Making Sense of $1.3T in Student Loan Debt - An Analysis of Student Loan Hero Users and Predictive Default Model

Trevor Ford

April 2, 2017

Contents

---

1 Introduction

Student loan debt is a $1.4T problem facing over 44 million Americans. The average 2016 graduate left college with over $37,000 of debt and is burdened with over $350 a month in student loan payments. Student Loan Hero was conceived as a resource to help student loan borrowers understand the various student loan repayment schemes, navigate the numerous federal programs designed to help borrowers with their loans, and recommend ways for borrowers to save money on their loans. Student Loan Hero developed a proprietary tool that aggregates all of a borrower's loans and displays the details all in one place, then generates a customized repayment plan based on each borrower's unique circumstances. To date, the tool has analyzed over $2 billion worth of student loan debt and helped 50,000 plus borrowers best pay off their student loans. This

course offered the perfect opportunity to explore the vast amount of borrower data gathered in the three years Student Loan Hero has been operating.

1.1 The Goal

Student loan default rates are on the rise [elaborate here, why default is a bad outcome for borrowers, etc]... By using a binary classification machine learning algorithm we hope to be able to predict whether a given borrower is likely to default on their student loans. The business application of this prediction is if a borrower is likely to default on their student loans, Student Loan Hero has the opportunity to connect with high-risk borrowers and help borrowers take appropriate action to avoid default.

2 Datasets

2.1 Student Loan Hero User Dataset

The private Student Loan Hero User dataset contains data from 24,000 anonymized student loans.

```
names(slhloans)
```

```
##  [1] "User.ID.."                    "User.DOB"
##  [3] "Loan.ID.."                    "Original.Principal"
##  [5] "Current.Principal"            "Rate"
##  [7] "Loan.Disbursement.Date"       "Monthly.Interest"
##  [9] "Monthly.Payment"              "Name"
## [11] "Status"                       "Type"
## [13] "Education.Degree"             "College"
## [15] "Major"                        "Employment.Status"
## [17] "Adjusted.Gross.Income"        "Joint.Federal.Income.Tax."
## [19] "Spouse.Adjusted.Gross.Income" "Employer.Type"
## [21] "Profession"                   "Family.Size"
## [23] "State.of.Residency"           "Credit.Score"
```

2.2 Carnegie University Classifications

The public Carnegie University Classifications dataset is a framework for recognizing and describing institutional diversity in U.S. higher education. The dataset contains detailed data about all Title IV colleges and universities.

3 Preparing and Cleaning Datasets

Since the goal of this project is to predict the likelihood a borrower will default on their loans by use of a binary classification machine learning model it was imperative to narrow the number of levels in a number of factors in the Student Loan Hero dataset.

3.1 Status

The most important factor to address is the loan "status", a factor containing a number of statuses a borrower's loan could be in.

```
str(slhloans$Status)
```

```
##  Factor w/ 149 levels "ADMIN DELINQ PRIOR TO IDR",..: 101 101 101 101 101 76 104 104 87 87 ...
```

```
head(levels(factor(slhloans$Status)))
```

```
## [1] "ADMIN DELINQ PRIOR TO IDR"
## [2] "ADMIN FORBEARANCE"
## [3] "ADMIN PRE-HARDSHIP FORB"
## [4] "ADMINISTRATIVE FORBEARANCE"
## [5] "ADMINISTRATIVE FORBEARANCE-ENDS 03/08/2016"
## [6] "ADMINISTRATIVE FORBEARANCE-ENDS 11/15/2016"
```

This factor with 149 levels needed to be reduced to just two levels for the purposes of the desired machine learning algorithm. A borrower's loan could have one of two statuses, either in "Repayment" or in "Default". This meant first breaking down the statuses by frequency, identifying duplicate but misnamed levels, and merging levels that had similar meanings.

```
levels(factor(slhloans$Status))
```

```
## [1] "Default" "RPM"
```

3.2 Major

Cleaning the major's obtained by borrowers was necessay to undertake since the non-standardized factor had over 1,300 different entries, the majority of which were duplicative in nature (for example, one borrower's degree read "International Studies" while another's read "Int'l Studies").

```
str(slhloans$Major)
```

```
##  Factor w/ 1340 levels " health science",..: 777 777 777 777 777 777 1130 1130 483 483 ...
```

Condensing the majors to the most frequently occuring and a basket titled "others" for any other majors made this factor much easier to work with.

```
levels(slhloans$Major) <- gsub(".*Law.*|.*Lawyer.*|.*attorney.*|.*JD.*|.*Juris Doctor.*|.*Paralegal.*",
levels(slhloans$Major) <- gsub(".*Business.*|.*Financ.*|.*Account.*|.*Economic.*|.*Marketing.*|.*Human
levels(slhloans$Major) <- gsub(".*Philosophy.*|.*Social.*|.*Politic.*|.*English.*|.*History.*|.*Sociolog
levels(slhloans$Major) <- gsub(".*Engineer.*", "Engineering", levels(slhloans$Major))
levels(slhloans$Major) <- gsub(".*Medicine.*|.*MD.*|.*Physician*|.*Doctor.*|.*Pharmacy.*|.*Medical.*|.*
levels(slhloans$Major) <- gsub(".*Teach.*|.*Education.*|.*Teach.*", "Education", levels(slhloans$Major)
levels(slhloans$Major) <- gsub(".*Technology.*|.*Computer.*|.*Systems.*", "Computer Science", levels(sl
levels(slhloans$Major) <- gsub(".*Communication.*|.*Relations.*", "Communications", levels(slhloans$Maj
levels(slhloans$Major) <- gsub(".*Science.*|.*Biology.*|.*Chemistry.*|.*Geology.*|.*Bioengineer.*|.*Phys
levels(slhloans$Major) <- gsub(".*Psychology.*", "Psychology", levels(slhloans$Major))
levels(slhloans$Major) <- gsub("Art|.*Graphic.*|.*Film.*|.*Drama.*|.*Music.*|.*Photography.*", "Arts",
levels(slhloans$Major) <- gsub(".*Nurs.*|.*nurs.*", "Nursing", levels(slhloans$Major))

OtherMajors <- !(slhloans$Major %in% c("Business","Sciences", "Higher Medical Degree", "Engineering", "

slhloans$Major[OtherMajors]<- "Other"

slhloans$Major <- factor(slhloans$Major)

levels(slhloans$Major)
```

```
##  [1] "Law"                   "Business"
```

```
##  [3] "Engineering"           "Sciences"
##  [5] "Education"             "Psychology"
##  [7] "Communications"        "Higher Medical Degree"
##  [9] "Nursing"               "MBA"
## [11] "Other"
```

3.3 Profession

Cleaning the profession's of borrowers again was important since to start borrowers had self-identified into over 1800 different professions.

```
str(slhloans$Profession)
```

```
##  Factor w/ 1854 levels "","," ","    ",..: 414 414 414 414 414 414 1 1 349 349 ...
```

By first merging duplicative or identical professions (example: accountant, certified public accountant) and then analyzing the most frequently occuring professions the profession level was narrowed to ten of the top professions and a catchall category of "Other" for all other professions.

```
levels(slhloans$Profession) <- gsub(".*Lawyer.*|.*attorney.*", "Attorney", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Engineer.*", "Engineer", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Teacher.*", "Teacher", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Nurse.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Doctor.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Therapist.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Professi
levels(slhloans$Profession) <- gsub(".*Physician.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Professi
levels(slhloans$Profession) <- gsub(".*Dentist.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Professio
levels(slhloans$Profession) <- gsub(".*Dental.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Physician's.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profes
levels(slhloans$Profession) <- gsub(".*Psychologist.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profe
levels(slhloans$Profession) <- gsub(".*Psychiatrist.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profe
levels(slhloans$Profession) <- gsub(".*Medical.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Professio
levels(slhloans$Profession) <- gsub(".*Chiropractor.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profe
levels(slhloans$Profession) <- gsub(".*Pharmacist.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Profess
levels(slhloans$Profession) <- gsub(".*Nursing.*", "Doctor/Nurse/Pharmacist", levels(slhloans$Professio
levels(slhloans$Profession) <- gsub(".*Computer.*", "Computer/Tech", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Developer.*", "Computer/Tech", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*IT.*", "Computer/Tech", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Teacher.*", "Teacher", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Librarian.*", "Teacher", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Professor.*", "Teacher", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Accounting.*", "Accountant", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Student.*|.*Resident.*|.*Researcher.*|.*student.*", "Student", le
levels(slhloans$Profession) <- gsub(".*Designer.*|.*Graphic.*", "Designer", levels(slhloans$Profession))
levels(slhloans$Profession) <- gsub(".*Business.*|.*Manager.*|.*Marketing.*|.*Financial.*|.*Analyst.*|.*
levels(slhloans$Profession) <- gsub(".*Retail.*|.*Cashier.*", "Retail", levels(slhloans$Profession))

OtherProfessions <- !(slhloans$Profession %in% c("Doctor/Nurse/Pharmacist","General Business", "Enginee

slhloans$Profession[OtherProfessions]<- "Other"

slhloans$Profession <- factor(slhloans$Profession)
```

```
levels(factor(slhloans$Profession))
```

```
##  [1] "General Business"        "Teacher"
```

```
##  [3] "Doctor/Nurse/Pharmacist" "Accountant"
##  [5] "Attorney"                "Engineer"
##  [7] "Computer/Tech"           "Designer"
##  [9] "Student"                 "Retail"
## [11] "Other"
```

3.4 Other Factors

By using domain knowledge the "Servicer" factor was removed as it has no bearing on whether a borrower will default or not.

4 Exploratory Data Analysis

In examning the new cleaned data set, we can see we have a fairly unbalanced data set with the majority of borrowers in "repayment" and only a handful in "default".

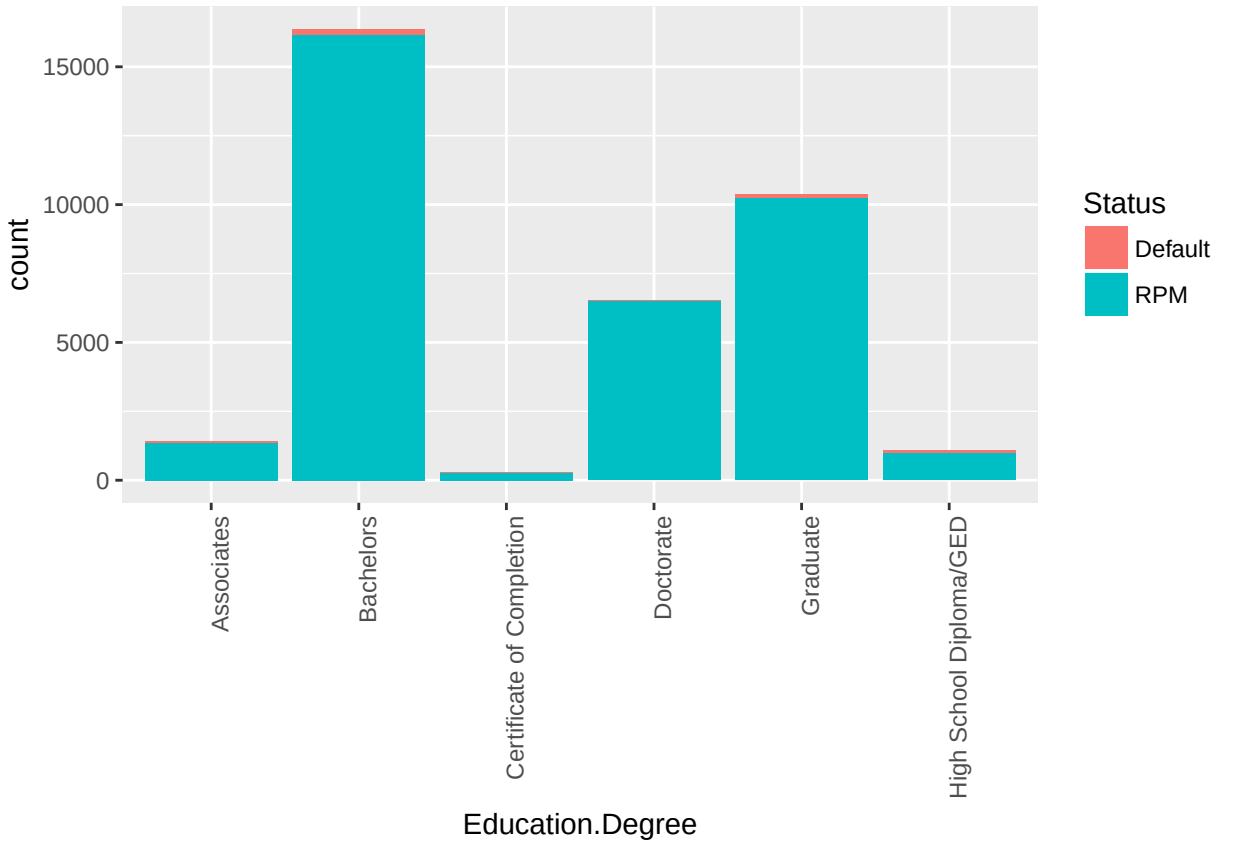
```
slhloans%>%group_by(Status)%>% dplyr::summarise(Count=n())
```

```
## # A tibble: 2 × 2
##    Status Count
##     <chr> <int>
## 1 Default   520
## 2     RPM 35553
```

[1.4% calculation]

4.1 Loans in repayment and default by degree type

```
ggplot(slhloans, aes(Education.Degree, fill=Status))+geom_bar()+theme(axis.text.x = element_text(angle =
```



In order to better understand where users are in their journey of paying off their student loan debt, a column calculating the difference between the original loan's balance and the current loan's balance was added.

```
#adding a new calculated column = difference between original and current principal

slhloans <- mutate(slhloans, Difference=Current.Principal-Original.Principal)

summary(slhloans$Difference)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -250669     -93     115    1872    1851  192648
```

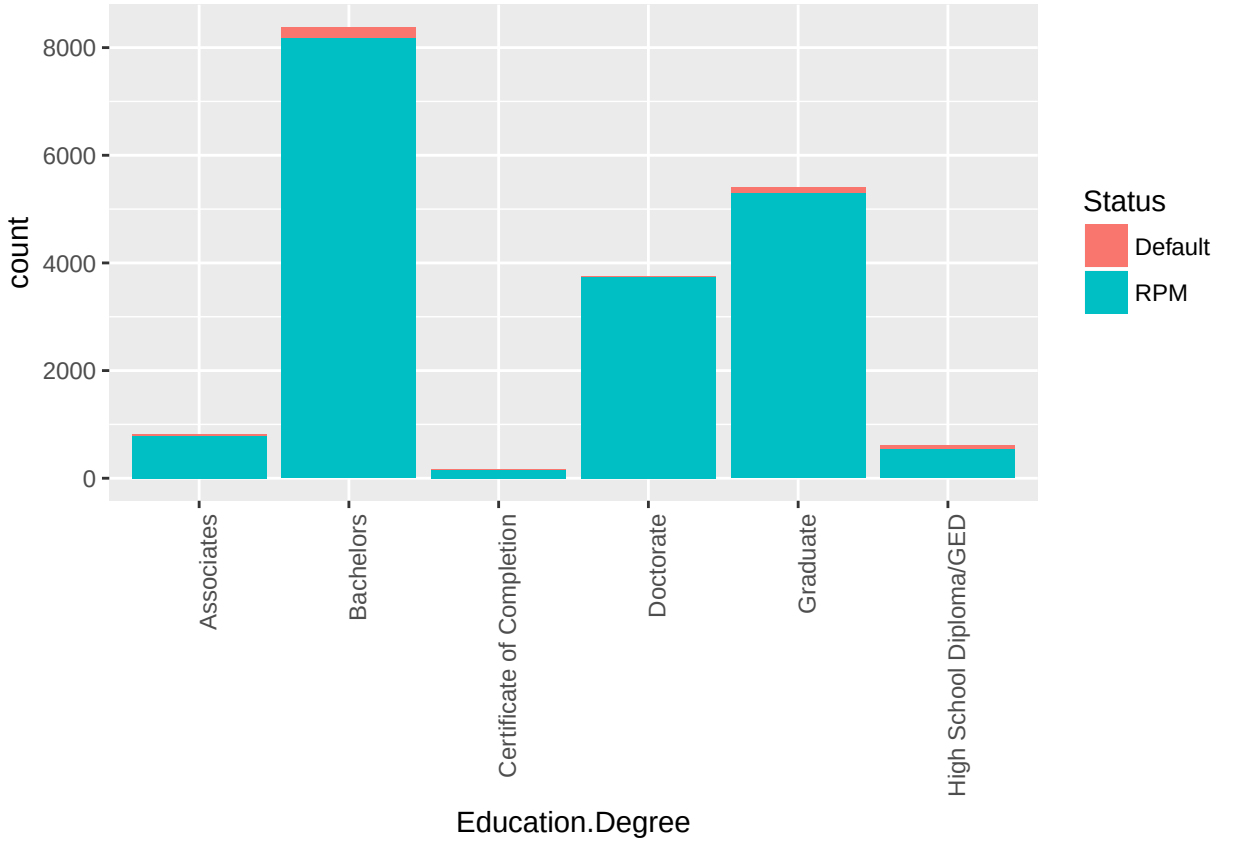
To ensure loans where there is currently a balance present are used, the previously created column "difference"

```
#Using only records where difference is positive, and rate is >0
slhloans <- filter(slhloans, Difference>=0&Rate>0)
```

To ensure only loans where there was an initial balance added by the user, any loans with an original balance of 0 were removed.

```
#getting rid of records where original principal is 0
slhloans <- filter(slhloans, Original.Principal>0)
```

```
ggplot(slhloans, aes(Education.Degree, fill=Status))+geom_bar()+theme(axis.text.x = element_text(angle =
```



Since the database where the loans was held had a non-standardized date field, using lubridate the dates were converted to a format where calculations could be performed on them.

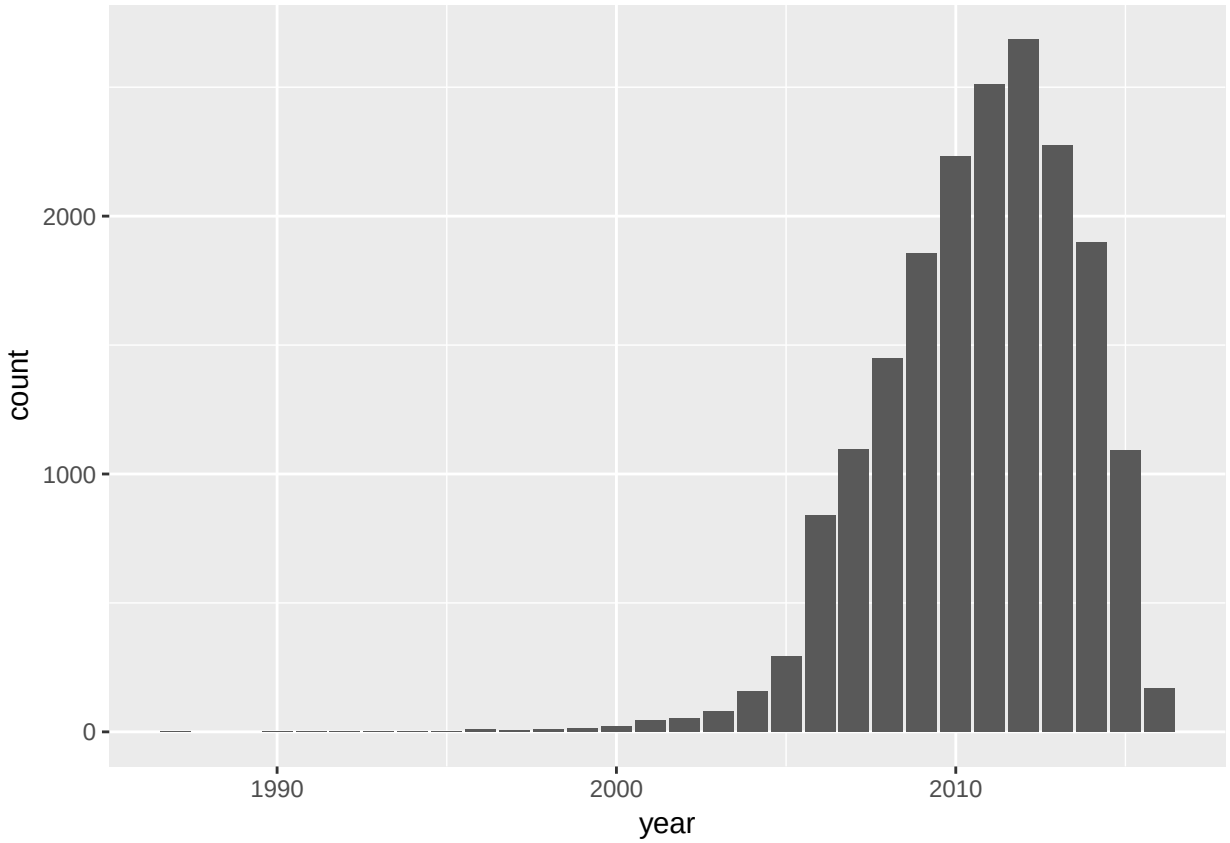
```
#Lubridating the date field from character to true date field
slhloans$Loan.Disbursement.Date <- mdy(slhloans$Loan.Disbursement.Date)
```

4.2 Loan disbursement distribution by year

Taking a look at the distribution of years when the users' loans were disbursed

```
yearwise_desc <- arrange(yearwise, desc(Count))

colnames(yearwise) <- c("year", "count")

ggplot(yearwise, (aes(x = year, y = count))) + geom_bar(stat="identity")
```

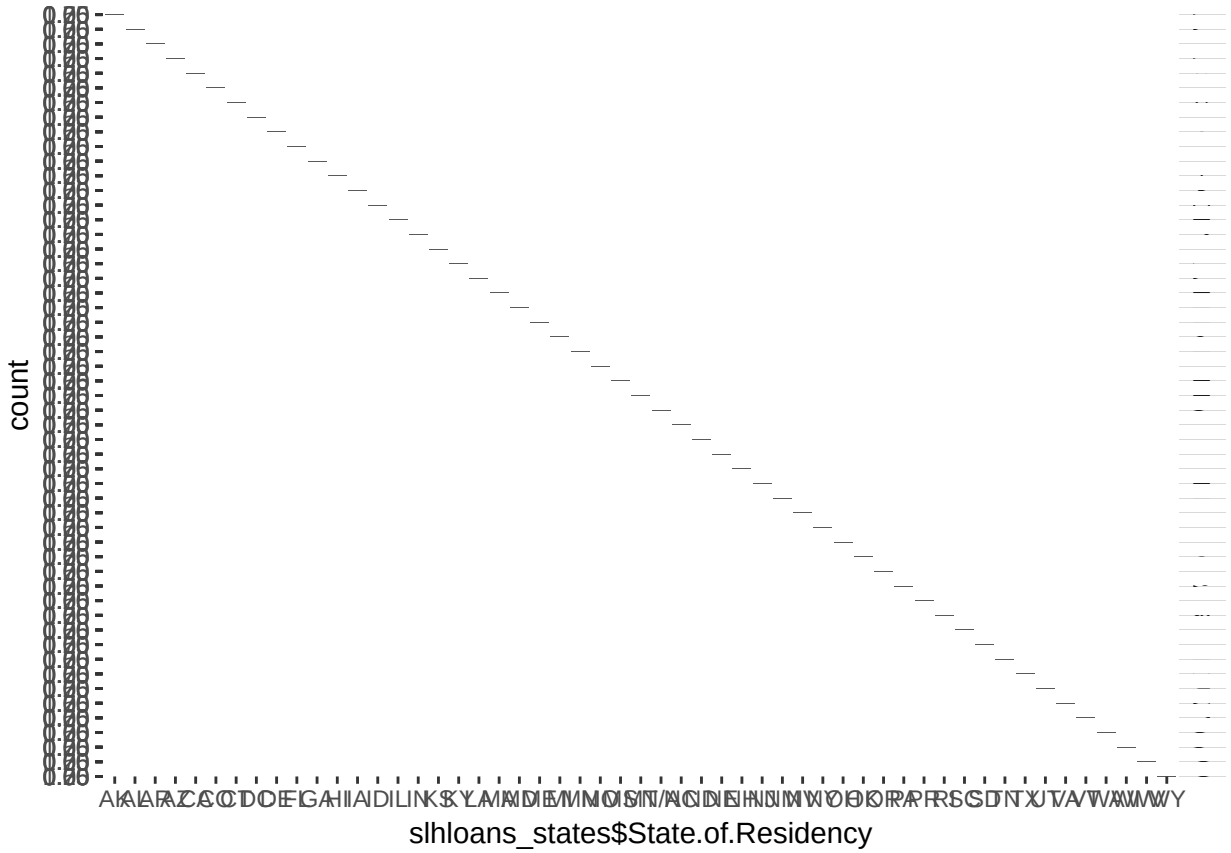
```
## Warning: Removed 1 rows containing missing values (position_stack).
```



4.3 Student Loan Hero users by state

```
##  State.of.Residency     Status
##  AK      : 1         Min.   :  14.0
##  AL      : 1         1st Qu.:  69.0
##  AR      : 1         Median : 165.0
##  AZ      : 1         Mean   : 360.9
##  CA      : 1         3rd Qu.: 495.0
##  CO      : 1         Max.   :1863.0
##  (Other):47
```

```
ggplot(slhloans_states, aes(slhloans_states$State.of.Residency)) + geom_bar() + facet_grid(slhloans_sta
```




```
#Adding new column for each user's original balance sum (since users can have multiple loans in their n
slhloans$Original.Principal[is.na(slhloans$Original.Principal)] <- 0
slhloans$Current.Principal[is.na(slhloans$Current.Principal)] <- 0

slhloansv2 <- slhloans %>% group_by(User.ID..) %>% mutate(origtotalbalance=sum(Original.Principal))
slhloansv2 <- slhloans %>% group_by(User.ID..) %>% mutate(currenttotalbalance=sum(Current.Principal))%>%

slhloansv2$progress <- mutate(slhloansv2, progress = currenttotalbalance / origtotalbalance)
```

4.4 Carnegie Mellon University Classifications

The Carnegie Mellon University Classification dataset is a framework for categorizing and classifying United States institues of higher education. All Title IV colleges and universities are listed along with accompanying classifying data including attributes such as enrollment profiles (demographics, student statuses, etc), university size and setting, any unique university characteristics such as whether the school is a technical or vocational school, historically black colleges and universities, women's only universities, etc.

```
cleanloans <- read.csv("slhloansclean.csv") #this is the post-cleaned version of "loans"
```

Once the Carnegie Mellon University Classification dataset was imported, the classifiers that were to be used for further analysis and matched up with the universities of SLH users were imported, including data such as the location (state and region) of university, type of university (is this a medical school, liberal arts school, womens school, etc).

```
univ <- read.csv("collegedetails.csv")
cleanloans$univctrl <- univ[match(cleanloans$College, univ$NAME),4]
```

```
cleanloans$univstate <- univ[match(cleanloans$College, univ$NAME),3]
cleanloans$univobereg <- univ[match(cleanloans$College, univ$NAME),5]
cleanloans$univlocale <- univ[match(cleanloans$College, univ$NAME),6]
cleanloans$univenrprofile <- univ[match(cleanloans$College, univ$NAME),8]
cleanloans$univmedical <- univ[match(cleanloans$College, univ$NAME),9]
cleanloans$univhbcu <- univ[match(cleanloans$College, univ$NAME),10]
cleanloans$univtribal <- univ[match(cleanloans$College, univ$NAME),11]
cleanloans$univhsi <- univ[match(cleanloans$College, univ$NAME),12]
cleanloans$univwomens <- univ[match(cleanloans$College, univ$NAME),14]
cleanloans$univlibarts <- univ[match(cleanloans$College, univ$NAME),15]
```

4.5 More Cleaning

```
levels(cleanloans$Major) <- gsub(".*Law.*|.*Lawyer.*|.*attorney.*|.*JD.*|.*Juris Doctor.*|.*Paralegal.*
levels(cleanloans$Major) <- gsub(".*Business.*|.*Financ.*|.*Account.*|.*Economic.*|.*Marketing.*|.*Humar
levels(cleanloans$Major) <- gsub(".*Philosophy.*|.*Social.*|.*Politic.*|.*English.*|.*History.*|.*Socio
levels(cleanloans$Major) <- gsub(".*Engineer.*", "Engineering", levels(cleanloans$Major))
levels(cleanloans$Major) <- gsub(".*Medicine.*|.*MD.*|.*Physician*|.*Doctor.*|.*Pharmacy.*|.*Medical.*|
levels(cleanloans$Major) <- gsub(".*Teach.*|.*Education.*|.*Teach.*", "Education", levels(cleanloans$Ma
levels(cleanloans$Major) <- gsub(".*Technology.*|.*Computer.*|.*Systems.*", "Computer Science", levels(
levels(cleanloans$Major) <- gsub(".*Communication.*|.*Relations.*", "Communications", levels(cleanloans$
levels(cleanloans$Major) <- gsub(".*Science.*|.*Biology.*|.*Chemistry.*|.*Geology.*|.*Bioengineer.*|.*Ph
levels(cleanloans$Major) <- gsub(".*Psychology.*", "Psychology", levels(cleanloans$Major))
levels(cleanloans$Major) <- gsub("Art|.*Graphic.*|.*Film.*|.*Drama.*|.*Music.*|.*Photography.*", "Arts"
levels(cleanloans$Major) <- gsub(".*Nurs.*|.*nurs.*", "Nursing", levels(cleanloans$Major))

OtherMajors <- !(cleanloans$Major %in% c("Business","Sciences", "Higher Medical Degree", "Engineering",

cleanloans$Major[OtherMajors]<- "Other"

cleanloans$Major <- factor(cleanloans$Major)

levels(cleanloans$Major)
```

```
##  [1] "Law"                   "Business"
##  [3] "Engineering"           "Sciences"
##  [5] "Psychology"            "Education"
##  [7] "Communications"        "Higher Medical Degree"
##  [9] "Nursing"               "MBA"
## [11] "Other"
```

```
cleanloans$Status[grep("forbearance",cleanloans$Status,ignore.case = TRUE)] <- "FBR"

levels(factor(cleanloans$Status)) #prints number of unique levels in status column
```

```
##  [1] "ADMIN DELINQ PRIOR TO IDR"  "ADMIN PRE-HARDSHIP FORB"
##  [3] "BANKRUPTCY CLAIM, ACTIVE"   "CANCELLED"
##  [5] "default"                    "DEFERRED"
##  [7] "DELINQUENT"                 "DFR"
##  [9] "Excess Debt Forb"           "FBR"
## [11] "grace"                      "Hardship Defer"
## [13] "In School"                  "IN SCHOOL"
## [15] "In School Defer"            "In school until  9/17/2023"
```

```
## [17] "Interest Only"              "LOAN ORIGINATED"
## [19] "RPM"                        "Unemployment Defer"
```

```
#further reducing duplicative levels
cleanloans$Status[grep("deferment", cleanloans$Status,ignore.case = TRUE)] <- "DFR"
cleanloans$Status[grep("repayment", cleanloans$Status,ignore.case = TRUE)] <- "RPM"
cleanloans$Status[grep("grace", cleanloans$Status,ignore.case = TRUE)] <- "grace"
cleanloans$Status[grep("default", cleanloans$Status,ignore.case = TRUE)] <- "default"

levels(cleanloans$Status) <- gsub(".*DEFERRED.*", "DFR", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*DEFERMENT.*", "DFR", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*repayment.*", "RPM", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*Grace.*", "DFR", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*default.*", "Default", ignore.case = TRUE, levels(cleanloans$Status
levels(cleanloans$Status) <- gsub(".*forbearance.*", "DFR", ignore.case = TRUE, levels(cleanloans$Status
levels(cleanloans$Status) <- gsub(".*bankruptcy.*", "Default", ignore.case = TRUE, levels(cleanloans$Sta
levels(cleanloans$Status) <- gsub(".*delinq.*", "Default", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*rpm.*", "RPM", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*defer.*", "DFR", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*Forb.*", "DFR", ignore.case = TRUE, levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub(".*In school.*", "In School", ignore.case = TRUE, levels(cleanloans$S


#levels(cleanloans$Status) <- gsub("DEFERRED", "DFR", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("LOAN ORIGINATED", "IN SCHOOL", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("FBR", "DFR", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("grace", "DFR", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("DFR", "DFR/GRC", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("DFR/GRC/GRC", "DFR/GRC", levels(cleanloans$Status))

levels(factor(cleanloans$Status)) #checking number of levels now
```

```
## [1] "Default"       "DFR/GRC"       "CANCELLED"     "In School"
## [5] "Interest Only" "IN SCHOOL"     "RPM"
```

```
#Cleaning up the Professions
head(levels(cleanloans$Profession))
```

```
## [1] ""                       " "                      " Manager (technical)"
## [4] "3D graphic simulation" "4th Grade Teacher"     "8,000"
```

```
head(sort(table(cleanloans$Profession)))
```

```
##
## Account Development Representative                 Accounting Analyst
##                                  1                                  1
##     Accounts Receivable Specialist                   Admin Assistance
##                                  1                                  1
##                      Administrative     Advertising Agency Coordinator
##                                  1                                  1
```

```
levels(cleanloans$Profession) <- gsub(".*Lawyer.*|.*attorney.*", "Attorney", levels(cleanloans$Professi
levels(cleanloans$Profession) <- gsub(".*Engineer.*", "Engineer", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Teacher.*", "Teacher", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Nurse.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Profess
levels(cleanloans$Profession) <- gsub(".*Doctor.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Profes
levels(cleanloans$Profession) <- gsub(".*Therapist.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Pro
```

```
levels(cleanloans$Profession) <- gsub(".*Physician.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Pro
levels(cleanloans$Profession) <- gsub(".*Dentist.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Profe
levels(cleanloans$Profession) <- gsub(".*Dental.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Profes
levels(cleanloans$Profession) <- gsub(".*Physician's.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$P
levels(cleanloans$Profession) <- gsub(".*Psychologist.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$
levels(cleanloans$Profession) <- gsub(".*Psychiatrist.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$
levels(cleanloans$Profession) <- gsub(".*Medical.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Profe
levels(cleanloans$Profession) <- gsub(".*Chiropractor.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$
levels(cleanloans$Profession) <- gsub(".*Pharmacist.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Pr
levels(cleanloans$Profession) <- gsub(".*Nursing.*", "Doctor/Nurse/Pharmacist", levels(cleanloans$Profe
levels(cleanloans$Profession) <- gsub(".*Computer.*", "Computer/Tech", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Developer.*", "Computer/Tech", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*IT.*", "Computer/Tech", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Teacher.*", "Teacher", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Librarian.*", "Teacher", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Professor.*", "Teacher", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Accounting.*", "Accountant", levels(cleanloans$Profession))
levels(cleanloans$Profession) <- gsub(".*Student.*|.*Resident.*|.*Researcher.*|.*student.*", "Student",
levels(cleanloans$Profession) <- gsub(".*Designer.*|.*Graphic.*", "Designer", levels(cleanloans$Profess
levels(cleanloans$Profession) <- gsub(".*Business.*|.*Manager.*|.*Marketing.*|.*Financial.*|.*Analyst.*
levels(cleanloans$Profession) <- gsub(".*Retail.*|.*Cashier.*", "Retail", levels(cleanloans$Profession)
OtherProfessions <- !(cleanloans$Profession %in% c("Doctor/Nurse/Pharmacist","General Business", "Engine

cleanloans$Profession[OtherProfessions]<- "Other"

cleanloans$Profession <- factor(cleanloans$Profession)

levels(factor(cleanloans$Profession))
```

```
##  [1] "General Business"        "Teacher"
##  [3] "Doctor/Nurse/Pharmacist" "Accountant"
##  [5] "Attorney"                "Engineer"
##  [7] "Computer/Tech"           "Designer"
##  [9] "Student"                 "Retail"
## [11] "Other"
```

```
#removing servicer column - unnecessary for predicting the chance a borrower will default
cleanloans$Servicer <- NULL
```

4.6 Segmenting Data

```
#adding a new calculated column = difference between original and current principal
cleanloans <- mutate(cleanloans, Difference=Current.Principal-Original.Principal)

summary(cleanloans$Difference)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.00      1.31    329.00   1350.17   1345.00 110108.00
```

```
#Pulling records where difference is positive, and rate is >0
cleanloans <- filter(cleanloans, Difference>=0&Rate>0)

#str(cleanloans)
```

```
#deleting variables that are null or useless
#cleanloans <- select(cleanloans, -(Created.At:Loan.ID..),-X)

#getting rid of records where original principal is 0
cleanloans <- filter(cleanloans, Original.Principal>0)

#Lubridating the date field from character to true date field
cleanloans$Loan.Disbursement.Date <- mdy(cleanloans$Loan.Disbursement.Date)

#Adding new column for each user's original balance sum
cleanloans$Original.Principal[is.na(cleanloans$Original.Principal)] <- 0
cleanloans <- cleanloans %>% group_by(User.ID..) %>% mutate(origtotalbalance=sum(Original.Principal))
#Adding new column for each user's current balance sum
cleanloans$Current.Principal[is.na(cleanloans$Current.Principal)] <- 0
cleanloans <- cleanloans %>% group_by(User.ID..) %>% mutate(currenttotalbalance=sum(Current.Principal))


#NEW R CHUNK?
#setup column for is current principal > original principal

#change borrower home state to obereg to match the university state obereg

levels(cleanloans$State.of.Residency) <- gsub("VT|CT|ME|MA|NH|RI", "1", levels(cleanloans$State.of.Resi
levels(cleanloans$State.of.Residency) <- gsub("DE|DC|MD|NJ|NY|PA", "2", levels(cleanloans$State.of.Resi
levels(cleanloans$State.of.Residency) <- gsub("IL|IN|MI|OH|WI", "3", levels(cleanloans$State.of.Residen
levels(cleanloans$State.of.Residency) <- gsub("IA|KS|MN|MO|NE|ND|SD", "4", levels(cleanloans$State.of.R
levels(cleanloans$State.of.Residency) <- gsub("AL|AR|FL|GA|KY|LA|MS|NC|SC|TN|VA|WV", "5", levels(cleanl
levels(cleanloans$State.of.Residency) <- gsub("AZ|NM|OK|TX", "6", levels(cleanloans$State.of.Residency)
levels(cleanloans$State.of.Residency) <- gsub("CO|ID|MT|UT|WY", "7", levels(cleanloans$State.of.Residen
levels(cleanloans$State.of.Residency) <- gsub("AK|CA|HI|NV|OR|WA", "8", levels(cleanloans$State.of.Resi
levels(cleanloans$State.of.Residency) <- gsub("AS|FM|GU|MH|MP|PR|PW|VI", "9", levels(cleanloans$State.o

#Cleaning up Loan Names to just 2 type: Federal Loans or Private Loans
head(levels(cleanloans$Name))
```

```
## [1] "  \tCitiAssistÃ® Undergraduate Loan"
## [2] " PRIVATE STUDENT LOAN CC0001"
## [3] " Sallie Mae Smart Option Student Loan #0709"
## [4] " Sallie Mae Smart Option Student Loan #2441"
## [5] " Sallie Mae Smart Option Student Loan #7004"
## [6] " Sallie Mae Smart Option Student Loan #8245 "
```

```
levels(cleanloans$Name) <- gsub(".*Private.*", "Private", levels(cleanloans$Name))
levels(cleanloans$Name) <- gsub(".*Fargo.*", "Private", levels(cleanloans$Name))
levels(cleanloans$Name) <- gsub(".*Sallie.*", "Private", levels(cleanloans$Name))
levels(cleanloans$Name) <- gsub(".*Stafford.*|.*STAFFORD.*|.*stafford.*", "Federal", levels(cleanloans$
levels(cleanloans$Name) <- gsub(".*Direct.*|.*direct.*|.*DIRECT.*", "Federal", levels(cleanloans$Name))
levels(cleanloans$Name) <- gsub(".*Federal.*|.*federal.*|.*FEDERAL.*", "Federal", levels(cleanloans$Name
levels(cleanloans$Name) <- gsub(".*Signature.*|.*signature.*|.*SIGNATURE.*", "Federal", levels(cleanloan
levels(cleanloans$Name) <- gsub(".*subsidized.*|.*Subsidized.*|.*SUBSIDIZED.*", "Federal", levels(clean
levels(cleanloans$Name) <- gsub(".*PLUS.*|.*plus.*|.*Plus.*", "Federal", levels(cleanloans$Name))
levels(cleanloans$Name) <- gsub(".*FFEL.*|.*ffel.*|.*Ffel.*", "Federal", levels(cleanloans$Name))
levels(cleanloans$Name) <- gsub(".*CONSOLIDATION.*|.*consolidation.*|.*Consolidation.*", "Federal", leve
```

```
OtherLoans <- !(cleanloans$Name %in% c("Private","Federal"))

cleanloans$Name[OtherLoans]<- "Private"

cleanloans$Name <- factor(cleanloans$Name)

levels(factor(cleanloans$Name))
```

```
## [1] "Private" "Federal"
```

```
sort(table(cleanloans$Name))
```

```
##
## Private Federal
##    1548   16959
```

```
#Reducing Loan Statuses to just two
table(cleanloans$Status)
```

```
##
##       Default       DFR/GRC     CANCELLED     In School Interest Only
##           396          4482             1           277            24
##     IN SCHOOL           RPM
##          1740         11587
```

```
levels(cleanloans$Status) <- gsub("DFR/GRC", "Repayment", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("CANCELLED", "Repayment", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("In School", "Repayment", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("Interest Only", "Repayment", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("IN SCHOOL", "Repayment", levels(cleanloans$Status))
levels(cleanloans$Status) <- gsub("RPM", "Repayment", levels(cleanloans$Status))
```

4.7 Next Data Points

Now that we've

```
cleanloans$User.DOB <- mdy(cleanloans$User.DOB)
cleanloans$User.ID.. <- as.factor(cleanloans$User.ID..)
cleanloans$Loan.Disbursement.Date <- mdy(cleanloans$Loan.Disbursement.Date)
```

```
## Warning: All formats failed to parse. No formats found.
```

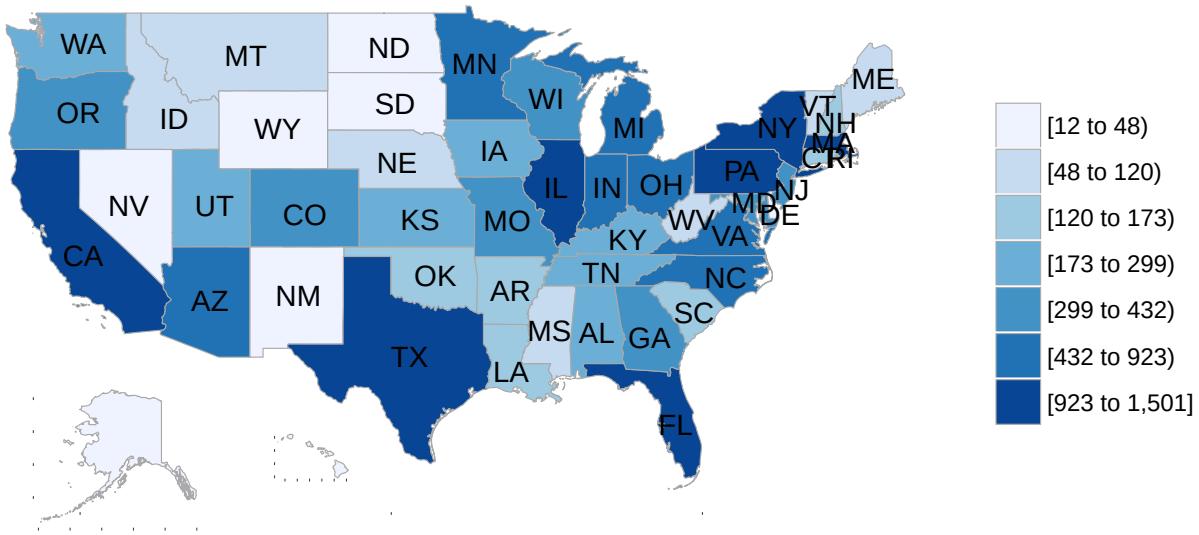
```
cleanloans$Joint.Federal.Income.Tax. <- as.character(cleanloans$Joint.Federal.Income.Tax.)

#In order to use choropleth maps, I need the state names
data("state.regions")

#Loans per state
state_loans <- cleanloans%>%group_by(abb=univstate)%>%dplyr::summarize(value=n())
state_loans <- state_loans[complete.cases(state_loans),]
state_loans <- left_join(state_loans, state.regions, by='abb')
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

```
state_choropleth(state_loans, title="Loans Per State")
```

```
## Warning in super$initialize(map.df, user.df): Your data.frame contains the
## following regions which are not mappable: NA
```

Loans Per State



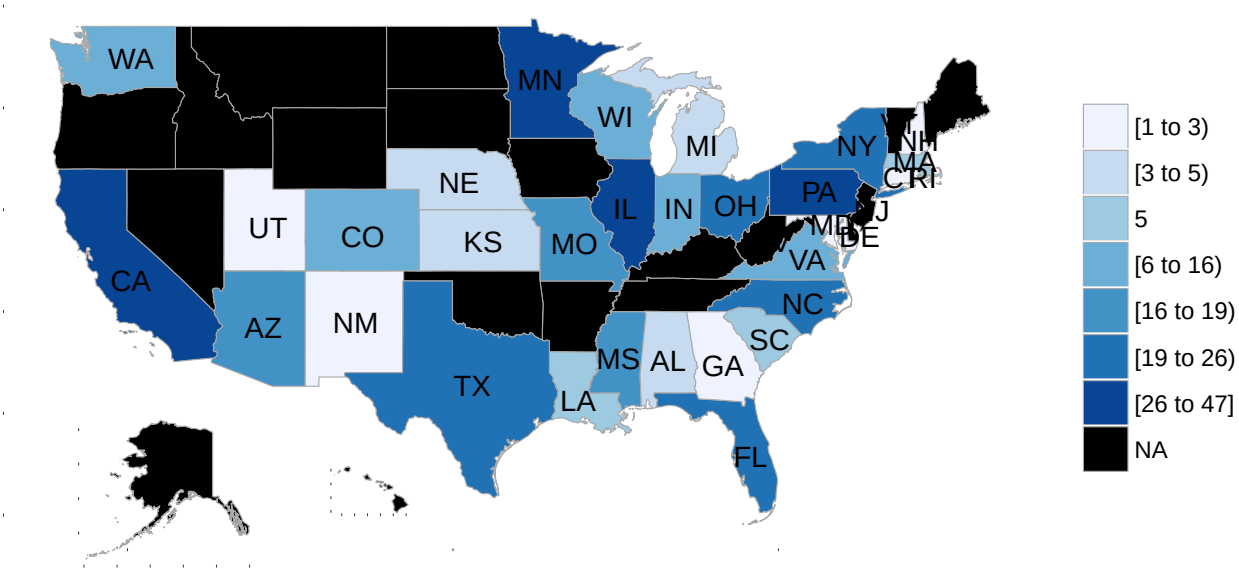

```
#Defaulted Loans as per status
status <- 'Default'
state_loans <- cleanloans%>%filter(Status==status)%>%group_by(abb=univstate)%>%dplyr::summarize(value=n
state_loans <- state_loans[complete.cases(state_loans),]
state_loans <- left_join(state_loans, state.regions, by='abb')
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

```
state_choropleth(state_loans,title=paste0("Loans As Per Status:",status))
```

```
## Warning in self$bind(): The following regions were missing and are being
## set to NA: arkansas, montana, north dakota, oklahoma, tennessee, delaware,
## west virginia, wyoming, alaska, idaho, new jersey, vermont, iowa, kentucky,
## maine, nevada, oregon, south dakota, hawaii
```

Loans As Per Status:Default



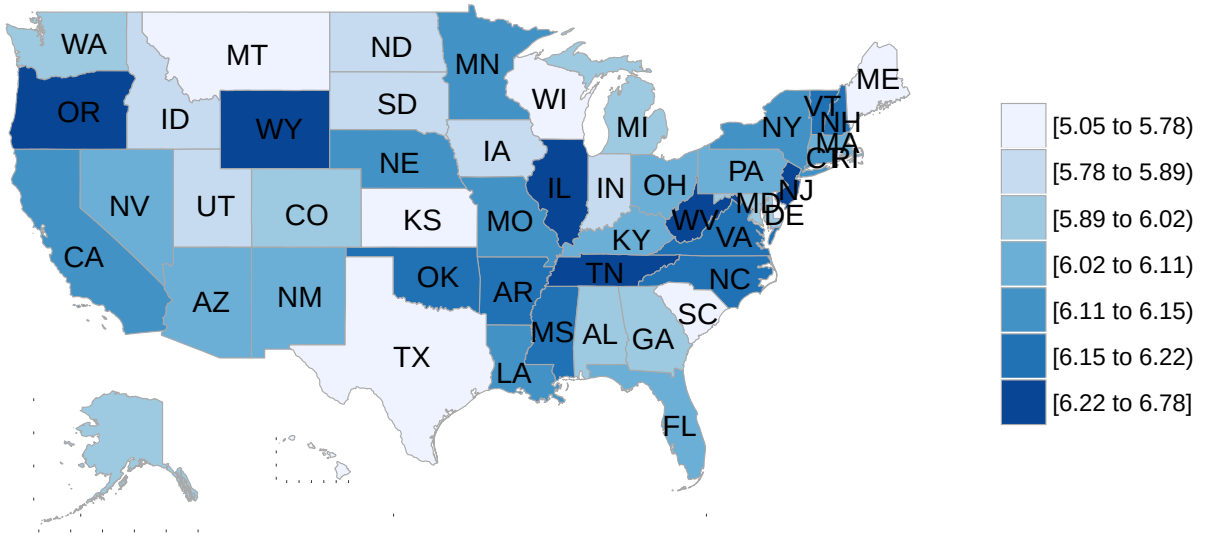

```
#Loans as per rate of interest
state_loans <- cleanloans%>%group_by(abb=univstate)%>%dplyr::summarize(value=mean(Rate))
state_loans <- state_loans[complete.cases(state_loans),]
state_loans <- left_join(state_loans, state.regions, by='abb')
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

```
state_choropleth(state_loans,title="Average Rate of Interest")
```

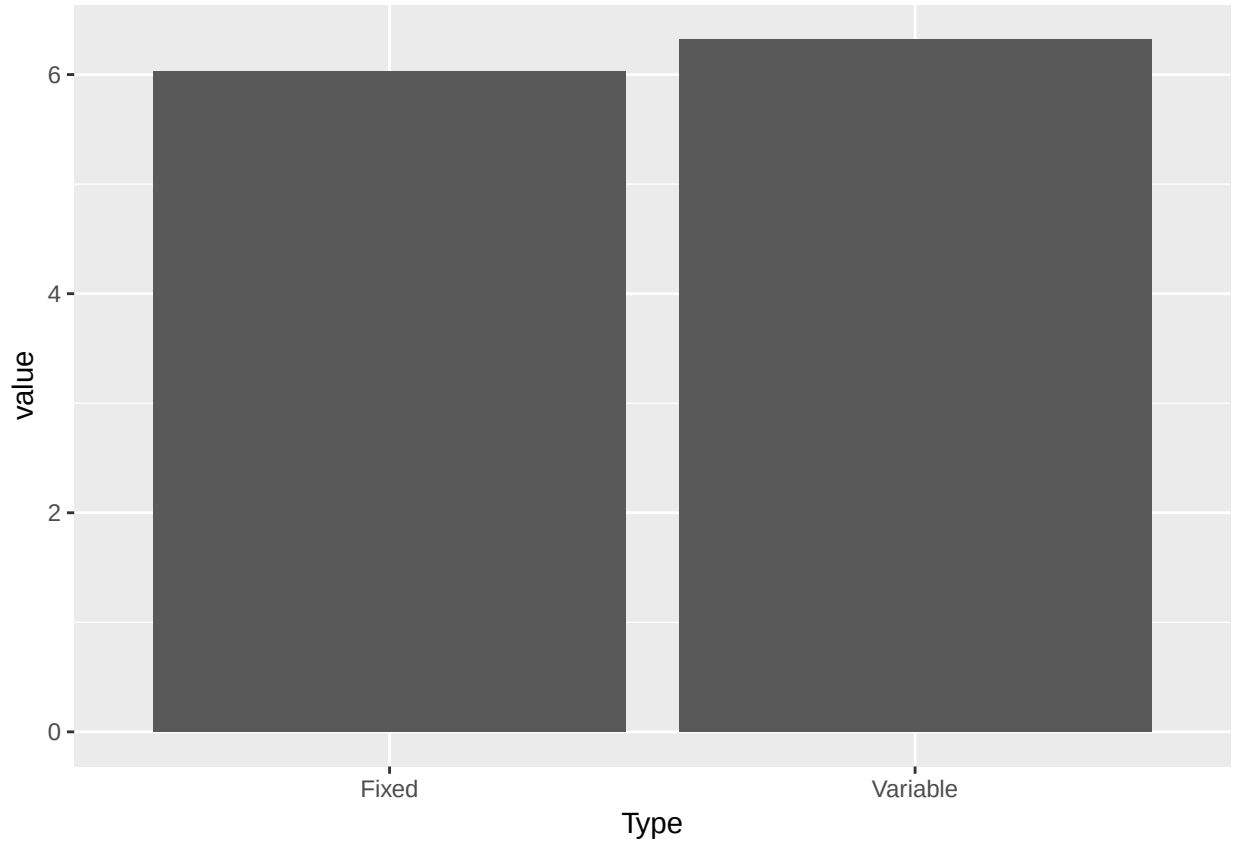
```
## Warning in super$initialize(map.df, user.df): Your data.frame contains the
## following regions which are not mappable: NA
```

Average Rate of Interest

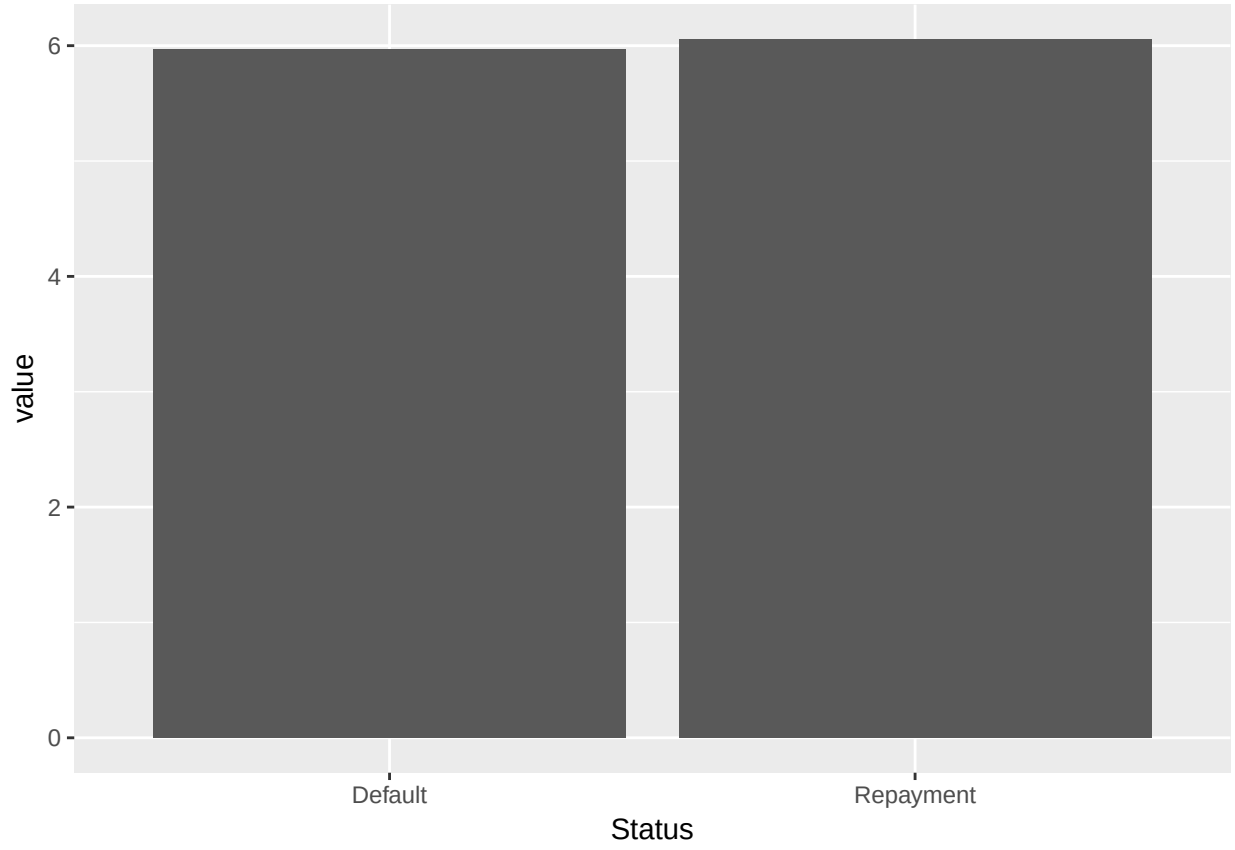


TEXT HERE

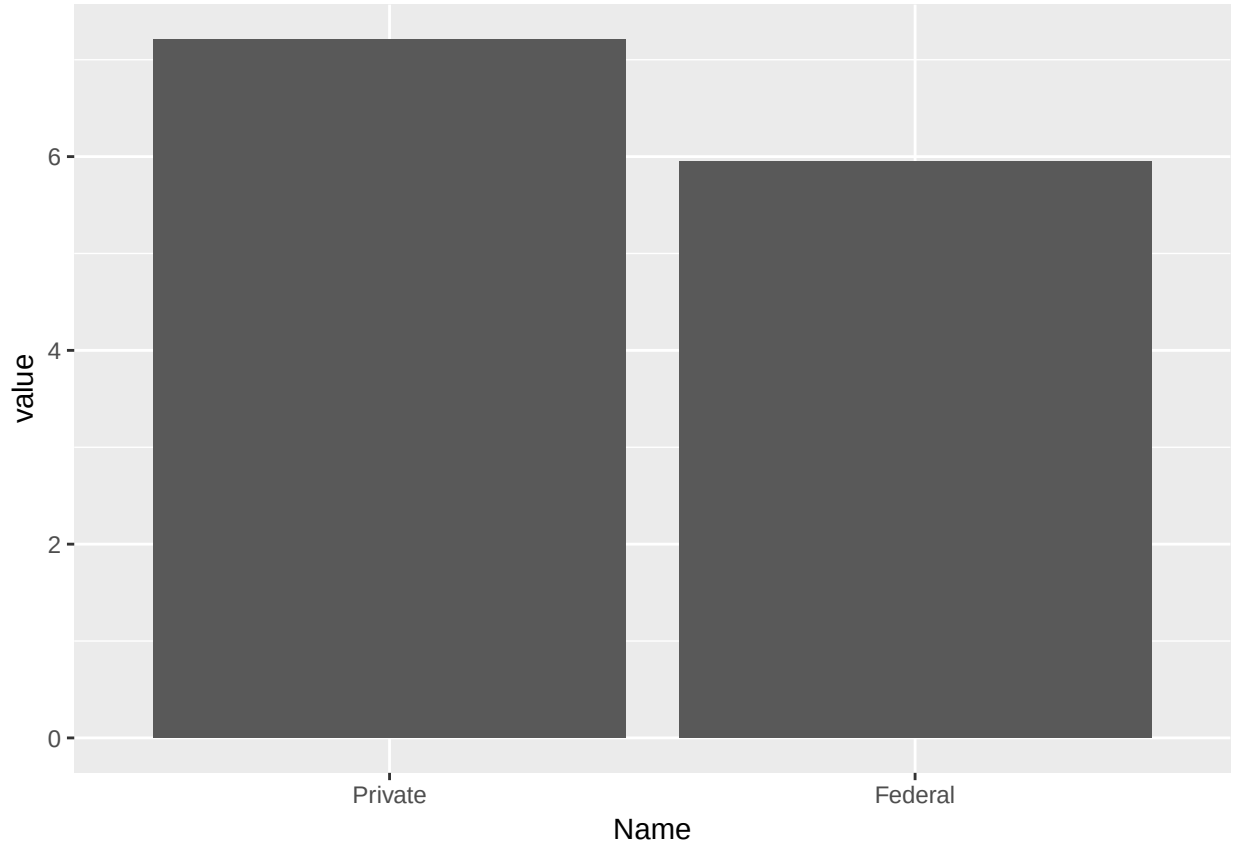
```
#AVG INTEREST RATES by Fixed vs Variable Interest Rates
#Type <- 'Fixed'
loan_rates <- cleanloans%>%group_by(Type)%>%dplyr::summarize(value=mean(Rate))

ggplot(loan_rates, (aes(x = Type, y = value))) + geom_bar(stat="identity")
```



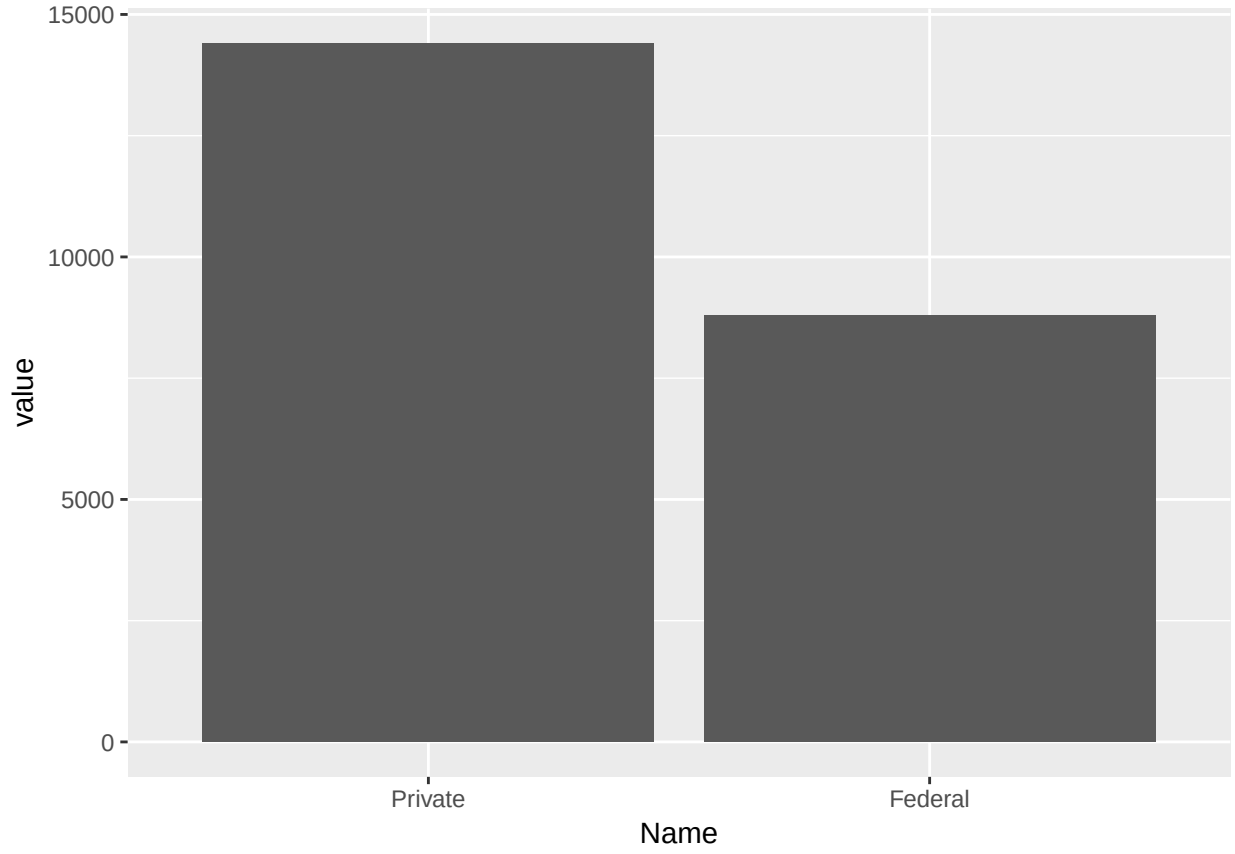

```
#AVG INTEREST RATES by Status
loan_rates_status <- cleanloans%>%group_by(Status)%>%dplyr::summarize(value=mean(Rate))

ggplot(loan_rates_status, (aes(x = Status, y = value))) + geom_bar(stat="identity")
```



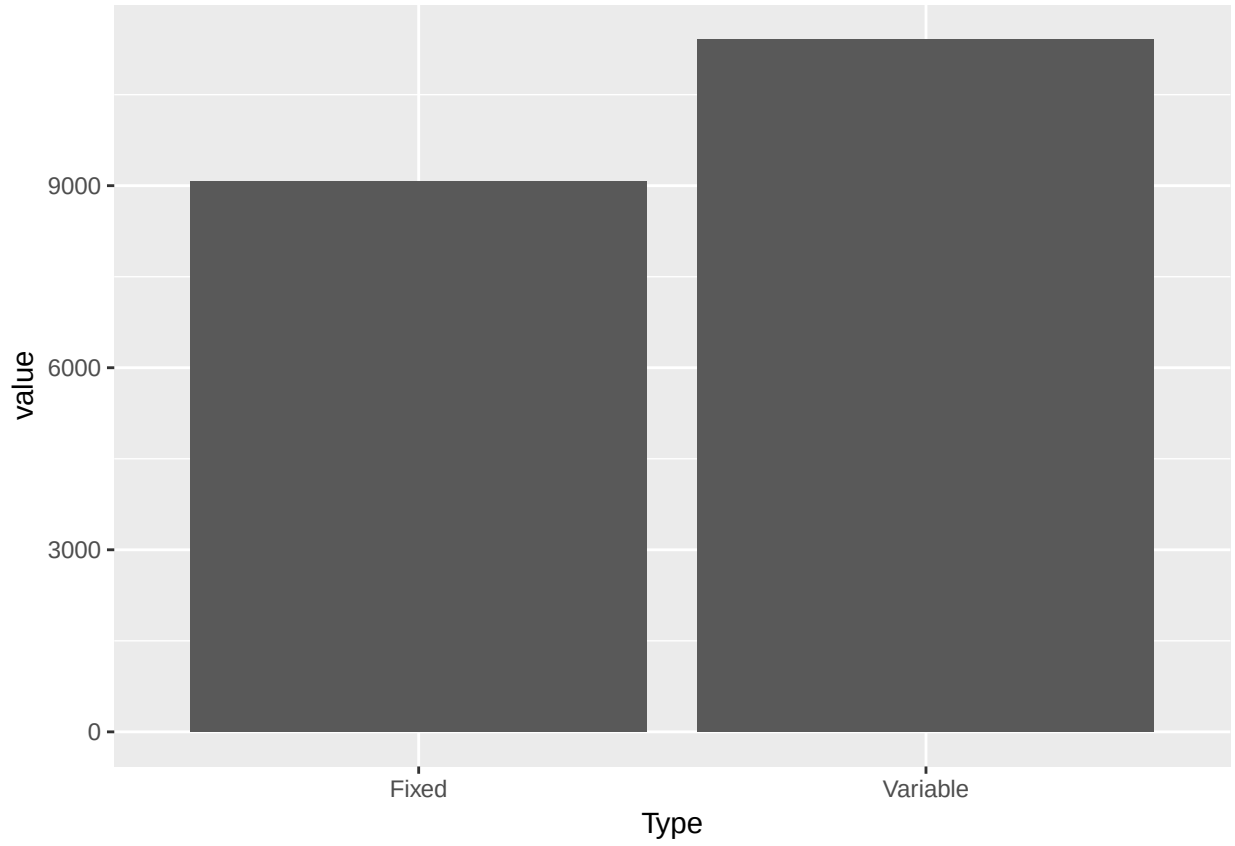

```
#AVG INTEREST RATES by Loan Type (Federal or Private)
loan_rates_name <- cleanloans%>%group_by(Name)%>%dplyr::summarize(value=mean(Rate))

ggplot(loan_rates_name, (aes(x = Name, y = value))) + geom_bar(stat="identity")
```



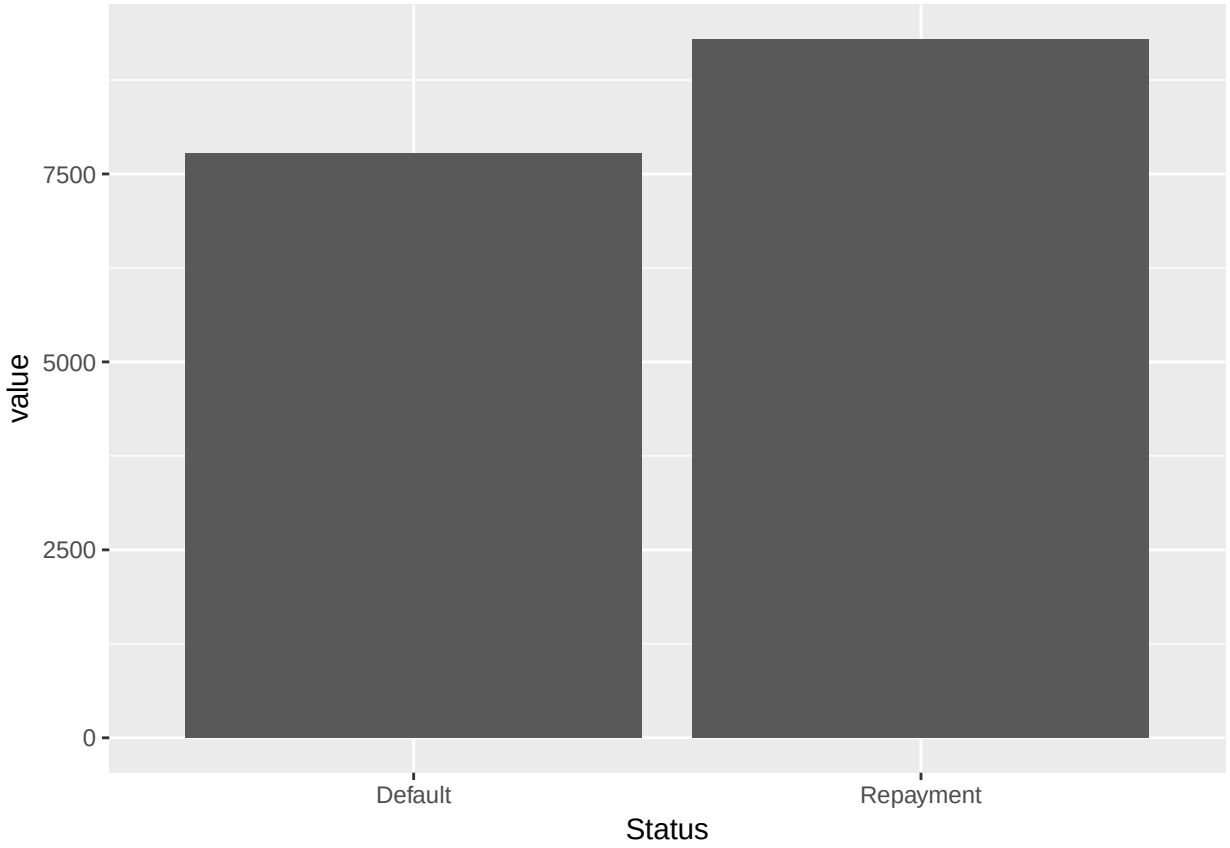

```
#AVG BALANCE by Loan Type (Federal or Private)
loan_balance_name <- cleanloans%>%group_by(Name)%>%dplyr::summarize(value=mean(Original.Principal))

ggplot(loan_balance_name, (aes(x = Name, y = value))) + geom_bar(stat="identity")
```



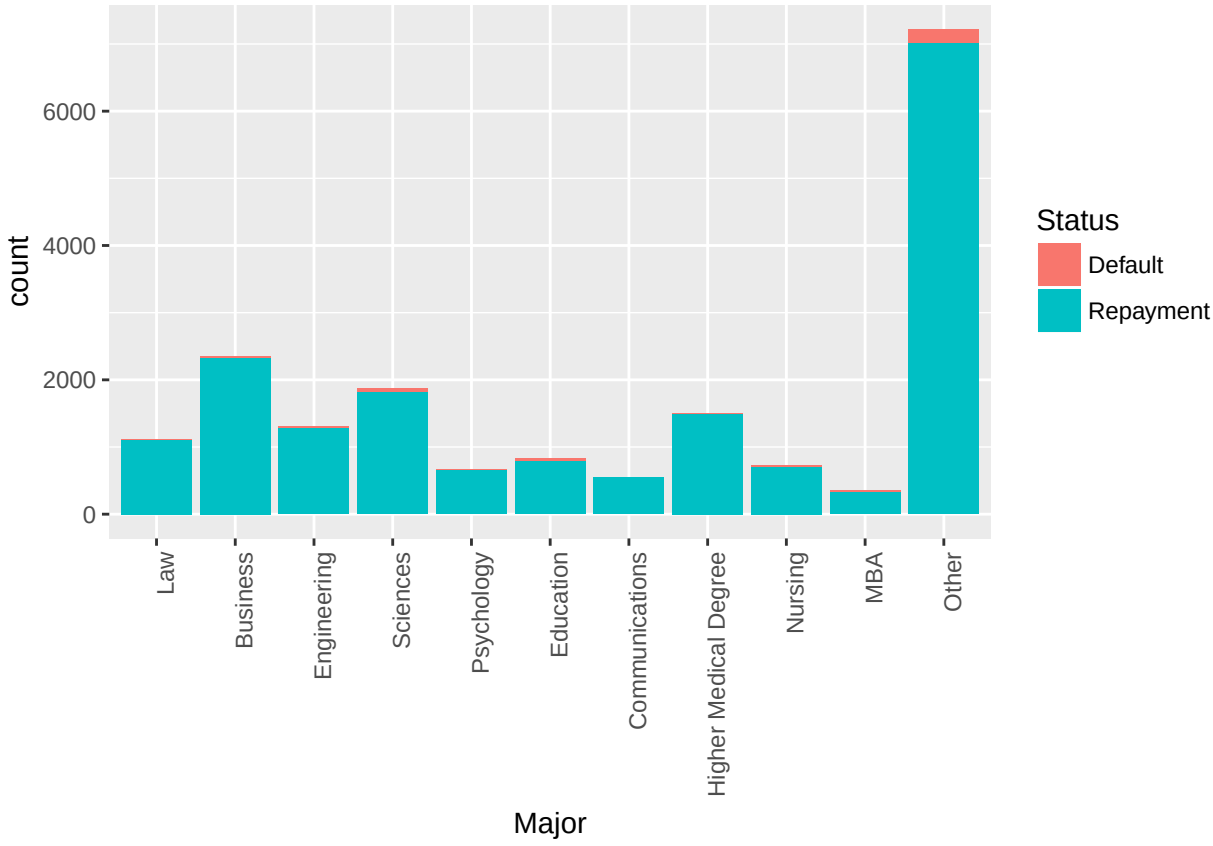

```
#AVG BALANCE by Rate Type (Fixed vs Variable)
loan_balance_type <- cleanloans%>%group_by(Type)%>%dplyr::summarize(value=mean(Original.Principal))

ggplot(loan_balance_type, (aes(x = Type, y = value))) + geom_bar(stat="identity")
```



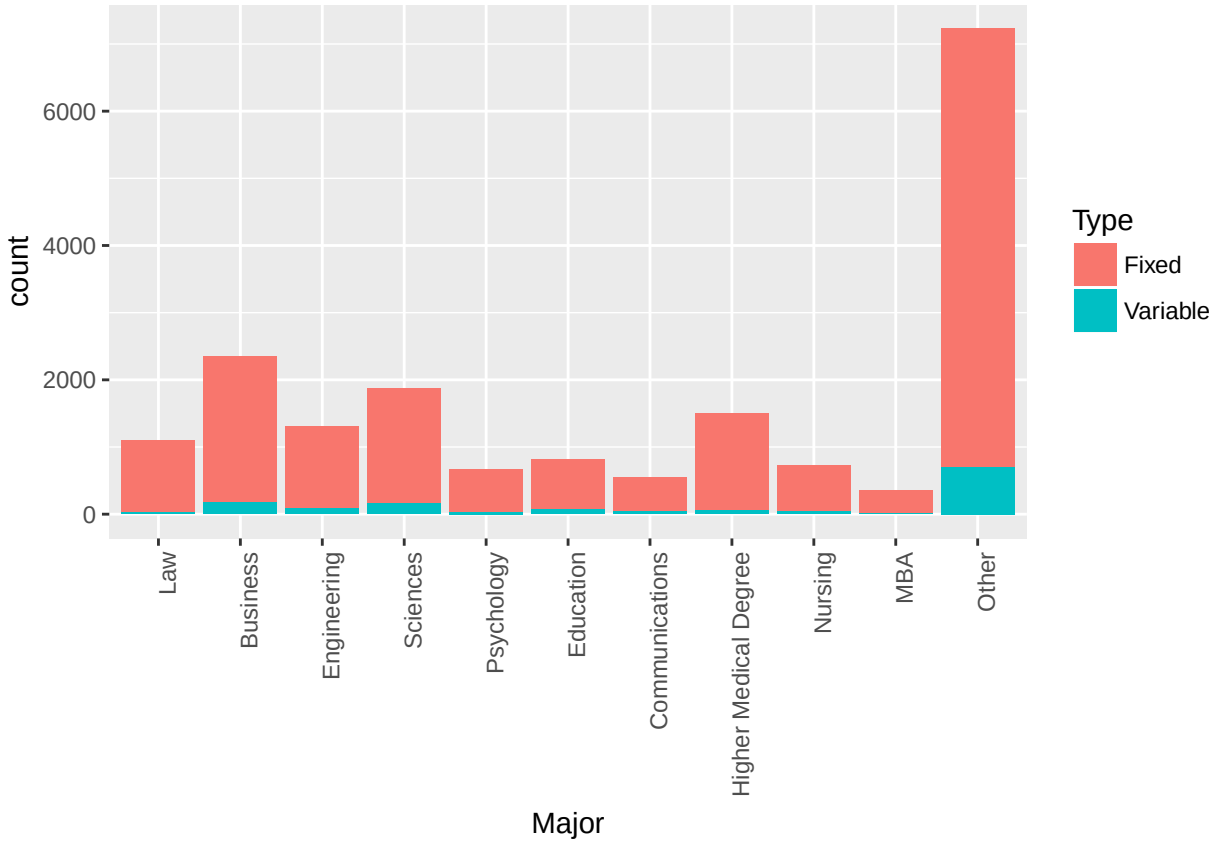

```
#AVG BALANCE by Status
loan_balance_status <- cleanloans%>%group_by(Status)%>%dplyr::summarize(value=mean(Original.Principal))

ggplot(loan_balance_status, (aes(x = Status, y = value))) + geom_bar(stat="identity")
```




```
#Status by Majors
ggplot(cleanloans, aes(Major, fill=Status))+geom_bar()+theme(axis.text.x = element_text(angle = 90, hju
```




```
#Fixed/Variable Loans by Major
ggplot(cleanloans, aes(Major, fill=Type))+geom_bar()+theme(axis.text.x = element_text(angle = 90, hjust
```



5 Predicting Default Rates

5.1 Machine Learning Algorithm - Random Forest

The goal of our default prediction model was to determine whether a given borrower would fall into one of two statuses, the "Default" status or the "Repayment" tatus. In determining the best machine learning algorithm to apply to our classifier model Random Forest was chosen for its

5.2 Variable Importance

6 Acknowledgements

I'd like to thank Dhiraj Khanna for his sustained and continued support in preparing this project and report and for the numerous suggestions surrounding how to work with unbalanced data sets.