Project 2: Credit Card Lead Prediction

Haocheng Liao, KAIQI PENG, Wenting Yang, Danlei Wang, Yucong Chen

Contents

1. Introduction

Happy Customer Bank is a mid-sized private bank that operates a variety of banking products, such as savings accounts, current accounts, investment products, credit products, etc. It also cross sells products to existing customers through different means of communication. In this case, the bank hopes to cross sell its credit cards to existing customers.

Our goal of this project is to help Happy Customer Bank to conduct a credit card lead prediction - to identify customers who have higher intention to recommended credit cards from a qualified group identified by the bank - through data analysis and establishing logistic and probit models as well as other related models.

2. Description of the Data

Happy Customer Bank has collected information from 105,312 customers and put the information into a dataset named "test.csv". For each observation in the dataset, there are eight variables relevant for our models: "Gender", "Age", "Occupation", "Channel_Code", "Vintage", "Credit_Product", "Avg_Account_Balance", and "Is_Active". The two variables "ID" and "Region_Code" are irrelevant because they serve more as identifications for customers, or in other words, they were to show us that every observation is unique. Thus, they will be deleted during our data analysis and modeling, but will be used after we move on to identify potential customers.

For the eight relevant variables, "Age", "Vintage", and "Avg_Account_Balance" are numerical variable. "Age" refers to the age of the customer. "Vintage" refers to the vintage for the customer in months. "Avg_Account_Balance" refers to the average account balance in the last 12 months.

"Gender", "Occupation", "Channel_Code", "Credit_Product", and "Is_Active" are categorical variables, so we will encode them into numbers. Among them, "Gender" and "Is_Active" take just two values, so they are dummy variables. "Gender" refers to the gender of the customer and it takes the value of "Female" or "Male". We will let "Female"=1 and "Male"=0. "Is_Active" describes if the customer is active in last three months and it takes the value of "Yes" or "No". We will let "Yes"=1 and "No"=0. We will take "Is_Active" as our dependent variable.

"Occupation" refers to the job of the customer. There are four different kinds of jobs: "Salaried", "Self-Employed", "Entrepreneur", and "Other". We will let "Other"=1, "Salaried"=2, "Self_Employed"=3, and "Entrepreneur"=4.

"Channel_Code" is the encoded acquisition channel code for the customer. There are also four different kinds: "X1", "X2", "X3", and "X4". We will model "X1"=1, "X2"=2, "X3"=3, and "X4"=4.

"Credit_Product" refers to if the customer has any active credit product, such as home loan, personal loan, credit card, etc. It takes the value of "Yes", "No", or "N/A". We will model "No"=0, "Yes"=1, "N/A"=2.

3. Data Analysis / Model

Data Analysis

In this section, we will visualize our variables and encode the categorical variables into numerical ones to prepare for our models.

```
# load the data test.csv
library(readxl)
library(readr)
data <- read_csv("test.csv")
```

```
## Parsed with column specification:
## cols(
##   ID = col_character(),
##   Gender = col_character(),
##   Age = col_double(),
##   Region_Code = col_character(),
##   Occupation = col_character(),
##   Channel_Code = col_character(),
##   Vintage = col_double(),
##   Credit_Product = col_character(),
##   Avg_Account_Balance = col_double(),
##   Is_Active = col_character()
## )
```

Histograms & Fitted Distribution & Pie Charts

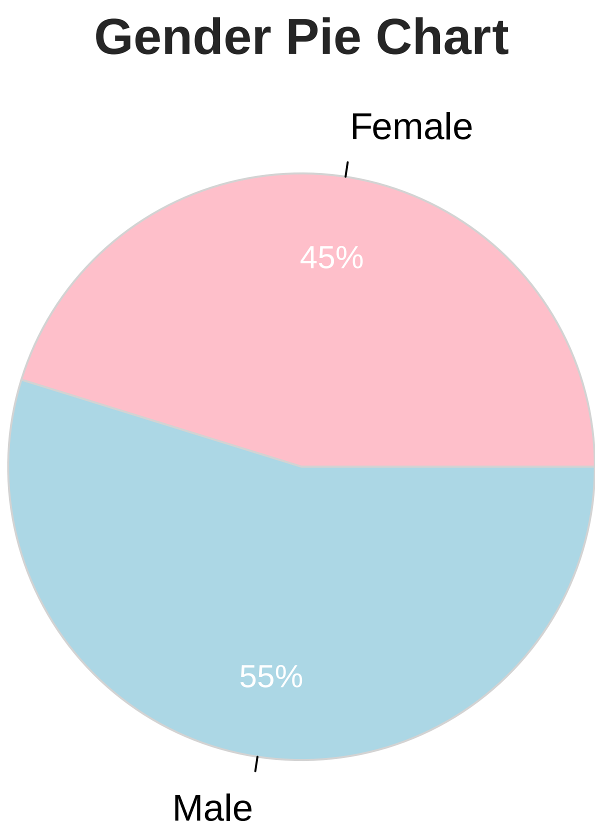
```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
attach(data)
library(lessR)
```

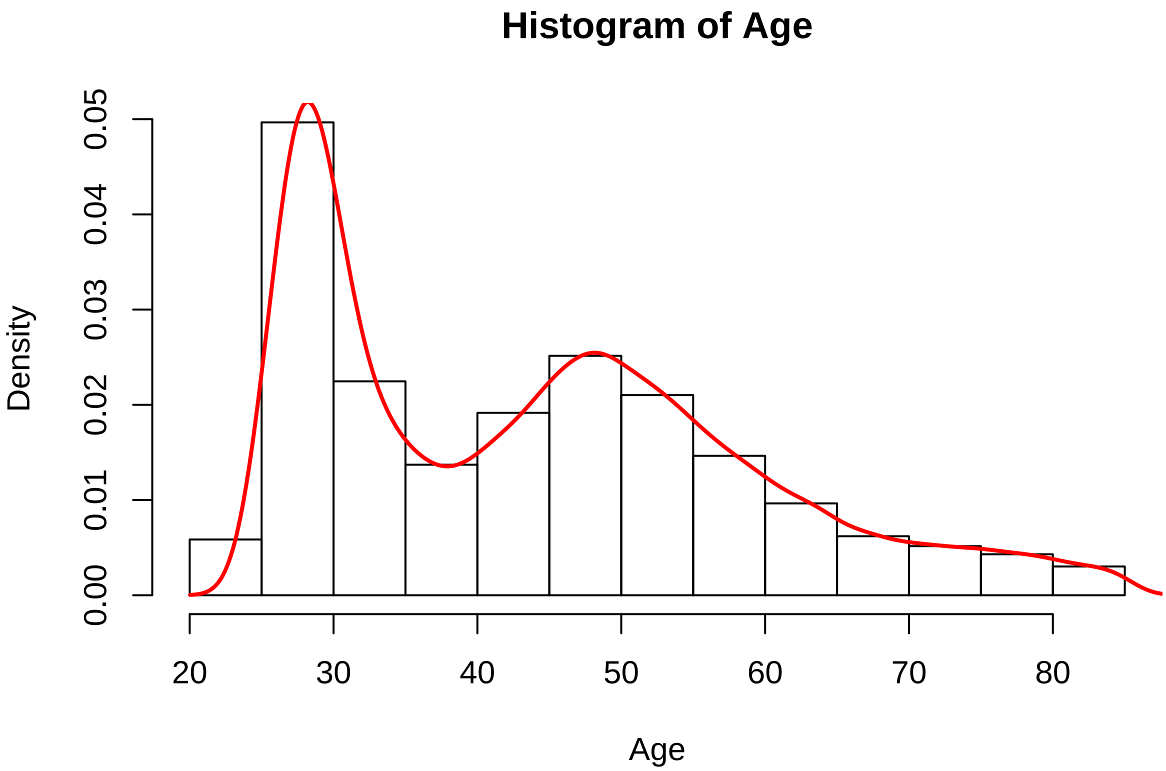
```
##
## lessR 3.9.9  feedback: gerbing@pdx.edu  web: lessRstats.com/new
## ---------------------------------------------------------------
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Many vignettes show by example how to use lessR. Topics are
## read, write, & manipulate data, graphics, means & models,
##   factor analysis, & customization. Two ways to view.
## Enter:  browseVignettes("lessR")
## Visit:  https://CRAN.R-project.org/package=lessR
```

```
# Gender
PieChart(Gender, hole = 0, values = "%", data = data,
         fill = c("pink","lightblue"), main = "Gender Pie Chart")
```

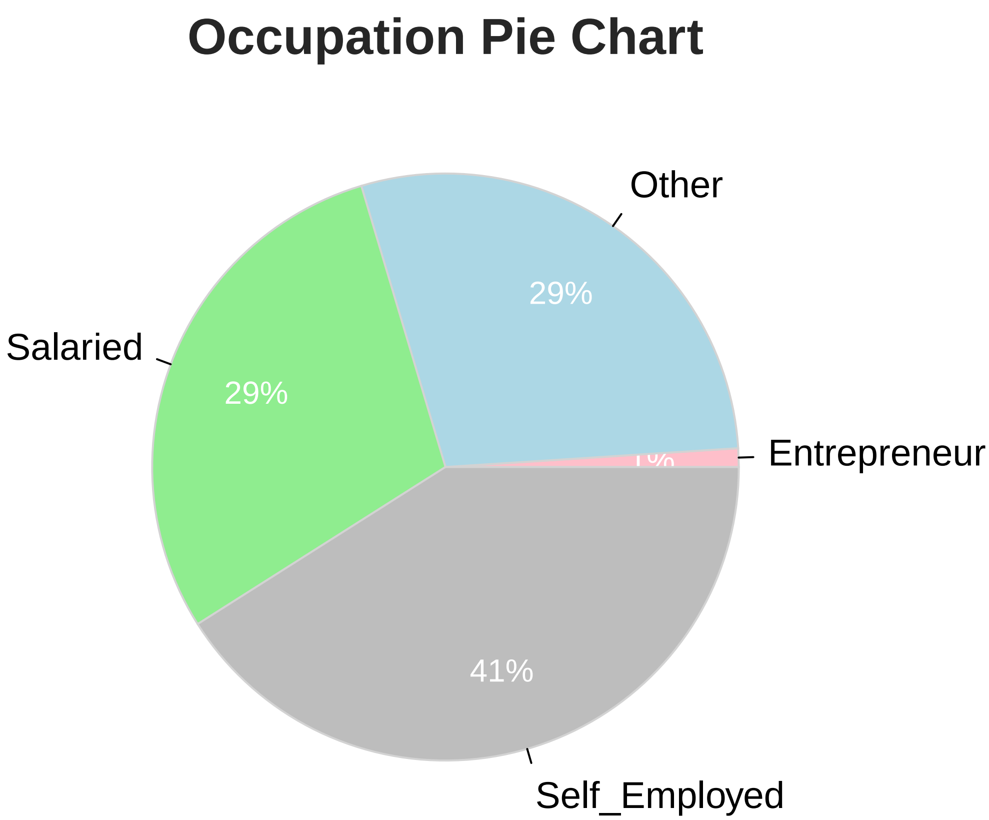



```
## >>> Suggestions
## PieChart(Gender, hole=0)  # traditional pie chart
## PieChart(Gender, values="%")  # display %'s on the chart
## BarChart(Gender)  # bar chart
## Plot(Gender)  # bubble plot
## Plot(Gender, values="count")  # lollipop plot
##
##
## --- Gender ---
##
##
##                 Female   Male        Total
## Frequencies:     47607  57705       105312
## Proportions:     0.452  0.548        1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 968.262, df = 1, p-value = 0.000
```

```
# 55% of the customers are male while 45% of them are female.


# Age
hist(Age,prob=TRUE)
lines(density(Age), lwd = 2, col ="red")
```



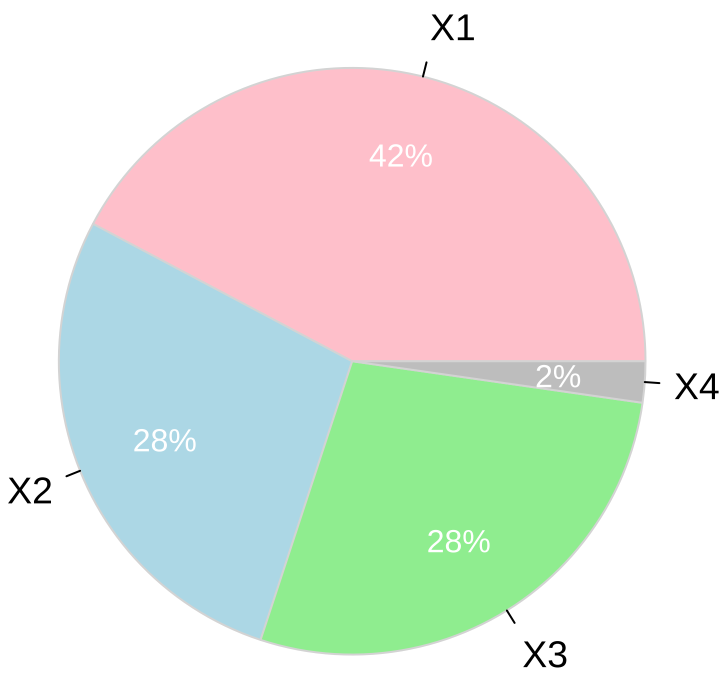

```
# By plotting the histogram of “Age”,
# we observe that most people’s age falls in the interval [20,80].
# However, people around 25 to 30 years old are the largest client crowd of Happy Bank.


# Occupation
PieChart(Occupation, hole = 0, values = "%", data = data,
         fill = c("pink","lightblue", "lightgreen", "grey"),
         main = "Occupation Pie Chart")
```




```
## >>> Suggestions
## PieChart(Occupation, hole=0)  # traditional pie chart
## PieChart(Occupation, values="%")  # display %'s on the chart
## BarChart(Occupation)  # bar chart
## Plot(Occupation)  # bubble plot
## Plot(Occupation, values="count")  # lollipop plot
##
##
## --- Occupation ---
##
##
##                Entrepreneur  Other  Salaried  Self_Employed        Total
## Frequencies:           1076  30131     30913          43192       105312
## Proportions:          0.010  0.286     0.294          0.410        1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 36369.760, df = 3, p-value = 0.000
```

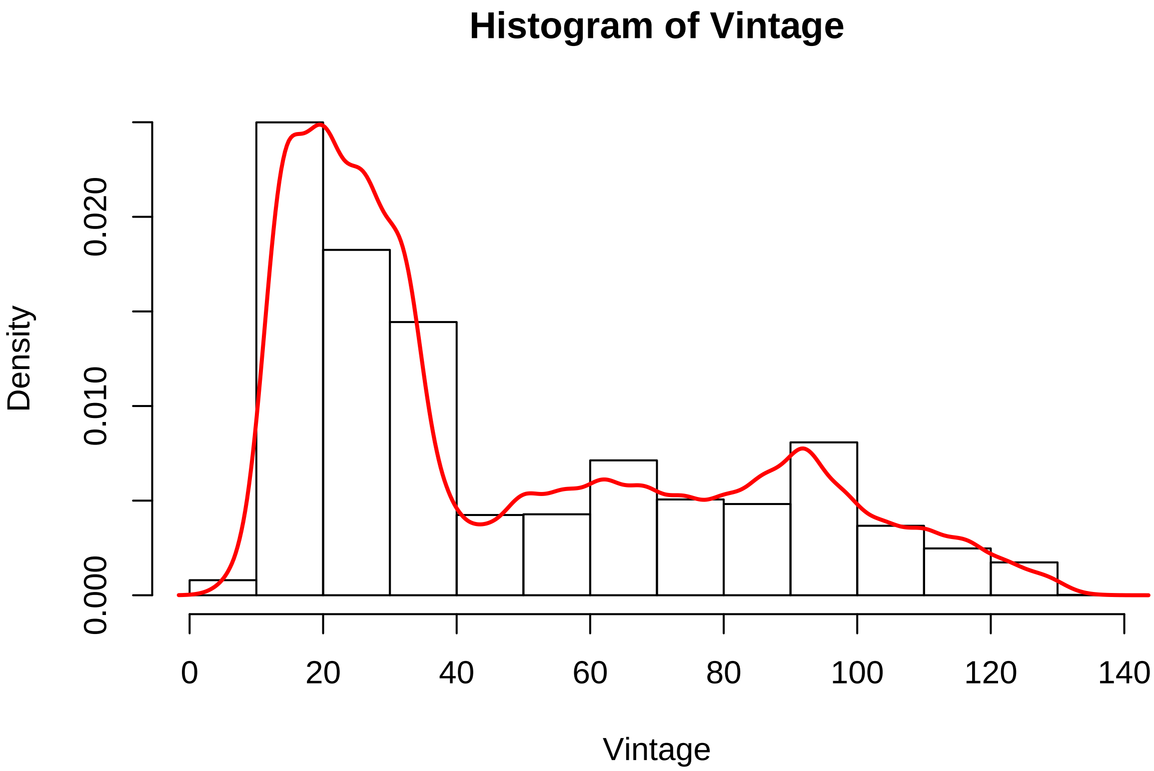
```
# 41% of customers are self-employed, 29% are salaried, 1% are entrepreneurs, and 29% are others.


# Channel Code
PieChart(Channel_Code, hole = 0, values = "%", data = data,
         fill = c("pink","lightblue", "lightgreen", "grey"),
         main = "Channel Code Pie Chart")
```

Channel Code Pie Chart

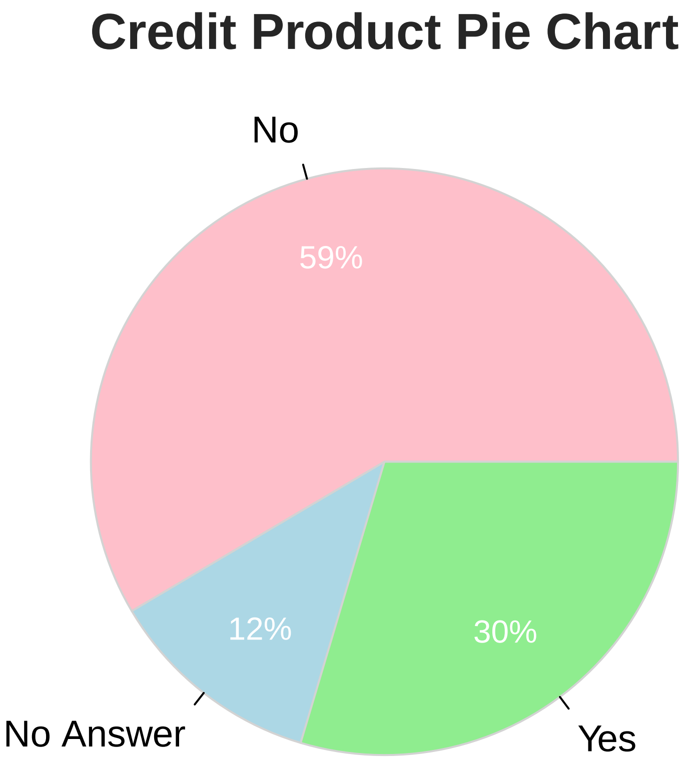



```
## >>> Suggestions
## PieChart(Channel_Code, hole=0)  # traditional pie chart
## PieChart(Channel_Code, values="%")  # display %'s on the chart
## BarChart(Channel_Code)  # bar chart
## Plot(Channel_Code)  # bubble plot
## Plot(Channel_Code, values="count")  # lollipop plot
##
##
## --- Channel_Code ---
##
##
##                    X1     X2     X3     X4        Total
## Frequencies:    44484  29176  29269   2383       105312
## Proportions:    0.422  0.277  0.278  0.023        1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 34934.820, df = 3, p-value = 0.000
```

```
# 42% of channel codes are X1, 28% are X2, 28% are X3, and 2% are X4.


# Vintage
hist(Vintage,prob=TRUE)
lines(density(Vintage), lwd = 2, col ="red")
```

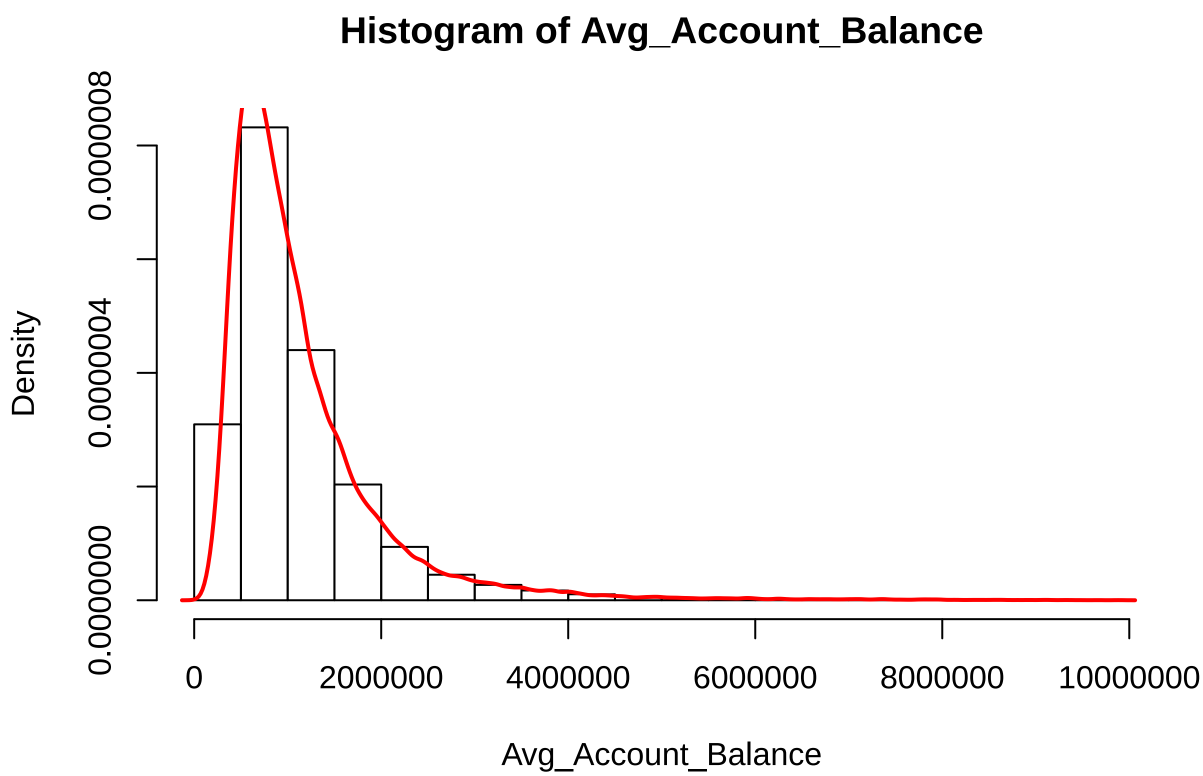



```
# The histogram of Vintage indicates that the vintage for most clients falls into the interval [0,140].


# Credit Product
# We change "N/A" to "No Answer" because N/As are not counted in pie charts
data$Credit_Product[is.na(data$Credit_Product)] <- "No Answer"
PieChart(Credit_Product, hole = 0, values = "%", data = data,
         fill = c("pink","lightblue", "lightgreen"),
         main = "Credit Product Pie Chart")
```

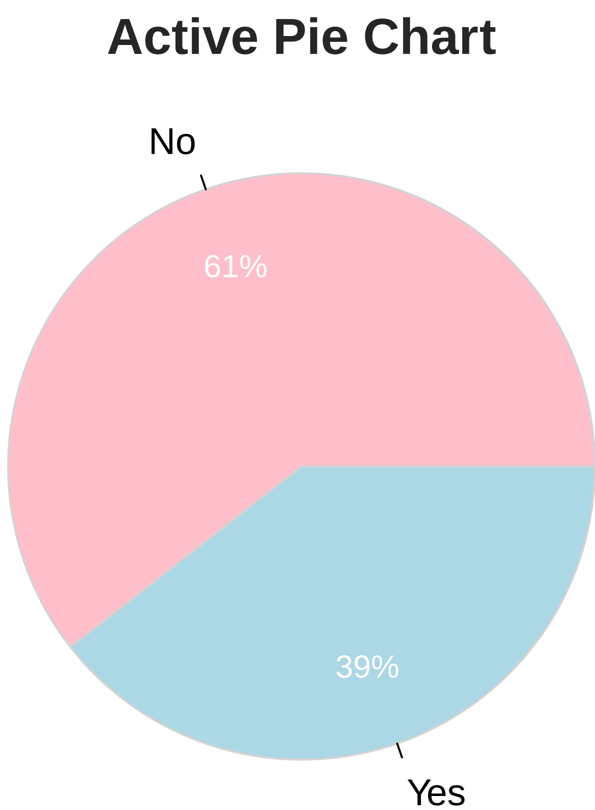



```
## >>> Suggestions
## PieChart(Credit_Product, hole=0)  # traditional pie chart
## PieChart(Credit_Product, values="%")  # display %'s on the chart
## BarChart(Credit_Product)  # bar chart
## Plot(Credit_Product)  # bubble plot
## Plot(Credit_Product, values="count")  # lollipop plot
##
##
## --- Credit_Product ---
##
##
##                    No  No Answer    Yes        Total
## Frequencies:    61608      12522  31182       105312
## Proportions:    0.585      0.119  0.296        1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 34975.810, df = 2, p-value = 0.000
```

```
# 30% of the customers answered Yes, 59% answered No, and 12% preferred not to answer.

# Average Account Balance
hist(Avg_Account_Balance,prob=TRUE)
lines(density(Avg_Account_Balance), lwd = 2, col ="red")
```




```r
# By plotting the histogram of Avg_Account_Balance,
# we observe that the average account balance of clients falls into the interval [0,4000000],
# while only a few clients have a balance that is above 4,000,000.
# This indicates that the financial resources of all the clients vary greatly.


# Is active
PieChart(Is_Active, hole = 0, values = "%", data = data,
         fill = c("pink","lightblue"),
         main = "Active Pie Chart")
```




```
## >>> Suggestions
## PieChart(Is_Active, hole=0)  # traditional pie chart
## PieChart(Is_Active, values="%")  # display %'s on the chart
## BarChart(Is_Active)  # bar chart
## Plot(Is_Active)  # bubble plot
## Plot(Is_Active, values="count")  # lollipop plot
##
##
## --- Is_Active ---
##
##
##                    No    Yes        Total
## Frequencies:    63797  41515       105312
## Proportions:    0.606  0.394        1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 4714.444, df = 1, p-value = 0.000
```

```
# 39% of the customers answered Yes while 61% of them answered No.
```

```
# Encode the categorical variables
data$Gender[data$Gender == "Female"] <- 1
data$Gender[data$Gender == "Male"] <- 0
data$Gender <- gsub(",", "", data$Gender)
data$Gender <- as.numeric(as.character(data$Gender))

data$Channel_Code[data$Channel_Code == "X1"] <- 1
data$Channel_Code[data$Channel_Code == "X2"] <- 2
```

```r
data$Channel_Code[data$Channel_Code == "X3"] <- 3
data$Channel_Code[data$Channel_Code == "X4"] <- 4
data$Channel_Code <- gsub(",", "", data$Channel_Code)
data$Channel_Code <- as.numeric(as.character(data$Channel_Code))

data$Occupation[data$Occupation == "Other"] <- 1
data$Occupation[data$Occupation == "Salaried"] <- 2
data$Occupation[data$Occupation == "Self_Employed"] <- 3
data$Occupation[data$Occupation == "Entrepreneur"] <- 4
data$Occupation <- gsub(",", "", data$Occupation)
data$Occupation <- as.numeric(as.character(data$Occupation))

data$Credit_Product[data$Credit_Product == "No Answer"] <- 1
data$Credit_Product[data$Credit_Product == "Yes"] <- 2
data$Credit_Product[data$Credit_Product == "No"] <- 3
data$Credit_Product <- gsub(",", "", data$Credit_Product)
data$Credit_Product <- as.numeric(as.character(data$Credit_Product))

data$Is_Active[data$Is_Active == "Yes"] <- 1
data$Is_Active[data$Is_Active == "No"] <- 0
data$Is_Active <- gsub(",", "", data$Is_Active)
data$Is_Active <- as.numeric(as.character(data$Is_Active))

data <- data[, -1] # Delete column "ID"
data <- data[, -3] # Delete column "Region_Code"

sapply(data, class) # Check all columns are numeric now
```

```
##            Gender                Age         Occupation       Channel_Code
##         "numeric"          "numeric"          "numeric"          "numeric"
##           Vintage     Credit_Product Avg_Account_Balance          Is_Active
##         "numeric"          "numeric"          "numeric"          "numeric"
```

Correlation

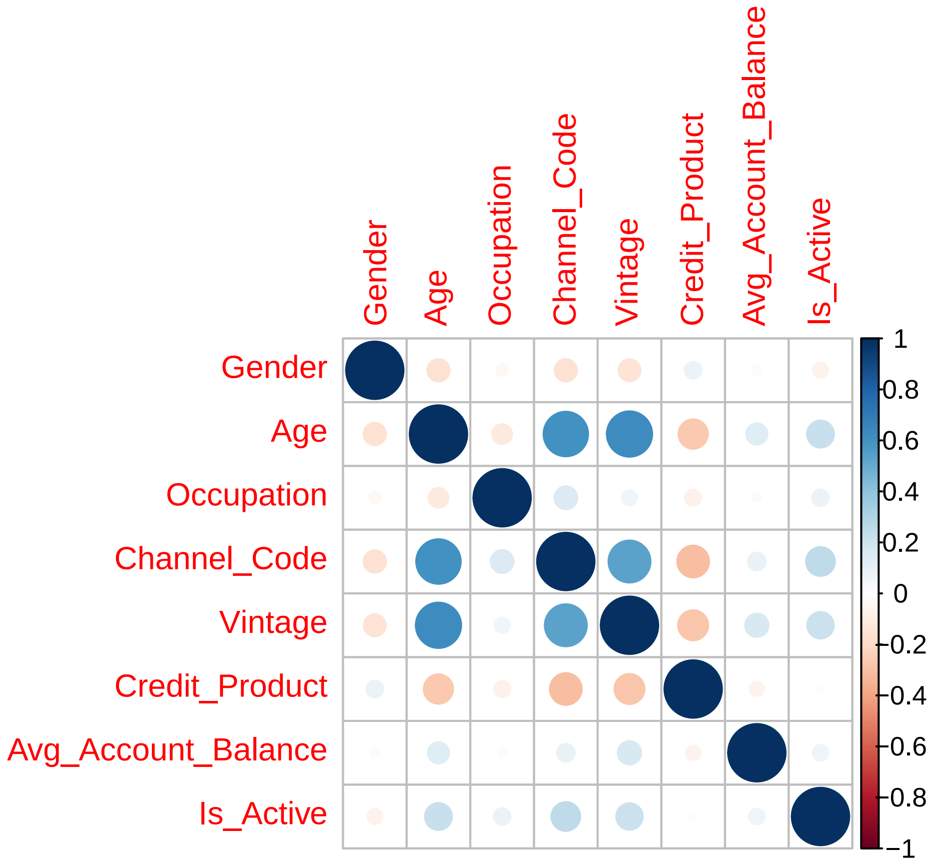
```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
attach(data)
```

```
## The following objects are masked from data (pos = 5):
##
##     Age, Avg_Account_Balance, Channel_Code, Credit_Product, Gender,
##     Is_Active, Occupation, Vintage
```

```r
corrplot(cor(data))
```



Comment: We have a correlation matrix here that shows the correlation coefficients between the following variables: Gender, Age, Occupation, Channel_Code, Vintage, Credit_Prodct, Avg_Account_Balance and Is_Active. Each cell in the matrix indicates the correlation between two specific variables. The color index on the right measures the level of association between the two variables from -1 to 1. Notice that the correlation coefficients along the diagonal of the matrix are all dark blue(=1) because each variable has a perfect uphill linear relationship with itself.

It is quite clear that some variables are very strongly and positively correlated. For example, "Channel_Code" & "Age", "Vintage" & "Age". Their values of correlation are all larger or close to 0.8 which means there is a very strong positive linear relationship between each two.

Furthermore, we have some other variables that are strongly and positively correlated, but not as much as above. For instance, "Channel_Code" & "Vintage". Their values of correlation are close to 0.7 which means there is a strong uphill linear relationship between each two.

The matrix also shows us some variables that are basically not correlated, or not correlated at all with another. For examples, "Gender" & "Age", "Gender" & "Channel_Code", "Gender" & "Vintage", "Gender" & "Occupation", "Gender" & "Credit Product", "Gender" & "Avg_Account_Balance", "Gender" & "Is_Active", "Age" & "Occupation", "Age" & "Avg_Account_Balance", "Occupation" & "Channel_Code", "Occupation" & "Vintage", "Occupation" & "Credit_Product", "Occupation" & "Avg_Account_Balance", "Occupation" & "Is_Active", "Channel_Code" & "Avg_Account_Balance", "Vintage" & "Occupation", "Channel_Code" & "Channel_Code" & "Avg_Account_Balance". The correlations between them are close or equal to zero which indicates very small or no linear correlation at all.

Last but not least, there are variables that have a weak negative linear relationship with another variable: "Age" & "Credit_Product", "Age" & "Is_Active", "Channel_Code" & "Credit_Product", "Channel_Code" & "Is_Active", "Vintage" & "Credit_Product", "Vintage" & "Is_Active". The correlations are close to -0.3 which prove that they are weakly negatively correlated with the other.

Statistical Summary

```r
summary(data)
```

```
##      Gender            Age          Occupation     Channel_Code  
##  Min.   :0.0000   Min.   :24.00   Min.   :1.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:30.00   1st Qu.:1.000   1st Qu.:1.000  
##  Median :0.0000   Median :43.00   Median :2.000   Median :2.000  
##  Mean   :0.4521   Mean   :43.87   Mean   :2.144   Mean   :1.901  
##  3rd Qu.:1.0000   3rd Qu.:54.00   3rd Qu.:3.000   3rd Qu.:3.000  
##  Max.   :1.0000   Max.   :85.00   Max.   :4.000   Max.   :4.000  
##     Vintage       Credit_Product  Avg_Account_Balance   Is_Active     
##  Min.   :  7.00   Min.   :1.000   Min.   :  22597     Min.   :0.0000  
##  1st Qu.: 20.00   1st Qu.:2.000   1st Qu.: 603982     1st Qu.:0.0000  
##  Median : 32.00   Median :3.000   Median : 896634     Median :0.0000  
##  Mean   : 46.84   Mean   :2.466   Mean   :1134195     Mean   :0.3942  
##  3rd Qu.: 73.00   3rd Qu.:3.000   3rd Qu.:1371598     3rd Qu.:1.0000  
##  Max.   :135.00   Max.   :3.000   Max.   :9908858     Max.   :1.0000
```

From our statistical summary, we notice that the minimum "Age" is 24 while the maximum is 85, so there are customers of all ages [24,85]. The median is 43.00 and the mean is 43.87 (i.e., there is not much difference), which means that our data is very unbiased. The minimum of "Vintage" is 7 while the maximum is 135, and the mean of it is 46.84. The range is very wide in this case, which means that the bank has investigated both "new" and "old" customers. The minimum of "Avg_Account_Balance" is 22597 while the maximum is 9908858, and the mean of it is 1134195. The range for this variable is wide as well.

For our dummy variables "Gender" and "Is_Active", their maximum is 1 and minimum is 0, and their means are 0.4521 and 0.3942 respectively, which indicates the distribution between females and males, and active and inactive customers are unbiased.

The minimum of "Occupation" is 1 while the maximum is 4, and the mean of it is 2.144. Note that we have four possible numbers for "Occupation" with respect to salaried, self-employed, entrepreneur, and other ("Other"=1, "Salaried"=2, "Self_Employed"=3, "Entrepreneur"=4). This indicates that our occupation distribution is close to even.

The minimum of "Channel_Code" is 1 while the maximum is 4, and the mean of it is 1.901. Note that we have four possible numbers for this variable with respect to X1, X2, X3, and X4 ("X1"=1, "X2"=2, "X3"=3, "X4"=4).

The minimum of "Credit_Product" is 1 while the maximum is 3, and the mean of it is 2.466. Note that we have three possible numbers for this variable with respect to "Yes", "No", and "NA" (NA=1, Yes=2, NO=3).

Model

In this section, we will predict customers' interests in recommended credit cards. Since this involves individuals making "either-or" choices, represented by the binary variable "Is_Active", we will start with the OLS model, probit model, and logit model. Then, we will divide the data into training and testing for evaluation. In addition, multinomial logit model and instrumental variables will be used for possible improvements. We will also provide a series of tests to find the best model and use that to identify potential customers for Happy Customer Bank.

OLS Model

```r
OLS.Model <- lm(Is_Active ~ Gender + Age + Occupation + Channel_Code + Vintage
                + Credit_Product + Avg_Account_Balance, data = data)
summary(OLS.Model)
```

```
##
## Call:
## lm(formula = Is_Active ~ Gender + Age + Occupation + Channel_Code +
##     Vintage + Credit_Product + Avg_Account_Balance, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8090 -0.3663 -0.2310  0.4848  0.9968
##
## Coefficients:
##                         Estimate      Std. Error t value
## (Intercept)         -0.299549257025  0.009612541207 -31.162
## Gender              -0.022986164668  0.002928938041  -7.848
## Age                  0.003414044624  0.000141362426  24.151
## Occupation           0.042592111664  0.001798125383  23.687
## Channel_Code         0.091150174537  0.002196421548  41.499
## Vintage              0.001197522106  0.000059795931  20.027
## Credit_Product       0.084426992521  0.002185858181  38.624
## Avg_Account_Balance  0.000000022464  0.000000001681  13.366
##                               Pr(>|t|)
## (Intercept)         < 0.0000000000000002
## Gender               0.00000000000000427
## Age                 < 0.0000000000000002
## Occupation          < 0.0000000000000002
## Channel_Code        < 0.0000000000000002
## Vintage             < 0.0000000000000002
## Credit_Product      < 0.0000000000000002
## Avg_Account_Balance < 0.0000000000000002
##
## Residual standard error: 0.4653 on 105304 degrees of freedom
## Multiple R-squared:  0.09339,    Adjusted R-squared:  0.09333
## F-statistic:  1550 on 7 and 105304 DF,  p-value: < 0.00000000000000022
```

```r
confint(OLS.Model)
```

```
##                                2.5 %            97.5 %
## (Intercept)         -0.31838970814349 -0.28070880590589
## Gender              -0.02872684372470 -0.01724548561119
## Age                  0.00313697617657  0.00369111307178
## Occupation           0.03906781016438  0.04611641316309
```

```
## Channel_Code         0.08684521792636  0.09545513114680
## Vintage              0.00108032288839  0.00131472132315
## Credit_Product       0.08014273996761  0.08871124507449
## Avg_Account_Balance  0.00000001916964  0.00000002575765
```

Comment: All the variables are statistically significant in the OLS model. The 95% confidence interval for each variable shows us that all estimators are statistically significantsince the all intervals do not contain 0. We will use this model as a benchmark.

From the model, we notice that most estimators are logical: one year increase in age will increase the probability of taking credit cards by 0.3414%; one month increase in vintage will increase the probability by 0.1198%; and, 1,000,000 dollars increase in average account balance will increase the probability by 0.02246%.

Different occupations and channel codes lead to different level of interest. A customer who is salaried has a higher probability of taking credit cards than one with other occupations by 4.259%, a self_employed customer has a higher probability than a salaried customer by 4.259%, and an entrepreneur has a higher probability than a self_employed customer by 4.259%.

For the four channel codes, a customer who chooses “X2” has a higher probability of taking credit cards than a customer who chooses “X1” by 9.115%, a customer who chooses “X3” has a higher probability than the one who chooses “X2” by 9.115%. The customers choose “X4” have a higher probability of than those who choose “X3” by 9.115%.

However, we also notice that females have a lower probability of 2.299% to take credit cards than males. This shows that gender plays an important role in customers’ interest in credit cards. In addition, a customer who has an active credit product has a lower probability of activating the account than a customer who does not by 8.443%.

```
# Accurate
ols.pred.classes <- ifelse(fitted(OLS.Model) > 0.5, 1, 0)
table(ols.pred.classes, data$Is_Active)
```

```
##
## ols.pred.classes     0     1
##                0 50837 24597
##                1 12960 16918
```

```
mean(ols.pred.classes == data$Is_Active)
```

```
## [1] 0.643374
```

The result shows that the accuracy is 64.3374% when we convert “Is_Active” to binary.

```
library(lmvar)
fit = lm(Is_Active ~ Gender + Age + Occupation + Channel_Code + Vintage
         + Credit_Product +Avg_Account_Balance, data = data, x=TRUE, y=TRUE)
cv.lm(fit, k=5) # perform a 5-fold cv
```

```
## Mean absolute error        :  0.4330525
## Sample standard deviation  :  0.001183757
##
## Mean squared error         :  0.2165537
## Sample standard deviation  :  0.001572842
##
## Root mean squared error    :  0.4653509
## Sample standard deviation  :  0.001690367
```

Root mean squared error is 0.4653289 < 0.5, so this is an okay model to use.

Probit Model

```
Probit.Model <- glm(Is_Active ~ Gender + Age+ Occupation + Channel_Code
                    + Vintage + Credit_Product + Avg_Account_Balance,
                    family=binomial(link="probit"), data = data)
summary(Probit.Model)
```

```
##
## Call:
## glm(formula = Is_Active ~ Gender + Age + Occupation + Channel_Code +
##     Vintage + Credit_Product + Avg_Account_Balance, family = binomial(link = "probit"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8032  -0.9430  -0.7242   1.1509   2.2053
##
## Coefficients:
##                            Estimate      Std. Error z value
## (Intercept)          -2.180335017947  0.027576324477 -79.065
## Gender               -0.064381114529  0.008242539477  -7.811
## Age                   0.009629843417  0.000391666488  24.587
## Occupation            0.119468005030  0.005014942079  23.822
## Channel_Code          0.245926846656  0.006053012264  40.629
## Vintage               0.003081554713  0.000164247726  18.762
## Credit_Product        0.226043718193  0.006165347867  36.664
## Avg_Account_Balance   0.000000063303  0.000000004686  13.508
##                                 Pr(>|z|)
## (Intercept)          < 0.0000000000000002
## Gender                0.00000000000000568
## Age                  < 0.0000000000000002
## Occupation           < 0.0000000000000002
## Channel_Code         < 0.0000000000000002
## Vintage              < 0.0000000000000002
## Credit_Product       < 0.0000000000000002
## Avg_Account_Balance < 0.0000000000000002
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 141243  on 105311  degrees of freedom
## Residual deviance: 131140  on 105304  degrees of freedom
## AIC: 131156
##
## Number of Fisher Scoring iterations: 4
```

```
confint(Probit.Model)
```

```
## Waiting for profiling to be done...
```

```
##                                 2.5 %           97.5 %
## (Intercept)          -2.23417029886423 -2.12655226692457
## Gender               -0.08054099303620 -0.04822090305290
## Age                   0.00886264219527  0.01039699211410
## Occupation            0.10966840911315  0.12926925405325
## Channel_Code          0.23405590725779  0.25779949948908
```

```
## Vintage              0.00276085765945  0.00340234543331
## Credit_Product       0.21408563878054  0.23800909019379
## Avg_Account_Balance  0.00000005417661  0.00000007242973
```

Comment: All the variables are statistically significant in the probit model. The 95% confidence interval for each variable shows us that all estimators are statistically significantsince the all intervals do not contain 0.

We notice all estimators have the same signs and significant codes as they are in the OLS model. Many estimators are also close in value, such as the estimators of "Age", "Vintage", and "Avg_Account_Balance". So, one year increase in age will increase the probability of taking credit cards by 0.963%; one month's increase in Vintage will increase the probability by 0.3082%; and 1,000,000 dollars increase in average account balance will increase the probability by 0.0633%.

Some estimators are larger, such as "Gender", "Credit_Product", "Occupation", and "Channel_Code". Males have a higher probability of 6.438% to take credit cards than females. A customer who has an active credit product has a lower probability of activating the account than a customer who does not by 22.6%.

A customer who is salaried has a higher probability of taking credit cards than one with other occupations by 11.95%, a self_employed customer has a higher probability than a salaried customer by 11.95%, and an entrepreneur has a higher probability than a self_employed customer by 11.95%.

A customer who chooses "X2" has a higher probability of taking credit cards than a customer who chooses "X1" by 24.59%, a customer who chooses "X3" has a higher probability than the one who chooses "X2" by 24.59%. The customers choose "X4" have a higher probability of than those who choose "X3" by 24.59%.

Here the estimators of "Gender", "Credit_Product", "Occupation", and "Channel_Code" are all more statistically significant, and differences in those variables cause a larger change in determining if the customer is active.

```r
# Accurate
probit.pred.classes <- ifelse(fitted(Probit.Model) > 0.5, 1, 0)
table(probit.pred.classes, data$Is_Active)
```

```
##
## probit.pred.classes     0     1
##                   0 50877 24639
##                   1 12920 16876
```

```r
mean(probit.pred.classes == data$Is_Active)
```

```
## [1] 0.643355
```

64.3355% accurate, which is a little bit worse than OLS model but almost the same.

Logit Model

```r
Logit.Model <- glm(Is_Active ~ Gender + Age+ Occupation + Channel_Code
                   + Vintage + Credit_Product + Avg_Account_Balance,
                   family=binomial(link="logit"), data = data)
summary(Logit.Model)
```

```
##
## Call:
## glm(formula = Is_Active ~ Gender + Age + Occupation + Channel_Code +
##     Vintage + Credit_Product + Avg_Account_Balance, family = binomial(link = "logit"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.7883  -0.9402  -0.7270   1.1498   2.1520
##
## Coefficients:
##                          Estimate      Std. Error z value
## (Intercept)         -3.559385787506  0.045834397349 -77.658
## Gender              -0.107130460020  0.013537195808  -7.914
## Age                  0.015755659588  0.000637563722  24.712
## Occupation           0.197108705206  0.008191904106  24.061
## Channel_Code         0.395956528697  0.009820731328  40.318
## Vintage              0.005048973287  0.000265879353  18.990
## Credit_Product       0.372790725550  0.010134072578  36.786
## Avg_Account_Balance  0.000000101389  0.000000007657  13.241
##                               Pr(>|z|)
## (Intercept)         < 0.0000000000000002
## Gender                0.0000000000000025
## Age                 < 0.0000000000000002
## Occupation          < 0.0000000000000002
## Channel_Code        < 0.0000000000000002
## Vintage             < 0.0000000000000002
## Credit_Product      < 0.0000000000000002
## Avg_Account_Balance < 0.0000000000000002
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 141243  on 105311  degrees of freedom
## Residual deviance: 131169  on 105304  degrees of freedom
## AIC: 131185
##
## Number of Fisher Scoring iterations: 4
```

```
confint(Logit.Model)
```

```
## Waiting for profiling to be done...

##                               2.5 %           97.5 %
## (Intercept)         -3.64932783252265 -3.4696589123572
## Gender              -0.13366275639513 -0.0805976281131
## Age                  0.01450593334352  0.0170051689946
## Occupation           0.18105683099962  0.2131687281415
## Channel_Code         0.37671474687297  0.4152119138685
## Vintage              0.00452800993617  0.0055702487087
## Credit_Product       0.35294709007543  0.3926722703695
## Avg_Account_Balance  0.00000008638431  0.0000001164014
```

Comment: All the variables are statistically significant in the logit model. The 95% confidence interval for each variable also shows us that all estimators are statistically significant since the all intervals do not contain 0.

Similarities are noticed. One year increase in age will increase the probability of taking credit cards by 1.576%. One unit increase in vintage will increase the probability by 0.5049%. 1,000,000 dollars increase in average account balance will increase the probability of activating the account by 0.1041%. These results are not too different from the previous two models.

However, we notice that males have a higher probability of 10.71% of taking credit cards than males. Also, a customer who has an active credit product has a lower probability of activating the account than a customer who does not by 37.28%. Logit model suggests that a customer's gender and experiences in having active credit product may have a stronger influence on his or her choice in credit cards.

Other estimators, such as "Occupation", suggests that a customer who is salaried has a higher probability of taking credit cards than one with other occupations by 19.71%, a self_employed customer has a higher probability than a salaried customer by 19.71%, and an entrepreneur has a higher probability than a self_employed customer by 19.71%.

For the four channel codes, a customer who chooses "X2" has a higher probability of taking credit cards than a customer who chooses "X1" by 39.6%, a customer who chooses "X3" has a higher probability than the one who chooses "X2" by 39.6%. The customers choose "X4" have a higher probability of than those who choose "X3" by 39.6%. These two results are similar to the probit model but very different from the OLS model.

```
# Accuate
logit.pred.classes <- ifelse(fitted(Logit.Model) > 0.5, 1, 0)
table(logit.pred.classes, data$Is_Active)
```

```
##
## logit.pred.classes     0     1
##                  0 50901 24608
##                  1 12896 16907
```

```
mean(logit.pred.classes == data$Is_Active)
```

```
## [1] 0.6438772
```

64.38% accurate, so it seems like the logit model is better than the OLS model and the probit model.

```
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
# OLS average marginal effects
sum_phi <-mean(dnorm(predict(OLS.Model,type="response")))
ame.ols=sum_phi*coef(OLS.Model)

# probit average marginal effects
sum_phi <-mean(dnorm(predict(Probit.Model,type="link")))
ame.probit=sum_phi*coef(Probit.Model)
# average marginal effects
sum_phi <-mean(dnorm(predict(Logit.Model,type="link")))
ame.logit=sum_phi*coef(Logit.Model)


stargazer(ame.ols, ame.probit, ame.logit, type = "text",column.labels = c("ols","probit","logit"))
```

```
##
## ===========================================================================================
## (Intercept) Gender   Age   Occupation Channel_Code Vintage Credit_Product Avg_Account_Balance
## -------------------------------------------------------------------------------------------
## -0.110      -0.008 0.001    0.016        0.033     0.0004      0.031                0
## -------------------------------------------------------------------------------------------
##
## ===========================================================================================
## (Intercept) Gender   Age   Occupation Channel_Code Vintage Credit_Product Avg_Account_Balance
## -------------------------------------------------------------------------------------------
## -0.774      -0.023 0.003    0.042        0.087      0.001      0.080             0.00000
```

```
## ------------------------------------------------------------------------------------------
##
## ==========================================================================================
## (Intercept) Gender  Age  Occupation Channel_Code Vintage Credit_Product Avg_Account_Balance
## ------------------------------------------------------------------------------------------
## -1.076      -0.032 0.005   0.060       0.120      0.002      0.113             0.00000
## ------------------------------------------------------------------------------------------
```

The average marginal effect has influence on probability.

From the results of OLS average marginal effect, only "Gender" has a negative effect, that is, males have a higher probability of taking credit cards than females by 0.8%. Both "Channel_Code" and "Credit_product" have dominated effect, which means that "Channel_Code" and "Credit_product" increase the probability of "Is_Active" for about 3%.

From the average marginal effect results of probit and logit models, the direction of marginal effects is consistent with OLS model. However, compared with OLS model, probit and logit model are more influential. In particular, "Channel_Code" and "Credit_Product" in logit model have a positive impact on customers' credit cards purchase intention about 12%.

```r
# Linear model predictions
(tab <- table(predict(OLS.Model) > 0.5, data$Is_Active))
```

```
##
##             0     1
##   FALSE 50837 24597
##   TRUE  12960 16918
```

```r
c(tab[1, 1]/sum(tab[, 1]), tab[2, 2]/sum(tab[,2]))
```

```
## [1] 0.7968557 0.4075154
```

```r
# Probit model predictions
(tab <- table(predict(Probit.Model, type = "response") > 0.5, data$Is_Active))
```

```
##
##             0     1
##   FALSE 50877 24639
##   TRUE  12920 16876
```

```r
c(tab[1, 1]/sum(tab[, 1]), tab[2, 2]/sum(tab[,2]))
```

```
## [1] 0.7974826 0.4065037
```

```r
# Logit model predictions
(tab <- table(predict(Logit.Model, type = "response") > 0.5, data$Is_Active))
```

```
##
##             0     1
##   FALSE 50901 24608
##   TRUE  12896 16907
```

```r
c(tab[1, 1]/sum(tab[, 1]), tab[2, 2]/sum(tab[,2]))
```

```
## [1] 0.7978588 0.4072504
```

The prediction accuracy of OLS model for inactive customers is 79.69% and that for active customers is 40.75%. The prediction accuracy of probit model is 79.75% for inactive customers and 40.65% for active customers. The prediction accuracy of logit model is 79.79% for inactive customers and 40.73% for active customers. It can be seen from the statistical data that the accuracy of these three models in predicting whether customers are active is very close, and logit model is slightly higher than the other two models.

From all results above, logit model is the best fitting model among all three models. In order to evaluate logit model and find if there is a more suitable one, we will divide the data into testing and training data, and compare the sensitivity, specificity, and RMSE.

Probit Training

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(e1071)
intraining <-createDataPartition(data$Is_Active, p=0.75, list=FALSE)
training <- data[intraining,]
testing <-data[-intraining,]
train_control <-trainControl(method="cv", number=5)
probit.mod <-train(as.factor(Is_Active)~.,
                  data=training,
                  method="glm",
                  family="binomial"(link="probit"),
                  trControl=train_control)
summary(probit.mod)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8079  -0.9422  -0.7228   1.1513   2.2104
##
## Coefficients:
##                          Estimate      Std. Error z value
## (Intercept)         -2.202259326833 0.031966464450  -68.89
## Gender              -0.058176727516 0.009520946347   -6.11
## Age                  0.009906796375 0.000453464836   21.85
## Occupation           0.118848774102 0.005797056548   20.50
## Channel_Code         0.248742713154 0.007006609765   35.50
## Vintage              0.003002212546 0.000189980865   15.80
## Credit_Product       0.227854865233 0.007140906955   31.91
## Avg_Account_Balance  0.000000063083 0.000000005369   11.75
##                                Pr(>|z|)
## (Intercept)         < 0.0000000000000002
## Gender                    0.000000000994
## Age                 < 0.0000000000000002
## Occupation          < 0.0000000000000002
## Channel_Code        < 0.0000000000000002
## Vintage             < 0.0000000000000002
## Credit_Product      < 0.0000000000000002
## Avg_Account_Balance < 0.0000000000000002
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105876  on 78983  degrees of freedom
```

```
## Residual deviance:  98255  on 78976  degrees of freedom
## AIC: 98271
##
## Number of Fisher Scoring iterations: 4
```

Compare to our previous probit model, the estimators of the variables haven't changed much and the signs are consistent.

```r
# Predict using the testing data
pred_is_active <-predict(probit.mod, newdata=testing)
# Evaluate performance
confusionMatrix(data=pred_is_active, reference=as.factor(testing$Is_Active))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 12622  6183
##          1  3262  4261
##
##                Accuracy : 0.6413
##                  95% CI : (0.6354, 0.6471)
##     No Information Rate : 0.6033
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.2128
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.7946
##             Specificity : 0.4080
##          Pos Pred Value : 0.6712
##          Neg Pred Value : 0.5664
##              Prevalence : 0.6033
##          Detection Rate : 0.4794
##    Detection Prevalence : 0.7143
##       Balanced Accuracy : 0.6013
##
##        'Positive' Class : 0
##
```

```r
# Training RMSE
sqrt(mean((training$Is_Active - predict(Probit.Model, training)) ^ 2))
```

```
## [1] 0.8649105
```

```r
# Testing RMSE
sqrt(mean((testing$Is_Active - predict(Probit.Model, testing)) ^ 2))
```

```
## [1] 0.8696569
```

From the result, we see that accuracy is about 0.64. Sensitivity is about 0.79, which is close to 1.0 and shows that the number of correct positive predictions is high. Specificity is about 0.40, which is a little bit low and shows that the number of correct negative predictions is a bit low.

Both RMSE for training and testing are about 0.86. This is a large number that indicates probit model may not be a suitable model for our data.

Logit Training

```r
logit.mod <-train(as.factor(Is_Active)~.,
                  data=training,
                  method="glm",
                  family="binomial"(link="logit"),
                  trControl=train_control)
summary(logit.mod)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7912  -0.9391  -0.7258   1.1503   2.1562
##
## Coefficients:
##                             Estimate      Std. Error z value
## (Intercept)          -3.594462934326  0.053147344206 -67.632
## Gender               -0.097471602017  0.015640876734  -6.232
## Age                   0.016201691662  0.000738201252  21.948
## Occupation            0.196233004787  0.009470603602  20.720
## Channel_Code          0.400554177617  0.011369008828  35.232
## Vintage               0.004922764574  0.000307541665  16.007
## Credit_Product        0.375574357094  0.011738523080  31.995
## Avg_Account_Balance   0.000000100832  0.000000008776  11.489
##                                  Pr(>|z|)
## (Intercept)          < 0.0000000000000002
## Gender                     0.000000000461
## Age                  < 0.0000000000000002
## Occupation           < 0.0000000000000002
## Channel_Code         < 0.0000000000000002
## Vintage              < 0.0000000000000002
## Credit_Product       < 0.0000000000000002
## Avg_Account_Balance  < 0.0000000000000002
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105876  on 78983  degrees of freedom
## Residual deviance:  98279  on 78976  degrees of freedom
## AIC: 98295
##
## Number of Fisher Scoring iterations: 4
```

Compare to our previous logit model, the estimators of the variables haven't changed much and the signs are consistent.

```r
# Predict using the testing data
pred_is_active <-predict(logit.mod, newdata=testing)
# Evaluate performance
confusionMatrix(data=pred_is_active, reference=as.factor(testing$Is_Active))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction     0     1
##          0 12622  6160
##          1  3262  4284
##
##                Accuracy : 0.6421
##                  95% CI : (0.6363, 0.6479)
##     No Information Rate : 0.6033
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.215
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.7946
##             Specificity : 0.4102
##          Pos Pred Value : 0.6720
##          Neg Pred Value : 0.5677
##              Prevalence : 0.6033
##          Detection Rate : 0.4794
##    Detection Prevalence : 0.7134
##       Balanced Accuracy : 0.6024
##
##        'Positive' Class : 0
##
```

From the sensitivity and specificity report, we can tell it is almost the same as probit model. We see that accuracy is the same, about 0.64. Sensitivity is about 0.79, which is close to 1.0 and shows that the number of correct positive predictions is high. Specificity is about 0.40, which is a little bit low and shows that the number of correct negative predictions is a bit low.

```
logit.mod <-train((Is_Active)~.,
                  data=training,
                  method="glm",
                  family="binomial"(link="logit"),
                  trControl=train_control)
#predict using the testing data
pred_is_active <-predict(logit.mod,newdata=testing)
# evaluate performance
postResample(pred = pred_is_active, obs = testing$Is_Active)
```

```
##       RMSE   Rsquared        MAE
## 0.46644981 0.09098525 0.43333957
```

RMSE is about 0.46 < 0.5, which is much smaller than what we had in probit model. This means that the logit model fits the dataset better.

For now, we still consider logit model as our best model. We are going to do some tests on it to prove or disprove our perspective.

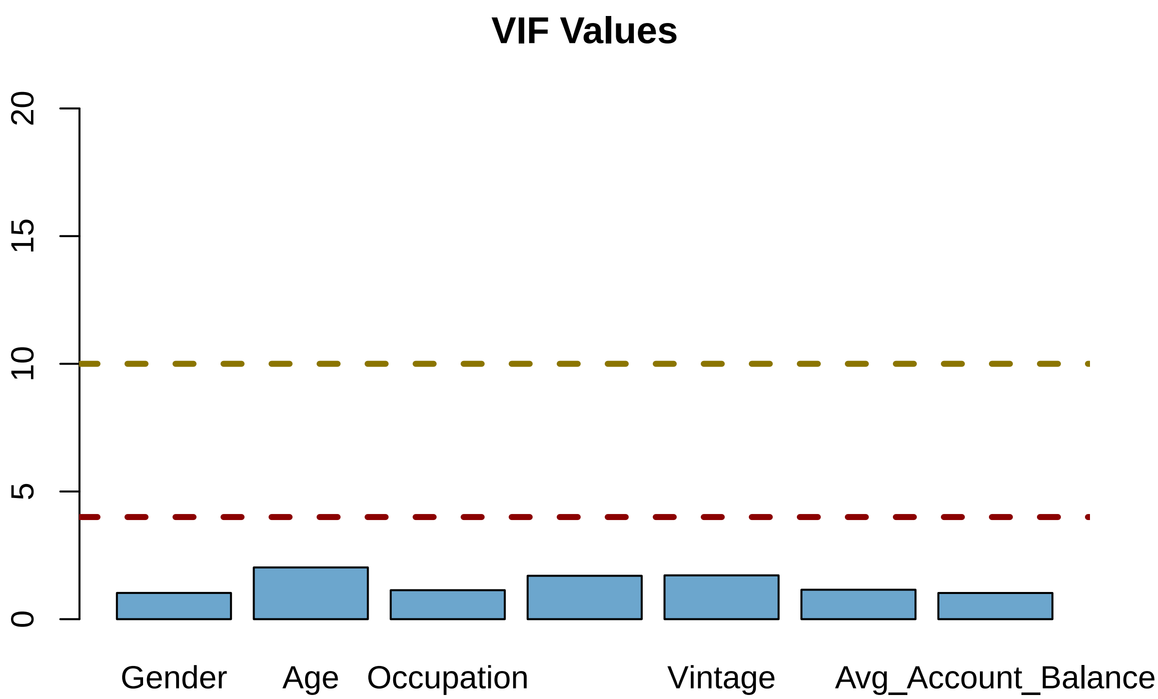
Tests for Logit Model

```
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following objects are masked from 'package:lessR':
##
##     bc, Recode, sp

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster
```

```
library(lmtest)
library(sandwich)

# VIF
barplot(vif(Logit.Model),main="VIF Values",col="skyblue3",ylim=c(0.0,20.0))
abline(h=4, lwd=3,lty=2,col="red4")
abline(h=10, lwd=3,lty=2,col="gold4")
```



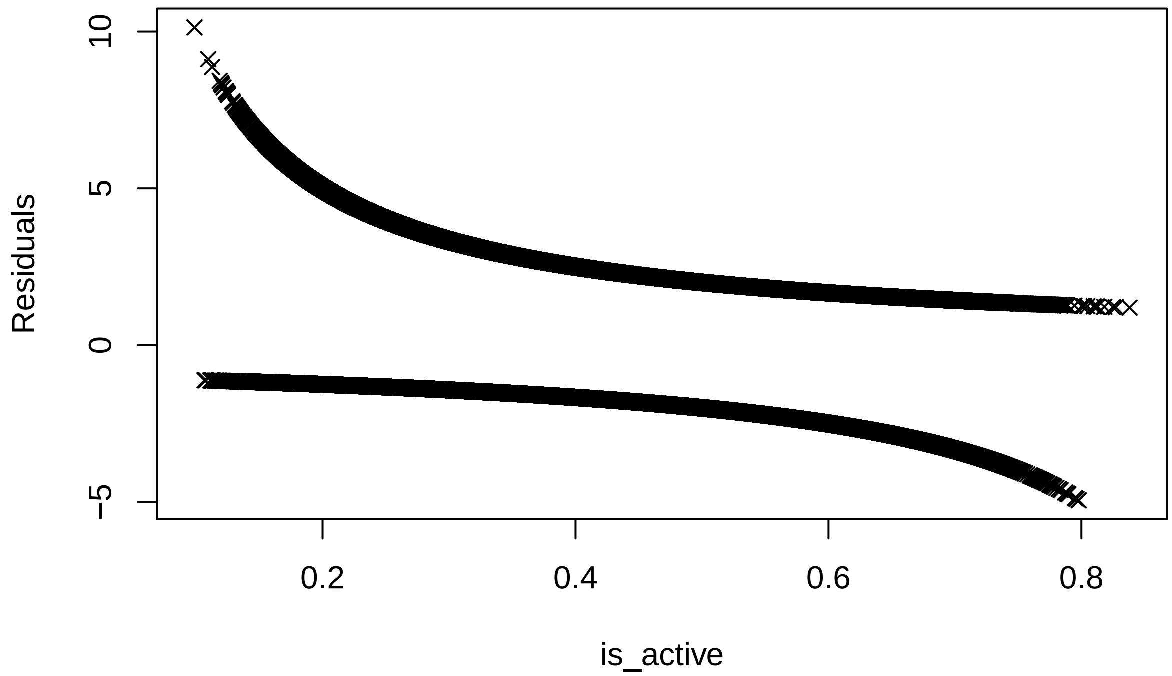
The result indicates that all of the VIF values are smaller than 4. Therefore, it is appropriate that we keep all the variables for the logit model.

```
# Reset
library(car)
library(lmtest)
resettest(Logit.Model, power=2, type="regressor")
```

```
##
##  RESET test
##
## data:  Logit.Model
## RESET = 622.23, df1 = 7, df2 = 105297, p-value < 0.00000000000000022
```

p-value is smaller than 0.05, which means we may need to include a higher power term or interaction term to improve our logit model.

```
# Plot the respective vs. y hat
plot(Logit.Model$fitted.values, Logit.Model$residuals, xlab="is_active", ylab="Residuals", pch=4)
```



From the graph, we may tell there is heteroskedasticity, and the residuals are not constant. We can use bptest to test the heteroskedastcity.

```
# Heteroskedasticity
bptest(Logit.Model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Logit.Model
## BP = 5004.4, df = 7, p-value < 0.00000000000000022
```

Since the p-value is smaller than 0.05, we reject the null that homoscedasticity is presented and conclude that heteroskedasticity exists.

```
# Robust standard error
cov1 <-hccm(Logit.Model,type="hc1")
Logit.Model.heter <-coeftest(Logit.Model, vcov.=cov1)
library(broom)
tidy(Logit.Model.heter)
```

```
## # A tibble: 8 x 5
##   term                  estimate    std.error statistic   p.value
##   <chr>                    <dbl>        <dbl>     <dbl>     <dbl>
## 1 (Intercept)         -3.56        0.113          -31.6  1.13e-219
## 2 Gender              -0.107       0.0324          -3.30 9.58e-   4
## 3 Age                  0.0158      0.00152         10.3  4.78e-  25
## 4 Occupation           0.197       0.0198           9.96 2.25e-  23
## 5 Channel_Code         0.396       0.0233          17.0  1.13e-  64
## 6 Vintage              0.00505     0.000640         7.89 3.05e-  15
## 7 Credit_Product       0.373       0.0252          14.8  1.60e-  49
## 8 Avg_Account_Balance  0.000000101 0.0000000187     5.42 6.01e-   8
```

Multinomial Logit Model

In our logit model, we noticed that the variable "Credit_Product" has a significant effect on the model. "Credit_Product" measures if the consumers have any active credit products, so intuitively, the variables such as "Gender" and "Age" may affect "Credit_Product" in some way, though this has not been seen in our model or tests. Then, we note that for customers, there may be multinomial choice situations, because they actually face more than two choices. In other words, if we ignore the choice of "Credit_Product"="N/A", there are at least three combinations of "Is_Active" and "Credit_Product", indicating that customers can make at least three different choices. We can put the customers into three groups: inactive customers, active cutsomers who have active credit product, and active cutsomers who does not have active credit product.

In order to help Happy Customer Bank to cross sell, it is necessary for us to analyze customers' behavior in different groups.

So, we create a new variable "ia_cp" based on "Is_Active" and "Credit_Product". We ignore the observations where "Credit_Product" = "N/A" as they could be confusing. We let the variable "ia_cp" ranges from 1 to 3 for three possible combinations of "Is_Active" and "Credit_Product", so we can use a multinomial logit model to see the similarities and differences among the four groups.

```
# Create new variable
data2 = data [data$Credit_Product != 1, ]
data2$ia_cp [data2$Is_Active == 0] <- 1
data2$ia_cp [data2$Is_Active == 1 & data2$Credit_Product == 3] <- 2
data2$ia_cp [data2$Is_Active == 1 & data2$Credit_Product == 2] <- 3

table(data2$ia_cp)
```

```
##
##     1     2     3
## 57387 25855  9548
```

From the table, it seems like our dataset is a little bit biased.

```
library(caret)
library(nnet)
attach(data)
```

```
## The following objects are masked from data (pos = 18):
##
##     Age, Avg_Account_Balance, Channel_Code, Credit_Product, Gender,
##     Is_Active, Occupation, Vintage

## The following objects are masked from data (pos = 21):
##
##     Age, Avg_Account_Balance, Channel_Code, Credit_Product, Gender,
##     Is_Active, Occupation, Vintage
```

```
multi <- multinom(ia_cp ~ Gender + Age + Occupation + Channel_Code + Vintage
                  + Avg_Account_Balance, data = data2)
```

```
## # weights:  24 (14 variable)
## initial  value 101940.234265
## iter  10 value 83544.812683
## iter  20 value 77162.953771
## final  value 77087.447339
## converged
```

```
summary(multi)
```

```
## Call:
## multinom(formula = ia_cp ~ Gender + Age + Occupation + Channel_Code +
##     Vintage + Avg_Account_Balance, data = data2)
##
## Coefficients:
##   (Intercept)     Gender        Age Occupation Channel_Code       Vintage
## 2   -2.537160 -0.0631628 0.01776665  0.1853642    0.2895855 -0.0005045724
## 3   -4.767244 -0.2393539 0.01388448  0.2224651    0.5839420  0.0119428055
##   Avg_Account_Balance
## 2     0.00000008731495
## 3     0.00000012917576
##
## Std. Errors:
##              (Intercept)                  Gender                  Age
## 2 0.000000000000003034117 0.000000000000001411385 0.0000000000001340968
## 3 0.000000000000003752311 0.000000000000001412151 0.0000000000001902959
##               Occupation            Channel_Code              Vintage
## 2 0.000000000000006446172 0.000000000000005657823 0.0000000000001377464
## 3 0.000000000000008335946 0.000000000000008821456 0.0000000000002457466
##   Avg_Account_Balance
## 2   0.000000005352584
## 3   0.000000007447146
##
## Residual Deviance: 154174.9
## AIC: 154202.9
```

Comment: From the summary, we notice that most estimators are similar to what we had in our logit model. Almost all estimators are consistent in values and signs. "ia_cp"=1 is missing, so it is the baseline. "Age" and "Avg_Account_Balance" increase the probability of taking credit cards, while the probability decreases in "Gender" (females have a lower probability). The estimators for "Occupation" and "Channel_Code" show that for customers with different occupations and different interests in channels, their response towards credit cards are different. However, the estimators for "Vintage" have different signs. This indicates that vintage may affect different groups of customers differently. We will come back to this after a few tests to decide if this model is worth to analyze.

```
# Tests for multinomial logit model
multi$AIC
```

```
## [1] 154202.9
```

```
multi$edf
```

```
## [1] 14
```

AIC is large indicates that the model does not fit the data very well, and edf is small implies that the (effective) number of degrees of freedom used by the model is low. Therefore, the multinomial logit model is questionable. Hence, we will not choose multinomial logit model as our best model.

Instrumental Model

In section 2 after plotting correlation plot, we noticed that the variables "Channel_Code" and "Vintage" have high correlations with "Age". Therefore, we have two potential instrumental variables: "Channel_Code" & "Vintage", with "Age" as the endogenous variable, and "Gender", "Occupation", "Credit_Product", and "Avg_Account_Balance" as exogenous variables. We will establish a new model here and conduct a range of tests for our new model.

```r
library(AER)
library(lmtest)
library(sandwich)
```

```r
# Model with two instrumental variables Channel_Code & Vintage
iv.mod <- ivreg(Is_Active ~ Gender + Age + Occupation + Credit_Product
                + Avg_Account_Balance | Gender + Occupation + Credit_Product
                + Avg_Account_Balance + Channel_Code + Vintage, data = data)
summary(iv.mod)
```

```
##
## Call:
## ivreg(formula = Is_Active ~ Gender + Age + Occupation + Credit_Product +
##     Avg_Account_Balance | Gender + Occupation + Credit_Product +
##     Avg_Account_Balance + Channel_Code + Vintage, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9908 -0.3723 -0.1742  0.4569  1.1186
##
## Coefficients:
##                           Estimate      Std. Error t value
## (Intercept)         -0.608840694249  0.011968922434 -50.868
## Gender              -0.012312605340  0.003024377414  -4.071
## Age                  0.013373474581  0.000153001911  87.407
## Occupation           0.082096256617  0.001782966241  46.045
## Credit_Product       0.091987247448  0.002278836936  40.366
## Avg_Account_Balance  0.000000016758  0.000000001726   9.709
##                                 Pr(>|t|)
## (Intercept)         < 0.0000000000000002
## Gender                         0.0000468
## Age                 < 0.0000000000000002
## Occupation          < 0.0000000000000002
## Credit_Product      < 0.0000000000000002
## Avg_Account_Balance < 0.0000000000000002
##
## Residual standard error: 0.4766 on 105306 degrees of freedom
## Multiple R-Squared: 0.04889, Adjusted R-squared: 0.04885
## Wald test:  1918 on 5 and 105306 DF,  p-value: < 0.00000000000000022
```

From the summary, females have a lower probability of 1.23% to take credit cards than males. One year increase in age will increase the probability of activating his or her account by 1.34%. 1,000,000 dollars increase in average account balance will increase the probability of activating the account by 0.0167%. A customer who has an active credit product has a lower probability of taking credit card than a customer who has not by 9.19%. In brief, we find the signs of the estimators are the same between the IV model and the logit model, and almost all estimators are smaller in value.

```r
# Hausman Test
summary(iv.mod, diagnostics=TRUE)
```

```
##
## Call:
## ivreg(formula = Is_Active ~ Gender + Age + Occupation + Credit_Product +
##     Avg_Account_Balance | Gender + Occupation + Credit_Product +
##     Avg_Account_Balance + Channel_Code + Vintage, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9908 -0.3723 -0.1742  0.4569  1.1186
##
## Coefficients:
##                            Estimate      Std. Error t value
## (Intercept)         -0.608840694249  0.011968922434 -50.868
## Gender              -0.012312605340  0.003024377414  -4.071
## Age                  0.013373474581  0.000153001911  87.407
## Occupation           0.082096256617  0.001782966241  46.045
## Credit_Product       0.091987247448  0.002278836936  40.366
## Avg_Account_Balance  0.000000016758  0.000000001726   9.709
##                                 Pr(>|t|)
## (Intercept)         < 0.0000000000000002
## Gender                         0.0000468
## Age                 < 0.0000000000000002
## Occupation          < 0.0000000000000002
## Credit_Product      < 0.0000000000000002
## Avg_Account_Balance < 0.0000000000000002
##
## Diagnostic tests:
##                     df1    df2 statistic              p-value
## Weak instruments      2 105305   47151.4 <0.0000000000000002
## Wu-Hausman            1 105305    2340.5 <0.0000000000000002
## Sargan                1     NA     195.3 <0.0000000000000002
##
## Residual standard error: 0.4766 on 105306 degrees of freedom
## Multiple R-Squared: 0.04889, Adjusted R-squared: 0.04885
## Wald test:  1918 on 5 and 105306 DF,  p-value: < 0.00000000000000022
```

In the Diagnostic test, the result indicates that our tested instruments are not weak, which is good. We reject the null which essentially means that the OLS is not consistent, and endogeneity is present. The least squares estimator is a better and more efficient estimator. Furthermore, all of the coefficients are statistically significantly different from 0 and both instruments ("Channel_Code" & "Vintage") are valid. p-value < 2e-16, so we conclude that both the least squares estimator and the instrumental variables estimator are consistent. The least squares estimator is a better and more efficient estimator.

```
# Test the validity of instrument
linearHypothesis(OLS.Model,c("Channel_Code=0","Vintage=0"),vcov= vcovHC,type="HC1")
```

```
## Linear hypothesis test
##
## Hypothesis:
## Channel_Code = 0
## Vintage = 0
##
## Model 1: restricted model
## Model 2: Is_Active ~ Gender + Age + Occupation + Channel_Code + Vintage +
##     Credit_Product + Avg_Account_Balance
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F                Pr(>F)
## 1 105306
```

```
## 2 105304  2 1199.9 < 0.00000000000000022
```

Since p-value < 0.05, we reject the null hypothesis and conclude that the estimators of "Channel_Code" and "Vintage" are significantly different from 0, and f > 10 suggests they are strong instruments.

```r
# t-test of coefficients
coeftest(iv.mod, vcov = vcovHC, type = "HC1")
```

```
##
## t test of coefficients:
##
##                          Estimate      Std. Error  t value
## (Intercept)         -0.608840694249  0.011836959890 -51.4356
## Gender              -0.012312605340  0.003017367535  -4.0806
## Age                  0.013373474581  0.000153410391  87.1745
## Occupation           0.082096256617  0.001843416159  44.5348
## Credit_Product       0.091987247448  0.002370877352  38.7988
## Avg_Account_Balance  0.000000016758  0.000000001814   9.2383
##                                   Pr(>|t|)
## (Intercept)         < 0.00000000000000022
## Gender                         0.00004496
## Age                 < 0.00000000000000022
## Occupation          < 0.00000000000000022
## Credit_Product      < 0.00000000000000022
## Avg_Account_Balance < 0.00000000000000022
```

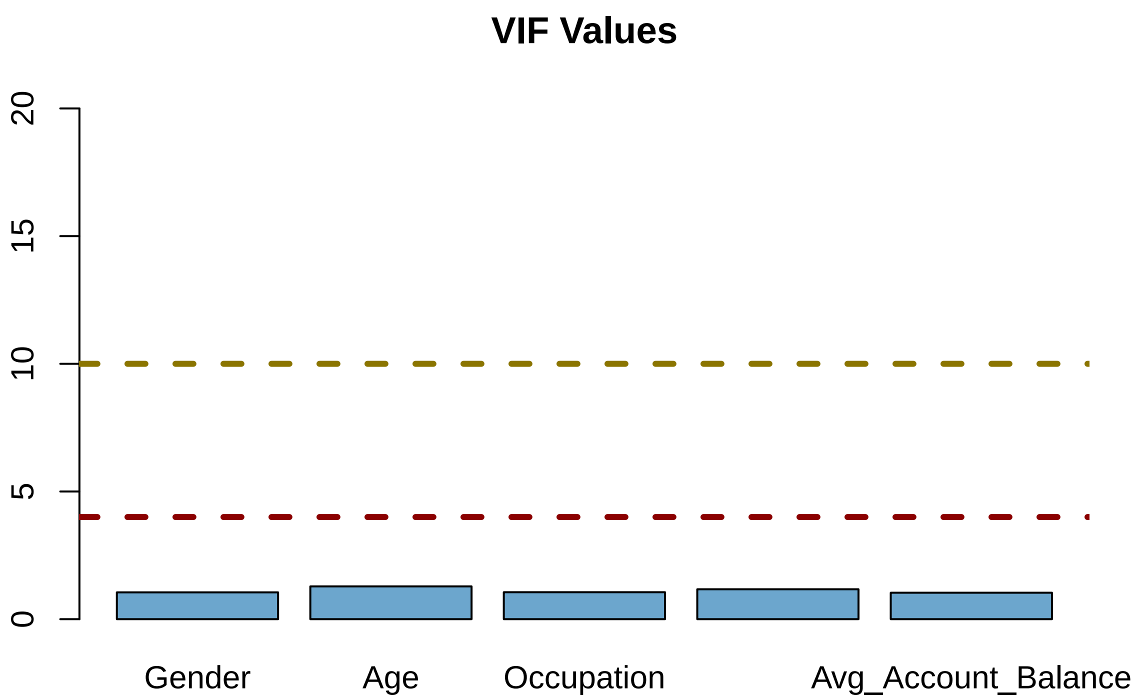
We conclude that all coefficients are statistically significantly different from 0.

```r
# Test the validity of the instruments (J -statistic) via
# the overidentifying restrictions test
iv_OR <- lm(residuals(iv.mod) ~ Gender + Occupation + Credit_Product
            + Avg_Account_Balance + Channel_Code + Vintage, data = data)
linearHypothesis(iv_OR, c("Channel_Code = 0", "Vintage = 0"), test = "Chisq")
```

```
## Linear hypothesis test
##
## Hypothesis:
## Channel_Code = 0
## Vintage = 0
##
## Model 1: restricted model
## Model 2: residuals(iv.mod) ~ Gender + Occupation + Credit_Product + Avg_Account_Balance +
##     Channel_Code + Vintage
##
##   Res.Df   RSS Df Sum of Sq  Chisq            Pr(>Chisq)
## 1 105307 23920
## 2 105305 23875  2    44.365 195.68 < 0.00000000000000022
```

Pr < 2.2e-16, so we conclude that both instruments are valid.

```r
# VIF for IV
barplot(vif(iv.mod), main="VIF Values", col="skyblue3", ylim=c(0.0,20.0))
abline(h=4, lwd=3,lty=2,col="red4")
abline(h=10, lwd=3,lty=2,col="gold4")
```



The result indicates that all of the vif values are smaller than 4, therefore, it is appropriate that we keep all the variables for the IV model.

```
# RESET for IV
library(car)
library(lmtest)
resettest(iv.mod, power=2, type="regressor")
```

```
##
##  RESET test
##
## data:  iv.mod
## RESET = 689.49, df1 = 7, df2 = 105299, p-value < 0.00000000000000022
```

With a statistic of 689.49 and a p-value of ~0.000, the RESET test suggests that the IV model is NOT correctly specified. So we reject H0.

```
# Heteroskedasticity for IV
bptest(iv.mod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  iv.mod
## BP = 4770.7, df = 5, p-value < 0.00000000000000022
```

Since the p-value is smaller than 0.05, we reject the null that homoscedasticity is presented and conclude that heteroskedasticity exists.

Therefore, our IV model is not so useful according to the tests. Logit model remains to be the best model here.

Identify Potential Customers

So, after going through different models and running different tests, we decide that logit model is the most suitable model for our data. Now we will use logit model to identify potential customers who will take credit card.

```
# load test.csv again because we need the customers' IDs
new_data <- read_csv("test.csv")
```

```
## Parsed with column specification:
## cols(
##   ID = col_character(),
##   Gender = col_character(),
##   Age = col_double(),
##   Region_Code = col_character(),
##   Occupation = col_character(),
##   Channel_Code = col_character(),
##   Vintage = col_double(),
##   Credit_Product = col_character(),
##   Avg_Account_Balance = col_double(),
##   Is_Active = col_character()
## )
```

```
# Store the coefficients
b1 = Logit.Model$coefficients[1]
b2 = Logit.Model$coefficients[2]
b3 = Logit.Model$coefficients[3]
b4 = Logit.Model$coefficients[4]
b5 = Logit.Model$coefficients[5]
b6 = Logit.Model$coefficients[6]
b7 = Logit.Model$coefficients[7]
b8 = Logit.Model$coefficients[8]
```

We identify potential customers by calculating the probability (or we can call it the level) of customers' interest in taking credit cards, and store the value as "Is_Lead".

```
# data$Is_Lead <- b1 + b2*data$Gender + b3*data$Age + b4*data$Occupation
# + b5*data$Channel_Code + b6*data$Vintage + b7*data$Credit_Product
# + b8*data$Avg_Account_Balance

data$Is_Lead <- b1 + b2*data$Gender + b3*data$Age + b4*data$Occupation + b5*data$Channel_Code + b6*data$

# Store "Is_Lead" to the new dataset with IDs
new_data$Is_Lead = data$Is_Lead
```

Now we have the level of customers' interest in taking credit cards. We expect "Is_Lead" to be between 0 and 1, but there may be numbers > 1 or < 0 due to the residuals. So we can now identify potential customers. We divide customers into three levels and follow the rule that the higher "Is_Lead" is, the more likely the customer is to be interested in a credit card. We will print some of the IDs of the customers from the first level, and we will provide the number of customers in all levels.

```
# First level: The most potential customers
level.1 <- new_data[new_data$Is_Lead > 0.7, ]
nrow(level.1)
```

```
## [1] 3591
```

```r
print(level.1$ID[1:100])
```

```
##   [1] "MZZAQMPT" "MXETLUP4" "JTYNEKRQ" "5NTYJJDV" "YRLEWZFQ" "NWKHMEOF"
##   [7] "LXHCDSVH" "BNIUPSVK" "KLHUI2SE" "CUAUZJH8" "6EDAM4EU" "NZKCSDZ4"
##  [13] "ENUB4N9S" "SGJFSSJK" "FPU4ZI8H" "27F5ZEQA" "ATJAEQCJ" "UGIADHRB"
##  [19] "UMVSSZ9N" "NCVH9WFE" "DIVE6ZUL" "JTCERL4D" "DHARALYV" "MPQQFX4I"
##  [25] "CJHC43KV" "FTHIZFGT" "VQQEMTVX" "7V6GQUOD" "QPQZPSMH" "5OY8PVUA"
##  [31] "5PRTDKXK" "4WTK9RBR" "JWPYWNEU" "GYKEBWCJ" "LGMCRTXE" "SXHNHYFX"
##  [37] "QADPUPRH" "SKAFGCWQ" "GY3GGMD6" "BKAUTCVF" "NX428PF5" "DAVGFGE3"
##  [43] "XRDU4PBA" "DAJT6YYA" "T5TGKHOM" "R4MRAHPP" "QWT7FVWQ" "8Q4FXOQI"
##  [49] "M8PSGEVT" "EIBVIG5Z" "BX8KYC5S" "KR8BXBYI" "KHUDKRPY" "XKAQ5I8F"
##  [55] "NMJN8TQV" "H6SXNKC7" "W4RJBEBS" "FAHIFIFF" "EZBSYWQU" "NCWSS9X9"
##  [61] "BDJWIXPB" "5S2GPQID" "ULZYVWRZ" "CYLS7DNT" "JMPMFHLR" "CHMWB8HB"
##  [67] "H8IKUYS2" "J9QWFQF9" "B4PB8ZAG" "T4IKMKME" "8CHLIVIY" "WECEW6V7"
##  [73] "GKUDDQSV" "ZZVS5V5X" "BKMHGYUD" "5PRAGM92" "4QVZMAT4" "ITOJTXUY"
##  [79] "A7K2XHTP" "QKTNQGVS" "XRZDNUSE" "XECJYFEB" "AKYV6SS7" "NR3BBJTI"
##  [85] "YTTS5ZR2" "FV8BZ7IQ" "KSZDM7NB" "LFYXBXAV" "FJUQUVFK" "FGR8EWNZ"
##  [91] "2TWCWRGK" "DFARFVGA" "PRM72CHV" "PFBPHDB6" "K8PVPHAP" "BCY5Q3NT"
##  [97] "Q29W7QSQ" "ZUEJN22B" "HH5NJGPO" "LS8U4YWW"
```

```r
# Second level: Also potential customers
level.2 <- new_data[new_data$Is_Lead <= 0.7, ]
level.2 <- level.2[level.2$Is_Lead > 0.5, ]
nrow(level.2)
```

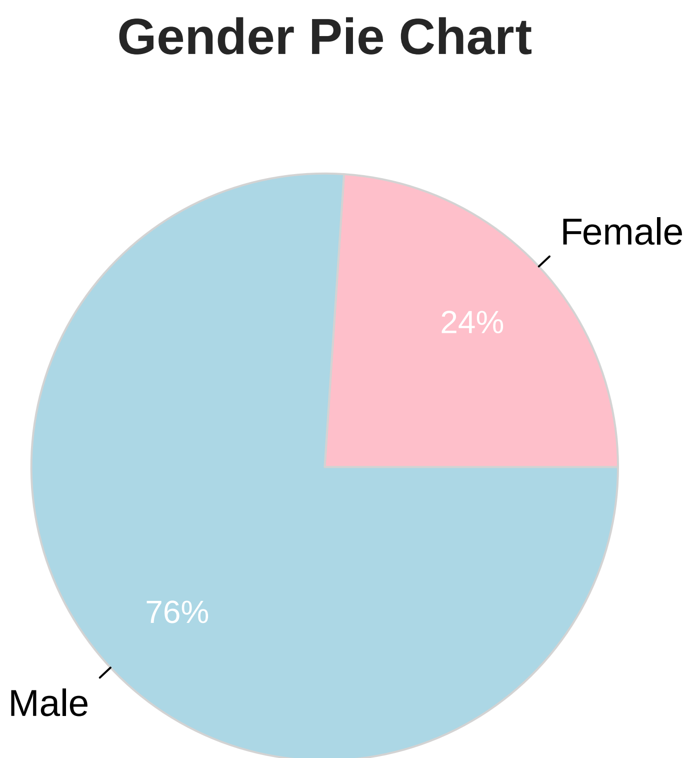
```
## [1] 5049
```

```r
# Third level: Least recommended customers, but the bank can still have a try
level.3 <- new_data[new_data$Is_Lead <= 0.5, ]
nrow(level.3)
```

```
## [1] 96672
```

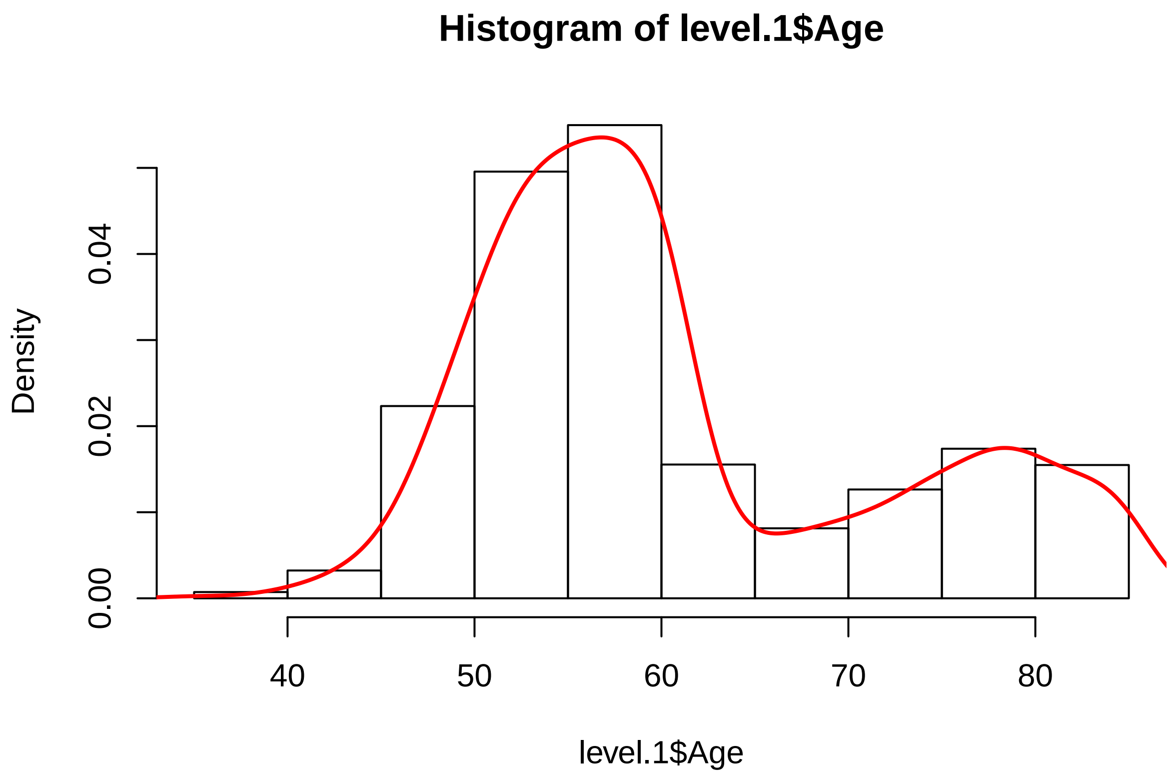
From here, we can see some of the IDs of the customers most likely to take credit cards. We will recommend customers to Happy Customer Bank according to their levels (i.e., the most recommended customers are from level.1, then level.2, and the least recommended customers are from level.3). Now, we are here at our last step before the conclusion. We are going to construct histograms and pie charts of the 3591 most potential customers in level.1 to find out the patterns in their information (i.e., how are "Gender", "Age", "Occupation", etc. distributed in potential customers).

```r
# Gender
PieChart(Gender, hole = 0, values = "%", data = level.1,
         fill = c("pink","lightblue"), main = "Gender Pie Chart")
```

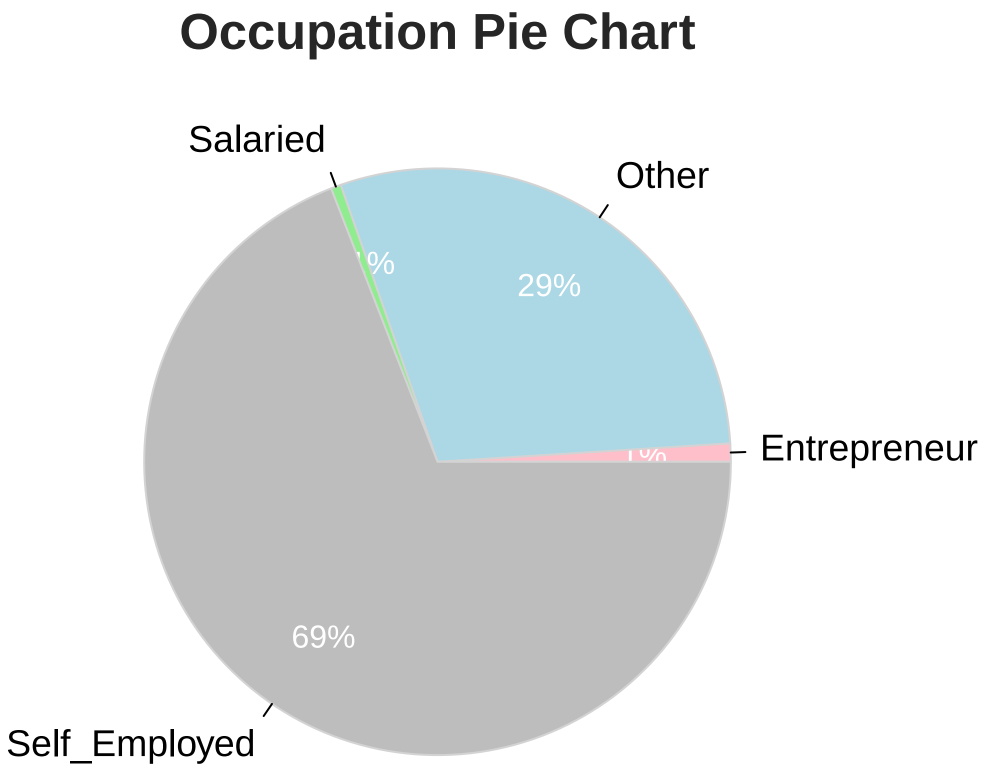



```
## >>> Suggestions
## PieChart(Gender, hole=0)  # traditional pie chart
## PieChart(Gender, values="%")  # display %'s on the chart
## BarChart(Gender)  # bar chart
## Plot(Gender)  # bubble plot
## Plot(Gender, values="count")  # lollipop plot
##
##
## --- Gender ---
##
##
##                  Female   Male      Total
## Frequencies:        860   2731       3591
## Proportions:      0.239  0.761      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 974.837, df = 1, p-value = 0.000
```

```
# 76% of the customers are male while only 24% of them are female.

# Age
hist(level.1$Age,prob=TRUE)
lines(density(level.1$Age), lwd = 2, col ="red")
```

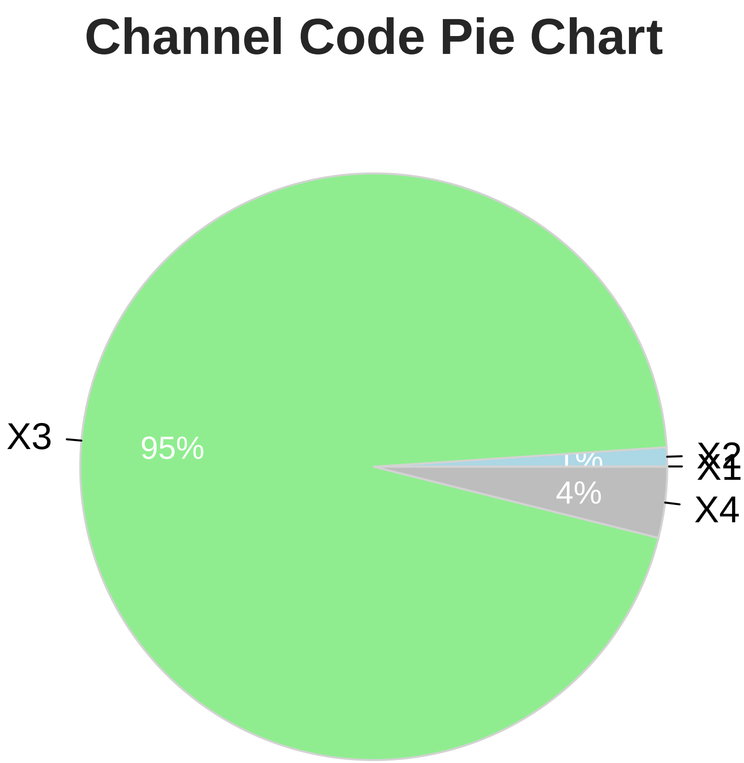



```r
# Recall that people aged from 25 to 30 are currently
# the largest client crowd of Happy Customer Bank;
# however, we observe that
# the most potential customers' age here falls in the interval [40,80].


# Occupation
PieChart(Occupation, hole = 0, values = "%", data = level.1,
         fill = c("pink","lightblue", "lightgreen", "grey"),
         main = "Occupation Pie Chart")
```



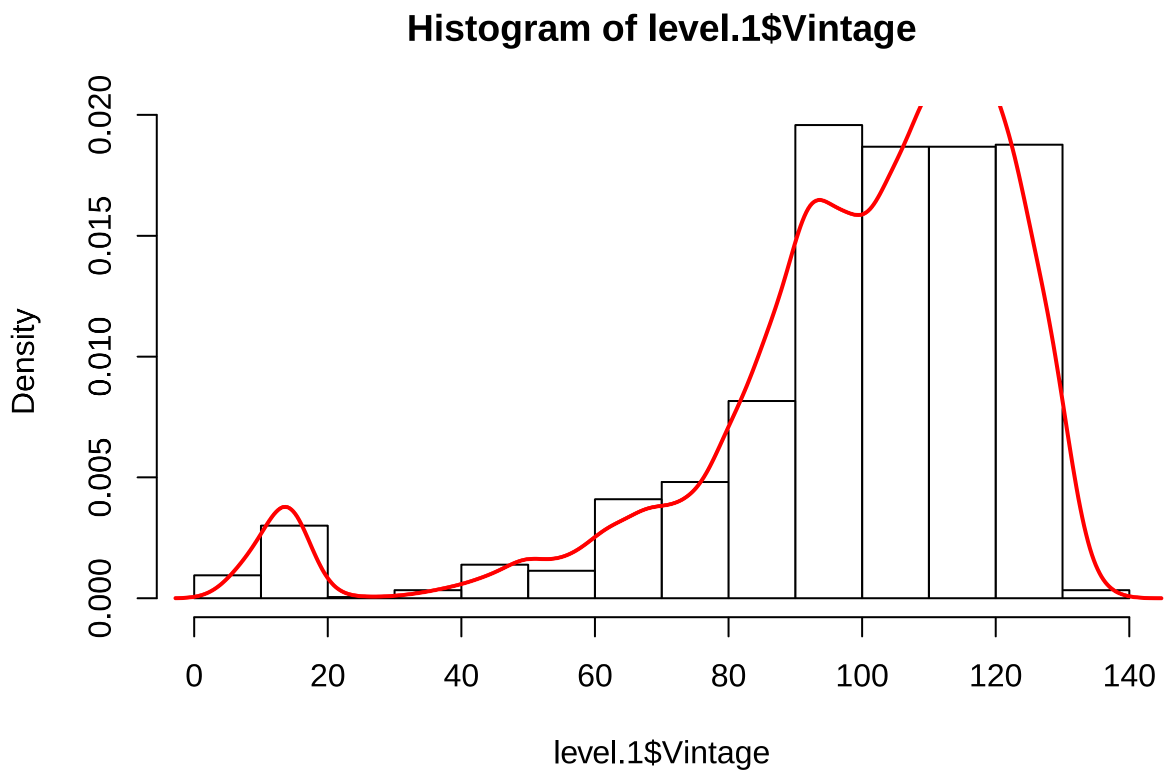

```
## >>> Suggestions
## PieChart(Occupation, hole=0)  # traditional pie chart
## PieChart(Occupation, values="%")  # display %'s on the chart
## BarChart(Occupation)  # bar chart
## Plot(Occupation)  # bubble plot
## Plot(Occupation, values="count")  # lollipop plot
##
##
## --- Occupation ---
##
##
##                 Entrepreneur  Other  Salaried  Self_Employed      Total
## Frequencies:              36   1054        20           2481       3591
## Proportions:           0.010  0.294     0.006          0.691      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 4504.765, df = 3, p-value = 0.000
```

```
# The chart shows that most customers are self-employed or other occupations.
# There are very few salaried customers or entrepreneurs.
# This shows that different occupation does lead to
# different probabilities of taking credit cards (but with bias).


# Channel Code
PieChart(Channel_Code, hole = 0, values = "%", data = level.1,
         fill = c("pink","lightblue", "lightgreen", "grey"),
         main = "Channel Code Pie Chart")
```

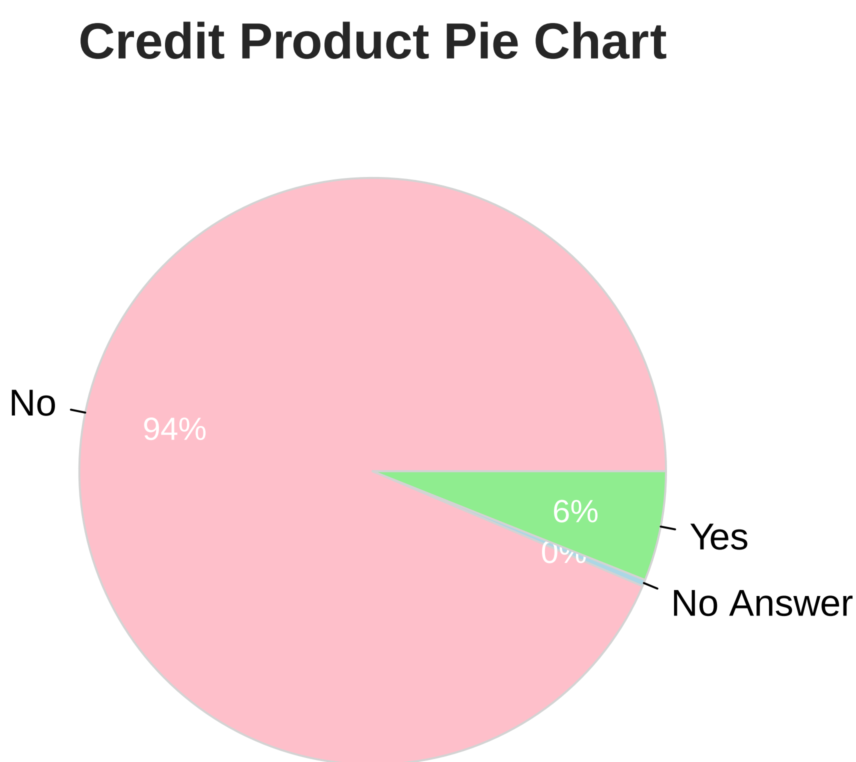



```
## >>> Suggestions
## PieChart(Channel_Code, hole=0)  # traditional pie chart
## PieChart(Channel_Code, values="%")  # display %'s on the chart
## BarChart(Channel_Code)  # bar chart
## Plot(Channel_Code)  # bubble plot
## Plot(Channel_Code, values="count")  # lollipop plot
##
##
## --- Channel_Code ---
##
##
##                    X1     X2     X3     X4      Total
## Frequencies:        1     37   3413    140       3591
## Proportions:    0.000  0.010  0.950  0.039      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 9407.651, df = 3, p-value = 0.000
```

```
# Surprisingly, almost all customers choose channel code X3.
# This may be investigated more in the future.


# Vintage
hist(level.1$Vintage,prob=TRUE)
lines(density(level.1$Vintage), lwd = 2, col ="red")
```

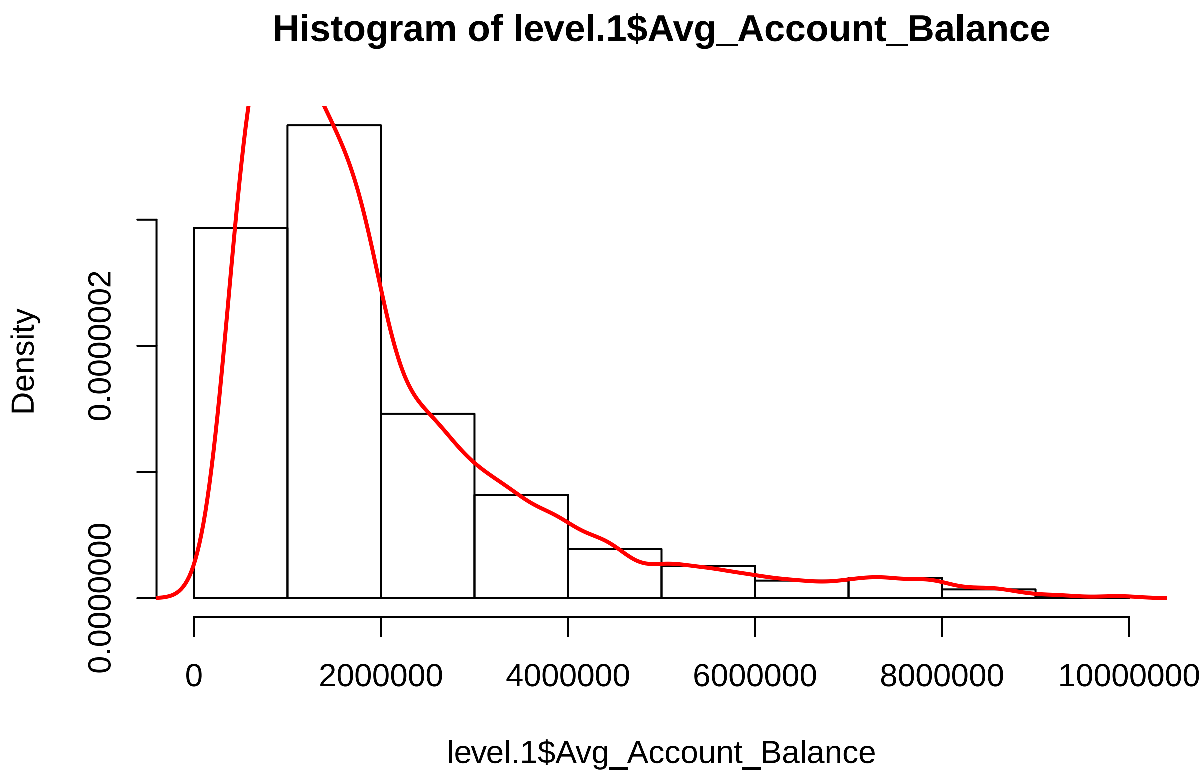



```
# The vintage is left-skewed and is clustered around 120.


# Credit Product
# We change "N/A" to "No Answer" because N/As are not counted in pie charts
level.1$Credit_Product[is.na(level.1$Credit_Product)] <- "No Answer"
PieChart(Credit_Product, hole = 0, values = "%", data = level.1,
         fill = c("pink","lightblue", "lightgreen"),
         main = "Credit Product Pie Chart")
```




```
## >>> Suggestions
## PieChart(Credit_Product, hole=0)  # traditional pie chart
## PieChart(Credit_Product, values="%")  # display %'s on the chart
## BarChart(Credit_Product)  # bar chart
## Plot(Credit_Product)  # bubble plot
## Plot(Credit_Product, values="count")  # lollipop plot
##
##
## --- Credit_Product ---
##
##
##                     No  No Answer    Yes      Total
## Frequencies:      3361         13    217       3591
## Proportions:     0.936      0.004  0.060      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 5885.674, df = 2, p-value = 0.000
```

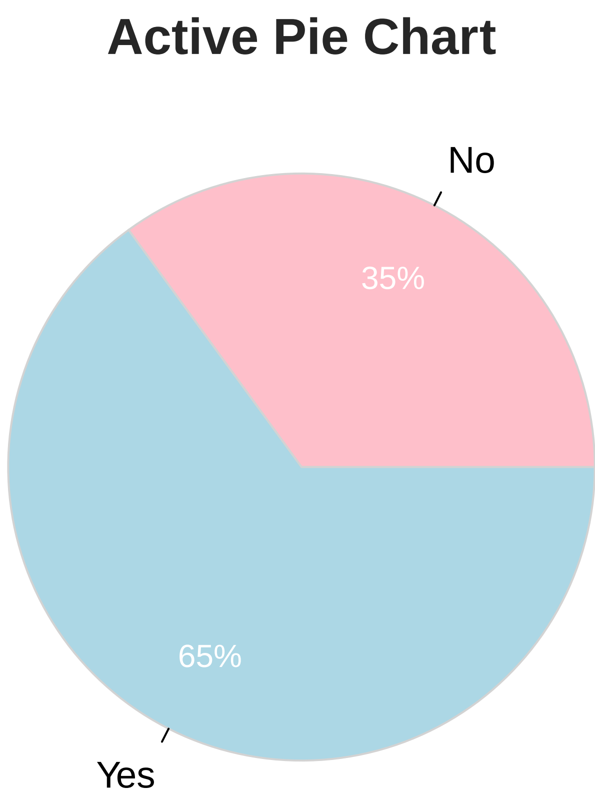
```
# 94% of the customers answered "No" while only 6% answered "Yes".


# Average Account Balance
hist(level.1$Avg_Account_Balance,prob=TRUE)
lines(density(level.1$Avg_Account_Balance), lwd = 2, col ="red")
```




```r
# The histogram is similar to what we had before.
# We calculate the mean to see if there is any changes.

mean(level.1$Avg_Account_Balance)
```

```
## [1] 1985474
```

```r
# Therefore, customers with average account balance of around 1985474
# have a higher probability of taking credit cards.


# Is active
PieChart(Is_Active, hole = 0, values = "%", data = level.1,
         fill = c("pink","lightblue"),
         main = "Active Pie Chart")
```




```
## >>> Suggestions
## PieChart(Is_Active, hole=0)  # traditional pie chart
## PieChart(Is_Active, values="%")  # display %'s on the chart
## BarChart(Is_Active)  # bar chart
## Plot(Is_Active)  # bubble plot
## Plot(Is_Active, values="count")  # lollipop plot
##
##
## --- Is_Active ---
##
##
##                    No    Yes      Total
## Frequencies:     1258   2333       3591
## Proportions:    0.350  0.650      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 321.811, df = 1, p-value = 0.000
```

```
# 65% of customers answered "Yes" while 35% answered "No"
```

```
Region <-as.factor(level.1$Region_Code)
Region <-Region[-length(Region)]
table(Region)
```

```
## Region
## RG250 RG251 RG252 RG253 RG254 RG255 RG256 RG257 RG258 RG259 RG260 RG261 RG262
##     9    72    14    41   505    11    12    35    10    17    23    39    22
## RG263 RG264 RG265 RG266 RG267 RG268 RG269 RG270 RG271 RG272 RG273 RG274 RG275
##    25    11    13     6     3   845    88    42     2    45    32    31    19
```

```
## RG276 RG277 RG278 RG279 RG280 RG281 RG282 RG283 RG284
##    21   127    21    33   137    45    57   736   441
```

```
# 845 most potential customers live in RG268
# The Happy Customer Bank could set more banking outlets
# for customers' convenience / attract more customers around RG268
```

From these graphs, we can see the specific range of the 3591 most recommended customers' information, specifically the share of their genders, occupations, channel codes, whether they have or have not active credit product, and whether they are active or not. We also have a more comprehensive understanding of the distribution of their ages, vintages, and average account balances. Hereto, we have chosen the best model for our data and used it to successfully identify potential customers, and have had a thorough understanding of the distributions of these customers.

Based on our analysis, we suggest that Happy Customer Bank could set more banking outlets for customers' convenience around RG268. Furthermore, medium-to-high income people have a higher probability of activating their account, so it is effective to set banking outlets in central city regions. Self-employed people consist of 70% of potential customers. Therefore, the Happy Customer Bank could develop some financing products or petty loans to attract them.

4. Conclusion

In conclusion, by using and comparing various models and analysis results, we have successfully identified the logit model as our best model, and come up with a recommended list of potential customers who are most likely to take credit cards from Happy Customer Bank.

The best model was chosen with some difficulty, as throughout our analysis, a range of possible models seemed suitable at first sight. For example, OLS model, probit model, and logit model showed similar results in their summary, their model accuracy, their average marginal effect for each variable, etc. All estimators were statistically significant in these three models, and the signs of the estimators were consistent. The logit model showed slightly better model accuracy. To evaluate their performance better, we divided the data into training data. And logit model still had higher accuracy as well as lower RMSE.

However, this was far from the time when we determined the logit model as our best model. A large range of tests were provided. For example, RESET test was used to check the model specification, and BP test was used to detect heteroskedasticity. Logit model passed most of the tests. Furthermore, a multinomial logit model and an IV model were used to challenge logit model to see if they were better, but they failed because other tests have shown that they were either questionable or not well specified.

Although we have identified logit model as our best model, we still have many shortcomings at present, such as heteroskedasticity. Though we converted our standard errors into robust standard errors, this did not help much.

Nevertheless, applying the logit model to our data, lists of recommended customers sorted according to their level of interest in taking credit cards came out. Based on our result, there are 3591 potential customers that are most likely to activate a recommended credit card. Top recommended customers turn out to be mostly males, aged between 40 and 80, self-employed, and are those who tend to choose channel code X3, have no active credit product, have a vintage of about 120 and average account balances between 2 million and 10 million dollars. Most are active users and live in RG268. These results show the common features of the most recommended customers and we hope it will help Happy Customer Bank make future decisions about recommending credit cards and other financial products.

5. Future Work

For future work, to improve our model, more data collection is necessary to minimize the standard errors of each term. This could help us to have a more unbiased confidence interval and better hypothesis tests. Also, according to the law of large numbers, as more data are collected, our sample means for each variable will become closer and closer to the true expected results.

For data collection, it can be more accurate and logical. This may help our model greatly. For example, the variable "Occupation" can be numbered in some way (e.g., based on the average income) to become an ordered variable. Then we can see a more reliable relationship between "Occupation" and "Is_Active", and may discover more correlations among variables, which may help improve our instrumental variable model.

In addition, if the variable "Credit_Product" only contains "Yes" or "No" (there is no such option as "N/A"), we can model it as a dummy variable. This may help us to make a better analysis in multinomial logit model, so as to truly realize the hope of cross-selling. Frequencies for each channel code can be included to make it into count data, so a Poisson regression model can be used.

Finally, we noticed that R^2 in the OLS is relatively low, which means our independent variables are poorly explained by the explanatory variables. In order to improve our model, it is also necessary to detect some new variables, such as pre-retirement wealth or debt, and then, check their feasibility. In addition, choosing better functional forms is also necessary. We could perform the Jarque-Bera Test for normality in this case, as more data are added.

6. References

https://www.kaggle.com/sajidhussain3/jobathon-may-2021-credit-card-lead-prediction?select=test.csv