Computational Reproducibility

Jürgen Schneider

Invalid Date

Table of contents

Preface

Here you'll find information on computational reproducibility.

Quarto Book

Create new project > "Quarto book"

- check "use renv with this project"
- check "create a git repository"

Create pages

Put this into .gitignore

/.quarto/ /_site/

Publish

First time, connect to github: renv::install("usethis") usethis::use_git_config(user.name = "Jane Doe", # <– change to your name user.email = "jane@example.org", # <– and your email init.defaultBranch = "main") # <– not necessary but kinder than 'master'

usethis::use_git()

usethis::create_github_token(description = "Token for project …") # <– Fill in a name of your token credentials::set_github_pat()

usethis::use_github()

Put this into the terminal:

quarto publish gh-pages

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Open file formats and software

Reproducibility in research relies on sharing both the steps used to analyze data and the data itself. Using open file formats and free, **open-source software** helps make these resources available to everyone. With software such as R and Python, researchers can document their data analysis procedures (in so-called scripts) that others can reuse or adapt.

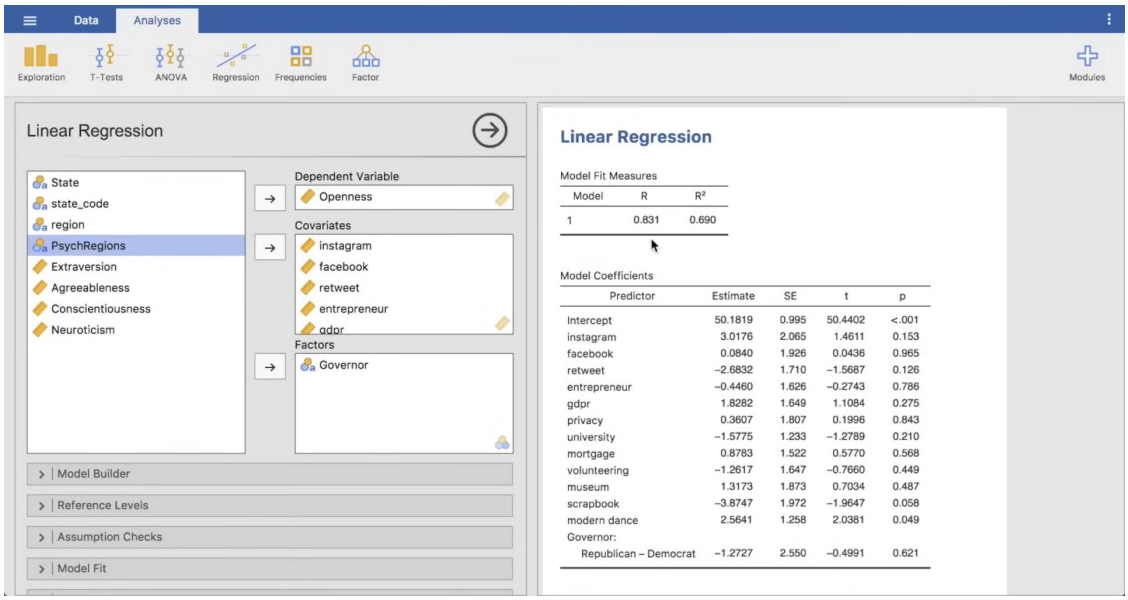
For those less familiar with programming, **point-and-click tools** like JASP, jamovi, and PSPP offer user-friendly alternatives. These free options remove financial barriers and make it easier for others to verify and build on your work. In contrast, proprietary software such as SPSS or MPlus restricts accessibility, limiting reproducibility to users who can afford expensive licenses.

Advantages

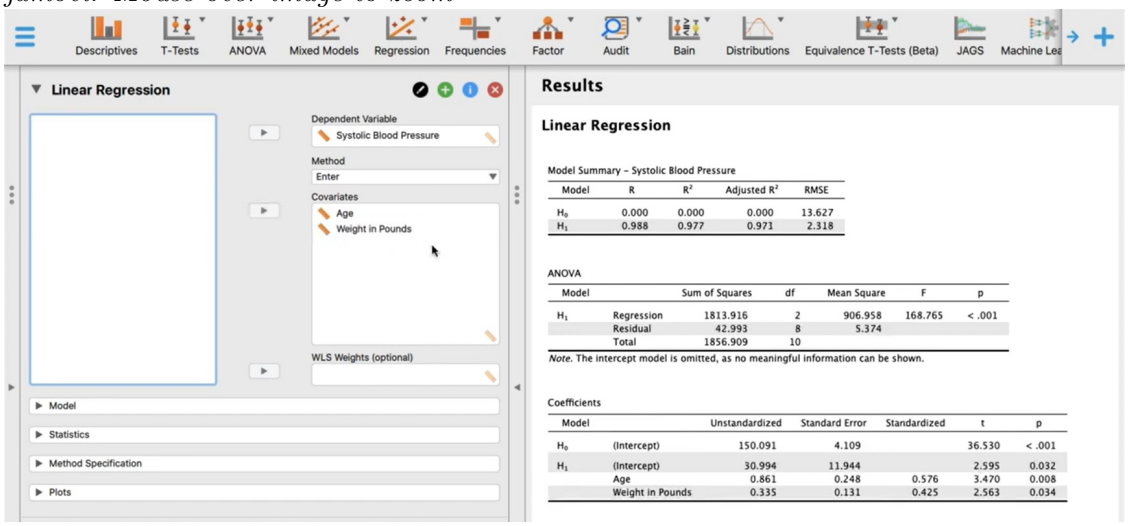
- Fundamentally enables reproducibility, as other researchers can open and run the analyses
- Open-source software lets researchers see how the software processes data when using its functions
- Promotes equity in access to research

First easy steps

Coming from SPSS



jamovi: Mouse over image to zoom



JASP: Mouse over image to zoom

In case your data is already saved and labelled in SPSS: **All** suggested **programs** can **read .sav files** and retain the labelling (such as variable and value labels)!

If you are used to **point-and-click user-interface**: jamovi and JASP are very similar to SPSS in look and feel. This means that there is usually no need for a longer period of familiarization with the program.

How to share:

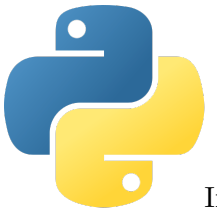
jamovi and JASP will automatically create project files that include everything including the data and output of your analyses. Therefore, after loading the data and computing analyses, you can simply share the .jasp (in JASP) or .omv (in jamovi) file with your colleagues or as a supplement to your publication. When others open this file, the data and all the calculated analyses will be available and ready to use in exactly the same way as with you.

Coming from MPlus, SAS or stata

Learning a new language such as R or python isn't necessarily easily done within a day, it is a **continuous learning process**. The good news is: There are **many free resources** for an introduction to the language (examples see below) and for solving concrete problems you will run into.
Getting help:

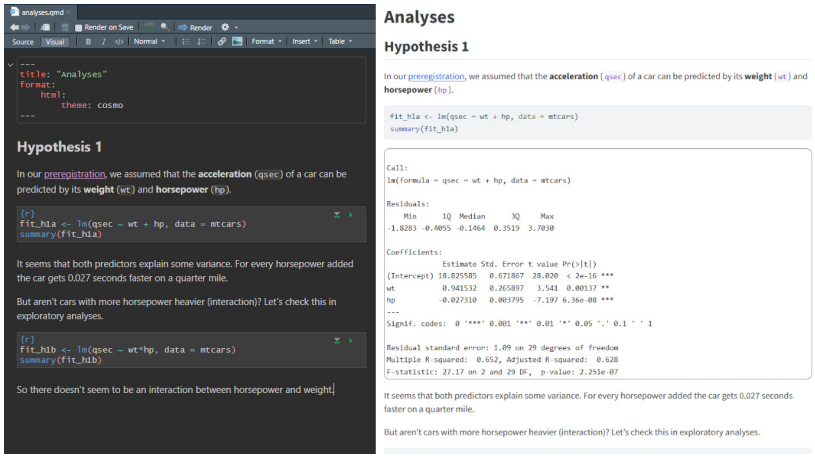
- You will usually find good documentation (or "help") of packages you use or vignettes (examples) of how they can be used. This is an example on how to call and use the R help.
- The Python and R communities are very active online. If you face a problem, chances are someone else has already asked about it. A great place to find solutions is stackoverflow.com.
- Chatbots like ChatGPT are also a fantastic resource for solving problems in your R or Python code. They are trained to provide explanations and suggest fixes to help you tackle your challenges.

Free Resources

 Intro to python(edX online course)  Intro to R(edX online course)

 Intro to JASP(youtube: JASP Statistics)  Intro to jamovi(youtube: freeCodeCamp.org)

3 Input-output-documents ("Notebooks")



Quarto Markdown in R: Mouse over image to zoom

Input-output documents integrate data analysis code (input) with corresponding results (output) in a **unified format**. Tools such as RMarkdown, Quarto Markdown, and Jupyter Notebooks (compatible with R or Python) enable the combination of code, results, and explanatory text—such as interpretations—into a single document exportable as HTML or PDF. Similarly, JASP and jamovi provide built-in functionality to achieve this integration within their platforms.

Advantages

- Provides provenance of results: Directly links transparent inputs to outputs
- Ensures error-free output: HTML/PDF documents are only rendered if all code, from data import to variable manipulation and analysis, runs without errors
- Enhances understandability: Enables detailed explanations of analytical approaches and result interpretations

Free Resources

 "Get started with Quarto"(youtube: Posit PBC)  Share & annotate JASP(youtube: JASP Statistics)

 Share jamovi(youtube: codecamp.org)  Annotate jamovi(youtube: Alexander Swan, Ph.D.)

4 Containerization and version management

R and Python use additional extensions ("packages") to expand their basic functions for data analysis. However, differences in the **versions** of these packages can create compatibility issues, even when researchers are using the same system. Packages like renv and groundhog for R help address this by maintaining consistent versions, ensuring that analyses remain accessible and reliable over time.

Beyond package management, maintaining compatibility across **system requirements**—such as operating systems and software versions—is critical for ensuring portability. Containerization tools like Docker and the Holepunch package for R provide solutions by encapsulating scripts and their dependencies into isolated environments. This approach preserves the original system configuration, enabling reproducibility of analysis scripts across different platforms. To what extent JASP and jamovi will be backwards compatible in the future is not entirely clear.

To tackle **both challenges** simultaneously, the rix package offers an integrated solution, combining system requirements management with package version control to streamline reproducibility efforts.

Advantages

- Data analysis is sustainable over longer period of time
- Equal system environment between researchers
- Equal package versions between researchers

Free Resources

Tutorial: Holepunch

Tutorial: renv

Tutorial: groundhog

Tutorial: rix 

video tutorial: rix(youtube: useR! Conference)

5 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.