Package ‘easysurv’

June 21, 2024

Type Package

Title Simplify Survival Data Analysis and Model Fitting

Version 2.0.1

Description Inspect survival data, plot Kaplan-Meier curves, assess the proportional hazards assumption, fit parametric survival models, predict and plot survival and hazards, and export the outputs to Excel. A simple interface for fitting survival models using 'flexsurv::flexsurvreg()', 'flexsurv::flexsurvspline()', 'flexsurvcure::flexsurvcure()', and 'survival::survreg()'.

License MIT + file LICENSE

URL `https://github.com/Maple-Health-Group/easysurv`, `https://maple-health-group.github.io/easysurv/`

BugReports `https://github.com/Maple-Health-Group/easysurv/issues`

Depends R (>= 4.1.0)

Imports bshazard, censored, cli, data.table, dplyr, flexsurv, flexsurvcure, ggplot2, ggsurvfit, openxlsx, parsnip, plotly, purrr, rlang, scales, stats, survival, tibble, tidyr, usethis, utils

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/Needs/website r-lib/asciicast

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Niall Davison [aut, cre] (<`https://orcid.org/0000-0002-9447-0825`>),
Brad Kievit [aut],
Maple Health Group, LLC [cph, fnd]

Maintainer Niall Davison <`niall.davison@maplehealthgroup.com`>

Repository CRAN

Date/Publication 2024-06-21 10:30:06 UTC

Contents

`easy_adtte` *Formatted Copy of ggsurvfit::adtte*

Description

Background The example simulated data set is based on large phase III clinical trials in breast cancer such as the ALTTO trial doi:10.1200/JCO.2015.62.1797. The example trial aims to determine if a combination of two therapies tablemab (T) plus vismab (V) improves outcomes for metastatic human epidermal growth factor 2-positive breast cancer and increases the pathologic complete response in the neoadjuvant setting (i.e. treatment given as a first step to shrink a tumor before the main treatment or surgery).

The trial has four treatment arms, patients with centrally confirmed human epidermal growth factor 2-positive early breast cancer were randomly assigned to 1 year of adjuvant therapy with V, T, their sequence (T to V), or their combination (T+V) for 52 weeks.

The primary end point was progression-free survival (PFS) as defined by Cancer.gov: ’"the length of time during and after the treatment of a disease, such as cancer, that a patient lives with the disease but it does not get worse. In a clinical trial, measuring the progression-free survival is one way to see how well a new treatment works"’.

A number of baseline measurements (taken at randomization) are also included such as age, hormone receptor status and prior radiotherapy treatment.

Additional details on reasons for study discontinuation and censoring event description are also included.

The data set adopts an abridged version of the CDISC ADaM ADTTE time to event data model. See here for more info on CDISC ADaM data standards `https://www.cdisc.org/standards/foundational/adam` and specifically the ADTTE time to event data model here `https://www.cdisc.org/standards/foundational/adam/adam-basic-data-structure-bds-time-event-tte-analyses-v1-0`.

Usage

`easy_adtte`

Format

The data set contains the following variables:

STUDYID The study identifier. A code unique to the clinical trial

SUBJID subject identifier. Numeric ID unique to each patient

USUBJID unique subject identifier. Text ID combining study and patient IDs

AGE age at randomisation (years)

STR01 Hormone receptor status at randomisation

STR01N Hormone receptor positive (Numeric)

STR01L Hormone receptor positive (Long format)

STR02 Prior Radiotherapy at randomisation

STR02N Prior Radiotherapy at randomisation (Numeric)

STR02L Prior Radiotherapy at randomisation (Long format)

TRT01P Planned treatment assigned at randomisation

TRT01PN Planned treatment assigned at randomisation (Numeric)

PARAM Analysis parameter: Progression free survival

PARAMCD Analysis parameter code

AVAL Analysis value (time to event (years)

CNSR Censoring (0 = Event, 1 = Censored)

EVNTDESC Event description

CNSDTDSC Censoring description

DCTREAS Discontinuation from study reason

Source

ggsurvfit::adtte

`easy_bc` *Formatted Copy of flexsurv::bc*

Description

This is a copy of the bc data set exported by the flexsurv package. This data set, however, has column labels assigned.

Usage

```
easy_bc
```

Format

The data set contains the following variables:

censrec 0 = Censored, 1 = Dead

rectime Time of censoring or death in days

group Prognostic group: Good, Medium, or Poor

recyrs Time of censoring or death in years

Source

flexsurv::bc

`easy_lung` *Formatted Copy of survival::lung*

Description

This is a copy of the lung data set exported by the survival package. This data set, however, has column labels assigned and time in months.

Usage

```
easy_lung
```

Format

The data set contains the following variables:

inst Institution code

time Survival time, months

status Censoring status, 1 = censored, 2 = dead

age Age

sex Sex, 1 = Male, 2 = Female

ph.ecog ECOG Performance Status (Physician)

ph.karno Karnofsky performance score (Physician)

pat.karno Karnofsky performance score (Patient)

meal.cal Calories consumed

wt.loss Weight loss, lbs

Source

survival::lung

`fit_models` *Fit Survival Models*

Description

Fits survival models to the provided data using the specified engine and returns various outputs including model parameters, goodness of fit, and estimates of median survival.

Usage

```
fit_models(
  data,
  time,
  event,
  predict_by = NULL,
  covariates = NULL,
 dists = c("exp", "gamma", "gengamma", "gompertz", "llogis", "lnorm", "weibull"),
  engine = "flexsurv",
  k = c(1, 2, 3),
  scale = "hazard",
  add_time_0 = TRUE,
  ...
)
```

Arguments

`data`	A data frame containing the survival data.
`time`	The name of the column in `data` containing the time-to-event information.
`event`	The name of the column in `data` indicating whether the event of interest occurred.
`predict_by`	(Optional) The name of the column in `data` defining the prediction variable.
`covariates`	(Optional) A character vector specifying the names of covariates to be included in the model.
`dists`	(Optional) A character vector specifying the distribution(s) to be fitted. When the engine parameter is set to "flexsurv", options are "exp", "exponential", "gamma", "genf", "genf.orig", "gengamma", "gengamma.orig", "gompertz", "llogis", "lnorm", "lognormal", "weibull", "weibullPH". When the engine parameter is set to "flexsurvcure", options are "exp", "gamma", "gengamma", "gompertz", "llogis", "lnorm", "weibull". When the engine parameter is set to "flexsurvspline", dists are ignored in favor of k and scale parameters. When the engine parameter is set to "survival", options are "exponential", "extreme", "gaussian", "loggaussian" (same as lognormal), "logistic", "lognormal", "rayleigh", "weibull". Default is `c("exp", "gamma", "gengamma", "gompertz", "llogis", "lnorm", "weibull")` which applies to flexsurv-related engines.
`engine`	(Optional) The survival analysis engine to be used. Options are "flexsurv", "flexsurvcure", "flexsurvspline", and "survival". Default is "flexsurv". • "flexsurv" uses `flexsurv::flexsurvreg()`. • "flexsurvspline" uses `flexsurv::flexsurvspline()`. • "flexsurvcure" uses `flexsurvcure::flexsurvcure()`. • "survival" uses `survival::survreg()`.
`k`	(Optional) A numeric vector specifying the number of knots for spline-based models. Default is `c(1, 2, 3)` to test different numbers.
`scale`	(Optional) A character vector specifying the scale parameter(s) for spline-based models. Options are "hazard", "odds", and "normal". Default is `"hazard"`.
`add_time_0`	Optional. Uses `survival::survfit0()` to add a starting time of 0 to the KM survfit object. This may be useful for plotting the KM at a subsequent stage (in surv_plots). Default is TRUE.
`...`	Additional arguments just to catch them and avoid errors.

Value

A list containing information about the fit_models() call, the distributions attempted, goodness of fit, fit averages, and cure fractions (if applicable).

Examples

```
models <- fit_models(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  predict_by = "group",
  covariates = "group"
)

models
```

`get_km` *Generate Kaplan-Meier estimates*

Description

Calculates Kaplan-Meier estimates for survival data and returns summary statistics, plots, and additional outputs.

Usage

```
get_km(
  data,
  time,
  event,
  group = NULL,
  group_labels = NULL,
  just_km = FALSE,
  ...
)
```

Arguments

`data`	A data frame containing the survival data.
`time`	The name of the column in `data` containing the time-to-event information.
`event`	The name of the column in `data` indicating whether the event of interest occurred.
`group`	(Optional) The name of the column in `data` defining the grouping variable. Default is `NULL`.
`group_labels`	Optional character vector containing the names of the strata (default is NULL). Provide in a consistent order with `levels(as.factor(data$group))`.
`just_km`	Logical. If `TRUE`, only the Kaplan-Meier estimates are returned. Default is `FALSE`.
`...`	(Optional) Parameters to pass to ggsurvfit.

Value

A list containing Kaplan-Meier estimates, summary statistics, and plots.

Examples

```
km_results <- get_km(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  group = "group",
  risktable_symbols = FALSE
)

km_results
```

`get_schoenfeld` *Extract Schoenfeld Residuals*

Description

This function extracts Schoenfeld residuals from a fitted `cox.zph` object and formats them into a tidy data frame.

Usage

```
get_schoenfeld(fit_zph)
```

Arguments

`fit_zph`	An object of class `cox.zph` produced by the `cox.zph` function, representing the Schoenfeld residuals of a Cox proportional hazards model.

Value

A tibble with the Schoenfeld residuals in long format, containing the columns:

`time`	The time variable from the Cox model.
`transform`	The transformation applied to the time variable.
`variable`	The variable names from the Cox model for which residuals are calculated.
`residual`	The Schoenfeld residuals for each variable at each time point.

Examples

```
library(survival)
test_fit <- survival::coxph(survival::Surv(time, status) ~ sex, data = lung)
test_fit_zph <- survival::cox.zph(test_fit)
get_schoenfeld(test_fit_zph)
```

`inspect_surv_data` *Inspect Survival Data*

Description

Quickly inspect the survival data to ensure it is in the correct format.

Usage

```
inspect_surv_data(data, time, event, group = NULL)
```

Arguments

`data`	A data frame containing the survival data.
`time`	The column name in `data` containing the time-to-event data.
`event`	The column name in `data` containing the event indicator data.
`group`	Optional. The column name in `data` containing the group indicator data.

Value

A list containing tibbles that summarise the first few rows of the survival data, the sample sizes, the events, and median survival.

Examples

```
inspect_surv_data(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  group = "group"
)
```

`plot.fit_models` *Plot method for* `fit_models`

Description

Plot method for `fit_models`

Usage

```
## S3 method for class 'fit_models'
plot(
  x,
  eval_time = NULL,
  km_include = TRUE,
  subtitle_include = TRUE,
  add_plotly = FALSE,
  ...
)
```

Arguments

`x`	An object of class `fit_models`
`eval_time`	Time points at which to evaluate the survival function. Default is `NULL`.
`km_include`	Logical value indicating whether to include Kaplan-Meier survival data. Default is `TRUE`.
`subtitle_include`	Logical value indicating whether to include a subtitle in the plot. Default is `TRUE`.
`add_plotly`	Logical value indicating whether to add plotly interactivity. Default is `FALSE`.
`...`	Additional arguments

Value

A list containing predictions and plots for the survival and hazards of models in a `fit_models` object.

Examples

```
models <- fit_models(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  predict_by = "group",
  covariates = "group"
)

plot(models)
```

`plot_cloglog` *Cumulative Log Log Plot*

Description

Generates a Cumulative Log Log survival curve plot using `ggsurvfit::ggsurvfit()` with customizable options.

Usage

```
plot_cloglog(
  fit,
  median_line = FALSE,
  legend_position = "top",
  plot_theme = theme_easysurv()
)
```

Arguments

`fit`	A survival::survfit object representing the survival data.
`median_line`	Logical value indicating whether to include a line representing the median survival time. Default is `FALSE`.
`legend_position`	Position of the legend in the plot. Default is "top".
`plot_theme`	ggplot2 theme for the plot. Default is `theme_easysurv()`.

Value

A ggplot object representing the cumulative log log plot.

Examples

```
library(ggsurvfit)
fit <- survfit2(Surv(time, status) ~ surg, data = df_colon)
plot_cloglog(fit)
```

`plot_km` *Plot Kaplan-Meier Data*

Description

Generates a Kaplan-Meier survival curve plot using `ggsurvfit::ggsurvfit()` with customizable options. This function provides sensible defaults while allowing for customization.

Usage

```
plot_km(
  fit,
  risktable = TRUE,
  risktable_symbols = TRUE,
  median_line = TRUE,
  legend_position = "top",
  plot_theme = theme_easysurv(),
  risktable_theme = theme_risktable_easysurv()
)
```

Arguments

`fit`	A survival::survfit object representing the survival data.
`risktable`	Logical value indicating whether to include a risk table below the plot. Default is `TRUE`.
`risktable_symbols`	Logical value indicating whether to include symbols instead of text to label risk table strata. Default is `TRUE`.
`median_line`	Logical value indicating whether to include a line representing the median survival time. Default is `TRUE`.
`legend_position`	Position of the legend in the plot. Default is "top".
`plot_theme`	ggplot2 theme for the plot. Default is `theme_easysurv()`.
`risktable_theme`	ggplot2 theme for the risk table. Default is `theme_risktable_easysurv()`.

Value

A ggplot object representing the Kaplan-Meier survival curve plot.

Examples

```
library(ggsurvfit)
fit <- survfit2(Surv(time, status) ~ surg, data = df_colon)
plot_km(fit, risktable_symbols = FALSE)
```

`plot_schoenfeld` *Plot Schoenfeld Residuals*

Description

Plot the residuals generated by the `get_schoenfeld` function. This function creates a visual representation of Schoenfeld residuals from a Cox proportional hazards model. It allows for customization of the plot, including the addition of horizontal and smoothed lines, and styling of points and plot elements.

Usage

```
plot_schoenfeld(
  residuals,
  hline = TRUE,
  sline = TRUE,
  sline_se = TRUE,
  hline_col = "#F8766D",
  hline_size = 1,
  hline_alpha = 1,
  hline_yintercept = 0,
```

```
  hline_lty = "dashed",
  sline_col = "#00BFC4",
  sline_size = 1,
  sline_alpha = 0.2,
  sline_lty = "dashed",
  point_col = "black",
  point_size = 1,
  point_shape = 19,
  point_alpha = 1,
  plot_theme = ggplot2::theme_bw()
)
```

Arguments

`residuals`	A data frame containing the Schoenfeld residuals, typically with columns `time`, `residual`, `transform`, and `variable`.
`hline`	Logical. If `TRUE`, a horizontal line is added to the plot. Default is `TRUE`.
`sline`	Logical. If `TRUE`, a smooth line is added to the plot. Default is `TRUE`.
`sline_se`	Logical. If `TRUE`, confidence intervals are displayed around the smooth line. Default is `TRUE`.
`hline_col`	Color of the horizontal line. Default is `"#F8766D"` (red).
`hline_size`	Line width of the horizontal line. Default is `1`.
`hline_alpha`	Transparency of the horizontal line. Default is `1`.
`hline_yintercept`	Y-intercept for the horizontal line. Default is `0`.
`hline_lty`	Line type for the horizontal line. Default is `"dashed"`.
`sline_col`	Color of the smooth line. Default is `"#00BFC4"` (blue).
`sline_size`	Line width of the smooth line. Default is `1`.
`sline_alpha`	Transparency of the smooth line. Default is `0.2`.
`sline_lty`	Line type for the smooth line. Default is `"dashed"`.
`point_col`	Color of the points representing residuals. Default is `"black"`.
`point_size`	Size of the points representing residuals. Default is `1`.
`point_shape`	Shape of the points representing residuals. Default is `19`.
`point_alpha`	Transparency of the points representing residuals. Default is `1`.
`plot_theme`	A ggplot2 theme for the plot. Default is `ggplot2::theme_bw()`.

Value

A ggplot object representing the plot of Schoenfeld residuals.

Examples

```
library(survival)
test_fit <- survival::coxph(survival::Surv(time, status) ~ sex, data = lung)
test_fit_zph <- survival::cox.zph(test_fit)
plot_schoenfeld(get_schoenfeld(test_fit_zph))
```

`predict.fit_models` *Predict method for* `fit_models`

Description

Predict method for `fit_models`

Usage

```
## S3 method for class 'fit_models'
predict(object, eval_time = NULL, type = c("survival", "hazard"), ...)
```

Arguments

`object`	An object of class `fit_models`
`eval_time`	(Optional) A vector of evaluation time points for generating predictions. Default is `NULL`, which if left as NULL, generates a sequence from 0 to 5 times the maximum observed time.
`type`	A character vector indicating the type of predictions to generate. Default is `c("survival", "hazard")`.
`...`	Additional arguments

Value

A list of predictions for each model in the `fit_models` object.

Examples

```
models <- fit_models(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  predict_by = "group",
  covariates = "group"
)

predict(models)
```

`predict_and_plot` *Predict and Plot Fitted Models*

Description

This function generates survival and hazard predictions and plots for each model in a `fit_models` object. Optionally, interactive `plotly` outputs can be added for each plot.

Usage

```
predict_and_plot(
  fit_models,
  eval_time = NULL,
  km_include = TRUE,
  subtitle_include = TRUE,
  add_plotly = FALSE
)
```

Arguments

`fit_models`	An object returned from fit_models.
`eval_time`	(Optional) A vector of evaluation time points for generating predictions. Default is `NULL`, which if left as NULL, generates a sequence from 0 to 5 times the maximum observed time.
`km_include`	A logical indicating whether to include Kaplan-Meier estimates in the plot outputs. Default is `TRUE`.
`subtitle_include`	A logical indicating whether to include the subtitle. Default is `TRUE`. The subtitle is the name of the group.
`add_plotly`	A logical indicating whether to add interactive plotly outputs for each plot. Default is `FALSE`.

Value

A list of predictions and plots for each model in the `fit_models` object.

Examples

```
models <- fit_models(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  predict_by = "group"
)

predict_and_plot(models)
```

`print.fit_models` *Print methods for* `fit_models()`

Description

Print methods for `fit_models()`

Usage

```
## S3 method for class 'fit_models'
print(x, ...)
```

Arguments

`x`	An object of class `fit_models`
`...`	Additional arguments

Value

A print summary of the `fit_models` object.

Examples

```
models <- fit_models(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  predict_by = "group",
  covariates = "group"
)

models
```

`print.get_km` *Print methods for* `get_km()`

Description

Print methods for `get_km()`

Usage

```
## S3 method for class 'get_km'
print(x, ...)
```

Arguments

`x`	An object of class `get_km`
`...`	Additional arguments

Value

The summary of the Kaplan-Meier estimates, printed via the console.

Examples

```
km_results <- get_km(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  group = "group",
  risktable_symbols = FALSE
)

print(km_results)
```

`print.inspect_surv_data` *Print methods for* `inspect_surv_data()`

Description

Print methods for `inspect_surv_data()`

Usage

```
## S3 method for class 'inspect_surv_data'
print(x, ...)
```

Arguments

`x`	An object of class `inspect_surv_data`
`...`	Additional arguments

Value

A print summary of the `inspect_surv_data` object.

`print.predict_and_plot` *Print methods for* `predict_and_plot()`

Description

Print methods for `predict_and_plot()`

Usage

```
## S3 method for class 'predict_and_plot'
print(x, ...)
```

Arguments

`x`	An object of class `predict_and_plot`
`...`	Additional arguments

Value

A print summary of the `predict_and_plot` object.

Examples

```
models <- fit_models(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  predict_by = "group"
)

predict_and_plot(models)
```

`print.test_ph` *Print methods for* `test_ph()`

Description

Print methods for `test_ph()`

Usage

```
## S3 method for class 'test_ph'
print(x, ...)
```

Arguments

`x`	An object of class `test_ph`
`...`	Additional arguments

Value

A print summary of the `test_ph` object.

Examples

```
ph_results <- test_ph(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  group = "group"
)

ph_results
```

`quick_start` *Launch Example Survival Analysis Script using the easy_lung Data Set*

Description

This function launches an example script for starting survival analysis using the easysurv package. The script uses a modified version of the lung data set exported from the survival package. The code is inspired by `usethis::use_template()` but modified to work outside the context of an .RProj or package.

Usage

```
quick_start(output_file_name = NULL)
```

Arguments

`output_file_name`	Optional. A file name to use for the script. Defaults to "easysurv_start.R" within a helper function.

Value

A new R script file with example code.

Examples

```
quick_start()
```

`quick_start2` *Launch Example Survival Analysis Script using the easy_bc Data Set*

Description

This function launches an example script for starting survival analysis using the easysurv package. The script uses a modified version of the bc data set exported from the flexsurv package. The code is inspired by `usethis::use_template()` but modified to work outside the context of an .RProj or package.

Usage

```
quick_start2(output_file_name = NULL)
```

Arguments

`output_file_name`	Optional. A file name to use for the script. Defaults to "easysurv_start.R" within a helper function.

Value

A new R script file with example code.

Examples

```
quick_start2()
```

`quick_start3` *Launch Example Survival Analysis Script using the easy_adtte Data Set*

Description

This function launches an example script for starting survival analysis using the easysurv package. The script uses simulated phase III breast cancer trial data available from the ggsurvfit package. The code is inspired by `usethis::use_template()` but modified to work outside the context of an .RProj or package.

Usage

```
quick_start3(output_file_name = NULL)
```

Arguments

`output_file_name`	Optional. A file name to use for the script. Defaults to "easysurv_start.R" within a helper function.

Value

A new R script file with example code.

Examples

```
quick_start3()
```

`test_ph` *Test Proportional Hazards Assumption*

Description

Assesses the proportional hazards assumption for survival data using a Cox proportional hazards model and related tests.

Usage

```
test_ph(data, time, event, group, plot_theme = theme_easysurv())
```

Arguments

`data`	A data frame containing the survival data.
`time`	The name of the column in `data` containing the time-to-event information.
`event`	The name of the column in `data` indicating whether the event of interest occurred.
`group`	The name of the column in `data` defining the grouping variable.
`plot_theme`	The theme to be used for the plots.

Value

A list containing plots and test results related to the assessment of the proportional hazards assumption.

`cloglog_plot`	A plot of the log cumulative hazard function. If the lines are roughly parallel, this suggests that the proportional hazards assumption holds."
`coxph_model`	The coefficients from the Cox proportional hazards model. The exp(coef) column shows the hazard ratio.
`survdiff`	The results of the log-rank test for differences in survival curves between groups. A p-value less than 0.05 suggests that survival differences between groups are statistically significant.
`coxph_test`	The results of the proportional hazards assumption test. A p-value less than 0.05 suggests that the proportional hazards assumption may be violated.
`schoenfeld_plot`	A plot of the Schoenfeld residuals. A flat smoothed line close to zero supports the proportional hazards assumption. A non-flat smoothed line with a trend suggests the proportional hazards assumption is violated.

Examples

```
ph_results <- test_ph(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  group = "group"
)

ph_results
```

`theme_easysurv` *Plot Theme for easysurv Survival and Hazard Plots*

Description

Plot Theme for easysurv Survival and Hazard Plots

Usage

```
theme_easysurv()
```

Value

A ggplot2 theme object.

Examples

```
library(ggsurvfit)
fit <- survfit2(Surv(time, status) ~ surg, data = df_colon)
fit |> ggsurvfit() + theme_easysurv()
```

`theme_risktable_easysurv` *Plot Theme for easysurv Risk Tables*

Description

To be used with `ggsurvfit::add_risktable()`.

Usage

```
theme_risktable_easysurv()
```

Value

A list containing a ggplot2 theme object.

Examples

```
library(ggsurvfit)
fit <- survfit2(Surv(time, status) ~ surg, data = df_colon)
fit <- fit |> ggsurvfit() +
  theme_easysurv() +
  add_risktable(theme = theme_risktable_easysurv())
fit
```

`write_to_xl` *Export easysurv output to Excel via* `openxlsx`

Description

Export easysurv output to Excel via `openxlsx`

Usage

```
write_to_xl(wb, object)
```

Arguments

`wb`	A Workbook object containing a worksheet
`object`	The output of an easysurv command

Value

An Excel workbook with the easysurv output.

Examples

```
km_results <- get_km(
  data = easysurv::easy_bc,
  time = "recyrs",
  event = "censrec",
  group = "group",
  risktable_symbols = FALSE
)

wb <- openxlsx::createWorkbook()

## Not run:
write_to_xl(wb, km_results)
openxlsx::saveWorkbook(wb, "km_results.xlsx", overwrite = TRUE)
openxlsx::openXL("km_results.xlsx")

## End(Not run)
```

Index