Perfect Lifetime Effect: Report 1

Reproducible analysis reports with eye-tracking reading time data

YOUR NAME HERE

2023-05-05

Table of contents

Report overview **1**

Project structure (10 points) **2**

RProject and folders (6 points) . . . 2

README (4 points) . . . 3

Reproducibility (30 points) **4**

Data analysis (110 points) **4**

Report structure (12 points) . . . 4

Data set-up and wrangling (5 points) . . . 5

Dataset exploration (20 points) . . . 6

Reading times (32 points) . . . 6

Exploration (10 points) . . . 7

Summarise (22 points) . . . 7

Response data (31 points) . . . 8

Reaction times (14 points) . . . 9

Naturalness judgement responses (17 points) . . . 9

Summary (10 points) . . . 10

Bonus task: Raincloud plots . . . 10

Submit **10**

Report overview

There are three main aspects that this assignment will be graded on:

1) the project structure (10 points)
    - folder structure
    - project documentation (i.e, README file, see below)
2) reproducibility (30 points)
    - of analyses script
    - of the workflow
3) analyses (110 points)
    - data wrangling
    - data visualisation
    - interpretation of figures and tables

The first two topics (project structure and analyses) are graded independent of each other, but the third topic (reproducibility) takes the first two into consideration.

None of the tasks involve any code that was not already included in the materials I shared on Moodle (i.e., all functions etc. were used in the slides). Remember that you can also see my *source code* by clicking the `<> Code` button at the top of the html sheet versions of the course materials.

The whole assignment is worth 150 points. In my view this high number doesn't reflect the amount of *work* required, but rather the cumulative value of each line of code and the folder structure. There are many small steps required to build a reproducible work flow, and this number reflects that. For example, if you forget to use the `filter()` function in one exercise where it is required, you are deducted one point rather than getting the whole exercise wrong.

If you are stuck on a task, please feel free to set up a meeting with me.

The due date has been extended to *Wednesday, June 14th.*

Project structure (10 points)

For this assignment, you will be handing in a compressed (i.e., zipped) version of your course RProject.

RProject and folders (6 points)

As a first step, make sure your RProject is neat and orderly. It should contain:

- an `.RProj` file
- the folders

  - `data`, which contains the relevant datasets
  - `materials` or some other such name, which contains the course materials I shared
  - `my_notes` or some other such name, which contains your course notes
  - `reports` or some other such name, which contains the report you will submit for this assignment
  - `README.md`, which is a README file you will create for this assignment

This should look something like the folder below (without `renv` and `renv.lock`):

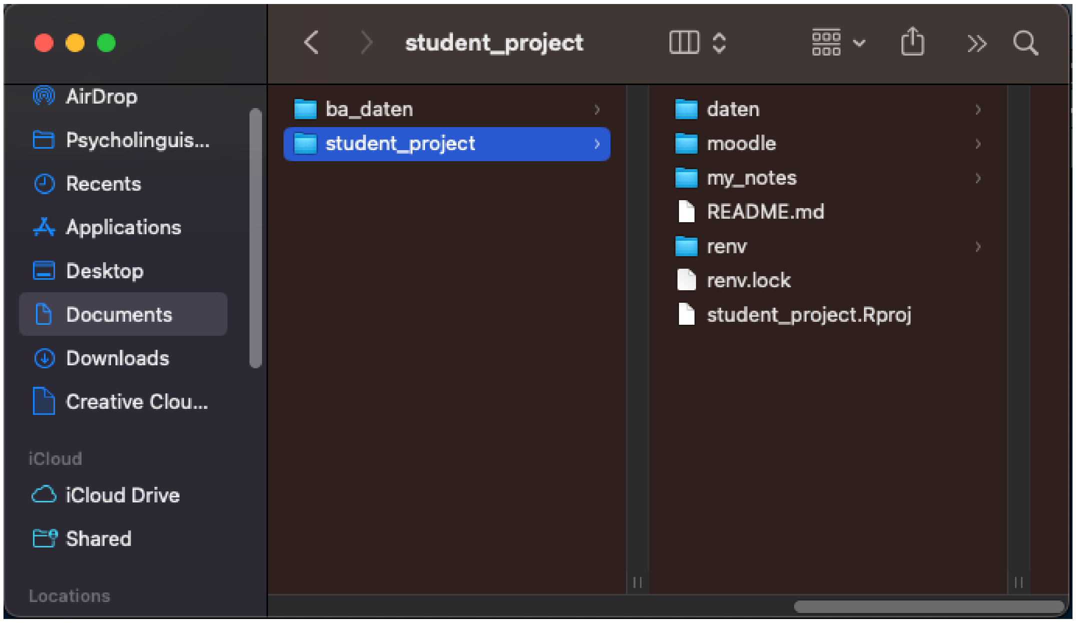


Figure 1: Example folder structure for an `.RProj`

Please make sure your report (this assignment) is structured using appropriate headings and subheadings.

README (4 points)

1. Create a `README` file (in RStudio: File > New File > Markdown File). This should contain a bit of information explaining the structure of the RProj folder and some information about the datasets. You can update it as the project grows (and should update it before submitting to make sure it's up-to-date). You can read this page for a short explanation and example of a README. At the very least, your README should contain:

    1. the project title
    2. your name
    3. a short description of the project (i.e., the course)
    4. a description of your folder structure and the files contained in the folders

Reproducibility (30 points)

To make sure your project is reproducible, but also reader-friendly, keep the following in mind:

- always restart R when you start a new session or new script (in RStudio: Tools > Restart R)
- render your document often to make sure it still runs as expected
- include a Session Info section at the end of your document, which runs `sessionInfo()`

Reproducibility is worth 30 points. If I can render your report *and* run through the source code chunk-by-chunk you get the full 30 points. If I cannot render the document, but can run all the code chunks then you will get half the points. If I can render the document but cannot run all the code chunks, you will be deducted up to half the points (depending on how much of the code will not run). If I can neither render the document nor run through the code chunks, you will not get these points. In this case, I will refer to the rendered document you send me (e.g., html or pdf) to evaluate the data analysis (as well as the source code).

Data analysis (110 points)

You will be creating a report written in Quarto (`.qmd`) that is quite similar to the exploratory analyses we ran on first-fixation times during class. You will be exploring the following measures:

- reading times
  - first-fixation duration
  - first-pass reading times
  - regression path duration
  - total reading times
- response data
  - reaction times
  - acceptances

Report structure (12 points)

Your report should have four main sections (highest-order headings) and subsections:

1) Data set-up and wrangling
2) Data exploration
3) Results

    - reading times
        - exploration
        - summary
    - response data
        - reaction times
        - naturalness judgements

4) Summary

Sub-headings should be identical to those in this document.

Your report should include (in the `YAML`):

- title of the assignment ("Perfect Lifetime Effect: Report 1")
- your name
- subtitle with the course title
- the date (you can try using `date: "2023-05-05"`)
- a table of contents
- the sections should be numbered
- if you are producing `html` documents, include the following under `html`:
    - `code-tools: true`
    - `self-contained: true`

Six points will be awarded for correct structure (headings, subheadings), and six points for the correct `YAML` settings.

Data set-up and wrangling (5 points)

1) Read in the `tidy_data_lifetime_pilot.csv` and, using pipes (`%>%`):
    - create a new variable `congruence` that contains the levels:
        - `+CON` when `lifetime` is `living`
        - `-CON` when `lifetime` is `dead`
    - `relocate` the variables `congruence`, `tense`, and `condition` to be after `eye`
    - include (i.e., `filter`)
        - all participants except `px3`
        - `type` is `critical`

Reminder: the 'codebook' on Moodle (Data > Documentation > data_description.pdf/.html) contains a table that defines each variable in our dataset. Refer to this if you need a reminder for how each variable is coded in the data.

Dataset exploration (20 points)

1) Print the first 10 lines of the data set, with only the following columns `selected` ***(2 points)***:

    - `px`, `trial`, `region_text`, `condition`, `ff`, `fp`, `rpd`, `tt`, `type`

2) Using R (i.e., don't simply count) find out how many participants had their right `eye` tracked and how many had their left `eye` tracked. Write the answer as a sentence. ***(3 points)***

2) Produce a table summarising reaction times per participant ***(5 points)***. Your code should:

    - keep a single `distinct` observation per participant per trial
    - `group` the data `by` participant
    - `summarise` the data by creating the variables
        - `N` which contains the number of observations
        - `mean` which contains the mean reaction time
        - `sd` which contains the standard deviation of reaction times
        - `min` which contains the minimum reaction time value
        - `max` which contains the maximum reaction time value
    - the table should include values rounded to one `digit`

Hint: we did something very similar in the Data Visualisation materials.

2) Do the same thing for `accuracy`. You don't need to include `min` or `max`. The table should round the values to 2 `digits`. ***(3 points)***

3) Add a few sentences under each table and briefly describe the results, e.g., what were the minimum and maximum mean reaction times and accuracies? How many observations were there per participant? How many participants were there? ***(5 points)***

Reading times (32 points)

You will be presenting and describing four reading times:

1. *first fixation time*
2. *first-pass reading time*
3. *regression path duration*
4. *total reading time*

The first two are considered "early" measures which include fixation durations during the first visit to a region of interest (the length of the first fixation (1), or the summation of all fixations within the region during the first visit (2)), the third is sometimes considered a "early" and sometimes a "late" measure (includes first-pass reading time plus the summation of all fixations to earlier regions and re-visits to the region of interest before continuing to the next region), while the fourth is considered a "late" measure that is the summation of all visits to the region of interest during a trial.

Exploration (10 points)

To explore the data, let's take a look at the distribution of observations.

1) Produce a boxplot by measure, per condition. Follow these steps:

   a) transform the data using `pivot_longer()` to create:
      - a singular column for `measure`, which contains the `names` for the variables the four reading time variables listed above
      - a single column for `time`, which contains the `values` of these four variables

   b) Generate a boxplot using this transformed data which plots

- the verb `region` only
- `condition` on the x-axis
- `time` on the y-axis
- boxes coloured by `condition`
- facet grids by `measure`
- add a theme to tidy up the plot (e.g., `theme_bw()`)
- add appropriate title and labels for the x-axis, y-axis, and legend

Summarise (22 points)

1) Print a table summarising the observations per reading time measure ***(6 points)***. The data summarised should:

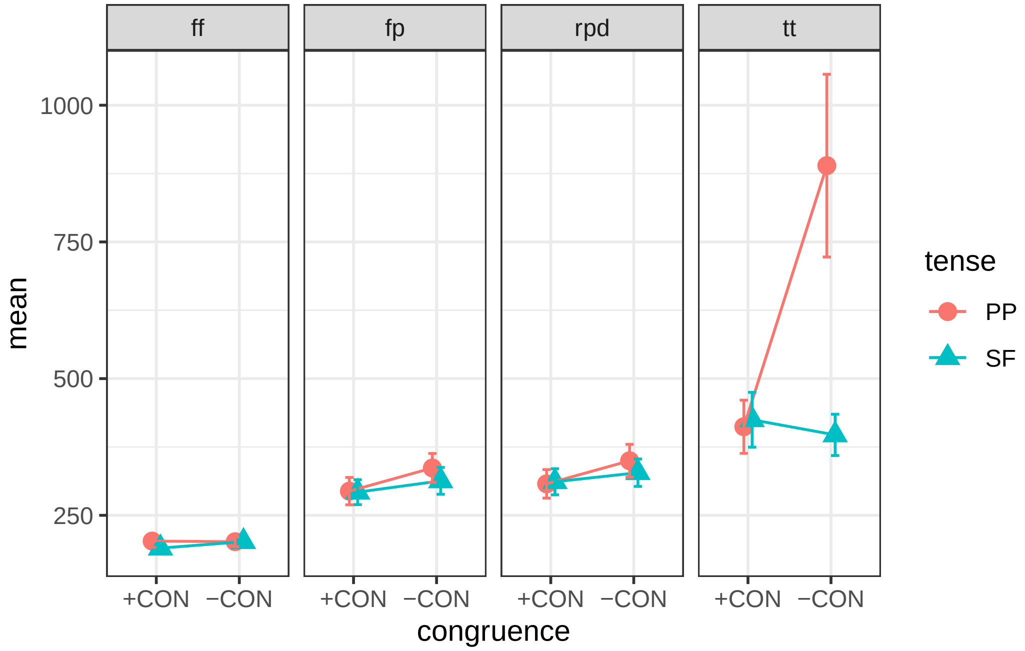
- contain data from the verb `region` only
- contain a `distinct` trial per participant
- `summarise` the number of observations, mean, standard deviation, minimum, and maximum observations per *reading time measure*

Hint: you will want to look into the `pivot_longer()` function, similar to the above exercise.

2) Reproduce the following interaction plot as best you can ***(8 points)***. Some things to consider:

- you will again want to use `pivot_longer()` to create `measure` and `time` variables

- filter the data to include only the verb region and only observations where `time` was larger than 0
- you'll need to summarise the data as we did before, but you also need to include e.g., `upper.ci` and `lower.ci` (consult the source code for the Data Visualisation materials)
- I've re-ordered the factor levels for `congruence`; we did this in Section 5 of the Data Visualisation materials



3) Summarise the results for each measure, comparing the two tenses (e.g., *The Present Perfect elicited longer/shorter first fixation durations than the Simple Future*), two levels of congruence (e.g., *Congruent conditions elicited longer/shorter first fixation durations than incongruent conditions*), and the overall conditions (e.g., *The incongruent condition elicited longer/shorter first fixation durations than the congruent condition in the Present Perfect*). ***(8 points)***

Response data (31 points)

Recall that in the experiment each trial ended with naturalness judgement task (was the last sentence natural given the preceding context? Yes or No). The button-press occured during critical sentence presentation. We have two measures corresponding to the naturalness response: reaction time (from onset of presentation of the critical sentence until the button press), and the naturalness acceptance (accept/reject). In the following tasks you will produce summary tables and plots of this data, and interpret the the data.

Reaction times (14 points)

3) Generate a table that summaries reaction times per condition ***(5 points)***. It should include:

    - a single observation per trial per participant
    - number of observations
    - mean acceptance rate per condition
    - standard deviation of acceptance rate
    - values rounded to 1 `digit`

3) Generate a plot that visualises reaction times per condition ***(5 points)***. Choose the plot type that you think would best visually represent this data type. Things to consider:

- what type of data is `rt`? Continuous, categorical, binary, etc.
- how should this data type best be visualised?
- what are some appropriate labels you can add?

4) Briefly describe the relationship between the conditions, referring to the values in the table and the plot you generated. ***(4 points)***

Naturalness judgement responses (17 points)

3) Generate a table that summaries acceptances per condition ***(5 points)***. It should include:

    - a single observation per trial per participant
    - number of observations
    - mean acceptance rate per condition
    - standard deviation of acceptance rate
    - values rounded to 2 `digits`

3) Generate a plot that visualises acceptances per condition. Choose the plot type that you think would best visually represent this data type. ***(8 points)*** Things to consider:

    - what type of data is `accept`? Continuous, categorical, binary, etc.
    - how should this data type best be visualised?
    - what are some appropriate labels you can add?

4) Briefly describe the relationship between the conditions, referring to the values in the table and the plot you generated. ***(4 points)***

Summary (10 points)

Summarise the findings from the reading times, reaction times, and acceptances. In which measures were there differences between conditions, in which were there no differences? What might we conclude based on these findings? Don't worry about being right or wrong, just try to interpret the data as best you can. ***(10 points)***

Bonus task: Raincloud plots

If you're feeling adventurous, try to generate three raincloud plots, one for first-pass reading time, one for regression path duration, and one for total reading times. I suggest looking into the package `ggrain` and its `geom_rain()` function. ***(8 bonus points)***

Submit

To submit your report, upload the following to Moodle:

- an output document of your rendered script (format of your choosing, e.g., PDF or html)
  - please save the file as **report1__LASTNAME__Firstname**.pdf/.html
- a compressed (i.e., *zipped*) version of the folder that contains your whole project