gesis Leibniz Institute for the Social Sciences

Tools and Workflows for Reproducible Research in the Quantitative Social Sciences

Recap & Outlook

Johannes Breuer, Bernd Weiss, & Arnim Bleier

2022-11-18

Leibniz Association

Recap - Day 1

Time	Topic
10:00 - 11:00	**Introduction**
11:00 - 12:00	**Computer literacy for reproducible research**
12:00 - 13:00	Lunch Break
13:00 - 15:00	**Introduction to R Markdown**
15:00 - 15:15	Break
15:15 - 17:00	**Git & GitHub - Part 1**

Recap - Day 2

Time	Topic
09:30 - 10:30	Git & RStudio
10:30 - 10:45	Break
10:45 - 11:45	Jupyter Notebooks & Binder
11:45 - 12:30	Build your own Binder
12:30 - 13:30	Lunch Break
13:30 - 14:00	Sharing & Publishing
14:00 - 14:15	Break
14:15 - 15:45	Git & GitHub - Part 2
15:45 - 17:00	Recap & Outlook

Other topics in reproducible research

As we said in the introduction, we cannot cover all tools and topics related to reproducible research in this workshop. However, we want to use this session to cover some additional tools as well as other topics related to reproducible research:

- Other options for collaboration
- `R` package dependency management
- Workflow & project templates

Collaboration & different tool stacks

As we all know from experience, people have different knowledge and preferences regarding the use of research tools. The workflows we have focused on in this course require the use of `R`, `R Markdown`, and `Git`, as well as *RStudio* and *GitHub*. However, it is quite likely that we collaborate with others who do not use (all of) these tools.

Collaboration & different tool stacks

There also are `R` packages that you can use for collaborating on `R Markdown` documents with people who do not (want to) use `R Markdown` (and `Git`):

- `trackdown` uses *Google Docs* for this
- `redoc` "is a package to enable a two-way R Markdown-Microsoft Word workflow" (*note*: the project is currently "in suspended animation ")

trackdown

The basic workflow for `trackdown` is that you upload the content of an `.Rmd` file to *Google Docs* where you can collaboratively edit the text parts. You can then download the document again (e.g., to edit the code in the `R Markdown` document in *RStudio*), and also update the file on *Google Docs* after changing the `.Rmd` locally. The `trackdown` documentation provides further details.

Advanced use of `trackdown` with `Git`

To combine `R Markdown` with collaborative text editing via `trackdown` and version control (and to avoid potential issues caused by - possibly unintended - changes to the code parts on *Google Docs*), the author of the `trackdown` package, Claudio Zandonella Callegher, proposes a solution in an issue in the *GitHub* repository for the package.

Essentially, the idea here is to create a `trackdown` branch in the `Git` repository and merge it with the `main` branch which is (mainly) used to edit the code.

Dependency management

The R world is highly dynamic and networked. This means that...

- R packages change (sometimes in ways that cause changes in results or even breaking code)
- R packages typically depend on other packages (which can also change)

Eventually, this can lead to a problem often described as "code rot", meaning that code ceases to function (or leads to different results).

Dependency management

Dependency management

- Our projects may use/require different package versions
- Manually managing dependencies is a nightmare
  - Keeping track of the dependencies and their changes
  - Restoring the `R` environment

Dependency management

In the previous sessions, we got to know *Binder*, which not only allows us to share our results and code with the world in an interactive manner, but also addresses the issue of code rot and dependency management (by sharing complete pre-specified `R` environments).

However, for various reasons, using *Binder* might not be an option. Luckily, there are also other (more lightweight) options for handling dependency management and preventing code rot when working with `R`.

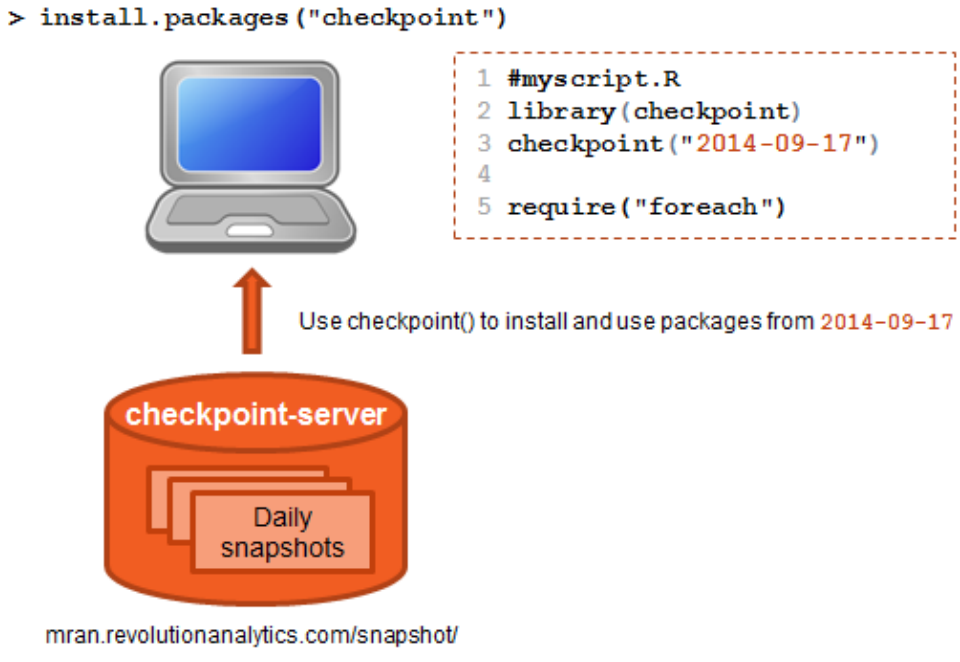
Solutions for `R`

1. `checkpoint` by *Microsoft*
    - Requires a project-based workflow
    - Package database will gradually grow
2. `groundhog`
    - Package database will gradually grow
3. `renv` by *RStudio*
    - Most flexible and powerful
    - Least straightforward to use

Note: To ensure that everything works as on your machine, the packages need to be the same (version) as in the `checkpoint`/`groundhog` library. So, to be extra sure about this, you may need to update your local package installations accordingly.

checkpoint

`checkpoint` is part of the Reproducible `R` Toolkit from *Microsoft*



https://cran.r-project.org/web/packages/checkpoint/vignettes/checkpoint.html

checkpoint

Dependencies are detected automatically

```
library("checkpoint")
checkpoint("2022-11-11")

library("ggplot2")
```

Uses a date-specific directory outside of the usual R package directory. By default, this is:

```
~/.checkpoint/...
```

checkpoint

To further increase reproducibility, it is also possible to specify an R version.

```
checkpoint("2022-11-11", r_version = "4.1.3")
```

Dependency management - one step further

Going beyond `R` packages (and potentially also the `R` version), a more holistic (albeit also technically demanding) approach to dependency management (and combating code rot) is the use of containers via *Docker*. If you want to get into this, you can have a look at the Introduction to Docker for R Users by Colin Fay or the mini-tutorial Enough Docker to be Dangerous by Sean Kross. There are different prespecified *Docker* files for `R` users available via *rocker*, and there also is a Docker workflow to work reproducibly with papaja.

Note: Remember that *Binder* also makes use of *Docker*.

Project setup and templates

In this workshop, we have shown you how to manually set up a reproducible research workflows. However, there are some tools that you can use to automate parts of this process. These can range from very simple to very elaborate solutions.

Project setup and templates

We have already seen and tested the file `create-project.sh` (which is small shell script for initializing a basic project folder structure that can be easily adapted and extended using any text editor). However, there also are several other (more complex) packages and templates that can be used for the creation and maintenance of reproducible research workflows, such as...

- `template`
- `WORCS` - *Workflow for Open Reproducible Code in Science*
- `workflowr`
- `starter`
- `rrtools` - Tools for Writing Reproducible Research in R

start your lab also provides an `R` Project Template.

Shoulders of giants... but sometimes also clay feet

As you may have already experienced, not all tools always play together nicely. Keep in mind, that most tools that we have covered in this workshop are free and open source software (FOSS). Also, tool stacks can have break points and many tools themselves depend on other tools/tool stacks. Hence, things may not always work perfectly.

But don't despair! There usually are solutions (*Stack Overflow* and issues in associated *GitHub* repositories are good places to find them) and the advantage of FOSS is that there usually is an active development community that you can also get involved in.

Showing appreciation 

The creation and maintenance of FOSS takes a lot of time and this is rarely recognized as much as it should be. One thing we can do to change this is to at least give credit where credit is due and cite the tools and resources that we use.

Showing appreciation 

```
citation("trackdown")
```

```
##
## To cite trackdown in publications use:
##
##   Emily Kothe, Claudio Zandonella Callegher, Filippo Gambarota, Janosch
##   Linkersdörfer and Mathew Ling (2021). trackdown: Collaborative Writing and Editing
##   of R Markdown (or Sweave) Documents in Google Drive.
##   https://doi.org/10.5281/zenodo.5167319
##
## Ein BibTeX-Eintrag für LaTeX-Benutzer ist
##
##   @Manual{,
##     title = {trackdown: Collaborative Writing and Editing of R Markdown (or Sweave) Documents in Google Drive},
##     author = {Emily Kothe and Claudio Zandonella Callegher and Filippo Gambarota and Janosch Linkersdörfer and
##     year = {2021},
##     note = {R package version 1.1.1},
##     url = {https://github.com/claudiozandonella/trackdown},
##     doi = {10.5281/zenodo.5167319},
##   }
```

FOSS is boss

"Open source is a hard requirement for reproducibility" (Bruno Rodrigues)

Looking back

You created a *GitHub* repository containing materials for a fully reproducible research pipeline! 

If you created a public *GitHub* repository: Head over to http://starlogs.net/ and paste the URL of the repository to recap your heroic journey into the universe of reproducible research! 

Any other questions or things you want to say/discuss?

Looking forward

We hope that we could get you started or help you with with making your research (more) reproducible. Of course, as always, there is much more to explore and learn. The only way to really get familiar with the tools and workflows is if you use them for your own research.

And remember that making your research (more) reproducible is an incremental process. Every step towards reproducibility is an improvement 👣. You don't have to (and probably should not) leap to a full `R Markdown` + `Git` + `Docker` workflow all at once.

Keep calm and stay reproducible! 😊

Thank you very much for participating in this workshop!

We hope that you learned something and also had some fun (at least a little bit...)