Credit Modelling

Chris Hua

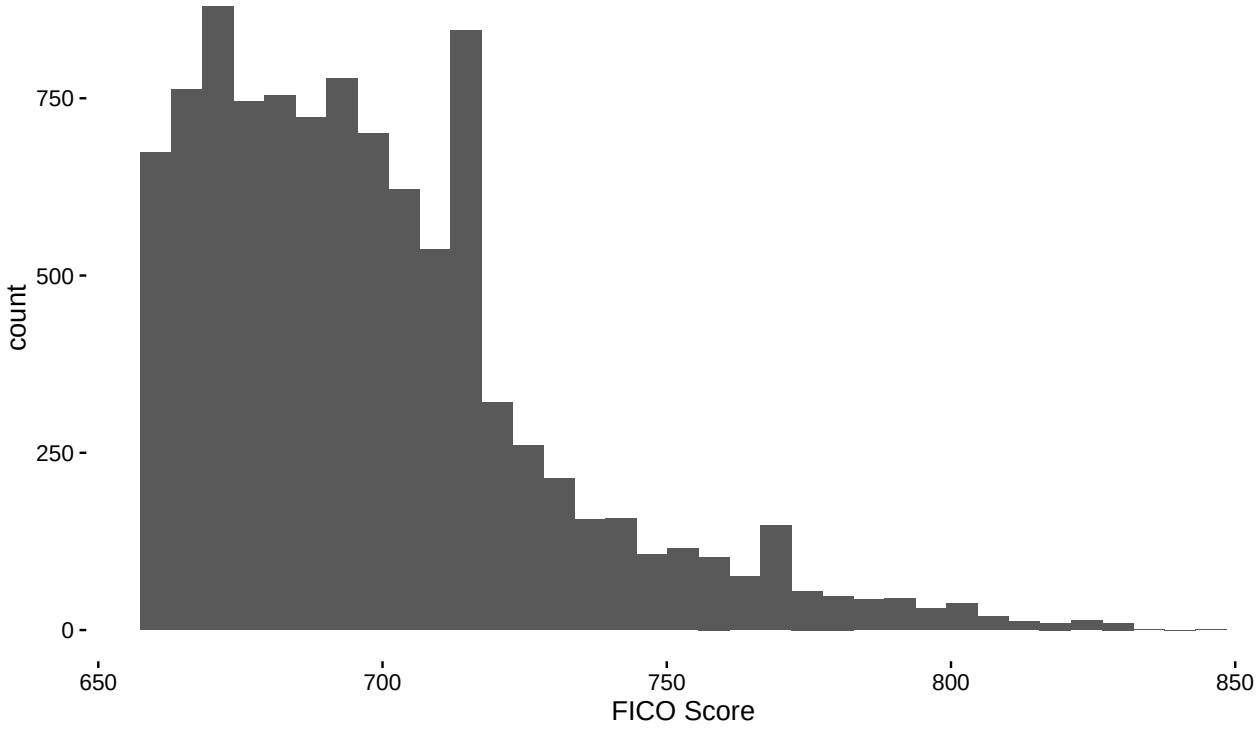
11/7/2016

0: Data Exploration

First, we note the distribution of the outcome variable:

```
.
Defaulted      Paid
     1623      8377
```

As well as the distribution of the FICO score, among applicants in this dataset. This is a pretty heavily left-skewed distribution, which probably represents the fact that the people who try to get loans on Lending Club are people who can't get loans traditionally.



1: Basic Modelling

We model "default probability" with respect to "FICO" score only. Here, a positive outcome denotes that the user defaulted, and a negative outcome denotes that the user did not default. Note that we have to relabel the factor levels for this to work by default.

a. We expect a negative coefficient on FICO. Intuitively, a customer is less likely to default the better their credit score is. We would also expect a positive intercept, because for a customer with 0 credit score (the condition for the intercept), they have terrible chances of paying off the bill, and thus high chance for default.

b. We estimate our model using a generalized linear model (`glm`) approach, with a binomial prior. Summary statistics of the model follow:

term	estimate	std.error	statistic	p.value
(Intercept)	7.68	0.76	10.17	0
fico	-0.01	0.00	-12.29	0

By p-value, `fico` is a statistically significant value at better than at 0.01 level, indicating that by itself, `fico` is a useful predictor for default outcomes.

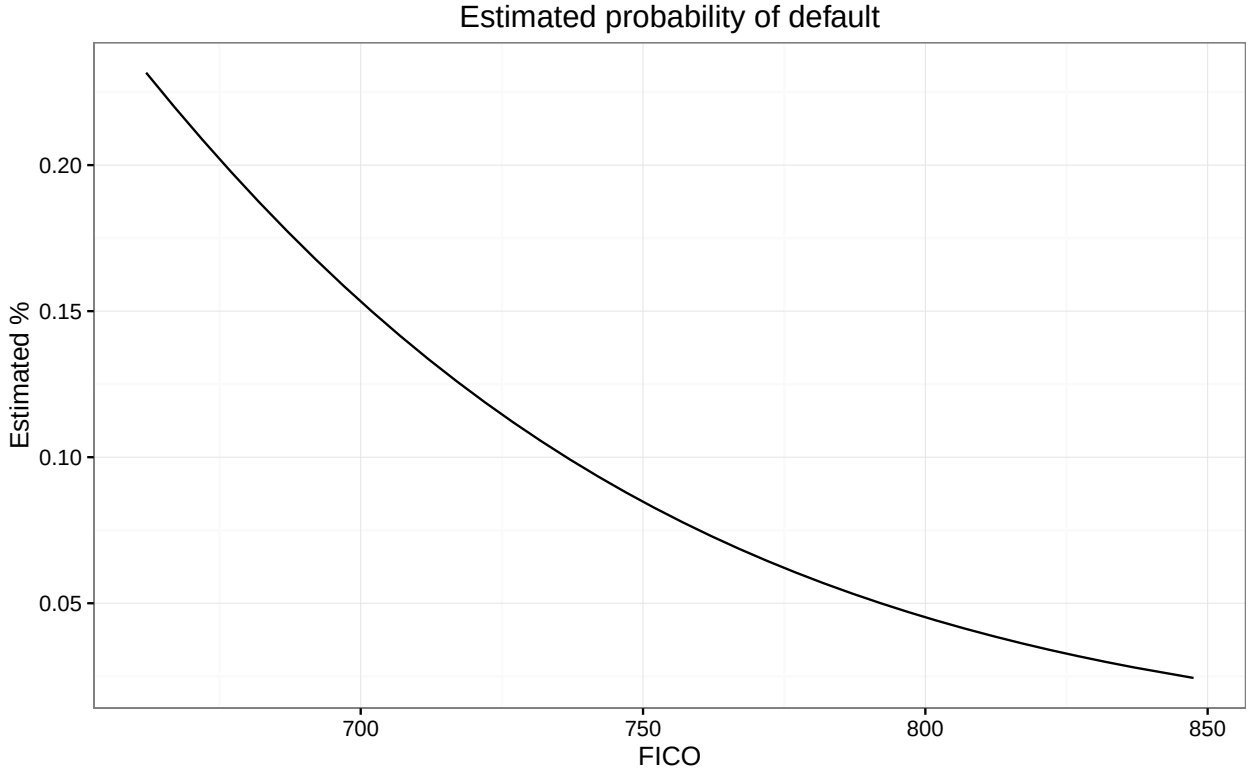
The intercept is strongly positive, which indicates that default is very likely for somebody with a `fico` score of 0. Alternatively, the intercept suggests that somebody with a score of 768 has 0 chance of default. The negative coefficient means that for each point of FICO score, the likelihood of default decreases by 0.01. Each of these observations lines up with our intuitions.

2: Model Evaluation

(a) We can estimate a probability of default via the formula from class 4:

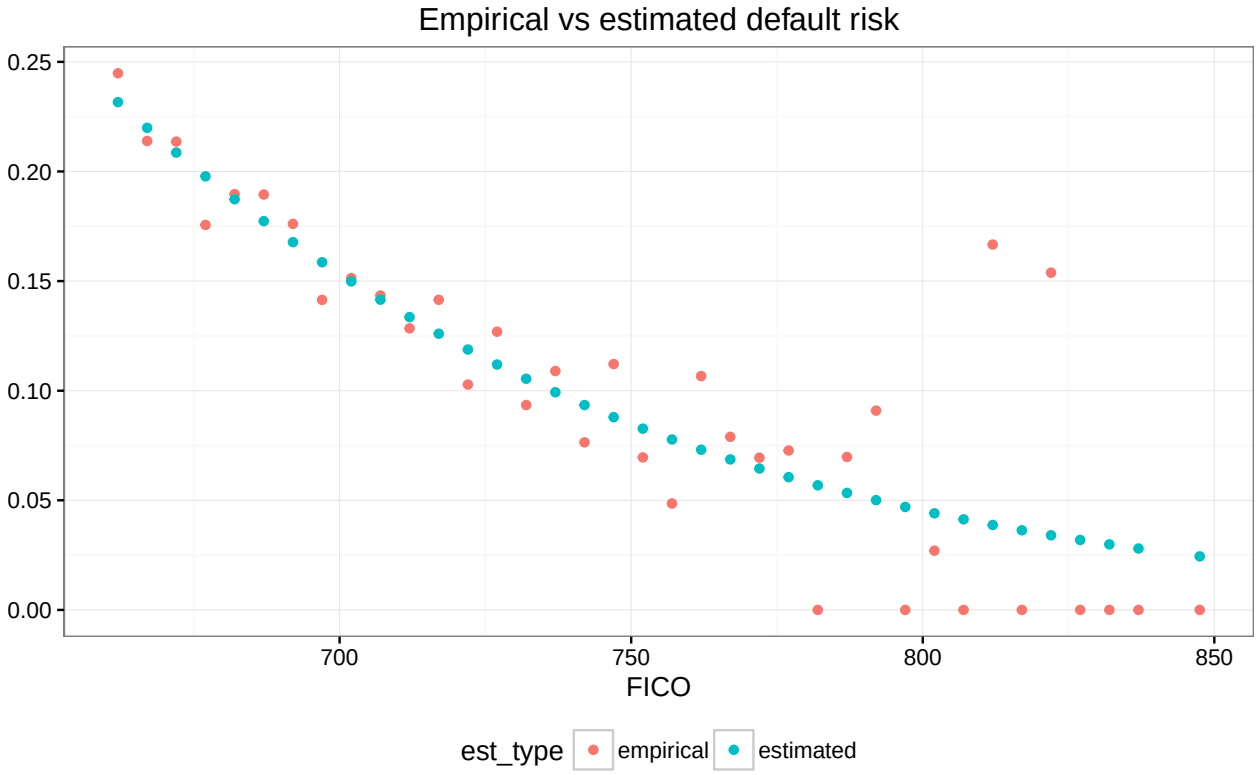
$$Pr(f) = \frac{\exp(\beta_0 - \beta_1 \times f)}{1 + \exp(\beta_0 - \beta_1 \times f)}$$

Since our model only includes FICO scores, we can easily plot FICO vs estimated probability of default.

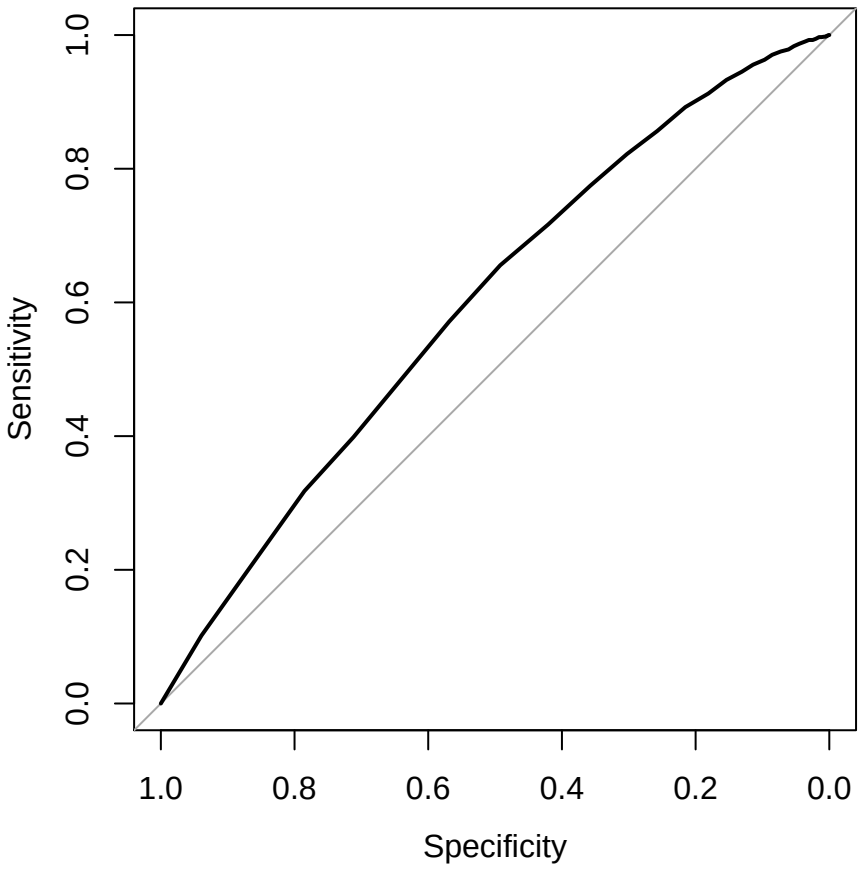


Interestingly, we can compare this plot to the empirically determined default risk. We can see that the estimated curve fits very well, except for the outliers in high credit score. This might tell us that high credit

score borrowers have asymmetric information about their ability to pay off loans, and they are the 'lemons' in this market.



(b) Then, we can plot a corresponding ROC curve for this model:




```
##
## Call:
## roc.formula(formula = default ~ prob, data = .)
##
## Data: prob in 8377 controls (default FALSE) < 1623 cases (default TRUE).
## Area under the curve: 0.5981
```

(c) For this model, we have AUC of 0.598. This is better than 0.5. It's consistent with the findings in part 1b, where we found that FICO was a statistically significant predictor alone of default.

(d) This is essentially using a 0.1 threshold on our probabilities of default. Then, we can create a confusion matrix:

	FALSE	TRUE
FALSE	1095	89
TRUE	7282	1534

The proportion correctly rejected is 0.131. The proportion mistakenly rejected is 0.869. Whether this is acceptable or not is dependent on the cost of a false positive vs false negative for the lender.

3: An out-of-sample analysis

Note that we use the first 9000 rows as training and last 1000 rows as test - in protest. It would be much more theoretically legitimate to randomly sample the rows to use.

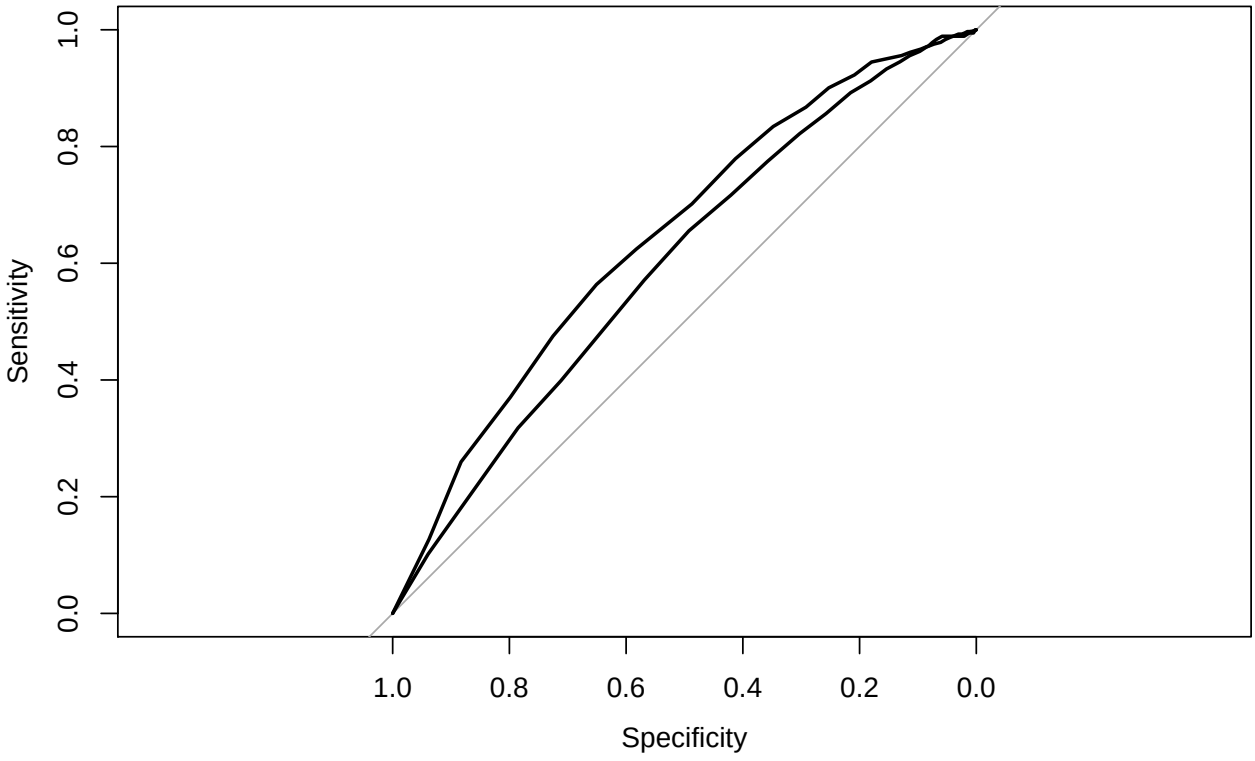
(a) The new model, estimated on the first 9000 rows, is given by these summary statistics:

term	estimate	std.error	statistic	p.value
(Intercept)	7.16	0.79	9.06	0
fico	-0.01	0.00	-11.11	0

This is pretty similar to the model that we estimated on the full 1000 data points, but has a less negative intercept.

(b) Then, we can predict probabilities for the remaining loans, and then create an ROC curve for both fits:

```
##
## Call:
## roc.formula(formula = default ~ prob, data = .)
##
## Data: prob in 8377 controls (default FALSE) < 1623 cases (default TRUE).
## Area under the curve: 0.5981
```




```
##
## Call:
## roc.formula(formula = default ~ pred, data = ., )
##
## Data: pred in 819 controls (default FALSE) < 181 cases (default TRUE).
## Area under the curve: 0.6459
```

(c) The area below the new ROC curve gets larger. My inclination is that we are no longer overfitting our dataset.

(d) Vacuously, you don't want to use all variables available, because some of the variables are unique to a person or a loan- e.g. "id". These could perfectly estimate somebody's probability of default in the sample set but have no predictive use.

If we fit every variable into the logistic model, we run the chance of overfitting. This is where we find some spurious behavior in the training data, that isn't representative of the underlying data, but rather an artifact of the existing data points.

Finally, we also may want parsimoniousness in our model, which is having fewer variables and being able to explain the data with these fewer variables. It is harder to explain a 10 variable model than a 3 variable model.

e Let's consider interest rate, employment length, and annual income. We're going to do this on test-train split as well. We make one adjustment to the data - we divide annual income by 1000. Ordinary least squares regression is scale-invariant

We fit the model and show the Anova table (type II tests):

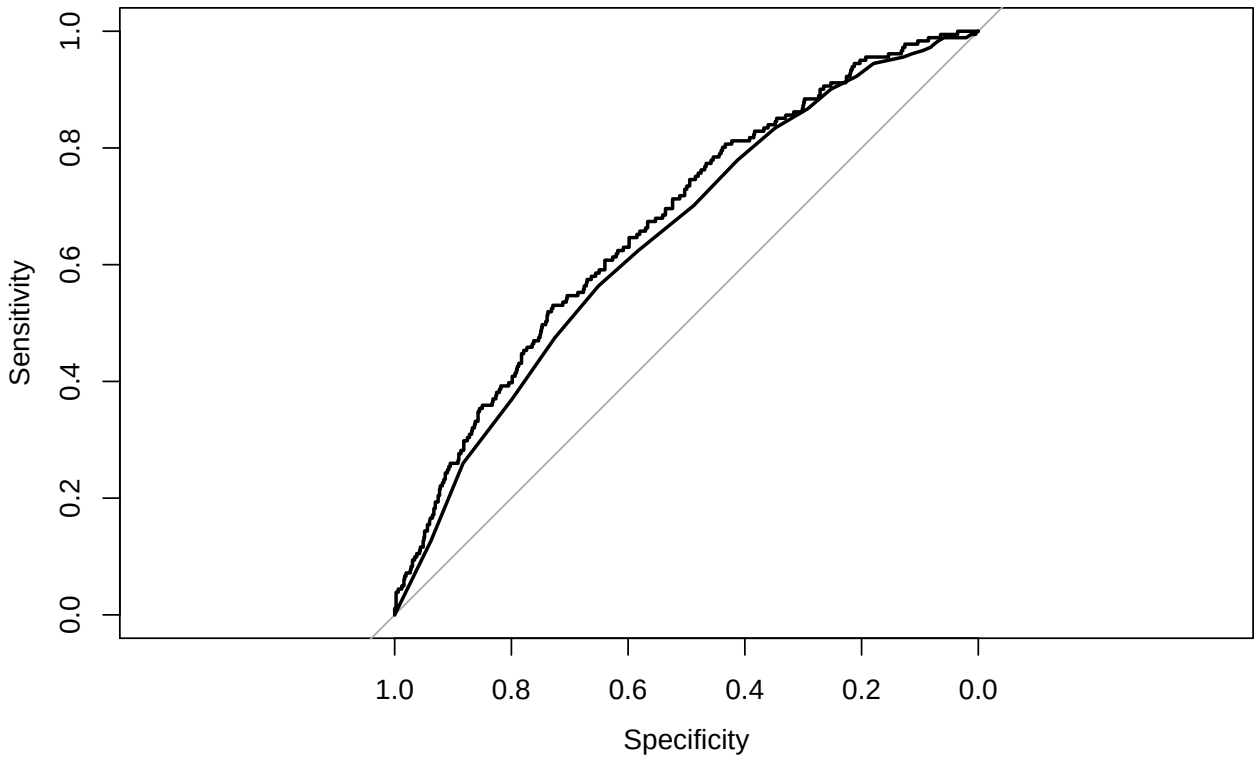
```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##            LR Chisq Df Pr(>Chisq)
## fico         5.8760  1    0.01535 *
## int_rate     7.8866  1    0.00498 **
## emp_length   7.2645 10    0.70026
## annual_inc   9.2386  1    0.00237 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Within this model, each variable is significant at the 0.05 level except the length of employment. We can kick out that variable to create a more parsimonious, 3 variable, model. We will use this as our final model.

```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##            LR Chisq Df Pr(>Chisq)
## fico         7.1142  1  0.0076476 **
## int_rate     8.2443  1  0.0040880 **
## annual_inc  11.1725  1  0.0008302 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can test the accuracy of this model in a similar way as before. We add the plot of the new model's ROC curve to the existing model's ROC curve plot, and compare them. The new curve is chunkier because it is comparing 1000 points vs 9000. However, we see that this is a better fit, with ROC of 0.67206673007778

```
##
## Call:
## roc.formula(formula = default ~ pred, data = ., )
##
## Data: pred in 819 controls (default FALSE) < 181 cases (default TRUE).
## Area under the curve: 0.6459
```




```
##
## Call:
## roc.formula(formula = default ~ pred, data = .)
##
## Data: pred in 819 controls (default FALSE) < 181 cases (default TRUE).
## Area under the curve: 0.6721
```

Thus, our final model is given by these summary statistics. Each variable is significant, and each has a clear interpretation.

term	estimate	std.error	statistic	p.value
(Intercept)	5.7793	3.2999	1.7513	0.0799
fico	-0.0113	0.0044	-2.5551	0.0106
int_rate	0.0808	0.0279	2.8929	0.0038
annual_inc	-0.0090	0.0030	-3.0508	0.0023

4: A business decision to make

Our expected value is, for any given loan i,:

$$EV_i = \frac{1-(1+r_i)^{-n}}{r_i} \times \Pr(payoff \mid X_i) \times A_i$$

Then, $\frac{1-(1+r_i)^{-n}}{r_i}$ is a standard present value of annuity formula, where r_i is the interest rate we charge and n is the number of years that the loan will run. Finally, $\Pr(payoff \mid X_i)$ is the probability that the user will pay off the loan, given some characteristics X_i at the user level, and A_i is the amount of the loan. Note that

the payoff probability comes from the model which we estimate in part 3. Additionally, note that we assume $n = 5$ for Lending Club loans.

This is a reasonable function because we care about the interest to be earned over the life of the loan, the chance of default, and the total amount of the loan. This is simplified somewhat because we have both liquidity constraints and investment constraints - in the absence of investment constraints we would allocate some money to this risky loan portfolio but also to a risk-free or market portfolio, and in the absence of liquidity constraints we would back each loan with a positive expected value. We additionally assume that loans are either fully paid off or fully written off, which isn't a reasonable assumption in the real world, since we can usually expect to recoup some of the value of a failed loan.

On these 100 loans, we expect a 4% return.

One improvement we could make to our model is to add a prior distribution on when loans are likely to default. We might expect the curve to be concave down, increasing then decreasing. This can give us a better idea of when the loans are likely to default, and then give us a better idea of how much we can actually earn from them. But: we don't have any data about that, and since our goal is to rank rather than predict the actual value, we should be fine wits this model.