Package ‘ezcox’

May 8, 2023

Type Package

Title Easily Process a Batch of Cox Models

Version 1.0.4

Date 2023-5-8

Maintainer Shixiang Wang <w_shixiang@163.com>

Description A tool to operate a batch of univariate or multivariate Cox models and return tidy result.

License GPL-3

URL https://github.com/ShixiangWang/ezcox

BugReports https://github.com/ShixiangWang/ezcox/issues

Depends R (>= 3.5)

Imports dplyr (>= 0.8.3), forestmodel, ggplot2, magrittr (>= 1.5), methods, purrr (>= 0.3.2), rlang (>= 0.1.2), scales, survival, tibble, utf8, utils

Suggests covr (>= 3.2.1), furrr, future, knitr, prettydoc, rmarkdown, roxygen2 (>= 6.1.1), testthat (>= 2.1.0)

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Shixiang Wang [aut, cre] (<https://orcid.org/0000-0001-9855-7357>)

Repository CRAN

Date/Publication 2023-05-08 09:50:11 UTC

R topics documented:

`clean_model_dir` *Clean ezcox Model File Directory*

Description

Clean ezcox Model File Directory

Usage

```
clean_model_dir(model_dir = file.path(tempdir(), "ezcox"))
```

Arguments

`model_dir`	a path for storing model results.

Value

nothing

Examples

```
clean_model_dir()
```

`ezcox` *Run Cox Analysis in Batch Mode*

Description

Run Cox Analysis in Batch Mode

Usage

```
ezcox(
  data,
  covariates,
  controls = NULL,
  time = "time",
  status = "status",
  global_method = c("likelihood", "wald", "logrank"),
  keep_models = FALSE,
  return_models = FALSE,
  model_dir = file.path(tempdir(), "ezcox"),
  verbose = TRUE,
  ...
)
```

Arguments

`data`	a `data.frame` containing variables, time and os status.
`covariates`	column names specifying variables.
`controls`	column names specifying controls. The names with pattern "*:|()" will be treated as interaction/combination term, please make sure all column names in `data` are valid R variable names.
`time`	column name specifying time, default is 'time'.
`status`	column name specifying event status, default is 'status'.
`global_method`	method used to obtain global p value for cox model, should be one of "likelihood", "wald", "logrank". The likelihood-ratio test, Wald test, and score logrank statistics. These three methods are asymptotically equivalent. For large enough N, they will give similar results. For small N, they may differ somewhat. The Likelihood ratio test has better behavior for small sample sizes, so it is generally preferred.
`keep_models`	If `TRUE`, keep models as local files.
`return_models`	default `FALSE`. If `TRUE`, return a `list` contains cox models.
`model_dir`	a path for storing model results.
`verbose`	if `TRUE`, print extra info.
`...`	other parameters passing to `survival::coxph()`.

Value

a `ezcox` object

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
library(survival)

# Build unvariable models
t1 <- ezcox(lung, covariates = c("age", "sex", "ph.ecog"))
t1

# Build multi-variable models
# Control variable 'age'
t2 <- ezcox(lung, covariates = c("sex", "ph.ecog"), controls = "age")
t2

# Return models
t3 <- ezcox(lung,
  covariates = c("age", "sex", "ph.ecog"),
  return_models = TRUE
)
t3
t4 <- ezcox(lung,
  covariates = c("sex", "ph.ecog"), controls = "age",
  return_models = TRUE
)
t4
```

`ezcox_group` *Group Cox Analysis and Visualization*

Description

Group Cox Analysis and Visualization

Usage

```
ezcox_group(
  data,
  grp_var,
  covariate,
  controls = NULL,
  time = "time",
  status = "status",
  sort = FALSE,
  decreasing = TRUE,
  add_all = FALSE,
  add_caption = TRUE,
  verbose = TRUE,
 headings = list(variable = "Group", n = "N", measure = "Hazard ratio", ci = NULL, p =
    "p"),
  ...
)
```

Arguments

`data`	a `data.frame` containing variables, time and os status.
`grp_var`	a group column.
`covariate`	a covariable for cox analysis.
`controls`	column names specifying controls. The names with pattern "*:|()" will be treated as interaction/combination term, please make sure all column names in `data` are valid R variable names.
`time`	column name specifying time, default is 'time'.
`status`	column name specifying event status, default is 'status'.
`sort`	if `TRUE`, sort the models by the HR values.
`decreasing`	logical, should the sort order be increasing or decreasing?
`add_all`	if `TRUE`, add a group for all data rows.
`add_caption`	if `TRUE`, add caption to the plot.
`verbose`	if `TRUE`, print extra info.
`headings`	a `list` for setting the heading text.
`...`	other arguments passing to `forestmodel::forest_model()`.

Value

a `list`.

Examples

```
library(survival)
ezcox_group(lung, grp_var = "sex", covariate = "ph.ecog")
ezcox_group(lung, grp_var = "sex", covariate = "ph.ecog", controls = "age")
p <- ezcox_group(lung,
  grp_var = "sex", covariate = "ph.ecog",
  controls = "age", add_all = TRUE
)
```

`ezcox_parallel` *Parallelly Run Cox Analysis in Batch Mode*

Description

Parallelly Run Cox Analysis in Batch Mode

Usage

```
ezcox_parallel(
  data,
  covariates,
  controls = NULL,
  time = "time",
  status = "status",
  batch_size = 100,
  global_method = c("likelihood", "wald", "logrank"),
  keep_models = FALSE,
  return_models = FALSE,
  model_dir = file.path(tempdir(), "ezcox"),
  parallel = TRUE,
  verbose = FALSE
)
```

Arguments

`data`	a `data.frame` containing variables, time and os status.
`covariates`	column names specifying variables.
`controls`	column names specifying controls.
`time`	column name specifying time, default is 'time'.
`status`	column name specifying event status, default is 'status'.
`batch_size`	processing size in a batch.
`global_method`	method used to obtain global p value for cox model, should be one of "likelihood", "wald", "logrank". The likelihood-ratio test, Wald test, and score logrank statistics. These three methods are asymptotically equivalent. For large enough N, they will give similar results. For small N, they may differ somewhat. The Likelihood ratio test has better behavior for small sample sizes, so it is generally preferred.
`keep_models`	If `TRUE`, keep models as local files.
`return_models`	default `FALSE`. If `TRUE`, return a `list` contains cox models.
`model_dir`	a path for storing model results.
`parallel`	if `TRUE`, do parallel computation by **furrr** package.
`verbose`	if `TRUE`, print extra info. If `parallel` is `TRUE`, set `verbose` to `FALSE` may speed up.

Value

a `ezcox` object

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
library(survival)
t <- ezcox_parallel(lung, covariates = c("sex", "ph.ecog"), controls = "age")
t
```

`filter_ezcox` *Filter ezcox*

Description

Filter ezcox

Usage

```
filter_ezcox(x, levels = "auto", type = c("both", "contrast", "ref"))
```

Arguments

`x`	a ezcox object from `ezcox()`.
`levels`	levels to filter, default is 'auto', it will filter all control variables.
`type`	default is 'both' for filtering both contrast level and reference level. It can also be 'contrast' for filtering only contrast level and 'ref' for filtering only reference level.

Value

a ezcox object

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
library(survival)
lung$ph.ecog <- factor(lung$ph.ecog)
zz <- ezcox(lung, covariates = c("sex", "age"), controls = "ph.ecog")
zz
filter_ezcox(zz)
filter_ezcox(zz, c("0", "2"))
filter_ezcox(zz, c("0", "2"), type = "contrast")
t <- filter_ezcox(zz, c("0", "2"), type = "ref")
t
```

`forester` *Create a forest plot for simple data*

Description

Create a forest plot for simple data

Usage

```
forester(
  data,
  display_cols = c("Variable", "HR", "lower_95", "upper_95"),
  estimate_precision = 2,
  null_line_at = 1,
  font_family = "mono",
  x_scale_linear = TRUE,
  xlim = NULL,
  xbreaks = NULL,
  point_sizes = 3,
  point_shape = 16,
  label_hjust = 0,
  label_vjust = -1,
  label_color = "blue",
  label_size = 3
)
```

Arguments

`data`	Data frame (required). The information to be displayed as the forest plot.
`display_cols`	4 columns stand for axis text and the forest data, default using `c("term", "HR", "conf.low", "conf.high")`.
`estimate_precision`	Integer. The number of decimal places on the estimate (default 2).
`null_line_at`	Numeric. Default 0. Change to 1 if using relative measures such as OR, RR.
`font_family`	String. The font to use for the ggplot. Default "mono".
`x_scale_linear`	Logical. Default TRUE, change to FALSE for log scale
`xlim`	Vector. Manually specify limits for the x axis as a vector length 2, i.e. c(low, high)
`xbreaks`	Vector. X axis breaks to label. Specify limits in xlim if using this option.
`point_sizes`	Vector. Length should be equal to 1 or nrow(left_side_data). The sizes of the points in the center plot, where 3.25 is the default.
`point_shape`	Vector. Length should be equal to 1 or nrow(left_side_data). The shapes of the points in the center plot, where 16 (a filled circle) is the default.
`label_hjust, label_vjust, label_color, label_size`	hjust, vjust color and size for the label text.

Value

a `ggplot` object.

Examples

```
library(survival)

t1 <- ezcox(lung, covariates = c(
  "age", "sex",
  "ph.karno", "pat.karno"
))
p <- forester(t1, xlim = c(0, 1.5))
p
p2 <- forester(t1, xlim = c(0.5, 1.5))
p2
```

`get_models` *Get Model List from ezcox Object*

Description

Models are renamed by the formulas.

Usage

```
get_models(x, variables = NULL)
```

Arguments

`x`	a `ezcox` object from `ezcox()`.
`variables`	a character vector representing variables to select.

Value

a named `list` with class `ezcox_models`

Examples

```
library(survival)
zz <- ezcox(lung, covariates = c("sex", "ph.ecog"), controls = "age", return_models = TRUE)
mds <- get_models(zz)
str(mds, max.level = 1)
```

`show_forest` *Show Forest Plot*

Description

This is a wrapper of function ezcox, get_models and show_models. It focus on generating forest plot easily and flexibly.

Usage

```
show_forest(
  data,
  covariates,
  controls = NULL,
  time = "time",
  status = "status",
  merge_models = FALSE,
  model_names = NULL,
  vars_to_show = NULL,
  drop_controls = FALSE,
  add_caption = TRUE,
  point_size = 3,
  point_shape = 15,
  color = "red",
  banded = TRUE,
 headings = list(variable = "Variable", n = "N", measure = "Hazard ratio", ci = NULL, p
    = "p"),
  model_dir = file.path(tempdir(), "ezcox"),
  verbose = TRUE,
  ...
)
```

Arguments

`data`	a `data.frame` containing variables, time and os status.
`covariates`	a character vector optionally listing the variables to include in the plot (defaults to all variables).
`controls`	column names specifying controls. The names with pattern "*:|()" will be treated as interaction/combination term, please make sure all column names in `data` are valid R variable names.
`time`	column name specifying time, default is 'time'.
`status`	column name specifying event status, default is 'status'.
`merge_models`	if 'TRUE', merge all models and keep the plot tight.
`model_names`	model names to show when `merge_models=TRUE`.

`vars_to_show`	default is `NULL`, show all variables (including controls). You can use this to choose variables to show, but remember, the models have not been changed.
`drop_controls`	works when `covariates=NULL` and `models` is a `ezcox_models`, if `TRUE`, it removes control variables automatically.
`add_caption`	if `TRUE`, add caption to the plot.
`point_size`	size of point.
`point_shape`	shape value of point.
`color`	color for point and segment.
`banded`	if `TRUE` (default), create banded background color.
`headings`	a `list` for setting the heading text.
`model_dir`	a path for storing model results.
`verbose`	if `TRUE`, print extra info.
`...`	other arguments passing to `forestmodel::forest_model()`.

Value

a `ggplot` object

Examples

```
library(survival)
show_forest(lung, covariates = c("sex", "ph.ecog"), controls = "age")
show_forest(lung, covariates = c("sex", "ph.ecog"), controls = "age", merge_models = TRUE)
show_forest(lung,
  covariates = c("sex", "ph.ecog"), controls = "age", merge_models = TRUE,
  drop_controls = TRUE
)
p <- show_forest(lung,
  covariates = c("sex", "ph.ecog"), controls = "age", merge_models = TRUE,
  vars_to_show = "sex"
)
p
```

`show_models` *Show Cox Models*

Description

Show Cox Models

Usage

```
show_models(
  models,
  model_names = NULL,
  covariates = NULL,
  merge_models = FALSE,
  drop_controls = FALSE,
 headings = list(variable = "Variable", n = "N", measure = "Hazard ratio", ci = NULL, p
   = "p"),
  ...
)
```

Arguments

`models`	a `ezcox_models` from `get_models()` or a (named) list of Cox models.
`model_names`	model names to show when `merge_models=TRUE`.
`covariates`	a character vector optionally listing the variables to include in the plot (defaults to all variables).
`merge_models`	if 'TRUE', merge all models and keep the plot tight.
`drop_controls`	works when `covariates=NULL` and `models` is a `ezcox_models`, if `TRUE`, it removes control variables automatically.
`headings`	a `list` for setting the heading text.
`...`	other arguments passing to `forestmodel::forest_model()`.

Value

a `ggplot` object

Examples

```
library(survival)
zz <- ezcox(lung, covariates = c("sex", "ph.ecog"), controls = "age", return_models = TRUE)
mds <- get_models(zz)
show_models(mds)
show_models(mds, model_names = paste0("Model ", 1:2))
show_models(mds, covariates = c("sex", "ph.ecog"))
show_models(mds, drop_controls = TRUE)
show_models(mds, merge_models = TRUE)
p <- show_models(mds, merge_models = TRUE, drop_controls = TRUE)
p
```

Index