

Fast, Fresh & Interactive: R Dashboards for Statisticians in 30 Minutes or Less!

Martin Brown

Statistical Science Director

PPD, Part of Thermo Fisher Scientific

PSI Conference 2026 Belfast

 The world leader in serving science



“What if the outputs you already create every day could become interactive in minutes — not days — using the tools you already have?”

No new platforms

No specialist development team

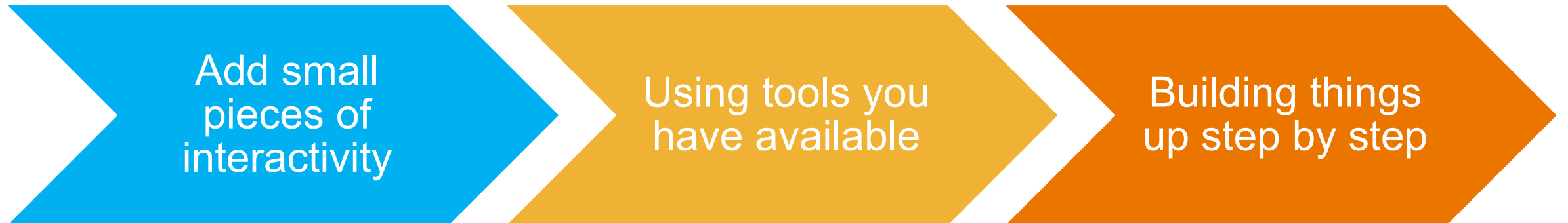
No complex setup

The Idea

Interactivity isn't always about complex dashboard application systems

Start small. Thinking about smaller use cases.

Build incrementally.



Why this Matters

Current Problems

- Static outputs limit exploration
- Rerunning analyses take time
- Small questions can lead to repeated work

Interactivity Benefits

- Interactivity enables exploration
- Reduces back-and-forth
- Reduces opening multiple files
- Improves understanding and communication

This isn't just for complex projects!

What we'll build today – step by step

RECIPES

01



Interactive figure

02



Interactive table

03



Shareable html dashboard or report

04



Shiny applications made simple

05



Shiny rapid prototyping / updating

RECIPE BOOKS



Teal

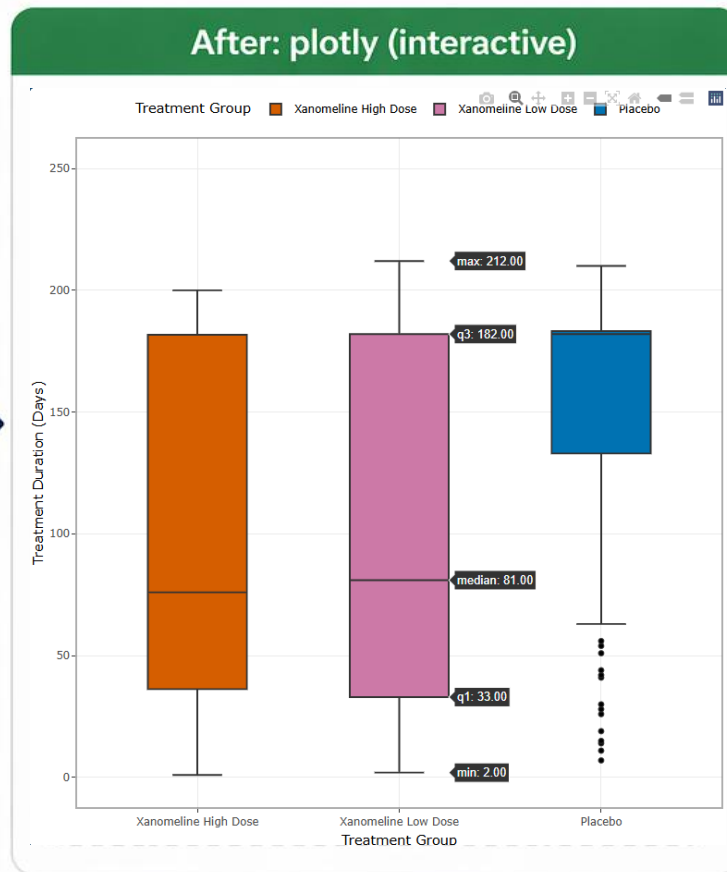
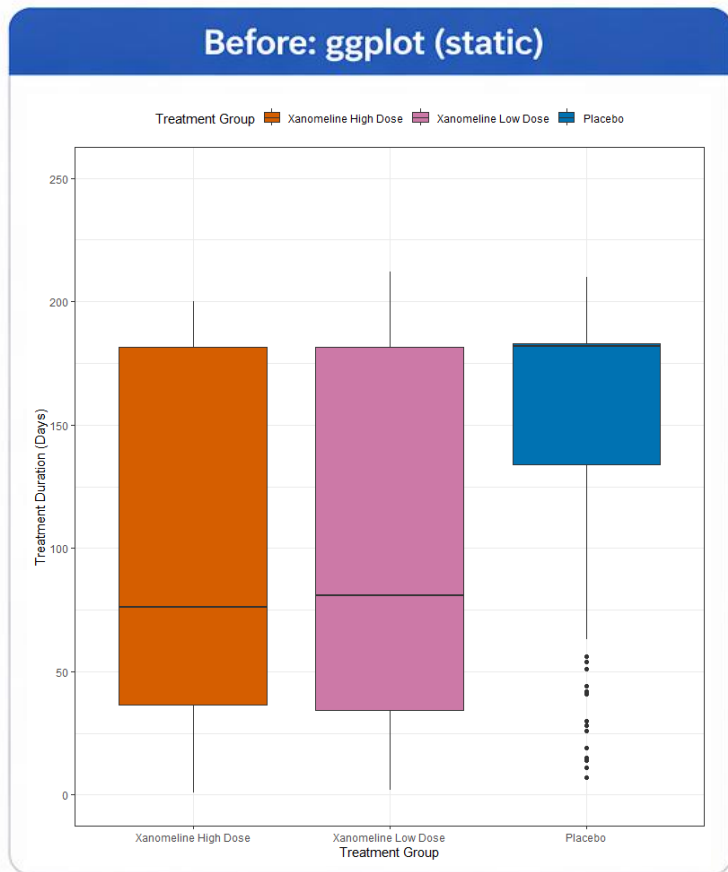


AI coding tools



RECIPE 1 Make a figure interactive

Take a static figure you have already created and make it interactive in seconds!



What you gain



Hover for details

See exact values and metadata on demand



Zoom & pan

Focus in, zoom out, and explore freely



Same analysis.
More insight.



How easy is it?

Often as simple as one line of code:

```
ggplotly(f1)
```

(Works with different ggplot2 plot types)



Learn more about Plotly

R interface to interactive web graphics

<https://plotly.com/ggplot2/>

RECIPE 2 Make a table explorable

Take a static dataset you have already created and make it interactive in seconds!

Before: dataset / table (static)

```
# A tibble: 252 x 5
  `Unique Subject Identifier` `Treatment Group` `Total Treatment Duration (Days)` `Baseline Weight (kg)` `Baseline Height (cm)`
  <chr> <chr> <dbl> <dbl> <dbl>
1 01-701-1015 Placebo 182 48 18
2 01-701-1023 Placebo 28 84 136
3 01-701-1028 Xanomeline High Dose 180 68 103
4 01-701-1033 Xanomeline Low Dose 14 63 31
5 01-701-1034 Xanomeline High Dose 183 75 47
# i 247 more rows
```



After: DT (interactive)

Show entries Search:

Unique Subject Identifier	Treatment Group	Total Treatment Duration (Days)	Baseline Weight (kg)	Baseline Height (cm)
<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>
01-701-1015	Placebo	182	48	18
01-701-1023	Placebo	28	84	136
01-701-1028	Xanomeline High Dose	180	68	103
01-701-1033	Xanomeline Low Dose	14	63	31
01-701-1034	Xanomeline High Dose	183	75	47

Showing 1 to 5 of 252 entries Previous 2 3 4 5 ... 51 Next

What you gain



Search & filter

Find what you need instantly



Sort & browse

Sort columns and browse with pagination



Same data.
More ways to explore.



How easy is it?

Often as simple as one line of code:

```
datatable(my_table)
```

(Works with any dataset)



Learn more about DT

R interface to the DataTables library
<https://rstudio.github.io/DT/>

RECIPE 3 Bring outputs together in one place

Combine figures and tables into a simple, interactive dashboard

After: one simple dashboard (Quarto Dashboard / flexdashboard)



What you gain



Everything in one place

Figures and tables together for a complete view



Explore and understand

Interact with the data to find insights



Easy to share

Open in any browser, no special software



Better experience.
More insight.
For everyone.



Straightforward using building blocks of previous recipes.

- Recipe 1: Build interactive figures with plotly
- Recipe 2: Create interactive tables with DT
- Recipe 3: Arrange them together in a dashboard layout



No complex setup.

Use Quarto
(or R Markdown for flexdashboard)



Learn more

Quarto dashboards (modern)
<https://quarto.org/docs/dashboards/>
flexdashboard (R Markdown alternative)
<https://pkgs.rstudio.com/flexdashboard/>

RECIPE 4 R Shiny and AI



From Quarto dashboard to Shiny app using a single prompt

Initial prompt



I have already created a Quarto dashboard using the code below.

Please convert this into a simple Shiny app that keeps the same data, plots, table, and overall purpose.

At the top of the script, include robust package setup code that:

- sets the CRAN repository to Posit Package Manager: <https://packagemanager.posit.co/cran/latest>
- installs required packages as binaries where possible using type = "binary"
- avoids reinstalling packages that are already installed
- does not load packages until all package checks are complete
- have library calls for all required packages after package install

Please reuse as much of the existing Quarto logic as possible, but restructure it into standard Shiny ui and server code.

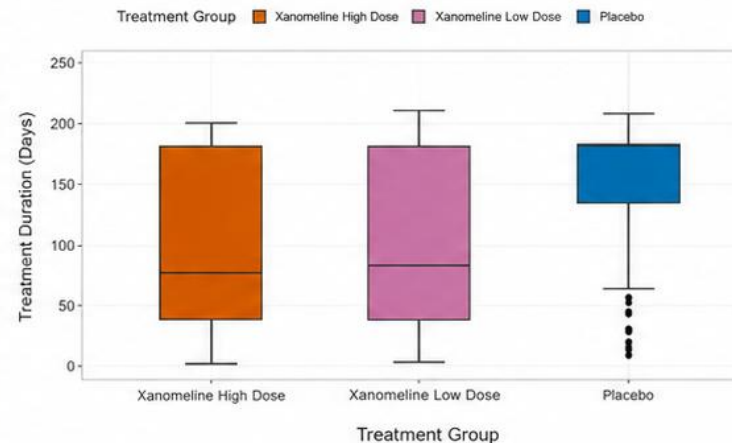
Here is the Quarto dashboard code:

<PASTE QUARTO DASHBOARD CODE FROM BEFORE HERE>

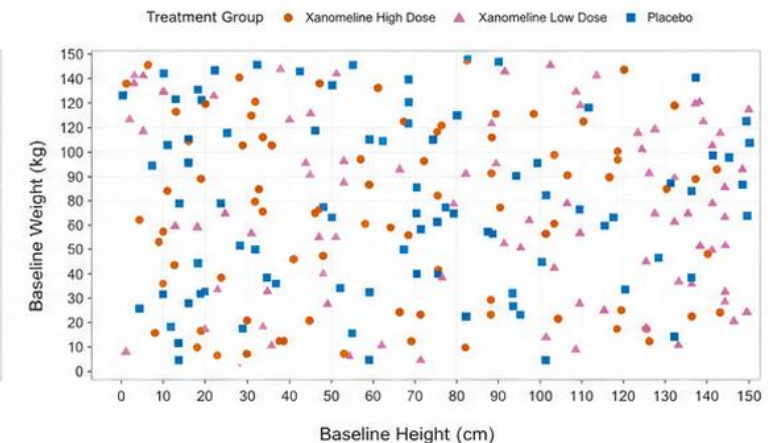
Generated Shiny app

Shiny Dashboard

Treatment Duration (Days) by Treatment Group



Baseline Weight (kg) vs Baseline Height (cm)



Subject-Level Data

Show 5 entries

Search:

Unique Subject Identifier	Treatment Group	Total Treatment Duration (Days)	Baseline Weight (kg)	Baseline Height (cm)
All	All	All	All	All
01-701-1015	Placebo	182	48	18
01-701-1023	Placebo	28	84	136
01-701-1028	Xanomeline High Dose	180	68	103
01-701-1033	Xanomeline Low Dose	14	63	31
01-701-1034	Xanomeline High Dose	183	75	47

Showing 1 to 5 of 252 entries

Previous 1 2 3 4 5 ... 51 Next

RECIPE 4 R Shiny and AI



Getting better AI prompts through prompts

Prompt

Give me a prompt to update this shiny app:

- giving the full code:
- just updating the first version of the code that was given previously
- Add a sidebar with
 - treatment group filter dropdown menu (multiple choice)
 - subject filter dropdown menu (multiple choice)
 - sliders for duration, weight, and height

Ensure all plots and the table respond to these filters



Update prompt

Please update the first Shiny app code you gave me.

Give me the full updated app.R code, not just a patch.

Keep the same data source, plots, table, colors, and overall layout/purpose, but add a sidebar with interactive filters:

- **Treatment Group** filter dropdown menu, allowing multiple selections
- **Subject** filter dropdown menu, allowing multiple selections
- **Slider** for Total Treatment Duration (Days)
- **Slider** for Baseline Weight (kg)
- **Slider** for Baseline Height (cm)

Requirements:

- The app should start with all treatment groups and all subjects selected.
- The slider ranges should be calculated from the data.
- All plots and the DT table must respond to all filters.
- Use a reactive filtered dataset in the server.
- Keep the existing robust package setup at the top:
 - set CRAN repo to <https://packagemanager.posit.co/cran/latest>
 - install missing required packages using type = "binary"
 - avoid reinstalling already installed packages
 - do not load packages until all package checks are complete
 - include library calls after package installation.
- Reuse as much of the existing Shiny app logic as possible.
- Return only the complete updated app.R code.

RECIPE 4 R Shiny and AI



Shiny Dashboard

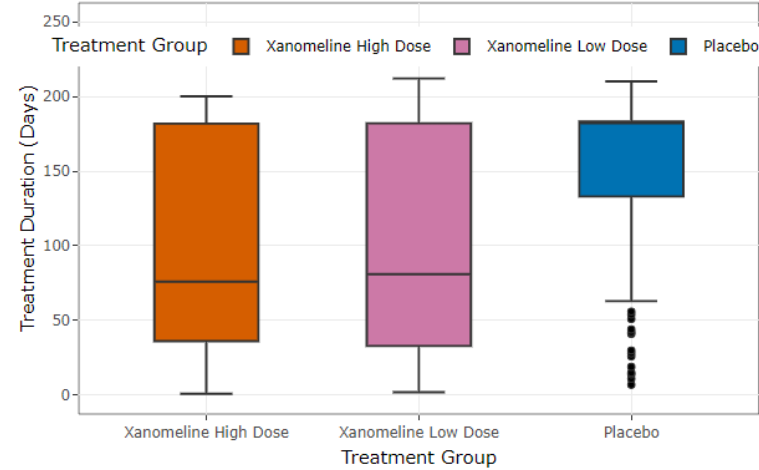
Treatment Group

Xanomeline High Dose Xanomeline Low Dose Placebo

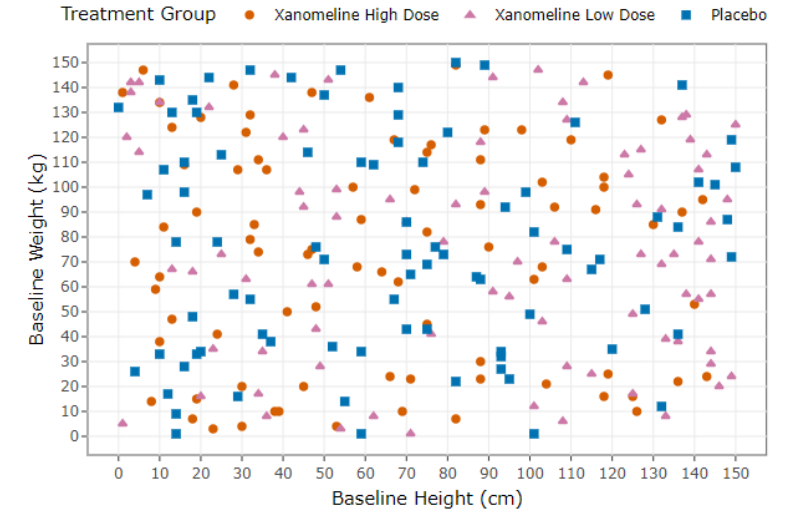
Subject

01-701-1015	01-701-1023	01-701-1028	01-701-1033	01-701-1034	
01-701-1047	01-701-1057	01-701-1097	01-701-1111	01-701-1115	01-701-1118
01-701-1130	01-701-1133	01-701-1145	01-701-1146	01-701-1148	01-701-1153
01-701-1162	01-701-1176	01-701-1180	01-701-1181	01-701-1188	01-701-1192
01-701-1203	01-701-1211	01-701-1234	01-701-1239	01-701-1240	
01-701-1275	01-701-1287	01-701-1294	01-701-1302	01-701-1307	
01-701-1317	01-701-1324	01-701-1341	01-701-1345	01-701-1356	
01-701-1360	01-701-1363	01-701-1369	01-701-1383	01-701-1386	
01-701-1387	01-701-1392	01-701-1411	01-701-1415	01-701-1429	
01-701-1440	01-701-1442	01-701-1444	01-702-1082	01-703-1042	
01-703-1076	01-703-1086	01-703-1096	01-703-1100	01-703-1119	
01-703-1175	01-703-1182	01-703-1197	01-703-1210	01-703-1258	
01-703-1279	01-703-1295	01-703-1299	01-703-1335	01-703-1379	
01-703-1396	01-703-1403	01-703-1439	01-704-1008	01-704-1009	
01-704-1010	01-704-1017	01-704-1025	01-704-1065	01-704-1074	
01-704-1093	01-704-1114	01-704-1120	01-704-1127	01-704-1135	01-704-1164
01-704-1218	01-704-1233	01-704-1241	01-704-1260	01-704-1266	
01-704-1323	01-704-1325	01-704-1332	01-704-1351	01-704-1388	
01-704-1435	01-704-1445	01-705-1011	01-705-1018	01-705-1031	
01-705-1058	01-705-1059	01-705-1112	01-705-1186	01-705-1199	
01-705-1243	01-705-1280	01-705-1281	01-705-1282	01-705-1292	
01-705-1303	01-705-1310	01-705-1349	01-705-1377	01-705-1382	
01-705-1393	01-705-1421	01-705-1431	01-706-1041	01-706-1049	
01-706-1384	01-707-1037	01-707-1206	01-707-1276	01-707-1430	
01-707-1434	01-708-1013	01-708-1019	01-708-1032	01-708-1054	
01-708-1067	01-708-1084	01-708-1087	01-708-1104	01-708-1158	
01-708-1171	01-708-1178	01-708-1184	01-708-1213	01-708-1216	
01-708-1236	01-708-1242	01-708-1253	01-708-1272	01-708-1286	
01-708-1296	01-708-1297	01-708-1316	01-708-1336	01-708-1342	
01-708-1347	01-708-1348	01-708-1352	01-708-1353	01-708-1372	
01-708-1378	01-708-1406	01-708-1428	01-709-1001	01-709-1007	

Treatment Duration (Days) by Treatment Group



Baseline Weight (kg) vs Baseline Height (cm)



Subject-Level Data

Show 5 entries

Search:

Unique Subject Identifier	Treatment Group	Total Treatment Duration (Days)	Baseline Weight (kg)	Baseline Height (cm)
All	All	All	All	All
01-701-1015	Placebo	182	48	18
01-701-1023	Placebo	28	84	136
01-701-1028	Xanomeline High Dose	180	68	103
01-701-1033	Xanomeline Low Dose	14	63	31
01-701-1034	Xanomeline High Dose	183	75	47

Showing 1 to 5 of 252 entries

Previous 1 2 3 4 5 ... 51 Next

RECIPE 4 R Shiny and AI

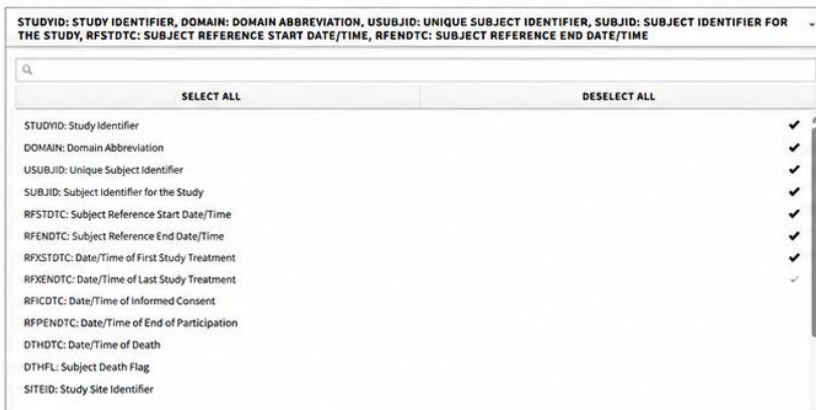


Getting better AI prompts through prompts

Prompt

Give me a prompt to update this shiny app:

- giving the full code:
- just updating the first version of the code that was given previously
- Change the treatment and subject filters to be like this image where we can still choose multiple values
- Add buttons in the sidebar for the user to be able to download static files



Update prompt

Please update the first Shiny app version you gave me previously.

Give me the full updated app.R code, not just a patch.

Keep the same data source, plots, table, colors, filters, downloads, and overall purpose.

Update only the first Shiny app code from earlier, incorporating the current sidebar filter and download-button version where appropriate.

Changes requested:

- Change the Treatment Group and Subject filters so they look and behave like the screenshot:
 - Use multi-select dropdowns that support selecting multiple values.
 - Show selected values in the collapsed input as comma-separated text.
 - Include a search box inside each dropdown.
 - Include Select All and Deselect All controls.
 - Show checkmarks for selected items.
 - Keep all treatment groups and all subjects selected by default.
- Add sidebar buttons that allow the user to download static files:
 - Download the Treatment Duration plot as a static image file.
 - Download the Baseline Weight vs Baseline Height plot as a static image file.
 - Download the filtered subject-level table as a CSV file.

Requirements:

- Keep the existing robust package setup at the top:
 - Set CRAN repository to <https://packagemanager.posit.co/cran/latest>
 - Install missing required packages using type = "binary"
 - Avoid reinstalling already installed packages
 - Do not load packages until all package checks are complete
 - Include library() calls after package installation
- Add any additional packages needed for the improved multi-select inputs or static downloads to the package setup.
- Use a reactive filtered dataset in the server.
- All plots and the DT table must respond to all filters.
- The static download buttons should download the currently filtered outputs.
- Reuse as much of the existing Shiny app logic as possible.
- Return only the complete updated app.R code.

(Include the image in the prompt both times)

RECIPE 4 R Shiny and AI



Quarto Dashboard

Treatment Group
Xanomeline High Dose, Xanomeline Low Dose, Placebo

Subject
01-701-1015, 01-701-1023, 01-701-1028, 01-701-1033, 01-701-1034, 01-701-1047, C

Total Treatment Duration (Days)
1 26 197 212

Baseline Weight (kg)
1 29 150

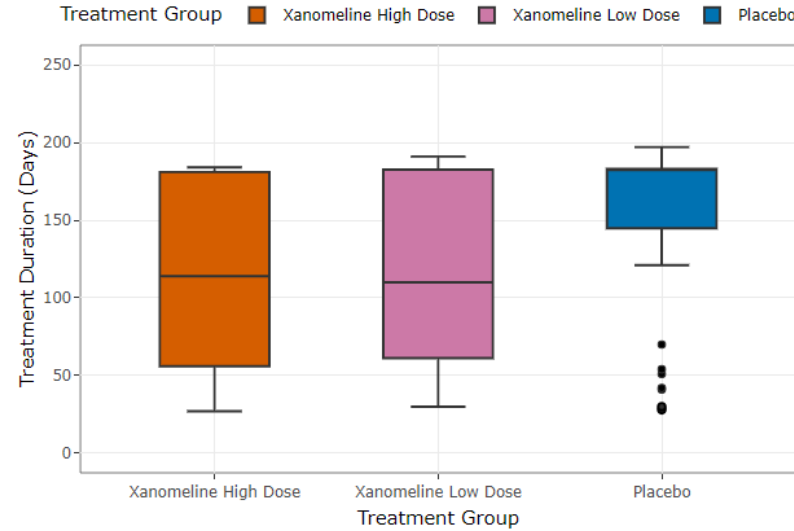
Baseline Height (cm)
0 57 150

Download Treatment Duration Plot

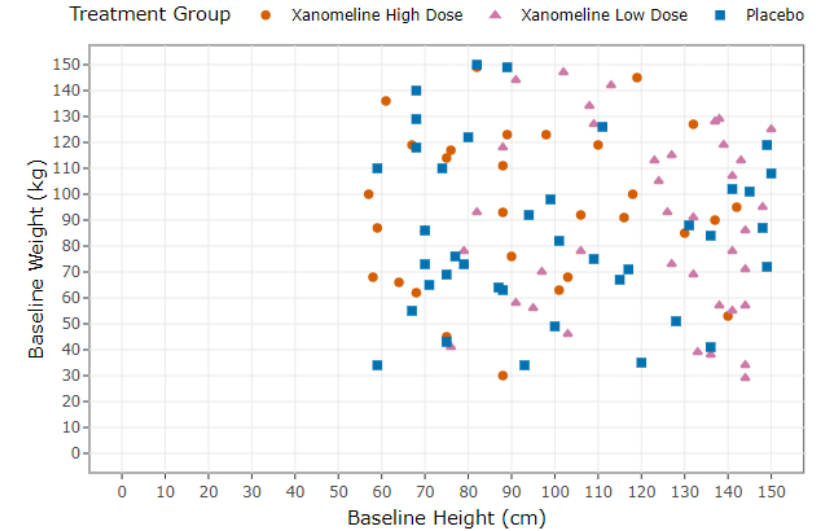
Download Height vs Weight Plot

Download Filtered Table CSV

Treatment Duration (Days) by Treatment Group



Baseline Weight (kg) vs Baseline Height (cm)



Subject-Level Data

Show 5 entries

Unique Subject Identifier	Treatment Group	Total Treatment Duration (Days)	Baseline Weight (kg)	Baseline Height (cm)
01-701-1023	Placebo	28	84	136
01-701-1028	Xanomeline High Dose	180	68	103
01-701-1115	Xanomeline Low Dose	55	105	124
01-701-1130	Placebo	183	86	70
01-701-1133	Xanomeline High Dose	183	117	76

Showing 1 to 5 of 107 entries

RECIPE 4 R Shiny and AI



Getting better AI prompts through prompts

Prompt

Give me a prompt to update this shiny app:

- giving the full code:
- just updating the first version of the code that was given previously
- Give the dashboard a meaningful name

Improve the look



Update prompt

Please update the first Shiny app version you gave me previously.

Give me the full updated app.R code, not just a patch.

Keep the same data source, plots, table, colors, filters, downloads, and overall purpose.

Update only the first Shiny app code from earlier, incorporating the current sidebar filter and download-button version where appropriate.

Changes requested:

- Give the dashboard a meaningful clinical-style name instead of "Quarto Dashboard".
- Improve the visual design of the app while keeping it simple and professional.
- Use a cleaner layout with a clear title/header area.
- Improve spacing, alignment, and section headings.
- Make the sidebar visually cleaner and easier to scan.
- Make the plot and table areas look more polished, for example by using card-style containers or similar Shiny/bslib layout elements.
- Keep the app appropriate for a clinical subject-level exploratory dashboard.

Requirements:

- Keep the robust package setup at the top.
- Reuse as much of the existing Shiny app logic as possible.
- All filters, plots, table, and download buttons must continue to work.
- Return only the complete updated app.R code.

RECIPE 4 R Shiny and AI



Clinical Subject-Level Explorer

Dashboard Controls

Filter the subject-level dataset and download the current filtered outputs.

Subject Selection

Treatment Group

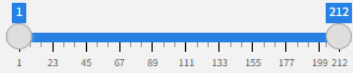
Xanomeline High Dose, Xanomeline Lc

Subject

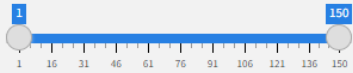
01-701-1015, 01-701-1023, 01-701-102

Clinical Measures

Total Treatment Duration (Days)



Baseline Weight (kg)



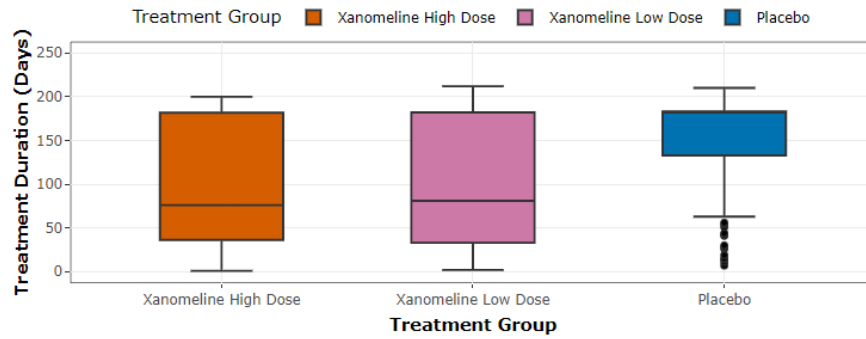
Baseline Height (cm)



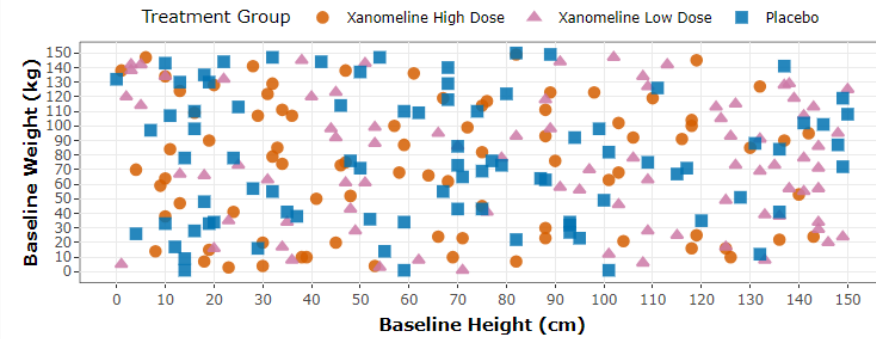
Clinical Subject-Level Explorer

Interactive review of treatment duration and baseline subject characteristics by treatment group.

Treatment Duration by Treatment Group



Baseline Weight vs Baseline Height



Filtered Subject-Level Data

Show 5 entries

Search:

Unique Subject Identifier	Treatment Group	Total Treatment Duration (Days)	Baseline Weight (kg)	Baseline Height (cm)
All	All	All	All	All
01-701-1015	Placebo	182	48	18
01-701-1023	Placebo	28	84	136
01-701-1028	Xanomeline High Dose	180	68	103

RECIPE 5 Live R Shiny Generation


Shiny Assistant (or R Studio with Posit AI or Positron)



Introduction
[Shiny - Shiny Assistant](#)

Online program
[Shiny Assistant](#)




 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... 

Enter any of the previous prompts

Or a new prompt

Create a polished R Shiny dashboard for exploring simulated clinical trial survival data.

Include:

- treatment group filter
- subgroup filter
- censoring rate slider

Display:

- Kaplan-Meier plot
- number at risk table
- hazard ratio summary
- downloadable survival dataset

Use a modern dashboard layout with reactive updates.
Keep the code concise and runnable as a single app.R file.

RECIPE 5 Live R Shiny Generation



Shiny Assistant (or R Studio with Posit AI or Positron)

The screenshot displays the Shiny Assistant interface. On the left, a chat window shows a user request: "I'll create a polished R Shiny dashboard for exploring clinical trial survival data with all the requested features." The assistant's response is a complete R script for a "Clinical Trial Survival Analysis Dashboard".

The central panel shows a preview of the dashboard. It includes a "Treatment Group" dropdown set to "All", a "Subgroup" dropdown set to "All", and a "Censoring Rate" slider set to 0.3. A "Download Dataset" button is visible. The dashboard contains three main components:

- Kaplan-Meier Survival Curve:** A plot showing survival probability over 150 months for Treatment A (blue) and Treatment B (yellow). The p-value is 0.00012.
- Number at Risk Table:** A table showing the number of subjects remaining in each treatment group over time.
- Hazard Ratio Summary:** A summary box showing HR: 0.63, 95% CI: [0.49, 0.80], and P-value: 0.0001.

At the bottom left, a chat input field is labeled "app.R".

Can I have the dataset on a second tab?

RECIPE 5 Live R Shiny Generation



Shiny Assistant (or R Studio with Posit AI or Positron)

Clinical Trial Survival Analysis Dashboard Analysis Dataset

Treatment Group:

Subgroup:

Censoring Rate:

[Download Dataset](#)

Survival Data Table

Search: _____

id	treatment	subgroup	age	sex	time	event
1	Treatment B	Subgroup 1	58	Male	7.8	1
2	Treatment B	Subgroup 2	65	Male	102.5	1
3	Treatment B	Subgroup 3	70	Male	28.2	1
4	Treatment A	Subgroup 2	56	Male	37.7	1
5	Treatment B	Subgroup 1	54	Male	34.6	1
6	Treatment B	Subgroup 3	42	Male	1.3	1
7	Treatment B	Subgroup 3	66	Male	59.8	0
8	Treatment B	Subgroup 2	50	Female	115	1
9	Treatment B	Subgroup 1	52	Female	0.3	0
10	Treatment B	Subgroup 3	45	Female	22.1	0
11	Treatment A	Subgroup 1	66	Male	13.5	1
12	Treatment B	Subgroup 2	44	Female	2.6	1
13	Treatment A	Subgroup 3	39	Male	35	1
14	Treatment A	Subgroup 3	56	Male	1.5	1
15	Treatment A	Subgroup 2	41	Female	6.5	1

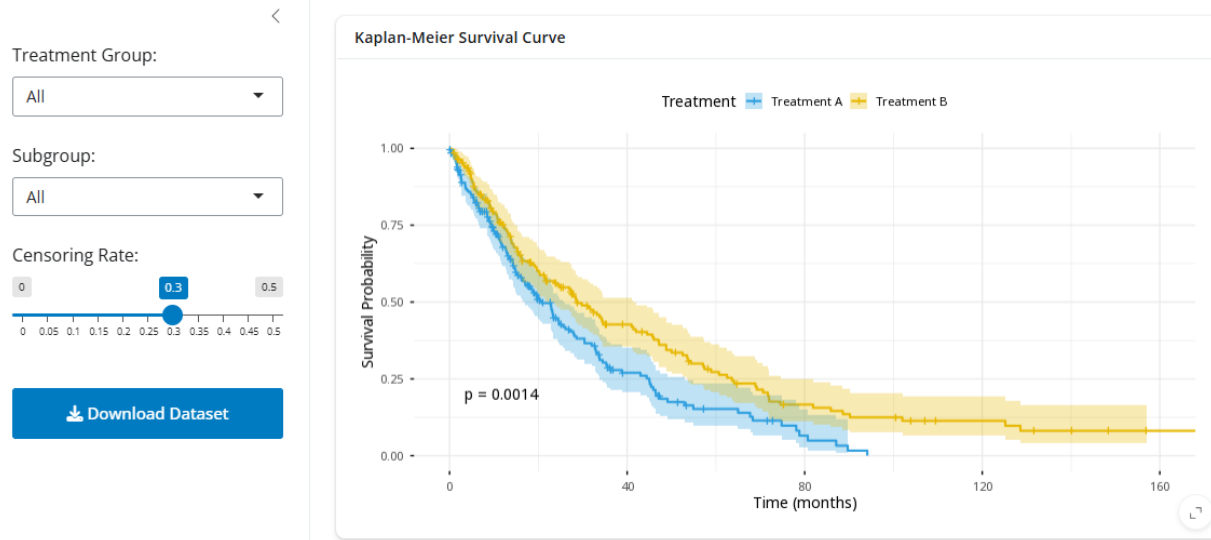
Can I have zoom in capabilities on each card?

RECIPE 5 Live R Shiny Generation

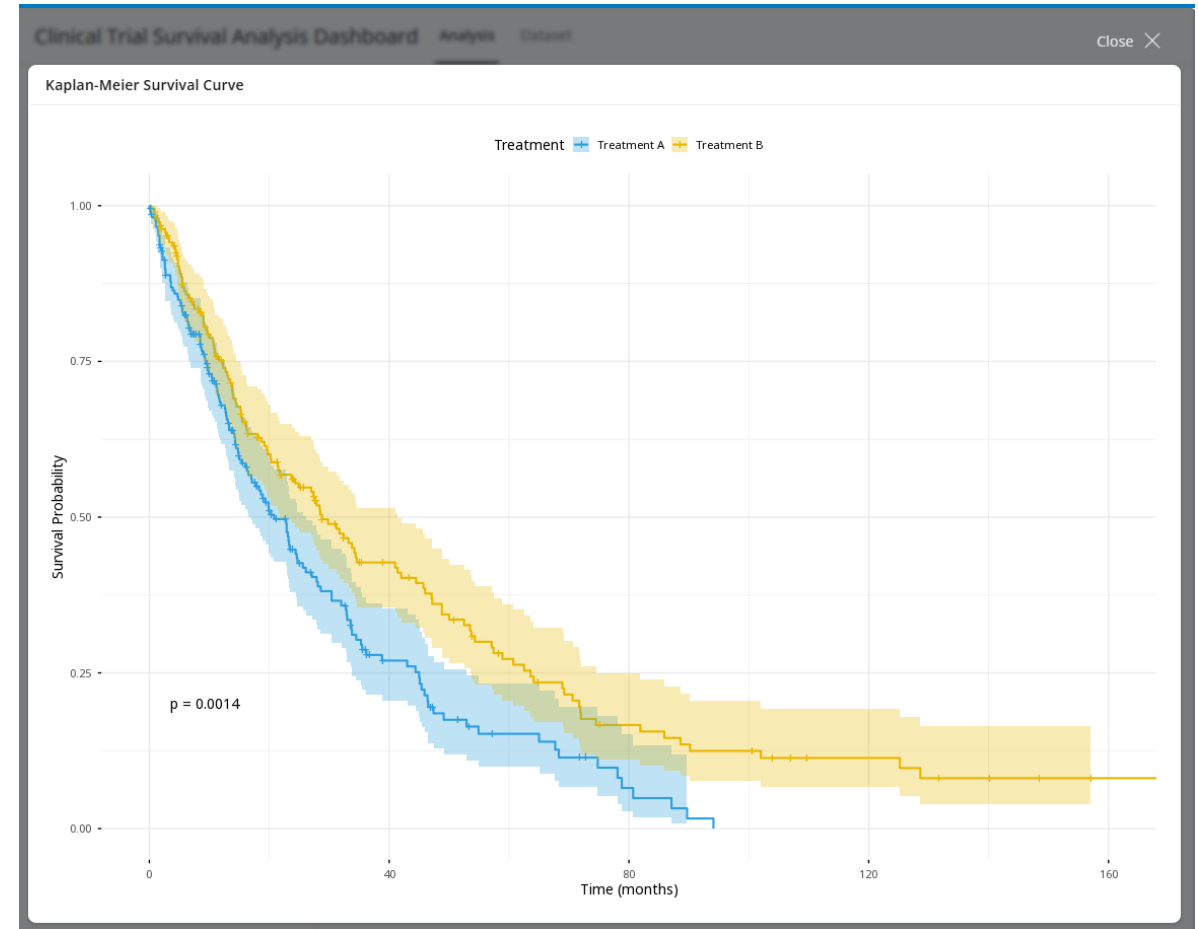


Shiny Assistant (or R Studio with Posit AI or Positron)

Clinical Trial Survival Analysis Dashboard Analysis Dataset



Add a checkbox called "Adjust Cox model for age and sex".



RECIPE 5 Live R Shiny Generation



Shiny Assistant (or R Studio with Posit AI or Positron)

Clinical Trial Survival Analysis Dashboard Analysis Dataset

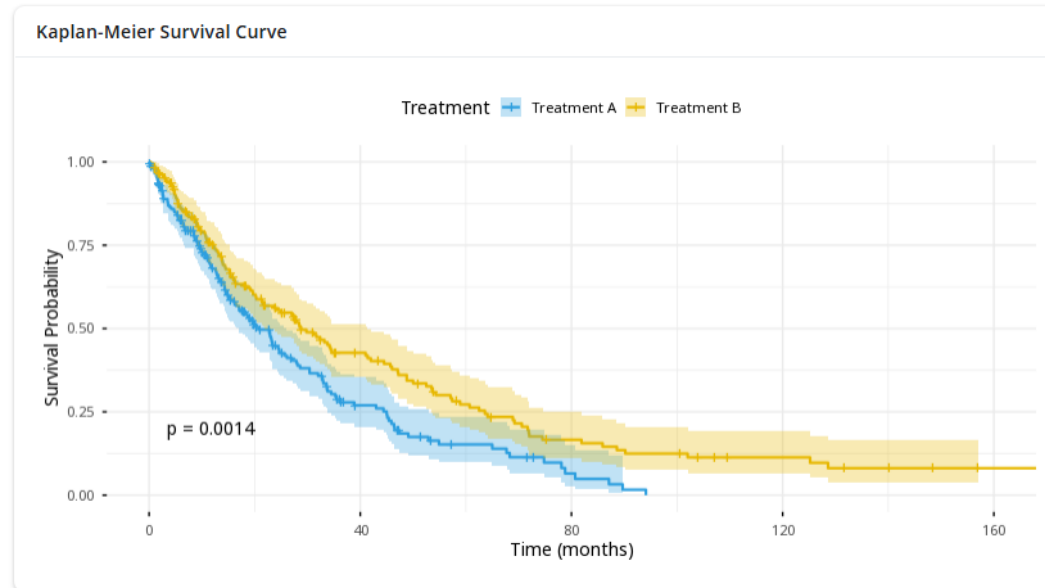
Treatment Group:

Subgroup:

Censoring Rate:

Adjust Cox model for age and sex

[Download Dataset](#)



Number at Risk Table

Number at risk

Strata	0	40	80	120	160
Treatment A	211	29	4	0	0
Treatment B	189	52	16	7	1

Hazard Ratio Summary

HR: 0.66

95% CI: [0.52, 0.85]

P-value: 0.0010

Adjusted for age and sex

Total subjects: 400
Events: 274 (68.5%)



RECIPE BOOK 1 - Teal



tea1 is a shiny-based interactive exploration framework

- pull in data (particular emphasis on CDISC)
- dynamic filtering
- reproducible code
- downloadable content and reports

Allows for modules that are prebuilt and custom

Biostatistics SDTM Dashboard  

Module (20) Report

Home / Data Table

Show only distinct rows:

DM AE CM DS EG EX LB MH QS VS

Select variables:

STUDYID: STUDY IDENTIFIER, DOMAIN: DOMAIN ABBREVIATION, USUBJID: UNIQUE SUBJECT IDENTIFIER, SUBJID: SUBJECT IDENTIFIER FOR THE STUDY, RFSTDTM: SUBJECT REFERENCE START DATE/TIME, RFENDTC: SUBJECT REFERENCE END DATE/TIME

Show 30 entries

	STUDYID	DOMAIN	USUBJID	SUBJID	RFSTDTM	RFENDTC
1	CDISCPLOT01	DM	01-701-1015	1015	2014-01-02	2014-07-02
2	CDISCPLOT01	DM	01-710-1235	1235	2012-09-26	2013-03-27
3	CDISCPLOT01	DM	01-710-1187	1187	2012-11-10	2013-05-12
4	CDISCPLOT01	DM	01-710-1183	1183	2013-11-16	2014-05-17
5	CDISCPLOT01	DM	01-710-1166	1166	2012-11-30	2013-03-30
6	CDISCPLOT01	DM	01-710-1154	1154	2014-03-29	2014-05-09
7	CDISCPLOT01	DM	01-710-1149	1149	<Missing>	<Missing>
8	CDISCPLOT01	DM	01-710-1142	1142	2012-10-02	2012-10-23
9	CDISCPLOT01	DM	01-710-1137	1137	2013-10-11	2013-11-13
10	CDISCPLOT01	DM	01-710-1129	1129	<Missing>	<Missing>
11	CDISCPLOT01	DM	01-710-1083	1083	2013-07-22	2013-08-03
12	CDISCPLOT01	DM	01-710-1078	1078	2013-09-17	2014-03-23
13	CDISCPLOT01	DM	01-710-1077	1077	2013-11-17	2014-05-18
14	CDISCPLOT01	DM	01-710-1070	1070	2012-09-08	2013-02-23

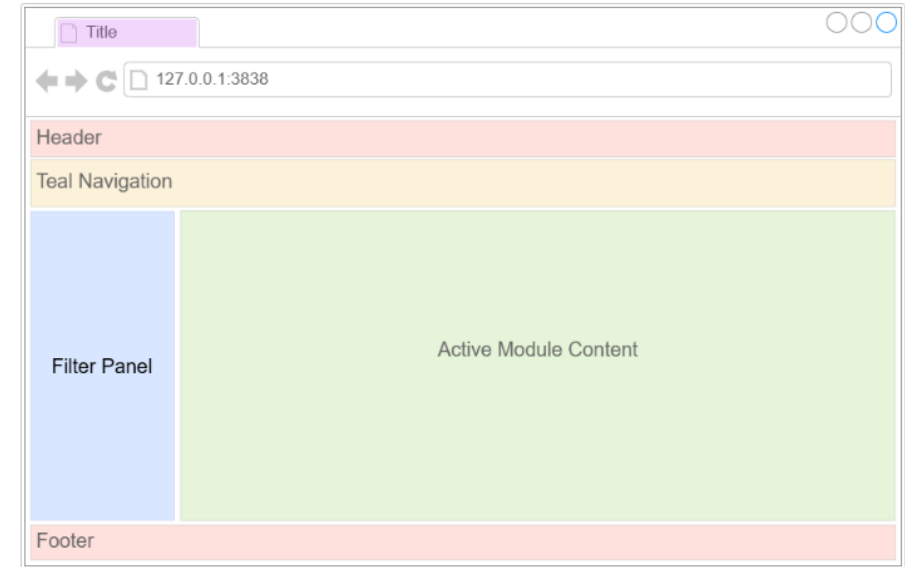
Filter Data

DM +

AE +

CM +

DS +



What tea1 gives you:

- Structure
- Standardisation
- Scalability
- Industry support



RECIPE BOOK 1 - Teal



ThermoFisher
SCIENTIFIC

Still can start off small

My first teal app

Module (1) ▾

Home / example teal module

Choose a dataset

IRIS ▾

Show R code

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa
20	5.1	3.8	1.5	0.3	setosa
21	5.4	3.4	1.7	0.2	setosa
22	5.1	3.7	1.5	0.4	setosa
23	4.6	3.6	1.0	0.2	setosa
24	5.1	3.3	1.7	0.5	setosa
25	4.8	3.4	1.9	0.2	setosa
26	5.0	3.0	1.6	0.2	setosa
27	5.0	3.4	1.6	0.4	setosa
28	5.2	3.5	1.5	0.2	setosa
29	5.2	3.4	1.4	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa
31	4.8	3.1	1.6	0.2	setosa

Module by module

Basic

Prebuilt

Custom

```
library(dplyr)
library(teal)

data <- teal_data(
  IRIS = iris,
  MTCARS = mtcars
)

app <- init(
  data = data,
  modules =example_module()) %>%
  modify_title("Basic Teal Demo App") %>%
  modify_header("My first teal app") %>%
  modify_footer("Powered by teal + shiny")
```

```
shinyApp(app$ui, app$server)
```

RECIPE BOOK 1 - Teal



Module by module



ThermoFisher
SCIENTIFIC

Still can start off small

Basic **Prebuilt** Custom

My first teal app

Module (1) Report

Home / Demographic Table

+ Add Report Card

Encodings
Dataset: ADSL
Select Column Variable(s)
Select
ARM Description of Plann

Add All Patients column

Summarize Variables
Select
AGE Age, SEX Sex, I

Additional table settings

Display NA counts
 if any
 no

Choose the statistics to display for numeric variables

- n
- Mean (SD)
- Mean 95% CI
- Geometric Mean
- Median
- Median 95% CI
- 25% and 75%-ile
- Min - Max

Description of Planned Arm	A: Drug X (N=134)	B: Placebo (N=134)	C: Combination (N=132)	All Patients (N=400)
Age				
n	134	134	132	400
Mean (SD)	33.8 (6.6)	35.4 (7.9)	35.4 (7.7)	34.9 (7.4)
Mean 95% CI	(32.65, 34.89)	(34.08, 36.78)	(34.10, 36.76)	(34.14, 35.61)
Geometric Mean	33.1	34.6	34.7	34.1
Median	33.0	35.0	35.0	34.0
Median 95% CI	(31.00, 35.00)	(32.00, 36.00)	(33.00, 36.00)	(33.00, 35.00)
25% and 75%-ile	28.0 - 39.0	30.0 - 40.0	30.0 - 40.0	29.0 - 39.0
Min - Max	21.0 - 50.0	21.0 - 62.0	20.0 - 69.0	20.0 - 69.0
Sex				
n	134	134	132	400
F	79 (59%)	82 (61.2%)	70 (53%)	231 (57.8%)
M	55 (41%)	52 (38.8%)	62 (47%)	169 (42.2%)
Race				
n	134	134	132	400
ASIAN	68 (50.7%)	67 (50%)	73 (55.3%)	208 (52%)
BLACK OR AFRICAN AMERICAN	31 (23.1%)	28 (20.9%)	32 (24.2%)	91 (22.8%)
WHITE	27 (20.1%)	26 (19.4%)	21 (15.9%)	74 (18.5%)
AMERICAN INDIAN OR ALASKA NATIVE	8 (6%)	11 (8.2%)	6 (4.5%)	25 (6.2%)

```
library(teal)
library(teal.widgets)
library(shiny)
library(DT)
library(teal.modules.clinical)

data <- teal_data(
  ADSL = rADSL
)

# Create join keys object
join_keys_obj <- join_keys(
  join_key("ADSL", keys = c("USUBJID"))
)

# Assign join keys
join_keys(data) <- join_keys_obj

# Define the Demographic Table Module
demo_table_module <- function() {
  tm_t_summary(
    label = "Demographic Table",
    dataname = "ADSL",
    arm_var = choices_selected(
      choices = choices_labeled(
        choices = "ARM",
        labels = col_labels(rADSL, fill = FALSE)["ARM"]
      ),
      selected = "ARM"
    ),
    summarize_vars = choices_selected(
      choices = variable_choices("ADSL"),
      selected = c("SEX", "AGE", "RACE")
    )
  )
}

app <- init(
  data = data,
  modules = modules(demo_table_module()) %>%

  modify_title("Basic Teal Demo App") %>%
  modify_header("My first teal app") %>%
  modify_footer("Powered by teal + shiny")

shinyApp(app$ui, app$server)
```



RECIPE BOOK 1 - Teal



Module by module



ThermoFisher
SCIENTIFIC

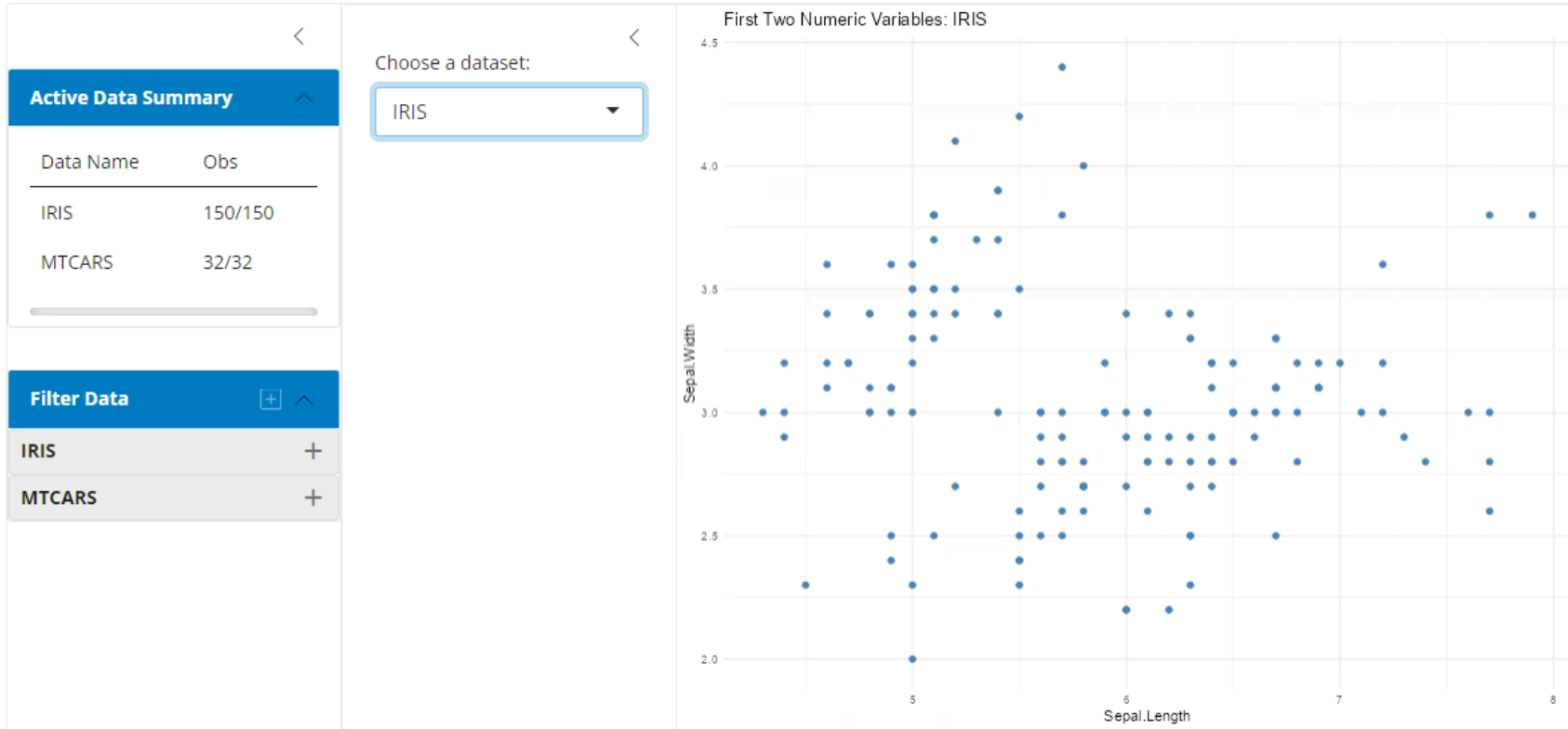
Still can start off small

Basic Prebuilt **Custom**

My first teal app

Module (1) ▾

Home / Plot



```
library(teal)
library(teal.widgets)
library(shiny)
library(ggplot2)

data <- teal_data(
  IRIS = iris,
  MTCARS = mtcars
)

plot_module <- function(label = "Plot") {
  module(
    label = label,

    ui = function(id) {
      ns <- NS(id)

      standard_layout(
        encoding = tagList(
          selectInput(
            ns("dataset"),
            "Choose a dataset:",
            choices = c("IRIS", "MTCARS")
          )
        ),
        output = plotoutput(ns("plot"), height = "80vh")
      ),
  )
}

server = function(id, data) {
  moduleServer(id, function(input, output, session) {

    output$plot <- renderPlot({

      df <- data()[[input$dataset]]

      # Keep numeric columns
      num_df <- df[, sapply(df, is.numeric), drop = FALSE]
      req(ncol(num_df) >= 2)

      x_name <- names(num_df)[1]
      y_name <- names(num_df)[2]

      ggplot(num_df, aes_string(x = x_name, y = y_name)) +
        geom_point(color = "steelblue", size = 2) +
        labs(
          title = paste("First Two Numeric Variables:", input$dataset),
          x = x_name,
          y = y_name
        ) +
        theme_minimal()
    })
  })
}

app <- init(
  data = data,
  modules = modules(plot_module()) %>%
  modify_title("Basic Teal Demo App") %>%
  modify_header("My first teal app") %>%
  modify_footer("Powered by teal + shiny")

shinyApp(app$ui, app$server)
```

Recipe Book 2 – AI Coding Tools

Codex, Claude code, RStudio with Posit AI, Positron

- Updating multiple scripts – no need to open each one individually
- Links to version control systems – e.g. github
- Inexpensive



Key Takeaways

Start simply if you need to

Build confidence incrementally

Plan – start small – build

Great tools are available to you (plotly, DT, rmarkdown, quarto, shiny, AI tools)

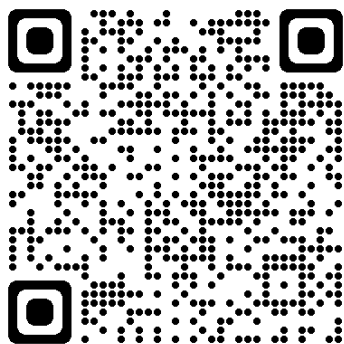
Use AI to accelerate

Interactive applications are becoming part of modern statistical workflows

But that doesn't have to mean huge applications / dashboards
– create smaller useful interactive tools!

AIMS and Open-source Lunch Bites

When it comes to R and Open-Source in the Pharma industry, we would like to point you in the right direction!



AIMS SIG - Open-Source Lunch Bites


Every first Friday of the month 12:00-12:30 GMT Online via Zoom

A series of 30-minute monthly webinar lunch and learns designed to provide practical, bite-sized learning experiences.

Includes a 20-minute presentation/demonstration followed by a 10-minute Q&A, focusing on Open-Source skills relevant to the pharmaceutical industry.

Join us each month to enhance your Open-Source knowledge in a convenient, pocket-sized format.

This is free for PSI members and non-members.



AIMS SIG

