General Sir John Kotelawala Defence University

Faculty of Management, Social Sciences and Humanities

Department of Languages

BSc in Applied Data Science Communication

D L S NADAVI - D/ADC/23/0028

Y M S S B SENAVIRATHNA - D/ADC/23/0030

A G A U S GUNASEKARA - D/ADC/23/0034

M L B T S PERERA - D/ADC/23/0047

Fundamentals of Data Mining / LB 2114

Year 2: Semester 1

Assignment number 2

07.05.2024

Table of contents Page no

<u>Task 01</u>

Association Rule Mining with Student Dataset

Table of contents

Page no

<u>Task 02</u>

Regression Analysis using Diabetes Dataset

Task 01 – Association Rule Mining with Student Dataset

1) Introduction

Association rule mining is a technique used to identify hidden links between variables in huge datasets. The goal of association rule mining is to find patterns or correlations between distinct items, which can then be used to predict whether specific goods would be purchased or used together. Association rule mining has a wide range of applications, including market basket research, consumer segmentation, and fraud detection.

This report describes about students' academic performance and behavior with respect to familial and educational background, lifestyle choices and socio-economic factors. The aim of creating this report is to explore the association rules and patterns that exist and their potential impact on students' academic performance and well-being. This paper outlines all of the procedures involved in creating association rules from a data set using R in a straightforward and logical manner.

2) Data Set

The data set was taken from:
https://github.com/Emmanuel96/apriori_association_rule_mining/tree/master/Dataset

This dataset includes information about various attributes of students, with a focus on factors that may influence their academic performance and behavior. These attributes covers a broad spectrum ranging from demographic details to familial and educational background, as well as lifestyle choices and socio-economic indicators. Each entry in the dataset corresponds to a student enrolled in a particular school, providing a rich repository of data for analysis.

3) Explanation and Preparation of the Data Set

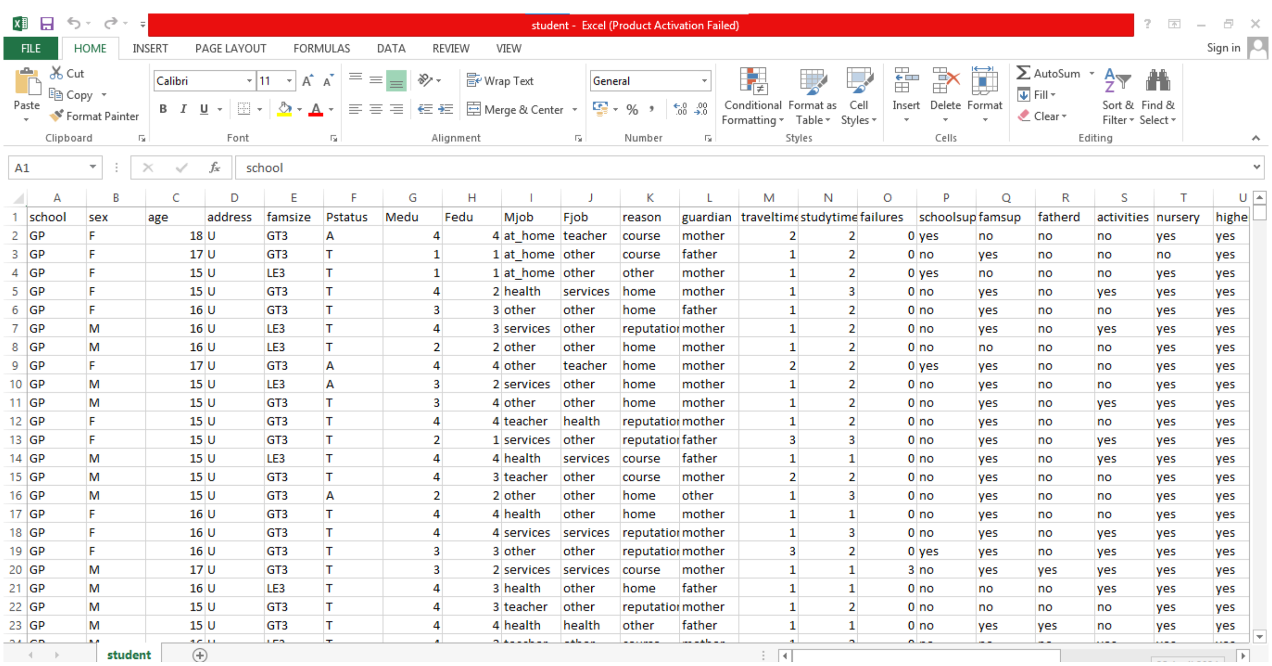
a. Explanation of the Data Set

Student data set has been used for the association rule mining task. There are 33 columns and 1046 rows in the data set.

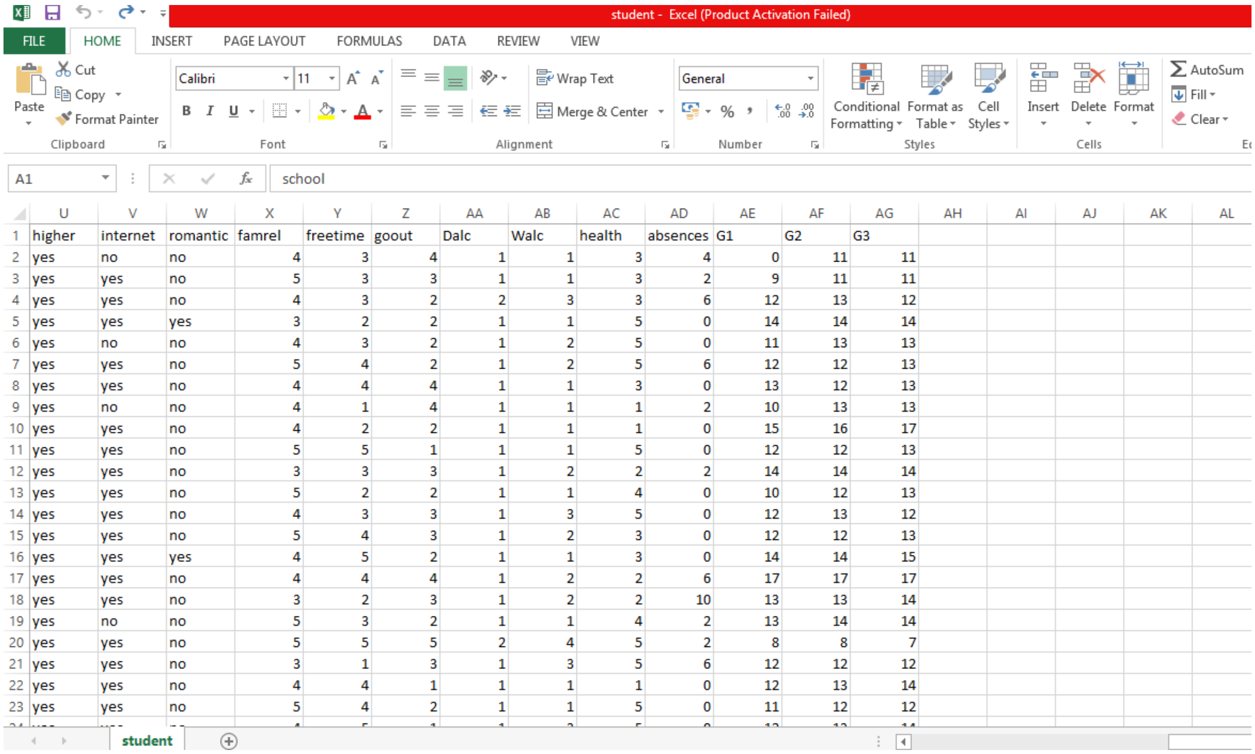
Attributes of the data set are,

1. School - The school the student attends
2. Sex - Gender of the student (Male or Female)
3. Age - Age of the student
4. Address - Type of address of the student (urban or rural)
5. Famsize - Family size (small or large)
6. Pstatus - Parent's cohabitation status ('T' - living together, 'A' - living apart)
7. Medu - Mother's education level (1 - none, 2 - primary education (4th grade), 3 - 5th to 9th grade, 4 - secondary or higher education)
8. Fedu - Father's education level (same scale as Medu)
9. Mjob - Mother's job
10. Fjob - Father's job

11. Reason - Reason for choosing the current school
12. Guardian - Student's guardian
13. Traveltime - Home to school travel time (1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. Studytime - Weekly study time (1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. Failures - Number of past class failures
16. Schoolsup - Whether the student receives educational support from the school (yes or no)
17. Famsup - Whether the student receives educational support from the family (yes or no)
18. Fatherd - Father's educational support level (1 - low, 2 - medium, or 3 - high)
19. Activities - Extra-curricular activities participation (yes or no)
20. Nursery - Whether the student attended nursery school (yes or no)
21. Higher - Desire to pursue higher education (yes or no)
22. Internet - Internet access at home (yes or no)
23. Romantic - In a romantic relationship (yes or no)
24. Famrel - Quality of family relationships (from 1 - very bad to 5 - excellent)
25. Freetime - Free time after school (from 1 - very low to 5 - very high)
26. Goout - Going out with friends frequency (from 1 - very low to 5 - very high)
27. Dalc - Workday alcohol consumption (from 1 - very low to 5 - very high)
28. Walc - Weekend alcohol consumption (from 1 - very low to 5 - very high)
29. Health - Current health status (from 1 - very bad to 5 - very good)
30. Absences - Number of school absences
31. G1 - First period grade (from 0 to 20)
32. G2 - Second period grade (from 0 to 20)
33. G3 - Final grade (from 0 to 20)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	fatherd	activities	nursery	highe
2	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no	no	no	yes	yes
3	GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	no	yes	no	no	no	yes
4	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	0	yes	no	no	no	yes	yes
5	GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	no	yes	no	yes	yes	yes
6	GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	no	yes	no	no	yes	yes
7	GP	M	16	U	LE3	T	4	3	services	other	reputatio	mother	1	2	0	no	yes	no	yes	yes	yes
8	GP	M	16	U	LE3	T	2	2	other	other	home	mother	1	2	0	no	no	no	no	yes	yes
9	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2	2	0	yes	yes	no	no	yes	yes
10	GP	M	15	U	LE3	A	3	2	services	other	home	mother	1	2	0	no	yes	no	no	yes	yes
11	GP	M	15	U	GT3	T	3	4	other	other	home	mother	1	2	0	no	yes	no	yes	yes	yes
12	GP	F	15	U	GT3	T	4	4	teacher	health	reputatio	mother	1	2	0	no	yes	no	no	yes	yes
13	GP	F	15	U	GT3	T	2	1	services	other	reputatio	father	3	3	0	no	yes	no	yes	yes	yes
14	GP	M	15	U	LE3	T	4	4	health	services	course	father	1	1	0	no	yes	no	yes	yes	yes
15	GP	M	15	U	GT3	T	4	3	teacher	other	course	mother	2	2	0	no	yes	no	no	yes	yes
16	GP	M	15	U	GT3	A	2	2	other	other	home	other	1	3	0	no	yes	no	no	yes	yes
17	GP	F	16	U	GT3	T	4	4	health	other	home	mother	1	1	0	no	yes	no	no	yes	yes
18	GP	F	16	U	GT3	T	4	4	services	services	reputatio	mother	1	3	0	no	yes	no	yes	yes	yes
19	GP	F	16	U	GT3	T	3	3	other	other	reputatio	mother	3	2	0	yes	yes	no	yes	yes	yes
20	GP	M	17	U	GT3	T	3	2	services	services	course	mother	1	1	3	no	yes	yes	yes	yes	yes
21	GP	M	16	U	LE3	T	4	3	health	other	home	father	1	1	0	no	no	no	yes	yes	yes
22	GP	M	15	U	GT3	T	4	3	teacher	other	reputatio	mother	1	2	0	no	no	no	no	yes	yes
23	GP	M	15	U	GT3	T	4	4	health	health	other	father	1	1	0	no	yes	yes	no	yes	yes



	higher	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
2	yes	no	no	4	3	4	1	1	3	4	0	11	11
3	yes	yes	no	5	3	3	1	1	3	2	9	11	11
4	yes	yes	no	4	3	2	2	3	3	6	12	13	12
5	yes	yes	yes	3	2	2	1	1	5	0	14	14	14
6	yes	no	no	4	3	2	1	2	5	0	11	13	13
7	yes	yes	no	5	4	2	1	2	5	6	12	12	13
8	yes	yes	no	4	4	4	1	1	3	0	13	12	13
9	yes	no	no	4	1	4	1	1	1	2	10	13	13
10	yes	yes	no	4	2	2	1	1	1	0	15	16	17
11	yes	yes	no	5	5	1	1	1	5	0	12	12	13
12	yes	yes	no	3	3	3	1	2	2	2	14	14	14
13	yes	yes	no	5	2	2	1	1	4	0	10	12	13
14	yes	yes	no	4	3	3	1	3	5	0	12	13	12
15	yes	yes	no	5	4	3	1	2	3	0	12	12	13
16	yes	yes	yes	4	5	2	1	1	3	0	14	14	15
17	yes	yes	no	4	4	4	1	2	2	6	17	17	17
18	yes	yes	no	3	2	3	1	2	2	10	13	13	14
19	yes	no	no	5	3	2	1	1	4	2	13	14	14
20	yes	yes	no	5	5	5	2	4	5	2	8	8	7
21	yes	yes	no	3	1	3	1	3	5	6	12	12	12
22	yes	yes	no	4	4	1	1	1	1	0	12	13	14
23	yes	yes	no	5	4	2	1	1	5	0	11	12	12

b. Preparation of the Data Set

As the dataset is completely suitable for do association rule mining and has no NULL values in the dataset, we didn't had much work to do to prepare the dataset. Therefore, first we read and understood the dataset and applied the association rule mining into the dataset using R software.

4) Association Rule Mining

Association rule mining is a type of unsupervised machine learning technique that discovers connections between two or more items in large datasets. It was proposed by Agrawal et al in 1993. It's a popular system in data mining which has a wide range of operations in various fields, such as request market basket analysis, customer segmentation, and fraud discovery. The two most important measures used in association rule mining are support and confidence.

- Support: This measures how frequently the particulars in the rule appear together in the dataset. A high support value indicates that the rule is constantly being.
- Confidence: This measures how likely it's that the consequent item will do if the precedent item occurs. Strong rules are indicated by a high confidence value.

A third metric called lift, can be used to compare confidence with anticipated confidence, or how numerous times an if- also statement is anticipated to be set up true.

5) Implementation in R

Packages used

1) **arules:** A complete R package for mining association rules and frequent item sets from transaction data is called `arules`. The association rules that describe the relationships between items in transactional datasets can be generated and evaluated by this package. Recommendation systems, market basket analysis, and other applications involving transactional data analysis frequently use this package.
2) **arulesviz:** Specifically created for the purpose of visualizing association rules and item sets, the `arulesviz` package is an extension of the `arules` package. To assist users in exploring and interpreting the outcomes of association rule mining, it provides a range of visualization techniques. Scatter plots, matrix plots, and graph-based representations of item sets and rules are some examples of these visualizations.

Explanation of the experimental procedure and Visualization of the results

Step 01

Import the dataset.

```
> #import data set
> data=read.csv("student.csv",header=T, colClasses="factor")
> data
   school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob
1      GP   F  18       U     GT3       A    4    4  at_home  teacher
2      GP   F  17       U     GT3       T    1    1  at_home    other
3      GP   F  15       U     LE3       T    1    1  at_home    other
4      GP   F  15       U     GT3       T    4    2   health services
5      GP   F  16       U     GT3       T    3    3    other    other
6      GP   M  16       U     LE3       T    4    3 services    other
7      GP   M  16       U     LE3       T    2    2    other    other
8      GP   F  17       U     GT3       A    4    4    other  teacher
9      GP   M  15       U     LE3       A    3    2 services    other
10     GP   M  15       U     GT3       T    3    4    other    other
11     GP   F  15       U     GT3       T    4    4  teacher   health
12     GP   F  15       U     GT3       T    2    1 services    other
13     GP   M  15       U     LE3       T    4    4   health services
```

Step 02

Use the 'name ()' function to get the column names of the dataset.

```
> names(data)
 [1] "school"     "sex"        "age"        "address"    "famsize"
 [6] "Pstatus"    "Medu"       "Fedu"       "Mjob"       "Fjob"
[11] "reason"     "guardian"   "traveltime" "studytime"  "failures"
[16] "schoolsup"  "famsup"     "fatherd"    "activities" "nursery"
[21] "higher"     "internet"   "romantic"   "famrel"     "freetime"
[26] "goout"      "Dalc"       "Walc"       "health"     "absences"
[31] "G1"         "G2"         "G3"
> 
```

Step 03

Use 'head ()' and 'tail ()' functions to get first and last 6 rows in the dataset.

```
> head(data)
  school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob
1     GP   F  18       U     GT3       A    4    4  at_home  teacher
2     GP   F  17       U     GT3       T    1    1  at_home    other
3     GP   F  15       U     LE3       T    1    1  at_home    other
4     GP   F  15       U     GT3       T    4    2   health services
5     GP   F  16       U     GT3       T    3    3    other    other
6     GP   M  16       U     LE3       T    4    3 services    other
      reason guardian traveltime studytime failures schoolsup famsup
1     course   mother          2         2        0       yes     no
2     course   father          1         2        0        no    yes
3      other   mother          1         2        0       yes     no
4       home   mother          1         3        0        no    yes
5       home   father          1         2        0        no    yes
6 reputation   mother          1         2        0        no    yes
  fatherd activities nursery higher internet romantic famrel freetime
1      no         no     yes    yes       no       no      4        3
2      no         no      no    yes      yes       no      5        3
3      no         no     yes    yes      yes       no      4        3
4      no        yes     yes    yes      yes      yes      3        2
5      no         no     yes    yes       no       no      4        3
6      no        yes     yes    yes      yes       no      5        4
  goout Dalc Walc health absences G1 G2 G3
1     4    1    1      3        4  0 11 11
2     3    1    1      3        2  9 11 11
3     2    2    3      3        6 12 13 12
4     2    1    1      5        0 14 14 14
5     2    1    2      5        0 11 13 13
6     2    1    2      5        6 12 12 13
> |
```

```
> tail(data)
     school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob reason guardian
1040     MS   F  18       U     GT3       T    1    1    other    other course   mother
1041     MS   M  20       U     LE3       A    2    2 services services course    other
1042     MS   M  17       U     LE3       T    3    1 services services course   mother
1043     MS   M  21       R     GT3       T    1    1    other    other course    other
1044     MS   M  18       R     LE3       T    3    2 services    other course   mother
1045     MS   M  19       U     LE3       T    1    1    other  at_home course   father
     traveltime studytime failures schoolsup famsup fatherd activities nursery higher
1040          2         2        1        no     no      no        yes     yes    yes
1041          1         2        2        no    yes     yes         no     yes    yes
1042          2         1        0        no     no      no         no      no    yes
1043          1         1        3        no     no      no         no      no    yes
1044          3         1        0        no     no      no         no      no    yes
1045          1         1        0        no     no      no         no     yes    yes
     internet romantic famrel freetime goout Dalc Walc health absences G1 G2 G3
1040       no       no      1        1     1    1    1      5        0  6  5  0
1041       no       no      5        5     4    4    5      4       11  9  9  9
1042      yes       no      2        4     5    3    4      2        3 14 16 16
1043       no       no      5        5     3    3    3      3        3 10  8  7
1044      yes       no      4        4     1    3    4      5        0 11 12 10
1045      yes       no      3        2     3    3    3      5        5  8  9  9
> |
```

Step 04

Use the ‘summary ()’ function to get the summary of the dataset.

```
> summary(data)
    school       sex           age          address        famsize         Pstatus      Medu
 GP     :772   F   :591   16      :281   address:  1   famsize:  1   A       :121    :  1
 MS     :272   M   :453   17      :277   R      :285   GT3    :738   Pstatus:  1   0:  9
 school:  1    sex:  1    18      :222   U      :759   LE3    :306   T       :923   1:202
                          15      :194                                              2:289
                          19      : 56                                              3:238
                          20      :  9                                              4:306
                          (Other):  6
 Fedu          Mjob          Fjob            reason         guardian      traveltime
  :  1   at_home :194   at_home : 62   course    :430   father   :243    :  1
 0:  9   health  : 82   Fjob    :  1   home      :258   guardian:  1   1:623
 1:256   Mjob    :  1   health  : 41   other     :108   mother   :728   2:320
 2:324   other   :399   other   :584   reason    :  1   other    : 73   3: 77
 3:231   services:239   services:292   reputation:248                  4: 24
 4:224   teacher :130   teacher : 65

 studytime failures      schoolsup      famsup     fatherd        activities      nursery
  :  1      :  1     no       :925   famsup:  1   no  :824   activities:  1   no      :209
 1:317     0:861     schoolsup:  1   no    :404   paid:  1   no        :528   nursery:  1
 2:503     1:120     yes      :119   yes   :640   yes :220   yes       :516   yes     :835
 3:162     2: 33
 4: 62     3: 30
```

Step 05

Use the ‘str ()’ function to get the structure of the dataset.

```
> str(data)
'data.frame':   1045 obs. of  33 variables:
 $ school    : Factor w/ 3 levels "GP","MS","school": 1 1 1 1 1 1 1 1 1 1 ...
 $ sex       : Factor w/ 3 levels "F","M","sex": 1 1 1 1 1 2 2 1 2 2 ...
 $ age       : Factor w/ 9 levels "","15","16","17",..: 5 4 2 2 3 3 3 4 2 2 ...
 $ address   : Factor w/ 3 levels "address","R",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ famsize   : Factor w/ 3 levels "famsize","GT3",..: 2 2 3 2 2 3 3 2 3 2 ...
 $ Pstatus   : Factor w/ 3 levels "A","Pstatus",..: 1 3 3 3 3 3 3 1 1 3 ...
 $ Medu      : Factor w/ 6 levels "","0","1","2",..: 6 3 3 6 5 6 4 6 5 5 ...
 $ Fedu      : Factor w/ 6 levels "","0","1","2",..: 6 3 3 4 5 5 4 6 4 6 ...
 $ Mjob      : Factor w/ 6 levels "at_home","health",..: 1 1 1 2 4 5 4 4 5 4 ...
 $ Fjob      : Factor w/ 6 levels "at_home","Fjob",..: 6 4 4 5 4 4 4 6 4 4 ...
 $ reason    : Factor w/ 5 levels "course","home",..: 1 1 3 2 2 5 2 2 2 2 ...
 $ guardian  : Factor w/ 4 levels "father","guardian",..: 3 1 3 3 1 3 3 3 3 3 ...
 $ traveltime: Factor w/ 5 levels "","1","2","3",..: 3 2 2 2 2 2 2 3 2 2 ...
 $ studytime : Factor w/ 5 levels "","1","2","3",..: 3 3 3 4 3 3 3 3 3 3 ...
 $ failures  : Factor w/ 5 levels "","0","1","2",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ schoolsup : Factor w/ 3 levels "no","schoolsup",..: 3 1 3 1 1 1 1 3 1 1 ...
 $ famsup    : Factor w/ 3 levels "famsup","no",..: 2 3 2 3 3 3 2 3 3 3 ...
 $ fatherd   : Factor w/ 3 levels "no","paid","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ activities: Factor w/ 3 levels "activities","no",..: 2 2 2 3 2 3 2 2 2 3 ...
 $ nursery   : Factor w/ 3 levels "no","nursery",..: 3 1 3 3 3 3 3 3 3 3 ...
 $ higher    : Factor w/ 3 levels "higher","no",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ internet  : Factor w/ 3 levels "internet","no",..: 2 3 3 3 2 3 3 2 3 3 ...
 $ romantic  : Factor w/ 3 levels "no","romantic",..: 1 1 1 3 1 1 1 1 1 1 ...
 $ famrel    : Factor w/ 6 levels "","1","2","3",..: 5 6 5 4 5 6 5 5 5 6 ...
 $ freetime  : Factor w/ 6 levels "","1","2","3",..: 4 4 4 3 4 5 5 2 3 6 ...
```

Step 06

Use the 'dim ()' function to get the dimension of the data set which includes the number of rows and columns in the data set.

```
> dim(data)
[1] 1045   33
> 
```

Step 07

Get columns for association rule mining.

```
> #get columns for Association Rule Mining
> data_new=data[,16:23]
> View(data_new)
> 
```

ARM.R* × | data_new ×

Filter

	schoolsup	famsup	fatherd	activities	nursery	higher	internet	romantic
1	yes	no	no	no	yes	yes	no	no
2	no	yes	no	no	no	yes	yes	no
3	yes	no	no	no	yes	yes	yes	no
4	no	yes	no	yes	yes	yes	yes	yes
5	no	yes	no	no	yes	yes	no	no
6	no	yes	no	yes	yes	yes	yes	no
7	no	no	no	no	yes	yes	yes	no
8	yes	yes	no	no	yes	yes	no	no
9	no	yes	no	no	yes	yes	yes	no
10	no	yes	no	yes	yes	yes	yes	no

Showing 1 to 11 of 1,045 entries, 8 total columns

Step 08

Use colSums () function to compute the sum of columns.

```
> #  Only YES columns
> yes=colSums(data_new=="yes")
> yes
 schoolsup     famsup    fatherd activities    nursery     higher   internet   romantic 
       119        640        220        516        835        955        827        371 
> 
```

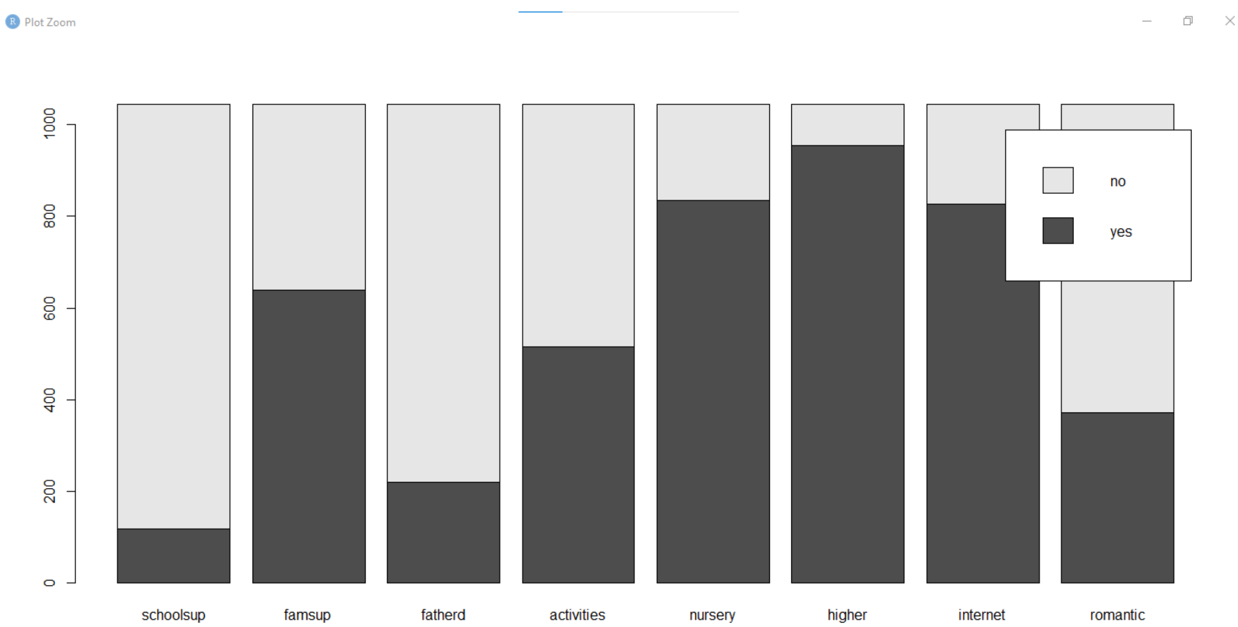
```
> # only NO columns
> no=colSums(data_new=="no")
> no
 schoolsup     famsup    fatherd activities    nursery     higher   internet   romantic 
       925        404        824        528        209         89        217        673 
> 
```

```
> #Get both YES & NO columns
> sub=rbind(yes,no)
> sub
    schoolsup famsup fatherd activities nursery higher internet romantic
yes       119    640     220        516     835    955      827      371
no        925    404     824        528     209     89      217      673
> 
```

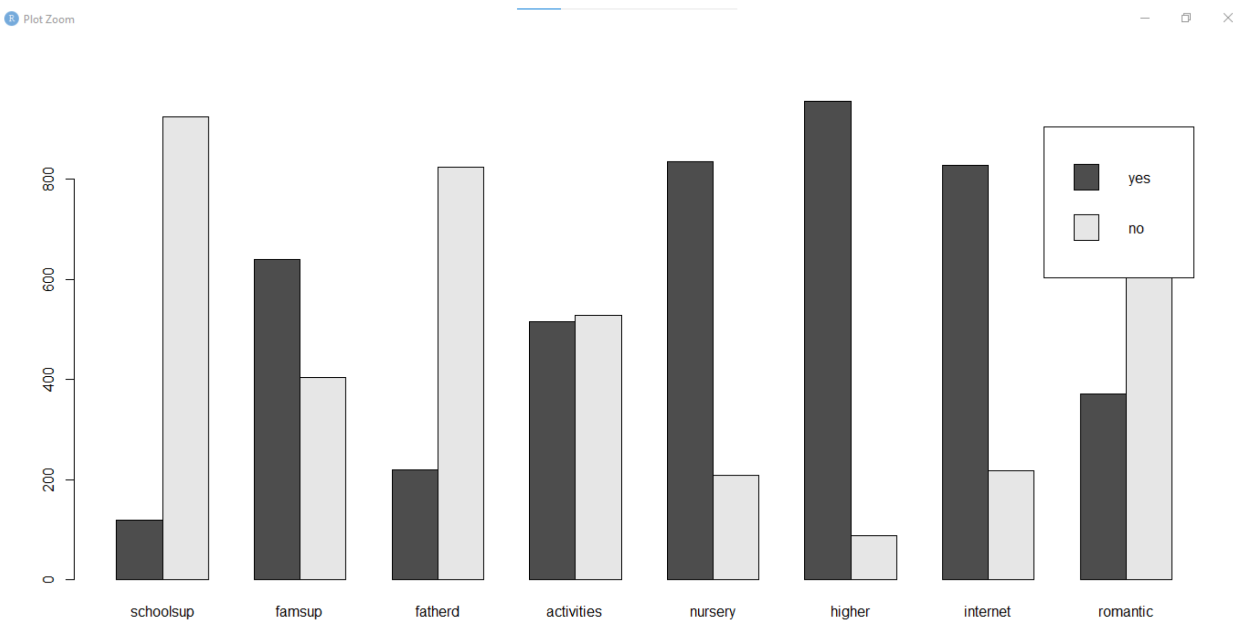
Step 09

Plot and explore the “student” dataset with barplot () function.

```
> barplot(sub,legend=rownames(sub))
> 
```



```
> barplot(sub,beside = T,legend=rownames(sub))
> 
```



Step 10

Install and activate “arules” package.

```
#Install "arules" package
install.packages("arules")
library(arules)
```

Step 11

Create Association rules.

According to the plot "higher" has the highest count of “Yes”. As we want to see rules where desire to pursue higher education is equal to yes, we used the following code to get those rules for higher.

Rule 01 – Get the rules under the confidence of 0.8

```
> #Get the rules under the confidence of 0.8
> rules_1=apriori(data_new,parameter = list(conf=0.8),
+                 appearance = list(rhs=c("higher=yes"),default="lhs"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime
        0.8    0.1    1 none FALSE            TRUE       5
 support minlen maxlen target   ext
     0.1      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 104

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[24 item(s), 1045 transaction(s)] done [0.00s].
sorting and recoding items ... [15 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 done [0.00s].
writing ... [344 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Get the summary of these rules.

In here, we got 344 rules associated with the student dataset.

```
> summary(rules_1)
set of 344 rules

rule length distribution (lhs + rhs):sizes
  1   2   3   4   5   6   7 
  1  14  59 111 102  49   8 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00    4.00    4.00    4.39    5.00    7.00 

summary of quality measures:
    support         confidence        coverage           lift       
 Min.   :0.1005   Min.   :0.8043   Min.   :0.1005   Min.   :0.8802  
 1st Qu.:0.1292   1st Qu.:0.8836   1st Qu.:0.1423   1st Qu.:0.9668  
 Median :0.1756   Median :0.9187   Median :0.1919   Median :1.0053  
 Mean   :0.2205   Mean   :0.9164   Mean   :0.2405   Mean   :1.0028  
 3rd Qu.:0.2622   3rd Qu.:0.9477   3rd Qu.:0.2883   3rd Qu.:1.0370  
 Max.   :0.9139   Max.   :1.0000   Max.   :1.0000   Max.   :1.0942  
     count      
 Min.   :105.0  
 1st Qu.:135.0  
 Median :183.5  
 Mean   :230.4  
 3rd Qu.:274.0  
 Max.   :955.0  

mining info:
     data ntransactions support confidence
 data_new          1045     0.1        0.8

call
 apriori(data = data_new, parameter = list(conf = 0.8), appearance = list(rh
s = c("higher=yes"), default = "lhs"))
> 
```

Inspect the above rules.

```
> inspect(rules_1)
```

```
1090    107
[342] {schoolsup=no,
       famsup=yes,
       fatherd=no,
       activities=no,
       nursery=yes,
       internet=yes}    => {higher=yes} 0.1110048  0.9133858 0.1215311 0.999
4641    116
[343] {schoolsup=no,
       fatherd=no,
       activities=no,
       nursery=yes,
       internet=yes,
       romantic=no}     => {higher=yes} 0.1224880  0.9014085 0.1358852 0.986
3579    128
[344] {schoolsup=no,
       famsup=yes,
       fatherd=no,
       nursery=yes,
       internet=yes,
       romantic=no}     => {higher=yes} 0.1550239  0.9818182 0.1578947 1.074
3455    162
>
```

Rule 02 - Get the rules under the confidence of 0.85

```
> #Get the rules under the confidence of 0.85
> rules_2=apriori(data_new,parameter = list(conf=0.85),
+                   appearance = list(rhs=c("higher=yes"),default="lhs"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
       0.85    0.1    1 none FALSE            TRUE       5     0.1      1
 maxlen target   ext
     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 104

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[24 item(s), 1045 transaction(s)] done [0.00s].
sorting and recoding items ... [15 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 done [0.00s].
writing ... [325 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Get the summary of the rules.

In here, we got 325 rules.

```
> summary(rules_2)
set of 325 rules

rule length distribution (lhs + rhs):sizes
  1   2   3   4   5   6   7 
  1  14  57 104  93  48   8 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   4.000   4.000   4.385   5.000   7.000 

summary of quality measures:
    support         confidence        coverage           lift       
 Min.   :0.1005   Min.   :0.8500   Min.   :0.1005   Min.   :0.9301  
 1st Qu.:0.1359   1st Qu.:0.8889   1st Qu.:0.1445   1st Qu.:0.9727  
 Median :0.1818   Median :0.9216   Median :0.1952   Median :1.0085  
 Mean   :0.2260   Mean   :0.9212   Mean   :0.2457   Mean   :1.0080  
 3rd Qu.:0.2699   3rd Qu.:0.9503   3rd Qu.:0.3014   3rd Qu.:1.0398  
 Max.   :0.9139   Max.   :1.0000   Max.   :1.0000   Max.   :1.0942  
     count      
 Min.   :105.0  
 1st Qu.:142.0  
 Median :190.0  
 Mean   :236.1  
 3rd Qu.:282.0  
 Max.   :955.0  

mining info:
     data ntransactions support confidence
 data_new          1045     0.1       0.85
call
 apriori(data = data_new, parameter = list(conf = 0.85), appearance = list(r
hs = c("higher=yes"), default = "lhs"))
> 
```

Inspect the above rules.

```
> inspect(rules_2)
```

```
1090   107
[323] {schoolsup=no,
       famsup=yes,
       fatherd=no,
       activities=no,
       nursery=yes,
       internet=yes}    => {higher=yes} 0.1110048  0.9133858 0.1215311 0.999
4641   116
[324] {schoolsup=no,
       fatherd=no,
       activities=no,
       nursery=yes,
       internet=yes,
       romantic=no}     => {higher=yes} 0.1224880  0.9014085 0.1358852 0.986
3579   128
[325] {schoolsup=no,
       famsup=yes,
       fatherd=no,
       nursery=yes,
       internet=yes,
       romantic=no}     => {higher=yes} 0.1550239  0.9818182 0.1578947 1.074
3455   162
```

Rule 03 - Get the rules under the confidence of 0.87

```
> #Get the rules under the confidence of 0.87
> rules_3=apriori(data_new,parameter = list(conf=0.87),
+                       appearance = list(rhs=c("higher=yes"),default="lhs"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
       0.87    0.1    1 none FALSE            TRUE       5     0.1
 minlen maxlen target  ext
      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 104

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[24 item(s), 1045 transaction(s)] done [0.00s].
sorting and recoding items ... [15 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 done [0.00s].
writing ... [292 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Get the summary of the rules.

In here, we got 292 rules.

```
> summary(rules_3)
set of 292 rules

rule length distribution (lhs + rhs):sizes
 1  2  3  4  5  6  7 
 1 14 52 90 83 44  8 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   4.000   4.000   4.384   5.000   7.000 

summary of quality measures:
    support         confidence        coverage           lift       
 Min.   :0.1005   Min.   :0.8701   Min.   :0.1005   Min.   :0.9521  
 1st Qu.:0.1376   1st Qu.:0.8977   1st Qu.:0.1452   1st Qu.:0.9823  
 Median :0.1861   Median :0.9264   Median :0.2010   Median :1.0137  
 Mean   :0.2328   Mean   :0.9281   Mean   :0.2518   Mean   :1.0155  
 3rd Qu.:0.2837   3rd Qu.:0.9568   3rd Qu.:0.3055   3rd Qu.:1.0469  
 Max.   :0.9139   Max.   :1.0000   Max.   :1.0000   Max.   :1.0942  
     count      
 Min.   :105.0  
 1st Qu.:143.8  
 Median :194.5  
 Mean   :243.3  
 3rd Qu.:296.5  
 Max.   :955.0  

mining info:
     data ntransactions support confidence
 data_new          1045     0.1       0.87
call
 apriori(data = data_new, parameter = list(conf = 0.87), appearance = list(r
hs = c("higher=yes"), default = "lhs"))
> 
```

Inspect the above rules.

```
> inspect(rules_3)
      lhs                     rhs          support   confidence  coverage
lift  count
[1]   {}                   => {higher=yes} 0.9138756  0.9138756 1.0000000 1.000
0000   955
[2]   {schoolsup=yes}      => {higher=yes} 0.1110048  0.9747899 0.1138756 1.066
6549   116
[3]   {nursery=no}         => {higher=yes} 0.1779904  0.8899522 0.2000000 0.973
8220   186
[4]   {internet=no}        => {higher=yes} 0.1827751  0.8801843 0.2076555 0.963
1336   191
[5]   {fatherd=yes}        => {higher=yes} 0.2066986  0.9818182 0.2105263 1.074
3455   216
[6]   {romantic=yes}       => {higher=yes} 0.3110048  0.8760108 0.3550239 0.958
5668   325
[7]   {famsup=no}          => {higher=yes} 0.3416268  0.8836634 0.3866029 0.966
9405   357
[8]   {activities=yes}     => {higher=yes} 0.4602871  0.9321705 0.4937799 1.020
0191   481
[9]   {activities=no}      => {higher=yes} 0.4535885  0.8977273 0.5052632 0.982
3298   474
[10]  {famsup=yes}         => {higher=yes} 0.5722488  0.9343750 0.6124402 1.022
4313   598
[11]  {romantic=no}        => {higher=yes} 0.6028708  0.9361070 0.6440191 1.024
```

Step 12

Visualize these rules.

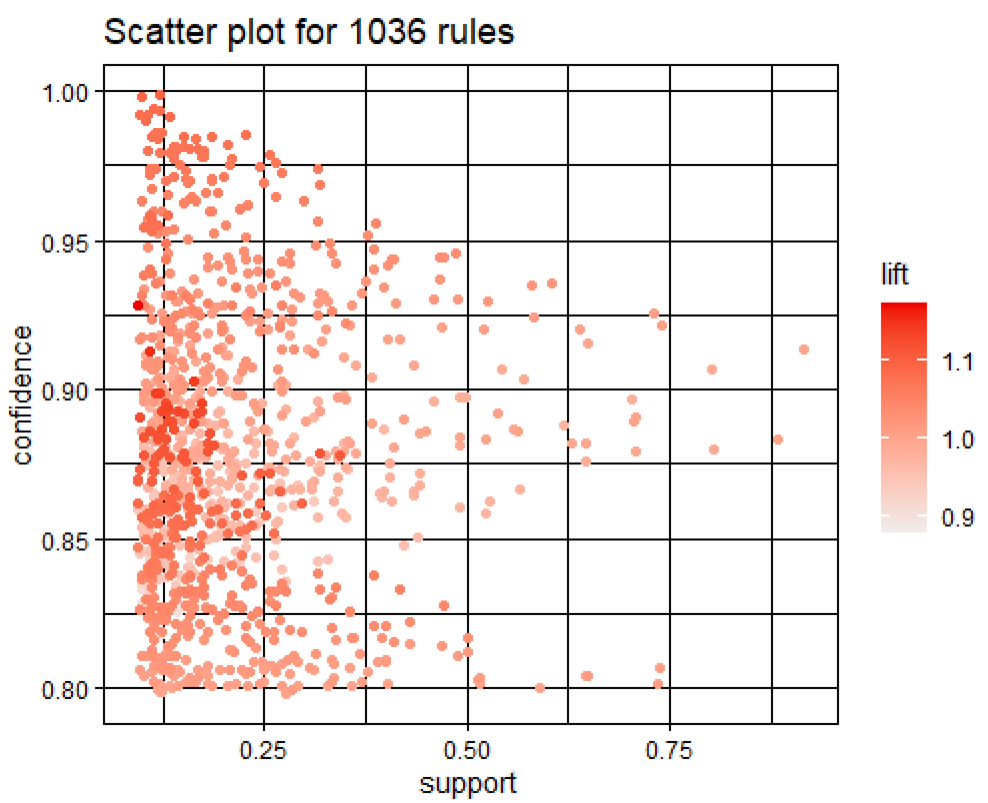
Install and load the arulesViz () package.

```
install.packages("arulesviz")
library(arulesviz)
```

Step 13

Plot the rules.

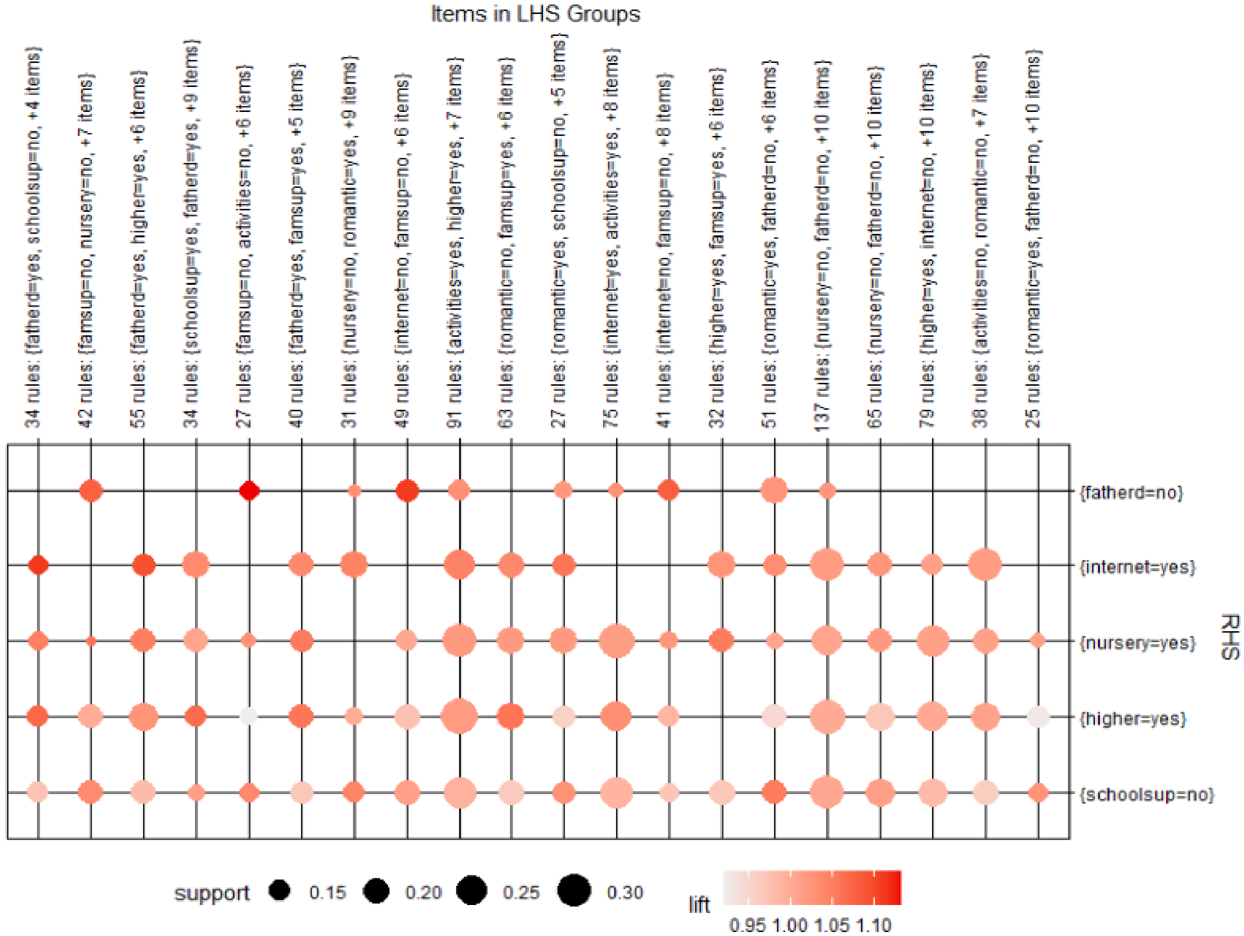
```
> plot(rules)
```



Step 14

Plot the rules in groups.

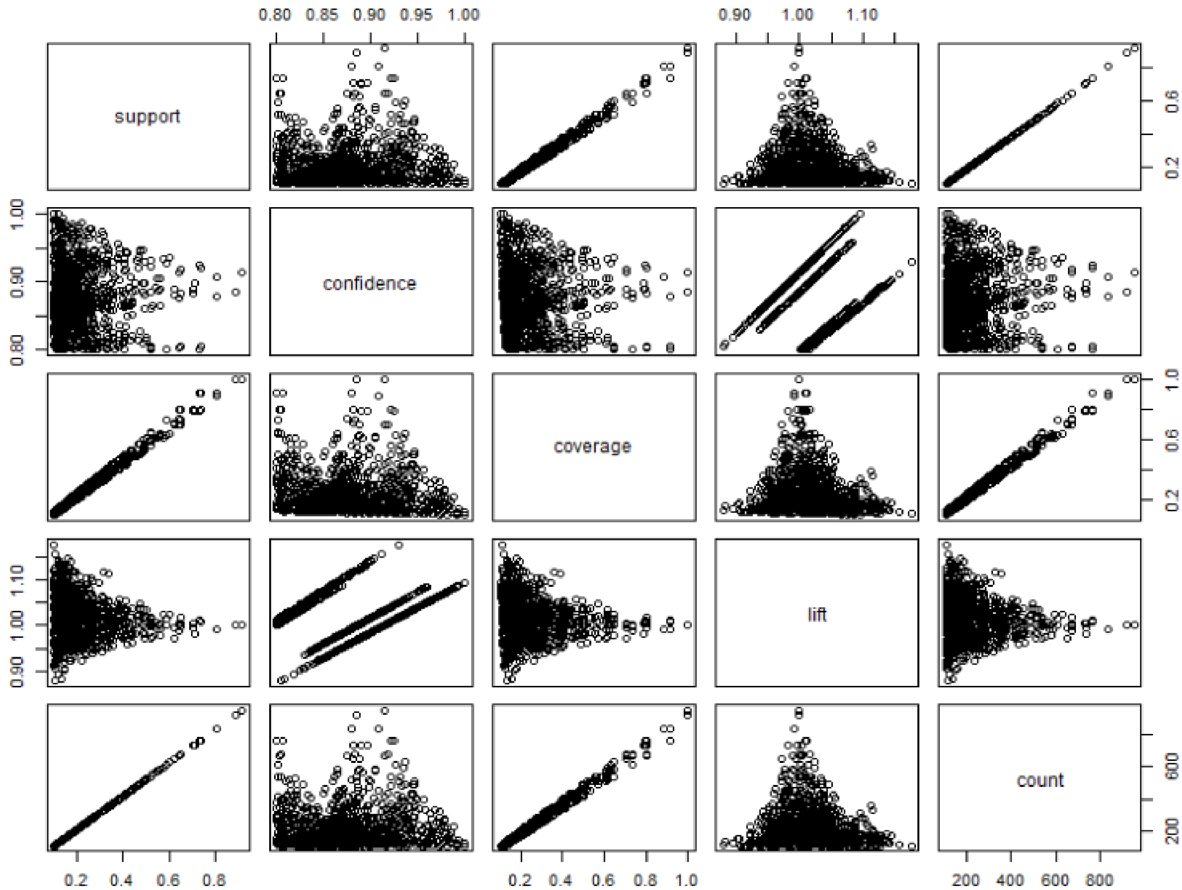
```
> plot(rules,method = "grouped")
>
```



Step 15

Display a scatterplot matrix to compare the support, confidence, and lift.

```
> plot(rules@quality)
>
```



Step 16

Get the rules with only “yes” items on the left hand side as well as on the right hand side.

```
> #get the rules with only items "Yes" on left hand side and right-hand sid
e:
> rules_new=apriori(data_new,parameter=list(conf=0.87),
+                   appearance=list(rhs=c("higher=yes"),
+                                   lhs=c("schoolsup=yes","fatherd=yes","act
ivities=yes","nursery=yes","internet=yes","romantic=yes"),
+                                   default="none"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
       0.87    0.1    1 none FALSE            TRUE       5     0.1
 minlen maxlen target   ext
      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 104

set item appearances ...[7 item(s)] done [0.00s].
set transactions ...[7 item(s), 1045 transaction(s)] done [0.00s].
sorting and recoding items ... [7 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [22 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Get the summary of the above rules.

In here, we got 22 rules.

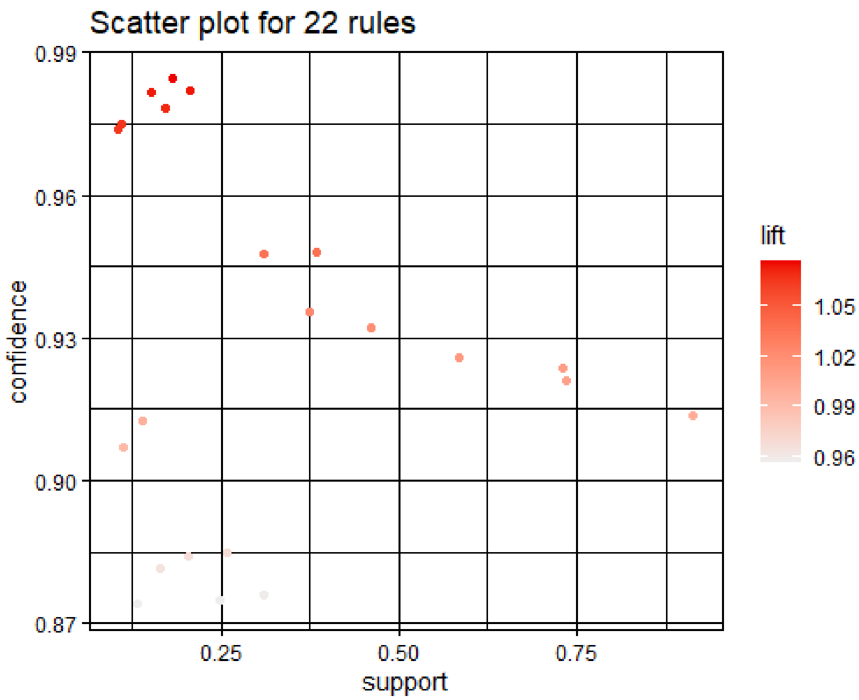
```
> summary(rules_new)
set of 22 rules

rule length distribution (lhs + rhs):sizes
1 2 3 4 5
1 6 9 5 1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   2.955   3.750   5.000

summary of quality measures:
    support          confidence        coverage           lift
 Min.   :0.1062   Min.   :0.8742   Min.   :0.1091   Min.   :0.9566
 1st Qu.:0.1550   1st Qu.:0.8904   1st Qu.:0.1605   1st Qu.:0.9743
 Median :0.2273   Median :0.9248   Median :0.2574   Median :1.0120
 Mean   :0.3181   Mean   :0.9281   Mean   :0.3444   Mean   :1.0155
 3rd Qu.:0.3821   3rd Qu.:0.9673   3rd Qu.:0.4043   3rd Qu.:1.0584
 Max.   :0.9139   Max.   :0.9845   Max.   :1.0000   Max.   :1.0773
     count
 Min.   :111.0
 1st Qu.:162.0
 Median :237.5
 Mean   :332.4
 3rd Qu.:399.2
 Max.   :955.0

mining info:
     data ntransactions support confidence
 data_new          1045     0.1       0.87
```

Inspect the above rules.

```
> inspect(rules_new)
     lhs                   rhs           support   confidence  coverage      l
ift count
[1]  {}                 => {higher=yes} 0.9138756  0.9138756 1.0000000 1.0000
000  955
[2]  {schoolsup=yes}    => {higher=yes} 0.1110048  0.9747899 0.1138756 1.0666
549  116
[3]  {fatherd=yes}      => {higher=yes} 0.2066986  0.9818182 0.2105263 1.0743
455  216
[4]  {romantic=yes}     => {higher=yes} 0.3110048  0.8760108 0.3550239 0.9585
668  325
[5]  {activities=yes}   => {higher=yes} 0.4602871  0.9321705 0.4937799 1.0200
191  481
[6]  {nursery=yes}      => {higher=yes} 0.7358852  0.9209581 0.7990431 1.0077
499  769
[7]  {internet=yes}     => {higher=yes} 0.7311005  0.9238210 0.7913876 1.0108
827  764
[8]  {fatherd=yes,
      activities=yes}   => {higher=yes} 0.1062201  0.9736842 0.1090909 1.0654
450  111
[9]  {fatherd=yes
```

Plot the result.

```
> plot(rules_new)
>
```



Step 15

Explore Association rules using interactive manipulations and viewing using shiny.

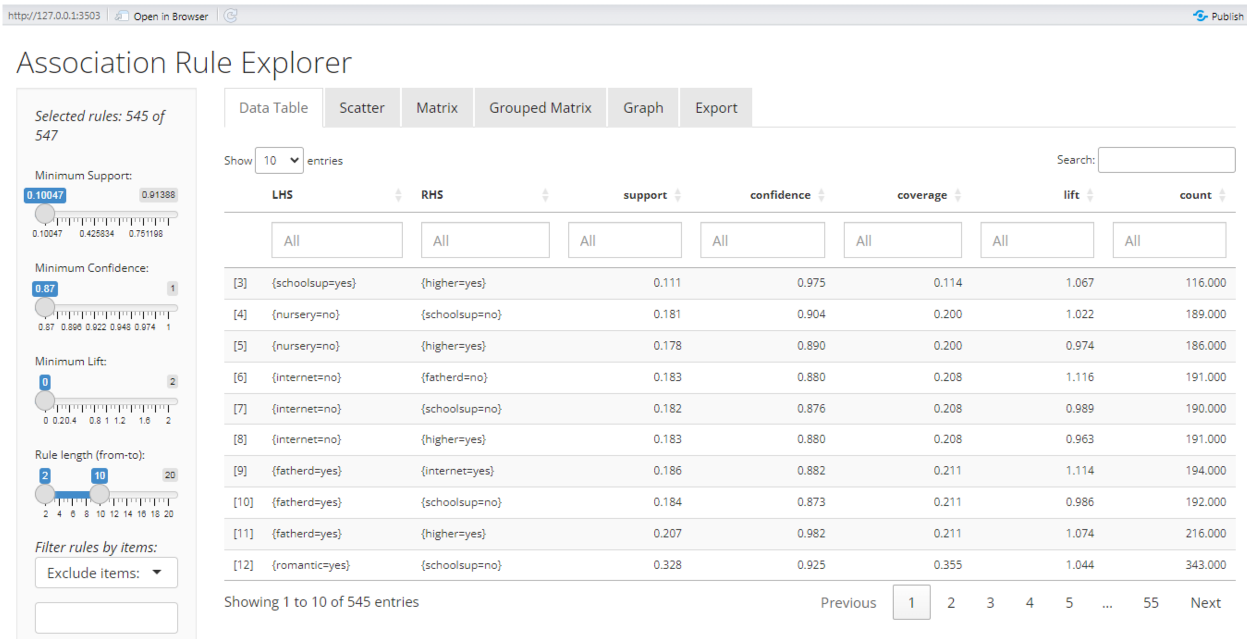
Install and load the arulesviz () package and get the rules under the confidence of 0.87.

```
> library(arulesviz)
> rules_ex=apriori(data_new,parameter = list(conf=0.87))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support
       0.87    0.1    1 none FALSE            TRUE       5     0.1
 minlen maxlen target   ext
      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 104

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[24 item(s), 1045 transaction(s)] done [0.00s].
sorting and recoding items ... [15 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 done [0.00s].
writing ... [547 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

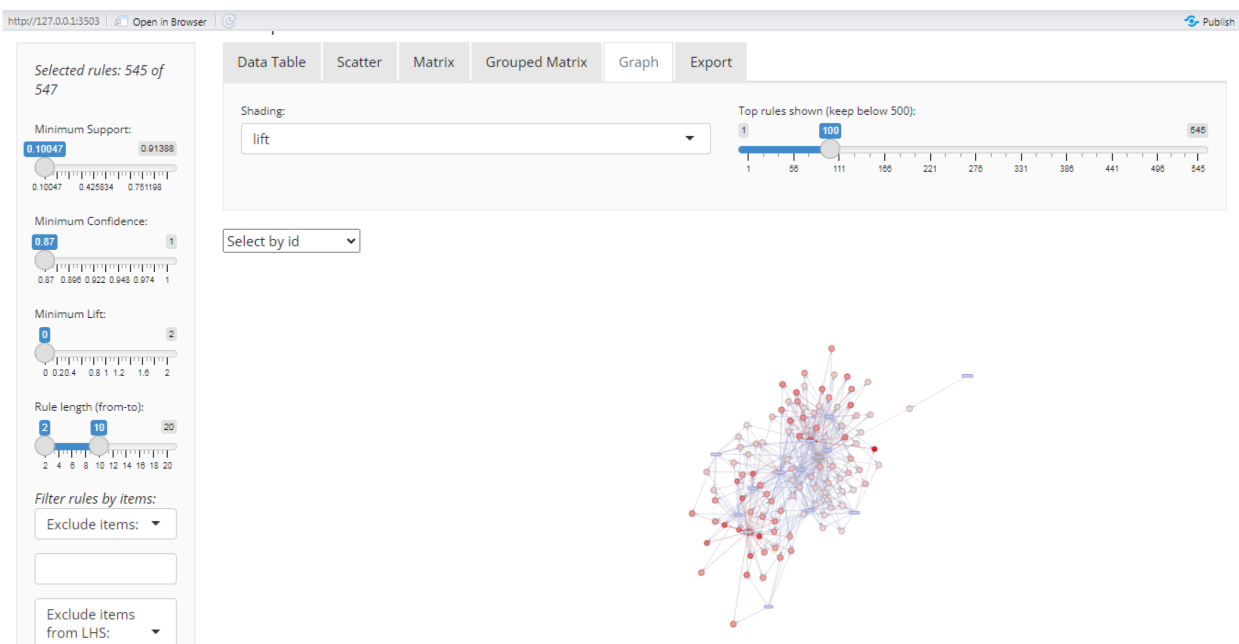
Step 16

Explore association rules using ruleExplorer() function.

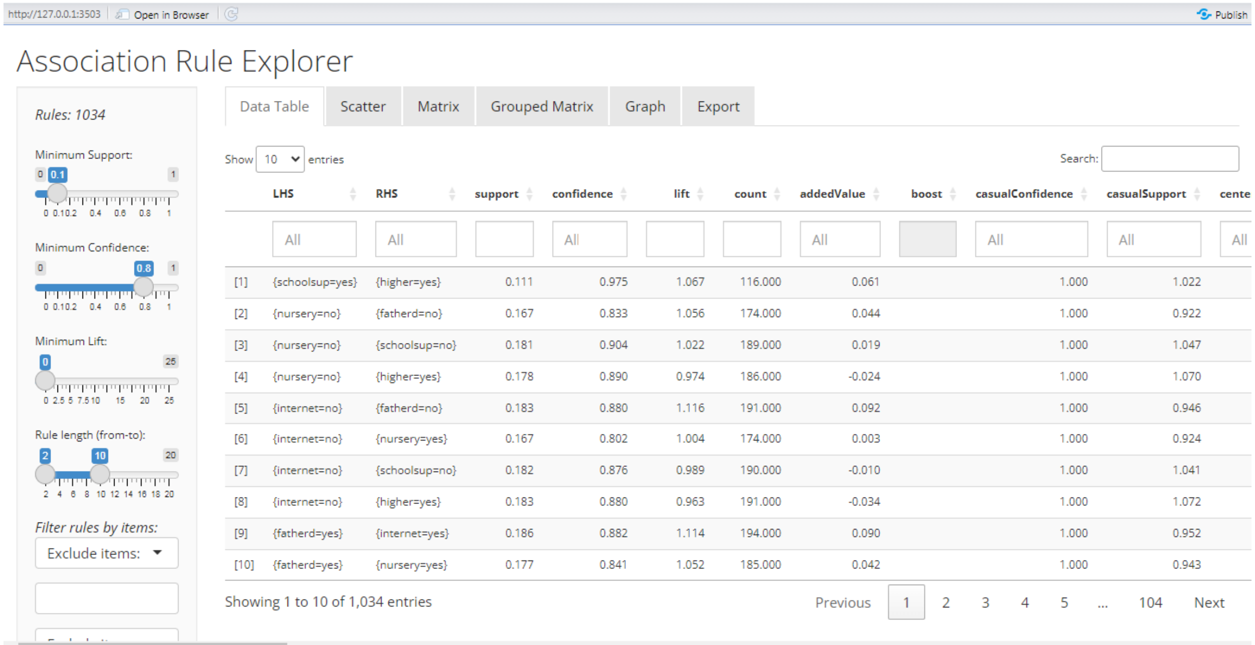
```
ruleExplorer(rules_ex)
```



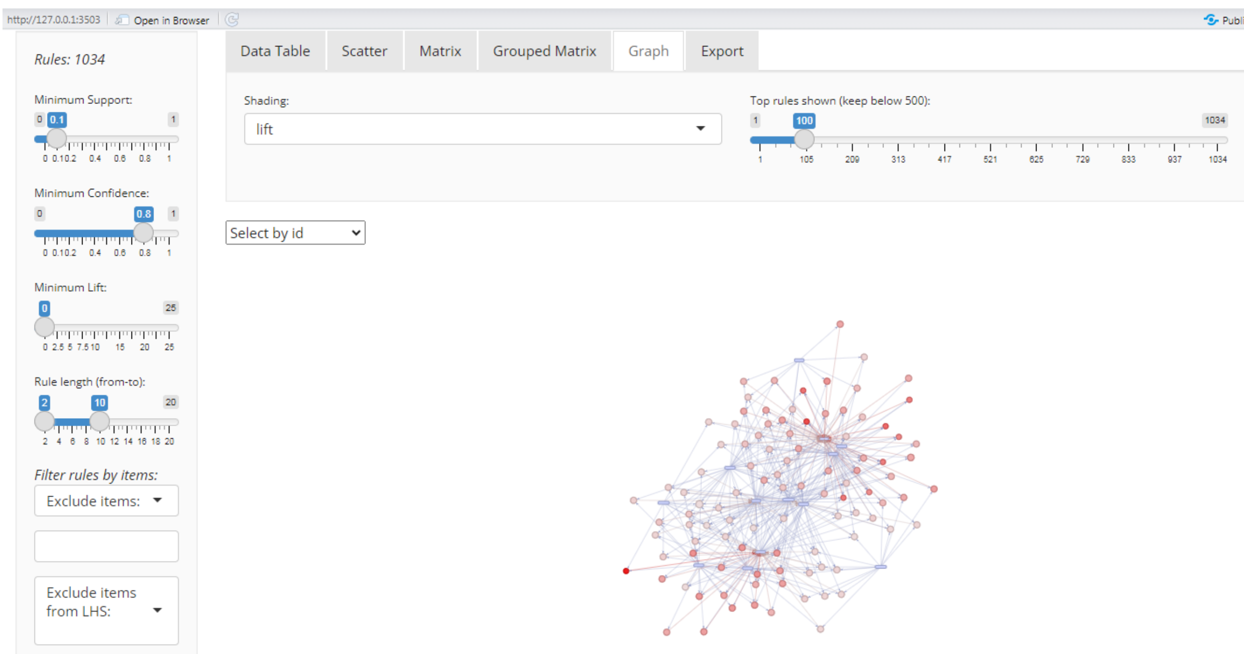
	LHS	RHS	support	confidence	coverage	lift	count
[3]	{schoolsup=yes}	{higher=yes}	0.111	0.975	0.114	1.067	116.000
[4]	{nursery=no}	{schoolsup=no}	0.181	0.904	0.200	1.022	189.000
[5]	{nursery=no}	{higher=yes}	0.178	0.890	0.200	0.974	186.000
[6]	{internet=no}	{fatherd=no}	0.183	0.880	0.208	1.116	191.000
[7]	{internet=no}	{schoolsup=no}	0.182	0.876	0.208	0.989	190.000
[8]	{internet=no}	{higher=yes}	0.183	0.880	0.208	0.963	191.000
[9]	{fatherd=yes}	{internet=yes}	0.186	0.882	0.211	1.114	194.000
[10]	{fatherd=yes}	{schoolsup=no}	0.184	0.873	0.211	0.986	192.000
[11]	{fatherd=yes}	{higher=yes}	0.207	0.982	0.211	1.074	216.000
[12]	{romantic=yes}	{schoolsup=no}	0.328	0.925	0.355	1.044	343.000



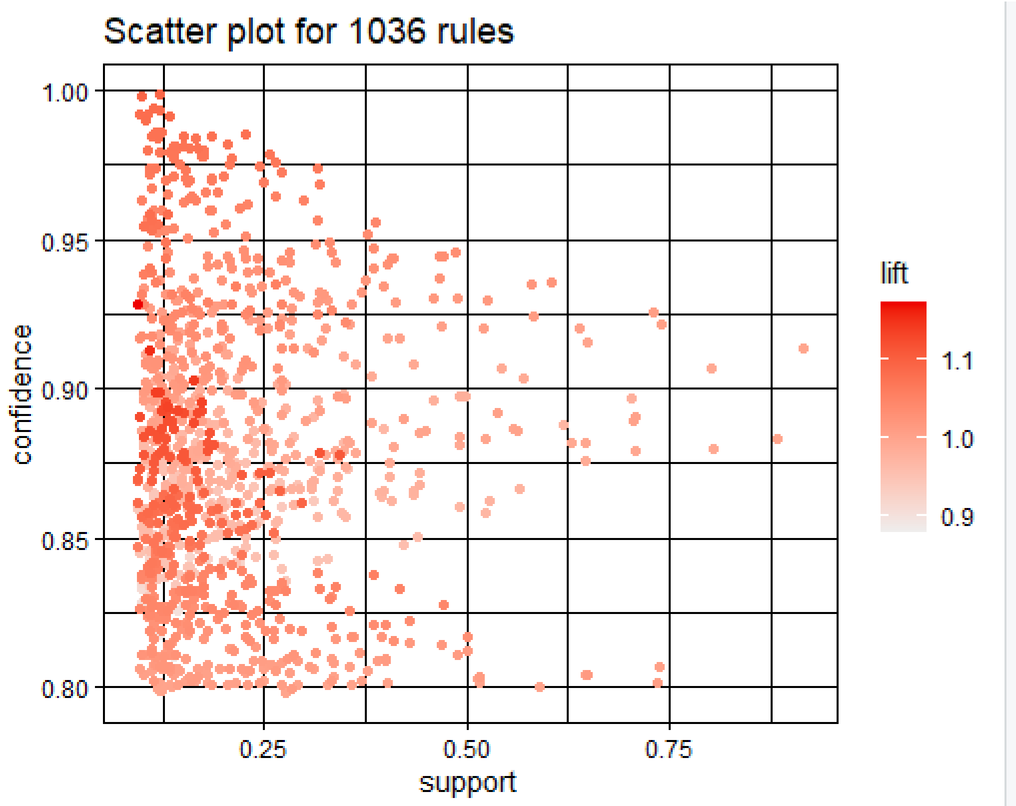
```
> ruleExplorer(data_new)
```



	LHS	RHS	support	confidence	lift	count	addedValue	boost	casualConfidence	casualSupport	cente
[1]	{schoolsup=yes}	{higher=yes}	0.111	0.975	1.067	116.000	0.061		1.000	1.022	
[2]	{nursery=no}	{fatherd=no}	0.167	0.833	1.056	174.000	0.044		1.000	0.922	
[3]	{nursery=no}	{schoolsup=no}	0.181	0.904	1.022	189.000	0.019		1.000	1.047	
[4]	{nursery=no}	{higher=yes}	0.178	0.890	0.974	186.000	-0.024		1.000	1.070	
[5]	{internet=no}	{fatherd=no}	0.183	0.880	1.116	191.000	0.092		1.000	0.946	
[6]	{internet=no}	{nursery=yes}	0.167	0.802	1.004	174.000	0.003		1.000	0.924	
[7]	{internet=no}	{schoolsup=no}	0.182	0.876	0.989	190.000	-0.010		1.000	1.041	
[8]	{internet=no}	{higher=yes}	0.183	0.880	0.963	191.000	-0.034		1.000	1.072	
[9]	{fatherd=yes}	{internet=yes}	0.186	0.882	1.114	194.000	0.090		1.000	0.952	
[10]	{fatherd=yes}	{nursery=yes}	0.177	0.841	1.052	185.000	0.042		1.000	0.943	



6) Results, Analysis, and Discussions

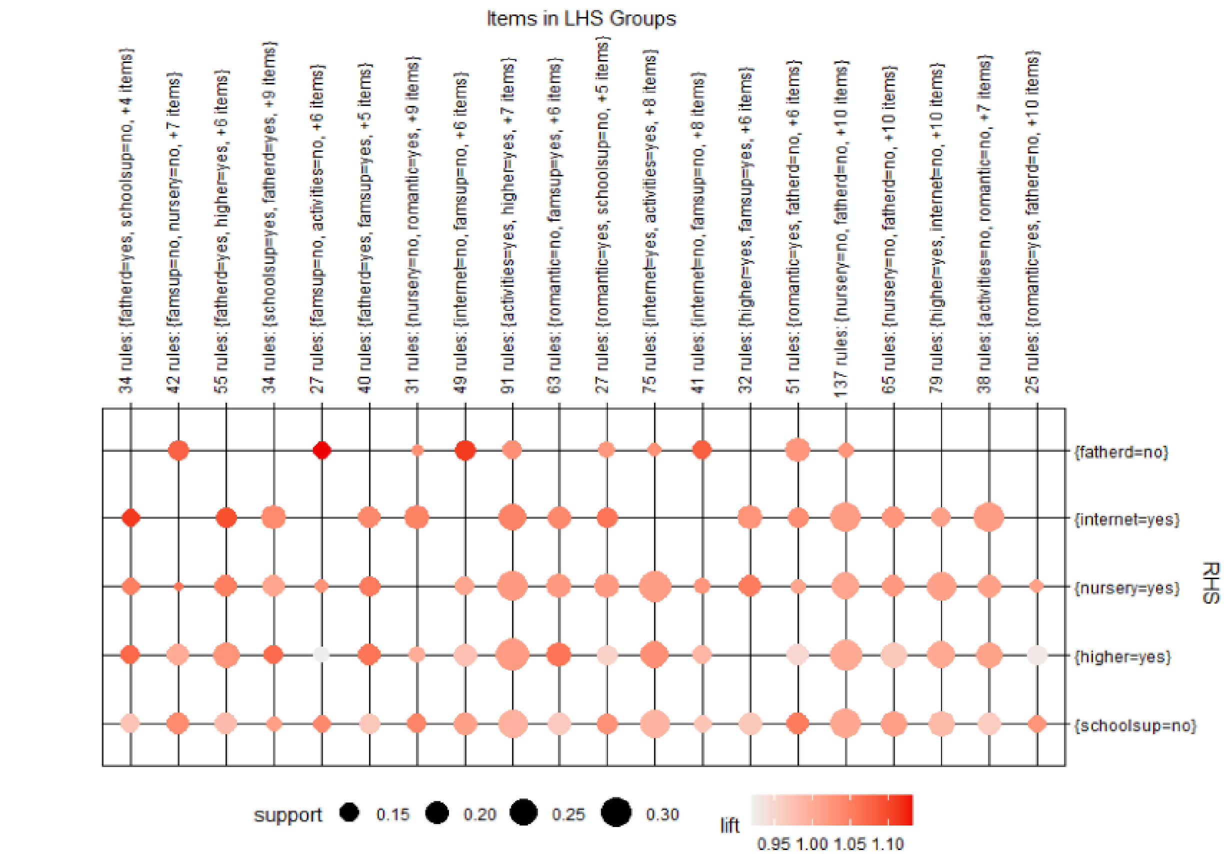


This is a scatter plot for 1036 rules. The x-axis is labeled "support" and the y-axis is labeled "lift". In scatter plots, data points are used to represent the relationship between two variables. In this case, each data point represents a rule, and the position of the point on the graph shows the rule's support and lift.

Support refers to the proportion of times that a rule applies to a data point. Lift refers to the ratio of the probability of a positive outcome occurring given the rule is applied, compared to the probability of a positive outcome occurring in general.

Based on the data points in the scatter plot, there appears to be a weak positive correlation between support and lift. This means that as the support of a rule increases, the lift of the rule also tends to increase. There are also a few outliers, which are data points that fall far away from the majority of the other points. These outliers may represent rules that have either very high or very low lift, even though they have high support.

Overall, the graph suggests that there is a positive correlation between support and the number of items in the group.



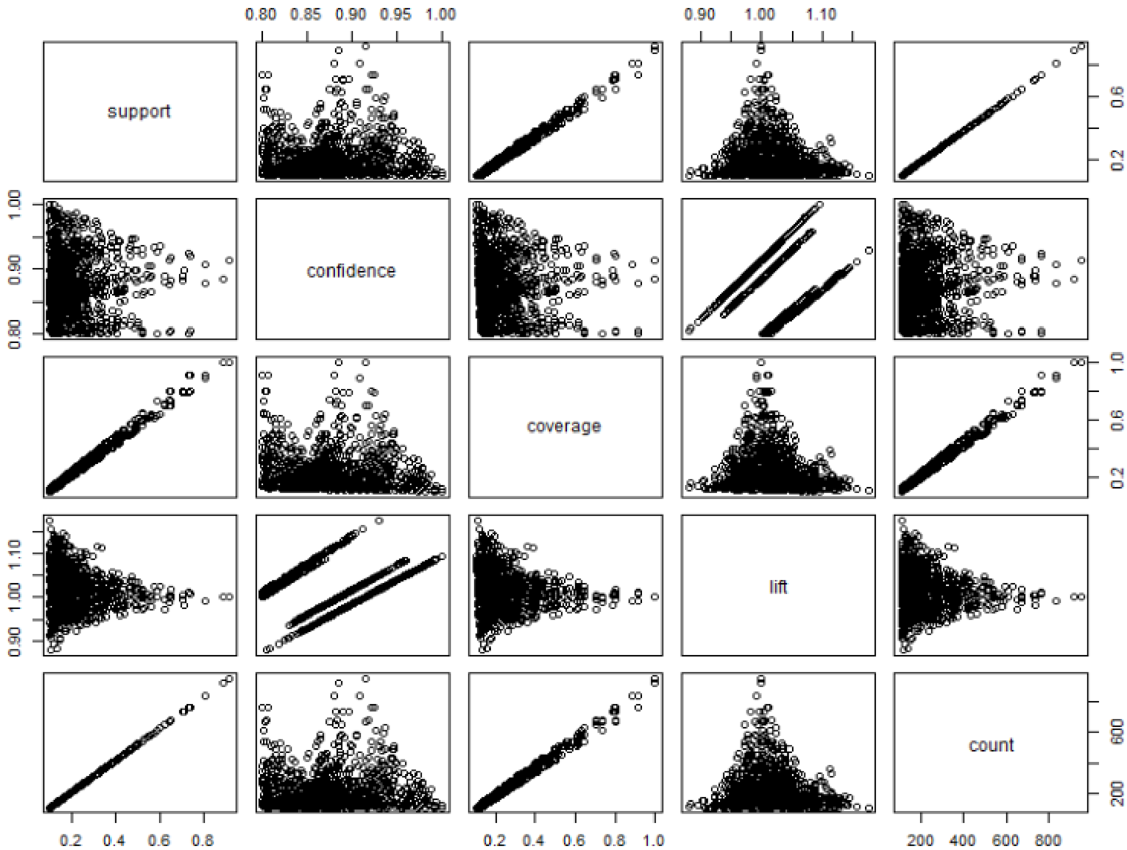
This is a line graph titled "Items in LHS Groups". The x-axis is labeled "support" with values ranging from 0.15 to 1.10. The y-axis represents the number of items in the group. There are several data series plotted on the graph, each representing a different rule group.

Line graphs are used to show trends over time or another continuous variable. In this case, the line graph shows how the number of items in a group changes as the support for the group increases.

Here are some additional details that can be seen from the graph:

- The data series with the label "{fatherd=no, higher yes, +10 items}" has the highest support values and the highest number of items in a group.
- The data series with the label "{famsup=no, activities=no, +6 items}" has a relatively low support value and a low number of items in a group.

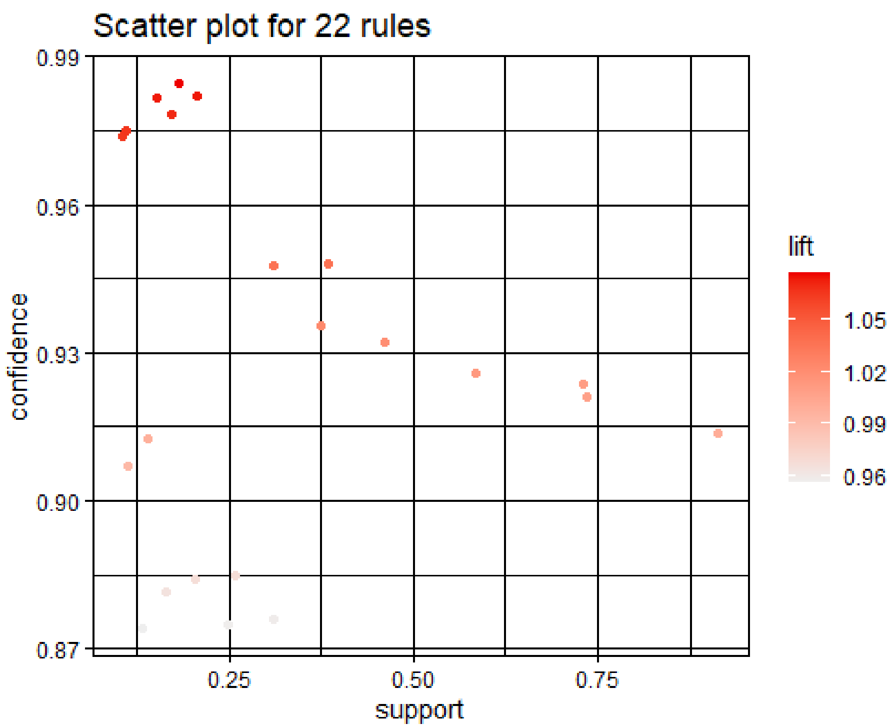
There is a lot of variability in the number of items in the group for a given level of support. This suggests that there may be other factors that influence the number of items in a group besides support. Overall, the graph suggests that there is a positive correlation between support and the number of items in the group.



This is a scatter plot for 1036 rules. It visualizes the performance of a machine learning model. It has a set of four scatter plots, along with precision-recall curves. These plots show the relationship between several metrics, including support, confidence, coverage, lift, and count.

- The top left scatter plot shows support on the x-axis and confidence on the y-axis. There are three clusters of data points in this plot.
- The top right scatter plot shows support on the x-axis and coverage on the y-axis. There's a faint curve going through a cloud of data points.
- The bottom left scatter plot shows confidence on the x-axis and lift on the y-axis. There are multiple curves in this plot, and the x-axis cuts through the y-axis at around 1.
- The bottom right scatter plot shows confidence on the x-axis and count on the y-axis. There's a curve that goes through a cloud of data points.

Overall, the visualization helps us to understand how the model is performing across a variety of metrics.



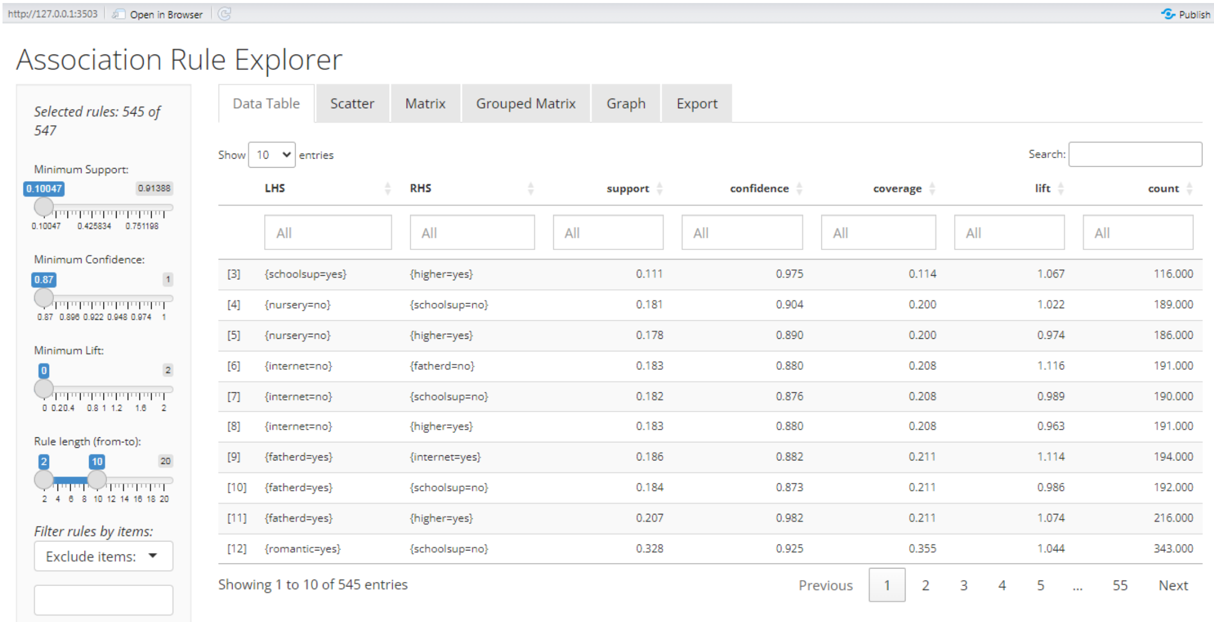
This is a scatter plot for 22 rules. The y-axis of the top left and bottom left scatter plot represents the confidence level. Confidence level is the number of transactions that satisfy both the antecedent and the consequent of a rule, divided by the number of transactions that satisfy the support.

The y-axis of the bottom left scatter plot represents lift. Lift is a ratio of the probability of a transaction satisfying both the antecedent and consequent of a rule, divided by the probability of the transaction satisfying only the support. A lift value greater than 1 indicates that the rule is interesting, because the consequent is more likely to happen given the antecedent, than if the support and consequent were independent.

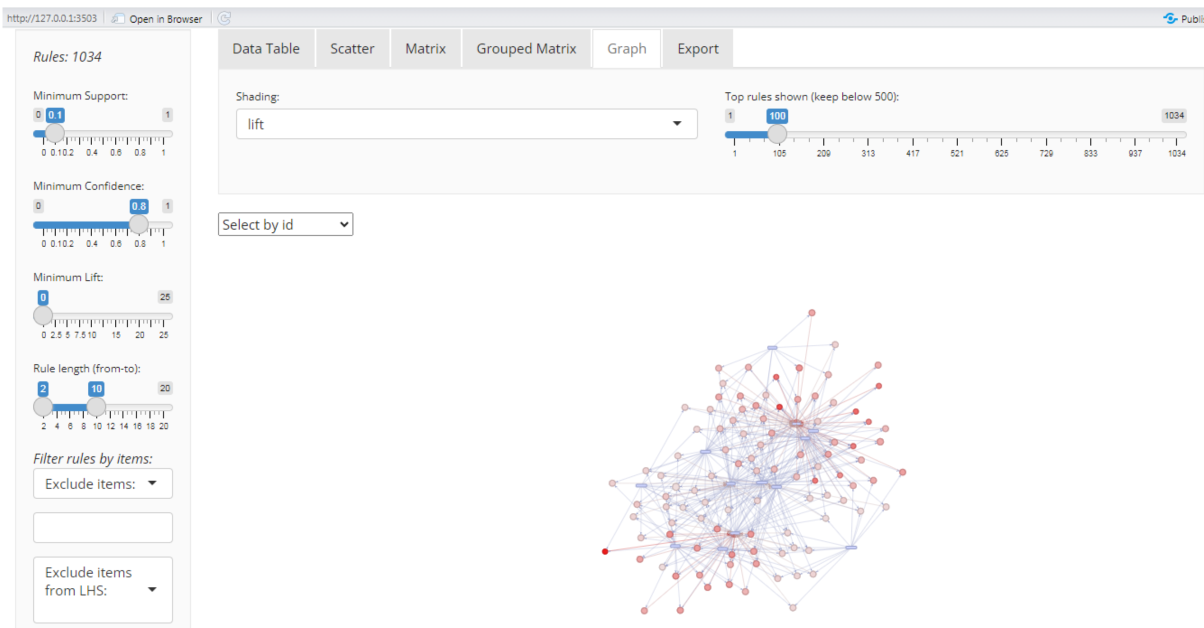
The x-axis of both the top left and top right scatter plots represents support. Support is the number of transactions in the dataset that satisfy both the antecedent and consequent of a rule, divided by the total number of transactions in the dataset.

The y-axis of the top right scatter plot represents coverage. Coverage is the proportion of transactions in the dataset that satisfy the antecedent of the rule.

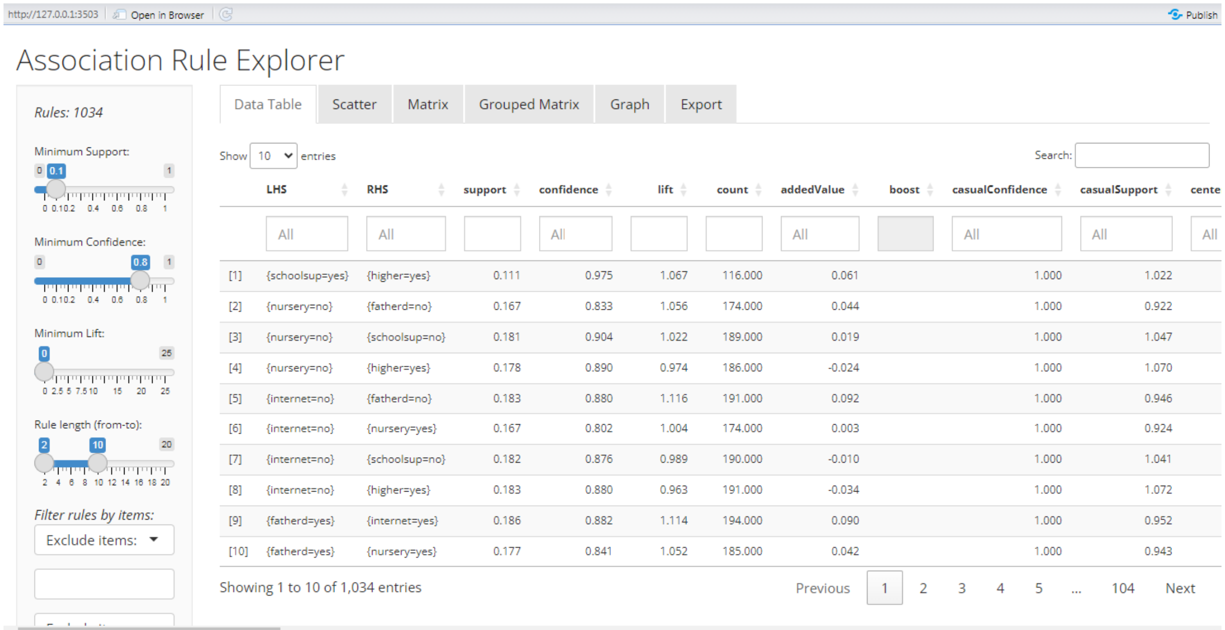
Overall, by looking at the graph we cannot find a clear correlation between confidence and lift.



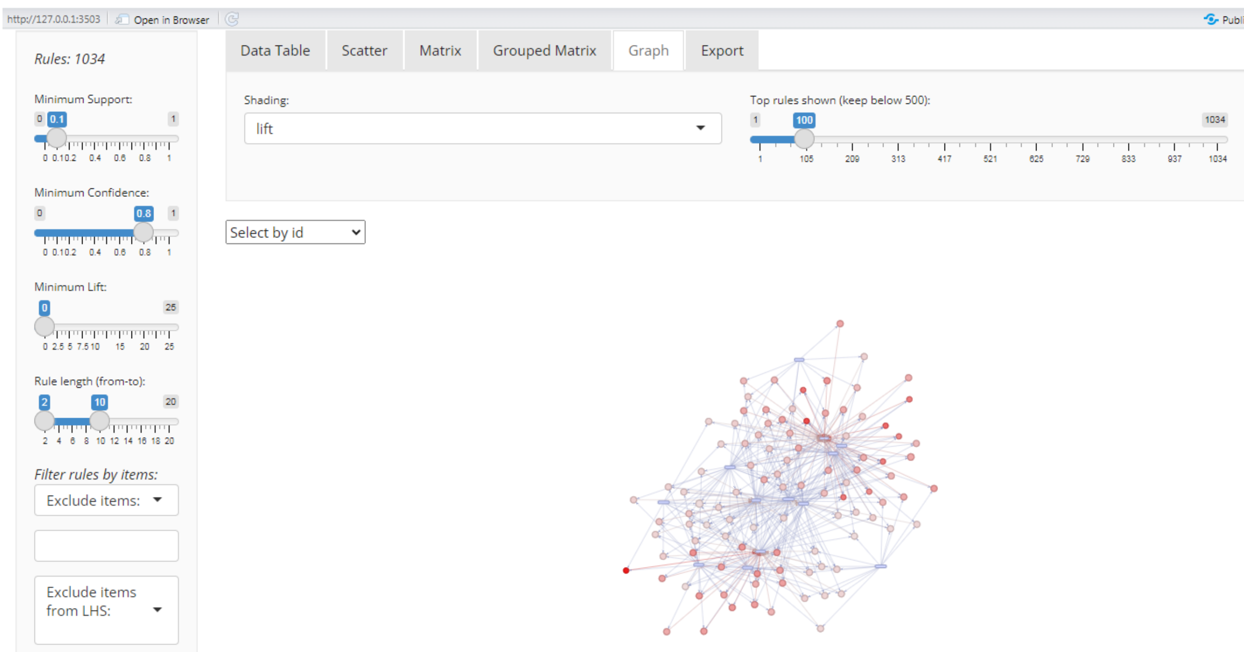
	LHS	RHS	support	confidence	coverage	lift	count
[3]	{schoolsup=yes}	{higher=yes}	0.111	0.975	0.114	1.067	116.000
[4]	{nursery=no}	{schoolsup=no}	0.181	0.904	0.200	1.022	189.000
[5]	{nursery=no}	{higher=yes}	0.178	0.890	0.200	0.974	186.000
[6]	{internet=no}	{fatherd=no}	0.183	0.880	0.208	1.116	191.000
[7]	{internet=no}	{schoolsup=no}	0.182	0.876	0.208	0.989	190.000
[8]	{internet=no}	{higher=yes}	0.183	0.880	0.208	0.963	191.000
[9]	{fatherd=yes}	{internet=yes}	0.186	0.882	0.211	1.114	194.000
[10]	{fatherd=yes}	{schoolsup=no}	0.184	0.873	0.211	0.986	192.000
[11]	{fatherd=yes}	{higher=yes}	0.207	0.982	0.211	1.074	216.000
[12]	{romantic=yes}	{schoolsup=no}	0.328	0.925	0.355	1.044	343.000



The above result shows 545 possible rules which we were generated by doing association rule mining for the student dataset. Each rule consists of two parts: the left-hand side (LHS) and the right-hand side (RHS). The LHS represents a condition that must be met, and the RHS represents the outcome that is likely to happen given the LHS condition. This helps us in identifying frequent patterns or relationships between different student attributes.



	LHS	RHS	support	confidence	lift	count	addedValue	boost	casualConfidence	casualSupport
[1]	{schoolsup=yes}	{higher=yes}	0.111	0.975	1.067	116.000	0.061		1.000	1.022
[2]	{nursery=no}	{fatherd=no}	0.167	0.833	1.056	174.000	0.044		1.000	0.922
[3]	{nursery=no}	{schoolsup=no}	0.181	0.904	1.022	189.000	0.019		1.000	1.047
[4]	{nursery=no}	{higher=yes}	0.178	0.890	0.974	186.000	-0.024		1.000	1.070
[5]	{internet=no}	{fatherd=no}	0.183	0.880	1.116	191.000	0.092		1.000	0.946
[6]	{internet=no}	{nursery=yes}	0.167	0.802	1.004	174.000	0.003		1.000	0.924
[7]	{internet=no}	{schoolsup=no}	0.182	0.876	0.989	190.000	-0.010		1.000	1.041
[8]	{internet=no}	{higher=yes}	0.183	0.880	0.963	191.000	-0.034		1.000	1.072
[9]	{fatherd=yes}	{internet=yes}	0.186	0.882	1.114	194.000	0.090		1.000	0.952
[10]	{fatherd=yes}	{nursery=yes}	0.177	0.841	1.052	185.000	0.042		1.000	0.943



The above result shows 1034 rules which we were discovered by doing association rule mining for the student dataset. This rules suggest that the specific student characteristics represented by the LHS codes are strongly associated with the outcome represented by the RHS code (confidence is high). This helps us in identifying frequent patterns or relationships between different student attributes.

7) Conclusion

Association rule mining is a technique in data mining used to discover interesting relationships, patterns, or associations among variables in large datasets. It identifies rules that describe the correlation between different variables or items within the dataset. By analyzing the above student dataset using association rule mining, it helped us to uncover meaningful associations between various attributes or characteristics of students. For example, we can say that the students who receive educational support from both the school and the family are more likely to have higher academic performance and the students who participate in extra-curricular activities are more likely to have a desire to pursue higher education. By identifying such patterns, educators, researchers, and policymakers can gain valuable insights into the factors that influence student outcomes. Also this information can be used to design targeted interventions, improve support systems, and tailor educational programs to better meet the needs of students. Overall, association rule mining serves as a powerful tool in uncovering hidden relationships within student datasets, ultimately it contributes in more informed decision-making and the enhancement of educational practices.

8) References

https://github.com/Emmanuel96/apriori_association_rule_mining/tree/master/Dataset

Task 02 - Regression Analysis using Diabetes Dataset

1) Introduction

Diabetes is a widespread illness that can afflict individuals of any age. Diabetes results from an excessively high blood sugar (glucose) level in the body. The primary energy source for our bodies is glucose, which is primarily derived from the carbohydrates found in food and beverages which we consume in our day to day lives. The majority of diabetes types are chronic but treatable with medication and lifestyle modifications. Diabetes health issues may be less likely to arise if diabetes is prevented or managed.

This report provides information on diabetic people with Pima Indian ancestry. The aim of creating this report is to forecast when diabetes will manifest by using diagnostic measurements. This report clearly describes each step of performing a regression analysis using R on the dataset in a clear and organized manner.

2) Data Set

The data set was taken from: https://data.world/data-society/pima-indians-diabetes-database

The National Institute of Diabetes and Digestive and Kidney Diseases is the original source of this dataset. The goal of this data set is to determine if a patient has diabetes or not using diagnostic measurements. These examples were chosen from a bigger database under a number of restrictions. Specifically, all of the patients in this dataset are Pima Indian women who are at least 21 years old.

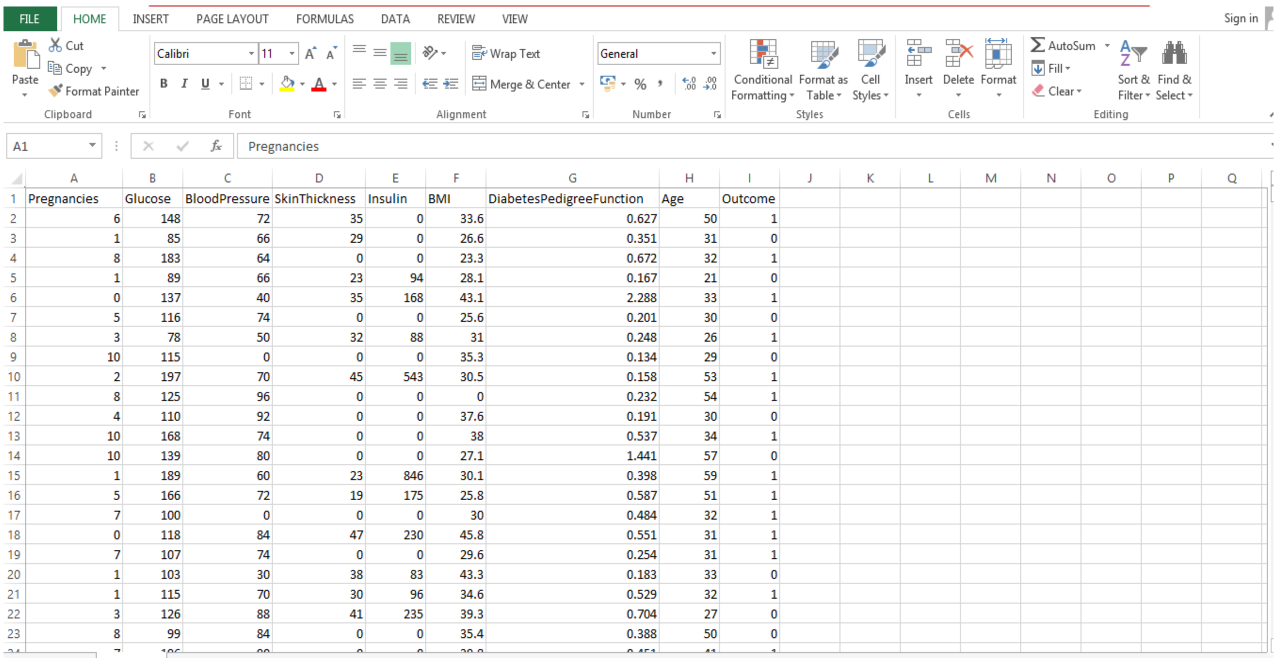
3) Explanation and Preparation of the Data Set

a. Explanation of the Data Set

This dataset contains information about the diabetes patients in Pima Indian heritage. There are 9 columns and 769 rows in the data set.

Attributes of the data set are,

1. Pregnancies: Number of times pregnant
2. Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. BloodPressure: Diastolic blood pressure (mm Hg)
4. SkinThickness: Triceps skin fold thickness (mm)
5. Insulin: 2-Hour serum insulin (mu U/ml)
6. BMI: Body mass index (weight in kg/(height in m)^2)
7. DiabetesPedigreeFunction: Diabetes pedigree function
8. Age: Age (years)
9. Outcome: Class variable (0 or 1)



Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0

b. Preparation of the dataset

Before doing the regression analysis, we have checked for missing values in the dataset.

```
> data=read.csv("diabetes.csv")
> View(data)
> data=read.csv("diabetes.csv")
> is.na(data)
      Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
 [1,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [2,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [3,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [4,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [5,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [6,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [7,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [8,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
 [9,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[10,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[11,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[12,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[13,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[14,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[15,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[16,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
[17,]       FALSE   FALSE         FALSE         FALSE   FALSE FALSE
```

Since there were not missing values found in the dataset we had changed the column order of the dataset as follows for the easier analysis purpose.

```
> data_1=data[,c(2:9)]
> View(data_1)
> final=data_1[,c(7,1:6,8)]
> View(final)
>
```

LR 1.R | final | data_1 | data

Filter

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	148	72	35	0	33.6	0.627	50	1
2	85	66	29	0	26.6	0.351	31	0
3	183	64	0	0	23.3	0.672	32	1
4	89	66	23	94	28.1	0.167	21	0
5	137	40	35	168	43.1	2.288	33	1
6	116	74	0	0	25.6	0.201	30	0
7	78	50	32	88	31.0	0.248	26	1
8	115	0	0	0	35.3	0.134	29	0
9	197	70	45	543	30.5	0.158	53	1

Showing 1 to 9 of 768 entries, 8 total columns

The final dataset can be shown as follows.

LR 1.R | final | data_1 | data

Filter

	Age	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Outcome
1	50	148	72	35	0	33.6	0.627	1
2	31	85	66	29	0	26.6	0.351	0
3	32	183	64	0	0	23.3	0.672	1
4	21	89	66	23	94	28.1	0.167	0
5	33	137	40	35	168	43.1	2.288	1
6	30	116	74	0	0	25.6	0.201	0
7	26	78	50	32	88	31.0	0.248	1
8	29	115	0	0	0	35.3	0.134	0
9	53	197	70	45	543	30.5	0.158	1

Showing 1 to 9 of 768 entries, 8 total columns

4) Regression Analysis

In data mining, regression analysis is a statistical method used to investigate and model the relationship between one or more independent variables (also known as predictors or features) and a dependent variable (also known as the outcome or target variable). The aim of regression analysis is to find the relationship between changes in the independent variables and changes in the dependent variable.

Regression analysis approaches come in several forms such as:

1. **Linear Regression:** This type of regression analysis is the most basic, assuming a linear connection between the independent and dependent variables. The goal of linear regression is to minimize the discrepancies between the dependent variable's observed and predicted values by fitting a straight line to the data.
2. **Multiple Regression:** The dependent variable is predicted using a number of independent factors in multiple regression. Each independent variable, while keeping other variables constant, has a coefficient that indicates the direction and intensity of its association with the dependent variable.
3. **Logistic Regression:** Logistic regression is a type of regression analysis that is frequently applied to data mining problems involving binary categorization. It simulates the likelihood of a binary result depending on one or more independent variables (such as the existence or lack of an illness). By fitting data to a logistic curve, logistic regression calculates the likelihood that an event will occur.

5) Implementation in R

Packages used

1) **party**: The `party` package is used for statistical learning and data mining with decision trees. It provides tools for fitting, visualizing, and interpreting classification and regression trees.

2) **epitools**: `epitools` is a package for epidemiologic data and analysis in R. It offers functions for calculating various epidemiological measures such as prevalence, incidence, and mortality rates. It also provides tools for analyzing contingency tables, calculating confidence intervals, and conducting hypothesis tests for epidemiological studies.

3) **ggplot2**: This is a popular package for data visualization in R. `ggplot2` supports a wide range of plot types, including scatter plots, bar plots, histograms, and more.

4) **GGally**: `GGally` extends the capabilities of `ggplot2` by providing additional functions for exploratory data analysis and visualization. It offers tools for creating scatterplot matrices, pairwise plots, and other types of multivariate visualizations. `GGally` is particularly useful for gaining insights into relationships between multiple variables in large datasets.

5) **tidyverse:** `tidyverse` is not a single package but rather a collection of R packages that share a common philosophy and design principles. It includes core packages such as `ggplot2`, `dplyr`, `tidyr`, and others, which are designed to work seamlessly together for data manipulation, visualization, and analysis.

6) **corrplot:** The `corrplot` package is used for visualizing correlation matrices in R. It offers various plotting methods for displaying correlation coefficients, including color-coded correlation matrices, clustered correlation matrices, and circular correlation plots. `corrplot` is helpful for exploring relationships between multiple variables and identifying patterns of correlation in data.

7) **RcolorBrewer**: `RcolorBrewer` provides access to color palettes, which are particularly useful for creating visually appealing and interpretable plots. These palettes offer a wide range of colors that are colorblind-friendly and suitable for both print and on-screen display. `RcolorBrewer` is commonly used in conjunction with `ggplot2` for customizing plots.

<u>Explanation of the experimental procedure and Visualization of the results</u>

Step 01

Install and activate packages.

```
install.packages("party")
install.packages("epitools")
install.packages("ggplot2")
install.packages("GGally")
install.packages("tidyverse")
install.packages("corrplot")
install.packages("RColorBrewer")

library(party)
library(epitools)
library(ggplot2)
library(GGally)
library(tidyverse)
library(corrplot)
library(RColorBrewer)
```

Step 02

Import the data set.

```
> #Import the data set
> data=read.csv("final.csv")
> View(data)
>
```

LR 1.R × data ×

Filter

	Age	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Outcome
1	50	148	72	35	0	33.6	0.627	1
2	31	85	66	29	0	26.6	0.351	0
3	32	183	64	0	0	23.3	0.672	1
4	21	89	66	23	94	28.1	0.167	0
5	33	137	40	35	168	43.1	2.288	1
6	30	116	74	0	0	25.6	0.201	0
7	26	78	50	32	88	31.0	0.248	1
8	29	115	0	0	0	35.3	0.134	0
9	53	197	70	45	543	30.5	0.158	1
10	54	125	96	0	0	0.0	0.232	1
11	30	110	92	0	0	37.6	0.191	0

Showing 1 to 11 of 768 entries, 8 total columns

Step 03

Remove all the NULL values.

```
> #remove NULL values
> data=na.omit(as.data.frame(data))
> data
   Age Glucose BloodPressure SkinThickness Insulin  BMI
1   50     148            72            35       0 33.6
2   31      85            66            29       0 26.6
3   32     183            64             0       0 23.3
4   21      89            66            23      94 28.1
5   33     137            40            35     168 43.1
6   30     116            74             0       0 25.6
7   26      78            50            32      88 31.0
8   29     115             0             0       0 35.3
9   53     197            70            45     543 30.5
10  54     125            96             0       0  0.0
11  30     110            92             0       0 37.6
12  34     168            74             0       0 38.0
13  57     139            80             0       0 27.1
14  59     189            60            23     846 30.1
15  51     166            72            19     175 25.8
16  32     100             0             0       0 30.0
```

Step 04

Get the summary of the dataset.

```
> summary(data)
      Age           Glucose       BloodPressure    SkinThickness  
 Min.   :21.00   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
 1st Qu.:24.00   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00  
 Median :29.00   Median :117.0   Median : 72.00   Median :23.00  
 Mean   :33.24   Mean   :120.9   Mean   : 69.11   Mean   :20.54  
 3rd Qu.:41.00   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00  
 Max.   :81.00   Max.   :199.0   Max.   :122.00   Max.   :99.00  
    Insulin           BMI        DiabetesPedigreeFunction
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780          
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437          
 Median : 30.5   Median :32.00   Median :0.3725          
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719          
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262          
 Max.   :846.0   Max.   :67.10   Max.   :2.4200          
    Outcome     
 Min.   :0.000  
 1st Qu.:0.000  
 Median :0.000  
 Mean   :0.349  
 3rd Qu.:1.000  
 Max.   :1.000  
> 
```

Step 05

Get the first 6 rows of the dataset.

Use of head () function.

```
> head(data)
  Age Glucose BloodPressure SkinThickness Insulin  BMI
1  50     148            72            35       0 33.6
2  31      85            66            29       0 26.6
3  32     183            64             0       0 23.3
4  21      89            66            23      94 28.1
5  33     137            40            35     168 43.1
6  30     116            74             0       0 25.6
  DiabetesPedigreeFunction Outcome
1                    0.627       1
2                    0.351       0
3                    0.672       1
4                    0.167       0
5                    2.288       1
6                    0.201       0
> 
```

Step 06

Get the dimension of the dataset.

```
> dim(data)
[1] 768   8
> 
```

Step 07

Get the structure of the dataset.

```
> str(data)
'data.frame':	768 obs. of  8 variables:
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125
...
 $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3
30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
> 
```

Step 08

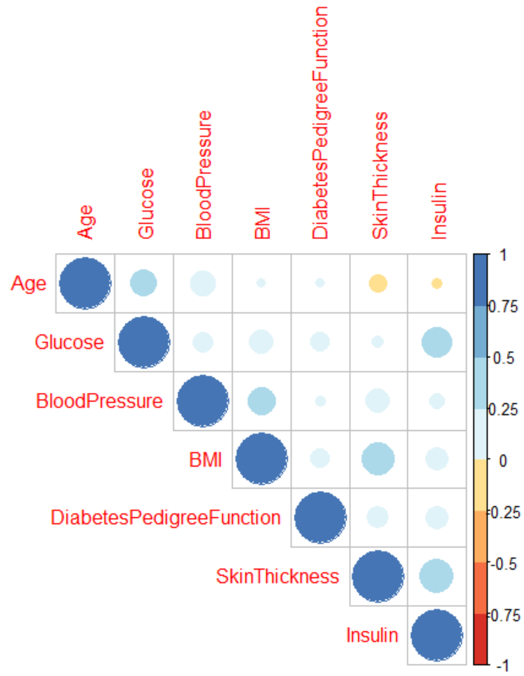
Convert the dependent variable (outcome) to factor and compute the variance of x and the correlation of x and y.

```
> #convert dependent variable (outcome) to factor
> data$Outcome=as.factor(data$Outcome)
> #cor() function compute the variance of x and the covariance or correlation of
x and y if these are vectors. If x and y are matrices then the covariances (or c
orrelations) between the columns of x and the columns of y are computed.
> data_cor=cor(data[,-8])
> data_cor
                                 Age    Glucose BloodPressure
Age                       1.00000000 0.26351432    0.23952795
Glucose                   0.26351432 1.00000000    0.15258959
BloodPressure             0.23952795 0.15258959    1.00000000
SkinThickness            -0.11397026 0.05732789    0.20737054
Insulin                  -0.04216295 0.33135711    0.08893338
BMI                       0.03624187 0.22107107    0.28180529
DiabetesPedigreeFunction  0.03356131 0.13733730    0.04126495
                         SkinThickness     Insulin        BMI
Age                        -0.11397026 -0.04216295 0.03624187
Glucose                     0.05732789  0.33135711 0.22107107
BloodPressure               0.20737054  0.08893338 0.28180529
SkinThickness               1.00000000  0.43678257 0.39257320
Insulin                     0.43678257  1.00000000 0.19785906
BMI                         0.39257320  0.19785906 1.00000000
DiabetesPedigreeFunction    0.18392757  0.18507093 0.14064695
                         DiabetesPedigreeFunction
Age                                    0.03356131
Glucose                                0.13733730
BloodPressure                          0.04126495
SkinThickness                          0.18392757
Insulin                                0.18507093
BMI                                    0.14064695
DiabetesPedigreeFunction               1.00000000
> 
```

Step 09

Visualize the matrix.

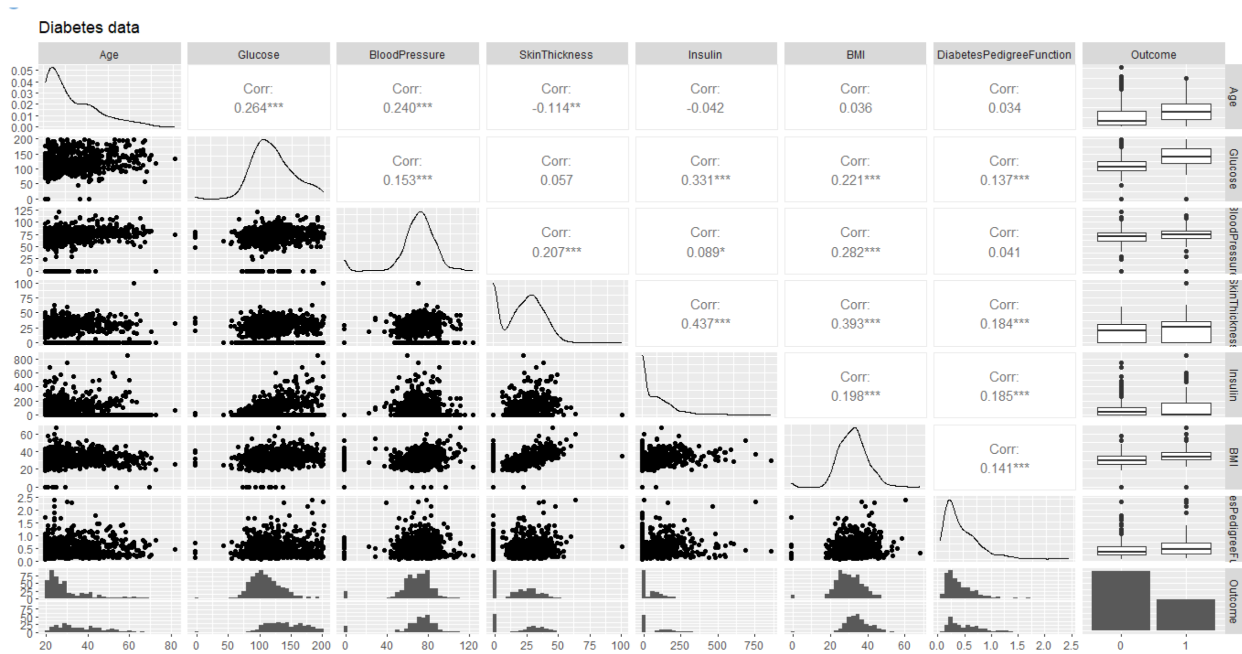
```
> corrplot(data_cor, type="upper", order="hclust", col=brewer.
pal(n=8,name="RdYlBu"))
> 
```



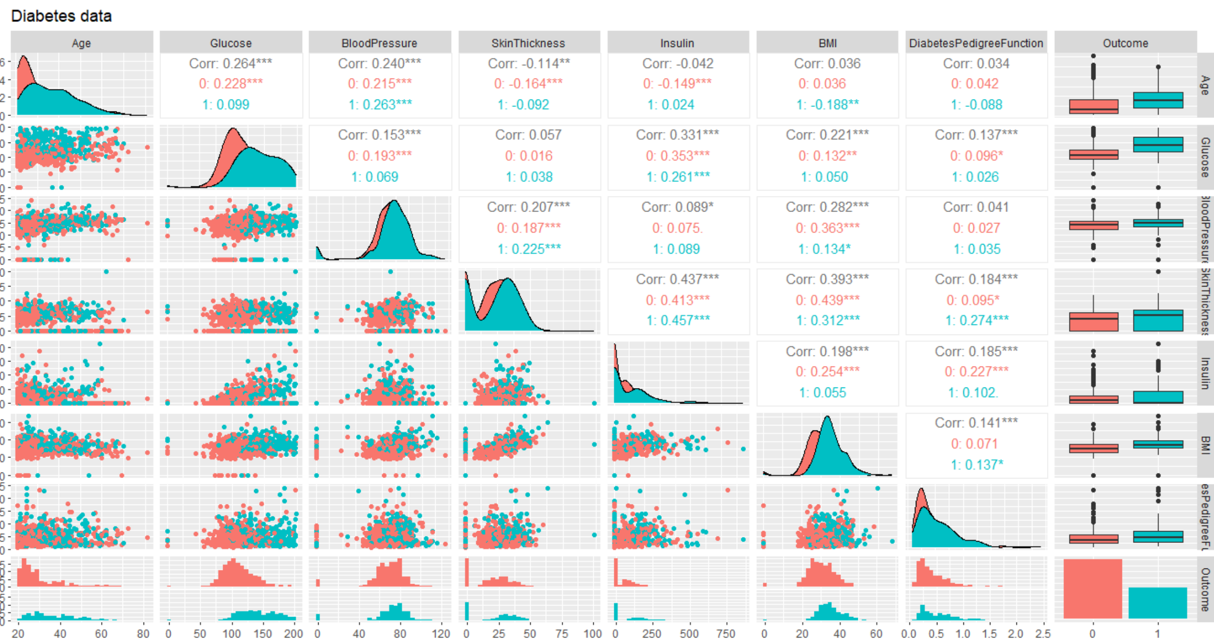
Step 10

Plot the results.

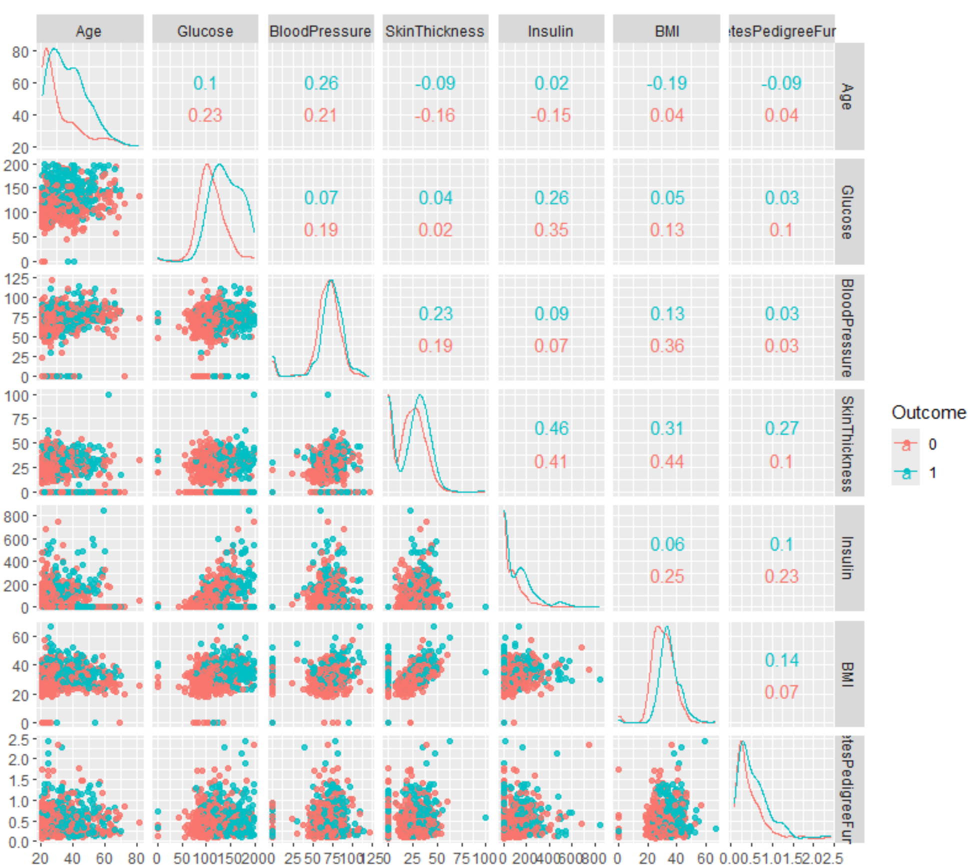
```
> ggpairs(data=data, title="Diabetes data")
```



```
> ggpairs(data=data, mapping = aes(color = Outcome), title="Diabetes data")
```




```
> ggscatmat(data=data, color="Outcome", alpha=0.8)
```



Step 11

Divide the data set sample into 70% training and 30% validation parts.

```
> #Now we will divide our sample into 70% Training and 30% Validation parts.
> pd=sample(2, nrow(data),replace=TRUE, prob=c(0.7,0.30))
> pd
  [1] 1 1 2 2 1 2 1 1 2 2 1 2 2 1 2 1 2 1 2 1 1 1 1 1 2 1 2 1 1 1 2 1 2 1
 [35] 2 2 2 2 1 1 2 2 2 1 1 1 1 2 2 2 1 1 1 2 2 1 1 2 2 1 2 1 1 2 1 1 1 1
 [69] 2 1 1 1 1 1 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 2 1 2 1 2 2 1 2 2 1 1 2
[103] 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 2 1 1 2 1 2 2 2 1 1 1 1 1 1 1 1 2 1
[137] 1 2 2 2 1 2 1 2 2 2 1 1 2 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 2
[171] 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 2 1 1 1 2 2 2 1 1 2 1 1 1 1 2 1
[205] 1 2 1 1 2 1 1 1 2 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 2 1 1 1
[239] 1 1 1 2 2 1 2 2 1 1 1 1 2 1 1 1 2 2 1 1 1 1 2 1 1 2 1 2 2 1 2 1 1 2
[273] 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
[307] 1 1 1 1 2 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 1 1 1 2 1 2 2 1 1 1
[341] 1 1 2 1 1 1 1 1 2 2 1 1 1 2 1 1 2 2 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1
[375] 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 2 2 2 1 2 1 2 2 2 1 1 1 1 2 1
[409] 1 1 2 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 2 2 1 2 2 1 1 2 2 1 1 1 1 1 2
[443] 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2 1 2 1 2 2 2 1 2 1 1 2
[477] 1 1 2 1 2 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2
[511] 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 2 1
[545] 1 2 1 1 1 1 2 1 2 1 1 1 2 1 2 1 1 1 2 1 1 2 2 1 1 1 1 2 2 2 2 1 2 1
[579] 2 2 1 1 2 2 2 2 2 1 1 2 1 1 1 1 2 2 2 1 1 1 2 2 1 2 1 2 1 1 2 1 2 1
[613] 1 2 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 1 2 1 1 1 1 2 1 2
[647] 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 2 1 1 1 1 1 1 1 2 2 1 2 1 1 2 1 1 2 1
[681] 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1
[715] 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1 2 1 2 2 1 2 1
[749] 1 2 1 1 1 1 2 2 2 1 1 2 2 1 1 1 1 1 1 2
> 
```

```
> train=data[pd==1,]
> head(train)
   Age Glucose BloodPressure SkinThickness Insulin  BMI
1   50     148            72            35       0 33.6
2   31      85            66            29       0 26.6
5   33     137            40            35     168 43.1
7   26      78            50            32      88 31.0
8   29     115             0             0       0 35.3
11  30     110            92             0       0 37.6
   DiabetesPedigreeFunction Outcome
1                     0.627       1
2                     0.351       0
5                     2.288       1
7                     0.248       1
8                     0.134       0
11                    0.191       0
> 
```

```
> validate=data[pd==2,]
> head(validate)
   Age Glucose BloodPressure SkinThickness Insulin  BMI
3   32     183            64             0       0 23.3
4   21      89            66            23      94 28.1
6   30     116            74             0       0 25.6
9   53     197            70            45     543 30.5
10  54     125            96             0       0  0.0
12  34     168            74             0       0 38.0
   DiabetesPedigreeFunction Outcome
3                     0.672       1
4                     0.167       0
6                     0.201       0
9                     0.158       1
10                    0.232       1
12                    0.537       1
> 
```

Step 12

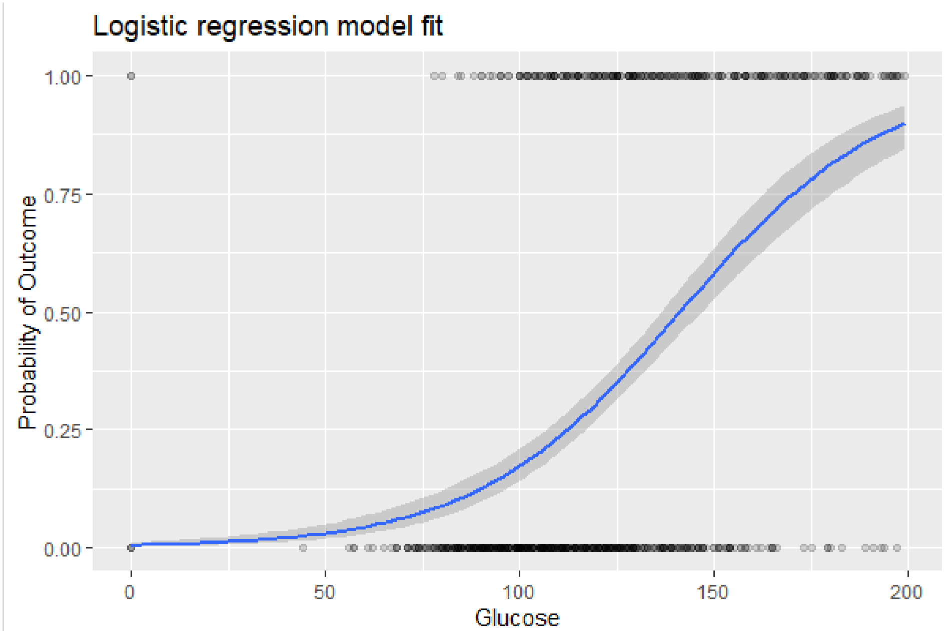
Creating Logistic Regression Models.

Model 01 – Outcome and Glucose

```
> #model 1 -Outcome and Glucose
> model_glm_1=glm(Outcome ~ Glucose, data = train, family = "binomial")
> summary(model_glm_1)

Call:
glm(formula = Outcome ~ Glucose, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3237  -0.7755  -0.5013   0.7733   2.3035

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.994988   0.549646  -10.91   <2e-16 ***
Glucose      0.043782   0.004287   10.21   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 682.11  on 516  degrees of freedom
Residual deviance: 530.81  on 515  degrees of freedom
AIC: 534.81

Number of Fisher Scoring iterations: 4

>
```

Plot the logistic regression model.

```
> #plot the logistic regression model
> data %>%
+   mutate(Out = ifelse(Outcome == "1", 1, 0)) %>%
+   ggplot(aes(Glucose, Out)) +
+   geom_point(alpha = .15) +
+   geom_smooth(method = "glm",method.args = list(family = "binomial")) +
+   ggtitle("Logistic regression model fit") +
+   xlab("Glucose") +
+   ylab("Probability of Outcome")
`geom_smooth()` using formula = 'y ~ x'
>
```



Step 13

Creating logistic regression model predictions.

```
> #Making predictions on the train data set
> trn_pred=ifelse(predict(model_glm_1, type = "response") >0.5, "1", "0")
> trn_tab=table(predicted = trn_pred, actual = train$Outcome)
> trn_tab
         actual
predicted   0   1
        0 282  91
        1  43 101
> 
```

```
> #Model Evaluation
> accuracy_train_1=sum(diag(trn_tab))/sum(trn_tab)
> accuracy_train_1
[1] 0.7408124
> 
```

```
> #Making predictions on the test data set
> tst_pred=ifelse(predict(model_glm_1, newdata = validate, type = "respon
se") > 0.5, "1", "0")
> tst_tab=table(predicted = tst_pred, actual = validate$Outcome)
> tst_tab
         actual
predicted   0   1
        0 142  35
        1  33  41
> 
```

```
> #Model Evaluation
> accuracy_validate_1=sum(diag(tst_tab))/sum(tst_tab)
> accuracy_validate_1
[1] 0.7290837
> 
```

Model 02 – Build a regression model to check whether we can predict a person has outcome for given all the independent variables.

```
> #model 2 -  Lets build a logistic regression model to check whether we
can predict a person has outcome for given  all the independent variable
s.
> model_glm_2=glm(Outcome~ ., data = train, family = "binomial")
> summary(model_glm_2)
```

```
Call:
glm(formula = Outcome ~ ., family = "binomial", data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.7977  -0.6936  -0.3916   0.6783   2.7052  

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)    
(Intercept)              -9.827506   0.947669 -10.370  < 2e-16 ***
Age                       0.039763   0.010634   3.739 0.000185 ***
Glucose                   0.041569   0.004954   8.391  < 2e-16 ***
BloodPressure            -0.009004   0.006850  -1.315 0.188667    
SkinThickness             0.010587   0.008576   1.234 0.217026    
Insulin                  -0.002265   0.001160  -1.952 0.050932 .  
BMI                       0.090422   0.018974   4.766 1.88e-06 ***
DiabetesPedigreeFunction  0.847554   0.367704   2.305 0.021167 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 682.11  on 516  degrees of freedom
Residual deviance: 476.73  on 509  degrees of freedom
AIC: 492.73

Number of Fisher Scoring iterations: 5

> 
```

```
> #We must “manually” convert the probabilities to classifications.
> trn_pred=ifelse(predict(model_glm_2, type = "response") >0.5, "1", "0")
> trn_pred
  1   2   5   7   8  11  14  16  18  20  21  22  23  24  26  28  29 
"1" "0" "1" "0" "0" "0" "1" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" 
 30  32  34  39  40  44  45  46  47  51  52  53  56  57  60  62  63 
"0" "1" "0" "0" "1" "1" "1" "1" "1" "0" "0" "0" "0" "1" "0" "0" "0" 
 65  66  67  68  70  71  72  73  74  75  77  78  79  80  81  83  86 
"0" "0" "0" "1" "0" "0" "0" "1" "0" "0" "0" "0" "1" "0" "0" "0" "0" 
 87  88  89  90  92  94  97 100 101 104 105 106 108 109 110 111 112 
"0" "0" "1" "0" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "1" "1" 
114 115 116 118 120 121 123 127 128 129 130 131 132 133 134 136 137 
"0" "1" "1" "0" "0" "1" "0" "0" "0" "0" "0" "1" "1" "1" "0" "0" "0" 
141 143 147 148 150 151 152 153 155 156 158 159 160 162 163 164 165 
"0" "0" "0" "0" "0" "0" "0" "1" "1" "1" "0" "0" "1" "0" "0" "0" "0" 
166 167 169 171 172 173 174 176 177 178 179 180 184 185 186 187 188 
"0" "1" "0" "0" "0" "0" "0" "1" "0" "1" "1" "1" "0" "0" "1" "1" "1" 
190 191 192 196 197 199 200 201 202 204 205 207 208 210 211 212 214 
"0" "0" "0" "1" "0" "0" "0" "0" "1" "0" "0" "1" "1" "1" "0" "1" "1" 
215 216 217 220 221 222 223 225 226 227 228 229 233 234 236 237 238 
"0" "1" "0" "0" "1" "1" "0" "0" "0" "0" "1" "1" "0" "0" "1" "1" "1" 
```

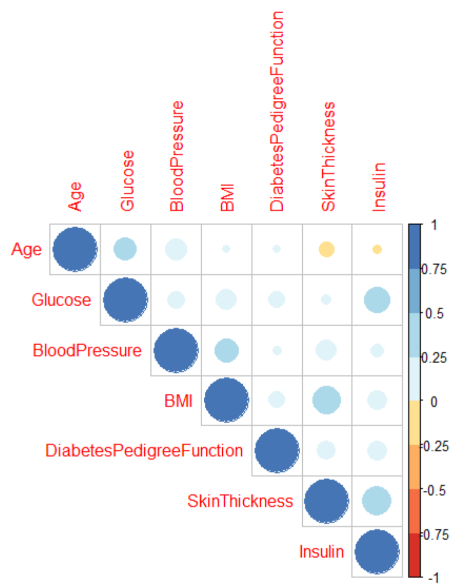
```
> #Making predictions on the train set.
> trn_tab=table(predicted = trn_pred, actual = train$Outcome)
> trn_tab
         actual
predicted   0   1
        0 291  71
        1  34 121
> #Model Evaluation
> accuracy_train_2=sum(diag(trn_tab))/sum(trn_tab)
> accuracy_train_2
[1] 0.7969052
> 
```

```
> #Making predictions on the test data set.
> tst_pred=ifelse(predict(model_glm_2, newdata = validate, type = "respon
se") > 0.5, "1", "0")
> tst_tab=table(predicted = tst_pred, actual = validate$Outcome)
> tst_tab
         actual
predicted   0   1
        0 138  32
        1  37  44
> #Model Evaluation
> accuracy_validate_2=sum(diag(tst_tab))/sum(tst_tab)
> accuracy_validate_2
[1] 0.7250996
> 
```

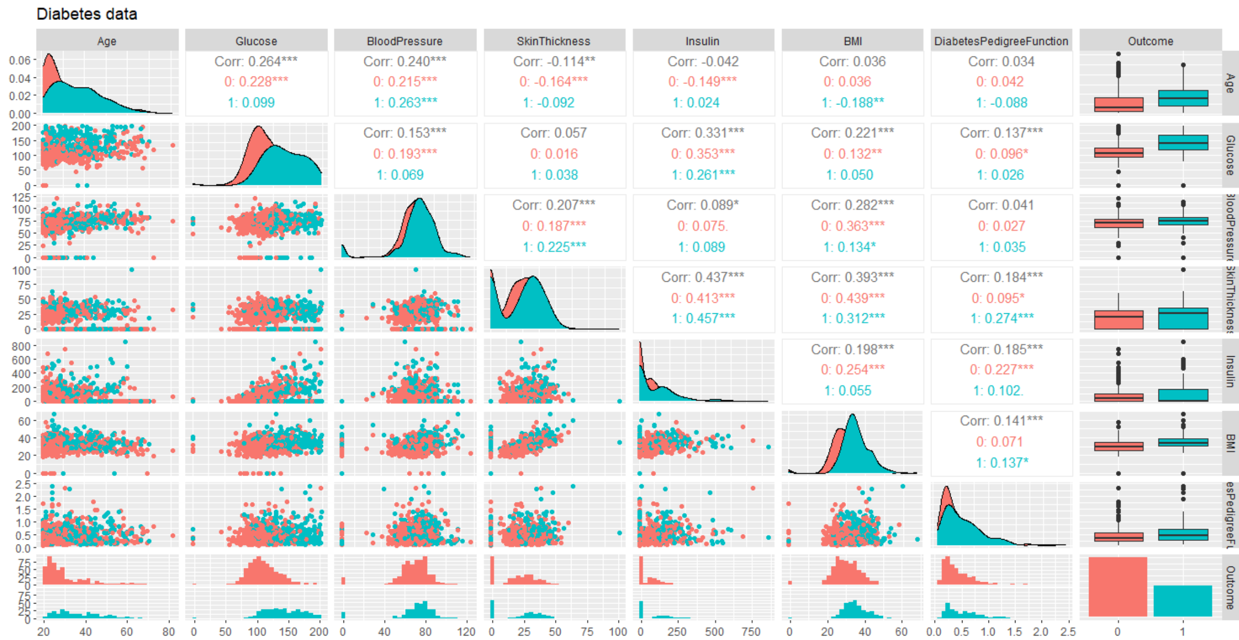
Final result can be shown as follows.

```
> cat("Training set accuracy_1:", accuracy_train_1, "\n")
Training set accuracy_1: 0.7408124
> cat("Validation set accuracy_1:", accuracy_validate_1, "\n")
Validation set accuracy_1: 0.7290837
> cat("Training set accuracy_2:", accuracy_train_2, "\n")
Training set accuracy_2: 0.7969052
> cat("Validation set accuracy_2:", accuracy_validate_2, "\n")
Validation set accuracy_2: 0.7250996
> 
```

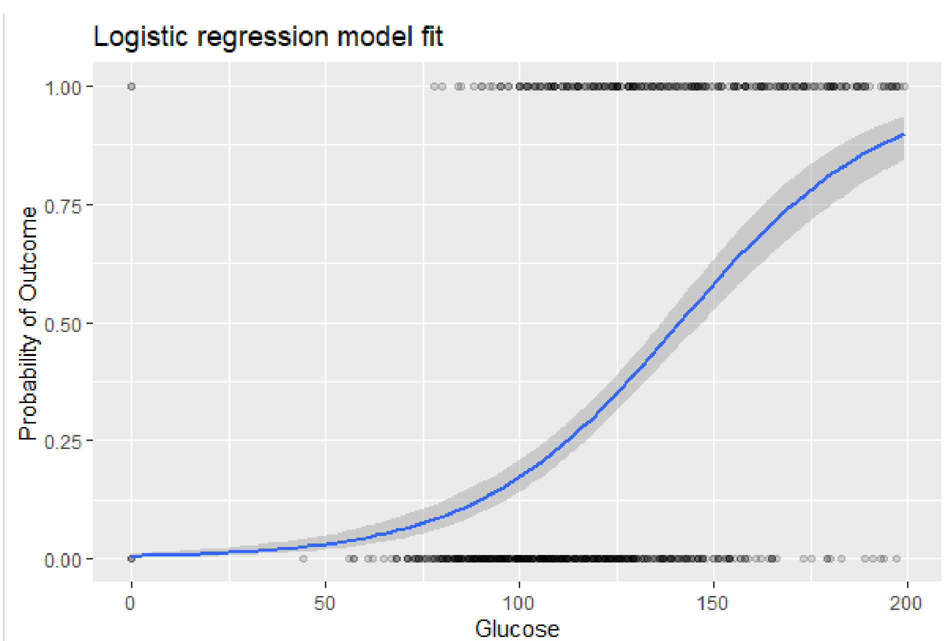
6) Results, Analysis, and Discussions



The above plot shows the correlation between different characteristics in the dataset related to diabetes. The upper left corner of the figure reads "Correlation Matrix". Each cell in the table shows the correlation between two features. For example, the value of 0.284 in the upper left corner represents the correlation between age and glucose. A correlation coefficient close to 1 indicates a strong positive correlation, and a coefficient close to -1 indicates a strong negative correlation. Features are listed on the x and y axis. Features include Age, Glucose, Blood pressure, Body Mass Index (BMI), Diabetes genetics, Skin thickness, and Insulin.



The above scatter plot shows how the different features in the diabetes data set are related to each other. The data includes six features: age, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), and a diabetes pedigree function. The scatter plot shows the correlation between each pair of features. For example, the text "0.284" in the second row, first column represents the correlation between age and glucose. A correlation coefficient closer to 1 indicates a stronger positive correlation, and a coefficient closer to -1 indicates a stronger negative correlation.



The x-axis of the above graph is glucose level, and the y-axis is the probability of the outcome. The curve shows that the probability of the outcome increases as the glucose level increases. For example, at a glucose level of 50, the probability of the outcome is very low. At a glucose level of 200, the probability of the outcome is much higher. This graph can be used to help diagnose a condition or to predict the likelihood of someone developing a condition. For example, a doctor might use a logistic regression model to help diagnose diabetes. The doctor would input a patient's blood glucose level into the model, and the model would output the probability that the patient has diabetes.

```
> cat("Training set accuracy_1:", accuracy_train_1, "\n")
Training set accuracy_1: 0.7408124
> cat("Validation set accuracy_1:", accuracy_validate_1, "\n")
Validation set accuracy_1: 0.7290837
> cat("Training set accuracy_2:", accuracy_train_2, "\n")
Training set accuracy_2: 0.7969052
> cat("Validation set accuracy_2:", accuracy_validate_2, "\n")
Validation set accuracy_2: 0.7250996
> 
```

The above final result shows that the training set accuracy_2 has a higher accuracy (79.69%) compared to training set accuracy_1 (74.08%). However, validation set accuracy_1 (72.91%) is closer to validation set accuracy_2 (72.51%) meaning the model generalizes better on the first dataset.

7) Conclusion

Regression analysis is a statistical method used to investigate and model the relationship between one or more independent variables and a dependent variable There are various types of techniques used for regression in data mining, including linear, multiple and logistic regression each with its strengths and weaknesses. The above regression analysis conducted on the diabetes dataset reveals a significant relationship between glucose levels and the probability of the outcome. Our findings indicate that as glucose levels increase, there is a corresponding increase in the likelihood of the outcome occurring. This suggests that glucose levels play a crucial role in predicting the outcome under consideration. Understanding this relationship is vital for identifying potential risk factors and developing effective interventions for managing diabetes and its associated complications.

8) References

https://data.world/data-society/pima-indians-diabetes-database