Capstone Project

1. Introduction

Lending Club is a Peer to Peer Lending Platform that makes use of investor's money to provide the funds for the loans that they provide for their customers. Almost anyone can become an investor and provide money for loans. Why would they do this?

Well, depending on the loan that you're providing money for, you may make more money than you've invested, depending on customer attributes.

High risk customers are a bigger risk because their credit history may not be perfect, which means they may have difficulty paying back the loan. However, since they are a higher risk, the loan will cost more and you can make more money on your investment.

A lower risk customer is a much safer bet, but since their credit is great, this indicates they have much higher chance of paying their loan back, so their loan's interest rate is lower and you won't make as much money off of that investment.

Then there are the customers who fail to pay their loan back, meaning you don't get the money back that you invested. These are the loans you want to avoid investing in at all cost because the money that you provide for that loan will never be recovered.

For this project, we will be focused on the indicators that lead to customers Charging Off their loans so that you can identify and avoid loans that have indicators showing that they're going to charge off.

1.2. My Client - Who cares about loan defaults?

My target audience for this work is an (imaginary) investment firm that wants to take a more intelligent approach to putting their money to work for them. Whenever you make a bet, or invest in any product, you have a native 50% chance that you get it right. (You either make money or lose money). What I'm looking to do is put information that will move those odds above random chance into the hands of the people making financial decisions so that their portfolio will contain loans that will be paid back and that they will have the greatest chance of making money off of their portfolio.

1.3. The Data & Where to find it

I'm going to analyze the loan data that Lending Club provides to determine how their portfolio has been trending since 2012 (when the alternative lending industry really took off) through 2017. I will be using this data to determine if investing in a loan from Lending Club is a good choice based on their company's performance, as finding the most important variables to someone looking to invest their money into a loan and make a return. The data is available online for free in .csv format here:

https://www.lendingclub.com/info/download-data.action
(https://www.lendingclub.com/info/download-data.action)

2. The Approach

First, I would like to use statistical and machine learning models to determine which variables have the highest correlation to a loan's fate (whether it gets paid in full or charged off). Once I've determined what those variables are, I want to visualize them so we can see and describe exactly what the relationship between those correlated variables is. This will show us what the most important factors are that you should be looking for before determining whether you want to invest in a loan.

2.1. Basics

Why did I choose this data set for my Capstone Project? I had a couple reasons for doing so. The first, is that I have prior experience working in the lending industry, so this data is familiar to me and I have insights into what variables would be important for different steps of this project. The second is that it is a big data set, and I want experience handling big sets of data and overcoming the concerns inherent in them to prepare me for future projects. Additionally, the recommendations from this analysis could hypothetically be leveraged in the real world by a potential client, giving real value to the insights I've found in the data.

2.2. Initial Questions and Brainstorming

In this section, I'm going to be walking through my thought process as I begin to think about what's important to this report, what I predict will be the important variables and the questions I'll be trying to answer with my data analysis.

When looking at data, we have to make sure that we're not only looking at the right data, but also asking the right questions. When brainstorming and initially looking at the data, these are the questions I kept in mind:

1. What questions does the data lead me to ask?
2. What variables do I want to look for and understand?
3. Why did I choose those?

4. What insights do I think we can gather from this data?
5. What's the relevant business-driven reason we want to spend time answering these questions?

The following are the initial questions that jump to mind when thinking about this data set and what we can understand from it:

"What are the percentages of loans in each status each year compared to the current total?"

EX:(I'm making these numbers up):

Year: 2012 / 2013
Charged Off: 3% / 4%
Fully Paid: 97% / 96%

"Based off of these percentages what can we tell about the trend of the portfolio's loan statuses over the last 5 years?"

Either they're increasing, decreasing or staying steady, which has implications for Lending Club's business. If the trend in their portfolio is that it's declining, we may want to think twice about putting any money into their company at all. Thinking ahead regarding visualization, I think the best ways to visualize some of these variables would be bar charts.

Here I'd want to examine the overall view of the loans that are either Fully Paid – "Good Loans" or in Default/ Charged Off – "Bad Loans".

Some initial Variables I can think to examine here include:

- "Loan Amount" ($0 - $40,000) Data Point -> loan_amount
- "Loan Term" (36 & 60 Months) Data Point -> term
- "Loan Interest Rate" (0% - 40%) Data Point -> int_rate
- "Loan Grade" Data Point" -> grade

"What were the grades of the loans that were Defaulted on / Charged off and those that were Fully Paid?"

- Loan Grades (Good Vs. Bad Loans)
- Sub Grades (Good Vs. Bad Loans)

"What is the link between Grade, Interest Rate, & Default rate?"

- Data Point -> int_rate vs. grade vs. loan_status

"Based on a loan's grade & Loan's interest rate, what is the probability that they'll default? For each grade category?""

"What variables can we find in the consumer's data related to the grade of the loans?"

Variables I anticipate will have high correlation to good loans vs. bad loans:

- Borrower's Income -> "loan_amnt"
- Borrower's DTI -> "dti"
- Revolving credit utilization -> "revol_util"
- Total Credit Lines -> "total_Acc"
- Delinquents -> "delinq_2yrs"
- Bankruptcies -> "pub_rec_bankruptcies"
- public records -> "pub_rec"

"What consumer factors correlate strongly to defaulting on their loan?"

"For investors - what factors would a consumer have that would mean it is a good/bad idea to invest in a particular loan?" (Higher risk vs. Lower Risk)

These were the general guidelines I used as the starting point for my project, knowing that along the way I'm going to find information and uncover insights that may change the questions I want to ask and also shape how I will go about finding the answers to the question:

"What variables do we ultimately want to look for or avoid when investing in a Lending Club loan?"

Now that I have some framework of the questions I want to ask, let's get to work putting the data into a clean, usable format.

3. Data Wrangling - Collecting, Cleaning and Preparing Our Data

Setting my working directory:

```
setwd("~/R Studio Directory/Lending Club Loan Datsets 2007 - 2017/Loan Stats CS
V files")
```

Just double checking that the directory is correct:

```
getwd()
[1] "/Users/Broseidon/R Studio Directory/Lending Club Loan Datsets 2007 - 2017/
Loan Stats CSV files"
```

Directory set!

Time to Load The Libraries I'll need: library(tidyverse) library(ggplot2) library(caret) library(rpart) library(stringr) library(lubridate) library(rms) library(kernlab) library(reshape) library(corrgram) library(gridExtra) library(caTools) library(ROSE)

```
knitr::opts_chunk$set(echo = TRUE)
setwd("~/R Studio Directory/Lending Club Loan Datsets 2007 - 2017/Loan Stats CS
V files")
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 2.2.1     ✔ purrr   0.2.4
## ✔ tibble  1.4.2     ✔ dplyr   0.7.4
## ✔ tidyr   0.8.0     ✔ stringr 1.3.0
## ✔ readr   1.1.1     ✔ forcats 0.3.0
```

```
## ── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(caret)
```

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

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

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
library(stringr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(rms)
```

```
## Loading required package: Hmisc
```

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

```
##
## Attaching package: 'survival'
```

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

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

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

```
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

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

```
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
library(corrgram)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

```
library(caTools)
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

3.1. Loading the CSV Files into R as variables

There are 11 of them in total. Before moving, on, we also have to take the file labeled: “LoanStats_2012-2013.csv” and filter all of the Loans from 2012 into one File and from 2013 onto another CSV (I tried doing this project without splitting the files and it was a nightmare). So now we have 12 files total:

```
1) LoanStats_2012.csv
2) LoanStats_2013.csv
3) LoanStats_2014.csv
4) LoanStats_2015.csv
5) LoanStats_2016Q1.csv
6) LoanStats_2016Q2.csv
7) LoanStats_2016Q3.csv
8) LoanStats_2016Q4.csv
9) LoanStats_2017Q1.csv
10) LoanStats_2017Q2.csv
11) LoanStats_2017Q3.csv
12) LoanStats_2017Q4.csv
```

Let’s try to load the first file:

```
LDF1 <- read_csv("LoanStats_2012.csv")
```

Returns a Warning:

```
Warning: 381 parsing failures.
row # A tibble: 5 x 5 col      row             col               expected  actu
al expected    <int>           <chr>                  <chr>   <chr> actual 1  4
8007 funded_amnt_inv no trailing characters .191259 file 2  48066 funded_amnt_i
nv no trailing characters .222838 row 3 140131 funded_amnt_inv no trailing char
acters .183293 col 4 140413 funded_amnt_inv no trailing characters .292467 expe
cted 5 142384 funded_amnt_inv no trailing characters .420323 actual # ... with
1 more variables: file <chr>
```

It turns out that the warning relates to the “funded_ amount_inv” Column in the Excel file provided by Lending Club. Now, since this is just a warning and not an error we could ignore it but I’d rather not encounter any errors later. So let’s do a little work on the actual files themselves before continuing in R:

3.2. Process to fix parsing

```
1) Open the CSV file
2) Click on the Column D – "funded_amount"
3) Change the Data Type from General to Number
4) Click Remove Decimal
5) Click Remove Decimal again
```

This standardizes the numbers to round integers and allows us to load the file into the Data Frame. I had to repeat this process for each file and multiple variables that had parsing errors including:

```
"funded_ amount_inv" - Column E
“total_rec_late_fee” - Column AQ
“recoveries” - Column AR
“collection_recovery_fee" -Column AS
“annual_inc_joint” - Column BB
```

Let’s go ahead and read the files in now:

```
> LDF1 <- read_csv("LoanStats_2012.csv")
> LDF2 <- read_csv("LoanStats_2013.csv")
> LDF3 <- read_csv("LoanStats_2014.csv")
> LDF4 <- read_csv("LoanStats_2015.csv")
```

The files for 2016 & 2017 are split up into quarters instead of one Excel file for the years, so I’m going to load them in, bind them together into one data frame and save that as a CSV, so I can just load that CSV File:

```
LDF2016Q1 <- read_csv("LoanStats_2016Q1.csv")
LDF2016Q2 <- read_csv("LoanStats_2016Q2.csv")
LDF2016Q3 <- read_csv("LoanStats_2016Q3.csv")
LDF2016Q4 <- read_csv("LoanStats_2016Q4.csv")
LDF5 <-  rbind(LoanDF2016Q1, LoanDF2016Q2, LoanDF2016Q3, LoanDF2016Q4)
write.csv(LCLoanDF5, file = "LoanStats_2016.csv")

When we load it back in:
LDF5 <- read_csv("LoanStats_2016.csv")
```

We also get this warning:

```
Warning messages:
1: Missing column names filled in: 'X1' [1]
2: In rbind(names(probs), probs_f) :
  number of columns of result is not a multiple of vector length (arg 1)
```

This happens because when saving a .csv file from R to a directory then loading it back in R likes to add an additional column to the left with numbered observations. I don't get this with the other files because I didn't have to save them from R and then Reload them. I would just go into Excel and delete the column so I don't have to Null the 'X1' column every time I want to load the data, but the files are too big and cause my computer to freeze when I attempt to open them so let's try this:

```
LDF5[,"X1"] <- NULL
```

Success! Life is good. I'm going to have to do the same thing for 2017:

```
LDF2017Q1 <- read_csv("LoanStats_2017Q1.csv")
LDF2017Q2 <- read_csv("LoanStats_2017Q2.csv")
LDF2017Q3 <- read_csv("LoanStats_2017Q3.csv")
LDF2017Q4 <- read_csv("LoanStats_2017Q4.csv")
```

Now to use the 'rbind' function to unite 2017's data into one Frame from the separate files:

```
LDF6 <-  rbind(LDF2017Q1, LDF2017Q2, LDF2017Q3, LDF2017Q4)
write.csv(LDF6, file = "LoanStats_2017.csv")
LDF6 <- read_csv("LoanStats_2017.csv")
LDF6[,"X1"] <- NULL
```

Right here, I'm going to delete the original data frames from Q1,Q2,Q3 & Q4 for 2016 & 2017 since I now have combined each of those year's data into their own data frames and we won't be needing the info for each quarter again.

Awesome. I will most likely need to load these in again if I exit my session or turn my computer off, so to save time I'm going to condense everything here:

3.3. Loading the fixed CSV files into R

```
LDF1 <- read_csv("LoanStats_2012.csv")
LDF2 <- read_csv("LoanStats_2013.csv")
LDF3 <- read_csv("LoanStats_2014.csv")
LDF4 <- read_csv("LoanStats_2015.csv")
LDF5 <- read_csv("LoanStats_2016.csv")
LDF5[,"X1"] <- NULL
LDF6 <- read_csv("LoanStats_2017.csv")
LDF6[,"X1"] <- NULL
```

The variable “issue_d” has caused me a lot of trouble throughout this project so I’m going to get rid of “issue_d” right now to make my life much easier:

```
LDF1[,"issue_d"] <- NULL
LDF2[,"issue_d"] <- NULL
LDF3[,"issue_d"] <- NULL
LDF4[,"issue_d"] <- NULL
LDF5[,"issue_d"] <- NULL
LDF6[,"issue_d"] <- NULL
```

And instead, I’m going to hard-code the year into each data frame so that I can combine them together and still analyze them by year (in essence replacing the “issue_d” variable with a new variable I came up with - “Year”)

```
LDF1$Year <- 2012
LDF2$Year <- 2013
LDF3$Year <- 2014
LDF4$Year <- 2015
LDF5$Year <- 2016
LDF6$Year <- 2017
```

3.4. Using rbind to finally unite the files into one data frame

```
BIGDF <- rbind(LDF1, LDF2, LDF3, LDF4, LDF5, LDF6)
```

So now we finally have the files prepped and combined in the most parsimonious way for us to use the data. It turns out that this data set has 1,722,935 Observations and 145 Variables! We will definitely need to clean and trim this data set to make it more usable so that we can analyze the data and see what results we find.

3.5. Cleaning Data and Eliminating Nulls/Fields with Zeroes/NAs

Now, Here I can use a function to find the percentage of null values in a given column: I’m going to determine the percentage of “NA” fields and delete anything greater than 80% NA:

```
BIGDF <- BIGDF
missing_values <- sapply(BIGDF, function(x){
    percentage <- sum(is.na(x))/length(x)
    percentage < 0.2
    })
BIGDF <- BIGDF[,missing_values ==TRUE]
BIGDF
```

```
This removes 58 variables:
[1] "id"
[2] "member_id"
[18] "url"
[19] "desc"
[28] "mths_since_last_delinq"
[29] "mths_since_last_record"
[47] "next_pymnt_d"
[50] "mths_since_last_major_derog"
[53] "annual_inc_joint"
[54] "dti_joint"
[55] "verification_status_joint"
[59] "open_acc_6m"
[60] "open_act_il"
[61] "open_il_12m"
[62] "open_il_24m"
[63] "mths_since_rcnt_il"
[64] "total_bal_il"
[65] "il_util"
[66] "open_rv_12m"
[67] "open_rv_24m"
[68] "max_bal_bc"
[69] "all_util"
[71] "inq_fi"
[72] "total_cu_tl"
[73] "inq_last_12m"
[86] "mths_since_recent_bc_dlq"
[88] "mths_since_recent_revol_delinq"
[111] "revol_bal_joint"
[112] "sec_app_earliest_cr_line"
[113] "sec_app_inq_last_6mths"
[114] "sec_app_mort_acc"
[115] "sec_app_open_acc"
[116] "sec_app_revol_util"
[117] "sec_app_open_act_il"
[118] "sec_app_num_rev_accts"
[119] "sec_app_chargeoff_within_12_mths"
[120] "sec_app_collections_12_mths_ex_med"
[121] "sec_app_mths_since_last_major_derog"
[123] "hardship_type"
[124] "hardship_reason"
[125] "hardship_status"
[126] "deferral_term"
[127] "hardship_amount"
[128] "hardship_start_date"
```

```
[129] "hardship_end_date"
[130] "payment_plan_start_date"
[131] "hardship_length"
[132] "hardship_dpd"
[133] "hardship_loan_status"
[134] "orig_projected_additional_accrued_interest"
[135] "hardship_payoff_balance_amount"
[136] "hardship_last_payment_amount"
[139] "debt_settlement_flag_date"
[140] "settlement_status"
[141] "settlement_date"
[142] "settlement_amount"
[143] "settlement_percentage"
[144] "settlement_term"
```

and now I'm going to remove the columns with only one unique value (This only removes the variable "policy_code"):

```
unique_values <- sapply(BIGDF, function(x) {
size <- length(unique(x[!is.na(x)]))
size > 1
})
BIGDF <- BIGDF[,unique_values == TRUE]
BIGDF
```

3.6. Manually removing variables we can intuititvely see won't help us

We're left with a data frame of 87 variables. Looking at the classes we can see right off the bat that there are some additional variables we can remove and others we have to think about:

1. "funded_amnt" & "funded_amnt_inv" are both the same as "loan_amnt", so we can take those two out as well
2. "emp_title" - won't help us in our analysis
3. "pymnt_plan" - All the observations are 'N'
4. "zip_code" - The last two elements are xx
5. "initial_list_status" - all observations are 'f'
6. "application_type" - Always individual
7. "collections_12_mths_ex_med","acc_now_delinq","chargeoff_within_12_mths","delinq_amnt" - All four of these are mostly zeroes
8. "next_pymnt_d" - Mostly blank
9. "title" - What lending club stated the loan's use was for. "Title", much like "desc" is a field that either a representative or customer filled in. This means that each observation is going to be unique and not going to have statistical value. The only analysis I could do with these would maybe be some type of text mining, but it's much easier to use the categorical "purpose" variable if I want to see if there is a correlation to what the loan was used for and whether or not it was paid back
10. "earliest_cr_line","last_pymnt_d","last_credit_pull_d" - I'm going to throw out any variable that has a date value in order to simplify the analysis since date values have not been converting correctly for me. It means I may have to sacrifice some accuracy, but at least I'll be able to build

the model.

These 10 are the ones that jump out at me immediately. Including these and after looking through the dictionary that Lending Club provided of definitions for each variable, I've come up with the list of variables that I can tell won't help us. I would love to use a model here to help eliminate the variables that have low correlation, but this data set now has 86 variables and 1,722,035 observations. Using a model right now would take an incredibly long amount of time and there are variables that we can see without doing any analysis that aren't useful. I'm going to start to whittle them down intuitively and once I've taken out all of the variables that can tell won't help us, then I'll use a Machine Learning algorithm to make sure we're left with only the most important factors related to Charge Offs.

I've put them all into one vector named "rem_cols" so that I can remove them easily from the data frame:

```
rem_cols <- c("funded_amnt",
              "funded_amnt_inv",
              "installment",
              "emp_title",
              "emp_length",
              "verification_status",
              "pymnt_plan",
              "title",
              "zip_code",
              "earliest_cr_line",
              "initial_list_status",
              "application_type",
              "collections_12_mths_ex_med",
              "acc_now_delinq",
              "chargeoff_within_12_mths",
              "delinq_amnt",
              "title",
              "last_pymnt_d",
              "last_credit_pull_d",
              "total_rec_late_fee",
              "recoveries",
              "collection_recovery_fee",
              "next_pymnt_d",
              "out_prncp_inv",
              "total_pymnt_inv",
              "hardship_flag",
              "inq_last_6mths",
              "disbursement_method",
              "debt_settlement_flag",
              "revol_bal_joint",
              "total_bc_limit",
              "last_pymnt_amount",
              "last_pymnt_amnt",
              "open_act_il",
              "application_type",
              "total_rec_prncp",
              "out_prncp",
              "mths_since_recent_bc",
              "tot_hi_cred_lim",
              "percent_bc_gt_75",
              "bc_util",
              "num_bc_sats",
              "num_sats",
              "bc_open_to_buy",
              "total_il_high_credit_limit",
              "num_rev_accts",
              "num_bc_tl",
              "mo_sin_old_acct",
              "mo_sin_old_il_acct",
              "total_rev_hi_lim",
              "num_actv_bc_tl",
              "num_op_rev_tl",
              "num_op_rev_tl_bal_gt_0",
```

```
              "num_rev_tl_bal_gt_0",
              "num_tl_120dpd_2m",
              "mths_since_recent_inq",
              "total_pymnt",
              "total_rec_int",
              "pct_tl_nvr_dlq",
              "num_tl_90g_dpd_24m",
              "num_tl_30dpd",
              "num_il_tl",
              "mo_sin_rcnt_tl")
```

Now to remove those columns:

```
BIGDF <- BIGDF[, !names(BIGDF) %in% rem_cols]
```

Let's remove some rows that have blank rows for the variable "revol_util":

```
BIGDF %>% filter(BIGDF$revol_util != "") -> BIGDF
```

Let's remove those pesky "%" signs from "int_rate" and "revol_util":

```
BIGDF$int_rate <- as.numeric(gsub('%', '',BIGDF$int_rate))
BIGDF$revol_util <- as.numeric(gsub('%', '',BIGDF$revol_util))
BIGDF$revol_util <- as.numeric(gsub('%', '',BIGDF$revol_util)) / 100
```

Let's get rid of the rows containing loans that are in statuses other than "Fully Paid" or "Charged Off" as these all represent loans that are currently being paid on or have extenuating circumstances. Pro tip - I just learned that any time we have a chunk of code we want to run, it's easiest to highlight it in R Markdown and click 'Run' OR just highlight it and hit Cmd + Return. This makes life SO much easier!!

```
BIGDF %>% filter(loan_status != 'Current') -> BIGDF
BIGDF %>% filter(loan_status != 'Default') -> BIGDF
BIGDF %>% filter(loan_status != 'Issued') -> BIGDF
BIGDF %>% filter(loan_status != 'In Grace Period') -> BIGDF
BIGDF %>% filter(loan_status != 'Late (16-30 days)') -> BIGDF
BIGDF %>% filter(loan_status != 'Late (31-120 days)') -> BIGDF
```

3.7. Re-organizing the data into a 'tidy' structure

Let's make a variable to store these categorical items:

```
categ_cols <- c('term','grade','sub_grade','home_ownership','purpose','addr_sta
te')
```

A Separate smaller Data Frame containing only those categorical variables we haven't already taken out:

```
BIGDF %>% select(one_of(categ_cols)) -> df_categ
```

And a Data Frame with only the numerical values:

```
BIGDF %>% select(-one_of(categ_cols)) -> df_numeric
```

Combine everything back together:

```
BIGDF <- cbind(df_categ,df_numeric)
```

I'm also going to add a column denoting factors for charged off and fully paid for data analysis:

```
Bad_indicators <- c("Charged Off")
BIGDF$is_bad <- ifelse(BIGDF$loan_status %in% Bad_indicators, 1,
              ifelse(BIGDF$loan_status =="", NA, 0))
```

3.8. Saving and loading the final cleaned Data Frame

Great! Now that I have finally cleaned and organized the data I have one big data frame to use to figure out what the variables that are important to this project are. From 145 variables and 1,722,935 observations to 721,719 observations and only 32 variables. Much easier to work with! Now to save - Remember, saving is key!:

```
write.csv(BIGDF, file = "BIGDF.csv")
BIGDF <- read_csv("BIGDF.csv")
BIGDF[,"X1"] <- NULL
BIGDF <- as.data.frame(BIGDF)
```

4. Exploratory Data Analysis (EDA) - Using Statistics and Machine Learning to determine the Statistical signifigance of variables

Getting the Baseline accuracy

```
BIGDFbaseline = table(BIGDF$is_bad, sign(BIGDF$is_bad))
BIGDFbaseline
returns:
         0      1
  0 573071      0
  1      0 148648
```

The amount of Charged Off loans in the data set is 148,648. Now, to find out the native accuracy of our data set, start by adding the number of charged-off loans to the number of fully paid:

```
148648 + 573071
returns:
[1] 721719
```

Now we have the amount of Charged off loans and the Total number of loans. Divide Charge Offs by the Total number of Loans:

```
148648/721719
returns:
[1] 0.2059638
```

and finally subtract from 1:

```
1-0.2059638
returns:
[1] 0.7940362
```

We can see clearly that our native accuracy is 79.40%, which is equal to the amount of loans in our data set that were fully paid.

We can also use the confusion matrix function to find out this information:

```
confusionMatrix(Model1)
Confusion Matrix and Statistics


          0      1
  0 573071      0
  1      0 148648

               Accuracy : 1
                 95% CI : (1, 1)
    No Information Rate : 0.794
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.000
            Specificity : 1.000
         Pos Pred Value : 1.000
         Neg Pred Value : 1.000
             Prevalence : 0.794
         Detection Rate : 0.794
   Detection Prevalence : 0.794
      Balanced Accuracy : 1.000

       'Positive' Class : 0
```

Of the data we have, 79.4% of the loans were “Fully Paid”. 20.6% of the loans have been “Charged Off”. Since this is a record of past data, this is a good representation of the native accuracy of the data as well.

If we were going to make a model predicting and trying to beat predictions, that is the baseline accuracy we would need to beat to be sure our model was performing better than random chance.

4.1. Splitting the data into test and training sets

```
Use caTools package to split the data into the test and train data set. Ratio i
s 80/20. 80% will be train data and 20% will be test data

set.seed(80)
split = sample.split(BIGDF$is_bad, SplitRatio = 0.80)
Train = subset(BIGDF, split == TRUE)
Test = subset(BIGDF, split == FALSE)
```

Train is the data based on the original data. Train_under will be used as this will give a clearer indication of the accuracy of the model. Train_will be used to refine the model which will be derived from the Train data set.

4.2. Choosing the Statistical Model

I'm looking to see if the data is "Fully Paid" or "Charged Off", which can also be thought of as 1 or 0. In order to do analysis on the data, I'll need to figure out which regression model I want to use. There are two options:

Linear Regression - Establish a relationship between Dependent and Independent variables Logistic Regression - Ascertain the probability of an event between 0 & 1

These two models really boil down to our data and what you're trying to look at. If you're wanting to discover a relationship between variables, toucan use a linear regression. If your data deals with binary outcomes, then you'll want a regression model. This will predict the probability of something happening based on different independent variables (which also describes the relationship between variables)

Since the dependent variable I'm interested in is binary (Did a loan get charged off, or did it get fully paid?) A Logistic regression model is what will work best here.

When doing a logistic regression, we need to make sure that any variables that are of class 'chr' are re-classified as 'factor' otherwise we will run into errors since a logistic regression model can't use variables that are character strings for statistical analysis:

```
Train$term <- as.factor(Train$term)
Train$grade <- as.factor(Train$grade)
Train$home_ownership <- as.factor(Train$home_ownership)

Test$term <- as.factor(Test$term)
Test$grade <- as.factor(Test$grade)
Test$home_ownership <- as.factor(Test$home_ownership)
```

Finally, here is the function for the logistical regression model we're going to use initially to see what the relationships are between our existing variables and the variable I added "is_bad" which is a proxy for loans that have the status "Charged Off"

```
Model1 <- glm(is_bad ~., data = Train, family = "binomial")
```

4.3. Making Sense of the Logistic Regression Model

So now I'm going to use the summary function to explore the logistical regression model

```
summary(Model1)
```

This gives us the output of the model. We are going to look through the output to see which variables are statistically significant, and what we can remove intuitively based on our knowledge. The output is below. I've made it a point to point out in the data where we'll be able to intuitively remove some variables as it's easier to follow along:

```
Call:
glm(formula = is_bad ~ . - loan_status, family = "binomial",
    data = Train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.2478  -0.7081  -0.5123  -0.2909   5.3385

Coefficients: (7 not defined because of singularities)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -3.027e+01  6.864e+00  -4.411 1.03e-05 ***
term60 months            4.674e-01  9.332e-03  50.081  < 2e-16 ***
gradeB                   2.438e+00  5.506e-02  44.283  < 2e-16 ***
gradeC                   3.395e+00  6.191e-02  54.844  < 2e-16 ***
gradeD                   4.299e+00  7.206e-02  59.659  < 2e-16 ***
gradeE                   5.140e+00  8.454e-02  60.802  < 2e-16 ***
gradeF                   5.994e+00  1.032e-01  58.097  < 2e-16 ***
gradeG                   6.235e+00  1.385e-01  45.025  < 2e-16 ***
sub_gradeA2              4.562e-01  6.116e-02   7.460 8.68e-14 ***
sub_gradeA3              6.240e-01  5.963e-02  10.464  < 2e-16 ***
sub_gradeA4              9.007e-01  5.432e-02  16.581  < 2e-16 ***
sub_gradeA5              1.223e+00  5.228e-02  23.384  < 2e-16 ***
sub_gradeB1             -9.879e-01  2.829e-02 -34.919  < 2e-16 ***
sub_gradeB2             -7.372e-01  2.545e-02 -28.966  < 2e-16 ***
sub_gradeB3             -4.676e-01  2.302e-02 -20.308  < 2e-16 ***
sub_gradeB4             -2.062e-01  2.175e-02  -9.481  < 2e-16 ***
sub_gradeB5                     NA         NA      NA       NA
sub_gradeC1             -7.604e-01  2.249e-02 -33.808  < 2e-16 ***
sub_gradeC2             -5.312e-01  2.129e-02 -24.945  < 2e-16 ***
sub_gradeC3             -3.593e-01  2.044e-02 -17.574  < 2e-16 ***
sub_gradeC4             -1.677e-01  1.969e-02  -8.517  < 2e-16 ***
sub_gradeC5                     NA         NA      NA       NA
sub_gradeD1             -6.367e-01  2.612e-02 -24.374  < 2e-16 ***
sub_gradeD2             -4.427e-01  2.576e-02 -17.185  < 2e-16 ***
sub_gradeD3             -3.367e-01  2.600e-02 -12.950  < 2e-16 ***
sub_gradeD4             -1.343e-01  2.565e-02  -5.234 1.66e-07 ***
sub_gradeD5                     NA         NA      NA       NA
sub_gradeE1             -6.690e-01  3.485e-02 -19.195  < 2e-16 ***
sub_gradeE2             -4.980e-01  3.469e-02 -14.355  < 2e-16 ***
sub_gradeE3             -3.448e-01  3.518e-02  -9.801  < 2e-16 ***
sub_gradeE4             -1.867e-01  3.583e-02  -5.211 1.88e-07 ***
sub_gradeE5                     NA         NA      NA       NA
sub_gradeF1             -7.309e-01  5.663e-02 -12.905  < 2e-16 ***
sub_gradeF2             -4.334e-01  5.864e-02  -7.391 1.46e-13 ***
sub_gradeF3             -3.706e-01  6.001e-02  -6.176 6.57e-10 ***
sub_gradeF4             -1.700e-01  6.308e-02  -2.695 0.007045 **
sub_gradeF5                     NA         NA      NA       NA
sub_gradeG1             -1.375e-01  1.123e-01  -1.225 0.220557
sub_gradeG2             -1.800e-01  1.166e-01  -1.545 0.122429
sub_gradeG3              8.608e-02  1.230e-01   0.700 0.483976
sub_gradeG4             -1.109e-01  1.318e-01  -0.842 0.400000
sub_gradeG5                     NA         NA      NA       NA
```

Seeing as these are just sub categories of Grade and we know intuitively that Grade is highly correlated (since it's determined by the customer's profile), we don't really need to keep these in.

```
Train[,"sub_grade"] <- NULL
Test[,"sub_grade"] <- NULL
BIGDF[,"sub_grade"] <- NULL
```

```
home_ownershipMORTGAGE     4.828e-01  4.353e-01   1.109 0.267435
home_ownershipNONE         8.075e-01  6.309e-01   1.280 0.200550
home_ownershipOTHER        6.860e-01  6.199e-01   1.107 0.268441
home_ownershipOWN          5.889e-01  4.354e-01   1.353 0.176212
home_ownershipRENT         7.452e-01  4.354e-01   1.712 0.086949 .
home_ownership:
```

Surprisingly, home_ownership doesn't seem to be statistically significant, so we'll go ahead and remove that as well:

```
Train[,"home_ownership"] <- NULL
Test[,"home_ownership"] <- NULL
BIGDF[,"home_ownership"] <- NULL
```

```
purposecredit_card         8.373e-02  4.182e-02   2.002 0.045277 *
purposedebt_consolidation  8.636e-02  4.124e-02   2.094 0.036268 *
purposeeducational        -5.858e+00  2.673e+01  -0.219 0.826528
purposehome_improvement    2.002e-01  4.362e-02   4.589 4.46e-06 ***
purposehouse              -1.076e-02  6.472e-02  -0.166 0.868009
purposemajor_purchase      2.089e-01  4.833e-02   4.323 1.54e-05 ***
purposemedical             2.363e-01  5.237e-02   4.512 6.41e-06 ***
purposemoving              1.710e-01  5.742e-02   2.978 0.002901 **
purposeother               7.473e-02  4.363e-02   1.713 0.086712 .
purposerenewable_energy    2.570e-01  1.324e-01   1.941 0.052273 .
purposesmall_business      5.326e-01  5.122e-02  10.397  < 2e-16 ***
purposevacation            5.574e-02  6.063e-02   0.919 0.357978
purposewedding            -2.798e-01  1.171e-01  -2.389 0.016911 *
```

It looks like there is definitely some correlation between the identified purpose of the loan, and whether or not the loan was charged off. However, since this variable is self-reported, and subject to human interpretation, as opposed to a static attribute associated with a credit profile, it is out of the scope of what I intend to investigate in this project. Investigating the relationship of this variable to propensity to charge off could be explored in a larger analysis of this data set.

```
Train[,"purpose"] <- NULL
Test[,"purpose"] <- NULL
BIGDF[,"purpose"] <- NULL
```

```
addr_stateAL              2.703e-01  7.459e-02   3.624 0.000291 ***
addr_stateAR              2.827e-01  7.839e-02   3.606 0.000311 ***
addr_stateAZ              4.207e-02  7.229e-02   0.582 0.560618
addr_stateCA             -1.649e-02  6.927e-02  -0.238 0.811821
addr_stateCO             -2.771e-01  7.323e-02  -3.784 0.000154 ***
addr_stateCT             -3.263e-02  7.533e-02  -0.433 0.664898
addr_stateDC             -3.288e-01  1.037e-01  -3.172 0.001516 **
addr_stateDE              6.443e-02  9.494e-02   0.679 0.497347
addr_stateFL              9.844e-02  6.986e-02   1.409 0.158789
addr_stateGA             -5.122e-02  7.157e-02  -0.716 0.474180
addr_stateHI             -1.855e-01  8.381e-02  -2.214 0.026838 *
addr_stateIA              1.475e+00  1.434e+00   1.028 0.303765
addr_stateID             -1.219e-01  1.588e-01  -0.768 0.442782
addr_stateIL              2.410e-02  7.108e-02   0.339 0.734586
addr_stateIN              1.067e-01  7.370e-02   1.448 0.147543
addr_stateKS             -1.954e-01  7.991e-02  -2.446 0.014457 *
addr_stateKY              1.072e-01  7.725e-02   1.388 0.165061
addr_stateLA              2.501e-01  7.527e-02   3.322 0.000893 ***
addr_stateMA              7.490e-02  7.265e-02   1.031 0.302541
addr_stateMD              1.086e-01  7.235e-02   1.501 0.133361
addr_stateME             -6.211e-01  1.565e-01  -3.968 7.23e-05 ***
addr_stateMI              6.028e-02  7.198e-02   0.837 0.402382
addr_stateMN              9.043e-02  7.334e-02   1.233 0.217573
addr_stateMO              1.529e-01  7.395e-02   2.068 0.038644 *
addr_stateMS              2.586e-01  8.474e-02   3.052 0.002275 **
addr_stateMT             -1.051e-01  9.658e-02  -1.089 0.276296
addr_stateNC              7.780e-02  7.163e-02   1.086 0.277451
addr_stateND              3.020e-02  1.382e-01   0.219 0.826989
addr_stateNE              1.139e-01  1.037e-01   1.098 0.272147
addr_stateNH             -4.268e-01  9.083e-02  -4.699 2.61e-06 ***
addr_stateNJ              1.225e-01  7.106e-02   1.724 0.084683 .
addr_stateNM              1.506e-01  8.198e-02   1.837 0.066162 .
addr_stateNV              2.035e-01  7.377e-02   2.759 0.005797 **
addr_stateNY              1.369e-01  6.970e-02   1.964 0.049480 *
addr_stateOH              1.344e-01  7.121e-02   1.887 0.059131 .
addr_stateOK              3.160e-01  7.684e-02   4.112 3.92e-05 ***
addr_stateOR             -3.094e-01  7.676e-02  -4.031 5.55e-05 ***
addr_statePA              7.147e-02  7.117e-02   1.004 0.315258
addr_stateRI             -9.000e-02  8.796e-02  -1.023 0.306194
addr_stateSC             -1.941e-01  7.696e-02  -2.522 0.011659 *
addr_stateSD              1.947e-01  1.007e-01   1.933 0.053233 .
addr_stateTN              1.385e-01  7.383e-02   1.876 0.060617 .
addr_stateTX              1.993e-02  6.976e-02   0.286 0.775097
addr_stateUT             -1.515e-01  8.012e-02  -1.892 0.058548 .
addr_stateVA              9.695e-02  7.164e-02   1.353 0.175938
addr_stateVT             -4.183e-01  1.143e-01  -3.660 0.000252 ***
addr_stateWA             -2.676e-01  7.319e-02  -3.656 0.000256 ***
addr_stateWI             -1.237e-01  7.600e-02  -1.627 0.103666
addr_stateWV             -1.267e-01  8.957e-02  -1.415 0.157213
addr_stateWY             -3.684e-02  1.031e-01  -0.357 0.720928
```

I also am going to take out “addr_state”. It would be interesting to make a map and possibly see the correlation between certain states and see if customers in certain states have higher chances of charging off their loans, however that is outside of the current scope of this project, so I’m going to remove that variable for now, as we just want to find what the best predictors of charges offs will be and visualize those data points.

```
Train[,"addr_state"] <- NULL
Test[,"addr_state"] <- NULL
BIGDF[,"addr_state"] <- NULL
```

```
loan_amnt                  1.274e-05  5.438e-07  23.432  < 2e-16 ***
int_rate                  -1.864e-01  3.821e-03 -48.778  < 2e-16 ***
annual_inc                -1.438e-06  1.224e-07 -11.740  < 2e-16 ***
dti                        2.100e-02  5.090e-04  41.253  < 2e-16 ***
delinq_2yrs                7.746e-02  3.960e-03  19.560  < 2e-16 ***
open_acc                  -3.475e-03  1.348e-03  -2.578 0.009937 **
pub_rec                    4.614e-02  1.627e-02   2.837 0.004559 **
revol_bal                 -2.184e-06  2.788e-07  -7.833 4.78e-15 ***
revol_util                 4.331e-01  1.859e-02  23.291  < 2e-16 ***
total_acc                 -9.600e-03  4.938e-04 -19.440  < 2e-16 ***
tot_coll_amt              -6.622e-08  4.251e-07  -0.156 0.876200
tot_cur_bal               -3.444e-07  7.458e-08  -4.619 3.86e-06 ***
acc_open_past_24mths       4.179e-02  1.800e-03  23.221  < 2e-16 ***
avg_cur_bal               -3.271e-06  6.549e-07  -4.995 5.88e-07 ***
mo_sin_old_rev_tl_op      -1.349e-04  4.398e-05  -3.068 0.002157 **
mo_sin_rcnt_rev_tl_op     -2.036e-03  2.807e-04  -7.252 4.10e-13 ***
mort_acc                  -2.638e-02  2.481e-03 -10.631  < 2e-16 ***
num_accts_ever_120_pd      1.064e-02  2.886e-03   3.685 0.000229 ***
num_actv_rev_tl            3.226e-02  1.641e-03  19.658  < 2e-16 ***
num_tl_op_past_12m         7.527e-03  2.878e-03   2.615 0.008921 **
pub_rec_bankruptcies      -2.081e-02  1.847e-02  -1.126 0.259974
tax_liens                 -4.212e-03  1.849e-02  -0.228 0.819807
total_bal_ex_mort         -2.001e-07  1.287e-07  -1.554 0.120104
Year                       1.344e-02  3.393e-03   3.960 7.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 566986  on 555148  degrees of freedom
Residual deviance: 508101  on 555021  degrees of freedom
  (22226 observations deleted due to missingness)
AIC: 508357

Number of Fisher Scoring iterations: 6
```

To re-cap the observations removed were:

```
sub_grade:
Train[,"sub_grade"] <- NULL

home_ownership:
Train[,"home_ownership"] <- NULL

purpose:
Train[,"purpose"] <- NULL

addr_state:
Train[,"addr_state"] <- NULL

I added the Year column in the original Data in order to make sense of it throu
ghout time, so of course it's going to be VERY correlated. Since I added it how
ever, I'm not looking to see if it's a predictor of Charge Offs:
Train[,"Year"] <- NULL

X1 is the dummy column R adds when loading data, so this one goes too:
Train[,"X1"] <- NULL
```

Additionally, I am actually incredibly surprised that these variables are NOT highly correlated to loan charge offs. Having worked in the lending industry myself, I know for a fact that these variables make someone SIGNIFIGANTLY more likely to charge off their loans. These are actually MAJOR predictors of someone charging off a loan. From my personal experience, I think that what's going on here is that the amount of loans that do have these issues (bankruptcies in particular) and still get approved is VERY small compared to our data set that has 721,719 observations. I think these are the loans that just barely manged to squeak through, while the majority of loans with bankruptcies didn't even get approved at all. A great exercise for future analysis would be to obtain, clean and compare Lending Club's data on loans that were declined as well with this data. I'm sure we'd have a much clearer picture of what's actually happening to the applications where customers do have bankruptcies (most likely declined before they ever even have a chance to get a loan):

```
tot_coll_amt              -6.622e-08  4.251e-07  -0.156 0.876200
pub_rec_bankruptcies      -2.081e-02  1.847e-02  -1.126 0.259974
tax_liens                 -4.212e-03  1.849e-02  -0.228 0.819807
total_bal_ex_mort         -2.001e-07  1.287e-07  -1.554 0.120104
```

Since I can't intuitively remove these, I am going to leave these in for now and let the model determine whether or not they'll be eliminated.

Great, now let's run the logistic regression again and see what we get:

```
Model1 <- glm(is_bad ~., data = Train, family = "binomial")
```

```
summary(Model1)
Call:
glm(formula = is_bad ~ ., family = "binomial", data = Train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.4368  -0.7109  -0.5213  -0.3134   5.2157

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)             2.023e+02  2.261e+01   8.950  < 2e-16 ***
term60 months           5.241e-01  9.126e-03  57.433  < 2e-16 ***
gradeB                  6.830e-01  1.843e-02  37.060  < 2e-16 ***
gradeC                  1.230e+00  2.348e-02  52.407  < 2e-16 ***
gradeD                  1.666e+00  3.089e-02  53.923  < 2e-16 ***
gradeE                  2.010e+00  3.870e-02  51.951  < 2e-16 ***
gradeF                  2.285e+00  4.917e-02  46.471  < 2e-16 ***
gradeG                  2.477e+00  6.180e-02  40.082  < 2e-16 ***
loan_amnt               1.269e-05  5.306e-07  23.917  < 2e-16 ***
int_rate               -4.394e-02  2.559e-03 -17.167  < 2e-16 ***
annual_inc             -1.369e-06  1.212e-07 -11.291  < 2e-16 ***
dti                     2.177e-02  5.054e-04  43.086  < 2e-16 ***
delinq_2yrs             8.679e-02  3.936e-03  22.048  < 2e-16 ***
open_acc               -4.399e-03  1.342e-03  -3.278  0.00105 **
pub_rec                 6.509e-02  1.614e-02   4.033 5.50e-05 ***
revol_bal              -2.310e-06  2.802e-07  -8.245  < 2e-16 ***
revol_util              4.609e-03  1.816e-04  25.379  < 2e-16 ***
total_acc              -1.008e-02  4.909e-04 -20.539  < 2e-16 ***
tot_coll_amt           -1.655e-08  3.824e-07  -0.043  0.96548
tot_cur_bal            -4.499e-07  7.447e-08  -6.042 1.53e-09 ***
acc_open_past_24mths    4.362e-02  1.790e-03  24.371  < 2e-16 ***
avg_cur_bal            -3.303e-06  6.562e-07  -5.034 4.80e-07 ***
mo_sin_old_rev_tl_op   -1.982e-04  4.377e-05  -4.527 5.98e-06 ***
mo_sin_rcnt_rev_tl_op  -2.233e-03  2.799e-04  -7.977 1.50e-15 ***
mort_acc               -2.701e-02  2.454e-03 -11.004  < 2e-16 ***
num_accts_ever_120_pd   1.512e-02  2.865e-03   5.276 1.32e-07 ***
num_actv_rev_tl         3.262e-02  1.630e-03  20.015  < 2e-16 ***
num_tl_op_past_12m      1.446e-02  2.859e-03   5.057 4.25e-07 ***
pub_rec_bankruptcies   -2.929e-02  1.836e-02  -1.596  0.11058
tax_liens              -1.554e-02  1.843e-02  -0.843  0.39923
total_bal_ex_mort      -5.804e-08  1.281e-07  -0.453  0.65045

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 566986  on 555148  degrees of freedom
Residual deviance: 512688  on 555111  degrees of freedom
  (22226 observations deleted due to missingness)
AIC: 512764

Number of Fisher Scoring iterations: 5
```

That looks much better.

4.4. Using the stepAIC function on our training data

Now that we've run a Logistical Regression and learned more about the data and how it's correlated, it's time to make sure that we are left with only the variables that are significant. One way to do this is an AIC model, which will test the variables in relation to "is_bad" which is our proxy for Charge Offs, so that we are only left with the ACTUAL predictors of loan Charge offs)

```
Aic_Model <- stepAIC(Model1,direction="both")
```

The AIC model has been taking quite a while and looking at the data again, I can see why. It looks like I have missed some NA values in some of the variables:

```
dti <dbl> - 24 NAs
tot_coll_amt <int> -22202
tot_cur_bal <int> -22202
acc_open_past_24mths <int> - 6016
avg_cur_bal <int> - 22202
mo_sin_old_rev_tl_op <int> - 22202
mo_sin_rcnt_rev_tl_op <int>- 22202
mort_acc  <int> - 6016
num_accts_ever_120_pd <int> - 22202
num_actv_rev_tl <int> - 22202
num_tl_op_past_12m <int> - 22202
total_bal_ex_mort <int> - 6016
```

There are a couple of ways to go about dealing with these leftover values. One is to replace them with the average values of each column, meaning they won't change the data at all. The other is simply to omit them. This is what I wanted to do in the first place, and honestly this data set is so large that we could drop all of those values and still be fine. I also suspect that since the numbers are all either 22202 or 6016, that these are the same observations in each of the variables so I'm going to go ahead and drop those to also help the processing of the Step AIC go faster.

```
TRAIN <- na.omit(Train)
TRAIN[,"X1"] <- NULL
BIGDF <- na.omit(BIGDF)
TEST <- na.omit(Test)
```

4.5. Subsetting the data for faster analysis

In order to help speed up processing time, I'm going to use a much smaller sample set. Let's use this to run regression and the StepAIC to figure out which variables are significant:

```
TRAINsample <- sample_frac(TRAIN, 0.1)
Model1 <- glm(is_bad ~., data = TRAINsample, family = "binomial")
Aic_Model <- stepAIC(Model1,direction="both")
```

This is the output of the AIC for us to inspect:

```
Start:  AIC=51310.41
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + open_acc + pub_rec + revol_bal +
    revol_util + total_acc + tot_coll_amt + tot_cur_bal + acc_open_past_24mths
+
    avg_cur_bal + mo_sin_old_rev_tl_op + mo_sin_rcnt_rev_tl_op +
    mort_acc + num_accts_ever_120_pd + num_actv_rev_tl + num_tl_op_past_12m +
    pub_rec_bankruptcies + tax_liens + total_bal_ex_mort

                        Df Deviance   AIC
- pub_rec                1    51238 51308
- pub_rec_bankruptcies   1    51239 51309
- avg_cur_bal            1    51239 51309
- tax_liens              1    51240 51310
- open_acc               1    51240 51310
- tot_coll_amt           1    51240 51310
- revol_bal              1    51240 51310
<none>                       51238 51310
- tot_cur_bal            1    51241 51311
- num_tl_op_past_12m     1    51241 51311
- total_bal_ex_mort      1    51243 51313
- num_accts_ever_120_pd  1    51243 51313
- mo_sin_rcnt_rev_tl_op  1    51246 51316
- annual_inc             1    51247 51317
- mo_sin_old_rev_tl_op   1    51251 51321
- mort_acc               1    51263 51333
- total_acc              1    51265 51335
- int_rate               1    51278 51348
- num_actv_rev_tl        1    51282 51352
- delinq_2yrs            1    51292 51362
- revol_util             1    51294 51364
- acc_open_past_24mths   1    51295 51365
- home_ownership         5    51316 51378
- loan_amnt              1    51310 51380
- dti                    1    51440 51510
- term                   1    51554 51624
- grade                  6    51581 51641

Step:  AIC=51308.43
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util +
    total_acc + tot_coll_amt + tot_cur_bal + acc_open_past_24mths +
    avg_cur_bal + mo_sin_old_rev_tl_op + mo_sin_rcnt_rev_tl_op +
    mort_acc + num_accts_ever_120_pd + num_actv_rev_tl + num_tl_op_past_12m +
    pub_rec_bankruptcies + tax_liens + total_bal_ex_mort

                        Df Deviance   AIC
- pub_rec_bankruptcies   1    51239 51307
- avg_cur_bal            1    51239 51307
- open_acc               1    51240 51308
- tot_coll_amt           1    51240 51308
- revol_bal              1    51240 51308
<none>                       51238 51308
```

```
- tot_cur_bal            1    51241 51309
- num_tl_op_past_12m     1    51241 51309
+ pub_rec                1    51238 51310
- total_bal_ex_mort      1    51243 51311
- num_accts_ever_120_pd  1    51243 51311
- tax_liens              1    51245 51313
- mo_sin_rcnt_rev_tl_op  1    51246 51314
- annual_inc             1    51247 51315
- mo_sin_old_rev_tl_op   1    51251 51319
- mort_acc               1    51263 51331
- total_acc              1    51265 51333
- int_rate               1    51278 51346
- num_actv_rev_tl        1    51282 51350
- delinq_2yrs            1    51292 51360
- revol_util             1    51294 51362
- acc_open_past_24mths   1    51295 51363
- home_ownership         5    51316 51376
- loan_amnt              1    51310 51378
- dti                    1    51440 51508
- term                   1    51554 51622
- grade                  6    51581 51639

Step:  AIC=51307.29
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util +
    total_acc + tot_coll_amt + tot_cur_bal + acc_open_past_24mths +
    avg_cur_bal + mo_sin_old_rev_tl_op + mo_sin_rcnt_rev_tl_op +
    mort_acc + num_accts_ever_120_pd + num_actv_rev_tl + num_tl_op_past_12m +
    tax_liens + total_bal_ex_mort

                       Df Deviance   AIC
- avg_cur_bal            1    51240 51306
- open_acc               1    51241 51307
- tot_coll_amt           1    51241 51307
- revol_bal              1    51241 51307
<none>                        51239 51307
- tot_cur_bal            1    51242 51308
- num_tl_op_past_12m     1    51242 51308
+ pub_rec_bankruptcies   1    51238 51308
+ pub_rec                1    51239 51309
- num_accts_ever_120_pd  1    51244 51310
- total_bal_ex_mort      1    51244 51310
- tax_liens              1    51246 51312
- mo_sin_rcnt_rev_tl_op  1    51247 51313
- annual_inc             1    51248 51314
- mo_sin_old_rev_tl_op   1    51251 51317
- mort_acc               1    51263 51329
- total_acc              1    51266 51332
- int_rate               1    51278 51344
- num_actv_rev_tl        1    51283 51349
- delinq_2yrs            1    51292 51358
- revol_util             1    51294 51360
- acc_open_past_24mths   1    51297 51363
- home_ownership         5    51317 51375
```

```
- loan_amnt              1    51310 51376
- dti                    1    51440 51506
- term                   1    51555 51621
- grade                  6    51584 51640

Step:  AIC=51306.2
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util +
    total_acc + tot_coll_amt + tot_cur_bal + acc_open_past_24mths +
    mo_sin_old_rev_tl_op + mo_sin_rcnt_rev_tl_op + mort_acc +
    num_accts_ever_120_pd + num_actv_rev_tl + num_tl_op_past_12m +
    tax_liens + total_bal_ex_mort

                       Df Deviance   AIC
- open_acc              1    51241 51305
- tot_coll_amt          1    51242 51306
- revol_bal             1    51242 51306
<none>                       51240 51306
- num_tl_op_past_12m    1    51243 51307
+ avg_cur_bal           1    51239 51307
+ pub_rec_bankruptcies  1    51239 51307
+ pub_rec               1    51239 51307
- total_bal_ex_mort     1    51244 51308
- num_accts_ever_120_pd 1    51245 51309
- tax_liens             1    51246 51310
- annual_inc            1    51249 51313
- mo_sin_rcnt_rev_tl_op 1    51249 51313
- mo_sin_old_rev_tl_op  1    51252 51316
- tot_cur_bal           1    51258 51322
- mort_acc              1    51264 51328
- total_acc             1    51267 51331
- int_rate              1    51279 51343
- num_actv_rev_tl       1    51285 51349
- delinq_2yrs           1    51293 51357
- revol_util            1    51294 51358
- acc_open_past_24mths  1    51298 51362
- loan_amnt             1    51310 51374
- home_ownership        5    51318 51374
- dti                   1    51442 51506
- term                  1    51555 51619
- grade                 6    51585 51639

Step:  AIC=51305.17
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_bal + revol_util +
    total_acc + tot_coll_amt + tot_cur_bal + acc_open_past_24mths +
    mo_sin_old_rev_tl_op + mo_sin_rcnt_rev_tl_op + mort_acc +
    num_accts_ever_120_pd + num_actv_rev_tl + num_tl_op_past_12m +
    tax_liens + total_bal_ex_mort

                       Df Deviance   AIC
- tot_coll_amt          1    51243 51305
- revol_bal             1    51243 51305
<none>                       51241 51305
```

```
- num_tl_op_past_12m     1    51244 51306
+ pub_rec_bankruptcies   1    51240 51306
+ open_acc               1    51240 51306
+ pub_rec                1    51240 51306
+ avg_cur_bal            1    51241 51307
- num_accts_ever_120_pd  1    51246 51308
- total_bal_ex_mort      1    51246 51308
- tax_liens              1    51247 51309
- annual_inc             1    51249 51311
- mo_sin_rcnt_rev_tl_op  1    51250 51312
- mo_sin_old_rev_tl_op   1    51253 51315
- tot_cur_bal            1    51260 51322
- mort_acc               1    51264 51326
- int_rate               1    51280 51342
- total_acc              1    51281 51343
- delinq_2yrs            1    51293 51355
- acc_open_past_24mths   1    51298 51360
- num_actv_rev_tl        1    51299 51361
- revol_util             1    51302 51364
- loan_amnt              1    51311 51373
- home_ownership         5    51319 51373
- dti                    1    51442 51504
- term                   1    51556 51618
- grade                  6    51586 51638

Step:  AIC=51304.9
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_bal + revol_util +
    total_acc + tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + num_tl_op_past_12m + tax_liens + total_bal_ex_mort

                        Df Deviance   AIC
- revol_bal              1    51245 51305
<none>                        51243 51305
+ tot_coll_amt           1    51241 51305
- num_tl_op_past_12m     1    51245 51305
+ pub_rec_bankruptcies   1    51242 51306
+ open_acc               1    51242 51306
+ pub_rec                1    51242 51306
+ avg_cur_bal            1    51243 51307
- total_bal_ex_mort      1    51248 51308
- num_accts_ever_120_pd  1    51249 51309
- tax_liens              1    51249 51309
- annual_inc             1    51251 51311
- mo_sin_rcnt_rev_tl_op  1    51252 51312
- mo_sin_old_rev_tl_op   1    51254 51314
- tot_cur_bal            1    51262 51322
- mort_acc               1    51266 51326
- int_rate               1    51282 51342
- total_acc              1    51283 51343
- delinq_2yrs            1    51294 51354
- acc_open_past_24mths   1    51300 51360
- num_actv_rev_tl        1    51301 51361
```

```
- revol_util             1    51303 51363
- loan_amnt              1    51312 51372
- home_ownership         5    51321 51373
- dti                    1    51444 51504
- term                   1    51557 51617
- grade                  6    51589 51639

Step:  AIC=51304.77
is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_util + total_acc +
    tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + num_tl_op_past_12m + tax_liens + total_bal_ex_mort

                        Df Deviance   AIC
<none>                       51245 51305
+ revol_bal              1    51243 51305
+ tot_coll_amt           1    51243 51305
- num_tl_op_past_12m     1    51247 51305
+ pub_rec_bankruptcies   1    51244 51306
+ pub_rec                1    51244 51306
+ open_acc               1    51244 51306
+ avg_cur_bal            1    51244 51306
- num_accts_ever_120_pd  1    51251 51309
- tax_liens              1    51251 51309
- total_bal_ex_mort      1    51252 51310
- mo_sin_rcnt_rev_tl_op  1    51253 51311
- annual_inc             1    51254 51312
- mo_sin_old_rev_tl_op   1    51258 51316
- tot_cur_bal            1    51265 51323
- mort_acc               1    51268 51326
- int_rate               1    51283 51341
- total_acc              1    51284 51342
- delinq_2yrs            1    51297 51355
- num_actv_rev_tl        1    51301 51359
- revol_util             1    51303 51361
- acc_open_past_24mths   1    51303 51361
- loan_amnt              1    51312 51370
- home_ownership         5    51322 51372
- dti                    1    51444 51502
- term                   1    51559 51617
- grade                  6    51592 51640
```

4.6 Results of the stepAIC

We can tell that the Step AIC Machine Learning Model is working, because every time the model runs another iteration, the result of the line:

Step: AIC=

becomes smaller. This is what we want. If we think about the mathematics of what's actually going on By the end of the AIC, we see that 20 variables are left that are statistically significant:

```
term
grade
home_ownership
loan_amnt
int_rate
annual_inc
dti
delinq_2yrs
revol_util
total_acc
tot_cur_bal
acc_open_past_24mths
mo_sin_old_rev_tl_op
mo_sin_rcnt_rev_tl_op
mort_acc
num_accts_ever_120_pd
num_actv_rev_tl
num_tl_op_past_12m
tax_liens
total_bal_ex_mort
```

4.7. Now let's inspect the final product

```
MODEL3 <- glm(is_bad ~ term + grade + home_ownership + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_util + total_acc +
    tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + num_tl_op_past_12m + tax_liens + total_bal_ex_mort, data
= TRAINsample, family = "binomial")
```

```
summary(MODEL3)
Call:
glm(formula = is_bad ~ term + grade + home_ownership + loan_amnt +
    int_rate + annual_inc + dti + delinq_2yrs + revol_util +
    total_acc + tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + num_tl_op_past_12m + tax_liens + total_bal_ex_mort,
    family = "binomial", data = TRAINsample)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9170  -0.7101  -0.5214  -0.3217   2.7759

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)           -2.958e+00  1.091e+00  -2.711 0.006702 **
term60 months          5.127e-01  2.881e-02  17.792  < 2e-16 ***
gradeB                 6.525e-01  5.754e-02  11.341  < 2e-16 ***
gradeC                 1.252e+00  7.309e-02  17.127  < 2e-16 ***
gradeD                 1.662e+00  9.653e-02  17.216  < 2e-16 ***
gradeE                 2.035e+00  1.207e-01  16.863  < 2e-16 ***
gradeF                 2.414e+00  1.538e-01  15.696  < 2e-16 ***
gradeG                 2.763e+00  1.959e-01  14.107  < 2e-16 ***
loan_amnt              1.353e-05  1.644e-06   8.227  < 2e-16 ***
int_rate              -4.965e-02  8.002e-03  -6.205 5.46e-10 ***
annual_inc            -1.096e-06  3.747e-07  -2.925 0.003439 **
dti                    2.255e-02  1.599e-03  14.105  < 2e-16 ***
delinq_2yrs            9.279e-02  1.255e-02   7.395 1.42e-13 ***
revol_util             4.124e-03  5.432e-04   7.592 3.15e-14 ***
total_acc             -8.410e-03  1.357e-03  -6.196 5.81e-10 ***
tot_cur_bal           -5.850e-07  1.302e-07  -4.494 7.00e-06 ***
acc_open_past_24mths   4.243e-02  5.516e-03   7.692 1.45e-14 ***
mo_sin_old_rev_tl_op  -4.879e-04  1.370e-04  -3.561 0.000369 ***
mo_sin_rcnt_rev_tl_op -2.556e-03  8.801e-04  -2.905 0.003678 **
mort_acc              -3.733e-02  7.791e-03  -4.792 1.65e-06 ***
num_accts_ever_120_pd  2.237e-02  8.754e-03   2.555 0.010619 *
num_actv_rev_tl        2.961e-02  3.918e-03   7.557 4.12e-14 ***
num_tl_op_past_12m     1.408e-02  8.950e-03   1.573 0.115803
tax_liens              7.651e-02  2.882e-02   2.654 0.007950 **
total_bal_ex_mort     -1.033e-06  3.877e-07  -2.664 0.007725 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 56570  on 55514  degrees of freedom
Residual deviance: 51245  on 55485  degrees of freedom
AIC: 51305

Number of Fisher Scoring iterations: 5
```

4.8. Tightening up the model

We see that we can still get rid of one more variable:

```
TRAINsample[,"num_tl_op_past_12m"] <- NULL
```

```
MODEL4 <- glm(is_bad ~ term + grade + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_util + total_acc +
    tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + tax_liens + total_bal_ex_mort, data = TRAINsample, family
= "binomial")
```

```
summary(MODEL4)
Call:
glm(formula = is_bad ~ term + grade + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_util + total_acc +
    tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + tax_liens + total_bal_ex_mort, family = "binomial",
    data = TRAINsample)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8757  -0.7110  -0.5234  -0.3242   2.8177

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)          -2.604e+00  8.800e-02 -29.587  < 2e-16 ***
term60 months         4.919e-01  2.863e-02  17.182  < 2e-16 ***
gradeB                6.605e-01  5.749e-02  11.488  < 2e-16 ***
gradeC                1.267e+00  7.296e-02  17.367  < 2e-16 ***
gradeD                1.682e+00  9.634e-02  17.457  < 2e-16 ***
gradeE                2.058e+00  1.205e-01  17.087  < 2e-16 ***
gradeF                2.441e+00  1.535e-01  15.905  < 2e-16 ***
gradeG                2.797e+00  1.955e-01  14.308  < 2e-16 ***
loan_amnt             1.287e-05  1.636e-06   7.867 3.63e-15 ***
int_rate             -4.901e-02  7.991e-03  -6.134 8.59e-10 ***
annual_inc           -8.165e-07  3.692e-07  -2.211 0.027014 *
dti                   2.208e-02  1.595e-03  13.845  < 2e-16 ***
delinq_2yrs           8.962e-02  1.252e-02   7.155 8.35e-13 ***
revol_util            3.972e-03  5.383e-04   7.377 1.61e-13 ***
total_acc            -8.329e-03  1.355e-03  -6.145 8.00e-10 ***
tot_cur_bal          -1.032e-06  1.217e-07  -8.483  < 2e-16 ***
acc_open_past_24mths  4.623e-02  4.389e-03  10.533  < 2e-16 ***
mo_sin_old_rev_tl_op -5.192e-04  1.362e-04  -3.811 0.000138 ***
mo_sin_rcnt_rev_tl_op -2.782e-03  8.691e-04  -3.201 0.001367 **
mort_acc             -5.310e-02  7.644e-03  -6.947 3.74e-12 ***
num_accts_ever_120_pd  2.460e-02  8.721e-03   2.821 0.004786 **
num_actv_rev_tl        2.900e-02  3.913e-03   7.412 1.25e-13 ***
tax_liens              7.204e-02  2.874e-02   2.506 0.012199 *
total_bal_ex_mort     -5.549e-07  3.842e-07  -1.444 0.148679
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 56570  on 55514  degrees of freedom
Residual deviance: 51324  on 55491  degrees of freedom
AIC: 51372

Number of Fisher Scoring iterations: 5
```

And still one more:

```
TRAINsample[,"total_bal_ex_mort"] <- NULL
MODEL5 <- glm(is_bad ~ term + grade + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_util + total_acc +
    tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + tax_liens, data = TRAINsample, family = "binomial")
```

```
summary(MODEL5)

Call:
glm(formula = is_bad ~ term + grade + loan_amnt + int_rate +
    annual_inc + dti + delinq_2yrs + revol_util + total_acc +
    tot_cur_bal + acc_open_past_24mths + mo_sin_old_rev_tl_op +
    mo_sin_rcnt_rev_tl_op + mort_acc + num_accts_ever_120_pd +
    num_actv_rev_tl + tax_liens, family = "binomial", data = TRAINsample)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8706  -0.7116  -0.5236  -0.3243   2.8242

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)          -2.585e+00  8.706e-02 -29.694  < 2e-16 ***
term60 months         4.922e-01  2.863e-02  17.190  < 2e-16 ***
gradeB                6.599e-01  5.749e-02  11.479  < 2e-16 ***
gradeC                1.266e+00  7.296e-02  17.356  < 2e-16 ***
gradeD                1.681e+00  9.634e-02  17.446  < 2e-16 ***
gradeE                2.057e+00  1.204e-01  17.075  < 2e-16 ***
gradeF                2.438e+00  1.535e-01  15.891  < 2e-16 ***
gradeG                2.793e+00  1.955e-01  14.291  < 2e-16 ***
loan_amnt             1.277e-05  1.635e-06   7.810 5.73e-15 ***
int_rate             -4.890e-02  7.990e-03  -6.120 9.34e-10 ***
annual_inc           -9.792e-07  3.539e-07  -2.767 0.005664 **
dti                   2.123e-02  1.482e-03  14.326  < 2e-16 ***
delinq_2yrs           9.018e-02  1.252e-02   7.205 5.80e-13 ***
revol_util            3.907e-03  5.364e-04   7.283 3.27e-13 ***
total_acc            -8.999e-03  1.274e-03  -7.064 1.62e-12 ***
tot_cur_bal          -1.101e-06  1.126e-07  -9.776  < 2e-16 ***
acc_open_past_24mths  4.631e-02  4.389e-03  10.552  < 2e-16 ***
mo_sin_old_rev_tl_op -5.082e-04  1.360e-04  -3.737 0.000186 ***
mo_sin_rcnt_rev_tl_op -2.807e-03  8.689e-04  -3.230 0.001237 **
mort_acc             -5.005e-02  7.343e-03  -6.817 9.31e-12 ***
num_accts_ever_120_pd  2.472e-02  8.719e-03   2.835 0.004587 **
num_actv_rev_tl        2.991e-02  3.863e-03   7.741 9.84e-15 ***
tax_liens              7.229e-02  2.875e-02   2.514 0.011932 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 56570  on 55514  degrees of freedom
Residual deviance: 51326  on 55492  degrees of freedom
AIC: 51372

Number of Fisher Scoring iterations: 5
```

Great! we’re finally left with a model where every value is significant:

I am also going to remove the variables for months since oldest and most recent trade line opened. The only time series data I want to use is the Year column I added to the data frame.

```
BIGDF[,"mo_sin_old_rev_tl_op"] <- NULL
BIGDF[,"mo_sin_rcnt_rev_tl_op"] <- NULL
```

4.9: The Bottom Line

These are the 16 Variables that have the most correlation to Charge Offs, and ultimately the ones we want to explore visually. I'm also including the Year variable that I added so that we can understand how some of these are affected by time (in years) for a total of 16 variables with strong correlations that we'll visualize.

```
1) Year
2) term
3) grade
4) loan_amnt
5) int_rate
6) annual_inc
7) dti
8) delinq_2yrs
9) revol_util
10) total_acc
11) tot_cur_bal
12) acc_open_past_24mths
13) mort_acc
14) num_accts_ever_120_pd
15) num_actv_rev_tl
16) tax_liens
```

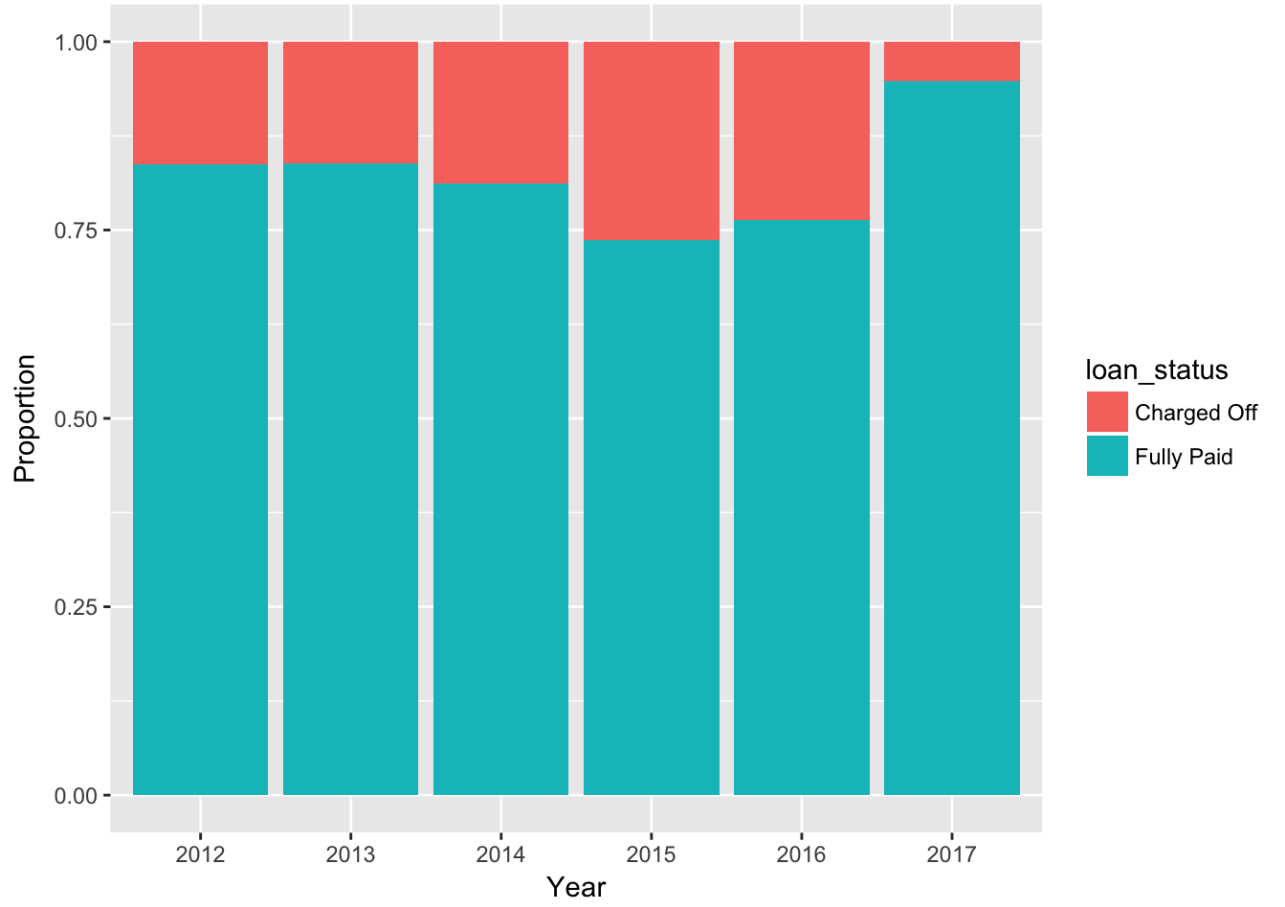
5. Visualizing the data

Great! So we now have our data completely cleaned up, removing any extra symbols, NA values, blanks, and removed all of the variables we didn't need. We've also loaded the data into one large workable frame and several smaller ones (Including subsets) for statistical analysis. For the relationships between variables, I would like to use graphs to visualize the data as we can get a clearer picture of what the relationships between variables are visually.

5.1. First let's visualize Total Paid Vs. Charged Off Loans over the last 5 years

```
## [1] "From the data, it looks like Lending Club started off pretty decently a
nd in 2014, 2015 and 2016 those years were a little rougher. In 2017, the portf
olio improved drastically, so there could be a couple of explanations."
```

```
plotyear <- ggplot(BIGDF, aes(x = as.factor(Year), fill = loan_status)) + geom_
bar(position = "fill") + labs(x = "Year", y = "Proportion" )
```




```
##
##                2012   2013   2014   2015   2016   2017
##   Charged Off   8609  20342  37203  53418  27766   1310
##   Fully Paid   44268 105780 159807 149623  89858  23735
```

From the data, it looks like Lending Club started off pretty decently and in 2014, 2015 and 2016 those years were a little rougher. In 2017, the portfolio improved drastically, so there could be a couple of explanations.

1. There could have been an issue in the data, either in how I cleaned it, or maybe something else happened along the way, possibly something I overlooked?
2. Lending club may have tightened up their lending parameters a couple years ago when they noticed a dip in their portfolio
3. There may be some foul play and the data downloaded from Lending Club might not actually represent their portfolio or (most likely since their former CEO was fired on May 9th, 2016 after it was discovered that he was dealing in loans that did not meet Lending Club's criteria).
4. There could have been an issue in the way the data was uploaded on Lending Club's side.
5. The values for 2017's Charge Offs and Fully Paid loans are both quite small. It could also be that the majority of loans in that year's data set were still currently being paid on, and removing them left me with smaller amounts of loans in each bucket.

For the purposes of this project, I'm going to go with the first explanation and assume that the data they've provided is correct, however this is a perfect example of real life issues and disparities in how data is recorded.

5.2. Plot of the Loan Status vs. Term Length

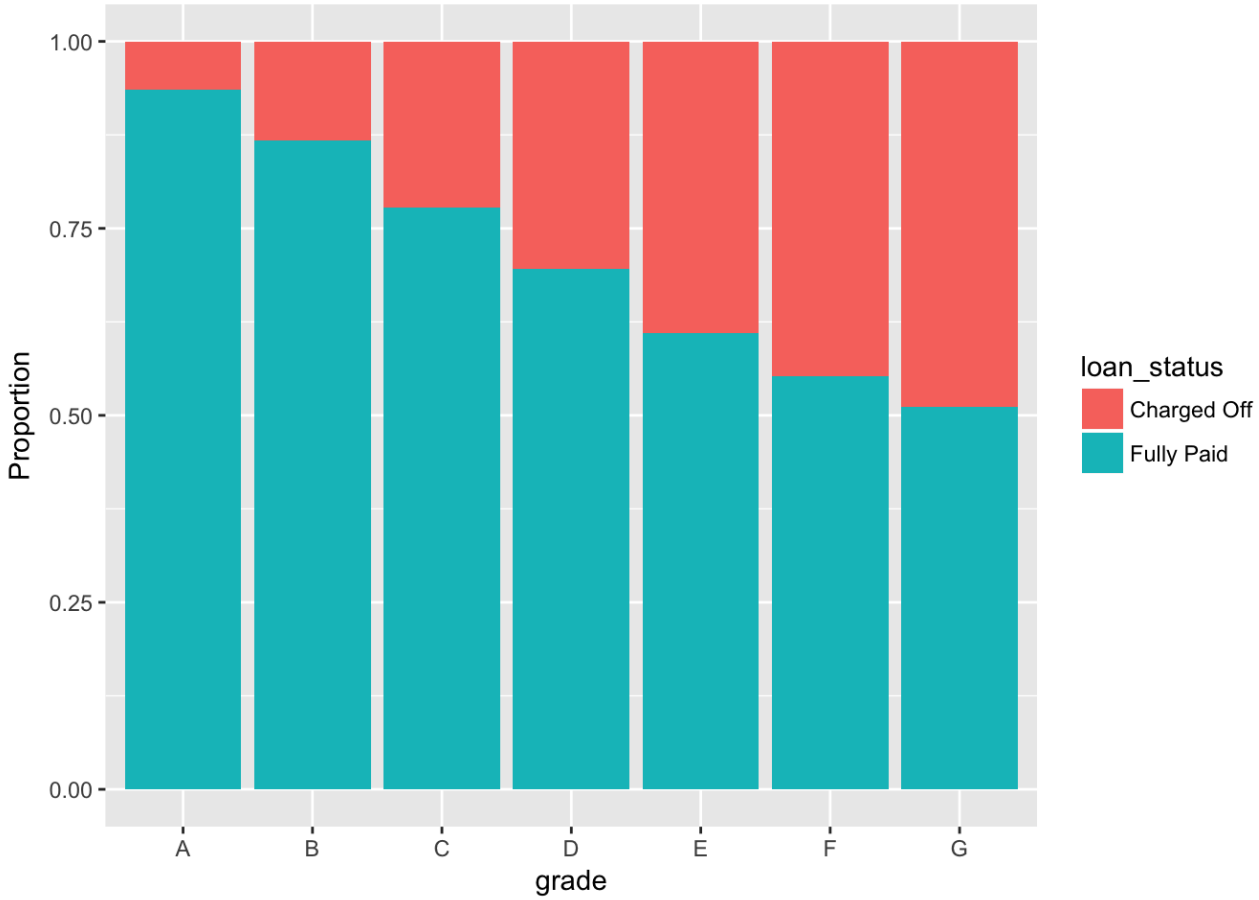
```
plotTerm <- ggplot(BIGDF, aes(x = as.factor(term), fill = loan_status)) + geom_
bar(position = "fill") + labs(x = "Term", y = "Proportion" )
grid.arrange(plotTerm, ncol = 2)
```


```
##
##                36 months 60 months
##   Charged Off      90897     57751
##   Fully Paid      458095    114976
```

We can very clearly see that a longer loan term has a much higher chance of charging off than do the shorter terms. Most likely due to the fact that the longer that any loan has to be paid back, the more exposure the money has since that's more time where something may affect the borrower's ability to pay the loan back.

5.3. Plot of Loan Status vs Grade

```
plotgrade <- ggplot(BIGDF, aes(x = as.factor(grade), fill = loan_status)) + geo
m_bar(position = "fill") + labs(x = "grade", y = "Proportion" )
grid.arrange(plotgrade, ncol = 1)
```

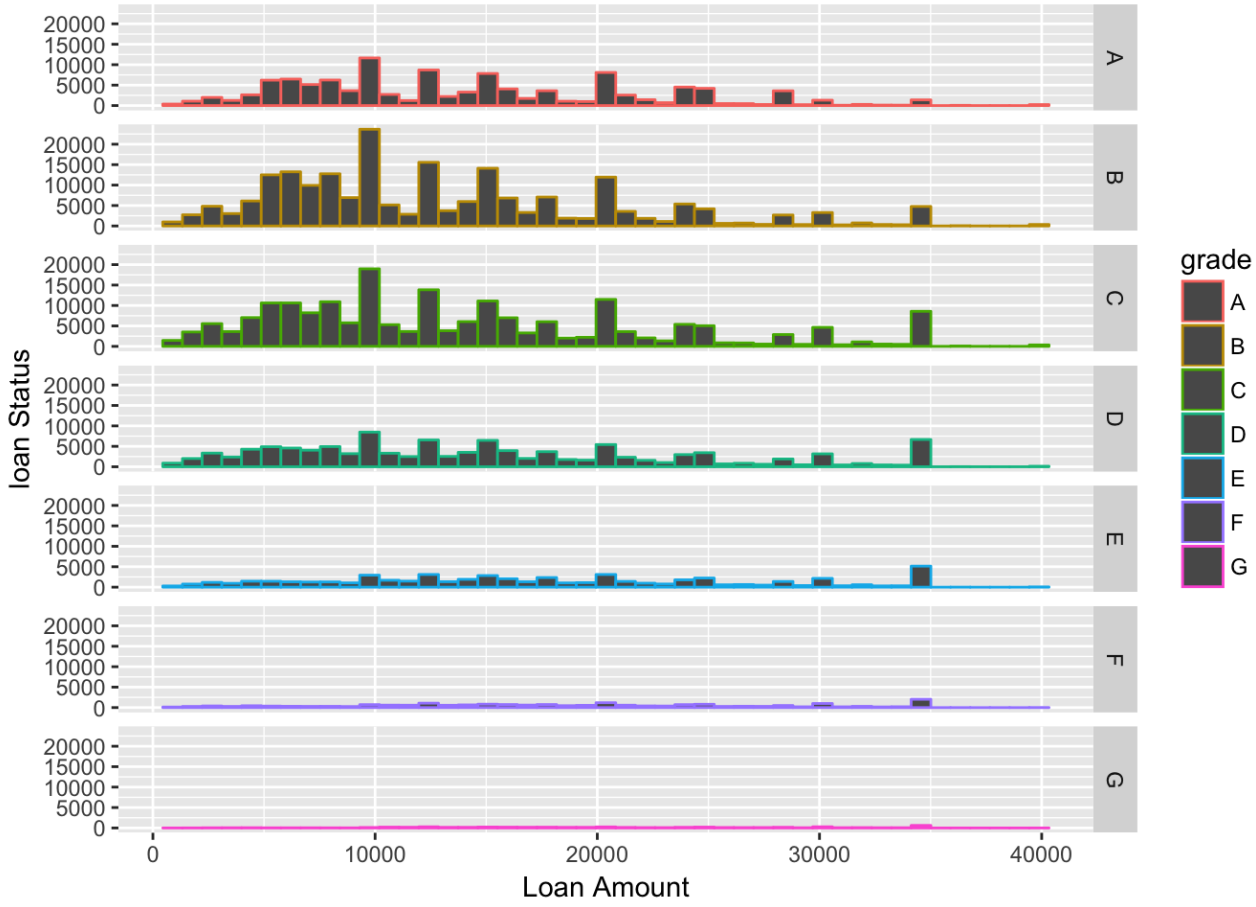



```
##
##                     A      B      C      D      E      F      G
##   Charged Off    7431  27603  44749  34761  22196   9300   2608
##   Fully Paid   106976 180583 156859  79760  34719  11446   2728
```

We can quite clearly see here that the worse grade the loan receives, the higher the chance that they will charging off. The grade is assigned to a loan by Lending Club according to a customer's credit, so we can clearly see, that those with better credit wind up paying their loans back, and those with worse credit wind up charging off their loans a higher percentage of the time.

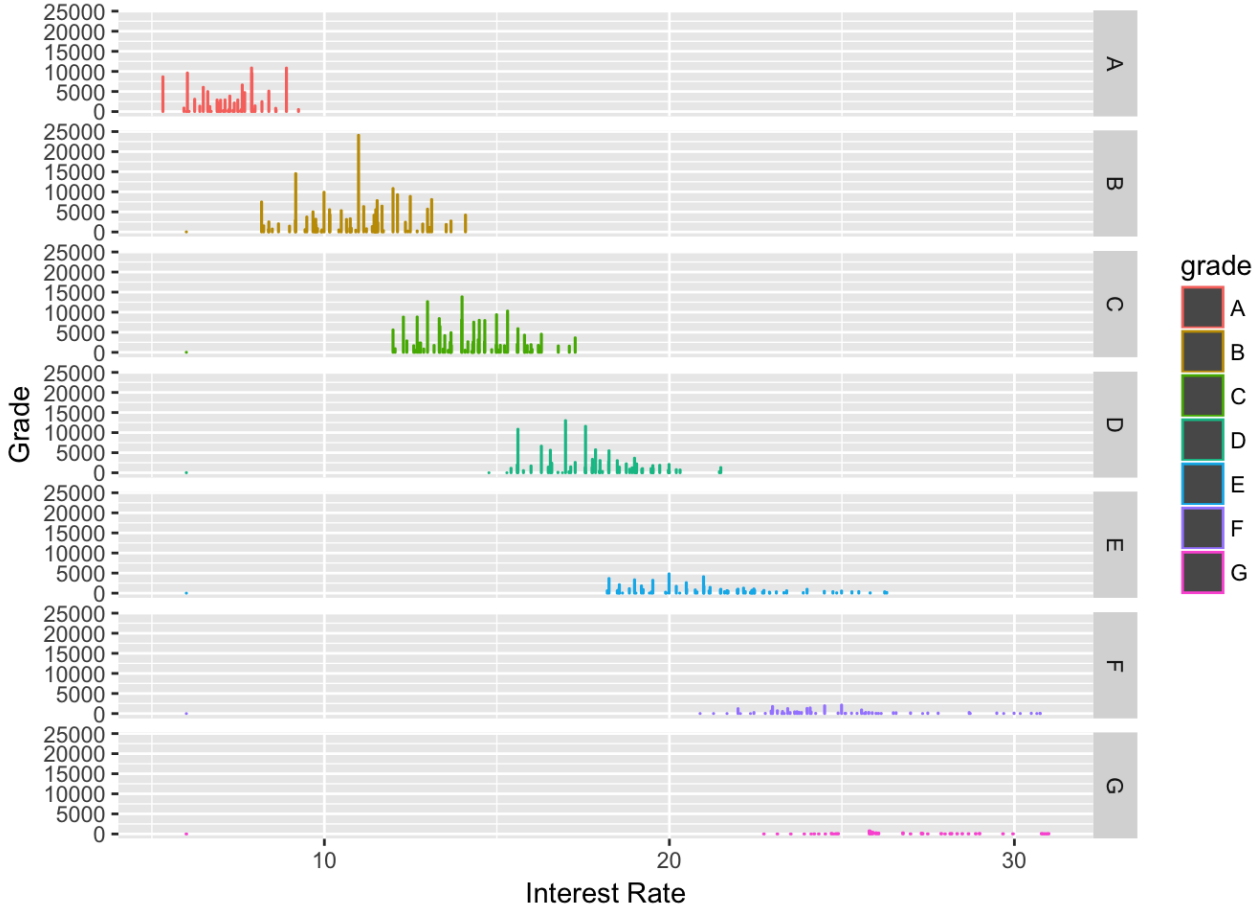
5.4. Plot of Loan Status vs. Loan Amount

```
ggplot(data = BIGDF, aes(loan_amnt, col = grade)) + geom_histogram(bins = 45) +
 xlab('Loan Amount') + ylab('loan Status') + facet_grid(grade~.)
```



5.5. Plot of the Interest Rate vs. Grade

```
ggplot(BIGDF, aes(int_rate, col = grade)) + geom_bar() + xlab('Interest Rate')
 + ylab('Grade') + facet_grid(grade~.)
```

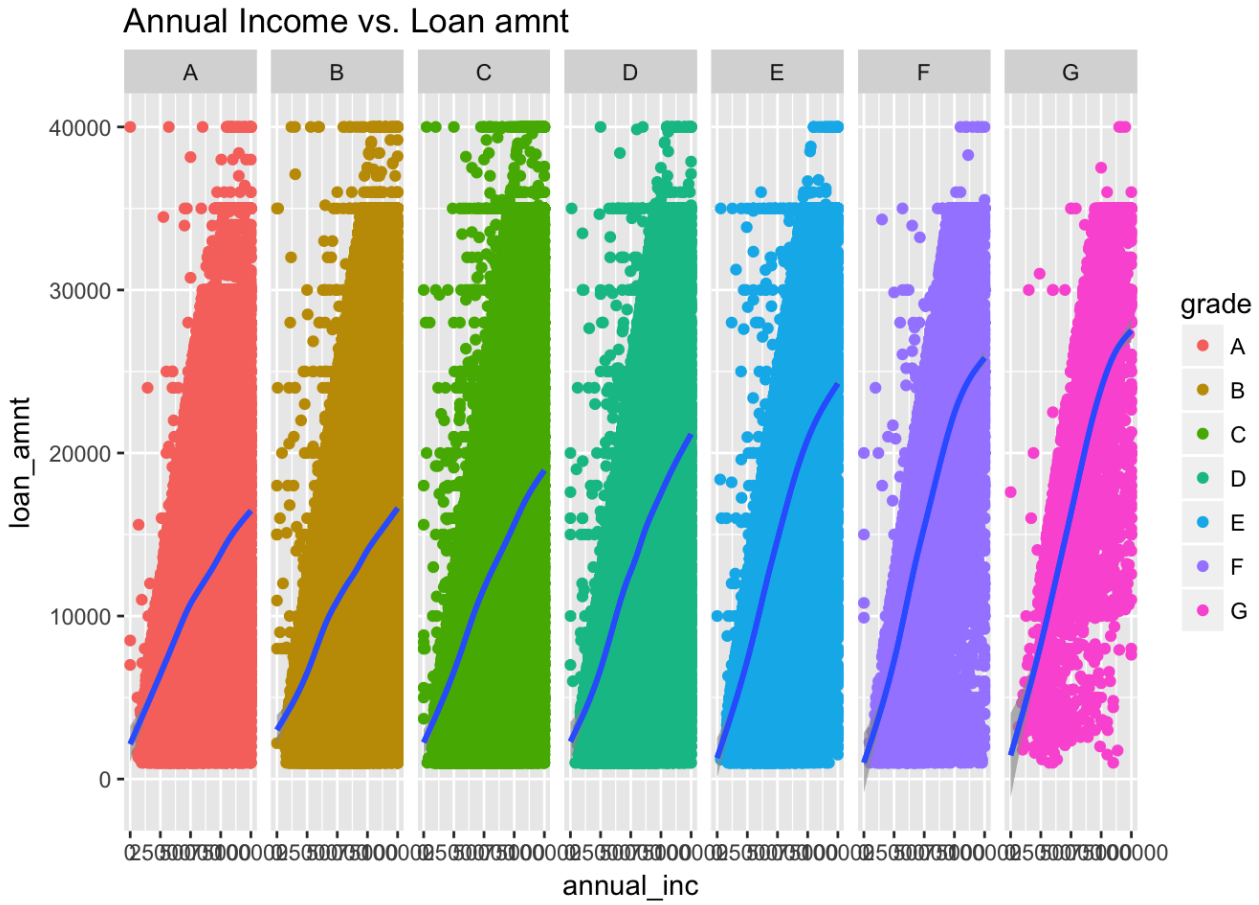


From this graph, we can see that the better the grade (and correspondingly the better the credit of the borrower), the lower the interest rate. Interestingly, we see that the highest volume of loans are in the B & C grade categories, with interest rates lower than 20%.

5.6. Plot of Annual Income vs. Loan amount by Grade

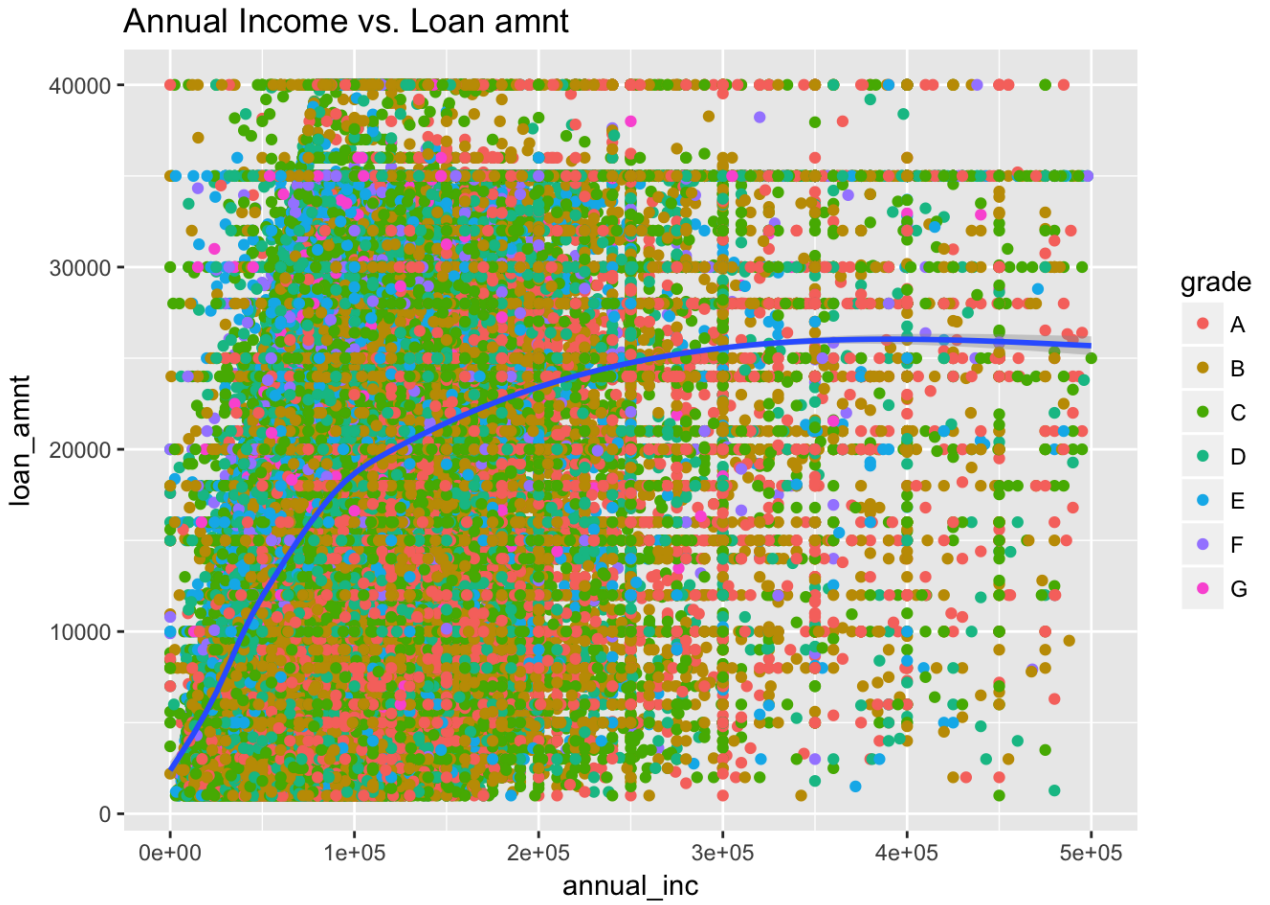
```
mydata <- filter(BIGDF, annual_inc < 500000)
p <- ggplot(mydata, aes(annual_inc, loan_amnt)) +
  geom_point(aes(colour = grade)) +
  labs(title = 'Annual Income vs. Loan amnt') +
  geom_smooth()
p + xlim(0,100000) + facet_grid(. ~ grade) + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam'
## `geom_smooth()` using method = 'gam'
```




```
p
```

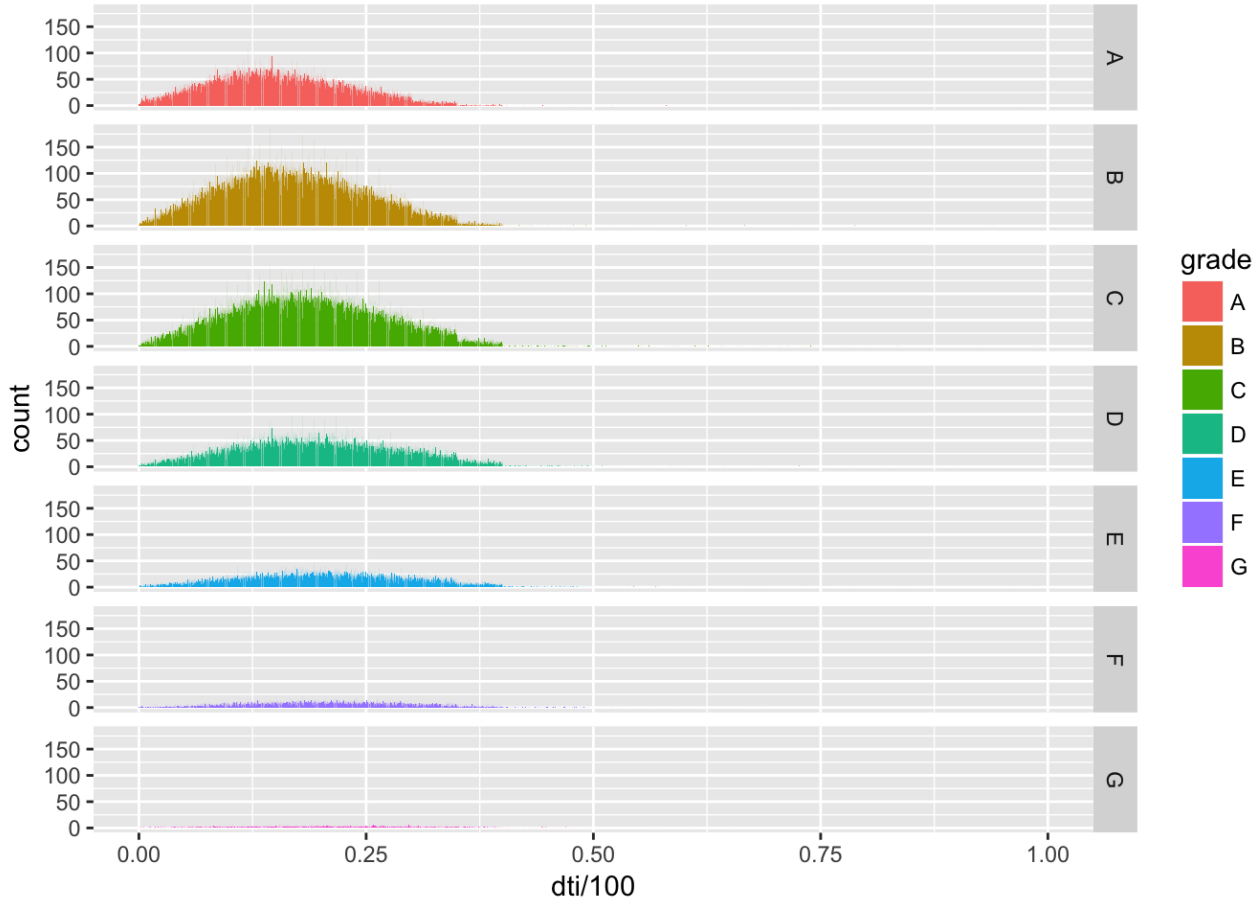
```
## `geom_smooth()` using method = 'gam'
```



This came out great! And we can see from the data that it confirms what common sense tells us. The steepness of the regression line shows us that the more money borrowed relative to income, the lower the grade is, due to the risk increasing. We also see that the line levels off once a customer's income is above a certain threshold.

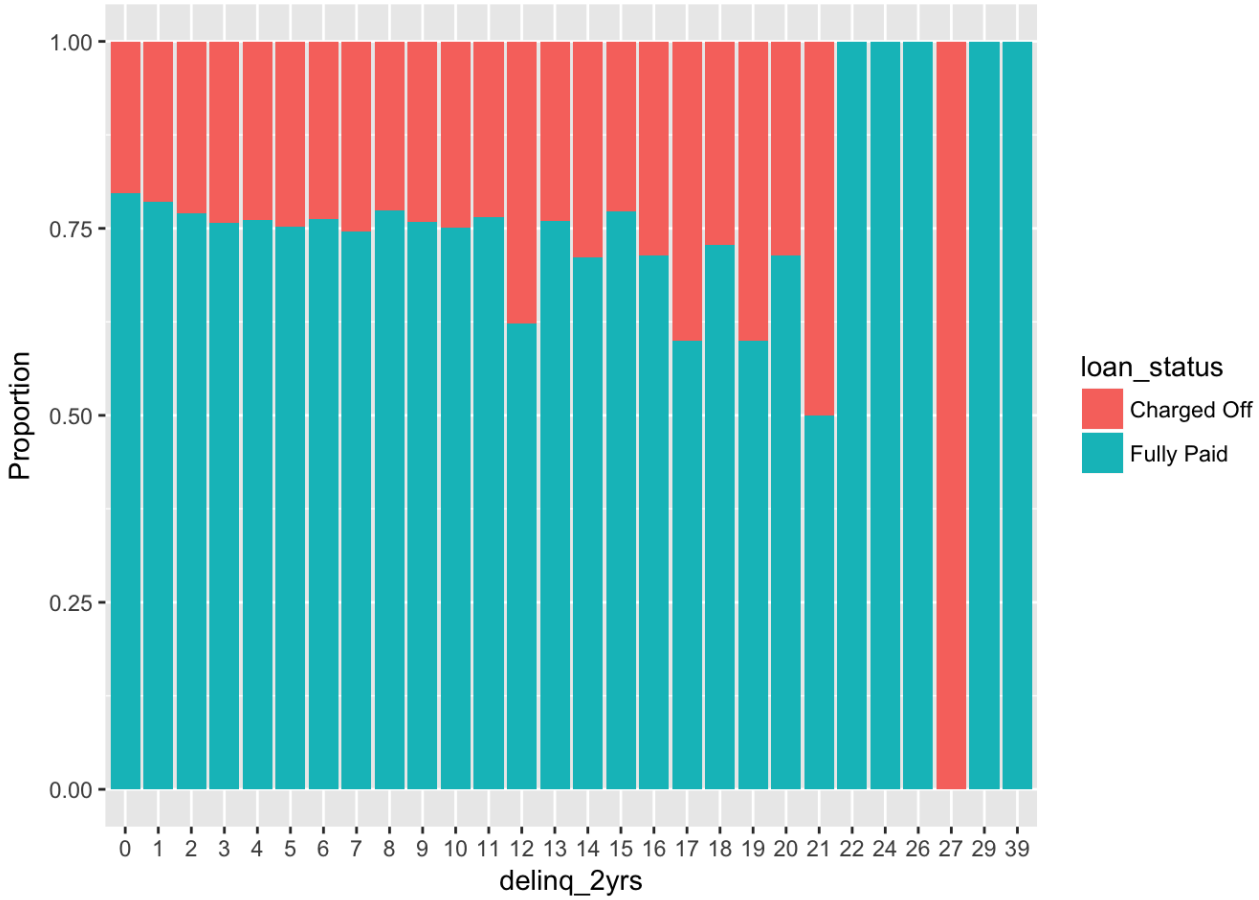
5.7. Plot of Loan Status Vs. DTI

```
dti_plot <- ggplot(data = BIGDF, aes(dti/100)) + xlim(0,1)
dti_plot <- dti_plot + geom_bar(aes(fill = grade))
dti_plot + facet_grid(grade ~ .)
```



5.8. Plot of Delinquents over the last 2 years vs. Loan Status

```
plotdelinquents <- ggplot(BIGDF, aes(x = as.factor(delinq_2yrs), fill = loan_st
atus)) + geom_bar(position = "fill") + labs(x = "delinq_2yrs", y = "Proportion"
 )
```

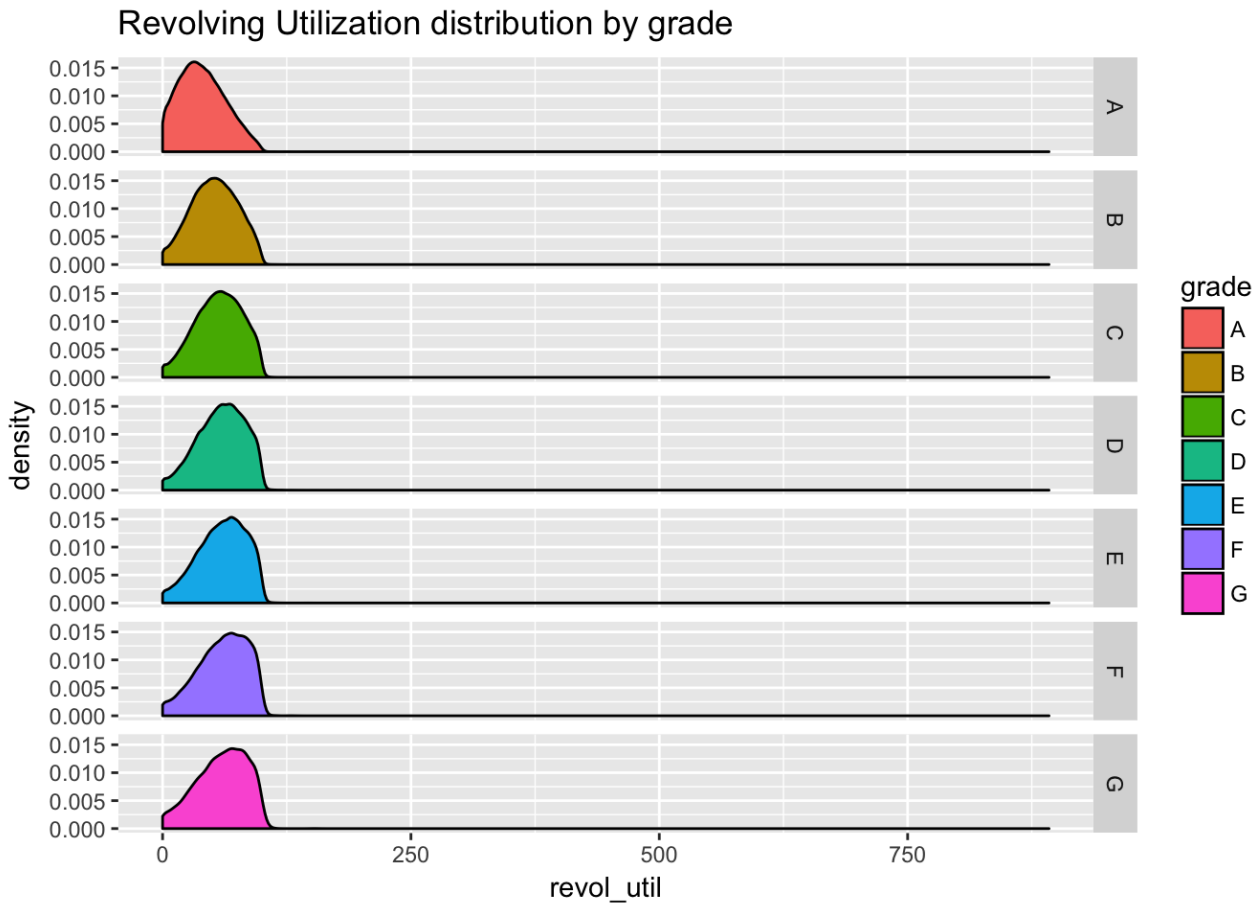



```
##
##                     0      1      2      3      4      5      6      7
##   Charged Off  118248  19646   6160   2310   1013    534    285    168
##   Fully Paid   466069  72007  20689   7196   3236   1616    916    493
##
##                     8      9     10     11     12     13     14     15
##   Charged Off      85     56     39     27     29     12     13      5
##   Fully Paid      291    176    118     88     48     38     32     17
##
##                    16     17     18     19     20     21     22     24
##   Charged Off       4      4      3      2      2      2      0      0
##   Fully Paid       10      6      8      3      5      2      2      1
##
##                    26     27     29     39
##   Charged Off       0      1      0      0
##   Fully Paid        2      0      1      1
```

This plot has some weirdness going on with it, although we can see a trend that looks like the higher the amount of delinquent accounts a customer has, the higher the chance they'll Charge Off their loan. I'm not sure what happened at the end though, it looks like there may have been an issue with the data. I still think that this variable is also a good indicator of likelihood a customer will charge off.

5.9. Plot of Revolving Credit Utilization

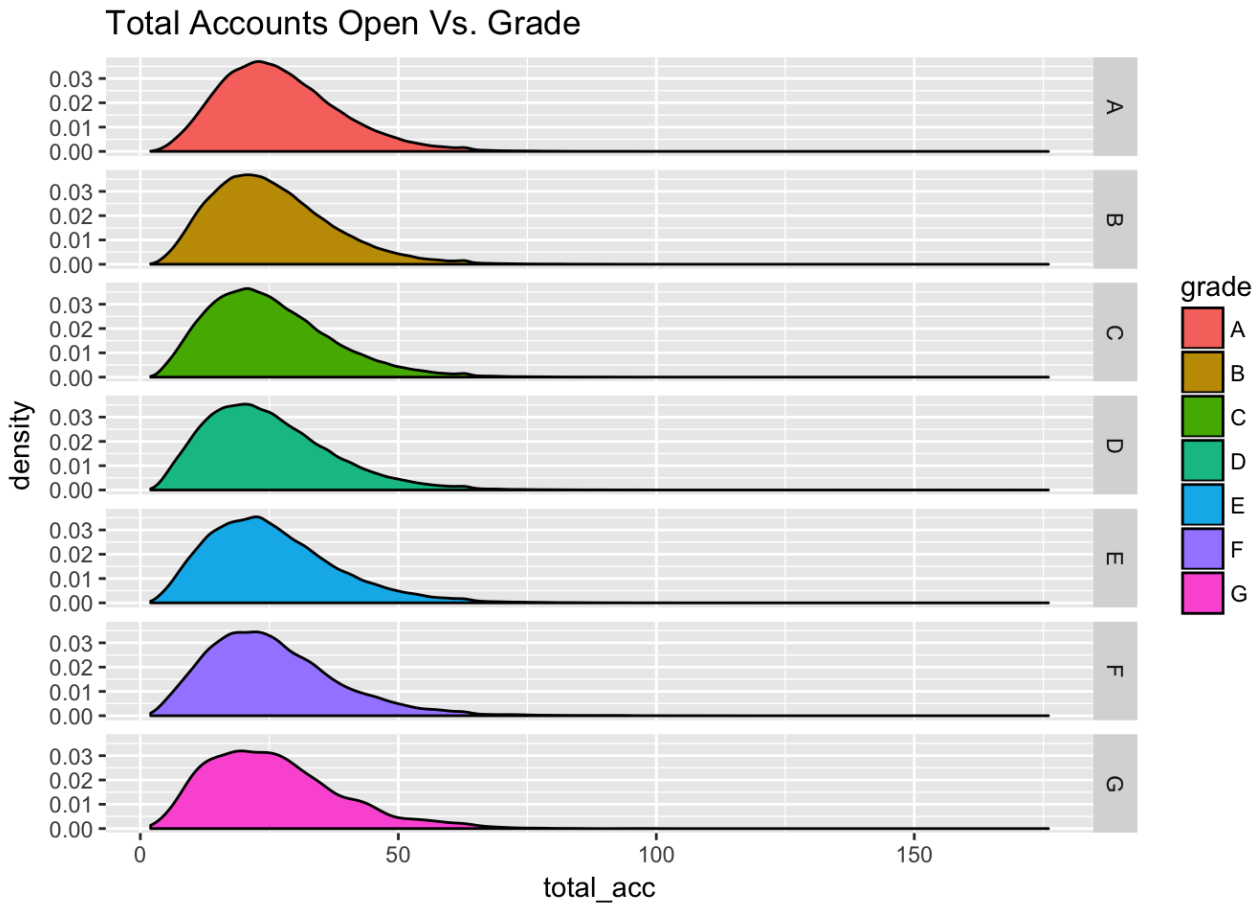
```
ggplot(data = BIGDF, aes(revol_util)) + geom_density(aes(fill = grade)) + labs
(title = 'Revolving Utilization distribution by grade') + facet_grid(grade~.)
```



From this plot, we can actually see quite clearly that as we go down in grade (and quality of credit) the amount of utilization moves much closer towards 100%. To be honest, I am actually surprised at how well this graph shows the linear relationship between the quality of a customer's total utilization of their credit and the grade they receive on their loan (and thus their interest rate). The lower your utilization of your revolving credit, the better your credit is, the better your profile will score with a lender and the lower the interest rate you'll receive on your loan

5.10. Plot of Total Accounts

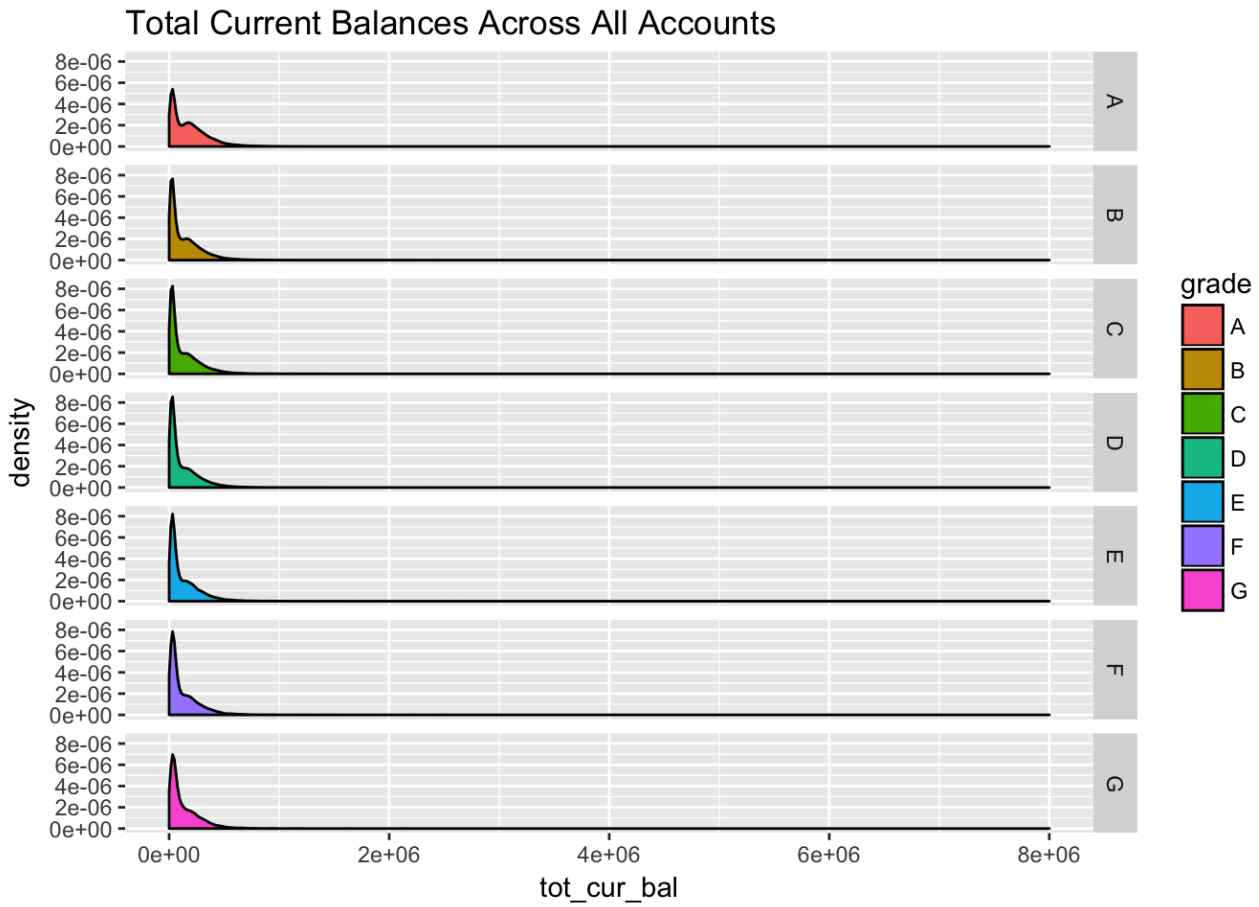
```
ggplot(data = BIGDF, aes(total_acc)) + geom_density(aes(fill = grade)) + labs(t
itle = 'Total Accounts Open Vs. Grade') + facet_grid(grade~.)
```



This is an interesting distribution. We can see that it's slightly reverse form the last visualization. This makes sense intuitively because the more credit you have, the more it shows you can handle credit and the more lenders trust you with more money, hence the higher grade. That's why we see the loans in the Grade A moving towards having more open accounts and the ones at Grade G have much fewer accounts open.

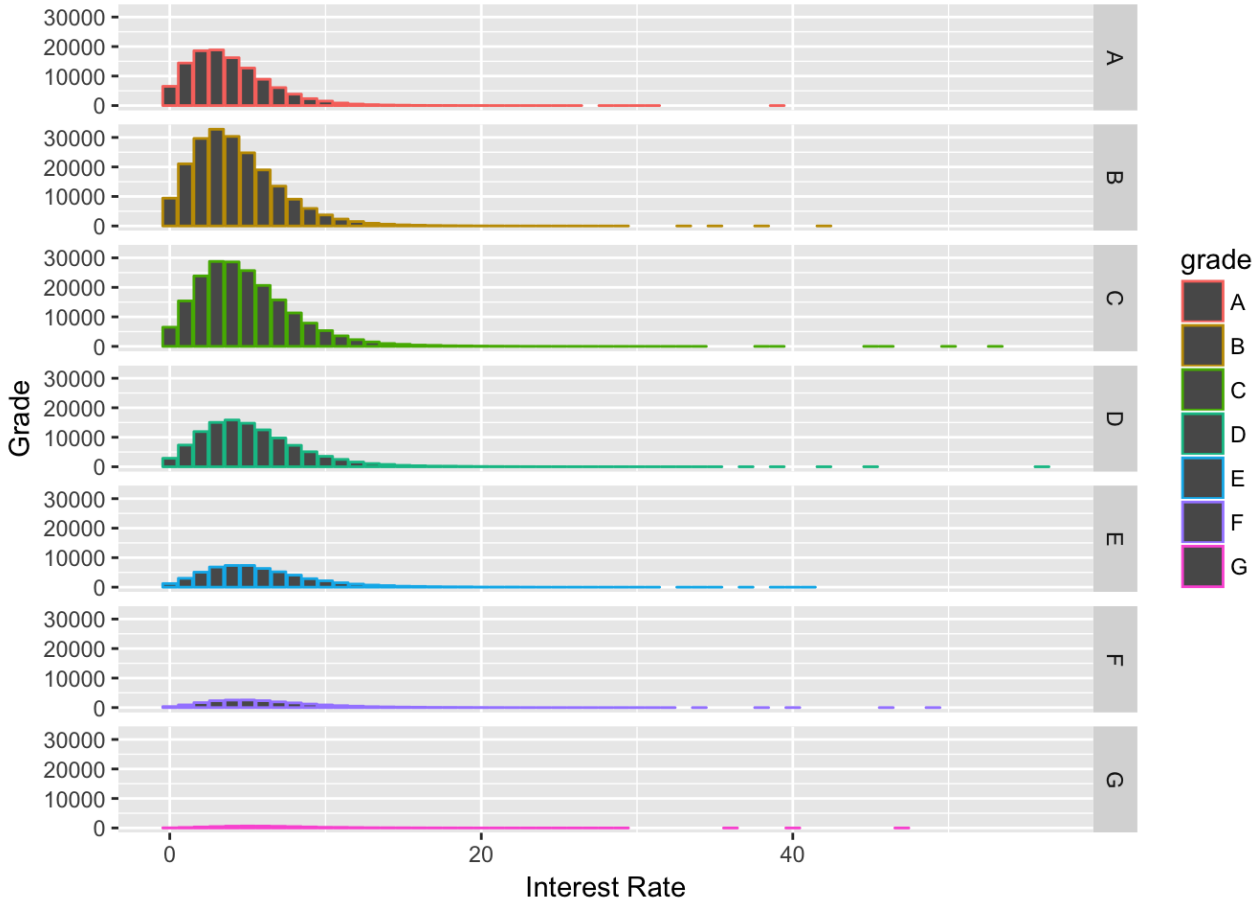
5.11. Plot of Total Current Balances

```
ggplot(data = BIGDF, aes(tot_cur_bal)) + geom_density(aes(fill = grade)) + labs
(title = 'Total Current Balances Across All Accounts') + facet_grid(grade~.)
```



5.12. Plot of Accounts Opened in the last 24 Months

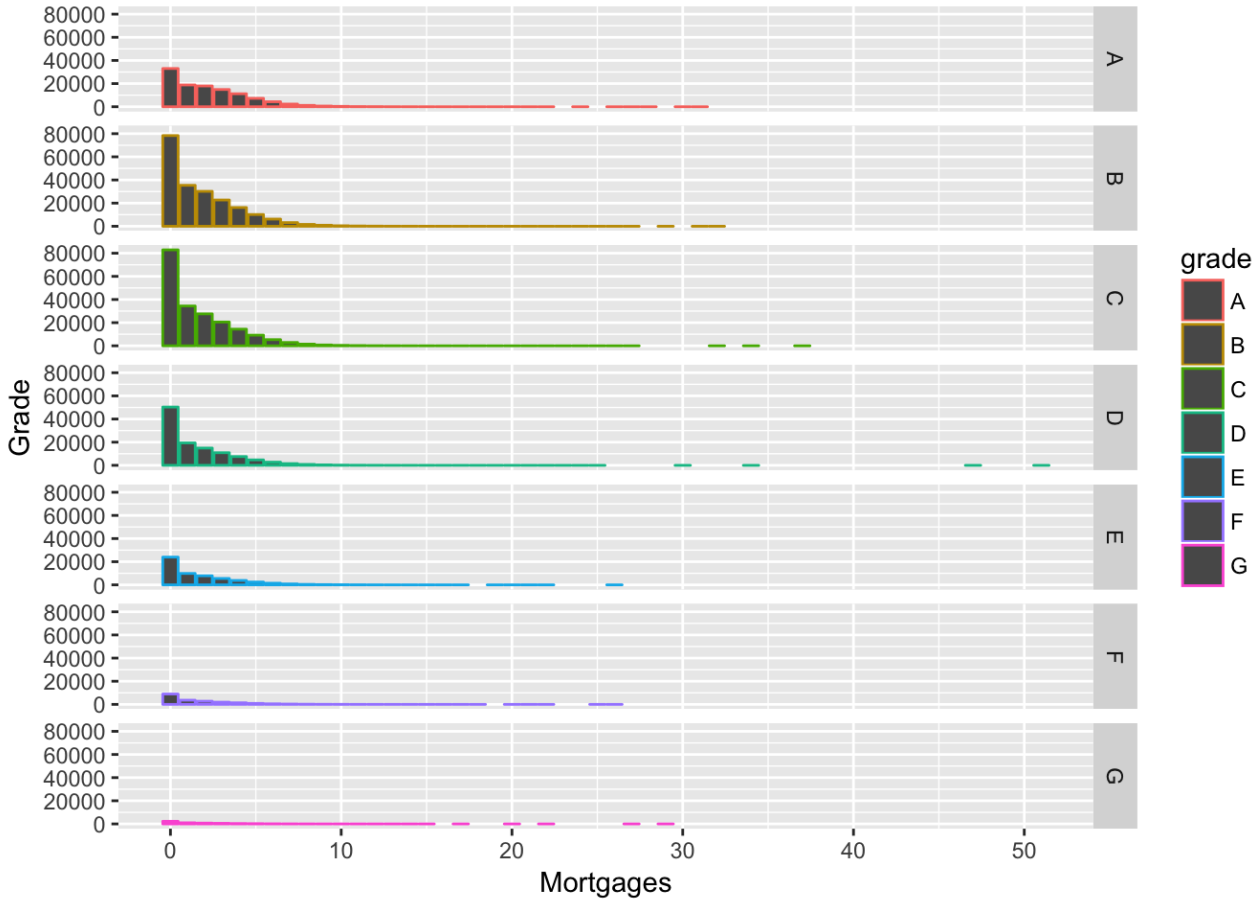
```
ggplot(BIGDF, aes(acc_open_past_24mths, col = grade)) + geom_bar() + xlab('Inte
rest Rate') + ylab('Grade') + facet_grid(grade~.)
```



We can see something interesting in this distribution as well. It looks like the highest grade loans aren't necessarily the ones opening the most credit lines. The lowest grade loans are barely opening any credit lines. It actually stands to reason that the C & B tier loans would be the ones opening credit lines the most, since opening credit lines can help to improve your credit score by increasing your credit depth, which is how you go from a lower score to a higher one.

5.13. Plot of Mortgage Accounts

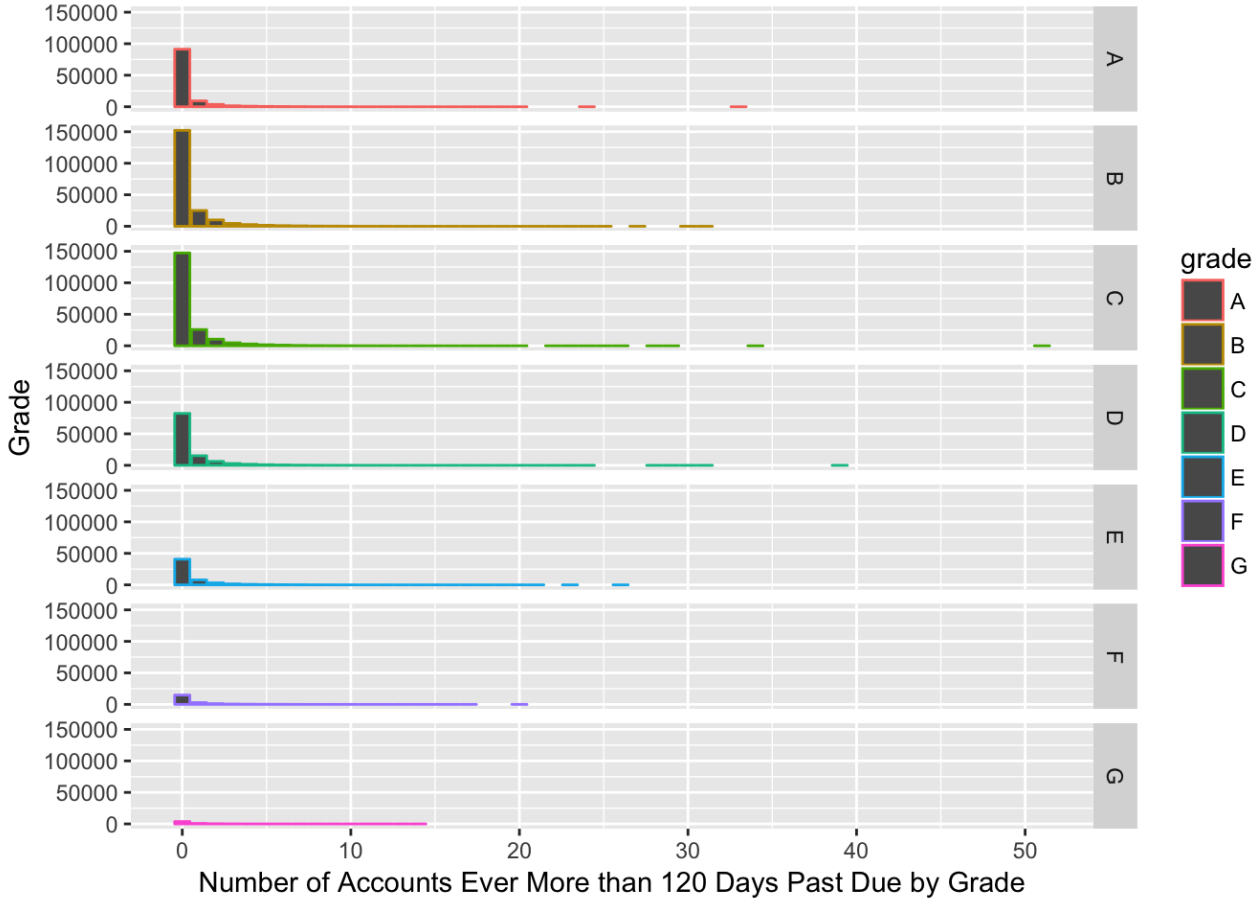
```
ggplot(BIGDF, aes(mort_acc, col = grade)) + geom_bar() + xlab('Mortgages') + yl
ab('Grade') + facet_grid(grade~.)
```



This distribution is very interesting. I am actually surprised that what looks like the majority of Mortgages wind with either B or C grades. Based on my prior lending experience and after doing a little reading, it looks like what actually may be happening is that when someone takes out a mortgage, the immediate imbalance in their Debt to Income ratio (dti) drops their credit score quite a bit since they're taking on a very large debt. However someone's income usually doesn't scale up along with the size of their mortgage. The individuals in these groups may still have their credit affected from having purchased the home.

5.14. Plot of Number of Accounts Ever 120 Days Past Due

```
ggplot(BIGDF, aes(num_accts_ever_120_pd, col = grade)) + geom_bar() + xlab('Num
ber of Accounts Ever More than 120 Days Past Due by Grade') + ylab('Grade') + f
acet_grid(grade~.)
```



This one completely perplexes me. I would expect that at the higher grades, we would have almost no accounts that had ever gone more than 120 days past due and we would see a linear relationship towards the lowest grades having the highest amount of accounts with more than 120 days past due, due to their credit managements skills being much worse. It may be that at the lower grades, those customers just don't even have enough open credit to even have any tales in the first place. I think this distribution can be explained by the fact that both data and real life are messy and that even if you have a very high credit rating, people aren't perfect. It does make sense that the amounts get lower at the highest grade though. That's indicative of those customers having better credit management skills

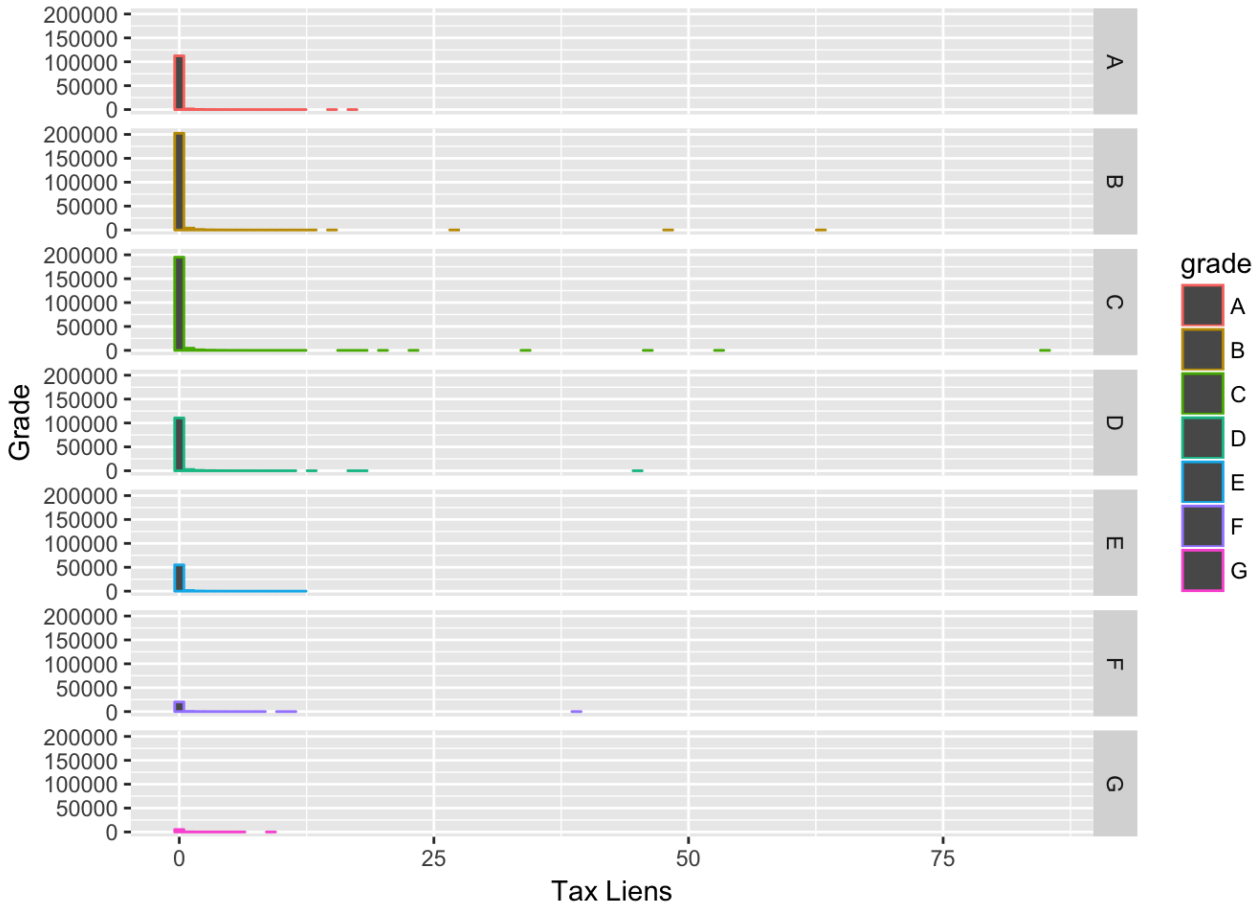
5.15. Plot of Number of Active Revolving Tradelines

```
ggplot(BIGDF, aes(num_actv_rev_tl, col = grade)) + geom_bar() + xlab('Number of
 Active Revolving Tradelines ') + ylab('Grade') + facet_grid(grade~.)
```

We see that this plot mimics the previous plot's distribution

5.16. Plot of Tax Liens

```
ggplot(BIGDF, aes(tax_liens, col = grade)) + geom_bar() + xlab('Tax Liens') + y
lab('Grade') + facet_grid(grade~.)
```



Once again, I'm surprised by the data. I think that the observations from Plots # 13, 14, 15 & 16 are not only highly correlated to loan charge offs, but also very highly correlated to each other.

6. Conclusion - Interpreting the story the data is telling us.

This project focused specifically on providing insights and recommendations on what factors to keep in mind when choosing a Loan to invest in on Lending Club's Peer-to-Peer platform. There were quite a few hurdles that needed to be overcome in order to get us to this point. First, we had several large data sets, which we had to combine into one large data set. Once we had that final large data set, there were many missing values and data in some formats that had to be dealt with. Due to the sheer size of the original data set we had to manually take out all of the variables that we could intuitively tell weren't going to be of use in this project. Then we also filtered out all of the observations that were loans in progress or any other status that wasn't "Fully Paid" or "Charged Off" since we were interested in how these variables correlated to a completed loan. From 145 variables and 1,722,935 observations, we were able to cut the data down to 32 Variables and 721,719 observations.

From there, we were able to use a stepAIC machine learning algorithm to find 16 variables that were statistically correlated enough for us to use as identifiers for what to avoid when looking to invest in loans with lending club (15 from the data and the 'Year' variable that I added).

These 16 variables were: 1) Year (a variable we added) 2) term
3) grade
4) loan_amnt
5) int_rate
6) annual_inc

7) dti
8) delinq_2yrs
9) revol_util
10) total_acc
11) tot_cur_bal
12) acc_open_past_24mths
13) mort_acc
14) num_accts_ever_120_pd 15) num_actv_rev_tl
16) tax_liens

Now, even after using a step AIC to find the variables with the highest correlations, we still needed to actually see the data to determine what those relationships were and how they would impact our investment decisions. Using ggplot2, we also able to visualize those variables to see exactly what their relationship to charge offs looked like. We've also uncovered a couple of interesting patterns that weren't apparent before we took the time to visualize them, some of which took me completely by surprise. We're ultimately left with 10 variables that will have the biggest impact on choosing a loan to invest in and making a secure return on your investment:

So to finally tie all of the data analysis together, here are the variables I recommend looking for when investing in a loan from Lending club and the variables I recommend avoiding if you want to make a return on your investment:

1. Term: Go for the 36 month term. As we can see, the shorter the term is, the higher the chance a customer will pay the loan back. The percentage of loans that charge off at 60 months is almost double the loans at 36 months!

2. Grade: You want to invest in loans that are graded A,B & C. Any Grades worse than C have a dangerously high chance of charging off and losing all of the money you invested.
3. Loan Amount: When looking for a loan to invest in, the sweet spot based on the graph seems to be right around 15-20k. Any more than that and you see loans with D,E,F & G Grades, heading into Charge Off territory, which we want to steer clear of. The higher the amount a customer borrows, the higher their monthly repayment will be and the more financial stress they'll be under.

4. Interest Rate: Clearly, the lower the better. Interest rate is going to be in lock-step with Grade, since grade is determined by the customer's credit attributes and FICO score (not provided to us). Lending Club then assigns a grade and determines the interest rate based off of that. Lower interest rates mean lower monthly repayment amounts, which means less stress on the customer to pay the loan back.The highest interest rate in the C Grade category looks to be around 18%. I would recommend investing in anything lower and not investing in anything higher than that.

5. Annual Income: As a general rule, the higher the customer's income the better the chance they'll be able to pay the loan back. I wouldn't necessarily target only the customers with the highest annual incomes, although that's not a bad bet. From the graph, I would look for a combination of annual income over 50k and lower loan amounts. This would be the safest best.

6. DTI (debt to income ratio): You want to look for loans that have A, B & C Grades since they'll have lower dti, meaning they'll be under less financial stress to pay the loan back. Any grade lower than C either has too many other financial obligations which would impact their ability to

pay back the Lending club loan, or at the very bottom of the barrel, they don't have any credit accounts, meaning they don't have any experience managing credit accounts and are at a much higher risk of charging off if they're using a financial instrument they don't understand.

7. Delinquents in the last 2 years: You really don't want to invest in a loan with a customer who's had any delinquent payments in the last 2 years, as we can clearly see that the more of those delinquent payments they have, the worse their loan grades are (which means the higher the chance they will charge off their loan). I would recommend not investing in ANY loans with delinquents.

8. Revolving Credit Utilized: It's actually okay if the customer has some of their credit utilized, they don't have to be at zero to have a great credit score. The lower end of utilization of the Grade A loans looks actually to be about 30%. What you do want to avoid however, is any loan where the customer has 50% or more of their revolving credit utilized. Those loans quickly head into D,E,F & G territory where the chance of charge off is pretty high.

9. Total Accounts Open: This distribution looks a little more even that I thought it would, but we can see that the loans at Grade A more towards having more accounts open, and the loans at the bottom, with the worst grades move towards having less accounts open. A good rule of thumb here would be that having access to more credit is better, as it means that the customer likely has more experience using their credit. You can be pretty flexible here, but I'd recommend to invest in loans where the customer has many credit accounts open (Along with a better grade, lower dti,)

10. Accounts Opened in the Last 24 Months: This is one of the ones that completely surprised me. The lowest grade loans are barely opening any credit lines. It actually stands to reason that the C & B tier loans would be the ones opening credit lines the most, since opening credit lines can help to improve your credit score by increasing your credit depth, which is how you go from a lower score to a higher one. Here also, it seems like more is better. If you're looking at investing and using number of accounts opened in the last 24 months, a safe bet is to go with the customers who have the highest amount, you're likely to get a B or C Grade loan with a very acceptable chance of paying their loan back.

The other variables outside of this, did have high statistical correlation, however, from what I can tell, it looks like the amount of those data points present in the data was too small for us to draw any real conclusions. I think this is because if we look at a variable like Tax Liens for example, the number of data points is so small, that what we see is very skewed.

So, to sum it all up: Your ideal customer would have an A,B or C Grade on their loan, higher amounts of accounts open, Interest rate under 18%, the shortest term possible, asking for a loan amount under 25k with an annual income of at least 50k (higher is better), a higher number of accounts opened in the last 24 months, and a DTI/ Revolving Credit Utilization ideally somewhere between 30%-50%, lower can be better, but you'd want to see some utilization as you want to make sure the customer knows how to use credit.

If you focus on these 10 attributes, you will have the most secure investment portfolio and the highest chance at making a return on your investment through Lending Club.

```
What to Look for:

Terms: 36 Months
Grades: A,B & C
Loan Amounts: Under 25k
Interest Rate: Anything under 18%
Annual Income: Over ~50k (Especially relative to loan amount)
DTI/Revolving Credit Utilized: in the 10%-50% Range (~30% is the sweet spot)
Delinquents: 0
Accounts Open/ in the last 24 months: More is better!
```

```
What to Avoid:

Terms: longer than 36 Months
Grades: D,E,F,G
Loan Amounts: in the higher end (anything really over 30k)
Interest rate: Anything over 18%
Annual Income: under ~50K (Especially when combined with higher loan amounts)
DTI/Revolving Credit Utilized: under 10% or over 50%
Delinquents in the past 2 years: ANY loans where the customer had delinquents
Accounts Open/in the last 24 moonths: Low to zero. You want to see them growing
their
credit
```

6.2 Going Further

This project focused specifically on providing insights and recommendations on what factors to look for when choosing a Lending Club loan to invest in. Beyond this question, there are some other interesting patterns that we can find in the data that may not directly answer the above question, but may be of interest to investigate at a further time.

In this project I did not include specific time data (I only compared the entire portfolio's performance over years), but it would be interesting to try and determine if there are seasonal trends.

I also only looked at loans that had been completed- either fully paid, or charged off. Expanding the data set to include analysis of loans that are still in repayment would offer additional insights on what data points may indicate an active loan that is in trouble, or a loan that is likelier to be fully paid eventually (such as looking at amount paid vs. default rate, etc.)

Since Lending club no longer shares the FICO scores of the borrowers of their loans, it would also be interesting to build a model to determine which variables affect the interest rate received on any loan and determine what grade and interest rate a loan would receive based off of those variables.

There were also some observations in the scope of this project that surprised me. I had thought that the relationship between loan charge offs, interest rates and consumer credit was going to be linear for each variable, but as we can see, that's definitely not the case. The last 4 data points in particular have very interesting relationships to the Loan grade (and thus the customer's credit profile). I'd be really interested in digging into those variables in a future project, perhaps to see if my hypotheses about them above are correct, or if there's something else there that I could be missing.

I'd also love to see what variables would be reliable enough predictors to build a predictive model around. I think I have the bulk of them covered in this project, but I would love to try and beat the native accuracy and come up with a model that would reliably out-perform random chance to aid in creating a lending club portfolio that would produce reliable returns.

Finally, taking this project one step further, the most useful way to put this information to work would be to not only provide the user information on specific loans in a dashboard, but also to write a program that can access your Lending Club account, invest in loans automatically based on certain criteria, and withdraw or invest more at certain thresholds. This would be the culmination of the entire project and in the real world, the actual deliverable, an algorithm that would identify the significant data points, take action upon them, and most importantly, generate a return - all automatically. Creating a porfolio and doing this for myself would be the ultimate test of whether or not I have the right loan predictors after all.