

Day 3 Lab Manual

UNIVARIATE ANALYSIS IN R - MEASURES OF CENTRAL TENDENCY

Exercise:

I. ARITHMETIC MEAN

a) Write suitable R code to compute the average of the following values.

12,7,3,4.2,18,2,54,-21,8,-5

b) Compute the mean after applying the trim option and removing 3 values from each end.

c) Compute the mean of the following vector .

(12,7,3,4.2,18,2,54,-21,8,-5,NA)

#If there are missing values, then the mean function returns NA.

Find mean dropping NA values.

#To drop the missing values from the calculation use na.rm = TRUE

The screenshot shows a web browser window with the URL `programiz.com/online-compiler/`. The page features the Programiz logo and a banner for a "Top-rated Data Engineering Program from IIT-M Incubated Company GUVI. for better future" with a "LEARN MORE" button. Below the banner is the "R Online Compiler" interface. On the left, there is a sidebar with icons for various programming languages (Python, JavaScript, PHP, etc.). The main area is divided into two panes: "main.r" on the left and "Output" on the right. The "main.r" pane contains the following R code:

```
1 # Original vector
2 values <- c(12, 7, 3, 4.2, 18, 2, 54, -21, 8, -5)
3
4 # Compute the mean of the original vector
5 mean_values <- mean(values)
6 print(mean_values)
7
8 # Compute the mean after applying the trim option and removing 3 values from each
   end
9 trimmed_values <- trimmean(values, trim = 3/length(values))
10 print(trimmed_values)
11
12 # Compute the mean of the vector with missing values
13 values_with_na <- c(12, 7, 3, 4.2, 18, 2, 54, -21, 8, -5, NA)
14 mean_values_na <- mean(values_with_na, na.rm = TRUE)
15 print(mean_values_na)
```

The "Output" pane displays the results of the code execution:

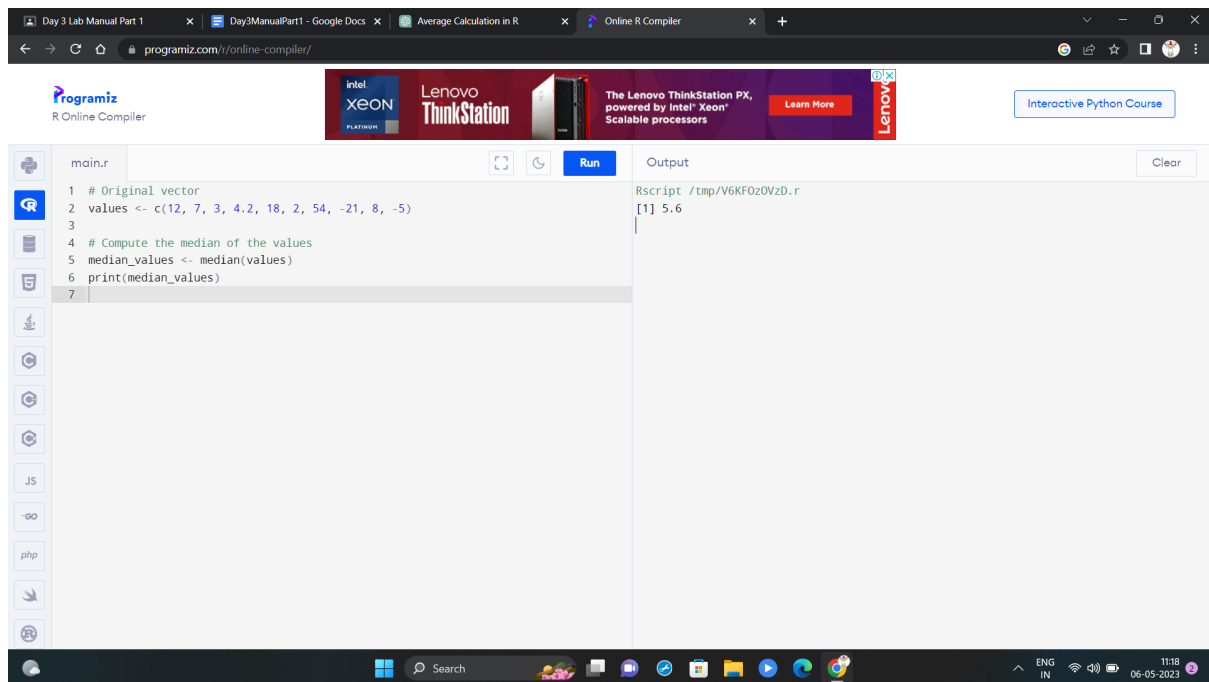
```
[1] 8.72
[1] 6.818182
[1] 7.272727
```

The bottom of the screenshot shows the Windows taskbar with the search bar and several application icons. The system clock in the bottom right corner indicates the time is 11:17 on 06-05-2023.

II.MEDIAN

Write suitable R code to compute the median of the following values.

12,7,3,4.2,18,2,54,-21,8,-5



The screenshot shows a web browser with the URL `programiz.com/online-compiler/`. The page features a header with logos for Intel Xeon, Lenovo ThinkStation, and an advertisement for the Lenovo ThinkStation PX. Below the header, there is a sidebar with icons for various programming languages (Python, JavaScript, PHP, etc.) and a main editor area. The editor contains the following R code:

```
main.r
1 # Original vector
2 values <- c(12, 7, 3, 4.2, 18, 2, 54, -21, 8, -5)
3
4 # Compute the median of the values
5 median_values <- median(values)
6 print(median_values)
7
```

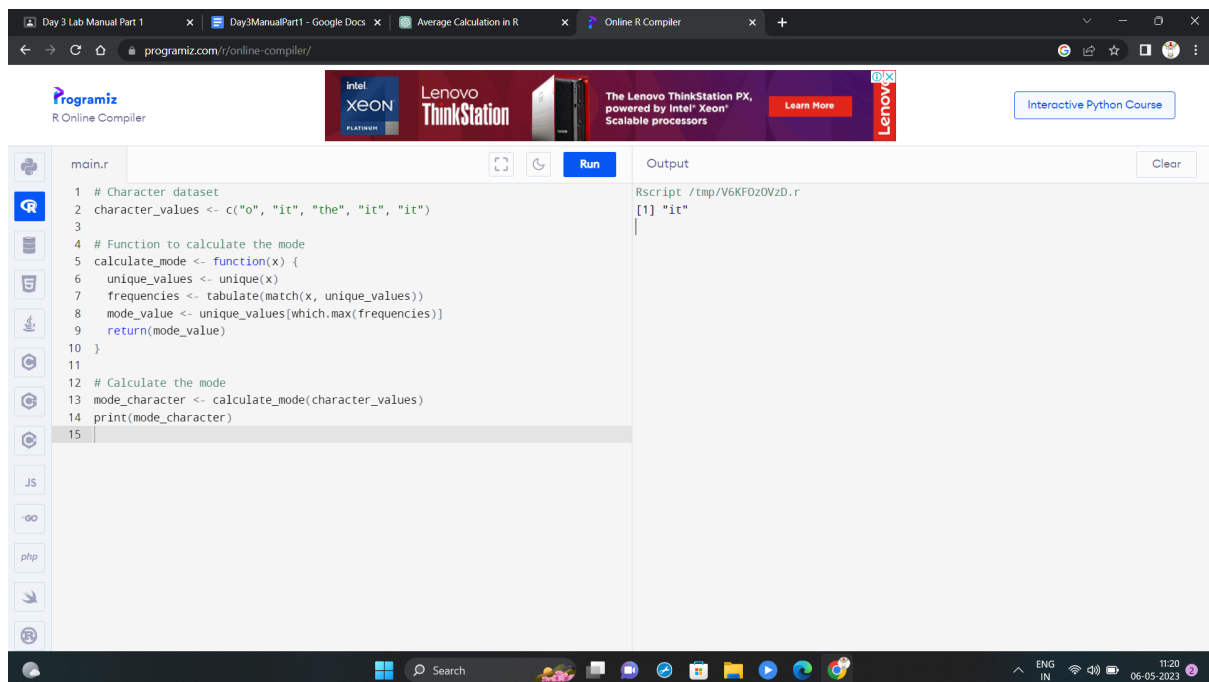
The output of the code is displayed on the right side of the editor:

```
Rscript /tmp/V6KF0z0VzD.r
[1] 5.6
```

III. MODE

Calculate the mode for the following numeric as well as character data set in R.

(2,1,2,3,1,2,3,4,1,5,5,3,2,3) , ("o", "it", "the", "it", "it")



The screenshot shows the same online R compiler interface. The editor contains the following R code:

```
main.r
1 # Character dataset
2 character_values <- c("o", "it", "the", "it", "it")
3
4 # Function to calculate the mode
5 calculate_mode <- function(x) {
6   unique_values <- unique(x)
7   frequencies <- tabulate(match(x, unique_values))
8   mode_value <- unique_values[which.max(frequencies)]
9   return(mode_value)
10 }
11
12 # Calculate the mode
13 mode_character <- calculate_mode(character_values)
14 print(mode_character)
15
```

The output of the code is displayed on the right side of the editor:

```
Rscript /tmp/V6KF0z0VzD.r
[1] "it"
```

UNIVARIATE ANALYSIS IN R - MEASURES OF DISPERSION

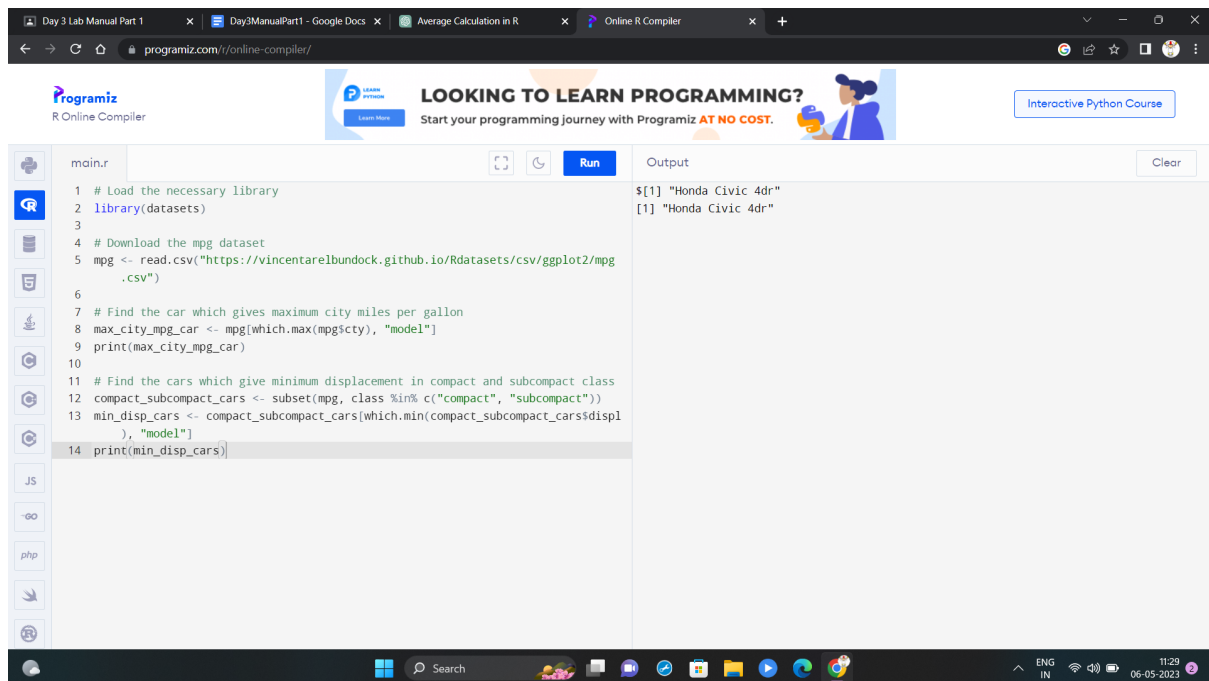
Exercise: 4

Download mpg dataset which contains Fuel economy data from 1999 and 2008 for 38 popular models of car from the URL given below.

<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

Answer the following queries

- i) Find the car which gives maximum city miles per gallon
- ii) Find the cars which gives minimum disp in compact and subcompact class



The screenshot shows a web browser window with the URL `programiz.com/online-compiler/`. The page features a header with the Programiz logo and a banner that says "LOOKING TO LEARN PROGRAMMING? Start your programming journey with Programiz AT NO COST." There is a button for an "Interactive Python Course". The main area is divided into two panels. The left panel, titled "main.r", contains the following R code:

```
1 # Load the necessary library
2 library(datasets)
3
4 # Download the mpg dataset
5 mpg <- read.csv("https://vincentarelbundock.github.io/Rdatasets/csv/ggplot2/mpg.csv")
6
7 # Find the car which gives maximum city miles per gallon
8 max_city_mpg_car <- mpg[which.max(mpg$city), "model"]
9 print(max_city_mpg_car)
10
11 # Find the cars which give minimum displacement in compact and subcompact class
12 compact_subcompact_cars <- subset(mpg, class %in% c("compact", "subcompact"))
13 min_disp_cars <- compact_subcompact_cars[which.min(compact_subcompact_cars$displ), "model"]
14 print(min_disp_cars)
```

The right panel, titled "Output", shows the results of the code execution:

```
$[1] "Honda Civic 4dr"
[1] "Honda Civic 4dr"
```

The bottom of the browser window shows the Windows taskbar with the search bar and various application icons. The system clock indicates the time is 11:29 on 06-05-2023.

Exercise: 5

Use the same dataset as used in Exercise 4 and perform the following queries

- i) Find the standard deviation of city miles per gallon
- ii) Find the variance of highway miles per gallon

Day 3 Lab Manual Part 1 | Day3ManualPart1 - Google Docs | Average Calculation in R | Online R Compiler

programiz.com/online-compiler/

Programiz
R Online Compiler

LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz **AT NO COST.**

Interactive Python Course

```

main.r
1 # Load the necessary library
2 library(datasets)
3
4 # Download the mpg dataset
5 mpg <- read.csv("https://vincentarelbundock.github.io/Rdatasets/csv/ggplot2/mpg
  .csv")
6
7 # Find the standard deviation of city miles per gallon
8 sd_city_mpg <- sd(mpg$cty)
9 print(sd_city_mpg)
10
11 # Find the variance of highway miles per gallon
12 var_highway_mpg <- var(mpg$hwyl)
13 print(var_highway_mpg)
14

```

Output

```

$ [1] 5.978895
[1] 34.59105

```

JS
GO
php

ENG IN 11:39 06-05-2023

Exercise 6

Use the same dataset and perform the following queries

- Find the range of the disp in the data set mpg
- Find the Quartile of the disp in the data set mpg
- Find the IQR of the disp column in the data set mpg

Day 3 Lab Manual Part 1 | Day3ManualPart1 - Google Docs | Average Calculation in R | Online R Compiler

programiz.com/online-compiler/

Programiz
R Online Compiler

LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz **AT NO COST.**

Interactive Python Course

```

main.r
1 # Load the necessary library
2 library(datasets)
3
4 # Download the mpg dataset
5 mpg <- read.csv("https://vincentarelbundock.github.io/Rdatasets/csv/ggplot2/mpg
  .csv")
6
7 # Find the range of the disp column
8 range_disp <- range(mpg$displ)
9 print(range_disp)
10
11 # Find the quartiles of the disp column
12 quartiles_disp <- quantile(mpg$displ, probs = c(0.25, 0.5, 0.75))
13 print(quartiles_disp)
14
15 # Find the IQR of the disp column
16 iqr_disp <- IQR(mpg$displ)
17 print(iqr_disp)
18

```

Output

```

$ [1] 1.6 7.0
25% 50% 75%
2.4 3.3 4.6
[1] 2.2

```

JS
GO
php

ENG IN 11:43 06-05-2023

Exercise 7

#Install Library

```
library(e1071)
```

- a. Find the skewness of city miles per mileage in the data set mpg ?

Use qplot function and display the graph for the city miles per mileage column

- b. Find the kurtosis of city miles per mileage in the data set mpg

The screenshot shows the Programiz R Online Compiler interface. The code editor on the left contains the following R code:

```
main.r
1 install.packages("e1071")
2 library(e1071)
3 library(ggplot2)
4 mpg <- read.csv("https://vincentarelbundock.github.io/Rdatasets/csv/ggplot2/mpg.csv")
5 skewness_city_mpg <- skewness(mpg$cty)
6 print(skewness_city_mpg)
7 qplot(mpg$cty, geom = "histogram", bins = 20, xlab = "City Miles per Gallon",
8       ylab = "Frequency")
9 kurtosis_city_mpg <- kurtosis(mpg$cty)
10 print(kurtosis_city_mpg)
```

The output window on the right displays the results of the code execution:

```
$ [1] -0.4706988
[1] 2.907606
```

The browser's address bar shows the URL <https://programiz.com/online-compiler/>. The Windows taskbar at the bottom indicates the date and time as 06-05-2023, 11:56.