KED2023 Assignment 1: Data Wrangling

Alex Flückiger | University of Lucerne

16 March 2023

Requirements

- Deadline: 23 March 2023, 23:59
- File format: executable shell script
- Naming schema: `SURNAME_KED2023_1.sh`
  Replace `SURNAME` with your surname.
- Use the shell template provided here.
- All tasks require shell commands unless stated otherwise.
- Submit your solutions on time via the respective exercise module on OLAT. The module is only open until midnight.
- Find solutions individually. When you are stuck, post your issue in the OLAT forum and ask friends. In terms of programming, Google may be your best friend.

Introduction

You learn how to perform basic shell commands and wrap them into a script to reproduce all steps.

Use a text editor to write your script (e.g., `Visual Studio`). You may want to try out the commands directly in your shell and, after successfully running them, copy them over into your script that you have opened in the text editor. Make sure that you include commands and comments only, while excluding the preamble like `Username@Computername$`. The command `history` shows the history of all used commands.

Follow this shell template when you write your script.

1 Organize your project

In this first task, you don't need to provide any interpretation, only the raw commands.

You set the structures of a new project in this task. As a project grows over time, it is crucial to organize your work properly. Otherwise, you get lost or waste too much time to find a particular file.

1. Create a new project folder with the following name:
   `KED2023_exercise_1`
2. Where did you create your project folder? In addition to the command, write the absolute path as a `# comment` into your script.
3. In the folder you have created, make the following subfolders:
   `reports`, `src`, `data`, `data/raw`, `data/interim`
4. In a project, you may have thousands of text files named inconsistently. To simulate this, create empty files with the following commands in the folder `data/raw`:

   ```
   touch data/raw/speeches_{2019..2023}_{a..z}.txt
   touch data/raw/text_{2019..2023}_{1..12}_{1..30}.txt
   ```

Don't forget to add these commands to your script.

5. Organize these files per year without modifying the original data directly. Thus, create folders for each year (2019-2023) in `data/interim`. Copy the created `.txt` files from above into the folder of the corresponding year. For example, a file with 2023 in its name goes into the directory named `2023`. Hint: Recall the expansion and wildcard operations.
6. Change into the main folder of this exercise `KED2023_exercise_1` by using the absolute path. Use `tree` to check if they are located in the correct folders with the correct name. When you are using macOS, you may need to install the program first with `brew install tree`.

Beyond this toy project, you may want to learn more about how to organize project. The cookie cutter website is a great resource that provides useful recommendation how to organize your data science project reasonably.

2 Report on file collection

In this second task, please give a short explanation accompanying your command.

What files do you have on your computer? Let's create some reports. You are free to choose the name of your output files. Yet, please recall the conventions that help others to understand the purpose of your scripts and outputs.

1. Navigate to the folder where you have saved most of your documents on the computer.[1] Print the path to this directory.
2. Use `ls` together with single and double asterisks to select all `.pdf` files in this folder, including its subfolders. Write the output directly into a new file using operators. For the correct use of asterisks, check this post on Stackoverflow. Use a single command only for the entire operation.
3. Write a single command to list all files in the current directory ordered by date, select the oldest file and write the output into a new file using a pipe and an operator.

3 Test your script

This task is a simple sanity check for your script. Your script has to pass this test, yet you don't need to include it in your submission.

Your deliverable has to be a runnable script comprising all the commands to accomplish the tasks above. To test your script, run the commands below. Once you call the script, it executes all commands, one after another. Everything should be reconstructed accordingly in the test folder. If not, correct the script so that it runs without any issue.

```
mkdir test_script
cd test_script
bash PATH_TO/SCRIPT.sh  # e.g. bash ../flueckiger_KED2023_1.sh
cd ..
rm -r test_script
```

4 Feedback

Please answer the following questions at the end of your script. Start your answers with the `#` symbol to mark them as comments that are ignored when running the script.

1. Do you have any questions concerning the exercise or the commands?
2. How long did it take to solve this exercise? Give a fair estimation.

[1] On Windows with Ubuntu installed, it should be located in `/mnt/c/Users/YOUR_USERNAME` (when you followed the installation guide, it can be accessed using the shortcut named `documents`. On macOS, it is located in `/home/YOUR_USERNAME`