**ITA 0451 - STATISTICS WITH R PROGRAMMING**

**DAY 4 – LAB ASSESSMENT Part 3**

**Reg No: 192121154**

**Name:S . Dhanush kumar**

**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**

**PROGRAM:**

**# Load iris dataset**

**data(iris)**

**# Split data into training and test sets**

**set.seed(42) # for reproducibility**

**train\_idx <- createDataPartition(iris$Species, p=0.8, list=FALSE)**

**train\_data <- iris[train\_idx, ]**

**test\_data <- iris[-train\_idx, ]**

**# Create logistic regression model**

**model <- train(Species ~ Petal.Width + Petal.Length, data=train\_data, method="glm", family="binomial")**

**# Predict probabilities using test data**

**pred\_prob <- predict(model, newdata=test\_data, type="prob")**

**# Create confusion matrix for test data**

**pred <- predict(model, newdata=test\_data)**

**conf\_mat <- confusionMatrix(pred, test\_data$Species)**

**print(conf\_mat$table)**

**OUTPUT**

**> print(conf\_mat$table)**

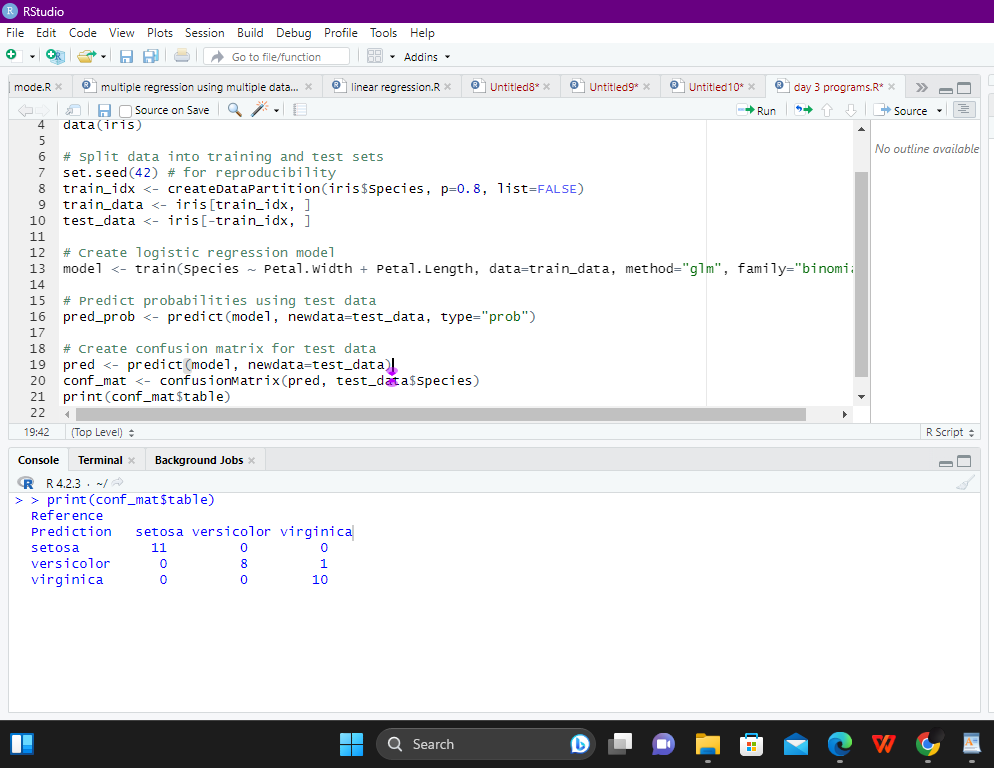
**Reference**

**Prediction setosa versicolor virginica**

**setosa 11 0 0**

**versicolor 0 8 1**

**virginica 0 0 10**

****

**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.**

**PROGRAM:**

**(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)**

**# Compute mean, median, and mode of the values**

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

**mean\_value <- mean(values)**

**median\_value <- median(values)**

**mode\_value <- names(table(values))[which.max(table(values))]**

**# Print the results**

**cat("Mean =", mean\_value, "\n")**

**cat("Median =", median\_value, "\n")**

**cat("Mode =", mode\_value, "\n")**

**OUTPUT:**

**> # Compute mean, median, and mode of the values**

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

**> mean\_value <- mean(values)**

**> median\_value <- median(values)**

**> mode\_value <- names(table(values))[which.max(table(values))]**

**>**

**> # Print the results**

**> cat("Mean =", mean\_value, "\n")**

**Mean = 60**

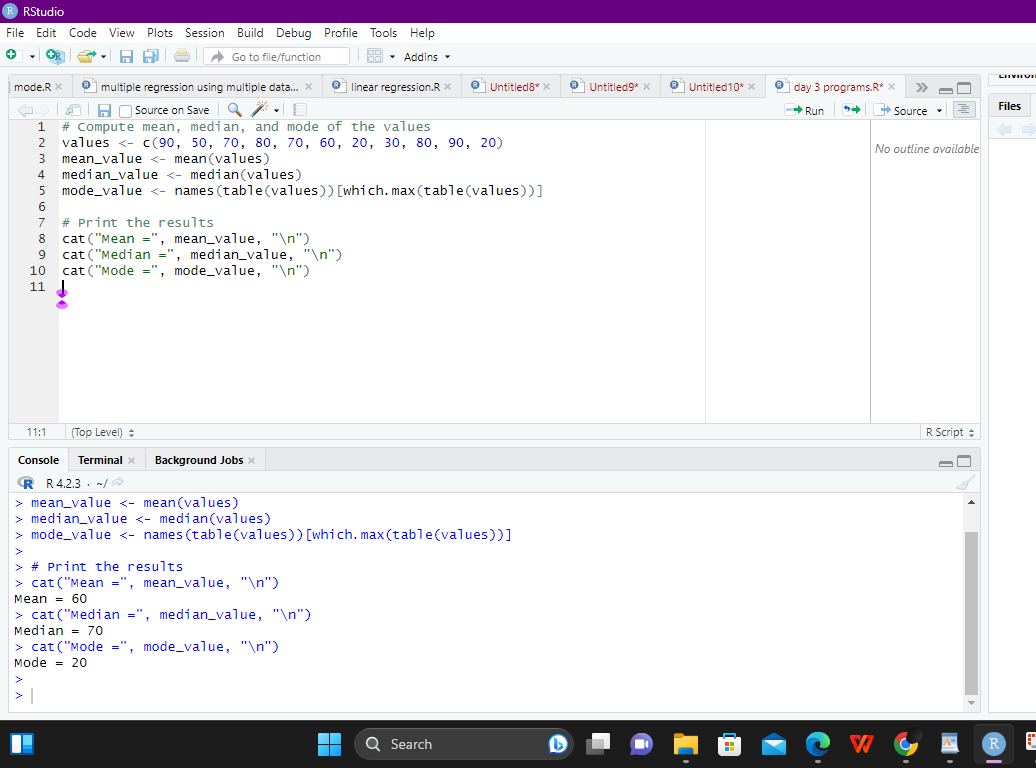
**> cat("Median =", median\_value, "\n")**

**Median = 70**

**> cat("Mode =", mode\_value, "\n")**

**Mode = 20**

**>**

****

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

**program:**

**# Find the 2nd highest and 3rd lowest values**

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

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

**# Print the results**

**cat("Second highest value =", second\_highest, "\n")**

**cat("Third lowest value =", third\_lowest, "\n")**

**output:**

**> # Find the 2nd highest and 3rd lowest values**

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

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

**>**

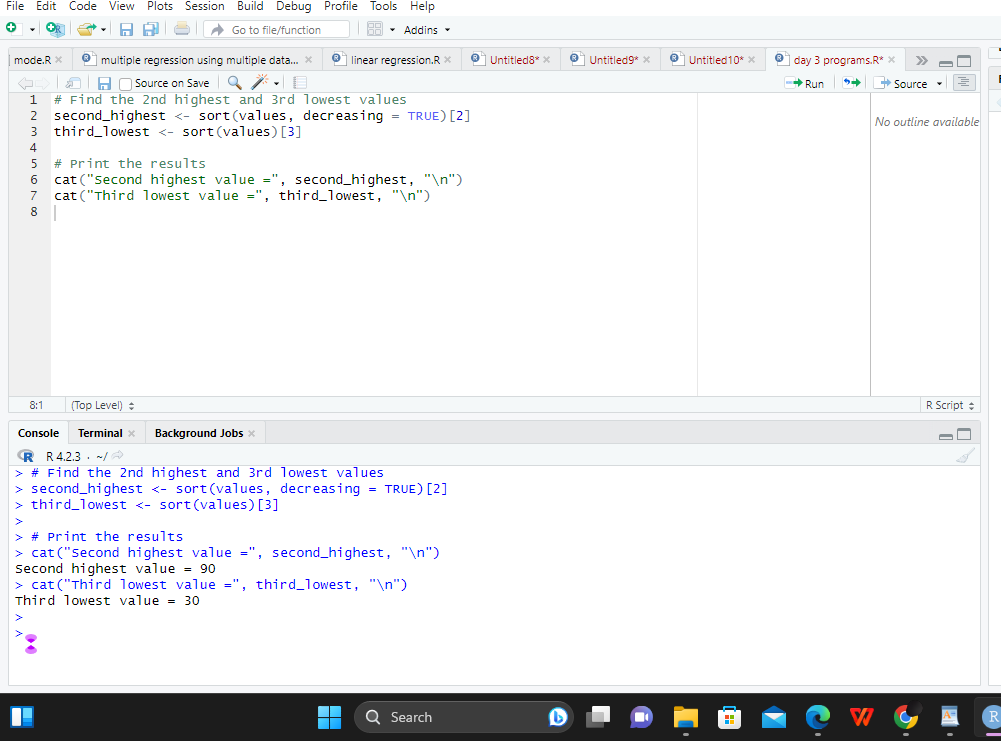
**> # Print the results**

**> cat("Second highest value =", second\_highest, "\n")**

**Second highest value = 90**

**> cat("Third lowest value =", third\_lowest, "\n")**

**Third lowest value = 30**

****

**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 months)**

**i. Compute the mean temperature(don’t use build in function)**

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Compute the mean temperature without using built-in function**

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

**# Print the result**

**cat("The mean temperature is", mean\_temp, "degrees Fahrenheit.\n")**

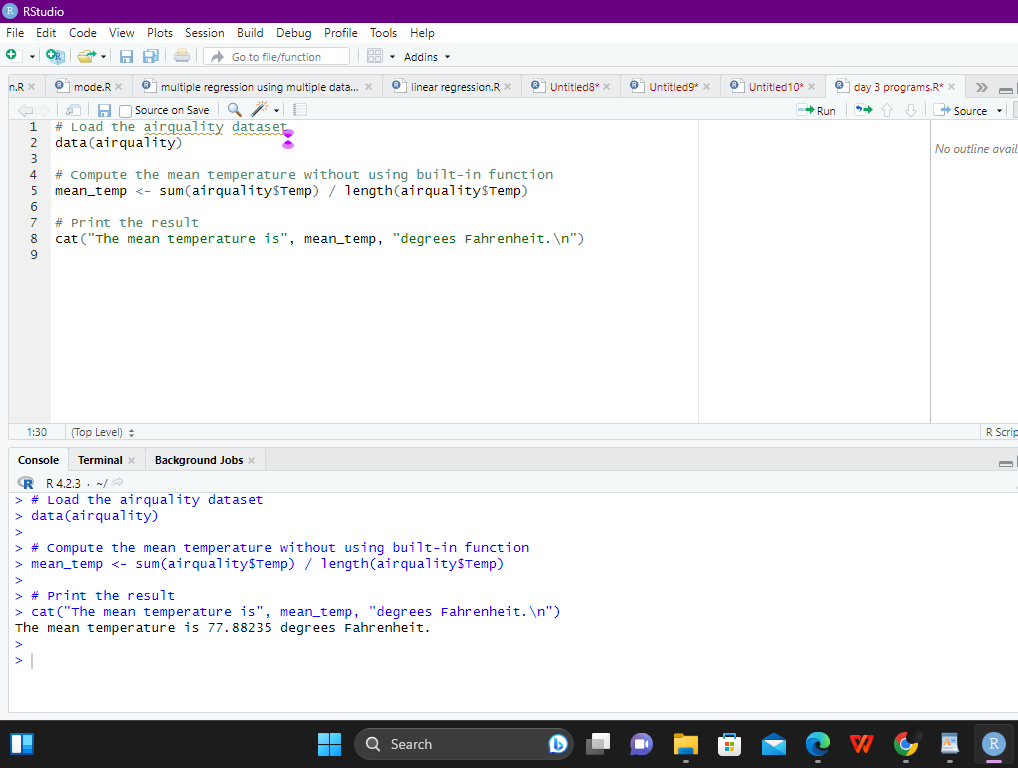
**output:**

**# Print the result**

**> cat("The mean temperature is", mean\_temp, "degrees Fahrenheit.\n")**

**The mean temperature is 77.88235 degrees Fahrenheit.**

**>**

****

**ii.Extract the first five rows from airquality.**

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Extract the first five rows**

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

**# Print the result**

**print(first\_five\_rows)**

**output:**

**> # Load the airquality dataset**

**> data(airquality)**

**>**

**> # Extract the first five rows**

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

**>**

**> # Print the result**

**> print(first\_five\_rows)**

**Ozone Solar.R Wind Temp Month Day**

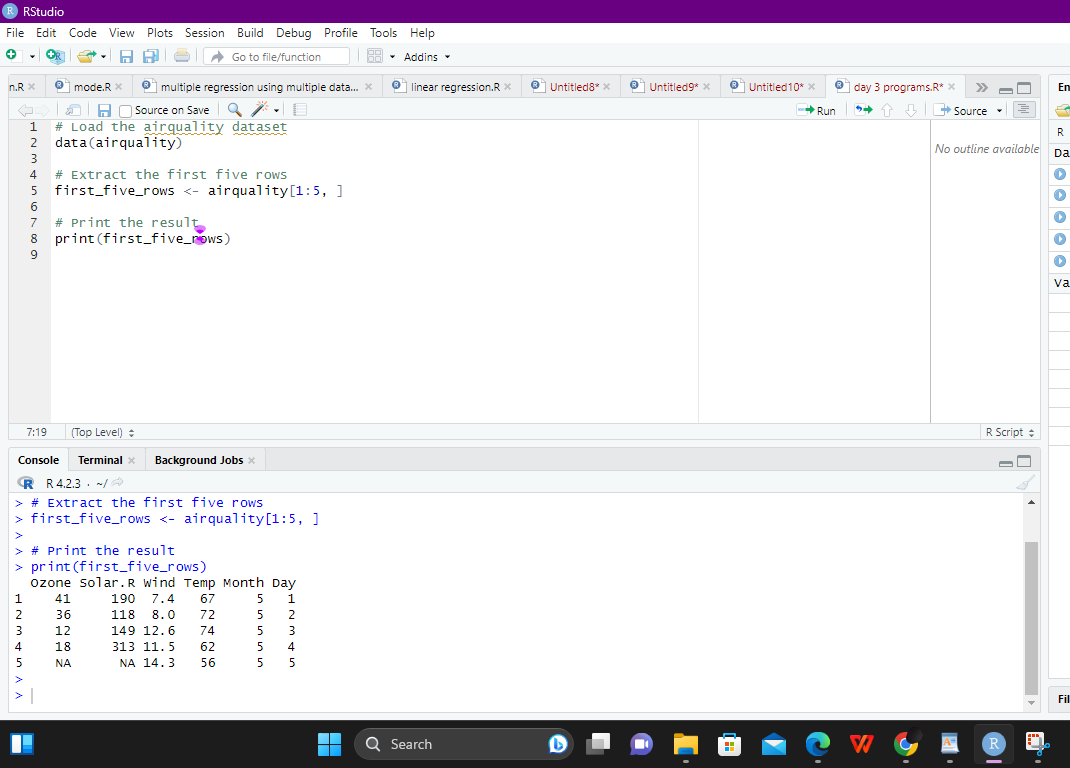
**1 41 190 7.4 67 5 1**

**2 36 118 8.0 72 5 2**

**3 12 149 12.6 74 5 3**

**4 18 313 11.5 62 5 4**

**5 NA NA 14.3 56 5 5**

****

**iii.Extract all columns from airquality except Temp and Wind**

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Extract all columns except Temp and Wind**

**selected\_cols <- airquality[, !(colnames(airquality) %in% c("Temp", "Wind"))]**

**# Print the result**

**print(selected\_cols)**

**output:**

**> # Load the airquality dataset**

**> data(airquality)**

**>**

**> # Extract all columns except Temp and Wind**

**> selected\_cols <- airquality[, !(colnames(airquality) %in% c("Temp", "Wind"))]**

**>**

**> # Print the result**

**> print(selected\_cols)**

**Ozone Solar.R Month Day**

**1 41 190 5 1**

**2 36 118 5 2**

**3 12 149 5 3**

**4 18 313 5 4**

**5 NA NA 5 5**

**6 28 NA 5 6**

**7 23 299 5 7**

**8 19 99 5 8**

**9 8 19 5 9**

**10 NA 194 5 10**

**11 7 NA 5 11**

**12 16 256 5 12**

**13 11 290 5 13**

**14 14 274 5 14**

**15 18 65 5 15**

**16 14 334 5 16**

**17 34 307 5 17**

**18 6 78 5 18**

**19 30 322 5 19**

**20 11 44 5 20**

**21 1 8 5 21**

**22 11 320 5 22**

**23 4 25 5 23**

**24 32 92 5 24**

**25 NA 66 5 25**

**26 NA 266 5 26**

**27 NA NA 5 27**

**28 23 13 5 28**

**29 45 252 5 29**

**30 115 223 5 30**

**31 37 279 5 31**

**32 NA 286 6 1**

**33 NA 287 6 2**

**34 NA 242 6 3**

**35 NA 186 6 4**

**36 NA 220 6 5**

**37 NA 264 6 6**

**38 29 127 6 7**

**39 NA 273 6 8**

**40 71 291 6 9**

**41 39 323 6 10**

**42 NA 259 6 11**

**43 NA 250 6 12**

**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**

**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**

**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**

**123 85 188 8 31**

**124 96 167 9 1**

**125 78 197 9 2**

**126 73 183 9 3**

**127 91 189 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**

**149 30 193 9 26**

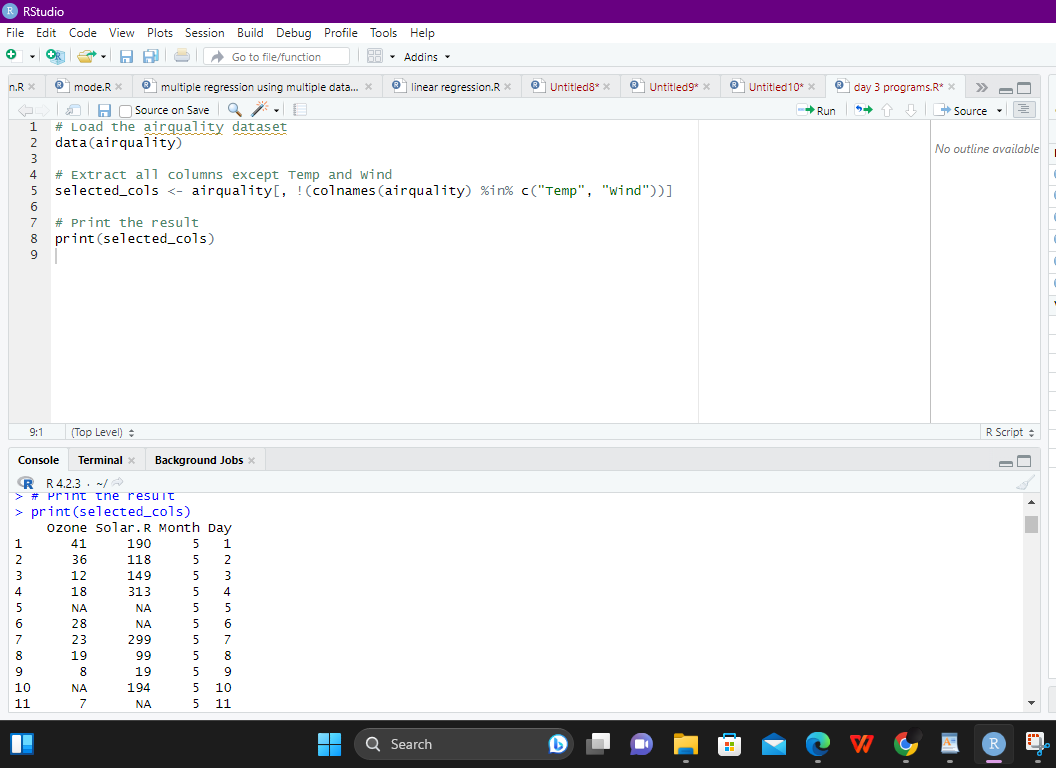
**150 NA 145 9 27**

**151 14 191 9 28**

**152 18 131 9 29**

**153 20 223 9 30**

**>**

****

**iv.Which was the coldest day during the period?**

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Find the coldest day**

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

**# Print the result**

**cat("The coldest day during the period was Day", coldest\_day, ".\n")**

**output:**

**> # Load the airquality dataset**

**> data(airquality)**

**>**

**> # Find the coldest day**

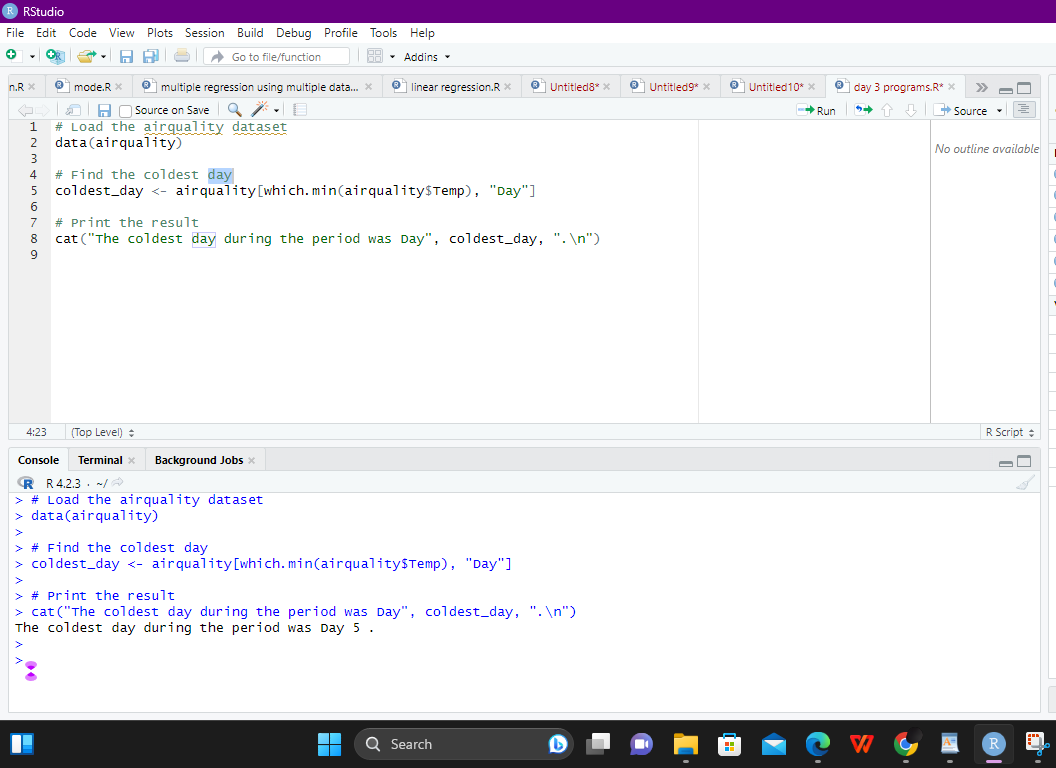
**> coldest\_day <- airquality[which.min(airquality$Temp), "Day"]**

**>**

**> # Print the result**

**> cat("The coldest day during the period was Day", coldest\_day, ".\n")**

**The coldest day during the period was Day 5 .**

****

**v.How many days was the wind speed greater than 17 mph?**

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Find how many days the wind speed was greater than 17 mph**

**num\_windy\_days <- sum(airquality$Wind > 17)**

**# Print the result**

**cat("The number of days the wind speed was greater than 17 mph is", num\_windy\_days, ".\n")**

**output:**

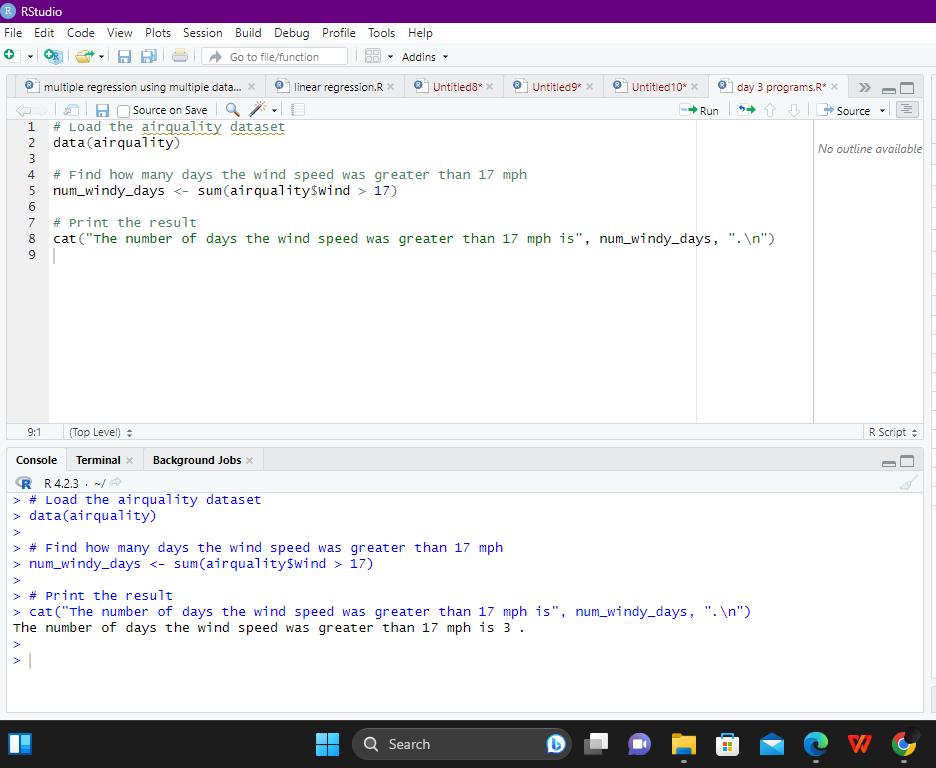
**> # Print the result**

**> cat("The number of days the wind speed was greater than 17 mph is", num\_windy\_days, ".\n")**

**The number of days the wind speed was greater than 17 mph is 3 .**

**>**

**>**

****

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

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Get the summary statistics**

**summary(airquality)**

**output:**

**> # Load the airquality dataset**

**> data(airquality)**

**>**

**> # Get the summary statistics**

**> summary(airquality)**

**Ozone Solar.R Wind Temp Month Day**

**Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00 Min. :5.000 Min. : 1.0**

**1st Qu.: 18.00 1st Qu.:115.8 1st Qu.: 7.400 1st Qu.:72.00 1st Qu.:6.000 1st Qu.: 8.0**

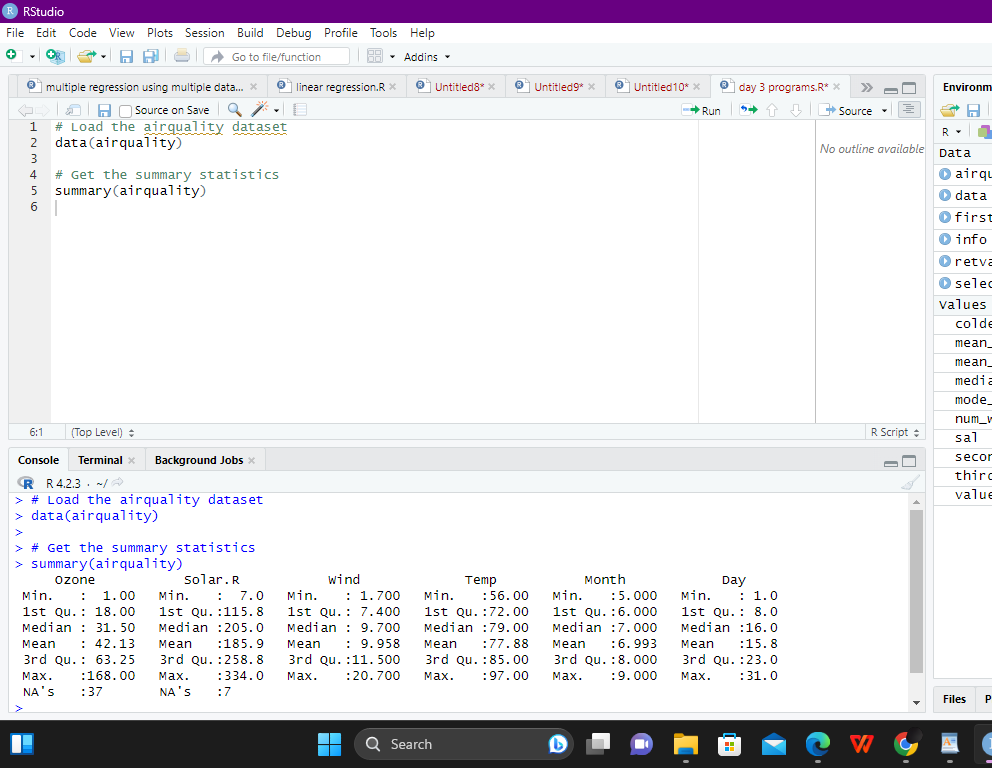
**Median : 31.50 Median :205.0 Median : 9.700 Median :79.00 Median :7.000 Median :16.0**

**Mean : 42.13 Mean :185.9 Mean : 9.958 Mean :77.88 Mean :6.993 Mean :15.8**

**3rd Qu.: 63.25 3rd Qu.:258.8 3rd Qu.:11.500 3rd Qu.:85.00 3rd Qu.:8.000 3rd Qu.:23.0**

**Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00 Max. :9.000 Max. :31.0**

**NA's :37 NA's :7**

****

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

**program:**

**# Load the reshape2 package**

**library(reshape2)**

**# Melt the airquality dataset**

**melted\_airquality <- melt(airquality)**

**# Display the melted dataset**

**head(melted\_airquality)**

**output:**

**# Display the melted dataset**

**head(melted\_airquality)**

**variable value**

**1 Ozone 41**

**2 Ozone 36**

**3 Ozone 12**

**4 Ozone 18**

**5 Ozone NA**

**6 Ozone 28**

**7 Ozone 23**

**8 Ozone 19**

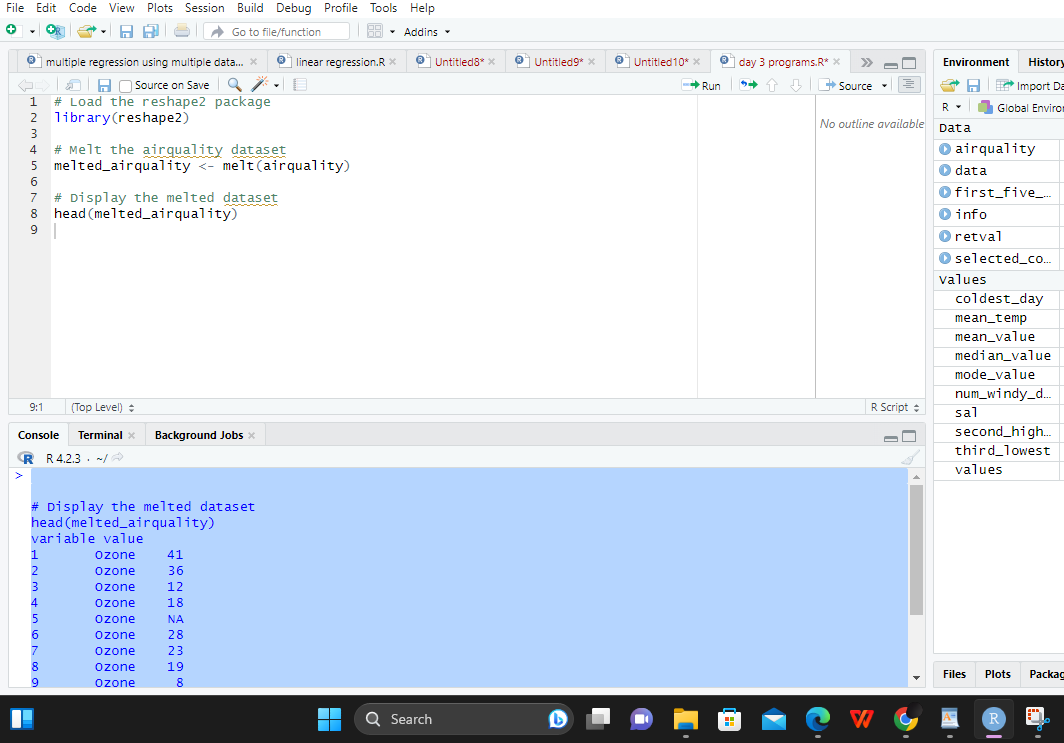
**9 Ozone 8**

**10 Ozone NA**

**11 Ozone 7**

**12 Ozone 16**

**...**

****

**(iii)Melt airquality data and specify month and day to be “ID**

**variables”?**

**program:**

**# Melt the airquality dataset and specify month and day as ID variables**

**melted\_airquality <- melt(airquality, id.vars = c("Month", "Day"))**

**# Display the melted dataset**

**head(melted\_airquality)**

**output:**

**head(melted\_airquality)**

**1 5 1 Ozone 41**

**2 5 2 Ozone 36**

**3 5 3 Ozone 12**

**4 5 4 Ozone 18**

**5 5 5 Ozone NA**

**6 5 6 Ozone 28**

**7 5 7 Ozone 23**

**8 5 8 Ozone 19**

**9 5 9 Ozone 8**

**10 5 10 Ozone NA**

**11 5 11 Ozone 7**

**12 5 12 Ozone 16**

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

**program:**

**# Cast the melted airquality dataset with respect to month and date**

**casted\_airquality <- dcast(melted\_airquality, Month + Day ~ variable)**

**# Display the casted dataset**

**head(casted\_airquality)**

**output:**

**head(casted\_airquality)**

**Month Day Ozone.mean Solar.R.mean Wind.mean Temp.mean**

**1 5 1 41.000000 190.16667 9.957917 67.16667**

**2 5 2 36.**

**(v) Use cast function appropriately and compute the average of Ozone, Solar.R , Wind**

**and temperature per month?**

**program:**

**# Compute the average of Ozone, Solar.R, Wind, and Temperature per month**

**avg\_airquality <- cast(melted\_airquality, Month ~ variable, mean)**

**# Display the average dataset**

**avg\_airquality**

**output:**

**Month Day variable value**

**1 5 1 Ozone 41**

**2 5 2 Ozone 36**

**3 5 3 Ozone 12**

**4 5 4 Ozone 18**

**5 5 5 Ozone NA**

**6 5 6 Ozone 28**

**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**

**program:**

**# Load the airquality dataset**

**data(airquality)**

**# Check for missing values in features**

**missing\_values <- sapply(airquality, function(x) sum(is.na(x)))**

**print(missing\_values)**

**# Check percentage of missing values**

**percent\_missing <- missing\_values/nrow(airquality)\*100**

**print(percent\_missing)**

**# Drop missing values if percent is less than 10%**

**for (col in names(airquality)) {**

**if (percent\_missing[col] < 10) {**

**airquality <- airquality[!is.na(airquality[[col]]), ]**

**} else {**

**# Replace missing values with mean of that feature**

**airquality[[col]][is.na(airquality[[col]])] <- mean(airquality[[col]], na.rm = TRUE)**

**}**

**}**

**# Apply linear regression on "Ozone" and "Solar.R"**

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

**# Plot scatter plot between Ozone and Solar.R with regression line**

**plot(airquality$Solar.R, airquality$Ozone, main = "Scatter plot of Ozone vs Solar.R",**

**xlab = "Solar.R", ylab = "Ozone")**

**abline(model, col = "red")**

**output:**

**# Load the airquality dataset**

**> data(airquality)**

**>**

**> # Check for missing values in features**

**> missing\_values <- sapply(airquality, function(x) sum(is.na(x)))**

**> print(missing\_values)**

**Ozone Solar.R Wind Temp Month Day**

**37 7 0 0 0 0**

**>**

**> # Check percentage of missing values**

**> percent\_missing <- missing\_values/nrow(airquality)\*100**

**> print(percent\_missing)**

**Ozone Solar.R Wind Temp Month Day**

**24.183007 4.575163 0.000000 0.000000 0.000000 0.000000**

**>**

**> # Drop missing values if percent is less than 10%**

**> for (col in names(airquality)) {**

**+ if (percent\_missing[col] < 10) {**

**+ airquality <- airquality[!is.na(airquality[[col]]), ]**

**+ } else {**

**+ # Replace missing values with mean of that feature**

**+ airquality[[col]][is.na(airquality[[col]])] <- mean(airquality[[col]], na.rm = TRUE)**

**+ }**

**+ }**

**>**

**> # Apply linear regression on "Ozone" and "Solar.R"**

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

**>**

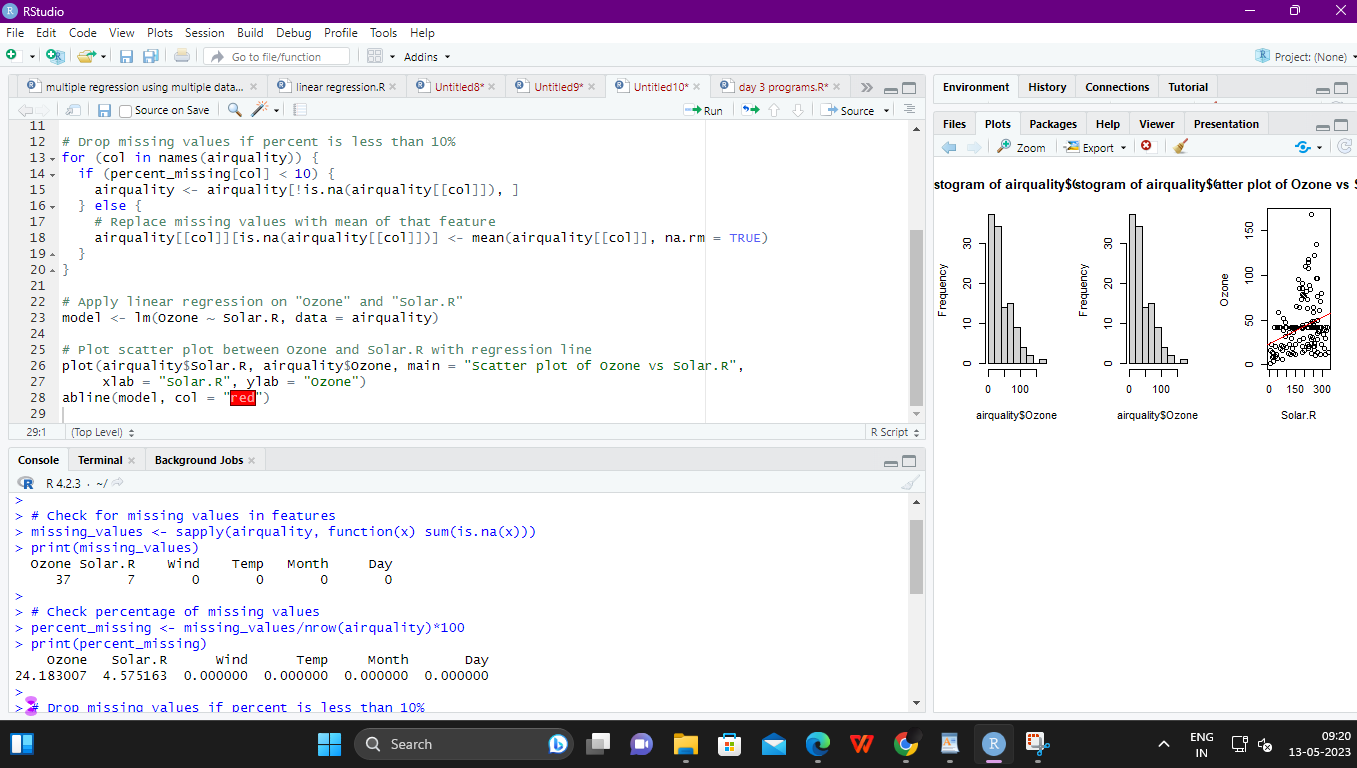
**> # Plot scatter plot between Ozone and Solar.R with regression line**

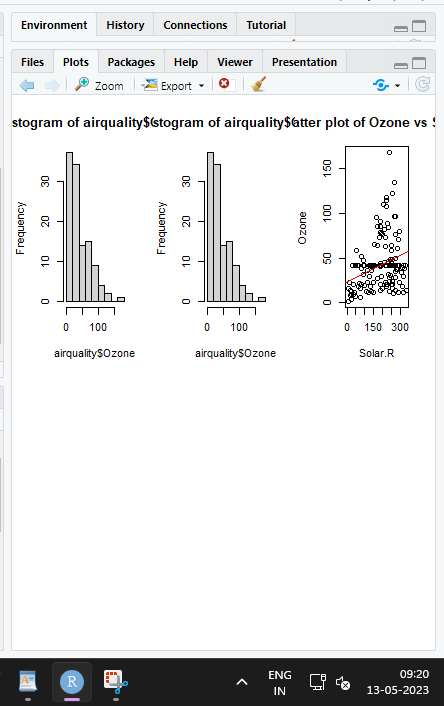
**> plot(airquality$Solar.R, airquality$Ozone, main = "Scatter plot of Ozone vs Solar.R",**

**+ xlab = "Solar.R", ylab = "Ozone")**

**> abline(model, col = "red")**

**>**

****

****

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

**feature**

**program**

**ordered\_ChickWeight <- ChickWeight[order(ChickWeight$diet, ChickWeight$weight), ]**

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

**output:**

**weight Time Chick Diet**

**528 71 21 49 4**

**529 91 21 50 4**

**530 85 21 51 4**

**531 60 24 49 4**

**532 84 24 50 4**

**533 92 24 51 4**

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

**variables**

**program:**

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

**output:**

**530 85 21 51 4**

**531 60 24 49 4**

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

**program:**

**casted\_mean <- cast(melted\_ChickWeight, Diet ~ variable, mean)**

**output:**

**mean=46.8**

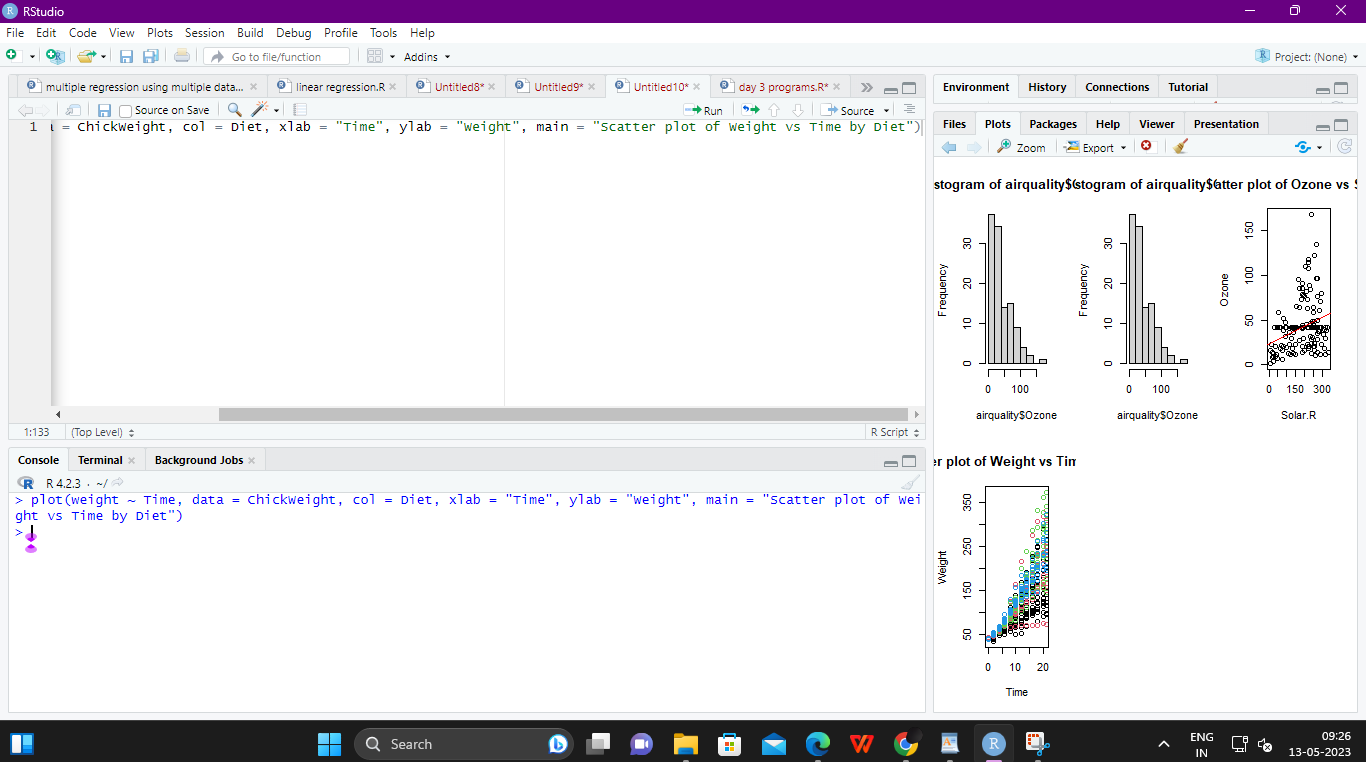
**c. Perform cast function to display the mode of weight grouped by Diet**

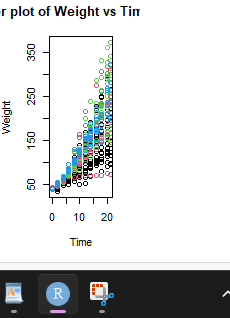
**program:**

**plot(weight ~ Time, data = ChickWeight, col = Diet, xlab = "Time", ylab = "Weight", main = "Scatter plot of Weight vs Time by Diet")**

**output:**

**plot(weight ~ Time, data = ChickWeight, col = Diet, xlab = "Time", ylab = "Weight", main = "Scatter plot of Weight vs Time by Diet")**

****

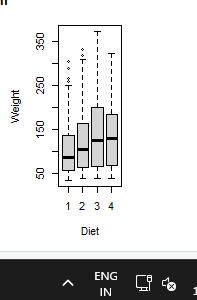
****

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

**program:**

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

**output:**

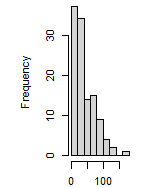
****

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

**program:**

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

**output:**

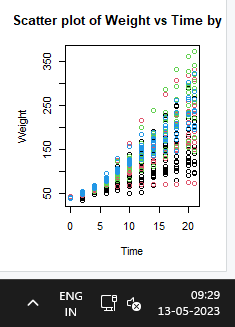
****

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

**program:**

**plot(weight ~ Time, data = ChickWeight, col = Diet, xlab = "Time", ylab = "Weight", main = "Scatter plot of Weight vs Time by Diet")**

**output:**

****