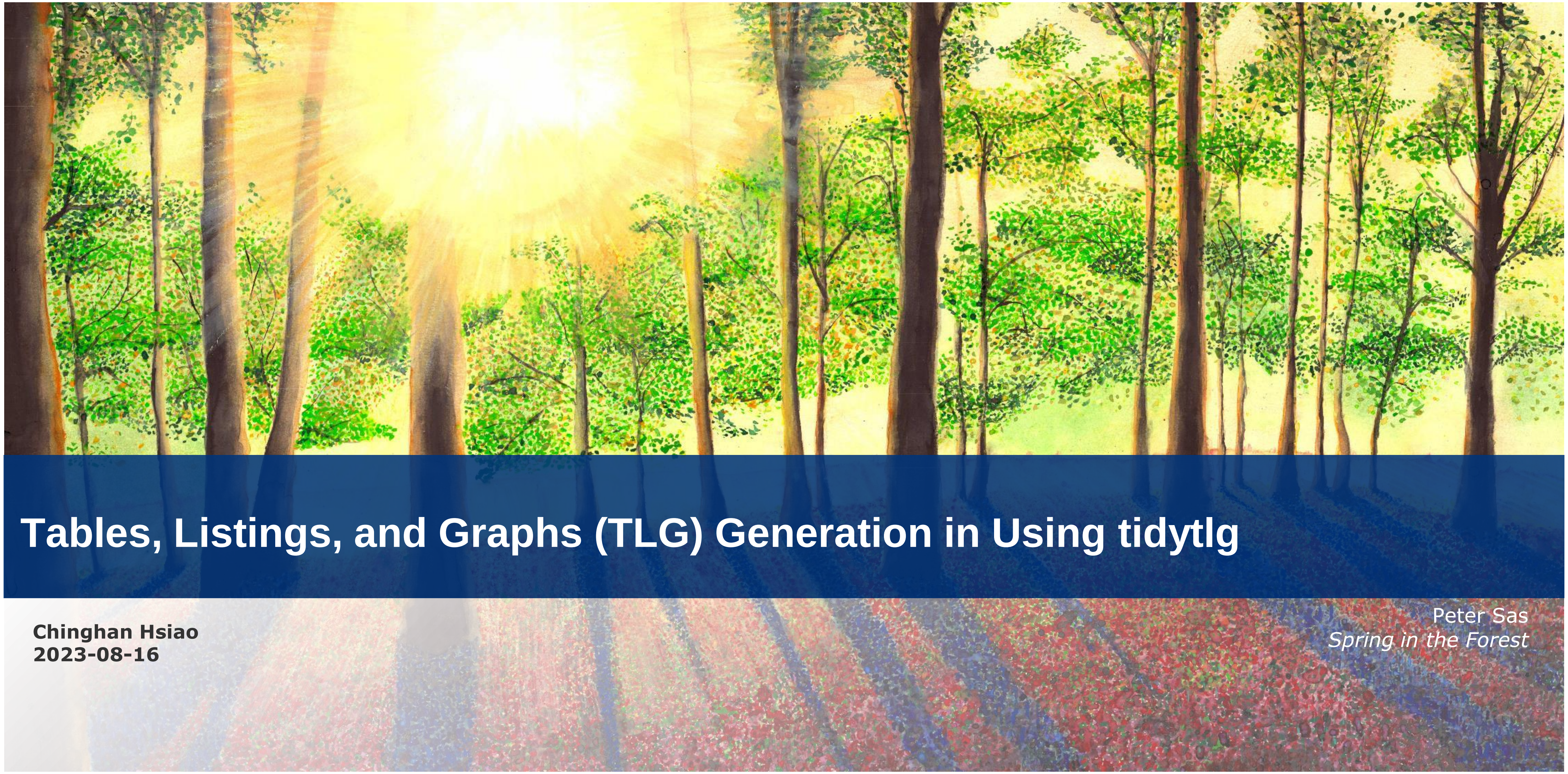


Tables, Listings, and Graphs (TLG) Generation in Using tidytlg

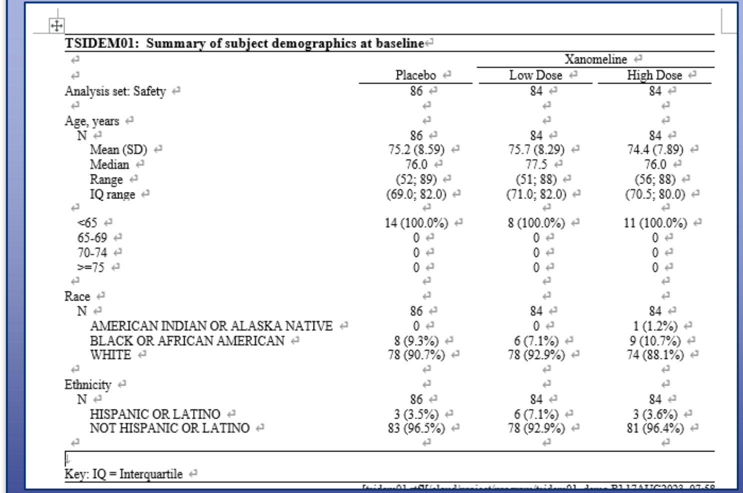
Chinghan Hsiao
2023-08-16

Disclaimer

Agenda

Introduction of tidytlg

Summarize data and generate tables, listings, and graphs (TLG) using the tidyverse suite of packages.

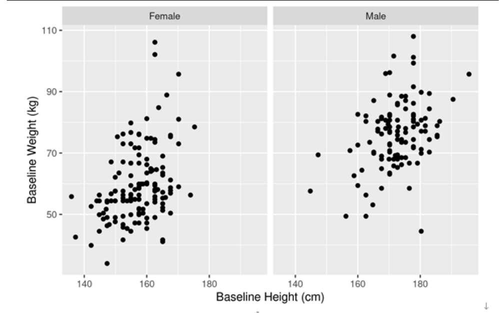


TSIDEM01: Summary of subject demographics at baseline

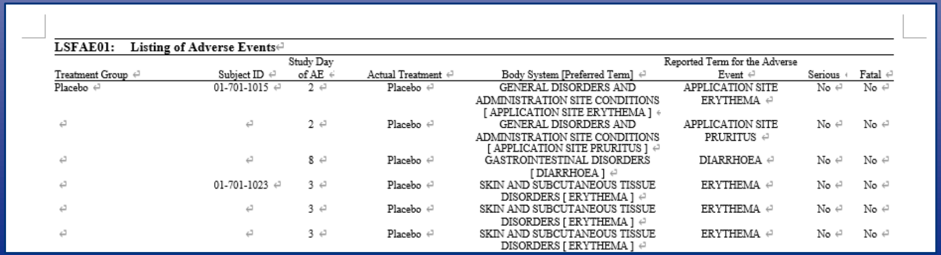
		Xanomeline	
	Placebo	Low Dose	High Dose
Analysis set: Safety	86	84	84
Age, years			
N	86	84	84
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)
Median	76.0	77.5	76.0
Range	(52; 89)	(51; 88)	(56; 88)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)
<65	14 (100.0%)	8 (100.0%)	11 (100.0%)
65-69	0	0	0
70-74	0	0	0
>=75	0	0	0
Race			
N	86	84	84
AMERICAN INDIAN OR ALASKA NATIVE	0	0	1 (1.2%)
BLACK OR AFRICAN AMERICAN	8 (9.3%)	6 (7.1%)	9 (10.7%)
WHITE	78 (90.7%)	78 (92.9%)	74 (88.1%)
Ethnicity			
N	86	84	84
HISPANIC OR LATINO	3 (3.5%)	6 (7.1%)	3 (3.6%)
NOT HISPANIC OR LATINO	83 (96.5%)	78 (92.9%)	81 (96.4%)

Key: IQ = Interquartile

GSIDEM01: Scatterplot of Baseline Weight and Height by Gender



[gsidem01.rtf][/cloud/project/program/gsidem01.R] 24AUG2023, 05:47



LSFAE01: Listing of Adverse Events

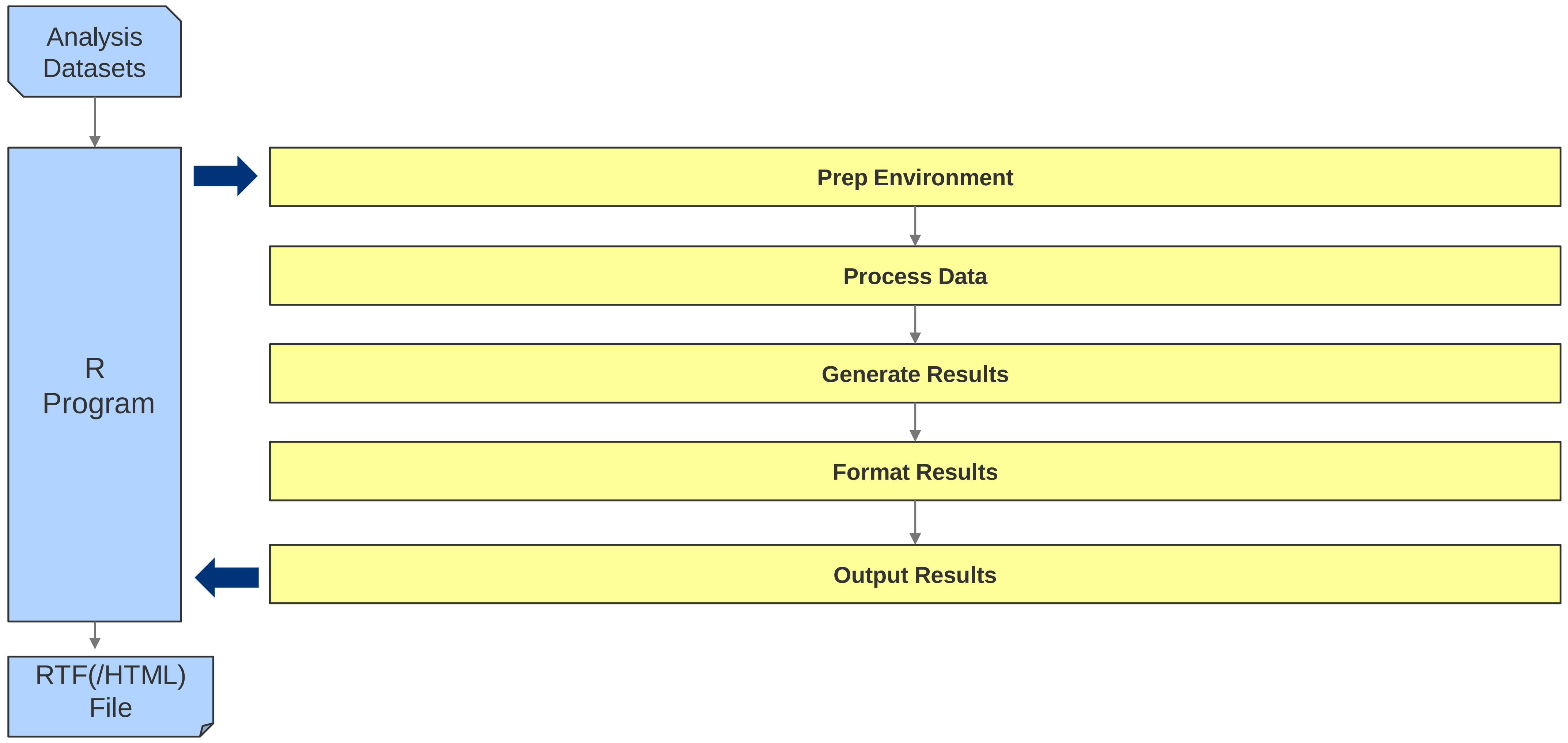
Treatment Group	Subject ID	Study Day of AE	Actual Treatment	Body System [Preferred Term]	Reported Term for the Adverse Event	Serious	Fatal
Placebo	01-701-1015	2	Placebo	GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS [APPLICATION SITE ERYTHEMA]	APPLICATION SITE ERYTHEMA	No	No
		2	Placebo	GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS [APPLICATION SITE PRURITUS]	APPLICATION SITE PRURITUS	No	No
		8	Placebo	GASTROINTESTINAL DISORDERS [DIARRHOEA]	DIARRHOEA	No	No
	01-701-1023	3	Placebo	SKIN AND SUBCUTANEOUS TISSUE DISORDERS [ERYTHEMA]	ERYTHEMA	No	No
		3	Placebo	SKIN AND SUBCUTANEOUS TISSUE DISORDERS [ERYTHEMA]	ERYTHEMA	No	No
		3	Placebo	SKIN AND SUBCUTANEOUS TISSUE DISORDERS [ERYTHEMA]	ERYTHEMA	No	No

Approaches (tables) :

– Functional method: build the summary table, one summary function call at a time.

– Metadata method: define the column and analysis metadata needed to produce the summary table.

Processing Flow



Helper Functions

The package contains several helper functions to aid the creation of TLG output:

- Define column variables and metadata (Metadata method): **tlgsetup()**
- Producing tabular summaries: **univar(), freq(), nested_freq()**
- Formatting tabular summaries: **bind_table()**
- Output your tabular summaries: **gentlg()**

Table Programming

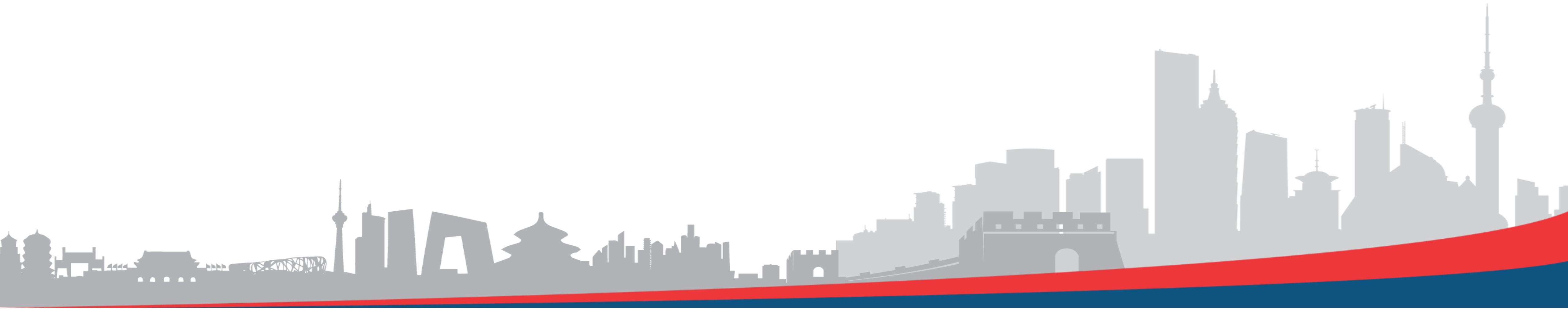
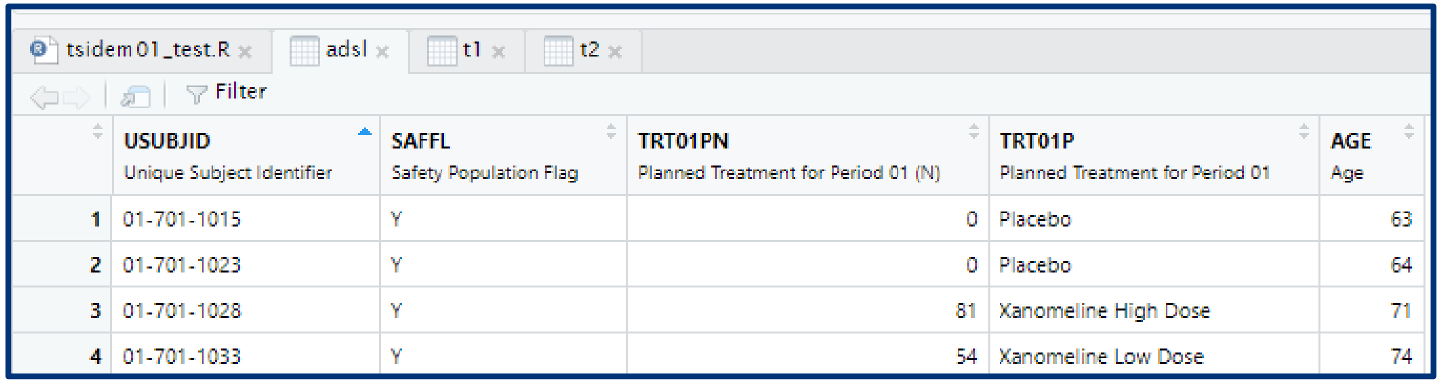


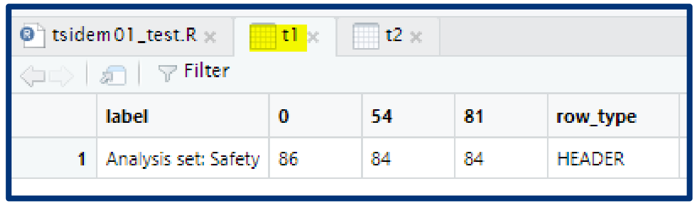
Table Programming - Functional method

```
# Prep Environment
library(dplyr)
library(haven)
library(tidytlg)

a_in <- "/cloud/project/data“              #datasets path
output <- "/cloud/project/output”     #outputs path

# Process Data
adsl <- read_sas(file.path(a_in, "adsl.sas7bdat")) %>%
  filter(SAFFL == "Y") %>%
  select(USUBJID, SAFFL, TRT01PN, TRT01P, AGE)
```



	USUBJID Unique Subject Identifier	SAFFL Safety Population Flag	TRT01PN Planned Treatment for Period 01 (N)	TRT01P Planned Treatment for Period 01	AGE Age
1	01-701-1015	Y	0	Placebo	63
2	01-701-1023	Y	0	Placebo	64
3	01-701-1028	Y	81	Xanomeline High Dose	71
4	01-701-1033	Y	54	Xanomeline Low Dose	74

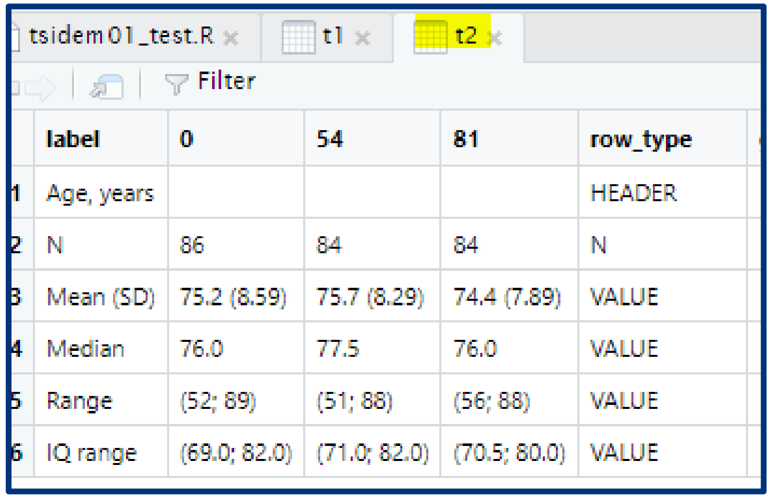
```r
# Generate Results

t1 <- freq(df        = adsl,
           colvar    = "TRT01PN",
           rowvar    = "SAFFL",
           statlist = statlist("n"),
          rowtext   = "Analysis set: Safety")

t2 <- univar(df          = adsl,
             colvar      = "TRT01PN",
             rowvar      = "AGE",
             decimal     = 0,
             statlist = statlist (c("N", "MEANSD", "MEDIAN",
"RANGE", "IQRANGE")),
             row_header = "Age, years")

# Format Results

tbl <- bind_table(t1, t2)
```



tsidem01_test.R | t1 | t2

Filter

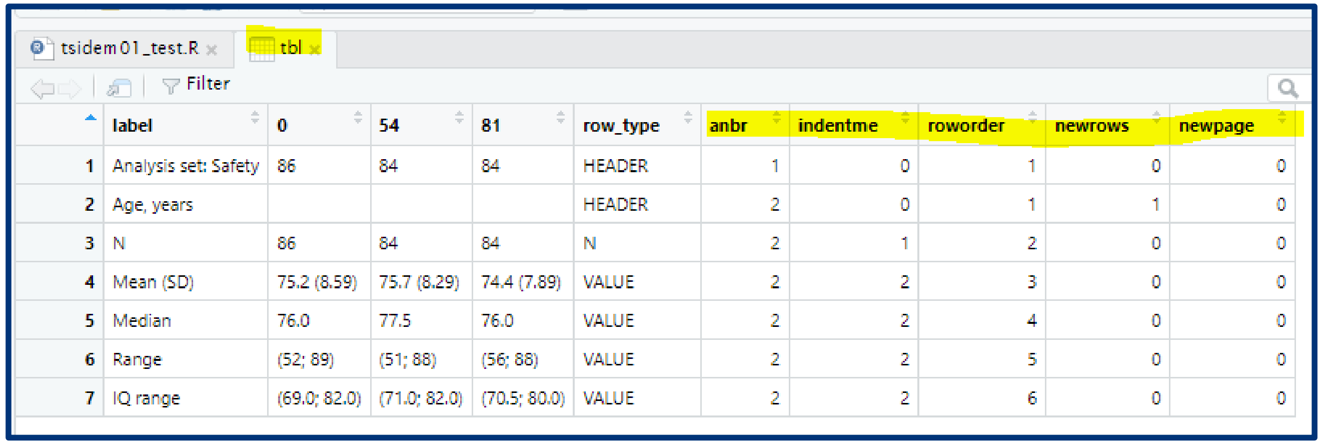
	label	0	54	81	row_type
1	Analysis set: Safety	86	84	84	HEADER



tsidem01_test.R | t1 | t2

Filter

	label	0	54	81	row_type
1	Age, years				HEADER
2	N	86	84	84	N
3	Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	VALUE
4	Median	76.0	77.5	76.0	VALUE
5	Range	(52; 89)	(51; 88)	(56; 88)	VALUE
6	IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	VALUE

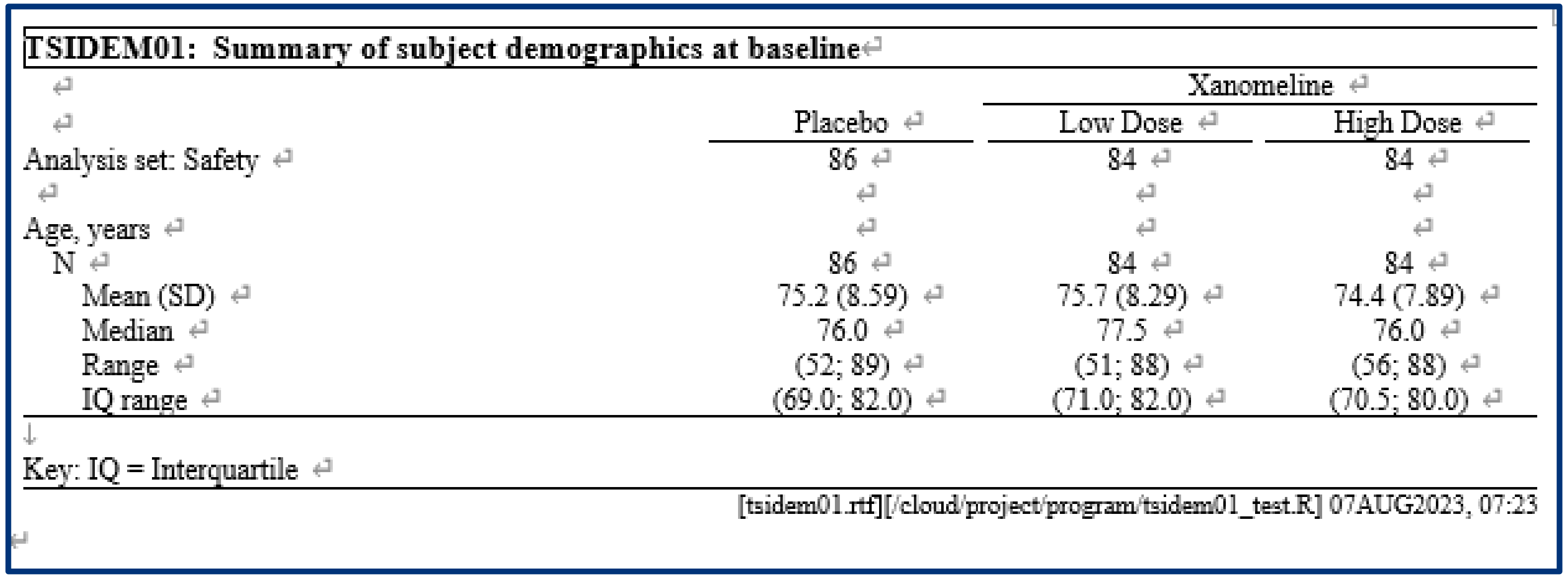


tsidem01_test.R | tbl

Filter

	label	0	54	81	row_type	anbr	indentme	roworder	newrows	newpage
1	Analysis set: Safety	86	84	84	HEADER	1	0	1	0	0
2	Age, years				HEADER	2	0	1	1	0
3	N	86	84	84	N	2	1	2	0	0
4	Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	VALUE	2	2	3	0	0
5	Median	76.0	77.5	76.0	VALUE	2	2	4	0	0
6	Range	(52; 89)	(51; 88)	(56; 88)	VALUE	2	2	5	0	0
7	IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	VALUE	2	2	6	0	0

```
# Output Results

gentlg(huxme        = tbl,
          file         = "TSIDEM01",
          title        = "Summary of subject
   demographics at baseline",
          footers      = "Key: IQ = Interquartile",
          colheader    = c(" ", "Placebo", "Low
   Dose",   "High Dose"),
          colspan      = list(c(" ", " ",
   "Xanomeline", "Xanomeline")),
          orientation = "portrait",
          opath = output)
```



TSIDEM01: Summary of subject demographics at baseline

		Xanomeline	
	Placebo	Low Dose	High Dose
Analysis set: Safety	86	84	84
Age, years			
N	86	84	84
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)
Median	76.0	77.5	76.0
Range	(52; 89)	(51; 88)	(56; 88)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)

Key: IQ = Interquartile

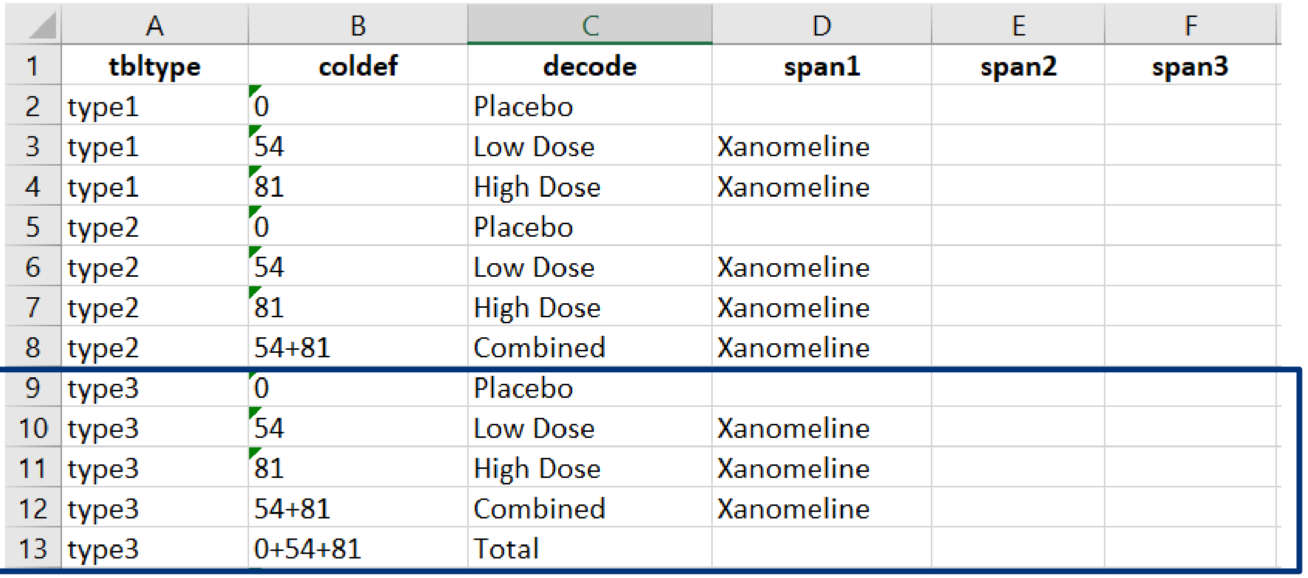
[tsidem01.rtf][/cloud/project/program/tsidem01_test.R] 07AUG2023, 07:23

Table Programming - Metadata method

This package also provides a function for a more efficient/effective mapping data to table columns (including the creation of combined & total columns):

tlgsetup() : maps the values of an existing variable (e.g., TRT01PN) to column variables (col1-coln) and associates those columns with column headers, etc.

with column_metadata.xlsx, provides the column structure of the table layout and includes the following variables.



	A	B	C	D	E	F
1	tbltype	coldef	decode	span1	span2	span3
2	type1	0	Placebo			
3	type1	54	Low Dose	Xanomeline		
4	type1	81	High Dose	Xanomeline		
5	type2	0	Placebo			
6	type2	54	Low Dose	Xanomeline		
7	type2	81	High Dose	Xanomeline		
8	type2	54+81	Combined	Xanomeline		
9	type3	0	Placebo			
10	type3	54	Low Dose	Xanomeline		
11	type3	81	High Dose	Xanomeline		
12	type3	54+81	Combined	Xanomeline		
13	type3	0+54+81	Total			

•**tbltype:** identifier used to group a table column layout

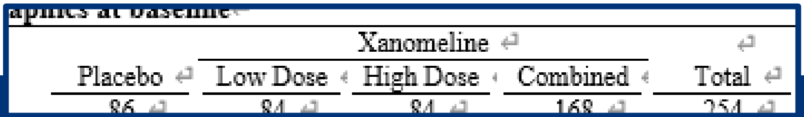
•**coldef:** distinct variable values used, typically numeric and typically a treatment variable(TRT01PN)

•**decode:** decode of coldef that will display as a column header in the table

•**span1:** spanning header to display across multiple columns (the lowest level)

•**span2:** spanning header to display across multiple columns, second level

•**span3:** spanning header to display across multiple columns, third level



aphics at baseline

	Xanomeline			
Placebo	Low Dose	High Dose	Combined	Total
86	84	84	168	254

Column Setup:

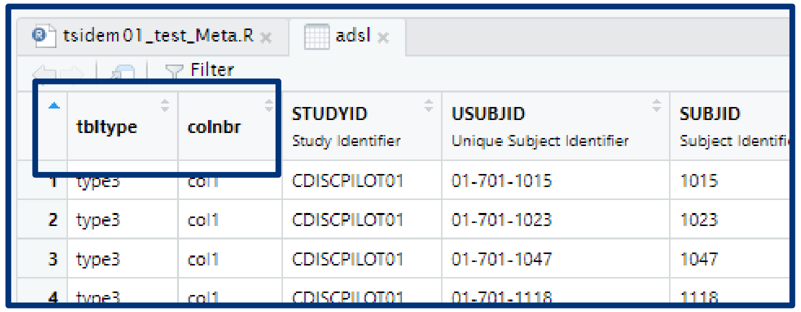
1) Create below column_metadata.xlsx in **input** folder

```
# Prep Environment
library(dplyr)
library(haven)
library(tidytlg)

a_in <- "/cloud/project/data“
#datasets path
output <- "/cloud/project/output”
#outputs path
input <- "/cloud/project/input"
#file path
column_metadata_file <-
file.path(input,"column_metadata.xlsx")
```

2) Call tl[illegible]les to represent table columns associated with the tbltype, by adding two variables (tbltype and colnbr) in the dataset.

```
# Process Data
adsl <- read_sas(file.path(a_in,
"adsl.sas7bdat")) %>%
             filter(SAFFL == "Y")
```

```
# Process Data
adsl <- read_sas(file.path(a_in,
"adsl.sas7bdat")) %>%
           tlgsetup(column_metadata_file =
column_metadata_file,

var                   = "TRT01PN",

tbltype               = "type3"  ) %>%
             filter(SAFFL == "Y")
```


tsidem01_test_Meta.R adsl

	tbltype	colnbr	STUDYID Study Identifier	USUBJID Unique Subject Identifier	SUBJID Subject Identifi
1	type3	col1	CDISCPILOT01	01-701-1015	1015
2	type3	col1	CDISCPILOT01	01-701-1023	1023
3	type3	col1	CDISCPILOT01	01-701-1047	1047
4	type3	col1	CDISCPILOT01	01-701-1118	1118

3) Update colvar to “colnbr”

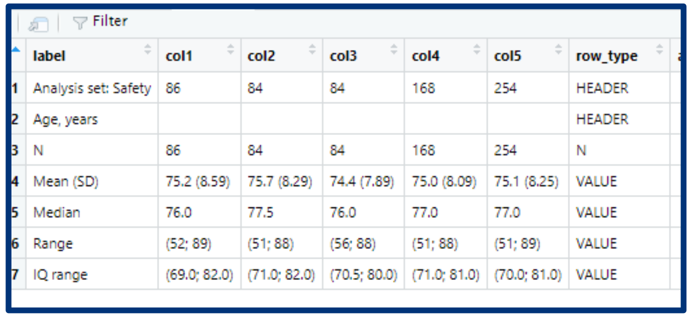
```
# Generate Results

t1 <- freq(df       = adsl,
           colvar   = "TRT01PN",
           rowvar   = "SAFFL",
           statlist = statlist("n"),
          rowtext  = "Analysis set: Safety")

t2 <- univar(df           = adsl,
             colvar       = "TRT01PN",
             rowvar       = "AGE",
             decimal      = 0,
             statlist = statlist (c("N", "MEANSD",
"MEDIAN", "RANGE", "IQRANGE")),
             row_header = "Age, years")
```

```
# Generate Results

t1 <- freq(df       = adsl,
           colvar   = "colnbr“,
           rowvar   = "SAFFL",
           statlist = statlist("n"),
          rowtext  = "Analysis set: Safety")

t2 <- univar(df           = adsl,
             colvar       = "colnbr“,
             rowvar       = "AGE",
             decimal      = 0,
             statlist = statlist (c("N", "MEANSD",
"MEDIAN", "RANGE", "IQRANGE")),
             row_header = "Age, years")
```

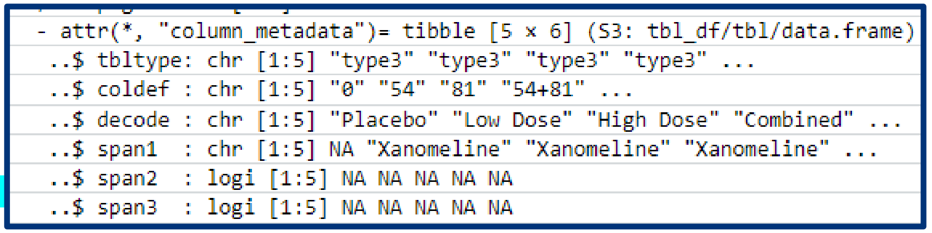
4) Specify the tbltype in the **bind_table()** call

```
# Format Results

tbl <- bind_table(t1, t2)
```

```
# Format Results

tbl <- bind_table(t1, t2,

column_metadata_file = column_metadata_file,
                                 tbltype
=     "type3")
```



	label	col1	col2	col3	col4	col5	row_type
1	Analysis set: Safety	86	84	84	168	254	HEADER
2	Age, years						HEADER
3	N	86	84	84	168	254	N
4	Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)	VALUE
5	Median	76.0	77.5	76.0	77.0	77.0	VALUE
6	Range	(52; 89)	(51; 88)	(56; 88)	(51; 88)	(51; 89)	VALUE
7	IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)	VALUE

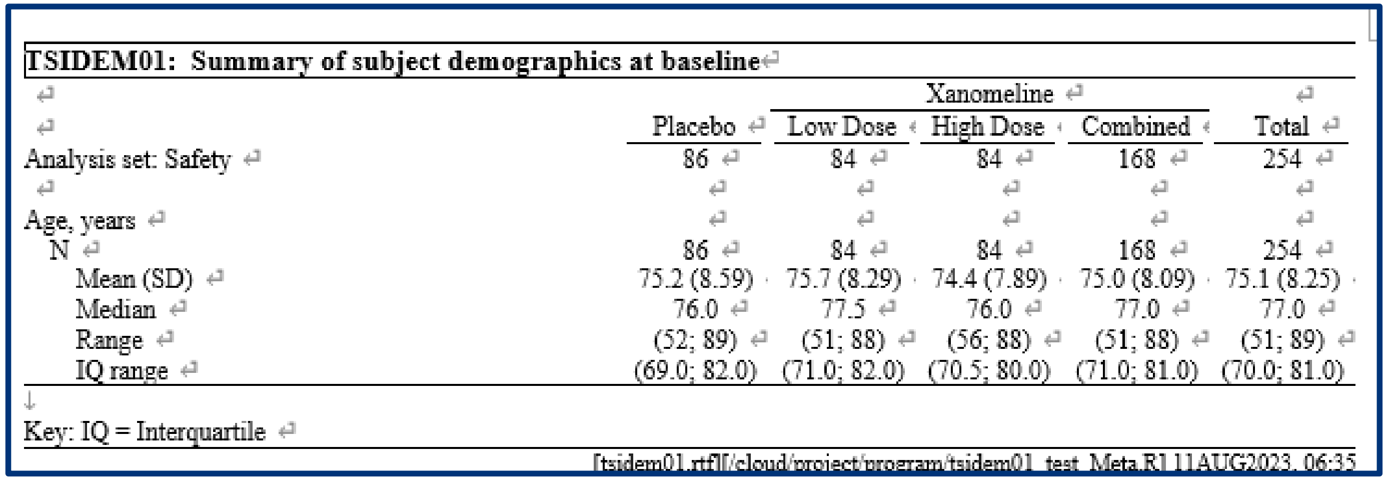


```
 - attr(*, "column_metadata")= tibble [5 x 6] (S3: tbl_df/tbl/data.frame)
  ..$ tbltype: chr [1:5] "type3" "type3" "type3" "type3" ...
  ..$ coldef : chr [1:5] "0" "54" "81" "54+81" ...
  ..$ decode : chr [1:5] "Placebo" "Low Dose" "High Dose" "Combined" ...
  ..$ span1  : chr [1:5] NA "Xanomeline" "Xanomeline" "Xanomeline" ...
  ..$ span2  : logi [1:5] NA NA NA NA NA
  ..$ span3  : logi [1:5] NA NA NA NA NA
```

5) Output results, remove colheader and colspan

```
# Output Results

gentlg(huxme       = tbl,
          file        = "TSIDEM01",
          title       = "Summary of subject
   demographics at baseline",
          footers     = "Key: IQ =
   Interquartile",
          colheader   = c(" ", "Placebo", "Low
   Dose",    "High Dose"),
          colspan     = list(c(" ", " ",
   "Xanomeline", "Xanomeline")),
          orientation = "portrait",
          opath = output)
```

```
# Output Results

gentlg(huxme                = tbl,
          file                   = "TSIDEM01",
          title                  = "Summary of
   subject demographics at baseline",
          footers            = "Key: IQ =
   Interquartile",
          orientation = "portrait",
          opath              = output)
```



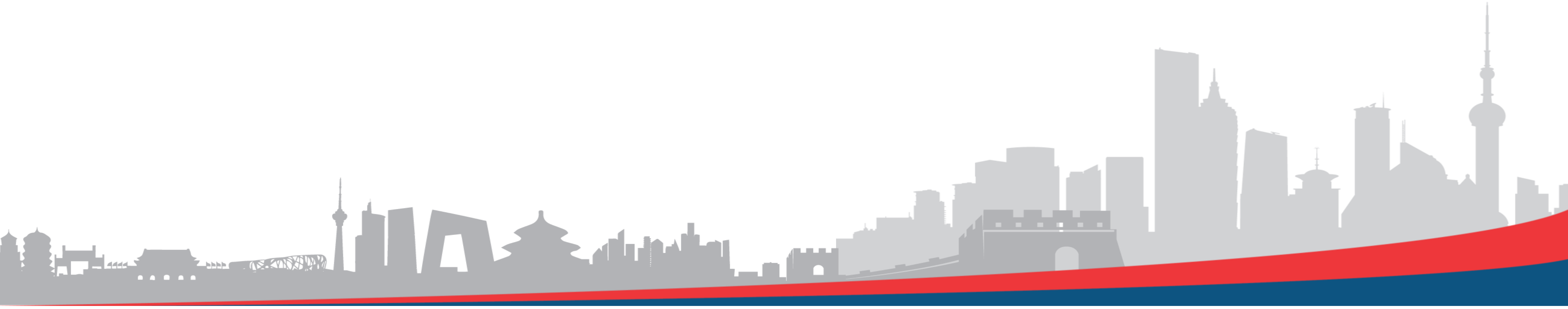
TSIDEM01: Summary of subject demographics at baseline

		Xanomeline			
	Placebo	Low Dose	High Dose	Combined	Total
Analysis set: Safety	86	84	84	168	254
Age, years					
N	86	84	84	168	254
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)
Median	76.0	77.5	76.0	77.0	77.0
Range	(52; 89)	(51; 88)	(56; 88)	(51; 88)	(51; 89)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)

Key: IQ = Interquartile

[tsidem01.rtf][/cloud/project/program/tsidem01_test_Meta.R] 11AUG2023, 06:35

Listing & Figure Programming



Listing Programming

Prep Environment

↓

Process Data

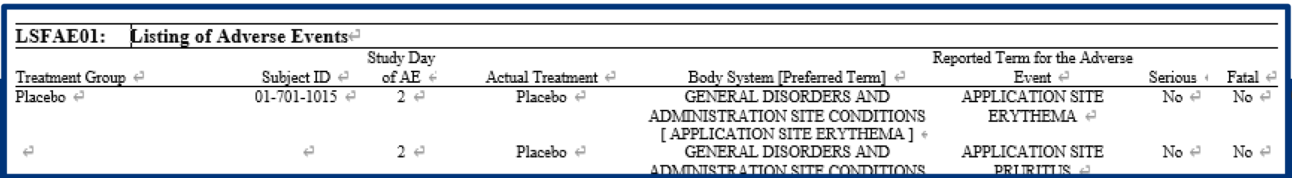
↓

Format Results:
- Create TBL dataframe
- Sort the dataframe
- Select variables to include

↓

Output Results:
Output TBL file to RTF.

```
# Prep Environment ----------------------------------------------------------------------------------

library(haven)
library(dplyr)
library(tidytlg)

# Process Data ---------------------------------------------------------------------------------------

adsl <- read_sas(file.path(a_in, "adsl.sas7bdat")) %>%
  filter(SAFFL == "Y") %>%
  select(USUBJID, SAFFL, TRT01AN, TRT01A)

adae <- read_sas(file.path(a_in, "adae.sas7bdat")) %>%
  filter(SAFFL == "Y" & TRTEMFL == "Y") %>%
  mutate(BSPT  = paste(AEBODSYS, "[", AEDECOD, "]"),
         SAEFL = if_else(AESER == "Y", "Yes", "No"),
         DTHFL = if_else(AEOUT == "FATAL", "Yes", "No")) %>%
   select(USUBJID, ASTDY, TRTA, BSPT, AETERM, SAEFL, DTHFL)

# Format Results -------------------------------------------------------------------------------------

tbl <- inner_join(adsl, adae, by = "USUBJID") %>%
  arrange(TRT01AN, USUBJID, ASTDY) %>%
  select(TRT01A, USUBJID, ASTDY, TRTA, BSPT, AETERM, SAEFL, DTHFL)

attr(tbl$TRT01A, 'label') <- "Treatment Group"
attr(tbl$USUBJID, 'label') <- "Subject ID"
attr(tbl$ASTDY, 'label') <- "Study Day of AE"
attr(tbl$BSPT, 'label') <- "Body System [Preferred Term]"
attr(tbl$SAEFL, 'label') <- "Serious"
attr(tbl$DTHFL, 'label') <- "Fatal"

# Output Results -------------------------------------------------------------------------------------

gentlg(huxme       = tbl,
       tlf         = "l",
       format      = "rtf",
       orientation = "landscape",
                                                                        0, 0.15, 0.05, 0.05))
```



LSFAE01: Listing of Adverse Events

Treatment Group	Subject ID	Study Day of AE	Actual Treatment	Body System [Preferred Term]	Reported Term for the Adverse Event	Serious	Fatal
Placebo	01-701-1015	2	Placebo	GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS [APPLICATION SITE ERYTHEMA]	APPLICATION SITE ERYTHEMA	No	No
		2	Placebo	GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS	APPLICATION SITE PRURITUS	No	No

Figure Programming

Prep Environment

↓

Process Data

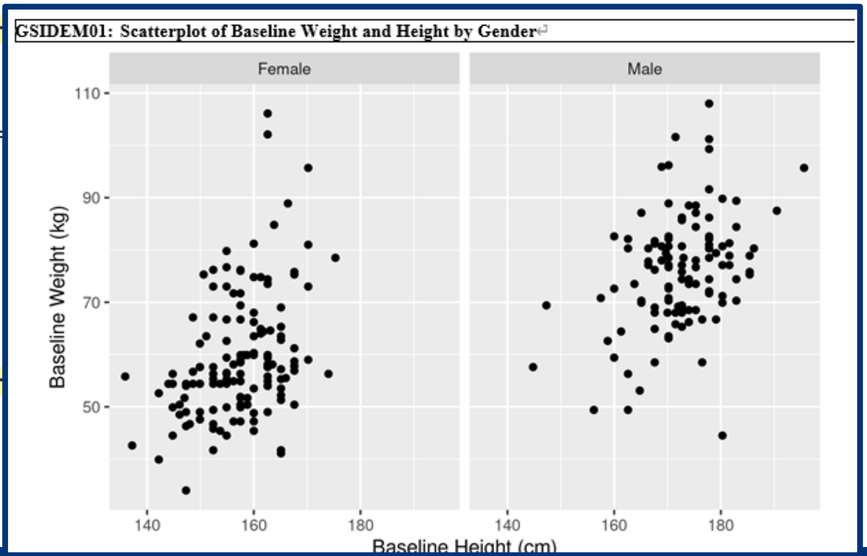
↓

Generate Results:
- Call ggplot()

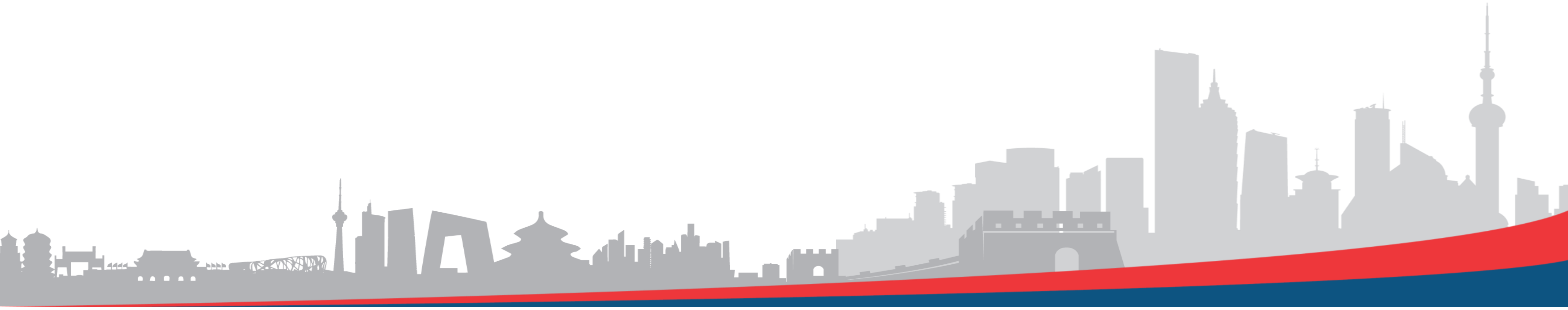
↓

Output Results:
Each call returns the requested analysis rows for the table output.

```
# Prep Environment ---------------------------------------
-----------------------------------------------

library(dplyr)
library(haven)
library(ggplot2)
library(tidytlg)

# Process Data -------------------------------------------
------------------------------------------------

adsl <- read_sas(file.path(a_in, "adsl.sas7bdat")) %>%
  filter(ITTFL == "Y") %>%
  select(USUBJID, ITTFL, TRT01PN, TRT01P, AGE, SEX,
HEIGHTBL, WEIGHTBL) %>%
  mutate(SEX = factor(SEX, c("F", "M", "U", "UNK"),
c("Female", "Male", "Undifferentiated", "Unknown")))

# Generate Results -------------------------------
------------------------------------------------

plot <- ggplot(data = adsl, aes(x = HEIGHTBL, y =
WEIGHTBL)) +
  geom_point() +
  labs(x = "Baseline Height (cm)",
       y = "Baseline Weight (kg)") +
  facet_wrap(~SEX, nrow=1)

# Output Results --------------------------------
------------------------------------------------

gentlg(huxme     = plot,
       tlf       = "Graph",
       plotwidth  = 6,
       plotheight = 4,
       file      = "GSIDEM01",
       title     = "Scatterplot of Baseline Weight and
```

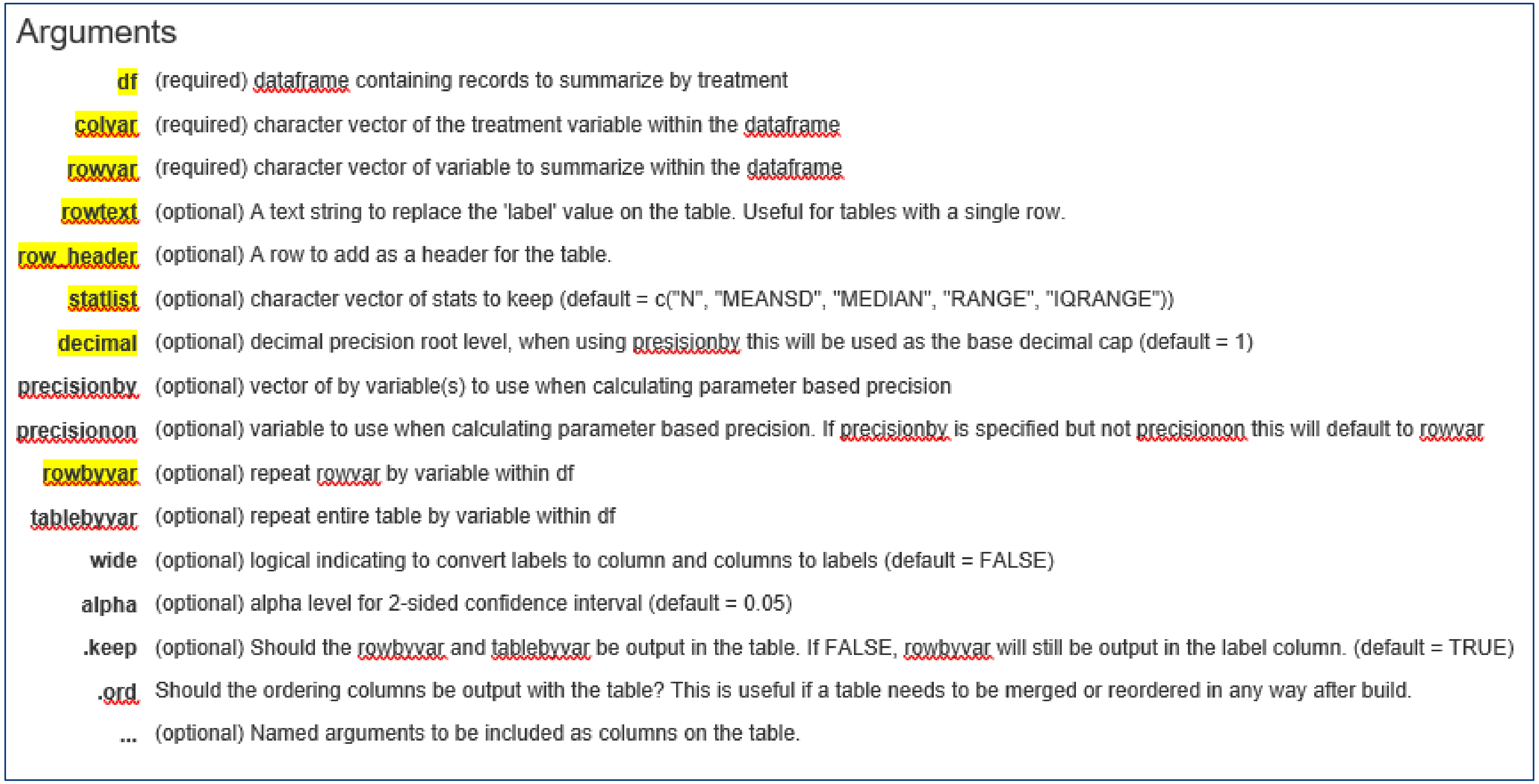

GSIDEM01: Scatterplot of Baseline Weight and Height by Gender

Functions Deep Dive



Deep Dive #1: univar()

Univariate statistics for a variable within the input data frame by treatment and/or group.

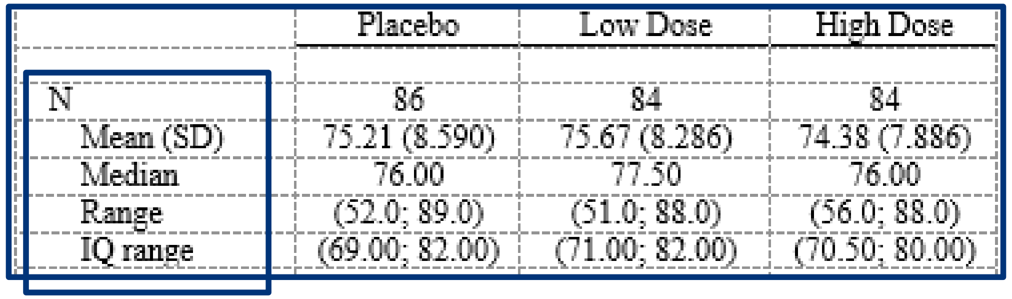


Arguments

df	(required) dataframe containing records to summarize by treatment
colvar	(required) character vector of the treatment variable within the dataframe
rowvar	(required) character vector of variable to summarize within the dataframe
rowtext	(optional) A text string to replace the 'label' value on the table. Useful for tables with a single row.
row_header	(optional) A row to add as a header for the table.
statlist	(optional) character vector of stats to keep (default = c("N", "MEANSD", "MEDIAN", "RANGE", "IQRANGE"))
decimal	(optional) decimal precision root level, when using presisionby this will be used as the base decimal cap (default = 1)
precisionby	(optional) vector of by variable(s) to use when calculating parameter based precision
precisionon	(optional) variable to use when calculating parameter based precision. If precisionby is specified but not precisionon this will default to rowvar
rowbyvar	(optional) repeat rowvar by variable within df
tablebyvar	(optional) repeat entire table by variable within df
wide	(optional) logical indicating to convert labels to column and columns to labels (default = FALSE)
alpha	(optional) alpha level for 2-sided confidence interval (default = 0.05)
.keep	(optional) Should the rowbyvar and tablebyvar be output in the table. If FALSE, rowbyvar will still be output in the label column. (default = TRUE)
.ord	Should the ordering columns be output with the table? This is useful if a table needs to be merged or reordered in any way after build.
...	(optional) Named arguments to be included as columns on the table.

Calling **univar()** with the core parameters:

```
#Examples (default):
ex1 <- univar(df        =
adsl,
              colvar    =
"colnbr",
              rowvar    =
"AGE" )
```



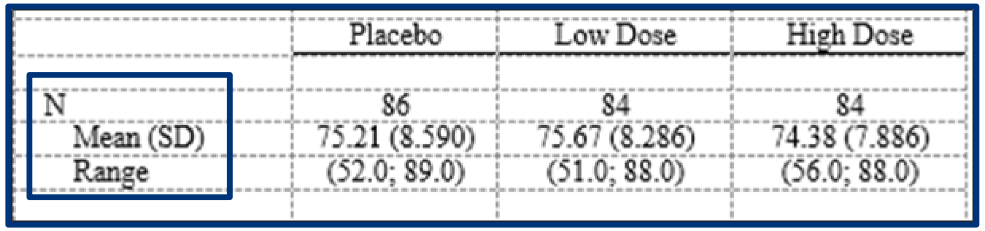
	Placebo	Low Dose	High Dose
N	86	84	84
Mean (SD)	75.21 (8.590)	75.67 (8.286)	74.38 (7.886)
Median	76.00	77.50	76.00
Range	(52.0; 89.0)	(51.0; 88.0)	(56.0; 88.0)
IQ range	(69.00; 82.00)	(71.00; 82.00)	(70.50; 80.00)

- **Statlist** - specifying the statistics to display:

Available statistics to pass to statlist:

•N •SUM •MEAN •GeoMEAN •SD •SE •CV	•GSD •GSE •MEANSD •MEANSE •MEDIAN •MIN •MAX	•RANGE •Q1 •Q3 •IQRANGE •MEDRANGE •MEDIQRANGE •MEAN_CI •GeoMEAN_CI

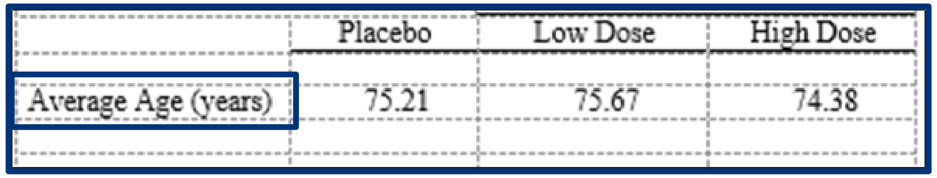
```
#Examples:
ex1b <- univar(df        = adsl,
               colvar    = "colnbr",
               rowvar    = "AGE" ,
               statlist  =
statlist(c("N","MEANSD","RANGE")))
```



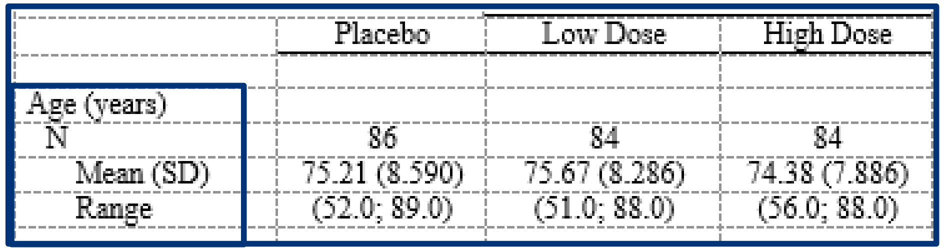
	Placebo	Low Dose	High Dose
N	86	84	84
Mean (SD)	75.21 (8.590)	75.67 (8.286)	74.38 (7.886)
Range	(52.0; 89.0)	(51.0; 88.0)	(56.0; 88.0)

- **Rowtext/row_header** - specifying a label for a row, or a header for a group of rows:

```
#Examples:
ex1c <- univar(df          = adsl,
               colvar      = ″colnbr″,
               rowvar      = ″AGE″ ,
               rowtext     =  “Average Age
(Years)” ,
               statlist    =
statlist(c(″MEAN″)))

ex1d <- univar(df          = adsl,
               colvar      = ″colnbr″,
               rowvar      = ″AGE″ ,
               row_header =  “Age (Years)” ,
               statlist    =
statlist(c(″N″,″MEANSD″,″RANGE″)))
```



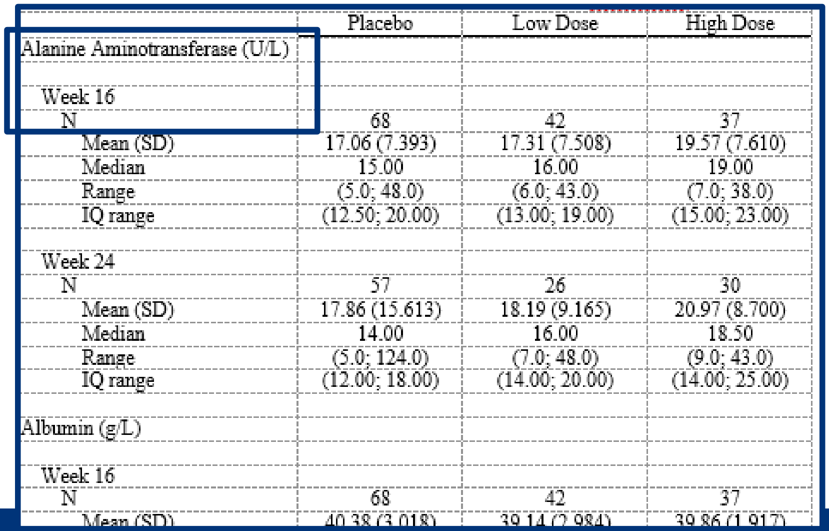
	Placebo	Low Dose	High Dose
Average Age (years)	75.21	75.67	74.38



	Placebo	Low Dose	High Dose
Age (years)			
N	86	84	84
Mean (SD)	75.21 (8.590)	75.67 (8.286)	74.38 (7.886)
Range	(52.0; 89.0)	(51.0; 88.0)	(56.0; 88.0)

- **Rowbyvar** - by-variables for summaries :

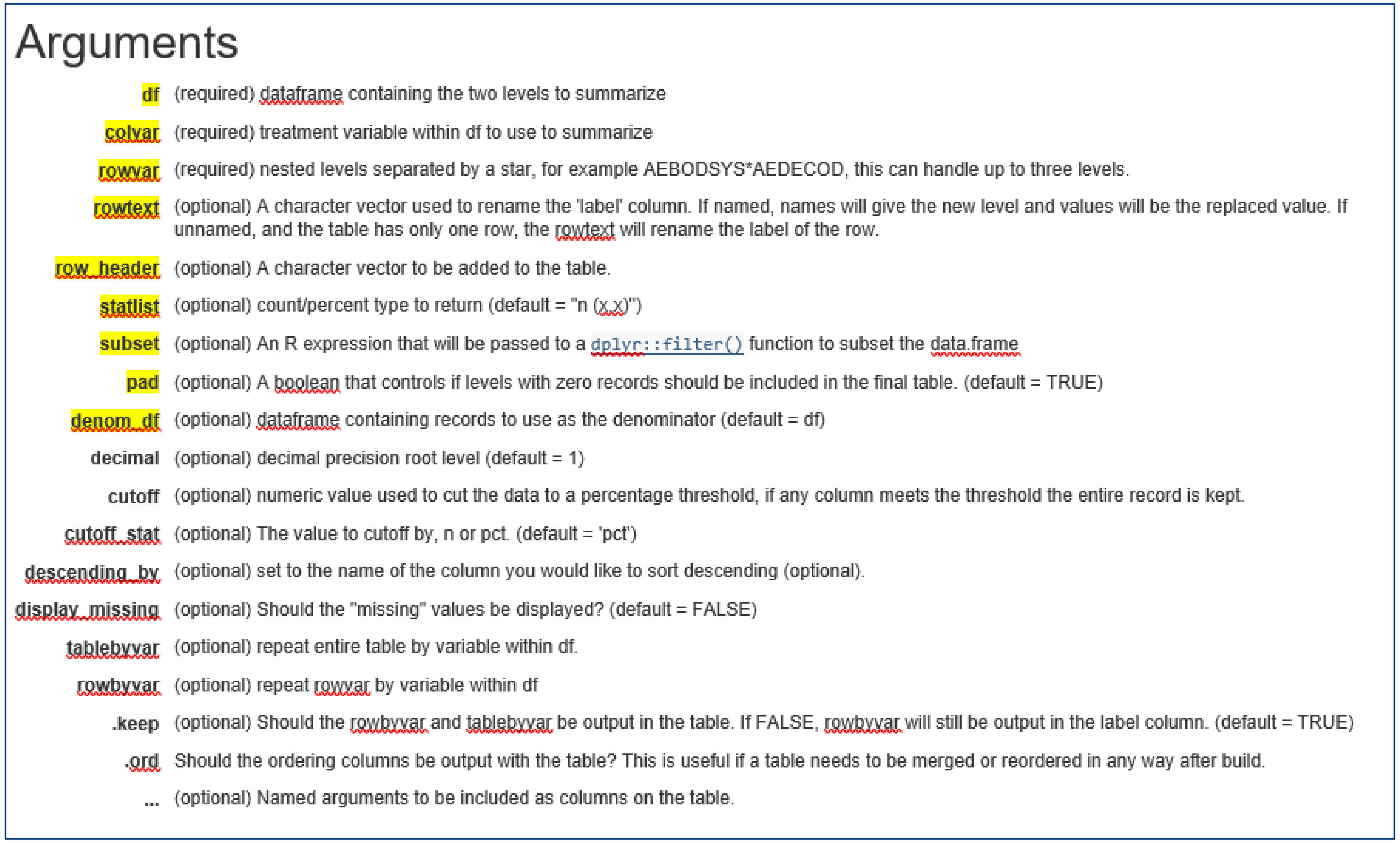
```
#Examples:
ex3 <- univar(df        = adlb,
              colvar    = ″colnbr″,
              rowvar    = ″AVAL″,
              rowbyvar =
c(″PARAM″,″AVISIT″))
```



	Placebo	Low Dose	High Dose
Alanine Aminotransferase (U/L)			
Week 16			
N	68	42	37
Mean (SD)	17.06 (7.393)	17.31 (7.508)	19.57 (7.610)
Median	15.00	16.00	19.00
Range	(5.0; 48.0)	(6.0; 43.0)	(7.0; 38.0)
IQ range	(12.50; 20.00)	(13.00; 19.00)	(15.00; 23.00)
Week 24			
N	57	26	30
Mean (SD)	17.86 (15.613)	18.19 (9.165)	20.97 (8.700)
Median	14.00	16.00	18.50
Range	(5.0; 124.0)	(7.0; 48.0)	(9.0; 43.0)
IQ range	(12.00; 18.00)	(14.00; 20.00)	(14.00; 25.00)
Albumin (g/L)			
Week 16			
N	68	42	37
Mean (SD)	40.38 (3.018)	39.14 (2.984)	39.86 (1.917)

Deep Dive #2: freq()

Frequency counts and percentages for a variable within the input data frame by treatment and/or group.

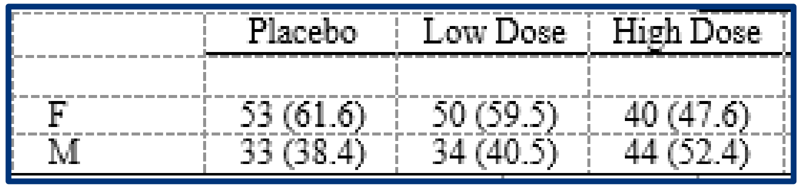


Arguments

df	(required) dataframe containing the two levels to summarize
colvar	(required) treatment variable within df to use to summarize
rowvar	(required) nested levels separated by a star, for example AEBODSYS*AEDECOD, this can handle up to three levels.
rowtext	(optional) A character vector used to rename the 'label' column. If named, names will give the new level and values will be the replaced value. If unnamed, and the table has only one row, the rowtext will rename the label of the row.
row_header	(optional) A character vector to be added to the table.
statlist	(optional) count/percent type to return (default = "n (x.x)")
subset	(optional) An R expression that will be passed to a `dplyr::filter()` function to subset the data.frame
pad	(optional) A boolean that controls if levels with zero records should be included in the final table. (default = TRUE)
denom_df	(optional) dataframe containing records to use as the denominator (default = df)
decimal	(optional) decimal precision root level (default = 1)
cutoff	(optional) numeric value used to cut the data to a percentage threshold, if any column meets the threshold the entire record is kept.
cutoff_stat	(optional) The value to cutoff by, n or pct. (default = 'pct')
descending_by	(optional) set to the name of the column you would like to sort descending (optional).
display_missing	(optional) Should the "missing" values be displayed? (default = FALSE)
tablebyvar	(optional) repeat entire table by variable within df.
rowbyvar	(optional) repeat rowvar by variable within df
.keep	(optional) Should the rowbyvar and tablebyvar be output in the table. If FALSE, rowbyvar will still be output in the label column. (default = TRUE)
.ord	Should the ordering columns be output with the table? This is useful if a table needs to be merged or reordered in any way after build.
...	(optional) Named arguments to be included as columns on the table.

Calling **freq()** with the core parameters:

```
#Examples:
ex4 <- freq(df
= adsl,
            colvar
= "colnbr",
            rowvar
= “SEX” )
```



	Placebo	Low Dose	High Dose
F	53 (61.6)	50 (59.5)	40 (47.6)
M	33 (38.4)	34 (40.5)	44 (52.4)

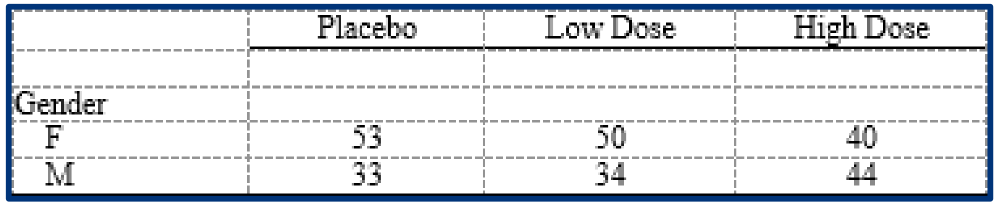
Notes:

- By default, the freq() function displays a data driven summary, returning a row for each unique value of rowvar present in the “df” dataframe.
- The default numerator is the number of distinct subjects (USUBJID) within each rowvar category.
- The default denominator is the number of subjects (USUBJID) with a **non-missing rowvar value**.

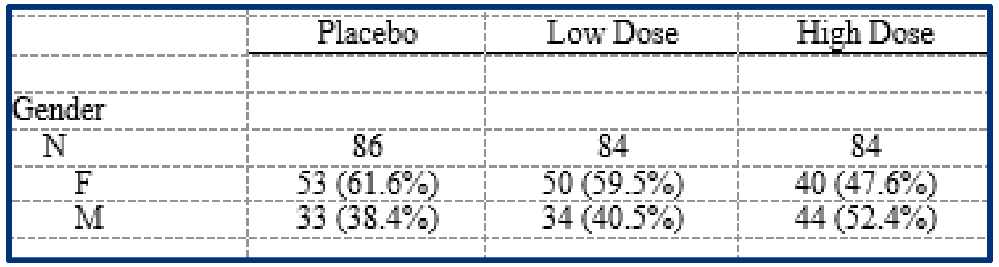
- **Statlist** - specifying the statistics to display:

Available statistics :
- N
- n
- n/N
- n (x.x)
- n (x.x%)
- n/N (x.x)
- n/N (x.x%)

```
#Examples:
ex4a <- freq(df         = adsl,
            colvar      = "colnbr",
            rowvar      =  “SEX” ,
            statlist    =
statlist(c( “n”)),
            row_header = "Gender")

ex4b <- freq(df        = adsl,
            colvar     = "colnbr",
            rowvar     = "SEX",
            statlist   = statlist(c("
N", "n(x.x%)")),
            row_header = "Gender")
```



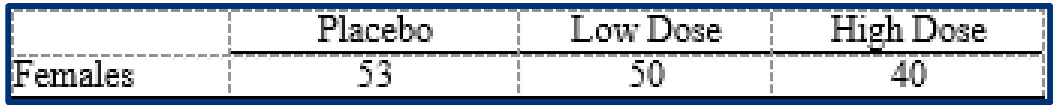
	Placebo	Low Dose	High Dose
Gender			
F	53	50	40
M	33	34	44



	Placebo	Low Dose	High Dose
Gender			
N	86	84	84
F	53 (61.6%)	50 (59.5%)	40 (47.6%)
M	33 (38.4%)	34 (40.5%)	44 (52.4%)

- **Subset** - summarizing a single category:

```
Examples:
ex5 <- freq(df        =
adsl,
            colvar    =
”colnbr”,
            rowvar    =
”SEX”,
            statlist =
statlist(”n”),
            subset    =
SEX == ”F”,
            rowtext   =
”Females “)
```

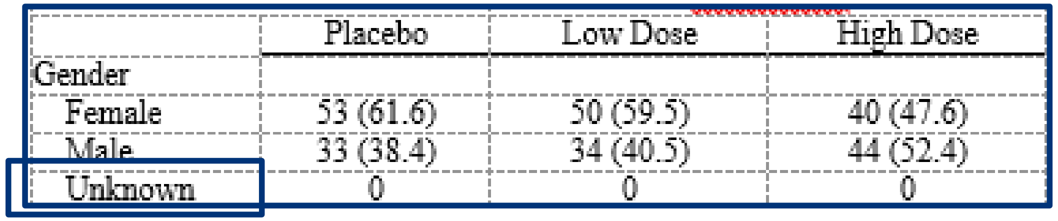


	Placebo	Low Dose	High Dose
Females	53	50	40

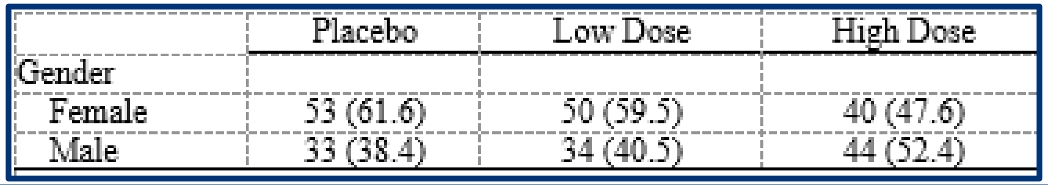
- **Rowvar/Pad** - summarizing a preset list of categories:

Notes: to prespecify a list of categories, “rowvar” must be a factor, with the desired levels set prior to the call to freq().

```
#Examples:
#SEX = factor(SEX,
         levels = c(”F”, ”M”, ”U”),
         labels = c(”Female”, ”Male”,
”Unknown”)))

ex6a <- freq(df          = adsl,
              colvar     = ”colnbr”,
              rowvar     = ”SEX”,
              row_header =
’Gender’ ))

#pad : remove with 0 records
ex6b <- freq(df          = adsl,
              colvar     = ”colnbr”,
              rowvar     = ”SEX”,
              pad        = FALSE,
              row_header =
```



	Placebo	Low Dose	High Dose
Gender			
Female	53 (61.6)	50 (59.5)	40 (47.6)
Male	33 (38.4)	34 (40.5)	44 (52.4)
Unknown	0	0	0



	Placebo	Low Dose	High Dose
Gender			
Female	53 (61.6)	50 (59.5)	40 (47.6)
Male	33 (38.4)	34 (40.5)	44 (52.4)

- **denom_df -** define your denominator using a second dataframe:

Notes: numerator and the denominator are both defined using the df parameter by default.

adae					
USUBJID	colnbr	TRTAN	TRTA	TRTEMFL	AEDECOD
01-701-1015	col1	0	Placebo	Y	APPLICATION SITE ERYTHEMA
01-701-1015	col1	0	Placebo	Y	APPLICATION SITE PRURITUS
01-701-1015	col1	0	Placebo	Y	DIARRHOEA
01-701-1023	col1	0	Placebo	Y	ERYTHEMA
01-701-1023	col1	0	Placebo	Y	ERYTHEMA
01-701-1023	col1	0	Placebo	Y	ERYTHEMA
01-701-1047	col1	0	Placebo	Y	HIATUS HERNIA
01-701-1234	col1	0	Placebo	Y	PYREXIA
…					

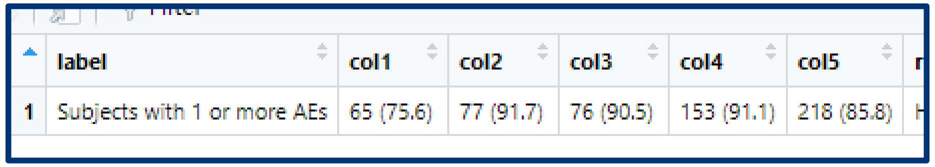
+

adsl				
USUBJID	colnbr	TRT01AN	TRT01A	SAFFL
01-701-1015	col1	0	Placebo	Y
01-701-1023	col1	0	Placebo	Y
01-701-1047	col1	0	Placebo	Y
01-701-1118	col1	0	Placebo	Y
01-701-1130	col1	0	Placebo	Y
01-701-1153	col1	0	Placebo	Y
01-701-1203	col1	0	Placebo	Y
01-701-1234	col1	0	Placebo	Y
. . .				

Numerator: Subjects with 1 or more AEs

Denominator: Treated Subjects - includes subjects with AEs (highlighted) and subjects without AEs)

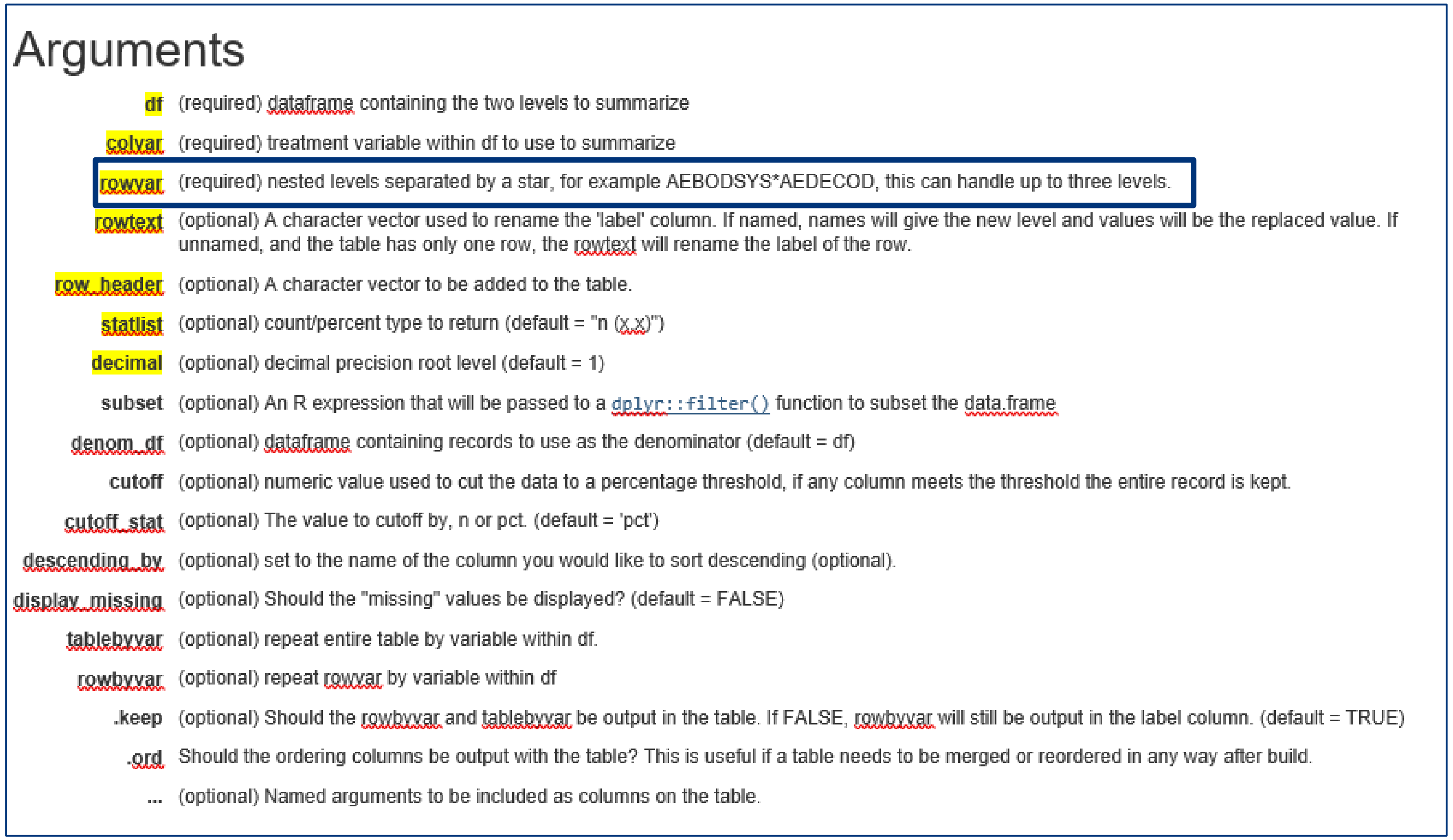
```
#Examples:
ex7  <-  freq(df       = adae,
              colvar   = "colnbr",
              denom_df = adsl,
              rowvar   = "TRTEMFL",
              subset   = TRTEMFL == "Y",
              rowtext  = "Subjects with 1
or more AEs")
```



	label	col1	col2	col3	col4	col5	r
1	Subjects with 1 or more AEs	65 (75.6)	77 (91.7)	76 (90.5)	153 (91.1)	218 (85.8)	H

Deep Dive #3: nested_freq()

Usage of arguments is the essentially the same as freq(), except for rowvar.



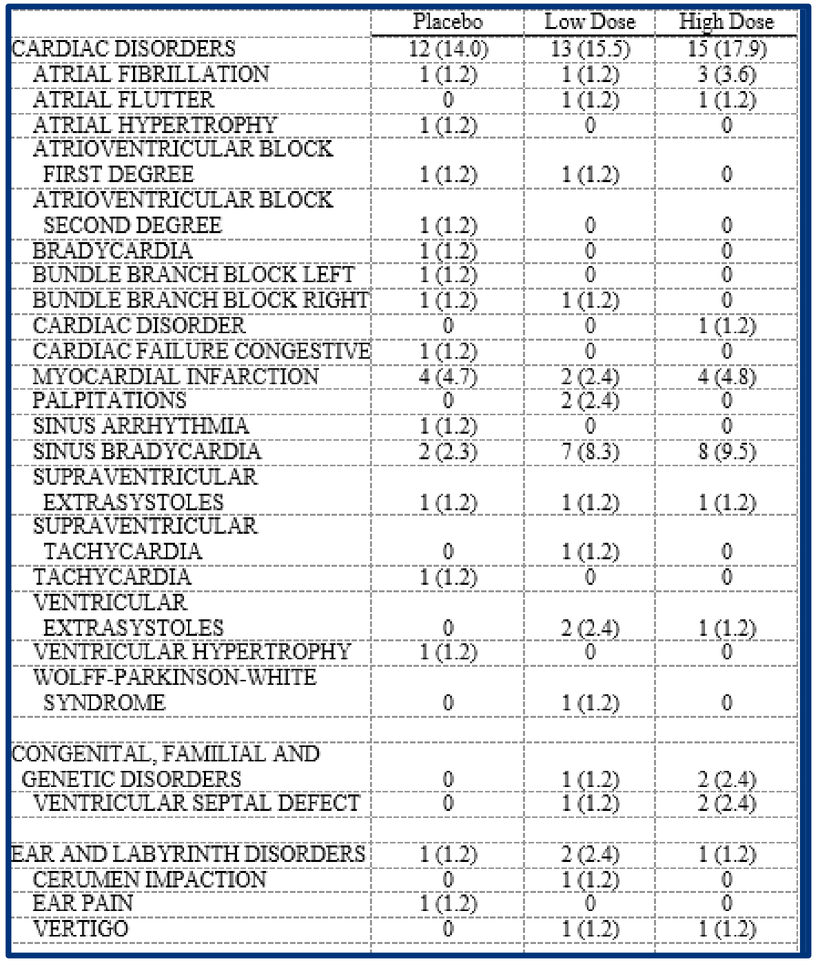
Arguments

df	(required) dataframe containing the two levels to summarize
colvar	(required) treatment variable within df to use to summarize
rowvar	(required) nested levels separated by a star, for example AEBODSYS*AEDECOD, this can handle up to three levels.
rowtext	(optional) A character vector used to rename the 'label' column. If named, names will give the new level and values will be the replaced value. If unnamed, and the table has only one row, the rowtext will rename the label of the row.
row_header	(optional) A character vector to be added to the table.
statlist	(optional) count/percent type to return (default = "n (x.x)")
decimal	(optional) decimal precision root level (default = 1)
subset	(optional) An R expression that will be passed to a `dplyr::filter()` function to subset the data.frame
denom_df	(optional) dataframe containing records to use as the denominator (default = df)
cutoff	(optional) numeric value used to cut the data to a percentage threshold, if any column meets the threshold the entire record is kept.
cutoff_stat	(optional) The value to cutoff by, n or pct. (default = 'pct')
descending_by	(optional) set to the name of the column you would like to sort descending (optional).
display_missing	(optional) Should the "missing" values be displayed? (default = FALSE)
tablebyvar	(optional) repeat entire table by variable within df.
rowbyvar	(optional) repeat rowvar by variable within df
.keep	(optional) Should the rowbyvar and tablebyvar be output in the table. If FALSE, rowbyvar will still be output in the label column. (default = TRUE)
.ord	Should the ordering columns be output with the table? This is useful if a table needs to be merged or reordered in any way after build.
...	(optional) Named arguments to be included as columns on the table.

- **rowvar -** specify list of variables to be nested separated by * :

```
#Examples:
ex_nf1 <- nested_freq(df       = adae,
                      denom_df = adsl,
                      colvar   =
"colnbr"
                      rowvar   =
```

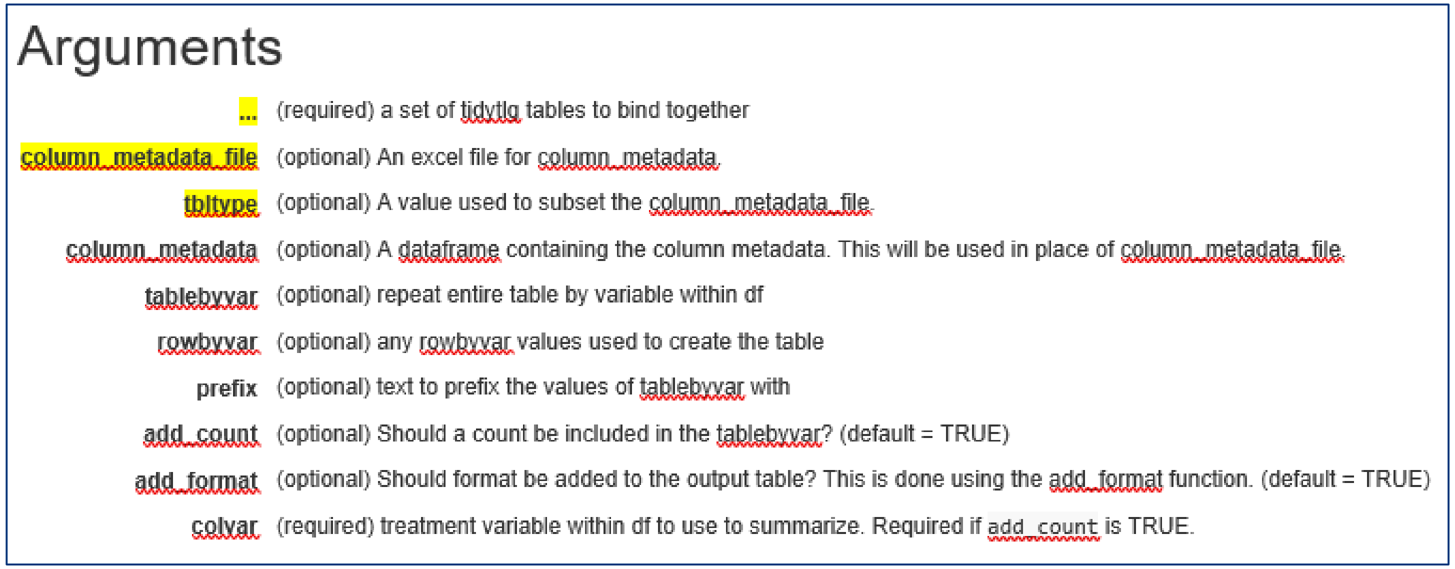
Notes: Passing in AEBODSYS*AEDECOD will summarize AE incidence by body system and preferred terms with preferred terms nested within body system.



	Placebo	Low Dose	High Dose
CARDIAC DISORDERS	12 (14.0)	13 (15.5)	15 (17.9)
ATRIAL FIBRILLATION	1 (1.2)	1 (1.2)	3 (3.6)
ATRIAL FLUTTER	0	1 (1.2)	1 (1.2)
ATRIAL HYPERTROPHY	1 (1.2)	0	0
ATRIOVENTRICULAR BLOCK FIRST DEGREE	1 (1.2)	1 (1.2)	0
ATRIOVENTRICULAR BLOCK SECOND DEGREE	1 (1.2)	0	0
BRADYCARDIA	1 (1.2)	0	0
BUNDLE BRANCH BLOCK LEFT	1 (1.2)	0	0
BUNDLE BRANCH BLOCK RIGHT	1 (1.2)	1 (1.2)	0
CARDIAC DISORDER	0	0	1 (1.2)
CARDIAC FAILURE CONGESTIVE	1 (1.2)	0	0
MYOCARDIAL INFARCTION	4 (4.7)	2 (2.4)	4 (4.8)
PALPITATIONS	0	2 (2.4)	0
SINUS ARRHYTHMIA	1 (1.2)	0	0
SINUS BRADYCARDIA	2 (2.3)	7 (8.3)	8 (9.5)
SUPRAVENTRICULAR EXTRASYSTOLES	1 (1.2)	1 (1.2)	1 (1.2)
SUPRAVENTRICULAR TACHYCARDIA	0	1 (1.2)	0
TACHYCARDIA	1 (1.2)	0	0
VENTRICULAR EXTRASYSTOLES	0	2 (2.4)	1 (1.2)
VENTRICULAR HYPERTROPHY	1 (1.2)	0	0
WOLFF-PARKINSON-WHITE SYNDROME	0	1 (1.2)	0
CONGENITAL, FAMILIAL AND GENETIC DISORDERS	0	1 (1.2)	2 (2.4)
VENTRICULAR SEPTAL DEFECT	0	1 (1.2)	2 (2.4)
EAR AND LABYRINTH DISORDERS	1 (1.2)	2 (2.4)	1 (1.2)
CERUMEN IMPACTION	0	1 (1.2)	0
EAR PAIN	1 (1.2)	0	0
VERTIGO	0	1 (1.2)	1 (1.2)

Deep Dive #4: bind_table()

Preparing your data to be passed into gentlg(), helping you bind your summary dataframe(s) into a single "TBL" file and will add any necessary row formatting metadata



Arguments

...	(required) a set of tidytlg tables to bind together
column_metadata_file	(optional) An excel file for column_metadata.
tbltype	(optional) A value used to subset the column_metadata_file.
column_metadata	(optional) A dataframe containing the column metadata. This will be used in place of column_metadata_file.
tablebyvar	(optional) repeat entire table by variable within df
rowbyvar	(optional) any rowbyvar values used to create the table
prefix	(optional) text to prefix the values of tablebyvar with
add_count	(optional) Should a count be included in the tablebyvar? (default = TRUE)
add_format	(optional) Should format be added to the output table? This is done using the add_format function. (default = TRUE)
colvar	(required) treatment variable within df to use to summarize. Required if add_count is TRUE.

```
#Examples:
tbl <- bind_table(pop, age, sex,
                  column_metadata_file =
column_metadata_file,
                  tbltype              = ”type1”)
```

```
- attr(*, "column_metadata")= tibble [3 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ tbltype: chr [1:3] "type1" "type1" "type1"
  ..$ coldef : chr [1:3] "0" "54" "81"
  ..$ decode : chr [1:3] "Placebo" "Low Dose" "High Dose"
  ..$ span1  : chr [1:3] NA "Xanomeline" "Xanomeline"
  ..$ span2  : logi [1:3] NA NA NA
  ..$ span3  : logi [1:3] NA NA NA
```

Core “TBL” data frame includes the column label, plus all other columns to be displayed (generally, col1, col2, …, coln).

All other columns contain row metadata; providing formatting instructions to gentlg().

Passing in summary dataframes will bind them into a single “tbl” file.

pop

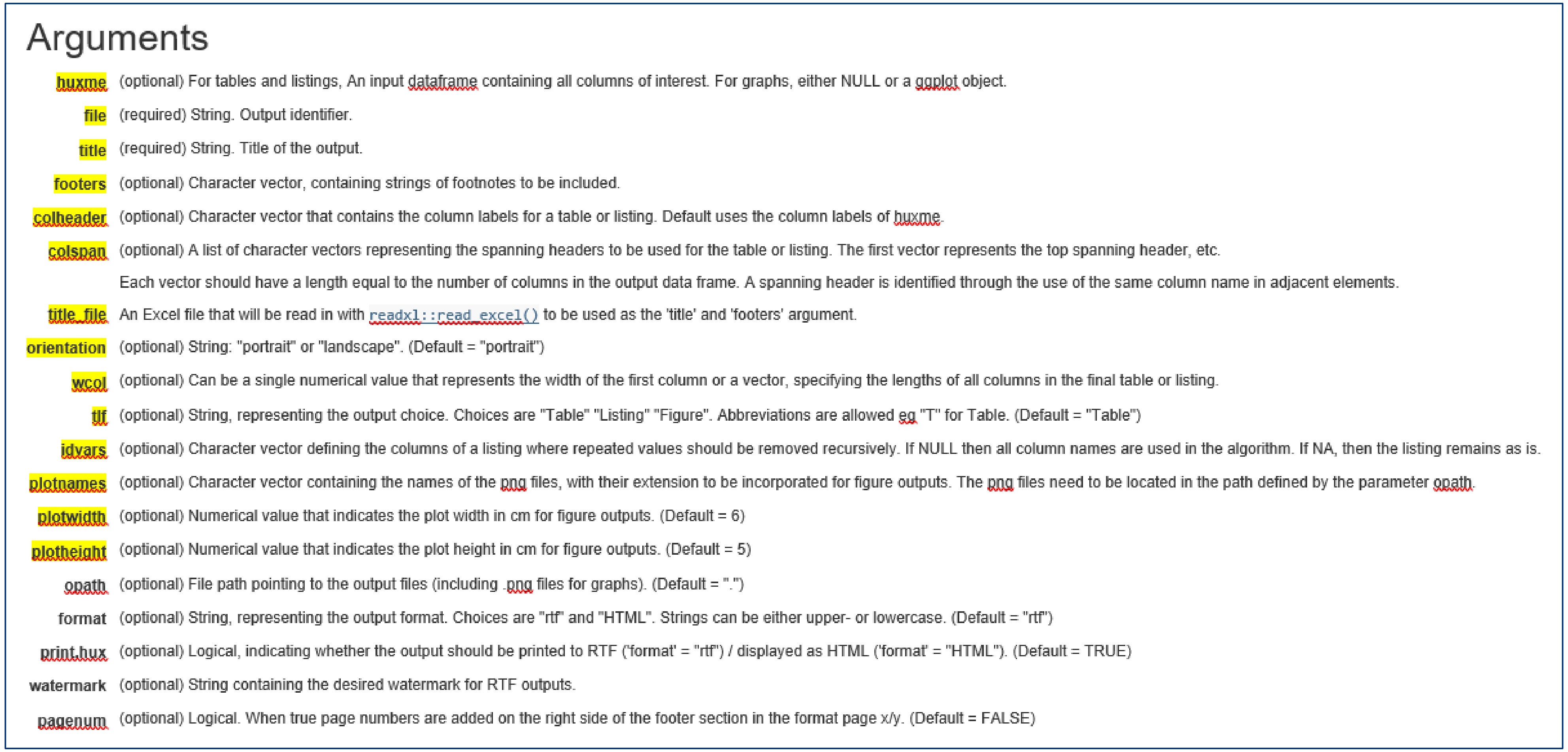
age

sex

label	col1	col2	col3	row_type	anbr	Indentme	roworder	newrows
Analysis set: ITT	86	84	84	HEADER	1	0	1	0
Age, years	*NA*	*NA*	*NA*	HEADER	2	0	1	1
N	86	84	84	N	2	1	2	0
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)	VALUE	2	2	3	0
Median	76	77.5	76	VALUE	2	2	4	0
Range	(52; 89)	(51; 88)	(56; 88)	VALUE	2	2	5	0
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)	VALUE	2	2	6	0
Gender				HEADER	3	0	1	1
N	86	84	84	N	3	1	2	0
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)	VALUE	3	2	3	0
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)	VALUE	3	2	4	0
Undifferentiated	0	0	0	VALUE	3	2	5	0
Unknown	0	0	0	VALUE	3	2	6	0

Deep Dive #5: gentlg()

Outputs a formatted version of your table, based on the formatting metadata supplied.

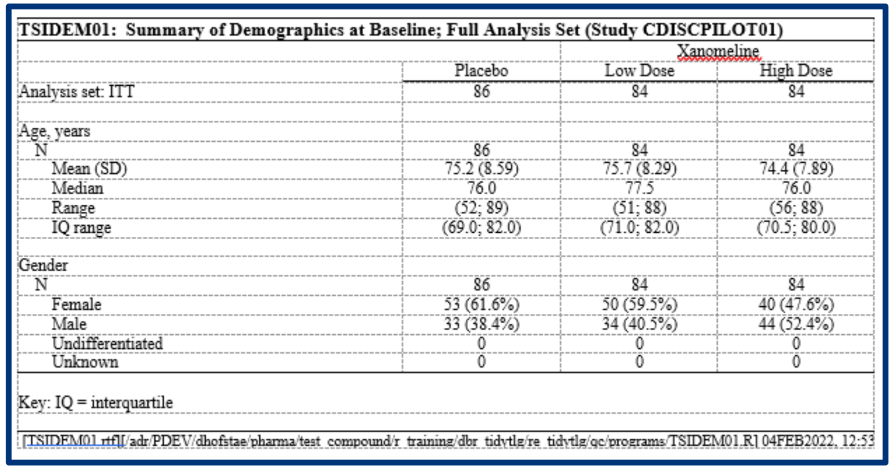


Arguments

huxme	(optional) For tables and listings, An input dataframe containing all columns of interest. For graphs, either NULL or a ggplot object.
file	(required) String. Output identifier.
title	(required) String. Title of the output.
footers	(optional) Character vector, containing strings of footnotes to be included.
colheader	(optional) Character vector that contains the column labels for a table or listing. Default uses the column labels of huxme.
colspan	(optional) A list of character vectors representing the spanning headers to be used for the table or listing. The first vector represents the top spanning header, etc. Each vector should have a length equal to the number of columns in the output data frame. A spanning header is identified through the use of the same column name in adjacent elements.
title_file	An Excel file that will be read in with `readxl::read_excel()` to be used as the 'title' and 'footers' argument.
orientation	(optional) String: "portrait" or "landscape". (Default = "portrait")
wcol	(optional) Can be a single numerical value that represents the width of the first column or a vector, specifying the lengths of all columns in the final table or listing.
tlf	(optional) String, representing the output choice. Choices are "Table" "Listing" "Figure". Abbreviations are allowed eg "T" for Table. (Default = "Table")
idvars	(optional) Character vector defining the columns of a listing where repeated values should be removed recursively. If NULL then all column names are used in the algorithm. If NA, then the listing remains as is.
plotnames	(optional) Character vector containing the names of the png files, with their extension to be incorporated for figure outputs. The png files need to be located in the path defined by the parameter opath.
plotwidth	(optional) Numerical value that indicates the plot width in cm for figure outputs. (Default = 6)
plotheight	(optional) Numerical value that indicates the plot height in cm for figure outputs. (Default = 5)
opath	(optional) File path pointing to the output files (including .png files for graphs). (Default = ".")
format	(optional) String, representing the output format. Choices are "rtf" and "HTML". Strings can be either upper- or lowercase. (Default = "rtf")
print.hux	(optional) Logical, indicating whether the output should be printed to RTF ('format' = "rtf") / displayed as HTML ('format' = "HTML"). (Default = TRUE)
watermark	(optional) String containing the desired watermark for RTF outputs.
pagenum	(optional) Logical. When true page numbers are added on the right side of the footer section in the format page x/y. (Default = FALSE)

- **Colheader/colspan** - if we've attached the column metadata by specifying the tabletype in our bind_table() call, we don't need to manually define our column headers and spanning headers.

```
#Examples:
gentlg(huxme      = tbl,
       file       = "TSIDEM01",
       title      = "Summary of Demographics at Baseline; Full Analysis Set
(Study CDISCPILOT01)",
       footers    = "Key: IQ = Interquartile ")
       ~~colheader = c(" ", "Placebo", "Low Dose",    "High Dose"),~~
~~       colspan     = list(c(" ", " ",         "Xanomeline", "Xanomeline")))~~
```

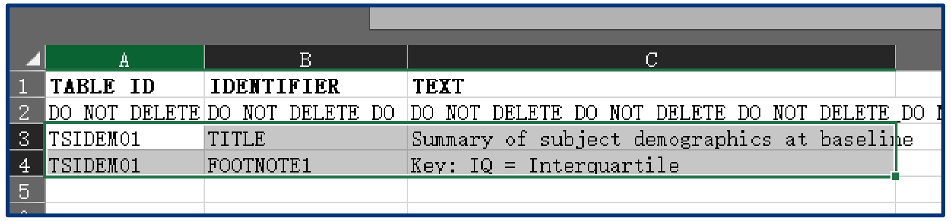


TSIDEM01: Summary of Demographics at Baseline; Full Analysis Set (Study CDISCPILOT01)			
		Xanomeline	
	Placebo	Low Dose	High Dose
Analysis set: ITT	86	84	84
Age, years			
N	86	84	84
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)
Median	76.0	77.5	76.0
Range	(52; 89)	(51; 88)	(56; 88)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)
Gender			
N	86	84	84
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)
Undifferentiated	0	0	0
Unknown	0	0	0

Key: IQ = interquartile

[TSIDEM01.rtf][/adr/PDEV/dhofstae/pharma/test_compound/r_training/dbr_tidytlg/re_tidytlg/qc/programs/TSIDEM01.R] 04FEB2022, 12:53

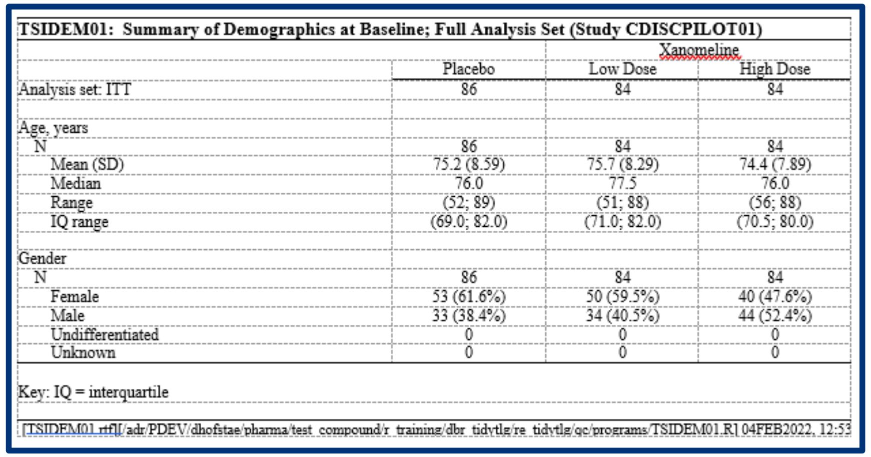
- **title_file** - if we point to a titles file, titles.xlsx, we don't need to manually define our titles and footnotes.



	A	B	C
1	TABLE ID	IDENTIFIER	TEXT
2	DO NOT DELETE	DO NOT DELETE DO	DO NOT DELETE DO NOT DELETE DO NOT DELETE DO N
3	TSIDEM01	TITLE	Summary of subject demographics at baseline
4	TSIDEM01	FOOTNOTE1	Key: IQ = Interquartile
5			

titles.xlsx: the information for titles and footnotes for each TLG can be stored in an excel file.

```
#Examples:
input <- "/cloud/project/input "

gentlg(huxme       = tbl,
       file        = "TSIDEM01",
       title_file =
file.path(input,"titles.xlsx"))
```



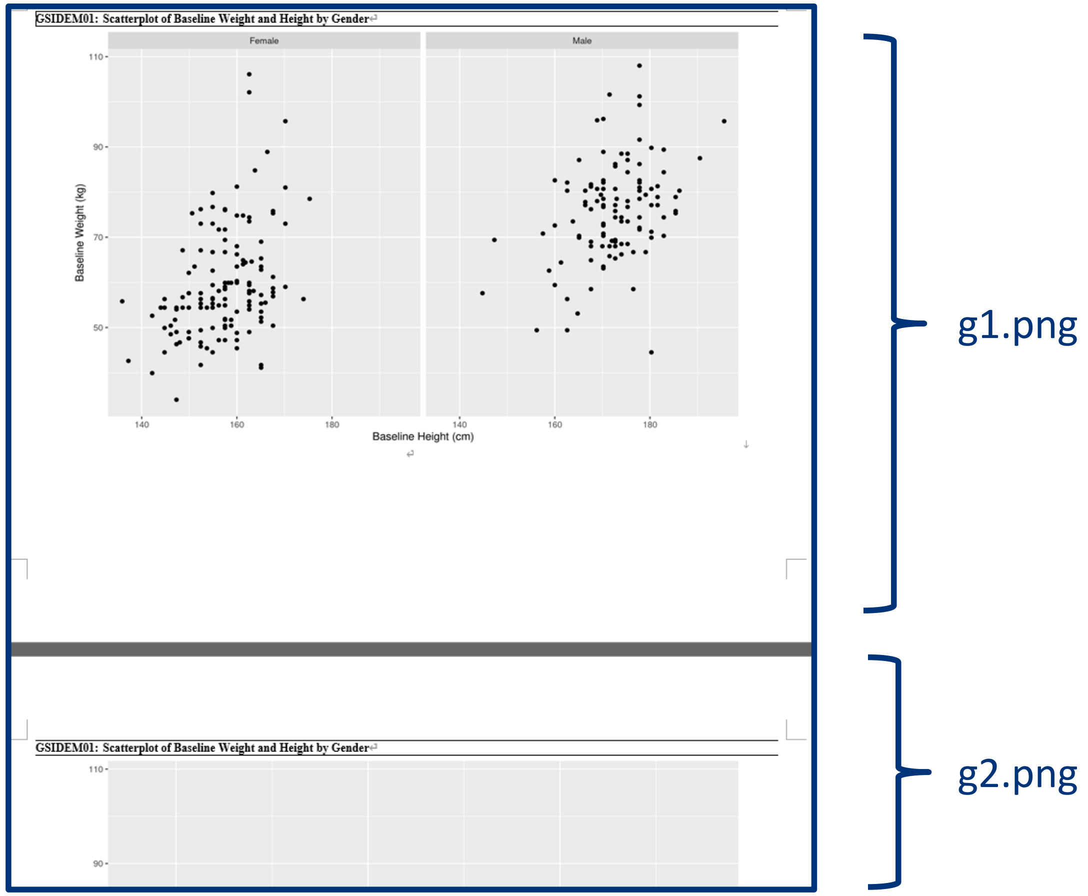
TSIDEM01: Summary of Demographics at Baseline; Full Analysis Set (Study CDISCPILOT01)			
		Xanomeline	
	Placebo	Low Dose	High Dose
Analysis set: ITT	86	84	84
Age, years			
N	86	84	84
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)
Median	76.0	77.5	76.0
Range	(52; 89)	(51; 88)	(56; 88)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)
Gender			
N	86	84	84
Female	53 (61.6%)	50 (59.5%)	40 (47.6%)
Male	33 (38.4%)	34 (40.5%)	44 (52.4%)
Undifferentiated	0	0	0
Unknown	0	0	0

Key: IQ = interquartile

[TSIDEM01.rtf][/adr/PDEV/dhofstae/pharma/test_compound/r_training/dbr_tidytlg/re_tidytlg/qc/programs/TSIDEM01.R] 04FEB2022, 12:53

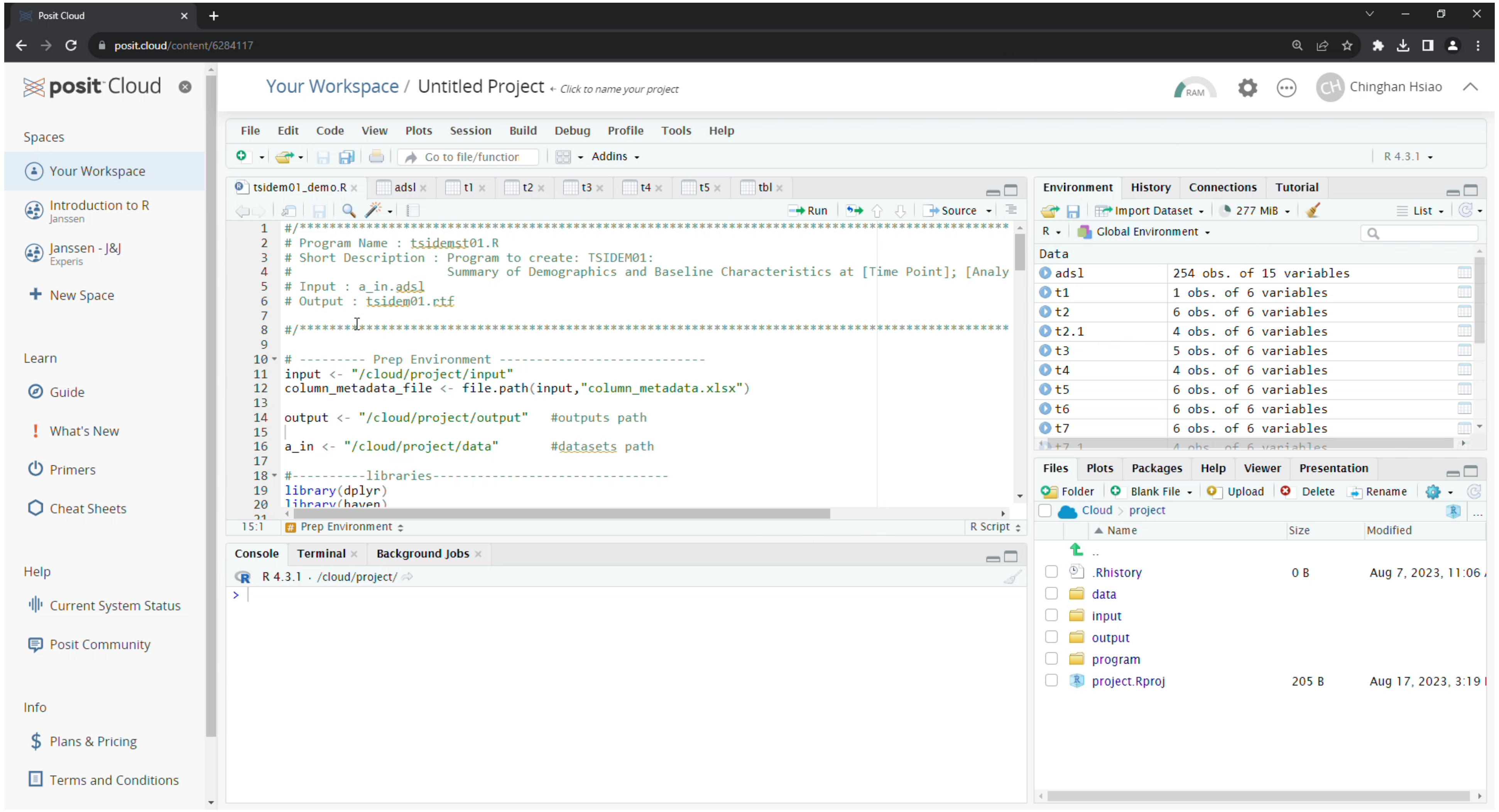
- **plotnames -** using png files to output the results

```
#Examples:
output <- "/cloud/project/output"
plot.name <- c("g1.png","g2.png")

gentlg(plotnames = file.path(output, plot.name),
       tlf       = "Graph",
       plotwidth  = 6,
       plotheight = 4,
       file      = "GSIDEM01",
       title     =  “Scatterplot of Baseline Weight and Height
by Gender” )
```



Demo

Tsidem01- Summary of Subject Demographics (Metadata Method)

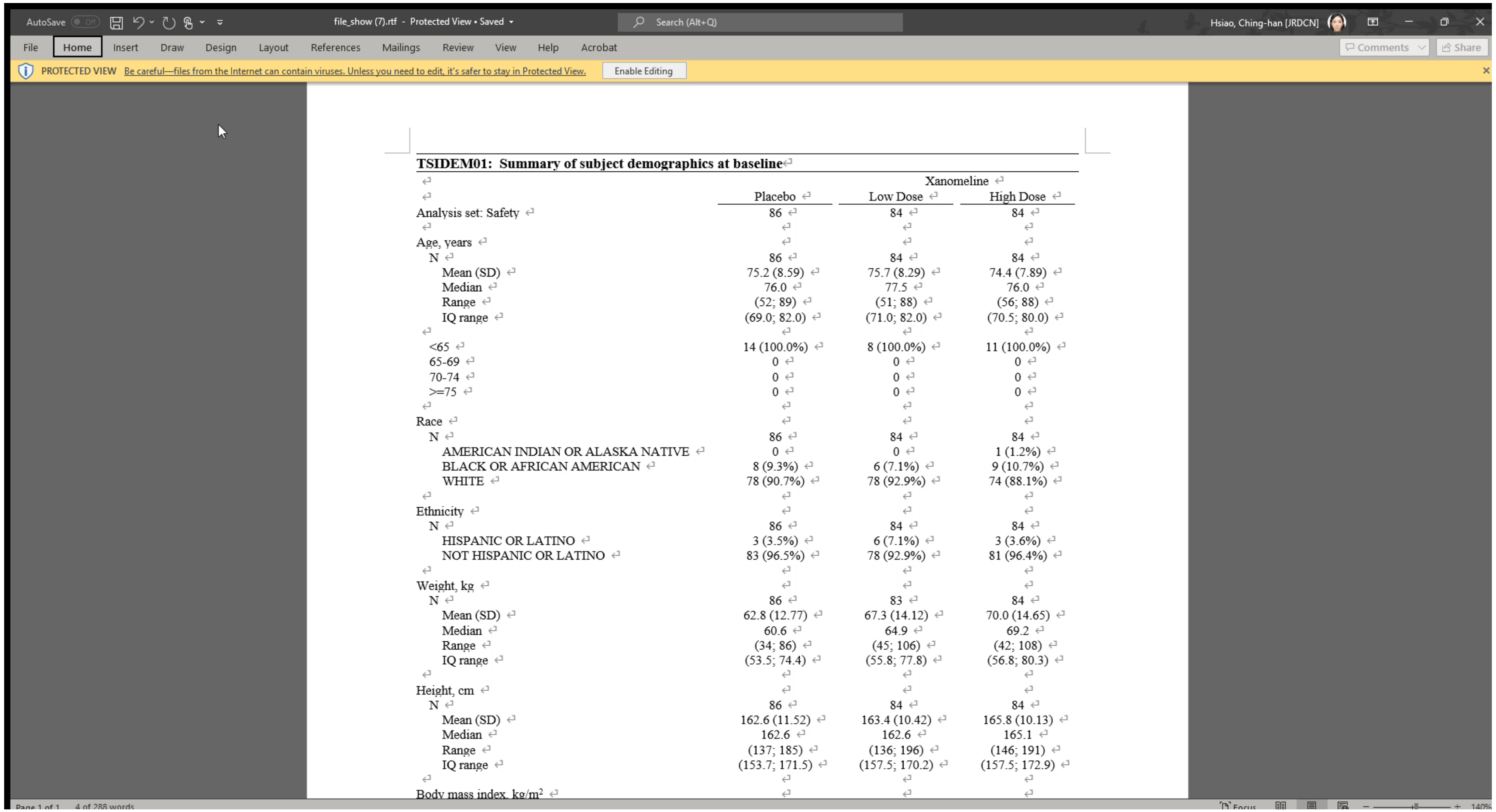


```
#/************************************************************************************
# Program Name : tsidemst01.R
# Short Description : Program to create: TSIDEM01:
#                     Summary of Demographics and Baseline Characteristics at [Time Point]; [Analy
# Input : a_in.adsl
# Output : tsidem01.rtf

#/************************************************************************************

# --------- Prep Environment ---------------------------
input <- "/cloud/project/input"
column_metadata_file <- file.path(input,"column_metadata.xlsx")

output <- "/cloud/project/output"   #outputs path

a_in <- "/cloud/project/data"       #datasets path

#---------libraries--------------------------------
library(dplyr)
library(haven)
```

```
Data
adsl   254 obs. of 15 variables
t1     1 obs. of 6 variables
t2     6 obs. of 6 variables
t2.1   4 obs. of 6 variables
t3     5 obs. of 6 variables
t4     4 obs. of 6 variables
t5     6 obs. of 6 variables
t6     6 obs. of 6 variables
t7     6 obs. of 6 variables
```

Name	Size	Modified
..		
.Rhistory	0 B	Aug 7, 2023, 11:06
data		
input		
output		
program		
project.Rproj	205 B	Aug 17, 2023, 3:19

```
R 4.3.1 · /cloud/project/
>
```

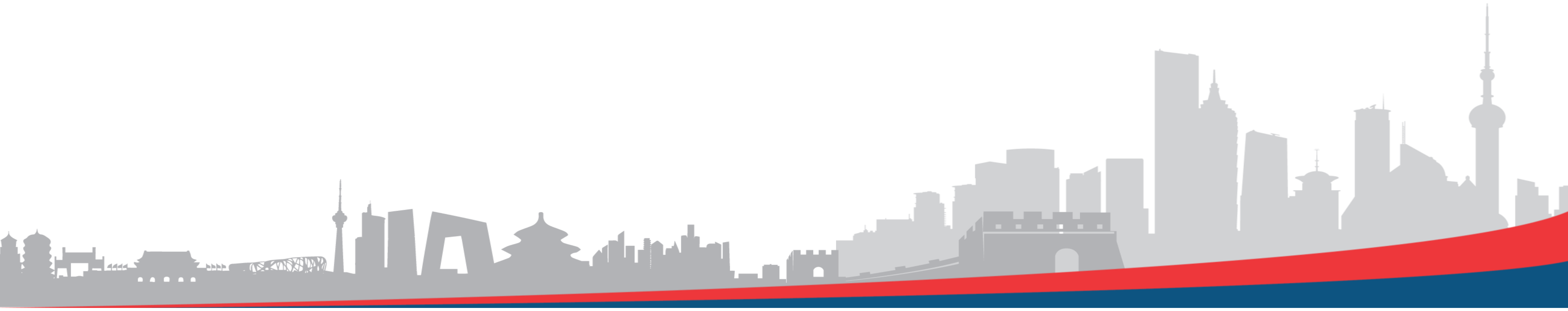


Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

TSIDEM01: Summary of subject demographics at baseline

		Xanomeline	
	Placebo	Low Dose	High Dose
Analysis set: Safety	86	84	84
Age, years			
N	86	84	84
Mean (SD)	75.2 (8.59)	75.7 (8.29)	74.4 (7.89)
Median	76.0	77.5	76.0
Range	(52; 89)	(51; 88)	(56; 88)
IQ range	(69.0; 82.0)	(71.0; 82.0)	(70.5; 80.0)
<65	14 (100.0%)	8 (100.0%)	11 (100.0%)
65-69	0	0	0
70-74	0	0	0
>=75	0	0	0
Race			
N	86	84	84
AMERICAN INDIAN OR ALASKA NATIVE	0	0	1 (1.2%)
BLACK OR AFRICAN AMERICAN	8 (9.3%)	6 (7.1%)	9 (10.7%)
WHITE	78 (90.7%)	78 (92.9%)	74 (88.1%)
Ethnicity			
N	86	84	84
HISPANIC OR LATINO	3 (3.5%)	6 (7.1%)	3 (3.6%)
NOT HISPANIC OR LATINO	83 (96.5%)	78 (92.9%)	81 (96.4%)
Weight, kg			
N	86	83	84
Mean (SD)	62.8 (12.77)	67.3 (14.12)	70.0 (14.65)
Median	60.6	64.9	69.2
Range	(34; 86)	(45; 106)	(42; 108)
IQ range	(53.5; 74.4)	(55.8; 77.8)	(56.8; 80.3)
Height, cm			
N	86	84	84
Mean (SD)	162.6 (11.52)	163.4 (10.42)	165.8 (10.13)
Median	162.6	162.6	165.1
Range	(137; 185)	(136; 196)	(146; 191)
IQ range	(153.7; 171.5)	(157.5; 170.2)	(157.5; 172.9)
Body mass index, kg/m^2			

Summary

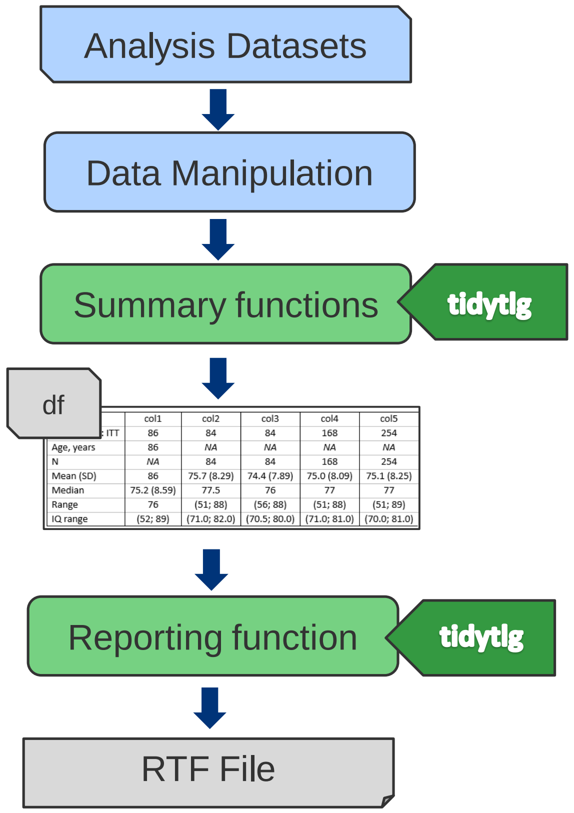


Empowering Innovation . Driving Growth

janssen | PHARMACEUTICAL COMPANIES OF Johnson & Johnson

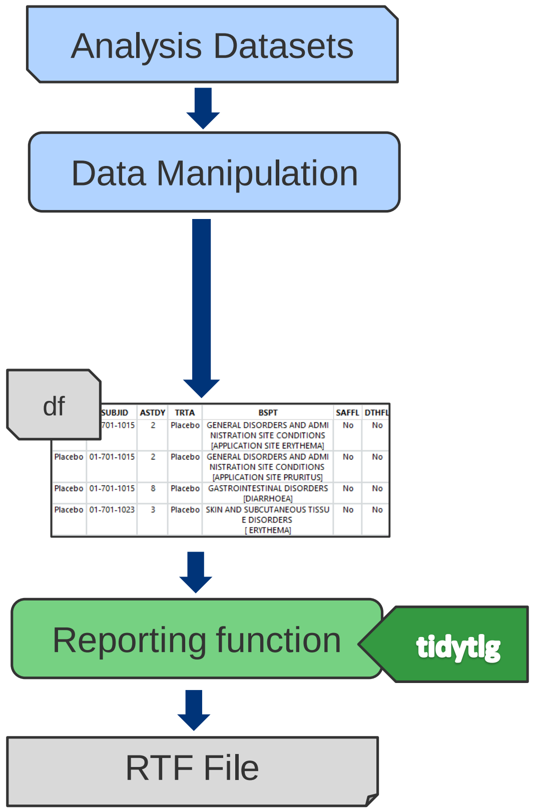
Summary

Tables:
Summarize the data and store in a dataframe; pass dataframe through the reporting function along with formatting metadata.



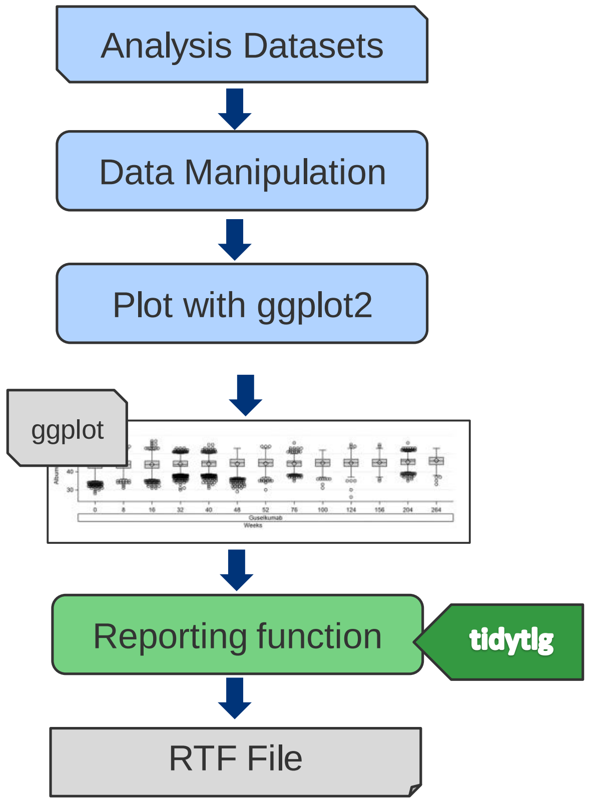
	col1	col2	col3	col4	col5
: ITT	86	84	84	168	254
Age, years	86	*NA*	*NA*	*NA*	*NA*
N	*NA*	84	84	168	254
Mean (SD)	86	75.7 (8.29)	74.4 (7.89)	75.0 (8.09)	75.1 (8.25)
Median	75.2 (8.59)	77.5	76	77	77
Range	76	(51; 88)	(56; 88)	(51; 88)	(51; 89)
IQ range	(52; 89)	(71.0; 82.0)	(70.5; 80.0)	(71.0; 81.0)	(70.0; 81.0)

Listings:
Gather the data and store in a dataframe; pass dataframe through the reporting function along with formatting metadata.



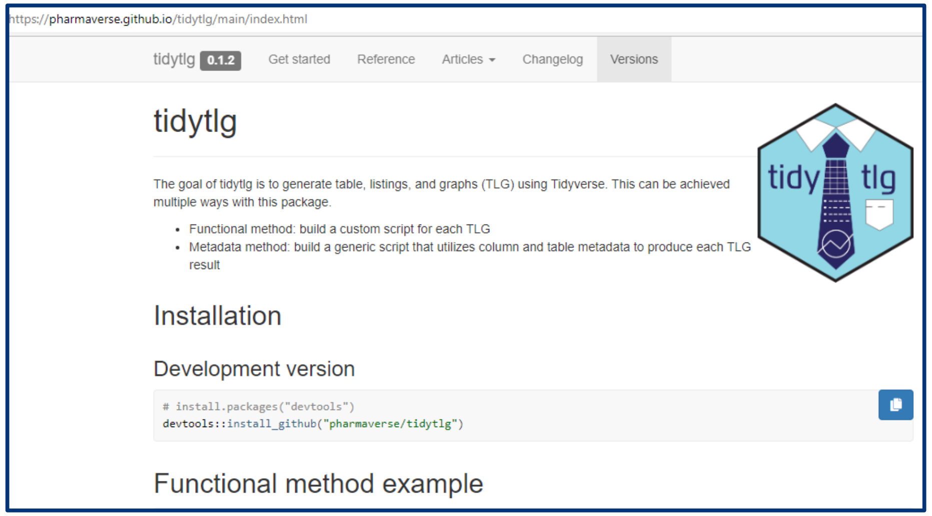
	SUBJID	ASTDY	TRTA	BSPT	SAFFL	DTHFL
	701-1015	2	Placebo	GENERAL DISORDERS AND ADMI NISTRATION SITE CONDITIONS [APPLICATION SITE ERYTHEMA]	No	No
Placebo	01-701-1015	2	Placebo	GENERAL DISORDERS AND ADMI NISTRATION SITE CONDITIONS [APPLICATION SITE PRURITUS]	No	No
Placebo	01-701-1015	8	Placebo	GASTROINTESTINAL DISORDERS [DIARRHOEA]	No	No
Placebo	01-701-1023	3	Placebo	SKIN AND SUBCUTANEOUS TISSU E DISORDERS [ERYTHEMA]	No	No

Graphs:
Summarize the data and plot with ggplot2. Pass the plot through the reporting function along with formatting metadata.



Helper Panels

- [tidytlg on Github]


https://pharmaverse.github.io/tidytlg/main/index.html

tidytlg

The goal of tidytlg is to generate table, listings, and graphs (TLG) using Tidyverse. This can be achieved multiple ways with this package.

- Functional method: build a custom script for each TLG
- Metadata method: build a generic script that utilizes column and table metadata to produce each TLG result

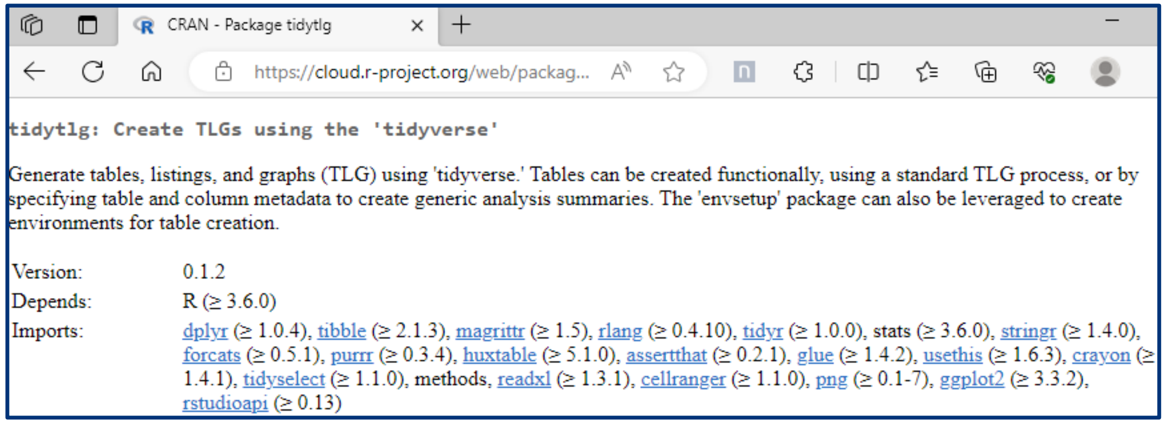
Installation

Development version

```
# install.packages("devtools")
devtools::install_github("pharmaverse/tidytlg")
```

Functional method example

- [tidytlg on CRAN]



https://cloud.r-project.org/web/packag...

tidytlg: Create TLGs using the 'tidyverse'

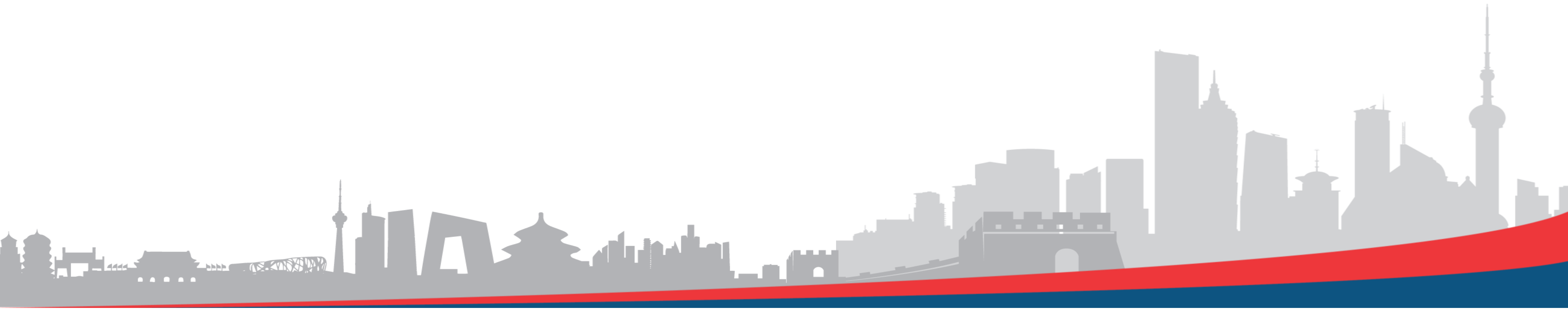
Generate tables, listings, and graphs (TLG) using 'tidyverse.' Tables can be created functionally, using a standard TLG process, or by specifying table and column metadata to create generic analysis summaries. The 'envsetup' package can also be leveraged to create environments for table creation.

Version:	0.1.2
Depends:	R (≥ 3.6.0)
Imports:	dplyr (≥ 1.0.4), tibble (≥ 2.1.3), magrittr (≥ 1.5), rlang (≥ 0.4.10), tidyr (≥ 1.0.0), stats (≥ 3.6.0), stringr (≥ 1.4.0), forcats (≥ 0.5.1), purrr (≥ 0.3.4), huxtable (≥ 5.1.0), assertthat (≥ 0.2.1), glue (≥ 1.4.2), usethis (≥ 1.6.3), crayon (≥ 1.4.1), tidyselect (≥ 1.1.0), methods, readxl (≥ 1.3.1), cellranger (≥ 1.1.0), png (≥ 0.1-7), ggplot2 (≥ 3.3.2), rstudioapi (≥ 0.13)

Reference

- All datasets are from Open CDISC dummy data (CDISCPILOT01)

Q&A



Thank you!