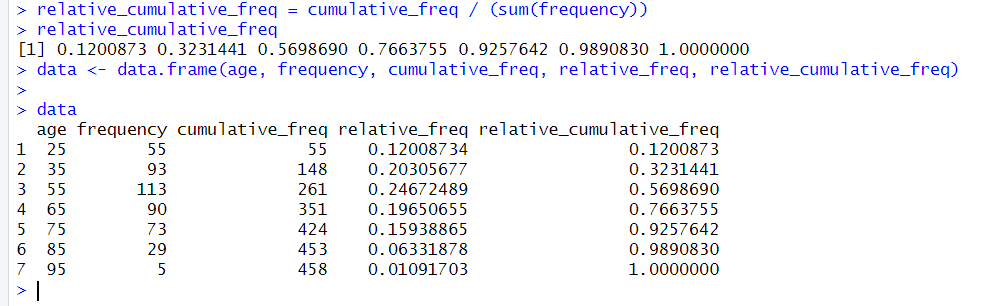
1. Add a column cumulative frequency.



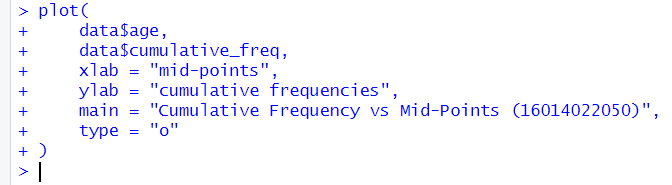
1. Add a column of relative frequency (frequency / total frequency).

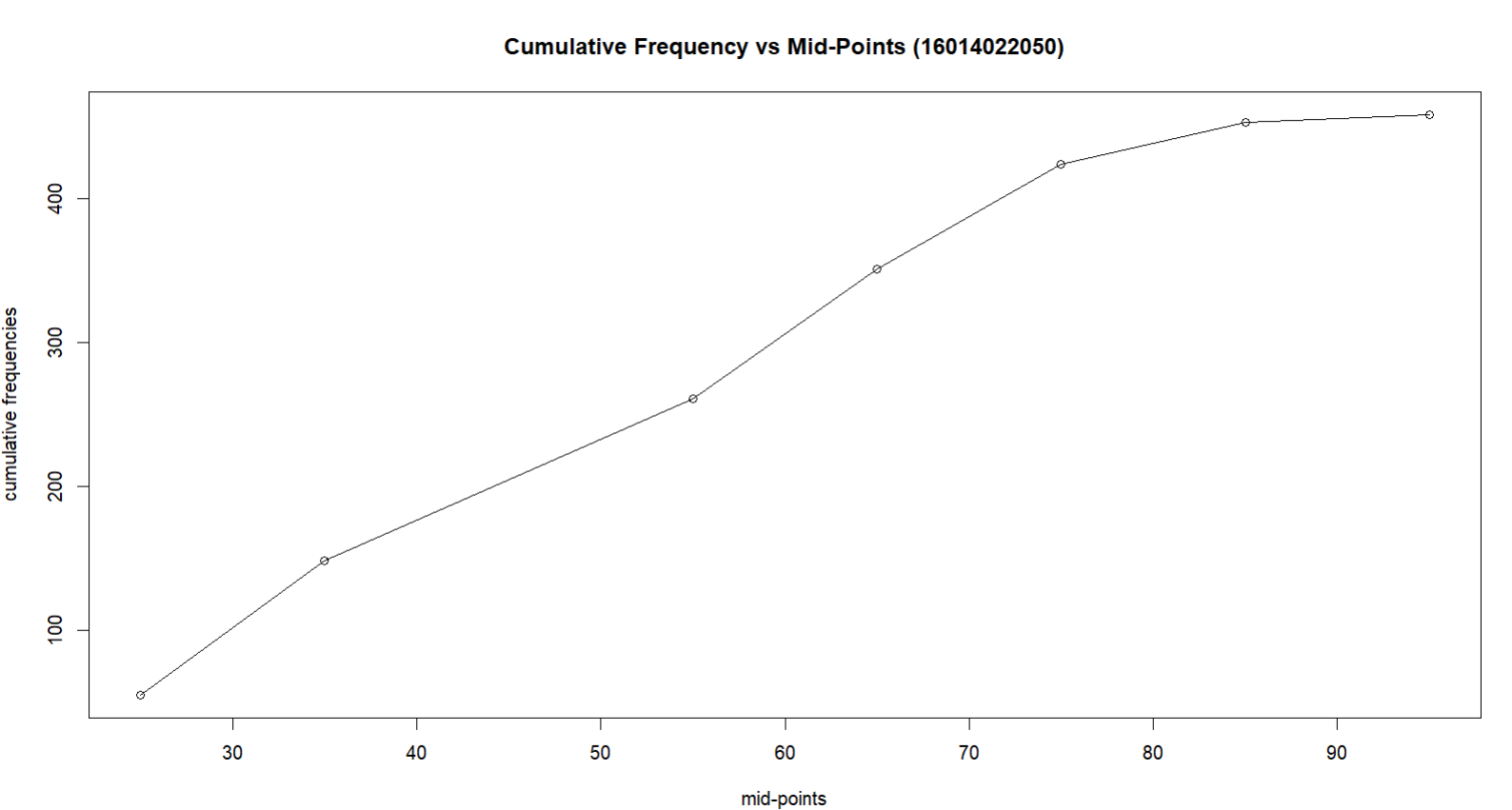


1. Add a column of relative cumulative frequency (cumulative frequency / total frequency).



1. Plot cumulative frequency vs mid points.

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion:**

Through the experimentation with R basics, including data types, operators, looping constructs, and graphics using packages like ggplot2, a foundational understanding has been developed. This hands-on exploration equips one with the essential skills to manipulate data, perform operations, and create visualizations, forming a solid groundwork for further proficiency in R programming.