The main purpose of the explanatory data analysis (EDA) I carried out was to, firstly, assist in data cleaning by identifying potential problems in the data, and secondly to assist in feature engineering.

Another common use of EDA is to discover interesting relationships between variables (e.g., a non-linear effect that we may try to model by including a variable's square term). Let me justify why I did not make much use of this: Usually, we are interested in the *partial* effect of a variable, i.e. holding other variables constant. If we use visualizations, it is generally only practical to plot the effects of two or three variables at a time. This tends to give a misleading picture of partial effects (e.g., correlated predictors will give rise to spurious correlations). While one might argue that it can't hurt to try finding something interesting that way, we need to take the opportunity cost into account, so I think time is better invested elsewhere. For instance, if we use a flexible model such as gradient boosting, we don't have to worry about trying to visually detect all nonlinear effects and interactions between predictors.

A better use of our time is to use EDA to find problems with the data. The reason is this often draws on our domain knowledge, as well as general human knowledge about the world. Both of these are hard to encode into a model, such as the fact that we know that certain variables has to fall within a particular range. Another common problem is that some missing values may have been encoded as zeros or missing values standing for "not applicable" rather than that the true value is unknown.

In fact, I detected both of these problems with Lending Club's data through data visualization. The first problem was revealed while I was deciding which variables have a high enough ratio of valid to missing values to impute the missing values (rather than dropping the whole variable). In making this decision, I not only considered the proportion of valid observations, but also visualized how the missing values are distributed over time. This showed that – unsurprisingly – a high fraction of missingness was usually caused by the fact that the variable was only collected for a subset of the sample period. However, it was suspicious that for some variables the missing values were distributed seemingly randomly across time. (About half the values were missing, so it is unlikely that such a high proportion simply failed to be properly recorded). This prompted me to take a closer look, which revealed that missingness seemed to denote not "unknown", but rather "not applicable": For example, the variable "time since last delinquency" was not applicable to borrowers who never experienced a delinquency. Unfortunately, we don't have two different types of missing values available to encode each. I solved this problem by setting the "time-since" variable to negative infinity for each observation where the event count (e.g., the number of delinquencies) was zero. (Setting it to negative infinity works because I later transformed the "time-since" variables by raising them to the power of -0.5, which gives a higher score to applicants to whom the event occurred more recently, and converges to zero as the time-since approaches infinity).

In the process of making sense of these inconsistencies, data visualization also uncovered a related problem, namely that "not applicable" was encoded not always as missing, but in some

cases instead as zero. It turned out that this practice seems to have changed at some point in the middle of the sample period. Since the data dictionary did not address this, I had to use a number of different plots to figure out what was going on. For example, I looked only at the subset of borrowers who had a zero for the number of months since the last delinquency, and then plotted the monthly average for the number of delinquencies. This revealed that the average monthly number of delinquencies was zero until 2010, but from 2013 on hovered around 2 (there were no observations for the years in between). This suggested that the practice of encoding missing values as zeros stopped somewhere between 2010 and 2013, but this interpretation raises some other inconsistencies, which further plotting showed to be due to a change in Lending Club's lending behavior. (This explained, for example, why there were no zeros at all during the middle of the sample period: By that time, "not applicable" was denoted by "missing", and the lending behavior was still so cautious as to not accept any applicants who had just experienced a delinquency zero months ago.)

A different use of data visualization was to assist in feature engineering. The goal was to create features that are less skewed, in particular by taking the logarithm where necessary. I eventually automated this task by quantifying skewness, but in the process it was still helpful to use data visualization in order to come up with good measures and to make sure no errors were introduced. For example, the measure of skewness I settled on looked at how much closer/further the median was from the upper quartile compared to the lower quartile. Thus, this measure did not work for variables that were so skewed that all three quartiles fell onto the same value, so I had to visually decide what was going on with these distributions. Furthermore, plotting some of the extremely skewed variables showed that for many variables, skewness was caused by an inflated number of zeros (located at the minimum) or ones (located at the maximum). For example, for most applicants the number of derogatory public records was zero. This prompted me to take a second step to remediate skewness, namely to create another binary feature for these extremely skewed variables, telling us which values are equal to this inflated minimum or maximum.

Overall, thus, exploratory data analysis through data visualization was able to both detect some problems with the data, and also helped create better features. While I tried to automate these manual tasks as much as possible (e.g., by writing functions that try if taking the logarithm decreases skewness), some of these tasks are (still) too hard or time-consuming to automate.