

# Reproducible manuscript

The current folder (‘manuscript’) contains a reproducible manuscript named ‘manuscript.Rmd’ that is built using the R package ‘papaja’, Version 0.1.0.9997 (Aust & Barth, 2020), within the R Markdown framework (Xie et al., 2020). The APA format is enabled by the ‘apa7’ document class (Weiss, 2021).<sup>1</sup> The manuscript includes appendices, which are automatically knitted (i.e., rendered) and added into the manuscript when the latter is knitted. The output file is named ‘manuscript.pdf’.

The knitting—or compilation—was originally performed in RStudio 2021.09.1, Build 372.<sup>2</sup>

A LaTeX compiler is necessary to knit the document. A lightweight compiler called TinyTeX can be installed by running `install.packages('tinytex')` in the R console.

The manuscript can be knitted through either of the following commands. Command (a) is based on a custom function created in the folder ‘R\_functions’, and it is designed to avoid common knitting errors. Command (b) is a standard function of R Markdown.

- (a) `source('R_functions/knit_deleting_service_files.R'); knit_deleting_service_files()`
- (b) `rmarkdown::render('manuscript/manuscript.Rmd')`

A third option is using the ‘Knit’ button in RStudio.

## Methods applied to run the code

The results from code are entered in the manuscript in one of three ways, depending on the length of the code and the amount of running time required:

---

<sup>1</sup>For further information, the websites of these frameworks are as follows. **papaja**: <https://github.com/crsh/papaja>. **R Markdown**: <https://rmarkdown.rstudio.com/manuscript.html>. **apa7**: <https://mirror.ox.ac.uk/sites/ctan.org/macros/latex/contrib/apa7/apa7.pdf>.

<sup>2</sup>RStudio 2021.09.2+382 “Ghost Orchid” Release (fc9e217980ee9320126e33cdf334d4f4e105dc4f, 2022-01-04) for macOS. Mozilla/5.0 (Macintosh; Intel Mac OS X 12\_\_2\_\_1) AppleWebKit/537.36 (KHTML, like Gecko) QtWebEngine/5.12.10 Chrome/69.0.3497.128 Safari/537.36

1. Code present in the `.Rmd` file, and run as the manuscript is rendered: used for concise, fast code. Example from `manuscript.Rmd`:

```
maxVIF_semanticpriming = car::vif(semanticpriming_lmerTest) %>% max %>% ceiling
```

2. Code sourced from separate scripts, and run as the manuscript is rendered: used for very long code. Example from Appendix C:

```
source('semanticpriming/power_analysis/semanticpriming_all_powercurves.R')
```

3. Code run separately, with only the result being presented in the manuscript: used for very slow code. Example from Appendix B:

```
semanticpriming =  
  read.csv('semanticpriming/data/final_dataset/semanticpriming.csv')
```

Although the reproducibility of the code can be more immediately tested when the complete code is present in the `.Rmd` file, the code from the second and third methods can also be tested by accessing the appropriate code scripts (for further information, see `README.pdf` in the root directory).