

Retrospective clinical data harmonisation Reporting Using R and Quarto

Jeremy Selva [in](#)

@JauntyJJS

<https://jeremy-selva.netlify.app>

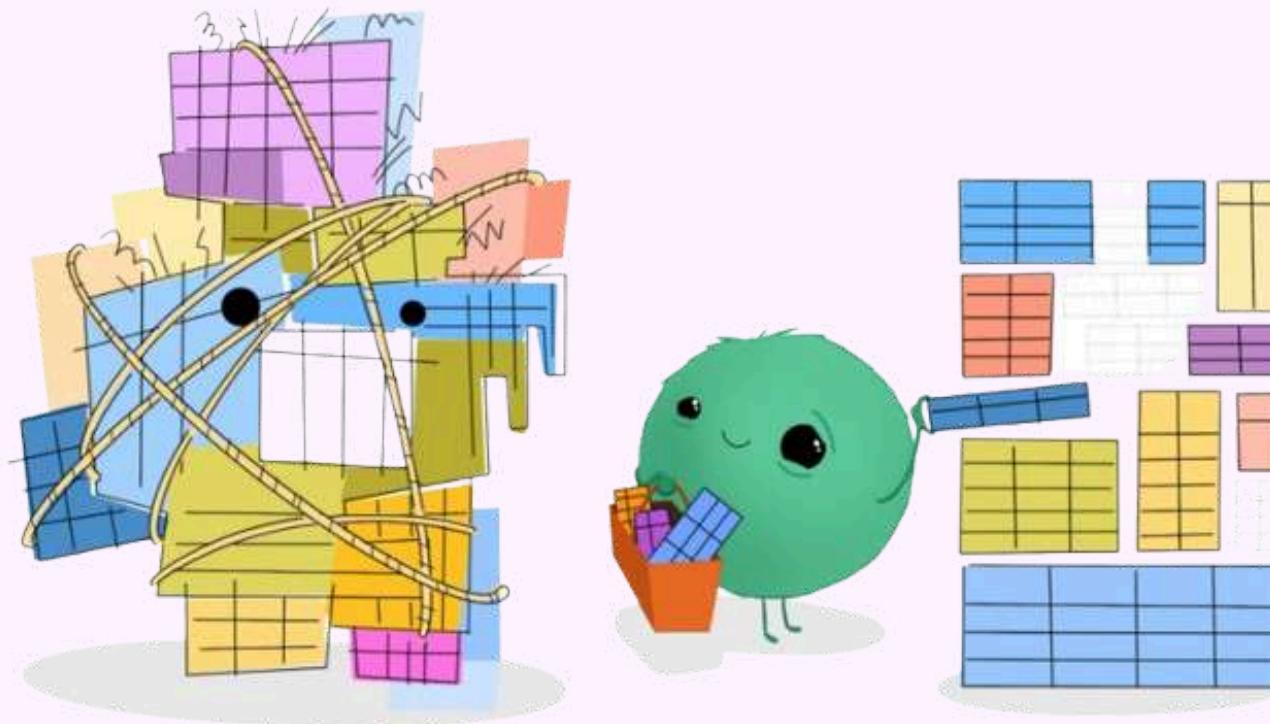
For Harvest Analytics Together (HAT) 2025

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.

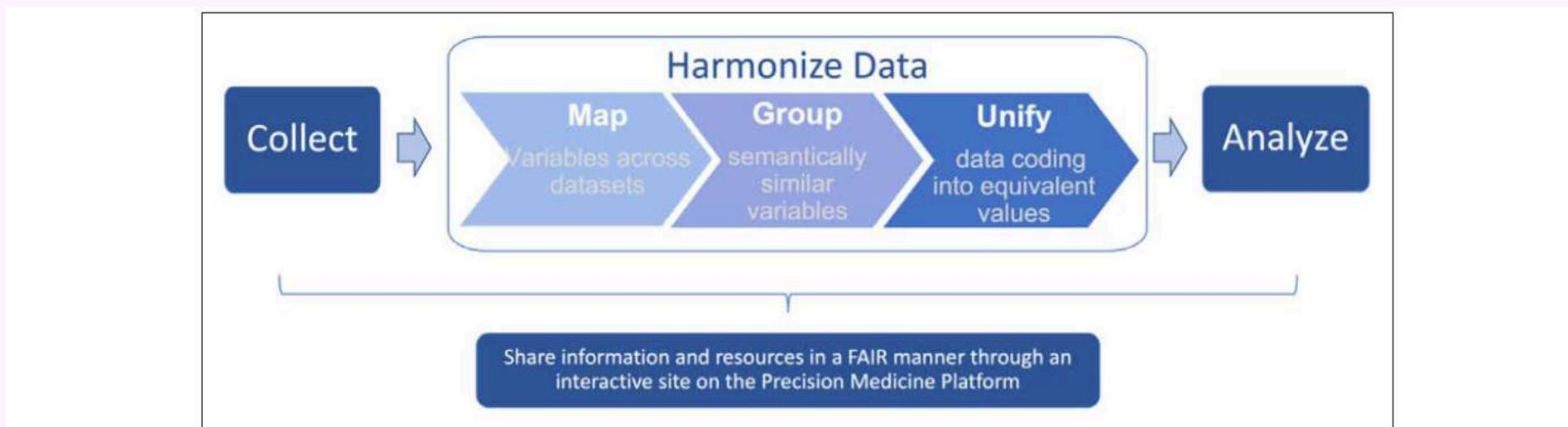
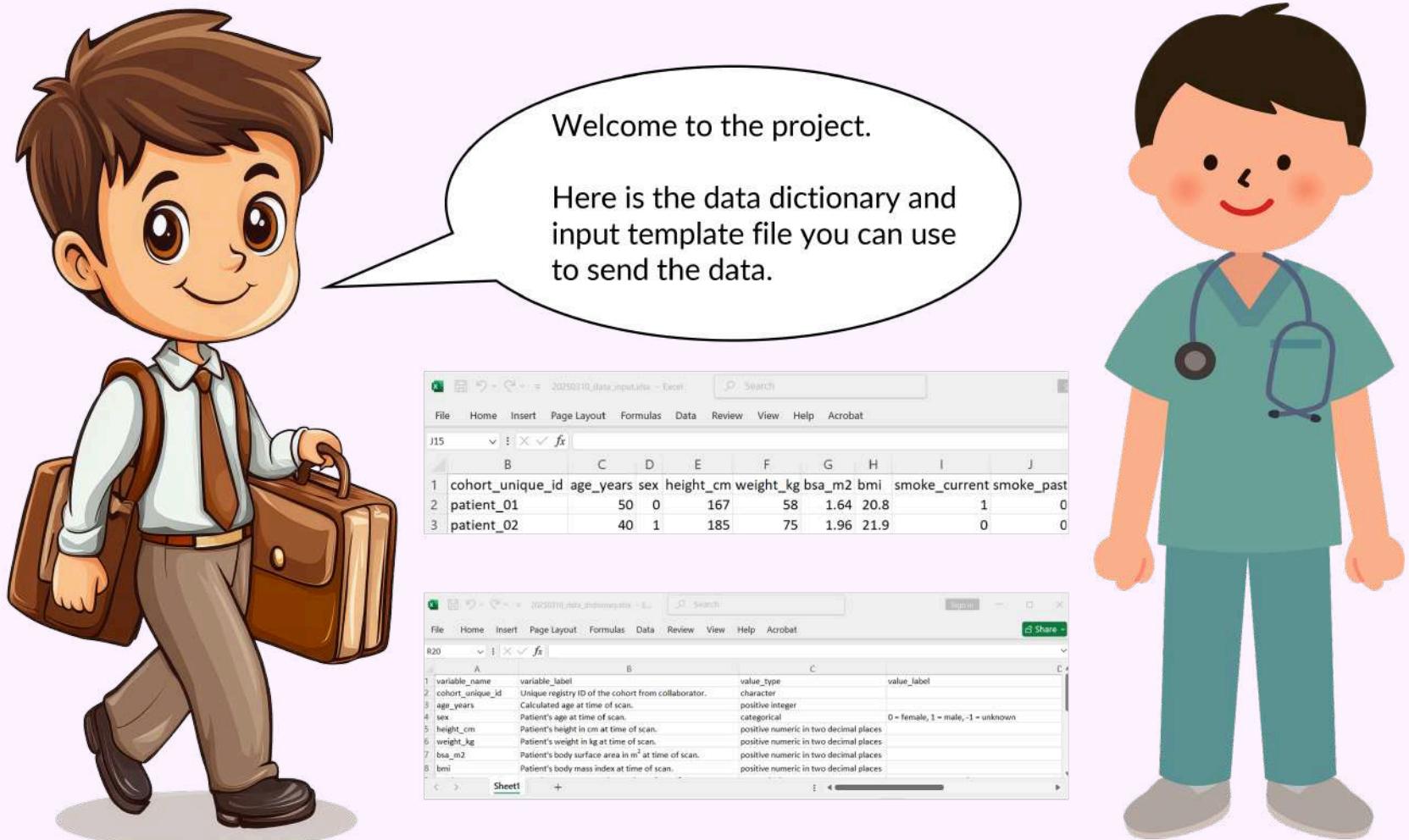


Figure 1. The data harmonization process.

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](https://doi.org/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



Received with thanks.

We don't have an analyst to do the mapping.

We can do it ourselves but our workload allow us to work on one data field per day...

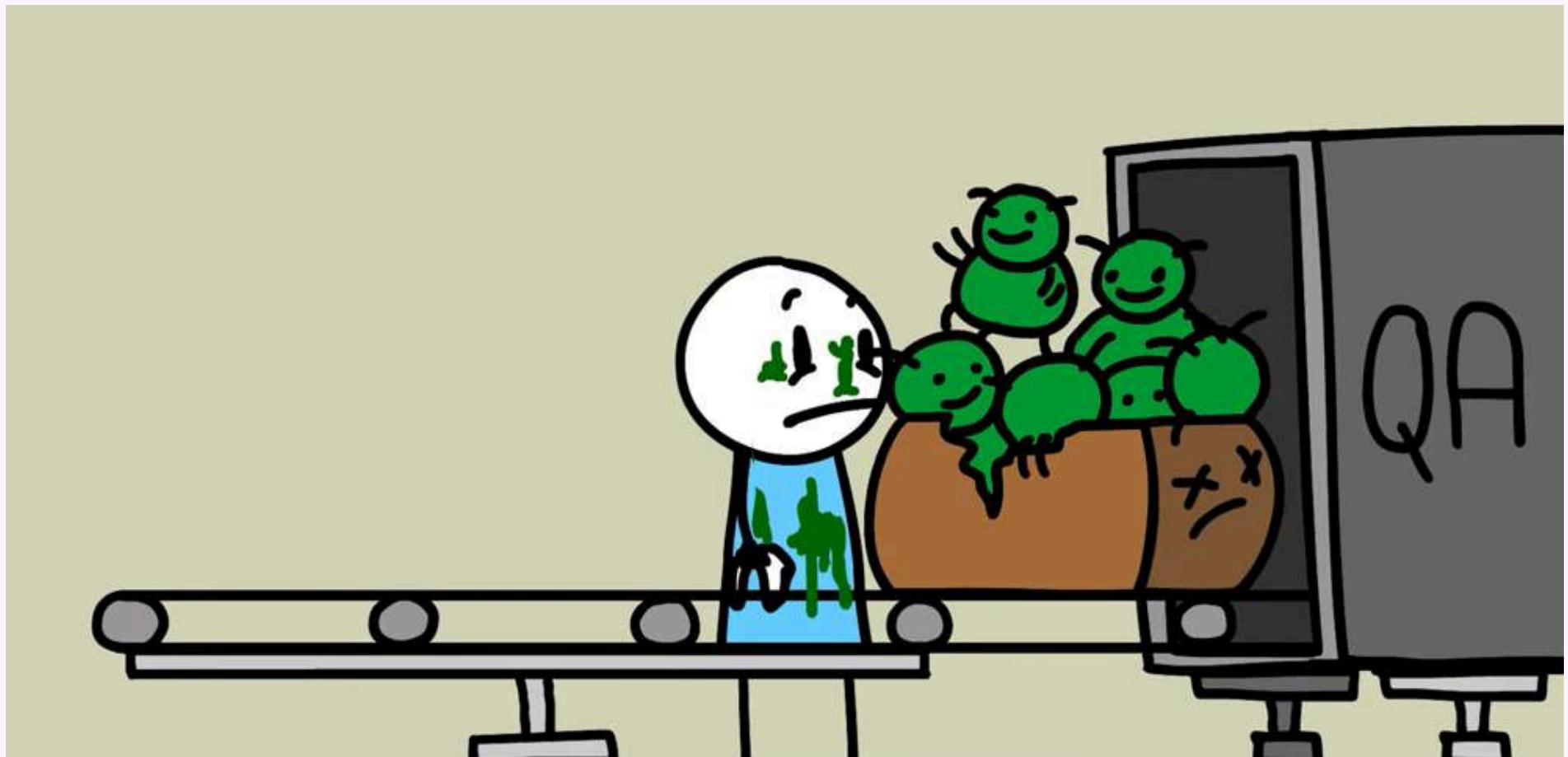


How it started



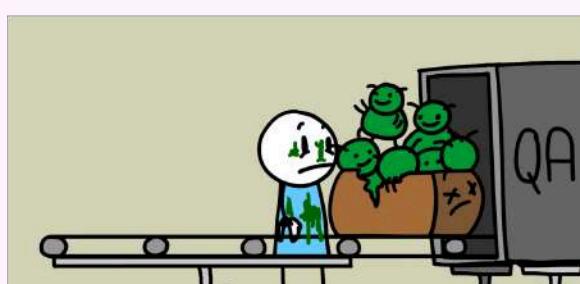
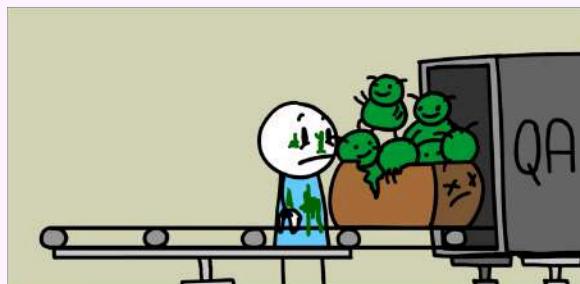
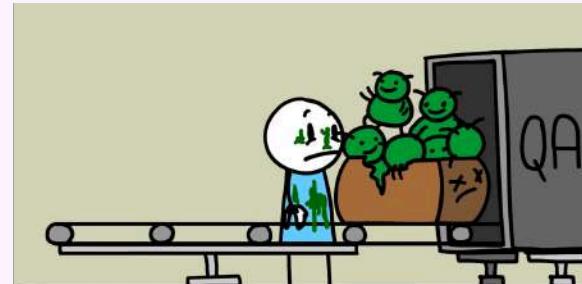
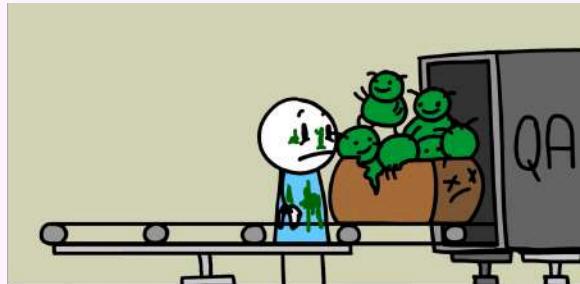
[Cheerful Businessman](#) designed by [Iftikhar Alam](#) from [Vecteezy](#) and [Medical Doctor Man](#) from [Creazilla](#).

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> 

For UseBI 2024 

Why a harmonisation report



Could you also send the harmonised data back to us with a report on how it is done ?

Our higher management needs it for an audit to show that the data is reliable.



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 4 Race/Ethnicity in text
Latino
White
Asian
Middle Eastern
Asian
Asian
White
Asian
Asian
Latino
Middle Eastern
African
etc ...

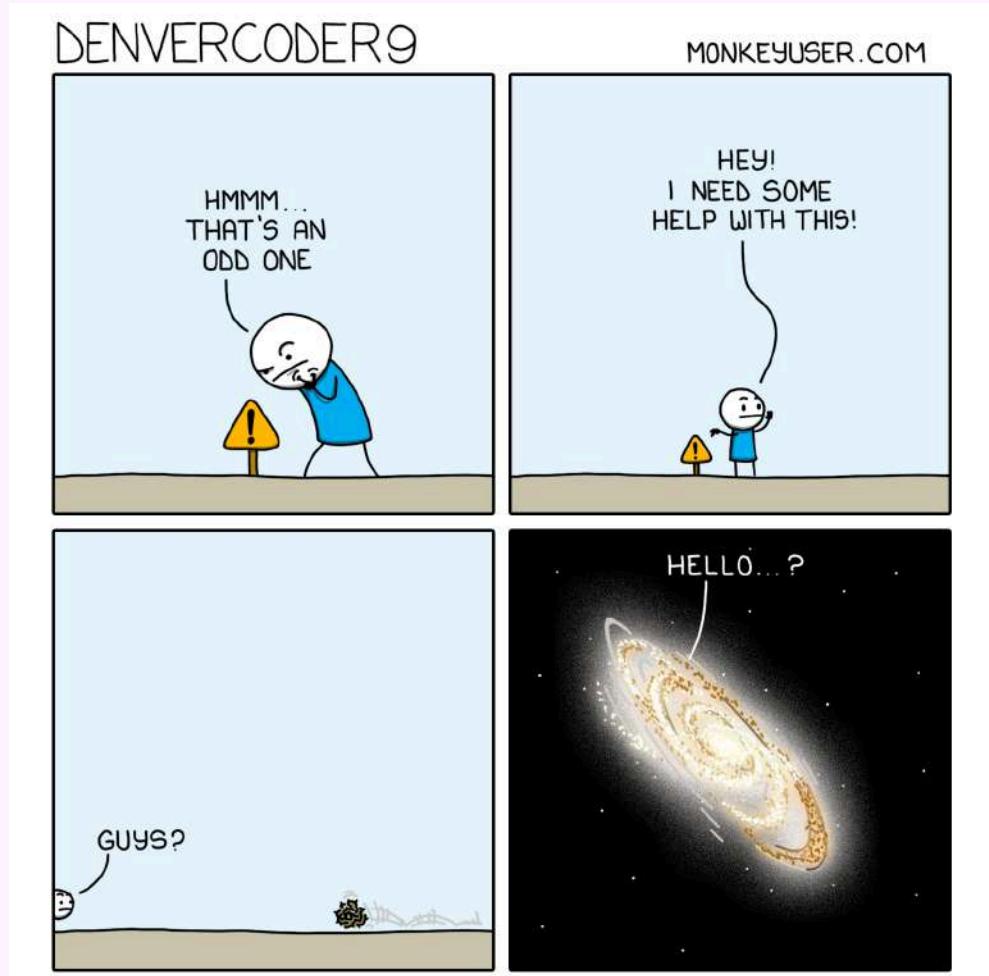
New Cohort Race/Ethnicity
White
African
Southeast Asian
East Asian
South Asian
Other Asians
Middle Eastern
Torres Straits Islanders
Aboriginal
Others

Issues

While there are   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/>]

harmonisation 1.0.0.0 Reference

Search for



Data Harmonisation Project Template

Table of Content

- [Motivation](#)
- [Acknowledgement](#)
- [File Structure](#)
- [Software Installation](#)
- [R Package Installation](#)
- [Using .renv](#)
- [R Functions Management](#)
- [R Packages Used](#)

Links

[Browse source code](#)

[Report a bug](#)

License

[Full license](#)

[MIT + file LICENSE](#)

Citation

[Citing harmonisation](#)

Developers

Jeremy Selva

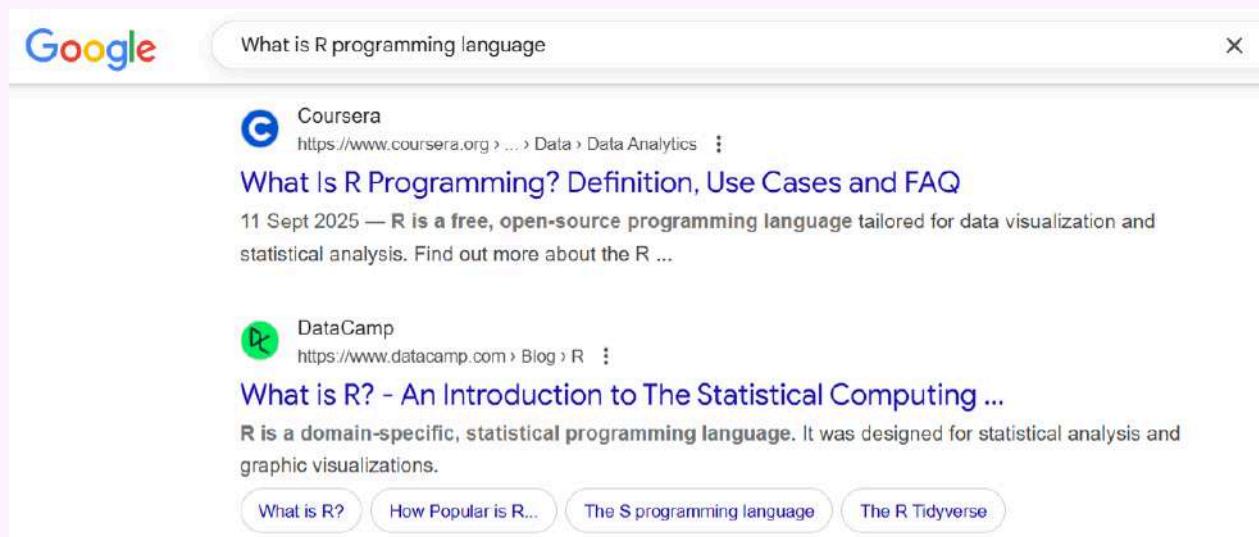
Author, maintainer A small circular icon with the letters "ID".

Tools to create documentation



[R Programming Logo](#) from [CleanPNG](#) and [Quarto Hex Sticker](#) from [Posit](#).

What *Did* We Forget to Teach You about ?



Everything else



Statistical
analysis

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 |>

```
1 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)

1 data_after_task_1 <- task_1(data, arg_1_2)
2
3 data_after_task_2 <- task_2(data_after_task_1, arg_2_2, arg_2_3)
4
5 data_after_task_3 <- task_3(data_after_task_2, arg_3_2, arg_3_3)
```

With |>

```
1 data_after_task_3 <- data |>
2   task_1(arg_1_2) |>
3   task_2(arg_2_2, arg_2_3) |>
4   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+   [magrittr](#) pipe `%>%`
- 2021+ ( $\geq 4.1.0$) native  pipe `|>`

↳ More details between the two pipes in [Understanding the native R pipe |>.](#)

Isabella Velásquez 

About Blog Talks Projects Today I Learned

Understanding the native R pipe `|>`

EXPLANATION

Or, why `mtcars |> plot(hp, mpg)` doesn't work and what you can do about it.

PUBLISHED

January 18, 2022



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)
```

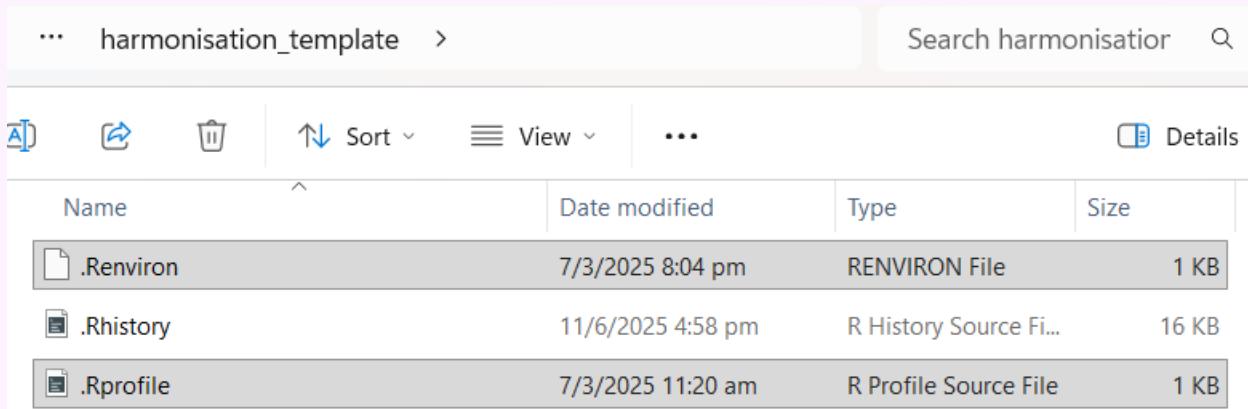
R Session

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.

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



Name	Date modified	Type	Size
.Renvironment	7/3/2025 8:04 pm	RENVIRON File	1 KB
.Rhistory	11/6/2025 4:58 pm	R History Source Fi...	16 KB
.Rprofile	7/3/2025 11:20 am	R Profile Source File	1 KB

what goes in `.Renviron`

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

```
1 APPDATA="D:/Jeremy/PortableR/RAppData/Roaming"
2 LOCALAPPDATA="D:/Jeremy/PortableR/RAppData/Local"
3 TEMP="D:/Jeremy/PortableR/RPortableWorkDirectory/temp"
4 TMP="D:/Jeremy/PortableR/RPortableWorkDirectory/temp"
5 _R_CHECK_SYSTEM_CLOCK_=0
6 RENV_CONFIG_PAK_ENABLED=TRUE
7 CONNECT_API_KEY=DaYK2hBURiSBYUEGIAiyXsRJHSjTYJN3
8 DB_USER=elephant
9 DB_PASS=p0stgr3s
```

user

`~/.Renviron`

project

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

what goes in .Rprofile

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

```
1 source("renv/activate.R")
2 options(
3   repos = c(
4     P3M_20250306 = "https://packagemanager.posit.co/cran/2025-10-13",
5     ropensci = "https://ropensci.r-universe.dev",
6     janmarvin = "https://janmarvin.r-universe.dev",
7     CRAN = 'https://cloud.r-project.org'
8   )
9 )
10 if (interactive()) prompt::set_prompt(prompt::prompt_fancy)
```

Examples: <https://github.com/search?q=.Rprofile&type=repositories>

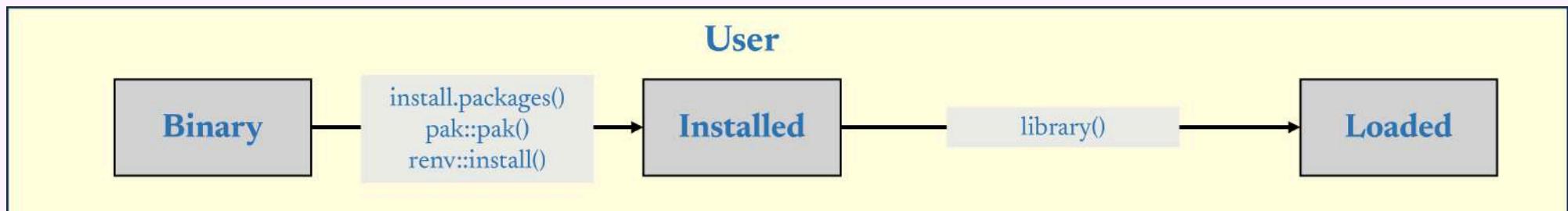
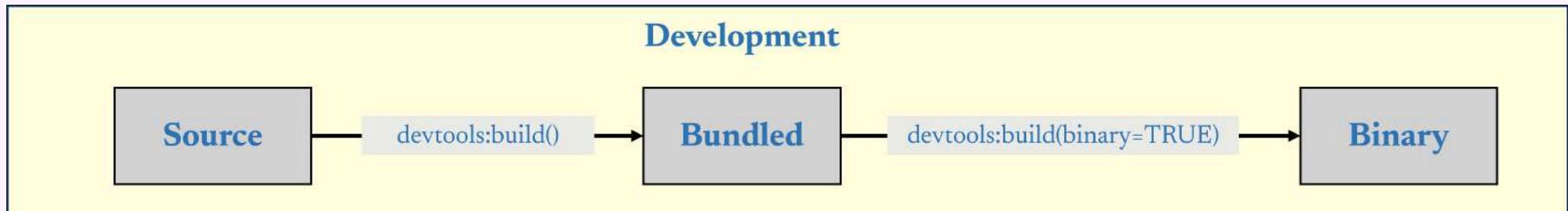
 [csgillespie/rprofile](#).

 [Me, Myself and my Rprofile](#).

 [Customising your Rprofile](#).

States of R

R - Structure



Binary R 📦 Installation

Go for (Windows | macOS) 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](https://doi.org/10.1016/j.jacc.2020.09.585)>, 2019 Reeh et al. basic and clinical models <[doi:10.1093/eurheartj/ehy806](https://doi.org/10.1093/eurheartj/ehy806)> and 2017 Fordyne et al. PROMISE Minimal-Risk Tool <[doi:10.1001/jamacardio.2016.5501](https://doi.org/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](https://doi.org/10.32614/CRAN.package.pretestcad)
Author: Jeremy Selva  [aut, cre]
Maintainer: Jeremy Selva <jeremy1189.jjs@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.

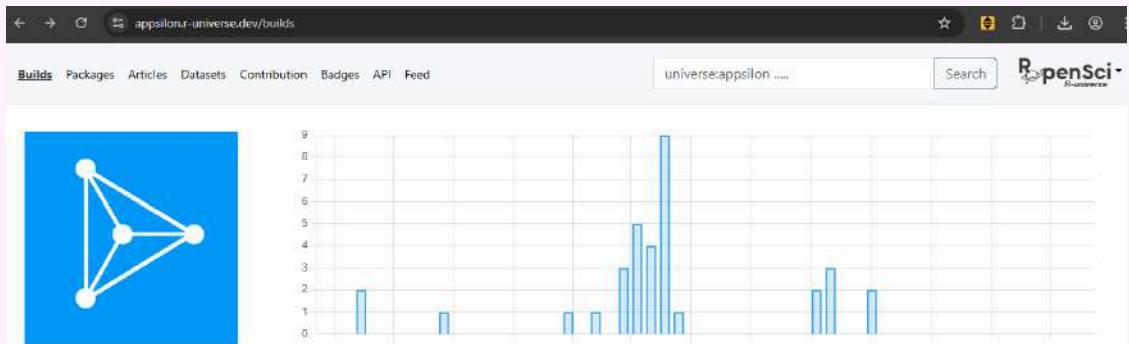
R-universe and Posit Public Package Manager

Consider installing (Windows | macOS | Linux) binaries from

- [R-universe](#)
- [Posit Public Package Manager.](#)

Set in your **.Rprofile** file.

```
1 options()  
2 repos = c(  
3   P3M_20250306 = "https://packagemanager.posit.com/r/universe",  
4   ropensci = "https://ropensci.r-universe.dev",  
5   CRAN = 'https://cloud.r-project.org'  
6 )  
7 )
```



Appslon	Commit	Package	Version	Maintainer	Src	R-dev R-release R-old	Built
Join a World-Class Team of Explorers (earth_americas: We're hiring!)	2025-07-15	box.inters	0.10.6.9000	Ricardo Rodrigo Basa			4 days ago
	2025-07-14	shiny.telemetry	0.3.1.9002	André Veríssimo			10 days ago
	2025-04-02	rhino	1.11.0.9000	Kamil Zyla			16 days ago



Welcome to Posit Public Package Manager

The best way to discover and install R and Python packages

Repository: cran Packages in the cran repository SETUP

3 repositories 30,300 CRAN packages 750,556 PyPI packages

How do I know I got a binary?

CRAN 

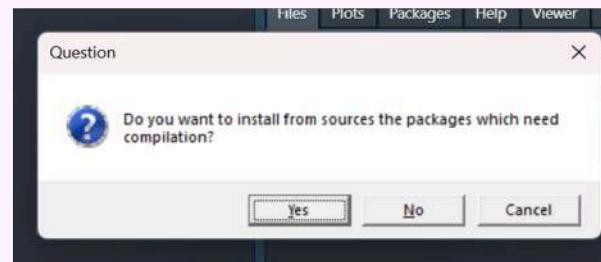
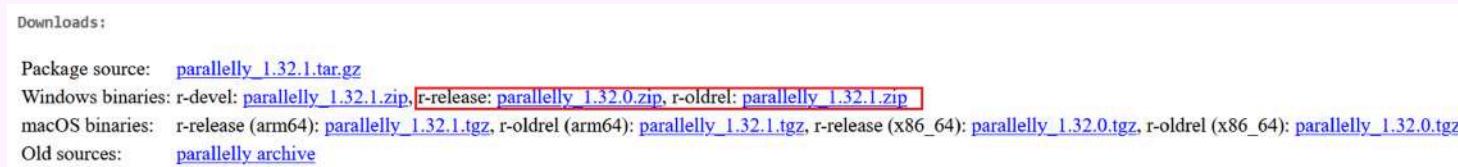
CRAN 

p3m

```
1 > install.packages("parallelly", repos = "https://cran.r-project.org")
2 Installing package into 'C:/Users/edavi/Documents/R/win-library/4.1'
3 (as 'lib' is unspecified)
4 trying URL 'https://cran.r-project.org/bin/windows/contrib/4.1/parallelly_1.32.1.zip'
5 Content type 'application/zip' length 306137 bytes (298 KB)
6 downloaded 298 KB
7
8 package 'parallelly' successfully unpacked and MD5 sums checked
9
10 The downloaded binary packages are in
11     C:/Users/edavi/AppData/Local/Temp/Rtmpa2s3e8/downloaded_packages
```

Source Installation

Go for source  if you need the latest version urgently.



 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 pak

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

Getting code to production > 6 Package installation

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:

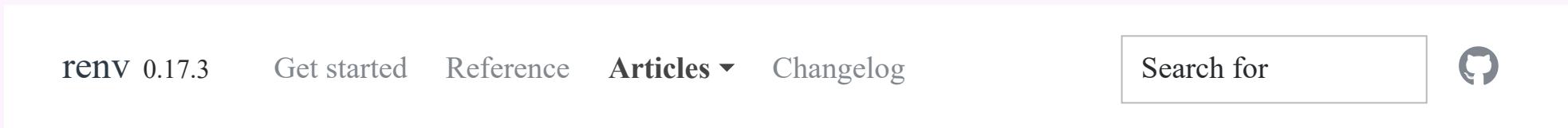
1. You're probably most used to installing packages on a Mac or Windows computer. There are some important differences with Linux and to understand them, you'll need some new vocabulary like binary packages and system libraries.
2. Production jobs are usually run in a throwaway container. That means 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   to create isolated project environments.

-  [You should be using renv.](#)
-  [Making your R project future-proof with renv.](#)



A screenshot of the renv website. At the top left, it shows "renv 0.17.3". To its right are navigation links: "Get started", "Reference", "Articles ▾", and "Changelog". Further to the right is a search bar with the placeholder "Search for" and a GitHub icon. The main content area below the header is currently empty.

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;
- The R repositories that were active when the lockfile was created.

“virtual environment” using renv

Some advice ...

- Set `RENV_CONFIG_PAK_ENABLED=TRUE` in the `.Renvironment` file for `renv::install()` uses `pak` at the backend to install  .
- In an existing project, use `renv::init(bare = TRUE)` to initiate renv with an empty  library and then install   manually.
- 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\(\)`.](#)

New Kid for R management Require

Require 1.0.1 GET STARTED REFERENCE CHANGELOG

Search for



Require

CRAN 1.0.1 downloads 105K R-CMD-check failing

`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 it *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

Links

[View on CRAN](#)

[Browse source code](#)

[Report a bug](#)

License

[GPL-3](#)

Community

[Contributing guide](#)

Citation

[Citing `Require`](#)

Developers

Personal R Administration

P <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   [rcompendium](#) but there are others ([orderly](#), [prodigenr](#) and [workflowr](#)) as well.

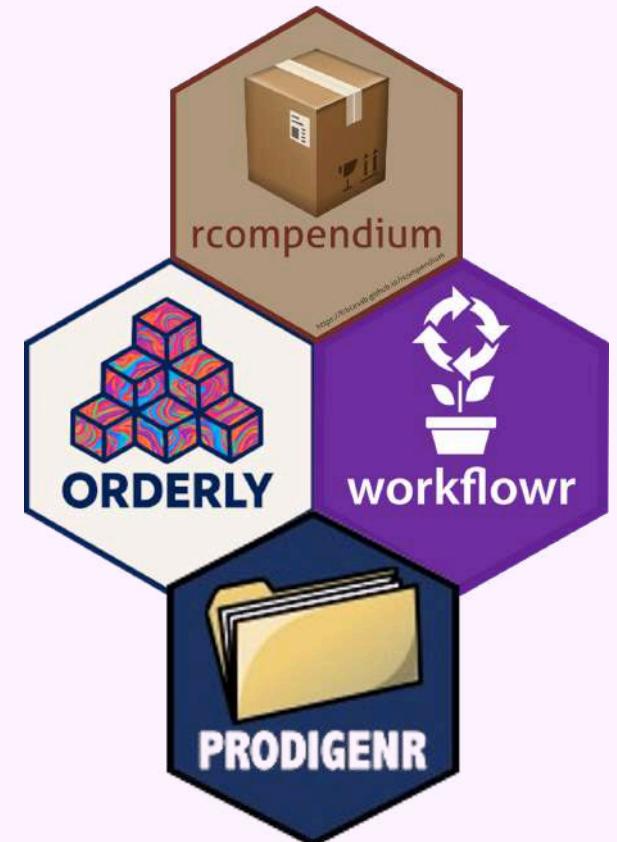
rcompendium 1.4

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 `NAMESPACE` files, and `R/` and `man/` folders).
- Creation of additional files (`LICENCE` and `FAIR/ CITATION`, etc.)



If the top of your script is

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

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

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

Tidyverse

Packages

Blog

Learn

Help

Contribute

Project-oriented workflow



Photo by secumem

Practise “safe paths”

QR  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 interface, is partially inspired by Rust’s [fs module](#).

Installation

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

Install the released version of here from CRAN:

```
install.packages\("here"\)
```

Practise “safe paths”

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

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

User’s home directory

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

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

 Project directory

```
1 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  do it for you.

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

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

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

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

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

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

 Use relative path within the  project directory.

```
1 readxl::read_excel(path = here::here("data-folder", "data.xlsx"))
2 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())`

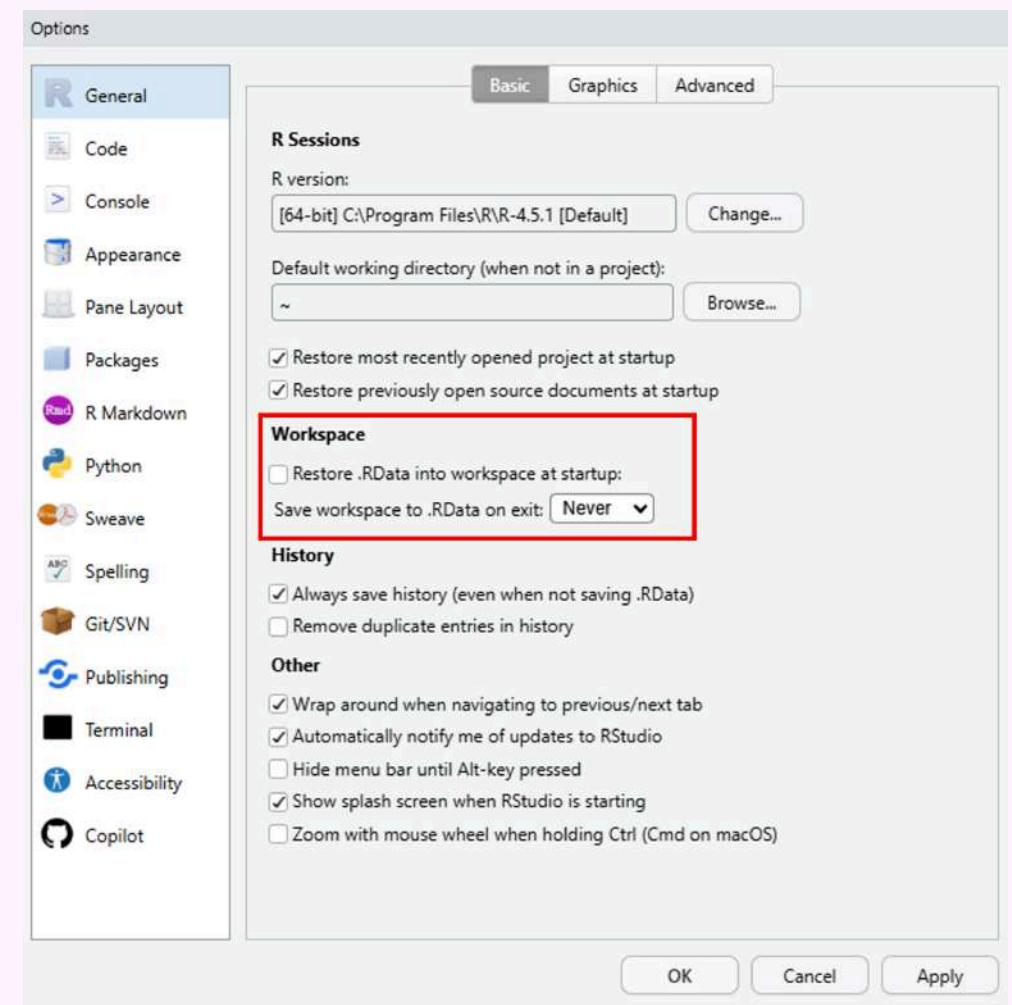
Which persist after <code>rm(list = ls())</code> ?	
Option	Persists?
A. <code>library(dplyr)</code>	✓
B. <code>summary <- head</code>	✗
C. <code>options(stringsAsFactors = FALSE)</code>	✓
D. <code>Sys.setenv(LANGUAGE = "fr")</code>	✓
E. <code>x <- 1:5</code>	✗
F. <code>attach(iris)</code>	✓

Slide from [Project oriented workflows](#).

✓ Use `usethis::use_blank_slate()`

```
R console
usethis::use_blank_slate()
```

✓ Tools -> Global Options in RStudio.



Project Oriented Workflow

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

Project oriented workflows

Shannon Pileggi

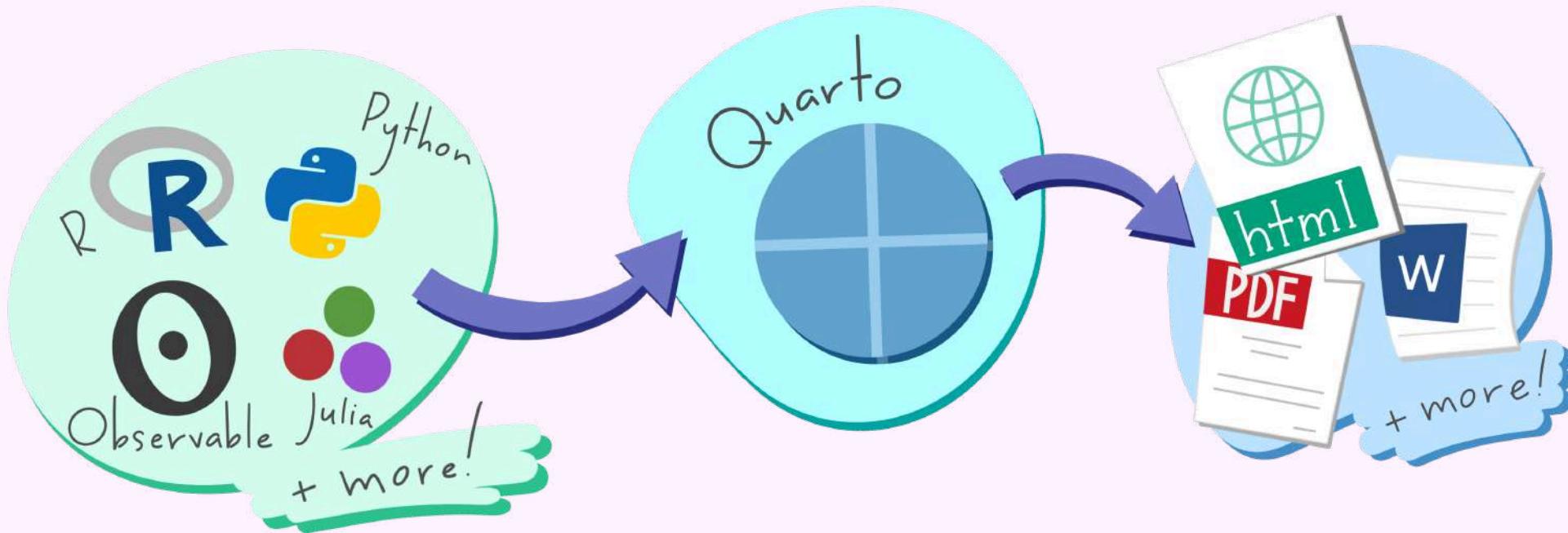
Getting started

Project oriented workflows
Licensing



Quarto

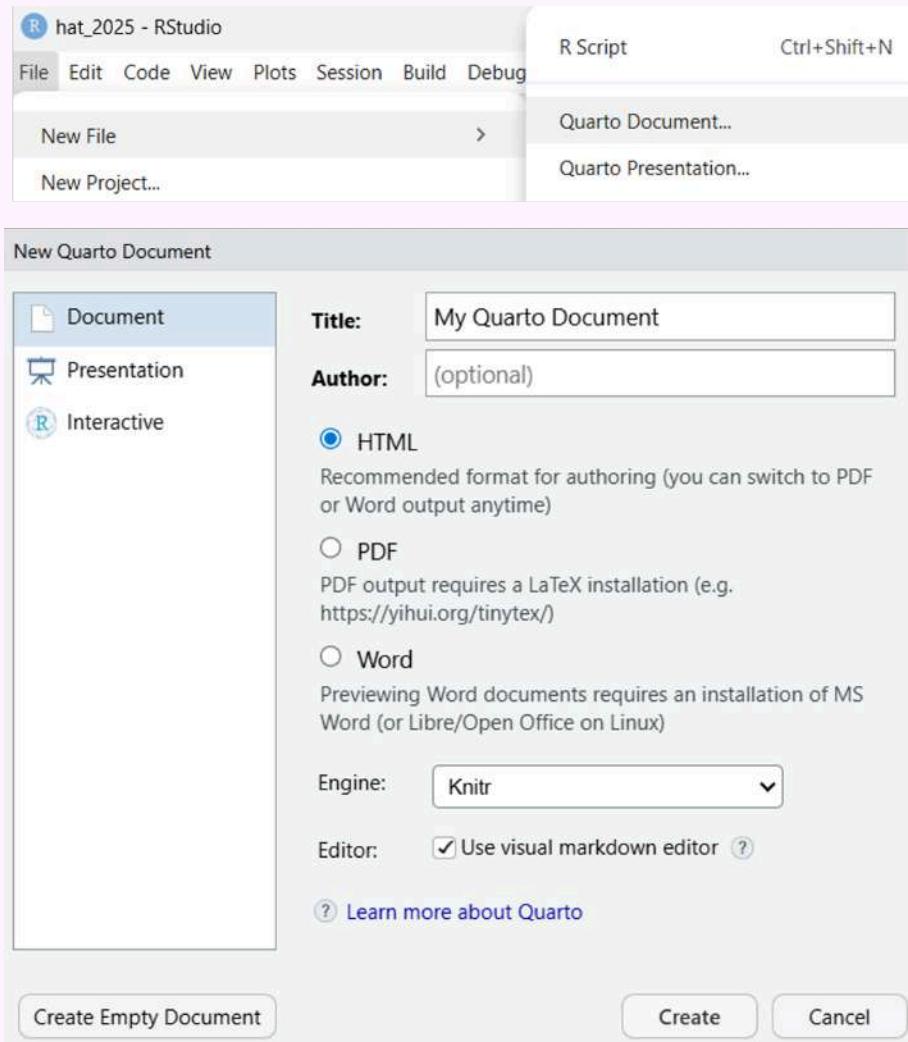
[Quarto](#) is an open-source software that weaves narrative and programming code together to produce elegantly formatted output as documents (in HTML, Word, PDF), presentations, books, web pages, and more.



[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

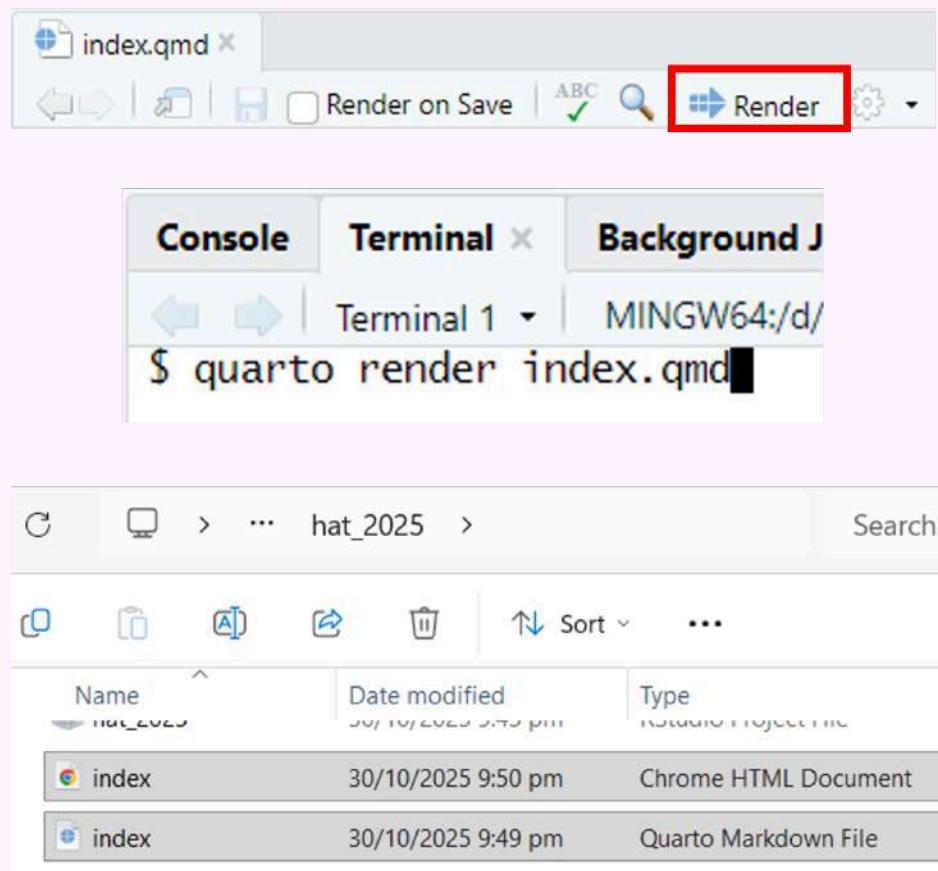
Create a Quarto document



The screenshot shows the RStudio Source editor with the file "Untitled1*". The code is written in Quarto format:

```
1 ---  
2 title: "My Quarto Document"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## Quarto  
8  
9 Quarto enables you to weave together content and executable code  
into a finished document. To learn more about Quarto see  
<https://quarto.org>.  
10  
11 ## Running Code  
12  
13 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:  
14  
15 ````{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ````{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only  
output is displayed).  
27
```

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:

The diagram illustrates the structure of a Quarto file. On the left, a sidebar shows the navigation menu for "R for Data Science (2e)". The main content area displays a Quarto document with the following structure:

- An (optional) YAML header surrounded by `---`'s.** (Purple box, top-left)
- Chunks of R code surrounded by ```'s.** (Green box, middle-left)
- Text mixed with markdown text formatting.** (Blue box, bottom-right)
- We have data about `r nrow(diamonds)` diamonds.
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.
The distribution of the remainder is shown below:** (Dark blue box, bottom-left)

Arrows point from the corresponding code snippets in the Quarto file to their respective descriptions. A blue arrow points from the text at the bottom to the "Text mixed with markdown text formatting" box.

```
---  
title: "Diamond sizes"  
date: 2022-09-12  
format: html  
---  
  
```{r}  
#| label: setup
#| include: false

library(tidyverse)

smaller <- diamonds |>
 filter(carat <= 2.5)
```  
  
We have data about `r nrow(diamonds)` diamonds.  
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.  
The distribution of the remainder is shown below:
```

Simple Quarto file example from [R for Data Science \(2e\) Chapter 28 Quarto](#)

Quarto Level 1

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

Klajdi Puka, PhD About Research Teaching Consulting Services Resources CV/Resume

R > Introduction to Quarto

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.

The screenshot shows a Microsoft Word document with a light gray background. At the top center, there is a logo consisting of a dark gray square with the white text "CRIME" and "De-Coder" stacked vertically. On the left side, the title "EXAMPLE TEMPLATE REPORT" is displayed in large, bold, black capital letters. Below the title, the subtitle "SUPERCOOL SUBTITLE" and the author's name "Andrew P. Wheeler, PhD" are listed in smaller black text. On the right side, a table of contents is shown in a standard Word style:

| Table of contents | |
|----------------------------|---|
| Introduction Section | 3 |
| Section 2 | 3 |
| A subsection! | 5 |
| Footnotes | 6 |
| A Superlong table | 7 |
| Mathy | 8 |
| Reference Notes | 8 |
| ToDo! | 8 |
| My References | 9 |

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



Christopher T. Kenny

[Research](#) [Teaching](#) [Software](#) [CV](#) [Quarto](#) [Blog](#)



Quarto Extensions

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

PUBLISHED

October 31, 2025

Journal templates

Templates for general science and social science journals.

Quarto Title for Annual Reviews

Instructions to Authors: Political Science Review
DOI: 10.1215/0032319X-108-1
Quarto Title for the APSR

AUTHOR ONE *An Organization*
AUTHOR TWO *Affiliation B*
AUTHOR THREE *A Third Organization*

This document is a template demonstrating the APSR journal. Make sure it is large enough to print on the standard desktop. The overall size need is four times four inches square. That is a lot of space for one example, but the APSR need 150 words in the abstract at the time of writing this example.

Word Count: 771

INTRODUCTION

This Quarto template is useful for using Quarto to write your article. This Quarto template is useful and based on Overleaf's APSR template. Your introduction goes here! Do make sure the first paragraph is a little less than six lines long, to accommodate the dropcap line. Some sections of content also need to be included in the document.

each other, though it's likely best to modify use, mathematical.

Use section (#) and subsection (##) commands to organize your document. \LaTeX and Quarto handle all the formatting and numbering automatically. Use the \# commands for cross-references.

Cambridge Journals Online © 2008 3–2

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Cambridge-Medium Template

F. Author,¹* S. Author,² T. Author,³ and P.T. Author^{4,5}

¹First Division, Organization, Boxes 4210–4212, USA
²Second Division, Organization, Boxes 2250, USA
³Third Division, Organization, New York, 10012, USA
⁴International Division, Organization, Chicago, 60607, USA
⁵Corresponding author. Email: Author@Cambridge.org

Abstract
This document is a template demonstrating the Cambridge-Medium Series.

Keywords: template, clean

Quarto Template for Springer Nature

Author One^{1,*}, Author Two^{2†}, Author Three³

¹Department of Government, Harvard University, 1775 Cambridge Street, Cambridge, 02138

²Department of Statistics, Harvard University, 1 Oxford Street, Cambridge, 02138

³Department of Political Science, Yale University, 113 Prospect Street, New Haven, 06511

*Corresponding author(s). E-mail(s): corresponding@email.edu

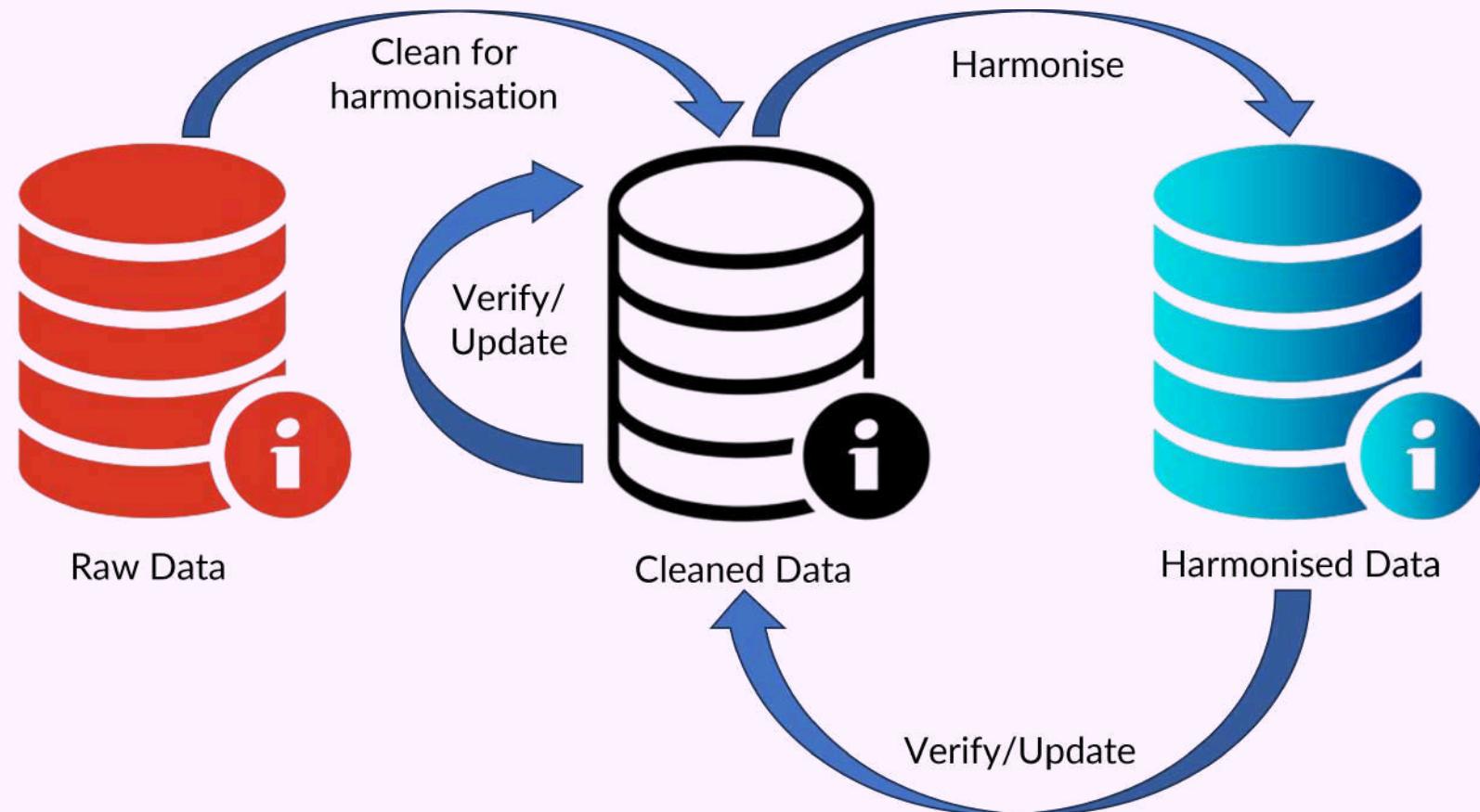
[†]These authors contributed equally to this work.

Abstract

The abstract section holds a general introduction to the logic and scope of a broad, multi-faceted taxonomy of the most recent and far-reaching findings. Authors are encouraged to include a brief summary of the main findings and their implications.

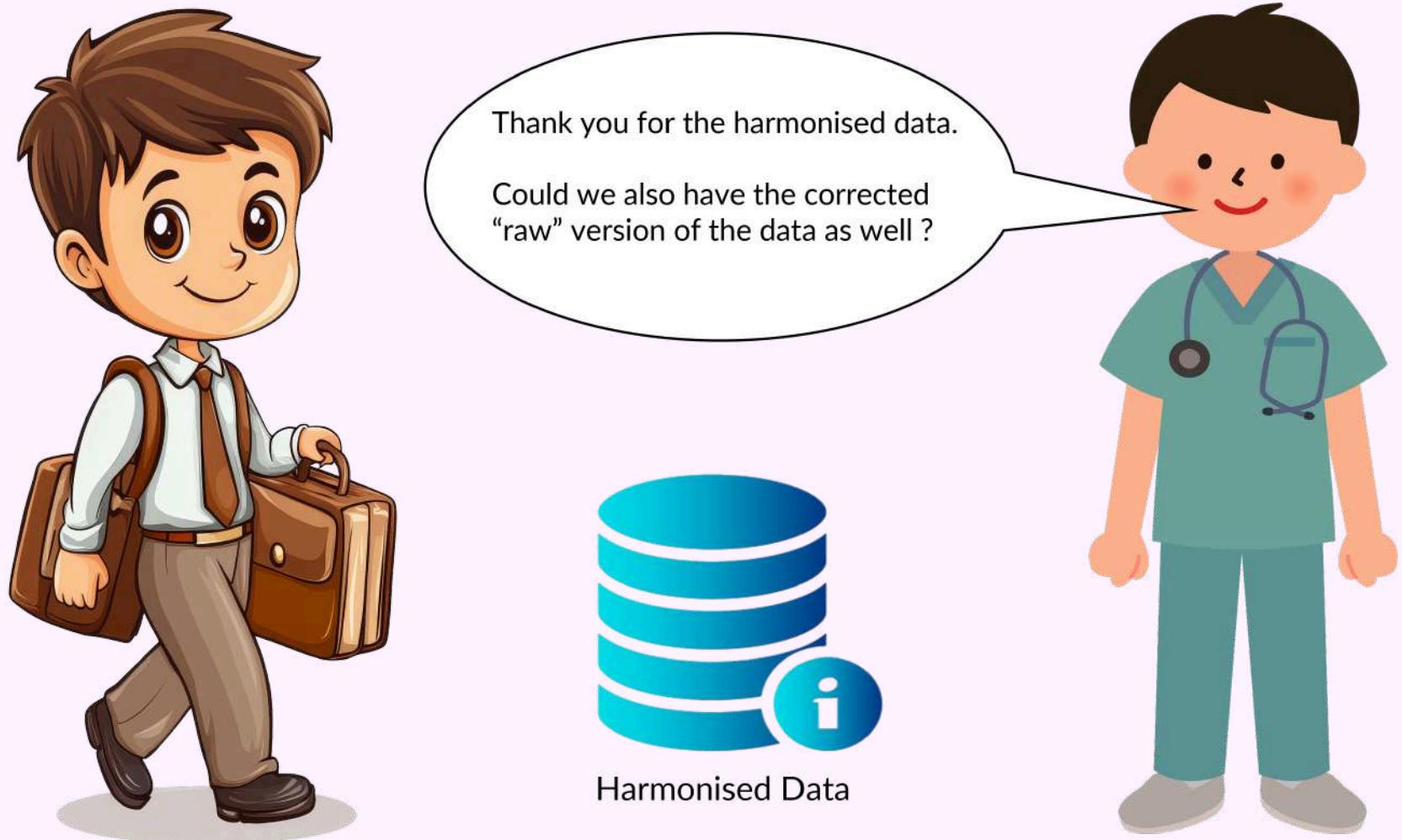
Workflow with collaborators

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



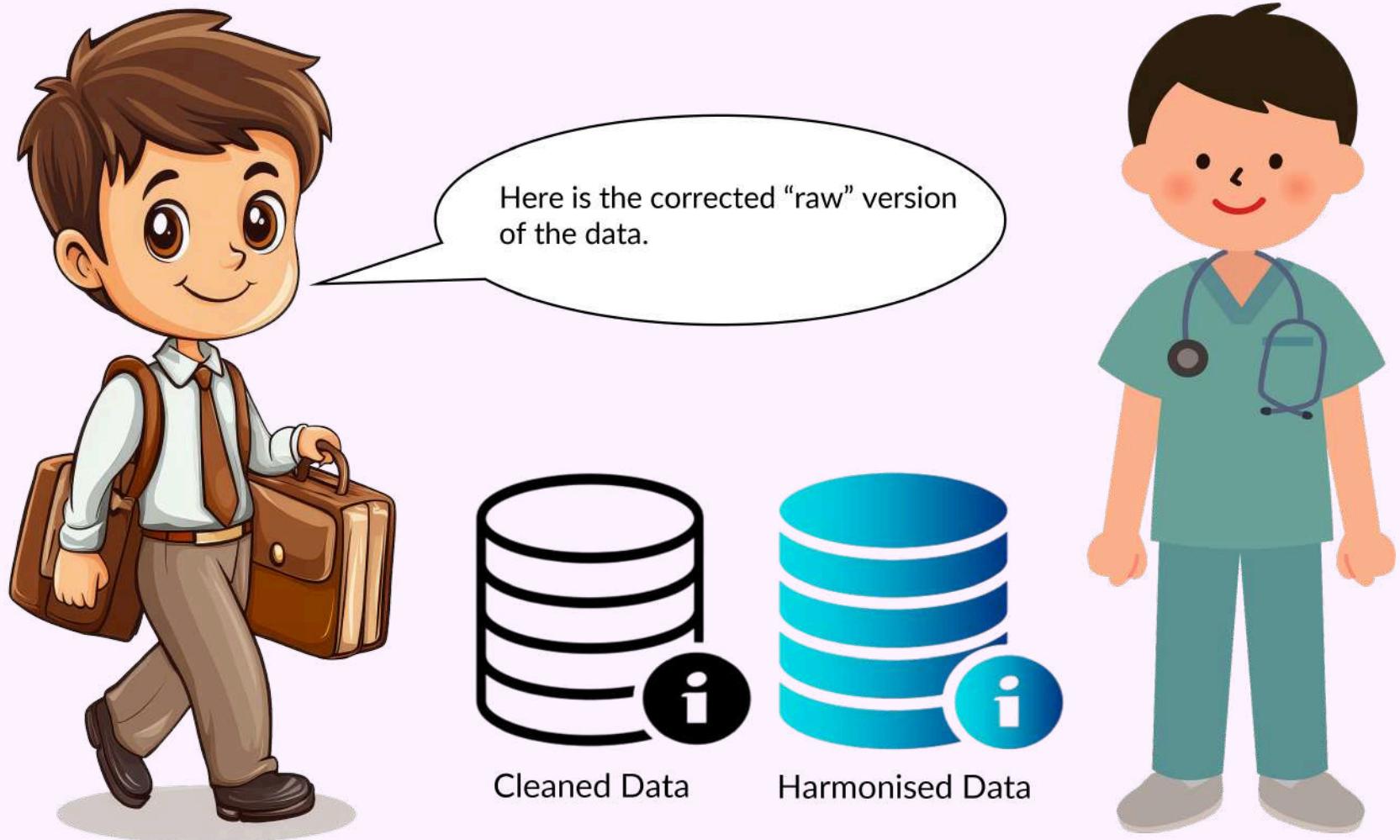
[Dataset vectors](#) by [Vectora Artworks](#) from [Vecteezy](#).

Workflow with collaborators



[Cheerful Businessman](#) designed by [Iftikhar Alam](#) from [Vecteezy](#) and [Medical Doctor Man](#) from [Creazilla](#).

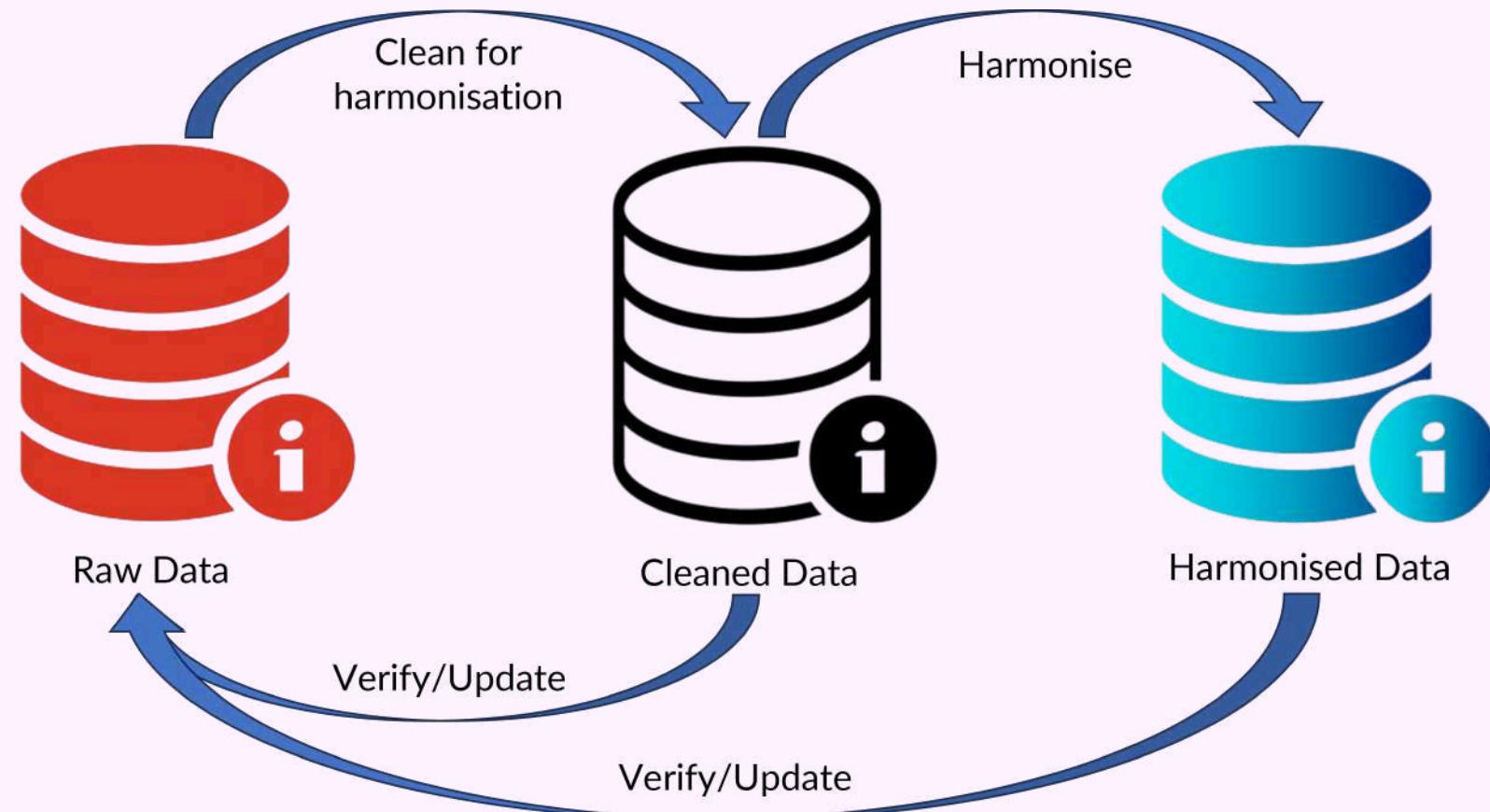
Workflow with collaborators



[Cheerful Businessman](#) designed by [Iftikhar Alam](#) from [Vecteezy](#) and [Medical Doctor Man](#) from [Creazilla](#).

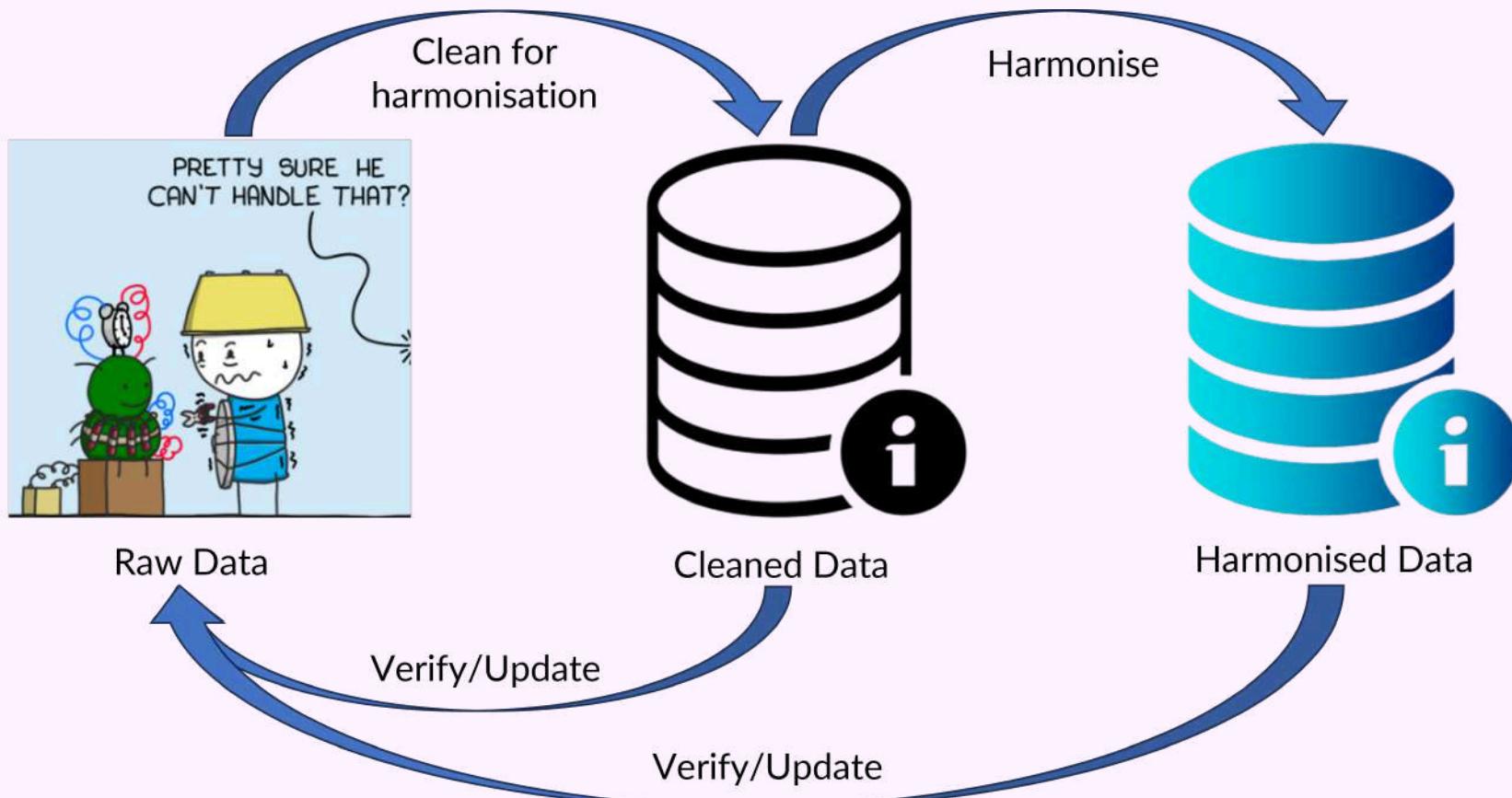
Workflow with collaborators

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



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 ?



Automated capture of warnings (csv)

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

```
1 cohort_data_csv <- vroom::vroom(  
2   file = here::here("data-raw", "Cohort_csv",  
3     "data_to_harmonise_age_issue.csv"),  
4   delim = ",",  
5   col_select = 1:2,  
6   show_col_types = FALSE,  
7   col_types = list(  
8     ID = vroom::col_character(),  
9     Age = vroom::col_integer()  
10    )  
11  )
```

```
1 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
```

| S10 | A | B | C | D | E | F |
|-----|------|---------|---|---|---|---|
| 1 | ID | Age | | | | |
| 2 | B001 | 32 | | | | |
| 3 | B002 | 52 | | | | |
| 4 | B003 | missing | | | | |
| 5 | B004 | 70 | | | | |
| 6 | B005 | 70 | | | | |
| 7 | B006 | 53 | | | | |
| 8 | B007 | 86 | | | | |
| 9 | B008 | 28 | | | | |
| 10 | B009 | missing | | | | |

Automated capture of warnings (csv)

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

```
1 cohort_data_csv |>  
2 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`.

```
1 cohort_data_csv |>  
2 vroom::problems() |>  
3 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.

```
1 cohort_data_csv <- vroom::vroom(  
2   file = here::here("data-raw", "Cohort_csv",  
3     "data_to_harmonise_age_issue_fixed.csv"),  
4   delim = ",",  
5   col_select = 1:2,  
6   show_col_types = FALSE,  
7   col_types = list(  
8     ID = vroom::col_character(),  
9     Age = vroom::col_integer()  
10    )  
11  )  
12  
13 cohort_data_csv |>  
14   vroom::problems()
```

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

```
1 cohort_data_csv |>  
2   vroom::problems() |>  
3   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 ?

```
1 cohort_data_excel <- readxl::read_excel(  
2   path = here::here("data-raw", "Cohort_Excel",  
3     "data_to_harmonise_age_issue.xlsx"),  
4   sheet = "Sheet1",  
5   col_types = c(  
6     "text", "numeric"  
7   )  
8 )
```

```
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`.

```
1 testthat::expect_no_condition(  
2   cohort_data_excel <- readxl::read_excel(  
3     path = here::here("data-raw", "Cohort_Excel",  
4       "data_to_harmonise_age_issue.xlsx"),  
5     sheet = "Sheet1",  
6     col_types = c("text", "numeric")  
7   )  
8 )
```

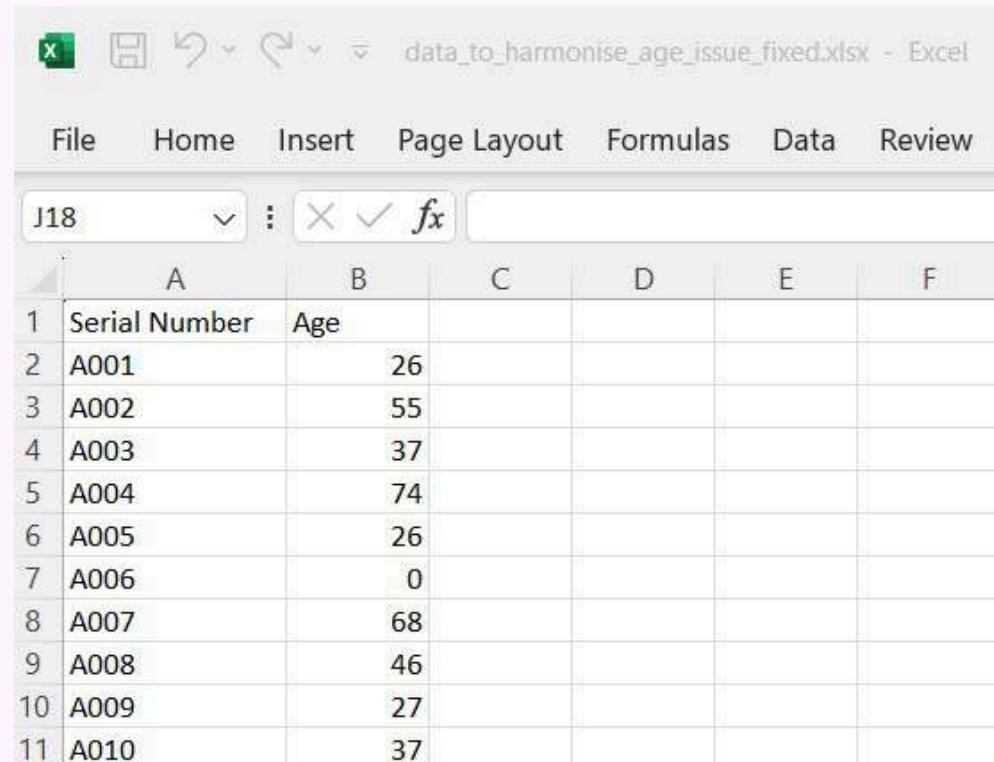
```
Error: Expected `... <- NULL` to run without any conditions.  
i Actually got a <simpleWarning> with text:  
  Expecting numeric in B7 / R7C2: got 'missing'
```

| | 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.

```
1 testthat::expect_no_condition(  
2   cohort_data_excel <- readxl::read_excel(  
3     path = here::here("data-raw", "Cohort_Excel",  
4       "data_to_harmonise_age_issue_fixed.xlsx"),  
5     sheet = "Sheet1",  
6     col_types = c("text", "numeric")  
7   )  
8 )  
9  
10 cohort_data_excel <- cohort_data_excel |>  
11   # Check if Serial Number is unique  
12   pointblank::rows_distinct(  
13     columns = "Serial Number",  
14   )
```



A screenshot of Microsoft Excel showing a table with two columns: 'Serial Number' and 'Age'. The data consists of 10 rows, each containing a unique serial number and its corresponding age. The table is located on a sheet titled 'data_to_harmonise_age_issue_fixed.xlsx'.

| | 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

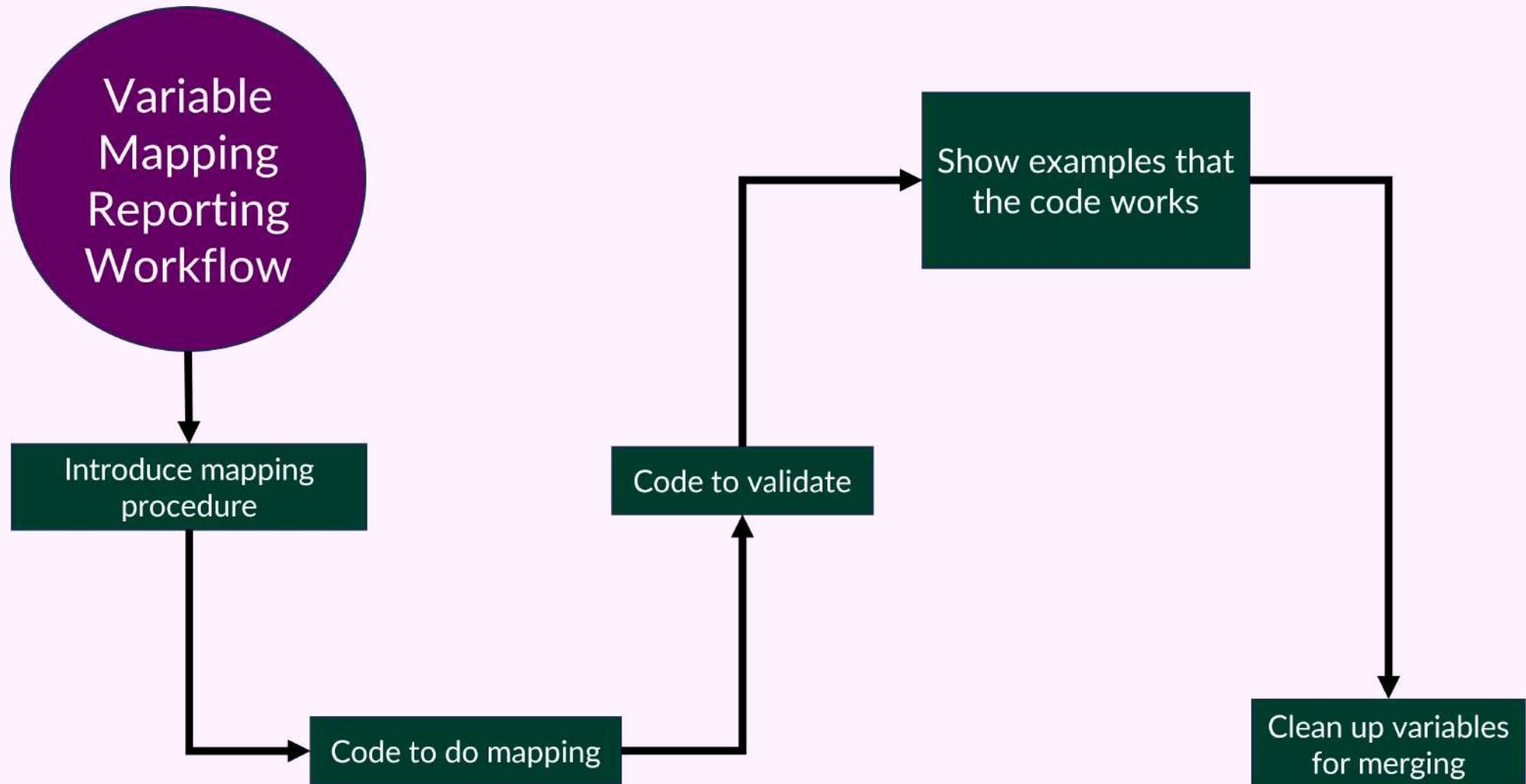
```
1 cohort_data_excel <- readxl::read_excel(  
2   path = here::here("data-raw", "Cohort_Excel",  
3     "data_to_harmonise_age_issue.xlsx"),  
4   sheet = "Sheet1",  
5   col_types = c(  
6     "text", "numeric"  
7   )  
8 ) %T>%  
9 testthat::expect_no_condition()
```

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

No Issues

```
1 cohort_data_excel_2 <- readxl::read_excel(  
2   path = here::here("data-raw", "Cohort_Excel",  
3     "data_to_harmonise_age_issue_fixed.xlsx"),  
4   sheet = "Sheet1",  
5   col_types = c("text", "numeric")  
6 ) %T>%  
7 testthat::expect_no_condition() |>  
8 # Check if Serial Number is unique  
9 pointblank::rows_distinct(  
10   columns = "Serial Number",  
11 )
```

Variable Mapping



Variable Mapping

Let take this data set as an example.

```
1 cohort_csv_data <- vroom::vroom(  
2   file = here::here("data-raw",  
3     "Cohort_csv",  
4     "data_to_harmonise.csv"),  
5   delim = ",",  
6   col_select = 1:8,  
7   show_col_types = FALSE,  
8   col_types = list(  
9     ID = vroom::col_character(),  
10    Age = vroom::col_integer(),  
11    Sex = vroom::col_character(),  
12    Height = vroom::col_double(),  
13    Weight = vroom::col_double(),  
14    `Smoke History` = vroom::col_character(),  
15    `Chest Pain Character` = vroom::col_character(),  
16    Dyspnea = vroom::col_character()  
17  )  
18 ) |>  
19 dplyr::rename(cohort_unique_id = "ID") |>  
20 # Remove rows when the ID value is NA  
21 dplyr::filter(!is.na(.data[["cohort_unique_id"]])) |>
```

| cohort_unique_id | Age | Sex | Height | Weight | S |
|------------------|-----|--------|--------|--------|----|
| B001 | 32 | Female | 170 | 63 | ne |
| B002 | 52 | Female | 167 | 71 | cu |
| B003 | 80 | Male | 184 | 77 | ne |
| B004 | 70 | Male | 160 | 83 | pa |
| B005 | 70 | Female | 155 | 61 | cu |

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

```
1 smoking_data <- cohort_csv_data |>  
2   dplyr::select(c("cohort_unique_id",  
3                   "Smoke History")) |>  
4   dplyr::mutate(  
5     smoke_current = dplyr::case_when(  
6       is.na(.data[["Smoke History"]]) ~ "-1",  
7       .data[["Smoke History"]] == "non-smoker" ~ "0",  
8       .data[["Smoke History"]] == "past smoker" ~ "0",  
9       .data[["Smoke History"]] == "current smoker" ~ "1",  
10      .default = NA_character_  
11    ),  
12    smoke_current = forcats::fct_relevel(  
13      .data[["smoke_current"]],  
14      c("0", "1")),  
15    smoke_past = dplyr::case_when(  
16      is.na(.data[["Smoke History"]]) ~ "-1",  
17      .data[["Smoke History"]] == "non-smoker" ~ "0",  
18      .data[["Smoke History"]] == "past smoker" ~ "1",  
19      .data[["Smoke History"]] == "current smoker" ~ "0",  
20      .default = NA_character_  
21    ),
```

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

```
1 smoking_data <- smoking_data |>
2   pointblank::col_vals_in_set(
3     columns = c("smoke_current", "smoke_past"),
4     set = c("0", "1", "-1")
5   ) |>
6   pointblank::col_vals_expr(
7     expr = pointblank::expr(
8       (.data[["smoke_current"]] == "1" & .data[["smoke_past"]]
9         (.data[["smoke_current"]] == "-1" & .data[["smoke_pas"
10           (.data[["smoke_current"]] == "0" & .data[["smoke_pas
11         )
12     )
```

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

```
1 ---
2 params:
3   show_table: TRUE
4 ---
5
6 ```{r}
7 #| label: output type
8 #| echo: false
9 #| warning: false
10 #| message: false
11
12 out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")
13 ```


```

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

```
1 smoking_data <- smoking_data |>  
2   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.

```
1 age_gender_data |>
2   reactable_with_download_csv_button(
3     defaultPageSize = 5,
4     paginationType = "jump",
5     style = list(fontSize = "1rem"),
6   )
```

| cohort_unique_id | age_years | sex |
|------------------|-----------|-----|
| 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

```
1 body_measurement_data |>
2   reactable_with_download_csv_button(
3     defaultPageSize = 5,
4     paginationType = "jump",
5     style = list(fontSize = "1rem"),
6   )
```

| 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.

```
1 join_specification <- dplyr::join_by("cohort_unique_id")
2
3 demo_behavior_data <- cohort_csv_data |>
4   dplyr::select(c("cohort_unique_id")) |>
5   dplyr::inner_join(age_gender_data,
6     by = join_specification,
7     unmatched = "error",
8     relationship = "one-to-one") |>
9   dplyr::inner_join(body_measurement_data,
10    by = join_specification,
11    unmatched = "error",
12    relationship = "one-to-one") |>
13  dplyr::inner_join(smoking_data,
14    by = join_specification,
15    unmatched = "error",
16    relationship = "one-to-one") |>
17  dplyr::relocate(c("bsa_m2", "bmi"),
18    .after = "sex")
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species, ~island,
3   1, "Adelie", "Torgersen",
4   2, "Gentoo", "Biscoe",
5 )
6
7 weight_extra <- tibble::tribble(
8   ~samp_id, ~body_mass_g,
9   1, 3220,
10  2, 4730,
11  4, 4725
12 )
13
14 three_penguins |>
15   dplyr::inner_join(
16     y = weight_extra,
17     by = dplyr::join_by("samp_id"),
18     unmatched = "error"
19   )
```

```
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.

```
1 join_specification <- dplyr::join_by("cohort_unique_id")
2
3 demo_behavior_data <- cohort_csv_data |>
4   dplyr::select(c("cohort_unique_id")) |>
5   dplyr::inner_join(age_gender_data,
6     by = join_specification,
7     unmatched = "error",
8     relationship = "one-to-one") |>
9   dplyr::inner_join(body_measurement_data,
10    by = join_specification,
11    unmatched = "error",
12    relationship = "one-to-one") |>
13   dplyr::inner_join(smoking_data,
14     by = join_specification,
15     unmatched = "error",
16     relationship = "one-to-one") |>
17   dplyr::relocate(c("bsa_m2", "bmi"),
18     .after = "sex")
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species, ~island,
3   1, "Adelie", "Torgersen",
4   2, "Gentoo", "Biscoe",
5   3, "Chinstrap", "Dream"
6 )
7
8 weight_extra <- tibble::tribble(
9   ~samp_id, ~body_mass_g,
10  1, 3220,
11  3, 4725
12 )
13
14 three_penguins |>
15   dplyr::inner_join(
16     y = weight_extra,
17     by = dplyr::join_by("samp_id"),
18     unmatched = "error"
19   )
```

```
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.

```
1 join_specification <- dplyr::join_by("cohort_unique_id")
2
3 demo_behavior_data <- cohort_csv_data |>
4   dplyr::select(c("cohort_unique_id")) |>
5   dplyr::inner_join(age_gender_data,
6     by = join_specification,
7     unmatched = "error",
8     relationship = "one-to-one") |>
9   dplyr::inner_join(body_measurement_data,
10    by = join_specification,
11    unmatched = "error",
12    relationship = "one-to-one") |>
13   dplyr::inner_join(smoking_data,
14     by = join_specification,
15     unmatched = "error",
16     relationship = "one-to-one") |>
17   dplyr::relocate(c("bsa_m2", "bmi"),
18     .after = "sex")
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species, ~island,
3   1, "Adelie", "Torgersen",
4   2, "Gentoo", "Biscoe",
5   3, "Chinstrap", "Dream"
6 )
7
8 weight_extra <- tibble::tribble(
9   ~samp_id, ~body_mass_g,
10  1, 3220,
11  2, 4730,
12  2, 4725,
13  3, 4000
14 )
15
16 three_penguins |>
17   dplyr::inner_join(
18     y = weight_extra,
19     by = dplyr::join_by("samp_id"),
20     relationship = "one-to-one"
21   )
```

```
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.

```
1 testthat::expect_false(
2   pointblank::has_columns(
3     demo_beave_data,
4     columns = c(
5       dplyr::ends_with(".x"),
6       dplyr::ends_with(".y")
7     )
8   )
9 )
10
11 testthat::expect_equal(
12   ncol(demo_beave_data), 9
13 )
14
15 testthat::expect_true(
16   pointblank::has_columns(
17     demo_beave_data,
18     columns = c(
19       "age_years", "sex",
20       "height_cm", "weight_kg", "bsa_m2", "bmi",
21       "smoke_current", "smoke_past"
22     )
23   )
24 )
```

```
1 three_penguins <- tibble::tribble(
2   ~samp_id, ~species, ~island,
3   1, "Adelie", "Torgersen",
4   2, "Gentoo", "Biscoe",
5   3, "Chinstrap", "Dream"
6 )
7
8 weight_extra <- tibble::tribble(
9   ~samp_id, ~island,
10  1, "Torgersen",
11  2, "Biscoe",
12  3, "Dream"
13 )
14
15 three_penguins <- three_penguins |>
16   dplyr::inner_join(
17     y = weight_extra,
18     by = dplyr::join_by("samp_id"),
19     unmatched = "error",
20     relationship = "one-to-one"
21   )
```

```
[1] TRUE
```

```
1 colnames(three_penguins)
```

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

Comparing Datasets

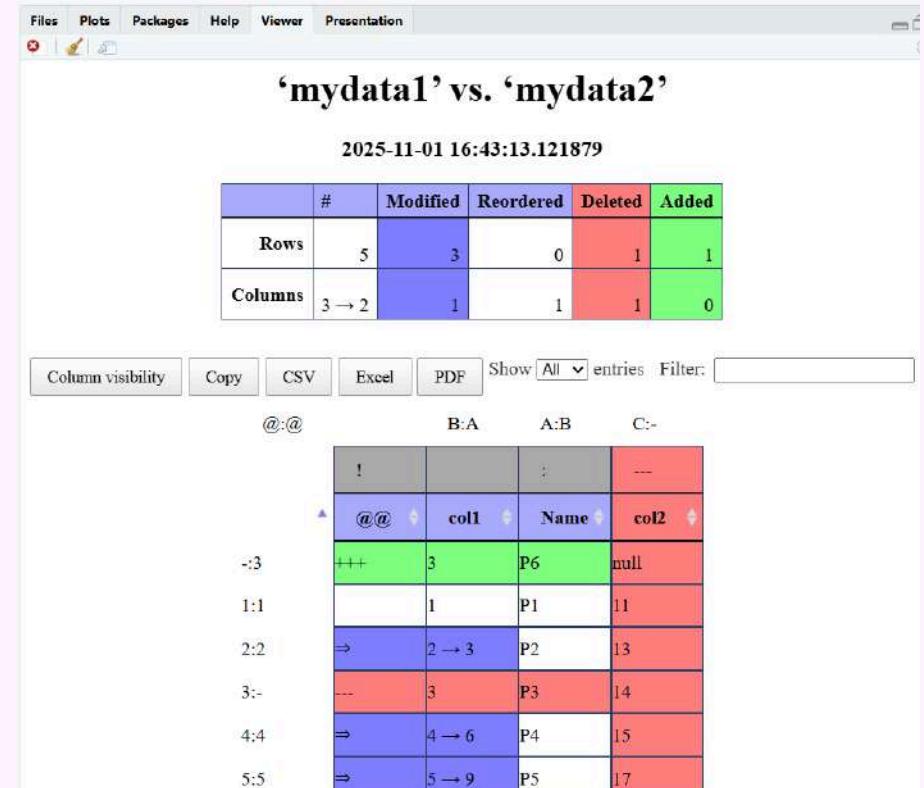
Use  [daff](#) to compare different version of harmonised datasets.

```
1 data1 <- data.frame(  
2   Name=c("P1", "P2", "P3", "P4", "P5"),  
3   col1=c(1,2,3,4,5),  
4   col2=c(11,13,14,15,17)  
5 )  
6  
7 data2 <- data.frame(  
8   col1=c(1,3,3,6,9),  
9   Name=c("P1", "P2", "P6", "P4", "P5")  
10 )  
11  
12 compare_results <- daff::diff_data(data1, data2)  
13 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 |

```
1 daff::render_diff(compare_results)
```



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

Column visibility Copy CSV Excel PDF Show All entries Filter:

| @:@ | B:A | A:B | C:- | |
|-----|------|-------|------|------|
| ! | | ; | --- | |
| @@ | col1 | Name | col2 | |
| -3 | +++ | 3 | P6 | null |
| 1:1 | | 1 | P1 | 11 |
| 2:2 | ⇒ | 2 → 3 | P2 | 13 |
| 3:- | --- | 3 | P3 | 14 |
| 4:4 | ⇒ | 4 → 6 | P4 | 15 |
| 5:5 | ⇒ | 5 → 9 | P5 | 17 |

Showing 1 to 6 of 6 entries

Previous 1 Next

Comparing Datasets

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

```
1 data1 <- data.frame(  
2   Name=c("P1", "P2", "P3", "P4", "P5"),  
3   col1=c(1,2,3,4,5),  
4   col2=c(11,13,14,15,17)  
5 )  
6  
7 data2 <- data.frame(  
8   col1=c(1,3,3,6,9),  
9   Name=c("P1", "P2", "P6", "P4", "P5")  
10 )  
11  
12 compare_different_summary <- daff::diff_data(  
13   data1, data2) |>  
14   summary()  
15  
16 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
```

```
1 data1 <- data.frame(  
2   Name=c("P1", "P2", "P3", "P4", "P5"),  
3   col1=c(1,2,3,4,5),  
4   col2=c(11,13,14,15,17)  
5 )  
6  
7 data2 <- data.frame(  
8   Name=c("P1", "P2", "P3", "P4", "P5"),  
9   col1=c(1,2,3,4,5),  
10  col2=c(11,13,14,15,17)  
11 )  
12  
13 compare_same_summary <- daff::diff_data(data1, data2) |>  
14   summary()  
15  
16 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.

```
1 tibble::tibble(
2   row_deletes = compare_different_summary$row_deletes,
3   row_inserts = compare_different_summary$row_inserts,
4   row_updates = compare_different_summary$row_updates,
5   row_reorders = compare_different_summary$row_reorders,
6   col_deletes = compare_different_summary$col_deletes,
7   col_inserts = compare_different_summary$col_inserts,
8   col_updates = compare_different_summary$col_updates,
9   col_reorders = compare_different_summary$col_reorders,
10 ) |>
11   pointblank::expect_col_vals_in_set(
12     columns = c(
13       "row_deletes", "row_inserts", "row_updates", "row_reo
14       "col_deletes", "col_inserts", "col_updates", "col_reo
15     ),
16     set = c(0)
17   )
```

```
1 tibble::tibble(
2   row_deletes = compare_same_summary$row_deletes,
3   row_inserts = compare_same_summary$row_inserts,
4   row_updates = compare_same_summary$row_updates,
5   row_reorders = compare_same_summary$row_reorders,
6   col_deletes = compare_same_summary$col_deletes,
7   col_inserts = compare_same_summary$col_inserts,
8   col_updates = compare_same_summary$col_updates,
9   col_reorders = compare_same_summary$col_reorders,
10 ) |>
11   pointblank::expect_col_vals_in_set(
12     columns = c(
13       "row_deletes", "row_inserts", "row_updates", "row_reo
14       "col_deletes", "col_inserts", "col_updates", "col_reo
15     ),
16     set = c(0)
17   )
```

```
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)
```

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](#)

Quarto Books/Websites

Harmonisation Template for Cohort B



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

```
YAML _quarto.yml x index.qmd x
< < > < > < > < >
1 project:
2   type: book
3   output-dir: reports/Cohort_B
4
5 book:
6   downloads: [pdf, docx]
7   title: "Harmonisation Template for Cohort B"
8   author: "My Name"
9 navbar:
10  search: true
11 sidebar:
12  collapse-level: 1
13
14 chapters:
15 - index.qmd
16 - part: Cohort B Cleaning
17 chapters:
18 - codes/Cohort_B/00_R_Package_And_Environment.qmd
19 - codes/Cohort_B/01_Read_Cohort_B_Data.qmd
20 - codes/Cohort_B/02_Extract_Demographic.qmd
21 - codes/Cohort_B/03_Export_To_Excel.qmd
22
```

The screenshot shows the Quarto IDE interface with the _quarto.yml file open. The file contains YAML configuration for a document. The code editor shows syntax highlighting for the YAML syntax. The interface includes a toolbar with icons for file operations, a status bar at the bottom, and a navigation bar at the top.

```
1 ---  
2 date: "2025-03-10"  
3 format:  
4   html:  
5     code-fold: true  
6     freeze: false  
7 params:  
8   show_table: TRUE  
9 ---  
10  
11 ``{r}  
12 #| label: output type  
13 #| echo: false  
14 #| warning: false  
15 #| message: false  
16  
17 out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")  
18 ``  
19  
20 # Preface {.unnumbered .unlisted}  
21  
22 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`

```
1  ---
2  project:
3    type: book
4    output-dir: reports/Cohort_B
5
6  book:
7    downloads: [pdf, docx]
8    title: "Harmonisation Template for Cohort B"
9    author: "My Name"
10   navbar:
11     search: true
12   sidebar:
13     collapse-level: 1
14
15  chapters:
16    - index.qmd
17    - part: Cohort B Cleaning
18      chapters:
19        - codes/Cohort_B/00_R_Package_And_Environment.qmd
20        - codes/Cohort_B/01_Read_Cohort_B_Data.qmd
21        - 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
```

```
1 ---  
2 date: "2025-03-10"  
3 format:  
4   html:  
5     code-fold: true  
6     freeze: false  
7 params:  
8   show_table: TRUE  
9 ---  
10  
11 ````{r}  
12 #| label: output type  
13 #| echo: false  
14 #| warning: false  
15 #| message: false  
16  
17 out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")  
18 ````  
19  
20 # Preface {.unnumbered .unlisted}  
21
```

Quarto (Reference)

<https://quarto.org/>

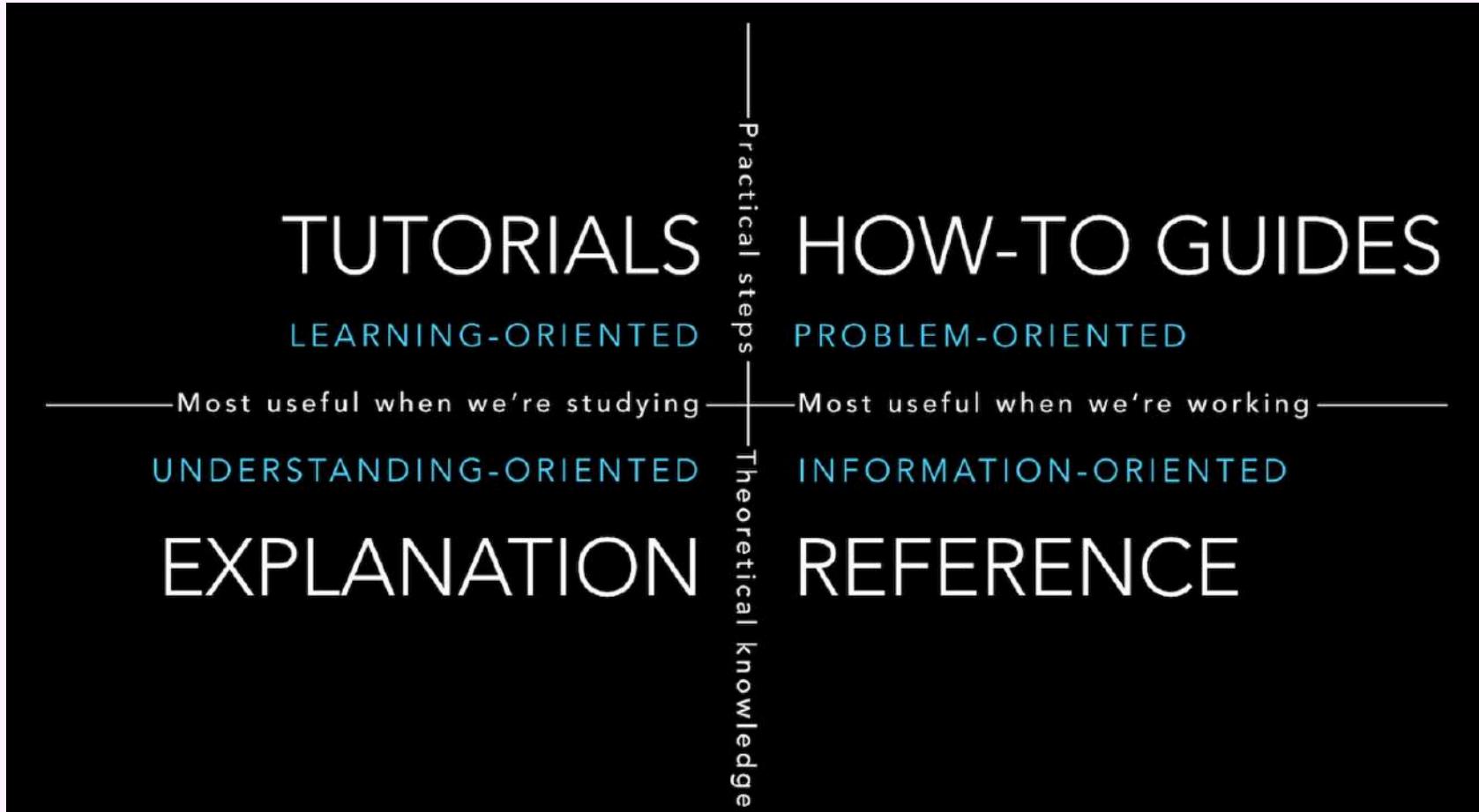
The screenshot shows the Quarto website homepage. At the top left is the Quarto logo (a blue circle with a white cross). To its right are links for Overview, Get Started, Guide, Extensions, Reference, Gallery, Blog, and Help. On the far right of the header are icons for a butterfly, a person, a feed, and a search bar. To the right of the search bar is a box labeled "supported by" with the Posit logo. The main content area features a large blue header "Welcome to Quarto®" followed by a subheader "An open-source scientific and technical publishing system". Below this is a bulleted list of features:

- Author using [Jupyter](#) notebooks or with plain text markdown in your favorite editor.
- Create dynamic content with [Python](#), [R](#), [Julia](#), and [Observable](#).
- Publish reproducible, production quality articles, presentations, dashboards, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
- Share knowledge and insights organization-wide by publishing to [Posit Connect](#), [Confluence](#), or other publishing systems.
- Write using [Pandoc](#) markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.

At the bottom of the page, there is a footer bar with links for Analytics, Share, Reproduce, and Contact, along with a "Feedback" button.

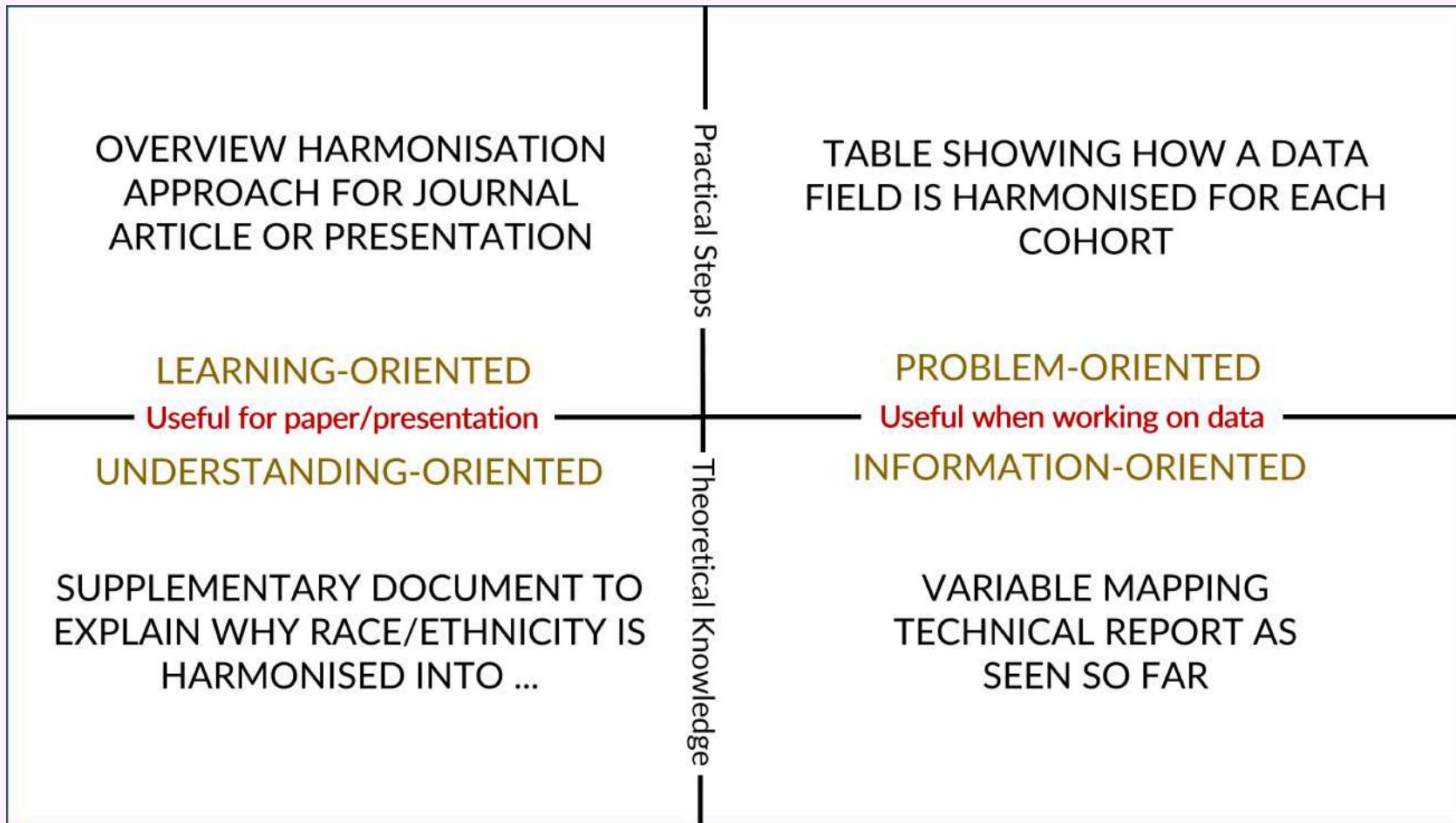
Harmonisation Report Types

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



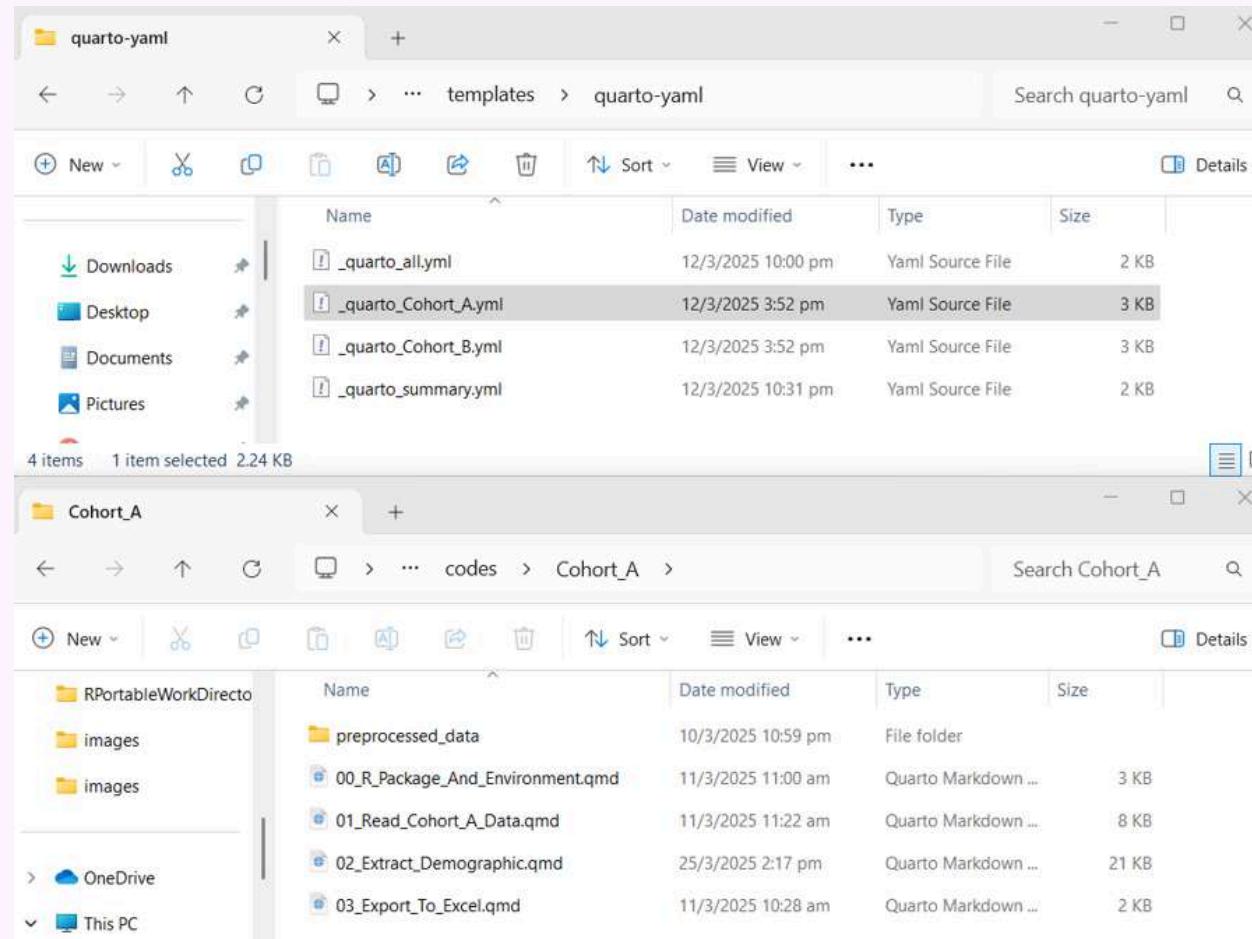
Harmonisation Report Types

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



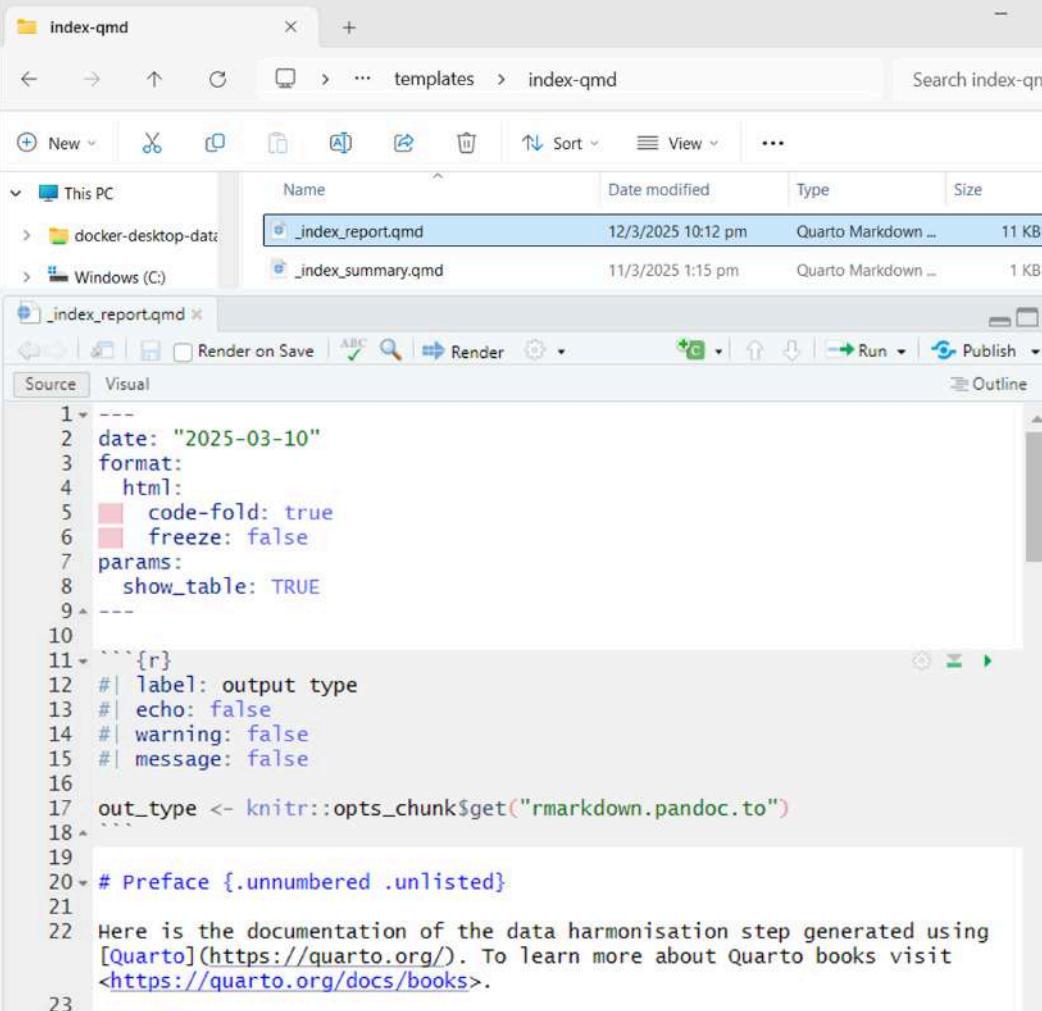
Automated Technical Report (Reference)

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.



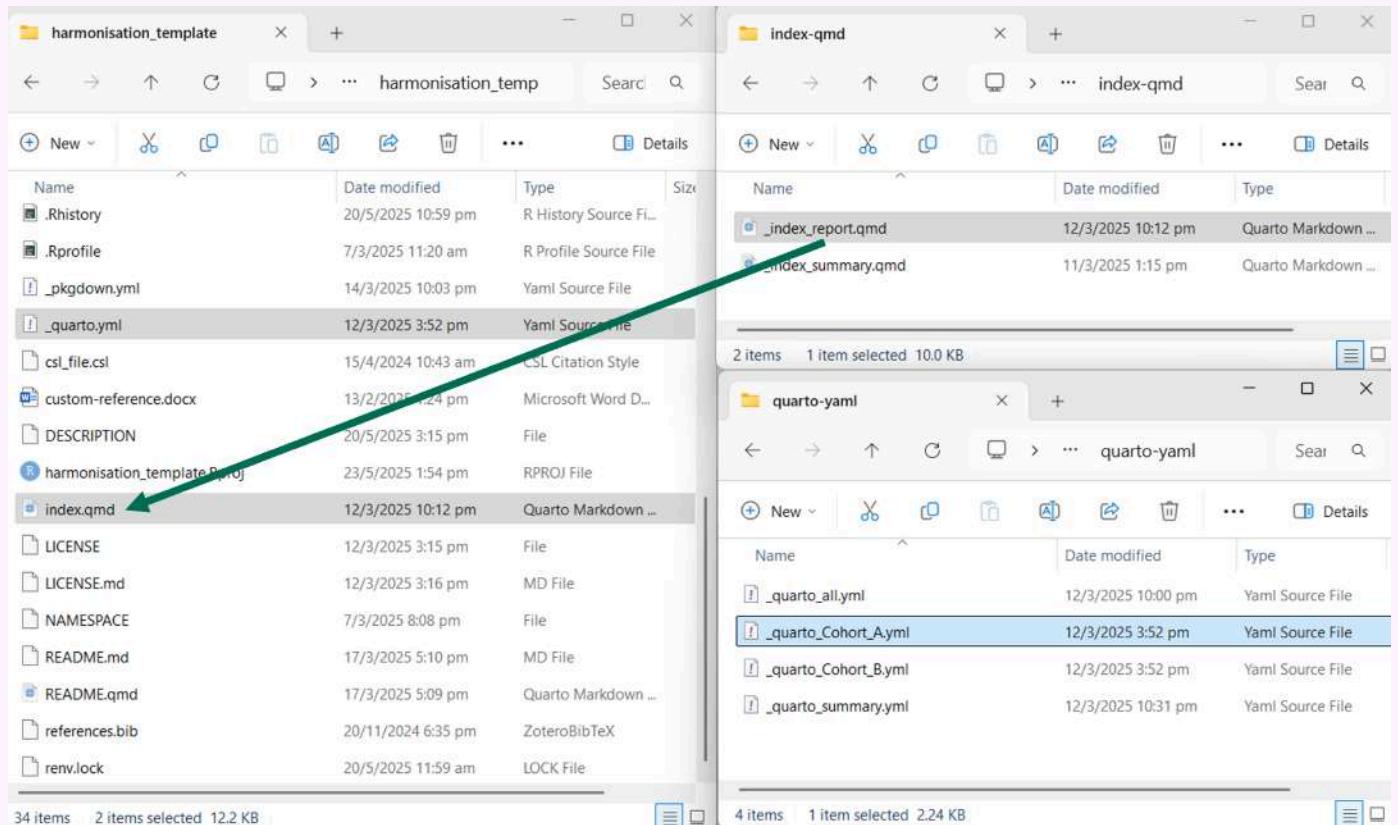
The screenshot shows a Windows File Explorer window titled "index-qmd". The address bar shows the path "templates > index-qmd". The main pane lists two files: "_index_report.qmd" and "_index_summary.qmd". The file "_index_report.qmd" is selected and its content is displayed in the bottom pane. The content is a Quarto Markdown document (qmd) with the following code:

```
1 ---  
2 date: "2025-03-10"  
3 format:  
4   html:  
5     code-fold: true  
6     freeze: false  
7 params:  
8   show_table: TRUE  
9 ---  
10  
11 `r`  
12 #| label: output type  
13 #| echo: false  
14 #| warning: false  
15 #| message: false  
16  
17 out_type <- knitr::opts_chunk$get("rmarkdown.pandoc.to")  
18 ...  
19  
20 # Preface {.unnumbered .unlisted}  
21  
22 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>.  
23
```

Automated Technical Report (Reference)

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

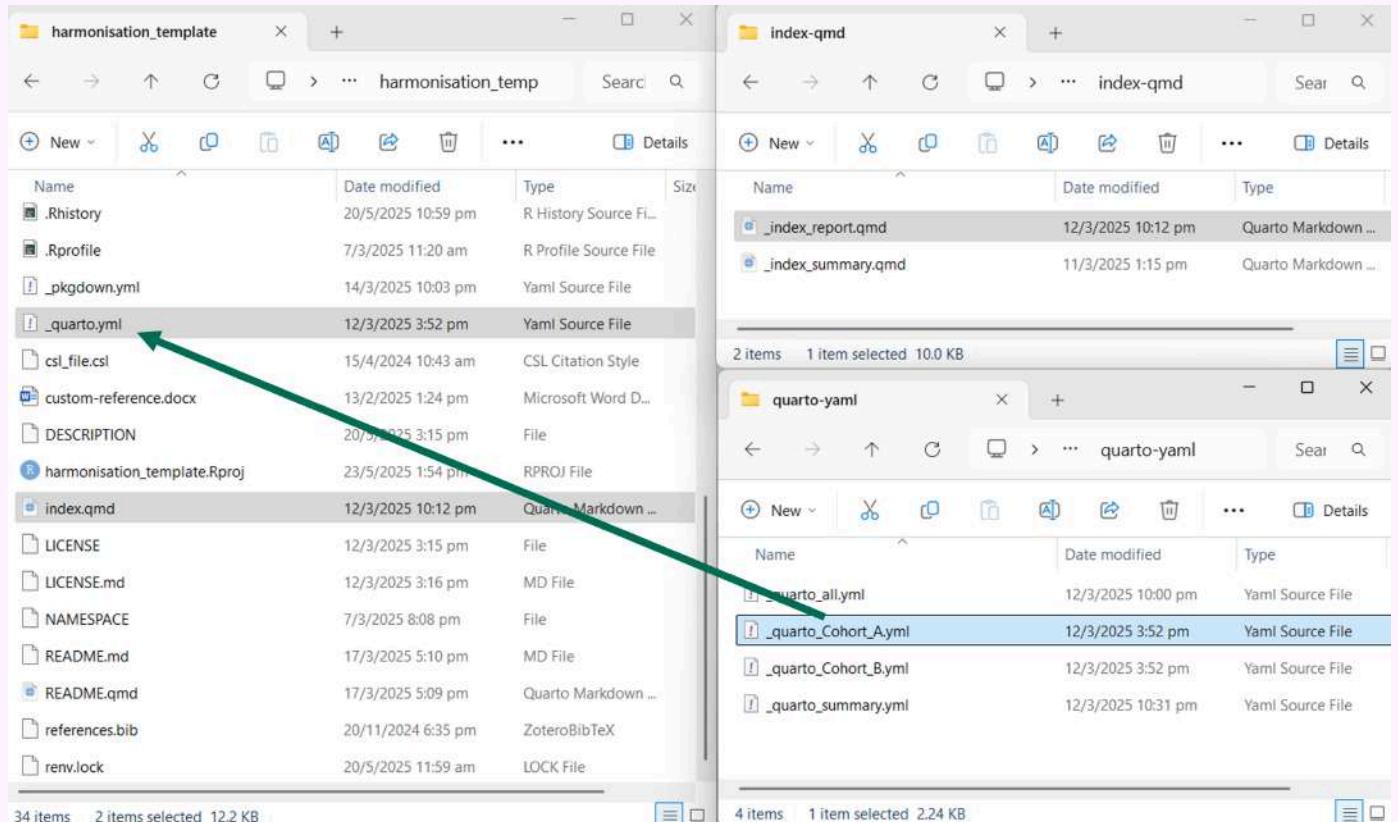
```
1 # Copy the right index.qmd
2 # file
3
4 index_qmd_file <- paste0(
5   "_index_",
6   "report",
7   ".qmd"
8 )
9
10 fs::file_copy(
11   path = here::here(
12     "templates",
13     "index-qmd",
14     index_qmd_file),
15   new_path = here::here(
16     "index.qmd"
17   ),
18   overwrite = TRUE
19 )
```



Automated Technical Report (Reference)

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

```
1 copy_and_render <- function(  
2   cohort  
3 ) {  
4   # Copy quarto.yml file  
5   # for each cohort  
6  
7   quarto_yml_file <- paste0(  
8     "_quarto_",
9     cohort,
10    ".yml"
11  )  
12  
13  fs::file_copy(  
14    path = here::here(  
15      "templates",
16      "quarto-yaml",
17      quarto_yml_file),
18    new_path = here::here("_quar",
19    overwise = TRUE
20  )
```

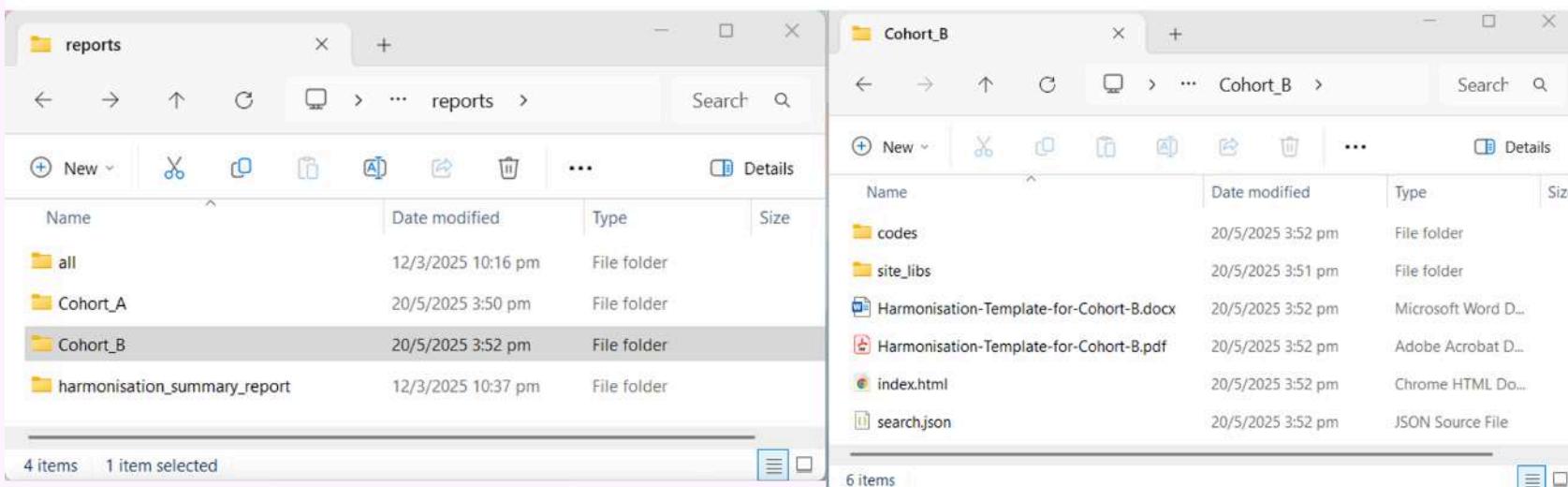


Automated Technical Report (Reference)

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   [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 <i>smoke_current_good</i> with
0 as no.
1 as yes.
-1 as unknown.
Column <i>smoke_past_good</i> with
0 as no.
1 as yes.
-1 as unknown. | <i>smoke_current</i> will take the values of <i>smoke_current_good</i> .
<i>smoke_past</i> will take the values of <i>smoke_past_good</i> . |
| Cohort B | Column <i>Smoke History</i> with
non-smoker as non-smoker. | Map the values of <i>Smoke History</i> to <i>smoke_current</i> as follows: |

past smoker as a past smoker.

current smoker as a current smoker.

NA as unknown.

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*.

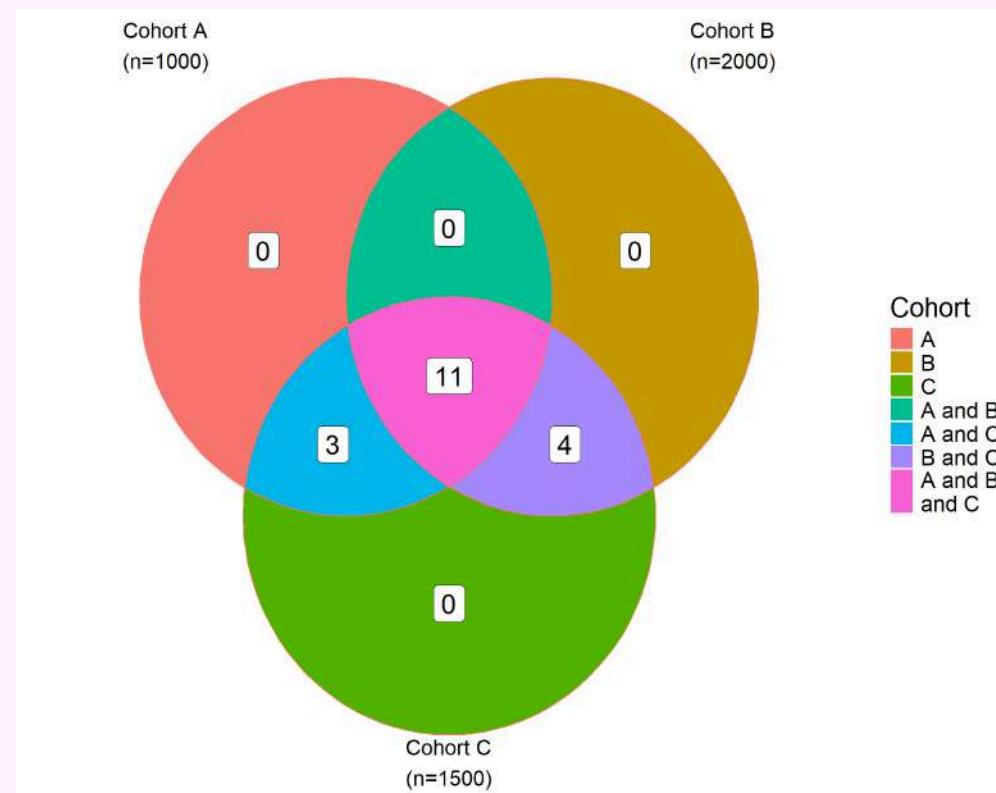
| Description | <i>smoke_current</i> | <i>smoke_past</i> |
|----------------|----------------------|-------------------|
| Non-smoker | 0 | 0 |
| Past smoker | 0 | 1 |
| Current smoker | 1 | 0 |
| Unknown | -1 | -1 |

Overview Diagrams

How many variables can each cohort provide ?

How many variables can be harmonised ?

```
1 demographic_list <- list(  
2   A = c("Age", "Sex",  
3     "Hypertension", "Dyslipidemia", "Family Hx CAD", "D  
4     "Smoke Current", "Smoke Past",  
5     "Have Chest Pain", "Chest Pain Character",  
6     "Dyspnea",  
7     "BMI", "Height", "Weight"),  
8   B = c("Age", "Sex",  
9     "Hypertension", "Dyslipidemia", "Family Hx CAD", "D  
10    "Smoke Current", "Smoke Past",  
11    "Have Chest Pain", "Chest Pain Character",  
12    "Dyspnea",  
13    "HDL", "Total Cholesterol",  
14    "Triglyceride", "LDL"),  
15   C = c("Age", "Sex",  
16     "Hypertension", "Dyslipidemia", "Family Hx CAD", "D  
17     "Smoke Current", "Smoke Past",  
18     "Have Chest Pain", "Chest Pain Character",  
19     "Dyspnea",  
20     "BMI", "Height", "Weight",  
21     "HDL", "Total Cholesterol",
```

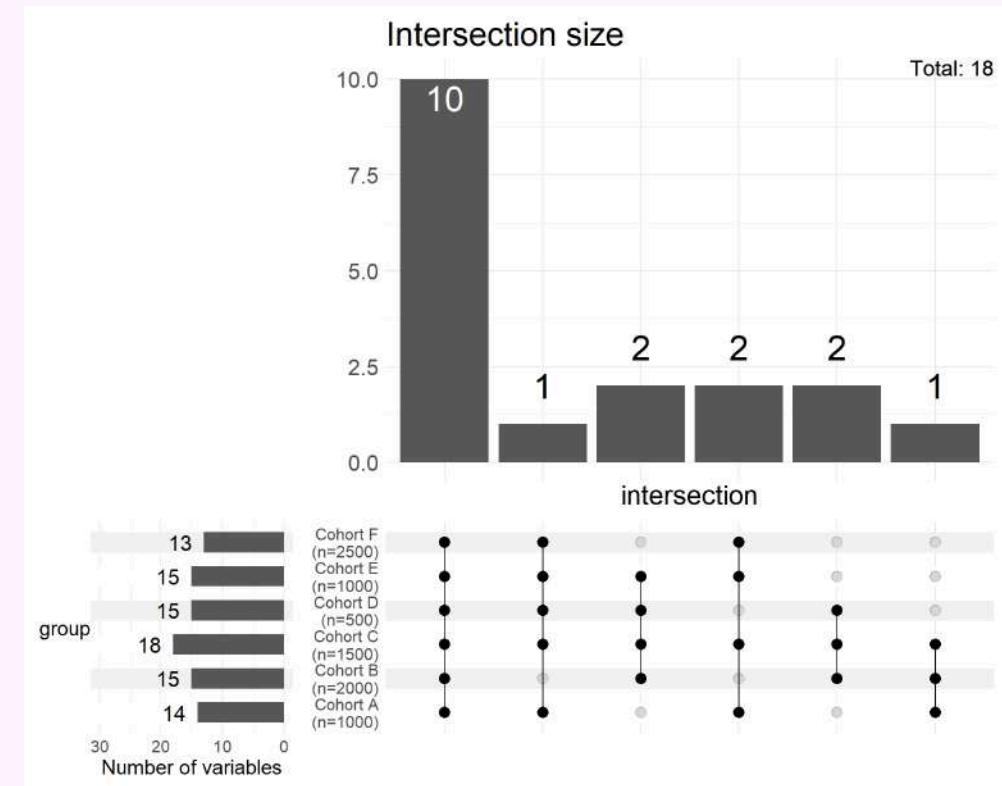


Venn diagram does not work for many (> 10) cohorts.

Overview Diagrams

Upset plots are too complicated for clinicians.

```
1 demographic_venn <- tibble::tibble(  
2   column_name = c("Age", "Sex",  
3     "Hypertension", "Dyslipidemia", "Family H  
4     "Smoke Current", "Smoke Past",  
5     "Have Chest Pain", "Chest Pain Character"  
6     "Dyspnea",  
7     "BMI", "Height", "Weight",  
8     "HDL", "Total Cholesterol",  
9     "Triglyceride", "LDL"),  
10   `Cohort A` = c(1, 1,  
11     1, 1, 1, 1,  
12     1, 1,  
13     1, 1,  
14     1,  
15     1, 1, 1,  
16     0, 0,  
17     0, 0),  
18   `Cohort B` = c(1, 1,  
19     1, 1, 1, 1,  
20     1, 1,  
21     1, 1,
```



Cannot answer follow-up questions:

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

Overview Diagrams

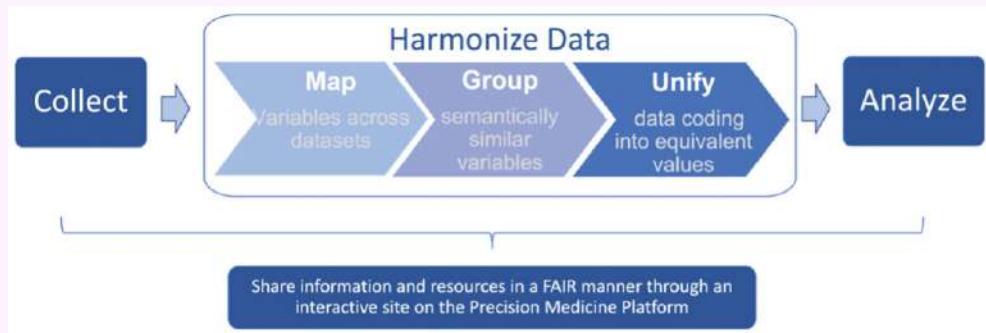
Create a “heatmap” using Microsoft PowerPoint.

| | | | | | | | | | |
|--------------|----------|------|----|----------------------|---------|------|--------|------|--------------|
| | | | | 10 | | | | | |
| | | | | Age | | | | | |
| | | | | Sex | | | | | |
| | | | | Hypertension | | | | | |
| | | | | Dyslipidemia | | | | | |
| | | | | Family Hx CAD | | | | | |
| | | | | Diabetes | | | | | |
| | | | | Smoke Current | | | | | |
| | | | | Smoke Past | | | | | |
| | | | | Have Chest Pain | 1 | 1 | 2 | 2 | 2 |
| | | | | Chest Pain Character | Dyspnea | BMI | Height | HDL | Triglyceride |
| 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 | Green |
| Sex | Yellow |
| Comorbidity | Light Blue |
| Smoking history | Grey |
| Symptoms | Cyan |
| Obesity | Pink |
| Blood lipid | Orange |

| Table Legend | |
|--------------|-----------------|
| ✓ | Available |
| ✗ | Not available |
| Grey | Pending arrival |

Summary



Project Organisation

here: find your PATH!

Workspace

Restore .RData into workspace at startup:
Save workspace to .RData on exit: Never

Welcome to Posit Public Package Manager

The best way to discover and install R and Python packages

R Administration

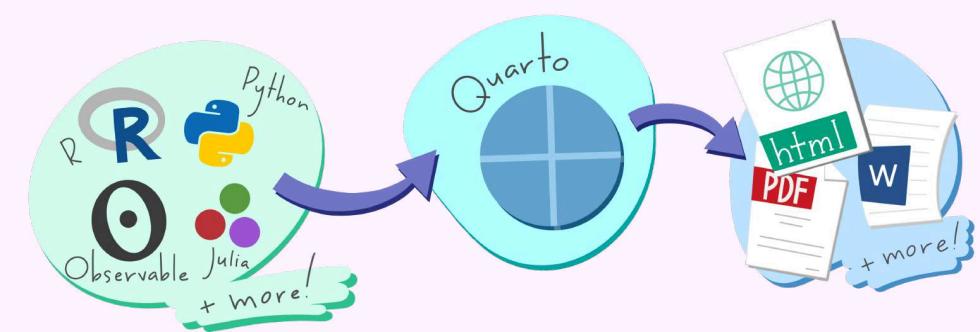
renv

Namespacing
dplyr::select()

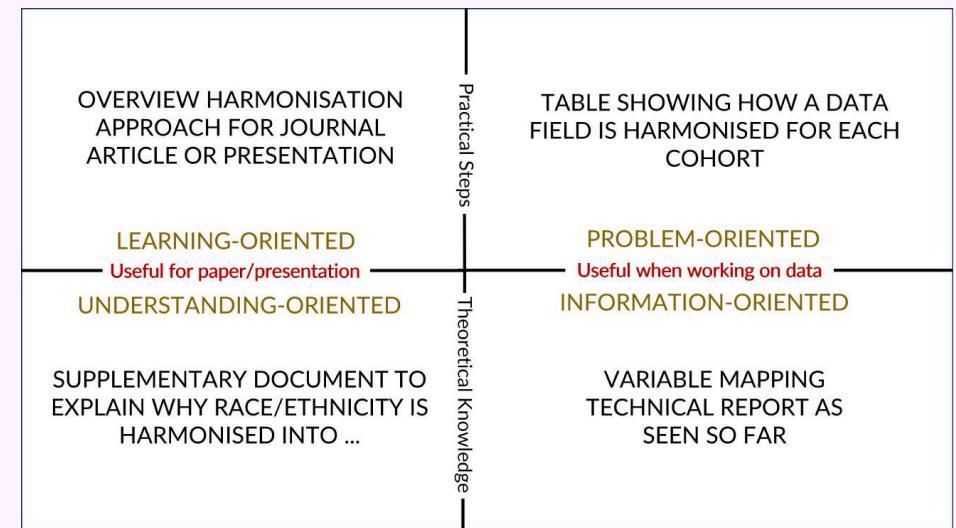
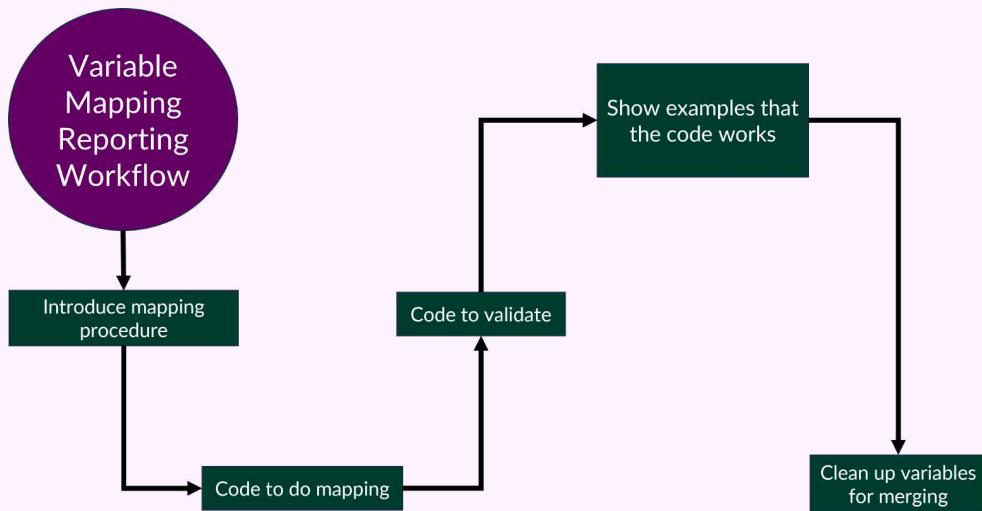
.Renvironment
.Rhistory
.Rprofile

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

| Country | Cohort | N | Variables | 2 | | 2 | | 2 | |
|--------------|----------|------|-----------|----------------------|---------|------|--------|------|--------------|
| | | | | Chest Pain Character | Dyspnea | BMI | Height | HDL | Triglyceride |
| Country A | Cohort A | 1000 | 15 | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| Country A | Cohort B | 2000 | 16 | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| Country B | Cohort C | 1500 | 18 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Country C | Cohort D | 500 | 16 | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ |
| Country D | Cohort E | 1000 | 16 | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| Country D | Cohort F | 2500 | 14 | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| Total | | | | 8500 | 4500 | 6500 | 6000 | 5000 | 4000 |

Variable Colour Legend

| | |
|-----------------|------------|
| Age | Green |
| Sex | Yellow |
| Comorbidity | Light Blue |
| Smoking history | Grey |
| Symptoms | Cyan |
| Obesity | Purple |
| Blood lipid | Orange |

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

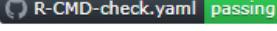
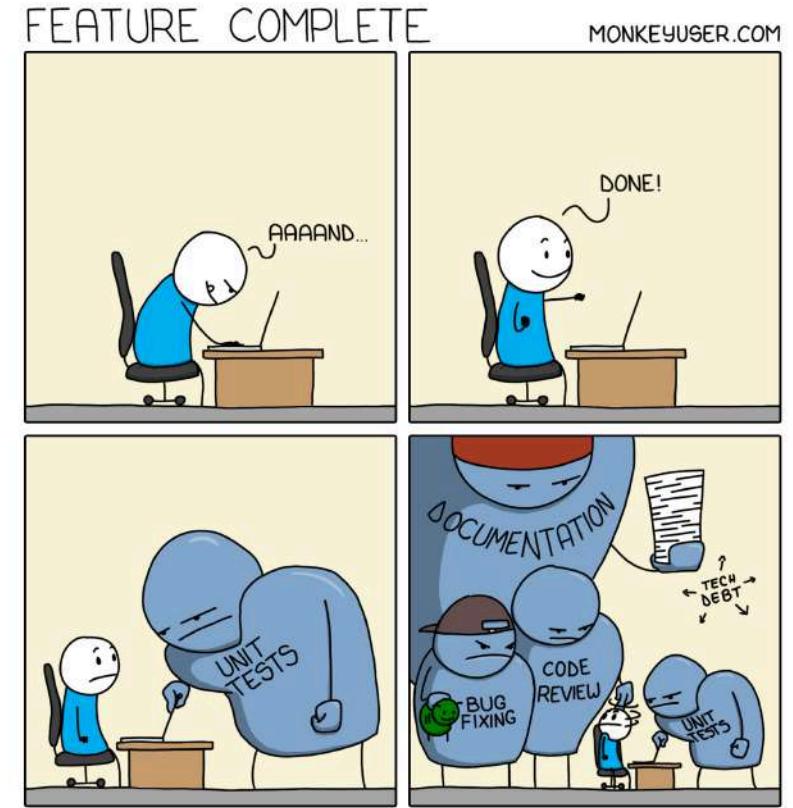
 R-CMD-check.yaml passing

Table of Content

- [Motivation](#)
- [Acknowledgement](#)
- [File Structure](#)
- [Software Installation](#)
- [R Package Installation](#)
- [Using `renv`](#)
- [R Functions Management](#)
- [R Packages Used](#)
- [R Platform Information](#)
- [Data Harmonisation Report For Each Cohort](#)
- [Combined Data Harmonisation Report For All Cohort](#)
- [Data Harmonisation Summary](#)
- [General Recommendations](#)



[Feature Complete from MonkeyUser.com](#)