Class Exercise #1

Dr. Andrea De Angelis
Spring Term 2024

Submit by Friday 29.03 h. 17:00

Introduction

This first class Exercise (CE#1) lets you experience a real-world fully-reproducible and collaborative workflow in a small data analytics team using GitHub and R. You will have to develop an empirical analysis end-to-end, from data collection to the communication of the findings in a reproducible R Markdown report. Thus, you will touch upon all data analysis stages: data import, pre-processing, visualization, modelling, and reporting. Furthermore, you will familiarize yourself with Version Control systems using GitHub.

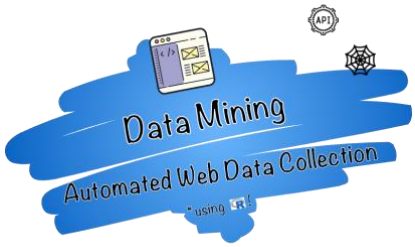
Goal

Conduct a simple data analysis using one of the following data sources:

- [Social science track] **European Social Survey** [link], **International Social Survey Programme** [link], or the **European Value Study** [link];
- [Political Science track] Any of the datasets in the GitHub repo by Erik Gahner [link]
- [Data Science track] Any of the Kaggle datasets [link]. Search and quickly self-learn what kind of variables are contained in these datasets: what could you use them for?

You must: setup a GitHub repository for the project and work with the Git toolbar in RStudio, create an appropriate folder structure, choose one dataset, and then work with both **R scripts** (e.g., one to import the data, one to make the data pre-processing…) and **one Rmarkdown report** (where you show your analysis) following the IMRaD (Introduction, Methods, Results, and Discussion) scheme [link].

More specifically, in terms of the analysis you must:

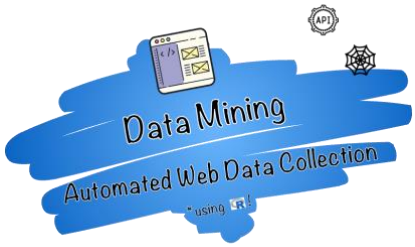


1. Find a viable **research question**. Examples: *Are older individuals more right-wing? Are women more post-materialist? Are individuals in Eastern-European countries less interested in politics?*
2. Specify an **empirical hypothesis** addressing the research question. The hypothesis should be specified with a direction. For example: *older individuals are richer and therefore may oppose progressive taxation. For this reason, we hypothesize that older individuals will report more conservative ideology.*
3. **Test** the empirical hypothesis. The tests should include at least one **visualization** (e.g. multiple boxplots showing the average left-right position by age cohort) and **one numeric/statistical test** (e.g. descriptive summaries showing the mean left-right ideology by age cohort or, even better, a statistical test of the hypothesis like a linear regression model reported in a table or prediction plot).
4. **Draw a conclusion** based on empirical evidence.

Expected deliverables

The expected list of deliverables includes:

- A **GitHub based project** using a fully-reproducible workflow, including:
  - A short introduction to the project (2/3 lines).
  - A consistent folder structure explained on GitHub in the `readme.md`. E.g.: `raw_data, recoded_data, src, figs`…
  - Scripts with meaningful names (e.g.: `00_setup.R`, `01_data_preproc.R`, `02_figures.R`)
  - Reader-friendly code: meaningful variable names (e.g., `party_size` instead of `v21`, foldable sections and comments).
  - All activities should occur on GitHub (i.e. pushing, pulling, opening pull requests, opening issues, discussing issues with @mentions…).
  - The GitHub repo must be public and placed inside the course organization [create it here, not in your personal GitHub space]. To do this you must be member of the organization (ask for access or send me an email if you are not).



- A **short report (max 250 words**, about one page or more to fit plots**)** with the research question, hypothesis and main tests, written using RMarkdown and stored in `/output/report.Rmd`. The report must be compiled to HTML (the .html file should also be there).
- A few short **R scripts** (e.g., one to import the raw data; one to preprocess the data and save it to the dedicated folder; one to create one figure, one with the statistical test). The R Markdown document can of course use the code from these scripts.
- Report the **summary of individual contributions** on the grades' file at this link (use the tab "Class exercise #1"). Indicate the following:
    1. Link to the repo and team name;
    2. number of individual commits (aim at having at least five, but the more the better);
    3. number of issues opened (at least two);
    4. number of pull requests opened, accepted and merged (at least two).

Notes

- You can find these values by clicking to "Insights" and "contributors" [example link]. At [this link] you can find an **example CE** from a past edition: click on "Code" and "Download as zip" to store it on your laptop (please use the folder "output" and not "docs" to place the report as in this example).
- **Important**: each team member should place the data in their local folder but should not push it to the repo. In fact, this may infringe the conditions for data release. Add the file to the **`.gitignore`** file as described here.

Deadline

Deliverables are expected **by Friday 29th of March at 17:00**. You don't have to send anything, I will just clone your GitHub repository and check that the grades' book is complete.

Assessment

The exercise is evaluated considering the individual GitHub activity as well as the report as a whole. As an individual learner, you should **make a real contributions** to the group project,

e.g.: cleaning the data set, building a beautiful visualization, writing a short description of the analytical strategy, revising and reviewing the code, adding one statistical test, etc...

To this end, you must adopt the standard GitHub flow:

- Making many `git commit / git push / git pull` cycles.
- **Open an issue** describing a new feature you are going to implement (e.g., a new plot)
- Open a **new branch** connected to the issue.
- Letting others check and contribute to your new feature opening a **pull request**.
- Provide **feedback on a pull requests** opened by your colleagues.
- Incorporate the new code into the project **merging** the pull request.

Feel free to experiment: we are just here to learn!

Tips

- ***Tip #1:*** embrace **imperfection** as that thing that allows you to get things… done!
- ***Tip #2***: **collaborate** also with the other teams!
- ***Tip #3:*** good code… just works. Don't waste time trying to make you code more efficient or elegant (although this is something to keep in mind as you progress).

Code of Conduct

Remember to "be nice", intended in its widest possible sense. Declinations of this principle demand you to:

- Use welcoming and inclusive language.
- Be respectful of different viewpoints and experiences.
- Gracefully accept constructive criticism.
- Focus on what is best for the community.
- Show courtesy and respect towards other community members.

Disagreement can naturally arise during collaborative work and its management is part of job of being a good data science professional (and human being). If you believe that someone is violating this code of conduct, I ask you to report the violation to me to allow me to take appropriate action. No form of harassment will be tolerated.