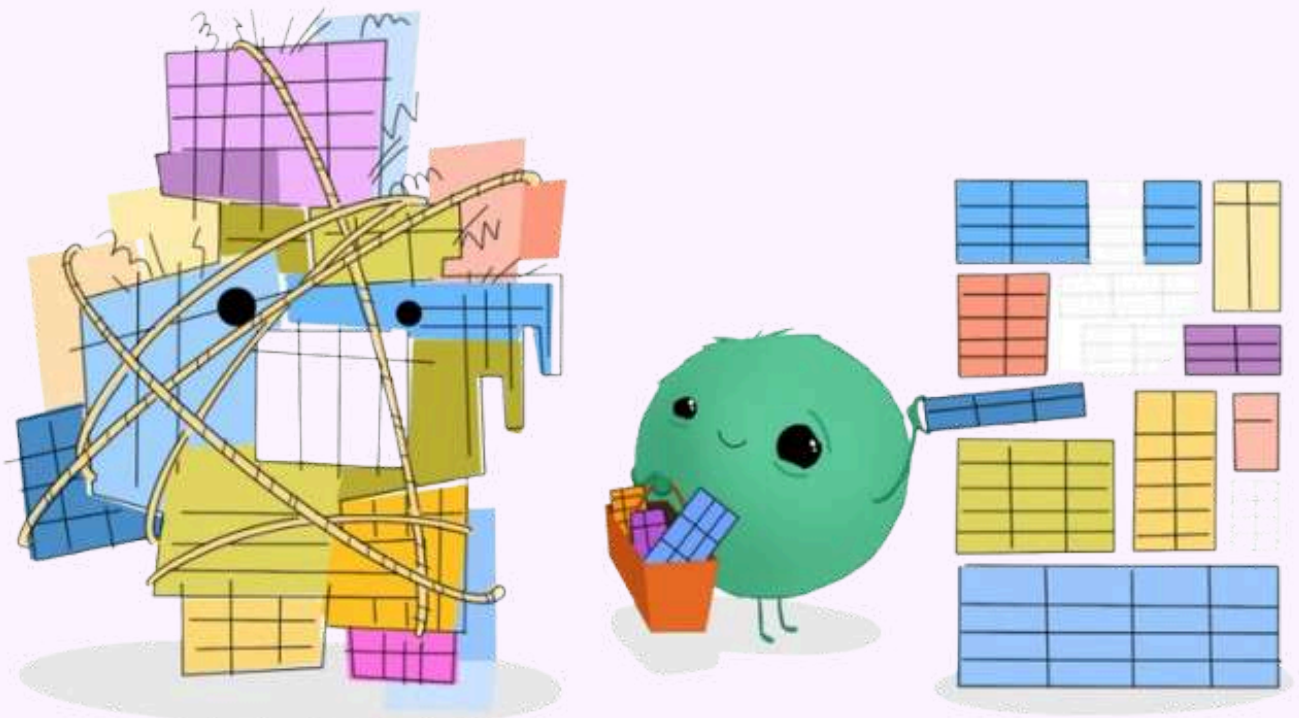
Retrospective clinical data harmonisation Reporting Using R and Quarto

Jeremy Selva
@JauntyJJS
https://jeremy-selva.netlify.app
For Harvest Analytics Together (HAT) 2025

14th November 2025

whoami

Research Officer from National Heart Centre Singapore who collects, cleans and harmonises clinical data.



Taming the Data Beast from "Cleaning Medical Data with R" workshop by Shannon Pileggi, Crystal Lewis and Peter Higgins presented at R/Medicine 2023.

Illustrated by Allison Horst.

About Data Harmonisation

Data harmonisation is part of data wrangling process where

- Similar variables from different datasets are identified.
- Grouped based on a generalised concept they represent.
- Transformed into unified harmonised variables for analysis.

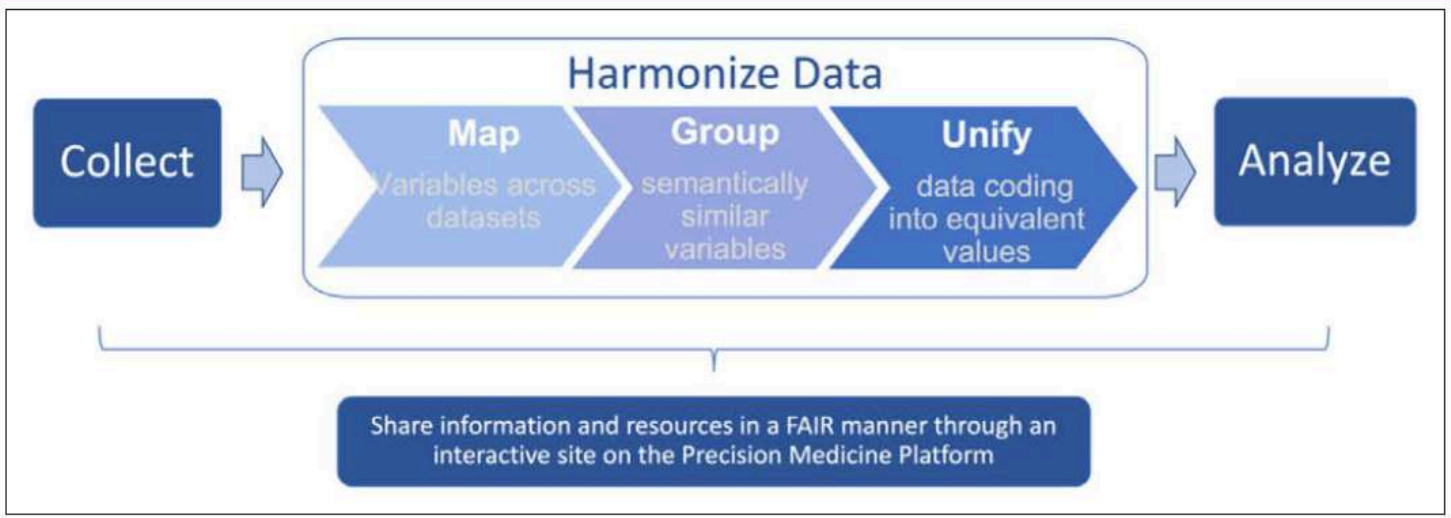
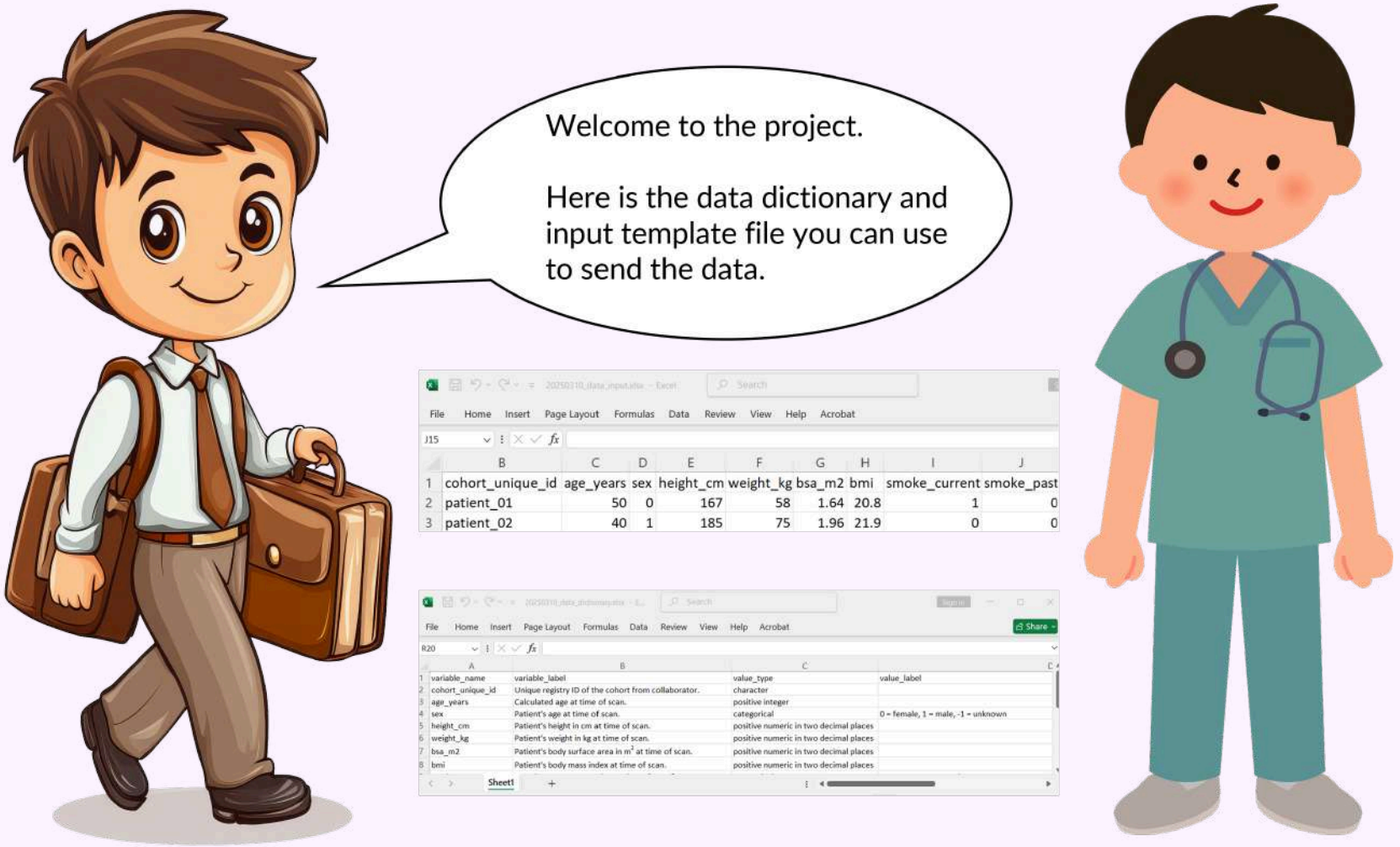


Figure 1. The data harmonization process.
Study data variables collected from different sources need to be mapped to one another (step 1), classified into the generalized concepts they represent (step 2), and transformed into unified harmonized variables (step 3) for analysis.

Image from Mallya et al. Circ Cardiovasc Qual Outcomes. 2023 Nov; 16(11):e009938 doi: 10.1161/CIRCOUTCOMES.123.009938.

How it started



Cheerful Businessman designed by Iftikhar Alam from Vecteezy and Medical Doctor Man from Creazilla.

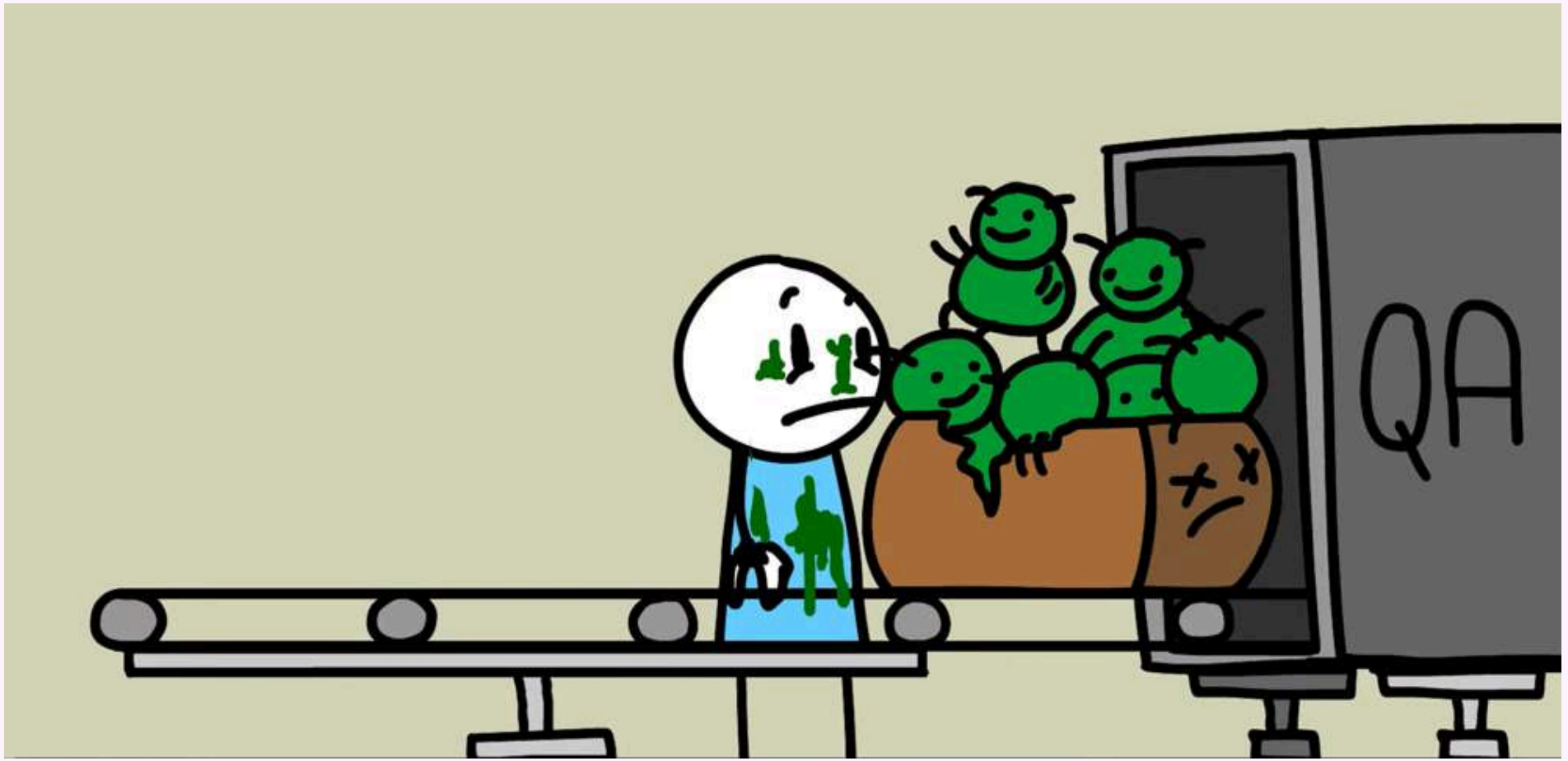
How it started

Cheerful Businessman designed by Iftikhar Alam from Vecteezy and Medical Doctor Man from Creazilla.

How it started

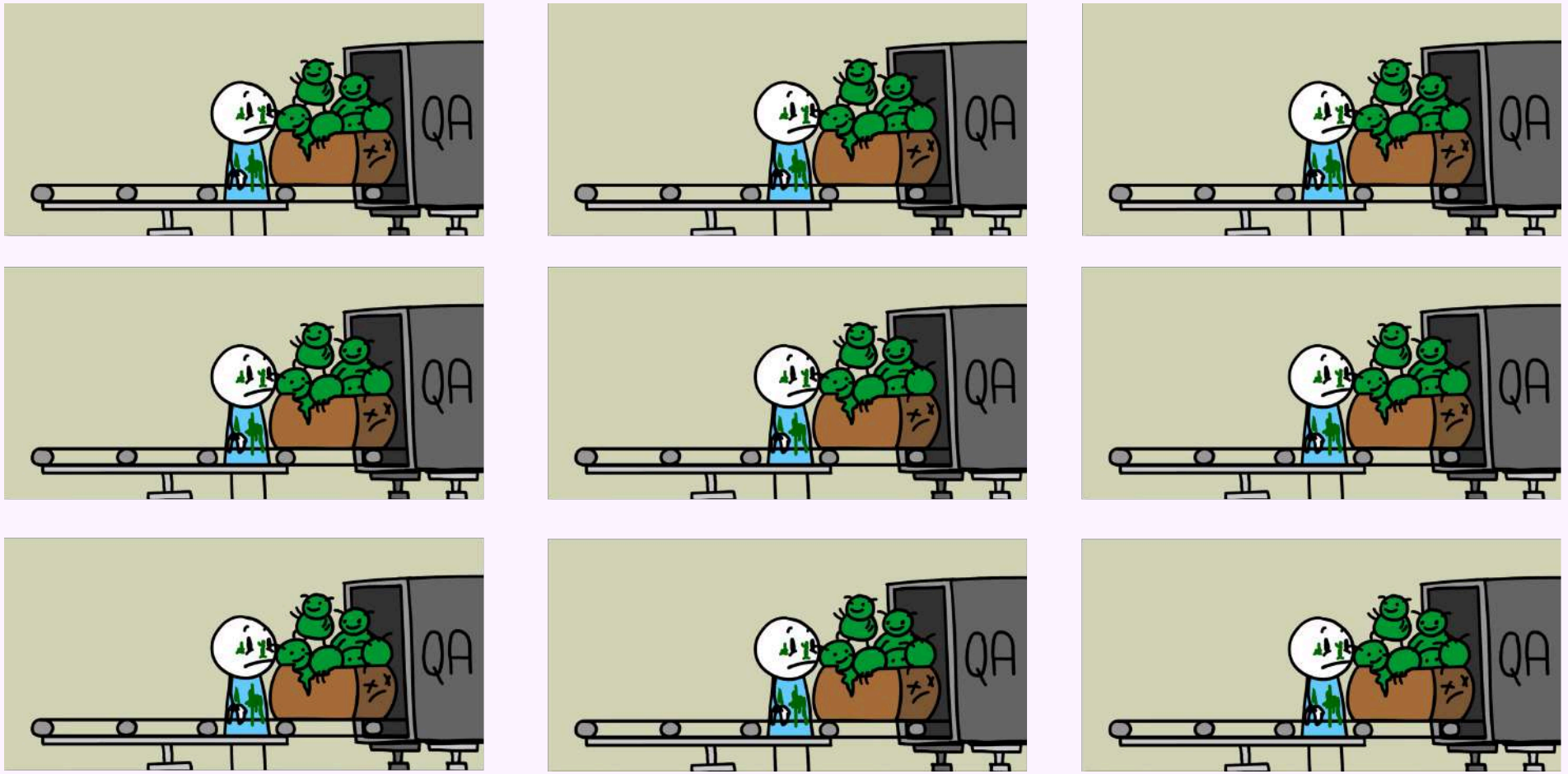
Cheerful Businessman designed by Iftikhar Alam from Vecteezy and Medical Doctor Man from Creazilla.

How it started



snapshot from Ready for QA | MonkeyUser 2SP Animation Video from MonkeyUser.com.

How it started



snapshot from Ready for QA | MonkeyUser 2SP Animation Video from MonkeyUser.com.

How it started

Turn my sorrow into opportunities.

Tackling Formatted Tabular Data from Excel

10th July 2024

Jeremy Selva
@JauntyJJS
https://jeremy-selva.netlify.app

Why a harmonisation report

Cheerful Businessman designed by Iftikhar Alam from Vecteezy and Medical Doctor Man from Creazilla.

Why a harmonisation report

Some data fields just cannot be planned in advanced.

Cohort 1 Race/Ethnicity
Chinese
Indian
Malay
Eurasian
Others

Cohort 2 Race/Ethnicity
White
Black
Asian
Mixed
Others

Cohort 3 Race/Ethnicity	
Race	**Ethnicity**
White	Hispanic/Latino
Black	Not Hispanic/Latino
Asian	
Native American	
Pacific Islanders	
Others	

Cohort 4 Race/Ethnicity in text	
Latino	Brazilian
White	British
Asian	Chinese
Middle Eastern	Egyptian
Asian	Korean
Asian	Filipino
White	German
Asian	Japanese
Asian	Korean
Asian	Pakistani
Latino	Peruvian
Middle Eastern	Saudi Arabian
African	Ugandan
etc ...	

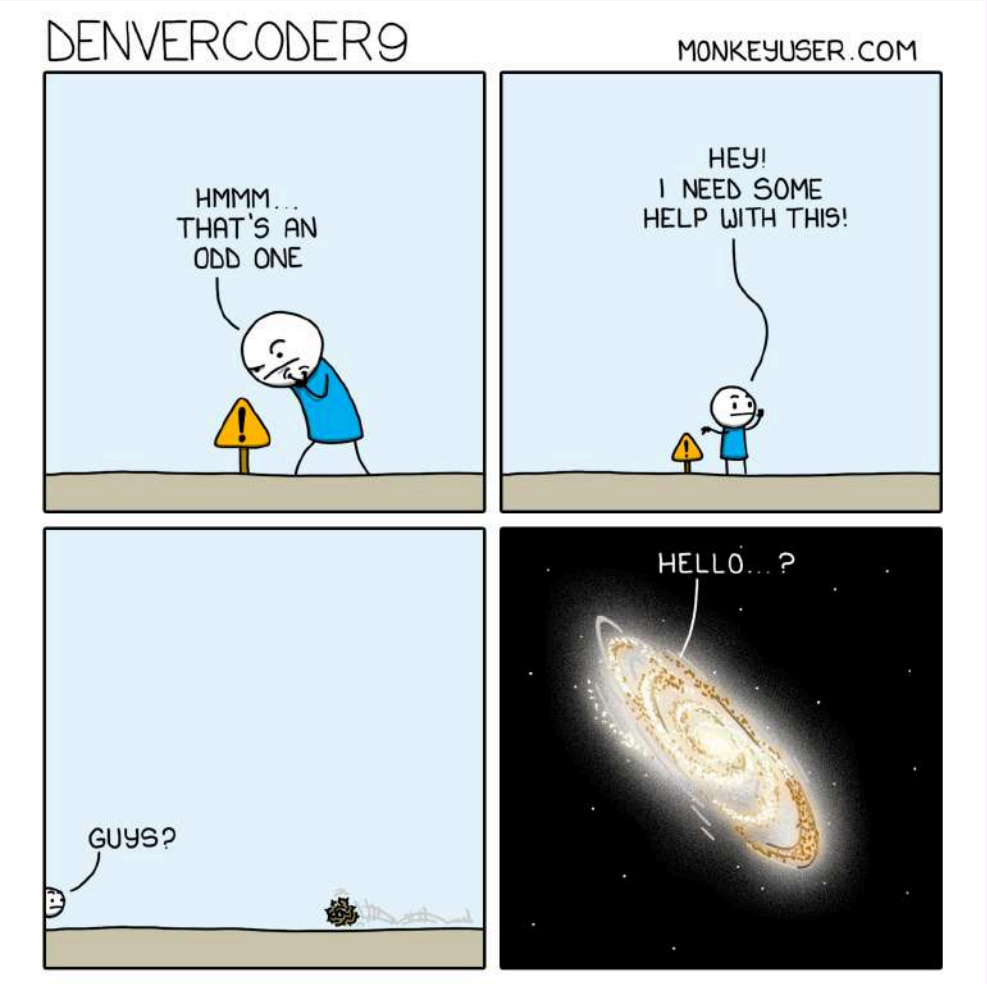
New Cohort Race/Ethnicity
White
African
Southeast Asian
East Asian
South Asian
Other Asians
Middle Eastern
Torres Straits Islanders
Aboriginal
Others

Issues

While there are R 📦 to facilitate data harmonisation,

- `retroharmonize` for survey data.
- `Rmonize` for epidemiological data.
- `psHarmonize` for health and education data.

There are limited resources on how to make a data harmonisation report.



Denvercoder9 from MonkeyUser.com

Harmonisation Project Template

A template to offer a systematic way to report data harmonisation processes.

Link: https://jauntyjjs.github.io/harmonisation/

Data Harmonisation Project Template

Table of Content

Links

Browse source code

Report a bug

License

Full license

MIT + file LICENSE

Citation

Citing harmonisation

Developers

Jeremy Selva

Author, maintainer

Tools to create documentation

R Programming Logo from CleanPNG and Quarto Hex Sticker from Posit.

What *Did* We Forget to Teach You about R?

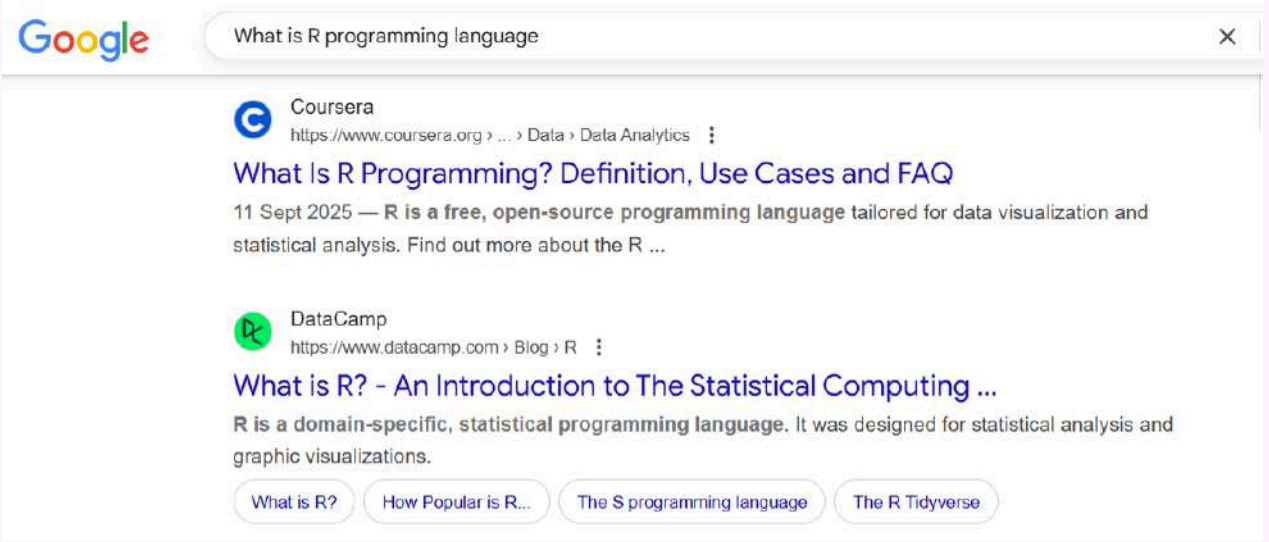


Image from Project Oriented Workflows slides from What They Forgot to Teach You About R.

We will share a glimpse of “Everything else” R and its friends can do.

Pipes

R ≥ 4.1.0 has a "pipe" symbol |> to make code easier to read.

Without |>

```
data_after_task_3 <- task_3(task_2(task_1(data, arg_1_2), arg_2_2, arg_2_3), arg_3_2, arg_3_3)
```

```
data_after_task_1 <- task_1(data, arg_1_2)

data_after_task_2 <- task_2(data_after_task_1, arg_2_2, arg_2_3)

data_after_task_3 <- task_3(data_after_task_2, arg_3_2, arg_3_3)
```

With |>

```
data_after_task_3 <- data |>
  task_1(arg_1_2) |>
  task_2(arg_2_2, arg_2_3) |>
  task_3(arg_3_2, arg_3_3)
```

Inspired from the Bash Pipe |

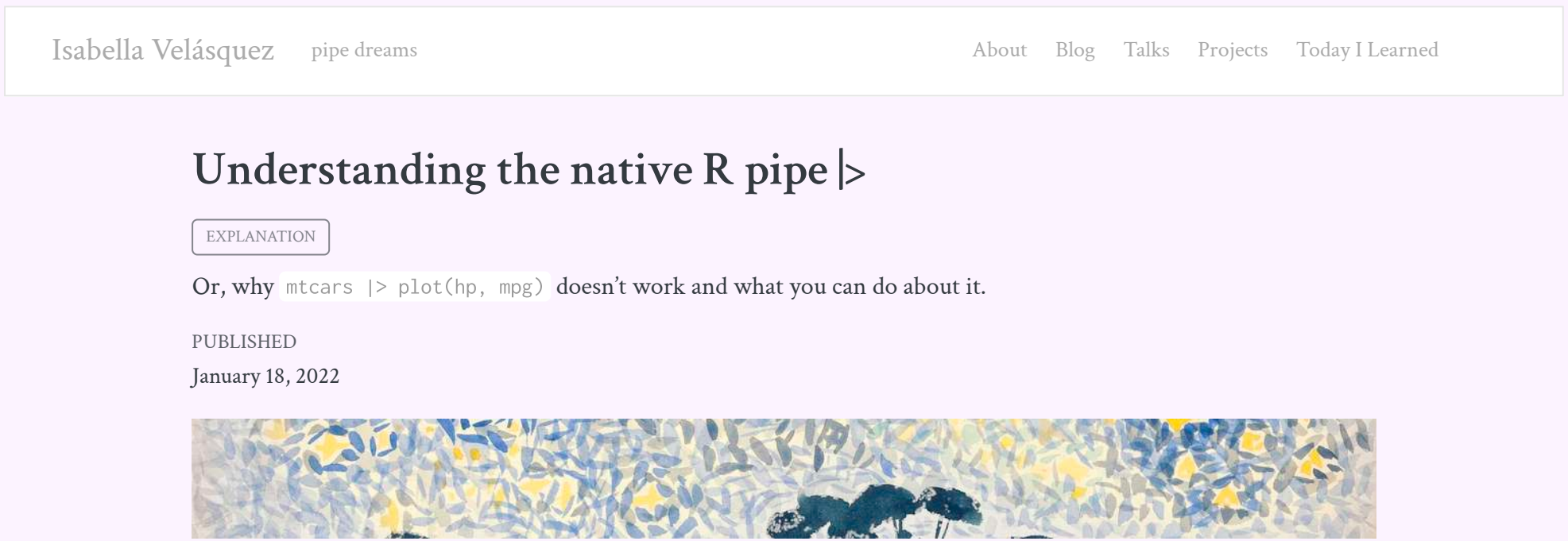
terminal

```
# List files, then filter by row, then filter by column, then sort.
ls -l | grep drw | awk '{print $9}' | sort
```

Pipes

- 2014+ R 📦 magrittr pipe `%>%`
- 2021+ (R ≥ 4.1.0) native R pipe `|>`

🛜 More details between the two pipes in Understanding the native R pipe |>.



Namespacing

`dplyr::select()`

- tells R explicitly to use the function `select` from the package `dplyr`
- can help to avoid name conflicts (e.g., `MASS::select()`)
- does not require `library(dplyr)`

Without Namespace

```
1 library(dplyr)
2
3 select(mtcars, mpg, cyl)
4
5 mtcars |>
6   select(mpg, cyl)
```

With Namespace

```
1 # library(dplyr) not needed
2
3 dplyr::select(mtcars, mpg, cyl)
4
5 mtcars |>
6   dplyr::select(mpg, cyl)
```

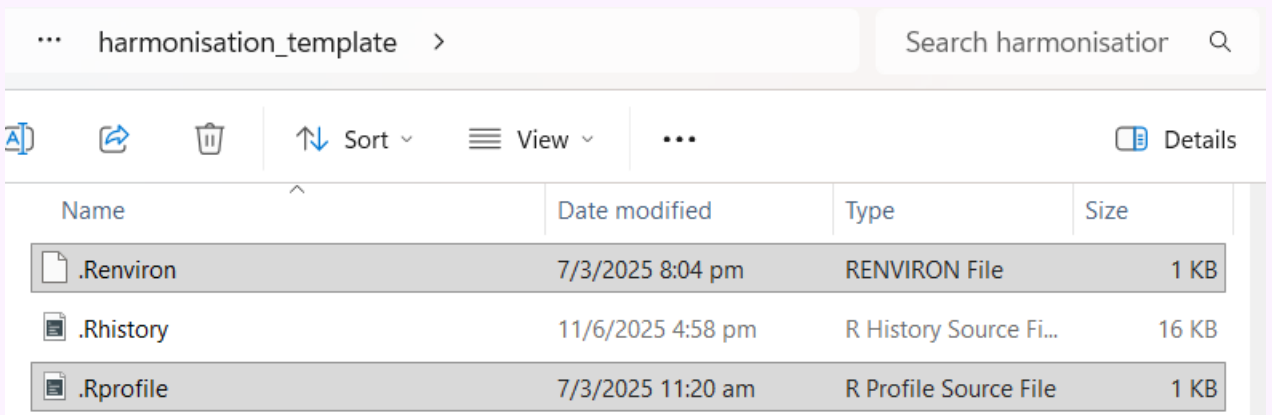
R Session

Before R starts up in a given project, it will perform the following steps

1. Set up Environment Variables.
2. Run startup script.
3. Set up the R session.

We can customise Step 1 and 2 using these two main text files.

- `.Renviron` (Contains environment variables to be set in R sessions.)
- `.Rprofile` (Contains R code to be run in each session.)



what goes in `.Renviron`

✅ R-specific environment variables.
✅ API keys or other secrets
❌ R code

```
APPDATA="D:/Jeremy/PortableR/RAppData/Roaming"
LOCALAPPDATA="D:/Jeremy/PortableR/RAppData/Local"
TEMP="D:/Jeremy/PortableR/RPortableWorkDirectory/temp"
TMP="D:/Jeremy/PortableR/RPortableWorkDirectory/temp"
_R_CHECK_SYSTEM_CLOCK_=0
RENV_CONFIG_PAK_ENABLED=TRUE
CONNECT_API_KEY=DaYK2hBUriSBYUEGIAiyXsRJHSjTYJN3
DB_USER=elephant
DB_PASS=p0stgr3s
```

user

`~/.Renviron`

project

`path/to/your/project/.Renviron`

```
Sys.getenv("RENV_CONFIG_PAK_ENABLED")
```

```
[1] "TRUE"
```

what goes in `.Rprofile`

✅ set a default CRAN mirror.
✅ customize R prompt.

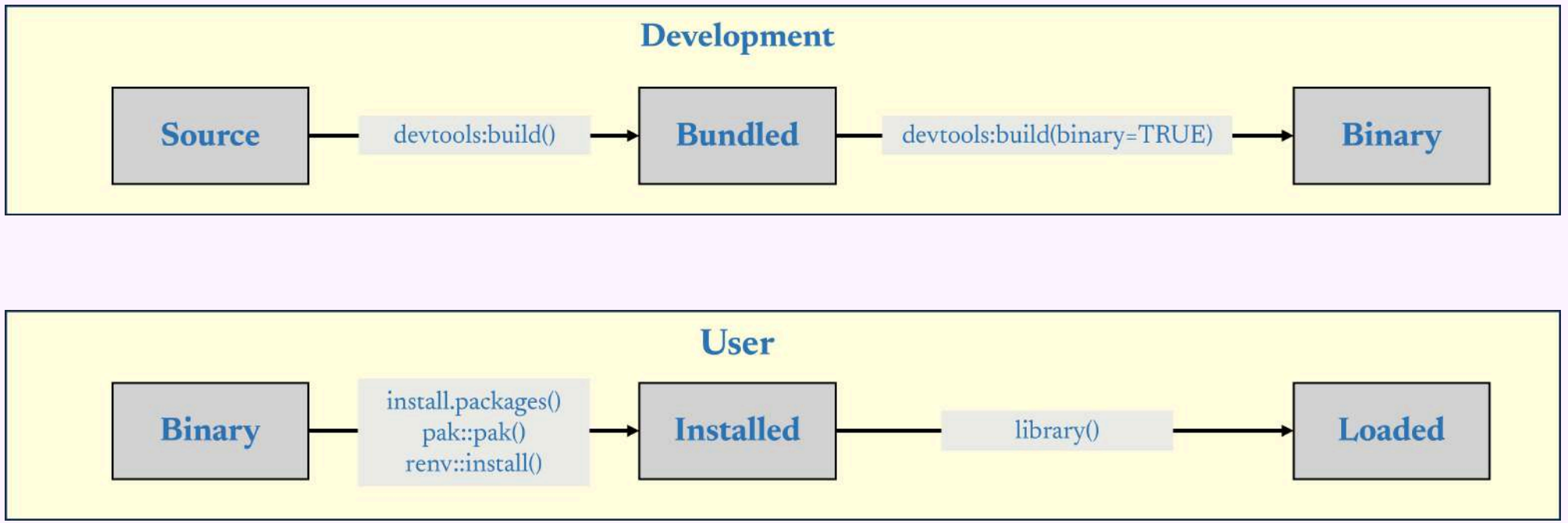
```r
source("renv/activate.R")
options(
  repos = c(
    P3M_20250306 = "https://packagemanager.posit.co/cran/2025-10-13",
    ropensci = "https://ropensci.r-universe.dev",
    janmarvin = "https://janmarvin.r-universe.dev",
    CRAN = 'https://cloud.r-project.org'
  )
)
if (interactive()) prompt::set_prompt(prompt::prompt_fancy)
```

Prompt Resources:

- 📦 – prompt.
- xinxxxin/rprofile-custom-prompt.R.
- Me, Myself and my Rprofile.
- Prompt-moting a custom R prompt.

States of R 📦

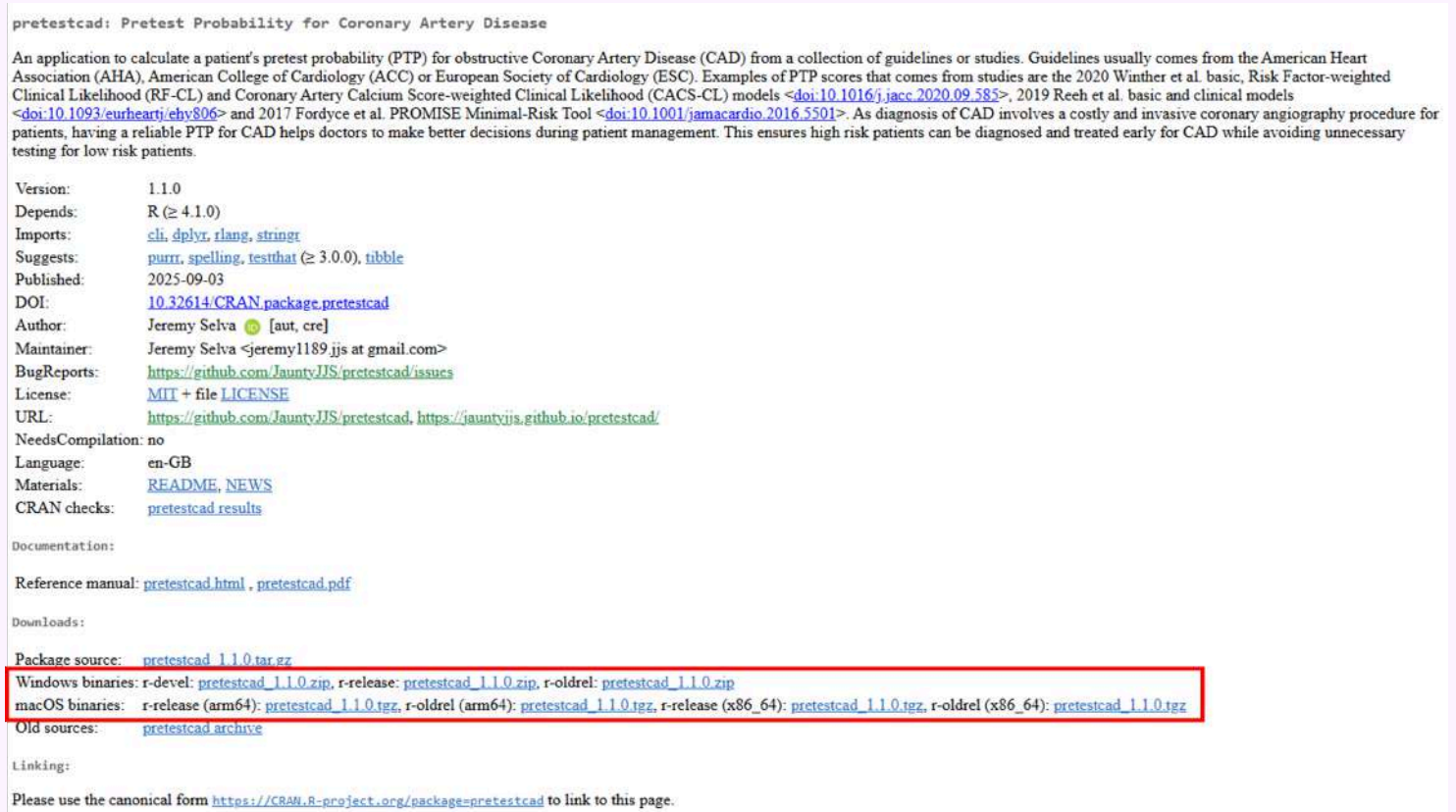
R 📦 – Structure



Binary R 📦 Installation

Go for (🪟 | 🍎) binary R 📦 in CRAN.

- compiled ahead of time -> easiest / fastest to install



pretestcad: Pretest Probability for Coronary Artery Disease

An application to calculate a patient's pretest probability (PTP) for obstructive Coronary Artery Disease (CAD) from a collection of guidelines or studies. Guidelines usually comes from the American Heart Association (AHA), American College of Cardiology (ACC) or European Society of Cardiology (ESC). Examples of PTP scores that comes from studies are the 2020 Winther et al. basic, Risk Factor-weighted Clinical Likelihood (RF-CL) and Coronary Artery Calcium Score-weighted Clinical Likelihood (CACS-CL) models <doi:10.1016/j.jacc.2020.09.585>, 2019 Reeh et al. basic and clinical models <doi:10.1093/eurheartj/ehy806> and 2017 Fordyce et al. PROMISE Minimal-Risk Tool <doi:10.1001/jamacardio.2016.5501>. As diagnosis of CAD involves a costly and invasive coronary angiography procedure for patients, having a reliable PTP for CAD helps doctors to make better decisions during patient management. This ensures high risk patients can be diagnosed and treated early for CAD while avoiding unnecessary testing for low risk patients.

Version:	1.1.0
Depends:	R (≥ 4.1.0)
Imports:	cli, dplyr, rlang, stringr
Suggests:	purrr, spelling, testthat (≥ 3.0.0), tibble
Published:	2025-09-03
DOI:	10.32614/CRAN.package.pretestcad
Author:	Jeremy Selva [aut, cre]
Maintainer:	Jeremy Selva <jeremy1189.jjs at gmail.com>
BugReports:	https://github.com/JauntyJJS/pretestcad/issues
License:	MIT + file LICENSE
URL:	https://github.com/JauntyJJS/pretestcad, https://jauntyjjs.github.io/pretestcad/
NeedsCompilation:	no
Language:	en-GB
Materials:	README, NEWS
CRAN checks:	pretestcad results

Documentation:

Reference manual: pretestcad.html , pretestcad.pdf

Downloads:

Package source:	pretestcad_1.1.0.tar.gz
Windows binaries:	r-devel: pretestcad_1.1.0.zip, r-release: pretestcad_1.1.0.zip, r-oldrel: pretestcad_1.1.0.zip
macOS binaries:	r-release (arm64): pretestcad_1.1.0.tgz, r-oldrel (arm64): pretestcad_1.1.0.tgz, r-release (x86_64): pretestcad_1.1.0.tgz, r-oldrel (x86_64): pretestcad_1.1.0.tgz
Old sources:	pretestcad archive

Linking:

Please use the canonical form https://CRAN.R-project.org/package=pretestcad to link to this page.

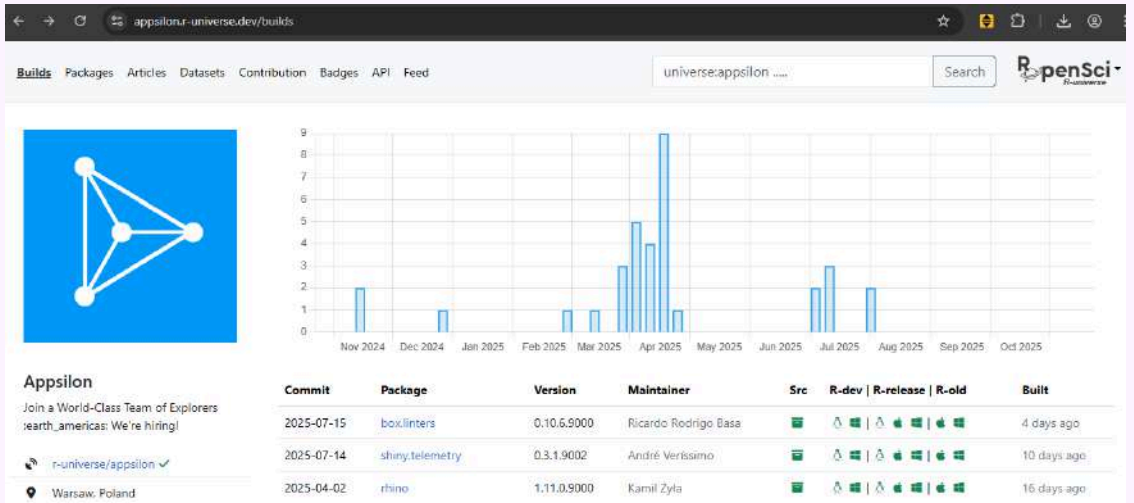
R-universe and Posit Public Package Manager

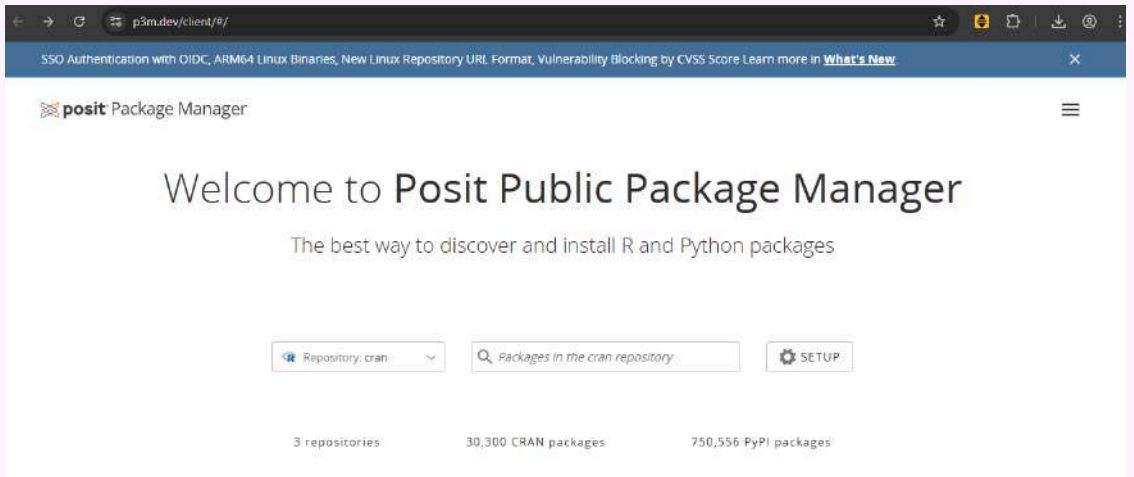
Consider installing (| |) binaries from

- R-universe
- Posit Public Package Manager.

Set in your `.Rprofile` file.

```
options(
  repos = c(
    P3M_20250306 = "https://packagemanager.pos
    ropensci = "https://ropensci.r-universe.de
    CRAN = 'https://cloud.r-project.org'
  )
)
```





How do I know I got a binary?

CRAN CRAN p3m

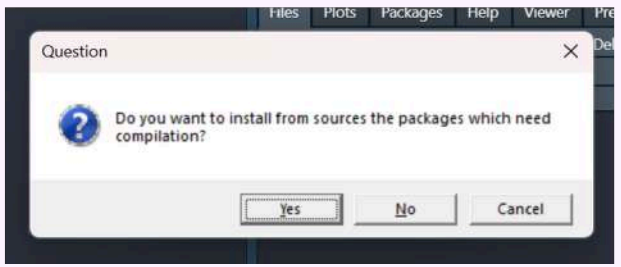
```
> install.packages("parallelly", repos = "https://cran.r-project.org")
Installing package into ‘C:/Users/edavi/Documents/R/win-library/4.1’
(as ‘lib’ is unspecified)
trying URL 'https://cran.r-project.org/bin/windows/contrib/4.1/parallelly_1.32.1.zip'
Content type 'application/zip' length 306137 bytes (298 KB)
downloaded 298 KB

package ‘parallelly’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\edavi\AppData\Local\Temp\Rtmpa2s3e8\downloaded_packages
```

Source 📦 Installation

Go for source 📦 if you need the latest version urgently.

Downloads:

Package source:	parallelly_1.32.1.tar.gz
Windows binaries:	r-devel: parallelly_1.32.1.zip, r-release: parallelly_1.32.0.zip, r-oldrel: parallelly_1.32.1.zip
macOS binaries:	r-release (arm64): parallelly_1.32.1.tgz, r-oldrel (arm64): parallelly_1.32.1.tgz, r-release (x86_64): parallelly_1.32.0.tgz, r-oldrel (x86_64): parallelly_1.32.0.tgz
Old sources:	parallelly archive



🐧 only have source 📦 in CRAN.

Source 📦 Installation

You will need additional tools and dependencies.

windows | macOS | linux

Install Rtools

Run `devtools::has_devel()` in console.

```
## Your system is ready to build packages!
```

Install using R 📦 pak

Consider using `pak::pkg_install` instead of `install.packages()`

6 Package installation

When working in production, you're much more likely to be using a Linux server. R package installations are a little different there, so in this chapter you'll learn more about the best ways to install R packages on Linux, regardless of whether it's your development or production environment. There are three challenges you'll need to overcome:

1. You're probably most used to installing packages on a Mac or Windows computer. There are some important differences with Linux and to understand them, you'll need some new vocabulary like binary packages and system libraries.
2. Production jobs are usually run in a throwaway container. That means packages are installed every time your production job runs and the speed of package installation becomes more much important than in your development environment.
3. You want to make sure that you're installing exactly the same package versions on your development and production environments.

We'll tackle each of those challenges in this chapter. But if you're already familiar with the problems and just want to hear the solutions, there are two many takeaways from this chapter:

Isolated project environment using `renv`

Most commonly used R 📦 to create isolated project environments.

- You should be using renv.
- Making your R project future-proof with renv.

renv 0.17.3 Get started Reference **Articles** Changelog Search for

Anatomy of a Lockfile

Source: vignettes/lockfile.Rmd

`renv` uses a **lockfile** to capture the state of your library at some point in time. It is stored as a collection of *records*, with different records defining:

- The version of `renv` used when generating the lockfile;
- The version of `R` used in that project;

R "virtual environment" using `renv`

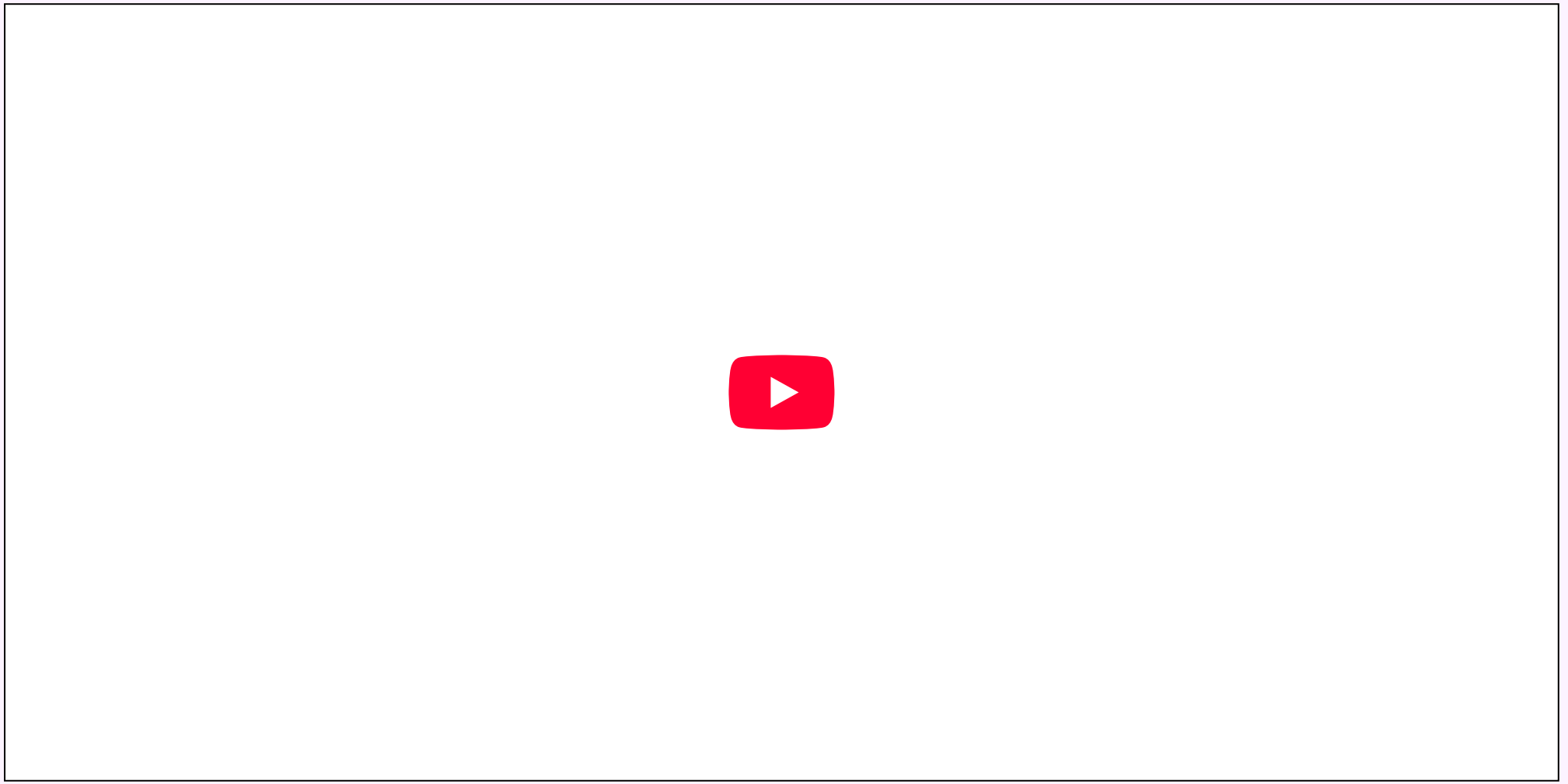
Some advice ...

- In an existing project, use `renv::init(bare = TRUE)` to initiate renv with an empty R library and then install R 📦 manaully.
- After installing R 📦 `pak` in the `renv` environment, set `RENV_CONFIG_PAK_ENABLED=TRUE` in the `.Renviron` file for `renv::install()` to use R 📦 `pak` at the backend to install R 📦.
- Indicate folders/files in the `.renvignore` file to ignore to speed up the snapshot process (`renv::snapshot`)
- You can update the repositories specified in the `renv.lock` file.
    - Shannon Pileggi's blog on `renv::restore()`.

“virtual environment” using `renv`

If you are frustrated about renv::restore() … please

watch Practical {renv} and read Practical {renv} Materials



New Kid for R 📦 managment Require

Require

`Require` is a single package that combines features of `base::install.packages`, `base::library`, `base::require`, as well as `pak::pkg_install`, `remotes::install_github`, and `versions::install_version`, plus the snapshotting capabilities of `renv`. It takes its name from the idea that a user could simply have one line like this:

```
Require(c("dplyr", "lmer", "PredictiveEcology/LandR@development"))
```

named after the `require` function, that would load packages. But with `Require`, it will also install the packages, if necessary. Set it and forget it. This makes if *very clear* what packages are being used in a project. `Require` also continues to work, even if packages are taken off CRAN. This means that even if there is a dependency that is removed from CRAN ("archived"), the line will still work.

Links

View on CRAN

Browse source code

Report a bug

License

GPL-3

Community

Contributing guide

Citation

Citing Require

Developers

Personal R Administration

https://rstats-wtf.github.io/wtf-personal-radmin-slides

personal radmin

it works ~~on my machine~~

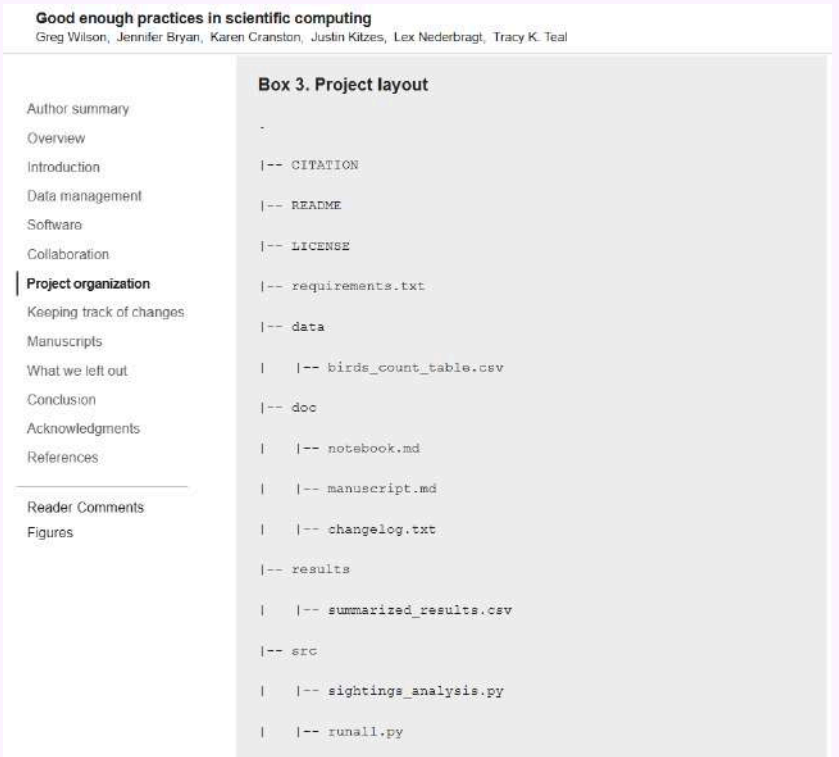
E. David Aja

about

Project Organisation

Organise your project as you go instead of waiting for “tomorrow”.

Data is cheap but time is expensive.



Good enough practices in scientific computing

Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, Tracy K. Teal

Author summary
Overview
Introduction
Data management
Software
Collaboration
Project organization
Keeping track of changes
Manuscripts
What we left out
Conclusion
Acknowledgments
References

Reader Comments
Figures

Box 3. Project layout

```
.
|-- CITATION
|-- README
|-- LICENSE
|-- requirements.txt
|-- data
|   |-- birds_count_table.csv
|-- doc
|   |-- notebook.md
|   |-- manuscript.md
|   |-- changelog.txt
|-- results
|   |-- summarized_results.csv
|-- src
|   |-- sightings_analysis.py
|   |-- runall.py
```

Research Compendium by Scriberia from The Turing Way project and Project Layout from Good enough practices in scientific computing.

Project Organisation

My harmoniation template organisation is based on the R 📦 `rcompendium` but there are others (`orderly`, `prodigenr` and `workflowr`) as well.

rcompendium 1.4

rcompendium

In the area of open science, making reproducible analyses is a strong prerequisite. But sometimes it is difficult 1) to find the good structure to organize files and 2) to set up the whole project. The aim of the package `rcompendium` is to make easier the creation of R package/research compendium (i.e. a predefined files/folders structure) so that users can focus on the code/analysis instead of wasting time organizing files.

A full ready-to-work structure will be set up with the following features:

- Initialization of version control with git.
- Creation of a minimal R package structure (`DESCRIPTION` and

Links

View on CRAN

Browse source code

Report a bug

License

Full license

GPL (>= 2)

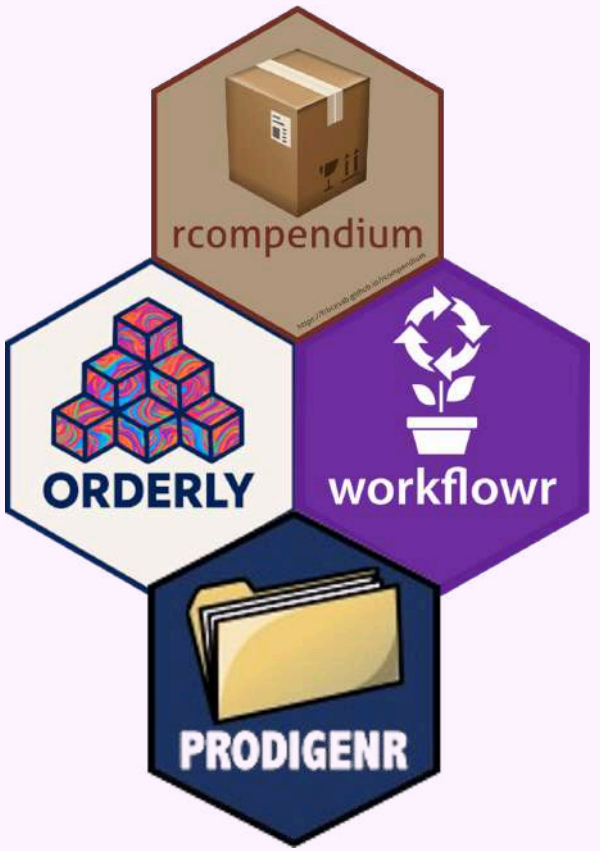
Community

Contributing guide

Code of conduct

Citation

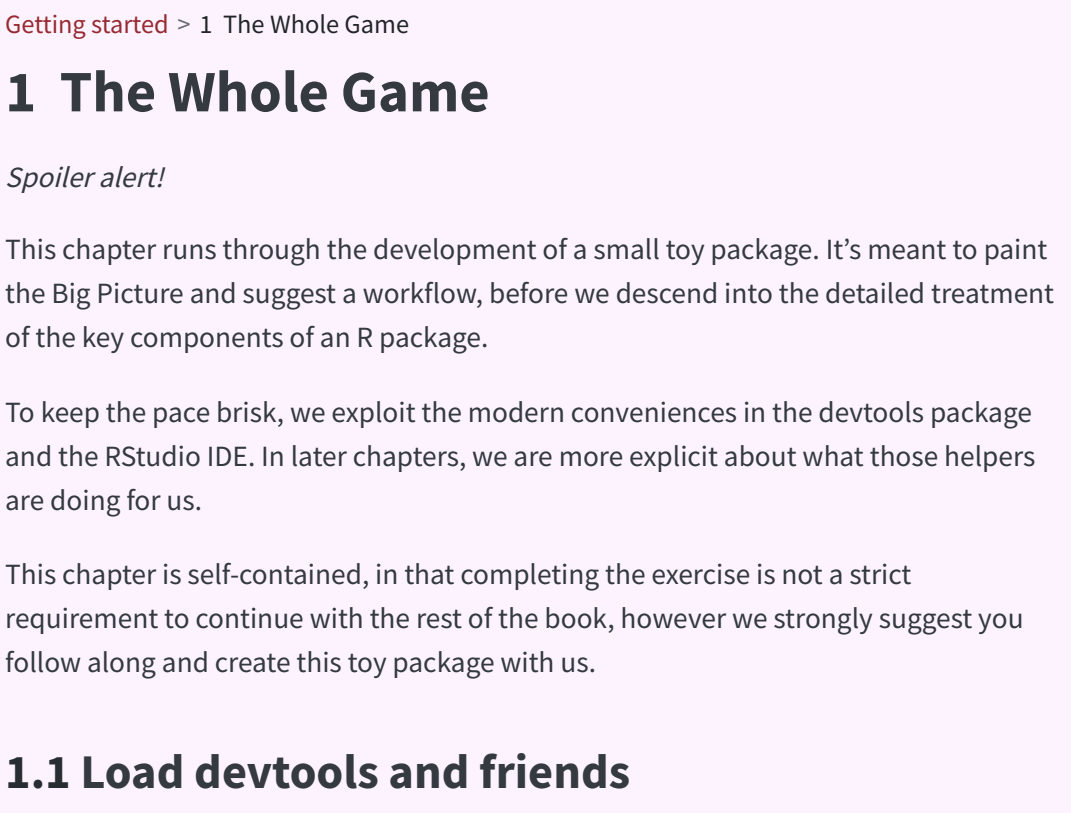
Citing rcompendium



Custom R Function Management

It is better to organise your custom R functions into a R 📦 to make your code easier to re-use, document, and test.

- 📖 R package (2e) Chapter 1: The Whole Game.



Getting started > 1 The Whole Game

1 The Whole Game

Spoiler alert!

This chapter runs through the development of a small toy package. It's meant to paint the Big Picture and suggest a workflow, before we descend into the detailed treatment of the key components of an R package.

To keep the pace brisk, we exploit the modern conveniences in the devtools package and the RStudio IDE. In later chapters, we are more explicit about what those helpers are doing for us.

This chapter is self-contained, in that completing the exercise is not a strict requirement to continue with the rest of the book, however we strongly suggest you follow along and create this toy package with us.

1.1 Load devtools and friends

If the top of your R script is

```r
setwd("C:\Users\jenny\path\that\only\I\have")
rm(list = ls())
```

Jenny will come into your your office and SET YOUR COMPUTER ON FIRE 🔥 .

https://tidyverse.org/blog/2017/12/workflow-vs-script/

Project-oriented workflow

Photo by secumem

Practise "safe paths"

 with file system functions (`fs` and `here`).

fs 1.6.6

fs

fs provides a cross-platform, uniform interface to file system operations. It shares the same back-end component as nodejs, the libuv C library, which brings the benefit of extensive real-world use and rigorous cross-platform testing. The name, and some of the inter-

here 1.0.2

here

The goal of the here package is to enable easy file referencing in project-oriented workflows. In contrast to using `setwd()`, which is fragile and dependent on the way you organize your files, here uses the top-level directory of a project to easily build paths to files.

Installation

Practise “safe paths”

❌ Avoid typing absolute path in R script. Let R do it for you.

```
BAD <- "D://Jeremy//PortableR//RPortableWorkDirectory//"
```

User’s home directory

```
fs::path_home()
```

```
C:/Users/Jeremy
```

R Project directory

```
here::here()
```

```
[1] "D:/Jeremy/PortableR/RPortableWorkDirectory/hat_2025"
```

`here::here()` does not create directories; that’s your job.

Practise "safe paths"

❌ Avoid typing / or \ manually. Let R do it for you.

```r
file.path("data", "raw-data.csv")
```

```
[1] "data/raw-data.csv"
```

```r
fs::path_home("data", "raw-data.csv")
```

```
C:/Users/Jeremy/data/raw-data.csv
```

```r
here::here("data", "raw-data.csv")
```

```
[1] "D:/Jeremy/PortableR/RPortableWorkDirectory/hat_2025/data/raw-data.csv"
```

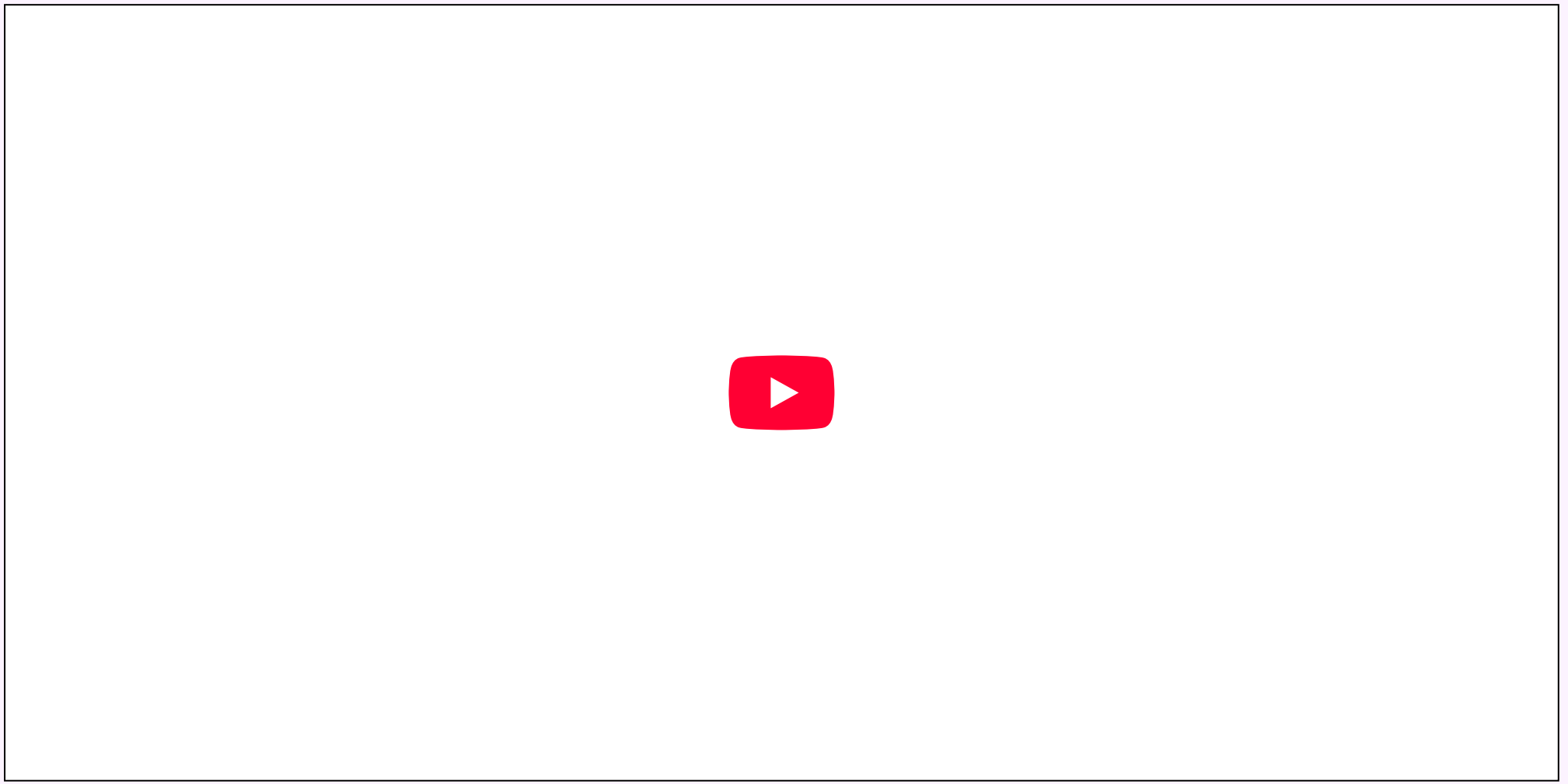
✅ Use relative path within the R project directory.

```r
readxl::read_excel(path = here::here("data-folder", "data.xlsx"))
ggplot2::ggsave(filename = here::here("figs", "built-barchart.png"))
```

Works on my machine, works on yours!

Practise “safe paths”

Efficient File Management in R with {fs} with Jadey Ryan



Start R Session in a "blank slate"

❌ Avoid `rm(list = ls())`

Which persist after rm(list = ls())?

Option	Persists?
A. library(dplyr)	✓
B. summary <- head	✗
C. options(stringsAsFactors = FALSE)	✓
D. Sys.setenv(LANGUAGE = "fr")	✓
E. x <- 1:5	✗
F. attach(iris)	✓

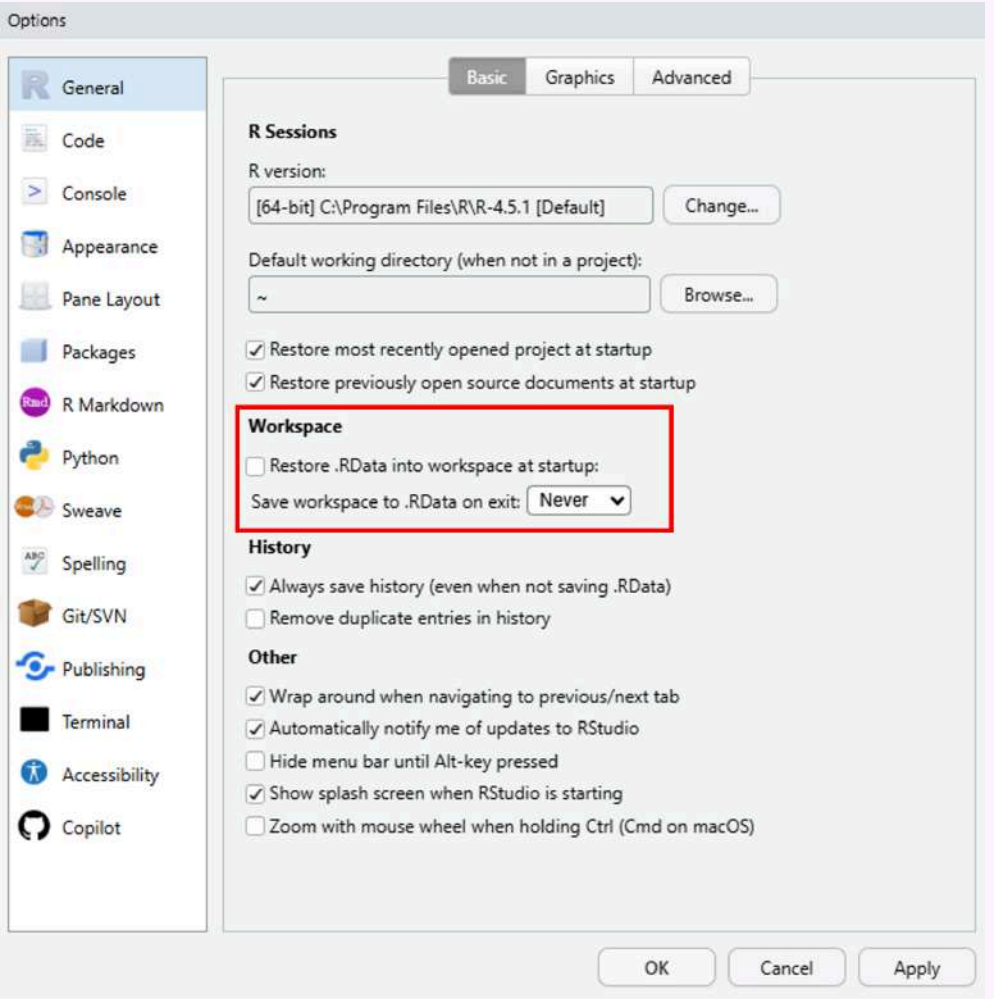
Slide from Project oriented workflows.

✅ Use `usethis::use_blank_slate()`

R console

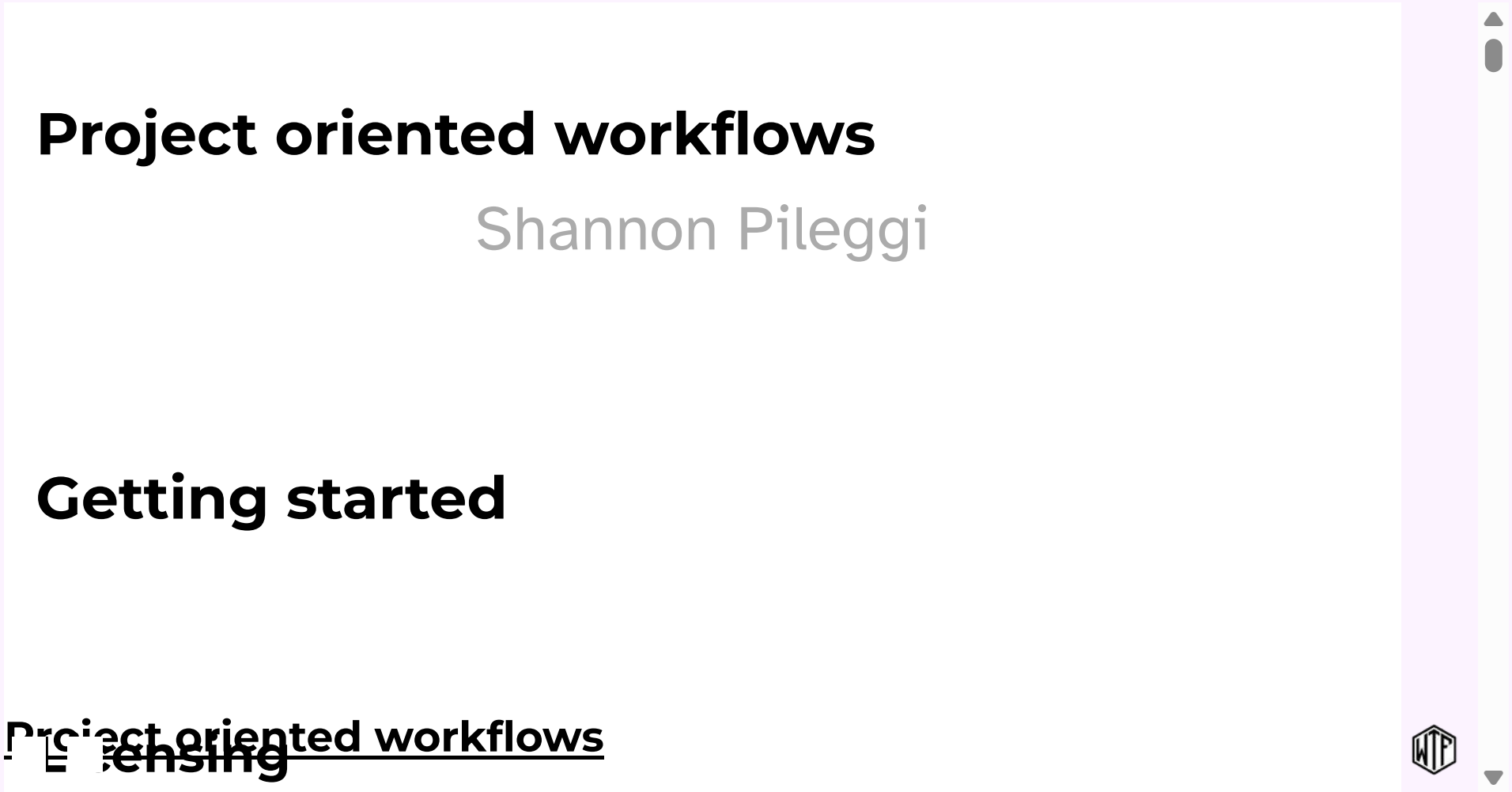
```
usethis::use_blank_slate()
```

✅ Tools -> Global Options in RStudio.



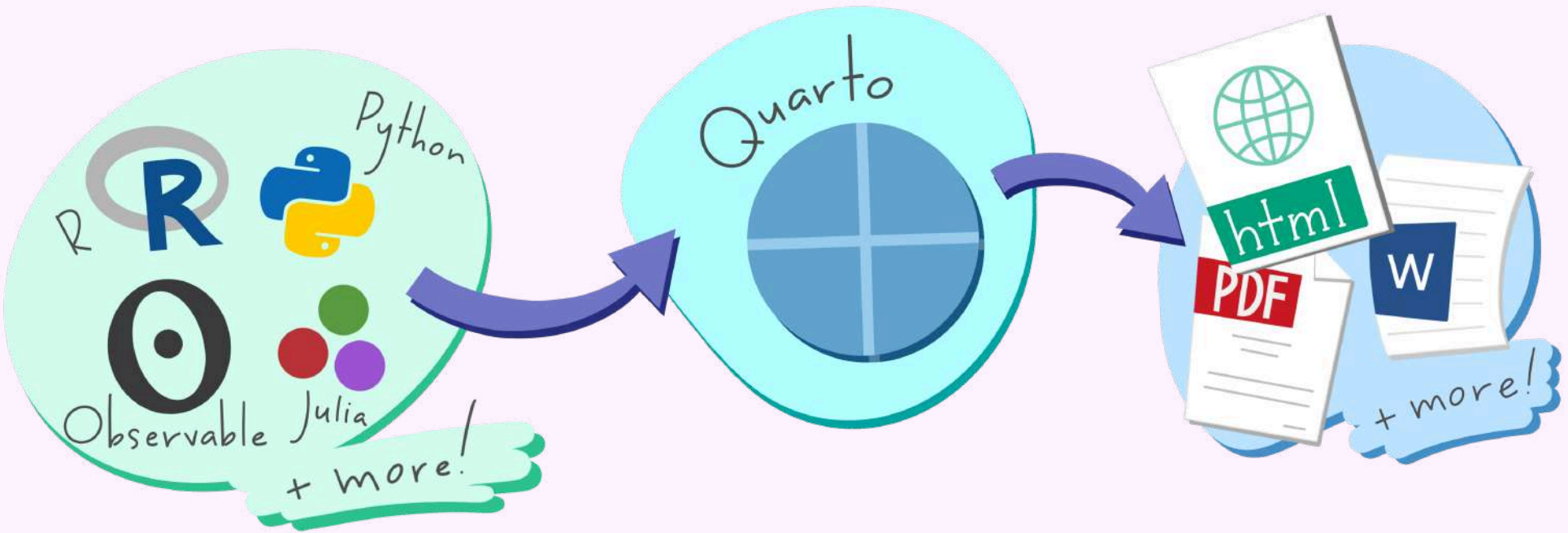
Project Oriented Workflow

https://rstats-wtf.github.io/wtf-project-oriented-workflow-slides



Quarto

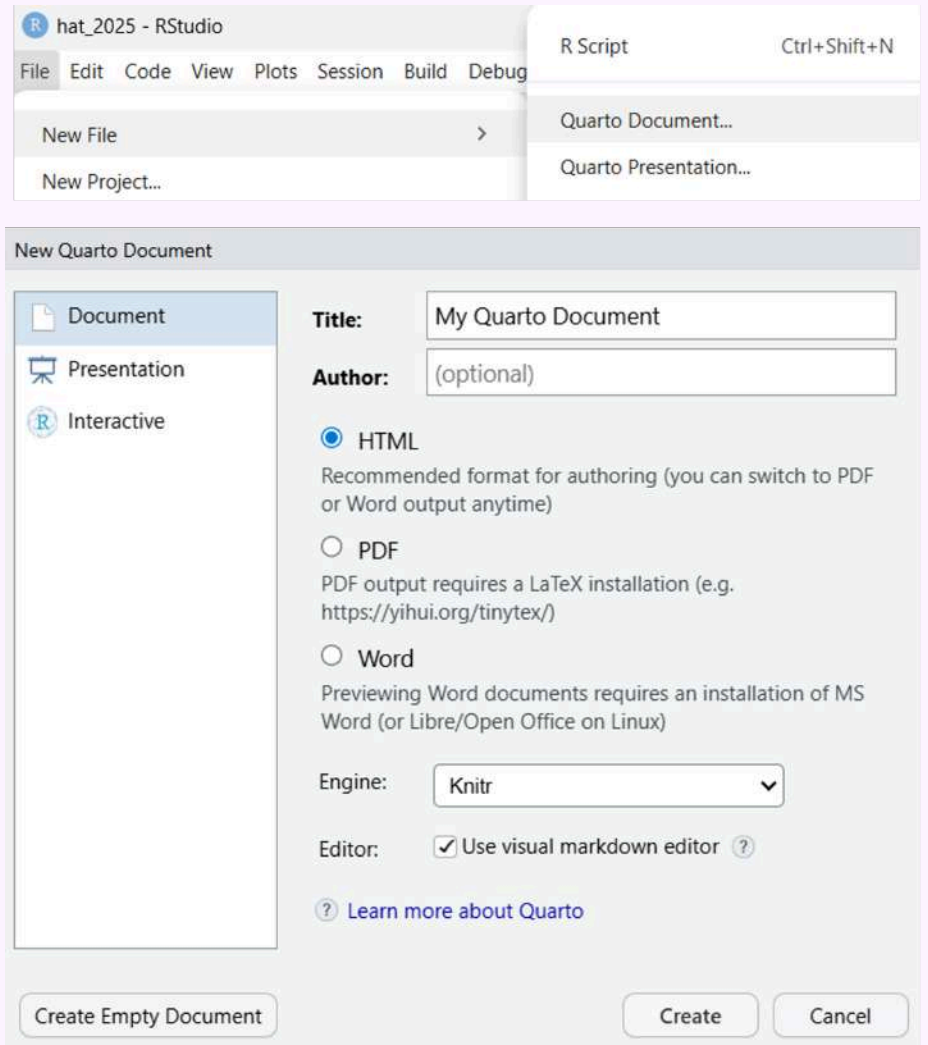
Quarto is an open-source software that weaves narrative and programming code together to produce elegantly formatted output as documents (in HTML, Word, PDF), presentations, books, web pages, and more.

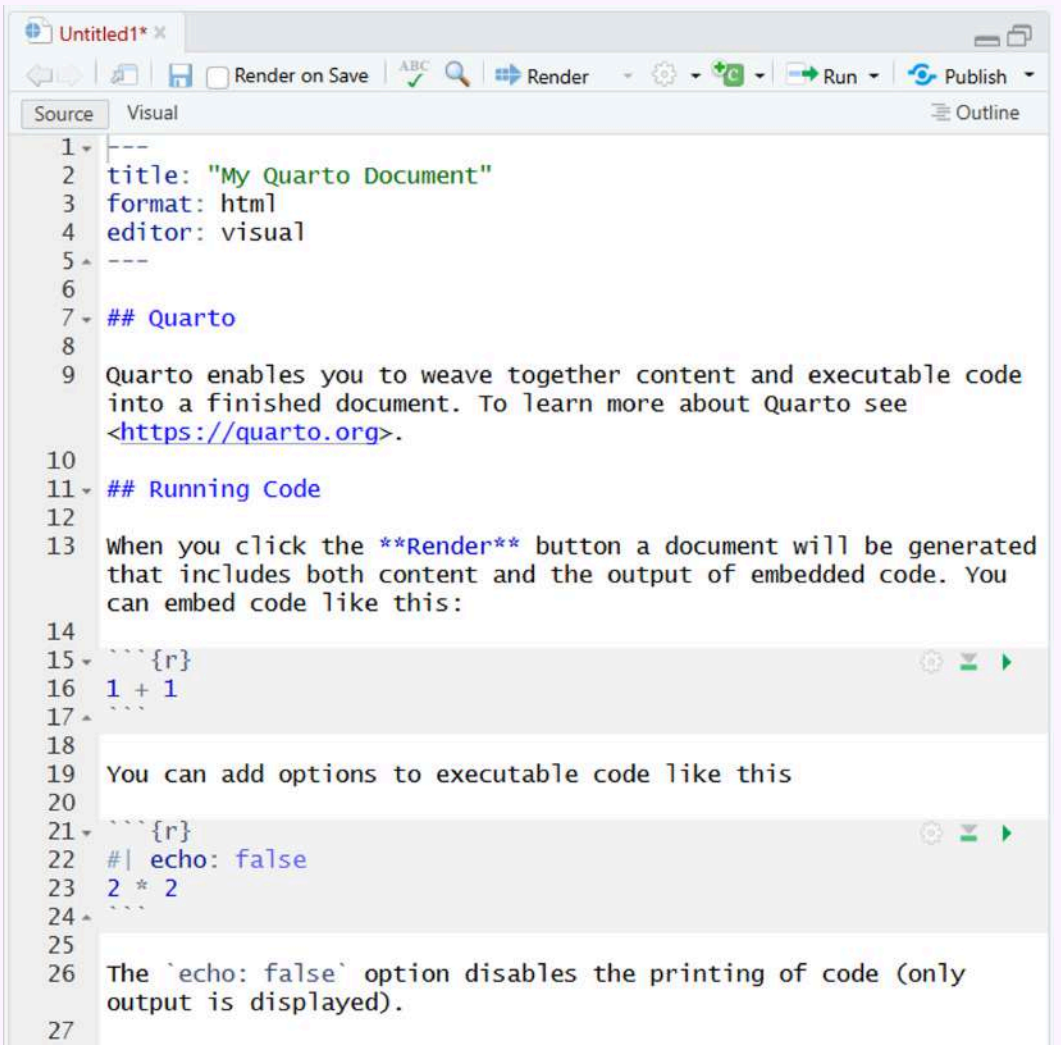


Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

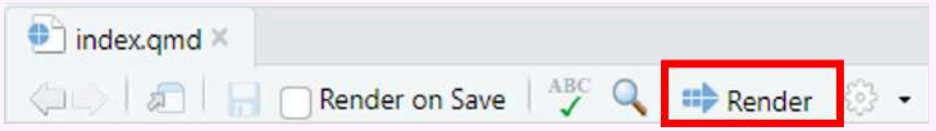
Open A Quarto Document

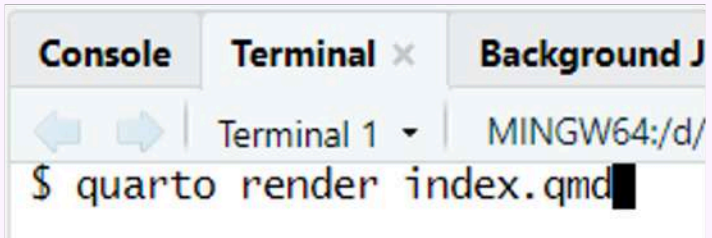
Create a Quarto document

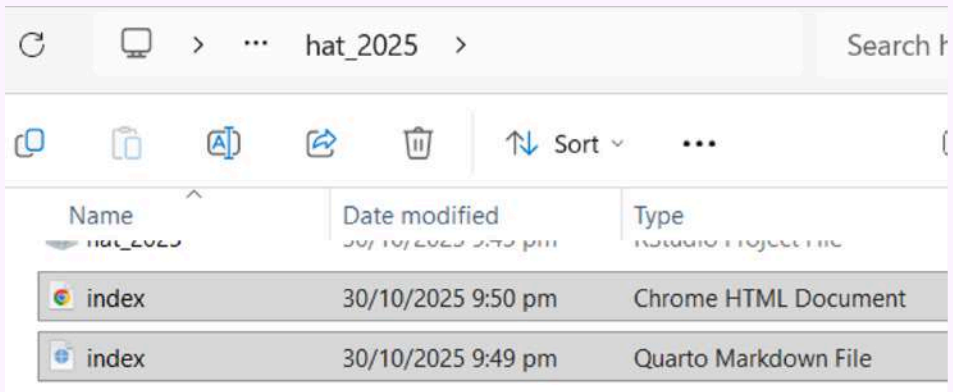




Render to HTML Report







My Quarto Document

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

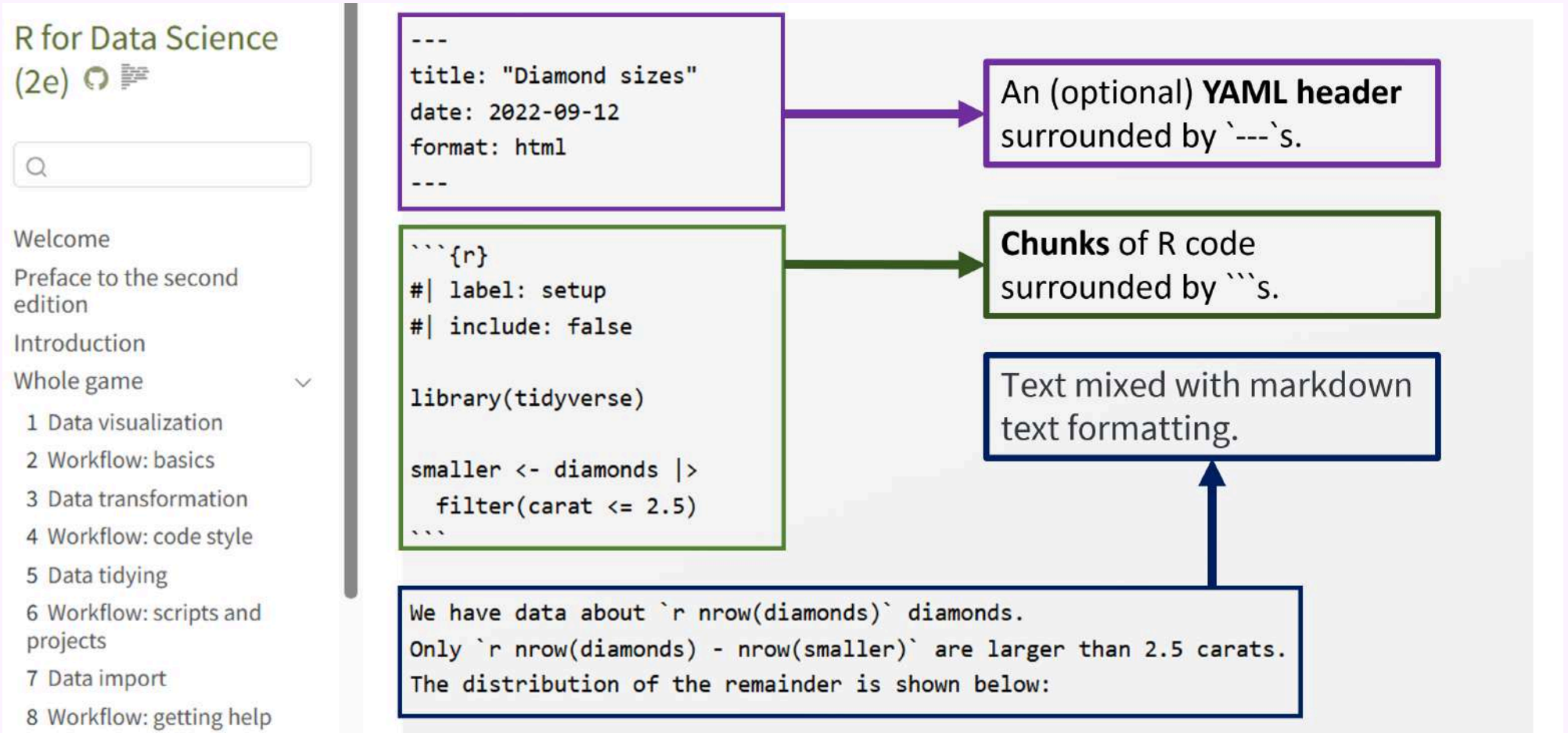
You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

Quarto Level 1

A Quarto file is a plain text file that has the extension `.qmd` containing three important types of content:



Simple Quarto file example from R for Data Science (2e) Chapter 28 Quarto

Quarto Level 1

https://kpuka.ca/resources/quarto_intro.html

Introduction to Quarto

1 Introduction

Quarto enables you to weave together content and executable code into a finished document. Quarto is a multi-language program, supporting multiple types of inputs languages (e.g., R, python, HTML), input software (e.g., RStudio, VScode, plain text), and outputs/documents (e.g., HTML document, presentation slides, PDFs, word documents).

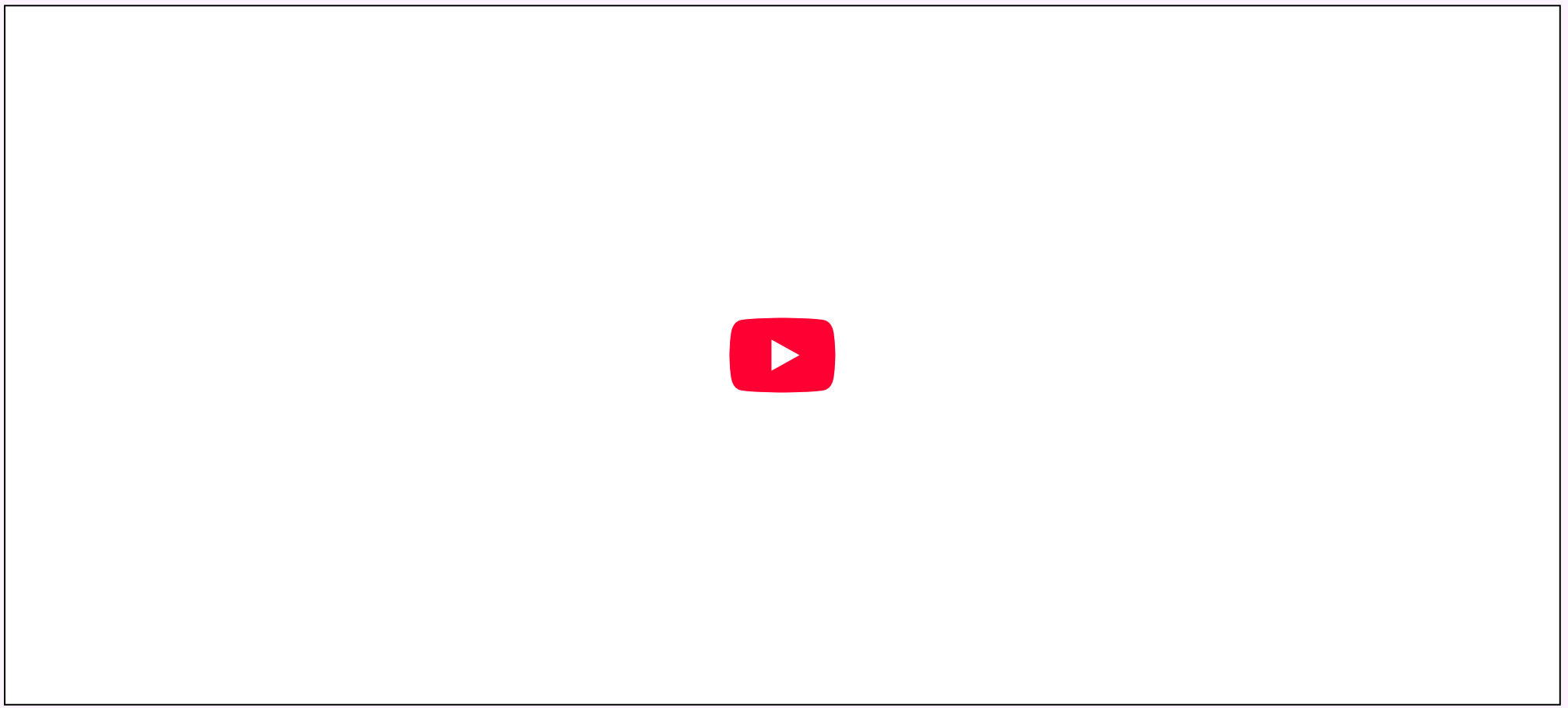
Quarto is the next generation of R Markdown, and is able to render most existing R Markdown (Rmd) files without modification. Most of the information presented below will work both in Quarto and Markdown. Syntax that is common to both will be noted as 'Syntax (Input)', whereas

Quarto Level 2

terminal

```
quarto pandoc -o custom-reference-doc.docx --print-default-data-file reference.docx
```

[How to change document fonts & formats in Quarto (Word/Docx)]



Quarto Level 2

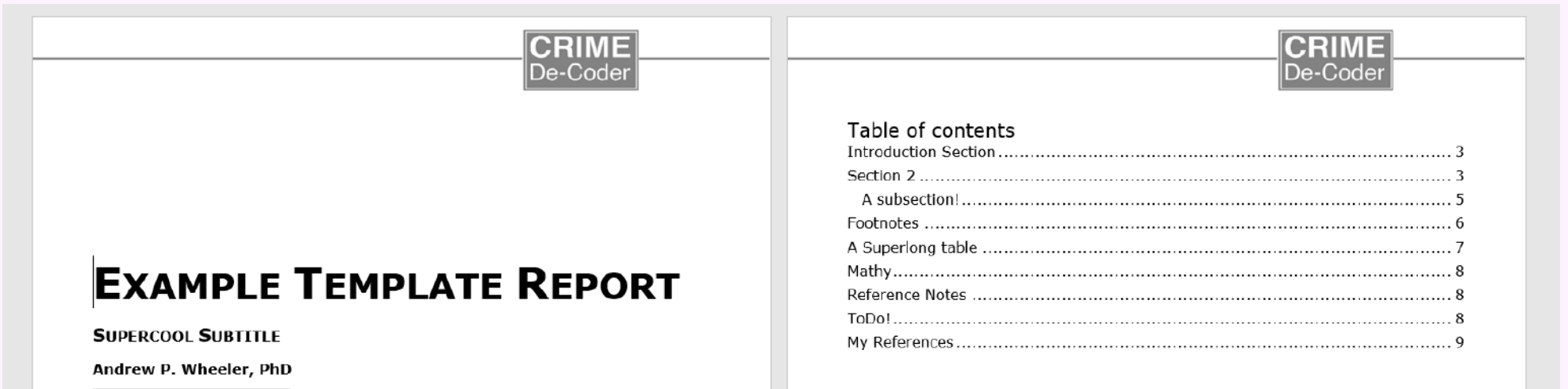
My word template for Quarto

My word template for Quarto

I have posted **on Github my notes on creating a word template to use with quarto**. And since Quarto is just feeding into pandoc, those who are just using pandoc (so not doing intermediate computations), should maybe find that template worthwhile as well.

So first, why word? Quarto by default looks pretty nice for HTML. That is fine for them to prioritize that, but the majority of reports I want to use quarto for HTML is not the best format. Many times I want a report that can be emailed in PDF and/or printed. And sometimes I (or my clients) want a semi-automated report that can be edited after the fact. In those cases word is a good choice.

Editing LaTeX is too hard, and I am pretty happy with the this template for small reports. I will be sharing my notes on **writing my python book** in Quarto soonish, but for now wanted to share how I created a word template.

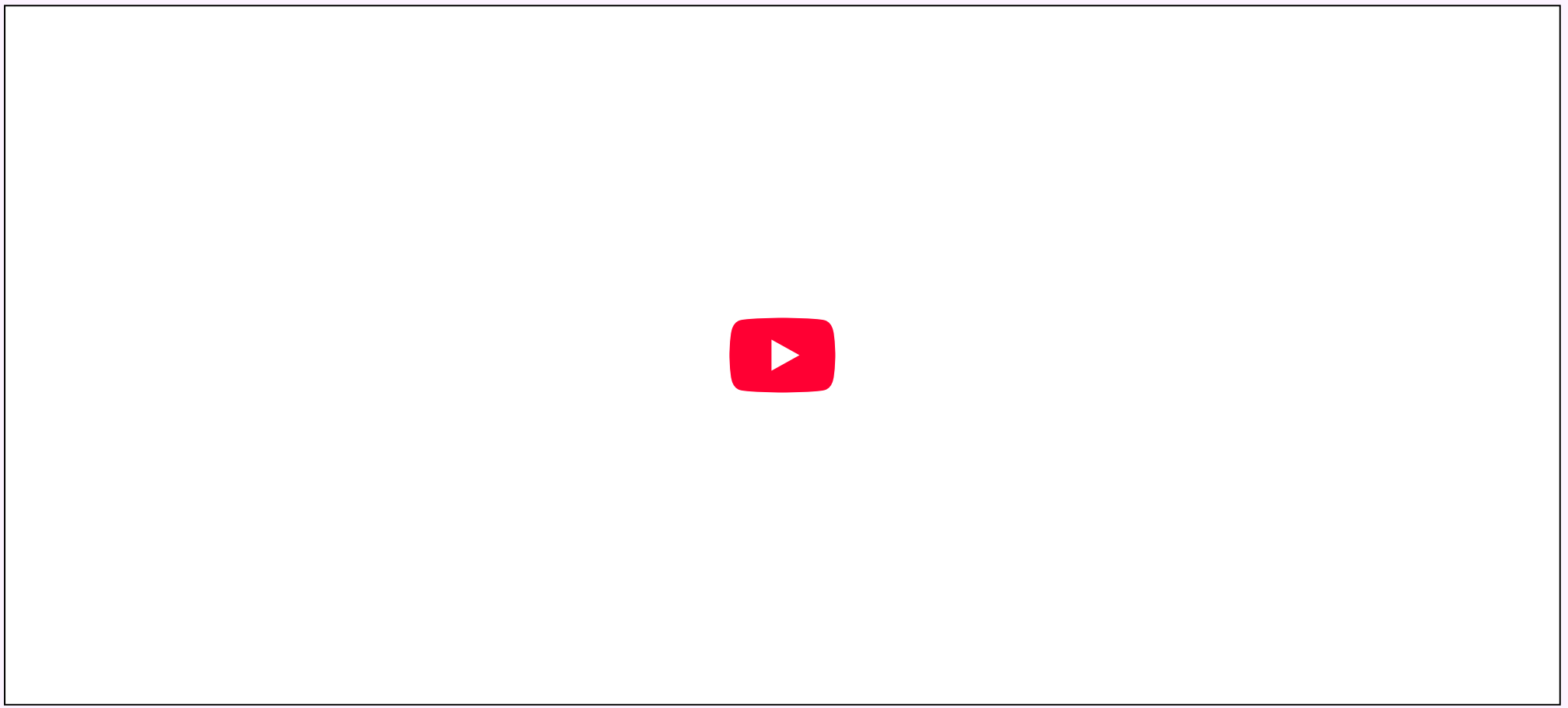


Quarto Level 2

terminal

```
quarto install tinytex
```

Preparing RStudio to Generate PDF Files with Quarto and tinyTeX



Quarto Level 2

Christopher Kenny's Quarto templates

Quarto Extensions

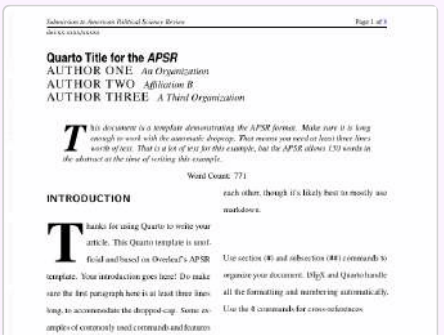
Templates and filters for reproducible reports made (mostly) for Quarto

PUBLISHED

November 12, 2025

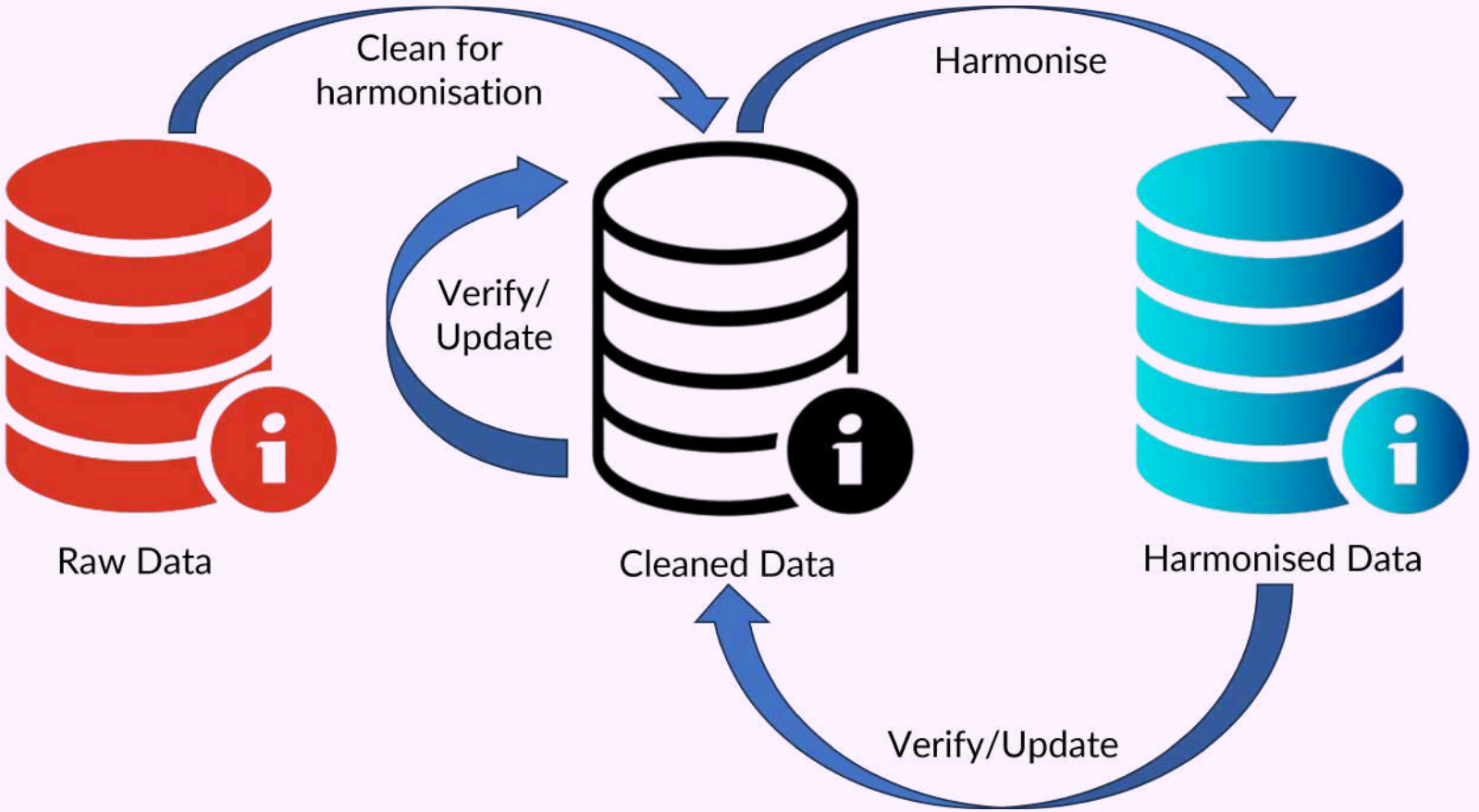
Journal templates

Templates for general science and social science journals.



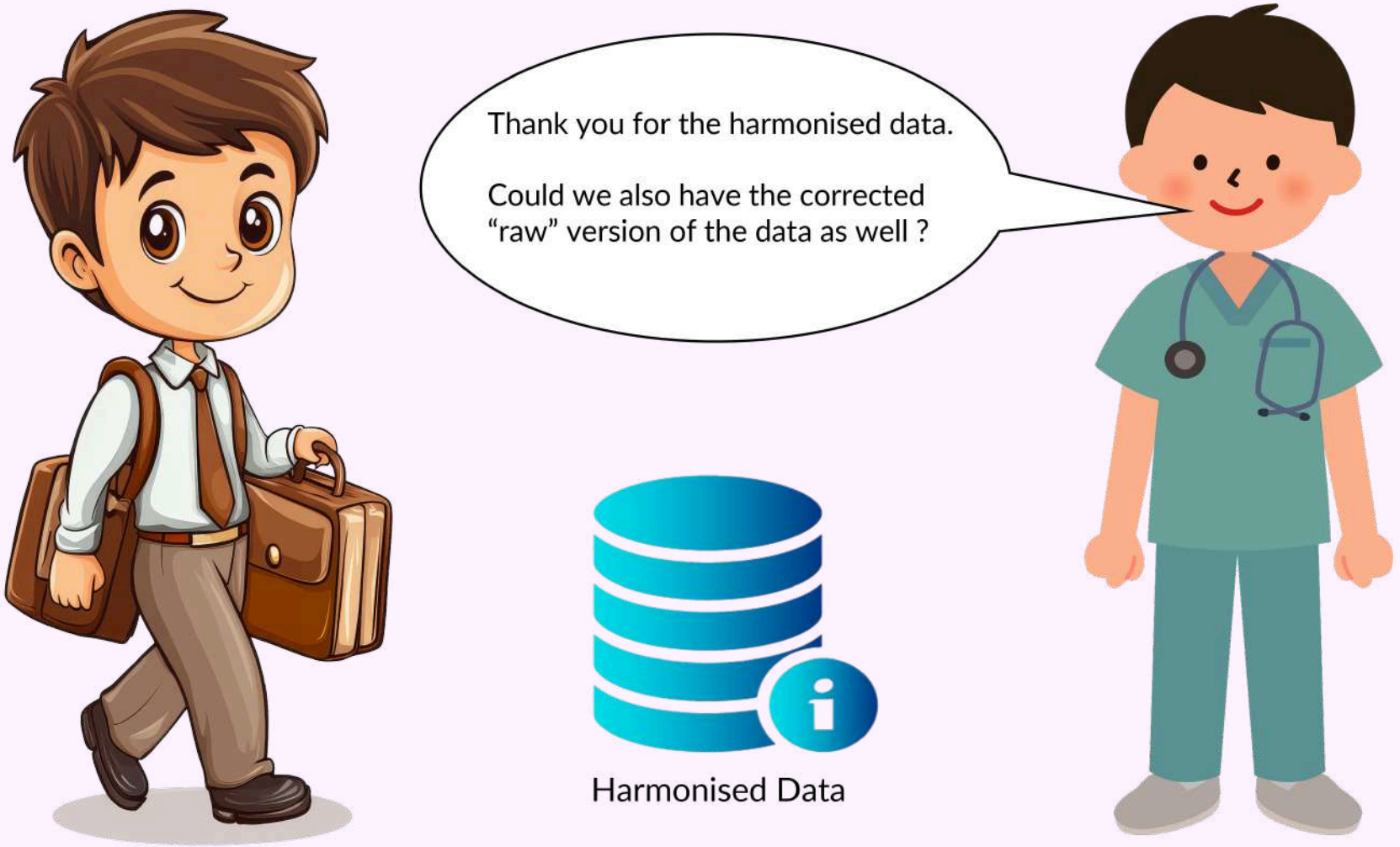
Workflow with collaborators

Collaborator can send the raw data once and you keep updating the cleaned data for harmonsation.



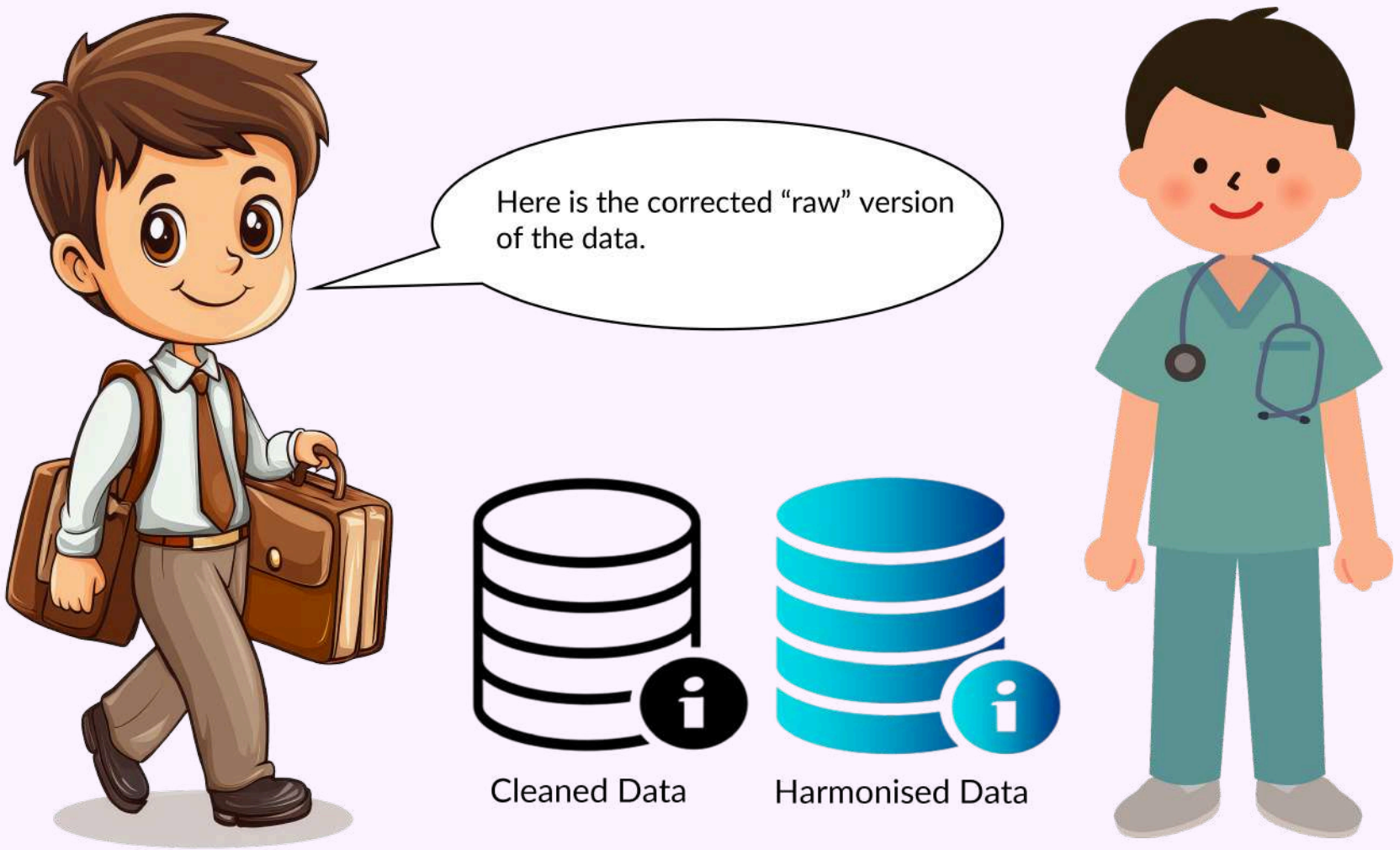
Dataset vectors by Vectora Artworks from Vecteezy.

Workflow with collaborators



Cheerful Businessman designed by Iftikhar Alam from Vecteezy and Medical Doctor Man from Creazilla.

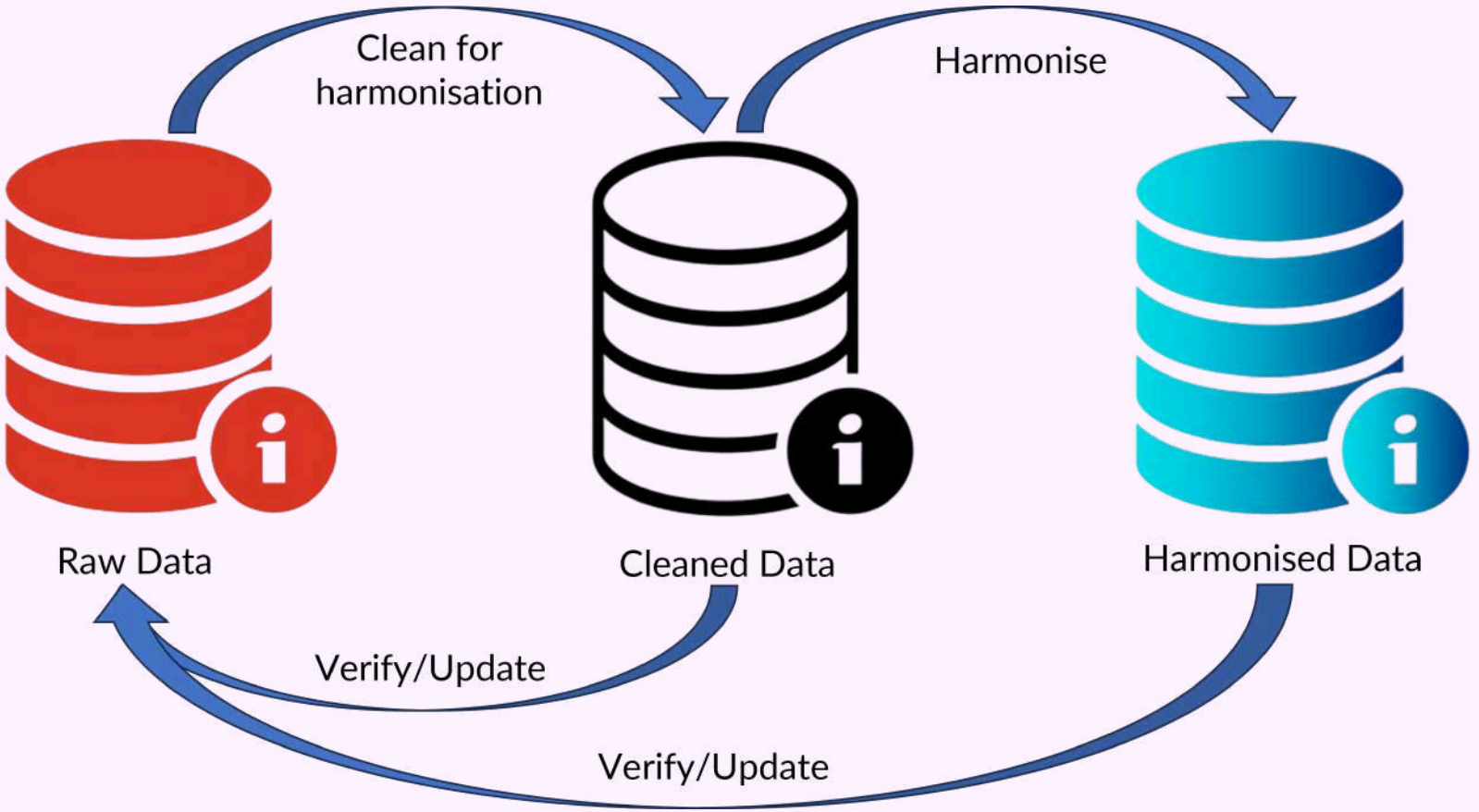
Workflow with collaborators



Cheerful Businessman designed by Iftikhar Alam from Vecteezy and Medical Doctor Man from Creazilla.

Workflow with collaborators

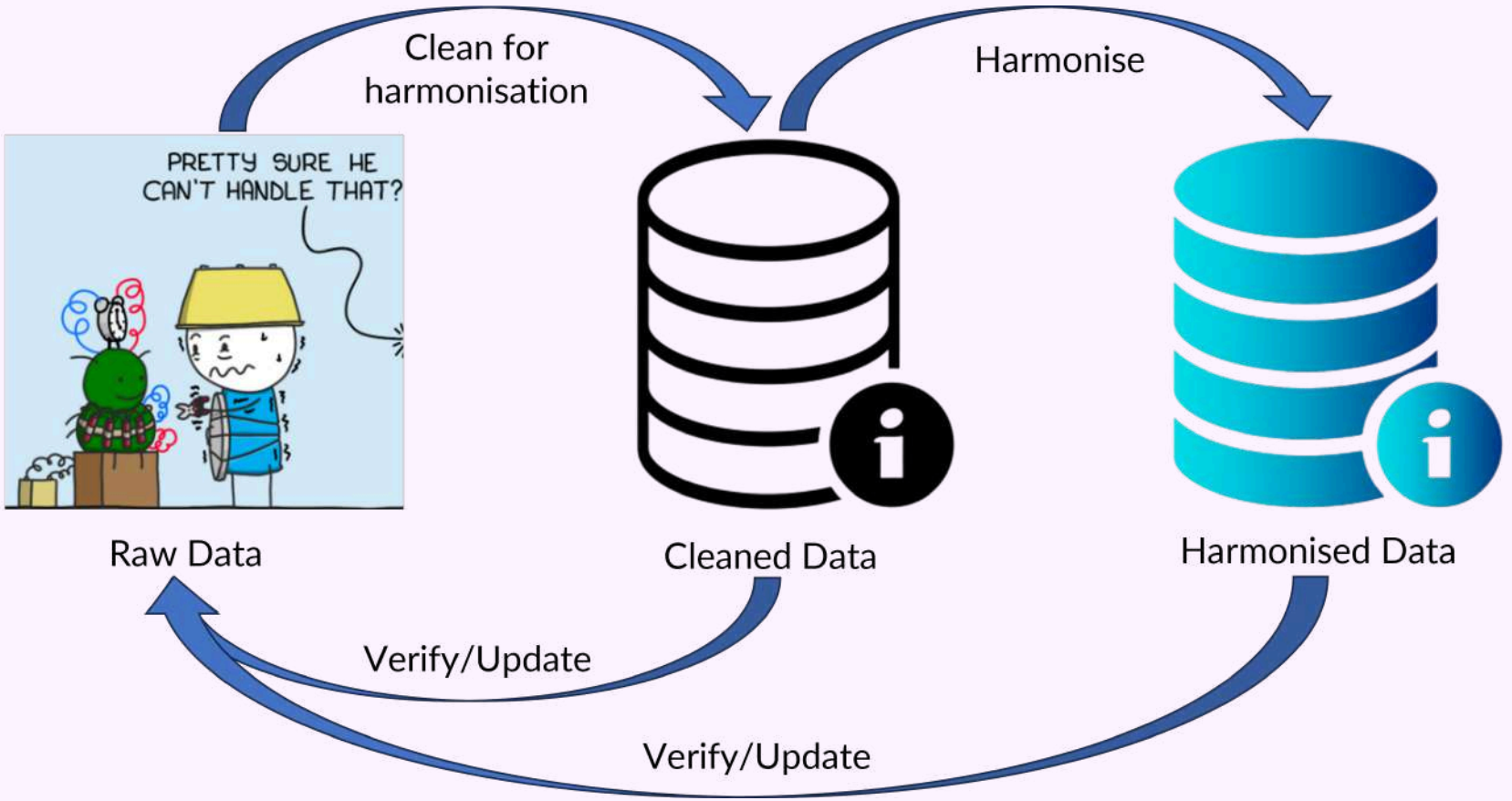
Collaborator can update the raw data. For example, adding new clinical data, add more patients, correct errors.



Dataset vectors by Vectora Artworks from Vecteezy.

Workflow with collaborators

New version means new bugs or reopen issues to fix. Is there an automated way to catch warnings/issues when reading these updated files ?



Dataset vectors by Vectora Artworks from Vecteezy. Anticipate from MonkeyUser.com

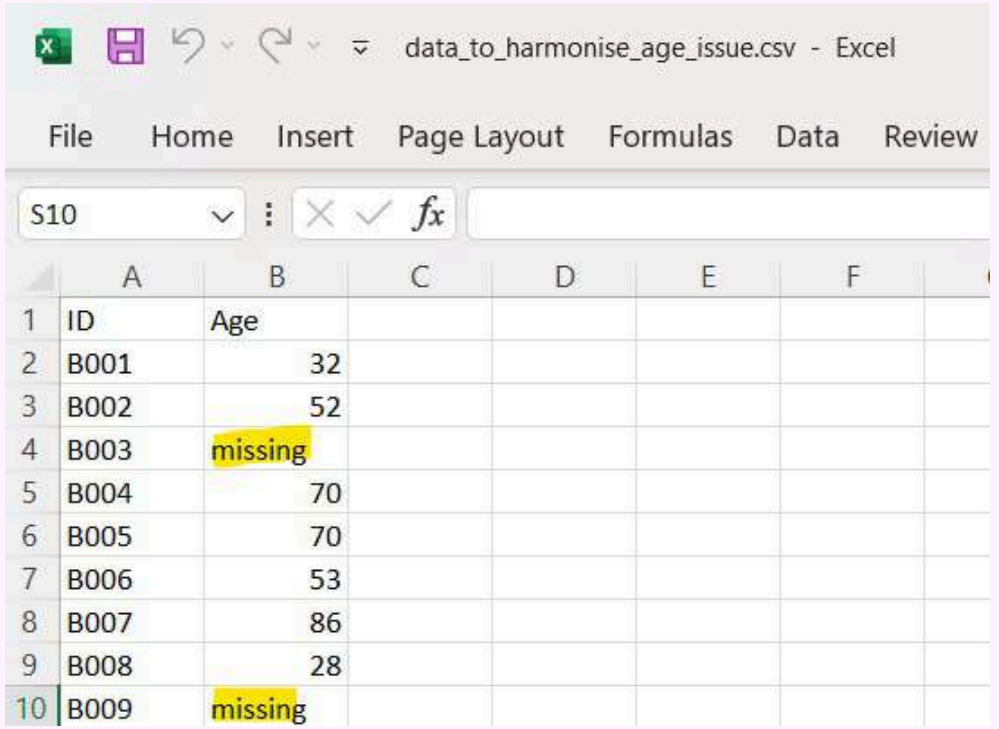
Automated capture of warnings (csv)

Is there an automated way to catch warnings/issues when reading csv files ?

```r
cohort_data_csv <- vroom::vroom(
  file = here::here("data-raw", "Cohort_csv",
    "data_to_harmonise_age_issue.csv"),
  delim = ",",
  col_select = 1:2,
  show_col_types = FALSE,
  col_types = list(
    ID = vroom::col_character(),
    Age = vroom::col_integer()
  )
)
```

```r
head(cohort_data_csv, n = 3)
```

```
# A tibble: 3 × 2

Warning: One or more parsing issues, call `problems()` on your
data frame for details,
e.g.:
  dat <- vroom(...)
  problems(dat)

  ID      Age
  <chr> <int>
1 B001     32
2 B002     52
3 B003     NA
```



Automated capture of warnings (csv)

If there are issues with the data, the output of `readr::problems` will be a tibble.

```
cohort_data_csv |>
  vroom::problems()
```

```
# A tibble: 4 × 5
    row   col expected   actual  file
  <int> <int> <chr>      <chr>   <chr>
1     2     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/hat…
2     4     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/hat…
3    10     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/hat…
4    17     2 an integer missing D:/Jeremy/PortableR/RPortableWorkDirectory/hat…
```

To check for this automatically, we can use `pointblank::expect_row_count_match`.

```
cohort_data_csv |>
  vroom::problems() |>
  pointblank::expect_row_count_match(count = 0)
```

```
Error: Row counts for the two tables did not match.
The `expect_row_count_match()` validation failed beyond the absolute threshold level (1).
* failure level (1) >= failure threshold (1)
```

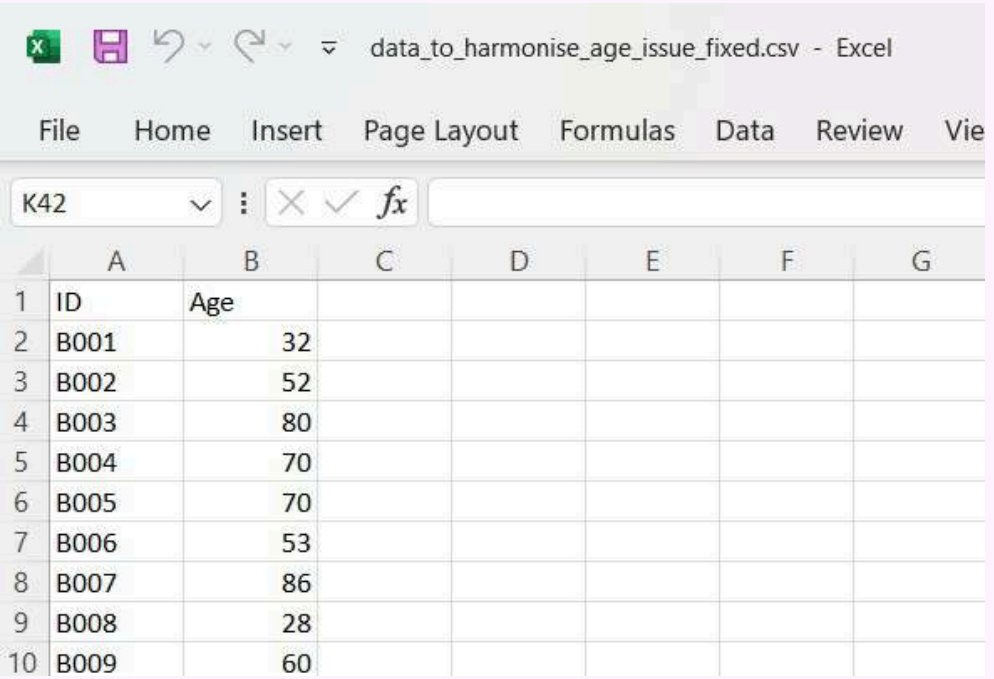
Automated capture of warnings (csv)

Here is a case with no issues.

```r
cohort_data_csv <- vroom::vroom(
  file = here::here("data-raw", "Cohort_csv",
    "data_to_harmonise_age_issue_fixed.csv"),
  delim = ",",
  col_select = 1:2,
  show_col_types = FALSE,
  col_types = list(
    ID = vroom::col_character(),
    Age = vroom::col_integer()
  )
)

cohort_data_csv |>
  vroom::problems()
```

```
# A tibble: 0 × 5
# ℹ 5 variables: row <int>, col <int>, expected <chr>, actual
<chr>, file <chr>
```

```r
cohort_data_csv |>
  vroom::problems() |>
  pointblank::expect_row_count_match(count = 0)
```



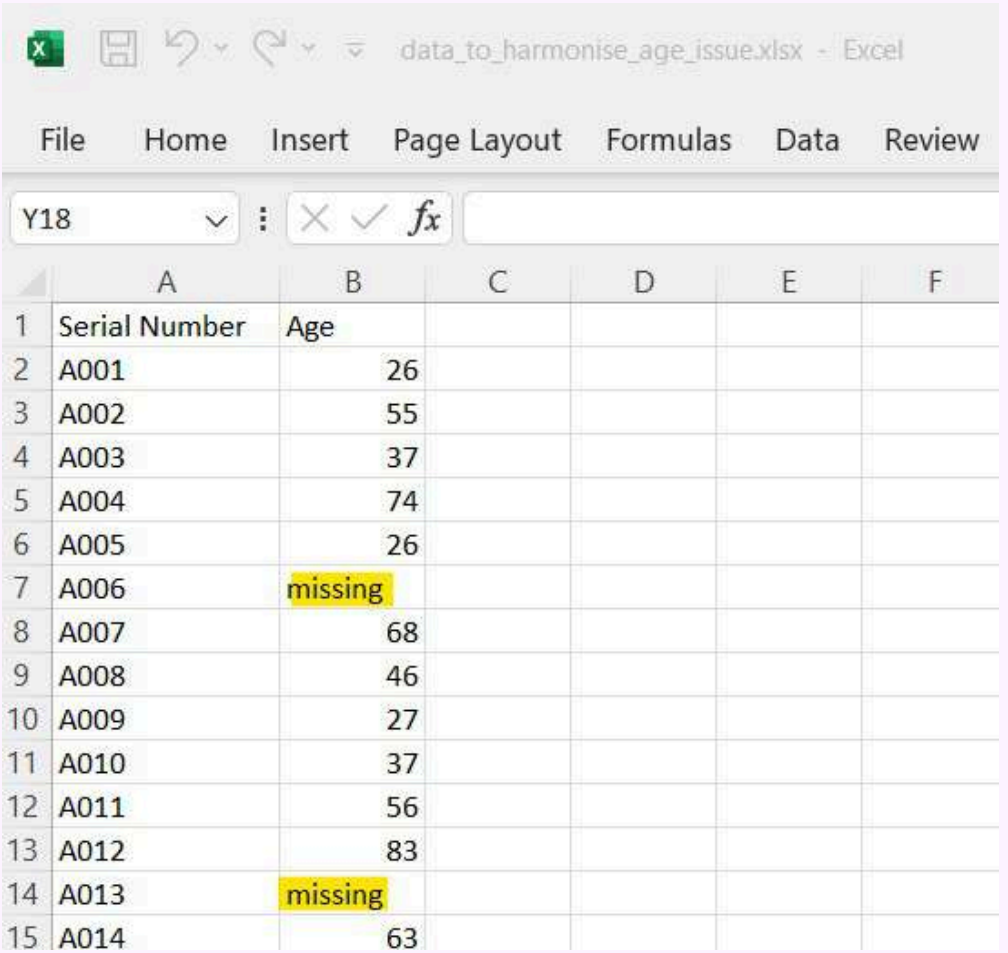
	A	B	C	D	E	F	G
1	ID	Age					
2	B001	32					
3	B002	52					
4	B003	80					
5	B004	70					
6	B005	70					
7	B006	53					
8	B007	86					
9	B008	28					
10	B009	60					

Automated capture of warnings (Excel)

Is there an automated way to catch warnings/issues when reading Excel files ?

```r
cohort_data_excel <- readxl::read_excel(
  path = here::here("data-raw", "Cohort_Excel",
    "data_to_harmonise_age_issue.xlsx"),
  sheet = "Sheet1",
  col_types = c(
    "text", "numeric"
    )
  )
```

```
Warning: Expecting numeric in B7 / R7C2: got 'missing'
Warning: Expecting numeric in B14 / R14C2: got 'missing'
```



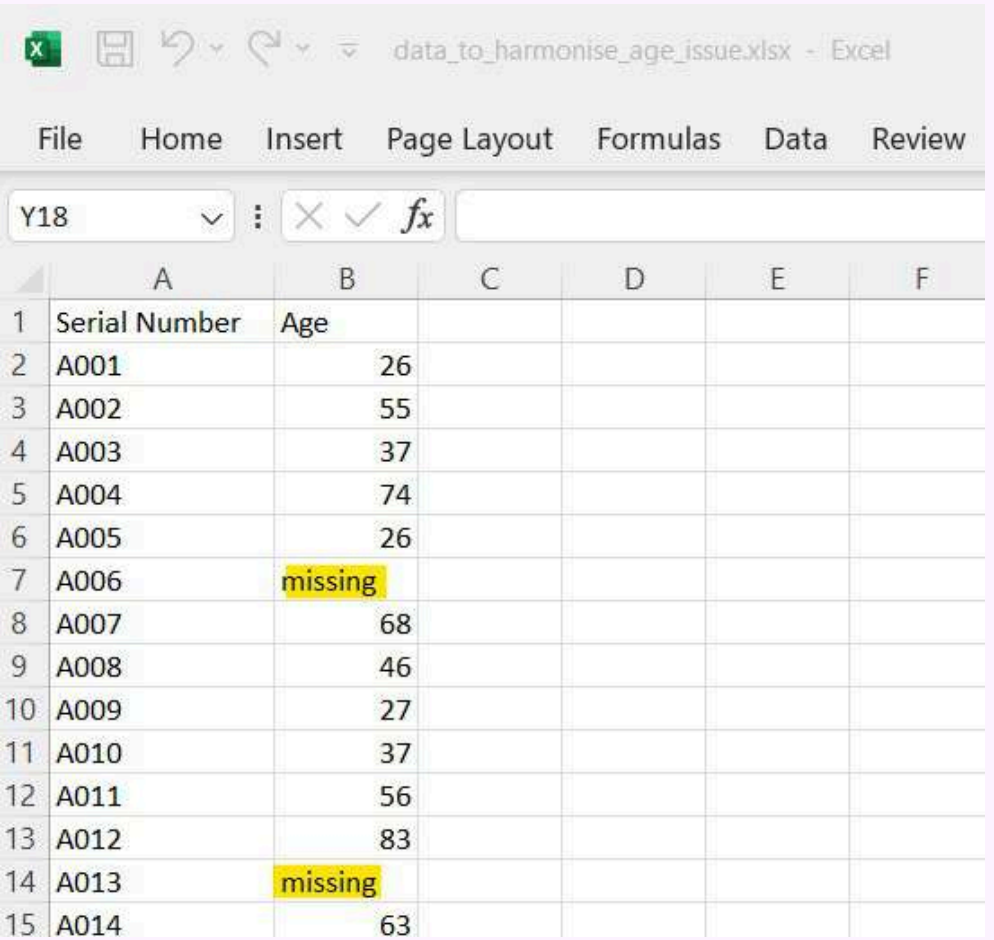
	A	B	C	D	E	F
1	Serial Number	Age				
2	A001	26				
3	A002	55				
4	A003	37				
5	A004	74				
6	A005	26				
7	A006	missing				
8	A007	68				
9	A008	46				
10	A009	27				
11	A010	37				
12	A011	56				
13	A012	83				
14	A013	missing				
15	A014	63				

Automated capture of warnings (Excel)

We can read the Excel file with testthat::expect_no_condition.

```
testthat::expect_no_condition(
  cohort_data_excel <- readxl::read_excel(
    path = here::here("data-raw", "Cohort_Excel",
      "data_to_harmonise_age_issue.xlsx"),
    sheet = "Sheet1",
    col_types = c("text", "numeric")
  )
)
```

```
Error: Expected `... <- NULL` to run without any conditions.
i Actually got a <simpleWarning> with text:
  Expecting numeric in B7 / R7C2: got 'missing'
```

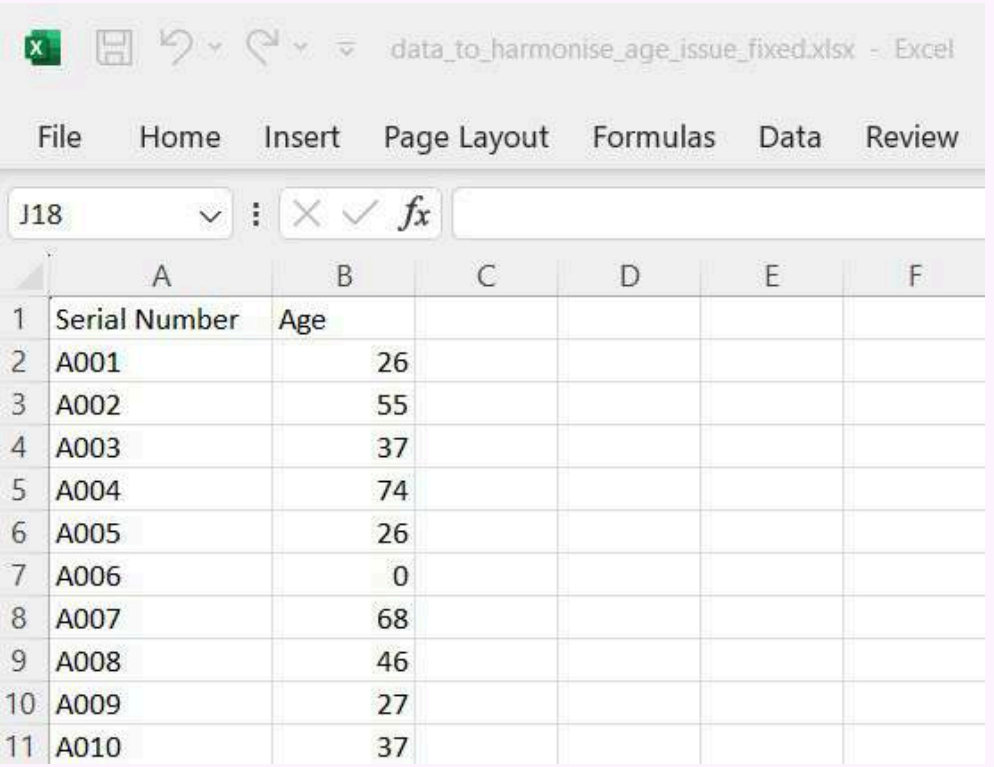


	A	B	C	D	E	F
1	Serial Number	Age				
2	A001	26				
3	A002	55				
4	A003	37				
5	A004	74				
6	A005	26				
7	A006	missing				
8	A007	68				
9	A008	46				
10	A009	27				
11	A010	37				
12	A011	56				
13	A012	83				
14	A013	missing				
15	A014	63				

Automated capture of warnings (Excel)

However, this method means that you will lose the pipe workflow.

```r
testthat::expect_no_condition(
  cohort_data_excel <- readxl::read_excel(
    path = here::here("data-raw", "Cohort_Excel",
      "data_to_harmonise_age_issue_fixed.xlsx"),
    sheet = "Sheet1",
    col_types = c("text", "numeric")
  )
)

cohort_data_excel <- cohort_data_excel |>
  # Check if Serial Number is unique
  pointblank::rows_distinct(
    columns = "Serial Number",
  )
```



	A	B	C	D	E	F
1	Serial Number	Age				
2	A001	26				
3	A002	55				
4	A003	37				
5	A004	74				
6	A005	26				
7	A006	0				
8	A007	68				
9	A008	46				
10	A009	27				
11	A010	37				

Automated capture of warnings (Excel)

We can use the tee pipe operator %T>% from magrittr.

With Issues

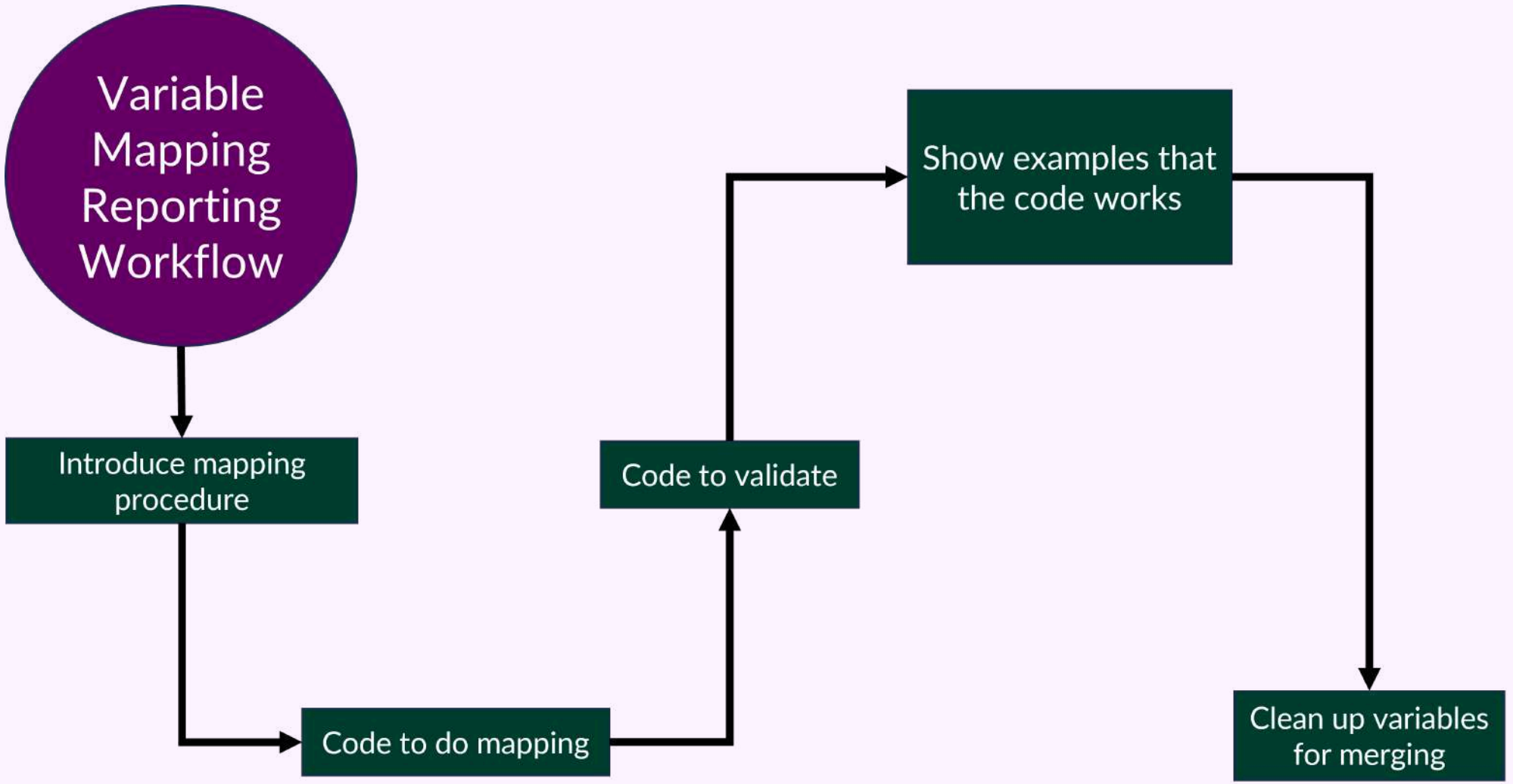
```r
cohort_data_excel <- readxl::read_excel(
  path = here::here("data-raw", "Cohort_Excel",
    "data_to_harmonise_age_issue.xlsx"),
  sheet = "Sheet1",
  col_types = c(
    "text", "numeric"
    )
  ) %T>%
  testthat::expect_no_condition()
```

```
Error: Expected `.` to run without any conditions.
ℹ Actually got a <simpleWarning> with text:
  Expecting numeric in B7 / R7C2: got 'missing'
```

No Issues

```r
cohort_data_excel_2 <- readxl::read_excel(
  path = here::here("data-raw", "Cohort_Excel",
    "data_to_harmonise_age_issue_fixed.xlsx"),
  sheet = "Sheet1",
  col_types = c("text", "numeric")
) %T>%
testthat::expect_no_condition() |>
# Check if Serial Number is unique
  pointblank::rows_distinct(
    columns = "Serial Number",
)
```

Variable Mapping



Variable Mapping

Let take this data set as an example.

```r
cohort_csv_data <- vroom::vroom(
  file = here::here("data-raw",
                    "Cohort_csv",
                    "data_to_harmonise.csv"),
  delim = ",",
  col_select = 1:8,
  show_col_types = FALSE,
  col_types = list(
    ID = vroom::col_character(),
    Age = vroom::col_integer(),
    Sex = vroom::col_character(),
    Height = vroom::col_double(),
    Weight = vroom::col_double(),
    `Smoke History` = vroom::col_character(),
    `Chest Pain Character` = vroom::col_character(),
    Dyspnea = vroom::col_character()
    )
  ) |>
  dplyr::rename(cohort_unique_id = "ID") |>
  # Remove rows when the ID value is NA
  dplyr::filter(!is.na(.data[["cohort_unique_id"]])) |>
```

cohort_unique_id	Age	Sex	Height	Weight	S H
		All			
B001	32	Female	170	63	n
B002	52	Female	167	71	cu sr
B003	80	Male	184	77	n
B004	70	Male	160	83	p
B005	70	Female	155	61	cu sr

1–5 of 20 rows Previous 1 of 4 Next

Download as CSV

Variable Mapping

Let the reader know how the collaborator's data `Smoke History` is going to be mapped.

Introduce mapping procedure

````
### Smoking History

`smoke_current` is grouped as follows:

```{r}
#| label: smoke current table
#| echo: false
#| message: false
#| warnings: false
#| results: asis

tabl <- "

+----------------------+----------------------+
| Smoke History        | smoke_current        |
+:=====================+:=====================+
| non-smoker           | 0                    |
+----------------------+----------------------+
| past smoker          | 0                    |
+----------------------+----------------------+
| current smoker       | 1                    |
+----------------------+----------------------+
| NA                   | -1                   |
+----------------------+----------------------+

"
cat(tabl)
```
````

2.3 Smoking History

`smoke_current` is grouped as follows:

Smoke History	smoke_current
non-smoker	0
past smoker	0
current smoker	1
NA	-1

`smoke_past` is grouped as follows:

Smoke History	smoke_past
non-smoker	0
past smoker	1
current smoker	0
NA	-1

We do a check to ensure that we can only have these scenarios

- `smoke_current` as 1 and `smoke_past` as 0 for current smokers
- `smoke_current` as 0 and `smoke_past` as 1 for past smokers
- `smoke_current` as 0 and `smoke_past` as 0 for non-smokers
- `smoke_current` as -1 and `smoke_past` as -1 for unknown

Variable Mapping

Code to do mapping

```
smoking_data <- cohort_csv_data |>
  dplyr::select(c("cohort_unique_id",
                  "Smoke History")) |>
  dplyr::mutate(
    smoke_current = dplyr::case_when(
      is.na(.data[["Smoke History"]]) ~ "-1",
      .data[["Smoke History"]] == "non-smoker" ~ "0",
      .data[["Smoke History"]] == "past smoker" ~ "0",
      .data[["Smoke History"]] == "current smoker" ~ "1",
      .default = NA_character_
    ),
    smoke_current = forcats::fct_relevel(
      .data[["smoke_current"]],
      c("0", "1")),
    smoke_past = dplyr::case_when(
      is.na(.data[["Smoke History"]]) ~ "-1",
      .data[["Smoke History"]] == "non-smoker" ~ "0",
      .data[["Smoke History"]] == "past smoker" ~ "1",
      .data[["Smoke History"]] == "current smoker" ~ "0",
      .default = NA_character_
    ),
```

2.3 Smoking History

`smoke_current` is grouped as follows:

Smoke History	smoke_current
non-smoker	0
past smoker	0
current smoker	1
NA	-1

`smoke_past` is grouped as follows:

Smoke History	smoke_past
non-smoker	0
past smoker	1
current smoker	0
NA	-1

Variable Mapping

Code to validate

```
smoking_data <- smoking_data |>
  pointblank::col_vals_in_set(
    columns = c("smoke_current", "smoke_past"),
    set = c("0", "1", "-1")
  ) |>
  pointblank::col_vals_expr(
    expr = pointblank::expr(
      (.data[["smoke_current"]] == "1" & .data[["smoke_past
      (.data[["smoke_current"]] == "-1" & .data[["smoke_pas
      (.data[["smoke_current"]] == "0" & .data[["smoke_past
    )
  )
```

We do a check to ensure that we can only have these scenarios

- `smoke_current` as 1 and `smoke_past` as 0 for current smokers
- `smoke_current` as 0 and `smoke_past` as 1 for past smokers
- `smoke_current` as 0 and `smoke_past` as 0 for non-smokers
- `smoke_current` as -1 and `smoke_past` as -1 for unknown

Reference: https://github.com/rstudio/pointblank/issues/578

Variable Mapping

Make use of Quarto's parameters, conditional content and `!expr` knitr engine syntax to choose what code/items to run/display on your html, pdf or word report.

Show examples that the code works

your_quarto_script.qmd

```yaml
---
params:
  show_table: TRUE
---
```

````
```{r}
#| label: output type
#| echo: false
#| warning: false
#| message: false

out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")
```
````

````
::: {.content-visible when-format="html"}

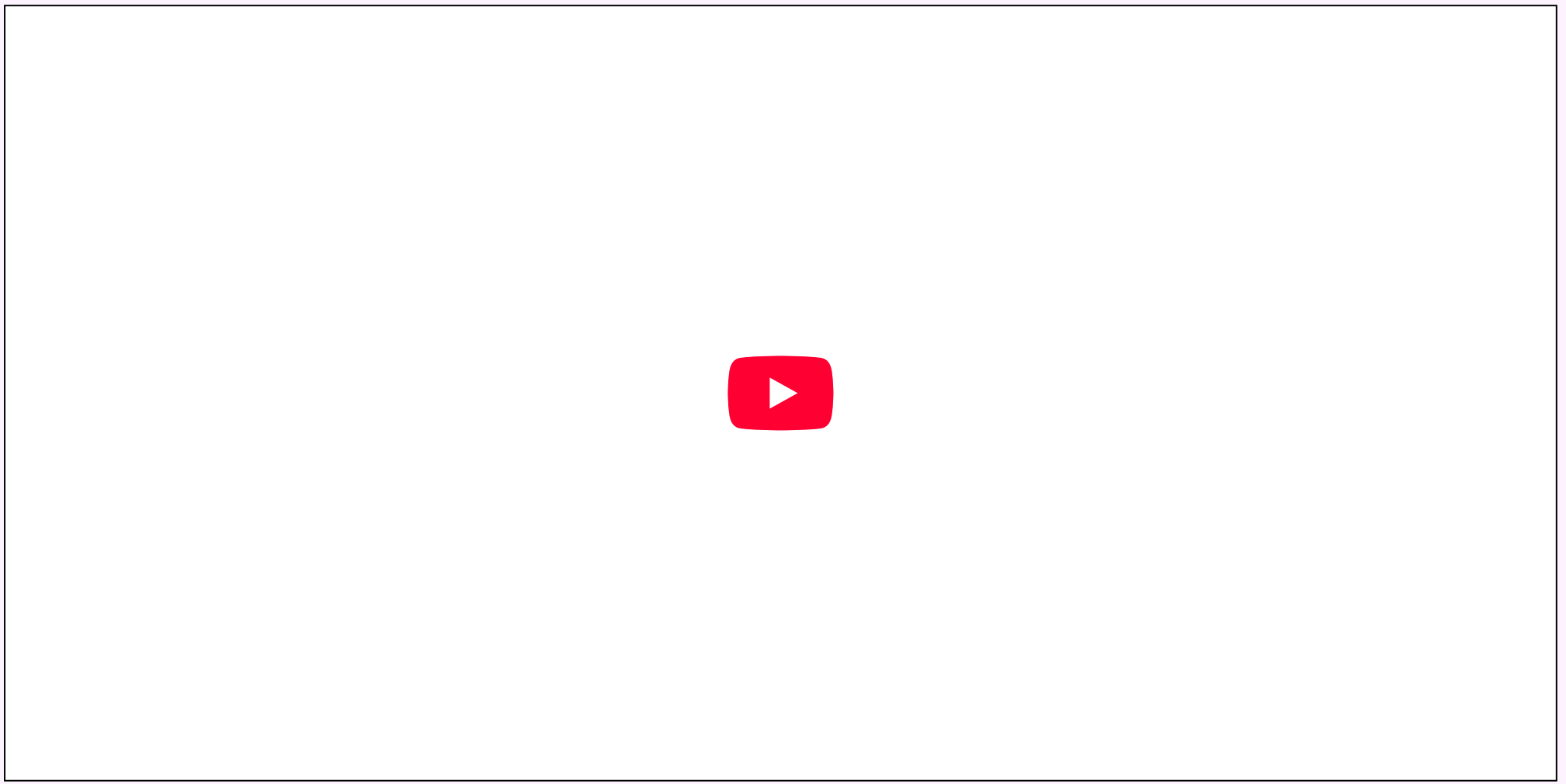
```{r}
#| label: smoking data html
#| eval: !expr out_type == "html"

if (params$show_table && knitr::is_html_output()) {
  smoking_data |>
    harmonisation::reactable_with_download_csv_button()
}
```

:::
````

Variable Mapping

Parameterized Quarto Reports Improve Understanding of Soil Health by Jadey Ryan.



Variable Mapping

Show examples that the code works

```
::: {.content-visible when-format="html"}

```{r}
#| label: smoking data html
#| eval: !expr out_type == "html"

if (params$show_table && knitr::is_html_output()) {
  smoking_data |>
    harmonisation::reactable_with_download_csv_button()
}

```

:::
```

Html Output

cohort_unique_id	Smoke History	smoke_current	smoke_past
	All	All	All
B001	non-smoker	0	0
B002	current smoker	1	0
B003	non-smoker	0	0
B004	past smoker	0	1
B005	current smoker	1	0

1–5 of 20 rows Previous 1 of 4 Next

Download as CSV

Variable Mapping

Show examples that the code works

````
::: {.content-visible unless-format="html"}

```{r}
#| label: smoking data not html
#| eval: !expr out_type != "html"

if (params$show_table) {
  smoking_data |>
    dplyr::distinct(.data[["Smoke History"]],
                    .keep_all = TRUE) |>
    knitr::kable()
}

```

:::
````

Pdf Output

```
if (params$show_table) {
  smoking_data |>
    dplyr::distinct(.data[["Smoke History"]],
                    .keep_all = TRUE) |>
    knitr::kable()
}
```

cohort_unique_id	Smoke History	smoke_current	smoke_past
B001	non-smoker	0	0
B002	current smoker	1	0
B004	past smoker	0	1
B017	NA	-1	-1

Variable Mapping

Clean up variables for merging

```
smoking_data <- smoking_data |>
  dplyr::select(-c("Smoke History"))
```

cohort_unique_id	smoke_current	smoke_past
	All	All
B001	0	0
B002	1	0
B003	0	0
B004	0	1
B005	1	0

1–5 of 20 rows Previous 1 of 4 Next

Download as CSV

Merging Harmonised Data

Suppose we have completed harmonising a batch of clinical data.

```
age_gender_data |>
  reactable_with_download_csv_button(
    defaultPageSize = 5,
    paginationType = "jump",
    style = list(fontSize = "1rem"),
  )
```

cohort_unique_id	age_years	sex
		All
B001	32	0
B002	52	0
B003	80	1
B004	70	1
B005	70	0

1–5 of 20 rows Previous 1 of 4 Next

Download as CSV

```
body_measurement_data |>
  reactable_with_download_csv_button(
    defaultPageSize = 5,
    paginationType = "jump",
    style = list(fontSize = "1rem"),
  )
```

cohort_unique_id	height_cm	weight_kg	bsa_m2	bmi
B001	170	63	1.72	21.8
B002	167	71	1.81	25.46
B003	184	77	1.98	22.74
B004	160	83	1.92	32.42
B005	155	61	1.62	25.39

1–5 of 20 rows Previous 1 of 4 Next

Download as CSV

How can we merge them without issues of missing rows or additional columns ?

Merging Harmonised Data

`unmatched = "error"` in `dplyr::inner_join` helps to avoid patients with no match.

```
join_specification <- dplyr::join_by("cohort_unique_id")

demo_behave_data <- cohort_csv_data |>
  dplyr::select(c("cohort_unique_id")) |>
  dplyr::inner_join(age_gender_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::inner_join(body_measurement_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::inner_join(smoking_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::relocate(c("bsa_m2", "bmi"),
                  .after = "sex")
```

```
three_penguins <- tibble::tribble(
  ~samp_id, ~species,    ~island,
  1,        "Adelie",    "Torgersen",
  2,        "Gentoo",    "Biscoe",
)

weight_extra <- tibble::tribble(
  ~samp_id,  ~body_mass_g,
  1,         3220,
  2,         4730,
  4,         4725
)

three_penguins |>
  dplyr::inner_join(
    y = weight_extra,
    by = dplyr::join_by("samp_id"),
    unmatched = "error"
  )
```

```
Error in `dplyr::inner_join()`:
! Each row of `y` must be matched by `x`.
i Row 3 of `y` was not matched.
```

Reference: https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/#improved-and-expanded-_join-functionality

Merging Harmonised Data

`unmatched = "error"` in `dplyr::inner_join` helps to avoid patients with no match.

```r
join_specification <- dplyr::join_by("cohort_unique_id")

demo_behave_data <- cohort_csv_data |>
  dplyr::select(c("cohort_unique_id")) |>
  dplyr::inner_join(age_gender_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::inner_join(body_measurement_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::inner_join(smoking_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::relocate(c("bsa_m2", "bmi"),
                  .after = "sex")
```

```r
three_penguins <- tibble::tribble(
  ~samp_id, ~species,    ~island,
  1,        "Adelie",    "Torgersen",
  2,        "Gentoo",    "Biscoe",
  3,        "Chinstrap", "Dream"
)

weight_extra <- tibble::tribble(
  ~samp_id,  ~body_mass_g,
  1,         3220,
  3,         4725
)

three_penguins |>
  dplyr::inner_join(
    y = weight_extra,
    by = dplyr::join_by("samp_id"),
    unmatched = "error"
  )
```

```
Error in `dplyr::inner_join()`:
! Each row of `x` must have a match in `y`.
i Row 2 of `x` does not have a match.
```

Reference: https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/#improved-and-expanded-_join-functionality

Merging Harmonised Data

`relationship = "one-to-one"` in `dplyr::inner_join` helps to avoid patients with multiple match.

```
join_specification <- dplyr::join_by("cohort_unique_id")

demo_behave_data <- cohort_csv_data |>
  dplyr::select(c("cohort_unique_id")) |>
  dplyr::inner_join(age_gender_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::inner_join(body_measurement_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::inner_join(smoking_data,
                    by = join_specification,
                    unmatched = "error",
                    relationship = "one-to-one") |>
  dplyr::relocate(c("bsa_m2", "bmi"),
                  .after = "sex")
```

```
three_penguins <- tibble::tribble(
  ~samp_id, ~species,    ~island,
  1,        "Adelie",    "Torgersen",
  2,        "Gentoo",    "Biscoe",
  3,        "Chinstrap", "Dream"
)

weight_extra <- tibble::tribble(
  ~samp_id,  ~body_mass_g,
  1,         3220,
  2,         4730,
  2,         4725,
  3,         4000
)

three_penguins |>
  dplyr::inner_join(
    y = weight_extra,
    by = dplyr::join_by("samp_id"),
    relationship = "one-to-one"
  )
```

```
Error in `dplyr::inner_join()`:
! Each row in `x` must match at most 1 row in `y`.
i Row 2 of `x` matches multiple rows in `y`.
```

Reference: https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/#improved-and-expanded-_join-functionality

Merging Harmonised Data

Use `pointblank::has_columns` to ensure we only have harmonised variables.

```
testthat::expect_false(
  pointblank::has_columns(
    demo_behave_data,
    columns = c(
      dplyr::ends_with(".x"),
      dplyr::ends_with(".y")
    )
  )
)

testthat::expect_equal(
  ncol(demo_behave_data), 9
)

testthat::expect_true(
  pointblank::has_columns(
    demo_behave_data,
    columns = c(
      "age_years", "sex",
      "height_cm", "weight_kg", "bsa_m2", "bmi",
      "smoke_current", "smoke_past"
```

```
three_penguins <- tibble::tribble(
  ~samp_id, ~species,    ~island,
  1,        "Adelie",    "Torgersen",
  2,        "Gentoo",    "Biscoe",
  3,        "Chinstrap", "Dream"
)

weight_extra <- tibble::tribble(
  ~samp_id,  ~island,
  1,         "Torgersen",
  2,         "Biscoe",
  3,         "Dream"
)

three_penguins <- three_penguins |>
  dplyr::inner_join(
    y = weight_extra,
    by = dplyr::join_by("samp_id"),
    unmatched = "error",
    relationship = "one-to-one"
 )
```

```
[1] TRUE
```

```
colnames(three_penguins)
```

```
[1] "samp_id"  "species"  "island.x" "island.y"
```

Comparing Datasets

Use daff to compare different version of harmonised datasets.

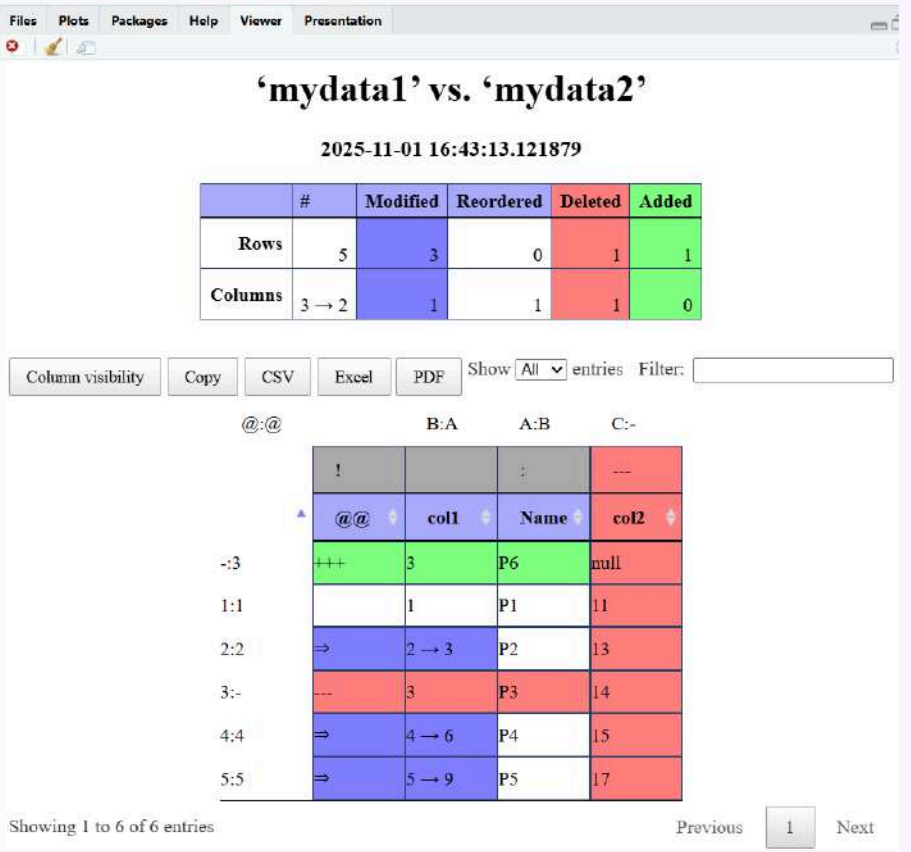
```r
data1 <- data.frame(
  Name=c("P1","P2","P3","P4","P5"),
  col1=c(1,2,3,4,5),
  col2=c(11,13,14,15,17)
)

data2 <- data.frame(
  col1=c(1,3,3,6,9),
  Name=c("P1","P2","P6","P4","P5")
)

compare_results <- daff::diff_data(data1, data2)
compare_results
```

```
Daff Comparison: 'data1' vs. 'data2'
        B:A  A:B  C:-
    !        :    ---
    @@  col1 Name col2
1:1     1    P1   11
2:2 ->  2->3 P2   13
-:3 +++ 3    P6   <NA>
3:- --- 3    P3   14
4:4 ->  4->6 P4   15
5:5 ->  5->9 P5   17
```

```r
daff::render_diff(compare_results)
```



Comparing Datasets

Use `summary()` to return a summary list.

```r
data1 <- data.frame(
  Name=c("P1","P2","P3","P4","P5"),
  col1=c(1,2,3,4,5),
  col2=c(11,13,14,15,17)
)

data2 <- data.frame(
  col1=c(1,3,3,6,9),
  Name=c("P1","P2","P6","P4","P5")
)

compare_different_summary <- daff::diff_data(
  data1, data2) |>
  summary()

compare_different_summary
```

```
Data diff: 'data1' vs. 'data2'
        #       Modified Reordered Deleted Added
Rows    5       3        0         1       1
Columns 3 --> 2 1        1         1       0
```

```r
data1 <- data.frame(
  Name=c("P1","P2","P3","P4","P5"),
  col1=c(1,2,3,4,5),
  col2=c(11,13,14,15,17)
)

data2 <- data.frame(
  Name=c("P1","P2","P3","P4","P5"),
  col1=c(1,2,3,4,5),
  col2=c(11,13,14,15,17)
)

compare_same_summary <- daff::diff_data(data1, data2) |>
  summary()

compare_same_summary
```

```
Data diff: 'data1' vs. 'data2'
        # Modified Reordered Deleted Added
Rows    5        0         0       0     0
Columns 3        0         0       0     0
```

Comparing Datasets

Use the summary list and `pointblank::expect_col_vals_in_set` to do the validation automatically.

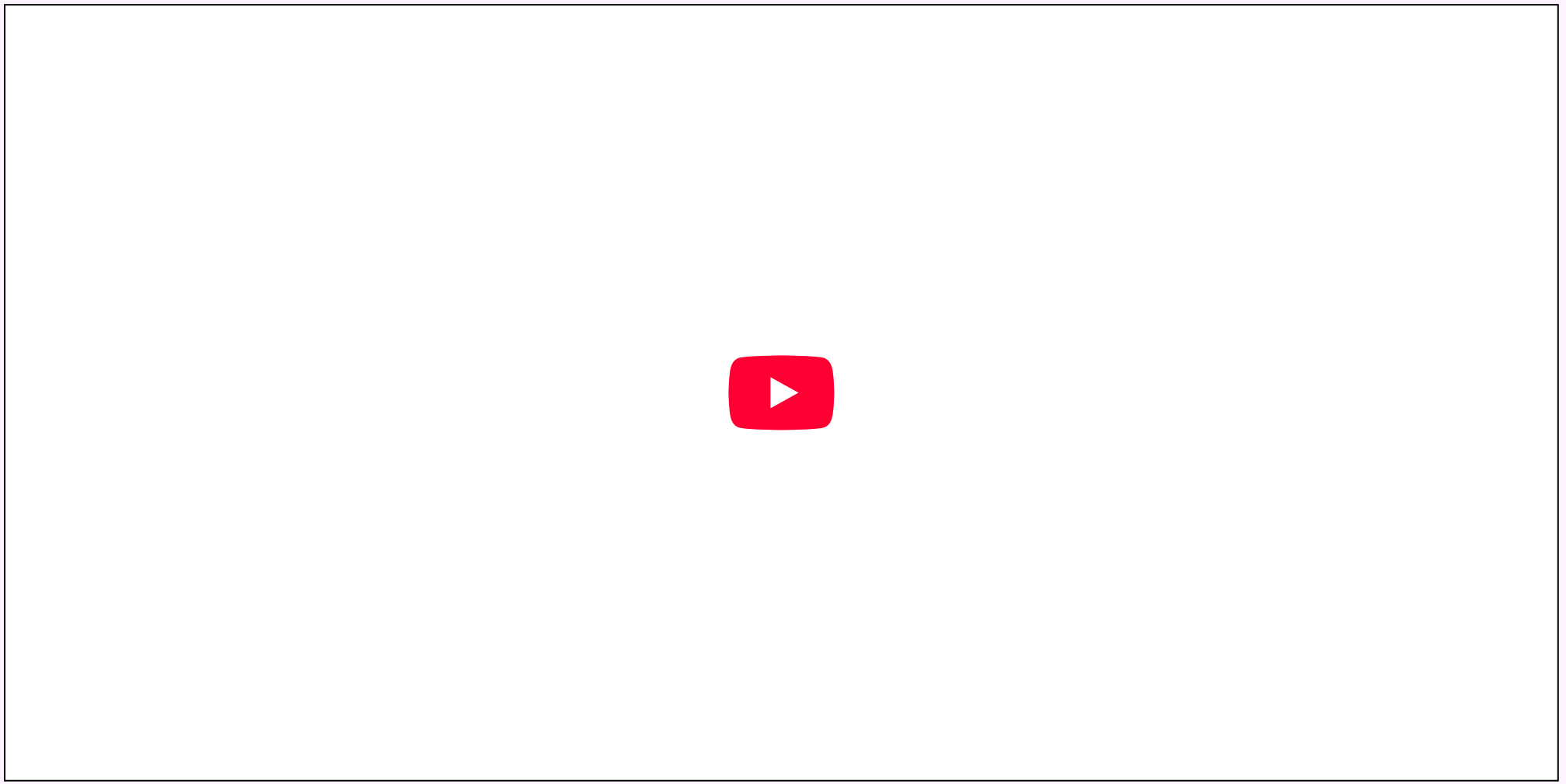
```
tibble::tibble(
  row_deletes = compare_different_summary$row_deletes,
  row_inserts = compare_different_summary$row_inserts,
  row_updates = compare_different_summary$row_updates,
  row_reorders = compare_different_summary$row_reorders,
  col_deletes = compare_different_summary$col_deletes,
  col_inserts = compare_different_summary$col_inserts,
  col_updates = compare_different_summary$col_updates,
  col_reorders = compare_different_summary$col_reorders,
) |>
  pointblank::expect_col_vals_in_set(
    columns = c(
      "row_deletes", "row_inserts",
      "row_updates", "row_reorders",
      "col_deletes", "col_inserts",
      "col_updates", "col_reorders"
    ),
    set = c(0)
  )
```

```
Error: Exceedance of failed test units where values in
`row_deletes` should have been in the set of `0`.
The `expect_col_vals_in_set()` validation failed beyond the
absolute threshold level (1).
* failure level (1) >= failure threshold (1)
```

```
tibble::tibble(
  row_deletes = compare_same_summary$row_deletes,
  row_inserts = compare_same_summary$row_inserts,
  row_updates = compare_same_summary$row_updates,
  row_reorders = compare_same_summary$row_reorders,
  col_deletes = compare_same_summary$col_deletes,
  col_inserts = compare_same_summary$col_inserts,
  col_updates = compare_same_summary$col_updates,
  col_reorders = compare_same_summary$col_reorders,
) |>
  pointblank::expect_col_vals_in_set(
    columns = c(
      "row_deletes", "row_inserts",
      "row_updates", "row_reorders",
      "col_deletes", "col_inserts",
      "col_updates", "col_reorders"
    ),
    set = c(0)
  )
```

Comparing Datasets

Find the difference between two datasets in R



Technical Report Challenge

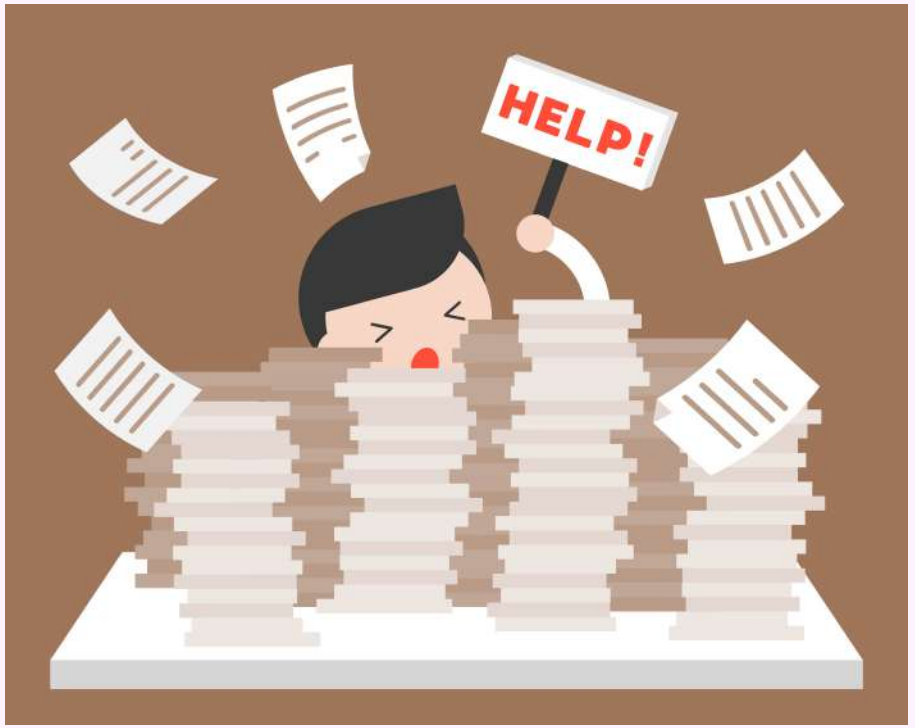
One variable mapping report takes at least one page.

On average, a clinical trial will have a few hundred variables.

- One hundred columns for clinical and demographics.
- Two hundred columns for medication.

Harmonisation report can have at least a few hundreds pages for each cohort.

There is a need to automate the creation of these reports.



Businessman in pile of documents asking for help by Amonrat Rungreangfangsai

Harmonisation Template for Cohort B

AUTHOR
My Name

PUBLISHED
March 10, 2025

Preface

Here is the documentation of the data harmonisation step generated using Quarto. To learn more about Quarto books visit https://quarto.org/docs/books.

File Structure

Here is the file structure of the project used to generate the document.

```
harmonisation/                          # Root of the project template.
|
├── quarto/ (not in repository)         # Folder to keep intermediate files/folders
```

Quarto Level 3

To make a Quarto book or website, we need a `_quarto.yml` and `index.qmd` file

_quarto.yml

```yaml
project:
  type: book
  output-dir: reports/Cohort_B

book:
  downloads: [pdf, docx]
  title: "Harmonisation Template for Cohort B"
  author: "My Name"
  navbar:
    search: true
  sidebar:
    collapse-level: 1

  chapters:
    - index.qmd
    - part: Cohort B Cleaning
      chapters:
        - codes/Cohort_B/00_R_Package_And_Environment.qmd
        - codes/Cohort_B/01_Read_Cohort_B_Data.qmd
        - codes/Cohort_B/02_Extract_Demographic.qmd
        - codes/Cohort_B/03_Export_To_Excel.qmd
```

index.qmd

```yaml
---
date: "2025-03-10"
format:
  html:
    code-fold: true
    freeze: false
params:
  show_table: TRUE
---
```

```{r}
#| label: output type
#| echo: false
#| warning: false
#| message: false

out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")
```

```
# Preface {.unnumbered .unlisted}

Here is the documentation of the data harmonisation step generated using
[Quarto](https://quarto.org/). To learn more about Quarto books visit
<https://quarto.org/docs/books>.
```

Quarto Level 3

`_quarto.yml` is a configuration file to tell Quarto to create a book.

_quarto.yml

```yaml
---
project:
  type: book
  output-dir: reports/Cohort_B

book:
  downloads: [pdf, docx]
  title: "Harmonisation Template for Cohort B"
  author: "My Name"
  navbar:
    search: true
  sidebar:
    collapse-level: 1

  chapters:
    - index.qmd
    - part: Cohort B Cleaning
      chapters:
        - codes/Cohort_B/00_R_Package_And_Environment.qmd
        - codes/Cohort_B/01_Read_Cohort_B_Data.qmd
        - codes/Cohort_B/02_Extract_Demographic.qmd
```

Quarto Level 3

`index.qmd` file gives the preface (homepage) content of the Quarto book (website). It is compulsory file needed for the rendering to work.

index.qmd

```
---
date: "2025-03-10"
format:
  html:
    code-fold: true
    freeze: false
params:
  show_table: TRUE
---

```{r}
#| label: output type
#| echo: false
#| warning: false
#| message: false

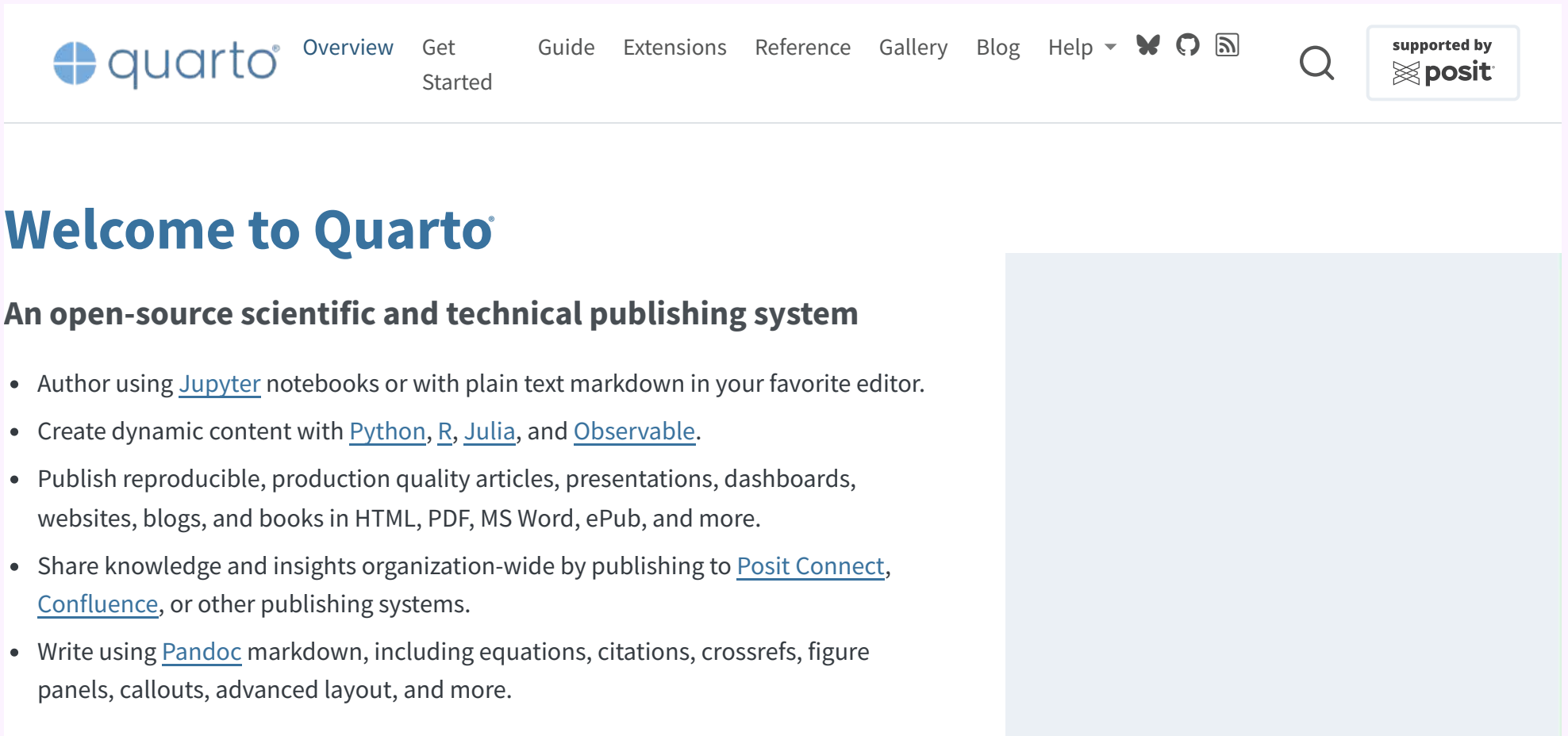
out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")
```

Preface {.unnumbered .unlisted}

```

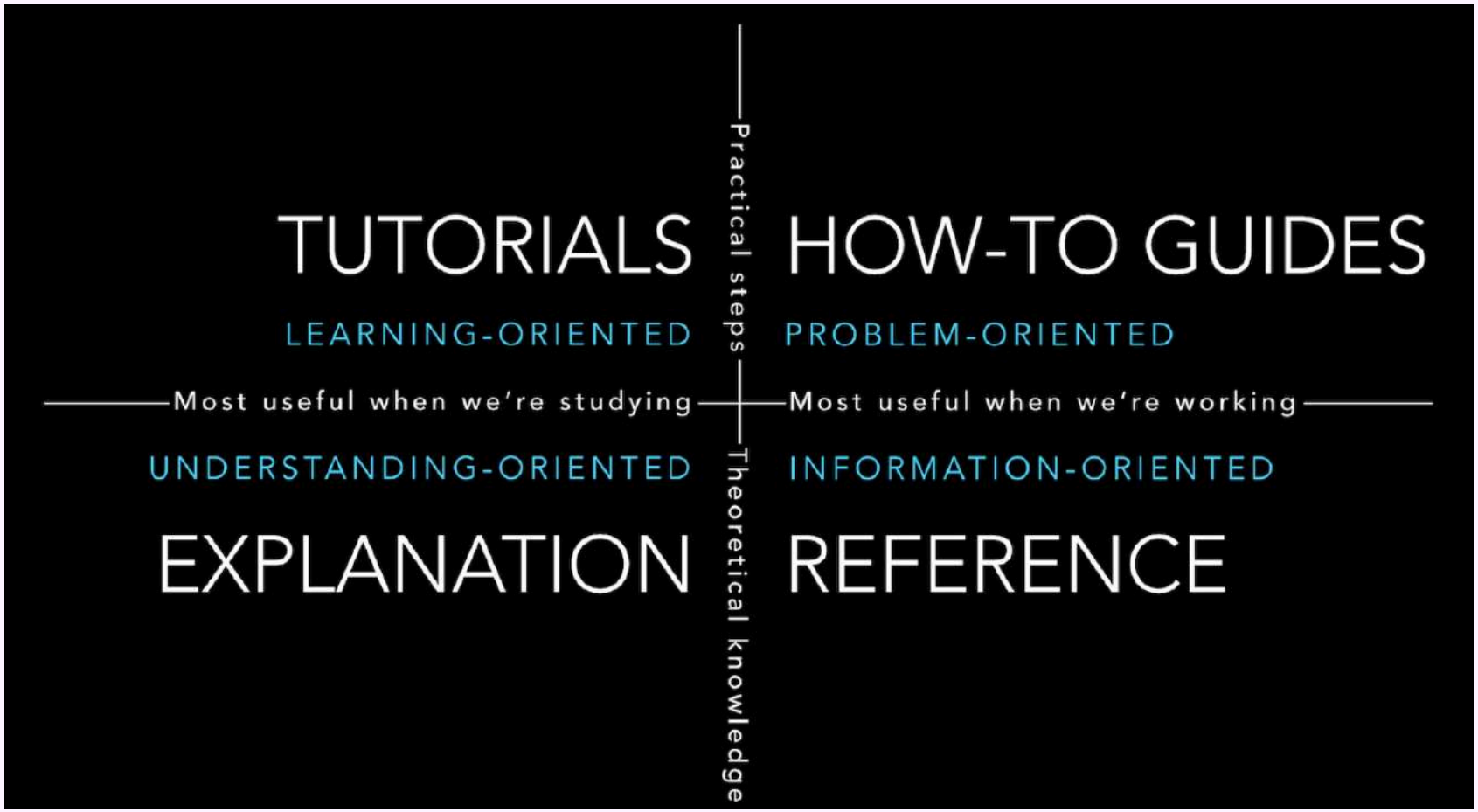
# Quarto (Reference)

https://quarto.org/



# Harmonisation Report Types

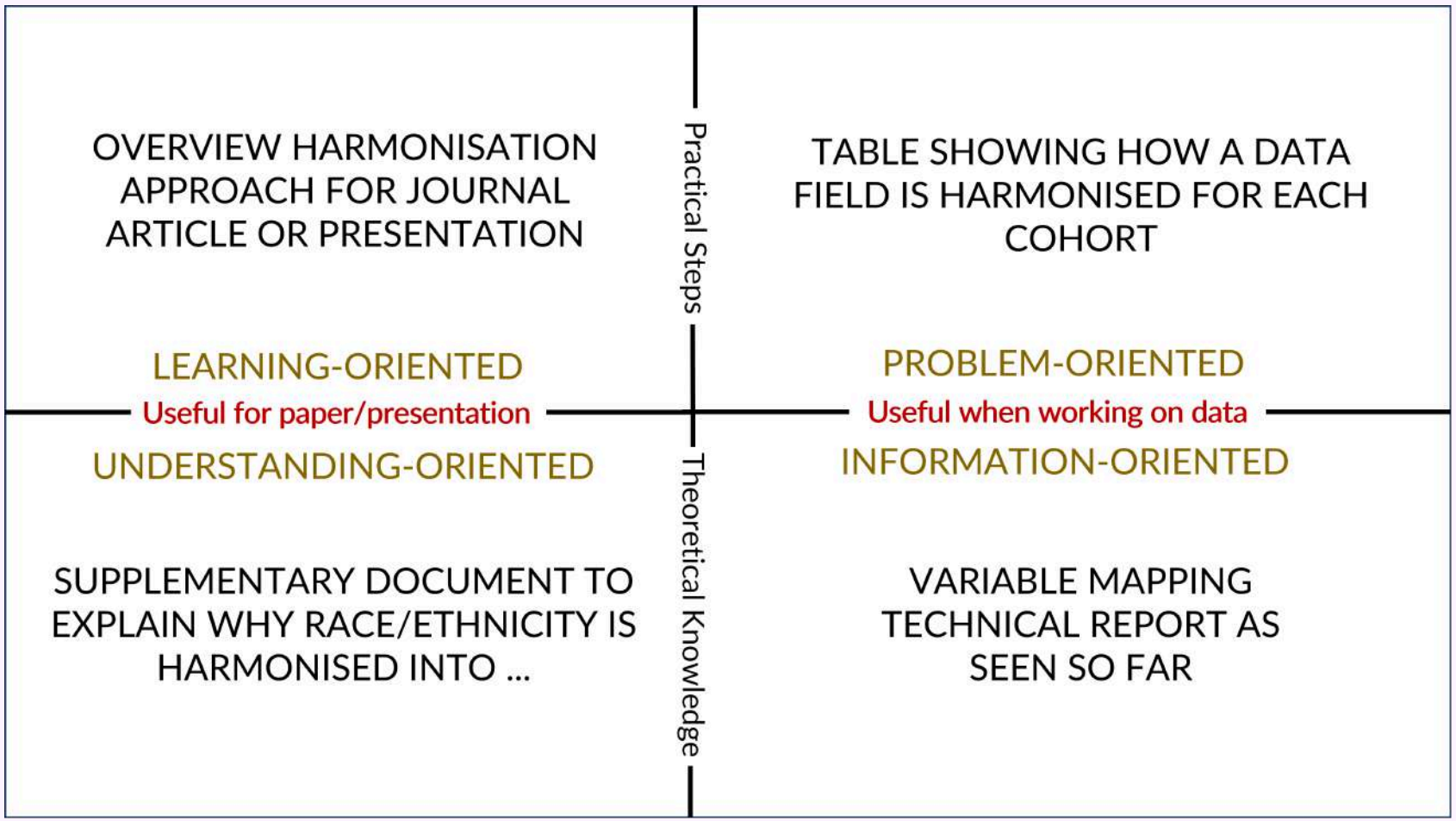
Collaborator wants different ways to report how data harmonisation is done.




The documentation system by Divio

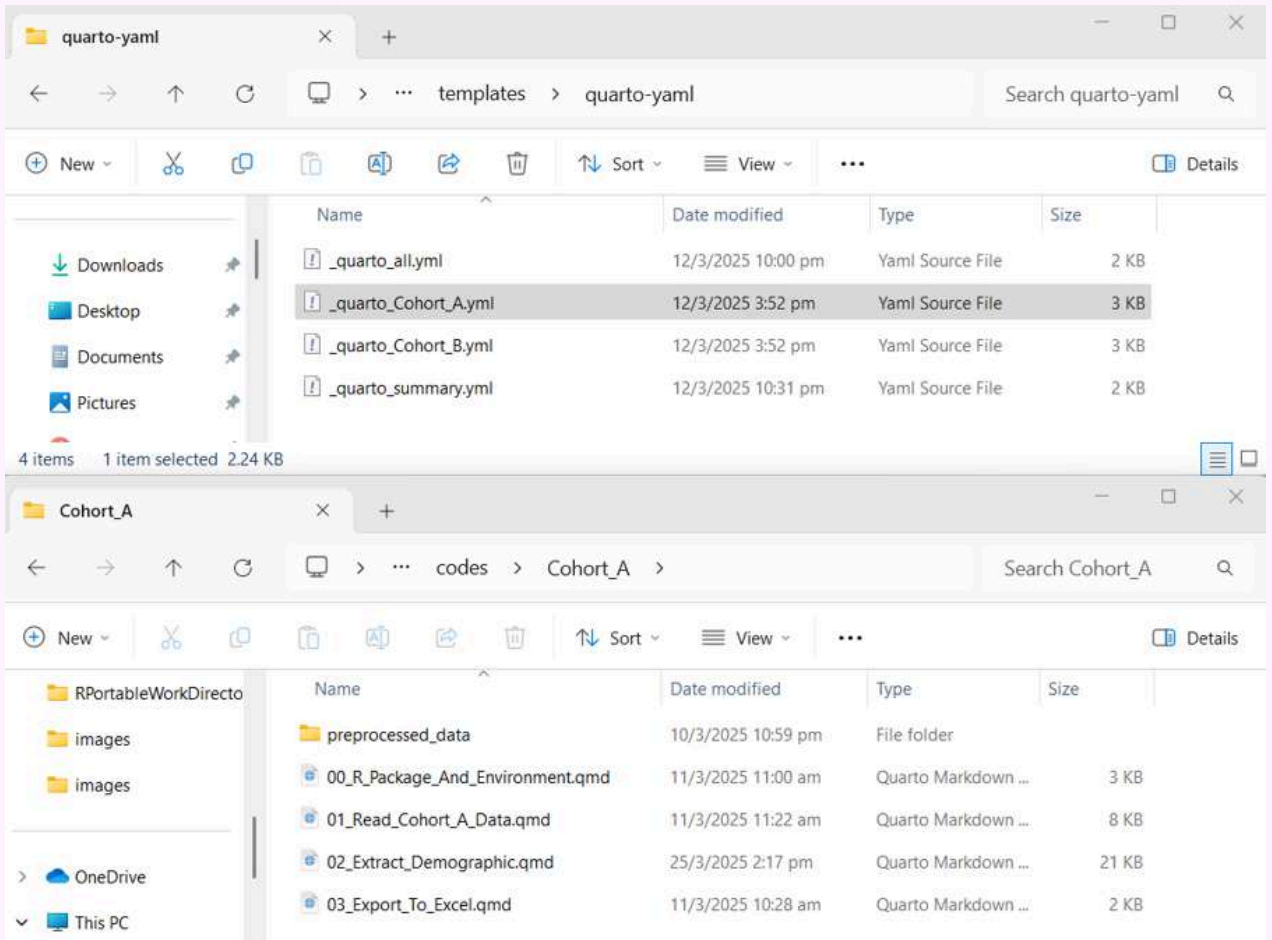
# Harmonisation Report Types

Collaborator wants different ways to report how data harmonisation is done.



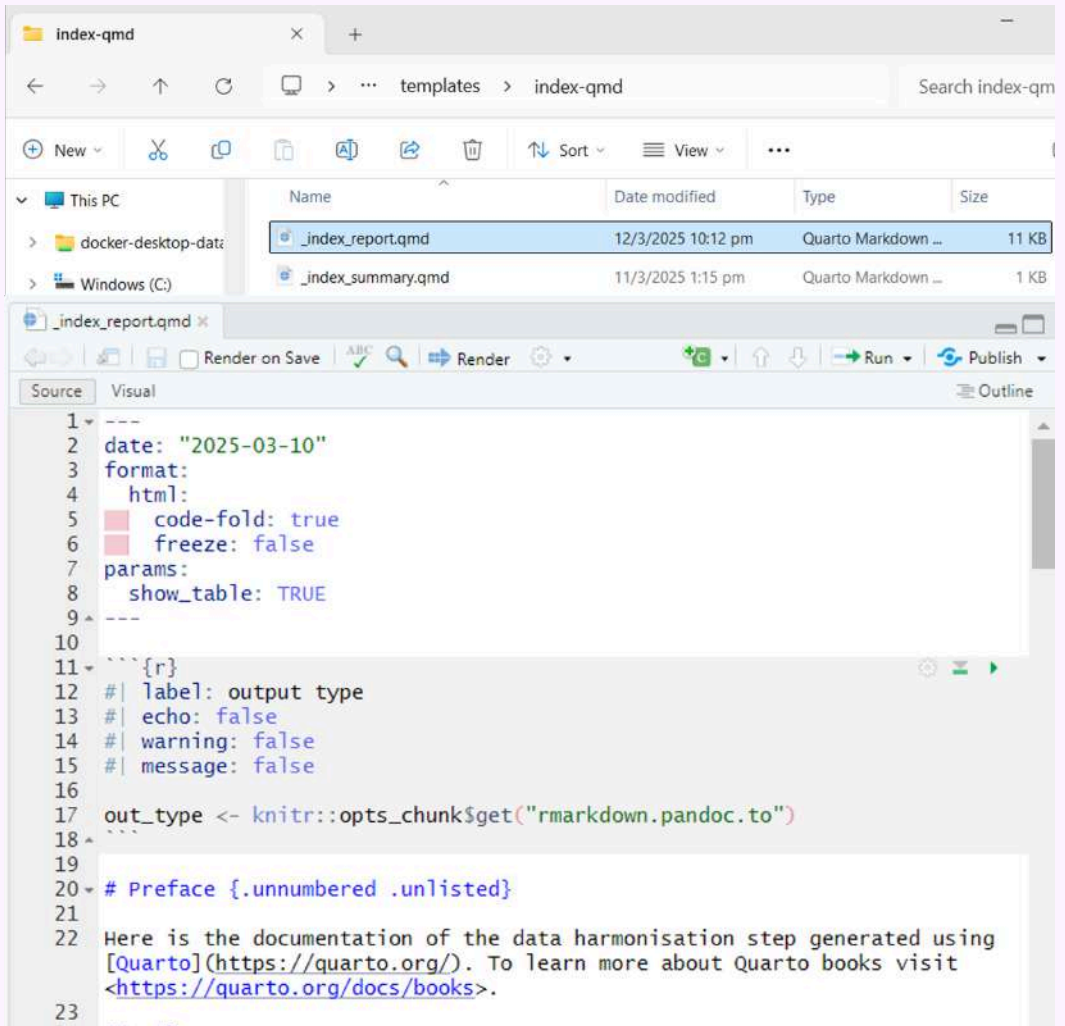
# Automated Technical Report (Reference)

We create a `_quarto.yml` file and relevant Quarto files for each cohort.



# Automated Technical Report (Reference)

We create an `index.qmd` file for each kind of report.



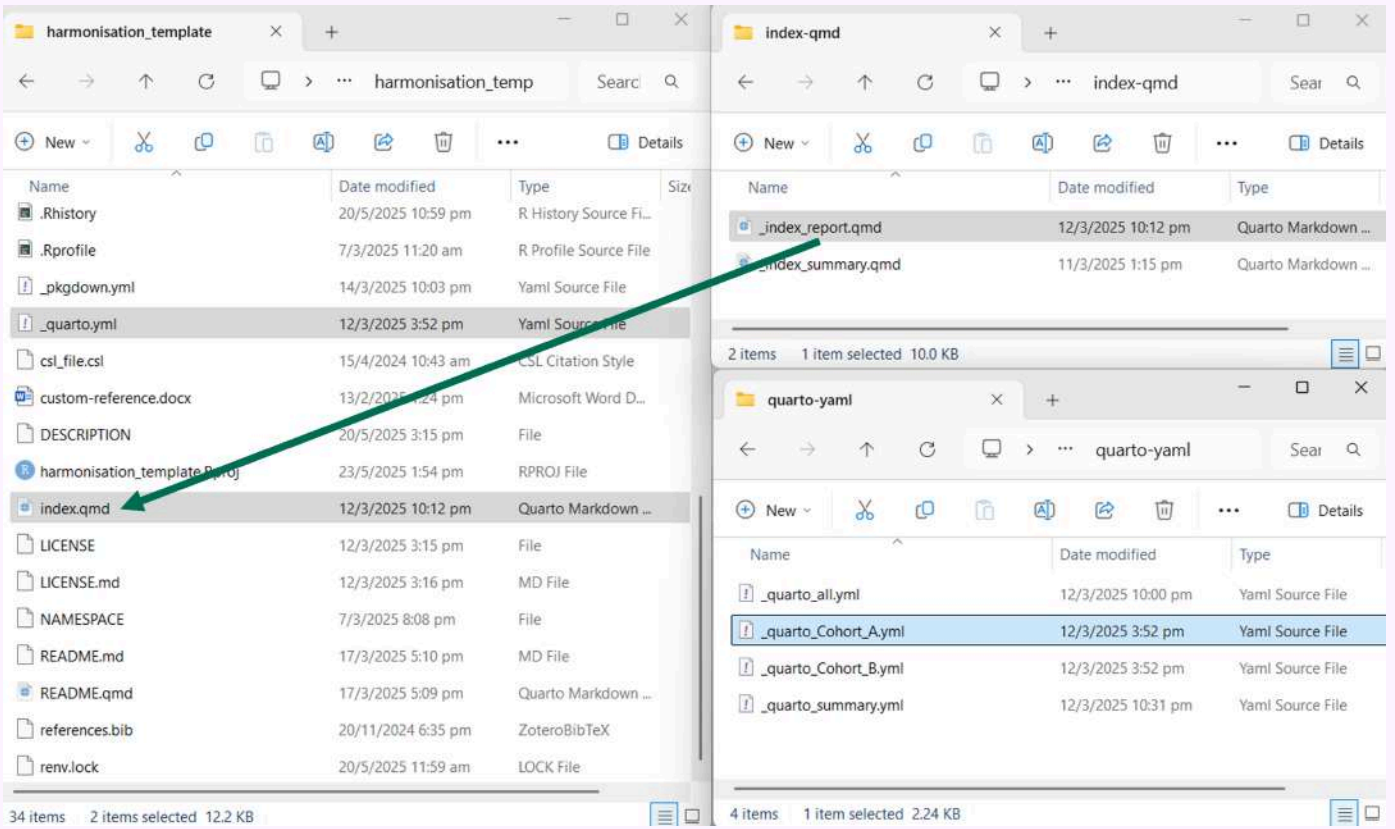
# Automated Technical Report (Reference)

Create a script to generate technical reports in pdf, word and html for each cohort.

```r
# Copy the right index.qmd
# file

index_qmd_file <- paste0(
  "_index_",
  "report",
  ".qmd"
)

fs::file_copy(
  path = here::here(
    "templates",
    "index-qmd",
    index_qmd_file),
  new_path = here::here(
    "index.qmd"
  ),
  overwrite = TRUE
)
```



Automated Technical Report (Reference)

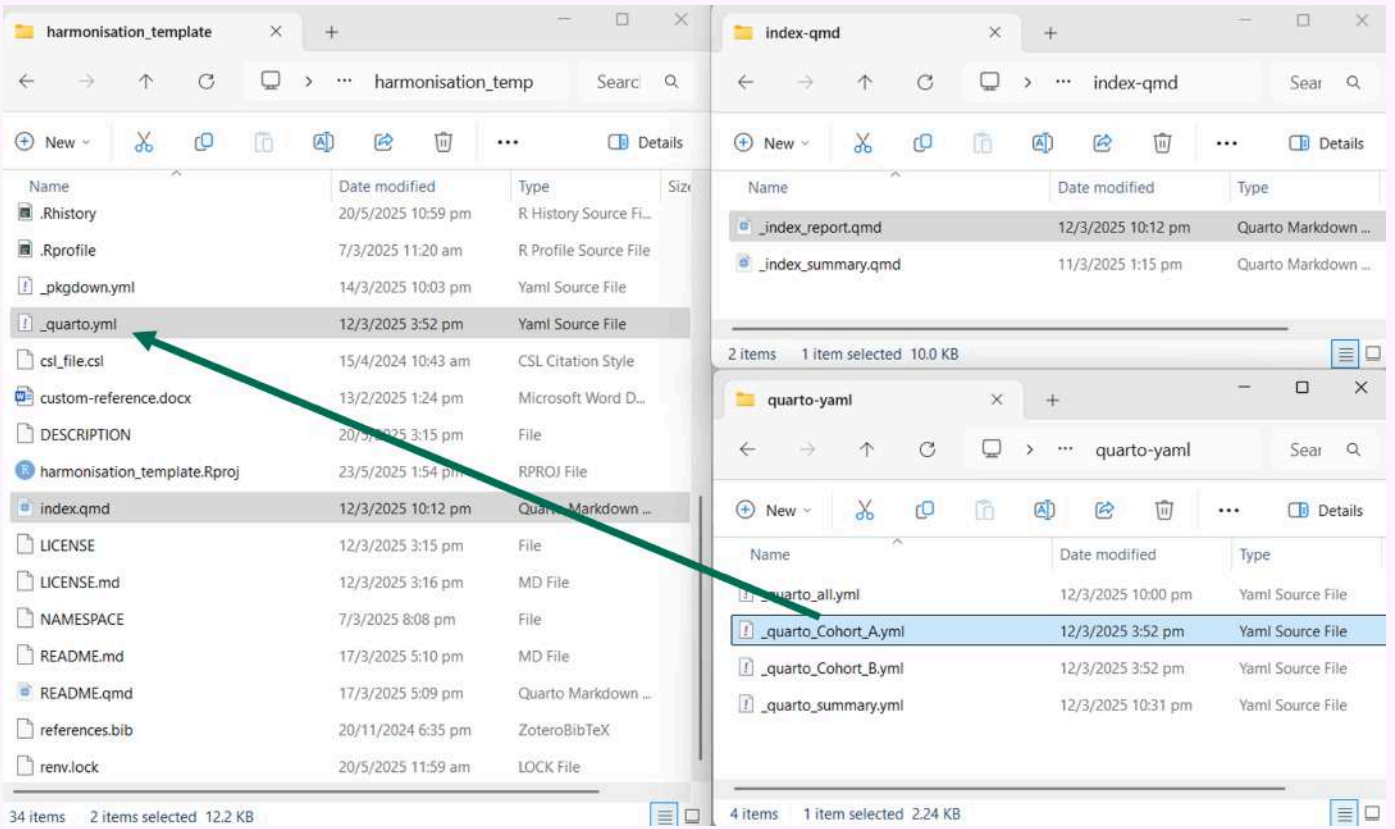
Create a script to generate technical reports in pdf, word and html for each cohort.

```r
copy_and_render <- function(
    cohort
) {

  # Copy quarto.yml file
  # for each cohort

  quarto_yml_file <- paste0(
    "_quarto_",
    cohort,
    ".yml"
  )

  fs::file_copy(
    path = here::here(
      "templates",
      "quarto-yaml",
      quarto_yml_file),
    new_path = here::here("_quar
    overwrite = TRUE
  )
```

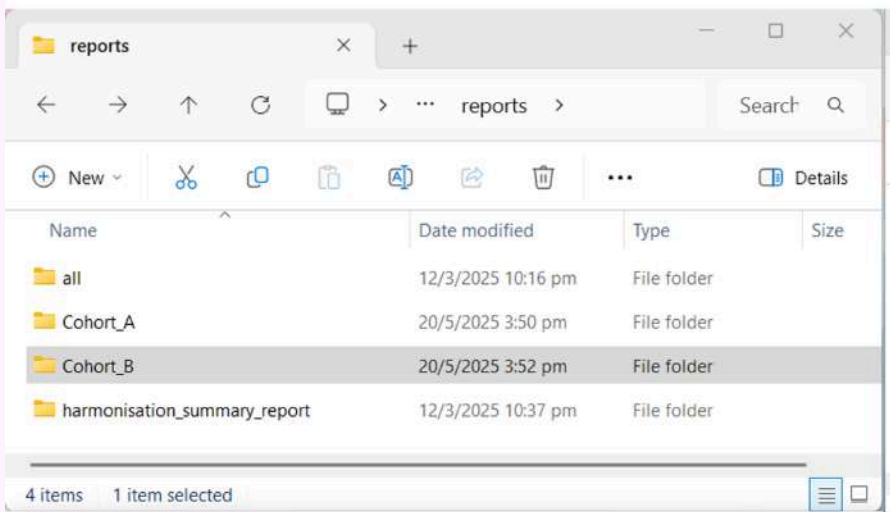


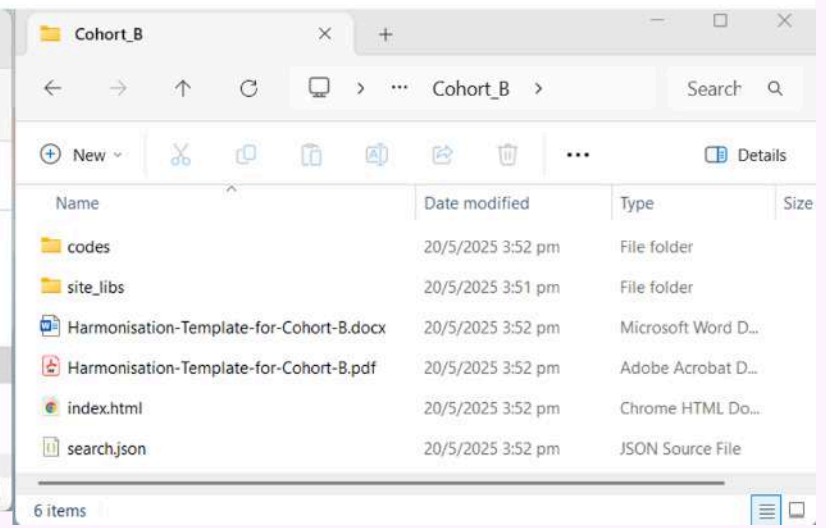
Automated Technical Report (Reference)

Output of these reports are as follows:

Run the R script `cohort_harmonisation_script.R` in `codes` folder to generate:

- Cohort_A Harmonisation Report:
  - 🌐 HTML: https://jauntyjjs-harmonisation-cohort-a.netlify.app
  - 📕 PDF : https://jauntyjjs-harmonisation-cohort-a.netlify.app/Harmonisation-Template-for-Cohort-A.pdf
  - 📘 Word: https://jauntyjjs-harmonisation-cohort-a.netlify.app/Harmonisation-Template-for-Cohort-A.docx
- Cohort_B Harmonisation Report:
  - 🌐 HTML: https://jauntyjjs-harmonisation-cohort-b.netlify.app
  - 📕 PDF : https://jauntyjjs-harmonisation-cohort-b.netlify.app/Harmonisation-Template-for-Cohort-B.pdf
  - 📘 Word: https://jauntyjjs-harmonisation-cohort-b.netlify.app/Harmonisation-Template-for-Cohort-B.docx





Automated Summary Report (How-to-Guide)

A similar method is done to create a summary report in word using R 📦 `flextable`.

2.4 Smoking History

smoke_current is the harmonised data field to denote if the patient is a current smoker during the time of the CT scan. *smoke_past* is the harmonised data field to denote if the patient is a past smoker during the time of the CT scan.

They hold the following values:

Table S6: Harmonised values of smoke_current and smoke_past.

Value	Description
0	no
1	yes
-1	unknown

They are harmonised as follows:

Table S7: Harmonised process of smoke_current and smoke_past.

Cohort ID	Original Response	Harmonisation Response
Cohort A	Column *smoke_current_good* with *0* as no. *1* as yes. *-1* as unknown. Column *smoke_past_good* with *0* as no. *1* as yes. *-1* as unknown.	*smoke_current* will take the values of *smoke_current_good*. *smoke_past* will take the values of *smoke_past_good*.
Cohort B	Column *Smoke History* with *non-smoker* as non-smoker. *past smoker* as a past smoker. *current smoker* as a current smoker. *NA* as unknown.	Map the values of *Smoke History* to *smoke_current* as follows: *non-smoker* and *past smoker* as 0. *current smoker* as 1. *NA* as -1.
		Map the values of *Smoke History* to *smoke_past* as follows: *non-smoker* and *current smoker* as 0. *past smoker* as 1. *NA* as -1.

After harmonisation, we validate the values of *smoke_current* and *smoke_past* to ensure that there can only be the following cases:

Table S8: Valid values of smoke_current and smoke_past.

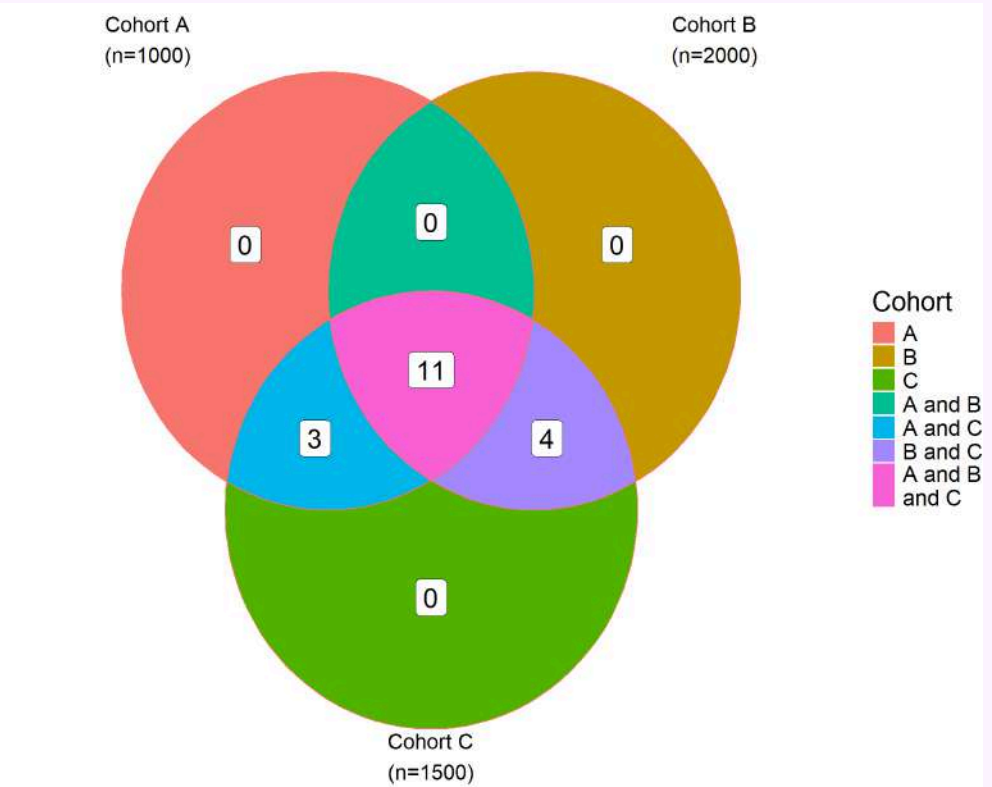
Description	*smoke_current*	*smoke_past*
Non-smoker	0	0
Past smoker	0	1
Current smoker	1	0
Unknown	-1	-1

Overview Diagrams

How many variables can each cohort provide ?

How many variables can be harmonised ?

```r
demographic_list <- list(
  A = c("Age", "Sex",
        "Hypertension", "Dyslipidemia", "Family Hx CAD", "D
        "Smoke Current", "Smoke Past",
        "Have Chest Pain", "Chest Pain Character",
        "Dyspnea",
        "BMI", "Height", "Weight"),
  B = c("Age", "Sex",
        "Hypertension", "Dyslipidemia", "Family Hx CAD", "D
        "Smoke Current", "Smoke Past",
        "Have Chest Pain", "Chest Pain Character",
        "Dyspnea",
        "HDL", "Total Cholesterol",
        "Triglyceride", "LDL"),
  C = c("Age", "Sex",
        "Hypertension", "Dyslipidemia", "Family Hx CAD", "D
        "Smoke Current", "Smoke Past",
        "Have Chest Pain", "Chest Pain Character",
        "Dyspnea",
        "BMI", "Height", "Weight",
        "HDL", "Total Cholesterol",
```

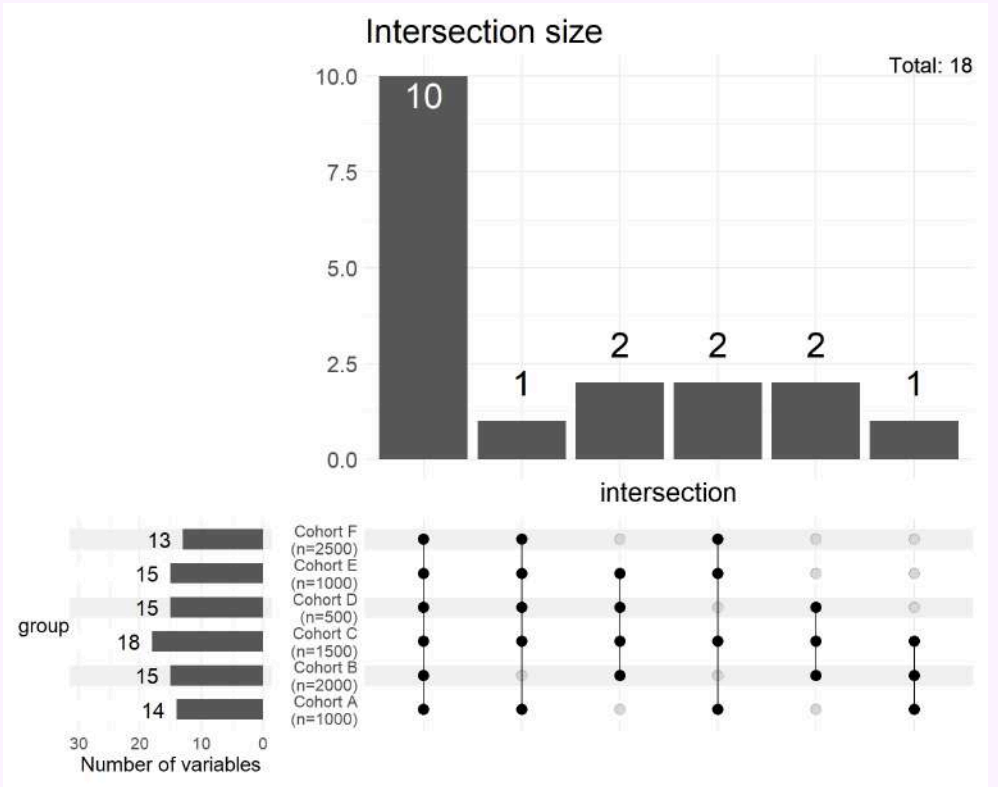


Venn diagram does not work for many (> 10) cohorts.

Overview Diagrams

Upset plots are too complicated for clinicians.

```r
demographic_venn <- tibble::tibble(
  column_name = c("Age", "Sex",
                  "Hypertension", "Dyslipidemia", "Family H
                  "Smoke Current", "Smoke Past",
                  "Have Chest Pain", "Chest Pain Character"
                  "Dyspnea",
                  "BMI", "Height", "Weight",
                  "HDL", "Total Cholesterol",
                  "Triglyceride", "LDL"),
  `Cohort A` = c(1, 1,
                 1, 1, 1, 1,
                 1, 1,
                 1, 1,
                 1,
                 1, 1, 1,
                 0, 0,
                 0, 0),
  `Cohort B` = c(1, 1,
                 1, 1, 1, 1,
                 1, 1,
                 1, 1,
```



Cannot answer follow-up questions:

How many cohorts provide patient's blood lipid information and how many patients have them ?

Overview Diagrams

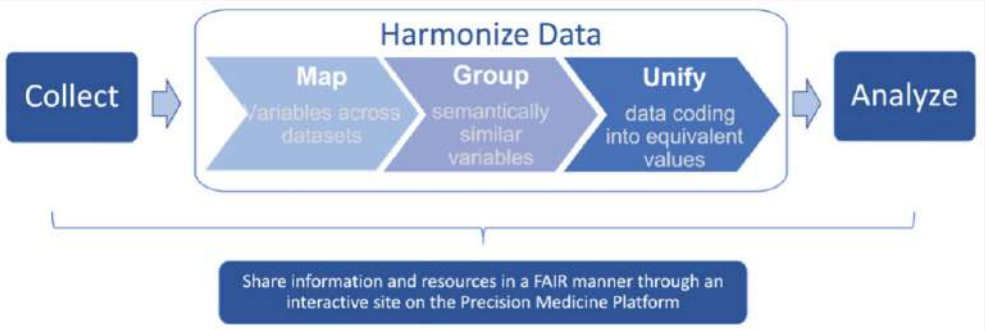
Create a “heatmap”using Microsoft PowerPoint.

				10			2	2	2
				Age					
				Sex					
				Hypertension					
				Dyslipidemia					
				Family Hx CAD					
				Diabetes					
				Smoke Current					
				Smoke Past					
				Have Chest Pain	1	1	Height	HDL	Triglyceride
Country	**Cohort**	**N**	**Variables**	Chest Pain Character	Dyspnea	BMI	Weight	Total Cholesterol	LDL
Country A	Cohort A	1000	15	✓	✓	✓	✓	×	×
	Cohort B	2000	16	✓	✓	×	×	✓	✓
Country B	Cohort C	1500	18	✓	✓	✓	✓	✓	✓
Country C	Cohort D	500	16	✓	×	✓	×	✓	✓
Country D	Cohort E	1000	16	✓	×	✓	✓	✓	×
	Cohort F	2500	14	✓	×	✓	✓		
Total				**8500**	**4500**	**6500**	**6000**	**5000**	**4000**

Variable Colour Legend	
	Age
	Sex
	Comorbidity
	Smoking history
	Symptoms
	Obesity
	Blood lipid

Table Legend	
✓	Available
×	Not available
	Pending arrival

Summary



Project Organisation

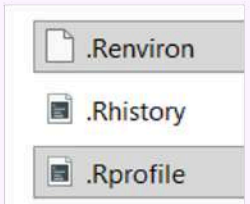
Welcome to **Posit Public Package Manager**

The best way to discover and install R and Python packages

R Administration

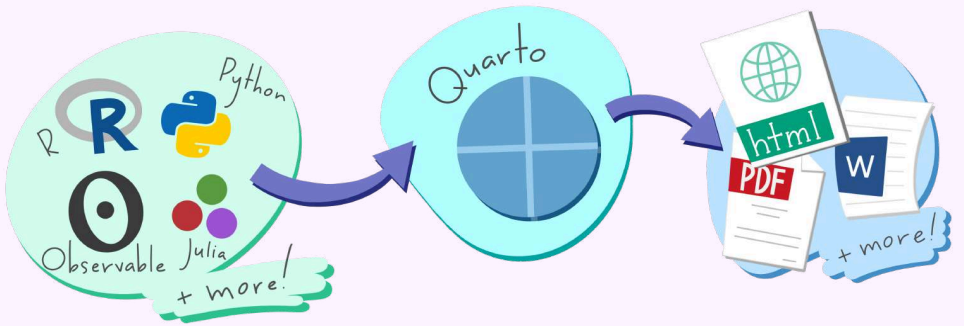
Namespacing

`dplyr::select()`

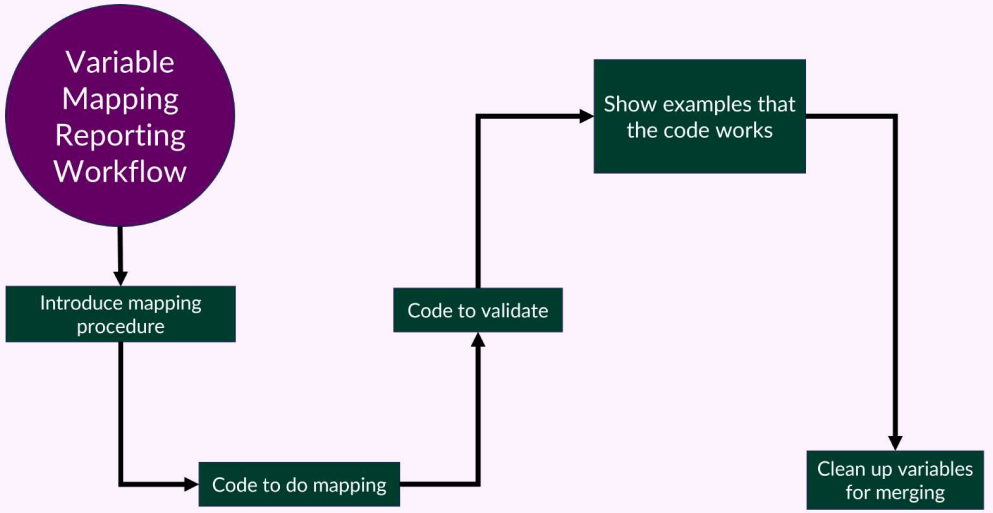


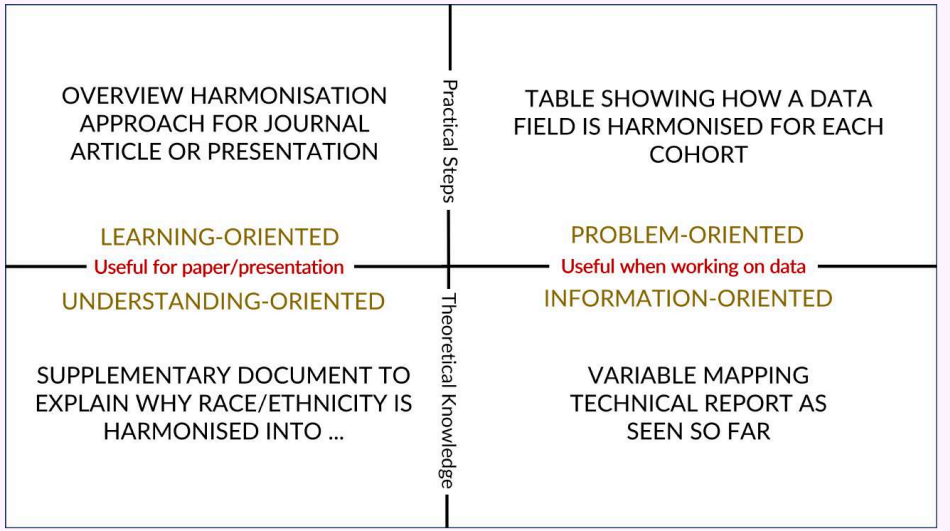
pak

A Fresh Approach to R Package Installation



Summary





‘data1’ vs. ‘data2’

2025-11-01 22:45:04.295812

	#	Modified	Reordered	Deleted	Added
Rows	5	3	0	1	1
Columns	3 → 2	1	1	1	0

				10			2	2	2
				Age					
				Sex					
				Hypertension					
				Dyslipidemia					
				Family Hx CAD					
				Diabetes					
				Smoke Current					
				Smoke Past					
				Have Chest Pain	1	1	Height	HDL	Triglyceride
Country	**Cohort**	**N**	**Variables**	Chest Pain Character	Dyspnea	BMI	Weight	Total Cholesterol	LDL
Country A	Cohort A	1000	15	✓	✓	✓	✓	×	×
	Cohort B	2000	16	✓	✓	×	×	✓	✓
Country B	Cohort C	1500	18	✓	✓	✓	✓	✓	✓
Country C	Cohort D	500	16	✓	×	✓	×	✓	✓
Country D	Cohort E	1000	16	✓	×	✓	✓	✓	×
	Cohort F	2500	14	✓	×	✓	✓		
Total				**8500**	**4500**	**6500**	**6000**	**5000**	**4000**

Variable Colour Legend	
	Age
	Sex
	Comorbidity
	Smoking history
	Symptoms
	Obesity
	Blood lipid

Table Legend	
✓	Available
×	Not available
	Pending arrival

Thank you

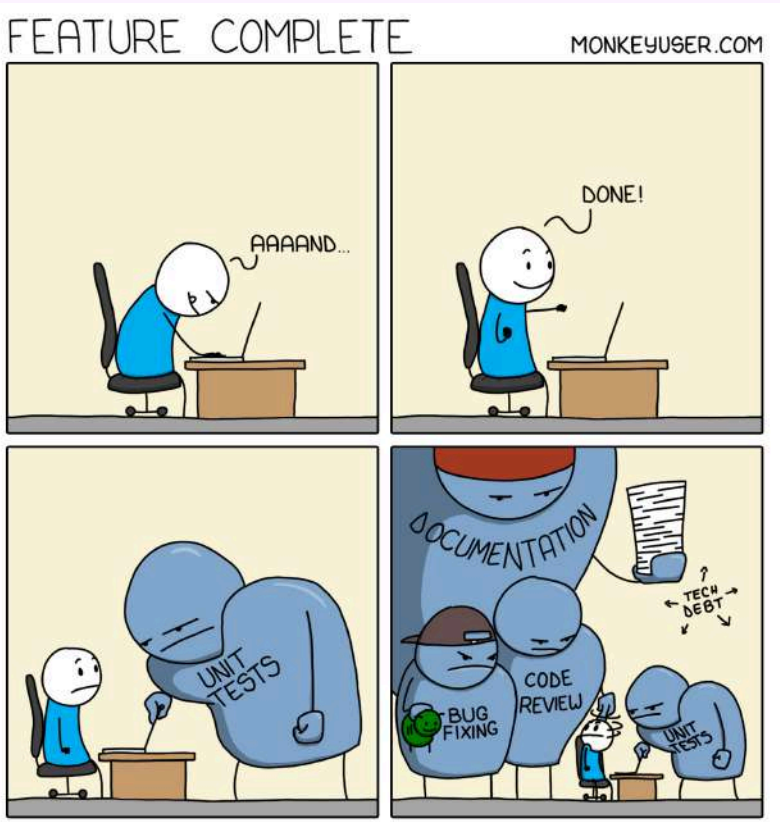
Harmonisation project template: https://github.com/JauntyJJS/harmonisation/

README License MIT license

Data Harmonisation Project Template

R-CMD-check.yaml passing

Table of Content



Feature Complete from MonkeyUser.com