PHUSE 2025 - Paper OS07

Building Custom Open-Source Statistical Programming Curriculum

Phil Bowsher, Posit/RStudio PBC
Yannick Vandendijck, Johnson & Johnson
Kieran Martin, Novartis

Abstract

This paper will introduce methods for building custom open-source statistical programming curriculum. While many areas of drug development utilize open-source, the focus of this paper will be on late-stage clinical reporting and specifically R, which some Python sprinkled in.

Introduction

This paper will highlight training initiatives at 3 large pharmas, each with similar but not identical approaches. The paper will then bridge from these efforts into how organizations can mimic with similar online and freely available curriculum.

Open-Source Training at JnJ, GSK & Roche

JnJ

At the R/Pharma Gathering at Harvard University in 2019, Janssen R&D presented its major undertaking to train users on R to help establish the foundation for clinical reporting in open-source. Paulo Bargo highlights below how Janssen R&D established a crowdsourcing approach where statisticians were divided in groups to tackle weekly training and engagements with Group Leads that spanned 8 months. Posit Cloud was used as the platform for housing the code, data and curriculum. The live environments were supplemented with internal videos and reference material. The initial training focused on graphics/visualizations using ggplot2 of the Tidyverse. You can read more about the initiative here:

Slides:
https://github.com/rinpharma/2019_presentations/blob/master/talks_folder/2019-Bargo-Using_RStudio_Cloud_to_advance_R_proficiency.pdf

Video: https://posit.co/resources/videos/scaling-data-science-with-r-at-janssen-pharmaceuticals/

https://www.youtube.com/watch?v=C2Suw5Trh0A

Over time, the training initiatives at Janssen evolved and an update was provided by Yannick Vandendijck at Phuse EU 2024 on SAIL.R as a comprehensive cross-functional initiative. Yannick discussed how the “live” classroom approach was used to support a statistical programming team consisting of approximately 300 programmers.

Paper: https://www.lexjansen.com/phuse/2024/pd/PAP_PD08.pdf

GSK

At the Phuse EU 2023 Conference, Andy Nicholls presented on the GSK Journey to achieve 50% of code in R by the end of 2025 for GSK Biostatistics consisting of around 900 Statisticians. Similar to JnJ, GSK embarked on its open-source journey around late 2017 by building a new in-house R training course, focused on the Tidyverse ecosystem. GSK created 3 primary courses available internally via R Markdown/Quarto books. Like JnJ, the focus was on the Tidyverse with data manipulation; graphics; and simulation. Further courses were added focused on advanced topics such as Shiny.

Slides: https://phuse.s3.eu-central-1.amazonaws.com/Archive/2023/Connect/EU/Birmingham/PRE_TT09.pdf

Paper: https://www.lexjansen.com/phuse/2023/tt/PAP_TT09.pdf

Video: https://www.youtube.com/watch?v=xDrt6txplek

Roche

Roche documented its learning journey for open source and a shift towards a language agnostic framework at Phuse EU in 2023: https://phuse.s3.eu-central-1.amazonaws.com/Archive/2023/Connect/EU/Birmingham/PRE_TT05.pdf

In the paper, Roche highlights its in-house **Data Science University**, which curates often freely available and public materials into learning paths for students. The self learning system is supplemented with additional guided help.

Roche explained further the effort in a webinar **“Shifting to an Open-Source Backbone in Clinical Trials with Roche”**: https://youtu.be/nqJsLSLd39A?t=1949

The learning system is built and directed via self assessments, and aligning users to the correct learning path, which utilizes often freely available learning material from the community. Roche bridges the learning material to active projects, which include various efforts such as hackathons and applications in real settings. This work often happens in teams and aligns users with mentors. Moreover, Roche/Genentech has created public learning pathways via Coursera such as: https://www.coursera.org/learn/hands-on-clinical-reporting-using-r

Kieran Martin presented at a single day event for PSI on **One-day Event: Change Management for Moving to R/Open-Source**. His talk, **Getting Ready for submission with R,** discussed Roche’s approach to onboarding, and how project teams joined in waves which helps the organization keep improving its tools along the way and how tool development is driven by iterative feedback. The presentation also discussed the *R Submissions Working Group*, which has active pilots with the FDA. These pilots are excellent resources for learning and can be forked and adjusted as needed (https://rconsortium.github.io/submissions-wg/). Janssen recently discussed an internal pilot based on this work at a Phuse event in New Jersey by Alicia Humphreys and Linshan Yuan here: https://phuse.s3.eu-central-1.amazonaws.com/Archive/2025/SDE/US/West+Windsor/PRE_WestWindsor03.pdf

Reviewing and running internal pilots using the repositories can be an excellent learning process such as studying the repo and eCTD submission package examples: https://github.com/RConsortium/submissions-pilot1

A similar theme with all the groups above is that they provide active support and engagement for new users. GSK talks about its AccelerateR Team, a group of internal R experts that supports GSK Biostatistics study teams by answering questions and mentoring during the transition. While workshops can spark interest and excitement for learning, on the spot support is a required ingredient for GSK as discussed here:

The Need for Speed - AccelerateR-ing R Adoption in GSK: https://www.youtube.com/watch?v=VDu2qdpYko8

Other organizations have documented its R training progression such as Novo Nordisk, Sanofi and PPD in the following documents:

https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/POS_PP02.pdf

https://www.psiweb.org/docs/default-source/conference/2025-conference-slides/monday-9-june/2-r-adoption-change-management---martin_brown.pdf?sfvrsn=c23eafdb_2

https://phuse.s3.eu-central-1.amazonaws.com/Archive/2022/Connect/EU/Belfast/PRE_TT03.pdf

Polyglot Statistical Computing Environments

Many pharmaceutical companies are migrating to or creating Next-Gen Polyglot Statistical Computing Environments (SCE) where open source languages are used for clinical reporting. James Black discusses “The importance of the SCE in enabling our shift to open-source data science” in the webinar here:
https://pos.it/enable-oss

As organizations migrate users to these new environments, training and change management are a major focus. Below we will discuss key themes for such transitions.

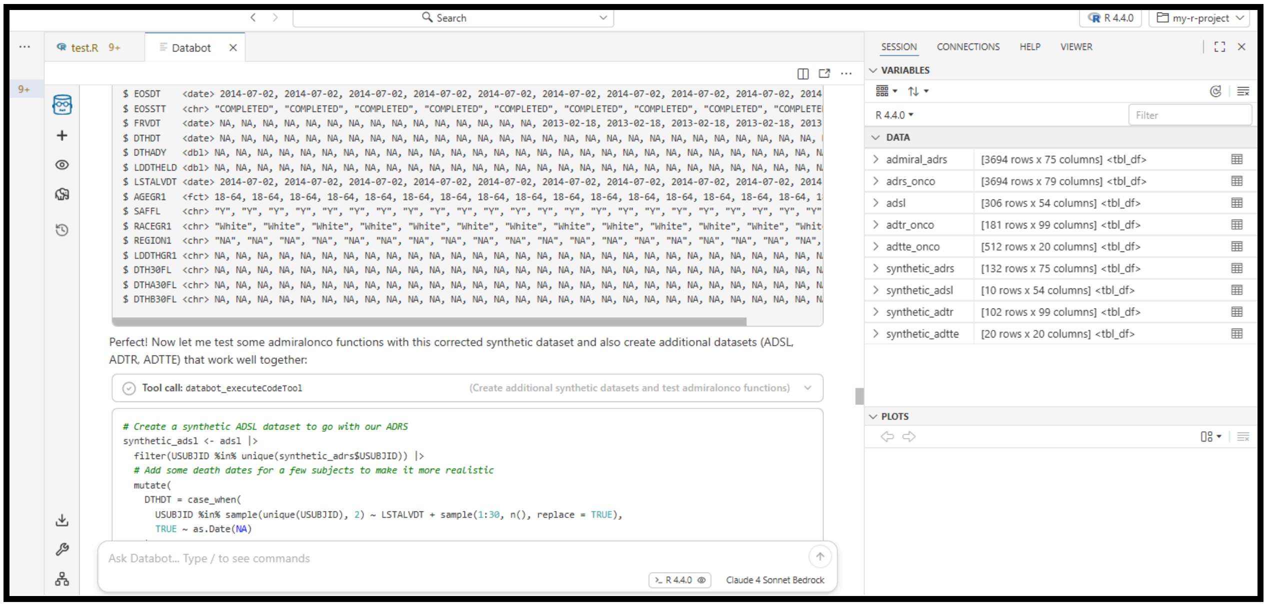
Clinical Reporting Data

As Roche mentioned above, critically important is using data similar to that used internally for clinical reporting, often in raw or CDISC format. As with both GSK and JnJ, it is important to keep the training closely tied to the clinical reporting functions, where data are raw, SDTM or ADaM datasets. While it is preferred to use in-house data, sometimes this is not possible. Luckily the CDISC organization provides sample data at: https://github.com/cdiscorg/sdtm-adam-pilot-project

Other methods for implementing this data are through the follow R packages:

- pharmaverseraw: https://github.com/pharmaverse/pharmaverseraw/
- pharmaversesdtm: https://github.com/pharmaverse/pharmaversesdtm
- pharmaverseadam: https://github.com/pharmaverse/pharmaverseadam
- random.cdisc.data: https://github.com/insightsengineering/random.cdisc.data

The data contained in the R packages above allow for complete end-to-end examples to be executed. It has been mentioned that it would be nice for further Admiral extension packages such as admiralonco, to have test data other than the CDISC pilot project data. One option is to use AI to help create synthetic data. Below is a test using Positron Databot to help create synthetic datasets for admiralonco:



```
$ EOSDT    <date> 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014
$ EOSSTT   <chr> "COMPLETED", "COMPLETED", "COMPLETED", "COMPLETED", "COMPLETED", "COMPLETED", "COMPLETED", "COMPLETED", "COMPLETE
$ FRVDT    <date> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 2013-02-18, 2013-02-18, 2013-02-18, 2013-02-18, 2013
$ DTHDT    <date> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
$ DTHADY   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
$ LDDTHELD <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
$ LSTALVDT <date> 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014-07-02, 2014
$ AGEGR1   <fct> 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 18-64, 1
$ SAFFL    <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y"
$ RACEGR1  <chr> "White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "White", "Whit
$ REGION1  <chr> "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA", "NA",
$ LDDTHGR1 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
$ DTH30FL  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
$ DTHA30FL <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
$ DTHB30FL <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
```

Perfect! Now let me test some admiralonco functions with this corrected synthetic dataset and also create additional datasets (ADSL, ADTR, ADTTE) that work well together:

Tool call: databot_executeCodeTool (Create additional synthetic datasets and test admiralonco functions)

```
# Create a synthetic ADSL dataset to go with our ADRS
synthetic_adsl <- adsl |>
  filter(USUBJID %in% unique(synthetic_adrs$USUBJID)) |>
  # Add some death dates for a few subjects to make it more realistic
  mutate(
    DTHDT = case_when(
      USUBJID %in% sample(unique(USUBJID), 2) ~ LSTALVDT + sample(1:30, n(), replace = TRUE),
      TRUE ~ as.Date(NA)
```

Ask Databot... Type / to see commands

R 4.4.0 Claude 4 Sonnet Bedrock

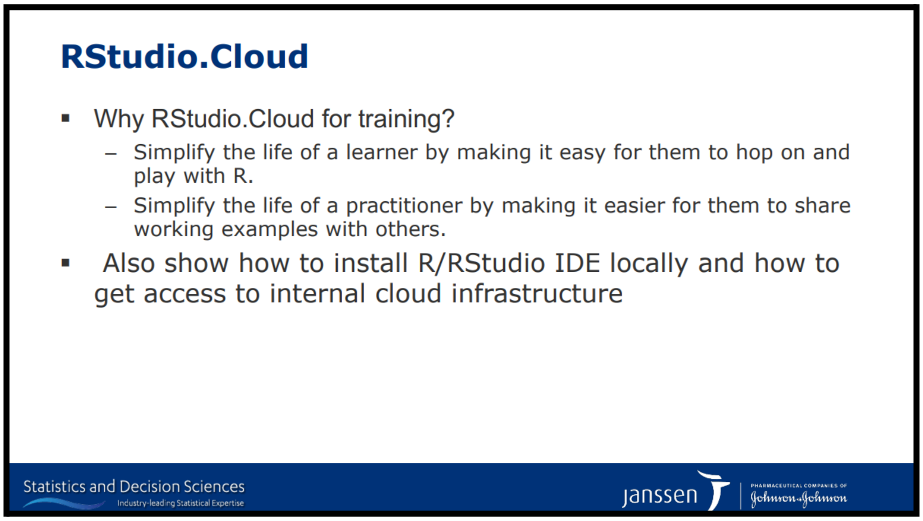
DATA	
admiral_adrs	[3694 rows x 75 columns] <tbl_df>
adrs_onco	[3694 rows x 79 columns] <tbl_df>
adsl	[306 rows x 54 columns] <tbl_df>
adtr_onco	[181 rows x 99 columns] <tbl_df>
adtte_onco	[512 rows x 20 columns] <tbl_df>
synthetic_adrs	[132 rows x 75 columns] <tbl_df>
synthetic_adsl	[10 rows x 54 columns] <tbl_df>
synthetic_adtr	[102 rows x 99 columns] <tbl_df>
synthetic_adtte	[20 rows x 20 columns] <tbl_df>

More info about Databot can be found here: https://positron.posit.co/databot.html

For SAS datasets in ADAM format, a small subset can be found here: https://github.com/elong0527/r4csr/tree/main/data-adam

Environment

Originally Janssen R&D used Posit.Cloud for trainings:



RStudio.Cloud

- Why RStudio.Cloud for training?
  - Simplify the life of a learner by making it easy for them to hop on and play with R.
  - Simplify the life of a practitioner by making it easier for them to share working examples with others.
- Also show how to install R/RStudio IDE locally and how to get access to internal cloud infrastructure

Statistics and Decision Sciences
Industry-leading Statistical Expertise
janssen
PHARMACEUTICAL COMPANIES OF Johnson & Johnson

Posit.Cloud is a great environment for building internal training as it allows organizations to create Spaces, which are classrooms administered by Instructors. Instructors fill the Spaces with R/Python Projects that are usually connected to an organization's github repository that contains the code and data. You can learn more about Posit Cloud's classroom management here: https://docs.posit.co/cloud/guide/collaborate/

One of the key advantages of Posit Cloud is that it makes it very easy for users to join preconfigured RStudio environments, where packages, code and data are ready to go! In fact, the *R/Pharma Conference* runs most of its workshops using this method, making it very easy for users to come to a public workshop and almost immediately be ready to execute and run code. Other organizations use Posit Cloud for these features like Harvard Medical School: https://posit.co/about/customer-stories/harvard-medical/

Please note that while there is a free tier of Posit Cloud, setting up organizations and spaces requires a paid account.

As Janssen R&D realized in its 2019 training effort, many users bridged from Posit Cloud to the internal SCE or environment. Both GSK and Janssen R&D have transitioned to teaching and providing support to users in its internal open-source environment. One method to bridge a training environment is to have an internal SCE for training. This separate environment would be installed and operated in the organization's internal Cloud provider, and would allow training to use internal data and intellectual property but would balance the needs of a playground for learning and have the needed AI configurations. This setup helps keep the GxP environments separate and allows users to mimic the processes and technology of the approved environment, such as Git integrations and CICD. **Posit Team Dedicated** is one such offering or organizations can spin up its own environment.

Tidyverse, Quarto/R Markdown & Quarto Curriculum

GSK as well as JnJ, started training users with the Tidyverse. Below are some core books and resources to help get started:

- R for Data Science Book: https://r4ds.hadley.nz/
- Tidy Modeling with R Book: https://www.tmwr.org/
- Mastering Shiny Book: https://mastering-shiny.org/
- Quarto: https://r4ds.hadley.nz/quarto.html

Another important package for clinical reporting is the Lubridate package of the Tidyverse. Working with clinical data often entails transforming between dates and times. Many organizations use Lubridate for this purpose. Below are resources for curriculum:

- Lubridate: https://lubridate.tidyverse.org/
- Dates and times: https://r4ds.hadley.nz/datetimes.html

A key topic is helping SAS users bridge to open source processes. Rather than focus on translating SAS to R, it is often suggested for users to focus on the learning process and approaching R and Python which allows users to fully utilize open source's potential. Below are some resources:

- How to learn R as a SAS user: https://posit.co/blog/how-to-learn-r-as-a-sas-user/
- SAS <-> R :: CHEAT SHEET: https://raw.githubusercontent.com/rstudio/cheatsheets/main/sas-r.pdf
- SAS Vs R in Pharma: https://rstudio.github.io/cheatsheets/SASvsRinPharma.pdf

Comparing Analysis Method Implementations in Software (CAMIS)(https://psiaims.github.io/CAMIS/) is focused on helping to demystify results in statistical analysis methods and results between primarily SAS, R, and Python programming languages. The website and resources provide comparisons and explanations which is great for learning. Curriculum can incorporate this material to help towards efficacy analysis for organizations that are early in the open source journey.

Tidyverse Curriculum

For teaching, there is Data Science in a Box, which contains materials to teach the Tidyverse, and is freely-available and open-source. Materials and examples can be found here: https://datasciencebox.org/

The code and examples can be forked and modified here: https://github.com/tidyverse/datascience-box

For Python, a great place to start is:

Python for Data Analysis: https://wesmckinney.com/book/

Python Polars: The Definitive Guide: https://polarsguide.com/

Clinical-Tables-in-Python-with-Great-Tables:
https://github.com/philbowsher/Clinical-Tables-in-Python-with-Great-Tables

A grammar of graphics for Python: https://plotnine.org/

Another great place to find curriculum materials is from the Posit Conference. For 2025, all the workshops are listed here: https://posit-conf-2025.github.io/workshops/

If organizations are looking for assistance in creating curriculum or having users take an evidence-based training course, Posit Academy does this for the Tidverse as seen here: https://posit.co/about/customer-stories/astrazeneca/

Pharma & Pharmaverse Curriculum

There are many excellent sources of Pharma curriculum for organizations to use, fork and customize as needed. Below will discuss various sources and tactics.

1. Pharmaverse Examples: https://pharmaverse.github.io/examples/

The Pharmaverse Examples were built to help provide end-to-end examples of using pharmaverse packages together for clinical reporting analyses, from raw data to SDTM, ADaM and Tables/Listings/Graphs. The examples use the test raw datasets mentioned above, pharmaverseraw, pharmaversesdtm and pharmaverseadam.

The Pharmaverse Examples website provides a link that will **instantly launch a free live and interactive Posit Cloud environment** (preconfigured with all required package installations). There is a *"Launch Posit Cloud"* on the site that will launch an RStudio Environment with the code, data and examples! Moreover, organizations can run the examples locally, by downloading the repository or forking it directly within RStudio or Positron. This process was explained in detail within the PharmaSUG paper **"Trying Out Positron: New IDE for Statistical Programming"** available here: https://pharmasug.org/proceedings/2025/HT/PharmaSUG-2025-HT-397.pdf

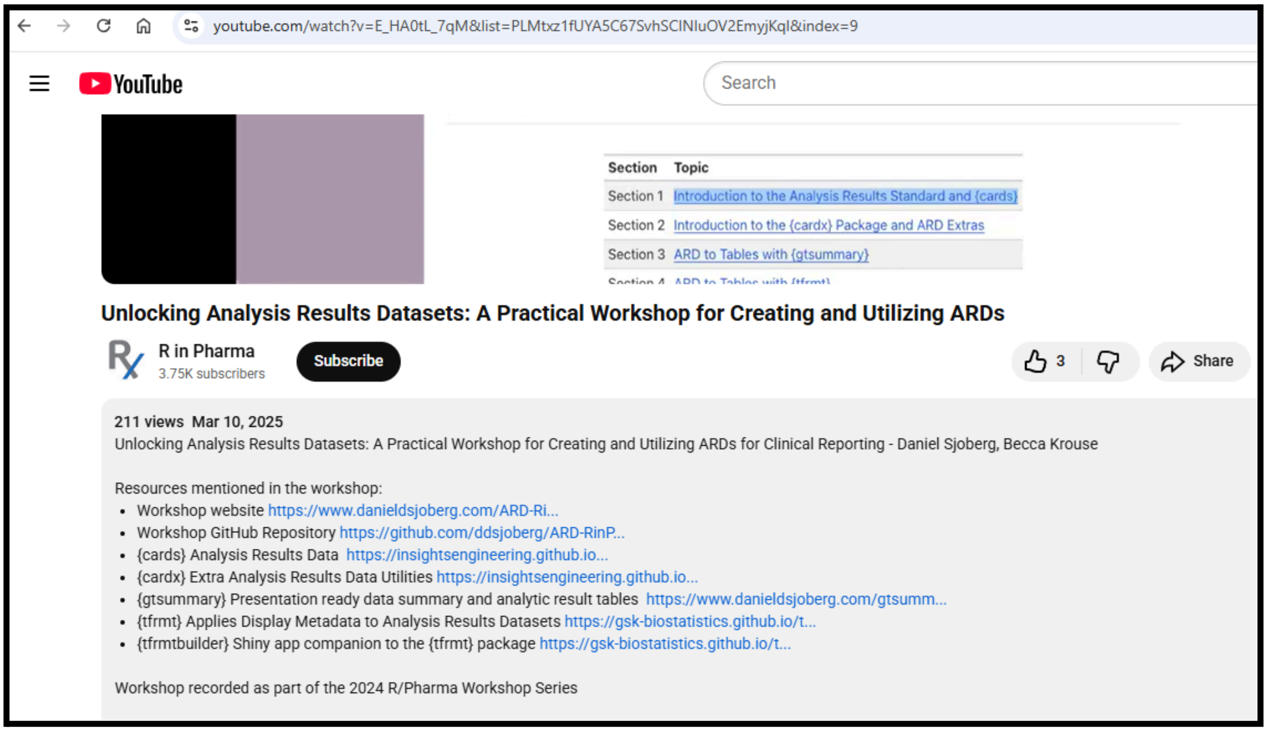
At the Posit 2025 Conference, there was the workshop, **End-to-End Submissions in R with the Pharmaverse**, which was built off of the examples above. The main page can be found at https://posit-conf-2025.github.io/pharmaverse/ and its github here: https://github.com/posit-conf-2025/pharmaverse

Another option is to fork the repository and customize the examples as needed. For example, many pharmas have unique ways for handling data and TLFs, such as using gtsummary, tfrmt, tables, flextable etc. An example of this is made available from Pfizer. Pfizer created a curriculum to train programmers in clinical research who are familiar with SAS for data transformation and visualization. You can find, fork and use the examples here: https://github.com/pfizer-opensource/pharma-hands-on-exercises

These examples can be made available internally, and or customized to match your pharmas specifications. Other organizations have created similar curriculum and made it available publicly for forking such as:

Bayer Oncology SBU SAS and R Book: https://bayer-group.github.io/sas2r/ https://github.com/Bayer-Group/sas2r
Merck R for Clinical Study Reports and Submission Book: https://r4csr.org/ https://github.com/elong0527/r4csr/

The R/Pharma Conference host its workshops on *youtube* and often shares github resources in the video descriptions as seen here: https://www.youtube.com/c/RinPharma


youtube.com/watch?v=E_HA0tL_7qM&list=PLMtxz1fUYA5C67SvhSCINluOV2EmyjKql&index=9
YouTube
Search
R in Pharma
3.75K subscribers
Subscribe
3
Share

Section	Topic
Section 1	Introduction to the Analysis Results Standard and {cards}
Section 2	Introduction to the {cardx} Package and ARD Extras
Section 3	ARD to Tables with {gtsummary}
Section 4	ARD to Tables with {tfrmt}

Unlocking Analysis Results Datasets: A Practical Workshop for Creating and Utilizing ARDs

211 views Mar 10, 2025
Unlocking Analysis Results Datasets: A Practical Workshop for Creating and Utilizing ARDs for Clinical Reporting - Daniel Sjoberg, Becca Krouse

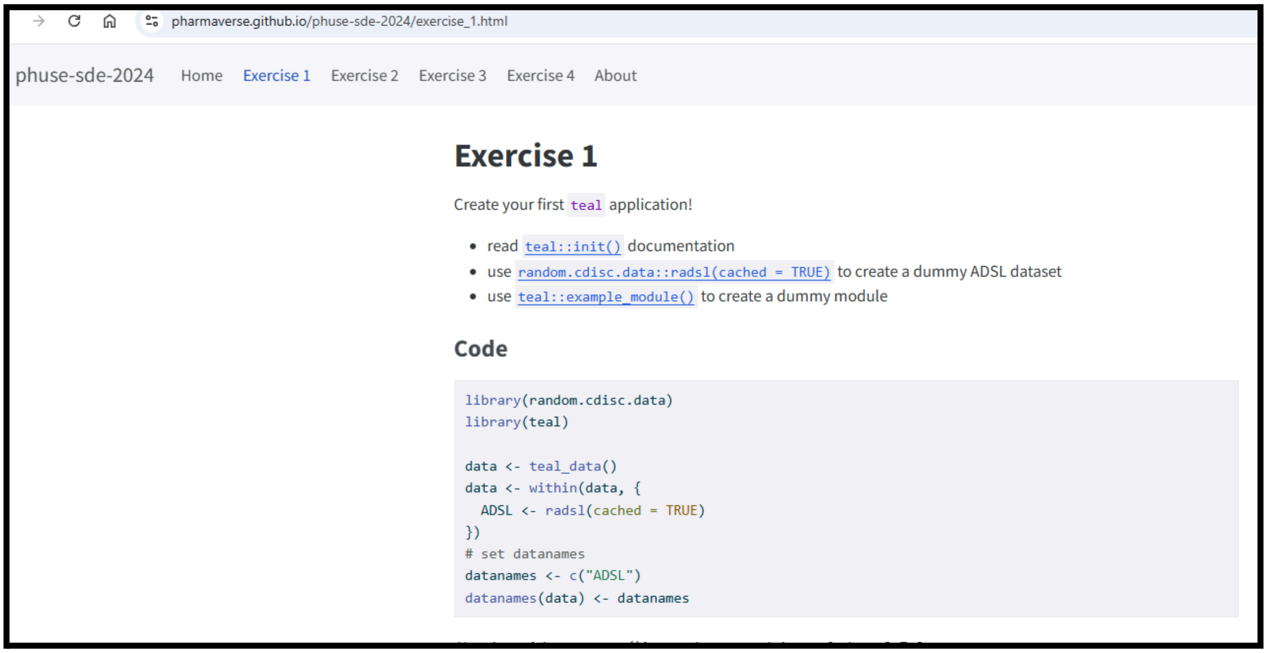
Resources mentioned in the workshop:

- Workshop website https://www.danieldsjoberg.com/ARD-Ri...
- Workshop GitHub Repository https://github.com/ddsjoberg/ARD-RinP...
- {cards} Analysis Results Data https://insightsengineering.github.io...
- {cardx} Extra Analysis Results Data Utilities https://insightsengineering.github.io...
- {gtsummary} Presentation ready data summary and analytic result tables https://www.danieldsjoberg.com/gtsumm...
- {tfrmt} Applies Display Metadata to Analysis Results Datasets https://gsk-biostatistics.github.io/t...
- {tfrmtbuilder} Shiny app companion to the {tfrmt} package https://gsk-biostatistics.github.io/t...

Workshop recorded as part of the 2024 R/Pharma Workshop Series

As your organization advances in its adoption of open-source, there may be interest in Shiny. Phuse held a single day event in 2024 in Basel on training in Shiny with Teal which can be found here: https://pharmaverse.github.io/phuse-sde-2024/

This training is innovative in many ways. First, it takes users through the process of creating interactive exploratory data analysis with Teal, a Shiny framework for pharma. The training does this through 4 different modules, all of which provide the code and also an interactive coding environment implemented with *WebR*, which provides R compiled for the browser and Node.js using WebAssembly, via Emscripten. It does this for Shiny with *Shinyliv*e, which runs the examples (Shiny apps) in the browser, without a hosted server, using WebAssembly via the webR project as seen here:



Exercise 1

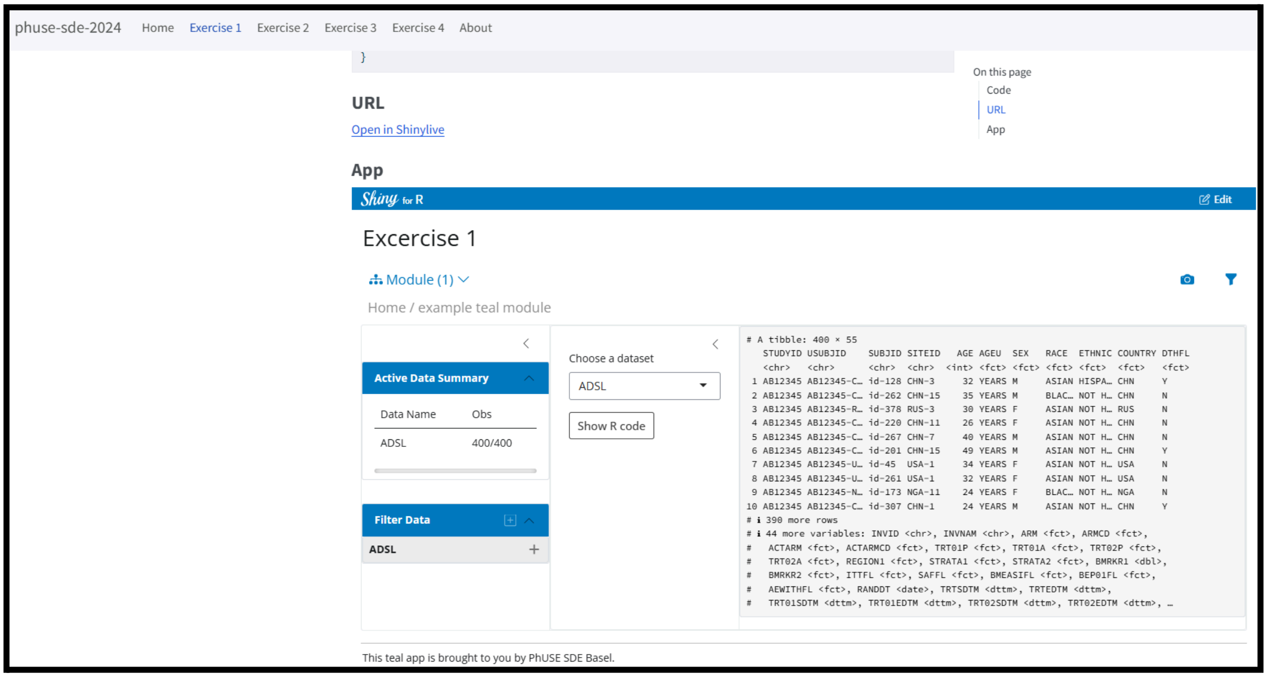
Create your first `teal` application!

- read `teal::init()` documentation
- use `random.cdisc.data::radsl(cached = TRUE)` to create a dummy ADSL dataset
- use `teal::example_module()` to create a dummy module

Code

```r
library(random.cdisc.data)
library(teal)

data <- teal_data()
data <- within(data, {
  ADSL <- radsl(cached = TRUE)
})
# set datanames
datanames <- c("ADSL")
datanames(data) <- datanames
```



```
}
```

URL

Open in Shinylive

App

Excercise 1

Active Data Summary

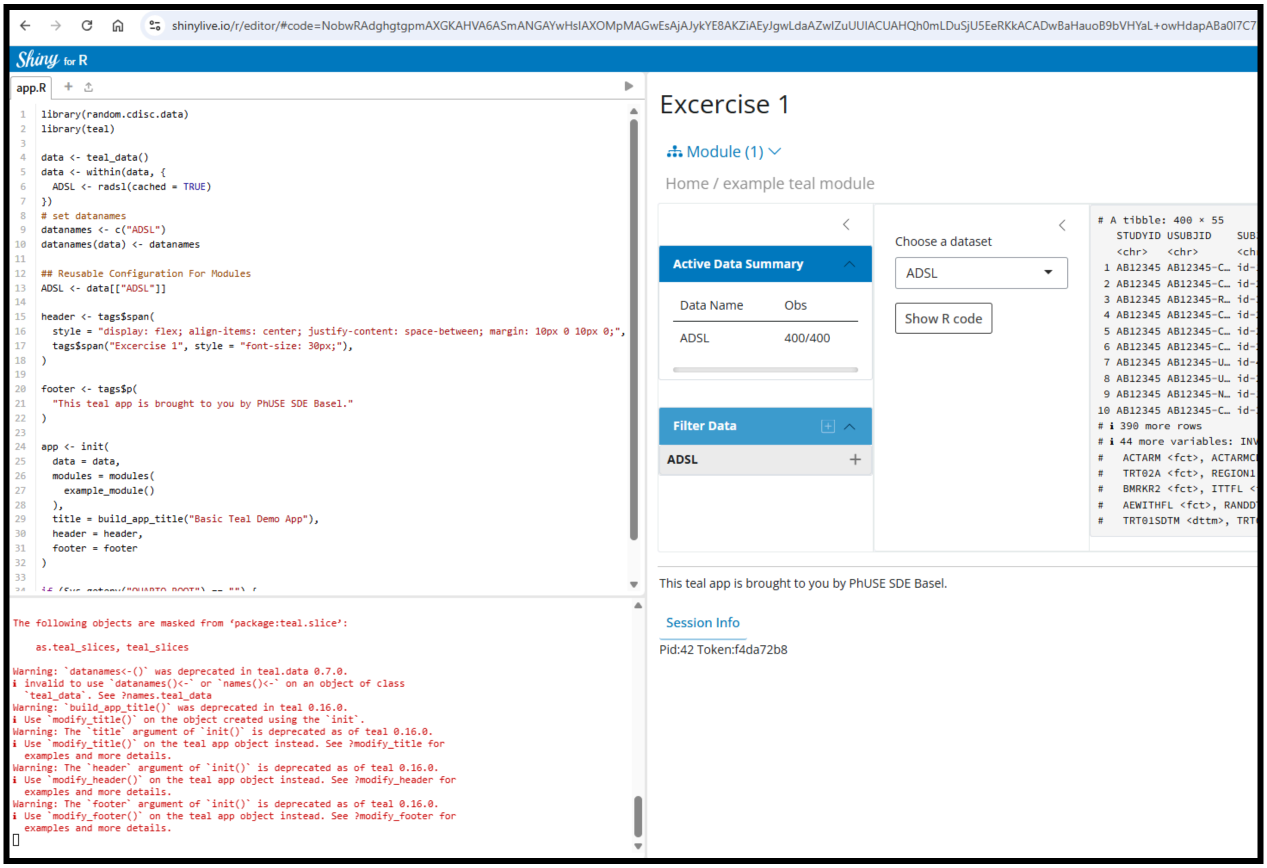
Data Name	Obs
ADSL	400/400

Filter Data

Choose a dataset: ADSL

```
# A tibble: 400 × 55
   STUDYID USUBJID    SUBJID SITEID   AGE AGEU  SEX   RACE  ETHNIC COUNTRY DTHFL
   <chr>   <chr>      <chr>  <chr>  <int> <fct> <fct> <fct> <fct>  <fct>   <fct>
 1 AB12345 AB12345-C… id-128 CHN-3     32 YEARS M     ASIAN HISPA… CHN     Y
 2 AB12345 AB12345-C… id-262 CHN-15    35 YEARS M     BLAC… NOT H… CHN     N
 3 AB12345 AB12345-R… id-378 RUS-3     30 YEARS F     ASIAN NOT H… RUS     N
 4 AB12345 AB12345-C… id-220 CHN-11    26 YEARS F     ASIAN NOT H… CHN     N
 5 AB12345 AB12345-C… id-267 CHN-7     40 YEARS M     ASIAN NOT H… CHN     N
 6 AB12345 AB12345-C… id-201 CHN-15    49 YEARS M     ASIAN NOT H… CHN     Y
 7 AB12345 AB12345-U… id-45  USA-1     34 YEARS F     ASIAN NOT H… USA     N
 8 AB12345 AB12345-U… id-261 USA-1     32 YEARS F     ASIAN NOT H… USA     N
 9 AB12345 AB12345-N… id-173 NGA-11    24 YEARS F     BLAC… NOT H… NGA     N
10 AB12345 AB12345-C… id-307 CHN-1     24 YEARS M     ASIAN NOT H… CHN     Y
# ℹ 390 more rows
# ℹ 44 more variables: INVID <chr>, INVNAM <chr>, ARM <fct>, ARMCD <fct>,
#   ACTARM <fct>, ACTARMCD <fct>, TRT01P <fct>, TRT01A <fct>, TRT02P <fct>,
#   TRT02A <fct>, REGION1 <fct>, STRATA1 <fct>, STRATA2 <fct>, BMRKR1 <dbl>,
#   BMRKR2 <fct>, ITTFL <fct>, SAFFL <fct>, BMEASIFL <fct>, BEP01FL <fct>,
#   AEWITHFL <fct>, RANDDT <date>, TRTSDTM <dttm>, TRTEDTM <dttm>,
#   TRT01SDTM <dttm>, TRT01EDTM <dttm>, TRT02SDTM <dttm>, TRT02EDTM <dttm>, …
```

This teal app is brought to you by PhUSE SDE Basel.


shinylive.io/r/editor/#code=NobwRAdghgtgpmAXGKAHVA6ASmANGAYwHsIAXOMpMAGwEsAjAJykYE8AKZiAEyJgwLdaAZwIZuUUIACUAHQh0mLDuSjU5EeRKkACADwBaHauoB9bVHYaL+owHdapABa0I7C7
Shiny for R
app.R
Excercise 1
Module (1)
Home / example teal module
Active Data Summary
Filter Data
ADSL
Choose a dataset
ADSL
Show R code
Session Info

```r
library(random.cdisc.data)
library(teal)

data <- teal_data()
data <- within(data, {
  ADSL <- radsl(cached = TRUE)
})
# set datanames
datanames <- c("ADSL")
datanames(data) <- datanames

## Reusable Configuration For Modules
ADSL <- data[["ADSL"]]

header <- tags$span(
  style = "display: flex; align-items: center; justify-content: space-between; margin: 10px 0 10px 0;",
  tags$span("Excercise 1", style = "font-size: 30px;"),
)

footer <- tags$p(
  "This teal app is brought to you by PhUSE SDE Basel."
)

app <- init(
  data = data,
  modules = modules(
    example_module()
  ),
  title = build_app_title("Basic Teal Demo App"),
  header = header,
  footer = footer
)
```

```
The following objects are masked from ‘package:teal.slice’:

    as.teal_slices, teal_slices

Warning: `datanames<-()` was deprecated in teal.data 0.7.0.
ℹ invalid to use `datanames()<-` or `names()<-` on an object of class
  `teal_data`. See ?names.teal_data
Warning: `build_app_title()` was deprecated in teal 0.16.0.
ℹ Use `modify_title()` on the object created using the `init`.
Warning: The `title` argument of `init()` is deprecated as of teal 0.16.0.
ℹ Use `modify_title()` on the teal app object instead. See ?modify_title for
  examples and more details.
Warning: The `header` argument of `init()` is deprecated as of teal 0.16.0.
ℹ Use `modify_header()` on the teal app object instead. See ?modify_header for
  examples and more details.
Warning: The `footer` argument of `init()` is deprecated as of teal 0.16.0.
ℹ Use `modify_footer()` on the teal app object instead. See ?modify_footer for
  examples and more details.
```

Data Name	Obs
ADSL	400/400

```
# A tibble: 400 × 55
   STUDYID USUBJID    SUB
   <chr>   <chr>      <ch
 1 AB12345 AB12345-C… id-
 2 AB12345 AB12345-C… id-
 3 AB12345 AB12345-R… id-
 4 AB12345 AB12345-C… id-
 5 AB12345 AB12345-C… id-
 6 AB12345 AB12345-C… id-
 7 AB12345 AB12345-U… id-
 8 AB12345 AB12345-U… id-
 9 AB12345 AB12345-N… id-
10 AB12345 AB12345-C… id-
# ℹ 390 more rows
# ℹ 44 more variables: INV
#   ACTARM <fct>, ACTARMC
#   TRT02A <fct>, REGION1
#   BMRKR2 <fct>, ITTFL <
#   AEWITHFL <fct>, RANDD
#   TRT01SDTM <dttm>, TRT
```

This teal app is brought to you by PhUSE SDE Basel.

Pid:42 Token:f4da72b8

All of the code for the workshop can be found here: https://github.com/pharmaverse/phuse-sde-2024

This exciting new way of hosting workshops introduces many new ways for organizations to host interactive curriculum within the organizations. For example, organizations can host serverless books, made with Quarto, all of which provide coding environments for running pharmaverse code! Anyone, anywhere can now try out the Phuse Shiny apps via:

https://shinylive.io/r/editor/#code=NobwRAdghgtgpmAXGKAHVA6ASmANGAYwHsIAXOMpMAGwEsAjAJykYE8AKZiAEyJgwLdaAZwIZuUUIACUAHQh0mLDuSjU5EeRKkACADwBaHauoB9bVHYaL+owHdapABa0I7C7h0h5OnQEEAEQBIABlbHWZuYWp2AigCJzhuHQBeHQAVLABVAFENAF8NAGIdYThSHQtoeGEtSShquGFwgnZZMEDQ9ut6xuF3eulwqtgm+Xkikqw4AFdhKHpqOB0AYRIAM1oAcxnmUloSHQAxIkYdAFkibhml2ohOsMNK+uBgdof2gF1P8YhEqG4cDOTykW2EABJhKgGm0IL5hKRWEtUjp2klodQoKxEDp1ksAB4Abh0am2EAMjjgMGEOIIFHHIjGJACs5vt1qwDMQyPScVD4nADPRynY4BRiTAWFtXDiAIwABIQ+J0cp08sVysJ7VwPmMUDBkOhbnaOXxdMYBBEyxlWtKiORaXa6xlpAMwloAC84DiAMwKondbUQDTyJ1EBnhUEQ1Cw3ztdluZomEnoHQiHRMlgzLZOCqklg6ViZ9OsHQABScWSCOR0QQC1YAQlAytQMO15MHoCmnq5HDHnro0h4dTArjcmiiR9dbn3fHB8bBUEtTJOx1YddJA759qR7emZrRqNxTGhUKZt0s2mBG26CBk4GodAEqfm-OgAzr-oCzmlP0DN7iiDDIEUVDBl21+Wh1h0dgglYYQMC2coKAAN0vABFLI-CwdIAHlTCwHCcPSbpUgdMASO8OFShcCBWFfaMT3BfdPEYspGGQoECjAfJPiAA

For more Teal examples, be sure to review the Teal Gallery, as with all the examples, can be forked via github and customized as needed!

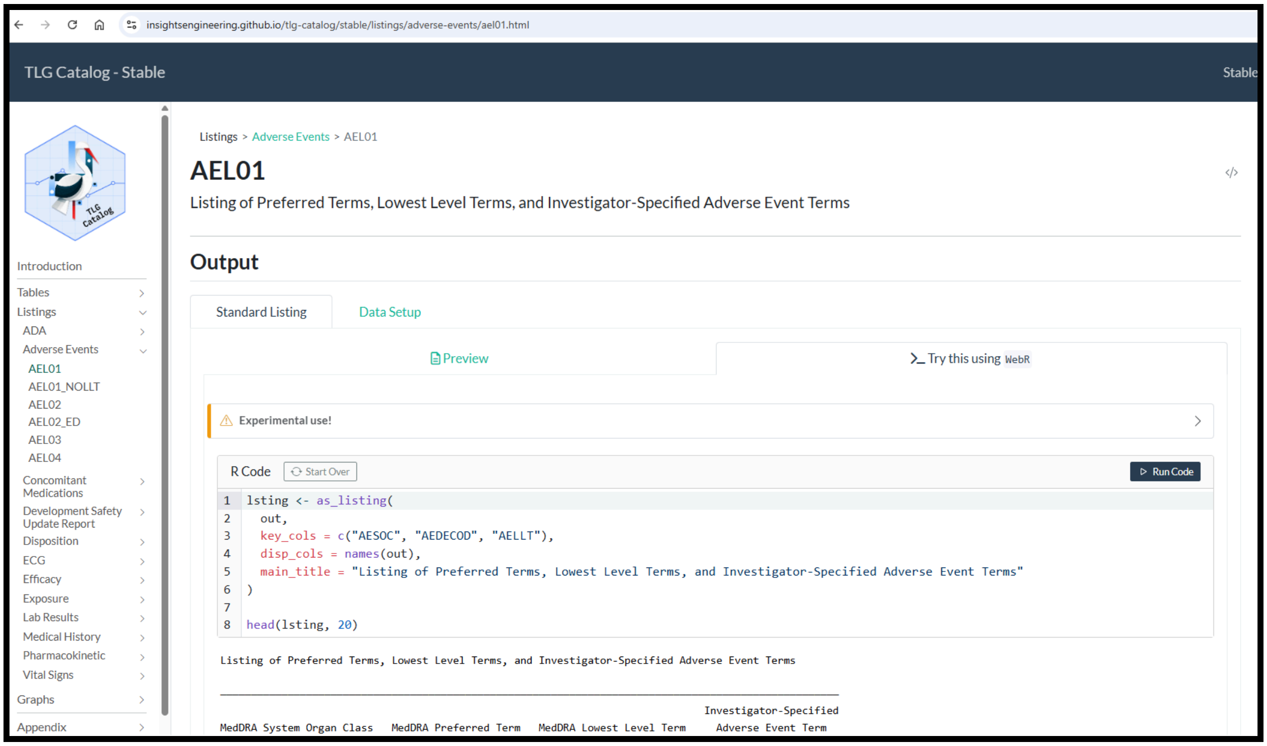
https://insightsengineering.github.io/teal.gallery/
https://github.com/insightsengineering/teal.gallery/

Examples are running on Shinyapps.io via https://rinpharma.shinyapps.io/nest_efficacy_dev/

Roche also provides its Standard TLFs available online via the TLG catalog. It is a catalog of Tables, Listings, and Graphs for clinical trials generated using Roche NEST packages. These could be customized as needed for other tables packages.You can see the live examples here: https://insightsengineering.github.io/tlg-catalog/stable/

Code are here: https://github.com/insightsengineering/tlg-catalog

The catalog also runs all examples using webR and in some cases Shinylive:



AEL01

Listing of Preferred Terms, Lowest Level Terms, and Investigator-Specified Adverse Event Terms

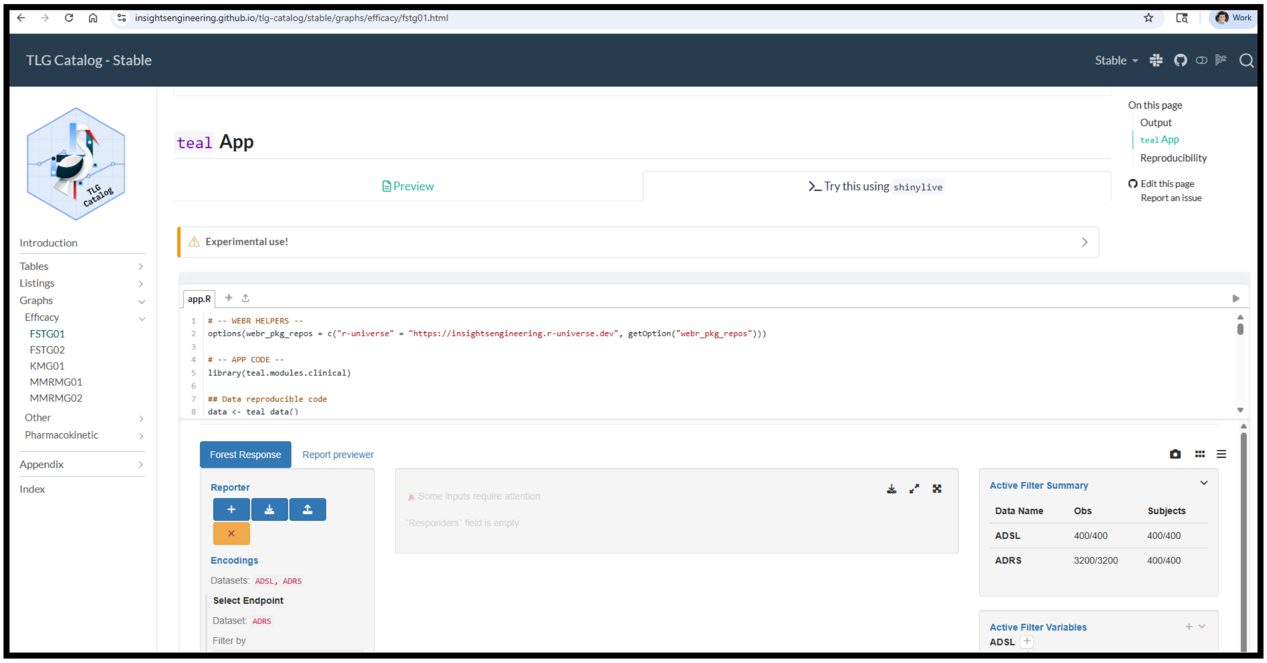
Output

```
lsting <- as_listing(
  out,
  key_cols = c("AESOC", "AEDECOD", "AELLT"),
  disp_cols = names(out),
  main_title = "Listing of Preferred Terms, Lowest Level Terms, and Investigator-Specified Adverse Event Terms"
)

head(lsting, 20)
```

```
Listing of Preferred Terms, Lowest Level Terms, and Investigator-Specified Adverse Event Terms

————————————————————————————————————————————————————————————————————————————————————————————————
                                                                       Investigator-Specified
MedDRA System Organ Class   MedDRA Preferred Term   MedDRA Lowest Level Term    Adverse Event Term
```



teal App

```
# -- WEBR HELPERS --
options(webr_pkg_repos = c("r-universe" = "https://insightsengineering.r-universe.dev", getOption("webr_pkg_repos")))

# -- APP CODE --
library(teal.modules.clinical)

## Data reproducible code
data <- teal_data()
```

Datasets: ADSL, ADRS

Dataset: ADRS

Some inputs require attention

`Responders` field is empty

Data Name	Obs	Subjects
ADSL	400/400	400/400
ADRS	3200/3200	400/400

Another fantastic tool to use for learning is the Shiny Assistant, an AI-powered tool for helping to create Shiny applications, in R or Python. Users can prompt and ask questions about Shiny, or even create a Shiny application from scratch. It can also be used to make changes to an existing application. Shiny Assistant builds on top of the tools mentioned above, like Shinylive, and combines it with AI. The tool can be seen here: https://gallery.shinyapps.io/assistant/

Lastly, creating or forking custom pharma curriculum can be challenging. There are organizations to help such as Atorus, A2-AI,Jumping Rivers, and Appsilon that can help:

https://a2-ai.com/tech/

https://www.appsilon.com/post/help-with-shiny

https://www.jumpingrivers.com/training/

https://www.atorusresearch.com/atorus-academy-overview/

Git

At the recent R/Pharma Summit in Atlanta (https://rinpharma.com/docs/summit/), one of the key items discussed for integrating open-source reporting for clinical trials is the need for SAS users to learn and adopt Git. Below will cover various areas for helping to learn Git. Kieran Martin explained how Roche uses Git at the Phuse EU 2024 in the following links: https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/PRE_TT06.pdf

https://www.lexjansen.com/phuse/2024/tt/PAP_TT06.pdf

GSK also uses Git to help measure its success in attaining 50% of its code in R by 2025. Alanah Jonas explains how Git is a critical part of the environment here: https://www.lexjansen.com/phuse/2024/os/PAP_OS13.pdf

https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/PRE_OS13.pdf

The first place to start is the Happy Git and GitHub for the useR book: https://happygitwithr.com/

It provides an excellent introduction to Git and integration with RStudio. For Positron, the following resource is helpful: https://positron.posit.co/git.html

The openstatsware organization provides a workshop called “Good Software Engineering Practice for R Packages” which provides excellent course on how to engineer reliable R packages used in statistics here: https://openstatsware.github.io/shortcourse-iscb2025/

https://github.com/openstatsware/shortcourse-iscb2025

The Git Cheatsheet in RStudio is helpful too: https://github.com/rstudio/cheatsheets/blob/main/git-github.pdf

From there, the “Deploying reproducible analytics environments for regulated use cases” workshop covers many important topics such as Git and how organizations manage reproducible environments: https://posit-conf-2025.github.io/reproducible-environments/#/title-slide

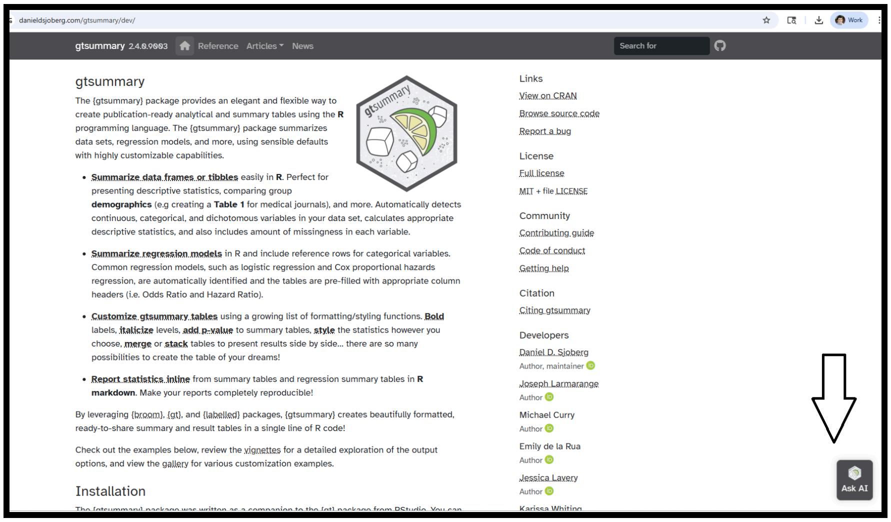
The PHUSE working PharmaDevOps provides information on getting started with CICD and its value. More information can be found below: https://phuse-org.github.io/devops/

https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/POS_PP03.pdf

AI

AI is often a tool used by programmers to help explain or provide code for training. Posit provides two tools in Positron, and coming to RStudio, to help with GenAI programming. The first is Databot (https://positron.posit.co/databot.html), an agentic assistant designed to support exploratory data analysis in Python or R. While it supports various LLM backends, such as Anthropic’s Claude, early tests have shown it is excellent for translating SAS to R and explaining functions needed to replicate legacy code in open-source. Second is Positron Assistant (https://positron.posit.co/assistant), an AI client in Positron that provides LLM integration within Positron, both for chat and for inline completions. Like Databot, it can have various LLM backends and does a great job of explaining programming logic and generating code.

In addition to the tools above, many package authors are adding kapa.ai features to the sites which provides users the ability to prompt and ask questions for packages. Below is an example for gtsummary:


danieldsjoberg.com/gtsummary/dev/

gtsummary 2.4.0.9003 Reference Articles News Search for

gtsummary

The {gtsummary} package provides an elegant and flexible way to create publication-ready analytical and summary tables using the **R** programming language. The {gtsummary} package summarizes data sets, regression models, and more, using sensible defaults with highly customizable capabilities.

- **Summarize data frames or tibbles** easily in **R**. Perfect for presenting descriptive statistics, comparing group **demographics** (e.g creating a **Table 1** for medical journals), and more. Automatically detects continuous, categorical, and dichotomous variables in your data set, calculates appropriate descriptive statistics, and also includes amount of missingness in each variable.
- **Summarize regression models** in R and include reference rows for categorical variables. Common regression models, such as logistic regression and Cox proportional hazards regression, are automatically identified and the tables are pre-filled with appropriate column headers (i.e. Odds Ratio and Hazard Ratio).
- **Customize gtsummary tables** using a growing list of formatting/styling functions. **Bold** labels, **italicize** levels, **add p-value** to summary tables, **style** the statistics however you choose, **merge** or **stack** tables to present results side by side… there are so many possibilities to create the table of your dreams!
- **Report statistics inline** from summary tables and regression summary tables in **R markdown**. Make your reports completely reproducible!

By leveraging {broom}, {gt}, and {labelled} packages, {gtsummary} creates beautifully formatted, ready-to-share summary and result tables in a single line of R code!

Check out the examples below, review the vignettes for a detailed exploration of the output options, and view the gallery for various customization examples.

Installation

The {gtsummary} package was written as a companion to the {gt} package from RStudio. You can

Links

View on CRAN

Browse source code

Report a bug

License

Full license

MIT + file LICENSE

Community

Contributing guide

Code of conduct

Getting help

Citation

Citing gtsummary

Developers

Daniel D. Sjoberg

Author, maintainer

Joseph Larmarange

Author

Michael Curry

Author

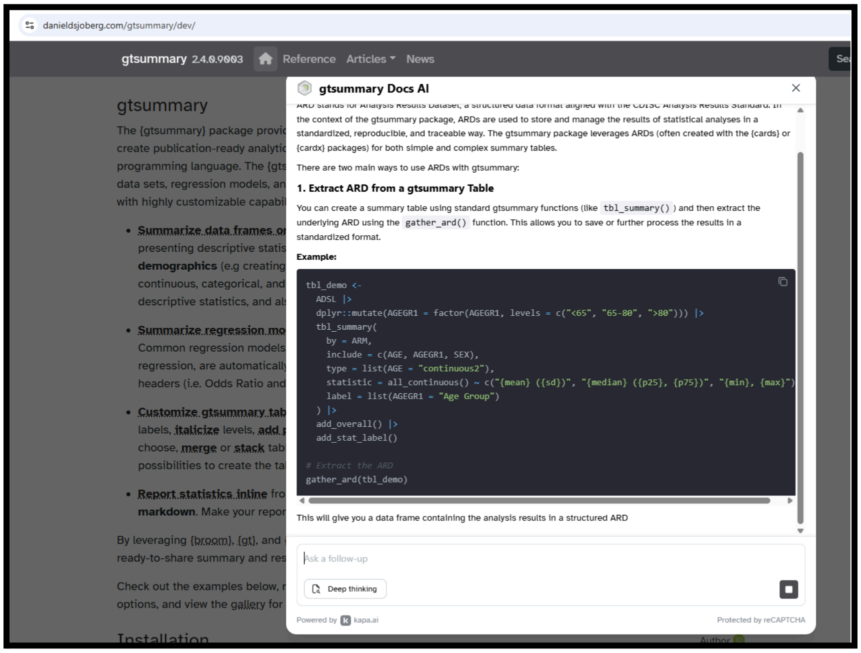
Emily de la Rua

Author

Jessica Lavery

Author

Karissa Whiting


danieldsjoberg.com/gtsummary/dev/

gtsummary 2.4.0.9003 Reference Articles News

gtsummary Docs AI

ARD stands for Analysis Results Dataset, a structured data format aligned with the CDISC Analysis Results Standard. In the context of the gtsummary package, ARDs are used to store and manage the results of statistical analyses in a standardized, reproducible, and traceable way. The gtsummary package leverages ARDs (often created with the {cards} or {cardx} packages) for both simple and complex summary tables.

There are two main ways to use ARDs with gtsummary:

1. Extract ARD from a gtsummary Table

You can create a summary table using standard gtsummary functions (like `tbl_summary()`) and then extract the underlying ARD using the `gather_ard()` function. This allows you to save or further process the results in a standardized format.

Example:

```
tbl_demo <-
  ADSL |>
  dplyr::mutate(AGEGR1 = factor(AGEGR1, levels = c("<65", "65-80", ">80"))) |>
  tbl_summary(
    by = ARM,
    include = c(AGE, AGEGR1, SEX),
    type = list(AGE = "continuous2"),
    statistic = all_continuous() ~ c("{mean} ({sd})", "{median} ({p25}, {p75})", "{min}, {max}")
    label = list(AGEGR1 = "Age Group")
  ) |>
  add_overall() |>
  add_stat_label()

# Extract the ARD
gather_ard(tbl_demo)
```

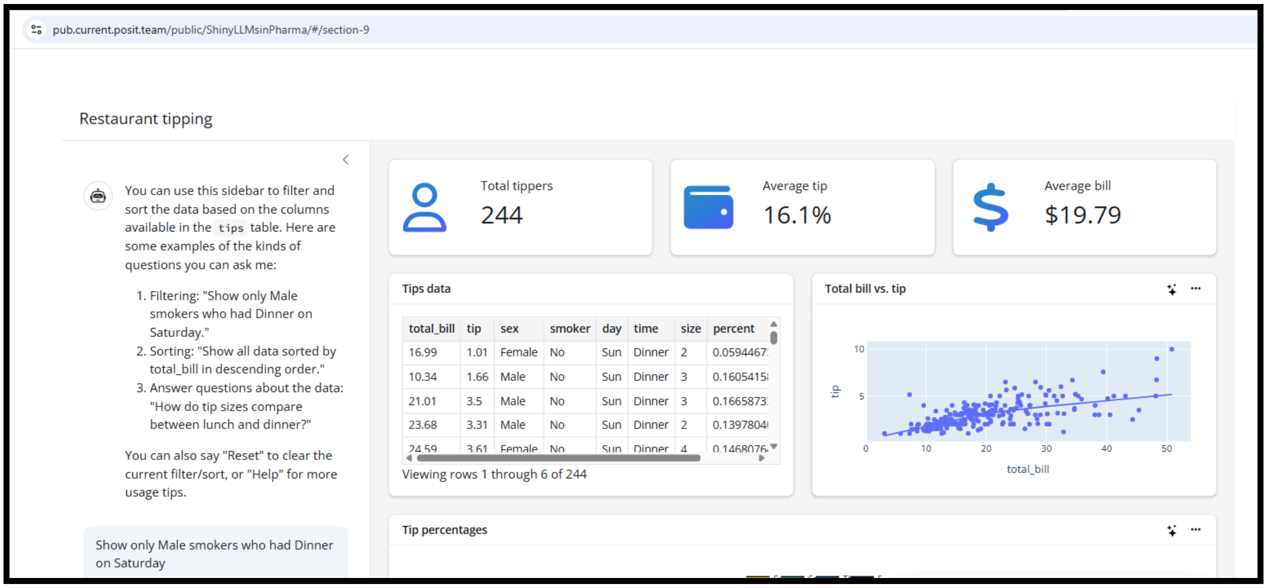
This will give you a data frame containing the analysis results in a structured ARD

Moreover, many organizations are building training documents using Shiny and incorporate training and prompting features via LLM and Shiny packages like:

https://ellmer.tidyverse.org/

https://posit-dev.github.io/shinychat/

https://github.com/posit-dev/querychat



Restaurant tipping

You can use this sidebar to filter and sort the data based on the columns available in the `tips` table. Here are some examples of the kinds of questions you can ask me:

1. Filtering: "Show only Male smokers who had Dinner on Saturday."
2. Sorting: "Show all data sorted by total_bill in descending order."
3. Answer questions about the data: "How do tip sizes compare between lunch and dinner?"

You can also say "Reset" to clear the current filter/sort, or "Help" for more usage tips.

Show only Male smokers who had Dinner on Saturday

Total tippers 244

Average tip 16.1%

Average bill $19.79

Tips data

total_bill	tip	sex	smoker	day	time	size	percent
16.99	1.01	Female	No	Sun	Dinner	2	0.0594467
10.34	1.66	Male	No	Sun	Dinner	3	0.1605415
21.01	3.5	Male	No	Sun	Dinner	3	0.1665873
23.68	3.31	Male	No	Sun	Dinner	2	0.1397804
24.59	3.61	Female	No	Sun	Dinner	4	0.1468076

Viewing rows 1 through 6 of 244

Total bill vs. tip

Tip percentages

You can learn more about Shiny and LLMs here: https://pub.current.posit.team/public/ShinyLLMsinPharma/

Users now also have the ability to chat with Shiny apps and ask questions via ggbot2: https://github.com/tidyverse/ggbot2

ggbot2 uses the shinyrealtime package: https://github.com/posit-dev/shinyrealtime

shinyrealtime integrates OpenAI’s Realtime API with Shiny apps written in either R or Python.I believe there will be more to come from these exciting innovations for learning R and Python in 2026!

Lastly, context/skill files are an important part of AI workflows. These instructions help provide starting context for LLMs and one could see a library or community of files shared for people to use when training users on specific workflows. One such tool is Talk Box, where organizations can use the PromptBuilder to create basic context and identity for your AI assistant. More can be found here: https://rich-iannone.github.io/talk-box/user-guide/prompt-builder.html

Posit is currently hosting free AI workshops at the link here to learn more about these topics: https://posit.co/ai-contact/

Conclusion & Contact

The information above highlights how organizations can find and use publicly available training materials to create and scale its own training programs, helping to transform users' skill for using open-source for clinical reporting. the information above highlights how open source languages like R and Python can help modernize clinical processes and how new innovative features and capabilities can help organizations train users.

Phil Bowsher

phil@posit.co

https://github.com/philbowsher

Posit PBC