SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

ITA 0451 - STATISTICS WITH R PROGRAMMING

DAY 4 – LAB ASSESSMENT Part 4

Reg No:192110399

Name: GOKUL S

1.   a. Create multi regression model to find a weight of the chicken , by “Time” and

“Diet” as  as

predictor variables

          b. Predict weight for Time=10 and Diet=1

           c. Find the error in model for same

Code:

time <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

diet <- c(1, 1, 1, 1, 1, 2, 2, 2, 2, 2)

weight <- c(2.1, 2.4, 2.6, 2.9, 3.2, 3.5, 3.8, 4.1, 4.4, 4.7)

data <- data.frame(time, diet, weight)

model <- lm(weight ~ time + diet, data=data)

new\_data <- data.frame(time=10, diet=1)

prediction <- predict(model, newdata=new\_data)

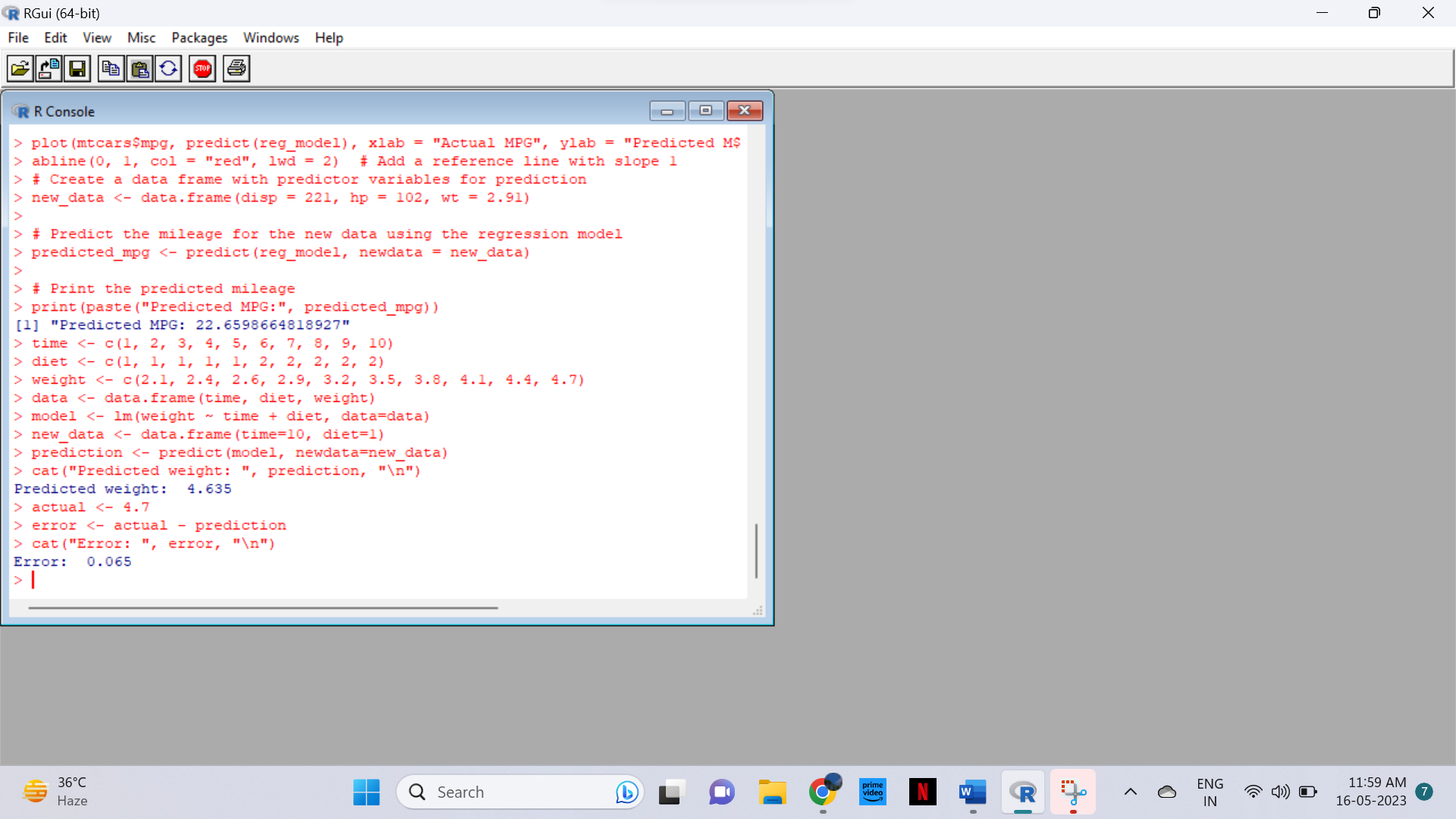
cat("Predicted weight: ", prediction, "\n")

actual <- 4.7

error <- actual - prediction

cat("Error: ", error, "\n")

Output:



3. Explore the USArrests dataset, contains the number of arrests for murder, assault, and

rape for each of the 50 states in 1973. It also contains the percentage of people in the state

who live in an urban area.

(i) a. Explore the summary of Data set, like number of Features and its type. Find the

number         of records for each feature. Print the statistical feature of data

         b. Print the state which saw the largest total number of rape

         c. Print the states with the max &amp; min crime rates for murder

   (ii).a. Find the correlation among the features

    b. Print the states which have assault arrests more than median of the country

    c. Print the states are in the bottom 25% of murder

  (iii). a. Create a histogram and density plot of murder arrests by US stat

b. Create the plot that shows the relationship between murder arrest rate and

proportion

of the population that is urbanised by state. Then enrich the chart by adding

assault

arrest rates (by colouring the points from blue (low) to red (high)).

        c. Draw a bar graph to show the murder rate for each of the 50 states .

Code:

(i)

data("USArrests")

dim(USArrests)

summary(USArrests)

(ii)

states <- rownames(USArrests)

states[which.max(USArrests$Rape)]

(iii)

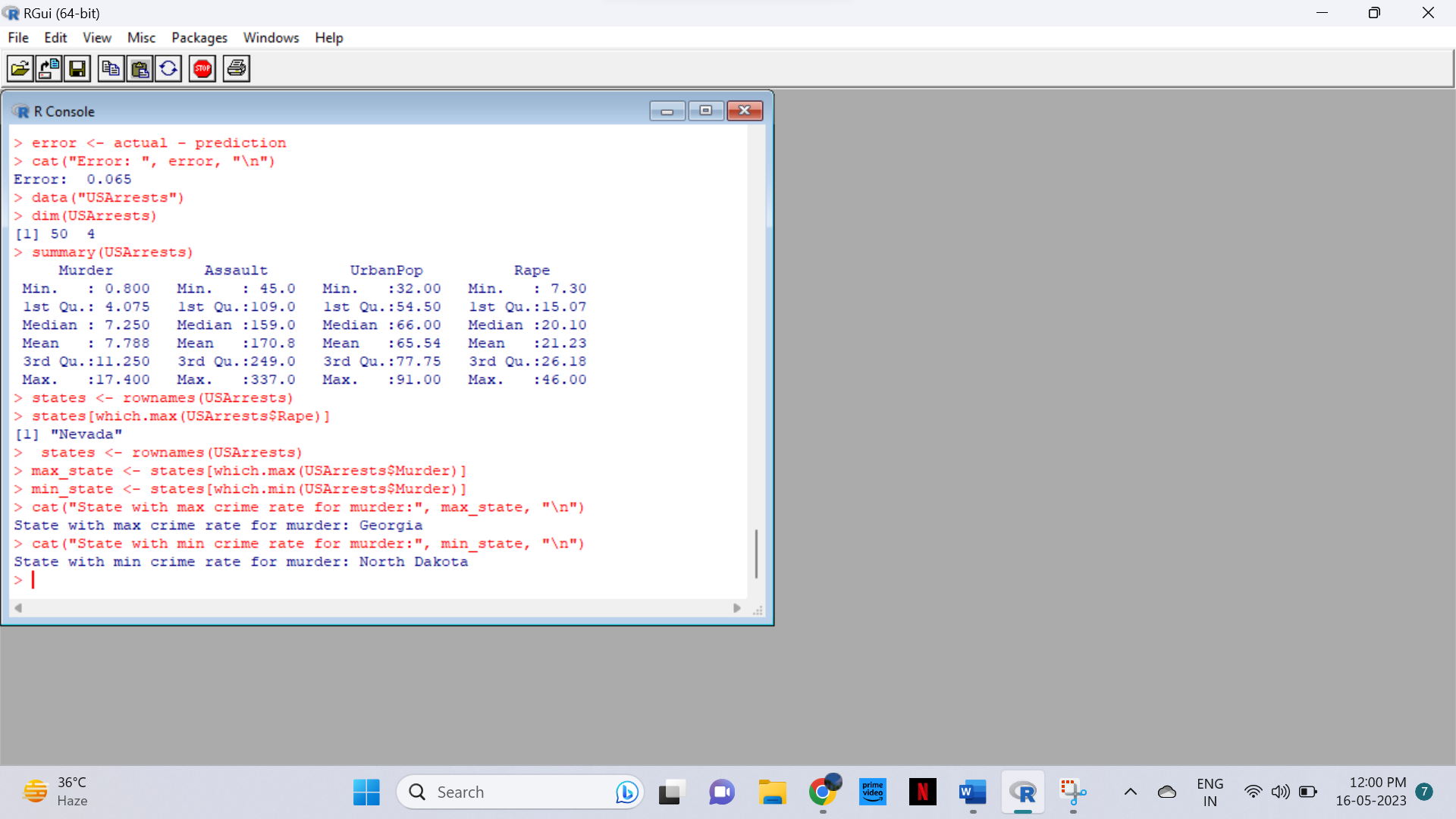
states <- rownames(USArrests)

max\_state <- states[which.max(USArrests$Murder)]

min\_state <- states[which.min(USArrests$Murder)]

cat("State with max crime rate for murder:", max\_state, "\n")

cat("State with min crime rate for murder:", min\_state, "\n")



2(i)

cor(USArrests

(ii)

states <- rownames(USArrests)

median\_assault <- median(USArrests$Assault)

subset(states, USArrests$Assault > median\_assault)

iii)

states <- rownames(USArrests)

q25 <- quantile(USArrests$Murder,

3)

Output: # Load the USArrests dataset

data("USArrests")

# Create a histogram of murder arrests by state

hist(USArrests$Murder, main = "Murder Arrests by US State")

# Create a density plot of murder arrests by state

plot(density(USArrests$Murder), main = "Density Plot of Murder Arrests by US State")

# Load the ggplot2 package

library(ggplot2)

# Create a scatter plot of murder arrest rate vs. proportion urban

ggplot(USArrests, aes(x = UrbanPop, y = Murder, color = Assault)) +

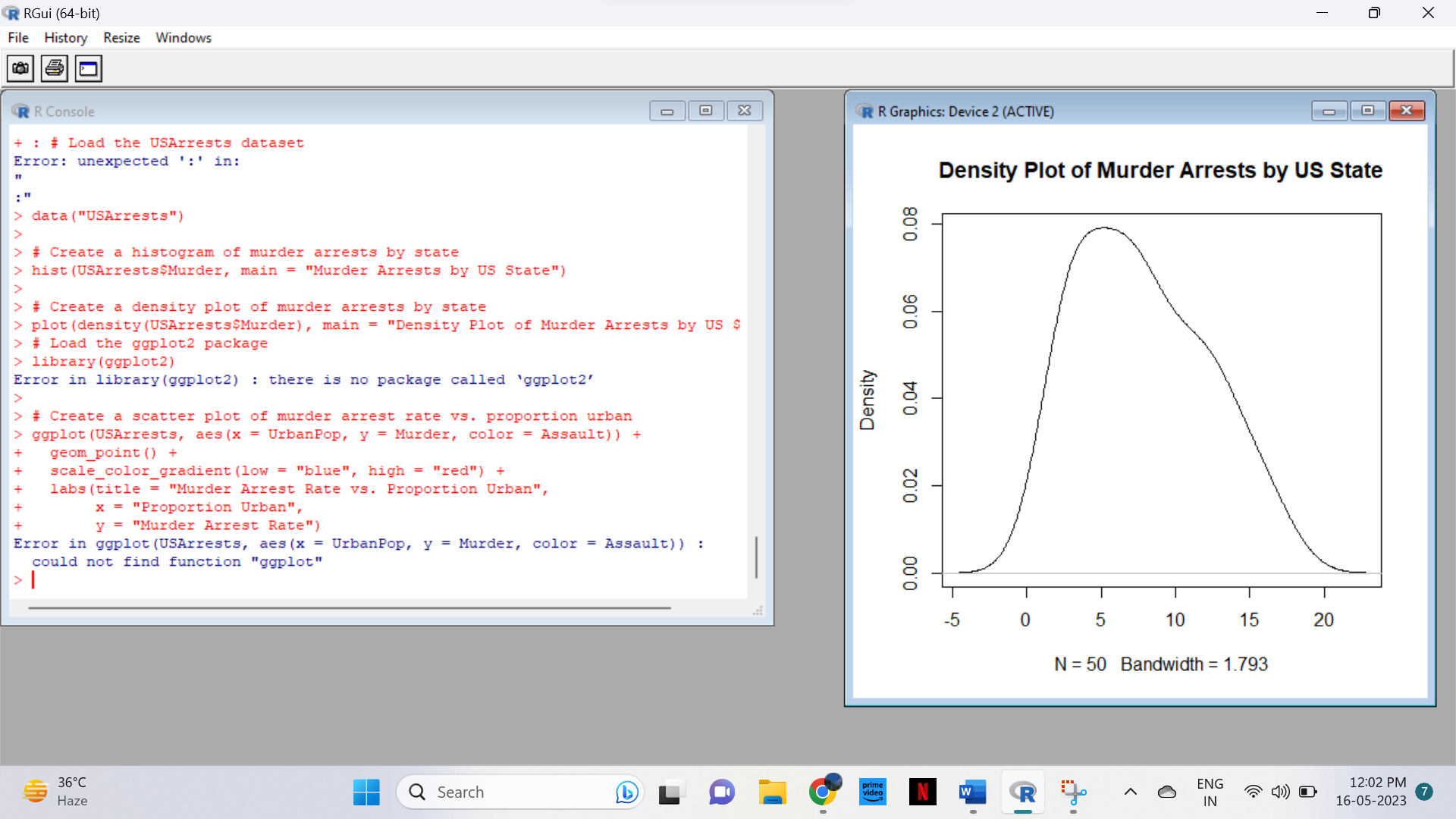
geom\_point() +

scale\_color\_gradient(low = "blue", high = "red") +

labs(title = "Murder Arrest Rate vs. Proportion Urban",

x = "Proportion Urban",

y = "Murder Arrest Rate")



4. a. Create a data frame based on below table.

Month 1 2 3 4 5 6 7 8 9 10 11 12

Spend

s

100

0 4000 5000 4500 3000 4000 9000 11000 15000 12000 7000 3000

Sales 991

4

4048

7

5432

4

5004

4

3471

9

4255

1

9487

1 118914 15848

4

13134

8

7850

4

3628

4

b. Create a regression model for that data frame table to show the amount of sales(Sales) based

on the how much the company spends (Spends) in advertising

c. Predict the Sales if Spend=13500

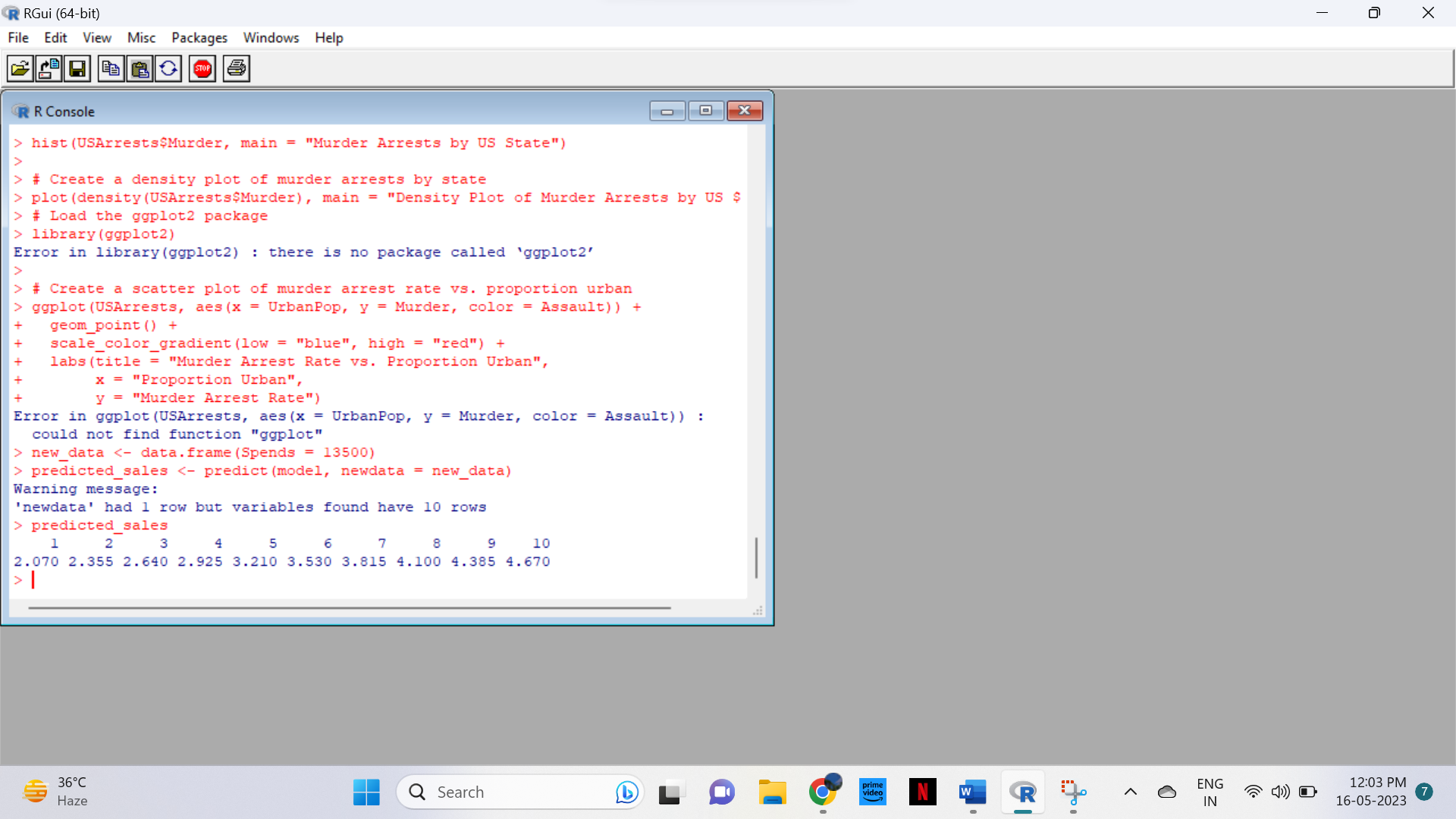
Code:

new\_data <- data.frame(Spends = 13500)

predicted\_sales <- predict(model, newdata = new\_data)

predicted\_sales

Output:



5.(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&lt;-(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&lt;- “matrix”,  output&lt;-2

Code:

(i)

# Create a random sample of 20 letters from LETTERS

set.seed(123)

letters\_sample <- sample(LETTERS, 20, replace = TRUE)

# Convert the sample into a factor

letters\_factor <- factor(letters\_sample)

levels(letters\_factor)[1:5]

(ii)

find\_range <- function(vec) {

range <- max(vec) - min(vec)

return(range)

}

# Test the function with the given input

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

find\_range(C)

(iii)

count\_vowels <- function(str) {

str <- tolower(str)

chars <- strsplit(str, "")[[1]]

# Count the number of vowels

num\_vowels <- sum(chars %in% c("a", "e", "i", "o", "u"))

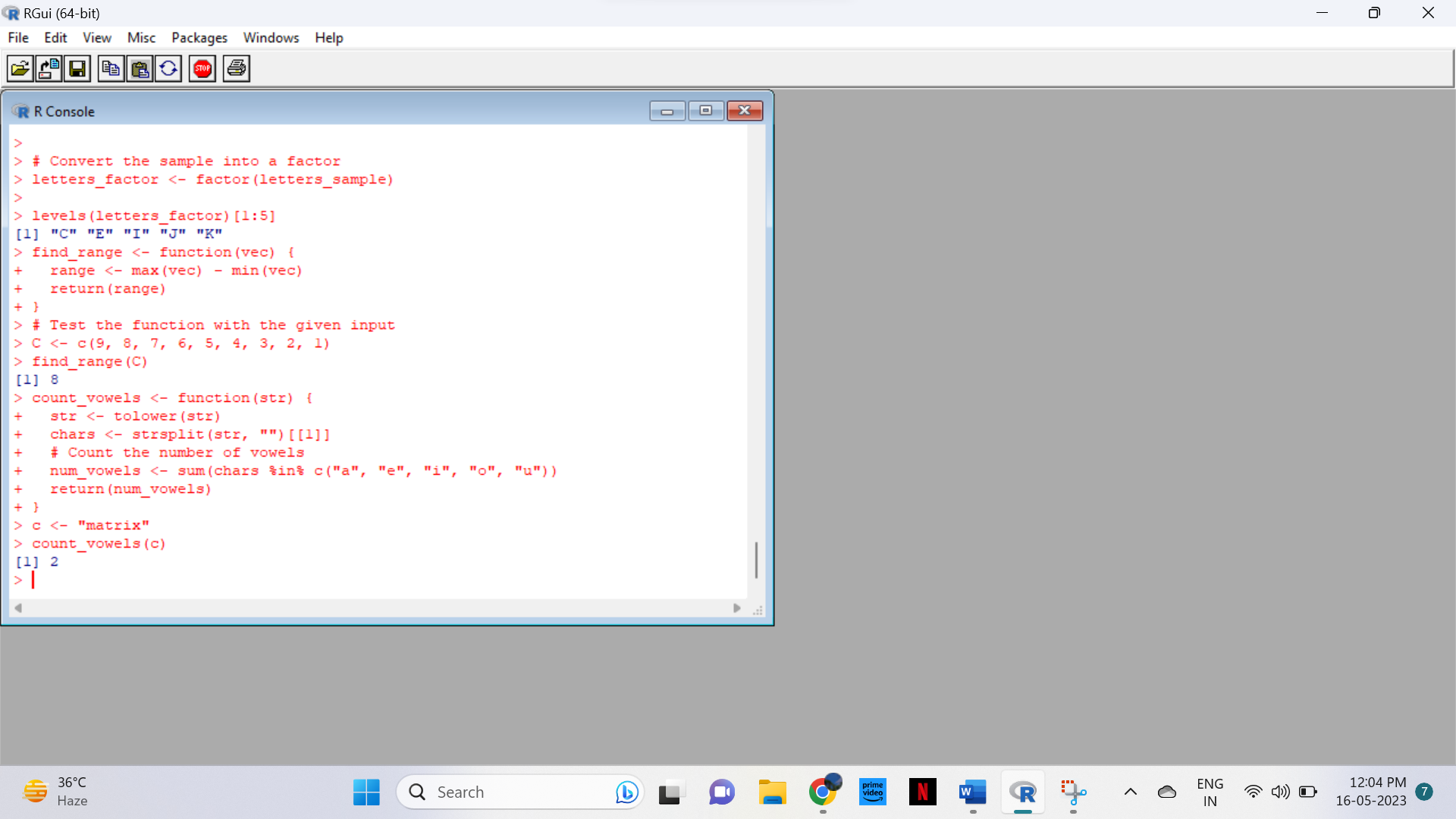
return(num\_vowels)

}

c <- "matrix"

count\_vowels(c)

Output:



6.Load inbuild dataset “ChickWeight” in R

(i) Explore the summary of Data set, like number of Features and its type. Fins 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&quot;,&quot;Time&quot;,&quot;Diet&quot;   features as ID variables

(v)Perform cast function to display the mean value of weight grouped by Diet

Code:

# Create the dataset

chickenWeight <- data.frame(

ChickenID = c(1, 2, 3, 4, 5), # Unique identifier for each chicken

Weight = c(2.3, 2.5, 2.1, 2.4, 2.2), # Weight of the chickens in kilograms

Diet = c("A", "B", "B", "A", "C"), # Diet of the chickens

Age = c(10, 12, 11, 10, 9) # Age of the chickens in weeks

)

data(ChickWeight)

# Number of features and their types

summary(ChickWeight)

# Number of records for each feature

table(ChickWeight$weight, useNA = "ifany")

table(ChickWeight$Time, useNA = "ifany")

table(ChickWeight$Chick, useNA = "ifany")

table(ChickWeight$Diet, useNA = "ifany")

last\_six <- tail(ChickWeight, 6)

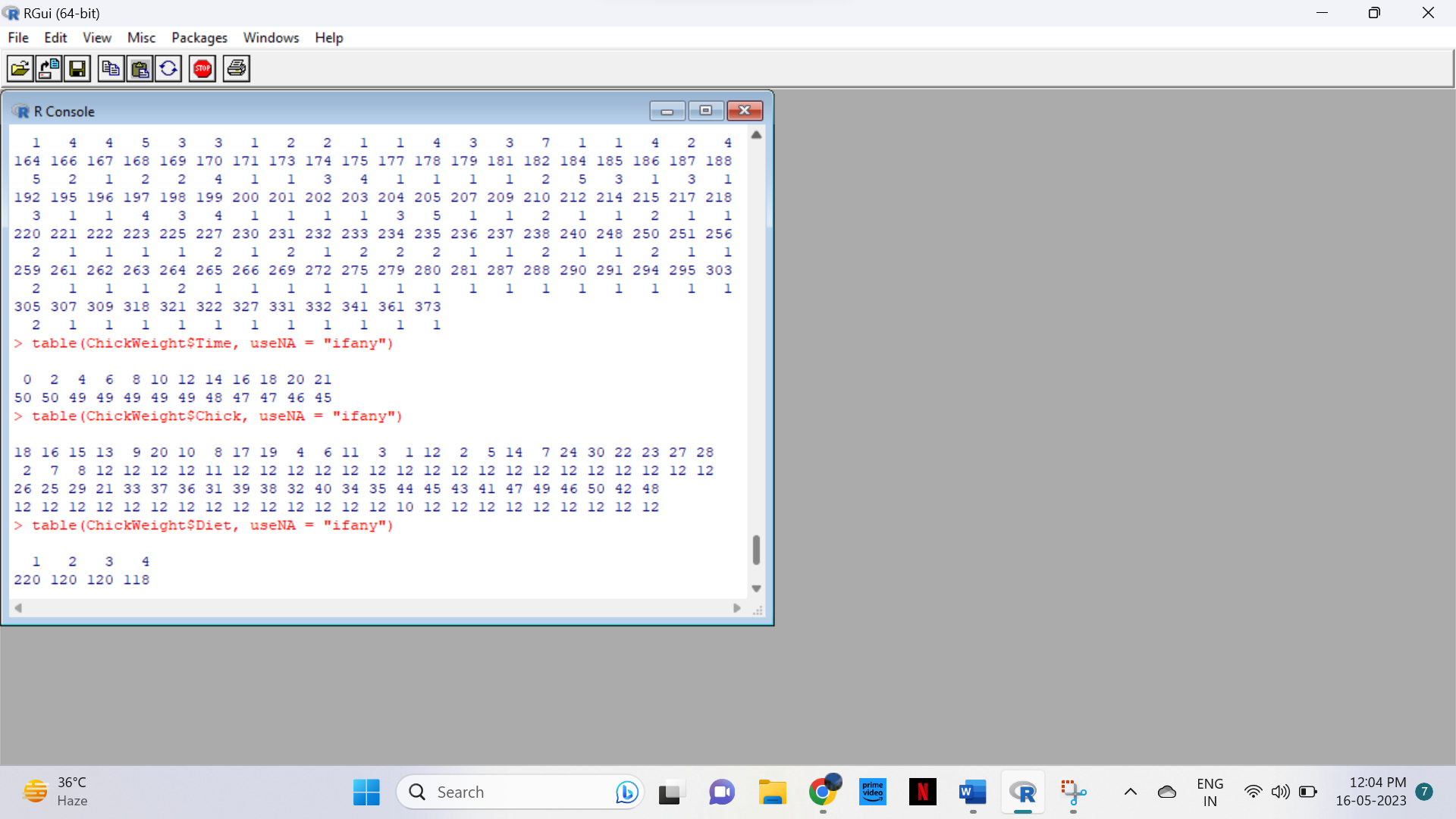
ordered\_df <- ChickWeight[order(ChickWeight$weight), ]

library(reshape2)

melted\_df <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))

cast\_df <- dcast(melted\_df, Diet ~ variable, mean)

Output:



7.(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

Code:

# Create the dataset

chickenWeight <- data.frame(

ChickenID = c(1, 2, 3, 4, 5), # Unique identifier for each chicken

Weight = c(2.3, 2.5, 2.1, 2.4, 2.2), # Weight of the chickens in kilograms

Diet = c("A", "B", "B", "A", "C"), # Diet of the chickens

Age = c(10, 12, 11, 10, 9) # Age of the chickens in weeks

)

summary(ChickWeight)

boxplot(weight ~ Diet, data = ChickWeight, main = "Weight by Diet", xlab = "Diet", ylab = "Weight")

hist(ChickWeight$weight[ChickWeight$Diet == 1], main = "Histogram of Weight (Diet-1)", xlab = "Weight")

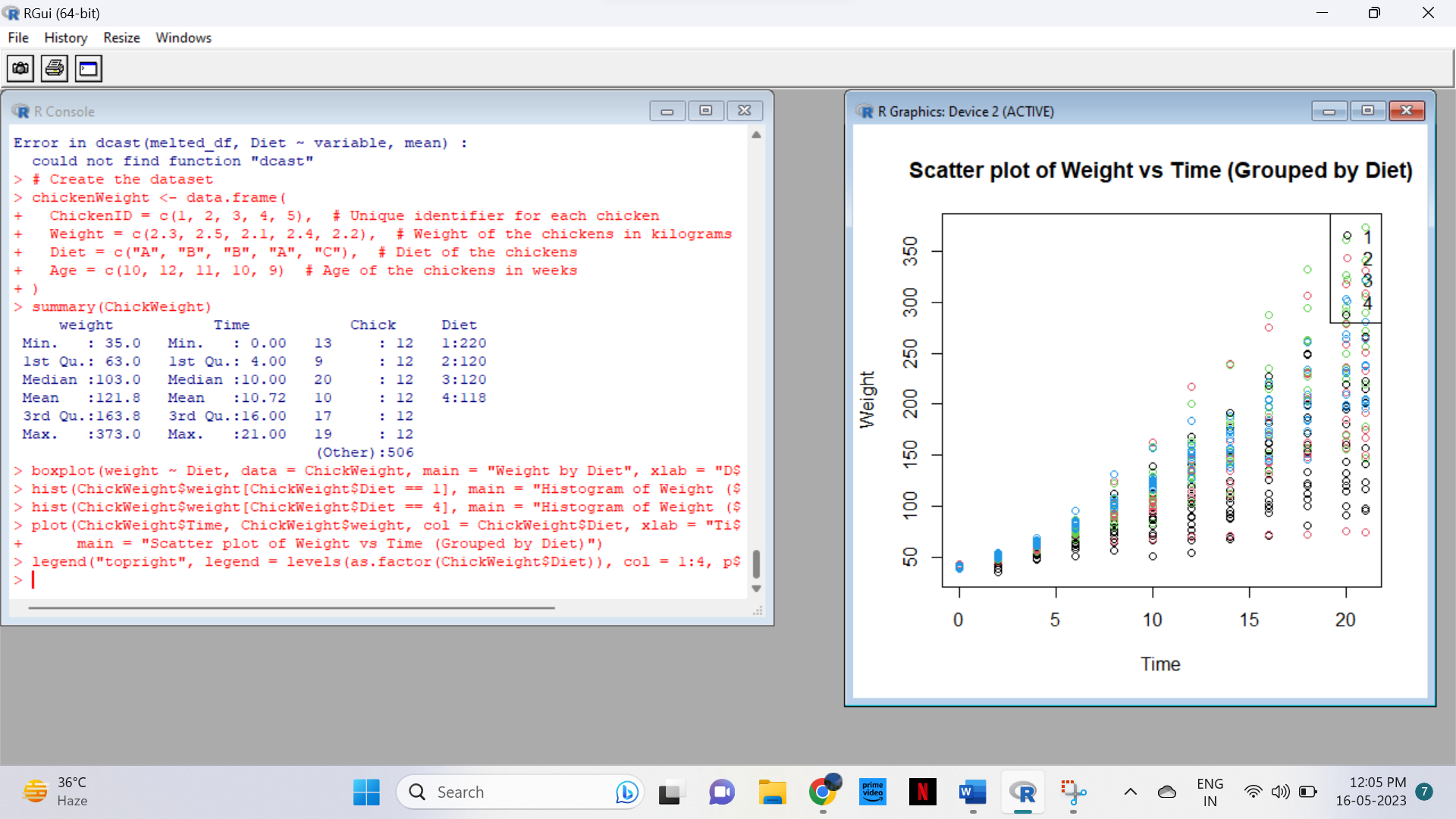
hist(ChickWeight$weight[ChickWeight$Diet == 4], main = "Histogram of Weight (Diet-4)", xlab = "Weight")

plot(ChickWeight$Time, ChickWeight$weight, col = ChickWeight$Diet, xlab = "Time", ylab = "Weight",

main = "Scatter plot of Weight vs Time (Grouped by Diet)")

legend("topright", legend = levels(as.factor(ChickWeight$Diet)), col = 1:4, pch = 1)

Output:

****

8.(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 in model for same

Code:

set.seed(123) # For reproducibility

train\_indices <- sample(1:nrow(chicken\_data), 0.7 \* nrow(chicken\_data))

train\_data <- chicken\_data[train\_indices, ]

test\_data <- chicken\_data[-train\_indices, ]

model <- lm(Weight ~ Time + Diet, data = train\_data)

predicted\_values <- predict(model, newdata = test\_data)

error <- sqrt(mean((test\_data$Weight - predicted\_values)^2))

new\_data <- data.frame(Time = 10, Diet = 1)

predicted\_weight <- predict(model, newdata = new\_data)

cat("Predicted weight for Time = 10 and Diet = 1:", predicted\_weight, "\n")

cat("Error in the model:", error, "\n")

Output:

