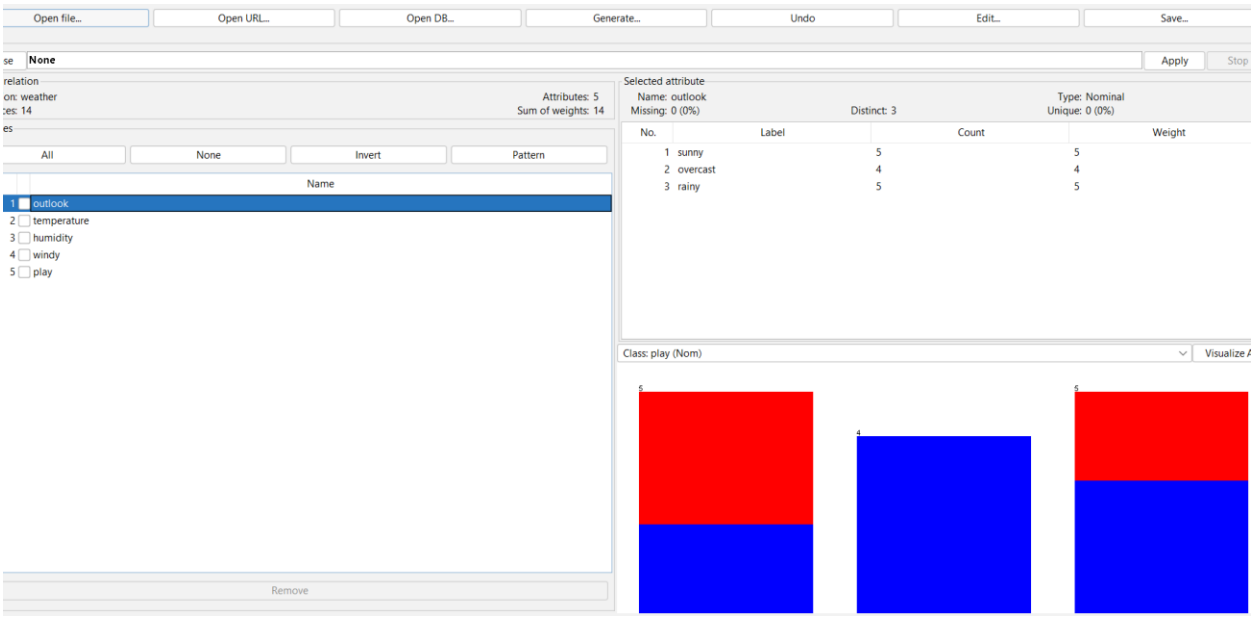
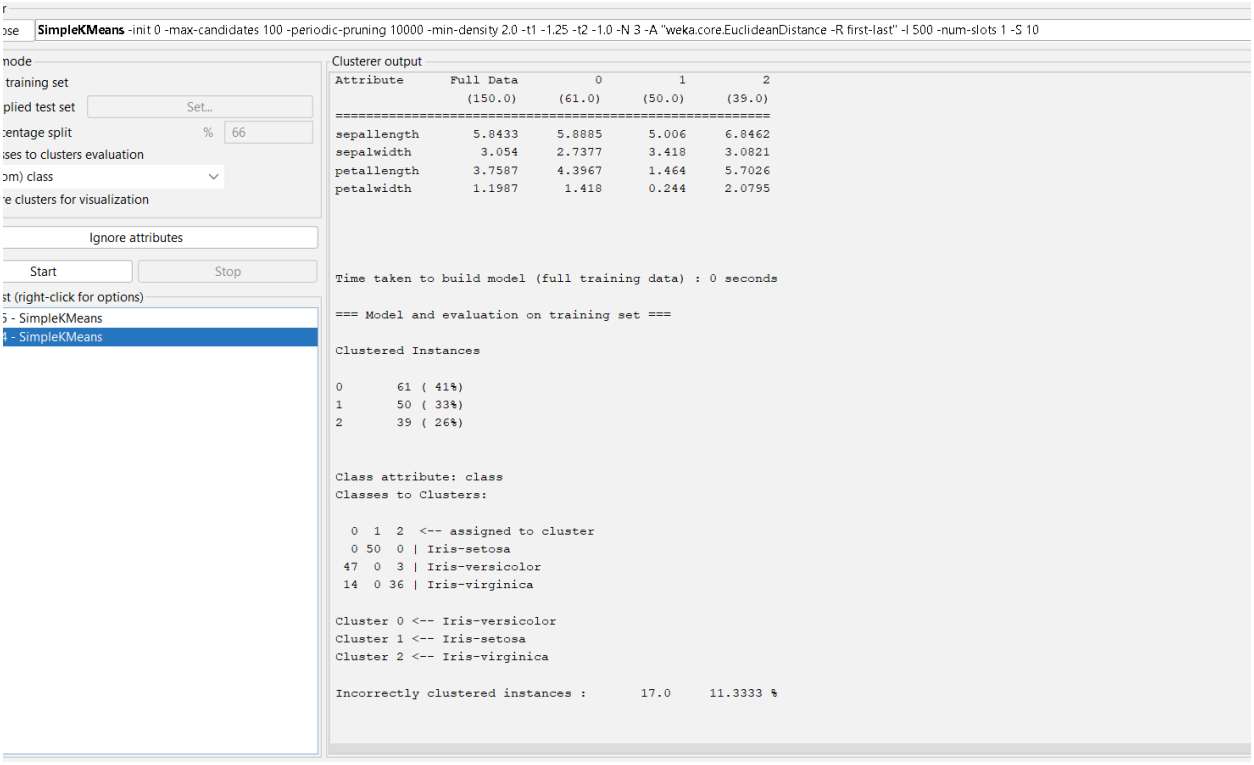
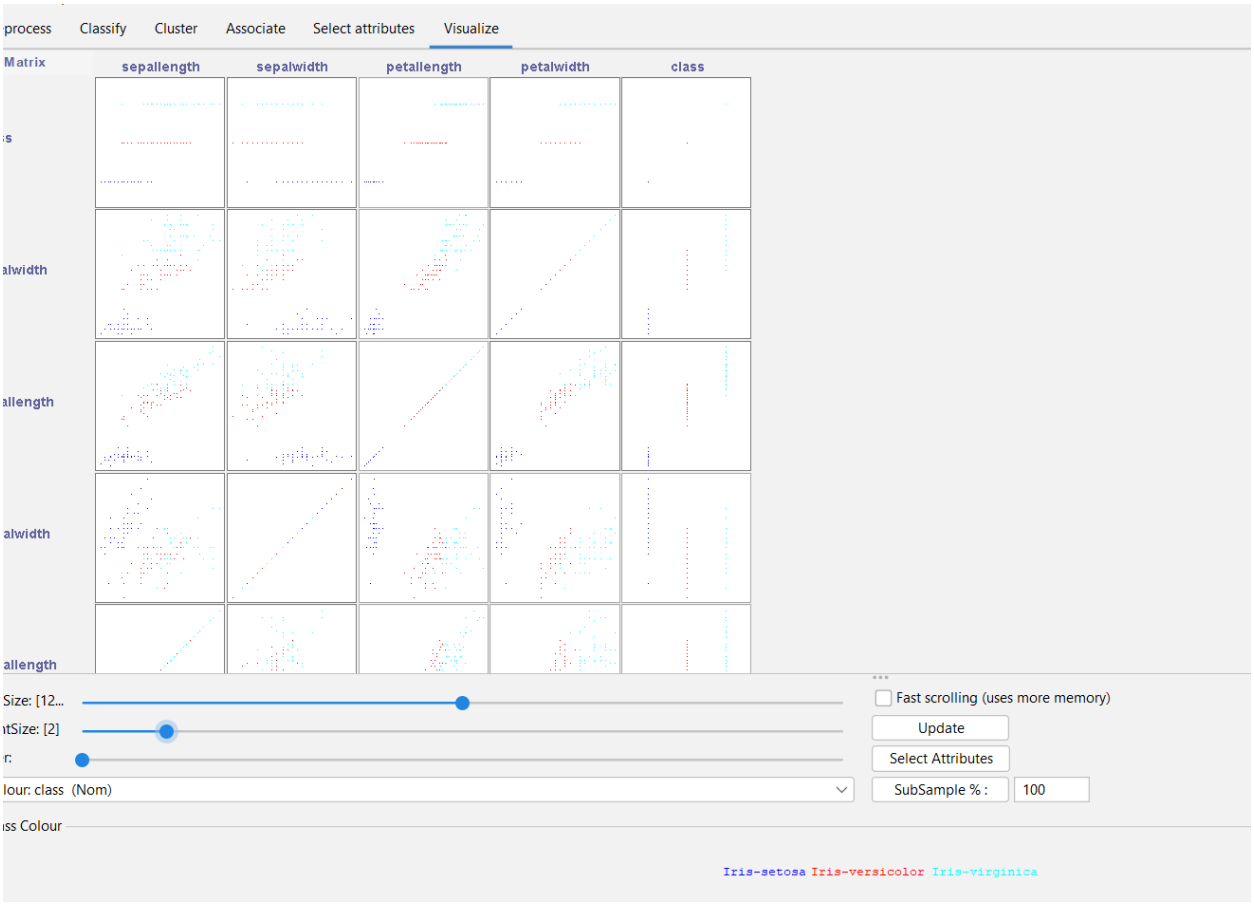
7. PREPROCESSING OF DATA USING WEKA



8. K-MEANS CLUSTERING BY WEKA

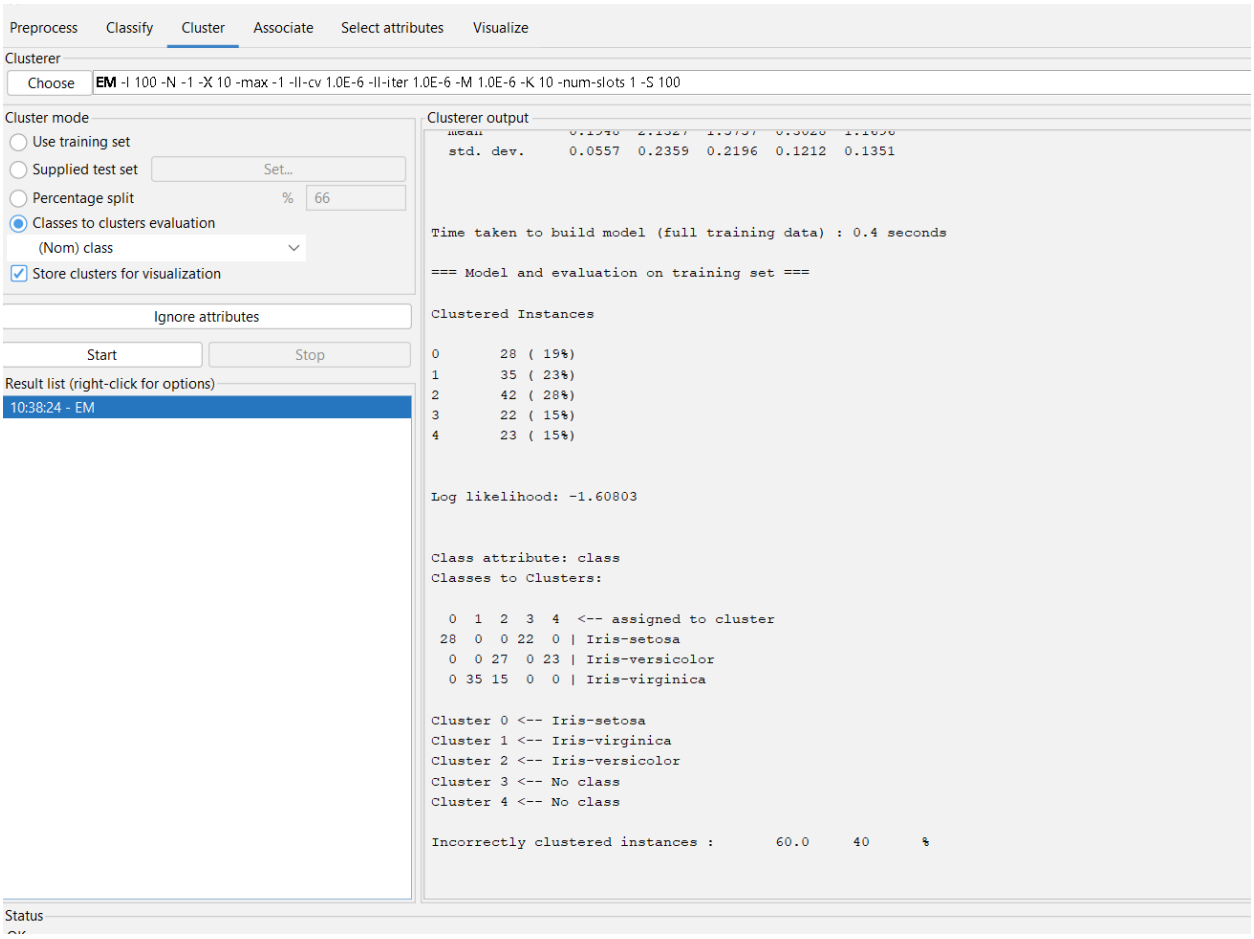


OUTPUT INFORMATION OF SIMPLE K MEANS CLUSTERING –CLUSTERER OUTPUT

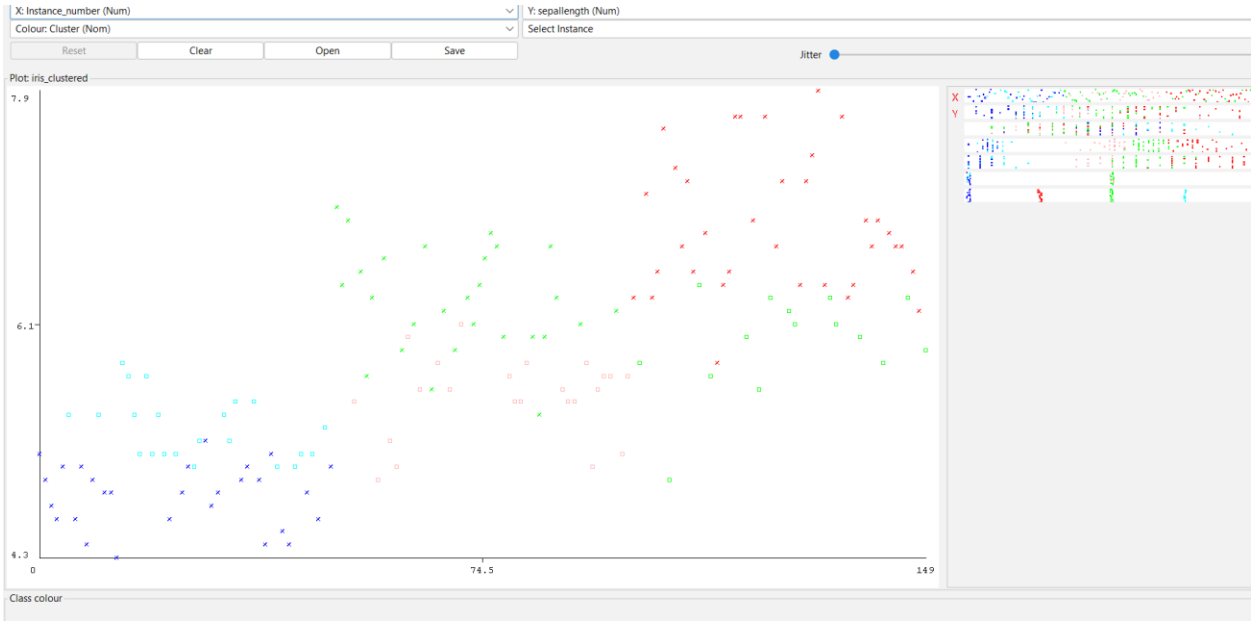


K MEANS CLUSTERING IN WEKAA USING IRIS INFORMATION – PLOTTING ..

9. DATA ANALYSIS BY EXPECTION MAXIMISATION ALGORITHM USING WEKA

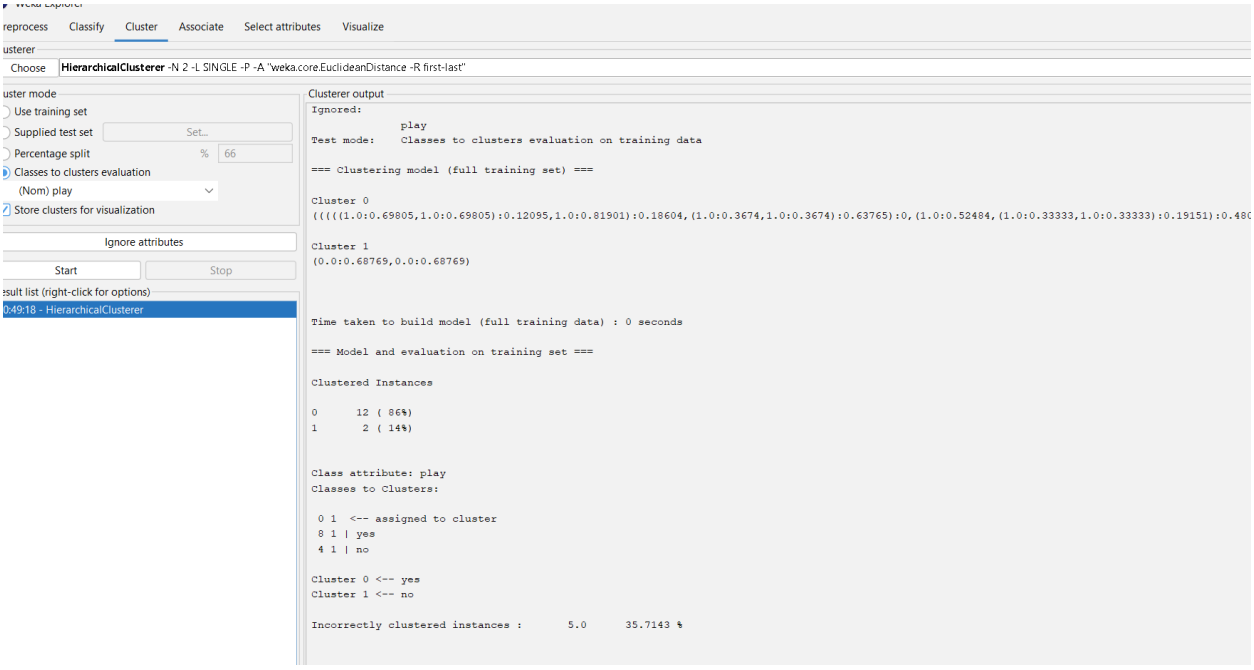


DATA ANALYSIS BY EXPECTATION MAXIMISATION ALGORITHM – OUTPUT INNFO OF IRIS

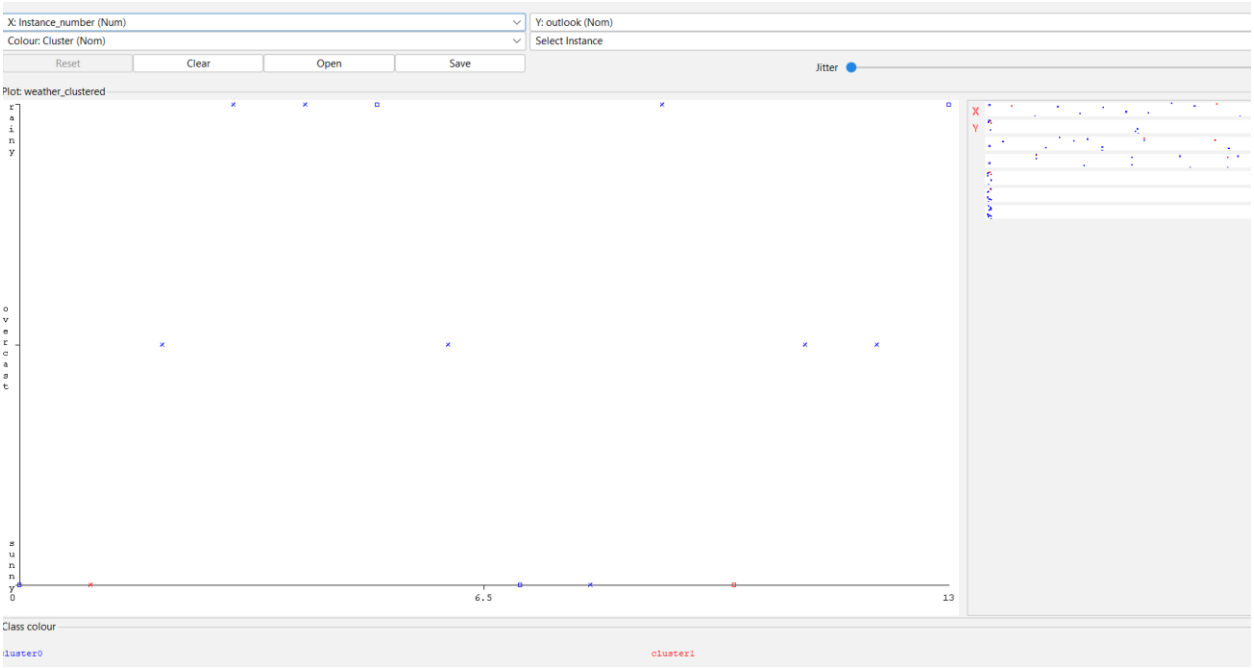


DATA ANALYSIS BY EXPECTATION MAXIMISATION ALGORITHM—PLOT DIAGRAM OF IRIS INFO

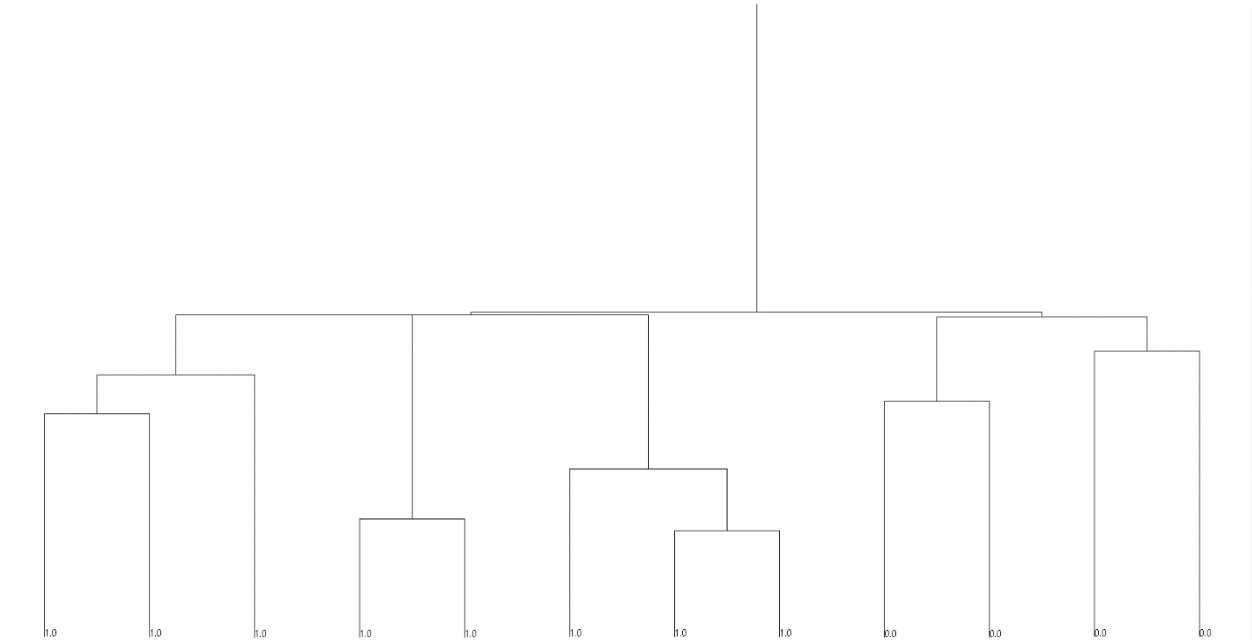
10.DATA ANALYSIS BY HIERARCHICAL CLUSTERING. IN WEKA



OUTPUT INFORMATION

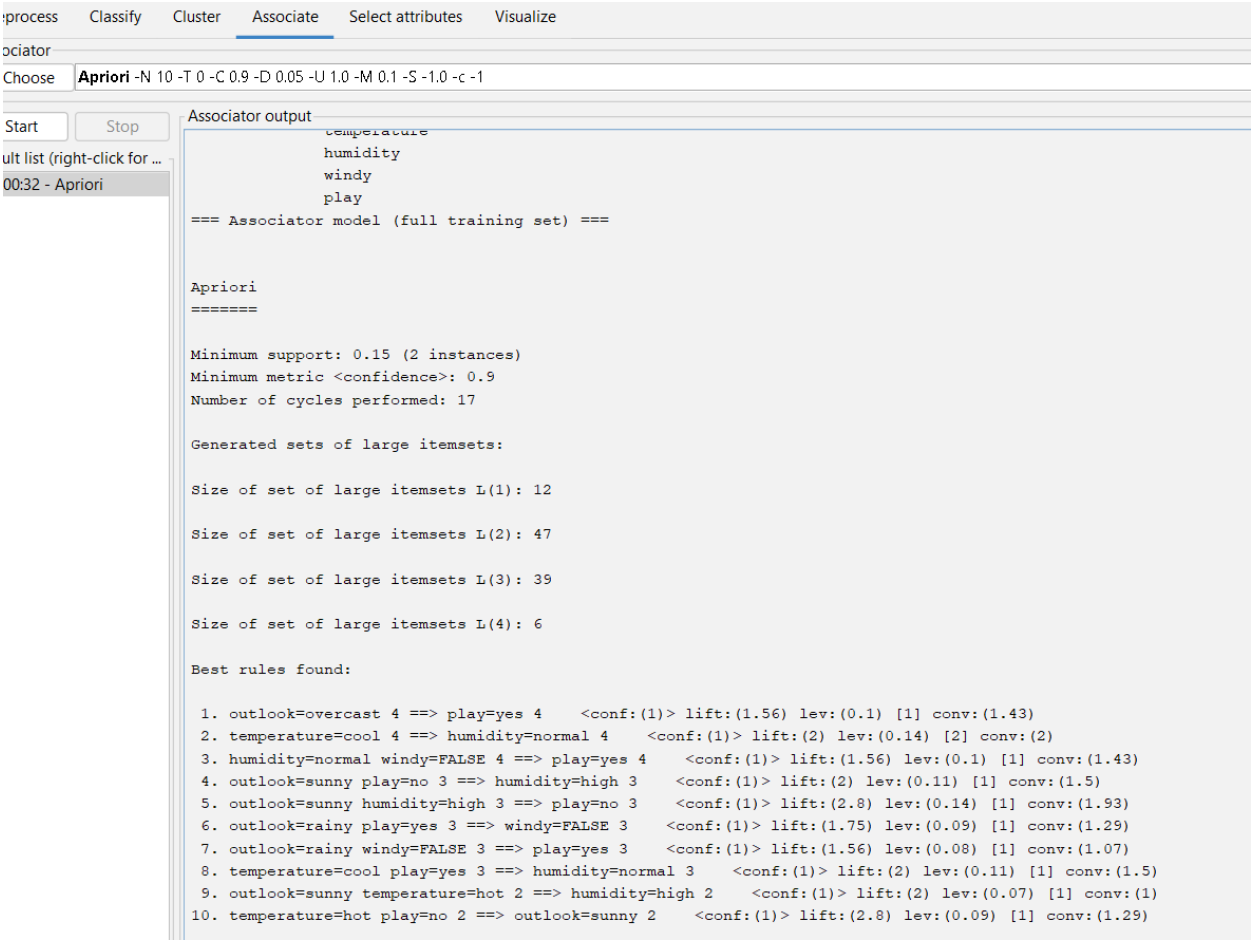


VISUALISING THE PLOT GRAPH OF CLUSTER ASSIGNMENTS



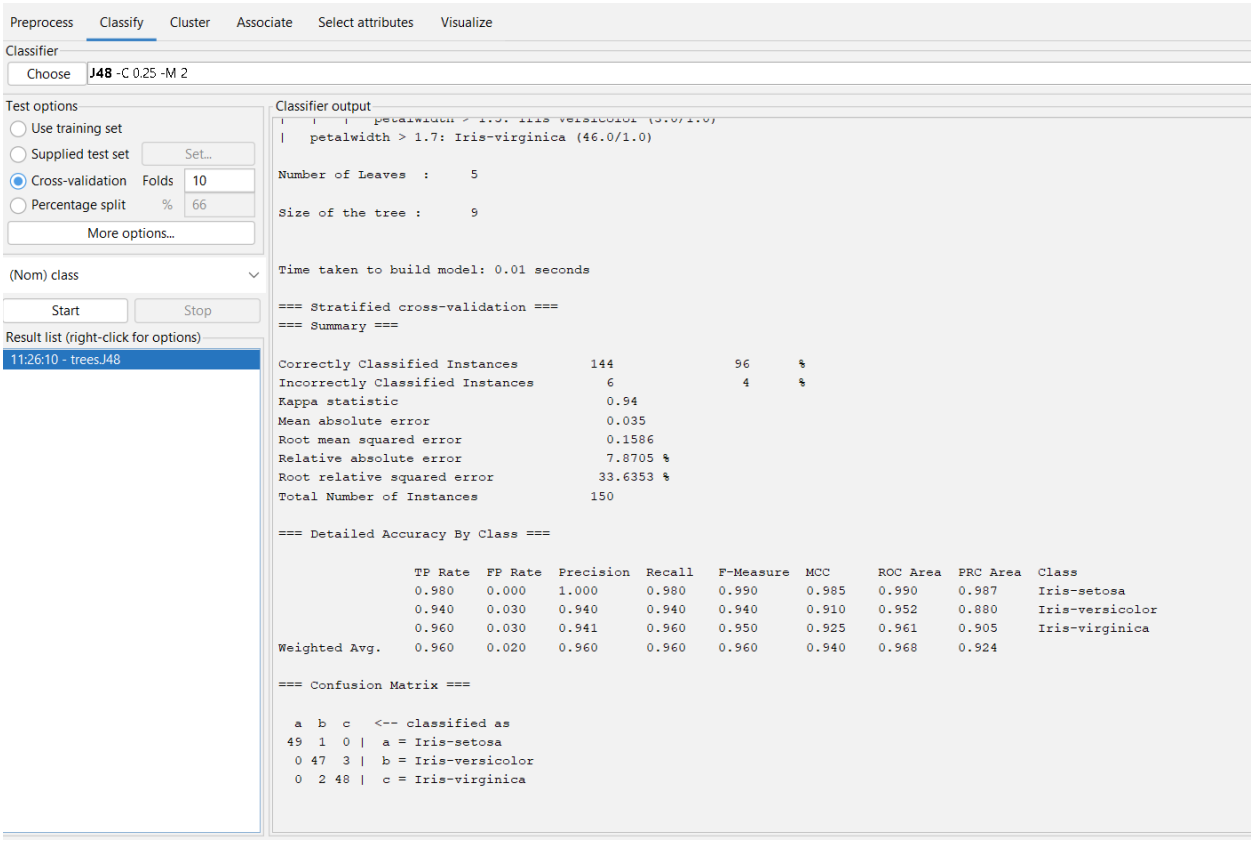
HIERARCHICAL TREE FOR DATA ANALYSIS.

11.KNOWLEDGE MINING FOR ASSOCIATION RULES USING WEKA

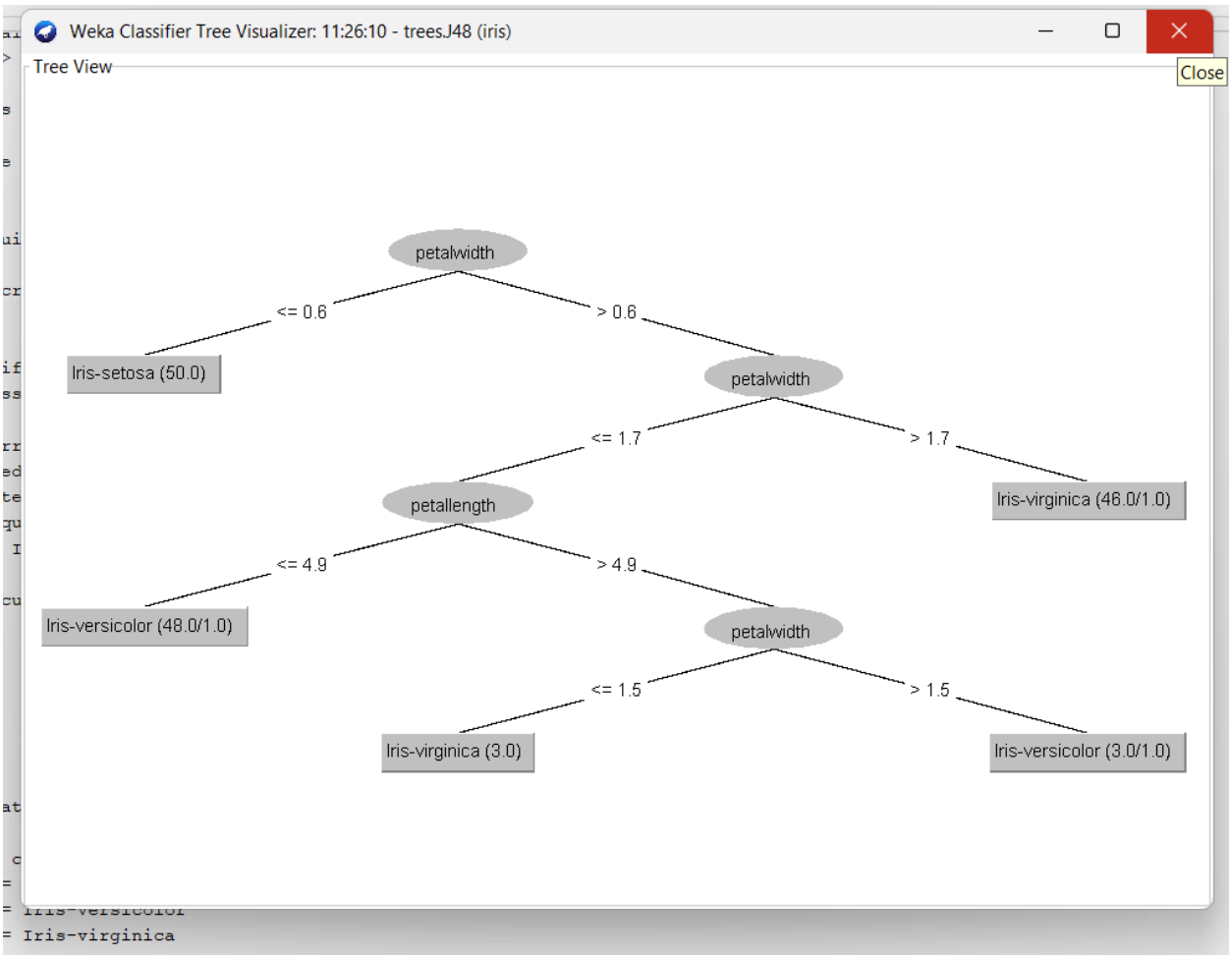


OUTPUT INFORMATION USING WEATHER NOMINAL DATA

15. EVALUATING THE ACCURACY OF THE CLASSIFIERS



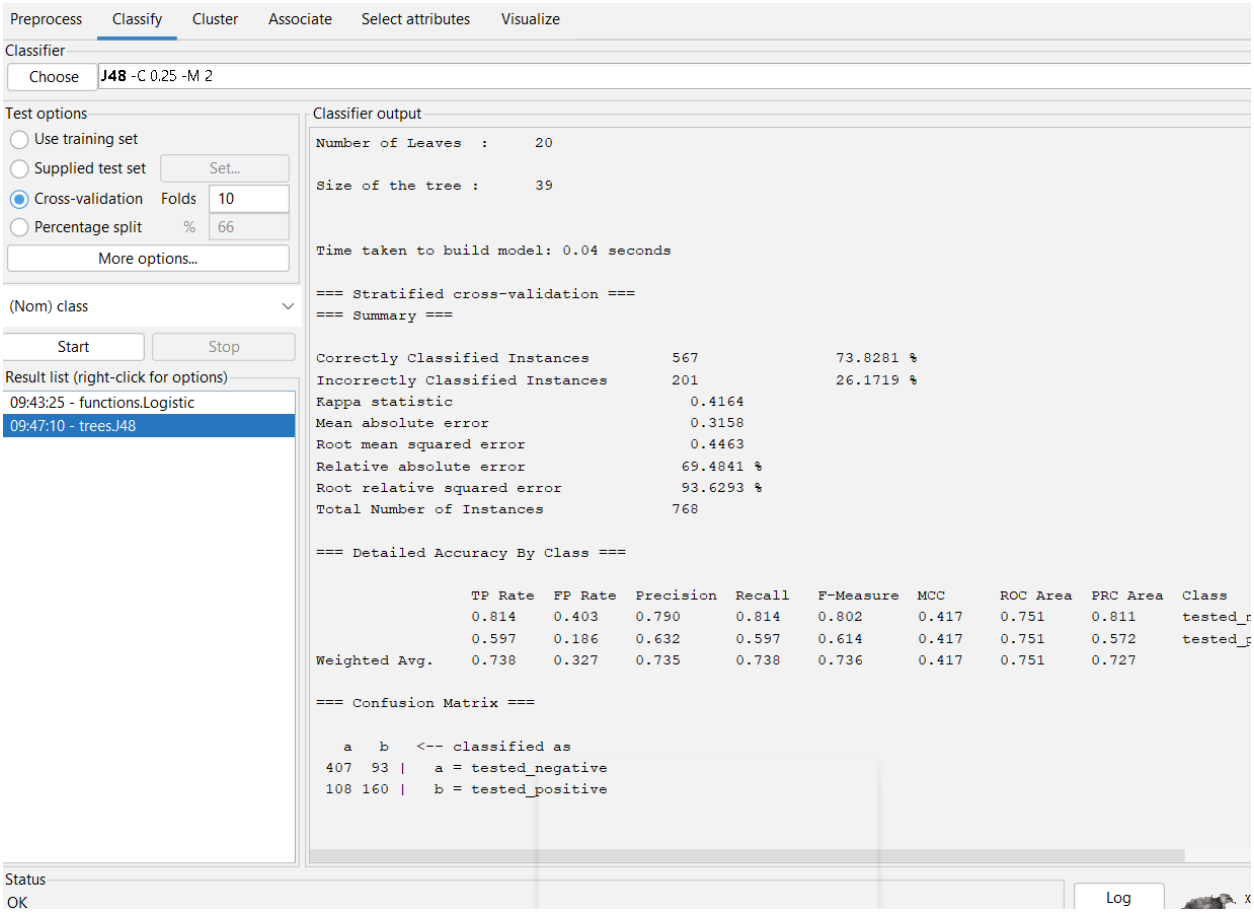
OUTPUT INFORMAATION OF IRIS IN ACCURACY FINDING.



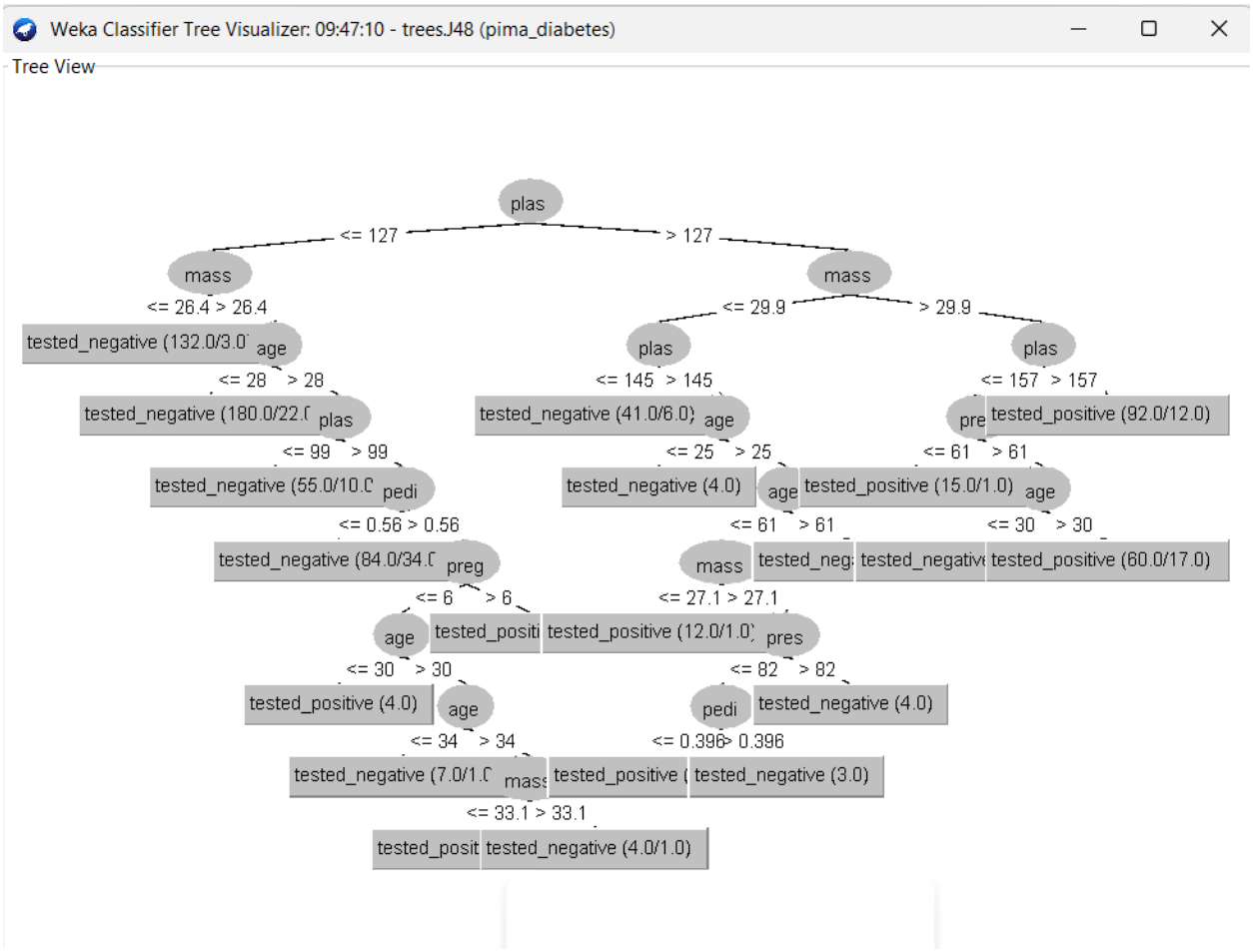
TREE DIAGRAM OF ACCURACY WITH IRIS INFO IN WEKA

ERROR ACCURACY RATE OF IRIS IN WEKA.

1. **CREATE THE ARFF FILE FOR THE DIABETES DATABASE AND PERFORM THE RULE BASED CLASSIFICATION.**

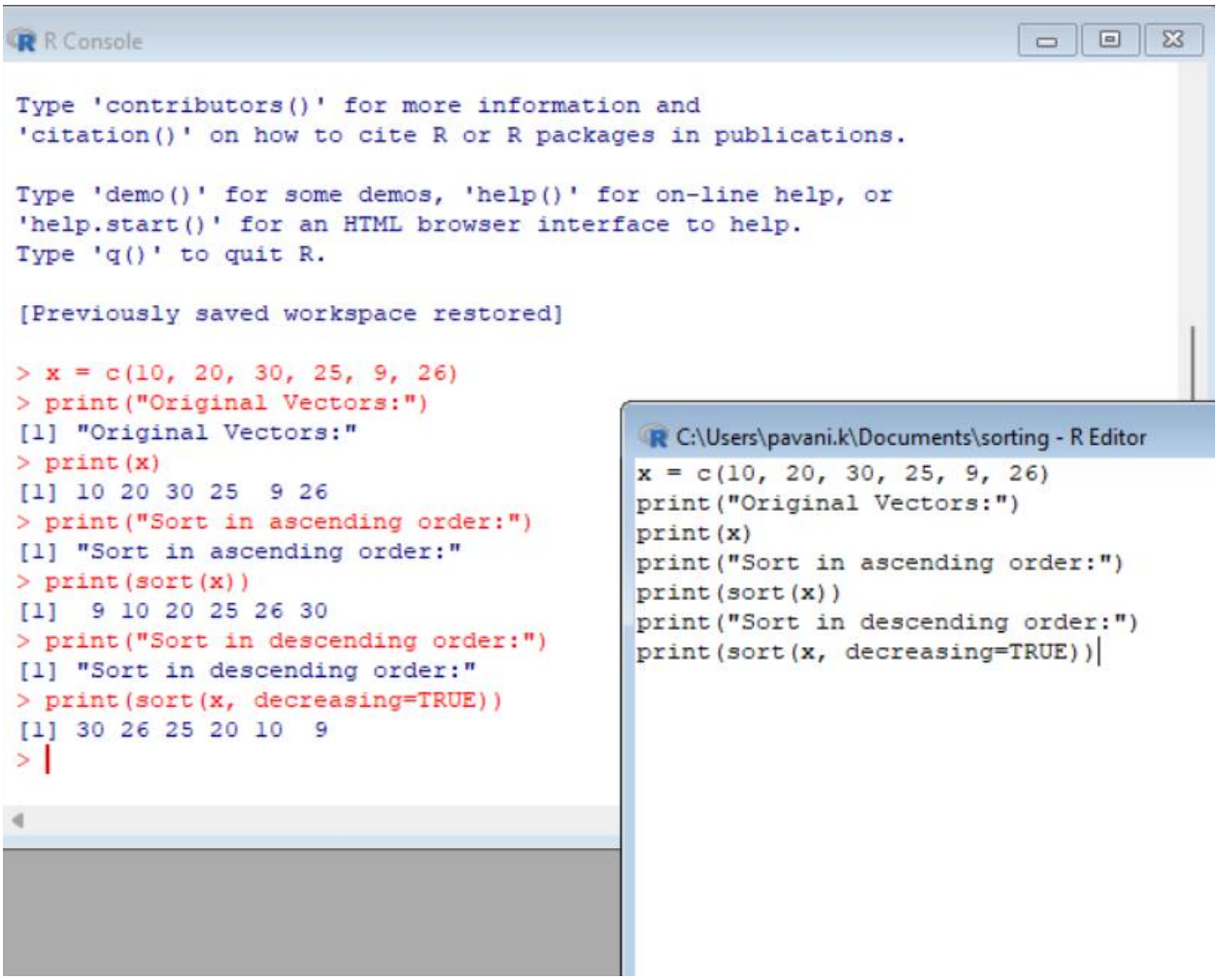


Output info of diabetes by rule based classification.

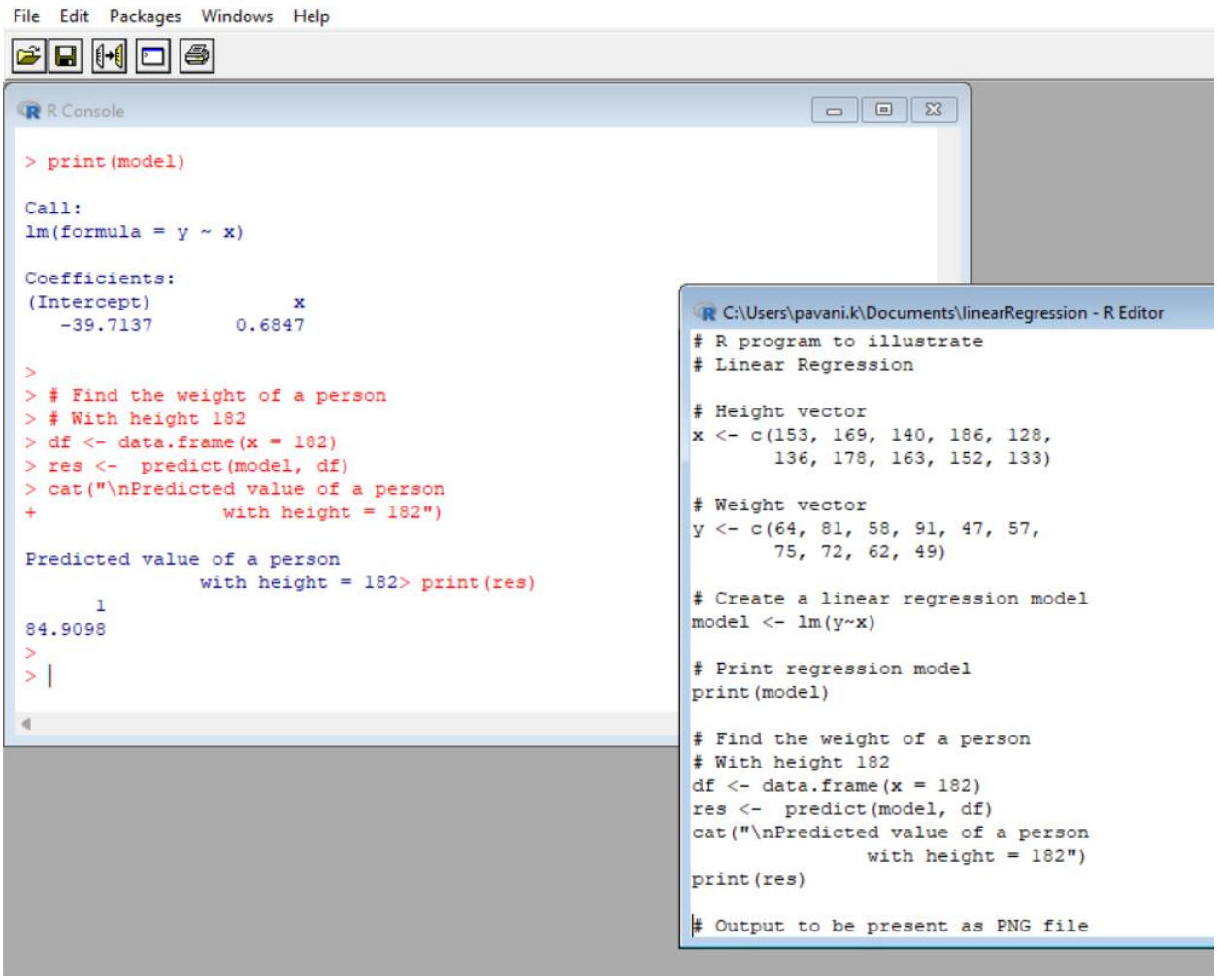


Tree view of diabetes on rule based classification.

*. R PROGRAM FOR SORTING VECTOR



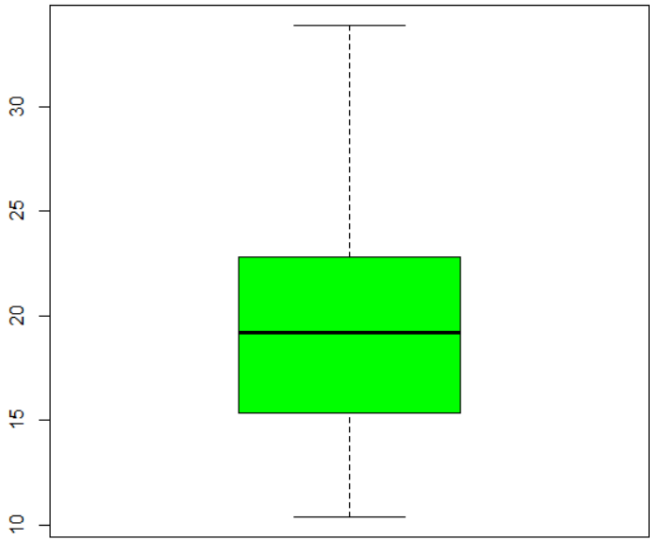
2.R PROGRAM FOR LINEAR REGRESSION.



3.PLOTTING GRAPH

C:\Users\pavani.k\Documents\plotting grapgh - R Editor

```
boxplot(mtcars$mpg, col="green")
```



4.

A) CENTRAL TENDENCY—MEAN:

R Console

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publicati

Type 'demo()' for some demos, 'help()' for on-line help,
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # Defining vector
> x <- c(1, 5, 8, 10)
>
> # Print Harmonic Mean
> print(1 / mean(1 / x))
```

C:\Users\pavani.k\Documents\central tendency - R Editor

```
###mean::
# Defining vector
x <- c(1, 5, 8, 10)

# Print Harmonic Mean
print(1 / mean(1 / x))

#####median::
# Defining vector
x <- c(3, 7, 5, 13, 20, 23, 39,
       23, 40, 23, 14, 12, 56, 23)

# Print Median
median(x)
```

B) CENTRAL TENDENCY –MEDIAN:

```
help.start()' for an HTML browser interface to help.
ype 'q()' to quit R.

Previously saved workspace restored]

 # Defining vector
 x <- c(1, 5, 8, 10)

 # Print Harmonic Mean
 print(1 / mean(1 / x))
1] 2.807018
 # Defining vector
 x <- c(3, 7, 5, 13, 20, 23, 39,
        23, 40, 23, 14, 12, 56, 23)

 # Print Median
 median(x)
1] 21.5
```

C:\Users\pavani.k\Documents\central tendency - R Editor

```
###mean::
# Defining vector
x <- c(1, 5, 8, 10)

# Print Harmonic Mean
print(1 / mean(1 / x))

#####median::
# Defining vector
x <- c(3, 7, 5, 13, 20, 23, 39,
       23, 40, 23, 14, 12, 56, 23)

# Print Median
median(x)

###mode::
# Defining vector
x <- c(3, 7, 5, 13, 20, 23, 39,
       23, 40, 23, 14, 12, 56,
       23, 29, 56, 37, 45, 1, 25, 8)

# Generate frequency table
y <- table(x)

# Print frequency table
print(y)
```

C) CENTRAL TENDENCY—MODE:

```
> y <- table(x)
>
> # Print frequency table
> print(y)
x
 1  3  5  7  8 12 13 14 20 23 25 29 37 39 40 45 56
 1  1  1  1  1  1  1  1  1  4  1  1  1  1  1  1  2
>
```

```
x <- c(3, 7, 5, 13, 20, 23, 39,
       23, 40, 23, 14, 12, 56,

# Print Median
median(x)

###mode::
# Defining vector
x <- c(3, 7, 5, 13, 20, 23, 39,
       23, 40, 23, 14, 12, 56,
       23, 29, 56, 37, 45, 1, 2

# Generate frequency table
y <- table(x)

# Print frequency table
print(y)
```

5.NORMALISATION AND ANALYSIS:

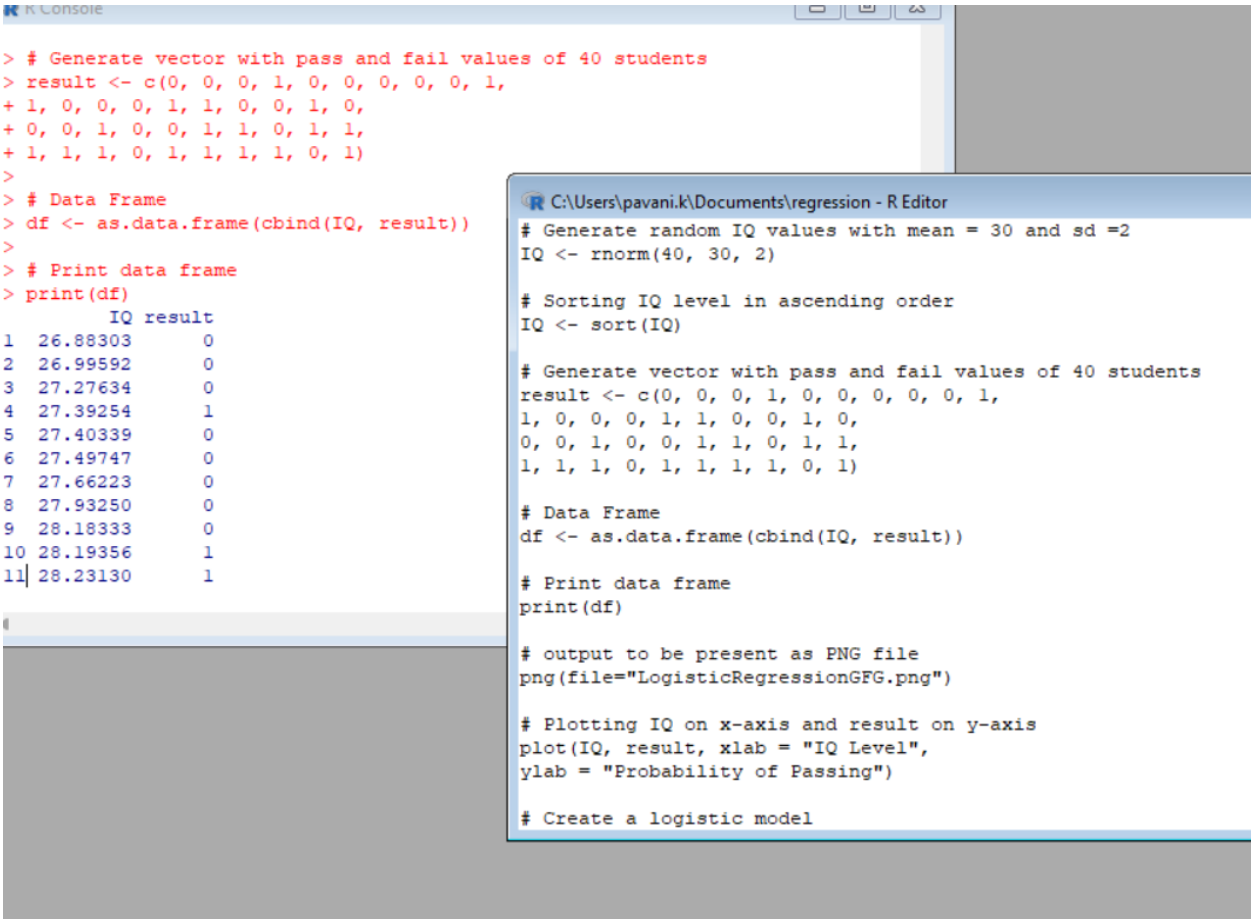
```
#apply Min-Max normalization to first four columns in iris datas
iris_norm <- as.data.frame(lapply(iris[1:4], min_max_norm))

#view first six rows of normalized iris dataset
head(iris_norm)
```

```
Sepal.Length Sepal.Width Petal.Length Petal.Width
   0.22222222   0.6250000   0.06779661  0.04166667
   0.16666667   0.4166667   0.06779661  0.04166667
   0.11111111   0.5000000   0.05084746  0.04166667
   0.08333333   0.4583333   0.08474576  0.04166667
   0.19444444   0.6666667   0.06779661  0.04166667
   0.30555556   0.7916667   0.11864407  0.12500000
```

C:\Users\pavani.k\Documents\normalisation - R Editor

```
#define Min-Max normalization function
min_max_norm <- function(x) {
    (x - min(x)) / (max(x) - min(x))
  }

#apply Min-Max normalization to first four columns in iris datase
iris_norm <- as.data.frame(lapply(iris[1:4], min_max_norm))

#view first six rows of normalized iris dataset
head(iris_norm)
```

6. REGRESSION:

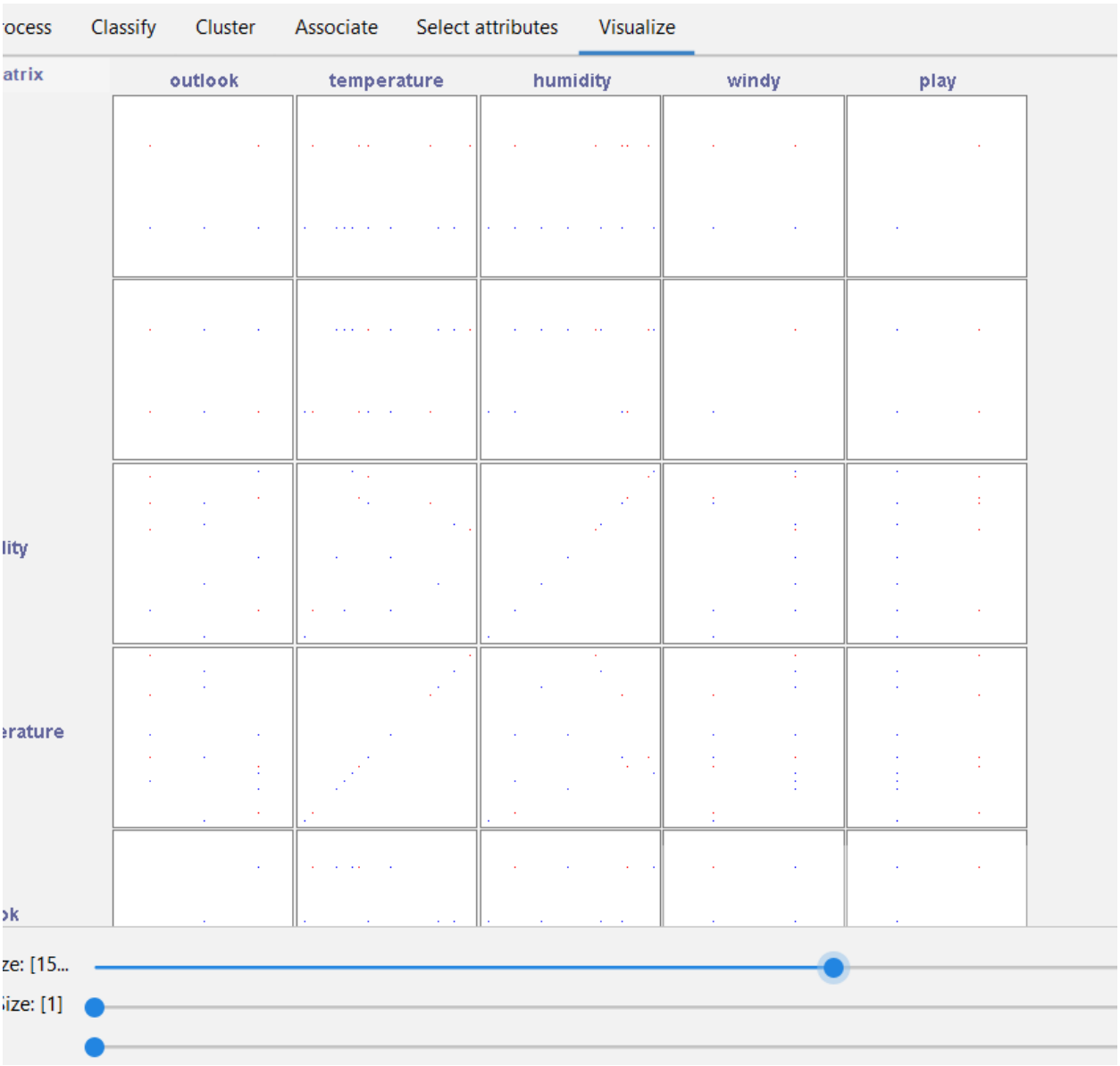


```
> # Generate vector with pass and fail values of 40 students
> result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
+ 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
+ 0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
+ 1, 1, 1, 0, 1, 1, 1, 1, 0, 1)
>
> # Data Frame
> df <- as.data.frame(cbind(IQ, result))
>
> # Print data frame
> print(df)
         IQ result
1  26.88303      0
2  26.99592      0
3  27.27634      0
4  27.39254      1
5  27.40339      0
6  27.49747      0
7  27.66223      0
8  27.93250      0
9  28.18333      0
10 28.19356      1
11 28.23130      1
```

C:\Users\pavani.k\Documents\regression - R Editor

```
# Generate random IQ values with mean = 30 and sd =2
IQ <- rnorm(40, 30, 2)

# Sorting IQ level in ascending order
IQ <- sort(IQ)

# Generate vector with pass and fail values of 40 students
result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
1, 1, 1, 0, 1, 1, 1, 1, 0, 1)

# Data Frame
df <- as.data.frame(cbind(IQ, result))

# Print data frame
print(df)

# output to be present as PNG file
png(file="LogisticRegressionGFG.png")

# Plotting IQ on x-axis and result on y-axis
plot(IQ, result, xlab = "IQ Level",
ylab = "Probability of Passing")

# Create a logistic model
```

12.FP GROWTH



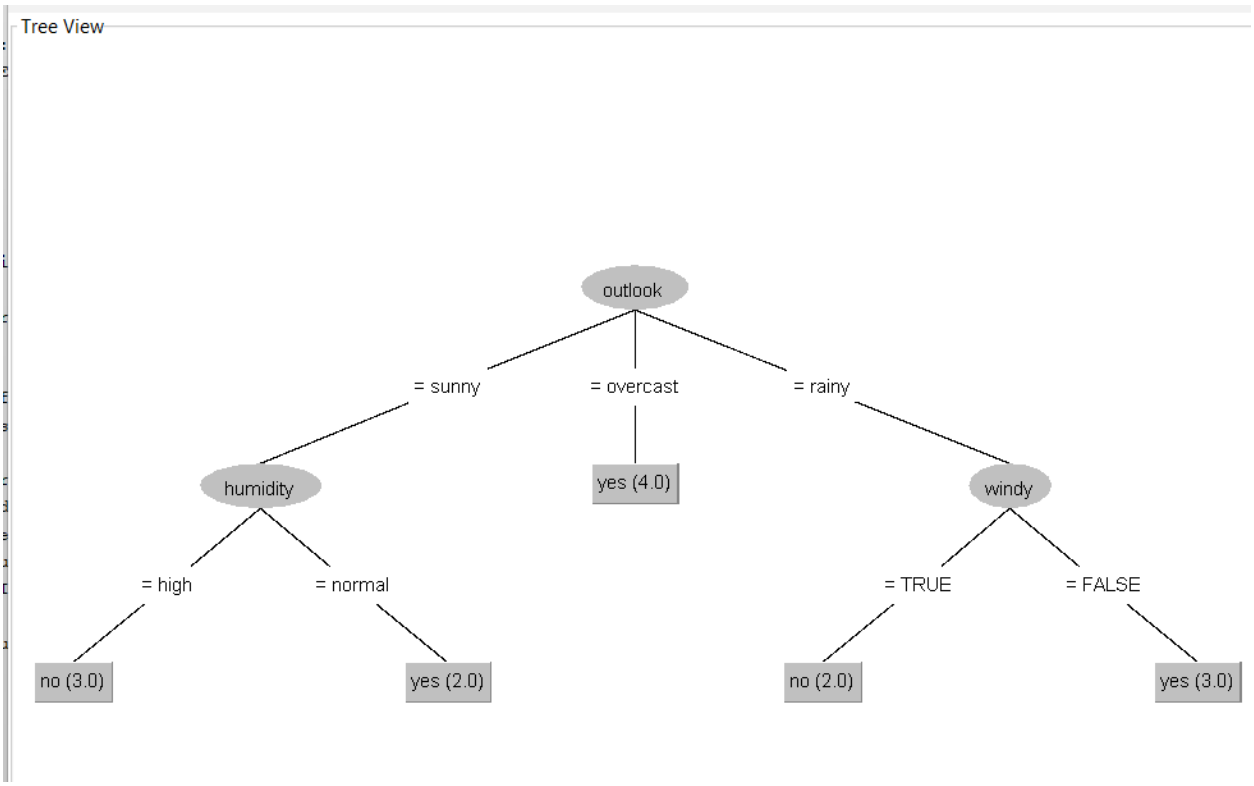
FP GROWTH VISUALISATION

th -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1

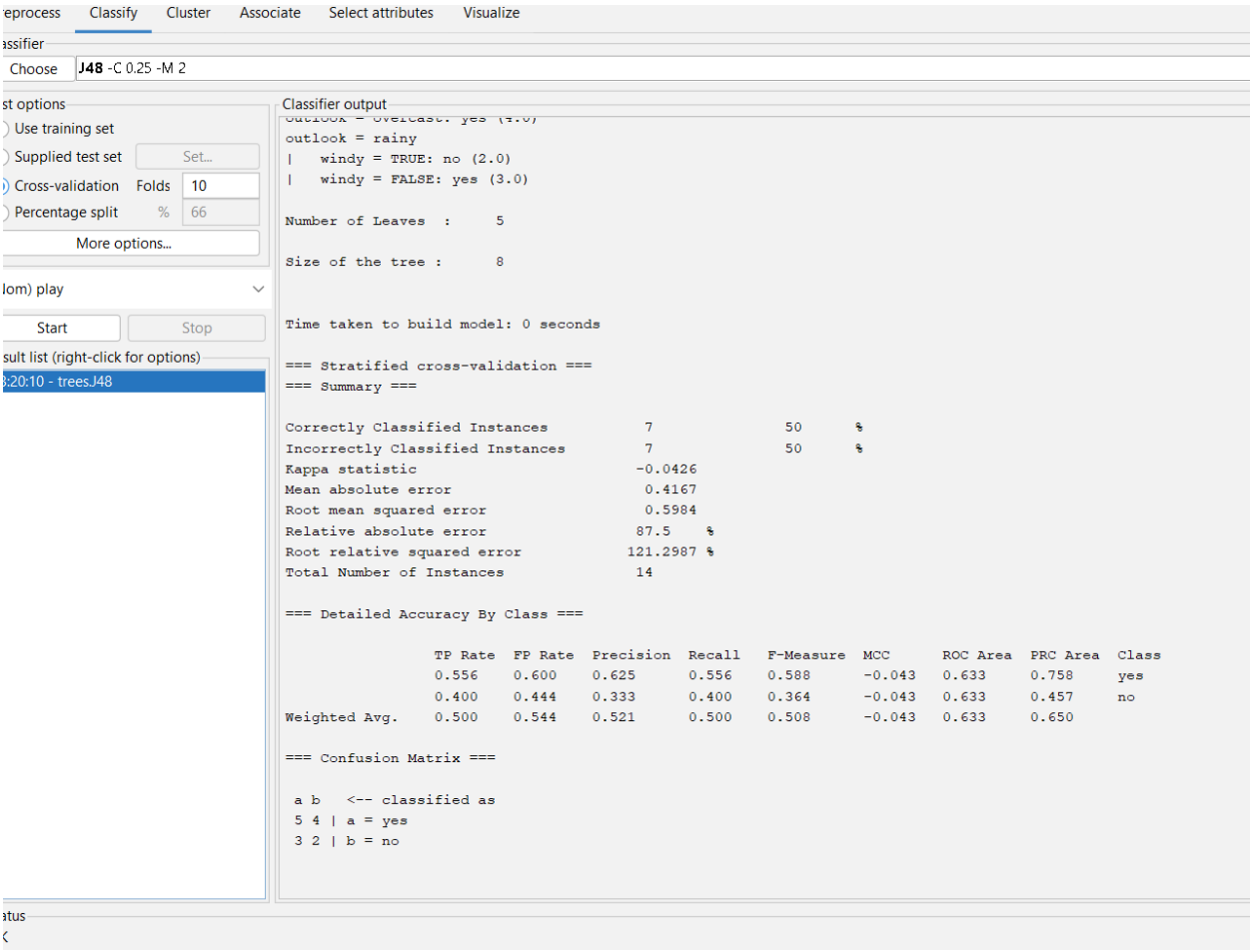
Associator output

```
               temperature
               humidity
               windy
               play
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. outlook=sunny play=no 3 ==> humidity=high 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 5. outlook=sunny humidity=high 3 ==> play=no 3    <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
 7. outlook=rainy windy=FALSE 3 ==> play=yes 3    <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
 8. temperature=cool play=yes 3 ==> humidity=normal 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 9. outlook=sunny temperature=hot 2 ==> humidity=high 2    <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2    <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
```

13.DECISION TREE

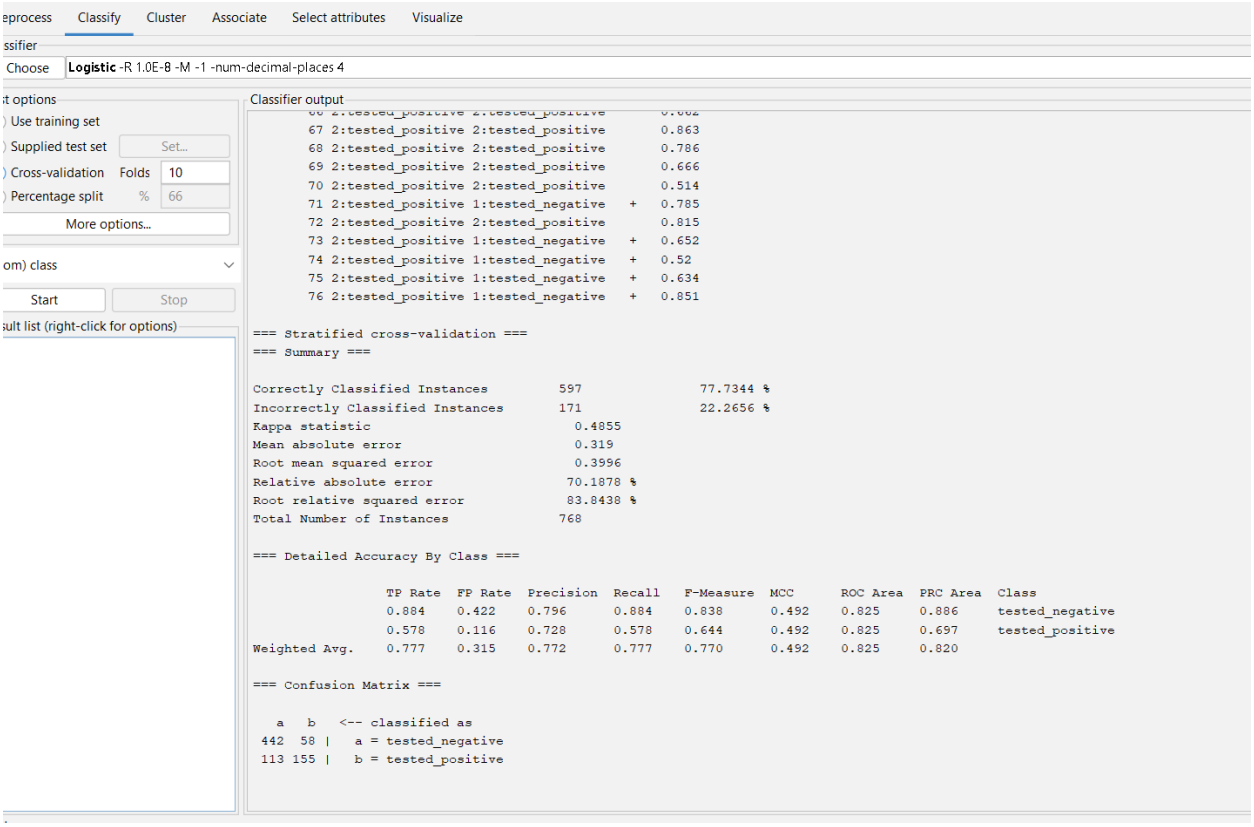


TREE VISUALISATION


eprocess
Classify
Cluster
Associate
Select attributes
Visualize
assifier
Choose
J48 -C 0.25 -M 2
st options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
lom) play
Start
Stop
sult list (right-click for options)
:20:10 - trees.J48
Classifier output
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)
Number of Leaves : 5
Size of the tree : 8
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 7 50 %
Incorrectly Classified Instances 7 50 %
Kappa statistic -0.0426
Mean absolute error 0.4167
Root mean squared error 0.5984
Relative absolute error 87.5 %
Root relative squared error 121.2987 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.556 0.600 0.625 0.556 0.588 -0.043 0.633 0.758 yes
0.400 0.444 0.333 0.400 0.364 -0.043 0.633 0.457 no
Weighted Avg. 0.500 0.544 0.521 0.500 0.508 -0.043 0.633 0.650
=== Confusion Matrix ===
a b <-- classified as
5 4 | a = yes
3 2 | b = no
atus

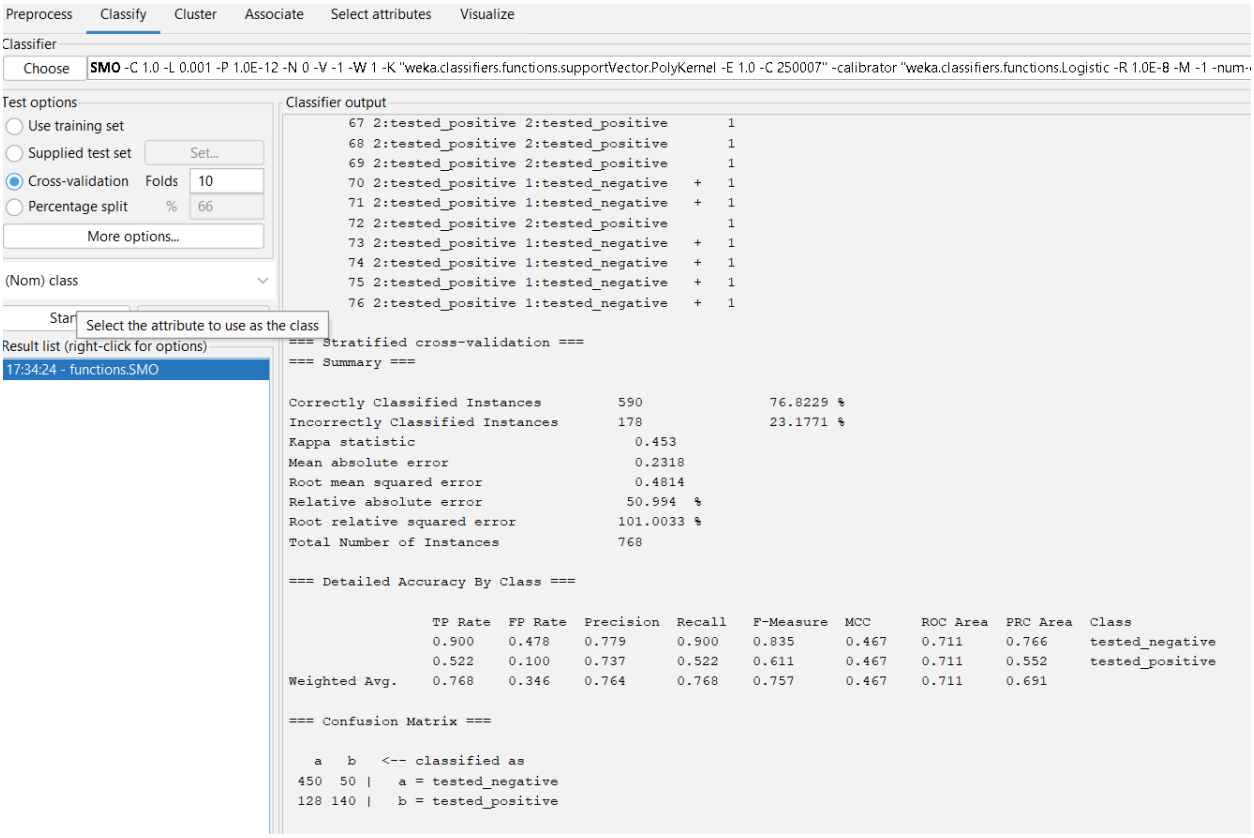
OUTPUT INFO.

16. CREATE THE ARFF FILE FOR THE DIABETES DATABASE AND PERFORM SVM BASED CLASSIFICATION



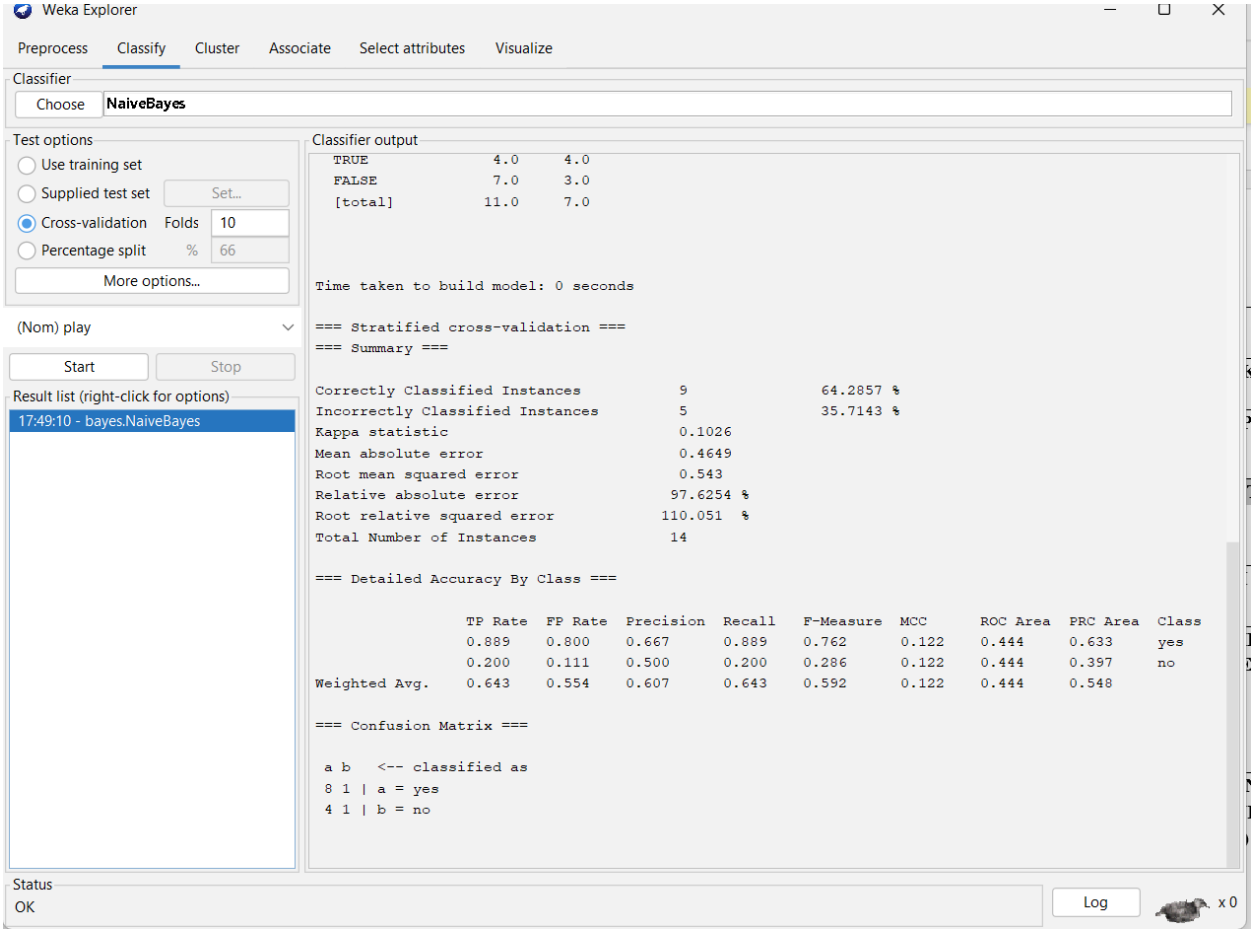
Output information of data diabetes with logistics.

14. **PREDICTION OF CATEGORICAL DATA USING SMO ALGORITHM USING WEKA.**



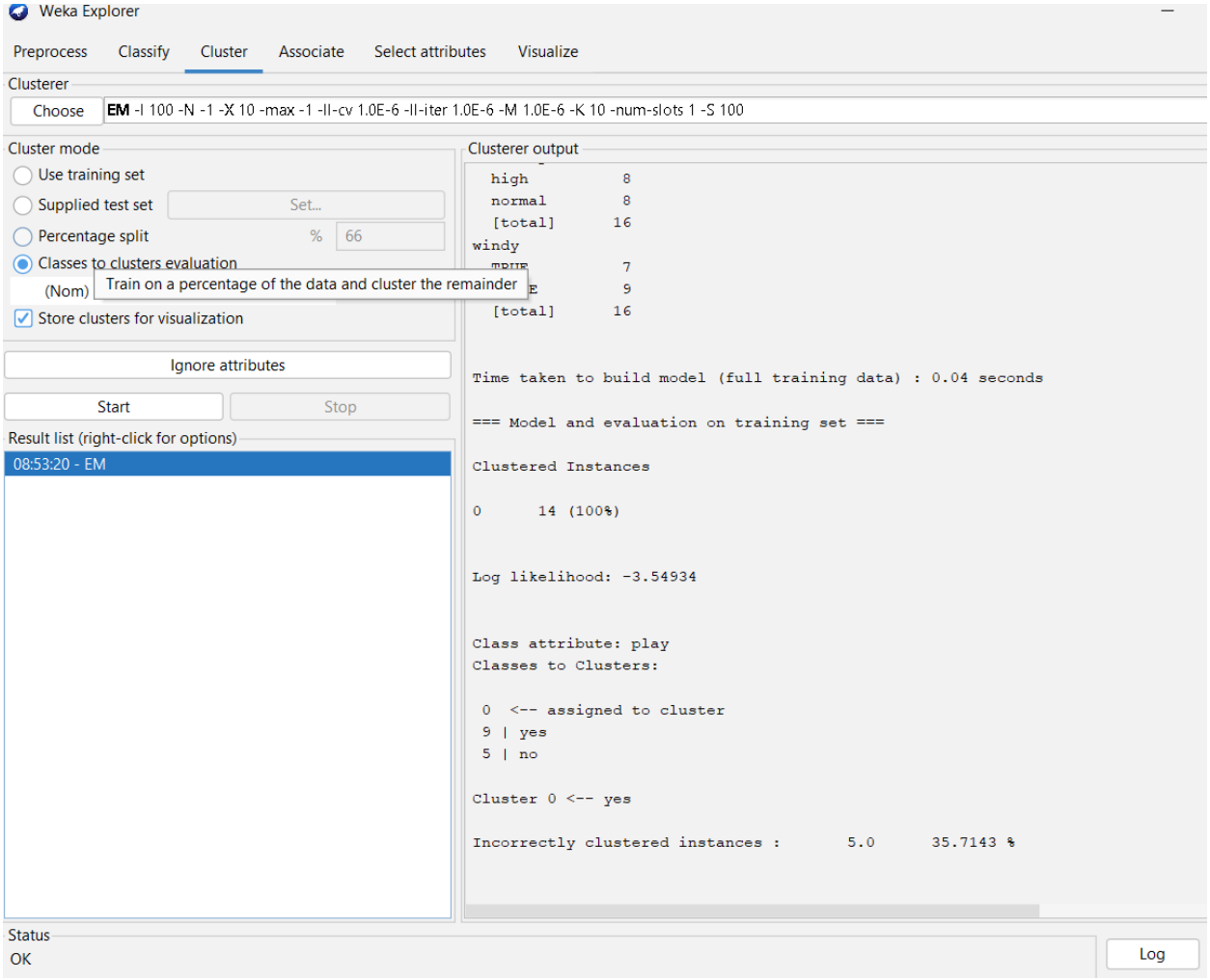
The output information of diabetes data using smo..

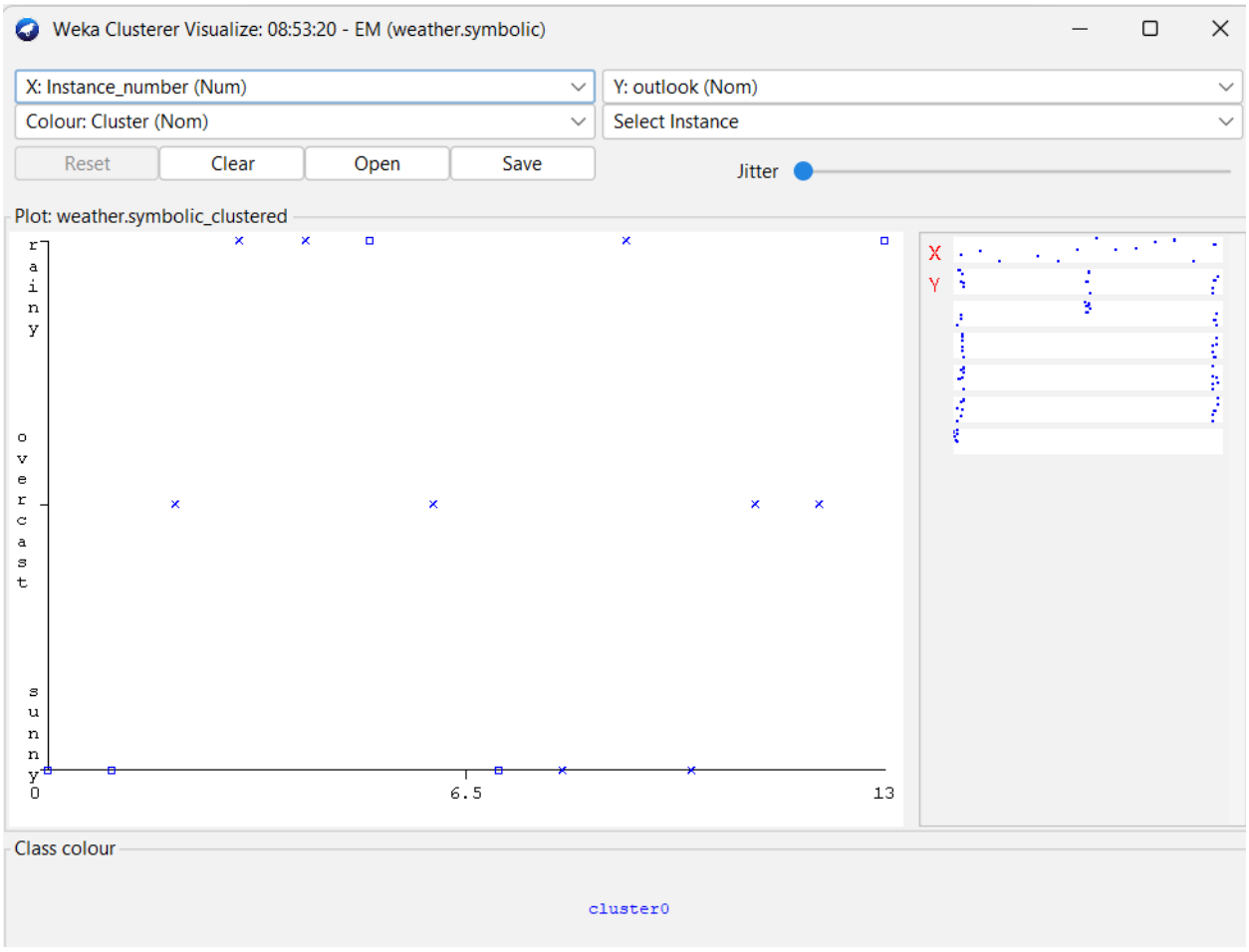
17. **PREDICTION OF CATEGORICAL DATA USING BAYESIAN ALGORITHM USING WEKA**



OUTPUT INFORMATION WEATHER NUMERIC BASED ON NAVIES BAYES RULE..

18. **DATA ANALYSIS BY DENSITY BASED CLUSTERING ALGORITHM USING WEKA.**





Output information...

19. **CREATE A BOXPLOT GRAPH FOR THE RELATION BETWEEN "MPG"(MILES PER GALLOON) AND "CYL"(NUMBER OF CYLINDERS) FOR THE DATASET "MTCARS" AVAILABLE IN R ENVIRONMENT**

Untitled - R Editor

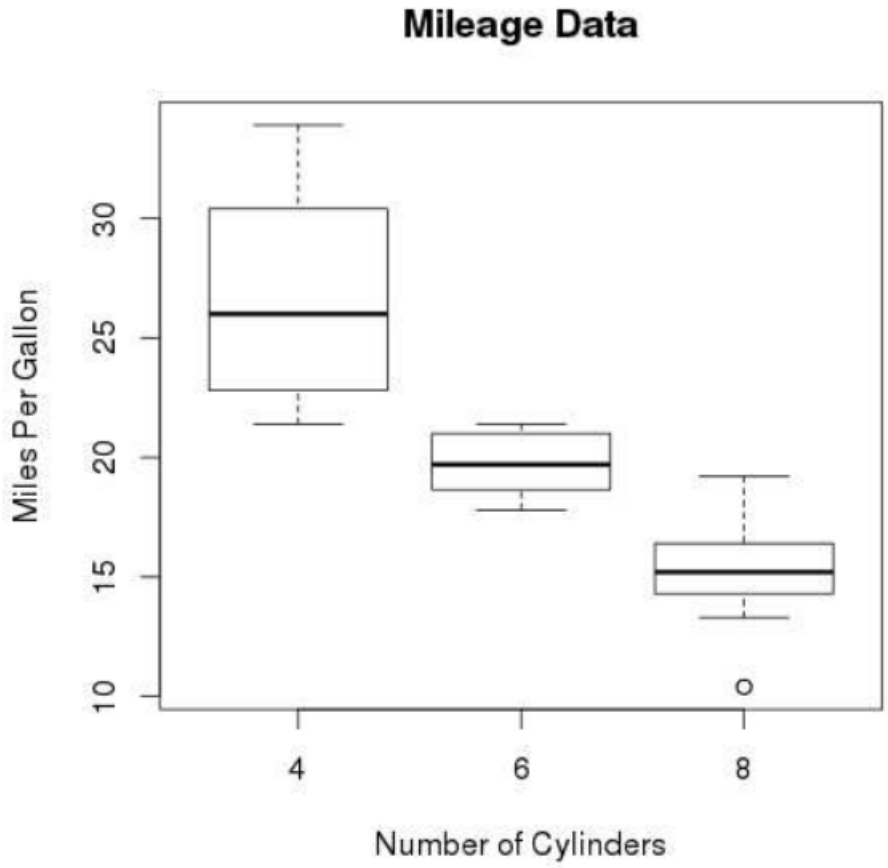
```
############code for creating boxplots#####
# Give the chart file a name.
png(file = "boxplot.png")
# Plot the chart.
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",
   ylab = "Miles Per Gallon", main = "Mileage Data")
# Save the file.
dev.off()
```

Code for creating boxplots in r programing.

R Console

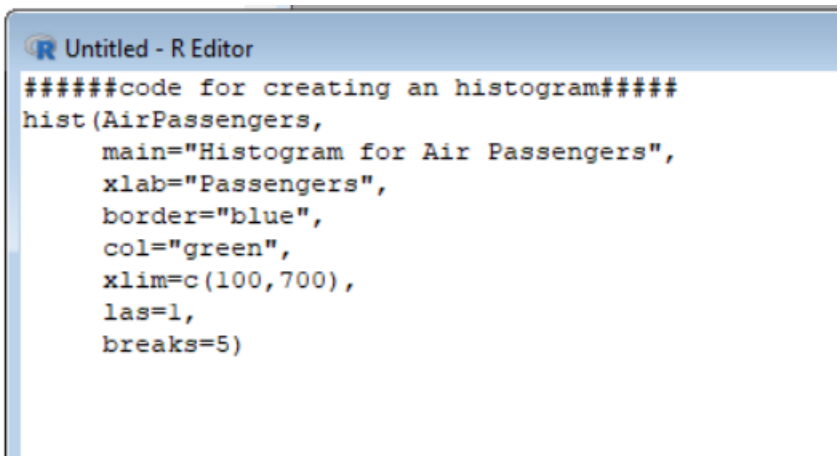
```
  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # Give the chart file a name.
> png(file = "boxplot.png")
> # Plot the chart.
> boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",
+    ylab = "Miles Per Gallon", main = "Mileage Data")
> # Save the file.
> dev.off()
null device
```

Output result of created boxplot..

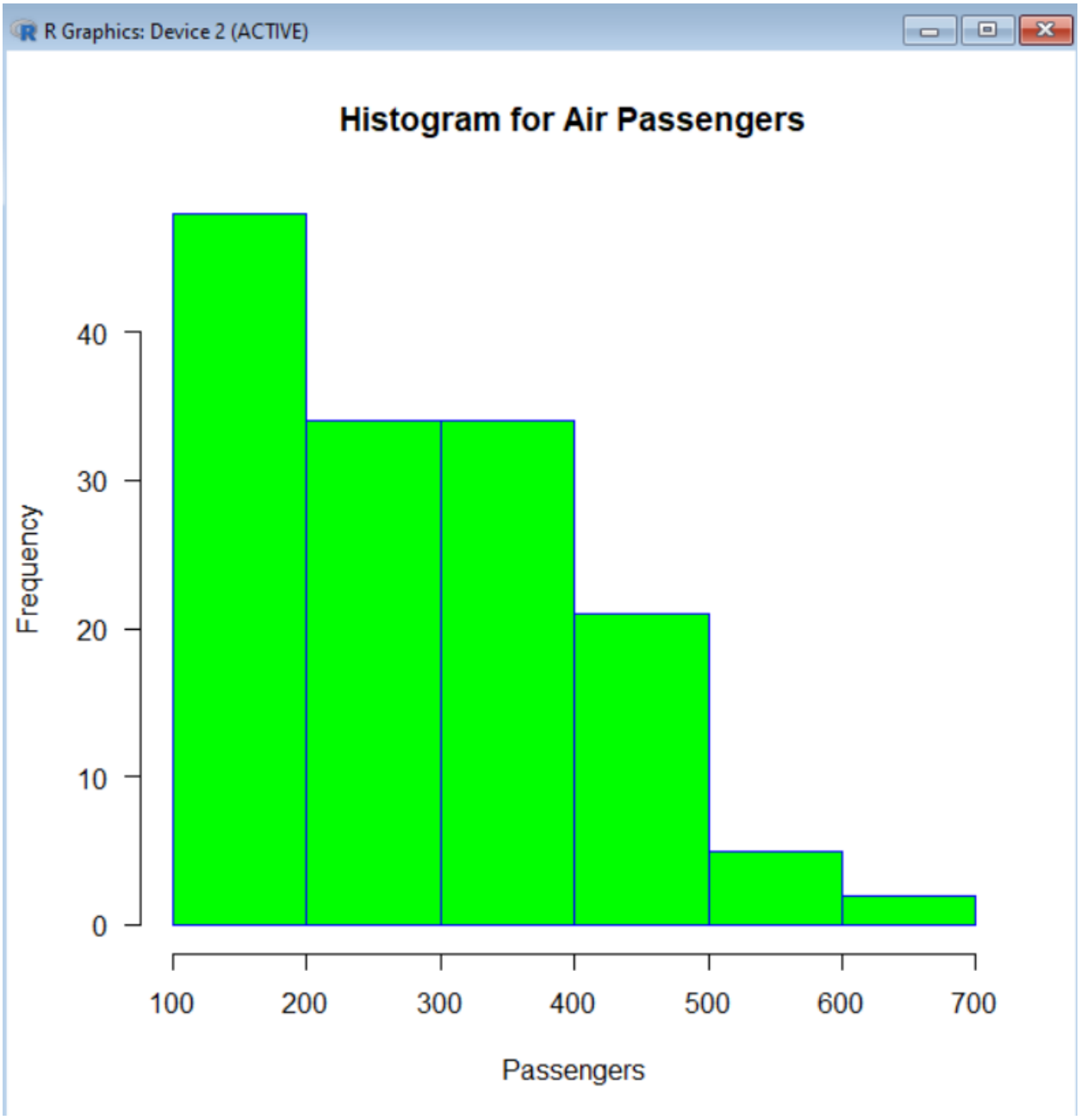


Output graph of boxplot..

22. USING R PROGRAM MAKE A HISTOGRAM FOR THE "AIRPASSENGERS "DATASET, START AT 100 ON THE X-AXIS, AND FROM VALUES 200 TO 700, MAKE THE BINS 150 WIDE

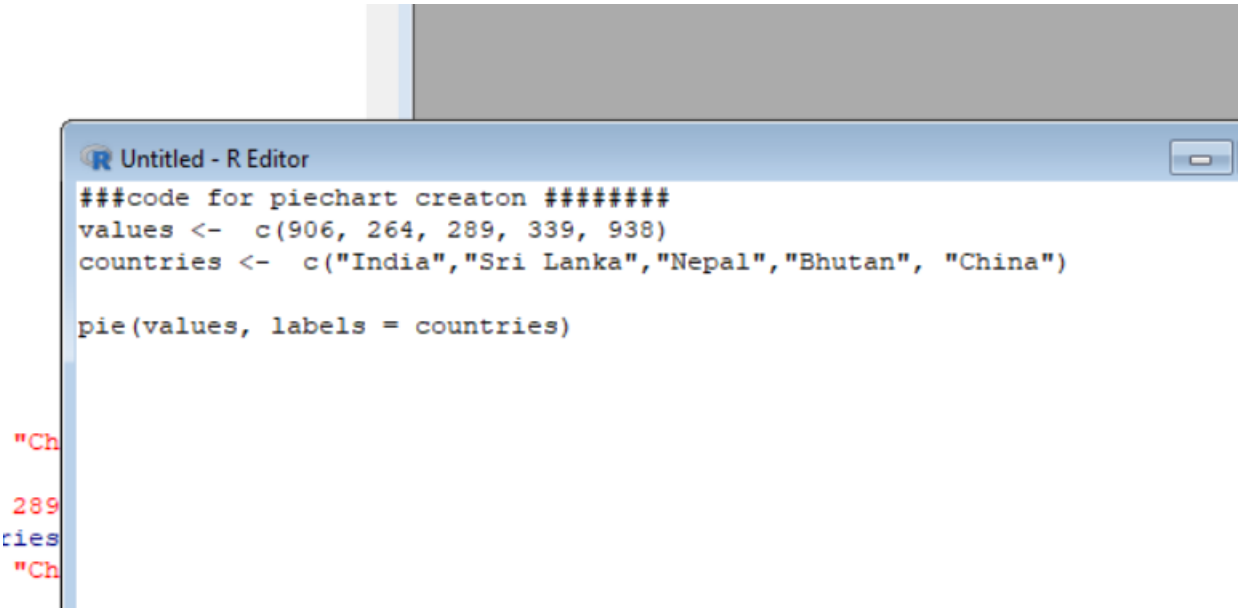


Code for creating an histogram..

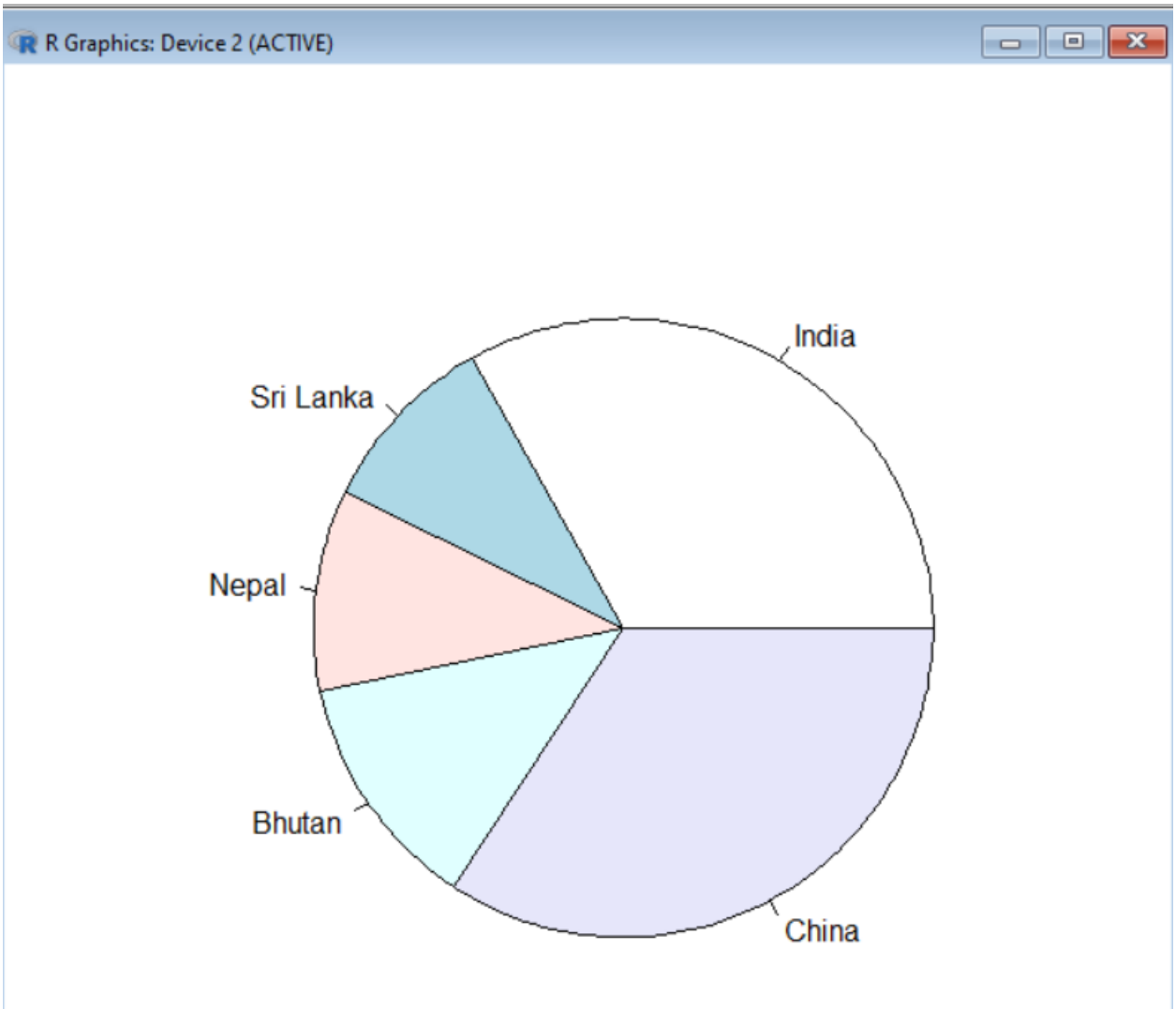


Output of histogram graphical representation..

23. **USING R PROGRAM CREATE A 3D PIE CHART FOR THE DATASET "POLITICAL KNOWLEDGE" WITH SUITABLE LABELS AND COLOURS.**

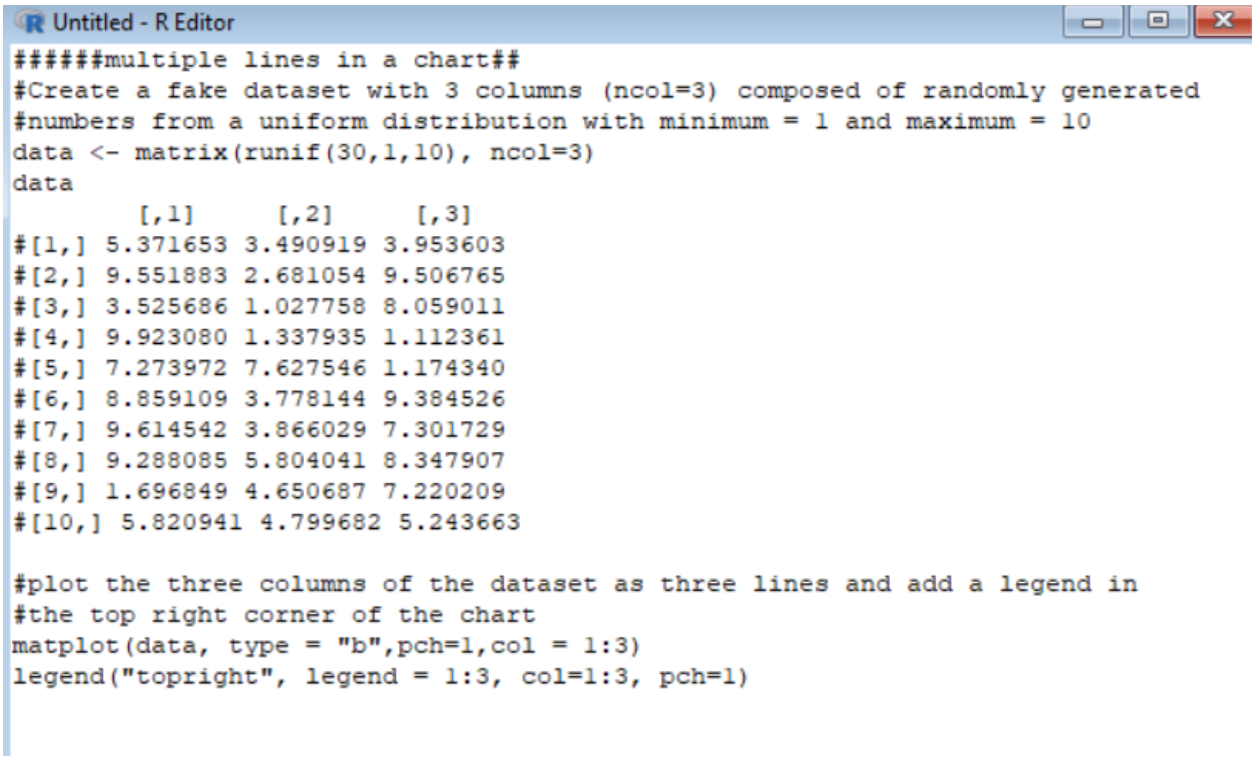


Code for creating the pie charts

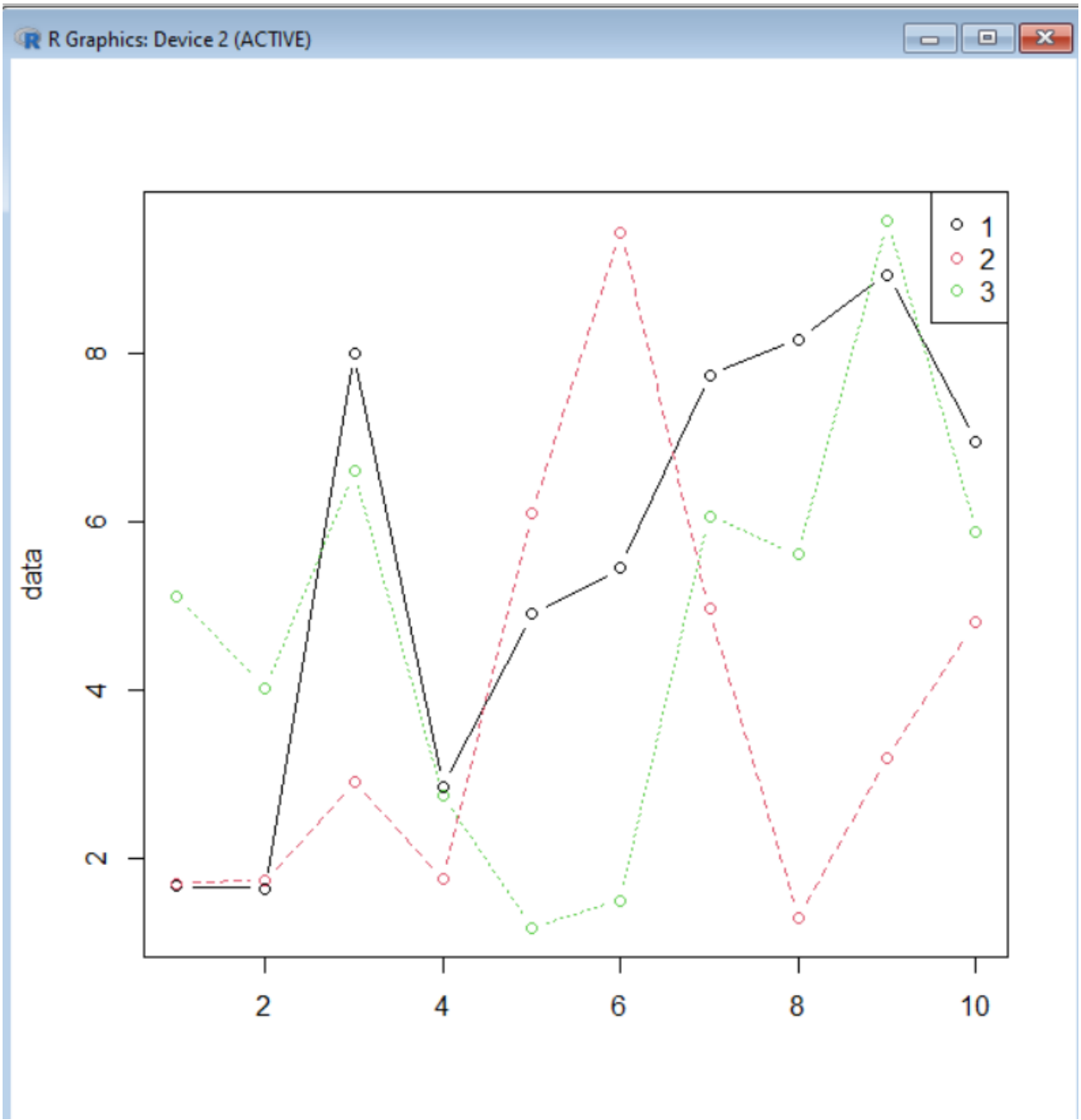


Output of created pie chart.

24. OBTAIN MULTIPLE LINES IN LINE CHART USING A SINGLE PLOT FUNCTION IN R.USE ATTRIBUTES "MPG"AND "QSEC" OF THE DATASET "MTCARS"

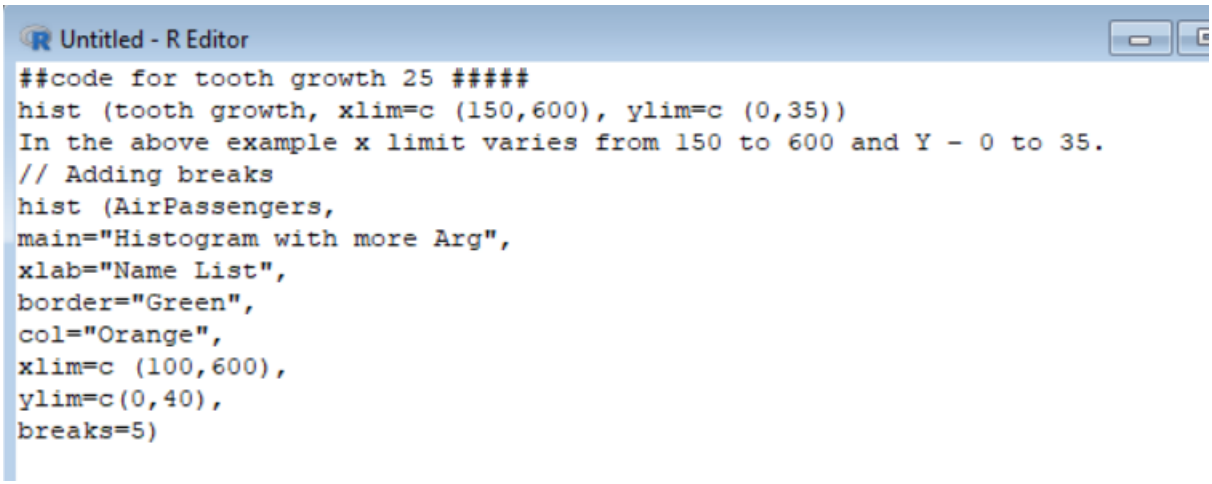


code for getting multiple lines in chart.

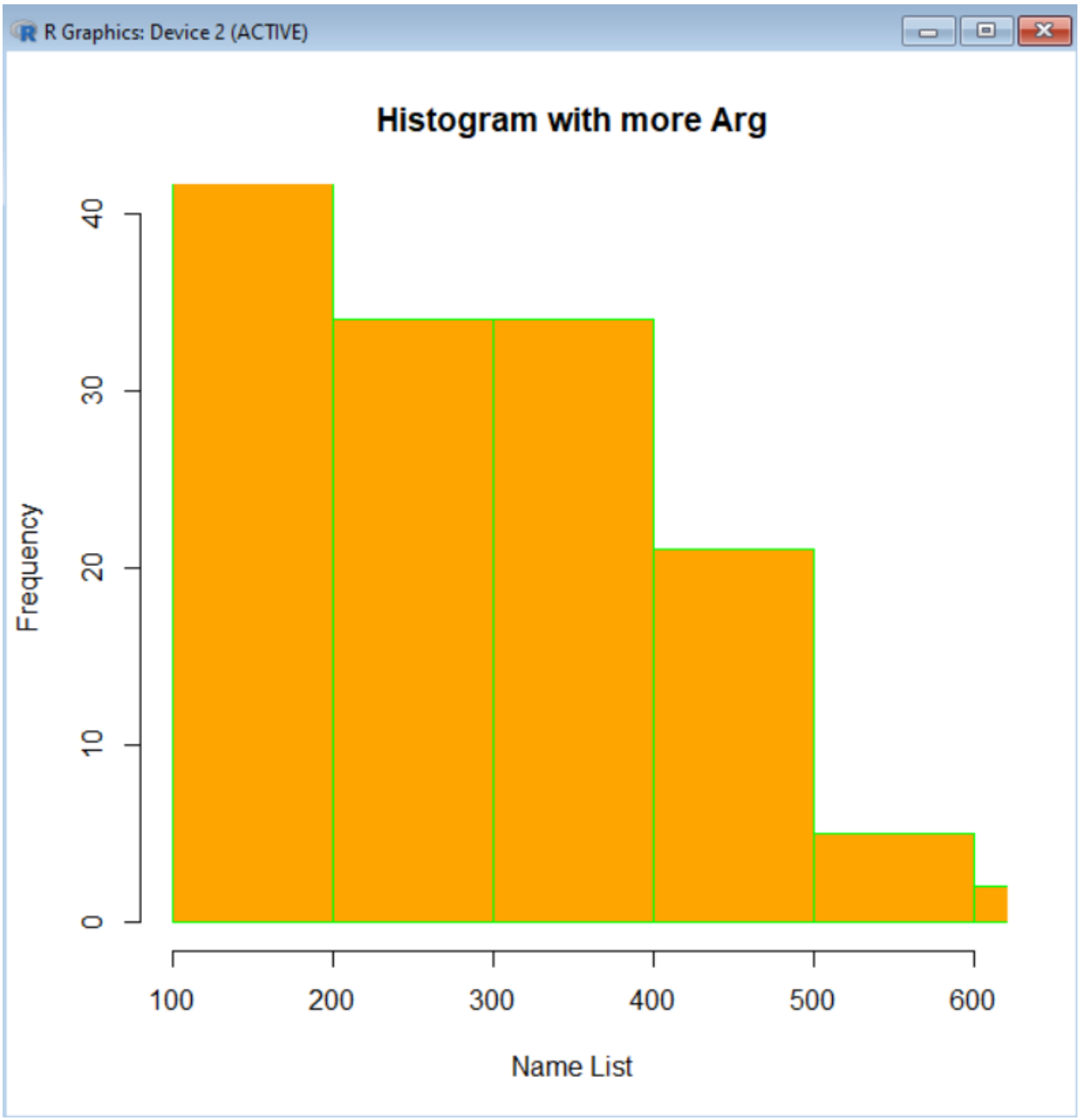


Output of multiple lines in chart.

25. USING R PROGRAM MAKE A HISTOGRAM FOR THE "TOOTHGROWTH" DATASET, START AT 100 ON THE X-AXIS, AND FROM VALUES 200 TO 700, MAKE THE BINS 150 WIDE



Code for creating a histogram for tooth growth.



Output of histogram, for tooth growth.