SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

ITA 0451 - STATISTICS WITH R PROGRAMMING

DAY 4 – LAB ASSESSMENT Part 3

Reg No: 192110320

Name: u.mohan srinivas

1.Randomly Sample the iris dataset such as 80% data for training and 20% for test and

create Logistics regression with train data, use species as target and petals width and

length as feature variables , Predict the probability of the model using test data, Create

Confusion matrix for above test model

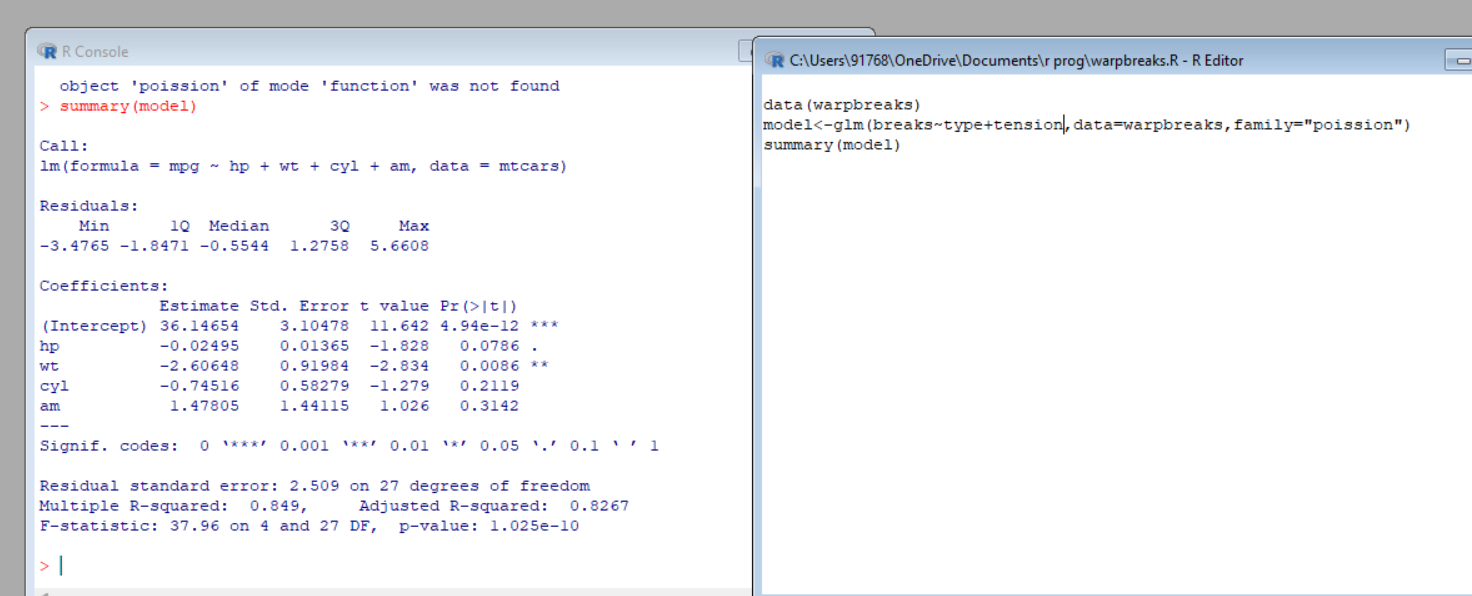
SOURCE CODE:

data(warpbreaks)

model<-glm(breaks~type+tension,data=warpbreaks,family="poission")

summary(model)

OUTPUT:



2. (i)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)

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

SOURCE CODE:

i)

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

mean\_value <- mean(values)

mean\_value

median\_value <- median(values)

median\_value

mode\_value <- names(table(values))[table(values) == max(table(values))]

mode\_value

ii)

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

second\_highest <- sort(unique(values), decreasing = TRUE)[2]

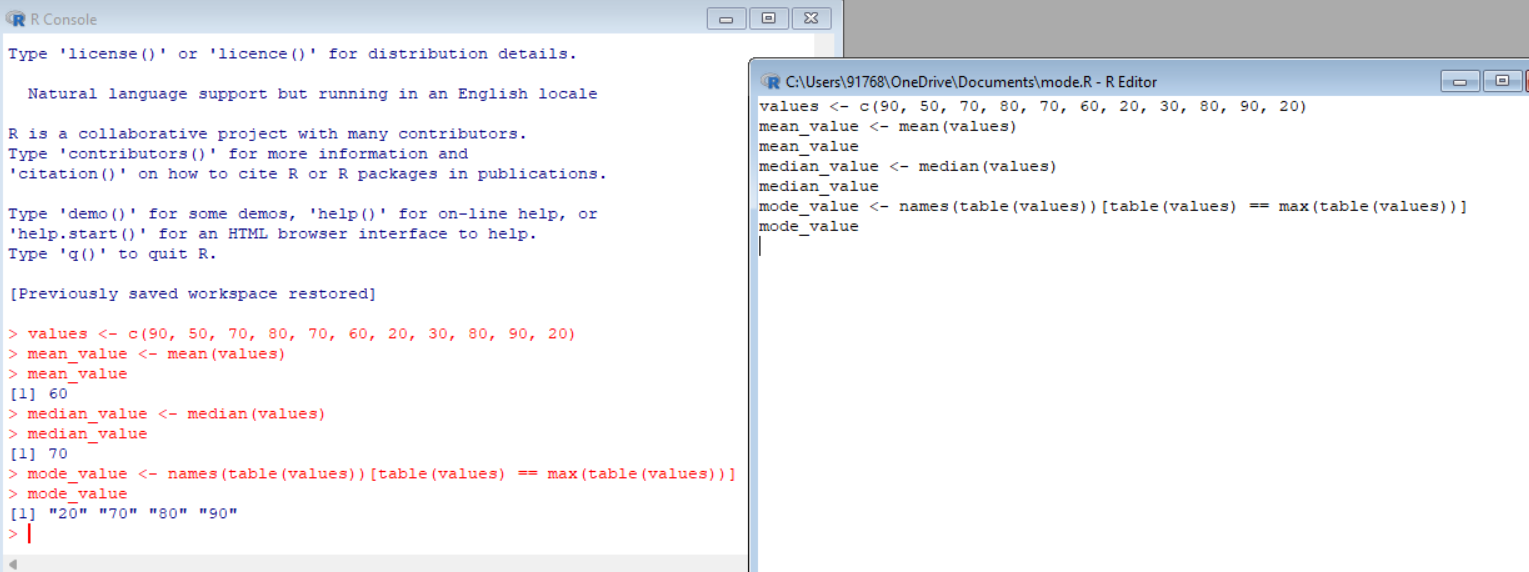
second\_highest

third\_lowest <- sort(unique(values))[3]

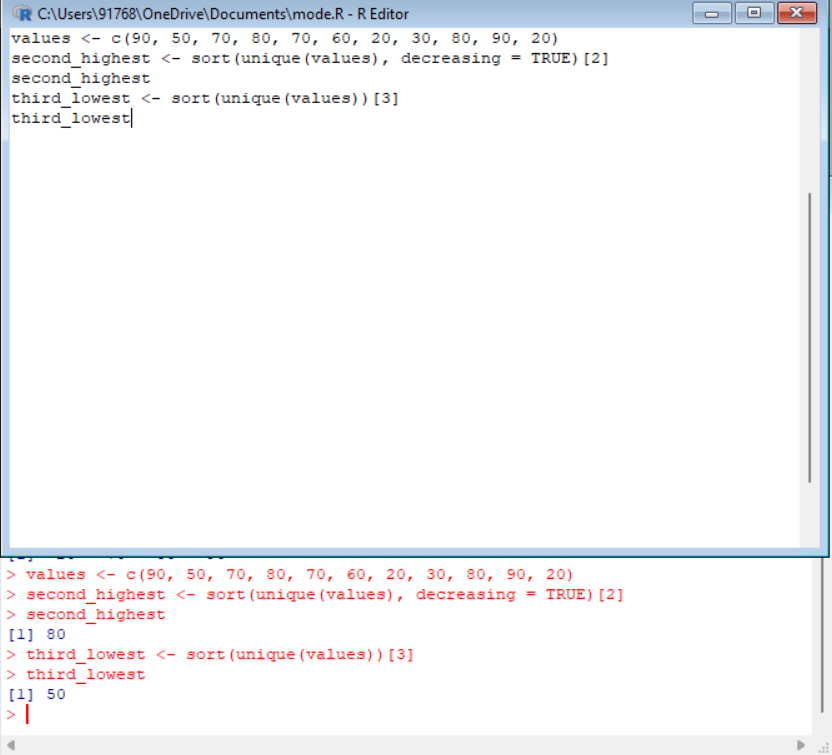
third\_lowest

OUTPUT:

i)



ii)



3. 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 -4).

SOURCE CODE :

ii)

str(airquality)

summary(airquality)

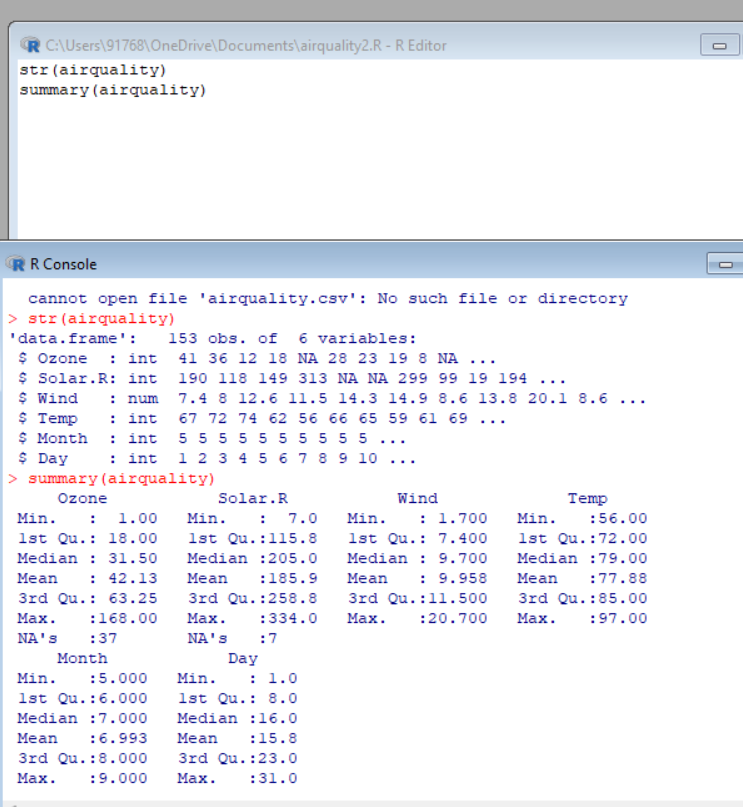
iii)

head(airquality)

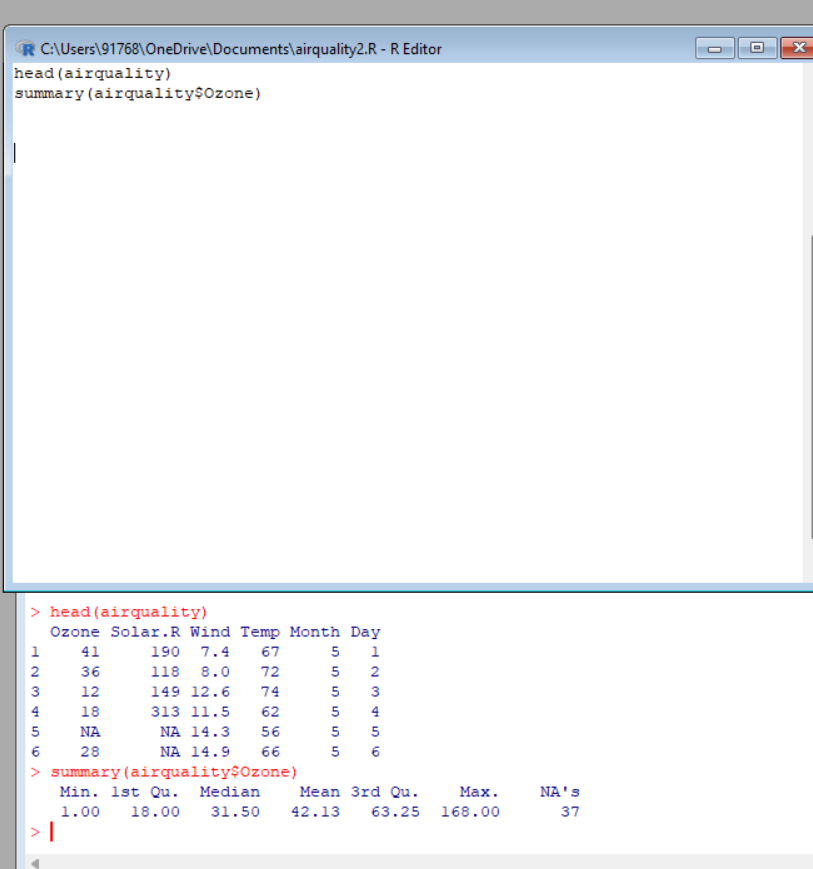
summary(airquality$Ozone)

OUTPUT:

II)



iii)



4)

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?

SOURCE CODE:

i)

data(airquality)

mean\_temp <- sum(airquality$Temp) / length(airquality$Temp)

mean\_temp

ii)

data(airquality)

first\_five\_rows <- airquality[1:5, ]

first\_five\_rows

iii)

data(airquality)

selected\_columns <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]

selected\_columns

Iv)

data(airquality)

coldest\_day <- airquality[which.min(airquality$Temp), ]

coldest\_day

v)

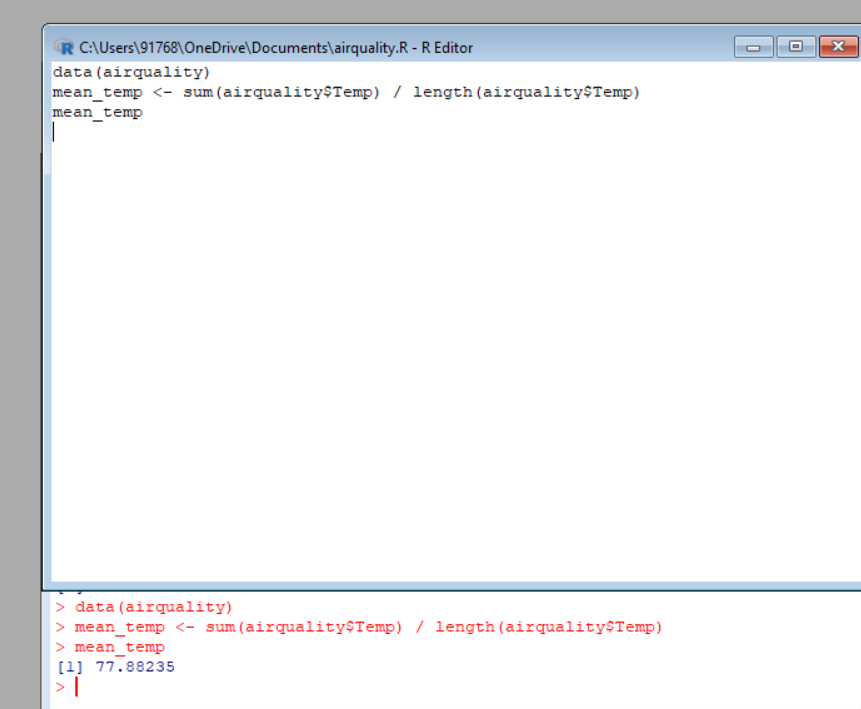
data(airquality)

wind\_speed\_gt\_17 <- sum(airquality$Wind > 17, na.rm = TRUE)

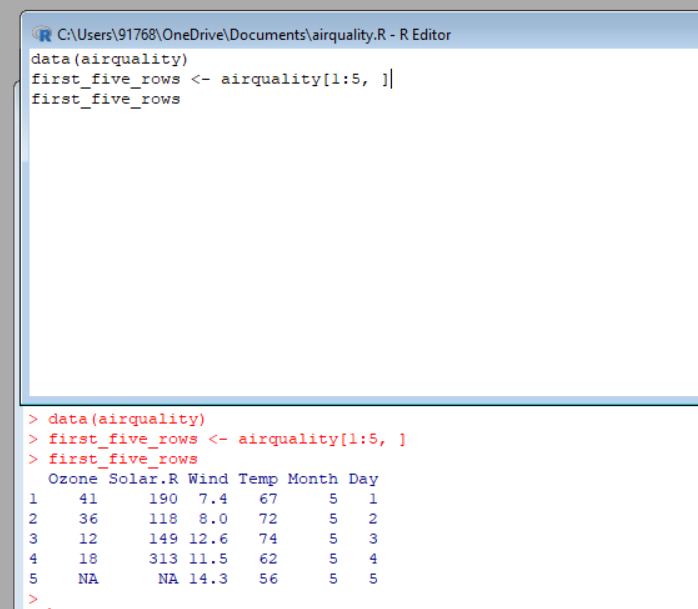
wind\_speed\_gt\_17

OUTPUT:

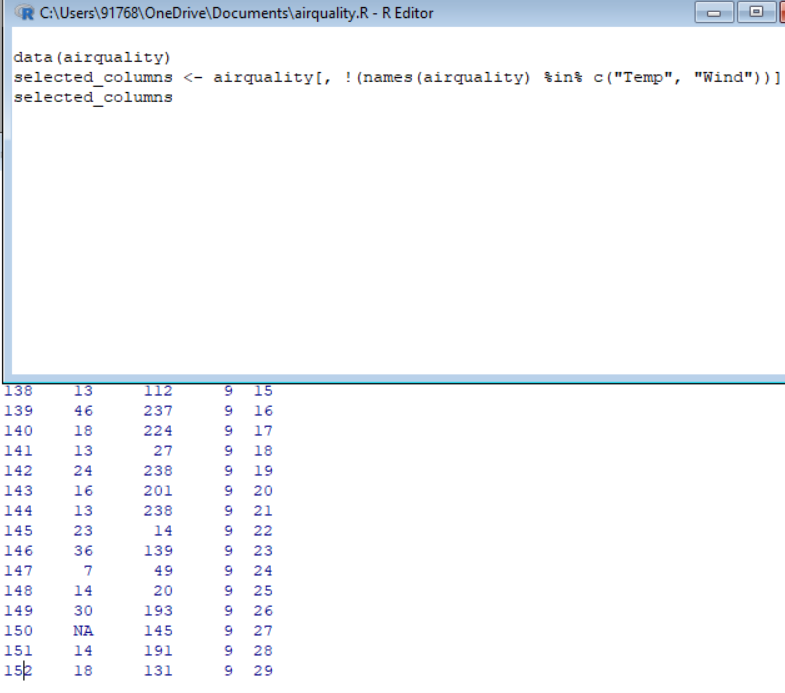
i)



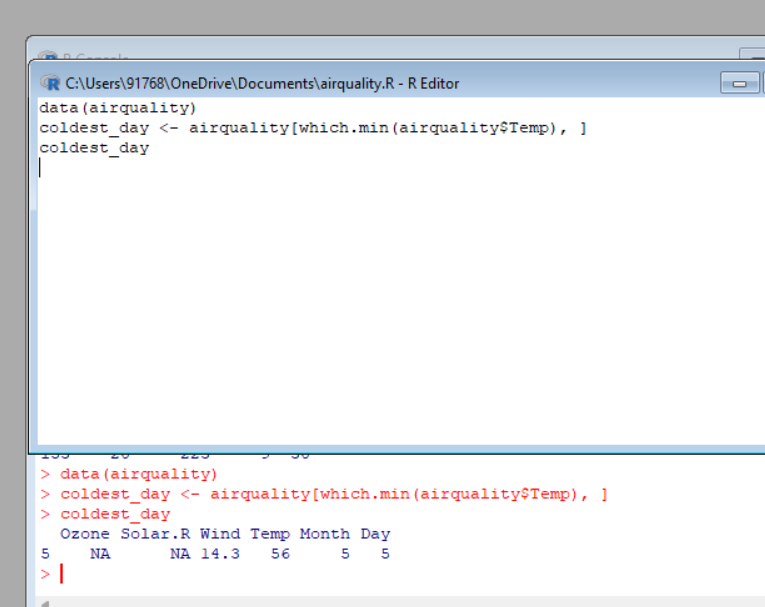
ii)



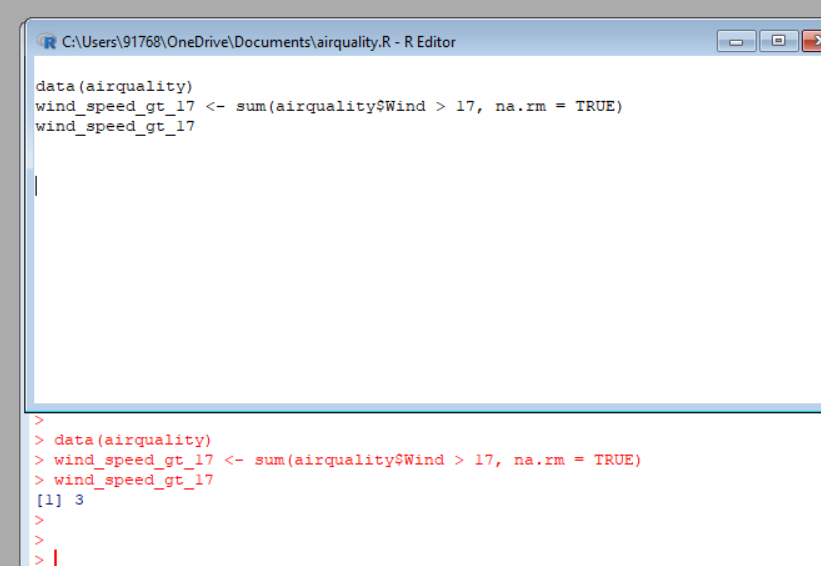
iii)



iv)



v)



4. (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?

SOURCE CODE:

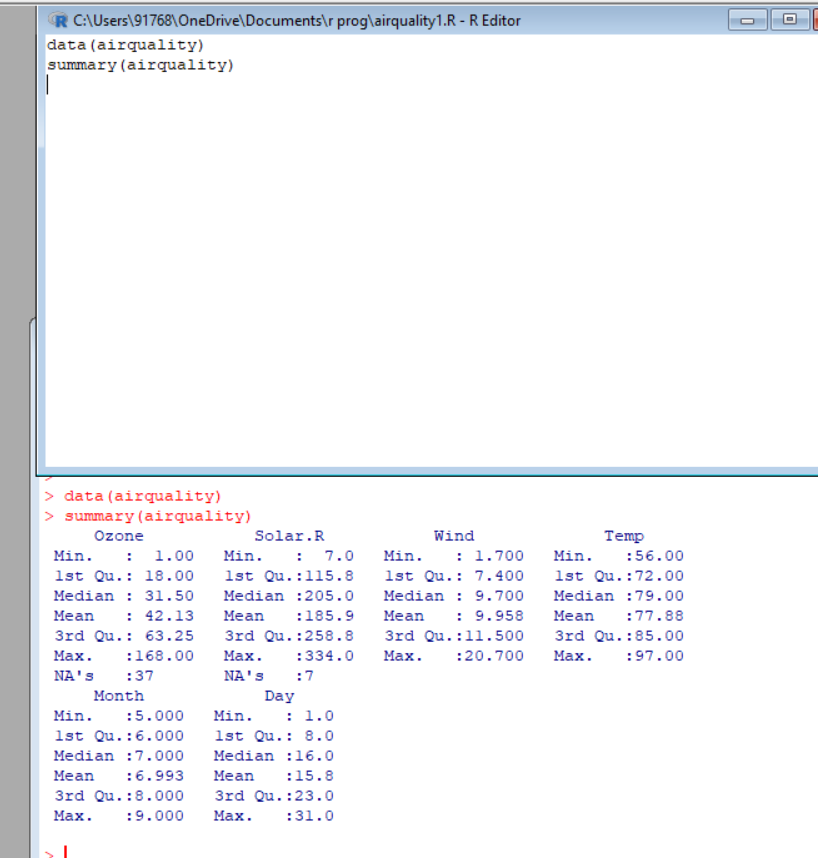
i)

data(airquality)

summary(aiirquality)

OUTPUT:

i)



5.(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

SOURCE CODE:

ii)

model<-(Ozone~Solar.R,data=airquality)

summary(model)

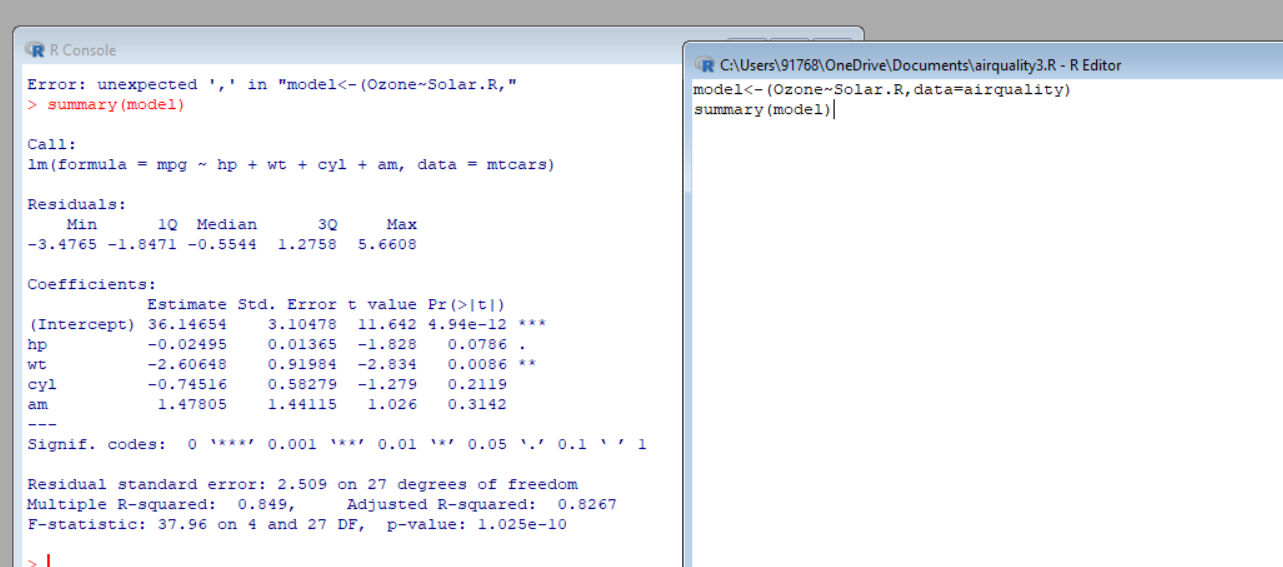
iii)

plot(airquality$Solar.R,airquality$Ozone,xlab="Solar.R",ylab="Ozone")

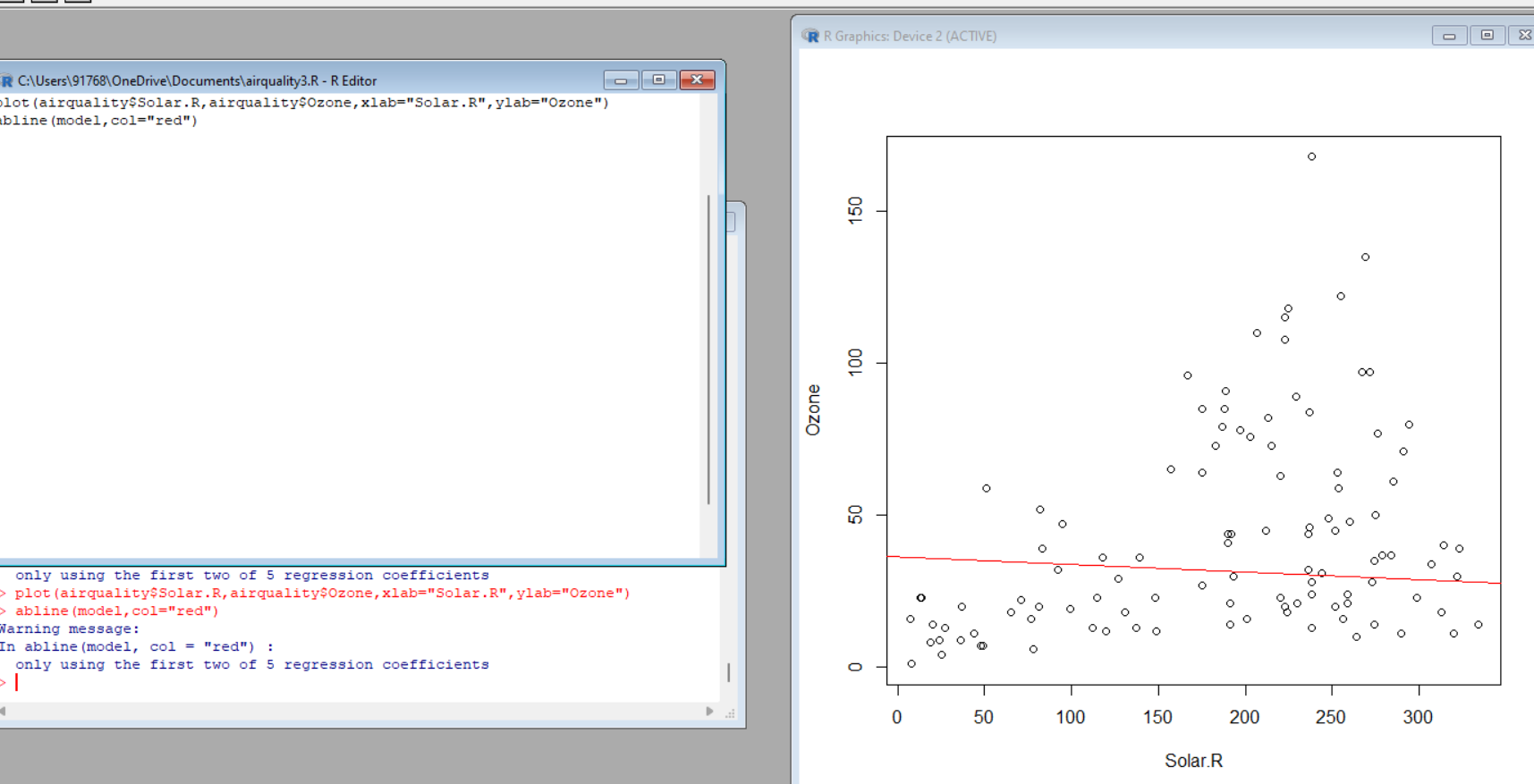
abline(model,col="red")

OUTPUT:

ii)



iii)



6. Load dataset named ChickWeight,

( i).Order the data frame, in ascending order by feature name “weight” grouped by

Feature “diet” and Extract the last 6 records from order data frame.

(ii).a Perform melting function based on “Chick&quot;, &quot;Time&quot;, &quot;Diet&quot; features as ID variables

b. Perform cast function to display the mean value of weight grouped by Diet

c. Perform cast function to display the mode of weight grouped by Diet

SOURCE CODE:

OUTPUT:

7. a. Create Box plot for “weight” grouped by “Diet”

b. Create a Histogram for “weight” features belong to Diet- 1 category

c. Create Scatter plot for “ weight” vs “Time” grouped by Diet

SOURCE CODE:

library(ggplot2)

weight <- c(150, 160, 170, 180, 190, 200, 210, 220, 230, 240)

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

diet <- c("A", "B", "A", "B", "A", "B", "A", "B", "A", "B")

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

boxplot <- ggplot(data, aes(x = diet, y = weight)) +

geom\_boxplot() +

labs(x = "Diet", y = "Weight", title = "Weight Distribution by Diet")

print(boxplot)

histogram <- ggplot(data[data$diet == "A", ], aes(x = weight)) +

geom\_histogram(binwidth = 10, fill = "steelblue", color = "white") +

labs(x = "Weight", y = "Count", title = "Weight Distribution for Diet-1")

print(histogram)

scatterplot <- ggplot(data, aes(x = time, y = weight, color = diet)) +

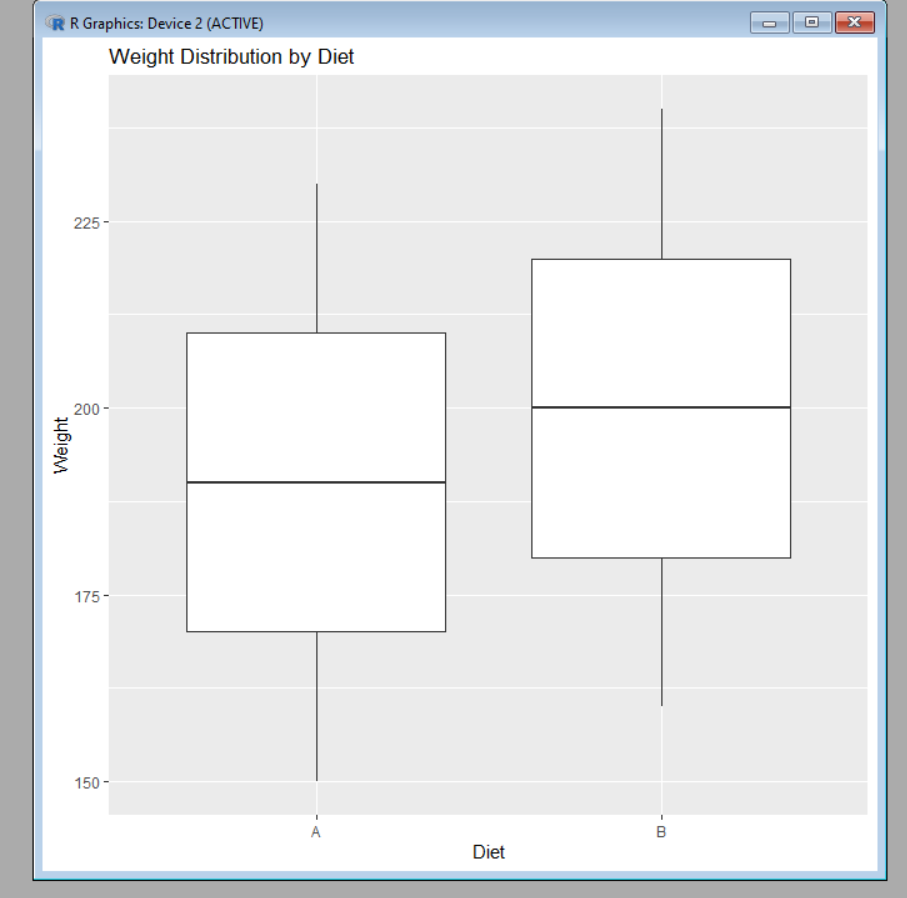
geom\_point() +

labs(x = "Time", y = "Weight", title = "Weight vs Time by Diet")

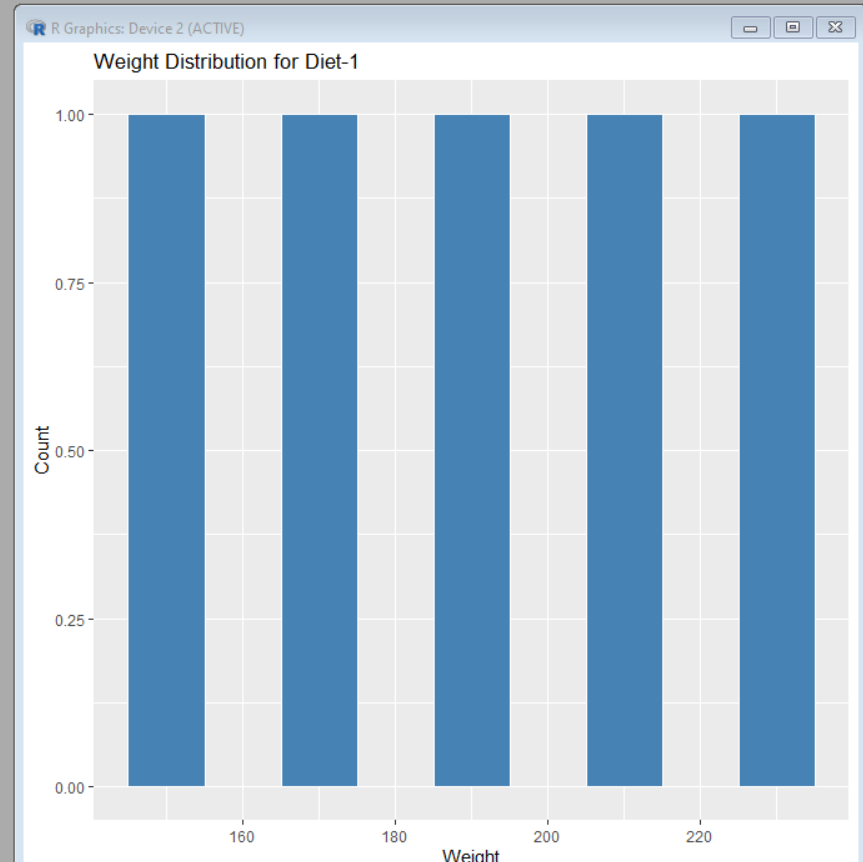
print(scatterplot)

OUTPUT:

Boxplot:



Histogram:



Scatterplot:

