# Practicing Learning from Data

## *Preliminaries*

### Data Set

For the course assignment a version of the **«Lending Club Loan Data»** is used. It contains real (anonymized) data describing personal loans issued through the Lending Club website (https://www.lendingclub.com/). The Lending Club operates an online credit marketplace. The data set contains loans from several years, including the current loan status (Current, Late, Fully Paid, etc.) and latest payment information. The original one can be found at *kaggle* (https://www.kaggle.com/search?q=lending+club). For the assignment we use modified versions of the data set. Please find them on Moodle (one for each assignment).

### Assignments

In this course 2 assignments need to be done:

- **Assignment 1:** Provide a regression model for predicting the interest rate (`int_rate`). Notice: In practice, such a model could be used, e.g., to provide automatic recommendations to Lending Club officials who are handling new loan requests.
- **Assignment 2:** Provide a classification model for the default status of customers using neuronal networks.

### Groups

Both assignments are done in **groups of 4 students**. During both assignments the groups remain the same. We recommend that at least one member of the group has a background in programming.

### Deliverables

At the end of the course your group hands in **2 deliverables**:

1. a PDF document containing a description and documentation of both assignments' solutions
2. a ZIP fil containing the code.

The PDF document contains the following:

- A description of all the operations you perform;
- An explanation / justification of *why* you perform them;
- An interpretation of your *all* results (including intermediate results).

The R-script containing the code:

- Is executable.
- Contains enough comments to guide a reader through your code.
- Please set the seed in R to one ("set.seed (1)"), so that we can reproduce your results.

## Competition

All groups participate in a competition: The final regression and classification models of each group will be applied to new data (a separate portion of the original data set that we give you later). The group who achieves the lowest error on the new data will be honored.

## Useful resources:

- https://nycdatascience.com/blog/r/p2p-loan-data-analysis-using-lending-club-data/
- http://blog.yhat.com/posts/machine-learning-for-predicting-bad-loans.html
- https://medium.com/@jiaminhan/peer-to-peer-loan-default-prediction-using-lending-club-data-3f75886cb1e
- https://www.datasciencecentral.com/profiles/blogs/analysis-of-lending-club-s-data

# Practicing Learning from Data

*Assignment 1 - Regression*

## Data Exploration & Preparation

- Manual feature engineering
  - Remove observations with too many missing entries.
  - Inspect the output variable `int_rate` (interest rate). Transform it and delete NAs if needed.
  - Inspect/explore the remaining attributes. Particularly, look for indications that help you decide how to proceed with feature engineering. Here are some possible steps you may want to take:
    - Remove sparse attributes. (Rule of thumb: remove attributes with 70% NAs or more.)
    - Type conversions: If possible, convert, e.g., the character data type, to data types that are more useful for regression tasks (such as factor, ordered factor or numeric).
    - Downsampling: Consider downsampling some categorical variables to reduce the number of levels. (E.g. reduce granularity from `month-year` to `year`). Alternatively, you may consider removing an attribute with too many levels. (Why do we want to get rid of too many levels?)
    - Imputation: Select an imputation method to replace NAs in the remaining attributes with appropriate approximations. You may check out packages for imputation, e.g. "mice" and "Hmisc", or just do it manually.
    - Select and apply a method of subset selection discussed in class to further reduce the number of predictors.
- Resampling
  - Apply the Validation Set Approach: reserve a meaningful amount of data for your testing phase.
- Check for linear patterns and collinearity
  - Compute the correlation matrix for the remaining predictors and the output variable. If useful, use graphical representation (e.g., use the `corrplot` package).
  - What attributes might be good linear predictors?
  - What input variables show high collinearity? What to do about it?

## Main task

- Select a method to perform regression. Justify your selection.
- Use cross-validation to find the best hyperparameters for your method.

- Compute training error, test error and CV error.

## Competition

For the regression model the group who achieves the lowest mean squared error on the new data will be honored.

## Some hints

- In `lm()`, if you have either too many variables (n~p) or collinearity in your attributes, you may get a warning “prediction from a rank-deficient fit may be misleading”. Check this, e.g., with `length(fit$coefficients) > fit$rank`
- Depending on your memory, you may run into performance problems with some computing intensive functions, getting an error such as “vector memory exhausted (limit reached?)”. In this case, set the environmental variable `R_MAX_VSIZE` to a high value, such as 700Gb. Use `Sys.getenv()` and `Sys.setenv()`. If it still doesn’t work, decrease the number of attributes further (in a sensible way).
- The best thing to learn from best subsets regression is not what the "best" model is, but rather that there rarely is a "best" model and that several models may have similarly good properties.