Package 'TTU'

April 8, 2021

Title Transfer to Utility Mapping Algorithm Toolkit

Version 0.0.0.9133

Description Tools for developping Transfer To Utility (TTU) mapping algorithms to predict health utility from other health measures. This development version of the TTU package has been made available as part of the process of testing and documenting the package. The tools contained in this development release automate a number of tasks which MODIFY THE DIRECTORY STRUCTURE OF YOUR LOCAL MACHINE. Therefore you should only trial this software if you feel confident that you understand what it does and have created a sandpit area in which you can safely undertake testing. If you have any questions, please contact the authors (matthew.hamilton@orygen.org.au). Some of the documentation for this package has been automatically generated by the ready4fun package and is therefore quite rudimentary. Human edits to improve the quality of documentation will follow in 2021.

License GPL-3 + file LICENSE

URL https://ready4-dev.github.io/TTU/, https://github.com/ready4-dev/TTU, https://ready4-dev.github.io/ready4/

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports assertthat,
boot,
Boruta,
brms,
caret,
cmdstanr (>= 0.3.0.9000),
dataverse (>= 0.3.7),
dplyr,
ggalt,
ggfortify,
ggplot2,
grDevices,
Hmisc,
knitr,
knitrBootstrap,

lifecycle,
lubridate,
magrittr,
MASS,
Matrix,
matrixcalc,
methods,
pacman,
psych,
purrr,
randomForest,
readr,
ready4class (>= 0.0.0.9193),
ready4fun (>= 0.0.0.9289),
ready4show (>= 0.0.0.9019),
ready4use (>= 0.0.0.9122),
rlang,
simstudy,
stats,
stringi,
stringr,
Surrogate,
synthpop,
testthat,
tibble,
tidyr,
tidyselect,
utils,
viridis,
youthvars (>= 0.0.0.9018)

VignetteBuilder knitr

Depends R (>= 2.10)

Remotes stan-dev/cmdstanr,
ready4-dev/ready4show,
ready4-dev/ready4use,
ready4-dev/youthvars,
iqss/dataverse-client-r,
ready4-dev/ready4class,
ready4-dev/ready4fun

R topics documented:

`TTU-package` *TTU: Transfer to Utility Mapping Algorithm Toolkit*

Description

Tools for developping Transfer To Utility (TTU) mapping algorithms to predict health utility from other health measures. This development version of the TTU package has been made available as part of the process of testing and documenting the package. The tools contained in this development release automate a number of tasks which MODIFY THE DIRECTORY STRUCTURE OF YOUR LOCAL MACHINE. Therefore you should only trial this software if you feel confident that you understand what it does and have created a sandpit area in which you can safely undertake testing.

If you have any questions, please contact the authors (matthew.hamilton@orygen.org.au). Some of the documentation for this package has been automatically generated by the ready4fun package and is therefore quite rudimentary. Human edits to improve the quality of documentation will follow in 2021.

Details

To learn more about TTU, start with the vignettes: `browseVignettes(package = "TTU")`

Author(s)

Maintainer: Matthew Hamilton <matthew.hamilton@orygen.org.au> (ORCID)

Authors:

- Caroline Gao <caroline.gao@orygen.org.au> (ORCID)

Other contributors:

- Orygen [copyright holder, funder]
- Headspace [funder]
- National Health and Medical Research Council [funder]

See Also

Useful links:

- https://ready4-dev.github.io/TTU/
- https://github.com/ready4-dev/TTU
- https://ready4-dev.github.io/ready4/

`abbreviations_lup` *Common abbreviations lookup table*

Description

A lookup table for abbreviations commonly used in object names in the TTUpackage.

Usage

```
abbreviations_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 455 rows and 3 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

plural_lgl Plural (a logical vector)

Source

https://doi.org/10.7910/DVN/2Y9VF9

`add_cors_and_utls_to_aqol6d_tbs_ls` *Add correlations and utilities to Assessment of Quality of Life Six Dimension tibbles*

Description

add_cors_and_utls_to_aqol6d_tbs_ls() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add correlations and utilities to assessment of quality of life six dimension tibbles list. Function argument aqol6d_tbs_ls specifies the object to be updated. The function returns Assessment of Quality of Life Six Dimension tibbles (a list).

Usage

```
add_cors_and_utls_to_aqol6d_tbs_ls(
  aqol6d_tbs_ls,
  aqol_scores_pars_ls,
  aqol_items_prpns_tbs_ls,
  temporal_cors_ls,
  prefix_chr,
  aqol_tots_var_nms_chr,
  id_var_nm_1L_chr = "fkClientID"
)
```

Arguments

`aqol6d_tbs_ls`	Assessment of Quality of Life Six Dimension tibbles (a list)
`aqol_scores_pars_ls`	Assessment of Quality of Life scores parameters (a list)
`aqol_items_prpns_tbs_ls`	Assessment of Quality of Life items proportions tibbles (a list)
`temporal_cors_ls`	Temporal correlations (a list)
`prefix_chr`	Prefix (a character vector)
`aqol_tots_var_nms_chr`	Assessment of Quality of Life totals variable names (a character vector)
`id_var_nm_1L_chr`	Identity variable name (a character vector of length one), Default: 'fkClientID'

Value

Assessment of Quality of Life Six Dimension tibbles (a list)

`add_interval_var` *Add interval variable*

Description

add_interval_var() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add interval variable. Function argument data_tb specifies the object to be updated. The function returns Updated data (a tibble).

Usage

```
add_interval_var(
  data_tb,
  id_var_nm_1L_chr = "fkClientID",
  msrmnt_date_var_nm_1L_chr = "d_interview_date",
  time_unit_1L_chr = "days",
  bl_date_var_nm_1L_chr = "bl_date_dtm",
  interval_var_nm_1L_chr = "interval_dbl",
  temp_row_nbr_var_nm_1L_chr = "temp_row_nbr_int",
  drop_bl_date_var_1L_lgl = F
)
```

Arguments

`data_tb` Data (a tibble)

`id_var_nm_1L_chr`
Identity variable name (a character vector of length one), Default: 'fkClientID'

`msrmnt_date_var_nm_1L_chr`
Measurement date variable name (a character vector of length one), Default: 'd_interview_date'

`time_unit_1L_chr`
Time unit (a character vector of length one), Default: 'days'

`bl_date_var_nm_1L_chr`
Baseline date variable name (a character vector of length one), Default: 'bl_date_dtm'

`interval_var_nm_1L_chr`
Interval variable name (a character vector of length one), Default: 'interval_dbl'

`temp_row_nbr_var_nm_1L_chr`
Temporary row number variable name (a character vector of length one), Default: 'temp_row_nbr_int'

`drop_bl_date_var_1L_lgl`
Drop baseline date variable (a logical vector of length one), Default: F

Value

Updated data (a tibble)

add_labels_to_aqol6d_tb *Add labels to Assessment of Quality of Life Six Dimension*

Description

add_labels_to_aqol6d_tb() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add labels to assessment of quality of life six dimension tibble. Function argument aqol6d_tb specifies the object to be updated. The function returns Assessment of Quality of Life Six Dimension (a tibble).

Usage

```
add_labels_to_aqol6d_tb(aqol6d_tb, labels_chr = NA_character_)
```

Arguments

aqol6d_tb	Assessment of Quality of Life Six Dimension (a tibble)
labels_chr	Labels (a character vector), Default: 'NA'

Value

Assessment of Quality of Life Six Dimension (a tibble)

add_uids_to_tbs_ls *Add unique identifiers to tibbles*

Description

add_uids_to_tbs_ls() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add unique identifiers to tibbles list. Function argument tbs_ls specifies the object to be updated. The function returns Tibbles (a list).

Usage

```
add_uids_to_tbs_ls(tbs_ls, prefix_1L_chr, id_var_nm_1L_chr = "fkClientID")
```

Arguments

tbs_ls	Tibbles (a list)
prefix_1L_chr	Prefix (a character vector of length one)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'

Value

Tibbles (a list)

`add_utility_predn_to_ds` *Add utility prediction to dataset*

Description

add_utility_predn_to_ds() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add utility prediction to dataset. Function argument data_tb specifies the object to be updated. The function returns Data (a tibble).

Usage

```
add_utility_predn_to_ds(
  data_tb,
  model_mdl,
  tfmn_1L_chr,
  depnt_var_nm_1L_chr,
  predr_vars_nms_chr = NULL,
  force_min_max_1L_lgl = T,
  utl_min_val_1L_dbl = 0.03,
  impute_1L_lgl = T,
  utl_cls_fn = NULL,
  rmv_tfd_depnt_var_1L_lgl = F
)
```

Arguments

`data_tb`	Data (a tibble)
`model_mdl`	Model (a model)
`tfmn_1L_chr`	Transformation (a character vector of length one)
`depnt_var_nm_1L_chr`	Dependent variable name (a character vector of length one)
`predr_vars_nms_chr`	Predictor variables names (a character vector), Default: NULL
`force_min_max_1L_lgl`	Force minimum maximum (a logical vector of length one), Default: T
`utl_min_val_1L_dbl`	Utility minimum value (a double vector of length one), Default: 0.03
`impute_1L_lgl`	Impute (a logical vector of length one), Default: T
`utl_cls_fn`	Utility class (a function), Default: NULL
`rmv_tfd_depnt_var_1L_lgl`	Remove transformed dependent variable (a logical vector of length one), Default: F

Value

Data (a tibble)

calculate_dpnt_var_tfmn *Calculate dpnt variable transformation*

Description

calculate_dpnt_var_tfmn() is a Calculate function that performs a numeric calculation. Specifically, this function implements an algorithm to calculate dpnt variable transformation. The function returns Transformed dep variable value (a double vector).

Usage

```
calculate_dpnt_var_tfmn(
  dep_var_val_dbl,
  tfmn_1L_chr = "NTF",
  tfmn_is_outp_1L_lgl = F
)
```

Arguments

dep_var_val_dbl	Dep variable value (a double vector)
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
tfmn_is_outp_1L_lgl	Transformation is output (a logical vector of length one), Default: F

Value

Transformed dep variable value (a double vector)

calculate_rmse *Calculate root mean square error*

Description

calculate_rmse() is a Calculate function that performs a numeric calculation. Specifically, this function implements an algorithm to calculate root mean square error. The function returns Root mean square error (a double vector).

Usage

```
calculate_rmse(y_dbl, yhat_dbl)
```

Arguments

y_dbl	Y (a double vector)
yhat_dbl	Yhat (a double vector)

Value

Root mean square error (a double vector)

calculate_rmse_tfmn *Calculate root mean square error transformation*

Description

calculate_rmse_tfmn() is a Calculate function that performs a numeric calculation. Specifically, this function implements an algorithm to calculate root mean square error transformation. The function returns Root mean square error transformation (a double vector).

Usage

```
calculate_rmse_tfmn(y_dbl, yhat_dbl)
```

Arguments

y_dbl	Y (a double vector)
yhat_dbl	Yhat (a double vector)

Value

Root mean square error transformation (a double vector)

fit_clg_log_tfmn *Fit complementary log log transformation*

Description

fit_clg_log_tfmn() is a Fit function that fits a model of a specified type to a dataset Specifically, this function implements an algorithm to fit complementary log log transformation. The function returns Model list (a list of models).

Usage

```
fit_clg_log_tfmn(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w_cloglog",
  predr_vars_nms_chr,
  id_var_nm_1L_chr = "fkClientID",
  backend_1L_chr = getOption("brms.backend", "rstan"),
  iters_1L_int = 4000L,
  seed_1L_int = 1000L
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w_cloglog'
predr_vars_nms_chr	Predictor variables names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
backend_1L_chr	Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
seed_1L_int	Seed (an integer vector of length one), Default: 1000

Value

Model list (a list of models)

---

fit_gsn_log_lnk *Fit gaussian log lnk*

Description

fit_gsn_log_lnk() is a Fit function that fits a model of a specified type to a dataset Specifically, this function implements an algorithm to fit gaussian log lnk. The function returns Model list (a list of models).

Usage

```
fit_gsn_log_lnk(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_vars_nms_chr,
  id_var_nm_1L_chr = "fkClientID",
  backend_1L_chr = getOption("brms.backend", "rstan"),
  iters_1L_int = 4000L,
  seed_1L_int = 1000L
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
predr_vars_nms_chr	Predictor variables names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
backend_1L_chr	Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
seed_1L_int	Seed (an integer vector of length one), Default: 1000

Value

Model list (a list of models)

`fit_ts_model_with_brm` *Fit time series model with bayesian regression model*

Description

fit_ts_model_with_brm() is a Fit function that fits a model of a specified type to a dataset Specifically, this function implements an algorithm to fit time series model with bayesian regression model. The function returns Model list (a list of models).

Usage

```
fit_ts_model_with_brm(
  data_tb,
  depnt_var_nm_1L_chr,
  predr_vars_nms_chr,
  id_var_nm_1L_chr,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  link_1L_chr = "identity",
  iters_1L_int = 4000L,
  seed_1L_int = 1000L
)
```

Arguments

`data_tb`	Data (a tibble)
`depnt_var_nm_1L_chr`	Dependent variable name (a character vector of length one)
`predr_vars_nms_chr`	Predictor variables names (a character vector)
`id_var_nm_1L_chr`	Identity variable name (a character vector of length one)
`backend_1L_chr`	Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")
`link_1L_chr`	Link (a character vector of length one), Default: 'identity'
`iters_1L_int`	Iterations (an integer vector of length one), Default: 4000
`seed_1L_int`	Seed (an integer vector of length one), Default: 1000

Value

Model list (a list of models)

fns_dmt_tb *TTU function documentation table*

Description

Meta-data on each TTU function used to create package documentation

Usage

```
fns_dmt_tb
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 85 rows and 10 columns.

Details

A tibble

fns_chr Functions (a character vector)

title_chr Title (a character vector)

desc_chr Description (a character vector)

details_chr Details (a character vector)

inc_for_main_user_lgl Include for main user (a logical vector)

output_chr Output (a character vector)

example_lgl Example (a logical vector)

args_ls Arguments (a list)

file_nm_chr File name (a character vector)

file_pfx_chr File prefix (a character vector)

Source

https://ready4-dev.github.io/TTU/

fn_type_lup_tb *Function type lookup table*

Description

A lookup table to find descriptions for different types of functions used within the TTU package suite.

Usage

```
fn_type_lup_tb
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 44 rows and 6 columns.

Details

A tibble

fn_type_nm_chr Function type name (a character vector)

fn_type_desc_chr Function type description (a character vector)

first_arg_desc_chr First argument description (a character vector)

second_arg_desc_chr Second argument description (a character vector)

is_generic_lgl Is generic (a logical vector)

is_method_lgl Is method (a logical vector)

Source

https://doi.org/10.7910/DVN/2Y9VF9

---

force_min_max_and_int_cnstrs

Force minimum maximum and integer vector constraints

Description

force_min_max_and_int_cnstrs() is a Force function that checks if a specified local or global environmental condition is met and if not, updates the specified environment to comply with the condition. Specifically, this function implements an algorithm to force minimum maximum and integer vector constraints. The function returns Table (a tibble).

Usage

```
force_min_max_and_int_cnstrs(tbl_tb, var_names_chr, min_max_ls, discrete_lgl)
```

Arguments

`tbl_tb`	Table (a tibble)
`var_names_chr`	Variable names (a character vector)
`min_max_ls`	Minimum maximum (a list)
`discrete_lgl`	Discrete (a logical vector)

Value

Table (a tibble)

force_vec_to_sum_to_int *Force vector to sum to*

Description

force_vec_to_sum_to_int() is a Force function that checks if a specified local or global environmental condition is met and if not, updates the specified environment to comply with the condition. Specifically, this function implements an algorithm to force vector to sum to integer vector. The function returns Vector (an integer vector).

Usage

```
force_vec_to_sum_to_int(vec_int, target_1L_int, item_ranges_dbl_ls)
```

Arguments

vec_int	Vector (an integer vector)
target_1L_int	Target (an integer vector of length one)
item_ranges_dbl_ls	Item ranges (a list of double vectors)

Value

Vector (an integer vector)

get_signft_covars *Get significant covariates*

Description

get_signft_covars() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get significant covariates. Function argument mdls_with_covars_smry_tb specifies the where to look for the required object. The function returns Signt covariates (a character vector).

Usage

```
get_signft_covars(mdls_with_covars_smry_tb, covar_var_nms_chr)
```

Arguments

mdls_with_covars_smry_tb	Models with covariates summary (a tibble)
covar_var_nms_chr	Covariate variable names (a character vector)

Value

Signt covariates (a character vector)

is_TTU_predictors_lup *Is TTU S3 class for candidate predictors lookup table*

Description

Check whether an object is a valid instance of the TTU S3 class for candidate predictors lookup table

Usage

```
is_TTU_predictors_lup(x)
```

Arguments

x	An object of any type

Details

TTU S3 class for candidate predictors lookup table

Value

A logical value, TRUE if a valid instance of the TTU S3 class for candidate predictors lookup table

knit_mdl_rprt *Knit model report*

Description

knit_mdl_rprt() is a Knit function that knits a rmarkdown file Specifically, this function implements an algorithm to knit model report. The function is called for its side effects and does not return a value.

Usage

```
knit_mdl_rprt(knit_pars_ls, path_to_mdl_rprt_tmpl_1L_chr)
```

Arguments

knit_pars_ls	Knit parameters (a list)
path_to_mdl_rprt_tmpl_1L_chr	Path to model report template (a character vector of length one)

`make_aqol6d_adol_pop_tbs_ls` *Make Assessment of Quality of Life Six Dimension adolescent pop tibbles*

Description

make_aqol6d_adol_pop_tbs_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make assessment of quality of life six dimension adolescent pop tibbles list. The function returns Assessment of Quality of Life Six Dimension adolescent pop tibbles (a list).

Usage

```
make_aqol6d_adol_pop_tbs_ls(
  aqol_items_prpns_tbs_ls,
  aqol_scores_pars_ls,
  series_names_chr,
  synth_data_spine_ls,
  temporal_cors_ls,
  id_var_nm_1L_chr = "fkClientID",
 prefix_chr = c(uid = "Participant_", aqol_item = "aqol6d_q", domain_unwtd_pfx_1L_chr
    = "aqol6d_subtotal_c_", domain_wtd_pfx_1L_chr = "aqol6d_subtotal_w_")
)
```

Arguments

`aqol_items_prpns_tbs_ls`	Assessment of Quality of Life items proportions tibbles (a list)
`aqol_scores_pars_ls`	Assessment of Quality of Life scores parameters (a list)
`series_names_chr`	Series names (a character vector)
`synth_data_spine_ls`	Synthetic data spine (a list)
`temporal_cors_ls`	Temporal correlations (a list)
`id_var_nm_1L_chr`	Identity variable name (a character vector of length one), Default: 'fkClientID'
`prefix_chr`	Prefix (a character vector), Default: c(uid = "Participant_", aqol_item = "aqol6d_q", domain_unwtd_pfx_1L_chr = "aqol6d_subtotal_c_", domain_wtd_pfx_1L_chr = "aqol6d_subtotal_w_")

Value

Assessment of Quality of Life Six Dimension adolescent pop tibbles (a list)

make_aqol6d_fns_ls *Make Assessment of Quality of Life Six Dimension functions*

Description

make_aqol6d_fns_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make assessment of quality of life six dimension functions list. The function returns Assessment of Quality of Life Six Dimension disu (a list of functions).

Usage

```
make_aqol6d_fns_ls(domain_items_ls)
```

Arguments

domain_items_ls	Domain items (a list)

Value

Assessment of Quality of Life Six Dimension disu (a list of functions)

make_aqol6d_items_tb *Make Assessment of Quality of Life Six Dimension items*

Description

make_aqol6d_items_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make assessment of quality of life six dimension items tibble. The function returns Assessment of Quality of Life Six Dimension items (a tibble).

Usage

```
make_aqol6d_items_tb(aqol_tb, old_pfx_1L_chr, new_pfx_1L_chr)
```

Arguments

aqol_tb	Assessment of Quality of Life (a tibble)
old_pfx_1L_chr	Old prefix (a character vector of length one)
new_pfx_1L_chr	New prefix (a character vector of length one)

Value

Assessment of Quality of Life Six Dimension items (a tibble)

make_brms_mdl_print_ls

Make bayesian regression models model print list

Description

make_brms_mdl_print_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make bayesian regression models model print list. The function returns Bayesian regression models model print (a list).

Usage

```
make_brms_mdl_print_ls(
  mdl_ls,
  label_stub_1L_chr,
  caption_1L_chr,
  output_type_1L_chr = "PDF",
  digits_1L_dbl = 2,
  big_mark_1L_chr = " "
)
```

Arguments

mdl_ls	Model list (a list of models)
label_stub_1L_chr	Label stub (a character vector of length one)
caption_1L_chr	Caption (a character vector of length one)
output_type_1L_chr	Output type (a character vector of length one), Default: 'PDF'
digits_1L_dbl	Digits (a double vector of length one), Default: 2
big_mark_1L_chr	Big mark (a character vector of length one), Default: ' '

Value

Bayesian regression models model print (a list)

make_brms_mdl_smry_tbl

Make bayesian regression models model summary table

Description

make_brms_mdl_smry_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make bayesian regression models model summary table. The function returns Bayesian regression models model summary (a tibble).

Usage

```
make_brms_mdl_smry_tbl(smry_mdl_ls, grp_1L_chr, popl_1L_chr, fam_1L_chr)
```

Arguments

smry_mdl_ls	Summary (a list of models)
grp_1L_chr	Group (a character vector of length one)
popl_1L_chr	Population (a character vector of length one)
fam_1L_chr	Fam (a character vector of length one)

Value

Bayesian regression models model summary (a tibble)

---

make_complete_prpns_tbs_ls

Make complete proportions tibbles

Description

make_complete_prpns_tbs_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make complete proportions tibbles list. The function returns Complete proportions tibbles (a list).

Usage

```
make_complete_prpns_tbs_ls(
  raw_prpns_tbs_ls,
  question_var_nm_1L_chr = "Question"
)
```

Arguments

raw_prpns_tbs_ls	Raw proportions tibbles (a list)
question_var_nm_1L_chr	Question variable name (a character vector of length one), Default: 'Question'

Value

Complete proportions tibbles (a list)

make_correlated_data_tb *Make correlated data*

Description

make_correlated_data_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make correlated data tibble. The function returns Correlated data (a tibble).

Usage

```
make_correlated_data_tb(synth_data_spine_ls, synth_data_idx_1L_dbl = 1)
```

Arguments

synth_data_spine_ls	Synthetic data spine (a list)
synth_data_idx_1L_dbl	Synthetic data index (a double vector of length one), Default: 1

Value

Correlated data (a tibble)

make_corstars_tbl_xx *Make corstars table*

Description

make_corstars_tbl_xx() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make corstars table output object of multiple potential types. The function is called for its side effects and does not return a value.

Usage

```
make_corstars_tbl_xx(
  x,
  method_chr = c("pearson", "spearman"),
  removeTriangle_chr = c("upper", "lower"),
  result_chr = c("none", "html", "latex")
)
```

Arguments

x	An object
method_chr	Method (a character vector), Default: c("pearson", "spearman")
removeTriangle_chr	RemoveTriangle (a character vector), Default: c("upper", "lower")
result_chr	Result (a character vector), Default: c("none", "html", "latex")

make_dim_sclg_cons_dbl *Make dimension scaling constants*

Description

make_dim_sclg_cons_dbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make dimension scaling constants double vector. The function returns Dimension scaling constants (a double vector).

Usage

```
make_dim_sclg_cons_dbl(domains_chr, dim_sclg_con_lup_tb)
```

Arguments

domains_chr	Domains (a character vector)
dim_sclg_con_lup_tb	Dimension scaling constant lookup table (a tibble)

Value

Dimension scaling constants (a double vector)

make_domain_items_ls *Make domain items*

Description

make_domain_items_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make domain items list. The function returns Domain items (a list).

Usage

```
make_domain_items_ls(domain_qs_lup_tb, item_pfx_1L_chr)
```

Arguments

domain_qs_lup_tb	Domain questions lookup table (a tibble)
item_pfx_1L_chr	Item prefix (a character vector of length one)

Value

Domain items (a list)

make_fake_ts_data *Make fake time series data*

Description

make_fake_ts_data() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make fake time series data. The function returns Fk data (a tibble).

Usage

```
make_fake_ts_data(outp_smry_ls)
```

Arguments

outp_smry_ls	Output summary (a list)

Value

Fk data (a tibble)

make_folds_ls *Make folds*

Description

make_folds_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make folds list. The function returns Folds (a list).

Usage

```
make_folds_ls(
  data_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  folds_1L_int = 10L
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
folds_1L_int	Folds (an integer vector of length one), Default: 10

Value

Folds (a list)

`make_knit_pars_ls` *Make knit parameters*

Description

make_knit_pars_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make knit parameters list. The function returns Knit parameters (a list).

Usage

```
make_knit_pars_ls(
  rltv_path_to_data_dir_1L_chr,
  mdl_types_chr,
  predr_vars_nms_ls,
  output_type_1L_chr = "HTML",
  mdl_types_lup = NULL,
  plt_types_lup = NULL,
  plt_types_chr = NA_character_,
  section_type_1L_chr = "#"
)
```

Arguments

`rltv_path_to_data_dir_1L_chr`
: Relative path to data directory (a character vector of length one)

`mdl_types_chr`
: Model types (a character vector)

`predr_vars_nms_ls`
: Predictor variables names (a list)

`output_type_1L_chr`
: Output type (a character vector of length one), Default: 'HTML'

`mdl_types_lup`
: Model types (a lookup table), Default: NULL

`plt_types_lup`
: Plot types (a lookup table), Default: NULL

`plt_types_chr`
: Plot types (a character vector), Default: 'NA'

`section_type_1L_chr`
: Section type (a character vector of length one), Default: '#'

Value

Knit parameters (a list)

make_make_item_wrst_wts_ls_ls *Make make item worst weights*

Description

make_make_item_wrst_wts_ls_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make make item worst weights list list. The function returns Make item worst weights (a list of lists).

Usage

```
make_make_item_wrst_wts_ls_ls(domain_items_ls, itm_wrst_wghts_lup_tb)
```

Arguments

domain_items_ls	Domain items (a list)
itm_wrst_wghts_lup_tb	Item worst wghts lookup table (a tibble)

Value

Make item worst weights (a list of lists)

make_mdl *Make*

Description

make_mdl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model. The function returns Model (a model).

Usage

```
make_mdl(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  covar_var_nms_chr = NA_character_,
  mdl_type_1L_chr = "OLS_NTF",
  mdl_types_lup = NULL,
  control_1L_chr = NA_character_,
  start_1L_chr = NULL
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
mdl_type_1L_chr	Model type (a character vector of length one), Default: 'OLS_NTF'
mdl_types_lup	Model types (a lookup table), Default: NULL
control_1L_chr	Control (a character vector of length one), Default: 'NA'
start_1L_chr	Start (a character vector of length one), Default: NULL

Value

Model (a model)

---

make_mdl_nms_ls *Make model names*

Description

make_mdl_nms_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model names list. The function returns Model names (a list).

Usage

```
make_mdl_nms_ls(predr_vars_nms_ls, mdl_types_chr)
```

Arguments

predr_vars_nms_ls	Predictor variables names (a list)
mdl_types_chr	Model types (a character vector)

Value

Model names (a list)

make_mdl_smry_elmt_tbl

Make model summary element table

Description

make_mdl_smry_elmt_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model summary element table. The function returns Model element sum (a tibble).

Usage

```
make_mdl_smry_elmt_tbl(mat, ctg_chr)
```

Arguments

mat	Matrix (a matrix)
ctg_chr	Category categories (a character vector)

Value

Model element sum (a tibble)

make_new_TTU_predictors_lup

Make new TTU S3 class for candidate predictors lookup table

Description

Create a new unvalidated instance of the TTU S3 class for candidate predictors lookup table

Usage

```
make_new_TTU_predictors_lup(x)
```

Arguments

x	A prototype for the TTU S3 class for candidate predictors lookup table

Details

TTU S3 class for candidate predictors lookup table

Value

An unvalidated instance of the TTU S3 class for candidate predictors lookup table

make_pdef_cor_mat_mat *Make positive definite correlation matrix*

Description

make_pdef_cor_mat_mat() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make positive definite correlation matrix matrix. The function returns Positive definite correlation (a matrix).

Usage

```
make_pdef_cor_mat_mat(lower_diag_mat)
```

Arguments

lower_diag_mat	Lower diag (a matrix)

Value

Positive definite correlation (a matrix)

make_predn_ds_with_one_predr *Make prediction dataset with one predictor*

Description

make_predn_ds_with_one_predr() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make prediction dataset with one predictor. The function returns Prediction dataset (a tibble).

Usage

```
make_predn_ds_with_one_predr(
  model_mdl,
  depnt_var_nm_1L_chr = "utl_total_w",
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  predr_vals_dbl,
  predn_type_1L_chr = NULL
)
```

Arguments

model_mdl	Model (a model)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'

predr_var_nm_1L_chr
: Predictor variable name (a character vector of length one)

predr_vals_dbl
: Predictor values (a double vector)

predn_type_1L_chr
: Prediction type (a character vector of length one), Default: NULL

Value

Prediction dataset (a tibble)

`make_predr_vals` *Make predictor values*

Description

make_predr_vals() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make predictor values. The function returns Predictor values (a double vector).

Usage

```
make_predr_vals(predr_var_nm_1L_chr, candidate_predrs_lup = NULL)
```

Arguments

predr_var_nm_1L_chr
: Predictor variable name (a character vector of length one)

candidate_predrs_lup
: Candidate predictors (a lookup table), Default: NULL

Value

Predictor values (a double vector)

`make_predr_vars_nms_ls`
Make predictor variables names

Description

make_predr_vars_nms_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make predictor variables names list. The function returns Predictor variables names (a list).

Usage

```
make_predr_vars_nms_ls(main_predrs_chr, covars_ls)
```

Arguments

main_predrs_chr	Main predictors (a character vector)
covars_ls	Covariates (a list)

Value

Predictor variables names (a list)

---

make_prefd_mdls_vec *Make preferred models vector*

Description

make_prefd_mdls_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make preferred models vector. The function returns Preferred models (a character vector).

Usage

```
make_prefd_mdls_vec(
  smry_of_sngl_predr_mdls_tb,
  choose_from_pfx_chr = c("GLM", "OLS"),
  mdl_types_lup = NULL
)
```

Arguments

smry_of_sngl_predr_mdls_tb	Summary of single predictor models (a tibble)
choose_from_pfx_chr	Choose from prefix (a character vector), Default: c("GLM", "OLS")
mdl_types_lup	Model types (a lookup table), Default: NULL

Value

Preferred models (a character vector)

`make_pt_TTU_predictors_lup` *Make prototype TTU S3 class for candidate predictors lookup table*

Description

Create a new prototype for the TTU S3 class for candidate predictors lookup table

Usage

```
make_pt_TTU_predictors_lup(
  short_name_chr = character(0),
  long_name_chr = character(0),
  min_val_dbl = numeric(0),
  max_val_dbl = numeric(0),
  class_chr = character(0),
  increment_dbl = numeric(0),
  class_fn_chr = character(0),
  mdl_scaling_dbl = numeric(0),
  covariate_lgl = logical(0)
)
```

Arguments

`short_name_chr`	Short name (a character vector), Default: character(0)
`long_name_chr`	Long name (a character vector), Default: character(0)
`min_val_dbl`	Minimum value (a double vector), Default: numeric(0)
`max_val_dbl`	Maximum value (a double vector), Default: numeric(0)
`class_chr`	Class (a character vector), Default: character(0)
`increment_dbl`	Increment (a double vector), Default: numeric(0)
`class_fn_chr`	Class function (a character vector), Default: character(0)
`mdl_scaling_dbl`	Model scaling (a double vector), Default: numeric(0)
`covariate_lgl`	Covariate (a logical vector), Default: logical(0)

Details

TTU S3 class for candidate predictors lookup table

Value

A prototype for TTU S3 class for candidate predictors lookup table

`make_shareable_mdl` *Make shareable*

Description

make_shareable_mdl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make shareable model. The function returns Model (a model).

Usage

```
make_shareable_mdl(
  data_tb,
  mdl_smry_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  id_var_nm_1L_chr = "fkClientID",
  tfmn_1L_chr = "CLL",
  mdl_type_1L_chr = "OLS_CLL",
  mdl_types_lup = NULL,
  control_1L_chr = NA_character_,
  start_1L_chr = NA_character_,
  seed_1L_int = 12345L
)
```

Arguments

`data_tb`	Data (a tibble)
`mdl_smry_tb`	Model summary (a tibble)
`depnt_var_nm_1L_chr`	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
`id_var_nm_1L_chr`	Identity variable name (a character vector of length one), Default: 'fkClientID'
`tfmn_1L_chr`	Transformation (a character vector of length one), Default: 'CLL'
`mdl_type_1L_chr`	Model type (a character vector of length one), Default: 'OLS_CLL'
`mdl_types_lup`	Model types (a lookup table), Default: NULL
`control_1L_chr`	Control (a character vector of length one), Default: 'NA'
`start_1L_chr`	Start (a character vector of length one), Default: 'NA'
`seed_1L_int`	Seed (an integer vector of length one), Default: 12345

Value

Model (a model)

make_smry_of_brm_mdl *Make summary of bayesian regression model*

Description

make_smry_of_brm_mdl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make summary of bayesian regression model model. The function returns Summary of bayesian regression model model (a tibble).

Usage

```
make_smry_of_brm_mdl(
  mdl_ls,
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_vars_nms_chr,
  fn = calculate_rmse,
  mdl_nm_1L_chr = NA_character_,
  seed_1L_dbl = 23456
)
```

Arguments

mdl_ls	Model list (a list of models)
data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
predr_vars_nms_chr	Predictor variables names (a character vector)
fn	Function (a function), Default: calculate_rmse
mdl_nm_1L_chr	Model name (a character vector of length one), Default: 'NA'
seed_1L_dbl	Seed (a double vector of length one), Default: 23456

Value

Summary of bayesian regression model model (a tibble)

make_smry_of_mdl_outp *Make summary of model output*

Description

make_smry_of_mdl_outp() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make summary of model output. The function returns Summary of one predictor model (a tibble).

Usage

```
make_smry_of_mdl_outp(
  data_tb,
  model_mdl,
  folds_1L_int = 10,
  depnt_var_nm_1L_chr = "utl_total_w",
  start_1L_chr = NULL,
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  covar_var_nms_chr = NA_character_,
  mdl_type_1L_chr = "OLS_NTF",
  mdl_types_lup = NULL,
  predn_type_1L_chr = NULL
)
```

Arguments

data_tb	Data (a tibble)
model_mdl	Model (a model)
folds_1L_int	Folds (an integer vector of length one), Default: 10
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
start_1L_chr	Start (a character vector of length one), Default: NULL
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
mdl_type_1L_chr	Model type (a character vector of length one), Default: 'OLS_NTF'
mdl_types_lup	Model types (a lookup table), Default: NULL
predn_type_1L_chr	Prediction type (a character vector of length one), Default: NULL

Value

Summary of one predictor model (a tibble)

---

make_smry_of_ts_mdl_outp

Make summary of time series model output

Description

make_smry_of_ts_mdl_outp() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make summary of time series model output. The function returns Summary of time series (a list of models).

Usage

```
make_smry_of_ts_mdl_outp(
  data_tb,
  fn,
  predr_vars_nms_chr,
  mdl_nm_1L_chr,
  path_to_write_to_1L_chr = NA_character_,
  depnt_var_nm_1L_chr = "utl_total_w",
  id_var_nm_1L_chr = "fkClientID",
  round_var_nm_1L_chr = "round",
  round_bl_val_1L_chr = "Baseline",
  predictors_lup,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  iters_1L_int = 4000L,
  seed_1L_int = 1000L
)
```

Arguments

data_tb	Data (a tibble)
fn	Function (a function)
predr_vars_nms_chr	Predictor variables names (a character vector)
mdl_nm_1L_chr	Model name (a character vector of length one)
path_to_write_to_1L_chr	Path to write to (a character vector of length one), Default: 'NA'
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
round_bl_val_1L_chr	Round baseline value (a character vector of length one), Default: 'Baseline'
predictors_lup	Predictors (a lookup table)
backend_1L_chr	Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
seed_1L_int	Seed (an integer vector of length one), Default: 1000

Value

Summary of time series (a list of models)

make_synth_series_tbs_ls

Make synthetic series tibbles

Description

make_synth_series_tbs_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make synthetic series tibbles list. The function returns Synthetic series tibbles (a list).

Usage

```
make_synth_series_tbs_ls(synth_data_spine_ls, series_names_chr)
```

Arguments

synth_data_spine_ls	Synthetic data spine (a list)
series_names_chr	Series names (a character vector)

Value

Synthetic series tibbles (a list)

make_unique_ls_elmt_idx_int

Make unique list element index

Description

make_unique_ls_elmt_idx_int() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make unique list element index integer vector. The function returns Unique list element index (an integer vector).

Usage

```
make_unique_ls_elmt_idx_int(data_ls)
```

Arguments

data_ls	Data (a list)

Value

Unique list element index (an integer vector)

make_vec_with_sum_of_int_val *Make vector with sum of integer vector value*

Description

make_vec_with_sum_of_int_val() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make vector with sum of integer vector value. The function returns Vector (an integer vector).

Usage

```
make_vec_with_sum_of_int_val(target_int, start_int, end_int, length_int)
```

Arguments

target_int	Target (an integer vector)
start_int	Start (an integer vector)
end_int	End (an integer vector)
length_int	Length (an integer vector)

Value

Vector (an integer vector)

mdl_types_lup *Model types lookup table*

Description

A lookup table of abbreviations to describe the different model types supported by TTU functions

Usage

```
mdl_types_lup
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 12 rows and 9 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

control_chr Control (a character vector)

family_chr Family (a character vector)

fn_chr Function (a character vector)

start_chr Start (a character vector)

predn_type_chr Prediction type (a character vector)

tfmn_chr Transformation (a character vector)

tfmn_for_bnml_lgl Transformation for binomial (a logical vector)

`plot_auto_lm` *Plot automatic linear model*

Description

plot_auto_lm() is a Plot function that plots data Specifically, this function implements an algorithm to plot automatic linear model. The function is called for its side effects and does not return a value.

Usage

```
plot_auto_lm(mdl, which_dbl = 1:6, ncol_1L_int = 3L, label_size_1L_int = 3)
```

Arguments

`mdl`	Model (a model)
`which_dbl`	Which (a double vector), Default: 1:6
`ncol_1L_int`	Ncol (an integer vector of length one), Default: 3
`label_size_1L_int`	Label size (an integer vector of length one), Default: 3

`plot_lnr_cmprsn` *Plot linear comparison*

Description

plot_lnr_cmprsn() is a Plot function that plots data Specifically, this function implements an algorithm to plot linear comparison. The function is called for its side effects and does not return a value.

Usage

```
plot_lnr_cmprsn(
  data_tb,
  predn_ds_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  predr_var_nm_1L_chr,
  depnt_var_desc_1L_chr = "AQoL-6D utility score",
  predr_var_desc_1L_chr
)
```

Arguments

data_tb	Data (a tibble)
predn_ds_tb	Prediction dataset (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'AQoL-6D utility score'
predr_var_desc_1L_chr	Predictor variable description (a character vector of length one)

---

plot_obsd_predd_dnst *Plot observed predicted density*

Description

plot_obsd_predd_dnst() is a Plot function that plots data Specifically, this function implements an algorithm to plot observed predicted density. The function is called for its side effects and does not return a value.

Usage

```
plot_obsd_predd_dnst(
  tfd_data_tb,
  depnt_var_desc_1L_chr = "AQoL-6D utility score",
  predd_val_var_nm_1L_chr = "Predicted"
)
```

Arguments

tfd_data_tb	Transformed data (a tibble)
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'AQoL-6D utility score'
predd_val_var_nm_1L_chr	Predicted value variable name (a character vector of length one), Default: 'Predicted'

plot_obsd_predd_sctr_cmprsn *Plot observed predicted scatter comparison*

Description

plot_obsd_predd_sctr_cmprsn() is a Plot function that plots data Specifically, this function implements an algorithm to plot observed predicted scatter comparison. The function is called for its side effects and does not return a value.

Usage

```
plot_obsd_predd_sctr_cmprsn(
  tfd_data_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  depnt_var_desc_1L_chr = "AQoL-6D utility score",
  round_var_nm_1L_chr = "round",
  args_ls = NULL,
  predd_val_var_nm_1L_chr = "Predicted"
)
```

Arguments

tfd_data_tb	Transformed data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'AQoL-6D utility score'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
args_ls	Arguments (a list), Default: NULL
predd_val_var_nm_1L_chr	Predicted value variable name (a character vector of length one), Default: 'Predicted'

plot_sctr_plt_cmprsn *Plot scatter plot comparison*

Description

plot_sctr_plt_cmprsn() is a Plot function that plots data Specifically, this function implements an algorithm to plot scatter plot comparison. The function is called for its side effects and does not return a value.

Usage

```
plot_sctr_plt_cmprsn(
  tfd_data_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  predd_val_var_nm_1L_chr = "Predicted"
)
```

Arguments

tfd_data_tb	Transformed data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
predd_val_var_nm_1L_chr	Predicted value variable name (a character vector of length one), Default: 'Predicted'

---

plt_types_lup *Model plot types lookup table*

Description

A lookup table of abbreviations to describe the different model plot types supported by TTU functions

Usage

```
plt_types_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4 rows and 2 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

predictors_lup *Predictors lookup table*

Description

A lookup table of the short name and long name of each predictor used in the models included with the youthu package.

Usage

```
predictors_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 7 rows and 9 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

min_val_dbl Minimum value (a double vector)

max_val_dbl Maximum value (a double vector)

class_chr Class (a character vector)

increment_dbl Increment (a double vector)

class_fn_chr Class function (a character vector)

mdl_scaling_dbl Model scaling (a double vector)

covariate_lgl Covariate (a logical vector)

predict_utility *Predict utility*

Description

predict_utility() is a Predict function that makes predictions from data using a specified statistical model. Specifically, this function implements an algorithm to predict utility. The function returns Predicted utility (a double vector).

Usage

```
predict_utility(
  data_tb,
  tfmn_1L_chr = "NTF",
  model_mdl,
  force_min_max_1L_lgl = T,
  utl_min_val_1L_dbl = 0.03,
  impute_1L_lgl = T,
  utl_cls_fn = NULL
)
```

Arguments

data_tb	Data (a tibble)
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
model_mdl	Model (a model)
force_min_max_1L_lgl	Force minimum maximum (a logical vector of length one), Default: T
utl_min_val_1L_dbl	Utility minimum value (a double vector of length one), Default: 0.03
impute_1L_lgl	Impute (a logical vector of length one), Default: T
utl_cls_fn	Utility class (a function), Default: NULL

Value

Predicted utility (a double vector)

---

print_all_plts_for_mdl_set *Print all plots for model set*

Description

print_all_plts_for_mdl_set() is a Print function that prints output to console Specifically, this function implements an algorithm to print all plots for model set. The function is called for its side effects and does not return a value.

Usage

```
print_all_plts_for_mdl_set(output_ls, start_from_1L_int = 0L)
```

Arguments

output_ls	Output (a list)
start_from_1L_int	Start from (an integer vector of length one), Default: 0

---

print_ts_mdl_plts *Print time series model plots*

Description

print_ts_mdl_plts() is a Print function that prints output to console Specifically, this function implements an algorithm to print time series model plots. The function is called for its side effects and does not return a value.

Usage

```
print_ts_mdl_plts(paths_to_plts_chr, title_1L_chr, label_refs_chr, mdl_smry_ls)
```

Arguments

paths_to_plts_chr	Paths to plots (a character vector)
title_1L_chr	Title (a character vector of length one)
label_refs_chr	Label references (a character vector)
mdl_smry_ls	Model summary (a list)

---

prototype_lup *Class prototype lookup table*

Description

Metadata on classes used in readyforwhatsnext suite

Usage

```
prototype_lup
```

Format

An object of class ready4_class_pt_lup (inherits from ready4_class_pt_lup, tbl_df, tbl, data.frame) with 28 rows and 6 columns.

Details

A tibble

type_chr Type (a character vector)

val_chr Value (a character vector)

pt_ns_chr Prototype namespace (a character vector)

fn_to_call_chr Function to call (a character vector)

default_val_chr Default value (a character vector)

old_class_lgl Old class (a logical vector)

---

randomise_changes_in_fct_levs *Randomise changes in factor vector levels*

Description

randomise_changes_in_fct_levs() is a Randomise function that randomly samples from data. Specifically, this function implements an algorithm to randomise changes in factor vector levels. The function is called for its side effects and does not return a value.

Usage

```
randomise_changes_in_fct_levs(vector_fct, prob_unchanged_dbl)
```

Arguments

vector_fct	Vector (a factor vector)
prob_unchanged_dbl	Probability unchanged (a double vector)

---

reorder_cndt_predrs_chr

Reorder candidate predictors

Description

reorder_cndt_predrs_chr() is a Reorder function that reorders an object to conform to a pre-specified schema. Specifically, this function implements an algorithm to reorder candidate predictors character vector. The function is called for its side effects and does not return a value.

Usage

```
reorder_cndt_predrs_chr(
  candidate_predrs_chr,
  data_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  method_1L_chr = "pearson"
)
```

Arguments

candidate_predrs_chr	Candidate predictors (a character vector)
data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
method_1L_chr	Method (a character vector of length one), Default: 'pearson'

Value

Reordered candidate (predictors)

---

reorder_tbs_for_target_cors

Reorder tibbles for target correlations

Description

reorder_tbs_for_target_cors() is a Reorder function that reorders an object to conform to a pre-specified schema. Specifically, this function implements an algorithm to reorder tibbles for target correlations. The function returns Tibbles (a list).

Usage

```
reorder_tbs_for_target_cors(
  tbs_ls,
  cor_dbl,
  cor_var_chr,
  id_var_to_rm_1L_chr = NA_character_
)
```

Arguments

tbs_ls	Tibbles (a list)
cor_dbl	Correlation (a double vector)
cor_var_chr	Correlation variable (a character vector)
id_var_to_rm_1L_chr	Identity variable to rm (a character vector of length one), Default: 'NA'

Value

Tibbles (a list)

---

replace_with_missing_vals

Replace with missing values

Description

replace_with_missing_vals() is a Replace function that edits an object, replacing a specified element with another specified element. Specifically, this function implements an algorithm to replace with missing values. Function argument data_tbl_tb specifies the object to be updated. Argument synth_data_spine_ls provides the object to be updated. The function is called for its side effects and does not return a value.

Usage

```
replace_with_missing_vals(data_tbl_tb, synth_data_spine_ls, idx_int)
```

Arguments

data_tbl_tb	Data table (a tibble)
synth_data_spine_ls	Synthetic data spine (a list)
idx_int	Index (an integer vector)

Value

Synthetic (a table)

rprt_lup *Report types lookup table*

Description

A lookup table of the different report types supported by TTU functions

Usage

```
rprt_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 7 columns.

Details

A tibble

rprt_nms_chr Report names (a character vector)

title_chr Title (a character vector)

paths_to_rmd_dir_1L_chr Paths to Markdown directory (a character vector of length one)

pkg_dirs_chr Package directories (a character vector)

packages_chr Packages (a character vector)

nms_of_rmd_chr Names of Markdown (a character vector)

rltv_paths_to_outpt_yaml_chr Relative paths to outpt yaml (a character vector)

scramble_xx *Scramble*

Description

scramble_xx() is a Scramble function that randomly reorders an object. Specifically, this function implements an algorithm to scramble output object of multiple potential types. The function returns Scrambled vector (an output object of multiple potential types).

Usage

```
scramble_xx(vector_xx)
```

Arguments

`vector_xx`	Vector (an output object of multiple potential types)

Value

Scrambled vector (an output object of multiple potential types)

`transform_data_tb_for_cmprsn` *Transform data tibble for comparison*

Description

transform_data_tb_for_cmprsn() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform data tibble for comparison. Function argument data_tb specifies the object to be updated. Argument model_mdl provides the object to be updated. The function returns Transformed data (a tibble).

Usage

```
transform_data_tb_for_cmprsn(
  data_tb,
  model_mdl,
  depnt_var_nm_1L_chr = "utl_total_w",
  source_data_nm_1L_chr = "Original",
  tf_type_1L_chr = "Predicted",
  predn_type_1L_chr = NULL,
  tfmn_for_bnml_1L_lgl = F,
  family_1L_chr = NA_character_
)
```

Arguments

`data_tb`	Data (a tibble)
`model_mdl`	Model (a model)
`depnt_var_nm_1L_chr`	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
`source_data_nm_1L_chr`	Source data name (a character vector of length one), Default: 'Original'
`tf_type_1L_chr`	Transform type (a character vector of length one), Default: 'Predicted'
`predn_type_1L_chr`	Prediction type (a character vector of length one), Default: NULL
`tfmn_for_bnml_1L_lgl`	Transformation for binomial (a logical vector of length one), Default: F
`family_1L_chr`	Family (a character vector of length one), Default: 'NA'

Value

Transformed data (a tibble)

transform_depnt_var_nm

Transform dependent variable name

Description

transform_depnt_var_nm() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform dependent variable name. Function argument depnt_var_nm_1L_chr specifies the object to be updated. Argument tfmn_1L_chr provides the object to be updated. The function returns Transformed dependent variable name (a character vector of length one).

Usage

```
transform_depnt_var_nm(depnt_var_nm_1L_chr, tfmn_1L_chr = "NTF")
```

Arguments

depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one)
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'

Value

Transformed dependent variable name (a character vector of length one)

transform_depnt_var_nm_for_cll

Transform dependent variable name for complementary log log

Description

transform_depnt_var_nm_for_cll() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform dependent variable name for complementary log log. Function argument depnt_var_nm_1L_chr specifies the object to be updated. The function returns Transformed dependent variable name (a character vector of length one).

Usage

```
transform_depnt_var_nm_for_cll(depnt_var_nm_1L_chr)
```

Arguments

depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one)

Value

Transformed dependent variable name (a character vector of length one)

transform_ds_for_mdlng

Transform dataset for modelling

Description

transform_ds_for_mdlng() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform dataset for modelling. Function argument data_tb specifies the object to be updated. Argument depnt_var_nm_1L_chr provides the object to be updated. The function returns Transformed data (a tibble).

Usage

```
transform_ds_for_mdlng(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_var_nm_1L_chr,
  covar_var_nms_chr = NA_character_
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'

Value

Transformed data (a tibble)

transform_ds_for_tstng

Transform dataset for testing

Description

transform_ds_for_tstng() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform dataset for testing. Function argument data_tb specifies the object to be updated. Argument depnt_var_nm_1L_chr provides the object to be updated. The function returns Transformed data (a tibble).

Usage

```
transform_ds_for_tstng(
  data_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  dep_var_max_val_1L_dbl = 0.999,
  candidate_predrs_chr = NA_character_,
  covar_var_nms_chr = NA_character_,
  round_var_nm_1L_chr = "round",
  round_val_1L_chr = "Baseline",
  remove_all_msng_1L_lgl = F
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
dep_var_max_val_1L_dbl	Dep variable maximum value (a double vector of length one), Default: 0.999
candidate_predrs_chr	Candidate predictors (a character vector), Default: 'NA'
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
round_val_1L_chr	Round value (a character vector of length one), Default: 'Baseline'
remove_all_msng_1L_lgl	Remove all missing (a logical vector of length one), Default: F

Value

Transformed data (a tibble)

---

transform_mdl_vars_with_clss

Transform model variables with classes

Description

transform_mdl_vars_with_clss() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform model variables with classes. Function argument ds_tb specifies the object to be updated. Argument predictors_lup provides the object to be updated. The function returns Transformed dataset (a tibble).

Usage

```
transform_mdl_vars_with_clss(
  ds_tb,
  predictors_lup = NULL,
  prototype_lup = NULL,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  class_fn_1L_chr = "youthvars::youthvars_aqol6d_adol"
)
```

Arguments

ds_tb	Dataset (a tibble)
predictors_lup	Predictors (a lookup table), Default: NULL
prototype_lup	Prototype (a lookup table), Default: NULL
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
class_fn_1L_chr	Class function (a character vector of length one), Default: 'youthvars::youthvars_aqol6d_adol'

Value

Transformed dataset (a tibble)

---

transform_tb_to_mdl_inp

Transform tibble to model input

Description

transform_tb_to_mdl_inp() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform tibble to model input. Function argument data_tb specifies the object to be updated. Argument depnt_var_nm_1L_chr provides the object to be updated. The function returns Transformed for model input (a tibble).

Usage

```
transform_tb_to_mdl_inp(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_vars_nms_chr,
  id_var_nm_1L_chr = "fkClientID",
  round_var_nm_1L_chr = "round",
  round_bl_val_1L_chr = "Baseline",
  drop_all_msng_1L_lgl = T,
  scaling_fctr_dbl = 0.01,
  ungroup_1L_lgl = F,
  add_cll_tfmn_1L_lgl = T
)
```

Arguments

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
predr_vars_nms_chr	Predictor variables names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
round_bl_val_1L_chr	Round baseline value (a character vector of length one), Default: 'Baseline'
drop_all_msng_1L_lgl	Drop all missing (a logical vector of length one), Default: T
scaling_fctr_dbl	Scaling factor (a double vector), Default: 0.01
ungroup_1L_lgl	Ungroup (a logical vector of length one), Default: F
add_cll_tfmn_1L_lgl	Add complementary log log transformation (a logical vector of length one), Default: T

Value

Transformed for model input (a tibble)

---

transform_ts_mdl_data *Transform time series model data*

Description

transform_ts_mdl_data() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform time series model data. Function argument mdl_ls specifies the object to be updated. Argument data_tb provides the object to be updated. The function returns Cnfdl (a list of models).

Usage

```
transform_ts_mdl_data(
  mdl_ls,
  data_tb,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  predr_vars_nms_chr,
  id_var_nm_1L_chr = "fkClientID",
  mdl_nm_1L_chr
)
```

Arguments

mdl_ls	Model list (a list of models)
data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
predr_vars_nms_chr	Predictor variables names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
mdl_nm_1L_chr	Model name (a character vector of length one)

Value

Cnfdl (a list of models)

TTU_predictors_lup *TTU S3 class for candidate predictors lookup table*

Description

Create a new valid instance of the TTU S3 class for candidate predictors lookup table

Usage

```
TTU_predictors_lup(x = make_pt_TTU_predictors_lup())
```

Arguments

x	A prototype for the TTU S3 class for candidate predictors lookup table, Default: make_pt_TTU_predictors_lup()

Details

TTU S3 class for candidate predictors lookup table

Value

A validated instance of the TTU S3 class for candidate predictors lookup table

validate_TTU_predictors_lup *Validate TTU S3 class for candidate predictors lookup table*

Description

Validate an instance of the TTU S3 class for candidate predictors lookup table

Usage

```
validate_TTU_predictors_lup(x)
```

Arguments

x	An unvalidated instance of the TTU S3 class for candidate predictors lookup table

Details

TTU S3 class for candidate predictors lookup table

Value

A prototpe for TTU S3 class for candidate predictors lookup table

write_all_alg_outps *Write all algorithm outputs*

Description

write_all_alg_outps() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write all algorithm outputs. The function returns Output summary (a list).

Usage

```
write_all_alg_outps(
  scored_data_tb,
  path_to_write_to_1L_chr,
  depnt_var_nm_1L_chr = "utl_total_w",
  candidate_predrs_chr,
  candidate_covar_nms_chr,
  id_var_nm_1L_chr = "fkClientID",
  round_var_nm_1L_chr = "round",
  round_bl_val_1L_chr = "Baseline",
  mdl_types_chr = NA_character_,
  prefd_mdl_types_chr = NA_character_,
  choose_from_pfx_chr = c("GLM", "OLS", "BET"),
  prefd_covars_chr = NA_character_,
  seed_1L_int = 12345,
```

```
  folds_1L_int = 10L,
  max_nbr_of_boruta_mdl_runs_int = 300L,
  mdl_types_lup = NULL
)
```

Arguments

scored_data_tb	Scored data (a tibble)
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
candidate_predrs_chr	Candidate predictors (a character vector)
candidate_covar_nms_chr	Candidate covariate names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
round_bl_val_1L_chr	Round baseline value (a character vector of length one), Default: 'Baseline'
mdl_types_chr	Model types (a character vector), Default: 'NA'
prefd_mdl_types_chr	Preferred model types (a character vector), Default: 'NA'
choose_from_pfx_chr	Choose from prefix (a character vector), Default: c("GLM", "OLS", "BET")
prefd_covars_chr	Preferred covariates (a character vector), Default: 'NA'
seed_1L_int	Seed (an integer vector of length one), Default: 12345
folds_1L_int	Folds (an integer vector of length one), Default: 10
max_nbr_of_boruta_mdl_runs_int	Maximum number of boruta model runs (an integer vector), Default: 300
mdl_types_lup	Model types (a lookup table), Default: NULL

Value

Output summary (a list)

`write_box_cox_tfmn` *Write box cox transformation*

Description

write_box_cox_tfmn() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write box cox transformation. The function returns Path to plot (a character vector of length one).

Usage

```
write_box_cox_tfmn(
  data_tb,
  predr_var_nm_1L_chr,
  path_to_write_to_1L_chr,
  depnt_var_nm_1L_chr = "aqol6d_total_w",
  covar_var_nms_chr = NA_character_,
  fl_nm_pfx_1L_chr = "A_RT",
  height_1L_dbl = 6,
  width_1L_dbl = 6,
  start_1L_chr = NULL,
  mdl_types_lup = NULL
)
```

Arguments

data_tb	Data (a tibble)
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'aqol6d_total_w'
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
fl_nm_pfx_1L_chr	File name prefix (a character vector of length one), Default: 'A_RT'
height_1L_dbl	Height (a double vector of length one), Default: 6
width_1L_dbl	Width (a double vector of length one), Default: 6
start_1L_chr	Start (a character vector of length one), Default: NULL
mdl_types_lup	Model types (a lookup table), Default: NULL

Value

Path to plot (a character vector of length one)

---

write_brm_mdl_plt_fl *Write bayesian regression model model plot file*

Description

write_brm_mdl_plt_fl() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write bayesian regression model model plot file. The function returns Path to plot (a character vector of length one).

Usage

```
write_brm_mdl_plt_fl(
  plt_fn = NULL,
  fn_args_ls = NULL,
  path_to_write_to_1L_chr,
  plt_nm_1L_chr,
  grpx_fn = grDevices::png,
  units_1L_chr = "in",
  width_1L_dbl = 6,
  height_1L_dbl = 6,
  rsl_1L_dbl = 300
)
```

Arguments

plt_fn	Plot (a function), Default: NULL
fn_args_ls	Function arguments (a list), Default: NULL
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
plt_nm_1L_chr	Plot name (a character vector of length one)
grpx_fn	Grpx (a function), Default: grDevices::png
units_1L_chr	Units (a character vector of length one), Default: 'in'
width_1L_dbl	Width (a double vector of length one), Default: 6
height_1L_dbl	Height (a double vector of length one), Default: 6
rsl_1L_dbl	Resolution (a double vector of length one), Default: 300

Value

Path to plot (a character vector of length one)

---

write_brm_model_plts *Write bayesian regression model model plots*

Description

write_brm_model_plts() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write bayesian regression model model plots. The function returns Model plots paths (a list).

Usage

```
write_brm_model_plts(
  mdl_ls,
  tfd_data_tb,
  mdl_nm_1L_chr,
  path_to_write_to_1L_chr,
  depnt_var_nm_1L_chr = "utl_total_w",
  depnt_var_desc_1L_chr = "Utility score",
```

```
  round_var_nm_1L_chr = "round",
  tfmn_fn = function(x) {     x },
  units_1L_chr = "in",
  height_dbl = c(rep(6, 2), rep(5, 2)),
  width_dbl = c(rep(6, 2), rep(6, 2)),
  rsl_dbl = rep(300, 4),
  args_ls = NULL,
  seed_1L_dbl = 23456
)
```

Arguments

mdl_ls	Model list (a list of models)
tfd_data_tb	Transformed data (a tibble)
mdl_nm_1L_chr	Model name (a character vector of length one)
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'Utility score'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
tfmn_fn	Transformation (a function), Default: function(x) x
units_1L_chr	Units (a character vector of length one), Default: 'in'
height_dbl	Height (a double vector), Default: c(rep(6, 2), rep(5, 2))
width_dbl	Width (a double vector), Default: c(rep(6, 2), rep(6, 2))
rsl_dbl	Resolution (a double vector), Default: rep(300, 4)
args_ls	Arguments (a list), Default: NULL
seed_1L_dbl	Seed (a double vector of length one), Default: 23456

Value

Model plots paths (a list)

---

write_mdls_with_covars_cmprsn

Write models with covariates comparison

Description

write_mdls_with_covars_cmprsn() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write models with covariates comparison. The function returns Output summary (a list).

Usage

```
write_mdls_with_covars_cmprsn(
  scored_data_tb,
  bl_tb,
  ds_smry_ls,
  mdl_smry_ls,
  output_data_dir_1L_chr,
  seed_1L_int = 1234,
  session_data_ls
)
```

Arguments

scored_data_tb	Scored data (a tibble)
bl_tb	Baseline (a tibble)
ds_smry_ls	Dataset summary (a list)
mdl_smry_ls	Model summary (a list)
output_data_dir_1L_chr	Output data directory (a character vector of length one)
seed_1L_int	Seed (an integer vector of length one), Default: 1234
session_data_ls	Session data (a list)

Value

Output summary (a list)

---

write_mdl_cmprsn *Write model comparison*

Description

write_mdl_cmprsn() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model comparison. The function returns Model comparison (a list).

Usage

```
write_mdl_cmprsn(
  scored_data_tb,
  ds_smry_ls,
  mdl_smry_ls,
  output_data_dir_1L_chr,
  seed_1L_int = 1234
)
```

Arguments

scored_data_tb	Scored data (a tibble)
ds_smry_ls	Dataset summary (a list)
mdl_smry_ls	Model summary (a list)
output_data_dir_1L_chr	Output data directory (a character vector of length one)
seed_1L_int	Seed (an integer vector of length one), Default: 1234

Value

Model comparison (a list)

---

write_mdl_plts *Write model plots*

Description

write_mdl_plts() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model plots. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_mdl_plts(
  data_tb,
  model_mdl,
  mdl_fl_nm_1L_chr = "OLS_NTF",
  depnt_var_nm_1L_chr = "utl_total_w",
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  predr_var_desc_1L_chr,
  predr_vals_dbl,
  covar_var_nms_chr = NA_character_,
  path_to_write_to_1L_chr,
  predn_type_1L_chr = NULL,
  tfmn_for_bnml_1L_lgl = F,
  family_1L_chr = NA_character_,
  plt_idxs_int = 1:5
)
```

Arguments

data_tb	Data (a tibble)
model_mdl	Model (a model)
mdl_fl_nm_1L_chr	Model file name (a character vector of length one), Default: 'OLS_NTF'
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'

tfmn_1L_chr Transformation (a character vector of length one), Default: 'NTF'

predr_var_nm_1L_chr
Predictor variable name (a character vector of length one)

predr_var_desc_1L_chr
Predictor variable description (a character vector of length one)

predr_vals_dbl Predictor values (a double vector)

covar_var_nms_chr
Covariate variable names (a character vector), Default: 'NA'

path_to_write_to_1L_chr
Path to write to (a character vector of length one)

predn_type_1L_chr
Prediction type (a character vector of length one), Default: NULL

tfmn_for_bnml_1L_lgl
Transformation for binomial (a logical vector of length one), Default: F

family_1L_chr Family (a character vector of length one), Default: 'NA'

plt_idxs_int Plot indices (an integer vector), Default: 1:5

---

write_mdl_type_covars_mdls

Write model type covariates models

Description

write_mdl_type_covars_mdls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model type covariates models. The function returns Summary of models with covariates (a tibble).

Usage

```
write_mdl_type_covars_mdls(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predrs_var_nms_chr,
  covar_var_nms_chr,
  mdl_type_1L_chr,
  path_to_write_to_1L_chr,
  new_dir_nm_1L_chr = "D_Covars_Selection",
  fl_nm_pfx_1L_chr = "D_CT",
  mdl_types_lup = NULL,
  start_1L_chr = NA_character_
)
```

Arguments

data_tb Data (a tibble)

depnt_var_nm_1L_chr
Dependent variable name (a character vector of length one), Default: 'utl_total_w'

predrs_var_nms_chr
Predictors variable names (a character vector)

covar_var_nms_chr
: Covariate variable names (a character vector)

mdl_type_1L_chr
: Model type (a character vector of length one)

path_to_write_to_1L_chr
: Path to write to (a character vector of length one)

new_dir_nm_1L_chr
: New directory name (a character vector of length one), Default: 'D_Covars_Selection'

fl_nm_pfx_1L_chr
: File name prefix (a character vector of length one), Default: 'D_CT'

mdl_types_lup
: Model types (a lookup table), Default: NULL

start_1L_chr
: Start (a character vector of length one), Default: 'NA'

Value

Summary of models with covariates (a tibble)

---

write_mdl_type_multi_outps

Write model type multi outputs

Description

write_mdl_type_multi_outps() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model type multi outputs. The function returns Summary of model single predictors (a tibble).

Usage

```
write_mdl_type_multi_outps(
  data_tb,
  folds_1L_int = 10,
  predrs_var_nms_chr,
  covar_var_nms_chr = NA_character_,
  start_1L_chr = NULL,
  mdl_type_1L_chr,
  depnt_var_nm_1L_chr = "utl_total_w",
  path_to_write_to_1L_chr,
  new_dir_nm_1L_chr,
  mdl_types_lup = NULL,
  fl_nm_pfx_1L_chr = "C_PREDR",
  plt_idxs_int = c(3, 5)
)
```

Arguments

data_tb
: Data (a tibble)

folds_1L_int
: Folds (an integer vector of length one), Default: 10

predrs_var_nms_chr
: Predictors variable names (a character vector)

covar_var_nms_chr
: Covariate variable names (a character vector), Default: 'NA'

start_1L_chr
: Start (a character vector of length one), Default: NULL

mdl_type_1L_chr
: Model type (a character vector of length one)

depnt_var_nm_1L_chr
: Dependent variable name (a character vector of length one), Default: 'utl_total_w'

path_to_write_to_1L_chr
: Path to write to (a character vector of length one)

new_dir_nm_1L_chr
: New directory name (a character vector of length one)

mdl_types_lup
: Model types (a lookup table), Default: NULL

fl_nm_pfx_1L_chr
: File name prefix (a character vector of length one), Default: 'C_PREDR'

plt_idxs_int
: Plot indices (an integer vector), Default: c(3, 5)

Value

Summary of model single predictors (a tibble)

---

write_mdl_type_sngl_outps

Write model type single outputs

Description

write_mdl_type_sngl_outps() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model type single outputs. The function returns Summary of one predictor model (a tibble).

Usage

```
write_mdl_type_sngl_outps(
  data_tb,
  folds_1L_int = 10,
  depnt_var_nm_1L_chr = "utl_total_w",
  start_1L_chr = NULL,
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  predr_var_desc_1L_chr,
  predr_vals_dbl,
  covar_var_nms_chr = NA_character_,
  mdl_type_1L_chr = "OLS_NTF",
  mdl_types_lup = NULL,
  path_to_write_to_1L_chr,
  mdl_fl_nm_1L_chr,
  plt_idxs_int = NA_integer_
)
```

Arguments

data_tb	Data (a tibble)
folds_1L_int	Folds (an integer vector of length one), Default: 10
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
start_1L_chr	Start (a character vector of length one), Default: NULL
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
predr_var_desc_1L_chr	Predictor variable description (a character vector of length one)
predr_vals_dbl	Predictor values (a double vector)
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
mdl_type_1L_chr	Model type (a character vector of length one), Default: 'OLS_NTF'
mdl_types_lup	Model types (a lookup table), Default: NULL
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
mdl_fl_nm_1L_chr	Model file name (a character vector of length one)
plt_idxs_int	Plot indices (an integer vector), Default: NA

Value

Summary of one predictor model (a tibble)

---

write_new_outp_dir *Write new output directory*

Description

write_new_outp_dir() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write new output directory. The function returns Output directory (a character vector of length one).

Usage

```
write_new_outp_dir(path_to_write_to_1L_chr, new_dir_nm_1L_chr)
```

Arguments

path_to_write_to_1L_chr	Path to write to (a character vector of length one)
new_dir_nm_1L_chr	New directory name (a character vector of length one)

Value

Output directory (a character vector of length one)

write_predr_and_covars_cmprsn

Write predictor and covariates comparison

Description

write_predr_and_covars_cmprsn() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write predictor and covariates comparison. The function returns Predictor and covariates comparison (a list).

Usage

```
write_predr_and_covars_cmprsn(
  scored_data_tb,
  bl_tb,
  ds_smry_ls,
  mdl_smry_ls,
  output_data_dir_1L_chr,
  seed_1L_int = 1234
)
```

Arguments

scored_data_tb	Scored data (a tibble)
bl_tb	Baseline (a tibble)
ds_smry_ls	Dataset summary (a list)
mdl_smry_ls	Model summary (a list)
output_data_dir_1L_chr	Output data directory (a character vector of length one)
seed_1L_int	Seed (an integer vector of length one), Default: 1234

Value

Predictor and covariates comparison (a list)

write_predr_and_mdl_tstng_results

Write predictor and model testing results

Description

write_predr_and_mdl_tstng_results() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write predictor and model testing results. The function returns Output summary (a list).

Usage

```
write_predr_and_mdl_tstng_results(
  scored_data_tb,
  ds_smry_ls,
  mdl_smry_ls,
  session_data_ls,
  output_data_dir_1L_chr,
  seed_1L_int = 1234
)
```

Arguments

scored_data_tb	Scored data (a tibble)
ds_smry_ls	Dataset summary (a list)
mdl_smry_ls	Model summary (a list)
session_data_ls	Session data (a list)
output_data_dir_1L_chr	Output data directory (a character vector of length one)
seed_1L_int	Seed (an integer vector of length one), Default: 1234

Value

Output summary (a list)

---

write_predr_cmprsn_outps

Write predictor comparison outputs

Description

write_predr_cmprsn_outps() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write predictor comparison outputs. The function returns Confirmed predictors (a tibble).

Usage

```
write_predr_cmprsn_outps(
  data_tb,
  path_to_write_to_1L_chr,
  new_dir_nm_1L_chr = "B_Candidate_Predrs_Cmprsn",
  depnt_var_nm_1L_chr = "utl_total_w",
  candidate_predrs_chr,
  max_nbr_of_boruta_mdl_runs_int = 300L
)
```

Arguments

data_tb	Data (a tibble)
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
new_dir_nm_1L_chr	New directory name (a character vector of length one), Default: 'B_Candidate_Predrs_Cmprsn'
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
candidate_predrs_chr	Candidate predictors (a character vector)
max_nbr_of_boruta_mdl_runs_int	Maximum number of boruta model runs (an integer vector), Default: 300

Value

Confirmed predictors (a tibble)

---

write_results_to_csv *Write results to comma separated variables file*

Description

write_results_to_csv() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write results to comma separated variables file. The function returns Datasets (a tibble).

Usage

```
write_results_to_csv(synth_data_spine_ls, output_dir_1L_chr = ".")
```

Arguments

synth_data_spine_ls	Synthetic data spine (a list)
output_dir_1L_chr	Output directory (a character vector of length one), Default: '.'

Value

Datasets (a tibble)

write_shareable_mdls *Write shareable models*

Description

write_shareable_mdls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write shareable models. The function returns Output summary (a list).

Usage

```
write_shareable_mdls(
  outp_smry_ls,
  new_dir_nm_1L_chr = "G_Shareable",
  shareable_title_detail_1L_chr = ""
)
```

Arguments

outp_smry_ls	Output summary (a list)
new_dir_nm_1L_chr	New directory name (a character vector of length one), Default: 'G_Shareable'
shareable_title_detail_1L_chr	Shareable title detail (a character vector of length one), Default: ''

Value

Output summary (a list)

write_shareable_mdls_to_dv *Write shareable models to dataverse*

Description

write_shareable_mdls_to_dv() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write shareable models to dataverse. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_shareable_mdls_to_dv(
  outp_smry_ls,
  new_dir_nm_1L_chr = "G_Shareable",
  shareable_title_detail_1L_chr = ""
)
```

Arguments

outp_smry_ls	Output summary (a list)
new_dir_nm_1L_chr	New directory name (a character vector of length one), Default: 'G_Shareable'
shareable_title_detail_1L_chr	Shareable title detail (a character vector of length one), Default: ''

---

write_sngl_predr_multi_mdls_outps

Write single predictor multi models outputs

Description

write_sngl_predr_multi_mdls_outps() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write single predictor multi models outputs. The function returns Summary of single predictor models (a tibble).

Usage

```
write_sngl_predr_multi_mdls_outps(
  data_tb,
  mdl_types_chr,
  predr_var_nm_1L_chr,
  predr_var_desc_1L_chr,
  predr_vals_dbl,
  path_to_write_to_1L_chr,
  new_dir_nm_1L_chr = "A_Candidate_Mdls_Cmprsn",
  start_1L_chr = NULL,
  covar_var_nms_chr = NA_character_,
  depnt_var_nm_1L_chr = "utl_total_w",
  folds_1L_int = 10,
  mdl_types_lup = NULL,
  fl_nm_pfx_1L_chr = "A_RT_",
  plt_idxs_int = NA_integer_
)
```

Arguments

data_tb	Data (a tibble)
mdl_types_chr	Model types (a character vector)
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
predr_var_desc_1L_chr	Predictor variable description (a character vector of length one)
predr_vals_dbl	Predictor values (a double vector)
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
new_dir_nm_1L_chr	New directory name (a character vector of length one), Default: 'A_Candidate_Mdls_Cmprsn'

start_1L_chr Start (a character vector of length one), Default: NULL

covar_var_nms_chr
Covariate variable names (a character vector), Default: 'NA'

depnt_var_nm_1L_chr
Dependent variable name (a character vector of length one), Default: 'utl_total_w'

folds_1L_int Folds (an integer vector of length one), Default: 10

mdl_types_lup Model types (a lookup table), Default: NULL

fl_nm_pfx_1L_chr
File name prefix (a character vector of length one), Default: 'A_RT_'

plt_idxs_int Plot indices (an integer vector), Default: NA

Value

Summary of single predictor models (a tibble)

---

write_ts_mdls *Write time series models*

Description

write_ts_mdls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write time series models. The function returns Models summary (a tibble).

Usage

```
write_ts_mdls(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_vars_nms_ls,
  id_var_nm_1L_chr = "fkClientID",
  round_var_nm_1L_chr = "round",
  round_bl_val_1L_chr = "Baseline",
  backend_1L_chr = getOption("brms.backend", "rstan"),
  fn_ls,
  mdl_nms_ls,
  mdl_smry_dir_1L_chr,
  predictors_lup,
  iters_1L_int = 4000L,
  seed_1L_int = 1000L
)
```

Arguments

data_tb Data (a tibble)

depnt_var_nm_1L_chr
Dependent variable name (a character vector of length one), Default: 'utl_total_w'

predr_vars_nms_ls
Predictor variables names (a list)

id_var_nm_1L_chr
: Identity variable name (a character vector of length one), Default: 'fkClientID'

round_var_nm_1L_chr
: Round variable name (a character vector of length one), Default: 'round'

round_bl_val_1L_chr
: Round baseline value (a character vector of length one), Default: 'Baseline'

backend_1L_chr
: Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")

fn_ls
: Function list (a list of functions)

mdl_nms_ls
: Model names (a list)

mdl_smry_dir_1L_chr
: Model summary directory (a character vector of length one)

predictors_lup
: Predictors (a lookup table)

iters_1L_int
: Iterations (an integer vector of length one), Default: 4000

seed_1L_int
: Seed (an integer vector of length one), Default: 1000

Value

Models summary (a tibble)

---

write_ts_mdls_from_alg_outp

Write time series models from algorithm output

Description

write_ts_mdls_from_alg_outp() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write time series models from algorithm output. The function returns Output summary (a list).

Usage

```
write_ts_mdls_from_alg_outp(
  outp_smry_ls,
  fn_ls,
  new_dir_nm_1L_chr = "F_TS_Mdls",
  predictors_lup,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  iters_1L_int = 4000L
)
```

Arguments

outp_smry_ls
: Output summary (a list)

fn_ls
: Function list (a list of functions)

new_dir_nm_1L_chr
: New directory name (a character vector of length one), Default: 'F_TS_Mdls'

predictors_lup
: Predictors (a lookup table)

backend_1L_chr
: Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")

iters_1L_int
: Iterations (an integer vector of length one), Default: 4000

Value

Output summary (a list)

Index