Homework 2

Introduction

In general, the data that you will receive or pull from a database will contain some issues that need to be solved before you can apply empirical models. It is a common phrase that 80% of time and effort goes into preparation of the data, e.g. cleaning up any inconsistencies, reformatting variables and values, and transforming the data.

In this homework exercise, we will apply the data handling from the exercise to clean up some data. The data that you receive from the bank's database contains information on the customers and a response variable: a binary indicator that depicts whether a customer missed three months of payments. The latter is called a 'default event' in the credit scoring literature.

Loading data

1. Read the (very short) description of the data set provided in the excel file **Loan_Data-Data_Dictionary**.
2. Load the data from the .csv-file into a data frame called *loans*.
3. Check the structure of both data frames using function **str()**. Check if the variable types match the information provided in the data dictionary and are plausible.
4. *Summarize* the distribution of all columns in the data set using an appropriate R function. In other words, find out the minimum, mean and maximum value of each variable. Remember that you can easily find appropriate functions and code snippets via the R help or Google. Every programmer constantly does this, so you should, too.

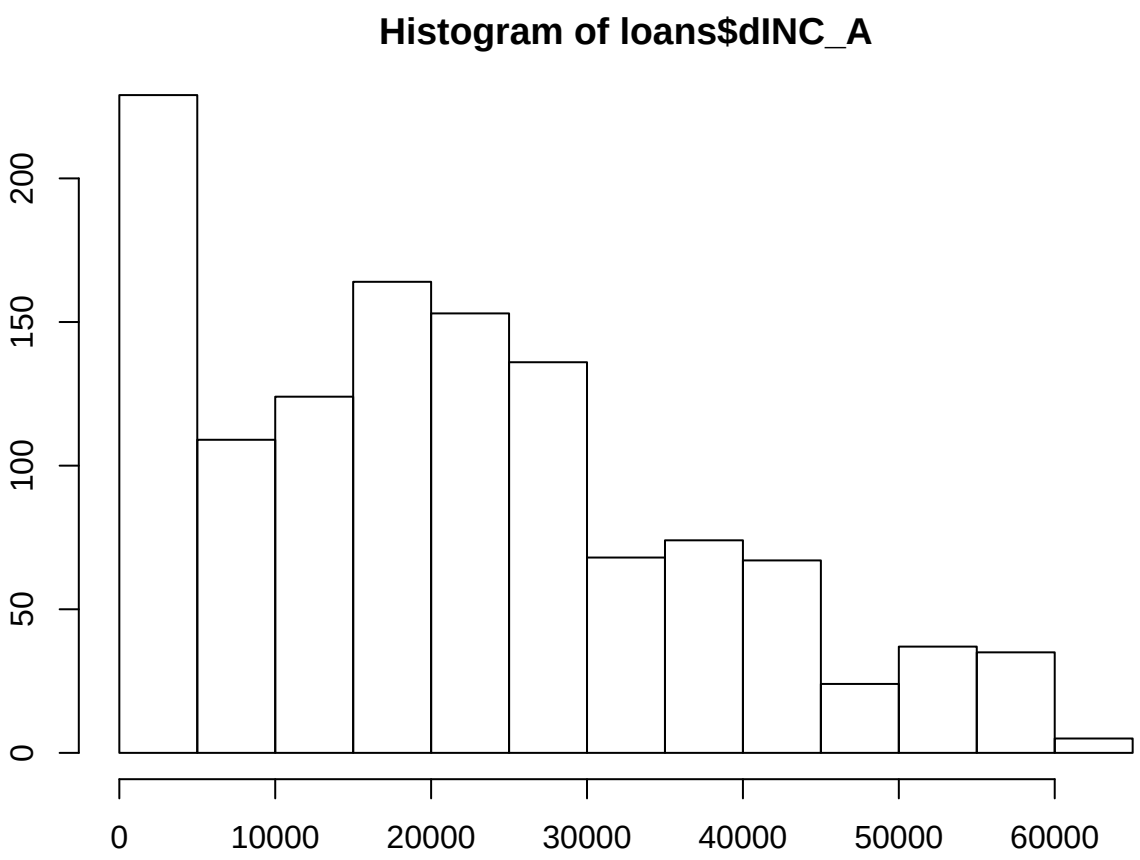
Packages and R Studio

A variety of useful R *packages* are available at the Comprehensive R Archive Network (CRAN). A package is a collection of functions centered around some task, which other people have created and uploaded. Package management is inbuilt into R, so it's easy to extend the base functionality for the specific applications that we need. This is done in two steps: First, you must download the package once from the CRAN archive to save its functions on your computer. Second, before you use the package (after every restart of R), you load the package content into your environment. This procedure makes sure that there are not too many conflicts between function names, for example.

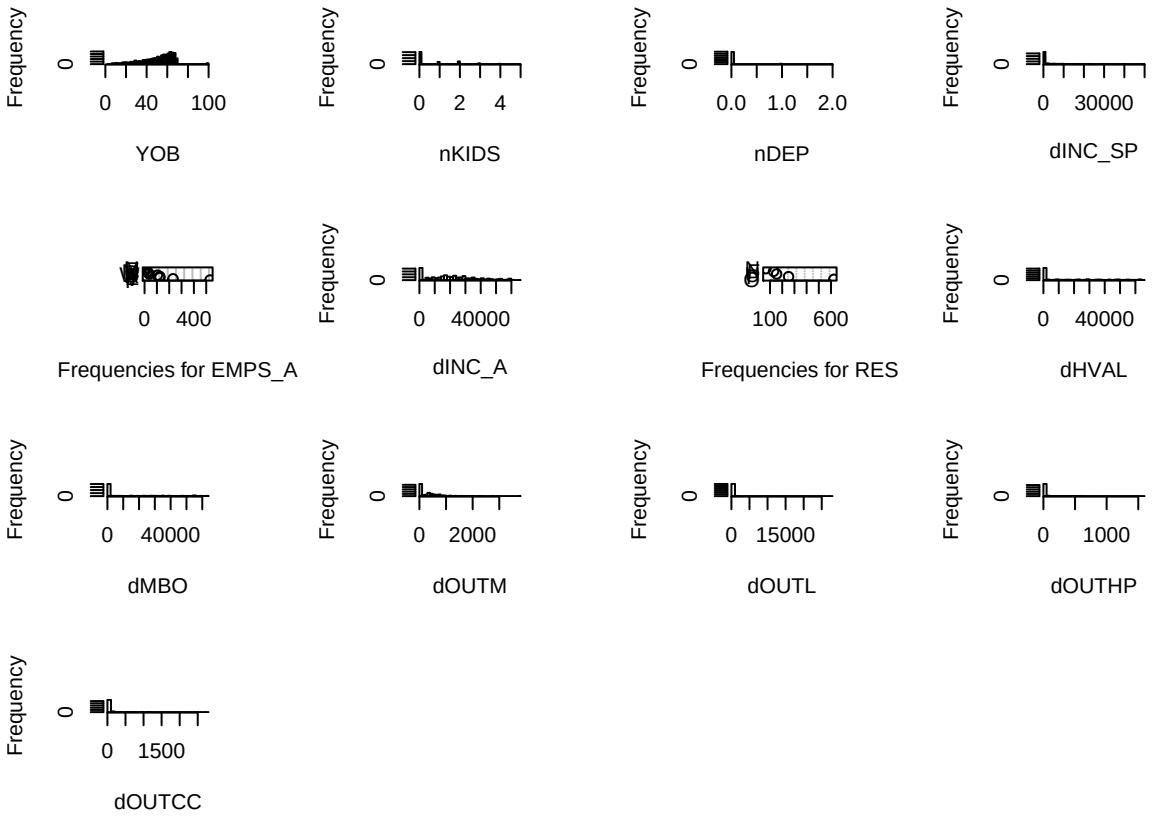
1. Navigate to http://cran.r-project.org/ or just google a package called *mlr*. Have a brief look into its documentation to learn about its functionality
2. Use R Studio to download and install the package. Note: Press 'n' if queried if you'd like to install from source
3. Load the package using function **library()**
4. Use the R help to query the functioning of the method **train**

Plotting a histogram

1. Create a histogram plot that depicts the distribution of the applicants' incomes dINC_A. There is a function to create histograms in base R (i.e. usable without installing additional packages).



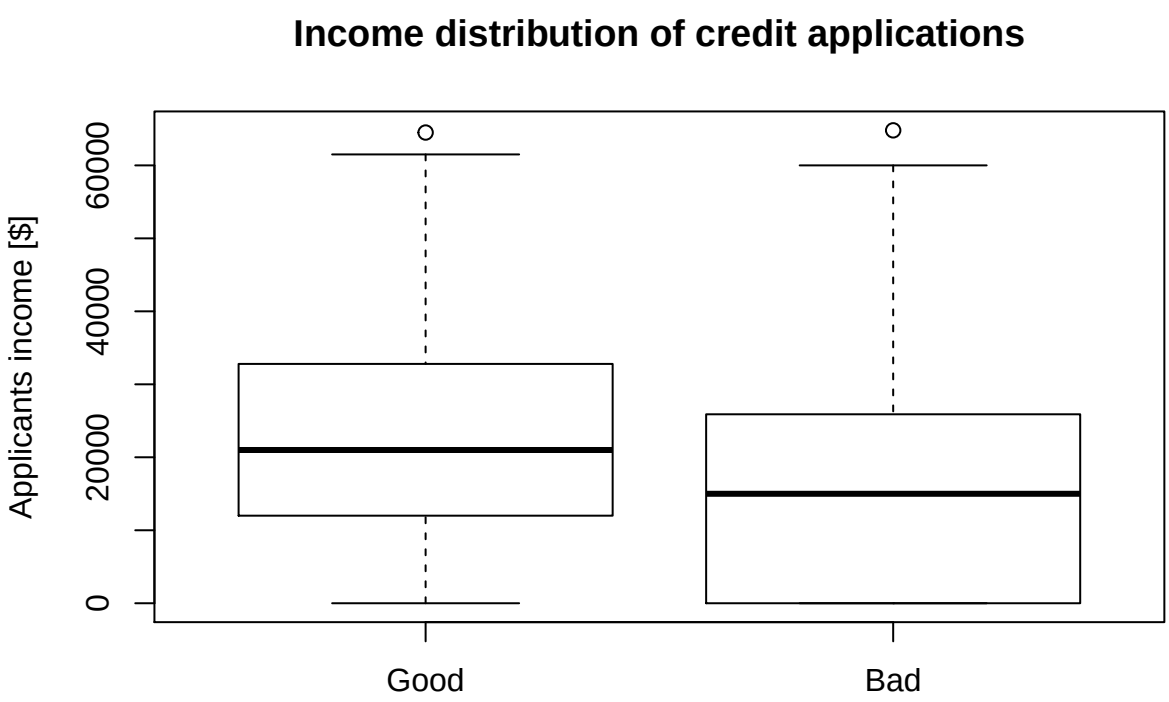
2. The **hist()** function gives us a way to plot the distribution of one variable in a data.frame. There is an alternative function that allows you to create a histogram plot for all variables in a data.frame. Which function is that? Hint: You will need to install package *Hmisc*.
3. Use the function identified in task 2 to depict the distribution of all variables in a matrix of histograms.



Statistical analysis

Column BAD gives the risk status (good or bad) for each applicant. Applicants income is available in the column *dINC_A*. To separate the good and the bad risks in the data.frame, you can use logical indexing (with variable[index] and matrix[row index, column index]).

1. Create two variables, inc.good and inc.bad, which contain the incomes of good and bad credit risks, respectively. You will need to use logical indexing to select the correct rows and data.frame indexing to select the right column.
2. Depict the distribution of the income of customers with a good and bad risk, respectively, by means of a boxplot. Try Google or search the R help with **??boxplot**. On average, which of the two groups earns more?



3. Calculate the difference between the average/**mean** income of good and bad credit applicants.
4. Identify an appropriate statistical test to verify whether the observed income difference is statistically significant. Perform the test and display its results. Hint: A Google search similar to "R statistical test difference in means" will help.
5. Assign the test result to a variable. Use the print() function to output a message that tells the user whether the observed income difference is significant. You can do this with an *if()* condition by checking the list entry *p.value* contained in the test result.

Well done, you have completed your first statistical analysis with R!

Well done! You are almost a data analyst!