Assignment #2 (summative)

Dr Thomas Robinson and Dr Dan de Kadt

AT 2023

Submission information

This assignment is due **Thursday 2 November 2023 at 4pm**:

- Please submit this assignment via the submission portal on Moodle
- You must submit your assignment as a *knitted, .html* file – .Rmd, .pdfs, or other file types will not be accepted

Please note, we will not contact you to recompile documents if they are submitted in the wrong format. It is your responsibility to ensure you submit your work correctly. Failure to do so will result in a mark of 0 for that assignment.

Introduction

This is a *summative* assignment, and will constitute 25% of your final grade. You should use feedback from seminars and your formative assessment to ensure you meet both the substantive and formatting standards for this module.

For clarity, the formatting requirements for each assignment are:

- You must present all results in full sentences, as you would in a report or academic piece of writing
    - If the exercise requires generating a table or figure, you should include at least one sentence introducing and explaining it. E.g. “The table below reports the counts of Wikipedia articles mentioning the LSE, by type of article.”
- Unless stated otherwise, all code used to answer the exercises should be included as a code appendix at the end of the script. This formatting can be achieved by following the guidance in the template.Rmd file (see Exercise 1).
- All code should be annotated with comments, to help the marker understand what you have done
- Your output should be replicable. Any result/table/figure that cannot be traced back to your code will not be marked

Exercise 1 (25 marks)

In this exercise you will simulate a version control workflow using git and GitHub. You should create a new public repository, and then use it to demonstrate the pipeline described below. This public repo should not be the same one you use to prepare this assessment. In your .html submission, your answer to this exercise should be a single sentence providing a clickable hyperlink URL for the public repository you create and manipulate (e.g. “My public repository can be found here (https://github.com/tsrobinson/firstrepository)).

We strongly recommend you read through the entirety of Exercise 1 carefully before beginning.

Pipeline

1. A developer creates a new GitHub repository, which contains an R script defining a function with a single argument `data`, that takes a dataset and performs some input transformation on it. This transformation can be as simple or complex as you like, but should work on at least one dataset (either a built-in R

dataset, or one provided by you in the public repository). The function should return the transformed data.
2. The same developer wishes to include new functionality by adding a second argument to *the same function*. Depending on the value passed to that second argument, the function should perform different data transformations. To preserve the integrity of the main branch, they add this functionality in a new branch called “dev”, which they do *not* immediately merge into main.
3. Having implemented this new functionality on the branch “dev”, the developer wants to demonstrate to users how it works. They create a new branch called “doc” *off the “dev” branch*, and add an RMarkdown file and knitted .html version that briefly shows what the function outputs depending on the value of this second argument.
4. Finally, once the developer is happy with these changes, they first use a pull request to merge the “doc” branch into “dev”, and then do the same to merge “dev” into “main”.

Hints:

- We will assess you partly by looking at the version control history – make sure that your commits and pull request descriptions are informative
- Marks will be available for concise, well-documented code
- You do not need to worry about the *number* of commits you make: you may make mistakes that need rectifying, or want to make multiple commits to achieve each stage of the above workflow. These actions are fine, so long as you document what each commit does (e.g. “fix issue in function documentation”).
- Before starting this exercise, think carefully about the files you will eventually need to make (and the structure of your repository). Make sure your final repository includes all files mentioned in the workflow and which are needed to run your code. You do not need to submit these separately as part of your submission on Moodle.

Exercise 2 (30 marks)

The OxCGRT study produced time-series data on governments’ policy responses to Covid-19, from 2020 to the end of 2022. You can find this data, including a codebook, here (https://github.com/OxCGRT/covid-policy-dataset) [https://github.com/OxCGRT/covid-policy-dataset/ (https://github.com/OxCGRT/covid-policy-dataset/)].

Using the `data/OxCGRT_compact_national_v1.csv` file from this repository, and the **ggplot2** package, you should generate a *single* visualization to help answer the following questions:

“To what extent did different regions of the world implement some form of recommendation or restriction for citizens to stay at home over the course of 2020-2022? How do the introduction of these restrictions compare to the regions’ implementation of income support over the same period?”

You should present your graphic (remember to introduce it in the text) and, in prose, answer the questions above by pointing to specific features of your visualisation.

Notes:

- By “single”, we mean something that can be generated by a single call to `plot()` (i.e. multiple facets in the same graphic are acceptable)
- You should read the documentation on the OxCGRT repository to find the relevant variables
- You may need to perform some data processing/manipulation/summarisation prior to visualising the data
    - In particular, you may want to simplify the scale of the stay at home directives and income support variables

Exercise 3 (45 marks)

Part 1 (15 marks)

Consider the following three string transformations (a-c) that when applied to some original string yield a transformed version (original -> transformed):

a. “apple” -> “pple” | “abacus” -> “bacus” | “Annapolis” -> “nnapolis”
b. “apple” -> “pple” | “abacus” -> “bcus” | “Annapolis” -> “Annpolis”
c. “C1_nat_a” -> “C_a” | “D2_state_g” -> “D_g” | “E_Loc_5_i” -> “E_i”

For each set of examples (a-c), write one function that would achieve these transformations **using regular expressions**. The input should be a string (character vector), and the function should return the transformed string (e.g. `my_fun("apple") -> "pple"`). Make your solution as general as possible, such that it would be robust to new cases that follow the same patterns as those given here.

Demonstrate each function by calling it on all three corresponding examples **and** one new example that follows the same pattern.

Report your code as part of this exercise (rather than just in the code appendix).

Hint: think about constructing the correct regex first, then consider what packages and functions might help replace the matched text.

Part 2 (30 marks)

The Gutenberg Project is an open library of over 70,000 books, accessible here (https://www.gutenberg.org/) [https://www.gutenberg.org/ (https://www.gutenberg.org/)]. In this exercise, you will come up with a research question that can be answered by counting the incidences of dictionary terms across a selection of books downloaded from this website.

First, develop a **simple** research question that should seek to compare how often a given concept is mentioned across a selection books. An example research question might be: “How often are house pets mentioned in the works of Jane Austen?” Briefly introduce your question and define your concept. Your concept must be measurable using a dictionary of words. For example, if our concept is house pets, we could measure whether they are mentioned by checking for incidences of “dog”,“cat”, and “budgerigar”.

Next, select and download at least two books as .txt files books from the Gutenberg Project that you will use to answer your research question. Your choice of books should be relevant to your research question.

Using the **quanteda** package, build a corpus from your selected books (each document should be the entire book) and a dictionary that encodes your concept. Then, use your dictionary to count the incidences of your concept across these books.

Finally, discuss your findings and provide an answer to your research question.

Notes:

- You do not need to perform a statistical test of any difference in counts
- Do not worry too much about the mapping of your concept to specific words. We are simply looking for something plausible (e.g., house pets -> “cat”, “dog”, and “budgerigar” even though there are many other kinds of house pets). We are primarily interested in assessing your ability to execute the data science task here, less the broader social science exercise.