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: 192125063

Name: KRISHVANTH KUMAR E

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

CODE:

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

# Set the seed for reproducibility
set.seed(123)

# Randomly sample the dataset
train_indices <- sample(1:nrow(iris), 0.8 * nrow(iris))  # 80% for training
train_data <- iris[train_indices, ]
test_data <- iris[-train_indices, ]

# Create a logistic regression model
library(nnet)
model <- multinom(Species ~ Petal.Width + Petal.Length, data = train_data)
```

Predict probabilities using the test data

predicted_probs <- predict(model, newdata = test_data, type = "probs")

Create the confusion matrix

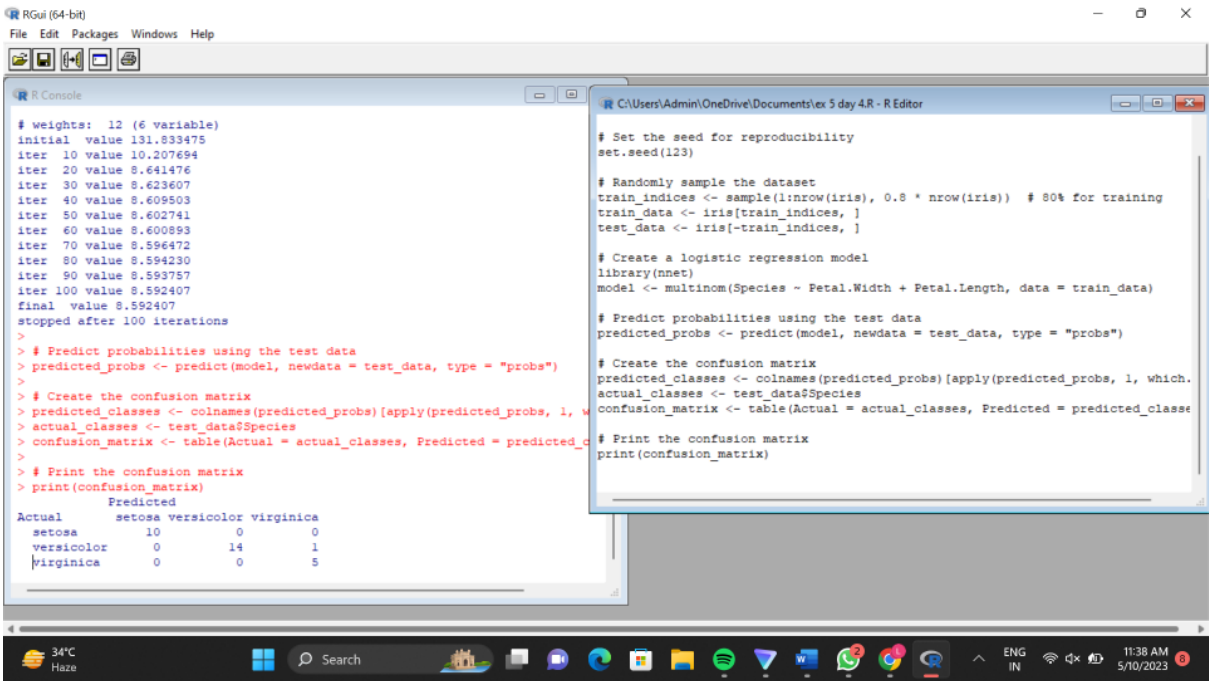
predicted_classes <- colnames(predicted_probs)[apply(predicted_probs, 1, which.max)]

actual_classes <- test_data$Species

confusion_matrix <- table(Actual = actual_classes, Predicted = predicted_classes)

Print the confusion matrix

print(confusion_matrix)



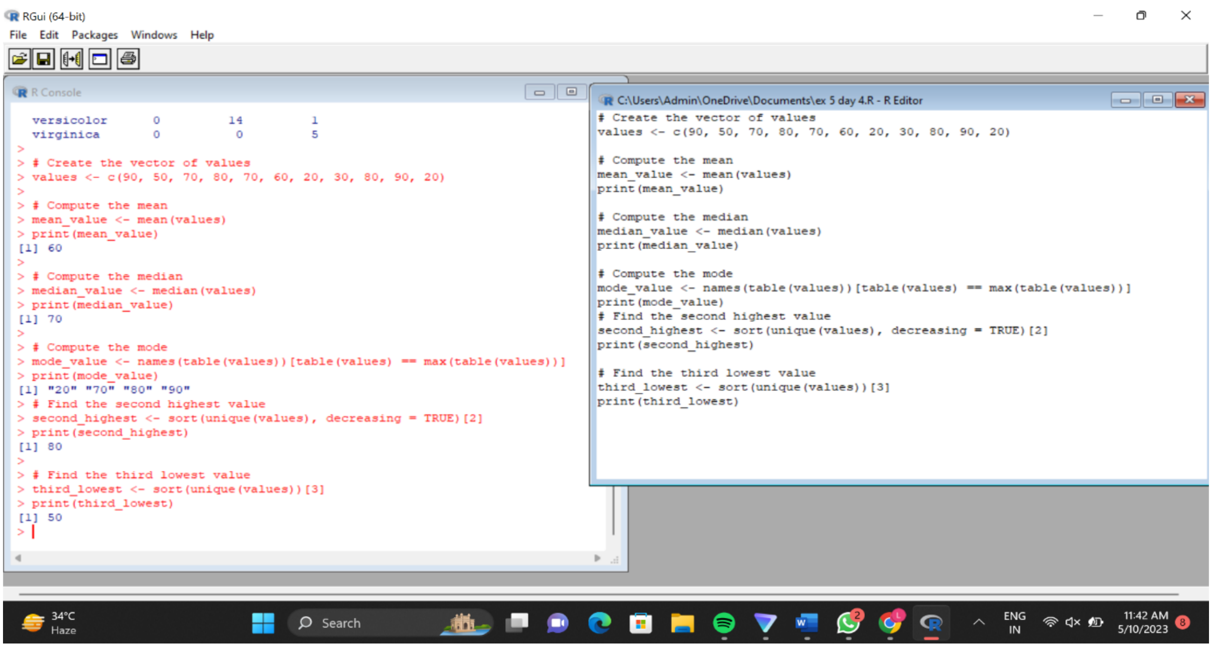
2. (i)Write suitable R code to compute the mean, median ,mode of the following values

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.

CODE:

```
# Compute the mean

mean_value <- mean(values)

print(mean_value)


# Compute the median

median_value <- median(values)

print(median_value)


# Compute the mode

mode_value <- names(table(values))[table(values) == max(table(values))]

print(mode_value)

# Find the second highest value

second_highest <- sort(unique(values), decreasing = TRUE)[2]

print(second_highest)


# Find the third lowest value

third_lowest <- sort(unique(values))[3]

print(third_lowest)
```



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).

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?

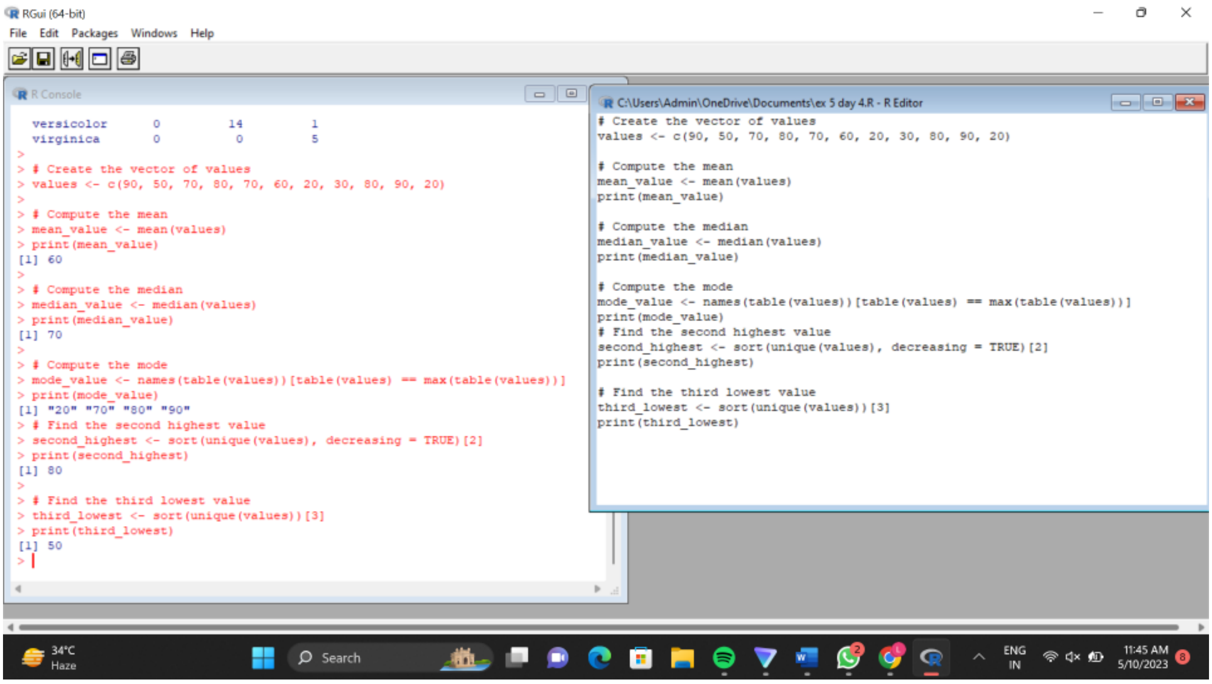
CODE:

```
# i. Compute the mean temperature (without using built-in function)
mean_temp <- sum(airquality$Temp) / length(airquality$Temp)
print(mean_temp)

# ii. Extract the first five rows from airquality
first_five_rows <- airquality[1:5, ]
print(first_five_rows)

# iii. Extract all columns from airquality except Temp and Wind
selected_columns <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]
print(selected_columns)

# iv. Identify the coldest day during the period
coldest_day <- airquality[which.min(airquality$Temp), ]
print(coldest_day)
```

v. Count the number of days with wind speed greater than 17 mph

wind_gt_17_count <- sum(airquality$Wind > 17)

print(wind_gt_17_count)



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?

CODE:

Load the airquality dataset

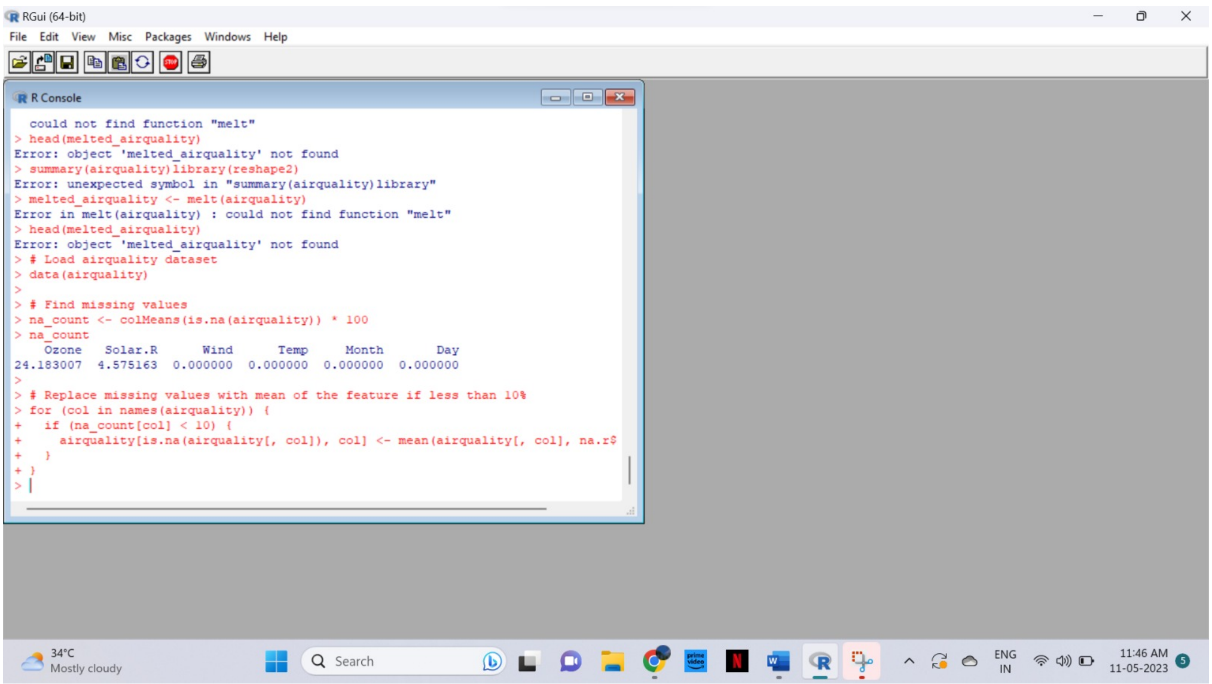
data(airquality)

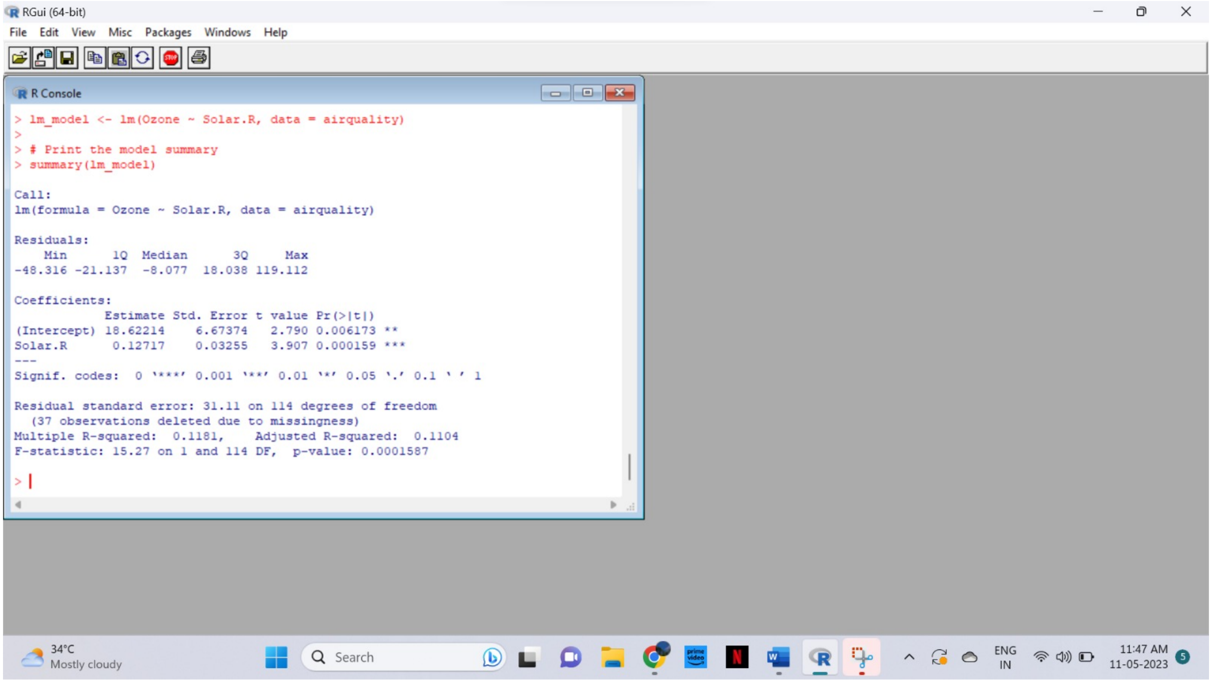
(i) Get the Summary Statistics of airquality dataset

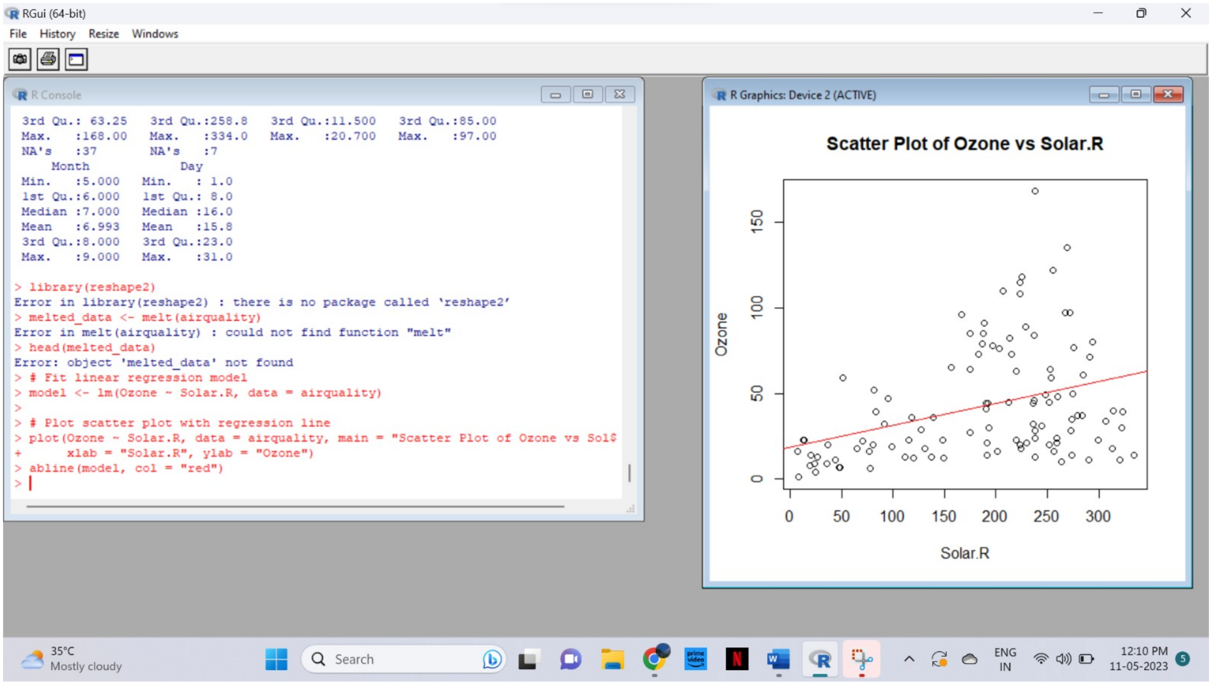
summary_stats <- summary(airquality)

print(summary_stats)

```
# (ii) Melt airquality dataset and display as long-format data

library(reshape2)

melted_data <- melt(airquality)

print(melted_data)


# (iii) Melt airquality dataset and specify month and day as "ID variables"

melted_data_id <- melt(airquality, id.vars = c("Month", "Day"))

print(melted_data_id)


# (iv) Cast the molten airquality dataset with respect to month and day features

casted_data <- dcast(melted_data_id, Month + Day ~ variable)

print(casted_data)


# (v) Compute the average of Ozone, Solar.R, Wind, and Temperature per month using cast function

avg_per_month <- dcast(melted_data, Month ~ variable, mean)

print(avg_per_month)
```







5.(i) Find any missing values(na) in features and drop the missing values if its less than 10%

Load airquality dataset

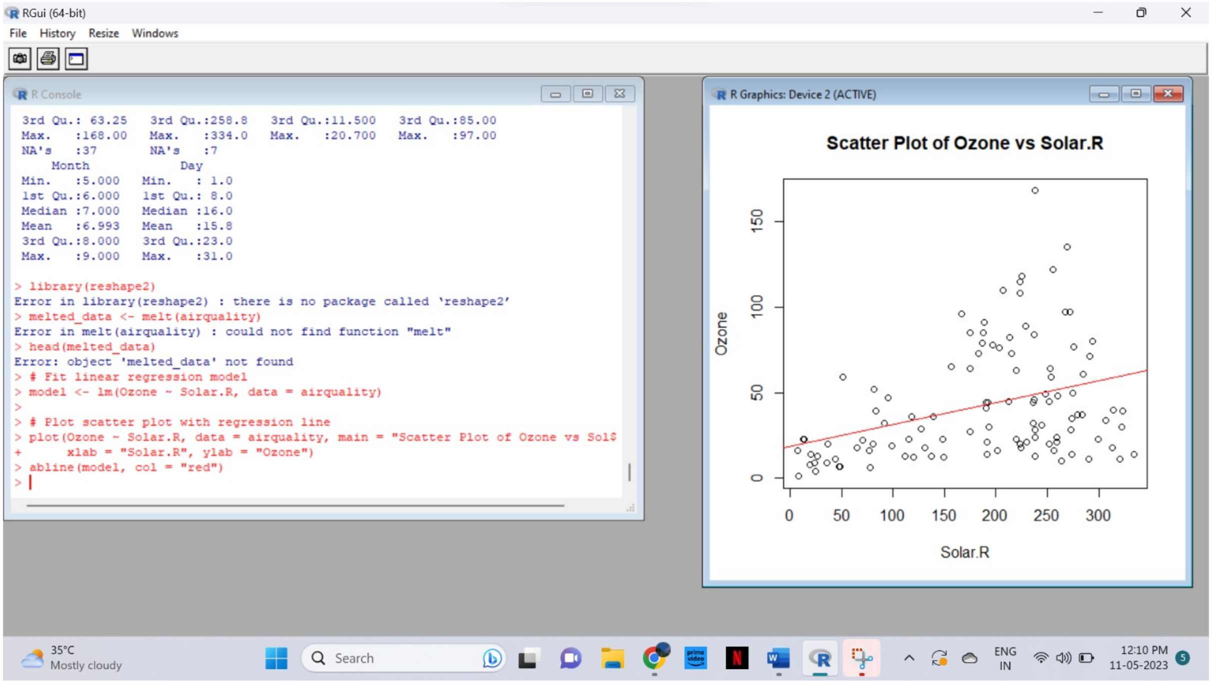
```
data(airquality)

# Find missing values
na_count <- colMeans(is.na(airquality)) * 100
na_count

# Replace missing values with mean of the feature if less than 10%
for (col in names(airquality)) {
  if (na_count[col] < 10) {
    airquality[is.na(airquality[, col]), col] <- mean(airquality[, col], na.rm = TRUE)
  }
}
```

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

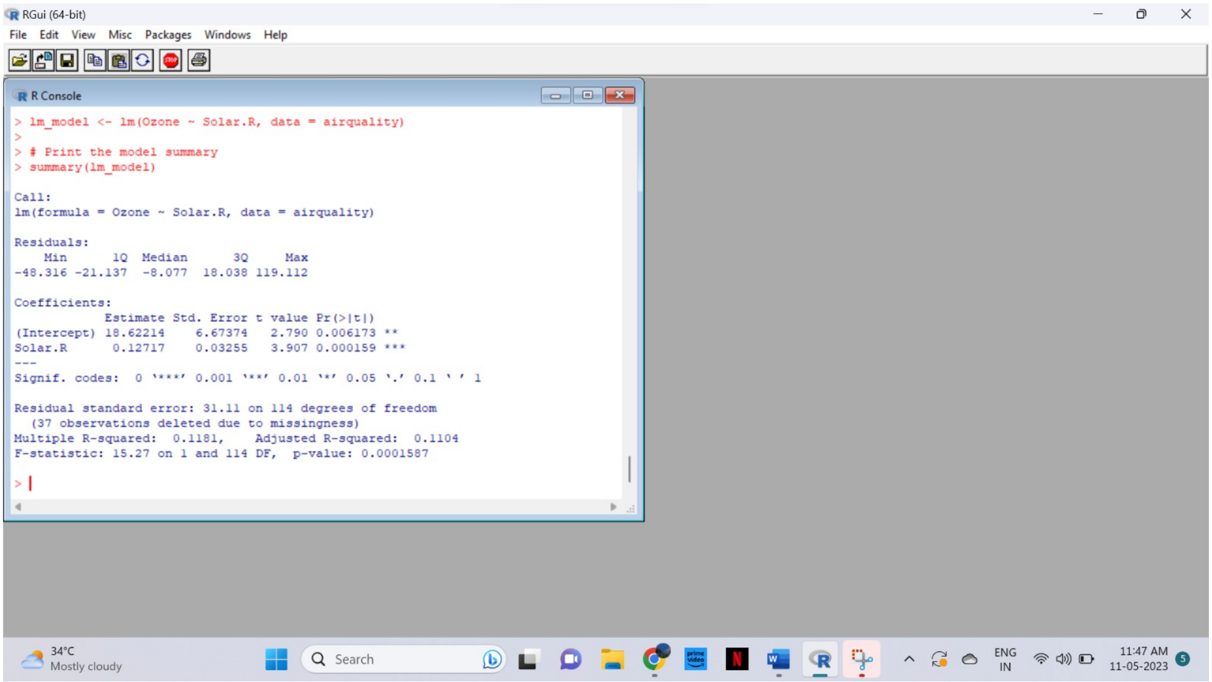
```
# Apply linear regression on "Ozone" and "Solar.R"
lm_model <- lm(Ozone ~ Solar.R, data = airquality)

# Print the model summary
summary(lm_model)
```

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

```
# Fit linear regression model
model <- lm(Ozone ~ Solar.R, data = airquality)

# Plot scatter plot with regression line
plot(Ozone ~ Solar.R, data = airquality, main = "Scatter Plot of Ozone vs Solar.R",
    xlab = "Solar.R", ylab = "Ozone")
abline(model, col = "red")
```



RGui (64-bit)
File History Resize Windows
R Console
3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
NA's   :37   NA's   :7
Month   Day
Min.   :5.000   Min.   : 1.0
1st Qu.:6.000   1st Qu.: 8.0
Median :7.000   Median :16.0
Mean   :6.993   Mean   :15.8
3rd Qu.:8.000   3rd Qu.:23.0
Max.   :9.000   Max.   :31.0
> library(reshape2)
Error in library(reshape2) : there is no package called 'reshape2'
> melted_data <- melt(airquality)
Error in melt(airquality) : could not find function "melt"
> head(melted_data)
Error: object 'melted_data' not found
> # Fit linear regression model
> model <- lm(Ozone ~ Solar.R, data = airquality)
> # Plot scatter plot with regression line
> plot(Ozone ~ Solar.R, data = airquality, main = "Scatter Plot of Ozone vs Sol$
+   xlab = "Solar.R", ylab = "Ozone")
> abline(model, col = "red")
R Graphics: Device 2 (ACTIVE)
Scatter Plot of Ozone vs Solar.R
Ozone
0
50
100
150
0
50
100
150
200
250
300
Solar.R
35°C
Mostly cloudy
Search
ENG
IN
12:10 PM
11-05-2023



RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> lm_model <- lm(Ozone ~ Solar.R, data = airquality)
> # Print the model summary
> summary(lm_model)
Call:
lm(formula = Ozone ~ Solar.R, data = airquality)
Residuals:
Min   1Q Median   3Q   Max
-48.316 -21.137 -8.077 18.038 119.112
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 18.62214   6.67374   2.790 0.006173 **
Solar.R   0.12717   0.03255   3.907 0.000159 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 31.11 on 114 degrees of freedom
(37 observations deleted due to missingness)
Multiple R-squared: 0.1181,   Adjusted R-squared: 0.1104
F-statistic: 15.27 on 1 and 114 DF, p-value: 0.0001587
34°C
Mostly cloudy
Search
ENG
IN
11:47 AM
11-05-2023