Customer Intelligence and Big Data

Alan Rijnders and Lorenzo Severi

11/4/2021

We start reading in the data to perform the analysis

```
#read in data
data <- read.csv("ch.csv")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
#install.packages('caret', dependencies = TRUE)
#install with dependencies = TRUE is important for the calculation of sensitivity and specficity
library(caret)
```

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

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(repr)
library(caTools)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
```

```
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(rpart)
library(rpart.plot)
library(ggpubr)
```

We now start exploring the dataset

```
#summary of the dataset
summary(data)
```

```
##        X            CLIENTNUM         Attrition_Flag      Customer_Age
##  Min.   :    1   Min.   :708082083   Length:10127       Min.   :26.00
##  1st Qu.: 2532   1st Qu.:713036770   Class :character   1st Qu.:41.00
##  Median : 5064   Median :717926358   Mode  :character   Median :46.00
##  Mean   : 5064   Mean   :739177606                      Mean   :46.33
##  3rd Qu.: 7596   3rd Qu.:773143533                      3rd Qu.:52.00
##  Max.   :10127   Max.   :828343083                      Max.   :73.00
##     Gender           Dependent_count Education_Level    Marital_Status
##  Length:10127       Min.   :0.000   Length:10127       Length:10127
##  Class :character   1st Qu.:1.000   Class :character   Class :character
##  Mode  :character   Median :2.000   Mode  :character   Mode  :character
##                     Mean   :2.346
##                     3rd Qu.:3.000
##                     Max.   :5.000
##  Income_Category    Card_Category      Months_on_book  Total_Relationship_Count
##  Length:10127       Length:10127       Min.   :13.00   Min.   :1.000
##  Class :character   Class :character   1st Qu.:31.00   1st Qu.:3.000
##  Mode  :character   Mode  :character   Median :36.00   Median :4.000
##                                        Mean   :35.93   Mean   :3.813
##                                        3rd Qu.:40.00   3rd Qu.:5.000
##                                        Max.   :56.00   Max.   :6.000
##  Months_Inactive_12_mon Contacts_Count_12_mon  Credit_Limit    Total_Trans_Amt
##  Min.   :0.000          Min.   :0.000         Min.   : 1438   Min.   :  510
##  1st Qu.:2.000          1st Qu.:2.000         1st Qu.: 2555   1st Qu.: 2156
##  Median :2.000          Median :2.000         Median : 4549   Median : 3899
##  Mean   :2.341          Mean   :2.455         Mean   : 8632   Mean   : 4404
##  3rd Qu.:3.000          3rd Qu.:3.000         3rd Qu.:11068   3rd Qu.: 4741
##  Max.   :6.000          Max.   :6.000         Max.   :34516   Max.   :18484
##  Total_Trans_Ct   Avg_Utilization_Ratio
##  Min.   : 10.00   Min.   :0.0000
##  1st Qu.: 45.00   1st Qu.:0.0230
##  Median : 67.00   Median :0.1760
##  Mean   : 64.86   Mean   :0.2749
##  3rd Qu.: 81.00   3rd Qu.:0.5030
##  Max.   :139.00   Max.   :0.9990
```

```
#we drop the X and client number column
data <- subset(data, select=-c(X,CLIENTNUM))
```

We are interested to see how many customers have remained at the company and how many have left.

```
table(data$Attrition_Flag)
```

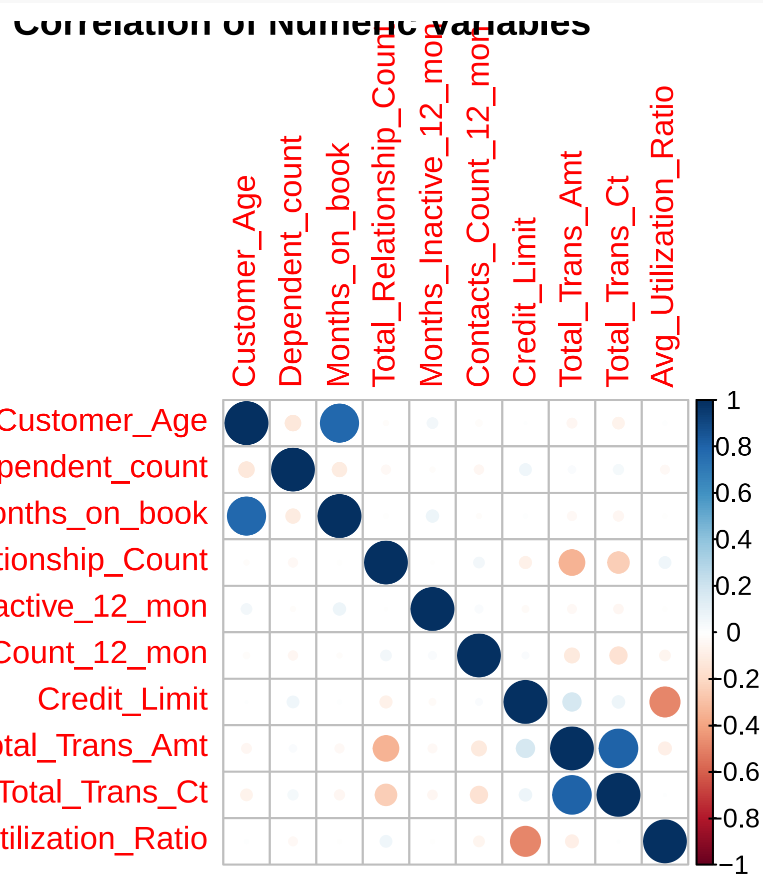
```
##
## Attrited Customer Existing Customer
##              1627              8500
```

In other words, out of total 10127 customers in the database we have 8500 customers that have remained at the company ,whereas 1627 customers have left.

Similarly we transform the variables Gender, Education Level, Marital Status, Income Category and Card Category to factor variables.

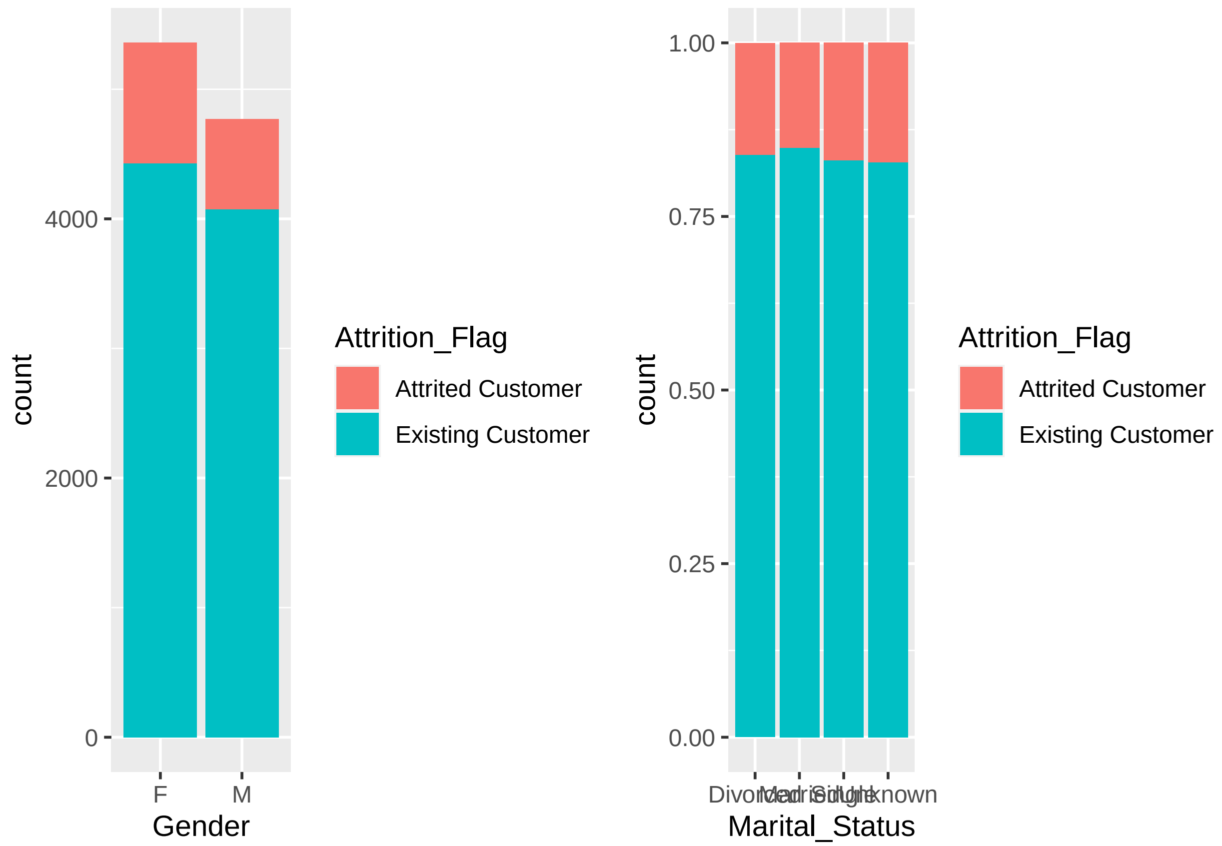
```
data <- transform(
  data,
  Attrition_Flag = as.factor(Attrition_Flag),
  Gender = as.factor(Gender),
  Education_Level = as.factor(Education_Level),
  Marital_Status = as.factor(Marital_Status),
  Income_Category = as.factor(Income_Category),
  Card_Category = as.factor(Card_Category))
```

```
nv <- sapply(data, is.numeric)
cormat <- cor(data[,nv])
corrplot::corrplot(cormat, title = "Correlation of Numeric Variables")
```

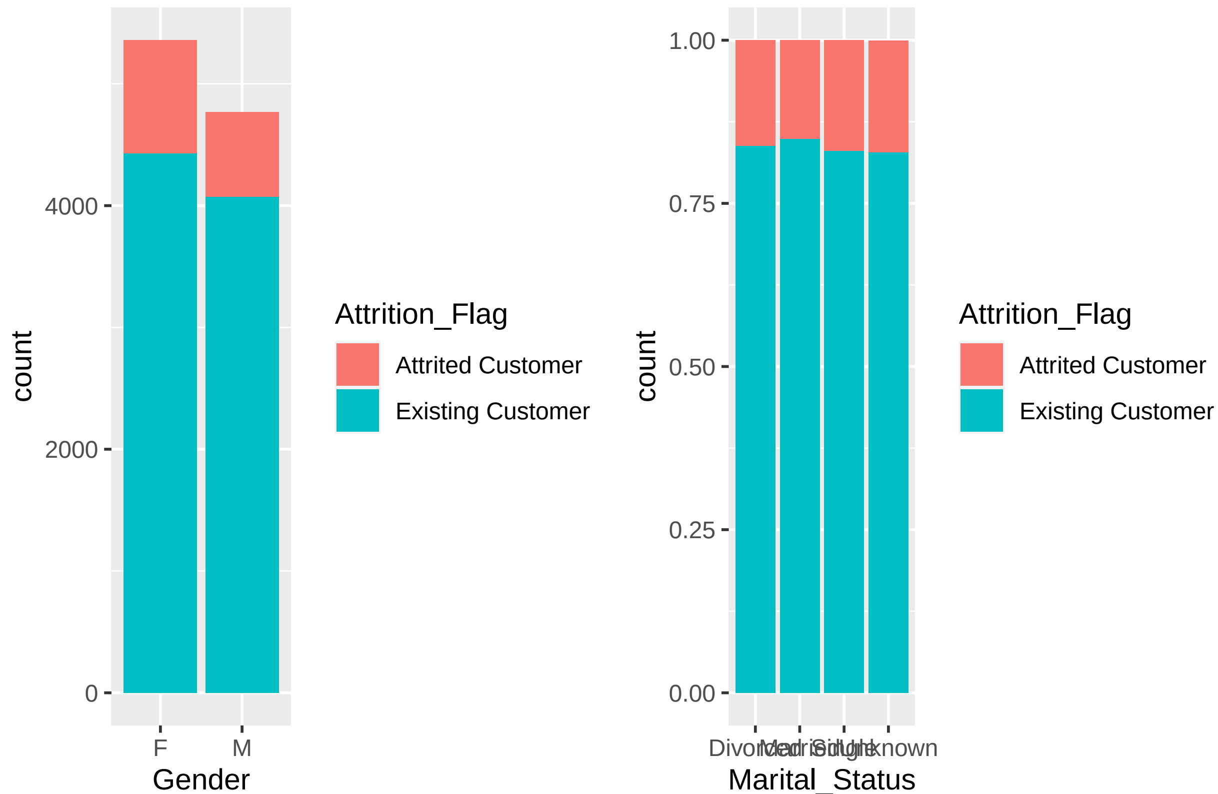



```
fig1 <-   ggarrange(ggplot(data, aes(x=Gender,fill=Attrition_Flag))+ geom_bar() ,
          ggplot(data, aes(x=Marital_Status,fill=Attrition_Flag))+ geom_bar(position = 'fill'))
```

```r
print(fig1)
```

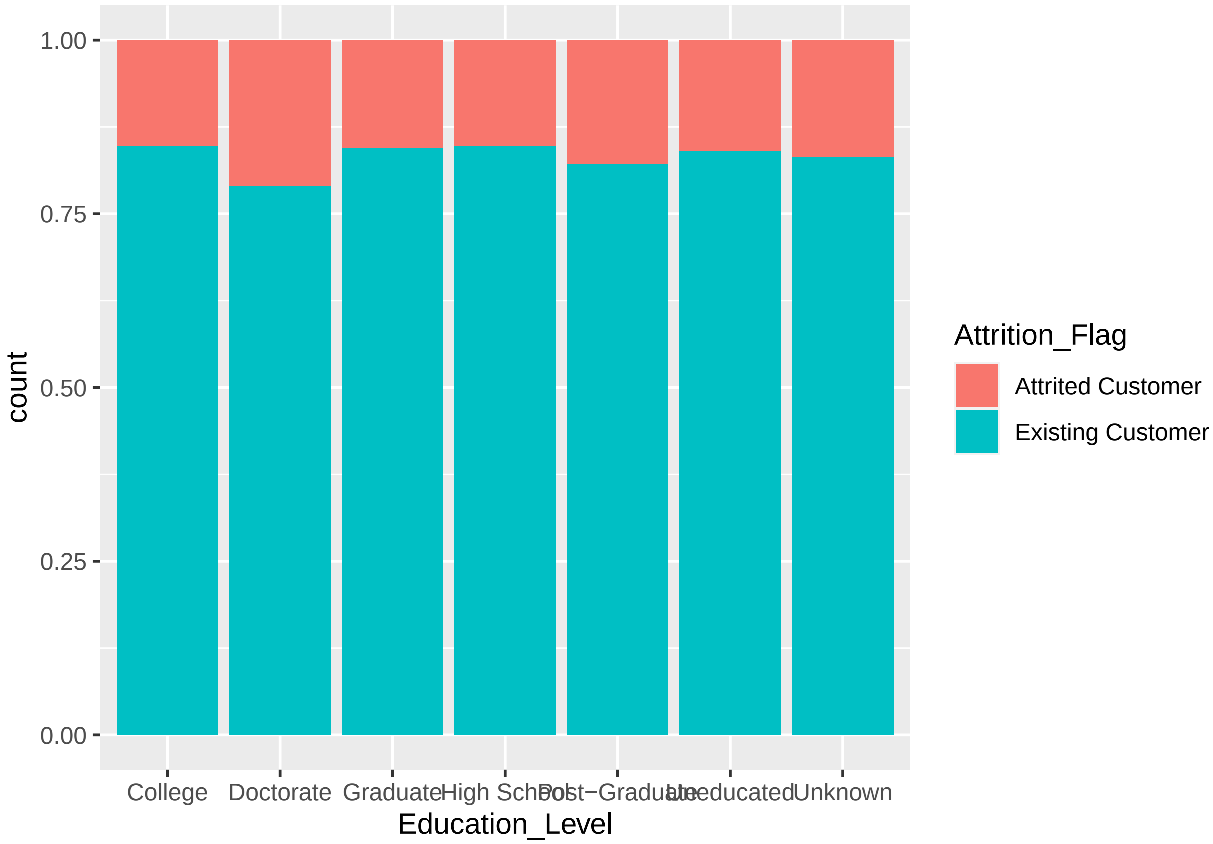



```r
annotate_figure(fig1, bottom  = text_grob("Attrition Percentage in Gender, Marital Status and Card Categ
```

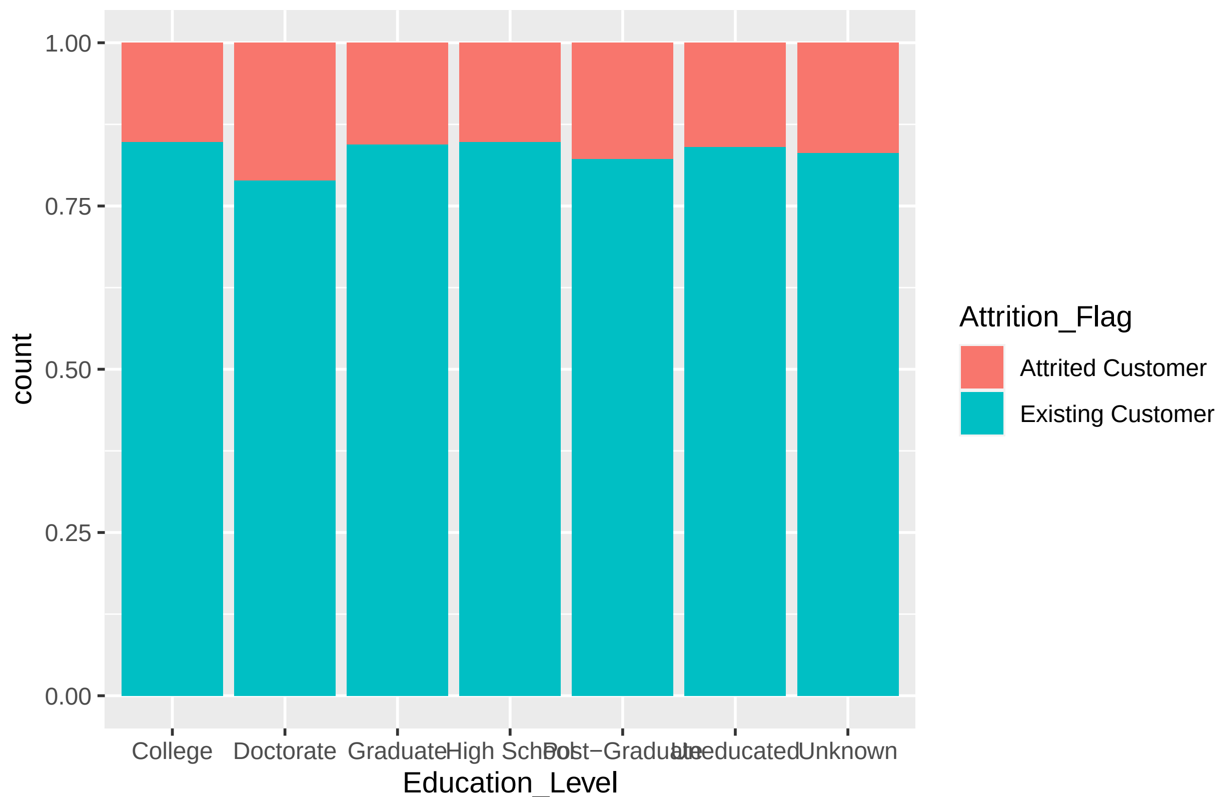


Attrition Percentage in Gender, Marital Status and Card Category

```
fig2 <- ggplot(data, aes(x=Education_Level,fill=Attrition_Flag))+ geom_bar(position = 'fill')
print(fig2)
```

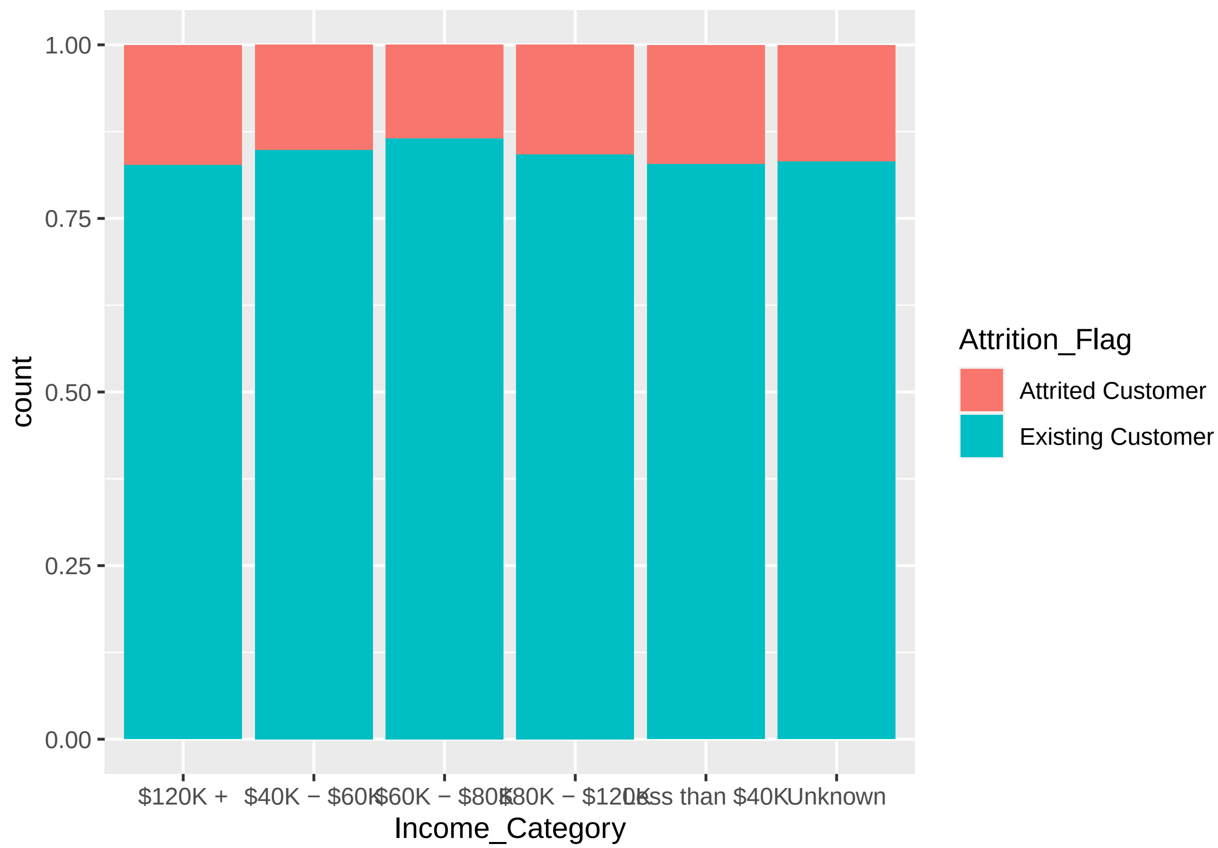



```
annotate_figure(fig2, bottom  = text_grob("Attrition Percentage for different levels of education", col
```

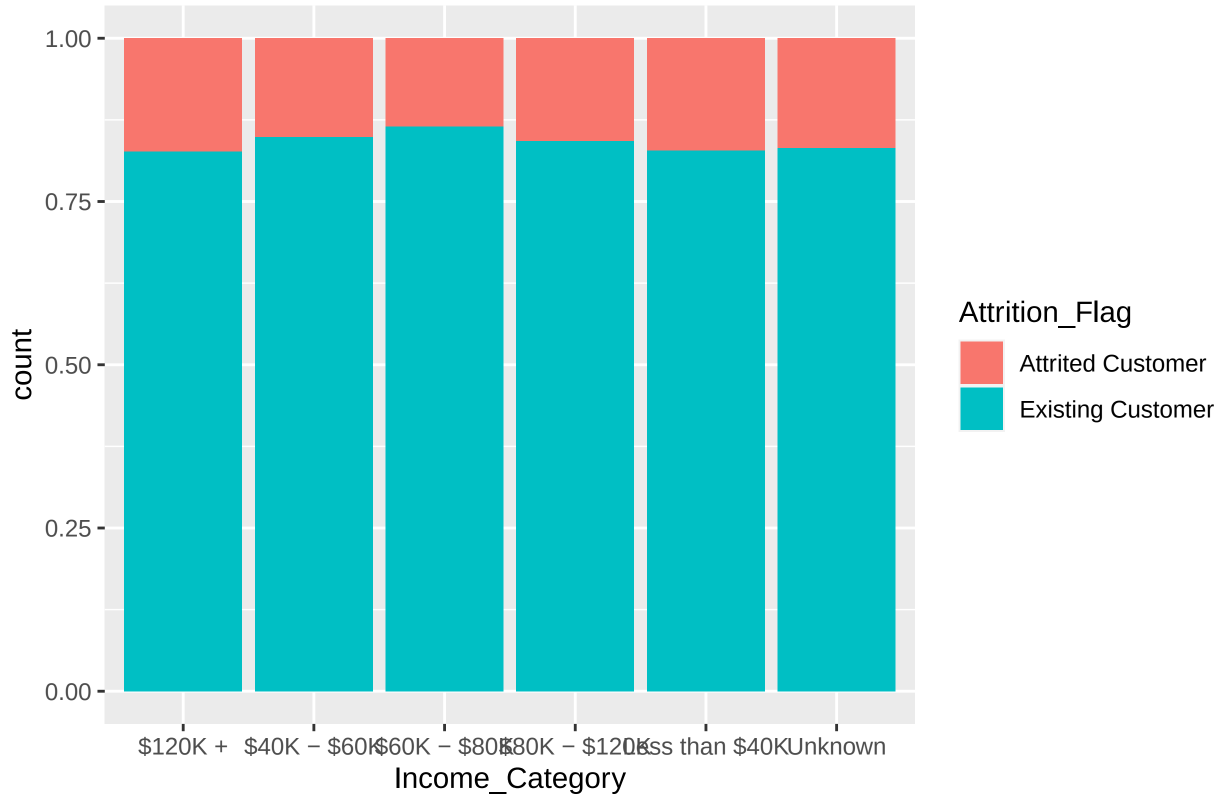


Attrition Percentage for different levels of education

```
fig3 <- ggplot(data, aes(x=Income_Category,fill=Attrition_Flag))+ geom_bar(position = 'fill')
print(fig3)
```

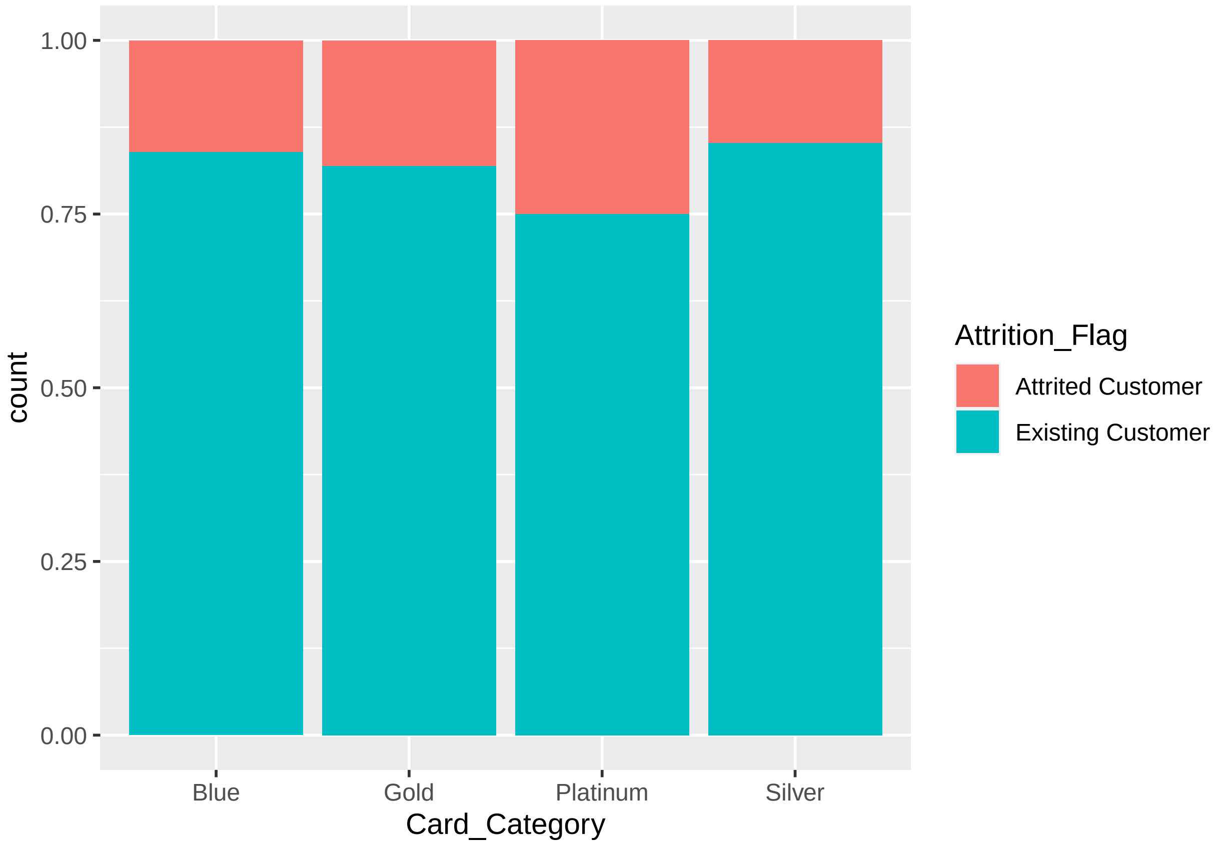



```
annotate_figure(fig3, bottom  = text_grob("Attrition Percentage for different income levels", col = "blu
```

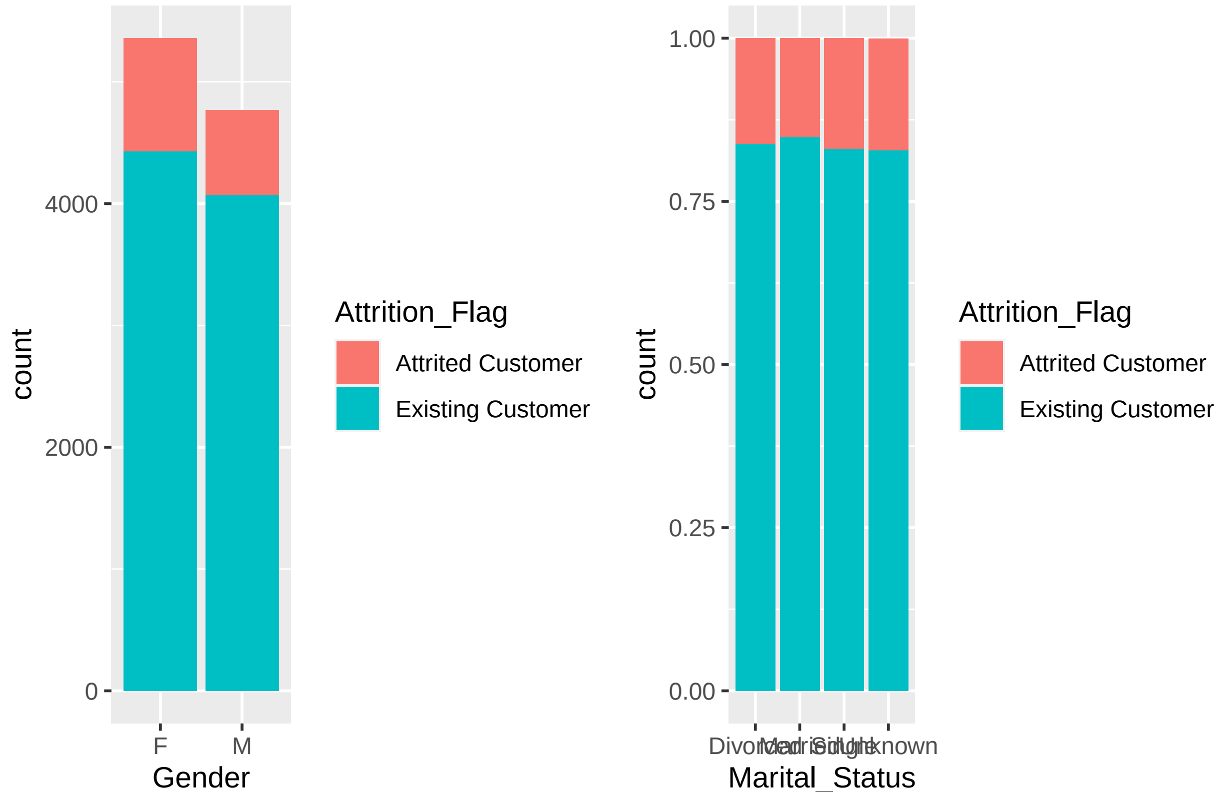


Attrition Percentage for different income levels

```
fig4 <- ggplot(data, aes(x=Card_Category,fill=Attrition_Flag))+ geom_bar(position = 'fill')
print(fig4)
```




```
annotate_figure(fig1, bottom  = text_grob("Attrition Percentage for different cardholder categories", co
```

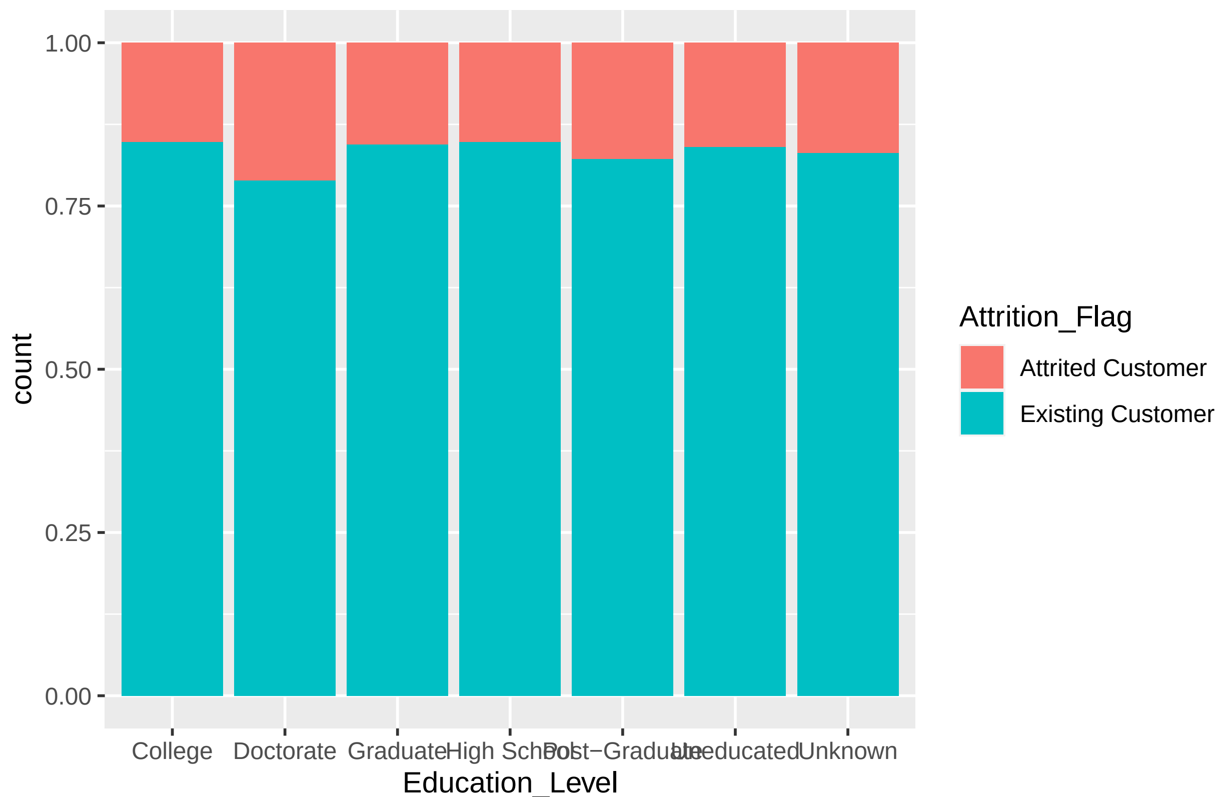


Attrition Percentage for different cardholder categories

```
fig5 <-   ggarrange(
          ggplot(data, aes(y= Dependent_count, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Months_on_book, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig5)
```

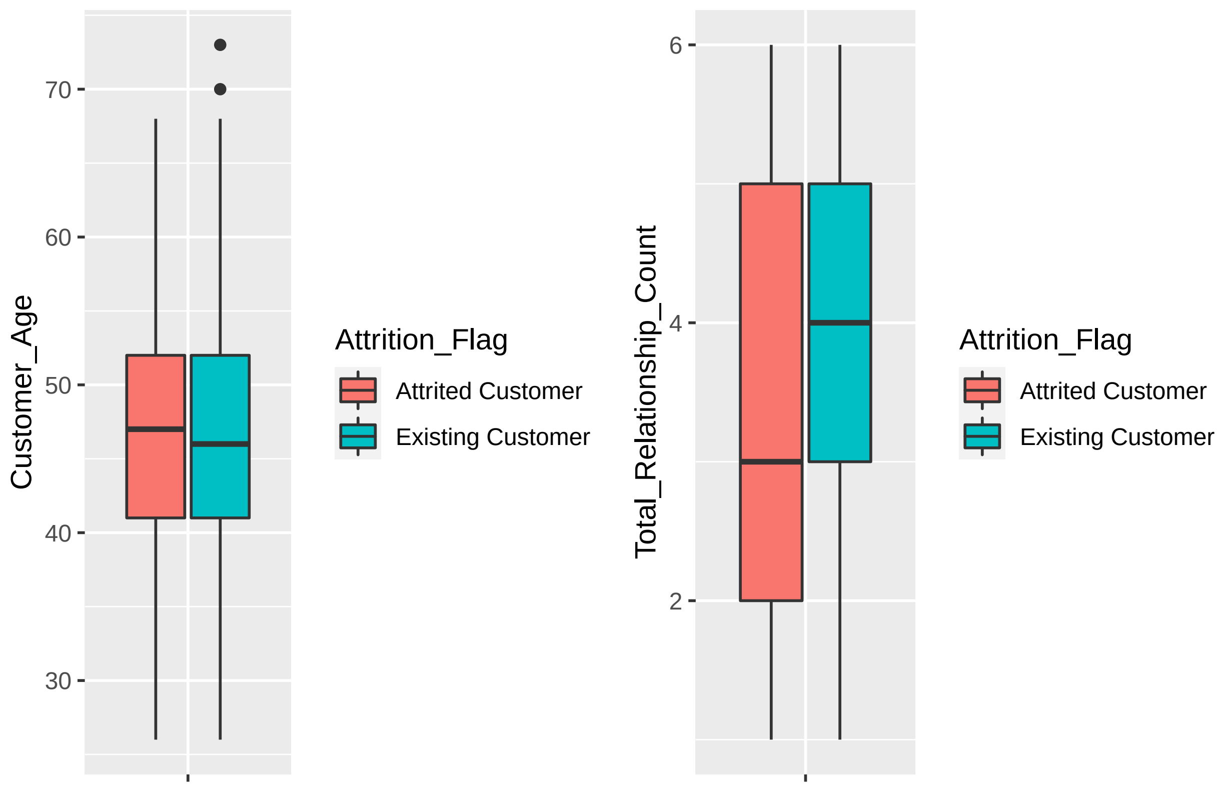

```
annotate_figure(fig2, bottom  = text_grob("Attrition Percentage for different number of dependents in fa
```



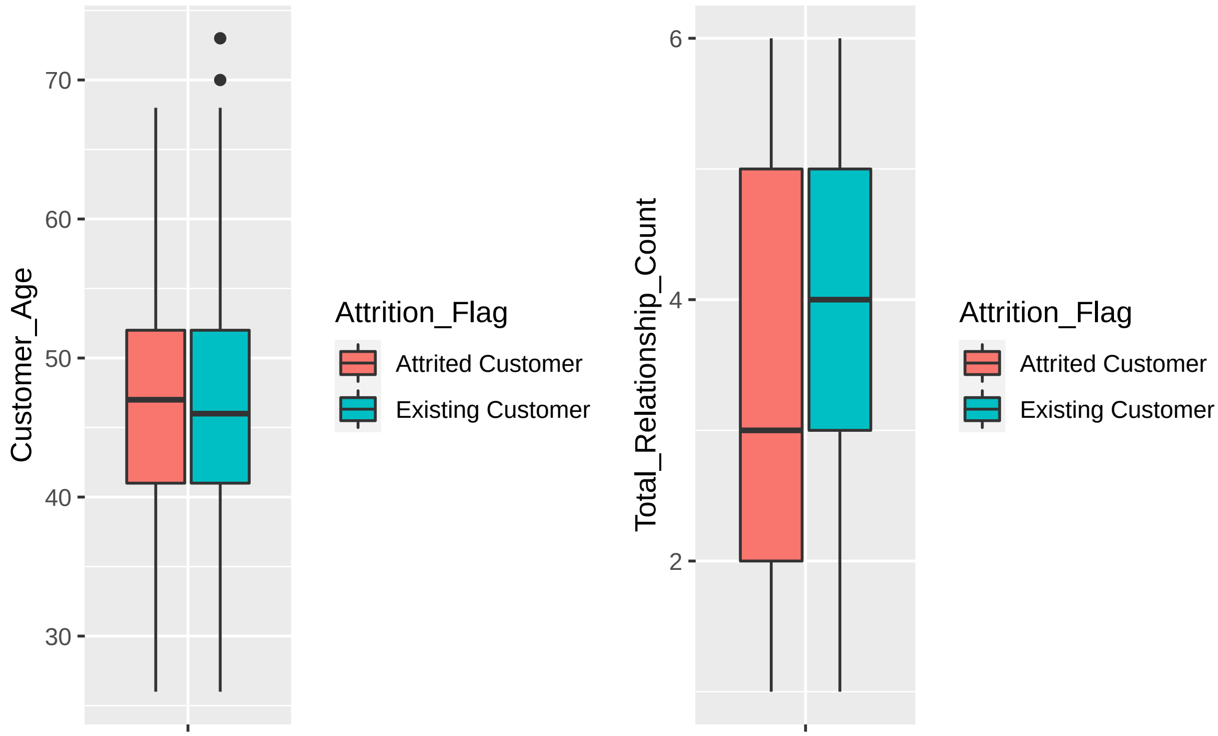
ge for different number of dependents in family and different number o

```
fig6 <-   ggarrange(
          ggplot(data, aes(y= Customer_Age, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Total_Relationship_Count, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig6)
```




```
annotate_figure(fig6, bottom  = text_grob("Attrition Percentage in Age and Relationship counts", col =
```

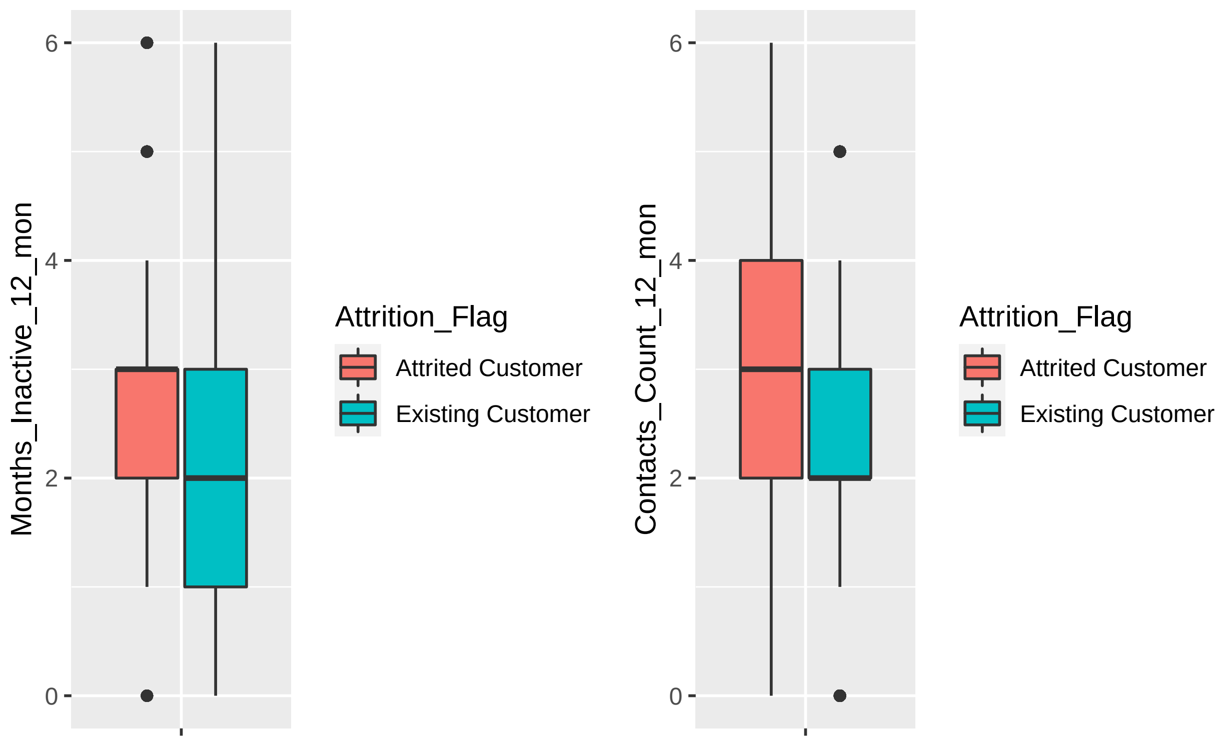


Attrition Percentage in Age and Relationship counts

```
fig7 <-   ggarrange(
          ggplot(data, aes(y= Months_Inactive_12_mon, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Contacts_Count_12_mon, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig7)
```

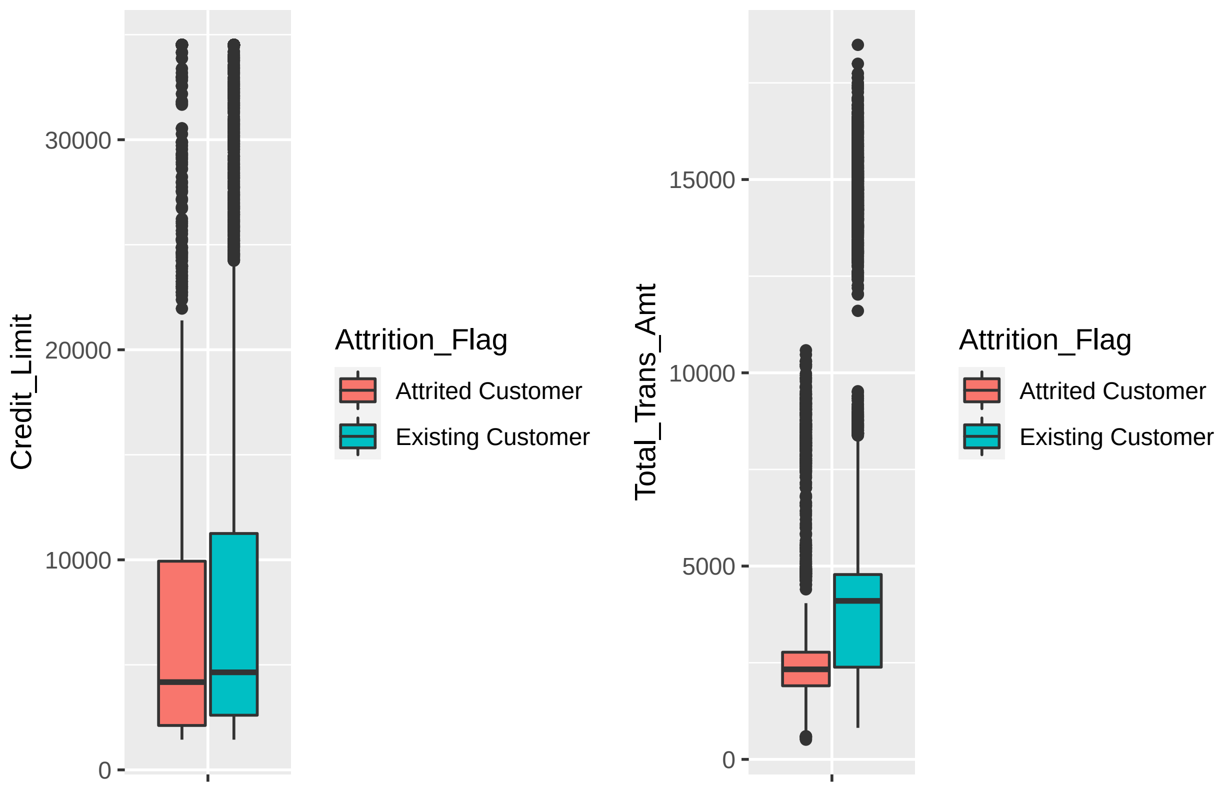

```
annotate_figure(fig7, bottom  = text_grob("Attrition Percentage in inactivity and number of contracts",
```



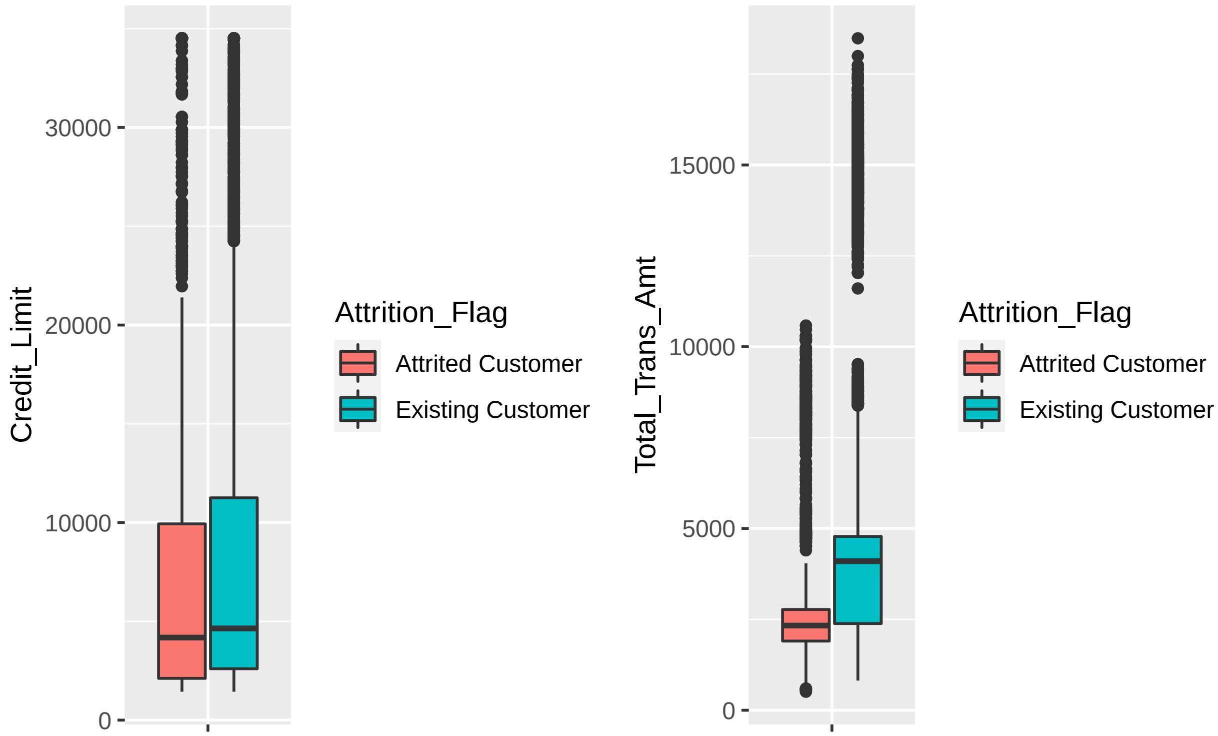
Attrition Percentage in inactivity and number of contracts

```
fig8 <-   ggarrange(
          ggplot(data, aes(y= Credit_Limit, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Total_Trans_Amt, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig8)
```



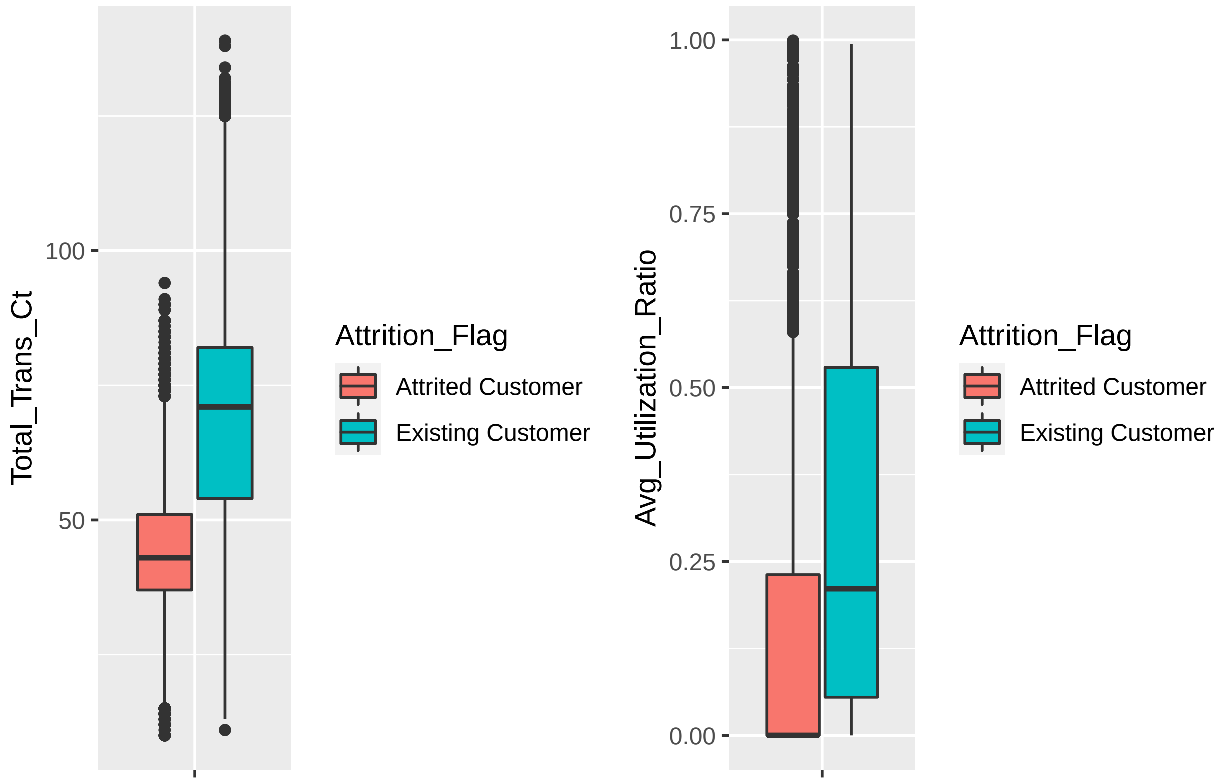

```
annotate_figure(fig8, bottom  = text_grob("Attrition Percentage in different levels of credit limit tran
```



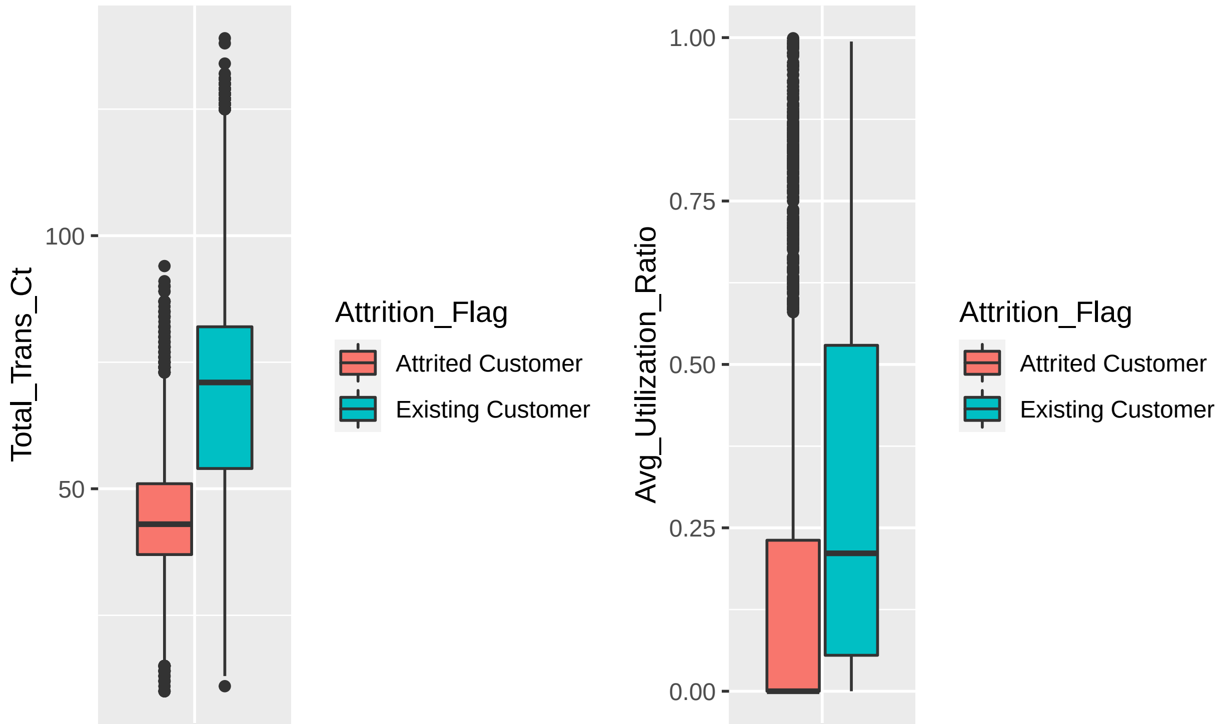
Attrition Percentage in different levels of credit limit transaction levels

```
fig9 <-   ggarrange(
          ggplot(data, aes(y= Total_Trans_Ct, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Avg_Utilization_Ratio, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig9)
```




```
annotate_figure(fig9, bottom  = text_grob("Attrition Percentage in number of transactions and utilizati
```



Attrition Percentage in number of transactions and utilization ratio

```
##            Attrition_Flag             Customer_Age                   Gender
##                  "factor"                "integer"                 "factor"
##           Dependent_count          Education_Level           Marital_Status
##                 "integer"                 "factor"                 "factor"
##           Income_Category            Card_Category           Months_on_book
##                  "factor"                 "factor"                "integer"
## Total_Relationship_Count   Months_Inactive_12_mon    Contacts_Count_12_mon
##                 "integer"                "integer"                "integer"
##              Credit_Limit          Total_Trans_Amt           Total_Trans_Ct
##                 "numeric"                "integer"                "integer"
##     Avg_Utilization_Ratio
##                 "numeric"
```

We now split the data into a training and a test sample, we use the training sample to train our model and the test sample to test our predictions to assess the power of our models.

```
smp_size <- floor(0.75 * nrow(data))

## set the seed to make your partition reproducible
set.seed(12345)
train_ind <- sample(seq_len(nrow(data)), size = smp_size)

train <- data[train_ind, ]
test <- data[-train_ind, ]
```

```
prop.table(table(train$Attrition_Flag))
```

```
##
```

```
## Attrited Customer Existing Customer
##          0.158262          0.841738
```

Since Attrition_Flag is a character variable with two possible values: either "Existing Customer" or "Attrited Customer" for modeling purposes we recode this variable into a factor with levels 0 and 1, where 1 represents a customer that has left the company when the person is still a customer at the company.

```
#recoding attrition_flag
train$Attrition_Flag <-ifelse(train$Attrition_Flag=="Attrited Customer",1,0)
test$Attrition_Flag <- ifelse(test$Attrition_Flag=="Attrited Customer", 1,0)


#logistic regression
glm <- glm(Attrition_Flag ~., data = train, family = "binomial")
summary(glm)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6806  -0.4127  -0.1936  -0.0800   3.3684
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   4.916e+00  4.913e-01  10.007  < 2e-16 ***
## Customer_Age                 -5.264e-04  8.325e-03  -0.063 0.949579
## GenderM                      -1.009e+00  1.652e-01  -6.105 1.03e-09 ***
## Dependent_count               1.094e-01  3.241e-02   3.375 0.000737 ***
## Education_LevelDoctorate      3.667e-01  2.200e-01   1.667 0.095536 .
## Education_LevelGraduate       1.094e-01  1.520e-01   0.720 0.471788
## Education_LevelHigh School   -3.988e-02  1.620e-01  -0.246 0.805586
## Education_LevelPost-Graduate  2.768e-01  2.222e-01   1.246 0.212839
## Education_LevelUneducated     9.045e-02  1.703e-01   0.531 0.595397
## Education_LevelUnknown        1.984e-01  1.683e-01   1.179 0.238356
## Marital_StatusMarried        -5.234e-01  1.698e-01  -3.082 0.002054 **
## Marital_StatusSingle          1.005e-01  1.705e-01   0.589 0.555614
## Marital_StatusUnknown         1.198e-01  2.132e-01   0.562 0.574139
## Income_Category$40K - $60K   -1.214e+00  2.208e-01  -5.499 3.82e-08 ***
## Income_Category$60K - $80K   -9.105e-01  1.905e-01  -4.780 1.75e-06 ***
## Income_Category$80K - $120K  -5.444e-01  1.776e-01  -3.065 0.002178 **
## Income_CategoryLess than $40K -1.048e+00  2.401e-01  -4.365 1.27e-05 ***
## Income_CategoryUnknown       -1.166e+00  2.517e-01  -4.634 3.58e-06 ***
## Card_CategoryGold             1.087e+00  3.964e-01   2.742 0.006111 **
## Card_CategoryPlatinum         2.162e+00  7.218e-01   2.996 0.002740 **
## Card_CategorySilver           5.485e-01  2.070e-01   2.650 0.008055 **
## Months_on_book               -1.004e-02  8.315e-03  -1.208 0.227239
## Total_Relationship_Count     -4.734e-01  3.007e-02 -15.746  < 2e-16 ***
## Months_Inactive_12_mon        4.953e-01  4.134e-02  11.982  < 2e-16 ***
## Contacts_Count_12_mon         5.425e-01  3.968e-02  13.672  < 2e-16 ***
## Credit_Limit                 -6.085e-05  7.126e-06  -8.539  < 2e-16 ***
## Total_Trans_Amt               4.121e-04  2.497e-05  16.504  < 2e-16 ***
## Total_Trans_Ct               -1.139e-01  3.964e-03 -28.738  < 2e-16 ***
## Avg_Utilization_Ratio        -2.959e+00  1.844e-01 -16.048  < 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: 6634.6  on 7594  degrees of freedom
## Residual deviance: 3958.3  on 7566  degrees of freedom
## AIC: 4016.3
##
## Number of Fisher Scoring iterations: 6
```

```r
pred <- predict(glm, data = train, type = "response")
# confusion matrix on training set
conmat <- table(train$Attrition_Flag, pred >= 0.5)
#show confusion matrix
print(conmat)
```

```
##
##     FALSE TRUE
##   0  6149  244
##   1   590  612
```

```r
accuracy <- (6149+612)/nrow(train)
specificity <- 6149/(6149+244)
sensitivity <- 612/(612+590)
precision <- 612/(612+244)
# observations on the test set
predtest <- predict(glm, newdata = test, type = "response")
conMattest <- table(test$Attrition_Flag, predtest >= 0.5)
#show confusion matrix
print(conMattest)
```

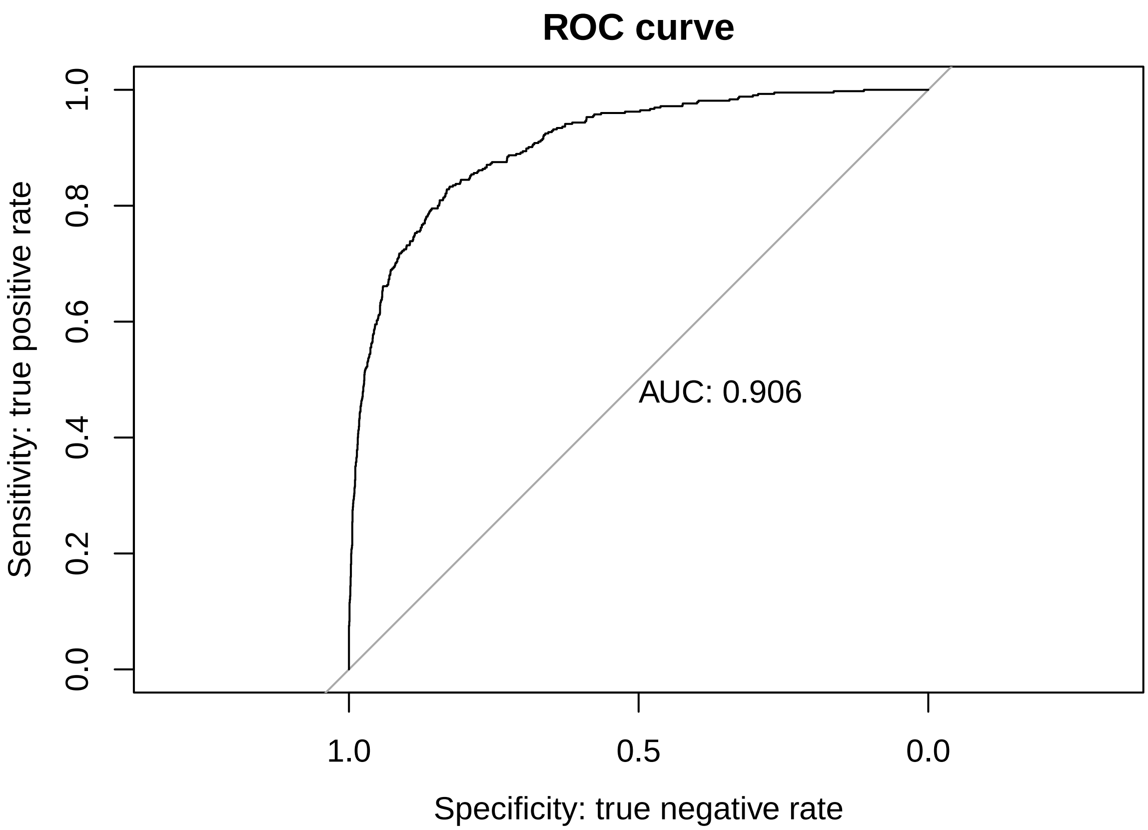
```
##
##     FALSE TRUE
##   0  2029   78
##   1   189  236
```

```r
accuracytest <- (2029+236)/nrow(test)
specificitytest <- 2029/(2029+78)
sensitivitytest <- 236/(236+189)
precisiontest <- 236/(236+78)
```

```r
par(mai=c(.9,.8,.2,.2))
plot(roc(test$Attrition_Flag, predtest), print.auc=TRUE,
     col="black", lwd=1, main="ROC curve", xlab="Specificity: true negative rate", ylab="Sensitivity: t
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

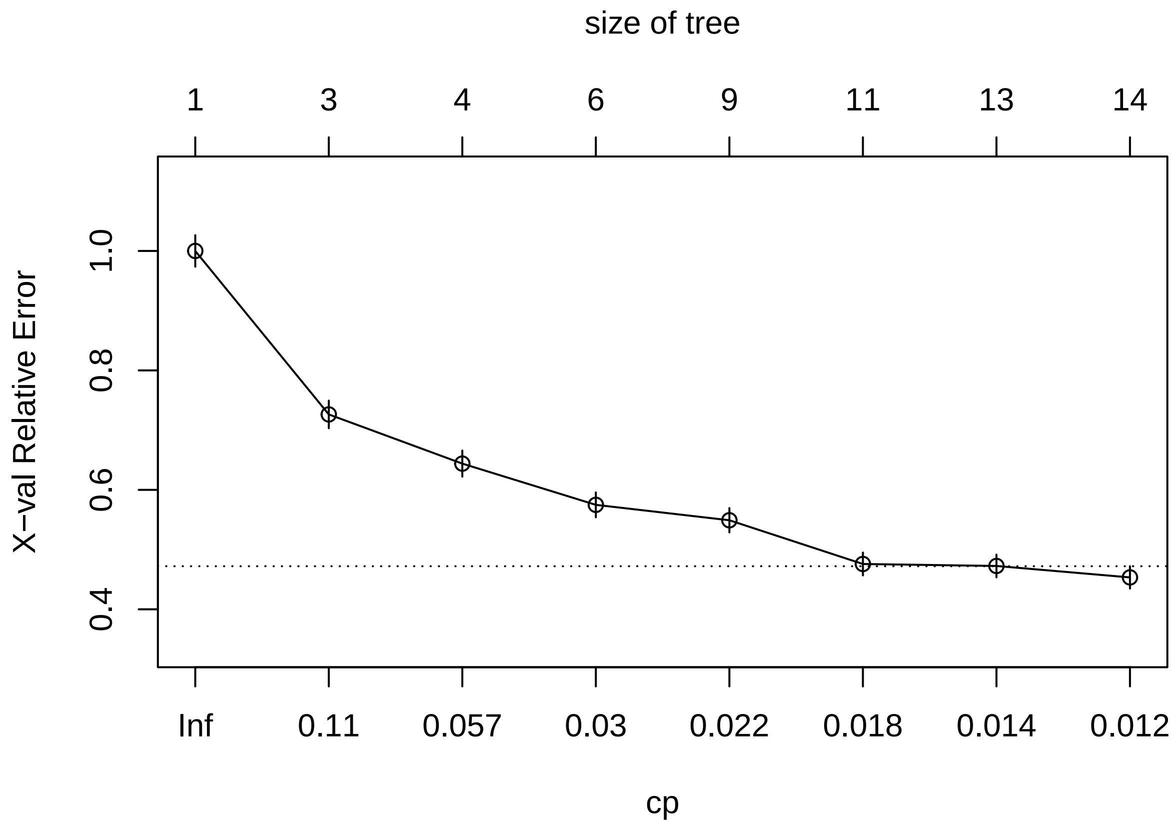



```
logisticvariableimportance <- varImp(glm, scale = FALSE)
print(logisticvariableimportance)
```

```
##                                  Overall
## Customer_Age                   0.0632359
## GenderM                        6.1052300
## Dependent_count                3.3753243
## Education_LevelDoctorate       1.6668892
## Education_LevelGraduate        0.7195736
## Education_LevelHigh School     0.2461248
## Education_LevelPost-Graduate   1.2457979
## Education_LevelUneducated      0.5310317
## Education_LevelUnknown         1.1791066
## Marital_StatusMarried          3.0822941
## Marital_StatusSingle           0.5893686
## Marital_StatusUnknown          0.5619667
## Income_Category$40K - $60K     5.4989659
## Income_Category$60K - $80K     4.7802932
## Income_Category$80K - $120K    3.0648737
## Income_CategoryLess than $40K  4.3652086
## Income_CategoryUnknown         4.6344535
## Card_CategoryGold              2.7417825
## Card_CategoryPlatinum          2.9955012
## Card_CategorySilver            2.6497523
## Months_on_book                 1.2075007
## Total_Relationship_Count      15.7457192
```

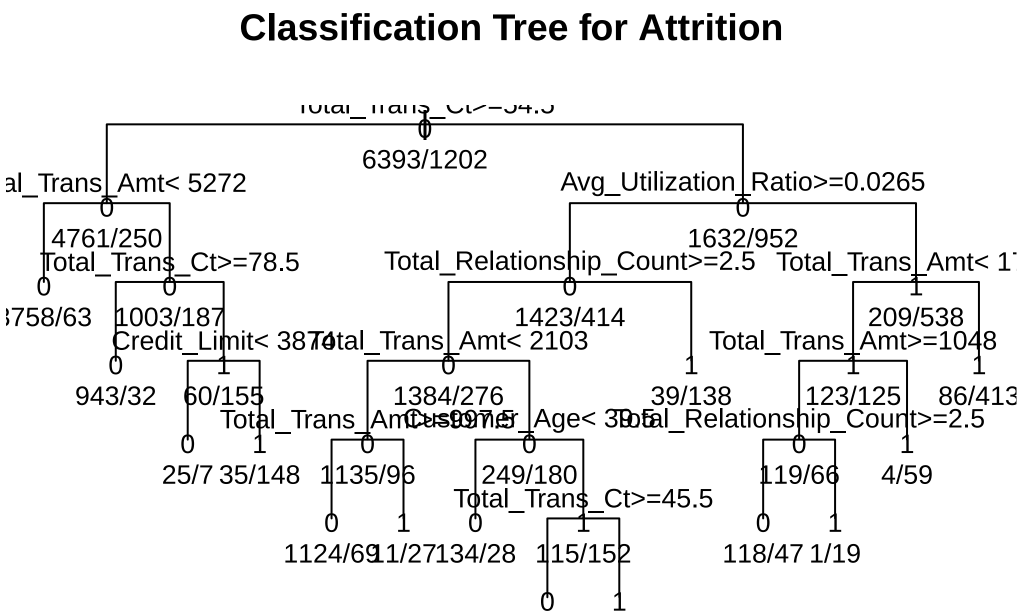
```
## Months_Inactive_12_mon         11.9818320
## Contacts_Count_12_mon          13.6722308
## Credit_Limit                    8.5389789
## Total_Trans_Amt                16.5035498
## Total_Trans_Ct                 28.7383614
## Avg_Utilization_Ratio          16.0475677
```

```
tree <- rpart(Attrition_Flag ~., method = "class", data = train)
printcp(tree)
plotcp(tree)
```




```
summary(tree)
```

```
# plot tree
plot(tree, uniform=TRUE,
   main="Classification Tree for Attrition")
text(tree, use.n=TRUE, all=TRUE, cex=.8)
```




```
#library caret is a comprehensive library support all sorts of model analysis
library(caret)
options(digits=4)
# assess the model's accuracy with train dataset by make a prediction on the train data.
Predict_model1_train <- predict(tree, train, type = "class")
#build a confusion matrix to make comparison
conMat <- confusionMatrix(as.factor(Predict_model1_train), as.factor(train$Attrition_Flag))
#show confusion matrix
conMat$table
```

```
##             Reference
## Prediction     0    1
##          0 6190  291
##          1  203  911
```

```
sensitivity(conMat$table)
```

```
## [1] 0.9682
```

```
specificity(conMat$table)
```

```
## [1] 0.7579
```

```
print(accuracy <- (6190+911)/(6190+911+291+203))
```

```
## [1] 0.935
```

```
print(precision <- 911/(911+203))
```

```
## [1] 0.8178
```

The model looks to do a decent job, our sensitivity seems to be quite higher than our specificity, which implies that our model is better at correctly classifying clients that left than at finding true loyal customers. This could be because of ...

Now that we have constructed the model we proceed by predicting the values in the test set in order to assess the suitability of the model.

```
Predict_model1_test <- predict(tree, test, type = "class")

conMattest <- confusionMatrix(as.factor(Predict_model1_test), as.factor(test$Attrition_Flag))

conMattest$table
```

```
##           Reference
## Prediction    0    1
##          0 2032  110
##          1   75  315
```

```
sensitivity(conMattest$table)
```

```
## [1] 0.9644
```

```
specificity(conMattest$table)
```

```
## [1] 0.7412
```

```
print(accuracy <- (2032+315)/(2032+315+110+75))
```

```
## [1] 0.9269
```

```
print(precision <- 315/(315+75))
```

```
## [1] 0.8077
```

There is not much difference between the accuracy for our model when comparing for the test and training set. The Sensitivity is slightly higher(0.01) and the specificity slightly lower(0.01). The accuracy is slightly lower than when predicting on the training set, however the difference is marginal

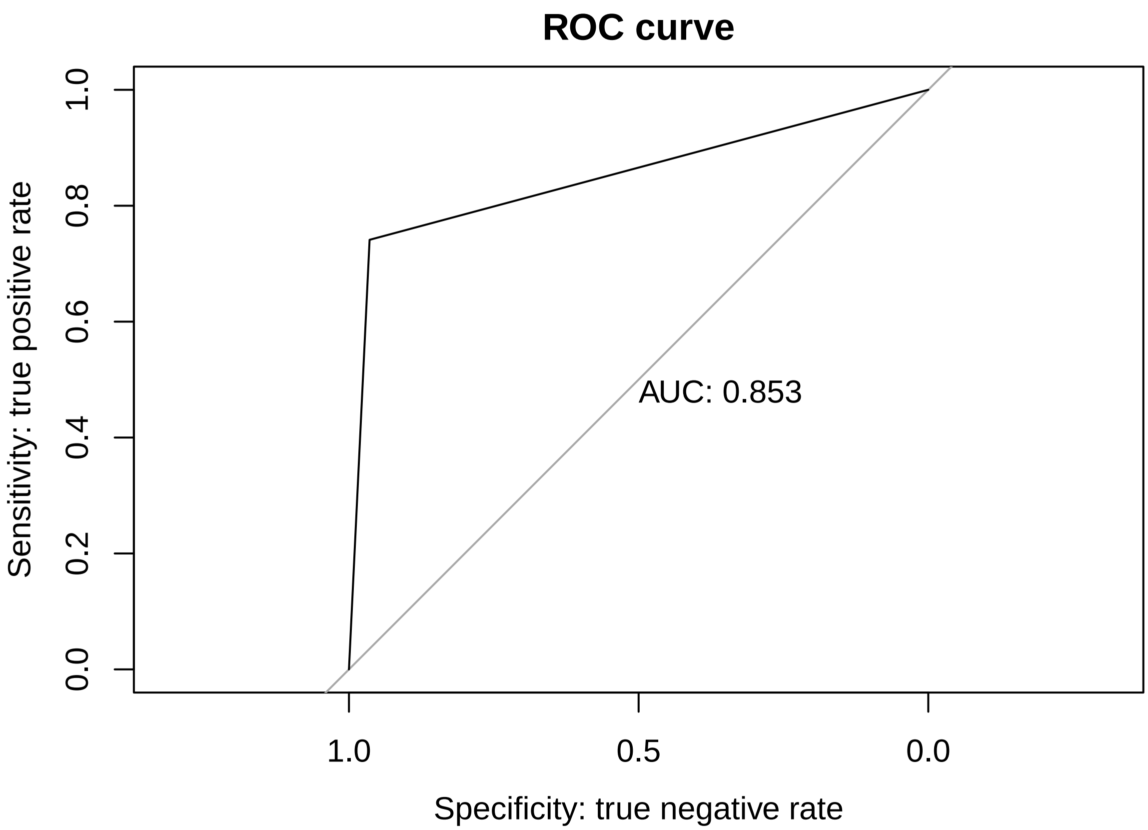
```
str(Predict_model1_test)
```

```
##  Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "names")= chr [1:2532] "6" "11" "12" "13" ...
```

```
par(mai=c(.9,.8,.2,.2))
plot(roc(test$Attrition_Flag, as.numeric(Predict_model1_test)), print.auc=TRUE,
     col="black", lwd=1, main="ROC curve", xlab="Specificity: true negative rate", ylab="Sensitivity: t
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



A 3rd type of model that we could implement is a random forest. CART decision trees are easily interpretable, however, output can be … because of … Therefore we implement a random forest model that uses a bagging procedure producing … regression trees and takes the average of each regression tree to improve the .. of the model results.

```
treevariableimportance <- varImp(tree, scale = FALSE)
print(treevariableimportance)
```

```
##                          Overall
## Avg_Utilization_Ratio     586.33
## Contacts_Count_12_mon     137.53
## Credit_Limit              205.62
## Customer_Age               31.69
## Gender                     80.31
## Months_Inactive_12_mon    203.83
## Months_on_book             17.75
## Total_Relationship_Count  413.91
## Total_Trans_Amt           872.06
## Total_Trans_Ct            625.79
## Dependent_count             0.00
## Education_Level             0.00
## Marital_Status              0.00
## Income_Category             0.00
## Card_Category               0.00
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

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

```
train$Attrition_Flag <- as.character(train$Attrition_Flag)
train$Attrition_Flag <- as.factor(train$Attrition_Flag)
rf <- randomForest(Attrition_Flag ~ ., , data = train, proximity=FALSE,importance = FALSE)
print(rf)
```

```
##
## Call:
##  randomForest(formula = Attrition_Flag ~ ., data = train, , proximity = FALSE,      importance = FALS
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 5.29%
## Confusion matrix:
##      0   1 class.error
## 0 6297  96     0.01502
## 1  306 896     0.25458
```

```
summary(rf)
```

```
##                 Length Class  Mode
## call                6  -none- call
## type                1  -none- character
## predicted        7595  factor numeric
## err.rate         1500  -none- numeric
## confusion           6  -none- numeric
## votes           15190  matrix numeric
## oob.times        7595  -none- numeric
## classes             2  -none- character
## importance         15  -none- numeric
## importanceSD        0  -none- NULL
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                7595  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
predrf <- predict(rf, data = "train", type = "response")
print(rftab <- table(predrf, train$Attrition_Flag))
```

```
##
```

```
## predrf    0    1
##      0 6297  306
##      1   96  896
```

```r
print(accuracyrf <- (6290+914)/nrow(train))
```

```
## [1] 0.9485
```

```r
print(sensitivityrf <- 914/(914+103))
```

```
## [1] 0.8987
```

```r
print(precisionrf <- 914/(914+288))
```

```
## [1] 0.7604
```

```r
print(specificityrf <- 6290/(6290 + 288))
```

```
## [1] 0.9562
```

```r
predrftest <- predict(rf, newdata = test, type = "response")
print(rftabtest <- table(predrftest, test$Attrition_Flag))
```

```
##
## predrftest    0    1
##          0 2076   92
##          1   31  333
```

```r
print(accuracyrftest <- (2073+335)/nrow(test))
```

```
## [1] 0.951
```

```r
print(sensitivityrftest <- 335/(335+34))
```

```
## [1] 0.9079
```

```r
print(precisionrftest <- 335/(335+90))
```

```
## [1] 0.7882
```

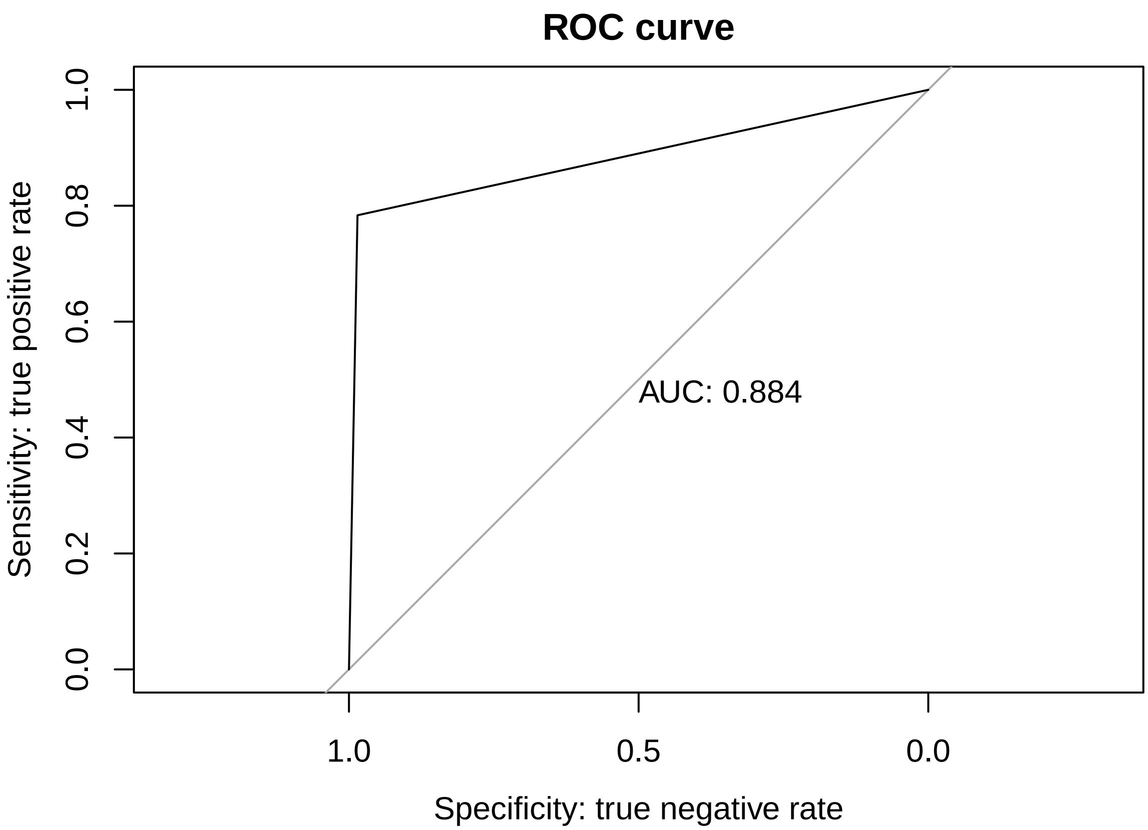
```r
print(specificityrftest <- 2073/(2073 + 90))
```

```
## [1] 0.9584
```

```r
par(mai=c(.9,.8,.2,.2))
plot(roc(test$Attrition_Flag, as.numeric(predrftest)), print.auc=TRUE,
     col="black", lwd=1, main="ROC curve", xlab="Specificity: true negative rate", ylab="Sensitivity: t
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```




```
library(pROC)
glm.roc <- roc(response = test$Attrition_Flag, predictor = as.numeric(predtest))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

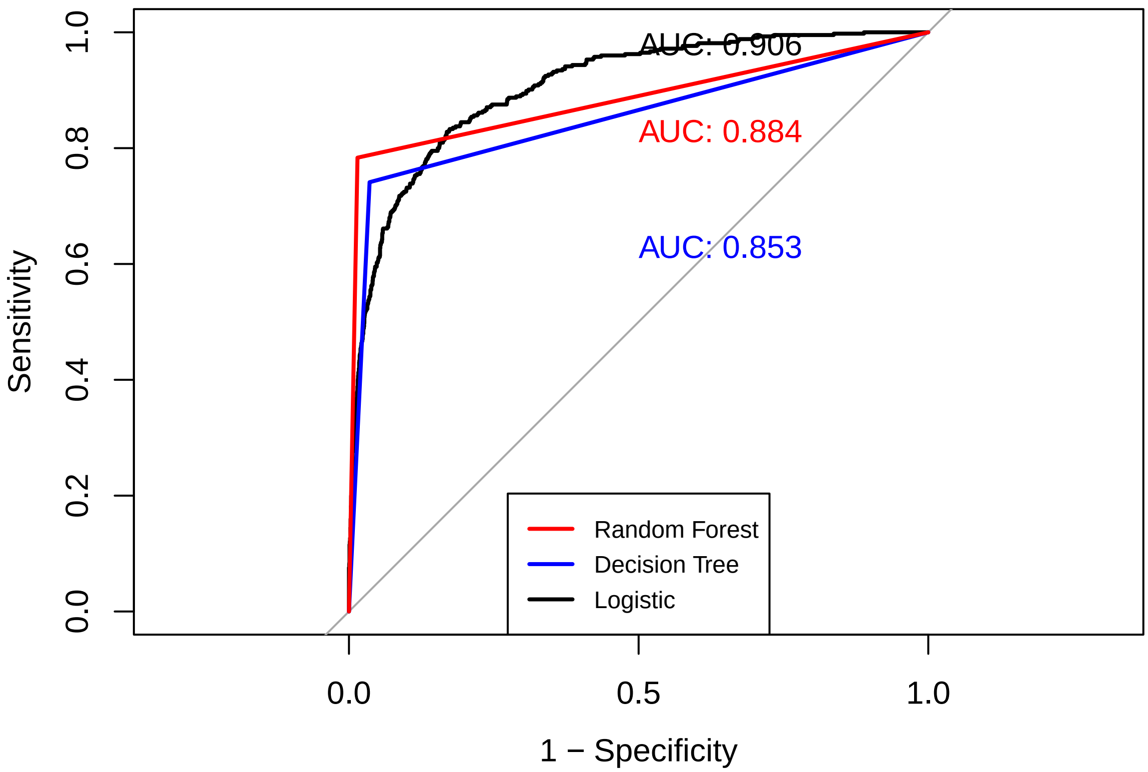
```
rpart.roc <- roc(response = test$Attrition_Flag, predictor = as.numeric(Predict_model1_test))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
rf.roc <- roc(response = test$Attrition_Flag, predictor = as.numeric(predrftest))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(glm.roc,      legacy.axes = TRUE, print.auc.y = 1.0, print.auc = TRUE)
plot(rpart.roc, col = "blue", add = TRUE, print.auc.y = 0.65, print.auc = TRUE)
plot(rf.roc, col = "red" , add = TRUE, print.auc.y = 0.85, print.auc = TRUE)
legend("bottom", c("Random Forest", "Decision Tree", "Logistic"),
       lty = c(1,1), lwd = c(2, 2), col = c("red", "blue", "black"), cex = 0.75)
```


AUC: 0.906
AUC: 0.884
AUC: 0.853
Random Forest
Decision Tree
Logistic
Sensitivity
1 − Specificity
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0