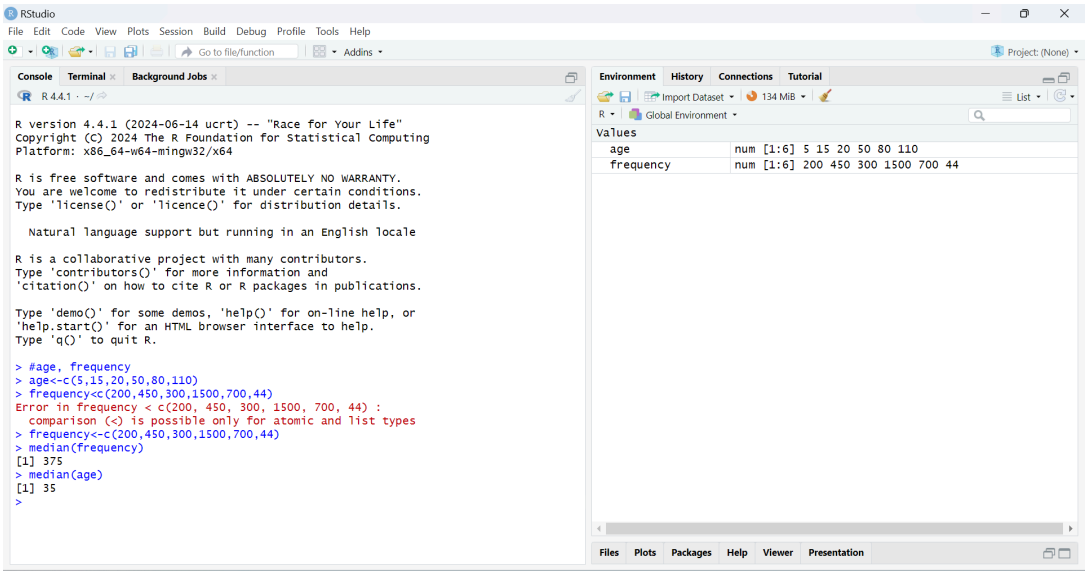
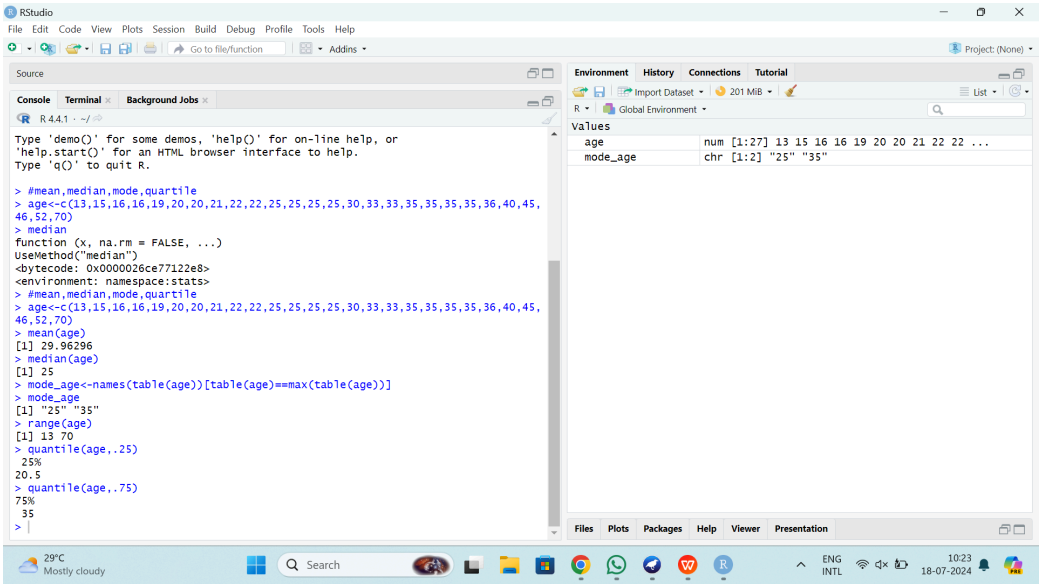
CSA1667-Data Warehousing And Data Mining

DAY-1

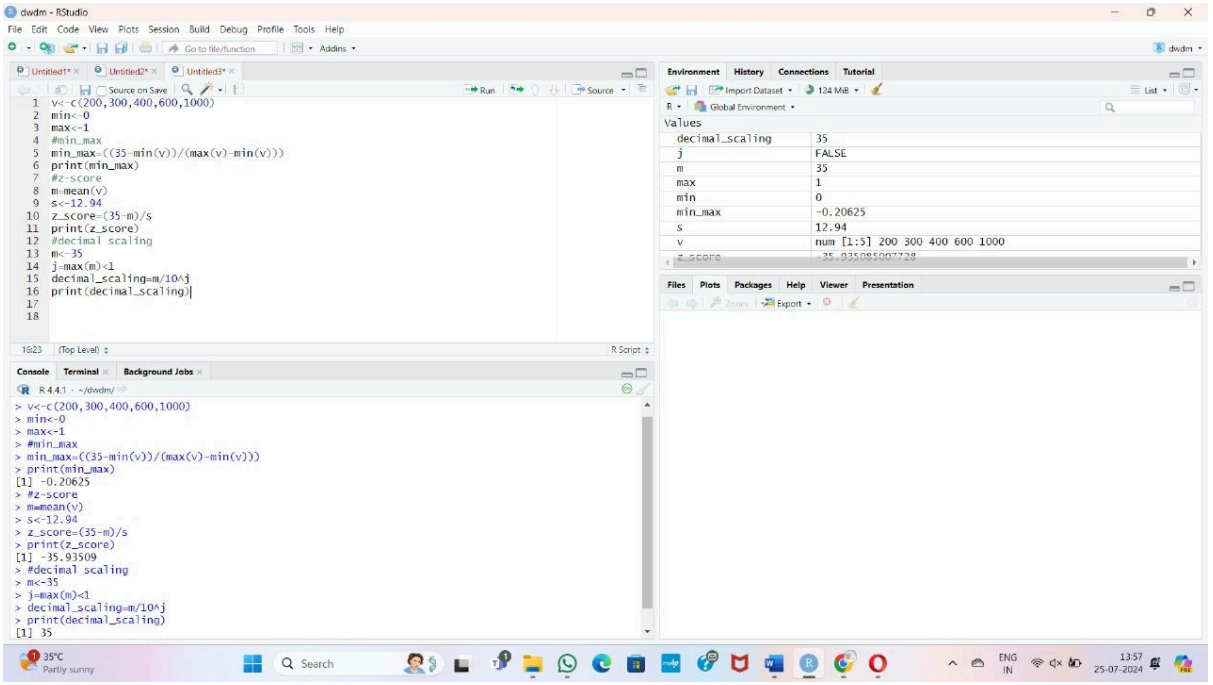
Q1:



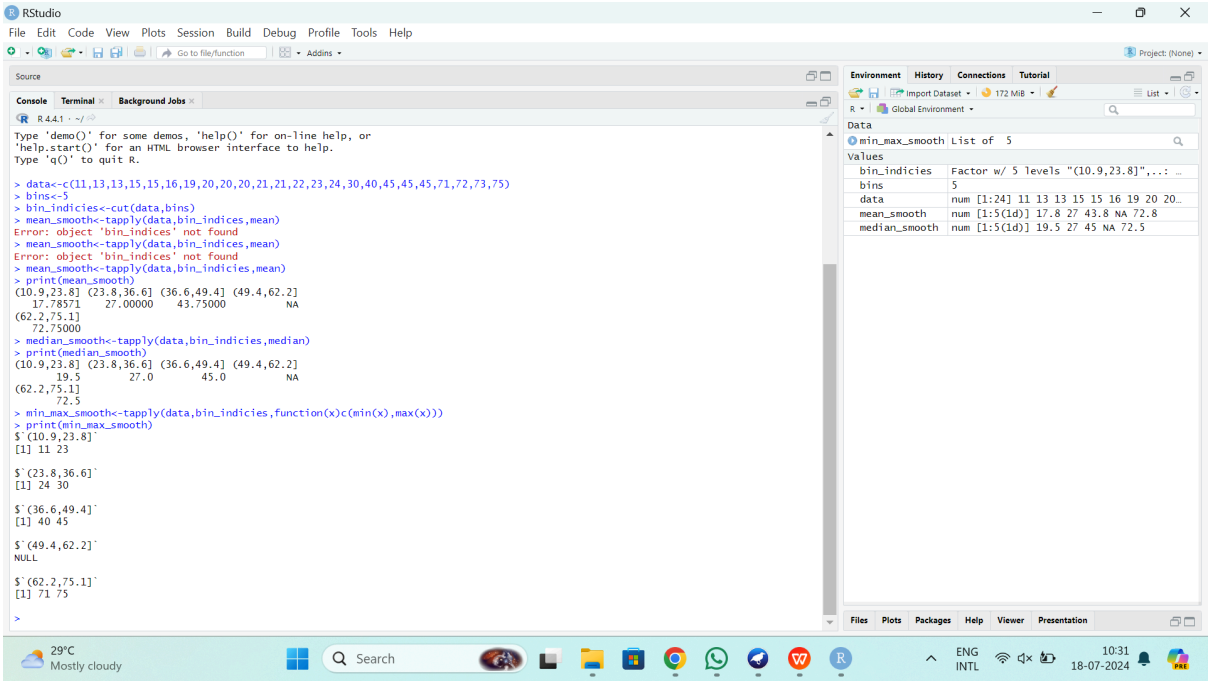
Q2:



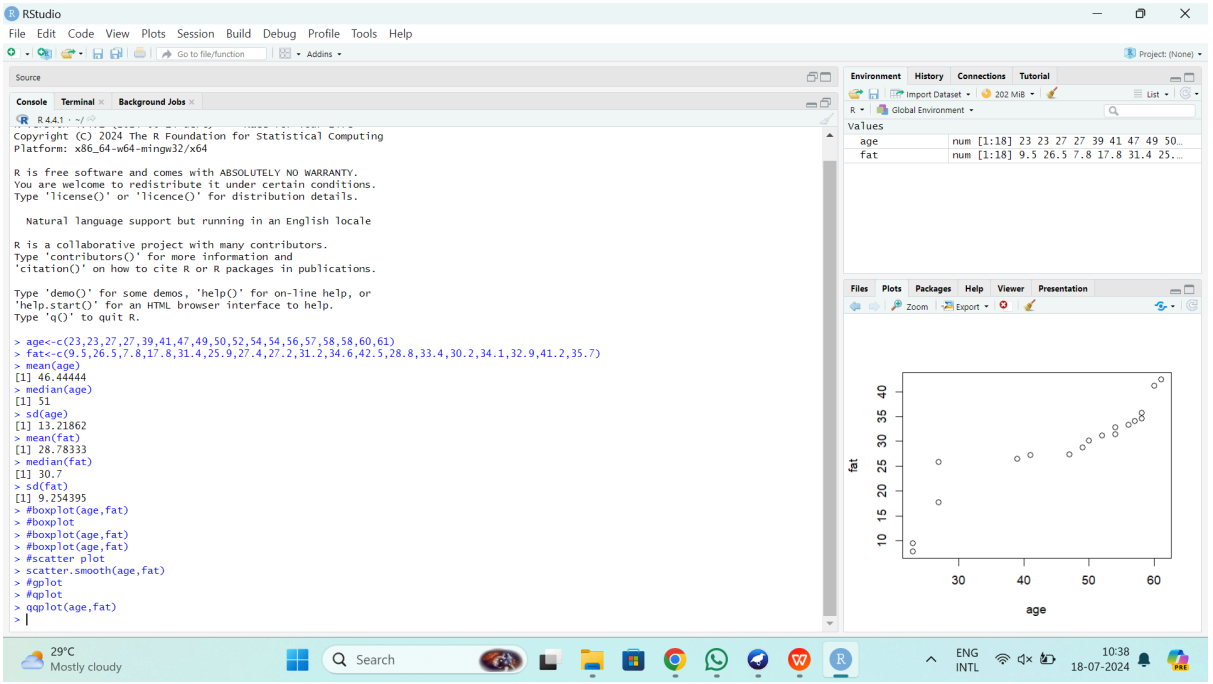
Q3:


dwdm - RStudio
v<-c(200,300,400,600,1000)
min<-0
max<-1
#min_max
min_max=((35-min(v))/(max(v)-min(v)))
print(min_max)
#z-score
m=mean(v)
s<-12.94
z_score=(35-m)/s
print(z_score)
#decimal scaling
m<-35
j=max(m)<1
decimal_scaling=m/10^j
print(decimal_scaling)
[1] -0.20625
[1] -35.93509
[1] 35

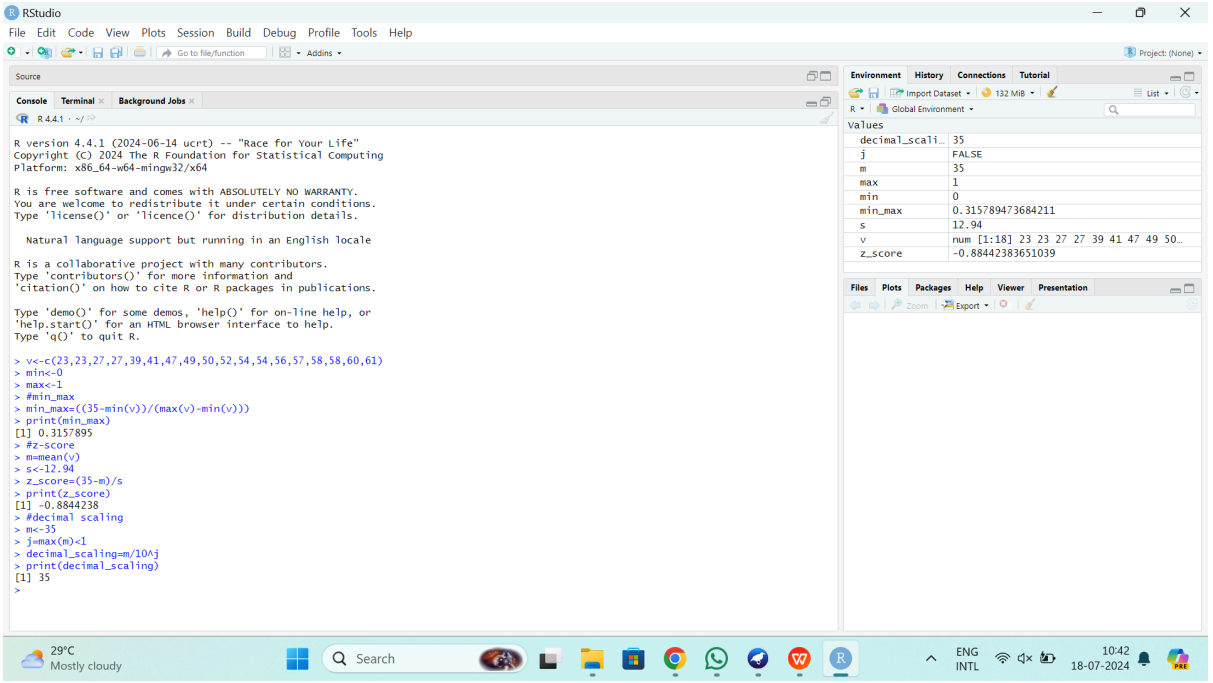
Q4:


RStudio
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)
bins<-5
bin_indicies<-cut(data,bins)
mean_smooth<-tapply(data,bin_indicies,mean)
Error: object 'bin_indices' not found
median_smooth<-tapply(data,bin_indicies,median)
min_max_smooth<-tapply(data,bin_indicies,function(x)c(min(x),max(x)))
print(min_max_smooth)
[1] 11 23
[1] 24 30
[1] 40 45
NULL
[1] 71 75

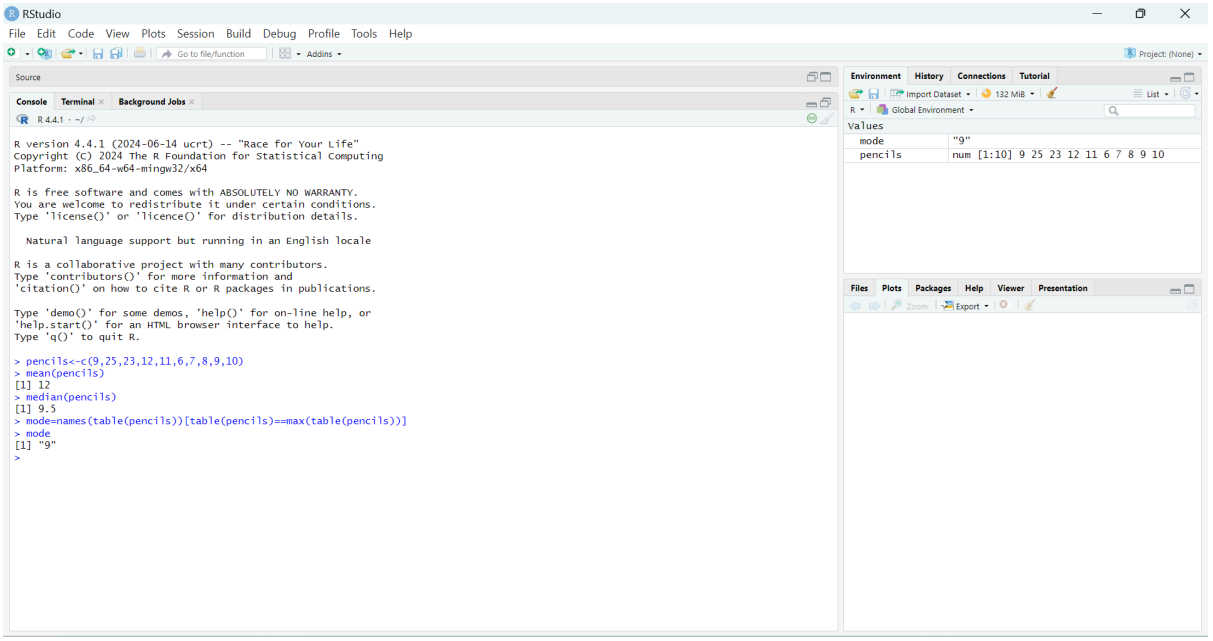
Q5:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Background Jobs
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64
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 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.
> age<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)
> fat<-c(9.5,26.5,7.8,17.8,31.4,25.9,27.4,27.2,31.2,34.6,42.5,28.8,33.4,30.2,34.1,32.9,41.2,35.7)
> mean(age)
[1] 46.44444
> median(age)
[1] 51
> sd(age)
[1] 13.21862
> mean(fat)
[1] 28.78333
> median(fat)
[1] 30.7
> sd(fat)
[1] 9.254395
> #boxplot(age,fat)
> #boxplot
> #boxplot(age,fat)
> #boxplot(age,fat)
> #scatter plot
> scatter.smooth(age,fat)
> #gplot
> #qplot
> qqplot(age,fat)
Environment History Connections Tutorial
Values
age num [1:18] 23 23 27 27 39 41 47 49 50...
fat num [1:18] 9.5 26.5 7.8 17.8 31.4 25...
Files Plots Packages Help Viewer Presentation
fat
age
10 15 20 25 30 35 40
30 40 50 60
29°C
Mostly cloudy
Search
ENG
INTL
10:38
18-07-2024

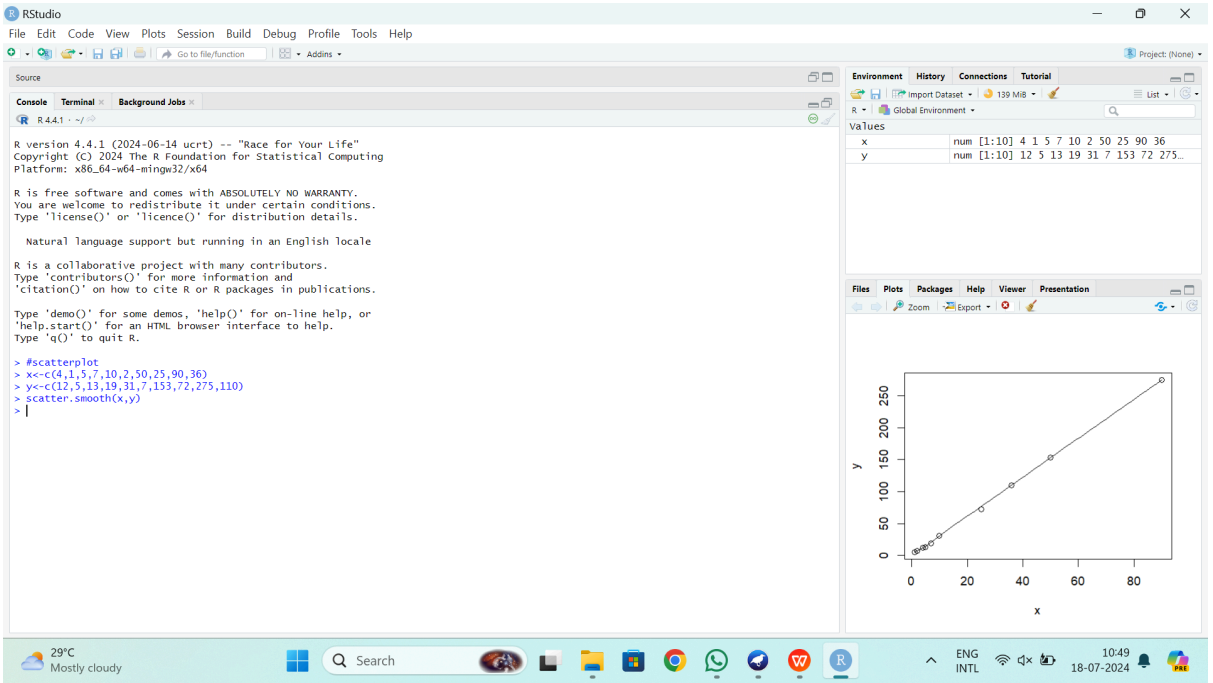
Q6:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Background Jobs
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64
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 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.
> v<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)
> min<-0
> max<-1
> #min_max
> min_max=((35-min(v))/(max(v)-min(v)))
> print(min_max)
[1] 0.3157895
> #z-score
> m=mean(v)
> s<-12.94
> z_score=(35-m)/s
> print(z_score)
[1] -0.8844238
> #decimal scaling
> m<-35
> j=max(m)<1
> decimal_scaling=m/10^j
> print(decimal_scaling)
[1] 35
Environment History Connections Tutorial
Values
decimal_scali... 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...
z_score -0.88442383651039
Files Plots Packages Help Viewer Presentation
29°C
Mostly cloudy
Search
ENG
INTL
10:42
18-07-2024

Q7:

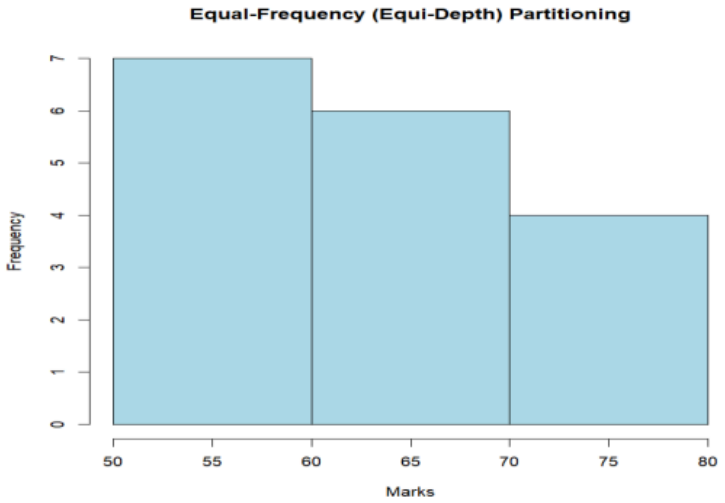

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R 4.4.1 · ~/
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64
R is free software and 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 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.
> pencils<-c(9,25,23,12,11,6,7,8,9,10)
> mean(pencils)
[1] 12
> median(pencils)
[1] 9.5
> mode=names(table(pencils))[table(pencils)==max(table(pencils))]
> mode
[1] "9"
>
Environment History Connections Tutorial
Values
mode "9"
pencils num [1:10] 9 25 23 12 11 6 7 8 9 10
Files Plots Packages Help Viewer Presentation

Q8:

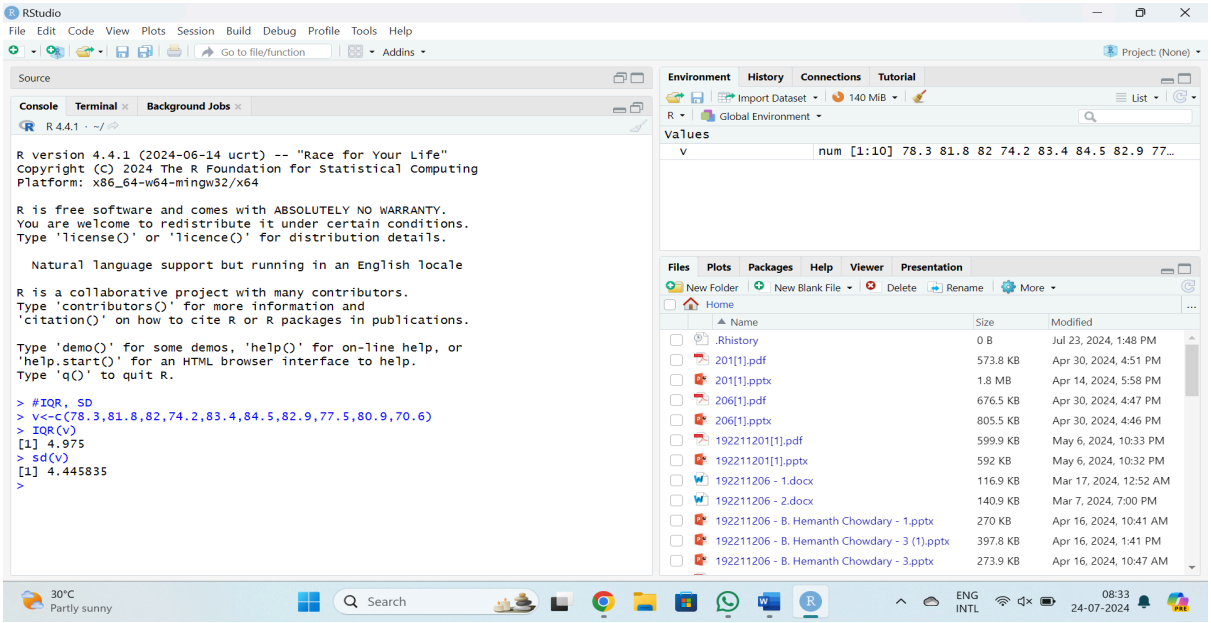

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R 4.4.1 · ~/
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64
R is free software and 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 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.
> #scatterplot
> x<-c(4,1,5,7,10,2,50,25,90,36)
> y<-c(12,5,13,19,31,7,153,72,275,110)
> scatter.smooth(x,y)
Environment History Connections Tutorial
Values
x num [1:10] 4 1 5 7 10 2 50 25 90 36
y num [1:10] 12 5 13 19 31 7 153 72 275...
Files Plots Packages Help Viewer Presentation
y
0 50 100 150 200 250
x
0 20 40 60 80
29°C
Mostly cloudy
Search
ENG
INTL
10:49
18-07-2024

Q9:

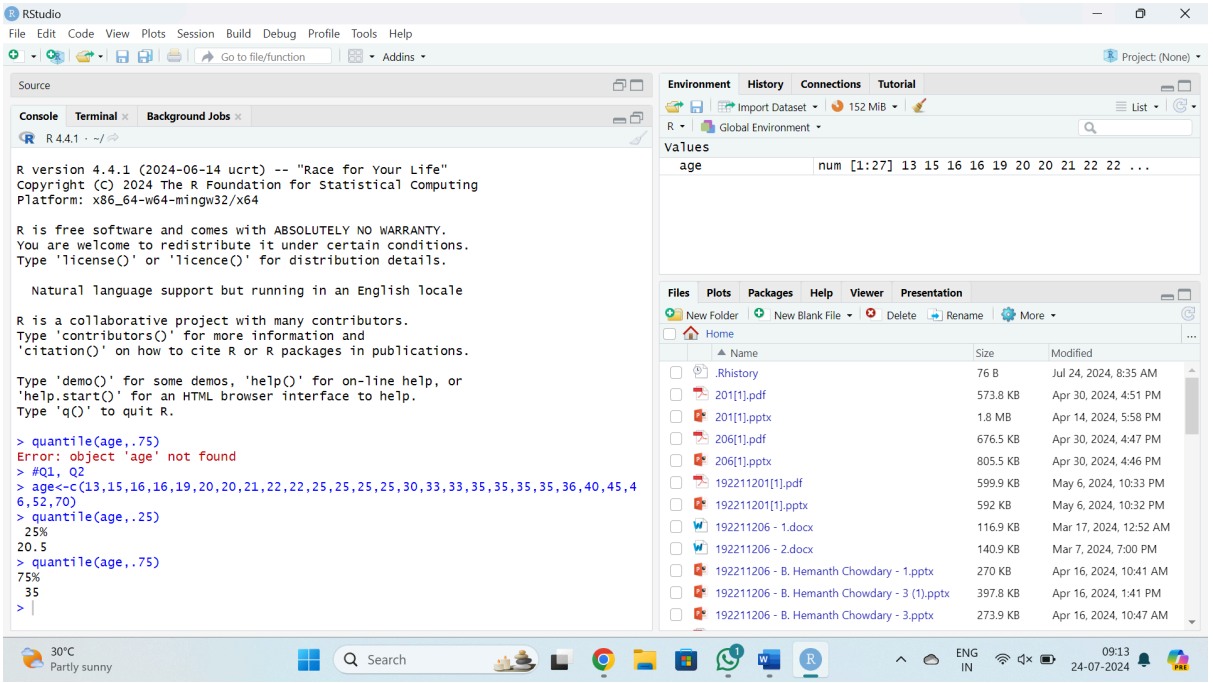
```
> print(smoothed_data_by_mean)
 [1] 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375
 [9] 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375
[17] 43.7500 43.7500 43.7500 43.7500 72.7500 72.7500 72.7500 72.7500
> print("Smoothed data by bin median:")
[1] "Smoothed data by bin median:"
> print(smoothed_data_by_median)
 [1] 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
[14] 20.0 20.0 20.0 45.0 45.0 45.0 45.0 72.5 72.5 72.5 72.5
> print("Smoothed data by bin boundaries:")
[1] "Smoothed data by bin boundaries:"
> print(smoothed_data_by_boundaries)
(10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1
          11           30           11           30           11
(10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2
          30           11           30           11           30
(10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1
          11           30           11           30           11
(10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2
          30           11           30           11           30
```



Q10:

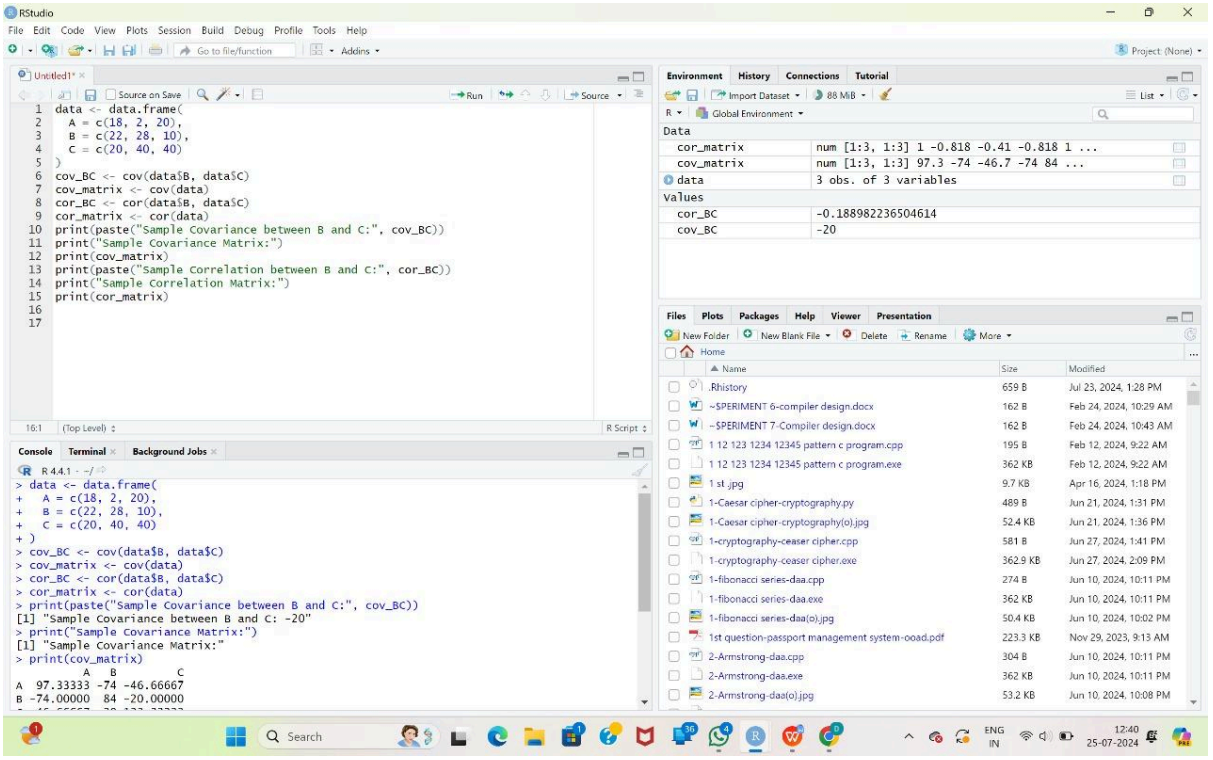


Q11:

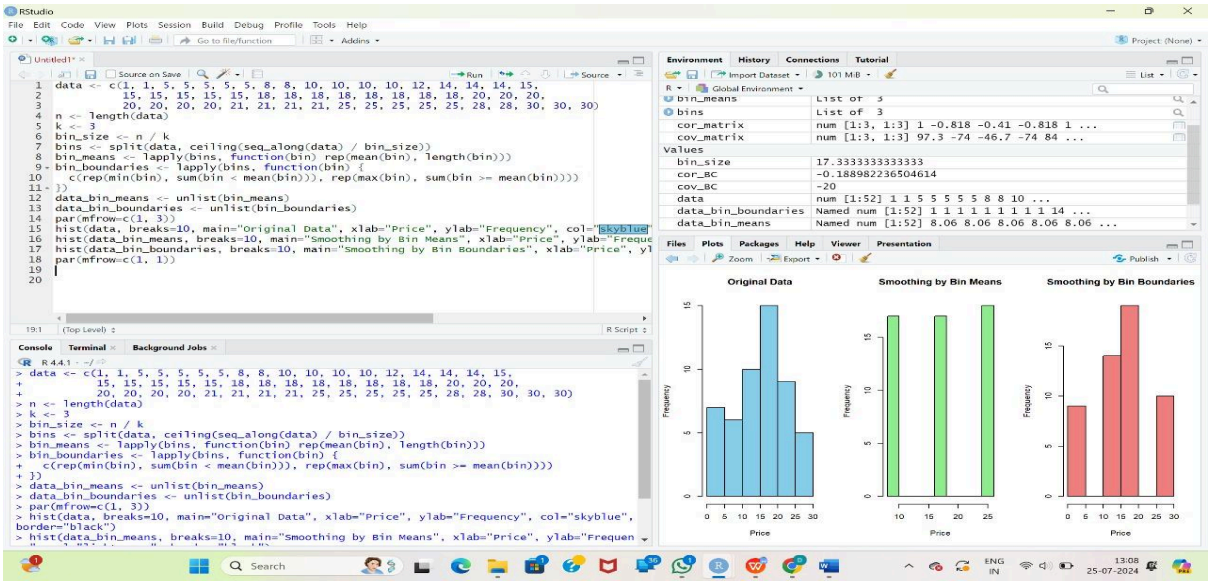

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R 4.4.1 · ~/
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64
R is free software and 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 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.
> quantile(age,.75)
Error: object 'age' not found
> #Q1, Q2
> 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 History Connections Tutorial
Import Dataset 152 MiB
Global Environment
Values
age num [1:27] 13 15 16 16 19 20 20 21 22 22 ...
Files Plots Packages Help Viewer Presentation
New Folder New Blank File Delete Rename More
Home
Name Size Modified
.Rhistory 76 B Jul 24, 2024, 8:35 AM
201[1].pdf 573.8 KB Apr 30, 2024, 4:51 PM
201[1].pptx 1.8 MB Apr 14, 2024, 5:58 PM
206[1].pdf 676.5 KB Apr 30, 2024, 4:47 PM
206[1].pptx 805.5 KB Apr 30, 2024, 4:46 PM
192211201[1].pdf 599.9 KB May 6, 2024, 10:33 PM
192211201[1].pptx 592 KB May 6, 2024, 10:32 PM
192211206 - 1.docx 116.9 KB Mar 17, 2024, 12:52 AM
192211206 - 2.docx 140.9 KB Mar 7, 2024, 7:00 PM
192211206 - B. Hemanth Chowdary - 1.pptx 270 KB Apr 16, 2024, 10:41 AM
192211206 - B. Hemanth Chowdary - 3 (1).pptx 397.8 KB Apr 16, 2024, 1:41 PM
192211206 - B. Hemanth Chowdary - 3.pptx 273.9 KB Apr 16, 2024, 10:47 AM
30°C Partly sunny
Search
ENG IN
09:13 24-07-2024

DAY-2

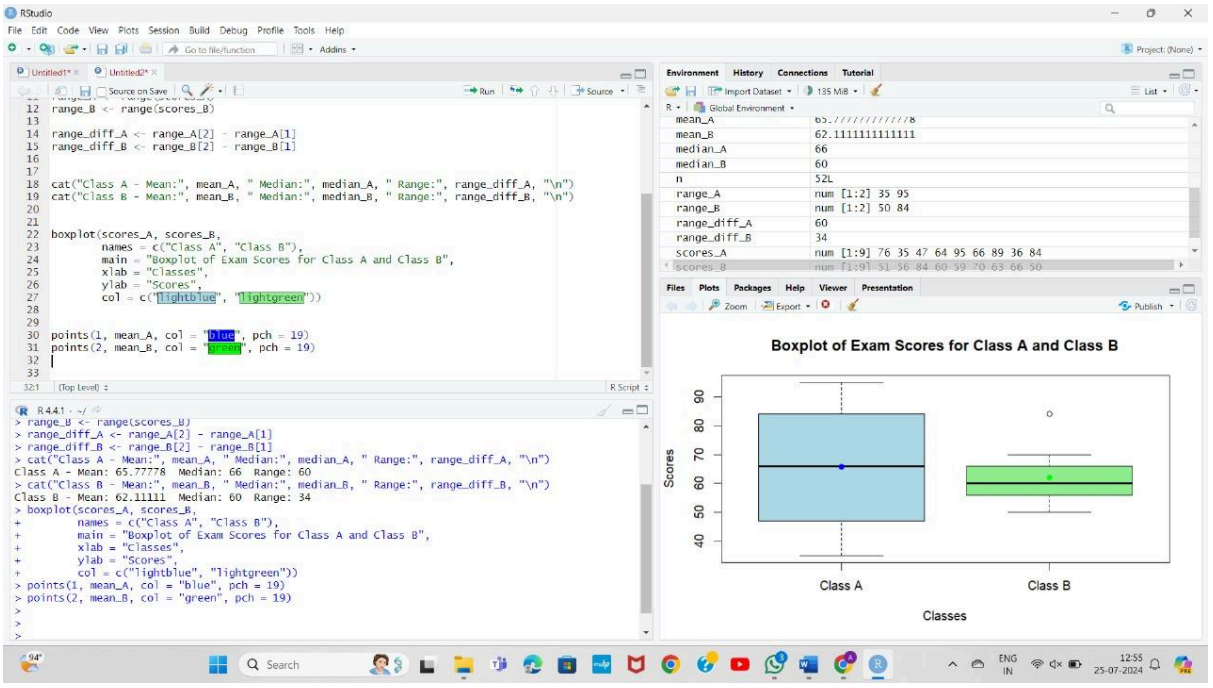
Q1:


data <- data.frame(
A = c(18, 2, 20),
B = c(22, 28, 10),
C = c(20, 40, 40)
)
cov_BC <- cov(data$B, data$C)
cov_matrix <- cov(data)
cor_BC <- cor(data$B, data$C)
cor_matrix <- cor(data)
print(paste("Sample Covariance between B and C:", cov_BC))
print("Sample Covariance Matrix:")
print(cov_matrix)
print(paste("Sample Correlation between B and C:", cor_BC))
print("Sample Correlation Matrix:")
print(cor_matrix)
[1] "Sample Covariance between B and C: -20"
[1] "Sample Covariance Matrix:"
A B C
A 97.33333 -74 -46.66667
B -74.00000 84 -20.00000

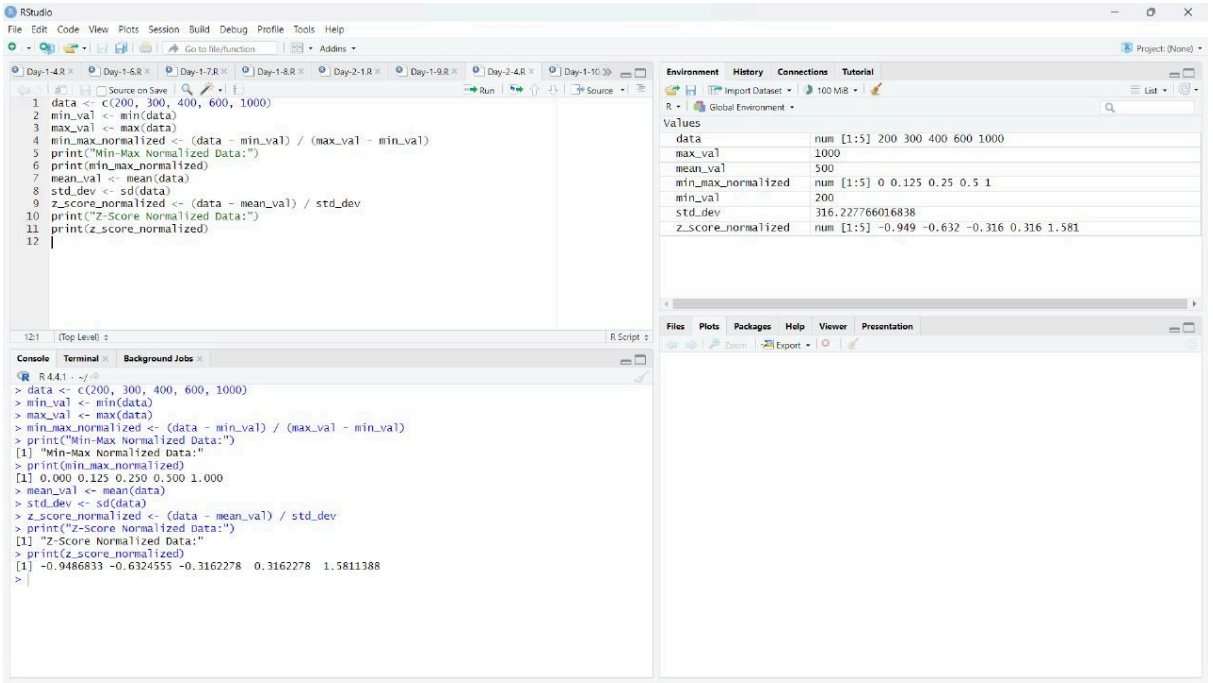
Q2:


data <- c(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, 20, 20, 20,
20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30)
n <- length(data)
k <- 3
bin_size <- n / k
bins <- split(data, ceiling(seq_along(data) / bin_size))
bin_means <- lapply(bins, function(bin) rep(mean(bin), length(bin)))
bin_boundaries <- lapply(bins, function(bin) {
c(rep(min(bin), sum(bin < mean(bin))), rep(max(bin), sum(bin >= mean(bin))))
})
data_bin_means <- unlist(bin_means)
data_bin_boundaries <- unlist(bin_boundaries)
par(mfrow=c(1, 3))
hist(data, breaks=10, main="Original Data", xlab="Price", ylab="Frequency", col="skyblue",
hist(data_bin_means, breaks=10, main="Smoothing by Bin Means", xlab="Price", ylab="Freque
hist(data_bin_boundaries, breaks=10, main="Smoothing by Bin Boundaries", xlab="Price", yl
par(mfrow=c(1, 1))
Original Data
Smoothing by Bin Means
Smoothing by Bin Boundaries
Frequency
Price

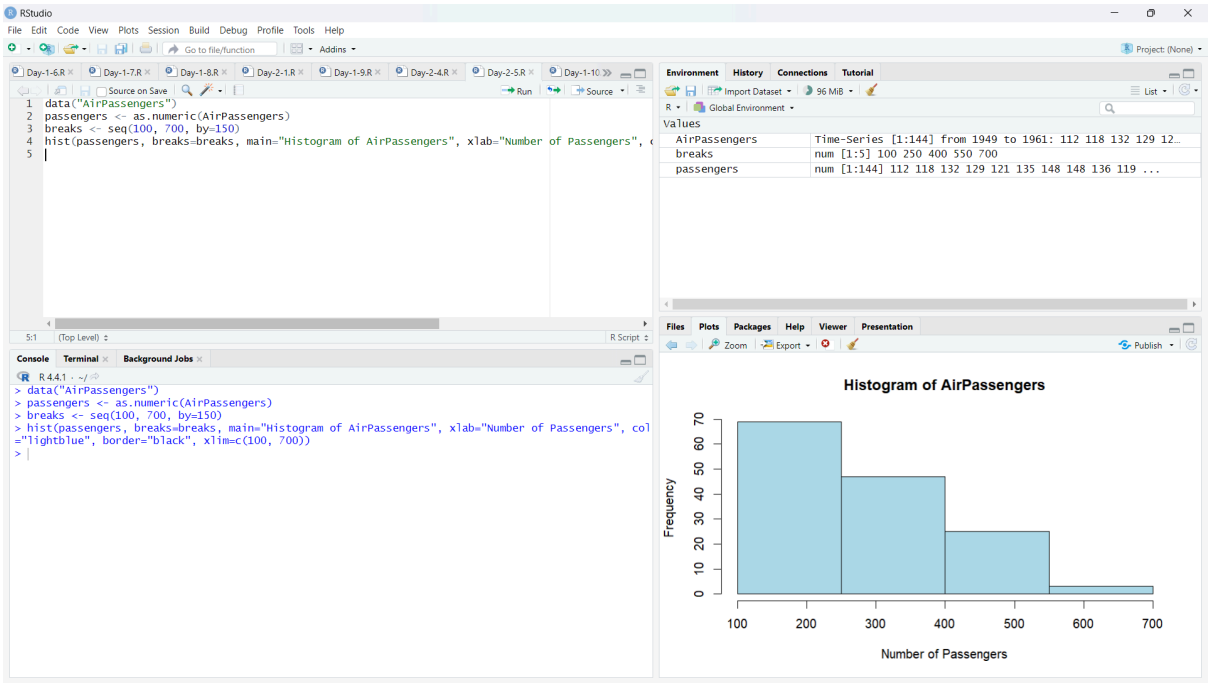
Q3:


RStudio
range_B <- range(scores_B)
range_diff_A <- range_A[2] - range_A[1]
range_diff_B <- range_B[2] - range_B[1]
cat("Class A - Mean:", mean_A, " Median:", median_A, " Range:", range_diff_A, "\n")
cat("Class B - Mean:", mean_B, " Median:", median_B, " Range:", range_diff_B, "\n")
boxplot(scores_A, scores_B,
names = c("Class A", "Class B"),
main = "Boxplot of Exam Scores for Class A and Class B",
xlab = "Classes",
ylab = "Scores",
col = c("lightblue", "lightgreen"))
points(1, mean_A, col = "blue", pch = 19)
points(2, mean_B, col = "green", pch = 19)
mean_B 62.1111111111111
median_A 66
median_B 60
n 52L
range_A num [1:2] 35 95
range_B num [1:2] 50 84
range_diff_A 60
range_diff_B 34
scores_A num [1:9] 76 35 47 64 95 66 89 36 84
Boxplot of Exam Scores for Class A and Class B
Scores
Class A
Class B
Classes

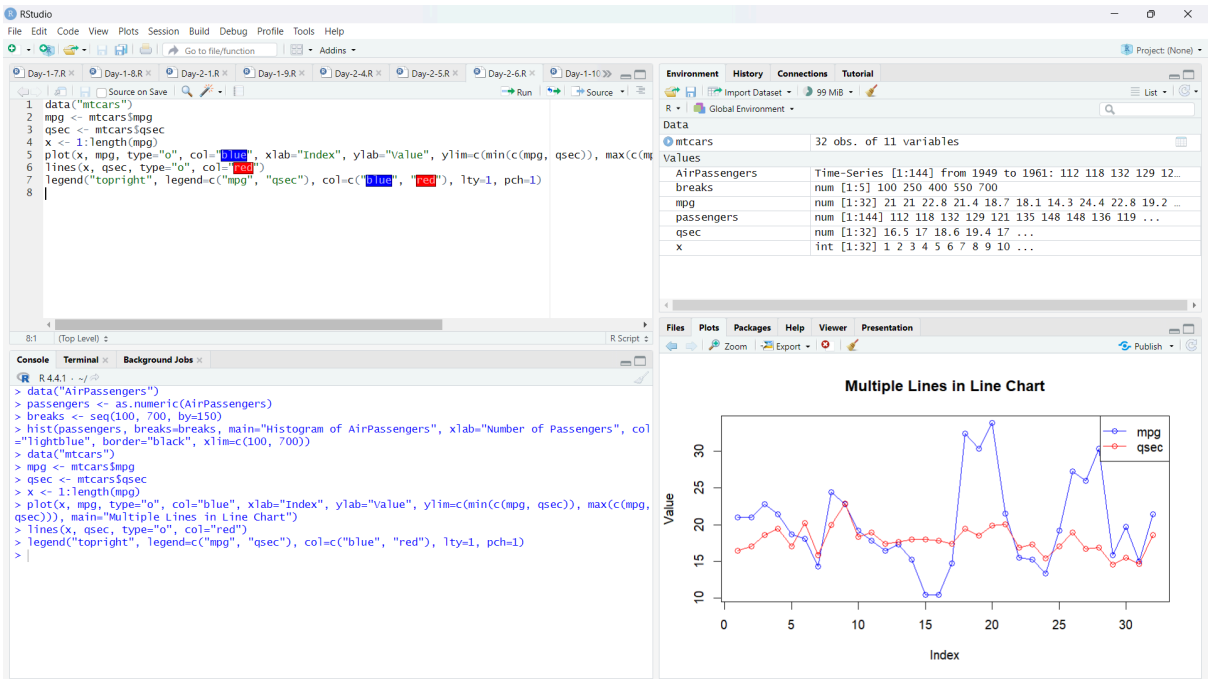
Q4:


RStudio
data <- c(200, 300, 400, 600, 1000)
min_val <- min(data)
max_val <- max(data)
min_max_normalized <- (data - min_val) / (max_val - min_val)
print("Min-Max Normalized Data:")
print(min_max_normalized)
mean_val <- mean(data)
std_dev <- sd(data)
z_score_normalized <- (data - mean_val) / std_dev
print("Z-Score Normalized Data:")
print(z_score_normalized)
[1] "Min-Max Normalized Data:"
[1] 0.000 0.125 0.250 0.500 1.000
[1] "Z-Score Normalized Data:"
[1] -0.9486833 -0.6324555 -0.3162278 0.3162278 1.5811388
data num [1:5] 200 300 400 600 1000
max_val 1000
mean_val 500
min_max_normalized num [1:5] 0 0.125 0.25 0.5 1
min_val 200
std_dev 316.227766016838
z_score_normalized num [1:5] -0.949 -0.632 -0.316 0.316 1.581

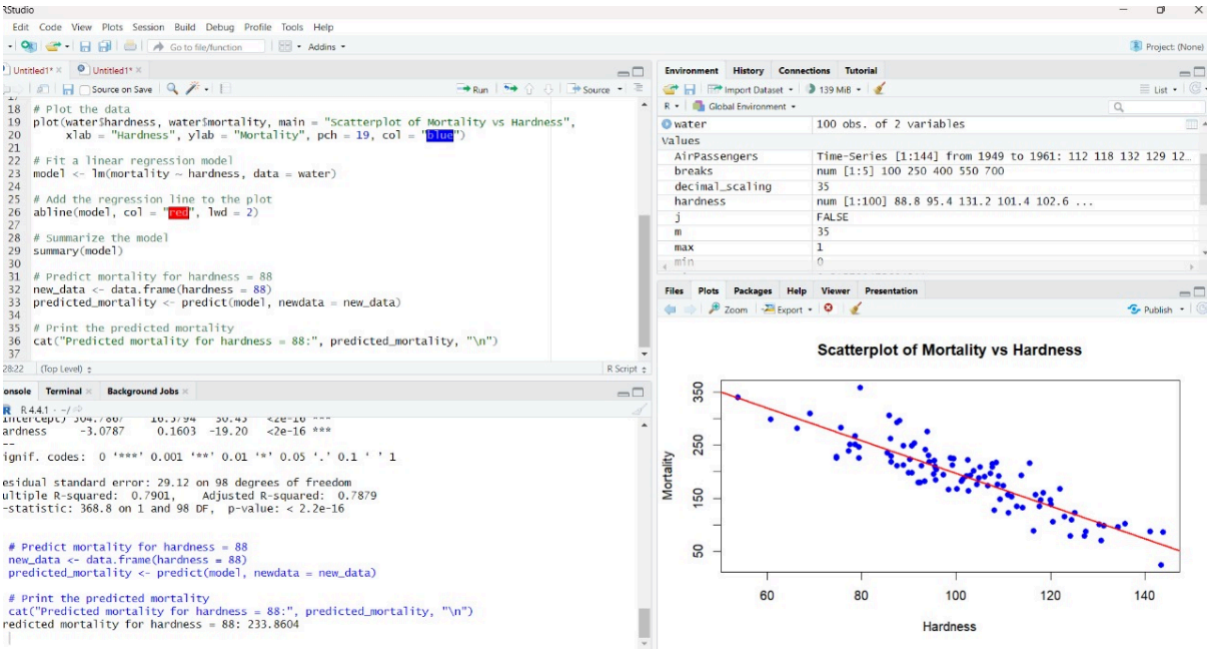
Q5:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Day-1-6.R Day-1-7.R Day-1-8.R Day-2-1.R Day-1-9.R Day-2-4.R Day-2-5.R Day-1-10
Source on Save Run Source
1 data("AirPassengers")
2 passengers <- as.numeric(AirPassengers)
3 breaks <- seq(100, 700, by=150)
4 hist(passengers, breaks=breaks, main="Histogram of AirPassengers", xlab="Number of Passengers", c
5
5:1 (Top Level) R Script
Console Terminal Background Jobs
R 4.4.1 · ~/
> data("AirPassengers")
> passengers <- as.numeric(AirPassengers)
> breaks <- seq(100, 700, by=150)
> hist(passengers, breaks=breaks, main="Histogram of AirPassengers", xlab="Number of Passengers", col
="lightblue", border="black", xlim=c(100, 700))
>
Environment History Connections Tutorial
Import Dataset 96 MiB List
R Global Environment
Values
AirPassengers Time-Series [1:144] from 1949 to 1961: 112 118 132 129 12…
breaks num [1:5] 100 250 400 550 700
passengers num [1:144] 112 118 132 129 121 135 148 148 136 119 ...
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
Histogram of AirPassengers
Frequency
0 10 20 30 40 50 60 70
100 200 300 400 500 600 700
Number of Passengers

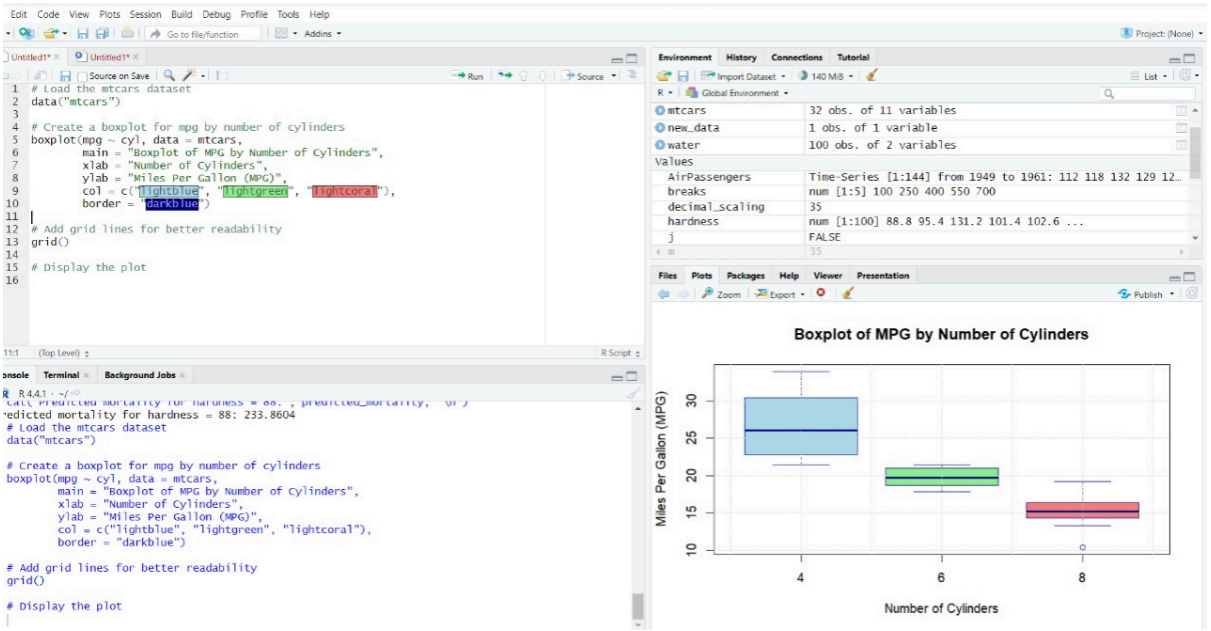
Q6:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Day-1-7.R Day-1-8.R Day-2-1.R Day-1-9.R Day-2-4.R Day-2-5.R Day-2-6.R Day-1-10
Source on Save Run Source
1 data("mtcars")
2 mpg <- mtcars$mpg
3 qsec <- mtcars$qsec
4 x <- 1:length(mpg)
5 plot(x, mpg, type="o", col="blue", xlab="Index", ylab="Value", ylim=c(min(c(mpg, qsec)), max(c(mp
6 lines(x, qsec, type="o", col="red")
7 legend("topright", legend=c("mpg", "qsec"), col=c("blue", "red"), lty=1, pch=1)
8
8:1 (Top Level) R Script
Console Terminal Background Jobs
R 4.4.1 · ~/
> data("AirPassengers")
> passengers <- as.numeric(AirPassengers)
> breaks <- seq(100, 700, by=150)
> hist(passengers, breaks=breaks, main="Histogram of AirPassengers", xlab="Number of Passengers", col
="lightblue", border="black", xlim=c(100, 700))
> data("mtcars")
> mpg <- mtcars$mpg
> qsec <- mtcars$qsec
> x <- 1:length(mpg)
> plot(x, mpg, type="o", col="blue", xlab="Index", ylab="Value", ylim=c(min(c(mpg, qsec)), max(c(mpg,
qsec))), main="Multiple Lines in Line Chart")
> lines(x, qsec, type="o", col="red")
> legend("topright", legend=c("mpg", "qsec"), col=c("blue", "red"), lty=1, pch=1)
>
Environment History Connections Tutorial
Import Dataset 99 MiB List
R Global Environment
Data
mtcars 32 obs. of 11 variables
Values
AirPassengers Time-Series [1:144] from 1949 to 1961: 112 118 132 129 12…
breaks num [1:5] 100 250 400 550 700
mpg num [1:32] 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 …
passengers num [1:144] 112 118 132 129 121 135 148 148 136 119 ...
qsec num [1:32] 16.5 17 18.6 19.4 17 ...
x int [1:32] 1 2 3 4 5 6 7 8 9 10 ...
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
Multiple Lines in Line Chart
mpg
qsec
Value
10 15 20 25 30
0 5 10 15 20 25 30
Index

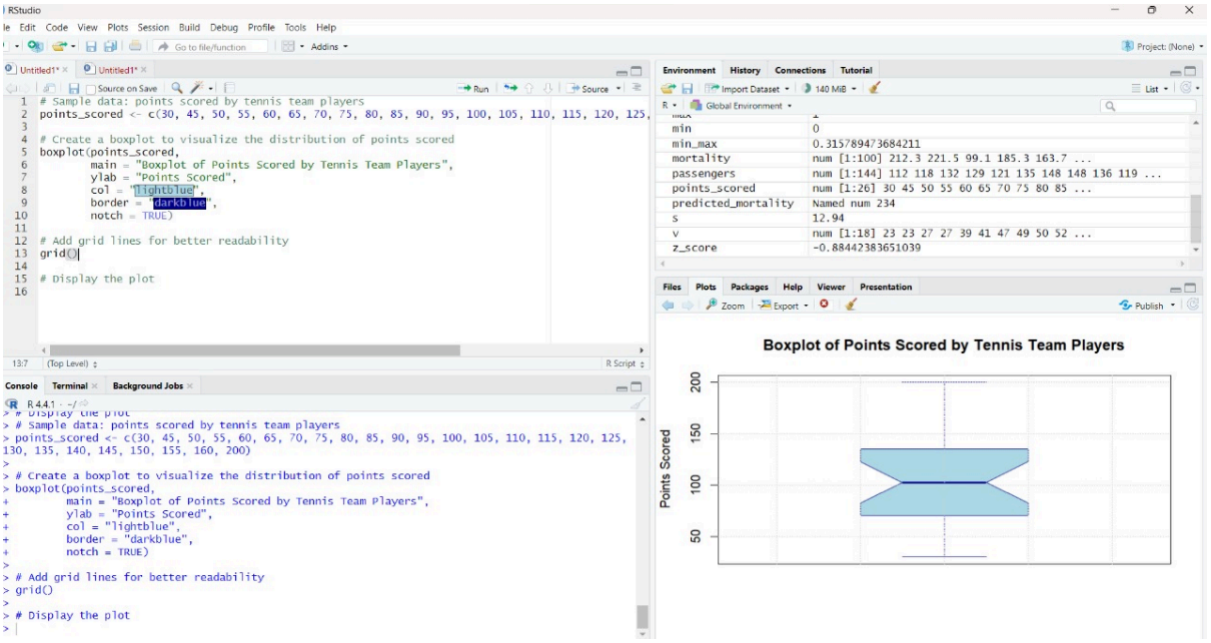
Q7:


Scatterplot of Mortality vs Hardness
Mortality
Hardness
50
150
250
350
60
80
100
120
140

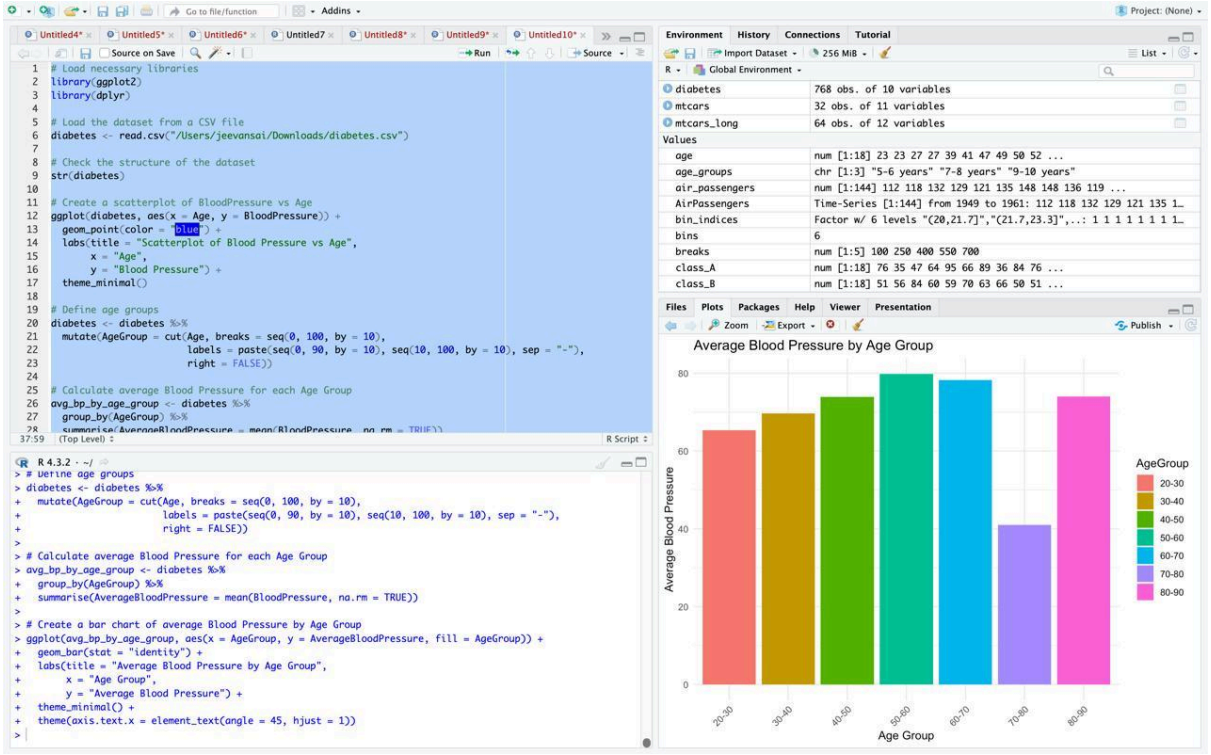
Q8:


Boxplot of MPG by Number of Cylinders
Miles Per Gallon (MPG)
Number of Cylinders
10
15
20
25
30
4
6
8

Q9:

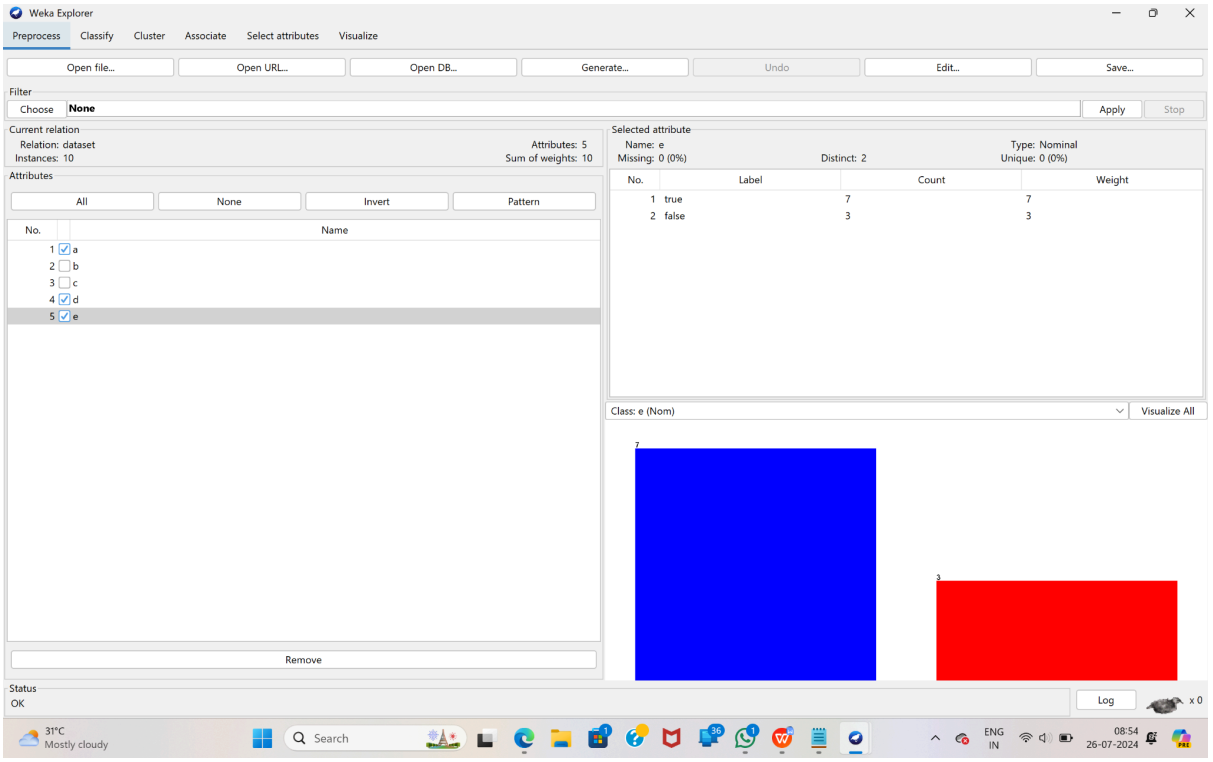

Boxplot of Points Scored by Tennis Team Players
Points Scored
200
150
100
50

Q10:

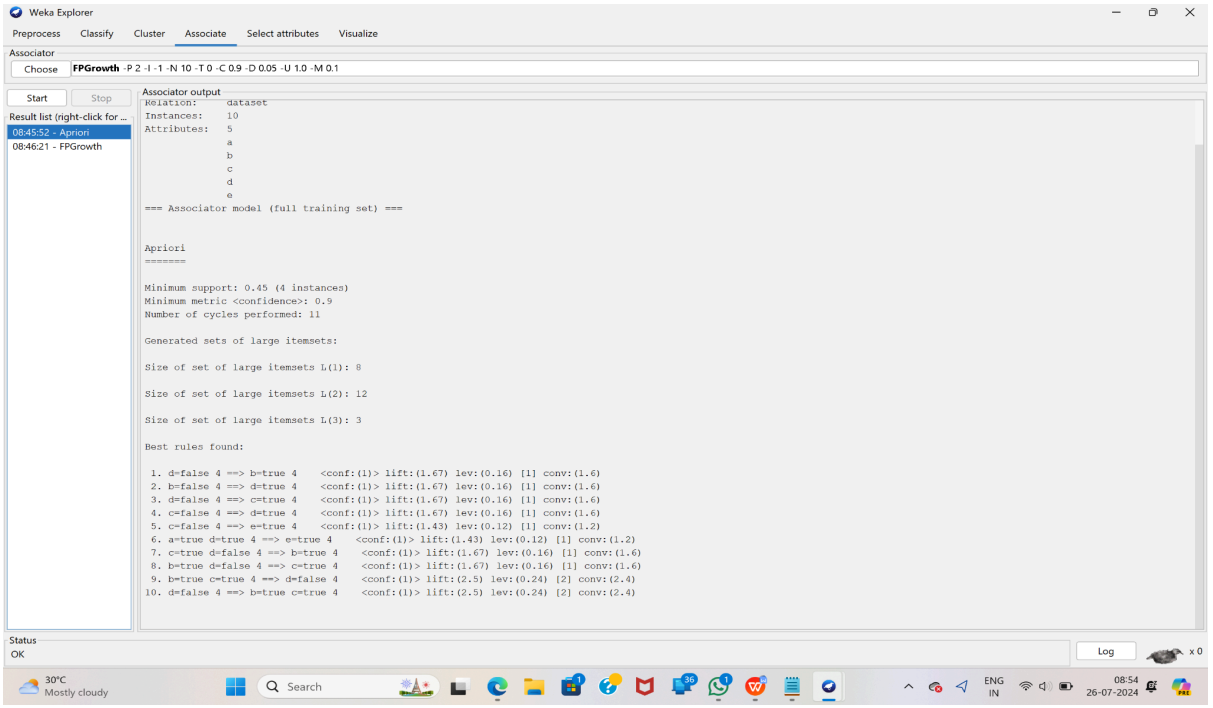

Average Blood Pressure by Age Group
Average Blood Pressure
Age Group
AgeGroup
20-30
30-40
40-50
50-60
60-70
70-80
80-90

DAY-3

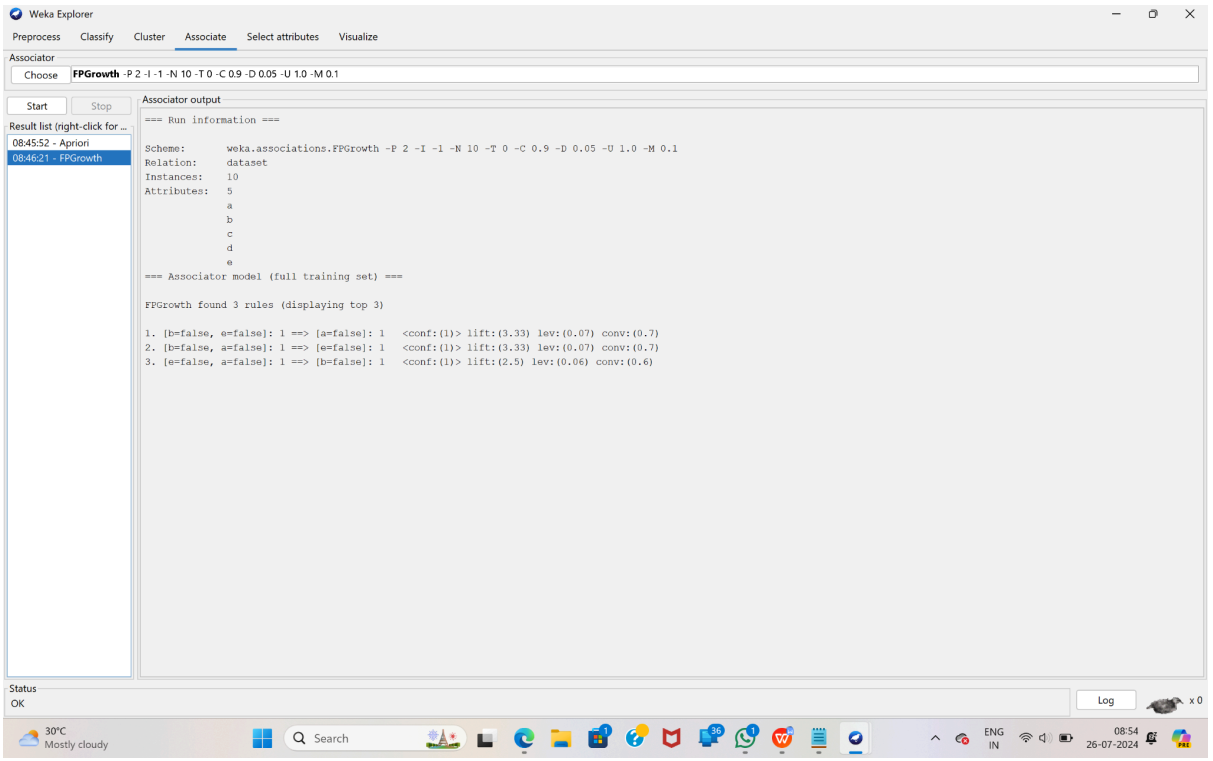
Q1:



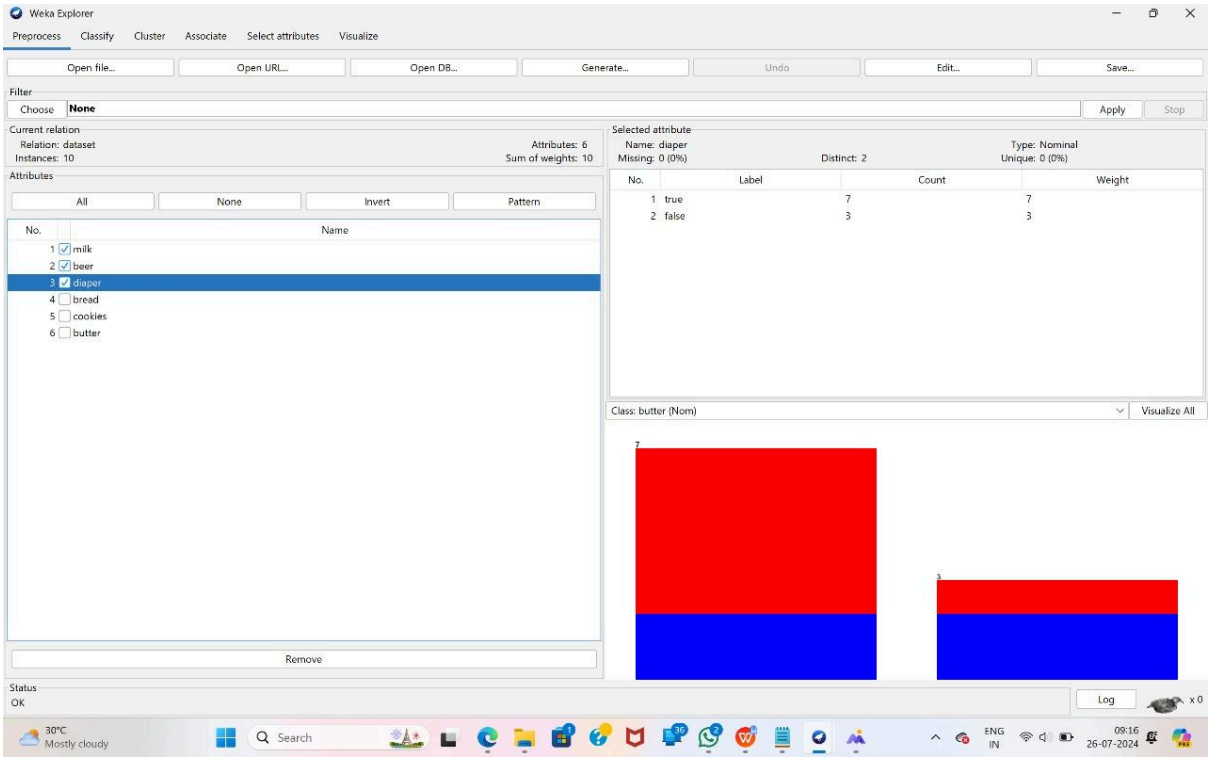
Apriori Algorithm:



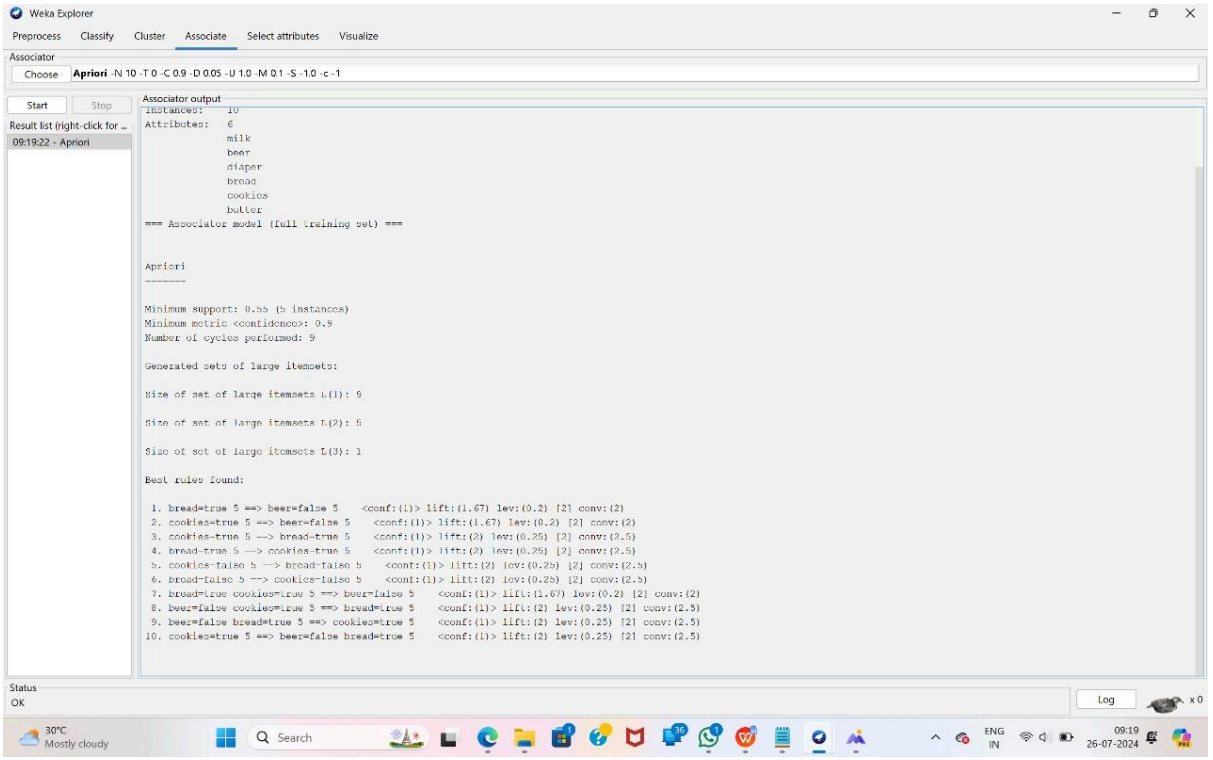
FP GROWTH:


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Associator
Choose FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Start Stop
Result list (right-click for ...
08:45:52 - Apriori
08:46:21 - FPGrowth
Associator output
=== Run information ===
Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation: dataset
Instances: 10
Attributes: 5
a
b
c
d
e
=== Associator model (full training set) ===
FPGrowth found 3 rules (displaying top 3)
1. [b=false, e=false]: 1 ==> [a=false]: 1 <conf:(1)> lift:(3.33) lev:(0.07) conv:(0.7)
2. [b=false, a=false]: 1 ==> [e=false]: 1 <conf:(1)> lift:(3.33) lev:(0.07) conv:(0.7)
3. [e=false, a=false]: 1 ==> [b=false]: 1 <conf:(1)> lift:(2.5) lev:(0.06) conv:(0.6)
Status
OK
Log
x 0
30°C
Mostly cloudy
Search
ENG
IN
08:54
26-07-2024

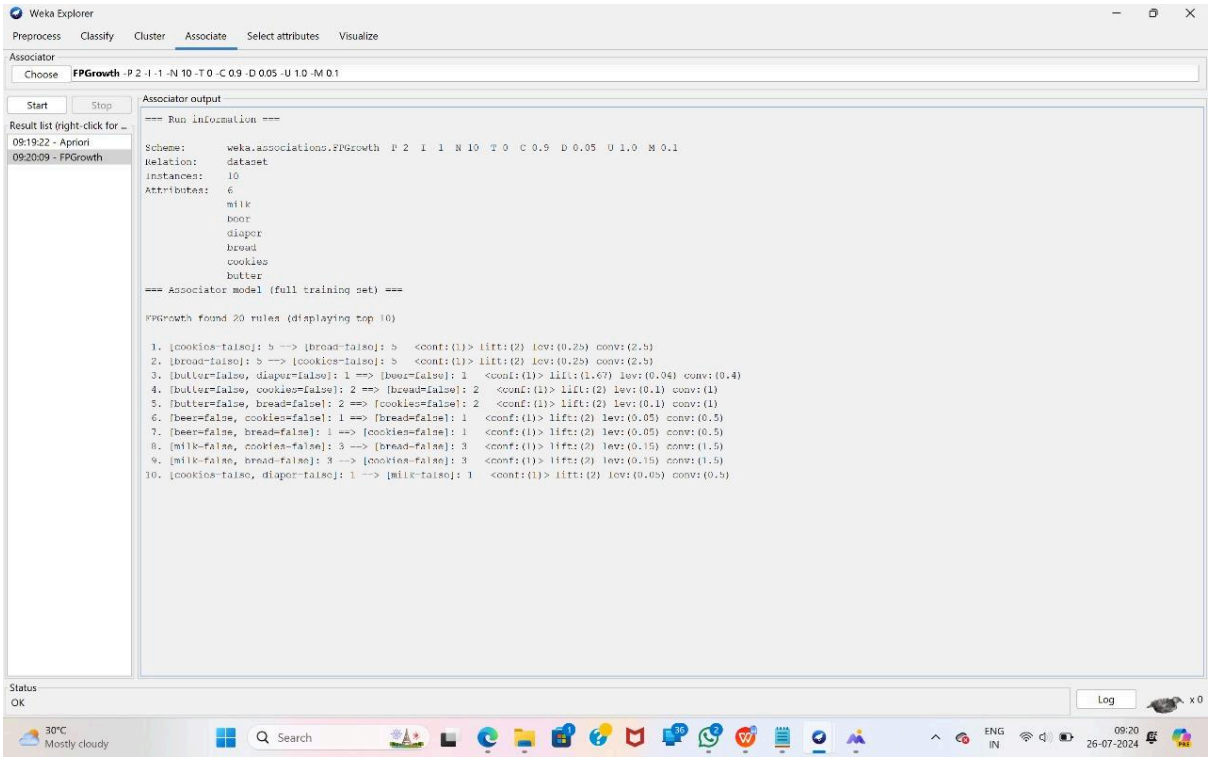
Q2:


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Choose None Apply Stop
Current relation
Relation: dataset
Instances: 10
Attributes: 6
Sum of weights: 10
Selected attribute
Name: diaper
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)
Attributes
All None Invert Pattern
No. Name
1 milk
2 beer
3 diaper
4 bread
5 cookies
6 butter
No. Label Count Weight
1 true 7 7
2 false 3 3
Class: butter (Nom)
Visualize All
Remove
Status
OK
Log
x 0
30°C
Mostly cloudy
Search
ENG
IN
09:16
26-07-2024

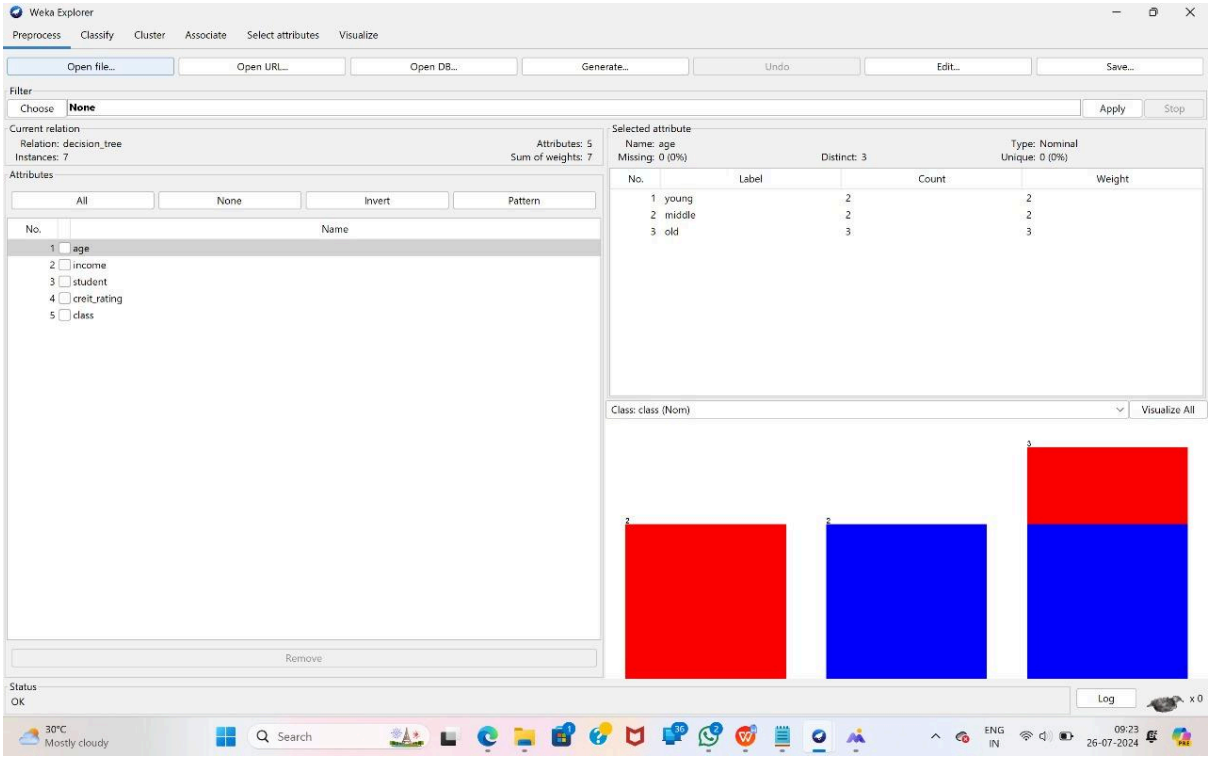
Apriori Algorithm:



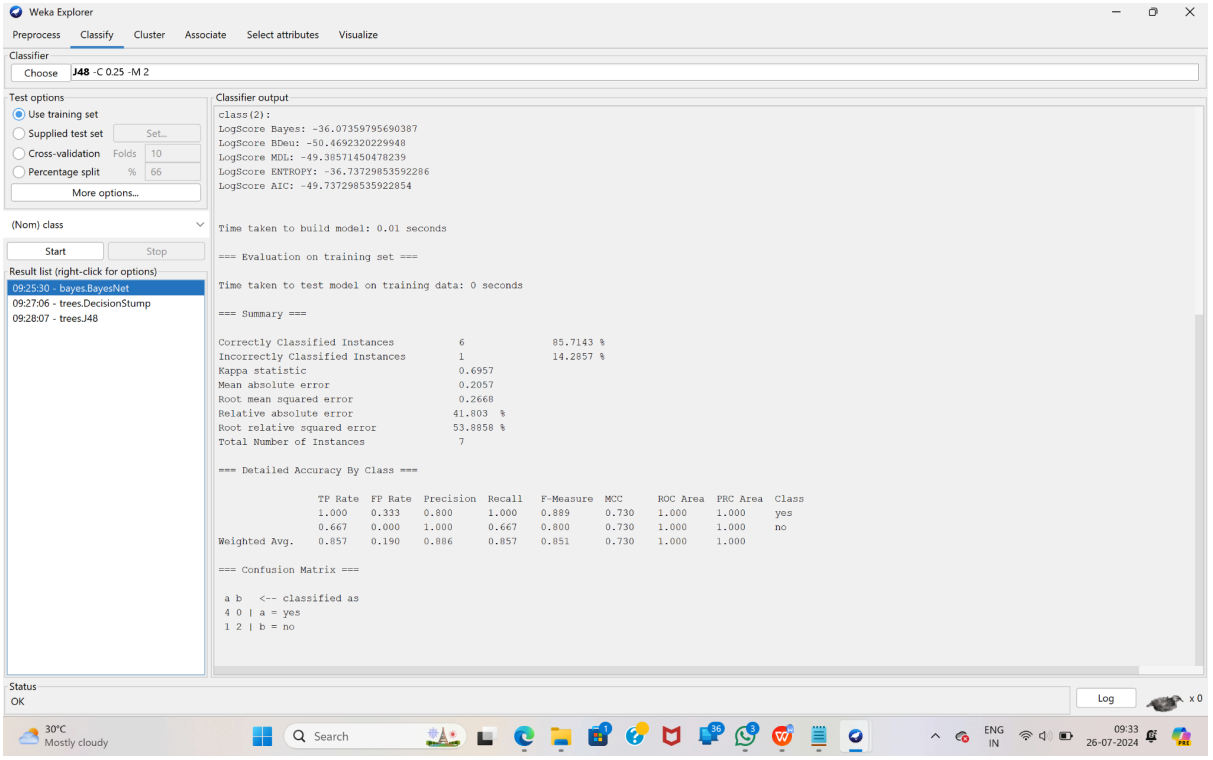
FP GROWTH:



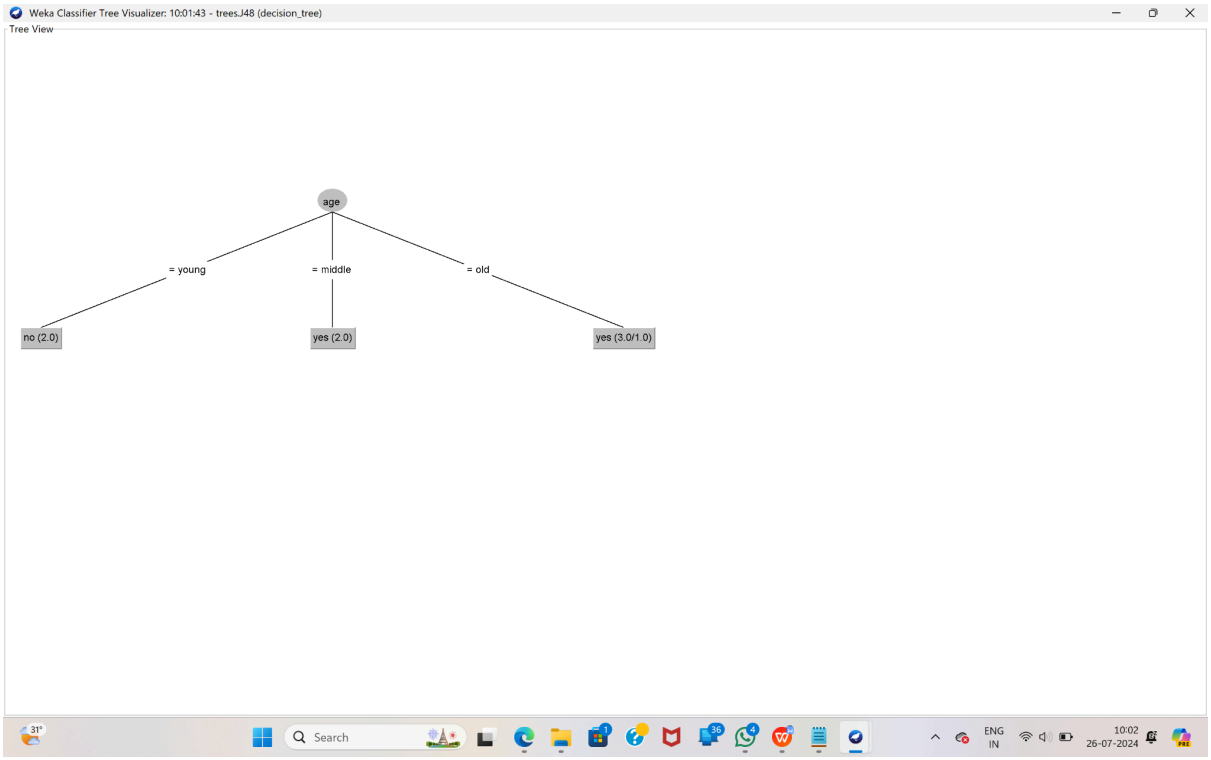
Q3:



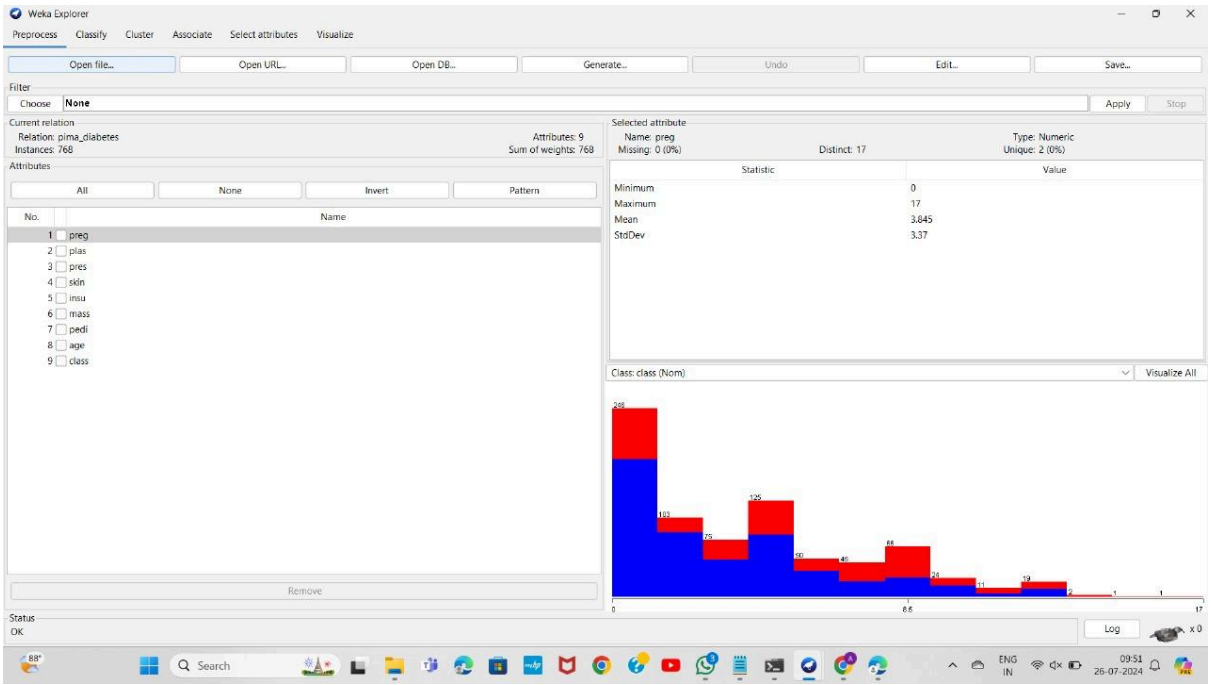
Bayes classification:



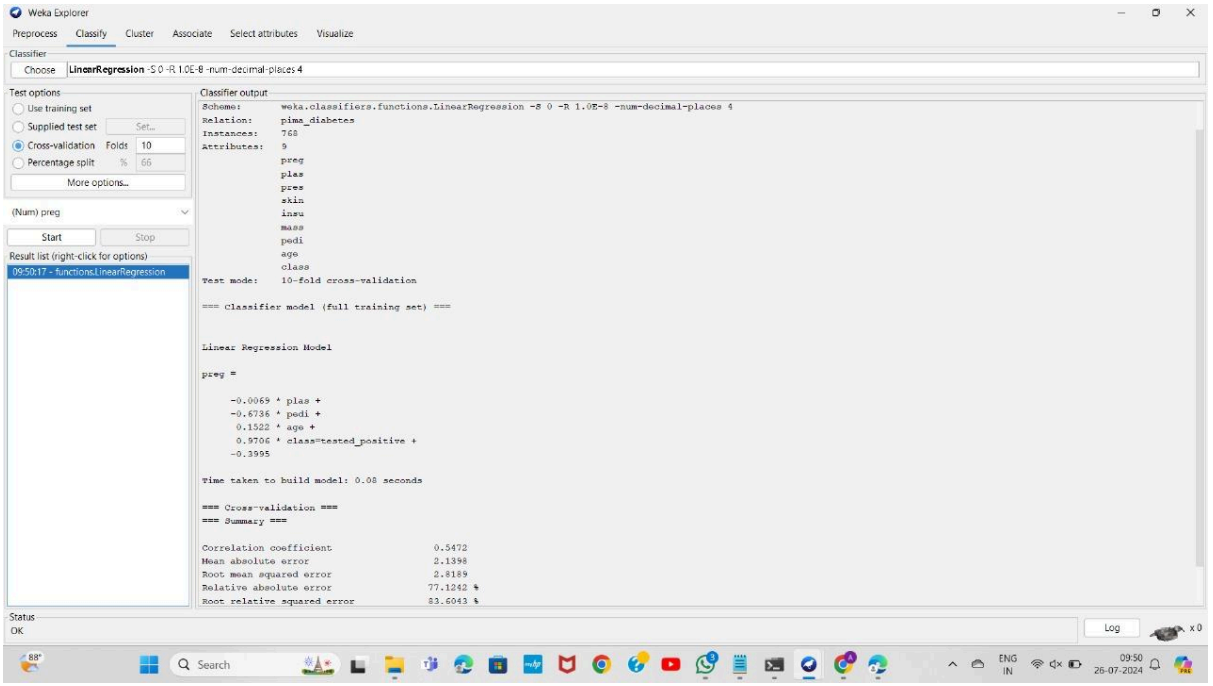
Descion tree:



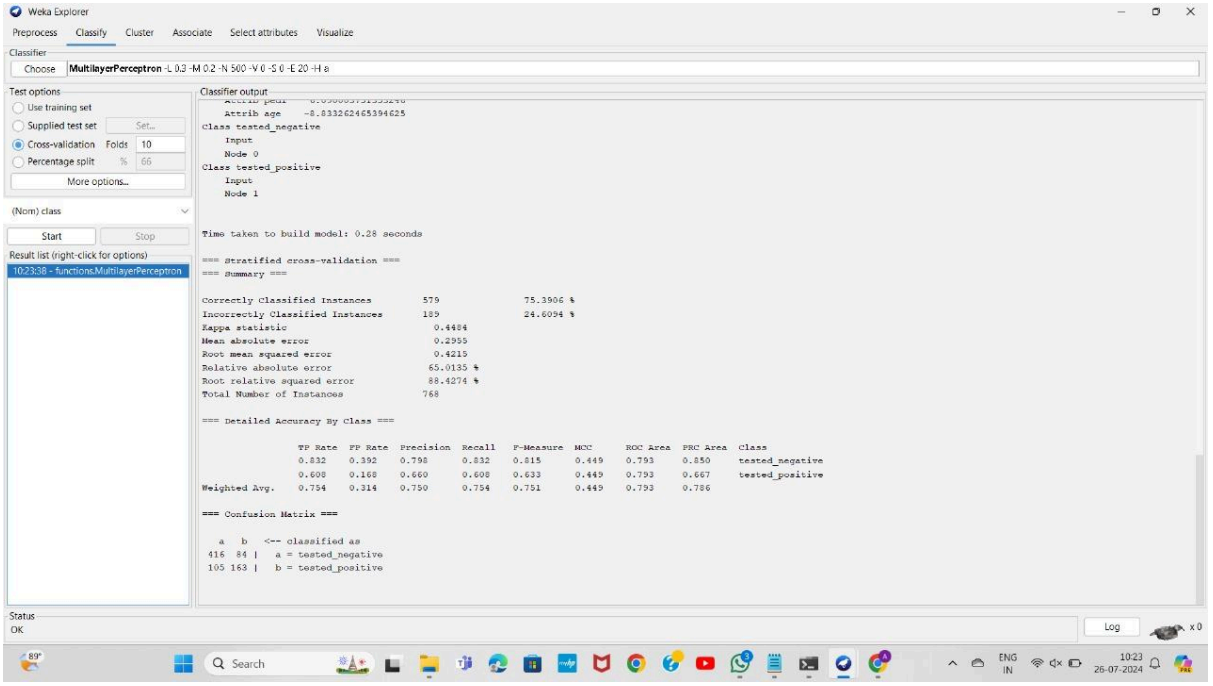
Q4:



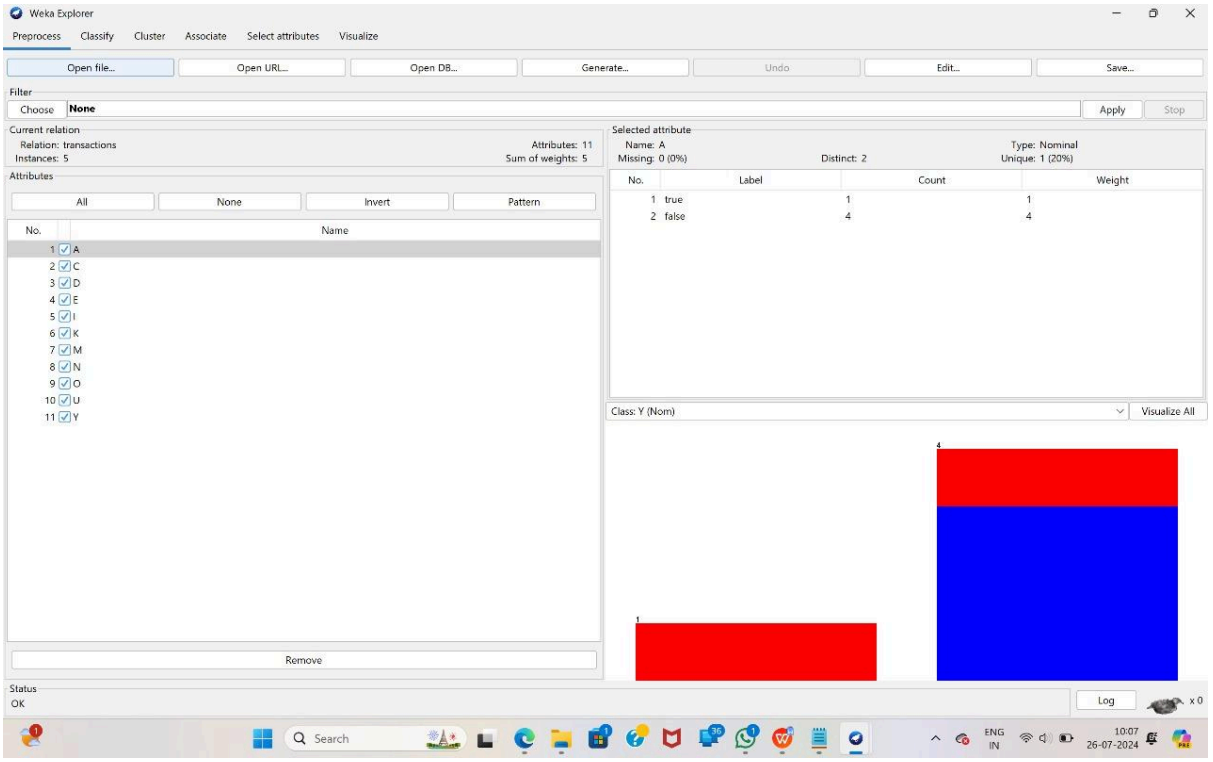
Linear regression:



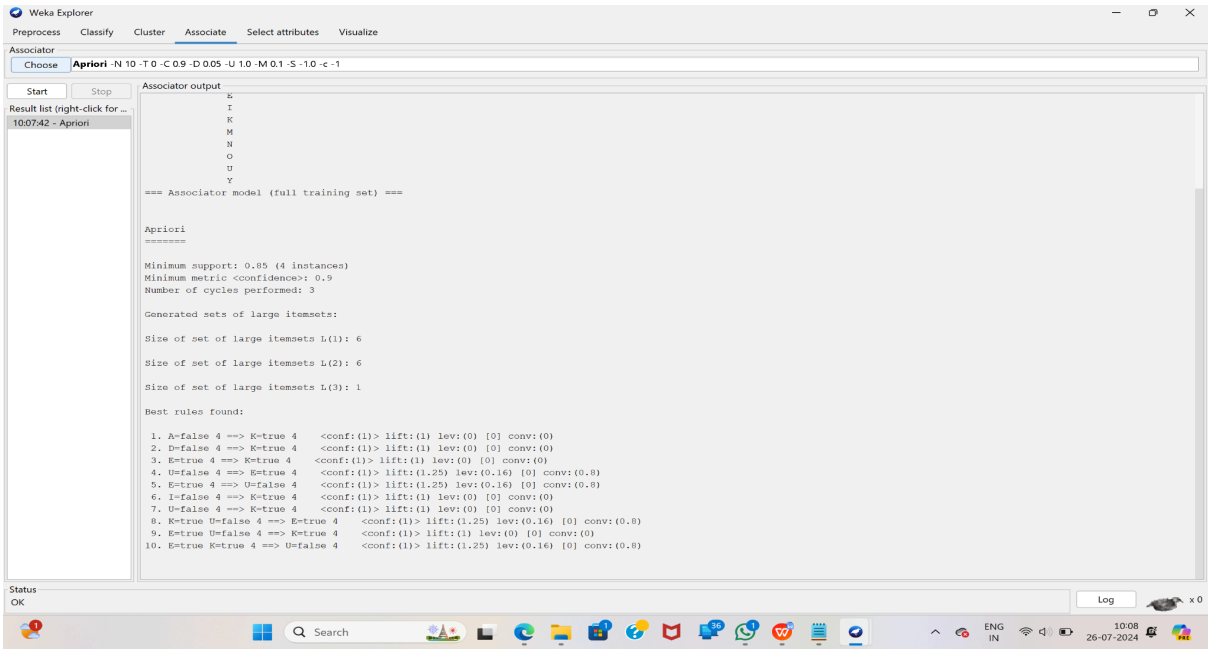
Multiple regression:



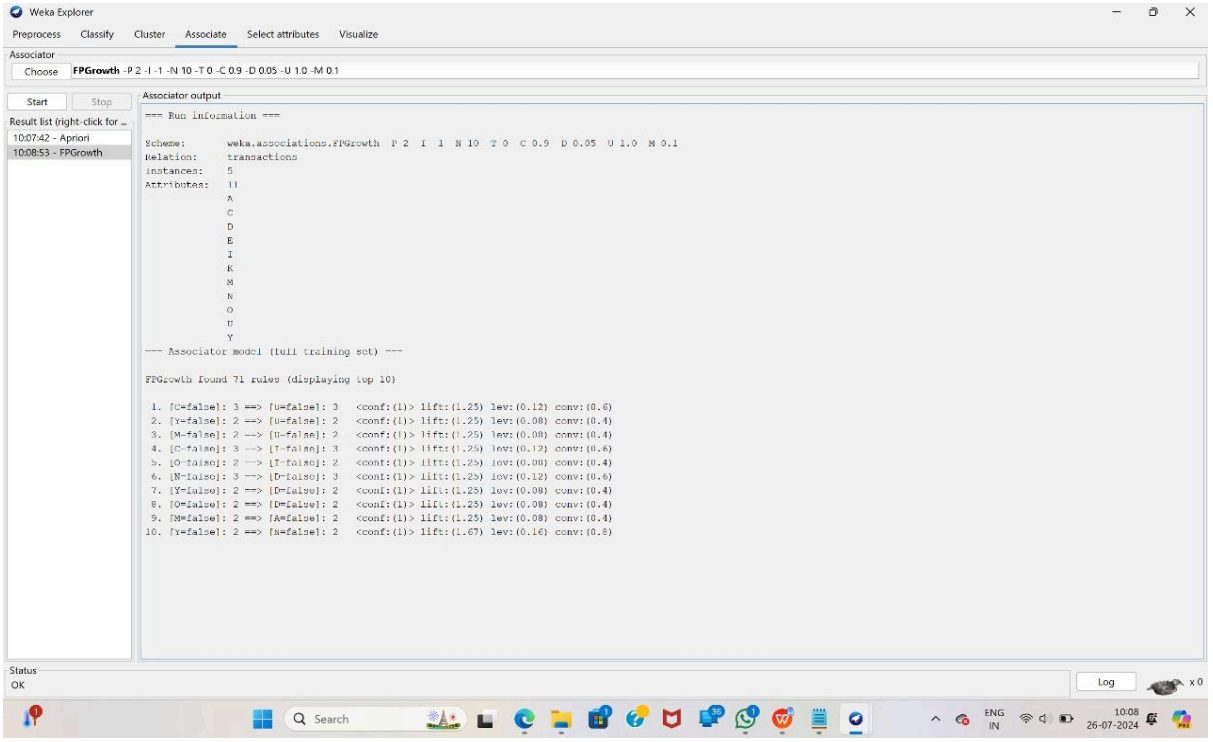
Q5:



Apriori Algorithm:

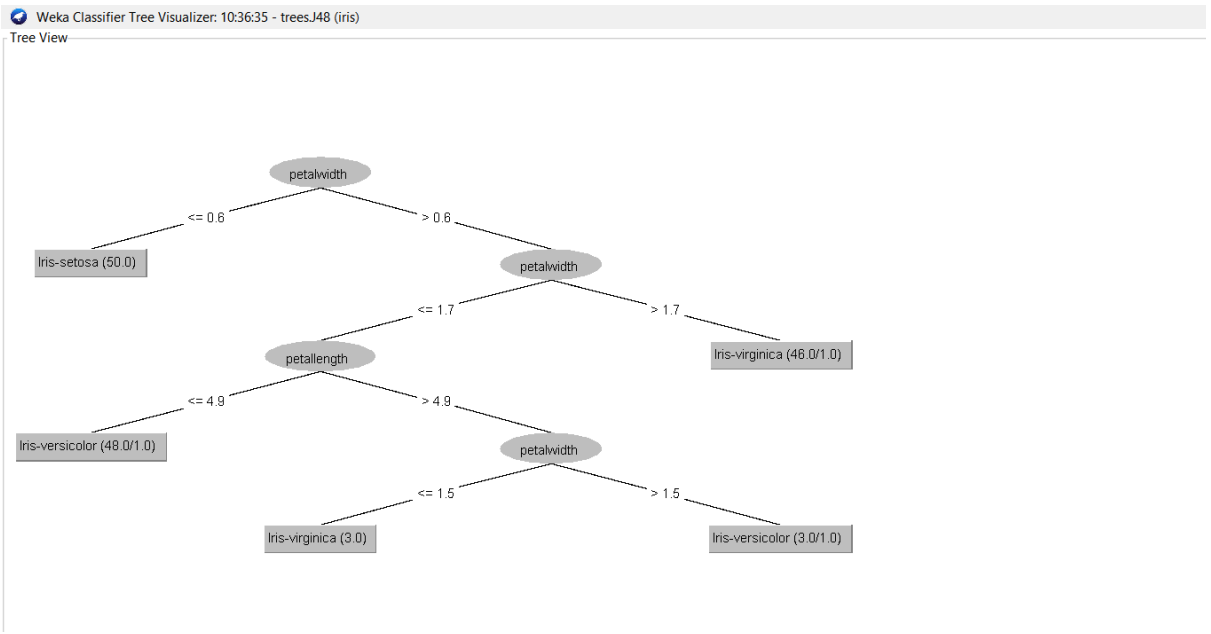


FP GROWTH:

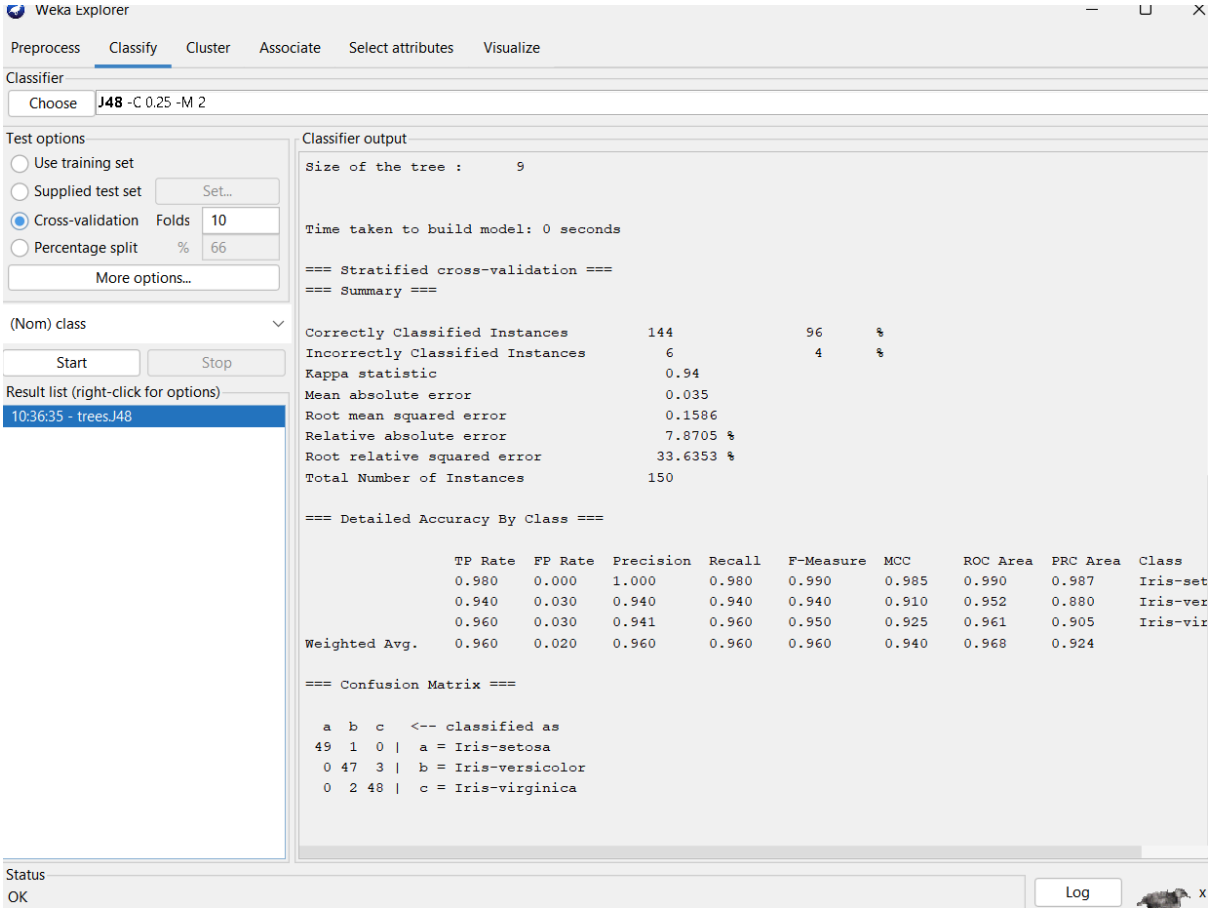


Q6:

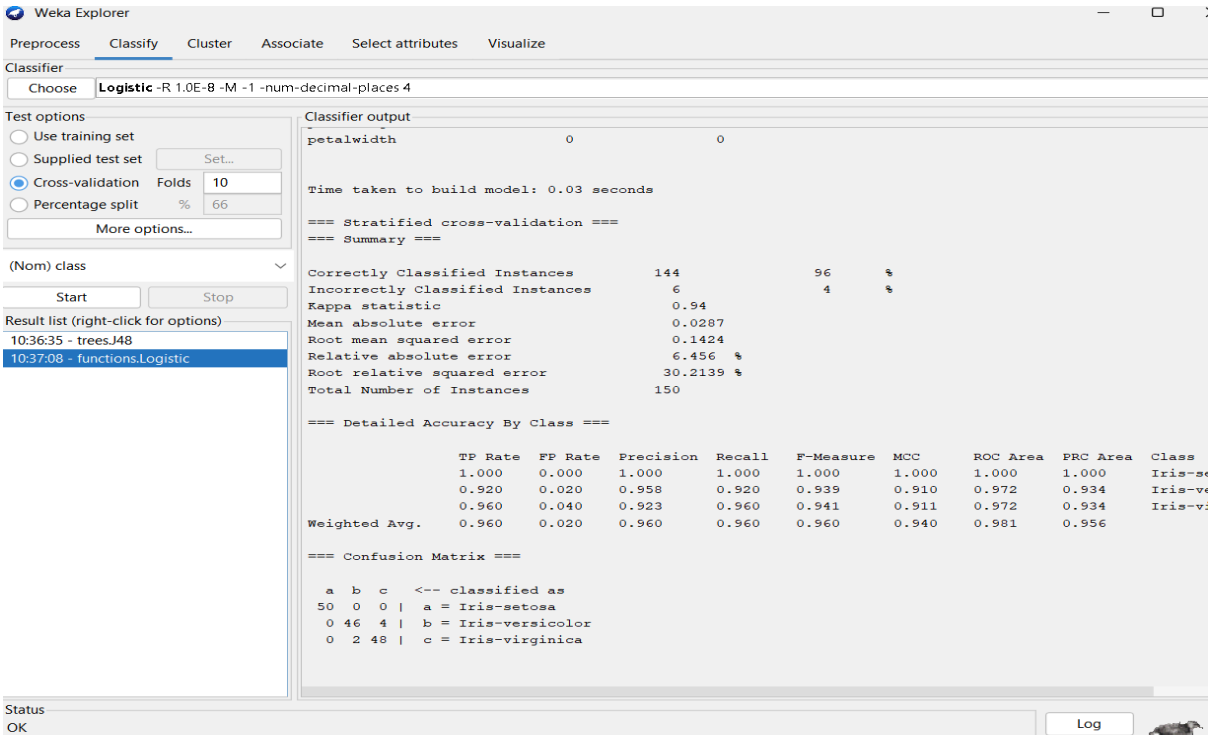
TREE:



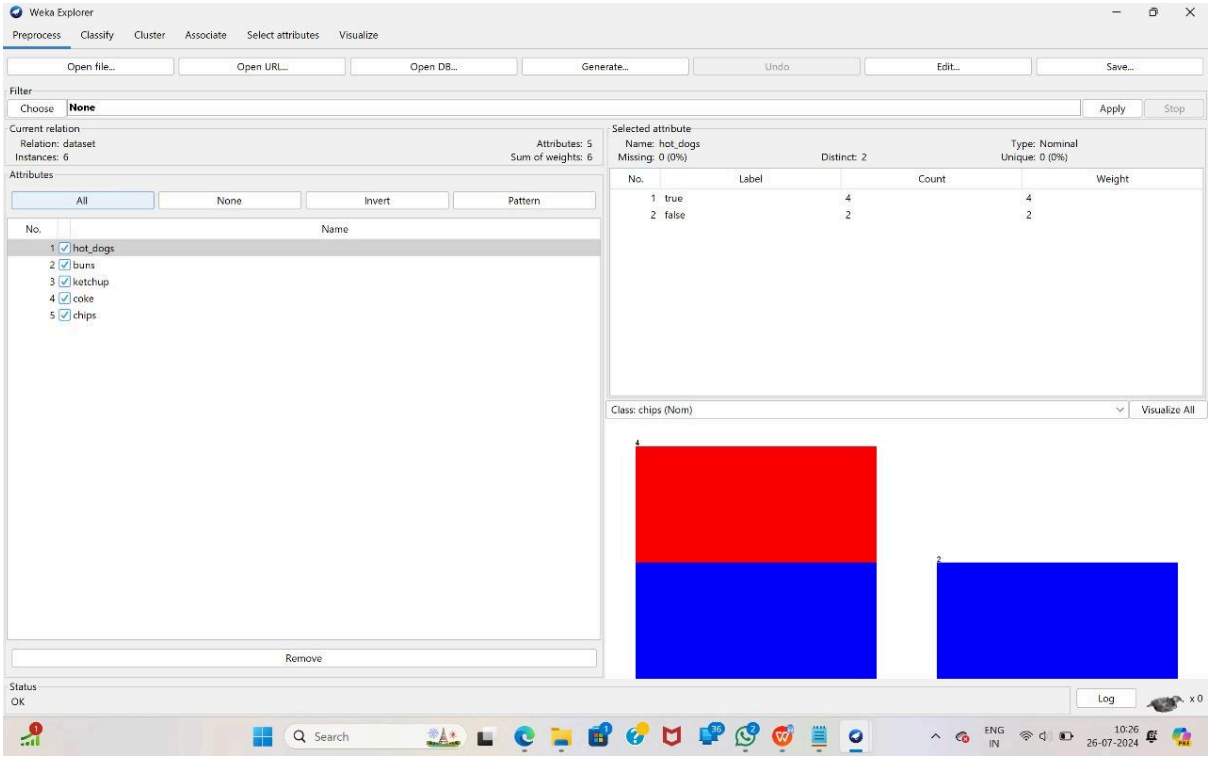
PREPROCESS:



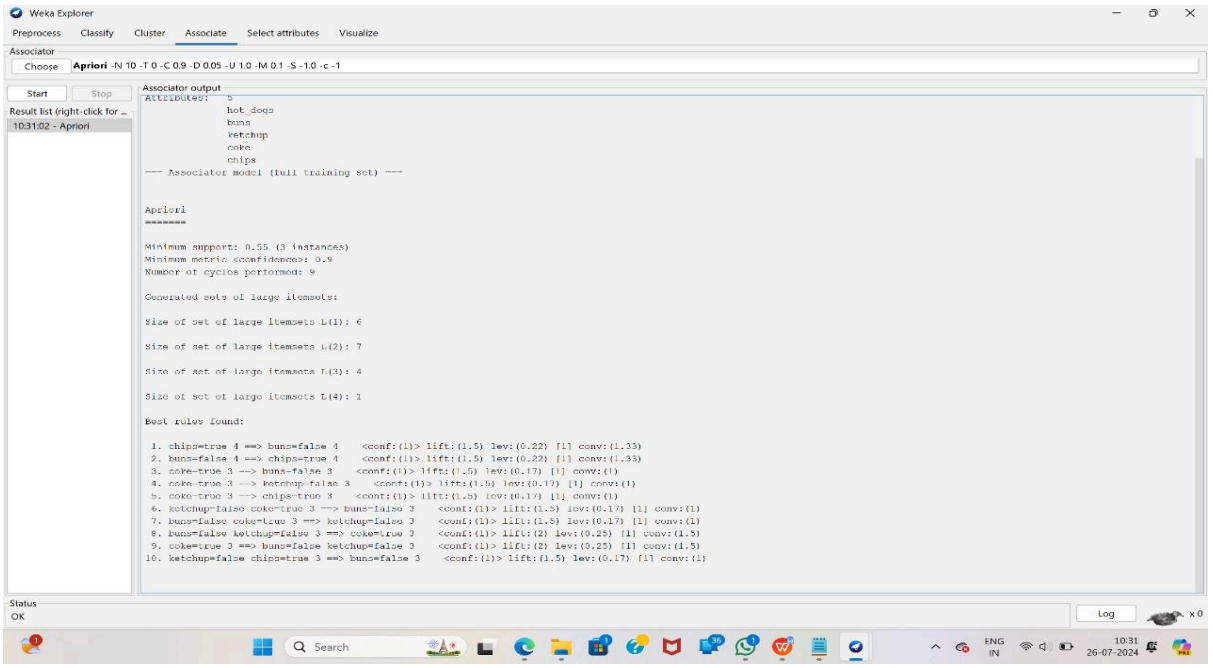
Logistic:



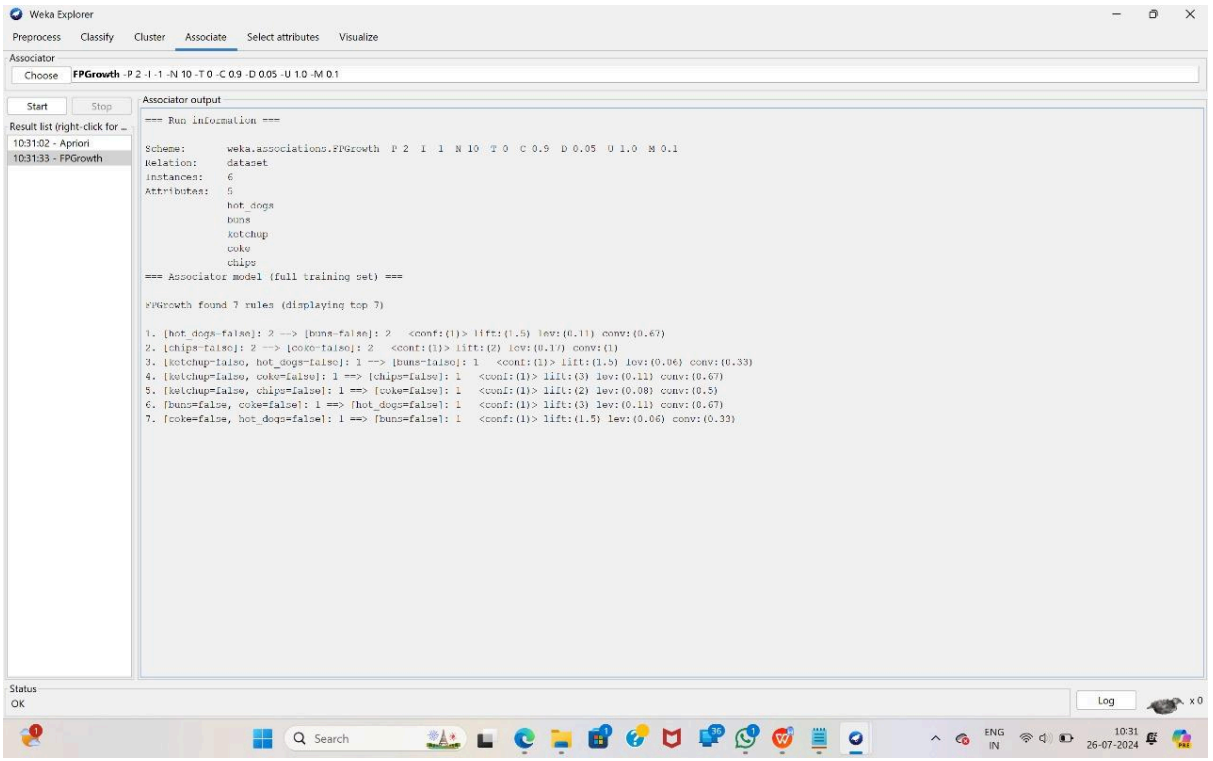
Q7:



Apriori Algorithm:

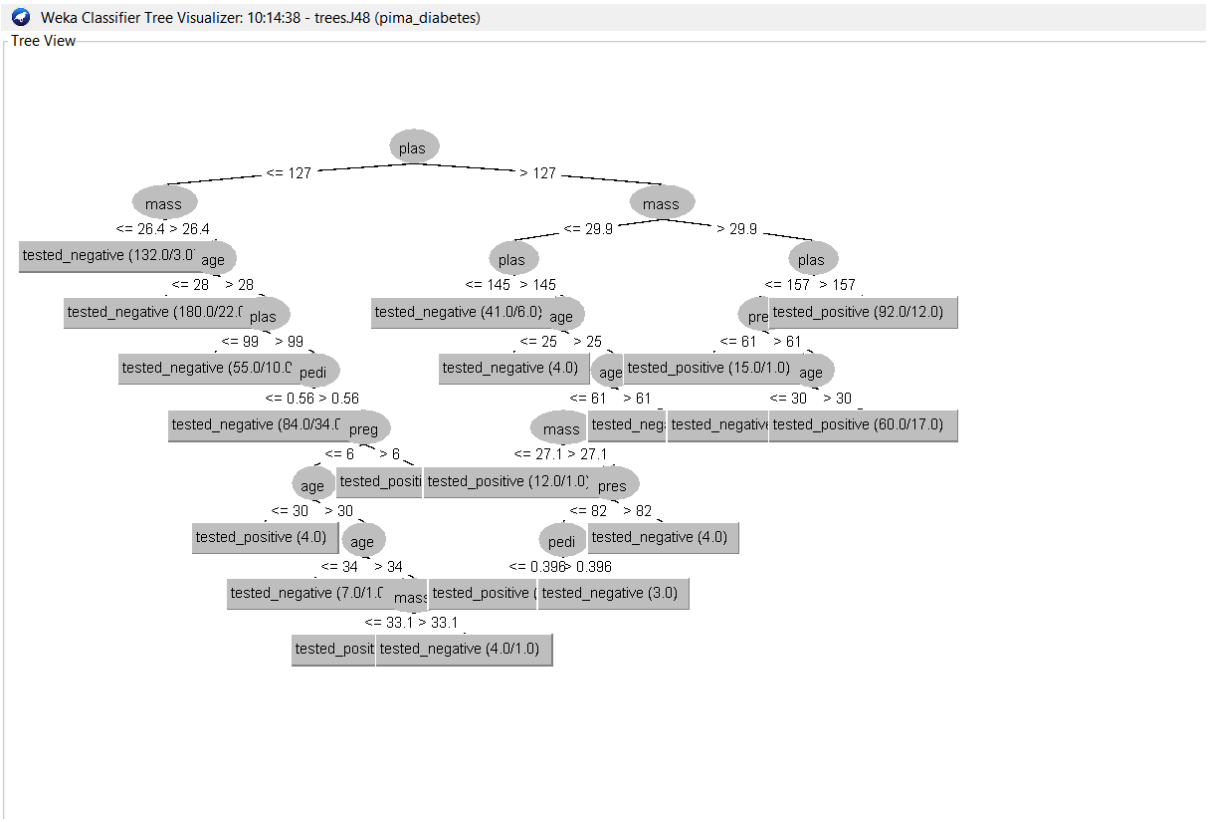


FP GROWTH:

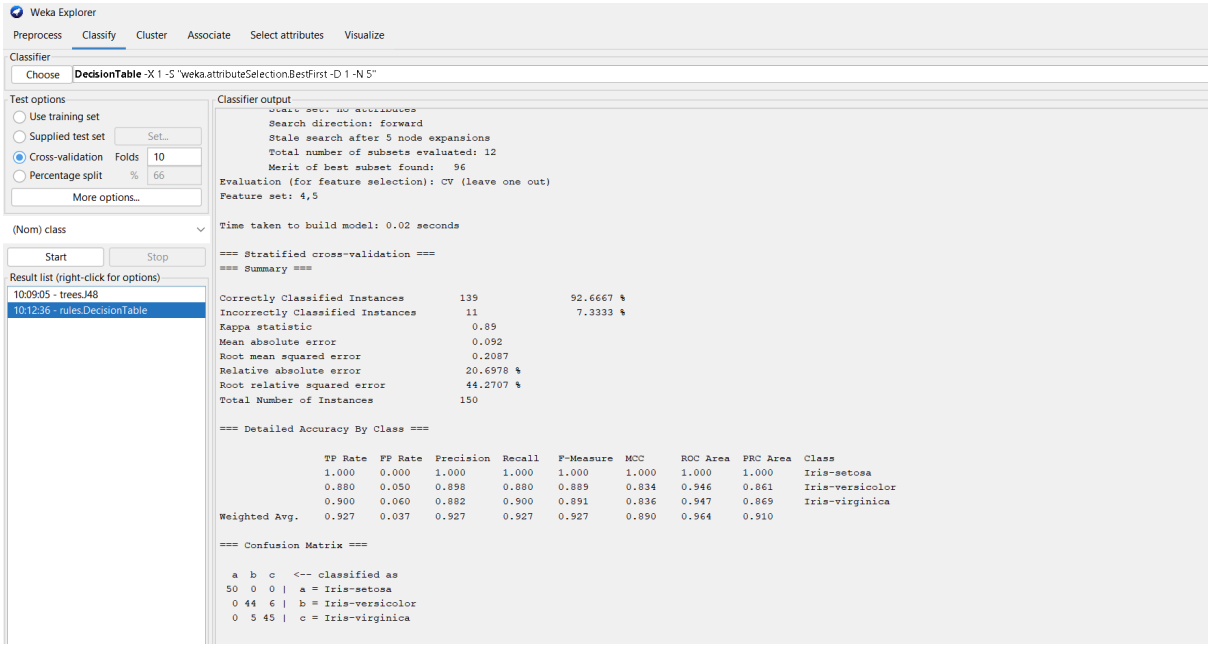


Q8:

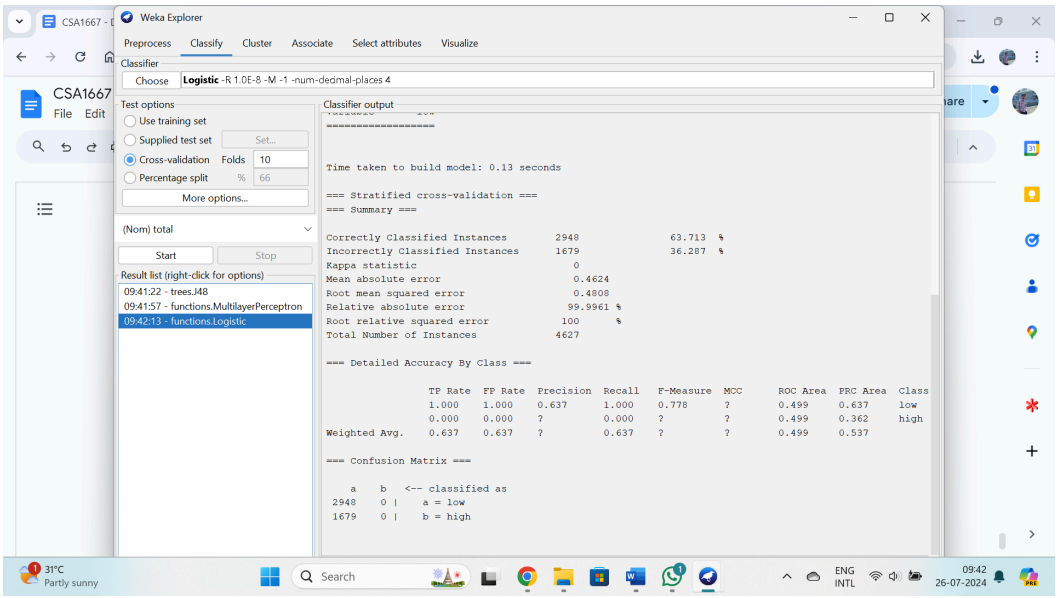
Tree:



Decision Tree:

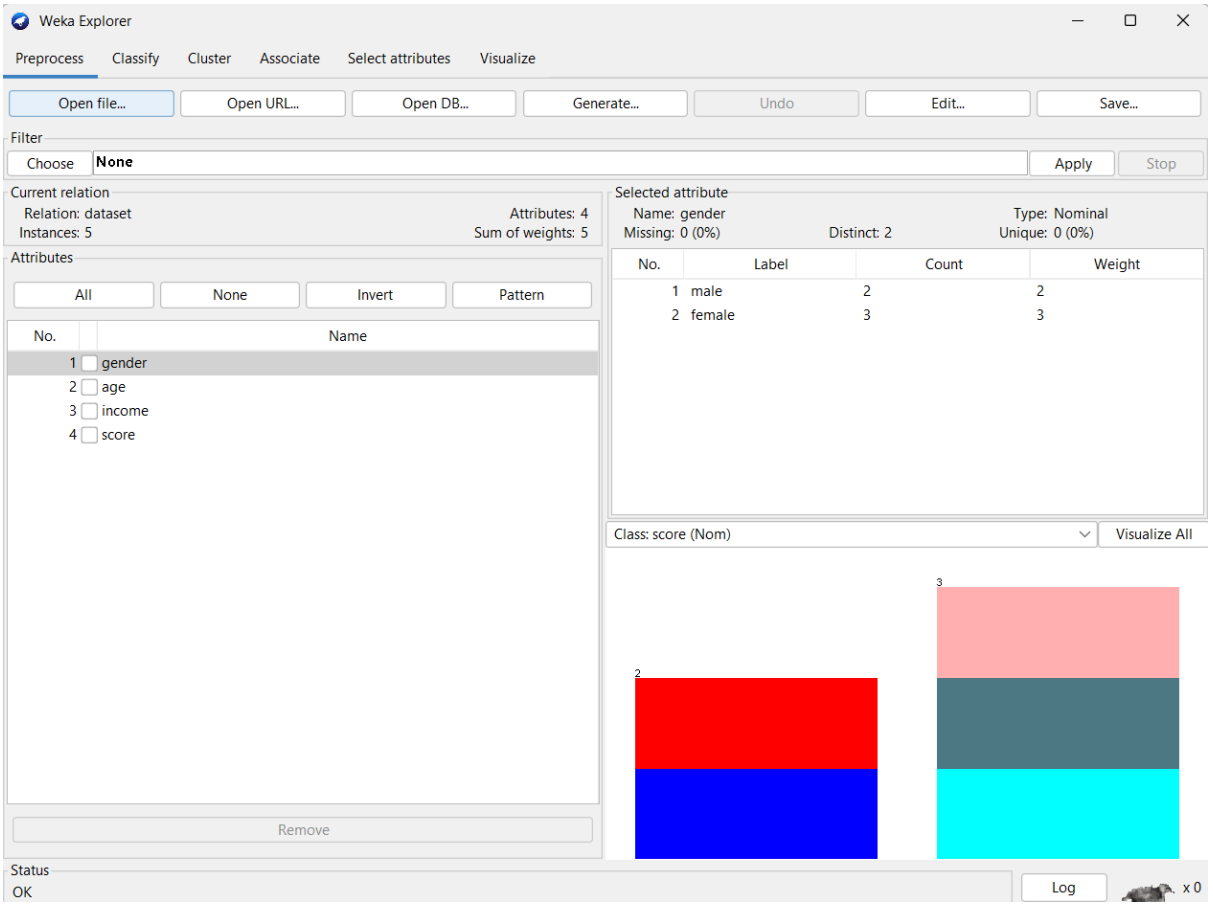


Logistic:

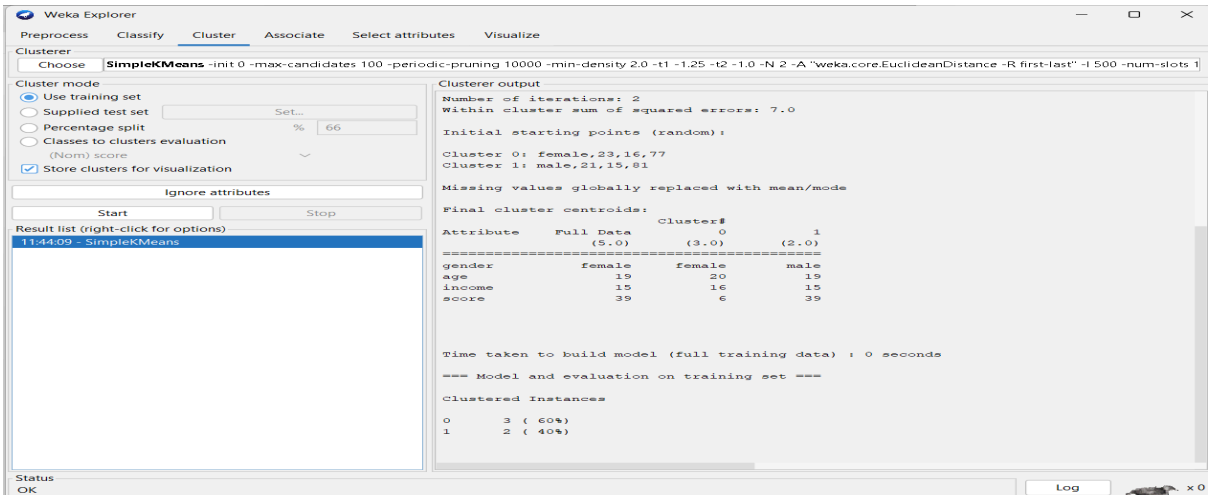


DAY-4

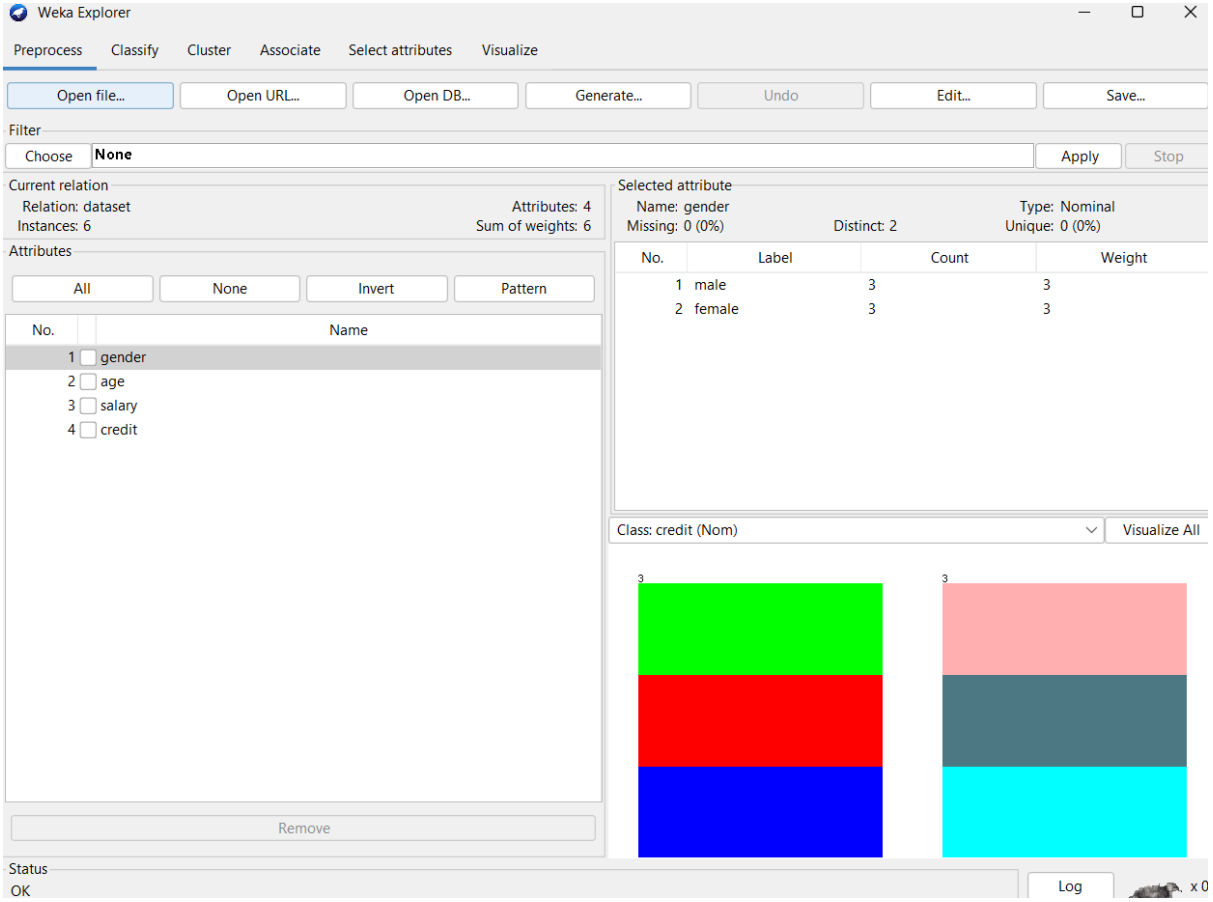
Q1:



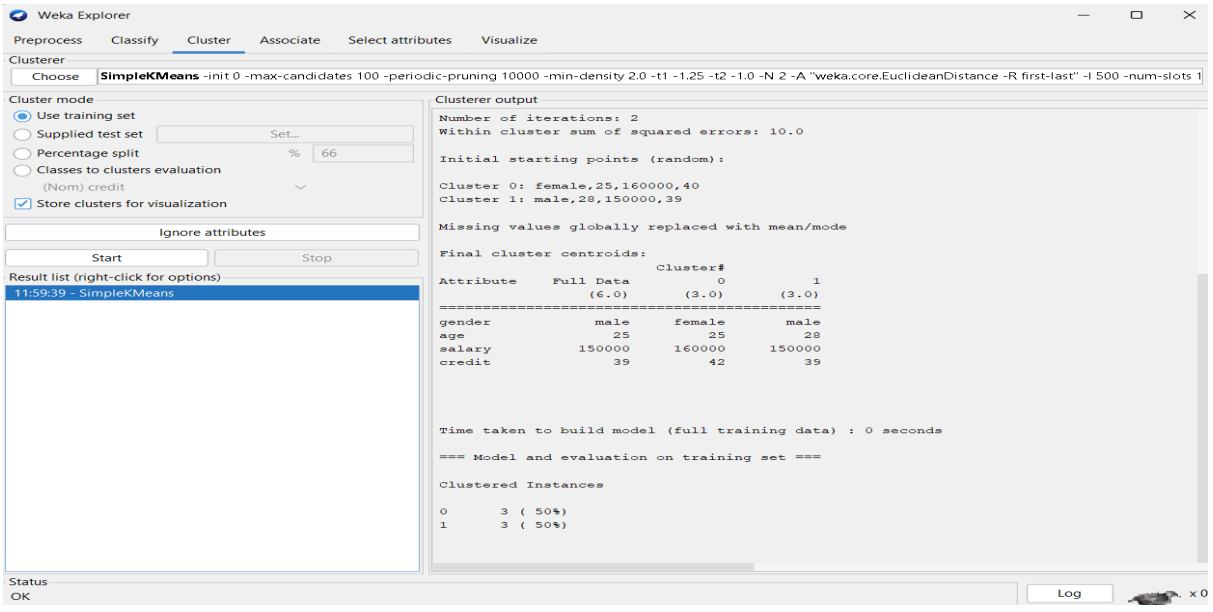
Clustering Analysis:



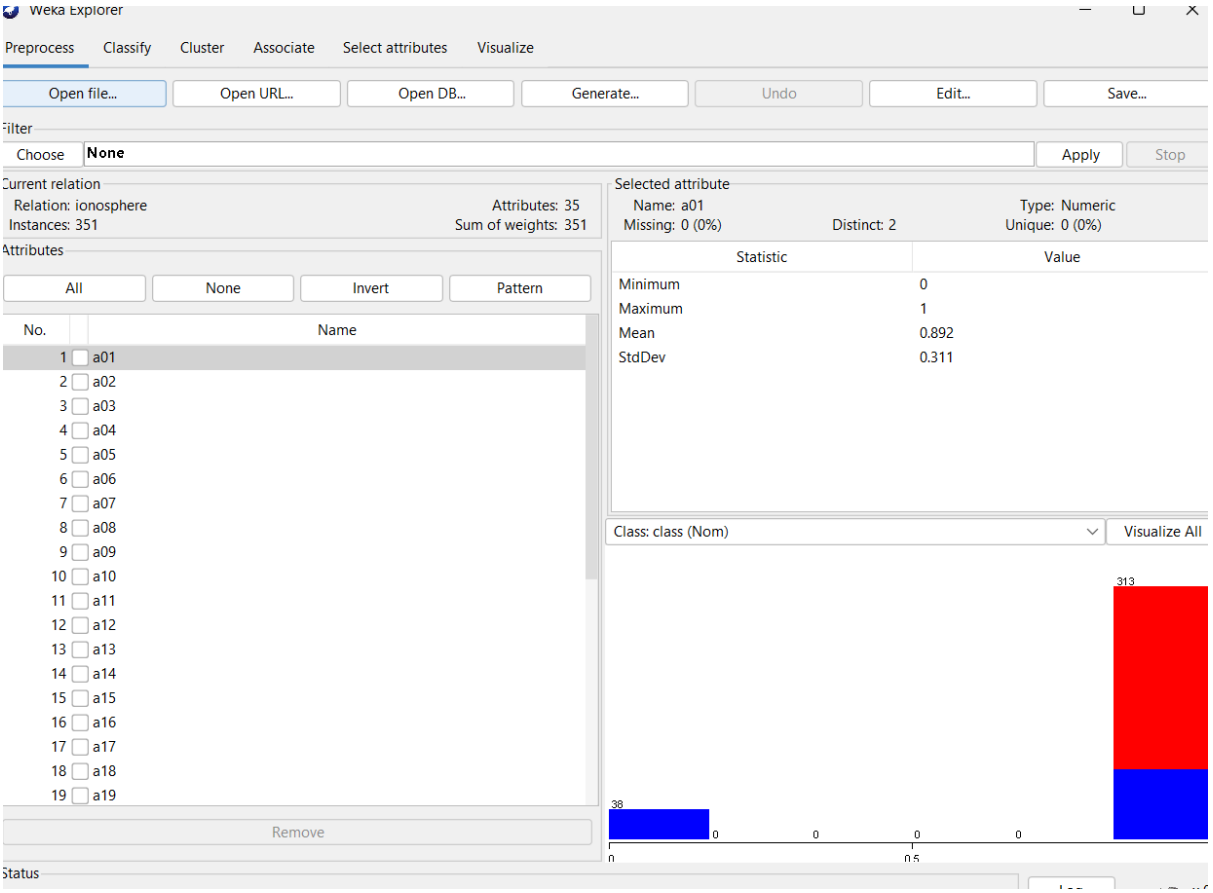
Q2:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: dataset
Instances: 6
Attributes: 4
Sum of weights: 6
Attributes
All
None
Invert
Pattern
No.
Name
1 gender
2 age
3 salary
4 credit
Remove
Selected attribute
Name: gender
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 male 3 3
2 female 3 3
Class: credit (Nom)
Visualize All
3
3
Status
OK
Log
x 0

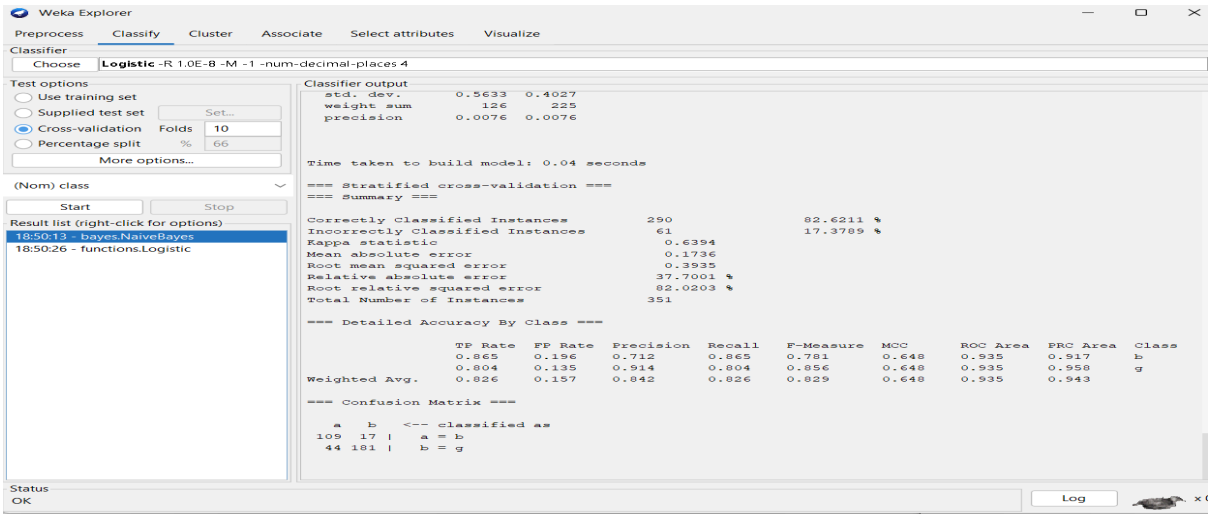
K- Means:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) credit
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
11:59:39 - SimpleKMeans
Clusterer output
Number of iterations: 2
Within cluster sum of squared errors: 10.0
Initial starting points (random):
Cluster 0: female,25,160000,40
Cluster 1: male,28,150000,39
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(6.0) (3.0) (3.0)
gender male female male
age 25 25 28
salary 150000 160000 150000
credit 39 42 39
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 3 (50%)
1 3 (50%)
Status
OK
Log
x 0

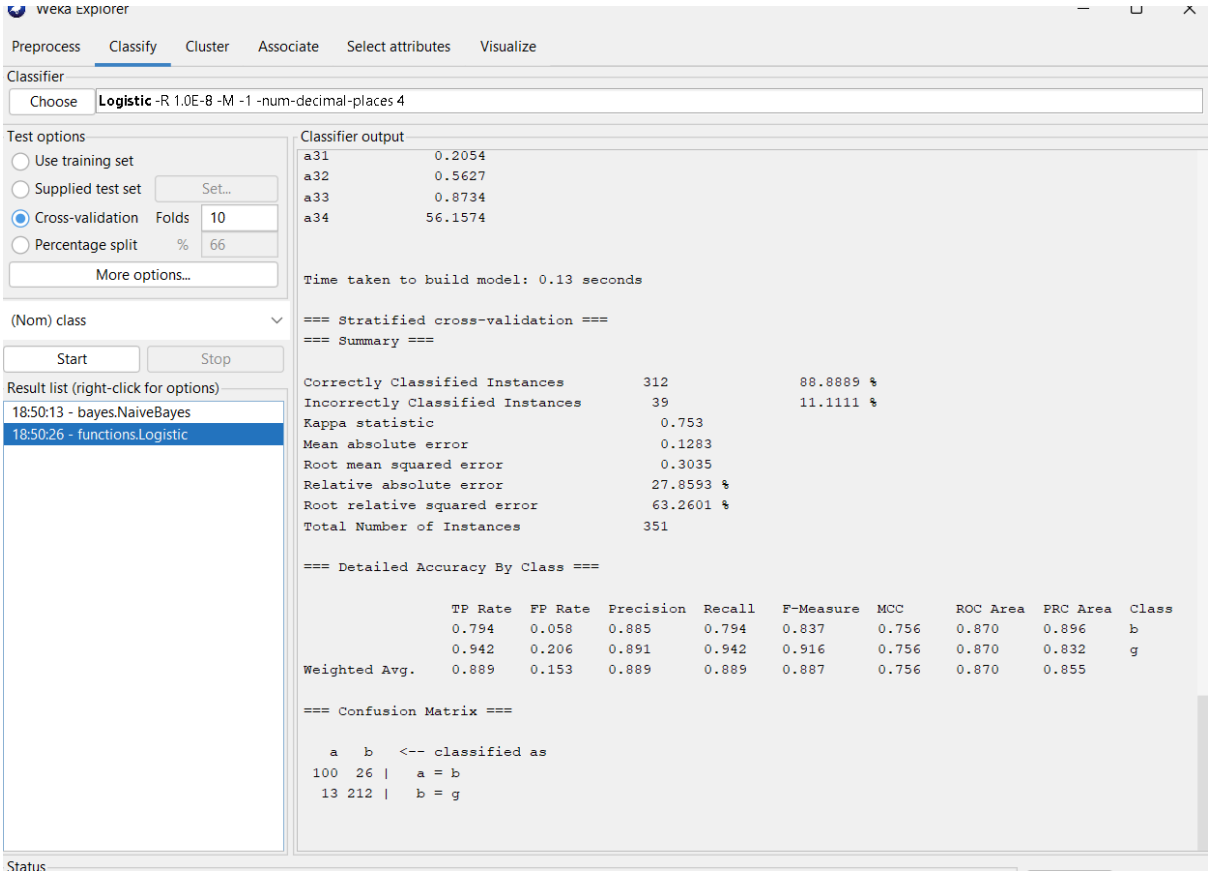
Q3:



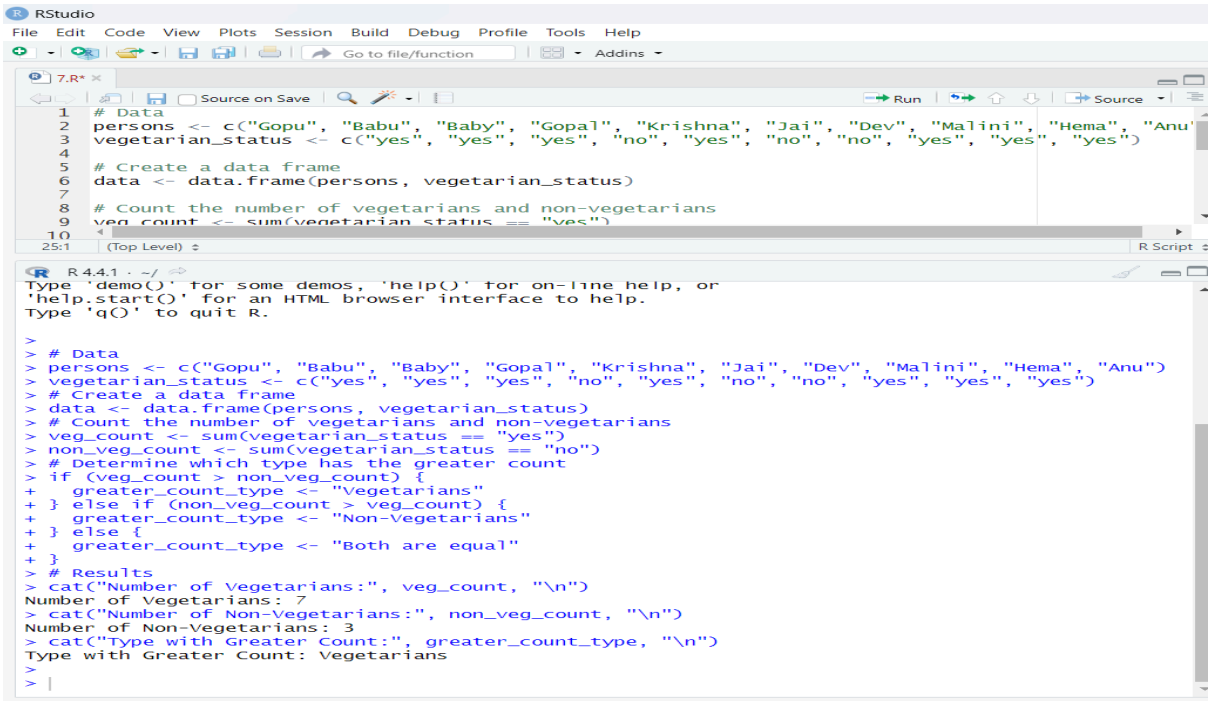
Naïve Bayes classification:



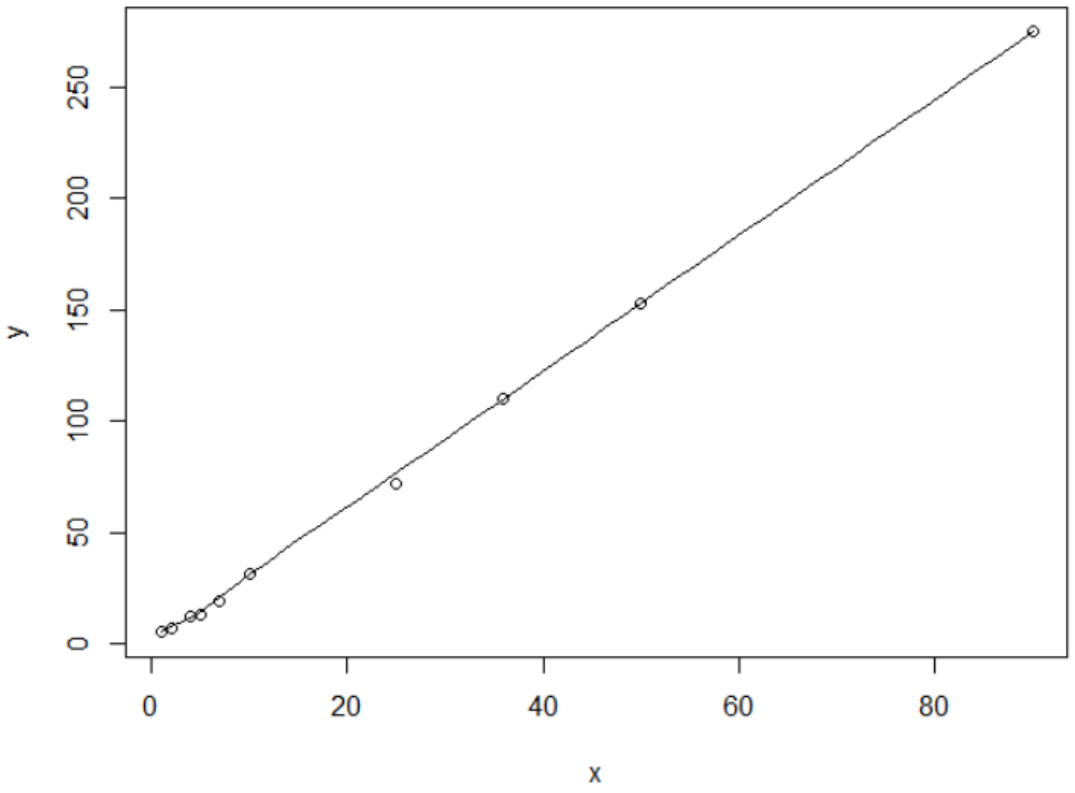
SVM:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
18:50:13 - bayes.NaiveBayes
18:50:26 - functions.Logistic
Classifier output
a31 0.2054
a32 0.5627
a33 0.8734
a34 56.1574
Time taken to build model: 0.13 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 312 88.8889 %
Incorrectly Classified Instances 39 11.1111 %
Kappa statistic 0.753
Mean absolute error 0.1283
Root mean squared error 0.3035
Relative absolute error 27.8593 %
Root relative squared error 63.2601 %
Total Number of Instances 351
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.794 0.058 0.885 0.794 0.837 0.756 0.870 0.896 b
0.942 0.206 0.891 0.942 0.916 0.756 0.870 0.832 g
Weighted Avg. 0.889 0.153 0.889 0.889 0.887 0.756 0.870 0.855
=== Confusion Matrix ===
a b <-- classified as
100 26 | a = b
13 212 | b = g
Status

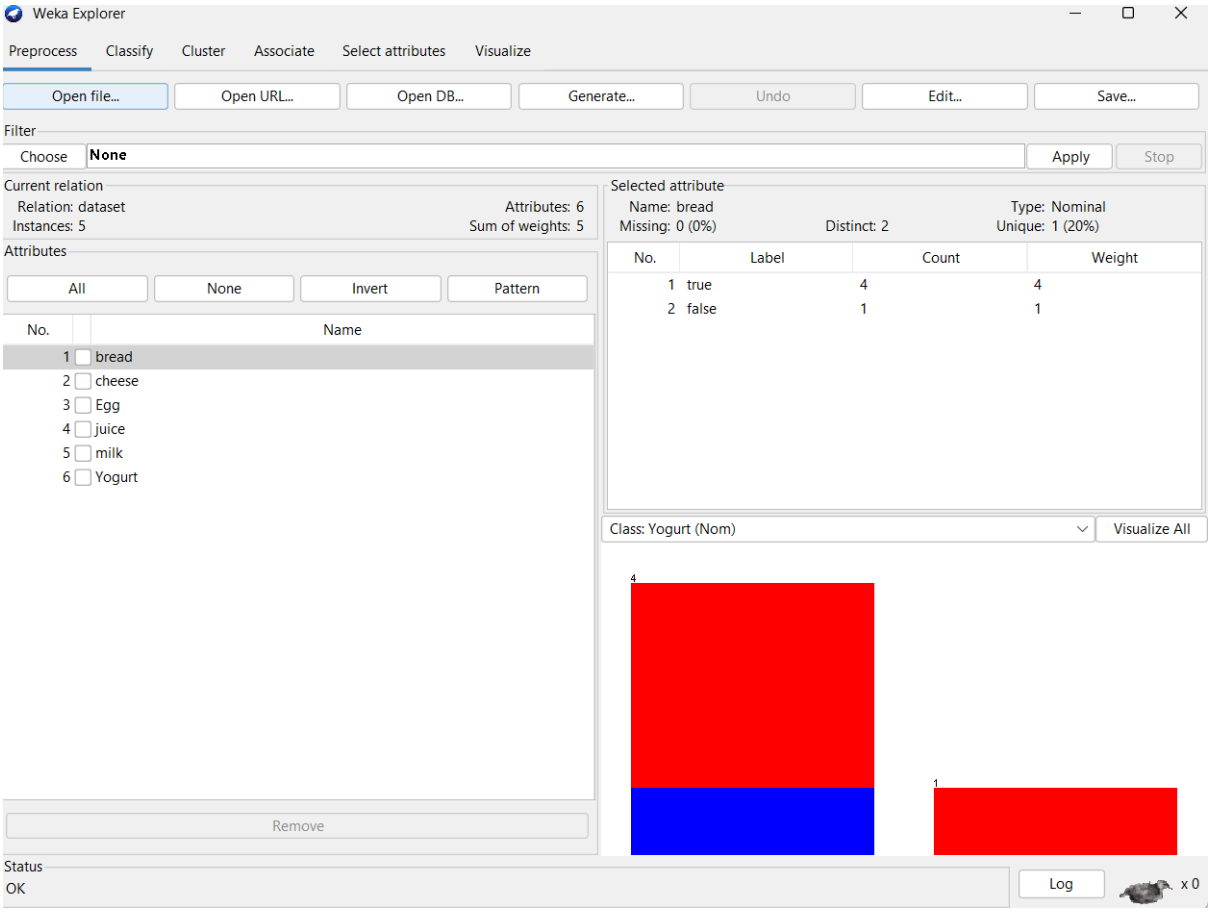
Q4:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
7.R*
Source on Save
Run
Source
# Data
persons <- c("Gopu", "Babu", "Baby", "Gopal", "Krishna", "Jai", "Dev", "Malini", "Hema", "Anu"
vegetarian_status <- c("yes", "yes", "yes", "no", "yes", "no", "no", "yes", "yes", "yes")
# Create a data frame
data <- data.frame(persons, vegetarian_status)
# Count the number of vegetarians and non-vegetarians
veg_count <- sum(vegetarian_status == "yes")
25:1 (Top Level)
R Script
R 4.4.1 · ~/
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.
> # Data
> persons <- c("Gopu", "Babu", "Baby", "Gopal", "Krishna", "Jai", "Dev", "Malini", "Hema", "Anu")
> vegetarian_status <- c("yes", "yes", "yes", "no", "yes", "no", "no", "yes", "yes", "yes")
> # Create a data frame
> data <- data.frame(persons, vegetarian_status)
> # Count the number of vegetarians and non-vegetarians
> veg_count <- sum(vegetarian_status == "yes")
> non_veg_count <- sum(vegetarian_status == "no")
> # Determine which type has the greater count
> if (veg_count > non_veg_count) {
+ greater_count_type <- "Vegetarians"
+ } else if (non_veg_count > veg_count) {
+ greater_count_type <- "Non-Vegetarians"
+ } else {
+ greater_count_type <- "Both are equal"
+ }
> # Results
> cat("Number of Vegetarians:", veg_count, "\n")
Number of Vegetarians: 7
> cat("Number of Non-Vegetarians:", non_veg_count, "\n")
Number of Non-Vegetarians: 3
> cat("Type with Greater Count:", greater_count_type, "\n")
Type with Greater Count: Vegetarians

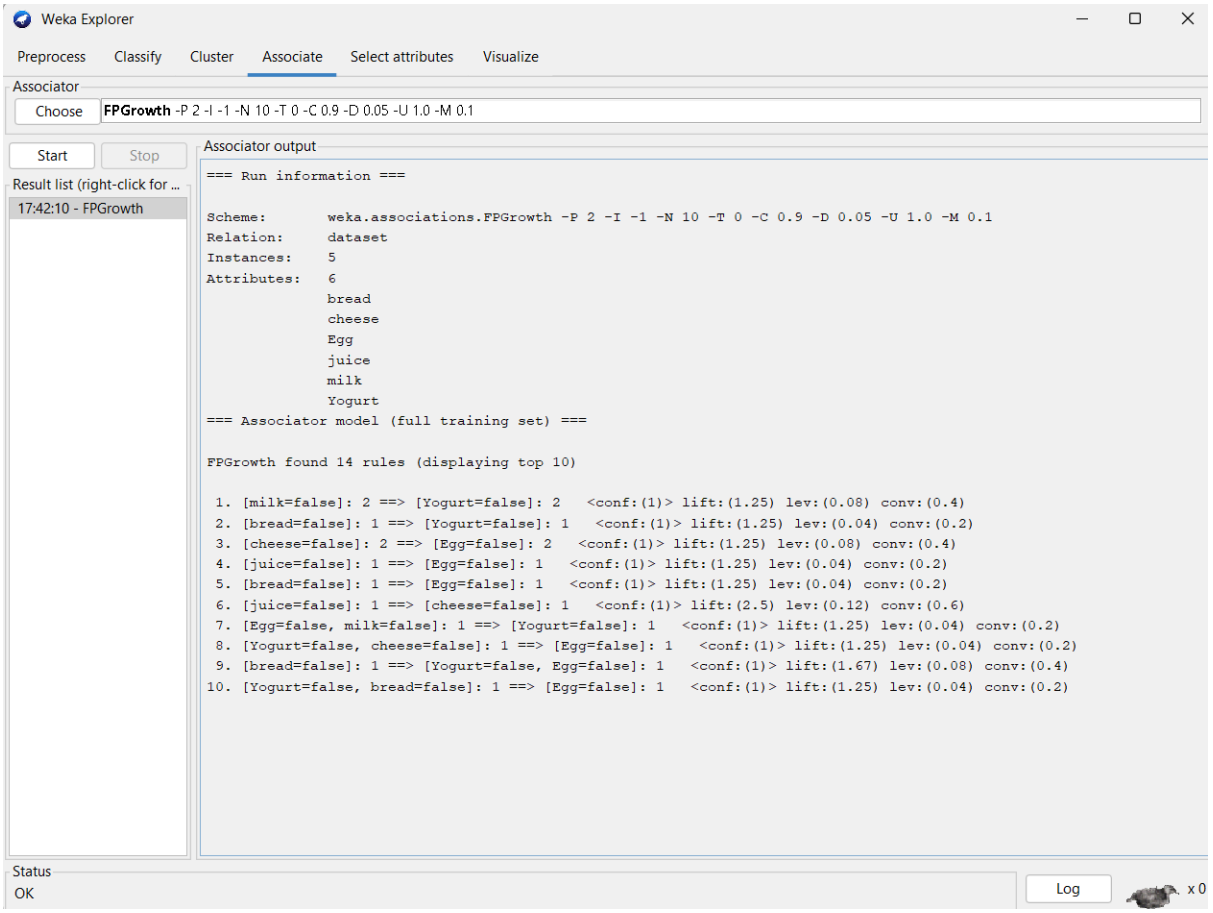
Q5:


y
0
50
100
150
200
250
0
20
40
60
80
x

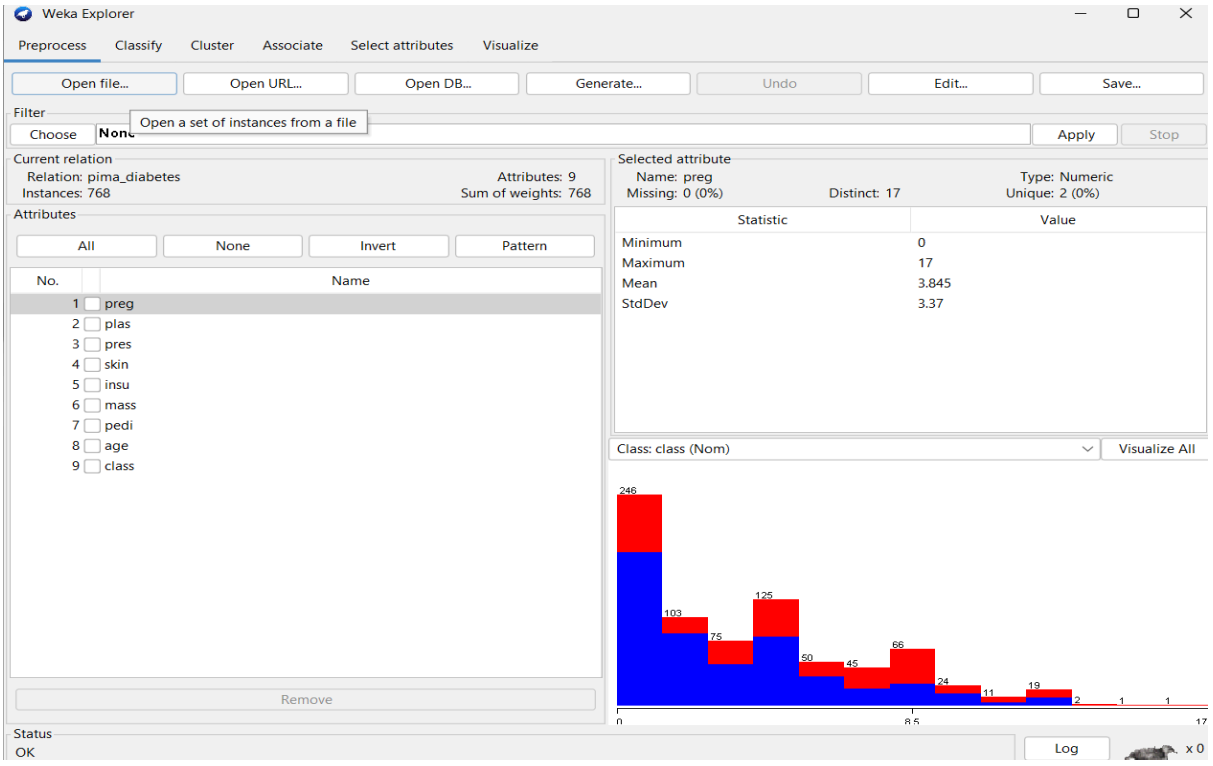
Q6:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: dataset
Instances: 5
Attributes: 6
Sum of weights: 5
Attributes
All
None
Invert
Pattern
No.
Name
1 bread
2 cheese
3 Egg
4 juice
5 milk
6 Yogurt
Remove
Selected attribute
Name: bread
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 1 (20%)
No.
Label
Count
Weight
1 true 4 4
2 false 1 1
Class: Yogurt (Nom)
Visualize All
4
1
Status
OK
Log
x 0

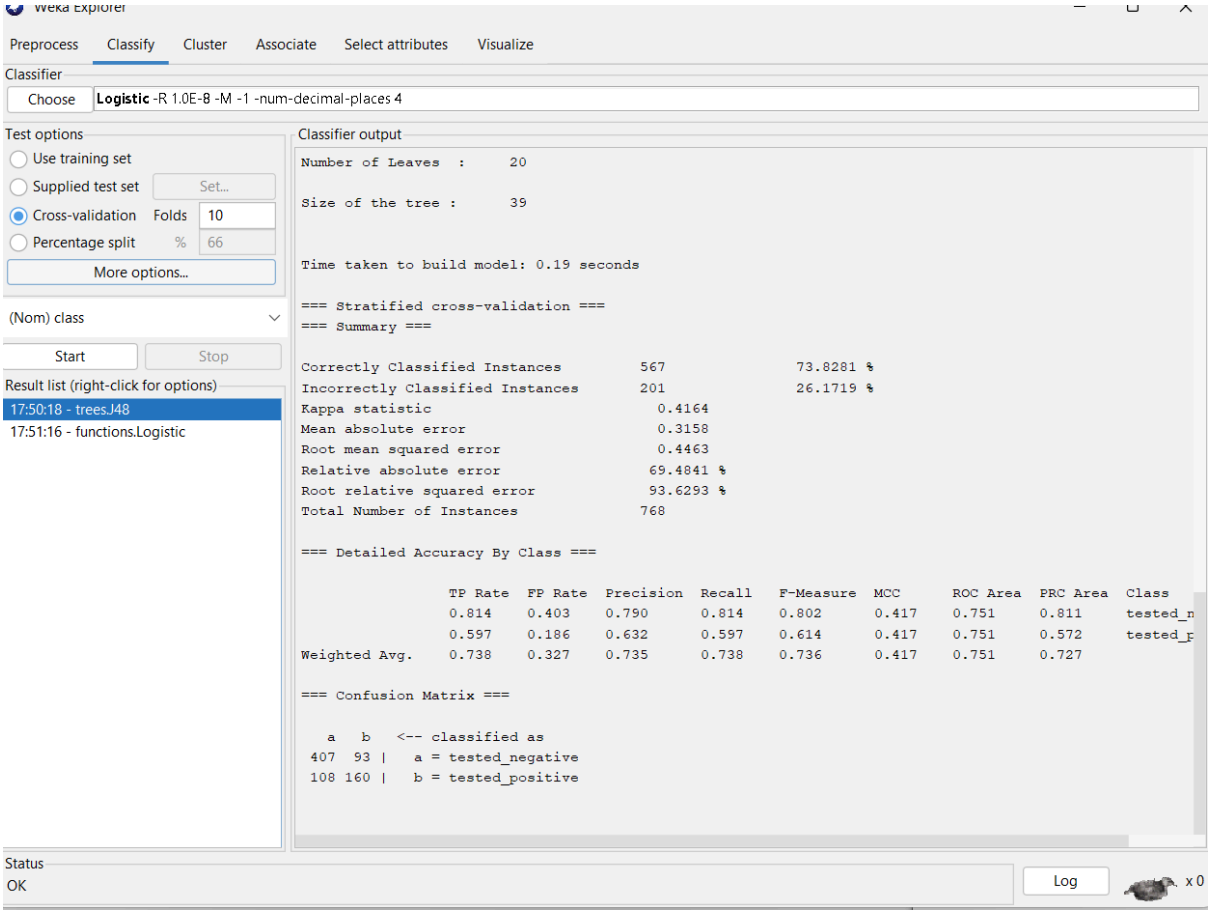
FP GROWTH:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Start
Stop
Result list (right-click for ...
17:42:10 - FPGrowth
Associator output
=== Run information ===
Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation: dataset
Instances: 5
Attributes: 6
bread
cheese
Egg
juice
milk
Yogurt
=== Associator model (full training set) ===
FPGrowth found 14 rules (displaying top 10)
1. [milk=false]: 2 ==> [Yogurt=false]: 2 <conf:(1)> lift:(1.25) lev:(0.08) conv:(0.4)
2. [bread=false]: 1 ==> [Yogurt=false]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv:(0.2)
3. [cheese=false]: 2 ==> [Egg=false]: 2 <conf:(1)> lift:(1.25) lev:(0.08) conv:(0.4)
4. [juice=false]: 1 ==> [Egg=false]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv:(0.2)
5. [bread=false]: 1 ==> [Egg=false]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv:(0.2)
6. [juice=false]: 1 ==> [cheese=false]: 1 <conf:(1)> lift:(2.5) lev:(0.12) conv:(0.6)
7. [Egg=false, milk=false]: 1 ==> [Yogurt=false]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv:(0.2)
8. [Yogurt=false, cheese=false]: 1 ==> [Egg=false]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv:(0.2)
9. [bread=false]: 1 ==> [Yogurt=false, Egg=false]: 1 <conf:(1)> lift:(1.67) lev:(0.08) conv:(0.4)
10. [Yogurt=false, bread=false]: 1 ==> [Egg=false]: 1 <conf:(1)> lift:(1.25) lev:(0.04) conv:(0.2)
Status
OK
Log
x 0

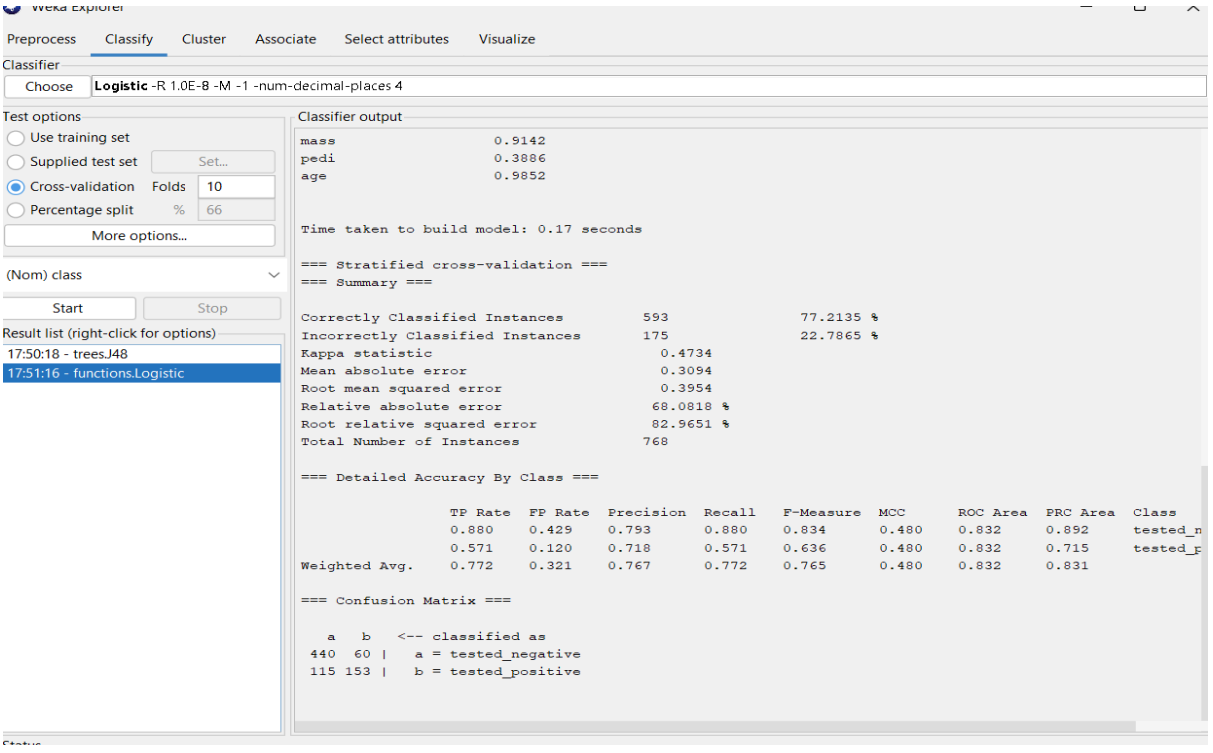
Q7:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Open a set of instances from a file
Apply
Stop
Current relation
Relation: pima_diabetes
Instances: 768
Attributes: 9
Sum of weights: 768
Attributes
All
None
Invert
Pattern
No.
Name
1 preg
2 plas
3 pres
4 skin
5 insu
6 mass
7 pedi
8 age
9 class
Remove
Selected attribute
Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)
Statistic
Value
Minimum 0
Maximum 17
Mean 3.845
StdDev 3.37
Class: class (Nom)
Visualize All
246
103
75
125
50
45
66
24
11
19
2
1
1
0
8.5
17
Status
OK
Log
x 0

Decision Tree Classifier:



Support Vector Machine Classifier:



Q8:
INPUT:marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)

bins_a <- cut(marks, breaks = 3, labels = c("Low", "Medium", "High"))

bins_b <- cut(marks, breaks = seq(min(marks), max(marks), length.out = 4), labels = c("Low", "Medium", "High"))

k <- 3

clusters <- kmeans(matrix(marks), centers = k)

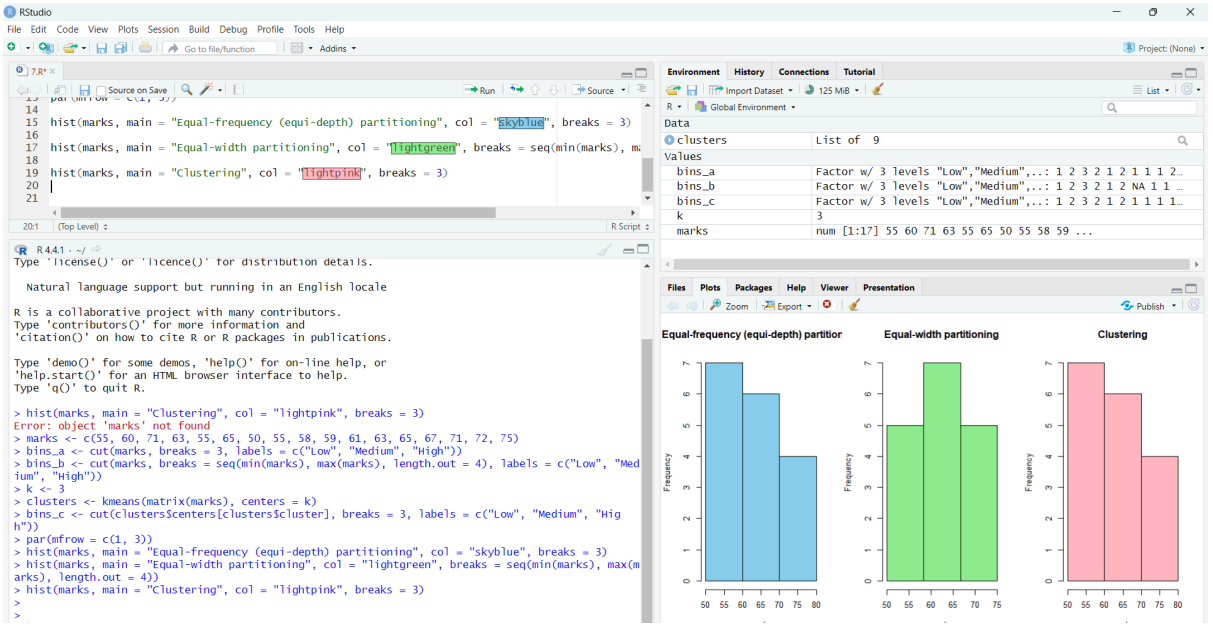
bins_c <- cut(clusters$centers[clusters$cluster], breaks = 3, labels = c("Low", "Medium", "High"))

par(mfrow = c(1, 3))

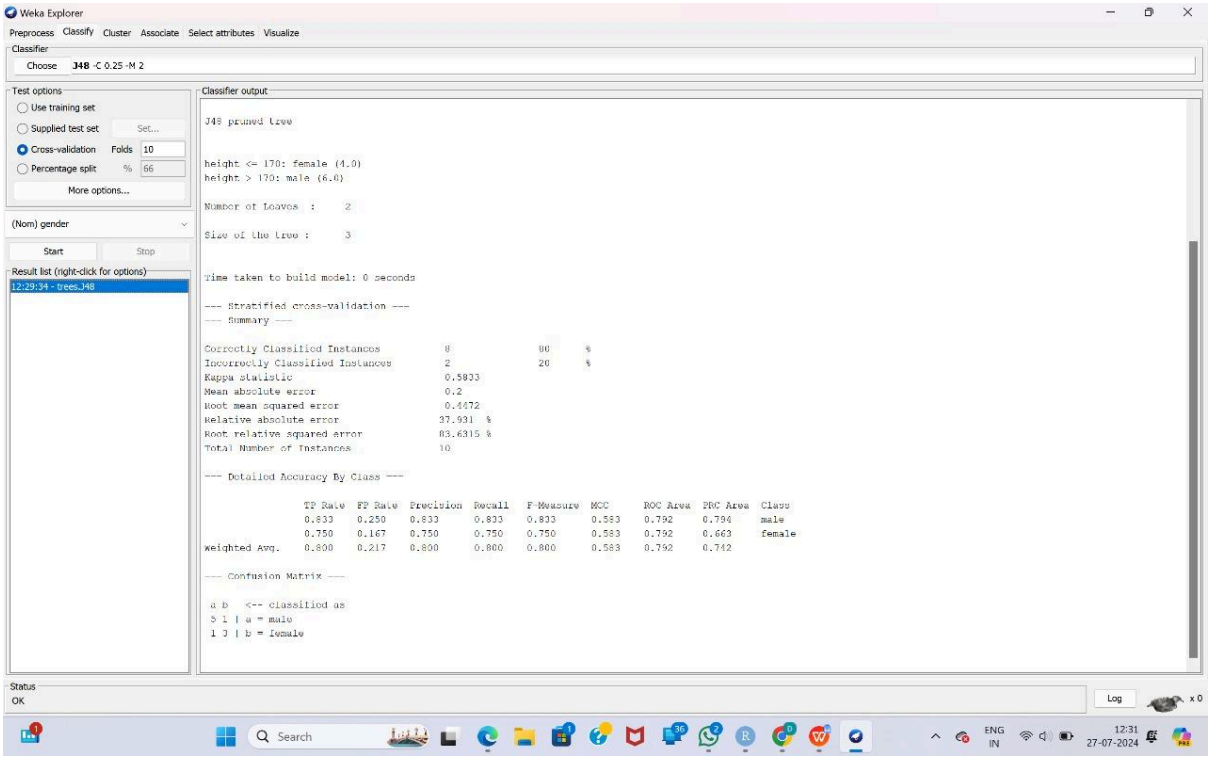
hist(marks, main = "Equal-frequency (equi-depth) partitioning", col = "skyblue", breaks = 3)

hist(marks, main = "Equal-width partitioning", col = "lightgreen", breaks = seq(min(marks), max(marks), length.out = 4))

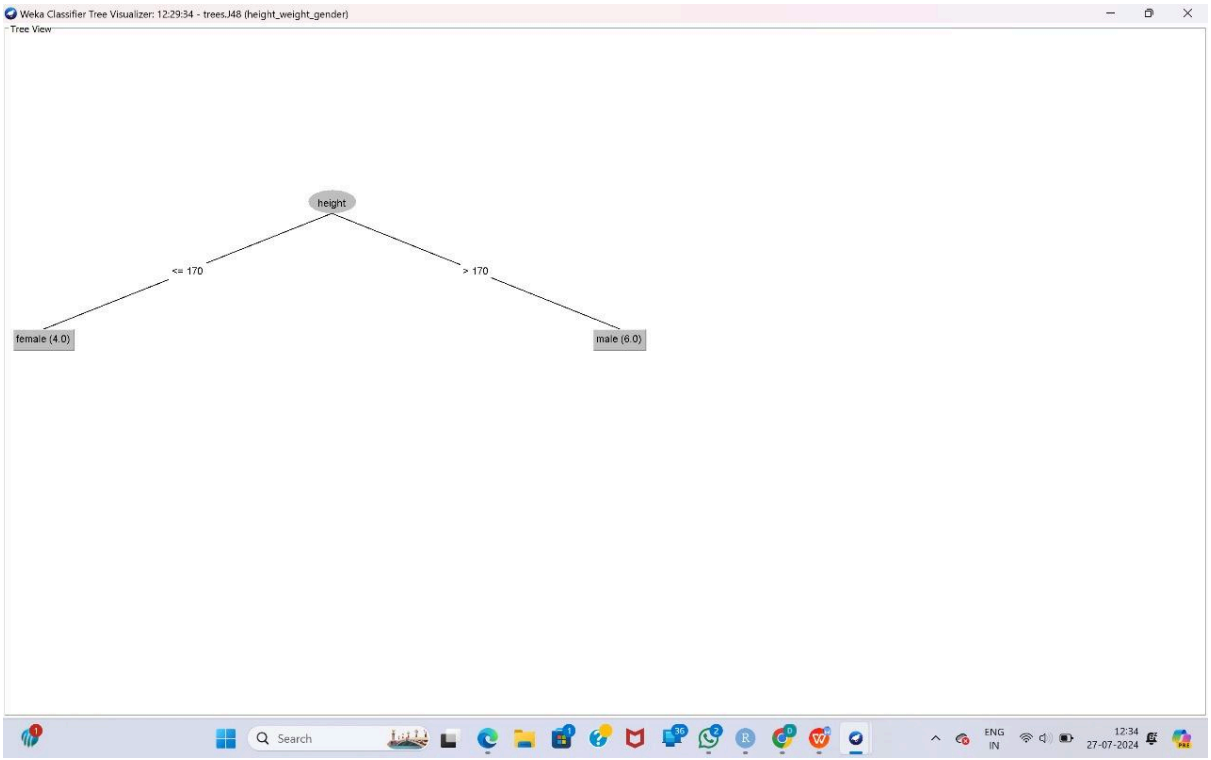
hist(marks, main = "Clustering", col = "lightpink", breaks = 3)



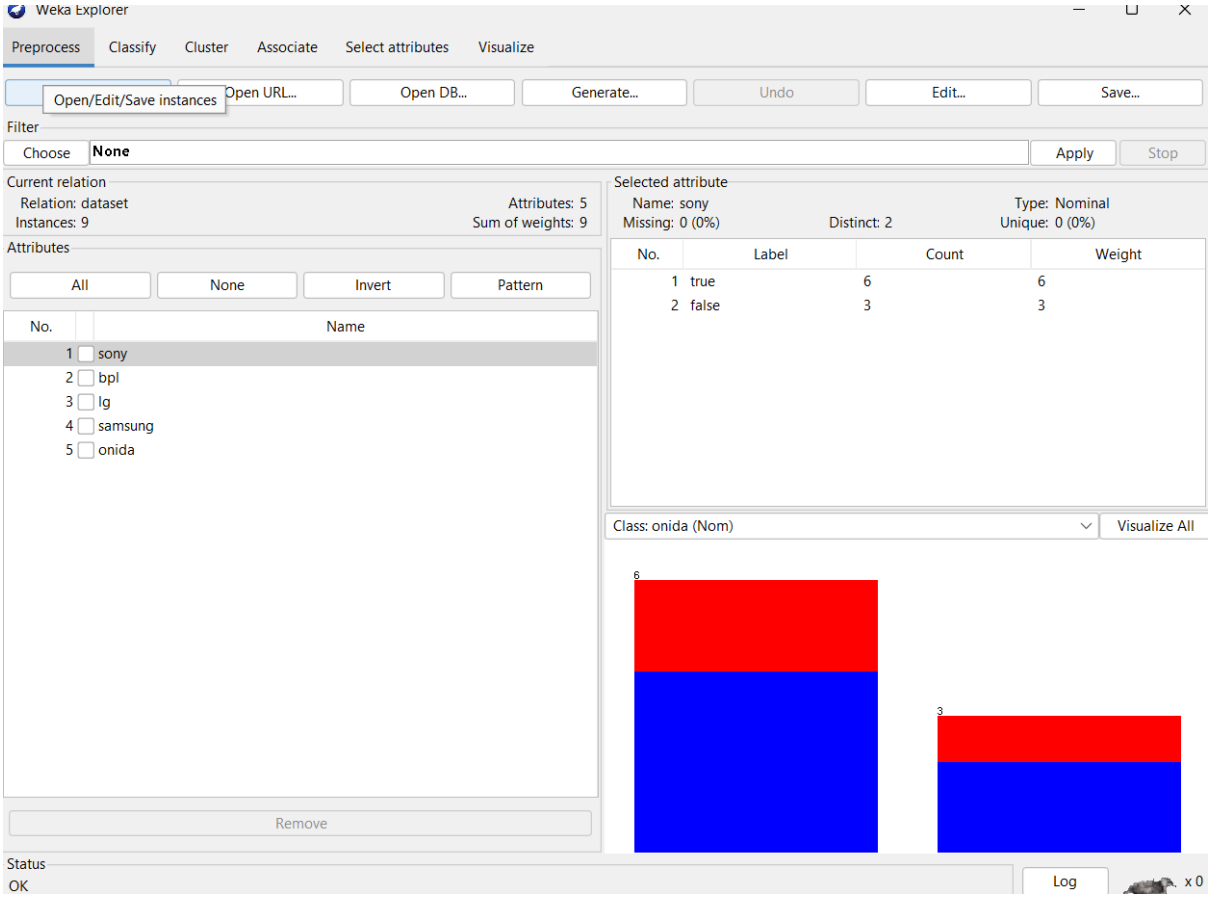
Q9:



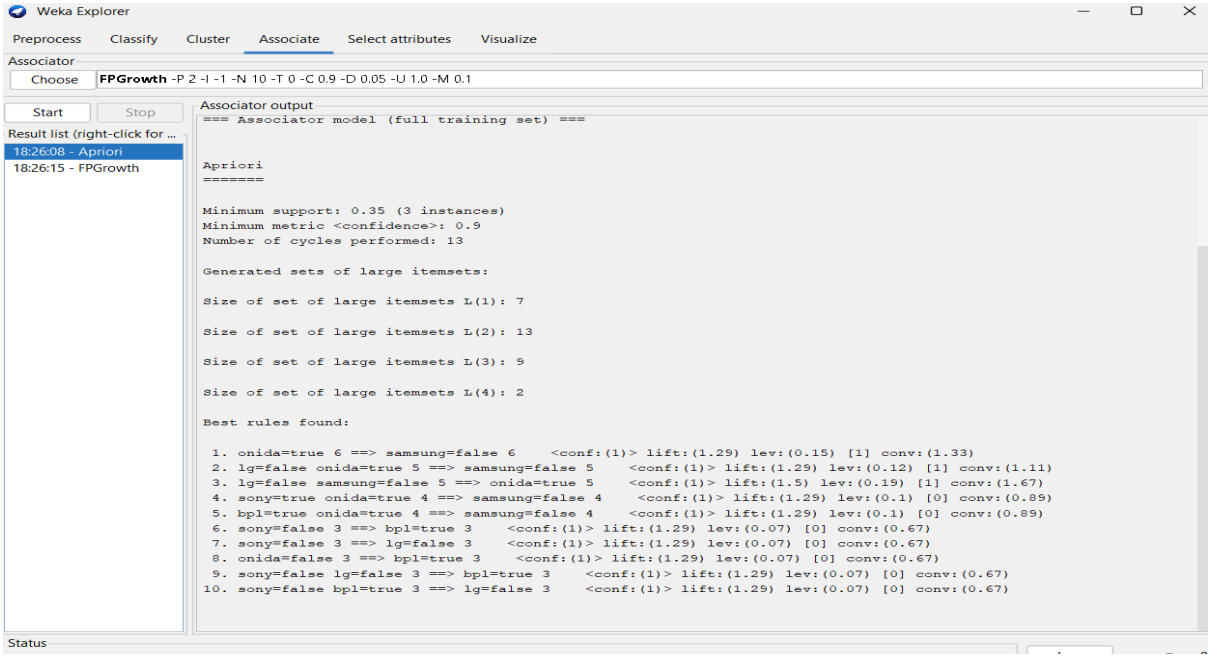
Decision Tree :



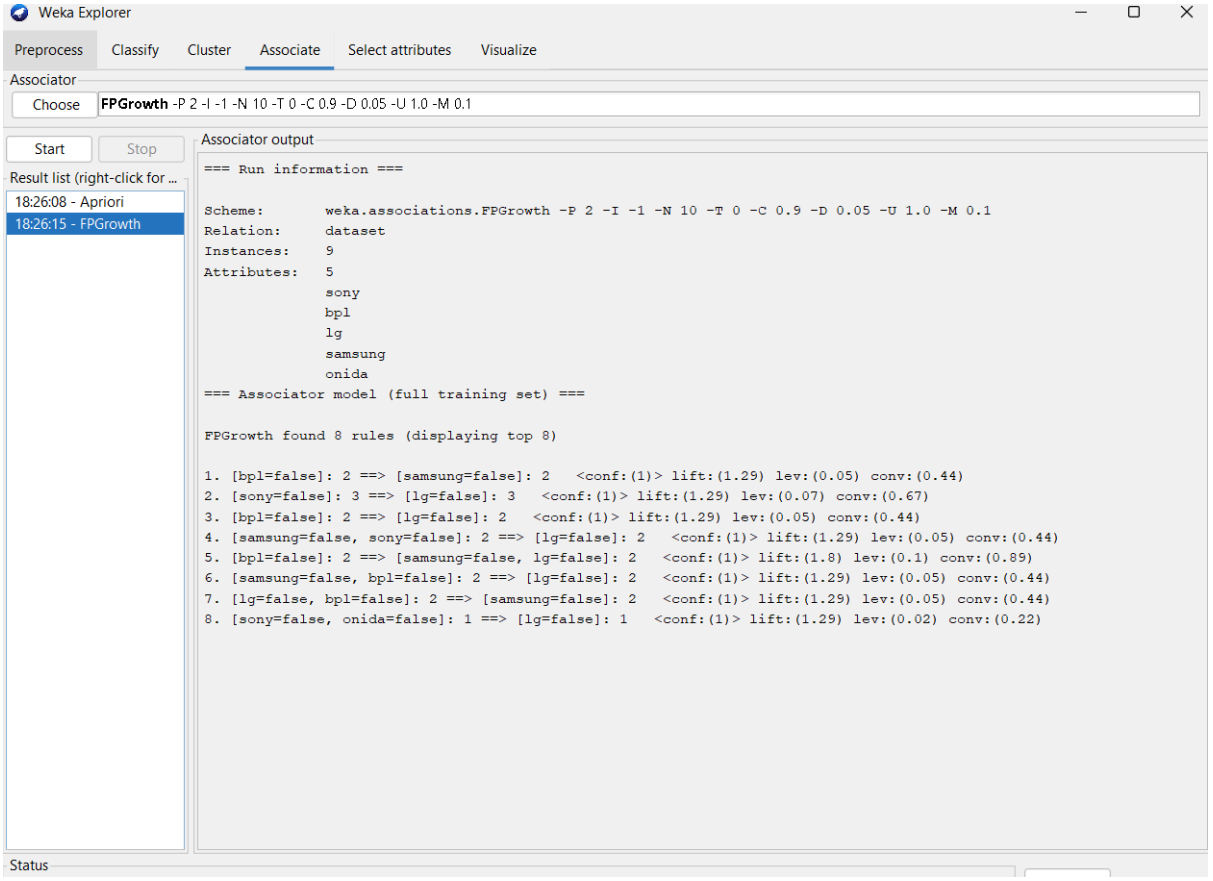
Q10:



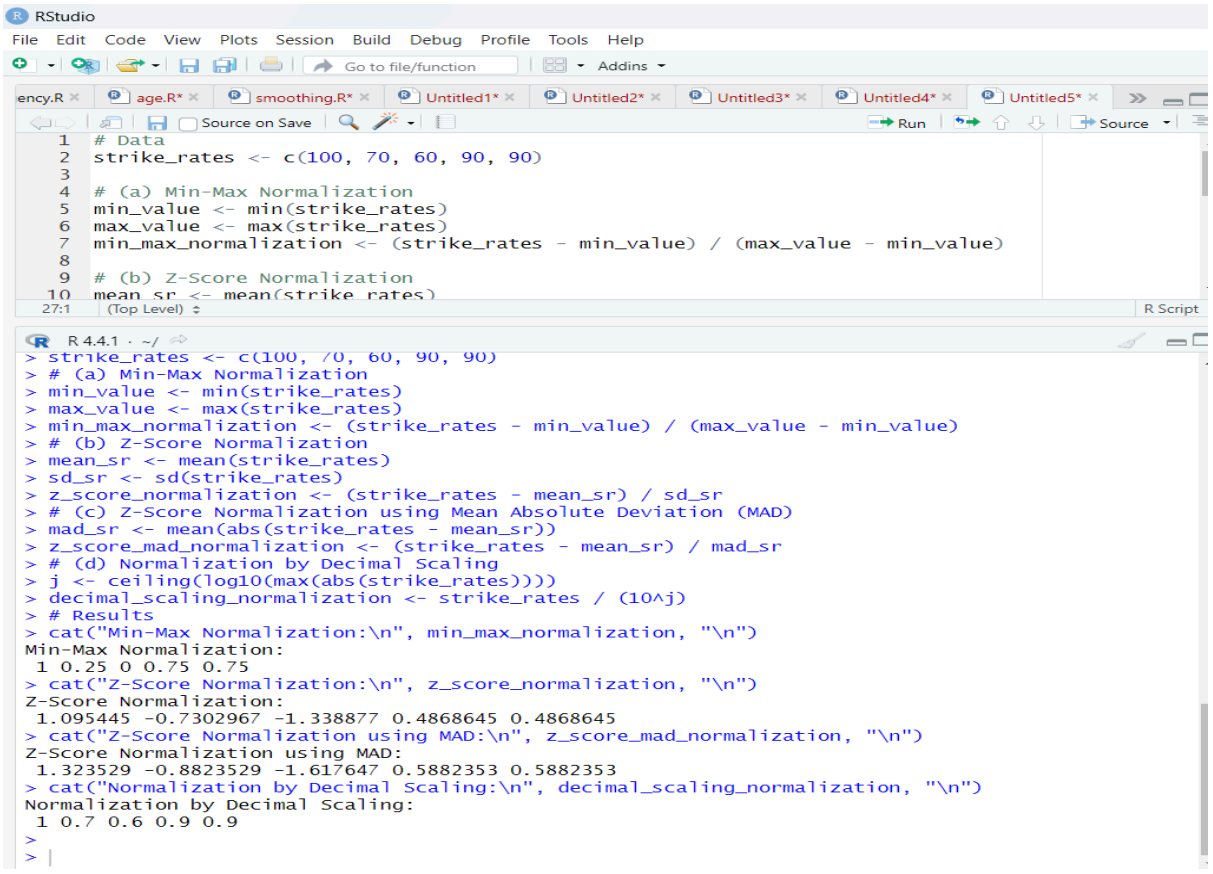
Apriori Algorithm:



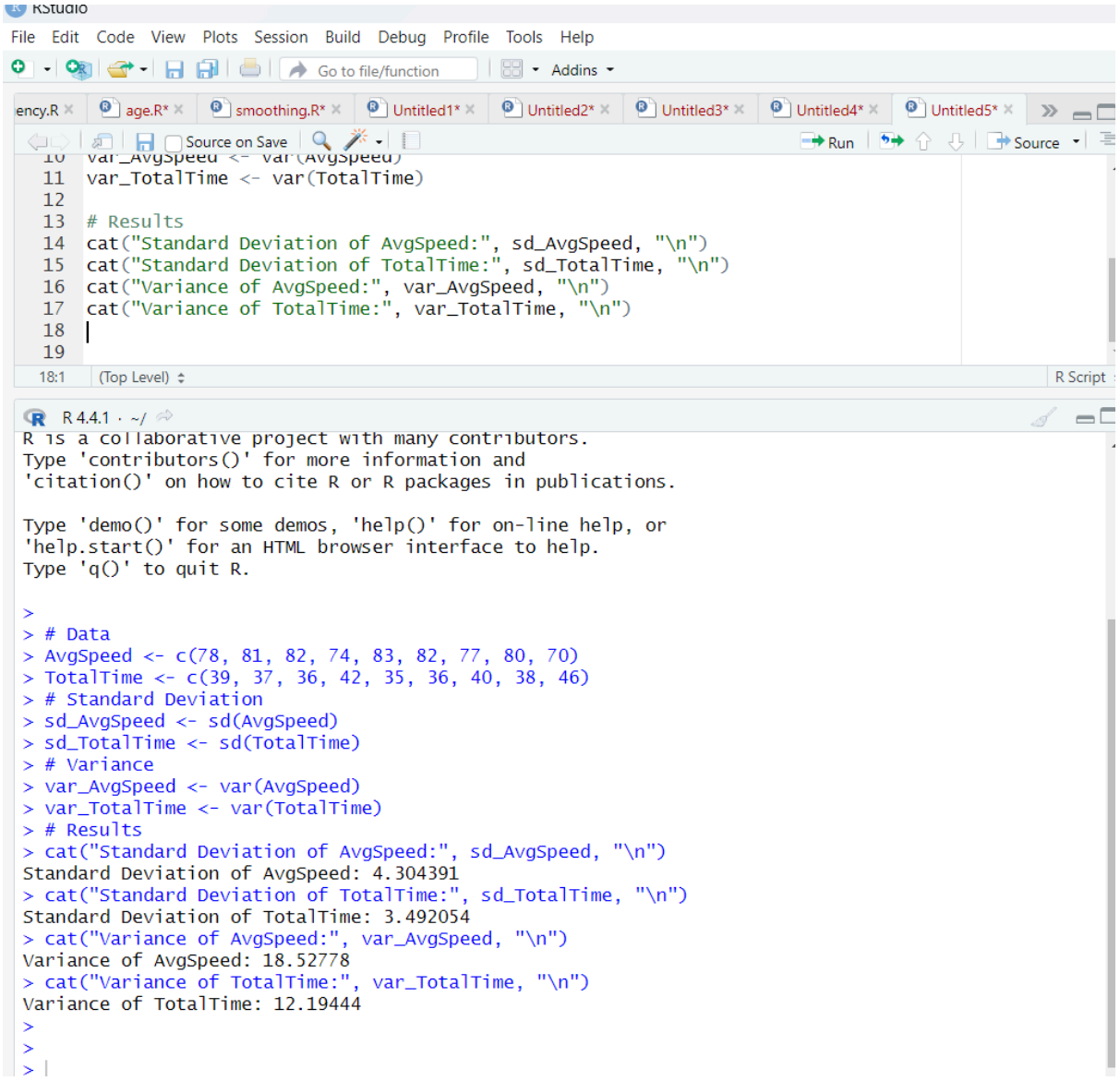
FP GROWTH:



Q11:



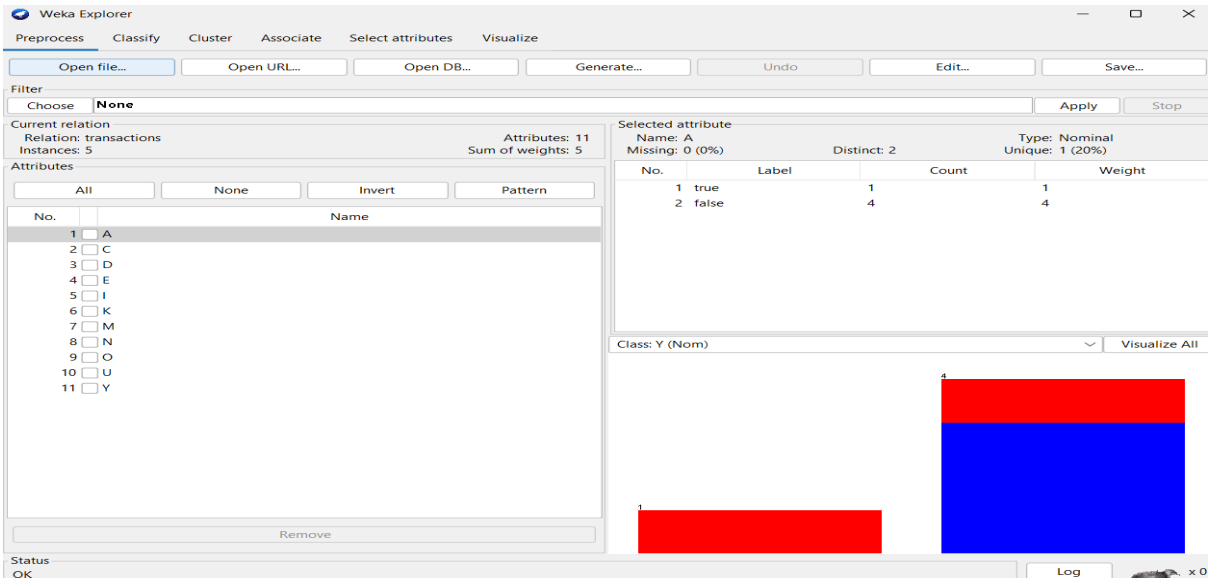
Q12:



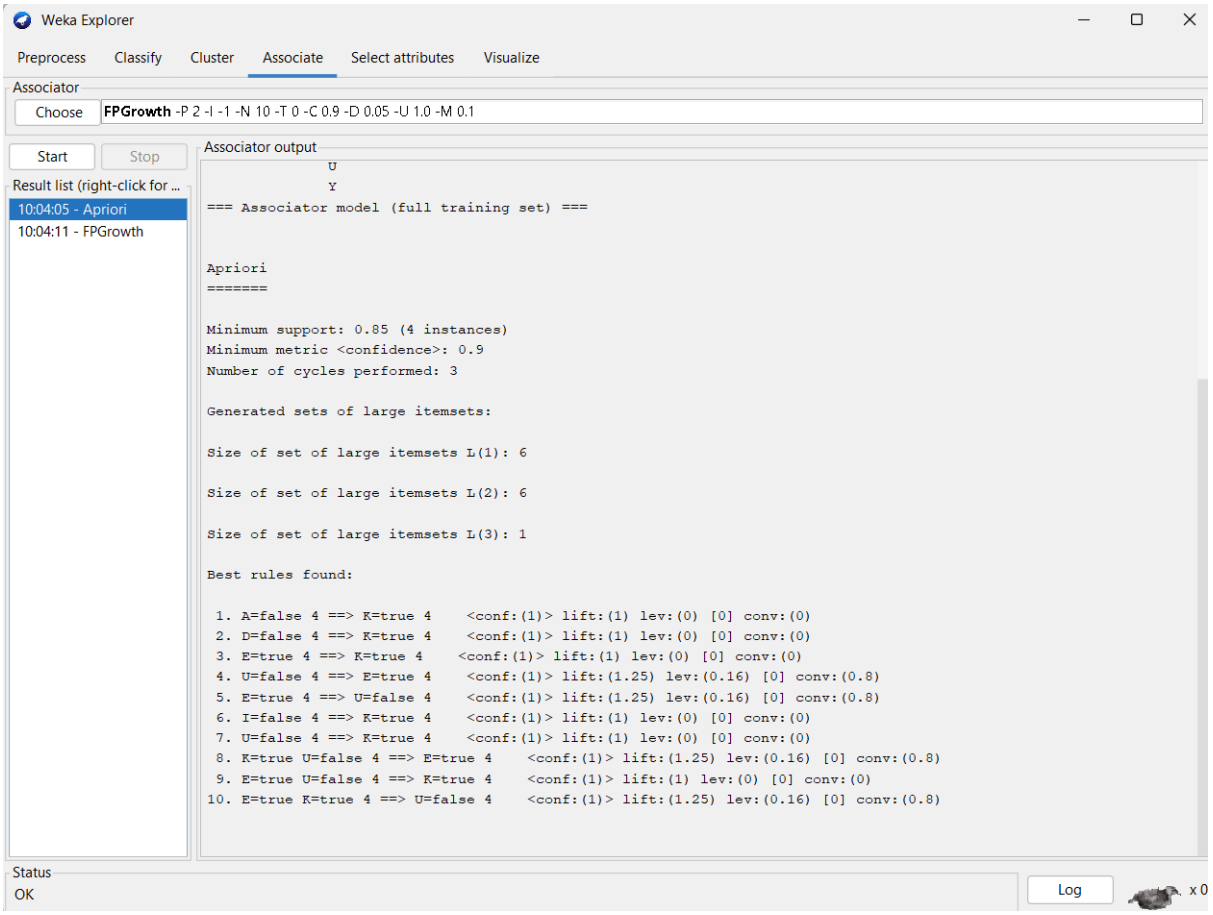
```
var_AvgSpeed <- var(AvgSpeed)
var_TotalTime <- var(TotalTime)

# Results
cat("Standard Deviation of AvgSpeed:", sd_AvgSpeed, "\n")
cat("Standard Deviation of TotalTime:", sd_TotalTime, "\n")
cat("Variance of AvgSpeed:", var_AvgSpeed, "\n")
cat("Variance of TotalTime:", var_TotalTime, "\n")
```

```
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.

> # Data
> AvgSpeed <- c(78, 81, 82, 74, 83, 82, 77, 80, 70)
> TotalTime <- c(39, 37, 36, 42, 35, 36, 40, 38, 46)
> # Standard Deviation
> sd_AvgSpeed <- sd(AvgSpeed)
> sd_TotalTime <- sd(TotalTime)
> # Variance
> var_AvgSpeed <- var(AvgSpeed)
> var_TotalTime <- var(TotalTime)
> # Results
> cat("Standard Deviation of AvgSpeed:", sd_AvgSpeed, "\n")
Standard Deviation of AvgSpeed: 4.304391
> cat("Standard Deviation of TotalTime:", sd_TotalTime, "\n")
Standard Deviation of TotalTime: 3.492054
> cat("Variance of AvgSpeed:", var_AvgSpeed, "\n")
Variance of AvgSpeed: 18.52778
> cat("Variance of TotalTime:", var_TotalTime, "\n")
Variance of TotalTime: 12.19444
>
>
>
```

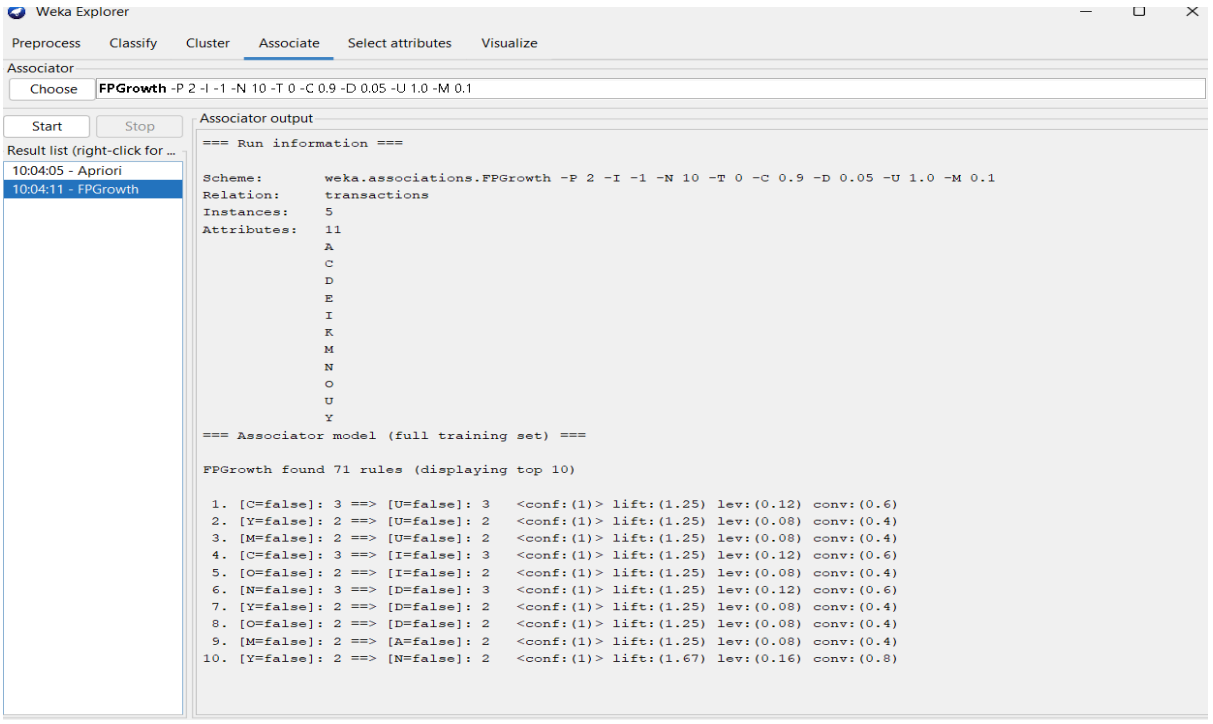
Q13:



Apriori Algorithm:



FP Growth:



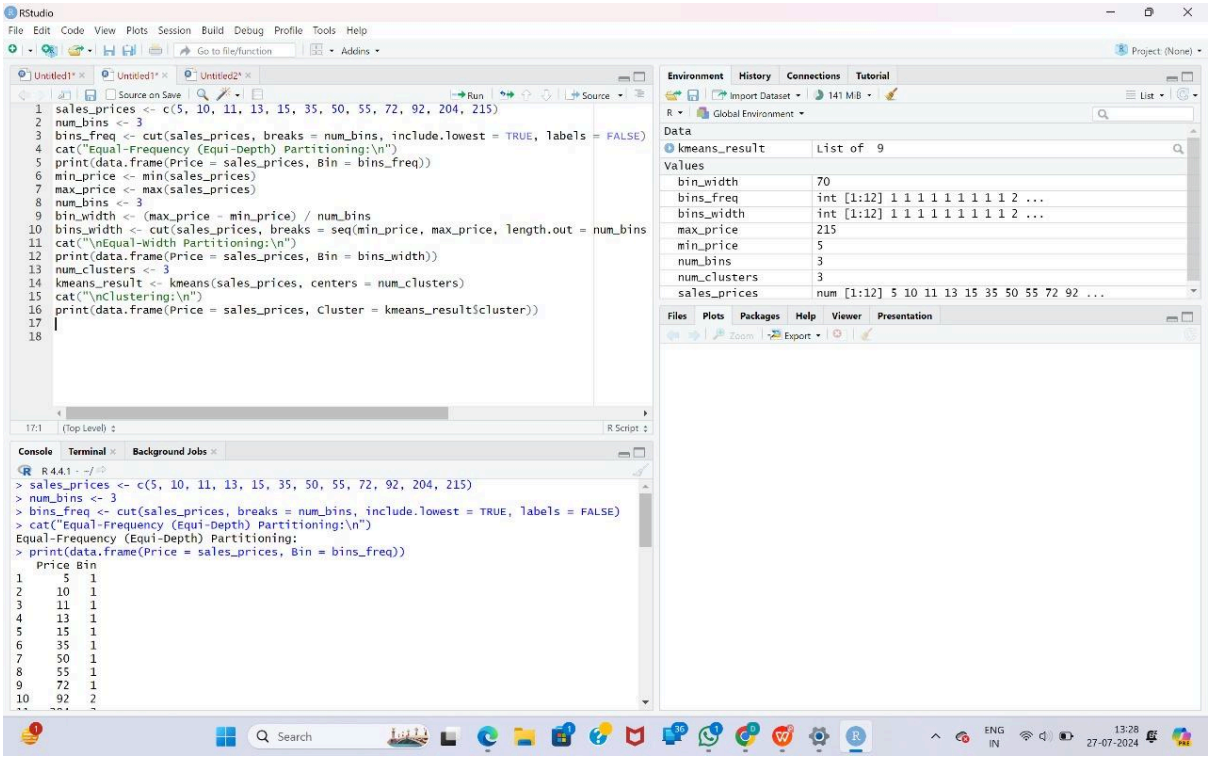
QUESTION BANK

Q1:

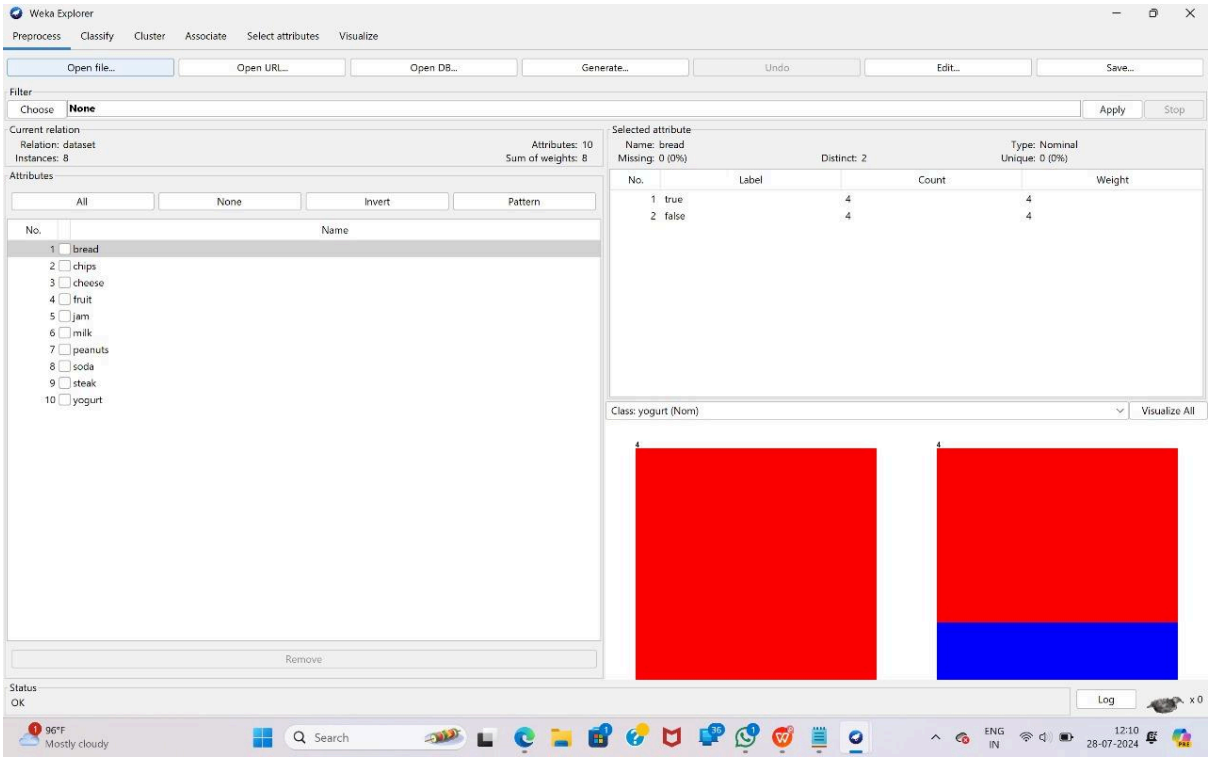
INPUT:

```
sales_prices <- c(5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215)
num_bins <- 3
bins_freq <- cut(sales_prices, breaks = num_bins, include.lowest = TRUE, labels = FALSE)
cat("Equal-Frequency (Equi-Depth) Partitioning:\n")
print(data.frame(Price = sales_prices, Bin = bins_freq))
min_price <- min(sales_prices)
max_price <- max(sales_prices)
num_bins <- 3
bin_width <- (max_price - min_price) / num_bins
bins_width <- cut(sales_prices, breaks = seq(min_price, max_price, length.out = num_bins + 1), include.lowest = TRUE, labels = FALSE)
cat("\nEqual-Width Partitioning:\n")
print(data.frame(Price = sales_prices, Bin = bins_width))
num_clusters <- 3
kmeans_result <- kmeans(sales_prices, centers = num_clusters)
cat("\nClustering:\n")
print(data.frame(Price = sales_prices, Cluster = kmeans_result$cluster))
```

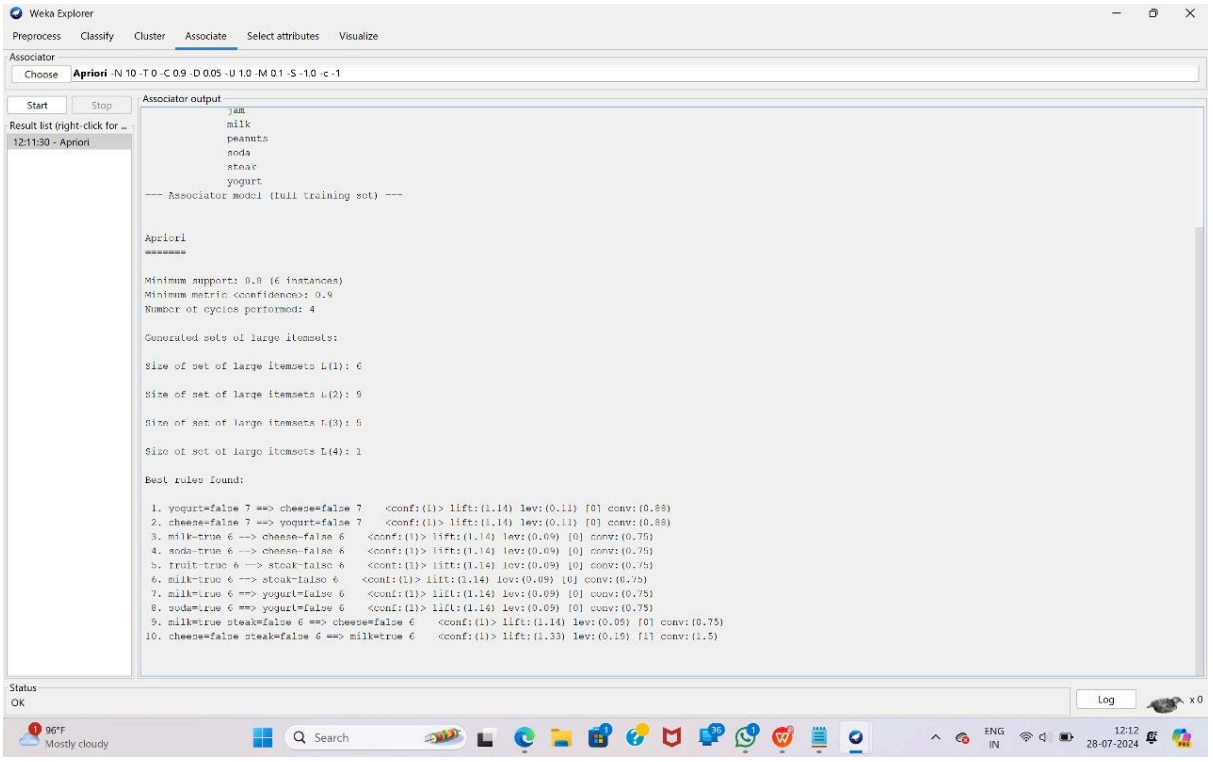
OUTPUT:



Q3:



Apriori Algorithm :

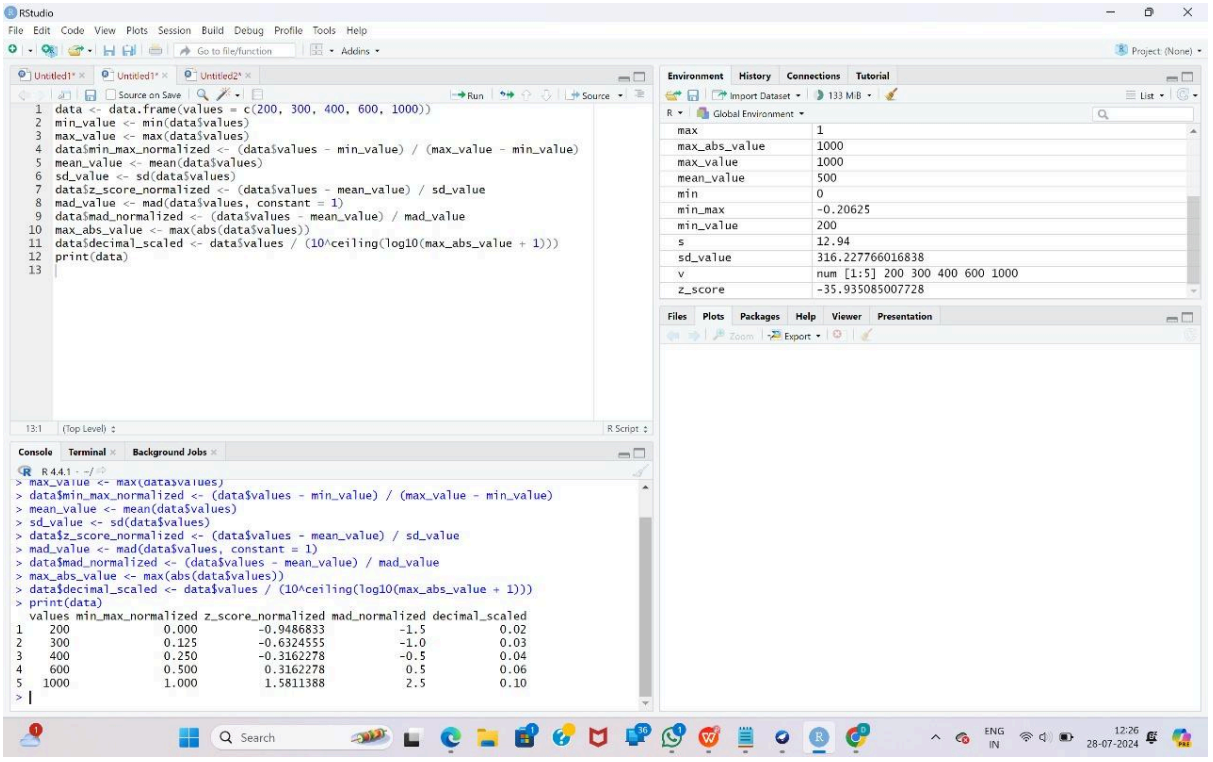


Q4:

INPUT:

```
data <- data.frame(values = c(200, 300, 400, 600, 1000))
min_value <- min(data$values)
max_value <- max(data$values)
data$min_max_normalized <- (data$values - min_value) / (max_value - min_value)
mean_value <- mean(data$values)
sd_value <- sd(data$values)
data$z_score_normalized <- (data$values - mean_value) / sd_value
mad_value <- mad(data$values, constant = 1)
data$mad_normalized <- (data$values - mean_value) / mad_value
max_abs_value <- max(abs(data$values))
data$decimal_scaled <- data$values / (10^ceiling(log10(max_abs_value + 1)))
print(data)
```

Output:

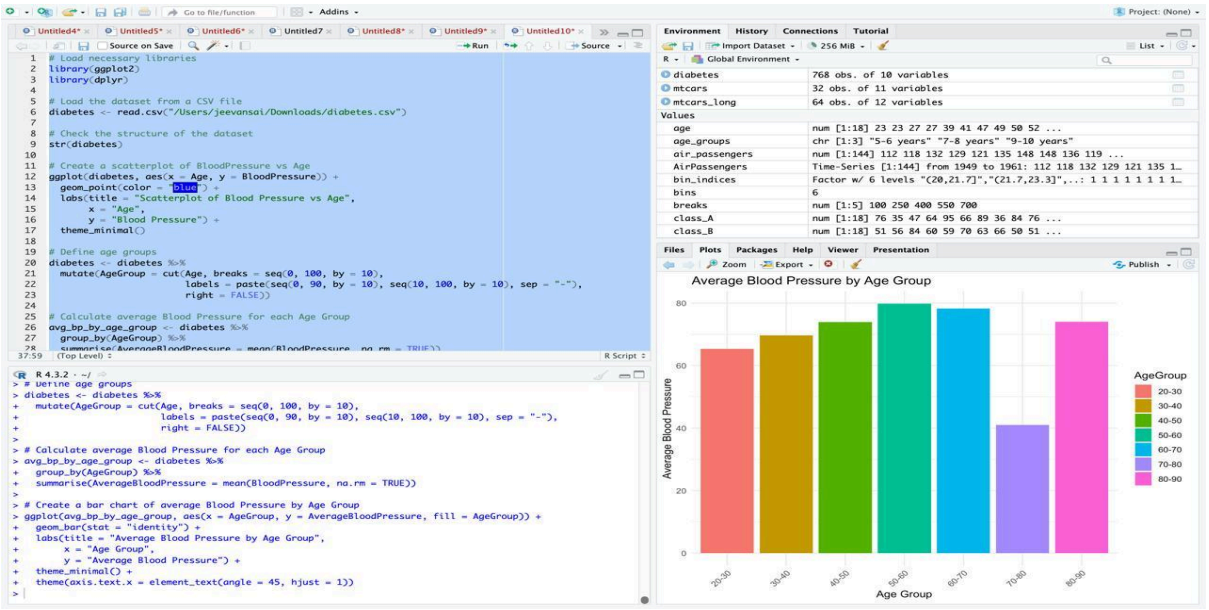


Q5:

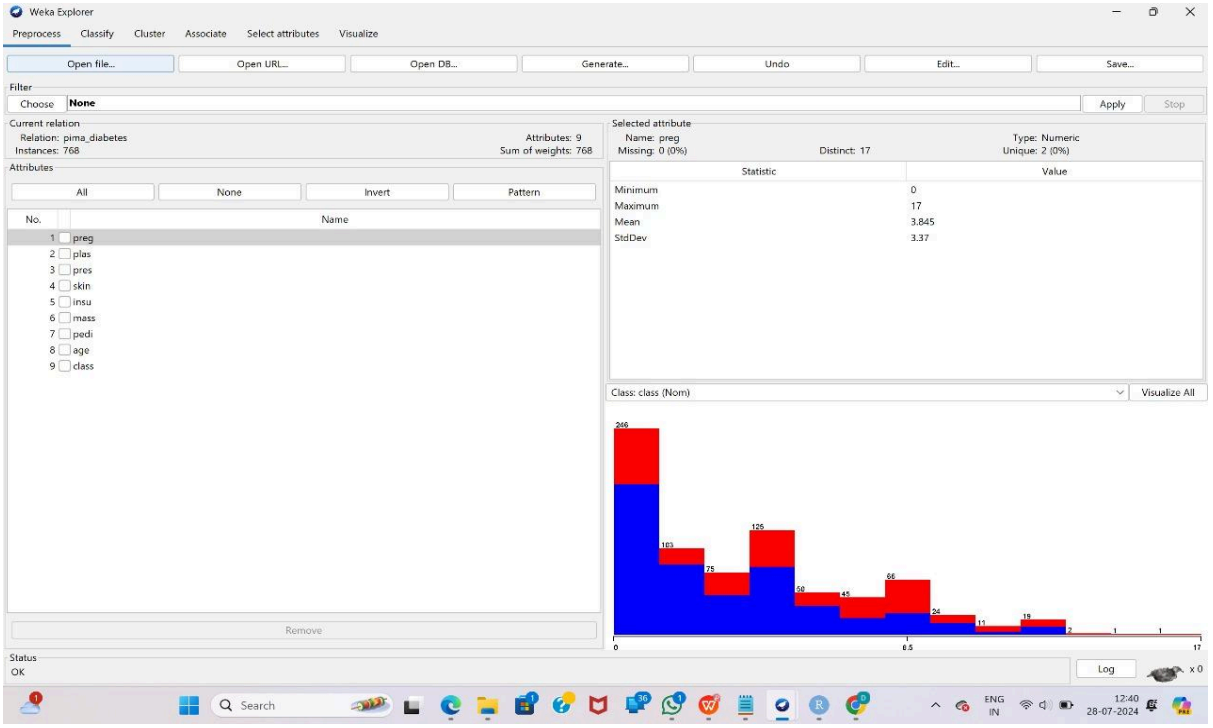
Input:

```
library(ggplot2)
library(dplyr)
diabetes <- read.csv("D:/DWDM/diabetes.csv")
str(diabetes)
ggplot(diabetes, aes(x = Age, y = BloodPressure)) +
  geom_point(color = "blue") +
  labs(title = "Scatterplot of Blood Pressure vs Age",
       x = "Age",
       y = "Blood Pressure") +
  theme_minimal()
diabetes <- diabetes %>%
  mutate(AgeGroup = cut(Age, breaks = seq(0, 100, by = 10),
                        labels = paste(seq(0, 90, by = 10), seq(10, 100, by = 10), sep = "-"),
                        right = FALSE))
avg_bp_by_age_group <- diabetes %>%
  group_by(AgeGroup) %>%
  summarise(AverageBloodPressure = mean(BloodPressure, na.rm = TRUE))
ggplot(avg_bp_by_age_group, aes(x = AgeGroup, y = AverageBloodPressure, fill =
AgeGroup)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Blood Pressure by Age Group",
       x = "Age Group",
       y = "Average Blood Pressure") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

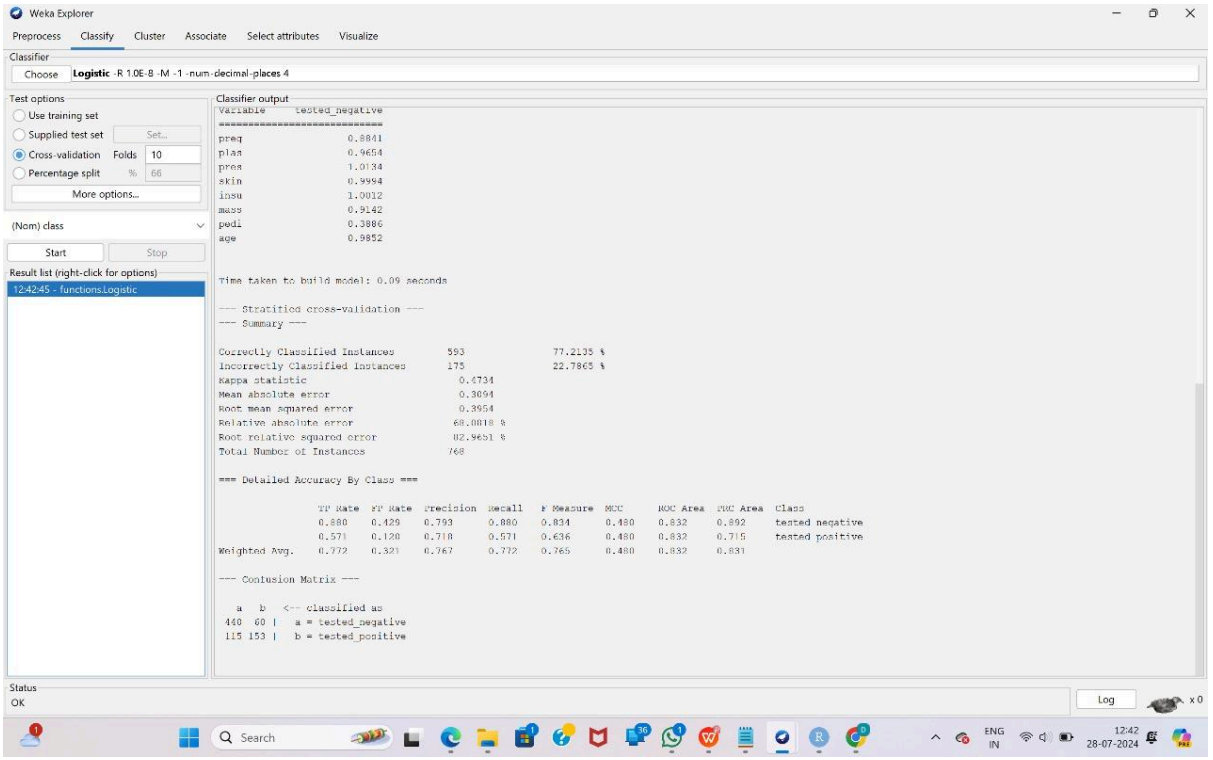
OUTPUT:



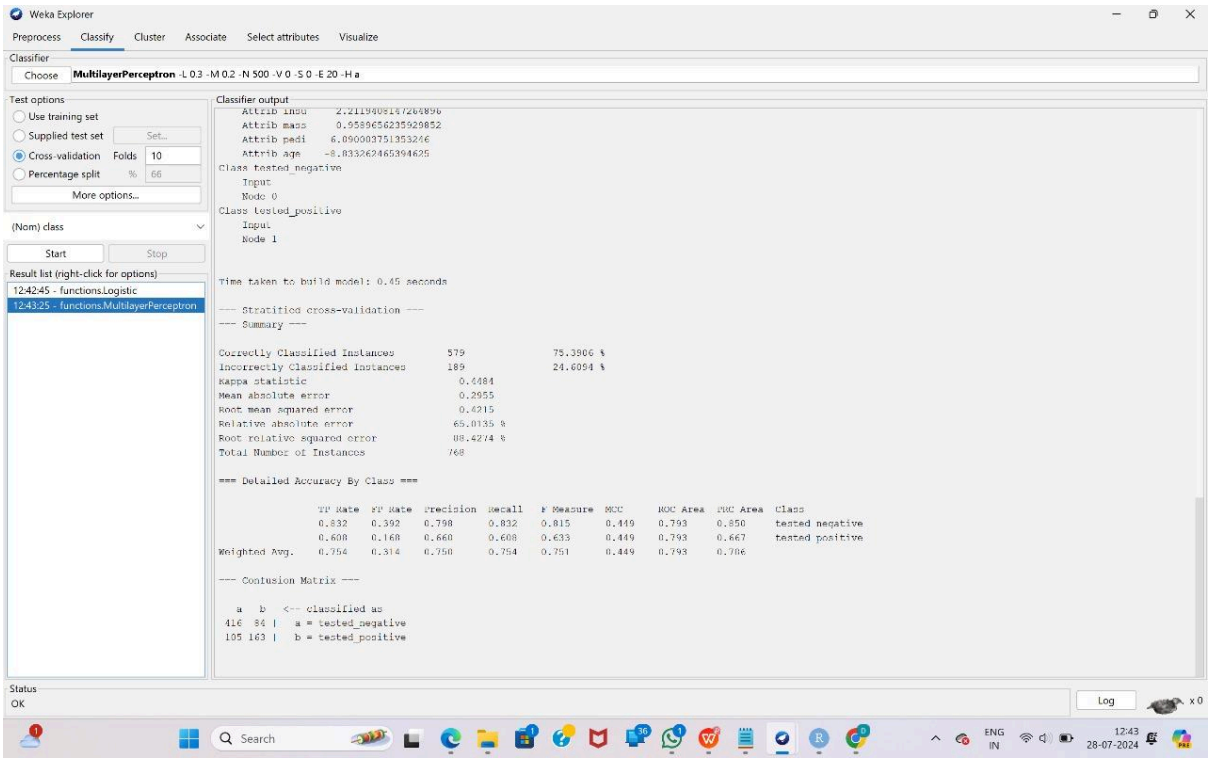
Q6:



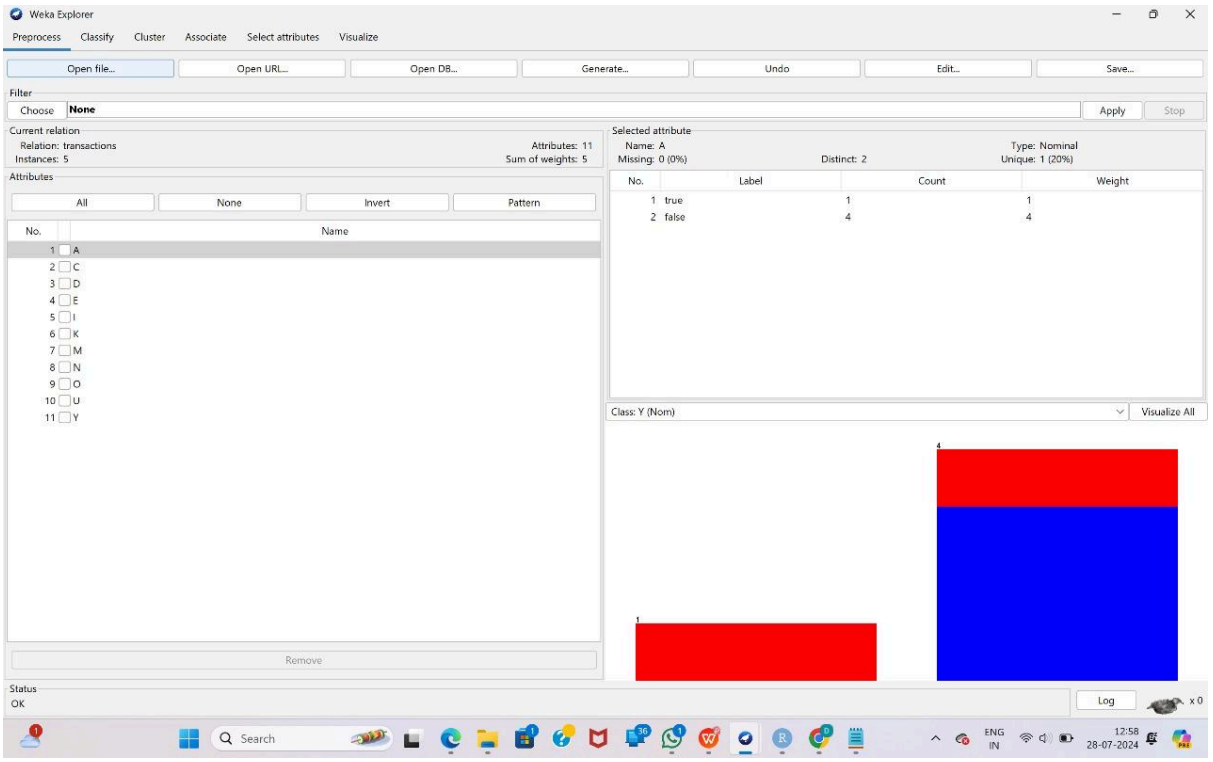
Linear regression:



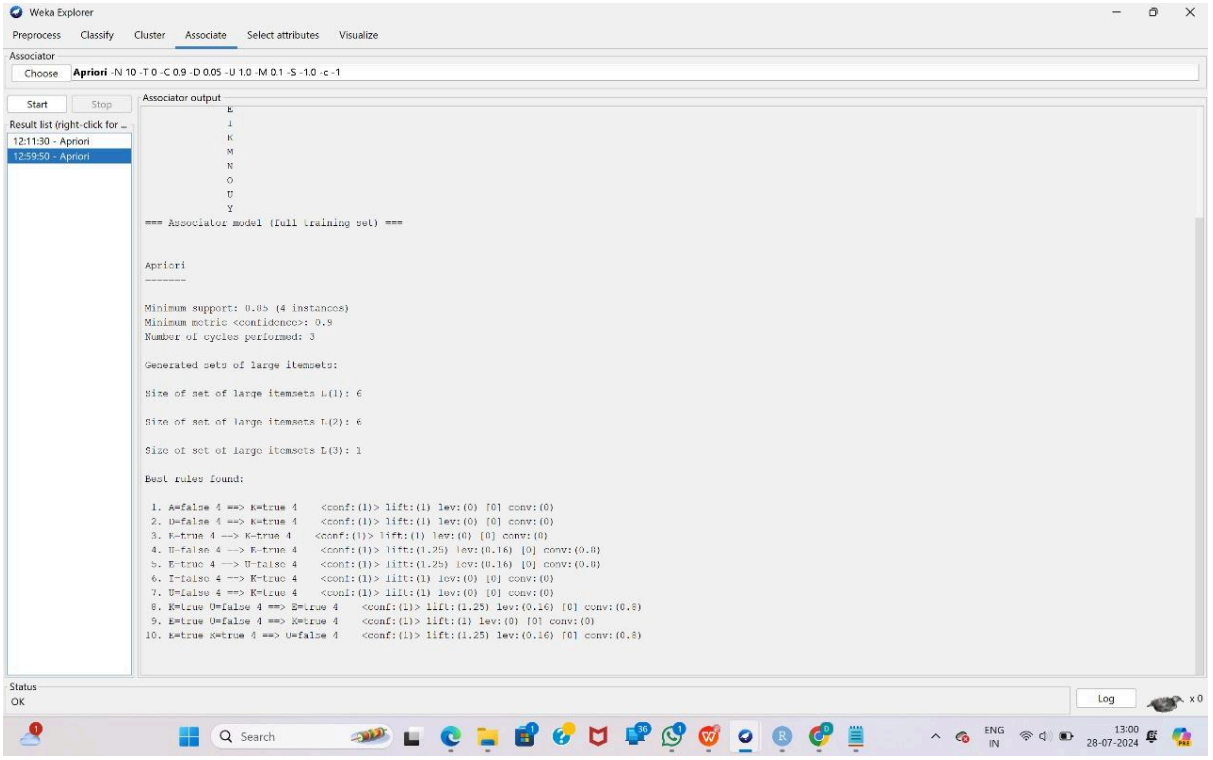
Multiple regression:



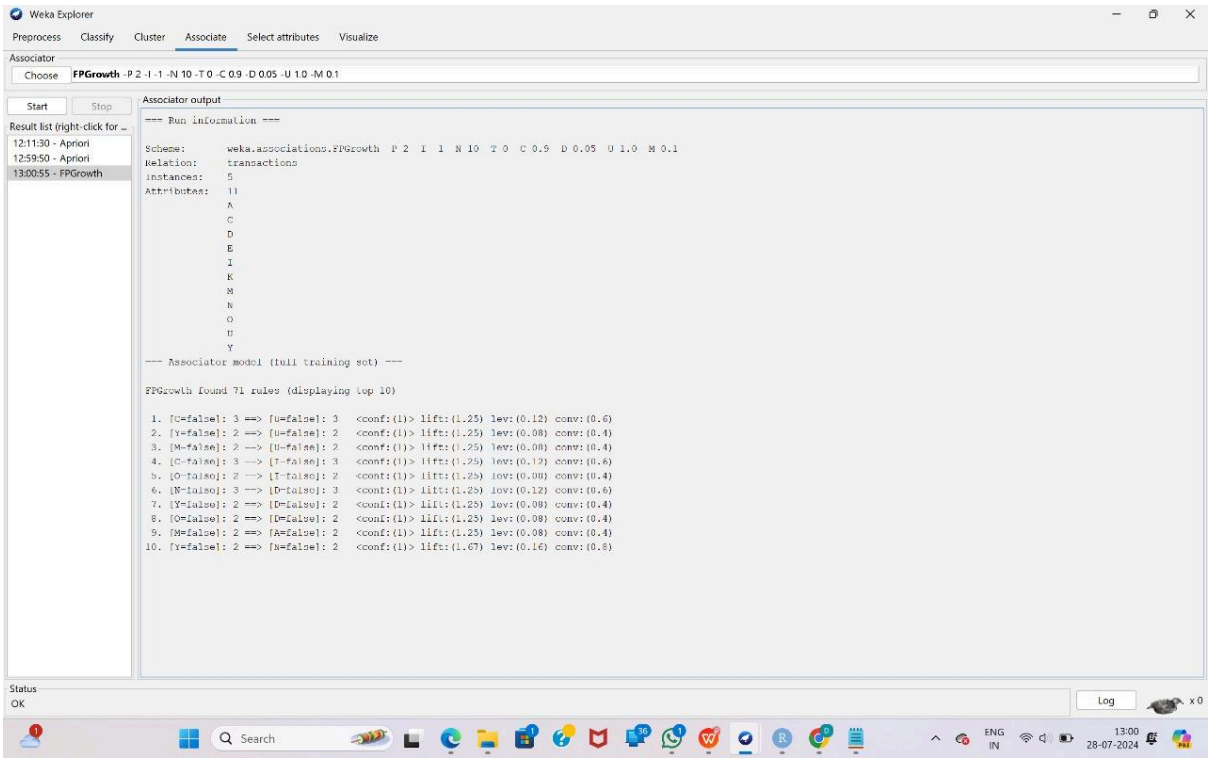
Q7:



Aprioralgorithm:

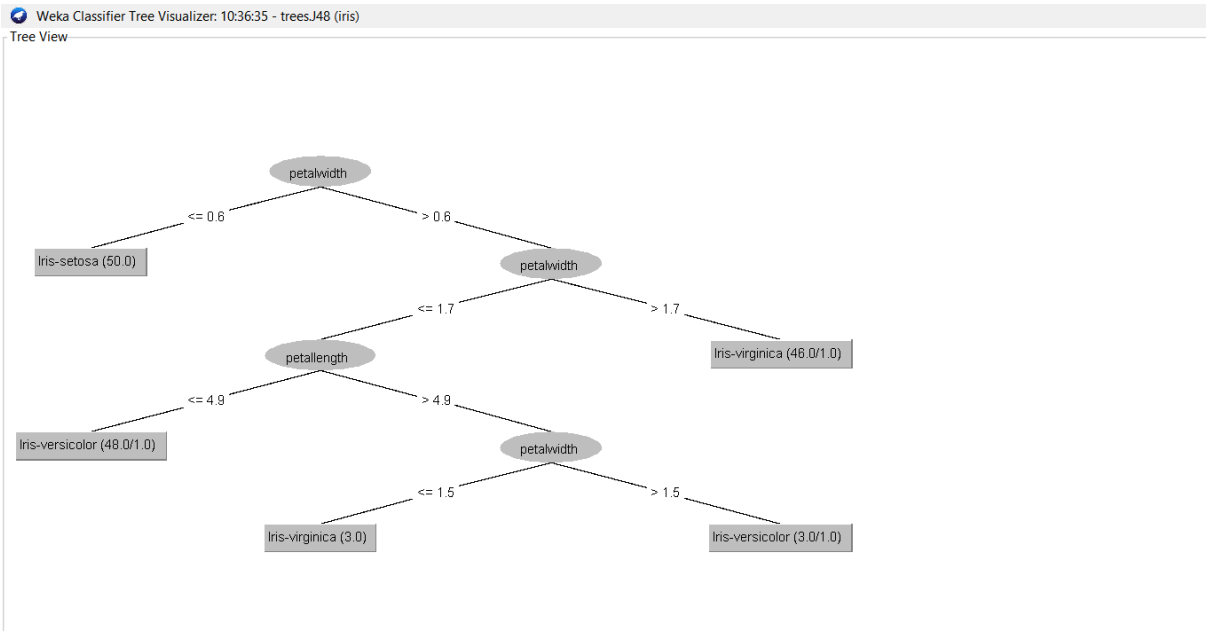


FP Growth:

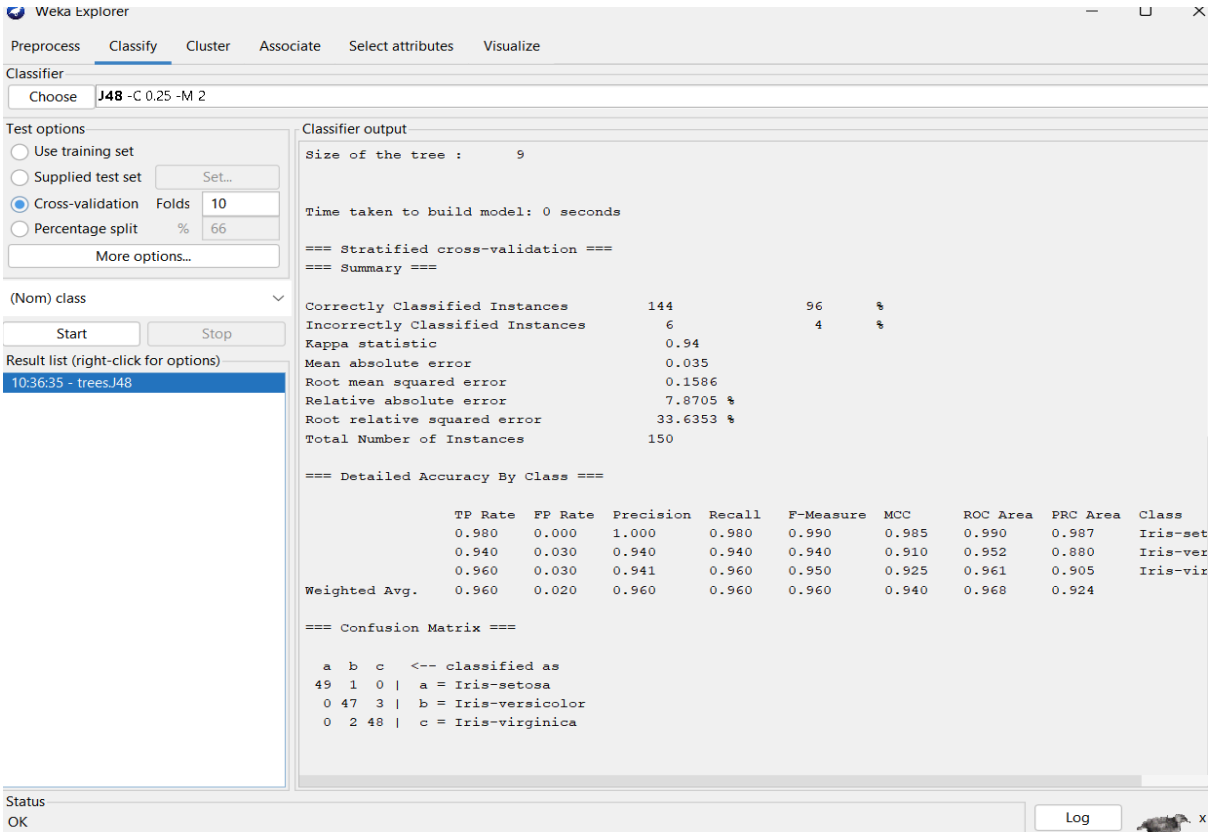


Q8:

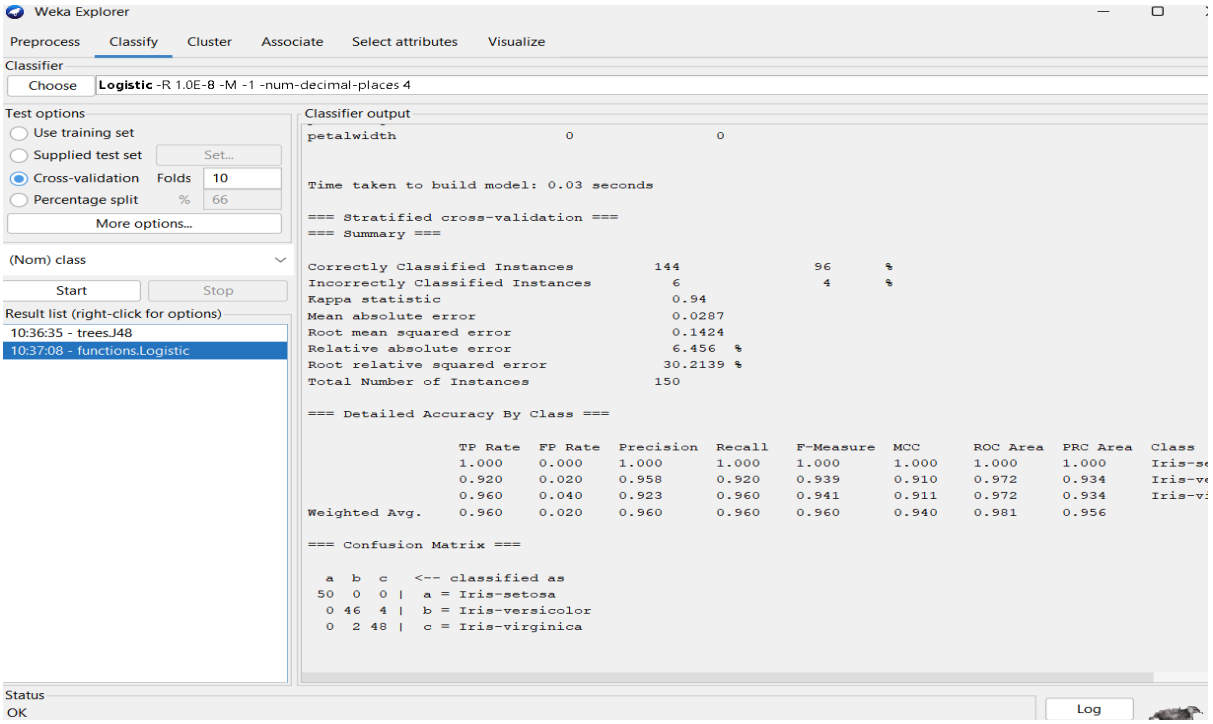
TREE:



PREPROCESS:



Logistic:



Q9:

INPUT:

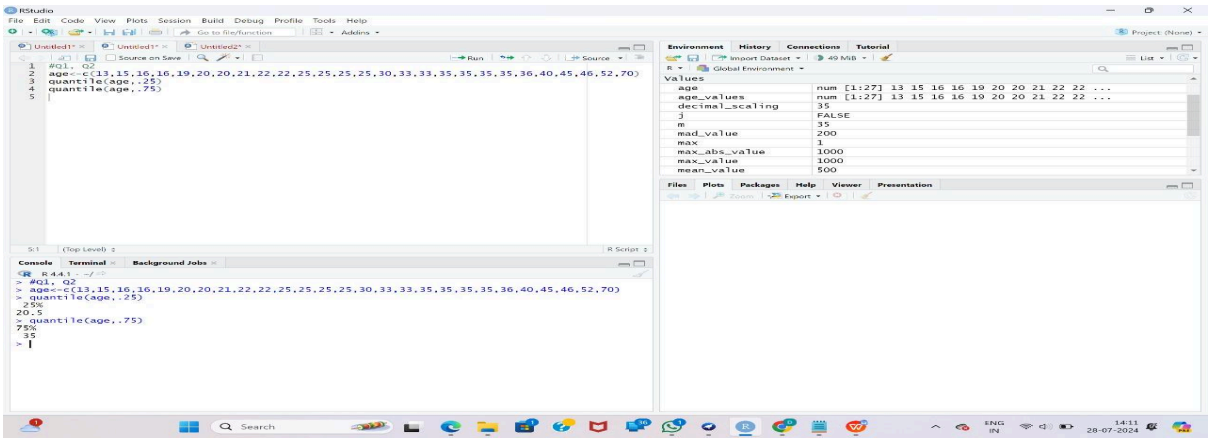
#Q1, Q2

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)

quantile(age,.75)

Output:

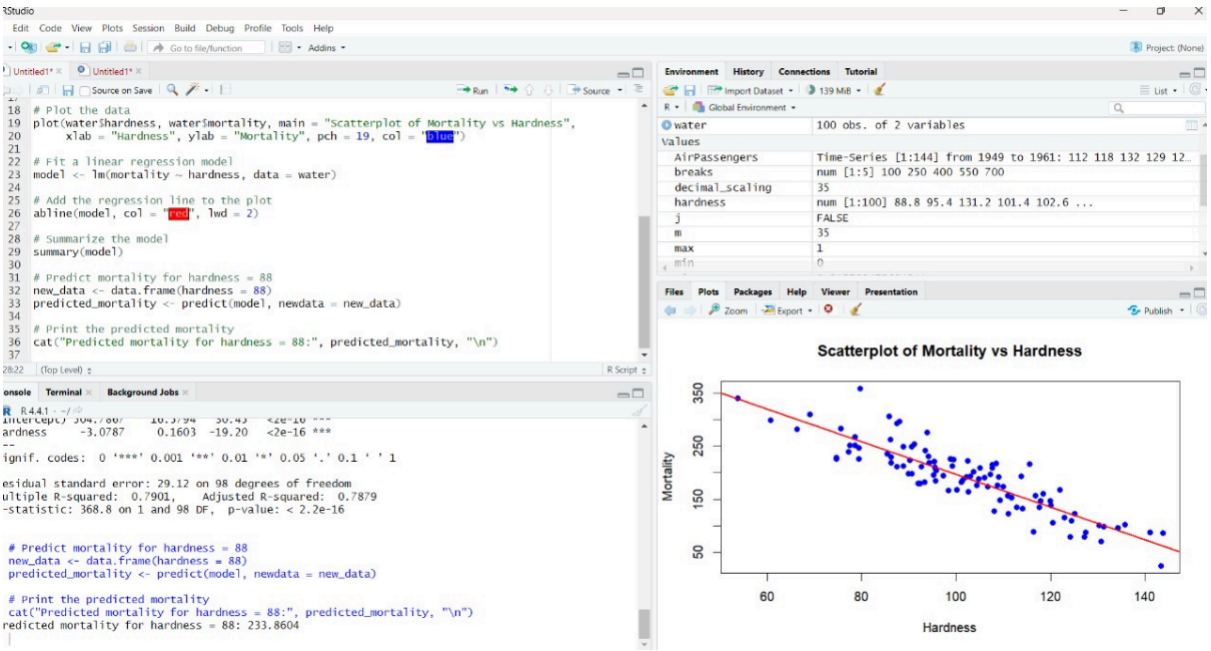


Q10:

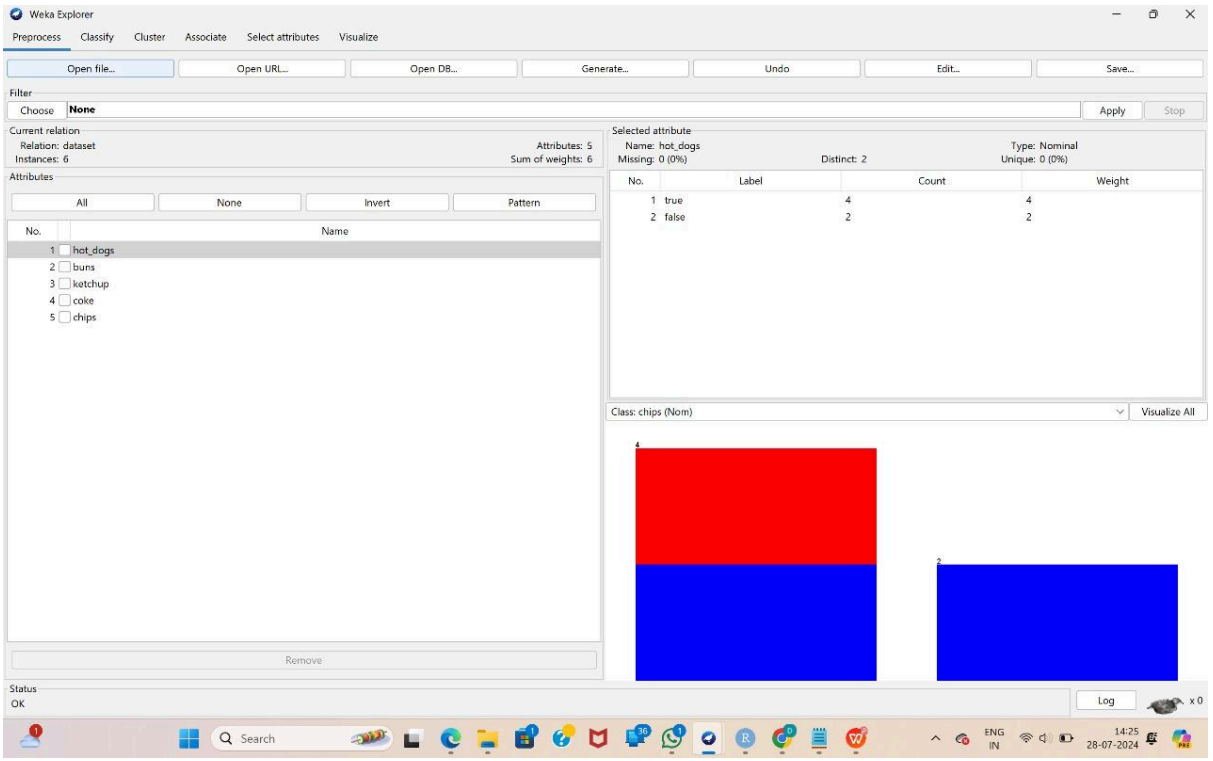
INPUT:

```
data(water, package="datasets")

plot(water$hardness, water$mortality, main="Scatterplot of Mortality vs Hardness",
xlab="Hardness", ylab="Mortality")

model <- lm(mortality ~ hardness, data=water)

abline(model, col="red")

predicted_mortality <- predict(model, newdata=data.frame(hardness=88))

predicted_mortality
```

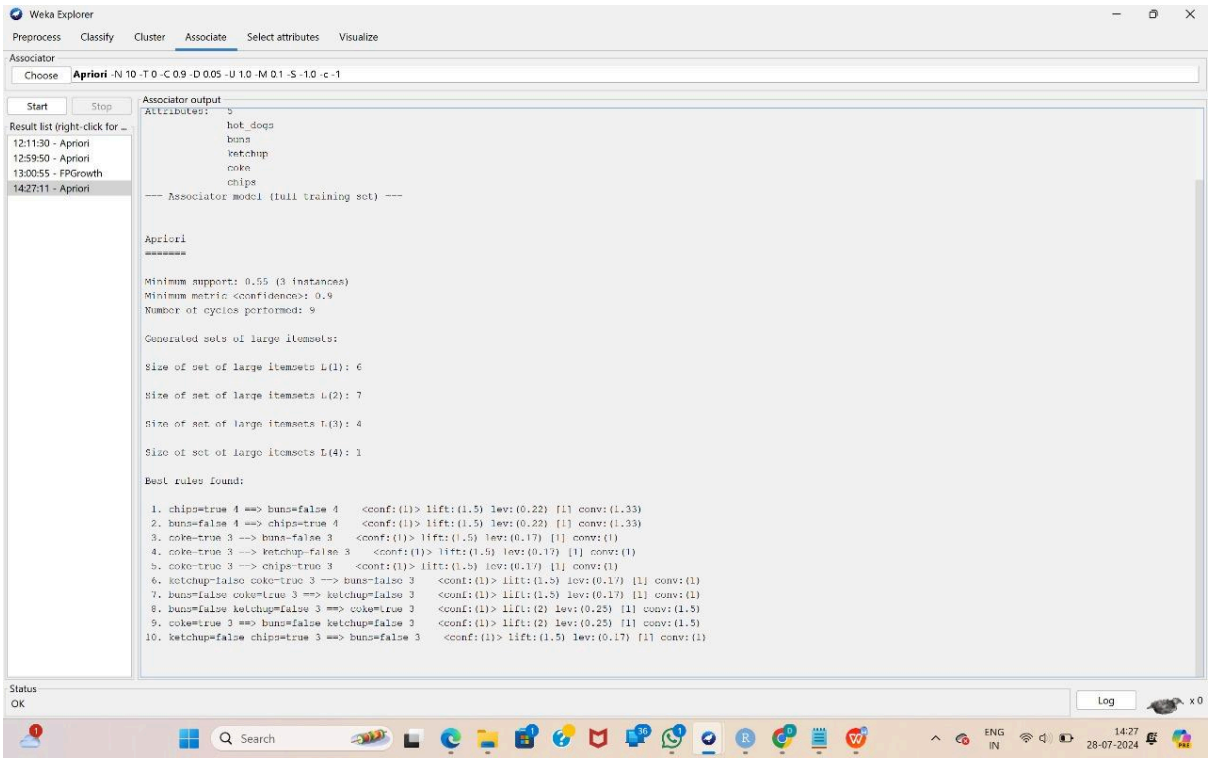
OUTPUT:



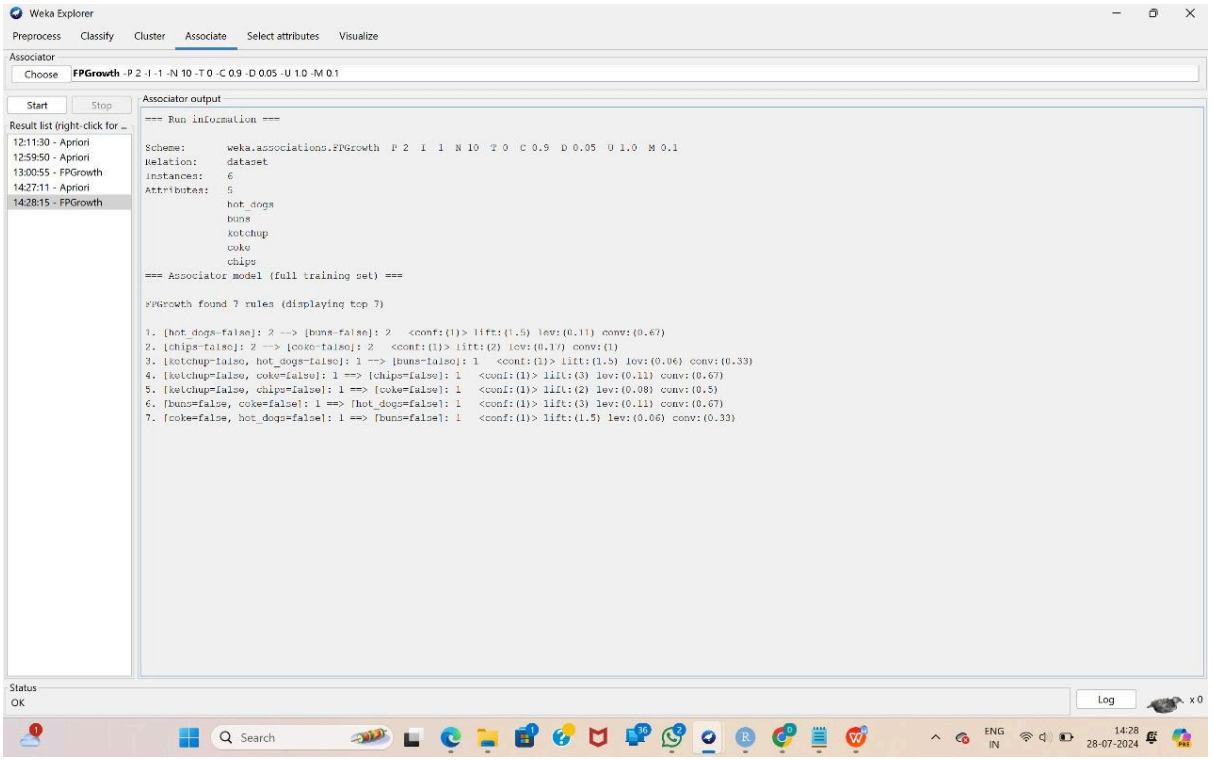
Q11:



Apriori Algorithm:

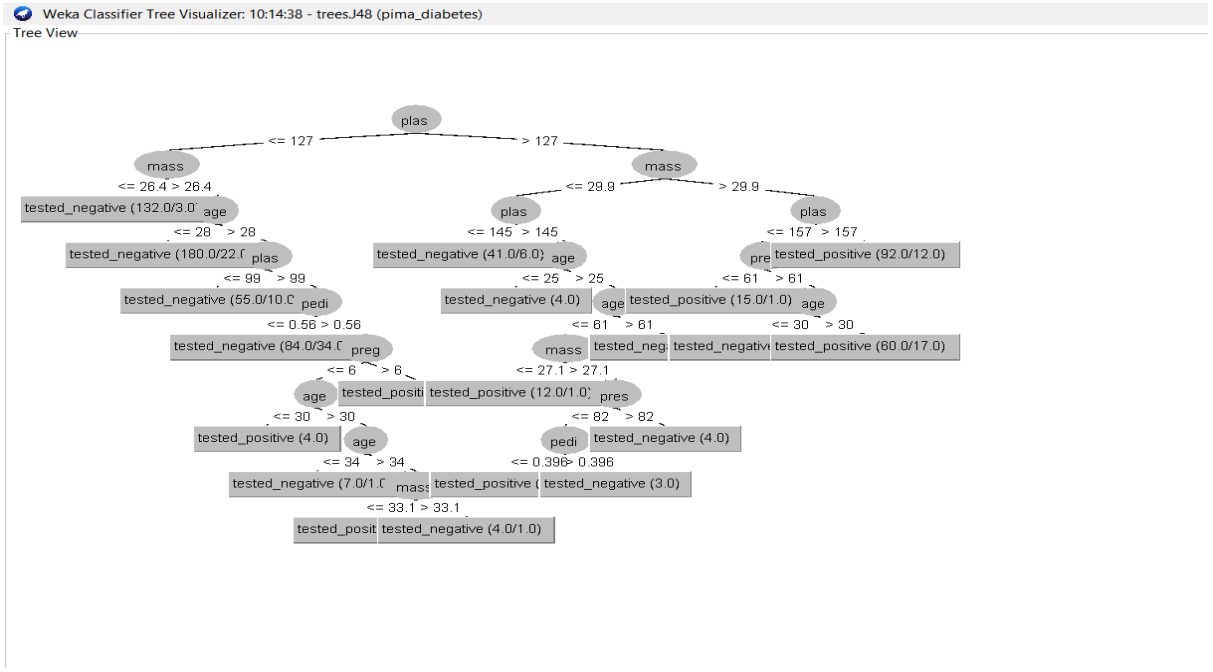


FP Growth:

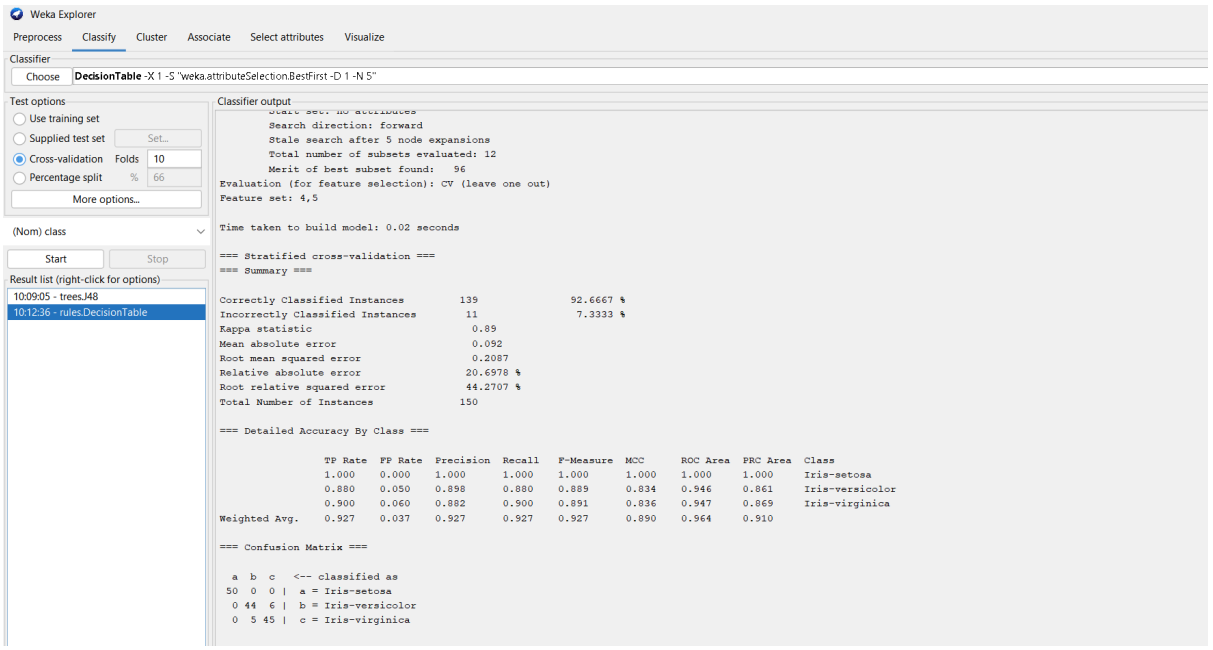


Q12:

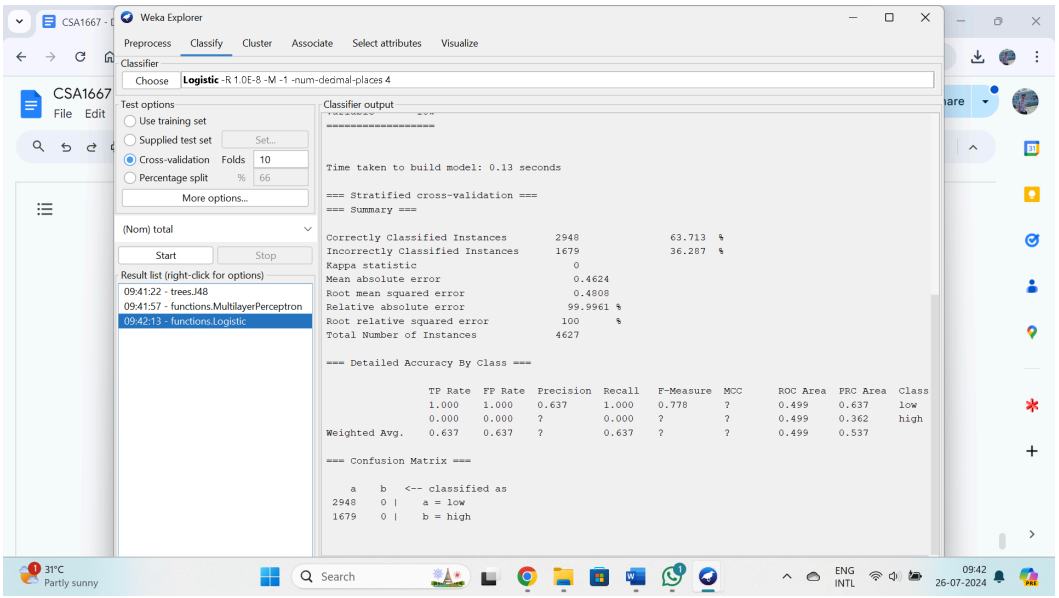
Tree:



Decision Tree:



Logistic:



Q13:

INPUT:

prices <- c(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, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30)

```
library(dplyr)

num_bins <- 3

bin_size <- length(prices) / num_bins

bins <- split(prices, ceiling(seq_along(prices) / bin_size))

print("Equal-frequency bins:")

print(bins)

bin_means <- lapply(bins, mean)

print("Smoothing by bin means:")

print(bin_means)

bin_boundaries <- lapply(bins, function(bin) {

  min_val <- min(bin)

  max_val <- max(bin)

  sapply(bin, function(x) {

    if (x - min_val < max_val - x) min_val else max_val

  })

})

print("Smoothing by bin boundaries:")

print(bin_boundaries)

bins_unlisted <- unlist(bins)
```

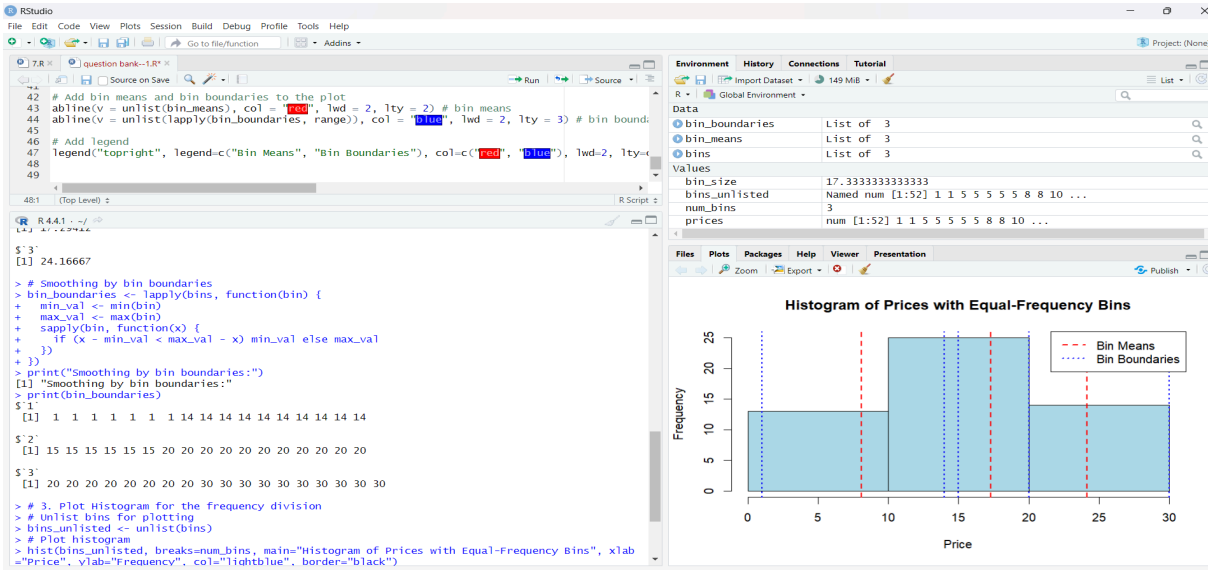
hist(bins_unlisted, breaks=num_bins, main="Histogram of Prices with Equal-Frequency Bins", xlab="Price", ylab="Frequency", col="lightblue", border="black")

abline(v = unlist(bin_means), col = "red", lwd = 2, lty = 2) # bin means

abline(v = unlist(lapply(bin_boundaries, range)), col = "blue", lwd = 2, lty = 3) # bin boundaries

legend("topright", legend=c("Bin Means", "Bin Boundaries"), col=c("red", "blue"), lwd=2, lty=c(2, 3))

OUTPUT:



Q14:

INPUT:

class_A <- c(76, 35, 47, 64, 95, 66, 89, 36, 84)

class_B <- c(51, 56, 84, 60, 59, 70, 63, 66, 50)

mean_A <- mean(class_A)

mean_B <- mean(class_B)

median_A <- median(class_A)

median_B <- median(class_B)

range_A <- range(class_A)

range_B <- range(class_B)

range_A_diff <- diff(range_A)

range_B_diff <- diff(range_B)

cat("Class A - Mean:", mean_A, " Median:", median_A, " Range:", range_A_diff, "\n")

cat("Class B - Mean:", mean_B, " Median:", median_B, " Range:", range_B_diff, "\n")

boxplot(class_A, class_B, names=c("Class A", "Class B"), main="Comparison of Class A and Class B Performance", ylab="Scores", col=c("lightblue", "lightgreen"))

abline(h=mean_A, col="blue", lty=2)

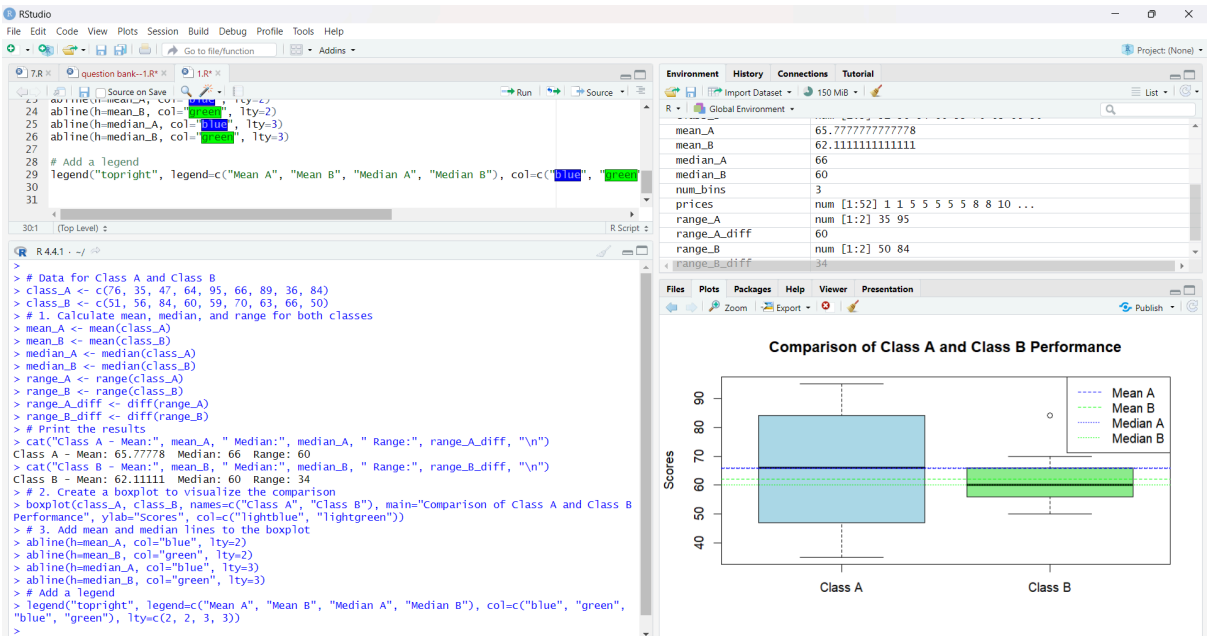
abline(h=mean_B, col="green", lty=2)

abline(h=median_A, col="blue", lty=3)

abline(h=median_B, col="green", lty=3)

legend("topright", legend=c("Mean A", "Mean B", "Median A", "Median B"), col=c("blue", "green", "blue", "green"), lty=c(2, 2, 3, 3))

OUTPUT:



Q16:

```
library(cluster)     # For clustering algorithms
library(factoextra)  # For clustering visualization
library(datasets)    # For sample datasets
set.seed(123)
customer_data <- data.frame(
  percentage_opened = runif(100, 0, 100),  # Percentage of emails opened
  clicks_per_email = runif(100, 0, 10),    # Number of clicks per email
  time_spent_viewing = runif(100, 0, 5)    # Time spent viewing email (in minutes)
)
customer_data_scaled <- scale(customer_data)
set.seed(123)
kmeans_result <- kmeans(customer_data_scaled, centers = 3, nstart = 25)
distances <- dist(customer_data_scaled)
hclust_result <- hclust(distances, method = "ward.D2")
cutree_result <- cutree(hclust_result, k = 3)
silhouette_kmeans <- silhouette(kmeans_result$cluster, distances)
silhouette_hclust <- silhouette(cutree_result, distances)
avg_silhouette_kmeans <- mean(silhouette_kmeans[, 3])
avg_silhouette_hclust <- mean(silhouette_hclust[, 3])
cat("Average Silhouette Score for k-means: ", avg_silhouette_kmeans, "\n")
cat("Average Silhouette Score for Hierarchical Clustering: ", avg_silhouette_hclust,
"\n")
fviz_cluster(kmeans_result, data = customer_data_scaled, geom = "point", stand =
FALSE, main = "K-means Clustering")
```

fviz_dend(hclust_result, k = 3, rect = TRUE, main = "Hierarchical Clustering Dendrogram")

if (avg_silhouette_kmeans > avg_silhouette_hclust) {

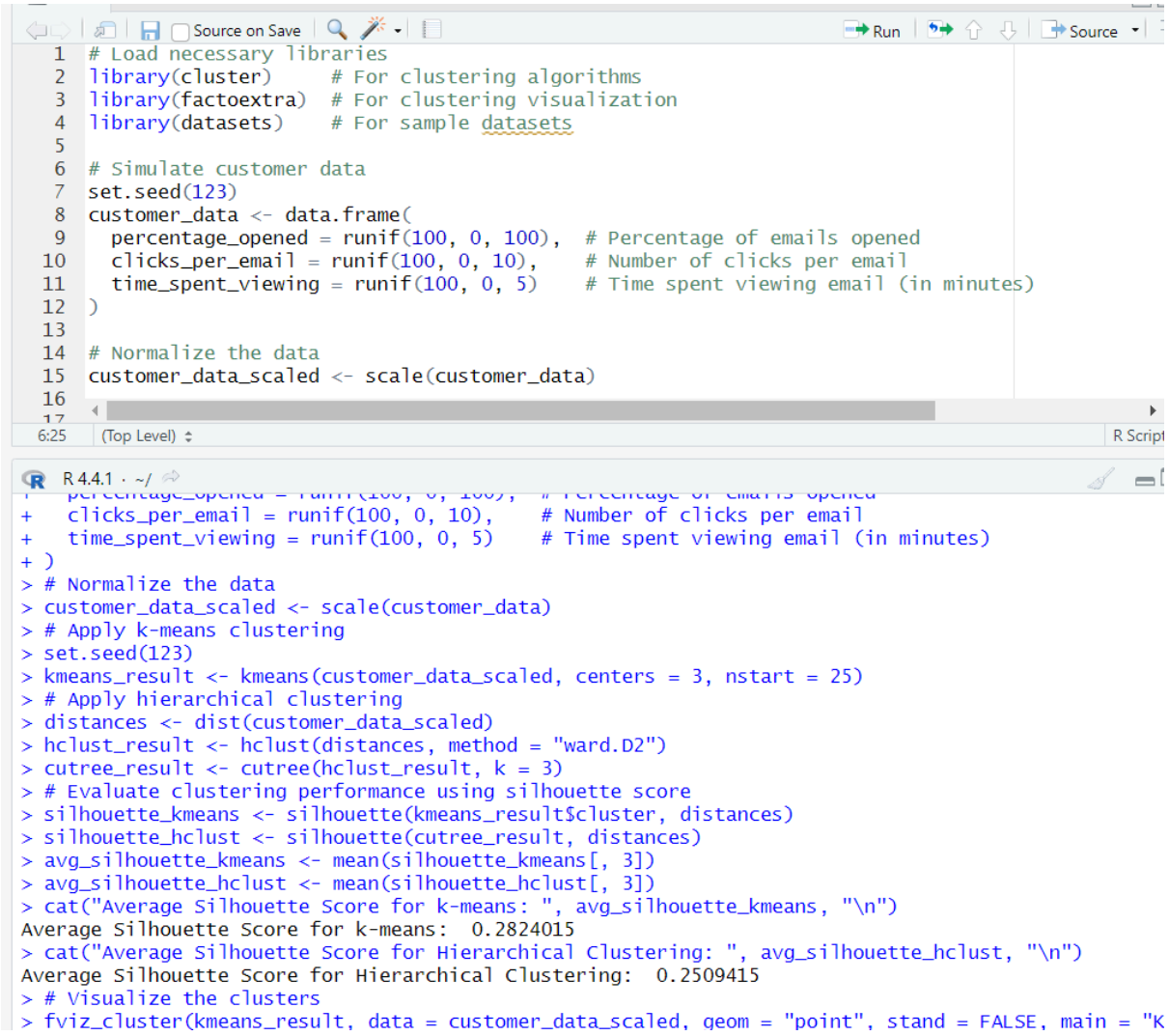
cat("K-means clustering performed better based on the silhouette score.\n")

} else {

cat("Hierarchical clustering performed better based on the silhouette score.\n")

}

OUTPUT:



```
# Load necessary libraries
library(cluster)     # For clustering algorithms
library(factoextra)  # For clustering visualization
library(datasets)    # For sample datasets

# Simulate customer data
set.seed(123)
customer_data <- data.frame(
  percentage_opened = runif(100, 0, 100),  # Percentage of emails opened
  clicks_per_email = runif(100, 0, 10),    # Number of clicks per email
  time_spent_viewing = runif(100, 0, 5)    # Time spent viewing email (in minutes)
)

# Normalize the data
customer_data_scaled <- scale(customer_data)
```

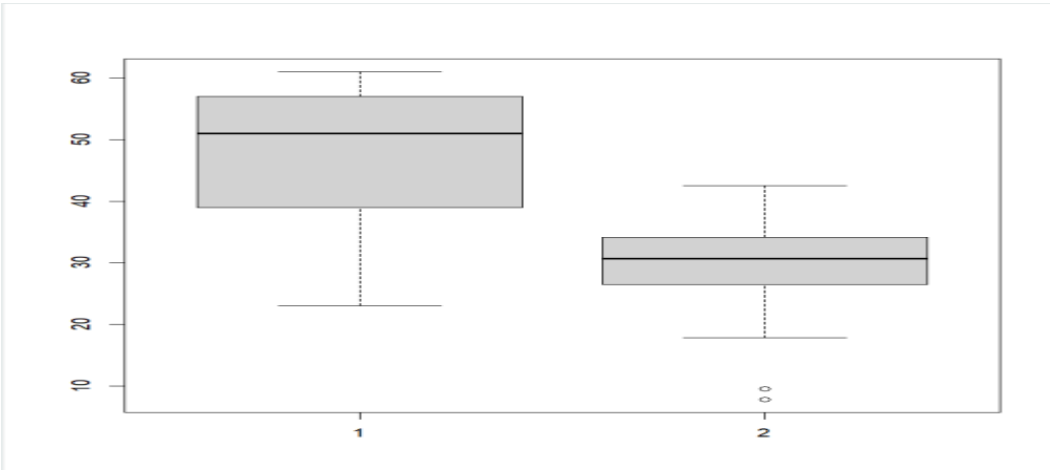
```
+   clicks_per_email = runif(100, 0, 10),    # Number of clicks per email
+   time_spent_viewing = runif(100, 0, 5)    # Time spent viewing email (in minutes)
+ )
> # Normalize the data
> customer_data_scaled <- scale(customer_data)
> # Apply k-means clustering
> set.seed(123)
> kmeans_result <- kmeans(customer_data_scaled, centers = 3, nstart = 25)
> # Apply hierarchical clustering
> distances <- dist(customer_data_scaled)
> hclust_result <- hclust(distances, method = "ward.D2")
> cutree_result <- cutree(hclust_result, k = 3)
> # Evaluate clustering performance using silhouette score
> silhouette_kmeans <- silhouette(kmeans_result$cluster, distances)
> silhouette_hclust <- silhouette(cutree_result, distances)
> avg_silhouette_kmeans <- mean(silhouette_kmeans[, 3])
> avg_silhouette_hclust <- mean(silhouette_hclust[, 3])
> cat("Average Silhouette Score for k-means: ", avg_silhouette_kmeans, "\n")
Average Silhouette Score for k-means:  0.2824015
> cat("Average Silhouette Score for Hierarchical Clustering: ", avg_silhouette_hclust, "\n")
Average Silhouette Score for Hierarchical Clustering:  0.2509415
> # Visualize the clusters
> fviz_cluster(kmeans_result, data = customer_data_scaled, geom = "point", stand = FALSE, main = "K
```

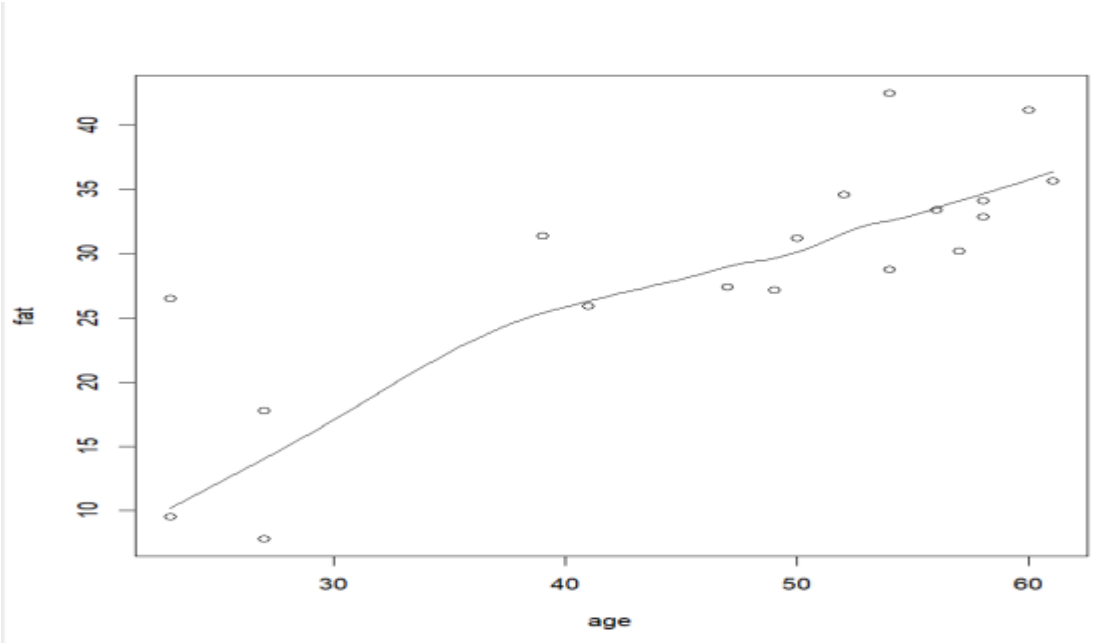
Q17:

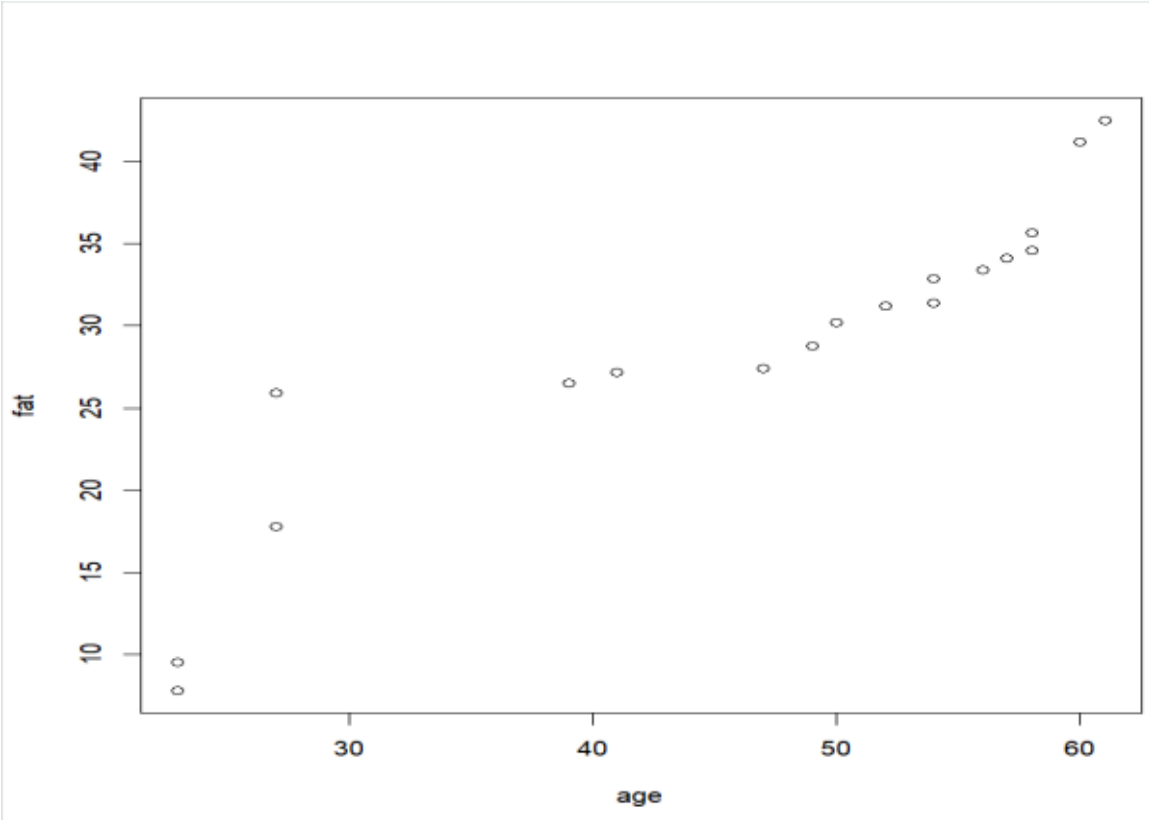
INPUT:

```
age<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)

fat<-c(9.5,26.5,7.8,17.8,31.4,25.9,27.4,27.2,31.2,34.6,42.5,28.8,33.4,30.2,34.1,32.9,41.2,35.7)

mean(age)

median(age)

sd(age)

mean(fat)

median(fat)

sd(fat)

#boxplot

boxplot(age,fat)

#scatter plot

scatter.smooth(age,fat)

#qplot

qqplot(age,fat)
```

Output:




fat
10
15
20
25
30
35
40
30
40
50
60
age


fat
10
15
20
25
30
35
40
30
40
50
60
age

Q18:

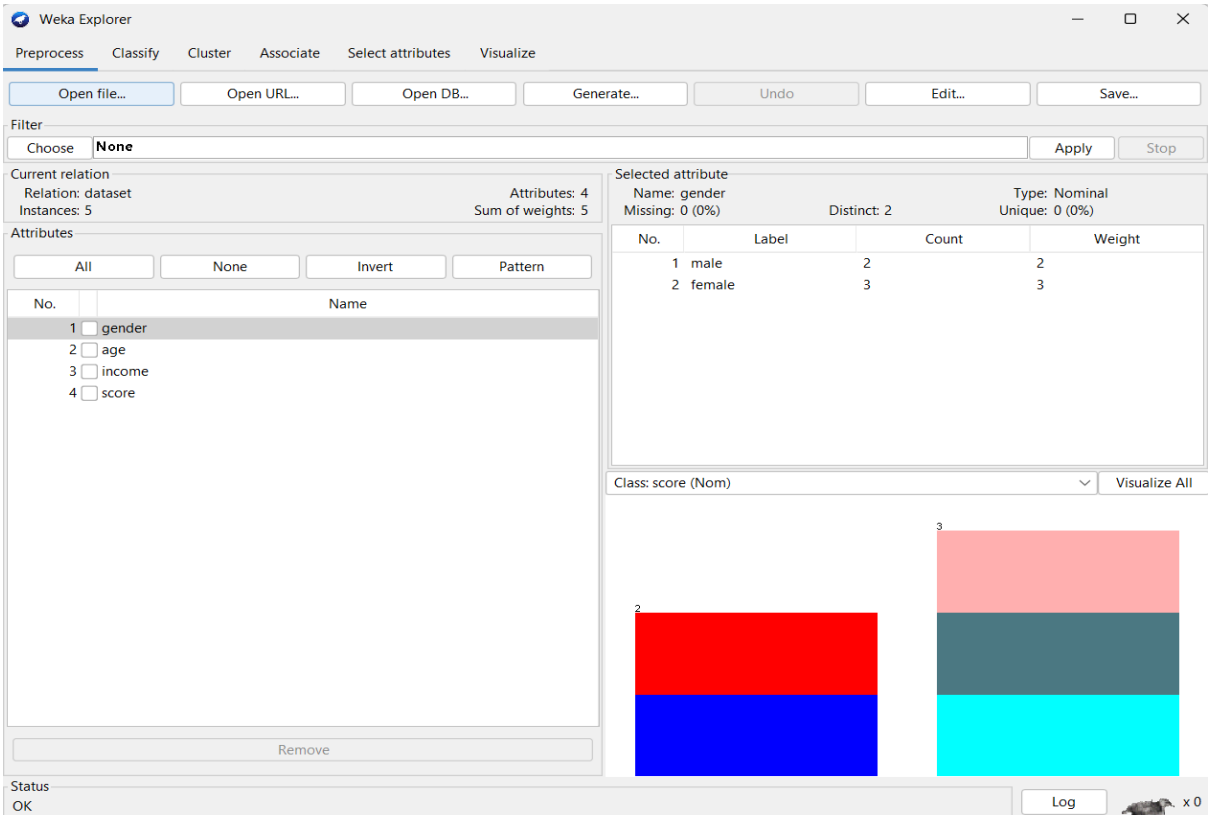
INPUT:

```
v<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)

min<-0

max<-1

#min_max

min_max=((35-min(v))/(max(v)-min(v)))

print(min_max)

#z-score

m=mean(v)

s<-12.94

z_score=(35-m)/s

print(z_score)

#decimal scaling

m<-35

j=max(m)<1

decimal_scaling=m/10^j

print(decimal_scaling)
```

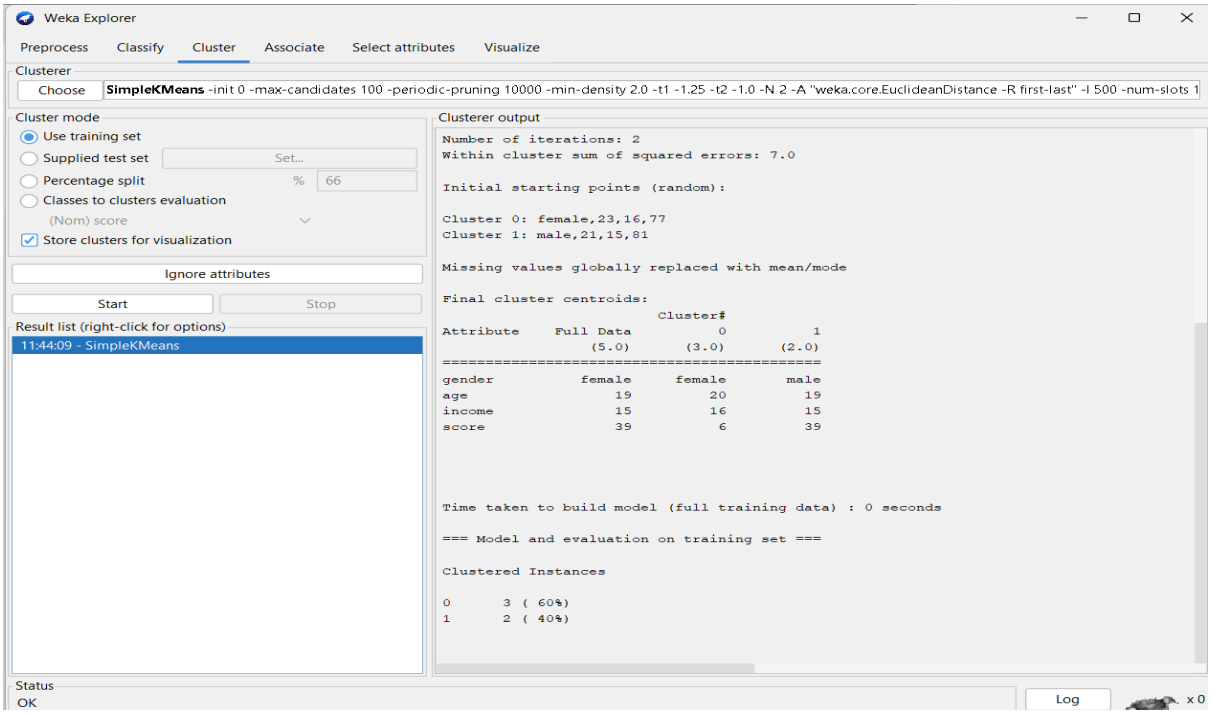
Output:

```
print(min_max)
] 0.3157895
#z-score
m=mean(v)
s<-12.94
z_score=(35-m)/s
print(z_score)
] -0.8844238
#decimal scaling
m<-35
j=max(m)<1
decimal_scaling=m/10^j
print(decimal_scaling)
] 35
```

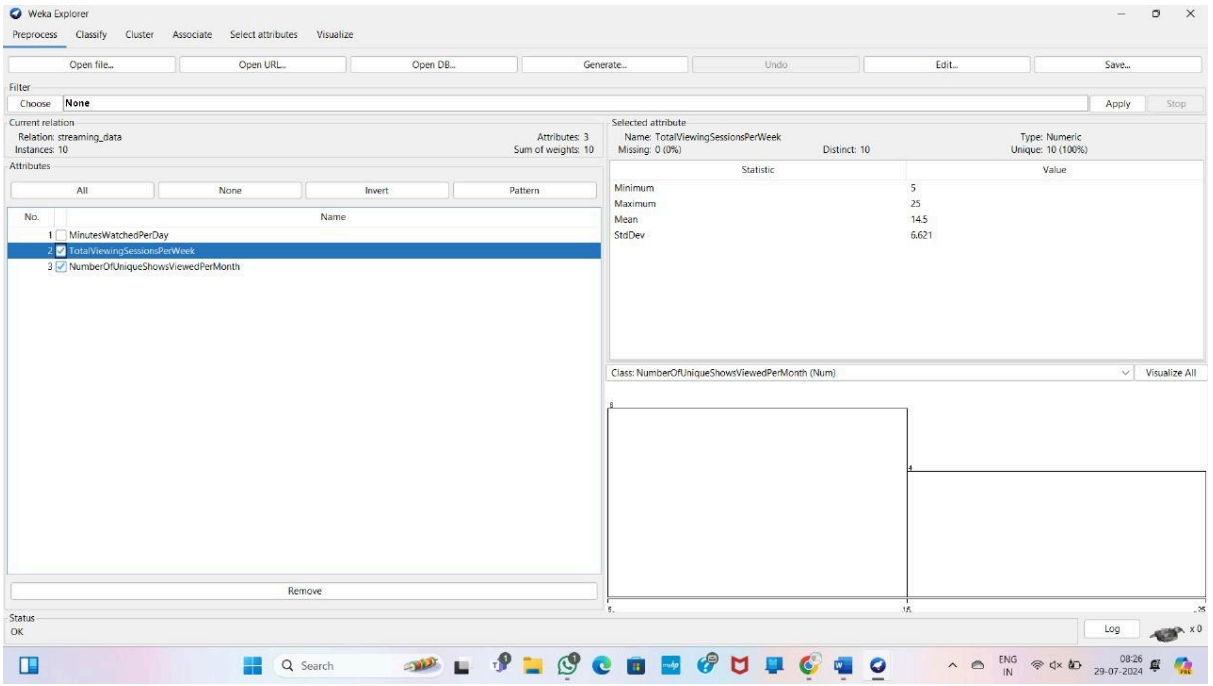
Q19:

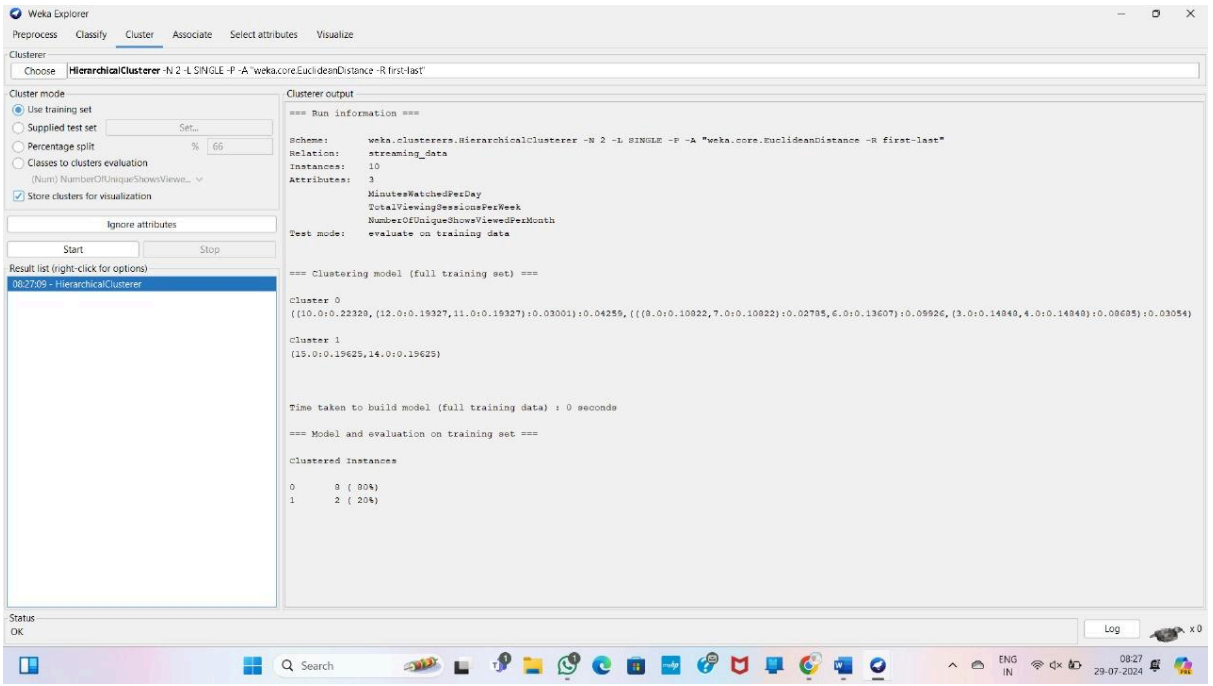


Clustering Analysis:



Q20:

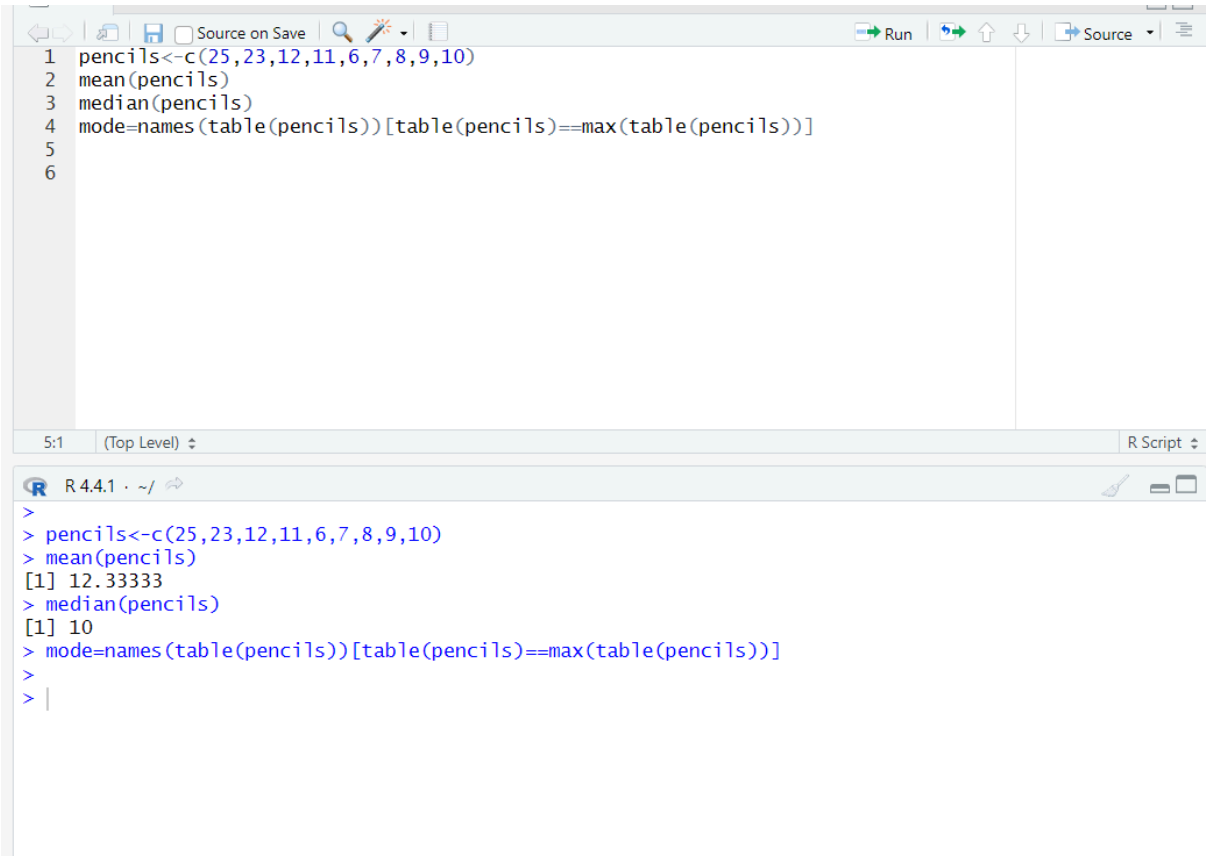

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: streaming_data
Instances: 10
Attributes: 3
Sum of weights: 10
Attributes
All
None
Invert
Pattern
No.
Name
1 MinutesWatchedPerDay
2 TotalViewingSessionsPerWeek
3 NumberOfUniqueShowsViewedPerMonth
Remove
Selected attribute
Name: TotalViewingSessionsPerWeek
Missing: 0 (0%)
Distinct: 10
Type: Numeric
Unique: 10 (100%)
Statistic
Value
Minimum 5
Maximum 25
Mean 14.5
StdDev 6.621
Class: NumberOfUniqueShowsViewedPerMonth (Num)
Visualize All
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) NumberOfUniqueShowsViewe...
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
08:27:09 - HierarchicalClusterer
Clusterer output
=== Run information ===
Scheme: weka.clusterers.HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"
Relation: streaming_data
Instances: 10
Attributes: 3
MinutesWatchedPerDay
TotalViewingSessionsPerWeek
NumberOfUniqueShowsViewedPerMonth
Test mode: evaluate on training data
=== Clustering model (full training set) ===
Cluster 0
((10.0:0.22329,(12.0:0.19327,11.0:0.19327):0.03001):0.04255,(((8.0:0.10022,7.0:0.10022):0.02705,6.0:0.13607):0.09926,(3.0:0.14848,4.0:0.14848):0.08685):0.03054)
Cluster 1
(15.0:0.19625,14.0:0.19625)
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 8 (80%)
1 2 (20%)
Status
OK
Log
x 0

Q21:
INPUT:

```
pencils<-c(25,23,12,11,6,7,8,9,10)
mean(pencils)
median(pencils)
mode=names(table(pencils))[table(pencils)==max(table(pencils))]
```

output:

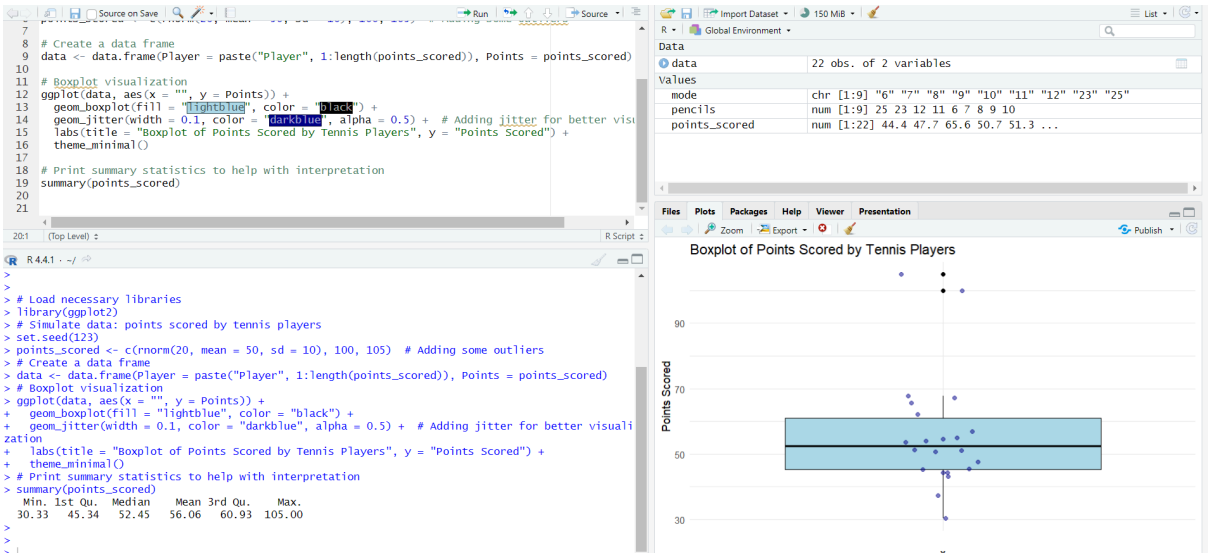


Q22:

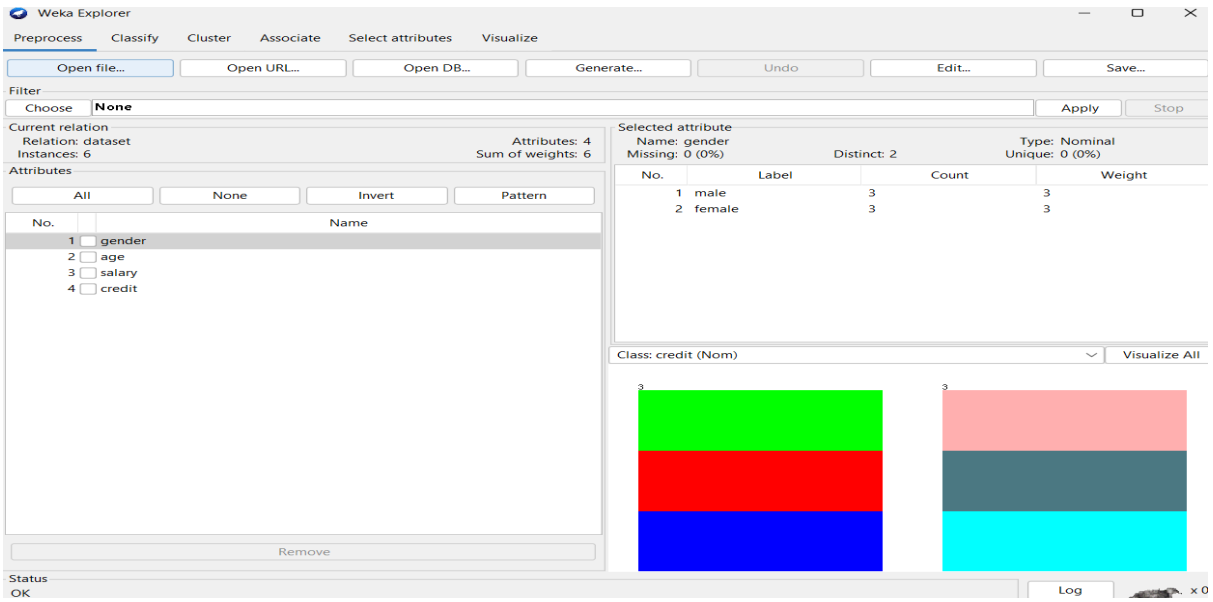
INPUT:

```
library(ggplot2)

set.seed(123)

points_scored <- c(rnorm(20, mean = 50, sd = 10), 100, 105)  # Adding some outliers

data <- data.frame(Player = paste("Player", 1:length(points_scored)), Points =
points_scored)
ggplot(data, aes(x = "", y = Points)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  geom_jitter(width = 0.1, color = "darkblue", alpha = 0.5) +  # Adding jitter for better
visualization
  labs(title = "Boxplot of Points Scored by Tennis Players", y = "Points Scored") +
  theme_minimal()
summary(points_scored)
```

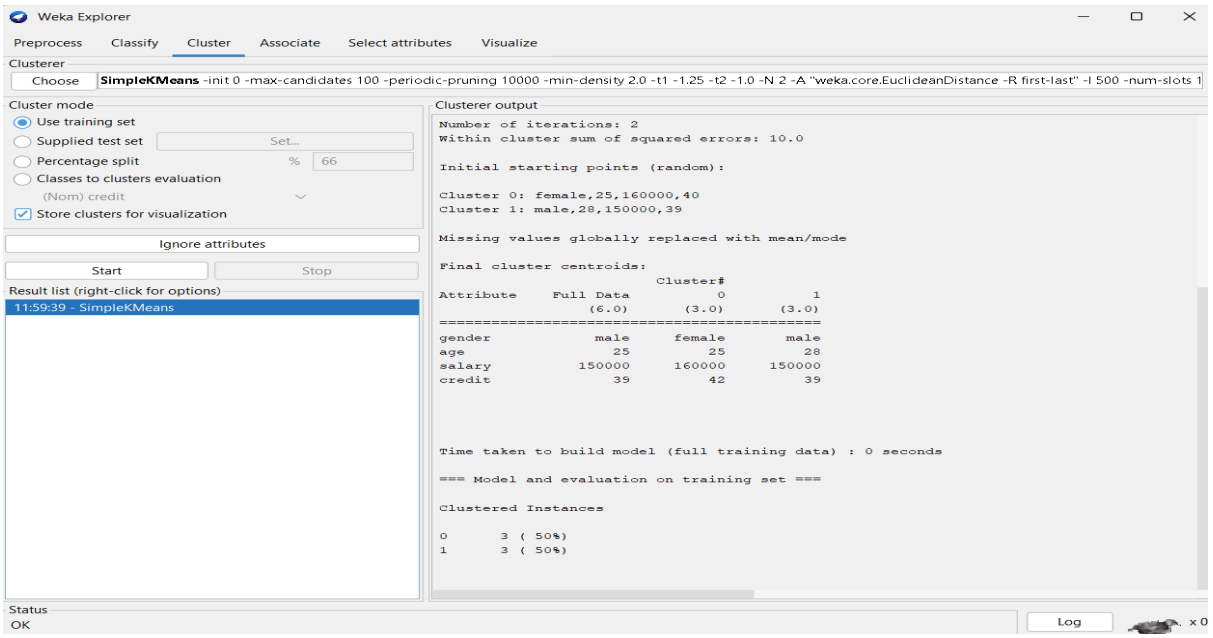
Output:


Boxplot of Points Scored by Tennis Players
Points Scored
90
70
50
30

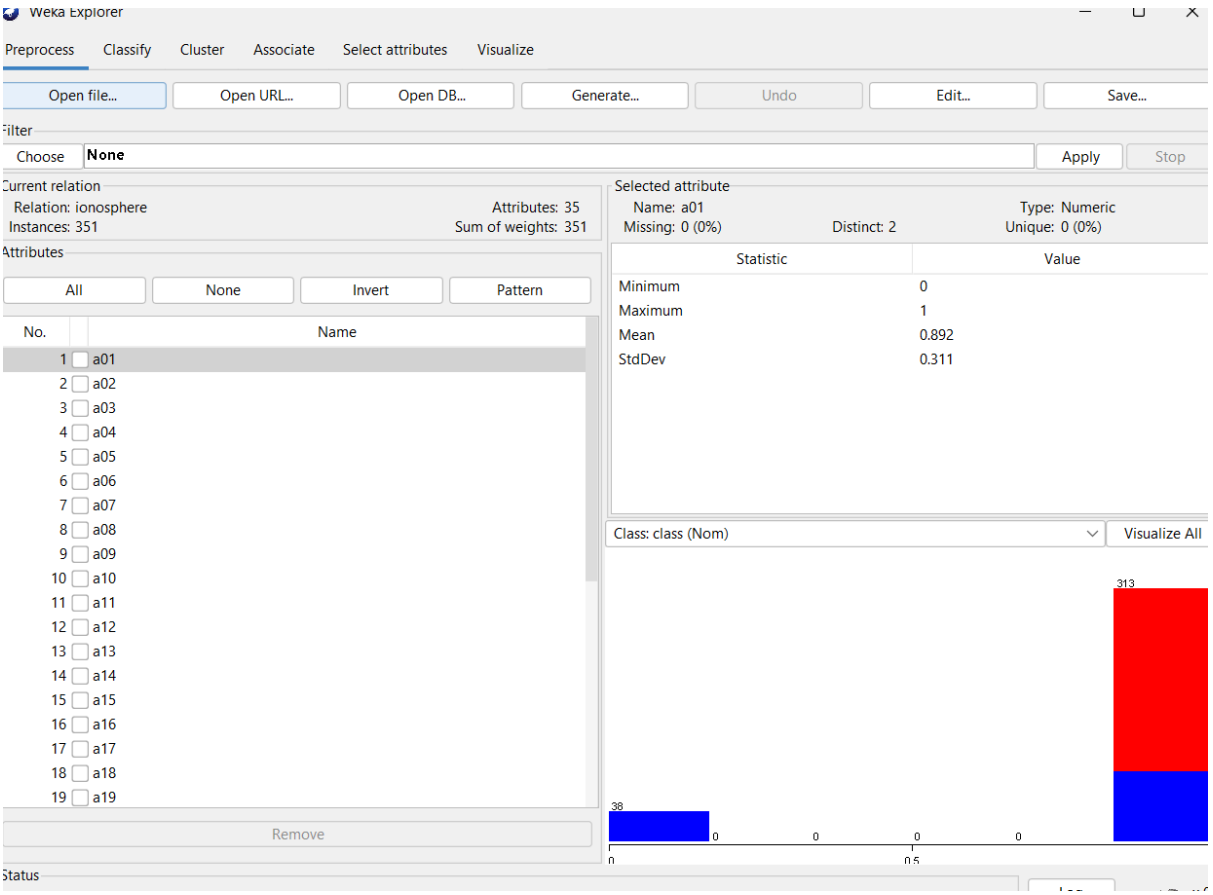
Q23:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: dataset
Instances: 6
Attributes: 4
Sum of weights: 6
Attributes
All
None
Invert
Pattern
No.
Name
1 gender
2 age
3 salary
4 credit
Remove
Selected attribute
Name: gender
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 male 3 3
2 female 3 3
Class: credit (Nom)
Visualize All
Status
OK
Log

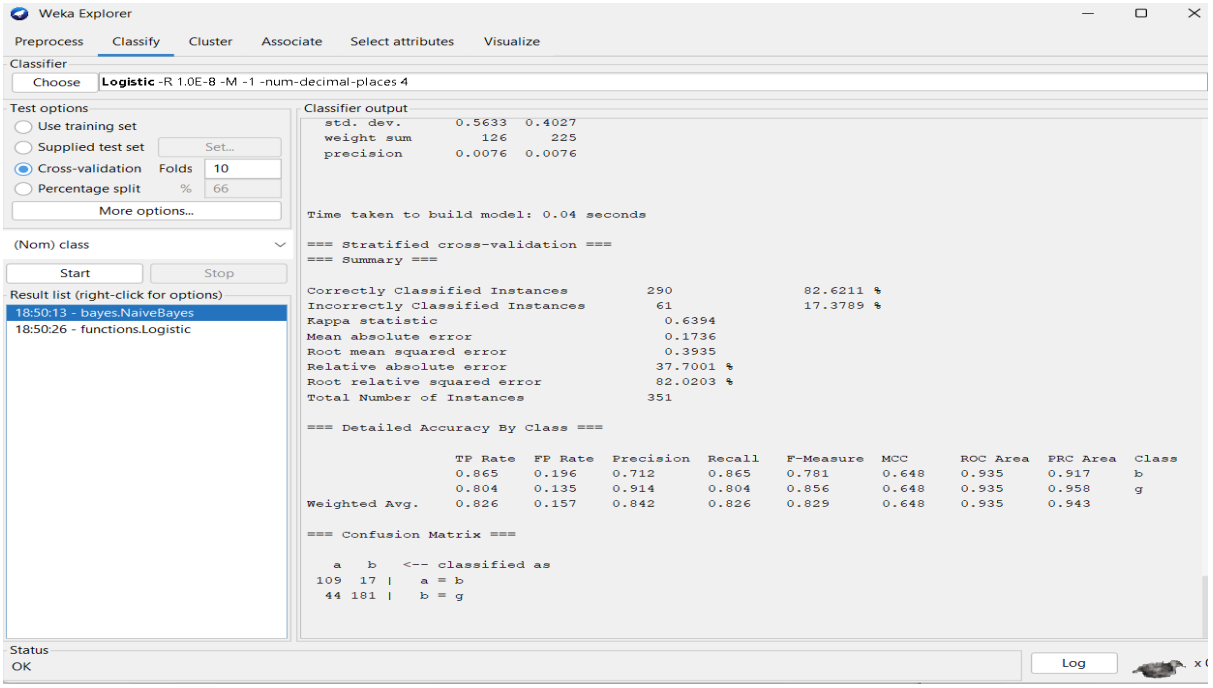
K Means:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) credit
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
11:59:39 - SimpleKMeans
Clusterer output
Number of iterations: 2
Within cluster sum of squared errors: 10.0
Initial starting points (random):
Cluster 0: female,25,160000,40
Cluster 1: male,28,150000,39
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(6.0) (3.0) (3.0)
gender male female male
age 25 25 28
salary 150000 160000 150000
credit 39 42 39
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 3 (50%)
1 3 (50%)
Status
OK
Log

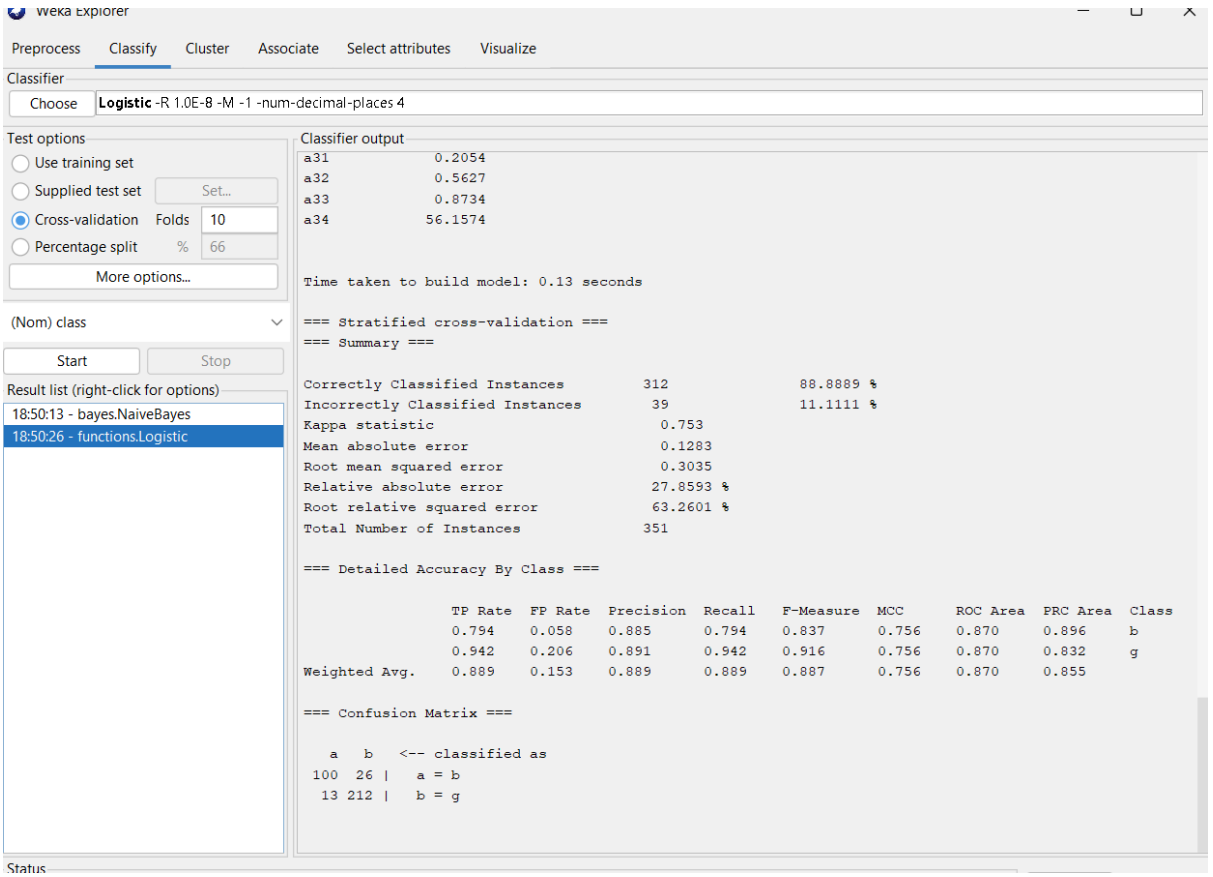
Q24:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: ionosphere
Instances: 351
Attributes: 35
Sum of weights: 351
Attributes
All
None
Invert
Pattern
No.
Name
1 a01
2 a02
3 a03
4 a04
5 a05
6 a06
7 a07
8 a08
9 a09
10 a10
11 a11
12 a12
13 a13
14 a14
15 a15
16 a16
17 a17
18 a18
19 a19
Remove
Selected attribute
Name: a01
Missing: 0 (0%)
Distinct: 2
Type: Numeric
Unique: 0 (0%)
Statistic
Value
Minimum 0
Maximum 1
Mean 0.892
StdDev 0.311
Class: class (Nom)
Visualize All
313
38
0
0
0
0
0
0.5
Status

Naïve Bayes classification:



SVM:



Q25:

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

7.R*

Source on Save | Run | Source

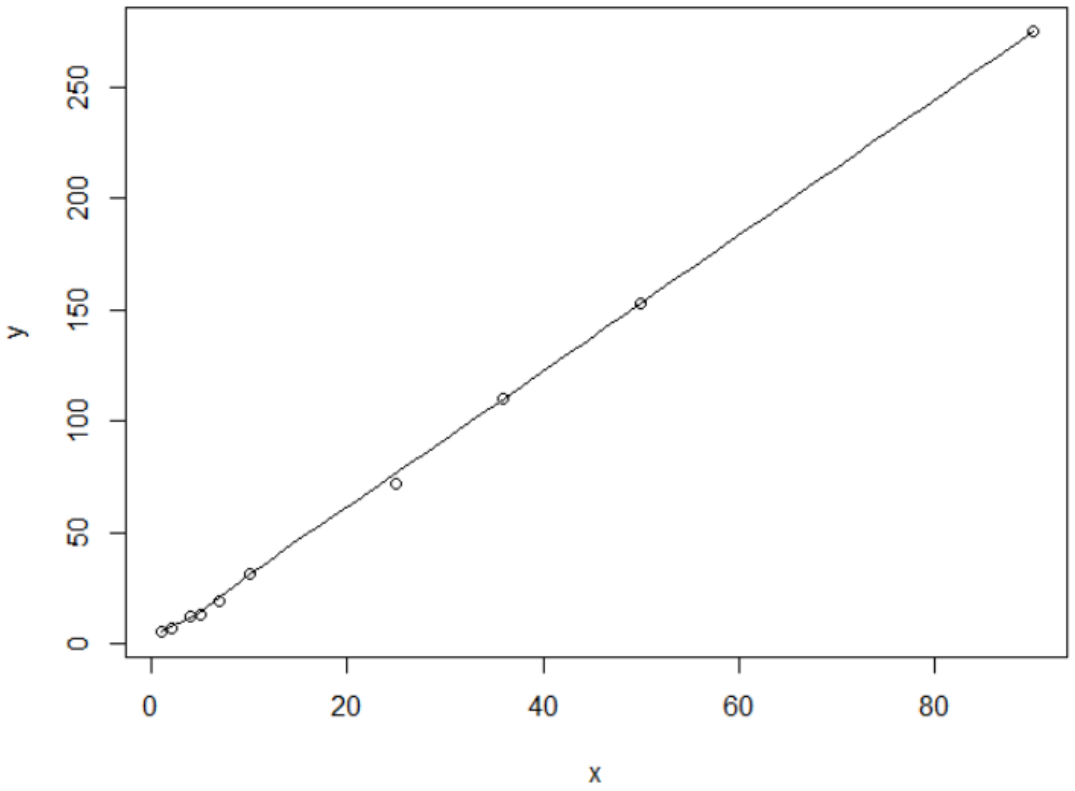
```
# Data
persons <- c("Gopu", "Babu", "Baby", "Gopal", "Krishna", "Jai", "Dev", "Malini", "Hema", "Anu"
vegetarian_status <- c("yes", "yes", "yes", "no", "yes", "no", "no", "yes", "yes", "yes")

# Create a data frame
data <- data.frame(persons, vegetarian_status)

# Count the number of vegetarians and non-vegetarians
veg_count <- sum(vegetarian_status == "yes")
```

25:1 (Top Level) R Script

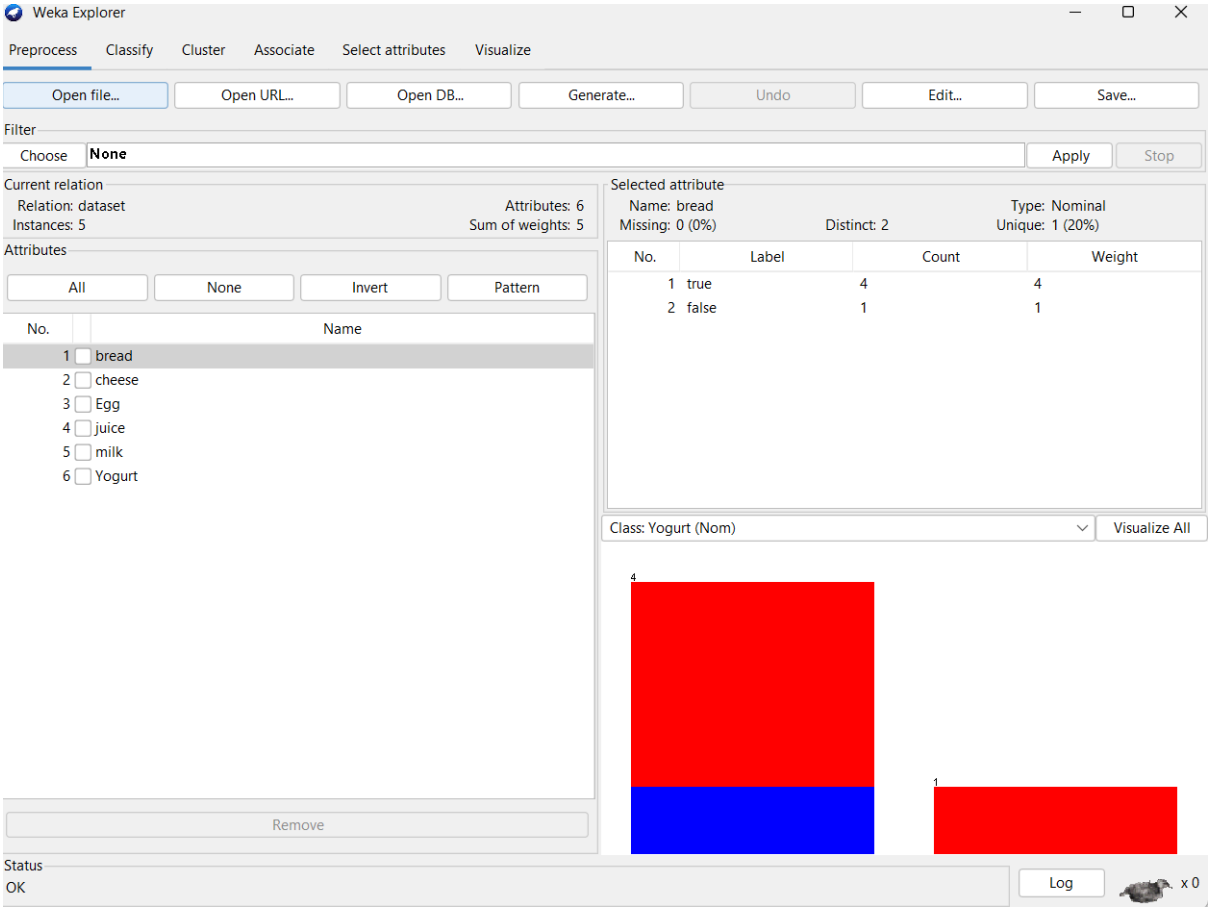
R 4.4.1 · ~/

```
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.

>
> # Data
> persons <- c("Gopu", "Babu", "Baby", "Gopal", "Krishna", "Jai", "Dev", "Malini", "Hema", "Anu")
> vegetarian_status <- c("yes", "yes", "yes", "no", "yes", "no", "no", "yes", "yes", "yes")
> # Create a data frame
> data <- data.frame(persons, vegetarian_status)
> # Count the number of vegetarians and non-vegetarians
> veg_count <- sum(vegetarian_status == "yes")
> non_veg_count <- sum(vegetarian_status == "no")
> # Determine which type has the greater count
> if (veg_count > non_veg_count) {
+   greater_count_type <- "Vegetarians"
+ } else if (non_veg_count > veg_count) {
+   greater_count_type <- "Non-Vegetarians"
+ } else {
+   greater_count_type <- "Both are equal"
+ }
> # Results
> cat("Number of Vegetarians:", veg_count, "\n")
Number of Vegetarians: 7
> cat("Number of Non-Vegetarians:", non_veg_count, "\n")
Number of Non-Vegetarians: 3
> cat("Type with Greater Count:", greater_count_type, "\n")
Type with Greater Count: Vegetarians
>
>
```

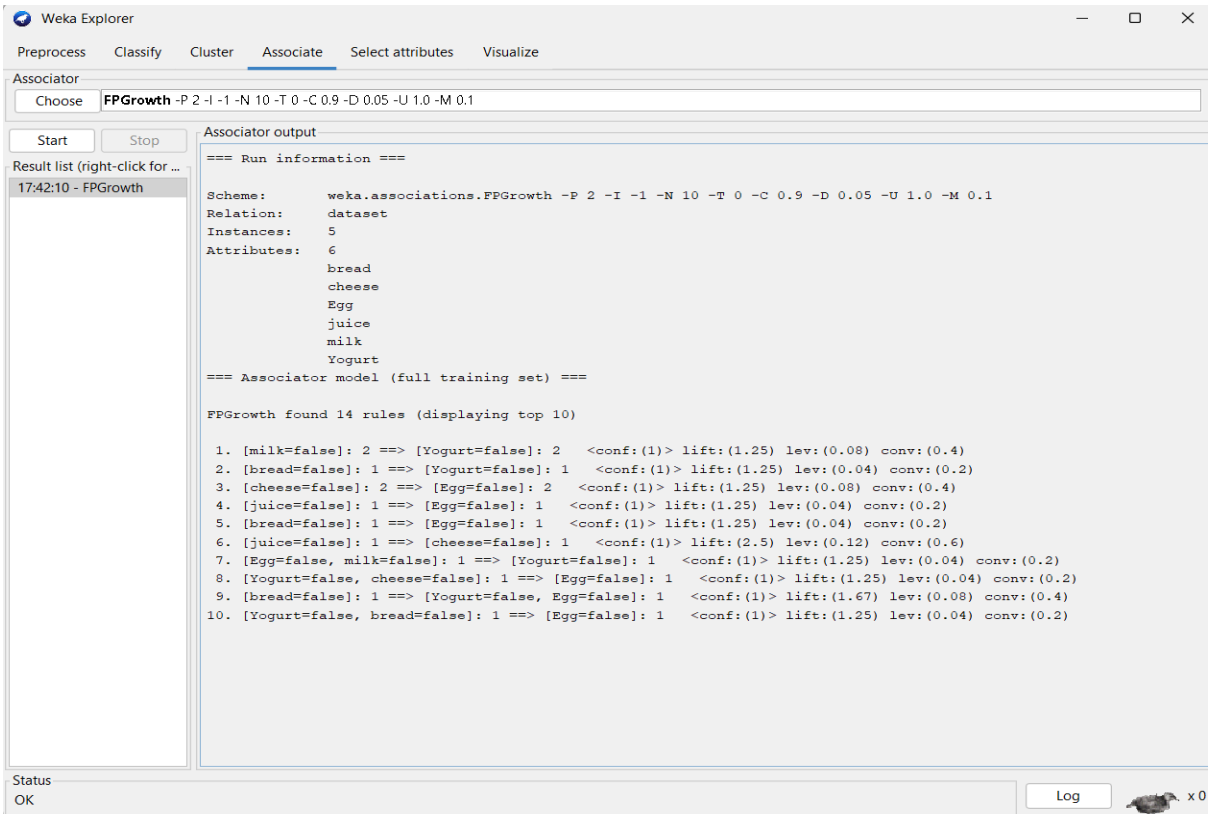
Q26:



Q27:



FP GROWTH:



Q28: