Omkar Gurav
TE IT Batch T3
Roll No:- 8048

Assignment - 6

- Aim:- Perform the following operations using R on the Air quality data set
  1. Data Cleaning
  2. Data integration
  3. Data transformation
  4. Error correcting
  5. Data model building (regression model for prediction of ozone value)

- Theory:-

Q.1. What is data cleaning or data preparation?

→ Data cleaning is a process of finding the incorrect or corrupted data and removing it.

The data cleaning process is required because incorrect data can produce the wrong conclusions and bad analysis, particularly when considered the massive quantities of Big Data.

Issues in data cleaning:-

1. Lack of validation
6. Dates

② Data from different sources

③ Personal names

④ Locations

⑤ Languages

⑦ Numbers

⑧ Currencies

⑨ Other issues

Cleaning Methods:-

① Histograms - Find out which values used less frequently & hence becomes invalid.

② Conversion Tables - Situation where data issues are already known the conversion tables can be preferable

③ Tools - Variety of vendors such as IBM, SAS, Oracle & Lavastorm Analytics offers solutions for data cleaning.

④ Algorithms - To fix some of the data, spell checking or phonetic algorithms are useful.

⑤ Manually - Even if there are multiple tools and algorithms are available, human interference is still required to understand & fix the data.

Q.2. Explain following functions in R

a. na.omit()

→ na.omit() function is used to omit all unnecessary cases from dataframe, matrix or vector.

If na.omit() removes cases, the row numbers of the cases form the 'na.action' attribute of the result of class 'omit'.

Syntax : na.omit(data)

data: set of specified values of dataframe, matrix or vector

Returns: Range of values after NA omission

Eg : na.omit(Data)

b. rbind

→ The name of rbind function stands for row-bind.
The rbind function can be used to combine several vectors, matrices and/or data frames by row.

Syntax : rbind(x, x1)

x : the input data

x1: The data need to be binded

Eg : x ← 2:4
x1 ← 5:7
rbind(x,x1)

The rbind is used to bind or combine multiple group of rows together.

c. cbind

→ The name of the cbind function stands for column-bind. The cbind function is used to combine vectors, matrices and/or data frames by columns.

Syntax : cbind(x,x1)
x: The input data
x1: The data need to be binded

Eg : x ← data.frame(x2=c(7,3,2,9,0), x3=c(4,4,1,8,4))
x1 ← c(9,5,7,1,8)
cbind(x,x1)

- Conclusion:- Thus we have successfully performed operations using R on the Air quality data set.

airquality.R

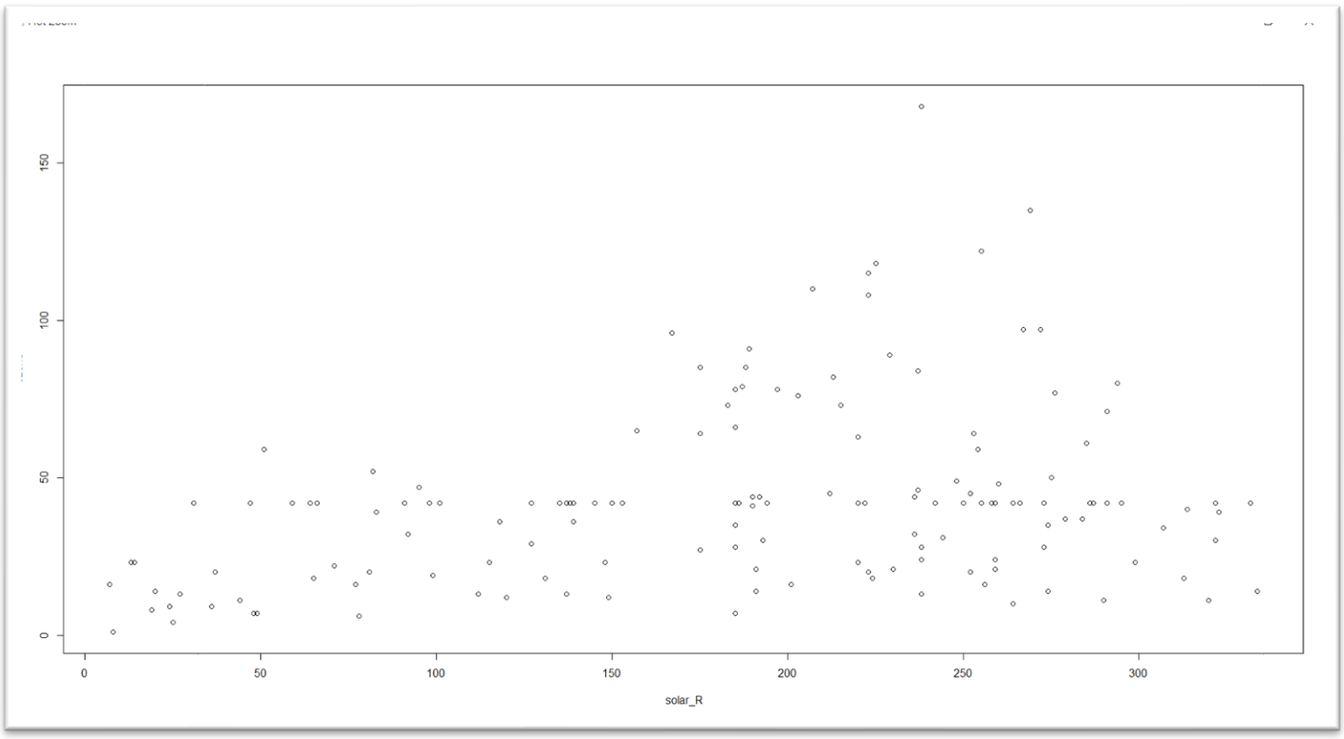
```
# Loading airquality dataset
data("airquality")
my_airquality_data <- airquality

# Summarising dataset
summary(my_airquality_data)



# Data cleaning

 # Solution 1 to remove NA values : Omit rows containing NA value
 copy <- airquality
 nrow(copy)

 nrow(na.omit(copy))

 # Solution 2 to remove NA values : Replace NA value with mean value

 my_airquality_data$Ozone[is.na(my_airquality_data$Ozone)] <- as.integer(mean(my_airquality_data$Ozone, na.rm = TRUE))

 my_airquality_data$Solar.R[is.na(my_airquality_data$Solar.R)] <- as.integer(mean(my_airquality_data$Solar.R, na.rm = TRUE))

 # Checking if there are any NA values
 sum(is.na(my_airquality_data))



# Data integration

 my_airquality_data_subset_1 <- my_airquality_data[1:10, c(2,3)]

 my_airquality_data_subset_2 <- my_airquality_data[1:10, c(4,5)]

 cbind(my_airquality_data_subset_1, my_airquality_data_subset_2)



# Data transformation
```

```
copy$Month <- month.abb[copy$Month]
str(copy)



# Data model building (regression model for prediction of Ozone value)

# Setting predictor attribute
solar_R <- my_airquality_data[, "Solar.R"]

# Setting target attribute
ozone <- my_airquality_data[, "Ozone"]

plot(ozone~solar_R)

# Fitting linear model
model_ozone_solar_R <- lm(ozone~solar_R)
model_ozone_solar_R       # Gives values of y-intercept and slope

abline(model_ozone_solar_R, col="blue")

# Prediction of 'Ozone' when 'Solar.R'= 10
p1 <- predict(model_ozone_solar_R, data.frame("solar_R" = 10))
p1
```

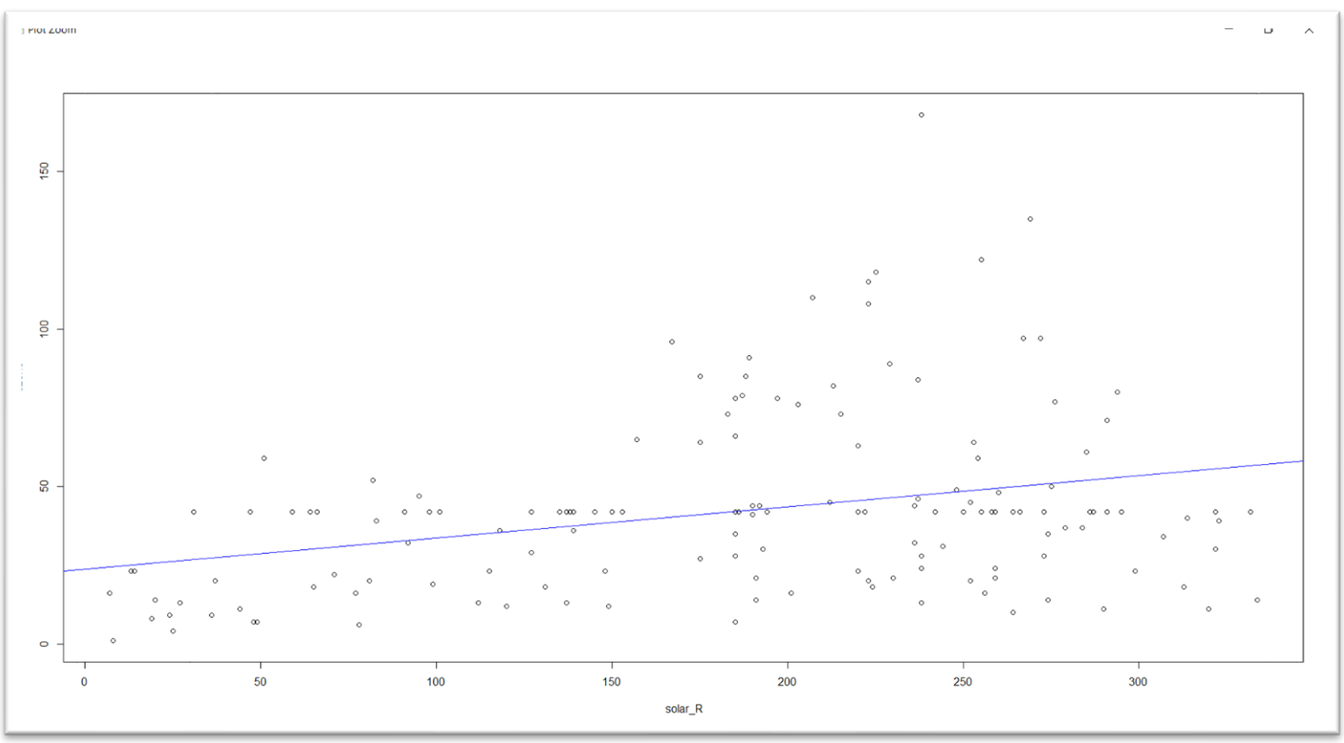
Output :

```
>   # Loading airquality dataset
>   data("airquality")
>   my_airquality_data <- airquality
>   # Summarising dataset
>   summary(my_airquality_data)
     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
>    # Solution 1 to remove NA values : Omit rows containing NA value
```

```
>   copy <- airquality

>   nrow(copy)

[1] 153

>   nrow(na.omit(copy))

[1] 111

>   my_airquality_data$Ozone[is.na(my_airquality_data$Ozone)] <- as.integer(mean(my_airquality_data$Ozone, na.rm = TRUE))

>   my_airquality_data$Solar.R[is.na(my_airquality_data$Solar.R)] <- as.integer(mean(my_airquality_data$Solar.R, na.rm = TRUE))

>   # Checking if there are any NA values

>   sum(is.na(my_airquality_data))

[1] 0

>   my_airquality_data_subset_1 <- my_airquality_data[1:10, c(2,3)]

>   my_airquality_data_subset_2 <- my_airquality_data[1:10, c(4,5)]

>   cbind(my_airquality_data_subset_1, my_airquality_data_subset_2)

   Solar.R Wind Temp Month

1     190  7.4   67     5

2     118  8.0   72     5

3     149 12.6   74     5

4     313 11.5   62     5

5     185 14.3   56     5

6     185 14.9   66     5

7     299  8.6   65     5

8      99 13.8   59     5

9      19 20.1   61     5

10    194  8.6   69     5

>   copy$Month <- month.abb[copy$Month]

>   str(copy)

'data.frame':     153 obs. of  6 variables:

 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...

 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...

 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...

 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...

 $ Month  : chr  "May" "May" "May" "May" ...

 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...

>   # Setting predictor attribute

>   solar_R <- my_airquality_data[, "Solar.R"]

>   # Setting target attribute

>   ozone <- my_airquality_data[, "Ozone"']

>   plot(ozone~solar_R)
```




```
> # Fitting linear model
> model_ozone_solar_R <- lm(ozone~solar_R)
> model_ozone_solar_R       # Gives values of y-intercept and slope

Call:
lm(formula = ozone ~ solar_R)

Coefficients:
(Intercept)      solar_R
   23.72956      0.09881

> abline(model_ozone_solar_R, col="blue")
```



```
>   # Prediction of 'Ozone' when 'Solar.R'= 10
>   p1 <- predict(model_ozone_solar_R, data.frame("solar_R" = 10))
>   p1
      1
24.71771
```