Final Project DS 705

Joe Athas

# Introduction

It is very important banks only give loans to those that are able to pay them back in full. Loans without a lien can be a potential risk for a bank. Loans that have a lien or collateral are also very costly in the event of a default. The goal for this data exploration and analysis is to determine a model with appropriate predictor variables using data often gathered by financial institutions and to use that model to predict the likelihood of a loan applicant defaulting or paying off the loan. To do this some manipulation of the data may be in order. I will be using R to do preliminary exploration, cleaning, and transformation of the provided sample data, “loans50k.csv”.

citation()

##   
## To cite R in publications use:  
##   
## R Core Team (2018). R: A language and environment for  
## statistical computing. R Foundation for Statistical Computing,  
## Vienna, Austria. URL https://www.R-project.org/.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {R: A Language and Environment for Statistical Computing},  
## author = {{R Core Team}},  
## organization = {R Foundation for Statistical Computing},  
## address = {Vienna, Austria},  
## year = {2018},  
## url = {https://www.R-project.org/},  
## }  
##   
## We have invested a lot of time and effort in creating R, please  
## cite it when using it for data analysis. See also  
## 'citation("pkgname")' for citing R packages.

# Section 3 Preparing and Cleaning the data

Upon receiving the data, the first thing I did was make the appropriate response variable: A binary variable (as is needed for logistic regression) produced from the original variable “status”. Fully paid loans were reclassified as good, those in “default” or “charged off” were reclassified as bad. All other levels of status (loans in progress or not complete) were removed from the dataset.

#Make new variable: Set "Good" and "Bad" Loans, all else "blank" to be removed later  
df <-  
 loans50k %>%  
 mutate(payment\_status = case\_when(  
 status == "Fully Paid" ~ "Good",   
 status == "Charged Off" ~ "Bad",  
 status == "Default" ~ "Bad",  
 TRUE ~ ""  
 ))  
#Probably best to get rid of the null values we aren't interested in payment\_status:  
df<-df %>% drop\_na(payment\_status)  
  
#change the empty string cells to NAs  
df[df==""]<-NA

Turning any fields into NA values makes it easier to eliminate later I also removed “employed” immediately as there were too many levels and empty values. Last thing I did before really digging into the data was removing all the rows containing NA values. All the remaining NAs were in I noticed it only reduced n by approximately 1% which didn’t compromise the integrity of the original set all too much. A few other variables were removed for redundancy.

#Test where all NAs are to   
#which(is.na(df), arr.ind=TRUE)  
#Found a bunch in "employment". Going to eliminate on account of too many levels/NA values.   
df$employment<- NULL  
  
#Test where the NAs are again:  
which(is.na(df), arr.ind=TRUE)

## row col  
## [1,] 34806 2  
## [2,] 34806 3  
## [3,] 34806 4  
## [4,] 34806 5  
## [5,] 34806 6  
## [6,] 34806 7  
## [7,] 34806 8  
## [8,] 34806 9  
## [9,] 34806 10  
## [10,] 34806 11  
## [ reached getOption("max.print") -- omitted 16391 rows ]

#last NA values are in bcOpen and bcRatio, of which there are roughtly 400 incomplete entries,   
#or roughly 1% of the dataset. Eliminating these wont be a huge issue.   
  
#Finally omit all the rows containing NA values. (This will not eliminate the NA values in "length" as they are strings)  
df1<-na.omit(df)

Here a covariance matrix was constructed of the numerical predictors to aid in this process of determining redundancy. I chose an arbitrary value of .8 of Pearson’s r to classify variables as strongly correlated. The strong correlations were as follows: Total limit of credit cards was heavily correlated with Total unused credit on cards. Unused credit (bcOpen) seemed like a more logical choice than totalBcLim (total limit of cc) and it eliminated some redundancy. ‘Payment’ was heavily associated with ‘amount’ as monthly payment is a function of the total amount of the loan. I deleted this variable to eliminate redundancy. Proportion of revolving credit use(revolRatio) was eliminated due to redundancy with ratio of total credit card limits (bcOpen). Average balance and total credit limit were both strongly correlated with total Balance. I figured there is a lot of redundancy between these three variables, keeping just totalBal would suffice.

#make subset of numerical data check covariance matrix on numeric factors.   
dfNum<-select\_if(df1, is.numeric)  
m<-cor(dfNum)  
#easy way to find strong correlations; arbitrary value set at .8  
strongCorrelations = m > .8  
#Found few redundant variables...will eliminate

A few predictor variables were initially removed due to lack of relevance. These included “totalPaid”, “state”, and “loanID”. “State” and “loanID” have nothing really to do with the actual loan, whereas “totalpaid” cannot be a predictor variable of a completed loan. A few more were eliminated based on redundancy as mentioned above.

#Final check for number of levels in each categorical variable  
dfChar<-select\_if(df1,is.character)   
sapply((sapply(dfChar,unique)),length)

## term grade length home verified   
## 2 7 12 3 3   
## status reason state payment\_status   
## 3 13 49 2

cor(df$avgBal,df$totalLim)

## [1] NA

#Eliminating: totalPaid, employment (already done), payment,revolRatio, avgBal, totalLim,state,status,loanID  
df\_New = as.data.frame(subset(df1, select = -c(totalPaid, payment,revolRatio, avgBal, totalLim,state,status,loanID,totalIlLim,totalBcLim)))

A few other variables I touched up were “verified” and “reason”. “Verified” had three levels, two of which were redundant; “source verified” and “verified”. I changed all “source verified” to just “verified” to eliminate the mistake. “Reason” contained a few levels with very few instances which is considered less powerful for logistic regression. I modified “wedding” and “renewable\_energy” to “other” and “house” to “major\_purchase” to consolidate a few levels. After going about data transformation in a later step, a few more concerning variables were mutated. ‘pubRec’ and ‘delinq2yr’ (number of derogatory public records and number of 30+ day late payments in last two years respectively) both contained many instances of 0 due to their nature. Since many loans have ‘0’ represented in these respective variables, choosing to set it as categorical made more sense. I chose three levels for both variables (the same since they were similar scales) “None”, “1-3” and “More than 3”. Instructions

#consolidate "verified" variable into two levels instead of three  
df1$verified<-revalue(df1$verified, c("Source Verified"="Verified"))  
unique(df1$verified)

## [1] "Verified" "Not Verified"

#going to combine a few levels in "reason" due to the lack of instances  
df\_New$reason<-revalue(df\_New$reason,c("wedding"="other","renewable\_energy"="other","house"="major\_purchase"))  
  
#make df of only continuous variables  
dfCont<-select\_if(df\_New, is.character)  
#Test for instances of each level  
x=sapply(dfCont,count)  
x

## term grade length home verified reason   
## x factor,2 factor,7 factor,12 factor,3 factor,3 factor,10   
## freq Integer,2 Integer,7 Integer,12 Integer,3 Integer,3 Integer,10  
## payment\_status  
## x factor,2   
## freq Integer,2

#lowest #of instances is now above 200

### After removing redundencies and a few predictor variables, initial cleaning of dataset brings n = 34271.

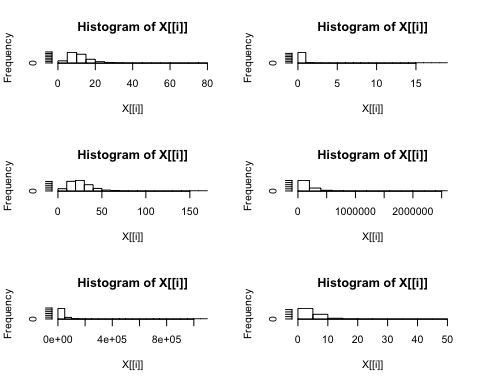
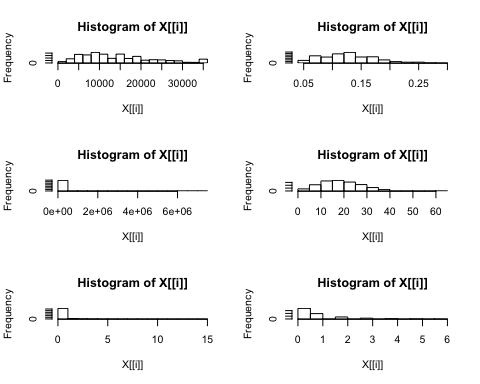
# Section 4 - “Exploring and Transforming the data”

Certain data transformations were necessary in order to meet the assumption of normality for logistic regression. I first analyzed the skewness of each distribution. Considering a skewness greater than the absolute value of 1 is considered “highly skewed”, I created a r function to attempt various transformations, specifically a log, natural log, square root, cube root, 4th root, and reciprocal, then return the skewness of the transformed distribution. I applied this function to every variable with an initial skewness greater than the absolute value of 1 and determined which transformation technique was the most successful at reducing skewness. Obviously this is a blanket technique, and eaach variable still needs to be looked at individually. After determining which transformation to use for each variable, I performed the transformations as follows income: log transformation openAcc: log transformation totalBal: cubed root transformation totalRevLim: cubed root transformation accOpen24: square root transformation bcOpen: cube root transformation totalRevBal: cube root transformation

## Section 4:

Be careful with log transformation.  log(0) produces NA.  So log(openAcc) might be problematic.  Use log(openAcc+1) is openAcc has any 0 values.  Same for other variables that take on 0 values.##

#make new df of only numeric variables.  
dfNum<-select\_if(df\_New, is.numeric)  
  
#check out skewness of all numeric variables  
par(mfrow=c(3,2))  
sapply(dfNum, hist)

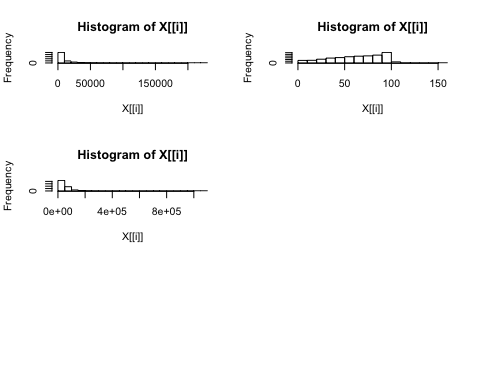


## amount rate income debtIncRat delinq2yr inq6mth   
## breaks Numeric,19 Numeric,14 Numeric,16 Numeric,14 Numeric,16 Numeric,13  
## openAcc pubRec totalAcc totalBal totalRevLim  
## breaks Numeric,17 Numeric,20 Numeric,18 Numeric,14 Numeric,23   
## accOpen24 bcOpen bcRatio totalRevBal  
## breaks Numeric,11 Numeric,24 Numeric,17 Numeric,23   
## [ reached getOption("max.print") -- omitted 5 rows ]

sapply(dfNum,skewness)

## amount rate income debtIncRat delinq2yr inq6mth   
## 0.7291701 0.4639609 47.6301319 0.2379811 4.9530701 1.7319227   
## openAcc pubRec totalAcc totalBal totalRevLim accOpen24   
## 1.3715815 5.1507168 0.9741543 2.5313429 6.8067971 1.3436559   
## bcOpen bcRatio totalRevBal   
## 3.8062866 -0.4488958 3.9351897

#Distributions that have a skewness greater than 1 are generally considered "highly skewed"  
#will transform: income, delinq2yr,inq6mth,openAcc,pubRec,totalBal,totalRevLim,accOpen24,bcOpen,totalRevBal



#function to success of various transformations  
tansformAndGetSkew<-function(dist1){  
 log=skew(log(dist1))  
 lognat=skew(log(dist1,base=exp(1)))  
 sqrt=skew(sqrt(dist1))  
 curt=skew(sign(dist1) \* abs(dist1)^(1/3) )  
 fthrt=skew(sign(dist1)^(1/4))  
 reciprical=skew(1/dist1)  
   
 list1<- c(log,lognat,sqrt,curt,fthrt,reciprical)  
   
 return(list1)  
}  
#apply function to all numerical variables  
sapply(dfNum, tansformAndGetSkew)

## amount rate income debtIncRat delinq2yr inq6mth  
## [1,] -0.66118527 -0.32793006 0.2154722 NaN NaN NaN  
## openAcc pubRec totalAcc totalBal totalRevLim accOpen24  
## [1,] -0.2240531 NaN -0.51300126 NaN -0.1432333 NaN  
## bcOpen bcRatio totalRevBal  
## [1,] NaN NaN NaN  
## [ reached getOption("max.print") -- omitted 5 rows ]

#Use results to infer which might be the best type of transformation  
#Changing pubRec & delinqyr to categorical variables due to too many instances of '0' in both.   
  
#use unique() to determine breaks  
unique(df\_New$pubRec)

## [1] 0 3 1 2 4 6 5 7 11 9 19 8 10 13

#Make three levels, one containing zero, 1-3, and more than 3  
df\_New$pubRec<-cut(df\_New$pubRec,breaks=c(-.10,.1,3,14))  
levels(df\_New$pubRec)<-c("None","1-3","More than 3")  
  
unique(df\_New$pubRec)

## [1] None 1-3 More than 3 <NA>   
## Levels: None 1-3 More than 3

#DO THIS AGAIN But with delinqyr  
df\_New$delinq2yr<-cut(df\_New$delinq2yr,breaks=c(-.10,.1,3,12))  
levels(df\_New$delinq2yr)<-c("None","1-3","More than 3")

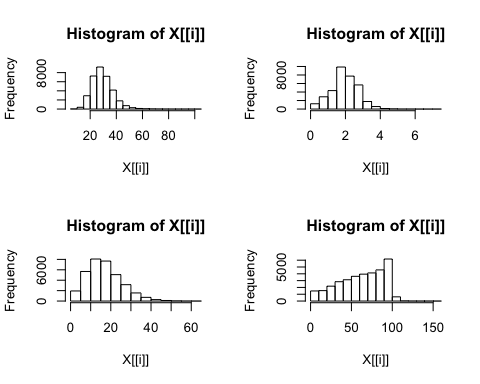
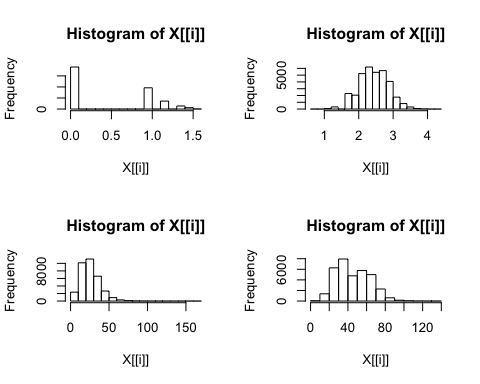
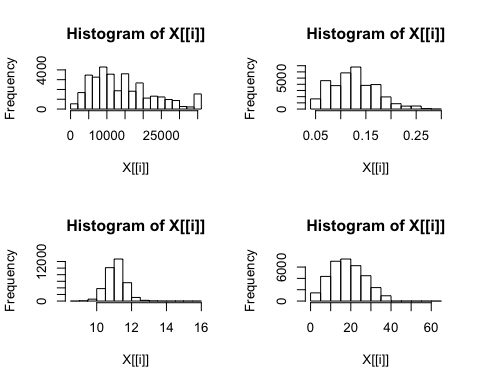
#begin transformations  
df\_New$income<-log(df\_New$income)  
df\_New$inq6mth<-(df\_New$inq6mth)^(1/4)  
df\_New$openAcc<-log(df\_New$openAcc+1) #add 1 to oppenAcc to avoid taking log(0)  
df\_New$totalBal<-(df\_New$totalBal)^(1/3)  
df\_New$totalRevLim<-(df\_New$totalRevLim)^(1/3)  
df\_New$accOpen24<-sqrt(df\_New$accOpen24)  
df\_New$bcOpen<-(df\_New$bcOpen)^(1/3)  
df\_New$totalRevBal<-df\_New$totalRevBal^(1/3)  
  
#Check if any NA values induced  
which(is.na(df\_New), arr.ind=TRUE)

## row col  
## [1,] 1892 11  
## [2,] 2866 11  
## [3,] 22405 11  
## [4,] 25195 11  
## [5,] 25893 11  
## [6,] 28786 11  
## [7,] 30522 11  
## [8,] 16021 14

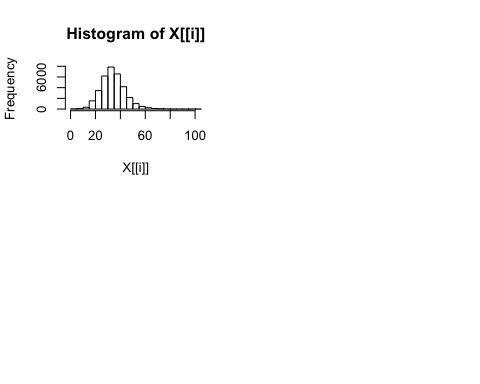
df\_New<-na.omit(df\_New)  
  
  
#make new data frame to test skew on numeric data to see if successful.  
dfNum1<-select\_if(df\_New, is.numeric)  
sapply(dfNum1,skew)

## amount rate income debtIncRat inq6mth openAcc   
## 0.72925623 0.46415616 0.21531227 0.23801258 0.30604804 -0.07696871   
## totalAcc totalBal totalRevLim accOpen24 bcOpen bcRatio   
## 0.97455211 0.39415850 0.90225311 -0.22130509 0.63049964 -0.44917374   
## totalRevBal   
## 0.54301044

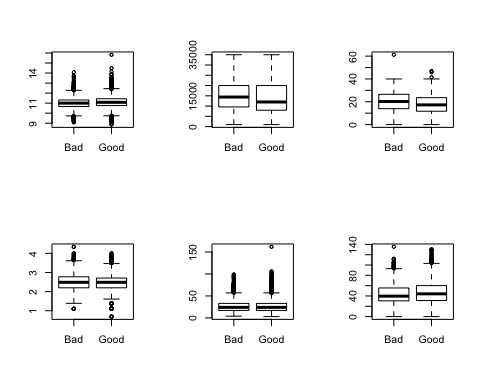
par(mfrow=c(2,2))  
sapply(dfNum1,hist)



## amount rate income debtIncRat inq6mth openAcc   
## breaks Numeric,19 Numeric,14 Numeric,16 Numeric,14 Numeric,17 Numeric,20  
## totalAcc totalBal totalRevLim accOpen24 bcOpen   
## breaks Numeric,18 Numeric,15 Integer,21 Numeric,16 Numeric,14  
## bcRatio totalRevBal  
## breaks Numeric,17 Numeric,22   
## [ reached getOption("max.print") -- omitted 5 rows ]

 #Exploration: Side-by-side boxplots for continuous data and frequency tables for categorical data were created to compare each factor to the dependent variable. Few variables popped out immediately as good indicators, however, few were found. The variable “term” has a much higher ratio of bad loans to good in the loans that are 60 months vs 36 months. Loans rated ‘D’ or lower also have a much higher ratio of being “bad” than their counterparts.

par(mfrow=c(2,3))  
  
#see visual differences if any of good loans vs bad  
boxplot(income~payment\_status,data=df\_New)  
boxplot(amount~payment\_status,data=df\_New)  
boxplot(debtIncRat~payment\_status,data=df\_New)  
boxplot(openAcc~payment\_status,data=df\_New)  
boxplot(totalAcc~payment\_status,data=df\_New)  
boxplot(totalBal~payment\_status,data=df\_New)



boxplot(totalRevLim~payment\_status,data=df\_New)  
boxplot(accOpen24~payment\_status,data=df\_New)  
boxplot(bcOpen~payment\_status,data=df\_New)  
boxplot(totalRevBal~payment\_status,data=df\_New)  
  
#compare frequency tables  
getFreqTable<-function(var1){  
 mytable <- table(var1,df\_New$payment\_status)   
 return(ftable(mytable))  
}  
par(2,2)

## [[1]]  
## NULL  
##   
## [[2]]  
## NULL

getFreqTable(df\_New$term)

## Bad Good  
## var1   
## 36 months 4344 21121  
## 60 months 3158 5640

getFreqTable(df\_New$grade)

## Bad Good  
## var1   
## A 389 5359  
## B 1250 8016  
## C 2260 7311  
## D 1860 3778  
## E 1177 1727  
## F 450 477  
## G 116 93

getFreqTable(df\_New$length)

## Bad Good  
## var1   
## < 1 year 607 2178  
## 1 year 499 1786  
## 10+ years 2333 9024  
## 2 years 643 2417  
## 3 years 595 2144  
## 4 years 461 1581  
## 5 years 462 1597  
## 6 years 357 1193  
## 7 years 345 1282  
## 8 years 402 1309  
## 9 years 284 968  
## n/a 514 1282

getFreqTable(df\_New$home)

## Bad Good  
## var1   
## MORTGAGE 3274 13755  
## OWN 790 2741  
## RENT 3438 10265

getFreqTable(df\_New$verified)

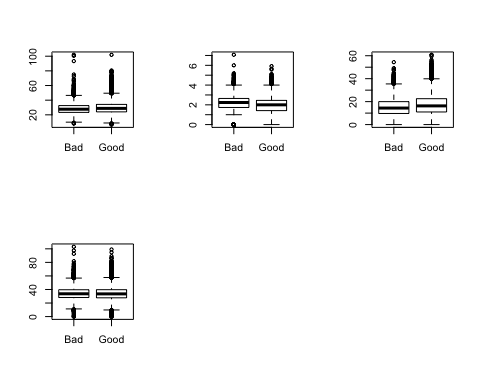
## Bad Good  
## var1   
## Not Verified 1631 8479  
## Source Verified 3254 10912  
## Verified 2617 7370

getFreqTable(df\_New$reason)

## Bad Good  
## var1   
## car 43 234  
## credit\_card 1477 6320  
## debt\_consolidation 4744 16078  
## home\_improvement 404 1581  
## major\_purchase 154 577  
## medical 92 276  
## moving 61 148  
## other 371 1185  
## small\_business 111 202  
## vacation 45 160

getFreqTable(df\_New$delinq2yr)

## Bad Good  
## var1   
## None 5900 21432  
## 1-3 1469 4951  
## More than 3 133 378



#Eventually...  
mylogit <- glm(as.factor(payment\_status) ~ ., data = df\_New, family = "binomial")  
summary(mylogit)

##   
## Call:  
## glm(formula = as.factor(payment\_status) ~ ., family = "binomial",   
## data = df\_New)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7394 0.2995 0.5127 0.7301 1.6608   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.516e+00 5.185e-01 4.852 1.23e-06 \*\*\*  
## amount -1.489e-05 2.422e-06 -6.147 7.89e-10 \*\*\*  
## term60 months -6.442e-01 3.651e-02 -17.646 < 2e-16 \*\*\*  
## rate -2.062e+00 1.225e+00 -1.684 0.092226 .   
## [ reached getOption("max.print") -- omitted 45 rows ]  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 36016 on 34262 degrees of freedom  
## Residual deviance: 32209 on 34214 degrees of freedom  
## AIC: 32307  
##   
## Number of Fisher Scoring iterations: 5

options(max.print = 1000)