Load

```
library(readr)
file='C:/Users/Arnold/OneDrive/R_Python_working_directory/IST 707 Data Analytics/bankdata_csv_all.csv'
df0 <- read_csv(file, col_types = cols(age = col_integer(),children = col_integer(), id = col_skip(),sex = col_factor(levels = c("MALE", "FEMALE"))))
```

Head of data

```
df=df0
head(df)

## # A tibble: 6 x 11
##     age sex   region income married children car   save_act current_act
##   <int> <fct> <chr>   <dbl> <chr>      <int> <chr> <chr>    <chr>
## 1    48 FEMA~ INNER~ 17546  NO             1 NO    NO       NO
## 2    40 MALE  TOWN   30085. YES            3 YES   NO       YES
## 3    51 FEMA~ INNER~ 16575. YES            0 YES   YES      YES
## 4    23 FEMA~ TOWN   20375. YES            3 NO    NO       YES
## 5    57 FEMA~ RURAL  50576. YES            0 NO    YES      NO
## 6    57 FEMA~ TOWN   37870. YES            2 NO    YES      YES
## # ... with 2 more variables: mortgage <chr>, pep <chr>
```

Data preprocess

Age binning

```
library(magrittr)
library(caret)
df$age=cut(df$age,seq(0,100,10))
```

Categorize Income to High, Medium, or Low

```
df$income=cut(df$income,breaks=3,labels = c('Low','Medium','High'))
```

Change children column values to YES or NO

```
df$children=ifelse(df$children==0,'NO','YES')
```

Change all columns to factor data type

```
library(purrr)
df=df %>% map_df(factor)
```

Next perform association rule discovery on the preprocessed data. Experiment with different parameters and preprocessing so that you get on the order of 20-30 strong rules, e.g. rules with high lift and confidence which at the same time have relatively good support. Don’t forget to report in details what you have tried. # First try

```r
library(arules)
rules=apriori(df, parameter = list(supp = 0.1, conf = 0.8))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 60
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[29 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 ... [112 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

There're too many rules, so I increase support to 0.2.

```r
rules=apriori(df, parameter = list(supp = 0.2, conf = 0.8))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.2      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 120
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[29 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [22 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [5 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

There're too little rules, so I decrease support to 0.15.

```
rules=apriori(df, parameter = list(supp = 0.15, conf = 0.8))

## 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 FALSE
##
## 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 ...[29 item(s), 600 transaction(s)] done [0.00s].
## sorting and recoding items ... [26 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [21 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

21 rules looks good, so I sort by confidence.

```
rules=sort(rules, by='confidence')
```

Inspect all the rules

```
inspect(rules)

##      lhs                       rhs                  support confidence
lift count
## [1]  {children=NO,
##       mortgage=NO,
##       pep=NO}               => {married=YES}     0.1733333  0.9719626
1.472671   104
## [2]  {age=(20,30]}          => {income=Low}      0.1883333  0.9495798
1.999115   113
## [3]  {married=YES,
##       children=NO,
##       save_act=YES}         => {pep=NO}          0.1783333  0.8991597
1.654895   107
## [4]  {married=YES,
##       children=NO,
##       mortgage=NO}          => {pep=NO}          0.1733333  0.8965517
1.650095   104
```

```
## [5]  {married=YES,
##       save_act=YES,
##       pep=YES}            => {children=YES}    0.1500000  0.8823529
1.570955    90
## [6]  {save_act=YES,
##       mortgage=NO,
##       pep=NO}             => {married=YES}     0.2000000  0.8450704
1.280410   120
## [7]  {children=NO,
##       pep=NO}             => {married=YES}     0.2350000  0.8443114
1.279260   141
## [8]  {save_act=YES,
##       current_act=YES,
##       mortgage=NO,
##       pep=NO}             => {married=YES}     0.1516667  0.8425926
1.276655    91
## [9]  {children=NO,
##       current_act=YES,
##       pep=NO}             => {married=YES}     0.1750000  0.8267717
1.252684   105
## [10] {married=NO,
##       save_act=YES}       => {current_act=YES} 0.1883333  0.8248175
1.087671   113
## [11] {mortgage=NO,
##       pep=NO}             => {married=YES}     0.2850000  0.8181818
1.239669   171
## [12] {children=NO,
##       save_act=YES,
##       pep=NO}             => {married=YES}     0.1783333  0.8167939
1.237567   107
## [13] {current_act=YES,
##       mortgage=NO,
##       pep=NO}             => {married=YES}     0.2150000  0.8164557
1.237054   129
## [14] {save_act=YES,
##       mortgage=NO,
##       pep=YES}            => {current_act=YES} 0.1733333  0.8125000
1.071429   104
## [15] {car=NO,
##       pep=YES}            => {current_act=YES} 0.1833333  0.8088235
1.066580   110
## [16] {sex=FEMALE,
##       mortgage=NO,
##       pep=NO}             => {married=YES}     0.1550000  0.8086957
1.225296    93
## [17] {car=NO,
##       save_act=YES,
##       mortgage=NO}        => {current_act=YES} 0.1733333  0.8062016
1.063123   104
## [18] {region=INNER_CITY,
```

```
##       save_act=YES,
##       mortgage=NO}       => {current_act=YES} 0.1500000  0.8035714
1.059655    90
## [19] {car=NO,
##       mortgage=NO}       => {current_act=YES} 0.2633333  0.8020305
1.057623   158
## [20] {sex=FEMALE,
##       region=INNER_CITY} => {current_act=YES} 0.1750000  0.8015267
1.056958   105
## [21] {children=YES,
##       mortgage=NO,
##       pep=YES}           => {current_act=YES} 0.1666667  0.8000000
1.054945   100
```

PEP as RHS

```
rules=apriori(data=df,parameter = list(supp=.1,conf=.7),appearance =
list(default='lhs',rhs=c('pep=YES','pep=NO')))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 60
##
## set item appearances ...[2 item(s)] done [0.00s].
## set transactions ...[29 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 ... [33 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

rules=sort(rules,by='confidence')
```

Top rules

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

##      lhs                  rhs        support confidence     lift count
## [1]  {married=YES,
##       children=NO,
```

```
##       save_act=YES,
##       current_act=YES} => {pep=NO} 0.1333333  0.9195402 1.692405    80
## [2]  {married=YES,
##       children=NO,
##       save_act=YES,
##       mortgage=NO}     => {pep=NO} 0.1216667  0.9125000 1.679448    73
## [3]  {married=YES,
##       children=NO,
##       current_act=YES,
##       mortgage=NO}     => {pep=NO} 0.1333333  0.9090909 1.673173    80
## [4]  {sex=FEMALE,
##       married=YES,
##       children=NO,
##       mortgage=NO}     => {pep=NO} 0.1050000  0.9000000 1.656442    63
## [5]  {married=YES,
##       children=NO,
##       save_act=YES}    => {pep=NO} 0.1783333  0.8991597 1.654895   107
## [6]  {married=YES,
##       children=NO,
##       mortgage=NO}     => {pep=NO} 0.1733333  0.8965517 1.650095   104
## [7]  {married=YES,
##       children=NO,
##       car=NO,
##       mortgage=NO}     => {pep=NO} 0.1000000  0.8955224 1.648201    60
## [8]  {sex=FEMALE,
##       married=YES,
##       children=NO,
##       current_act=YES} => {pep=NO} 0.1000000  0.8450704 1.555344    60
## [9]  {sex=FEMALE,
##       married=YES,
##       children=NO}     => {pep=NO} 0.1300000  0.8297872 1.527216    78
## [10] {married=YES,
##       children=NO,
##       car=NO,
##       current_act=YES} => {pep=NO} 0.1000000  0.8108108 1.492290    60
```

How support, confidence, & lift are calculated? (Rule 1 as an example)

- Support - (Number of rows with married=YES, children=NO, save_act=YES, current_act=YES, & pep=NO) / (Total number of rows)
- Confidence - (Number of rows with married=YES, children=NO, save_act=YES, current_act=YES, & pep=NO) / (Number of rows with married=YES, children=NO, save_act=YES, & current_act=YES)
- Lift - Confidence / support(pep=NO)

First we look at the top rule of highest confidence. It has support of 0.13, confidence of 0.92, & lift of 1.69. It's a interesting rule, because we can see what combination of characteristics of people are very unlikey to buy PEP. According to the LHS, we see that 92% of people from the data who are married with no kids, have saving account, & have current account, didn't buy PEP. Based on these characteristics, the company could do some further analysis to figure out why are these people very unlikely to buy PEP. To do so, the company could try to collect more data by survey or some other means. By providing some discount to people with some or all of these combination of characteristics could help increase their willingness to buy PEP.

Another interesting rule to look at is the 4th rule. It has support of 0.1, confidence of 0.9, & lift of 1.66. This rule is interesting because the LHS of this rule is a bit different than the first one. This rule says women married with no kids and don't have mortgage are very unlikely to buy PEP. Just like the first rule, the company could do more analysis to understand the low willingness of buying PEP fomr this group of women. Discount targeting this group of women might help increase the willingness to buy PEP as well.

After some inspections, another interesting rule was found at 25th row

```
inspect(rules[25])
```

```
##     lhs                            rhs      support   confidence lift
## [1] {region=TOWN,income=Low} => {pep=NO} 0.1016667 0.7261905  1.336547
##     count
## [1] 61
```

It has support of 0.1, confidence of 0.73, & lift of 1.34. It's interesting, because this group of people are also very unlikely to buy PEP, and they have different characteristics not included in the LHS in other rules mentioned. They have low income & are from town. Same as before, more analysis could be done to improve the plan for this group of people. And of course discount might increase their willingness to buy PEP, especially they are low income.

We havn't seen people who are likely to buy PEP yet, so lets create new rules.

```
rules=apriori(data=df,parameter = list(supp=.1,conf=.6),appearance =
list(default='lhs',rhs='pep=YES'))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
```

```
##     0.1 TRUE TRUE   FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 60
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[29 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 ... [5 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules=sort(rules,by='confidence')
```

Inspection

```
inspect(rules)
```

```
##     lhs                     rhs         support  confidence     lift  count
## [1] {married=NO,
##      save_act=YES,
##      mortgage=NO}       => {pep=YES} 0.1066667  0.7441860 1.629604    64
## [2] {married=NO,
##      current_act=YES,
##      mortgage=NO}       => {pep=YES} 0.1216667  0.7156863 1.567196    73
## [3] {married=NO,
##      mortgage=NO}       => {pep=YES} 0.1533333  0.7076923 1.549691    92
## [4] {income=Medium,
##      children=YES,
##      current_act=YES} => {pep=YES} 0.1033333  0.6200000 1.357664    62
## [5] {children=YES,
##      save_act=YES,
##      current_act=YES,
##      mortgage=NO}       => {pep=YES} 0.1250000  0.6000000 1.313869    75
```

Lets look the top rule again. It has support of 0.1, confidence of 0.74, & lift of 1.62. It's interesting to look at, because we can see what type of people are most likely to buy PEP. It seems like single people with saving account, & with no mortgage are most likely to buy PEP. To further understand why is this, there's a need of further analysis just like the rules mentioned before. The results of further analysis could be used to improve the plan to increase the willingnesss to buy PEP of the group of people who are unlikely to. The company could also reach out to noncurrent customer who has similar characteristics like this group of customers. They might also be likely to buy PEP.

Another interesting rule to look at is the 4th one above. It has support of 0.1, confidence of 0.62, & lift of 1.36. It's interesting becuase it is quite different LHS than the one mentioned before this, and this group of people are also likely to buy PEP. This group of people are the middle class people with kids and have current account. Again we can study further more why these people are likely to buy PEP, and the results could be used to improve the plan.

And again, the company could reach out to the group of noncustomer with similar characteristics.