London Business School

Data Science

Kamalini Ramdas & Kanishka Bhattacharya

www.london.edu

Classroom First

Rules of Engagement

!

- **Roomies** - keep your face covering on **at all times** while in the lecture theatre
- **Roomies** - touch in to the attendance monitoring system SEAtS using the card readers **before entering** the lecture theatre.
- **Zoomies** – your live attendance will be monitored through Zoom.
- **All students (Roomies and Zoomies)** should have laptops with webcams that run Zoom

To maximise your Zoom experience:

- Join the Zoom meeting **without** audio and if you have joined with audio, select “Leave Audio”
- Add “R” (for Roomie) in front of your name if you’re in the LT and “Z” (for Zoomie) if you are remote
- Use a headset with a microphone (noise cancelling is better) for breakout sessions
- Turn **on** your camera
- Open both the ‘Participant’ and ‘Chat’ windows and in the View Options menu select “Side-by-side” mode
- Zoomies should raise your hand (in Zoom) if you have a question and wait to be asked to speak
- For a better audio experience only 1 person should speak at any one time
- If you have a question you can also write it in the Zoom chat window
- Address technical issues in a private message to the Facilitator in the Zoom chat window

London Business School

Kamalini Ramdas

BIO

- BSc. (Hons) in Mathematics, St. Stephens College, Delhi, MS in Operations Research, U. of Delaware, Ph.D in Operations Management, Wharton
- Faculty at UT Austin for 2 years, Darden, U. of Virginia for 12 years, LBS for 13 years
- Research: Operational innovation (design / evaluation of radical operational innovations in healthcare, ICT and other areas)
- Teaching: MBA, EMBA Operations Management, EMBA Entrepreneurship, Business Model Experiments (elective) and executive education.

London Business School

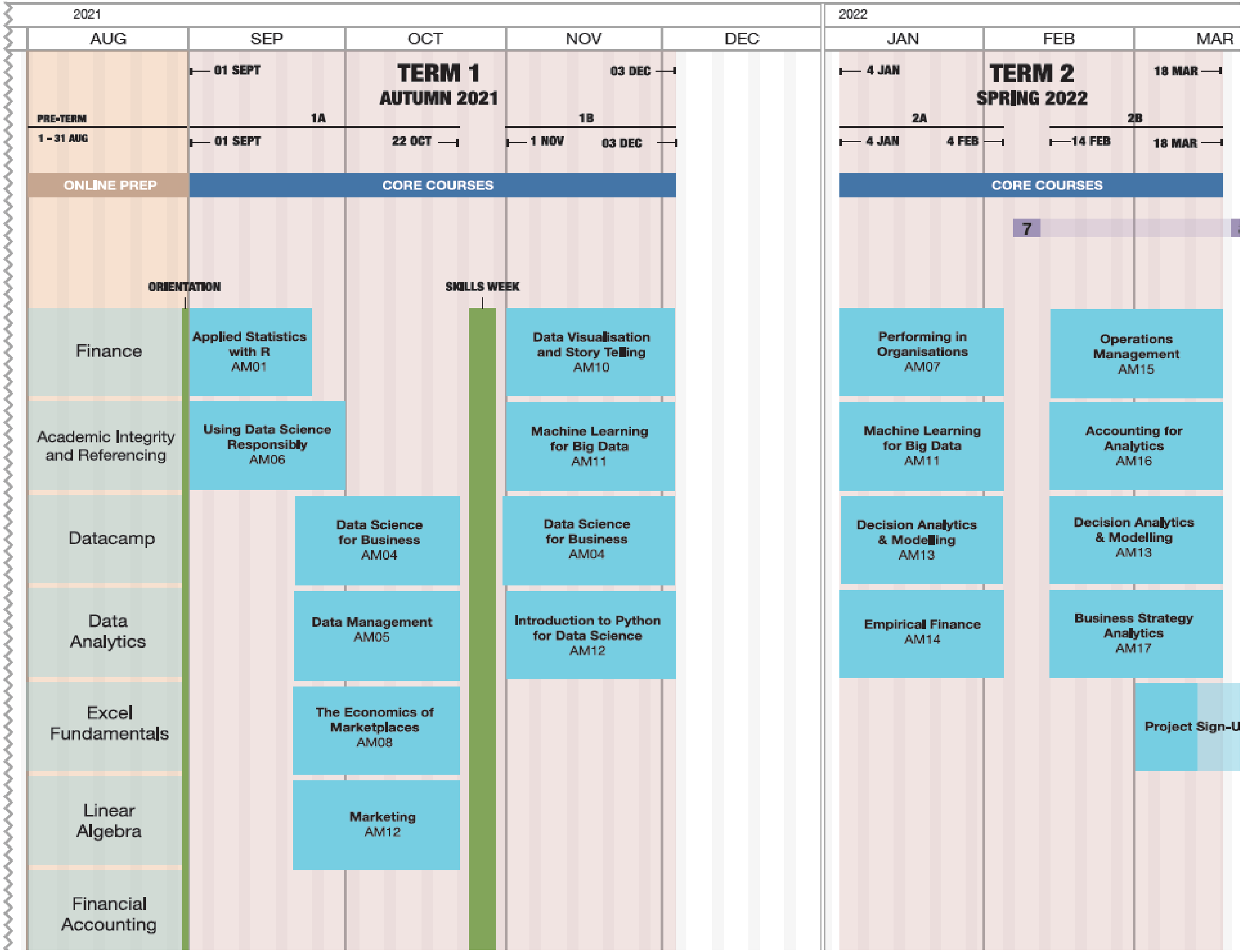
Kanishka Bhattacharya

BIO

- DPhil in Applied Statistics (in Statistical Genomics) and an MSc in Applied Statistics from Oxford University.
- research interests include building (novel) computationally efficient statistical algorithms for large scale compute challenges in the world of Statistical Genomics.
- With over 15 years of experience in statistical analysis of large scale, real life, and noisy datasets, Kanishka has developed cutting edge data science solutions in the domains of finance, retail, human genetics, epidemiology, media and advertising.
- worked as a Quant for two hedge funds and a digital advertising start up, alongside leading teams/practices and delivering data science programs for large global consulting firms.

London Business School

MAM2022 Programme Overview



MAM Overview

Skills Courses

- Intro to R
- Intro to Python
- Data Management
- Excel
- Interpersonal skills

Experiential Learning

- London LAB
- Global Immersion Field Trips (GIFTs)

Core Analytics courses

- Descriptive Analytics → What has happened and why?
  - Applied Statistics
  - Visualization
- Predictive Analytics → What will happen in the future?
  - Data Science
  - Machine Learning for Big Data
- Prescriptive Analytics → What should we do about it?
  - Decision Analysis & Modelling

Management Applications

- Marketing, Operations, Finance, Accounting

Management Background

- Using Data Science Responsibly
- Economics of Marketplaces
- Performing in Organizations
- Business Strategy

Course contents (first part of the course – Kamalini)

- Session 1: The Art & Science of Regression Models For Prediction
- Session 2: More on Using Linear Regression For Prediction
- Session 3: Workshop I – Engineer an algorithm that sets interest rates for new Lending Club loans
  - Group assignment 1, due 6 days after the workshop
- Session 4: Classification using Logistic Regression
- Session 5: Workshop – Invest in a portfolio of Lending Club loans
  - Individual project 1, due 13 days after the end of the workshop

Course contents (second part of the course – Kanishka)

- See canvas syllabus

The Lending Club case

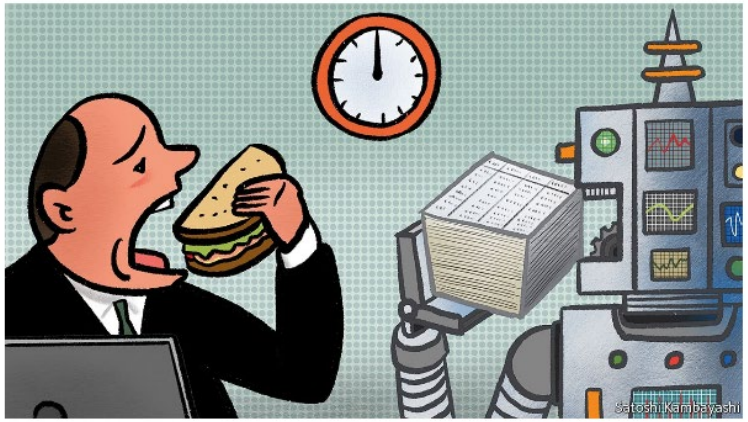
Engineer an algorithm that sets interest rates for new Lending Club loans

– Why would such an algorithm be useful?

Unshackled algorithms

Machine-learning promises to shake up large swathes of finance

In fields from trading to credit assessment to fraud prevention, machine-learning is advancing



Print edition | Finance and economics ›

May 25th 2017

MACHINE-LEARNING is beginning to shake up finance. A subset of artificial intelligence (AI) that excels at finding patterns and making predictions, it used to be the preserve of technology firms. The financial

Innovation transforms lending

Lending Club is the world's largest marketplace connecting borrowers and investors, where consumers and small business owners lower the cost of their credit and enjoy a better experience than traditional bank lending, and investors earn attractive risk-adjusted returns.[4]

Here's how it works:

- **Customers interested in a loan complete a simple application at LendingClub.com**
- **We leverage online data and technology to quickly assess risk, determine a credit rating and assign appropriate interest rates. Qualified applicants receive offers in just minutes and can evaluate loan options with no impact to their credit score**
- **Investors ranging from individuals to institutions select loans in which to invest and can earn monthly returns**

The entire process is online, using technology to lower the cost of credit and pass the savings back in the form of lower rates for borrowers and solid returns for investors.

The Art & Science of Using Linear Regression for Prediction

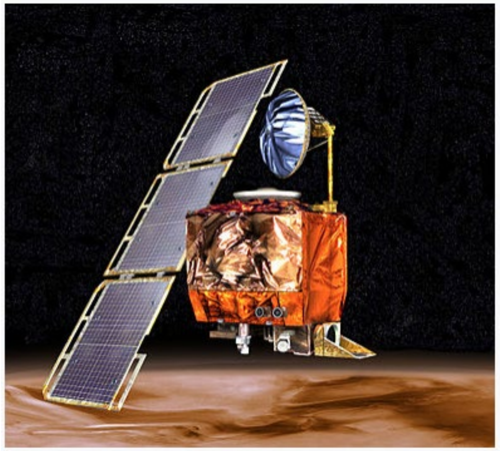
- **ICE the data: Inspect, Clean, Explore**

- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression

- **Choose a model and use it responsibly!**

ICE the data

- **Inspect**
  - Understand each variable definition
  - Note the units of measurement (e.g., do not confuse lbs with Kg)
  - Identify any data issues (missing values, incorrect entries, etc.)
- **Clean**
  - Decide what to do with missing values
  - Code variables correctly (e.g., numerical variables, factors, dates)
  - Rearrange the data set if needed to be tidy (each row should be one unit of analysis, each column should be one feature associated with this unit of analysis)
- **Explore**
  - Articulate hypotheses as to what may be happening – discuss with colleagues / experts
  - Create correlation charts, histograms, scatter plots, etc
- **Most project failures are due to improper ICE-ing!**



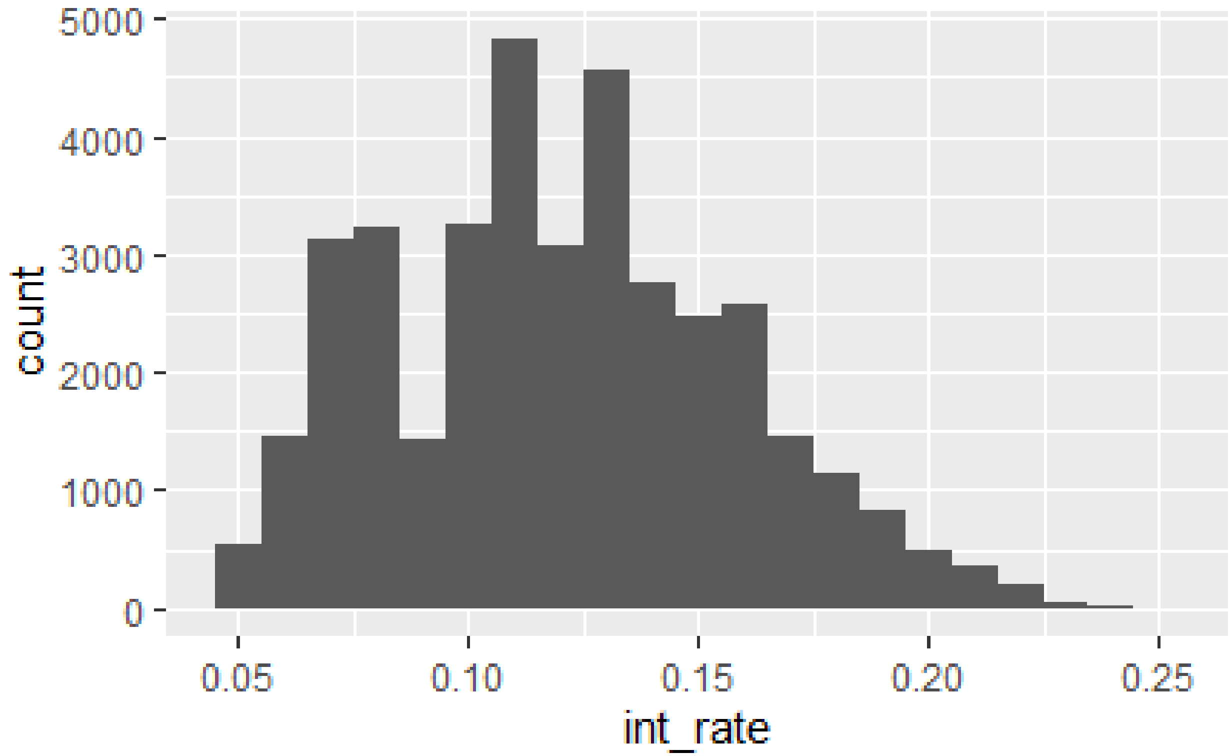
Artist's conception of the Mars Climate Orbiter

London Business School

ICE the data

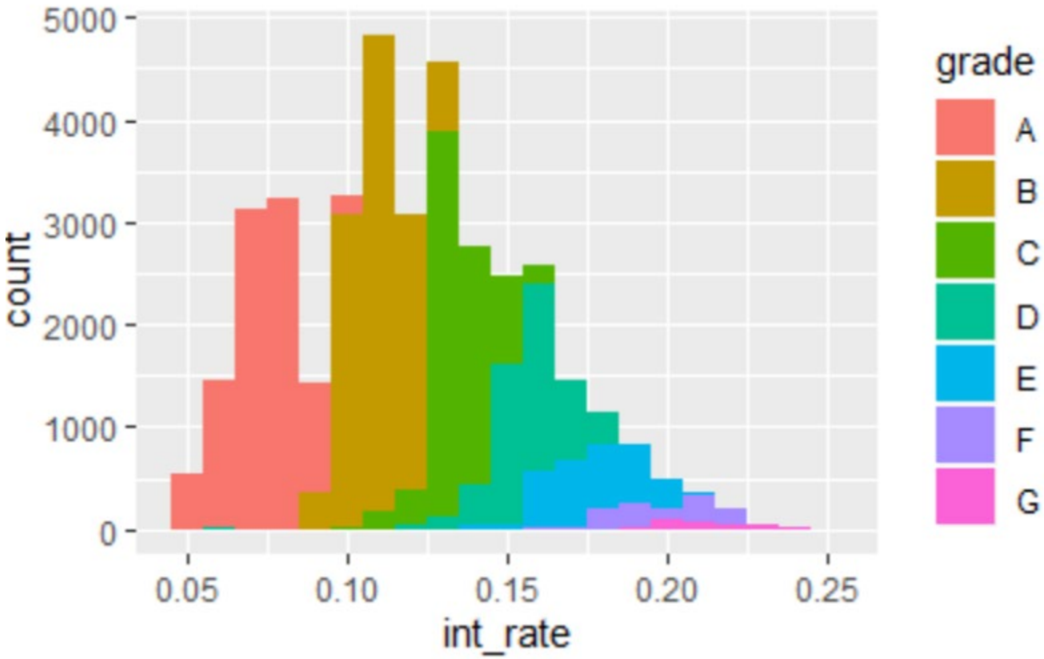
- Lending Club – our goal is to predict interest rates of new loans

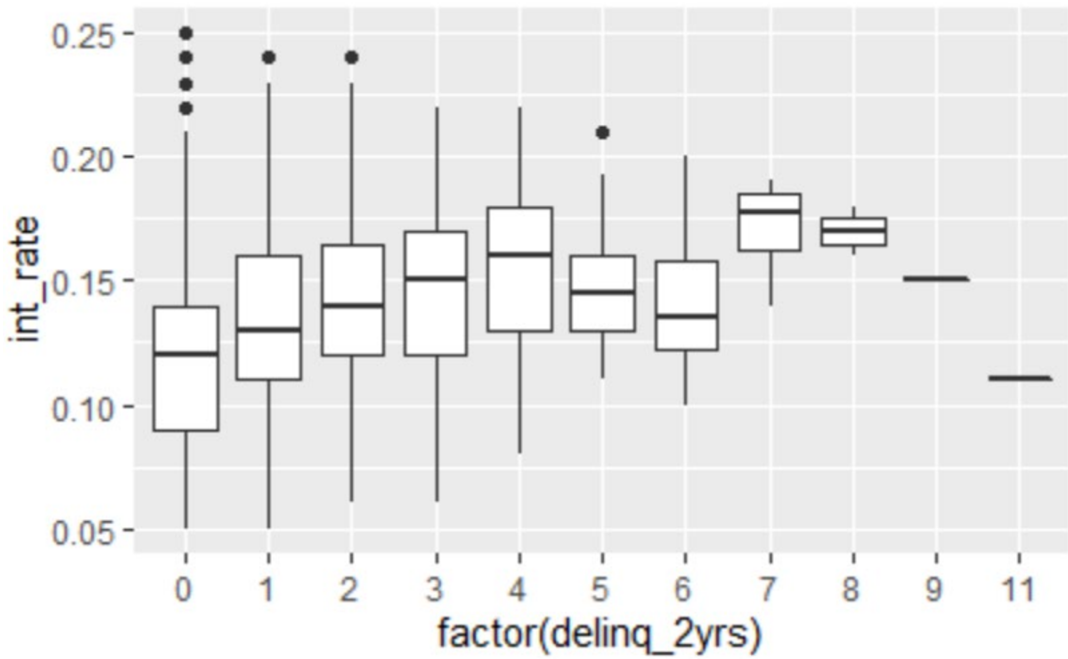
- Histogram: Based on this, what would be reasonable guess for the interest rate of a new loan?



ICE the data

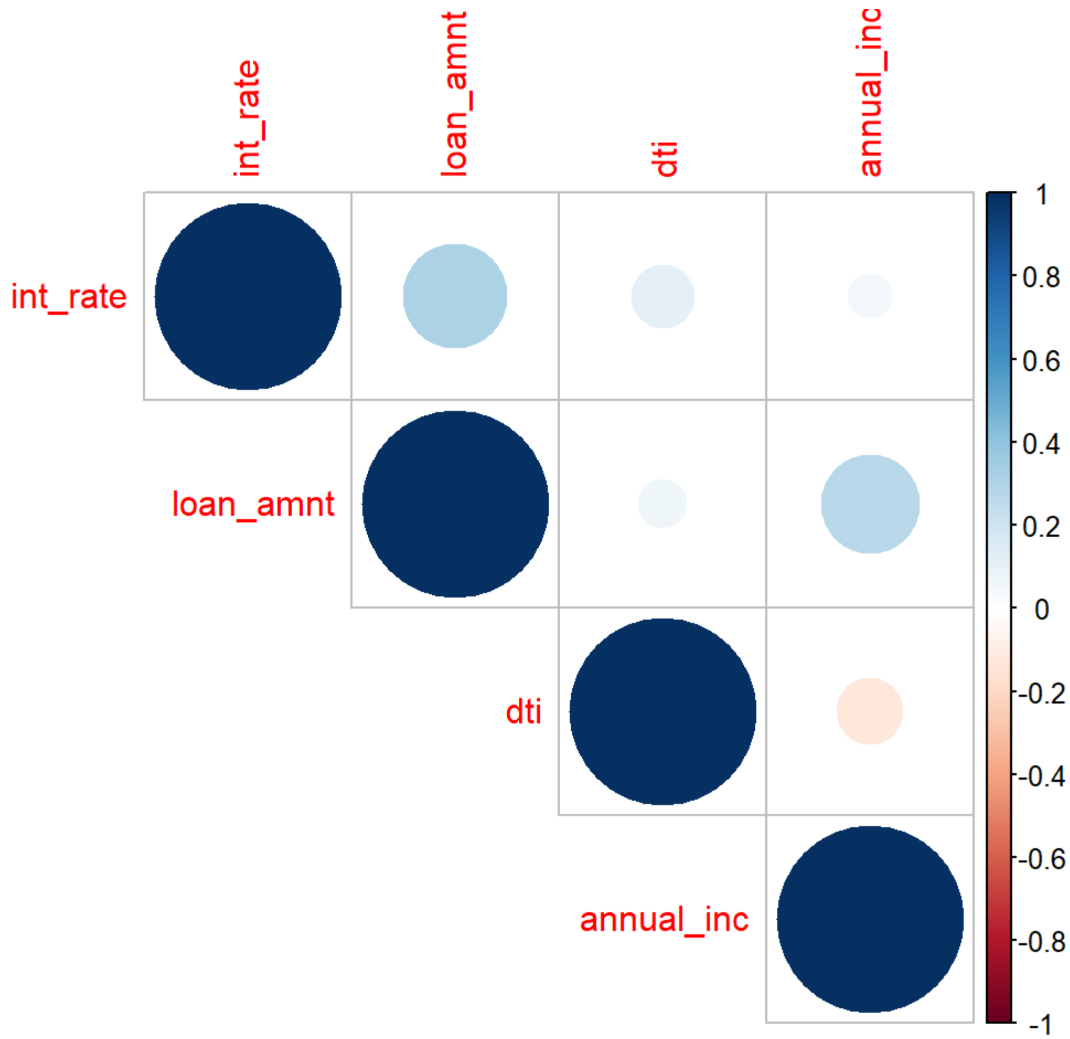
- Based on this, what would be reasonable guess for the interest rate of a new loan?





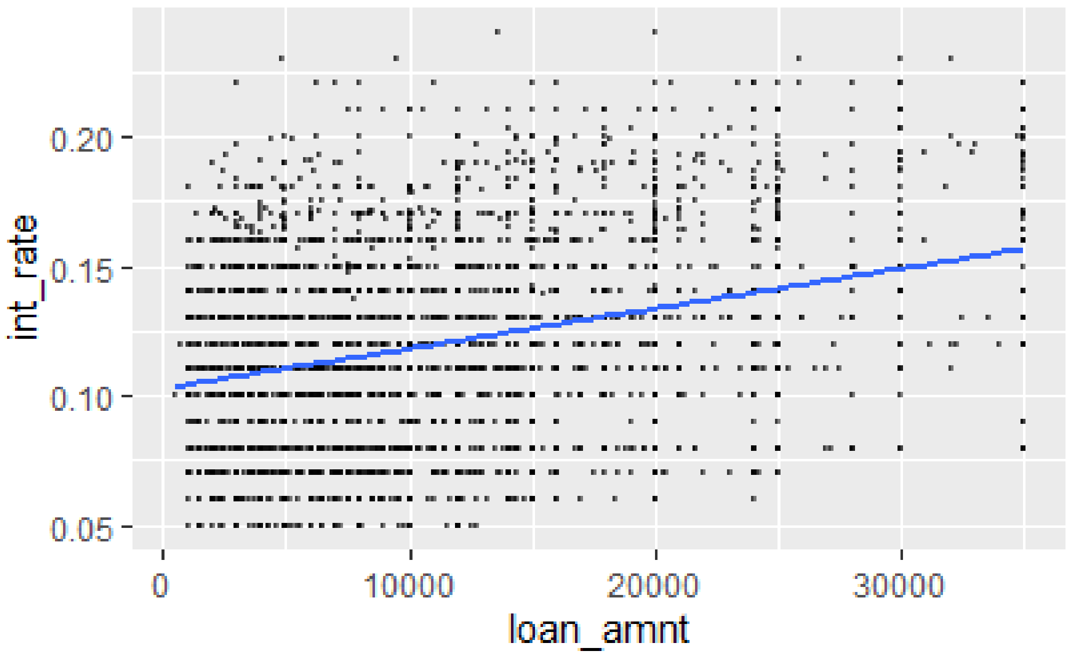
ICE the data

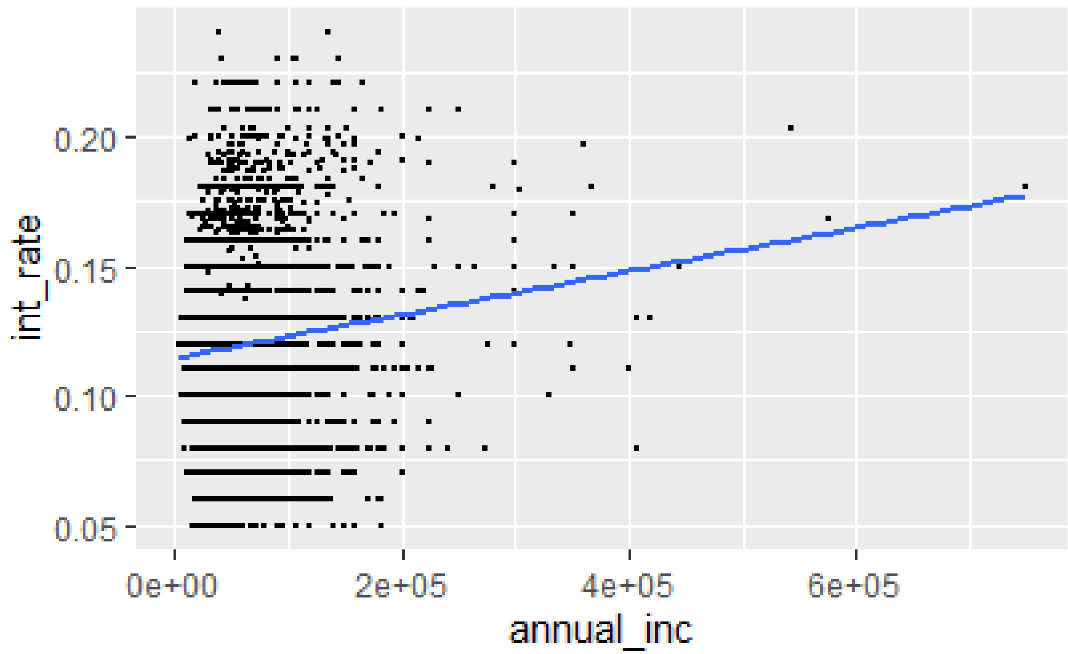
- **Correlation table: Does this help you form hypotheses?**



ICE the data

- Based on this, what would be reasonable hypotheses for the drivers of interest rates?





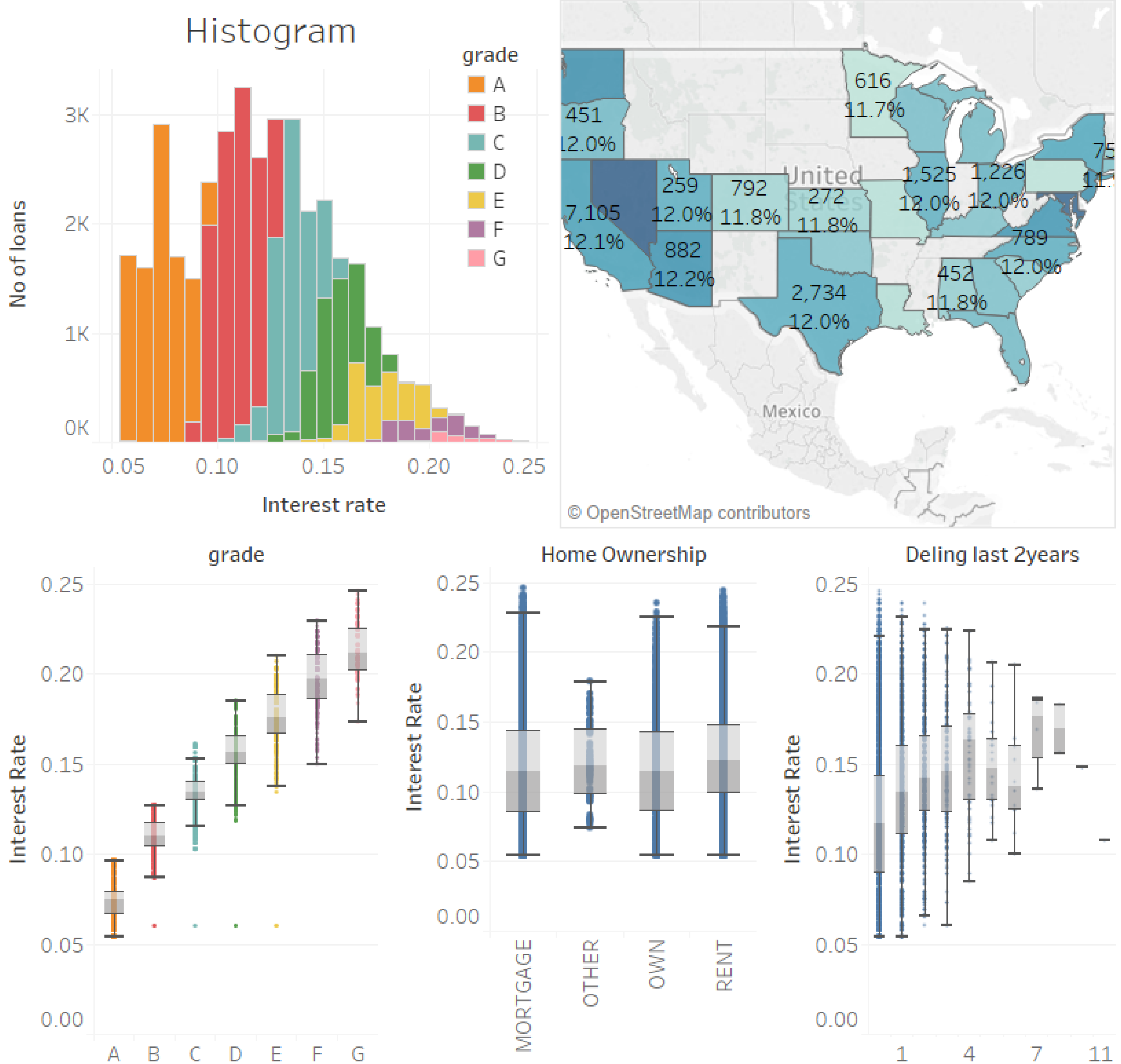
Using R

```r
#histogram of interest rtes
ggplot(lc_clean, aes(x=int_rate))+
  geom_histogram(binwidth=0.01)+scale_x_continuous(labels = scales::percent) +labs(x="Interest Rate")

#histogram with colour for different grades and term.
ggplot(lc_clean, aes(x=int_rate, fill=grade))+ geom_histogram(binwidth=0.01)+scale_x_continuous(labels = scales::percent)+ labs(x="Interest Rate")

ggplot(lc_clean, aes(x=int_rate, fill=term))+ geom_histogram(binwidth=0.01)+scale_x_continuous(labels = scales::percent)+ labs(x="Interest Rate")

#density plot with colour for different grades.
ggplot(lc_clean, aes(x=int_rate, fill=grade, alpha = 0.2))+
  geom_density()+
  facet_grid(rows = vars(grade))+
  theme_bw()+
  theme(legend.position = "none")+
  scale_x_continuous(labels = scales::percent)+ labs(x="Interest Rate")

#boxplot with colour for different home_ownerhsip
ggplot(lc_clean, aes(x=home_ownership, y=int_rate, colour=home_ownership))+
  geom_boxplot()+
  theme_bw()+
  theme(legend.position = "none")+
  coord_flip()+ scale_y_continuous(labels=scales::percent)+
  labs(y="Interest Rate", x="Home Ownership")
```

```r
#scatter plots
ggplot(lc_clean[seq(1, nrow(lc_clean), 10), ] , aes(y=int_rate, x=loan_amnt)) +
  geom_point(size=0.1, alpha=0.5)+
  geom_smooth(method="lm", se=0) + labs(y="Interest Rate", x="Loan Amount ($)")


ggplot(lc_clean[seq(1, nrow(lc_clean), 10), ] , aes(y=int_rate, x=annual_inc)) +
  geom_point(size=0.1)+
  geom_smooth(method="lm", se=0) +labs(y="Interest Rate", x="Annual Income ($)")

#box plot for delinquencies
ggplot(lc_clean , aes(y=int_rate, x=delinq_2yrs, colour= delinq_2yrs)) +
  geom_boxplot()+
  # geom_jitter()+
  theme_bw()+
   scale_y_continuous(labels=scales::percent)+
  theme(legend.position = "none")+
  labs(
    title = "Do delinquencies in the last two years impact interest rate charged?",
    x= "Number of delinquecies in last two years", y="Interest Rate"
  )
```

See “Lending_Club_session1_and2.Rmd” for a guide to these graphics

You can also use Tableau!



Wait until you've had the visualization course

Visualization is all I need…NOT

- **Visualization is great for**
  - Understanding the data
    - Get a sense of what different variables mean
    - Investigate if there are any data-quality issues
  - Quickly (and very roughly!) testing intuition and generating hypotheses

- **Visualization is not so good at**
  - **Inference:** Do interest rates increase due to past delinquencies once we control for credit rating and sampling error? Does higher income lead to higher interest rate once we control for loan amount and sampling error? How much would the interest rate decrease if you increased the loan amount by $10K?
    - Here we care about the estimated coefficients and their errors
  - **Predicting the future: What interest rate should a new loan be charged? How likely is it that this estimate is off?**
    - Here we care about the ability to make accurate predictions (out of sample prediction error)

The Art & Science of Using Linear Regression for Prediction

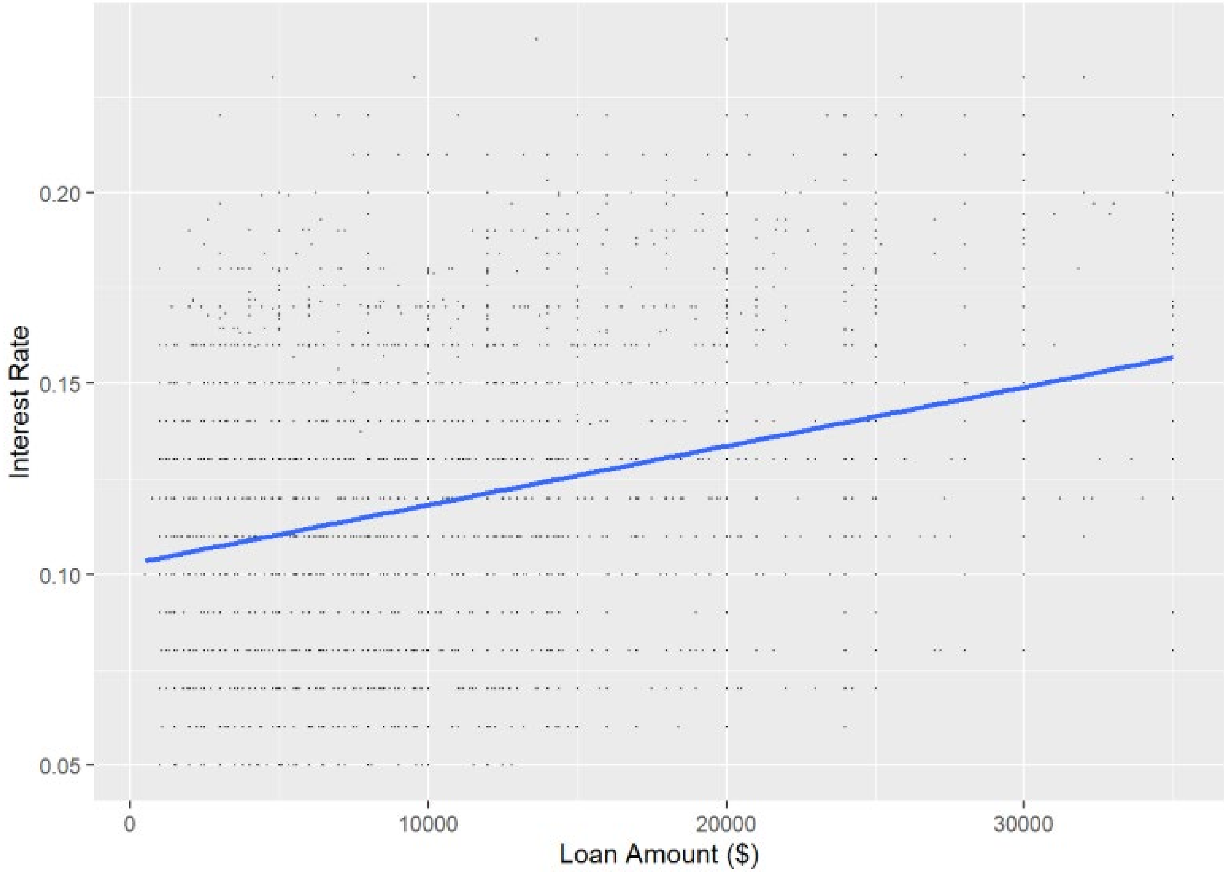
- **ICE the data: Inspect, Clean, & Explore**

- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression

- **Choose a model and use it responsibly!**

Let's start with a single variable

- **Is there a relationship between interest rate and loan amount?**
  - What would you expect this relationship to be?
  - How would investigate if there really is a relationship?



London Business School

Setting up a model

- **Express the relationship between the interest rate and the loan as a mathematical equation**

$$Int = \beta_0 + \beta_1 \times loanA + \epsilon$$

Dependent variable

Intercept (constant)

Coefficient (slope)

Explanatory (Independent) Variable/feature

Error term

  – Intercept β_0 is the expected interest rate for a loan with 0 loan amount

  – Slope β_1 is the expected increase in the interest rate when loan amount increases by 1 unit

  – ϵ is the part of the variation in interest rates that cannot be explained by loan amount. This is assumed to be a random variable with mean zero and variance σ^2

Estimate the model

$$Int = \beta_0 + \beta_1 \times loanA + \epsilon$$

- Estimation question: How do we choose the "best" line?
  - In other words, how do we choose the values of $\beta_0, \beta_1, \sigma^2$ that best describe the data?
- Since we are interested in forecasts, maybe choose the line that will
  - minimise sum (or average) forecast error? not useful
  - minimise sum of absolute deviations? possible
  - **minimise sum of squared deviations?** focuses on avoiding big forecast errors and minimises standard error of forecast (BLUE: Best Linear Unbiased Estimator)
- Can run Ordinary Least Squares (OLS) regression using software
  - Excel Data Analysis tool pack
  - R and the basic "*lm*" command or more likely using the ***caret*** library

London Business School

Least squares error algorithm

$$Int = \beta_0 + \beta_1 \times loanA + \epsilon$$

- Assume we use our sample to estimate β_0 and β_1 as $b_0 = 0.12, b_1 = 0.01$

- For the first loan, the residual would be
  - $e_1 = 0.1095 - (0.12 + 0.01 \times 5) = -0.0605$
  - Similarly, we can estimate $e_2, e_3, e_4, \ldots, e_{10,000}$
  - The sum of squared residuals is $RSS = (e_1^2 + e_2^2 + \cdots + e_{10,000}^2)$

- The least squares error principle asks
  - What values of b_0, b_1 make RSS the smallest?
  - Can use a "solver" to find these values
  - The problem turns out to be "simple" enough to solve analytically (i.e., we don't need to run a solver as there is a (complicated) formula we can use)

Loan	(K USD) Amount	Interest Rate
1	5	10.95%
2	2.5	14.27%
3	2.4	15.96%
4	10	13.49%
5	3	11.69%
6	5	15.00%
7	7	15.96%
8	8	18.64%
9	5.6	21.28%
10	5.375	12.69%
11	6.5	14.65%
12	12	17.69%

12 randomly chosen observations

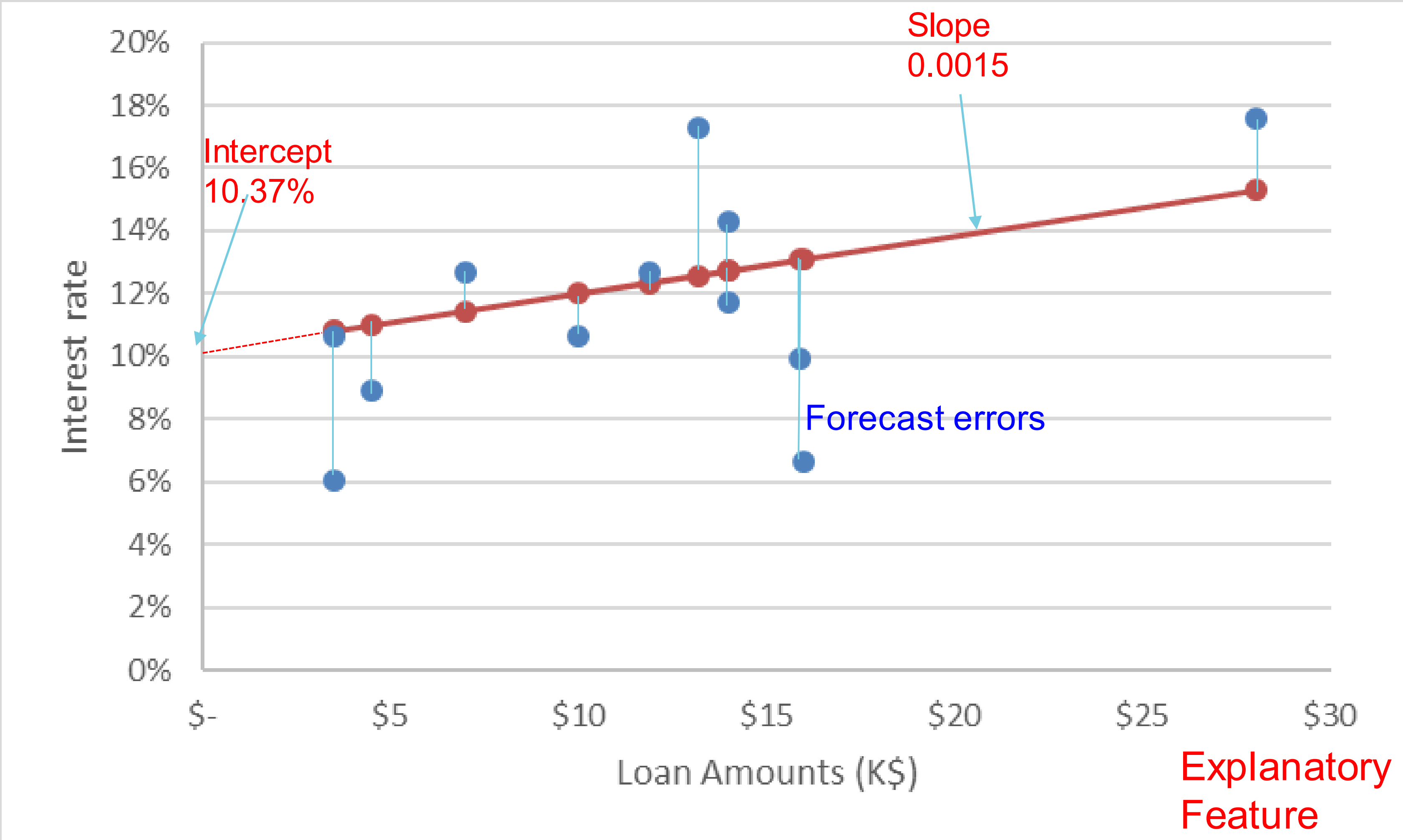
Mean interest rate =15.19%
Standard Deviation = 2.97%

Watch Video

Line of best fit: Minimize sum of squared errors

$$Int = 10.37\% + 0.0015 \times loanA + \epsilon$$

Dependent Variable (interest rate)



Explanatory Feature

Assess Goodness of Fit

Models fitted on the whole dataset using R

Model with only intercept

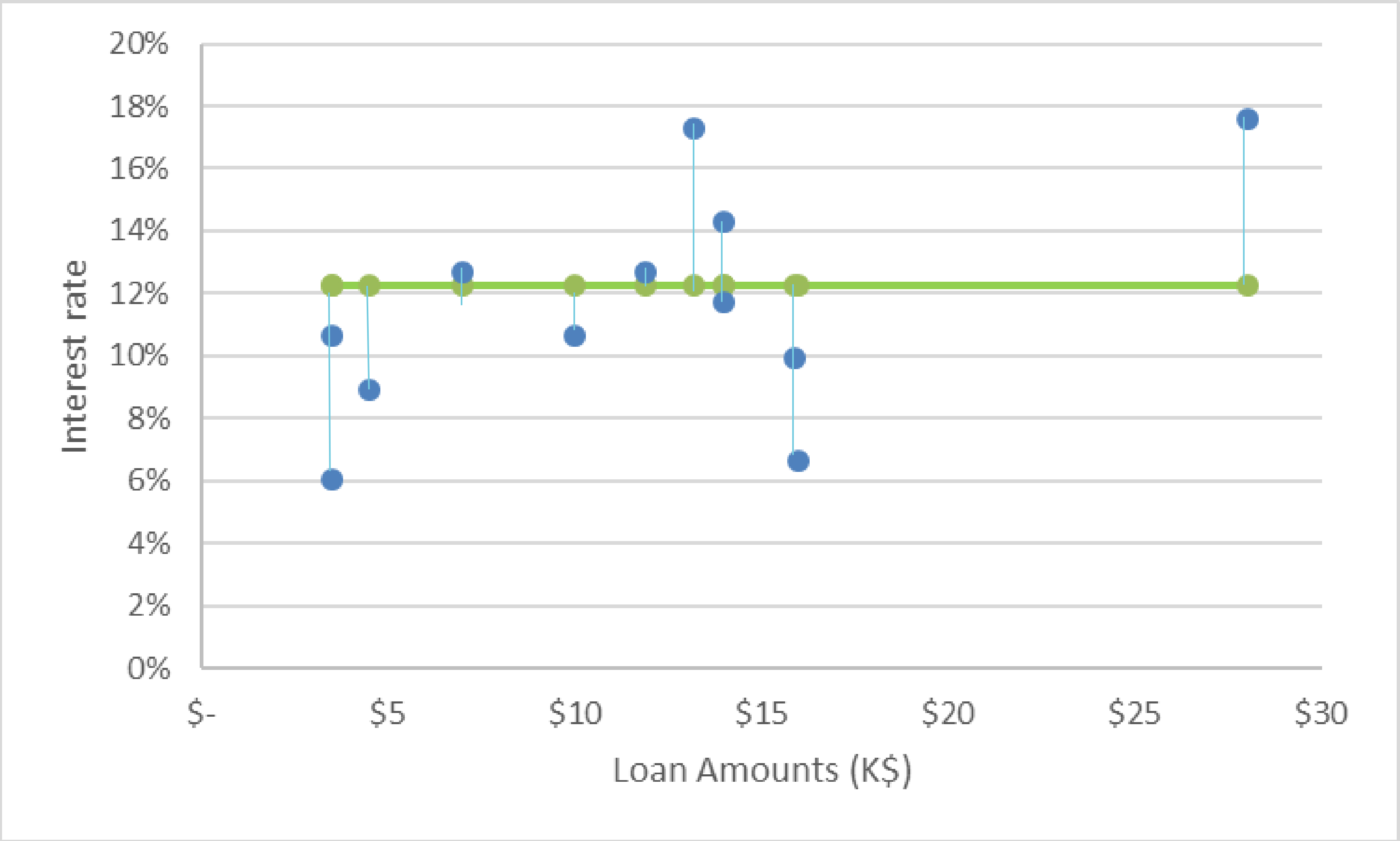
```
Call:
lm(formula = int_rate ~ 1, data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max
-0.070264 -0.030264 -0.000264  0.029736  0.129736

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.1202639  0.0001915     628   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03726 on 37868 degrees of freedom
```

Model with intercept & Loan amount

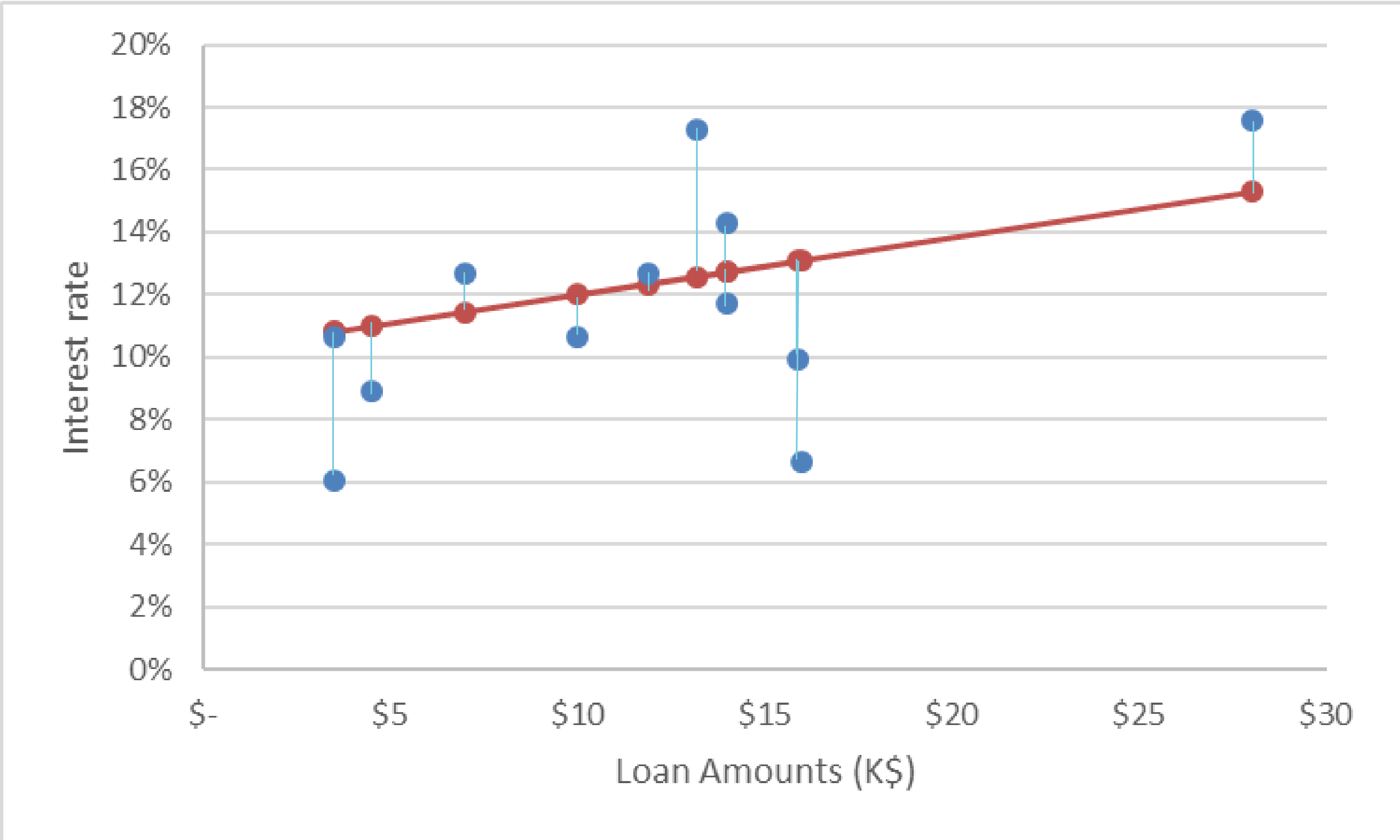
```
Call:
lm(formula = int_rate ~ loan_amnt, data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max
-0.089412 -0.028316 -0.001426  0.024900  0.128370

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.028e-01  3.292e-04  312.15   <2e-16 ***
loan_amnt   1.555e-06  2.438e-08   63.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03541 on 37867 degrees of freedom
Multiple R-squared:  0.09698,    Adjusted R-squared:  0.09695
F-statistic:  4067 on 1 and 37867 DF,  p-value: < 2.2e-16
```

- **R-squared (explained variance / total variance)**
  - proportion of the total variance "explained" by the model
  - adjusted: makes an adjustment for the number of variables in the model

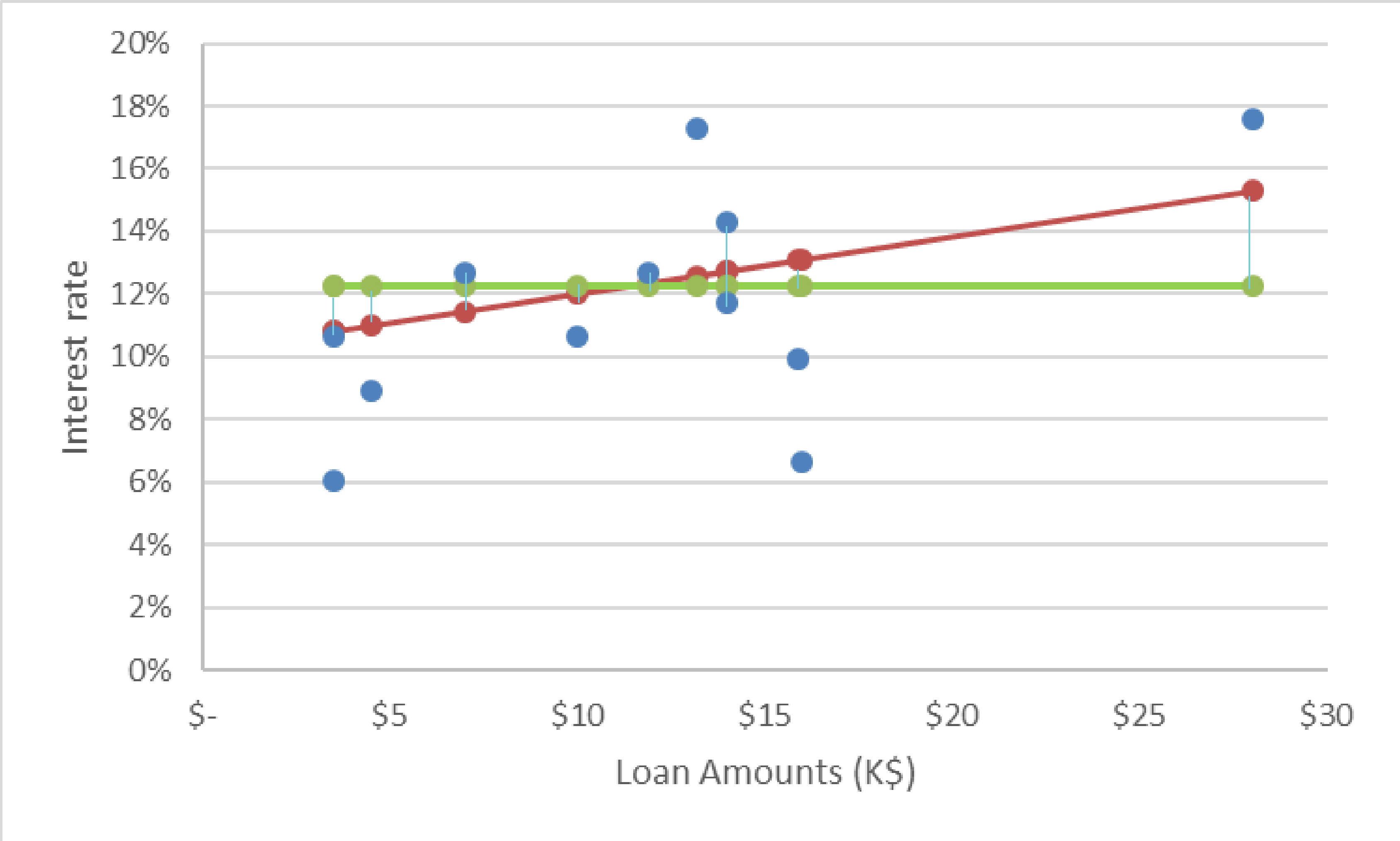
Total Variance
Model with only intercept



Model with intercept & Loan amount Residual Variance



Explained Variance



R^2 = Explained Variance/Total Variance

Significance of individual features

```
Call:
lm(formula = int_rate ~ loan_amnt, data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max
-0.089412 -0.028316 -0.001426  0.024900  0.128370

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.028e-01  3.292e-04  312.15   <2e-16 ***
loan_amnt   1.555e-06  2.438e-08   63.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.03541 on 37867 degrees of freedom
Multiple R-squared:  0.09698,    Adjusted R-squared:  0.09695
F-statistic:  4067 on 1 and 37867 DF,  p-value: < 2.2e-16
```

Regression standard error

Coefficients

Standard errors of coefficients

Probability that coefficient is zero

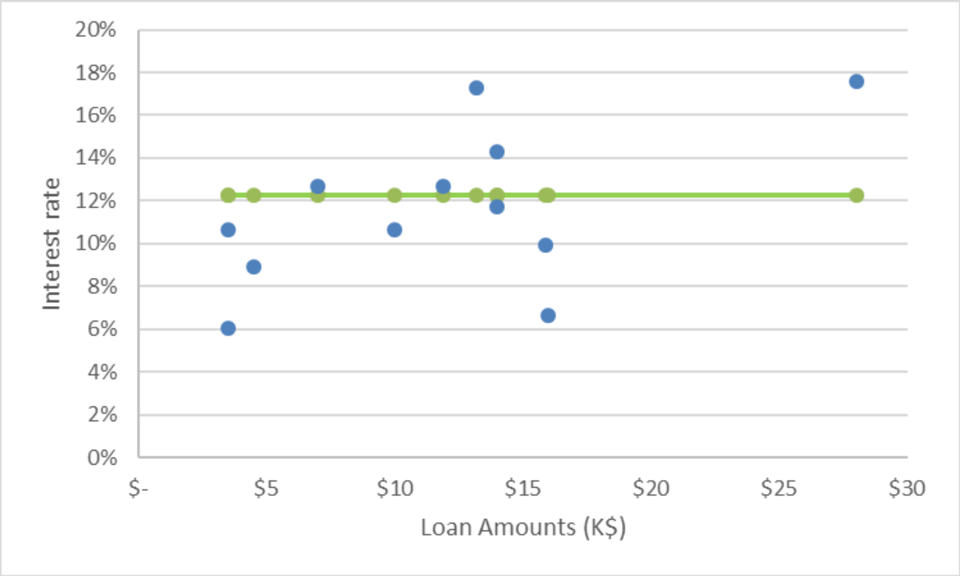
Goodness of fit

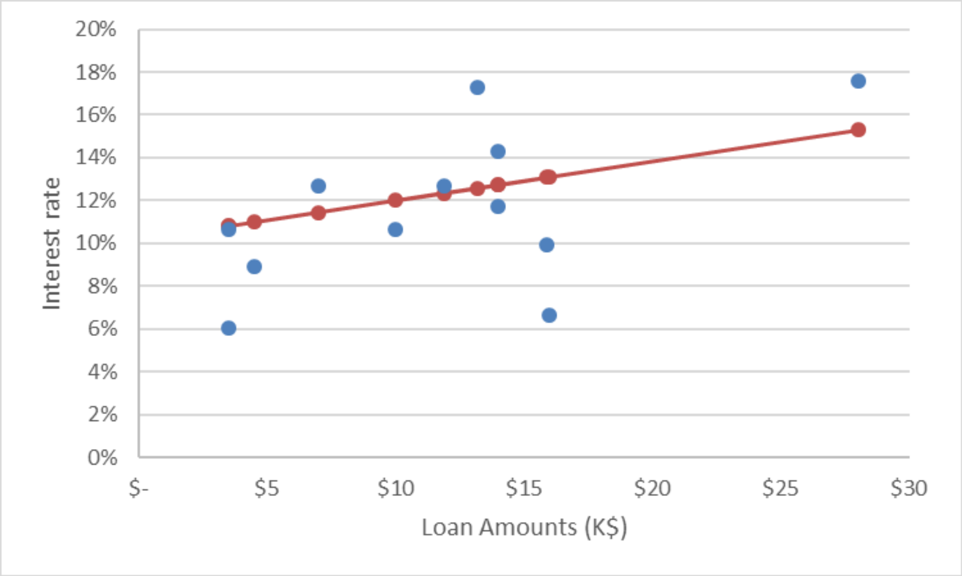
Probability that the model has no explanatory power

t-value: Coefficient divided by its Standard Error

Significance of individual feature

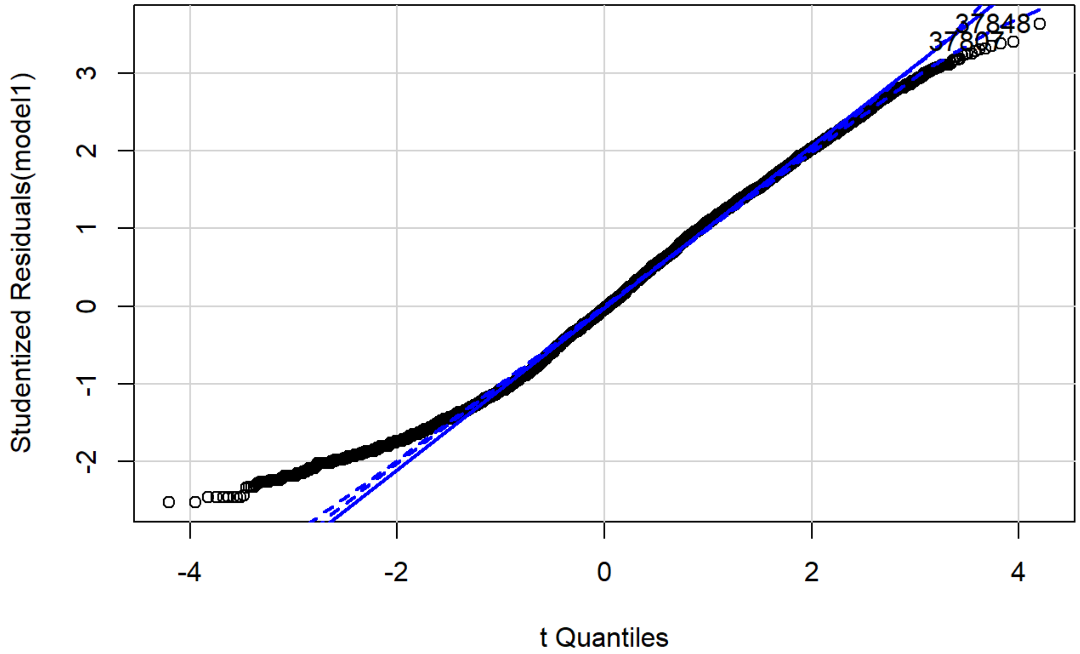
- **Is there a (linear) relationship between interest rate and loanA ?**
  - Yes $\Rightarrow$ slope (b) is different from zero
  - No $\Rightarrow$ slope (b) is zero (we obtained a non-zero slope by chance only)

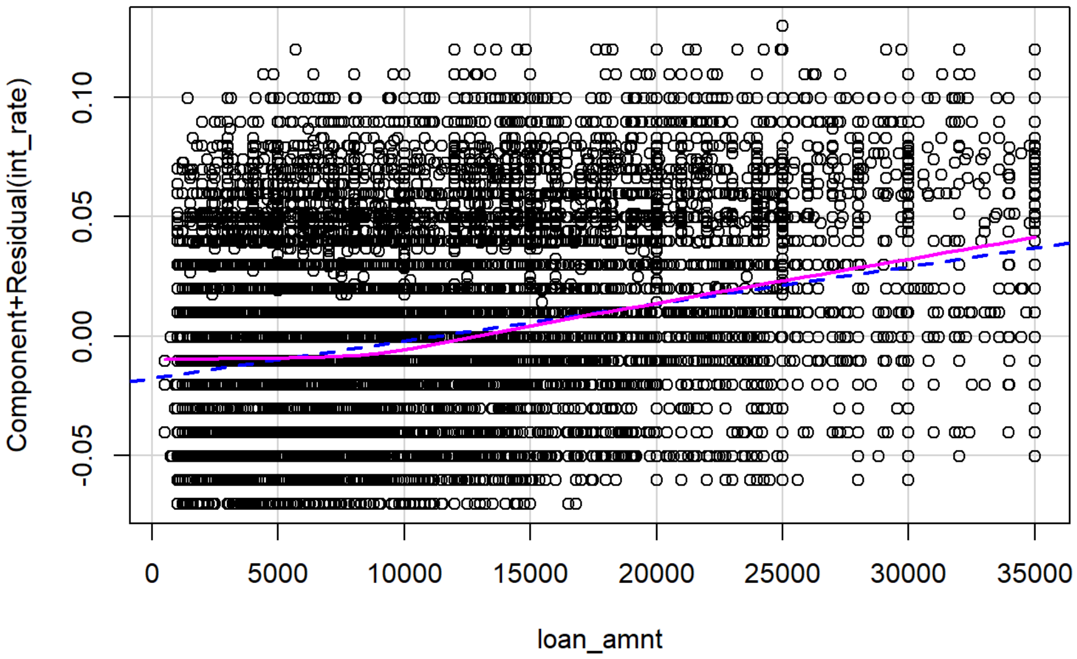




  - Test: Could b be equal to 0 ? (Could it be that there is no relationship ?)
    - compute t-statistic (b / standard error of b)
    - if t-statistic > 2, b is probably not zero (95% confident)
    - p-value = probability that b could be zero (if this is <5%, confident that $b \neq 0$)

Goodness of fit
Examining residuals





QQ plot (left) examines how well the residuals follow the normal distribution – if they don't then the standard errors estimated are not reliable

Residual Scatter plot (right) examines if the residuals are randomly distributed for different loan amounts – if they are not then perhaps there is a non-linear relationship → investigate

London Business School

Significance of individual features

```
Call:
lm(formula = int_rate ~ loan_amnt, data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max
-0.089412 -0.028316 -0.001426  0.024900  0.128370

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.028e-01  3.292e-04  312.15   <2e-16 ***
loan_amnt   1.555e-06  2.438e-08   63.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03541 on 37867 degrees of freedom
Multiple R-squared:  0.09698,    Adjusted R-squared:  0.09695
F-statistic:  4067 on 1 and 37867 DF,  p-value: < 2.2e-16
```

Regression standard error

Coefficients

Probability that coefficient is zero

Goodness of fit

Standard errors of coefficients

Probability that the model has no explanatory power

t-value: Coefficient divided by its Standard Error

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**

- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression

- **Choose a model and use it responsibly!**

Feature Engineering

- **Use information from the dataset**
  - Candidates: ***Term, income, dti, grade***, number of delinquencies, employment length, etc
    - Some of these are numerical others are factors. How do we use factor variables?
  - Interaction terms: *Perhaps the loan amount affects 36-month loans differently than 60-month loans***.** How do you model this?
    - Interactions between two factor features, a factor and a numerical feature, two numerical features
  - Non-linear terms: *Perhaps a small increase in the loan amount doesn't affect interest rate so much but a large increase does.* How would you model this?
    - ***Polynomial terms*** (powers of a feature) or any ***other non-linear transformation*** (better have a good reason for the non-linear transformation)
    - Dummy variable creation → converting a numerical variable into a factor variable (e.g., low, mid, high income, or ***deciles of income***). This is a non-parametric way of modelling non-linear relationships
- **Look for data outside your model**
- **Feature engineering is more of an art than science! Know your context (or work with people who do)!**

Classroom First

Rules of Engagement

!

- **Roomies** - keep your face covering on **at all times** while in the lecture theatre
- **Roomies** - touch in to the attendance monitoring system SEAtS using the card readers **before entering** the lecture theatre.
- **Zoomies** – your live attendance will be monitored through Zoom.
- **All students (Roomies and Zoomies)** should have laptops with webcams that run Zoom

To maximise your Zoom experience:

- Join the Zoom meeting **without** audio and if you have joined with audio, select “Leave Audio”
- Add “R” (for Roomie) in front of your name if you’re in the LT and “Z” (for Zoomie) if you are remote
- Use a headset with a microphone (noise cancelling is better) for breakout sessions
- Turn **on** your camera
- Open both the ‘Participant’ and ‘Chat’ windows and in the View Options menu select “Side-by-side” mode
- Zoomies should raise your hand (in Zoom) if you have a question and wait to be asked to speak
- For a better audio experience only 1 person should speak at any one time
- If you have a question you can also write it in the Zoom chat window
- Address technical issues in a private message to the Facilitator in the Zoom chat window

All workshops will be virtual (Kamalini & TA's will join virtually) however, we have booked the following rooms below for those who will be on campus.

STUDY GROUP	ROOM
1	AG01
2	AG02
3	AG03
4	AG04
5	AG05
6	AG06
7	AG07
8	AG08
9	AG09
10	AG10
11	AG11
12	AG12

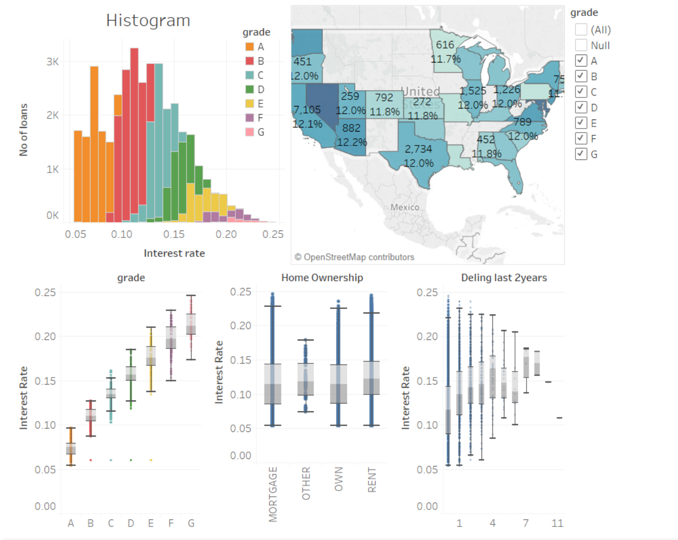
Office Hours- Virtual

Date & Time	Meeting Link	Meeting ID	Passcode
13/10/21 16:30-17:30	https://zoom.us/j/97620086295	976 2008 6295	703473
20/10/21 16:00-17:00	https://zoom.us/j/91582880144	915 8288 0144	372574
27/10/21 16:00-17:00	https://zoom.us/j/91582880144	915 8288 0144	372574
03/11/21 16:00-17:00	https://zoom.us/j/91582880144	915 8288 0144	372574

London Business School

Recap from last time

- Our goal is to come up with an algorithm that predict the interest rate that lending club will charge to new loans
  - We ICEd the Data: Inspect, Clean, Explore
  - We used the method of **ordinary least squares** to fit a model that uses loan amount as a predictor
    - This is the model that reduces the prediction error as much as possible
- How good is the model?
  - Is the loan amount statistically significant?
  - Does the model have any explanatory power?
  - How much error remains in our predictions?
- What can we do to improve predictions?




```
Call:
lm(formula = int_rate ~ loan_amnt, data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.089412 -0.028316 -0.001426  0.024900  0.128370 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.028e-01  3.292e-04  312.15   <2e-16 ***
loan_amnt   1.555e-06  2.438e-08   63.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03541 on 37867 degrees of freedom
Multiple R-squared:  0.09698,    Adjusted R-squared:  0.09695 
F-statistic:  4067 on 1 and 37867 DF,  p-value: < 2.2e-16
```

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**

- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression

- **Choose a model and use it responsibly!**

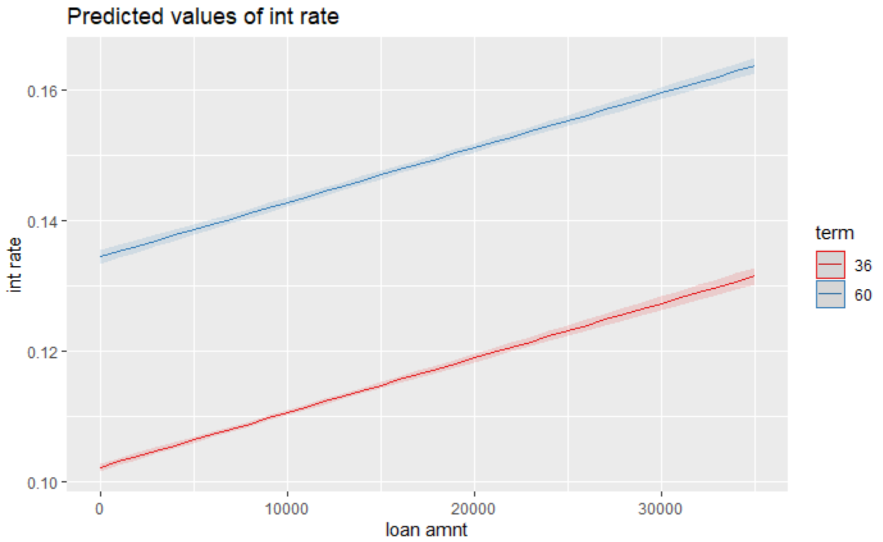
Feature Engineering

- **Use information from the dataset**
  - Candidates: ***Term, income, dti, grade***, number of delinquencies, employment length, etc
    - Some of these are numerical others are factors. How do we use factor variables?
  - Interaction terms: *Perhaps the loan amount affects 36-month loans differently than 60-month loans*. How do you model this?
    - Interactions between two factor features, a factor and a numerical feature, two numerical features

Factor Features

- Have added *dti* and *annual income* as features. Was this a good idea?
- Loans can be taken for either 36 or 60 months
  - Create a dummy variable that takes the value 1 if the loan is 60 months or 0 otherwise – add this variable to the model
  - No need to create another dummy that takes the value 1 if the loan is 36 months and zero otherwise (The two dummy variables convey they same information – they are colinear)
  - The coefficient of the "Term60" can be interpreted as the average interest rate difference between 60 month and 36 month loans: on average, a 60 month loan has a ***3.22%*** higher interest rate than a 36 month loan!
  - Did you expect it to be positive? Does it have explanatory power?
- How much better is this model?

```
Call:
lm(formula = int_rate ~ loan_amnt + term + dti + annual_inc,
    data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max
-0.091963 -0.027528  0.000165  0.024122  0.119277

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.707e-02  4.713e-04 205.967   <2e-16 ***
loan_amnt    8.384e-07  2.521e-08  33.261   <2e-16 ***
term60       3.226e-02  4.070e-04  79.269   <2e-16 ***
dti          3.830e-04  2.546e-05  15.040   <2e-16 ***
annual_inc  -1.840e-10  2.869e-09  -0.064    0.949
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03264 on 37864 degrees of freedom
Multiple R-squared:  0.233,	Adjusted R-squared:  0.2329
F-statistic:  2875 on 4 and 37864 DF,  p-value: < 2.2e-16
```

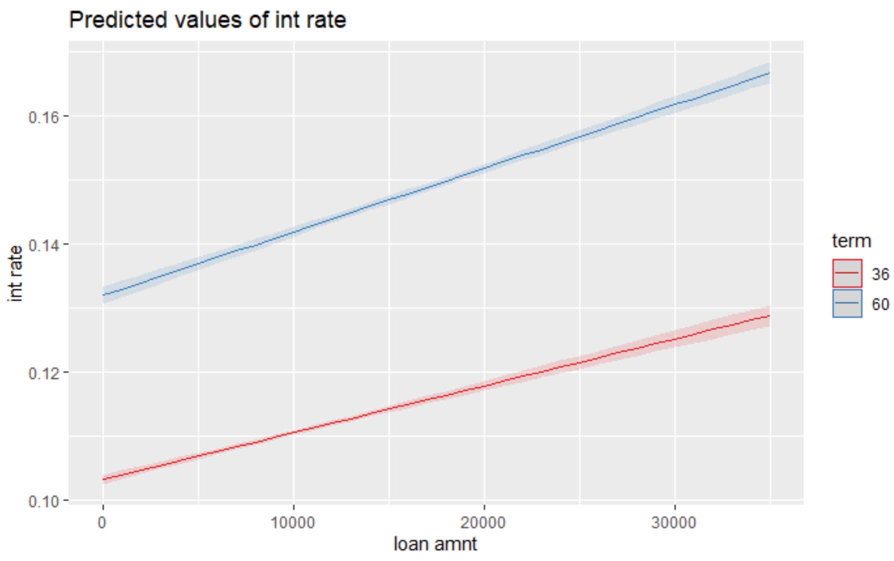


Interaction effects

- What if the loan amount influences differently the interest rate for 60 month loans than 36 month loans. How would you investigate this hypothesis?

Interaction term in R

```
Call:
lm(formula = int_rate ~ loan_amnt + term + dti + annual_inc +
    term:loan_amnt, data = lc_clean)

Residuals:
      Min        1Q    Median        3Q       Max
-0.092635 -0.027663  0.000206  0.024082  0.118431

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      9.806e-02  5.066e-04 193.575  < 2e-16 ***
loan_amnt        7.337e-07  3.201e-08  22.924  < 2e-16 ***
term60           2.879e-02  7.703e-04  37.377  < 2e-16 ***
dti              3.822e-04  2.546e-05  15.012  < 2e-16 ***
annual_inc       2.614e-10  2.869e-09   0.091    0.927
loan_amnt:term60 2.615e-07  4.929e-08   5.306 1.12e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03263 on 37863 degrees of freedom
Multiple R-squared:  0.2336,    Adjusted R-squared:  0.2334
F-statistic:  2308 on 5 and 37863 DF,  p-value: < 2.2e-16
```



- The coefficient of LoanA is 7.337e-07 – this implies than on average an additional $10,000 of loan will increase the interest rate of a 36 month loan by 0.73%
- The coefficient of interaction “Loan x Term60_dummy” is 2.615e-07 -- this implies that on average an additional $10K of loan will increase the interest rate of a 60month loan by 0.73%+0.26%=0.99%

Feature Engineering

- **Use information from the dataset**
  - Candidates: ***Term, income, dti, grade***, number of delinquencies, employment length, etc
    - Some of these are numerical others are factors. How do we use factor variables?
  - Interaction terms: *Perhaps the loan amount affects 36-month loans differently than 60-month loans*. How do you model this?
    - Interactions between two factor features, a factor and a numerical feature, two numerical features
  - Non-linear terms: *Perhaps a small increase in the loan amount doesn't affect interest rate so much but a large increase does.* How would you model this?
    - ***Polynomial terms*** (powers of a feature) or any ***other non-linear transformation*** (better have a good reason for the non-linear transformation)
    - Dummy variable creation → converting a numerical variable into a factor variable (e.g., low, mid, high income, or ***deciles of income***). This is a non-parametric way of modelling non-linear relationships

Model Comparison

- Suppose I run multiple models. How do I choose between them?
  - Smaller standard error → leading to more precise forecasts
  - Higher R^2 /adjusted R^2 → more variation explained
  - All coefficients significant ($p < 0.05$)
  - Simple models / sensible relationships

- Let's see some examples

London Business School

Model Comparison in R

- Model 2:

```
Call:
lm(formula = int_rate ~ loan_amnt + term + dti + annual_inc +
    grade, data = lc_clean)
```

- Model 3:

```
Call:
lm(formula = int_rate ~ loan_amnt + term + dti + annual_inc +
    grade + loan_amnt:grade, data = lc_clean)
```

Polynomial term of order 2

- Model 4:

```
Call:
lm(formula = int_rate ~ poly(loan_amnt, 2) + term + term:grade +
    I(1/(dti + 1)) + annual_inc + grade, data = lc_clean)
```

Deciles of loan amount as factor

- Model 5:

```
lc_clean <- lc_clean %>% mutate(loan_amnt_decile = as.factor(ntile(loan_amnt, 10)))
Call:
lm(formula = int_rate ~ loan_amnt_decile + term + dti + annual_inc +
    grade, data = lc_clean)
```

- ANOVA

```
  Res.Df    RSS Df Sum of Sq        F    Pr(>F)
1  37867 47.485
2  37858  4.219  9    43.266 43157.40 < 2.2e-16 ***
3  37852  4.186  6     0.033    49.60 < 2.2e-16 ***
4  37851  4.106  1     0.080   719.22 < 2.2e-16 ***
5  37850  4.216  1    -0.110
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Feature Engineering & Multicollinearity

- **What if one feature is perfectly correlated with another feature (or a linear combination of other features)?**
  - For example, this would be the case if
    - We estimate a model $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$, and
    - It happens to be the case that $x_1 = \alpha x_2 + \gamma x_3$
  - Not possible to estimate the model as the feature matrix is not full rank
- **What if some features are highly correlated but not perfectly so?**
  - Model is still consistent
    - Predictions unbiased, prediction errors estimated correctly; R-squared can be interpreted as usual
  - Estimated coefficients can become unreliable (large confidence intervals, large p-values)
  - This is not a problem for predictions, only a problem for inference!
  - If you are interested in inference you can calculate variance inflation factors (VIF) to assess the problem

Feature Engineering

- **Use information from the dataset**
  - Candidates: ***Term, income, dti, grade***, number of delinquencies, employment length, etc
    - Some of these are numerical others are factors. How do we use factor variables?
  - Interaction terms: *Perhaps the loan amount affects 36-month loans differently than 60-month loans*. How do you model this?
    - Interactions between two factor features, a factor and a numerical feature, two numerical features
  - Non-linear terms: *Perhaps a small increase in the loan amount doesn't affect interest rate so much but a large increase does.* How would you model this?
    - ***Polynomial terms*** (powers of a feature) or any ***other non-linear transformation*** (better have a good reason for the non-linear transformation)
    - Dummy variable creation → converting a numerical variable into a factor variable (e.g., low, mid, high income, or ***deciles of income***). This is a non-parametric way of modelling non-linear relationships
- **Look for data outside your model**
- **Feature engineering is more of an art than science! Know your context (or work with people who do)!**

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**

- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation, bootstrap methods
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression

- **Choose a model and use it responsibly!**

Out of sample testing

- Assessing how good the predictions of a model are in-simple is helpful but could be misleading → problem of overfitting
- Gold standard – out-of-sample goodness of fit
- Hold out method: Randomly partition the data in two sets
  - Training set (~75% of the data): Data in this dataset is used to fit the model
  - Testing set (~25% of the data): Data in this dataset is used to assess how good is the model
    - Check RMSE and R^2 of the validation set and compare them to the training set. If difference is small then overfitting is not a problem. In any case, report the out of sample RMSE and R^2
  - Typically, the fit out-of-sample is worse than in-sample and ***more representative*** of the model's true predictive value -- it does not suffer from overfitting

Out-of-Sample testing in R

- For model 2

```r
set.seed(4444)
train_test_split <- initial_split(lc_clean, prop = 0.75)
training <- training(train_test_split)
testing <- testing(train_test_split)

#Fit model2 to the training set
model2<-lm(int_rate ~ loan_amnt + term+ dti + annual_inc + grade , data = training)

#Calculate the in sample RMSE of the model
rmse_training<-sqrt(mean(residuals(model2)^2))

#USe the model to make predictions out of sample in the testing set
pred<-predict(model2,testing)

# Calculate the out of sample RMSE of the model
rmse_testing<- RMSE(pred,testing$int_rate)
```

```
[1] "RMSE in sample: 0.0105306346945014"
[1] "RMSE out of sample: 0.0106305940013813"
[1] "Increase in error: 0.9492%."
```

k-fold Cross-validation

- Splitting the data in training and validation means that
  - we reduce the data used for training which may be a problem if we have a relatively small dataset
  - we don't get to use every data point for training and for testing (only one of the two)

- K-fold cross-validation overcomes this problem:
  - Randomly divide data into K equal-size groups (referred to as folds)
  - Use the first fold as validation and the other K-1 for training and compute RMSE and R^2
  - Repeat this K times; each time a different fold is treated as a validation set
  - Use the average RMSE and average R^2 to assess goodness of fit
  - Estimate the model again using all of the data and report these coefficients but report RMSE and R^2 estimated out of sample

- Typically set K = 5 or 10
  - The more folds the more accurate the out of sample estimation but the longer it takes to run

Data

Validation	Train	Train	Train	Train
1	2	3	4	5

London Business School

k-fold cross validation in R

```
#the method "cv" stands for cross validation. We re going to create 10 folds.

control <- trainControl (
    method="cv",
    number=10,
    verboseIter=TRUE) #by setting this to true the model will report its progress after each estimation

#we are going to train the model and report the results using k-fold cross validation
plsFit<-train(
    int_rate ~ loan_amnt + term+ dti + annual_inc + grade ,
    lc_clean,
   method = "lm",
    trControl = control
   )


summary(plsFit)
```

```
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118827 -0.007035 -0.000342  0.006828  0.035081
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  2.072e-01  6.347e-04  326.404  < 2e-16 ***
## loan_amnt    1.475e-07  8.284e-09   17.809  < 2e-16 ***
## term60       3.608e-03  1.419e-04   25.431  < 2e-16 ***
## dti          4.328e-05  8.269e-06    5.234 1.66e-07 ***
## annual_inc  -9.734e-10  9.283e-10   -1.049    0.294
## gradeA      -1.355e-01  6.208e-04 -218.245  < 2e-16 ***
## gradeB      -9.994e-02  6.142e-04 -162.713  < 2e-16 ***
## gradeC      -7.533e-02  6.172e-04 -122.039  < 2e-16 ***
## gradeD      -5.376e-02  6.213e-04  -86.534  < 2e-16 ***
## gradeE      -3.550e-02  6.328e-04  -56.091  < 2e-16 ***
## gradeF      -1.594e-02  6.846e-04  -23.288  < 2e-16 ***
## gradeG             NA         NA       NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01056 on 37858 degrees of freedom
## Multiple R-squared:  0.9198, Adjusted R-squared:  0.9197
## F-statistic: 4.34e+04 on 10 and 37858 DF,  p-value: < 2.2e-16
```

The coefficients reported are based on the whole dataset.

RMSE and R squared are based on the average of the 10 out-of-of sample validations

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**
- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation, bootstrap methods
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression
- **Choose a model and use it responsibly!**

Sample size determination

- Remember, any predictions we make come with error

$$Int = \beta_0 + \beta_1 \times loanA + \epsilon$$

- Any prediction will be subject to two sources of error
  - Estimation error → because we estimate the coefficients β_0, β_1 with error
  - Stochastic (random) error → because of the error term ϵ
- Having more data (i.e., more rows) helps us reduce estimation error → more accurate estimates of the coefficients → more accurate forecasts
- Having more datapoints does not reduce the second source of error
  - What can we do to reduce the second source of error?
- How much data is enough?

Sample size determination
Learning curves

- We can check if collecting more rows of data would improve the out-of-sample performance of a model
  - Fix a testing data set (say 25% of the data randomly drown)
  - From the remaining 75% of the data draw training sets of different size. Start small and progressively increase the sample size.
  - Estimate the model on this training set and evaluate the performance of the model on the fixed testing dataset. Record how the performance changes as we increase the sample size of the training set.
  - If the performance stabilizes then estimation error is not a big deal and we have enough data
  - If we use all the training data and we are still in the “steep” part of the curve the we do not have enough data → Need to collect more and/or run a LASSO regression instead (more on this later)

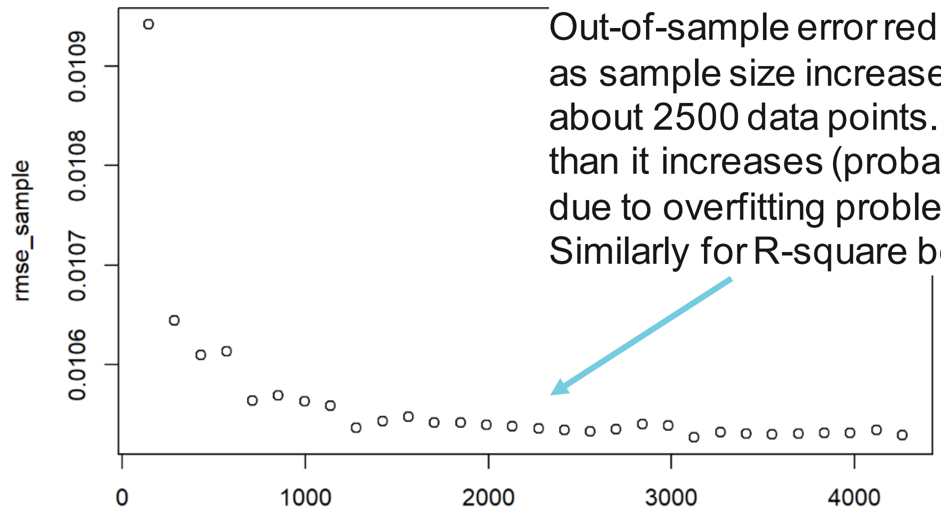
London Business School

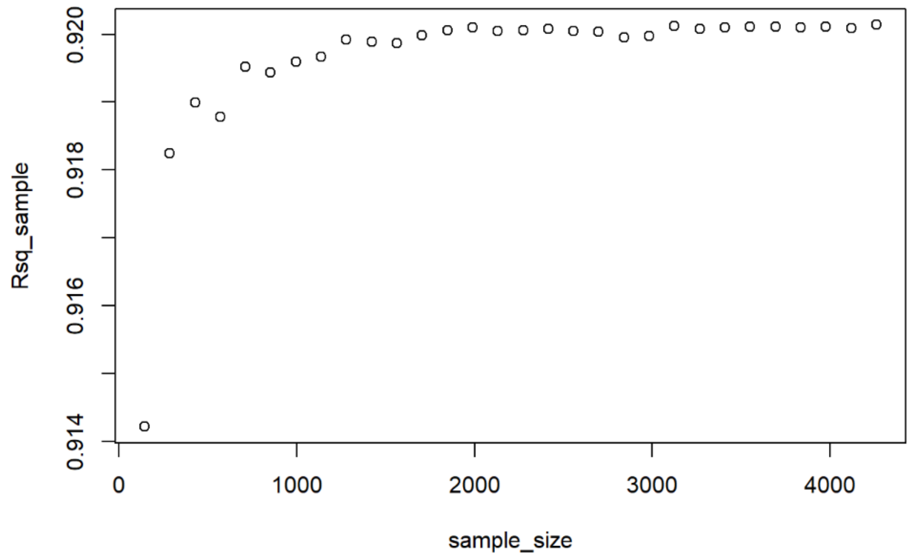
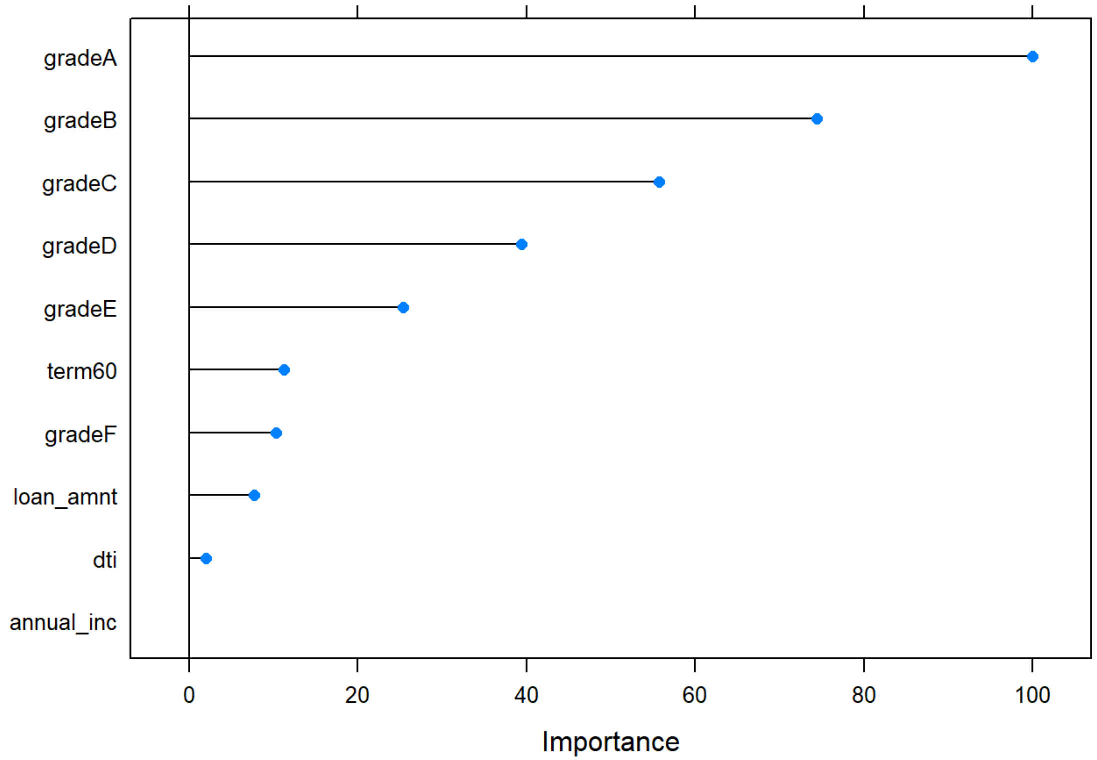
Learning Curves in R

```
```{r, learning curves}
#select a testing dataset (25% of all data)
set.seed(102)

train_test_split <- initial_split(lc_clean, prop = 0.75)
testing <- testing(train_test_split)
remaining <- training(train_test_split)

#We are now going to run 30 models starting from a tiny training set and progressively increasing the size of the training set. The testing set remains the same in all iterations.

#define some variables
rmse_sample <- 0
sample_size<-0
Rsq_sample<-0

#start a for loop
for(i in 1:30) {
#from the remaining dataset select a smaller subset to traing the data
set.seed(100)
train_test_split <- initial_split(remaining, prop = 0.005+(i-1)/200)
training <- training(train_test_split)

sample_size[i]=nrow(training)

#train the model on the small dataset
model<-lm(int_rate ~ loan_amnt + term+ dti + annual_inc + grade , training)
#test the performance of the model on the large testing dataset
pred1<-predict(model,testing)
rmse_sample[i]<-RMSE(pred1,testing$int_rate)
Rsq_sample[i]<-R2(pred1,testing$int_rate)
}
plot(sample_size,rmse_sample)
plot(sample_size,Rsq_sample)
```

For Model 2

Out-of-sample error reduces as sample size increases to about 2500 data points. After than it increases (probably due to overfitting problems). Similarly for R-square below.




```

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**

- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation, bootstrap methods
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression

- **Choose a model and use it responsibly!**

Not all features are equally important

- For model 2, graphical representation of the relative importance of different features in explaining interest rates
- Clearly, the annual income and dti have tiny importance compared to Grade A
- Can we use a plot such as this to select a subset of important features?
  - Not so easy because off correlations! We need to check all subsets to be sure which combination produces the best results.



Feature selection

- Let say we have 10 candidate features to select from. How many possible subsets are there? What if it was 100 candidate features?
- Automated feature selection algorithms such as ***stepwise regression*** try to identify a sensible combination of features that performs well out of sample without having to check all possible combinations
  - Backward step: Start with the full model and remove variables one at a time based on explanatory power
  - Forward step: Start with a null model and add variables one at a time based on their correlation with the dependent variable (or some other measure)
  - Mixed step: A combination of the above
- ***Stepwise regression*** selects the best model in terms of RMSE or R^2. It can (and should be) combined with some out-of-sample validation method (e.g., k-fold cross validation) to avoid overfitting the data
- These are easy to implement but in general they do not guarantee to find the best model. They can be slow for large models and are a bit of a black box… Use them cautiously!

Stepwise regression in R

```
```{r, automated variable selection}

#set the out-of-sample validation method
control <- trainControl (
    method="CV",
    number=5,
    verboseIter=FALSE)

#Find the best model with 10 to 16 variables with backward induction
BackFit <- train(int_rate ~ loan_amnt + term+ dti + annual_inc + grade
+grade:loan_amnt,
    lc_clean,
    method = "leapBackward", #can chance method to "leapSeq", "leapForward"
    tuneGrid = data.frame(nvmax = 10:16), #Will find the best model with
10:16 variables.
    trControl = control
)

#show the results of all models
BackFit$results
#simmarize the model of best fit and its coefficients
summary(BackFit$finalModel) #depending on the number of models estimated,
the output of this command could be long
coef(BackFit$finalModel,BackFit$bestTune$nvmax)

```
```

The only difference in the train command is the “method.”

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**
- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation, bootstrap methods
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression
- **Choose a model and use it responsibly!**

Regularization

- All methods of choosing between models we have discussed (e.g., manual comparisons, stepwise regression) relied on estimating a bunch of models using the Ordinary Least Squares (OLS) algorithm and then comparing their performance based on some (out-of-sample) performance measure (e.g., RMSE)
- As such they are passive – if the algorithm overfits the data we find out but we cannot do anything to correct it!
- An alternative method that tries to actively avoid overfitting is regularization
  - Main idea is to modify the OLS algorithm so that the estimated model becomes less sensitive to the training set
  - By changing the least squares model the estimated coefficients and the predictions become biased
  - But the bias is something worth tolerating as the model's predictions become less variable (lower error)
- There are two popular regularization methods, Ridge regression and LASSO
  - They work much better than ordinary linear regression especially for small datasets ("steep part" of the learning curve)
  - LASSO regression is considered better for models with a lot of potentially irrelevant variables as it forces the estimated coefficient of irrelevant variables to be zero more so than ridge regression
  - LASSO stands for Least Absolute Shrinkage and Selection Operator. We will focus on this.
  - For more information I recommend these you tube clips from StatQuest Part 1 & Part 2

LASSO Regression

- Set up a linear model as usual, e.g., $Int = \beta_0 + \beta_1 \times loanA + \epsilon$
- OLS Regression finds the coefficients that minimize the sum of squared errors
- LASSO regression finds the coefficients that minimize the following objective:

Sum of Squared Errors + λ [sum of absolute value of estimated coefficients]

- The parameter $\lambda \geq 0$ (pronounced ***lambda***) is called a ***hyper-parameter*** and it is user specified
  - If λ = 0 the model reduces to the OLS algorithm
  - If λ > 0 the model penalizes the objective for any coefficient that is different to zero
  - Therefore, for a coefficient to be different from zero by 1 unit it needs to reduce the sum of square errors by at least λ units
  - The larger the λ the more coefficients will move towards zero or become zero → this is the notion of shrinkage in LASSO
- We typically estimate the model using several values for λ and choose the best one using out of sample predictions based on k-fold cross validation.
  - This is called hyper-parameter optimization.

LASSO Regression

- Since coefficients of different variables are measured in different units, it is important to ***standardize*** any continuous variable (subtract the mean and divide by standard deviation). Otherwise, results will be misleading!
- Unlike linear regression, LASSO regression
  - Allows us to estimate a model even if we have more parameters to estimate than data points. Useful in a world of big data (e.g., detecting combinations of genes that may be associated with specific phenotypes / disease)
  - Allows us to estimate the model even if there are multicollinearity problems (even perfect collinearity)
  - Generates biased estimates for variable coefficients. So do not use if your goal is inference instead of prediction
- Most modern data-science applications working on big data would be using a LASSO model for prediction!

LASSO in R

```
```{r, LASSO compared to OLS, warning=FALSE, message=FALSE}

#split the data in testing and training. The training test is really small.
set.seed(1234)
train_test_split <- initial_split(lc_clean, prop = 0.01)
training <- training(train_test_split)
testing <- testing(train_test_split)

#we will look for the optimal lambda in this sequence
lambda_seq <- seq(0, 0.01, length = 1000)
#lasso regression with using k-fold cross validation to select the best lambda
lasso <- train(
 int_rate ~ poly(loan_amnt,3) + term+ dti + annual_inc + grade +grade:poly(loan_amnt,3):term +poly(loan_amnt,3):term +grade:term,
 data = training,
 method = "glmnet",
  preProc = c("center", "scale"), #This option standardizes the data before running the LASSO regression
  trControl = control,
  tuneGrid = expand.grid(alpha = 1, lambda = lambda_seq) #alpha=1 specifies to run a LASSO regression. If alpha=0 the model would run ridge regression.
  )
# Model coefficients
coef(lasso$finalModel, lasso$bestTune$lambda)
#Best lambda
lasso$bestTune$lambda
# Count of how many coefficients are greater than zero and how many are equal to zero
sum(coef(lasso$finalModel, lasso$bestTune$lambda)!=0)
sum(coef(lasso$finalModel, lasso$bestTune$lambda)==0)

```

```

# Make predictions
predictions <- predict(lasso,testing)

# Model prediction performance
LASSO_results<-data.frame(  RMSE = RMSE(predictions, testing$int_rate),
                            Rsquare = R2(predictions, testing$int_rate)
)
LASSO_results
#compare the out of sample performance of the lasso regression to a linear model's predictions on the same training/testing datasets
model_lm<-lm(int_rate ~ poly(loan_amnt,3) + term+ dti + annual_inc + grade +grade:poly(loan_amnt,3):term +poly(loan_amnt,3):term +grade:term, training)
predictions <- predict(model_lm,testing)

# Model prediction performance
OLS_results<-data.frame(
  RMSE = RMSE(predictions, testing$int_rate),
  Rsquare = R2(predictions, testing$int_rate)
)

OLS_results
```
```

- We select a really small training set with only 380 loans (1% of the dataset)
- The model I try to fit has 66 coefficients (multiple interaction terms)
  - OLS model out of sample estimation results

RMSE	Rsquare
0.0446	0.352

  - LASSO out of sample results using 5-fold cross validation to determine the best $\lambda=0.0007$. Only 20 coefficients are not zero.

RMSE	Rsquare
0.0111	0.916

  - Remember RMSE was 0.0105 when we used 75% of the data to train the model! So LASSO with 1% of data performs almost as well as linear regression with 75% of the data (error is only 5.9% higher from 0.0111 to 0.0105)

The Art & Science of Using Linear Regression for Prediction

- **ICE the data: Inspect, Clean, & Explore**
- **Fit several reasonable models (iterative process!)**
  - Ordinary Least Squares (OLS) estimation
  - Feature engineering: Non-linear terms, interactions, categorical variables, look beyond the data
  - In-sample vs Out-of-sample testing: Hold out method, k-fold cross-validation, bootstrap methods
  - Sample size determination
  - Automated feature selection and stepwise regression
  - Regularization and LASSO regression
- **Choose a model and use it responsibly!**

Forecasting

- Forecast the interest rate of a new loan

loan_amnt (K$)	term (months)	dti	delinq_2yrs	annual_inc
25	60	25.74	0	78000

  - Model 1: $Int = \beta_0 + \beta_1 \times loanA + \epsilon$
    - Interest rate prediction: 14.17%
    - 95% Confidence interval (+/- 2 x standard error) = [7.27%-21.08%]
  - Model 2: $Int = b_1 + b_2 loanA + other\ explanatory\ variables + \epsilon$
    - Interest rete prediction = 16.00%
    - 95% Confidence Interval (+/- 2 x standard error) = [9.74%-22.25%]
  - Best model I could come up with (using more features)
    - Interest rate prediction = 11.66%
    - 95% Confidence Interval (+/- 2 x standard error) = [10.10%-13.22%]
  - Actual value 11.99%
- Always report C.I. using out-of-sample validation statistics
- Continuously monitor and review your models!

London Business School

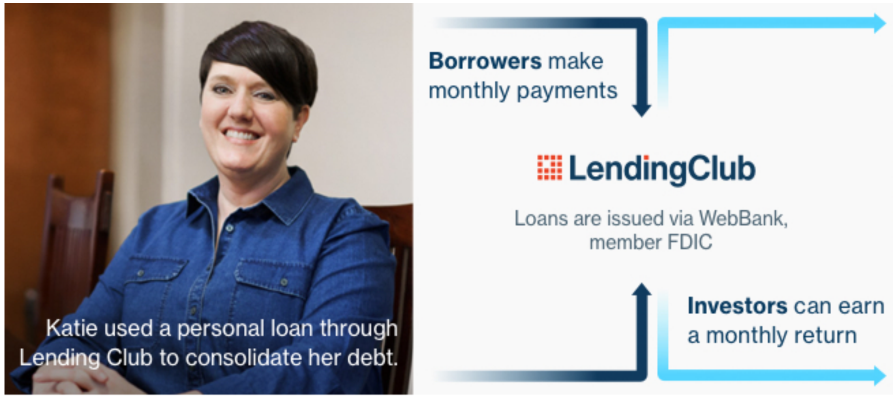
Next Lecture

Course contents (first part of the course – Kamalini)

- Session 1: The Art & Science of Regression Models For Prediction
- Session 2: More on Using Linear Regression For Prediction
- **Session 3: Workshop I – Engineer an algorithm that sets interest rates for new Lending Club loans**
  - Group assignment 1, due 6 days after the workshop
- Session 4: Classification using Logistic Regression
- Session 5: Workshop – Invest in a portfolio of Lending Club loans
  - Individual project 1, due 13 days after the end of the workshop

Course contents (second part of the course – Kanishka)

- See canvas syllabus



Innovation transforms lending

Lending Club is the world's largest marketplace connecting borrowers and investors, whe consumers and small business owners lower the cost of their credit and enjoy a better experience than traditional bank lending, and investors earn attractive risk-adjusted retur

Here's how it works:

- **Customers interested in a loan complete a simple application at LendingClub.com**
- **We leverage online data and technology to quickly assess risk, determine a credit ratii appropriate interest rates. Qualified applicants receive offers in just minutes and can e with no impact to their credit score**
- **Investors ranging from individuals to institutions select loans in which to invest and e returns**

The entire process is online, using technology to lower the cost of credit and pass the savings back in borrowers and solid returns for investors.