A New Era for R Programming: How AI is Changing the Way We Work

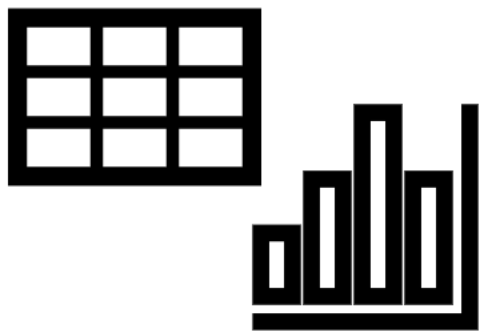
Jiaqi Song
Clinical & Statistical Programming, IDAR
2025.3.28
Shanghai

J&J Innovative Medicine

Disclaimer

- To avoid conflicts of interest, this presentation does not specify which AI tools are used when discussing AI.
- Some answers in this presentation are generated by AI and do not represent personal views. Please cite with caution.
- All images in this presentation are AI-generated and do not have copyright issues.
- The data analysis in this presentation is based on dummy data, with no real data involved. The results are for illustrative purposes only and have no practical significance.
- This presentation is for informational purposes only and does not represent professional guidance or advice. Any views and opinions expressed during this presentation are those of the presenters and do not necessarily reflect the views or policies of Johnson & Johnson.

R + AI

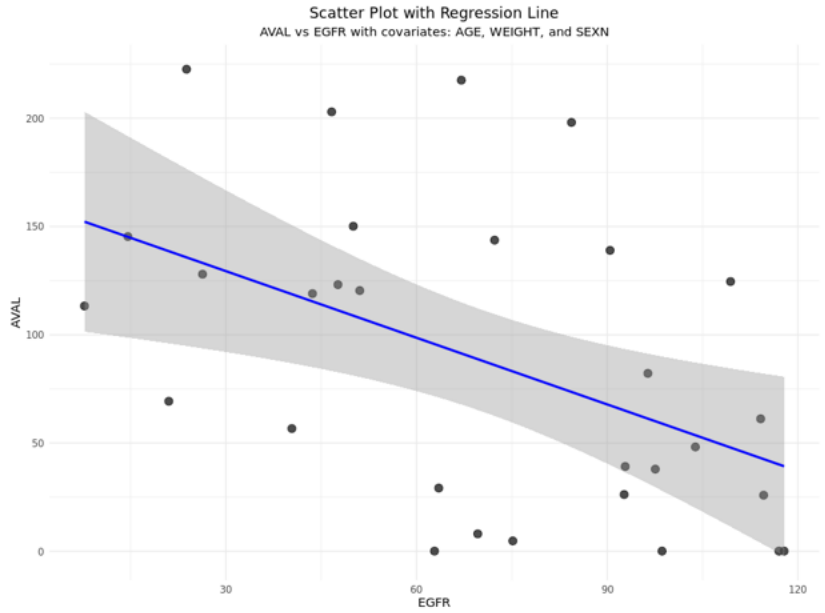


R for TLG

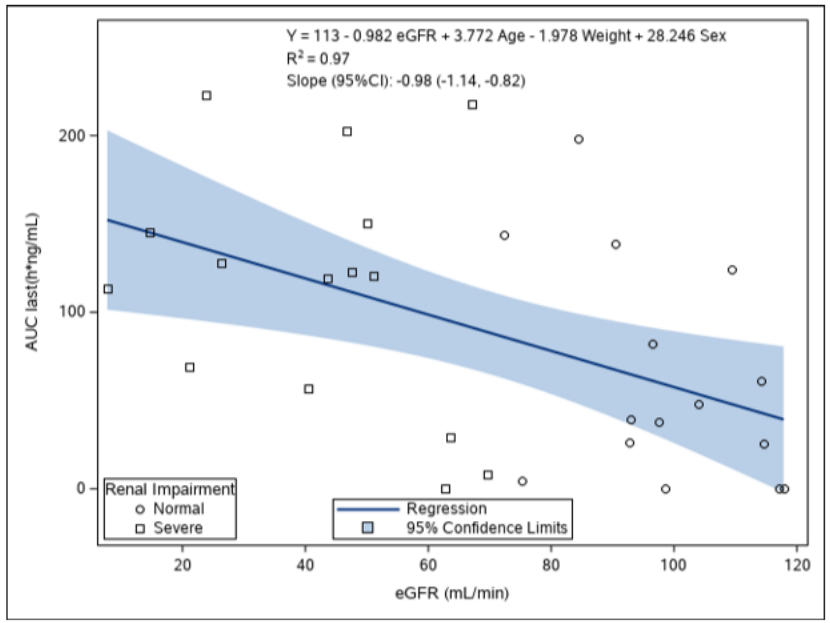
R Shiny

R Learning

QC by R

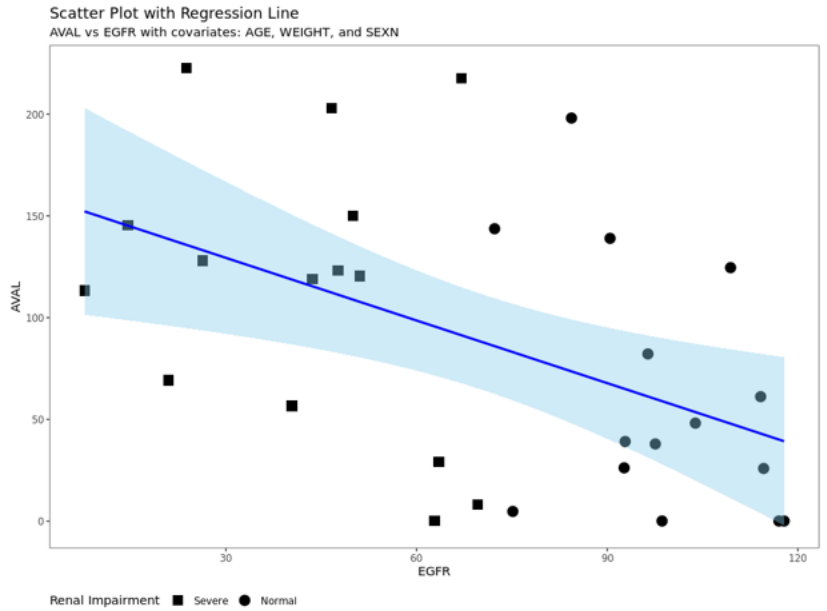


Production by SAS

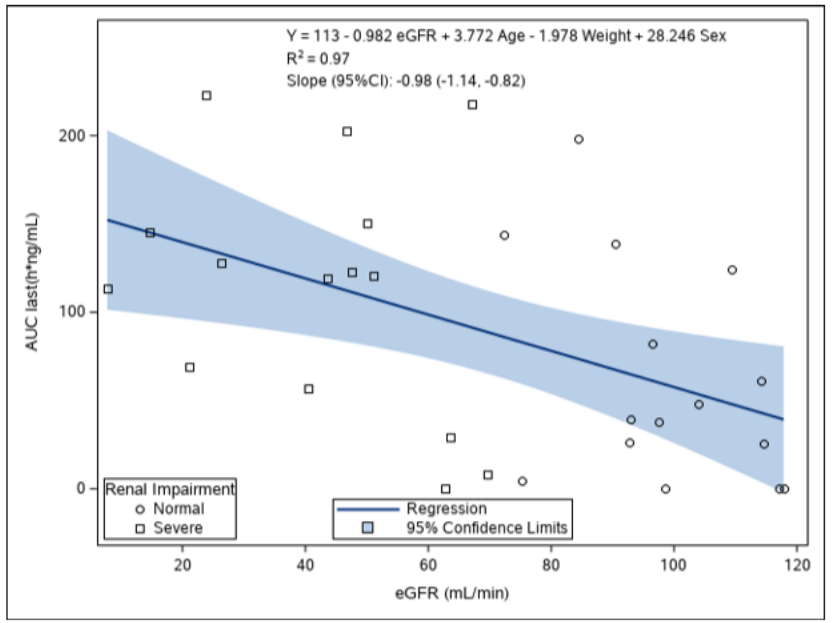



```
dataframe pkdata contains the following variables:
'AVAL'(type:numeric);'AGE'(type:numeric);'WEIGHT'(type:numeric)
'SEXN'(type:numeric);'RFGROUP'(type:character);'EGFRN'(type:numeric)
Create a R graph based on pkdata, y var is AVAL, x var is EGFRN.
Add scatter plot and a regression line with 95%CI band.
```

QC by R



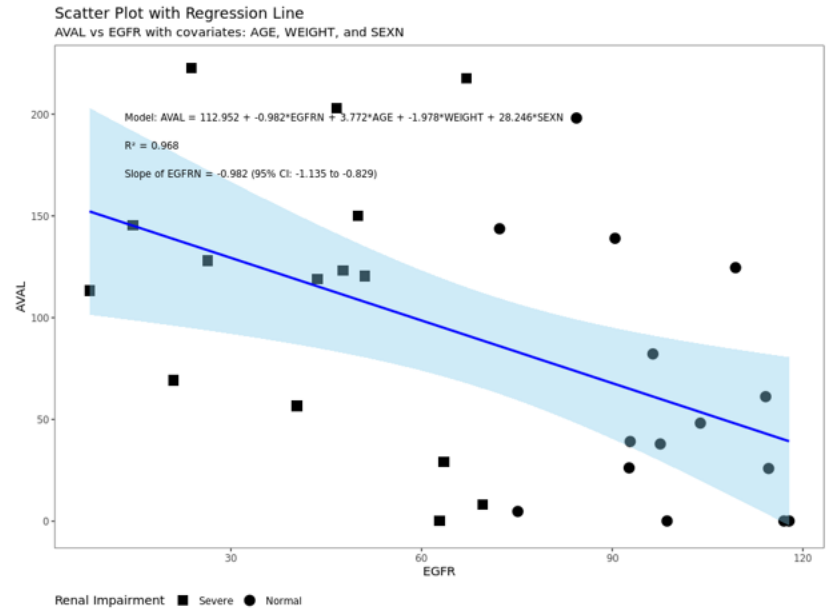
Production by SAS



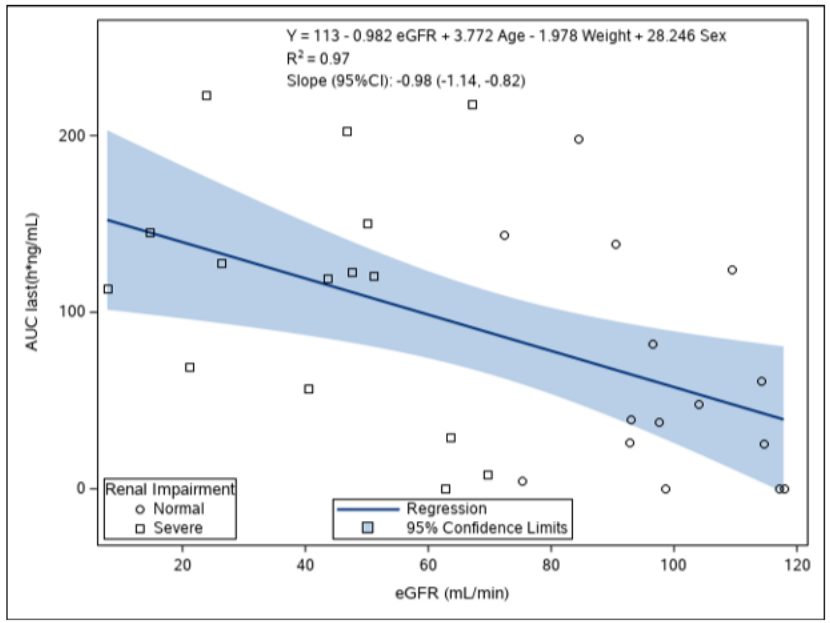
dataframe pkdata contains the following variables:
'AVAL'(type:numeric);'AGE'(type:numeric);'WEIGHT'(type:numeric)
'SEXN'(type:numeric);'RFGROUP'(type:character);'EGFRN'(type:numeric)
Create a R graph based on pkdata, y var is AVAL, x var is EGFRN.
Add scatter plot and a regression line with 95%CI band.

Group scatter shape by RFGROUP, 'Severe' to square, 'Normal' to circle,
set the legend title to 'Renal Impairment', put legend at bottom.
Change the band color to skyblue, theme to theme_test().

QC by R



Production by SAS

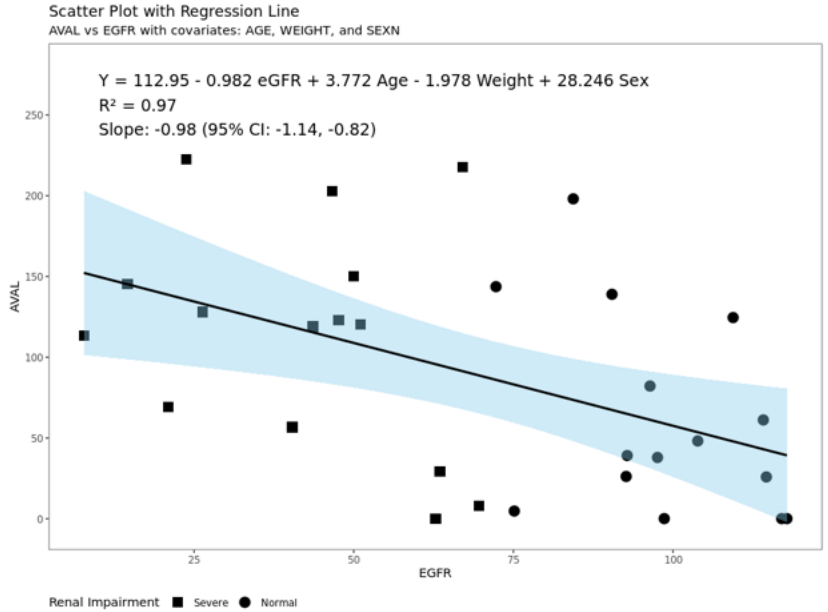


dataframe pkdata contains the following variables:
'AVAL'(type:numeric);'AGE'(type:numeric);'WEIGHT'(type:numeric)
'SEXN'(type:numeric);'RFGROUP'(type:character);'EGFRN'(type:numeric)
Create a R graph based on pkdata, y var is AVAL, x var is EGFRN.
Add scatter plot and a regression line with 95%CI band.

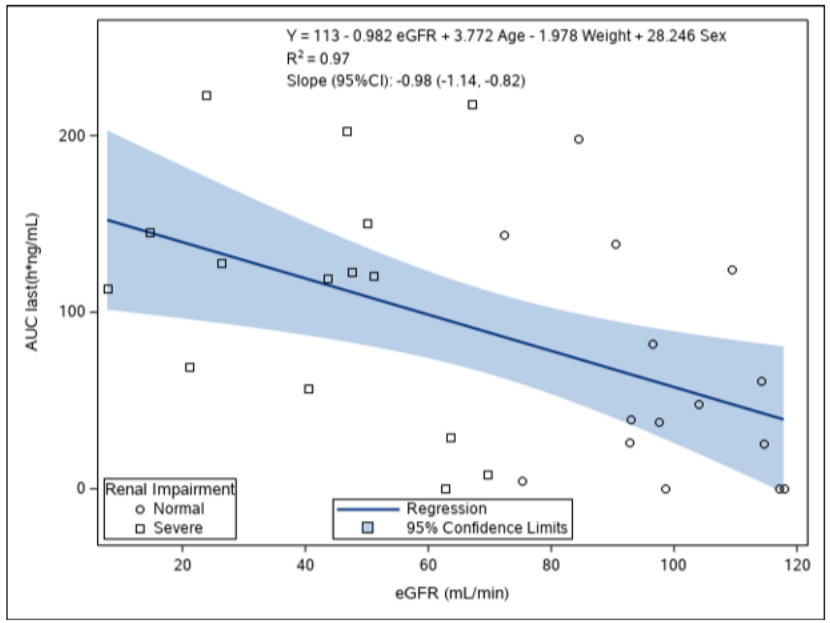
Group scatter shape by RFGROUP, 'Severe' to square, 'Normal' to circle, set the legend title to 'Renal Impairment', put legend at bottom.
Change the band color to skyblue, theme to theme_test().

Add annotation to the graph include:
formulation of the regression model AVAL as Y, EGFRN, AGE, WEIGHT and SEXN as covariates, with the coefficients,
R square of the regression model,
Slope of EGFRN and 95%CI.

QC by R



Production by SAS



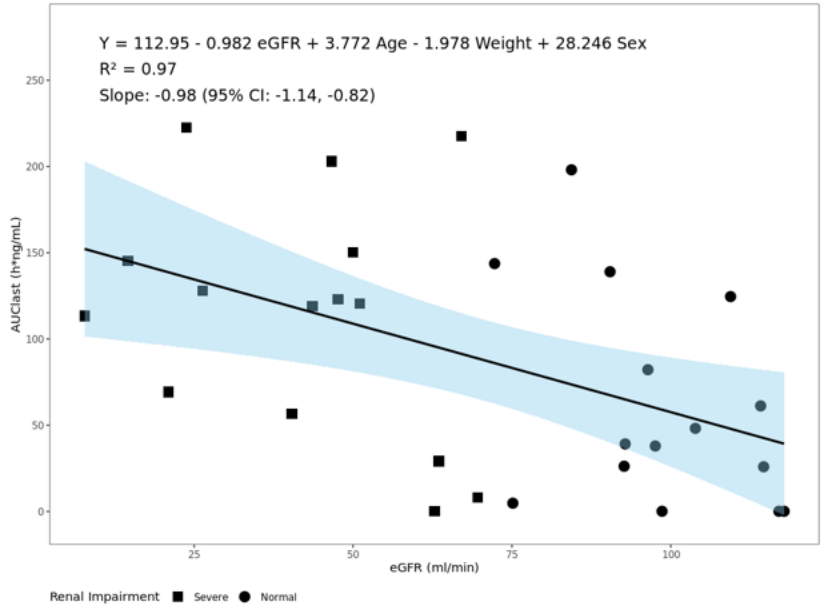
dataframe pkdata contains the following variables:
'AVAL'(type:numeric);'AGE'(type:numeric);'WEIGHT'(type:numeric)
'SEXN'(type:numeric);'RFGROUP'(type:character);'EGFRN'(type:numeric)
Create a R graph based on pkdata, y var is AVAL, x var is EGFRN.
Add scatter plot and a regression line with 95%CI band.

Group scatter shape by RFGROUP, 'Severe' to square, 'Normal' to circle, set the legend title to 'Renal Impairment', put legend at bottom.
Change the band color to skyblue, theme to theme_test().

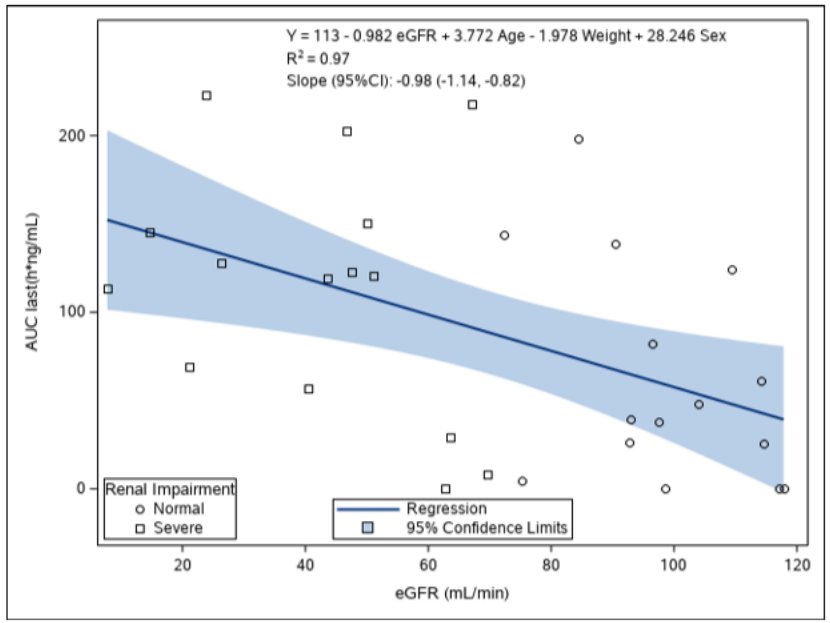
Add annotation to the graph include:
formulation of the regression model AVAL as Y, EGFRN, AGE, WEIGHT and SEXN as covariates, with the coefficients,
R square of the regression model,
Slope of EGFRN and 95%CI.

Set ymax to 280 and put the annotations on the top left.
For the formulation if '+' a negative number, then replace by '-',
Update 'AVAL' to 'Y', 'EGFRN' to 'eGFR', 'AGE' to 'Age', 'WEIGHT' to 'Weight', 'SEXN' to 'Sex'.
Remove ' of EGFRN' from slope.

QC by R



Production by SAS



dataframe pkdata contains the following variables:
'AVAL'(type:numeric);'AGE'(type:numeric);'WEIGHT'(type:numeric)
'SEXN'(type:numeric);'RFGROUP'(type:character);'EGFRN'(type:numeric)
Create a R graph based on pkdata, y var is AVAL, x var is EGFRN.
Add scatter plot and a regression line with 95%CI band.

Group scatter shape by RFGROUP, 'Severe' to square, 'Normal' to circle, set the legend title to 'Renal Impairment', put legend at bottom.
Change the band color to skyblue, theme to theme_test().

Add annotation to the graph include:
formulation of the regression model AVAL as Y, EGFRN, AGE, WEIGHT and SEXN as covariates, with the coefficients,
R square of the regression model,
Slope of EGFRN and 95%CI.

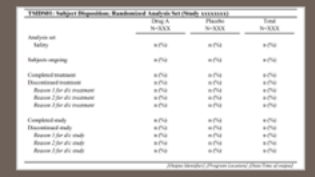
Set ymax to 280 and put the annotations on the top left.
For the formulation if '+' a negative number, then replace by '-',
Update 'AVAL' to 'Y', 'EGFRN' to 'eGFR', 'AGE' to 'Age', 'WEIGHT' to 'Weight', 'SEXN' to 'Sex'.
Remove ' of EGFRN' from slope.

Remove title and subtitle, set Y label as 'AUClast(h*ng/mL)',
set X label as 'eGFR (ml/min)'.

TSIDS01: Subject Disposition; Randomized Analysis Set (Study xxxxxxx)

	Drug A N=XXX	Placebo N=XXX	Total N=XXX
Analysis set			
Safety	n (%)	n (%)	n (%)
Subjects ongoing	n (%)	n (%)	n (%)
Completed treatment	n (%)	n (%)	n (%)
Discontinued treatment	n (%)	n (%)	n (%)
Reason 1 for d/c treatment	n (%)	n (%)	n (%)
Reason 2 for d/c treatment	n (%)	n (%)	n (%)
Reason 3 for d/c treatment	n (%)	n (%)	n (%)
Completed study	n (%)	n (%)	n (%)
Discontinued study	n (%)	n (%)	n (%)
Reason 1 for d/c study	n (%)	n (%)	n (%)
Reason 2 for d/c study	n (%)	n (%)	n (%)
Reason 3 for d/c study	n (%)	n (%)	n (%)

```
> # Add the "Safety" row with indentation
> lyt <- analyze(
+   lyt,
+   vars = "USUBJID",
+   afun = function(x, .N_col) format_count_pct(x, .N_col),
+   subset = expression(SAFFL == "Y"),
+   .indent = 1,
+   .labels = c("USUBJID" = "Safety")
+ )
Error in analyze(lyt, vars = "USUBJID", afun = function(x, .N_col) format_count_pct(x,  :
  unused arguments (subset = expression(SAFFL == "Y"), .indent = 1, .labels = c(USUBJID = "Safety"))

> lyt <- summarize_row_groups( lyt, group = function(df, .N_col) {
+   # Filter to safety population
+   safety_df <- df[df$SAFFL == "Y", ]
+   n <- nrow(safety_df)
+   pct <- 100 * n / .N_col
+   return(paste0(n, " (", sprintf("%.1f", pct), "%)"))
+   },
+   label = " Safety"   # Using spaces for indentation
+   )
Error in summarize_row_groups(lyt, group = function(df, .N_col) { :
  unused argument (group = function(df, .N_col) {
    safety_df <- df[df$SAFFL == "Y", ]
    n <- nrow(safety_df)
    pct <- 100 * n/.N_col
    return(paste0(n, " (", sprintf("%.1f", pct), "%)"))
})

> lyt <- add_row_groups(lyt, "Analysis set")
Error in add_row_groups(lyt, "Analysis set") :
  could not find function "add_row_groups"
```



Create a table using rtables and tern package, refer to the attached mockup. Use adsl dataset, TRT01P as group var, SAFFL = Y for the safety row. Start with the first 2 rows.

▼ Code

```r
lyt <- basic_table(show_colcounts = TRUE) %>%
  split_cols_by(
    "ACTARM",
    split_fun = add_overall_level("All Patients", first = FALSE)
  ) %>%
  count_occurrences(
    "EOSSTT",
    show_labels = "hidden"
  ) %>%
  analyze_vars(
    "DCSREAS",
    .stats = "count_fraction",
    denom = "N_col",
    show_labels = "hidden",
    .indent_mods = c(count_fraction = 1L)
  )

result1 <- build_table(lyt = lyt, df = adsl)
result1
```

	A: Drug X (N=134)	B: Placebo (N=134)	C: Combination (N=132)	All Patients (N=400)
COMPLETED	68 (50.7%)	66 (49.3%)	73 (55.3%)	207 (51.7%)
ONGOING	24 (17.9%)	28 (20.9%)	21 (15.9%)	73 (18.2%)
DISCONTINUED	42 (31.3%)	40 (29.9%)	38 (28.8%)	120 (30.0%)
ADVERSE EVENT	3 (2.2%)	6 (4.5%)	5 (3.8%)	14 (3.5%)
DEATH	25 (18.7%)	23 (17.2%)	22 (16.7%)	70 (17.5%)
LACK OF EFFICACY	2 (1.5%)	2 (1.5%)	3 (2.3%)	7 (1.8%)
PHYSICIAN DECISION	2 (1.5%)	3 (2.2%)	2 (1.5%)	7 (1.8%)
PROTOCOL VIOLATION	5 (3.7%)	3 (2.2%)	4 (3%)	12 (3%)
WITHDRAWAL BY PARENT/GUARDIAN	4 (3%)	2 (1.5%)	1 (0.8%)	7 (1.8%)
WITHDRAWAL BY SUBJECT	1 (0.7%)	1 (0.7%)	1 (0.8%)	3 (0.8%)

▼ Code

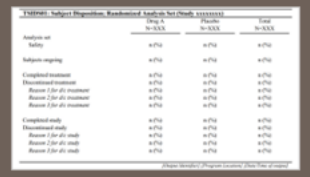
```r
vars <- c("AGE", "AGEGR1", "SEX", "ETHNIC", "RACE", "BMRKR1")
var_labels <- c(
  "Age (yr)",
  "Age Group",
  "Sex",
  "Ethnicity",
  "Race",
  "Continous Level Biomarker 1"
)

result <- basic_table(show_colcounts = TRUE) %>%
  split_cols_by(var = "ACTARM") %>%
  add_overall_col("All Patients") %>%
  analyze_vars(
    vars = vars,
    var_labels = var_labels
  ) %>%
  build_table(adsl)

result
```

	A: Drug X (N=134)	B: Placebo (N=134)	C: Combination (N=132)	All Patients (N=400)
Age (yr)				
n	134	134	132	400
Mean (SD)	33.8 (6.6)	35.4 (7.9)	35.4 (7.7)	34.9 (7.4)
Median	33.0	35.0	35.0	34.0
Min - Max	21.0 - 50.0	21.0 - 62.0	20.0 - 69.0	20.0 - 69.0
Age Group				
n	134	134	132	400
18-40	113 (84.3%)	103 (76.9%)	106 (80.3%)	322 (80.5%)
41-64	21 (15.7%)	31 (23.1%)	25 (18.9%)	77 (19.2%)
>=65	0	0	1 (0.8%)	1 (0.2%)

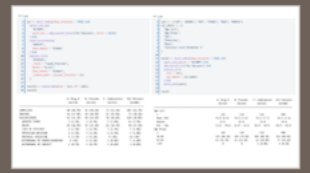
TSIDS01: Subject Disposition; Randomized Analysis Set (Study ABC1001)

	Drug A (N=134)	Placebo (N=134)	Total (N=268)
Analysis set			
Safety	134 (100.0%)	134 (100.0%)	268 (100.0%)

[tsids01v1.rtf][tsids01.R] 27MAR2025, 06:04



Create a table using rtables and tern package, refer to the attached mockup. Use adsl dataset, TRT01P as group var, SAFFL = Y for the safety row. Start with the first 2 rows.

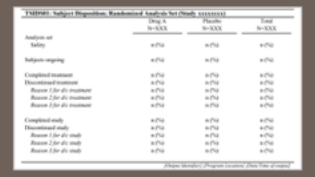


Refer to the attached sample code, use var_labels for “Analysis” and in_rows for “Safety”.

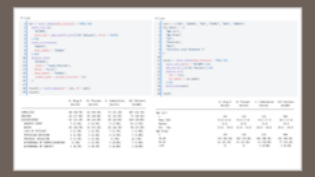
TSIDS01: Subject Disposition; Randomized Analysis Set (Study ABC1001)

	Drug A N=134	Placebo N=134	Total N=268
Analysis set			
Safety	134 (100.0%)	134 (100.0%)	268 (100.0%)

[tsids01v2.rtf][tsids01.R] 27MAR2025, 06:04



Create a table using rtables and tern package, refer to the attached mockup. Use adsl dataset, TRT01P as group var, SAFFL = Y for the safety row. Start with the first 2 rows.



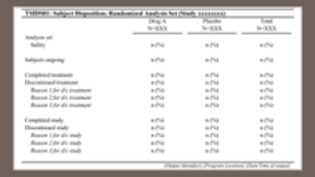
Refer to the attached sample code, use var_labels for “Analysis” and in_rows for “Safety”.

Remove “()” from the header.

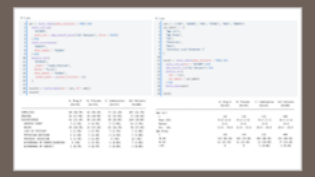
TSIDS01: Subject Disposition; Randomized Analysis Set (Study ABC1001)

	Drug A N=134	Placebo N=134	Total N=268
Analysis set			
Safety	134 (100.0%)	134 (100.0%)	268 (100.0%)
Subjects ongoing	24 (17.9%)	28 (20.9%)	52 (19.4%)

[tsids01v3.rtf][tsids01.R] 27MAR2025, 06:36



Create a table using rtables and tern package, refer to the attached mockup. Use adsl dataset, TRT01P as group var, SAFFL = Y for the safety row. Start with the first 2 rows.



Refer to the attached sample code, use var_labels for “Analysis” and in_rows for “Safety”.

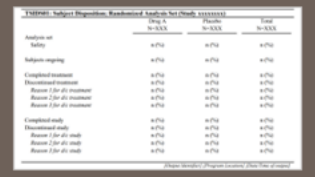
Remove “()” from the header.

Add a blank row, then add ongoing subjects count with EOSSTT = ONGOING.

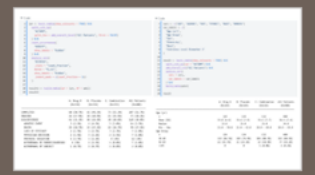
TSIDS01: Subject Disposition; Randomized Analysis Set (Study ABC1001)

	Drug A N=134	Placebo N=134	Total N=268
Safety	134 (100.0%)	134 (100.0%)	268 (100.0%)
Subjects ongoing	24 (17.9%)	28 (20.9%)	52 (19.4%)
Completed treatment	51 (38.1%)	46 (34.3%)	97 (36.2%)

[tsids01v4.rtf][tsids01.R] 27MAR2025, 06:52



Create a table using rtables and tern package, refer to the attached mockup. Use adsl dataset, TRT01P as group var, SAFFL = Y for the safety row. Start with the first 2 rows.



Refer to the attached sample code, use var_labels for “Analysis” and in_rows for “Safety”.

Remove “()” from the header.

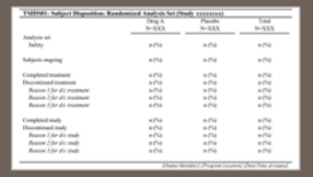
Add a blank row, then add ongoing subjects count with EOSSTT = ONGOING.

Add a blank row, then add Completed treatment with EOTSTT = COMPLETED.

TSIDS01: Subject Disposition; Randomized Analysis Set (Study ABC1001)

	Drug A N=134	Placebo N=134	Total N=268
Analysis set			
Safety	134 (100.0%)	134 (100.0%)	268 (100.0%)
Subjects ongoing	24 (17.9%)	28 (20.9%)	52 (19.4%)
Completed treatment	51 (38.1%)	46 (34.3%)	97 (36.2%)
Discontinued treatment	49 (36.6%)	40 (29.9%)	89 (33.2%)
Adverse Event	11 (8.2%)	5 (3.7%)	16 (6%)
Death	8 (6%)	5 (3.7%)	13 (4.9%)
Lost to Follow-Up	4 (3%)	8 (6%)	12 (4.5%)
Progressive Disease	8 (6%)	9 (6.7%)	17 (6.3%)
Protocol Deviation	10 (7.5%)	7 (5.2%)	17 (6.3%)
Other	8 (6%)	6 (4.5%)	14 (5.2%)
Completed study	68 (50.7%)	66 (49.3%)	134 (50.0%)
Discontinued study	42 (31.3%)	40 (29.9%)	82 (30.6%)
Adverse Event	6 (4.5%)	7 (5.2%)	13 (4.9%)
Death	11 (8.2%)	6 (4.5%)	17 (6.3%)
Lost to Follow-Up	3 (2.2%)	4 (3%)	7 (2.6%)
Progressive Disease	12 (9%)	7 (5.2%)	19 (7.1%)
Protocol Deviation	5 (3.7%)	8 (6%)	13 (4.9%)
Other	5 (3.7%)	8 (6%)	13 (4.9%)

[tsids01v5.rtf][tsids01.R] 27MAR2025, 06:58



Create a table using rtables and tern package, refer to the attached mockup. Use adsl dataset, TRT01P as group var, SAFFL = Y for the safety row. Start with the first 2 rows.

Refer to the attached sample code, use var_labels for “Analysis” and in_rows for “Safety”.

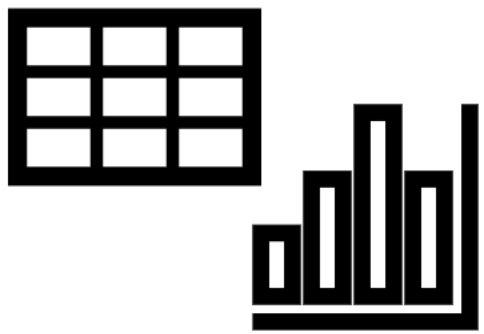
Remove “()” from the header.

Add a blank row, then add ongoing subjects count with EOSSTT = ONGOING.

Add a blank row, then add Completed treatment with EOTSTT = COMPLETED.

Add count for EOT reasons by DCTREAS, and add Completed study and EOS reasons below. The reasons should be sorted as "Adverse Event", "Death", "Lost to Follow-Up", "Progressive Disease", "Protocol Deviation” and "Other”.

R + AI



R for TLG

R Shiny

R Learning

SAS TO R CODE TRANSLATOR

SAS code

```
data c;
  merge a b;
  by group;
run;
```

R code

Some explainations

.....

.....

.....

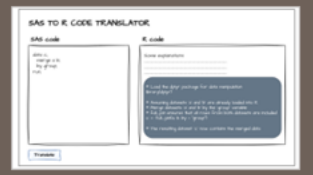
```
# Load the dplyr package for data manipulation
library(dplyr)

# Assuming datasets 'a' and 'b' are already loaded into R
# Merge datasets 'a' and 'b' by the 'group' variable
# full_join ensures that all rows from both datasets are included
c <- full_join(a, b, by = "group")

# The resulting dataset 'c' now contains the merged data
```

Translate

SAS to R Code Translator

SAS Code

⇄ Translate →



Create a R shiny app, refer to the attached sketch, to translate SAS code to R code. UI first.

SAS to R Code Translator

SAS Code

```
data c;
  merge a b;
  by group;
run;
```

Translate →

Here is the equivalent R code using the `tidyverse` syntax, with detailed comme
explain each step:

Copy

```r
# Load the tidyverse package, which includes dplyr for data manipulation
library(tidyverse)

# Assuming 'a' and 'b' are data frames already loaded in R

# Merge the two data frames 'a' and 'b' by the common column 'group'
# The `by` argument specifies the column(s) to merge by
# The `full_join` function from dplyr performs a full outer join, similar to SAS's mer
ge
c <- full_join(a, b, by = "group")

# The resulting data frame 'c' contains all rows from both 'a' and 'b',
# with missing values filled in where there is no match in the other data frame
```

Explanation:
1. **`library(tidyverse)`**: Loads the `tidyverse` package, which includes `dplyr` for data manipulation.
2. **`full_join(a, b, by = "group")`**: This function merges the two data frames `a` and `b` by the column `group`. The `full_join` function performs a full outer join, which means it includes all rows from both data frames, filling in `NA` where there are no matches.
3. **`c <- ...`**: Assigns the result of the merge to a new data frame `c`.

This R code maintains the structure of the original SAS code and uses `tidyverse` syntax for clarity and consistency.

Create a R shiny app, refer to the attached sketch, to translate SAS code to R code. UI first.

Link to API of AI model with **api_key** and **api_url**, and add the translating functionality based on the API.

SAS to R Code Translator

SAS Code

```
data c;
  merge a b;
  by group;
run;
```

⇄ Translate →

Translation Process Explanation:

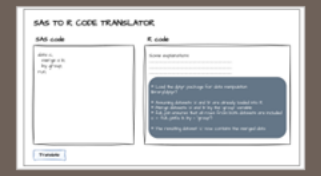
The SAS code provided merges two datasets (`a` and `b`) by a common variable `group` . In R, the equivalent operation can be performed using the `dplyr` package from the tidyverse, specifically the `full_join()` function. This function combines rows from two datasets based on a common key (`group` in this case), ensuring all rows from both datasets are included in the result.

R Code:

```r
# Load the tidyverse package
library(dplyr)

# Assuming datasets 'a' and 'b' are already loaded in the R environm
ent
# Merge datasets 'a' and 'b' by the 'group' column
c <- full_join(a, b, by = "group")

# Explanation:
# - `full_join()` merges two datasets, keeping all rows from both da
tasets.
# - The `by = "group"` argument specifies that the merge should be d
one based on the 'group' column.
# - The result is stored in a new dataset `c`.
```

How the R Code Works:

1. `library(dplyr)` : Loads the `dplyr` package, which provides the `full_join()`



Create a R shiny app, refer to the attached sketch, to translate SAS code to R code. UI first.

Link to API of AI model with **api_key** and **api_url**, and add the translating functionality based on the API.

Highlight R code with Rmarkdown format and add a copy button on topright.

SAS to R Code Translator

SAS Code

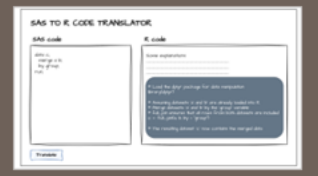
```
data c;
  merge a b;
  by group;
run;
```

Translation Mode:
- (●) Pure translation
- ( ) With explanation

⇄ Translate →

R Code

```r
# Translated from SAS:
#  data c;
#    merge a b;
#    by group;
#  run;

# Basic SAS to R translation
# For the SAS code 'data a; merge b c; by group; run;'
a <- merge(b, c, by = "group", all = TRUE)
```



Create a R shiny app, refer to the attached sketch, to translate SAS code to R code. UI first.

Link to API of AI model with **api_key** and **api_url**, and add the translating functionality based on the API.

Highlight R code with Rmarkdown format and add a copy button on topright.

Add selection of pure translation (display R code only) and with explanation.

SAS to R Code Translator

SAS Code

```
data c;
  merge a b;
  by group;
run;
```

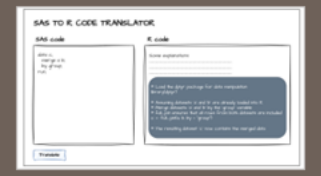
Translation Mode:
- (●) Pure translation
- ( ) With explanation

Model Selection:
- ( ) Chat
- (●) Reasoner (slow)

⇄ Translate →

R Code

```r
# Translated from SAS:
#  data c;
#    merge a b;
#    by group;
#  run;

# Basic SAS to R translation
# For the SAS code 'data a; merge b c; by group; run;'
a <- merge(b, c, by = "group", all = TRUE)
```



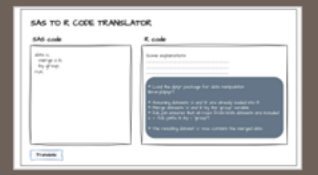
Create a R shiny app, refer to the attached sketch, to translate SAS code to R code. UI first.

Link to API of AI model with **api_key** and **api_url**, and add the translating functionality based on the API.

Highlight R code with Rmarkdown format and add a copy button on topright.

Add selection of pure translation (display R code only) and with explanation.

Add selection of chat model and reasoner model.



Create a R shiny app, refer to the attached sketch, to translate SAS code to R code. UI first.

Link to API of AI model with **api_key** and **api_url**, and add the translating functionality based on the API.

Highlight R code with Rmarkdown format and add a copy button on topright.

Add selection of pure translation (display R code only) and with explanation.

Add selection of chat model and reasoner model.

Add a chat-box for extra requests for the translation, like specific package. Also, update the title to look better, add Jiaqi and AI as author.

SAS to R Code Translator

Convert your SAS code to R with ease

Developed by Jiaqi Song and AI

SAS Code

```
data c;
  merge a b;
  by group;
run;
```

Extra request

use dplyr package

Translation Mode:

- ○ Pure translation
- ◉ With explanation

Model Selection:

- ○ Chat
- ◉ Reasoner (slow)

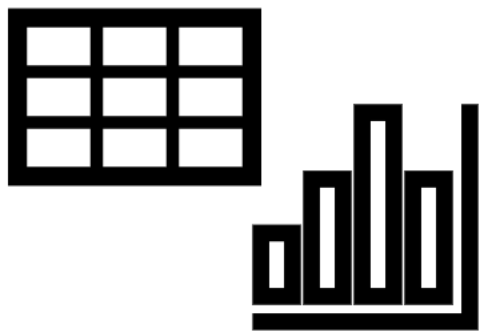
⇄ Translate →

```r
library(dplyr)

# Identify overlapping columns (excluding the 'group' key)
common_vars <- intersect(names(a), names(b)) %>%
  setdiff("group")

# Perform full outer join with suffixes to track source
c <- full_join(a, b, by = "group", suffix = c(".a", ".b")) %>%
  # Process overlapping variables: prioritize 'b' values
  mutate(
    across(
      all_of(common_vars),
      # Coalesce 'b' values first, then 'a' values
      ~ coalesce(.data[[paste0(cur_column(), ".b")]],
                 .data[[paste0(cur_column(), ".a")]]),
      .names = "{.col}"
    )
  ) %>%
  # Remove temporary columns with suffixes
  select(-ends_with(c(".a", ".b")))

# Explanation:
# 1. full_join combines all rows from both datasets
# 2. across() applies to all common variables (non-group)
# 3. coalesce() prioritizes values from dataset 'b' (SAS default)
# 4. select() removes temporary suffixed columns
# 5. Maintains unique columns from both datasets automatically
```

Key features:

- `full_join` ensures no data loss from either dataset
- `coalesce` replicates SAS's value precedence (b overwrites a)
- Suffix handling maintains original column names
- Efficient piping structure matches SAS's single data step

R + AI



R for TLG

R Shiny

R Learning

R Learning

Traditional way

Read Books

AI empowered way

Smart Reading with
AI Assistance

1 Data visualization

1.1 Introduction

> "The simple graph has brought more information to the data analyst's mind than any other device." — John Tukey

R has several systems for making graphs, but ggplot2 is one of the most elegant and most versatile. ggplot2 implements the **grammar of graphics**, a coherent system for describing and building graphs. With ggplot2, you can do more and faster by learning one system and applying it in many places.

This chapter will teach you how to visualize your data using **ggplot2**. We will start by creating a simple scatterplot and use that to introduce aesthetic mappings and geometric objects – the fundamental building blocks of ggplot2. We will then walk you through visualizing distributions of single variables as well as visualizing relationships between two or more variables. We'll finish off with saving your plots and troubleshooting tips.

1.1.1 Prerequisites

This chapter focuses on ggplot2, one of the core packages in the tidyverse. To access the datasets, help pages, and functions used in this chapter, load the tidyverse by running:

```
library(tidyverse)
#> ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
#> ✓ dplyr     1.1.4     ✓ readr     2.1.5
#> ✓ forcats   1.0.0     ✓ stringr   1.5.1
#> ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
#> ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
```

visualizations, starting with scatterplots and progressing to visualizations of variable distributions and relationships. Users are guided to load the tidyverse and necessary packages, including palmerpenguins for dataset access and ggthemes for color palettes. Key terms like variable, value, observation, and the concept of tidy data are defined. The chapter uses the penguins dataset to explore relationships between flipper lengths and body masses across species, ultimately guiding readers to recreate a specific visualization. Initial steps involve loading data and creating a ggplot object, setting the stage for additional layers and custom visualizations.

Related Questions

- What is ggplot2 and why is it important for data visualization?
- How many columns are in the penguins dataset and what are some of them?
- What are the prerequisites for working with ggplot2 in this chapter?

summarize this page with bullets

Sider Fusion

Here's a summary of the provided content, using bullet points:

- **Introduction to ggplot2:** The chapter introduces ggplot2, a versatile and elegant data visualization package in R, based on the grammar of graphics .
- **Core Concepts:** It covers creating scatterplots, visualizing variable distributions, and relationships between variables . Key terms like variable, value, observation, and tidy data are defined .
- **Prerequisites:** The chapter requires loading the `tidyverse` package . It may need to be installed if it's not already available. The `palmerpenguins` and `ggthemes` packages are also used .
- **Penguins Dataset:** The `penguins` dataset from the `palmerpenguins` package is used to explore relationships between flipper length and body mass across different penguin species .
- **Creating a ggplot:** The process starts with creating a `ggplot()` object and adding layers to it, such as geometric objects (`geom_point()` , `geom_smooth()`) .
- **Aesthetic Mappings:** The `mapping` argument in `ggplot()` defines how variables are mapped to visual properties like x, y, and color .
- **Adding Aesthetics and Layers:** Aesthetics can be added to `geom_point()` to represent species with different colored points . `geom_smooth()` can add a smooth curve to display the relationship between variables .
- **Improving Plots:** The chapter explains how to use different shapes for each species and improve labels using `labs()` and `scale_color_colorblind()` .
- **Visualizing Distributions:** It covers visualizing distributions of categorical variables using bar charts (`geom_bar()`) and numerical variables using histograms (`geom_histogram()`) and density plots (`geom_density()`) .
- **Visualizing Relationships:** Discusses visualizing relationships between numerical and categorical

Sider Fusion

Type a message...



documentation.

Generally, however, we recommend that you assemble your final reports using Quarto, a reproducible authoring system that allows you to interleave your code and your prose and automatically include your plots in your write-ups. You will learn more about Quarto in Chapter 28.

1.6.1 Exercises

1. Run the following lines of code. Which of the two plots is saved as `mpg-plot.png`? Why?

```
ggplot(mpg, aes(x = class)) +
  geom_bar()
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
ggsave("mpg-plot.png")
```

2. What do you need to change in the code above to save the plot as a PDF instead of a PNG? How could you find out what types of image files would work in `ggsave()`?

1.7 Common problems

As you start to run R code, you're likely to run into problems. Don't worry — it happens to everyone. We have all been writing R code for years, but every day we still write code that doesn't work on the first try!

Start by carefully comparing the code that you're running to the code in the book. R is extremely picky, and a misplaced character can make all the difference. Make sure that every `(` is matched with a `)` and every `"` is paired with another `"`. Sometimes you'll run the code and nothing happens. Check the left-hand of your console: if it's a `+`, it means that R doesn't think you've typed a complete expression and it's waiting for you to

R Learning

Traditional way

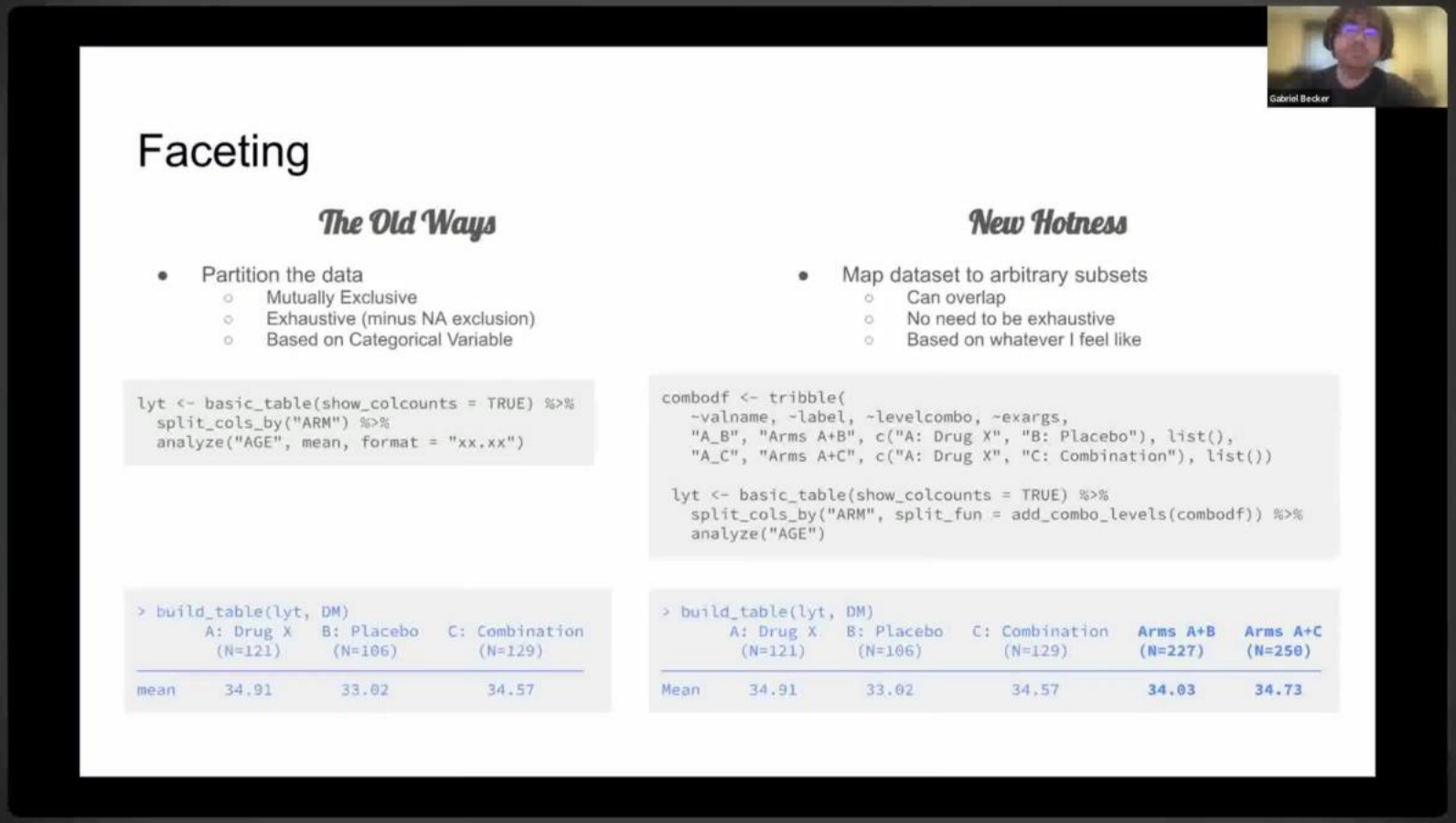
Read Books

Watch Videos

AI empowered way

Smart Reading with AI Assistance

Video Insights through AI Summarization



Rtables: Leveraging Data Visualization Concepts To Declare And Create Clinical Trial Tables

R Consortium
21.1K subscribers

Subscribe

12 Share Download Clip Save

503 views Aug 3, 2023
Gabriel Becker, Consulting R package developer

Reporting tables form a central component of regulatory filings in the pharmaceutical industry. These tables are often complex, both in their conceptual structure and in the computations required to generate their individual cell values. In this session presentation, Gabe introduced the rtables package and show how visualization concepts, particularly faceting and the grammar of graphics, apply to tables generally and inform rtables' ability to succinctly declare and create complex structured tables such as those included in clinical trial filings. Gabe also gives a brief update of the work of the R Consortium's R Tables For Regulatory Submission (RTRS) working group.

Main Sections

EN

Subtitles Bilingual

00:00 next talk is going to be by Gabriel
00:02 Becker who will be speaking to us about
00:05 our tables leveraging data visualization
00:07 Concepts to declare and create clinical
00:09 trial tables and I think they will be
00:12 speaking live I see you're here and
00:14 unmuted so yeah
00:16 share your slides and get started yep
00:26 that do you see the schedule or do you
00:29 see my slides
00:31 here slides great
00:36 yeah so this is going to be a pretty

Watch Highlights

In this video

Timeline Chapters Transcript

14:47
Faceting 15:24
Subplots 17:05
Marginal summaries 19:02

generalizations made. I will focus on these generalizations to answer the user's query.

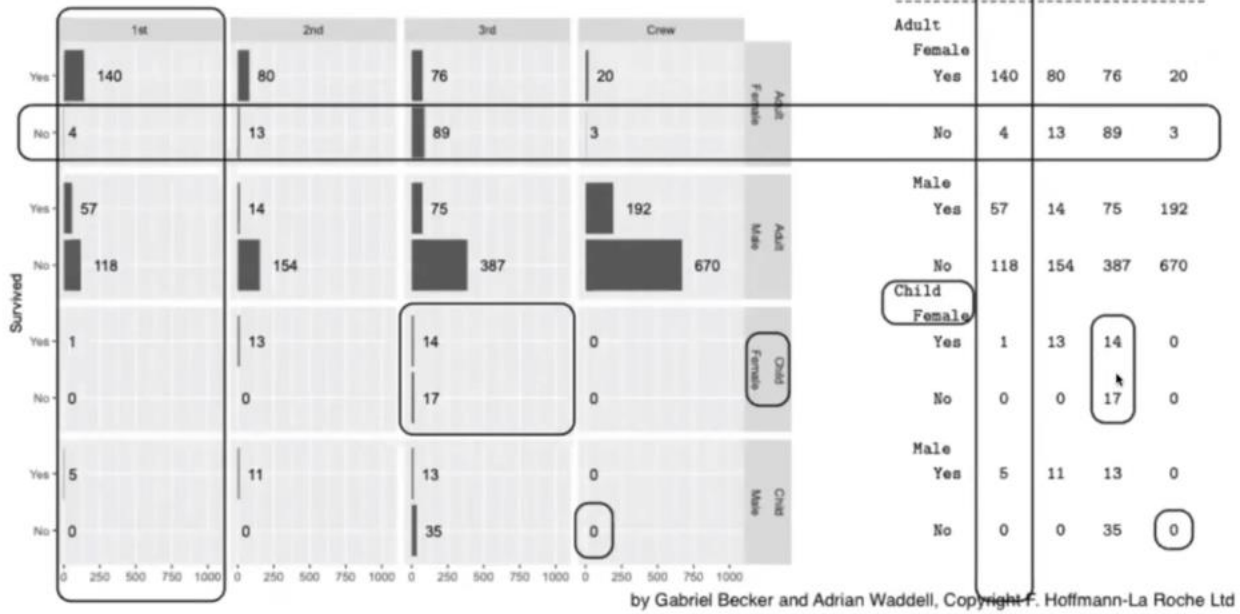
Faceting, according to the video, traditionally involves partitioning data into mutually exclusive and exhaustive subsets. However, `rtables` generalizes this concept, offering the following benefits:

1. **Overlapping Facets:** `rtables` allows facets to overlap, which is useful for creating categories like "all patients" or combining treatment arms for analysis.
2. **Dissimilar Subplots:** Unlike typical faceted plots where subplots are uniform, `rtables` enables different types of content within facets (e.g., mean and range). It also allows columns to have different meanings.
3. **Marginal Summaries:** `rtables` facilitates marginal summaries, generating group summaries and incorporating them into tables. It can handle multiple levels of information being marginally summarized, which is difficult to achieve in plots.

Compare with GPT-4o

Type a message...

Tables Are Faceted Data Visualizations



		1st	2nd	3rd	Crew
Adult					
Female					
	Yes	140	80	76	20
	No	4	13	89	3
Male					
	Yes	57	14	75	192
	No	118	154	387	670
Child					
Female					
	Yes	1	13	14	0
	No	0	0	17	0
Male					
	Yes	5	11	13	0
	No	0	0	35	0

by Gabriel Becker and Adrian Waddell, Copyright F. Hoffmann-La Roche Ltd

02:41 Data visualization is crucial for creating effective clinical trial tables. Understanding the relationship between tables and faceted visualizations can enhance reporting and data interpretation.

04:05 Tables can be viewed as faceted plots, which represent data through a structured layout of rows and columns. Understanding this relationship helps in effectively visualizing and interpreting data.

08:09 Defining a table structure involves a small set of low-level verbs that can be combined for complex functionalities. This process includes declaring faceting and deriving cell values effectively.

12:13 Data visualization can leverage complex structured tables for effective analysis, particularly in clinical trials and exploratory data analysis. This approach enhances the understanding of faceted data visualizations.

16:18 Faceting in data visualization allows for multiple subplots to be created for different categories, providing a clearer analytical perspective. This technique goes beyond basic comparisons, showcasing relationships in various data dimensions.

20:22 The TLG catalog is an open-source resource that aids in generating tables for clinical trial analysis. It includes 225 variants

R Learning

Traditional way

Read Books

Watch Videos

Ask Experts

AI empowered way

Smart Reading with AI Assistance

Video Insights through AI Summarization

24/7 AI Expert Support

What does "+" mean in R package ggplot2?

In the R package ggplot2, the "+" operator is used to add new layers to a plot. It allows you to sequentially add multiple layers of elements (such as data, aesthetics, and geoms) to build up a complex visualization.

For example:

```
library(ggplot2)
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy
```

What does "+" mean in R package ggplot2?

示 。

图层叠加

1. **创建画布**：首先使用 `ggplot()` 函数创建一个绘图空间，指定数据和基本的映射关系 。

2. **添加图层**：随后使用 `geom_` 系列函数添加几何对象图层，如 `geom_point()` 添加散点图层， `geom_line()` 添加线条图层等 。

3. **修饰图层**：进一步使用 `scale_` 系列函数调整标度，如颜色、大小等； `theme()` 函数调整主题，如背景、字体等；以及 `labs()` 函数添加图例和标题等 。

示例

R 复制

```r
# 加载ggplot2包
library(ggplot2)

# 创建基础图层，指定数据和映射
ggplot(data = mtcars, aes(x = wt, y = mpg))

# 添加散点图层
```

接着问

What does the plus sign mean in R package GG plot two?

It's a way to piece together a plot. You can use it to add layers like geom_bar() for bar charts or geom_line() for line graphs. It lets you combine different visual elements to create the final plot you want. Have you used it to make any interesting plots?

说话或点击打断

R Learning

Traditional way

Read Books

Watch Videos

Ask Experts

AI empowered way

Smart Reading with AI Assistance

Video Insights through AI Summarization

24/7 AI Expert Support

Customized Learning

I want to learn the ggplot2 package in R. I have a basic understanding of R, so please create a 4-week study plan for me.

4-Week ggplot2 Learning Plan

Week 1: Introduction and Basic Plotting
Objective: Understand ggplot2 basics, including concepts and basic plots.
Day 1: Introduction to ggplot2
Day 2: Basic Scatter Plots
Day 3: Customizing Plots
Day 4: Bar Plots and Histograms
Day 5: Saving and Exporting

Week 2: Aesthetic Mappings and Customizations
Objective: Master aesthetic customizations and understand mapping data.
Day 1: Aesthetic Mappings
Day 2: Enhanced Plots
Day 3: Complex Charts
Day 4: Colors and Scales
Day 5: Coordinate Systems

Day 4: Colors and Scales
Day 5: Coordinate Systems

Week 3: Faceting, Themes, and Extensions
Objective: Learn to use faceting, themes, and package extensions.
Day 1: Faceting
Day 2: Themes
Day 3: Theme Elements
Day 4: Extensions
Day 5: Theme Practice

Week 4: Advanced Techniques and Integration
Objective: Apply advanced techniques and integrate with other R packages.
Day 1: Complex Visualizations
Day 2: Package Integration
Day 3: Interactive Elements
Day 4: Review
Day 5: Final Project

More details about Week 1, Day 3 and 4

Detailed ggplot2 Learning Plan

Week 1: Introduction and Basic Plotting
Objective: Understand ggplot2 basics, including concepts and basic plots.

Day 3: Customizing Plots
- Understanding the layers concept
- Working with colors and titles
- Basic geoms: geom_point(), geom_line()

Day 4: Bar Plots and Histograms
- Creating bar plots with geom_bar()
- Building histograms with geom_histogram()
- Understanding distributions

```
Start the learning of week 1 day 4 as a demo. You should give me introduction, syntax,
examples and exercises. Start with the first bullet point. Deliver as .rmd
```

Start the learning of week 1 day 4 as a demo. You should give me introduction, syntax, examples and exercises. Start with the first bullet point. Deliver as .rmd

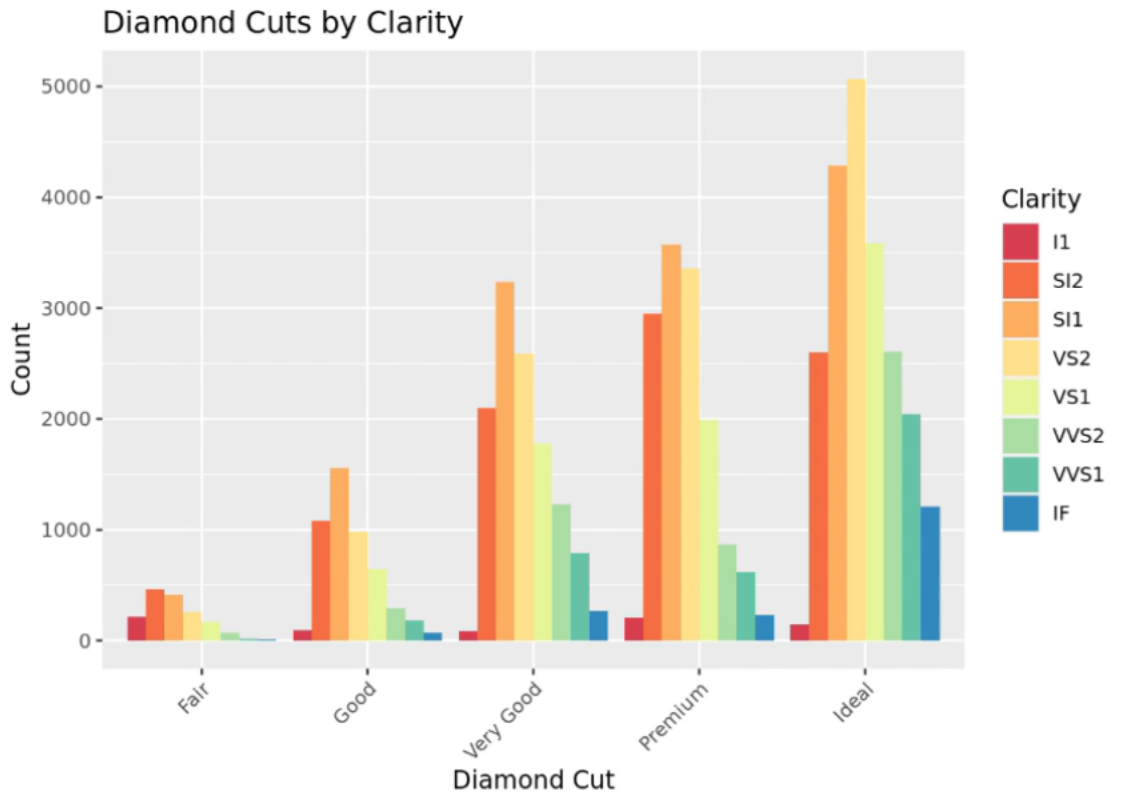
I'll create a comprehensive lesson for Week 1, Day 4, focusing on bar plots.

ggplot2 Tutorial: Bar Plots (RMarkdown)
Click to preview

This lesson covers the first part of bar plots in detail. Would you like to proceed with the histogram portion next, or would you like to practice with these concepts first?

Exercise 3: Grouped Bar Plot

```
ggplot(diamonds, aes(x = cut, fill = clarity)) +
  geom_bar(position = "dodge") +
  labs(x = "Diamond Cut",
       y = "Count",
       title = "Diamond Cuts by Clarity",
       fill = "Clarity") +
  scale_fill_brewer(palette = "Spectral") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Challenge Task

Create a stacked percentage bar plot showing the proportion of clarity categories within each cut type:

```
ggplot(diamonds, aes(x = cut, fill = clarity)) +
```

Start the learning of week 1 day 4 as a demo. You should give me introduction, syntax, examples and exercises. Start with the first bullet point. Deliver as .rmd

I'll create a comprehensive lesson for Week 1, Day 4, focusing on bar plots.

ggplot2 Tutorial: Bar Plots (RMarkdown)
Click to preview

This lesson covers the first part of bar plots in detail. Would you like to proceed with the histogram portion next, or would you like to practice with these concepts first?

Proceed with histogram. Create the RMarkDown file in Shiny presentation format.

I'll create a comprehensive lesson for Week 1, Day 4, focusing on Histograms.

ggplot2 Tutorial: Histograms (Shiny Presentation)
Click to preview

Would you like me to:
- Include different transition effects?
- Add more keyboard shortcuts?
- Adjust the sizing of specific elements?

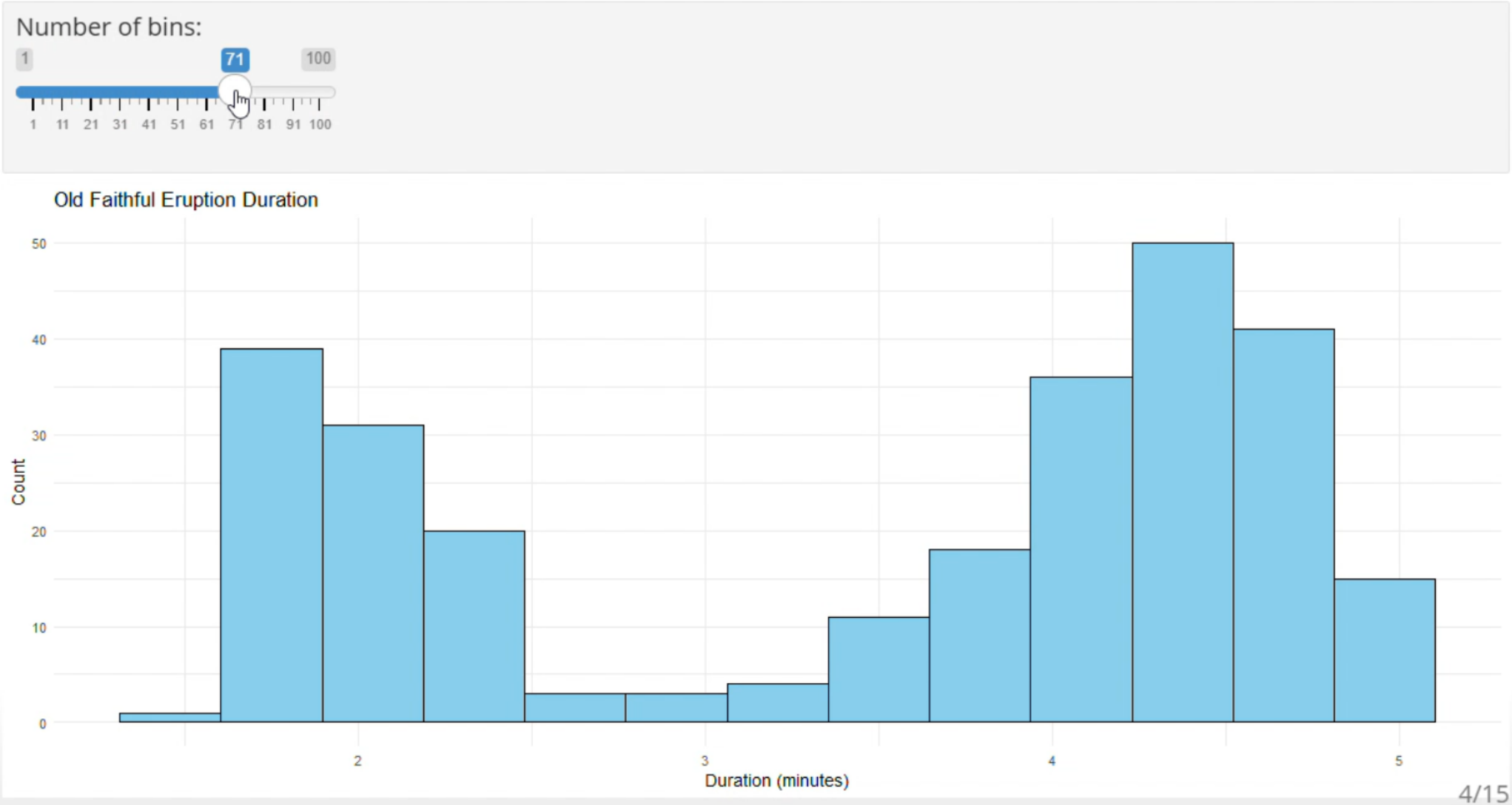
Histograms in ggplot2

Week 1 - Day 4 | Lesson 2

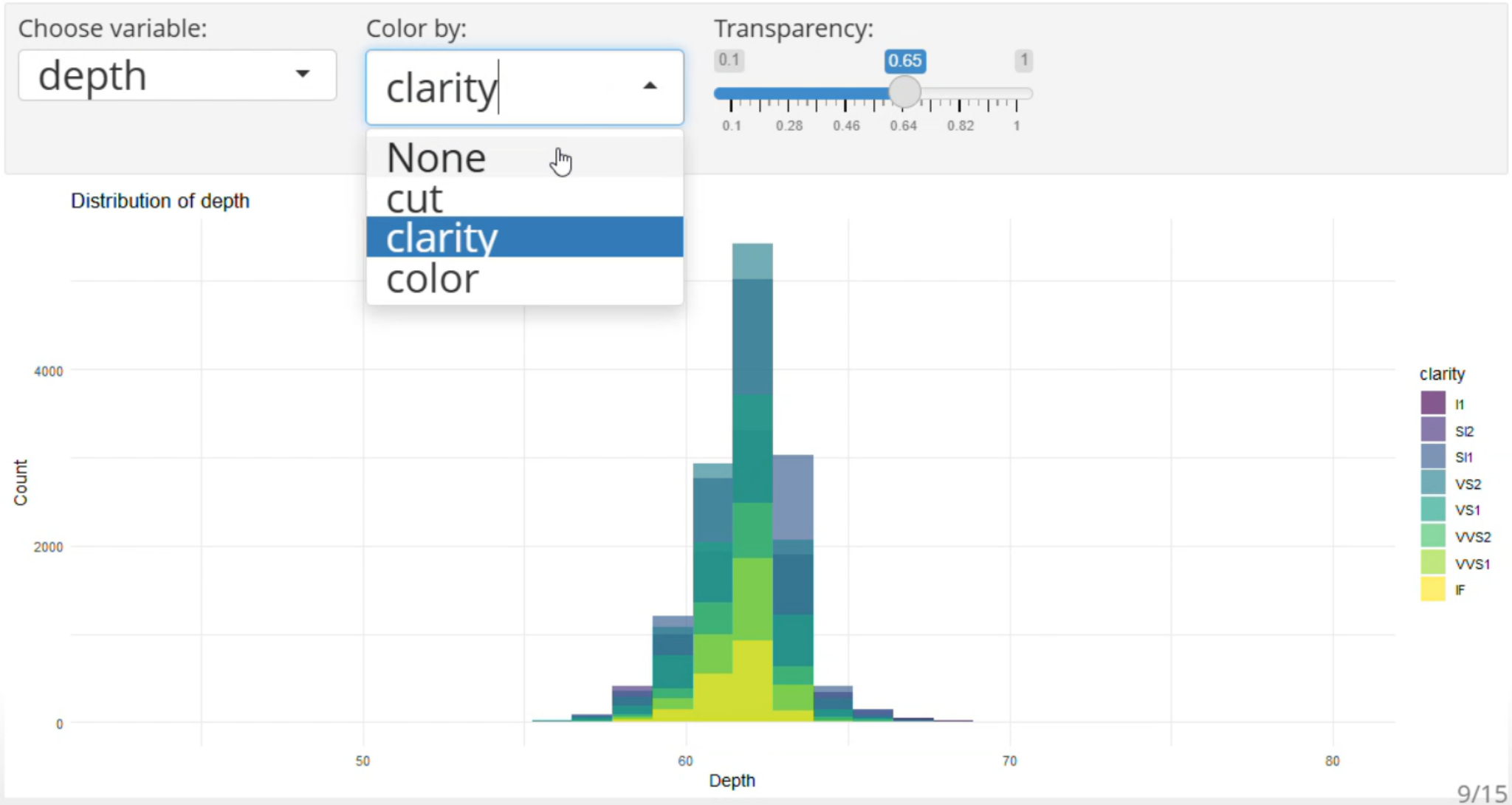
Jiaqi

2025-03-27

Interactive Example: Bin Width Effect



Interactive Exercise: Create Your Histogram

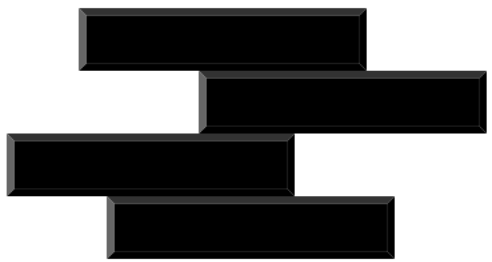


I'll draw swim plot in an oncology clinical trial. Give me a learning plan for ggplot2 in R and other things needed for a swim plot. Summarize in a mindmap.

Tips

1. **Build from the bottom, not as a whole**

Build from the bottom

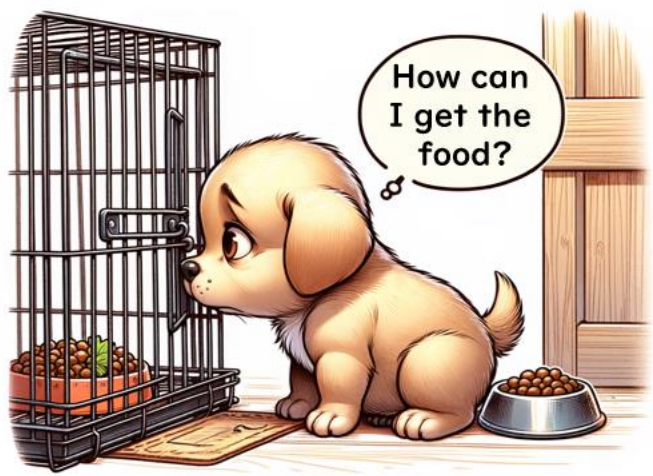


Build as a whole

Tips

1. **Build from the bottom, not as a whole**

2. **Start a new chat, when you notice the AI has reached a dead end**



Tips

1. **Build from the bottom, not as a whole**
2. **Start a new chat, when you notice the AI has reached a dead end**
3. **Try something that you think you can’t with the help of AI**
4. **Always follow the guideline of using AI in your company**

Thank you

If you have more questions, please contact:
Jiaqi Song
jsong67@jnj.com

Johnson&Johnson
Innovative Medicine

Q&A