#HW3: Association Rules Mining

```
# Enter your name here: Hrishikesh Telang
# SUID: 889489533
```

#1. I am loading the libraries

```
library(arules) #Load the package 'arules'
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz) #Load the package 'arulesViz'
library(readr) #Load the package 'readr'
```

#2. Next, I am loading the csv data file

```
bank <- read.csv('bankdata_csv_all.csv')
```

#3. I am checkling the first five and the last five columns of the bank dataset

```
head(bank)
```

```
##        id age    sex     region  income married children car save_act
## 1 ID12101  48 FEMALE INNER_CITY 17546.0      NO        1  NO       NO
## 2 ID12102  40   MALE       TOWN 30085.1     YES        3 YES       NO
## 3 ID12103  51 FEMALE INNER_CITY 16575.4     YES        0 YES      YES
## 4 ID12104  23 FEMALE       TOWN 20375.4     YES        3  NO       NO
## 5 ID12105  57 FEMALE      RURAL 50576.3     YES        0  NO      YES
## 6 ID12106  57 FEMALE       TOWN 37869.6     YES        2  NO      YES
##   current_act mortgage pep
## 1          NO       NO YES
## 2         YES      YES  NO
## 3         YES       NO  NO
## 4         YES       NO  NO
## 5          NO       NO  NO
## 6         YES       NO YES
```

```
tail(bank)
```

```
##          id age    sex     region   income married children car save_act
## 595 ID12695  59 FEMALE      RURAL 30971.80     YES        3 YES      YES
## 596 ID12696  61 FEMALE INNER_CITY 47025.00      NO        2 YES      YES
## 597 ID12697  30 FEMALE INNER_CITY  9672.25     YES        0 YES      YES
## 598 ID12698  31 FEMALE       TOWN 15976.30     YES          0 YES      YES
## 599 ID12699  29   MALE INNER_CITY 14711.80     YES          0  NO      YES
## 600 ID12700  38   MALE       TOWN 26671.60      NO          0 YES       NO
##     current_act mortgage pep
## 595         YES      YES  NO
## 596         YES      YES  NO
## 597         YES       NO  NO
## 598          NO       NO YES
## 599          NO      YES  NO
## 600         YES      YES YES
```

#4. Next I am checking the structure of the dataset. I can see that all columns have character datatype, age, income and children are integers and they have quartiles, mean, median and mode.

```
summary(bank) # What is the structure?
```

```
##       id                  age            sex                region
##  Length:600          Min.   :18.00   Length:600         Length:600
##  Class :character   1st Qu.:30.00   Class :character   Class :character
##  Mode  :character   Median :42.00   Mode  :character   Mode  :character
##                      Mean   :42.40
##                      3rd Qu.:55.25
##                      Max.   :67.00
##      income        married             children         car
##  Min.   : 5014   Length:600         Min.   :0.000   Length:600
##  1st Qu.:17264   Class :character   1st Qu.:0.000   Class :character
##  Median :24925   Mode  :character   Median :1.000   Mode  :character
##  Mean   :27524                      Mean   :1.012
##  3rd Qu.:36173                      3rd Qu.:2.000
##  Max.   :63130                      Max.   :3.000
##    save_act         current_act          mortgage             pep
##  Length:600         Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

#5. I am now checking the structure of the bank dataset (to check the datatypes)

```
str(bank) #returns the structure (datatypes) of the bank dataset
```

```
## ’data.frame’:    600 obs. of  12 variables:
##  $ id         : chr  "ID12101" "ID12102" "ID12103" "ID12104" ...
##  $ age        : int  48 40 51 23 57 57 22 58 37 54 ...
##  $ sex        : chr  "FEMALE" "MALE" "FEMALE" "FEMALE" ...
##  $ region     : chr  "INNER_CITY" "TOWN" "INNER_CITY" "TOWN" ...
##  $ income     : num  17546 30085 16575 20375 50576 ...
##  $ married    : chr  "NO" "YES" "YES" "YES" ...
##  $ children   : int  1 3 0 3 0 2 0 0 2 2 ...
##  $ car        : chr  "NO" "YES" "YES" "NO" ...
##  $ save_act   : chr  "NO" "NO" "YES" "NO" ...
##  $ current_act: chr  "NO" "YES" "YES" "YES" ...
##  $ mortgage   : chr  "NO" "YES" "NO" "NO" ...
##  $ pep        : chr  "YES" "NO" "NO" "NO" ...
```

#5. With nrow and ncol functions, I know that my dataframe is 600x12

```
nrow(bank) #returns number of rows
```

```
## [1] 600
```

```
ncol(bank) #returns number of columns
```

```
## [1] 12
```

#6. I wanted to view the whole dataframe.

```
View(bank)
```

#7. Now, I am focusing on converting all the variables to factor variables. (optional step)

```
bank_new <- data.frame(sex=as.factor(bank$sex),
                      region=as.factor(bank$region),
                      married=as.factor(bank$married),
                      children=as.factor(bank$children),
                      car=as.factor(bank$car),
                      save_act=as.factor(bank$save_act),
                      current_act=as.factor(bank$current_act),
                      mortgage=as.factor(bank$mortgage),
                      pep=as.factor(bank$pep))
```

#8. I wanted to know how many customers bought the personal equity plan as agaisnt those who didn't

```
table(bank_new$pep)
```

```
##
##  NO YES
## 326 274
```

#9. I am checking the percentages of the yes variables from the no.

```
prop.table(table(bank_new$pep))
```

```
##
##        NO       YES
## 0.5433333 0.4566667
```

#10. I am coercing the **bank_new** dataframe into a **sparse transactions matrix** called **bankX**.

```
bankX <- as(bank_new, "transactions")
bankX
```

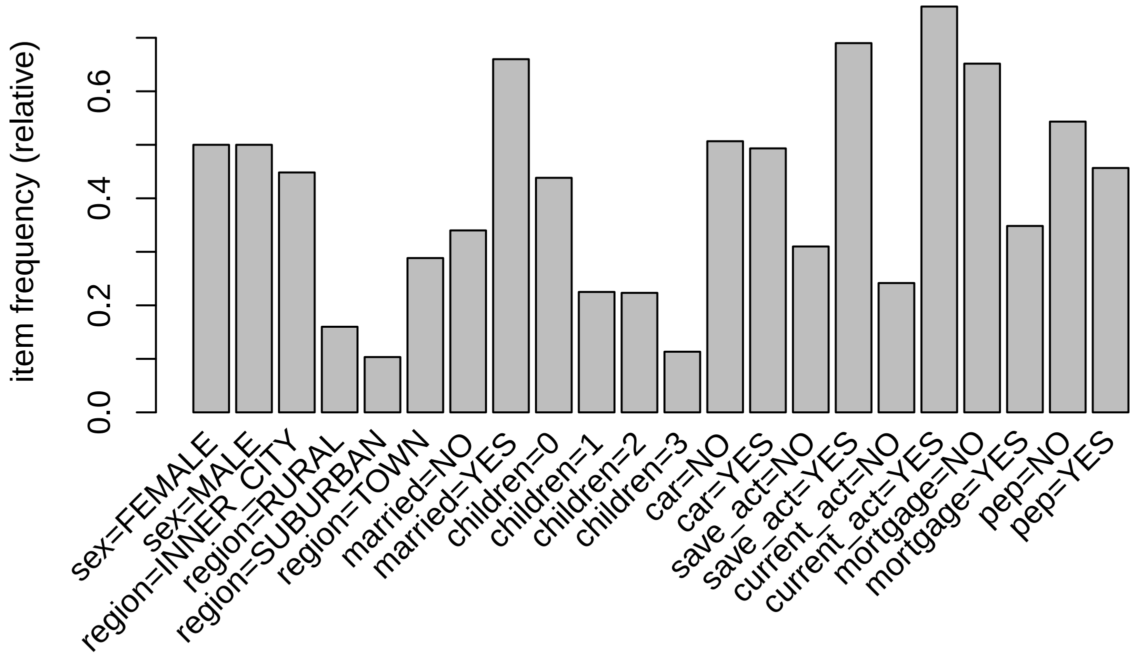
```
## transactions in sparse format with
##  600 transactions (rows) and
##  22 items (columns)
```

#11. I check the item frequency of the bankX matrix

```
itemFrequency(bankX) #Shows item frequency for each categorical value
```

```
##        sex=FEMALE          sex=MALE region=INNER_CITY      region=RURAL
##         0.5000000         0.5000000         0.4483333         0.1600000
##   region=SUBURBAN       region=TOWN        married=NO       married=YES
##         0.1033333         0.2883333         0.3400000         0.6600000
##        children=0        children=1        children=2        children=3
##         0.4383333         0.2250000         0.2233333         0.1133333
##            car=NO           car=YES       save_act=NO      save_act=YES
##         0.5066667         0.4933333         0.3100000         0.6900000
##     current_act=NO   current_act=YES       mortgage=NO      mortgage=YES
##         0.2416667         0.7583333         0.6516667         0.3483333
##            pep=NO           pep=YES
##         0.5433333         0.4566667
```

```
itemFrequencyPlot(bankX) #Plots the frequency distribution
```



#12. I am checking the bankX matrix

```
inspect(bankX[1:10])
```

```
##      items                transactionID
## [1]  {sex=FEMALE,
##       region=INNER_CITY,
##       married=NO,
##       children=1,
##       car=NO,
##       save_act=NO,
##       current_act=NO,
##       mortgage=NO,
##       pep=YES}                       1
## [2]  {sex=MALE,
##       region=TOWN,
##       married=YES,
##       children=3,
##       car=YES,
##       save_act=NO,
##       current_act=YES,
##       mortgage=YES,
##       pep=NO}                        2
## [3]  {sex=FEMALE,
##       region=INNER_CITY,
##       married=YES,
##       children=0,
##       car=YES,
##       save_act=YES,
##       current_act=YES,
##       mortgage=NO,
##       pep=NO}                        3
## [4]  {sex=FEMALE,
##       region=TOWN,
##       married=YES,
##       children=3,
##       car=NO,
##       save_act=NO,
##       current_act=YES,
##       mortgage=NO,
##       pep=NO}                        4
## [5]  {sex=FEMALE,
##       region=RURAL,
##       married=YES,
##       children=0,
##       car=NO,
##       save_act=YES,
##       current_act=NO,
##       mortgage=NO,
##       pep=NO}                        5
## [6]  {sex=FEMALE,
##       region=TOWN,
##       married=YES,
##       children=2,
##       car=NO,
##        save_act=YES,
##        current_act=YES,
##        mortgage=NO,
##        pep=YES}                       6
## [7]  {sex=MALE,
##        region=RURAL,
##        married=NO,
##        children=0,
##        car=NO,
##        save_act=NO,
##        current_act=YES,
##        mortgage=NO,
##        pep=YES}                       7
## [8]  {sex=MALE,
##        region=TOWN,
##        married=YES,
##        children=0,
##        car=YES,
##        save_act=YES,
##        current_act=YES,
##        mortgage=NO,
##        pep=NO}                        8
## [9]  {sex=FEMALE,
##        region=SUBURBAN,
##        married=YES,
##        children=2,
##        car=YES,
##        save_act=NO,
##        current_act=NO,
##        mortgage=NO,
##        pep=NO}                        9
## [10] {sex=MALE,
##        region=TOWN,
##        married=YES,
##        children=2,
##        car=YES,
##        save_act=YES,
##        current_act=YES,
##        mortgage=NO,
##        pep=NO}                        10
```

#13. I finally used **apriori** to generate a set of rules with support over 0.008 and confidence over 0.98, and trying to predict what external situations made customers sign up for a Persoal Equity Plan (PEP). I sortef the dataset in descending order of importance and only wanted to check the top 5 rules.

```
rules <- apriori(bank_new, parameter = list(supp=0.008, conf= 0.98),
                 appearance=list(default="lhs", rhs="pep=YES"),
                 control=list(verbose=F))
rules <- sort(rules, decreasing=TRUE,by='support')
inspect(rules[1:7])
```

```
##     lhs                     rhs          support confidence   coverage     lift count
## [1] {region=TOWN,
##      married=YES,
##      children=1,
##      current_act=YES}    => {pep=YES} 0.03500000          1 0.03500000 2.189781    21
## [2] {region=INNER_CITY,
##      children=0,
##      save_act=NO,
##      mortgage=YES}       => {pep=YES} 0.03166667          1 0.03166667 2.189781    19
## [3] {sex=MALE,
##      married=NO,
##      children=0,
##      save_act=YES,
##      mortgage=NO}        => {pep=YES} 0.03000000          1 0.03000000 2.189781    18
## [4] {region=TOWN,
##      married=YES,
##      children=1,
##      save_act=YES,
##      current_act=YES}    => {pep=YES} 0.02833333          1 0.02833333 2.189781    17
## [5] {region=TOWN,
##      children=1,
##      save_act=YES,
##      mortgage=NO}        => {pep=YES} 0.02666667          1 0.02666667 2.189781    16
## [6] {married=NO,
##      children=0,
##      car=NO,
##      save_act=YES,
##      mortgage=NO}        => {pep=YES} 0.02666667          1 0.02666667 2.189781    16
## [7] {region=TOWN,
##      married=YES,
##      children=1,
##      mortgage=NO}        => {pep=YES} 0.02500000          1 0.02500000 2.189781    15
```

#Inferences: The chances of a customer buying a PEP overall is 45.6% high. The chances that customers who lived in towns and inner cities and who had no savings account had a higher probability of buying PEP, with support being nearly 0.0300. Besides, if the customer owns a current account, has a mortgage and lives in the suburban areas, the chances increase to a confidence to about 100%. It is surprising to note that even the probability that a customer is not married maintains a support of 83.3%

#14. I used the same code to find out what conditions did not let customers buy PEP.

```
rules2 <- apriori(bank_new, parameter = list(supp=0.008, conf= 0.98),
                  appearance=list(default="lhs", rhs="pep=NO"),
                  control=list(verbose=F))
rules2 <- sort(rules2, decreasing=TRUE,by='support')
inspect(rules2[1:5])
```

```
##     lhs                      rhs        support confidence   coverage     lift count
## [1] {region=TOWN,
##      married=YES,
##      children=0,
##      save_act=YES,
##      current_act=YES}  => {pep=NO} 0.04333333          1 0.04333333 1.840491    26
## [2] {married=NO,
##      children=0,
##      save_act=YES,
##      mortgage=YES}     => {pep=NO} 0.03833333          1 0.03833333 1.840491    23
## [3] {children=3,
##      save_act=NO}      => {pep=NO} 0.03666667          1 0.03666667 1.840491    22
## [4] {region=TOWN,
##      married=YES,
##      children=0,
##      car=NO,
##      save_act=YES}     => {pep=NO} 0.03333333          1 0.03333333 1.840491    20
## [5] {married=NO,
##      children=0,
##      save_act=YES,
##      current_act=YES,
##      mortgage=YES}     => {pep=NO} 0.03333333          1 0.03333333 1.840491    20
```

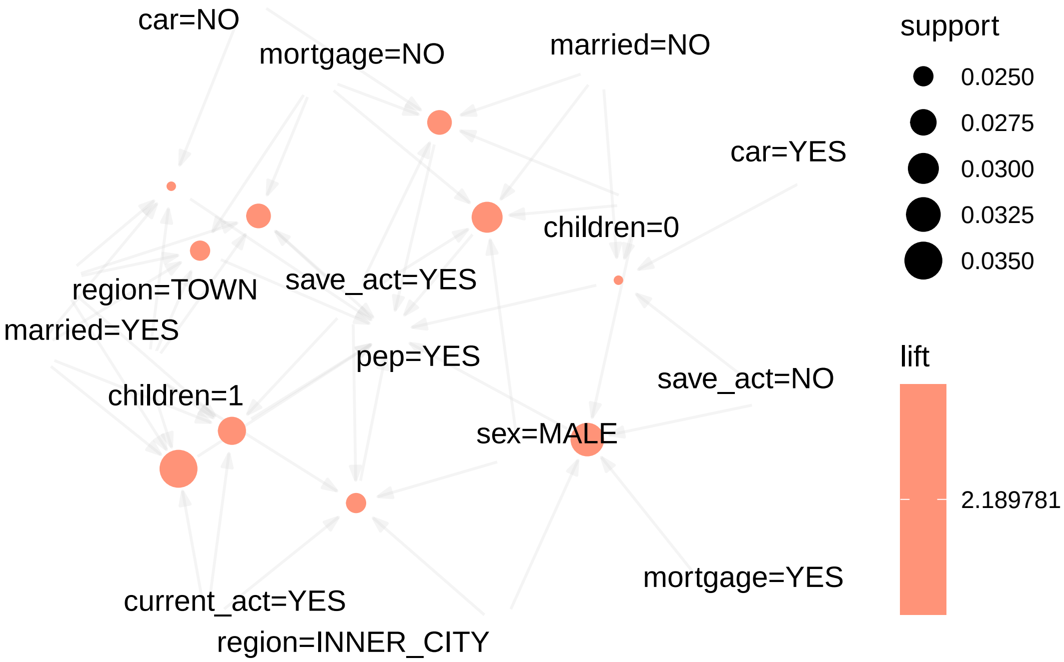
#Inferences: The chances of a customer not buying a PEP overall is 54.3% high. This percentage is found to be greater. The chances that customers who lived in inner cities and had 3 children with no savings account had a higher probability of not buying PEP, with support being nearly 0.0360. Whereas, some families which are married and have both savings and current accounts don't avail PEP. On the other hand, single customers who have both accounts and a mortagage do not avail PEP.

#15. Get the top 10 rules sorted by lift

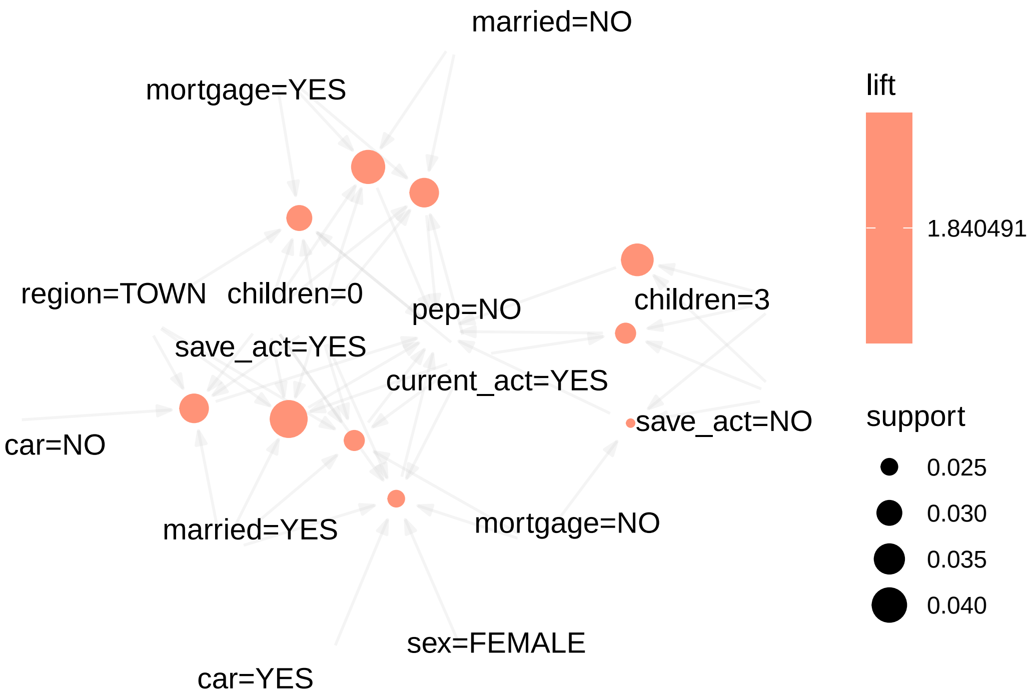
```
subrules <- head(sort(rules, by="lift"), 10)
subrules2 <- head(sort(rules2, by="lift"), 10)
```

#16. I am plotting 'subrules' and 'subrules2'

```
plot(subrules, method="graph")
```



```
plot(subrules2, method="graph")
```



#Strategy: #The PEP should be pitched to individuals who have children and who also don't have savings accounts. The PEP scheme should be readily available in the rural towns.