CS2006 Python Project 2: Analysing a dataset

Alexander Konovalov

Due date: Tuesday 18th April 2017 (week 11), 21:00
33.3% of Overall Mark for the Module

Overview

The objective of this project is to get an experience of using Python to work with large datasets. By completing this project, you should further enhance your Python programming skills and get more detailed insights into the projects from the Python ecosystem, for example, libraries for data analysis and visualisation, and Jupyter notebooks.

The Dataset

The dataset [1] contains a random sample of 1% of people in the 2011 Census output database for England and Wales. It includes records of 569,741 individuals. Each individual corresponds to a line in the CSV file containing information on 18 separate or variables. This dataset is provided for educational purposes[1].

The dataset is stored in the directory `Practical/Python2/data`, in particular:

- `census2011.csv.gz` - compressed CSV file with the dataset (renamed version of the CSV file from [1]; also line 1 is deleted to ensure that headers are in line 1 instead of line 2)
- `MicroDataTeachingUserGuide.pdf` - user guide on the dataset
- `MicroDataTeachingVariables.pdf` - variable list which explains each of the 18 census variables within the dataset

Basic Requirements

In this project, you have to:

1. check the consistency of the initial (raw) data, and refine them if needed
2. carry out certain data analysis and visualisation tasks
3. provide a Jupyter notebook for the replication of your results

Your project should demonstrate the following characteristics, as inspired by [2]:

1. **correctness**
2. **repeatability** (i.e. you should be able to rerun easily the whole procedure from checking the consistency of the raw data to getting the final outcomes on the lab machine or another machine used to work on the project)
3. **replicability** (i.e. the marker should be able to replicate it on the lab machine with the same outcomes)

[1] for terms and conditions, see `https://www.ons.gov.uk/census/2011census/2011censusdata/censusmicrodata/microdatateachingfile/microdatauserguide`

4. **reproducibility** (i.e. it should be possible to use it to work with another data set with a similar structure and analyse the same set of metrics)

5. **reusability** (i.e. it should be possible to reuse some project components in different experiments).

While satisfying the 1st three requirements (correctness, repeatability and replicability) will be directly assessed in the course of marking your project, satisfying the latter two requirements (reproducibility and reusability) will be mainly judged by the project organisation and design choices (however, you may consider analysing additional datasets to demonstrate these requirements in practice).

You should first understand the structure of the raw data and the meaning of all information contained in the dataset. For that purpose, you have to read both `MicroDataTeachingUserGuide.pdf` and `MicroDataTeachingVariables.pdf`. Then you may open the CSV file using the Jupyter notebook provided in `Practical/Python2/code/census2011.ipynb`, revise the Jupyter notebook for Lecture 8 "Exploring a dataset" and apply the same techniques to the dataset from this Practical. You will need to check that all entries match the specification from `MicroDataTeachingVariables.pdf` and refine them if necessary. At this stage, you may find some useful suggestions in [3].

You should use the following libraries:

- pandas (`http://pandas.pydata.org/`) – Python Data Analysis Library
- Matplotlib (`http://matplotlib.org/`) – Python 2D plotting library

Useful links are Matplotlib tutorial by Nicolas Rougier (`http://www.labri.fr/perso/nrougier/teaching/matplotlib/`) and Pandas Cheat Sheet (`https://github.com/pandas-dev/pandas/blob/master/doc/cheatsheet/Pandas_Cheat_Sheet.pdf`).

Keeping Jupyter notebooks under version control, you may discover that standard tools to inspect changes and perform merges do not work well with Jupyter notebooks. If you will be using Git for version control, then you may find useful the specialised tool for diffing and merging Jupyter Notebooks, called nbdime (`http://nbdime.readthedocs.io/en/latest/`).

You may use other specialised libraries as well, provide all requirements being documented, supply with clear installation instructions and work on a lab machine. You can also use venv or virtualenv to have your own setup. For further instructions, see School's Wiki at `https://systems.wiki.cs.st-andrews.ac.uk/index.php/Python_on_Linux`. To install Jupyter (`http://jupyter.org/`), it is recommended to use `pip install --user jupyter` and then open Jupyter notebooks with `env PATH=~/.local/bin:$PATH jupyter notebook`.

The minimal requirement for the project is to:

- refine the dataset:
  - develop a procedure that will check that the data match the specification contained in the `MicroDataTeachingVariables.pdf` file.
  - in case of any inconsistencies found, produce new file with refined data to be used in the subsequent analysis (this process should be automated in case one may need to re-run it)
- perform the descriptive analysis of the dataset:
  - determine the total number of records in the dataset
  - determine the type of each variable in the dataset
  - for each variable except "Person ID", find all values that it takes, and the number of occurrences for each value.
- build the following plots:
  - bar chart for the number of records for each region
  - bar chart for the number of records for each occupation
  - pie chart for the distribution of the sample by age
  - pie chart for the distribution of the sample by the economic activity

- provide the Jupyter notebook to re-run the analysis (starting from the raw or refined data).

Important! Addressing both basic and additional requirements, you need to ensure that the output provides textual interpretations for values of the variables (see `MicroDataTeachingVariables.pdf`) instead of their alphanumeric codes.

Completing these requirements and demonstrating reasonable coding standards, efficient use of pandas and Matplotlib, accurate and informative graphics would be marked with the highest grade 13. In order to achieve a grade higher than 13, you should implement some of the additional requirements specified in the next section.

Additional Requirements

Note: It is strongly recommended to ensure that you have completed the Basic Requirements and have something to submit before you attempt to deal with the Additional Requirements.

In order to achieve a grade higher than 13, you should implement some of the additional requirements below. You should not be limited by these and should feel free to discuss and implement any other feature that you consider useful (please make sure that it will be described in the report!)

- **Easy:** using `groupby` objects[2], produce the following tables:
  - number of records by region and industry
  - number of records by occupation and social grade
- **Easy:** Learn how to use pandas to perform various queries, for example, to analyse:
  - the number of economically active people (see "Economic activity") by region.
  - the number of economically active people (see "Economic activity") by age.
  - whether there are any discrepancies between the student status given as a yes/no answer to the question "Student (Schoolchild or full-time student)" and answers on the question on "Economic activity".
  - the number of working hours per week for students (codes 4 and 6 in "Economic activity").

  You may use plots as you would feel appropriate to illustrate your findings.
- **Medium:** Build 3D plots based on tables from the 1st Easy requirement.
- **Medium:** Use ipywidgets (`https://ipywidgets.readthedocs.io/en/latest/`) to control plot properties (for example, to select the region with the dropdown widget, and age with the the slider).
- **Hard:** Use the map to display the data for each region on the map (for example, for each region put a pie chart at its geographic centre to describe the distribution of the sample by some variable, with the diameter depending on the number of records for that region).
- **Hard:** Analyse some other, possibly much larger, data sets. You may find some interesting datasets from:
  - Office for National Statistics: `https://www.ons.gov.uk/`
  - The Scottish Government: `http://statistics.gov.scot/`
  - HESA: `https://www.hesa.ac.uk/data-and-analysis`
  - figshare: `https://figshare.com/`

  (Please use datasets with at least 10000 records)
- **Hard:** Use virtualisation tools, such as e.g. Docker (`https://www.docker.com/`) to provide a complete environment for reproducing your experiment.

[2] see `http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.groupby.html`

Deliverables

For this practical, you will have to submit a reproducible report in the form of a Jupyter notebook. This section explains how to organise this and other parts of the submission.

Hand in via MMS, by the deadline of 9pm on Tuesday of Week 11, a single `.zip` or `.tar.gz` file containing two files called `README.txt` and `CONTRIBUTION.txt` file, and three top level subdirectories called `data`, `code` and `report`.

The file `CONTRIBUTION.txt` should be prepared individually by each member of the group. It should contain your matriculation number and describe your own contributions to the project.

The rest of the submission should be the same for everyone in the group.

The `README.txt` file should briefly describe the content of your submission and provide installation instructions for the dependencies needed to run your code. The marker should be able to run your code on CS lab machines.

Furthermore, the `data` directory should contain the data (raw and, if needed, refined), and the `code` directory should contain the code in the form of Jupyter notebooks and supplementary `.py` files if needed. There may be further subdirectories and other files, if necessary. The Python source code may be contained in Jupyter notebook or in separate files that the notebook will import, but in both cases it should be well-commented.

There is a special procedure for the submission of Jupyter notebooks: for each notebook, you should provide both an `.ipynb` file with all outputs, located in the `code` directory, and its export to PDF format, located in the `report` directory. The marker should be able to clear all outputs in the `.ipynb` file and rerun it to produce the same output as shown in the `.pdf` file.

The recommended procedure to export a Jupyter notebook to PDF is to go to "File" → "Print Preview" in the Jupyter menu. This will open a new tab in the browser. You need now to use the browser menu to print this tab into a PDF file. DO NOT use "File" → "Download as" → "PDF via LaTeX" option.

You do not have to write a separate report for this Practical, since the Jupyter notebook will combine the code and its output with the project report. Instead, you should be using markdown cells in Jupyter notebook, interleaving them with code cells to describe each steps of your analysis, and also include the following:

- An introduction with the summary of the project indicating the level of completeness with respect to the Basic Requirements, and any Additional Requirements.
- A description of any known problems with your project, e.g. any Basic or Additional Requirements that are not met, but were attempted to be implemented.
- Details about any specific problems you encountered which you were able to solve, and how you solved them.
- A note on the level of reproducibility and reusability of your analysis.
- An accurate summary of provenance, i.e. stating which files or code fragments were:
  1. written by you;
  2. modified by you from the course material;
  3. sourced from elsewhere and who wrote them.

Also, remember that the place to document your individual contributions is the `CONTRIBUTION.txt` file in the root directory of your submission.

Marking Guidelines

This practical will be marked according to the guidelines at `https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html`.

To give an idea of how the guidelines will be applied to this project:

- A simple prototype implementation which opens the CSV file in Jupyter notebook and reports the number of records in it, combined with a report, should get you a grade 7.

- A solid basic implementation which addresses all the Basic Requirements with a well commented, documented and fully working code, combined with a clear and informative report, should get you a grade 13.
- To achieve a grade between 13 and 17, you have to implement between one and two requirements marked as **Easy** and between one and two requirements marked as **Medium**.
- To achieve a grade higher than 17 you have to go beyond the requirements described above by either suggesting and implementing extra Easy and Medium additional requirements, or by implementing some of the Hard ones, providing well-commented and documented code, accompanied by a clear and informative report.

Finally, remember that:

- Standard lateness penalties apply as outlined in the student handbook[3].
- Guidelines for good academic practice are outlined in the student handbook[4].

Finally

The project is very much open-ended, so please feel free to discuss your own agenda in addition to implementing Easy and Medium requirements in order to score a high grade. Be creative, have fun, and produce something which you would be interested to make and would be proud of!

Finally, please let me or your tutor know if you have any questions or problems! There will be four Tutorials during your work on this project: March 27th, April 3rd, April 10th and April 17th. Please be prepared to discuss the following questions:

1. How did you divide responsibilities within your team?
2. Which requirements have you implemented so far?
3. Any requirements or language features you are having difficulty with?

A rough guideline is that you familiarise yourself with the dataset and documentation and be prepared to talk about technical details and implementation plan at the Tutorial on March 27th.

You should have a prototype by April 3rd and complete the the basic implementation and Easy requirements by April 10th. That gives you roughly a week to work on Medium and Hard ones. Of course, these are just guidelines, and you may achieve much faster progress!

You may contact me by email `alexander.konovalov@st-andrews.ac.uk` or find me in my office JC 1.02, and we will be also meeting at the Tutorials at 11am on Mondays and in the JH lab at 11am on Fridays.

Alexander Konovalov
March 10th, 2017

References

[1] 2011 Census: Teaching file. Office for National Statistics. Release date: 23 January 2014. Accessed on 8 March 2017. `https://www.ons.gov.uk/peoplepopulationandcommunity/educationandchildcare/datasets/2011censusteachingfile`.

[2] N. Chue Hong (2014): Better Software, Better Research: Why reproducibility is important for your research. figshare. `https://dx.doi.org/10.6084/m9.figshare.1126304.v1`.

[3] P.J. Guo (2015): Parsing Raw Data, `http://pgbovine.net/parsing-raw-data.htm`

[3] `https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html`
[4] `https://info.cs.st-andrews.ac.uk/student-handbook/academic/gap.html`