Reproducible analysis reports with eye-tracking reading time data

Summer Semester 2023

Daniela Palleschi

Invalid Date

Table of contents

1 Welcome/About

Welcome to the website for the course "Reproducible analysis reports with eye-tracking reading time data" for the Summer Semester 2023. Some quick info about the course:

- the language of instruction is English
- Block course:
  - April 12-14 (10am-4pm)
  - June 30th (2-6pm)
  - July 1st (10am-4pm)

Most documents are available as slides, html, and PDF on Moodle. Choose whichever you prefer (I suggest html).

1.1 Course description

- develop skills and know-how
  - create reproducible **reports & presentations** of eye-tracking reading data
  - common measures in **eye-tracking reading**
  - importance of **reproducible workflow**
  - **communicate** findings
- hands-on exercises in RStudio with the R programming language
  - data **wrangling** (`tidyverse`)
  - data **visualisation** (`ggplot2`),
  - descriptive and inferential **statistics** (`lme4` and `lmerTest`)

1.2 Course credits

- 4 LP
  - attendance and participation: 1LP
  - In-class exercises and preparation: 1LP
  - Assignments: 2 LP

1. Reproducible (pilot) analysis report + Pre-registration
2. Reproducible analysis report

2 Reading list

- this course does not have a heavy reading load, but a few readings are strongly recommended:
  - Open Science: (**kathawalla_easing_2021?**)
  - Eye-tracking reading: (**clifton_eye_2007?**); (**vasishth_what_2013?**);
  - A short recommendation for statistics for psycholinguists: (**vasishth_statistical_2016?**)
  - Statistics for Linguistics (textbook): (**winter_statistics_2019?**) (E-book available via Grimm)

2.1 Further readings

- there are lots of useful resources out there, specifically:
  - Bodo Winter's tutorials on linear (mixed) models (**winter_linear_2013?**; **winter_very_2014?**)
  - the PsyTeachR website is a *great* resource for hands-on stats and/or data analysis in R from the University of Glasgow School of Psychology and Neuroscience

Session Info

Save your session info at the end of each document. Our results very often depend on the version of R/RStudio/a package we used. This is a great first step towards creating a reproducible workflow!

```
R version 4.3.0 (2023-04-21)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.2.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  l

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.3.0  fastmap_1.1.1   cli_3.6.1       tools_4.3.0
 [5] htmltools_0.5.5 rstudioapi_0.14 yaml_2.3.7      rmarkdown_2.22
 [9] knitr_1.43      jsonlite_1.8.5  xfun_0.39       digest_0.6.31
[13] rlang_1.1.1     evaluate_0.21
```

References

3 Reproducible Analyses

What is it and why should I care?

3.1 Replication

> "There is increasing concern that in modern research, false findings may be the majority or even the vast majority of published research claims"
>
> – (**ioannidis_why_2005?**)

- replication refers to re-running a previous experiment with as few differences as possible
  - aim: determine whether the original results were *robust* and are *replicable*
  - if yes, great! the original findings are reliable
  - if no, hmm, maybe the original findings were false positives? or due to some other factor?
- in recent years, researchers have tried to *replicate* classic studies in their field
  - but in many cases, they did not get the same effects the original study reported (and were famous for)
- this began the ***replication crisis***

3.1.1 An example from language research

- (**nieuwland_large-scale_2018?**): a *direct* EEG[1] replication (versus *conceptual* replication)
- a multi-lab replication of (**delong_probabilistic_2005?**)'s impactful paper

[1] electroencephalography

  - (**delong_probabilistic_2005?**): reported N400 effects elicted at unexpected nouns, but also on preceding determiners (English *a/an*) when it signalled an unexpected word,
    * e.g., *The day was breezy so the boy went outside to fly…a kite/*an airplane*
    * taken as evidence of pre-activation of phonological form, graded by cloze probability
  - (**nieuwland_large-scale_2018?**): replicated N400 at noun, but not at adjective
    * i.e., *failure to replicate* a famous finding

3.2 Reproducibility

- reproducibility refers to the ability to *reproduce* somebody's analyses with their
  - data
  - *and* code
- it is not something we do once, nor is it something that will get us published
  - but it's important for open science and encourages transparency

3.2.1 Replication vs. Reproducibility

- **replication** of a study
  - repeating an **experiment**
  - getting *similar* results
- **reproducibility** of analyses
  - repeating **analyses** of the *same data*
  - getting the *same* results
- e.g., when you submit a paper to a journal, they make ask for your data and code so reviewers can *reproduce* your analyses
  - requires data and code
- if you have interesting findings, other researchers (or future you) may want to *replicate* your study to see if they can *replicate* your findings
  - (may require) stimuli, set-up and presentation information, participant demographics

3.3 Open Science: Why should I care?

1. Science is cumulative
    - We should ensure we're building on reliable, robust findings
    - i.e., it's *good* scientific practice
2. Because the field cares
    - replication/reproducibility are beginning to be foregrounded by e.g., journals/job advertisements
3. Helps future you
    - pre-registration, reproducible analyses, clean and shareable data: all help *future you*

3.3.1 What can I do?

- there's a variety of open science practices that we can choose to implement
- some suggestions from (**kathawalla_easing_2021?**):

Level: Easy

1. Journal Club
2. Project workflow
3. Pre-prints

Level: Medium

4. Reproducible code
5. Sharing data
6. Transparent manuscripts
7. Pre-registration

Level: Difficult

8. Registered reports

3.3.2 How to do better science

- don't be afraid of making mistakes
    - (most) researchers aren't statisticians or programmers
    - do the best you can, and ***be transparent***
- doing *some* of the steps is better than doing *none*

3.3.3 What will we learn here?

Design and Reporting

- Preregistration/Registered Reports
- Transparent writing

Analysis

- Reproducible code
  - with open source software (R, RStudio, packages)
  - dynamic reports with Quarto/Rmarkdown
- Project workflow
  - folder structure
    - how to sensibly set up your folders
  - contained environments
    - using RProjects and the `here` package

3.4 R is for Reproducibility

- we will be working with R, RStudio, Quarto, and RProjects
  - R: a programming language for statistical computing and graphics
  - RStudio: an integrated development environment (IDE)
    - RStudio Desktop
    - RStudio Server
  - Quarto (similar to Rmarkdown): dynamic reports
    - combining text, code, and printed tables and figures
  - RProjects: a workflow tool
    - contains all files necessary for a project
    - works with *relative* file paths

3.5 Exercises

3.5.1 RStudio

1. Open RStudio

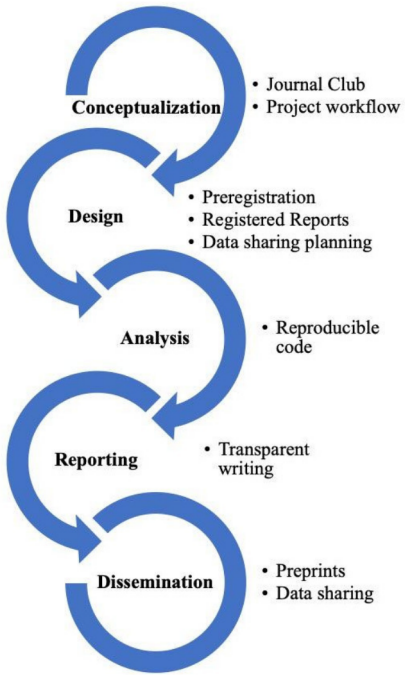

Figure 1. Open Science research practices across the research cycle

Figure 3.1: Image source: (**kathawalla_easing_2021?**) (all rights reserved)

- locate the Environment, Files, and Console panes
- File > New File > R script
- write `[your birth-month number]*[the your birth day]` and hit Enter
- write `print("Hello World!")`
- write `number <- 3*32`; this will create an object/variable 'number'
- write `string <- "Hello World!"`; this will create an object/variable 'string'
- write `number`
- write `string`
- add comments describing each step using `#`
- File > Save As

```
# multiply 5 by 7
5*7
```

```
[1] 35
```

```
# print some text
print("Hello World!")
```

```
[1] "Hello World!"
```

```
# save an object 'number' with 5*7
number <- 5*7
```

```
# save an object 'string' with text
string <- "Hello World!"
```

```
# print number
number
```

```
[1] 35
```

```
# print string
string
```

```
[1] "Hello World!"
```

```
# do math with objects
number+number
```

```
[1] 70
```

```
number*number
```

```
[1] 1225
```

```
number*2
```

```
[1] 70
```

```
month <- 5
```

```
day <- 7
```

```
month*day
```

```
[1] 35
```

3.5.1 Quarto[2]

- R scripts are a great way to keep track of what you did
  - however, the output is not saved, and adding comments with # gets kind of chunky
  - enter: dynamic reports!
- dynamic reports are those that combine text, code, and output
  - they are a great tool for communicating, collaborating, and documenting
  - they are also fantastic for note-taking
- Rmarkdown vs. Quarto

[2] https://r4ds.hadley.nz/quarto.html#workflow

  - both can combine text with code, outputting PDFs, Word Documents, html, or slides
  - main difference: Quarto has native support of a wider range of programming languages (e.g., Python and Julia)
- Want to know more? Check out Hadley Wickham's intro (**wickham_r_nodate?**)

3.5.1.1 YAML

```
---
title: "My title"
author: "My name"
format: html
---
```

- YAML is a human-readable programming language used to configure documents
- formatting is important: but be sandwiched between `---` and `---`
- in Quarto the output type must at least be given (with R: pdf, html, revealjs)

3.5.1.2 Headings and text

```
# This is a heading

This is text.

## This is a sub-heading

This is more text.
```

- headings are indicated by `#`
  - the number of `#`'s indicates the heading level

3.5.1.3 Code snippets

```
# do some math
year <- 1989
dog <- "Lola"
```

- sandwiched between `markdown```{r}` and `markdown```
  - shortcut: Ctrl/Cmd+Alt+I

3.5.1.4 In-line code

```
I was born on `r month`/`r day`/`r year`. My dog's name is `r dog`.
```

I was born on 5/7/1989. My dog's name is Lola.

- code output that was run *above* text can be called in-line using '`r `'

3.5.1.5 Altogether

````
---
title: "My title"
author: "My name"
format: html
---

# This is a heading

This is text.

## This is a sub-heading

This is more text.

Add some code chunks.

```{r}
# do some math
year <- 1989
dog <- "Lola"
```

And use call objects for in-line code: I was born on `r month`/`r day`/`r year`. My dog's
````

3.5.2 Quarto Exercises

3. Create a new Quarto document
    - File > New File > Quarto Document
    - Read the instructions
    - Practice running the chunks individually

- render the document
- verify that you can modify the code, re-run it, and see modified output

4. Create one new Quarto document for each of the three built-in formats: HTML, PDF and Word.
    - Render each of the three documents
    - How do the outputs differ?
    - How do the inputs differ?[3]

3.5.3 Quarto cont'd

- Choose a Quarto document:
    - give it a title, your name (author), and unclick 'Use visual markdown editor'
- Render
- YAML:

```
title: "Eye-tracking during reading"
subtitle: "Lecture 2 notes"
author: "[YOUR NAME HERE]"
lang: en
date: `r Sys.Date()`
```

- Render
- you can now try writing your class notes in this document (if you're brave)

[3] You may need to install LaTeX in order to build the PDF output — RStudio will prompt you if this is necessary.

References