Empirical Project: Data Science in Economics (BEE2041)

March 26, 2024

Background

The final empirical project for BEE2041 entails the generation and writing of a data driven blog post. This project covers all of the contents of the course, is graded out of a total of 100, and seeks to assess your mastery of a range of different tools discussed during the course which are relevant for data science work. You can also explore the use of other platforms and ways to present and visualise information.

While you have flexibility with regards to the tools you use to do this assignment, there are a number of specific requirements. Firstly, you must use git to track your work and publish a repository of your project to github. Secondly, you should use Python, the command line, or SQL (or various of these if necessary) for any data cleaning, and use Python for any web-scraping, modelling and generation of graphs and output. Thirdly, it is expected that some element of your project either uses web-scraping to generate data, or works with analysis using tools such as those laid out in unit 5 (regression modelling, causal forests, etc.)

Note that **it is not** necessary for you to use both web-scraping and the more advanced modelling tools discussed in unit 5 of the course. While you could use both if you like, the only requirement is that at least one of these is incorporated in some way in your work. Also, please note that while this is viewed as a wholistic evaluation of what we have learned in the course, there are surely elements which we have studied which will **not** be incorporated in your empirical project, and this is perfectly fine. I am aware, for example, that this project does not lend itself particularly to work with relational databases and SQL and this is fine. Apart from the requirements discussed in the paragraph above, you should see your goal in this project to use any tools you consider neccessary to generate a compelling and striking data science project, additionally ensuring that all steps can be followed and replicated by someone else. Specific details of this are laid out in the **Instructions** section of this document.

Submission Instructions

Please submit your project as a single compressed .zip file named as STUDENTNUMBER.zip (replacing STUDENTNUMBER with your student number). This file should contain (at least) the following elements:

- A README file containing instructions on how to replicate your work
- A well-structured project folder including all required source code, data, and so forth
- The output from running the code when you run it on your machine

- The git folder which tracks this code (provided that you have used git you will not need to do anything, as this should be available in the .git folder)
- A link to the blog post.
  - If you include your "blog post" simply as a jupyter notebook, this can just be an .ipynb file. Please call this blog.ipynb
  - If you publish your blog post on some web platform, please include the link to this URL ensuring that I can access it from the web. Please include this address in the file called blog.txt (which just requires the URL that I can then copy and paste)
- A link to the github repository
  - If you include your "blog post" simply as a jupyter notebook, please include the github link at the end of this file
  - If you publish your blog post on some web platform, please include the link to your github folder in the in the file called blog.txt

Thank you.

Your scripts must be sufficient to reproduce your work, and by reading the README and scripts I should have a good idea of what I need to do to replicate everything you do. If you submit your blog post as a Jupyter Notebook file, it is your responsibility to ensure that this will open without errors. If you wish to be sure that the Jupyter file can be read easily, you could also hand in the output of your Jupyter file as a separate pdf (but please also include the .ipynb file too).

The deadline to hand in this empirical project is **Thursday, April 25th 2024, at 15:00**. The project must be handed in using the ELE page.

Collaboration and Academic Conduct

You are encouraged to think about this project in groups or ask each other for help. If you do, please simply ensure that the idea you produce in your blog post is your own (and this is not the same as what others examine), and ensure that you write your own code.

The University takes poor academic practice and academic misconduct very seriously and expects all students to behave in a manner which upholds the principles of academic honesty. Please make sure you familiarise yourself with the general guidelines and rules from the following link `http://as.exeter.ac.uk/academic-policy-standards/tqa-manual/aph/managingacademicmisconduct/`. Thank you.

Grading

The assignment will be marked by replicating code and by reading your blog post. In particular, the final mark will consider the following criteria, as well as their weighting:

Table 1: Grade Description

Element	Score
Clear README and documentation of project	5
Logical Structure of project directory	5
Evidence of git tracking and provision of github link	5
Code replicability	5
Quality of analysis and output	40
Use of appropriate tools including some element from unit 5 of the course	30
Creativity of product and final format of blog	10
Total	/100

The empirical project makes up 70% of the final grade for the course. A mark of 50 or above on this assignment is a passing mark. 70 or above indicates a very good performance, with 80 (or above) indicating a truly outstanding project.

Instructions

For this empirical project, you will need to write a data-driven blog post. This data-driven blog post can be based on whatever data that you wish, and can be on any theme that is of interest to you! It is not necessary that this be some sort of "academic" theme: it could range anywhere from entirely academic themes (for example documenting links between economic growth and well being or working with data from some paper), to entirely interest-based (for example, some analysis of sports data, or data from some television program, music, and so forth). What is important for this post is NOT the theme you choose (though ideally it should be something interesting for you!), but rather how you use data and computational tools to provide your readers some insights related to this topic.

Structure: There are plenty of tutorials online that provide examples. One particular source you may wish to use to guide you is the following: https://playbook.datopian.com/dojo/writing-a-data-oriented-blog-post/#_11-simple-steps-to-create-data-driven-blog-posts. There are some things you will not need to worry about here (such as discussion of search engine optimisation).

Some examples of data-driven blog posts are provided below. The first set (from Pudding) are very elaborate and fancy:

- https://pudding.cool/2017/08/the-office/

- https://pudding.cool/2017/08/screen-direction/

There are some more examples on their website, including some interactive posts: https://pudding.cool. You are **NOT** expected to write a post to this level of production (and certainly not an interactive one), but you may consider these as an example of what the upper limit for what work like this can be when done by a professional team. Nevertheless, you may find their tutorial useful: https://pudding.cool/process/how-to-make-dope-shit-part-1.

Thinking about the type of work which you may be able to feasibly do as a single person under a time deadling, the following examples might give you a more realistic idea of posts which have done a nice job in a slightly more feasible way:

- https://buffer.com/resources/social-media-language/
- https://datahub.io/blog/automated-kpis-collection-and-visualization-of-the-funnels

Length: Your post should have between 1,000–2,500 words, and it should contain 4-7 pieces of output (e.g., plots, tables, summary statistics, and so forth).

Code/Platform: You may wish to simply generate your post in a Jupyter notebook where you include the actual code you have used to generate output, and text as markdown. However, you may also prefer to use a more elaborate set of online tools for publishing the post, and you are more than welcome to do so! You could explore many possibilities, such as things like HackMD (https://hackmd.io) to create an online link to present your post. If you do this, you don't even necessarily need to create an account on HackMD; you can login using your github account. This will provide you with quite a light and fresh visualisation, for example a simple output looks like this: https://hackmd.io/@linnil1/Sy0p1s9ZX. Alternatively, you may wish to consider alternative platforms like github sites, quarto, and so forth. Provided you can send me the post in some format that I can read, I am happy for you to use whichever tools you feel best allow you to create a creative and interesting post.

Audience: You should assume that the audience of your post has some knowledge and interest about the topic you are showing, but they have no idea about the data set. Assume that your audience is well-educated and has knowledge that is comparable to a university 2nd-year student, but one who has never worked with programming.

Some final considerations: While there is a marking scheme discussed above, thinking about the following elements will perhaps help with your output.

- Insightfulness: is the analysis insightful and compelling? Are there some interesting, thought-provoking, and/or surprising findings?

- Soundness: Is the analysis sound? Are all comparisons fair (not comparing apples to oranges)? Was proper filtering done? Are plots used appropriately based on data types (e.g., using bar plots for categorical data points, scatter plots for independent data points, lines for time series)?
- Presentation: Is the narrative coherent? does it all read as a one story? Is it easy to understand the post? Is it easy to follow the ideas and the main story? Are there appropriate (and readable) labels, legends, and captions provided for plots?
- Visual appeal: are the plots visually appealing (beautiful, appropriate colours, non-trivial yet simple plots)?
- Pre-processing: Has there been an appropriate level of pre-processing of data? Is the code easy to follow? Is the code efficient? Did the student put some effort in cleaning or re-shaping data?

Questions?

If you have questions related to the interpretation of this project, please post them on the Forum on the ELE page of the class. I have made a specific thread for this, as this way any of my responses to queries related to general project information will be availalble for all.