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:192110320

Name:U.mohan srinivas

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

SOURCE CODE:

a)

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

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

weight <- c(1.2, 1.8, 1.4, 2.1, 1.6, 2.2, 1.9, 2.4, 2.0, 2.6)

model <- lm(weight ~ time + diet)

summary(model)

b)

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

predict(model, newdata = new\_data)

c)

install.packages("caret")

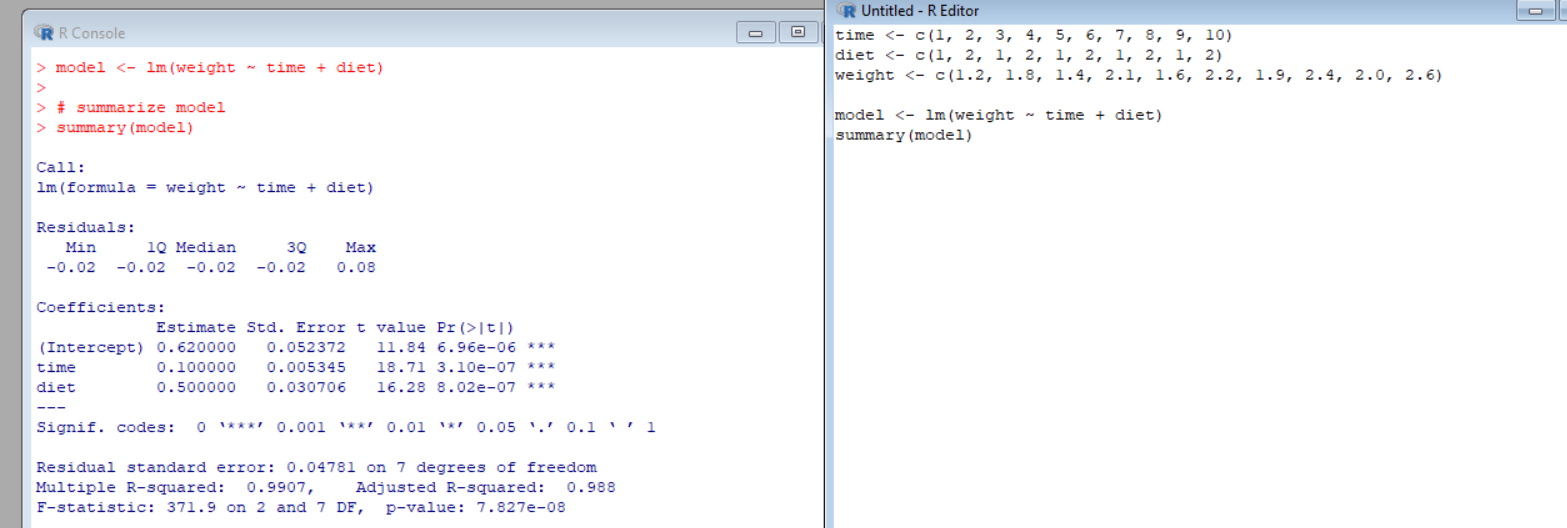
library(caret)

rmse <- sqrt(mean((model$fitted.values - weight)^2))

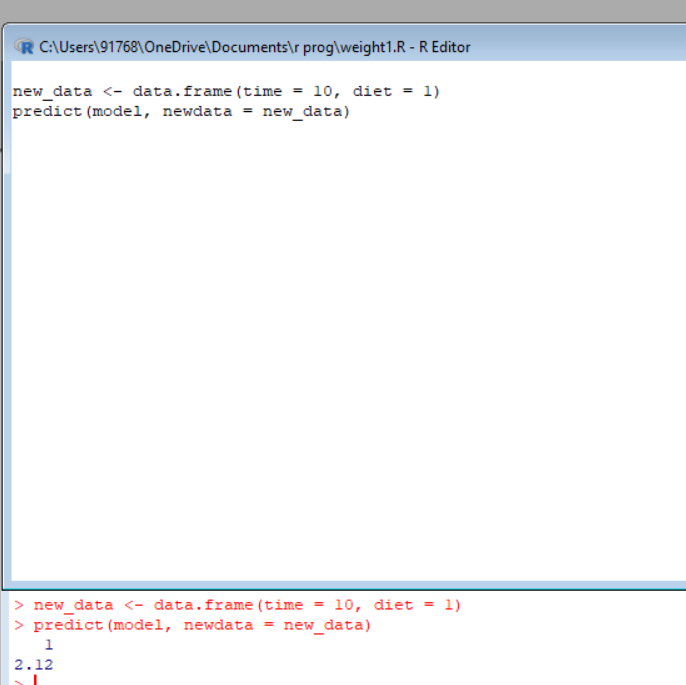
rmse

OUTPUT:

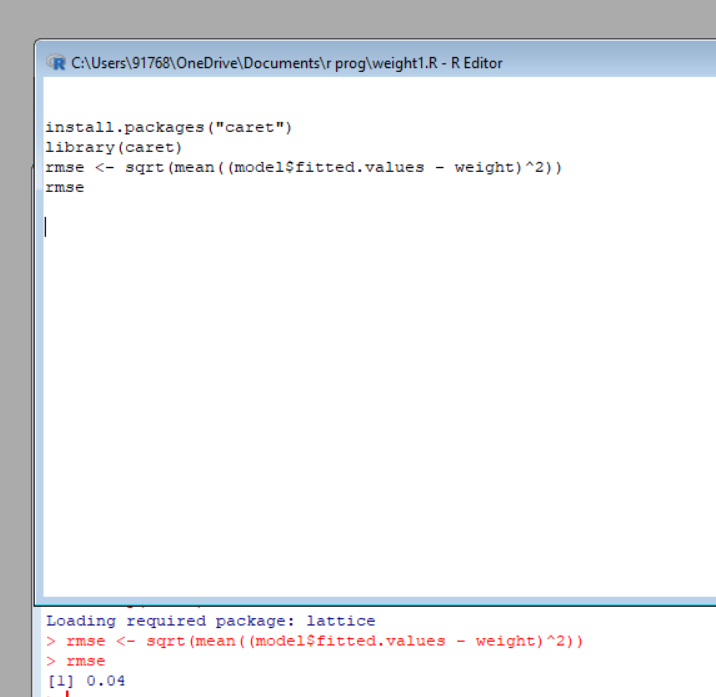
a)



b)



c)



2 ..For this exercise, use the (built-in) dataset Titanic.

a. Draw a Bar chart to show details of “Survived” on the Titanic based on passenger

Class

b. Modify the above plot based on gender of people who survived

c. Draw histogram plot to show distribution of feature “Age”

SOURCE CODE:

A)

data(Titanic)

survived\_by\_class <- aggregate(Freq ~ Class, data = Titanic, FUN = sum)

barplot(survived\_by\_class$Freq, names.arg = survived\_by\_class$Class,

xlab = "Passenger Class", ylab = "Number of Survivors",

main = "Survivors on the Titanic by Passenger Class",

col = "steelblue")

B)

survived\_by\_class\_gender <- aggregate(Freq ~ Class + Sex, data = Titanic, FUN = sum)

survived\_by\_class\_gender\_wide <- reshape(survived\_by\_class\_gender,

idvar = "Class",

timevar = "Sex",

direction = "wide")

barplot(height = t(survived\_by\_class\_gender\_wide[, -1]),

beside = TRUE,

names.arg = survived\_by\_class\_gender\_wide$Class,

legend.text = colnames(survived\_by\_class\_gender\_wide)[-1],

xlab = "Passenger Class",

ylab = "Number of Survivors",

main = "Survivors on the Titanic by Passenger Class and Gender",

col = c("steelblue", "darkorange"))

C)

data(Titanic)

titanic\_age <- Titanic[complete.cases(Titanic$Age), ]

hist(titanic\_age$Age,

breaks = 20,

xlab = "Age",

ylab = "Frequency",

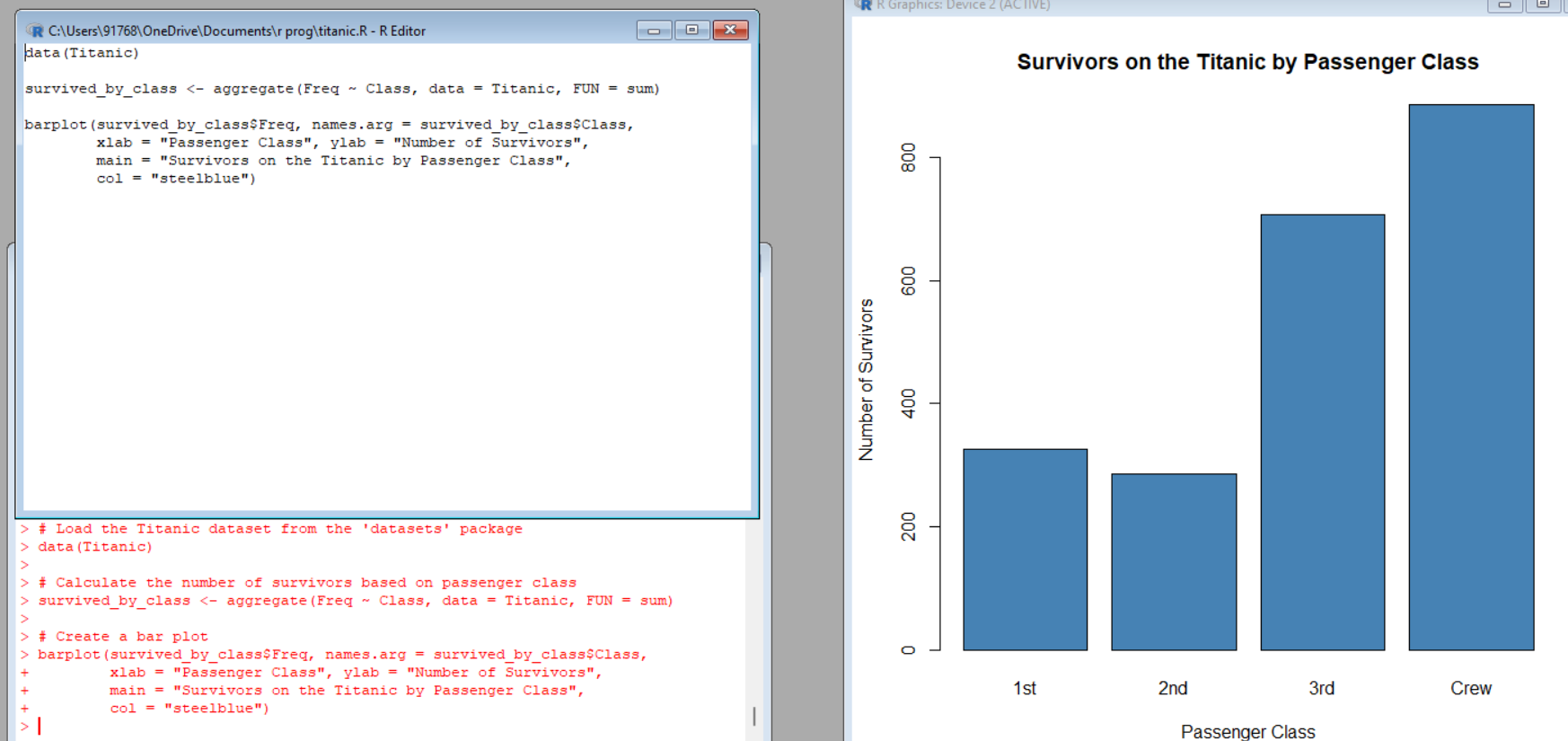
main = "Distribution of Age on the Titanic",

col = "steelblue",

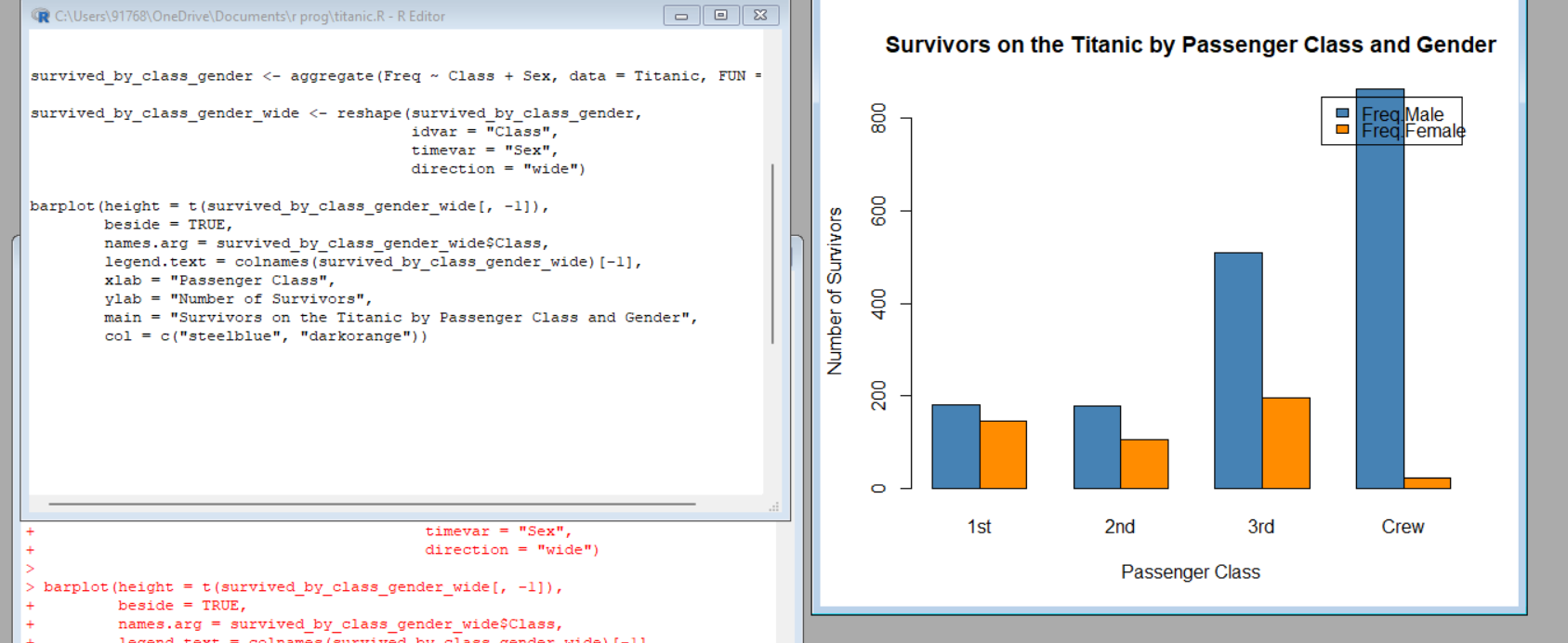
border = "white")

OUTPUT:

A)



B)



C)

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 .

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

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

SOURCE CODE:

a)

df <- data.frame(

Month = 1:12,

Spend = c(100, 0, 4000, 5000, 4500, 3000, 4000, 9000, 11000, 15000, 12000, 7000),

Sales = c(991, 4, 4048, 7, 5432, 4, 5004, 4, 3471, 9, 4255, 1)

)

df

b)

model <- lm(Sales ~ Spend, data = df)

summary(model)

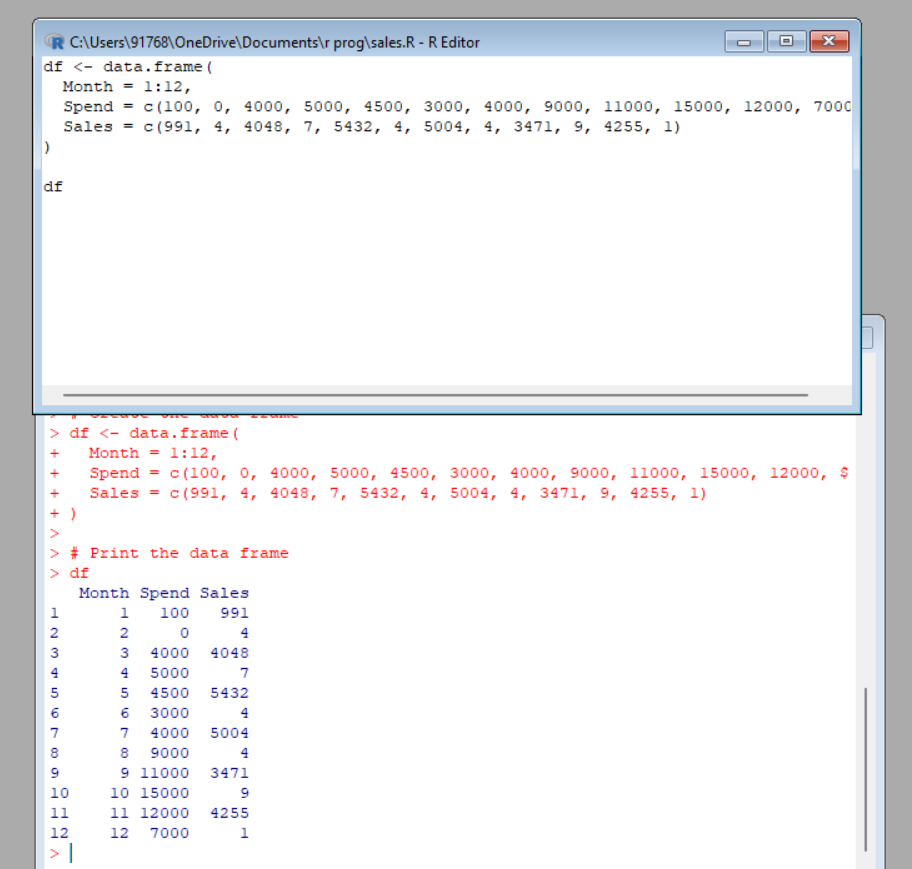
c)

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

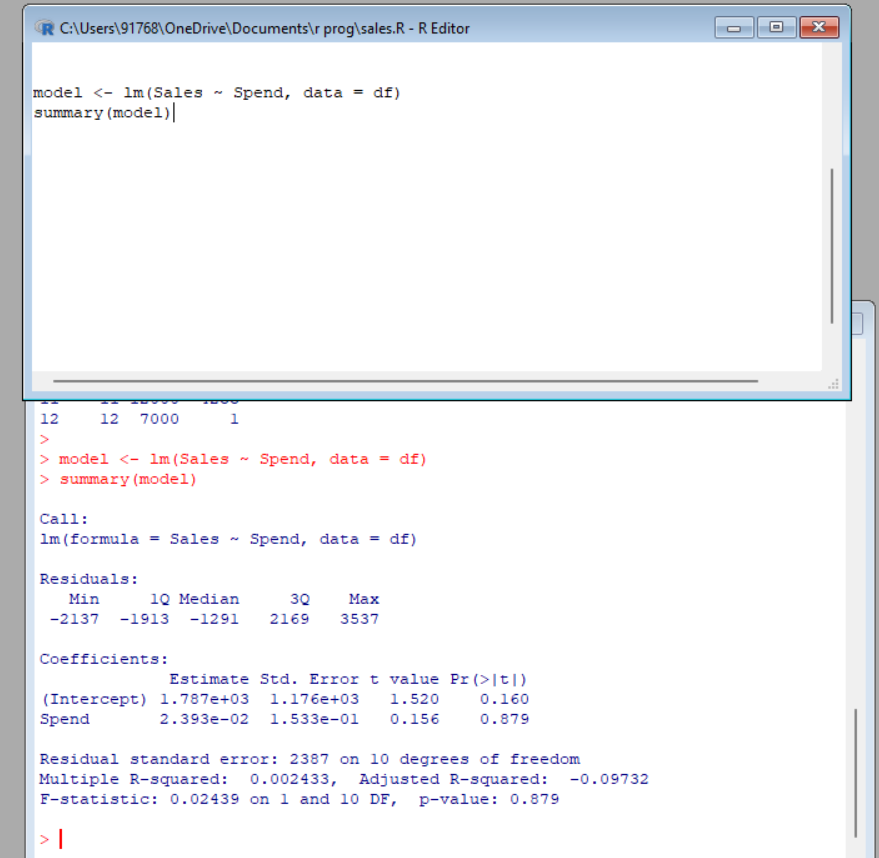
predicted\_sales <- predict(model, newdata = new\_data)

predicted\_sales

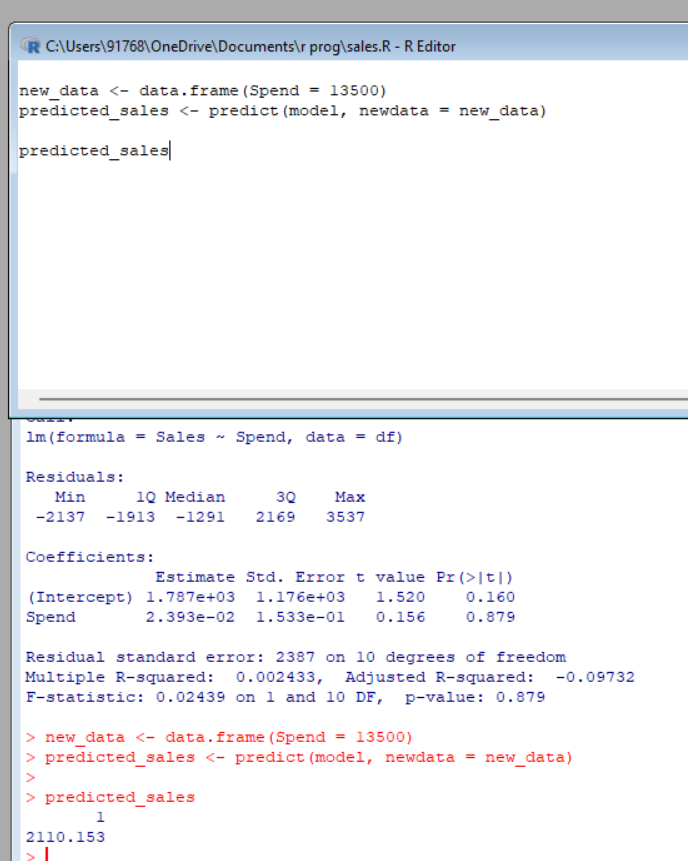
OUTPUT:



b)



c)



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

SOUCE CODE:

i)

data("LETTERS")

sample\_letters <- sample(LETTERS, 10)

sample\_factor <- as.factor(sample\_letters)

five\_levels <- levels(sample\_factor)[1:5]

print(five\_levels)

ii)

find\_range <- function(vec) {

max\_val <- max(vec)

min\_val <- min(vec)

range\_val <- max\_val - min\_val

return(range\_val)

}

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

result <- find\_range(C)

print(result)

iii)

count\_vowels <- function(str) {

str\_lower <- tolower(str)

vowels <- c("a", "e", "i", "o", "u")

vowel\_count <- sum(strsplit(str\_lower, "")[[1]] %in% vowels)

return(vowel\_count)

}

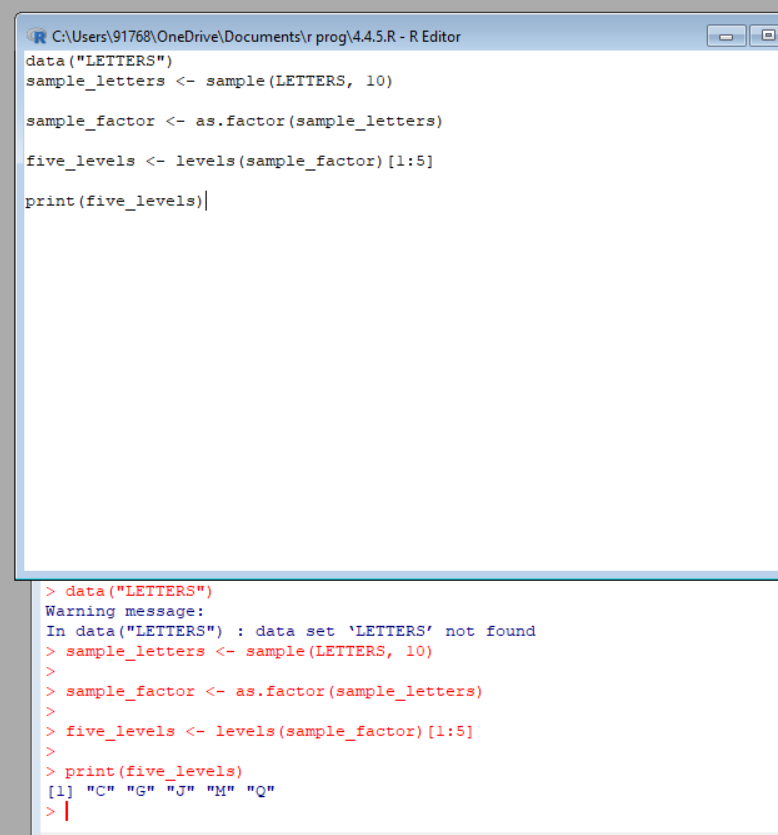
input\_str <- "matrix"

result <- count\_vowels(input\_str)

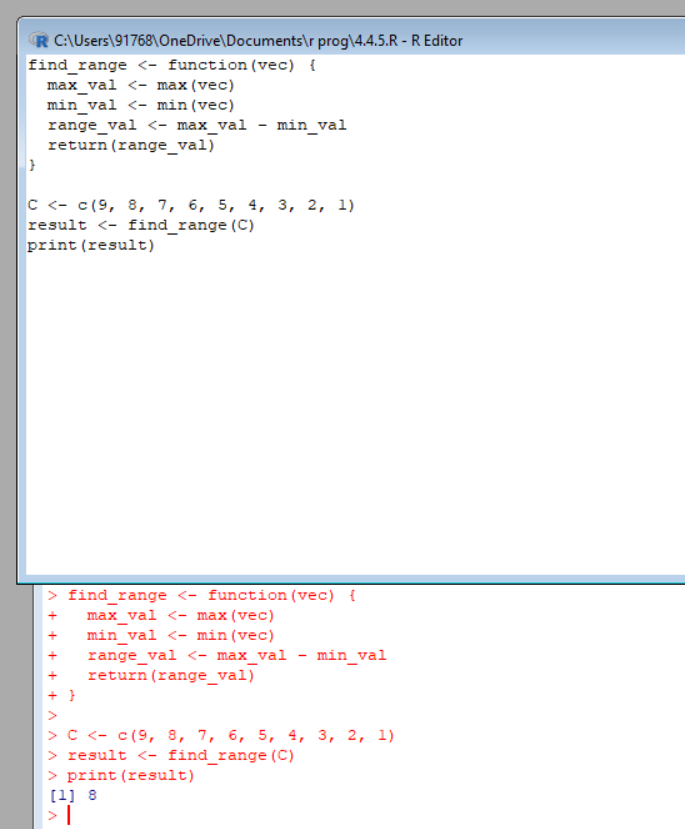
print(result)

OUTPUT:

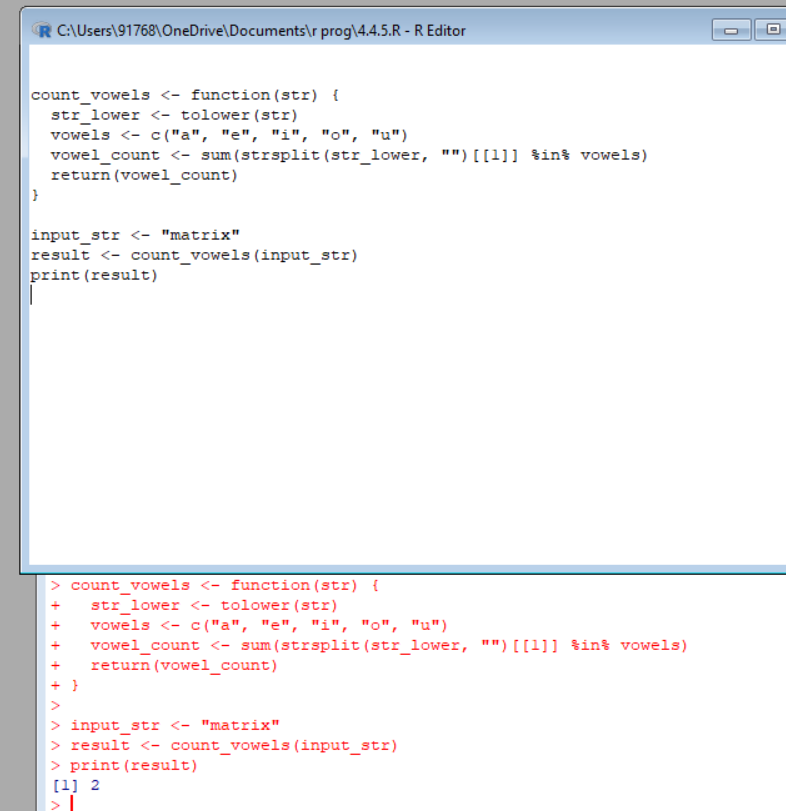
i)



ii)



iii)



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

SOURCE CODE:

i)

data(ChickWeight)

summary(ChickWeight)

sapply(ChickWeight, length)

ii)

tail(ChickWeight, 6)

iii)

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

ordered\_data

iv)

library(reshape2)

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

melted\_data

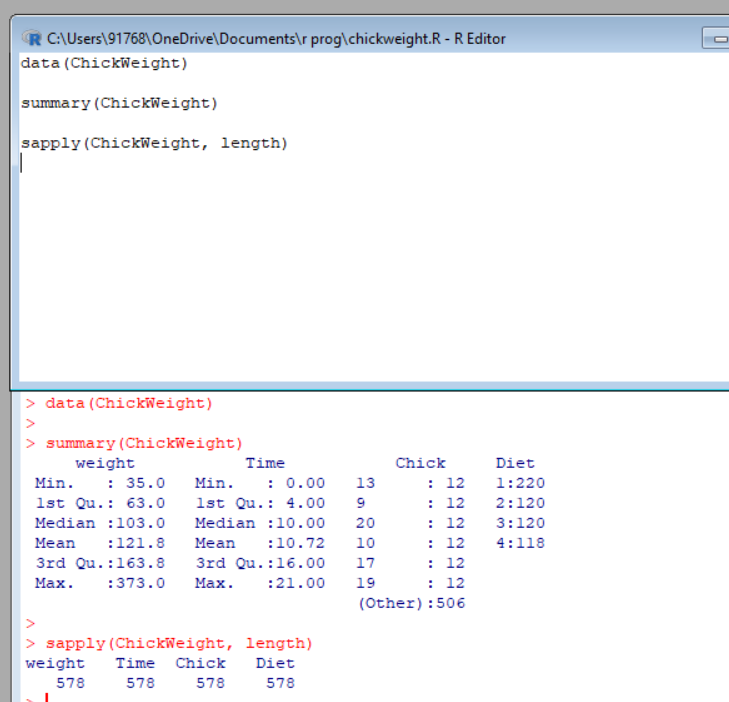
v)

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

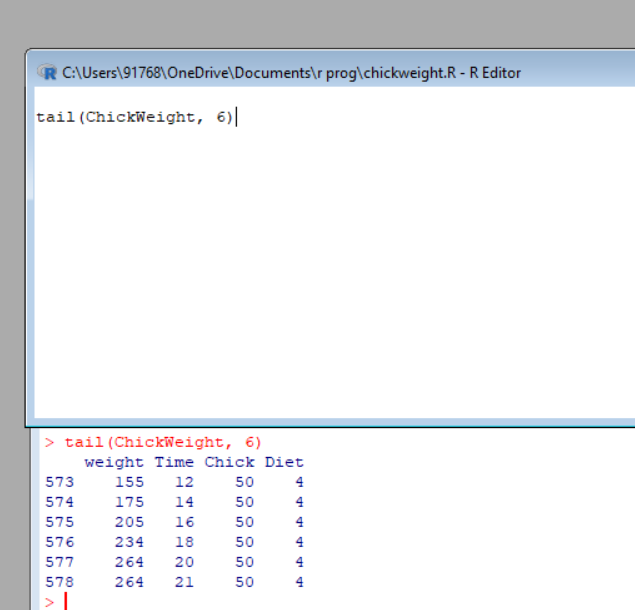
cast\_data

OUTPUT:

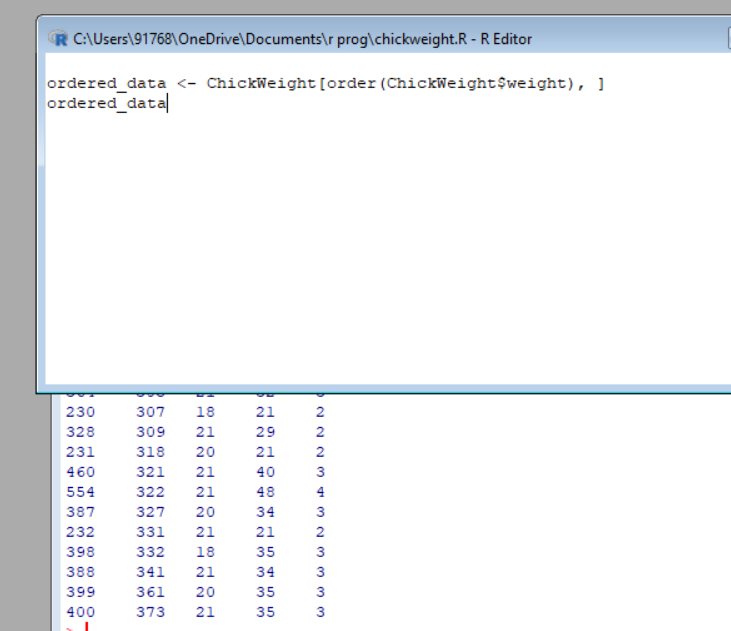
i)



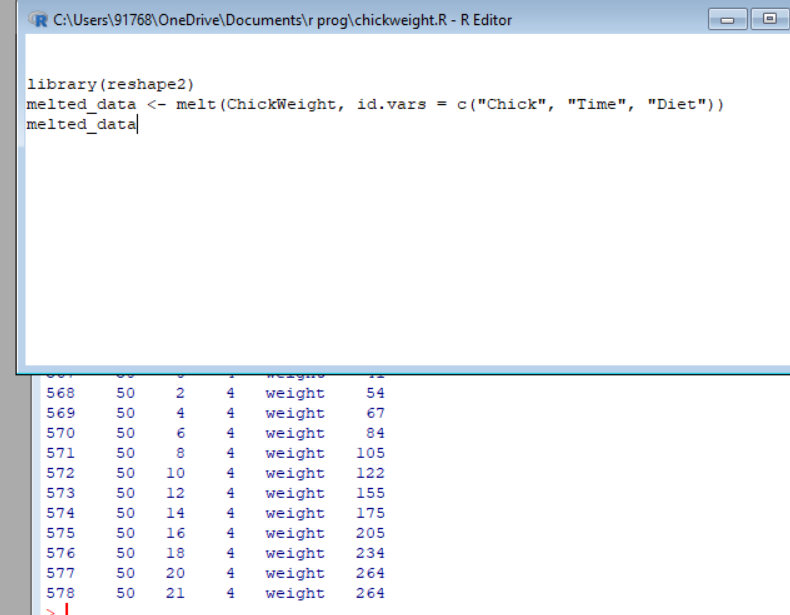
ii)



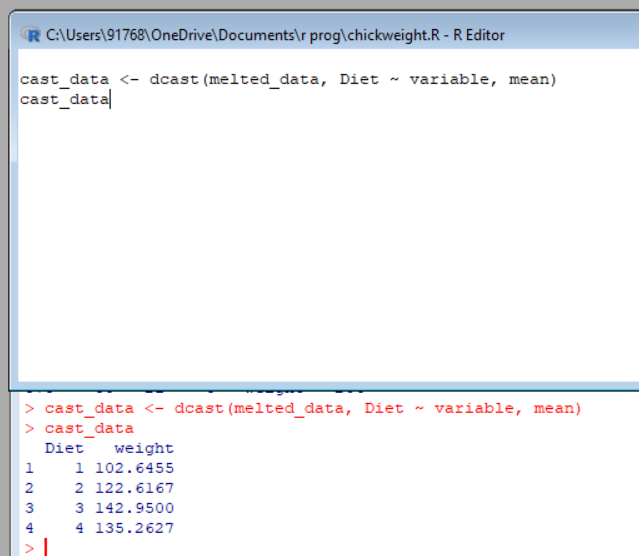
iii)



iv)



v)



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

SOURCE CODE:

i)

data(ChickWeight)

summary(ChickWeight)

ii)

data(ChickWeight)

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

iii)

data(ChickWeight)

diet1\_data <- subset(ChickWeight, Diet == 1)

hist(diet1\_data$weight, main = "Histogram of Weight (Diet-1)", xlab = "Weight", ylab = "Frequency")

iv)

data(ChickWeight)

diet4\_data <- subset(ChickWeight, Diet == 4)

hist(diet4\_data$weight, main = "Histogram of Weight (Diet-4)", xlab = "Weight", ylab = "Frequency")

v)

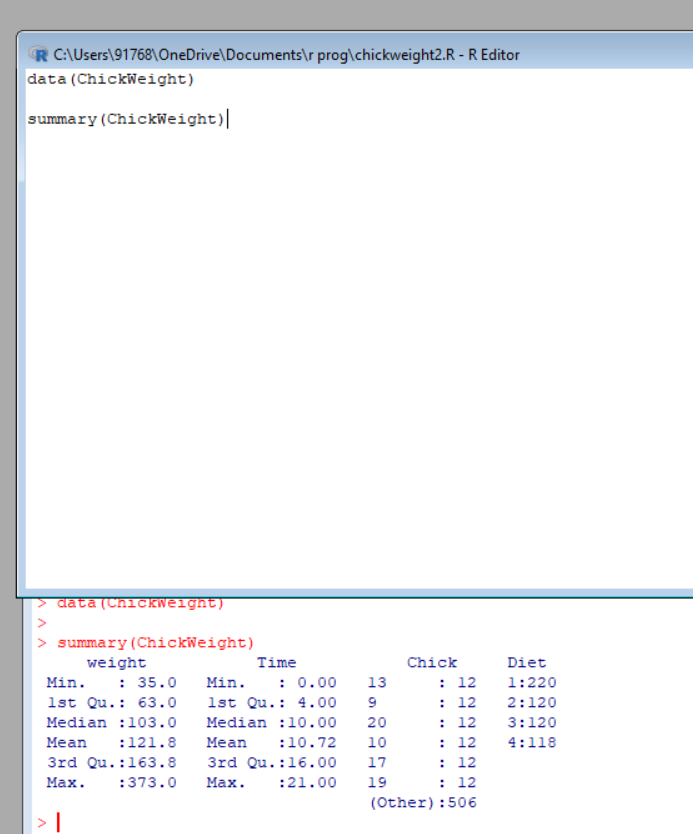
data(ChickWeight)

plot(weight ~ Time, data = ChickWeight, col = Diet, xlab = "Time", ylab = "Weight")

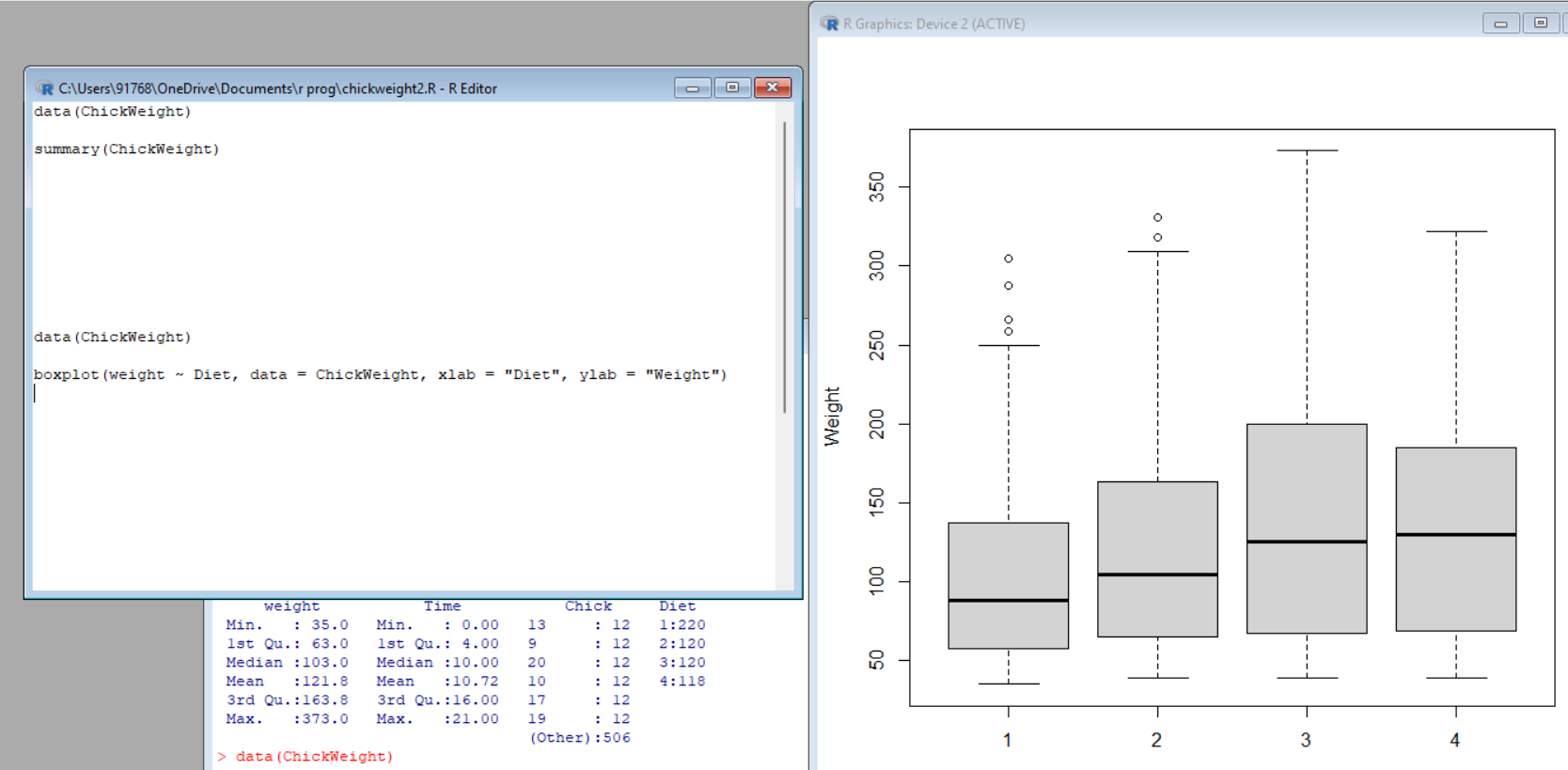
legend("topright", legend = levels(as.factor(ChickWeight$Diet)), col = 1:4, pch = 1, title = "Diet")

OUTPUT:

i)



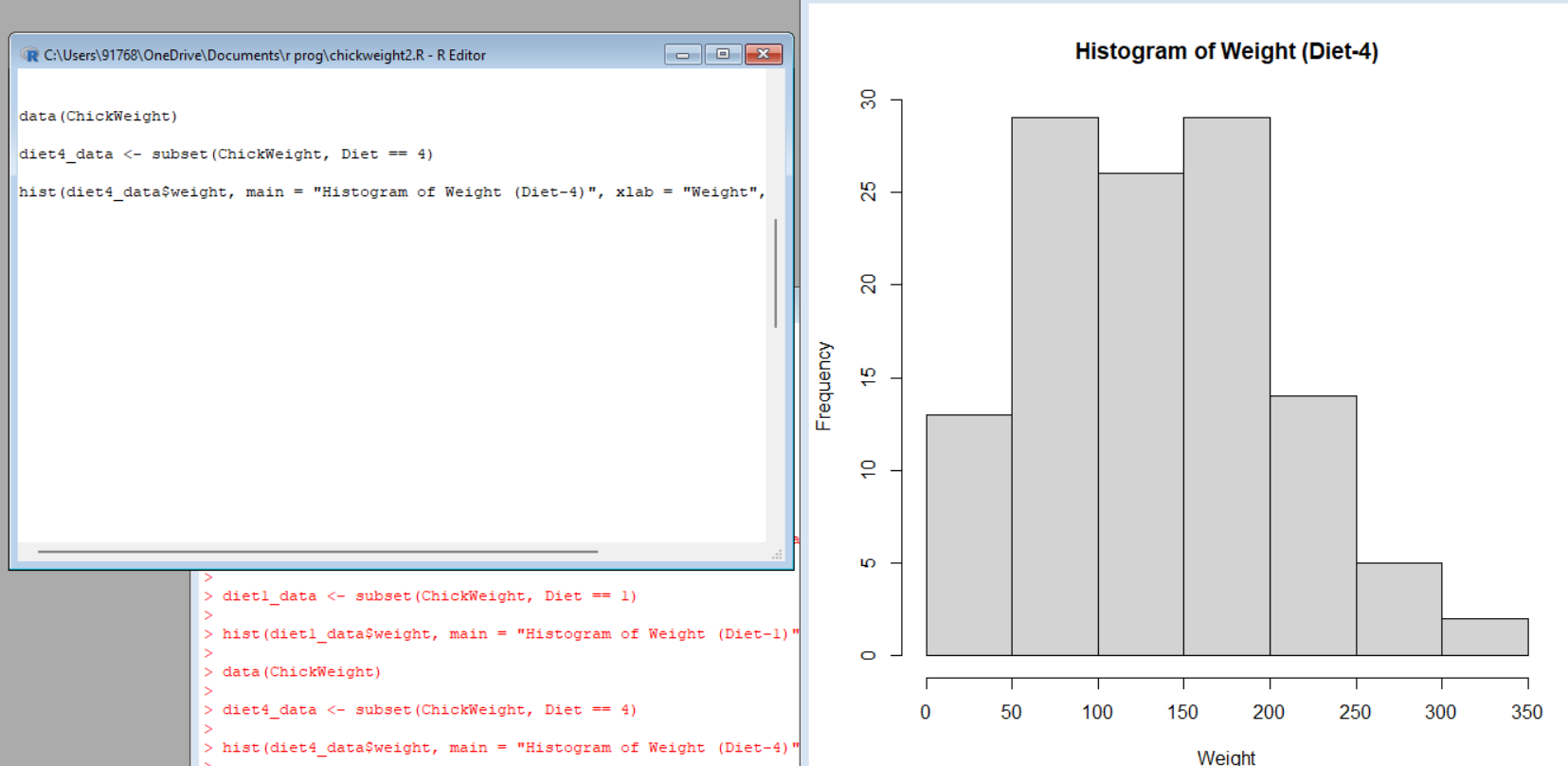
ii)



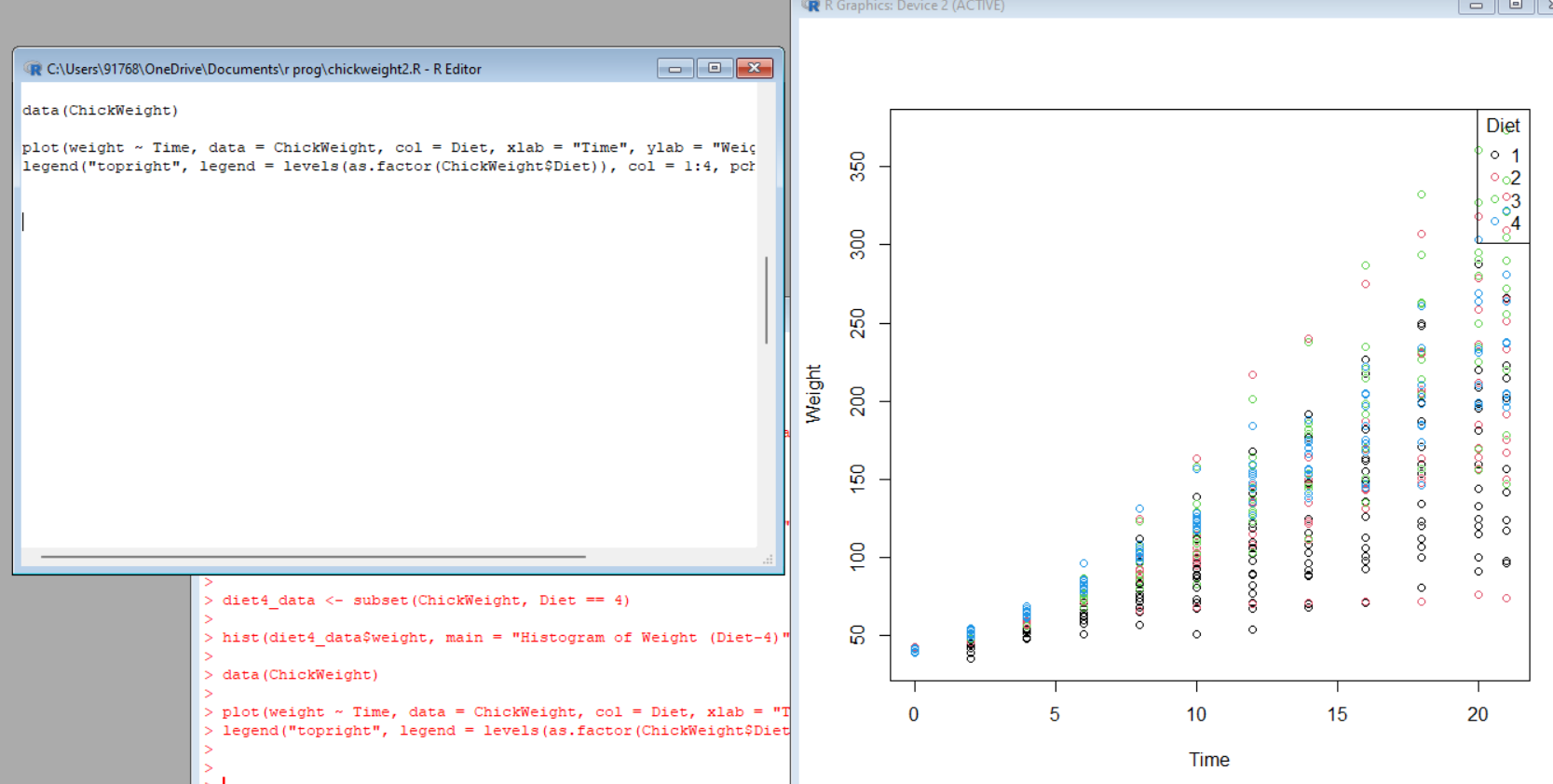
iii)



iv)



v)



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

SOURCE CODE:

i)

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

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

Weight <- c(2.1, 2.5, 2.4, 2.8, 2.3, 2.6, 2.5, 2.9, 2.7, 3.1)

data <- data.frame(Time, Diet, Weight)

model <- lm(Weight ~ Time + Diet, data)

summary(model)

ii)

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

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

predicted\_weight

iii)

predictions <- predict(model)

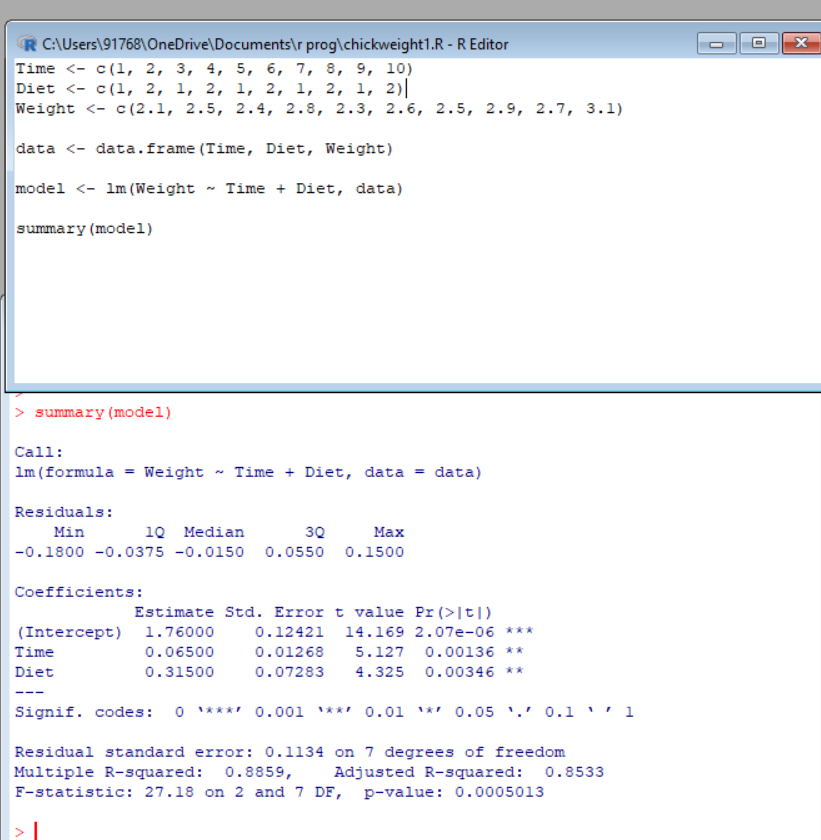
error <- predictions - data$Weight

rse <- sqrt(mean(error^2))

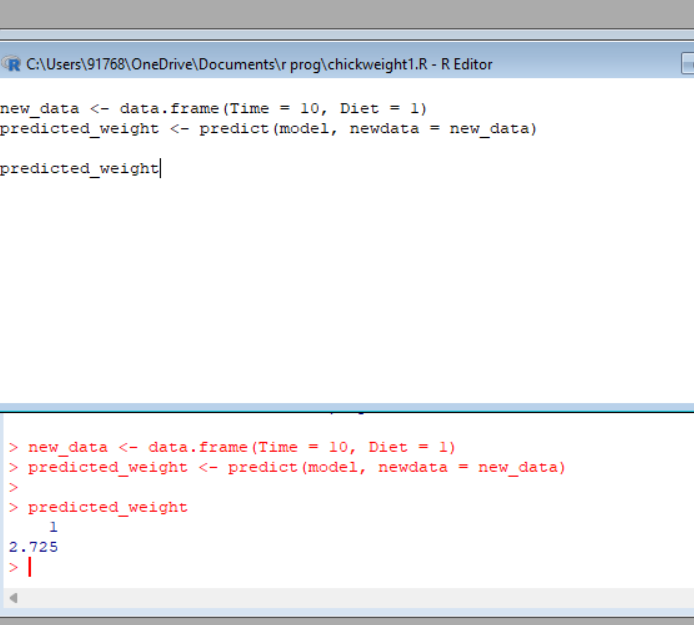
rse

OUTPUT:

i)



ii)



iii)

