admiral

The dplyr for the pharmaceutical industry?

Stefan Thoma & Edoardo Mancini

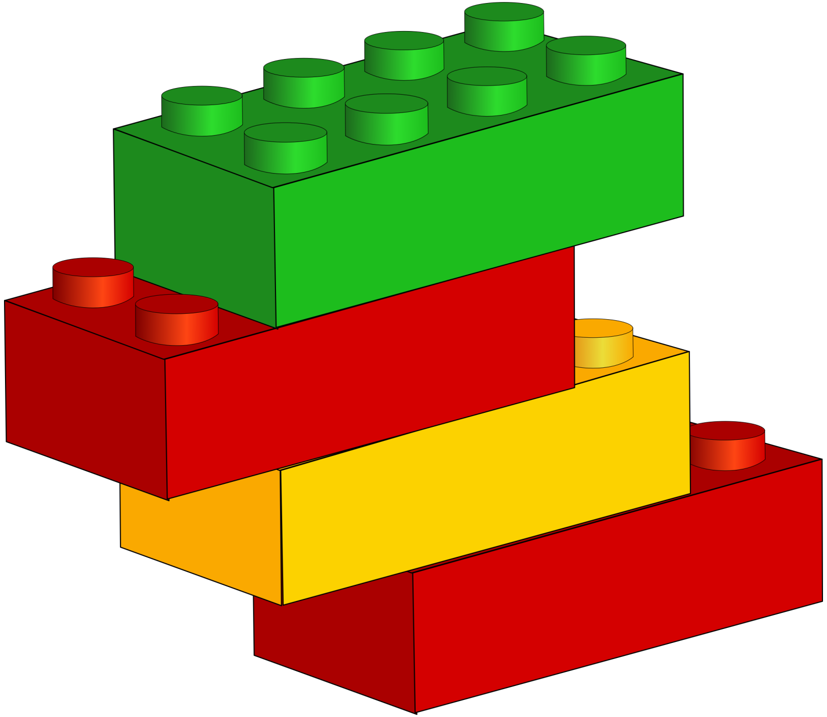
2024-07-04

Another side to “Big Pharma”?

What is ADaM, and why {admiral}?

- In 2016 CDISC introduced the **Analysis Data Model (ADaM)**. This is a therapeutic-area-agnostic set of guidelines for clinical trial dataset structure.
- ADaM lends itself to a modular tool for dataset generation.

cdisc

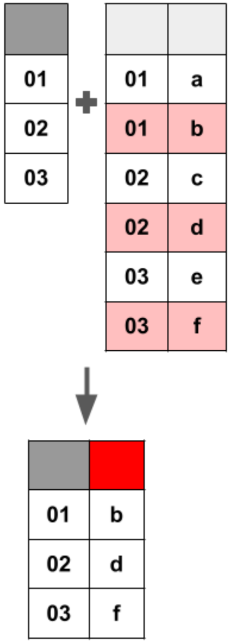


{admiral}

- **Modular functions** extending dplyr, and **manuals** and **templates**
  - Core functionality that can be extended to specific therapeutic areas
  - Specific to very standardized data and processes
- Created by users for users

{admiral} code example

```
adae_dates <- ae %>%
  # join ADSL to AE
  derive_vars_merged(
    dataset_add = adsl,
    new_vars = adsl_derv_vars,
    by_vars = get_admiral_option("subject_keys")
  ) %>%
  # analysis end date/time (AENDTM), imputation flags
(AENDTF/AENTMF)
  slice_derivation(
    derivation = derive_vars_dtm,
    args = params(
      dtc = AEENDTC,
      new_vars_prefix = "AEN",
      date_imputation = "last",
      time_imputation = "last",
      preserve = TRUE
    ),
    derivation_slice(
      filter = AEOUT == "FATAL",
      args = params(
        highest_imputation = "Y",
        max_dates = exprs(DTHDT)
      )
    ),
    derivation_slice(
      filter = TRUE,
      args = params(
        highest_imputation = "M",
        max_dates = NULL
      )
    )
  )
```



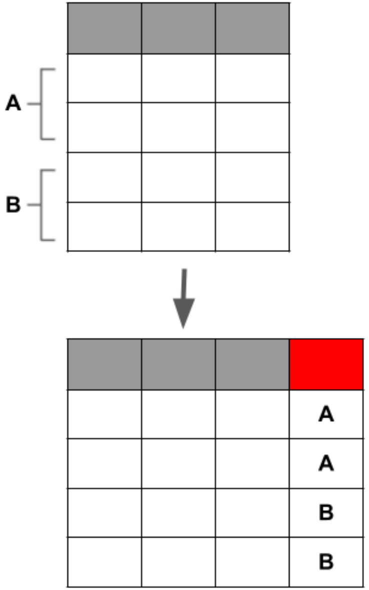
01
02
03

01	a
01	b
02	c
02	d
03	e
03	f

01	b
02	d
03	f

{admiral} code example

```
adae_dates <- ae %>%
  # join ADSL to AE
  derive_vars_merged(
    dataset_add = adsl,
    new_vars = adsl_derv_vars,
    by_vars = get_admiral_option("subject_keys")
  ) %>%
  # analysis end date/time (AENDTM), imputation flags
(AENDTF/AENTMF)
  slice_derivation(
    derivation = derive_vars_dtm,
    args = params(
      dtc = AEENDTC,
      new_vars_prefix = "AEN",
      date_imputation = "last",
      time_imputation = "last",
      preserve = TRUE
    ),
    derivation_slice(
      filter = AEOUT == "FATAL",
      args = params(
        highest_imputation = "Y",
        max_dates = exprs(DTHDT)
      )
    ),
    derivation_slice(
      filter = TRUE,
      args = params(
        highest_imputation = "M",
        max_dates = NULL
      )
    )
  )
```



Goal of {admiral}

- Establish {admiral} workflow as *the* standard for ADaM creation by …
  - building trust in {admiral}
  - encouraging R adoption across the industry
- 15 pharma/biotech companies are already using {admiral}
- Part of overall effort to have an open source workflow for clinical data science, what we call the pharmaverse

Building trust

- Started as a collaboration of GSK and Roche
  - Permissive license, free to use for anyone
  - Commitment and funding from both Roche and GSK makes it hard to ignore
  - Hard for either company to monetize
  - Ensures that the package stays open source and free to use
  - Maintenance
- Open source from the start
- Involve community early on
  - Serve the needs of many, not just of one company

Extension packages

- Same user “feel” & language
- Same development practices
- Compatible modularity
- Stability and trust
  - Below par {admiral} extension package harms reputation of {admiral}
  - Less trust means fewer users and less adoption of open source in the industry

Setting them up for success

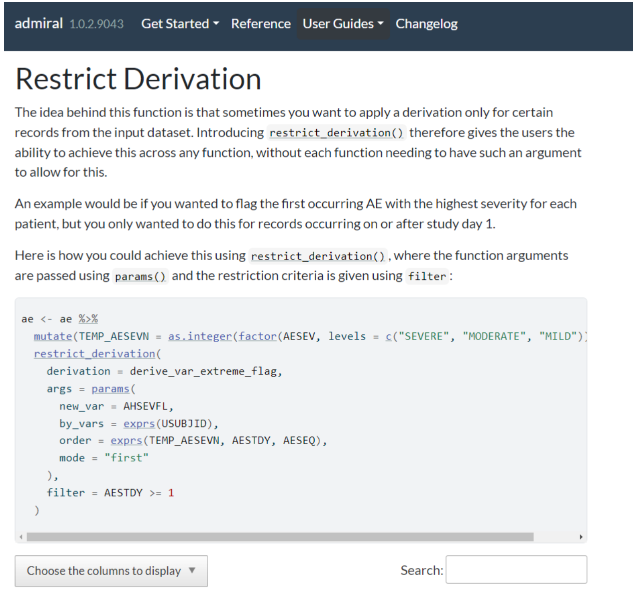
- Instructions
  - {admiraldev} rules & templates
  - Git Structure & CICD
  - One core member representative
- Expectation management:
  - Small scope!
  - Effort required
  - Assigned FTE

A success story: {admiralvaccine}

- The COVID-19 pandemic was a worldwide catastrophe, but it was also an opportunity to **collaborate further** within pharma…
- … Enter {admiralvaccine}!

{admiral}: Looking ahead

Training models on {admiral} documentation: could genAI be the next frontier?



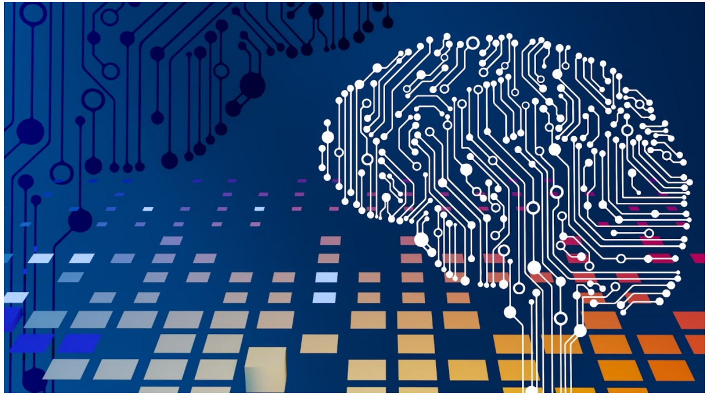
Restrict Derivation

The idea behind this function is that sometimes you want to apply a derivation only for certain records from the input dataset. Introducing `restrict_derivation()` therefore gives the users the ability to achieve this across any function, without each function needing to have such an argument to allow for this.

An example would be if you wanted to flag the first occurring AE with the highest severity for each patient, but you only wanted to do this for records occurring on or after study day 1.

Here is how you could achieve this using `restrict_derivation()`, where the function arguments are passed using `params()` and the restriction criteria is given using `filter`:

```
ae <- ae %>%
  mutate(TEMP_AESEVN = as.integer(factor(AESEV, levels = c("SEVERE", "MODERATE", "MILD"))
  restrict_derivation(
    derivation = derive_var_extreme_flag,
    args = params(
      new_var = AHSEVFL,
      by_vars = exprs(USUBJID),
      order = exprs(TEMP_AESEVN, AESTDY, AESEQ),
      mode = "first"
    ),
    filter = AESTDY >= 1
  )
```



Q&A and resources

- {admiral}
- Cheat sheet
- {admiraldev}
- Pharmaverse blog
- Pharmaverse
- CDISC