Modern academic publishing with Quarto, R, GitHub, and friends – A publisher's view

Gunther Maier
Modul University
REGION Technical Director
Vienna, Austria

Introduction: “research process”

- Problematic transitions between the steps
  - Bring data into machine readable form
  - Select additional data
  - Transfer results of the analysis into the paper
- Typically, these transitions are done “by hand”
- Some changes may be forgotten or overlooked
- Can the reader of the paper trust the reported results?

Introduction: The reproducibility crisis

- In 2005, Ioannidis argued that “most published research findings are false”
- Issues of reproducibility (same data, same method) and replicability (different data, same method)
- Substantial evidence of unreproducible published research across disciplines (Psychology, Medicine, Pharmacology, …)
  - In Economics, about 39% of experimental studies published in AER and QJE could not be reproduced.
- Many reasons:
  - Incomplete and/or imprecise documentation of the research process
  - Publication bias (a negative result is a result, but difficult to publish)
  - Hunting for significance – publish or perish pressure


Open access, freely available online

Essay

Why Most Published Research Findings Are False

John P. A. Ioannidis

Summary

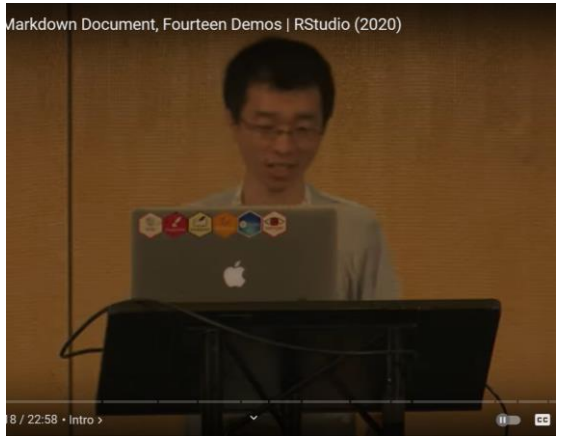
It can be proven that most claimed research findings are false.

Can we trust published research results?

- Reputation problem for research, researchers, and research institutions
- Quarto, R, GitHub and friends are a major step toward a solution
- They do not solve all these problems, but many
- They can contribute to more reproducible research
- They streamline the research process and contribute to open science

What is Quarto?

- Quarto is a system for academic and technical publishing
- uses text files (*.**qmd**)
- Is free and open source and uses only free and open source components
- A simpler and more advanced version of RMarkdown (integrates a range of packages)
- Uses existing and well tested tools (knitr, pandoc, LaTeX, BibTeX)
- One input document (qmd-file) can produce a wide range of output documents (HTML, pdf, Word, Powerpoint, Beamer, …) – Article manuscript, book manuscript, report, website, blogsite, interactive page, …



Yihui Xie | One R Markdown Document, Fourteen Demos | RStudio (2020)
https://posit.co/resources/videos/one-r-markdown-document-fourteen-demos/

Elements of a Quarto document

```
---
title: "Webr in Quarto“
format: html
---
```

1. A YAML header
   - Defines the output format, styling, parameters, extensions, templates
   - Special structure is required
2. A Markdown text
   - Text of the document with embedded logical structure
   - No formatting
3. Embedded code chunks
   - R (Python, Julia, …) code to be executed
   - Runs in its own instance
   - Many chunk options
4. Inline code
   - E.g.; `r mean(data)`

```
## Webr

Webr is a standard that lets you run R in a *web-browser*. To develop a webpage using Webr in Quarto, you need to load the quarto-webr extension.

You do this with the command `quarto add coatless/quarto-webr` in the **Terminal window** of RStudio.

## Activating Webr in Quarto
```

````
```{r}
data <- rnorm(20, mean=0, sd=1)
mean(data)
```
````

````
```{r}
plot(data)
```
````

A simple example

````
---
title: "ggplot2 demo"
author: "Norah Jones"
date: "5/22/2021"
format:
  html:
    code-fold: true
---

## Air Quality

@fig-airquality further explores the impact of temperature on ozone level.

```{r}
#| label: fig-airquality
#| fig-cap: "Temperature and ozone level."
#| warning: false

library(ggplot2)

ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess")
```
````

ggplot2 demo

Norah Jones

May 22nd, 2021

Air Quality

Figure 1 further explores the impact of temperature on ozone level.

▶ Code

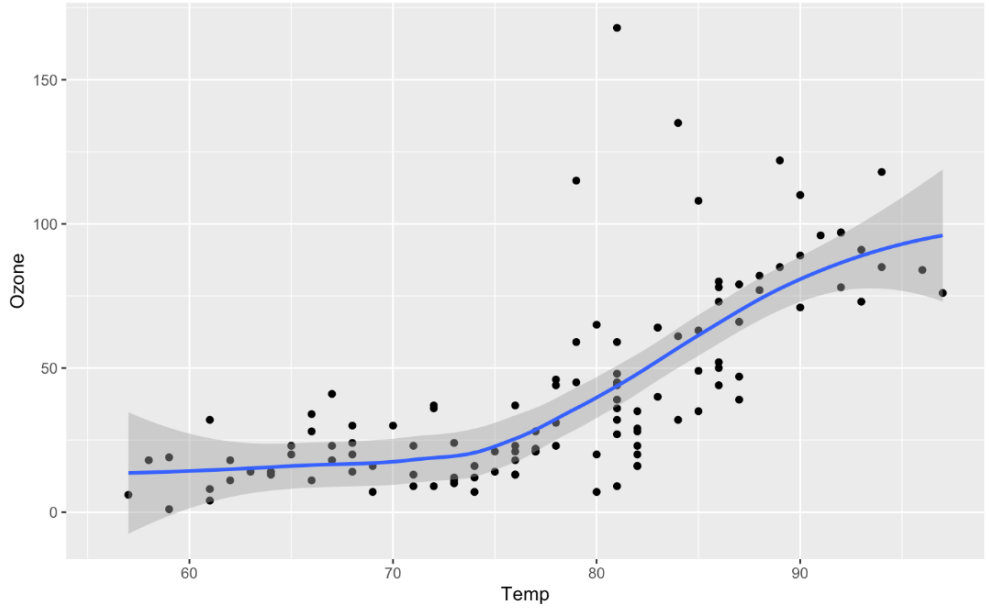
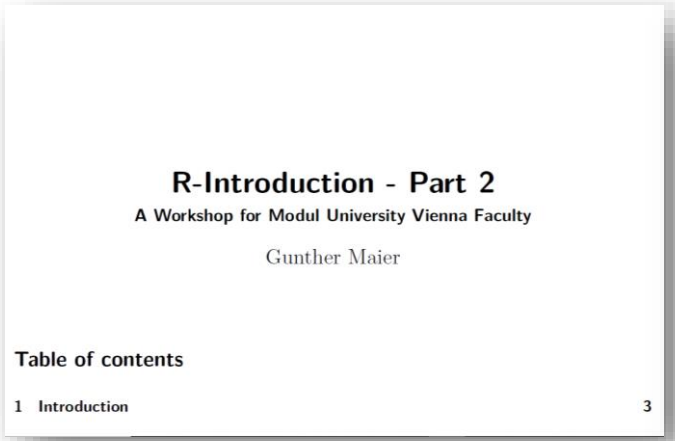


Figure 1: Temperature and ozone level.

Markdown text and Quarto Extensions

- Text can include (among others)
  - Formulas, equations, inline mathematics (LaTeX syntax)
  - References
    - Various formats (e.g., BibTeX, Zotero, Crossref lookup)
    - By default, only references cited in the text go into the list of references
  - Figures, tables
  - Footnotes
  - Hyperlinks
  - Numbered and unnumbered lists

- Quarto can be extended
  - Shortcodes/Filters
  - Journal Article templates
  - Custom Formats
  - Revealjs plugin
- Managed and made available via GitHub repositories


R-Introduction - Part 2

A Workshop for Modul University Vienna Faculty

Gunther Maier

Table of contents

1 Introduction 3


REGION

The Journal of ERSA
Powered by WU

Volume 1, Number 1, 2020, 999–999
DOI: 10.18335/region.v??i??.???

journal homepage: region.ersa.org
ISSN: 2409-5370

R-Introduction - Part 2
A Workshop for Modul University Vienna Faculty

Gunther Maier

Received: /Accepted:

How Quarto works

- All is free and open software
- **knitr** executes the code chunks and outputs the result
- **pandoc** is the swiss army knife of file conversion
  - Converts between many different file formats
  - Details can be specified with parameters, templates and Lua-filters
- Quarto manages the whole process
- Full documentation on https://quarto.org

What is R?

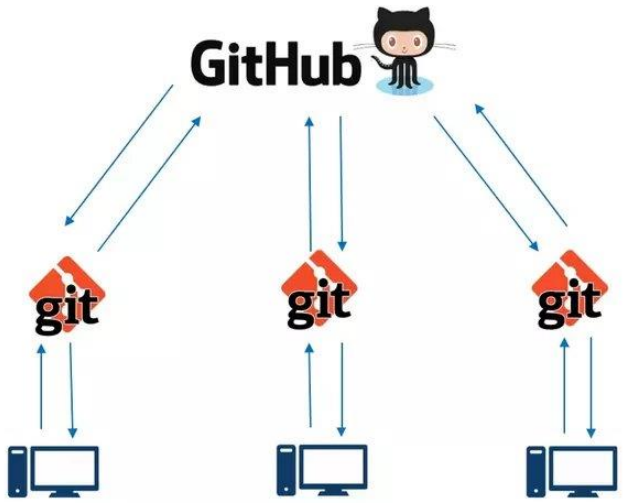
- A programming language
- A data management tool
- A tool for statistical analysis
- A tool for generating graphics
- Free and open source
- Very well supported by a huge community
- Very well documented
- Completely open structure

- R is somewhat difficult to use
- R has many parameters
- Inconsistent naming and structure
- Huge number of packages that add functionality
- The core of R is fairly small, most functions are in packages
- Which packages do I need? How do they work?

Git and GitHub

- Git
  - Local version control system running on local computer
  - keeps track of changes (changed, new, deleted files)

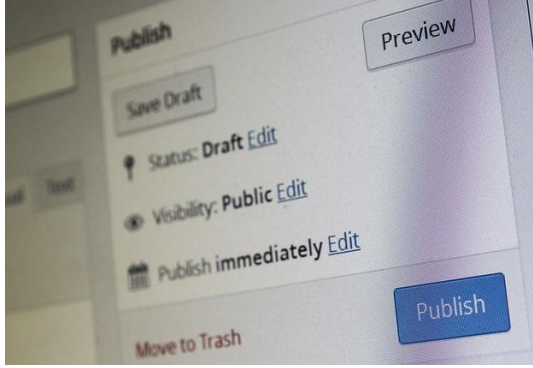
- GitHub
  - Web based repository in the cloud
  - Allows for collaboration with other users
  - Source for “inofficial” R-packages, Quarto-templates, etc.
  - Publishing framework (GitHub-Pages) with automatic updating



Advantages for the author

- Avoids using proprietary formats.
- qmd-files are plain text – long term readable, reusable
- Embedded R-code improves reproducibility
- Also documents the research process
- BibTeX-based literature databases are reusable
  - only cited literature is put into the list of references
  - with a proper journal template, formatting is adjusted automatically
  - If necessary, you can copy and adjust a style definition
- The final format is selected at the end and easily changed
- One can produce different versions from the same source

The publisher's traditional workflow



1. Author submits an article
2. Editor sends article to reviewers

A
3. Reviewers suggest changes
4. Author revises the article

5. Accepted article sent to copyediting

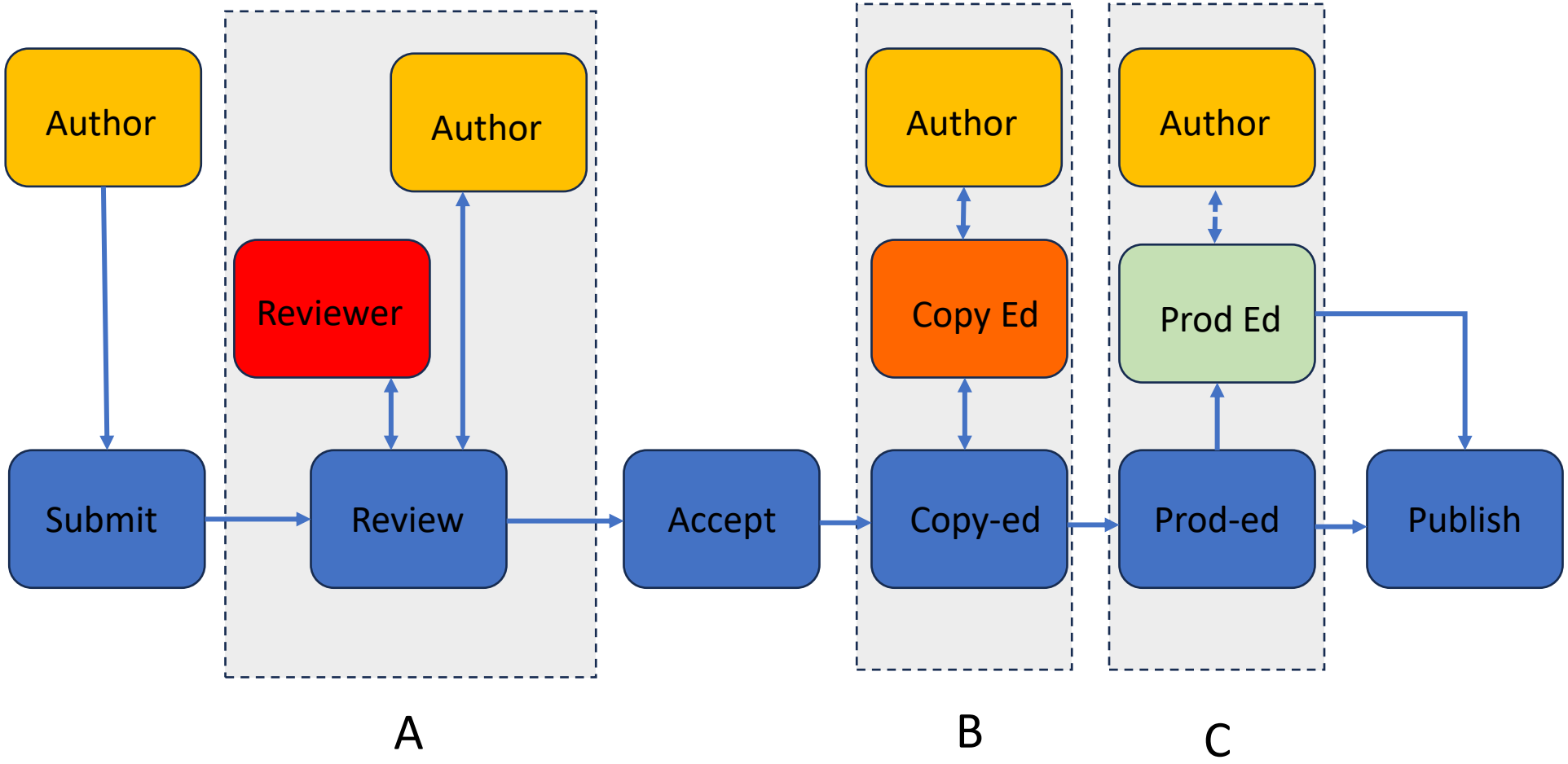
B
6. Copyeditor suggests changes
7. Author implements or rejects

8. Article sent to production

C
9. Production editor produces final version

- Open Journals System (OJS) supports this workflow
- Reviewing, copyediting, and production happen outside of OJS (media breaks A, B, C)
- Most articles are submitted in DOCX-format
- Copyediting and reviewing typically use Word Track Changes
- DOCX is tedious in production (no clear Markup structure)

The publisher’s traditional workflow



Advantages of a Quarto-based workflow

- Quarto implements the Markup principles
  - structure and formatting are separated
- Quarto documents can be converted into other Markup
  - HTML, LaTeX, E-Pub, PDF – Pandoc provides that
- Formatting can be implemented with a Quarto template
- Quarto, R, and Pandoc can run on the server
- Integration into OJS should be possible with an OJS-Plugin
- Process can become faster, easier, and more reliable
- Removing the media breaks can save time, cost, and effort
- Step toward open science

Implementing a Quarto-based workflow

- Reviewing step (A):
  - Involves untrained and untrainable actors with opt-out option
  - Must be easy and fool-proof (no special format, no code to enter)
  - Must implement double blind
  - Could be limited to commenting (no text editing)
  - Could be implemented as: Article on a web-page with commenting option (mark text, add comment)
- Copyediting step (B):
  - Involves untrained but trainable actors
  - Should be easy and fool-proof (avoid support requests and error correction)
  - Requires text editing (comment, delete, insert, replace; suggest, accept, reject)
  - Could be implemented as: Article on a web-page with editing option; Problem of transferring edits back to QMD-file.

Implementing a Quarto-based workflow

- Solutions for A and B should ideally be identical (maybe with de-activated functions)
- Production step (C):
  - Done by an expert (trained person)
  - Supported by Quarto template
  - No communication needed
  - Easy to implement
  - Ideally done with one click on the server
- Implementation strategy could be:
  1. Install tools on the server,
  2. Implement C,
  3. Implement B, test thouroughly
  4. Implement A

Summary and conclusions

- Quarto is an attractive option for academic publishing
- Quarto can integrate text and analysis (R, Python, etc)
- Offers advantages for authors and for publishers
- Advantages for publishers are not yet implemented
- Implementation requires detailed knowledge of OJS plugins, publisher’s workflow, Quarto workflow, various Markup languages, CSS, Javascript, etc. → team of professionals needed

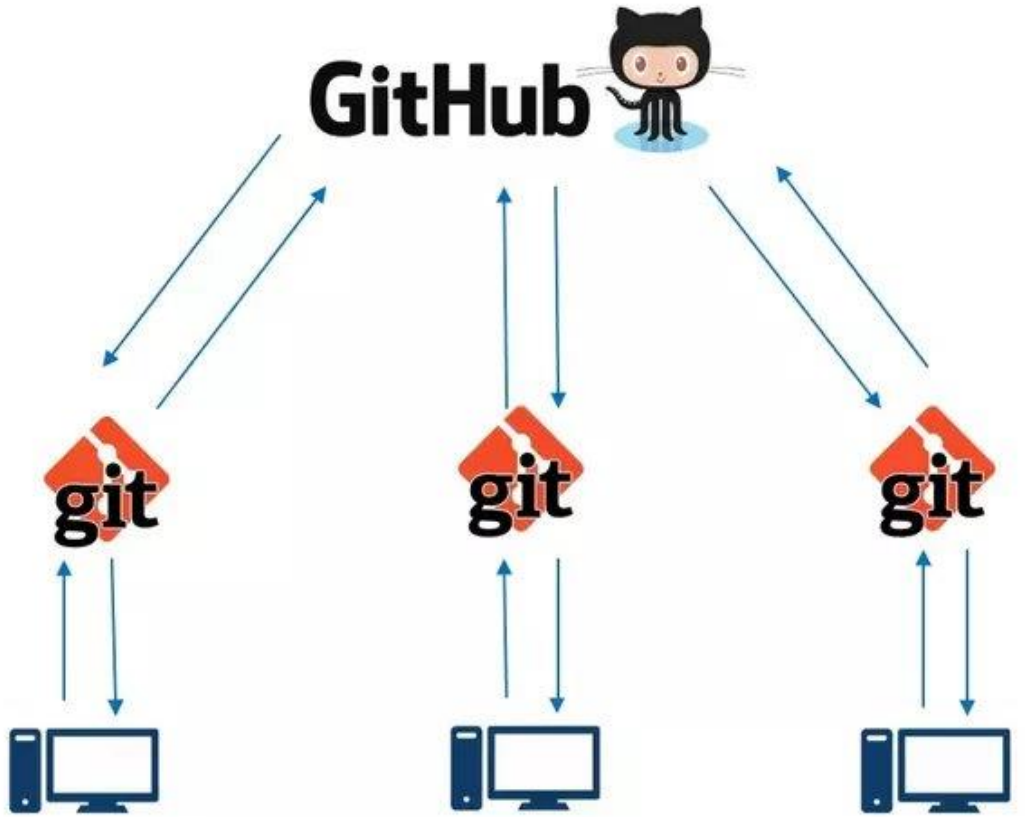
Thank you for your attention!

Where can you publish your work?

- ✓ Traditional journal (e.g., Review of Regional Research)
- ✓ Online journal (e.g., REGION)
- ✓ Traditional book publisher (e.g., Routledge)
- ✓ Your own (university's, employer's) web page
- ✓ Online book repository (e.g., Open Textbook Library)
- ✓ With Print on Demand (e.g., BoD) in online bookstores
- ✓ Any future web-based format

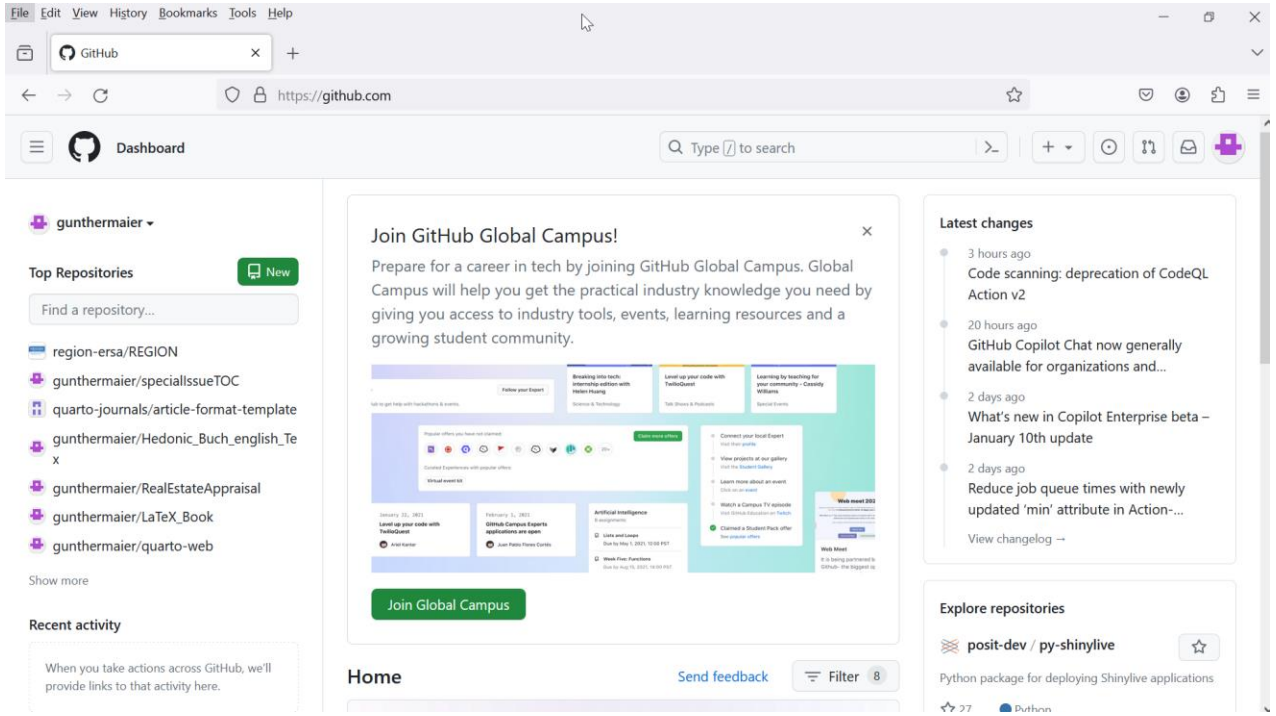
Git and GitHub

- Git is installed on your local computer
- GitHub is a Web application
  - Alternatives: Bitbucket, GitLab
- You link your local Git repos with repos on your GitHub accounts
- You try to keep them in sync
- In Git, you “pull” from and “push” to GitHub
- Work best with text-files



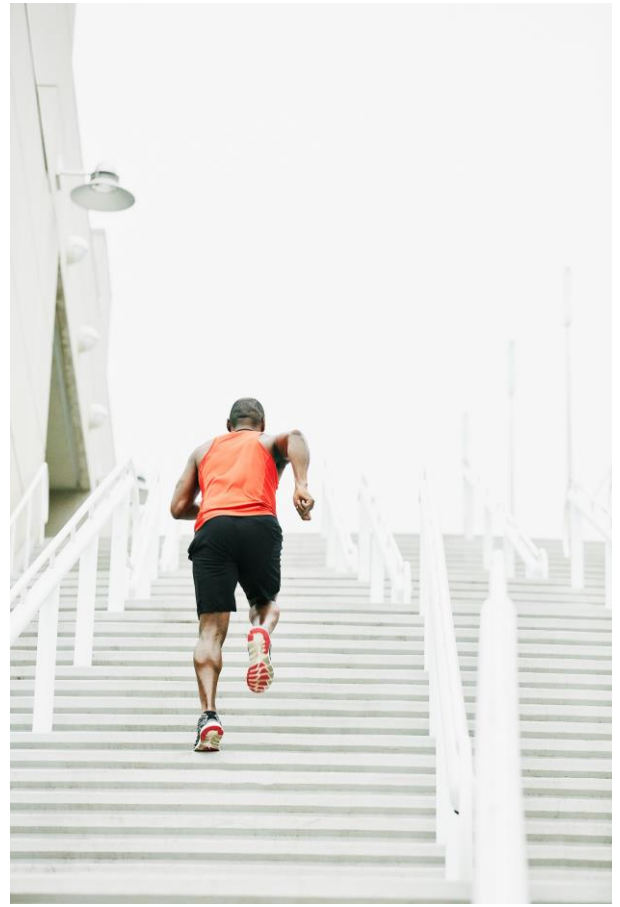
What is GitHub?

- Web-based git repository hosting service
- Stores copies of local git repositories
- Version control & backup
- Collaboration with other users
  - To develop code & content
- Source for “inofficial” R-packages, Quarto-templates,etc.
- Publishing framework (GitHub-Pages) with automatic updating



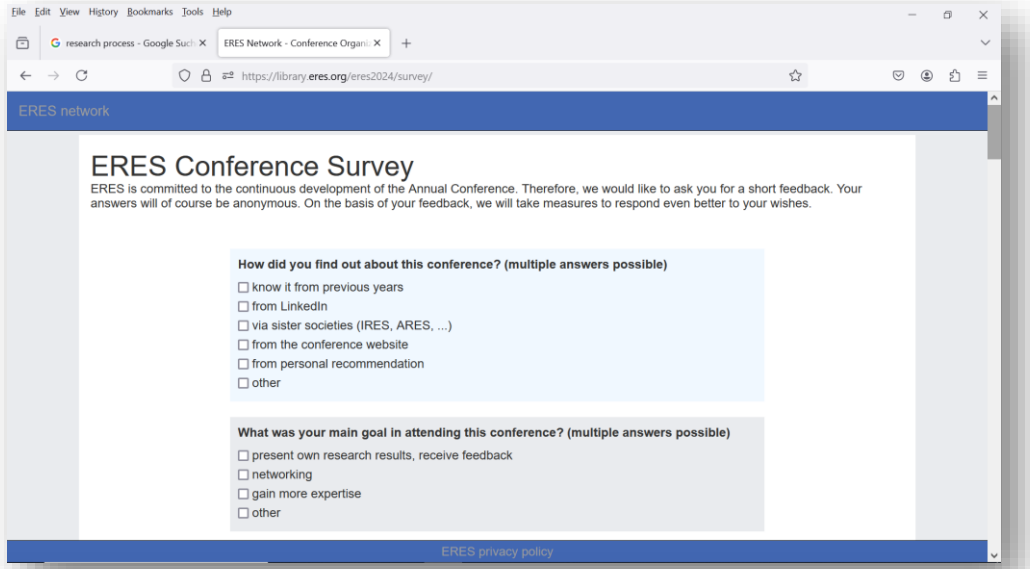
Publishing in this framework – step by step

1. Start a new Quarto project (creates a new folder)
2. *Initialize Git, link to GitHub repo. Invite co-authors*
3. Write your article in Quarto markdown
4. *Embed code chunks with R-code to get analysis, tables, figures, etc*
5. Link bibliographic records in BibTeX-format
6. Install a journal template from GitHub (if available)
7. Define a style of reference (if necessary)
8. Render your document to PDF, HTML, docx, …
9. Submit to the journals, publish on GitHub Pages, in a repository, …



A short example

- After this conference, you will be asked to fill in this survey
- Data are stored on the ERES server
- We want to read and analyze those data.
- We want to get a written report of the results
- My solution: Write a Quarto document with text and analysis combined



Introduction

This document reports the results of the ERES Survey for the conference `r params$conf`. The conference is selected via the parameter "conf" in the header of this file. To see the full report as a web-page, click the "Render" button in RStudio. If you want a PDF-document, change the "format" in the header to "pdf".

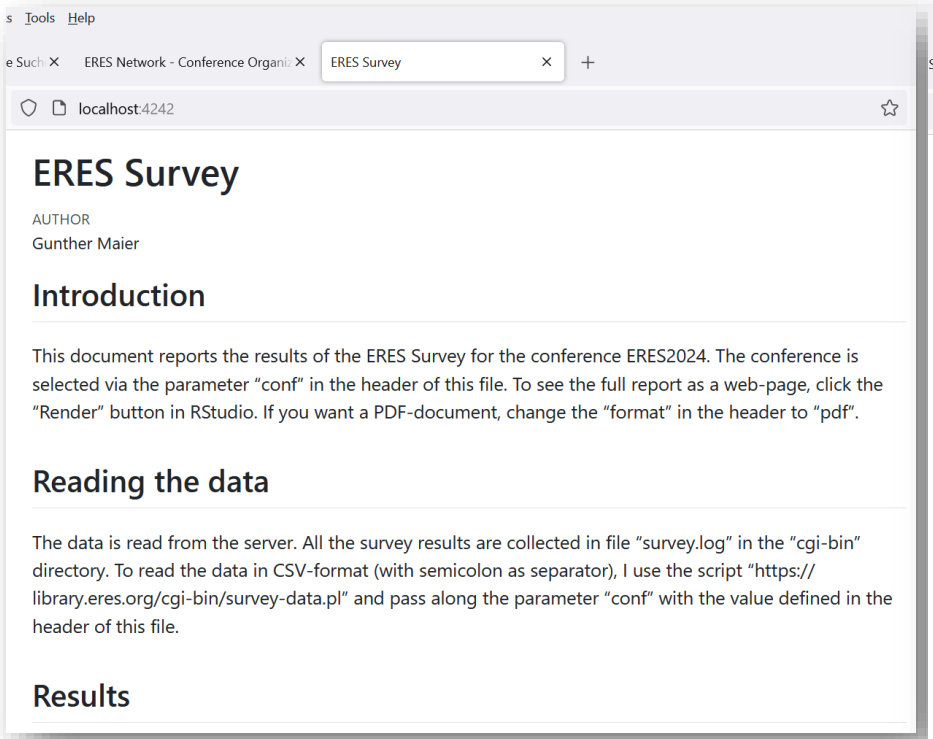
Reading the data

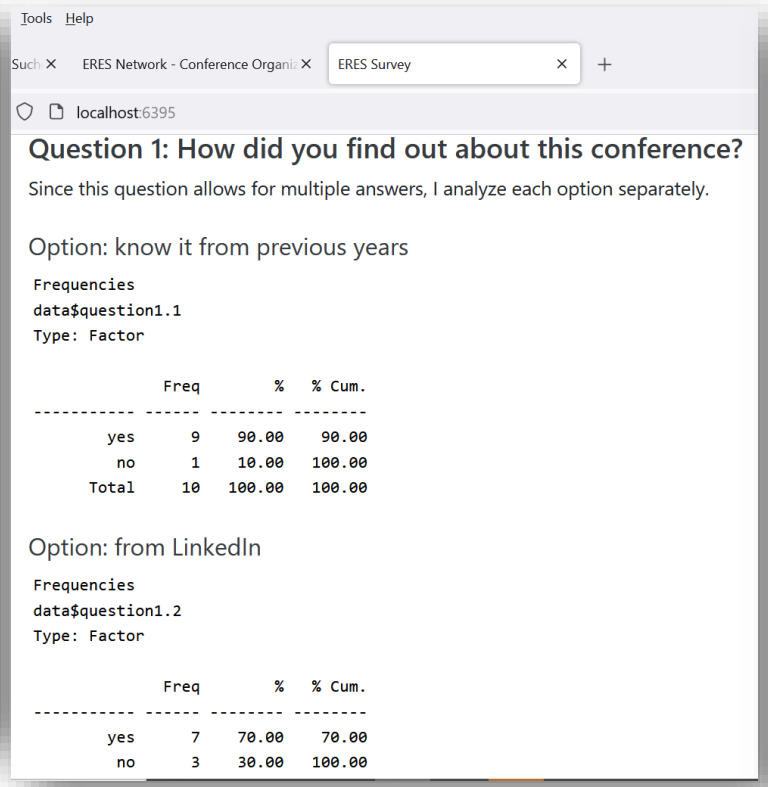
The data is read from the server. All the survey results are collected in file "survey.log" in the "cgi-bin" directory. To read the data in CSV-format (with semicolon as separator), I use the script "https://library.eres.org/cgi-bin/survey-data.pl" and pass along the parameter "conf" with the value defined in the header of this file.

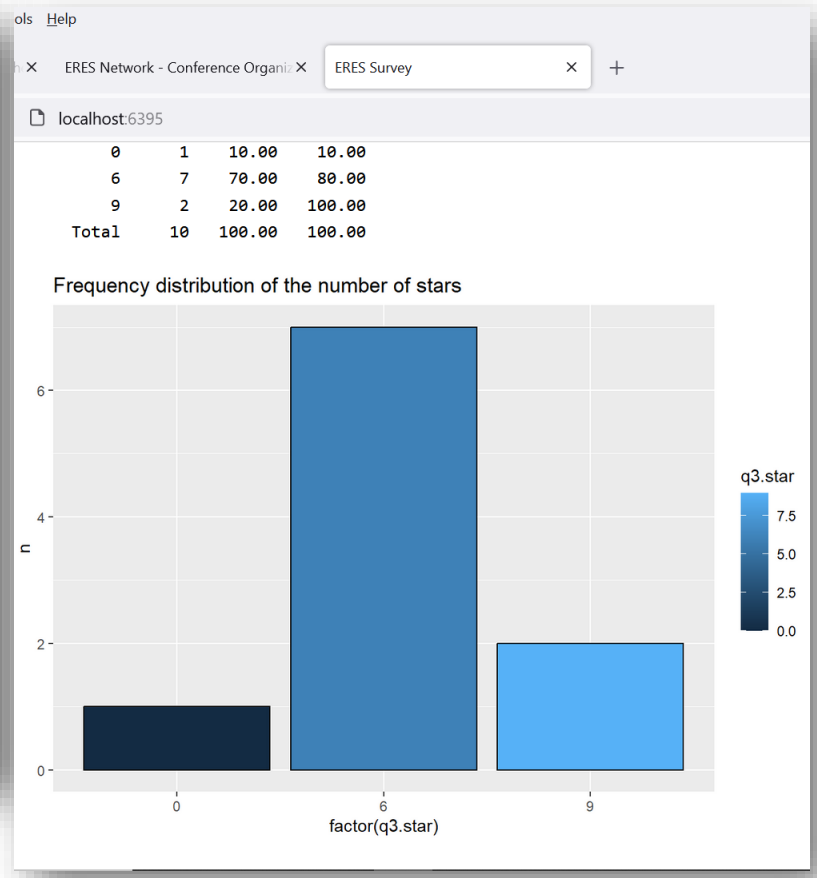
The following code chunk downloads the data and stores it in a data-frame named "data".

```
{r}
url <- paste0("https://library.eres.org/cgi-bin/survey-data.pl","?conf="
,params$conf)
data <- read.csv(url, header = TRUE, sep = ";")
data[sapply(data, is.character)] <- lapply(data[sapply(data, is.character)], as
.factor)
data$question13 <- factor(data$question13, levels = c("MTL", "TL", "AR", "TS",
"MTS"), labels = c("Much too Long", "Too Long", "About Right", "Too Short", "Much
too Short"))
data$question18 <- factor(data$question18, levels = c("PhD", "Acad", "Board",
```

Standard Webpage output







Some other formats **`format: pdf`**

ERES Survey

Gunther Maier

Introduction

This document reports the results of the ERES Survey for the conference ERES2024. The conference is selected via the parameter "conf" in the header of this file. To see the full report as a web-page, click the "Render" button in RStudio. If you want a PDF-document, change the "format" in the header to "pdf".

Reading the data

The data is read from the server. All the survey results are collected in file "survey.log" in the "cgi-bin" directory. To read the data in CSV-format (with semicolon as separator), I use the script "https://library.eres.org/cgi-bin/survey-data.pl" and pass along the parameter "conf" with the value defined in the header of this file.

Results

Question 1: How did you find out about this conference?

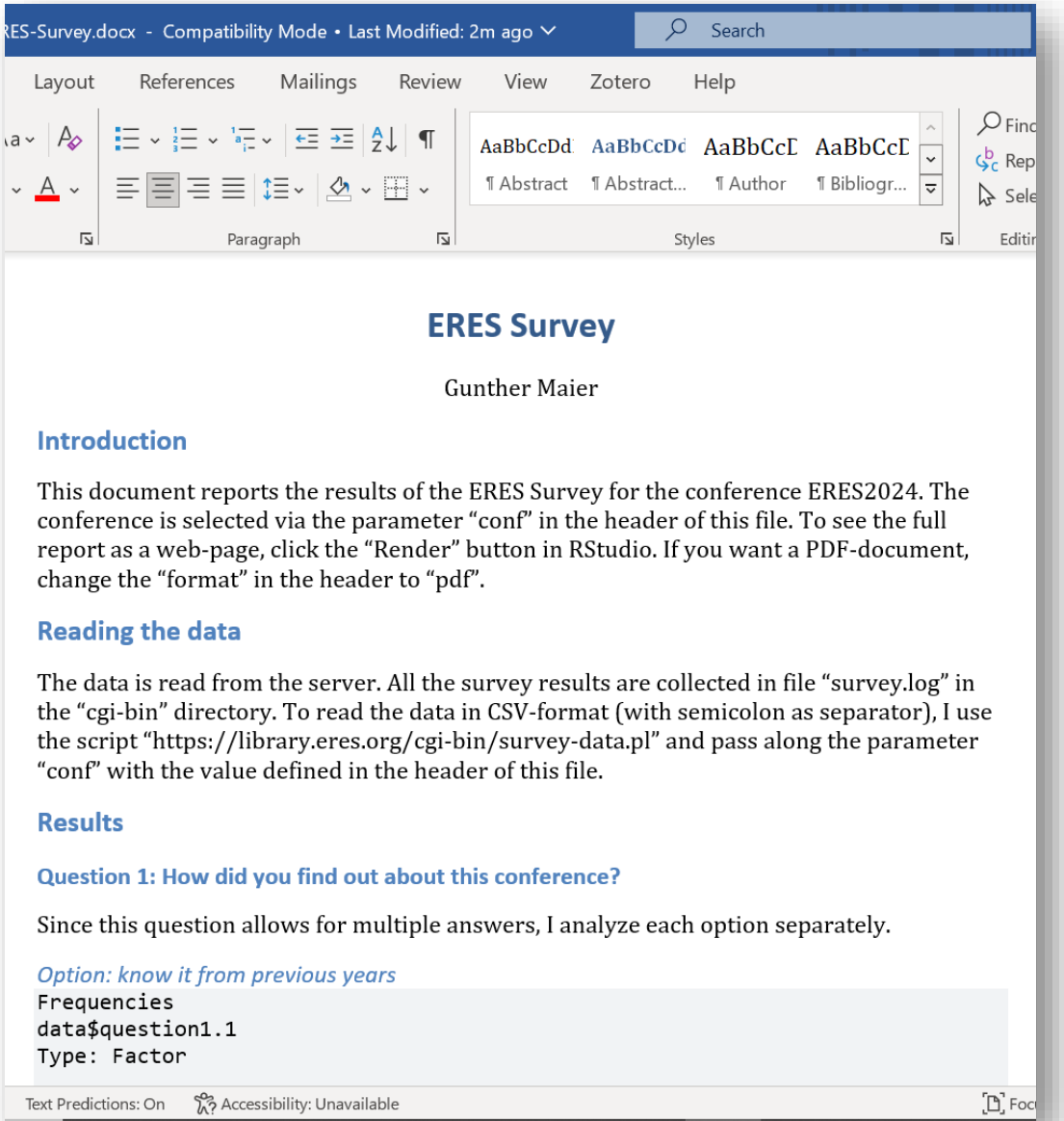
Since this question allows for multiple answers, I analyze each option separately.

Option: know it from previous years

```
Frequencies
data$question1.1
Type: Factor

            Freq       %   % Cum.
---------- ------ -------- --------
       yes      9    90.00    90.00
```

Some other formats **`format: docx`**



Some other formats `format: epub`