# GESIS Fall Seminar in Computational Social Science 2022

Syllabus for week 1:

# "Tools for Efficient Workflows, Smooth Collaboration and Optimized Research Outputs"


| | | |
|---|---|---|
| Lecturers: | Dr. Julia Schulte-Cloos | Lukas Lehner |
| Affiliation: | University of Munich (LMU) | University of Oxford |
| Email: | julia.schulte-cloos@gsi.lmu.de | lukas.lehner@spi.ox.ac.uk |


Date: September 05-09, 2022
Time: 10:00-17:00

## About the Lecturers

Julia Schulte-Cloos is a Marie Skłodowska-Curie funded research fellow at the University of Munich (LMU). She has earned her PhD in Political Science from the European University Institute. Julia is passionate about developing tools for generating reproducible workflows and research outputs with R Markdown. Reflecting her commitment to Open Science, she is also a member of the board of the Open Science Center at LMU Munich and a catalyst of the Berkeley Initiative for Transparency in the Social Sciences (BITSS).

Lukas Lehner is a PhD candidate at the University of Oxford, Department of Social Policy and Intervention, and at the Institute for New Economic Thinking. His research interest is labor market policy using experimental, quasi-experimental, and comparative methods. He conducts a randomized control trial on a pilot job guarantee scheme and has founded the Oxford Supertracker: The Global Directory of COVID Policy Trackers and Surveys. Previously, Lukas worked as a Junior Economist at the OECD and at the International Labour Organization.

## Course Description

How can we create efficient workflows and facilitate optimal collaboration in teams? How can we ensure that our research processes, our data collection, and our complex (Big) Data analyses can be re-traced by ourselves and other researchers, both in the near and distant future? How can we build our analyses in a way that they can be run reliably and stably on other researchers' computers, regardless of the hardware and software environment? Efficient and reproducible workflows are essential to keep up with the increasing amount of data and complexity of analyses. In recent years, exciting new tools have emerged that enable effective data management and research collaboration. Not only do these new tools help us streamline our workflows, but they also make our research outputs more visible, citable, and sustainable. From the first day we begin adapting our research practices, we benefit from greater efficiency, ease of tracing our research progress, and smoother collaboration with other researchers. In the long run, these practices help us maintain a high quality of research outputs and meet the replicability and transparency standards that an increasing number of journals require for publication. This course provides participants with the skills to harness the potential of new tools that help create efficient workflows and optimize research outputs. It equips them with a toolkit to conduct research that is well organized and documented, and can be readily disseminated and reproduced, both when working on independent projects and in collaborations with others.

## Keywords

Research Workflows, Collaboration; Version Control; Reproducibility; Data Management

## Course Prerequisites

- Basic knowledge of a statistical programming language such as R or a general-purpose programming language such as Python.

## Target Group

Participants will find the course useful if:

- Researchers at any stage of their career, who rely on data-driven approaches in their work.

## Course and Learning Objectives

By the end of the course participants will:

- confidently master tools that enable efficient workflows and collaboration;
- be able to write executable code and create automatable reports using RMarkdown, Pandoc, and Lua;
- be able to collaborate effectively with other researchers and document work processes with version control through Git and DVC;
- have an in-depth understanding of key Git operations, including branching, merging, forking, resolving merge conflicts;
- rely on Veracrypt for advanced data protection and encryption;
- be able to effectively disseminate their findings online, e.g. on their own academic website created using GitHub Pages, Hugo, and Blogdown;
- successfully containerize their projects using Docker and Binder;
- understand how to ensure interoperability of programming languages when generating reports;
- be able to rely on the command line and shell scripts for advanced programming and to solve tricky computational issues.

## Organizational Structure of the Course

This course is a one-week full-time program designed to turn participants into experts in modern approaches to workflow management. Participants are expected to do some essential preparatory reading and install the required software before attending the course. All necessary instructions and tutorials will be provided in advance. The seminars consist of lectures, laboratory exercises, and group exercises. In the lab sessions, participants work on practical exercises and complete tasks both individually and in small groups while the lecturers assist them. This allows participants with different levels of prior knowledge to acquire new skills and progress at their own pace. The instructors are also available for one-on-one meetings to clarify questions and give advice on participants' projects.

## Software and Hardware Requirements

Participants should bring their own laptops. This course is based on open-source programming languages and software environments and supports the principles of 'Open Data', 'Open Code' and the integration of narrative text and code. We will use a variety of software and tools, such as:

- R and R Studio;
- TinyTex and Pandoc;
- Git, GitHub, and GitHub Desktop;
- Veracrypt;
- Docker Desktop.

Participants will also need to register for free online accounts with GitHub and DockerHub. Course participants will receive detailed instructions on the required software, packages and how to install them in sufficient time prior to the start of the course.

## Recommended Literature to Look at in Advance

Suggested literature for those unfamiliar with R or who would like to get a refresher by skimming through:

- Venables, Smith and the R Core Team: *An Introduction to R*. https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
- *RStudio Cheat Sheets*. https://www.rstudio.com/resources/cheatsheets/
- Wickham: *Advanced R*. https://adv-r.hadley.nz/

Suggested literature for those unfamiliar with Markdown:
*Markdown Guide*. https://www.markdownguide.org/

## Day-to-day Schedule and Literature

**Day 1: Automatable reports**

- In the lectures, we will cover foundational concepts of efficient workflows and collaboration, which will be implemented in the days to follow.
- In the lab exercises, participants will write executable code and create automatable reports using R Markdown, Pandoc, and Lua. Participants will learn how to ensure interoperability of programming languages when generating reports.

*Compulsory reading:*

- Trisovic, A., Lau, M.K., Pasquier, T. et al. 2022: A large-scale study on research code quality and execution. *Sci Data* 9, 60. https://doi.org/10.1038/s41597-022-01143-6
- Nosek et al. 2015. Promoting an open research culture: Author guidelines for journals could help to promote transparency, openness, and reproducibility. *Science* 348.6242. https://dx.doi.org/10.1126%2Fscience.aab2374

*Suggested reading:*

- Schulte-Cloos. 2021. *ReproducR: Drafting and publishing reproducible scientific articles with R Markdown.* R package version 0.1.0, https://github.com/jschultecloos/reproducr
- Blumenau. *Reproduce_me. Reproducible Papers with R Markdown*. https://github.com/jblumenau/reproduce_me
- *Quarto Guide*. https://quarto.org/docs/guide/
- Piper. 2020. Science has been in a "replication crisis" for a decade. Have we learned anything? *Vox*. https://www.vox.com/future-perfect/21504366/science-replication-crisis-peer-review-statistics

**Day 2: Version control**

- In the lectures, we will cover the conceptional underpinning of modern version control tools such as Git and DVC.
- In the lab exercises, participants will work in teams to collaborate on a small coding challenge, familiarizing themselves with the Git version control system and key Git operations, including branching, merging, forking, and resolving merge conflicts. In doing so, participants will learn to integrate code review and merge request into their day-to-day coding workflow.

*Compulsory reading:*

- Bryan. 2018. Excuse Me, Do You Have a Moment to Talk About Version Control? *The American Statistician*, 72(1), 20-27.
- Turner et al. 2020. *Open Code and Software: a Primer from UKRN*. https://doi.org/10.31219/osf.io/qw9ck

*Suggested reading:*

- *Git and GitHub for R*. https://happygitwithr.com/
- Brailey. 2022. *Using GitHub and Rstudio*. https://osf.io/dgw2s/

- Chacon and Straub. 2014. *Pro Git*. Apress. https://git-scm.com/book/en/v2

**Day 3: Dissemination and academic websites**

- In the lectures, we will discuss how to combine executable reports (day 1) with version control (day 2) to effectively disseminate our findings online.
- In the lab exercises, participants will use GitHub Pages, Hugo, and Blogdown to create their own academic website.

*Compulsory reading:*

- Visconti. 2016. Building a static website with Jekyll and GitHub Pages. *Programming Historian*. https://doi.org/10.46430/phen0048
- Williams. 2020. *Building an academic Website*. https://jayrobwilliams.com/posts/2020/06/academic-website/

*Suggested reading:*

- The Carpentries: *Building Websites with Jekyll and GitHub*. https://carpentries-incubator.github.io/jekyll-pages-novice/
- Turner et al. 2020. *Open Code and Software: a Primer from UKRN*. https://doi.org/10.31219/osf.io/qw9ck
- Spitschan et al. 2020. *Preprints: a Primer from UKRN*. https://doi.org/10.31219/osf.io/m4zyh

**Day 4: Containerisation for reproducible environments**

- In the lectures, we will cover the key idea underlying containerization to bundle software, libraries and configuration files with a particular focus on recent advances in academia to ship fully reproducible virtual environments as a part of scientific replication packages.
- In the lab exercises, participants will use Docker and Binder to containerize their projects to ensure full software and code reproducibility. Participants will learn how to write a Dockerfile that is seamlessly integrated with the most important R packages, pin their specific library versions and prepare their containers for dissemination.

*Compulsory reading:*

- Lopp. 2019. *Reproducible Environments. R Views*. https://rviews.rstudio.com/2019/04/22/reproducible-environments/
- Boettiger, Carl, 2015. An introduction to Docker for reproducible research. *Operating systems review*, 49(1), pp.71–79.
- Sangole. 2021. *Reproducible Work in R*. https://towardsdatascience.com/reproducible-work-in-r-e7d160d5d198

*Suggested reading:*

- R-bloggers. 2021. *Setting up a transparent reproducible R environment with Docker + renv*. https://www.r-bloggers.com/2021/08/setting-up-a-transparent-reproducible-r-environment-with-docker-renv/
- Ihle, Jaquiery, Robinson, Gibson, George. 2021. *Community call: reproducible environment*. Reproducible Research Oxford. https://osf.io/xp9zn/
- Nüst D, Sochat V, Marwick B, Eglen SJ, Head T, Hirst T, et al. 2020- Ten simple rules for writing Dockerfiles for reproducible data science. *PLoS Comput Biol* 16(11): e1008316. https://doi.org/10.1371/journal.pcbi.1008316

**Day 5: Encryption and advanced programming**

- In the lectures, we will discuss modern encryption methods and advanced programming to solve tricky computational issues.
- In the lab exercises, participants will use Veracrypt for advanced data protection and encryption. Participants will also learn how to rely on the command line and write short shell scripts to advance their workflow.

*Compulsory reading:*

- Rahal. 2017. *An Introduction to the Command Line.* https://crahal.github.io/teaching/AnIntroductionToTheCommandLine
- *VeraCrypt. Documentation*. https://www.veracrypt.fr/en/Documentation.html

*Suggested reading:*

- The Carpentries: *Extra Unix Shell Material*. https://carpentries-incubator.github.io/shell-extras/
- The Carpentries: *Introduction to the Command Line for Metagenomics*. https://carpentries-incubator.github.io/shell-metagenomics/

## Additional Recommended Literature

- Christensen. 2018. *Manual of Best Practices in Transparent Social Science Research*. https://github.com/garretchristensen/bestpracticesmanual
- Miguel, Camerer, Casey, Cohen, Esterling, Gerber, Glennerster, et al. (2014). Promoting Transparency in Social Science Research. *Science* 343 (6166). 30–31. https://doi.org/10.1126/science.1245317
- Schulte-Cloos, Gonzalez-Torres, Belot. 2018. *PhD Toolkit on Research Transparency and Reproducibility*. European University Institute, Florence. https://osf.io/ygdn9/
- *Berkeley Initiative for Transparency in the Social Sciences (BITSS)*. https://www.bitss.org/
- *The Carpentries Lessons*. https://carpentries.org/community-lessons/
- Rumsey, S., Lunny, C., & Kennedy, B. J. (2020). *Open Access: a Primer from UKRN*. https://doi.org/10.31219/osf.io/94rsp
- Towse, J. N., Rumsey, S., Owen, N., Langford, P., Jaquiery, M., & Bolibaugh, C. (2020). *Data Sharing: a Primer from UKRN*. https://doi.org/10.31219/osf.io/wp4zu
- Danchev. 2021. *Reproducible Data Science with Open-Source Python Tools and Real-World Data.* https://valdanchev.github.io/reproducible-data-science-python/intro.html#