

Day_1

1) The intervals and corresponding frequencies are as follows. age frequency

1-5. 200

5-15 450

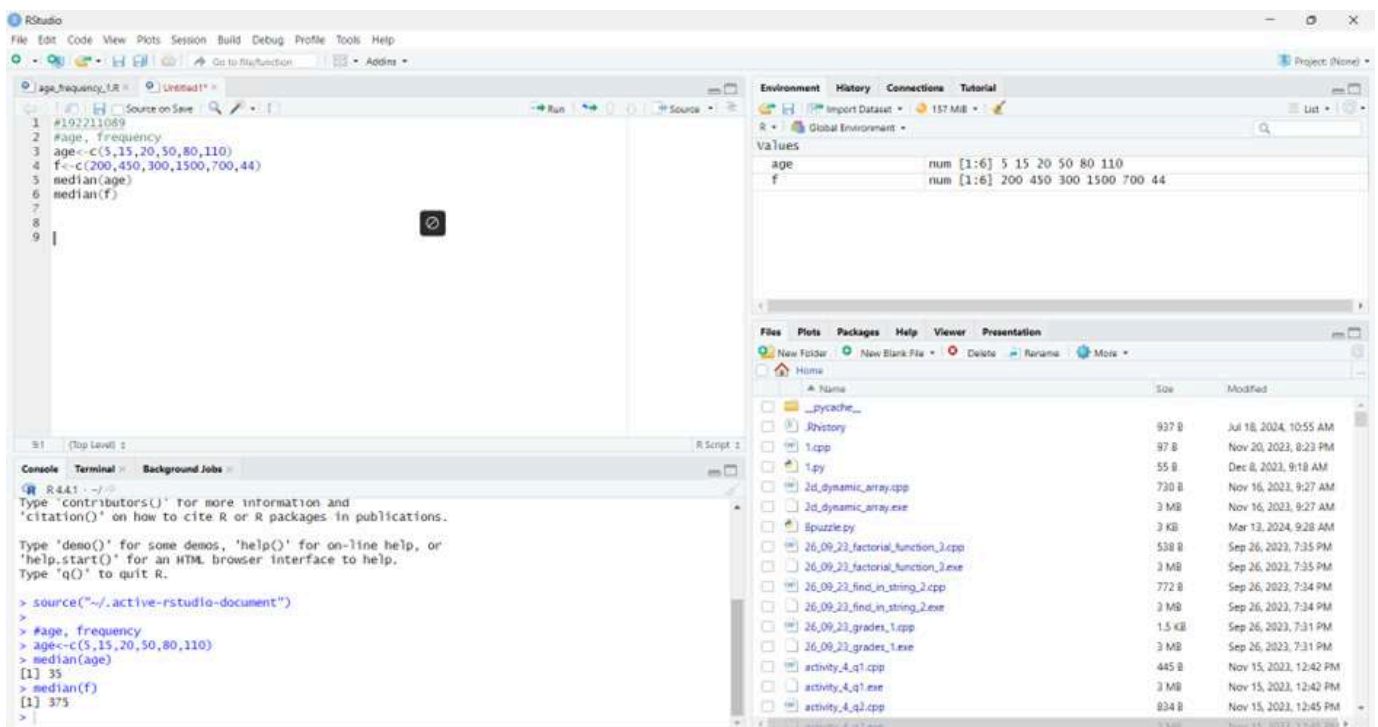
15-20 300

20-50 1500

50-80 700

80-110 44

Compute an approximate median value for the data



- 2) Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the mean of the data? What is the median?

(b) What is the mode of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).

(c) What is the midrange of the data?

(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

The screenshot shows the RStudio interface with a script editor, environment pane, and console. The script defines a vector 'age' and calculates various statistics. The console output shows the results of these calculations.

```
#mean, median, mode, quartile
age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
mean(age)
median(age)
mode_age<-names(table(age))[table(age)==max(table(age))]
mode_age
range(age)
quantile(age,.25)
quantile(age,.75)
```

Environment pane shows:

Variable	Value
age	num [1:27] 13 15 16 16 19 20 20 21 22 22 ...
mode_age	chr [1:2] "25" "35"

Console output:

```
R 4.4.1 ~ / ~
> #mean, median, mode, quartile
> age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
> mean(age)
[1] 29.96296
> median(age)
[1] 25
> mode_age
[1] "25" "35"
> range(age)
[1] 13 70
> quantile(age,.25)
25%
20.5
> quantile(age,.75)
75%
35
```

3&4 in same code:

3) Data Preprocessing: Reduction and Transformation

Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000 (a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization

4.Data:11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75

a) Smoothing by bin mean

b) Smoothing by bin median

c) Smoothing by bin boundaries

```
#192211089
1 data <- c(11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75)
2 bins <- 5
3 bin_indices <- cut(data, bins)
4 mean_smooth <- tapply(data, bin_indices, mean)
5 print(mean_smooth)
6 median_smooth <- tapply(data, bin_indices, median)
7 print(median_smooth)
8 min_max_smooth <- tapply(data, bin_indices, function(x) c(min(x), max(x)))
9 print(min_max_smooth)
```

Console Output:

```
> #192211089
> data <- c(11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75)
> print(mean_smooth)
(10.9,23.8] (23.8,36.6] (36.6,49.4] (49.4,62.2] (62.2,75.1]
17.78571 27.00000 43.75000 NA 72.75000
> print(median_smooth)
$ (10.9,23.8]
[1] 11 23

$ (23.8,36.6]
[1] 24 30

$ (36.6,49.4]
[1] 40 45
```

Environment Panel:

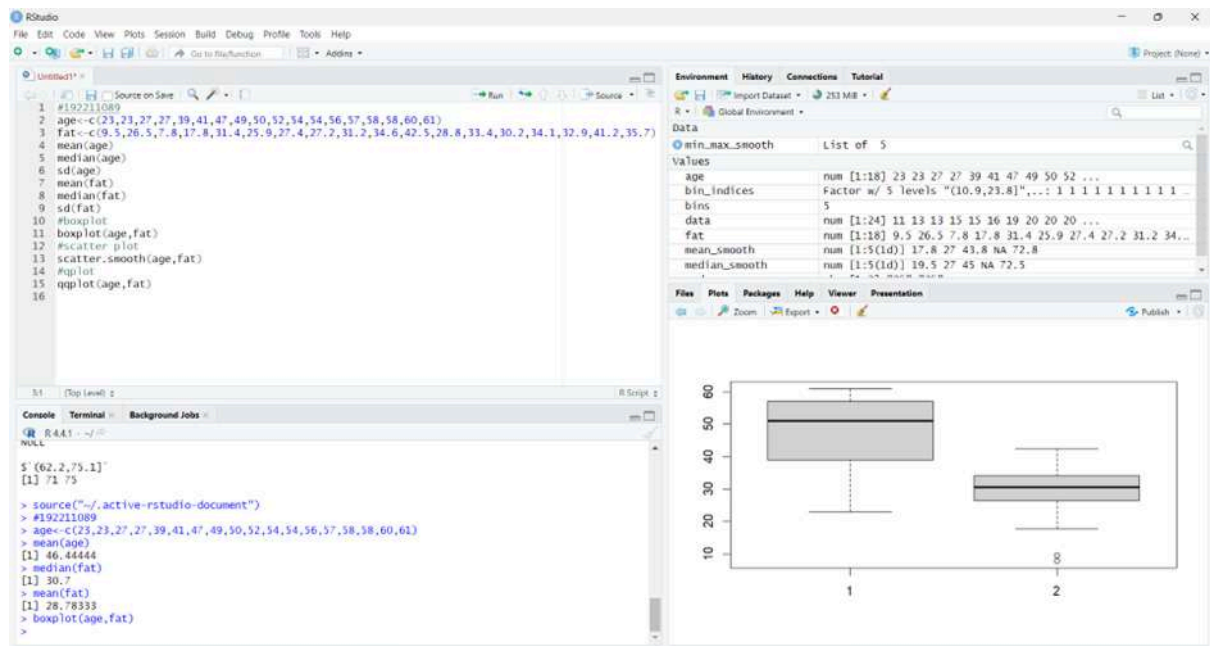
min_max_smooth	List of 5
age	num [1:27] 13 15 16 16 19 20 20 21 22 22 ...
bin_indices	Factor w/ 5 levels "(10.9,23.8]"...: 1 1 1 1 1 1 1 1 1 1
bins	5
data	num [1:24] 11 13 13 15 15 16 19 20 20 20 ...
mean_smooth	num [1:5(1d)] 17.8 27 43.8 NA 72.8
median_smooth	num [1:5(1d)] 19.5 27 45 NA 72.5
mode_age	chr [1:2] "25" "35"

5) Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

(a) Calculate the mean, median, and standard deviation of age and %fat.

(b) Draw the boxplots for age and %fat.

(c) Draw a scatter plot and a q-q plot based on these two variables.

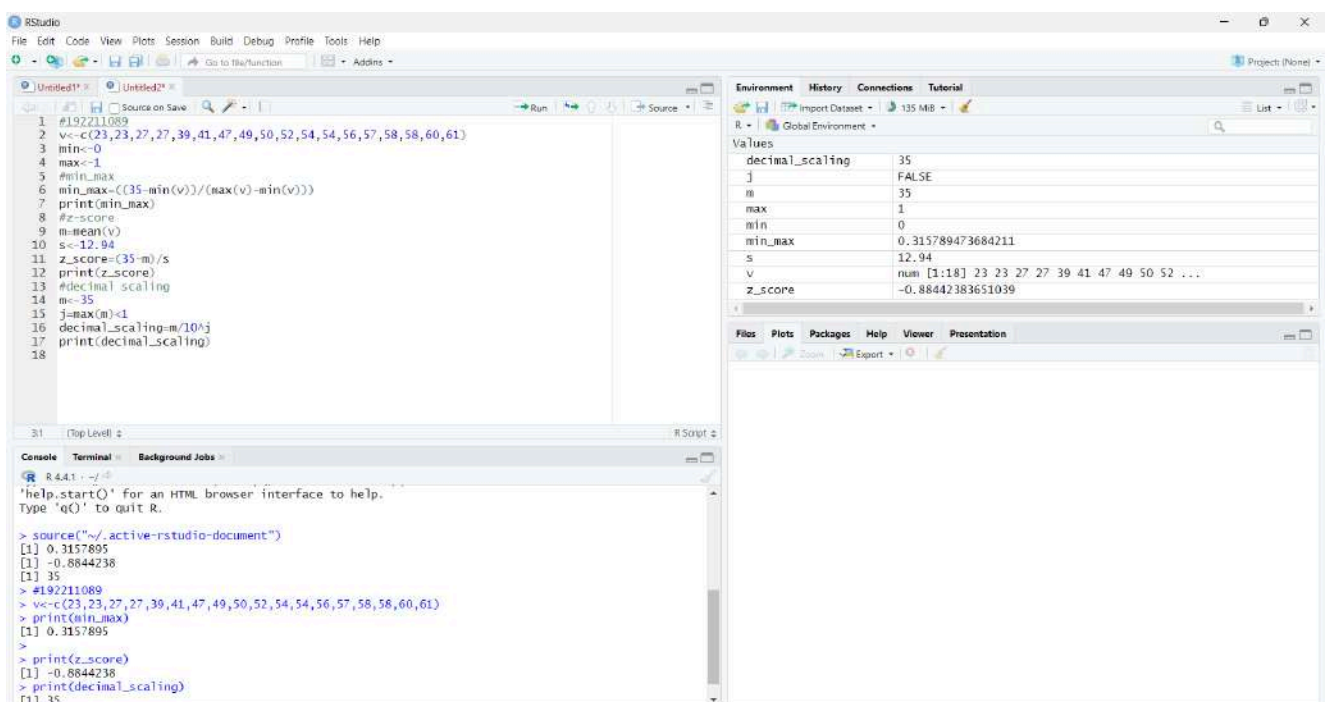


6) Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

(i) Use min-max normalization to transform the value 35 for age onto the range [0.0, 1.0].

(ii) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years.

(iii) Use normalization by decimal scaling to transform the value 35 for age. Perform the above functions using R – tool



7) The following values are the number of pencils available in the different boxes. Create a vector and find out the mean, median and mode values of set of pencils in the given data.

Box1	Box2	Box3	Box4	Box5	Box6	Box7	Box8	Box9	Box 10
9	25	23	12	11	6	7	8	9	10

The screenshot shows the RStudio interface with the following content:

```
#192211089
1 pencils<-c(9,25,23,12,11,6,7,8,9,10)
2 mean(pencils)
3 median(pencils)
4 mode=names(table(pencils))[table(pencils)==max(table(pencils))]
5 mode
6
7
```

Environment

mode	num
"g"	[1:10] 9 25 23 12 11 6 7 8 9 10

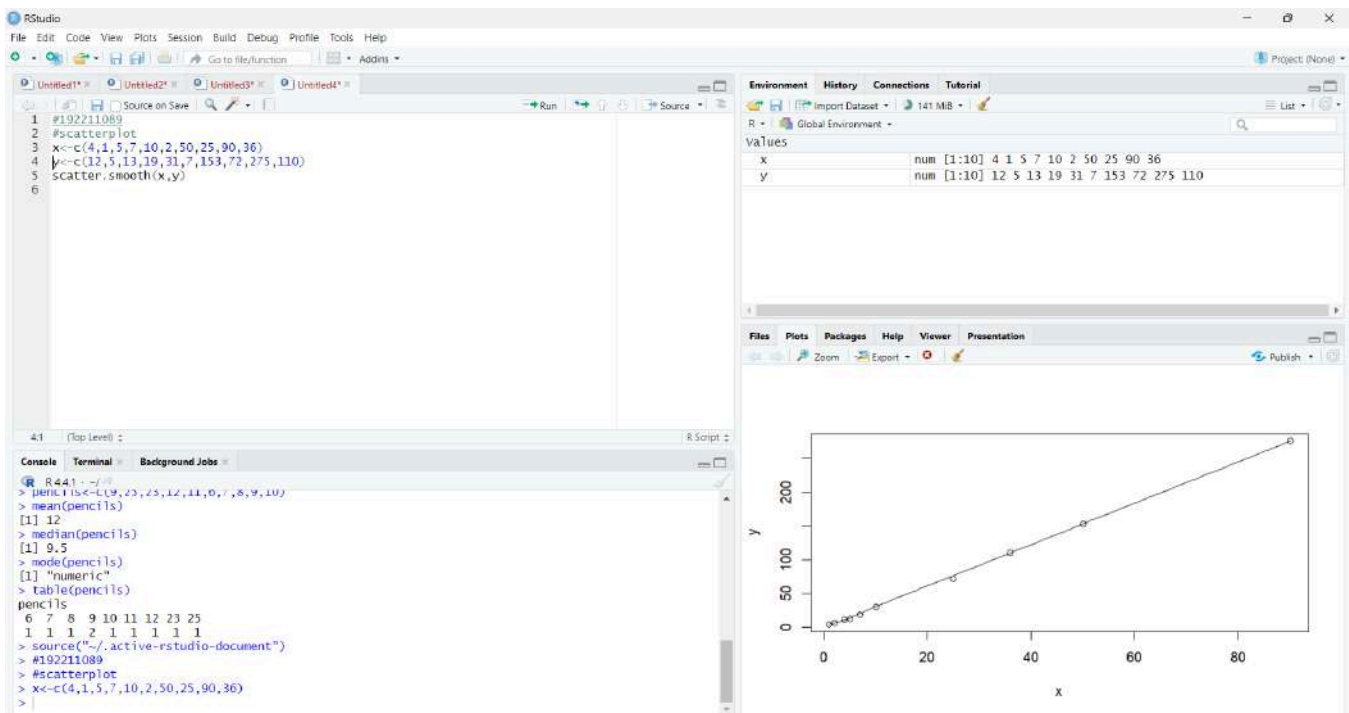
Console

```
R 4.4.1 ~ /...
[1] >>
> source("~/active-rstudio-document")
> source("~/active-rstudio-document")
> #192211089
> pencils<-c(9,25,23,12,11,6,7,8,9,10)
> mean(pencils)
[1] 12
> median(pencils)
[1] 9.5
> mode(pencils)
[1] "numeric"
> table(pencils)
pencils
 6  7  8  9 10 11 12 23 25
1  1  1  2  1  1  1  1  1
> |
```

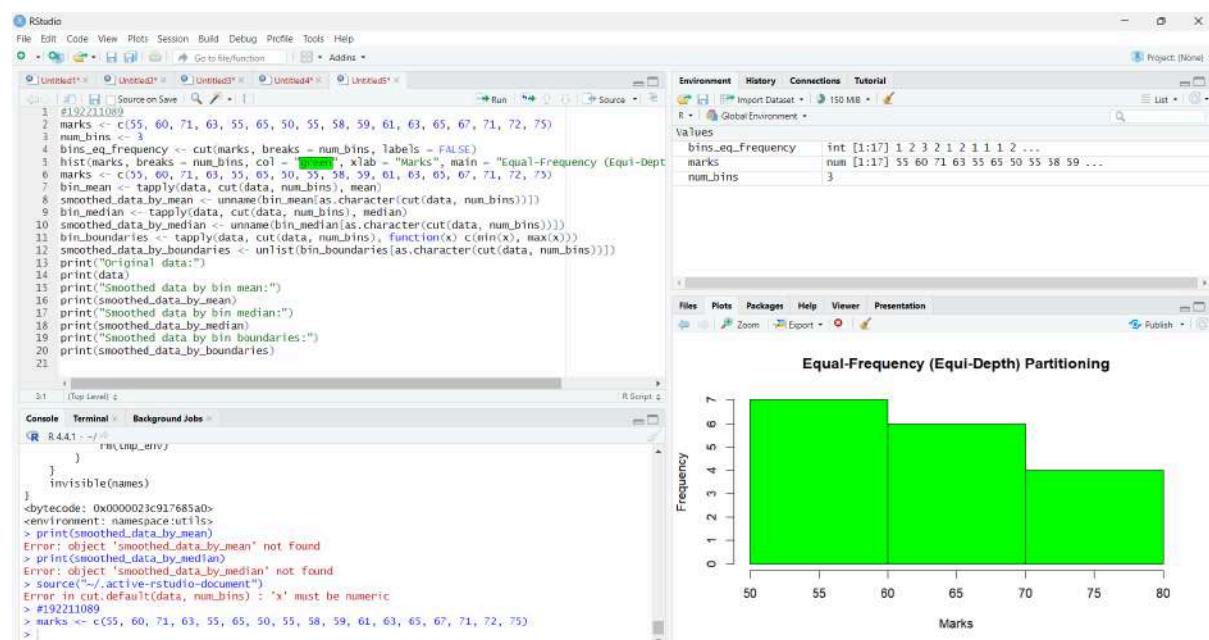
8) . The following table would be plotted as (x,y) points, with the first column being the x values as several mobile phones sold and the second column being the y values as money. To use the scatter plot for how many mobile phones are sold.

x:4 1 5 7 10 2 50 25 90 36

y:12 5 13 19 31 7 153 72 275 110



9) Implementing the R script using marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75. They are partitioned into three bins using each of the following methods. Plot the data points using the histogram.a) equal-frequency (equi-depth) partitioning (b) equal-width partitioning



10) Suppose that the speed car is mentioned in different driving style.Regular 78.3 81.8 82 74.2 83.4 84.5 82.9 77.5 80.9 70.6 SpeedCalculate the Inter quantile and standard deviation of the given data.

```

1 #192211089
2 #IQR, SD
3 v<-c(78.3,81.8,82,74.2,83.4,84.5,82.9,77.5,80.9,70.6)
4 IQR(v)
5 sd(v)
6

```

```

R 4.4.1 ~ />
> source("~/active-rstudio-document")
> scatter.smooth(x,y)
> #scatterplot
> x<-c(4,1,3,7,10,2,50,25,90,36)
> y<-c(12,5,13,19,31,7,153,72,275,110)
> scatter.smooth(x,y)
>
> source("~/active-rstudio-document")
> #192211089
> #IQR, SD
> v<-c(78.3,81.8,82,74.2,83.4,84.5,82.9,77.5,80.9,70.6)
> IQR(v)
[1] 4.975
> sd(v)
[1] 4.445835
>

```

Environment: Global Environment
 Values: v num [1:10] 78.3 81.8 82 74.2 83.4 84.5 82.9 77.5 80.9 70.6

11) Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

```

1 #192211089
2 #Q1, Q3
3 age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
4 quantile(age,.25)
5 quantile(age,.75)
6

```

```

R 4.4.1 ~ />
> source("~/active-rstudio-document")
> #192211089
> #Q1, Q3
> age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
> quantile(age,.25)
25%
20.5
> quantile(age,.75)
75%
35
>

```

Environment: Global Environment
 Values: age num [1:27] 13 15 16 16 19 20 20 21 22 22 ...

DAY_02

1) 1.Covariance and correlation Children of three ages are asked to indicate their preference for three photographs of adults. Do the data suggest that there is a significant relationship between age and photograph preference? What is wrong with this study?

Photograph: Age

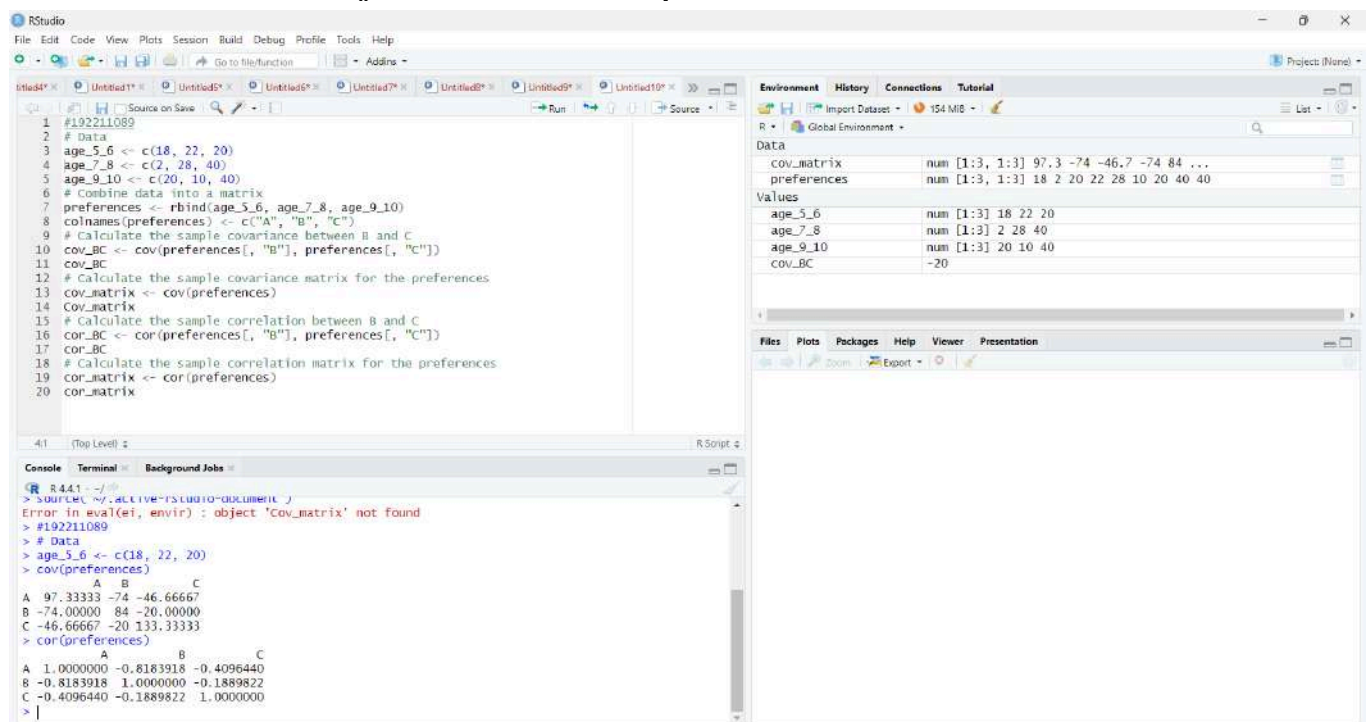
of child A B C 5-6 years: 18 22 20 7-8 years: 2 28 40 9-10 years: 20 10 40

1. Use `cov()` to calculate the sample covariance between B and C.

2. Use another call to `cov()` to calculate the sample covariance matrix for the preferences.

3. Use `cor()` to calculate the sample correlation between B and C.

4. Use another call to `cor()` to calculate the sample correlation matrix for the



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
#192211089
# Data
age_5_6 <- c(18, 22, 20)
age_7_8 <- c(2, 28, 40)
age_9_10 <- c(20, 10, 40)
# Combine data into a matrix
preferences <- rbind(age_5_6, age_7_8, age_9_10)
colnames(preferences) <- c("A", "B", "C")
# Calculate the sample covariance between B and C
cov_BC <- cov(preferences[, "B"], preferences[, "C"])
cov_BC
# Calculate the sample covariance matrix for the preferences
cov_matrix <- cov(preferences)
cov_matrix
# Calculate the sample correlation between B and C
cor_BC <- cor(preferences[, "B"], preferences[, "C"])
cor_BC
# Calculate the sample correlation matrix for the preferences
cor_matrix <- cor(preferences)
cor_matrix
```

The Environment pane shows the following objects:

Object	Class	Dimensions	Values
cov_matrix	num	[1:3, 1:3]	97.3 -74 -46.7 -74 84 ...
preferences	num	[1:3, 1:3]	18 2 20 22 28 10 20 40 40
age_5_6	num	[1:3]	18 22 20
age_7_8	num	[1:3]	2 28 40
age_9_10	num	[1:3]	20 10 40
cov_BC	num		-20

The Console pane shows the following output:

```
R4.4.1 ~ /~
> source("~/active-rstudio-document")
Error in eval(ei, envir) : object 'cov_matrix' not found
> #192211089
> # Data
> age_5_6 <- c(18, 22, 20)
> cov(preferences)
      A      B      C
A 97.33333 -74 -46.66667
B -74.00000 84 -20.00000
C -46.66667 -20 133.33333
> cor(preferences)
      A      B      C
A 1.0000000 -0.8183918 -0.4096440
B -0.8183918 1.0000000 -0.1889822
C -0.4096440 -0.1889822 1.0000000
> |
```

2) Imagine that you have selected data from the All Electronics data warehouse for analysis.

The data set will be huge! The following data are a list of All Electronics prices for commonly

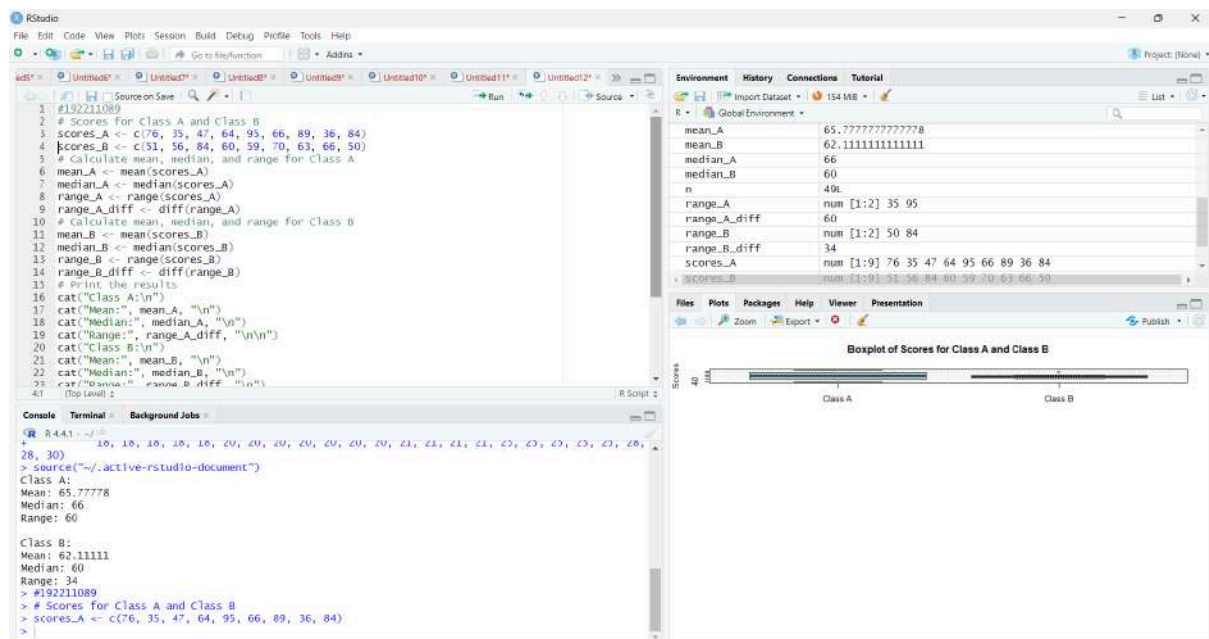
sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5,

8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 8, 18, 20, 20,

20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30 the dataset using an

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data manipulation and plotting. The code defines a vector 'data', calculates bin sizes and means, and uses 'ggplot2' to create three histograms: 'Original Data Histogram', 'Data Smoothed by Bin Means', and 'Data Smoothed by Bin Boundaries'.
- Console:** Shows the execution output of the R code, including the creation of the 'data' vector and the execution of the 'ggplot' functions.
- Environment:** Lists the objects in the R environment, including 'age_3_6', 'age_7_8', 'age_9_10', 'bin_means', 'bin_size', 'cov_BC', 'data', 'k', 'n', 'smoothed_by_means', and 'smoothed_by_boundaries'.
- Plots:** Three histograms are displayed:
 - Original Data Histogram:** A blue bar chart showing the frequency of data points across price bins.
 - Data Smoothed by Bin Means:** A green bar chart showing the smoothed data using bin means.
 - Data Smoothed by Bin Boundaries:** A red bar chart showing the smoothed data using bin boundaries.

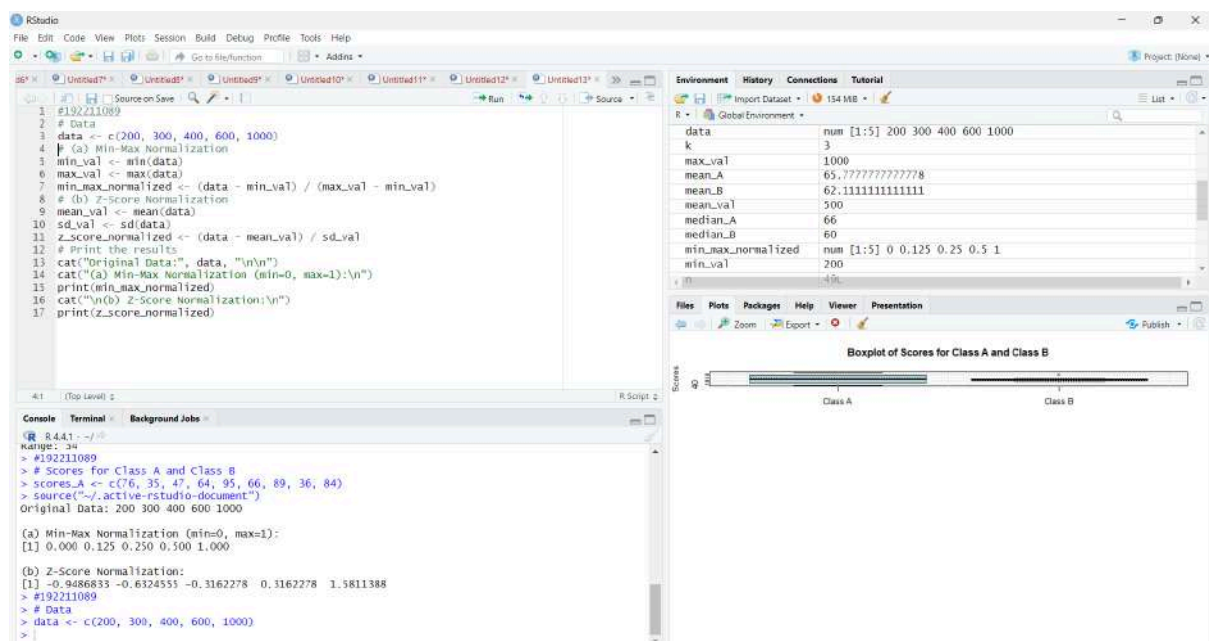


4) Let us consider one example to make the calculation method clear.
Assume that the

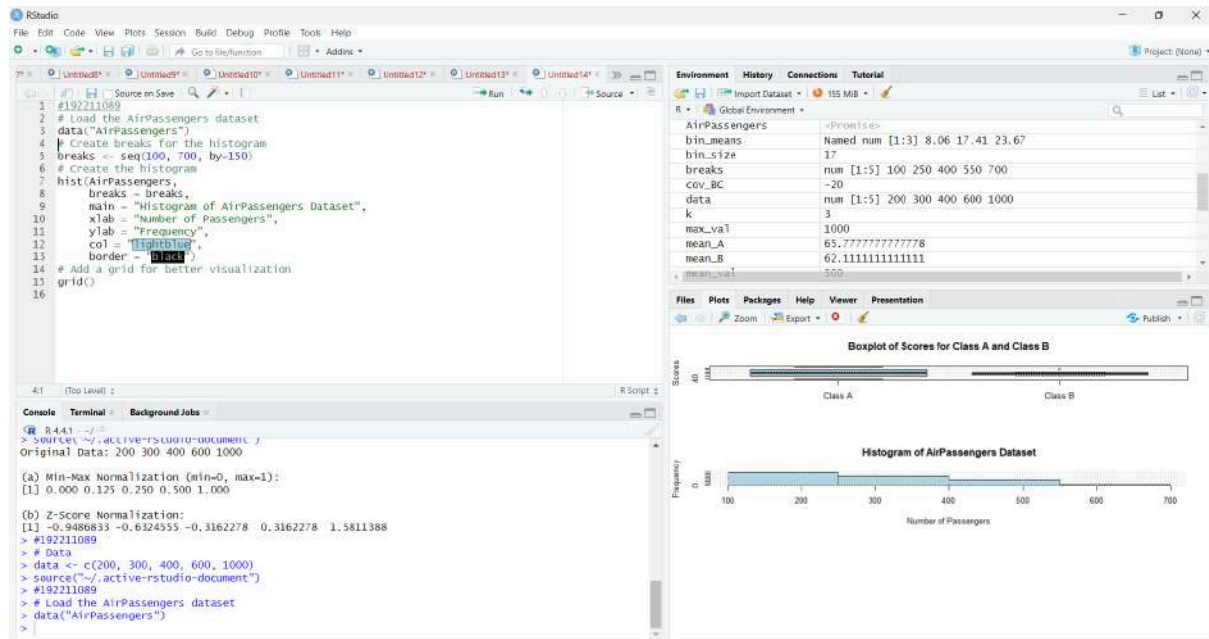
minimum and maximum values for the feature F are \$50,000 and \$100,000 correspondingly.

It needs to range F from 0 to 1. By min-max normalization, $v = \$80$, b) Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000

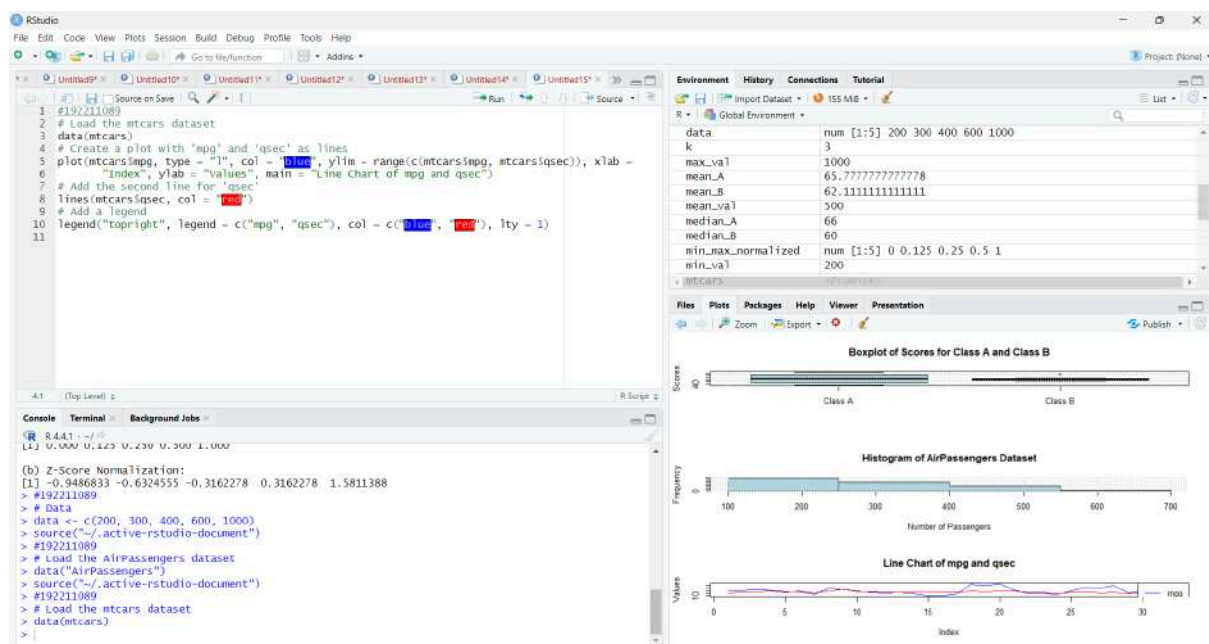
(a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization



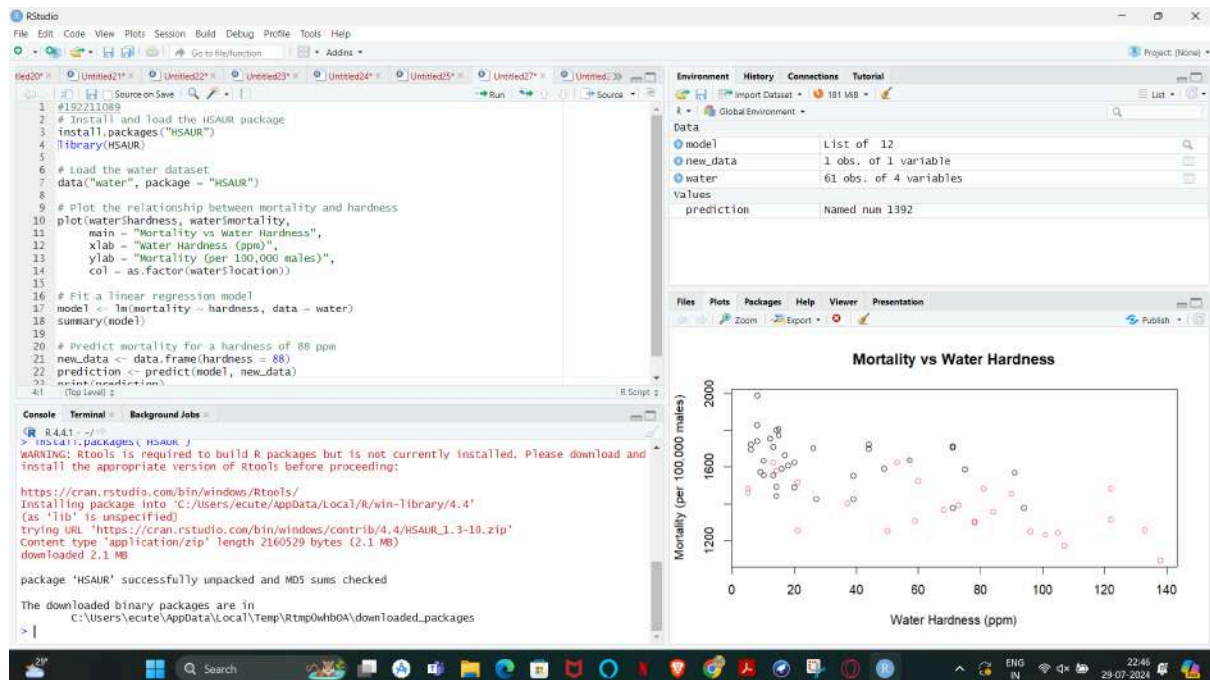
5) Make a histogram for the “AirPassengers” dataset, starting at 100 on the x-axis, and from values 200 to 700, make the bins 150 wide



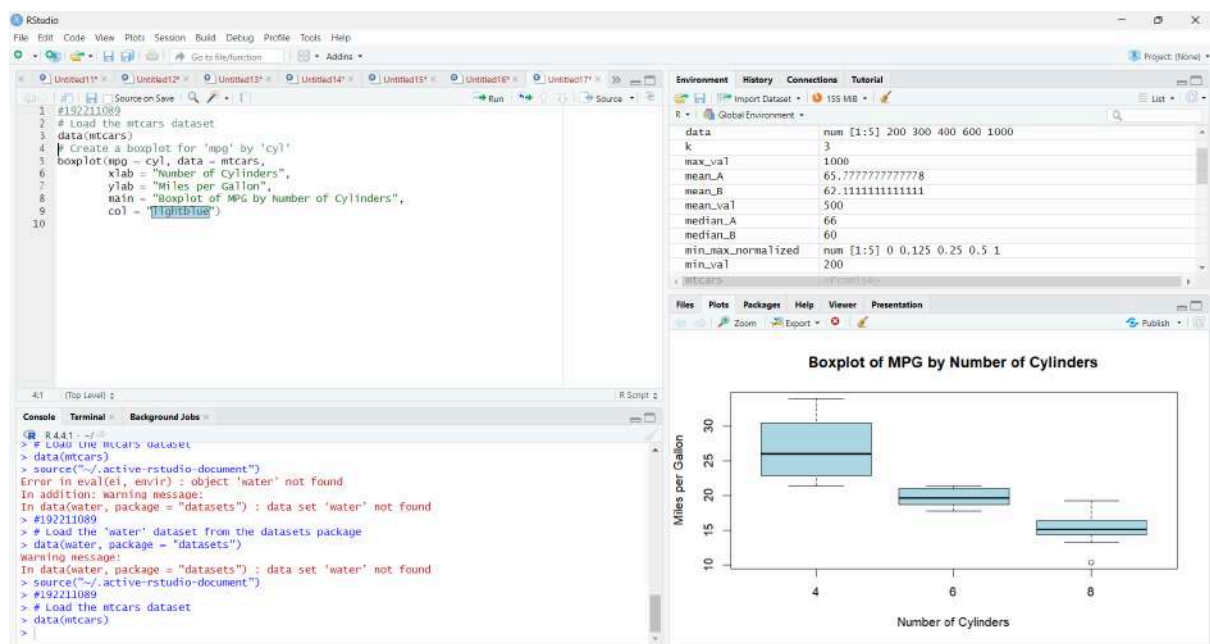
6) Obtain Multiple Lines in a Line Chart using a single Plot Function in Use attributes “mpg” and “sec” of the dataset “mtcars”
Code:



7) Download the Dataset "water" From the R dataset Link. Find out whether there is a linear relation between attributes "mortality" and "hardness" by plot function. Fit

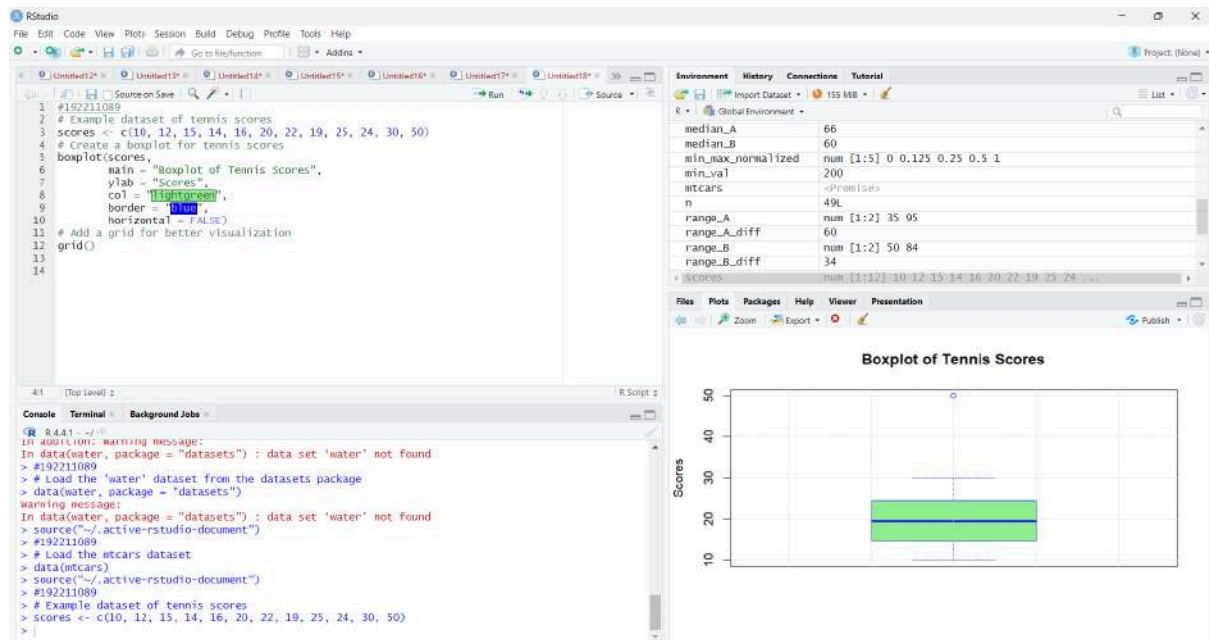


8) 8.Create a Boxplot graph for the relation between "mpg"(miles per gallon) and "cyl"(number of Cylinders) for the dataset "mtcars" available in R Environment
Code:



9) Assume the Tennis coach wants to determine if any of his team players are scoring outliers. To visualize the distribution of points scored by his players, then how can he decide to develop the box plot? Give a suitable example using the Boxplot visualization technique.

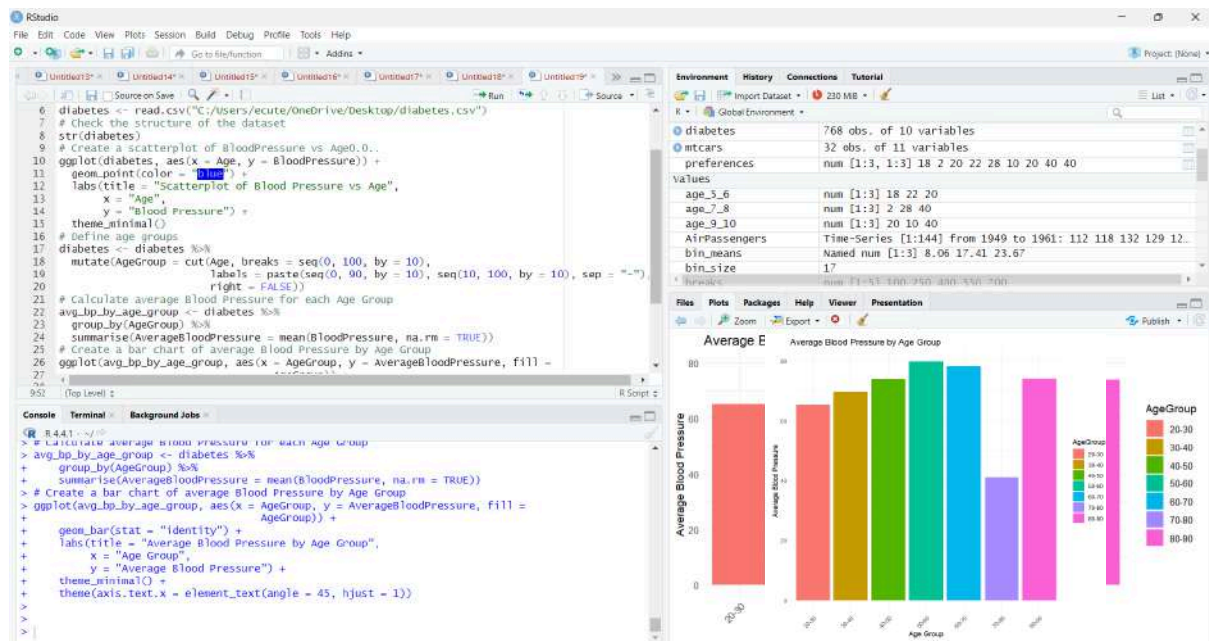
Code:



10) Implement using R language in which age group of people are affected by blood pressure based on the diabetes dataset show it using scatterplot and bar chart (that is BloodPressure

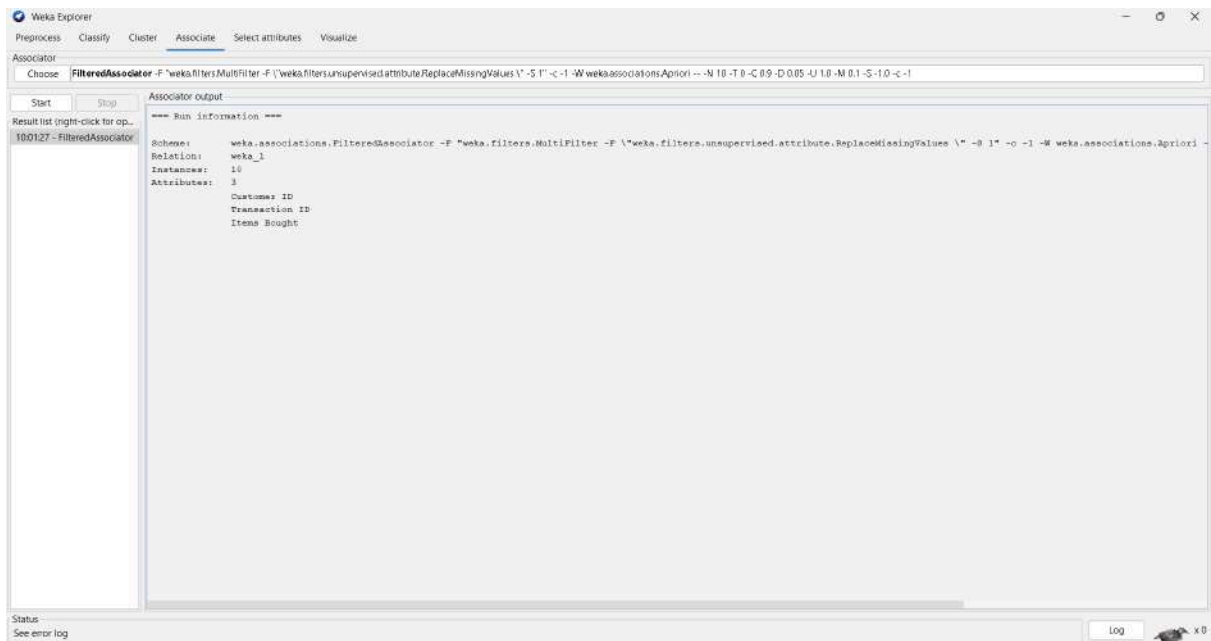
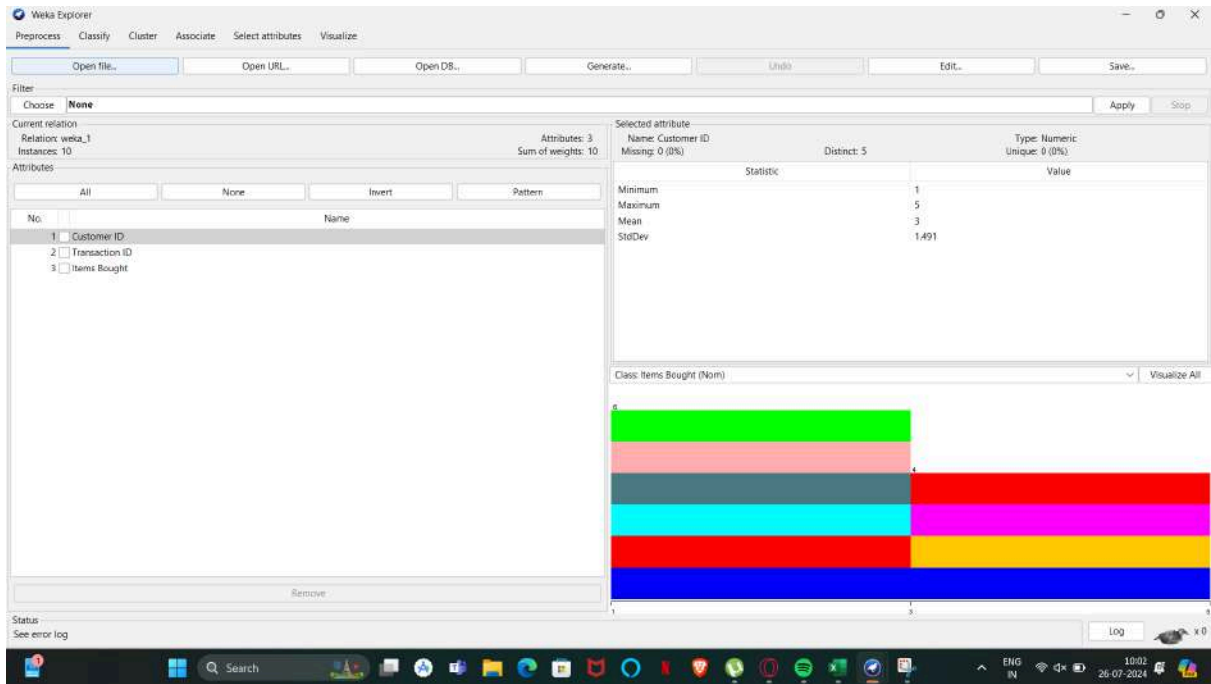
vs Age using dataset "diabetes.csv")

Code:



Day_3

- 1) Consider the data set and perform the Apriori Algorithm and FP algorithm support:3 and confidence=50%



2)

Transaction ID	Items Bought
1	{Milk, Beer, Diapers}
2	{Bread, Butter, Milk}
3	{Milk, Diapers, Cookies}
4	{Bread, Butter, Cookies}
5	{Beer, Cookies, Diapers}
6	{Milk, Diapers, Bread, Butter}
7	{Bread, Butter, Diapers}
8	{Beer, Diapers}
9	{Milk, Diapers, Bread, Butter}
10	{Beer, Cookies}

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Associator

Choose **FilteredAssociator** -f "weka.filters.MultiFilter -f "weka.filters.unsupervised.attribute.ReplaceMissingValues" -S 1" -c 1 -W weka.associators.Apriori -- -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Start Stop

Result list (right-click for op...)

100954 - FilteredAssociator

Associator output

=== Run information ===

Schema: weka.associations.FilteredAssociator -f "weka.filters.MultiFilter -f "weka.filters.unsupervised.attribute.ReplaceMissingValues" -S 1" -c 1 -W weka.associations.Apriori -- -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: d1g2

Instances: 10

Attributes: 2

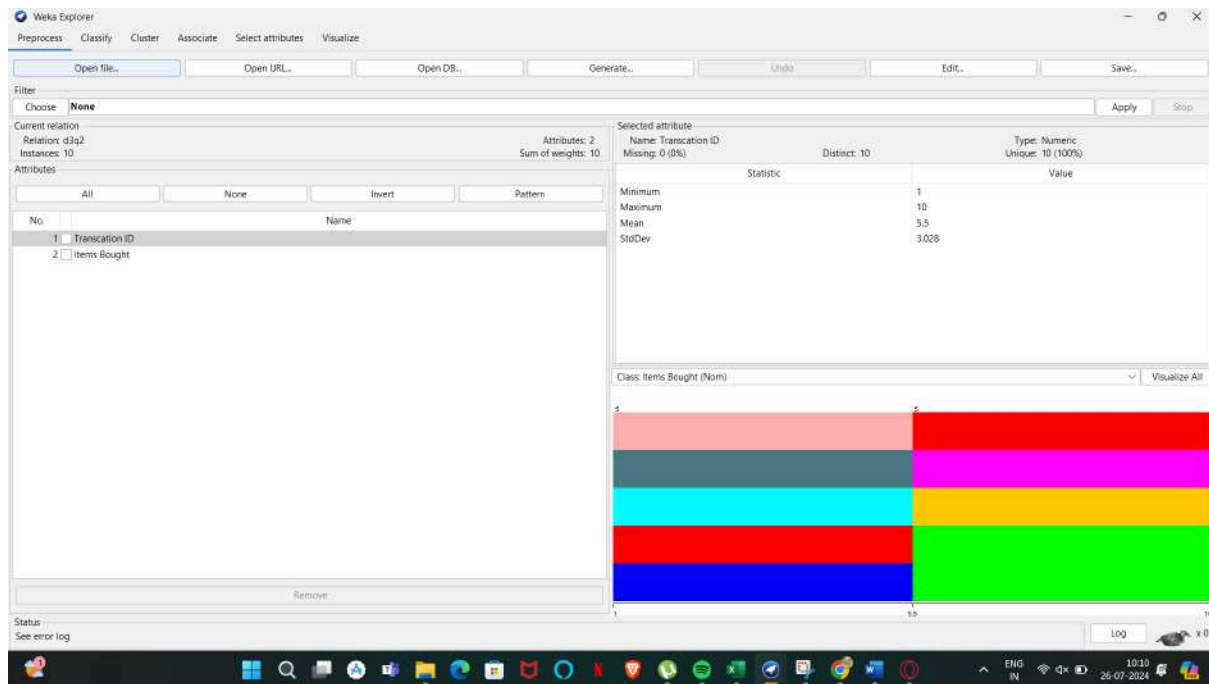
Transaction ID

Items Bought

Status: See error log

Log x 0

26-07-2024 10:09



3)

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 ... 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 ... 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 ... 40	medium	no	excellent	yes
13	31 ... 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **DecisionStump**

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 66
 More options...

(Nom) classbays_computer

Start Stop

Result list (right-click for options):
 101257 - trees.DecisionStump

Classifier output:

```

no      yes
0.0     1.0
age <= 31-40
no      yes
0.4     0.6
age is missing
no      yes
0.2857142857142857  0.7142857142857143
  
```

Time taken to build model: 0 seconds

==== Stratified cross-validation ====

==== Summary ====

Correctly Classified Instances 10 71.4286 %
 Incorrectly Classified Instances 4 28.5714 %
 Kappa statistic 0
 Mean absolute error 0.4751
 Root mean squared error 0.5717
 Relative absolute error 110.0466 %
 Root relative squared error 122.614 %
 Total Number of Instances 14

==== Detailed Accuracy By Class ====

	TP Rate	FP Rate	Precision	Recall	F-Measure	MD	ROC Area	PRC Area	Class
	0.000	0.000	?	0.000	?	?	0.200	0.243	no
	1.000	1.000	0.714	1.000	0.833	?	0.200	0.613	yes
Weighted Avg.	0.714	0.714	?	0.714	?	?	0.200	0.507	

==== Confusion Matrix ====

```

a b <-- classified as
0 4 | a = no
0 10 | b = yes
  
```

Status: OK

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **NaiveBayes**

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 66
 More options...

(Nom) classbays_computer

Start Stop

Result list (right-click for options):
 101257 - trees.DecisionStump
 101340 - bayes.NaiveBayes

Classifier output:

```

yes      no
[total]  6.0 12.0

credit_rating
fair      3.0 7.0
excellent 3.0 5.0
[total]   6.0 12.0
  
```

Time taken to build model: 0 seconds

==== Stratified cross-validation ====

==== Summary ====

Correctly Classified Instances 7 50 %
 Incorrectly Classified Instances 7 50 %
 Kappa statistic -0.3243
 Mean absolute error 0.5427
 Root mean squared error 0.6361
 Relative absolute error 131.5952 %
 Root relative squared error 134.6459 %
 Total Number of Instances 14

==== Detailed Accuracy By Class ====

	TP Rate	FP Rate	Precision	Recall	F-Measure	MD	ROC Area	PRC Area	Class
	0.000	0.300	0.000	0.000	0.000	-0.330	0.025	0.156	no
	0.700	1.000	0.636	0.700	0.667	-0.330	0.025	0.538	yes
Weighted Avg.	0.500	0.600	0.455	0.500	0.476	-0.330	0.025	0.440	

==== Confusion Matrix ====

```

a b <-- classified as
0 4 | a = no
3 7 | b = yes
  
```

Status: OK

4) Analysis the dataset “diabetes.csv” how the diabetes trend is for different age people, using linear regression and multiple regression.

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'Logistic - R 1.0 E 8 - M -1 - num-decimal-places 4'. The test options are set to 'Cross-validation' with 'Folds' set to 10. The classifier output shows the following results:

Classifier output

Attribute	Value
diab	0.9654
preg	1.0134
skin	0.9994
insu	1.0012
mass	0.9142
pedi	0.9886
age	0.9932

Time taken to build model: 0.05 seconds

==== Stratified cross-validation ====

==== Summary ====

Metric	Value
Correctly Classified Instances	555 / 77.2135 %
Incorrectly Classified Instances	175 / 22.7865 %
Kappa statistic	0.4734
Mean absolute error	0.2094
Root mean squared error	0.3954
Relative absolute error	68.0018 %
Root relative squared error	81.9651 %
Total Number of Instances	768

==== Detailed Accuracy By Class ====

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	PRC Area	Class
Weighted Avg.	0.571	0.120	0.718	0.571	0.634	0.832	0.715	tested_negative
	0.772	0.321	0.767	0.772	0.765	0.832	0.831	tested_positive

==== Confusion Matrix ====

a \ b	tested_negative	tested_positive
tested_negative	540	60
tested_positive	115	153

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'MultilayerPerceptron - 1.0.3 - M 0.2 - N 500 - V 0 - S 0 - E 20 - H a'. The test options are set to 'Cross-validation' with 'Folds' set to 10. The classifier output shows the following results:

Classifier output

Inputs Weights

Attribute	Weight
Attrib insu	-3.0550593615541673
Attrib mass	-8.775026462333566
Attrib pedi	-5.1523481596094115
Attrib age	5.26952973132554

Sigmoid Node 5

Inputs Weights

Attribute	Weight
Attrib preg	5.122015480585302
Attrib plas	-12.642913808440132
Attrib pres	5.675059474758244
Attrib skin	-0.06085501320153377
Attrib insu	2.3070185070104217
Attrib mass	-5.23208091635601
Attrib pedi	-0.7354842513650297
Attrib age	-19.2656337017977

Sigmoid Node 6

Inputs Weights

Attribute	Weight
Attrib preg	12.836762781789405
Attrib plas	-6.062276016682731
Attrib pres	-1.389684458164472
Attrib skin	0.3459708482063716
Attrib insu	-2.2115408147264896
Attrib mass	-0.9589656225258552
Attrib pedi	6.090003701393266
Attrib age	-0.833262465394625

Class tested_negative

Input

Node 0

Class tested_positive

Input

Node 1

Time taken to build model: 0.73 seconds

5)

Implement using WEKA for the given Suppose a database has five transactions. Let min sup= 50%(2) and min con f = 80%.

Transactions

Items

T1 (M, O, N, K, E, Y)

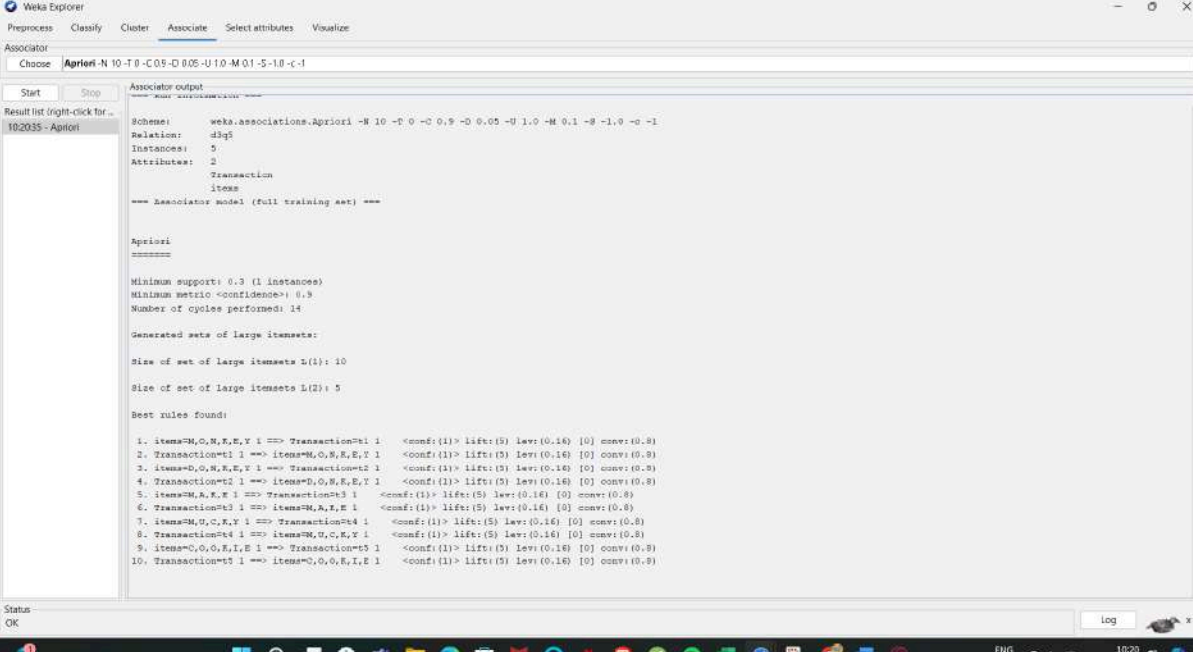
T2 (D, O, N, K, E, Y)

T3 (M, A, K, E)

T4 (M, U, C, K, Y)

T5 (C,O, O, K, I,E)

- Find all frequent item sets using Apriori algorithm
- Also draw FP-Growth Tree



The screenshot shows the Weka Explorer interface with the Apriori algorithm results. The 'Associate' tab is selected, and the 'Apriori' model is chosen. The 'Associate output' pane displays the following text:

```
==== Associate output =====
Scheme: weka.associations.Apriori -N 10 -P 0 -D 0.9 -O 0.05 -U 1.0 -M 0.1 -B -1.0 -d -1
Relation: dig3
Instances: 5
Attributes: 2
Transaction
Items

=== Apriori model (full training set) ===

Apriori
=====

Minimum supports: 0.3 (1 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 14

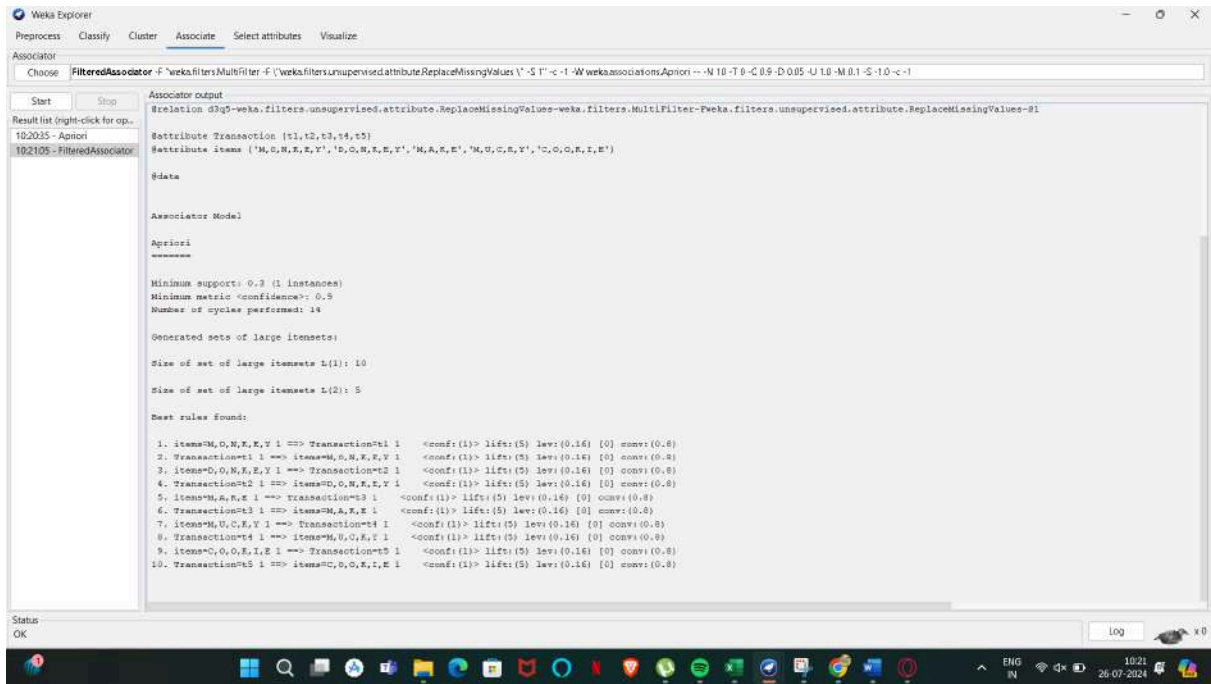
Generated sets of large itemsets:

Size of set of large itemsets L(1): 10
Size of set of large itemsets L(2): 5

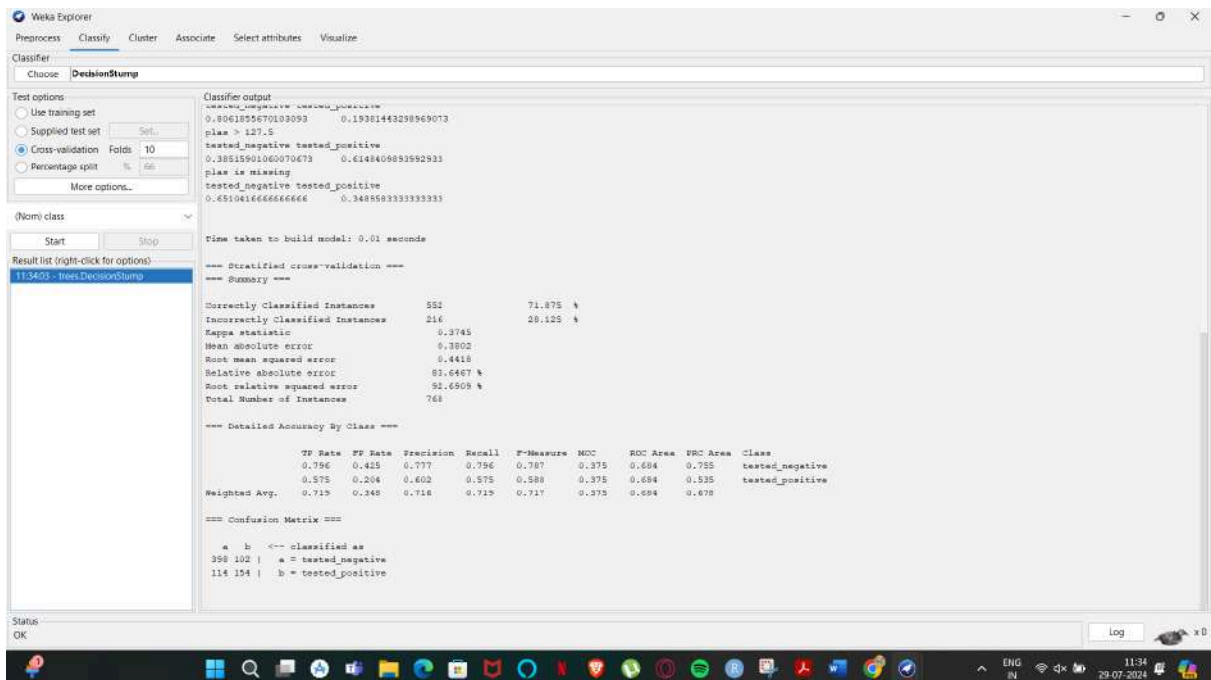
Best rules found:

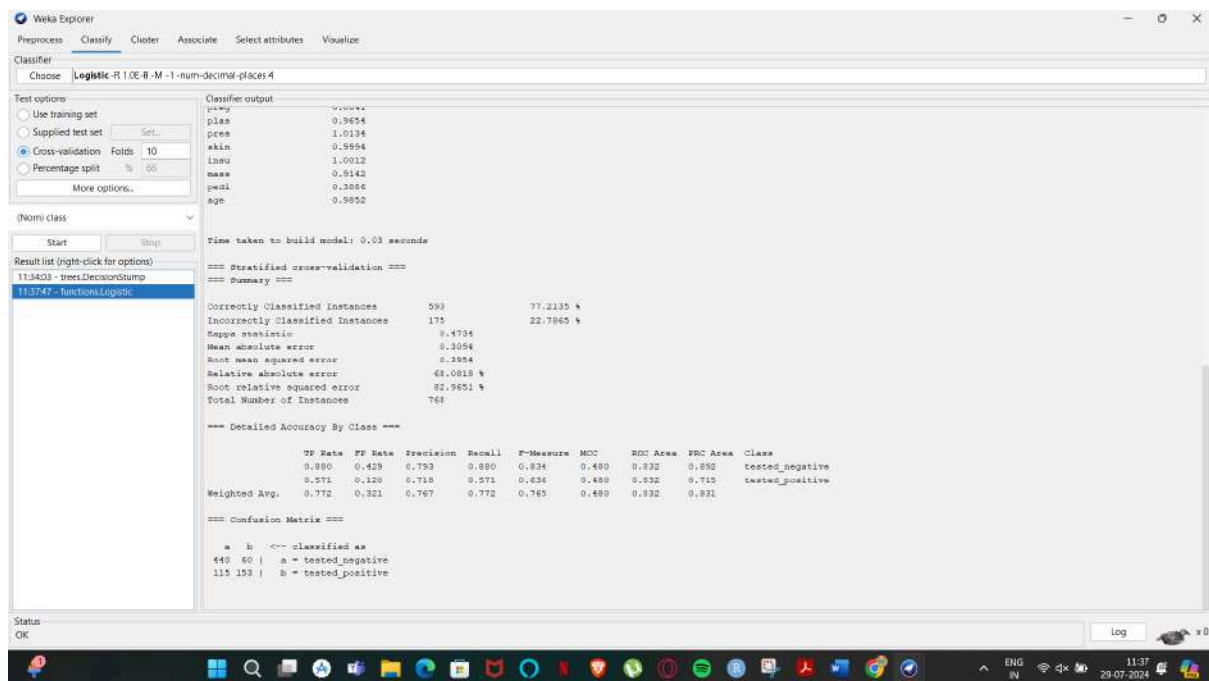
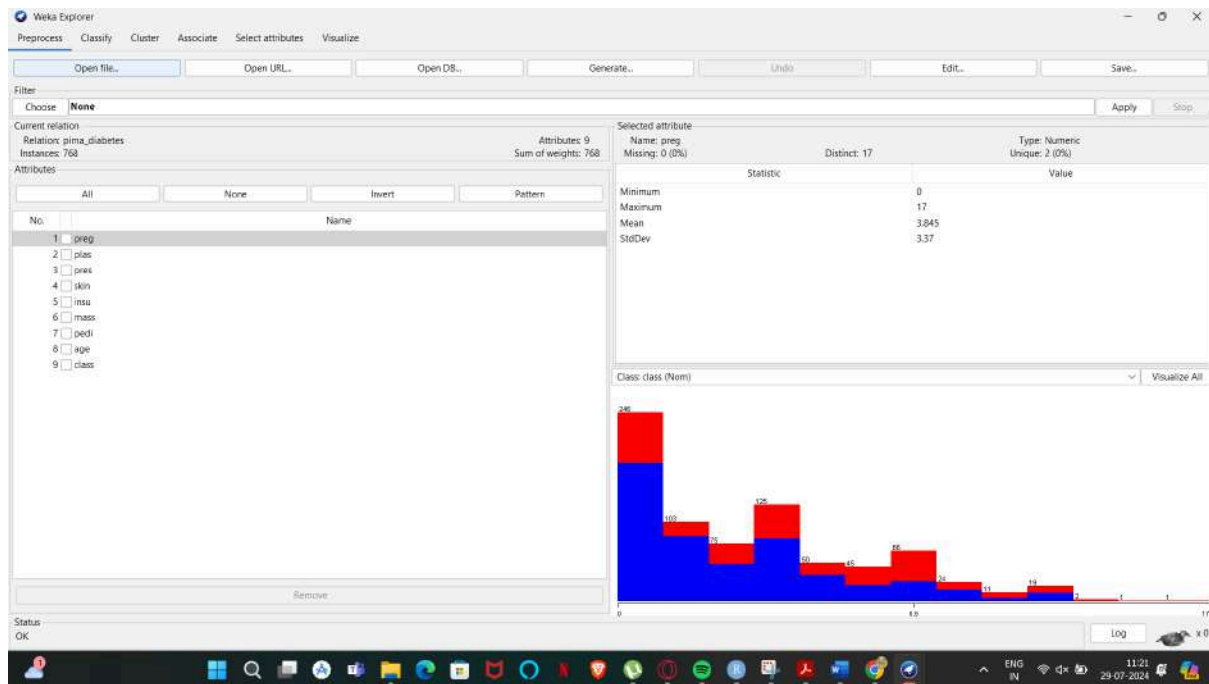
1. items=M,O,N,K,E,Y 1 ==> Transaction=t1 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
2. Transaction=t1 1 ==> items=M,O,N,K,E,Y 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
3. items=D,O,N,K,E,Y 1 ==> Transaction=t2 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
4. Transaction=t2 1 ==> items=D,O,N,K,E,Y 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
5. items=M,A,K,E 1 ==> Transaction=t3 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
6. Transaction=t3 1 ==> items=M,A,K,E 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
7. items=M,U,C,K,Y 1 ==> Transaction=t4 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
8. Transaction=t4 1 ==> items=M,U,C,K,Y 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
9. items=C,O,O,K,I,E 1 ==> Transaction=t5 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
10. Transaction=t5 1 ==> items=C,O,O,K,I,E 1 <conf: (1)> lift: (5) lev: (0.16) [0] conv: (0.8)
```

The status bar at the bottom shows 'Status: OK' and a 'Log' button.



6) Prediction of Categorical Data using Decision Tree Algorithm through WEKA using any datasets. a) Tree b) Preprocess c) Logistic

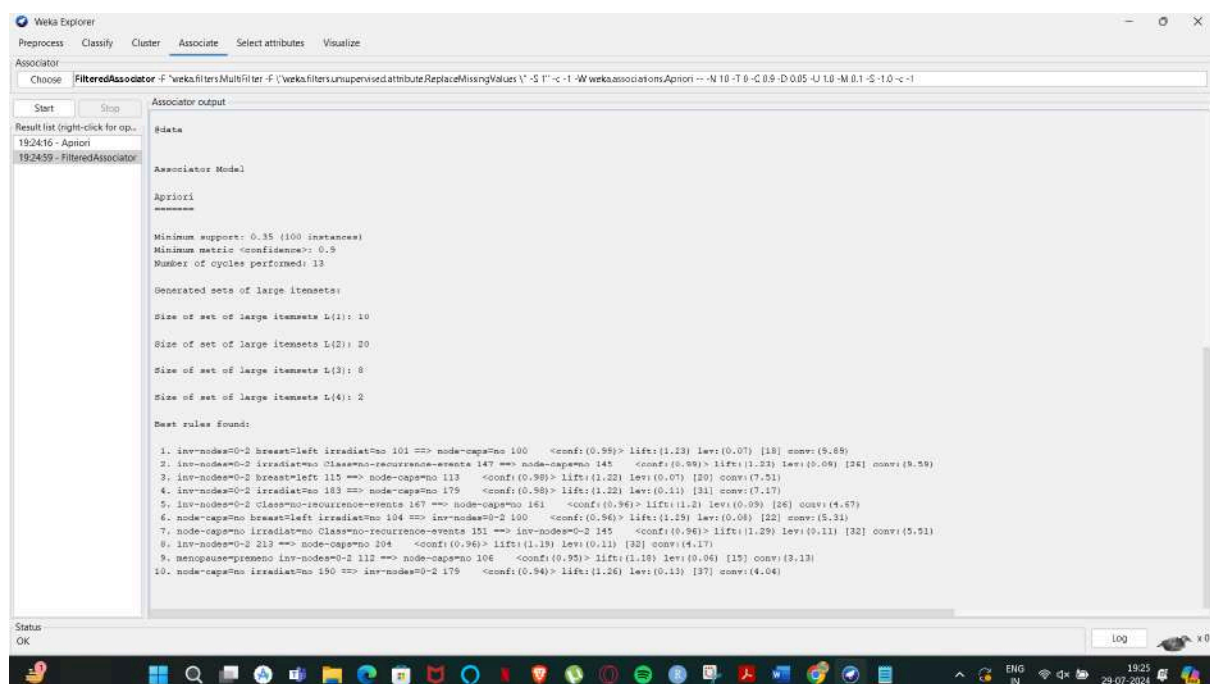
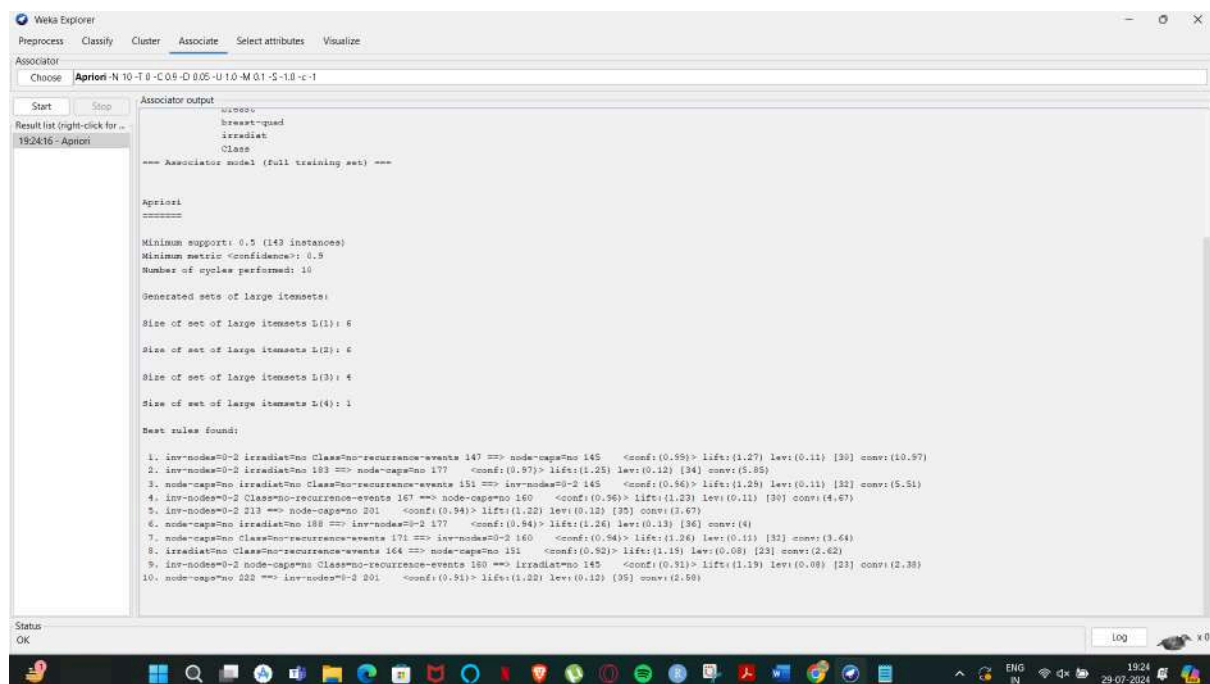




7) Create the dataset using ARFF file format:

a. Find the frequent itemsets and generate association rules on this. Assume that minimum support threshold ($s = 33.33\%$) and minimum confident threshold ($c = 60\%$).

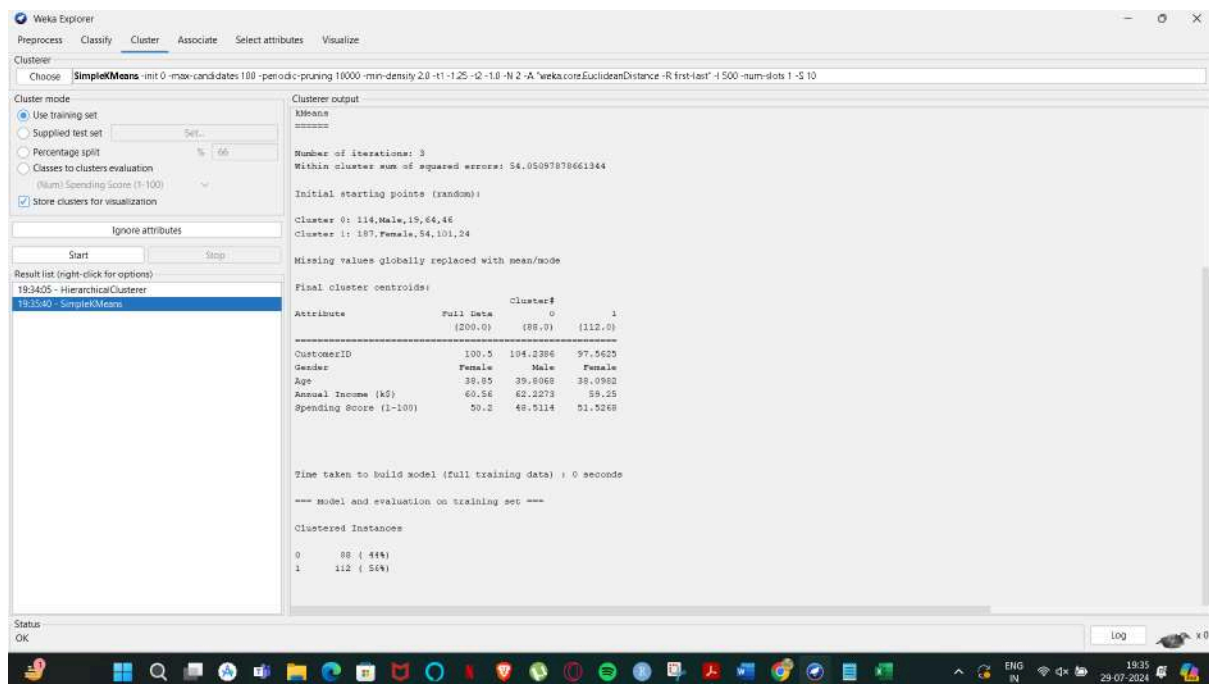
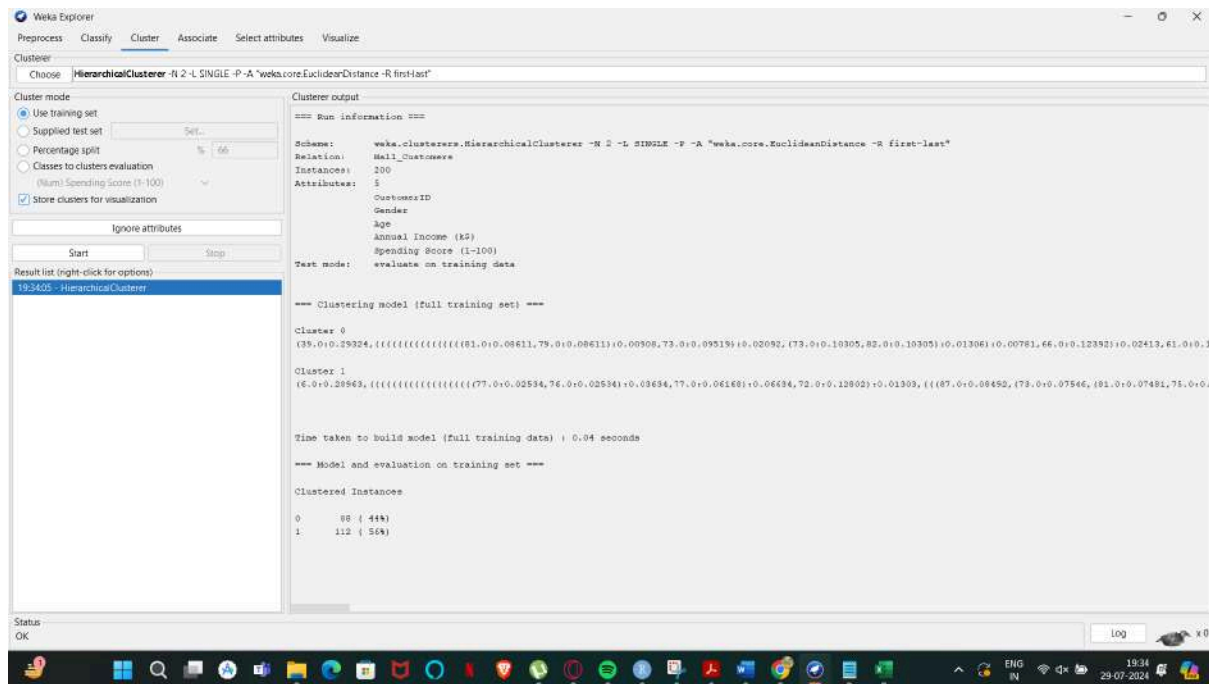
b. List the various rule generated by apriori and FP tree algorithm, mention wheather accepted or rejected.



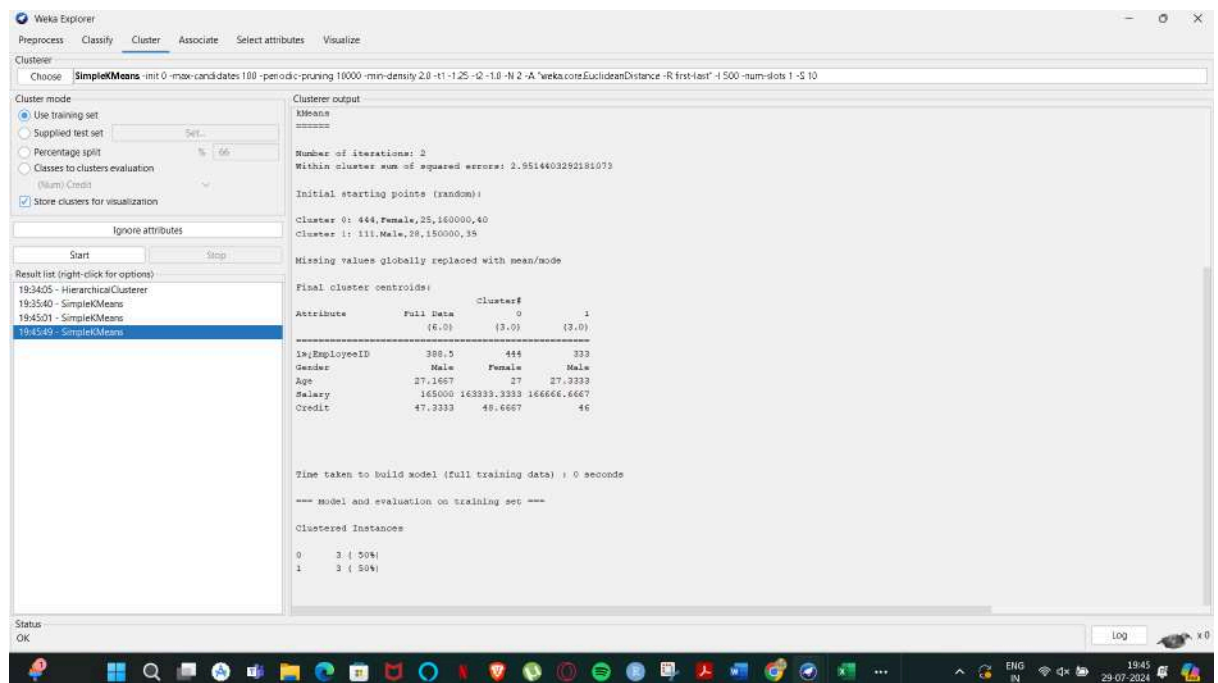
Day_04

- 1) Consider that you are owning a supermarket mall and through membership cards, you have some basic data about your customers like Customer ID, age, gender, annual income and spending score. For the above scenario, the Problem

Statement was You want to understand the customers who can easily converge [Target Customers] so that the data can be given to the marketing team and plan the strategy accordingly. For the above scenario prepare a dataset and perform Clustering Analysis to segment the customers in the Mall. There are clearly Five segments of Customers based on their Annual Income and Spending Score namely *Usual Customers, Priority Customers, Senior Citizen Target Customers, and Young Target Customers.*



- 2) Create the following dataset using CSV file format. To perform cluster analysis using K-Means in WEKA. To change the cluster size and plot the graph and illustrate the visualization of cluster.



- 3) Prediction of categorical data using Naïve Bayes classification through WEKA using any datasets. Compare the Naïve Bayes algorithm with SVM using the summary of results given by the classifiers and plot the graph.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **NaiveBayes**

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 99
 More options...

(Nom) class: **Start** **Stop**

Result list (right-click for options):
 113403 - trees.DecisionStump
 113747 - functions.Logistic
194909 - bayes.NaiveBayes

Classifier output:

```

=====
age      31.2494      37.0808
std. dev. 11.6025      16.2146
weight sum      500      268
precision 1.1765      1.1765

Time taken to build model: 0.01 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      586      76.3021 %
Incorrectly Classified Instances    182      23.6979 %
Kappa statistic      0.4666
Mean absolute error      0.2841
Root mean squared error      0.4169
Relative absolute error      62.5028 %
Root relative squared error      87.4949 %
Total Number of Instances      768

==== Detailed Accuracy By Class ====

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
      -----
0.612  0.156  0.678  0.612  0.643  0.468  0.815  0.671  tested_negative
Weighted Avg.  0.763  0.307  0.755  0.763  0.760  0.468  0.815  0.815  tested_positive

==== Confusion Matrix ====

  a  b  <-- classified as
422 78 |  a = tested_negative
104 164 | b = tested_positive

```

Status: OK

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **RandomForest** P 100 -I 100 -num-dots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 99
 More options...

(Nom) class: **Start** **Stop**

Result list (right-click for options):
 113403 - trees.DecisionStump
 113747 - functions.Logistic
 194909 - bayes.NaiveBayes
195021 - trees.RandomForest

Classifier output:

```

=====
Test model: 10-fold cross validation

==== Classifier model (full training set) ====

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -S 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.22 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      582      75.7813 %
Incorrectly Classified Instances    186      24.2188 %
Kappa statistic      0.4566
Mean absolute error      0.2106
Root mean squared error      0.4031
Relative absolute error      61.3405 %
Root relative squared error      84.5604 %
Total Number of Instances      768

==== Detailed Accuracy By Class ====

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
      -----
0.636  0.188  0.601  0.636  0.618  0.458  0.820  0.686  tested_negative
0.612  0.164  0.667  0.612  0.638  0.458  0.820  0.679  tested_positive
Weighted Avg.  0.758  0.310  0.754  0.758  0.755  0.458  0.820  0.814  tested_positive

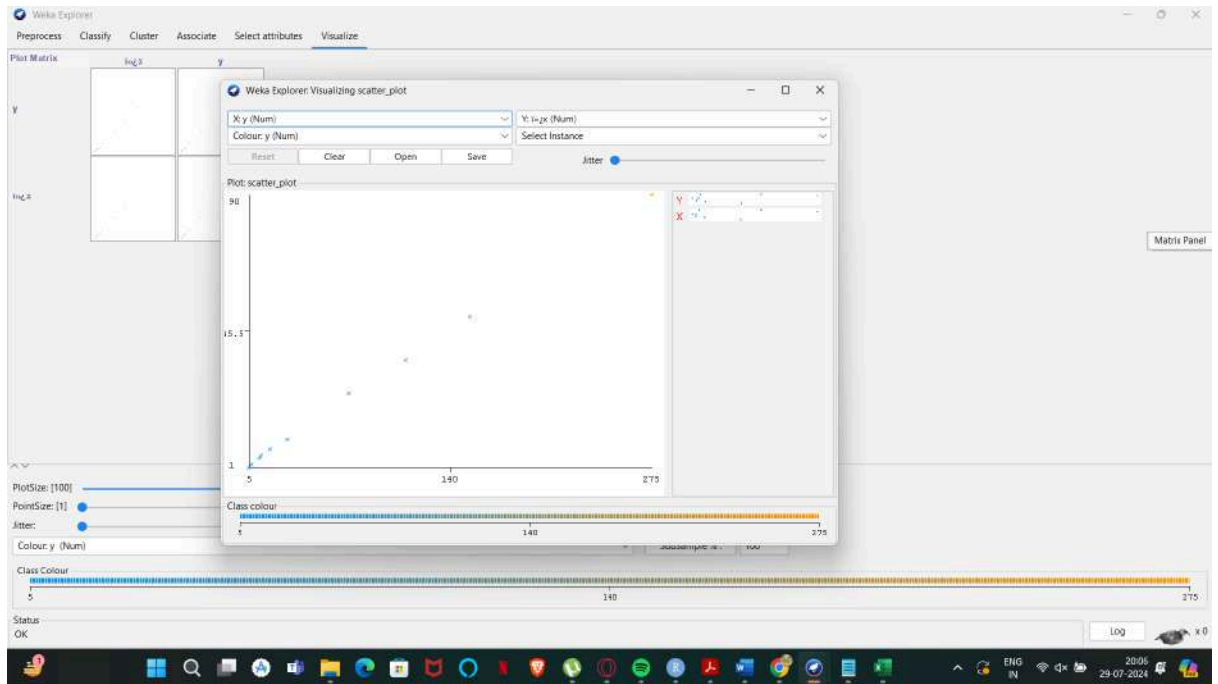
==== Confusion Matrix ====

  a  b  <-- classified as
418  82 |  a = tested_negative
104 164 | b = tested_positive

```

Status: OK

- 4) The following list of persons with vegetarian or not details is given in the table. How will you find out how many of them are vegetarian and how many of them are non-vegetarian? Which type of the person's total count has greater value?



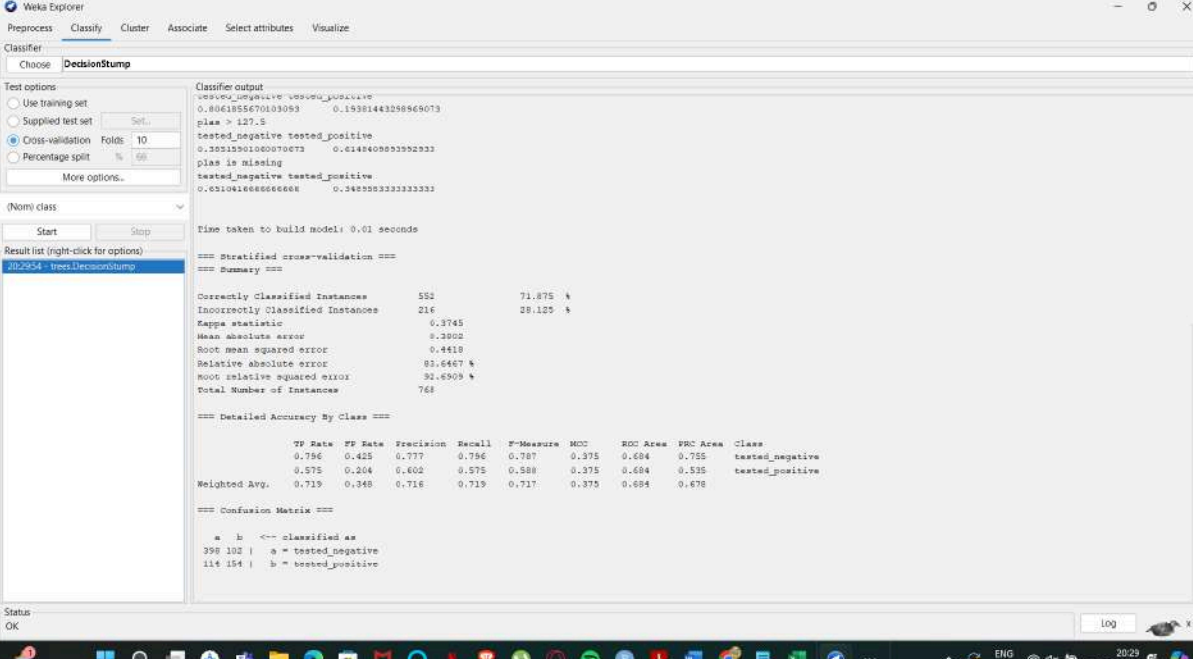
- 6) Generate rules using FP growth algorithm using the given dataset which has the following transactions with items purchased: Consider the values as support=50% and confidence=75%.

The screenshot shows the Weka Explorer interface with the 'Associate' tab selected. The 'FilteredAssociator' is chosen. The 'Result list' shows '20:25:43 - FilteredAssociator'. The 'Associator output' section displays the following information:

```

=== Run information ===
Scheme: weka.associations.FilteredAssociator -F "weka.filters.MultiFilter -F "weka.filters.unsupervised.attribute.ReplaceMissingValues" -S 1" -C -1 -W weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: TRANSACTIONS
Instances: 5
Attributes: 2
TransactionID
ItemPurchased
  
```

7) Prediction of Diabetes Data using Decision tree classifier in WEKA. Compare it with Support Vector Machine classifier. Show the result accuracy and F1 measure calculation .Plot the graph and explain the summary of results.



The screenshot shows the WEKA Explorer interface with the 'Classify' tab selected. The 'DecisionStump' classifier is chosen. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Result list' on the left shows '20:29:54 - trees.DecisionStump' selected. The 'Classifier output' pane displays the following results:

```

Classifier output
=====
0.8061855670103053    0.1938144325866073
plus > 127.5
tested_negative tested_positive
0.38519501060070673    0.6148049893992933
plus is missing
tested_negative tested_positive
0.6310416666666666    0.3689583333333333

Time taken to build model: 0.01 seconds

==== Stratified cross-validation ====
==== Summary ====

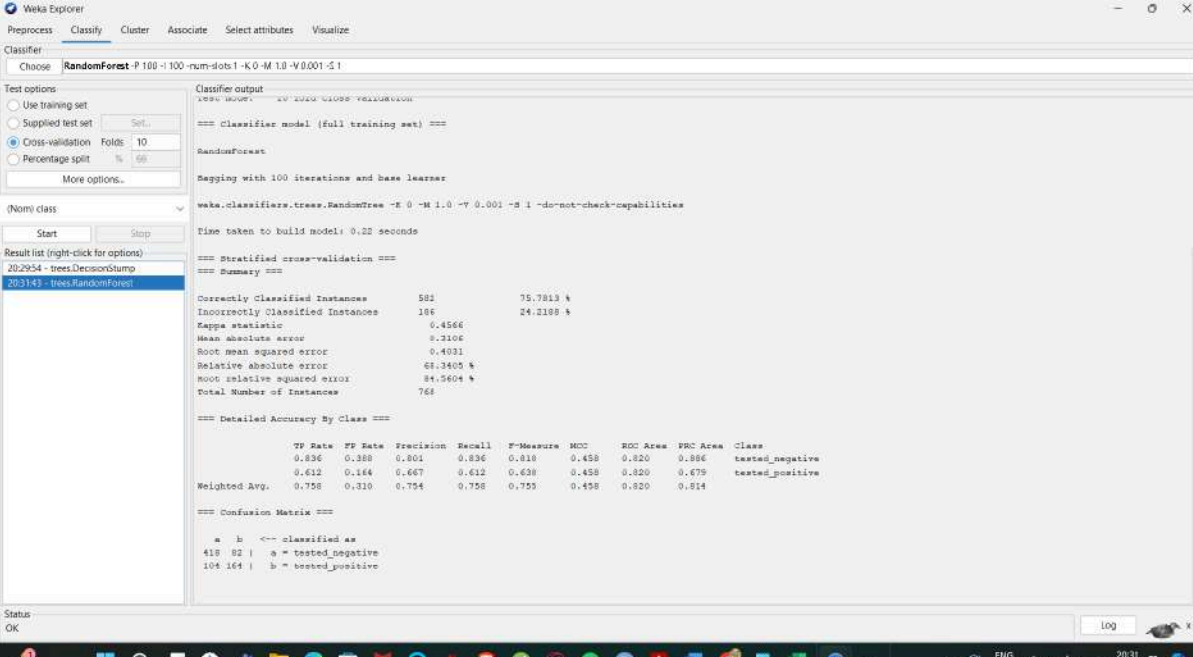
Correctly Classified Instances      552           71.875 %
Incorrectly Classified Instances    216           28.125 %
Kappa statistic                     0.3745
Mean absolute error                 0.2902
Root mean squared error            0.4418
Relative absolute error             81.6467 %
Root relative squared error        92.6509 %
Total Number of Instances          768

==== Detailed Accuracy By Class ====

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
-----
      0.796    0.425    0.777    0.796    0.787    0.375    0.684    0.755    tested_negative
      0.575    0.204    0.602    0.575    0.588    0.375    0.684    0.535    tested_positive
Weighted Avg.    0.719    0.348    0.716    0.719    0.717    0.375    0.684    0.678

==== Confusion Matrix ====

  a  b  <-- classified as
398 102 |  a = tested_negative
114 154 |  b = tested_positive
  
```



The screenshot shows the WEKA Explorer interface with the 'Classify' tab selected. The 'RandomForest' classifier is chosen. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Result list' on the left shows '20:31:43 - trees.RandomForest' selected. The 'Classifier output' pane displays the following results:

```

Classifier output
=====
100 iterations
==== Classifier model (full training set) ====

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -E 0 -M 1.0 -Y 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.22 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      582           75.7813 %
Incorrectly Classified Instances    186           24.2188 %
Kappa statistic                     0.4566
Mean absolute error                 0.2106
Root mean squared error            0.4031
Relative absolute error             61.3405 %
Root relative squared error        84.5604 %
Total Number of Instances          768

==== Detailed Accuracy By Class ====

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
-----
      0.836    0.388    0.801    0.836    0.818    0.458    0.820    0.886    tested_negative
      0.612    0.144    0.667    0.612    0.638    0.458    0.820    0.679    tested_positive
Weighted Avg.    0.758    0.310    0.754    0.758    0.755    0.458    0.820    0.814

==== Confusion Matrix ====

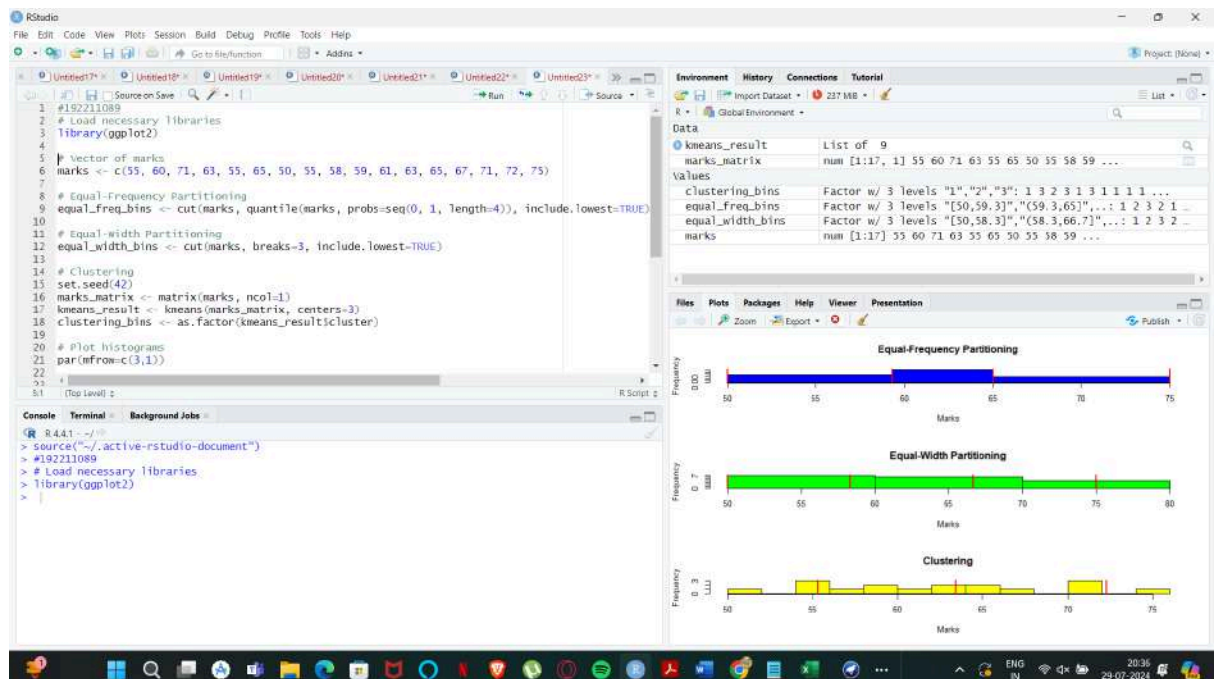
  a  b  <-- classified as
418  82 |  a = tested_negative
108 164 |  b = tested_positive
  
```

- 8) .Implement of the R script using marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75. Partition them into three bins by each of the following methods. Plot the data points using histogram.

(a) equal-frequency (equi-depth) partitioning

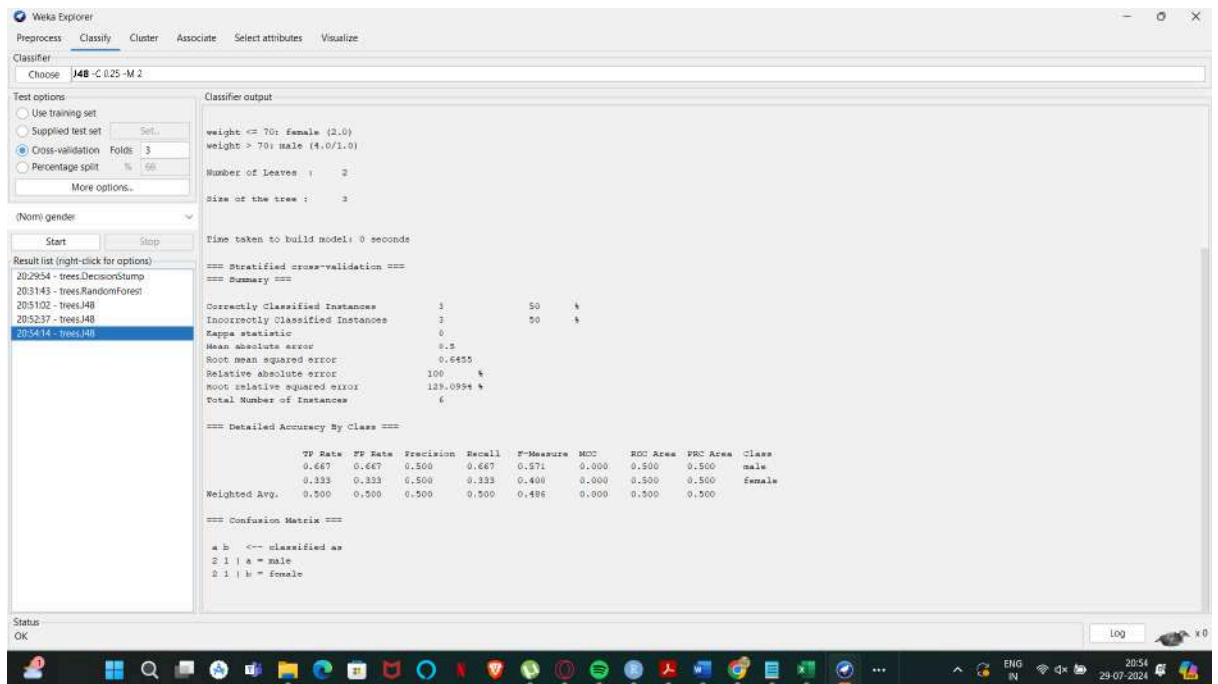
(b) equal-width partitioning

(c) clustering



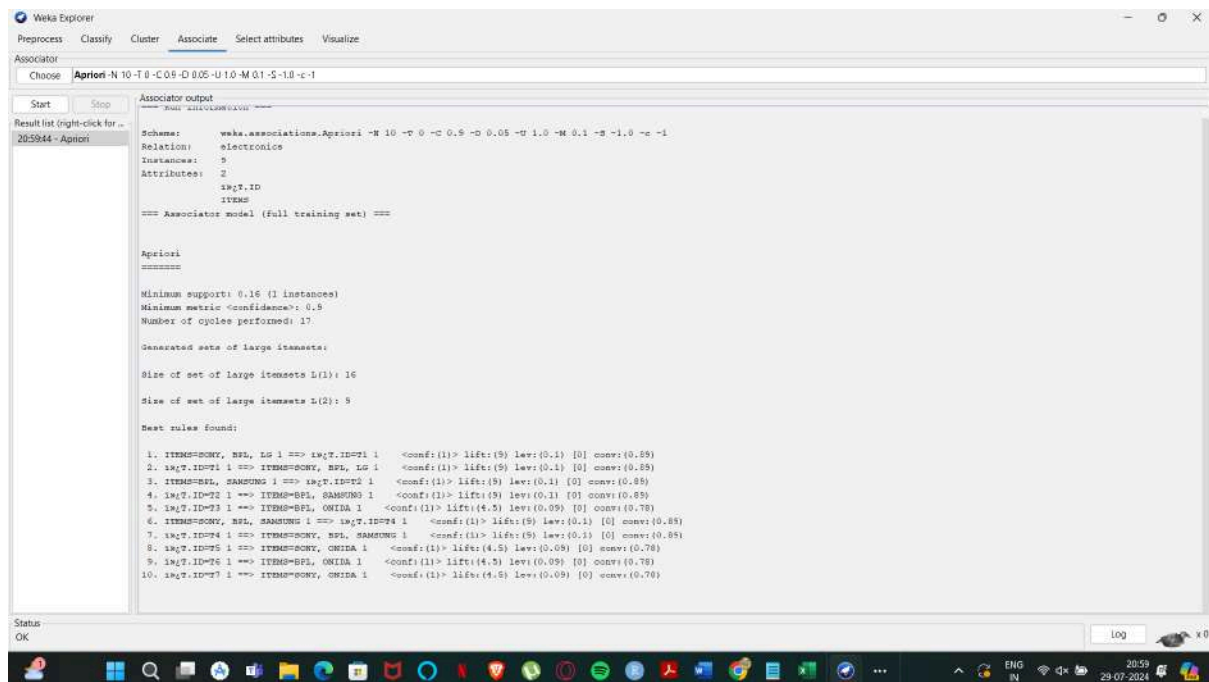
- 9) Consider this Decision tree :

- create the data set for the below tree using ARFF format and calculate accuracy and decision for the same
- Using this decision tree generate the rules based on rule based induction.
- Compare both the algorithms and plot the confusion matrix.



10). Create an ARFF file for the table below and implement for the Apriori Algorithm and FP growth algorithm and compare the rules generated by both the algorithms. Identify the unique rules generated by the above algorithms.

NOTE: Assume Min_sup=2 and confidence= 50%



12) Suppose some car is tested for the AvgSpeed and TotalTime data for 9 randomly selected car with the following result

AvgSpeed d (in kph)	78	81	82	74	83	82	77	80	70
TotalTime e (in mins)	39	37	36	42	35	36	40	38	46

- Calculate the standard deviation of AvgSpeed and TotalTime.
- Calculate the Variance of AvgSpeed and TotalTime for the above dataset.

```

1 #192211089
2 # Define the data
3 AvgSpeed <- c(78, 81, 82, 74, 83, 82, 77, 80, 70)
4 TotalTime <- c(39, 37, 36, 42, 35, 36, 40, 38, 46)
5
6 # Calculate standard deviation
7 sd_AvgSpeed <- sd(AvgSpeed)
8 sd_TotalTime <- sd(TotalTime)
9
10 # Calculate variance
11 var_AvgSpeed <- var(AvgSpeed)
12 var_TotalTime <- var(TotalTime)
13
14 # Print the results
15 cat("Standard Deviation of AvgSpeed:", sd_AvgSpeed, "\n")
16 cat("Standard Deviation of TotalTime:", sd_TotalTime, "\n")
17
18 cat("Variance of AvgSpeed:", var_AvgSpeed, "\n")
19 cat("Variance of TotalTime:", var_TotalTime, "\n")
20

```

Environment

Variable	Value
AvgSpeed	num [1:9] 78 81 82 74 83 82 77 80 70
sd_AvgSpeed	4.30439052338165
sd_TotalTime	3.49205447329283
TotalTime	num [1:9] 39 37 36 42 35 36 40 38 46
var_AvgSpeed	18.5277777777778
var_TotalTime	12.1944444444444

Console

```

R 4.4.1 - ~/...
> source("../.active-rstudio-document")
Standard Deviation of AvgSpeed: 4.304391
Standard Deviation of TotalTime: 3.492054
Variance of AvgSpeed: 18.52778
Variance of TotalTime: 12.19444
> #192211089
> # Define the data
> AvgSpeed <- c(78, 81, 82, 74, 83, 82, 77, 80, 70)
> cat()
> var(AvgSpeed)
[1] 18.52778
> var(TotalTime)
[1] 12.19444
>

```

13) Consider this table

a) TID items bought

b) T100 {M, O, N, K, E, Y}

c) T200 {D, O, N, K, E, Y}

d) T300 {M, A, K, E}

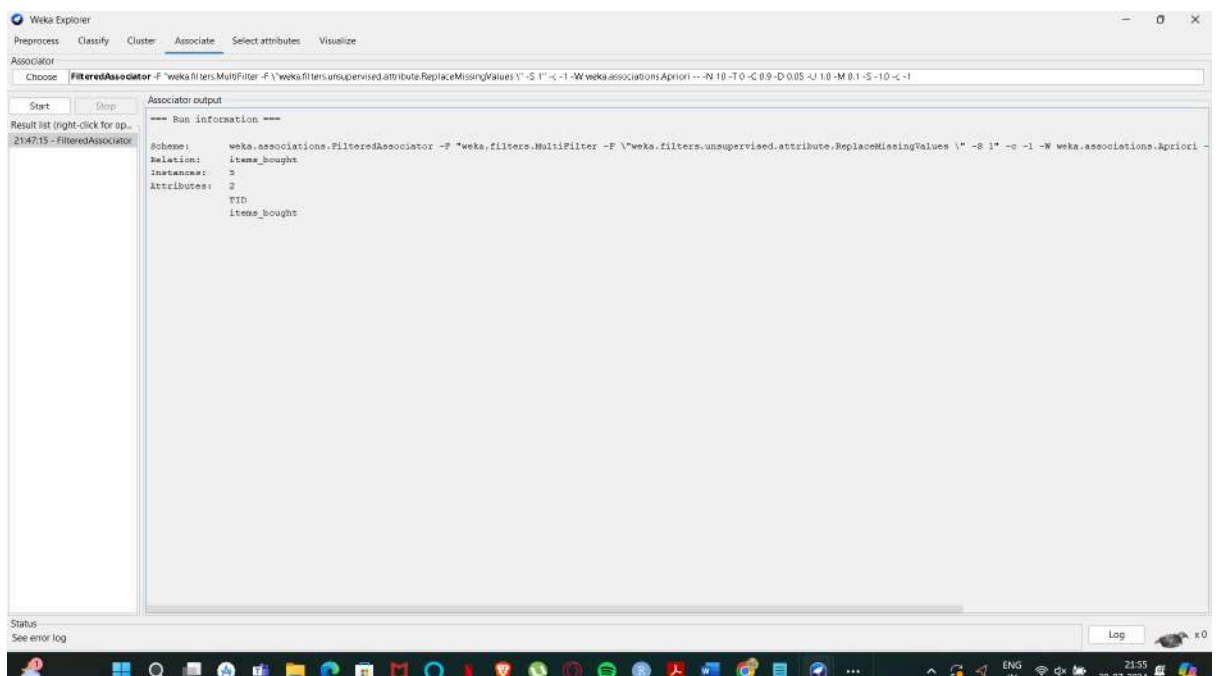
e) T400 {M, U, C, K, Y}

f) T500 {C, O, O, K, I, E}

g) (a) Find all frequent item set using Apriori and FP-growth, respectively. Compare the efficiency of the two mining processes.

h) (b) List all of the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and $item_i$ denotes variables representing items (e.g., “A”, “B”, etc.):

i) $\forall x \in \text{transaction}, \text{buys}(X, \text{item1}) \wedge \text{buys}(X, \text{item2}) \Rightarrow \text{buys}(X, \text{item3})$



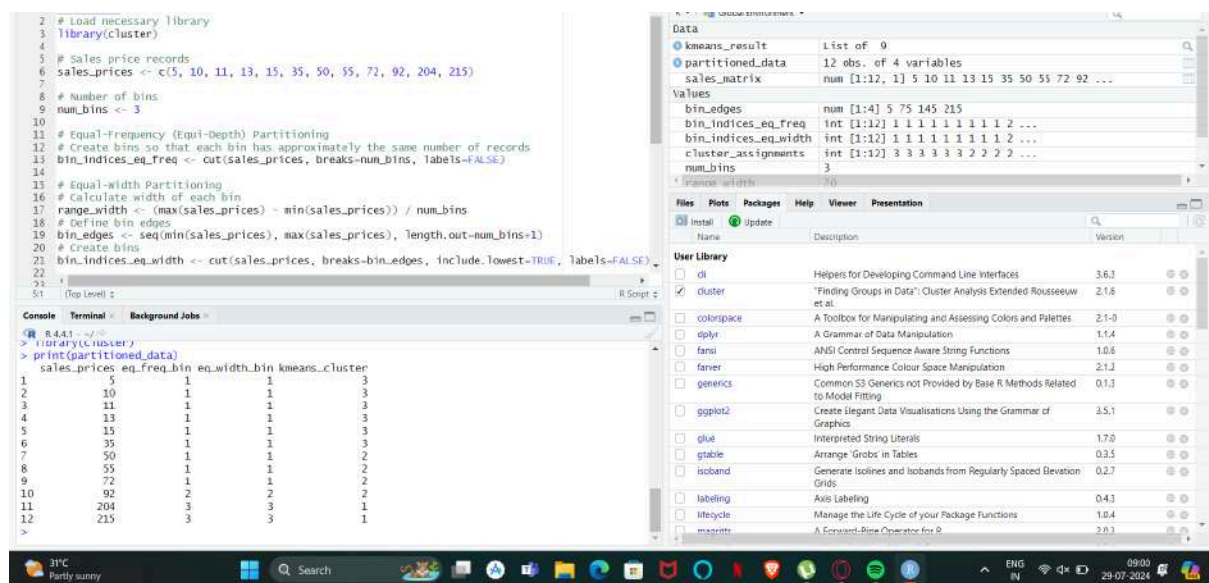
QUESTION_BANK :

1) Implement of the R script using a group of 12 sales price records has been sorted as follows: 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215. Partition them into three bins by each of the following methods.

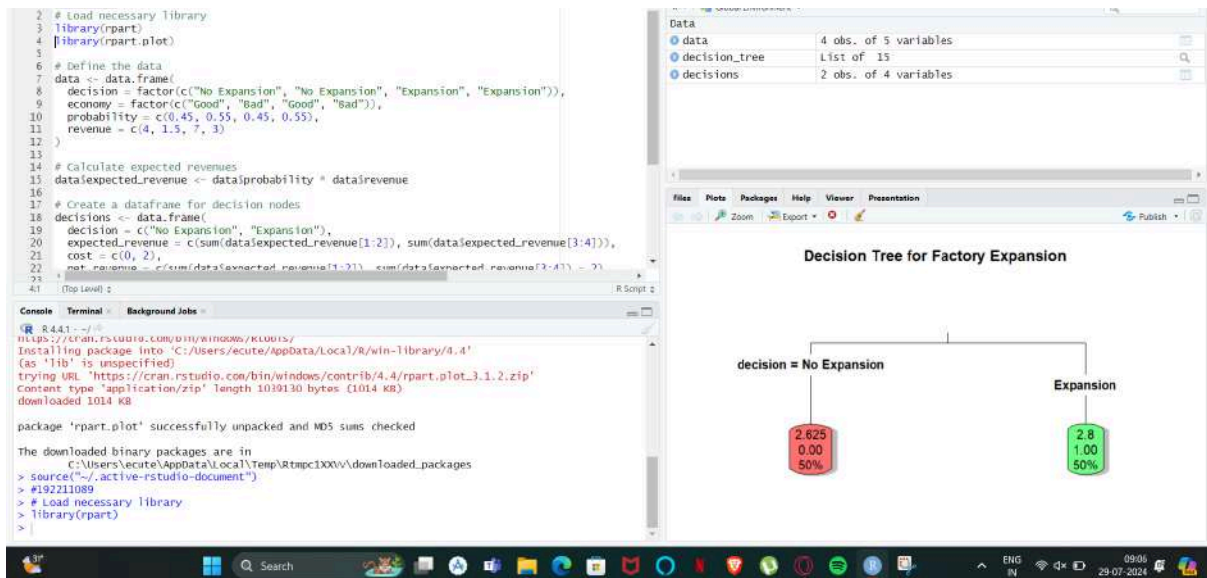
(a) equal-frequency (equi depth) partitioning

(b) equal-width partitioning

(c) clustering



2) A gadget factory has been quite successful for the past 10 years and Ms.Marry, the manager of the company wondering whether to expand the factory this year or not. The cost to expand factory is \$2M. With no expansion, expected revenue is \$4M if the economy stays good; while only \$1.5M if the economy is bad. If manager expands the factory, expected to receive \$7M. if economy is good and \$3M if economy is bad. Assume that there is a 45% chance of a good economy and a 55% chance of a bad economy. Draw a Decision Tree showing these choices.



3) Implement using WEKA for the given Apply Apriori Algorithm for given database below Assume Min_sup=2 TID Items

1

Bread, Peanuts, Milk, Fruit, Jam

2

Bread, Jam, Soda, Chips, Milk, Fruit

3

Steak, Jam, Soda, Chips, Bread

4

Jam, Soda, Peanuts, Milk, Fruit

5

Jam, Soda, Chips, Milk, Bread

6

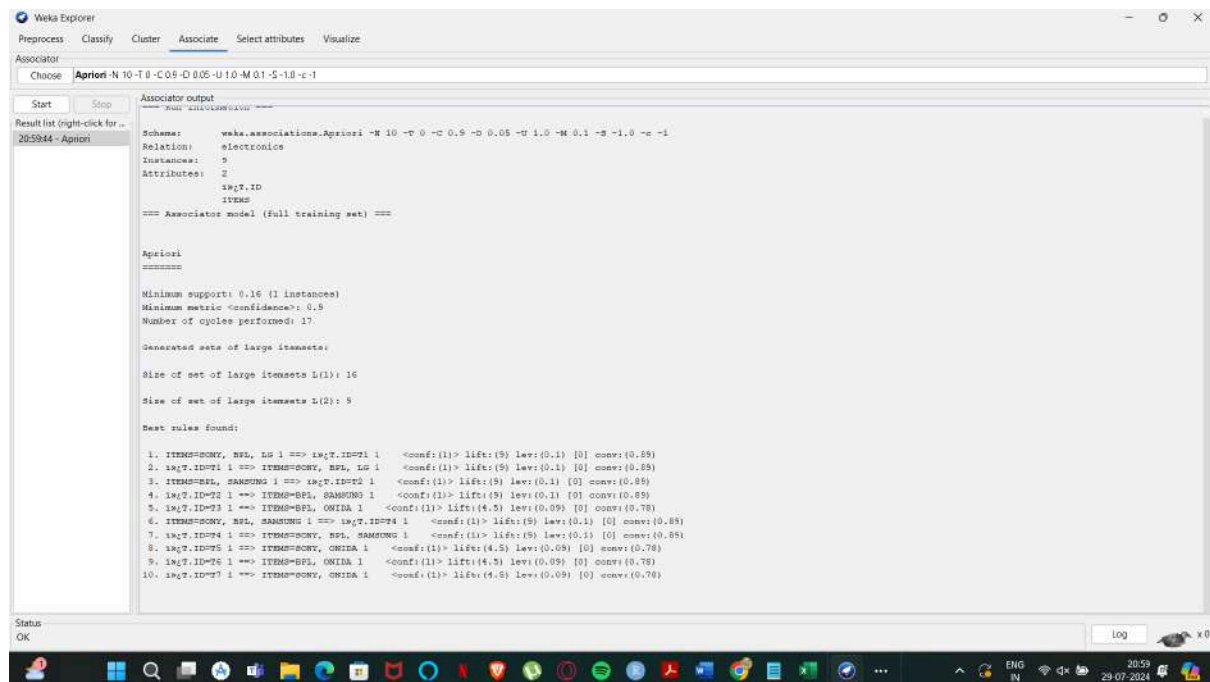
Fruit, Soda, Chips, Milk

7

Fruit, Soda, Peanuts, Milk

8

Fruit, Peanuts, Cheese, Yogurt

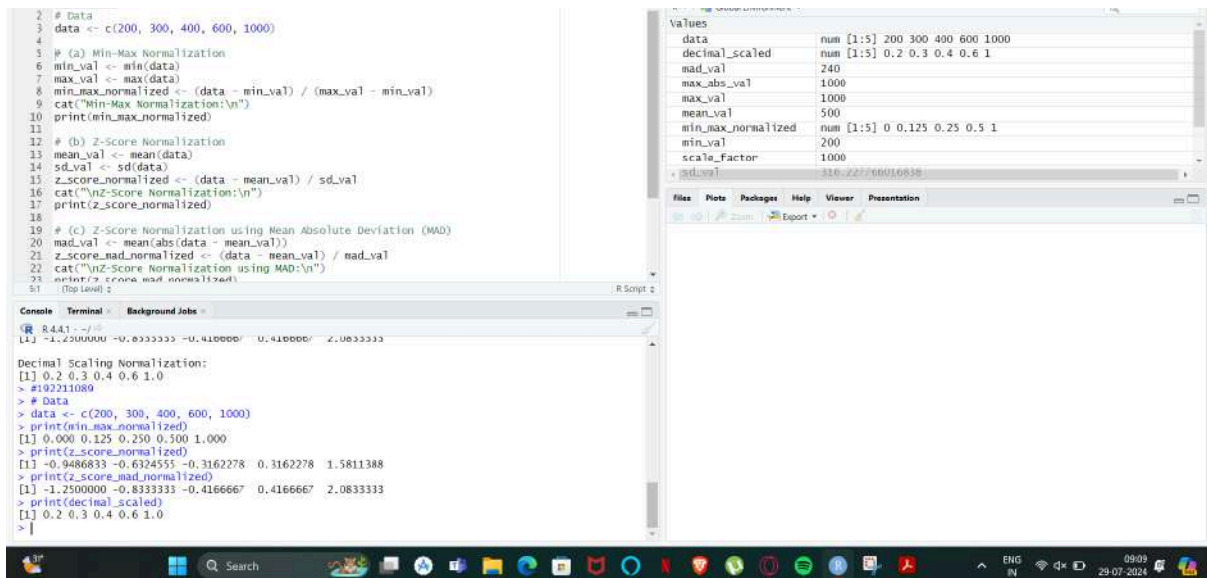


4) Use following group of data: 200, 300, 400, 600, 1000

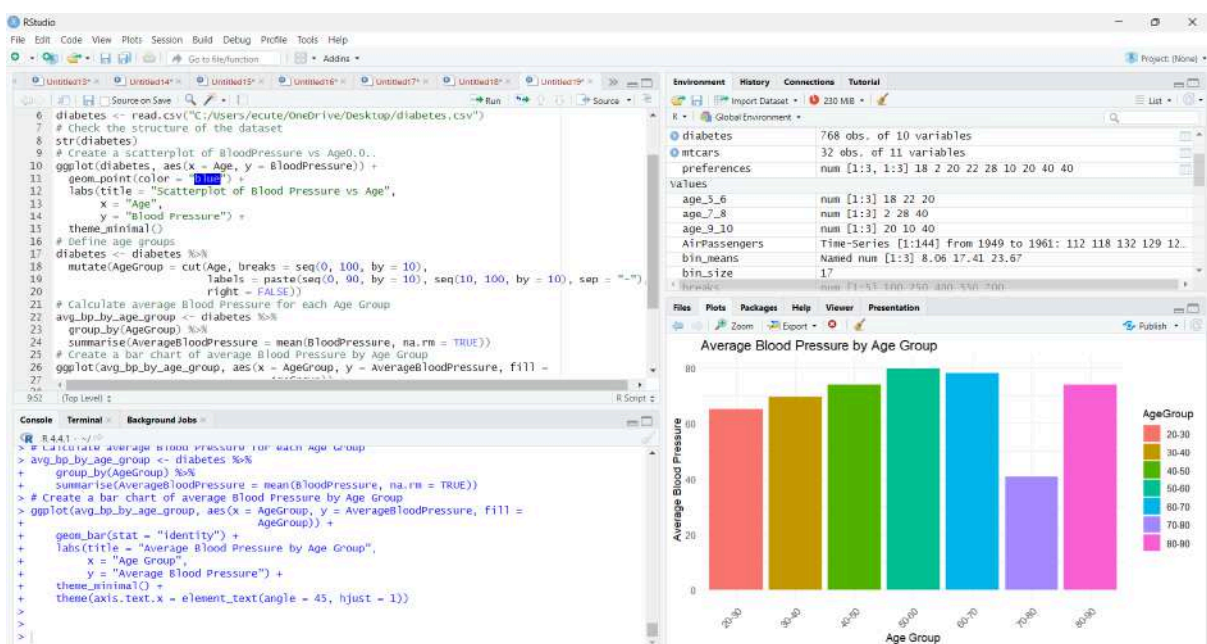
(a) min-max normalization by setting min = 0 and max = 1 (b)

(b) z-score normalization

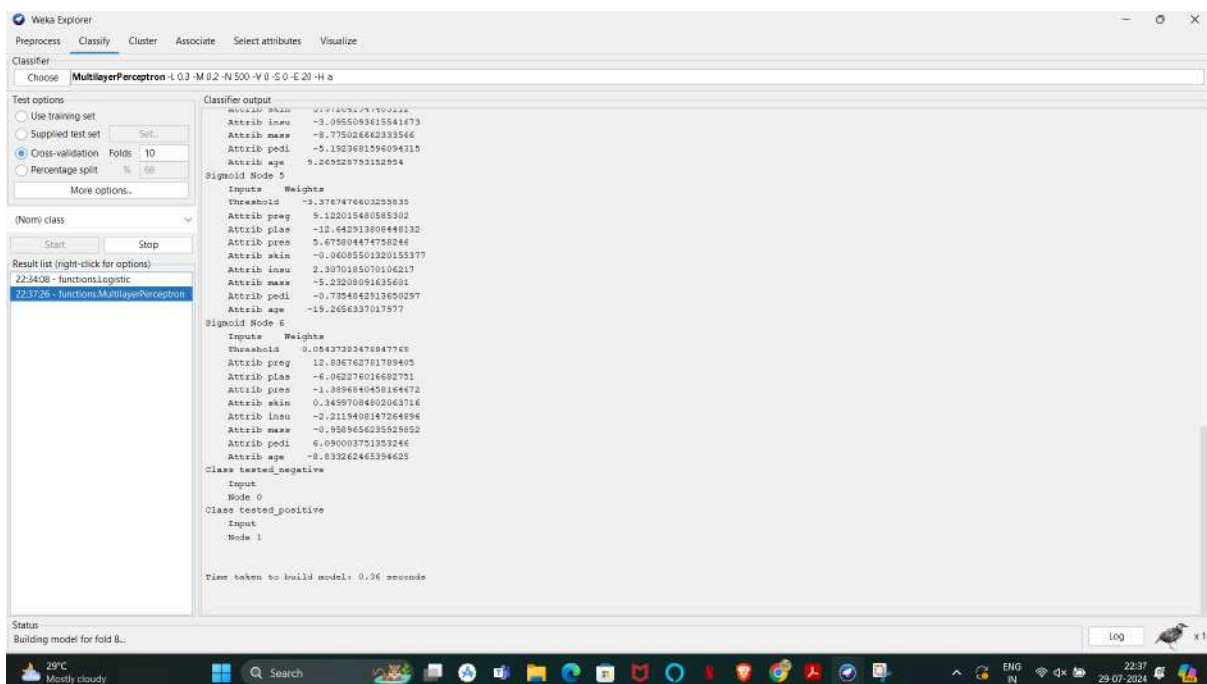
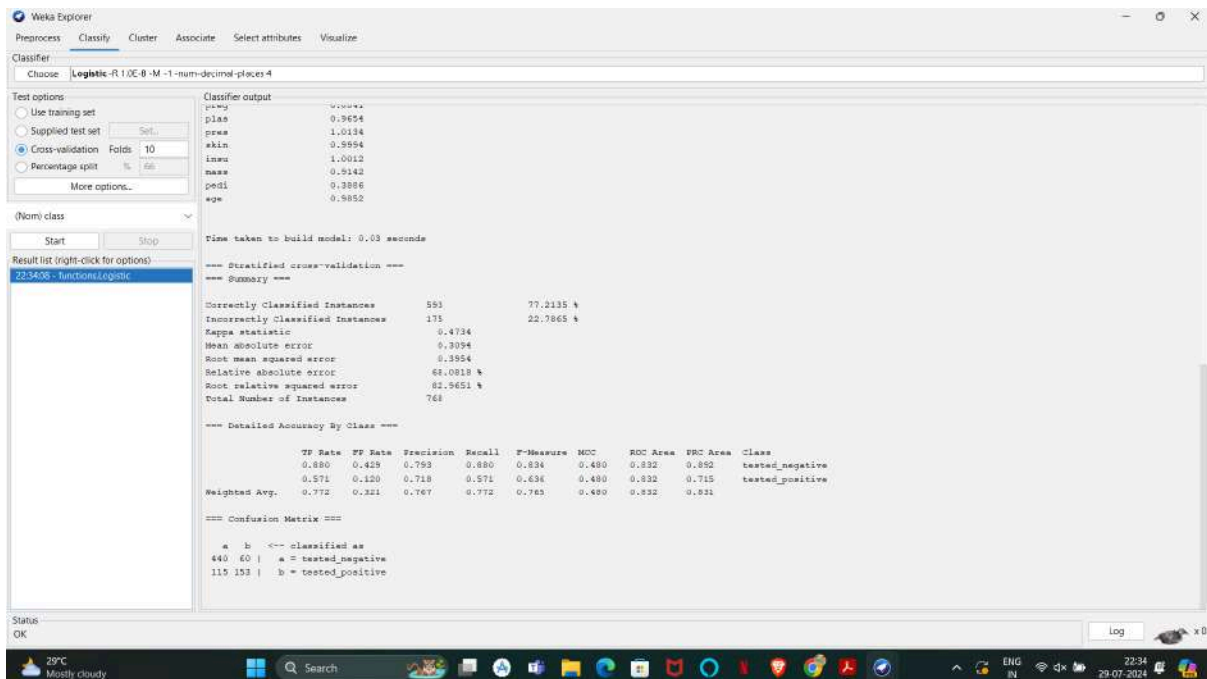
(c) (c) z-score normalization using the mean absolute deviation instead of standard deviation (d) normalization by decimal scaling



5) Implement using R language in which age group of people are affected by blood pressure based on the diabetes dataset show it using scatterplot and bar chart (that is Blood Pressure vs Age using dataset “diabetes.csv”)



6) Analysis the dataset “diabetes. csv” how the diabetes trend is for different age people, using linear regression and multiple regression.



7) Implement using WEKA for the given Suppose a database has five transactions. Let min sup= 50%(2) and min con f = 80%.

Transactions Items

T1 (M, O, N, K, E, Y)

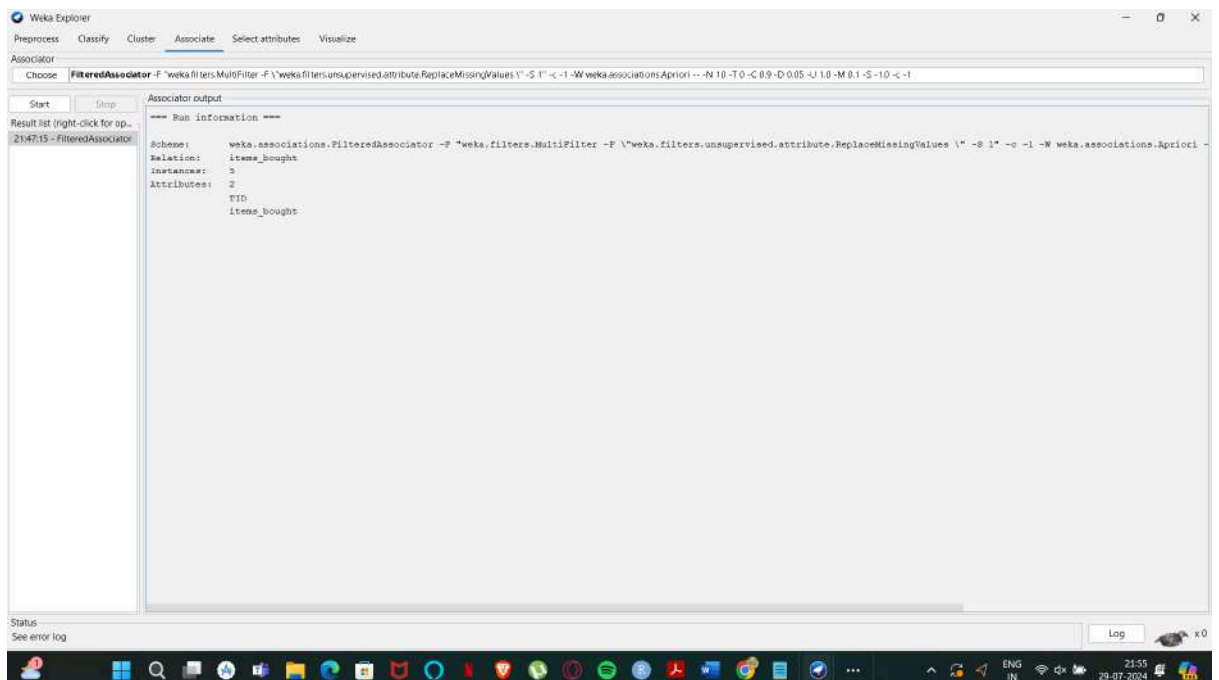
T2 (D, O, N, K, E, Y)

T3 (M, A, K, E)

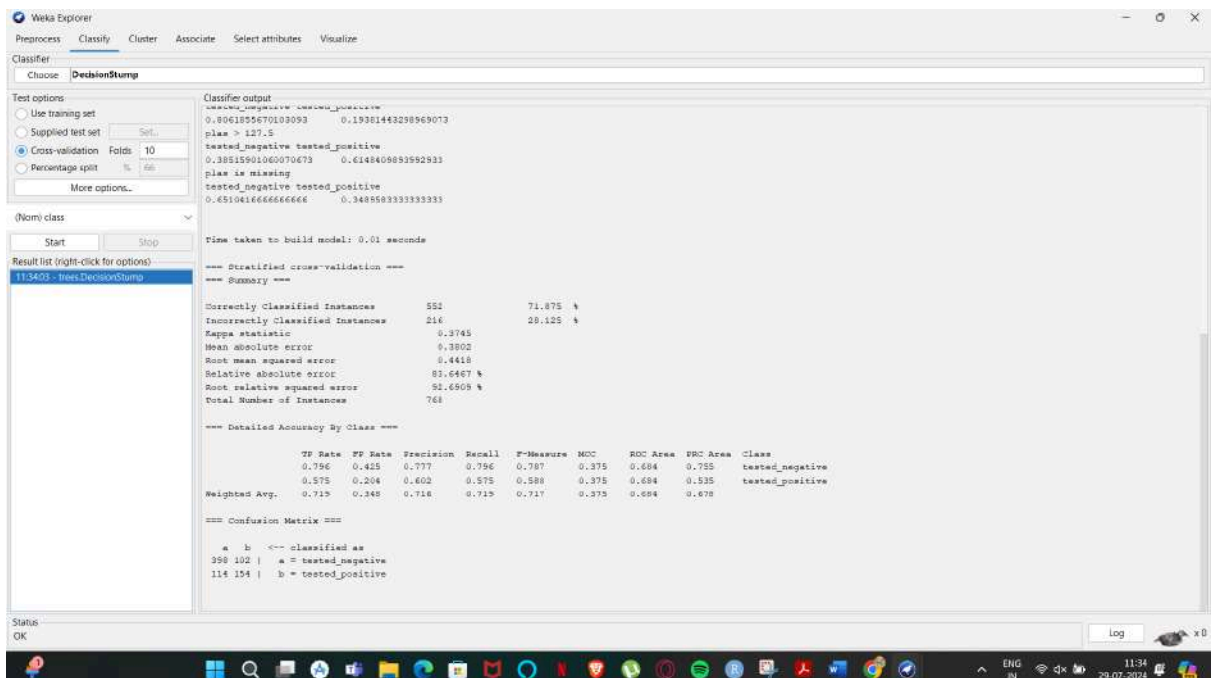
T4 (M, U, C, K, Y)

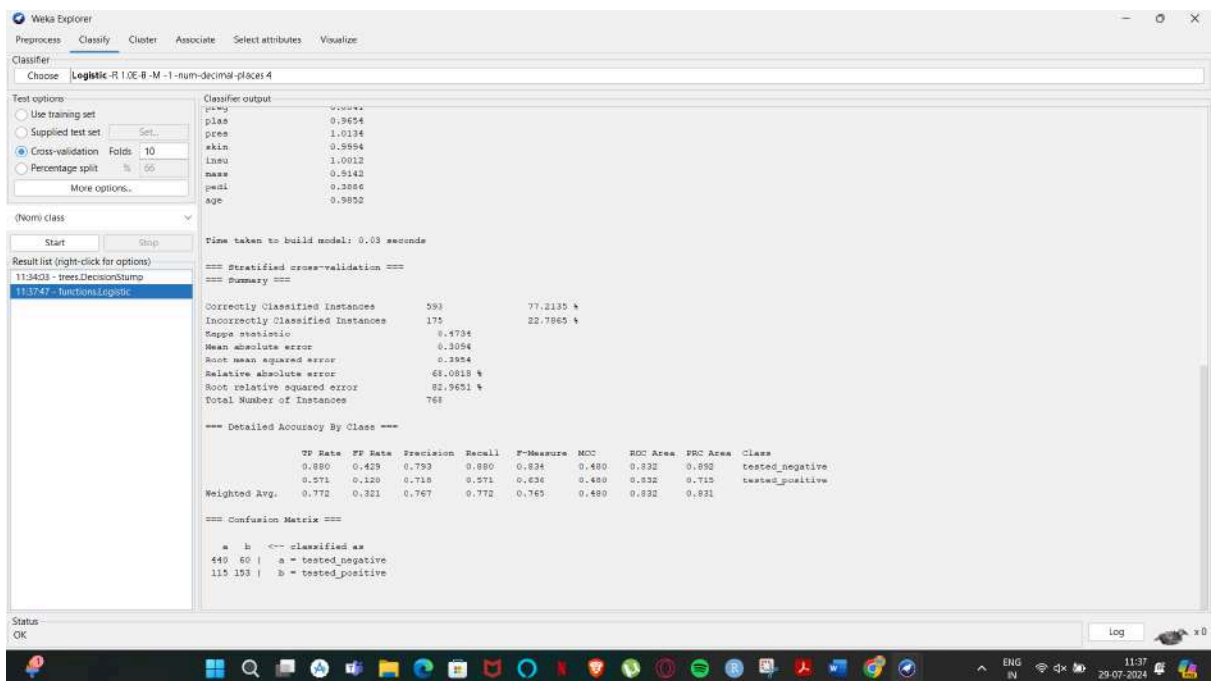
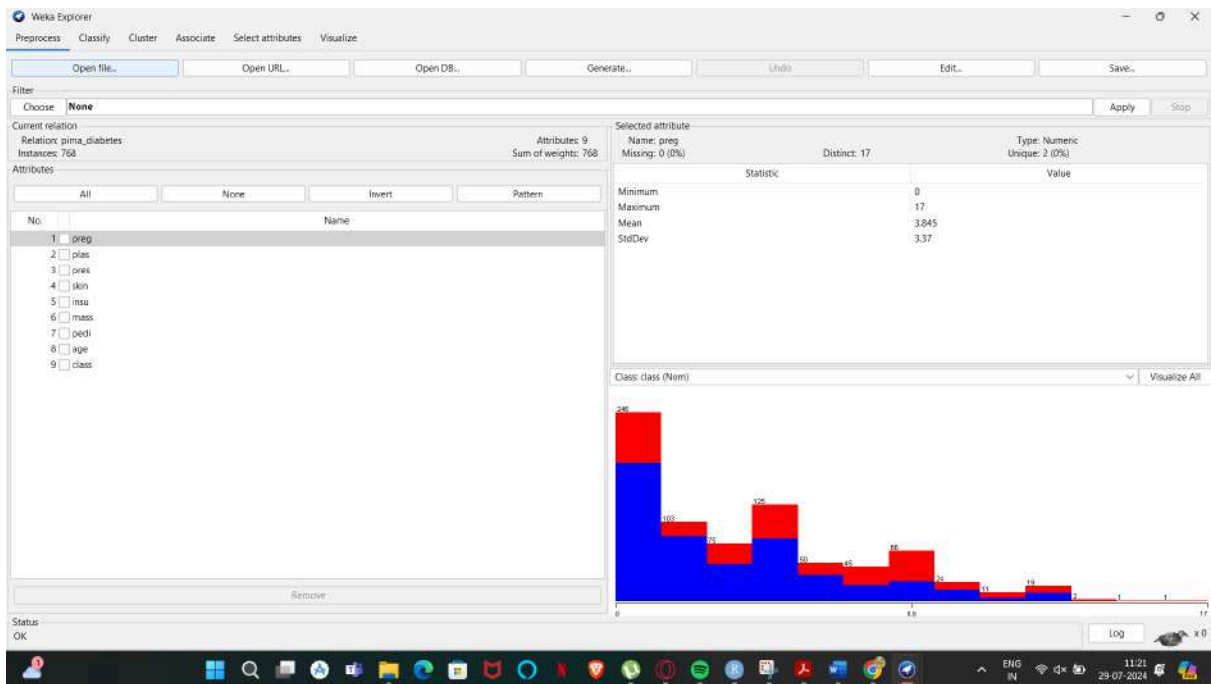
T5 (C,O, O, K, I, E)

- Find all frequent item sets using Apriori algorithm
- Also draw FP-Growth Tree



8) Prediction of Categorical Data using Decision Tree Algorithm through WEKA using any datasets. a) Tree b) Preprocess c) Logistic





9) Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

The screenshot displays the R Studio environment. The script editor on the left contains the following R code:

```
2 #mean, median, mode, quartile
3 age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
4 mean(age)
5 median(age)
6 mode_age<-names(table(age))[table(age)==max(table(age))]
7 mode_age
8 range(age)
9 quantile(age,.25)
10 quantile(age,.75)
11
```

The console on the bottom left shows the execution results:

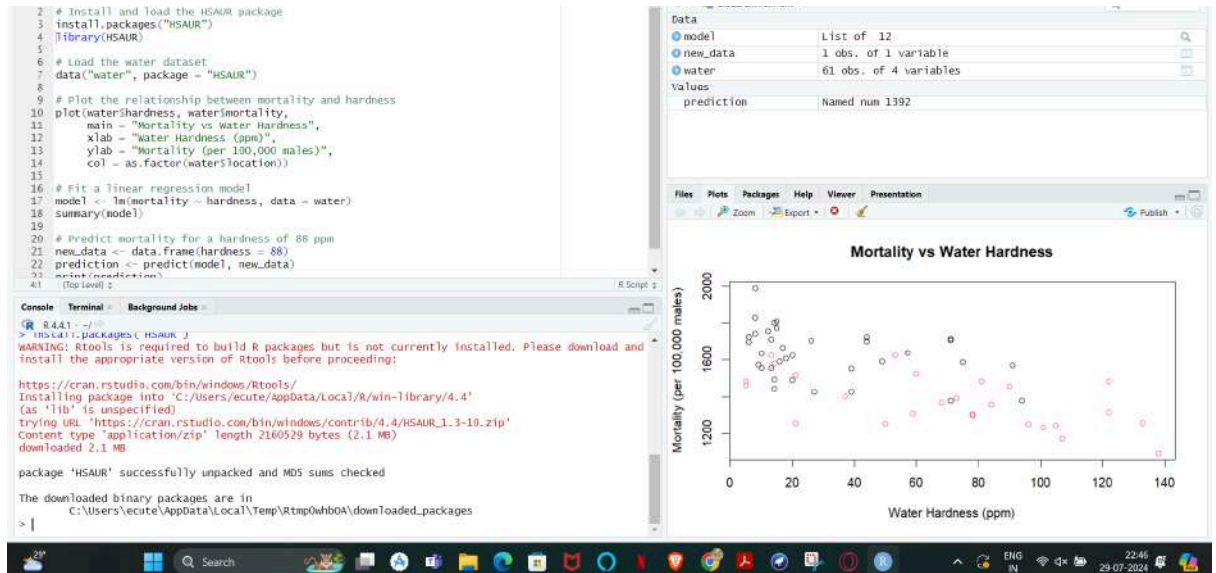
```
> #mean, median, mode, quartile
> age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
> mean(age)
[1] 29.96296
> median(age)
[1] 25
> mode_age
[1] "25" "35"
> range(age)
[1] 13 70
> quantile(age,.25)
25%
20.5
> quantile(age,.75)
75%
35
```

On the right, the 'Values' pane shows the data for 'age' and 'mode_age':

Variable	Value
age	num [1:27] 13 15 16 16 19 20 20 21 22 22 ...
mode_age	chr [1:2] "25" "35"

Below the 'Values' pane is a file explorer showing the project's file structure, including folders like '26_09_23_factorial_function_3.cpp' and '26_09_23_factorial_function_3.exe'.

10) Download the Dataset "water" From R dataset Link. Find out whether there is a linear relation between attributes "mortality" and "hardness" by plot function. Fit the Data into the Linear Regression model. Predict the mortality for the hardness=88.



11) Create the dataset using ARFF file format:

a. Find the frequent itemsets and generate association rules on this. Assume that minimum support threshold ($s = 33.33\%$) and minimum confident threshold ($c = 60\%$).

b. List the various rule generated by apriori and FP tree algorithm, mention whether accepted or rejected.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Associate

Choose **Apriori** - N 10 - T 0 - C 0.5 - D 0.05 - U 1.0 - M 0.1 - S 1.0 - c 1

Start Stop

Result list (right-click for...)

192416 - Apriori

Associate output

```

=====
Apriori
=====

Minimum support: 0.5 (143 instances)
Minimum metric (confidence): 0.5
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6
Size of set of large itemsets L(2): 6
Size of set of large itemsets L(3): 4
Size of set of large itemsets L(4): 1

Best rules found:

1. inv-nodes=0-2 irradiat=no class=no-recurrence-events 147 ==> node-caps=no 145 <conf:(0.59)> lift:(1.27) lev:(0.11) [30] conv:(10.97)
2. inv-nodes=0-2 irradiat=no 183 ==> node-caps=no 177 <conf:(0.97)> lift:(1.25) lev:(0.12) [34] conv:(5.85)
3. node-caps=no irradiat=no class=no-recurrence-events 151 ==> inv-nodes=0-2 145 <conf:(0.96)> lift:(1.28) lev:(0.11) [32] conv:(5.51)
4. inv-nodes=0-2 class=no-recurrence-events 167 ==> node-caps=no 160 <conf:(0.96)> lift:(1.23) lev:(0.11) [30] conv:(4.67)
5. inv-nodes=0-2 213 ==> node-caps=no 201 <conf:(0.94)> lift:(1.22) lev:(0.12) [35] conv:(3.67)
6. node-caps=no irradiat=no 188 ==> inv-nodes=0-2 177 <conf:(0.94)> lift:(1.26) lev:(0.13) [36] conv:(4)
7. node-caps=no class=no-recurrence-events 171 ==> inv-nodes=0-2 160 <conf:(0.94)> lift:(1.26) lev:(0.11) [32] conv:(3.64)
8. irradiat=no class=no-recurrence-events 164 ==> node-caps=no 151 <conf:(0.92)> lift:(1.15) lev:(0.08) [23] conv:(2.62)
9. inv-nodes=0-2 node-caps=no class=no-recurrence-events 180 ==> irradiat=no 145 <conf:(0.91)> lift:(1.19) lev:(0.09) [23] conv:(2.38)
10. node-caps=no 222 ==> inv-nodes=0-2 201 <conf:(0.81)> lift:(1.20) lev:(0.12) [35] conv:(2.50)

```

Status OK

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Associate

Choose **Filtered Apriori** - F "weka.filters.Unsupervised.attribute.ReplaceMissingValues" - S "1" - c 1 - W weka.associations.Apriori -- N 10 - T 0 - C 0.5 - D 0.05 - U 1.0 - M 0.1 - S 1.0 - c 1

Start Stop

Result list (right-click for op...)

192416 - Apriori

192439 - Filtered Apriori

Associate output

```

#data

Associate Model

Apriori
=====

Minimum support: 0.35 (100 instances)
Minimum metric (confidence): 0.5
Number of cycles performed: 13

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10
Size of set of large itemsets L(2): 20
Size of set of large itemsets L(3): 8
Size of set of large itemsets L(4): 2

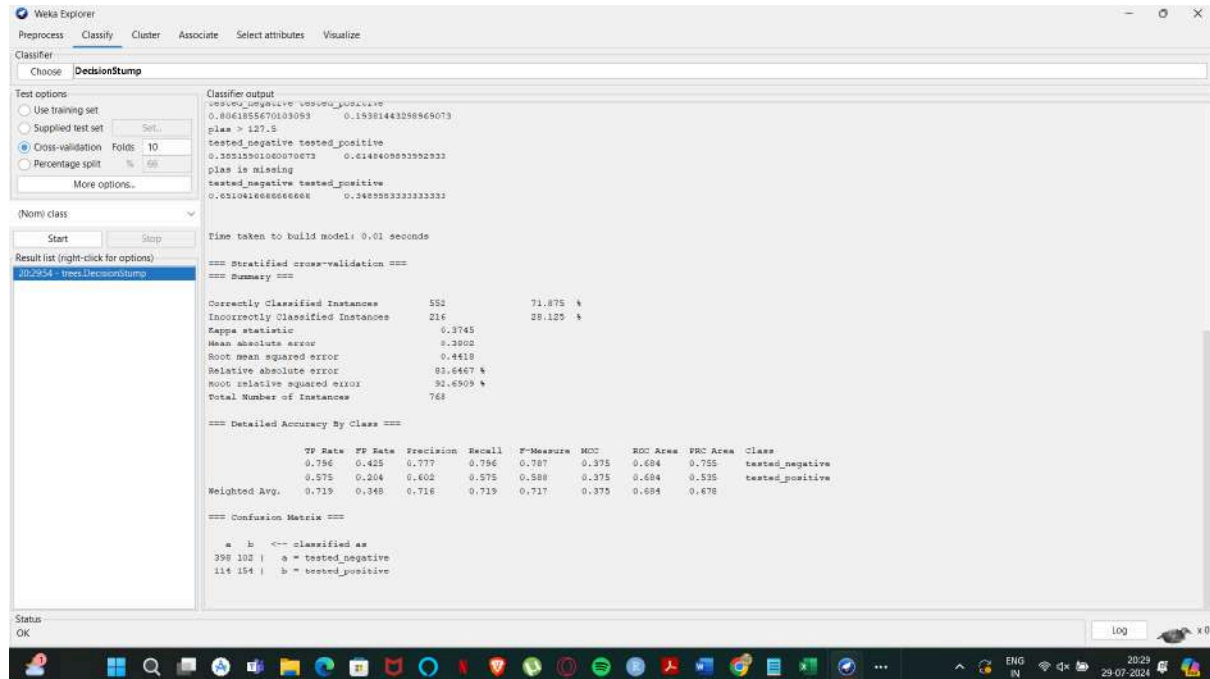
Best rules found:

1. inv-nodes=0-2 breast=left irradiat=no 101 ==> node-caps=no 100 <conf:(0.98)> lift:(1.13) lev:(0.07) [18] conv:(6.88)
2. inv-nodes=0-2 irradiat=no class=no-recurrence-events 147 ==> node-caps=no 145 <conf:(0.88)> lift:(1.23) lev:(0.06) [28] conv:(8.58)
3. inv-nodes=0-2 breast=left 115 ==> node-caps=no 113 <conf:(0.98)> lift:(1.22) lev:(0.07) [29] conv:(7.51)
4. inv-nodes=0-2 irradiat=no 183 ==> node-caps=no 175 <conf:(0.98)> lift:(1.22) lev:(0.11) [31] conv:(7.17)
5. inv-nodes=0-2 class=no-recurrence-events 167 ==> node-caps=no 161 <conf:(0.96)> lift:(1.21) lev:(0.09) [26] conv:(4.67)
6. node-caps=no breast=left irradiat=no 104 ==> inv-nodes=0-2 100 <conf:(0.96)> lift:(1.15) lev:(0.04) [22] conv:(5.31)
7. node-caps=no irradiat=no class=no-recurrence-events 151 ==> inv-nodes=0-2 145 <conf:(0.96)> lift:(1.29) lev:(0.11) [32] conv:(5.51)
8. inv-nodes=0-2 213 ==> node-caps=no 204 <conf:(0.96)> lift:(1.19) lev:(0.11) [32] conv:(4.17)
9. nonrecurrence=no inv-nodes=0-2 112 ==> node-caps=no 106 <conf:(0.95)> lift:(1.18) lev:(0.04) [15] conv:(3.13)
10. node-caps=no irradiat=no 190 ==> inv-nodes=0-2 175 <conf:(0.94)> lift:(1.26) lev:(0.12) [37] conv:(4.04)

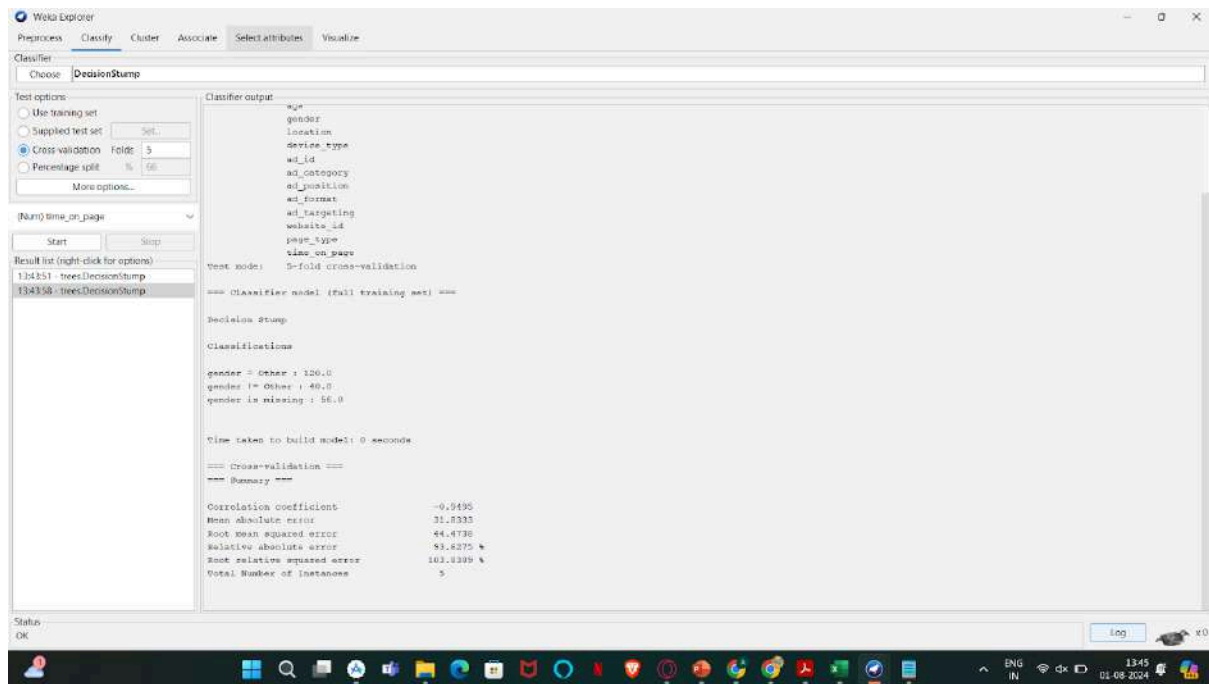
```

Status OK

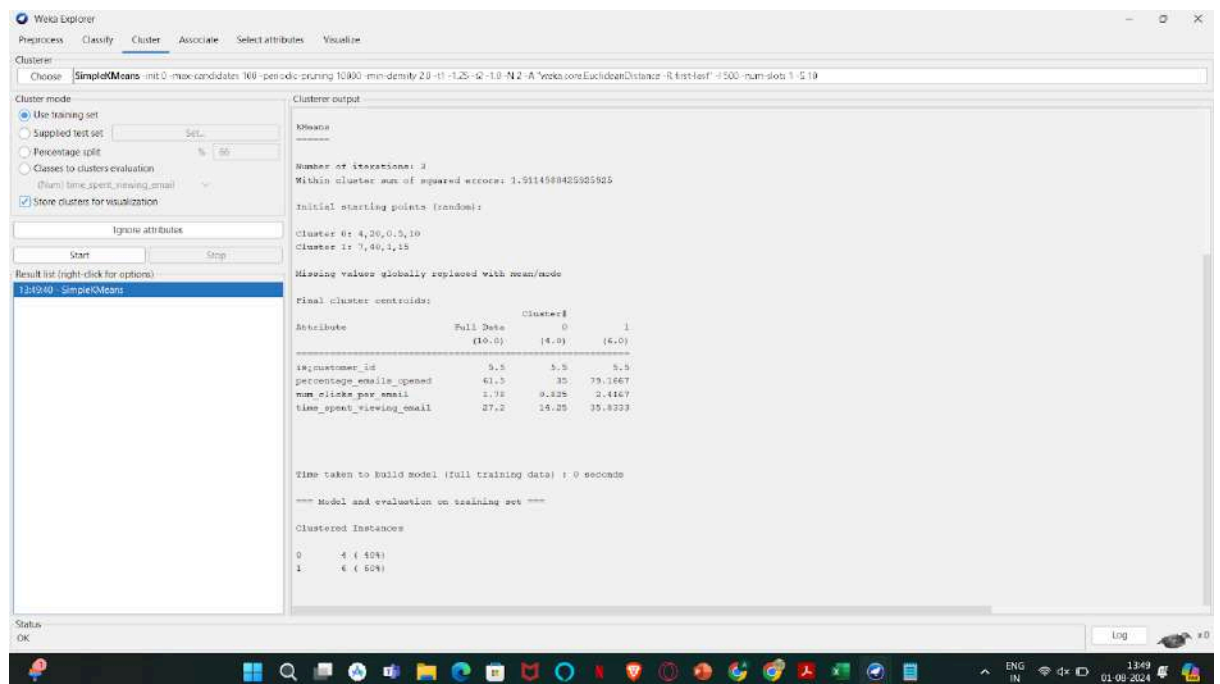
12) Prediction of Categorical Data using Rule base classification and decision tree classification through WEKA using any datasets. Compare the accuracy using two algorithm and plot the graph



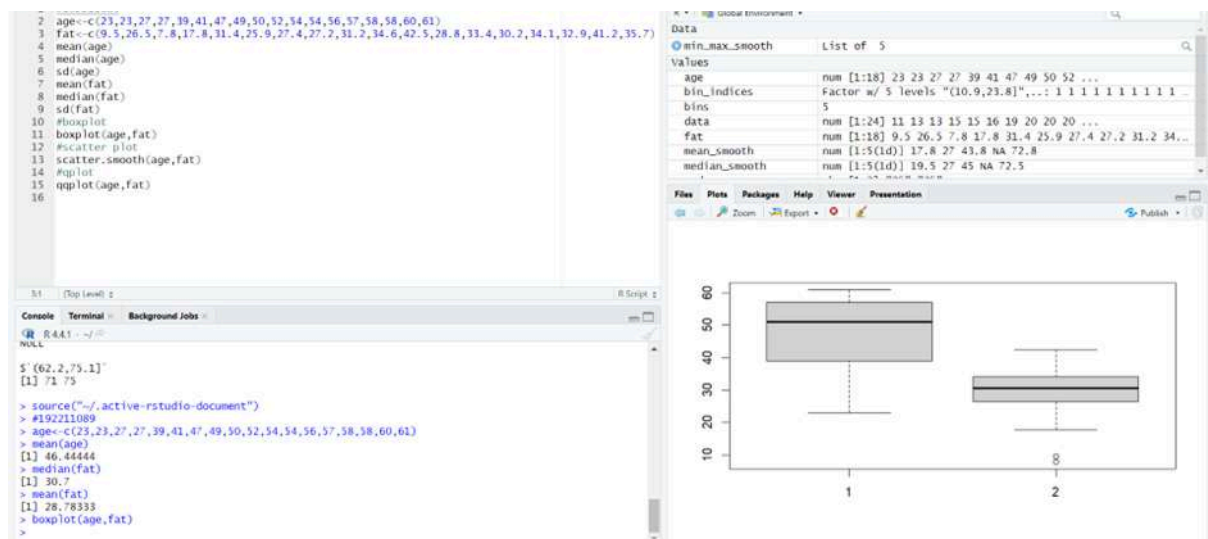
13) Imagine that you have selected data from the All Electronics data warehouse for analysis. The data set will be huge! The following data are a list of All Electronics prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 28, 28, 30, 30, 30. (i) Partition the dataset using an equal-frequency partitioning method with bin equal to 3 (ii) apply data smoothing using bin means and bin boundary. (iii) Plot Histogram for the above frequency division



16) Consider that Many businesses use cluster analysis to identify consumers who are similar to each other so they can tailor their emails sent to consumers in such a way that maximizes their revenue. Consider a business may collect the following information about consumers: Percentage of emails opened Number of clicks per email Time spent viewing email Using these metrics, a business can perform various cluster analyses to identify consumers who use email in similar ways and tailor the types of emails and frequency of emails they send to different clusters of customers. Compare the performance of the applied clustering algorithm.



17) Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:



18) Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

```

2 v<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)
3 min<-0
4 max<-1
5 #min_max
6 min_max<-((35-min(v))/(max(v)-min(v)))
7 print(min_max)
8 #z-score
9 m=mean(v)
10 s<-12.94
11 z_score=(35-m)/s
12 print(z_score)
13 #decimal_scaling
14 m<-35
15 j=max(m)<1
16 decimal_scaling=m/10^j
17 print(decimal_scaling)
18

```

Console Output:

```

> source("~/active-rstudio-document")
[1] 0.3157895
[1] -0.8844238
[1] 35
> #192211089
> v<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)
> print(min_max)
[1] 0.3157895
>
> print(z_score)
[1] -0.8844238
> print(decimal_scaling)
[1] 35

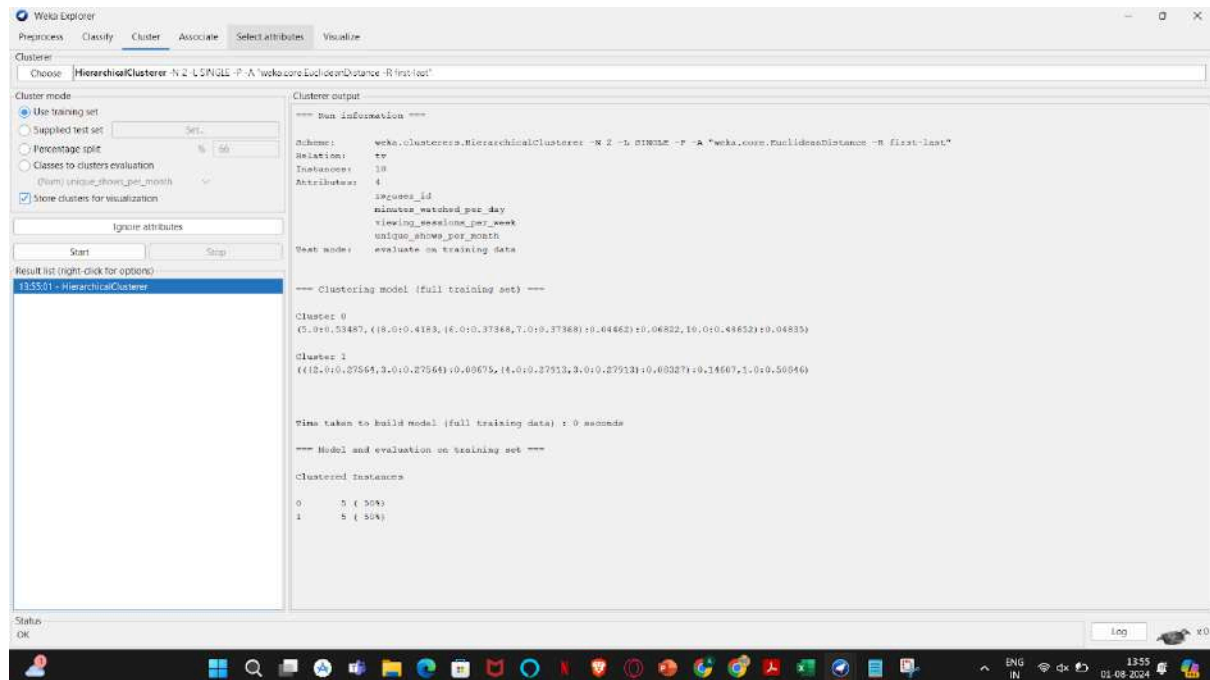
```

Variable Viewer:

Variable	Value
decimal_scaling	35
j	FALSE
m	35
max	1
min	0
min_max	0.315789473684211
s	12.94
v	num [1:18] 23 23 27 27 39 41 47 49 50 52 ...
z_score	-0.88442383651039

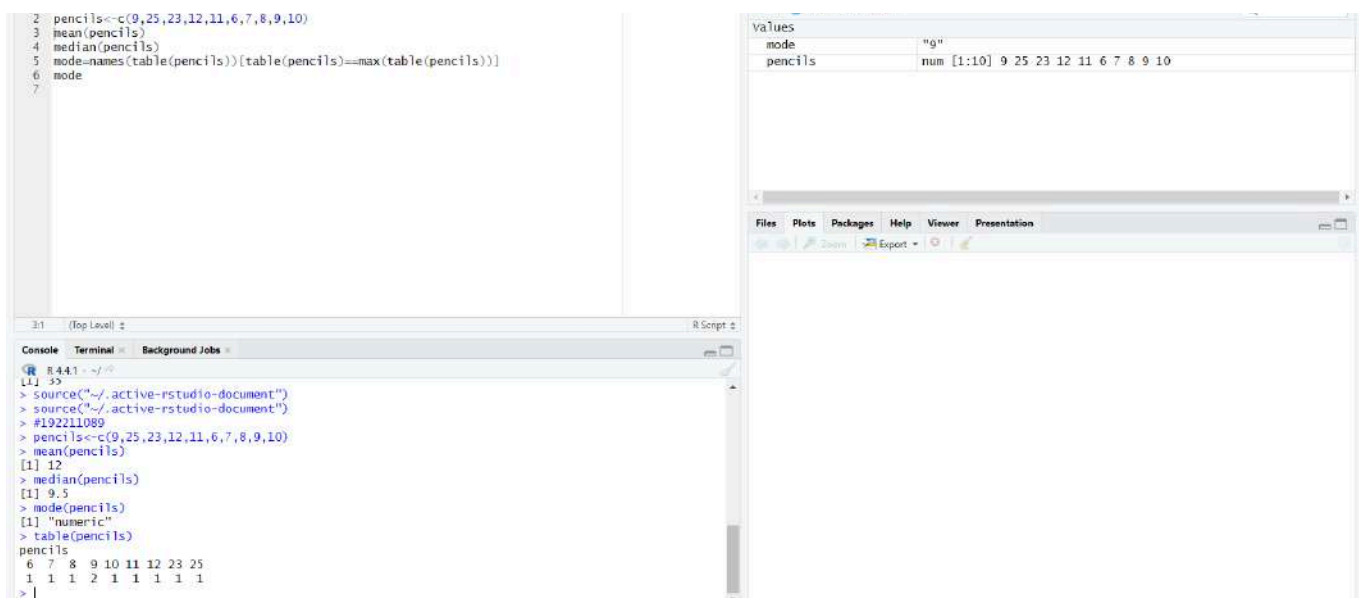
19) Consider that you are owning a supermarket mall and through membership cards, you have some basic data about your customers like Customer ID, age, gender, annual income and spending score. For the above scenario, the Problem Statement was You want to understand the customers who can easily converge [Target Customers] so that the data can be given to the marketing team and plan the strategy accordingly. For the above scenario prepare a dataset and perform Clustering Analysis to segment the customers in the Mall. There are clearly Five segments of Customers based on their Annual Income and Spending Score namely Usual Customers, Priority Customers, Senior Citizen Target Customers, and Young Target Customers. Sample data

low-usage users so that they can know whom they should spend most of their advertising dollars on. Apply the Hierarchical clustering algorithm and EM clustering algorithm to identify and compare the performance of the clustering technique.

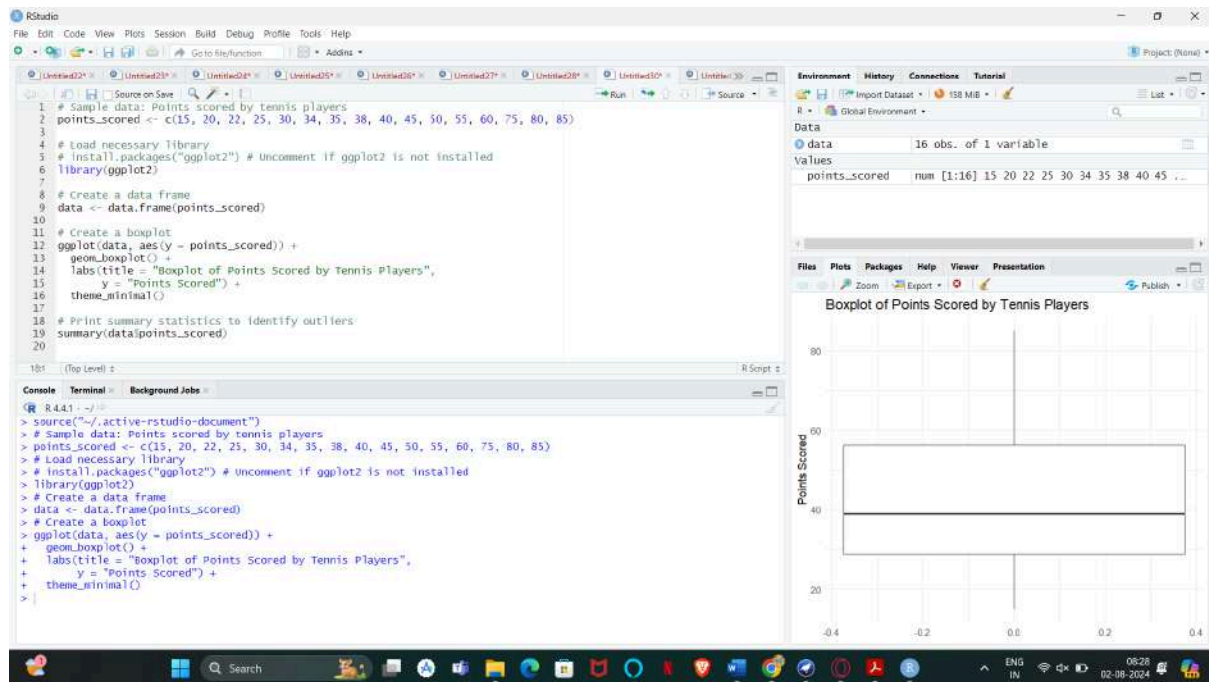


21)The following values are the number of pencils available in the different boxes. Create a

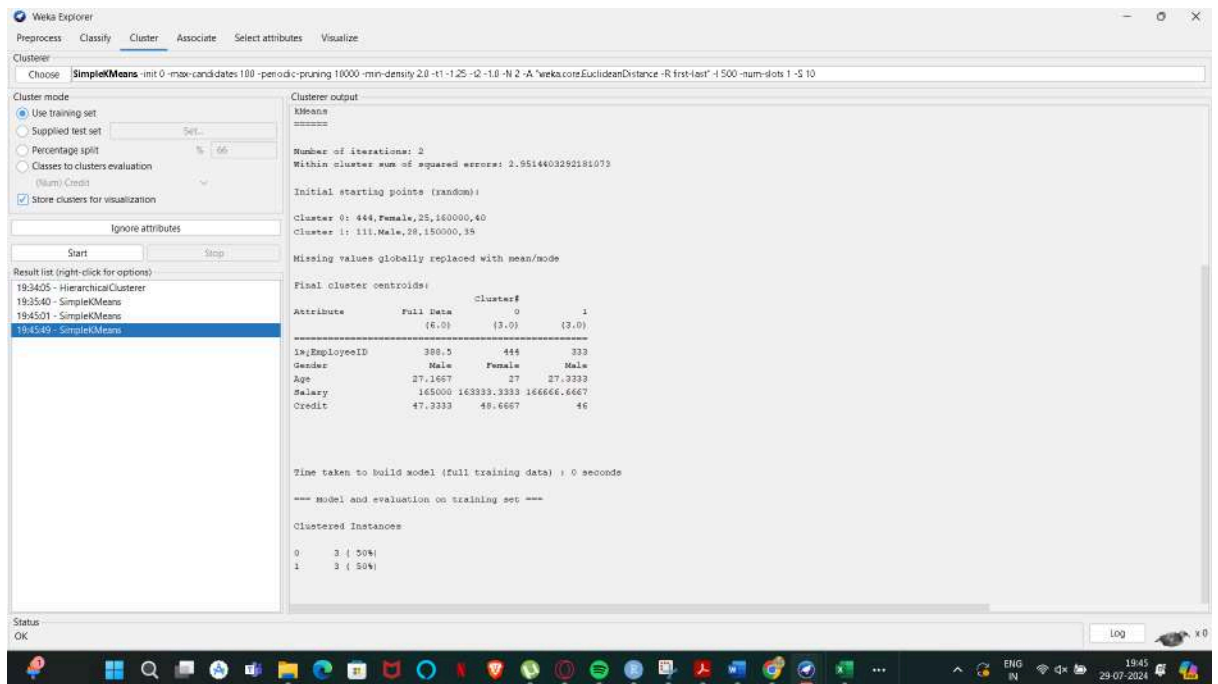
vector and find out the mean, median and mode values of set of pencils in the given data.



22) Assume the Tennis coach wants to determine if any of his team players are scoring outliers. To visualize the distribution of points scored by his players, then how can he decide to develop the box plot? Give suitable example using Boxplot visualization technique.



23) Create the following dataset using CSV file format. To perform cluster analysis using K- Means in WEKA. To change the cluster size and plot the graph and illustrate the visualization of cluster.



24) Prediction of categorical data using Naïve Bayes classification through WEKA using any datasets. Compare the Naïve Bayes algorithm with SVM using the summary of results given by the classifiers and plot the graph.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **NaiveBayes**

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 99
 More options...

(Nom) class: Start Stop

Result list (right-click for options):
 113403 - trees.DecisionStump
 113747 - functions.Logistic
194909 - bayes.NaiveBayes

Classifier output:

```

=====
Time taken to build model: 0.01 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      586      76.3021 %
Incorrectly Classified Instances    182      23.6979 %
Kappa statistic                    0.4666
Mean absolute error                0.2841
Root mean squared error            0.4169
Relative absolute error            62.5028 %
Root relative squared error        87.4949 %
Total Number of Instances          768

==== Detailed Accuracy By Class ====

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
-----
      0.844    0.388    0.802    0.844    0.823    0.468    0.819    0.892    tested_negative
      0.612    0.156    0.678    0.612    0.643    0.468    0.818    0.671    tested_positive
Weighted Avg.    0.763    0.307    0.755    0.763    0.760    0.468    0.819    0.815

==== Confusion Matrix ====

  a  b  <-- classified as
422 78 |  a = tested_negative
104 164 |  b = tested_positive
  
```

Status: OK Log

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **RandomForest** P 100 -I 100 -num-dots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 99
 More options...

(Nom) class: Start Stop

Result list (right-click for options):
 113403 - trees.DecisionStump
 113747 - functions.Logistic
 194909 - bayes.NaiveBayes
195021 - trees.RandomForest

Classifier output:

```

=====
Time taken to build model: 0.22 seconds

==== Stratified cross-validation ====
==== Summary ====

Correctly Classified Instances      582      75.7813 %
Incorrectly Classified Instances    186      24.2188 %
Kappa statistic                    0.4566
Mean absolute error                0.2106
Root mean squared error            0.4031
Relative absolute error            68.3405 %
Root relative squared error        84.5604 %
Total Number of Instances          768

==== Detailed Accuracy By Class ====

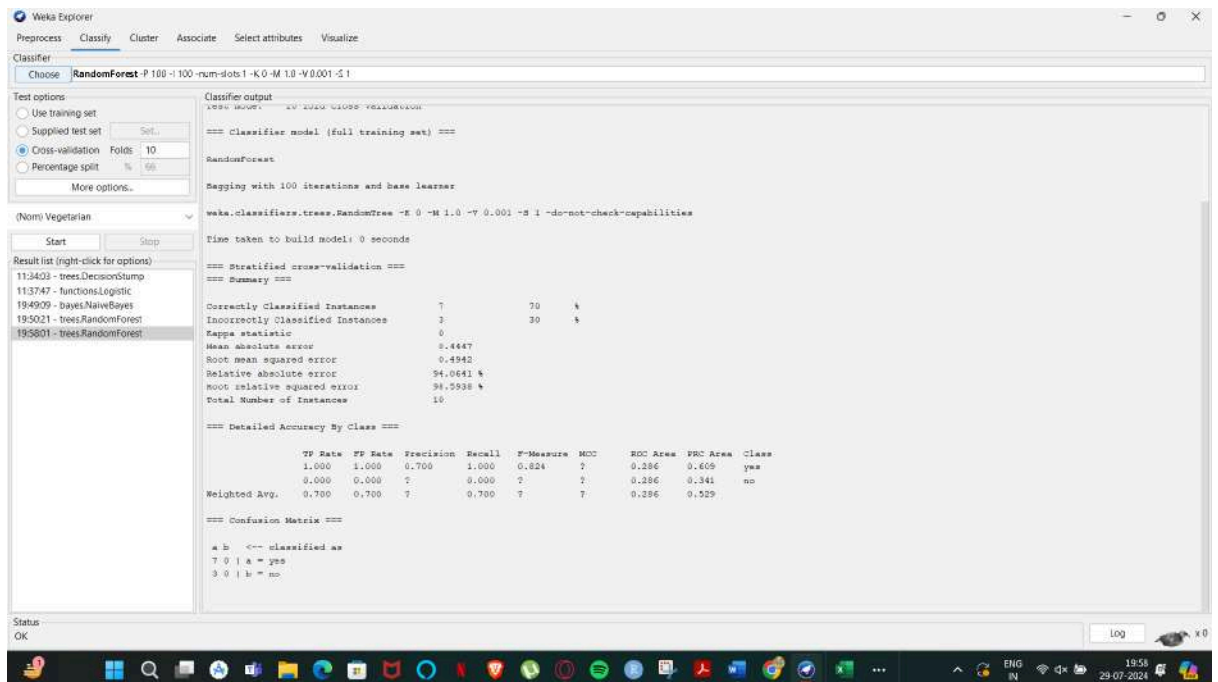
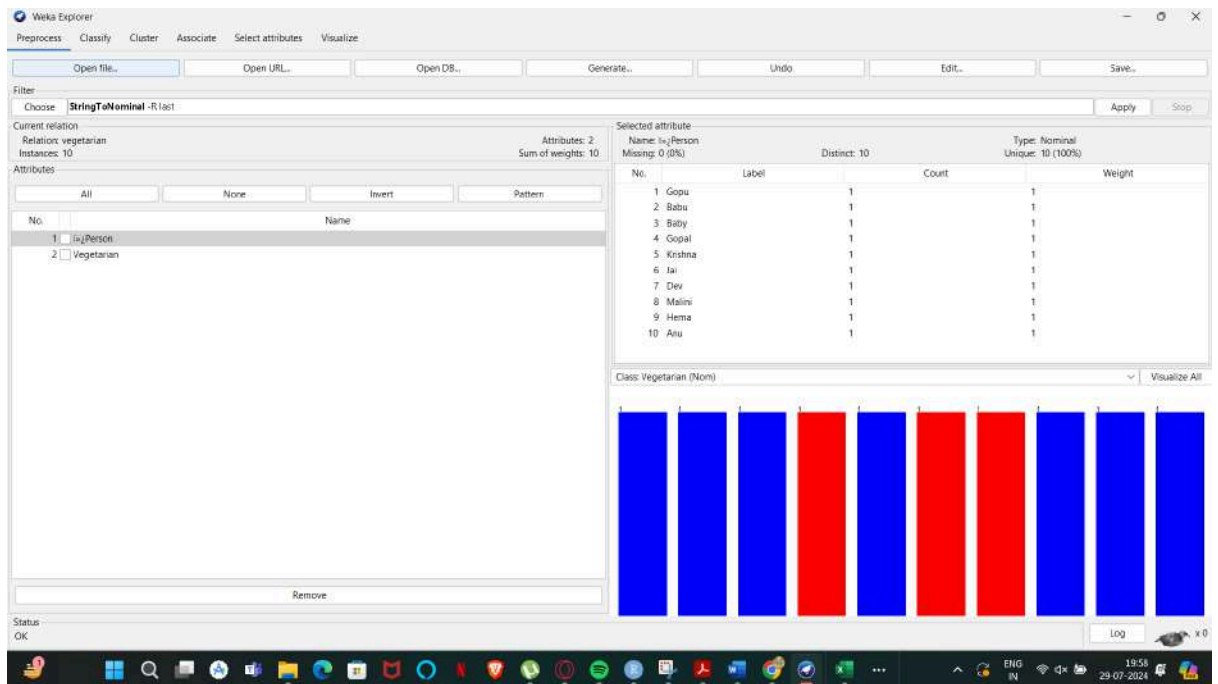
      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
-----
      0.836    0.388    0.801    0.836    0.818    0.458    0.820    0.886    tested_negative
      0.612    0.164    0.667    0.612    0.638    0.458    0.820    0.679    tested_positive
Weighted Avg.    0.758    0.310    0.754    0.758    0.755    0.458    0.820    0.814

==== Confusion Matrix ====

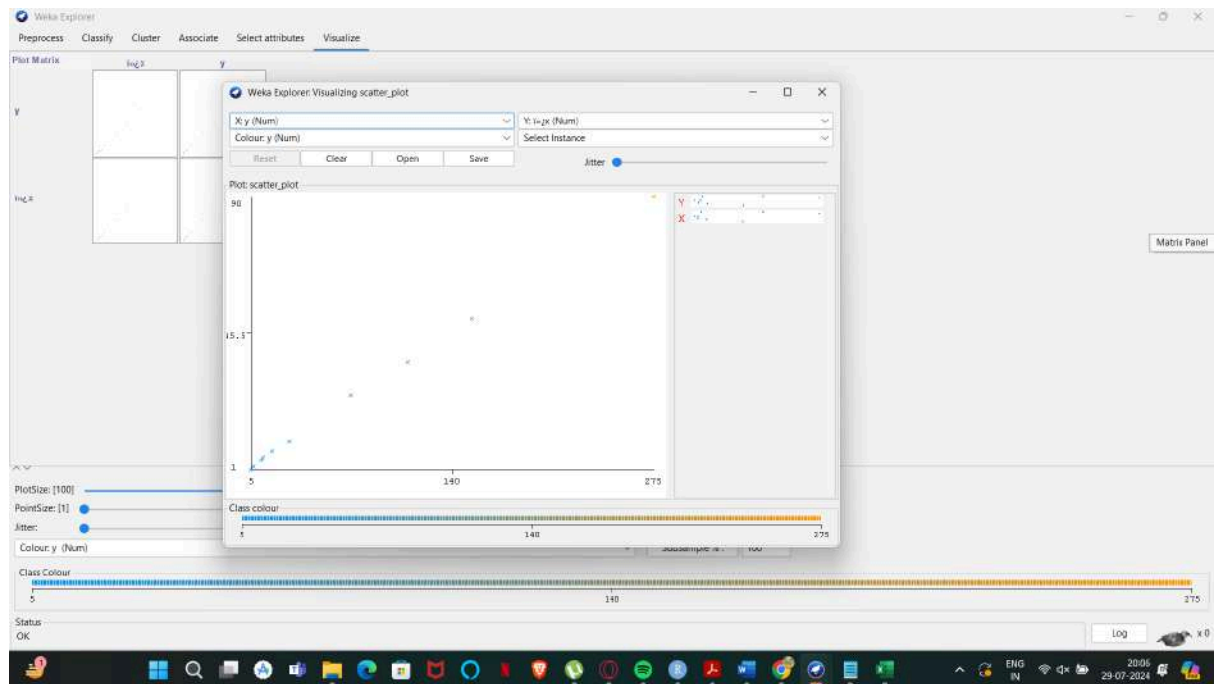
  a  b  <-- classified as
418 82 |  a = tested_negative
104 164 |  b = tested_positive
  
```

Status: OK Log

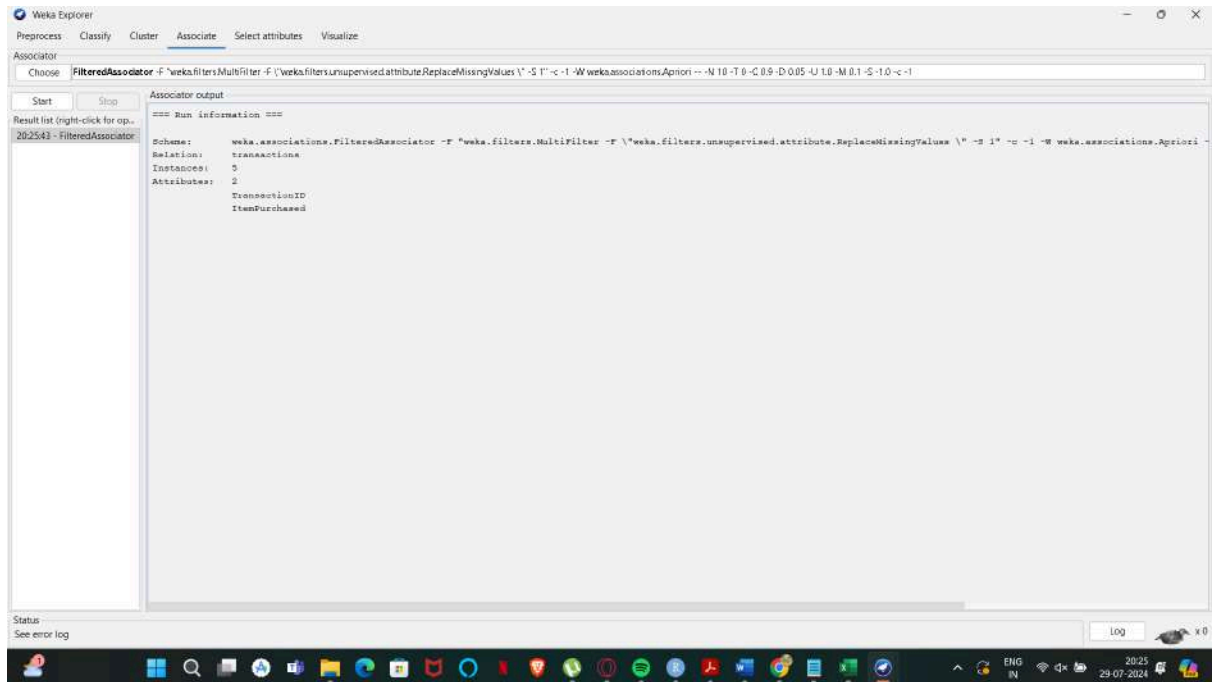
25) The following list of persons with vegetarian or not details given in the table. How will you find out how many of them are vegetarian and how many of them are non-vegetarian? Which type of the person total count is greater value?



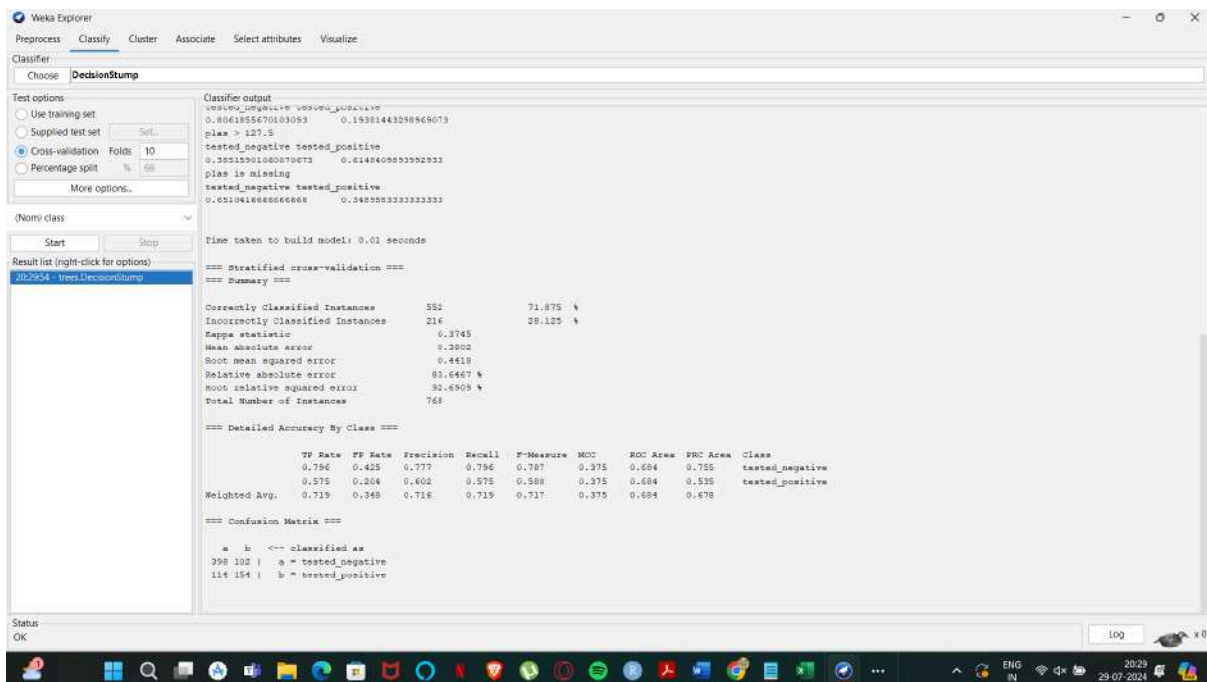
26) The following table would be plotted as (x,y) points, with the first column being the x values as number of mobile phones sold and the second column being the y values as money. To use the scatter plot for how many mobile phones sold.

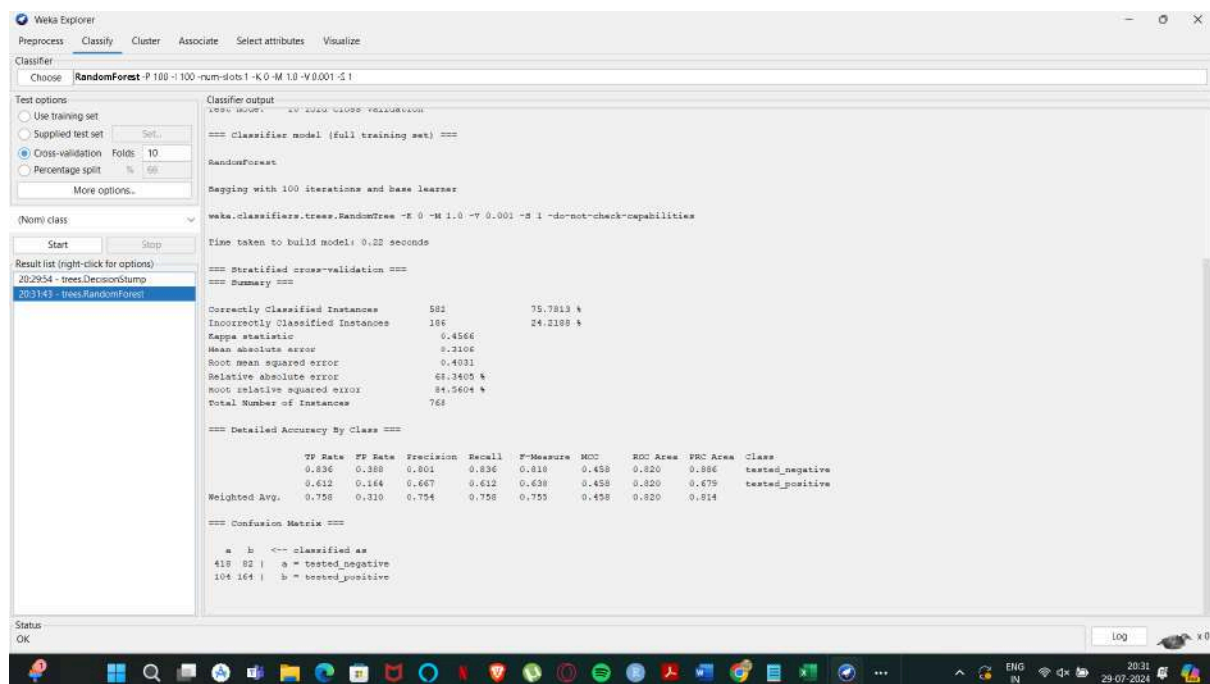


27) Generate rules using FP growth algorithm using the given dataset which has the following transactions with items purchased: Consider the values as support=50% and confidence=75%.



28) Prediction of Diabetes Data using Decision tree classifier in WEKA. Compare it with Support Vector Machine classifier. Show the result accuracy and F1 measure calculation .Plot the graph and explain the summary of results.



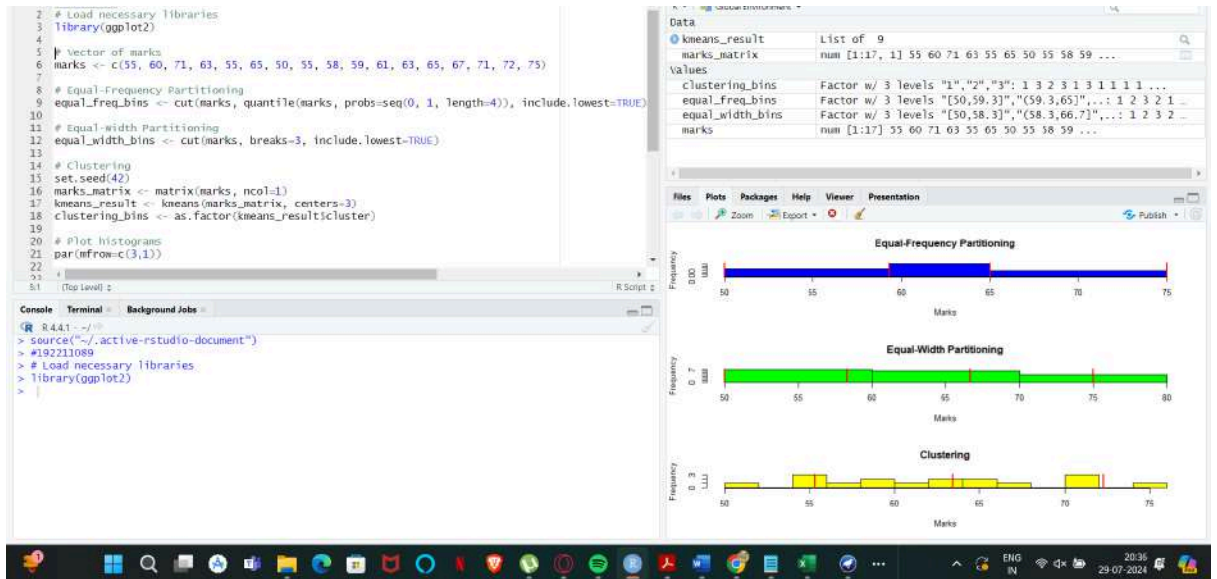


29) Implement of the R script using marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75. Partition them into three bins by each of the following methods. Plot the data points using histogram.

(a) equal-frequency (equi-depth) partitioning

(b) equal-width partitioning

(c) clustering

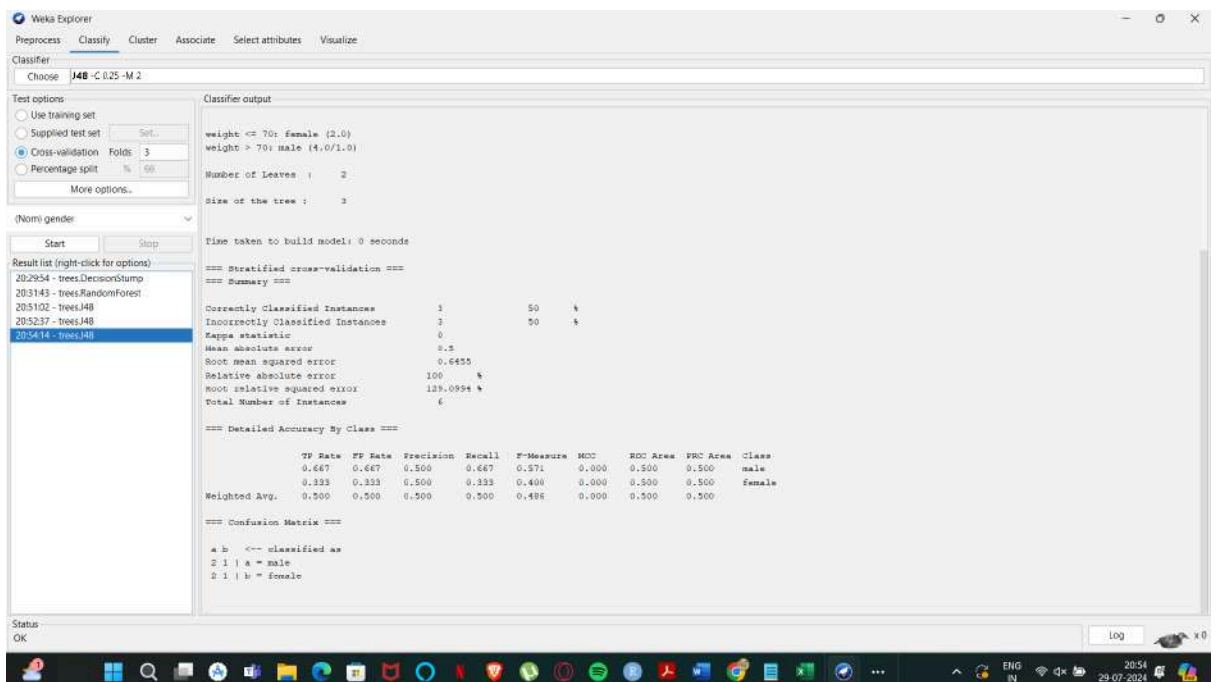


30) Consider this Decision tree :

a) create the data set for the below tree using ARFF format and calculate accuracy and decision for the same

b) Using this decision tree generate the rules based on rule based induction.

c) Compare both the algorithms and plot the confusion matrix.



31) Create an ARFF file for the table below and implement for the Apriori Algorithm and FP growth algorithm and compare the rules generated by both the algorithms. Identify the unique rules generated by the above algorithms.

```

Weka Explorer
Preprocess  Classify  Cluster  Associate  Select attributes  Visualize

Associate
Choose  Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -e -1

Start  Stop
Result list (right-click for...)
205944 - Apriori

Associate output
Show associated rules
Schema:      weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -e -1
Relation:    electronics
Instances:    9
Attributes:    2
Attributes:    2
Items

=== Associate model (Full training set) ===

Apriori
=====

Minimum support: 0.16 (1 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 16
Size of set of large itemsets L(2): 9

Best rules found:

1. ITEMS=SONY, SPL, LG 1 ==> I9_T.IDT1 1  <conf: (1)> lift: (9) lev: (0.1) [0] conv: (0.85)
2. I9_T.IDT1 1 ==> ITEMS=SONY, SPL, LG 1  <conf: (1)> lift: (9) lev: (0.1) [0] conv: (0.85)
3. ITEMS=SPL, SAMSUNG 1 ==> I9_T.IDT2 1  <conf: (1)> lift: (9) lev: (0.1) [0] conv: (0.85)
4. I9_T.IDT2 1 ==> ITEMS=SPL, SAMSUNG 1  <conf: (1)> lift: (9) lev: (0.1) [0] conv: (0.85)
5. I9_T.IDT3 1 ==> ITEMS=SPL, ONIDA 1  <conf: (1)> lift: (4.5) lev: (0.09) [0] conv: (0.78)
6. ITEMS=SONY, SPL, SAMSUNG 1 ==> I9_T.IDT4 1  <conf: (1)> lift: (9) lev: (0.1) [0] conv: (0.85)
7. I9_T.IDT4 1 ==> ITEMS=SONY, SPL, SAMSUNG 1  <conf: (1)> lift: (9) lev: (0.1) [0] conv: (0.85)
8. I9_T.IDT5 1 ==> ITEMS=SONY, ONIDA 1  <conf: (1)> lift: (4.5) lev: (0.09) [0] conv: (0.78)
9. I9_T.IDT6 1 ==> ITEMS=SPL, ONIDA 1  <conf: (1)> lift: (4.5) lev: (0.09) [0] conv: (0.78)
10. I9_T.IDT7 1 ==> ITEMS=SONY, ONIDA 1  <conf: (1)> lift: (4.5) lev: (0.09) [0] conv: (0.78)

Status
OK
Log
x 0
  
```

32) The given are the strike-rates scored by a batsman in season 1 in different tournaments. 100, 70, 60, 90, 90

(a) min-max normalization by setting min = 0 and max = 1

(b) z-score normalization

(c) z-score normalization using the mean absolute deviation instead of standard deviation

(d) normalization by decimal scaling

```

2 # Define the strike rates
3 strike_rates <- c(100, 70, 60, 90, 90)
4
5 # Min-Max Normalization
6 min_max_normalized <- (strike_rates - min(strike_rates)) / (max(strike_rates) - min(strike_rates))
7 print("Min-Max Normalization:")
8 print(min_max_normalized)
9
10 # Z-Score Normalization
11 mean_sr <- mean(strike_rates)
12 sd_sr <- sd(strike_rates)
13 z_score_normalized <- (strike_rates - mean_sr) / sd_sr
14 print("Z-Score Normalization:")
15 print(z_score_normalized)
16
17 # Z-Score Normalization using Mean Absolute Deviation
18 mad_sr <- mean(abs(strike_rates - mean_sr))
19 z_score_mad_normalized <- (strike_rates - mean_sr) / mad_sr
20 print("Z-Score Normalization using Mean Absolute Deviation:")
21 print(z_score_mad_normalized)
22
23 # Normalization by Decimal Scaling:
24 decimal_scaling_norm <- (strike_rates - min(strike_rates)) / 100
25 print("Normalization by Decimal Scaling:")
26 print(decimal_scaling_norm)

```

Console Output:

```

R 4.4.1 ~./
[1] "Min-Max Normalization:"
[1] 1.00 0.25 0.00 0.75 0.75
[1] "Z-Score Normalization:"
[1] 1.054451 -0.7302967 -1.3388774 0.4868645 0.4868645
[1] "Z-Score Normalization using Mean Absolute Deviation:"
[1] 1.3235294 -0.8823529 -1.6176471 0.5882353 0.5882353
[1] "Normalization by Decimal Scaling:"
[1] 1.0 0.7 0.6 0.9 0.9
> #192211089
> # Define the strike rates
> strike_rates <- c(100, 70, 60, 90, 90)
> print(min_max_normalized)
[1] 1.00 0.25 0.00 0.75 0.75
> print(z_score_normalized)
[1] 1.3235294 -0.8823529 -1.6176471 0.5882353 0.5882353
>

```

Values:

	num [1:5]
decimal_scaling_norm	1 0.7 0.6 0.9 0.9
j	2
mad_sr	13.6
max_abs_sr	100
mean_sr	82
min_max_normalized	num [1:5] 1 0.25 0 0.75 0.75
sd_sr	16.431676725155
strike_rates	num [1:5] 100 70 60 90 90
z_score_mad_normalized	num [1:5] 1.324 -0.882 -1.618 0.588 0.588
z_score_normalized	num [1:5] 1.055 -0.731 -1.339 0.487 0.487

33) Suppose some car is tested for the AvgSpeed and TotalTime data for 9 randomly selected car with the following result

- Calculate the standard deviation of AvgSpeed and TotalTime.
- Calculate the Variance of AvgSpeed and TotalTime for the above dataset.

```

3 AvgSpeed <- c(78, 81, 82, 74, 83, 82, 77, 80, 70)
4 TotalTime <- c(39, 37, 36, 42, 35, 36, 40, 38, 46)
5
6 # Calculate standard deviation
7 sd_AvgSpeed <- sd(AvgSpeed)
8 sd_TotalTime <- sd(TotalTime)
9
10 # Calculate variance
11 var_AvgSpeed <- var(AvgSpeed)
12 var_TotalTime <- var(TotalTime)
13
14 # Print the results
15 cat("Standard Deviation of AvgSpeed:", sd_AvgSpeed, "\n")
16 cat("Standard Deviation of TotalTime:", sd_TotalTime, "\n")
17
18 cat("Variance of AvgSpeed:", var_AvgSpeed, "\n")
19 cat("Variance of TotalTime:", var_TotalTime, "\n")
20

```

Console Output:

```

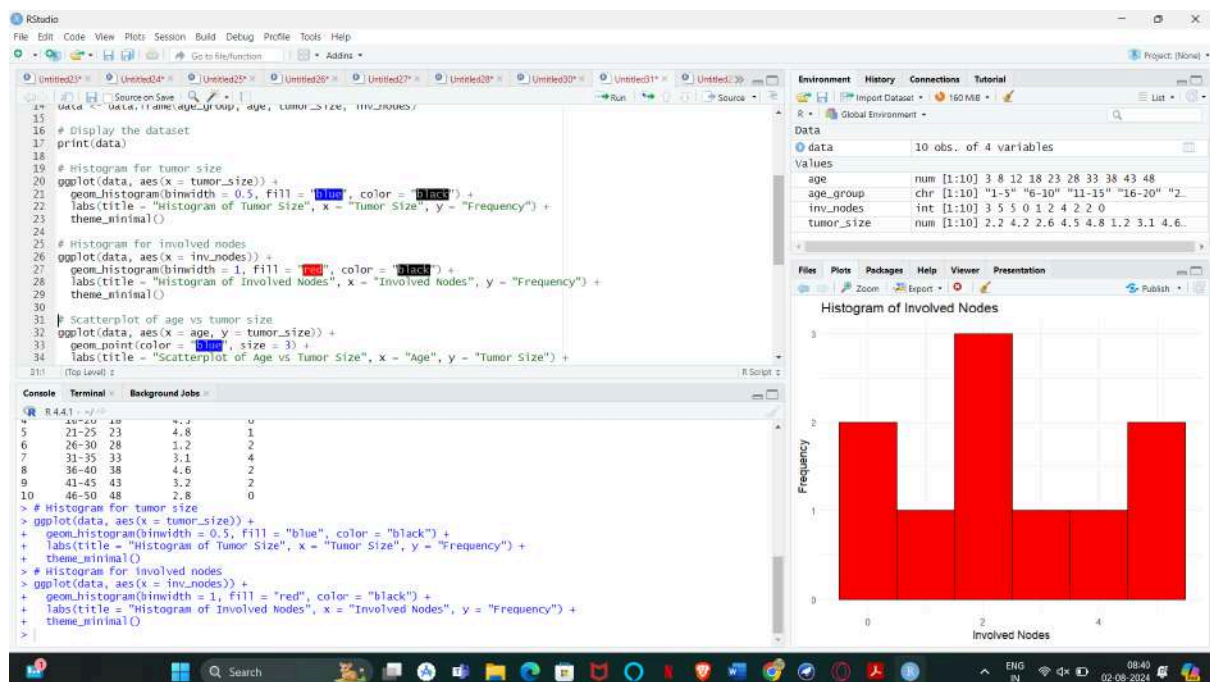
R 4.4.1 ~./
> source("~/active-rstudio-document")
Standard Deviation of AvgSpeed: 4.304391
Standard Deviation of TotalTime: 3.492054
Variance of AvgSpeed: 18.52778
Variance of TotalTime: 12.19444
> #192211089
> # Define the data
> AvgSpeed <- c(78, 81, 82, 74, 83, 82, 77, 80, 70)
> cat()
> var(AvgSpeed)
[1] 18.52778
> var(TotalTime)
[1] 12.19444
>

```

Values:

	num [1:9]
AvgSpeed	78 81 82 74 83 82 77 80 70
sd_AvgSpeed	4.30439052338165
sd_TotalTime	3.49205447329283
TotalTime	num [1:9] 39 37 36 42 35 36 40 38 46
var_AvgSpeed	18.5277777777778
var_TotalTime	12.1944444444444

34) Consider a person want to take a census / plot for the breast-cancer affected people through the years. Create a own dataset with this parameters age, tumorsize, inv-nodes
[example between age 1-5 = no.of.count, 6-10=no.of.count,etc]
Draw the Histogram, scatterplot, boxplot.



35) Create the Confusion matrix using this scenario:

A shepherd boy gets bored tending the town's flock. To have some fun, he cries out, "Wolf!" even though no wolf is in sight. The villagers run to protect the flock, but then get really mad when they realize the boy was playing a joke on them. One night, the shepherd boy sees a real wolf approaching the flock and calls out, "Wolf!" The villagers refuse to be fooled again and stay in their houses. The hungry wolf turns the flock into lamb chops. The town goes hungry. Panic ensues.

True Positive (TP):

- Reality: A wolf threatened.
- Shepherd said: "Wolf."
- Outcome: Shepherd is a hero.

False Positive (FP):

- Reality: No wolf threatened.
- Shepherd said: "Wolf."
- Outcome: Villagers are angry at shepherd for waking them up.

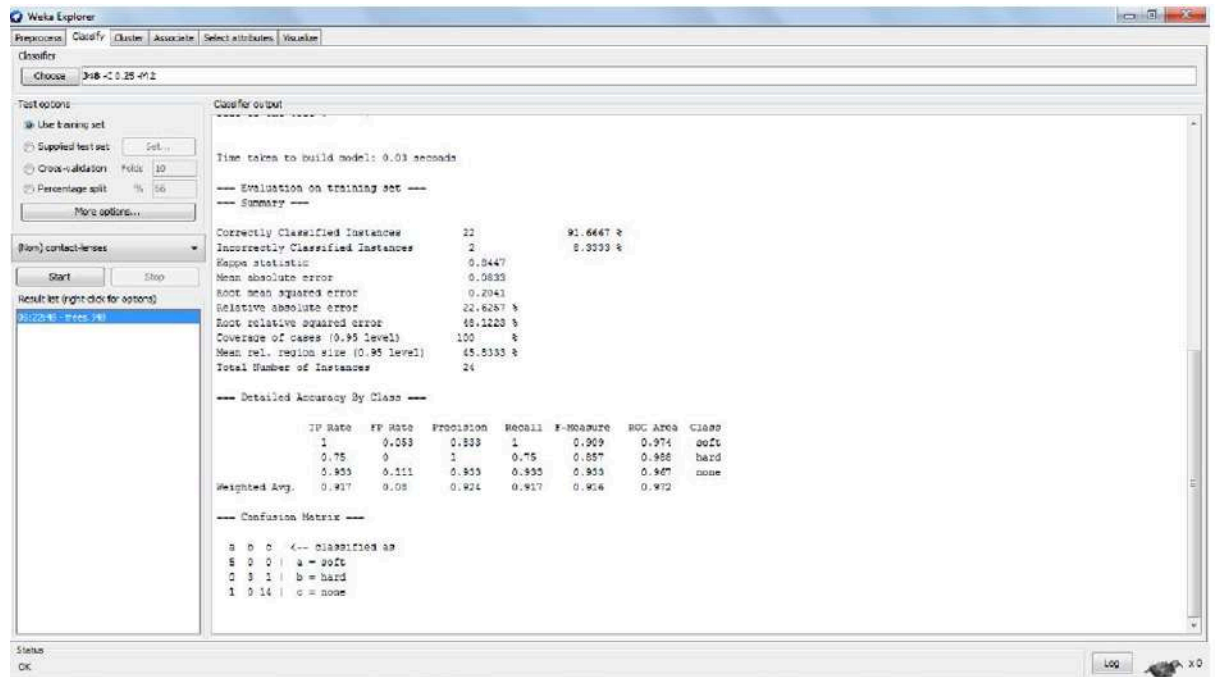
False Negative (FN):

- Reality: A wolf threatened.
- Shepherd said: "No wolf."
- Outcome: The wolf ate all the sheep.

True Negative (TN):

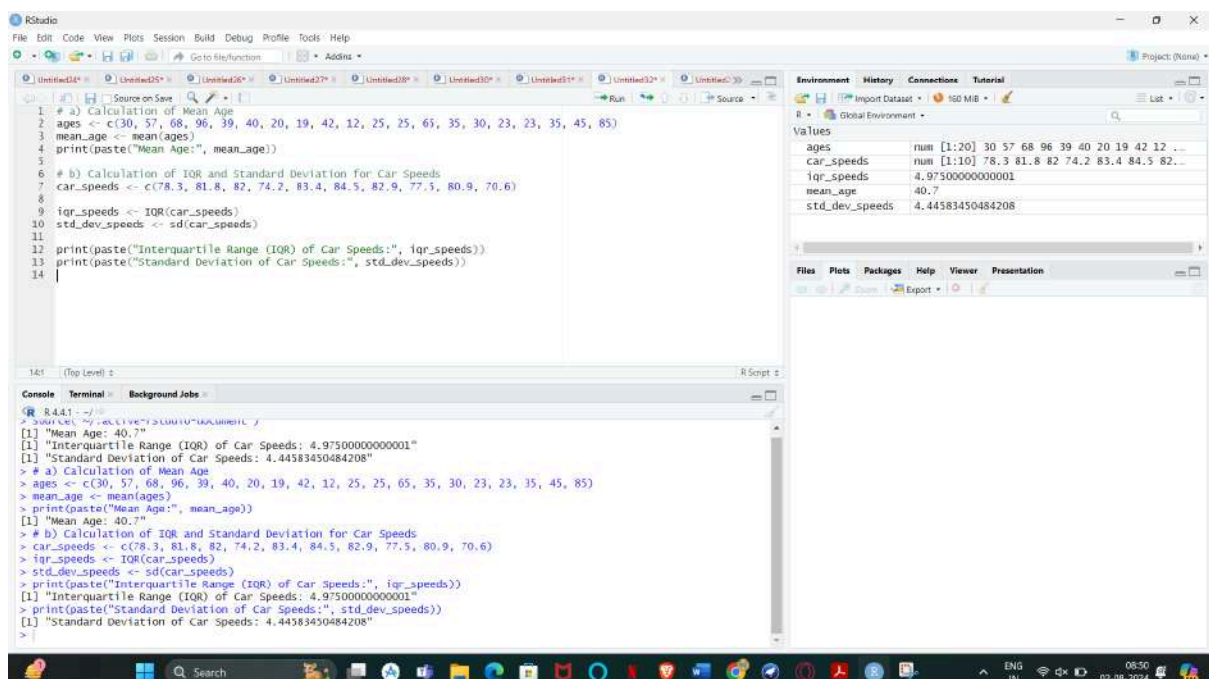
- Reality: No wolf threatened.
- Shepherd said: "No wolf."
- Outcome: Everyone is fine.

36) Create the ARFF data set for the below mentioned dataset perform the bayes theorem in addition to that compare the same with decision tree. identify the efficient classifier with accuracy with F1 Score.



37) a) Suppose that the “Diabetes data set ” data for analysis includes the attribute age. The age values for the data are (in increasing order) 30, 57, 68, 96, 39, 40, 20, 19, 42, 12, 25, 25, 65, 35, 30, 23, 23, 35, 45, 85. What is the mean?

b) Suppose that the speed car is mentioned in different driving style.



38) a) Let us consider one example to make the calculation method clear. Assume that the minimum and maximum values for the feature F are \$50,000 and \$100,000 correspondingly. It needs to range F from 0 to 1. In accordance with min-max normalization, $v = \$80$,

b) Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000

(a) min-max normalization by setting $\min = 0$ and $\max = 1$

(b) z-score normalization

The screenshot shows the RStudio interface with the following content:

Script Editor:

```

1 # Given data
2 data <- c(200, 300, 400, 600, 1000)
3
4 # a) Min-Max Normalization
5 min_d <- min(data)
6 max_d <- max(data)
7 min_max_normalized <- (data - min_d) / (max_d - min_d)
8 print("Min-Max Normalized Data:")
9 print(min_max_normalized)
10
11 # b) Z-Score Normalization
12 mean_d <- mean(data)
13 std_dev_d <- sd(data)
14 z_score_normalized <- (data - mean_d) / std_dev_d
15 print("Z-Score Normalized Data:")
16 print(z_score_normalized)
17

```

Environment Pane:

Variable	Class	Value
data	num [1:5]	200 300 400 600 1000
max_d	num	1000
mean_d	num	500
min_d	num	200
min_max_normalized	num [1:5]	0 0.125 0.25 0.5 1
std_dev_d	num	316.227766016838
z_score_normalized	num [1:5]	-0.949 -0.632 -0.316 0.316 1.581

Console:

```

> # Given data
> data <- c(200, 300, 400, 600, 1000)
> # a) Min-Max Normalization
> min_d <- min(data)
> max_d <- max(data)
> min_max_normalized <- (data - min_d) / (max_d - min_d)
> print("Min-Max Normalized Data:")
[1] "Min-Max Normalized Data:"
> print(min_max_normalized)
[1] 0.000 0.125 0.250 0.500 1.000
> # b) Z-Score Normalization
> mean_d <- mean(data)
> std_dev_d <- sd(data)
> z_score_normalized <- (data - mean_d) / std_dev_d
> print("Z-Score Normalized Data:")
[1] "Z-Score Normalized Data:"
> print(z_score_normalized)
[1] -0.9486833 -0.6324555 -0.3162278 0.3162278 1.5811388

```

39) Consider this table

TID items bought

T100 {M, O, N, K, E, Y}

T200 {D, O, N, K, E, Y}

T300 {M, A, K, E}

T400 {M, U, C, K, Y}

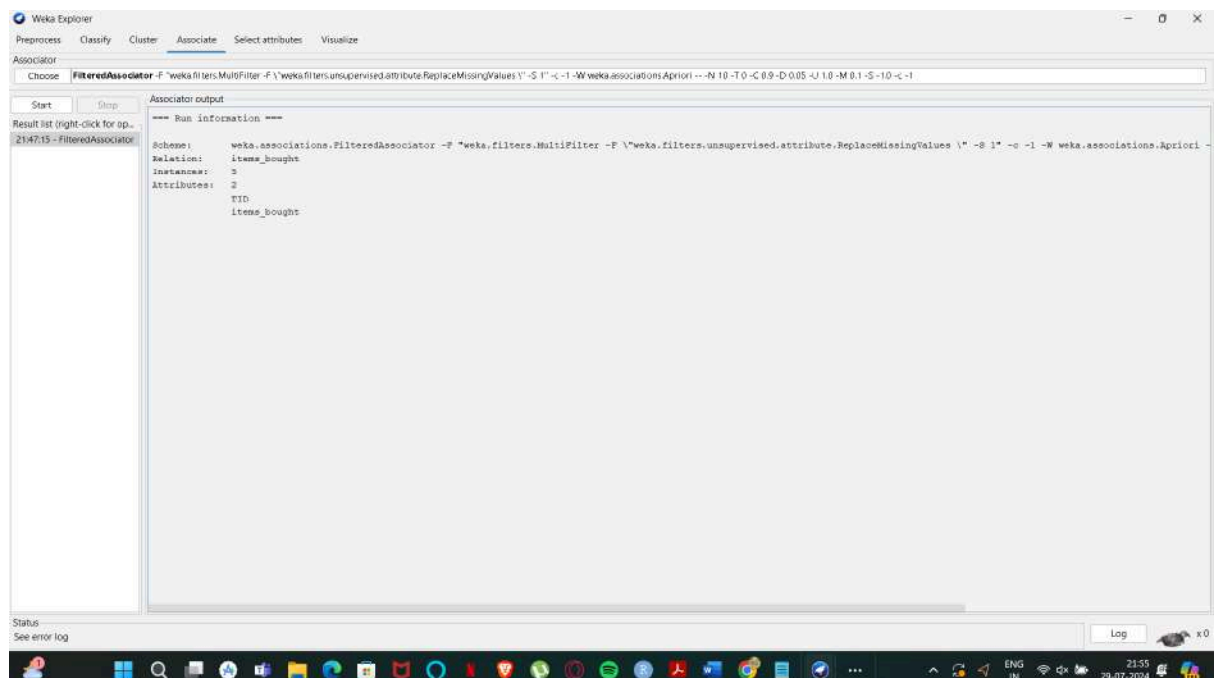
T500 {C, O, O, K, I, E}

(a) Find all frequent item set using Apriori and FP-growth, respectively.

Compare the efficiency of the two mining processes.

(b) List all of the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and itemi denotes variables representing items (e.g., “A”, “B”, etc.):

$$\forall x \in \text{transaction}, \text{buys}(X, \text{item1}) \wedge \text{buys}(X, \text{item2}) \Rightarrow \text{buys}(X, \text{item3})$$



40) Suppose we want to classify potential bank customers as good creditors or bad creditors for loan applications. We have a training dataset describing past customers using the following attributes: Marital status {married, single, divorced}, Gender {male, female}, Age {[18..30[, [30..50[, [50..65[, [65+]], Income {[10K..25K[, [25K..50K[, [50K..65K[, [65K..100K[, [100K+]}. Using Weka tool solve this problem.

Weka Explorer

PreprocessClassifyClusterAssociateSelect attributesVisualize

Classifier

ChooseJ48C 0.25-M 2

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation

☐ Percentage split

Folds5

%66

More options...

(Nom) Class

StartStop

Result list (right-click for options)

08:58:17 - treesJ48

08:58:30 - treesJ48

Classifier output

Marital_status = married: good (3.0)
Marital_status = single: bad (2.0)
Marital_status = divorced: good (1.0)
Number of leaves : 3
Size of the tree : 4
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances360 %
Incorrectly Classified Instances240 %
Kappa statistic0
Mean absolute error0.6
Root mean squared error0.6124
Relative absolute error105.8824 %
Root relative squared error106.9611 %
Total Number of Instances5

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	AUC Area	Class
	1.000	1.000	0.600	1.000	0.750	?	0.000	good
	0.000	0.000	?	0.000	?	?	0.000	bad
Weighted Avg.	0.600	0.600	?	0.600	?	?	0.000	0.520

=== Confusion Matrix ===

a \ b	Classified as
3 0	a = good
2 0	b = bad

StatusOK

Logx0

ENG IN

08:58
02-08-2024