

Package ‘ezcox’

January 10, 2020

Type Package

Title Easily Process a Batch of Cox Models

Version 0.4.0

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), magrittr (>= 1.5), purrr (>= 0.3.2), survival, rlang (>= 0.1.2), forestmodel, utils

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

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

VignetteBuilder knitr

NeedsCompilation no

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

Repository CRAN

Date/Publication 2020-01-10 14:30:02 UTC

R topics documented:

clean_model_dir	2
ezcox	2
ezcox_parallel	4
filter_ezcox	5

get_models	6
show_forest	6
show_models	8

Index 9

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

<code>data</code>	a <code>data.frame</code> containing variables, time and os status.
<code>covariates</code>	column names specifying variables.
<code>controls</code>	column names specifying controls.
<code>time</code>	column name specifying time, default is 'time'.
<code>status</code>	column name specifying event status, default is 'status'.
<code>global_method</code>	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.
<code>keep_models</code>	If TRUE, keep models as local files.
<code>return_models</code>	default FALSE. If TRUE, return a list contains cox models.
<code>model_dir</code>	a path for storing model results.
<code>verbose</code>	if TRUE, print extra info.

Value

a `ezcox` object

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
library(survival)

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

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

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

 ezcox_parallel *Parallely Run Cox Analysis in Batch Mode*

Description

Parallely 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)
ezcox_parallel(lung, covariates = c("sex", "ph.ecog"), controls = "age")
```

filter_ezcox	<i>Filter ezcox</i>
--------------	---------------------

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")
filter_ezcox(zz, c("0", "2"), type = "ref")
```

get_models	<i>Get Model List from ezcox Object</i>
------------	---

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	<i>Show Forest Plot</i>
-------------	-------------------------

Description

This is a wrapper of function `ezcox`, `get_models` and `show_models`. If 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,
    model_dir = file.path(tempdir(), "ezcox"),
    ...
  )

```

Arguments

<code>data</code>	a <code>data.frame</code> containing variables, time and os status.
<code>covariates</code>	column names specifying variables.
<code>controls</code>	column names specifying controls.
<code>time</code>	column name specifying time, default is 'time'.
<code>status</code>	column name specifying event status, default is 'status'.
<code>merge_models</code>	if 'TRUE', merge all models and keep the plot tight.
<code>model_names</code>	model names to show when <code>merge_models=TRUE</code> .
<code>vars_to_show</code>	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.
<code>drop_controls</code>	works when <code>covariates=NULL</code> and <code>models</code> is a <code>ezcox_models</code> , if TRUE, it removes control variables automatically.
<code>model_dir</code>	a path for storing model results.
<code>...</code>	other arguments passing to <code>forestmodel::forest_model()</code> .

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
)
show_forest(lung,
  covariates = c("sex", "ph.ecog"), controls = "age", merge_models = TRUE,
  vars_to_show = "sex"
)

```

 show_models

Show Cox Models

Description

Show Cox Models

Usage

```
show_models(
  models,
  model_names = NULL,
  covariates = NULL,
  merge_models = FALSE,
  drop_controls = FALSE,
  ...
)
```

Arguments

models	a ezcox_models from <code>get_models()</code> 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.
...	other arguments passing to <code>forestmodel::forest_model()</code> .

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)
show_models(mds, merge_models = TRUE, drop_controls = TRUE)
```

Index

`clean_model_dir`, 2

`ezcox`, 2, 6

`ezcox()`, 5, 6

`ezcox_parallel`, 4

`filter_ezcox`, 5

`forestmodel::forest_model()`, 7, 8

`get_models`, 6, 6

`get_models()`, 8

`show_forest`, 6

`show_models`, 6, 8