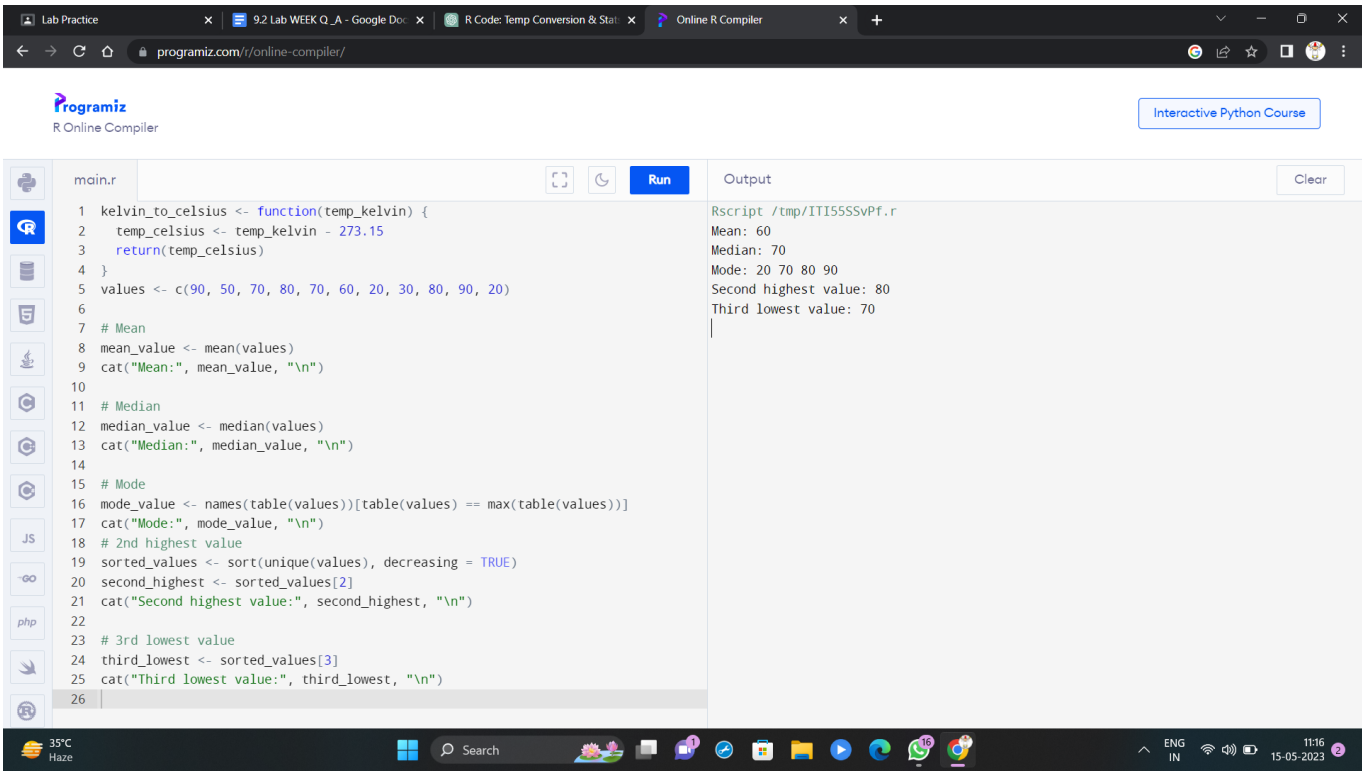
Name: KRISHVANTH KUMAR E
REG NO:192125063

Set-I

1. .(i) Write a function called kelvin_to_celsius() that takes a temperature in Kelvin and returns that temperature in Celsius (**Hint:** To convert from Kelvin to Celsius you subtract 273.15) (ii) Write suitable R code to compute the mean, median ,mode of the following values c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

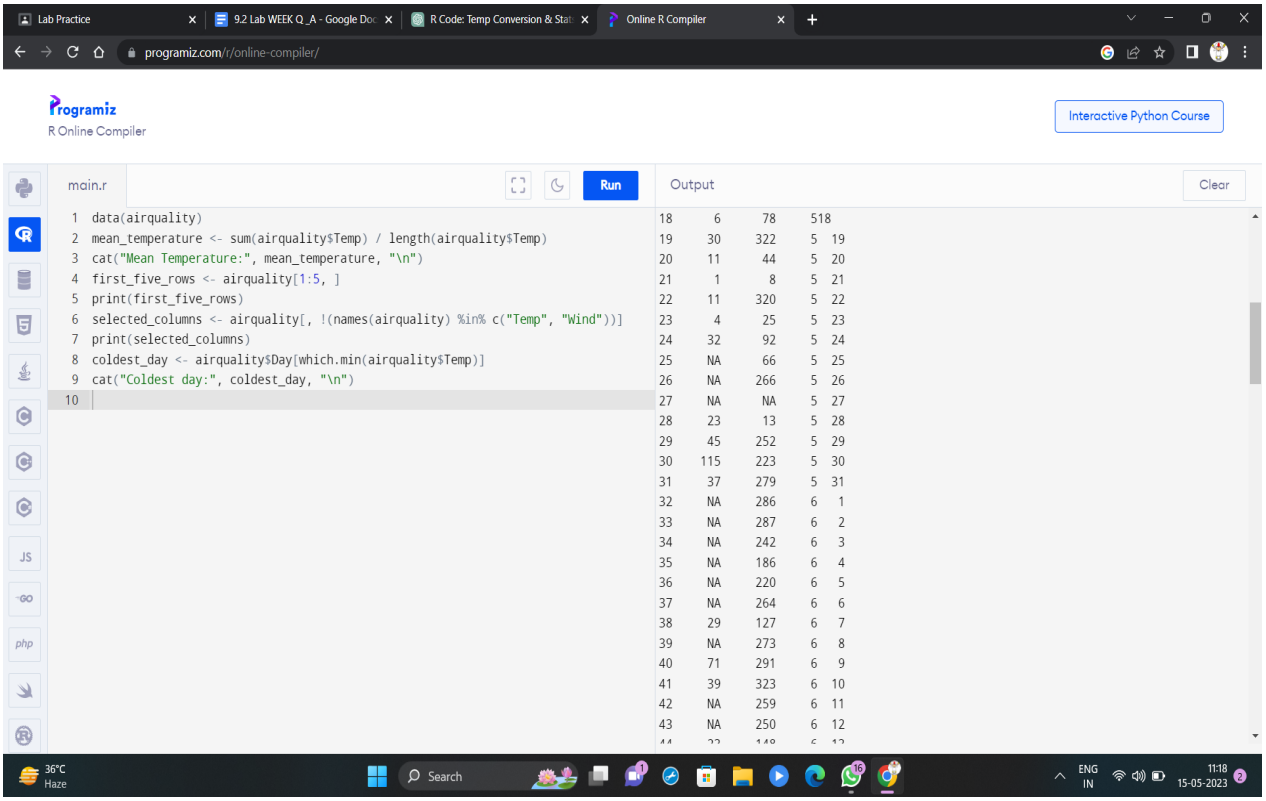
(iii) Write R code to find 2nd highest and 3rd Lowest value of above problem.

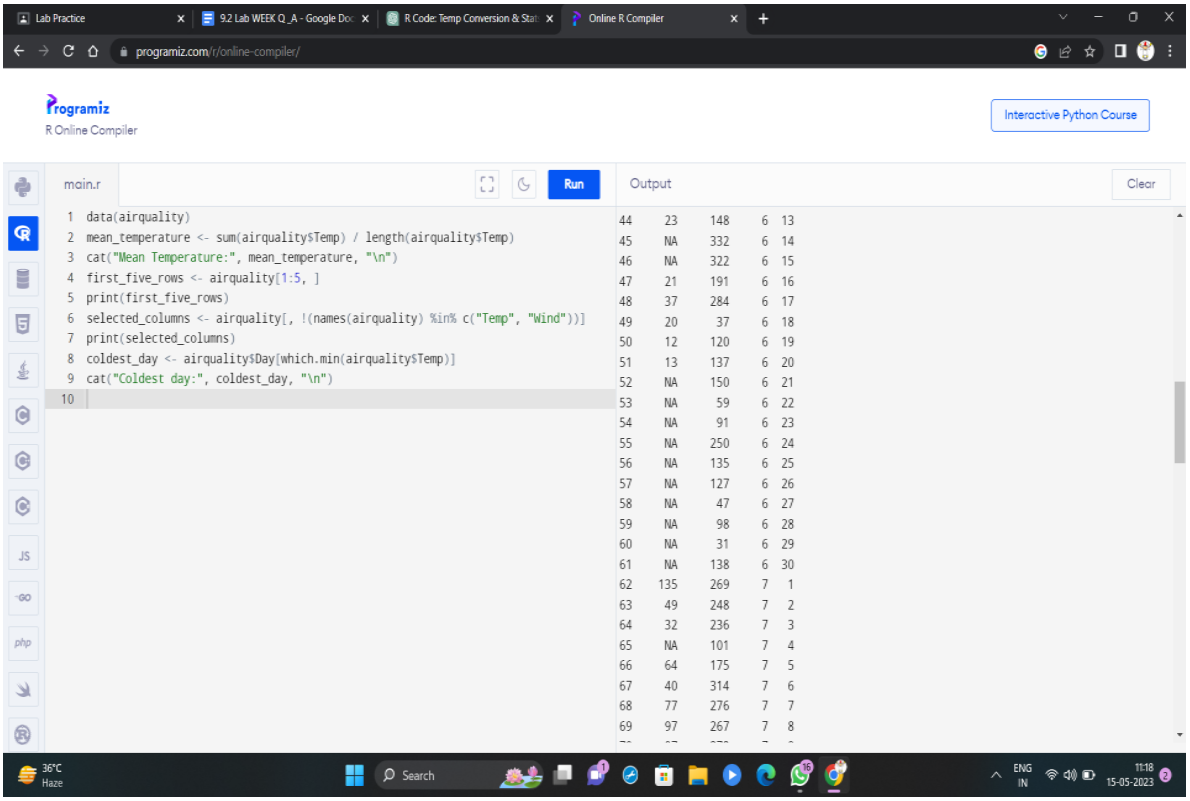


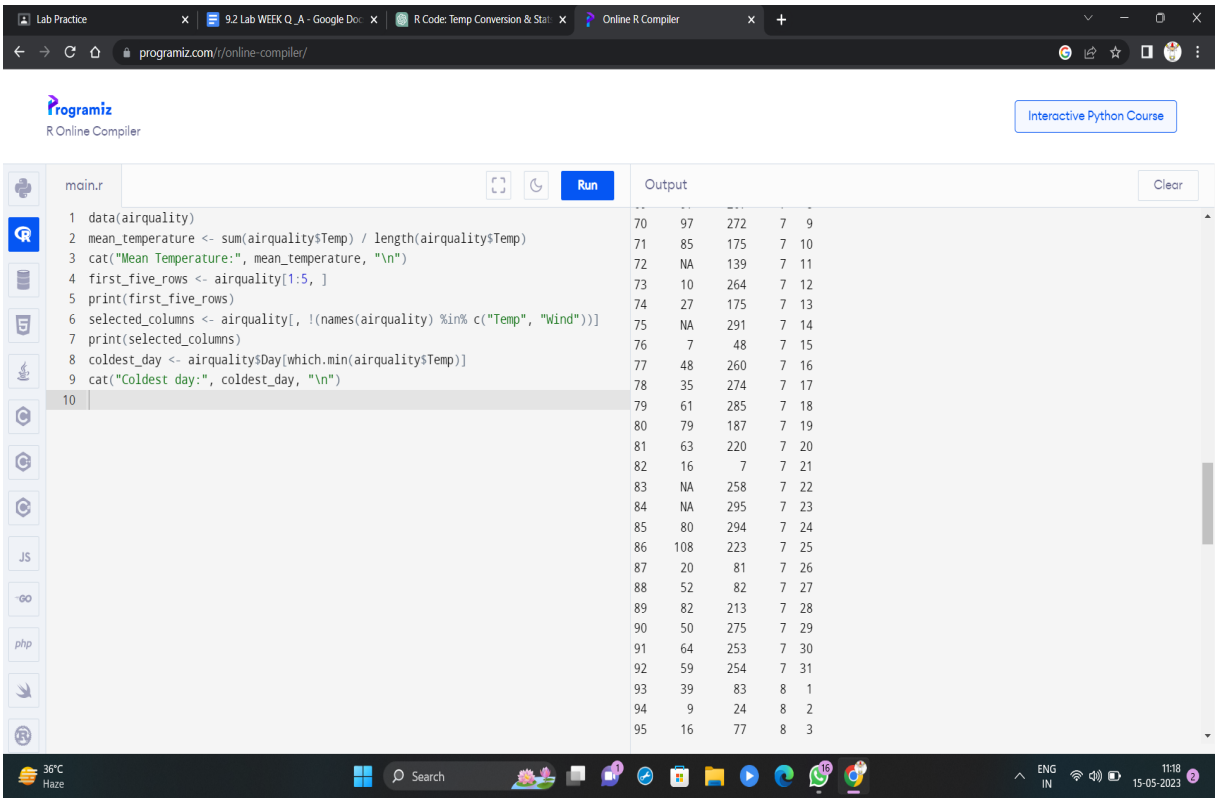
2. Explore the airquality dataset. It contains daily air quality measurements from New York during a period of five months:
   - Ozone: mean ozone concentration (ppb),
   - Solar.R: solar radiation (Langley),

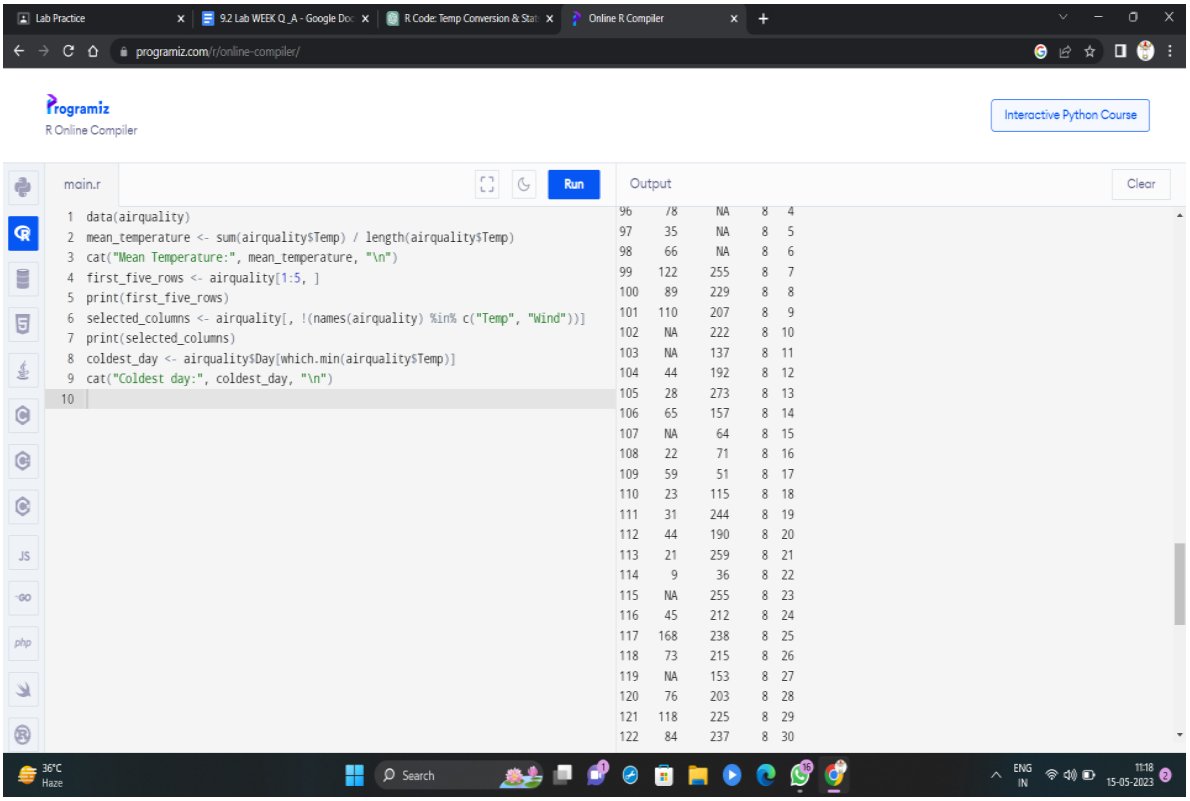
• Wind: average wind speed (mph),
• Temp: maximum daily temperature in degrees Fahrenheit,
• Month: numeric month (May=5, June=6, and so on),
• Day: numeric day of the month (1-31).

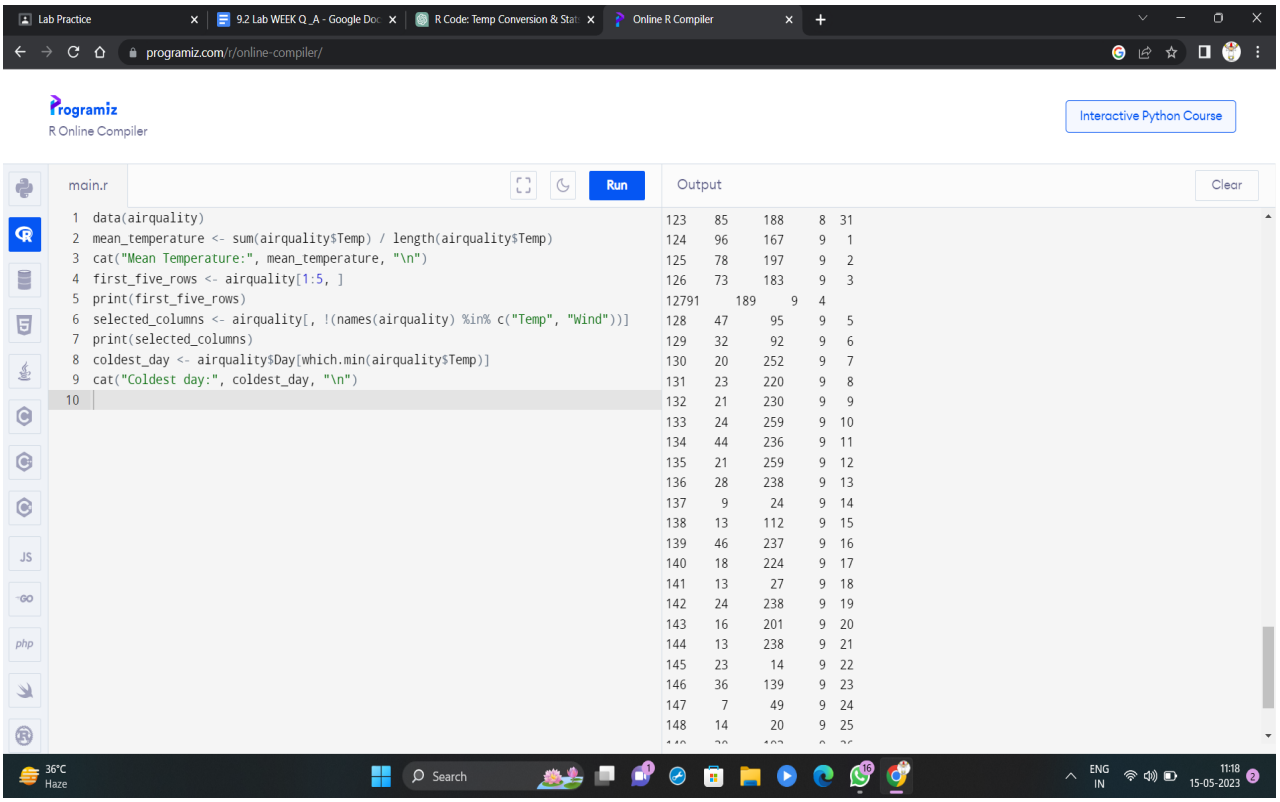
i. Compute the mean temperature(don't use build in function)
ii. Extract the first five rows from airquality.
iii. Extract all columns from airquality *except* Temp and Wind
iv. Which was the coldest day during the period?
v. How many days was the wind speed greater than 17 mph?

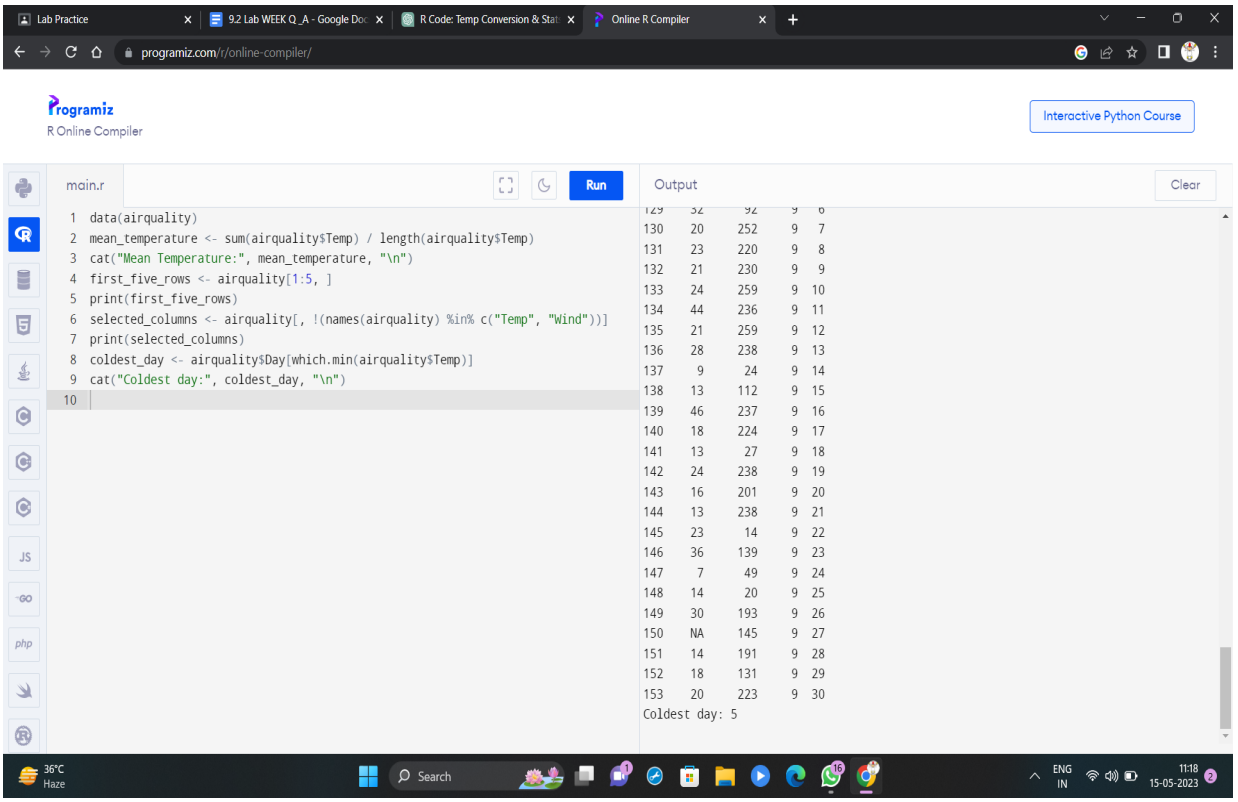



Lab Practice
9.2 Lab WEEK Q_A - Google Doc
R Code: Temp Conversion & Stat
Online R Compiler
programiz.com/r/online-compiler/
Programiz
R Online Compiler
Interactive Python Course
main.r
Run
Output
Clear
1 data(airquality)
2 mean_temperature <- sum(airquality$Temp) / length(airquality$Temp)
3 cat("Mean Temperature:", mean_temperature, "\n")
4 first_five_rows <- airquality[1:5,]
5 print(first_five_rows)
6 selected_columns <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]
7 print(selected_columns)
8 coldest_day <- airquality$Day[which.min(airquality$Temp)]
9 cat("Coldest day:", coldest_day, "\n")
10
44 23 148 6 13
45 NA 332 6 14
46 NA 322 6 15
47 21 191 6 16
48 37 284 6 17
49 20 37 6 18
50 12 120 6 19
51 13 137 6 20
52 NA 150 6 21
53 NA 59 6 22
54 NA 91 6 23
55 NA 250 6 24
56 NA 135 6 25
57 NA 127 6 26
58 NA 47 6 27
59 NA 98 6 28
60 NA 31 6 29
61 NA 138 6 30
62 135 269 7 1
63 49 248 7 2
64 32 236 7 3
65 NA 101 7 4
66 64 175 7 5
67 40 314 7 6
68 77 276 7 7
69 97 267 7 8
36°C
Haze
Search
ENG
IN
11:18
15-05-2023


Lab Practice
9.2 Lab WEEK Q_A - Google Doc
R Code: Temp Conversion & Stat
Online R Compiler
programiz.com/r/online-compiler/
Programiz
R Online Compiler
Interactive Python Course
main.r
Run
Output
Clear
1 data(airquality)
2 mean_temperature <- sum(airquality$Temp) / length(airquality$Temp)
3 cat("Mean Temperature:", mean_temperature, "\n")
4 first_five_rows <- airquality[1:5,]
5 print(first_five_rows)
6 selected_columns <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]
7 print(selected_columns)
8 coldest_day <- airquality$Day[which.min(airquality$Temp)]
9 cat("Coldest day:", coldest_day, "\n")
10
70 97 272 7 9
71 85 175 7 10
72 NA 139 7 11
73 10 264 7 12
74 27 175 7 13
75 NA 291 7 14
76 7 48 7 15
77 48 260 7 16
78 35 274 7 17
79 61 285 7 18
80 79 187 7 19
81 63 220 7 20
82 16 7 7 21
83 NA 258 7 22
84 NA 295 7 23
85 80 294 7 24
86 108 223 7 25
87 20 81 7 26
88 52 82 7 27
89 82 213 7 28
90 50 275 7 29
91 64 253 7 30
92 59 254 7 31
93 39 83 8 1
94 9 24 8 2
95 16 77 8 3
36°C
Haze
Search
ENG
IN
11:18
15-05-2023


Lab Practice
9.2 Lab WEEK Q_A - Google Doc
R Code: Temp Conversion & Stat
Online R Compiler
programiz.com/r/online-compiler/
Programiz
R Online Compiler
Interactive Python Course
main.r
Run
Output
Clear
1 data(airquality)
2 mean_temperature <- sum(airquality$Temp) / length(airquality$Temp)
3 cat("Mean Temperature:", mean_temperature, "\n")
4 first_five_rows <- airquality[1:5,]
5 print(first_five_rows)
6 selected_columns <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]
7 print(selected_columns)
8 coldest_day <- airquality$Day[which.min(airquality$Temp)]
9 cat("Coldest day:", coldest_day, "\n")
10
96 78 NA 8 4
97 35 NA 8 5
98 66 NA 8 6
99 122 255 8 7
100 89 229 8 8
101 110 207 8 9
102 NA 222 8 10
103 NA 137 8 11
104 44 192 8 12
105 28 273 8 13
106 65 157 8 14
107 NA 64 8 15
108 22 71 8 16
109 59 51 8 17
110 23 115 8 18
111 31 244 8 19
112 44 190 8 20
113 21 259 8 21
114 9 36 8 22
115 NA 255 8 23
116 45 212 8 24
117 168 238 8 25
118 73 215 8 26
119 NA 153 8 27
120 76 203 8 28
121 118 225 8 29
122 84 237 8 30
36°C
Haze
Search
ENG
IN
11:18
15-05-2023


Lab Practice
9.2 Lab WEEK Q_A - Google Doc
R Code: Temp Conversion & Stat
Online R Compiler
programiz.com/r/online-compiler/
Programiz
R Online Compiler
Interactive Python Course
main.r
Run
Output
Clear
1 data(airquality)
2 mean_temperature <- sum(airquality$Temp) / length(airquality$Temp)
3 cat("Mean Temperature:", mean_temperature, "\n")
4 first_five_rows <- airquality[1:5,]
5 print(first_five_rows)
6 selected_columns <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]
7 print(selected_columns)
8 coldest_day <- airquality$Day[which.min(airquality$Temp)]
9 cat("Coldest day:", coldest_day, "\n")
10
123 85 188 8 31
124 96 167 9 1
125 78 197 9 2
126 73 183 9 3
12791 189 9 4
128 47 95 9 5
129 32 92 9 6
130 20 252 9 7
131 23 220 9 8
132 21 230 9 9
133 24 259 9 10
134 44 236 9 11
135 21 259 9 12
136 28 238 9 13
137 9 24 9 14
138 13 112 9 15
139 46 237 9 16
140 18 224 9 17
141 13 27 9 18
142 24 238 9 19
143 16 201 9 20
144 13 238 9 21
145 23 14 9 22
146 36 139 9 23
147 7 49 9 24
148 14 20 9 25
36°C
Haze
Search
ENG
IN
11:18
15-05-2023



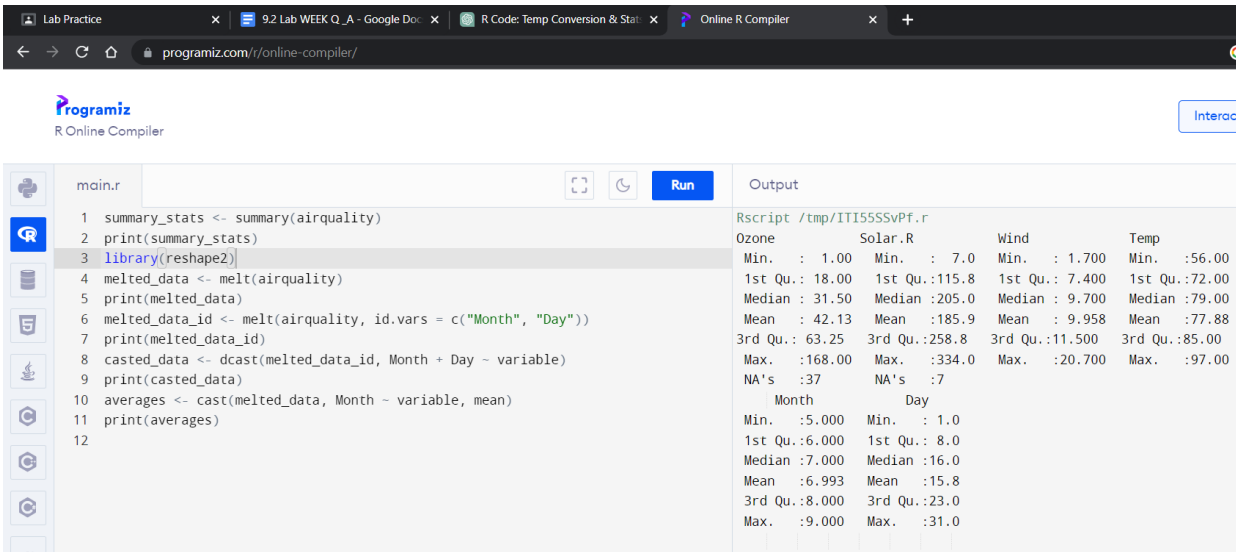
3. (i) Get the Summary Statistics of air quality dataset

(ii)Melt airquality data set and display as a long – format data?

(iii)Melt airquality data and specify month and day to be “ID variables”?

(iv)Cast the molten airquality data set with respect to month and date features

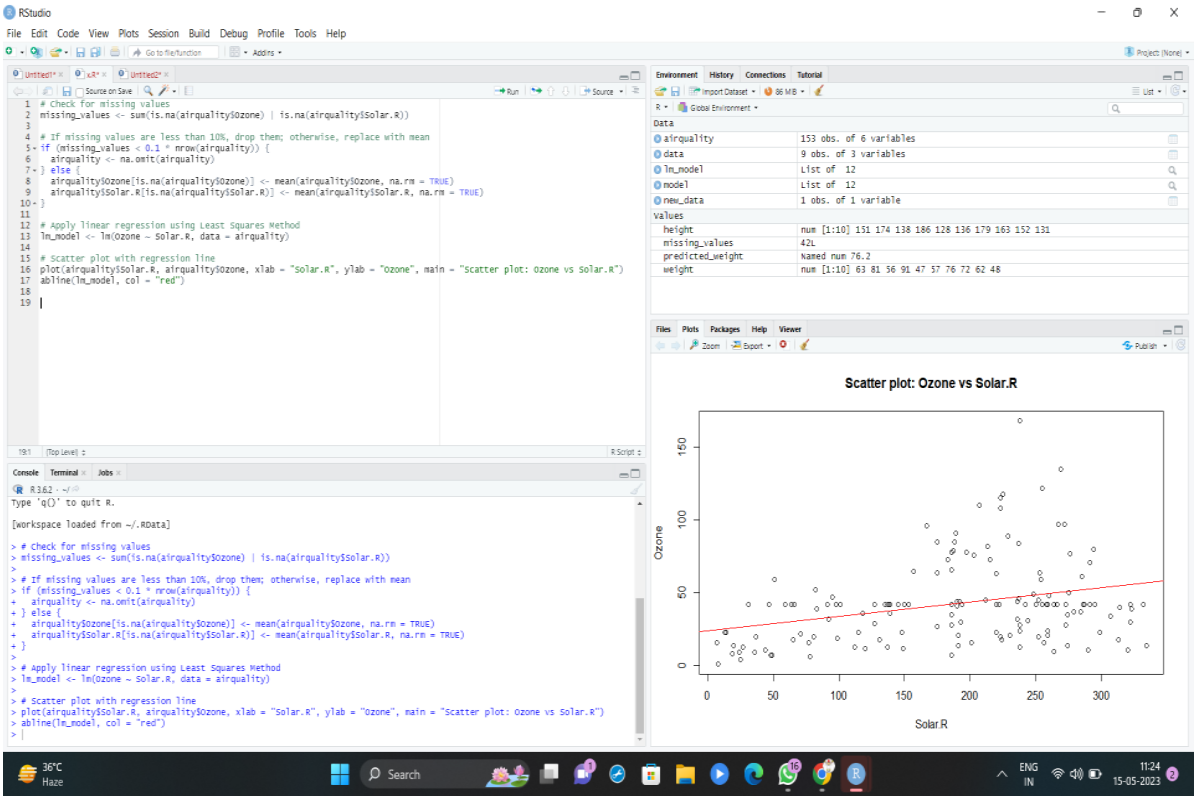
(v) Use cast function appropriately and compute the average of Ozone, Solar.R , Wind and temperature per month?



4.(i) Find any missing values(na) in features and drop the missing values if its less than 10% else replace that with mean of that feature.

(ii) Apply a linear regression algorithm using Least Squares Method on “Ozone” and “Solar.R”

(iii)Plot Scatter plot between Ozone and Solar and add regression line created by above model

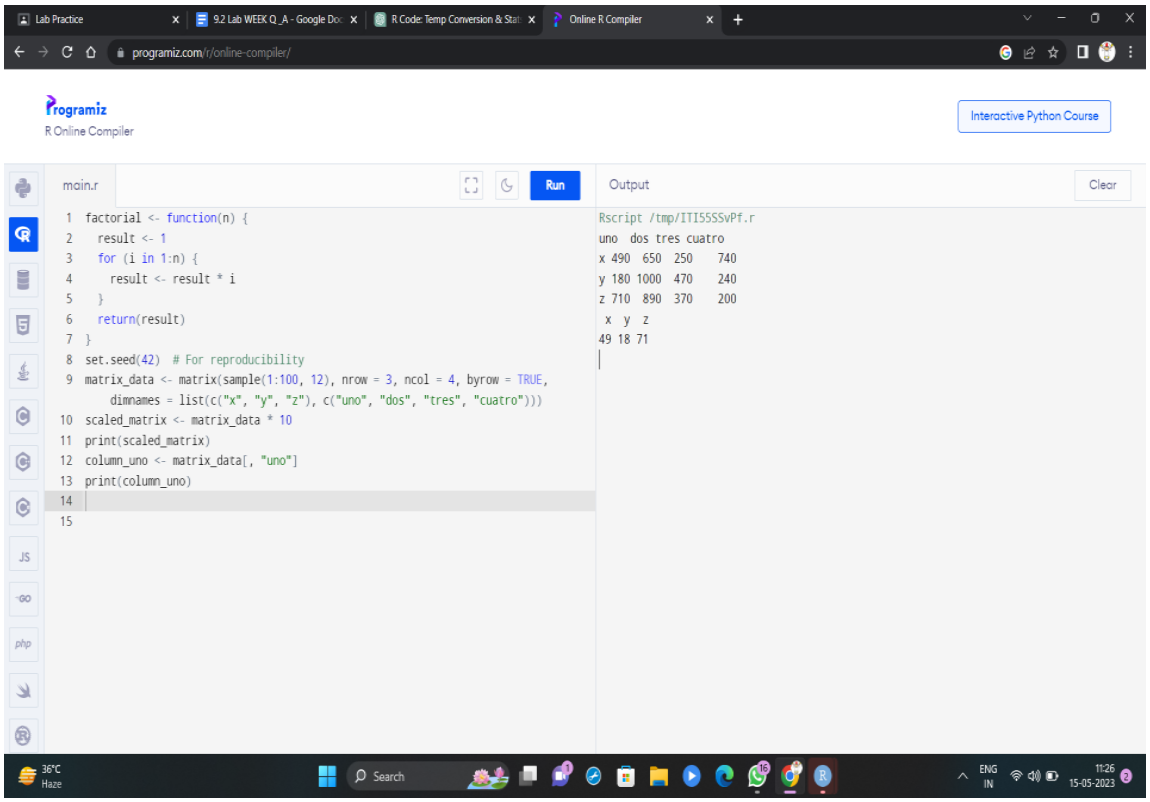


Set-II

1. (i)Write a function to find the factorial of a given number using “for” Loop

   (ii) Create a 3x4 matrix with 12 random numbers between 1-100; have the matrix be filled our row by-row, instead of column-by-column. Name the columns of the matrix *uno*, *dos*, *tres*, *cuatro*, and the rows *x*, *y*, *z*. Scale the matrix by 10 and save the result.

   (iii) Extract the column called “uno” as a vector from the original matrix and save the result



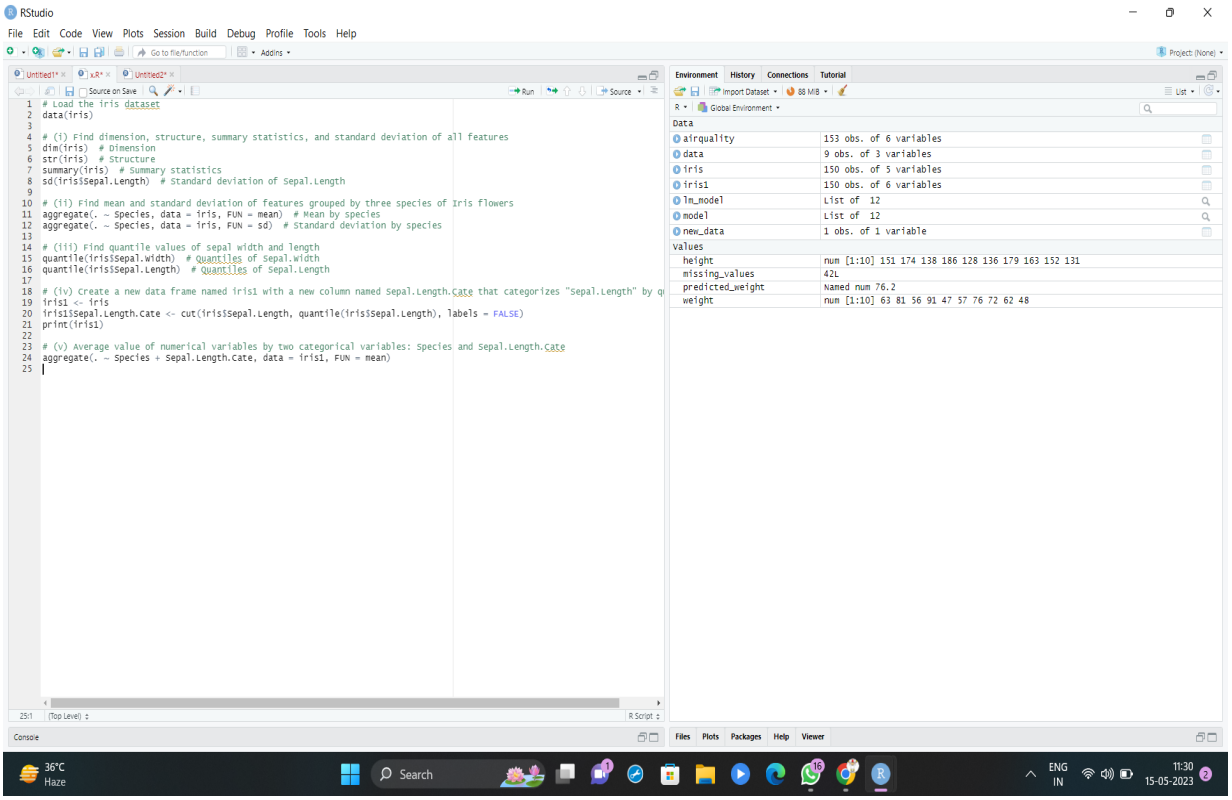
2. In 1936, Edgar Anderson collected data to quantify the geographic variations of iris flowers. The data set consists of 50 samples from each of the three sub-species (*iris setosa*, *iris virginica,* and *iris versicolor*).Four features were measured in centimeters (cm): the lengths and the widths of both sepals and petals

(i)Find dimension, Structure, Summary statistics, Standard Deviation of all features. (ii)Find mean and standard deviation of features groped by three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor)

(iii)Find quantile value of sepal width and length

(iv)create new data frame named iris1 which have a new column name **Sepal.Length.Cate** that categorizes "Sepal.Length" by quantile

(v) Average value of numerical varialbes by two categorical variables: Species and Sepal.Length.Cate.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1* xR* Untitled2*
Source on Save
Run
Source
# Load the iris dataset
data(iris)
# (i) Find dimension, structure, summary statistics, and standard deviation of all features
dim(iris) # Dimension
str(iris) # Structure
summary(iris) # Summary statistics
sd(iris$Sepal.Length) # Standard deviation of Sepal.Length
(ii) Find mean and standard deviation of features grouped by three species of Iris flowers
aggregate(. ~ Species, data = iris, FUN = mean) # Mean by species
aggregate(. ~ Species, data = iris, FUN = sd) # Standard deviation by species
(iii) Find quantile values of sepal width and length
quantile(iris$Sepal.Width) # Quantiles of Sepal.Width
quantile(iris$Sepal.Length) # Quantiles of Sepal.Length
# (iv) Create a new data frame named iris1 with a new column named Sepal.Length.Cate that categorizes "Sepal.Length" by q
iris1 <- iris
iris1$Sepal.Length.Cate <- cut(iris$Sepal.Length, quantile(iris$Sepal.Length), labels = FALSE)
print(iris1)
# (v) Average value of numerical variables by two categorical variables: Species and Sepal.Length.Cate
aggregate(. ~ Species + Sepal.Length.Cate, data = iris1, FUN = mean)
25:1 (Top Level) R Script
Console
Environment History Connections Tutorial
Import Dataset 88 MiB
List
Global Environment
Data
airquality 153 obs. of 6 variables
data 9 obs. of 3 variables
iris 150 obs. of 5 variables
iris1 150 obs. of 6 variables
lm_model List of 12
model List of 12
new_data 1 obs. of 1 variable
Values
height num [1:10] 151 174 138 186 128 136 179 163 152 131
missing_values 42L
predicted_weight Named num 76.2
weight num [1:10] 63 81 56 91 47 57 76 72 62 48
Files Plots Packages Help Viewer
36°C Haze
Search
ENG IN
11:30 15-05-2023

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

```
[1] 150   5
> str(iris)  # Structure
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> summary(iris)  # Summary statistics
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> sd(iris$Sepal.Length)  # Standard deviation of Sepal.Length
[1] 0.8280661
>
> # (ii) Find mean and standard deviation of features grouped by three species of Iris flowers
> aggregate(. ~ Species, data = iris, FUN = mean)  # Mean by species
     Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa        5.006       3.428        1.462       0.246
2 versicolor        5.936       2.770        4.260       1.326
3  virginica        6.588       2.974        5.552       2.026
> aggregate(. ~ Species, data = iris, FUN = sd)  # Standard deviation by species
     Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1     setosa    0.3524897   0.3790644   0.1736640   0.1053856
2 versicolor    0.5161711   0.3137983   0.4699110   0.1977527
3  virginica    0.6358796   0.3224966   0.5518947   0.2746501
>
> # (iii) Find quantile values of sepal width and length
> quantile(iris$Sepal.Width)  # Quantiles of Sepal.Width
  0%  25%  50%  75% 100%
 2.0  2.8  3.0  3.3  4.4
> quantile(iris$Sepal.Length)  # Quantiles of Sepal.Length
  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9
>
> # (iv) Create a new data frame named iris1 with a new column named Sepal.Length.Cate that categorizes "Sepal.Length" by quantile
> iris1 <- iris
> iris1$Sepal.Length.Cate <- cut(iris$Sepal.Length, quantile(iris$Sepal.Length), labels = FALSE)
> print(iris1)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species Sepal.Length.Cate
1            5.1         3.5          1.4         0.2     setosa                 1
2            4.9         3.0          1.4         0.2     setosa                 1
3            4.7         3.2          1.3         0.2     setosa                 1
4            4.6         3.1          1.5         0.2     setosa                 1
5            5.0         3.6          1.4         0.2     setosa                 1
6            5.4         3.9          1.7         0.4     setosa                 2
7            4.6         3.4          1.4         0.3     setosa                 1
8            5.0         3.4          1.5         0.2     setosa                 1
9            4.4         2.9          1.4         0.2     setosa                 1
10           4.9         3.1          1.5         0.1     setosa                 1
11           5.4         3.7          1.5         0.2     setosa                 2
```

Environment | History | Connections | Tutorial

Data	
airquality	153 obs. of 6 variables
data	9 obs. of 3 variables
iris	150 obs. of 5 variables
iris1	150 obs. of 6 variables
lm_model	List of 12
model	List of 12
new_data	1 obs. of 1 variable
values	
height	num [1:10] 151 174 138 186 128 136 179 163 152 131
missing_values	42L
predicted_weight	Named num 76.2
weight	num [1:10] 63 81 56 91 47 57 76 72 62 48

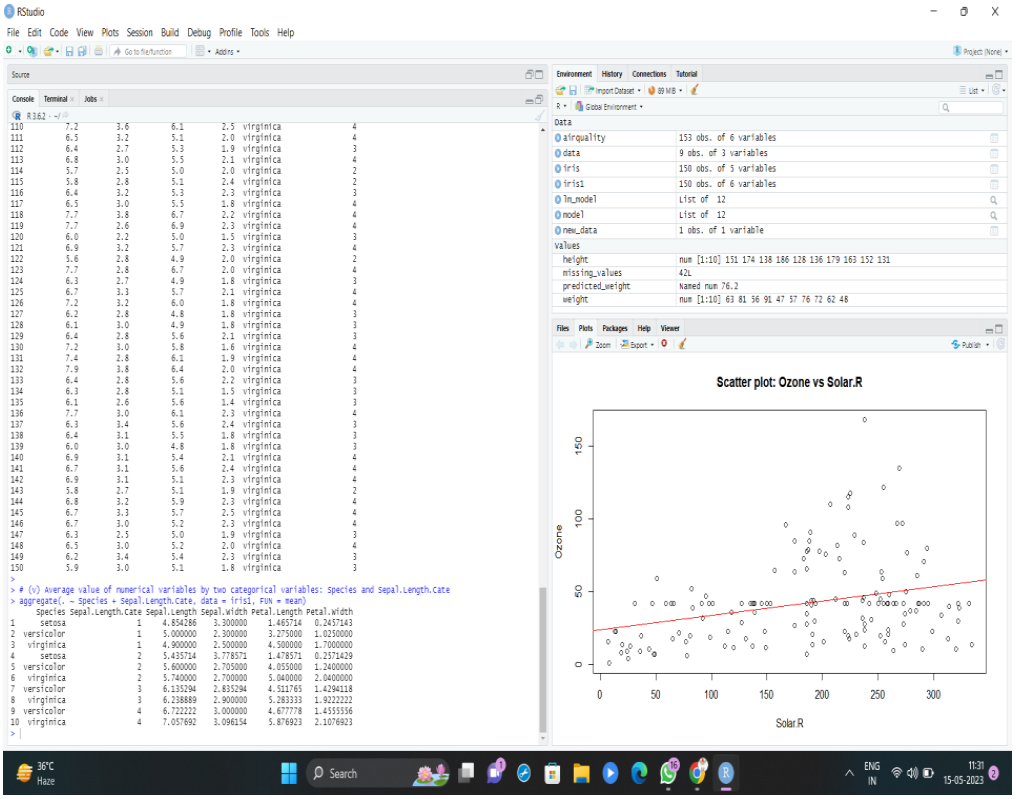
Files | Plots | Packages | Help | Viewer

36°C Haze

Search

ENG IN

11:31 15-05-2023



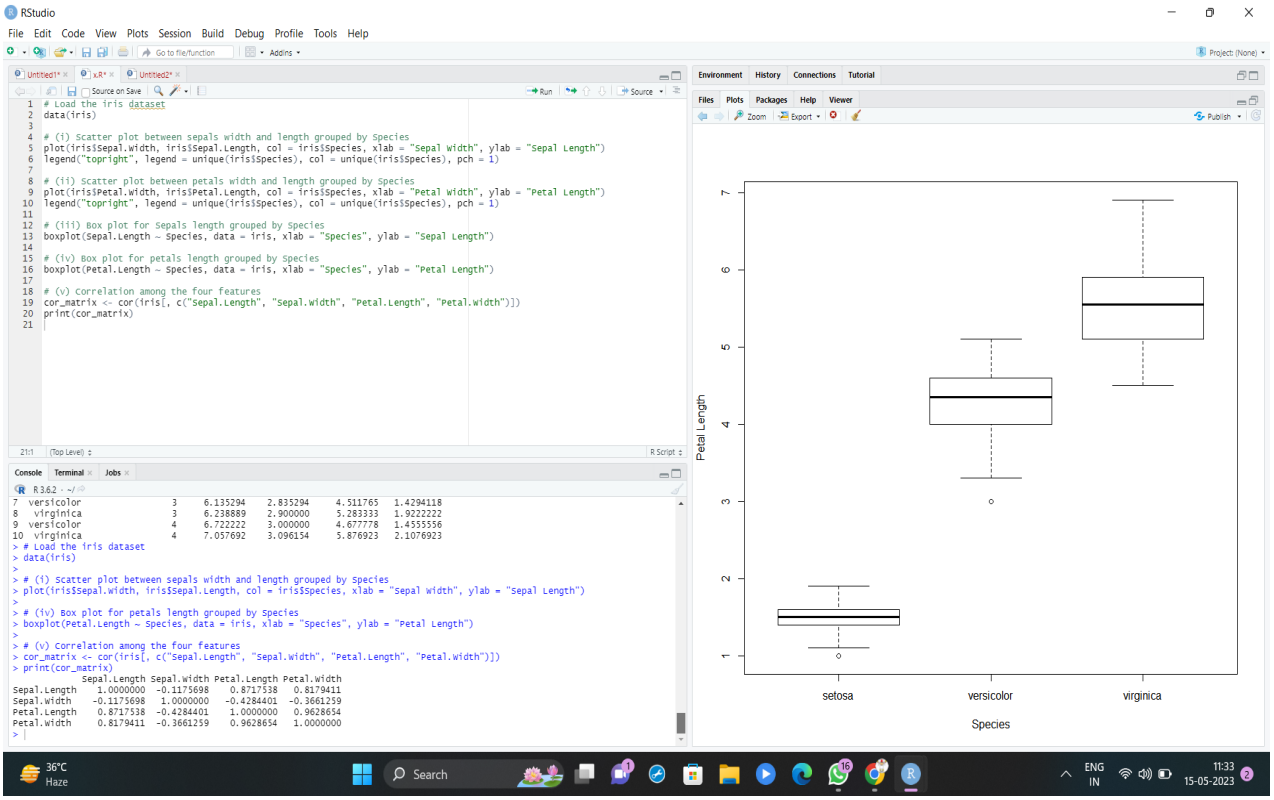
3. (i)Plot Scatter plot between sepals width and length grouped by Species

(ii) Plot Scatter plot between petals width and length grouped by Species

(iii)Draw the Box plot for Sepals length grouped by Species

(iv) Draw the Box plot for petals length grouped by Species

(v)Find the correlation among the four features

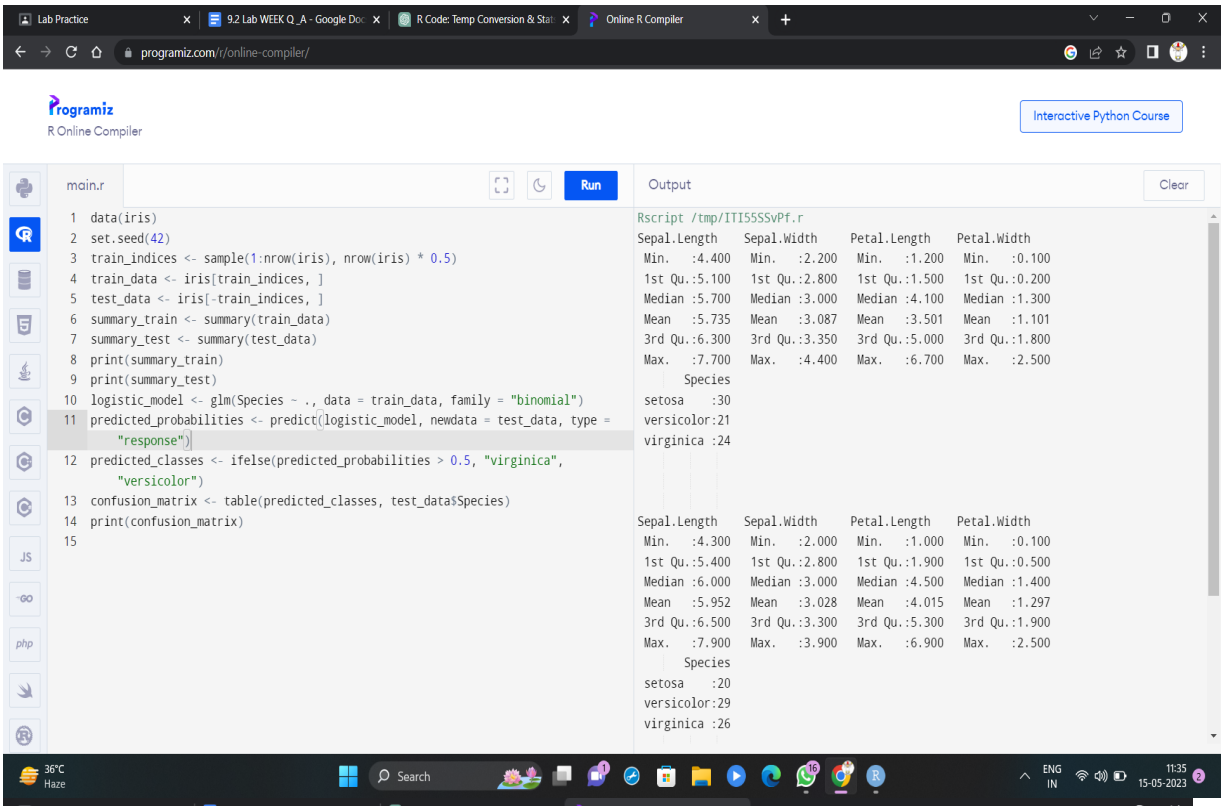


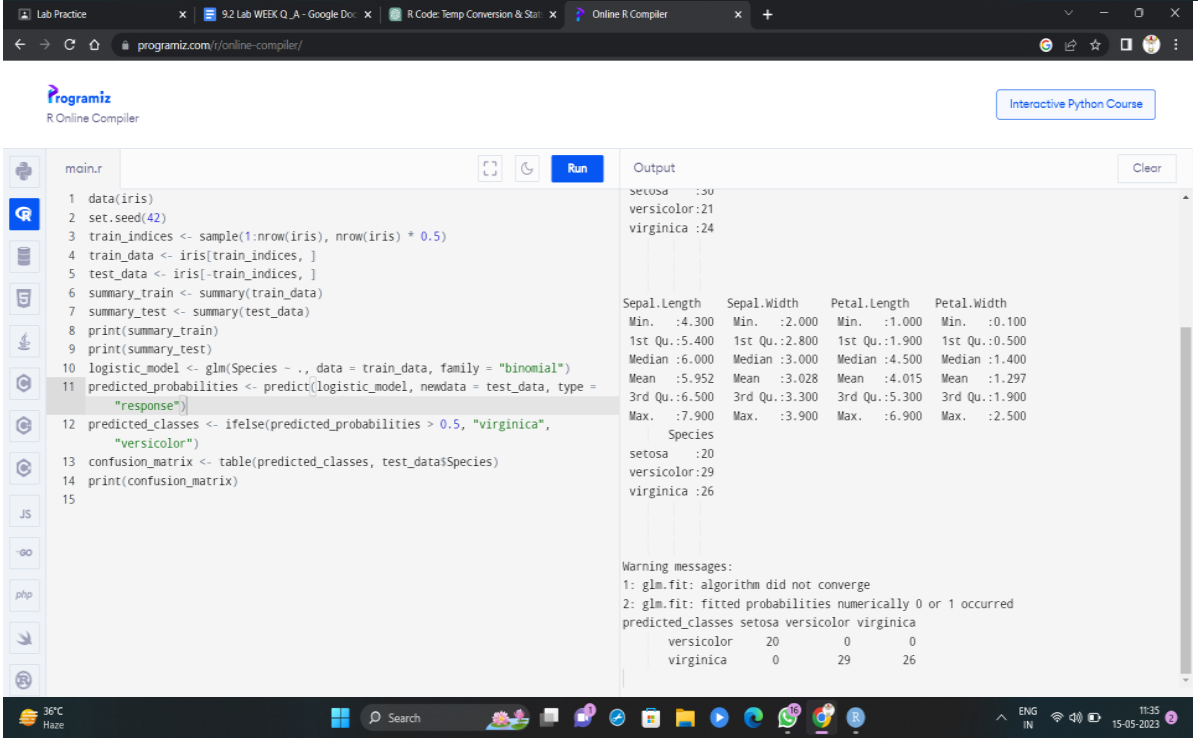
4.(i) Randomly Sample the iris dataset such as 50% data for training and 50% for test (ii)find summary statistics of above train and test dataset.

(iii)Create Logistics regression with train data

(iv)Predict the probability of the model using test data

(v)Create Confusion matrix for above test model


Lab Practice
9.2 Lab WEEK Q_A - Google Doc
R Code: Temp Conversion & Stat
Online R Compiler
programiz.com/r/online-compiler/
Programiz
R Online Compiler
Interactive Python Course
main.r
Run
Output
Clear
1 data(iris)
2 set.seed(42)
3 train_indices <- sample(1:nrow(iris), nrow(iris) * 0.5)
4 train_data <- iris[train_indices,]
5 test_data <- iris[-train_indices,]
6 summary_train <- summary(train_data)
7 summary_test <- summary(test_data)
8 print(summary_train)
9 print(summary_test)
10 logistic_model <- glm(Species ~ ., data = train_data, family = "binomial")
11 predicted_probabilities <- predict(logistic_model, newdata = test_data, type = "response")
12 predicted_classes <- ifelse(predicted_probabilities > 0.5, "virginica", "versicolor")
13 confusion_matrix <- table(predicted_classes, test_data$Species)
14 print(confusion_matrix)
15
Rscript /tmp/ITI55SSvPf.r
Sepal.Length Sepal.Width Petal.Length Petal.Width
Min. :4.400 Min. :2.200 Min. :1.200 Min. :0.100
1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.500 1st Qu.:0.200
Median :5.700 Median :3.000 Median :4.100 Median :1.300
Mean :5.735 Mean :3.087 Mean :3.501 Mean :1.101
3rd Qu.:6.300 3rd Qu.:3.350 3rd Qu.:5.000 3rd Qu.:1.800
Max. :7.700 Max. :4.400 Max. :6.700 Max. :2.500
Species
setosa :30
versicolor:21
virginica :24
Sepal.Length Sepal.Width Petal.Length Petal.Width
Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100
1st Qu.:5.400 1st Qu.:2.800 1st Qu.:1.900 1st Qu.:0.500
Median :6.000 Median :3.000 Median :4.500 Median :1.400
Mean :5.952 Mean :3.028 Mean :4.015 Mean :1.297
3rd Qu.:6.500 3rd Qu.:3.300 3rd Qu.:5.300 3rd Qu.:1.900
Max. :7.900 Max. :3.900 Max. :6.900 Max. :2.500
Species
setosa :20
versicolor:29
virginica :26
36°C
Haze
Search
ENG
IN
11:35
15-05-2023


Lab Practice
9.2 Lab WEEK Q_A - Google Doc
R Code: Temp Conversion & Stat
Online R Compiler
programiz.com/r/online-compiler/
Programiz
R Online Compiler
Interactive Python Course
main.r
Run
Output
Clear
1 data(iris)
2 set.seed(42)
3 train_indices <- sample(1:nrow(iris), nrow(iris) * 0.5)
4 train_data <- iris[train_indices,]
5 test_data <- iris[-train_indices,]
6 summary_train <- summary(train_data)
7 summary_test <- summary(test_data)
8 print(summary_train)
9 print(summary_test)
10 logistic_model <- glm(Species ~ ., data = train_data, family = "binomial")
11 predicted_probabilities <- predict(logistic_model, newdata = test_data, type = "response")
12 predicted_classes <- ifelse(predicted_probabilities > 0.5, "virginica", "versicolor")
13 confusion_matrix <- table(predicted_classes, test_data$Species)
14 print(confusion_matrix)
15
versicolor:21
virginica :24
Sepal.Length Sepal.Width Petal.Length Petal.Width
Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100
1st Qu.:5.400 1st Qu.:2.800 1st Qu.:1.900 1st Qu.:0.500
Median :6.000 Median :3.000 Median :4.500 Median :1.400
Mean :5.952 Mean :3.028 Mean :4.015 Mean :1.297
3rd Qu.:6.500 3rd Qu.:3.300 3rd Qu.:5.300 3rd Qu.:1.900
Max. :7.900 Max. :3.900 Max. :6.900 Max. :2.500
Species
setosa :20
versicolor:29
virginica :26
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
predicted_classes setosa versicolor virginica
versicolor 20 0 0
virginica 0 29 26
36°C
Haze
Search
ENG
IN
11:35
15-05-2023

Set-III

1. Suppose you track your commute times for two weeks (10 days) and you find the following times in minutes 17 16 20 24 22 15 21 15 17 22 Enter this into R as vector data type. (i)create function maxi to find the longest commute time, the function avger to find the average and the function mini to find the minimum.

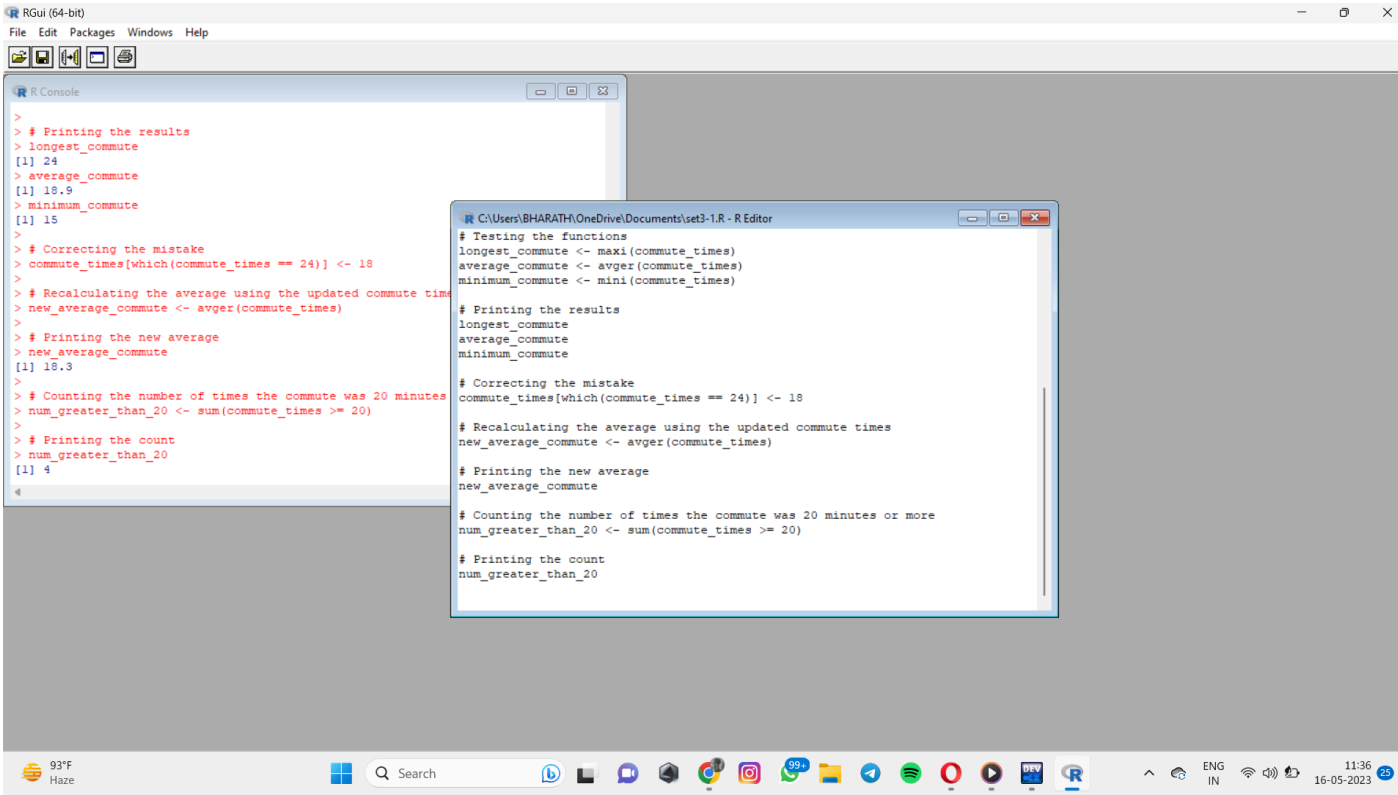
   (ii)Oops, the 24 was a mistake. It should have been 18. How can you fix this? Do so, and then find the new average using above functions.

   (iii)How many times was your commute 20 minutes or more?

   CODE:

```
# Vector of commute times
commute_times <- c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)

# Function to find the longest commute time
maxi <- function(commute_times) {
  max(commute_times)
}

# Function to find the average commute time
avger <- function(commute_times) {
  mean(commute_times)
}

# Function to find the minimum commute time
mini <- function(commute_times) {
  min(commute_times)
}

# Testing the functions
longest_commute <- maxi(commute_times)
average_commute <- avger(commute_times)
minimum_commute <- mini(commute_times)

# Printing the results
longest_commute
average_commute
```

```
minimum_commute

# Correcting the mistake
commute_times[which(commute_times == 24)] <- 18

# Recalculating the average using the updated commute times
new_average_commute <- avger(commute_times)

# Printing the new average
new_average_commute

# Counting the number of times the commute was 20 minutes or more
num_greater_than_20 <- sum(commute_times >= 20)

# Printing the count
num_greater_than_20
```



2. There is a popular built-in data set in R called "**mtcars**" (Motor Trend Car Road

Tests), which is retrieved from the 1974 Motor Trend US Magazine.

(i)Find the dimension of the data set

(ii)Give the statistical summary of the features.

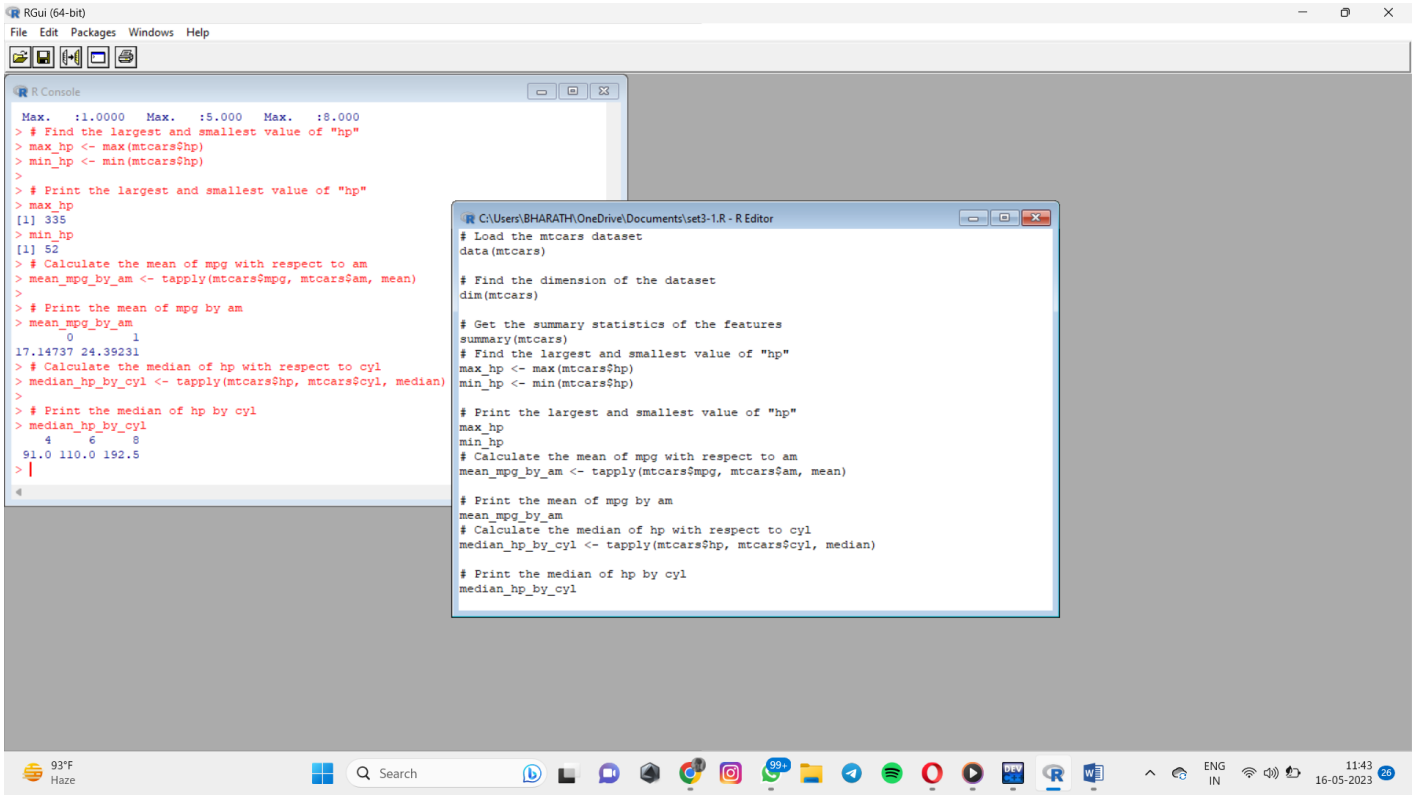
(iii)Find the largest and smallest value of the variable hp (horsepower).

(iv)Give the mean of mileage per gallon (mpg) with respect to transmission model (feature named as 'am')

(v)Give the median of horsepower (hp) with respect to cylinder displacement(cyl)

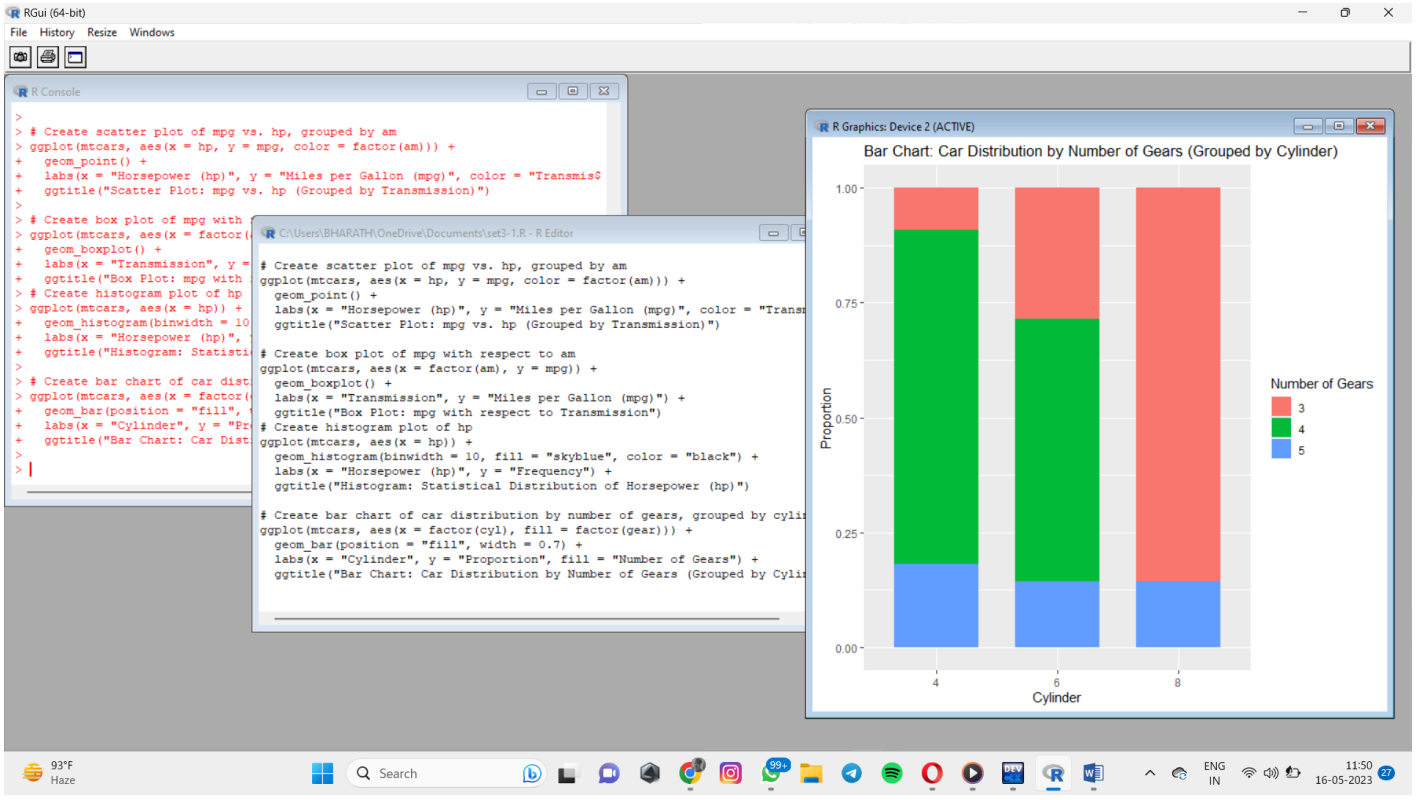
CODE:

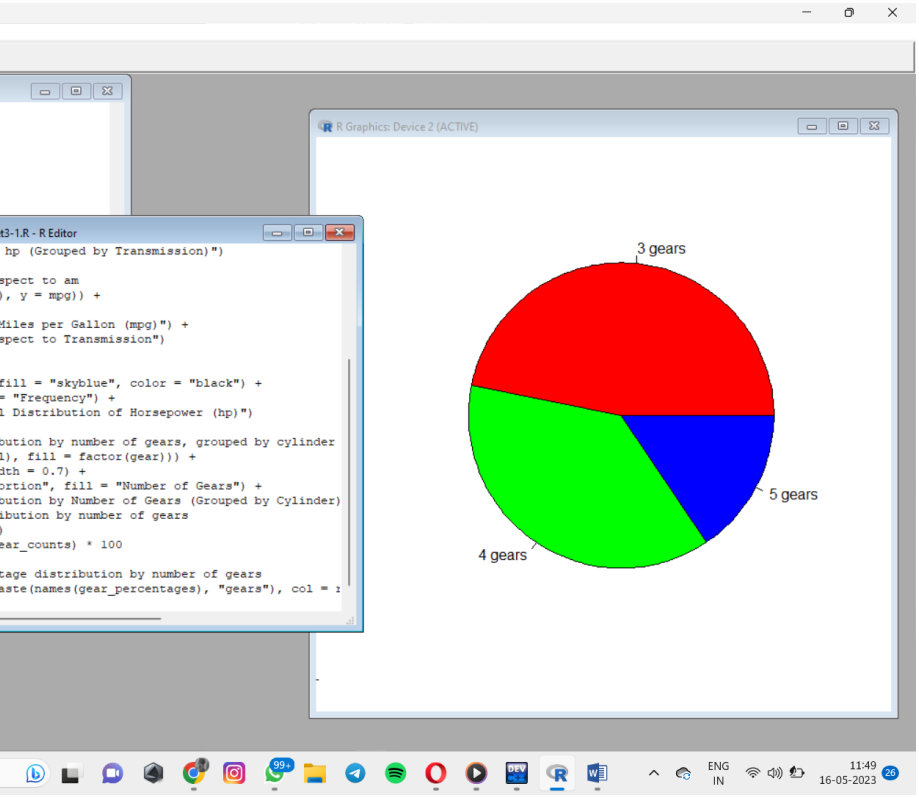
```
# Load the mtcars dataset
data(mtcars)

# Find the dimension of the dataset
dim(mtcars)

# Get the summary statistics of the features
summary(mtcars)
# Find the largest and smallest value of "hp"
max_hp <- max(mtcars$hp)
min_hp <- min(mtcars$hp)

# Print the largest and smallest value of "hp"
max_hp
min_hp
# Calculate the mean of mpg with respect to am
mean_mpg_by_am <- tapply(mtcars$mpg, mtcars$am, mean)

# Print the mean of mpg by am
mean_mpg_by_am
# Calculate the median of hp with respect to cyl
median_hp_by_cyl <- tapply(mtcars$hp, mtcars$cyl, median)

# Print the median of hp by cyl
median_hp_by_cyl
```

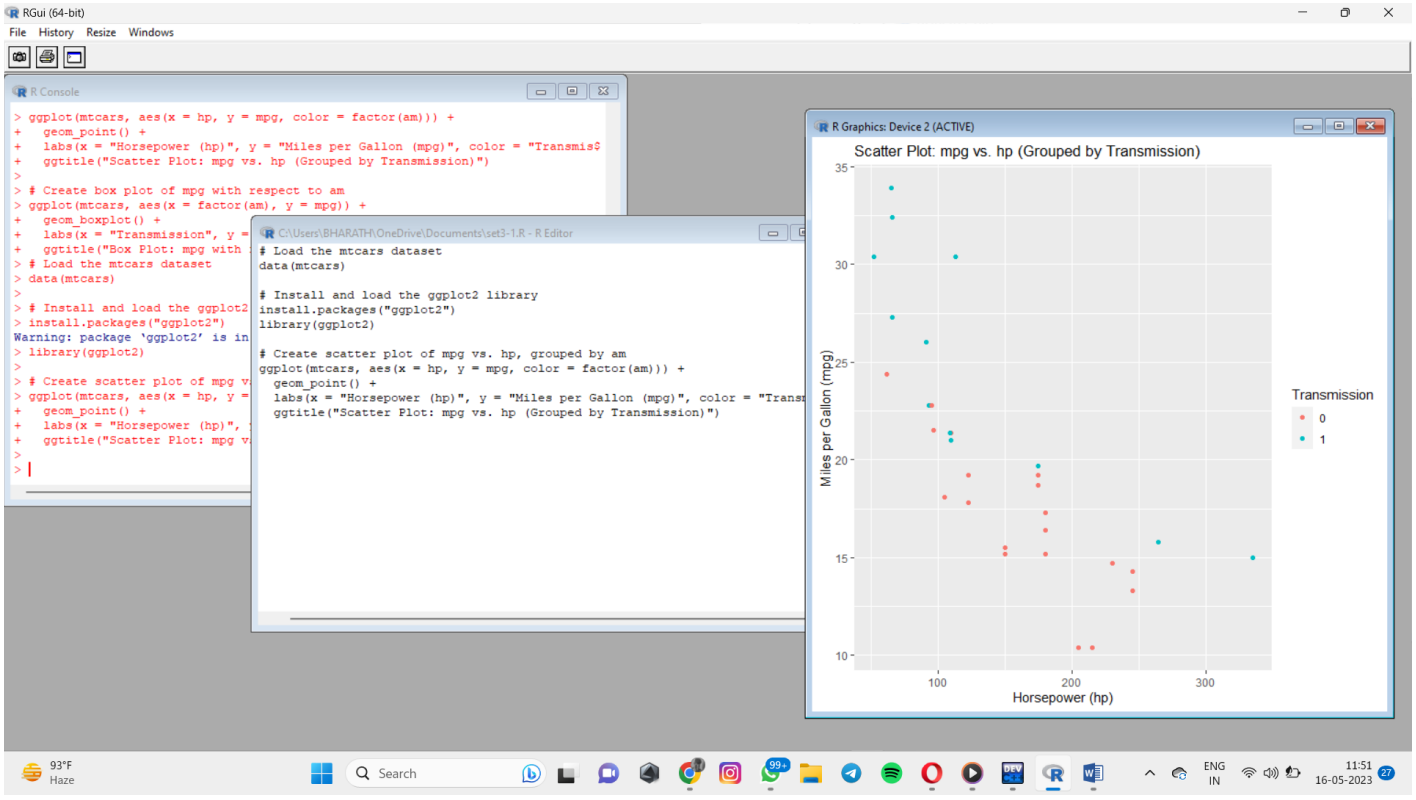


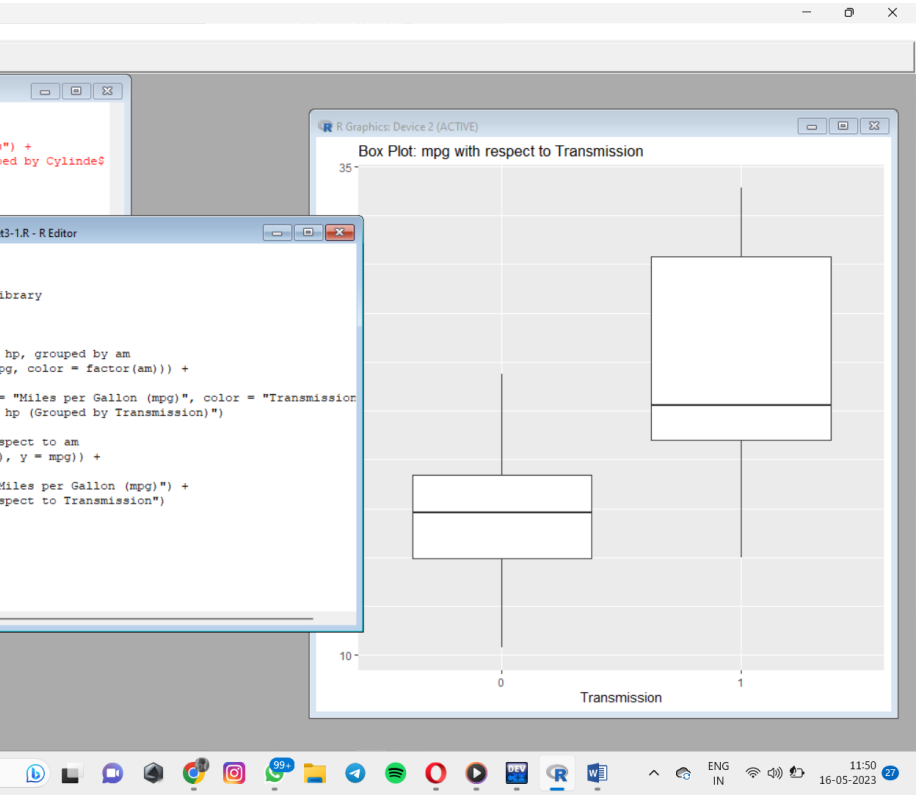
3.(i)Create Scatter plot mpg vs hp, grouped by transmission model (feature named as 'am') (ii)Create Box plot for mpg with respect to transmission model (feature named as 'am') (iii)Create histogram plot which shows statistical distribution of hp (iv)Draw the Bar Chart to show car distribution with respect to number of gears grouped by cylinder.(Grouped or multiple bar chart) (v)Draw Pie chart which shows the percentage of distribution by number of gears. 4. (i)Generate a multiple regression model using the built-in dataset mtcars. Establish the relationship between "mpg" as a response variable with "disp","hp" and "wt" as predictor variables . (ii)Plot the multiple regression line model with above model parameters.

(iii) Predict the mileage of the car with dsp=221, hp=102 and wt=2.91


RGui (64-bit)
File History Resize Windows
R Console
> # Create scatter plot of mpg vs. hp, grouped by am
> ggplot(mtcars, aes(x = hp, y = mpg, color = factor(am))) +
+ geom_point() +
+ labs(x = "Horsepower (hp)", y = "Miles per Gallon (mpg)", color = "Transmis$
+ ggtitle("Scatter Plot: mpg vs. hp (Grouped by Transmission)")
> # Create box plot of mpg with
> ggplot(mtcars, aes(x = factor(
+ geom_boxplot() +
+ labs(x = "Transmission", y =
+ ggtitle("Box Plot: mpg with
> # Create histogram plot of hp
> ggplot(mtcars, aes(x = hp)) +
+ geom_histogram(binwidth = 10
+ labs(x = "Horsepower (hp)",
+ ggtitle("Histogram: Statisti
> # Create bar chart of car dist
> ggplot(mtcars, aes(x = factor(
+ geom_bar(position = "fill",
+ labs(x = "Cylinder", y = "Pr
+ ggtitle("Bar Chart: Car Dist
C:\Users\BHARATH\OneDrive\Documents\set3-1.R - R Editor
Create scatter plot of mpg vs. hp, grouped by am
ggplot(mtcars, aes(x = hp, y = mpg, color = factor(am))) +
geom_point() +
labs(x = "Horsepower (hp)", y = "Miles per Gallon (mpg)", color = "Transr
ggtitle("Scatter Plot: mpg vs. hp (Grouped by Transmission)")
# Create box plot of mpg with respect to am
ggplot(mtcars, aes(x = factor(am), y = mpg)) +
geom_boxplot() +
labs(x = "Transmission", y = "Miles per Gallon (mpg)") +
ggtitle("Box Plot: mpg with respect to Transmission")
# Create histogram plot of hp
ggplot(mtcars, aes(x = hp)) +
geom_histogram(binwidth = 10, fill = "skyblue", color = "black") +
labs(x = "Horsepower (hp)", y = "Frequency") +
ggtitle("Histogram: Statistical Distribution of Horsepower (hp)")
# Create bar chart of car distribution by number of gears, grouped by cylin
ggplot(mtcars, aes(x = factor(cyl), fill = factor(gear))) +
geom_bar(position = "fill", width = 0.7) +
labs(x = "Cylinder", y = "Proportion", fill = "Number of Gears") +
ggtitle("Bar Chart: Car Distribution by Number of Gears (Grouped by Cylin
R Graphics: Device 2 (ACTIVE)
Bar Chart: Car Distribution by Number of Gears (Grouped by Cylinder)
Proportion
1.00
0.75
0.50
0.25
0.00
4
6
8
Cylinder
Number of Gears
3
4
5
93°F
Haze
Search
ENG
IN
11:50
16-05-2023


R Graphics: Device 2 (ACTIVE)
3 gears
5 gears
4 gears
t3-1.R - R Editor
hp (Grouped by Transmission)")
spect to am
), y = mpg)) +
Miles per Gallon (mpg)") +
spect to Transmission")
fill = "skyblue", color = "black") +
= "Frequency") +
l Distribution of Horsepower (hp)")
bution by number of gears, grouped by cylinder
l), fill = factor(gear))) +
dth = 0.7) +
ortion", fill = "Number of Gears") +
bution by Number of Gears (Grouped by Cylinder)
ibution by number of gears
)
ear_counts) * 100
tage distribution by number of gears
aste(names(gear_percentages), "gears"), col =
ENG
IN
11:49
16-05-2023
26


RGui (64-bit)
File History Resize Windows
R Console
> ggplot(mtcars, aes(x = hp, y = mpg, color = factor(am))) +
+ geom_point() +
+ labs(x = "Horsepower (hp)", y = "Miles per Gallon (mpg)", color = "Transmis$
+ ggtitle("Scatter Plot: mpg vs. hp (Grouped by Transmission)")
> # Create box plot of mpg with respect to am
> ggplot(mtcars, aes(x = factor(am), y = mpg)) +
+ geom_boxplot() +
+ labs(x = "Transmission", y =
+ ggtitle("Box Plot: mpg with
> # Load the mtcars dataset
> data(mtcars)
> # Install and load the ggplot2
> install.packages("ggplot2")
Warning: package 'ggplot2' is in
> library(ggplot2)
> # Create scatter plot of mpg v
> ggplot(mtcars, aes(x = hp, y =
+ geom_point() +
+ labs(x = "Horsepower (hp)",
+ ggtitle("Scatter Plot: mpg v
C:\Users\BHARATH\OneDrive\Documents\set3-1.R - R Editor
# Load the mtcars dataset
data(mtcars)
# Install and load the ggplot2 library
install.packages("ggplot2")
library(ggplot2)
# Create scatter plot of mpg vs. hp, grouped by am
ggplot(mtcars, aes(x = hp, y = mpg, color = factor(am))) +
geom_point() +
labs(x = "Horsepower (hp)", y = "Miles per Gallon (mpg)", color = "Transm
ggtitle("Scatter Plot: mpg vs. hp (Grouped by Transmission)")
R Graphics: Device 2 (ACTIVE)
Scatter Plot: mpg vs. hp (Grouped by Transmission)
Miles per Gallon (mpg)
35
30
25
20
15
10
100
200
300
Horsepower (hp)
Transmission
0
1
93°F
Haze
Search
ENG
IN
11:51
16-05-2023



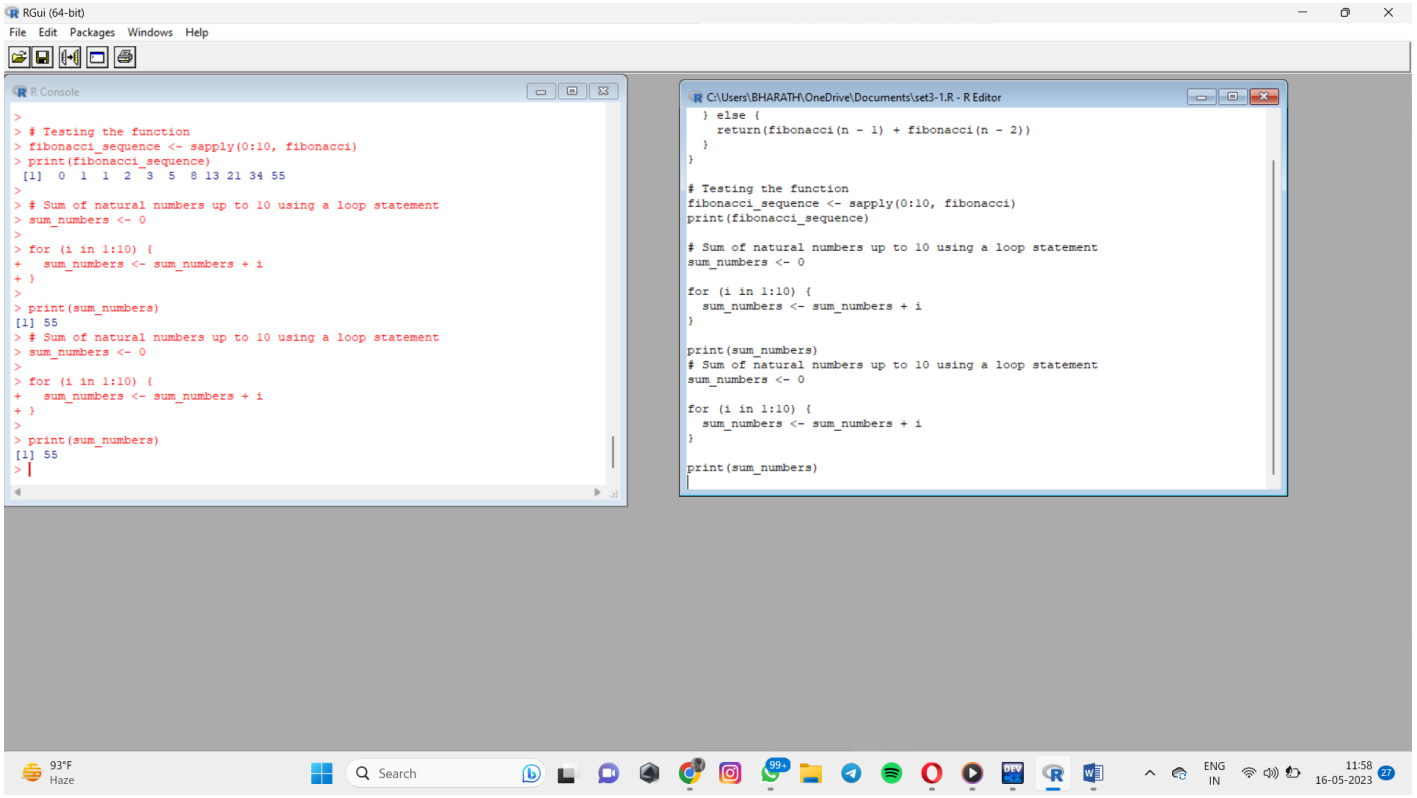
Set IV

1. (i) Write a function in R programming to print generate Fibonacci sequence using Recursion in R .
   (ii) Find sum of natural numbers up-to 10, without formula using loop statement.
   (iii) create a vector 1:10 and Find a square of each number and store that in a separate list.

CODE:

```
fibonacci <- function(n) {
  if (n <= 1) {
    return(n)
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))
  }
}
```

```
# Testing the function
fibonacci_sequence <- sapply(0:10, fibonacci)
print(fibonacci_sequence)

# Sum of natural numbers up to 10 using a loop statement
sum_numbers <- 0

for (i in 1:10) {
  sum_numbers <- sum_numbers + i
}

print(sum_numbers)
# Sum of natural numbers up to 10 using a loop statement
sum_numbers <- 0

for (i in 1:10) {
  sum_numbers <- sum_numbers + i
}

print(sum_numbers)
```



2. **mtcars**(motor trend car road test) comprises fuel consumption, performance and 10 aspects of automobile design for 32 automobiles. It comes pre-installed with **dplyr** package in R.

(i)Find the dimension of the data set

(ii)Give the statistical summary of the features.

(iii)Print the categorical features in Dataset

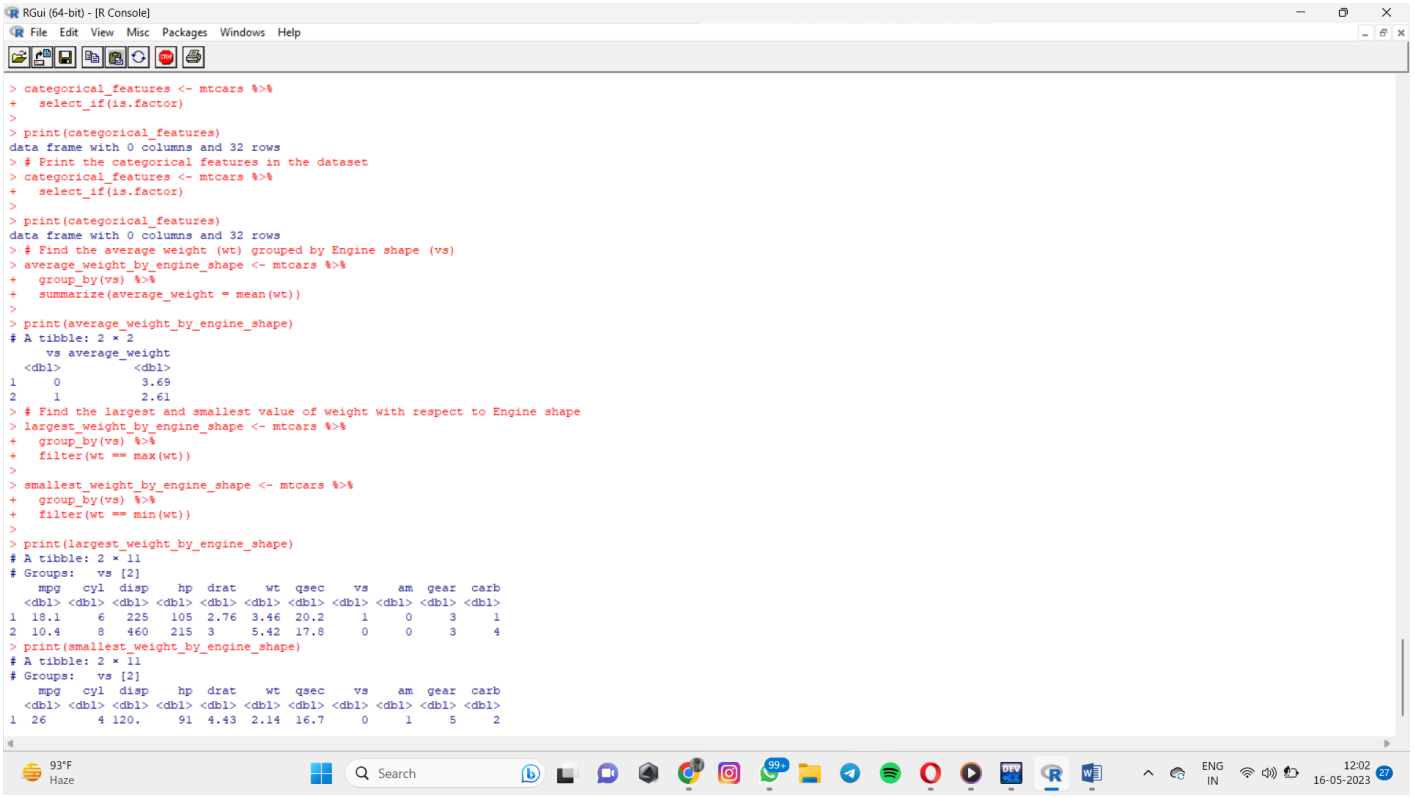
(iv)Find the average weight(wt) grouped by Engine shape(vs)

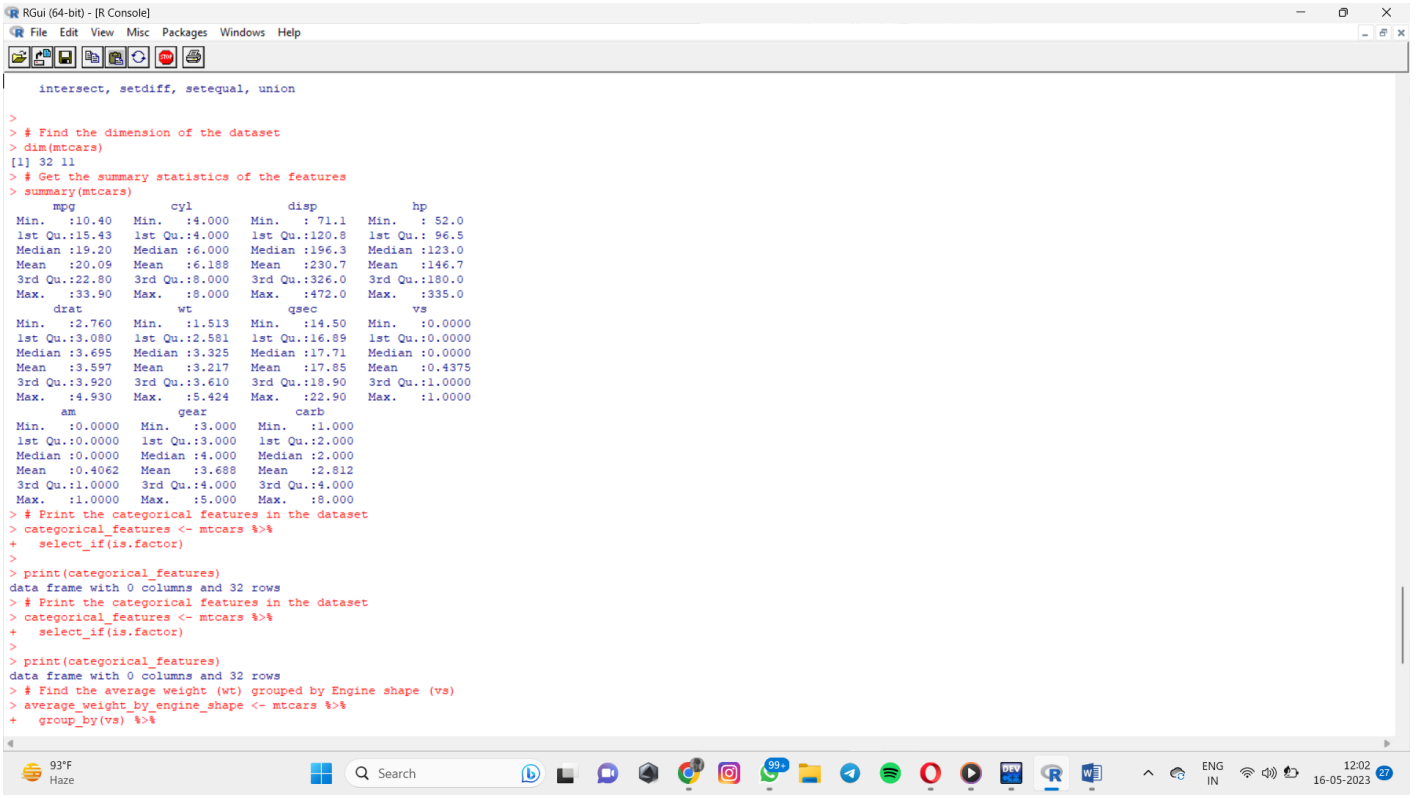
(v)Find the largest and smallest value of the variable weight with respect to Engine shape

CODE:

```
# Load the dplyr package
library(dplyr)

# Find the dimension of the dataset
dim(mtcars)
```

```
# Get the summary statistics of the features
summary(mtcars)
# Print the categorical features in the dataset
categorical_features <- mtcars %>%
  select_if(is.factor)

print(categorical_features)
# Print the categorical features in the dataset
categorical_features <- mtcars %>%
  select_if(is.factor)

print(categorical_features)
# Find the average weight (wt) grouped by Engine shape (vs)
average_weight_by_engine_shape <- mtcars %>%
  group_by(vs) %>%
  summarize(average_weight = mean(wt))

print(average_weight_by_engine_shape)
# Find the largest and smallest value of weight with respect to Engine shape
largest_weight_by_engine_shape <- mtcars %>%
  group_by(vs) %>%
  filter(wt == max(wt))

smallest_weight_by_engine_shape <- mtcars %>%
  group_by(vs) %>%
  filter(wt == min(wt))

print(largest_weight_by_engine_shape)
print(smallest_weight_by_engine_shape)
```


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
> categorical_features <- mtcars %>%
+ select_if(is.factor)
>
> print(categorical_features)
data frame with 0 columns and 32 rows
> # Print the categorical features in the dataset
> categorical_features <- mtcars %>%
+ select_if(is.factor)
>
> print(categorical_features)
data frame with 0 columns and 32 rows
> # Find the average weight (wt) grouped by Engine shape (vs)
> average_weight_by_engine_shape <- mtcars %>%
+ group_by(vs) %>%
+ summarize(average_weight = mean(wt))
>
> print(average_weight_by_engine_shape)
A tibble: 2 × 2
vs average_weight
<dbl> <dbl>
1 0 3.69
2 1 2.61
> # Find the largest and smallest value of weight with respect to Engine shape
> largest_weight_by_engine_shape <- mtcars %>%
+ group_by(vs) %>%
+ filter(wt == max(wt))
>
> smallest_weight_by_engine_shape <- mtcars %>%
+ group_by(vs) %>%
+ filter(wt == min(wt))
>
> print(largest_weight_by_engine_shape)
A tibble: 2 × 11
Groups: vs [2]
mpg cyl disp hp drat wt qsec vs am gear carb
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 18.1 6 225 105 2.76 3.46 20.2 1 0 3 1
2 10.4 8 460 215 3 5.42 17.8 0 0 3 4
> print(smallest_weight_by_engine_shape)
A tibble: 2 × 11
Groups: vs [2]
mpg cyl disp hp drat wt qsec vs am gear carb
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 26 4 120. 91 4.43 2.14 16.7 0 1 5 2
93°F
Haze
Search
ENG
IN
12:02
16-05-2023

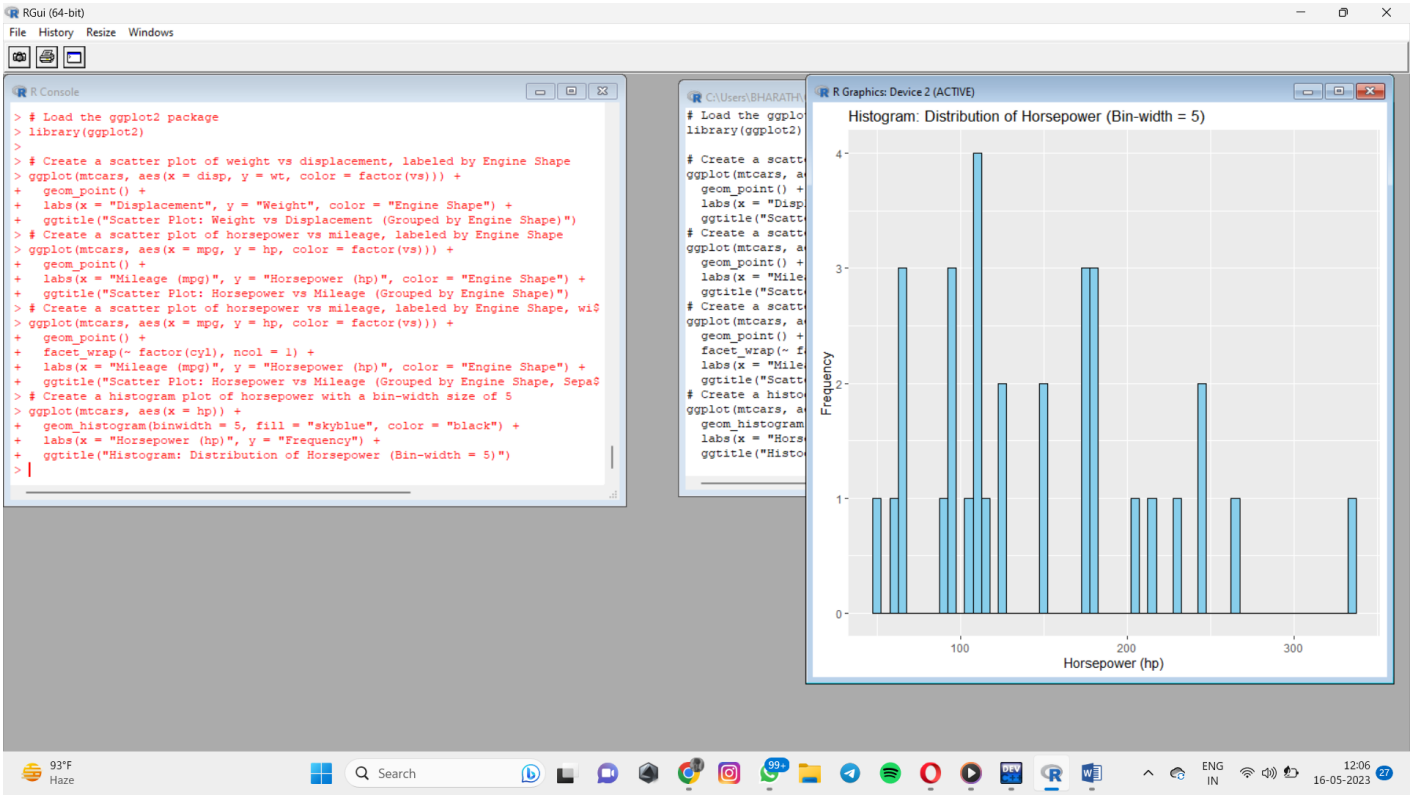


3.Use ggplot package to plot below EDA questions label the plot accordingly (i)Create weight(wt) vs displacement(disp) scatter plot factor by Engine Shape(vs) (ii) Create horsepower (hp) vs mileage (mgp) scatter plot factor by Engine Shape(vs) (iv)In above(ii) plot , Separate columns according to cylinders(cyl) size

(v) Create histogram plot for horsepower (hp) with bin-width size of 5

```
# Load the ggplot2 package
library(ggplot2)

# Create a scatter plot of weight vs displacement, labeled by Engine Shape
ggplot(mtcars, aes(x = disp, y = wt, color = factor(vs))) +
  geom_point() +
  labs(x = "Displacement", y = "Weight", color = "Engine Shape") +
  ggtitle("Scatter Plot: Weight vs Displacement (Grouped by Engine Shape)")
```

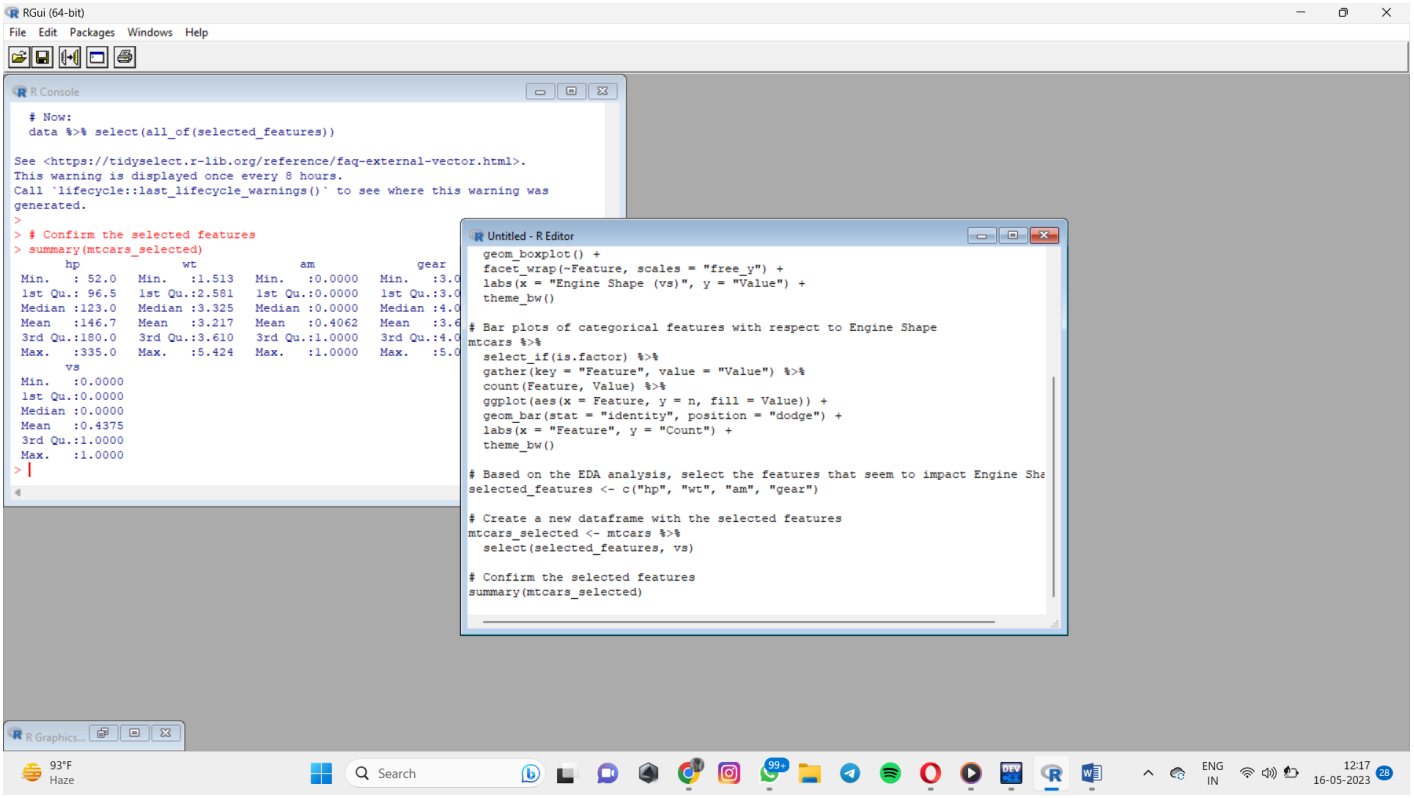
```
# Create a scatter plot of horsepower vs mileage, labeled by Engine Shape
ggplot(mtcars, aes(x = mpg, y = hp, color = factor(vs))) +
  geom_point() +
  labs(x = "Mileage (mpg)", y = "Horsepower (hp)", color = "Engine Shape") +
  ggtitle("Scatter Plot: Horsepower vs Mileage (Grouped by Engine Shape)")


# Create a scatter plot of horsepower vs mileage, labeled by Engine Shape, with separate columns for cylinder size
ggplot(mtcars, aes(x = mpg, y = hp, color = factor(vs))) +
  geom_point() +
  facet_wrap(~ factor(cyl), ncol = 1) +
  labs(x = "Mileage (mpg)", y = "Horsepower (hp)", color = "Engine Shape") +
  ggtitle("Scatter Plot: Horsepower vs Mileage (Grouped by Engine Shape, Separate Columns for Cylinder Size)")

# Create a histogram plot of horsepower with a bin-width size of 5
ggplot(mtcars, aes(x = hp)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black") +
  labs(x = "Horsepower (hp)", y = "Frequency") +
  ggtitle("Histogram: Distribution of Horsepower (Bin-width = 5)")
```

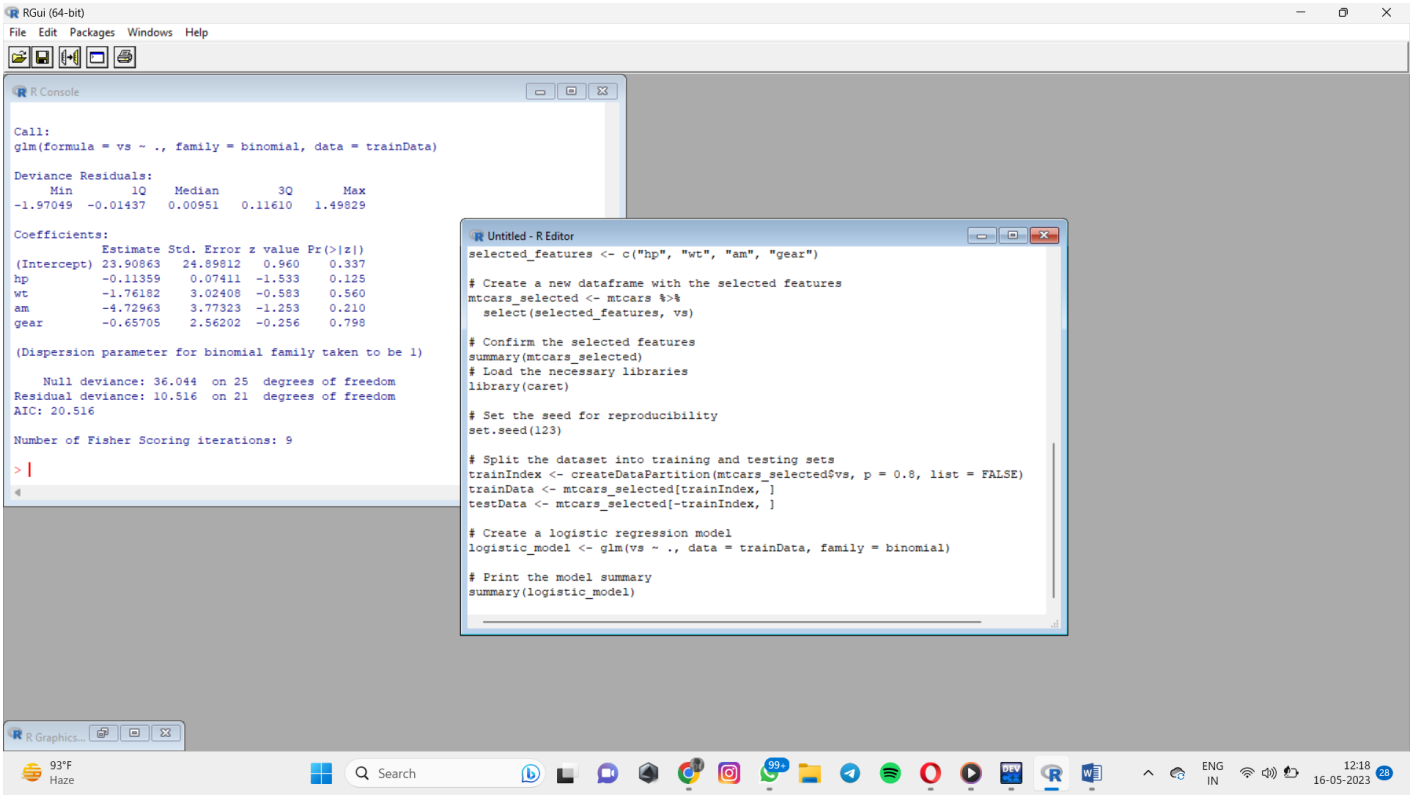


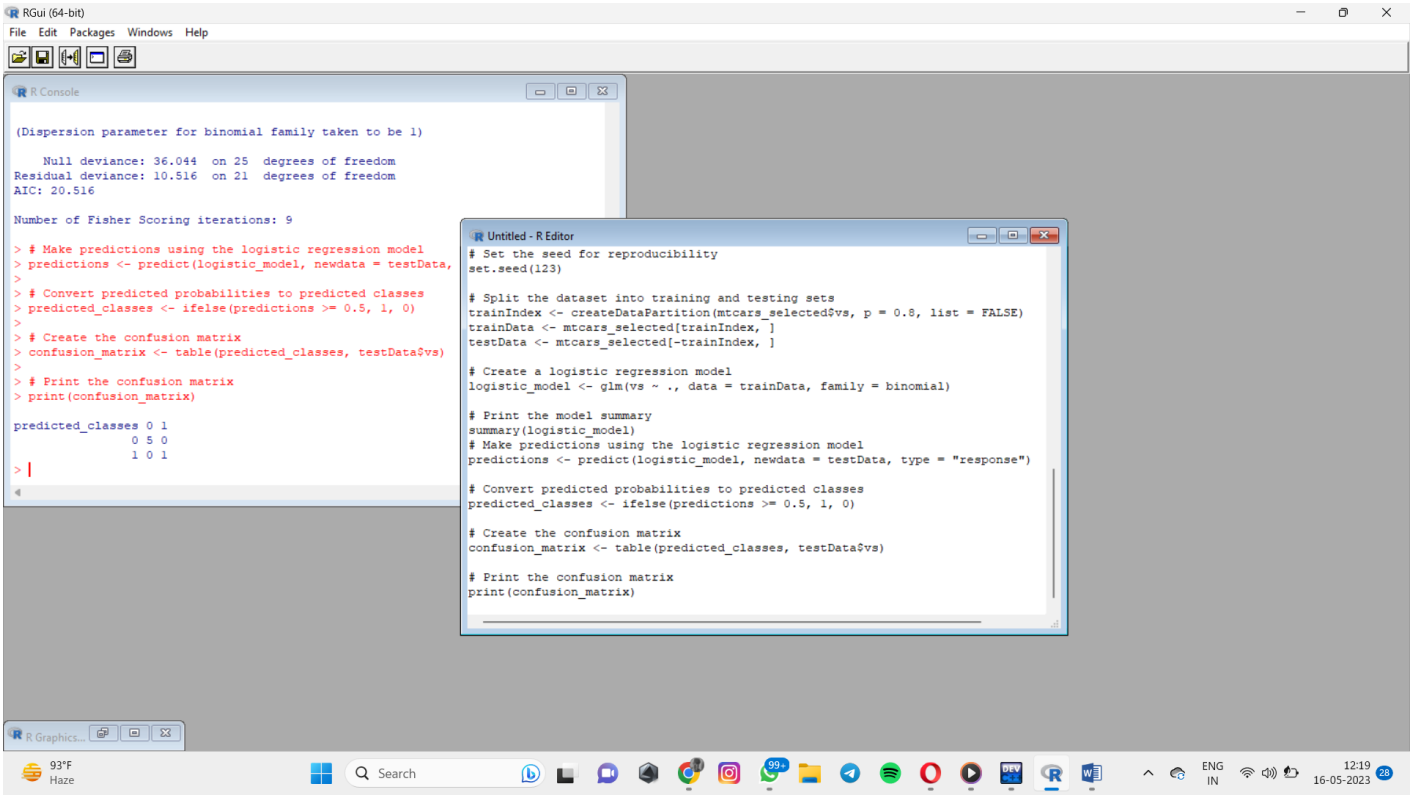
4. Performing Logistic regression on dataset to predict the cars Engine shape(vs) . (i)Do the EDA analysis and find the features which is impact the Engine shape and use this for model.

(ii) Split the data set randomly with 80:20 ration to create train and test dataset and create logistic model

(iii)Create the Confusion matrix among prediction and test data.


RGui (64-bit)
File Edit Packages Windows Help
R Console
Now:
data %>% select(all_of(selected_features))
See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> # Confirm the selected features
> summary(mtcars_selected)
hp wt am gear
Min. : 52.0 Min. :1.513 Min. :0.0000 Min. :3.0
1st Qu.: 96.5 1st Qu.:2.581 1st Qu.:0.0000 1st Qu.:3.0
Median :123.0 Median :3.325 Median :0.0000 Median :4.0
Mean :146.7 Mean :3.217 Mean :0.4062 Mean :3.6
3rd Qu.:180.0 3rd Qu.:3.610 3rd Qu.:1.0000 3rd Qu.:4.0
Max. :335.0 Max. :5.424 Max. :1.0000 Max. :5.0
vs
Min. :0.0000
1st Qu.:0.0000
Median :0.0000
Mean :0.4375
3rd Qu.:1.0000
Max. :1.0000
Untitled - R Editor
geom_boxplot() +
facet_wrap(~Feature, scales = "free_y") +
labs(x = "Engine Shape (vs)", y = "Value") +
theme_bw()
# Bar plots of categorical features with respect to Engine Shape
mtcars %>%
select_if(is.factor) %>%
gather(key = "Feature", value = "Value") %>%
count(Feature, Value) %>%
ggplot(aes(x = Feature, y = n, fill = Value)) +
geom_bar(stat = "identity", position = "dodge") +
labs(x = "Feature", y = "Count") +
theme_bw()
# Based on the EDA analysis, select the features that seem to impact Engine Sha
selected_features <- c("hp", "wt", "am", "gear")
# Create a new dataframe with the selected features
mtcars_selected <- mtcars %>%
select(selected_features, vs)
# Confirm the selected features
summary(mtcars_selected)
R Graphics...
93°F Haze
Search
ENG IN
12:17 16-05-2023


RGui (64-bit)
File Edit Packages Windows Help
R Console
Call:
glm(formula = vs ~ ., family = binomial, data = trainData)
Deviance Residuals:
Min 1Q Median 3Q Max
-1.97049 -0.01437 0.00951 0.11610 1.49829
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) 23.90863 24.89812 0.960 0.337
hp -0.11359 0.07411 -1.533 0.125
wt -1.76182 3.02408 -0.583 0.560
am -4.72963 3.77323 -1.253 0.210
gear -0.65705 2.56202 -0.256 0.798
(Dispersion parameter for binomial family taken to be 1)
Null deviance: 36.044 on 25 degrees of freedom
Residual deviance: 10.516 on 21 degrees of freedom
AIC: 20.516
Number of Fisher Scoring iterations: 9
>
Untitled - R Editor
selected_features <- c("hp", "wt", "am", "gear")
# Create a new dataframe with the selected features
mtcars_selected <- mtcars %>%
select(selected_features, vs)
# Confirm the selected features
summary(mtcars_selected)
# Load the necessary libraries
library(caret)
# Set the seed for reproducibility
set.seed(123)
# Split the dataset into training and testing sets
trainIndex <- createDataPartition(mtcars_selected$vs, p = 0.8, list = FALSE)
trainData <- mtcars_selected[trainIndex,]
testData <- mtcars_selected[-trainIndex,]
# Create a logistic regression model
logistic_model <- glm(vs ~ ., data = trainData, family = binomial)
# Print the model summary
summary(logistic_model)
R Graphics...
93°F
Haze
Search
ENG
IN
12:18
16-05-2023



Set-V

1.(i) Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)

(ii)Write R function to find the range of given vector. Range=Max-Min

Sample input, C<-(9,8,7,6,5,4,3,2,1), output=8

(iii)Wirte the R function to find the number of vowels in given string

Sample input c<- “matrix”, output<-2

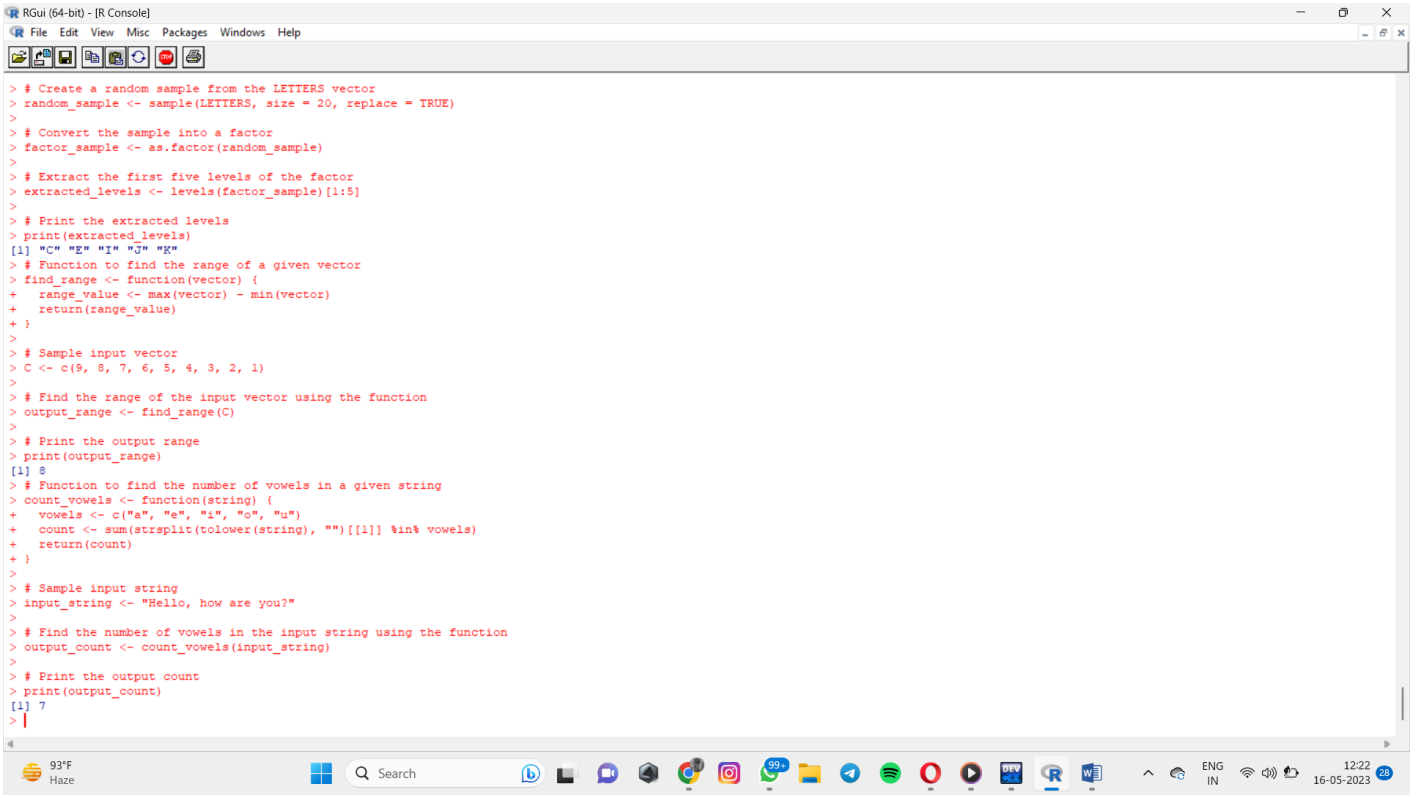
CODE:

Set the seed for reproducibility

set.seed(123)

```
# Create a random sample from the LETTERS vector
random_sample <- sample(LETTERS, size = 20, replace = TRUE)

# Convert the sample into a factor
factor_sample <- as.factor(random_sample)

# Extract the first five levels of the factor
extracted_levels <- levels(factor_sample)[1:5]

# Print the extracted levels
print(extracted_levels)
# Function to find the range of a given vector
find_range <- function(vector) {
  range_value <- max(vector) - min(vector)
  return(range_value)
}

# Sample input vector
C <- c(9, 8, 7, 6, 5, 4, 3, 2, 1)

# Find the range of the input vector using the function
output_range <- find_range(C)
```

```
# Print the output range
print(output_range)
# Function to find the number of vowels in a given string
count_vowels <- function(string) {
  vowels <- c("a", "e", "i", "o", "u")
  count <- sum(strsplit(tolower(string), "")[[1]] %in% vowels)
  return(count)
}

# Sample input string
input_string <- "Hello, how are you?"

# Find the number of vowels in the input string using the function
output_count <- count_vowels(input_string)

# Print the output count
print(output_count)
```



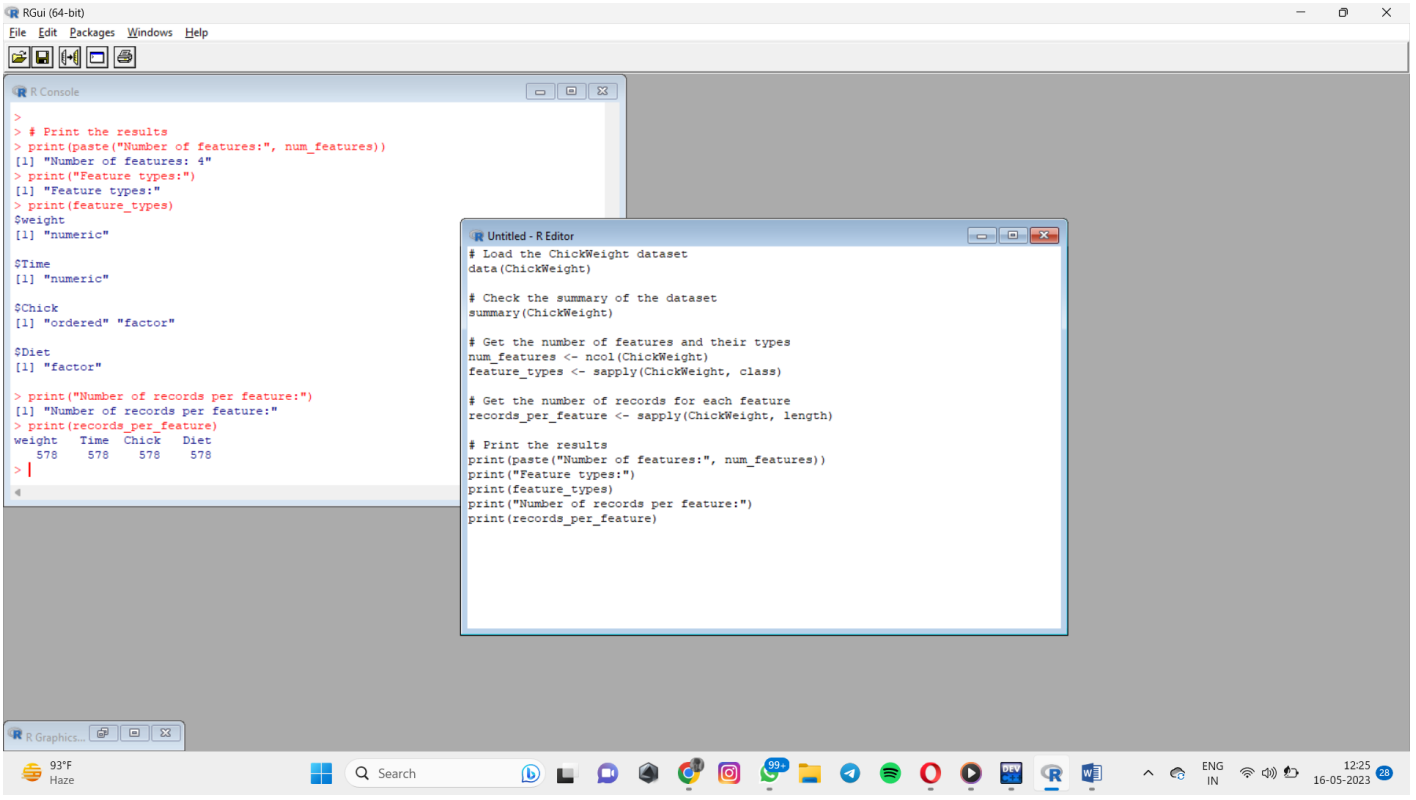
2.Load inbuild dataset “ChickWeight” in R

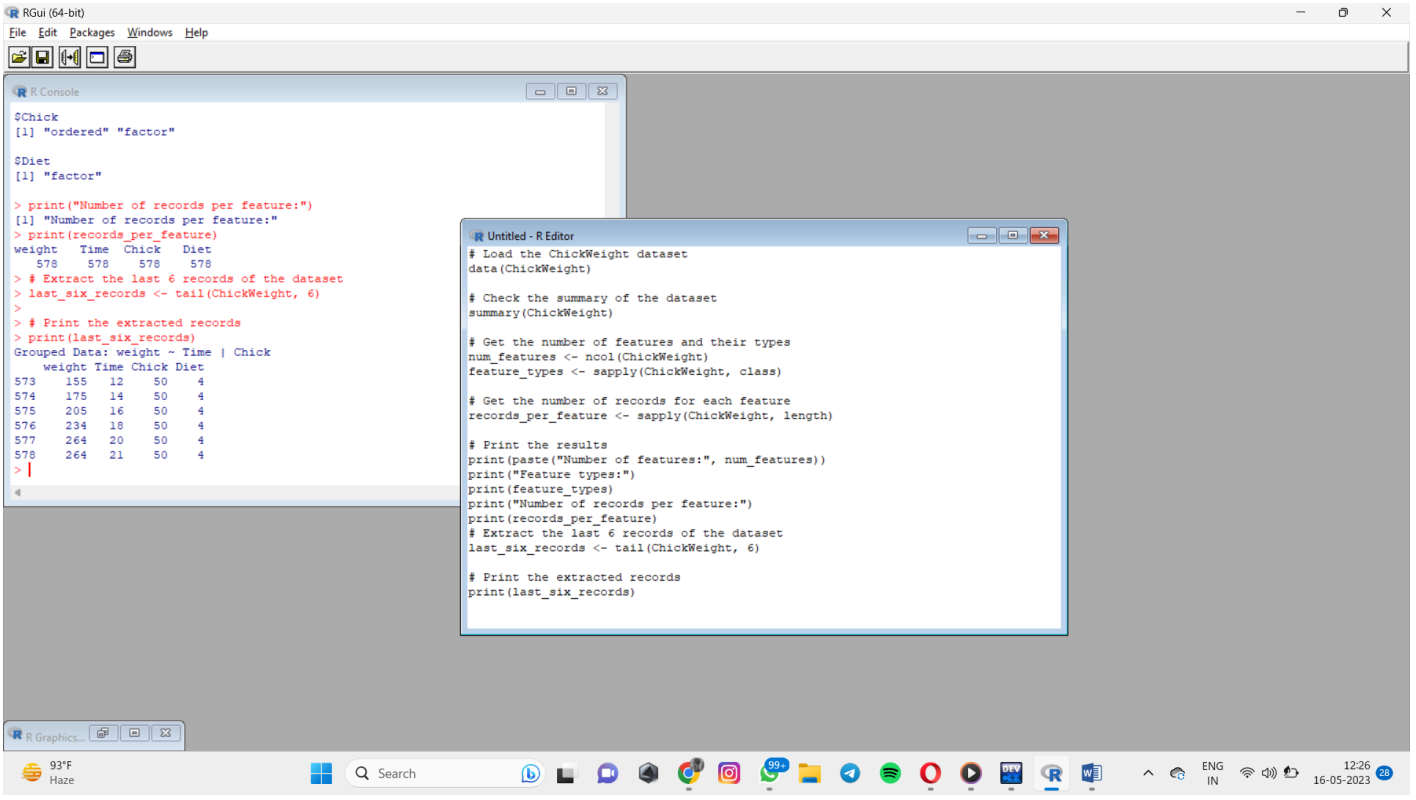
(i) Explore the summary of Data set, like number of Features and its type. Finds the number of records for each features

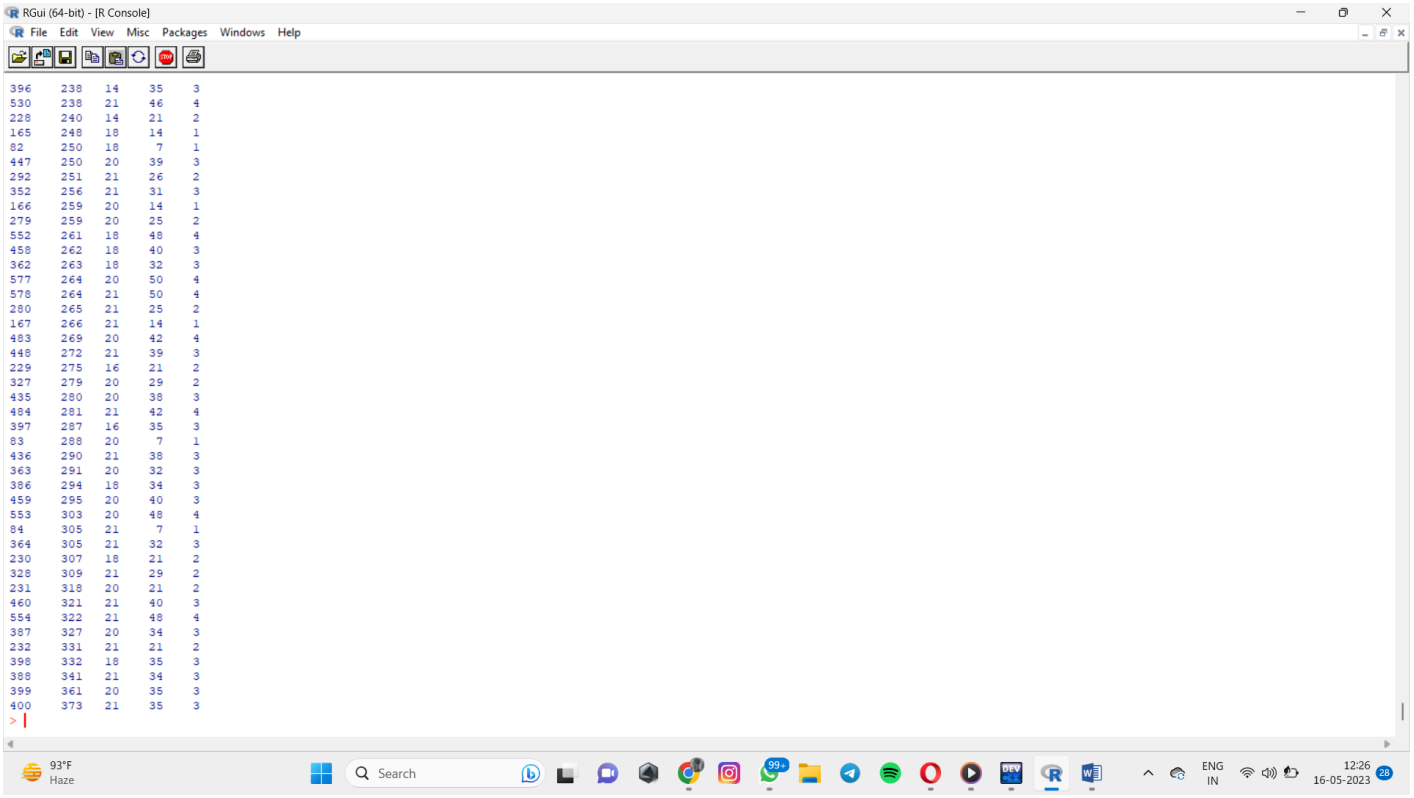
(ii)Extract last 6 records of dataset

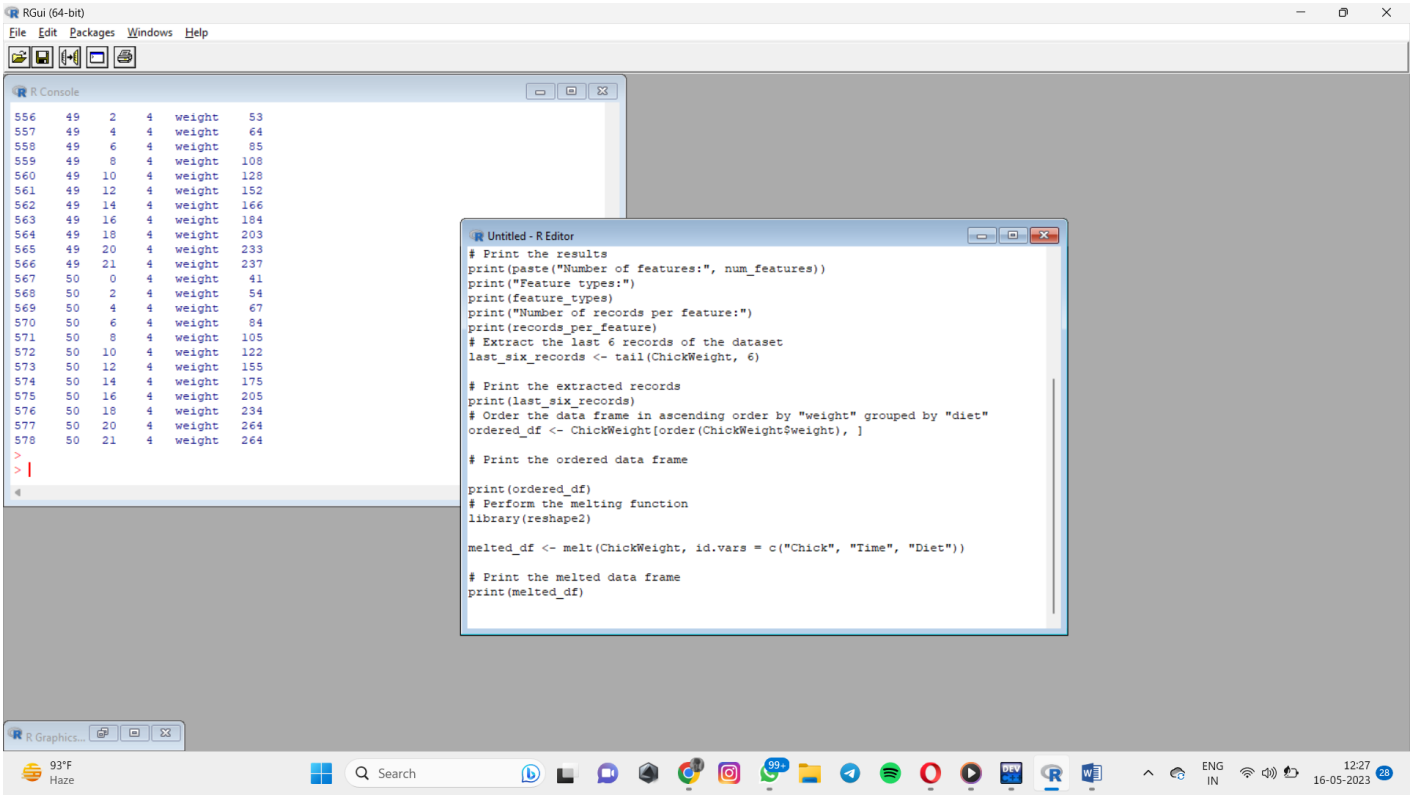
(iii) order the data frame, in ascending order by feature name “weight” grouped by feature “diet”

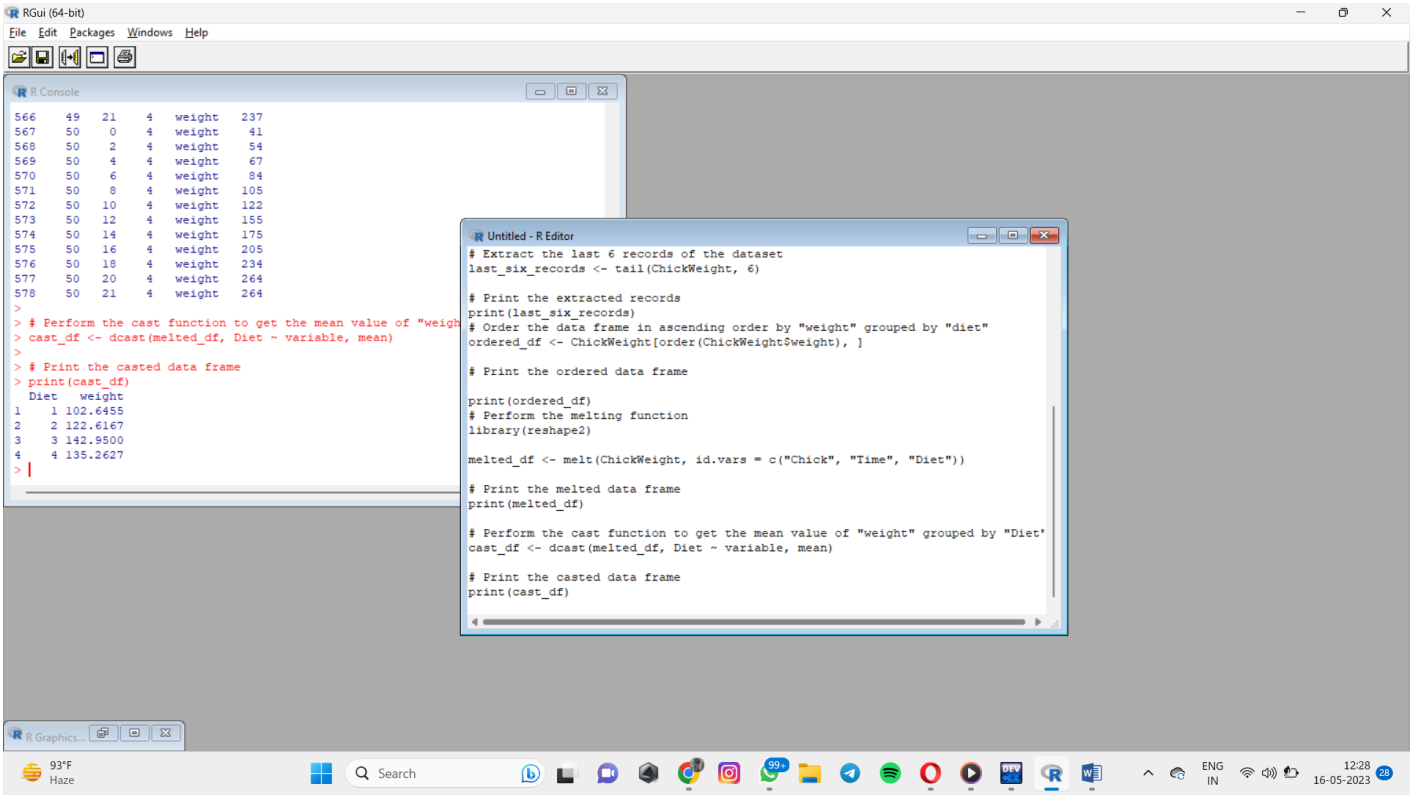
(iv)Perform melting function based on “Chick","Time","Diet" features as ID variables (v)Perform cast function to display the mean value of weight grouped by Diet


RGui (64-bit)
File Edit Packages Windows Help
R Console
> # Print the results
> print(paste("Number of features:", num_features))
[1] "Number of features: 4"
> print("Feature types:")
[1] "Feature types:"
> print(feature_types)
$weight
[1] "numeric"
$Time
[1] "numeric"
$Chick
[1] "ordered" "factor"
$Diet
[1] "factor"
> print("Number of records per feature:")
[1] "Number of records per feature:"
> print(records_per_feature)
weight Time Chick Diet
578 578 578 578
Untitled - R Editor
# Load the ChickWeight dataset
data(ChickWeight)
# Check the summary of the dataset
summary(ChickWeight)
# Get the number of features and their types
num_features <- ncol(ChickWeight)
feature_types <- sapply(ChickWeight, class)
# Get the number of records for each feature
records_per_feature <- sapply(ChickWeight, length)
# Print the results
print(paste("Number of features:", num_features))
print("Feature types:")
print(feature_types)
print("Number of records per feature:")
print(records_per_feature)
R Graphics...
93°F
Haze
Search
ENG
IN
12:25
16-05-2023


RGui (64-bit)
File Edit Packages Windows Help
R Console
$Chick
[1] "ordered" "factor"
$Diet
[1] "factor"
> print("Number of records per feature:")
[1] "Number of records per feature:"
> print(records_per_feature)
weight Time Chick Diet
578 578 578 578
> # Extract the last 6 records of the dataset
> last_six_records <- tail(ChickWeight, 6)
>
> # Print the extracted records
> print(last_six_records)
Grouped Data: weight ~ Time | Chick
weight Time Chick Diet
573 155 12 50 4
574 175 14 50 4
575 205 16 50 4
576 234 18 50 4
577 264 20 50 4
578 264 21 50 4
>
Untitled - R Editor
# Load the ChickWeight dataset
data(ChickWeight)
# Check the summary of the dataset
summary(ChickWeight)
# Get the number of features and their types
num_features <- ncol(ChickWeight)
feature_types <- sapply(ChickWeight, class)
# Get the number of records for each feature
records_per_feature <- sapply(ChickWeight, length)
# Print the results
print(paste("Number of features:", num_features))
print("Feature types:")
print(feature_types)
print("Number of records per feature:")
print(records_per_feature)
# Extract the last 6 records of the dataset
last_six_records <- tail(ChickWeight, 6)
# Print the extracted records
print(last_six_records)
R Graphics...
93°F
Haze
Search
ENG
IN
12:26
16-05-2023


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
396 238 14 35 3
530 238 21 46 4
228 240 14 21 2
165 248 18 14 1
82 250 18 7 1
447 250 20 39 3
292 251 21 26 2
352 256 21 31 3
166 259 20 14 1
279 259 20 25 2
552 261 18 48 4
458 262 18 40 3
362 263 18 32 3
577 264 20 50 4
578 264 21 50 4
280 265 21 25 2
167 266 21 14 1
483 269 20 42 4
448 272 21 39 3
229 275 16 21 2
327 279 20 29 2
435 280 20 38 3
484 281 21 42 4
397 287 16 35 3
83 288 20 7 1
436 290 21 38 3
363 291 20 32 3
386 294 18 34 3
459 295 20 40 3
553 303 20 48 4
84 305 21 7 1
364 305 21 32 3
230 307 18 21 2
328 309 21 29 2
231 318 20 21 2
460 321 21 40 3
554 322 21 48 4
387 327 20 34 3
232 331 21 21 2
398 332 18 35 3
388 341 21 34 3
399 361 20 35 3
400 373 21 35 3
>
93°F
Haze
Search
ENG
IN
12:26
16-05-2023


RGui (64-bit)
File Edit Packages Windows Help
R Console
556 49 2 4 weight 53
557 49 4 4 weight 64
558 49 6 4 weight 85
559 49 8 4 weight 108
560 49 10 4 weight 128
561 49 12 4 weight 152
562 49 14 4 weight 166
563 49 16 4 weight 184
564 49 18 4 weight 203
565 49 20 4 weight 233
566 49 21 4 weight 237
567 50 0 4 weight 41
568 50 2 4 weight 54
569 50 4 4 weight 67
570 50 6 4 weight 84
571 50 8 4 weight 105
572 50 10 4 weight 122
573 50 12 4 weight 155
574 50 14 4 weight 175
575 50 16 4 weight 205
576 50 18 4 weight 234
577 50 20 4 weight 264
578 50 21 4 weight 264
>
>
Untitled - R Editor
# Print the results
print(paste("Number of features:", num_features))
print("Feature types:")
print(feature_types)
print("Number of records per feature:")
print(records_per_feature)
# Extract the last 6 records of the dataset
last_six_records <- tail(ChickWeight, 6)
# Print the extracted records
print(last_six_records)
# Order the data frame in ascending order by "weight" grouped by "diet"
ordered_df <- ChickWeight[order(ChickWeight$weight),]
# Print the ordered data frame
print(ordered_df)
# Perform the melting function
library(reshape2)
melted_df <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))
# Print the melted data frame
print(melted_df)
R Graphics...
93°F
Haze
Search
ENG
IN
12:27
16-05-2023



3.(i)Get the Statistical Summary of "ChickWeight" dataset

(ii)Create Box plot for "weight" grouped by "Diet"

(iii)Create a Histogram for "Weight" features belong to Diet- 1 category

(iv) Create a Histogram for "Weight" features belong to Diet- 4 category

(v) Create Scatter plot for weight vs Time grouped by Diet

4.(i) Create multi regression model to find a weight of the chicken , by "Time" and "Diet" as as predictor variables

(ii) Predict weight for Time=10 and Diet=1

(iii)Find the error(MAE) in model for same