Package 'BiMaUmisc'

November 2, 2026

Type Package

Title BiMaU Miscellaneous

Version 0.2.1

Maintainer Anna Felip-Badia <annafelipibadia@gmail.com>

Description Contains a function to plot publication-ready survival curves using the Kaplan-Meier method (1958) <doi:10.2307/2281868>, a function to format p-values, and a function to automatically select statistical tests for comparing quantitative variables between groups, which are useful for repetitive analyses. BiMaU stands for the Biostatistics and Mathematics Research Unit at the Sant Joan de Déu - Pediatric Cancer Center Barcelona <https://github.com/BiMaU-PCCB>.

License GPL (>= 2)

Depends R (>= 4.1.0)

Encoding UTF-8

URL https://github.com/BiMaU-PCCB/BiMaUmisc, https://bimau-pccb.github.io/BiMaUmisc/

Imports exams, survival (>= 3.8-6), latex2exp, plotfunctions, graphics, utils

Suggests gtsummary

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Anna Felip-Badia [aut, cre] (ORCID: <https://orcid.org/0009-0002-8751-5228>), Sara Perez-Jaume [aut] (ORCID: <https://orcid.org/0000-0002-9805-4810>)

Repository CRAN

Date/Publication 2026-11-02 15:40:15 UTC

Contents

find.test 2
show.p 3
survival.plot 4

Index 9

`find.test` *Automatically select statistical tests for quantitative variables*

Description

This function evaluates quantitative variables across stratification groups and automatically determines the appropriate parametric or non-parametric hypothesis test. See **Details** for an exhaustive explanation of the criteria followed. Designed to work seamlessly with `tbl_summary`, this function generates the input lists for arguments `test` and `test.args` of the function `add_p`, specifying the recommended test function and any additional arguments different from the defaults.

Usage

```
find.test(data, variables, strata, alpha = 0.05)
```

Arguments

`data`	name of the dataset, of class `data.frame`, containing at least two columns (quantitative variable, strata).
`variables`	column name(s) of the quantitative variable(s) to analyse. Can be specified as a character string (with quotes) or as an unquoted symbol.
`strata`	strata indicator column name. Can be specified as a character string (with quotes) or as an unquoted symbol. Must be a factor with at least 2 levels.
`alpha`	significance level threshold used for normality (`shapiro.test`) and homoscedasticity (`var.test` or `bartlett.test`) testing. The default value is 0.05.

Details

If any level of `strata` has less than five non-missing observations, no statistical test is performed for that variable, and a warning is issued. Otherwise, the appropriate test is selected based on the following criteria:

Two Groups (`strata` with 2 levels):

- ***t*-test** (`t.test`): selected if sample sizes in both groups are $\geq$ 30 and Shapiro-Wilk tests indicate normality in both groups. Variance equality is tested using an F-test.
- **Mann-Whitney test** (`wilcox.test`): selected if sample sizes are small, this is $<$ 30 in some group, or if normality is rejected in either group.

More than Two Groups (`strata` with > 2 levels):

- **ANOVA** (`oneway.test`): selected if ANOVA model residuals adhere to a normal distribution according to the Shapiro-Wilk test. Homoscedasticity is tested via Bartlett's test.
- **Kruskal-Wallis test** (`kruskal.test`): selected if ANOVA residuals deviate from normality.

Value

A `list` of length 2 intended to be passed directly to arguments `test` and `test.args` of the function `add_p` when using `tbl_summary`:

`test` a list of formulas matching each variable name to its selected test string (e.g., `age ~ "t.test"`).

`test.args` a list of formulas specifying additional test parameters (e.g., `age ~ list(var.equal = TRUE)`). Variables requiring no additional parameters are omitted from this second list.

Examples

```
df <- as.data.frame(gtsummary::trial)
df$trt <- factor(df$trt)
df$grade <- factor(df$grade)

# Two levels variable (trt: Drug A, Drug B)
tests2 <- find.test(df, variables = c(age, marker), strata = trt)
tests2

# Use outputs directly with tbl_summary()
gtsummary::tbl_summary(df, include = c(age, marker), by = trt) |>
  gtsummary::add_p(test = tests2$test, test.args = tests2$test.args)

# Three levels variable (grade: I, II, III)
tests3 <- find.test(df, variables = c(age, marker), strata = grade)
tests3

# Use outputs directly with tbl_summary()
gtsummary::tbl_summary(df, include = c(age, marker), by = grade) |>
  gtsummary::add_p(test = tests3$test, test.args = tests3$test.args)

# Using function show.p() from BiMaUmisc package
gtsummary::tbl_summary(df, include = c(age, marker), by = grade) |>
  gtsummary::add_p(test = tests3$test, test.args = tests3$test.args,
                   pvalue_fun = function(x) show.p(x))
```

---

show.p *Format* p*-values for display*

Description

Convert *p*-values to a compact textual representation. The function prints decimals up to the first non-zero decimal and the following one, applying standard rounding.

Usage

```
show.p(p, add.p = FALSE)
```

Arguments

p	a value or vector of the *p*-value(s) to be shown, where $0 < p < 1$.
add.p	logical indicating whether the text *p*= (or *p*<, if it applies) should be added before the *p*-value. The default value is FALSE.

Details

The number of decimals shown depends on the magnitude of the *p*-value so that two meaningful digits are displayed after the leading zeros.

Specifically:

- For $p \geq 0.1$, two decimal places are shown.
- For $0.01 \leq p < 0.1$, three decimal places are shown.
- For $0.001 \leq p < 0.01$, four decimal places are shown.
- For $0.0001 \leq p < 0.001$, five decimal places are shown.
- For $p < 0.0001$, the string "<0.0001" is returned.

This formatting ensures that the first non-zero decimal of the p-value and the following digit are displayed, while avoiding an excessive number of leading zeros.

If add.p = TRUE, the prefix "*p*=" (or "*p*<" when applicable) is added to the formatted value. This will ensure the *p* appears in cursive when calling the function in line on R markdown documents.

Value

A character vector with the formatted *p*-values.

Examples

```
show.p(0.00785)
show.p(c(0.03042, 0.1579, 0.0000025))
show.p(0.00785, add.p = TRUE)
show.p(c(0.03042, 0.1579, 0.0000025), add.p = TRUE)
```

---

survival.plot *Plot survival curves*

Description

This function produces an elegant survival curve plot. It can automatically append the number-at-risk table and display the log-tank test *p*-value (obtained from a call to survdiff). Highly customizable, it offers numerous arguments to easily modify the plot's appearance.

Usage

```
survival.plot(data, time, status, strata = 1,
              legend = TRUE, legend.pos = "bottomleft", legend.labs = NULL,
              legend.title = NULL, legend.cex = 1,
              events = FALSE, events.pos = NULL, events.text = NULL,
              mark.time = TRUE, lwd = 3,
              risk = TRUE, risk.text = TRUE, risk.labs = NULL, risk.cex = 1,
              p.value = TRUE, p.value.pos = NULL,
              units = "days", xlab = NULL, times = NULL,
              yscale = 1, ylab = NULL,
              main = "Survival curve", col = NULL, ...)
```

Arguments

`data`	name of the dataset, of class `data.frame`, containing at least 2 columns (follow up time, status indicator).
`time`	follow-up time column name. Can be specified as a character string (with quotes) or as an unquoted symbol.
`status`	status indicator column name. Can be specified as a character string (with quotes) or as an unquoted symbol.
`strata`	strata indicator column name. Can be specified as a character string (with quotes) or as an unquoted symbol. Up to four categories. Must be a factor. The default value is 1, this is, there are no groups.
`legend`	logical indicating whether a legend should be added to the plot. If `strata` is set to 1, this argument is ignored. The default value is `TRUE`.
`legend.pos`	this argument can be a single keyword from the list in `legend`, such as `"bottomleft"` or `"topright"`, which places the legend on the inside of the plot frame at the given location. In this case, this argument is passed to `match.arg`. Otherwise, a 2-dimensional numerical vector indicating the x-y coordinates can be indicated. If `legend` is set to `FALSE`, this argument is ignored. The default value is `"bottomleft"`.
`legend.labs`	a character vector of length ≥ 2 to appear as the legend labels. If `legend` is set to `FALSE`, this argument is ignored. The default value, `NULL`, indicates that the levels of `strata` are to be used as the legend labels.
`legend.title`	title for the legend. If `legend` is set to `FALSE`, this argument is ignored. The default value, `NULL`, indicates on title should be added.
`legend.cex`	legend's character expansion factor. A numerical value giving the amount by which the legend text should be magnified relative to the default. If `legend` is set to `FALSE`, this argument is ignored. The default value is 1.
`events`	logical indicating whether the number of events should be added to the text in the legend. If `strata` is set to 1, the total number of events is added directly to the plot instead. If `legend` is set to `FALSE` and `strata` is not 1, this argument is

ignored.
The default value is FALSE.

`events.pos` 2-dimensional numerical vector indicating the x-y coordinates where the number of events should be added to the plot. If `strata` is not 1 or `events` is set to FALSE, this argument is ignored.
The default value, NULL, indicates that the text should be placed on the bottom-left corner.

`events.text` a character indicating the text to display before the number of events in the plot. If `events` is set to FALSE, this argument is ignored.
The default value, NULL, indicates that the text "Events" should be added when `strata` is set to 1 and the text "events" should be added when `strata` is not set to 1.

`mark.time` logical indicating whether the curves should be marked at each censoring time.
The default value is TRUE.

`lwd` a number indicating the line width(s). Alternatively, when `strata` is not set to 1, an n-dimensional numerical vector can be indicated, where n is the number of levels in `strata`.
The default value is 3.

`risk` logical indicating whether the number-at-risk table should be appended to the plot.
The default value is TRUE.

`risk.text` logical indicating whether the legend for the number-at-risk table should display text labels. When FALSE, line segments are used instead. If `risk` is set to FALSE or `strata` is set to 1, this argument is ignored.
The default value is TRUE.

`risk.labs` n-dimensional character vector, where n is the number of levels in `strata`, indicating the text labels in the number-at-risk table. If `risk.text` is set to FALSE or `strata` is set to 1, this argument is ignored.
The default value, NULL, indicates that the levels of `strata` should be used.

`risk.cex` number-at-risk table's character expansion factor. A numerical value giving the amount by which the number-at-risk table text should be magnified relative to the default. If `risk` is set to FALSE, this argument is ignored.
The default value is 1.

`p.value` logical indicating whether the p-value obtained from a call to `survdiff` with `rho` equal to 0, this is the log-rank or Mantel-Haenszel test, should be displayed in the plot. The p-value will be displayed in the format specified by the `show.p` function. If `strata` is set to 1, this argument is ignored.
The default value is TRUE.

`p.value.pos` 2-dimensional numerical vector indicating the x-y coordinates where the number of events should be added to the plot. If `strata` is not 1 or `events` is set to FALSE, this argument is ignored.
The default value, NULL, indicates that the text should be placed on the upper-right corner.

`units` character indicating the time units. Options are: "days", "weeks", "months" or "years". This argument will be used to create the default label and tick-marks

for the x-axis. This argument is not case-sensitive and is passed to match.arg. The default value is "days".

xlab — character indicating a label for the x-axis.
The default value, NULL, indicates that indicates that the text "Time (units)" should be used, where "units" is specified by the units argument.

times — numerical vector with the time points at which tick-marks are to be drawn in the x-axis.
The default value, NULL, indicates that a time vector will be automatically constructed based on the units argument:

- If units is set to "days": from 0 to the maximum survival time rounded up to the nearest multiple of 7.
- If units is set to "weeks": from 0 to the maximum survival time rounded up to the nearest integer.
- If units is set to "months": from 0 to the maximum survival time rounded up to the nearest multiple of 12.
- If units is set to "years": from 0 to the maximum survival time rounded up to the nearest integer.

yscale — a numeric value used to multiply the labels on the y-axis. A value of 100, for instance, would be used to give a percent scale. Notice only the labels are changed, not the actual plot coordinates.
The default value is 1.

ylab — character indicating a label for the y-axis.
The default value, NULL, indicates that indicates that the text "Survival" should be used. Moreover, if yscale is set to 1 (or 100) the text "(proportion)" or "(percentage)" will also be added.

main — overall title for the plot.
The default value is "Survival curve".

col — colour palette to be used.
The default value, NULL, indicates that a different colour should be used per level in strata. This is, a vector from 1 to n, where n is the number of levels in strata, will be used.

... — other graphical parameters (to be passed to plot.survfit).

Value

A customized survival curve plot displaying the requested graphical and statistical modifications.

Note

A call to par is used in this function. Notice that the arguments font.axis, font.lab, cex.lab, las and xpd are always set to 2, 2, 1.2, 1, and TRUE, respectively. Moreover, the argument mar is always modified and depends on the number of levels in strata as well as whether or not the number-at-risk table is appended.
For optimal resolution and layout alignment when using the function in an R file, adjust the plot window. For example, ideal dimensions are approximately 7 x 5" for 24" screens and 5.9 x 4.2" for 13" screens.

For optimal resolution and layout alignment when using the function in an R markdown file or similar, use `fig.dim = c(7, 5)` in the corresponding chunk.

Examples

```
# CAUTION: Before running through the Console, read Note to ensure proper
# visualization (how to modify dimensions of the Plot window).
lung <- survival::cancer
lung$time_y <- lung$time/365.25
lung$sex <- factor(lung$sex, levels = 1:2, labels = c("Male", "Female"))

# no strata
survival.plot(lung, time_y, status, units = "y")
# customizing
survival.plot(lung, time_y, status, units = "y",
              xlab = "Time since diagnosis (in years)",
              yscale = 100,
              main = "Overall survival", mark.col = "darkgray")

# 2 strata
survival.plot(lung, time_y, status, sex, units = "y")
# customizing
survival.plot(lung, time_y, status, sex,
              legend.pos = "topright", legend.title = "Sex",
              events = TRUE, risk.text = FALSE, p.value.pos = c(2.4, 0.4),
              units = "y", xlab = "Time since diagnosis (in years)",
              main = "Overall survival", col = c("darkviolet", "darkgreen"))
```

Index

add_p, *2, 3*

bartlett.test, *2*

find.test, 2

kruskal.test, *2*

legend, *5*

match.arg, *5*, *7*

oneway.test, *2*

par, *7*
plot.survfit, *7*

shapiro.test, *2*
show.p, 3, *6*
survdiff, *4*, *6*
survival.plot, 4

t.test, *2*
tbl_summary, *2, 3*

var.test, *2*

wilcox.test, *2*