Package ‘forestly’

June 12, 2026

Title Interactive Forest Plot

Version 0.1.5

Description Interactive forest plot for clinical trial safety analysis using 'metalite', 'reactable', 'plotly', and Analysis Data Model (ADaM) datasets. Includes functionality for adverse event filtering, incidence-based group filtering, hover-over reveals, and search and sort operations. The workflow allows for metadata construction, data preparation, output formatting, and interactive plot generation.

License GPL (>= 3)

URL `https://merck.github.io/forestly/`,
`https://github.com/Merck/forestly`

BugReports `https://github.com/Merck/forestly/issues`

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports brew, crosstalk, glue, htmltools, metalite, metalite.ae, reactable, reactR, rlang, ggplot2, uuid

Suggests covr, dplyr, knitr, r2rtf, rmarkdown, patchwork, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Yilong Zhang [aut],
Benjamin Wang [aut],
Yujie Zhao [aut, cre],
Nan Xiao [ctb],
Hiroaki Fukuda [aut],
Yulia Sidi [ctb],
Xuan Deng [ctb],

Jeetener Chauhan [ctb],
Li Ma [ctb],
Chen Wang [ctb],
Madhusudhan Ginnaram [ctb],
Merck & Co., Inc., Rahway, NJ, USA and its affiliates [cph]

Maintainer Yujie Zhao <`yujie.zhao@merck.com`>

Repository CRAN

Date/Publication 2026-06-12 07:10:20 UTC

Contents

`ae_forestly` *Display interactive forest plot*

Description

Display interactive forest plot

Usage

```
ae_forestly(
  outdata,
  display_soc_toggle = TRUE,
  display_diff_toggle = FALSE,
  filter = c("prop", "n"),
  filter_label = NULL,
  filter_range = NULL,
  ae_label = NULL,
  width = 1400,
```

```
  max_page = NULL,
  dowload_button = FALSE
)
```

Arguments

`outdata`	An `outdata` object created by `format_ae_forestly()`.
`display_soc_toggle`	A boolean value to display SOC toggle button.
`display_diff_toggle`	A boolean value to display risk difference toggle button.
`filter`	A character value of the filter variable. If NULL, the slider bar will not be displayed.
`filter_label`	A character value of the label for slider bar.
`filter_range`	A numeric vector of length 2 for the range of the slider bar.
`ae_label`	A character value of the label for criteria. If NULL (default), the range is automatically calculated from the data. If only one value is provided, it will be used as the maximum and minimum will be 0.
`width`	A numeric value of width of the table in pixels.
`max_page`	A numeric value of max page number shown in the table.
`dowload_button`	A logical value to display download button.

Value

An AE forest plot saved as a `shiny.tag.list` object.

Examples

```
adsl <- forestly_adsl[1:100, ]
adae <- forestly_adae[1:100, ]
if (interactive()) {
  meta_forestly(
    dataset_adsl = adsl,
    dataset_adae = adae,
  ) |>
    prepare_ae_forestly() |>
    format_ae_forestly() |>
    ae_forestly()
}
```

`background_panel` *Add background for creating plot with customized color*

Description

Creates colored background for panels of rainfall or forest plot.

Usage

```
background_panel(
  g,
  background_color = c("#69B8F7", "#FFFFFF"),
  background_alpha = 0.3
)
```

Arguments

`g`	A ggplot object for adding colored background.
`background_color`	A vector of colors that defines the color for the plot background. Default is c("#69B8F7", "#FFFFFF"), which are pastel blue and white. The colors will be recycled.
`background_alpha`	Opacity of a geom. Default is 0.3.

Value

Plot as a colored background to add panels for rainfall or forest plot.

Examples

```
library(ggplot2)

df <- data.frame(
  study = c("S1", "S2", "S3", "S4", "S5", "S6", "S7"),
  item = as.factor(1:7),
  effect = c(-.4, -.25, -.1, .1, .15, .2, .3),
  lower = c(-.43, -.29, -.17, -.02, .04, .17, .27),
  upper = c(-.37, -.21, -.03, .22, .24, .23, .33)
)

ggplot(data = df) |>
  background_panel(background_color = c("grey", "white"), background_alpha = 0.4) +
  geom_point(aes(y = item, x = effect)) +
  geom_errorbar(aes(y = item, x = effect, xmin = lower, xmax = upper), width = 0.4) +
  scale_y_discrete(name = "", breaks = 1:nrow(df), labels = df$study)
```

`forestly_adae` *An adverse event dataset*

Description

A dataset containing the adverse event information of a clinical trial following the CDISC ADaM standard.

Usage

```
forestly_adae
```

Format

A data frame with 736 rows and 56 variables.

Details

Definition of each variable can be found at https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc.

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

---

`forestly_adae_3grp` *An adverse event dataset*

Description

A dataset containing the adverse event information of a clinical trial following the CDISC ADaM standard.

Usage

```
forestly_adae_3grp
```

Format

A data frame with 1191 rows and 56 variables.

Details

Definition of each variable can be found at https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc.

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

---

`forestly_adsl` *A subject level demographic dataset*

Description

A dataset containing the demographic information of a clinical trial following the CDISC ADaM standard.

Usage

```
forestly_adsl
```

Format

A data frame with 170 rows and 49 variables.

Details

Definition of each variable can be found at https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc.

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc

---

`forestly_adsl_3grp` *A subject level demographic dataset*

Description

A dataset containing the demographic information of a clinical trial following the CDISC ADaM standard.

Usage

```
forestly_adsl_3grp
```

Format

A data frame with 254 rows and 49 variables.

Details

Definition of each variable can be found at `https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc`.

Source

`https://github.com/phuse-org/phuse-scripts/tree/master/data/adam/cdisc`

`format_ae_forestly` *Format outdata for interactive forest plot*

Description

Format outdata for interactive forest plot

Usage

```
format_ae_forestly(
  outdata,
  display = c("n", "prop", "fig_prop", "fig_diff"),
  digits = 1,
  width_term = 200,
  width_fig = 320,
  width_n = 40,
  width_prop = 60,
  width_diff = 80,
  footer_space = 90,
  prop_range = NULL,
  diff_range = NULL,
  color = NULL,
  ae_col_header = NULL,
  diff_label = "Treatment <- Favor -> Placebo",
  diff_col_header = NULL,
  diff_fig_header = NULL
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_forestly()`.
`display`	A character vector of measurement to be displayed. • `n`: Number of subjects with AE. • `prop`: Proportion of subjects with AE. • `total`: Total columns. • `diff`: Risk difference.
`digits`	A number of digits after decimal point to be displayed for proportion and risk difference.

`width_term`	Width in px for AE term column.
`width_fig`	Width in px for proportion and risk difference figure.
`width_n`	Width in px for "N" columns.
`width_prop`	Width in px for "(%)" columns.
`width_diff`	Width in px for risk difference columns.
`footer_space`	Space in px for footer to display legend.
`prop_range`	A vector of lower and upper limit of x-axis for proportion figure.
`diff_range`	A vector of lower and upper limit of x-axis for risk difference figure.
`color`	A vector of colors for analysis groups. Default value supports up to 4 groups.
`ae_col_header`	Column header for adverse events item columns. If NULL (default) and "par" specified in `components` from `prepare_ae_forestly()`, uses "Adverse Event". If NULL and "soc" specified in `components` from `prepare_ae_forestly()`, uses "System Organ Class" for "soc".
`diff_label`	x-axis label for risk difference.
`diff_col_header`	Column header for risk difference table columns. If NULL (default), uses "Risk Difference (%) vs. Reference Group".
`diff_fig_header`	Column header for risk difference figure. If NULL (default), uses "Risk Difference (%) + 95% CI vs. Reference Group".

Value

An `outdata` object.

Examples

```
adsl <- forestly_adsl[1:100, ]
adae <- forestly_adae[1:100, ]
meta_forestly(
  dataset_adsl = adsl,
  dataset_adae = adae
) |>
  prepare_ae_forestly() |>
  format_ae_forestly()
```

`meta_forestly` *Create metadata for interactive forest plot*

Description

Create metadata for interactive forest plot

Usage

```
meta_forestly(
  dataset_adsl,
  dataset_adae,
  population_term = "apat",
  observation_term = "safety",
  parameter_term = "any;rel",
  population_subset,
  observation_subset,
  treatment_group = "TRTA"
)
```

Arguments

`dataset_adsl`	ADSL source dataset.
`dataset_adae`	ADAE source dataset.
`population_term`	A character value of population term name.
`observation_term`	A character value of observation term name.
`parameter_term`	A character value of parameter term name.
`population_subset`	An unquoted condition for selecting the populations from ADSL dataset.
`observation_subset`	An unquoted condition for selecting the observations from ADAE dataset.
`treatment_group`	A character value of treatment group name.

Value

A metalite object.

Examples

```
meta_forestly(
  forestly_adsl,
  forestly_adae,
  population_term = "apat",
  observation_term = "safety",
  parameter_term = "any;rel"
)
```

`plot_dot` *Dot plot*

Description

Create a dot plot by item. For instance, this could be used to create AEs incidence plot by Preferred Term and treatment group, as part of a rainfall plot.

Usage

```
plot_dot(
  tbl,
  prop_cols = c("prop_1", "prop_2"),
  y_var,
  label,
  x_breaks = NULL,
  color = NULL,
  shape = NULL,
  title = "AE (%)",
  background_color = c("#69B8F7", "#FFFFFF"),
  background_alpha = 0.3,
  theme = theme_panel(show_text = TRUE, show_ticks = TRUE),
  legend_nrow = 1
)
```

Arguments

`tbl`	A data frame selected from input data set to display on this plot. y and x variables are required.
`prop_cols`	A character vector of proportion columns to be used for a plot.
`y_var`	A character string that specifies a variable to be displayed on the y-axis.
`label`	A character vector of labels for each treatment group. The control group label should be specified as the last element of the vector.
`x_breaks`	A numeric vector for x-axis breaks. Default is `NULL`, which uses a default ggplot2 x-axis breaks presentation.
`color`	Color for each treatment group.
`shape`	Shape for each treatment group. Default is circle and square. Input values could be either a character or numeric value, For details, see `https://ggplot2.tidyverse.org/articles/ggplot2-specs.html`.
`title`	Panel title. Default is `"AE (%)"`.
`background_color`	Plot background color. Default is `c("#69B8F7", "#FFFFFF")`, which are pastel blue and white. The value of this argument is used as input for the `background_color` argument in `background_panel()`.

`background_alpha`	Opacity of the background. Default is 0.3. The value of this argument is the input for background_alpha argument in `background_panel()`.
`theme`	Panel theme, including the y-axis text, ticks, and plot margin. Default is `theme_panel(show_text = TRUE, show_ticks = TRUE)`. For more details, refer to `theme_panel`.
`legend_nrow`	Integer, the number of rows for a legend display. Must be smaller than or equal to the number of the treatment groups. To omit the legend, set this to `NULL`. Default is 1.

Value

AEs incidence plot by item and treatment group.

Examples

```
forestly_adsl$TRTA <- factor(
  forestly_adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
forestly_adae$TRTA <- factor(
  forestly_adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

meta <- meta_forestly(
  dataset_adsl = forestly_adsl,
  dataset_adae = forestly_adae,
  population_term = "apat",
  observation_term = "wk12",
  parameter_term = "any;rel;ser"
) |>
  prepare_ae_forestly() |>
  format_ae_forestly()

meta_any <- meta$tbl[1:20, ] |> dplyr::filter(parameter == "any")
meta_any |>
  plot_dot("name", prop_cols = c("prop_1", "prop_2"), label = c("Treatment", "Placebo"))
```

`plot_errorbar` *Plot to display risk difference*

Description

Create a plot to display risk difference for each item.

Usage

```
plot_errorbar(
  tbl,
  ci_cols = c("diff_1", "lower_1", "upper_1"),
  y_var,
  errbar_width = 0.4,
  color = NULL,
  shape = NULL,
  label,
  x_breaks = NULL,
  grp_abbrev = "paired",
  favor_direction = "negative",
  vline = NULL,
  line_type = 1,
  title = "Risk Diff. + 95% CI \n (Percentage Points)",
  background_color = c("#69B8F7", "#FFFFFF"),
  background_alpha = 0.3,
  theme = theme_panel(show_text = TRUE, show_ticks = TRUE),
  legend_nrow = 1
)
```

Arguments

`tbl`	A data frame selected from input data set to display on this plot. y and x variables are required.
`ci_cols`	A character vector of columns for a risk difference to be used for a plot. Need 3 columns, risk difference, lower bound, and upper bound.
`y_var`	A character string that specifies a variable to be displayed on the y-axis.
`errbar_width`	A numeric value to define the error bar width. Default is 0.4. Value of this argument will be a half length of the error bar, for example, `errorbar_width = 0.2` means half of the error bar width is 0.2 unit length. If y = 4, the error bar will range from y = 3.8 to y = 4.2.
`color`	Color for each treatment group.
`shape`	Shape for each treatment group. Default is circle and square. Input values could be either a character or numeric value, For details, see `https://ggplot2.tidyverse.org/articles/ggplot2-specs.html`.
`label`	A character vector of labels for each treatment group. The control group label should be specified as the last element of the vector.
`x_breaks`	A numeric vector for x-axis breaks. Default is `NULL`, which uses a default ggplot2 x-axis breaks presentation.
`grp_abbrev`	A character vector for displaying the treatment groups on a favor bar. If `grp_abbrev = "paired"`, treatment label on the error bar will be the same as in the `label` argument. If `grp_abbrev = "none"`, the error bar will not be shown. Also, for customized terms, users can provide an alternative vector of treatment labels. Default is `"paired"`.

`favor_direction`	The position of a favor label under the condition "comparison is treatment – control". For AEs, `favor_direction` should be `"negative"`; for efficacy, `favor_direction` should be `"positive"`.
`vline`	Vertical reference line position. Default is `NULL`. Users can define one or multiple numeric values in a vector as a reference line position.
`line_type`	Reference line type. Default is solid line. Users can define one or multiple line types in a vector (can be numeric such as 1, 2, 3 or character such as `"solid"`, `"dashed"`). The values will be recycled and the order will be consistent with the argument `vline`.
`title`	Plot title. Default is `"Risk Diff. + 95% CI \\n (Percentage Points)"`.
`background_color`	Plot background color. Default is `c("#69B8F7", "#FFFFFF")`, which are pastel blue and white. The value of this argument is used as input for the `background_color` argument in `background_panel()`.
`background_alpha`	Opacity of the background. Default is 0.3. The value of this argument is the input for `background_alpha` argument in `background_panel()`.
`theme`	Panel theme, including the y-axis text, ticks, and plot margin. Default is `theme_panel(show_text = TRUE, show_ticks = TRUE)`. For more details, refer to `theme_panel`.
`legend_nrow`	Integer, the number of rows for a legend display. Must be smaller than or equal to the number of the treatment groups. To omit the legend, set this to `NULL`. Default is 1.

Value

A risk difference plot for each item.

Examples

```
forestly_adsl$TRTA <- factor(
  forestly_adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
forestly_adae$TRTA <- factor(
  forestly_adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

meta <- meta_forestly(
  dataset_adsl = forestly_adsl,
  dataset_adae = forestly_adae,
  population_term = "apat",
  observation_term = "wk12",
  parameter_term = "any;rel;ser"
) |>
  prepare_ae_forestly() |>
```

```
  format_ae_forestly()

meta_any <- meta$tbl[1:20, ] |> dplyr::filter(parameter == "any")
meta_any |>
  dplyr::select(name, diff_1, lower_1, upper_1) |>
  plot_errorbar(
    y_var = "name",
    ci_cols = c("diff_1", "lower_1", "upper_1"),
    label = c("Treatment", "Placebo")
  )
```

`prepare_ae_forestly` *Prepare datasets for interactive forest plot*

Description

Prepare datasets for interactive forest plot

Usage

```
prepare_ae_forestly(
  meta,
  population = NULL,
  observation = NULL,
  parameter = NULL,
  components = "par",
  reference_group = NULL,
  ae_listing_display = c("USUBJID", "SITEID", "SEX", "RACE", "AGE", "ASTDY", "AESER",
    "AEREL", "AEACN", "AEOUT", "ADURN", "ADURU"),
  ae_listing_unique = FALSE,
  bisection = 100,
  ...
)
```

Arguments

`meta`	A metadata object created by metalite.
`population`	A character value of population term name. The term name is used as key to link information.
`observation`	A character value of observation term name. The term name is used as key to link information.
`parameter`	A character value of parameter term name. The term name is used as key to link information.
`components`	A character vector of components name.
`reference_group`	An integer to indicate reference group. Default is 2 if there are 2 groups, otherwise, the default is 1.

`ae_listing_display`	A vector of name of variables used to display on AE listing table.
`ae_listing_unique`	A logical value to display only unique records on AE listing table.
`bisection`	A numeric value. A control parameter for the bisection method used to calculate confidence the lower and upper confidence interval bounds for the risk. The default value is `1e2`.
`...`	Additional arguments passed to `metalite.ae::rate_compare_sum()`.

Value

An `outdata` object.

Examples

```
adsl <- forestly_adsl[1:100, ]
adae <- forestly_adae[1:100, ]
meta_forestly(
  dataset_adsl = adsl,
  dataset_adae = adae
) |>
  prepare_ae_forestly()
```

`rtf_static_forestly` *Output static forest plot*

Description

Output static forest plot

Usage

```
rtf_static_forestly(
  outdata,
  plot_calls,
  source,
  parameter = "any",
  n_rows = 25,
  orientation = "portrait",
  fig_size = c(6, 6),
  title = c("analysis", "observation", "population"),
  footnotes = NULL,
  text_font_size = 9,
  path_outdata = tempfile(fileext = ".Rdata"),
  path_outtable = tempfile(fileext = ".rtf")
)
```

Arguments

`outdata`	An `outdata` object created by `format_ae_forestly()`.
`plot_calls`	A list or vector of function calls.
`source`	A character value of the data source.
`parameter`	A character value of parameter term name.
`n_rows`	An integer value of the number of rows per page in a plot.
`orientation`	Orientation in 'portrait' or 'landscape'.
`fig_size`	A numeric vector of length 2 of figure width and height. The length should be 2 (width, height). The unit is inch.
`title`	Term "analysis", "observation" and "population") for collecting title from metadata or a character vector of table titles.
`footnotes`	A character vector of table footnotes.
`text_font_size`	Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
`path_outdata`	A character string of the outdata path.
`path_outtable`	A character string of the outtable path.

Value

RTF file and source dataset for baseline characteristic table.

Examples

```
forestly_adsl$TRTA <- factor(
  forestly_adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
forestly_adae$TRTA <- factor(
  forestly_adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
outdata <- meta_forestly(
  dataset_adsl = forestly_adsl[1:40, ],
  dataset_adae = forestly_adae[1:40, ]
) |>
  prepare_ae_forestly() |>
  format_ae_forestly()

p1 <- substitute(
  plot_dot(
    tbl,
    prop_cols = c("prop_1", "prop_2"),
    y_var = "name",
    label = c("Treatment", "Placebo")
  )
```

```
)
p2 <- substitute(
  plot_errorbar(
    tbl,
    ci_cols = c("diff_1", "lower_1", "upper_1"),
    y_var = "name",
    label = c("Treatment", "Placebo"),
    legend_nrow = NULL,
    theme = theme_panel(show_ticks = FALSE, show_text = FALSE)
  )
)
p3 <- substitute(
  table_panel(
    tbl,
    n_cols = c("n_1", "n_2"),
    prop_cols = c("prop_1", "prop_2"),
    y_var = "name",
    theme = theme_panel(show_ticks = FALSE, show_text = FALSE),
    x_label = c("Treatment \n n(%)", "Placebo \n n(%)")
  )
)
outdata |> rtf_static_forestly(
  plot_calls = c(p1, p2, p3),
  source = "Source:  [CDISCpilot: adam-adsl; adae]",
  path_outdata = tempfile(fileext = ".Rdata"),
  path_outtable =  tempfile(fileext = ".rtf")
)
```

`table_panel` *Create table panel ggplot2 object for rainfall or forest plot*

Description

Creates a table panel ggplot2 object for rainfall or forest plot.

Usage

```
table_panel(
  tbl,
  n_cols = c("n_1", "n_2"),
  prop_cols = c("prop_1", "prop_2"),
  y_var,
  x_label = NULL,
  text_color = NULL,
  text_size = 8,
  text_format_by = "column",
  background_color = c("#69B8F7", "#FFFFFF"),
  theme = theme_panel(show_ticks = TRUE, show_text = TRUE),
  background_alpha = 0.3
)
```

Arguments

`tbl`	A data frame to be displayed in this table.
`n_cols`	A character vector of columns for subject count to be used for a plot.
`prop_cols`	A character vector of proportion columns to be used for a plot.
`y_var`	A string of a variable name from `tbl` for the y axis variable.
`x_label`	Labels displayed on the top of table for each column of table. Default is `NULL`, variable name will display as label.
`text_color`	Defines colors to display each treatment group.
`text_size`	Numeric font size for data on each column. Default is 8 for each column.
`text_format_by`	An option for formatting a data by columns or rows. Default is `"column"` and text color will be varied by column. If `text_format_by = "row"`, then text color will be varied by row. If `text_format_by = "group"`, then text color will be varied by treatment group.
`background_color`	Color for the plot background. Default is `c("#69B8F7", "#FFFFFF")` which are pastel blue and white. The value of this argument will be the input value for the `background_color` argument in `background_panel()`.
`theme`	Controls display of y axis text, ticks and plot margin. By default, `theme_panel(show_text = TRUE, show_ticks = TRUE)` is used. Users are suggested to use `theme_panel()`.
`background_alpha`	Opacity of the background. Default is 0.3. The value of this argument will be the input value for the `background_alpha` argument in `background_panel()`.

Value

A ggplot2 object for table panel.

Examples

```
forestly_adsl$TRTA <- factor(
  forestly_adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
forestly_adae$TRTA <- factor(
  forestly_adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

meta <- meta_forestly(
  dataset_adsl = forestly_adsl,
  dataset_adae = forestly_adae,
  population_term = "apat",
  observation_term = "wk12",
  parameter_term = "any;rel;ser"
) |>
```

```
    prepare_ae_forestly() |>
    format_ae_forestly()

  meta_any <- meta$tbl[1:20, ] |> dplyr::filter(parameter == "any")

  meta_any |>
    table_panel(y_var = "name")
```

`theme_panel` *Theme function for plot with multiple panels*

Description

Specifies theme for a plot with multiple panels.

Usage

```
theme_panel(show_text = TRUE, show_ticks = TRUE)
```

Arguments

`show_text`	A logical value that controls text display on the y axis. Default is `TRUE`.
`show_ticks`	A logical value that controls ticks display on the y axis. Default is `TRUE`.

Value

Theme for a specific panel.

Examples

```
library(ggplot2)

p <- ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()

p
p + theme_panel()
```

Index