ACIC 2022 Track 1 (Patient–Year) Data Structure and Merge Guide

Yiyang Gao

2025-10-07

Contents

1 1. The Four Core Files (Structure and Meaning)

Track 1 provides four CSVs per replicate (e.g., `xxxx = 0001`, replicates range **0001–1200**; files are split across zip parts a/b/c):

File	Level (Uniqueness)	Key(s)	Main contents
`patient_xxxx.csv`	Patient (time-invariant)	`id.patient`	One row per patient. Patient covariates **V1–V5** and clinic ID **id.practice**.
`patient_year_xxxx.csv`	Patient–Year (time-varying)	`id.patient`, `year`	One row per patient per year. Outcome **Y** = monthly average medical expenditure for that year. Patients do **not** appear in all years by design (aging in/out, death).
`practice_xxxx.csv`	Practice (time-invariant)	`id.practice`	One row per practice. Practice covariates **X1–X9**.
`practice_year_xxxx.csv`	Practice–Year (time-varying)	`id.practice`, `year`	One row per practice per year: **Z** (treatment), **post** (post period), **n.patients**, practice-year aggregates (e.g., **V1_C–V5_C** or **V1_avg–V5_avg**), and a practice-level **Y** (drop this for Track-1 outcome modeling).

> **Track-1 outcome:** keep **patient-level** `Y` from `patient_year_xxxx.csv`; **drop** the practice-level `Y` from `practice_year_xxxx.csv`.

2 2. Hierarchical Relationships

Patients are **nested within** practices, and both levels vary over **years**:

```
practice (X1-X9)
  -> practice_year (Z, post, n.patients, aggregated V*, practice Y [drop in Track 1])
       -> patient (V1-V5, id.practice)
            -> patient_year (patient Y, by year)
† Practice-level Y exists but should be dropped for Track 1 outcome modeling.
```

This yields a **cluster-randomised** (practice-level assignment) longitudinal RCT.

3 3. Merge Rules (file merging instructions applied)

1. **Add outcomes to patients:** join `patient` → `patient_year` **by** **`id.patient`**.
2. **Attach practice covariates:** join with `practice` **by** **`id.practice`**.
3. **Attach practice-year treatment & context:** join with `practice_year` **by** **`c(id.practice, year)`**.
4. **Drop practice-level** `Y` to avoid ambiguity.

```r
# ---- Parameters ----
base_dir <- params$data_dir
rid      <- sprintf("%04d", as.integer(params$replicate_id))   # enforce "0001" format
read_engine <- getOption("acic.read_engine", "readr")          # "readr", "vroom", or "fread"

# ---- Paths ----
dir_patient       <- fs::path(base_dir, "patient")
dir_patient_year  <- fs::path(base_dir, "patient_year")
dir_practice      <- fs::path(base_dir, "practice")
dir_practice_year <- fs::path(base_dir, "practice_year")

fname <- function(prefix) glue("acic_{prefix}_{rid}.csv")
files <- list(
  patient       = fs::path(dir_patient,       fname("patient")),
  patient_year  = fs::path(dir_patient_year,  fname("patient_year")),
  practice      = fs::path(dir_practice,      fname("practice")),
  practice_year = fs::path(dir_practice_year, fname("practice_year"))
)

# ---- Null-coalescing helper ----
`%||%` <- function(a, b) if (is.null(a)) b else a

# ---- Column specs (adjust if your files differ) ----
colspec_patient_readr <- readr::cols(
  id.patient  = readr::col_integer(),
  id.practice = readr::col_integer(),
  V1 = readr::col_double(),
  V2 = readr::col_integer(),
  V3 = readr::col_integer(),
  V4 = readr::col_double(),
  V5 = readr::col_character()
```

```
)

colspec_patient_year_readr <- readr::cols(
  id.patient = readr::col_integer(),
  year       = readr::col_integer(),
  Y          = readr::col_double()
)

colspec_practice_readr <- readr::cols(
  id.practice = readr::col_integer(),
  X1 = readr::col_integer(),
  X2 = readr::col_character(),
  X3 = readr::col_integer(),
  X4 = readr::col_character(),
  X5 = readr::col_integer(),
  X6 = readr::col_double(),
  X7 = readr::col_double(),
  X8 = readr::col_double(),
  X9 = readr::col_double()
)

colspec_practice_year_readr <- readr::cols(
  id.practice = readr::col_integer(),
  year        = readr::col_integer(),
  Y           = readr::col_double(),   # to be dropped later
  Z           = readr::col_integer(),
  post        = readr::col_integer(),
  n.patients  = readr::col_integer(),
  .default    = readr::col_double()    # covers V*_avg or V*_C
)

# ---- Reader wrapper ----
read_csv_smart <- function(path, spec_readr = NULL) {
  switch(read_engine,
    readr = readr::read_csv(path, col_types = spec_readr %||% readr::cols(), show_col_types = FALSE),
    vroom = vroom::vroom(path, altrep = TRUE),
    fread = data.table::fread(path, data.table = FALSE),
    stop("Unknown read_engine: ", read_engine)
  )
}

# ---- Safety checks ----
missing <- names(files)[!fs::file_exists(unname(files))]
if (length(missing)) {
  stop(glue("Missing files for rid={rid}: {paste(missing, collapse=', ')} under base_dir='{base_dir}'."
}

# ---- Read ----
patient       <- read_csv_smart(files$patient,       colspec_patient_readr)
patient_year  <- read_csv_smart(files$patient_year,  colspec_patient_year_readr)
practice      <- read_csv_smart(files$practice,      colspec_practice_readr)
practice_year <- read_csv_smart(files$practice_year, colspec_practice_year_readr)
```

```r
# ---- Merge 1: keep observed patient-years only ----
d <- patient %>%
  inner_join(patient_year, by = "id.patient")  # ensures 'year' exists

# ---- Merge 2: + practice (X1-X9) ----
d <- d %>%
  left_join(practice, by = "id.practice")

# ---- Merge 3: + practice_year (Z, post, aggregates) ----
d <- d %>%
  left_join(practice_year, by = c("id.practice", "year"), suffix = c("", ".practice"))

# ---- Keep only the patient-level Y ----
d <- d %>% select(-any_of("Y.practice"))

# ---- Basic checks ----
stopifnot(all(c("id.patient","id.practice","year","Y","Z","post") %in% names(d)))
stopifnot(!any(is.na(d$year)))
glue::glue("Rows: {nrow(d)}, Patients: {dplyr::n_distinct(d$id.patient)}, Practices: {dplyr::n_distinct

# Quick peek
dplyr::glimpse(d, width = 80)
```

4 4. Variable Roles in Modeling

Variable	Level	Typical role
`Y` (patient-level)	Patient–Year	**Outcome** (monthly avg expenditure by year).
`Z`	Practice–Year	**Treatment assignment** (cluster-RCT at practice-year level).
`post`	Practice–Year	Post-intervention period indicator.
`V1-V5`	Patient	Individual covariates (heterogeneity / adjustment / interactions).
`X1-X9`	Practice	Time-invariant practice context (between-cluster differences).
`V*_avg` or `V*_C`, `n.patients`	Practice–Year	Time-varying context / scale, useful for trend control.

Common pitfalls

- Don't mix the two `Y`s—use **patient** `Y` for Track-1.
- Join keys must align: `id.patient`, `id.practice`, `year`.
- Patients won't appear in every year; this is **by design**, not missingness.