DAY 4 – LAB ASSESSMENT Part 3

Reg No:192124215

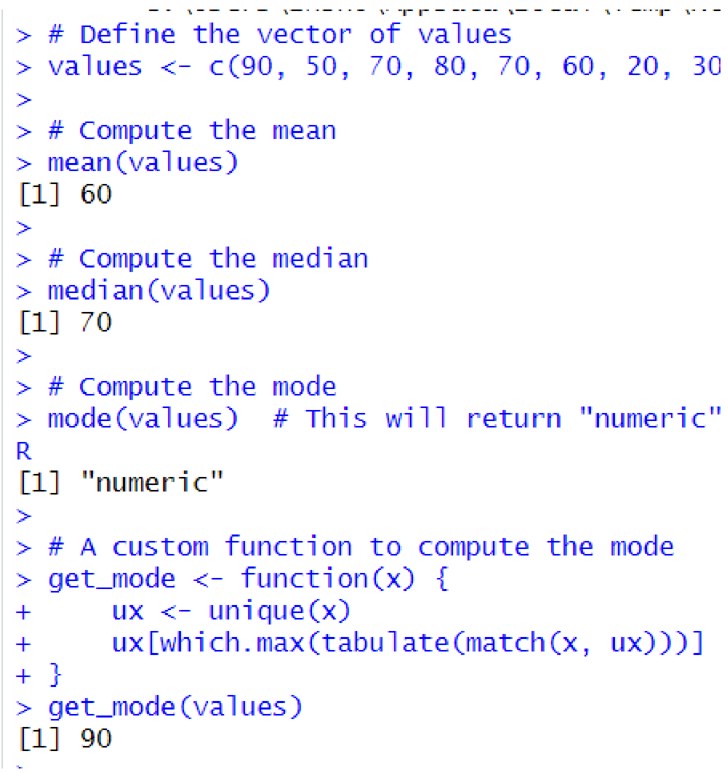
Name:Indhumathi v

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

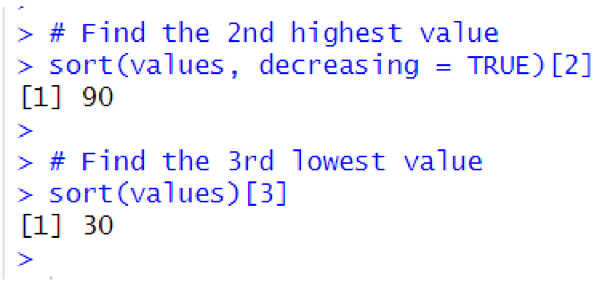
2. (i)Write suitable R code to compute the mean, median ,mode of the following values

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.



```
> # Define the vector of values
> values <- c(90, 50, 70, 80, 70, 60, 20, 30
>
> # Compute the mean
> mean(values)
[1] 60
>
> # Compute the median
> median(values)
[1] 70
>
> # Compute the mode
> mode(values)  # This will return "numeric"
R
[1] "numeric"
>
> # A custom function to compute the mode
> get_mode <- function(x) {
+     ux <- unique(x)
+     ux[which.max(tabulate(match(x, ux)))]
+ }
> get_mode(values)
[1] 90
```



```
> # Find the 2nd highest value
> sort(values, decreasing = TRUE)[2]
[1] 90
>
> # Find the 3rd lowest value
> sort(values)[3]
[1] 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).

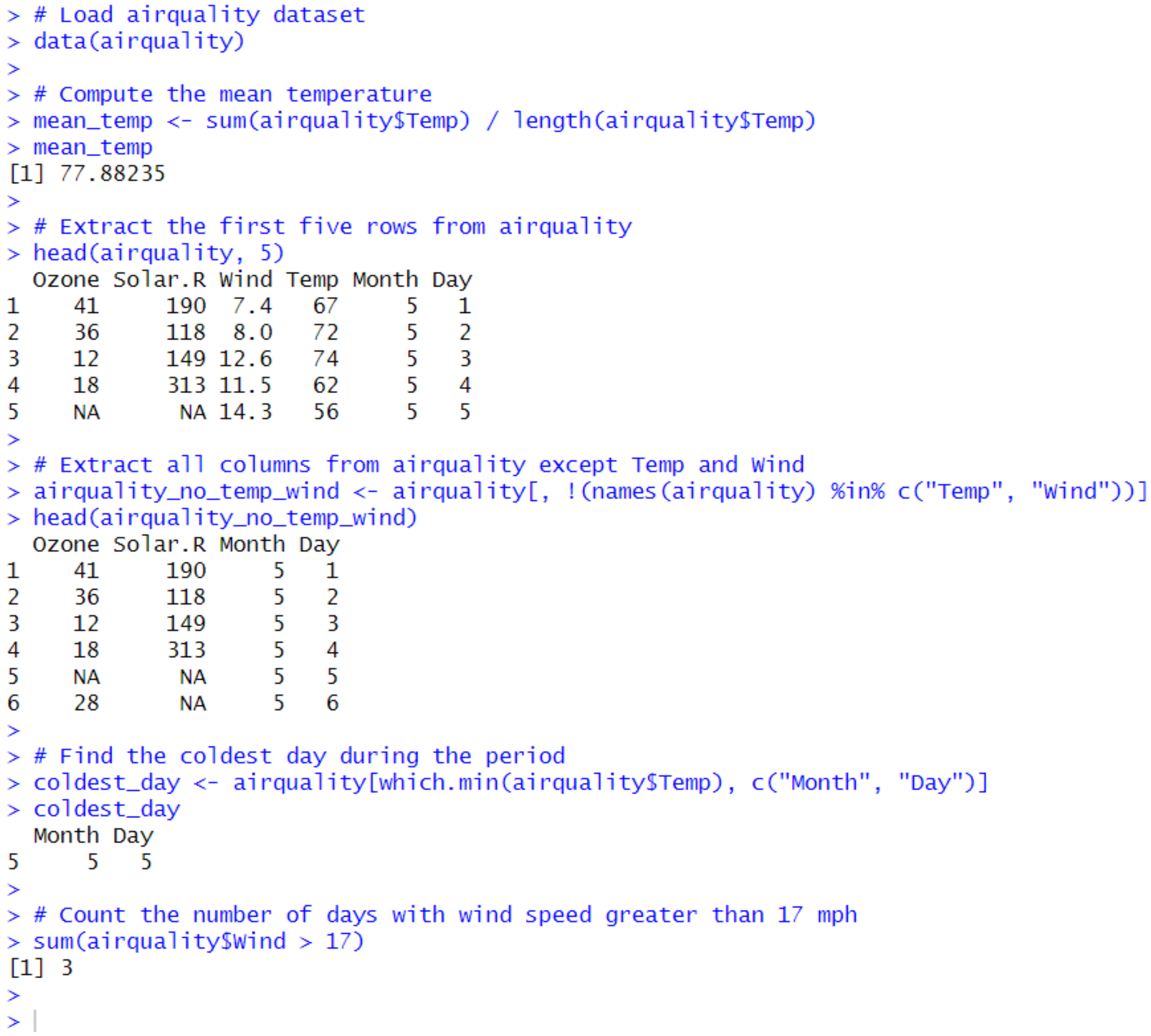
i. Compute the mean temperature(don't use build in function)

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?



```
> # Load airquality dataset
> data(airquality)
>
> # Compute the mean temperature
> mean_temp <- sum(airquality$Temp) / length(airquality$Temp)
> mean_temp
[1] 77.88235
>
> # Extract the first five rows from airquality
> head(airquality, 5)
  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
>
> # Extract all columns from airquality except Temp and Wind
> airquality_no_temp_wind <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]
> head(airquality_no_temp_wind)
  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
>
> # Find the coldest day during the period
> coldest_day <- airquality[which.min(airquality$Temp), c("Month", "Day")]
> coldest_day
  Month Day
5     5   5
>
> # Count the number of days with wind speed greater than 17 mph
> sum(airquality$Wind > 17)
[1] 3
>
> |
```

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

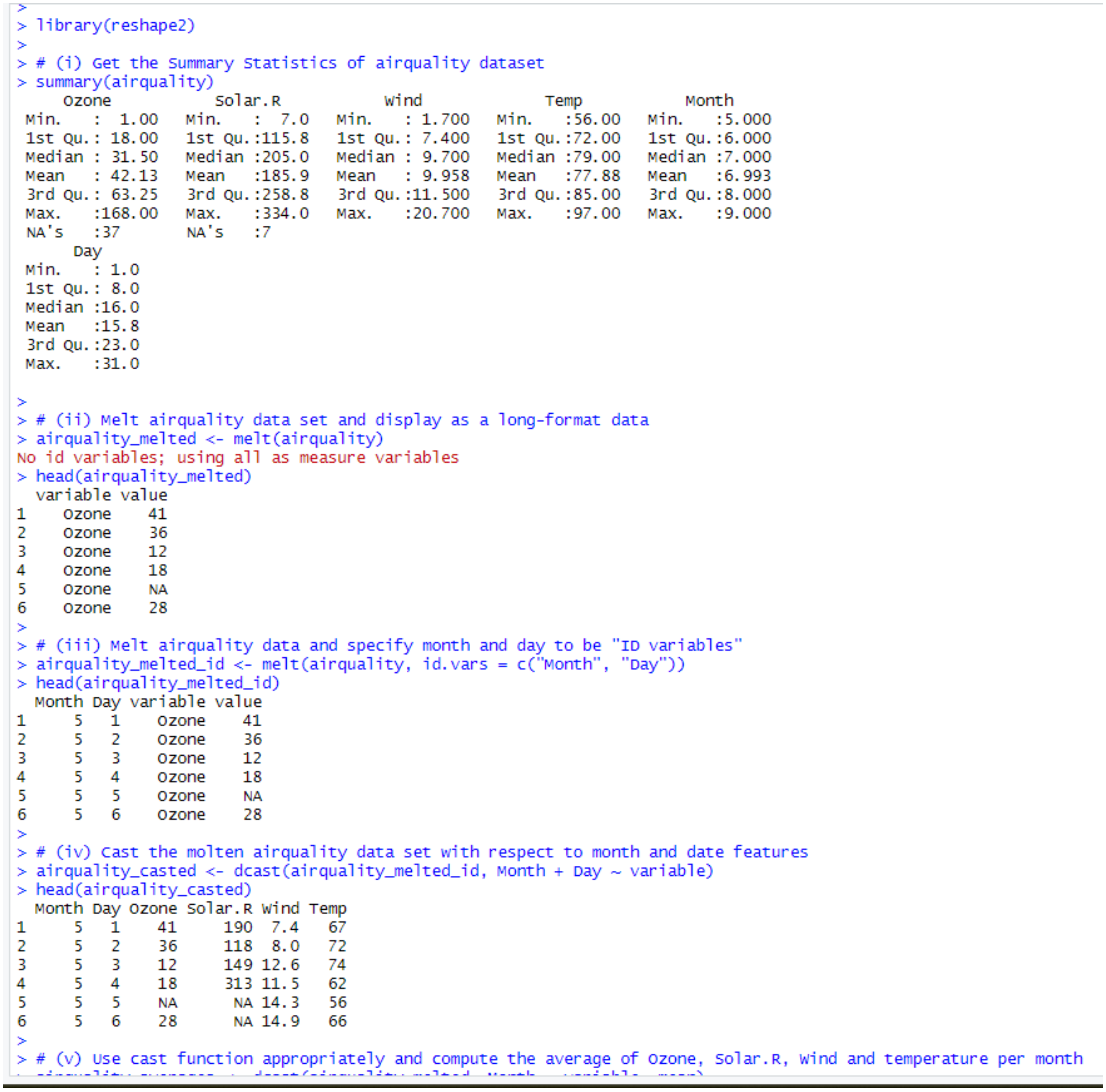
(ii)Melt airquality data set and display as a long – format data?

(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?



```
>
> library(reshape2)
>
> # (i) Get the Summary Statistics of airquality dataset
> summary(airquality)
     Ozone           Solar.R           Wind             Temp           Month      
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00   Min.   :5.000  
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00   1st Qu.:6.000  
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00   Median :7.000  
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88   Mean   :6.993  
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00   3rd Qu.:8.000  
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00   Max.   :9.000  
 NA's   :37       NA's   :7                                                       
      Day      
 Min.   : 1.0  
 1st Qu.: 8.0  
 Median :16.0  
 Mean   :15.8  
 3rd Qu.:23.0  
 Max.   :31.0  

>
> # (ii) Melt airquality data set and display as a long-format data
> airquality_melted <- melt(airquality)
No id variables; using all as measure variables
> head(airquality_melted)
  variable value
1    Ozone    41
2    Ozone    36
3    Ozone    12
4    Ozone    18
5    Ozone    NA
6    Ozone    28
>
> # (iii) Melt airquality data and specify month and day to be "ID variables"
> airquality_melted_id <- melt(airquality, id.vars = c("Month", "Day"))
> head(airquality_melted_id)
  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
>
> # (iv) Cast the molten airquality data set with respect to month and date features
> airquality_casted <- dcast(airquality_melted_id, Month + Day ~ variable)
> head(airquality_casted)
  Month Day Ozone Solar.R Wind Temp
1     5   1    41     190  7.4   67
2     5   2    36     118  8.0   72
3     5   3    12     149 12.6   74
4     5   4    18     313 11.5   62
5     5   5    NA      NA 14.3   56
6     5   6    28      NA 14.9   66
>
> # (v) Use cast function appropriately and compute the average of Ozone, Solar.R, Wind and temperature per month
```

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

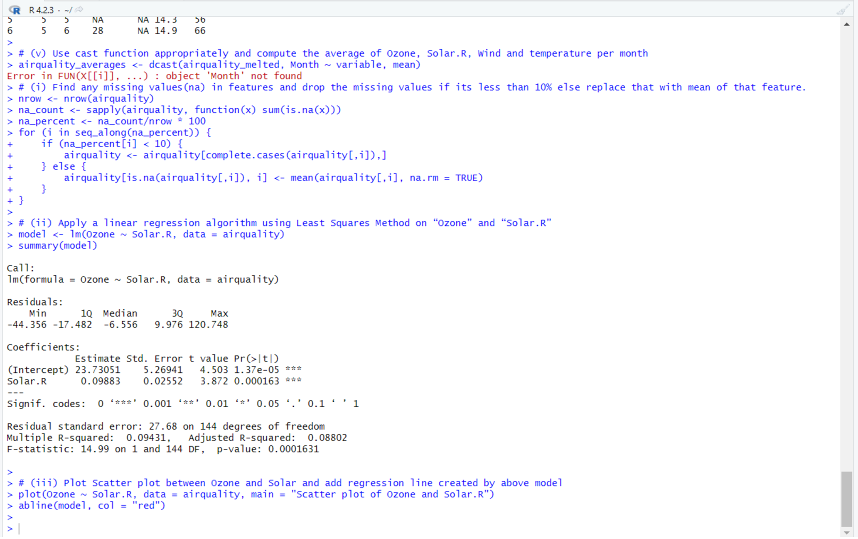
10%

else replace that with mean of that feature.

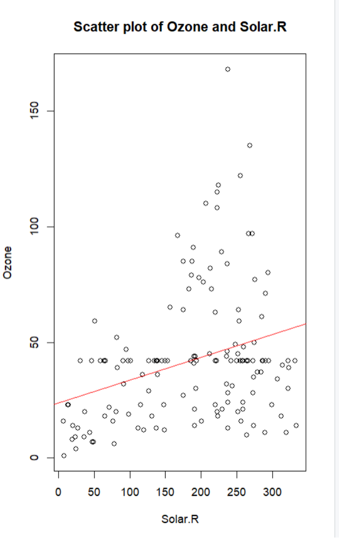
(ii) Apply a linear regression algorithm using Least Squares Method on “Ozone” and

“Solar.R”

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


R 4.2.3 · ~/

```
5       5   5     NA        NA 14.3   56
6       5   6     28        NA 14.9   66
>
> # (v) Use cast function appropriately and compute the average of Ozone, Solar.R, Wind and temperature per month
> airquality_averages <- dcast(airquality_melted, Month ~ variable, mean)
Error in FUN(X[[i]], ...) : object 'Month' not found
> # (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.
> nrow <- nrow(airquality)
> na_count <- sapply(airquality, function(x) sum(is.na(x)))
> na_percent <- na_count/nrow * 100
> for (i in seq_along(na_percent)) {
+     if (na_percent[i] < 10) {
+         airquality <- airquality[complete.cases(airquality[,i]),]
+     } else {
+         airquality[is.na(airquality[,i]), i] <- mean(airquality[,i], na.rm = TRUE)
+     }
+ }
>
> # (ii) Apply a linear regression algorithm using Least Squares Method on "Ozone" and "Solar.R"
> model <- lm(Ozone ~ Solar.R, data = airquality)
> summary(model)

Call:
lm(formula = Ozone ~ Solar.R, data = airquality)

Residuals:
    Min      1Q  Median      3Q     Max
-44.356 -17.482  -6.556   9.976 120.748

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.73051    5.26941   4.503 1.37e-05 ***
Solar.R      0.09883    0.02552   3.872 0.000163 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.68 on 144 degrees of freedom
Multiple R-squared:  0.09431,	Adjusted R-squared:  0.08802
F-statistic: 14.99 on 1 and 144 DF,  p-value: 0.0001631

>
> # (iii) Plot Scatter plot between Ozone and Solar and add regression line created by above model
> plot(Ozone ~ Solar.R, data = airquality, main = "Scatter plot of Ozone and Solar.R")
> abline(model, col = "red")
>
> |
```



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

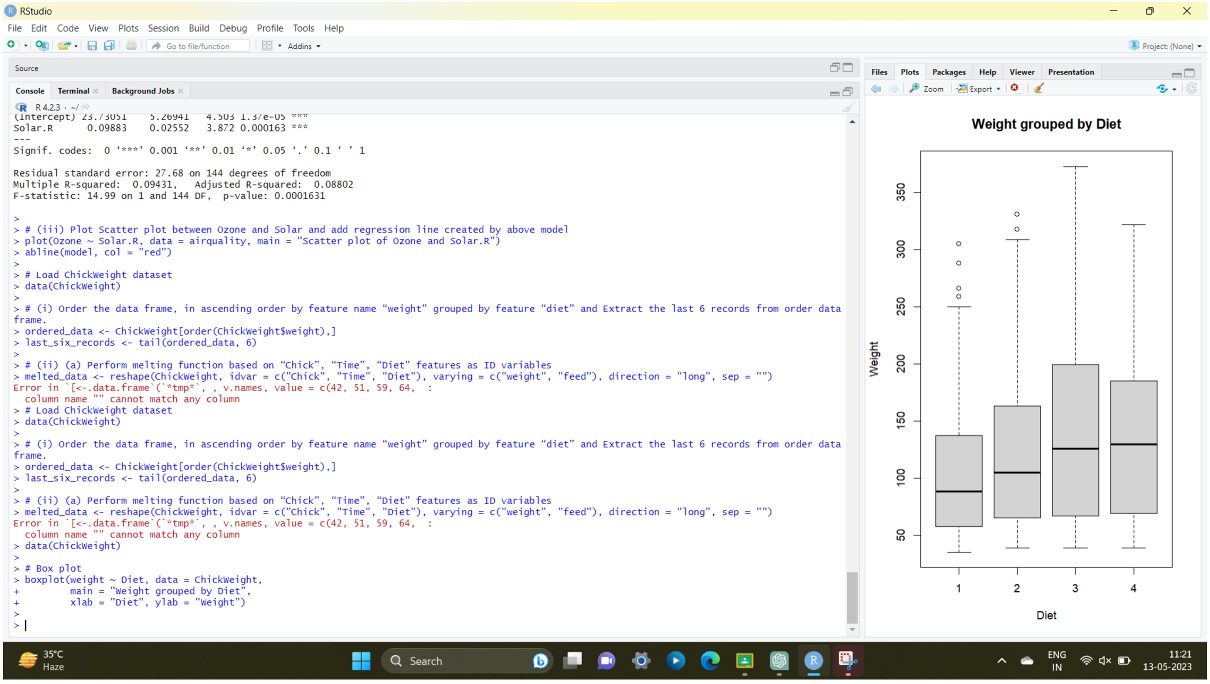
b. Perform cast function to display the mean value of weight grouped by Diet

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

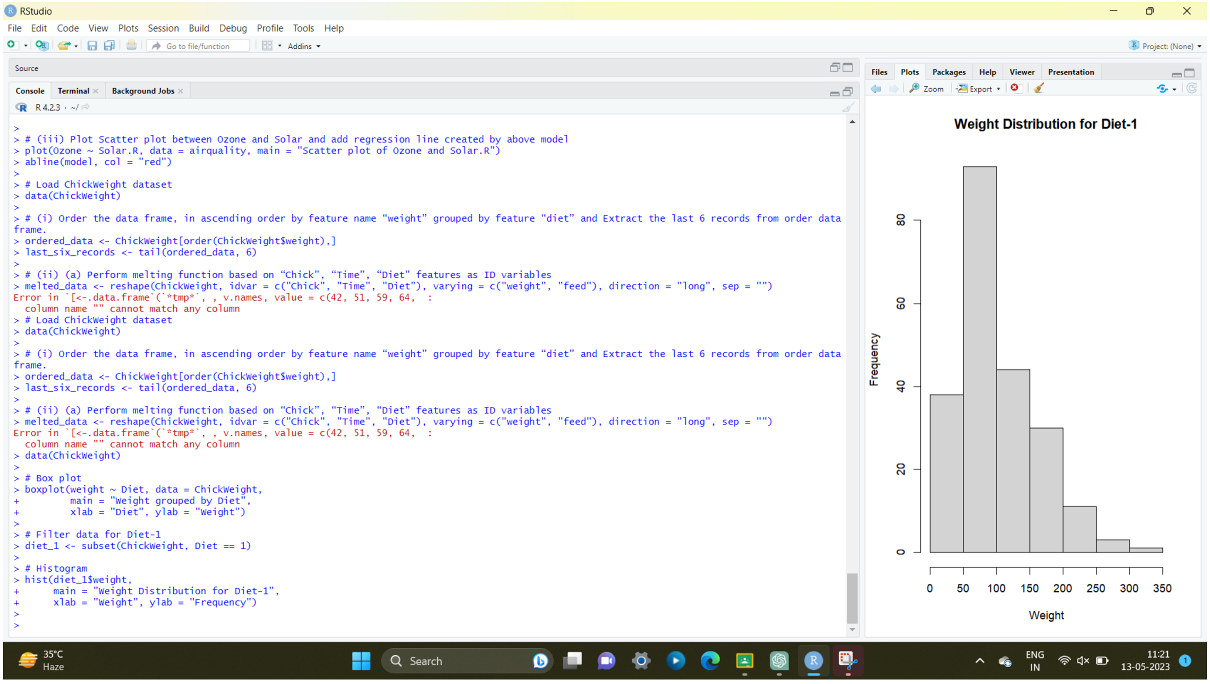
7. a. Create Box plot for “weight” grouped by “Diet”

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

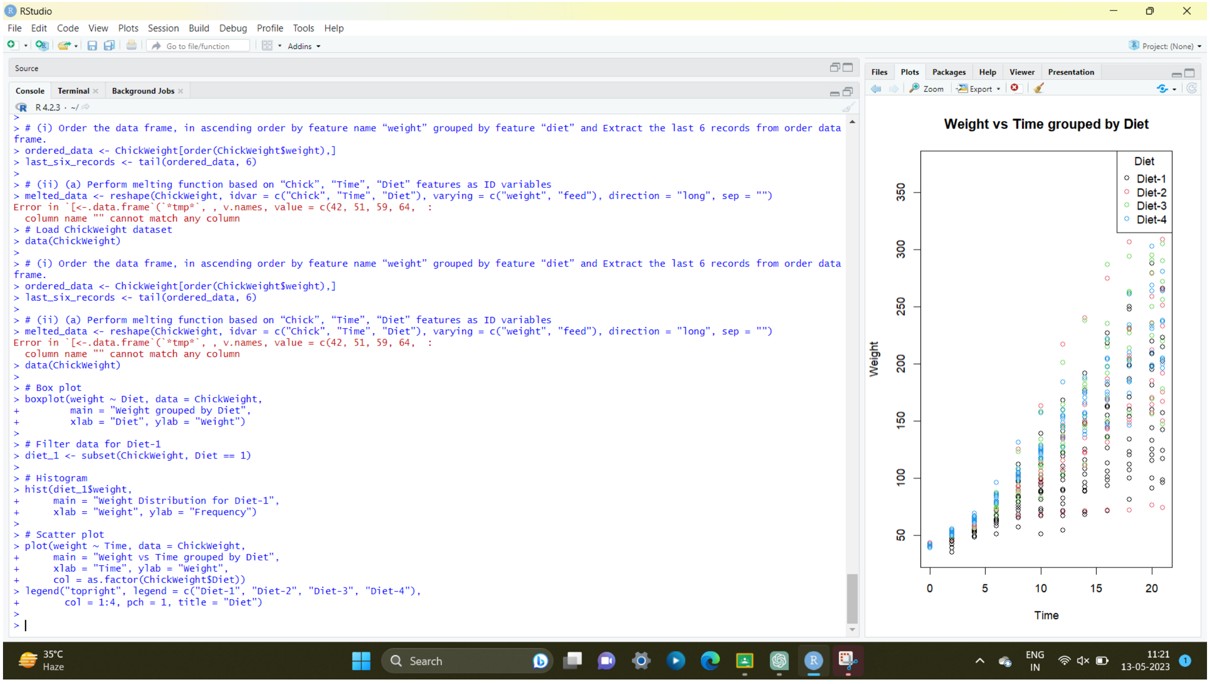
c. Create Scatter plot for “ weight” vs “Time” grouped by Diet



```
(Intercept) 23.73051    5.26941   4.503 1.37e-05 ***
Solar.R      0.09883    0.02552   3.872 0.000163 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.68 on 144 degrees of freedom
Multiple R-squared:  0.09431,	Adjusted R-squared:  0.08802 
F-statistic: 14.99 on 1 and 144 DF,  p-value: 0.0001631

> 
> # (iii) Plot Scatter plot between Ozone and Solar and add regression line created by above model
> plot(Ozone ~ Solar.R, data = airquality, main = "Scatter plot of Ozone and Solar.R")
> abline(model, col = "red")
> 
> # Load ChickWeight dataset
> data(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.
> ordered_data <- ChickWeight[order(ChickWeight$weight),]
> last_six_records <- tail(ordered_data, 6)
> 
> # (ii) (a) Perform melting function based on “Chick”, “Time”, “Diet” features as ID variables
> melted_data <- reshape(ChickWeight, idvar = c("Chick", "Time", "Diet"), varying = c("weight", "feed"), direction = "long", sep = "")
Error in `[<-.data.frame`(`*tmp*`, , v.names, value = c(42, 51, 59, 64,  : 
  column name "" cannot match any column
> # Load ChickWeight dataset
> data(ChickWeight)
> 
> # (i) Order the data frame, in ascending order by feature name “weight” grouped by feature “diet” and Extract the last 6 records from order data
frame.
> ordered_data <- ChickWeight[order(ChickWeight$weight),]
> last_six_records <- tail(ordered_data, 6)
> 
> # (ii) (a) Perform melting function based on “Chick”, “Time”, “Diet” features as ID variables
> melted_data <- reshape(ChickWeight, idvar = c("Chick", "Time", "Diet"), varying = c("weight", "feed"), direction = "long", sep = "")
Error in `[<-.data.frame`(`*tmp*`, , v.names, value = c(42, 51, 59, 64,  : 
  column name "" cannot match any column
> data(ChickWeight)
> 
> # Box plot
> boxplot(weight ~ Diet, data = ChickWeight,
+         main = "Weight grouped by Diet",
+         xlab = "Diet", ylab = "Weight")
> 
> |
```



```
> 
> # (iii) Plot Scatter plot between Ozone and Solar and add regression line created by above model
> plot(Ozone ~ Solar.R, data = airquality, main = "Scatter plot of Ozone and Solar.R")
> abline(model, col = "red")
> 
> # Load ChickWeight dataset
> data(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.
> ordered_data <- ChickWeight[order(ChickWeight$weight),]
> last_six_records <- tail(ordered_data, 6)
> 
> # (ii) (a) Perform melting function based on “Chick”, “Time”, “Diet” features as ID variables
> melted_data <- reshape(ChickWeight, idvar = c("Chick", "Time", "Diet"), varying = c("weight", "feed"), direction = "long", sep = "")
Error in `[<-.data.frame`(`*tmp*`, , v.names, value = c(42, 51, 59, 64,  : 
  column name "" cannot match any column
> # Load ChickWeight dataset
> data(ChickWeight)
> 
> # (i) Order the data frame, in ascending order by feature name “weight” grouped by feature “diet” and Extract the last 6 records from order data
frame.
> ordered_data <- ChickWeight[order(ChickWeight$weight),]
> last_six_records <- tail(ordered_data, 6)
> 
> # (ii) (a) Perform melting function based on “Chick”, “Time”, “Diet” features as ID variables
> melted_data <- reshape(ChickWeight, idvar = c("Chick", "Time", "Diet"), varying = c("weight", "feed"), direction = "long", sep = "")
Error in `[<-.data.frame`(`*tmp*`, , v.names, value = c(42, 51, 59, 64,  : 
  column name "" cannot match any column
> data(ChickWeight)
> 
> # Box plot
> boxplot(weight ~ Diet, data = ChickWeight,
+         main = "Weight grouped by Diet",
+         xlab = "Diet", ylab = "Weight")
> 
> # Filter data for Diet-1
> diet_1 <- subset(ChickWeight, Diet == 1)
> 
> # Histogram
> hist(diet_1$weight,
+      main = "Weight Distribution for Diet-1",
+      xlab = "Weight", ylab = "Frequency")
> 
> 
```



```
>
> # (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.
> ordered_data <- ChickWeight[order(ChickWeight$weight),]
> last_six_records <- tail(ordered_data, 6)
>
> # (ii) (a) Perform melting function based on “Chick”, “Time”, “Diet” features as ID variables
> melted_data <- reshape(ChickWeight, idvar = c("Chick", "Time", "Diet"), varying = c("weight", "feed"), direction = "long", sep = "")
Error in `[<-.data.frame`(`*tmp*`, , v.names, value = c(42, 51, 59, 64,  :
  column name "" cannot match any column
> # Load ChickWeight dataset
> data(ChickWeight)
>
> # (i) Order the data frame, in ascending order by feature name “weight” grouped by feature “diet” and Extract the last 6 records from order data
frame.
> ordered_data <- ChickWeight[order(ChickWeight$weight),]
> last_six_records <- tail(ordered_data, 6)
>
> # (ii) (a) Perform melting function based on “Chick”, “Time”, “Diet” features as ID variables
> melted_data <- reshape(ChickWeight, idvar = c("Chick", "Time", "Diet"), varying = c("weight", "feed"), direction = "long", sep = "")
Error in `[<-.data.frame`(`*tmp*`, , v.names, value = c(42, 51, 59, 64,  :
  column name "" cannot match any column
> data(ChickWeight)
>
> # Box plot
> boxplot(weight ~ Diet, data = ChickWeight,
+         main = "Weight grouped by Diet",
+         xlab = "Diet", ylab = "Weight")
>
> # Filter data for Diet-1
> diet_1 <- subset(ChickWeight, Diet == 1)
>
> # Histogram
> hist(diet_1$weight,
+      main = "Weight Distribution for Diet-1",
+      xlab = "Weight", ylab = "Frequency")
>
> # Scatter plot
> plot(weight ~ Time, data = ChickWeight,
+      main = "Weight vs Time grouped by Diet",
+      xlab = "Time", ylab = "Weight",
+      col = as.factor(ChickWeight$Diet))
> legend("topright", legend = c("Diet-1", "Diet-2", "Diet-3", "Diet-4"),
+        col = 1:4, pch = 1, title = "Diet")
>
> |
```