

Package ‘coveffectsplot’

May 8, 2026

Title Produce Forest Plots to Visualize Covariate Effects

Version 1.0.7

Description Produce forest plots to visualize covariate effects using either the command line or an interactive 'Shiny' application.

URL <https://smouksassi.github.io/coveffectsplot/>,
<https://github.com/smouksassi/coveffectsplot>

BugReports <https://github.com/smouksassi/coveffectsplot/issues>

Depends R (>= 4.0.0)

Imports colourpicker, data.table (>= 1.9.8), egg, grid, ggplot2 (>= 4.0.0), shiny, stats, utils

Suggests markdown, dplyr, tidyr, shinyjs, shinymeta, table1, clipr, formatR, MASS, knitr, rmarkdown, mrgsolve, GGally, ggridges, ggrepel, ggstance, patchwork, plotly, scales, shinyAce, Rcpp, gamlss.dist, ggdist, ggh4x, ggpmisc, quantreg

License MIT + file LICENSE

SystemRequirements pandoc with https support

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.3

Encoding UTF-8

NeedsCompilation no

Author Samer Mouksassi [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-7152-6654>>),
Benjamin Rich [aut],
Dean Attali [ctb]

Maintainer Samer Mouksassi <samermouksassi@gmail.com>

Repository CRAN

Date/Publication 2025-09-16 00:00:02 UTC

Contents

covdatasim	2
deltamethod	3
draw_key	4
expand_modelframe	5
forest_plot	6
get_sample_data	15
prezista	16
run_interactiveforestplot	17
wtage	17
Index	19

covdatasim	<i>Correlated Covariates data</i>
------------	-----------------------------------

Description

A example dataset used to illustrate multivariate joint covariate effects.

Usage

```
covdatasim
```

Format

A dataset with 2000 rows and 5 variables

ID Subject ID

AGE Age in years

WT Weight in kg

Sex 0=male; 1=female

ALB Albumin in g/dL

Source

simulated based on a real dataset

deltamethod	<i>The delta method</i>
-------------	-------------------------

Description

This is a copy of `msm` `deltamethod` function with the modification to add an environment to make it usable within a function or within a shiny app. Delta method for approximating the standard error of a transformation $g(X)$ of a random variable $X = (x_1, x_2, \dots)$, given estimates of the mean and covariance matrix of X .

Usage

```
deltamethod(g, mean, cov, ses = TRUE, envir = parent.frame())
```

Arguments

<code>g</code>	A formula representing the transformation. The variables must be labelled <code>x1</code> , <code>x2</code> , ... <code>{}</code> For example, $\sim 1 / (x1 + x2)$ If the transformation returns a vector, then a list of formulae representing (g_1, g_2, \dots) can be provided, for example <code>list(~ x1 + x2, ~ x1 / (x1 + x2))</code>
<code>mean</code>	The estimated mean of X
<code>cov</code>	The estimated covariance matrix of X
<code>ses</code>	If <code>TRUE</code> , then the standard errors of $g_1(X), g_2(X), \dots$ are returned. Otherwise the covariance matrix of $g(X)$ is returned.
<code>envir</code>	defaults to <code>parent.frame()</code>

Details

The delta method expands a differentiable function of a random variable about its mean, usually with a first-order Taylor approximation, and then takes the variance. For example, an approximation to the covariance matrix of $g(X)$ is given by

$$Cov(g(X)) = g'(\mu)Cov(X)[g'(\mu)]^T$$

where μ is an estimate of the mean of X . This function uses symbolic differentiation via `deriv`.

A limitation of this function is that variables created by the user are not visible within the formula `g`. To work around this, it is necessary to build the formula as a string, using functions such as `sprintf`, then to convert the string to a formula using `as.formula`. See the example below.

If you can spare the computational time, bootstrapping is a more accurate method of calculating confidence intervals or standard errors for transformations of parameters. Simulation from the asymptotic distribution of the MLEs (see e.g. Mandel 2013) is also a convenient alternative.

Value

A vector containing the standard errors of $g_1(X), g_2(X), \dots$ or a matrix containing the covariance of $g(X)$.

Author(s)

C. H. Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Oehlert, G. W. (1992) *A note on the delta method*. American Statistician 46(1).

Mandel, M. (2013) *Simulation based confidence intervals for functions with complicated derivatives*. The American Statistician 67(2):76-81.

Examples

```
## Simple linear regression, E(y) = alpha + beta x
x <- 1:100
y <- rnorm(100, 4*x, 5)
toy.lm <- lm(y ~ x)
estmean <- coef(toy.lm)
estvar <- summary(toy.lm)$cov.unscaled * summary(toy.lm)$sigma^2

## Estimate of (1 / (alphahat + betahat))
1 / (estmean[1] + estmean[2])
## Approximate standard error
deltamethod (~ 1 / (x1 + x2), estmean, estvar)

## We have a variable z we would like to use within the formula.
z <- 1
## deltamethod (~ z / (x1 + x2), estmean, estvar) will not work.
## Instead, build up the formula as a string, and convert to a formula.
form <- sprintf("~ %f / (x1 + x2)", z)
form
deltamethod(as.formula(form), estmean, estvar)
```

draw_key

Horizontal key drawing functions from ggstance in case it is deprecated

Description

Horizontal key drawing functions from ggstance in case it is deprecated

Usage

```
draw_key_hpath(data, params, size)

draw_key_pointrangeh(data, params, size)
```

Arguments

data	A single row data frame containing the scaled aesthetics to display in this key
params	A list of additional parameters supplied to the geom.
size	Width and height of key in mm.

Value

A grid grob.

expand_modelframe	<i>Expand covariate values choices and reference values varying one at a time</i>
-------------------	---

Description

Expand covariate values choices and reference values varying one at a time

Usage

```
expand_modelframe(rv, covcol = "covname", ...)
```

Arguments

rv	a data.frame with columns names of covariate(s) and values equal reference
covcol	column name for the covariate being varied
...	Arguments to be passed to methods

Value

A data.frame with combination of covariates

Examples

```
reference.values <- data.frame(WT = 85, ALB = 45, SEX = 0)
covcomb <- expand_modelframe(
  WT = c(56, 72, 98, 128), # P05, P25, P75, P95 # ref is P50
  ALB = c(40, 50),        # P05, P95 # ref is P50
  SEX = c(1),            # Reference is for SEX=0 (female)
  rv = reference.values)
covcomb
```

`forest_plot`*Forest plot*

Description

Produce forest plots to visualize covariate effects

Usage

```
forest_plot(  
  data,  
  facet_formula = "covname~paramname",  
  xlabel = "",  
  ylabel = "",  
  x_facet_text_size = 13,  
  y_facet_text_size = 13,  
  x_facet_text_angle = 0,  
  y_facet_text_angle = 0,  
  x_facet_text_vjust = 0.5,  
  y_facet_text_vjust = 0.5,  
  x_facet_text_hjust = 0.5,  
  y_facet_text_hjust = 0.5,  
  x_facet_text_col = "black",  
  y_facet_text_col = "black",  
  xy_facet_text_bold = TRUE,  
  x_label_text_size = 16,  
  y_label_text_size = 16,  
  legend_title_size = 12,  
  break_ylabel = FALSE,  
  y_label_text_width = 25,  
  table_text_size = 7,  
  table_text_colour_overwrite = FALSE,  
  table_text_colour = "none",  
  base_size = 22,  
  theme_benrich = FALSE,  
  table_title = "",  
  table_title_size = 15,  
  ref_legend_text = "",  
  area_legend_text = "",  
  interval_legend_text = "",  
  interval_legend_title = "",  
  shape_legend_title = "",  
  legend_order = c("pointinterval", "ref", "area", "shape"),  
  combine_area_ref_legend = TRUE,  
  combine_interval_shape_legend = FALSE,  
  legend_position = "top",  
  show_ref_area = TRUE,  
)
```

```
ref_area = c(0.8, 1.25),
ref_area_col = "#BEBEBE50",
show_ref_value = TRUE,
ref_value = 1,
ref_value_col = "black",
ref_value_size = 1,
ref_value_linetype = "dashed",
ref_value_by_panel = FALSE,
ref_value_by_panel_data = NULL,
interval_col = "blue",
interval_size = 1,
interval_fatten = 4,
interval_linewidth = 1,
interval_shape = "circle small",
bsv_col = "red",
bsv_shape = "circle small",
bsv_text_id = c("BSV", "bsv", "IIV", "Bsv"),
interval_bsv_text = "",
strip_col = "#E5E5E5",
paramname_shape = FALSE,
paramname_color = FALSE,
legend_shape_reverse = FALSE,
legend_color_reverse = FALSE,
facet_switch = c("both", "y", "x", "none"),
facet_scales = c("fixed", "free_y", "free_x", "free"),
facet_space = c("fixed", "free_x", "free_y", "free"),
facet_labeller = "label_value",
label_wrap_width = 55,
facet_labeller_multiline = FALSE,
strip_placement = c("inside", "outside"),
strip_outline = TRUE,
facet_spacing = 5.5,
major_x_ticks = NULL,
major_x_labels = NULL,
minor_x_ticks = NULL,
x_range = NULL,
logxscale = FALSE,
show_yaxis_gridlines = TRUE,
show_xaxis_gridlines = TRUE,
show_table_facet_strip = "none",
table_facet_switch = c("both", "y", "x", "none"),
show_table_yaxis_tick_label = FALSE,
reserve_table_xaxis_label_space = TRUE,
table_panel_border = TRUE,
table_position = c("right", "below", "none"),
plot_table_ratio = 4,
vertical_dodge_height = 0.8,
legend_space_x_mult = 1,
```

```

legend_ncol_interval = 1,
legend_ncol_shape = 1,
plot_margin = c(5.5, 5.5, 5.5, 5.5),
table_margin = c(5.5, 5.5, 5.5, 5.5),
legend_margin = c(0, 0.1, -0.1, 0),
parse_xlabel = FALSE,
parse_ylabel = FALSE,
plot_title = "\n",
return_list = FALSE
)

```

Arguments

<code>data</code>	Data to use.
<code>facet_formula</code>	Facet formula.
<code>xlabel</code>	X axis title.
<code>ylabel</code>	Y axis title.
<code>x_facet_text_size</code>	Facet text size X.
<code>y_facet_text_size</code>	Facet text size Y.
<code>x_facet_text_angle</code>	Facet text angle X.
<code>y_facet_text_angle</code>	Facet text angle Y.
<code>x_facet_text_vjust</code>	Facet text vertical justification.
<code>y_facet_text_vjust</code>	Facet text vertical justification.
<code>x_facet_text_hjust</code>	Facet text horizontal justification.
<code>y_facet_text_hjust</code>	Facet text horizontal justification.
<code>x_facet_text_col</code>	Facet text color default to black.
<code>y_facet_text_col</code>	Facet text color default to black.
<code>xy_facet_text_bold</code>	Bold Facet text. Logical TRUE FALSE.
<code>x_label_text_size</code>	X axis labels size.
<code>y_label_text_size</code>	Y axis labels size.
<code>legend_title_size</code>	Legend title size if present.
<code>break_ylabel</code>	Split Y axis labels into multiple lines. Logical FALSE TRUE.

y_label_text_width Number of characters to break Y axis labels.
 table_text_size Table text size.
 table_text_colour_overwrite Logical TRUE FALSE.
 table_text_colour Table text color to be used and overwrites mapped color
 base_size theme_bw base_size for the plot and table.
 theme_benrich apply Benjamin Rich's theming.
 table_title What text to use for table title (theme_benrich has a default).
 table_title_size table title size.
 ref_legend_text Reference legend text.
 area_legend_text Area legend text.
 interval_legend_text Pointinterval legend text.
 interval_legend_title Pointinterval legend title defaults to empty.
 shape_legend_title Shape legend title defaults to empty.
 legend_order Legend order. A four-element vector with the following items ordered in your desired order: "pointinterval", "ref", "area", "shape". if an item is absent the legend will be omitted.
 combine_area_ref_legend Combine reference and area legends if they share the same text?
 combine_interval_shape_legend Combine interval and shape legends?
 legend_position where to put the legend: "top", "bottom", "right", "none"
 show_ref_area Show reference window?
 ref_area Reference area. Two-element numeric vector multiplying the ref_value.
 ref_area_col Reference area background color.
 show_ref_value Show reference line?
 ref_value X intercept of reference line.
 ref_value_col Reference line color.
 ref_value_size Reference line size.
 ref_value_linetype Reference line linetype.
 ref_value_by_panel The ref_value vary by panel TRUE or FALSE.

ref_value_by_panel_data
 if ref_value_by_panel is TRUE, data.frame to use for Reference (lines).

interval_col Point range color. One or Multiple values.

interval_size Point range size. Default to 1

interval_fatten
 Point range fatten. Default to 4

interval_linewidth
 Point range line width. Default to 1

interval_shape Shape used for the Point Range. Default to "circle small".

bsv_col BSV pointinterval color. One value.

bsv_shape Shape used for the BSV Point Range. Default to "circle small".

bsv_text_id Text string(s) to identify BSV. Default to c("BSV", "bsv", "IIV", "Bsv")

interval_bsv_text
 BSV legend text.

strip_col Strip background color.

paramname_shape
 Map symbol to parameter(s) name? TRUE or FALSE.

paramname_color
 Map color to parameter(s) name? TRUE or FALSE.

legend_shape_reverse
 TRUE or FALSE.

legend_color_reverse
 TRUE or FALSE.

facet_switch Facet switch to near axis. Possible values: "both", "y", "x", "none".

facet_scales Facet scales. Possible values: "free_y", "fixed", "free_x", "free".

facet_space Facet spaces. Possible values: "fixed", "free_x", "free_y", "free".

facet_labeller Facet Labeller. Default "label_value" any other valid 'facet_grid' labeller can be specified.

label_wrap_width
 How many characters before breaking the line. Numeric value. any other valid 'facet_grid' labeller can be specified.

facet_labeller_multiline
 break facet strips into multiple lines. Logical TRUE FALSE.

strip_placement
 Strip placement. Possible values: "inside", "outside".

strip_outline Draw rectangle around the Strip. Logical TRUE FALSE.

facet_spacing Control the space between facets in points.

major_x_ticks X axis major ticks. Numeric vector.

major_x_labels X axis labels. Character vector should be same length as major_x_ticks.

minor_x_ticks X axis minor ticks. Numeric vector.

x_range Range of X values. Two-element numeric vector.

logxscale	X axis log scale. Logical TRUE FALSE.
show_yaxis_gridlines	Draw the y axis gridlines. Logical TRUE FALSE.
show_xaxis_gridlines	Draw the x axis gridlines. Logical TRUE FALSE.
show_table_facet_strip	Possible values: "none", "both", "y", "x".
table_facet_switch	Table facet switch to near axis. Possible values: "both", "y", "x", "none".
show_table_yaxis_tick_label	Show table y axis ticks and labels?
reserve_table_xaxis_label_space	keep space for the x axis label to keep alignment.
table_panel_border	Draw the panel border for the table. Logical TRUE FALSE.
table_position	Table position. Possible values: "right", "below", "none".
plot_table_ratio	Plot-to-table ratio. Suggested value between 1-5.
vertical_dodge_height	Amount of vertical dodging to apply on segments and table text.
legend_space_x_mult	Multiplier to adjust the spacing between legend items.
legend_ncol_interval	Control the number of columns for the pointinterval legend.
legend_ncol_shape	Control the number of columns for the shape legend.
plot_margin	Control the white space around the main plot. Vector of four numeric values for the top, right, bottom and left sides.
table_margin	Control the white space around the table. Vector of four numeric values for the top, right, bottom and left sides.
legend_margin	Control the white space around the plot legend. Vector of four numeric values for the top, right, bottom and left sides.
parse_xlabel	treat xlabel as an expression. Logical FALSE TRUE.
parse_ylabel	treat ylabel as an expression. Logical FALSE TRUE.
plot_title	main plot title default to a line break.
return_list	What to return if True a list of the main and table plots is returned instead of the gtable/plot.

Examples

```
library(dplyr)
library(ggplot2)

# Example 1
```

```

plotdata <- get_sample_data("forest-plot-table.csv")
plotdata <- plotdata %>%
  mutate(midlabel = format(round(mid,2), nsmall = 2),
         lowerlabel = format(round(lower,2), nsmall = 2),
         upperlabel = format(round(upper,2), nsmall = 2),
         LABEL = paste0(midlabel, " [", lowerlabel, "-", upperlabel, "]"))
param <- "BZD AUC"
plotdata <- filter(plotdata,paramname==param)
plotdata$covname <- reorder(plotdata$covname,plotdata$upper,FUN =max)
plotdata$label <- reorder(plotdata$label,plotdata$scen)
covs <- c("WEIGHT","AGE")
plotdata <- filter(plotdata,covname%in%covs)
forest_plot(plotdata,
            ref_legend_text = "Reference (vertical line)",
            area_legend_text = "Reference (vertical line)",
            xlabel = paste("Fold Change in", param, "Relative to Reference"),
            logxscale = TRUE, major_x_ticks =c(0.1,1,1.5),
            show_ref_area = FALSE,
            paramname_color =TRUE,
            interval_col =c("steelblue","red","steelblue","red"),
            facet_formula = "covname~.",
            facet_scales = "free_y",
            facet_space = "free_y",
            show_table_facet_strip = "none",
            table_position = "right",
            plot_title = "",
            plot_table_ratio = 4)

# Example 2

plotdata <- get_sample_data("forest-plot-table.csv")
plotdata <- plotdata %>%
  mutate(midlabel = format(round(mid,2), nsmall = 2),
         lowerlabel = format(round(lower,2), nsmall = 2),
         upperlabel = format(round(upper,2), nsmall = 2),
         LABEL = paste0(midlabel, " [", lowerlabel, "-", upperlabel, "]"))
param <- c("BZD AUC","BZD Cmax")
plotdata <- filter(plotdata,paramname%in%param)
plotdata <- filter(plotdata,covname%in%"WEIGHT")
plotdata$covname <- reorder(plotdata$covname,plotdata$upper,FUN =max)
plotdata$label <- reorder(plotdata$label,plotdata$scen)
forest_plot(plotdata,
            ref_legend_text = "Reference (vertical line)",
            area_legend_text = "Reference (vertical line)",
            xlabel = paste("Fold Change of Parameter", "Relative to Reference"),
            show_ref_area = FALSE,
            facet_formula = "covname~paramname",
            facet_scales = "free_y",
            facet_space = "free_y",
            x_facet_text_size = 10,
            y_facet_text_size = 10,
            y_label_text_size = 10,

```

```

        y_label_text_width = 15,
        x_label_text_size = 10,
        facet_switch = "both",
        show_table_facet_stripe = "both",
        show_table_yaxis_tick_label = TRUE,
        table_position = "below",
        plot_title = "",
        plot_table_ratio = 1)
## Not run:

# Example 3a

plotdata <- get_sample_data("forest-plot-table.csv")
plotdata <- plotdata %>%
  mutate(midlabel = format(round(mid,2), nsmall = 2),
         lowerlabel = format(round(lower,2), nsmall = 2),
         upperlabel = format(round(upper,2), nsmall = 2),
         LABEL = paste0(midlabel, " [", lowerlabel, "-", upperlabel, "]"))
plotdata$covname <- reorder(plotdata$covname,plotdata$upper,FUN =max)
plotdata$label <- reorder(plotdata$label,plotdata$scen)

plotdata$compound <- c(rep("1-OH",30),rep("BZD",30))
plotdata$paramname <- c(rep("AUC",15),rep("CMAX",15),rep("AUC",15),rep("CMAX",15))

forest_plot(plotdata,
            ref_area = c(0.8, 1.2),
            x_facet_text_size = 13,
            y_facet_text_size = 13,
            ref_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
            area_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
            xlabel = "Fold Change Relative to Parameter",
            facet_formula = covname~compound,
            facet_switch = "both",
            facet_scales = "free",
            facet_space = "fixed",
            paramname_shape = TRUE,
            legend_shape_reverse = TRUE,
            interval_shape = c("square","triangle"),
            paramname_color = FALSE,
            combine_interval_shape_legend = FALSE,
            table_position = "right", plot_title = "",
            ref_area_col = rgb( col2rgb("gray50")[1], col2rgb("gray50")[2],col2rgb("gray50")[3],
                               max = 255, alpha = 0.1*255 ) ,
            interval_col = c("steelblue"),
            strip_col = "lightblue",
            plot_table_ratio = 1.5)

# Example 3b

plotdata$paramname <- c(rep("1-OH",30),rep("BZD",30))
plotdata$paramname2 <- c(rep("AUC",15),rep("CMAX",15),rep("AUC",15),rep("CMAX",15))
forest_plot(plotdata,
            show_ref_area = TRUE,

```

```

x_facet_text_size = 13,
y_facet_text_size = 13,
ref_legend_text = "Reference (vertical line)",
area_legend_text = "Reference (vertical line)",
xlabel = "Fold Change Relative to Parameter",
facet_formula = covname~paramname2,
facet_switch = "both",
facet_scales = "free",
facet_space = "free",
legend_order = c("shape", "pointinterval", "ref"),
paramname_shape = TRUE,
interval_shape = c("diamond", "diamond filled",
                  "diamond", "diamond filled"),
paramname_color = TRUE,
combine_interval_shape_legend = TRUE,
legend_shape_reverse = TRUE,
legend_color_reverse = TRUE,
interval_legend_title="Median (points)\n95% CI (horizontal lines)",
table_position = "right", plot_title = "",
ref_area_col = "gray85" ,
interval_col = c("#ee3124", "#fdbb2f"),
strip_col = "#475c6b",
y_facet_text_col = "white", x_facet_text_col = "white",
major_x_labels = c("1/2", "0.8", "1", "1.25", "2"),
logxscale = TRUE, major_x_ticks = c(0.5, 0.8, 1, 1.25, 2),
table_text_size = 5,
plot_table_ratio = 1.5,
ref_value_by_panel = TRUE,
ref_value_by_panel_data = as.data.frame(
  plotdata %>%
  distinct(paramname2, covname) %>%
  dplyr::mutate(xintercept=ifelse(paramname2=="CMAX", 1, 1.2)))

```

Example 3

```

plotdata <- get_sample_data("forestplotdatacpidata.csv")
forest_plot(plotdata,
  ref_area = c(0.8, 1.2),
  x_facet_text_size = 12,
  y_facet_text_size = 12,
  y_label_text_size = 10,
  x_label_text_size = 10,
  table_text_size = 6,
  plot_table_ratio = 1.5,
  ref_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
  area_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
  xlabel = "Fold Change Relative to RHZE",
  facet_formula = "covname~paramname",
  table_position = "below",
  show_table_facet_strip = "both",
  show_table_yaxis_tick_label = TRUE)

```

Example 4

```

plotdata <- get_sample_data("dataforest.csv")
plotdata <- plotdata %>%
  dplyr::mutate(midlabel = format(round(mid,2), nsmall = 2),
               lowerlabel = format(round(lower,2), nsmall = 2),
               upperlabel = format(round(upper,2), nsmall = 2),
               LABEL = paste0(midlabel, " [", lowerlabel, "-", upperlabel, "]"))
plotdata <- plotdata %>%
  filter(covname%in%c("Weight"))
plotdata$label <- as.factor(as.character(plotdata$label))
plotdata$label <- factor(plotdata$label, c("36.2 kg", "66 kg", "110 kg"))
forest_plot(plotdata,
            ref_area = c(0.8, 1.2),
            x_facet_text_size = 13,
            y_facet_text_size = 13,
            ref_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
            area_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
            xlabel = "Fold Change Relative to Parameter",
            facet_formula = "covname~paramname",
            facet_switch = "both",
            facet_scales = "free",
            facet_space = "fixed",
            table_position = "below",
            plot_table_ratio = 1,
            show_table_facet_stripe = "both",
            show_table_yaxis_tick_label = TRUE)

# Example 5

forest_plot(plotdata,
            ref_area = c(0.8, 1.2),
            x_facet_text_size = 13,
            y_facet_text_size = 13,
            ref_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
            area_legend_text = "Reference (vertical line)\n+/- 20% limits (colored area)",
            xlabel = "Fold Change Relative to Parameter",
            facet_formula = "covname~.",
            facet_switch = "both",
            facet_scales = "free",
            facet_space = "fixed",
            paramname_shape = TRUE,
            table_position = "none",
            ref_area_col = rgb( col2rgb("gray50")[1], col2rgb("gray50")[2], col2rgb("gray50")[3],
                               max = 255, alpha = 0.1*255 ) ,
            interval_col = "steelblue",
            stripe_col = "lightblue",
            plot_table_ratio = 1)

## End(Not run)

```

Description

Get a sample dataset that is included with the package to plot a forest plot.

Usage

```
get_sample_data(dataset = "dfall.csv")
```

Arguments

dataset A sample dataset file.

```
prezista
```

```
Prezista Drug Label Data
```

Description

A dataset containing an excerpt from the official Prezista FDA Drug Label to help in the app exploration.

Usage

```
prezista
```

Format

A dataset with 33 rows and 6 variables

covname Covariate Name, a character variable with two values Protease Inhibitors and Other Antiretrovirals

label Covariate value label, a character variable with several possible values

paramname Parameter on which the effects are shown, a character variable with three possible values Cmax, AUC and Cmin

mid Middle value for the effects, the median from the uncertainty distribution

lower Lower value for the effects usually the 5% from the uncertainty distribution

upper Upper value for the effects usually the 95% from the uncertainty distribution

Source

Table 16 from https://www.accessdata.fda.gov/drugsatfda_docs/label/2017/021976s045_202895s0201b1.pdf

`run_interactiveforestplot`*Run the interactiveforestplot application*

Description

Run the interactiveforestplot application.

Usage

```
run_interactiveforestplot(data = NULL)
```

Arguments

`data` optional data to load when the app is launched

Examples

```
if (interactive()) {  
  run_interactiveforestplot()  
}
```

`wtage`*Weight Age CDC growth charts data*

Description

Weight-for-age, 2 to 20 years, LMS parameters and selected smoothed weight percentiles in kilograms, by sex and age.

Usage`wtage`**Format**

A dataset with 436 rows and 14 variables

Sex 1=male; 2=female

Agemos Age in months

L skewness distribution parameter

M location distribution parameter

S scale distribution parameter

P3 Smoothed third percentile

- P5** Smoothed fifth percentile
- P10** Smoothed tenth percentile
- P25** Smoothed twenty fifth percentile
- P50** Smoothed fiftieth percentile
- P75** Smoothed seventy fifth percentile
- P90** Smoothed ninetieth percentile
- P95** Smoothed ninety fifth percentile
- P97** Smoothed ninety seventh percentile

Source

CDC website <https://www.cdc.gov/growthcharts/data/zscore/wtage.csv>

Index

* **datasets**

covdatasim, [2](#)

prezista, [16](#)

wtage, [17](#)

* **math**

deltamethod, [3](#)

covdatasim, [2](#)

deltamethod, [3](#)

deriv, [3](#)

draw_key, [4](#)

draw_key_hpath (draw_key), [4](#)

draw_key_pointrangeh (draw_key), [4](#)

expand_modelframe, [5](#)

forest_plot, [6](#)

get_sample_data, [15](#)

prezista, [16](#)

run_interactiveforestplot, [17](#)

wtage, [17](#)