Harmonisation Template for Cohort A

My Name

2025-07-03

Table of contents

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.

Acknowledgement

Layout of this page is inspired from R package `rcompendium`.

File Structure

Here is the file structure of this project.

```
harmonisation_template/                  # Root of the compendium

   harmonisation_template.Rproj          # RStudio project file
|
   .quarto/                              # Intermediate files/folders
generated
|                                        # Quarto renders to the
documents.
|
   archive/                              # Folder to keep previous books
and harmonised data
      reports/                           # Folder containing previous
documentation
|   |                                    # of data harmonisation
|     harmonised/                        # Folder containing previous
GPS-CAD harmonised data
|
   reports/                              # Documentation of data
harmonisation

   data-raw/                             # Cohort raw data (.csv, .gpkg,
etc.)
      {cohort name}/                     # Folder containing cohort raw
data,
|   |                                    # and data dictionary
```

```
|      data-dictionary/                  # Data dictionary for harmonised
data
|      data-input/                       # Data input file from
collaborators
|
   docs/                                 # R functions documentation
generating using
|                                          # pkgdown:::build_site_external()
|
   inst/                                 # Arbitrary additional files to
include in the
|   |                                      # package.
|   |
|      WORDLIST                          # File generating by
spelling::update_wordlist()
|
   man/                                  # R functions helps (automatically
updated)
       {fun-demo}.Rd                     # Documentation of the demo R
function
       harmonisation-template.Rd         # High-level documentation
|
   quarto-yaml-template/                 # Folder containing template files
for quarto book generation
       _quarto_{cohort name}.yml         # Quarto book generation for each
cohort
       _quarto_all.yml                   # Quarto book generation for all
cohorts
|
   R/                                    # R functions location
       {fun-demo}.R                      # Example of an R function
       harmonisation-template-package.R  # Dummy R file for high-level
documentation

   renv/                                 # Folder that contains all
packages
|                                          # installed in the renv
environment.
|
   codes/                                # R/Quarto scripts to run data
harmonisation
|      quarto_script.R                   # R script to render each {cohort
name}_Cleaning/ folder.
|   |                                      # folder into html, pdf and word
document.
```

```
      {cohort name}_Cleaning/           # Quarto scripts to run data
harmonisation
|   |                                    # and output them for each
cohort.
|     Combine/                          # Quarto scripts to filter
harmonised data
|                                        # based on inclusion/exclusion
criteria,
|   |                                    # combined the filtered data for
preliminary analysis.
|
  tests/                                # Test units file created by R
package testhat

  .lintr                                # Configuration for linting
|                                        # R projects and packages using
linter
|
  .Rbuildignore                         # List of files/folders to be
ignored while
                                         # checking/installing the package
|
  .renvignore                           # List of files/folders to be
ignored when
                                         # renv is doing its snapshot
|
  _pkgdown.yml                          # Configuration for R package
documentation
|                                        # using
pkgdown:::build_site_external()
|
  _quarto.yml                           # Configuration for Quarto book
generation
|                                        # Also the project configuration
file
|
  custom-reference.docx                 # Microsoft word template for data
harmonisation
|                                        # documentation to Word
|
  DESCRIPTION                           # Project metadata[*]
|
  index.qmd                             # Home page of Quarto book content
|
  LICENSE                               # Content of the MIT license
generated via
```

```
|                                       # usethis::use_mit_license()
   LICENSE.md                          # Content of the MIT license
generated via
|                                       # usethis::use_mit_license()
|
   NAMESPACE                           # Automatically generated

   README.md                           # GitHub README (automatically
generated)
   README.Rmd                          # GitHub README (ignore for now)

|
   references.bib                      # Bibtex file for Quarto book
|
   references.qmd                      # Reference document for Quarto
book
|
   renv.lock                           # Metadata of R packages installed
generated
|                                       # using renv::snapshot
|
   csl_file.csl                        # Citation Style Language (CSL)
file to ensure
|                                       # citations follows the Lancet
journal
```

[*] These files are automatically created but user needs to manually add some information.

Installation

Installing R

Go to https://cran.rstudio.com/. Choose a version of R that matches the computer's operating system.

Installing RStudio

Go to https://posit.co/download/rstudio-desktop/. Scroll down and choose a version of RStudio that matches the computer's operating system.

Installing Rtools

Go to https://cran.r-project.org/bin/windows/Rtools/. Choose a version of Rtools that matches the R version that was installed.

Quarto

Quarto converts R scripts into a technical report or notebook in html, pdf, Microsoft Word, etc. It is installed together with RStudio. User can also go to https://quarto.org/docs/get-started/ to install it separately. For Quarto to be able to create pdf files, a pdf engine must be installed as well. For ease, it is suggested to install TinyTex using the terminal command `quarto install tinytex`.

R Package Installation

Use Posit Public Package Manager PPM to set up your repository environment to install R packages from CRAN. This is because PPM allows installation of frozen R package versions based on a snapshot date.

One way to do that is to set in the `.Rprofile` file with the code `options(repos = c(P3M = "{link to repository url form Posit Public Package Manager}"))`

R packages can be installed using the package `pak` as an alternative to `install.packages()` and `remotes::install_github()](https://remotes.r-lib.org/reference/install_github.html). Benefits of using [pak‘` can be found here

You can also view your respository environment using the command `pak::repo_get()`

R package can be loaded using the command `library({package_name})`. You can use the R package `annotater` to add additional information on what the loaded package does.

Using `renv`

You can increase reproducibility by using the package `renv`. Install `renv` from CRAN with `pak::pak("renv")`. If this is your first time using `renv`, start with the `Introduction to renv vignette`. Use `renv::init(bare = TRUE)` to start with an empty `renv` environment.

`renv` will freeze the exact package versions you depend on (in `renv.lock`). This ensures that each collaborator (or you in the future) will use the exact same versions of these packages. Moreover `renv` provides to each project its own private package library making each project isolated from others.

Install required dependencies locally with `install.packages()` or `renv::install()` from CRAN, Bioconductor, Github, explicit file path, etc.

Sometimes the right downloader (libcurl or others) needs to set for installation of R packages inside the `renv` environment to be successful. Setting the R environmental variable RENV_DOWNLOAD_FILE_METHOD = "libcurl" may help.

Save the local environment with `renv::snapshot()` to create the `renv.lock` file.

R Functions Management

R functions heavily used in this project can be found in the `R` folder. Documentation (`man` folder), test units (`test` folder) corresponding to these functions are structured the same as creating an R package. Relevant R packages required for R package development (and available on Posit Public Package Manager PPM) are

```
library("usethis")
library("devtools")
library("roxygen2")
library("testthat")
library("covr")
library("spelling")
library("lintr")
library("sinew")
library("pkgdown")
```

Here is an example of the command to use `pak::pak("{package name}")` to install packages from PPM.

There is no need to source the functions in the R folder. Use `devtools::load_all()` instead. `devtools::load_all()` will load required dependencies listed in `DESCRIPTION` and R functions stored in `R/`. Prior installation of these dependencies is required for the load to be successful.

After loading, R functions can be documented (using `devtools::document()`), tested (using `devtools::test()` and then `devtools::check()`) and even installed as an R package (using `devtools::install`).

More information of this workflow can be found in Chapter 1: The Whole Game of the R Packages (2e) book.

R Packages

R packages installed from Posit Public Package Manager PPM using command `pak::pak("{package name}")` are

```
library("renv")
library("sessioninfo")
library("knitr")
library("rmarkdown")
library("quarto")
library("rlang")
library("cli")

library("fs")
library("here")
library("fst")
library("readxl")
library("vroom")

library("dplyr")
library("tidyr")
library("magrittr")
library("stringr")
library("forcats")
library("purrr")
library("lubridate")
library("tibble")
library("glue")

library("collateral")
library("pointblank")
library("testthat")

library("htmltools")
library("htmlwidgets")
library("fontawesome")
library("reactable")
library("flextable")

library("openxlsx")

library("harmonisation")
```

Here are all the R packages used in this analysis.

```
harmonisation::get_r_package_info() |>
  knitr::kable()
```

package	version	date	source
cli	3.6.4	2025-02-13	RSPM
collateral	0.5.2	2021-10-25	RSPM
covr	3.6.4	2023-11-09	RSPM
devtools	2.4.5	2022-10-11	RSPM
dplyr	1.1.4	2023-11-17	RSPM
flextable	0.9.7	2024-10-27	RSPM
fontawesome	0.5.3	2024-11-16	RSPM
forcats	1.0.0	2023-01-29	RSPM
fs	1.6.5	2024-10-30	RSPM
fst	0.9.8	2022-02-08	RSPM
glue	1.8.0	2024-09-30	RSPM
harmonisation	0.0.0.9999	2025-03-09	local
here	1.0.1	2020-12-13	RSPM
htmltools	0.5.8.1	2024-04-04	RSPM
htmlwidgets	1.6.4	2023-12-06	RSPM
knitr	1.49	2024-11-08	RSPM
lintr	3.2.0	2025-02-12	RSPM
lubridate	1.9.4	2024-12-08	RSPM
magrittr	2.0.3	2022-03-30	RSPM
openxlsx	4.2.8	2025-01-25	RSPM
pkgdown	2.1.1	2024-09-17	RSPM
pointblank	0.12.2	2024-10-23	RSPM
purrr	1.0.4	2025-02-05	RSPM
quarto	1.4.4	2024-07-20	RSPM
reactable	0.4.4	2023-03-12	RSPM
readxl	1.4.4	2025-02-27	RSPM
renv	1.1.0	2025-01-29	RSPM (R 4.4.0)
rlang	1.1.5	2025-01-17	RSPM
rmarkdown	2.29	2024-11-04	RSPM
roxygen2	7.3.2	2024-06-28	RSPM
sessioninfo	1.2.2	2021-12-06	CRAN (R 4.4.2)
sinew	0.4.0	2022-03-31	RSPM
spelling	2.3.1	2024-10-04	RSPM
stringr	1.5.1	2023-11-14	RSPM
testthat	3.2.3	2025-01-13	RSPM
tibble	3.2.1	2023-03-20	RSPM
tidyr	1.3.1	2024-01-24	RSPM
usethis	3.1.0	2024-11-26	RSPM
vroom	1.6.5	2023-12-05	RSPM

R Platform Information

Here are the R platform environment used in this analysis.

```
harmonisation::get_r_platform_info() |>
  knitr::kable()
```

setting	value
version	R version 4.4.2 (2024-10-31 ucrt)
os	Windows 11 x64 (build 26100)
system	x86_64, mingw32
ui	RTerm
language	(EN)
collate	English_Singapore.utf8
ctype	English_Singapore.utf8
tz	Asia/Singapore
date	2025-03-09
pandoc	3.2 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)
quarto	1.6.37 @ C:/Program Files/Quarto/bin/quarto.exe/ (via quarto)
knitr	1.49 from RSPM

DESCRIPTION

The `DESCRIPTION` file contains important compendium metadata. Though `DESCRIPTION` file is specific to R package, it can be used to work with research compendia (see below). For further information on how to edit this file, please read https://r-pkgs.org/description.html.

Data Harmonisation

To start the harmonisation of data, run the R script `quarto_script.R` in `reports` folder.

For each cohort, the script will clean the raw data and create a Quarto book for each cohort in html, word and pdf.

This involves copying a specific `yml` file (`_quarto_{cohort name}.yml`) from the `quarto-yaml-template` folder to the project folder `harmonisation_template` and rename it as `_quarto.yml`, overwriting any existing `_quarto.yml` file. Using the `_quarto.yml` file. Quarto will then start running the Quarto scripts in the `reports/{cohort_name}_Cleaning` folder. This involves reading the raw data in the `data-raw/{cohort_name}` folder, placing preprocessing data in the

`reports/{cohort_name}_Cleaning/preprocessed_data` folder, outputting the harmonised data as excel file called `cleaned_{cohort_name}.xlsx` in the `reports` folder. Also, the data harmonisation process documentation will be created in the `books/{cohort_name}` folder as a Quarto book in html, word and pdf.

After data harmonisation, data combining for all cohorts, data filtering and preliminary analysis will be done by copying `_quarto_Prelim.yml` file from the `quarto-yaml-template` folder to the project folder `harmonisation_template` and rename it as `_quarto.yml`, overwriting any existing `_quarto.yml` file. Using the `_quarto.yml` file, Quarto runs the Quarto scripts in the `reports/Combine` folder. Results will be outputted as excel files called `harmonised.xlsx`, `harmonised_batch1.xlsx`, `harmonised_batch2.xlsx` in the `reports` folder. In addition, the preliminary results will be created in the `books/Prelim` folder as a Quarto book in html, word and pdf.

After doing this for each cohort, the script will then create a combined data harmonisation process documentation (for all the cohorts) as a Quarto book in html. The specific `yml` file (`_quarto_all.yml`) in the `quarto-yaml-template` folder will be used and the documentation will be created in the `books/all` folder. Data combining for all cohorts, data filtering and preliminary analysis will also be done by running Quarto scripts in the `reports/Combine` folder.

General Recommendations

- Ensure the workspace is always in a blank state. Use `usethis::use_blank_slate(scope = c("user", "project"))` to create this setting.
- Keep the root of the project as clean as possible
- Store your raw data in `data-raw`
- Document raw data modifications. See `Flowchart.xlsx`.
- Export modified raw data in `reports/{cohort_name}_Cleaning/preprocessed_data`
- Store only **R functions** in `R/`
- Store only **R scripts** and/or **qmd** in `reports/{cohort_name}_Cleaning`
- Built relative paths using `here::here()`
- Call external functions as `{package_name}::{function()}`
- Use `devtools::document()` to update the `NAMESPACE`
- Use `rcompendium::add_dependencies` to update the list of required dependencies in `DESCRIPTION`
- Do not source your functions but use instead `devtools::load_all()`. `devtools::load_all()` will load required dependencies listed in `DESCRIPTION` and R functions stored in `R/`

Part I

Cohort A Cleaning

1 R Package And Environment

1.1 R Packages Used

Here are the R packages used in this analysis.

```
harmonisation::get_r_package_info() |>
  knitr::kable()
```

package	version	date	source
dplyr	1.1.4	2023-11-17	RSPM
fontawesome	0.5.3	2024-11-16	RSPM
forcats	1.0.0	2023-01-29	RSPM
glue	1.8.0	2024-09-30	RSPM
harmonisation	0.0.0.9999	2025-03-09	local
here	1.0.1	2020-12-13	RSPM
htmltools	0.5.8.1	2024-04-04	RSPM
lubridate	1.9.4	2024-12-08	RSPM
magrittr	2.0.3	2022-03-30	RSPM
openxlsx	4.2.8	2025-01-25	RSPM
pointblank	0.12.2	2024-10-23	RSPM
purrr	1.0.4	2025-02-05	RSPM
quarto	1.4.4	2024-07-20	RSPM
reactable	0.4.4	2023-03-12	RSPM
readxl	1.4.4	2025-02-27	RSPM
sessioninfo	1.2.2	2021-12-06	CRAN (R 4.4.2)
stringr	1.5.1	2023-11-14	RSPM
testthat	3.2.3	2025-01-13	RSPM
tibble	3.2.1	2023-03-20	RSPM
tidyr	1.3.1	2024-01-24	RSPM

1.2 R Platform Information

Here are the R platform environment used in this analysis.

```
harmonisation::get_r_platform_info() |>
  knitr::kable()
```

setting	value
version	R version 4.4.2 (2024-10-31 ucrt)
os	Windows 11 x64 (build 26100)
system	x86_64, mingw32
ui	RTerm
language	(EN)
collate	English_Singapore.utf8
ctype	English_Singapore.utf8
tz	Asia/Singapore
date	2025-03-09
pandoc	3.2 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)
quarto	1.6.37 @ C:/Program Files/Quarto/bin/quarto.exe/ (via quarto)
knitr	1.49 from RSPM

1.3 Data dictionary

Check to see if the data dictionary 20250307_data_dictionary.xlsx exists.

```
dict_relative_path <- fs::path(
  "data-raw",
  "data_dictionary",
  params$data_dictionary
)


dict_path <- here::here(dict_relative_path)

if (!file.exists(dict_path)) {
  stop(glue::glue("Input data dictionary {dict_path} cannot be found"))
}
```

2 Read Cohort A Data

2.1 Read Data

We read the data and have the following warnings

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

This warning occurs because we expect the second column `Age` to be numeric but there exists some text columns.

Suppose we ask the collaborator to fix the age column and the collaborator returns a new file. To ensure that there are no messages, we can use `testthat::expect_no_condition`.

Here is an example when it gives an error with the old file

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

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

We can read the new file in the following way. However, this method means that you will need to read the file twice.

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

cohort_A_data <- readxl::read_excel(
  path = here::here("data-raw",
                    "Cohort_A",
                    "data_to_harmonise_age_issue_fixed.xlsx"),
  sheet = "Sheet1",
  col_types = c(
    "text", "numeric"
    )
  )
```

To read the file only once, we can use the tee pipe operator `%T>%`.

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

2.2 Check for unique patient id

We can use `pointblank::rows_distinct` to check if the column `Serial Number` has unique values.

```
cohort_A_data <- readxl::read_excel(
  path = here::here("data-raw",
                    "Cohort_A",
                    "data_to_harmonise_age_issue_fixed.xlsx"),
  sheet = "Sheet1",
  col_types = c(
    "text", "numeric"
    )
  ) %T>%
  testthat::expect_no_condition() |>
  dplyr::rename(cohort_unique_id = "Serial Number") |>
  # Remove rows when the ID value is NA
  dplyr::filter(!is.na(.data[["cohort_unique_id"]])) |>
  dplyr::mutate(
    cohort_unique_id = as.character(cohort_unique_id)
  ) |>
  # Remove white spaces in column names
  dplyr::rename_all(stringr::str_trim) |>
  # Check if cohort id is unique
  pointblank::rows_distinct(
    columns = "cohort_unique_id",
  )
```

2.3 Clean Age columns

Sometimes the collaborator will not give you a new file and will only respond with an email acknowledging that it is an error.

You will need to edit the values yourself. It is best not to edit the file as you may forget to make the manual change if the collaborator gives you a new version a few months later with the same error.

It is also advised to record such changes before data harmonisation.

We read the data with the some issues with the age.

```
cohort_A_data <- readxl::read_excel(
  path = here::here("data-raw",
                    "Cohort_A",
                    "data_to_harmonise.xlsx"),
  sheet = "Sheet1",
  col_types = c(
    "text", "text", "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric"
    )
```

```
  ) %T>%
  testthat::expect_no_condition() |>
  dplyr::rename(cohort_unique_id = "Serial Number") |>
  # Remove rows when the ID value is NA
  dplyr::filter(!is.na(.data[["cohort_unique_id"]])) |>
  dplyr::mutate(
    cohort_unique_id = as.character(cohort_unique_id)
  ) |>
  # Remove white spaces in column names
  dplyr::rename_all(stringr::str_trim) |>
  # Check if cohort id is unique
  pointblank::rows_distinct(
    columns = "cohort_unique_id",
  )
```

As the `Age` is a mixture of type text and numeric in Excel, we need to convert those in type text to the appropriate values.

```
cohort_A_data <- cohort_A_data |>
  dplyr::select(c("cohort_unique_id", "Age")) |>
  dplyr::mutate(
    `Age before` = .data[["Age"]],
    `Age` = dplyr::case_when(
      .data[["Age"]] == "missing" ~ NA,
      .default = .data[["Age"]]
    ),
    `Age before` = forcats::fct(
      as.character(.data[["Age before"]])
    )
  ) |>
  # Check that these are the only patients with missing age
  pointblank::col_vals_null(
    columns = c("Age"),
    preconditions = ~ . %>% dplyr::filter(
      .data[["cohort_unique_id"]] %in% c("A006", "A013")
    )
  )
```

Remove columns `Age before`.

```
cohort_A_data <- cohort_A_data |>
  dplyr::select(-c("Age before"))
```

2.4 Check corrections

We check if the corrections are made based on the collaborator request are made.

- `Age` changed from missing to `NA` for patient `A006` and `A013`.

```
cohort_A_data |>
  # Check if these patient IDs are present
  pointblank::expect_col_vals_make_subset(
    columns = c("cohort_unique_id"),
    set = c("A006", "A013")
  ) |>
  pointblank::expect_col_exists(
    columns = c("Age")
  ) |>
  pointblank::expect_col_vals_expr(
    expr = pointblank::expr(
      dplyr::case_when(
      .data[["cohort_unique_id"]] %in% "A006" ~ is.na(.data[["Age"]])
      )
    )
  ) |>
  pointblank::expect_col_vals_expr(
    expr = pointblank::expr(
      dplyr::case_when(
      .data[["cohort_unique_id"]] %in% "A013" ~ is.na(.data[["Age"]])
      )
    )
  )
```

2.5 Write Preprocessed File

We output data to be used for the next session.

```
cohort_A_data |>
  fst::write_fst(
    path = here::here(params$analysis_folder,
                      params$harmonisation_folder,
                      params$preprocessing_folder,
                      "01_Cohort_A_cleaned.fst")
)
```