Victoria_Haley_HW3

Victoria Haley

2023-04-24

HW 3 Association Rules

Introduction

This report examines the use of association rules to identify patterns in bank data that could be used to predict which customers are most likely to obtain a personal equity plan (PEP). By analyzing a range of customer attributes, such as age and income, the report aims to identify significant associations that could help to inform targeted marketing campaigns and other business strategies.

Importing the data

```
library(readr)
setwd("/Users/victoriahaley")
bank <-read_csv("Downloads/bankdata_csv_all.csv")
```

```
## Rows: 600 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): id, sex, region, married, car, save_act, current_act, mortgage, pep
## dbl (3): age, income, children
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Examine the data

The information provided in this section will give us some insight into the data structure, as well as a summary of some basic descriptive statistics of the variables.

```
str(bank)
```

```
## spc_tbl_ [600 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id         : chr [1:600] "ID12101" "ID12102" "ID12103" "ID12104" ...
##  $ age        : num [1:600] 48 40 51 23 57 57 22 58 37 54 ...
##  $ sex        : chr [1:600] "FEMALE" "MALE" "FEMALE" "FEMALE" ...
##  $ region     : chr [1:600] "INNER_CITY" "TOWN" "INNER_CITY" "TOWN" ...
##  $ income     : num [1:600] 17546 30085 16575 20375 50576 ...
##  $ married    : chr [1:600] "NO" "YES" "YES" "YES" ...
##  $ children   : num [1:600] 1 3 0 3 0 2 0 0 2 2 ...
##  $ car        : chr [1:600] "NO" "YES" "YES" "NO" ...
##  $ save_act   : chr [1:600] "NO" "NO" "YES" "NO" ...
##  $ current_act: chr [1:600] "NO" "YES" "YES" "YES" ...
##  $ mortgage   : chr [1:600] "NO" "YES" "NO" "NO" ...
##  $ pep        : chr [1:600] "YES" "NO" "NO" "NO" ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_character(),
##   ..   age = col_double(),
##   ..   sex = col_character(),
##   ..   region = col_character(),
##   ..   income = col_double(),
##   ..   married = col_character(),
##   ..   children = col_double(),
##   ..   car = col_character(),
##   ..   save_act = col_character(),
##   ..   current_act = col_character(),
##   ..   mortgage = col_character(),
##   ..   pep = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
sum(is.na(bank)) #There are no NAs in the dataframe as they were removed via the import wizard.
```

```
## [1] 0
```

```r
summary(bank)
```

```
##       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  
## 
## 
## 
```

The following are the first necessary steps required for association rule mining:

- The id fields will need be removed.
- The "*age*" and "*income*" fields will need to be discretized so that the data can be managed more easily.
- The "*sex*", "*region*", "*married*", "*car*", "*save_act*", "*current_act*", "*mortgage*", and "*pep*" fields will need to be converted from character to factor, and the "*children*" field will need to be converted to ordinal so that data can be analyzed.

Data Cleaning/Prep

Here, all steps identified above will be addressed.

```
bank <- bank[,-1]
head(bank) #id field removed
```

```
## # A tibble: 6 x 11
##     age sex    region income married child~1 car   save_~2 curre~3 mortg~4 pep
##   <dbl> <chr>  <chr>   <dbl> <chr>     <dbl> <chr> <chr>   <chr>   <chr>   <chr>
## 1    48 FEMALE INNER~ 17546  NO            1 NO    NO      NO      NO      YES
## 2    40 MALE   TOWN   30085. YES           3 YES   NO      YES     YES     NO
## 3    51 FEMALE INNER~ 16575. YES           0 YES   YES     YES     NO      NO
## 4    23 FEMALE TOWN   20375. YES           3 NO    NO      YES     NO      NO
## 5    57 FEMALE RURAL  50576. YES           0 NO    YES     NO      NO      NO
## 6    57 FEMALE TOWN   37870. YES           2 NO    YES     YES     NO      YES
## # ... with abbreviated variable names 1: children, 2: save_act, 3: current_act,
## #   4: mortgage
```

```
#discretization of age and income
bank$age <- cut(bank$age, breaks = c(0,20,30,40,50,60,100),labels = c("0-19", "20-29","30-39","40-49",
bank$income <- cut(bank$income, breaks = c(0, 15000, 25000, 35000, 45000, 55000,65000), labels = c("0-1

table(bank$age)
```

```
##
##  0-19 20-29 30-39 40-49 50-59   60+
##    21   123   117   141   100    98
```

```
table(bank$income)
```

```
##
##      0-14,999 15,000-24,999 25,000-34,999 35,000-44,999 45,000-54,999
##           102           200           142            82            51
## 55,000-65,000
##            23
```

This allows the data to be organized into “bins” where each customer will be placed into based on the age and income criteria set above.

```
#convert character fields to factors
bank$sex <-as.factor(bank$sex)
bank$region <- as.factor(bank$region)
bank$married <- as.factor(bank$married)
bank$car <- as.factor(bank$car)
bank$save_act <- as.factor(bank$save_act)
bank$current_act <- as.factor(bank$current_act)
bank$mortgage <- as.factor(bank$mortgage)
bank$pep <- as.factor(bank$pep)

#convert children to ordinal factor
bank$children <-ordered(bank$children)

str(bank)
```

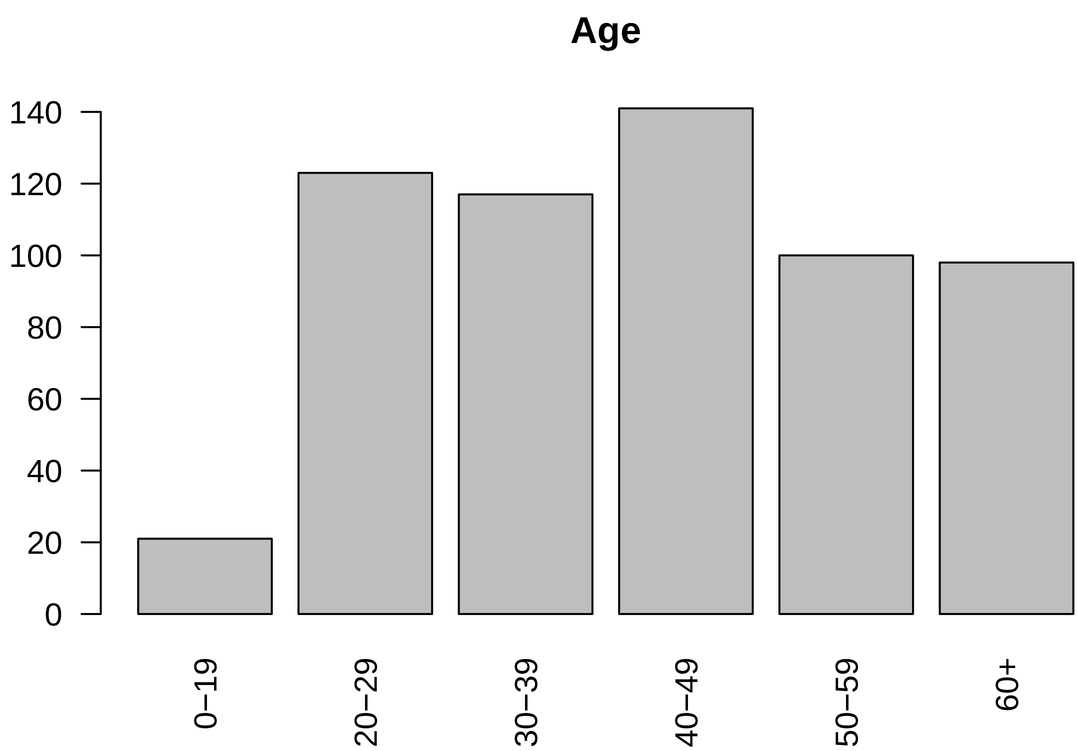
```
## tibble [600 x 11] (S3: tbl_df/tbl/data.frame)
##  $ age        : Factor w/ 6 levels "0-19","20-29",..: 4 4 5 2 5 5 2 5 3 5 ...
##  $ sex        : Factor w/ 2 levels "FEMALE","MALE": 1 2 1 1 1 1 2 2 1 2 ...
##  $ region     : Factor w/ 4 levels "INNER_CITY","RURAL",..: 1 4 1 4 2 4 2 4 3 4 ...
##  $ income     : Factor w/ 6 levels "0-14,999","15,000-24,999",..: 2 3 2 2 5 4 1 2 3 2 ...
##  $ married    : Factor w/ 2 levels "NO","YES": 1 2 2 2 2 2 1 2 2 2 ...
```

```
##  $ children   : Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 2 4 1 4 1 3 1 1 3 3 ...
##  $ car        : Factor w/ 2 levels "NO","YES": 1 2 2 1 1 1 1 2 2 2 ...
##  $ save_act   : Factor w/ 2 levels "NO","YES": 1 1 2 1 2 2 1 2 1 2 ...
##  $ current_act: Factor w/ 2 levels "NO","YES": 1 2 2 2 1 2 2 2 1 2 ...
##  $ mortgage   : Factor w/ 2 levels "NO","YES": 1 2 1 1 1 1 1 1 1 1 ...
##  $ pep        : Factor w/ 2 levels "NO","YES": 2 1 1 1 1 2 2 1 1 1 ...
```

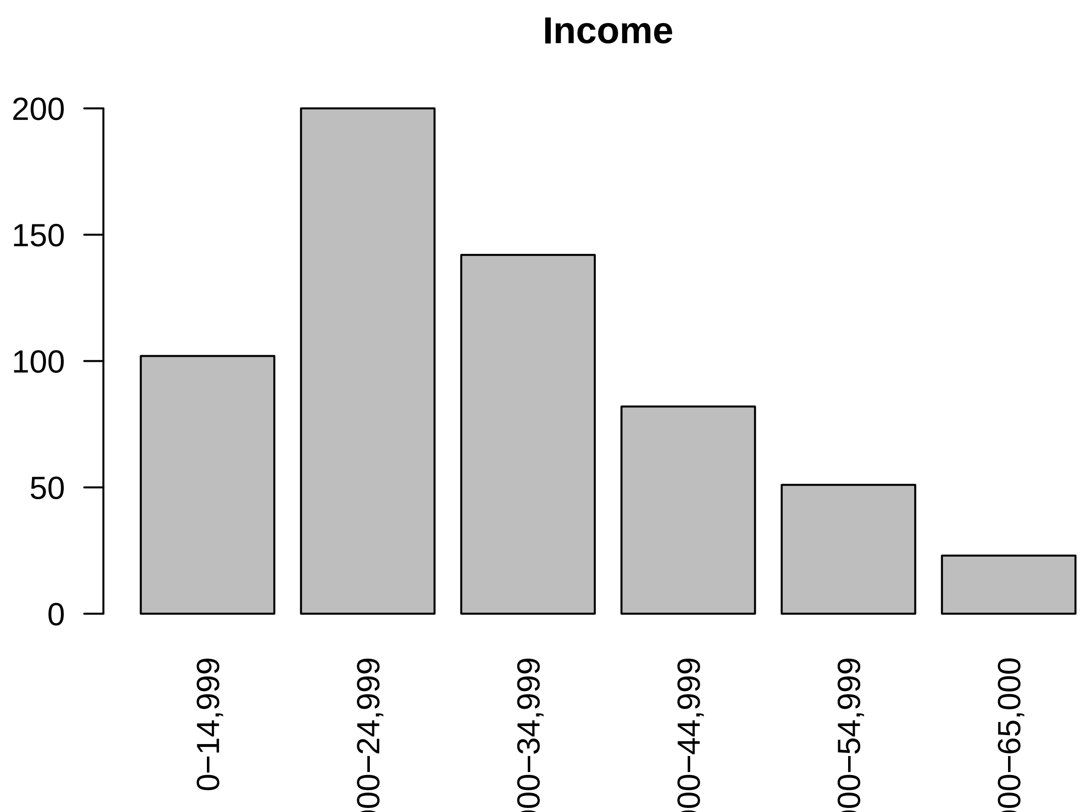
EDA and Visualization

Time to dive in and explore the data.

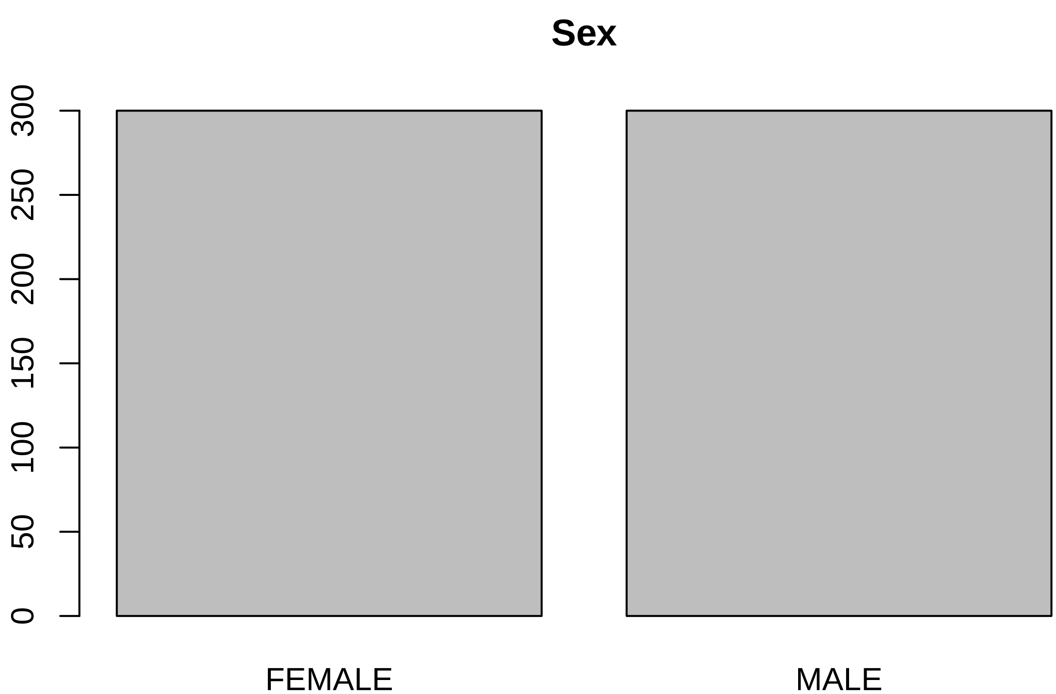
```
plot(bank$age, main = "Age",las=2)
```



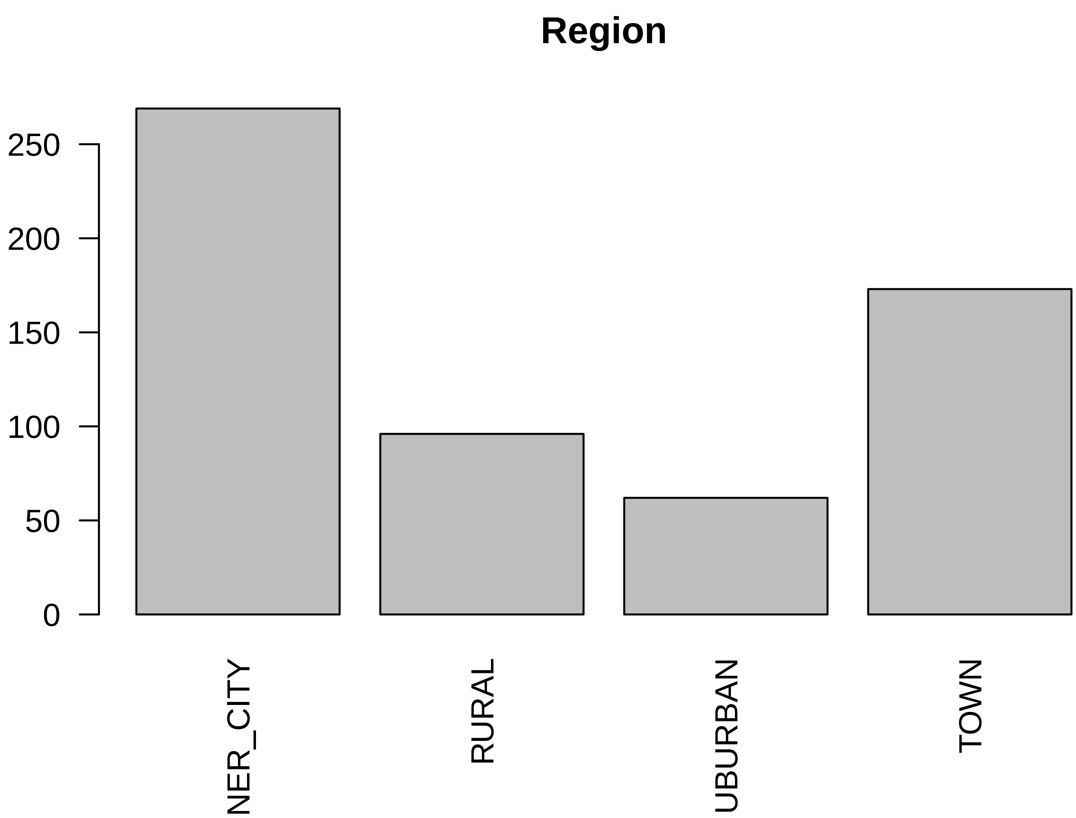

```
plot(bank$income, main = "Income", las=2)
```



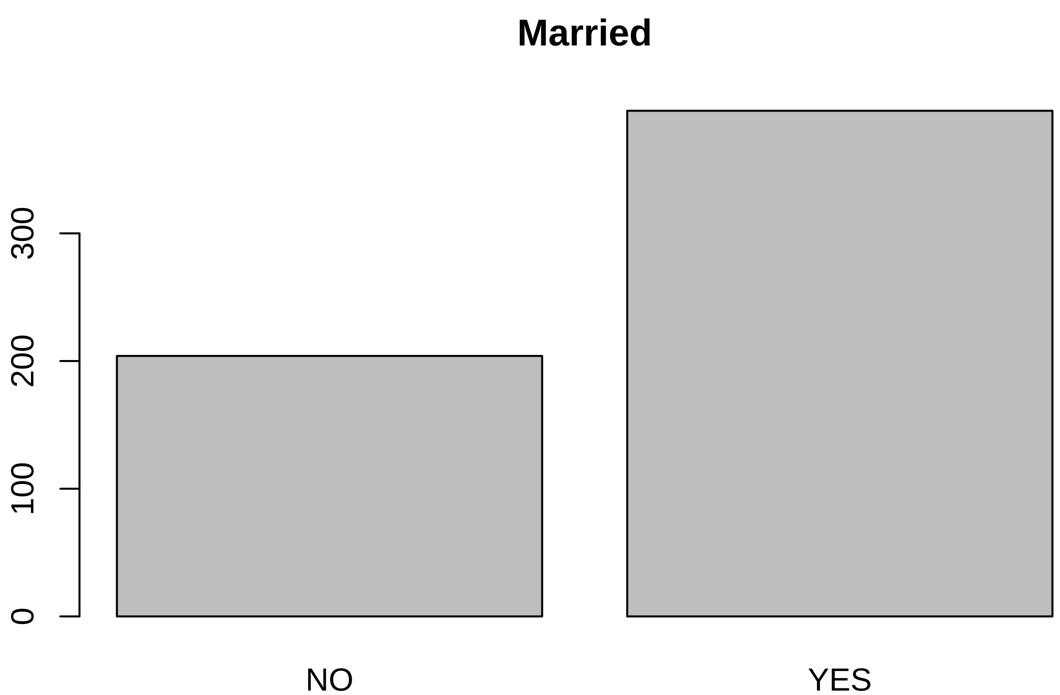

```
plot(bank$sex, main = "Sex")
```



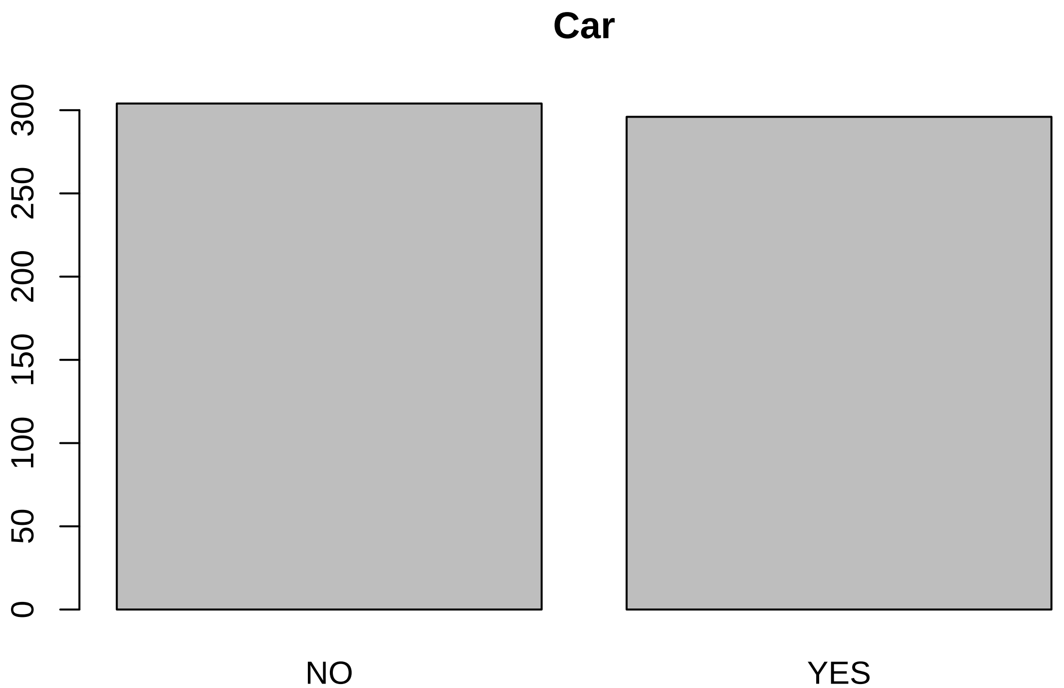

```
plot(bank$region, main = "Region", las=2)
```



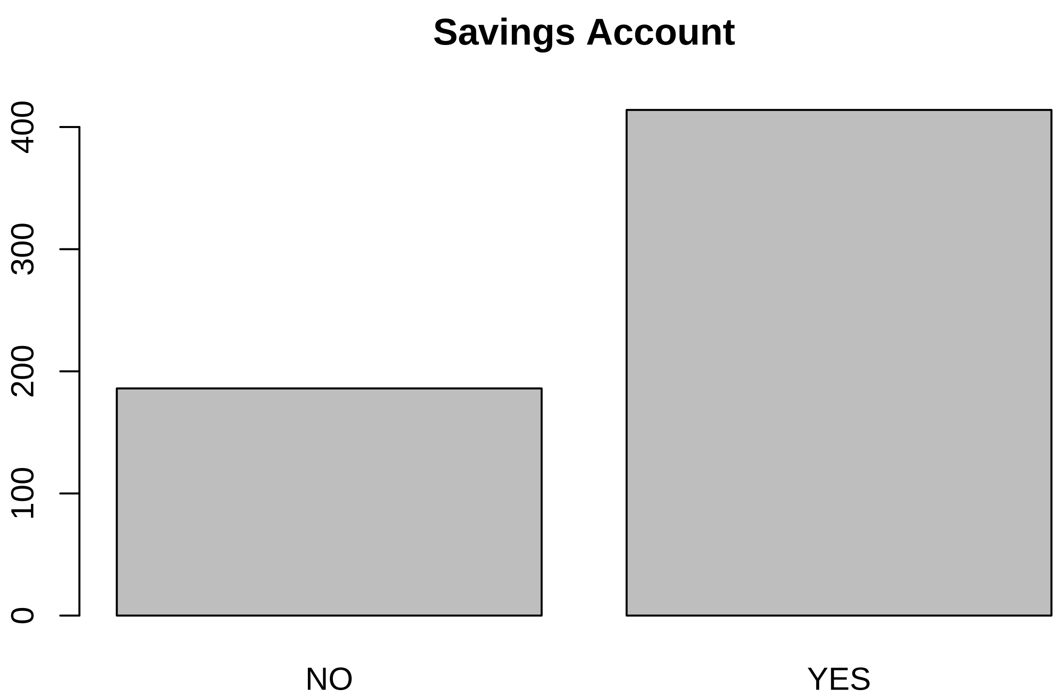

```
plot(bank$married, main = "Married")
```



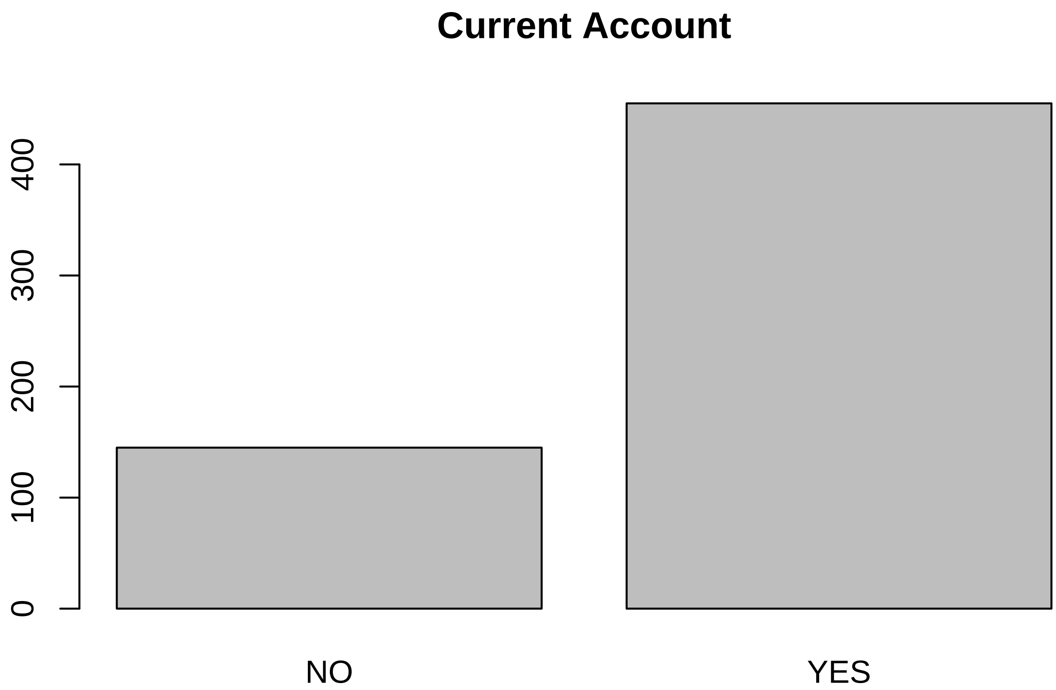

```
plot(bank$car, main = "Car")
```




```r
plot(bank$save_act, main = "Savings Account")
```

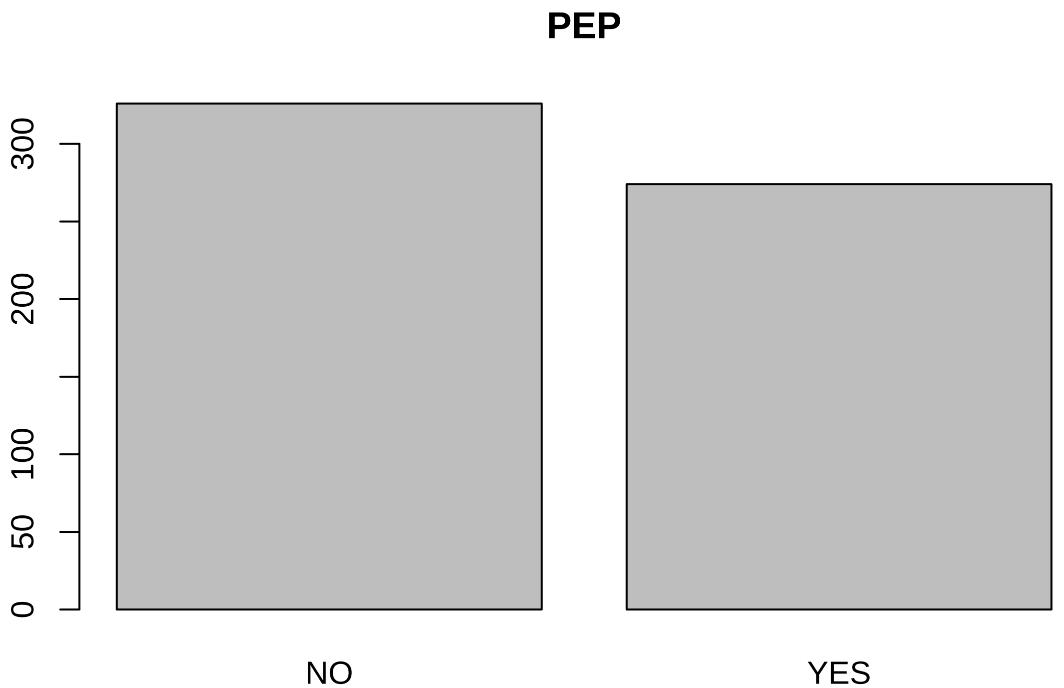



```r
plot(bank$current_act, main = "Current Account")
```

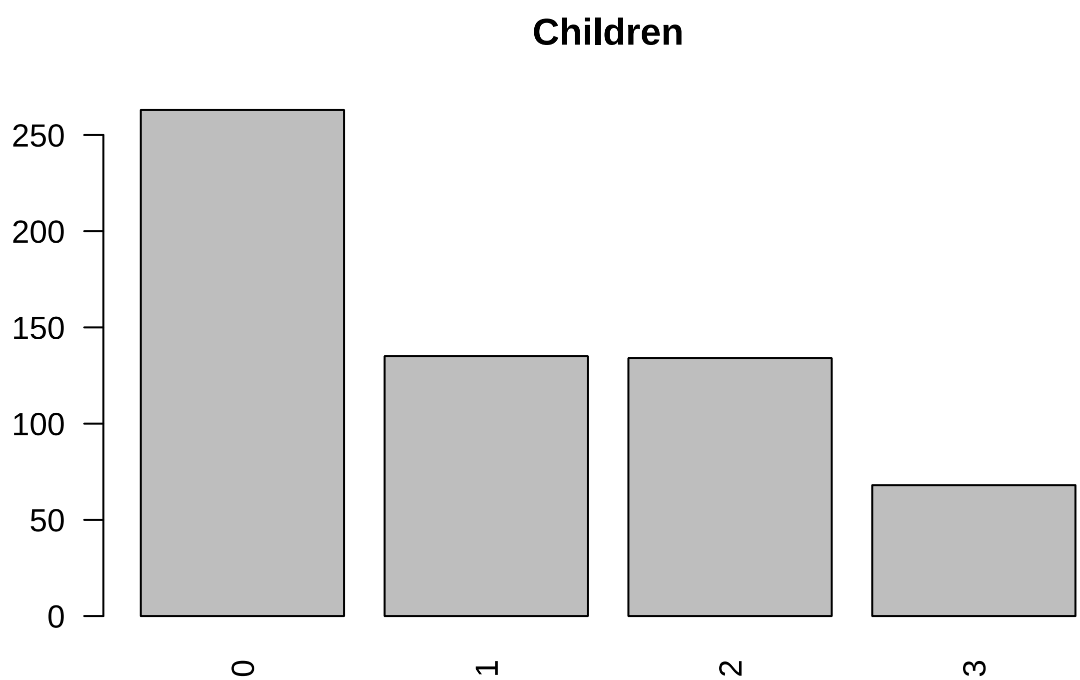



```
plot(bank$mortgage, main = "Mortgage")
```


```
plot(bank$pep, main = "PEP")
```




```
plot(bank$children, main = "Children", las=2)
```



Interesting things to note:

- Most income ranges fall between $15,000-$24,999 and $25,000-$34,999.
- Gender and whether or not the customer has a car is split pretty evenly.
- Most live in the inner city, while the suburban area has the least amount of customers.
- Over half of the customers are married.
- Most customers have both a savings and current account.

- Most customers do not have a mortgage or PEP.
- Most customers do not have any children.

Association Model (Apriori)

In order to find which customers are likely to obtain the Personal Equity Plan (PEP), the apriori algorithm can be used to evaluate association rules based on how popular an itemset is (support), how often items A and B occur together (confidence), and the strength of a rule (lift).

```
library(arules)
```

```
## Loading required package: Matrix

##
## Attaching package: 'arules'

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

```
pep_rules <- apriori(bank, parameter = list(support=0.02, conf=0.95))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.95    0.1    1 none FALSE            TRUE       5    0.02      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: 12
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[34 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.01s].
## writing ... [736 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

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

```
##     lhs                         rhs              support confidence   coverage     lift count
## [1] {married=NO,
##      children=0,
##      mortgage=YES,
##      pep=YES}                => {save_act=NO}  0.02000000  1.0000000 0.02000000 3.225806    12
## [2] {region=INNER_CITY,
##      income=15,000-24,999,
##      children=0,
##      mortgage=YES,
##      pep=YES}                => {save_act=NO}  0.02166667  1.0000000 0.02166667 3.225806    13
```

```
## [3] {income=15,000-24,999,
##      children=0,
##      mortgage=YES,
##      pep=YES}                => {save_act=NO}  0.03166667  0.9500000 0.03333333 3.064516    19
## [4] {sex=MALE,
##      married=NO,
##      children=0,
##      save_act=YES,
##      pep=NO}                 => {mortgage=YES} 0.02000000  1.0000000 0.02000000 2.870813    12
## [5] {married=NO,
##      children=0,
##      save_act=YES,
##      pep=NO}                 => {mortgage=YES} 0.03833333  0.9583333 0.04000000 2.751196    23
```

This first run attempted to find rules that had at least 2% support with a confidence level of at least 95%. Unfortunately, these results were not very strong, with the top rule having a lift of only 3.2.

```r
pep_rules <- apriori(bank, parameter = list(supp= 0.025, conf=0.9))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5   0.025      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: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[34 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.01s].
## writing ... [814 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
pep_rules <- sort(pep_rules, decreasing = TRUE, by="lift")
inspect(pep_rules[1:5])
```

```
##     lhs                           rhs            support confidence   coverage     lift count
## [1] {income=55,000-65,000}     => {age=60+}     0.03500000  0.9130435 0.03833333 5.590062    21
## [2] {income=55,000-65,000,
##      save_act=YES}             => {age=60+}     0.03500000  0.9130435 0.03833333 5.590062    21
## [3] {income=55,000-65,000,
##      current_act=YES}          => {age=60+}     0.03166667  0.9047619 0.03500000 5.539359    19
## [4] {income=55,000-65,000,
##      save_act=YES,
##      current_act=YES}          => {age=60+}     0.03166667  0.9047619 0.03500000 5.539359    19
## [5] {income=15,000-24,999,
##      children=0,
##      mortgage=YES,
```

```
##      pep=YES}                => {save_act=NO} 0.03166667  0.9500000 0.03333333 3.064516    19
```

After adjusting the support level to 0.025 and the confidence to 0.9, it appears as though the strength of our rules has improved. Using our new found minimum support and confidence levels, we can try to find association rules for customers who are most likely to get the PEP.

```r
pep_rules <- apriori(bank, parameter = list(supp= 0.025, conf=0.9), appearance = list(default="lhs", rhs
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5   0.025      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: 15
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[34 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.01s].
## writing ... [129 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
pep_rules <- sort(pep_rules, decreasing = TRUE, by="lift")
inspect(pep_rules[1:5])
```

```
##     lhs                          rhs          support confidence   coverage     lift count
## [1] {age=40-49,
##      children=1}             => {pep=YES} 0.06000000          1 0.06000000 2.189781    36
## [2] {age=60+,
##      children=1,
##      mortgage=NO}            => {pep=YES} 0.02666667          1 0.02666667 2.189781    16
## [3] {age=60+,
##      children=1,
##      current_act=YES}        => {pep=YES} 0.02500000          1 0.02500000 2.189781    15
## [4] {age=40-49,
##      region=TOWN,
##      children=1}             => {pep=YES} 0.02666667          1 0.02666667 2.189781    16
## [5] {age=40-49,
##      income=15,000-24,999,
##      children=1}             => {pep=YES} 0.03000000          1 0.03000000 2.189781    18
```

The algorithm above found the top 5 rules that indicate which customers were most likely to obtain the Personal Equity Plan. This list was sorted by support rather than lift above to show the most popular rules since the strength did not vary much. The results of the rules are as follows:

1. *{age=[40,49), children=1} => {pep=YES}*
    - support = 0.06
    - confidence = 1

- lift = 2.2
- This rule, the most popular by far, suggests that customers between the ages of 40-50 with 1 child are the most likely to get the PEP. The bank would do well by marketing this to parents of older children as a good investment for college/their future.

2. *{age=[60+), children=1, mortgage=NO} => {pep=YES}*
   - support = 0.027
   - confidence = 1
   - lift = 2.2
   - This rule suggests that elderly customers with 1 child and no mortgage payments are the second most likely demographic to get the PEP. If going this route, the bank would best market this as a retirement option for older customers.
3. *{age=[60+), children=1, current_act=YES} => {pep=YES}*
   - support = 0.025
   - confidence = 1
   - lift = 2.2
   - Similar to the rule above, except this rule indicates that older customers with a current account with the bank are slightly (but not by much) less likely to get a PEP than those with no mortgage payments.
4. *{age=[40,49), region = TOWN, children=1} => {pep=YES}*
   - support = 0.027
   - confidence = 1
   - lift = 2.2
   - The demographic in this rule are similar to the top rule, however living in town sets this group apart. Again, marketing the PEP as an investment in the future would work well.
5. *{age=[40,49), income=15,000-24,999, children=1} => {pep=YES}*
   - support = 0.3
   - confidence = 1
   - lift = 2.2
   - Similar to rules 1 and 4, except that the customer's with income on the lower end of the overall range is a factor in this rule.

Conclusions

To summarize, this analysis of the bank data by using association rules has yielded several valuable insights for identifying potential customers likely to obtain a Personal Equity Plan (PEP). After preprocessing the data and applying the Apriori algorithm, the most frequent itemsets and generate rules with high support, confidence, and lift metrics were identified.

This analysis suggests that customers between the ages of 40-50 with 1 child are the most likely demographic to acquire the PEP. This information could be crucial for the bank's marketing strategy as it can now target this specific group of customers to improve PEP sales. Furthermore, the high confidence and lift values of the rule indicate that it is a robust and dependable association between the attributes.

Overall, the findings presented in this report provide valuable insights that can be used to inform the bank's marketing strategy and improve sales performance. With this information, the bank can make better-informed decisions and increase its success and profitability