PROGRAM NO: 01 DATE:	**Data Pre-Processing using R Tool**	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
Dataset=read.csv('Dataset.csv')

Dataset$Age=ifelse(is.na(Dataset$Age),ave(Dataset$Age, FUN= function(x)
mean(x,na.rm=TRUE)),Dataset$Age)

Dataset$Salary=ifelse(is.na(Dataset$Salary),ave(Dataset$Salary, FUN= function(x)
mean(x,na.rm=TRUE)),Dataset$Salary)

Dataset$Country=factor(Dataset$Country,levels=c('France','Spain','Germany'),labels=c(1,2,3))

Dataset$Purchased=factor(Dataset$Purchased,levels=c('No','Yes'),labels=c(0,1))

install.packages('caTools')

library('caTools')

set.seed(123)

split=sample.split(Dataset,SplitRatio = 0.7)

training_set=subset(Dataset,split==TRUE)

test_set=subset(Dataset,split==FALSE)

training_set[,2:3]=scale(training_set[,2:3])

test_set[,2:3]=scale(test_set[,2:3])

print(training_set)

print(test_set)
```

OUTPUT:

```
> print(training_set)
  Country         Age     Salary Purchased
1       1  1.04208619  1.3707154         0
3       3 -1.33290093 -0.6259336         0
5       3  0.36351844  0.1030017         1
7       2 -0.07270369 -0.8477835         0
> print(test_set)
  Country        Age      Salary Purchased
2       2 -1.1521431 -1.04466136         1
4       2  0.1152143 -0.03869116         0
6       1 -0.2304286 -0.27083813         1
8       1  1.2673574  1.35419066         1
```

RESULT:

PROGRAM NO: 02 DATE:	**Data Visualization Technique using R Tool**	PAGE NO:

AIM:

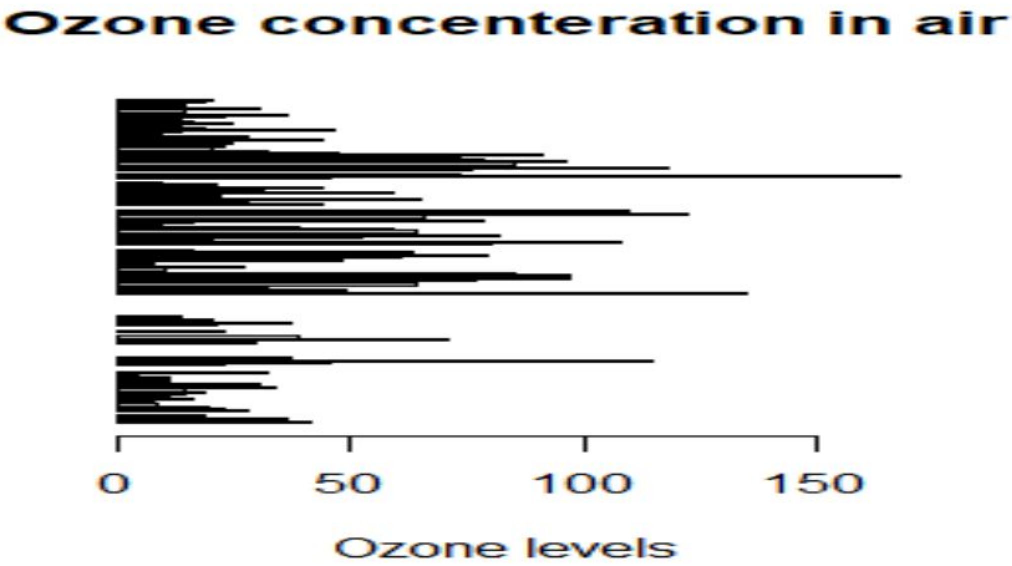
ALGORITHM:

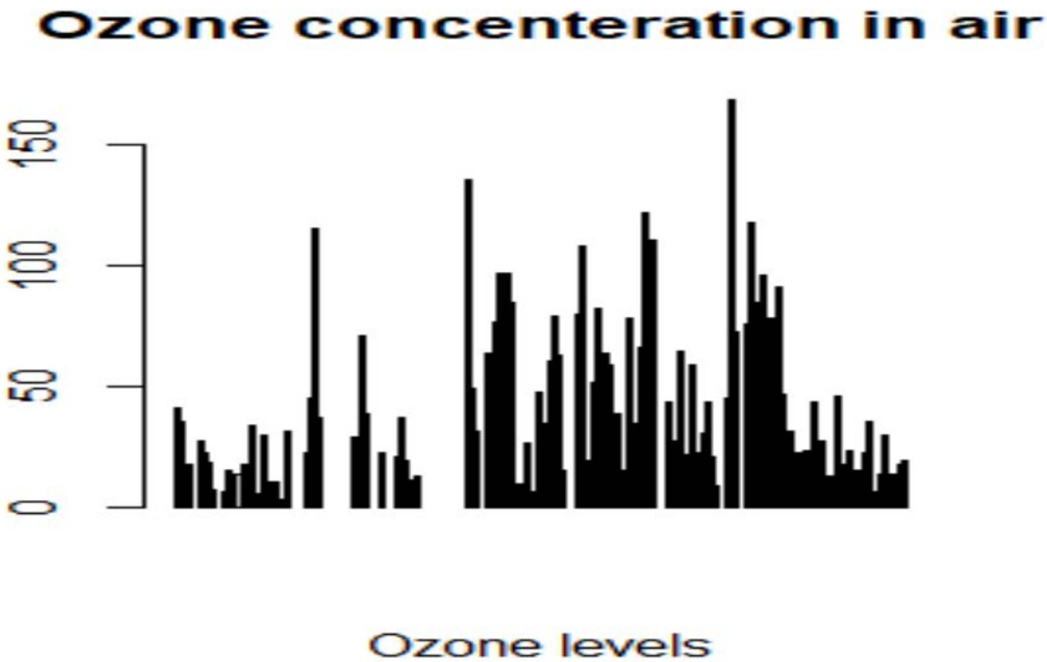
SOURCE CODE:

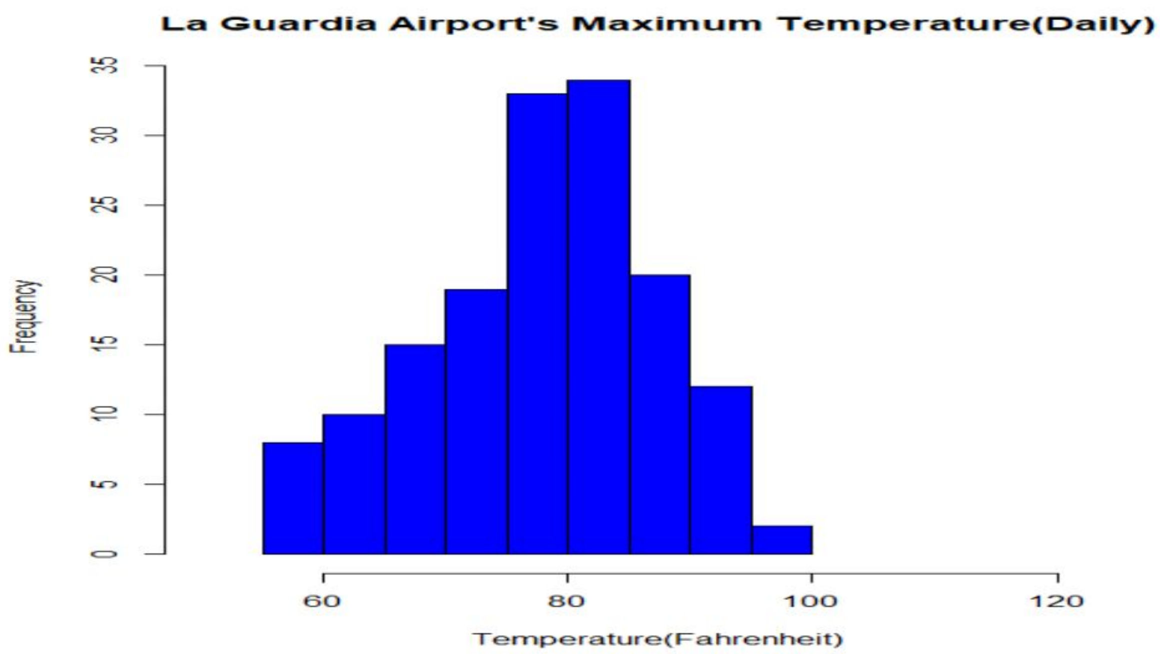
```
data("airquality")
#Barplot
barplot(airquality$Ozone,
      main ='Ozone concenteration in air',
      xlab ='Ozone levels',horiz = TRUE)
barplot(airquality$Ozone,
      main='Ozone concenteration in air',
      xlab='Ozone levels',horiz = FALSE)
#Histogram
hist(airquality$Temp, main = "La Guardia Airport's\ Maximum Temperature(Daily)",
    xlab = "Temperature(Fahrenheit)",
    xlim = c(50, 125),
    col = "blue", freq = TRUE)
#Boxplot
boxplot(airquality$Wind,main = "Average wind speed\ at La Guardia Airport",
      xlab = "miles per hour",ylab = "wind",
      col = "orange",border = "brown",
      horizontal = TRUE,notch = TRUE)
boxplot(airquality[,0:4],
      main = 'box plots for air parameters')
#ScatterPlot
plot(airquality$Ozone,airquality$month,
    main = "scatterplot example",
    xlab = "Ozone concentration in parts per billion",
    ylab = "month of observation",pch = 19)
#Heatmap
Ozone_miss = which(is.na(airquality$Ozone))
```

```
airquality$Ozone[Ozone_miss] = mean(airquality$Ozone,na.rm = TRUE)
Solar.R_miss = which(is.na(airquality$Solar.R))
airquality$Solar.R[Solar.R_miss] = mean(airquality$Solar.R,na.rm = TRUE)
data<-matrix(rnorm(50,0,5),nrow=5,ncol=5)
colnames(data)<-paste0("col",1:5)
rownames(data)<-paste0("row",1:5)
heatmap(data)
```

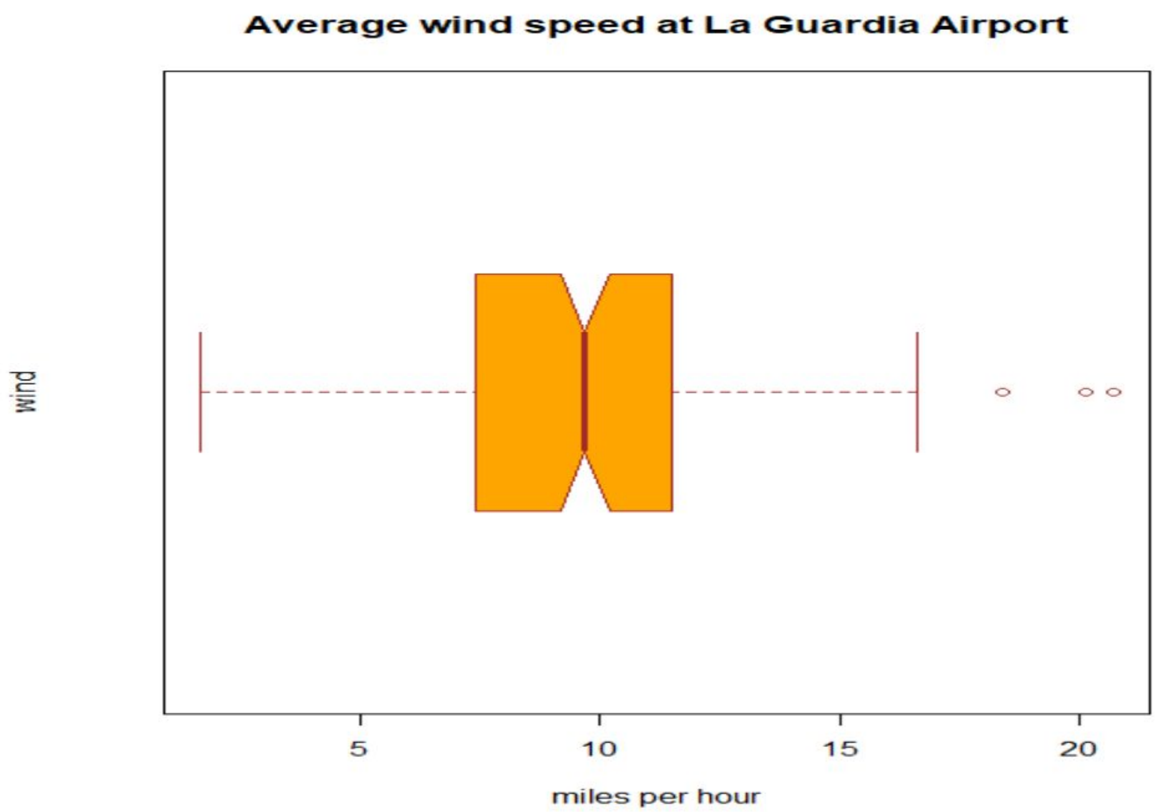
OUTPUT:


Ozone concenteration in air
0
50
100
150
Ozone levels

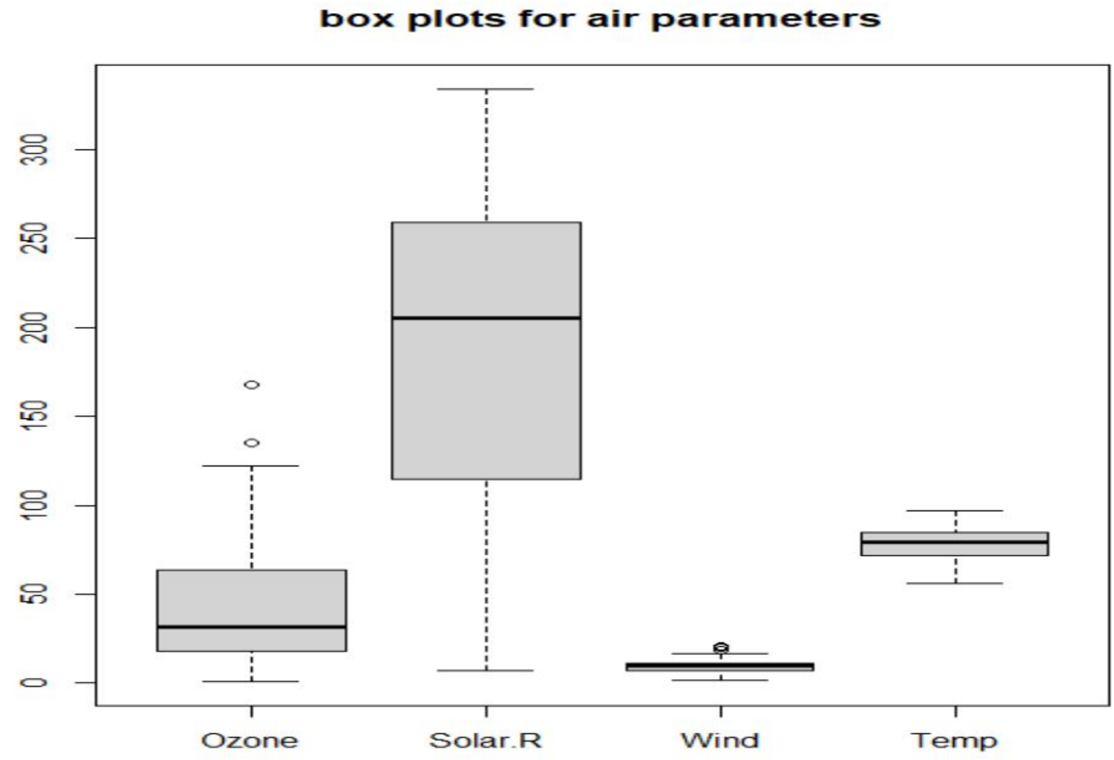

Ozone concenteration in air
0
50
100
150
Ozone levels

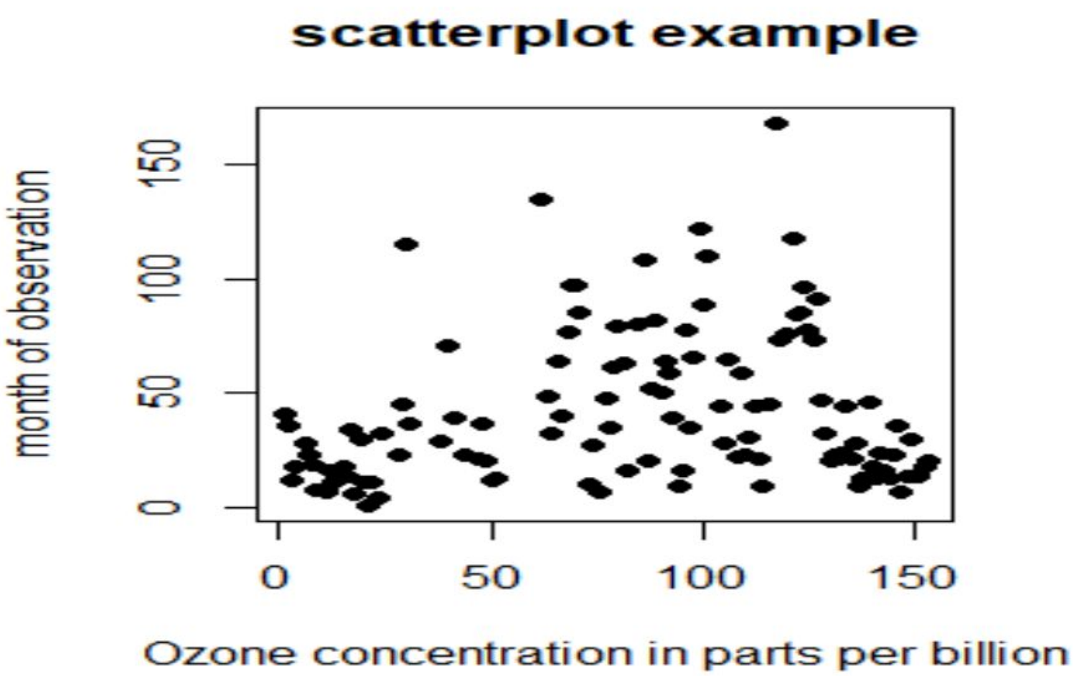


La Guardia Airport's Maximum Temperature(Daily)
Frequency
0
5
10
15
20
25
30
35
60
80
100
120
Temperature(Fahrenheit)

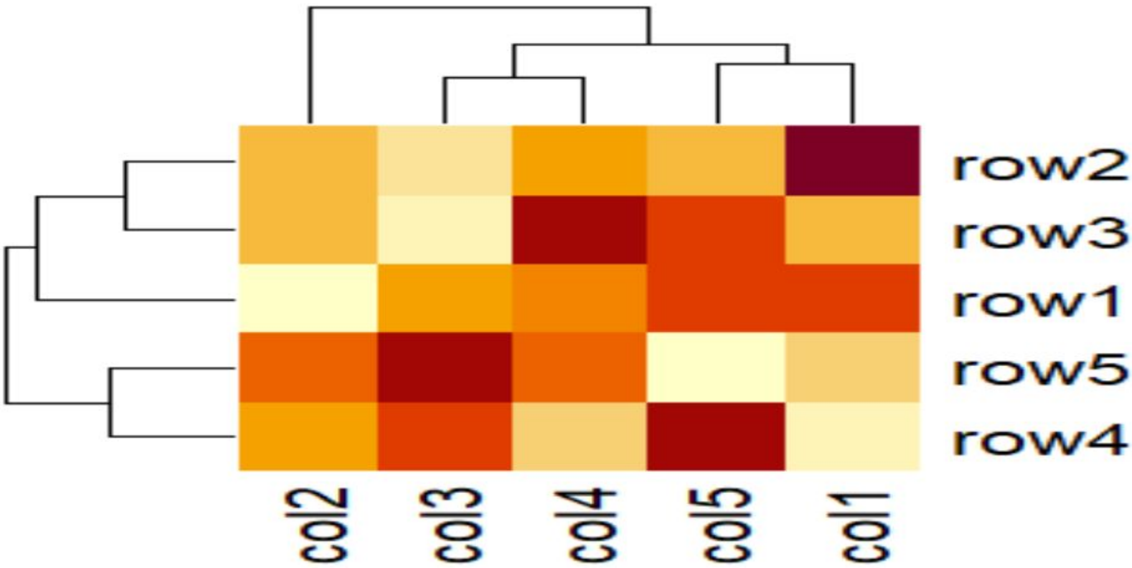


Average wind speed at La Guardia Airport
wind
5
10
15
20
miles per hour


box plots for air parameters
0
50
100
150
200
250
300
Ozone
Solar.R
Wind
Temp


scatterplot example
month of observation
0
50
100
150
0
50
100
150
Ozone concentration in parts per billion

```
> heatmap(data)
```



RESULT:

PROGRAM NO: 03 DATE:	**Implementing Correlation, linear regression using R Tool**	**PAGE NO:**

AIM

ALGORITHM

SOURCE CODE:

```
df = trees[2:3]
df
result = cor(df$Height, df$Volume, method = "pearson")
cat("Pearson correlation coefficient is:", result)
install.packages('caTools')
library(caTools)
split = sample.split(df$Height, SplitRatio = 0.7)
trainingset = subset(df, split == TRUE)
testset = subset(df, split == FALSE)
lm.r= lm(formula = Height ~ Volume,data = trainingset)
coef(lm.r)
ypred = predict(lm.r, newdata = testset)
summary(lm.r)
```

OUTPUT:

```
> cat("Pearson correlation coefficient is:", result)
Pearson correlation coefficient is: 0.5982497>
```

```
> coef(lm.r)
(Intercept)      Volume
 69.8656605   0.2222897
```

```
> summary(lm.r)

Call:
lm(formula = Height ~ Volume, data = trainingset)

Residuals:
   Min     1Q Median     3Q    Max
-7.333 -2.335  0.018  1.377 10.044

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 69.86566    2.23159  31.308   <2e-16 ***
Volume       0.22229    0.06385   3.481   0.0025 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.944 on 19 degrees of freedom
Multiple R-squared:  0.3895,	Adjusted R-squared:  0.3573
F-statistic: 12.12 on 1 and 19 DF,  p-value: 0.002498
```

RESULT:

PROGRAM NO: 04 DATE:	Classification using R Tool	PAGE NO:

AIM

ALGORITHM

SOURCE CODE:

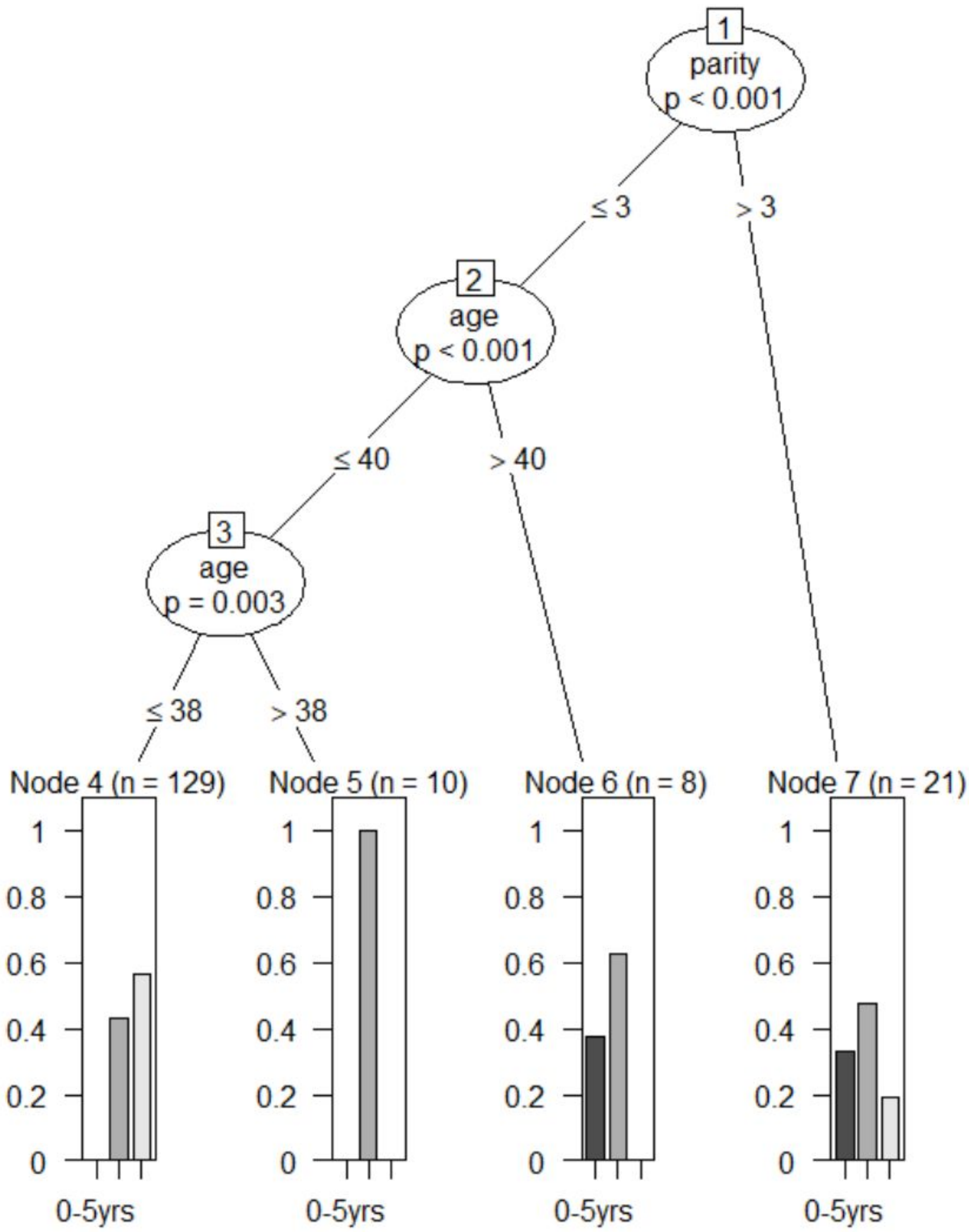
```
install.packages("party")
library(party)
df = infert
View(df)
ind <-sample(2,nrow(df),replace = TRUE,prob = c(0.7,0.30))
train.data <- df[ind==1,]
test.data <- df[ind==2,]
myf <- education~age+parity+induced+case
infert_ctree <- ctree(myf , data = train.data)
table(predict(infert_ctree) , train.data$education)
plot(infert_ctree)
testpred<-predict(infert_ctree,newdata = test.data)
table(testpred,test.data$education)
```

OUTPUT:

```
> table(predict(infert_ctree), train.data$education)

          0-5yrs 6-11yrs 12+ yrs
  0-5yrs       0       0       0
  6-11yrs     10      25       4
  12+ yrs      0      56      73
```

```
> plot(infert_ctree)
```



```
> table(testpred,test.data$education)

testpred  0-5yrs 6-11yrs 12+ yrs
  0-5yrs       0       0       0
  6-11yrs      2      11       4
  12+ yrs      0      28      35
```

RESULT:

PROGRAM NO: 05 DATE:	Clustering using R Tool	PAGE NO:

AIM

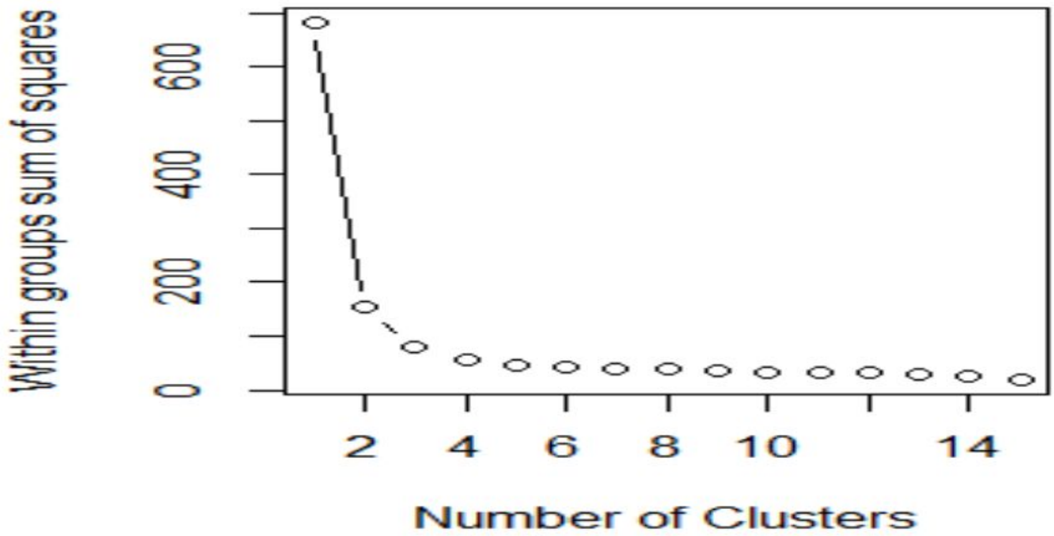
ALGORITHM

SOURCE CODE:

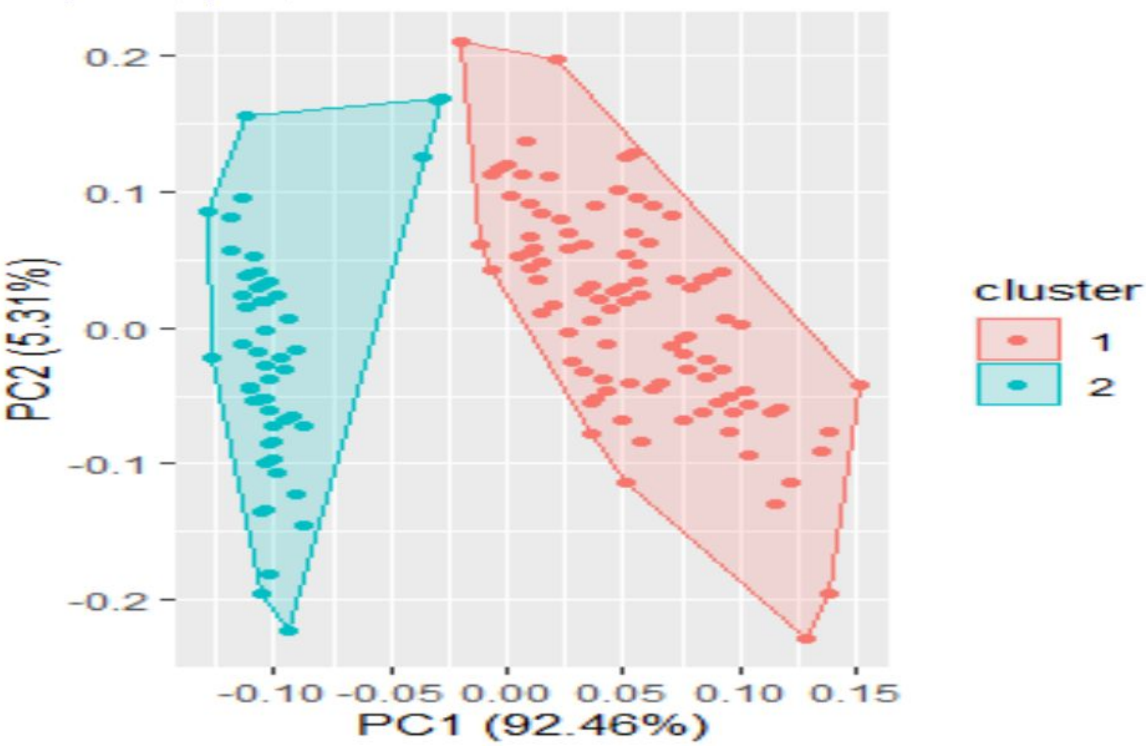
```
install.packages("stats ")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("ggfortify")
library(stats)
library(dplyr)
library(ggplot2)
library(ggfortify)
wssplot<- function(data, nc=15, seed=1234){
 wss<- (nrow(data)-1)*sum(apply(data,2,var))
 for (i in 2:nc){
  set.seed(seed)
  wss[i] <- sum(kmeans(data, centers=i)$withinss)}
 plot(1:nc, wss, type="b", xlab="Number of Clusters",
    ylab="Within groups sum of squares")
 wss
}
View(iris)
mydata=select(iris,c(1,2,3,4))
wssplot(mydata)
KM=kmeans(mydata,2)
autoplot(KM,mydata,frame=TRUE)
KM$centers
```

OUTPUT:

```
> wssplot(mydata)
 [1] 681.37060 152.34795  78.85144  57.26562  46.46117  41.70442  40.66047  39.03110
34.20191  33.40363  32.53526
[12]  30.71360  29.34359  27.01944  19.61325
```




```
> autoplot(KM,mydata,frame=TRUE)
```



```
> KM$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.301031    2.886598     4.958763    1.695876
2     5.005660    3.369811     1.560377    0.290566
```

RESULT:

PROGRAM NO: 06 DATE:	Association Rules using R Tool	PAGE NO:

AIM

ALGORITHM

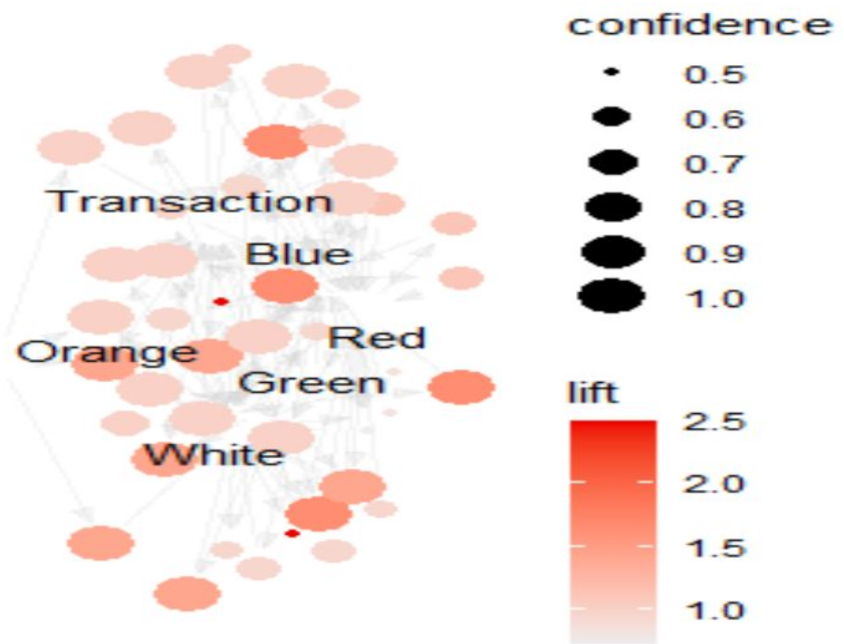
SOURCE CODE:

```
install.packages("arules")
install.packages("arulesViz")
library(arules)
library(arulesViz)
qq=as.matrix(Faceplate)
qq=as(qq,"transactions")
rules=apriori(qq,parameter = list(supp=0.2,conf=0.5,minlen=2))
rules=sort(rules,by="lift")
inspect(rules)
plot(rules, method = "graph", measure = "confidence", shading = "lift")
```

OUTPUT:

```
> inspect(rules)
     lhs                             rhs       support confidence coverage lift      count
[1]  {Red, White}                 => {Green}   0.2     0.5000000  0.4      2.5000000 2
[2]  {Transaction, Red, White}    => {Green}   0.2     0.5000000  0.4      2.5000000 2
[3]  {Green}                      => {Red}     0.2     1.0000000  0.2      1.6666667 2
[4]  {White, Green}               => {Red}     0.2     1.0000000  0.2      1.6666667 2
[5]  {Transaction, Green}         => {Red}     0.2     1.0000000  0.2      1.6666667 2
[6]  {Transaction, White, Green}  => {Red}     0.2     1.0000000  0.2      1.6666667 2
[7]  {Orange}                     => {White}   0.2     1.0000000  0.2      1.4285714 2
[8]  {Green}                      => {White}   0.2     1.0000000  0.2      1.4285714 2
[9]  {Transaction, Orange}        => {White}   0.2     1.0000000  0.2      1.4285714 2
[10] {Red, Green}                 => {White}   0.2     1.0000000  0.2      1.4285714 2
[11] {Transaction, Green}         => {White}   0.2     1.0000000  0.2      1.4285714 2
[12] {Transaction, Red, Green}    => {White}   0.2     1.0000000  0.2      1.4285714 2
```

```
> plot(rules, method = "graph", measure = "confidence", shading = "lift")
```



RESULT:

PROGRAM No.: 01 DATE: 12/01/2024	Data Pre-Processing using R Tool	PAGE No.: 01

AIM:

To write a program for Data preprocessing using R Tool.

ALGORITHM:

step 1 : Start the program

step 2 : Import the dataset from created file, hence import text.

step 3 : Read the dataset using statement dataset = read.csv('data.csv')

step 4 : check if there 'NA' is any column using if else statement.

step 5 : If there is 'NA' then find the mean of whole column and give the average value in the place of 'NA'.

step 6 : Split the dataset into two: training set and test set.

step 7 : Test set is also treated in the same scale.

step 8 ; stop the process.

PROGRAM No.: 02 DATE: 29/01/2024	Data Visualization Technique using R Tool	PAGE No.: 04

AIM:

To write a program for data visualization Technique using R Tools.

ALGORITHM:

step 1 : Start the program

step 2 : Import the dataset and analyze the data. call the barplot data and using barplot call the first column then run the process.

step 3 : use the histogram to define the color frequency.

step 4 : use the boxplot for defining border, color, column etc.

step 5 : Use scatterplot to plot in parts.

step 6 : use heatmap to remove 'na' and replace average air quality value.

step 7 : Stop the process.

PROGRAM No.: 03 DATE: 12/02/2024	Implementing Correlation, linear regression using R Tool	PAGE No.: 11

AIM

To perform the implementing correlation, linear regression using R Tool.

ALGORITHM

step 1 : start the program.

step 2 : check of the dataset is already prevent by using the command dataset using of present import it.

step 3 : Assign the dataset use to dataset we test the correlation between sepal length and sepal width cor-test.

step 4 : The kind all method evaluate the degree of similarity between sepal length.

step 5 : The speaiman method is and to summarize the strength.

step 6 : Install the corrplot package.

step 7 : The corrplot is depicted in circle method

step 8 : stop the process.

PROGRAM No.: 04 DATE: 19/02/2024	Classification using R Tool	PAGE No.: 4

AIM

To write a program to perform a classification using R Tool.

ALGORITHM

step 1: start the program.

step 2: create a dataframe 'df' from the 'infert' dataset to view it.

step 3: Generate a vector 'ind' with random sample value of 1 and 2.

step 4: create a training dataset as 'train' data and a testing dataset as 'test' data.

step 5: Define a formula 'myf' for a condition infert-ctree with predictor variables.

step 6: predict the education levels for the testing data and create config table to evaluate model performance.

step 7: stop the process.

PROGRAM No.: 05 DATE: 23/03/2024	Clustering using R Tool	PAGE No.: 18

AIM

To write a program clustering using R Tool.

ALGORITHM

step 1 : start the program.

step 2 : install the packages stats, dplyr, ggplot2, ggfortify and view it.

step 3 : use the wssplot and select the column 1 to 4.

step 4 : use the Kmeans cluster to classify or group the data.

step 5 : Add the Autoplot to find the center of the cluster.

step 6 : stop the program.

PROGRAM No.: 06 DATE: 04/03/2024	Association Rules using R Tool	PAGE No.: 22

AIM

To write a program to perform Association rules using R Tool.

ALGORITHM

step 1: start the process.

step 2: Install packages 'arules' and 'aruleViz'

step 3: Insert the dataset 'Faceplate' and view it.

step 4: And use the aprior algorithm to classify the data.

step 5: Insert Rules to sort the data and list.

step 6: Draw the plot using 'confidence' and 'lift'.

step 7: stop the process.