Package: BlanketStatsments (via r-universe)

May 11, 2026

Title Build and Compare Statistical Models

Version 0.1.3

Description Build and compare nested statistical models with sets of equal and different independent variables. An analysis using this package is Marquardt et al. (2021) <https://github.com/p-mq/Percentile_based_averaging>.

Depends R (>= 4.0.0)

Imports basecamb, survival, survAUC, DescTools, Hmisc, stats, utils

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

URL https://github.com/p-mq/BlanketStatsments

BugReports https://github.com/p-mq/BlanketStatsments/issues

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev

Repository https://p-mq.r-universe.dev

Date/Publication 2026-04-07 06:19:44 UTC

RemoteUrl https://github.com/p-mq/blanketstatsments

RemoteRef HEAD

RemoteSha 03c397df08e7800aec6f07cd35c82b1adbb4c9d7

Contents

.build_model_formula 2
blanket_c_statistic 3
blanket_redundancy_analysis 3
blanket_stats 4
blanket_statsments 5

build_cox_model 6
build_reg_model 7
calculate_Uno_c 8
redundancy_analysis 9
table_blanket_redundancies 10
table_blanket_statsments 11
table_predictors 12

Index **13**

`.build_model_formula` *Build formula for statistical models*

Description

Build formula used in statistical models from vectors of strings. Copied from basecamb package to avoid dependency

Usage

```
.build_model_formula(outcome, predictors, censor_event = NULL)
```

Arguments

`outcome`	character denoting the column with the outcome.
`predictors`	vector of characters denoting the columns with the predictors.
`censor_event`	character denoting the column with the censoring event, for use in Survival-type models.

Value

formula for use in statistical models

Author(s)

J. Peter Marquardt

Source

build_model_formula

blanket_c_statistic *Generic wrapper method to calculate C-statistics*

Description

Calculate concordance statistics for a list of statistical models on the same data set

Usage

```
blanket_c_statistic(df, model_list, modality = "logistic", verbose = FALSE)
```

Arguments

df	data.frame containing the data set. If evaluating independently, use the test set.
model_list	list of statistical models of type lm, glm or coxph to be evaluated.
modality	character specifying model type. Currently accepts 'linear', 'logistic', and 'cox'
verbose	logical. TRUE activates printout messages.

Value

list of doubles with the AUC values for the evaluated models on the specified data set.

Author(s)

J. Peter Marquardt

blanket_redundancy_analysis *Blanket redundancy analysis*

Description

Perform a blanket redundancy analysis on a list of existing models

Usage

```
blanket_redundancy_analysis(
  model_list,
  data,
  r2_threshold = 0.9,
  nk = 0,
  verbose = FALSE
)
```

Arguments

model_list	a list of statistical regression model of class linear, logistic or coxph
data	data.frame used to create the models
r2_threshold	float threshold value to consider a parameter redundant
nk	number of knots in splicing
verbose	ctivate printouts of key findings

Value

an list of objects of class "redun"

Author(s)

J. Peter Marquardt

See Also

[blanket_stats()]

Examples

```
data <- survival::lung
models_to_run <- list(
'OS' = list('outcome' = 'time', 'modality' = 'cox', 'event_censor' = 'status'),
'weight_loss' = list('outcome' = 'wt.loss', 'modality' = 'linear', 'event_censor' = NA))
predictor_sets <- list('age' = c('age'), 'age_ecog' = c('age', 'ph.ecog'))
covariates = c('sex')
bl_stats <- blanket_statsments(data, models_to_run, predictor_sets, covariates)
blanket_redundancy_analysis(bl_stats, data)
```

---

blanket_stats *Run multiple slightly different models of same type*

Description

Run the same model (type, outcome, and covariates) with different sets of predictors

Usage

```
blanket_stats(
  df,
  outcome,
  predictor_sets,
  covariates = c(),
  modality = "linear",
  event_censor = NA,
  verbose = FALSE
)
```

Arguments

df	data.frame containing the data set.
outcome	character designating the column with the outcome of interest
predictor_sets	named list or character vectors containing columns with predictors
covariates	vector of characters denoting columns with covariables
modality	character denoting model type. Currently limited to 'linear', 'logistic', and 'cox'
event_censor	character denoting column with censor event. For coxph models only
verbose	logical. TRUE activates printout messages.

Value

named list of models

Author(s)

J. Peter Marquardt

Examples

```
data <- survival::lung
outcome <- 'time'
predictor_sets <- list('age' = c('age'),'age_ecog' = c('age', 'ph.ecog'))
covariates = c('sex')
modality <- 'cox'
event_censor <- 'status'
bl_stats <- blanket_stats(data, outcome, predictor_sets, covariates, modality, event_censor)
```

---

blanket_statsments *Run multiple different models with different sets of predictors*

Description

Wraps blanket_stats. Run a list of models with different modalities/outcomes for a list of different predictor sets with the same covariables.

Usage

```
blanket_statsments(
  df,
  models_to_run,
  predictor_sets,
  covariates = c(),
  verbose = FALSE
)
```

Arguments

df	data.frame containing the data set.
models_to_run	either a named list or data.frame type, with every entry/row having the keys/columns outcome, modality, and event_censor
predictor_sets	named list of lists containing the set of predictors. See blanket_stats for details
covariates	vector of characters denoting columns with covariables
verbose	logical. TRUE activates printout messages.

Value

named list of named lists of models

Author(s)

J. Peter Marquardt

Examples

```
data <- survival::lung
models_to_run <- list('OS' = list(
'outcome' = 'time', 'modality' = 'cox', 'event_censor' = 'status'),
'weight_loss' = list('outcome' = 'wt.loss', 'modality' = 'linear', 'event_censor' = NA))
predictor_sets <- list('age' = c('age'),'age_ecog' = c('age', 'ph.ecog'))
covariates = c('sex')
bl_stats <- blanket_statsments(data, models_to_run, predictor_sets, covariates)
```

---

build_cox_model *Build a cox model*

Description

Build a Cox proportional hazards model from data and meta-parameters

Usage

```
build_cox_model(
  df,
  event_time,
  event_censor,
  predictors,
  covariates = c(),
  verbose = FALSE
)
```

Arguments

df	data.frame containing the data set
event_time	character denoting column with event time
event_censor	character denoting column specifying events/censoring
predictors	character vector denoting columns with independent variables of interest
covariates	character vector denoting columns with independent variables not of interest. Covariates are mathematically identical to predictors but will be ignored in reporting
verbose	logical. TRUE activates printout messages

Value

A Cox proportional hazards model

Author(s)

J. Peter Marquardt

Examples

```
data <- survival::lung
mod <- build_cox_model(data, 'time', 'status', c('age', 'sex'))
```

---

build_reg_model *Build a generic regression model model*

Description

Build a generic regression model from data and meta-parameters. Currently only available for linear and logistic types.

Usage

```
build_reg_model(
  df,
  outcome,
  predictors,
  covariates = c(),
  modality = "linear",
  verbose = FALSE
)
```

Arguments

df	data.frame containing the data set
outcome	character denoting column with the outcome of interest
predictors	character vector denoting columns with independent variables of interest
covariates	character vector denoting columns with independent variables not of interest. Covariates are mathematically identical to predictors but will be ignored in reporting
modality	character designating type. Currently limited to 'linear' and 'logistic'.
verbose	logical. TRUE activates printout messages

Value

A regression model of linear or logistic type

Author(s)

J. Peter Marquardt

Examples

```
mod <- build_reg_model(data.frame('outcome' = c(1,2), 'pred' = c(3,4)), 'outcome', c('pred'))
```

---

calculate_Uno_c *Calculate Uno's C for a given model.*

Description

Calculate Uno's concordance statistic for any model. CAVE: If you want to evaluate a model trained on a different dataset, df should be limited to the test set.

Usage

```
calculate_Uno_c(df, model, verbose = FALSE)
```

Arguments

df	data.frame containing the data set. If evaluating independently, use the test set.
model	statistical model of type coxph to be evaluated.
verbose	logical. TRUE activates printout messages.

Value

double AUC value for the evaluated model on the specified data set.

Author(s)

J. Peter Marquardt

Examples

```
data <- survival::lung
cancer_mod <- survival::coxph(survival::Surv(time, status)~age, data = data)
calculate_Uno_c(data, cancer_mod)
```

---

redundancy_analysis *Redundancy analysis*

Description

Perform a redundancy analysis on an existing model

Usage

```
redundancy_analysis(model, data, r2_threshold = 0.9, nk = 0)
```

Arguments

model	a statistical regression model of class linear, logistic or coxph
data	data.frame used to create the model
r2_threshold	float threshold value to consider a parameter redundant
nk	number of knots in splicing

Value

an object of class "redun"

Author(s)

J. Peter Marquardt

Examples

```
data <- survival::lung
mod <- build_reg_model(data, 'meal.cal', c('sex', 'age'))
redundancy_analysis(mod, data)
```

table_blanket_redundancies *Table results of blanket redundancy analysis*

Description

Table results of a blanket redundancy analysis on a list of existing models

Usage

```
table_blanket_redundancies(blanket_redundancies, digits = 2)
```

Arguments

blanket_redundancies	list of lists of redun objects generated by blanket_redundancy_analysis()
digits	integer number of decimals to include

Value

a data.frame tabling the key results

Author(s)

J. Peter Marquardt

See Also

[table_predictors()], [blanket_redundancy_analysis()]

Examples

```
data <- survival::lung
models_to_run <- list(
'OS' = list('outcome' = 'time', 'modality' = 'cox', 'event_censor' = 'status'),
'weight_loss' = list('outcome' = 'wt.loss', 'modality' = 'linear', 'event_censor' = NA))
predictor_sets <- list('age' = c('age'), 'age_ecog' = c('age', 'ph.ecog'))
covariates = c('sex')
bl_stats <- blanket_statsments(data, models_to_run, predictor_sets, covariates)
bl_redun <- blanket_redundancy_analysis(bl_stats, data)
table_blanket_redundancies(bl_redun)
```

table_blanket_statsments *Table results of multiple different models with different sets of predictors*

Description

Wraps blanket_stats. Run a list of models with different modalities/outcomes for a list of different predictor sets with the same covariables.

Usage

```
table_blanket_statsments(df, blanket_statsment_models)
```

Arguments

df	data.frame containing the data set.
blanket_statsment_models	list of models produced by blanket_statsments()

Value

data.frame with tabled results

Author(s)

J. Peter Marquardt

See Also

[blanket_statsments()] for models and [table_predictors()] for tabling results

Examples

```
data <- survival::lung
models_to_run <- list('OS' = list(
'outcome' = 'time', 'modality' = 'cox', 'event_censor' = 'status'),
'weight_loss' = list('outcome' = 'wt.loss', 'modality' = 'linear', 'event_censor' = NA))
predictor_sets <- list('age' = c('age'),'age_ecog' = c('age', 'ph.ecog'))
covariates = c('sex')
bl_stats <- blanket_statsments(data, models_to_run, predictor_sets, covariates)
tbl <- table_blanket_statsments(data, bl_stats)
```

`table_predictors` *Table model predictor performance*

Description

Extract coefficients and p-values only for regression models and table them

Usage

```
table_predictors(df, model, predictors)
```

Arguments

`df`	data.frame containing the data set. If evaluating independently, use the test set.
`model`	statistical model to be evaluated.
`predictors`	vector of characters designating columns of interest. Non-specified independent variables will not be included.

Value

data.frame with coefficients and p-values for predictor variables

Author(s)

J. Peter Marquardt

Examples

```
data <- survival::lung
mod <- build_reg_model(data, 'age', 'sex')
tbl <- table_predictors(data, mod, 'sex')
```

Index

.build_model_formula, 2

blanket_c_statistic, 3
blanket_redundancy_analysis, 3
blanket_stats, 4
blanket_statsments, 5
build_cox_model, 6
build_model_formula, 2
build_reg_model, 7

calculate_Uno_c, 8

redundancy_analysis, 9

table_blanket_redundancies, 10
table_blanket_statsments, 11
table_predictors, 12