Lending Club data wrangling
Thomas Loeber

I start by checking for any duplicate observations, but I did not find any.
Next, I set the issue date of the loan as the index, because this will allow us to more easily slice and group loans by date. Since oftentimes multiple loans were issued on the same date, I create a hierarchical index that also includes the loan ID.

I go on to drop any observations that are irrelevant because our target variable – whether someone defaulted on their loan or not – is yet indeterminate. These consist of observations where the loan is still current, and where the loan is late by up to 120 days.
Having gotten rid of unnecessary rows, we now drop unnecessary columns, starting with variables whose values were not yet available at the time the loan was issued . Doing so is important because using such information from the future would create the appearance of better predictive accuracy than the model actually possesses (endogeneity). At the same time, this step is also tricky, because the data documentation is often lacking; as a result, a few such variables are only discovered later on. Furthermore, I delete a few variables which are not relevant, as revealed by the data dictionary.

Next, I address the topic of missing values. This is a problem for roughly a third of our variables, because over time Lending Club started collecting additional groups of variables. I decided to discard all variables that have at least 30% missing values (though any threshold between 15% and 65% would have lead to the same results, because no variables had a proportion of missing values falling within this interval).
I made sure not to drop variables where missingness stands for "not applicable" rather than "not collected." Examples are variables such as time since last delinquency, because not all people in the sample have already experienced this event. I identified those variables by the fact that for those cases, missingness is distributed seemingly randomly across time, rather than confined to certain sample periods (usually the beginning of the sample).
Since all variables for which this problem occurred referred to the time that has passed since a specific event, I chose the following transformation for them: First, since for some observations the event occurred zero months ago, I added 1 to each value to make all the counts positive. Then, I raised each variable to the power of -0.5. Finally, I set the value to zero for all observations for which the underlying event did not occur. (The fact that the event did not occur was indicated either by the value being missing or – for earlier observations – zero.) The result is that observations for which the result occurred more recently receive a higher score; and the longer ago the event occurred, the closer the score moves to zero. For borrowers for which the event did not occur at all, the score equals exactly zero, which models that this is equivalent to the event occurring an infinite time ago.

Down the line, I will deal with the remaining missing values by performing a multivariate imputation. The advantage of this – compared to the conventional approach of using the mean, median, or mode – is that it uses the conditional expected value given other attributes of the

observation, rather than the unconditional expected value. The former is a (usually much) better estimate, because it makes use of additional information available to us.
However, this imputation will be deferred until the predictive modeling phase. Specifically, it will be performed only on the training set, in order to avoid giving our model an unfair advantage by already showing it the test data.

I then go on to check whether variables are of the right data type. Originally, all data are imported either as floats or objects. For floats, I identify variables that are in fact integers, and also don't have any missing values (which would require storing them as floats anyway). I then convert them to integers.

Next, I manually inspect all variables of type object. This identifies two variables that should be datetime objects, as well as three variables that should be numeric, but were imported as objects because the unit was appended to the value. I convert all these to the proper type. In addition, I change the datetime variables from absolute date to relative date (time since the loan was issued).
The remaining variables are in fact categorical. After the EDA, I will transform them using one-hot-encoding, after dropping variables with too many unique categories.