Customer Churning

Lorraine Schemenauer

03/16/2023

Customer Churning

##Objective Our objective is to determine whether or not a customer will get churned. By looking at the predictors: months on book, credit limit, customer age, dependent count, and average open to buy, we can find the probability that an existing customer will get churned or not.

We can predict that customers that are older are going to have more experience and make better financial decisions compared to younger customers. A higher credit limit means that banks trust the customer and they are less likely to be churned. We can expect that people with less dependents are more likely to not be churned since they have less expenses. The higher the average open to buy, which is the difference between the credit limit and the present balance a person has on their account, the less likely a customer will be churned.

##Preparing Data When preparing our data, we changed Attrition flag to a binary variable where 1 represents Existing Customer and 0 for the Attrited Customer. The binary data allows us to obtain the probability of a customer staying with the bank.

```
Bank <- read.csv("~/Downloads/Bank.csv", header=FALSE, skip=1)
View(Bank)
colnames(Bank)
```

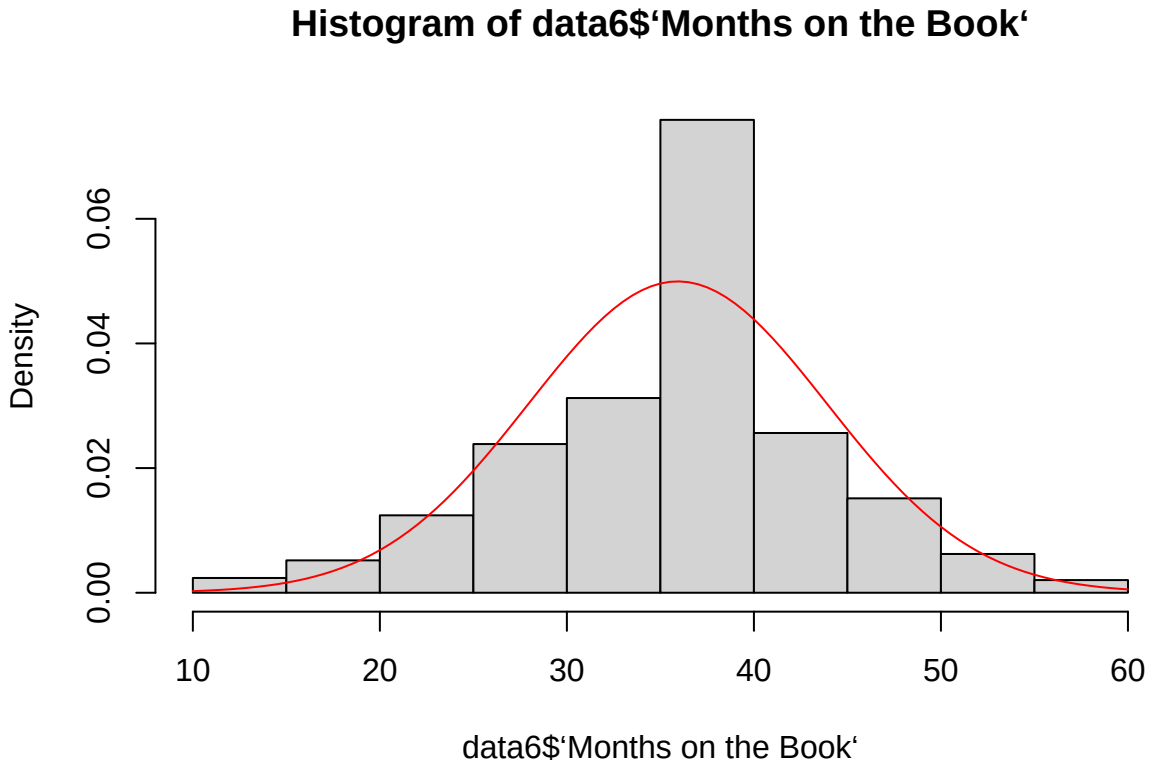
```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23"
```

```
data6<-data.frame(Bank$V2, Bank$V10, Bank$V14, Bank$V3, Bank$V5, Bank$V16)
colnames(data6)<-c("Attrition Flag", "Months on the Book", "Credit Limit", "Customer Age", "Dependent C

for(i in 1:nrow(data6))
  {
    if(data6$`Attrition Flag` [i] == "Existing Customer")
    {
      data6$`Attrition Flag` [i]=1
    }
    else
    {
      data6$`Attrition Flag` [i]=0
    }
  }
View(data6)
```

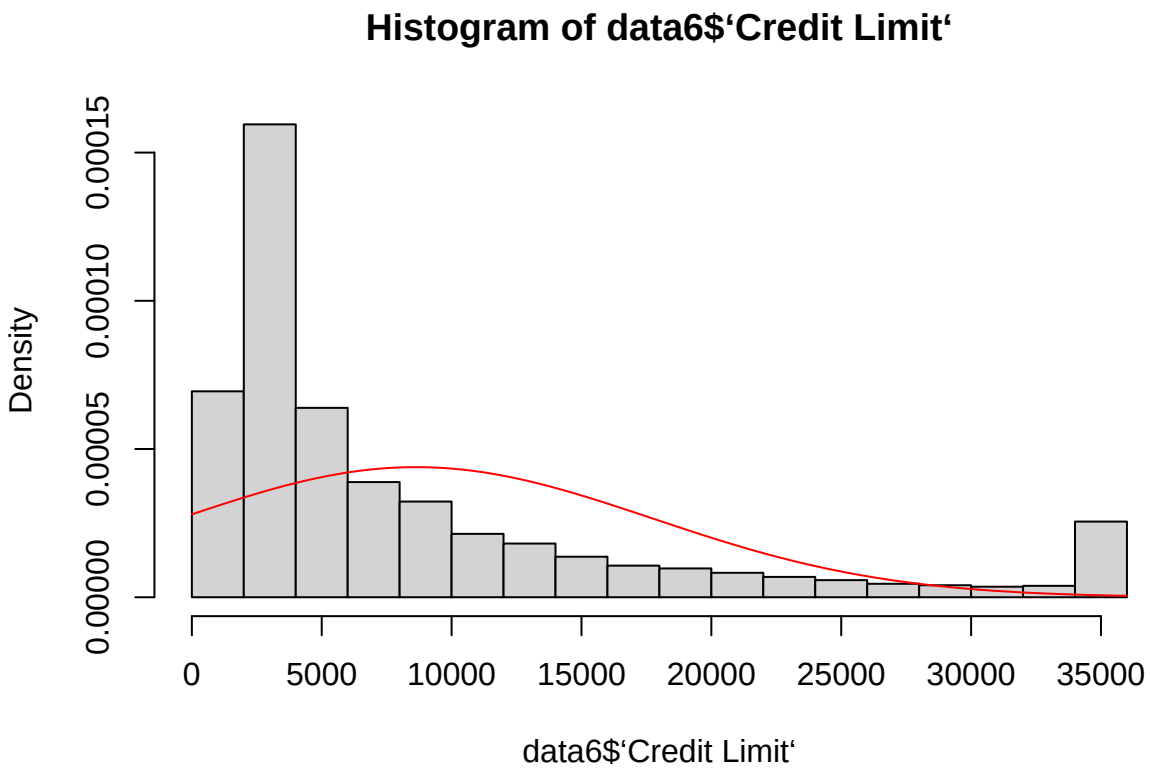
#Descriptive Statistics ## Histograms Months of the book is normally distributed with a mean around 35. This means people are with the bank for around 35 months, around 3 years. Credit limit is skewed right

with a mean around 6000. This means that $6000 is a frequent credit limit for customers. Customer age is normally distributed with a mean at about 45 years old. This could mean that less people will be churned as they are higher in age with more experience. The dependent count is normally distributed with about 2 or 3 dependents. This could be a sign that having children means people are more responsible and reliable so it is possible less people in this bank will be churned. Average open to buy is skewed right with a mean of 5000. This means people are not reaching their credit limit so less people will be churned. The attrition flag is skewed left with the mean at 1. This means people are less likely to be churned at this bank based on what we have already described from the predictors.

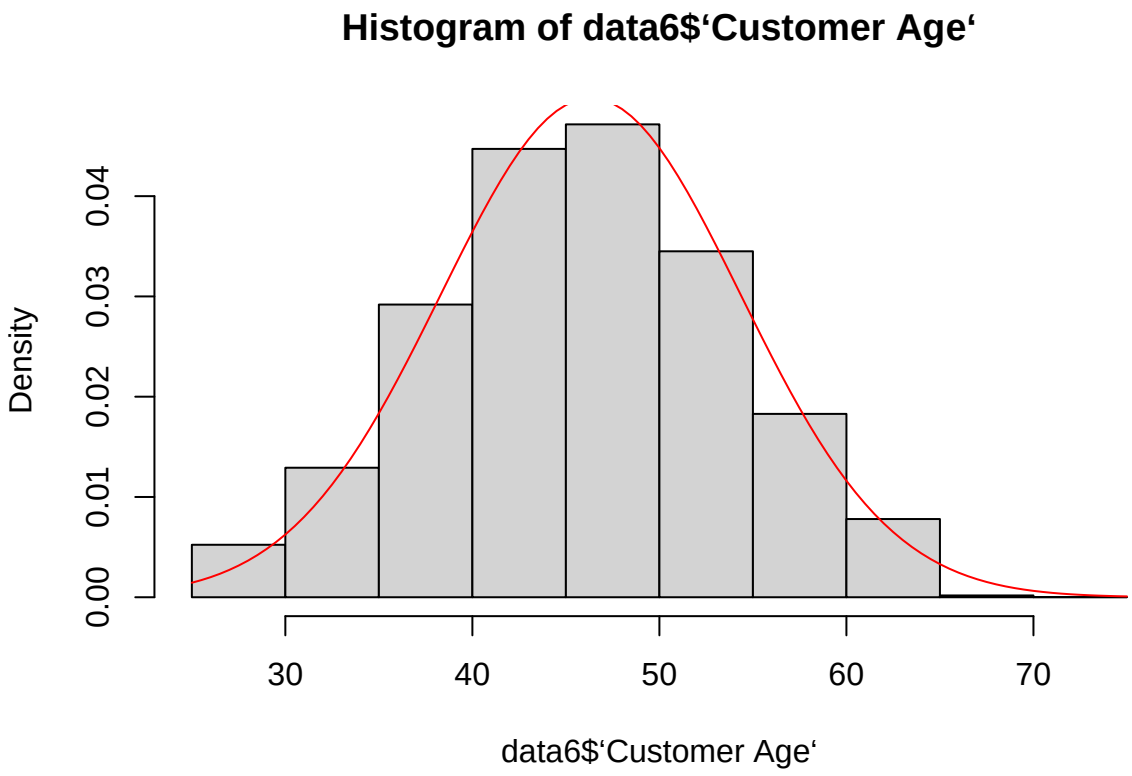
```
library(MASS)
hist(data6$`Months on the Book`, prob = TRUE)
fit1<-fitdistr(data6$`Months on the Book`, densfun="normal")
curve(dnorm(x,fit1$estimate[1], fit1$estimate[2]), col="red", add=T)
```



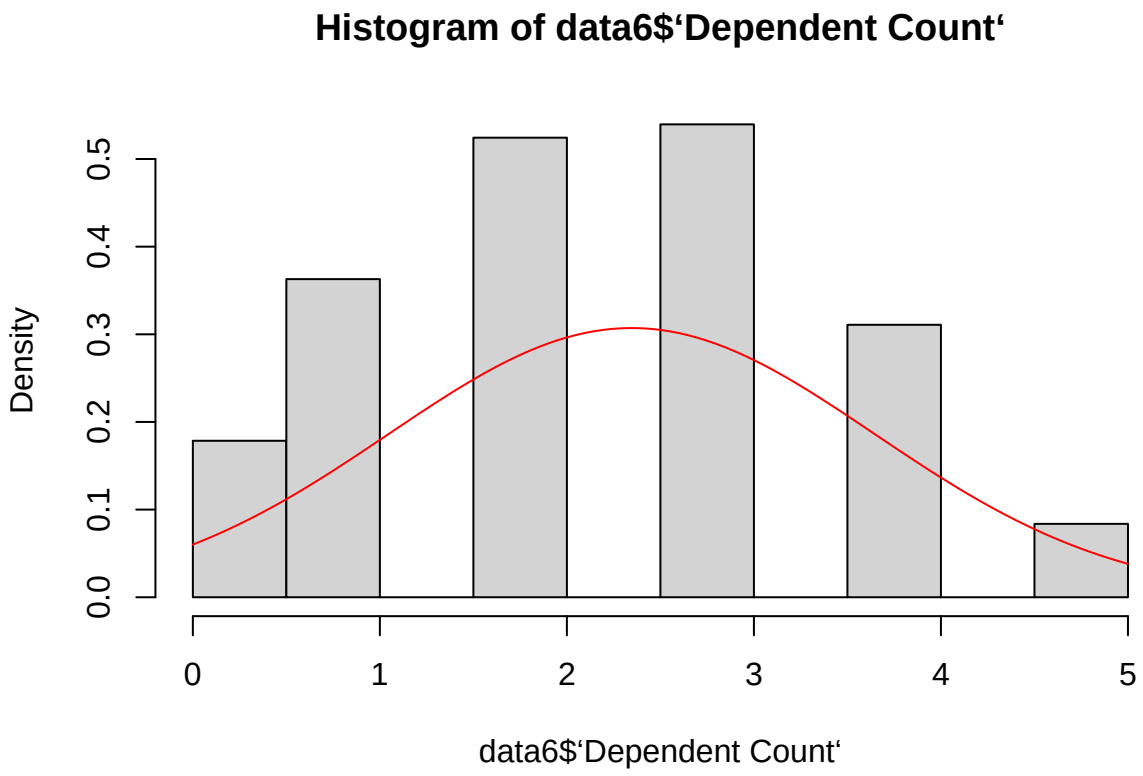

```
hist(data6$`Credit Limit`, prob = TRUE)
fit1<-fitdistr(data6$`Credit Limit`, densfun="normal")
curve(dnorm(x,fit1$estimate[1], fit1$estimate[2]), col="red", add=T)
```



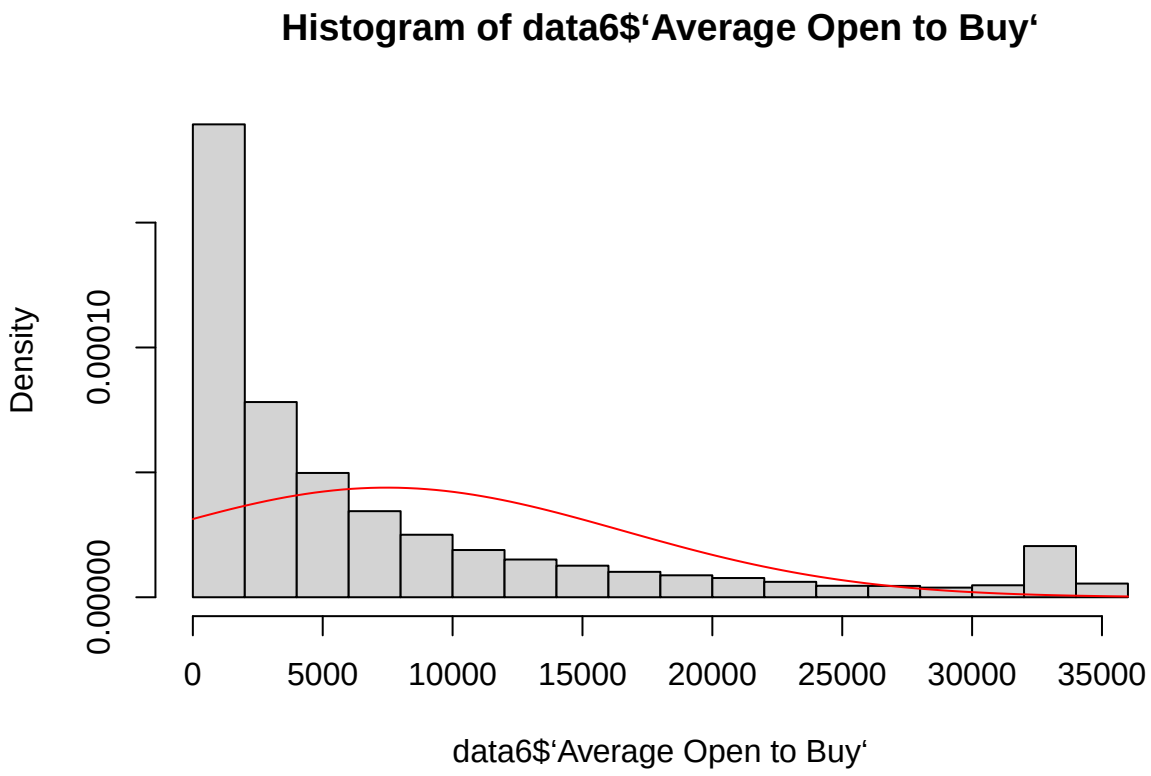

```
hist(data6$`Customer Age`, prob = TRUE)
fit1<-fitdistr(data6$`Customer Age`, densfun="normal")
curve(dnorm(x,fit1$estimate[1], fit1$estimate[2]), col="red", add=T)
```



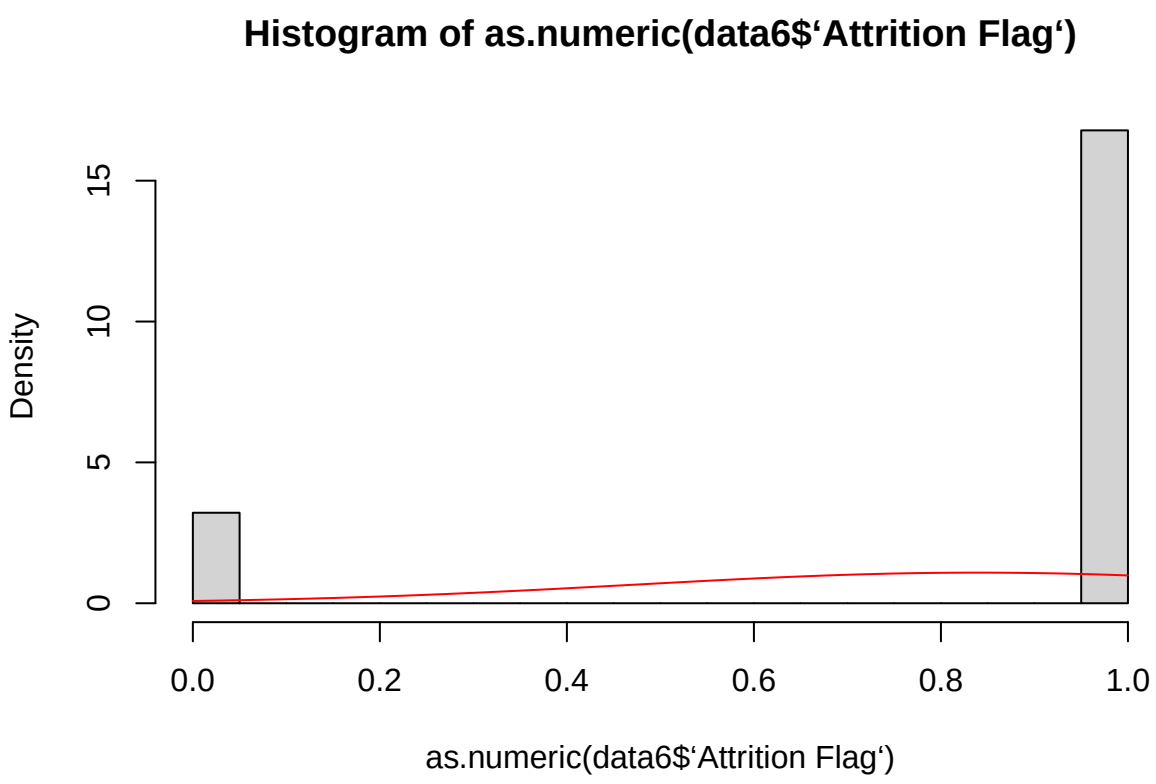

```
hist(data6$`Dependent Count`, prob = TRUE)
fit1<-fitdistr(data6$`Dependent Count`, densfun="normal")
curve(dnorm(x,fit1$estimate[1], fit1$estimate[2]), col="red", add=T)
```




```
hist(data6$`Average Open to Buy`, prob = TRUE)
fit1<-fitdistr(data6$`Average Open to Buy`, densfun="normal")
curve(dnorm(x,fit1$estimate[1], fit1$estimate[2]), col="red", add=T)
```



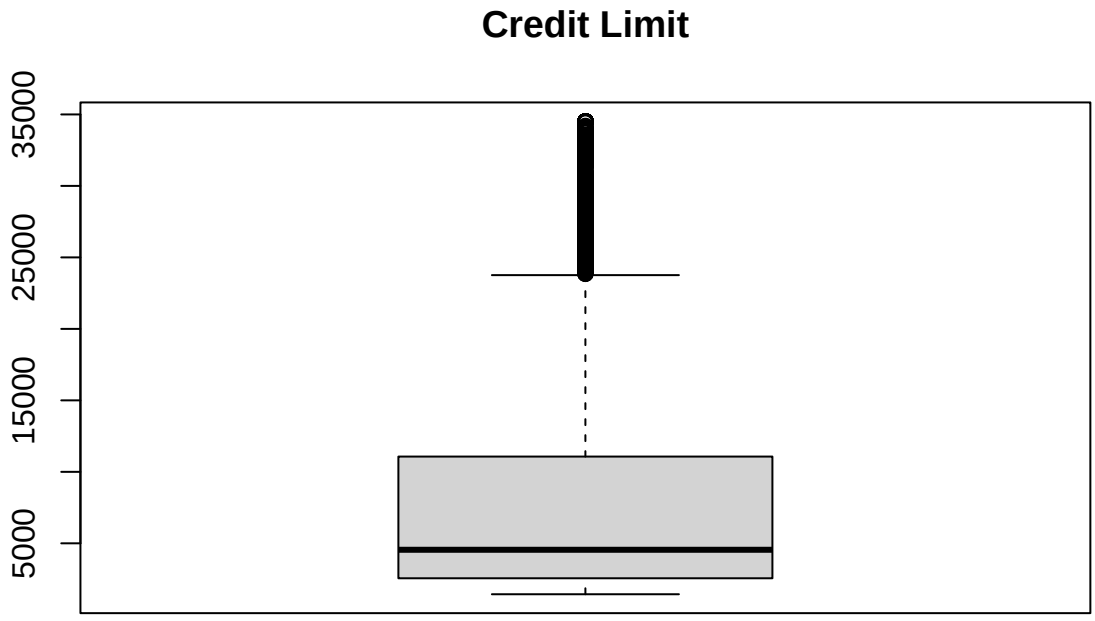

```
hist(as.numeric(data6$`Attrition Flag`), prob = TRUE)
fit1<-fitdistr(as.numeric(data6$`Attrition Flag`), densfun="normal")
curve(dnorm(x,fit1$estimate[1], fit1$estimate[2]), col="red", add=T)
```



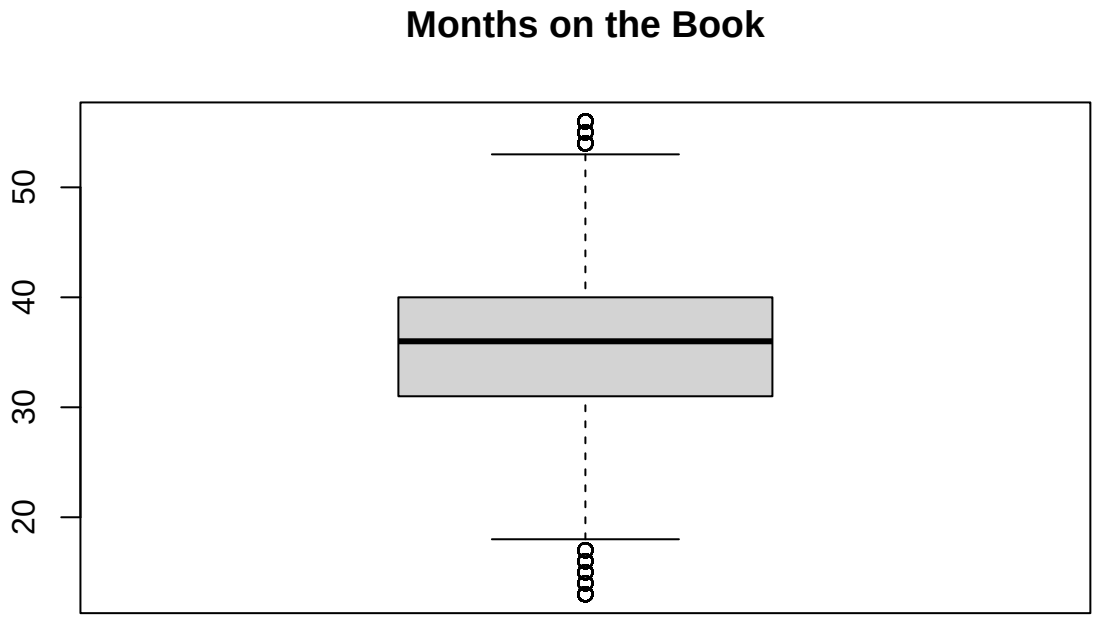
Box Plots

Credit limit has a median around 5000 but a lot of outliers up to 35000. This makes sense since the average customer is around 45 years old, so they are able to obtain a higher credit limit. Months on the book has a median of 35 months with 25% lower quartile being around 20 months and the upper quartile at around 55 months. The attrition flag again shows that majority customers will not be churned. The customer age has a median at 45 years old. The dependent count has a median at 2. The average open to buy has a lot of outliers which is a good sign that less people will be churned.

```
boxplot(data6$`Credit Limit`, main = "Credit Limit")
```

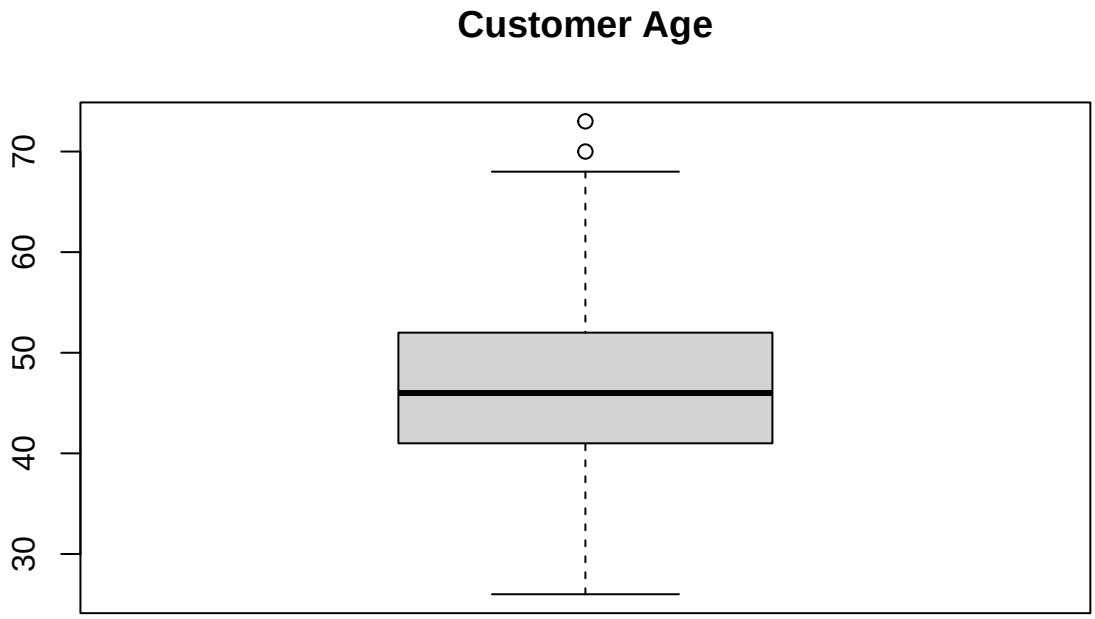



```
boxplot(data6$`Months on the Book`, main = "Months on the Book")
```

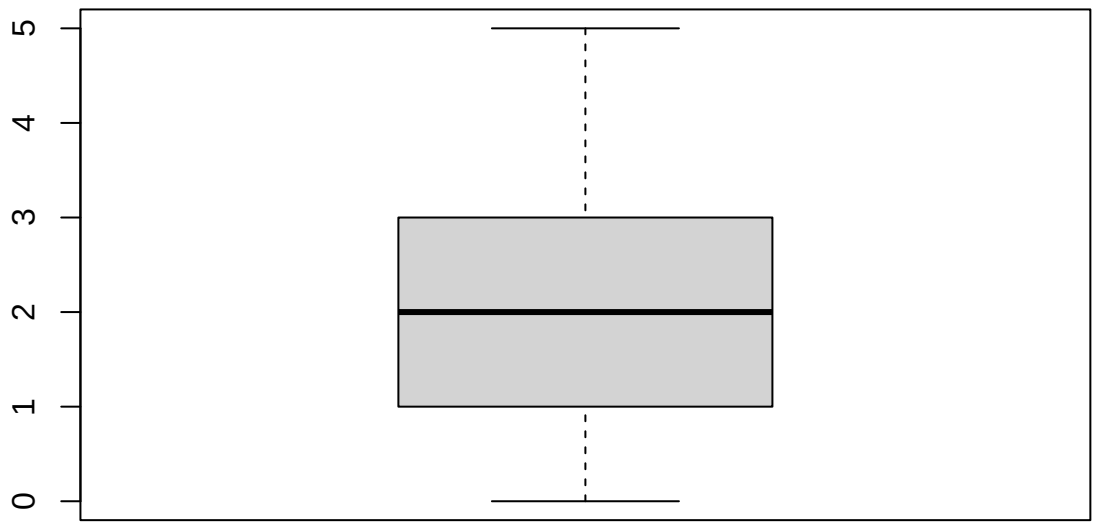



```
boxplot(as.numeric(data6$`Attrition Flag`), main = "Attrition Flag")
```

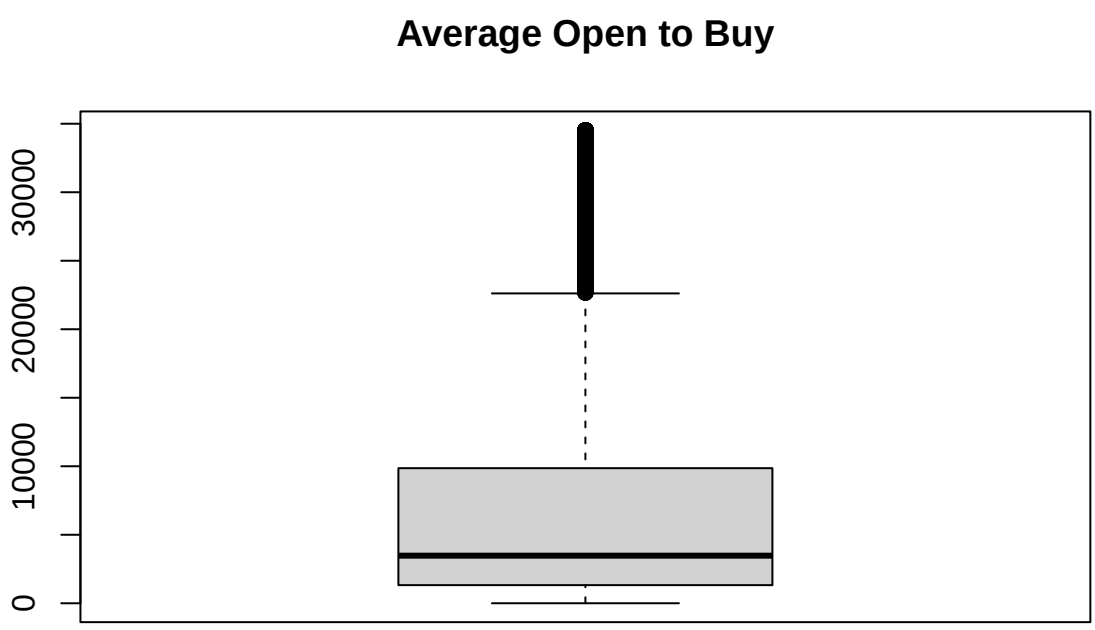

```
boxplot(data6$`Customer Age`, main = "Customer Age")
```




```
boxplot(data6$`Dependent Count`, main = "Dependent Count")
```




```
boxplot(data6$`Average Open to Buy`, main = "Average Open to Buy")
```



Scatterplots

For months on the book versus attrition flag, there the scatter plot does not show a difference between someone who is churned versus someone who is depending on the months of the bank. For credit limit versus attrition flag, someone who is going to be churned has a lower credit limit while someone who is not going to be churned has a higher credit limit. For age versus attrition flag, those who are not going to be churned have more outliers, meaning that older people are less likely to be churned. For dependent count versus attrition flag, the scatter plot does not show a difference between someone being churned and someone not being churned. This could mean that the dependent count could not have as much of an influence. For average open to buy versus attrition flag, the scatter plot has a higher concentration of average open to buy people for people who are not going to be churned.

```
library(car)
```

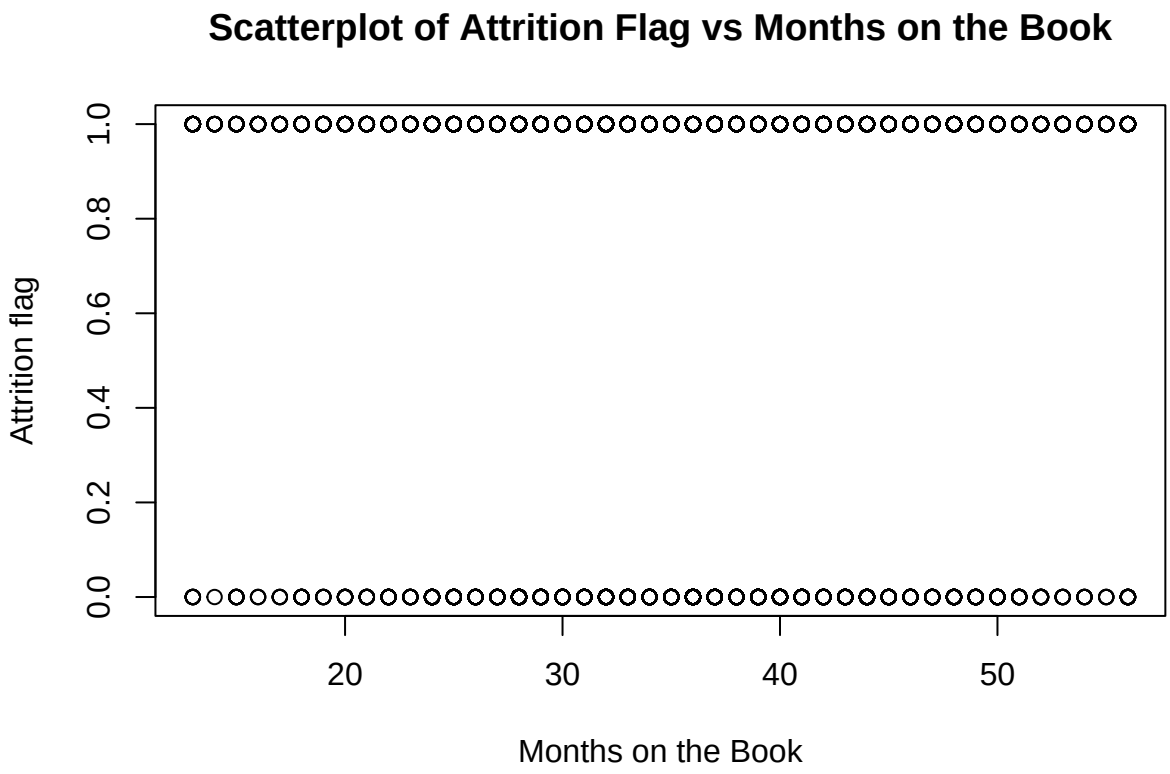
```
## Loading required package: carData
```

```
library(gplots)
```

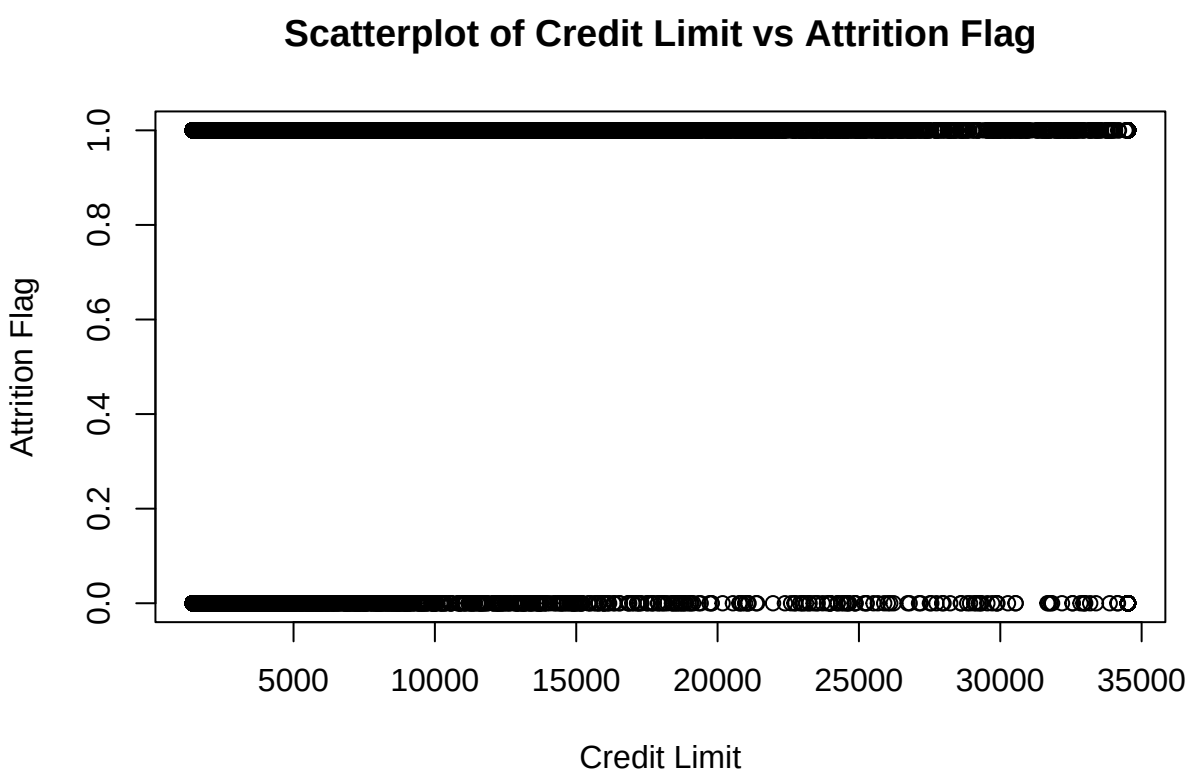
```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

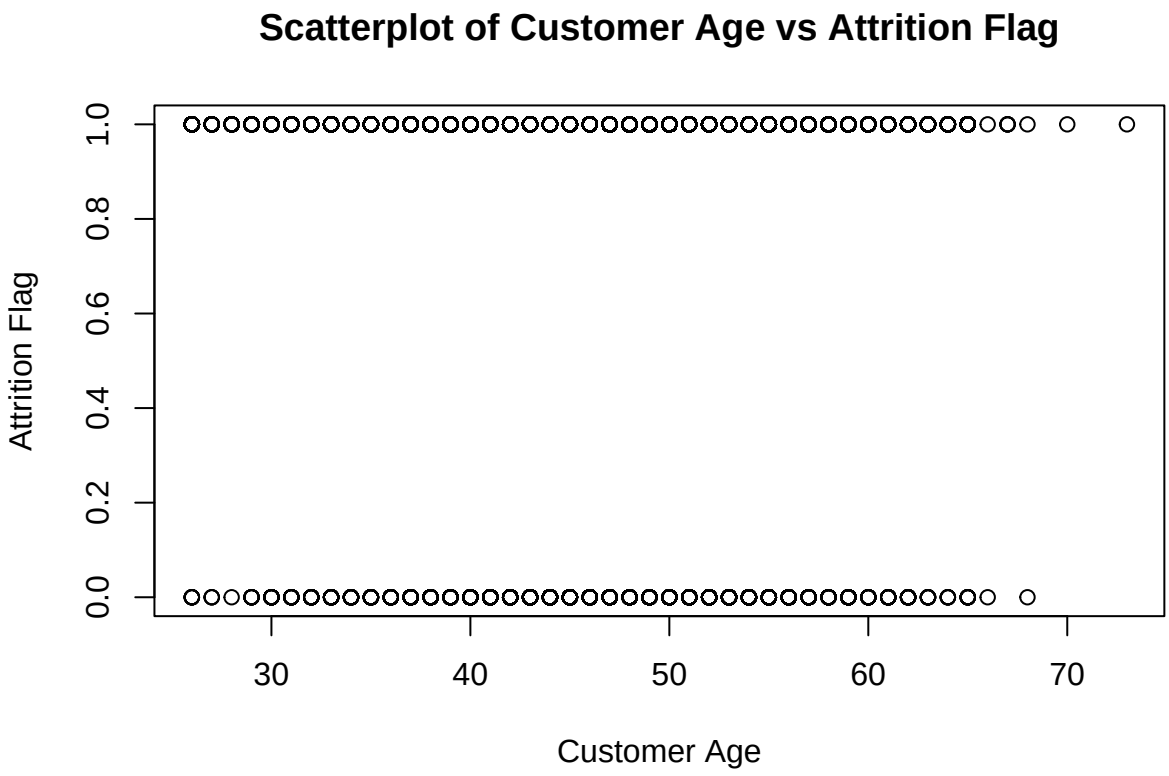
```
plot(data6$`Months on the Book`, data6$`Attrition Flag`, xlab="Months on the Book", ylab="Attrition flag
```



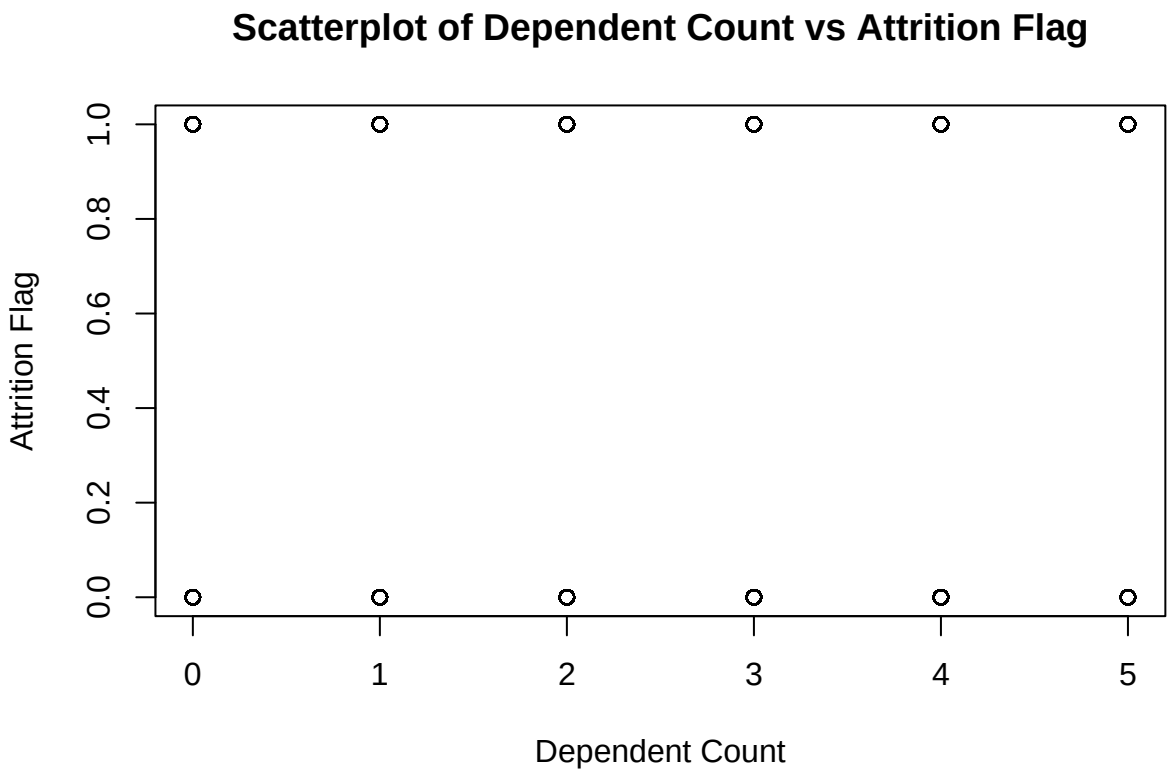

```
plot(data6$`Credit Limit`, data6$`Attrition Flag`, xlab="Credit Limit", ylab="Attrition Flag", main="Sca
```



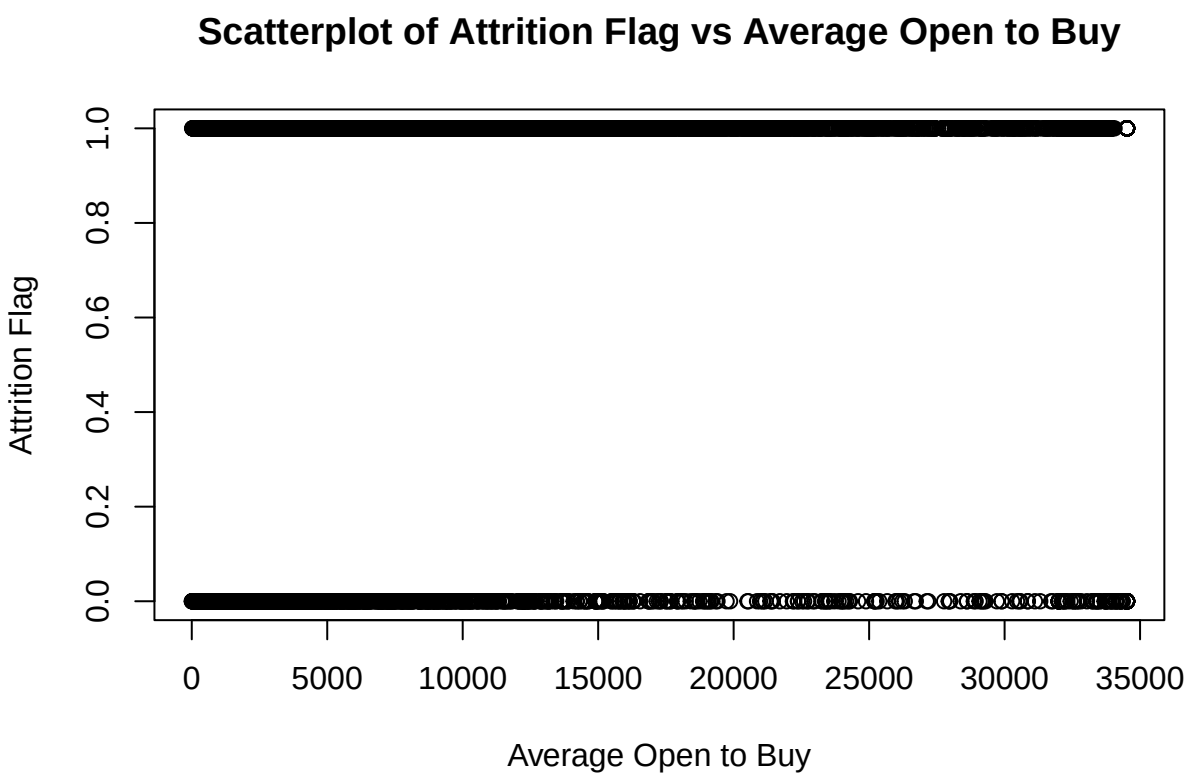

```
plot(data6$`Customer Age`, data6$`Attrition Flag`, xlab="Customer Age", ylab="Attrition Flag", main="Sca
```




```
plot(data6$`Dependent Count`, data6$`Attrition Flag`, xlab="Dependent Count", ylab="Attrition Flag", ma
```




```
plot(data6$`Average Open to Buy`, data6$`Attrition Flag`, xlab="Average Open to Buy", ylab="Attrition Fl
```



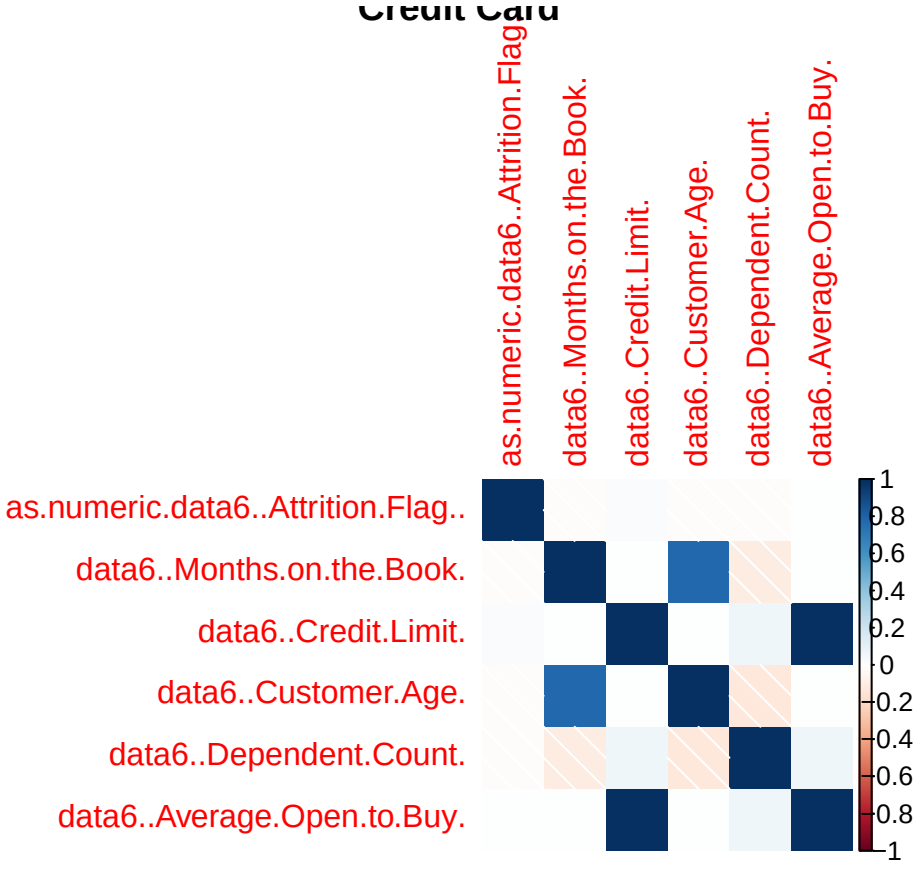
Correlation Plots

Attrition flag has only a weakly positive correlation with credit limit. The month on the book has a positive correlation with customer age and negative correlation with dependent count. The credit limit has a strong positive correlation with average open to buy and a slight positive correlation with dependent count. The customer age has a strong postitive correlation with months on the book and a slight negative correlation with dependent count. Dependent count has low negative correlation with months on the book and customer age while it also has a slight positve correlation with credit limit and average open to buy. The average open to buy has a strong positive correlation with credit limit and a slight positive correlation with dependent count.

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrborr<-data.frame(as.numeric(data6$`Attrition Flag`), data6$`Months on the Book`, data6$`Credit Limi
M1 = cor(corrborr)
corrplot(M1, method = 'shade', main = "Credit Card")
```



Statistical Summary

The statistical summary confirms what we visually saw in the box plot, scatter plot, and histograms. The means of this output leads us to suspect that people are less likely to be churned at this bank. The following models will help us confirm our hypothesis.

```
summary(data6)
```

```
##  Attrition Flag     Months on the Book  Credit Limit    Customer Age
##  Length:10127       Min.   :13.00      Min.   : 1438   Min.   :26.00
##  Class :character   1st Qu.:31.00      1st Qu.: 2555   1st Qu.:41.00
##  Mode  :character   Median :36.00      Median : 4549   Median :46.00
##                     Mean   :35.93      Mean   : 8632   Mean   :46.33
##                     3rd Qu.:40.00      3rd Qu.:11068   3rd Qu.:52.00
##                     Max.   :56.00      Max.   :34516   Max.   :73.00
##  Dependent Count Average Open to Buy
##  Min.   :0.000   Min.   :    3
##  1st Qu.:1.000   1st Qu.: 1324
##  Median :2.000   Median : 3474
##  Mean   :2.346   Mean   : 7469
##  3rd Qu.:3.000   3rd Qu.: 9859
##  Max.   :5.000   Max.   :34516
```

Linear Probability Model

The linear probability model is our typical OLS. The variables credit limit, dependent count, average open to buy, and customer age (with .1 significance level) are statistically significant, given the low p value. With the linear probability model, we can interpret the betas directly. For example, a unit change in credit limit means a 1.190e-04 percent increase in attrition flag.

```
linearProb <- lm(`Attrition Flag`~ `Months on the Book`+`Credit Limit`+ `Customer Age`+ `Dependent Coun
summary(linearProb)
```

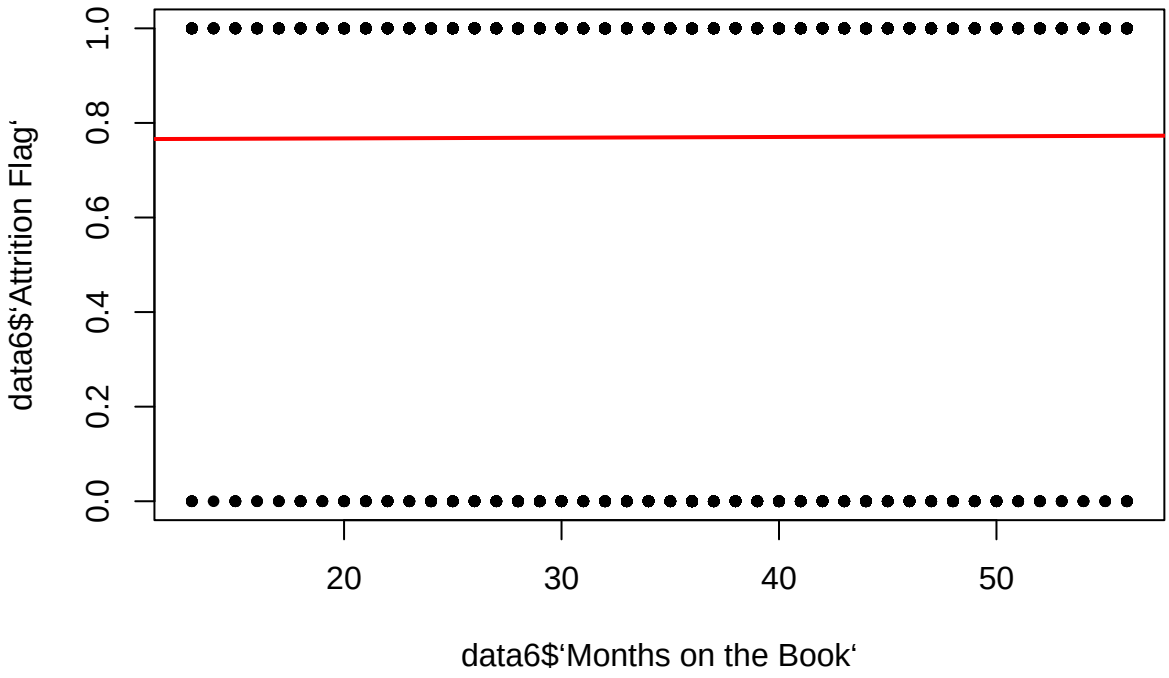
```
##
## Call:
## lm(formula = `Attrition Flag` ~ `Months on the Book` + `Credit Limit` +
##     `Customer Age` + `Dependent Count` + `Average Open to Buy`,
##     data = data6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04145  0.03852  0.11730  0.18824  0.32684
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             7.643e-01  2.311e-02  33.071   <2e-16 ***
## `Months on the Book`    1.516e-04  7.172e-04   0.211   0.8326
## `Credit Limit`          1.190e-04  4.324e-06  27.516   <2e-16 ***
## `Customer Age`         -1.257e-03  7.161e-04  -1.756   0.0792 .
## `Dependent Count`      -6.297e-03  2.737e-03  -2.301   0.0214 *
## `Average Open to Buy` -1.184e-04  4.323e-06 -27.388   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3542 on 10121 degrees of freedom
## Multiple R-squared:  0.07034,    Adjusted R-squared:  0.06988
## F-statistic: 153.2 on 5 and 10121 DF,  p-value: < 2.2e-16
```

LPM Plots

When we plot our LPM for months on the book, customer age, and dependent count, our fitted value line has a very small slope with a large intercept. Comparing our points to this line gives us an idea of how well we classified 0 and 1. The y hat values are closer to what we classified as 1 while further away from what we classified as 0. This matches what we assigned as 0 is more likely to be misclassified. This conclusion also matches our results from our confusion matrix. For the credit limit and average open to buy, the y hats pass the 0 and 1 bounds, given it is a LPM. This can be fixed later with the probit/logit model.

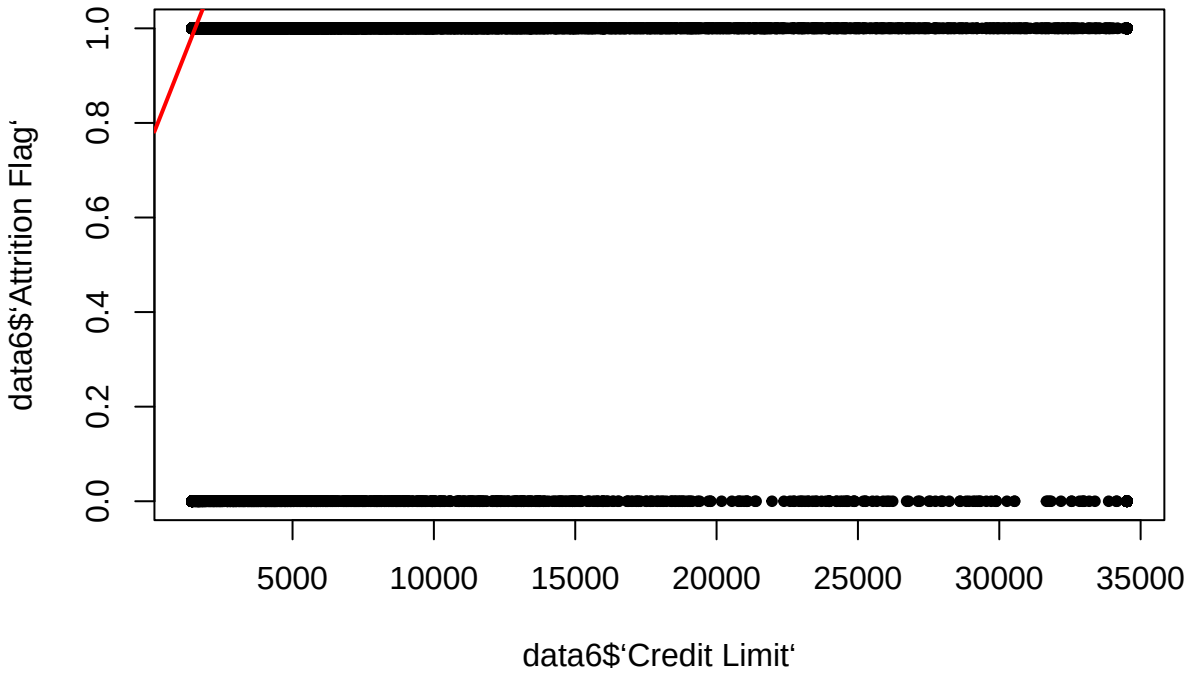
```
plot(data6$`Months on the Book`, data6$`Attrition Flag`, pch=20)
abline(linearProb, col="red", lwd=2)
```

```
## Warning in abline(linearProb, col = "red", lwd = 2): only using the first two
## of 6 regression coefficients
```



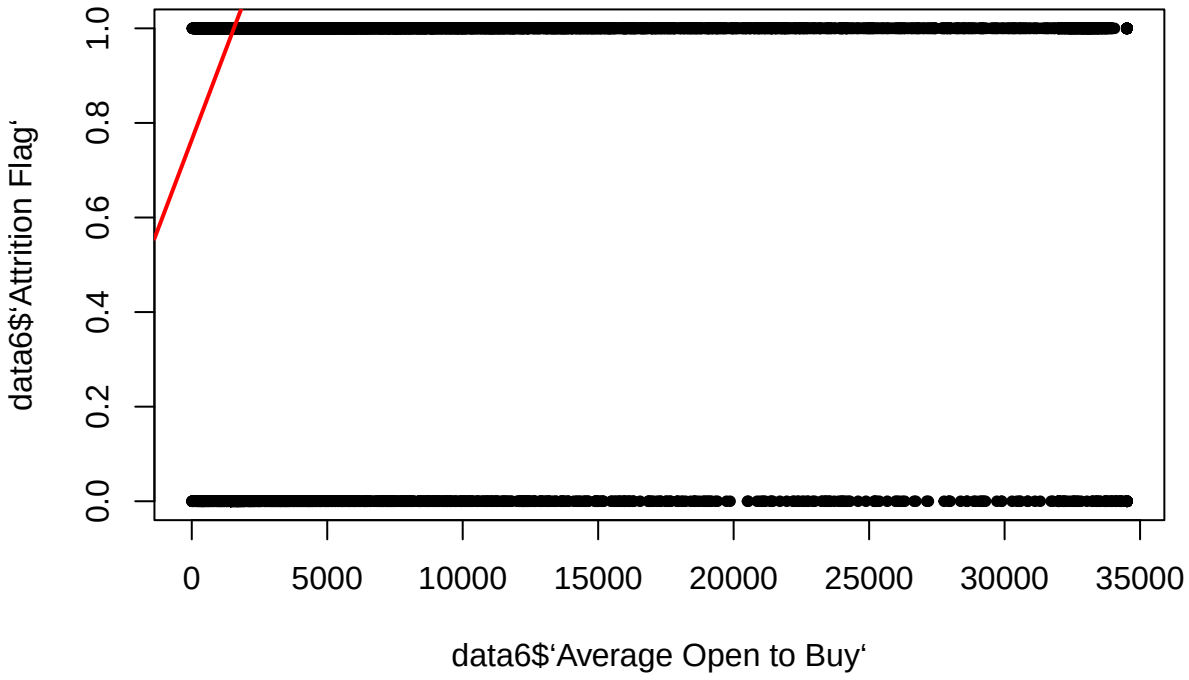

```
plot(data6$`Credit Limit`, data6$`Attrition Flag`, pch=20)
abline(linearProb, col="red", lwd=2)
```

```
## Warning in abline(linearProb, col = "red", lwd = 2): only using the first two
## of 6 regression coefficients
```



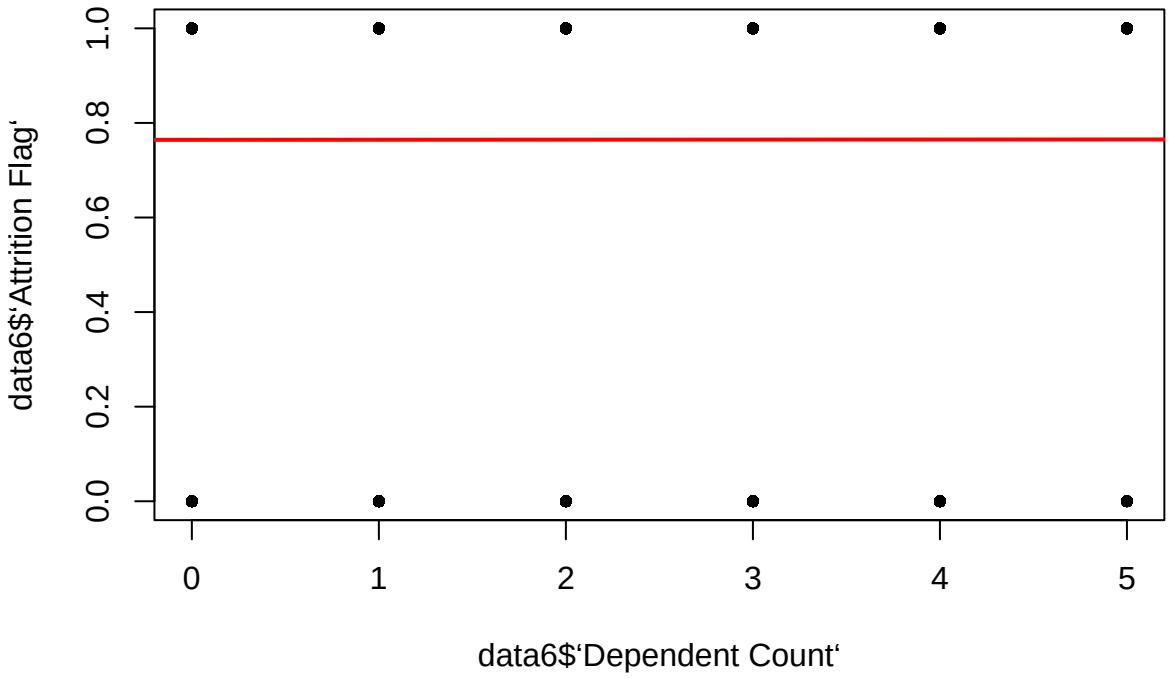

```
plot(data6$`Average Open to Buy`, data6$`Attrition Flag`, pch=20)
abline(linearProb, col="red", lwd=2)
```

```
## Warning in abline(linearProb, col = "red", lwd = 2): only using the first two
## of 6 regression coefficients
```



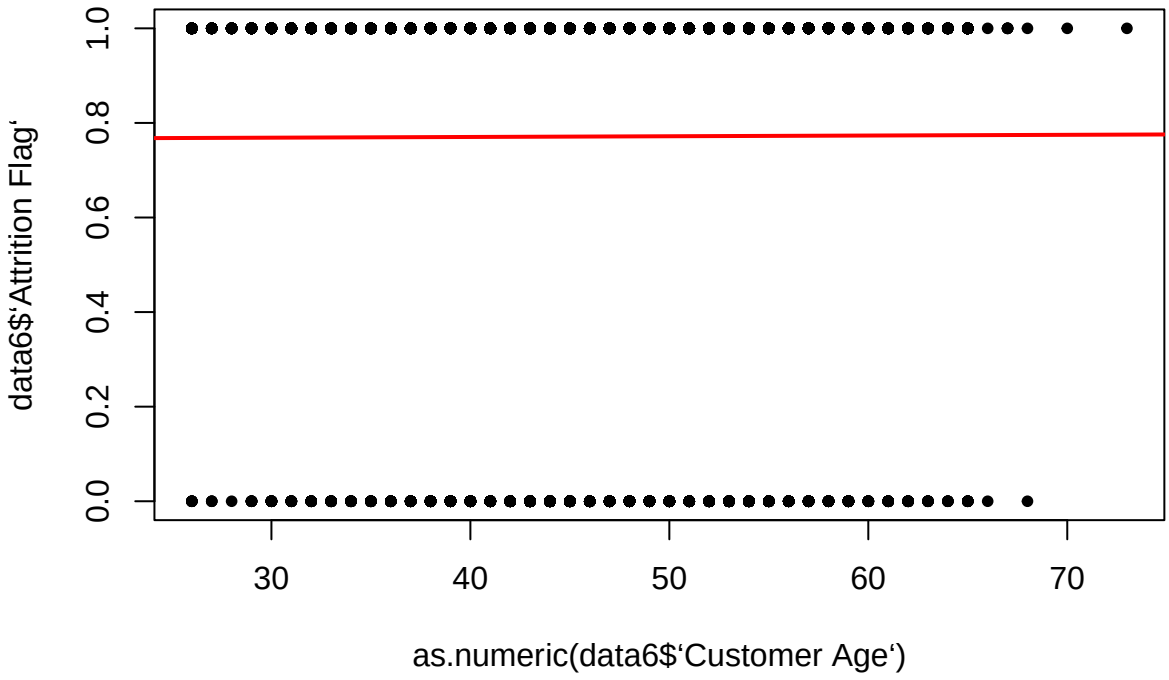

```
plot(data6$`Dependent Count`, data6$`Attrition Flag`, pch=20)
abline(linearProb, col="red", lwd=2)
```

```
## Warning in abline(linearProb, col = "red", lwd = 2): only using the first two
## of 6 regression coefficients
```




```
plot(as.numeric(data6$`Customer Age`), data6$`Attrition Flag`, pch=20)
abline(linearProb, col="red", lwd=2)
```

```
## Warning in abline(linearProb, col = "red", lwd = 2): only using the first two
## of 6 regression coefficients
```



Confusion Matrix for LPM

When conducting the confusion matrix with a threshold of 0.5, the row for 0 did not show so we increased our threshold to .7 to get a full matrix. With the 0.7 threshold, we obtain an accuracy of 80%. We got this by adding what was classified correclty divided by the total.

```
#confusion matrix
confint(linearProb)
```

```
##                                2.5 %        97.5 %
## (Intercept)             0.7189549056  0.8095531971
## `Months on the Book`   -0.0012543009  0.0015575958
## `Credit Limit`          0.0001105026  0.0001274542
## `Customer Age`         -0.0026609934  0.0001464439
## `Dependent Count`      -0.0116613082 -0.0009317755
## `Average Open to Buy` -0.0001268760 -0.0001099279
```

```
ols.pred.classes <- ifelse(fitted(linearProb) > .7, 1, 0)
table(ols.pred.classes, data6$`Attrition Flag`)
```

```
##
## ols.pred.classes    0    1
##                0  479  801
##                1 1148 7699
```

Probit Model

The probit model is better than the LPM given that it bounds the intervals between 0 and 1 and it is able to capture nonlinear values of x. From our statistical summary, we can only comment on the direction of the betas for interpretation. Thus, customer age, dependent count, and average to buy has a negative effect while months on the book and credit limit have a positive effect To further interpret our betas, we can find the average marginal effects. This gives the direct estimates of our variables. For example, a unit increase in credit limit will have a 1.067924e-04 percentage increase on the probability that a customer will not be churned.

```
probit.mod = glm(as.numeric(`Attrition Flag`)~ `Months on the Book`+`Credit Limit`+ `Customer Age`+ `Dep
summary(probit.mod)
```

```
##
## Call:
## glm(formula = as.numeric(‘Attrition Flag‘) ~ ‘Months on the Book‘ +
##     ‘Credit Limit‘ + ‘Customer Age‘ + ‘Dependent Count‘ + ‘Average Open to Buy‘,
##     family = binomial(link = "probit"), data = data6)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6396   0.3553   0.4795   0.6112   0.9273
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             7.881e-01  1.030e-01   7.654 1.95e-14 ***
## ‘Months on the Book‘    1.450e-04  3.170e-03   0.046   0.9635
## ‘Credit Limit‘          4.693e-04  1.954e-05  24.015  < 2e-16 ***
## ‘Customer Age‘         -4.912e-03  3.179e-03  -1.545   0.1223
## ‘Dependent Count‘      -2.942e-02  1.209e-02  -2.434   0.0149 *
## ‘Average Open to Buy‘ -4.666e-04  1.953e-05 -23.896  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8927.2  on 10126  degrees of freedom
## Residual deviance: 8254.0  on 10121  degrees of freedom
## AIC: 8266
##
## Number of Fisher Scoring iterations: 4
```

```
confint(probit.mod)
```

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

##                              2.5 %        97.5 %
## (Intercept)             0.5858231705  0.9916372162
## ‘Months on the Book‘   -0.0060649341  0.0063449633
## ‘Credit Limit‘          0.0004328683  0.0005058556
## ‘Customer Age‘         -0.0111496926  0.0013208451
## ‘Dependent Count‘      -0.0532691275 -0.0055913221
## ‘Average Open to Buy‘ -0.0005031461 -0.0004302411
```

```
sum_phi <- mean(dnorm(predict(probit.mod,, type = "link")))
ame = sum_phi*coef(probit.mod)
ame
```

```
##           (Intercept)  `Months on the Book`        `Credit Limit`
##          1.793345e-01          3.299468e-05          1.067924e-04
##        `Customer Age`     `Dependent Count` `Average Open to Buy`
##         -1.117840e-03         -6.694493e-03         -1.061855e-04
```

Confusion Matrix

The confusion matrix gives us an accuracy level of 80%. This is the same accuracy as our linear probability model. Therefore, we can conclude that the probit model will be preferred compared to the linear probability model.

```
#confusion matrix
probit.mod.classes <- ifelse(fitted(probit.mod ) > 0.7, 1, 0)
table(probit.mod.classes, data6$`Attrition Flag`)
```

```
##
## probit.mod.classes    0    1
##                  0  571  927
##                  1 1056 7573
```

Probit model plots

The plot shows that the probit model is able to fit a nonlinear function which now can capture more of the missclassifications that we had seen with just the linear probability model. With the LPM, the fitted values were only a horizontal line which made it difficult to account for the 0's and 1's classification. Here, we can see that most of the points are closer to the fitted values. For months on the book, the fitted value is closer to the 1 classification, meaning that the customer is not likely to be churned. For the credit limit, all of the points for 0, the customers who will be churned, are misclassified because the fitted value do not hover around 0. The customer who will not be churned has a credit limit up to 10,000 is classified because that it where the fitted values lie. For the customer age, the customer who will not be churned (1), are close to the fitted values so they are classified correctly. In contrast, the customer who will be churned does not have any points that are close to the fitted value. For the dependent count, having 0 children is misclassified for both a customer being churned and not churned. Having 4 children is the most classified point for a customer who will not be churned. Lastly, the average open to buy plot shows us that all of the points for 0, the customers who will be churned, are misclassified because the fitted value do not hover around 0. The customer who will not be churned has a average open to buy of up to 10,000 is classified because that it where the fitted values lie.

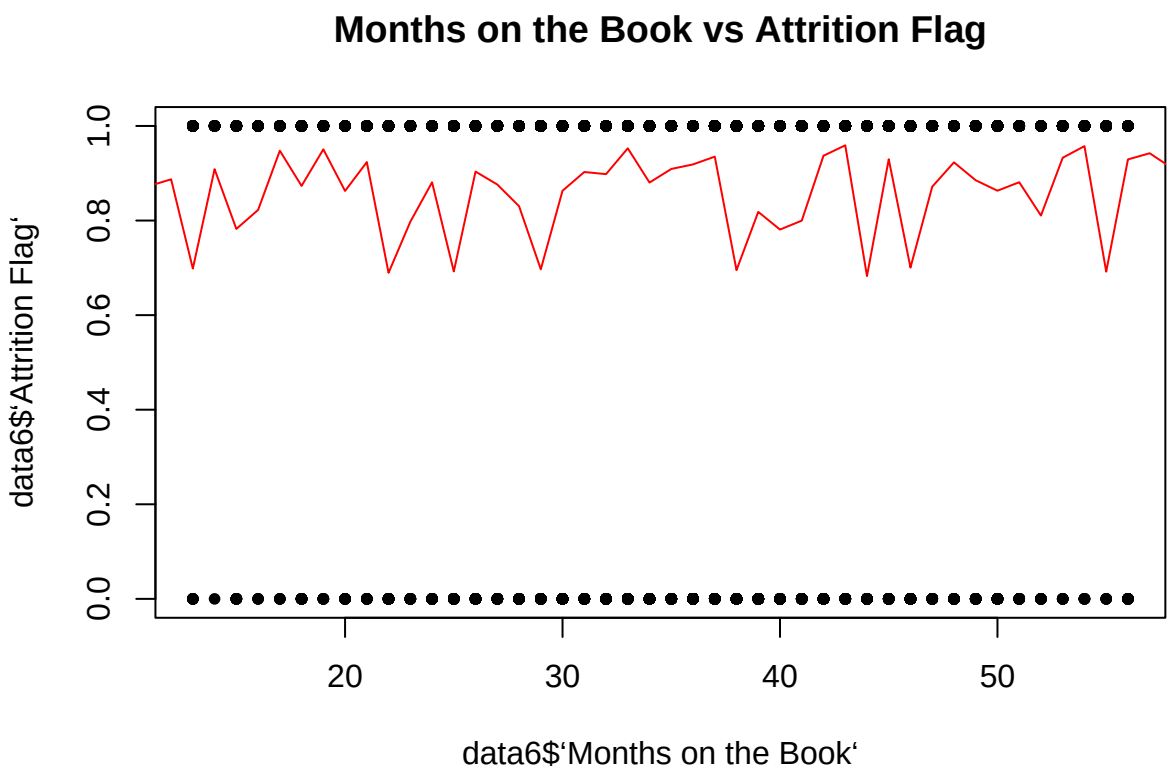
```
#plot months on the book
attach(data6)
library(ggplot2)
library(survMisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survMisc'
```

```
## The following object is masked from 'package:ggplot2':
##
##     autoplot
```

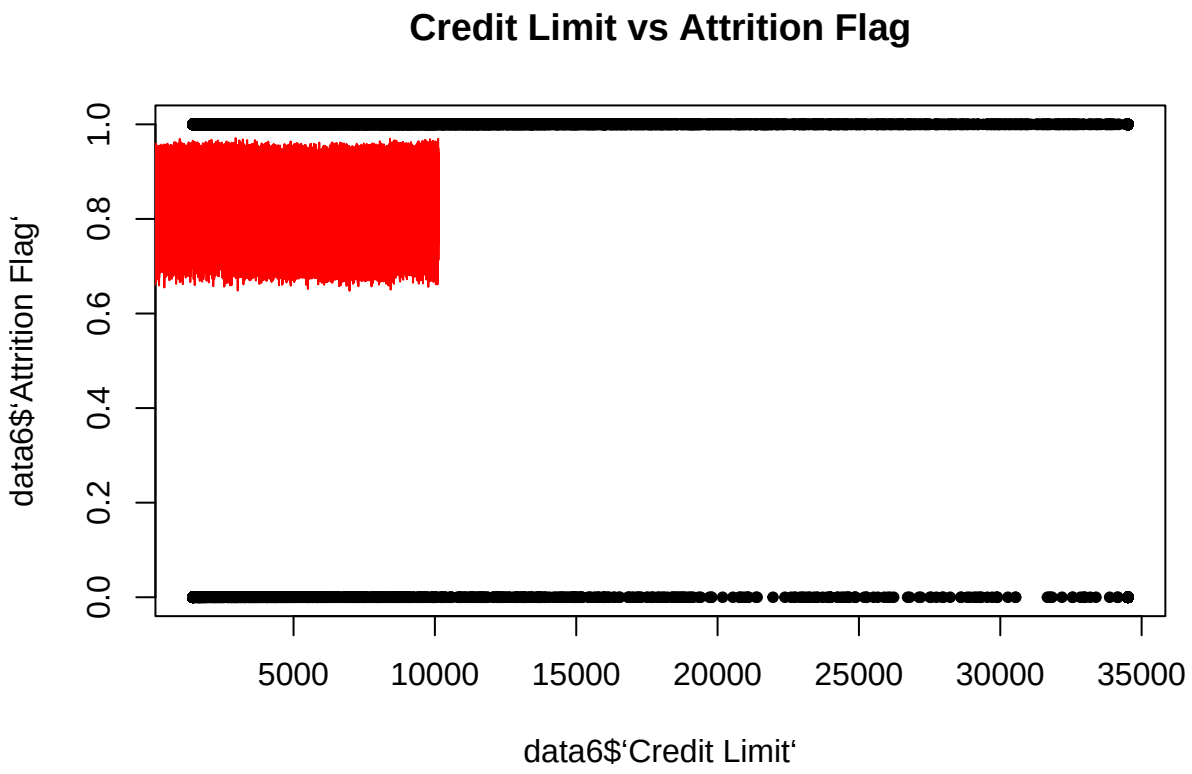
```
x = seq(length.out=10127)
yhat = predict(probit.mod, type = "response", se.fit = TRUE, list(x=data6$`Months on the Book`))
plot(data6$`Months on the Book`, data6$`Attrition Flag`,pch=20, main="Months on the Book vs Attrition Fl
lines(x, yhat$fit, col="red")
```




```
#plot credit limit
attach(data6)
```

```
## The following objects are masked from data6 (pos = 6):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(probit.mod, type = "response", se.fit = TRUE, list(x=data6$`Credit Limit`))
plot(data6$`Credit Limit`, data6$`Attrition Flag`,pch=20, main="Credit Limit vs Attrition Flag")
lines(x, yhat$fit, col="red")
```



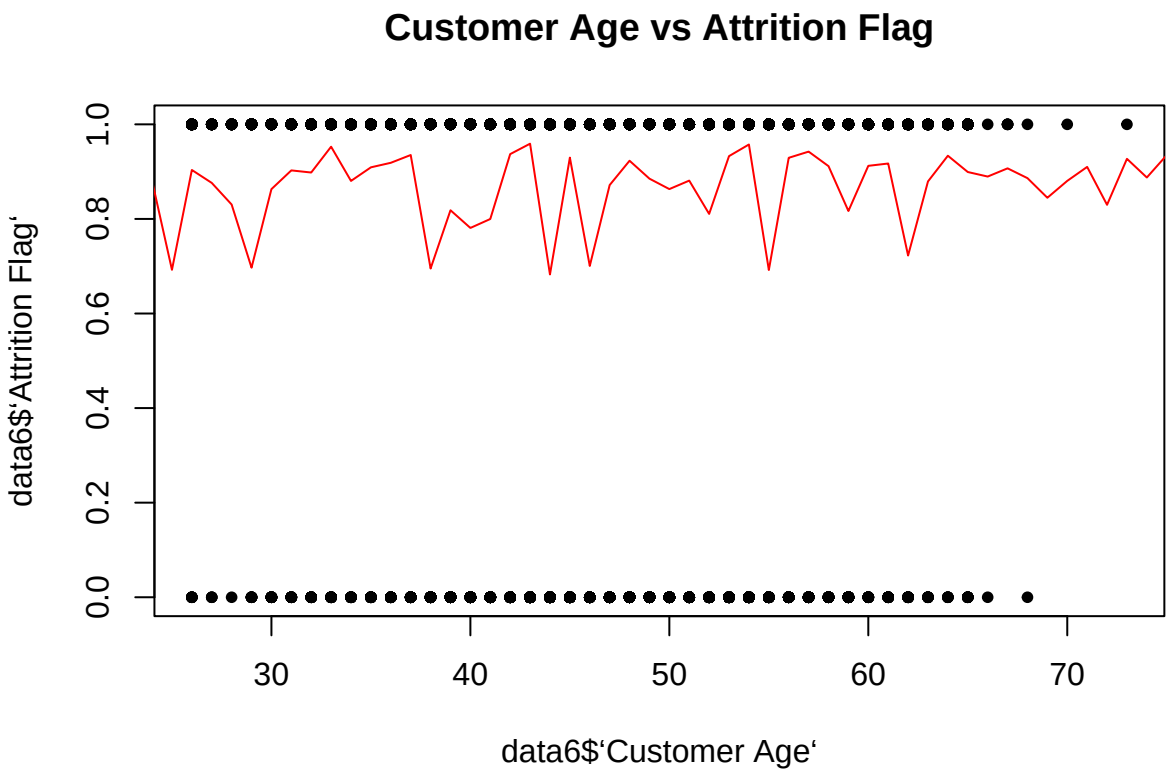

```
length(yhat$fit)
```

```
## [1] 10127
```

```
#plot customer age
attach(data6)
```

```
## The following objects are masked from data6 (pos = 3):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 7):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(probit.mod, type = "response", se.fit = TRUE, list(x=data6$`Customer Age`))
plot(data6$`Customer Age`, data6$`Attrition Flag`,pch=20, main="Customer Age vs Attrition Flag")
lines(x, yhat$fit, col="red")
```



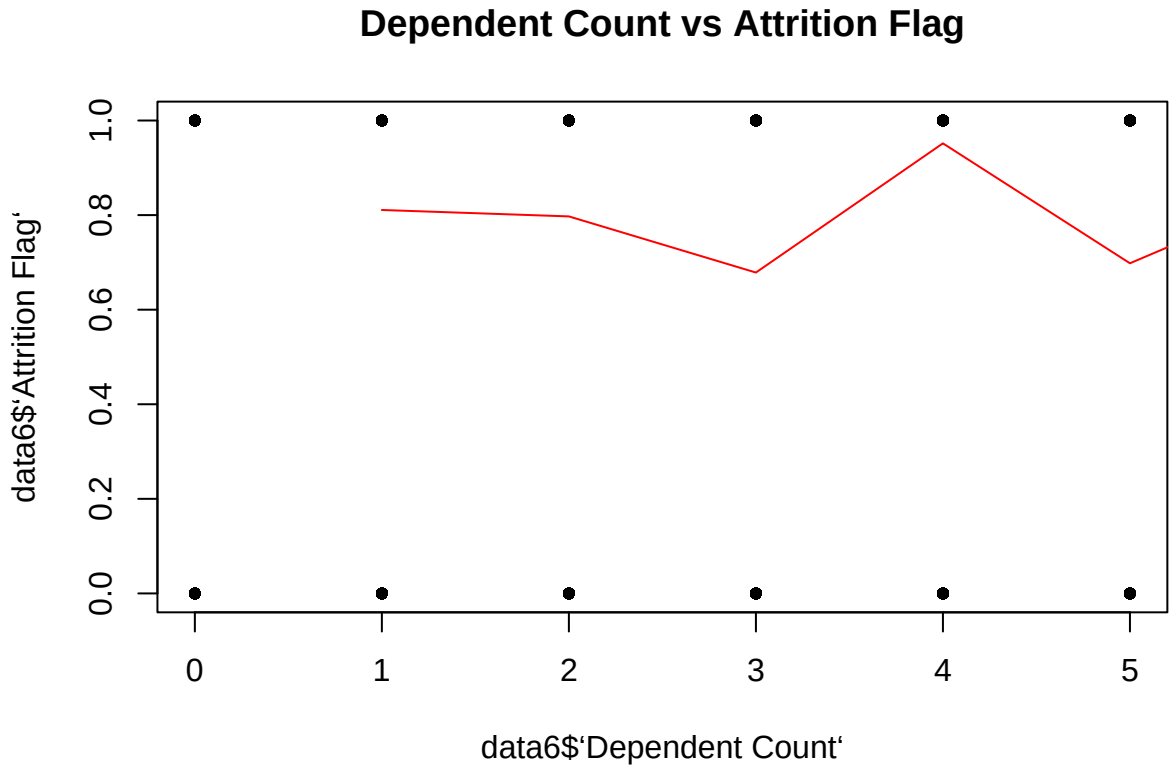

```
length(yhat$fit)
```

```
## [1] 10127
```

```
#plot dependent
attach(data6)
```

```
## The following objects are masked from data6 (pos = 3):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 4):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 8):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(probit.mod, type = "response", se.fit = TRUE,list(x=data6$`Dependent Count`))
plot(data6$`Dependent Count`, data6$`Attrition Flag`,pch=20, main="Dependent Count vs Attrition Flag")
lines(x, yhat$fit, col="red")
```




```
length(yhat$fit)
```

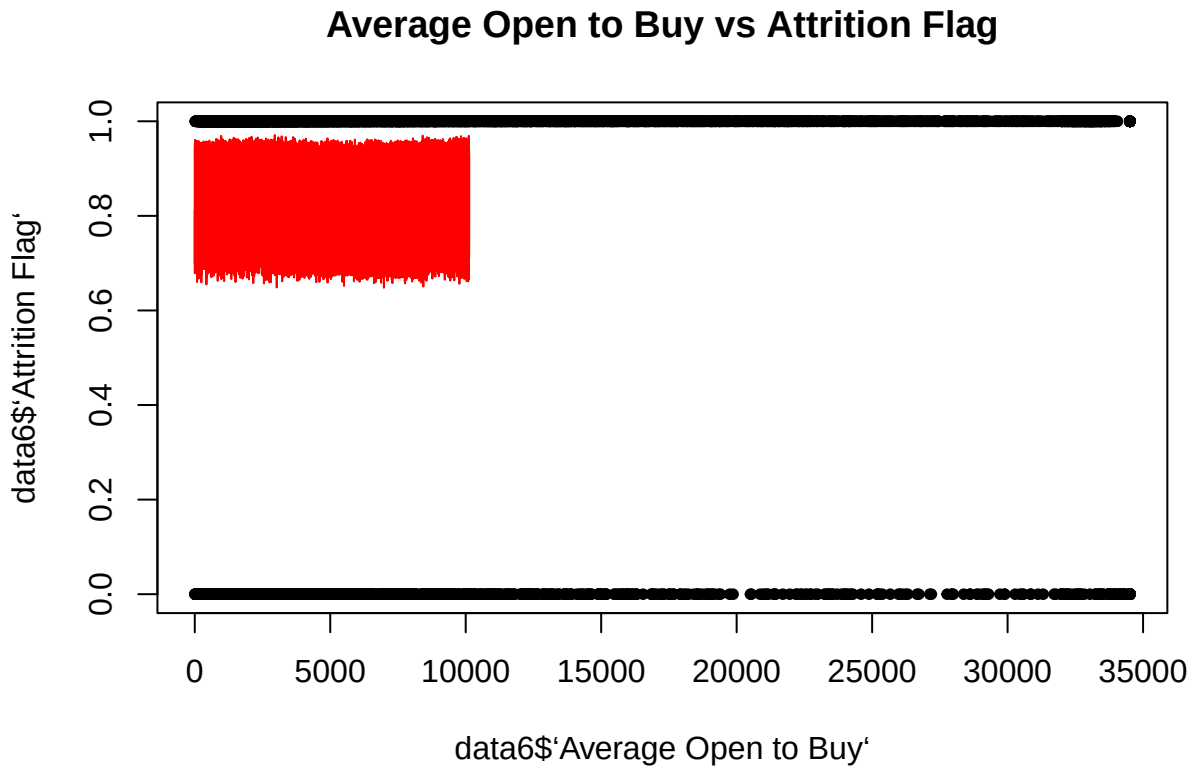
```
## [1] 10127
```

```
#plot average open to buy
attach(data6)
```

```
## The following objects are masked from data6 (pos = 3):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 4):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
## The following objects are masked from data6 (pos = 5):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 9):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(probit.mod, type = "response", se.fit = TRUE, list(x=data6$`Average Open to Buy`))
plot(data6$`Average Open to Buy`, data6$`Attrition Flag`,pch=20, main="Average Open to Buy vs Attrition
lines(x, yhat$fit, col="red")
```



Logit Model

The logit model is like the probit model in the fact that we can not interpret the betas directly. We can only look at the direction. Customer age, dependent count, and average open to buy have a negative effect while months on the book, credit limit have a positive effect. To further interpret the betas, we look at the average marginal effects. This delivers the direct effect. For example, a unit increase in credit limit leads to 0.0001051903 percent increase in the probability that the customer will not be churned.

```r
#logit mod
logit.mod = glm(as.numeric(`Attrition Flag`)~ `Months on the Book`+`Credit Limit`+ `Customer Age`+ `Depe
summary(logit.mod)
```

```
##
## Call:
## glm(formula = as.numeric(‘Attrition Flag‘) ~ ‘Months on the Book‘ +
##     ‘Credit Limit‘ + ‘Customer Age‘ + ‘Dependent Count‘ + ‘Average Open to Buy‘,
##     family = binomial(link = "logit"), data = data6)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6351   0.3433   0.4616   0.5988   0.9587
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.274e+00  1.881e-01   6.772 1.27e-11 ***
## ‘Months on the Book‘    1.370e-03  5.760e-03   0.238   0.8120
## ‘Credit Limit‘          9.319e-04  3.671e-05  25.387  < 2e-16 ***
## ‘Customer Age‘         -1.031e-02  5.793e-03  -1.779   0.0752 .
## ‘Dependent Count‘      -5.046e-02  2.197e-02  -2.297   0.0216 *
## ‘Average Open to Buy‘ -9.273e-04  3.668e-05 -25.281  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8927.2  on 10126  degrees of freedom
## Residual deviance: 8199.7  on 10121  degrees of freedom
## AIC: 8211.7
##
## Number of Fisher Scoring iterations: 5
```

```r
#Marginal effect
confint(logit.mod)
```

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

```
##                               2.5 %        97.5 %
## (Intercept)             0.9074989141  1.6450358828
## ‘Months on the Book‘   -0.0099376211  0.0126450190
## ‘Credit Limit‘          0.0008603910  0.0010043059
## ‘Customer Age‘         -0.0216742188  0.0010371062
## ‘Dependent Count‘      -0.0935547711 -0.0074245964
## ‘Average Open to Buy‘ -0.0009996381 -0.0008558336
```

```r
sum_phi <- mean(dnorm(predict(logit.mod,, type = "link")))
ame = sum_phi*coef(logit.mod)
ame
```

```
##           (Intercept)  ‘Months on the Book‘         ‘Credit Limit‘
##          0.1437824954          0.0001546696          0.0001051903
##        ‘Customer Age‘     ‘Dependent Count‘ ‘Average Open to Buy‘
##         -0.0011635532         -0.0056949451         -0.0001046699
```

Confusion Matrix

The confusion matrix gives us an accuracy level of 79%

```
#Confusion matrix
ols.pred.classes <- ifelse(fitted(logit.mod) > .7, 1, 0)
table(ols.pred.classes, data6$`Attrition Flag`)
```

```
##
## ols.pred.classes    0    1
##                0  756 1259
##                1  871 7241
```

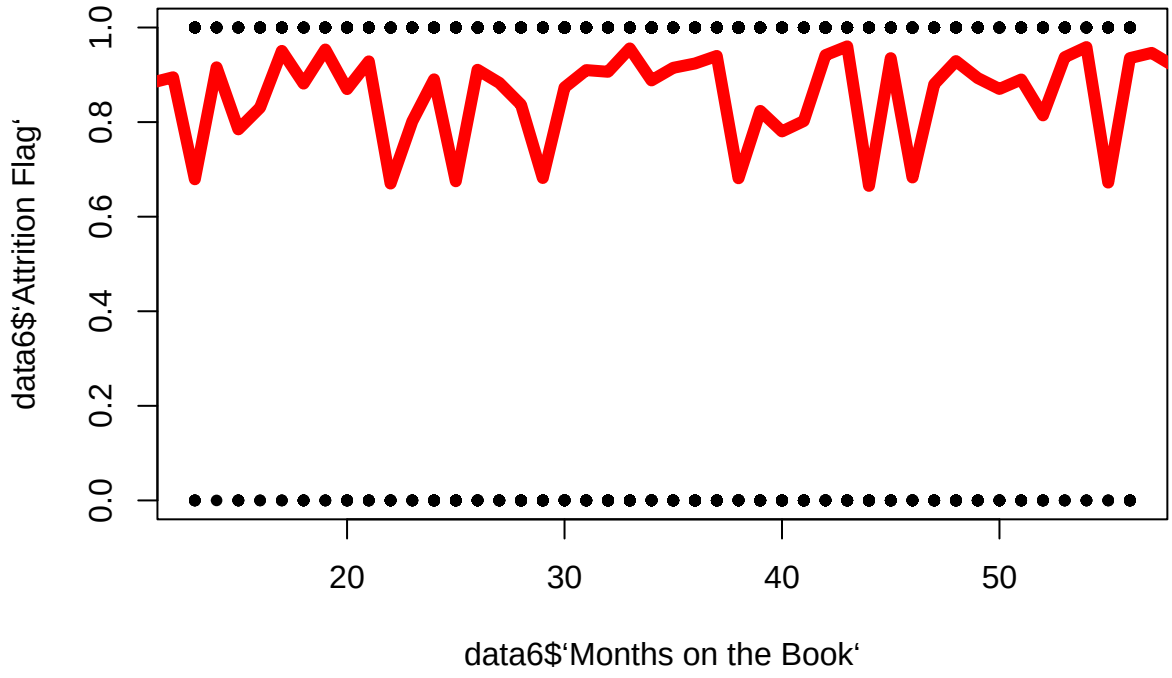
Logit Plots

For the months on the books, the fitted values leans towards the 1 classification that customers are less likely to be churned. For the credit limit, all of the points for 0, the customers who will be churned, are misclassified because the fitted value do not hover around 0. The customer who will not be churned has a credit limit up to 10,000 is classified because that it where the fitted values lie. For dependent count, it is also leaning towards the classification of 1. Specifically, 4 children is best classified. For customer age, the fitted values continue to lean towards 1. It is a good fit overall. Lastly, the average open to buy plot shows us that all of the points for 0, the customers who will be churned, are misclassified because the fitted value do not hover around 0. The customer who will not be churned has a average open to buy of up to 10,000 is classified because that it where the fitted values lie.

```
#Logit plot
attach(data6)
```

```
## The following objects are masked from data6 (pos = 3):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 4):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 5):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 6):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 10):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

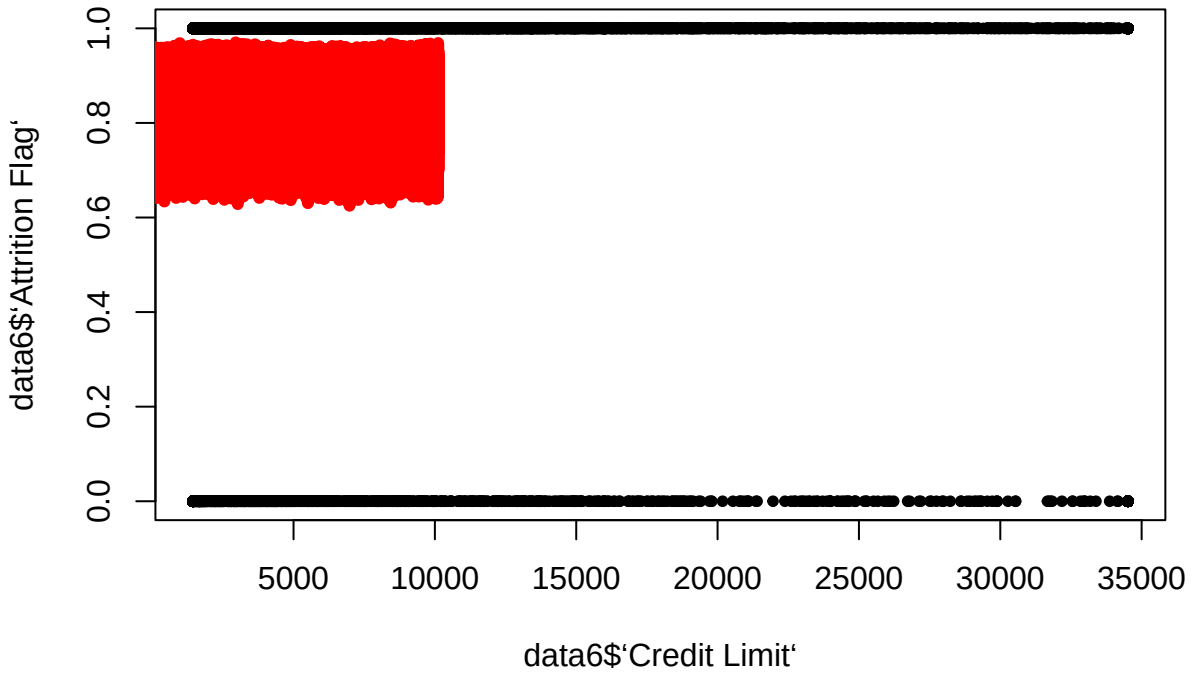
```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(logit.mod, type = "response", se.fit = TRUE, list(x=data6$`Months on the Book`))
plot(data6$`Months on the Book`, data6$`Attrition Flag`,pch=20)
lines(x, yhat$fit,lwd=6, col ="red")
```




```
length(yhat$fit)
```

```
## [1] 10127
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(logit.mod, type = "response", se.fit = TRUE, list(x=data6$`Credit Limit`))
plot(data6$`Credit Limit`, data6$`Attrition Flag`,pch=20)
lines(x, yhat$fit,lwd=6, col ="red")
```




```
length(yhat$fit)
```

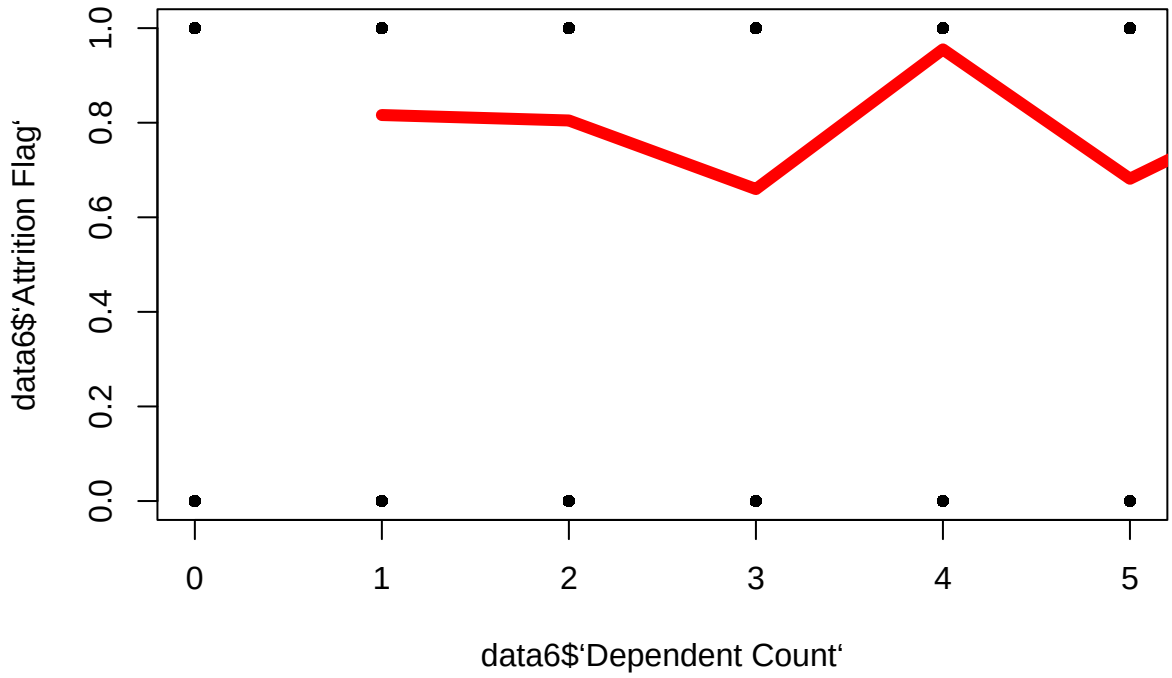
```
## [1] 10127
```

```
attach(data6)
```

```
## The following objects are masked from data6 (pos = 3):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 4):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 5):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 6):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
## The following objects are masked from data6 (pos = 7):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 11):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(logit.mod, type = "response", se.fit = TRUE, list(x=data6$`Dependent Count`))
plot(data6$`Dependent Count`, data6$`Attrition Flag`,pch=20)
lines(x, yhat$fit,lwd=6, col ="red")
```

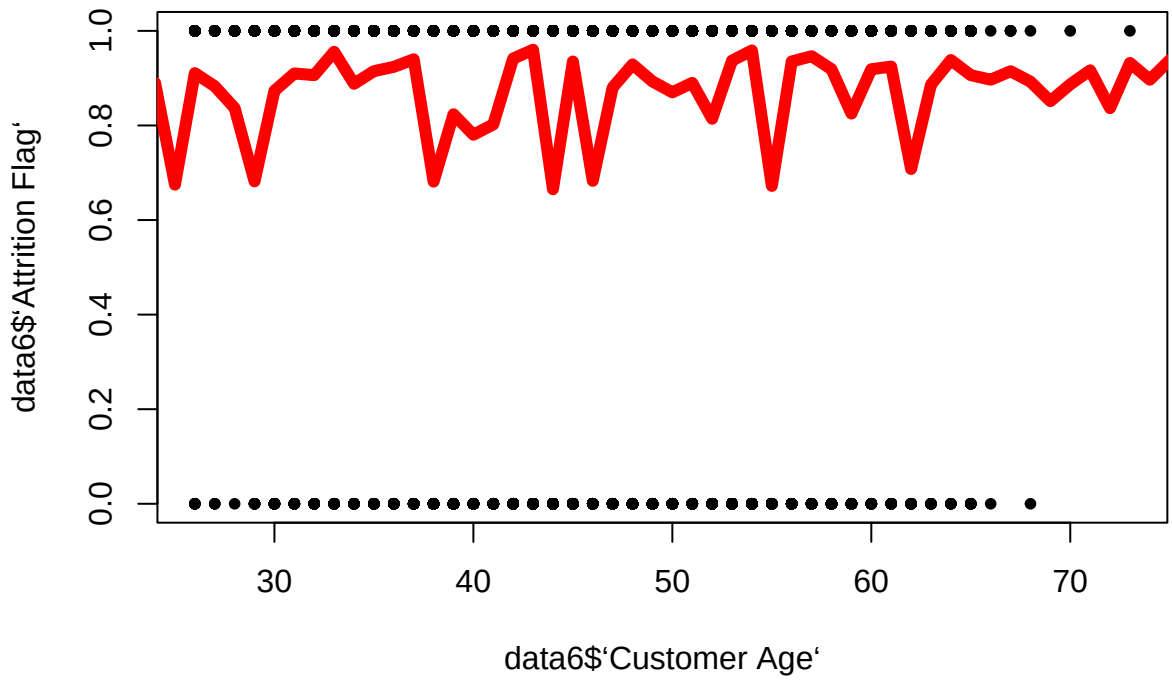



```
length(yhat$fit)
```

```
## [1] 10127
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
```

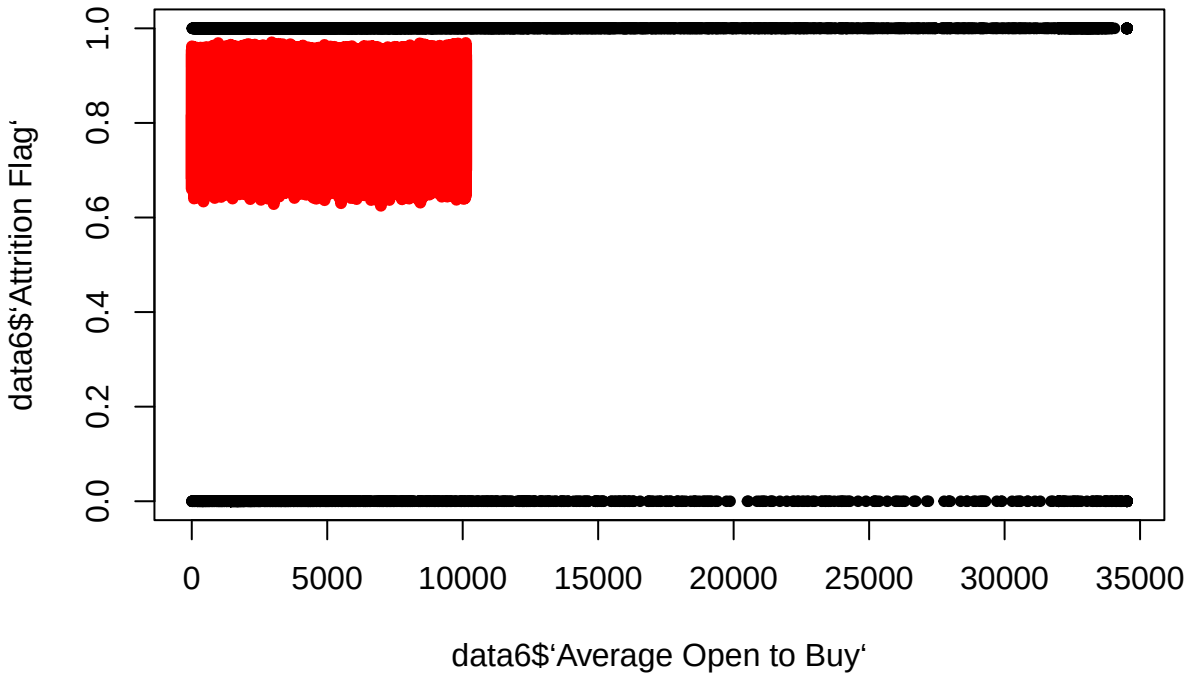
```
yhat = predict(logit.mod, type = "response", se.fit = TRUE, list(x=data6$`Customer Age`))
plot(data6$`Customer Age`, data6$`Attrition Flag`,pch=20)
lines(x, yhat$fit,lwd=6, col ="red")
```




```
length(yhat$fit)
```

```
## [1] 10127
```

```
library(ggplot2)
library(survMisc)
x = seq(length.out=10127)
yhat = predict(logit.mod, type = "response", se.fit = TRUE, list(x=data6$`Average Open to Buy`))
plot(data6$`Average Open to Buy`, data6$`Attrition Flag`,pch=20)
lines(x, yhat$fit,lwd=6, col ="red")
```




```
length(yhat$fit)
```

```
## [1] 10127
```

Logit Cross Validation

We use the .66 training/testing and the cross validation to assess how well the logit model works. We can use our balanced accuracy of .5 as a reference point to determine the performance of the model. The accuracy measures at .8394 which is a good measurement given it beats the threshold of .5. Also, the specificity measures at 1 which is greater than .5 too. However, the sensitivity is measured at 0 is not good. This means that assigning 1 is a good classifier while assigning 0 was not. We can conclude that based on the predictors we have used, they mostly influence the classifier of 1. We had noticed that this could happen from the beginning just by analyzing our variables. For example, when most of our the data showed that the customer age average was around 45, we made a hypothesis that they would be more financially experienced and so it would be less likely that a customer would be churned, making sense why the best classifier is 1. To conclude, we will choose the logit model because our accuracies between the models are very similar. The plots from our probit and logit model also were the same, so we are safe to choose the better model of logit. Logit is better to analyze analytically.

```
#Cross Validation
library(caret)
```

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

```
##
## Attaching package: 'caret'

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

```
dataFORYOU<-data.frame(data6$`Attrition Flag`, data6$`Months on the Book`, data6$`Credit Limit`, data6$
View(dataFORYOU)
colnames(dataFORYOU)<-c("Attrition Flag", "Months on the Book", "Credit Limit", "Customer Age", "Depend

inTraining <- createDataPartition(dataFORYOU$`Attrition Flag`, p = .66, list = FALSE)
training <- dataFORYOU[ inTraining,]
testing  <- dataFORYOU[-inTraining,]
train_control <- trainControl(method = "cv",
                              number = 5)


logit_model <- train(as.factor(`Attrition Flag`)~., data = training,
                         method = "glm",
                         family = "binomial",
                         trControl = train_control)


# Predict (probabilities) using the testing data
pred_att = predict(logit_model, newdata = testing)

# Evaluate performance
confusionMatrix(data=pred_att, reference=as.factor(testing$`Attrition Flag`))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0    0
##          1  553 2890
##
##                Accuracy : 0.8394
##                  95% CI : (0.8267, 0.8515)
##     No Information Rate : 0.8394
##     P-Value [Acc > NIR] : 0.5114
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.8394
##              Prevalence : 0.1606
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
```

```
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

Logit Predictions

Our preferred model is logit. When deciding what values to use to predict, for the first prediction, we chose the average for each variable. We included all predictors when conducting a prediction. For the first four periods, our predicted probabilities that a customer will not be churned are 0.8162438, 0.8047168, 0.6596746, and 0.9550266. The standard errors for these four periods are also 0.006225332, 0.011695526, 0.011827132, 0.003653452. Given these standard errors are small, we can determine that these predictions are reliable.

For our second prediction we used the median values from the statistical summary, and obtained the predictions of 0.8162438, 0.8047168, 0.6596746, and 0.9550266 for the first four periods. The standard errors are 0.006225332, 0.011695526, 0.011827132, and 0.003653452.

For our third prediction, we used the minimum values from the statistical summary and obtained the predictions of 0.8162438, 0.8047168, 0.6596746, and 0.9550266 for the first four periods. The standard errors are 0.006225332, 0.011695526, 0.011827132, and 0.003653452.

For our fourth prediction, we used the maximum values from the statistical summary and obtained the predictions of 0.8162438, 0.8047168, 0.6596746, and 0.9550266 for the first four periods. The standard errors are 0.006225332, 0.011695526, 0.011827132, and 0.003653452.

```
attach(data6)
```

```
## The following objects are masked from data6 (pos = 5):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 6):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 7):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 8):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 9):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
## The following objects are masked from data6 (pos = 10):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book

## The following objects are masked from data6 (pos = 14):
##
##     Attrition Flag, Average Open to Buy, Credit Limit, Customer Age,
##     Dependent Count, Months on the Book
```

```
x.mean<-data.frame(`Credit Limit`= 8632)+ data.frame(`Average Open to Buy`= 7469)+ data.frame(`Dependen
predictdata1<-predict(logit.mod, x.mean, type="response", se.fit=TRUE)
```

```
## Warning: 'newdata' had 1 row but variables found have 10127 rows
```

```
head(predictdata1$fit, n=4)
```

```
##         1         2         3         4
## 0.8162438 0.8047168 0.6596746 0.9550266
```

```
head(predictdata1$se, n=4)
```

```
##           1           2           3           4
## 0.006225332 0.011695526 0.011827132 0.003653452
```

```
x.median<-data.frame(`Credit Limit`= 4549)+ data.frame(`Average Open to Buy`= 474)+ data.frame(`Depende
predictdata2<-predict(logit.mod, x.median, type="response", se.fit=TRUE)
```

```
## Warning: 'newdata' had 1 row but variables found have 10127 rows
```

```
head(predictdata2$fit, n=4)
```

```
##         1         2         3         4
## 0.8162438 0.8047168 0.6596746 0.9550266
```

```
head(predictdata2$se, n=4)
```

```
##           1           2           3           4
## 0.006225332 0.011695526 0.011827132 0.003653452
```

```
x.min<-data.frame(`Credit Limit`= 1438)+ data.frame(`Average Open to Buy`= 3)+ data.frame(`Dependent Co
predictdata3<-predict(logit.mod, x.min, type="response", se.fit=TRUE)
```

```
## Warning: 'newdata' had 1 row but variables found have 10127 rows
```

```
head(predictdata3$fit, n=4)
```

```
##         1         2         3         4
## 0.8162438 0.8047168 0.6596746 0.9550266
```

```
head(predictdata3$se, n=4)
```

```
##           1           2           3           4
## 0.006225332 0.011695526 0.011827132 0.003653452
```

```
x.max<-data.frame(`Credit Limit`= 34516)+ data.frame(`Average Open to Buy`= 34516)+ data.frame(`Depende
predictdata4<-predict(logit.mod, x.max, type="response", se.fit=TRUE)
```

```
## Warning: 'newdata' had 1 row but variables found have 10127 rows
```

```
head(predictdata4$fit, n=4)
```

```
##         1         2         3         4
## 0.8162438 0.8047168 0.6596746 0.9550266
```

```
head(predictdata4$se, n=4)
```

```
##           1           2           3           4
## 0.006225332 0.011695526 0.011827132 0.003653452
```