Package: hotfun (via r-universe)

November 5, 2024

Title Collection of Functions Used in the Health Outcomes Team at MSKCC

Version 0.3.0

Description A mixed-bag of utility functions to ease analyses and reporting results.

License MIT + file LICENSE

Depends R (>= 3.4)

Imports broom (>= 0.7.0), cli (>= 3.0.1), dplyr (>= 1.0.0), forcats (>= 0.5.0), fs (>= 1.5.0), glue (>= 1.4.1), gt (>= 0.2.1), gtsummary (>= 1.4.1), here (>= 0.1), Hmisc (>= 4.4.0), knitr (>= 1.29), labelled (>= 2.5.0), lifecycle (>= 1.0.1), lubridate (>= 1.7.9), magrittr (>= 1.5), purrr (>= 0.3.4), readxl (>= 1.3.1), rlang (>= 0.4.7), rstudio.prefs (>= 0.1.5), starter (>= 0.1.5), stringr (>= 1.4.0), tibble (>= 3.0.3), tidyr (>= 1.1.0), tidyselect (>= 1.1.0)

Suggests covr (>= 3.5.0), ggplot2 (>= 3.3.2), spelling (>= 2.1), testthat (>= 2.3.2)

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Config/pak/sysreqs git make libgit2-dev libicu-dev libxml2-dev libssl-dev libnode-dev libx11-dev zlib1g-dev

Repository https://ddsjoberg.r-universe.dev

RemoteUrl https://github.com/ddsjoberg/hotfun

RemoteRef v0.3.0

RemoteSha f30f4d729352eb789891aff5dcf8931adee1100e

Contents

`add_splines` *Add spline terms to a data frame*

Description

Adds spline terms calculated via `Hmisc::rcspline.eval()` to a data frame.

Usage

```
add_splines(data, variable, knots = NULL, nk = 5, norm = 2, new_names = NULL)
```

Arguments

`data`	a data frame
`variable`	name of column in data
`knots`	knot locations. If not given, knots will be estimated using default quantiles of `x`. For 3 knots, the outer quantiles used are 0.10 and 0.90. For 4-6 knots, the outer quantiles used are 0.05 and 0.95. For `nk` > 6, the outer quantiles are 0.025 and 0.975. The knots are equally spaced between these on the quantile scale. For fewer than 100 non-missing values of `x`, the outer knots are the 5th smallest and largest `x`.
`nk`	number of knots. Default is 5. The minimum value is 3.

norm	'0' to use the terms as originally given by *Devlin and Weeks (1986)*, '1' to normalize non-linear terms by the cube of the spacing between the last two knots, '2' to normalize by the square of the spacing between the first and last knots (the default). norm=2 has the advantage of making all nonlinear terms beon the x-scale.
new_names	Optionally specify names of new spline columns

Value

data frame

Knot Locations

Knot locations are returned in `attr(data[[variable]], "knots")`

Examples

```
trial %>%
  add_splines(age)
```

---

assign_timepoint *Assign a timepoint to a long dataset with multiple measures*

Description

Given a data set that has a measure collected over time and you want to extract, for example the 3 month measurement, this function will find the measure closest to 3 months within a defined window.

Usage

```
assign_timepoint(
  data,
  id,
  ref_date,
  measure_date,
  timepoints,
  windows,
  time_units = c("days", "weeks", "months", "years"),
  new_var = "timepoint"
)
```

Arguments

data	data frame
id	id variable name, such as "mrn"
ref_date	baseline or reference date column name
measure_date	date the measure was collected
timepoints	vector of timepoint to identify
windows	list of windows around a timepoint that are acceptable
time_units	one of c("days", "weeks", "months", "years")
new_var	name of new variable, default is "timepoint"

Value

data frame passed in data with additional column new_var

Examples

```
ggplot2::economics_long %>%
  dplyr::group_by(variable) %>%
  dplyr::mutate(min_date = min(date)) %>%
  dplyr::ungroup() %>%
  assign_timepoint(
    id = "variable",
    ref_date = "min_date",
    measure_date = "date",
    timepoints = c(6, 12, 24),
    windows = list(c(-2, 2), c(-2, 2), c(-2, 2)),
    time_units = "months"
  ) %>%
  dplyr::filter(!is.na(timepoint))
```

---

auc_density *Calculate exact AUCs based on the distribution of risk in a population*

Description

Provided a distribution of risk in a population, this function calculates the exact AUC of a model that produces the risk estimates. For example, a logistic regression model built with a normal linear predictor yields logit-normal distributed predicted risks. The AUC from the logistic regression model is the same as the AUC estimated from the distribution of the predicted risks, independent of the outcome. This method for AUC calculation is useful for simulation studies where the predicted risks are a mixture of two distributions. The exact prevalence of the outcome can easily be calculated, along with the exact AUC of the model.

Usage

```
auc_density(density, cut.points = seq(from = 0, to = 1, by = 0.001), ...)
```

Arguments

density	a function name that describes the continuous probability density function of the risk from 0 to 1.
cut.points	sequence of points in [0, 1] where the sensitivity and specificity are calculated. More points lead to a more precise estimate of the AUC. Default is seq(from = 0, to = 1,by = 0.001).
...	arguments for the function specified in density. For example, dbeta(x, shape1=1, shape2=1) has need for two additional arguments to specify the density function (shape1 and shape2).

Value

Returns a list sensitivity and specificity at each cut point, the expected value or mean risk, and the AUC associated with the distribution.

Author(s)

Daniel D Sjoberg <sjobergd@mskcc.org>

Examples

```
auc_density(density = dbeta, shape1 = 1, shape2 = 1)
```

---

auc_histogram *Calculate an AUC from a histogram*

Description

Uses a histogram of event probabilities to calculate a precise AUC. This is a discrete approximation. Use this function with many break points with a large number of data points.

Usage

```
auc_histogram(x)
```

Arguments

x	histogram object from graphics::hist

Author(s)

Daniel D. Sjoberg

Examples

```
runif(10000) %>%
  hist(breaks = 250) %>%
  auc_histogram()
```

clean_mrn *Check and Format MRNs*

Description

An MRN follows specific rules

1. Must be character
2. Must contain only numeric components
3. Must be eight characters long and include leading zeros.

This function converts numeric MRNs to character and ensures it follows MRN conventions. Character MRNs can also be passed, and leading zeros will be appended and checked for consistency.

Usage

```
clean_mrn(x, allow_na = FALSE, check_unique = FALSE)
```

Arguments

x	vector to be converted and checked to MRN
allow_na	logical indicating whether NA values are accepted. Default is FALSE
check_unique	Check if MRNs are unique

Value

character MRN vector

Examples

```
1000:1001 %>%
  clean_mrn()
```

count_map *Checks variable creation for new derived variables at once*

Description

Checks variable creation for new derived variables at once

Usage

```
count_map(data, checks)
```

Arguments

data	data frame
checks	list of variables to check.

Examples

```
count_map(
  mtcars,
  list(c("cyl", "am"), c("gear", "carb"))
)
```

---

count_na *Assess pattern of missing data*

Description

Pass a data frame and the missing pattern of all columns in the data frame. The data frame is returned unmodified.

Usage

```
count_na(data, include = NULL, exclude = NULL)
```

Arguments

data	data frame
include	character vector of names to include
exclude	character vector of names to exclude

Examples

```
trial %>% count_na()
```

---

create_hot_project *Start a new H.O.T. project*

Description

Creates a directory with the essential files for a new project. The function can be used on existing project directories as well. This is a thin wrapper for `starter::create_project()` that sets the default template to `template = hotfun::project_template`

Usage

```
create_hot_project(
  path,
  path_data = NULL,
  template = hotfun::project_template,
  ...
)
```

Arguments

path	A path. If it exists, it is used. If it does not exist, it is created.
path_data	A path. The directory where the secure data exist. Default is NULL. When supplied, a symbolic link to data folder will be created.
template	Specifies template for starter::create_project(template=). Default is hotfun::project_templat
...	Arguments passed on to starter::create_project
	git Logical indicating whether to create Git repository. Default is TRUE When NA, user will be prompted whether to initialise Git repo.
	renv Logical indicating whether to add renv to a project. Default is TRUE. When NA user is asked interactively for preference.
	symlink Logical indicating whether to place a symbolic link to the location in path_data=. Default is to place the symbolic link if the project is a git repository.
	overwrite Logical indicating whether to overwrite existing files if they exist. Options are TRUE, FALSE, and NA (aka ask interactively). Default is NA
	open Logical indicating whether to open new project in fresh RStudio session

Examples

```
## Not run: \donttest{
# specifying project folder location (folder does not yet exist)
project_path <- fs::path(tempdir(), "My Project Folder")

# creating folder where secure data would be stored (typically will be a network drive)
secure_data_path <- fs::path(tempdir(), "secure_data")
dir.create(secure_data_path)

# creating new project folder
create_hot_project(project_path, path_data = secure_data_path)
}
## End(Not run)
```

egfr_mdrd *Calculate eGFR*

Description

Calculate eGFR

Usage

```
egfr_mdrd(creatinine, age, female, aa, label = "eGFR, mL/min/1.73m²")
```

Arguments

creatinine	serum creatinine level in mg/dL
age	patient age
female	logical indicating whether patient is female
aa	logical indicating whether patient is African-American
label	label that will be applied to result, e.g. `attr("label", 'eGFR, mL/min/1.73m\U00B2')`

Value

numeric vector

Examples

```
egfr_mdrd(creatinine = 1.2, age = 60, female = TRUE, aa = TRUE)
```

get_mode *Calculates the mode(s) of a set of values*

Description

This function calculates the most common value(s) of a given set

Usage

```
get_mode(x, moden = 1, quiet = FALSE)
```

Arguments

x	A variable or vector (numeric, character or factor)
moden	If there are multiple modes, which mode to use. The default is the first mode.
quiet	By default, messages are printed if multiple modes are selected. To hide these messages, set `quiet` to `TRUE`

Value

A vector of length 1 containing the mode

Examples

```
get_mode(trial$stage)
get_mode(trial$trt)
get_mode(trial$response)
get_mode(trial$grade)
```

---

list_labels *Get variable labels and store in named list*

Description

Get variable labels and store in named list

Usage

```
list_labels(data)
```

Arguments

data	Data frame

Author(s)

Daniel D. Sjoberg

Examples

```
list_labels(trial)
```

---

project_template *H.O.T. project template*

Description

The `project_template` object defines the contents of the H.O.T. project template used in `create_hot_project()` and `use_hot_file()`.

Usage

```
project_template
```

Format

A quoted list defining the H.O.T. project template. Each item of the list identifies one script or document that appears in the project template.

See Also

`create_hot_project()`

`use_hot_file()`

Examples

```
## Not run:
create_hot_project(
  path = file.path(tempdir(), "Sjoberg New Project"),
  template = hotfun::project_template
)

## End(Not run)
```

`rm_logs` *Deletes log files created by Rscript on the Linux cluster*

Description

When an R script is submitted to the server, Linux generates a log file named `myRscript.R.o######`. This program searches a folder for files named like this and removes/deletes them.

Usage

```
rm_logs(path = here::here(), recursive = FALSE)
```

Arguments

`path`	folder location to search for log files
`recursive`	logical. should log files in subdirectories also be deleted?

set_derived_variables *Apply variable labels to data frame*

Description

Takes labels from the Derived Variables excel file and applies them to the passed data frame. The function is meant to be used in the pipe.

Usage

```
set_derived_variables(data, path, sheet = NULL, drop = TRUE)
```

Arguments

data	Data frame
path	Path to Derived Variables xls/xlsx file
sheet	Sheet to read. Either a string (the name of a sheet), or an integer (the position of the sheet). Ignored if the sheet is specified via range. If neither argument specifies the sheet, defaults to the first sheet.
drop	Logical indicating whether to drop unlabelled variables

Author(s)

Daniel D. Sjoberg

Examples

```
## Not run:
\donttest{
trial %>%
  set_derived_variables("derived_variables_sjoberg.xlsx")
}

## End(Not run)
```

tbl_propdiff *Calculating unadjusted and adjusted differences in rates*

Description

This function calculates the unadjusted or adjusted difference in rates with confidence interval.

Usage

```
tbl_propdiff(
  data,
  y,
  x,
  formula = "{y} ~ {x}",
  label = NULL,
  statistic = "{n} ({p}%)",
  method = c("chisq", "exact", "boot_sd", "boot_centile"),
  conf.level = 0.95,
  bootstrapn = ifelse(method == "boot_centile", 2000, 200),
  estimate_fun = style_sigfig,
  pvalue_fun = style_pvalue
)
```

Arguments

data	A data frame
y	vector of binary outcome variables. Outcome variables can be numeric, character or factor, but must have two and only two non-missing levels
x	string indicating the binary stratifying variable. The stratifying variable can be numeric, character or factor, but must have two and only two non-missing levels
formula	By default, "{y} ~ {x}". To include covariates for an adjusted risk difference, add covariate names to the formula, e.g. "{y} ~ {x} + age"
label	List of formulas specifying variables labels, If a variable's label is not specified here, the label attribute (attr(data$high_grade, "label")) is used. If attribute label is NULL, the variable name will be used.
statistic	Statistics to display for each group. Default "{n} ({p}%)"
method	The method for calculating p-values and confidence intervals around the difference in rates. The options are "chisq", "exact", "boot_centile", and "boot_sd". See below for details. Default method is "chisq".
conf.level	Confidence level of the returned confidence interval. Must be a single number between 0 and 1. The default is a 95% confidence interval.
bootstrapn	The number of bootstrap resamples to use. The default is 2000 for "boot_centile" and 200 for "boot_sd"
estimate_fun	Function to round and format estimates. By default style_sigfig, but can take any formatting function
pvalue_fun	Function to round and format p-value. By default style_pvalue, but can take any formatting function

Value

A tbl_propdiff object, with sub-class "gtsummary"

Methods

- The `chisq` option returns a p-value from the `prop.test` function and a confidence interval for the unadjusted difference in proportions based on the normal approximation.
- The `exact` option returns a p-value from the `fisher.test` function. The confidence interval returned by this option is the same as the confidence interval returned by the `chisq` option and is based on the normal approximation.
- The `boot_centile` option calculates the adjusted difference between groups in all bootstrap samples (the default for this method is 2000 resamples) and generates the confidence intervals from the distribution of these differences. For the default, a 95% confidence interval, the 2.5 and 97.5 centiles are used. The p-value presented is from a logistic regression model.
- The `boot_sd` option calculates the adjusted difference between groups in all bootstrap samples (the default for this method is 200 resamples). The mean and standard deviation of the adjusted difference across all resamples are calculated. The standard deviation is then used as the standard error to calculate the confidence interval based on the true adjusted difference. The p-value presented is from a logistic regression model.

Examples

```
tbl_propdiff(
  data = trial,
  y = "response",
  x = "trt"
)

tbl_propdiff(
  data = trial,
  y = "response",
  x = "trt",
  formula = "{y} ~ {x} + age + stage",
  method = "boot_sd",
  bootstrapn = 25
)
```

---

trial *Results from a simulated study of two chemotherapy agents: Drug A and Drug B*

Description

A dataset containing the baseline characteristics of 200 patients who received Drug A or Drug B. Dataset also contains the outcome of tumor response to the treatment.

Usage

```
trial
```

Format

A data frame with 200 rows–one row per patient

trt Chemotherapy Treatment

age Age, yrs

marker Marker Level, ng/mL

stage T Stage

grade Grade

response Tumor Response

death Patient Died

ttdeath Months to Death/Censor

---

use_hot_file *Write a template file*

Description

Rather than using `create_hot_project()` to start a new project folder, you may use `use_hot_file()` to write a single file from any project template. The functions `use_hot_gitignore()` and `use_hot_readme()` are shortcuts for `use_hot_file("gitignore")` and `use_hot_file("readme")`.

Usage

```
use_hot_file(
  name = NULL,
  filename = NULL,
  template = hotfun::project_template,
  open = interactive()
)

use_hot_gitignore(filename = NULL, template = NULL)

use_hot_readme(filename = NULL, template = NULL)
```

Arguments

`name`	Name of file to write. Not sure of the files available to you? Run the function without specifying a name, and all files available within the template will print.
`filename`	Optional argument to specify the name of the file to be written. Paths/filename is relative to project base (e.g. `here::here()`)
`template`	A project template. See vignette for details.
`open`	If `TRUE`, opens the new file.

See Also

create_hot_project()

Examples

```
## Not run:
# create gitignore file
use_project_file("gitignore")
use_project_gitignore()

# create README.md file
use_project_file("readme")
use_project_readme()

## End(Not run)
```

---

use_hot_rstudio_prefs *Use H.O.T. RStudio Preferences*

Description

The function wraps `rstudio.prefs::use_rstudio_prefs()` and sets the following preferences in RStudio.

Preference	Value
always_save_history	FALSE
load_workspace	FALSE
margin_column	80
rainbow_parentheses	TRUE
restore_last_project	FALSE
rmd_chunk_output_inline	FALSE
show_hidden_files	TRUE
show_invisibles	TRUE
show_last_dot_value	TRUE
show_line_numbers	TRUE
show_margin	TRUE
save_workspace	never

Usage

```
use_hot_rstudio_prefs()
```

Examples

```
## Not run: \donttest{
use_hot_rstudio_prefs()
}
## End(Not run)
```

Index