IST 707 Homework 3

Due Date: 4/21/2021

Introduction

A personal equity plan (PEP) product offering was introduced by a financial firm and it was marketed by sending out a direct mail piece to customers. Data was collected about customers and whether or not they purchased the PEP, along with demographic information about the customer.

There might be patterns that indicate which customers are more likely than others to purchase the PEP depending on their characteristics. The following report goes through the data and applies data mining to try and find these patterns. The goal is to find interesting association rules, and based on the discoveries, provide a business recommendation.

Data Preparation

Ensure packages are installed and active.

```
if ("readr" %in% installed.packages()) {require(readr)} else {install.packages('readr'); req
uire(readr)}

if ("dplyr" %in% installed.packages()) {require(dplyr)} else {install.packages('dplyr'); req
uire(dplyr)}

if ("reshape2" %in% installed.packages()) {require(reshape2)} else {install.packages('reshape2');
require(reshape2)}

if ("sqldf" %in% installed.packages()) {require(sqldf)} else {install.packages('sqldf'); requ
ire(sqldf)}

if ("stringr" %in% installed.packages()) {require(stringr)} else {install.packages('stringr'); req
uire(stringr)}

if ("reticulate" %in% installed.packages()) {require(reticulate)} else {install.packages('reticulate'); r
equire(reticulate)}

if ("arules" %in% installed.packages()) {require(arules)} else {install.packages('arules'); req
uire(arules)}

if ("ggplot2" %in% installed.packages()) {require(ggplot2)} else {install.packages('ggplot2');
require(ggplot2)}
```

Read the data and store it.

```
# make sure that the data-storyteller.csv file is in the current working directory

bank <- as.data.frame(read_csv("bankdata_csv_all.csv", col_types = cols()))
```

Show a sample of the 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
```

Remove the ID field.

```
# double check the ID field is unique and if yes then delete.

bank <- if (length(unique(bank$id)) == dim(bank) [1]) {   #logical test - is every id unique?
        bank <- bank[,-which(colnames(bank) == 'id')] }   #condition if true - remove the column

bankcopy <- bank
```

Discretize the age variable.

```
# using equal width approach with 5 x 10 bins

`age_18-27` <- c()
`age_28-37` <- c()
`age_38-47` <- c()
`age_48-57` <- c()
`age_58-67` <- c()

# populate the categories accordingly
```

```
for (x in bank$age) {

if (between(x, 18, 27)) {`temp_18-27` <- 1} else {`temp_18-27` <- 0}
if (between(x, 28, 37)) {`temp_28-37` <- 1} else {`temp_28-37` <- 0}
if (between(x, 38, 47)) {`temp_38-47` <- 1} else {`temp_38-47` <- 0}
if (between(x, 48, 57)) {`temp_48-57` <- 1} else {`temp_48-57` <- 0}
if (between(x, 58, 67)) {`temp_58-67` <- 1} else {`temp_58-67` <- 0}

`age_18-27` <- c(`age_18-27`, `temp_18-27`)
`age_28-37` <- c(`age_28-37`, `temp_28-37`)
`age_38-47` <- c(`age_38-47`, `temp_38-47`)
`age_48-57` <- c(`age_48-57`, `temp_48-57`)
`age_58-67` <- c(`age_58-67`, `temp_58-67`)
}

# append the categories to dataframe

bank$`age_18-27` <- `age_18-27`
bank$`age_28-37` <- `age_28-37`
bank$`age_38-47` <- `age_38-47`
bank$`age_48-57` <- `age_48-57`
bank$`age_58-67` <- `age_58-67`

# remove the original age variable

bank <- bank[,-which(colnames(bank) == 'age')]

# clean up the environment

rm(`age_18-27`, `age_28-37`, `age_38-47`, `age_48-57`, `age_58-67`,
   `temp_18-27`, `temp_28-37`, `temp_38-47`, `temp_48-57`, `temp_58-67`, x)
```

Discretize the income variable.

```
# determine splits using equal frequency approach

incomebrackets <- discretize(bank$income, method = "frequency", breaks = 3, onlycuts = TRUE)
incomebrackets

## [1]  5014.21 20253.80 31132.77 63130.10

incomesplit1 <- incomebrackets[2]
incomesplit2 <- incomebrackets[3]

# three income brackets - low, medium, high

paste('low income <= $', round(incomesplit1,2))

## [1] "low income <= $ 20253.8"
```

```r
paste('$', round(incomesplit1,2), '< middle income <= $', round(incomesplit2,2))

## [1] "$ 20253.8 < middle income <= $ 31132.77"

paste('high income > $', round(incomesplit2,2))

## [1] "high income > $ 31132.77"

bank$lowIncome <- ifelse(bank$income <= incomesplit1, 1, 0)
bank$middleIncome <- ifelse(bank$income > incomesplit1 & bank$income <= incomesplit2, 1, 0)
bank$highIncome <- ifelse(bank$income > incomesplit2, 1, 0)

# check that the discretization is correct

cat('count of low income = ', sum(bank$lowIncome),'\n',
    'count of middle income = ', sum(bank$middleIncome), '\n',
    'count of high income = ', sum(bank$highIncome))

## count of low income =  200
##  count of middle income =  200
##  count of high income =  200

# remove the original income variable

bank <- bank[,-which(colnames(bank) == 'income')]

# clean up the environment

rm(incomebrackets, incomesplit1, incomesplit2)
```

Discretize the rest of the variables.

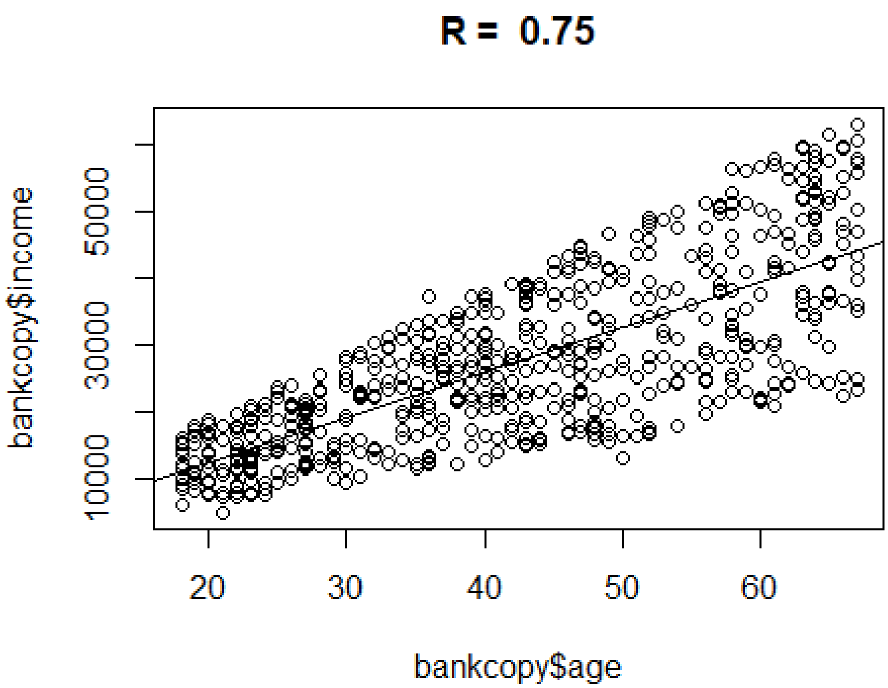
```r
# create a function for discretizing the rest

discretizebankdata <- function(dataframe, columnstodiscretize) {

  tempdf <- dataframe

  for (column in columnstodiscretize) {

    .GlobalEnv$tempcolname <- as.character(column)

    .GlobalEnv$columnvalues <- tempdf[,which(colnames(tempdf) == column)]

    valuenames <- unique(columnvalues)

      for (valuename in valuenames) {

      discretizedvalue <- ifelse(.GlobalEnv$columnvalues == valuename, 1, 0)
```

```
    tempdf <- cbind(tempdf, discretizedvalue)

    colnames(tempdf) [dim(tempdf) [2]] <- paste0(.GlobalEnv$tempcolname, as.character(valuename))

    }
  }

 return(tempdf)

}

bank <- discretizebankdata(bank, colnames(bank) [1:9])

# remove the original variables

bank <- bank[,-c(1:9)]

# clean up the environment

rm(columnvalues, tempcolname, discretizebankdata)
```

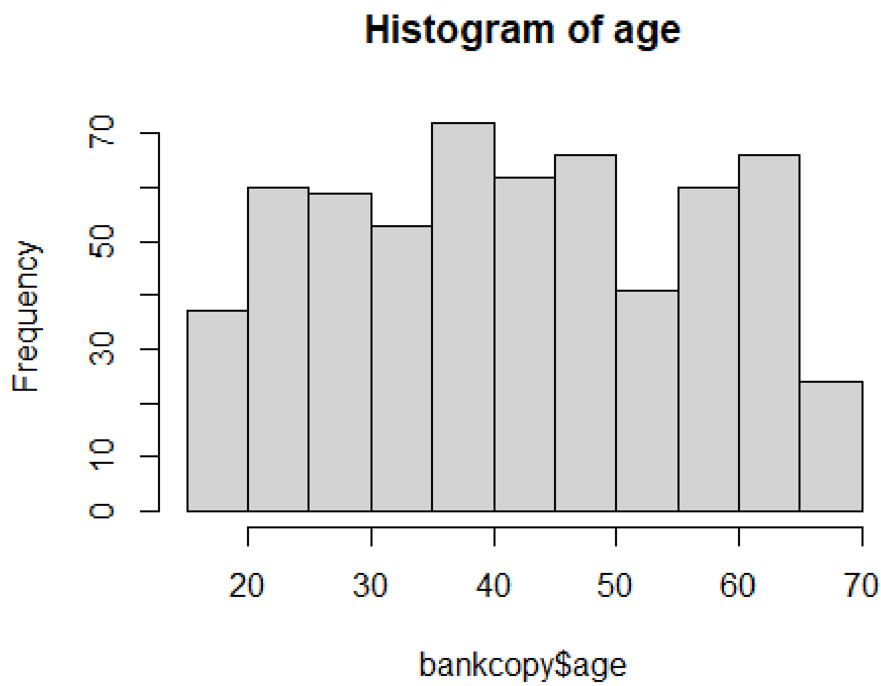
Initial Data Exploration

```
# summarize the data.

summary(bankcopy)

##      age            sex               region              income
##  Min.   :18.00   Length:600         Length:600         Min.   : 5014
##  1st Qu.:30.00   Class :character   Class :character   1st Qu.:17265
##  Median :42.00   Mode  :character   Mode  :character   Median :24925
##  Mean   :42.40                                         Mean   :27524
##  3rd Qu.:55.25                                         3rd Qu.:36173
##  Max.   :67.00                                         Max.   :63130
##    married             children         car              save_act
##  Length:600         Min.   :0.000   Length:600         Length:600
##  Class :character   1st Qu.:0.000   Class :character   Class :character
##  Mode  :character   Median :1.000   Mode  :character   Mode  :character
##                     Mean   :1.012
##                     3rd Qu.:2.000
##                     Max.   :3.000
##  current_act          mortgage             pep
##  Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

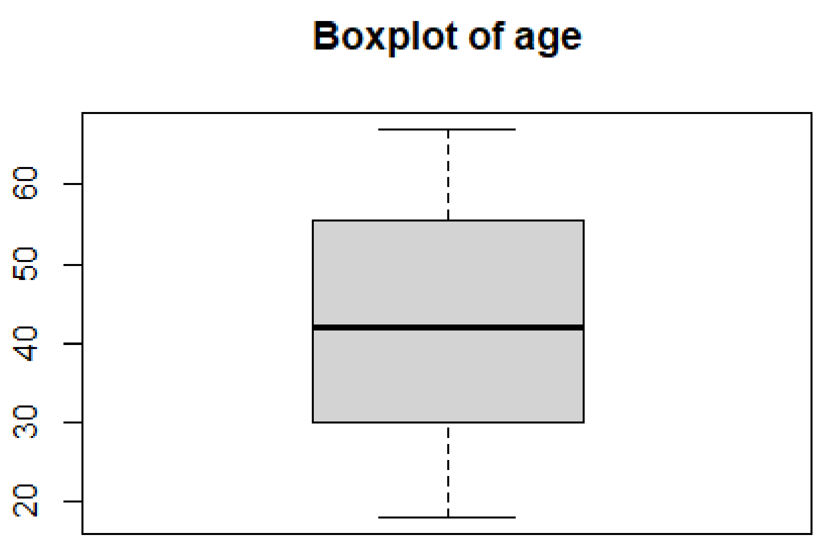
```r
# correlation of age and income.

plot(x = bankcopy$age, y = bankcopy$income,
  main = paste('R = ', round(cor(bankcopy$age, bankcopy$income),2)))
abline(lm(income ~ age, data = bankcopy))
```



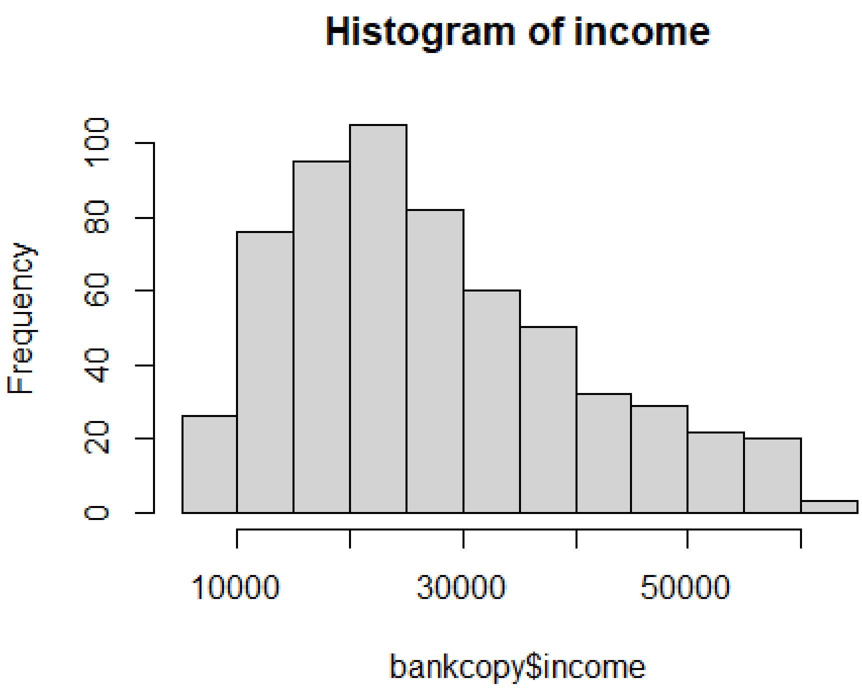

```r
# summarize data for the age variable.

summary(bankcopy$age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   30.00   42.00   42.40   55.25   67.00

hist(bankcopy$age, main = "Histogram of age")
```



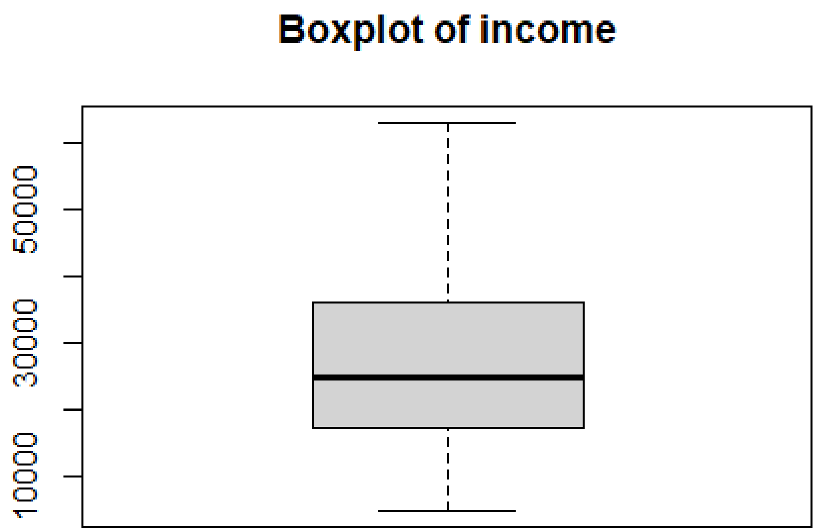

```
boxplot(bankcopy$age, main = "Boxplot of age")
```



```
# summarize data for the income variable.

summary(bankcopy$income)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5014   17265   24925   27524   36173   63130

hist(bankcopy$income, main = "Histogram of income")
```




```
boxplot(bankcopy$income, main = "Boxplot of income")
```




```
# average income grouped by variables.

incomematrix <- data.frame()

for (column in colnames(bankcopy)[which( ! colnames(bankcopy) %in% c('age', 'income'))]) {
tempquery         <- paste0('select ', column, ' as grouped, income from bankcopy')
tempdata          <- as.data.frame(sqldf(tempquery))
tempdata$measure   <- column
incomematrix       <- rbind(incomematrix, tempdata)}

incomematrix <- as.data.frame(sqldf(

'select measure, grouped, avgIncome, recordCount, round(recordCount / 600, 2) as pctTotal from (
select measure, grouped, cast(avg(income) as int) as avgIncome, cast(count(measure || grouped) as float)
as recordCount from
incomematrix group by measure, grouped order by measure, grouped asc) sub'))

incomematrix

##       measure    grouped avgIncome recordCount pctTotal
## 1         car        NO     26486         304     0.51
## 2         car       YES     28589         296     0.49
## 3    children         0     27063         263     0.44
## 4    children         1     27305         135     0.23
## 5    children         2     28435         134     0.22
## 6    children         3     27942          68     0.11
## 7  current_act        NO     26802         145     0.24
```

```
## 8  current_act       YES     27754        455     0.76
## 9      married        NO     27674        204     0.34
## 10     married       YES     27446        396     0.66
## 11    mortgage        NO     27662        391     0.65
## 12    mortgage       YES     27265        209     0.35
## 13         pep        NO     24900        326     0.54
## 14         pep       YES     30644        274     0.46
## 15      region INNER_CITY     26843        269     0.45
## 16      region      RURAL     30027         96     0.16
## 17      region   SUBURBAN     28656         62     0.10
## 18      region       TOWN     26786        173     0.29
## 19    save_act        NO     22405        186     0.31
## 20    save_act       YES     29823        414     0.69
## 21         sex    FEMALE     27831        300     0.50
## 22         sex      MALE     27216        300     0.50

#clean up the environment

rm(tempdata, column, tempquery)
```

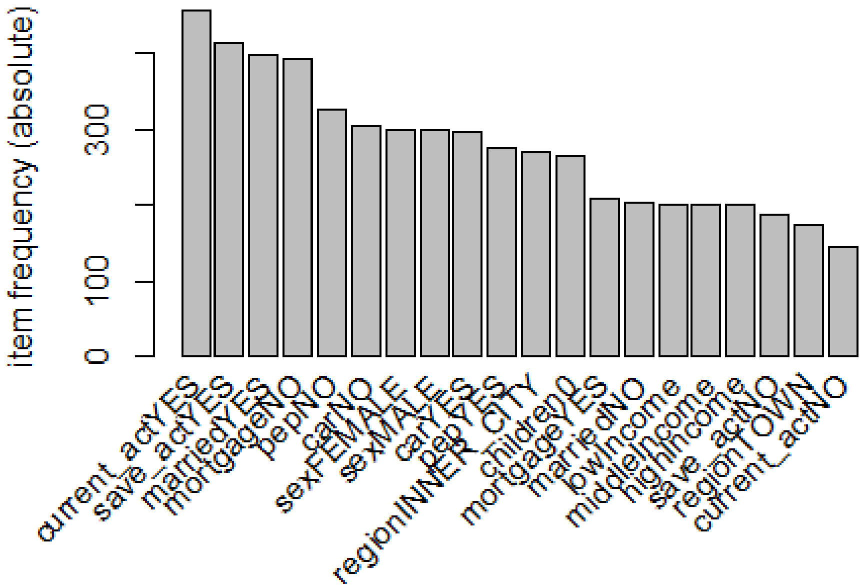
Observations:

- there are a near equal amount of customers in each age group
- the income variable is normally distributed but right skewed
- there is a strong positive correlation with age and income
- males and females are an even 50/50 split in the data
- 46% of customers purchased the PEP and 54% did not
- avg income is $6,000 higher for those who purchased the PEP

Data Analysis

Read transactions and generate rules.

```
# read the transactions

banktransactions <- as(as.matrix(bank), "transactions")

# item frequency plot

itemFrequencyPlot(banktransactions, topN=20,type="absolute")
```




```
# generate rules

#bankrules <- apriori(banktransactions, parameter = list(supp = 0.5, conf = 0.8)) #attempt 1 - did not pr
oduce any rules
#bankrules <- apriori(banktransactions, parameter = list(supp = 0.01, conf = 0.8)) #attempt 2 - produce
d too many rules
bankrules <- apriori(banktransactions, parameter = list(supp = 0.15, conf = 0.8)) #attempt 3 - produced 31
rules, all strong

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.15      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: 90
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[30 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
```

```
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [31 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# top 5 rules sorted by highest support

inspect((sort(bankrules, decreasing = TRUE, by = "support"))[1:5])
```

```
##     lhs                            rhs               support   confidence
## [1] {highIncome}                => {save_actYES}    0.2850000 0.8550000
## [2] {mortgageNO,pepNO}          => {marriedYES}     0.2850000 0.8181818
## [3] {carNO,mortgageNO}          => {current_actYES} 0.2633333 0.8020305
## [4] {children0,pepNO}           => {marriedYES}     0.2350000 0.8443114
## [5] {highIncome,current_actYES} => {save_actYES}    0.2233333 0.8589744
##     coverage  lift     count
## [1] 0.3333333 1.239130 171
## [2] 0.3483333 1.239669 171
## [3] 0.3283333 1.057623 158
## [4] 0.2783333 1.279260 141
## [5] 0.2600000 1.244890 134
```

```r
# top 5 rules sorted by highest confidence

inspect((sort(bankrules, decreasing = TRUE, by = "confidence"))[1:5])
```

```
##     lhs                                  rhs             support   confidence
## [1] {children0,mortgageNO,pepNO}       => {marriedYES}  0.1733333 0.9719626
## [2] {marriedYES,children0,save_actYES} => {pepNO}       0.1783333 0.8991597
## [3] {marriedYES,children0,mortgageNO}  => {pepNO}       0.1733333 0.8965517
## [4] {age_18-27}                        => {lowIncome}   0.1850000 0.8809524
## [5] {highIncome,marriedYES}            => {save_actYES} 0.1866667 0.8750000
##     coverage  lift     count
## [1] 0.1783333 1.472671 104
## [2] 0.1983333 1.654895 107
## [3] 0.1933333 1.650095 104
## [4] 0.2100000 2.642857 111
## [5] 0.2133333 1.268116 112
```

```r
# top 5 rules sorted by highest lift

inspect((sort(bankrules, decreasing = TRUE, by = "lift"))[1:5])
```

```
##     lhs                                  rhs            support   confidence
## [1] {age_18-27}                        => {lowIncome}  0.1850000 0.8809524
## [2] {children1}                        => {pepYES}     0.1833333 0.8148148
## [3] {marriedYES,children0,save_actYES} => {pepNO}      0.1783333 0.8991597
## [4] {marriedYES,children0,mortgageNO}  => {pepNO}      0.1733333 0.8965517
## [5] {children0,mortgageNO,pepNO}       => {marriedYES} 0.1733333 0.9719626
##     coverage  lift     count
## [1] 0.2100000 2.642857 111
## [2] 0.2250000 1.784266 110
## [3] 0.1983333 1.654895 107
```

```
## [4] 0.1933333 1.650095 104
## [5] 0.1783333 1.472671 104
```

Analyze rules with PEP on the RHS.

```
#create new bank rules with PEP on rhs all possible

newbankrules <- apriori(banktransactions, parameter = list(maxlen = 8, supp = .0001, conf = .0001), appe
arance = list(rhs = c("pepNO", "pepYES"), default = "lhs"))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##       1e-04    0.1    1 none FALSE            TRUE       5   1e-04      1
##  maxlen target  ext
##       8  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 0
##
## set item appearances ...[2 item(s)] done [0.00s].
## set transactions ...[30 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [30 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8

## Warning in apriori(banktransactions, parameter = list(maxlen = 8, supp =
## 1e-04, : Mining stopped (maxlen reached). Only patterns up to a length of 8
## returned!

##  done [0.18s].
## writing ... [177717 rule(s)] done [0.68s].
## creating S4 object  ... done [0.13s].

newbankrulesdf <- as(newbankrules, "data.frame")

# top 10 rules sorted by highest support

inspect((sort(newbankrules, decreasing = TRUE, by = "support"))[1:10])

##      lhs                          rhs      support   confidence coverage
## [1]  {}                        => {pepNO}  0.5433333 0.5433333  1.0000000
## [2]  {}                        => {pepYES} 0.4566667 0.4566667  1.0000000
## [3]  {current_actYES}          => {pepNO}  0.4066667 0.5362637  0.7583333
## [4]  {marriedYES}              => {pepNO}  0.4033333 0.6111111  0.6600000
## [5]  {save_actYES}             => {pepNO}  0.3916667 0.5676329  0.6900000
## [6]  {current_actYES}          => {pepYES} 0.3516667 0.4637363  0.7583333
## [7]  {mortgageNO}              => {pepNO}  0.3483333 0.5345269  0.6516667
```

```
## [8]  {mortgageNO}               => {pepYES} 0.3033333 0.4654731  0.6516667
## [9]  {save_actYES}              => {pepYES} 0.2983333 0.4323671  0.6900000
## [10] {save_actYES,current_actYES} => {pepNO}  0.2983333 0.5611285  0.5316667
##      lift      count
## [1]  1.0000000 326
## [2]  1.0000000 274
## [3]  0.9869885 244
## [4]  1.1247444 242
## [5]  1.0447230 235
## [6]  1.0154809 211
## [7]  0.9837918 209
## [8]  1.0192843 182
## [9]  0.9467894 179
## [10] 1.0327519 179
```

```
# top 10 rules sorted by highest confidence

inspect((sort(newbankrules, decreasing = TRUE, by = "confidence"))[1:10])
```

```
##      lhs                                    rhs      support
## [1]  {children3,save_actNO}                  => {pepNO}  0.036666667
## [2]  {age_18-27,regionSUBURBAN,children3}     => {pepNO}  0.001666667
## [3]  {age_58-67,regionSUBURBAN,children3}     => {pepNO}  0.001666667
## [4]  {age_38-47,regionSUBURBAN,children3}     => {pepYES} 0.001666667
## [5]  {regionSUBURBAN,children3,current_actNO} => {pepNO}  0.001666667
## [6]  {regionSUBURBAN,children3,save_actNO}    => {pepNO}  0.001666667
## [7]  {middleIncome,regionSUBURBAN,children3}  => {pepNO}  0.001666667
## [8]  {regionSUBURBAN,children3,mortgageYES}   => {pepNO}  0.001666667
## [9]  {age_48-57,regionSUBURBAN,children1}     => {pepYES} 0.003333333
## [10] {age_48-57,regionSUBURBAN,current_actNO} => {pepNO}  0.001666667
##      confidence coverage    lift     count
## [1]  1          0.036666667 1.840491 22
## [2]  1          0.001666667 1.840491  1
## [3]  1          0.001666667 1.840491  1
## [4]  1          0.001666667 2.189781  1
## [5]  1          0.001666667 1.840491  1
## [6]  1          0.001666667 1.840491  1
## [7]  1          0.001666667 1.840491  1
## [8]  1          0.001666667 1.840491  1
## [9]  1          0.003333333 2.189781  2
## [10] 1          0.001666667 1.840491  1
```

```
# top 10 rules sorted by highest lift

inspect((sort(newbankrules, decreasing = TRUE, by = "lift"))[1:10])
```

```
##      lhs                                    rhs      support
## [1]  {age_38-47,regionSUBURBAN,children3}     => {pepYES} 0.001666667
## [2]  {age_48-57,regionSUBURBAN,children1}     => {pepYES} 0.003333333
## [3]  {age_28-37,regionSUBURBAN,marriedNO}     => {pepYES} 0.005000000
## [4]  {age_18-27,middleIncome,regionSUBURBAN}  => {pepYES} 0.003333333
```

```
## [5]  {age_58-67,regionSUBURBAN,children1}     => {pepYES} 0.003333333
## [6]  {highIncome,regionSUBURBAN,children2}    => {pepYES} 0.013333333
## [7]  {regionSUBURBAN,children2,mortgageYES}   => {pepYES} 0.006666667
## [8]  {age_38-47,regionSUBURBAN,children1}     => {pepYES} 0.005000000
## [9]  {regionSUBURBAN,children1,current_actNO} => {pepYES} 0.003333333
## [10] {middleIncome,regionSUBURBAN,children1}  => {pepYES} 0.008333333
##      confidence coverage    lift     count
## [1]  1          0.001666667 2.189781 1
## [2]  1          0.003333333 2.189781 2
## [3]  1          0.005000000 2.189781 3
## [4]  1          0.003333333 2.189781 2
## [5]  1          0.003333333 2.189781 2
## [6]  1          0.013333333 2.189781 8
## [7]  1          0.006666667 2.189781 4
## [8]  1          0.005000000 2.189781 3
## [9]  1          0.003333333 2.189781 2
## [10] 1          0.008333333 2.189781 5
```

My top five interesting rules.

1. {age_58-67} => {pepYES}

support = 0.13, confidence = 0.61, lift = 1.34

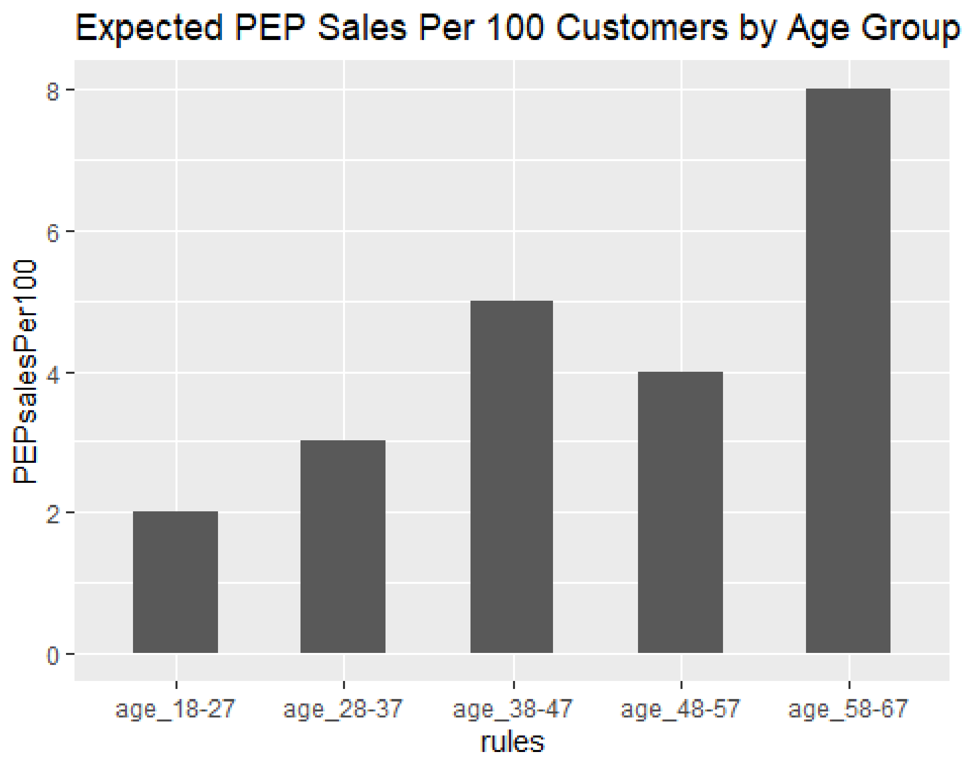
Customers who are in the age bracket of 58 to 67 are 61% likely to purchase PEP. 13% of all customers fall into this age bracket. This means that every 100 customers will produce about 8 PEP sales. Out of all the other age groups, the next highest was the 38 to 47 age bracket with 5 PEP sales per 100 customers.

It makes sense that the older demographic is more likely to purchase the PEP, because they are also probably more likely to respond to direct mail in general than younger generations are. Also, as shown earlier, there is a positive correlation with age, income, and a PEP being purchased, which aligns with this rule. Below is a graph illustrating the estimated PEP sales for all of the age groups.

Cumulative expected PEP sales per 100 customers = 8 Rule 1 = 8

```
# rule number 1 supporting evidence

temprules <- head(newbankrulesdf[which(grepl('age_', newbankrulesdf$rules)),], 10)
temprules <- temprules[which(grepl('pepYES', temprules$rules)),]
temprules$PEPsalesPer100 <- round((temprules$support * 100) * temprules$confidence, 0)
temprules <- temprules[,which(colnames(temprules) %in% c('rules', 'PEPsalesPer100'))]
temprules$rules <- substr(temprules$rules, 2, (unique(unlist(str_locate_all(temprules$rules, '\\}'))) [1]) -1)

ggplot(temprules, aes(rules, PEPsalesPer100)) + geom_col(aes(width = 0.5)) +
  ggtitle("Expected PEP Sales Per 100 Customers by Age Group")
```

```
## Warning: Ignoring unknown aesthetics: width
```



2. {highIncome,children1} => {pepYES} support = 0.08, confidence = 0.96, lift = 2.10
{highIncome,children2} => {pepYES} support = 0.07, confidence = 0.89, lift = 1.96
{middleIncome,children1} => {pepYES} support = 0.06, confidence = 0.95, lift = 2.08

My second rule includes the combination of of three rules. If the customer has either one or two children and and is in the high income bracket, or the customer has one child and is in the middle income bracket, there are highly likely to purhase PEP. About 21% of all transactions constitute this rule, and so out of every 100 customers, this should produce a total of 20 PEP sales.

Some of the 58 to 67 age bracket will cross over with this subsegment. Since these customers are already accounted for via the previous rule, I will be excluding them from the cumulative totals. After the deduction, the expected PEP sales for this rule is (20 - 7.72) = 12 expected PEP sales.

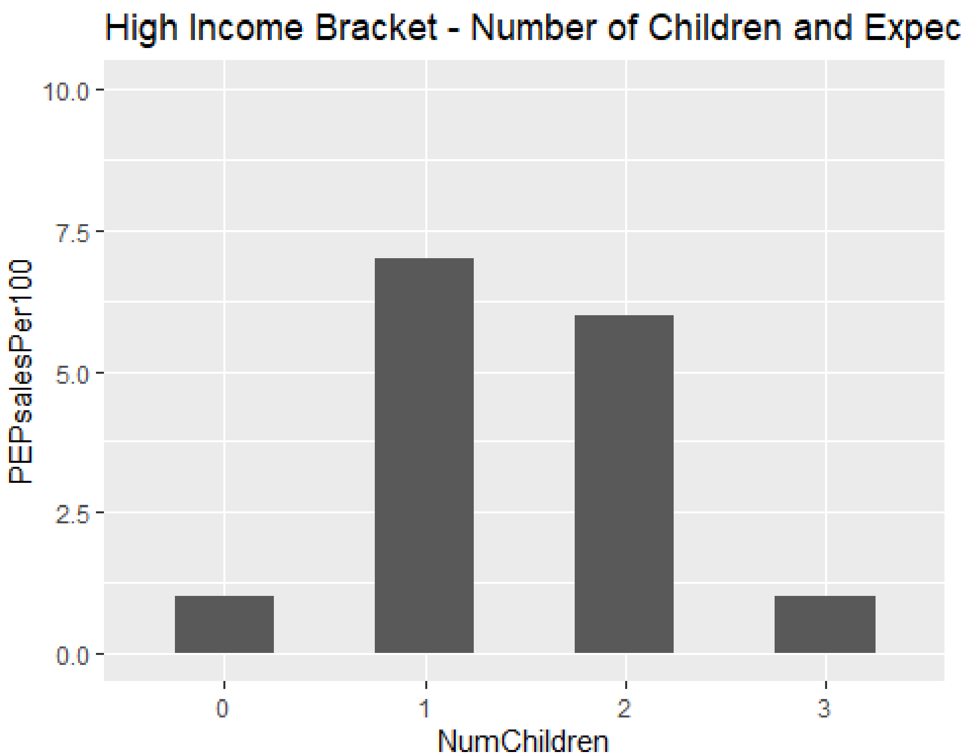
highIncome, children1, and age 58-67 ——> ((23 / 600) x 100) x .96 = 3.68
highIncome, children2, and age 58-67 ——> ((23 / 600) x 100) x .89 = 3.41
middleIncome, children1, and age 58-67 ——> ((4 / 600) x 100) x .95 = 0.63

The graph below shows the expected PEP sales for each of the income brackets and for how many children the customers have in each of those income brackets. Although the low income bracket with 1 child does produce an estimated 2 PEP sales per 100 customers, this not nearly as much as the other demographics.

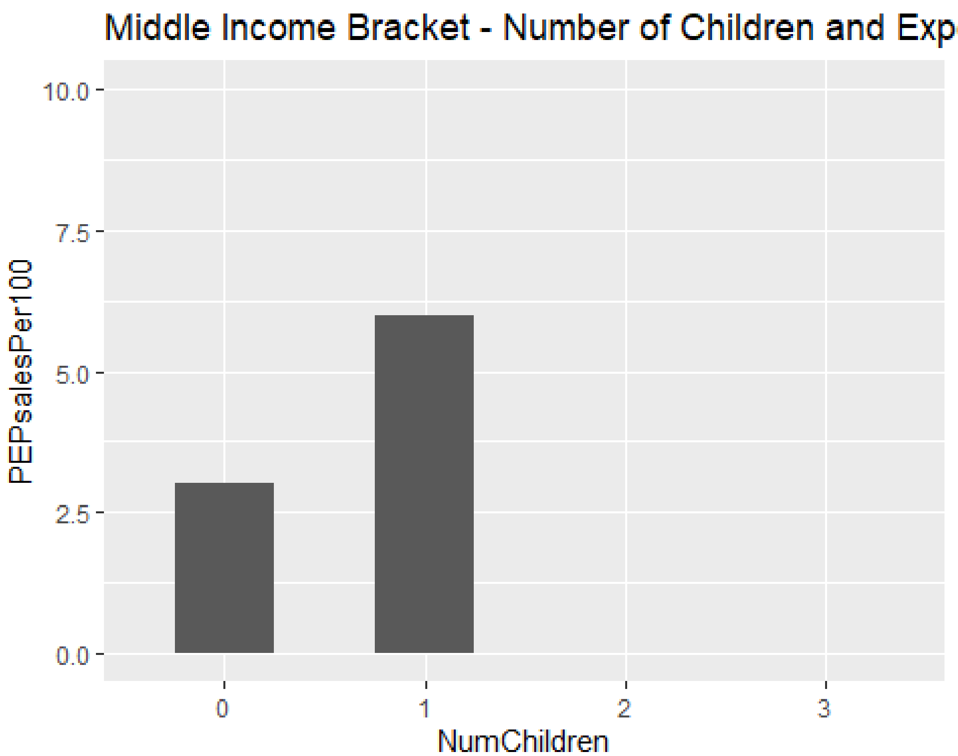
I am recommending not to market to the low income group altogether. The data for low income in aggregate shows that for every 100 customers, there will be 4 PEP sales. This is a much smaller percentage. I am proposing that this group is not worth marketing to and the resources should be prioritized for middle and high income bracket. The support for this group and confidence is not high enough.

Cumulative expected PEP sales per 100 customers = 20
Rule 1 = 8, Rule 2 = 12

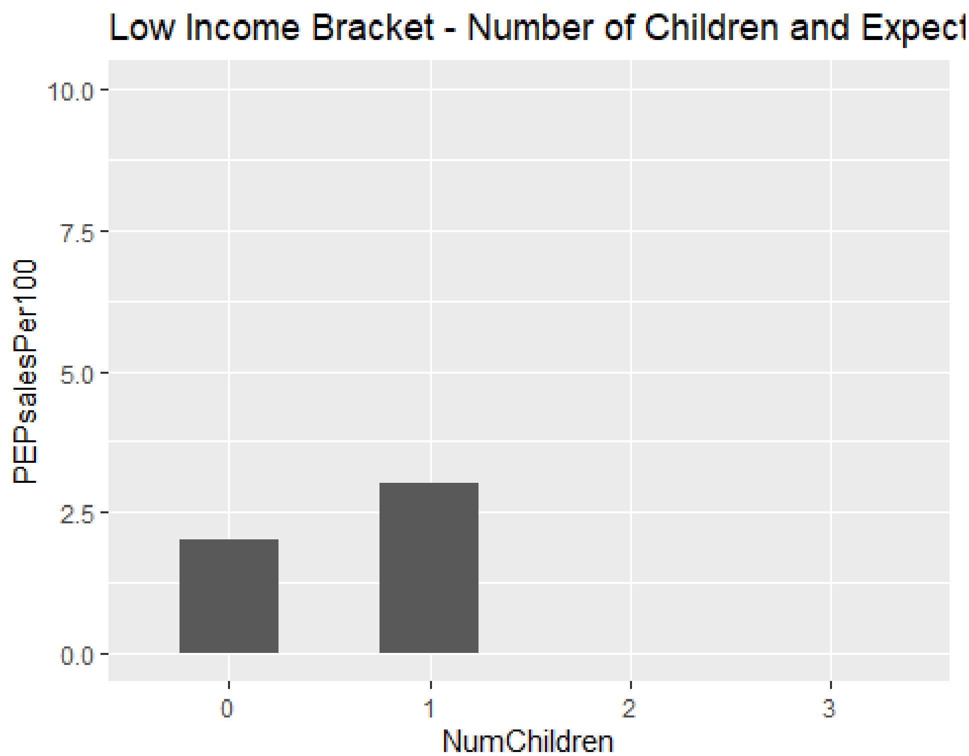
```
# rule number 2 supporting evidence

temprules             <- newbankrulesdf[which(grepl('Income', newbankrulesdf$rules)),]
temprules             <- temprules[which(grepl('children', temprules$rules)),]
temprules             <- temprules[which(grepl('pepYES', temprules$rules)),]
temprules             <- head(temprules, 12)
temprules$PEPsalesPer100   <- round((temprules$support * 100) * temprules$confidence, 0)

# parse through the rules column text

cutpoints <- c(); for (rule in temprules$rules) {
  rulecutpoint <- (unique(unlist(str_locate_all(rule, '\\}'))) [1]) -1
  cutpoints <- c(cutpoints, rulecutpoint)}
temprules$cutpoints <- cutpoints
temprules$rules <- substr(temprules$rules, 2, temprules$cutpoints)

cutpoints <- c(); for (rule in temprules$rules) {
  rulecutpoint <- (unique(unlist(str_locate_all(rule, ','))) [1]) -1
  cutpoints <- c(cutpoints, rulecutpoint)}
temprules$cutpoints <- cutpoints
temprules$IncomeBracket <- substr(temprules$rules, 1, temprules$cutpoints)

cutpoints <- c(); for (rule in temprules$rules) {
  rulecutpoint <- nchar(rule)
  cutpoints <- c(cutpoints, rulecutpoint)}
temprules$cutpoints <- cutpoints
temprules$NumChildren <- substr(temprules$rules, temprules$cutpoints, temprules$cutpoints)

temprules <- temprules[,which(colnames(temprules) %in% c('IncomeBracket', 'NumChildren', 'PEPsalesP
er100'))]

# plot each of the income brackets

highincomebracket <- sqldf('select * from temprules where IncomeBracket = \'highIncome\'')
ggplot(highincomebracket, aes(x = NumChildren, y = PEPsalesPer100)) + geom_col(width = 0.5,) + ylim
(0, 10) +
  ggtitle("High Income Bracket - Number of Children and Expected PEP Sales per 100 Customers")
```




```
middleincomebracket <- sqldf('select * from temprules where IncomeBracket = \'middleIncome\'')
ggplot(middleincomebracket, aes(x = NumChildren, y = PEPsalesPer100)) + geom_col(width = 0.5) + yli
m(0, 10) +
  ggtitle("Middle Income Bracket - Number of Children and Expected PEP Sales per 100 Customers")
```




```
lowincomebracket <- sqldf('select * from temprules where IncomeBracket = \'lowIncome\'')
ggplot(lowincomebracket, aes(x = NumChildren, y = PEPsalesPer100)) + geom_col(width = 0.5) + ylim(
0, 10) +
  ggtitle("Low Income Bracket - Number of Children and Expected PEP Sales per 100 Customers")
```




```
# clean up the environment

rm(cutpoints, rule, rulecutpoint, temprules)
```

3. {marriedNO,save_actYES,mortgageNO} => {pepYES} support = 0.11, confidence = 0.74, lift = 1.63
   {marriedNO,current_actYES,mortgageNO} => {pepYES} support = 0.12, confidence = 0.72, lift = 1.57

   I am recommending the combination of these two groups, given that they are somewhat similar. There is some overlap between these two rules. There is also some overlap with the previous two rules. For example, some of these customers may also fall in the age 58-67 bracket. To prevent double counting, the deductions are applied, and then an adj. support and avg. confidence is used to calculate the estimated PEP sales per 100 customers.

   After the deductions are applied, the total adj. support comes out to be (71 / 600) = 0.12. The avg. confidence is (0.74 + 0.72) / 2 = 0.73. Therefore, the expected PEP sales per 100 customers is = (0.12 x 100) x 0.72 = 8.64 = 8 (rounded down).

   As seen in the evidence presented below, when mortgage is not taken into account, the percentage of confidence for not being married along with a savings or current account is around 56-58%. When not having a mortgage is introduced as a factor, that jumps up to 70%, leading to an extra 1-2 PEP sales from the target segment.

Cumulative expected PEP sales per 100 customers = 28
Rule 1 = 8, Rule 2 = 12, Rule 3 = 8

```
# rule number 3 supporting evidence

temprules               <- newbankrulesdf[which(grepl('married', newbankrulesdf$rules)),]
temprules               <- temprules[which(grepl('pepYES', temprules$rules)),]
temprules               <- temprules[which(grepl('marriedNO', temprules$rules)),]
temprules               <- temprules[which(grepl('current_act', temprules$rules) | grepl('save_act', temprules
$rule)),]
temprules               <- sqldf('select * from temprules order by count desc')
temprules               <- head(temprules, 4)
temprules$PEPsalesPer100   <- round((temprules$support * 100) * temprules$confidence, 0)

temprules

##                                          rules   support confidence
## 1           {marriedNO,current_actYES} => {pepYES} 0.1583333  0.5864198
## 2              {marriedNO,save_actYES} => {pepYES} 0.1283333  0.5620438
## 3 {marriedNO,current_actYES,mortgageNO} => {pepYES} 0.1216667  0.7156863
## 4    {marriedNO,save_actYES,mortgageNO} => {pepYES} 0.1066667  0.7441860
##    coverage     lift count PEPsalesPer100
## 1 0.2700000 1.284131    95              9
## 2 0.2283333 1.230753    77              7
## 3 0.1700000 1.567196    73              9
## 4 0.1433333 1.629604    64              8

# calculate deductions

rule3deductions          <- bankcopy[which(

                         (bankcopy$married == 'NO' &
                          bankcopy$save_act == 'YES' &
                          bankcopy$mortgage == 'NO') |

                          # OR

                         (bankcopy$married == 'NO' &
                          bankcopy$current_act == 'YES' &
                          bankcopy$mortgage == 'NO')),]

rule3deductions          <- rule3deductions[which( ! rule3deductions$age >= 58),] # 30 deductions
rule3deductions          <- rule3deductions[which( ! (rule3deductions$income > 31132.77 & rule3deducti
ons$children == 1)),]   # 7 deductions
rule3deductions          <- rule3deductions[which( ! (rule3deductions$income > 31132.77 & rule3deducti
ons$children == 2)),]   # 7 deductions
rule3deductions          <- rule3deductions[which( ! ((rule3deductions$income > 20253.80 & rule3deduct
ions$income <= 31132.77) & rule3deductions$children == 1)),]  # 3 deductions
```

4. {children0,save_actNO,mortgageYES} => {pepYES}

    support = 0.06, confidence = 0.92, lift = 2.01

    To find this rule, I first filtered the rules down by removing subsegments that have been looked at so far. This rule encompasses a section of the market that has not been addressed yet by any of the previous rules.Therefore, there are no deductions needed for this rule. The expected PEP sales per 100 customers comes out to 5 (rounded down).

    Discuss the support, confidence and lift values and how they are interpreted in this data set.

    The support value of 0.06 means that 6% of all of the transactions in the data set contain children = 0, savings account = NO, and mortgage = YES. The confidence of 0.92 means that in these transactions, 92% of them resulted in a pep = YES. The lift of 2.01 indicates that pep = YES is highly likely to occur along with occurences of the LHS.

    Cumulative expected PEP sales per 100 customers = 33
    Rule 1 = 8, Rule 2 = 12, Rule 3 = 8, Rule 4 = 5

```
# rule number 4 supporting evidence

# remove all of the previous rules

temprules <- newbankrulesdf[which( ! grepl('age_58-67', newbankrulesdf$rules)),]
temprules <- temprules[which( ! (grepl('highIncome', temprules$rules) & grepl('children1', temprules$rul
es))),]
temprules <- temprules[which( ! (grepl('highIncome', temprules$rules) & grepl('children2', temprules$rul
es))),]
temprules <- temprules[which( ! (grepl('middleIncome', temprules$rules) & grepl('children1', temprules$r
ules))),]
temprules <- temprules[which( ! (grepl('marriedNO', temprules$rules) & grepl('save_actYES', temprules$
rules) & grepl('mortgageNO', temprules$rules))),]
temprules <- temprules[which( ! (grepl('marriedNO', temprules$rules) & grepl('current_actYES', temprul
es$rules) & grepl('mortgageNO', temprules$rules))),]

# find the rule

temprules              <- temprules[which(grepl('pepYES', temprules$rules)),]
temprules$PEPsalesPer100   <- round((temprules$support * 100) * temprules$confidence, 0)
temprules              <- temprules[which(temprules$PEPsalesPer100 >= 2),]
temprules              <- temprules[which( ! grepl('children1', temprules$rules)),]

# are there any deductions needed for age 58-67?
# (this is the only potential area of cross over)

length(bankcopy[which(

  bankcopy$age >= 58 &
  bankcopy$children == 0 &
```

```
 bankcopy$save_act == 'NO' &
 bankcopy$mortgage == 'Yes'), 1])

## [1] 0
```

5. {regionINNER_CITY,carYES,save_actNO} => {pepYES} support = 0.05, confidence = 0.60, lift = 1.31
   {regionINNER_CITY,carYES,current_actNO} => {pepYES} support = 0.03, confidence = 0.57, lift = 1.24

   The last market segment includes customers who live in the inner city region, own a car, and do not have either a savings account or a current account. The intent of this rule is to try and find a niche in the market that has not been addressed yet by the previous rules. This was difficult because the previous rules have already accounted for most of the customers.

   Deductions are necessary for this category because there is some overlap with previous category. After the deductions are applied, the total adj. support comes out to be (25 / 600) = 0.04. Given that there is some overlap between these two rules, the estimated PEP sales per 100 customers is calculated using an avg. confidence.

```
# adj. support = 0.04, avg. confidence = 0.585

cat('Expected PEP sales per 100 customers = ', ((0.04 * 100) * 0.585))

## Expected PEP sales per 100 customers =  2.34
```

Cumulative expected PEP sales per 100 customers = 35 Rule 1 = 8, Rule 2 = 12, Rule 3 = 8, Rule 4 = 5, Rule 5 = 2

```
# rule number 5 supporting evidence

# find the rule

temprules               <- temprules[which(grepl('regionINNER_CITY', temprules$rules)),]
temprules               <- temprules[which(grepl('carYES', temprules$rules)),]

# calculate deductions

rule5deductions          <- bankcopy[which(

                        (bankcopy$region == 'INNER_CITY' &
                         bankcopy$car == 'YES' &
                         bankcopy$current_act == 'NO') |

                         # OR

                        (bankcopy$region == 'INNER_CITY' &
                         bankcopy$car == 'YES' &
```

```
                          bankcopy$save_act == 'NO')),]

rule5deductions           <- rule5deductions[which( ! rule5deductions$age >= 58),] # 15 deductions
rule5deductions           <- rule5deductions[which( ! (rule5deductions$income > 31132.77 & rule5deducti
ons$children == 1)),]   # 0 deductions
rule5deductions           <- rule5deductions[which( ! (rule5deductions$income > 31132.77 & rule5deducti
ons$children == 2)),]   # 3 deductions
rule5deductions           <- rule5deductions[which( ! ((rule5deductions$income > 20253.80 & rule5deduct
ions$income <= 31132.77) & rule5deductions$children == 1)),]  # 4 deductions
rule5deductions           <- rule5deductions[which( ! (rule5deductions$married == 'NO' & rule5deductions
$save_act == 'YES' & rule5deductions$mortgage == 'NO')),]   # 4 deductions
rule5deductions           <- rule5deductions[which( ! (rule5deductions$married == 'NO' & rule5deductions
$current_act == 'YES' & rule5deductions$mortgage == 'NO')),]   # 4 deductions
rule5deductions           <- rule5deductions[which( ! (rule5deductions$children == 0 & rule5deductions$s
ave_act == 'NO' & rule5deductions$mortgage == 'YES')),]   # 8 deductions
```

Efficiency = total number of customers marketed to / expected PEP sales per 100 customers

The efficiency is the ratio of the total number of customers marketed to over the the expected PEP sales per 100 customers. If this number is low, it indicates good utilization of resources. If this number is high, it indicates poor utilization of resources. While the target customer segment segment does not achieve the full 45 PEP sales that it could if it marketed to the entire set of customers, it gets close and with much less resources. The rules that were chosen maximize the amount of PEP sales while minimizing the amount of resources used.

Efficiency of marketing to all customers (current state) = 13.3

total number of customers marketed to = 600 expected PEP sales per 100 customers = 45

Efficiency of marketing to target customers (future state) = 9.6

total number of customers marketed to = 335 expected PEP sales per 100 customers = 35

```
# total number of customers marketed to in the target segment

newcustomersegment <- bankcopy[which(

  bankcopy$age >= 58 |
  (bankcopy$income > 31132.77 & bankcopy$children == 1) |
  (bankcopy$income > 31132.77 & bankcopy$children == 2) |
  ((bankcopy$income > 20253.80 & bankcopy$income <= 31132.77) & bankcopy$children == 1) |
  (bankcopy$married == 'NO' & bankcopy$save_act == 'YES' & bankcopy$mortgage == 'NO') |
  (bankcopy$married == 'NO' & bankcopy$current_act == 'YES' & bankcopy$mortgage == 'NO') |
  (bankcopy$children == 0 & bankcopy$save_act == 'NO' & bankcopy$mortgage == 'YES') |
  (bankcopy$region == 'INNER_CITY' & bankcopy$car == 'YES' & bankcopy$save_act == 'NO') |
  (bankcopy$region == 'INNER_CITY' & bankcopy$car == 'YES' & bankcopy$current_act == 'NO')),]
```

Conclusion

From the original data provided, there is a little under a 50/50 shot for any given customer to purchase the PEP. So why not just continue to market the PEP to every customer? Almost half of them will end up purchasing the PEP anyway. For one, there are expenses that come with marketing. That would be a lot of money wasted on customers who don’t end up purchasing the PEP - money that could have been used for something else.

In this report, I attempted to identify pockets of customers that are most likely to purchase the PEP and it turns out that there were several of them. By narrowing down the amount of customers, the cost of marketing is able to be minimized while the number if PEP sales per customer is able to maximized. Therefore, I recommend continuing to use this method for these customers, as the direct mail advertising is shwown to be effective.

For the customer segments that were not likely to purchase the PEP, I would recommend collecting more data on. There could be a business opportunity to introduce a new product. For example, customers on the lower end of income and who are young are not likely to purchase the PEP. Was this the outcome that was expected and if not what is the reason that these customers do not purchase the PEP? Is there something else that they would purchase? Another factor to consider would be if a different type of marketing would be more effective. For example, a lot of marketing is done electronically nowadays, so this might lead to more purchases of the PEP from younger generations.

To recap, my final recommendation is to continue using the direct mail advertising for the PEP for the customer segments that this works best for. This will cut down the cost of marketing significantly without losing too many sales. The cost savings from narrowing down the marketing for PEP could be put towards further researching the market, developing a new product, or trying out an electronic marketing campaign.