STAT1003

Introduction to Data Science

Workshop 4

Contents

Wrangling and Exploring Data about Movies

The publicly-available portion of the Internet Movie Database is a popular website that contains virtually all the information you need to know about a movie: year of release, cast, crew, classification, and so on. Another website, The Numbers, contains financial information for a relatively small subset of these films. Data from both these sites is available at the data-aggregation site StatCrunch, and in this workshop, you will be putting two datasets together and carrying out some data-cleaning and exploratory data analysis and visualization.

Data that is available on, or scraped from, the web is encoded in many, many different formats and then has to be imported into R. One of the more convenient ones is a CSV - comma separated value - file. As Baumer *et al.* (2017) write, "[I]t is a non-proprietary comma separated text format that is widely used for data exchange between different software packages. CSV files are easy to understand, but are not compressed, and therefore can take up more space on disk than other formats." The data files you'll be analyzing are CSV files (`IMDB.csv` and `MovieFinances.csv`). Download them from Blackboard into your `I:\STAT1003` folder.

The steps you'll be going through below are simply to 'get to know' the data as a prelude to analyzing it, or to building predictive models. The first step is to get the data into R and then to carry out some simple 'data wrangling' operations; the second is to prepare the data for exploratory analysis (EDA); and the third is to carry out the EDA. Note that there might be some iteration between the second and third steps! There is no one way of doing this: individuals will have their own preferences, and the steps here are only a guide.

1. Go to the StatCrunch website to find out a little bit more about the content of these files. In the list that you will see, `MovieFinances.csv` comes from the entry entitled "Movie Budgets and Box Office Earnings (Updated Fall 2016)", and `IMDB.csv` from the entry entitled "IMDB Movie Database".
2. What is the size of these two files?
3. Most operating systems will have utilities that will allow you to determine the number of lines in a file (and other information as well) without having to open the file in a program. Google how to do this at the Windows command prompt.
4. Before importing a file into R, it's useful to know something about its structure. Again, use Google to find out how to view the contents of a text file at the command line.
5. Using the command `read.csv`, read in the data from these two files to create two data frames. For the purposes of this workshop, call them `IMDB` and `MovieFinances`. The values of some of the arguments will depend on the structure of the file that you saw in 3. above.

6. After importing the `.csv` files into *R*, have a look at the variable names in both data frames, and then decide whether you need to modify some of them so that they are more compact or more meaningful.

7. Are there any variables that both datasets have in common? Which ones? Are there any superfluous columns/variables? If there are, remove them.

8. Use `head` to look at the first few rows of each data frame, or view them in the *RStudio* data viewer. Do you notice anything that might be unusual?

9. Examine the structure of the variables in each data frame. What do you notice about the type of the variables `title` and `mpaa` in IMDB and `Movie` and `Month` in `MovieFinances`? What should we do about them?

10. So it looks like we're going to do need to do some manipulation, but perhaps it's better to merge the data sets together because they'll only have some of the same common elements. Have a look at the help file for the function `merge`, and then merge the two datasets together. Call the result `AllData`. What variables should we merge on?

11. How large is the merged dataset?

12. Try out the function `summary` using the data frame as the argument. What kind of information does it produce? What unusual aspects do you notice?

13. Unfortunately, `summary` doesn't really tell us about missing values, but the function `describe` in the package `Hmisc` does. Install the package from the `Tools` menu, and then load the library `Hmisc` to be able to use `describe`. Scan the output of `describe` and try to understand what it's telling you.

14. The function `describe` also has a nice feature: if you save the results of `describe` into an object, and then `plot` that object, you'll get a couple of plots that might be useful. What do those plots tell you about the distributions of the variables?

15. Note that there are two columns with budget information. what are the characteristics of those columns? Do they give the same information? What plot could you construct to determine whether they do?