Self-assessment #3

Sicheng Wang

Table of contents

1 Take-away

(1) Data can be obtained through manual download or programmatic download using functions such as `download.file(url = , destfile = )`.

(2) `untar(tarfile = , exdir = )` function can be used to uncompress a file.

(3) Use `tempfile()` to create a temporary file to avoid leaving the unzipped file on the disk.

(4) You can use `?` before a function name to check its usage.

(5) Control statements like `if` can be utilized to prevent repetitive downloads when running the code multiple times.

(6) `!` is used to negate an argument's value, changing it from `TRUE` to `FALSE` or vice versa.

(7) API, or Application Programming Interface, refers to resources that enable interaction with R code.

(8) The `unnest(cols = )` function from `tidyr` (Wickham, Vaughan, and Girlich 2024) can be used to expand variables that contain lists of variables within themselves.

(9) `dir_create()` is a function that can be used to create a directory.

(10) `write_csv()` function is used to save data frames as CSV files.

(11) In the `readr` package (Wickham, Hester, and Bryan 2024), `read_file()` reads the entire file, whereas `read_lines()` reads text line by line. Additionally, the `readtext()` function from the `readtext` package (Benoit and Obeng 2024) can read multiple files simultaneously or extract text content from PDF and DOCX files.

(12) `length()` function returns the number of lines in each file.

(13) The `head()` function is used to preview the first few lines of each file.

(14) The `tibble()` function from the `tibble` package (Müller and Wickham 2023) is used to create data frames.

(15) `pluck()` function from the `purrr` package (Wickham and Henry 2023) is used to extract the first line node.

(16) The `mutate()` function assigns values to columns.

(17) `str_remove()`, `str_replace()`, and `str_extract()` are used to edit matched values. `str_subset()` is used to isolate vector elements. `str_split()` is used to split vector elements. `str_detect()` combined with `filter()` can identify matched patterns to edit them. `str_trim()` is used to remove whitespace.

(18) `separate_wider_delim()` function is used to separate a column by its delimiter.

(19) `case_when()` function is used to change values according to different conditions, allowing for more than two conditions compared to `if`.

(20) `group_by()` function is used to group data by specific variables.

(21) The `dir_tree()` function creates a directory structure tree.

(22) The `unnest_tokens()` function converts variables into words.

(23) Setting `cache: true` allows results to be saved for later use, reducing the need for heavy traffic runs.

(24) `left_join()` keeps data in the left dataset and includes matching data from the right dataset, while `full_join()` retains all observations from both datasets when they share a common attribute.

(25) The `distinct()` function from the `dplyr` package (Wickham et al. 2023) is used to eliminate redundancy in data.

(26) `bind_rows()` is used to add rows from two datasets that share the same attributes.

(27) In regular expressions, `+` matches one or more occurrences, `*` matches zero or more occurrences, and `?` makes these two regular expressions match the shortest possible match.

(28) An idealized structure for the datasets can serve as a guideline during the curation and transformation of data.

(29) Documentation should encompass code comments, prose descriptions, information about data origin files (such as source, name, URL, credentials, etc.), and a data dictionary file describing variables (`create_data_dictionary()`).

(30) Acquisition involves obtaining data and storing it on a local computer, curation entails tidying the data into a tabular format while preserving the structure of the original data, and transformation involves customizing datasets to fit one's specific project requirements.

(31) Normalization involves extracting artifacts from datasets.

(32) Tokenization changes the units of observation.

2 Challenges

The most challenging aspect for me is practicing the entire process on my own. Watching others code is one thing, but coding by myself is another.

3 Resources

I primarily rely on the textbook and Recipes for guidance. Additionally, I consult ChatGPT when I encounter code I don't understand in the textbook. Another resource I use is Stack Overflow, where I often find answers or discussions about similar coding questions when searching on Google.

4 Explore

I believe I would benefit from having more time to explore these processes on my own, especially when working with real data. I've noticed that many things differ from what's presented in the textbook, and this presents an opportunity for me to practice applying the knowledge I've learned to real situations.

References

Benoit, Kenneth, and Adam Obeng. 2024. *Readtext: Import and Handling for Plain and Formatted Text Files*. https://github.com/quanteda/readtext.

Müller, Kirill, and Hadley Wickham. 2023. *Tibble: Simple Data Frames*. https://tibble.tidyverse.org/.

Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. 2023. *Dplyr: A Grammar of Data Manipulation*. https://dplyr.tidyverse.org.

Wickham, Hadley, and Lionel Henry. 2023. *Purrr: Functional Programming Tools*. https://purrr.tidyverse.org/.

Wickham, Hadley, Jim Hester, and Jennifer Bryan. 2024. *Readr: Read Rectangular Text Data*. https://readr.tidyverse.org.

Wickham, Hadley, Davis Vaughan, and Maximilian Girlich. 2024. *Tidyr: Tidy Messy Data*. https://tidyr.tidyverse.org.