Illinois Small Business Loan Analysis

Contents

Abstract

The purpose of this analysis is to find out what are the significant factors that lead to default in the state of Illinois. We use the SBA Loans dataset to find out those factors. In this report, we focus on seven predictors that we think are significant to predict the loan status. We use descriptive statistics, data visualization, and logistic regression to get a brief idea about the related factors.

Introduction

As a bank, one main source of income is to grant loans to customers and get interests and principals back. Granting loans to small businesses with high potential is a great way to generate this kind of income. In order to maintain a reliable source of income, banks need to determine the credibility of the small businesses. How should bank determine it?

An organization called "Small Business Association (SBA)" can serve as a source. The organization is founded in 1953 and is working with banks to provide loans guaranteed by SBA to support small businesses.

The SBA Loans Data contains small business loans information from 1987-2014. The dataset includes 899,164 observations and 27 variables. The variables contain important information such as names of the businesses, their locations, term of the loans, the condition of the businesses, disbursement amount, loan default status, default amount, whether the business has gone through the Great Recession, and the industry of the business. The data is provided by US Small Business Administration at www.sba.gov.

In our report, we focus on the loan status of Illinois, which is the state we study and live in. We are interested to see how small businesses perform in relate to their loan status. We choose 7 variables as the predictors that we think are significant to whether the loan status. The selection is based on our prior experience and knowledge. The predictors are as follow: the business condition when granted the loan (NewExist), area of the small business (UrbanRural), whether the loan was active during the Great Recession (Recession), industry of the business (Industry), term of the loan (term), SBA guaranteed portion of the total loan (SBAGuaranteedPortion) and gross disbursement amount (DisbursementGross). We are using descriptive statistics, data visualizations and logistic model to see if the variables are important to our loan status, which is the "MIS_Status" variable in our dataset.

```r
library(readr)

sba <- read_csv("https://uofi.box.com/shared/static/vi37omgitiaa2yyplrom779qvwk1g14x.csv",
    col_types = cols(ApprovalDate = col_date(format = "%d-%b-%y"),
        BalanceGross = col_number(), ChgOffDate = col_date(format = "%d-%b-%y"),
```

```r
        ChgOffPrinGr = col_number(), DisbursementDate = col_date(format = "%d-%b-%y"),
        DisbursementGross = col_number(), GrAppv = col_number(),
        SBA_Appv = col_number()))
colnames(sba) <- tolower(colnames(sba))
```

Analysis

Data Cleaning

Before the analysis, we did a few data cleanings with our dataset. We subset our data to only include "IL". Besides, we remove the missing values because they are relatively small compared to the whole dataset and it's hard for us to trace the actual value of the missing ones. We also change some of the variables to factors so that they are more meaningful for our analysis, including NewExist, UrbanRural, and Industry.

Recession and SBAGuaranteedPortion were not in the original dataset. We created the dummy variable Recession based on the disbursement and term. If the loan exists for at least one month during the Great Recession (December 2007 to June 2009), Recession will show as 1, otherwise Recession will be 0. SBAGuaranteedPortion is calculated by SBA Approved Amount divided by Total Approved Amount.]

After the above processes, we come up with a clean dataset with 19,200 observations and 8 variables.

```r
library(tidyverse)
# head(sba)
sba_IL = sba[sba$state == "IL", ]
# summary(sba_IL)
colSums(is.na(sba_IL))  # check which variables have NA values
```

```
##     loannr_chkdgt              name              city             state               zip
##                14                15                16                14                14
##              bank         bankstate             naics      approvaldate        approvalfy
##               110               112                14                14                15
##              term             noemp          newexist         createjob       retainedjob
##                14                14                19                14                14
##     franchisecode        urbanrural         revlinecr            lowdoc        chgoffdate
##                14                14               194                57             22760
##  disbursementdate disbursementgross      balancegross        mis_status      chgoffpringr
##               139                14                14                89                14
##            grappv          sba_appv
##                14                14
```

```r
sba_IL = sba_IL[!is.na(sba_IL$newexist), ]
sba_IL = sba_IL[!is.na(sba_IL$mis_status), ]
sba_IL = sba_IL[!is.na(sba_IL$disbursementdate), ]
# re-check which variables have NA values; NA values only in
# the variables that we are not using, so ignore those
# variables
colSums(is.na(sba_IL))
```

```
##     loannr_chkdgt              name              city             state               zip
##                 0                 1                 2                 0                 0
##              bank         bankstate             naics      approvaldate        approvalfy
##                95                97                 0                 0                 0
##              term             noemp          newexist         createjob       retainedjob
##                 0                 0                 0                 0                 0
##     franchisecode        urbanrural         revlinecr            lowdoc        chgoffdate
```

```
##                 0                 0               180                40             22593
##  disbursementdate disbursementgross      balancegross        mis_status      chgoffpringr
##                 0                 0                 0                 0                 0
##            grappv          sba_appv
##                 0                 0
```

```r
# NewExist should only have values of 1 and 2, clean up
# levels of 0.  NewExist = 1 means business is new, exists
# less than or equal to 2 years; NewExist = 2 means business
# is existing for more than 2 years
sba_IL = sba_IL[sba_IL$newexist != 0, ]

# UrbanRural should only have values of 1 and 2, clean up
# levels of 0.  UrbanRural = 1 means Urban; Urbanrural = 2
# means Rural
sba_IL = sba_IL[sba_IL$urbanrural != 0, ]

# Group industries based on the first two digits of NAICS
# codes
sba_IL$naics2 = floor(sba_IL$naics/10000)
table(sba_IL$naics2)  # check if there's any non-exist NAICS codes
```

```
## 
##    0   11   21   22   23   31   32   33   42   44   45   48   49   51   52   53   54   55 
##  567   33   16   15 1600  243  474  948 1222 2128  944 1002   80  364  323  449 2071    1 
##   56   61   62   71   72   81   92 
##  905  240 1404  427 2118 2188    5
```

```r
sba_IL = sba_IL[sba_IL$naics2 != 0, ]  # 0 is not a code so that we need to remove it
sba_IL$naics3 = as.factor(sba_IL$naics2)
levels(sba_IL$naics3)
```

```
##  [1] "11" "21" "22" "23" "31" "32" "33" "42" "44" "45" "48" "49" "51" "52" "53" "54" "55"
## [18] "56" "61" "62" "71" "72" "81" "92"
```

```r
# 31-33: Manufacturing; 44-45: Retail trade; 48-49:
# Transportation and warehousing Remap levels to reflect
# industries
industry = c("Agriculture, forestry, fishing & hunting", "Mining, quarrying, & oil & gas extraction",
    "Utilites", "Construction", rep("Manufacturing", 3), "Wholesale trade",
    rep("Retail trade", 2), rep("Transportation & warehousing",
        2), "Information", "Finance & insurance", "Real estate & rental & leasing",
    "Professional, scientific, & technical services", "Mgmt of companies & enterprises",
    "Admin & support & waste mgmt & remediation services", "Educational services",
    "Health care & social assistance", "Arts, entertainment, & recreation",
    "Accommodation & food services", "Other services (except public admin)",
    "Public admin")
levels(sba_IL$naics3) = c(industry)

# Create dummy variable for Recession Recession is identified
# as the loan exists at least one month during the Great
# Recession (12-01-2007 to 06-30-2009) Recession = 1 means
# loan is active during recession; Recession = 0 means loan
# is inactive during recession Used identification method
# from the website to define recession or not: The loans that
# were coded as "Recession=1" include those that were active
```

```r
# for at least a month during the Great Recession time frame.
# This was calculated by adding the length of the loan term
# in days to the disbursement date of the loan.  The coding
# in SAS for this is: Recession=0; daysterm=Term*30;
# xx=DisbursementDate+daysterm; if xx ge '1DEC2007'd AND xx
# le '30JUN2009'd then Recession=1.

daysterm = sba_IL$term * 30
sba_IL$disbursement.30 = sba_IL$disbursementdate + daysterm
sba_IL$recession = with(sba_IL, ifelse(disbursement.30 >= "2007-12-01" &
    disbursement.30 <= "2009-06-30", 1, 0))
nrow(sba_IL[sba_IL$recession == 0, ])
```

```
## [1] 17322
```

```r
nrow(sba_IL[sba_IL$recession == 1, ])
```

```
## [1] 1878
```

```r
# data reorganize and cleaning
sba_clean = data.frame(MIS_Status = factor(sba_IL$mis_status,
    levels = c("CHGOFF", "P I F"), labels = c("ChgOff", "PIF")),
    DisbursementGross = sba_IL$disbursementgross, Term = sba_IL$term,
    NewExist = factor(sba_IL$newexist, levels = c(1, 2), labels = c("New",
        "Existing")), SBAGuaranteedPortion = round(sba_IL$sba_appv/sba_IL$grappv,
        2), UrbanRural = factor(sba_IL$urbanrural, levels = c(1,
        2), labels = c("Urban", "Rural")), Recession = factor(sba_IL$recession,
        levels = c(0, 1), labels = c("Inactive", "Active")),
    Industry = sba_IL$naics3)
```

Descriptive Statistics

We provide the descriptive statistics for all the variables so that we can get an overall understanding about the data. We also compare the default rate of Illinois with the overall default rate of the U.S. to see how Illinois small businesses perform. We find out that Illinois does not perform so well because it has a much larger default rate (28.9%) compared to overall default rate of the U.S. (17.6%).

```r
# Descriptive Statistics
library(tidyverse)

dim(sba_clean)
```

```
## [1] 19200     8
```

```r
summary(sba_clean)
```

```
##   MIS_Status    DisbursementGross      Term            NewExist     SBAGuaranteedPortion
##  ChgOff: 5544   Min.   :   4000   Min.   :  0.00   New     :13049   Min.   :0.260
##  PIF   :13656   1st Qu.:  33404   1st Qu.: 58.00   Existing: 6151   1st Qu.:0.500
##                 Median :  77367   Median : 84.00                    Median :0.500
##                 Mean   : 178255   Mean   : 87.93                    Mean   :0.646
##                 3rd Qu.: 178000   3rd Qu.: 84.00                    3rd Qu.:0.850
##                 Max.   :8995000   Max.   :360.00                    Max.   :1.000
##
##  UrbanRural       Recession                                                 Industry
##  Urban:17111   Inactive:17322   Retail trade                                    :3072
##  Rural: 2089   Active  : 1878   Other services (except public admin)            :2188
```

```
##                                  Accommodation & food services                 :2118
##                                  Professional, scientific, & technical services:2071
##                                  Manufacturing                                 :1665
##                                  Construction                                  :1600
##                                  (Other)                                       :6486
```

```r
# Overall Default Rate
sba_overall = sba[!is.na(sba$mis_status), ]
sba_overall$mis_status = factor(sba_overall$mis_status)
overall_table = sba_overall %>% group_by(mis_status) %>% dplyr::count() %>%
    dplyr::mutate(percent = scales::percent(n/nrow(sba_overall)))
# overall_table

IL_table = sba_clean %>% group_by(MIS_Status) %>% dplyr::count() %>%
    dplyr::mutate(percent = scales::percent(n/nrow(sba_IL)))
# IL_table

Default_Comparison = data.frame(Loan_Status = IL_table$MIS_Status,
    Count_IL = IL_table$n, Percentage_IL = IL_table$percent,
    Count_National = overall_table$n, Percentage_National = overall_table$percent)
Default_Comparison
```

```
##   Loan_Status Count_IL Percentage_IL Count_National Percentage_National
## 1      ChgOff     5544           29%         157558                 18%
## 2         PIF    13656           71%         739609                 82%
```

Data Visualizations

We provide the data visualizations for all seven predictors.

1. NewExist: We think that existing businesses, which are businesses that have existed for at least two years when granted the loan, should have lower default rates compared to new businesses because existing businesses are more stable and are able to generate more cash. This can be seen from the horizontal bar chart. Existing businesses have lower default rate (26.5%) than new businesses (30%).

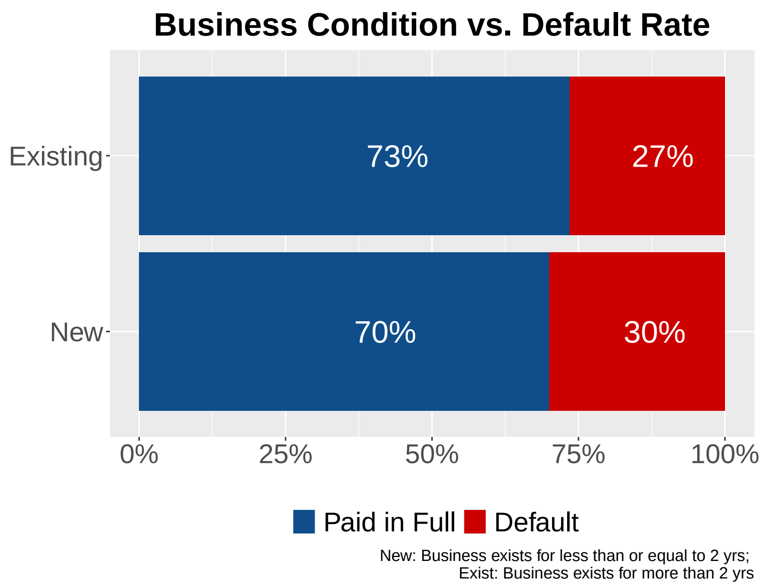
2. UrbanRural: We think that businesses in urban areas should have lower default rates compared to new businesses because urban areas have more opportunities and more likely to be successful. But to our surprise, the bar chart shows that rural areas (27.8%) perform slightly better than urban areas (29%). The reason could be that the agricultural industry is doing pretty well in Illinois and it contributes to the lower default rate in rural areas.

3. Recession: In our opinion, loans that are active during the Great Recession are more likely to be defaulted than other loans. The bar chart is showing this trend and the difference between the two categories are obvious. Loans that are active during the Great Recession has a default rate of 47.92% while inactive loans only have a default rate of 26.8%.

```r
# Data Visualization Part 1
library(tidyverse)

## NewExist vs. MIS_Status
NewExist_table = sba_clean %>% group_by(NewExist, MIS_Status) %>%
    dplyr::summarise(n = n()) %>% group_by(NewExist) %>% dplyr::mutate(percent = scales::percent(n/sum(n
# NewExist_table
p_NewExist <- ggplot(data = NewExist_table) + geom_bar(mapping = aes(x = NewExist,
    y = n, fill = MIS_Status), position = "fill", stat = "identity") +
    coord_flip() + scale_y_continuous(labels = scales::percent) +
    ggtitle(label = "Business Condition vs. Default Rate") +
```

```r
    scale_fill_manual("legend", values = c(PIF = "dodgerblue4",
        ChgOff = "red3"), labels = c("Default", "Paid in Full")) +
    xlab("") + ylab("") + geom_text(aes(x = NewExist, y = n,
    fill = MIS_Status, label = percent), position = position_fill(vjust = 0.6),
    size = 8, color = "white") + guides(fill = guide_legend(title = "",
    reverse = T)) + theme(legend.position = "bottom", legend.text = element_text(size = 20),
    plot.title = element_text(hjust = 0.5, face = "bold", size = 24),
    axis.text = element_text(size = 20), plot.caption = element_text(size = 12)) +
    labs(caption = "New: Business exists for less than or equal to 2 yrs;
       Exist: Business exists for more than 2 yrs")
p_NewExist
```

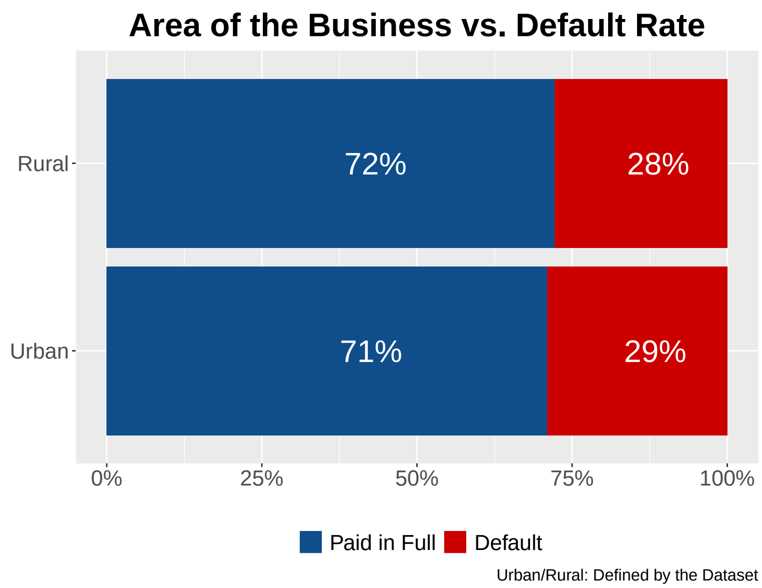



```r
## UrbanRural vs. MIS_Status
UrbanRural_table = sba_clean %>% group_by(UrbanRural, MIS_Status) %>%
    dplyr::summarise(n = n()) %>% group_by(UrbanRural) %>% dplyr::mutate(percent = scales::percent(n/su
UrbanRural_table
```

```
## # A tibble: 4 x 4
## # Groups:   UrbanRural [2]
##   UrbanRural MIS_Status     n percent
##   <fct>      <fct>      <int> <chr>
## 1 Urban      ChgOff      4963 29%
## 2 Urban      PIF        12148 71%
## 3 Rural      ChgOff       581 28%
## 4 Rural      PIF         1508 72%
```

```r
p_UrbanRural <- ggplot(data = UrbanRural_table) + geom_bar(mapping = aes(x = UrbanRural,
    y = n, fill = MIS_Status), position = "fill", stat = "identity") +
    coord_flip() + scale_y_continuous(labels = scales::percent) +
    ggtitle(label = "Area of the Business vs. Default Rate") +
    scale_fill_manual("legend", values = c(PIF = "dodgerblue4",
        ChgOff = "red3"), labels = c("Default", "Paid in Full")) +
    xlab("") + ylab("") + geom_text(aes(x = UrbanRural, y = n,
    fill = MIS_Status, label = percent), position = position_fill(vjust = 0.6),
    size = 8, color = "white") + guides(fill = guide_legend(title = "",
    reverse = T)) + theme(legend.position = "bottom", legend.text = element_text(size = 16),
```

```r
    plot.title = element_text(hjust = 0.5, face = "bold", size = 24),
    axis.text = element_text(size = 16), plot.caption = element_text(size = 12)) +
    labs(caption = "Urban/Rural: Defined by the Dataset")
p_UrbanRural
```

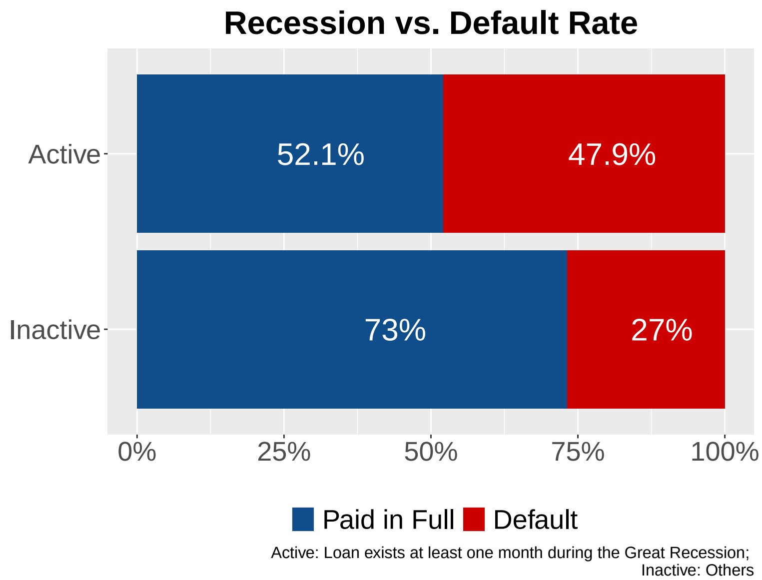



```r
## Recession vs. MIS_Status
Recession_table = sba_clean %>% group_by(Recession, MIS_Status) %>%
    dplyr::summarise(n = n()) %>% group_by(Recession) %>% dplyr::mutate(percent = scales::percent(n/sum
Recession_table
```

```
## # A tibble: 4 x 4
## # Groups:   Recession [2]
##   Recession MIS_Status     n percent
##   <fct>     <fct>      <int> <chr>
## 1 Inactive  ChgOff      4644 27%
## 2 Inactive  PIF        12678 73%
## 3 Active    ChgOff       900 47.9%
## 4 Active    PIF          978 52.1%
```

```r
p_Recession <- ggplot(data = Recession_table) + geom_bar(mapping = aes(x = Recession,
    y = n, fill = MIS_Status), position = "fill", stat = "identity") +
    coord_flip() + scale_y_continuous(labels = scales::percent) +
    ggtitle(label = "Recession vs. Default Rate") + scale_fill_manual("legend",
    values = c(PIF = "dodgerblue4", ChgOff = "red3"), labels = c("Default",
        "Paid in Full")) + xlab("") + ylab("") + geom_text(aes(x = Recession,
    y = n, fill = MIS_Status, label = percent), position = position_fill(vjust = 0.6),
    size = 8, color = "white") + guides(fill = guide_legend(title = "",
    reverse = T)) + theme(legend.position = "bottom", legend.text = element_text(size = 20),
    plot.title = element_text(hjust = 0.5, face = "bold", size = 24),
    axis.text = element_text(size = 20), plot.caption = element_text(size = 12)) +
    labs(caption = "Active: Loan exists at least one month during the Great Recession;
       Inactive: Others")

p_Recession
```



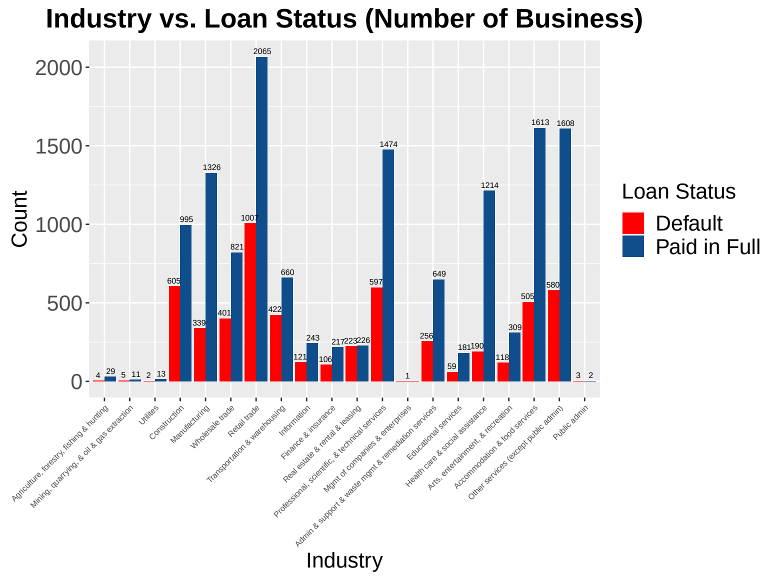
4. Industry: The data visualization for the Industry variable is to give us an overall picture about which industries have the greatest number of small businesses and which industries have higher default rates. We can get the information from the two bar charts: 1) the top three industries that have the greatest number of small businesses are Retail trade, Accommodation & food services and Professional, scientific, & technical services; 2) the top three industries with the highest default rates are Management of companies & enterprises, Public admin and Real estate & rental & leasing.

```
Industry_table = sba_clean %>% group_by(Industry, MIS_Status) %>%
    dplyr::summarise(n = n()) %>% group_by(Industry) %>% dplyr::mutate(percent = scales::percent(n/sum(
Industry_table
```

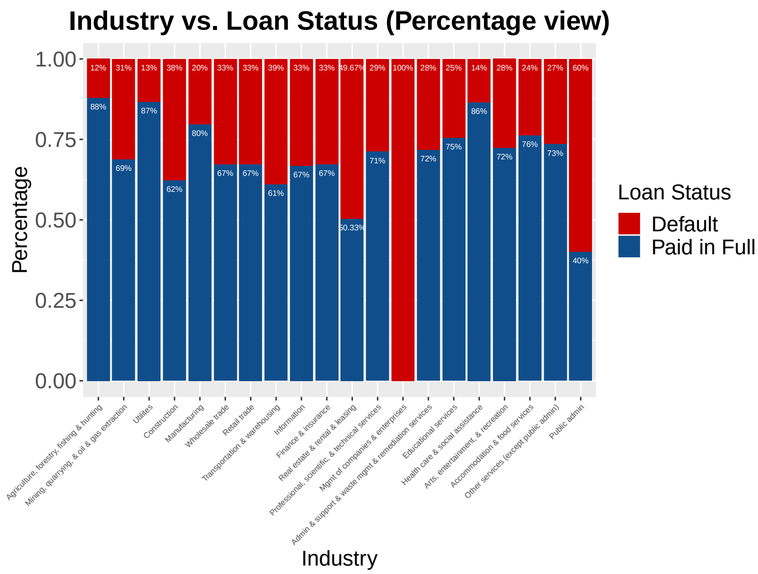
```
## # A tibble: 39 x 4
## # Groups:   Industry [20]
##    Industry                                  MIS_Status     n percent
##    <fct>                                     <fct>      <int> <chr>
##  1 Agriculture, forestry, fishing & hunting  ChgOff         4 12%
##  2 Agriculture, forestry, fishing & hunting  PIF           29 88%
##  3 Mining, quarrying, & oil & gas extraction ChgOff         5 31%
##  4 Mining, quarrying, & oil & gas extraction PIF           11 69%
##  5 Utilites                                  ChgOff         2 13%
##  6 Utilites                                  PIF           13 87%
##  7 Construction                              ChgOff       605 38%
##  8 Construction                              PIF          995 62%
##  9 Manufacturing                             ChgOff       339 20%
## 10 Manufacturing                             PIF         1326 80%
## # ... with 29 more rows
```

```
p_Industry1 = ggplot(Industry_table, aes(x = Industry, y = n,
    fill = MIS_Status)) + geom_bar(position = "dodge", stat = "identity") +
    labs(title = "Industry vs. Loan Status (Number of Business)",
        x = "Industry", y = "Count") + scale_fill_manual("Loan Status",
    values = c(PIF = "dodgerblue4", ChgOff = "red"), labels = c("Default",
        "Paid in Full")) + geom_text(aes(x = Industry, y = n,
    fill = MIS_Status, label = n), vjust = -0.5, position = position_dodge(width = 1),
    size = 2) + theme(axis.text.x = element_text(hjust = 1, angle = 45,
    size = 6), axis.text.y = element_text(size = 16), plot.title = element_text(size = 20,
```

```
    face = "bold", hjust = 0.5), axis.title = element_text(size = 16),
    plot.caption = element_text(size = 16), legend.title = element_text(size = 16),
    legend.text = element_text(size = 16))
p_Industry1
```




```
p_Industry2 = ggplot(Industry_table, aes(x = Industry, y = n,
    fill = MIS_Status)) + geom_bar(position = "fill", stat = "identity") +
    labs(title = "Industry vs. Loan Status (Percentage view)",
        x = "Industry", y = "Percentage") + scale_fill_manual("Loan Status",
    values = c(PIF = "dodgerblue4", ChgOff = "red3"), labels = c("Default",
        "Paid in Full")) + geom_text(aes(x = Industry, y = n,
    fill = MIS_Status, label = percent), vjust = 2, position = position_fill(),
    size = 2, color = "white") + theme(axis.text.x = element_text(hjust = 1,
    angle = 45, size = 6), axis.text.y = element_text(size = 16),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 16), legend.title = element_text(size = 16),
    legend.text = element_text(size = 16))
p_Industry2
```

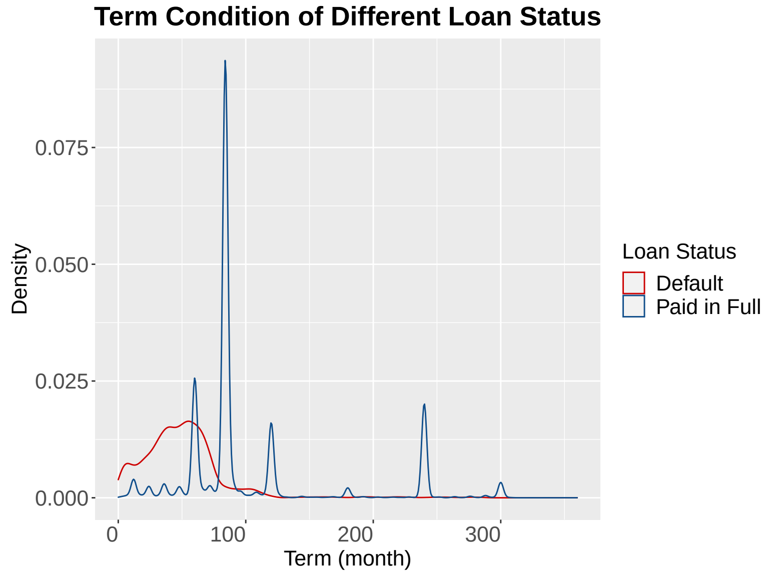


5. Term: From our view, we think loans with shorter term have higher default rates compared to longer term loans because loans with longer terms will have more time to establish their businesses and are more able to pay off the loans. This is proved from the density plot, where the default line (red) has a bump in the shorter term.

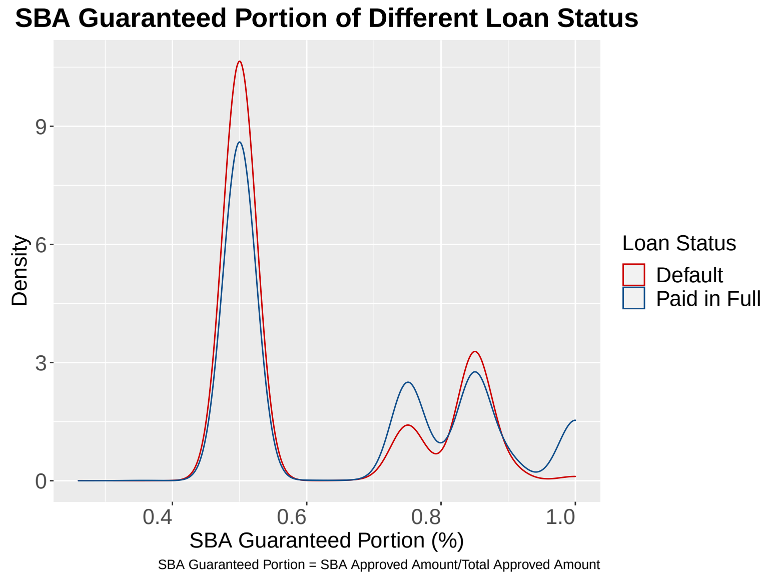
6. SBAGuaranteedPortion: From our instinct, we think that the more the SBA guarantees, the lower the default rate would be. But it turns out that this is not the case. In fact, this plot is the most interesting plot. The red line (default) and the blue line (paid-in-full) switched 3 times. The default rate has a higher density at around 50% and then goes down quickly; the paid-in-full line goes up and reaches a peak at around 75%; finally, the default line goes up and has a higher density at around 85%. This reaches to an interesting conclustion: the default rate is lower if SBA guarantees between 55% to 80%. SBA should not guarantee too much nor too little to avoid default.

7. DisbursementGross: This plot shows the trend of Gross Disbursement Amount. We can see that most loans are less than $2,500,000.00.

```
p_Term = ggplot(sba_clean, aes(x = Term, color = MIS_Status)) +
    geom_density() + labs(title = "Term Condition of Different Loan Status",
    x = "Term (month)", y = "Density") + scale_color_manual("Loan Status",
    values = c(PIF = "dodgerblue4", ChgOff = "red3"), labels = c("Default",
        "Paid in Full")) + theme(axis.text.x = element_text(hjust = 1,
    size = 16), axis.text.y = element_text(size = 16), plot.title = element_text(size = 20,
    face = "bold", hjust = 0.5), axis.title = element_text(size = 16),
    legend.title = element_text(size = 16), legend.text = element_text(size = 16))
p_Term
```

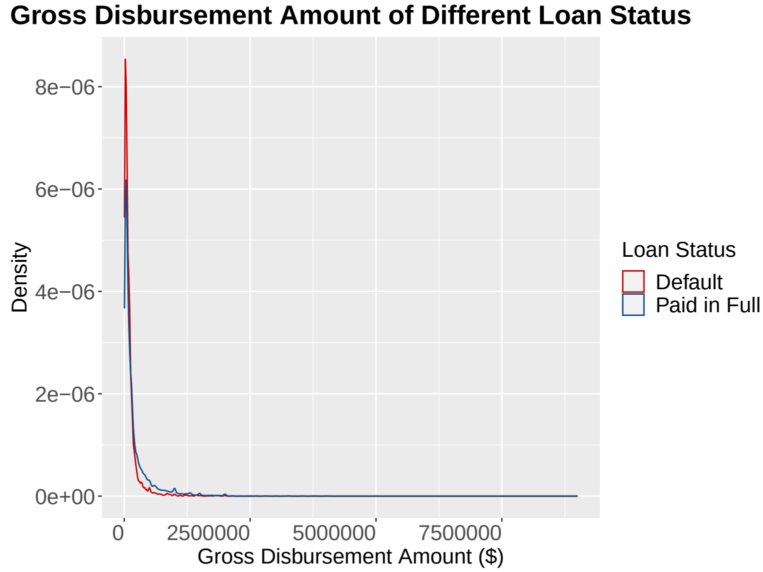



```r
p_SBAGuaranteedPortion = ggplot(sba_clean, aes(x = SBAGuaranteedPortion,
    color = MIS_Status)) + geom_density() + labs(title = "SBA Guaranteed Portion of Different Loan Statu
    x = "SBA Guaranteed Portion (%)", y = "Density") + scale_color_manual("Loan Status",
    values = c(PIF = "dodgerblue4", ChgOff = "red3"), labels = c("Default",
        "Paid in Full")) + theme(axis.text.x = element_text(hjust = 1,
    size = 16), axis.text.y = element_text(size = 16), plot.title = element_text(size = 20,
    face = "bold", hjust = 0.5), axis.title = element_text(size = 16),
    plot.caption = element_text(size = 10), legend.title = element_text(size = 16),
    legend.text = element_text(size = 16)) + labs(caption = "SBA Guaranteed Portion = SBA Approved Amoun
p_SBAGuaranteedPortion
```




```r
p_DisbursementGross = ggplot(data = sba_clean, aes(x = DisbursementGross,
    color = MIS_Status)) + geom_density() + labs(title = "Gross Disbursement Amount of Different Loan St
    x = "Gross Disbursement Amount ($)", y = "Density") + scale_color_manual("Loan Status",
    values = c(PIF = "dodgerblue4", ChgOff = "red3"), labels = c("Default",
        "Paid in Full")) + theme(axis.text.x = element_text(hjust = 1,
    size = 16), axis.text.y = element_text(size = 16), plot.title = element_text(size = 20,
```

```
    face = "bold", hjust = 0.5), axis.title = element_text(size = 16),
    legend.title = element_text(size = 16), legend.text = element_text(size = 16))
p_DisbursementGross
```



Logistic Regression

In the previous part, we analyzed 7 variables by visualizing them. Next, we use them as predictors to fit a logistic regression model in order to predict the loan status of small businesses in Illinois. First, we briefly introduce logistic regression.

Logistic regreesion is a widely-used model when the response variable is categorical.If the response variable has two possible outcomes, the distribution is binomial, which is exactly our case. Consider the response variable Y = $\{0, 1\}$. The logistic regression model can be written in the following equation:

$$log\frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \beta_0 + \beta^T x.$$

When a model contains too many predictors, there exist risks of overfitting and multicollinearity of predictors. Thus we introduce the penalized logistic regression. The objective function of a penalized logistic regression is as following:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} -\left[\frac{1}{N}\sum_{i=1}^{N} y_i(\beta_0 + x_i^T\beta) - log(1+e^{\beta_0+x_i^T\beta})\right] + \lambda\left[((1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1)\right].$$

Here λ is the penalty parameter that controls the overall penalty strength. As λ increases, the magnitude of coefficients shrink. If a variable contributes less to the response variable, its coefficient shrinks more than the others.Thus by applying the penalty term, we can put more weights on the predictors that has higher influence to the response.

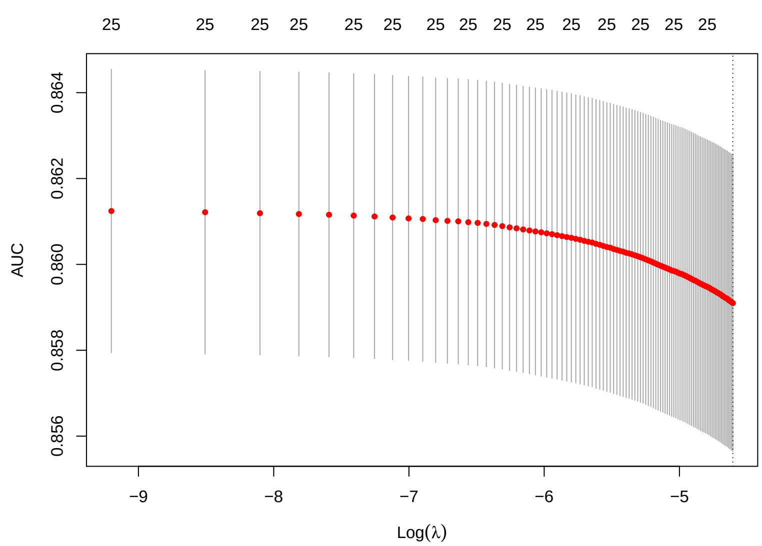
Another parameter, α, represents the gap between ridge and lasso, and it is called the elastic-net penalty. When $\alpha = 0$, the penalty term contains only the L2-norm of coefficients, and hence we get a pure ridge regression. When $\alpha = 1$, only L1-norm of coefficients is left, so we get a lasso regression. Ridge regression put coefficients of predictors with minor contribution closer to zero, but it incorporate all the predictors in the model. Lasso regression forces all the coefficients of less significant variables to be 0. Thus it also performs variable selection. Since we already decided the predictors used in our model from our prior knowledge and data visualization, we choose to use $\alpha = 0$, i.e., the ridge regression, to avoid variable selection in this part.

The next step is to find an optimal value of λ to fit a logistic regression model. To do this, we plot the log of λ against the AUC of model. AUC is short for area under the ROC (Receiver Operating Characteristics) curve. It is a commonly-used performance measurement for classification problem. It measures how well the model distinguishes one class from another. Therefore, higher AUC implies better performance of the model.

```
library(glmnet)

# Create dummy variables
sba_clean$Recession = NULL
levels(sba_clean$NewExist) = c(0, 1)
levels(sba_clean$UrbanRural) = c(0, 1)
sba_clean$NewExist = as.numeric(sba_clean$NewExist)
sba_clean$UrbanRural = as.numeric(sba_clean$UrbanRural)
sba_clean$Industry = as.factor(sba_IL$naics2)
levels(sba_clean$Industry)[levels(sba_clean$Industry) == "32"] = "31"
levels(sba_clean$Industry)[levels(sba_clean$Industry) == "33"] = "31"
levels(sba_clean$Industry)[levels(sba_clean$Industry) == "45"] = "44"
levels(sba_clean$Industry)[levels(sba_clean$Industry) == "49"] = "48"
for (i in 1:length(levels(sba_clean$Industry))) {
    newcol = paste("Industry_", levels(sba_clean$Industry)[i],
        sep = "")
    sba_clean[, newcol] = with(sba_clean, ifelse(sba_clean$Industry ==
        levels(sba_clean$Industry)[i], 1, 0))
}
sba_clean$Industry = NULL

# Plot lambda against AUC
X = as.matrix(sba_clean[, -1])
Y = sba_clean$MIS_Status
cv.model = cv.glmnet(X, Y, family = "binomial", type.measure = "auc",
    alpha = 0, lambda = seq(0, 0.01, length.out = 100))
plot(cv.model)
```



From the above plot we can see that as λ increases, the value of AUC decreases. Therefore, the best λ of our model is 0. This implies that there is almost no collinearity among our predictors, and we prevent the problem of overfitting. But it also suggests that we possibly did not include all the predictors that should be in the "full" model.

Then we fit the logistic regression model with both λ and α equal 0. We first fit the model with all the observations of small businesses in Illinois and take a look at the coefficients of the predictors.

```
full = glmnet(X, Y, family = "binomial", alpha = 0, lambda = 0)
coef(full)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                                 s0
## (Intercept)          -1.990865e+00
## DisbursementGross     3.999059e-07
## Term                  4.423446e-02
## NewExist              1.091666e-01
## SBAGuaranteedPortion -8.351620e-01
## UrbanRural            2.160779e-01
## Industry_11           1.210474e+00
## Industry_21           8.542433e-02
## Industry_22           8.959575e-01
## Industry_23          -2.999265e-01
## Industry_31           3.896018e-01
## Industry_42          -4.485504e-03
## Industry_44          -2.857769e-01
## Industry_48          -2.553286e-01
## Industry_51          -2.678709e-02
## Industry_52          -1.520563e-01
## Industry_53          -8.532312e-01
## Industry_54           2.427835e-02
## Industry_55          -9.645566e+00
## Industry_56           1.717182e-01
## Industry_61           2.281343e-01
## Industry_62           8.128218e-01
## Industry_71          -1.524082e-01
## Industry_72          -2.091087e-01
## Industry_81          -3.966750e-03
## Industry_92          -9.607593e-01
```

To test how well our model performs, we split the original dataset randomly into 10 parts. Each time we use one part as the testing data, and set the remaining data as the training data to fit the model. That is, we do a 10-fold cross validation. After fitting with all the splits, we gather the results and derive the confusion matrix of the testing responses and our predict values.

```
library(caret)
ind = createFolds(Y, k = 10, list = FALSE)
confs = 0
for (i in 1:10) {
    testY = Y[ind == i]
    trainY = Y[ind != i]
    testX = X[ind == i, ]
    trainX = X[ind != i, ]
    model = glmnet(trainX, trainY, family = "binomial", lambda = 0,
        alpha = 0)
    pred = predict(model, testX, type = "class")
    confs = confs + table(pred, testY)
}
rate = confs
rate[1, 1] = confs[1, 1]/(confs[1, 1] + confs[2, 1])
rate[2, 1] = 1 - confs[1, 1]/(confs[1, 1] + confs[2, 1])
```

```r
rate[1, 2] = confs[1, 2]/(confs[1, 2] + confs[2, 2])
rate[2, 2] = 1 - confs[1, 2]/(confs[1, 2] + confs[2, 2])
confs
```

```
##         testY
## pred     ChgOff   PIF
##   ChgOff   2936   966
##   PIF      2608 12690
```

```r
rate
```

```
##         testY
## pred         ChgOff        PIF
##   ChgOff 0.52958153 0.07073814
##   PIF    0.47041847 0.92926186
```

From the confusion matrix, we can see that when the actual response is *Paid in Full*, we get 92.99% of observations classified correctly. When the actual response is *Charged Off*, we still have 52.98% of classifications correct.

We also want to know the accruacy and the precision of our model. The accuracy and the precision are calculated as the following:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total observations}},$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}.$$

```r
acc = (confs[1, 1] + confs[2, 2])/length(ind)
pre = confs[1, 1]/(confs[1, 1] + confs[1, 2])
data.frame(Accuracy = acc, Precision = pre)
```

```
##    Accuracy Precision
## 1 0.8138542 0.7524346
```

For our model, we define *Charged Off* as positive and *Paid in Full* as negative. The calculated accuracy is 81.44%, and the precision is 75.42%. Both of them are pretty high, so the performance of our logistic regression model is satisfying.

Conlusion

In this report, we solve two questions: which predictors should we use to fit a logistic regression model with loan status of small businesses in Illinois as the response variable, and how well this model performs when predicting the loan status. By data visualization and cross valiation on our fitted model, we finalize the logistic regression model as:

$$\begin{aligned}\text{Loan Status } &= \text{Gross Disbursement} + \text{Term} + \text{New vs. Existing Business} \\ &\quad + \text{SBA Guaranteed Portion} + \text{Urban vs. Rural} + \text{Recession} + \text{Industry}.\end{aligned}$$

The perfomance of the prediction is quatified with the accuracy and the precision, which equal 81.44% and 75.42% respectively. Both of them suggest that our model is appropriate.

Our model still has some limitations. For example, the current value of λ we use is 0, so it is possible that we miss out some predictors that do have an influence on the response variable. In the future, we could consider adding more variables into the model to achieve better performance.