Exercise 1: Information Coding & Data Structures

Exercise 1 for the lecture ‘Foundations of Data Science’

Prof. Dr. Karsten Donnay, Assistant: Marcel Blum

This session covers

- General data science process
- Introduction to git
- Introduction to our working case
- Data import in R

University of Zurich[UZH]

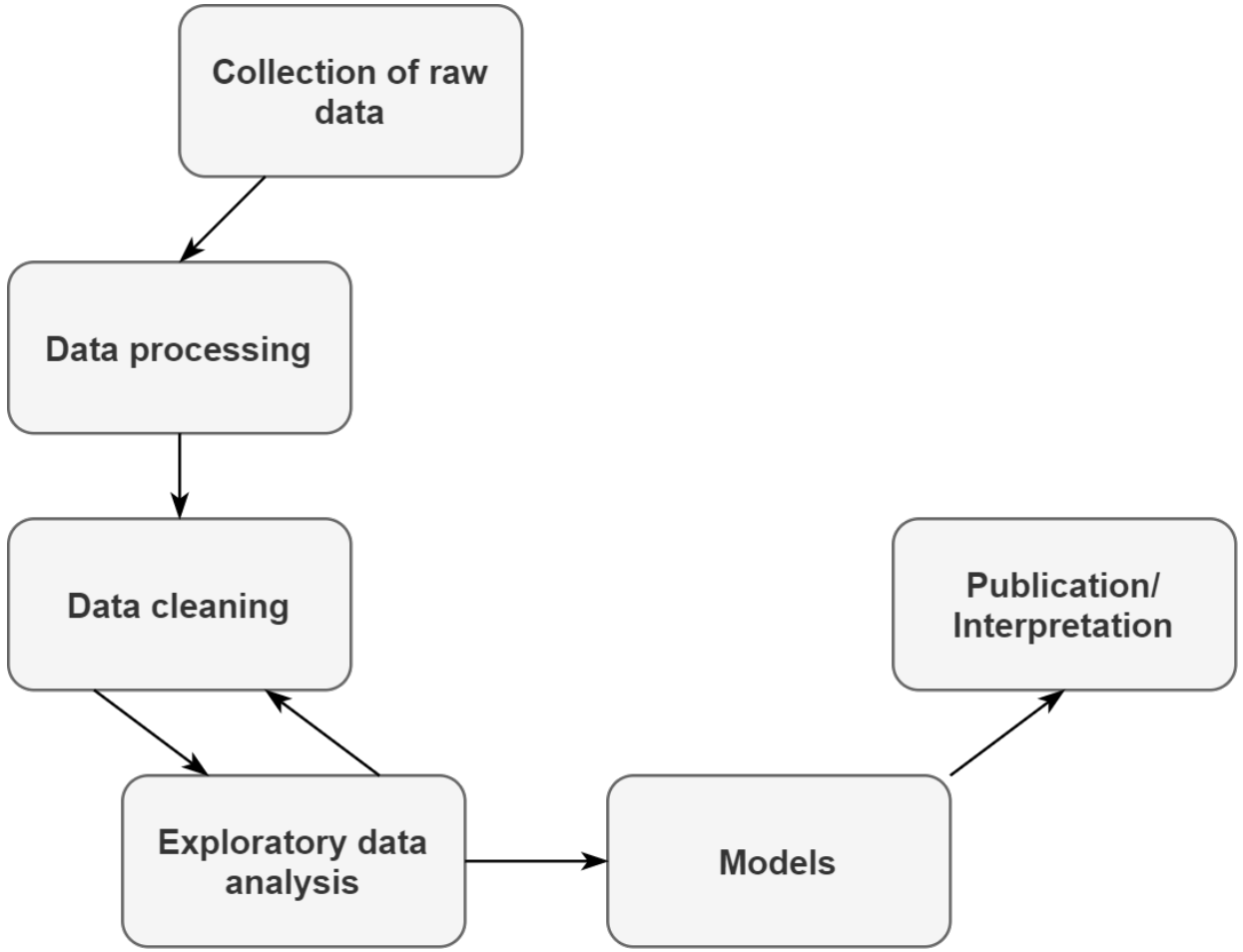
Department of Political Science

Data science process

Data science process

"Make sense of new and/or large data and communicate insight"

- Access innovative and **large data** resources.
- **Process data** to make it machine readable.
- Use statistical methods or machine learning to **detect structure** in the data.
- **Provide** meaningful **insights.**



Data science process

What is good science?

- **peer review**, **transparency** and **replicability** (Apart from other criteria)
  - Karl Popper (1934): “non-reproducible single occurrences are of no significance to science”.
  - Emphasizes the need for **publication** of employed **methods**, **documentation** of the **data collection** and cleaning process, and the **provision of datasets**.

Data science process

Why is reproducibility in data science difficult?

- Available **resources** (e.g. computing power, storage)
- Data on the Internet often **in flux** (e.g. websites change, Tweets get deleted…)
- **Permission** to use data (e.g. Facebook data)

- **Git** is one way to improve on one part of the reproducibility crisis: make method transparent and easily accessible.

University of Zurich[UZH]

Department of Political Science

Introduction to git

Git (Version control)

- The ‘Dropbox’ for programming
- Documents the different stages (versions) of files
- Makes it easy to track changes and restore previous versions.
- Enables controlled collaborations with others
- Eases the publication of work and increases transparency

Version control with git

- What's the fundamental idea?
  - We want to systematically keep track of files for multiple people working on the same project while avoiding editing conflict
  - Need people to be able to work on it locally (offline) but regularly merge their edits into a central (online) repository
  - Requires a set of standard procedures and methods to avoid/handle conflicts and standardize how changes are tracked

- Two step-process for making changes
  - "committing" (local) changes to the data/ files
  - "pushing/pulling" to/from the remote repository

Version control with git

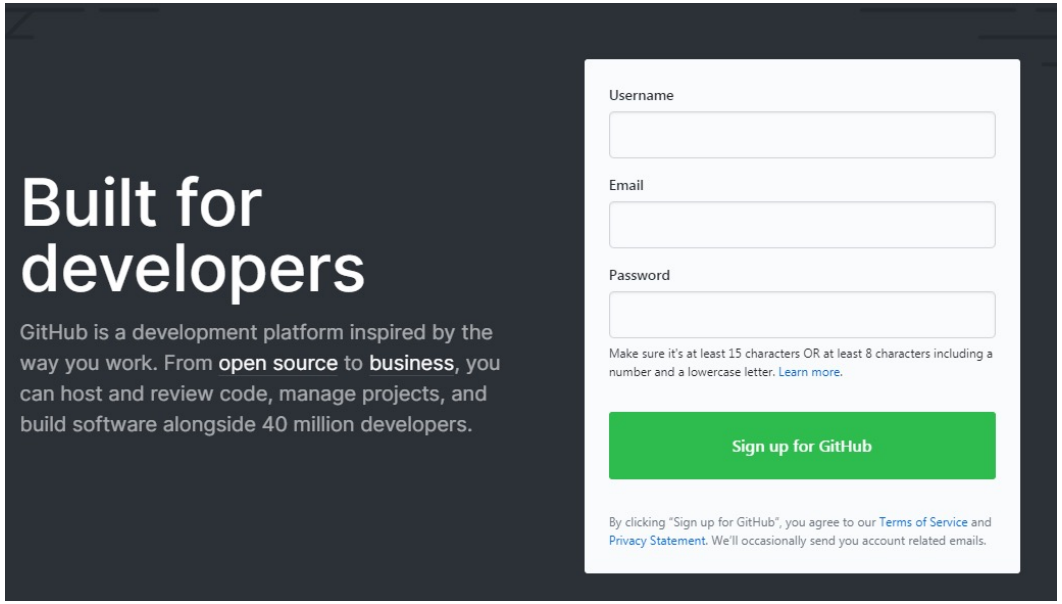
- How does it work concretely?
  - If a repository already exists, we need to “clone” it to our local machine, i.e., we download a full copy of the remote repository including history of changes etc.
  - If we add a new file that has to be version tracked, we need to “add” it to the system so that it knows that you are now tracking it
  - If you make a local change, you have to “commit” it to the remote repository and you should document your change (with a short title and, optional, a more detailed info)
  - The synchronization with the remote repository is then via “pushing” changes to the remote or “pulling” changes from the remote
    - If there are any conflicts, they have to be resolved at this stage

Popular platform: Github (alternatives: bitbucket, Gitlab)

- Go to https://github.com/
- Select your role (student) and the purpose of usage and confirm the email.
- Create your first repository

  e.g. `github.com/css-zurich/datascience`

- Download and install git
  - Easiest is to use one of the GUI Git clients
  - Great overview: https://git-scm.com/downloads/guis
  - Alternatively: install Git directly, use command line tools to do the same things
  - Introduction: https://git-scm.com/book/en/v2/Getting-Started-The-Command-Line



Version control with GitHub

- If you decide to use one of the Git clients, then all of the procedures described above are already conveniently implemented and the client even reminds you what to do next…

- If you want to use GitHub for your work, you can apply for educational discount (GitHub Pro for free) that allows “private” repositories:

  https://education.github.com/

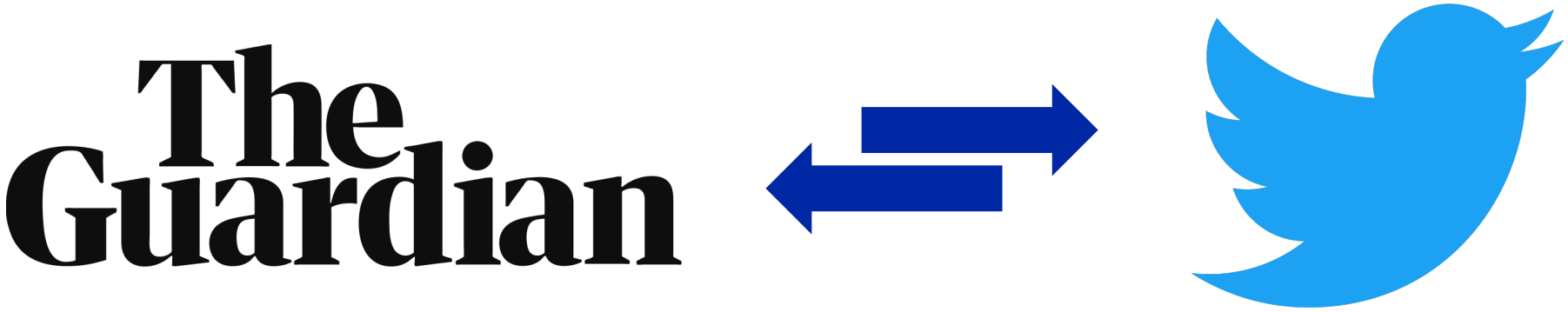
Introduction to our working case

Introduction to our working case

- While we talk about the case, you can already “clone” the repository to get the exercise files.
  - https://github.com/css-zurich/fds-2020/
  - All files are also available through OLAT

Introduction to our working case

- Hypothetical use case: How do characteristics of news articles relate to reactions on social media?
- Our two examples: The Guardian for news and Twitter for social media data



Introduction to our working case

- **Goal**: combine news data with social media data
- There are already **established packages** in **R** that retrieve data from these platforms. However, we will use these platforms to **build some applications from scratch** and demonstrate core concepts of data science using R.
- Keep in mind: before starting to build your own application, **do some research on existing work**. Often there are already established ways that work efficiently.
- After the five exercises you will be able to…

  …manage and **process data efficiently**.

  …**manipulate text** into formats that you can work with.

  …**read data** from **websites** into R.

  …retrieve data from application programming interfaces (**APIs**)

Introduction to our working case

Why use the **Internet** to collect data?

- Has a plethora of **useful data sources**:
  - Government publishes data (e.g. speeches, voting…)
  - Social media data to analyze human communication
  - News data for public discourse and attention to events
  - User interactions with e.g. products (Amazon reviews), Films (IMDB)…
- Why is this **relevant**?
  - Re-evaluation of existing research with new data
  - Enables entirely new research questions
  - Cost and time efficient
  - Theoretically easily reproducible

Introduction to our working case

Why use the **R**?

- **Free** and open source
- Excels in **data visualization** and application of statistical methods
- Also: can be used to collect data on the Internet
- **Beginner friendly** for people with no programming background

➡ Can be used at **every stage** of our **workflow** (no need to switch programs)!

University of Zurich[UZH]

Department of Political Science

Data import in R

Data import in R

- You can save R objects (e.g. a dataframe) in in .Rda-files
- You use “load()” to import an .Rda file into R.

- Most datasets will not be prepared for R (e.g. .csv-files, Excel files, etc.) and we will learn in the next exercise more about the ways to recognize data formats and how to import them.

Outlook

- There are many packages suitable to load specific types of data into R:
  - **jsonlite**: for JSON data
  - **xml2**: for XML data
  - **readr**: for Text data
  - **haven**: for SPSS, SAS, Stata files
  - **readxl**: for Microsoft excel files (.xls or .xlsx)
  - **DBI**: for connections to data bases
  - **httr**: to retrieve data from APIs
  - **rvest**: to retrieve data from websites/html