

**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:192011283

Name: Sreeja Yennam

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:

```
# Load the iris dataset
data(iris)
# Set the seed for reproducibility
set.seed(123)

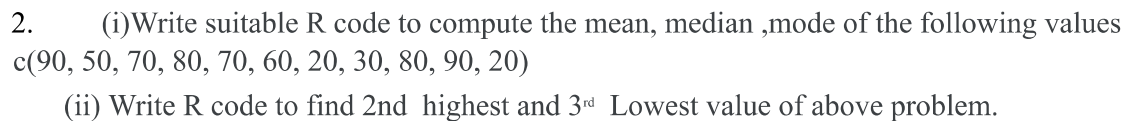
# Randomly sample the iris dataset
iris_sample <- iris[sample(nrow(iris)),]

# Split the data into training and test sets
train <- iris_sample[1:round(0.8*nrow(iris_sample)),]
test <- iris_sample[(round(0.8*nrow(iris_sample))+1):nrow(iris_sample),]

glm(Species ~ Petal.Length + Petal.Width, data = train, family = "binomial", method =
"Newton")
# Predict the probabilities of the model using the test set
probabilities <- predict(model, newdata = test, type = "response")
probabilities

# Convert probabilities to predicted species
predicted_species <- ifelse(probabilities > 0.5, "versicolor", "setosa")
```

```
# Print the confusion matrix
print(confusion_matrix)
```



```
#2i)
# Given values
x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)
```

```
# Compute the median
median_x <- median(x)
```

```
print(median_x)
```

```
# Compute the mode
```

```
mode_x <- names(table(x))[table(x) == max(table(x))]
```

```
print(mode_x)
```

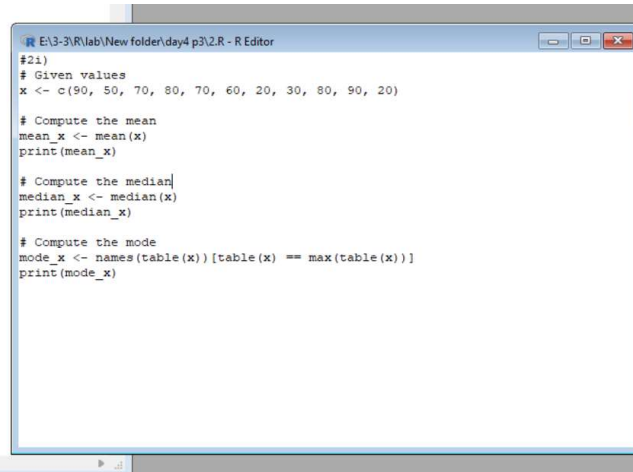
```
#21)
# Given values
x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

# Compute the mean
mean_x <- mean(x)
print(mean_x)
[1] 60

# Compute the median
median_x <- median(x)
print(median_x)
[1] 70

# Compute the mode
mode_x <- names(table(x))[table(x) == max(table(x))]
```

```
print(mode_x)
[1] "20" "70" "80" "90"
```



2II)

SOURCE CODE:

```
#2ii)# Given values
```

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

```
# Find the 2nd highest value
```

```
x_sorted <- sort(unique(x), decreasing = TRUE)
```

```
second_highest <- x_sorted[2]
```

```
print(second_highest)
```

```
# Find the 3rd lowest value
```

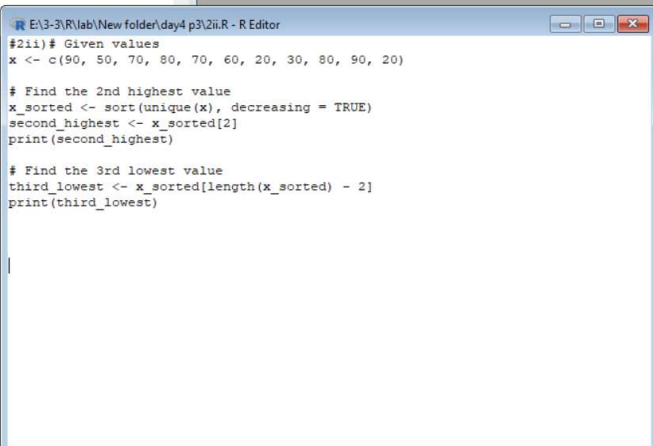
```
third_lowest <- x_sorted[length(x_sorted) - 2]
```

```
print(third_lowest)
```

```

>
>
> #2ii) # Given values
> x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)
>
> # Find the 2nd highest value
> x_sorted <- sort(unique(x), decreasing = TRUE)
> second_highest <- x_sorted[2]
> print(second_highest)
[1] 80
>
> # Find the 3rd lowest value
> third_lowest <- x_sorted[length(x_sorted) - 2]
> print(third_lowest)
[1] 50
>
>
>
>
>
>
>
>
>
>

```



```

E:\3-3\R\lab\New folder\day4 p3\2ii.R - R Editor
#2ii) # Given values
x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

# Find the 2nd highest value
x_sorted <- sort(unique(x), decreasing = TRUE)
second_highest <- x_sorted[2]
print(second_highest)

# Find the 3rd lowest value
third_lowest <- x_sorted[length(x_sorted) - 2]
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?

SOURCE CODE:

```

# I)Load the airquality dataset
data(airquality)

```

```

# Compute the mean temperature
mean_temp <- sum(airquality$Temp) / length(airquality$Temp)
print(mean_temp)

```

```
# II)Load the airquality dataset
```

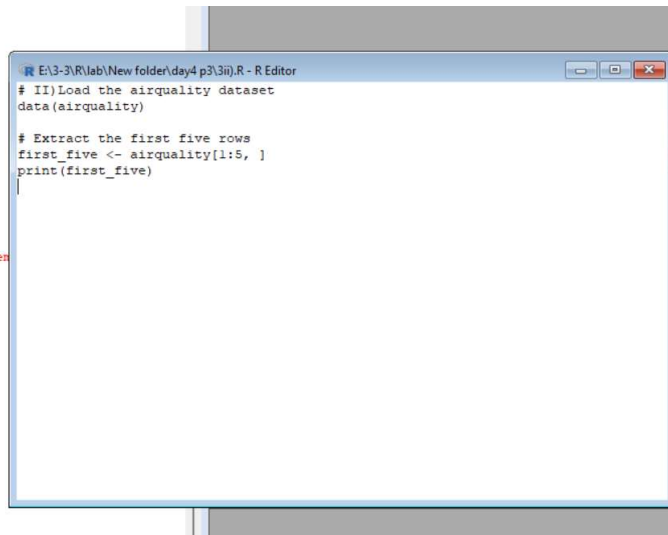
```
data(airquality)
```

```
# Extract the first five rows
```

```
first_five <- airquality[1:5, ]
```

```
print(first_five)
```

```
>
>
>
>
>
>
>
> # Load the airquality dataset
> data(airquality)
>
> # Compute the mean temperature
> mean_temp <- sum(airquality$Temp) / length(airquality$Temp)
> print(mean_temp)
[1] 77.88235
>
>
>
> # II)Load the airquality dataset
> data(airquality)
>
> # Extract the first five rows
> first_five <- airquality[1:5, ]
> print(first_five)
  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
```



```
E:\3-3\R\lab\New folder\day4 p3\3in.R - R Editor
# II)Load the airquality dataset
data(airquality)

# Extract the first five rows
first_five <- airquality[1:5, ]
print(first_five)
```

```
# III)Load the airquality dataset
```

```
data(airquality)
```

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

```
cols_to_keep <- c("Ozone", "Solar.R", "Month", "Day")
```

```
subset_data <- airquality[cols_to_keep]
```

```
print(subset_data)
```

```

> data(airquality)
> # Extract all columns except Temp and Wind
> cols_to_keep <- c("Ozone", "Solar.R", "Month", "Day")
> subset_data <- airquality[cols_to_keep]
> print(subset_data)
  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

```

```

R E:\3-3\lab\New folder\day4 p3\3iii\R - R Editor
# III)Load the airquality dataset
data(airquality)
# Extract all columns except Temp and Wind
cols_to_keep <- c("Ozone", "Solar.R", "Month", "Day")
subset_data <- airquality[cols_to_keep]
print(subset_data)

```

IV)Load the airquality dataset

```
data(airquality)
```

Find the coldest day

```
coldest_day <- airquality$Day[which.min(airquality$Temp)]
```

```
print(coldest_day)
```

```

> # IV)Load the airquality dataset
> data(airquality)
> # Find the coldest day
> coldest_day <- airquality$Day[which.min(airquality$Temp)]
> print(coldest_day)
[1] 5

```

```

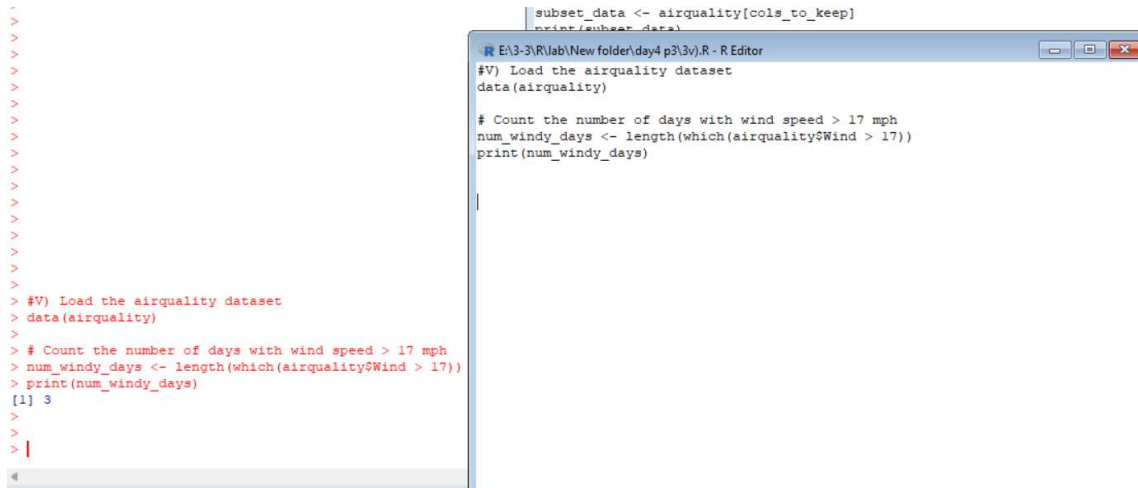
R E:\3-3\lab\New folder\day4 p3\3iv\R - R Editor
# IV)Load the airquality dataset
data(airquality)
# Find the coldest day
coldest_day <- airquality$Day[which.min(airquality$Temp)]
print(coldest_day)

```

#V) Load the airquality dataset

```
data(airquality)
```

```
# Count the number of days with wind speed > 17 mph
num_windy_days <- length(which(airquality$Wind > 17))
print(num_windy_days)
```



The image shows two windows from an R environment. On the left is the R console, and on the right is the R Editor. The console shows the execution of the code from the previous block, resulting in the output [1] 3. The R Editor shows the source code being executed.

```
subset_data <- airquality[cols_to_keep]
print(subset_data)

# E:\3-3\R\lab\New folder\day4 p3\3v\R - R Editor
#V) Load the airquality dataset
data(airquality)

# Count the number of days with wind speed > 17 mph
num_windy_days <- length(which(airquality$Wind > 17))
print(num_windy_days)
```

```
> #V) Load the airquality dataset
> data(airquality)
>
> # Count the number of days with wind speed > 17 mph
> num_windy_days <- length(which(airquality$Wind > 17))
> print(num_windy_days)
[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?

SOURCE CODE:

#i)

```
summary(airquality)
```

```

> # Count the number of days with wind speed > 17 mph
> num_windy_days <- length(which(airquality$Wind > 17))
> print(num_windy_days)
[1] 3
>
> #I
> summary(airquality)
      Ozone      Solar.R      Wind      Temp
Min.   : 1.00   Min.   : 7.0   Min.   : 1.700   Min.   :56.00
1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
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
> |

```

```

E:\3-3\lab\New folder\day4 p3\4ii).R - R Editor
#I
summary(airquality)

```

ii)

```

library(reshape2)
melted_airquality <- melt(airquality)
melted_airquality

```

```

> library(reshape2)
> melted_airquality <- melt(airquality)
No id variables; using all as measure variables
> melted_airquality
  variable value
1     Ozone  41.0
2     Ozone  36.0
3     Ozone  12.0
4     Ozone  18.0
5     Ozone   NA
6     Ozone  28.0
7     Ozone  23.0
8     Ozone  19.0
9     Ozone   8.0
10    Ozone   NA
11    Ozone   7.0
12    Ozone  16.0
13    Ozone  11.0
14    Ozone  14.0
15    Ozone  18.0
16    Ozone  14.0
17    Ozone  34.0
18    Ozone   6.0
19    Ozone  30.0
20    Ozone  11.0
21    Ozone   1.0
22    Ozone  11.0
23    Ozone   4.0
24    Ozone  32.0
25    Ozone   NA
26    Ozone   NA
27    Ozone   NA
28    Ozone  23.0
29    Ozone  45.0
30    Ozone 115.0
31    Ozone  37.0
32    Ozone   NA
33    Ozone   NA

```

```

E:\3-3\lab\New folder\day4 p3\4ii).R - R Editor
library(reshape2)
melted_airquality <- melt(airquality)
melted_airquality

```

iii)

```

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

```



```
> melted_airquality <- melt(airquality, id.vars = c("Month", "Day"))
> melted_airquality
```

	Month	Day	variable	value
1	5	1	Ozone	41.0
2	5	2	Ozone	36.0
3	5	3	Ozone	12.0
4	5	4	Ozone	18.0
5	5	5	Ozone	NA
6	5	6	Ozone	28.0
7	5	7	Ozone	23.0
8	5	8	Ozone	19.0
9	5	9	Ozone	8.0
10	5	10	Ozone	NA
11	5	11	Ozone	7.0
12	5	12	Ozone	16.0
13	5	13	Ozone	11.0
14	5	14	Ozone	14.0
15	5	15	Ozone	18.0
16	5	16	Ozone	14.0
17	5	17	Ozone	34.0
18	5	18	Ozone	6.0
19	5	19	Ozone	30.0
20	5	20	Ozone	11.0
21	5	21	Ozone	1.0
22	5	22	Ozone	11.0
23	5	23	Ozone	4.0
24	5	24	Ozone	32.0
25	5	25	Ozone	NA
26	5	26	Ozone	NA
27	5	27	Ozone	NA
28	5	28	Ozone	23.0
29	5	29	Ozone	45.0
30	5	30	Ozone	115.0
31	5	31	Ozone	37.0
32	6	1	Ozone	NA
33	6	2	Ozone	NA
34	6	3	Ozone	NA
35	6	4	Ozone	NA
36	6	5	Ozone	NA
37	6	6	Ozone	NA

```
E:\3-3\R\lab\New folder\day4 p3\4iii.R - R Editor
melted_airquality <- melt(airquality, id.vars = c("Month", "Day"))
melted_airquality
```

iv)

```
casted_airquality <- dcast(melted_airquality, Month + Day ~ variable)
```

```
casted_airquality
```

```
> casted_airquality <- dcast(melted_airquality, Month + Day ~ variable)
> casted_airquality
```

	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
7	5	7	23	299	8.6	65
8	5	8	19	99	13.8	59
9	5	9	8	19	20.1	61
10	5	10	NA	194	8.6	69
11	5	11	7	NA	6.9	74
12	5	12	16	256	9.7	69
13	5	13	11	290	9.2	66
14	5	14	14	274	10.9	68
15	5	15	18	65	13.2	58
16	5	16	14	334	11.5	64
17	5	17	34	307	12.0	66
18	5	18	6	78	18.4	57
19	5	19	30	322	11.5	68
20	5	20	11	44	9.7	62
21	5	21	1	8	9.7	59
22	5	22	11	320	16.6	73
23	5	23	4	25	9.7	61
24	5	24	32	92	12.0	61
25	5	25	NA	66	16.6	57
26	5	26	NA	266	14.9	58
27	5	27	NA	NA	8.0	57
28	5	28	23	13	12.0	67
29	5	29	45	252	14.9	81
30	5	30	115	223	5.7	79
31	5	31	37	279	7.4	76
32	6	1	NA	286	8.6	78
33	6	2	NA	287	9.7	74
34	6	3	NA	242	16.1	67
35	6	4	NA	186	9.2	84
..

```
E:\3-3\R\lab\New folder\day4 p3\4iv.R - R Editor
casted_airquality <- dcast(melted_airquality, Month + Day ~ variable)
casted_airquality
```

v)

```
# Load reshape2 package
```

```
library(reshape2)
```

```
# Melt airquality dataset
melted_airquality <- melt(airquality, id.vars = c("Month", "Day"))

# Cast molten airquality dataset with respect to month and date features
cast_airquality <- dcast(melted_airquality, Month + Day ~ variable)

# Compute the average of Ozone, Solar.R, Wind and temperature per month
average_airquality <- aggregate(cast_airquality[, c("Ozone", "Solar.R", "Wind", "Temp")],
                               by = list(cast_airquality$Month), mean)
names(average_airquality)[1] <- "Month"
average_airquality
```

```
# Load reshape2 package
library(reshape2)

# Melt airquality dataset
melted_airquality <- melt(airquality, id.vars = c("Month", "Day"))

# Cast molten airquality dataset with respect to month and date features
cast_airquality <- dcast(melted_airquality, Month + Day ~ variable)

# Compute the average of Ozone, Solar.R, Wind and temperature per month
average_airquality <- aggregate(cast_airquality[, c("Ozone", "Solar.R", "Wind", "Temp")],
                               by = list(cast_airquality$Month), mean)
names(average_airquality)[1] <- "Month"
average_airquality
```

Month	Ozone	Solar.R	Wind	Temp
5	NA	NA	11.622581	65.54839
6	NA	190.1667	10.266667	79.10000
7	NA	216.4839	8.941935	83.90323
8	NA	NA	8.793548	83.96774
9	NA	167.4333	10.180000	76.90000

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:

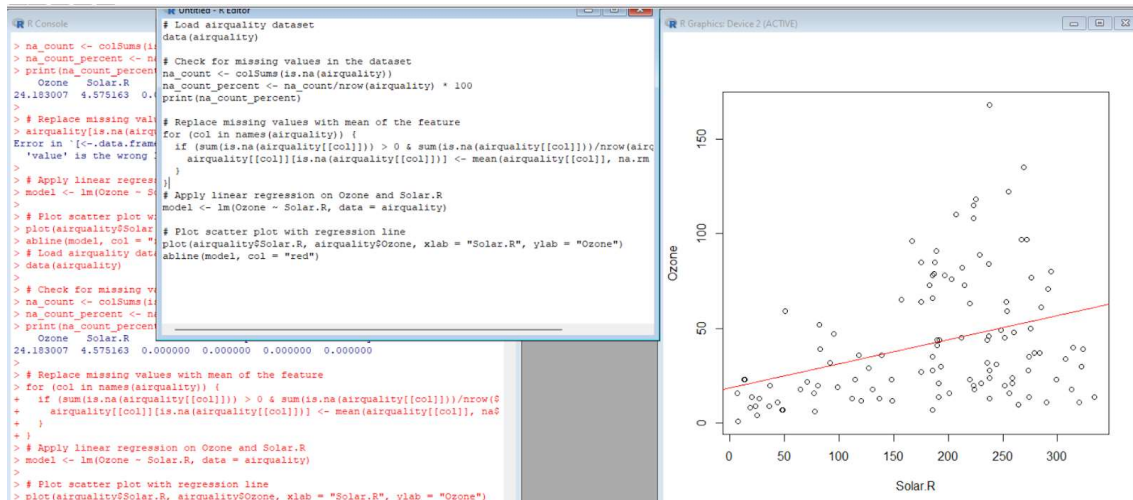
```
# Load airquality dataset
data(airquality)

# Check for missing values in the dataset
na_count <- colSums(is.na(airquality))
na_count_percent <- na_count/nrow(airquality) * 100
print(na_count_percent)
```

```
# Replace missing values with mean of the feature
for (col in names(airquality)) {
  if (sum(is.na(airquality[[col]])) > 0 & sum(is.na(airquality[[col]]))/nrow(airquality) < 0.1) {
    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 with regression line
plot(airquality$Solar.R, airquality$Ozone, xlab = "Solar.R", ylab = "Ozone")
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”, “Time”, “Diet” 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:

```
# 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
```

```

ordered_data <- ChickWeight[order(ChickWeight$weight), ]
last_6 <- tail(ordered_data, 6)

# (ii) Melt and cast functions
library(reshape2)

# (ii) a. Melt function
melted_chickweight <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))

# (ii) b. Cast function to display the mean value of weight grouped by Diet
mean_weight <- cast(melted_chickweight, Diet ~ ., mean)

# (ii) c. Cast function to display the mode of weight grouped by Diet
library(modeest)
mode_weight <- cast(melted_chickweight, Diet ~ ., modeest::mfv)

```

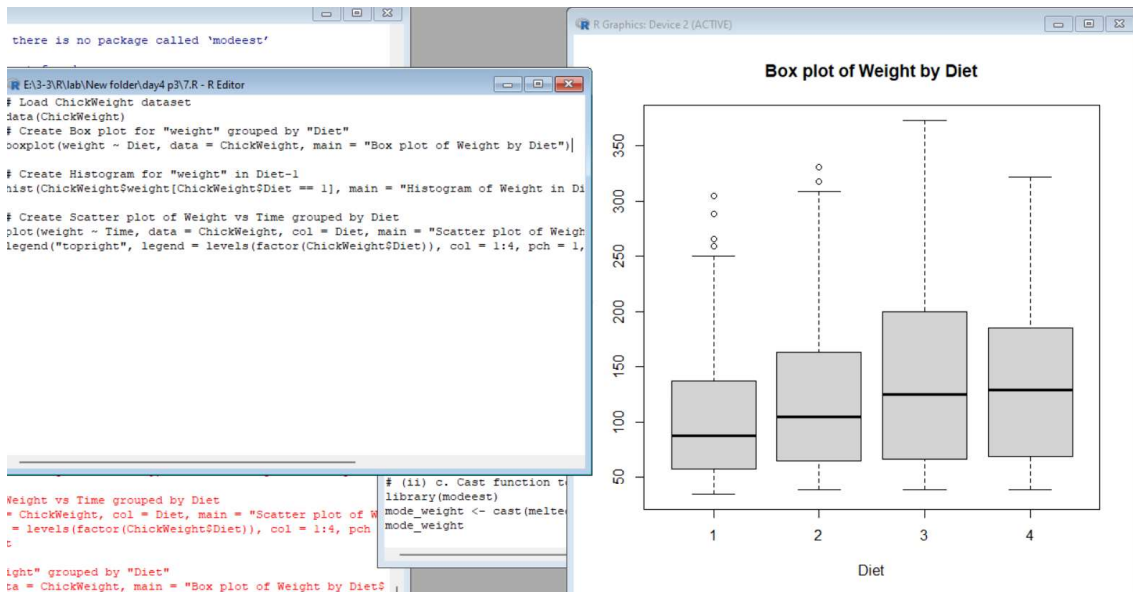
7. a. Create Box plot for “weight” grouped by “Diet”
 - b. Create a Histogram for “weight” features belong to Diet- 1 category
 - c. Create Scatter plot for “ weight” vs “Time” grouped by Diet

SOURCE CODE:

```

A.
# Load ChickWeight dataset
data(ChickWeight)
# Create Box plot for "weight" grouped by "Diet"
boxplot(weight ~ Diet, data = ChickWeight, main = "Box plot of Weight by Diet")

```



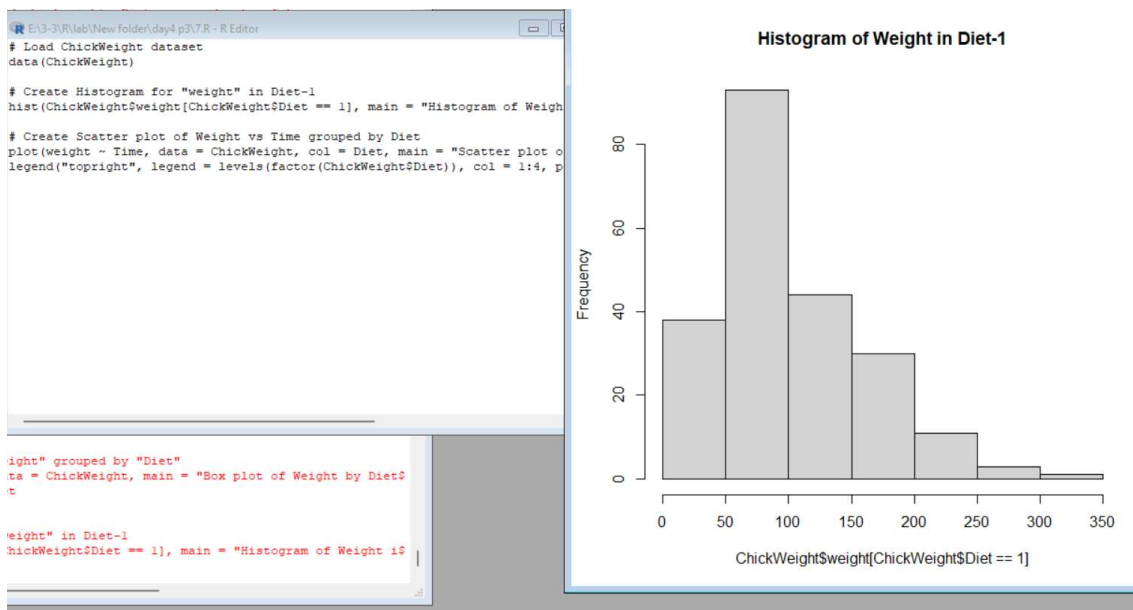
b.

```
# Load ChickWeight dataset
```

```
data(ChickWeight)
```

```
# Create Histogram for "weight" in Diet-1
```

```
hist(ChickWeight$weight[ChickWeight$Diet == 1], main = "Histogram of Weight in Diet-1")
```



c.

```
# Load ChickWeight dataset
```

```
data(ChickWeight)
```

```
# Create Scatter plot of Weight vs Time grouped by Diet
```

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

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

```
E:\3-3\lab\New folder\day4\p3\7.R - R Editor
# Load ChickWeight dataset
data(ChickWeight)
# Create Scatter plot of Weight vs Time grouped by Diet
plot(weight ~ Time, data = ChickWeight, col = Diet, main = "Scatter plot of Weight vs Time  
by Diet", xlab = "Time", ylab = "Weight")
legend("topright", legend = levels(factor(ChickWeight$Diet)), col = 1:4, pch = 1, title =  
"Diet")
# Create Histogram for "weight" in Diet-1
hist(ChickWeight$weight[ChickWeight$Diet == 1], main = "Histogram of Weight in Diet-1",  
xlab = "Weight", ylab = "Frequency")

# for "weight" in Diet-1
weight[ChickWeight$Diet == 1, main = "Histogram of Weight in Diet-1",  
: dataset

# Plot of Weight vs Time grouped by Diet
plot(weight ~ Time, data = ChickWeight, col = Diet, main = "Scatter plot of Weight vs Time  
by Diet", xlab = "Time", ylab = "Weight")
legend("topright", legend = levels(factor(ChickWeight$Diet)), col = 1:4, pch = 1, title =  
"Diet")
```

