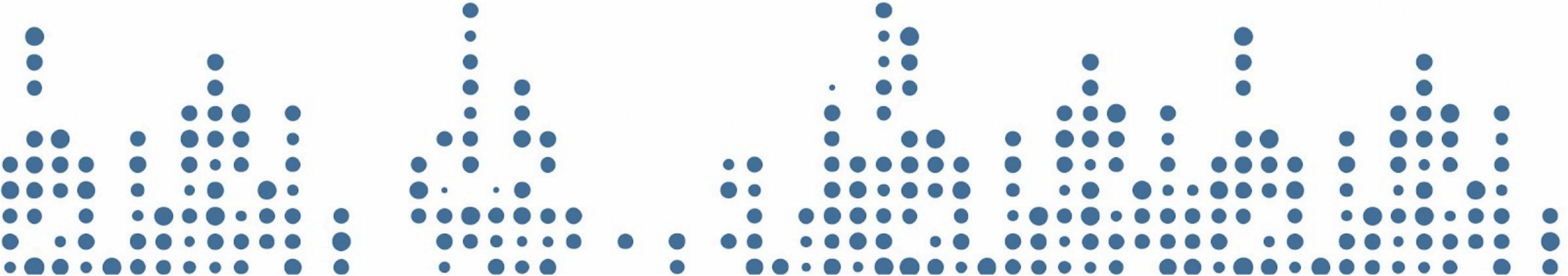
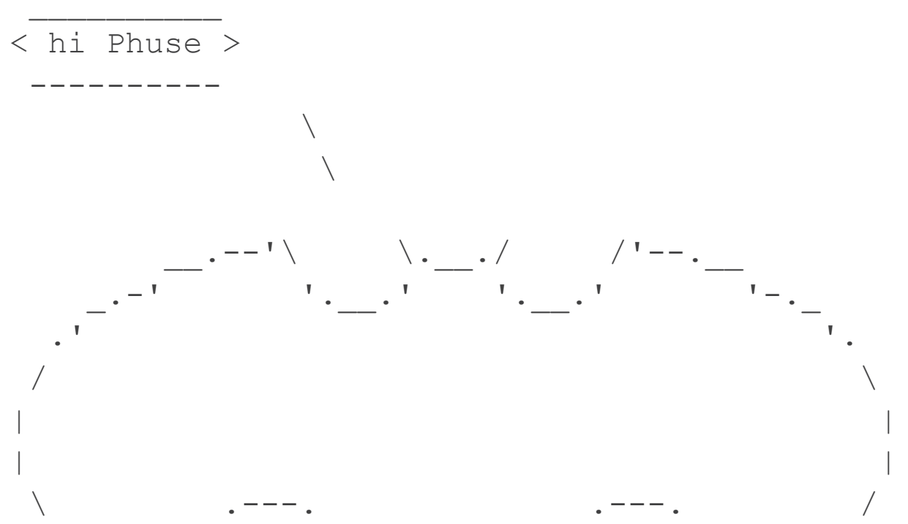
Building Custom Open-Source Statistical Programming Curriculum

Phil Bowsher | 2025 Nov 18 *Phuse OS07*



Hello!



Hello!

@philbowsher

youtube.com/rinpharma

rinpharma.com

pharmaverse.github.io/examples/

Building Custom...

👩:

Open-Source

Building Custom...

Open-Source Statistical Programming

Building Custom...

👦👂:

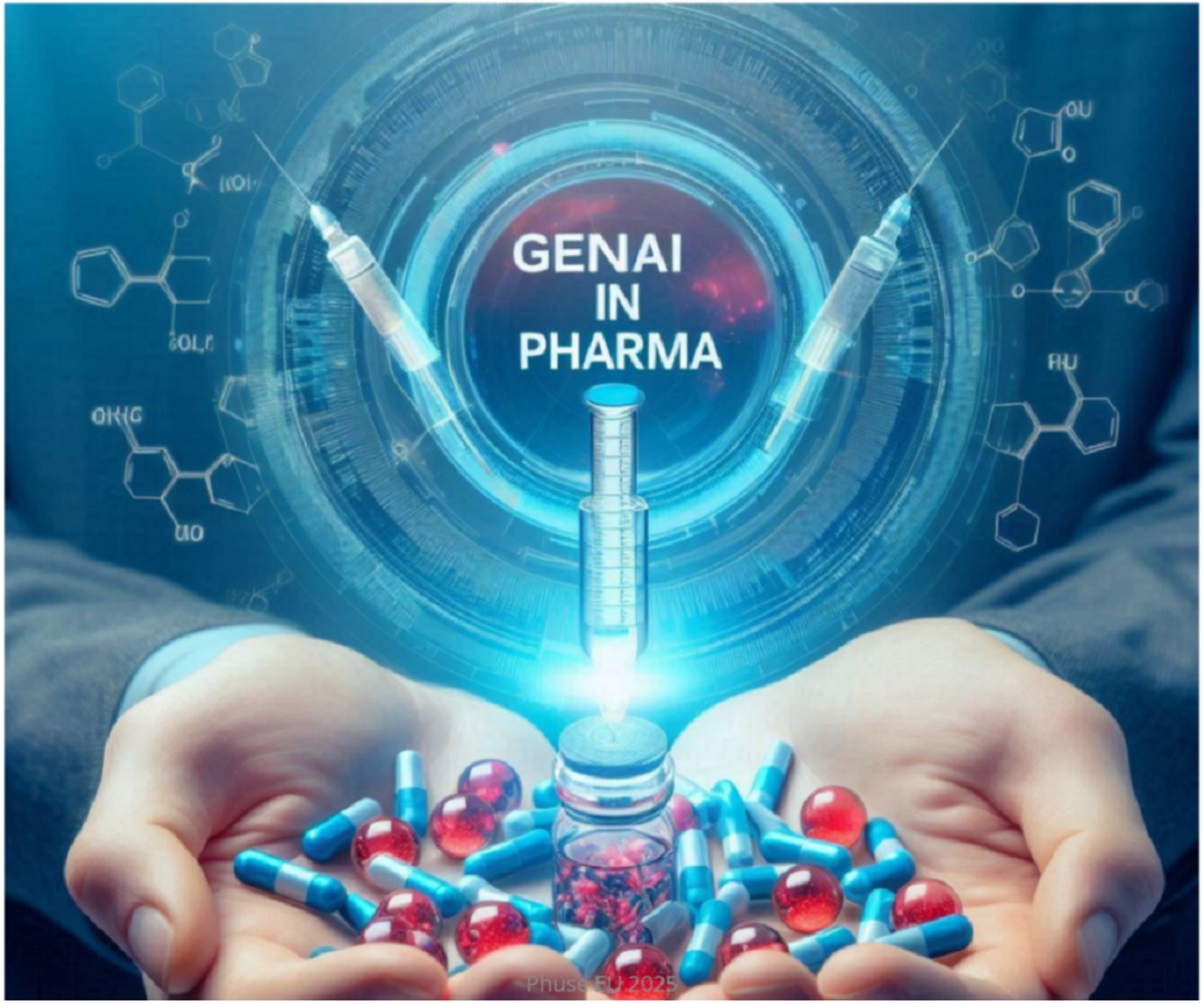
Open-Source Statistical Programming **Curriculum**!

Building Custom...

👩‍💼:

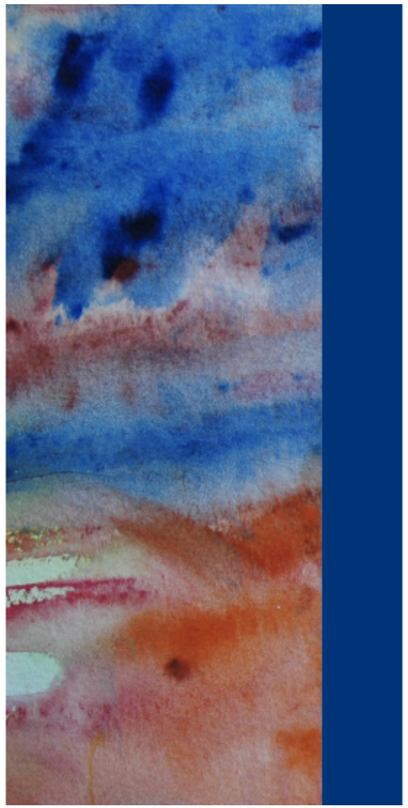
Open-Source Statistical Programming **Curriculum** with Community Resources!

GenAI in Pharma 2025



Open-Source Training at JnJ, GSK & Roche:

- Roch: Data Science University
- GSK: AccelerateR Team
- JnJ: SAIL.R



Rhonda Fenwick, *Time is Now I*
Through her art, Rhonda has explored psoriasis, a chronic skin disorder she has lived with since the age of six.

Using RStudio.Cloud to advance R proficiency: a crowdsourcing training experience

R/Pharma 2019

Paulo R. Bargo, Rudradev Sengupta, Bill Pikounis and Surya Mohanty
Statistics & Decision Sciences, Janssen R&D

janssen | PHARMACEUTICAL COMPANIES OF Johnson & Johnson

Clinical Reporting Data

Clinical Reporting Data: Internal, Sample & Synthetic

- pharmaverseraw
- pharmaversesdtm
- pharmaverseadam
- random.cdisc.data

 pharmaverse.github.io/examples/tlg/adverse_events.html 📊

Pharmaverse Data


tidyverse
tidyr
Shiny
rmarkdown
knitr
haven
gt
ggplot2
dplyr
tfrmt
rtables
NEST
pkglite
metacore
quarto
tidy tlg
%>%
Ceci n'est pas un pipe.
admiral
logrx
renv
purrr


admiral
GSK
Roche
admiralonco
Roche
GSK
AMGEN
Bristol Myers Squibb
admiralophtha
Roche
NOVARTIS
admiralvaccine
Pfizer
GSK
Johnson&Johnson
admiralpeds
Cytel
Roche
atorus
Pfizer
admiralmetabolic
novo nordisk
Roche
Boehringer Ingelheim

ADSL

Introduction

This guide will show you how four pharmaverse packages, along with some from tidyverse, can be used to create an ADaM such as `ADSL` end-to-end, using `{pharmaversesdtm}` SDTM data as input.

The four packages used with a brief description of their purpose are as follows:

- `{metacore}`: provides harmonized metadata/specifications object.
- `{metatools}`: uses the provided metadata to build/enhance and check the dataset.
- `{admiral}`: provides the ADaM derivations. (Find functions and related variables by searching admiraldiscovery)
- `{xportr}`: delivers the SAS transport file (XPT) and eSub checks.

It is important to understand `{metacore}` objects by reading through the above linked package site, as these are fundamental to being able to use `{metatools}` and `{xportr}`. Each company may need to build a specification reader to create these objects from their source standard specification templates.

>

```r
library(pharmaverseadam)
adae <- adae
```

<

Adverse Events

Introduction

This guide will show you how pharmaverse packages, along with some from tidyverse, can be used to create an Adverse Events table, using the `{pharmaverseadam}` `ADSL` and `ADAE` data as an input.

The packages used with a brief description of their purpose are as follows:

- `{rtables}`: designed to create and display complex tables with R.
- `{tern}`: contains analysis functions to create tables and graphs used for clinical trial reporting.

Load Data and Required pharmaverse Package

After installation of packages, the first step is to load our pharmaverse packages and input data. Here, we are going to encode missing entries in a data frame `adsl` ar

Note that `{tern}` depends on `{rtables}` so th

```
1 library(pharmaverseadam)
2 library(tern)
3 library(dplyr)
4
5 adsl <- adsl %>%
6   df_explicit_na()
7
8 adae <- adae %>%
9   df_explicit_na()
```

```r
library(dplyr)

# Create categorical variables, remove screen failures, and assign column labels
adsl <- pharmaverseadam::adsl |>
  filter(!ACTARM %in% "Screen Failure") |>
  mutate(
    SEX = case_match(SEX, "M" ~ "MALE", "F" ~ "FEMALE"),
    AGEGR1 =
      case_when(
        between(AGE, 18, 40) ~ "18-40",
        between(AGE, 41, 64) ~ "41-64",
        AGE > 64 ~ ">=65"
      ) |>
        factor(levels = c("18-40", "41-64", ">=65"))
  ) |>
  labelled::set_variable_labels(
    AGE = "Age (yr)",
    AGEGR1 = "Age group",
```

2. Environments

Ideas on Environments...

- posit.cloud
- Polyglot Statistical Computing Environment
- Shinylive

pharmaverse.github.io/examples/

pharmaverse examples

The true beauty of pharmaverse (and open source in general) is when efforts from various different developers come together to complement each other as a whole greater than the sum of the individual parts. By design in R, no single package will ever completely cover all your needs, but by piecing them together we can make complex tasks increasingly simple.

This book contains end-to-end examples of using pharmaverse packages together to achieve common clinical reporting analyses, such as SDTM, ADaM and Tables/Listings/Graphs. The examples use consistent source test raw datasets from `{pharmaverseraw}`, SDTMs and ADaMs from `{pharmaversesdtm}` and `{pharmaverseadam}` respectively.

We'll endeavour to include a selection of examples here over time, e.g. to help users when trying out the packages for PK/PD or Therapeutic Area specific (such as Oncology or Vaccines) analyses.

Note that this examples book should only be used to show how collections of packages can be used in conjunction - more thorough examples of individual package usages would always be covered in the package site vignettes and no need to repeat here.

Running the examples

Posit Cloud

Each example can be explored via a live and interactive Posit Cloud e
Posit Cloud" to try out any of the examples code. You can do this by c
free to try out customizing any of the examples to better fit any of yc

Locally

To run examples locally, download the repository and run the followi

```
if(!require(pak)) {
  install.packages("pak")
```


R Training Strategies at Janssen
Watch later
Share
R Training Strategies at Janssen
R consortium

 Tidyverse, Quarto/R Markdown & Quarto Curriculum 

Start at the Foundation...

- R for Data Science Book:
- Tidy Modeling with R Book:
- Mastering Shiny Book:
- Quarto:
- Lubridate:
- SAS Vs R in Pharma:
- CAMIS:

🤨 But We Need This for Pharma? 🤔

Pharma & Pharmaverse Curriculum

Fork Free Material

📖 pharmaverse.github.io/examples/

📄 github.com/posit-conf-2025/pharmaverse

📖 github.com/pfizer-opensource/pharma-hands-on-exercises

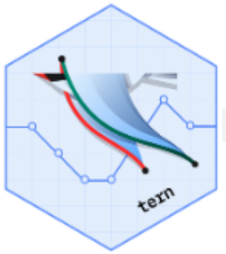
📄 youtube.com/c/RinPharma

phuse-sde-2024

Welcome in the PHUSE Single Day Event!

Packages used:

`teal` - Shiny-based interactive exploration framework for analyzing data



`tern` - Table, Listings, and Graphs (TLG) library for common outputs used in clinical trials

`rtables` - Reporting tables with R

`random.cdisc.data` - Create random CDISC data

Shiny for R

app.R

```
1  library(random.cdisc.data)
2  library(teal)
3
4  data <- teal_data()
5  data <- within(data, {
6    ADSL <- radsl(cached = TRUE)
7  })
8  # set datanames
9  datanames <- c("ADSL")
10 datanames(data) <- datanames
11
12 ## Reusable Configuration For Modules
13 ADSL <- data[["ADSL"]]
14
15 header <- tags$span(
16   style = "display: flex; align-items: center; justify-content: space-between; margin: 10px 0 10px 0;",
17   tags$span("Excercise 1", style = "font-size: 30px;"),
18 )
19
20 footer <- tags$p(
21   "This teal app is brought to you by PhUSE SDE Basel."
22 )
23
24 app <- init(
25   data = data,
26   modules = modules(
27     example_module()
28   ),
29   title = build_app_title("Basic Teal Demo App"),
30   header = header,
```

```
    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.
```

Excercise 1

Module (1)

Home / example teal module

```
# 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.

Session Info

Pid:42 Token:c6b98d19

Teal Gallery Home Demo

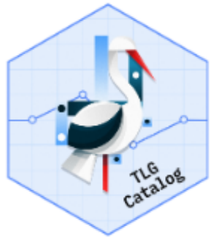
teal.gallery

A gallery of sample apps based on the `teal` framework.

Links to apps on `shinyapps.io`

The Stable version of the apps use the latest released packages while the dev version of the apps use the development packages which are installed from the `main` branch of the respective package repository. The specific versions used can be seen in the `Session Info` of the deployed app.

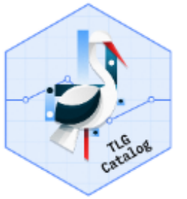
Stable version	Dev version
basic-teal	basic-teal
teal-as-shiny-module	teal-as-shiny-module
delayed-data	delayed-data
custom-transform	custom-transform
exploratory	exploratory
safety	safety
efficacy	efficacy
patient-profile	patient-profile
early-dev	early-dev
longitudinal	longitudinal
RNA-seq	RNA-seq



TLG Catalog

The TLG catalog is a catalog of **T**ables, **L**istings, and **G**raphs for clinical trials generated using NEST packages.

This repository provides a comprehensive collection of clinical trials outputs generated using the R language. The target audience is the clinical trials community, including statisticians, data scientists, and other professionals interested in applying R to clinical trials data.



Usage

Each TLG is represented on a separate article page, typically including the following sections:

- Setup and pre-processing of synthetic data.
- Steps to produce the TLG.
- The output TLG generated by the given code (including any available variants).
- An interactive application that can alternatively be used to produce and interact with the TLG.
- Reproducibility information.

See the full list of available TLGs on the Index page.

Interacting with Catalog R Code

The full source code of each article can be viewed by clicking on the “Source Code” button at the top of the page and copied using the “Copy to Clipboard”

4. Shiny Assistant

Shiny Assistant...

- Ai-powered tool that helps users create Shiny apps
- Users can create new Shiny applications or ask questions and iterate on existing applications
- Create Shiny apps in both R and in Python
- Chat to ask the assistant to fix mistakes
- Uses the Shinylive web interface (WebAssembly)
- https://posit.co/blog/ai-powered-shiny-app-prototyping/

gallery.shinyapps.io/assistant/

R Python Concise

Hello, I'm Shiny Assistant! I'm here to help you with Shiny, a web framework for data driven apps. You can ask me questions about how to use Shiny, to explain how certain things work in Shiny, or even ask me to build a Shiny app for you.

Here are some examples:

- "How do I add a plot to an application?"
- "Create an app that shows a normal distribution."
- "Show me how make it so a table will update only after a button is clicked."
- Ask me, "Open the editor", then copy and paste your existing Shiny code into the editor, and then ask me to make changes to it.

Let's get started! 🚀

Privacy Notice

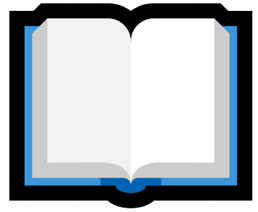
Enter a message...

But What About Git?

Git Learning...

- happygitwithr.com
- openstatsware.github.io/shortcourse-iscb2025/
- posit-conf-2025.github.io/reproducible-environments/#/title-slide
- phuse-org.github.io/devops/

What about AI?



Positron-Assistant

Positron-Databot

kapa.ai

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.

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

Ask AI

github.com/tidyverse/ggbot2 📖

>

 **pub.current.posit.team/public/ShinyLLMsinPharma/** 

<

Shiny & AI...

- Initialize a chat_client (e.g., ChatAnthropic()) to interact with the LLM using chatlas
- Initialize a Chat() instance
- Display its UI element with chat.ui()
- Decorate a @chat.on_user_submit function to fire when the user submits input
- Chat() makes it especially easy to use interfaces from OpenAI, Anthropic, Google, LangChain, and Ollama

Welcome | app.py U

chat-ai-anthropic > app.py > handle_user_input

```python
# ------------------------------------------------------------------------------------
# A basic Shiny Chat example powered by Anthropic's Claude model.
# ------------------------------------------------------------------------------------
import os

from app_utils import load_dotenv
from chatlas import ChatAnthropic

from shiny.express import ui

# ChatAnthropic() requires an API key from Anthropic.
# See the docs for more information on how to obtain one.
# https://posit-dev.github.io/chatlas/reference/ChatAnthropic.html
load_dotenv()
chat_client = ChatAnthropic(
    api_key=os.environ.get("ANTHROPIC_API_KEY"),
```

CONSOLE | TERMINAL | PROBLEMS | OUTPUT | PORTS | DEBUG CONSOLE

Python 3.12.6 (Global) ~/py-shiny-templates

```
Python 3.12.6 (main, Feb 25 2025, 22:33:32) [GCC 11.4.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.1.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: `?` alone on a line will brings up IPython's help
Python 3.12.6 (Global) was disconnected from the extension host.

Extension host is shutting down
Python 3.12.6 (Global) reconnected.

Python 3.12.6 (Global) shut down successfully.
Python 3.12.6 (Global) restarted.
Python 3.12.6 (main, Feb 25 2025, 22:33:32) [GCC 11.4.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.1.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: Run your doctests from within IPython for development and debugging. The special
%doctest_mode command toggles a mode where the prompt, output and exceptions display matches as
closely as possible that of the default Python interpreter.
>>>
```

SESSION | HELP | VIEWER | CONNECTIONS

https://dev.current.posit.team/s/c1a7a6ae9dc1b5e09ddbf/p/7cf34906/

Hello Anthropic Claude Chat

Hello! How can I help you today?

Create a Shiny for python application for analyzing adverse events and use ADAM data

Shiny for Python: Adverse Event Analysis Application

Below is a complete Shiny for Python application designed to analyze adverse events using ADAM (Analysis Data Model) data. This application allows users to upload ADAM datasets, particularly ADAE (Adverse Events Analysis Dataset), and provides interactive visualization and analysis capabilities.

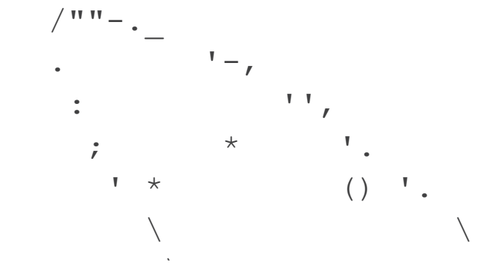
```python
from shiny import App, ui, render, reactive
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
from pathlib import Path
import io
```

Enter a message...

```r
library(shiny)
library(shinychat)

ui <- bslib::page_fluid(
  chat_ui("chat")
)

server <- function(input, output, session) {
  chat <- ellmer::chat_openai(system_prompt = "You're a trickster who answers in riddles")

  observeEvent(input$chat_user_input, {
    stream <- chat$stream_async(input$chat_user_input)
    chat_append("chat", stream)
  })
}

shinyApp(ui, server)
```

Done!

```
 ____________________________________________________
/Q: What do sharks order at McDonald’s? A: A quarter\
\ flounder with cheese.                              /
 ----------------------------------------------------
          \
           \
```



Thank you!

@philbowsher

youtube.com/rinpharma

rinpharma.com