Package 'TTU'

August 5, 2021

Title Transfer to Utility Mapping Algorithm Toolkit

Version 0.0.0.9311

Description Tools for developping, sharing and reporting Transfer To Utility (TTU) mapping algorithms that 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. 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. If you have any questions, please contact the authors (matthew.hamilton@orygen.org.au).

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,
betareg,
boot,
Boruta,
brms,
caret,
cmdstanr (>= 0.4.0.9000),
cowplot,
dataverse (>= 0.3.9),
dplyr,
enrichwith,
eq5d,
faux,
ggalt,
ggfortify,
ggplot2,
here,
Hmisc,
kableExtra,

knitr,
knitrBootstrap,
lifecycle,
lubridate,
magrittr,
MASS,
methods,
pacman,
psych,
purrr,
randomForest,
readr,
ready4class (>= 0.0.0.9199),
ready4fun (>= 0.0.0.9298),
ready4show (>= 0.0.0.9038),
ready4use (>= 0.0.0.9133),
rlang,
rmarkdown,
stats,
stringi,
stringr,
synthpop,
testthat,
tibble,
tidyr,
tidyselect,
utils,
viridis,
xfun,
youthvars (>= 0.0.0.9064)

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, sharing and reporting Transfer To Utility (TTU) mapping algorithms that 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. 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. If you have any questions, please contact the authors (matthew.hamilton@orygen.org.au).

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 503 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_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,
  force_min_max_1L_lgl = T,
  force_new_data_1L_lgl = F,
  impute_1L_lgl = T,
  is_brms_mdl_1L_lgl = T,
  new_data_is_1L_chr = "Predicted",
  predn_type_1L_chr = NULL,
  predr_vars_nms_chr = NULL,
  rmv_tfd_depnt_var_1L_lgl = F,
  sd_dbl = NA_real_,
  utl_cls_fn = NULL,
  utl_min_val_1L_dbl = -1
)
```

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)

`force_min_max_1L_lgl`
Force minimum maximum (a logical vector of length one), Default: T

`force_new_data_1L_lgl`
Force new data (a logical vector of length one), Default: F

`impute_1L_lgl` Impute (a logical vector of length one), Default: T

`is_brms_mdl_1L_lgl`
Is bayesian regression models model (a logical vector of length one), Default: T

`new_data_is_1L_chr`
New data is (a character vector of length one), Default: 'Predicted'

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

`predr_vars_nms_chr`
Predictor variables names (a character vector), Default: NULL

`rmv_tfd_depnt_var_1L_lgl`
Remove transformed dependent variable (a logical vector of length one), Default: F

`sd_dbl` Standard deviation (a double vector), Default: NA

`utl_cls_fn` Utility class (a function), Default: NULL

`utl_min_val_1L_dbl`
Utility minimum value (a double vector of length one), Default: -1

Value

Data (a tibble)

`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 194 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 46 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

get_background_text *Get background text*

Description

get_background_text() 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 background text. Function argument results_ls specifies the where to look for the required object. The function returns Text (a character vector of length one).

Usage

```
get_background_text(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Text (a character vector of length one)

get_cndts_for_mxd_mdls *Get candidates for mxd models*

Description

get_cndts_for_mxd_mdls() 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 candidates for mxd models. Function argument mdl_types_lup specifies the where to look for the required object. The function returns Candidates for mxd models (a lookup table).

Usage

```
get_cndts_for_mxd_mdls(mdl_types_lup = NULL)
```

Arguments

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

Value

Candidates for mxd models (a lookup table)

get_conclusion_text *Get conclusion text*

Description

get_conclusion_text() 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 conclusion text. Function argument results_ls specifies the where to look for the required object. The function returns Text (a character vector of length one).

Usage

```
get_conclusion_text(results_ls)
```

Arguments

results_ls	Results (a list)

Value

Text (a character vector of length one)

get_covars_by_ctg *Get covariates by category categories*

Description

get_covars_by_ctg() 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 covariates by category categories. Function argument results_ls specifies the where to look for the required object. The function returns Covariates by category categories (a list).

Usage

```
get_covars_by_ctg(results_ls, collapse_1L_lgl = F)
```

Arguments

results_ls	Results (a list)
collapse_1L_lgl	Collapse (a logical vector of length one), Default: F

Value

Covariates by category categories (a list)

`get_covar_ctgs` *Get covariate category categoriess*

Description

get_covar_ctgs() 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 covariate category categoriess. Function argument results_ls specifies the where to look for the required object. The function returns Covariate category categoriess (a character vector).

Usage

```
get_covar_ctgs(results_ls, collapse_1L_lgl = T)
```

Arguments

`results_ls`	Results (a list)
`collapse_1L_lgl`	Collapse (a logical vector of length one), Default: T

Value

Covariate category categoriess (a character vector)

`get_hlth_utl_nm` *Get health utility name*

Description

get_hlth_utl_nm() 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 health utility name. Function argument results_ls specifies the where to look for the required object. The function returns Health utility name (a character vector of length one).

Usage

```
get_hlth_utl_nm(results_ls, short_nm_1L_lgl = T)
```

Arguments

`results_ls`	Results (a list)
`short_nm_1L_lgl`	Short name (a logical vector of length one), Default: T

Value

Health utility name (a character vector of length one)

get_hlth_utl_stat *Get health utility statistic*

Description

get_hlth_utl_stat() 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 health utility statistic. Function argument results_ls specifies the where to look for the required object. The function returns Health utility statistic (a character vector of length one).

Usage

```
get_hlth_utl_stat(results_ls, stat_1L_chr = "bl_mean")
```

Arguments

results_ls	Results (a list)
stat_1L_chr	Statistic (a character vector of length one), Default: 'bl_mean'

Value

Health utility statistic (a character vector of length one)

get_link_from_tfmn *Get link from transformation*

Description

get_link_from_tfmn() 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 link from transformation. Function argument tfmn_1L_chr specifies the where to look for the required object. The function returns Link (a character vector of length one).

Usage

```
get_link_from_tfmn(tfmn_1L_chr, is_OLS_1L_lgl = F)
```

Arguments

tfmn_1L_chr	Transformation (a character vector of length one)
is_OLS_1L_lgl	Is OLS (a logical vector of length one), Default: F

Value

Link (a character vector of length one)

`get_lngl_ttu_types` *Get lngl ttu types*

Description

get_lngl_ttu_types() 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 lngl ttu types. Function argument results_ls specifies the where to look for the required object. The function returns Model types (a character vector).

Usage

```
get_lngl_ttu_types(results_ls, collapse_1L_lgl = T)
```

Arguments

`results_ls`	Results (a list)
`collapse_1L_lgl`	Collapse (a logical vector of length one), Default: T

Value

Model types (a character vector)

`get_mdls_with_signft_covars` *Get models with significant covariates*

Description

get_mdls_with_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 models with significant covariates. Function argument outp_smry_ls specifies the where to look for the required object. The function returns Models with significant covariates (a list).

Usage

```
get_mdls_with_signft_covars(outp_smry_ls, params_ls_ls)
```

Arguments

`outp_smry_ls`	Output summary (a list)
`params_ls_ls`	Params (a list of lists)

Value

Models with significant covariates (a list)

get_mdl_cmprsns *Get model comparisons*

Description

get_mdl_cmprsns() 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 model comparisons. Function argument results_ls specifies the where to look for the required object. The function returns Model comparisons (an output object of multiple potential types).

Usage

```
get_mdl_cmprsns(
  results_ls,
  describe_1L_lgl = T,
  mixed_1L_lgl = F,
  as_list_1L_lgl = F
)
```

Arguments

results_ls	Results (a list)
describe_1L_lgl	Describe (a logical vector of length one), Default: T
mixed_1L_lgl	Mixed (a logical vector of length one), Default: F
as_list_1L_lgl	As list (a logical vector of length one), Default: F

Value

Model comparisons (an output object of multiple potential types)

get_mdl_type_from_nm *Get model type from name*

Description

get_mdl_type_from_nm() 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 model type from name. Function argument mdl_nm_1L_chr specifies the where to look for the required object. The function returns Model type (a character vector of length one).

Usage

```
get_mdl_type_from_nm(mdl_nm_1L_chr, mdl_types_lup = NULL)
```

Arguments

mdl_nm_1L_chr	Model name (a character vector of length one)
mdl_types_lup	Model types (a lookup table), Default: NULL

Value

Model type (a character vector of length one)

get_nbr_of_predrs *Get number of predictors*

Description

get_nbr_of_predrs() 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 number of predictors. Function argument results_ls specifies the where to look for the required object. The function returns Number of predictors (an output object of multiple potential types).

Usage

```
get_nbr_of_predrs(results_ls, as_words_1L_lgl = T)
```

Arguments

`results_ls`	Results (a list)
`as_words_1L_lgl`	As words (a logical vector of length one), Default: T

Value

Number of predictors (an output object of multiple potential types)

get_nbr_of_predrs_by_ctg *Get number of predictors by category categories*

Description

get_nbr_of_predrs_by_ctg() 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 number of predictors by category categories. Function argument results_ls specifies the where to look for the required object. The function returns Predictors by category categories (a character vector of length one).

Usage

```
get_nbr_of_predrs_by_ctg(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Predictors by category categories (a character vector of length one)

get_nbr_of_scndry_analyses

Get number of scndry analyses

Description

get_nbr_of_scndry_analyses() 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 number of scndry analyses. Function argument results_ls specifies the where to look for the required object. The function returns Number of scndry analyses length one (an output object of multiple potential types).

Usage

```
get_nbr_of_scndry_analyses(
  results_ls,
  as_words_1L_lgl = T,
  capitalise_1L_lgl = T
)
```

Arguments

`results_ls`	Results (a list)
`as_words_1L_lgl`	As words (a logical vector of length one), Default: T
`capitalise_1L_lgl`	Capitalise (a logical vector of length one), Default: T

Value

Number of scndry analyses length one (an output object of multiple potential types)

get_ordered_sngl_csnl_mdls

Get ordered single csnl models

Description

get_ordered_sngl_csnl_mdls() 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 ordered single csnl models. Function argument results_ls specifies the where to look for the required object. The function returns Ordered single csnl models (a character vector).

Usage

```
get_ordered_sngl_csnl_mdls(results_ls, select_int = NULL, collapse_1L_lgl = F)
```

Arguments

`results_ls`	Results (a list)
`select_int`	Select (an integer vector), Default: NULL
`collapse_1L_lgl`	Collapse (a logical vector of length one), Default: F

Value

Ordered single csnl models (a character vector)

`get_popl_descvs` *Get population descriptives*

Description

get_popl_descvs() 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 population descriptives. Function argument results_ls specifies the where to look for the required object. The function returns Population descriptives (a character vector of length one).

Usage

```
get_popl_descvs(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Population descriptives (a character vector of length one)

`get_predrs_by_ctg` *Get predictors by category categories*

Description

get_predrs_by_ctg() 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 predictors by category categories. Function argument results_ls specifies the where to look for the required object. The function returns Predictors by category categories (a list).

Usage

```
get_predrs_by_ctg(
  results_ls,
  long_desc_1L_lgl = F,
  transform_1L_lgl = F,
  collapse_1L_lgl = F
)
```

Arguments

`results_ls` Results (a list)

`long_desc_1L_lgl` Long description (a logical vector of length one), Default: F

`transform_1L_lgl` Transform (a logical vector of length one), Default: F

`collapse_1L_lgl` Collapse (a logical vector of length one), Default: F

Value

Predictors by category categories (a list)

`get_predr_ctgs` *Get predictor category categoriess*

Description

get_predr_ctgs() 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 predictor category categoriess. Function argument results_ls specifies the where to look for the required object. The function returns Predictor category categoriess (a character vector).

Usage

```
get_predr_ctgs(results_ls, collapse_1L_lgl = T)
```

Arguments

`results_ls` Results (a list)

`collapse_1L_lgl` Collapse (a logical vector of length one), Default: T

Value

Predictor category categoriess (a character vector)

`get_prefd_mdl_predrs` *Get preferred model predictors*

Description

get_prefd_mdl_predrs() 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 preferred model predictors. Function argument results_ls specifies the where to look for the required object. The function returns Predictors (a character vector of length one).

Usage

```
get_prefd_mdl_predrs(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Predictors (a character vector of length one)

`get_scndry_anlys_descs` *Get scndry anlys descriptions*

Description

get_scndry_anlys_descs() 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 scndry anlys descriptions. Function argument results_ls specifies the where to look for the required object. The function returns Scndry anlys descriptions (a character vector).

Usage

```
get_scndry_anlys_descs(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Scndry anlys descriptions (a character vector)

`get_selected_mixed_mdls` *Get selected mixed models*

Description

get_selected_mixed_mdls() 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 selected mixed models. Function argument results_ls specifies the where to look for the required object. The function returns Mixed models (an output object of multiple potential types).

Usage

```
get_selected_mixed_mdls(results_ls, collapse_1L_lgl = T)
```

Arguments

`results_ls`	Results (a list)
`collapse_1L_lgl`	Collapse (a logical vector of length one), Default: T

Value

Mixed models (an output object of multiple potential types)

`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

make_abstract_args_ls *Make abstract arguments*

Description

make_abstract_args_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make abstract arguments list. The function returns Abstract arguments (a list).

Usage

```
make_abstract_args_ls(results_ls, fl_nm_1L_chr = "abstract.txt")
```

Arguments

`results_ls`	Results (a list)
`fl_nm_1L_chr`	File name (a character vector of length one), Default: 'abstract.txt'

Value

Abstract arguments (a list)

make_analysis_ds_smry_ls *Make analysis dataset summary*

Description

make_analysis_ds_smry_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make analysis dataset summary list. The function returns Analysis dataset summary (a list).

Usage

```
make_analysis_ds_smry_ls(ds_descvs_ls, candidate_covar_nms_chr, predictors_lup)
```

Arguments

`ds_descvs_ls`	Dataset descriptives (a list)
`candidate_covar_nms_chr`	Candidate covariate names (a character vector)
`predictors_lup`	Predictors (a lookup table)

Value

Analysis dataset summary (a list)

`make_cmpst_sctr_and_dnsty_plt`
Make cmpst scatter and dnsty

Description

make_cmpst_sctr_and_dnsty_plt() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make cmpst scatter and dnsty plot. The function is called for its side effects and does not return a value.

Usage

```
make_cmpst_sctr_and_dnsty_plt(
  outp_smry_ls,
  output_data_dir_1L_chr,
  predr_var_nms_chr,
  labels_chr = c("A", "B", "C", "D"),
  label_x_1L_dbl = 0.1,
  label_y_1L_dbl = 0.9,
  label_size_1L_dbl = 22
)
```

Arguments

`outp_smry_ls`	Output summary (a list)
`output_data_dir_1L_chr`	Output data directory (a character vector of length one)
`predr_var_nms_chr`	Predictor variable names (a character vector)
`labels_chr`	Labels (a character vector), Default: c("A", "B", "C", "D")
`label_x_1L_dbl`	Label x (a double vector of length one), Default: 0.1
`label_y_1L_dbl`	Label y (a double vector of length one), Default: 0.9
`label_size_1L_dbl`	Label size (a double vector of length one), Default: 22

make_cndt_predr_text *Make candidate predictor text*

Description

make_cndt_predr_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make candidate predictor text. The function returns Text (a character vector of length one).

Usage

```
make_cndt_predr_text(results_ls, type_1L_chr = "description")
```

Arguments

`results_ls`	Results (a list)
`type_1L_chr`	Type (a character vector of length one), Default: 'description'

Value

Text (a character vector of length one)

make_coi_text *Make coi text*

Description

make_coi_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make coi text. The function returns Text (a character vector of length one).

Usage

```
make_coi_text(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Text (a character vector of length one)

make_correlation_text *Make correlation text*

Description

make_correlation_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make correlation text. The function returns Correlation text (a character vector of length one).

Usage

```
make_correlation_text(results_ls)
```

Arguments

results_ls	Results (a list)

Value

Correlation text (a character vector of length one)

make_covariates_text *Make covariates text*

Description

make_covariates_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make covariates text. The function returns Text (a character vector of length one).

Usage

```
make_covariates_text(results_ls)
```

Arguments

results_ls	Results (a list)

Value

Text (a character vector of length one)

`make_covar_ttu_tbl_refs` *Make covariate ttu table references*

Description

make_covar_ttu_tbl_refs() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make covariate ttu table references. The function returns Text (a character vector of length one).

Usage

```
make_covar_ttu_tbl_refs(params_ls)
```

Arguments

`params_ls`	Params (a list)

Value

Text (a character vector of length one)

`make_covar_ttu_tbl_title` *Make covariate ttu table title*

Description

make_covar_ttu_tbl_title() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make covariate ttu table title. The function returns Title (a character vector of length one).

Usage

```
make_covar_ttu_tbl_title(results_ls, ref_1L_int = 1)
```

Arguments

`results_ls`	Results (a list)
`ref_1L_int`	Reference (an integer vector of length one), Default: 1

Value

Title (a character vector of length one)

make_data_availability_text

Make data availability text

Description

make_data_availability_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make data availability text. The function returns Text (a character vector of length one).

Usage

```
make_data_availability_text(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Text (a character vector of length one)

make_dnsty_and_sctr_plt_title

Make dnsty and scatter plot title

Description

make_dnsty_and_sctr_plt_title() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make dnsty and scatter plot title. The function returns Title (a character vector of length one).

Usage

```
make_dnsty_and_sctr_plt_title(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Title (a character vector of length one)

make_ds_descvs_ls *Make dataset descriptives*

Description

make_ds_descvs_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make dataset descriptives list. The function returns Dataset descriptives (a list).

Usage

```
make_ds_descvs_ls(
  candidate_predrs_chr,
  cohort_descv_var_nms_chr,
  dictionary_tb,
  id_var_nm_1L_chr,
  msrmnt_date_var_nm_1L_chr,
  round_var_nm_1L_chr,
  round_vals_chr,
  maui_item_pfx_1L_chr,
  utl_wtd_var_nm_1L_chr = "wtd_utl_dbl",
  utl_unwtd_var_nm_1L_chr = "unwtd_utl_dbl",
  candidate_covar_nms_chr = NULL,
  is_fake_1L_lgl = NULL
)
```

Arguments

`candidate_predrs_chr`
Candidate predictors (a character vector)

`cohort_descv_var_nms_chr`
Cohort descriptive variable names (a character vector)

`dictionary_tb` Dictionary (a tibble)

`id_var_nm_1L_chr`
Identity variable name (a character vector of length one)

`msrmnt_date_var_nm_1L_chr`
Measurement date variable name (a character vector of length one)

`round_var_nm_1L_chr`
Round variable name (a character vector of length one)

`round_vals_chr` Round values (a character vector)

`maui_item_pfx_1L_chr`
Maui item prefix (a character vector of length one)

`utl_wtd_var_nm_1L_chr`
Utility weighted variable name (a character vector of length one), Default: 'wtd_utl_dbl'

`utl_unwtd_var_nm_1L_chr`
Utility unwtd variable name (a character vector of length one), Default: 'unwtd_utl_dbl'

`candidate_covar_nms_chr`
Candidate covariate names (a character vector), Default: NULL

`is_fake_1L_lgl` Is fake (a logical vector of length one), Default: NULL

Value

Dataset descriptives (a list)

make_ds_smry_ls *Make dataset summary*

Description

make_ds_smry_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make dataset summary list. The function returns Dataset summary (a list).

Usage

```
make_ds_smry_ls(
  candidate_predrs_chr,
  candidate_covar_nms_chr,
  depnt_var_nm_1L_chr,
  dictionary_tb,
  id_var_nm_1L_chr,
  round_var_nm_1L_chr,
  round_bl_val_1L_chr,
  predictors_lup
)
```

Arguments

`candidate_predrs_chr`
Candidate predictors (a character vector)

`candidate_covar_nms_chr`
Candidate covariate names (a character vector)

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

`dictionary_tb` Dictionary (a tibble)

`id_var_nm_1L_chr`
Identity variable name (a character vector of length one)

`round_var_nm_1L_chr`
Round variable name (a character vector of length one)

`round_bl_val_1L_chr`
Round baseline value (a character vector of length one)

`predictors_lup` Predictors (a lookup table)

Value

Dataset summary (a list)

`make_eq5d_ds_dict` *Make eq5d dataset dictionary*

Description

make_eq5d_ds_dict() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make eq5d dataset dictionary. The function returns Dictionary (a tibble).

Usage

```
make_eq5d_ds_dict(
  data_tb = make_fake_eq5d_ds(),
  predictors_lup = make_psych_predrs_lup()
)
```

Arguments

`data_tb`	Data (a tibble), Default: make_fake_eq5d_ds()
`predictors_lup`	Predictors (a lookup table), Default: make_psych_predrs_lup()

Value

Dictionary (a tibble)

`make_ethics_text` *Make ethics text*

Description

make_ethics_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ethics text. The function returns Text (a character vector of length one).

Usage

```
make_ethics_text(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Text (a character vector of length one)

make_fake_eq5d_ds *Make fake eq5d dataset*

Description

make_fake_eq5d_ds() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make fake eq5d dataset. The function returns Data (a tibble).

Usage

```
make_fake_eq5d_ds(
  country_1L_chr = "UK",
  version_1L_chr = "5L",
  type_1L_chr = "CW",
  force_attach_1L_lgl = T,
  prop_with_fup_data_1L_dbl = 0.65,
  seed_1L_int = 1234,
  sample_from_1L_int = 10000
)
```

Arguments

`country_1L_chr`	Country (a character vector of length one), Default: 'UK'
`version_1L_chr`	Version (a character vector of length one), Default: '5L'
`type_1L_chr`	Type (a character vector of length one), Default: 'CW'
`force_attach_1L_lgl`	Force attach (a logical vector of length one), Default: T
`prop_with_fup_data_1L_dbl`	Prop with follow-up data (a double vector of length one), Default: 0.65
`seed_1L_int`	Seed (an integer vector of length one), Default: 1234
`sample_from_1L_int`	Sample from (an integer vector of length one), Default: 10000

Value

Data (a tibble)

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, dep_vars_are_NA_1L_lgl = T)
```

Arguments

outp_smry_ls Output summary (a list)

dep_vars_are_NA_1L_lgl
Dep variables are NA (a logical vector of length one), Default: T

Value

Fk data (a tibble)

---

make_funding_text *Make funding text*

Description

make_funding_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make funding text. The function returns Text (a character vector of length one).

Usage

```
make_funding_text(results_ls)
```

Arguments

results_ls Results (a list)

Value

Text (a character vector of length one)

---

make_header_yaml_args_ls *Make header yaml arguments*

Description

make_header_yaml_args_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make header yaml arguments list. The function returns Header yaml arguments (a list).

Usage

```
make_header_yaml_args_ls(
  authors_tb,
  institutes_tb,
  title_1L_chr,
  keywords_chr,
  fl_nm_1L_chr = "header_common.yaml",
  use_fake_data_1L_lgl = F
)
```

Arguments

`authors_tb`	Authors (a tibble)
`institutes_tb`	Institutes (a tibble)
`title_1L_chr`	Title (a character vector of length one)
`keywords_chr`	Keywords (a character vector)
`fl_nm_1L_chr`	File name (a character vector of length one), Default: 'header_common.yaml'
`use_fake_data_1L_lgl`	Use fake data (a logical vector of length one), Default: F

Value

Header yaml arguments (a list)

`make_indpnt_predrs_lngl_tbls_ref`

Make indpnt predictors lngl tables reference

Description

make_indpnt_predrs_lngl_tbls_ref() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make indpnt predictors lngl tables reference. The function returns Text (a character vector of length one).

Usage

```
make_indpnt_predrs_lngl_tbls_ref(params_ls)
```

Arguments

`params_ls`	Params (a list)

Value

Text (a character vector of length one)

`make_indpnt_predrs_lngl_tbl_title`

Make indpnt predictors lngl table title

Description

make_indpnt_predrs_lngl_tbl_title() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make indpnt predictors lngl table title. The function returns Title (a character vector of length one).

Usage

```
make_indpnt_predrs_lngl_tbl_title(results_ls, ref_1L_int = 1)
```

Arguments

results_ls Results (a list)

ref_1L_int Reference (an integer vector of length one), Default: 1

Value

Title (a character vector of length one)

---

make_input_params *Make input params*

Description

make_input_params() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make input params. The function returns Params (a list of lists).

Usage

```
make_input_params(
  ds_tb,
  ds_descvs_ls,
  header_yaml_args_ls,
  maui_params_ls,
  predictors_lup,
  control_ls = NULL,
  dv_ds_nm_and_url_chr = NULL,
  iters_1L_int = 4000L,
  mdl_smry_ls = make_mdl_smry_ls(),
  output_format_ls = make_output_format_ls(),
  path_params_ls = NULL,
  prefd_covars_chr = NULL,
  prefd_mdl_types_chr = NULL,
  prior_ls = NULL,
  seed_1L_int = 12345,
  scndry_anlys_params_ls = NULL,
  write_new_dir_1L_lgl = T
)
```

Arguments

ds_tb Dataset (a tibble)

ds_descvs_ls Dataset descriptives (a list)

header_yaml_args_ls
Header yaml arguments (a list)

maui_params_ls Maui params (a list)

predictors_lup Predictors (a lookup table)

control_ls Control (a list), Default: NULL

dv_ds_nm_and_url_chr
Dataverse dataset name and url (a character vector), Default: NULL

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

`mdl_smry_ls` Model summary (a list), Default: make_mdl_smry_ls()

`output_format_ls`
Output format (a list), Default: make_output_format_ls()

`path_params_ls` Path params (a list), Default: NULL

`prefd_covars_chr`
Preferred covariates (a character vector), Default: NULL

`prefd_mdl_types_chr`
Preferred model types (a character vector), Default: NULL

`prior_ls` Prior (a list), Default: NULL

`seed_1L_int` Seed (an integer vector of length one), Default: 12345

`scndry_anlys_params_ls`
Scndry anlys params (a list), Default: NULL

`write_new_dir_1L_lgl`
Write new directory (a logical vector of length one), Default: T

Value

Params (a list of lists)

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_lngl_ttu_r2_text *Make lngl ttu r2 text*

Description

make_lngl_ttu_r2_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make lngl ttu r2 text. The function returns Text (a character vector of length one).

Usage

```
make_lngl_ttu_r2_text(results_ls, part_int = 1)
```

Arguments

`results_ls`	Results (a list)
`part_int`	Part (an integer vector), Default: 1

Value

Text (a character vector of length one)

make_lngl_ttu_with_covars_text *Make lngl ttu with covariates text*

Description

make_lngl_ttu_with_covars_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make lngl ttu with covariates text. The function returns Text (a character vector of length one).

Usage

```
make_lngl_ttu_with_covars_text(results_ls)
```

Arguments

`results_ls` Results (a list)

Value

Text (a character vector of length one)

`make_maui_params_ls` *Make maui params*

Description

make_maui_params_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make maui params list. The function returns Maui params (a list).

Usage

```
make_maui_params_ls(
  maui_itm_short_nms_chr,
  maui_domains_pfcs_1L_chr = NULL,
  maui_scoring_fn = NULL,
  short_and_long_nm = NULL,
  utl_min_val_1L_dbl = -1
)
```

Arguments

`maui_itm_short_nms_chr`
Maui item short names (a character vector)

`maui_domains_pfcs_1L_chr`
Maui domains pfcs (a character vector of length one), Default: NULL

`maui_scoring_fn`
Maui scoring (a function), Default: NULL

`short_and_long_nm`
PARAM_DESCRIPTION, Default: NULL

`utl_min_val_1L_dbl`
Utility minimum value (a double vector of length one), Default: -1

Value

Maui params (a list)

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_coef_range_text *Make model coefficient range text*

Description

make_mdl_coef_range_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model coefficient range text. The function returns Coefficient range text (a character vector).

Usage

```
make_mdl_coef_range_text(coef_ratios_dbl, nbr_of_digits_1L_int = 2L)
```

Arguments

`coef_ratios_dbl`
Coefficient ratios (a double vector)

`nbr_of_digits_1L_int`
Number of digits (an integer vector of length one), Default: 2

Value

Coefficient range text (a character vector)

make_mdl_desc_lines *Make model description lines*

Description

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

Usage

```
make_mdl_desc_lines(outp_smry_ls, mdl_nm_1L_chr, output_type_1L_chr = "PDF")
```

Arguments

`outp_smry_ls` Output summary (a list)

`mdl_nm_1L_chr` Model name (a character vector of length one)

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

Value

Model description lines (a character vector)

`make_nbr_at_fup_text` *Make number at follow-up text*

Description

make_nbr_at_fup_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make number at follow-up text. The function returns Number at follow-up (a character vector of length one).

Usage

```
make_nbr_at_fup_text(results_ls)
```

Arguments

`results_ls`	Results (a list)

Value

Number at follow-up (a character vector of length one)

`make_nbr_included_text` *Make number included text*

Description

make_nbr_included_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make number included text. The function is called for its side effects and does not return a value.

Usage

```
make_nbr_included_text(results_ls)
```

Arguments

`results_ls`	Results (a list)

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_output_format_ls *Make output format list*

Description

make_output_format_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make output format list. The function returns Output format (a list).

Usage

```
make_output_format_ls(
  manuscript_outp_1L_chr = "Word",
  manuscript_digits_1L_int = 2L,
  supplementary_outp_1L_chr = "PDF",
  supplementary_digits_1L_int = 2L
)
```

Arguments

manuscript_outp_1L_chr	Manuscript output (a character vector of length one), Default: 'Word'
manuscript_digits_1L_int	Manuscript digits (an integer vector of length one), Default: 2
supplementary_outp_1L_chr	Supplementary output (a character vector of length one), Default: 'PDF'
supplementary_digits_1L_int	Supplementary digits (an integer vector of length one), Default: 2

Value

Output format (a list)

`make_path_params_ls` *Make path params*

Description

make_path_params_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make path params list. The function returns Path params (a list).

Usage

```
make_path_params_ls(
  path_to_data_from_top_level_chr = NULL,
  path_from_top_level_1L_chr = NULL,
  path_to_current_1L_chr = NULL,
  dv_ds_nm_and_url_chr = NULL,
  write_new_dir_1L_lgl = F,
  use_fake_data_1L_lgl = F,
  R_fl_nm_1L_chr = "aaaaaaaaaa.txt"
)
```

Arguments

`path_to_data_from_top_level_chr`
Path to data from top level (a character vector), Default: NULL

`path_from_top_level_1L_chr`
Path from top level (a character vector of length one), Default: NULL

`path_to_current_1L_chr`
Path to current (a character vector of length one), Default: NULL

`dv_ds_nm_and_url_chr`
Dataverse dataset name and url (a character vector), Default: NULL

`write_new_dir_1L_lgl`
Write new directory (a logical vector of length one), Default: F

`use_fake_data_1L_lgl`
Use fake data (a logical vector of length one), Default: F

`R_fl_nm_1L_chr` R file name (a character vector of length one), Default: 'aaaaaaaaaa.txt'

Value

Path params (a list)

make_predrs_for_best_mdls *Make predictors for best models*

Description

make_predrs_for_best_mdls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make predictors for best models. The function returns Predictors for best models (a character vector).

Usage

```
make_predrs_for_best_mdls(outp_smry_ls, old_nms_chr = NULL, new_nms_chr = NULL)
```

Arguments

`outp_smry_ls`	Output summary (a list)
`old_nms_chr`	Old names (a character vector), Default: NULL
`new_nms_chr`	New names (a character vector), Default: NULL

Value

Predictors for best models (a character vector)

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, existing_predrs_ls = NULL)
```

Arguments

`main_predrs_chr`
: Main predictors (a character vector)

`covars_ls`
: Covariates (a list)

`existing_predrs_ls`
: Existing predictors (a list), Default: NULL

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("BET", "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("BET", "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_random_forest_text *Make random forest text*

Description

make_random_forest_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make random forest text. The function returns Text (a character vector of length one).

Usage

```
make_random_forest_text(results_ls, for_abstract_1L_lgl = F)
```

Arguments

results_ls	Results (a list)
for_abstract_1L_lgl	For abstract (a logical vector of length one), Default: F

Value

Text (a character vector of length one)

make_results_ls *Make results*

Description

make_results_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make results list. The function returns Results (a list).

Usage

```
make_results_ls(
  spine_of_results_ls = NULL,
  abstract_args_ls = NULL,
  dv_ds_nm_and_url_chr = NULL,
  output_format_ls = NULL,
  params_ls_ls = NULL,
  path_params_ls = NULL,
  study_descs_ls = NULL,
  fn_ls = NULL,
  include_idx_int = NULL,
  var_nm_change_lup = NULL,
  ctgl_vars_regrouping_ls = NULL,
  sig_covars_some_predrs_mdls_tb = NULL,
  sig_thresh_covars_1L_chr = NULL,
  version_1L_chr = NULL
)
```

Arguments

`spine_of_results_ls`
: Spine of results (a list), Default: NULL

`abstract_args_ls`
: Abstract arguments (a list), Default: NULL

`dv_ds_nm_and_url_chr`
: Dataverse dataset name and url (a character vector), Default: NULL

`output_format_ls`
: Output format (a list), Default: NULL

`params_ls_ls` Params (a list of lists), Default: NULL

`path_params_ls` Path params (a list), Default: NULL

`study_descs_ls` Study descriptions (a list), Default: NULL

`fn_ls` Function list (a list of functions), Default: NULL

`include_idx_int`
: Include index (an integer vector), Default: NULL

`var_nm_change_lup`
: Variable name change (a lookup table), Default: NULL

`ctgl_vars_regrouping_ls`
: Ctgl variables regrouping (a list), Default: NULL

`sig_covars_some_predrs_mdls_tb`
: Sig covariates some predictors models (a tibble), Default: NULL

`sig_thresh_covars_1L_chr`
: Sig thresh covariates (a character vector of length one), Default: NULL

`version_1L_chr` Version (a character vector of length one), Default: NULL

Value

Results (a list)

---

make_scaling_text *Make scaling text*

Description

make_scaling_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make scaling text. The function returns Text (a character vector of length one).

Usage

```
make_scaling_text(results_ls, table_1L_chr = "cfscl")
```

Arguments

`results_ls` Results (a list)

`table_1L_chr` Table (a character vector of length one), Default: 'cfscl'

Value

Text (a character vector of length one)

make_scndry_anlys_params

Make scndry anlys params

Description

make_scndry_anlys_params() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make scndry anlys params. The function returns New params (a list).

Usage

```
make_scndry_anlys_params(
  scndry_anlys_params_ls = NULL,
  candidate_covar_nms_chr = NULL,
  candidate_predrs_chr = NULL,
  predictors_lup = NULL,
  prefd_covars_chr = NA_character_
)
```

Arguments

`scndry_anlys_params_ls`
Scndry anlys params (a list), Default: NULL

`candidate_covar_nms_chr`
Candidate covariate names (a character vector), Default: NULL

`candidate_predrs_chr`
Candidate predictors (a character vector), Default: NULL

`predictors_lup` Predictors (a lookup table), Default: NULL

`prefd_covars_chr`
Preferred covariates (a character vector), Default: 'NA'

Value

New params (a list)

make_scndry_anlys_text

Make scndry anlys text

Description

make_scndry_anlys_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make scndry anlys text. The function returns Text (a character vector of length one).

Usage

```
make_scndry_anlys_text(results_ls)
```

Arguments

results_ls Results (a list)

Value

Text (a character vector of length one)

---

make_selected_mdl_text *Make selected model text*

Description

make_selected_mdl_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make selected model text. The function returns Text (a character vector of length one).

Usage

```
make_selected_mdl_text(results_ls, for_abstract_1L_lgl = F)
```

Arguments

results_ls Results (a list)

for_abstract_1L_lgl
For abstract (a logical vector of length one), Default: F

Value

Text (a character vector of length one)

---

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(
  fake_ds_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

fake_ds_tb	Fake dataset (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_study_descs_ls *Make study descriptions*

Description

make_study_descs_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make study descriptions list. The function returns Input params (a list).

Usage

```
make_study_descs_ls(
  input_params_ls = NULL,
  time_btwn_bl_and_fup_1L_chr,
  background_1L_chr = "",
  coi_1L_chr = "None declared.",
  conclusion_1L_chr = "",
  ethics_1L_chr = NULL,
  funding_1L_chr = NULL,
  health_utl_nm_1L_chr = NULL,
  params_ls_ls = NULL,
  predr_ctgs_ls = NULL,
  sample_desc_1L_chr = NULL,
  var_nm_change_lup = NULL
)
```

Arguments

`input_params_ls`
: Input params (a list), Default: NULL

`time_btwn_bl_and_fup_1L_chr`
: Time btwn baseline and follow-up (a character vector of length one)

`background_1L_chr`
: Background (a character vector of length one), Default: ”

`coi_1L_chr`
: Coi (a character vector of length one), Default: ’None declared.’

`conclusion_1L_chr`
: Conclusion (a character vector of length one), Default: ”

`ethics_1L_chr`
: Ethics (a character vector of length one), Default: NULL

`funding_1L_chr`
: Funding (a character vector of length one), Default: NULL

`health_utl_nm_1L_chr`
: Health utility name (a character vector of length one), Default: NULL

`params_ls_ls`
: Params (a list of lists), Default: NULL

`predr_ctgs_ls`
: Predictor category categoriess (a list), Default: NULL

`sample_desc_1L_chr`
: Sample description (a character vector of length one), Default: NULL

`var_nm_change_lup`
: Variable name change (a lookup table), Default: NULL

Value

Input params (a list)

---

make_ten_folds_tbl_title

Make ten folds table title

Description

make_ten_folds_tbl_title() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ten folds table title. The function returns Title (a character vector of length one).

Usage

```
make_ten_folds_tbl_title(results_ls, ref_1L_int = 1)
```

Arguments

`results_ls`
: Results (a list)

`ref_1L_int`
: Reference (an integer vector of length one), Default: 1

Value

Title (a character vector of length one)

make_ten_fold_text *Make ten fold text*

Description

make_ten_fold_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ten fold text. The function returns Text (a character vector of length one).

Usage

```
make_ten_fold_text(results_ls, for_abstract_1L_lgl = F)
```

Arguments

`results_ls`	Results (a list)
`for_abstract_1L_lgl`	For abstract (a logical vector of length one), Default: F

Value

Text (a character vector of length one)

make_tfmn_cmprsn_plt *Make transformation comparison*

Description

make_tfmn_cmprsn_plt() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make transformation comparison plot. The function returns Transformation comparison (a plot).

Usage

```
make_tfmn_cmprsn_plt(data_tb, depnt_var_nm_1L_chr, dictionary_tb)
```

Arguments

`data_tb`	Data (a tibble)
`depnt_var_nm_1L_chr`	Dependent variable name (a character vector of length one)
`dictionary_tb`	Dictionary (a tibble)

Value

Transformation comparison (a plot)

make_within_between_ratios_text *Make within between ratios text*

Description

make_within_between_ratios_text() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make within between ratios text. The function returns Text (a character vector of length one).

Usage

```
make_within_between_ratios_text(results_ls, exclude_covars_1L_lgl = F)
```

Arguments

`results_ls`	Results (a list)
`exclude_covars_1L_lgl`	Exclude covariates (a logical vector of length one), Default: F

Value

Text (a character vector of length one)

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 13 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)
- **fixed_acronym_chr** Fixed acronym (a character vector)
- **mixed_acronym_chr** Mixed acronym (a character vector)
- **mixed_type_chr** Mixed type (a character vector)
- **with_chr** With (a character vector)

`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 10 rows and 2 columns.

Details

A tibble

- **short_name_chr** Short name (a character vector)
- **long_name_chr** Long name (a character 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,
  force_new_data_1L_lgl = F,
  utl_min_val_1L_dbl = 0.03,
  impute_1L_lgl = T,
  utl_cls_fn = NULL,
  new_data_is_1L_chr = "Predicted",
  predn_type_1L_chr = NULL,
  sd_dbl = NA_real_,
  tfmn_for_bnml_1L_lgl = F,
  family_1L_chr = NA_character_,
  is_brms_mdl_1L_lgl = T
)
```

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
`force_new_data_1L_lgl`	Force new data (a logical vector of length one), Default: F
`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
`new_data_is_1L_chr`	New data is (a character vector of length one), Default: 'Predicted'
`predn_type_1L_chr`	Prediction type (a character vector of length one), Default: NULL
`sd_dbl`	Standard deviation (a double vector), Default: NA
`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'
`is_brms_mdl_1L_lgl`	Is bayesian regression models model (a logical vector of length one), Default: T

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_cohort_table` *Print cohort table*

Description

print_cohort_table() is a Print function that prints output to console Specifically, this function implements an algorithm to print cohort table. The function is called for its side effects and does not return a value.

Usage

```
print_cohort_table(params_ls, caption_1L_chr, mkdn_tbl_ref_1L_chr)
```

Arguments

`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`mkdn_tbl_ref_1L_chr`	Markdown table reference (a character vector of length one)

`print_corls_tbl` *Print corls table*

Description

print_corls_tbl() is a Print function that prints output to console Specifically, this function implements an algorithm to print corls table. The function is called for its side effects and does not return a value.

Usage

```
print_corls_tbl(params_ls, caption_1L_chr, mkdn_tbl_ref_1L_chr)
```

Arguments

`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`mkdn_tbl_ref_1L_chr`	Markdown table reference (a character vector of length one)

`print_covar_ttu_tbls` *Print covariate ttu tables*

Description

print_covar_ttu_tbls() is a Print function that prints output to console Specifically, this function implements an algorithm to print covariate ttu tables. The function is called for its side effects and does not return a value.

Usage

```
print_covar_ttu_tbls(params_ls, caption_1L_chr, table_1L_chr, ref_1L_int = 1)
```

Arguments

`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`table_1L_chr`	Table (a character vector of length one)
`ref_1L_int`	Reference (an integer vector of length one), Default: 1

`print_indpnt_predrs_coefs_tbl` *Print indpnt predictors coefficients table*

Description

print_indpnt_predrs_coefs_tbl() is a Print function that prints output to console Specifically, this function implements an algorithm to print indpnt predictors coefficients table. The function is called for its side effects and does not return a value.

Usage

```
print_indpnt_predrs_coefs_tbl(params_ls, caption_1L_chr, mkdn_tbl_ref_1L_chr)
```

Arguments

`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`mkdn_tbl_ref_1L_chr`	Markdown table reference (a character vector of length one)

`print_indpnt_predrs_lngl_mdl_coefs` *Print indpnt predictors lngl model coefficients*

Description

print_indpnt_predrs_lngl_mdl_coefs() is a Print function that prints output to console Specifically, this function implements an algorithm to print indpnt predictors lngl model coefficients. The function is called for its side effects and does not return a value.

Usage

```
print_indpnt_predrs_lngl_mdl_coefs(
  params_ls,
  caption_1L_chr,
  ref_1L_int = 1,
  table_1L_chr
)
```

Arguments

`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`ref_1L_int`	Reference (an integer vector of length one), Default: 1
`table_1L_chr`	Table (a character vector of length one)

print_lngl_ttu_tbls *Print lngl ttu tables*

Description

print_lngl_ttu_tbls() is a Print function that prints output to console Specifically, this function implements an algorithm to print lngl ttu tables. The function is called for its side effects and does not return a value.

Usage

```
print_lngl_ttu_tbls(
  table_df,
  params_ls,
  caption_1L_chr,
  table_1L_chr,
  ref_1L_int = 1
)
```

Arguments

`table_df`	Table (a data.frame)
`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`table_1L_chr`	Table (a character vector of length one)
`ref_1L_int`	Reference (an integer vector of length one), Default: 1

print_ten_folds_tbl *Print ten folds table*

Description

print_ten_folds_tbl() is a Print function that prints output to console Specifically, this function implements an algorithm to print ten folds table. The function is called for its side effects and does not return a value.

Usage

```
print_ten_folds_tbl(
  params_ls,
  caption_1L_chr,
  mkdn_tbl_ref_1L_chr,
  ref_1L_int = 1
)
```

Arguments

`params_ls`	Params (a list)
`caption_1L_chr`	Caption (a character vector of length one)
`mkdn_tbl_ref_1L_chr`	Markdown table reference (a character vector of length one)
`ref_1L_int`	Reference (an integer vector of length one), Default: 1

`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)

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 = "utl_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: 'utl_total_w'

`method_1L_chr` Method (a character vector of length one), Default: 'pearson'

Value

Reordered candidate (predictors)

`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 3 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)

`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",
  new_data_is_1L_chr = "Predicted",
  predn_type_1L_chr = NULL,
  family_1L_chr = NA_character_,
  impute_1L_lgl = F,
  is_brms_mdl_1L_lgl = F,
  sd_dbl = NA_real_,
  sfx_1L_chr = "",
  tfmn_for_bnml_1L_lgl = F,
  tfmn_1L_chr = "NTF",
  utl_cls_fn = NULL,
  utl_min_val_1L_dbl = NA_real_
)
```

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'

`new_data_is_1L_chr` New data is (a character vector of length one), Default: 'Predicted'

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

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

`impute_1L_lgl` Impute (a logical vector of length one), Default: F

`is_brms_mdl_1L_lgl` Is bayesian regression models model (a logical vector of length one), Default: F

`sd_dbl` Standard deviation (a double vector), Default: NA

`sfx_1L_chr` Suffix (a character vector of length one), Default: ”

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

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

`utl_cls_fn` Utility class (a function), Default: NULL

`utl_min_val_1L_dbl` Utility minimum value (a double vector of length one), Default: NA

Value

Transformed data (a tibble)

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_to_predn_ds

Transform dataset to prediction dataset

Description

transform_ds_to_predn_ds() 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 to prediction dataset. Function argument data_tb specifies the object to be updated. Argument predr_vars_nms_chr provides the object to be updated. The function returns Data (a tibble).

Usage

```
transform_ds_to_predn_ds(
  data_tb,
  predr_vars_nms_chr,
  tfmn_1L_chr,
  depnt_var_nm_1L_chr,
  id_var_nm_1L_chr,
  round_var_nm_1L_chr,
  round_bl_val_1L_chr,
  predictors_lup
)
```

Arguments

data_tb	Data (a tibble)
predr_vars_nms_chr	Predictor variables names (a character vector)
tfmn_1L_chr	Transformation (a character vector of length one)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one)
id_var_nm_1L_chr	Identity variable name (a character vector of length one)
round_var_nm_1L_chr	Round variable name (a character vector of length one)
round_bl_val_1L_chr	Round baseline value (a character vector of length one)
predictors_lup	Predictors (a lookup table)

Value

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 = "utl_total_w",
  class_fn_1L_chr = "as.numeric"
)
```

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: 'utl_total_w'
`class_fn_1L_chr`	Class function (a character vector of length one), Default: 'as.numeric'

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 = 1,
  tfmn_1L_chr = "NTF",
  ungroup_1L_lgl = F
)
```

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: 1
`tfmn_1L_chr`	Transformation (a character vector of length one), Default: 'NTF'
`ungroup_1L_lgl`	Ungroup (a logical vector of length one), Default: F

Value

Transformed for model input (a tibble)

`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_analyses` *Write analyses*

Description

write_analyses() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write analyses. 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_analyses(
  input_params_ls,
  abstract_args_ls = NULL,
  start_at_int = c(2, 1)
)
```

Arguments

`input_params_ls`	Input params (a list)
`abstract_args_ls`	Abstract arguments (a list), Default: NULL
`start_at_int`	Start at (an integer vector), Default: c(2, 1)

write_csp_output *Write csp output*

Description

write_csp_output() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write csp output. The function returns Results (a list).

Usage

```
write_csp_output(path_to_CSP_1L_chr)
```

Arguments

path_to_CSP_1L_chr	Path to CSP (a character vector of length one)

Value

Results (a list)

write_main_oupt_dir *Write main oupt directory*

Description

write_main_oupt_dir() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write main oupt directory. The function returns Paths (a list).

Usage

```
write_main_oupt_dir(
  params_ls = NULL,
  use_fake_data_1L_lgl = F,
  R_fl_nm_1L_chr = "aaaaaaaaaa.txt"
)
```

Arguments

params_ls	Params (a list), Default: NULL
use_fake_data_1L_lgl	Use fake data (a logical vector of length one), Default: F
R_fl_nm_1L_chr	R file name (a character vector of length one), Default: 'aaaaaaaaaa.txt'

Value

Paths (a list)

write_manuscript *Write manuscript*

Description

write_manuscript() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write manuscript. The function returns Results (a list).

Usage

```
write_manuscript(
  abstract_args_ls = NULL,
  input_params_ls = NULL,
  results_ls = NULL,
  figures_in_body_lgl = NULL,
  output_type_1L_chr = NULL,
  tables_in_body_lgl = NULL,
  title_1L_chr = "Scientific manuscript",
  version_1L_chr = "0.4",
  write_to_dv_1L_lgl = F
)
```

Arguments

`abstract_args_ls`
Abstract arguments (a list), Default: NULL

`input_params_ls`
Input params (a list), Default: NULL

`results_ls` Results (a list), Default: NULL

`figures_in_body_lgl`
Figures in body (a logical vector), Default: NULL

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

`tables_in_body_lgl`
Tables in body (a logical vector), Default: NULL

`title_1L_chr` Title (a character vector of length one), Default: 'Scientific manuscript'

`version_1L_chr` Version (a character vector of length one), Default: '0.4'

`write_to_dv_1L_lgl`
Write to dataverse (a logical vector of length one), Default: F

Value

Results (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 = NULL
)
```

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), Default: NULL

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_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_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_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_report *Write report*

Description

write_report() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write report. 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_report(
  params_ls,
  paths_ls,
  rprt_nm_1L_chr,
  abstract_args_ls = NULL,
  header_yaml_args_ls = NULL,
  rprt_lup = NULL
)
```

Arguments

params_ls	Params (a list)
paths_ls	Paths (a list)
rprt_nm_1L_chr	Report name (a character vector of length one)
abstract_args_ls	Abstract arguments (a list), Default: NULL
header_yaml_args_ls	Header yaml arguments (a list), Default: NULL
rprt_lup	Report (a lookup table), Default: NULL

write_reporting_dir *Write reporting directory*

Description

write_reporting_dir() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write reporting directory. The function returns Path to CSP (a character vector of length one).

Usage

```
write_reporting_dir(
  path_to_write_to_1L_chr = getwd(),
  new_dir_nm_1L_chr = "TTU_Project",
  overwrite_1L_lgl = FALSE
)
```

Arguments

`path_to_write_to_1L_chr`
: Path to write to (a character vector of length one), Default: getwd()

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

`overwrite_1L_lgl`
: Overwrite (a logical vector of length one), Default: FALSE

Value

Path to CSP (a character vector of length one)

`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 = "",
  write_mdls_to_dv_1L_lgl = F
)
```

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_mdls_to_dv_1L_lgl`
: Write models to dataverse (a logical vector of length one), Default: F

Value

Output summary (a list)

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_,
  dictionary_tb
)
```

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
`dictionary_tb`	Dictionary (a tibble)

Value

Summary of single predictor models (a tibble)

`write_to_delete_ds_copies`

Write to delete dataset copies

Description

write_to_delete_ds_copies() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to delete dataset copies. 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_to_delete_ds_copies(input_params_ls = NULL, paths_ls = NULL)
```

Arguments

`input_params_ls`	Input params (a list), Default: NULL
`paths_ls`	Paths (a list), Default: NULL

`write_to_delete_mdl_fls`

Write to delete model files

Description

write_to_delete_mdl_fls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to delete model files. 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_to_delete_mdl_fls(outp_smry_ls)
```

Arguments

`outp_smry_ls`	Output summary (a list)

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,
  predictors_lup,
  utl_min_val_1L_dbl = -1,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  iters_1L_int = 4000L,
  new_dir_nm_1L_chr = "F_TS_Mdls",
  prior_ls = NULL,
  control_ls = NULL
)
```

Arguments

`outp_smry_ls`	Output summary (a list)
`predictors_lup`	Predictors (a lookup table)
`utl_min_val_1L_dbl`	Utility minimum value (a double vector of length one), Default: -1
`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
`new_dir_nm_1L_chr`	New directory name (a character vector of length one), Default: 'F_TS_Mdls'
`prior_ls`	Prior (a list), Default: NULL
`control_ls`	Control (a list), Default: NULL

Value

Output summary (a list)

Index