

# Package: goshawk (via r-universe)

September 4, 2024

**Type** Package

**Title** Longitudinal Visualization Functions

**Version** 0.1.18.9003

**Date** 2024-09-04

**Description** Functions that plot and summarize biomarkers/labs of interest. Visualizations include: box plot, correlation plot, density distribution, line plot and spaghetti plot. Data are expected in ADaM structure. Requires analysis subject level (ADSL) and analysis laboratory (ADLB) data sets. Beyond core variables, Limit of Quantification flag variable (LOQFL) is expected with levels 'Y', 'N' or NA.

**License** Apache License 2.0 | file LICENSE

**URL** <https://insightsengineering.github.io/goshawk/>,  
<https://github.com/insightsengineering/goshawk/>

**BugReports** <https://github.com/insightsengineering/goshawk/issues>

**Depends** dplyr (>= 1.1.0), R (>= 3.6)

**Imports** checkmate (>= 2.1.0), cowplot, ggnewscale (>= 0.4.1), ggplot2 (>= 3.4.4), grDevices, grid, lifecycle (>= 0.2.0), magrittr, mcr, rlang (>= 1.1.0), stringr (>= 1.4.1)

**Suggests** knitr (>= 1.42), nestcolor (>= 0.1.0), rmarkdown (>= 2.23), testthat (>= 3.0.4), tidyr (>= 1.3.1)

**VignetteBuilder** knitr

**Config/Needs/verdepcheck** tidyverse/dplyr, mllg/checkmate, wilkelab/cowplot, eliocamp/ggnewscale, tidyverse/ggplot2, r-lib/lifecycle, tidyverse/magrittr, r-lib/rlang, tidyverse/stringr, yihui/knitr, insightsengineering/nestcolor, rstudio/rmarkdown, r-lib/testthat, tidyverse/tidyr

**Config/Needs/website** insightsengineering/nesttemplate

**Encoding** UTF-8

**Language** en-US

**LazyData** true  
**Roxygen** list(markdown = TRUE)  
**RoxygenNote** 7.3.2  
**Repository** https://pharmaverse.r-universe.dev  
**RemoteUrl** https://github.com/insightsengineering/goshawk  
**RemoteRef** HEAD  
**RemoteSha** 433cafb87803708fc5163931db64914a8c0862a2

## Contents

g_boxplot . . . . .	2
g_correlationplot . . . . .	5
g_density_distribution_plot . . . . .	11
g_lineplot . . . . .	14
g_scatterplot . . . . .	20
g_spaghettoplot . . . . .	23
t_summarytable . . . . .	28

<b>Index</b>	<b>31</b>
--------------	-----------

---

g_boxplot	<i>Function to create a boxplot.</i>
-----------	--------------------------------------

---

## Description

A box plot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the term box-and-whisker. Outliers may be plotted as individual points. Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. The spacings between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. In addition to the points themselves, they allow one to visually estimate various L-estimators, notably the interquartile range, midhinge, range, mid-range, and trimean.

## Usage

```
g_boxplot(
  data,
  biomarker,
  param_var = "PARAMCD",
  yaxis_var,
  trt_group,
  xaxis_var = NULL,
  loq_flag_var = "LOQFL",
  loq_legend = TRUE,
```

```

unit = NULL,
color_manual = NULL,
shape_manual = NULL,
box = TRUE,
ylim = c(NA, NA),
dot_size = 2,
alpha = 1,
facet_ncol = NULL,
rotate_xlab = FALSE,
font_size = NULL,
facet_var = NULL,
hline_arb = numeric(0),
hline_arb_color = "red",
hline_arb_label = "Horizontal line",
hline_vars = character(0),
hline_vars_colors = "green",
hline_vars_labels = hline_vars
)

```

### Arguments

data	ADaM structured analysis laboratory data frame e.g. ADLB.
biomarker	biomarker to visualize e.g. IGG.
param_var	name of variable containing biomarker codes e.g. PARAMCD.
yaxis_var	name of variable containing biomarker results displayed on Y-axis e.g. AVAL.
trt_group	name of variable representing treatment trt_group e.g. ARM.
xaxis_var	variable used to group the data on the x-axis.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL.
loq_legend	logical whether to include LoQ legend.
unit	biomarker unit label e.g. (U/L)
color_manual	vector of color for trt_group
shape_manual	vector of shapes (used with loq_flag_var)
box	add boxes to the plot (boolean)
ylim	(‘numeric vector’) optional, a vector of length 2 to specify the minimum and maximum of the y-axis if the default limits are not suitable.
dot_size	plot dot size.
alpha	dot transparency (0 = transparent, 1 = opaque)
facet_ncol	number of facets per row. NULL = Use the default for ggplot2: : facet_wrap
rotate_xlab	45 degree rotation of x-axis label values.
font_size	point size of text to use. NULL is use default size
facet_var	variable to facet the plot by, or "None" if no faceting required.
hline_arb	(‘numeric vector’) value identifying intercept for arbitrary horizontal lines.

**hline\_arb\_color** ('character vector') optional, color for the arbitrary horizontal lines.  
**hline\_arb\_label** ('character vector') optional, label for the legend to the arbitrary horizontal lines.  
**hline\_vars** ('character vector'), names of variables (ANR\*) or values (\*LOQ) identifying intercept values. The data inside of the ggplot2 object must also contain the columns with these variable names  
**hline\_vars\_colors** ('character vector') colors for the horizontal lines defined by variables.  
**hline\_vars\_labels** ('character vector') labels for the legend to the horizontal lines defined by variables.

### Value

ggplot object

### Author(s)

Balazs Toth

Jeff Tomlinson (tomlinsonj) jeffrey.tomlinson@roche.com

### Examples

```

# Example using ADaM structure analysis dataset.

library(nestcolor)

ADLB <- rADLB
var_labels <- lapply(ADLB, function(x) attributes(x)$label)
ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
    AVISIT == "SCREENING" ~ "SCR",
    AVISIT == "BASELINE" ~ "BL",
    grepl("WEEK", AVISIT) ~
      paste(
        "W",
        trimws(
          substr(
            AVISIT,
            start = 6,
            stop = stringr::str_locate(AVISIT, "DAY") - 1
          )
        ),
        TRUE ~ NA_character_
      )) %>%
  mutate(AVISITCDN = case_when(
    AVISITCD == "SCR" ~ -2,
    AVISITCD == "BL" ~ 0,
    grepl("W", AVISITCD) ~ as.numeric(gsub("\\D+", "", AVISITCD)),
  
```

```

    TRUE ~ NA_real_
  )) %>%
  mutate(ANRLO = .5, ANRHI = 1) %>%
  rowwise() %>%
  group_by(PARAMCD) %>%
  mutate(LBSTRESC = ifelse(
    USUBJID %in% sample(USUBJID, 1, replace = TRUE),
    paste("<", round(runif(1, min = .5, max = 1))), LBSTRESC
  )) %>%
  mutate(LBSTRESC = ifelse(
    USUBJID %in% sample(USUBJID, 1, replace = TRUE),
    paste(">", round(runif(1, min = 1, max = 1.5))), LBSTRESC
  )) %>%
  ungroup()
attr(ADLB[["ARM"]], "label") <- var_labels[["ARM"]]
attr(ADLB[["ANRLO"]], "label") <- "Analysis Normal Range Lower Limit"
attr(ADLB[["ANRHI"]], "label") <- "Analysis Normal Range Upper Limit"

# add LLOQ and ULOQ variables
ADLB_LOQS <- goshawk::h_identify_loq_values(ADLB, "LOQFL")
ADLB <- left_join(ADLB, ADLB_LOQS, by = "PARAM")

g_boxplot(ADLB,
  biomarker = "CRP",
  param_var = "PARAMCD",
  yaxis_var = "AVAL",
  trt_group = "ARM",
  loq_flag_var = "LOQFL",
  loq_legend = FALSE,
  unit = "AVALU",
  shape_manual = c("N" = 1, "Y" = 2, "NA" = NULL),
  facet_var = "AVISIT",
  xaxis_var = "STUDYID",
  alpha = 0.5,
  rotate_xlab = TRUE,
  hline_arb = c(.9, 1.2),
  hline_arb_color = "blue",
  hline_arb_label = "Hori_line_label",
  hline_vars = c("ANRHI", "ANRLO", "ULOQN", "LLOQN"),
  hline_vars_colors = c("pink", "brown", "purple", "gray"),
  hline_vars_labels = c("A", "B", "C", "D")
)

```

---

`g_correlationplot`      *Function to create a correlation plot.*

---

### Description

Default plot displays correlation faceted by visit with color attributed treatment arms and symbol attributed LOQ values.

**Usage**

```
g_correlationplot(  
  label = "Correlation Plot",  
  data,  
  param_var = "PARAMCD",  
  xaxis_param = "CRP",  
  xaxis_var = "BASE",  
  xvar,  
  yaxis_param = "IGG",  
  yaxis_var = "AVAL",  
  yvar,  
  trt_group = "ARM",  
  visit = "AVISITCD",  
  loq_flag_var = "LOQFL_COMB",  
  visit_facet = TRUE,  
  loq_legend = TRUE,  
  unit = "AVALU",  
  xlim = c(NA, NA),  
  ylim = c(NA, NA),  
  title_text = title_text,  
  xaxis_lab = xaxis_lab,  
  yaxis_lab = yaxis_lab,  
  color_manual = NULL,  
  shape_manual = NULL,  
  facet_ncol = 2,  
  facet = FALSE,  
  facet_var = "ARM",  
  reg_line = FALSE,  
  hline_arb = numeric(0),  
  hline_arb_color = "red",  
  hline_arb_label = "Horizontal line",  
  hline_vars = character(0),  
  hline_vars_colors = "green",  
  hline_vars_labels = hline_vars,  
  vline_arb = numeric(0),  
  vline_arb_color = "red",  
  vline_arb_label = "Vertical line",  
  vline_vars = character(0),  
  vline_vars_colors = "green",  
  vline_vars_labels = vline_vars,  
  rotate_xlab = FALSE,  
  font_size = 12,  
  dot_size = 2,  
  reg_text_size = 3  
)
```

**Arguments**

label	text string to used to identify plot.
data	ADaM structured analysis laboratory data frame e.g. ADLB.
param_var	name of variable containing biomarker codes e.g. PARAMCD.
xaxis_param	x-axis biomarker to visualize e.g. IGG.
xaxis_var	name of variable containing biomarker results displayed on X-axis e.g. BASE.
xvar	x-axis analysis variable from transposed data set.
yaxis_param	y-axis biomarker to visualize e.g. IGG.
yaxis_var	name of variable containing biomarker results displayed on Y-axis.g. AVAL.
yvar	y-axis analysis variable from transposed data set.
trt_group	name of variable representing treatment group e.g. ARM.
visit	name of variable containing nominal visits e.g. AVISITCD.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL_COMB.
visit_facet	visit facet toggle.
loq_legend	logical whether to include LoQ legend.
unit	name of variable containing biomarker unit e.g. AVALU.
xlim	('numeric vector') optional, a vector of length 2 to specify the minimum and maximum of the x-axis if the default limits are not suitable.
ylim	('numeric vector') optional, a vector of length 2 to specify the minimum and maximum of the y-axis if the default limits are not suitable.
title_text	plot title.
xaxis_lab	x-axis label.
yaxis_lab	y-axis label.
color_manual	vector of colors applied to treatment values.
shape_manual	vector of symbols applied to LOQ values. (used with loq_flag_var).
facet_ncol	number of facets per row.
facet	set layout to use treatment faceting.
facet_var	variable to use for treatment faceting.
reg_line	include regression line and annotations for slope and coefficient. Use with facet = TRUE.
hline_arb	('numeric vector') value identifying intercept for arbitrary horizontal lines.
hline_arb_color	('character vector') optional, color for the arbitrary horizontal lines.
hline_arb_label	('character vector') optional, label for the legend to the arbitrary horizontal lines.
hline_vars	('character vector'), names of variables (ANR*) or values (*LOQ) identifying intercept values. The data inside of the ggplot2 object must also contain the columns with these variable names

**hline\_vars\_colors** ('character vector') colors for the horizontal lines defined by variables.  
**hline\_vars\_labels** ('character vector') labels for the legend to the horizontal lines defined by variables.  
**vline\_arb** ('numeric vector') value identifying intercept for arbitrary vertical lines.  
**vline\_arb\_color** ('character vector') optional, color for the arbitrary vertical lines.  
**vline\_arb\_label** ('character vector') optional, label for the legend to the arbitrary vertical lines.  
**vline\_vars** ('character vector'), names of variables (ANR\*) or values (\*LOQ) identifying intercept values. The data inside of the ggplot2 object must also contain the columns with these variable names  
**vline\_vars\_colors** ('character vector') colors for the vertical lines defined by variables.  
**vline\_vars\_labels** ('character vector') labels for the legend to the vertical lines defined by variables.  
**rotate\_xlab** 45 degree rotation of x-axis label values.  
**font\_size** font size control for title, x-axis label, y-axis label and legend.  
**dot\_size** plot dot size.  
**reg\_text\_size** font size control for regression line annotations.

### Details

Regression uses deming model.

### Author(s)

Nick Paszty (npaszty) paszty.nicholas@gene.com

Balazs Toth (tothb2) toth.balazs@gene.com

### Examples

```

# Example using ADaM structure analysis dataset.

library(stringr)
library(tidyr)

# original ARM value = dose value
arm_mapping <- list(
  "A: Drug X" = "150mg QD",
  "B: Placebo" = "Placebo",
  "C: Combination" = "Combination"
)
color_manual <- c("150mg QD" = "#000000", "Placebo" = "#3498DB", "Combination" = "#E74C3C")
# assign LOQ flag symbols: circles for "N" and triangles for "Y", squares for "NA"
shape_manual <- c("N" = 1, "Y" = 2, "NA" = 0)

```



```

ADLB <- rADLB
var_labels <- lapply(ADLB, function(x) attributes(x)$label)
ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
    AVISIT == "SCREENING" ~ "SCR",
    AVISIT == "BASELINE" ~ "BL",
    grepl("WEEK", AVISIT) ~
      paste(
        "W",
        trimws(
          substr(
            AVISIT,
            start = 6,
            stop = str_locate(AVISIT, "DAY") - 1
          )
        )
      ),
    TRUE ~ NA_character_
  )) %>%
  mutate(AVISITCDN = case_when(
    AVISITCD == "SCR" ~ -2,
    AVISITCD == "BL" ~ 0,
    grepl("W", AVISITCD) ~ as.numeric(gsub("\\d+", "", AVISITCD)),
    TRUE ~ NA_real_
  )) %>%
  # use ARMCD values to order treatment in visualization legend
  mutate(TRTORD = ifelse(grepl("C", ARMCD), 1,
    ifelse(grepl("B", ARMCD), 2,
      ifelse(grepl("A", ARMCD), 3, NA)
    )
  )) %>%
  mutate(ARM = as.character(arm_mapping[match(ARM, names(arm_mapping))])) %>%
  mutate(ARM = factor(ARM) %>%
    reorder(TRTORD)) %>%
  mutate(
    ANRHI = case_when(
      PARAMCD == "ALT" ~ 60,
      PARAMCD == "CRP" ~ 70,
      PARAMCD == "IGA" ~ 80,
      TRUE ~ NA_real_
    ),
    ANRLO = case_when(
      PARAMCD == "ALT" ~ 20,
      PARAMCD == "CRP" ~ 30,
      PARAMCD == "IGA" ~ 40,
      TRUE ~ NA_real_
    )
  ) %>%
  rowwise() %>%
  group_by(PARAMCD) %>%
  mutate(LBSTRESC = ifelse(
    USUBJID %in% sample(USUBJID, 1, replace = TRUE),
    paste("<", round(runif(1, min = 25, max = 30))), LBSTRESC
  )

```

```

)) %>%
mutate(LBSTRESC = ifelse(
  USUBJID %in% sample(USUBJID, 1, replace = TRUE),
  paste(">", round(runif(1, min = 70, max = 75))), LBSTRESC
)) %>%
ungroup()
attr(ADLB[["ARM"]], "label") <- var_labels[["ARM"]]
attr(ADLB[["ANRHI"]], "label") <- "Analysis Normal Range Upper Limit"
attr(ADLB[["ANRLO"]], "label") <- "Analysis Normal Range Lower Limit"

# add LLOQ and ULOQ variables
ADLB_LOQS <- goshawk::h_identify_loq_values(ADLB, flag_var = "LOQFL")
ADLB <- left_join(ADLB, ADLB_LOQS, by = "PARAM")

# given the 2 param and 2 analysis vars we need to transform the data
plot_data_t1 <- ADLB %>%
gather(
  ANLVARS, ANLVALS, PARAM, LBSTRESC, BASE2, BASE, AVAL, BASE, LOQFL,
  ANRHI, ANRLO, ULOQN, LLOQN
) %>%
mutate(ANL.PARAM = ifelse(ANLVARS %in% c("PARAM", "LBSTRESC", "LOQFL"),
  paste0(ANLVARS, "_", PARAMCD),
  paste0(ANLVARS, ".", PARAMCD)
)) %>%
select(USUBJID, ARM, ARMCD, AVISITN, AVISITCD, ANL.PARAM, ANLVALS) %>%
spread(ANL.PARAM, ANLVALS)

# the transformed analysis value variables are character and need to be converted to numeric for
# ggplot
# remove records where either of the analysis variables are NA since they will not appear on the
# plot and will ensure that LOQFL = NA level is removed
plot_data_t2 <- plot_data_t1 %>%
  filter(!is.na(BASE.CRP) & !is.na(AVAL.ALT)) %>%
  mutate_at(vars(contains(".")), as.numeric) %>%
  mutate(
    LOQFL_COMB = ifelse(LOQFL_CRP == "Y" | LOQFL_ALT == "Y", "Y", "N")
  )

g_correlationplot(
  label = "Correlation Plot",
  data = plot_data_t2,
  param_var = "PARAMCD",
  xaxis_param = c("CRP"),
  xaxis_var = "AVAL",
  xvar = "AVAL.CRP",
  yaxis_param = c("ALT"),
  yaxis_var = "BASE",
  yvar = "BASE.ALT",
  trt_group = "ARM",
  visit = "AVISITCD",
  visit_facet = TRUE,
  loq_legend = TRUE,
  unit = "AVALU",

```

```

title_text = "Correlation of ALT to CRP",
xaxis_lab = "CRP",
yaxis_lab = "ALT",
color_manual = color_manual,
shape_manual = shape_manual,
facet_ncol = 4,
facet = FALSE,
facet_var = "ARM",
reg_line = FALSE,
hline_arb = c(15, 25),
hline_arb_color = c("gray", "green"),
hline_arb_label = "Hori_line_label",
vline_arb = c(.5, 1),
vline_arb_color = c("red", "black"),
vline_arb_label = c("Vertical Line A", "Vertical Line B"),
hline_vars = c("ANRHI.ALT", "ANRLO.ALT", "ULOQN.ALT", "LLOQN.ALT"),
hline_vars_colors = c("green", "blue", "purple", "cyan"),
hline_vars_labels = c("ANRHI ALT Label", "ANRLO ALT Label", "ULOQN ALT Label", "LLOQN ALT Label"),
vline_vars = c("ANRHI.CRP", "ANRLO.CRP", "ULOQN.CRP", "LLOQN.CRP"),
vline_vars_colors = c("yellow", "orange", "brown", "gold"),
vline_vars_labels = c("ANRHI CRP Label", "ANRLO CRP Label", "ULOQN CRP Label", "LLOQN CRP Label"),
rotate_xlab = FALSE,
font_size = 14,
dot_size = 2,
reg_text_size = 3
)

```

---

g\_density\_distribution\_plot

*Function to create a density distribution plot.*

---

## Description

Default plot displays overall density faceted by visit with treatment arms and combined treatment overlaid.

## Usage

```

g_density_distribution_plot(
  label = "Density Distribution Plot",
  data,
  param_var = "PARAMCD",
  param = "CRP",
  xaxis_var = "AVAL",
  trt_group = "ARM",
  unit = "AVALU",
  loq_flag_var = "LOQFL",
  xlim = c(NA, NA),
  ylim = c(NA, NA),

```

```

color_manual = NULL,
color_comb = "#39ff14",
comb_line = TRUE,
facet_var = "AVISITCD",
hline_arb = character(0),
hline_arb_color = "red",
hline_arb_label = "Horizontal line",
facet_ncol = 2,
rotate_xlab = FALSE,
font_size = 12,
line_size = 2,
rug_plot = FALSE
)

```

### Arguments

label	text string used to identify plot.
data	ADaM structured analysis laboratory data frame e.g. ADLB.
param_var	name of variable containing biomarker codes e.g. PARAMCD.
param	biomarker to visualize e.g. IGG.
xaxis_var	name of variable containing biomarker results displayed on X-axis e.g. AVAL.
trt_group	name of variable representing treatment group e.g. ARM.
unit	name of variable containing biomarker unit e.g. AVALU.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL.
xlim	(‘numeric vector’) optional, a vector of length 2 to specify the minimum and maximum of the x-axis if the default limits are not suitable.
ylim	(‘numeric vector’) optional, a vector of length 2 to specify the minimum and maximum of the y-axis if the default limits are not suitable.
color_manual	vector of colors applied to treatment values.
color_comb	name or hex value for combined treatment color.
comb_line	display combined treatment line toggle.
facet_var	variable to use for faceting.
hline_arb	(‘numeric vector’) value identifying intercept for arbitrary horizontal lines.
hline_arb_color	(‘character vector’) optional, color for the arbitrary horizontal lines.
hline_arb_label	(‘character vector’) optional, label for the legend to the arbitrary horizontal lines.
facet_ncol	number of facets per row.
rotate_xlab	45 degree rotation of x-axis label values.
font_size	font size control for title, x-axis label, y-axis label and legend.
line_size	plot line thickness.
rug_plot	should a rug plot be displayed under the density plot. Note this option is most useful if the data only contains a single treatment group.

**Author(s)**

Nick Paszty (npaszty) paszty.nicholas@gene.com

Balazs Toth (tothb2) toth.balazs@gene.com

**Examples**

```
# Example using ADaM structure analysis dataset.

library(stringr)

# original ARM value = dose value
arm_mapping <- list(
  "A: Drug X" = "150mg QD", "B: Placebo" = "Placebo", "C: Combination" = "Combination"
)
color_manual <- c("150mg QD" = "#000000", "Placebo" = "#3498DB", "Combination" = "#E74C3C")

ADLB <- rADLB
var_labels <- lapply(ADLB, function(x) attributes(x)$label)
ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
    AVISIT == "SCREENING" ~ "SCR",
    AVISIT == "BASELINE" ~ "BL",
    grepl("WEEK", AVISIT) ~
      paste(
        "W",
        trimws(
          substr(
            AVISIT,
            start = 6,
            stop = str_locate(AVISIT, "DAY") - 1
          )
        )
      ),
    TRUE ~ NA_character_
  )) %>%
  mutate(AVISITCDN = case_when(
    AVISITCD == "SCR" ~ -2,
    AVISITCD == "BL" ~ 0,
    grepl("W", AVISITCD) ~ as.numeric(gsub("\\d+", "", AVISITCD)),
    TRUE ~ NA_real_
  )) %>%
  # use ARMCD values to order treatment in visualization legend
  mutate(TRTORD = ifelse(grepl("C", ARMCD), 1,
    ifelse(grepl("B", ARMCD), 2,
      ifelse(grepl("A", ARMCD), 3, NA)
    )
  )) %>%
  mutate(ARM = as.character(arm_mapping[match(ARM, names(arm_mapping))])) %>%
  mutate(ARM = factor(ARM) %>%
    reorder(TRTORD))
attr(ADLB[["ARM"]], "label") <- var_labels[["ARM"]]
```

```

g_density_distribution_plot(
  label = "Density Distribution Plot",
  data = ADLB,
  param_var = "PARAMCD",
  param = c("CRP"),
  xaxis_var = "AVAL",
  unit = "AVALU",
  color_manual = color_manual,
  color_comb = "#39ff14",
  comb_line = FALSE,
  facet_var = "AVISITCD",
  hline_arb = 1.75,
  hline_arb_color = "black",
  hline_arb_label = "Horizontal Line A",
  facet_ncol = 2,
  rotate_xlab = FALSE,
  font_size = 10,
  line_size = .5
)

```

---

g\_lineplot

*Function to create line plot of summary statistics over time.*


---

## Description

Function to create line plot of summary statistics over time.

## Usage

```

g_lineplot(
  label = "Line Plot",
  data,
  biomarker_var = "PARAMCD",
  biomarker_var_label = "PARAM",
  biomarker,
  value_var = "AVAL",
  unit_var = "AVALU",
  loq_flag_var = "LOQFL",
  ylim = c(NA, NA),
  trt_group,
  trt_group_level = NULL,
  shape = NULL,
  shape_type = NULL,
  time,
  time_level = NULL,
  color_manual = NULL,
  line_type = NULL,
  median = FALSE,

```

```

hline_arb = numeric(0),
hline_arb_color = "red",
hline_arb_label = "Horizontal line",
xtick = ggplot2::waiver(),
xlabel = xtick,
rotate_xlab = FALSE,
plot_font_size = 12,
dot_size = 3,
dodge = 0.4,
plot_height = 989,
count_threshold = 0,
table_font_size = 12,
display_center_tbl = TRUE
)

```

### Arguments

label	text string to be displayed as plot label.
data	ADaM structured analysis laboratory data frame e.g. ADLB.
biomarker_var	name of variable containing biomarker names.
biomarker_var_label	name of variable containing biomarker labels.
biomarker	biomarker name to be analyzed.
value_var	name of variable containing biomarker results.
unit_var	name of variable containing biomarker result unit.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL.
ylim	(‘numeric vector’) optional, a vector of length 2 to specify the minimum and maximum of the y-axis if the default limits are not suitable.
trt_group	name of variable representing treatment group.
trt_group_level	vector that can be used to define the factor level of trt_group.
shape	categorical variable whose levels are used to split the plot lines.
shape_type	vector of symbol types.
time	name of variable containing visit names.
time_level	vector that can be used to define the factor level of time. Only use it when x-axis variable is character or factor.
color_manual	vector of colors.
line_type	vector of line types.
median	boolean whether to display median results.
hline_arb	(‘numeric vector’) value identifying intercept for arbitrary horizontal lines.
hline_arb_color	(‘character vector’) optional, color for the arbitrary horizontal lines.

hline_arb_label	('character vector') optional, label for the legend to the arbitrary horizontal lines.
xtick	a vector to define the tick values of time in x-axis. Default value is <code>ggplot2::waiver()</code> .
xlabel	vector with same length of <code>xtick</code> to define the label of x-axis tick values. Default value is <code>ggplot2::waiver()</code> .
rotate_xlab	boolean whether to rotate x-axis labels.
plot_font_size	control font size for title, x-axis, y-axis and legend font.
dot_size	plot dot size. Default to 3.
dodge	control position dodge.
plot_height	height of produced plot. 989 pixels by default.
count_threshold	integer minimum number observations needed to show the appropriate bar and point on the plot. Default: 0
table_font_size	float controls the font size of the values printed in the table. Default: 12
display_center_tbl	boolean whether to include table of means or medians

### Details

Currently, the output plot can display mean and median of input value. For mean, the error bar denotes 95\ quartile.

### Value

ggplot object

### Author(s)

Balazs Toth (toth.balazs@gene.com)

Wenyi Liu (wenyi.liu@roche.com)

### Examples

```
# Example using ADaM structure analysis dataset.

library(stringr)
library(dplyr)
library(nestcolor)

# original ARM value = dose value
arm_mapping <- list(
  "A: Drug X" = "150mg QD", "B: Placebo" = "Placebo", "C: Combination" = "Combination"
)
color_manual <- c("150mg QD" = "thistle", "Placebo" = "orange", "Combination" = "steelblue")
type_manual <- c("150mg QD" = "solid", "Placebo" = "dashed", "Combination" = "dotted")

ADSL <- rADSL %>% filter(!(ARM == "B: Placebo" & AGE < 40))
```



```

ADLB <- rADLB
ADLB <- right_join(ADLB, ADSL[, c("STUDYID", "USUBJID")])
var_labels <- lapply(ADLB, function(x) attributes(x)$label)

ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
    AVISIT == "SCREENING" ~ "SCR",
    AVISIT == "BASELINE" ~ "BL",
    grepl("WEEK", AVISIT) ~
      paste(
        "W",
        trimws(
          substr(
            AVISIT,
            start = 6,
            stop = str_locate(AVISIT, "DAY") - 1
          )
        )
      ),
    TRUE ~ NA_character_
  )) %>%
  mutate(AVISITCDN = case_when(
    AVISITCD == "SCR" ~ -2,
    AVISITCD == "BL" ~ 0,
    grepl("W", AVISITCD) ~ as.numeric(gsub("\\D+", "", AVISITCD)),
    TRUE ~ NA_real_
  )) %>%
  # use ARMCD values to order treatment in visualization legend
  mutate(TRTORD = ifelse(grepl("C", ARMCD), 1,
    ifelse(grepl("B", ARMCD), 2,
      ifelse(grepl("A", ARMCD), 3, NA)
    )
  )) %>%
  mutate(ARM = as.character(arm_mapping[match(ARM, names(arm_mapping))])) %>%
  mutate(ARM = factor(ARM) %>%
    reorder(TRTORD))
attr(ADLB[["ARM"]], "label") <- var_labels[["ARM"]]

g_lineplot(
  label = "Line Plot",
  data = ADLB,
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  shape = NULL,
  time = "AVISITCDN",
  color_manual = color_manual,
  line_type = type_manual,
  median = FALSE,
  hline_arb = c(.9, 1.1, 1.2, 1.5),
  hline_arb_color = c("green", "red", "blue", "pink"),
  hline_arb_label = c("A", "B", "C", "D"),

```

```

xtick = c(0, 1, 5),
xlabel = c("Baseline", "Week 1", "Week 5"),
rotate_xlab = FALSE,
plot_height = 600
)

g_lineplot(
  label = "Line Plot",
  data = ADLB,
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  shape = NULL,
  time = "AVISITCD",
  color_manual = NULL,
  line_type = type_manual,
  median = TRUE,
  hline_arb = c(.9, 1.1, 1.2, 1.5),
  hline_arb_color = c("green", "red", "blue", "pink"),
  hline_arb_label = c("A", "B", "C", "D"),
  xtick = c("BL", "W 1", "W 5"),
  xlabel = c("Baseline", "Week 1", "Week 5"),
  rotate_xlab = FALSE,
  plot_height = 600
)

g_lineplot(
  label = "Line Plot",
  data = ADLB,
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  shape = NULL,
  time = "AVISITCD",
  color_manual = color_manual,
  line_type = type_manual,
  median = FALSE,
  hline_arb = c(.9, 1.1, 1.2, 1.5),
  hline_arb_color = c("green", "red", "blue", "pink"),
  hline_arb_label = c("A", "B", "C", "D"),
  xtick = c("BL", "W 1", "W 5"),
  xlabel = c("Baseline", "Week 1", "Week 5"),
  rotate_xlab = FALSE,
  plot_height = 600,
  count_threshold = 90,
  table_font_size = 15
)

g_lineplot(
  label = "Line Plot",
  data = ADLB,

```

```
biomarker_var = "PARAMCD",
biomarker = "CRP",
value_var = "AVAL",
trt_group = "ARM",
shape = NULL,
time = "AVISITCDN",
color_manual = color_manual,
line_type = type_manual,
median = TRUE,
hline_arb = c(.9, 1.1, 1.2, 1.5),
hline_arb_color = c("green", "red", "blue", "pink"),
hline_arb_label = c("A", "B", "C", "D"),
xtick = c(0, 1, 5),
xlabel = c("Baseline", "Week 1", "Week 5"),
rotate_xlab = FALSE,
plot_height = 600
)
```

```
g_lineplot(
  label = "Line Plot",
  data = subset(ADLB, SEX %in% c("M", "F")),
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  shape = "SEX",
  time = "AVISITCDN",
  color_manual = color_manual,
  line_type = type_manual,
  median = FALSE,
  hline_arb = c(.9, 1.1, 1.2, 1.5),
  hline_arb_color = c("green", "red", "blue", "pink"),
  hline_arb_label = c("A", "B", "C", "D"),
  xtick = c(0, 1, 5),
  xlabel = c("Baseline", "Week 1", "Week 5"),
  rotate_xlab = FALSE,
  plot_height = 1500,
  dot_size = 1
)
```

```
g_lineplot(
  label = "Line Plot",
  data = subset(ADLB, SEX %in% c("M", "F")),
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  shape = "SEX",
  time = "AVISITCDN",
  color_manual = NULL,
  median = FALSE,
  hline_arb = c(.9, 1.1, 1.2, 1.5),
  hline_arb_color = c("green", "red", "blue", "pink"),
```

```
hline_arb_label = c("A", "B", "C", "D"),
xtick = c(0, 1, 5),
xlabel = c("Baseline", "Week 1", "Week 5"),
rotate_xlab = FALSE,
plot_height = 1500,
dot_size = 4
)
```

---

g\_scatterplot

*Function to create a scatter plot.*

---

## Description

### [Deprecated]

g\_scatterplot() is deprecated. Please use [g\\_correlationplot\(\)](#) instead. Default plot displays scatter faceted by visit with color attributed treatment arms and symbol attributed LOQ values.

## Usage

```
g_scatterplot(
  label = "Scatter Plot",
  data,
  param_var = "PARAMCD",
  param = "CRP",
  xaxis_var = "BASE",
  yaxis_var = "AVAL",
  trt_group = "ARM",
  visit = "AVISITCD",
  loq_flag_var = "LOQFL",
  unit = "AVALU",
  xlim = c(NA, NA),
  ylim = c(NA, NA),
  color_manual = NULL,
  shape_manual = NULL,
  facet_ncol = 2,
  facet = FALSE,
  facet_var = "ARM",
  reg_line = FALSE,
  hline = NULL,
  vline = NULL,
  rotate_xlab = FALSE,
  font_size = 12,
  dot_size = NULL,
  reg_text_size = 3
)
```

**Arguments**

label	text string to used to identify plot.
data	ADaM structured analysis laboratory data frame e.g. ADLB.
param_var	name of variable containing biomarker codes e.g. PARAMCD.
param	biomarker to visualize e.g. IGG.
xaxis_var	name of variable containing biomarker results displayed on X-axis e.g. BASE.
yaxis_var	name of variable containing biomarker results displayed on Y-axis e.g. AVAL.
trt_group	name of variable representing treatment group e.g. ARM.
visit	name of variable containing nominal visits e.g. AVISITCD.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL.
unit	name of variable containing biomarker unit e.g. AVALU.
xlim	('numeric vector') optional, a vector of length 2 to specify the minimum and maximum of the x-axis if the default limits are not suitable.
ylim	('numeric vector') optional, a vector of length 2 to specify the minimum and maximum of the y-axis if the default limits are not suitable.
color_manual	vector of colors applied to treatment values.
shape_manual	vector of symbols applied to LOQ values.
facet_ncol	number of facets per row.
facet	set layout to use treatment facetting.
facet_var	variable to use for treatment facetting.
reg_line	include regression line and annotations for slope and coefficient in visualization. Use with facet = TRUE.
hline	y-axis value to position a horizontal line.
vline	x-axis value to position a vertical line.
rotate_xlab	45 degree rotation of x-axis label values.
font_size	font size control for title, x-axis label, y-axis label and legend.
dot_size	plot dot size.
reg_text_size	font size control for regression line annotations.

**Details**

Regression uses deming model.

**Author(s)**

Nick Paszty (npaszty) paszty.nicholas@gene.com

Balazs Toth (tothb2) toth.balazs@gene.com

## Examples

```

# Example using ADaM structure analysis dataset.

library(stringr)

# original ARM value = dose value
arm_mapping <- list(
  "A: Drug X" = "150mg QD", "B: Placebo" = "Placebo", "C: Combination" = "Combination"
)
color_manual <- c("150mg QD" = "#000000", "Placebo" = "#3498DB", "Combination" = "#E74C3C")
# assign LOQ flag symbols: circles for "N" and triangles for "Y", squares for "NA"
shape_manual <- c("N" = 1, "Y" = 2, "NA" = 0)

ADLB <- rADLB
var_labels <- lapply(ADLB, function(x) attributes(x)$label)
ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
    AVISIT == "SCREENING" ~ "SCR",
    AVISIT == "BASELINE" ~ "BL",
    grepl("WEEK", AVISIT) ~
      paste(
        "W",
        trimws(
          substr(
            AVISIT,
            start = 6,
            stop = str_locate(AVISIT, "DAY") - 1
          )
        )
      ),
    TRUE ~ NA_character_
  )) %>%
  mutate(AVISITCDN = case_when(
    AVISITCD == "SCR" ~ -2,
    AVISITCD == "BL" ~ 0,
    grepl("W", AVISITCD) ~ as.numeric(gsub("\\d+", "", AVISITCD)),
    TRUE ~ NA_real_
  )) %>%
  # use ARMCD values to order treatment in visualization legend
  mutate(TRTORD = ifelse(grepl("C", ARMCD), 1,
    ifelse(grepl("B", ARMCD), 2,
      ifelse(grepl("A", ARMCD), 3, NA)
    )
  )) %>%
  mutate(ARM = as.character(arm_mapping[match(ARM, names(arm_mapping))])) %>%
  mutate(ARM = factor(ARM)) %>%
  reorder(TRTORD)
attr(ADLB[["ARM"]], "label") <- var_labels[["ARM"]]

g_scatterplot(
  label = "Scatter Plot",
  data = ADLB,

```

```
param_var = "PARAMCD",
param = c("ALT"),
xaxis_var = "BASE",
yaxis_var = "AVAL",
trt_group = "ARM",
visit = "AVISITCD",
loq_flag_var = "LOQFL",
unit = "AVALU",
color_manual = color_manual,
shape_manual = shape_manual,
facet_ncol = 2,
facet = TRUE,
facet_var = "ARM",
reg_line = TRUE,
hline = NULL,
vline = .5,
rotate_xlab = TRUE,
font_size = 14,
dot_size = 2,
reg_text_size = 3
)
```

---

g\_spaghettpoint

*Function to create a spaghetti plot.*

---

## Description

This function is rendered by teal.goshawk module

## Usage

```
g_spaghettpoint(
  data,
  subj_id = "USUBJID",
  biomarker_var = "PARAMCD",
  biomarker_var_label = "PARAM",
  biomarker,
  value_var = "AVAL",
  unit_var = "AVALU",
  trt_group,
  trt_group_level = NULL,
  loq_flag_var = "LOQFL",
  time,
  time_level = NULL,
  color_manual = NULL,
  color_comb = "#39ff14",
  ylim = c(NA, NA),
  alpha = 1,
  facet_ncol = 2,
```

```

facet_scales = c("fixed", "free", "free_x", "free_y"),
xtick = ggplot2::waiver(),
xlabel = xtick,
rotate_xlab = FALSE,
font_size = 12,
dot_size = 2,
group_stats = "NONE",
hline_arb = numeric(0),
hline_arb_color = "red",
hline_arb_label = "Horizontal line",
hline_vars = character(0),
hline_vars_colors = "green",
hline_vars_labels = hline_vars
)

```

### Arguments

data	data frame with variables to be summarized and generate statistics which will display in the plot.
subj_id	unique subject id variable name.
biomarker_var	name of variable containing biomarker names.
biomarker_var_label	name of variable containing biomarker labels.
biomarker	biomarker name to be analyzed.
value_var	name of variable containing biomarker results.
unit_var	name of variable containing biomarker units.
trt_group	name of variable representing treatment group.
trt_group_level	vector that can be used to define the factor level of trt_group.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL.
time	name of variable containing visit names.
time_level	vector that can be used to define the factor level of time. Only use it when x-axis variable is character or factor.
color_manual	vector of colors.
color_comb	name or hex value for combined treatment color.
ylim	('numeric vector') optional, a vector of length 2 to specify the minimum and maximum of the y-axis if the default limits are not suitable.
alpha	subject line transparency (0 = transparent, 1 = opaque)
facet_ncol	number of facets per row.
facet_scales	passed to scales in <code>ggplot2::facet_wrap</code> . Should scales be fixed ("fixed", the default), free ("free"), or free in one dimension ("free_x", "free_y")?
xtick	a vector to define the tick values of time in x-axis. Default value is <code>ggplot2::waiver()</code> .
xlabel	vector with same length of xtick to define the label of x-axis tick values. Default value is <code>ggplot2::waiver()</code> .



rotate_xlab	boolean whether to rotate x-axis labels.
font_size	control font size for title, x-axis, y-axis and legend font.
dot_size	plot dot size. Default to 2.
group_stats	control group mean or median overlay.
hline_arb	('numeric vector') value identifying intercept for arbitrary horizontal lines.
hline_arb_color	('character vector') optional, color for the arbitrary horizontal lines.
hline_arb_label	('character vector') optional, label for the legend to the arbitrary horizontal lines.
hline_vars	('character vector'), names of variables (ANR*) or values (*LOQ) identifying intercept values. The data inside of the ggplot2 object must also contain the columns with these variable names
hline_vars_colors	('character vector') colors for the horizontal lines defined by variables.
hline_vars_labels	('character vector') labels for the legend to the horizontal lines defined by variables.

**Value**

ggplot object

**Author(s)**

Wenyi Liu (wenyi.liu@roche.com)

**Examples**

```
# Example using ADaM structure analysis dataset.

library(stringr)

# original ARM value = dose value
arm_mapping <- list(
  "A: Drug X" = "150mg QD", "B: Placebo" = "Placebo", "C: Combination" = "Combination"
)
color_manual <- c("150mg QD" = "#000000", "Placebo" = "#3498DB", "Combination" = "#E74C3C")

ADLB <- rADLB
var_labels <- lapply(ADLB, function(x) attributes(x)$label)
ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
    AVISIT == "SCREENING" ~ "SCR",
    AVISIT == "BASELINE" ~ "BL",
    grepl("WEEK", AVISIT) ~
      paste(
        "W",
        trimws(
          substr(
```

```

        AVISIT,
        start = 6,
        stop = str_locate(AVISIT, "DAY") - 1
      )
    )
  ),
  TRUE ~ NA_character_
)) %>%
mutate(AVISITCDN = case_when(
  AVISITCD == "SCR" ~ -2,
  AVISITCD == "BL" ~ 0,
  grepl("W", AVISITCD) ~ as.numeric(gsub("\\d+", "", AVISITCD)),
  TRUE ~ NA_real_
)) %>%
# use ARMCD values to order treatment in visualization legend
mutate(TRTORD = ifelse(grepl("C", ARMCD), 1,
  ifelse(grepl("B", ARMCD), 2,
    ifelse(grepl("A", ARMCD), 3, NA)
  )
)) %>%
mutate(ARM = as.character(arm_mapping[match(ARM, names(arm_mapping))])) %>%
mutate(ARM = factor(ARM) %>%
  reorder(TRTORD)) %>%
mutate(ANRLO = .5, ANRHI = 1) %>%
rowwise() %>%
group_by(PARAMCD) %>%
mutate(LBSTRESC = ifelse(USUBJID %in% sample(USUBJID, 1, replace = TRUE),
  paste("<", round(runif(1, min = .5, max = .7))), LBSTRESC
)) %>%
mutate(LBSTRESC = ifelse(USUBJID %in% sample(USUBJID, 1, replace = TRUE),
  paste(">", round(runif(1, min = .9, max = 1.2))), LBSTRESC
)) %>%
ungroup()
attr(ADLB[["ARM"]], "label") <- var_labels[["ARM"]]
attr(ADLB[["ANRLO"]], "label") <- "Analysis Normal Range Lower Limit"
attr(ADLB[["ANRHI"]], "label") <- "Analysis Normal Range Upper Limit"

# add LLOQ and ULOQ variables
ADLB_LOQS <- goshawk::h_identify_loq_values(ADLB, "LOQFL")
ADLB <- left_join(ADLB, ADLB_LOQS, by = "PARAM")

g_spaghettoplot(
  data = ADLB,
  subj_id = "USUBJID",
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  time = "AVISITCD",
  color_manual = color_manual,
  color_comb = "#39ff14",
  alpha = .02,
  xtick = c("BL", "W 1", "W 4"),

```

```

    xlabel = c("Baseline", "Week 1", "Week 4"),
    rotate_xlab = FALSE,
    group_stats = "median",
    hline_vars = c("ANRHI", "ANRLO"),
    hline_vars_colors = c("pink", "brown")
  )

g_spaghettoplot(
  data = ADLB,
  subj_id = "USUBJID",
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  time = "AVISITCD",
  color_manual = color_manual,
  color_comb = "#39ff14",
  alpha = .02,
  xtick = c("BL", "W 1", "W 4"),
  xlabel = c("Baseline", "Week 1", "Week 4"),
  rotate_xlab = FALSE,
  group_stats = "median",
  hline_arb = 1.3,
  hline_vars = c("ANRHI", "ANRLO", "ULOQN", "LLOQN"),
  hline_vars_colors = c("pink", "brown", "purple", "gray"),
  dot_size = 3
)

g_spaghettoplot(
  data = ADLB,
  subj_id = "USUBJID",
  biomarker_var = "PARAMCD",
  biomarker = "CRP",
  value_var = "AVAL",
  trt_group = "ARM",
  time = "AVISITCDN",
  color_manual = color_manual,
  color_comb = "#39ff14",
  alpha = .02,
  xtick = c(0, 1, 4),
  xlabel = c("Baseline", "Week 1", "Week 4"),
  rotate_xlab = FALSE,
  group_stats = "median",
  hline_arb = c(.5, .7, 1),
  hline_arb_color = c("blue", "red", "green"),
  hline_arb_label = c("Arb_Hori_line_A", "Arb_Hori_line_B", "Arb_Hori_line_C"),
  hline_vars = c("ANRHI", "ANRLO"),
  dot_size = 4
)

# removing missing levels from the plot with facet_scales

g_spaghettoplot(

```

```

data = ADLB,
subj_id = "USUBJID",
biomarker_var = "PARAMCD",
biomarker = "CRP",
value_var = "AVAL",
trt_group = "ARM",
time = "RACE",
color_manual = color_manual,
color_comb = "#39ff14",
alpha = .02,
facet_scales = "fixed",
rotate_xlab = FALSE,
group_stats = "median",
hline_arb = c(.5, .7, 1),
hline_arb_color = c("blue", "red", "green"),
hline_arb_label = c("Arb_Hori_line_A", "Arb_Hori_line_B", "Arb_Hori_line_C"),
hline_vars = c("ANRHI", "ANRLO")
)

g_spaghettoplot(
data = ADLB,
subj_id = "USUBJID",
biomarker_var = "PARAMCD",
biomarker = "CRP",
value_var = "AVAL",
trt_group = "ARM",
time = "RACE",
color_manual = color_manual,
color_comb = "#39ff14",
alpha = .02,
facet_scales = "free_x",
rotate_xlab = FALSE,
group_stats = "median",
hline_arb = c(.5, .7, 1),
hline_arb_color = c("blue", "red", "green"),
hline_arb_label = c("Arb_Hori_line_A", "Arb_Hori_line_B", "Arb_Hori_line_C"),
hline_vars = c("ANRHI", "ANRLO"),
dot_size = 1
)

```

---

t\_summarytable

*Function to create a table of descriptive summary statistics to accompany plots.*

---

### Description

Output descriptive summary statistics table as a data frame. Includes biomarker, treatment, visit, n, mean, median, SD, min, max, %missing values, % LOQ values.

**Usage**

```
t_summarytable(
  data,
  trt_group,
  param_var,
  param,
  xaxis_var,
  facet_var = "AVISITCD",
  loq_flag_var = "LOQFL",
  ...
)
```

**Arguments**

data	name of data frame to summarize.
trt_group	treatment group variable name e.g. ARM.
param_var	name of variable containing biomarker codes e.g. PARAMCD.
param	biomarker to visualize e.g. IGG.
xaxis_var	name of variable containing biomarker results displayed on X-axis e.g. AVAL.
facet_var	name of variable faceted on typically containing visit values e.g. AVISITCD. If NULL then ignored. It defaults to "AVISITCD" when not provided.
loq_flag_var	name of variable containing LOQ flag e.g. LOQFL. Defaults to "LOQFL".
...	additional options

**Details**

provide additional information as needed. link to specification file <https://posit.co/>

**Author(s)**

Nick Paszty (npaszty) paszty.nicholas@gene.com  
Balazs Toth (tothb2) toth.balazs@gene.com

**Examples**

```
# Example using ADaM structure analysis dataset.

library(stringr)

# original ARM value = dose value
arm_mapping <- list(
  "A: Drug X" = "150mg QD", "B: Placebo" = "Placebo", "C: Combination" = "Combination"
)

ADLB <- rADLB
ADLB <- ADLB %>%
  mutate(AVISITCD = case_when(
```

```

AVISIT == "SCREENING" ~ "SCR",
AVISIT == "BASELINE" ~ "BL",
grepl("WEEK", AVISIT) ~
  paste(
    "W",
    trimws(
      substr(
        AVISIT,
        start = 6,
        stop = str_locate(AVISIT, "DAY") - 1
      )
    )
  ),
  TRUE ~ NA_character_
)) %>%
mutate(AVISITCDN = case_when(
  AVISITCD == "SCR" ~ -2,
  AVISITCD == "BL" ~ 0,
  grepl("W", AVISITCD) ~ as.numeric(gsub("\\D+", "", AVISITCD)),
  TRUE ~ NA_real_
)) %>%
# use ARMCD values to order treatment in visualization legend
mutate(TRTORD = ifelse(grepl("C", ARMCD), 1,
  ifelse(grepl("B", ARMCD), 2,
    ifelse(grepl("A", ARMCD), 3, NA)
  )
)) %>%
mutate(ARM = as.character(arm_mapping[match(ARM, names(arm_mapping))])) %>%
mutate(ARM = factor(ARM) %>%
  reorder(TRTORD))

tbl <- t_summarytable(
  data = ADLB,
  trt_group = "ARM",
  param_var = "PARAMCD",
  param = c("CRP"),
  xaxis_var = "AVAL",
  facet_var = "AVISITCD",
  loq_flag_var = "LOQFL"
)
tbl

```

# Index

`g_boxplot`, [2](#)  
`g_correlationplot`, [5](#)  
`g_correlationplot()`, [20](#)  
`g_density_distribution_plot`, [11](#)  
`g_lineplot`, [14](#)  
`g_scatterplot`, [20](#)  
`g_spaghettoplot`, [23](#)  
`ggplot2::facet_wrap`, [24](#)  
  
`t_summarytable`, [28](#)