Part 1

Florin Andrei

10/15/2020

Introduction

We are going to use logistic regression to predict which applicants are likely to default on their bank loans. The dataset is a sample of 50k loans, with 30 variables. The dataset file, along with a description of the variables, can be found here:

https://datascienceuwl.github.io/Project2018/TheData.html

The response variable ("fate") is constructed from the "status" variable in the dataset. "fate" has only two possible values: Good or Bad. Good loans are those with status = "Fully Paid". Bad loans are those with status = c("Charged Off", "Default"). All other status values are irrelevant and will be removed.

Preparing and cleaning the data

Obligatory library() chunk:

```
library(dplyr)
library(readr)
library(ggformula)
library(grid)
library(gridExtra)
library(tidyr)
```

Let's load the data:

```
dsfile_url <- 'https://datascienceuwl.github.io/Project2018/loans50k.csv'
dsfile <- 'loans50k.csv'
if (!file.exists(dsfile)) {
  download.file(dsfile_url, destfile = dsfile)
}
loans <- read_csv(dsfile)
```

Preparing the response variable

Some categories in the "status" variable are irrelevant, and we're going to remove them. The remainder will be collapsed into two main categories: Good and Bad, contained in the new "fate" column. The "status" column is not needed after this, so we'll drop it.

```
status_good   <- c("Fully Paid")
status_bad    <- c("Charged Off", "Default")
status_remove <- c("Late (16-30 days)", "Late (31-120 days)", "Current", "In Grace Period", NA)

loans <- loans %>% filter(!(status %in% status_remove))

loans <- loans %>% mutate(fate = case_when(
  status %in% status_good ~ 'Good',
  status %in% status_bad  ~ 'Bad'
))

loans <- loans %>% select(-c('status'))
```

Eliminating useless predictors

The sample is large enough (N = 50k) that correlations between the response variable ("fate") and the various other variables should be visible already, if there is any correlation. We're going to try and find out if the ratio Bad / (Good + Bad) loans is correlated with any categories (for categorical variables) or range buckets (for numeric variables).

Variables that show no correlation with "fate" are going to be dropped.

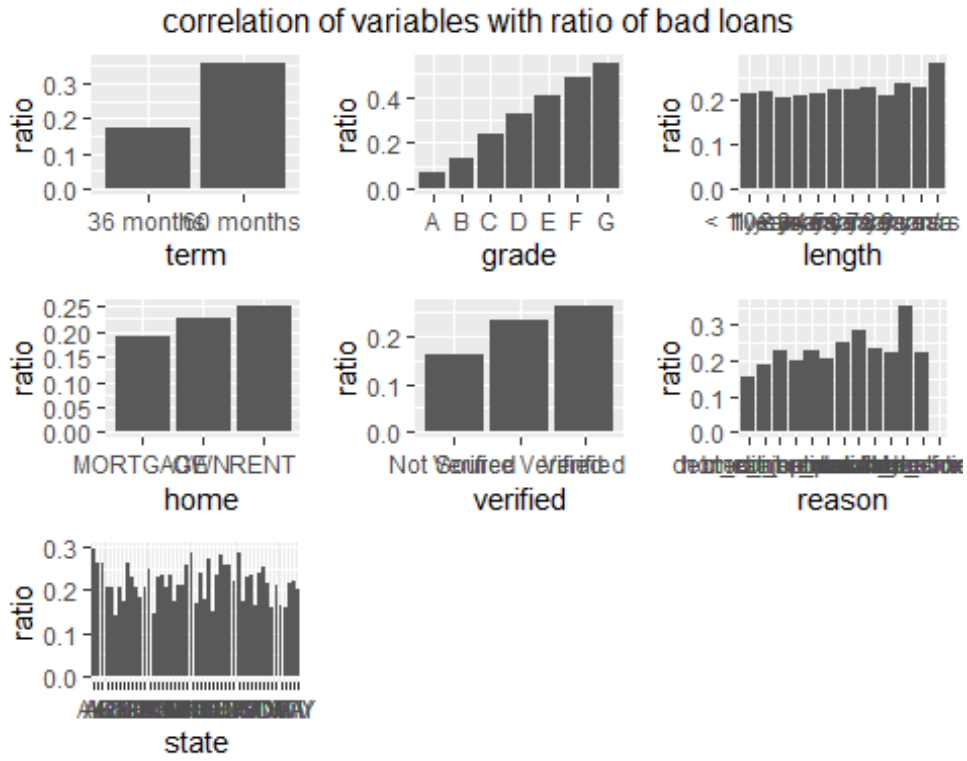
In other words: I am making no assumptions, I will let the numbers decide which variables ought to be dropped.

Categorical variables:

```
cat_vars <- c('term', 'grade', 'employment', 'length', 'home', 'verified', 'reason', 'state')
```

Employment has far too many little categories that seem to be quite ad-hoc and only have a few cases each. I think it's just noise, it doesn't seem to contribute useful data. I am going to drop it.

Let's visualize the ratio Bad / (Good + Bad) loans for all categories:



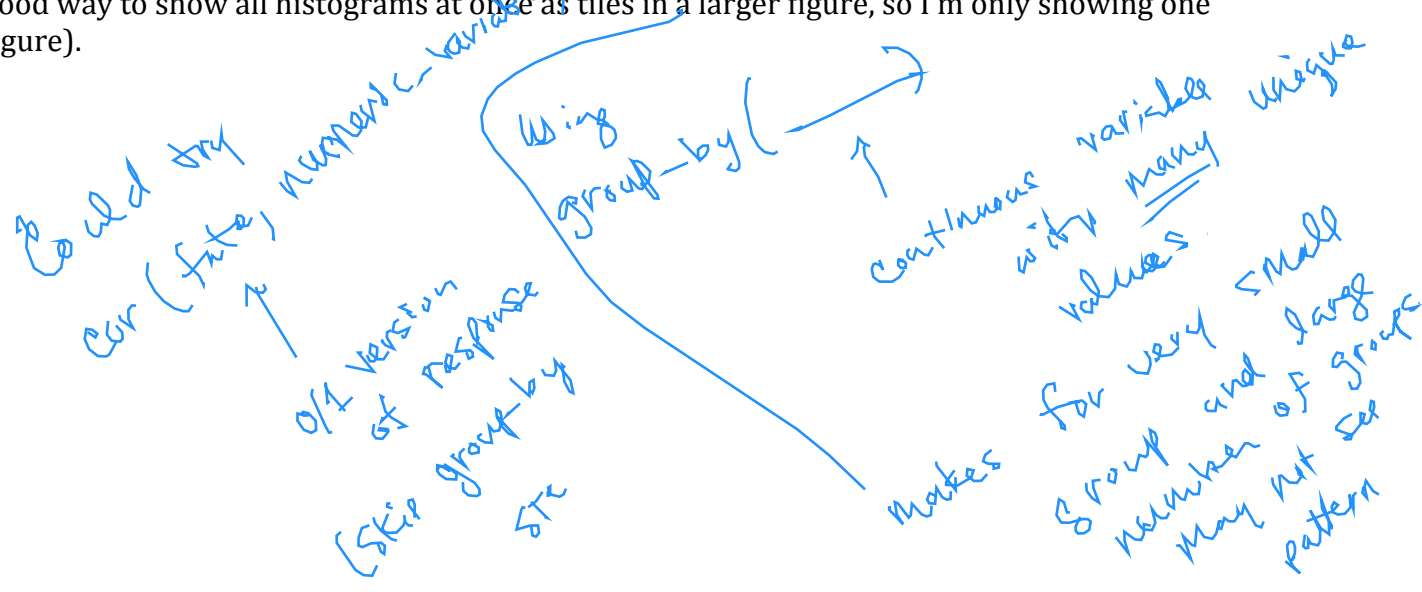
- length doesn't seem strongly correlated with good-vs-bad loans.
- term, grade, and state are strongly correlated. grade is **very** strongly correlated (it's a pretty decent answer to the prediction problem, right there).
- home, verified, and reason are mildly correlated.

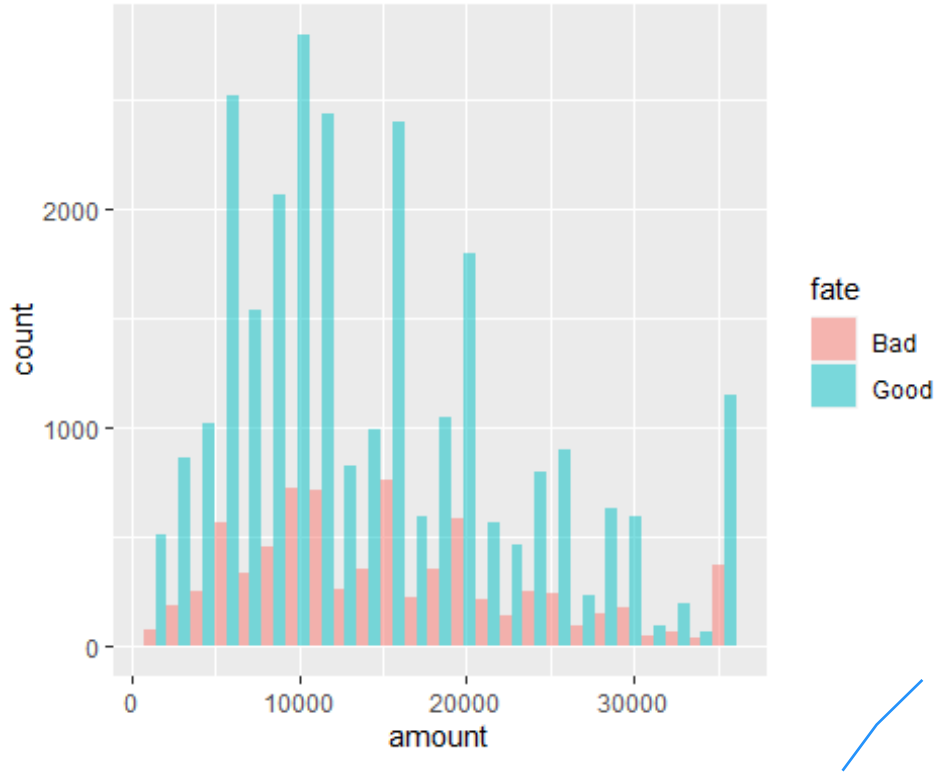
Useless vars to remove so far:

```
vars2remove <- c('totalPaid', 'employment', 'length')
```

Numeric variables

Here's an example of the number of Good / Bad loans as a function of the "amount" variable. Same visualization could be done for all other numeric variables (I could not find a good way to show all histograms at once as tiles in a larger figure, so I'm only showing one figure).





(It should really be the ratio Bad / (Good + Bad), not side by side Good vs Bad. I did not have time to figure how to do that in R without simulating the logic of gf_histogram() in discrete code - and that seemed like a lot of work for the time I have available. Side by side might be deceptive, however.)

Based on the histograms, these are the numeric variables that do not seem to be correlated with the fraction of bad loans:

```
vars2remove <- c(vars2remove, 'amount', 'payment', 'delinq2yr', 'inq6mth', 'openAcc', 'pubRec',
'revolRatio', 'totalAcc', 'bcRatio', 'totalRevBal', 'totalIlLim')
```

Let's remove the non-correlated variables from the dataframe:

```
loans <- loans %>% select(-all_of(vars2remove))
```

Feature engineering

The "reason" variable has a couple categories that have only a few cases each, not enough to draw conclusions at any reasonable confidence level. Let's lump those together as "other".

```
loans <- loans %>%
  mutate(reason = replace(reason, reason == 'wedding', 'other')) %>%
  mutate(reason = replace(reason, reason == 'renewable_energy', 'other'))
```

Missing values

summary(loans) reveals that the "bcOpen" column has 360 NA entries. There is no way to infer these from other data:

- each row in the table is independent
- the bcOpen column does not seem like it could be deduced from other columns - if it could, then we ought to remove it altogether anyway

No other columns contain NAs at this point. So let's drop the NAs.

```
loans <- loans %>% drop_na()
```

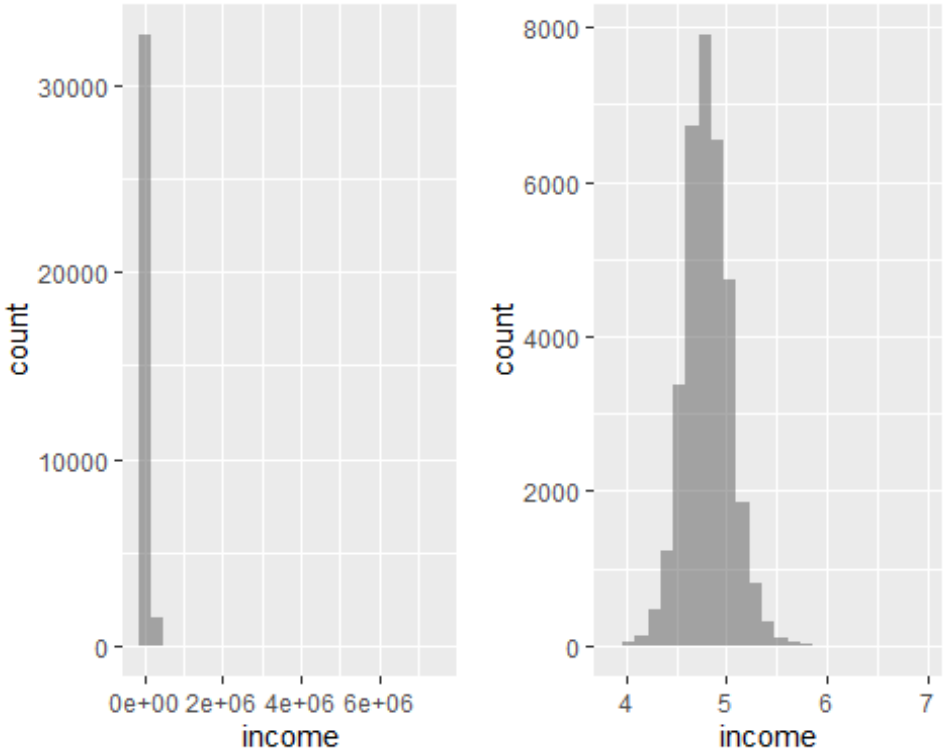
Exploring and Transforming the data

Fixing skewed variables

Some quantitative variables have strong right skew. This can be fixed with log10(). In some cases, those variables have values of 0, which log10 will not accept; typical values for those variables in our sample data are in the thousands, so the difference between 0 and 1 is irrelevant - we will replace 0 with 1 to keep log10 happy.

Here is "income" before and after fixing skew with log10:

```
p1 <- loans %>% gf_histogram(~ income)
p2 <- loans %>% mutate(income = log10(income)) %>% gf_histogram(~ income)
grid.arrange(p1, p2, ncol = 2)
```



We're going to remove right skew with log10() for those numeric variables that are skewed. Where necessary, 0 will be replaced with 1 to avoid log10(0).

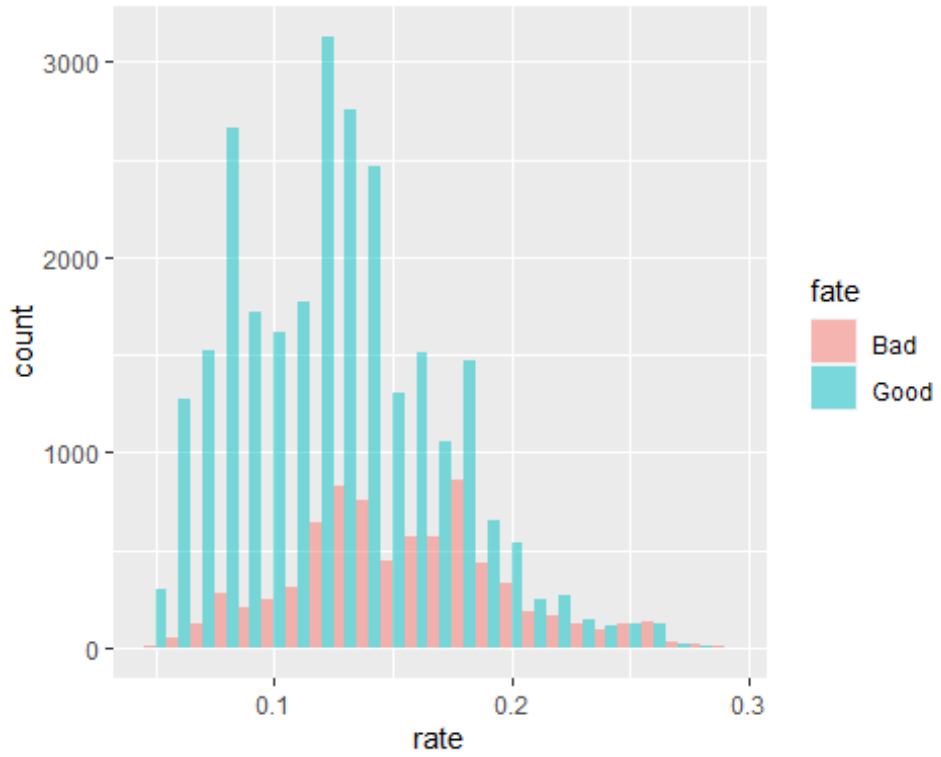
```
loans <- loans %>%
  mutate(income = log10(income)) %>%
  mutate(totalRevLim = log10(totalRevLim)) %>%
  mutate(totalLim = log10(totalLim))

loans <- loans %>%
  mutate(avgBal = replace(avgBal, avgBal == 0, 1)) %>% mutate(avgBal = log10(avgBal)) %>%
  mutate(totalBal = replace(totalBal, totalBal == 0, 1)) %>% mutate(totalBal = log10(totalBal)) %>%
  mutate(bcOpen = replace(bcOpen, bcOpen == 0, 1)) %>% mutate(bcOpen = log10(bcOpen)) %>%
  mutate(totalBcLim = replace(totalBcLim, totalBcLim == 0, 1)) %>% mutate(totalBcLim =
log10(totalBcLim))
```

Data exploration

I've done a bit of this before, in the section **Eliminating useless predictors**. In fact, in that section we've eliminated all variables that did not seem to show different behavior for Good vs Bad loans. But let's sample a few variables again, after all these changes.

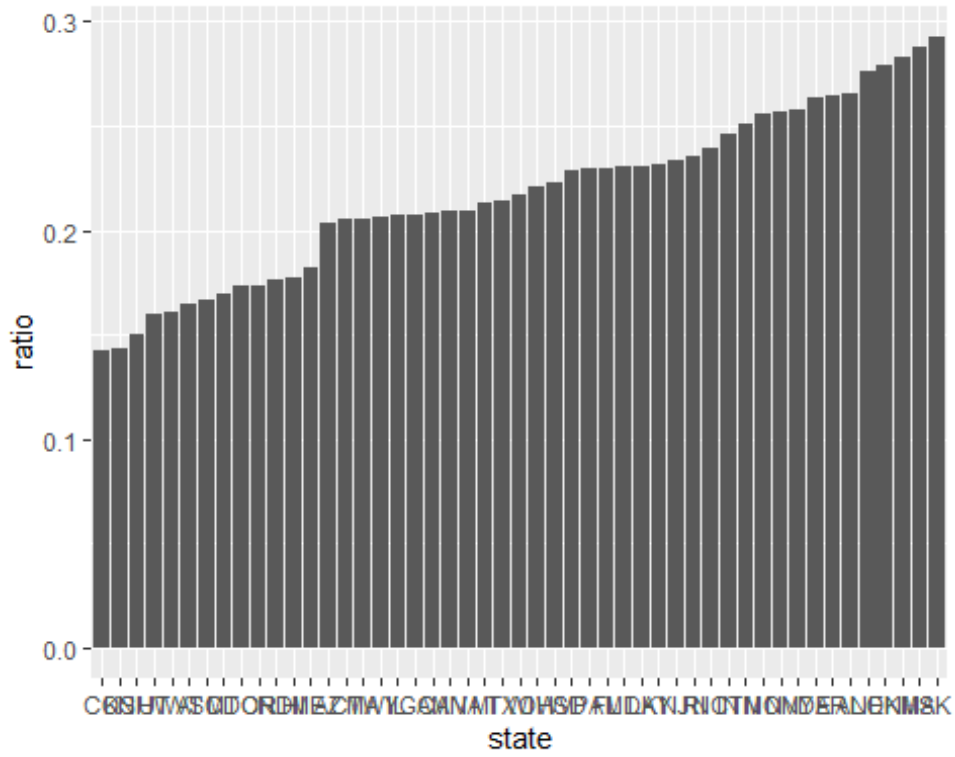
As an example of numeric variable, "rate" shows marked differences between Good and Bad loans. At low "rate" values, there are far more Good loans. At high "rate" values, the Bad loans rise to the point where they become equal to Good (this is intuitive behavior - it's harder to pay off loans at higher rates).

```
loans %>% gf_histogram(fill =~ fate, position = position_dodge(), ~ rate)
```



Among categorical variables, “state” shows some correlation as well. In some states (such as CO), the Bad loans are only about 14% of the total, whereas other states (such as AK) show as much as 28% Bad loans (twice as much as the other end of the scale).

```
loans %>% group_by(state) %>% summarise(ratio = sum(fate == 'Bad') / (sum(fate == 'Good') + sum(fate == 'Bad'))) %>% mutate(state = reorder(state, ratio)) %>% gf_col(ratio ~ state)
```



(the figure ought to be stretched sideways a bit to show the state names more clearly)

All remaining variables actually show some correlation with the proportion of Bad loans, but it's hard to display all of them without taking too many pages in this document. The samples shown above are typical.