Intro to Data Science - HW 10


```
# Rutwik Ghag
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
```

Association mining can be applied to many data problems beyond the well-known example of **finding relationships between different products in customer shopping data**. In this homework assignment, we will explore **real data** from the banking sector and look for **patterns associated with the likelihood of responding positively to a direct marketing campaign and signing up for a term deposit with the bank (stored in the variable "y")**.
You can find out more about the variables in this dataset here: https://archive.ics.uci.edu/ml/datasets/bank+marketing

Part 1: Explore Data Set

A. Read the contents of the following URL to a dataframe called **bank**
https://intro-datascience.s3.us-east-2.amazonaws.com/bank-full.csv

Hint: Even though this is a .csv file, chances are R won't be able to read it in correctly using the read_csv() function. If you take a closer look at the contents of the URL file, you may notice each field is separated by a **semicolon** (;) rather than a comma.

In situations like this, consider using either read.csv or read.table, with two additional parameters. *sep=";"* defines how the data is seperated (the default is a comma), and *header=TRUE* defines that there is a header line in the dataset.

```
urll <- 'https://intro-datascience.s3.us-east-2.amazonaws.com/bank-full.csv'
bank <- read.csv(urll, sep = ";",header = TRUE)
```

Make sure there are **41,188** rows and **21** columns in your **bank** df.

```
summary(bank)
```

```
##       age            job              marital            education        
##  Min.   :17.00   Length:41188       Length:41188       Length:41188      
##  1st Qu.:32.00   Class :character   Class :character   Class :character  
##  Median :38.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :40.02                                                            
##  3rd Qu.:47.00                                                            
##  Max.   :98.00                                                            
##    default            housing              loan             contact         
##  Length:41188       Length:41188       Length:41188       Length:41188      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                            
##                                                                            
##                                                                            
##     month            day_of_week           duration         campaign     
##  Length:41188       Length:41188       Min.   :   0.0   Min.   : 1.000  
##  Class :character   Class :character   1st Qu.: 102.0   1st Qu.: 1.000  
##  Mode  :character   Mode  :character   Median : 180.0   Median : 2.000  
##                                        Mean   : 258.3   Mean   : 2.568  
##                                        3rd Qu.: 319.0   3rd Qu.: 3.000  
##                                        Max.   :4918.0   Max.   :56.000  
##      pdays          previous       poutcome           emp.var.rate     
##  Min.   :  0.0   Min.   :0.000   Length:41188       Min.   :-3.40000  
##  1st Qu.:999.0   1st Qu.:0.000   Class :character   1st Qu.:-1.80000  
##  Median :999.0   Median :0.000   Mode  :character   Median : 1.10000  
##  Mean   :962.5   Mean   :0.173                      Mean   : 0.08189  
##  3rd Qu.:999.0   3rd Qu.:0.000                      3rd Qu.: 1.40000  
##  Max.   :999.0   Max.   :7.000                      Max.   : 1.40000  
##  cons.price.idx  cons.conf.idx     euribor3m      nr.employed  
##  Min.   :92.20   Min.   :-50.8   Min.   :0.634   Min.   :4964  
##  1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.344   1st Qu.:5099  
##  Median :93.75   Median :-41.8   Median :4.857   Median :5191  
##  Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5167  
##  3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228  
##  Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5228  
##       y            
##  Length:41188      
##  Class :character  
##  Mode  :character  
##                    
##                    
## 
```

```
nrow(bank)
```

```
## [1] 41188
```

```
ncol(bank)
```

```
## [1] 21
```

```
# we used nrow and ncol to find the number of rows and columns
```

B. Next, we will focus on some key factor variables from the dataset, and convert a few numeric ones to factor variables. Execute the following command. Write a comment describing how the conversion for each numeric variable works and what are the variables in the resulting dataframe.

```
bank_new <- data.frame(job=as.factor(bank$job),
                     marital=as.factor(bank$marital),
                     housing_loan=as.factor(bank$housing),
                     young=as.factor(bank$age<median(bank$age)),
                     contacted_more_than_once=as.factor(bank$campaign>1),
                     contacted_before_this_campaign=as.factor(bank$previous<0),
                     success=as.factor(bank$y))
```

C. Count the number of successful term deposit sign-ups, using the table( ) command on the **success** variable.

```
table(bank_new$success)
```

```
## 
##    no   yes 
## 36548  4640
```

```
# number of successful term deposits are 4640
```

D. Express the results of problem C as percentages by sending the results of the table( ) command into the prop.table( ) command.

```
dim(table(bank_new$success))
```

```
## [1] 2
```

```
prop.table(table(bank_new$success))*100
```

```
## 
##       no      yes 
## 88.73458 11.26542
```

```
proportions(table(bank_new$success))*100
```

```
## 
##       no      yes 
## 88.73458 11.26542
```

E. Using the same techniques, show the percentages for the **marital** and **housing_loan** variables as well.

```
# multiplying by 100 to express the percentage from the proportions
proportions(table(bank_new$marital))*100
```

```
## 
##   divorced    married     single    unknown 
## 11.1974361 60.5224823 28.0858502  0.1942313
```

```
proportions(table(bank_new$housing_loan))*100
```

```
## 
##        no   unknown       yes 
## 45.212198  2.403613 52.384190
```

Part 2: Coerce the data frame into transactions

F. Install and library two packages: **arules** and **arulesViz**.

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

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

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

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

```
#install.packages("arulesViz")
library(arulesViz)
```

G. Coerce the **bank_new** dataframe into a **sparse transactions matrix** called **bankX**.

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

```
## transactions as itemMatrix in sparse format with
##  41188 rows (elements/itemsets/transactions) and
##  26 columns (items) and a density of 0.2692308 
## 
## most frequent items:
## contacted_before_this_campaign=FALSE                           success=no 
##                                41188                                36548 
##                      marital=married        contacted_more_than_once=TRUE 
##                                24928                                23546 
##                     housing_loan=yes                              (Other) 
##                                21576                               140530 
## 
## element (itemset/transaction) length distribution:
## sizes
##     7 
## 41188 
## 
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       7       7       7       7       7       7 
## 
## includes extended item information - examples:
##             labels variables       levels
## 1       job=admin.       job       admin.
## 2  job=blue-collar       job  blue-collar
## 3 job=entrepreneur       job entrepreneur
## 
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

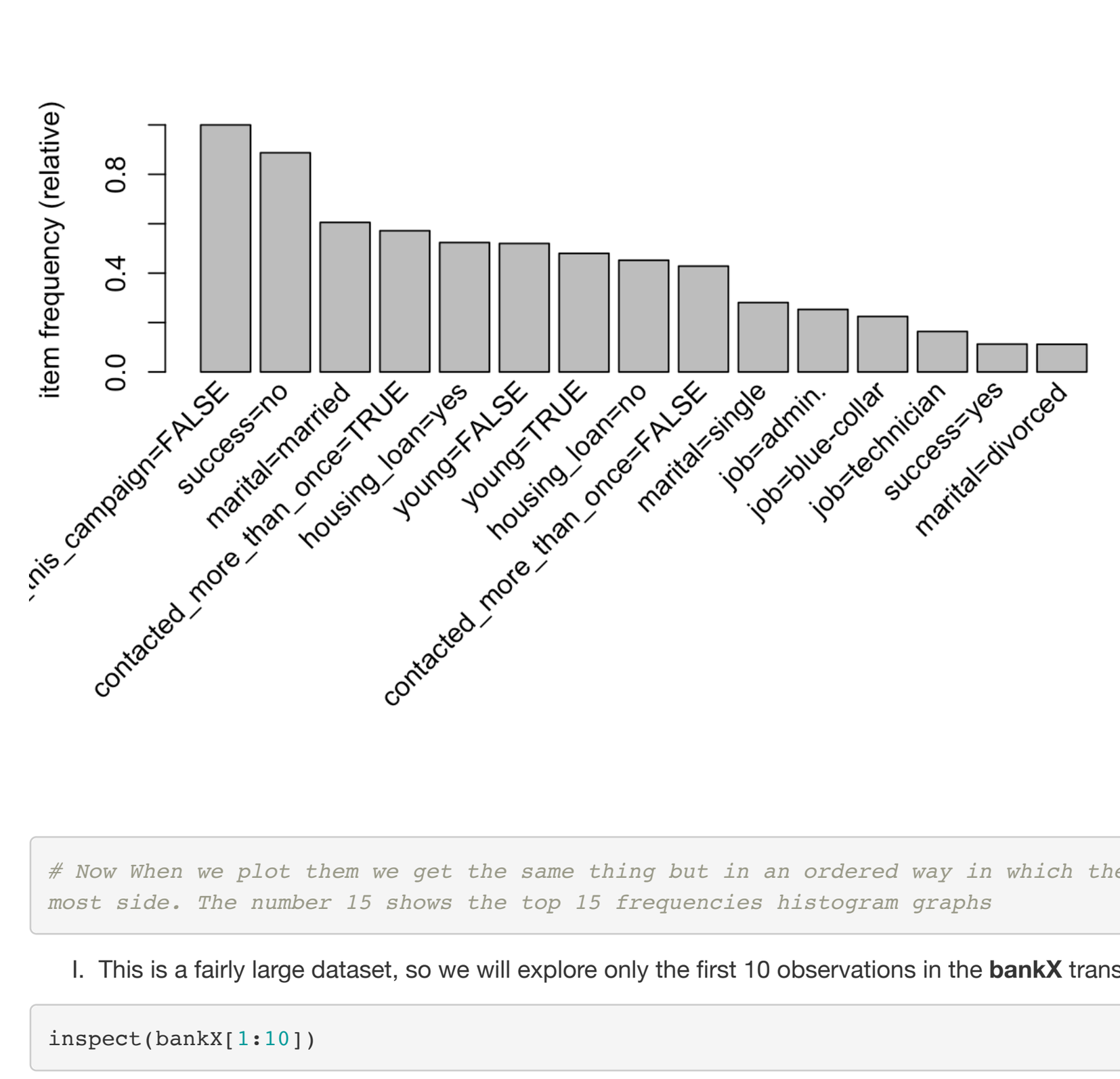
H. Use the itemFrequency( ) and itemFrequencyPlot( ) commands to explore the contents of **bankX**. What do you see?

```
itemFrequency(bankX)
```

```
##                           job=admin.                      job=blue-collar 
##                          0.253034865                          0.224677090 
##                     job=entrepreneur                        job=housemaid 
##                          0.035350102                          0.025735651 
##                       job=management                          job=retired 
##                          0.070991551                          0.041759736 
##                    job=self-employed                         job=services 
##                          0.034500340                          0.096363018 
##                          job=student                       job=technician 
##                          0.021244052                          0.163712732 
##                       job=unemployed                          job=unknown 
##                          0.024618821                          0.008012042 
##                     marital=divorced                      marital=married 
##                          0.111974361                          0.605224823 
##                       marital=single                      marital=unknown 
##                          0.280858502                          0.001942313 
##                      housing_loan=no                 housing_loan=unknown 
##                          0.452121977                          0.024036127 
##                     housing_loan=yes                          young=FALSE 
##                          0.523841896                          0.520054385 
##                           young=TRUE       contacted_more_than_once=FALSE 
##                          0.479945615                          0.428328639 
##        contacted_more_than_once=TRUE contacted_before_this_campaign=FALSE 
##                          0.571671361                          1.000000000 
##                           success=no                          success=yes 
##                          0.887345829                          0.112654171
```

```
# In the item frequency plot I see frequency of items in a particular column as a proportion of the total number of items present in that respective column

itemFrequencyPlot(bankX,topN = 15)
```




```
# Now When we plot them we get the same thing but in an ordered way in which the highest frequency is at the left most side. The number 15 shows the top 15 frequencies histogram graphs
```

I. This is a fairly large dataset, so we will explore only the first 10 observations in the **bankX** transaction matrix:

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

```
##      items                                  transactionID
## [1]  {job=housemaid,                                     
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       1
## [2]  {job=services,                                      
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       2
## [3]  {job=services,                                      
##       marital=married,                                   
##       housing_loan=yes,                                  
##       young=TRUE,                                        
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       3
## [4]  {job=admin.,                                        
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       4
## [5]  {job=services,                                      
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       5
## [6]  {job=services,                                      
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       6
## [7]  {job=admin.,                                        
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       7
## [8]  {job=blue-collar,                                   
##       marital=married,                                   
##       housing_loan=no,                                   
##       young=FALSE,                                       
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       8
## [9]  {job=technician,                                    
##       marital=single,                                    
##       housing_loan=yes,                                  
##       young=TRUE,                                        
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                       9
## [10] {job=services,                                      
##       marital=single,                                    
##       housing_loan=yes,                                  
##       young=TRUE,                                        
##       contacted_more_than_once=FALSE,                    
##       contacted_before_this_campaign=FALSE,              
##       success=no}                                      10
```

Explain the difference between **bank_new** and **bankX** in a block comment:

```
#install.packages("XQuartz")
View(bank_new)
#View(bankX)
summary(bank_new)
```

```
##           job            marital       housing_loan     young      
##  admin.     :10422   divorced: 4612   no     :18622   FALSE:21420  
##  blue-collar: 9254   married :24928   unknown:  990   TRUE :19768  
##  technician : 6743   single  :11568   yes    :21576                
##  services   : 3969   unknown :   80                                
##  management : 2924                                                 
##  retired    : 1720                                                 
##  (Other)    : 6156                                                 
##  contacted_more_than_once contacted_before_this_campaign success   
##  FALSE:17642              FALSE:41188                    no :36548  
##  TRUE :23546                                             yes: 4640  
##                                                                     
##                                                                     
##                                                                     
##                                                                     
## 
```

```
#summary(bankX)

# bankX is a sparse transaction matrix whereas bank_new is the dataframe. Though the data might be same in bankX and bank_new since bank_new was created from the columns of bank but first converting them in factor values, how the values are represented is different. Using summary in them shows us the different way of representation gives us different analysis. When using View we can see that bankX is more like an object whereas bank_new is a databa se with entries
```

Part 3: Use arules to discover patterns

Support is the proportion of times that a particular set of items occurs relative to the whole dataset.
Confidence is proportion of times that the consequent occurs when the antecedent is present.

J. Use **apriori** to generate a set of rules with support over 0.005 and confidence over 0.3, and trying to predict who successfully signed up for a term deposit.
Hint: You need to define the **right-hand side rule (rhs)**.

```
ruleset <- apriori(bankX,parameter = list(supp= 0.0051, conf = 0.31),control=list(verbose=F), appearance = list(d
efault="lhs",rhs=("success=yes")))
# bankX is our sparse transaction matrix. I used values slighly above or else I was not getting any rules
```

K. Use inspect() to review of the **ruleset**.

```
inspectDT(ruleset)
```

Show 10 entries Search:

	LHS	RHS	support	confidence	coverage	lift	count
	All	All		All	A		
[1]	{job=student}	{success=yes}	0.007	0.314	0.021	2.790	275.000
[2]	{job=student,marital=single}	{success=yes}	0.006	0.320	0.020	2.844	264.000
[3]	{job=student,young=TRUE}	{success=yes}	0.007	0.318	0.021	2.823	271.000
[4]	{job=student,contacted_before_this_campaign=FALSE}	{success=yes}	0.007	0.314	0.021	2.790	275.000
[5]	{job=student,marital=single,young=TRUE}	{success=yes}	0.006	0.323	0.020	2.871	260.000
[6]	{job=student,marital=single,contacted_before_this_campaign=FALSE}	{success=yes}	0.006	0.320	0.020	2.844	264.000
[7]	{job=student,young=TRUE,contacted_before_this_campaign=FALSE}	{success=yes}	0.007	0.318	0.021	2.823	271.000
[8]	{job=student,marital=single,young=TRUE,contacted_before_this_campaign=FALSE}	{success=yes}	0.006	0.323	0.020	2.871	260.000

Showing 1 to 8 of 8 entries Previous 1 Next

L. Use the output of inspect( ) or inspectDT( ) and describe **any 2 rules** the algorithm found.

```
inspect(ruleset[c(5)])
```

```
##     lhs                  rhs               support confidence   coverage     lift count
## [1] {job=student,                                                                      
##      marital=single,                                                                   
##      young=TRUE}      => {success=yes} 0.006312518  0.3233831 0.01952025 2.870582   260
```

```
# these 2 rules set have the same support , confidence, coverage,lift and count. But yet they are somewhat differ
ent scenarios
# LHS for the first one is {job=student,marital=single,young=TRUE}
# LHS for the second one is {job=student,marital=single,young=TRUE,contacted_before_this_campaign=FALSE}}
# thus one can say if we want the scenario who successfully signed up for a term deposit, there is no need to co
ntact them before the campaign for this particular scenario
inspect(ruleset[c(8)])
```

```
##     lhs                                       rhs               support confidence   coverage     lift count
## [1] {job=student,                                                                                           
##      marital=single,                                                                                        
##      young=TRUE,                                                                                            
##      contacted_before_this_campaign=FALSE} => {success=yes} 0.006312518  0.3233831 0.01952025 2.870582   260
```