IT skills for research

syllabus

Dmitry Borisenko Igor Pozdeev

Contents

1 introduction

This course's sole purpose is to introduce an aspiring researcher to certain useful software and IT concepts which are to research what icing is to cake: not the defining part but not inessential either. After all, research is not only idea generation, data analysis and publication – there are elements such as project and data management, collaboration, knowledge transfer etc., and while universities do a good job at teaching the former, the latter are often underrepresented in curricula, despite being efficiency boosters and impact multipliers.

Many a concept to be taught in this course are no big news in software development, a field in lots of ways similar to academic research: a developer starts with an idea, sets up a team, writes code, runs tests, corrects errors, creates documentation, ensures user friendliness and publishes the product – to any of these a scholar would find an analogue in the own routine. The similarities ensure that knowledge transfer is possible, and since the IT is the more dynamic field, the scholars mostly end up at the receiving end. Good for us – we can learn!

2 objectives

Learn some of the best practices required to efficiently conduct research today:

- **command line**:
  use cases, commands, scripts, environment variables, file manipulations, remote connection, authentication, cronjob;
- **project management**:
  development environment, directory layout, text editors, coding modes;
- **version control (git)**:
  use cases, installation, git areas, tracking and ignoring files, undoing things, concurrent editing, branching, github;
- **data management**:
  organizing a database, database api, memoization, data file types, web api, introduction to SQL;
- **visualization**:
  cognition science basics, colors, tricks for better visuals;
- **latex**:
  installation, formats and engines, command line interface, packages, beamer, custom styles, bibliographies;
- **research apps**:
  interactivity, jupyter notebooks, R shiny apps;
- **reproducibility**:
  open source concepts, environment export, makefiles, testing.
- **cloud computing**:
  remote connections, virtual machines, Amazon Web Services (AWS);

3 requirements

Intermediate knowledge of Python or R: how to write a function, plot a chart, format string values, read to and write from text files etc.; previous exposure to research projects.

4 schedule

week 1	introduction + software + command line
week 2	environment
week 3	version control p1: working solo
week 4	version control p2: collaboration
week 5	data management p1: setting up a database
week 6	data management p2: web api
week 7	data management p3: introduction to SQL
week 8	visualization
week 9	publication p1: latex
week 10	publication p2: apps
week 11	reproducibility
week 12	cloud computing

5 software

We will be using Slack, a command line terminal, Python or R plus jupyter (potentially with an IDE), an advanced text editor (such as Sublime Text, Atom or VS Code), SQL management software, git (potentially with dedicated software or integration into the text editor of choice), as well as LaTeX (potentially with dedicated editors).

6 examination

Please work in groups of 2-4 people and set up one repository per group – this will be the place to keep the midterm and final assignments on separate branches. Groups must be formed before the 4th lecture starts, the only allowed change after this deadline being leaving one.

6.1 midterm

(25%, group assignment, same grade)

Please do the first 6 end-of-chapter exercise sets by the beginning of the 7th lecture and push the solutions to branch ‘midterm‘.

6.2 final

(75%, group assignment, same grade)

Please submit a little research project adhering to the following requirements:

- make it hosted as a single repository on Github with all group members as contributors;
- structure it neatly and concisely, avoid the clutter of folders and files for which third people have no use (use .gitignore);
- keep working on it consistently, avoid bulk commits;
- create a little well-documented database with the data you use, ideally with update methods;
- design figures and tables to support your findings, keeping them in line with visualization standards discussed in class;
- write a short paper about it using LaTeX or Jupyter/Shiny, populating it with sections, a table of contents, as well as the above tables and figures;
- write a beamer presentation for your project using LaTeX;
- create one interactive app (Shiny or Jupyter notebook) describing the main finding and presenting several robustness checks;
- make sure the results are reproducible by ensuring that the coding environment is exported and concise documentation is present;
- in the ideal case, anyone who could pass this course should be able to see and reproduce your findings from scratch.