Big Data and Machine Learning with Python – Exercises

Michael E. Rose, PhD
Course at LMU, March 2019

All question can and should be answered in teams of up to two people (that is, you answer as a team).

Create one GitHub repository for your team with each team member being a collaborator. For each question below, create one script that contains both code and answers (as comments at the end of the script). All scripts need to adhere to PEP8 and must be readable to someone that knows Python.

Save the script in the main folder, properly named in correspondence to the name of the exercise. Apart from the script, there should be one folder named "output" to store output such as figures and tables.

1 Exercises for Pandas, APIs and Plotting

1. Coding workflow
   - Read `http://web.stanford.edu/~gentzkow/research/CodeAndData.pdf`"Code and Data" by M. Gentzkow and J. Shapiro, chapters 2, 3 and 4.
   - What brought Gentzkow and Shapiro to the conclusion, that version control is a necessity?

2. Occupations
   - Import the pipe-separated dataset from `https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user` into a DataFrame. The data is on occupations and demographic information.
   - Set "user_id" as index and name the index "User".
   - Print the last 10 entries and the first 25 entries.
   - What is the type of each column?
   - How many different occupations are there in the dataset? What is the most frequent occupation?
   - What is the age with the least occurrence?
   - Create a histogram for occupations, sorted alphabetically.
   - Save the figure as `./out/occupations.pdf`

3. Countries' alcoholic consumption
   - Load the data from `./data/drink.csv` into a DataFrame. The data is on country's alcoholic consumption.
   - Which continent drinks most beer and wine on average?
   - Create a Boxenplot of "beer_servings" by continent.
   - Reshape the data and create a 3x1 figure for "beer_servings", "wine_servings" and "spirit_servings" by continent (i.e. three boxenplots that share their y-axis and a their color coding).
   - Save the figure as `./out/alcohol.pdf`

4. Tips
   - Load seaborn's tips dataset using `seaborn.load_dataset("iris")`.
   - Plot "tips" on "total_bill" with markers, with line styles and color by "day", and facets by "sex".
   - Label the axis so that the unit becomes apparent.
   - Add a title to the legend and use the full day of the week-name (i.e. "Thursday" instead of "Thu").
   - Save the figure as `./out/tips.pdf`

5. Euro 2012 I
   - Read the data from `./data/Euro_2012.csv` into a DataFrame with column "Teams" as index. The data is on the UEFA Champtionship 2012 (Euro 2012).

- How many teams played in the Euro 2012?
- Which team has the highest shooting accuracy?
- Plot shooting accuracy versus passing accuracy.
- Which team has the second-most shots on target?
- Eliminate Italy from the dataset. Which team has the second-most shots on target now?
- How many penalty goals did England score?
- Present only the Shooting Accuracy from England, Italy and Russia.
- Create a new DataFrame called discipline using the columns "Yellow Cards" and "Red Cards" (and the index).
- Sort discipline primarily by red cars and secondarily by yellow cards.
- Output the data as tab-separated textfile `./out/discipline.tsv`.

6. Iris

   - Read the Iris dataset from `https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data` directly from the Internet.
   - Name the columns in the following way: "sepal_length (in cm)", "sepal_width (in cm)", "petal_length (in cm)", "petal_width (in cm)" and "class".
   - Set values of the rows 10 to 29 of the column 'petal_length (in cm)' to missing.
   - Replace missing values with 1.0.
   - Save the comma-separated file as `./out/iris.csv` without index.
   - Visualize the distribution of all of the continuous variables by "class" with a catplot of your choice.
   - Save the figure as `./out/iris.pdf`.

7. Big data and memory efficiency

   - Load the comma-separated data from `https://query.data.world/s/wsjbxdqhw6z6izgdxijv5p2lfqh7gx` into a DataFrame '.read_csv()'
   - Inspect the DataFrame using `.info()` and with `.info(memory\_usage="deep")`. What is the difference between the two calls? How much space does the DataFrame require in memory?
   - Create a copy of the object with only columns of type object by using `.select\_dtypes(include=['object'])`.
   - Look at the summary of this object (using `.describe()`). which columns have very few unique values compared to the number of observations?
   - Does it make sense to convert a column of type object to type category if more than 50% of the observations contain unique values? Why/Why not?
   - Convert all columns of type object to type category where you deem this appropriate.
   - What is the final size in memory?
   - Could above routine have speeded up somewhere?

8. Euro 2012 II

   - Load the data from `./data/Euro_2012.csv` into a DataFrame.
   - Add a column "Wikipedia" displaying the Wikipedia page ID of the country (use `.apply()` on column "Team" to apply the corresponding function which queries Wikipedia individually).
   - Output the data as semicolon-separated `./out/wikipedia.ssv`.

9. Google Maps I

   - Read Google's address from Google Maps API as *json*
   - Convert *address_components* to pandas DataFrame
   - Add new column name "firm" with "google"
   - Add latitude and longitude as new columns
   - Remove the row where "short_name" is "US".
   - Output as comma-separated file called `./out/google.csv` without header.

10. Google Maps II

    - Read the address of the MPI for Innovation and Competition from Google Maps API as *json*
    - Convert *address_components* to pandas DataFrame
    - Add new column name "firm" with "MPI for Innovation and Competition"
    - Add latitude and longitude as new columns
    - Output the DataFrame as semicolon-separated file called `./out/MPI.ssv` without index.