

# 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 ( $\geq 0.0.0.9199$ ),  
 ready4fun ( $\geq 0.0.0.9298$ ),  
 ready4show ( $\geq 0.0.0.9038$ ),  
 ready4use ( $\geq 0.0.0.9133$ ),  
 rlang,  
 rmarkdown,  
 stats,  
 stringi,  
 stringr,  
 synthpop,  
 testthat,  
 tibble,  
 tidyr,  
 tidyselect,  
 utils,  
 viridis,  
 xfun,  
 youthvars ( $\geq 0.0.0.9064$ )

**VignetteBuilder** knitr

**Depends** R ( $\geq 2.10$ )

**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 . . . . .	6
abbreviations_lup . . . . .	7
add_prefd_predr_var_to_md1_smry_ls . . . . .	8
add_tfmd_var_to_ds . . . . .	8
add_uids_to_tbs_ls . . . . .	9
add_utility_predn_to_ds . . . . .	10
add_utl_predn_to_new_ds . . . . .	11
calculate_dpnt_var_tfmm . . . . .	12
calculate_rmse . . . . .	13

fit_ts_model_with_brm . . . . .	13
fns_dmt_tb . . . . .	14
fn_type_lup_tb . . . . .	15
get_background_text . . . . .	15
get_cndts_for_mxd_mdls . . . . .	16
get_conclusion_text . . . . .	16
get_covars_by_ctg . . . . .	17
get_covar_ctgs . . . . .	17
get_hlth_utl_nm . . . . .	18
get_hlth_utl_stat . . . . .	18
get_link_from_tfmn . . . . .	19
get_lngl_ttu_types . . . . .	19
get_mdls_with_signft_covars . . . . .	20
get_mdl_cmprsns . . . . .	20
get_mdl_type_from_nm . . . . .	21
get_nbr_of_predrs . . . . .	21
get_nbr_of_predrs_by_ctg . . . . .	22
get_nbr_of_scndry_analyses . . . . .	22
get_ordered_sngl_csnl_mdls . . . . .	23
get_popl_descvs . . . . .	23
get_predrs_by_ctg . . . . .	24
get_predr_ctgs . . . . .	24
get_prefd_mdl_predrs . . . . .	25
get_random_intercept . . . . .	25
get_scndry_anlys_descs . . . . .	26
get_selected_mixed_mdls . . . . .	26
get_signft_covars . . . . .	27
get_table_predn_mdl . . . . .	27
is_TTU_predictors_lup . . . . .	28
knit_from_tmpl . . . . .	28
knit_mdl_rpvt . . . . .	29
make_abstract_args_ls . . . . .	29
make_all_mdl_types_smry_tbl . . . . .	30
make_analysis_core_params_ls . . . . .	30
make_analysis_ds_smry_ls . . . . .	31
make_bl_fup_add_to_row_ls . . . . .	32
make_brms_mdl_print_ls . . . . .	32
make_brms_mdl_smry_tbl . . . . .	33
make_cmpst_sctr_and_dnsty_plt . . . . .	34
make_cndt_predr_text . . . . .	34
make_cohort_ls . . . . .	35
make_coi_text . . . . .	35
make_correlation_text . . . . .	36
make_covariates_text . . . . .	36
make_covar_ttu_tbl_refs . . . . .	37
make_covar_ttu_tbl_title . . . . .	37
make_cs_ts_ratios_tb . . . . .	38
make_data_availability_text . . . . .	38
make_dnsty_and_sctr_plt_title . . . . .	39
make_ds_descvs_ls . . . . .	39
make_ds_smry_ls . . . . .	40
make_eq5d_ds_dict . . . . .	41

make_ethics_text . . . . .	42
make_fake_eq5d_ds . . . . .	42
make_fake_ts_data . . . . .	43
make_folds_ls . . . . .	44
make_funding_text . . . . .	44
make_header_yaml_args_ls . . . . .	45
make_hlth_utl_and_predrs_ls . . . . .	45
make_indpnt_predrs_lngl_tbls_ref . . . . .	46
make_indpnt_predrs_lngl_tbl_title . . . . .	47
make_input_params . . . . .	47
make_knit_pars_ls . . . . .	48
make_lngl_ttu_r2_text . . . . .	49
make_lngl_ttu_with_covars_text . . . . .	50
make_maui_params_ls . . . . .	50
make_mdl . . . . .	51
make_mdls_ls . . . . .	52
make_mdls_smry_tbls_ls . . . . .	52
make_mdl_coef_range_text . . . . .	53
make_mdl_coef_ratio_ls . . . . .	53
make_mdl_desc_lines . . . . .	54
make_mdl_nms_ls . . . . .	54
make_mdl_smry_elmt_tbl . . . . .	55
make_mdl_smry_ls . . . . .	55
make_mdl_type_smry_tbl . . . . .	56
make_nbr_at_fup_text . . . . .	56
make_nbr_included_text . . . . .	57
make_new_TTU_predictors_lup . . . . .	57
make_output_format_ls . . . . .	58
make_paths_to_ss_plts_ls . . . . .	58
make_path_params_ls . . . . .	59
make_predn_ds_with_one_predr . . . . .	60
make_predrs_for_best_mdls . . . . .	60
make_predr_ctgs_ls . . . . .	61
make_predr_vals . . . . .	61
make_predr_vars_nms_ls . . . . .	62
make_prefd_mdls_vec . . . . .	62
make_prmry_analysis_params_ls . . . . .	63
make_psych_predrs_lup . . . . .	64
make_pt_TTU_predictors_lup . . . . .	64
make_random_forest_text . . . . .	65
make_ranked_predrs_ls . . . . .	66
make_results_ls . . . . .	66
make_results_ls_spine . . . . .	68
make_scaling_text . . . . .	69
make_scndry_anlys_params . . . . .	69
make_scndry_anlys_text . . . . .	70
make_selected_mdl_text . . . . .	70
make_shareable_mdl . . . . .	71
make_smry_of_brm_mdl . . . . .	72
make_smry_of_mdl_outp . . . . .	73
make_smry_of_ts_mdl_outp . . . . .	74
make_sngl_mdl_smry_tb . . . . .	75

make_ss_tbls_ls . . . . .	76
make_study_descs_ls . . . . .	76
make_ten_folds_tbl_title . . . . .	77
make_ten_fold_text . . . . .	78
make_tfd_sngl_predr_mdls_tb . . . . .	78
make_tfmn_cmprsn_plt . . . . .	79
make_ttu_cs_ls . . . . .	79
make_uid_rename_lup . . . . .	80
make_unique_ls_elmt_idx_int . . . . .	80
make_valid_params_ls_ls . . . . .	81
make_within_between_ratios_text . . . . .	82
mdl_types_lup . . . . .	82
plot_auto_lm . . . . .	83
plot_lnr_cmprsn . . . . .	84
plot_obsd_predd_dnst . . . . .	84
plot_obsd_predd_sctr_cmprsn . . . . .	85
plot_sctr_plt_cmprsn . . . . .	86
plt_types_lup . . . . .	86
predict_from_shareable_mdl . . . . .	87
predict_shrble_betareg . . . . .	88
predict_shrble_glm . . . . .	88
predict_shrble_lm . . . . .	89
predict_uncnstrd_utl . . . . .	90
predict_utility . . . . .	91
predict_utl_from_k10 . . . . .	92
print_all_plts_for_mdl_set . . . . .	93
print_cohort_table . . . . .	94
print_corls_tbl . . . . .	94
print_covar_ttu_tbls . . . . .	95
print_indpnt_predrs_coefs_tbl . . . . .	95
print_indpnt_predrs_lngl_mdl_coefs . . . . .	96
print_lngl_ttu_tbls . . . . .	96
print_ten_folds_tbl . . . . .	97
print_ts_mdl_plts . . . . .	97
prototype_lup . . . . .	98
randomise_changes_in_fct_levs . . . . .	98
rename_from_nmd_vec . . . . .	99
reorder_cndt_predrs_chr . . . . .	99
reorder_tbs_for_target_cors . . . . .	100
rppt_lup . . . . .	101
scramble_xx . . . . .	101
transform_chr_digit_pairs . . . . .	102
transform_data_tb_for_cmprsn . . . . .	102
transform_depnt_var_nm . . . . .	103
transform_dict_with_rename_lup . . . . .	104
transform_ds_for_all_cmprsn_plts . . . . .	105
transform_ds_for_mdlng . . . . .	106
transform_ds_to_predn_ds . . . . .	106
transform_mdl_vars_with_clss . . . . .	107
transform_names . . . . .	108
transform_params_ls_from_lup . . . . .	109
transform_params_ls_to_valid . . . . .	109

transform_paths_ls_for_scndry . . . . .	110
transform_predd_var_nm . . . . .	110
transform_predr_nm_part_of_phrases . . . . .	111
transform_rprt_lup . . . . .	112
transform_tbl_to_rnd_vars . . . . .	112
transform_tb_to_mdl_inp . . . . .	113
transform_timepoint_vals . . . . .	114
transform_ts_mdl_data . . . . .	115
transform_uid_var . . . . .	115
TTU_predictors_lup . . . . .	116
validate_TTU_predictors_lup . . . . .	117
write_analyses . . . . .	117
write_box_cox_tfmn . . . . .	118
write_csp_output . . . . .	119
write_main_oupt_dir . . . . .	119
write_manuscript . . . . .	120
write_mdls_with_covars_cmprsn . . . . .	121
write_mdl_cmprsn . . . . .	121
write_mdl_plts . . . . .	122
write_mdl_smry_rprt . . . . .	123
write_mdl_type_covars_mdls . . . . .	124
write_mdl_type_multi_outps . . . . .	125
write_mdl_type_sngl_outps . . . . .	127
write_new_outp_dir . . . . .	128
write_predr_and_covars_cmprsn . . . . .	128
write_predr_and_mdl_tstng_results . . . . .	129
write_predr_cmprsn_outps . . . . .	130
write_report . . . . .	131
write_reporting_dir . . . . .	131
write_rprt_with_rcrd . . . . .	132
write_scndry_analysis . . . . .	133
write_scndry_analysis_dir . . . . .	134
write_shareable_dir . . . . .	135
write_shareable_mdls . . . . .	135
write_shareable_mdls_to_dv . . . . .	136
write_sngl_predr_multi_mdls_outps . . . . .	137
write_study_outp_ds . . . . .	138
write_to_delete_ds_copies . . . . .	139
write_to_delete_mdl_fls . . . . .	140
write_ts_mdls . . . . .	140
write_ts_mdls_from_alg_outp . . . . .	141
write_ts_mdl_plts . . . . .	142

---

TTU-package

*TTU: Transfer to Utility Mapping Algorithm Toolkit*

---

## Description

Tools for developing, 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](mailto: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](mailto:matthew.hamilton@orygen.org.au)> ([ORCID](#))

Authors:

- Caroline Gao <[caroline.gao@orygen.org.au](mailto: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\_prefd\_predr\_var\_to\_md1\_smry\_ls

*Add preferred predictor variable to model summary*

---

**Description**

`add_prefd_predr_var_to_md1_smry_ls()` is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add preferred predictor variable to model summary list. Function argument `mdl_smry_ls` specifies the object to be updated. The function returns Model summary (a list).

**Usage**

```
add_prefd_predr_var_to_md1_smry_ls(mdl_smry_ls, ds_smry_ls)
```

**Arguments**

`mdl_smry_ls` Model summary (a list)

`ds_smry_ls` Dataset summary (a list)

**Value**

Model summary (a list)

---

add\_tfmd\_var\_to\_ds      *Add tfmd variable to dataset*

---

### Description

add\_tfmd\_var\_to\_ds() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add tfmd variable to dataset. Function argument data\_tb specifies the object to be updated. The function returns Data (a tibble).

### Usage

```
add_tfmd_var_to_ds(
  data_tb,
  depnt_var_nm_1L_chr,
  tfmn_1L_chr,
  dep_var_max_val_1L_dbl = NULL
)
```

### Arguments

data\_tb                  Data (a tibble)  
 depnt\_var\_nm\_1L\_chr                  Dependent variable name (a character vector of length one)  
 tfmn\_1L\_chr                  Transformation (a character vector of length one)  
 dep\_var\_max\_val\_1L\_dbl                  Dep variable maximum value (a double vector of length one), Default: NULL

### Value

Data (a tibble)

---

add\_uids\_to\_tbs\_ls      *Add unique identifiers to tibbles*

---

### Description

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

### Usage

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

### Arguments

tbs\_ls                  Tibbles (a list)  
 prefix\_1L\_chr          Prefix (a character vector of length one)  
 id\_var\_nm\_1L\_chr                  Identity variable name (a character vector of length one), Default: 'fkClientID'

**Value**

Tibbles (a list)

---

 add\_utility\_predn\_to\_ds

*Add utility prediction to dataset*


---

**Description**

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

**Usage**

```
add_utility_predn_to_ds(
  data_tb,
  model_md1,
  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_md1_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_md1	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_md1_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)

---

add\_utl\_predn\_to\_new\_ds

*Add utility prediction to new dataset*

---

**Description**

add\_utl\_predn\_to\_new\_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 new dataset. Function argument data\_tb specifies the object to be updated. The function returns Updated (a tibble).

**Usage**

```
add_utl_predn_to_new_ds(
  data_tb,
  ingredients_ls,
  mdl_nm_1L_chr,
  analysis_1L_chr = NULL,
  deterministic_1L_lgl = T,
  force_min_max_1L_lgl = T,
  id_var_nm_1L_chr = NULL,
  model_mdl = NULL,
  new_data_is_1L_chr = "Simulated",
  predr_vars_nms_chr = NULL,
  round_var_nm_1L_chr = "Timepoint",
  round_bl_val_1L_chr = "BL",
  utl_cls_fn = NULL,
  utl_var_nm_1L_chr = NULL
)
```

**Arguments**

data_tb	Data (a tibble)
ingredients_ls	Ingredients (a list)
mdl_nm_1L_chr	Model name (a character vector of length one)

analysis\_1L\_chr  
Analysis (a character vector of length one), Default: NULL

deterministic\_1L\_lgl  
Deterministic (a logical vector of length one), Default: T

force\_min\_max\_1L\_lgl  
Force minimum maximum (a logical vector of length one), Default: T

id\_var\_nm\_1L\_chr  
Identity variable name (a character vector of length one), Default: NULL

model\_md1  
Model (a model), Default: NULL

new\_data\_is\_1L\_chr  
New data is (a character vector of length one), Default: 'Simulated'

predr\_vars\_nms\_chr  
Predictor variables names (a character vector), Default: NULL

round\_var\_nm\_1L\_chr  
Round variable name (a character vector of length one), Default: 'Timepoint'

round\_bl\_val\_1L\_chr  
Round baseline value (a character vector of length one), Default: 'BL'

utl\_cls\_fn  
Utility class (a function), Default: NULL

utl\_var\_nm\_1L\_chr  
Utility variable name (a character vector of length one), Default: NULL

**Value**

Updated (a tibble)

---

calculate\_dpnt\_var\_tfmn

*Calculate dpnt variable transformation*

---

**Description**

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

**Usage**

```
calculate_dpnt_var_tfmn(
  dep_var_val_dbl,
  tfmn_1L_chr = "NTF",
  tfmn_is_outp_1L_lgl = F,
  dep_var_max_val_1L_dbl = NULL
)
```

**Arguments**

dep_var_val_dbl	Dep variable value (a double vector)
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
tfmn_is_outp_1L_lgl	Transformation is output (a logical vector of length one), Default: F
dep_var_max_val_1L_dbl	Dep variable maximum value (a double vector of length one), Default: NULL

**Value**

Transformed dep variable value (a double vector)

---

calculate_rmse	<i>Calculate root mean square error</i>
----------------	---

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

Root mean square error (a double vector)

---

fit_ts_model_with_brm	<i>Fit time series model with bayesian regression model</i>
-----------------------	---

---

**Description**

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

**Usage**

```
fit_ts_model_with_brm(
  data_tb,
  depnt_var_nm_1L_chr,
  predr_vars_nms_chr,
  id_var_nm_1L_chr,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  family_fn_1L_chr,
  iters_1L_int = 4000L,
  seed_1L_int = 1000L,
  prior_ls = NULL,
  control_ls = NULL
)
```

**Arguments**

data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one)
predr_vars_nms_chr	Predictor variables names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one)
backend_1L_chr	Backend (a character vector of length one), Default: getOption("brms.backend", "rstan")
family_fn_1L_chr	Family function (a character vector of length one)
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
seed_1L_int	Seed (an integer vector of length one), Default: 1000
prior_ls	Prior (a list), Default: NULL
control_ls	Control (a list), Default: NULL

**Value**

Model list (a list of models)

---

fns\_dmt\_tb

*TTU function documentation table*

---

**Description**

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

**Usage**

```
fns_dmt_tb
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 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	<i>Function type lookup table</i>
----------------	-----------------------------------

---

**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	<i>Get covariate category categories</i>
----------------	--

---

### 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 categories. Function argument results\_ls specifies the where to look for the required object. The function returns Covariate category categories (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 categories (a character vector)

---

get_hlth_utl_nm	<i>Get health utility name</i>
-----------------	--------------------------------

---

### 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	<i>Get health utility statistic</i>
-------------------	-------------------------------------

---

**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	<i>Get link from transformation</i>
--------------------	-------------------------------------

---

**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	<i>Get model comparisons</i>
-----------------	------------------------------

---

### 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	<i>Get model type from name</i>
----------------------	---------------------------------

---

### 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	<i>Get predictor category categories</i>
----------------	--

---

**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 categories. Function argument results\_ls specifies the where to look for the required object. The function returns Predictor category categories (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 categories (a character vector)

---

get_prefd_md1_predrs	<i>Get preferred model predictors</i>
----------------------	---------------------------------------

---

**Description**

get\_prefd\_md1\_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\_random\_intercept      *Get random intercept*

**Description**

get\_random\_intercept() 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 random intercept. Function argument mdl\_s\_mry\_tb specifies the where to look for the required object. The function returns Standard deviation (a double vector).

**Usage**

```
get_random_intercept(mdl_s_mry_tb, mdl_nm_1L_chr, deterministic_1L_lgl = T)
```

**Arguments**

mdl\_s\_mry\_tb      Models summary (a tibble)  
 mdl\_nm\_1L\_chr    Model name (a character vector of length one)  
 deterministic\_1L\_lgl  
                   Deterministic (a logical vector of length one), Default: T

**Value**

Standard deviation (a double vector)

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)

---

get\_table\_predn\_md1 *Get table prediction*

---

**Description**

get\_table\_predn\_md1() 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 table prediction model. Function argument mdl\_nm\_1L\_chr specifies the where to look for the required object. The function returns Table prediction (a model).

**Usage**

```
get_table_predn_md1(mdl_nm_1L_chr, ingredients_ls, analysis_1L_chr = NULL)
```

**Arguments**

mdl\_nm\_1L\_chr Model name (a character vector of length one)

ingredients\_ls Ingredients (a list)

analysis\_1L\_chr  
Analysis (a character vector of length one), Default: NULL

**Value**

Table prediction (a model)

---

is\_TTU\_predictors\_lup *Is TTU S3 class for candidate predictors lookup table*

---

**Description**

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

**Usage**

```
is_TTU_predictors_lup(x)
```

**Arguments**

x An object of any type

**Details**

TTU S3 class for candidate predictors lookup table

**Value**

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

---

knit_from_tmpl	<i>Knit from template</i>
----------------	---------------------------

---

**Description**

knit\_from\_tmpl() is a Knit function that knits a rmarkdown file Specifically, this function implements an algorithm to knit from template. The function is called for its side effects and does not return a value.

**Usage**

```
knit_from_tmpl(params_to_expand_ls, path_to_tmpl_1L_chr)
```

**Arguments**

params_to_expand_ls	Params to expand (a list)
path_to_tmpl_1L_chr	Path to template (a character vector of length one)

---

knit_md1_rprt	<i>Knit model report</i>
---------------	--------------------------

---

**Description**

knit\_md1\_rprt() is a Knit function that knits a rmarkdown file Specifically, this function implements an algorithm to knit model report. The function is called for its side effects and does not return a value.

**Usage**

```
knit_md1_rprt(knit_pars_ls, path_to_md1_rprt_tmpl_1L_chr)
```

**Arguments**

knit_pars_ls	Knit parameters (a list)
path_to_md1_rprt_tmpl_1L_chr	Path to model report template (a character vector of length one)

---

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\_all\_md1\_types\_smry\_tbl  
*Make all model types summary table*

---

### Description

make\_all\_md1\_types\_smry\_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make all model types summary table. The function returns All model types summary table (a tibble).

### Usage

```
make_all_md1_types_smry_tbl(outp_smry_ls, mdl_s_tb)
```

### Arguments

outp\_smry\_ls    Output summary (a list)  
mdl\_s\_tb        Models (a tibble)

### Value

All model types summary table (a tibble)

---

```
make_analysis_core_params_ls
      Make analysis core params
```

---

### Description

make\_analysis\_core\_params\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make analysis core params list. The function returns Analysis core params (a list).

### Usage

```
make_analysis_core_params_ls(
  ds_descvs_ls,
  mdl_smry_ls = make_mdl_smry_ls(),
  output_format_ls = make_output_format_ls(),
  predictors_lup,
  control_ls = NULL,
  iters_1L_int = 4000L,
  prefd_covars_chr = NULL,
  prefd_mdl_types_chr = NULL,
  prior_ls = NULL,
  seed_1L_int = 12345,
  candidate_covar_nms_chr = NULL,
  use_fake_data_1L_lgl = NULL
)
```

### Arguments

ds_descvs_ls	Dataset descriptives (a list)
mdl_smry_ls	Model summary (a list), Default: make_mdl_smry_ls()
output_format_ls	Output format (a list), Default: make_output_format_ls()
predictors_lup	Predictors (a lookup table)
control_ls	Control (a list), Default: NULL
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
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
candidate_covar_nms_chr	Candidate covariate names (a character vector), Default: NULL
use_fake_data_1L_lgl	Use fake data (a logical vector of length one), Default: NULL

### Value

Analysis core params (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_bl_fup_add_to_row_ls
```

*Make baseline follow-up add to row*

---

### Description

make\_bl\_fup\_add\_to\_row\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make baseline follow-up add to row list. The function returns Add to row (a list).

### Usage

```
make_bl_fup_add_to_row_ls(df, n_at_bl_1L_int, n_at_fup_1L_int)
```

### Arguments

```
df                Data.frame (a data.frame)
n_at_bl_1L_int    N at baseline (an integer vector of length one)
n_at_fup_1L_int   N at follow-up (an integer vector of length one)
```

### Value

Add to row (a list)

---

 make\_brms\_md1\_print\_ls

*Make bayesian regression models model print list*


---

### Description

make\_brms\_md1\_print\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make bayesian regression models model print list. The function returns Bayesian regression models model print (a list).

### Usage

```
make_brms_md1_print_ls(
  mdl_ls,
  label_stub_1L_chr,
  caption_1L_chr,
  output_type_1L_chr = "PDF",
  digits_1L_dbl = 2,
  big_mark_1L_chr = " "
)
```

### Arguments

mdl\_ls            Model list (a list of models)  
 label\_stub\_1L\_chr            Label stub (a character vector of length one)  
 caption\_1L\_chr    Caption (a character vector of length one)  
 output\_type\_1L\_chr            Output type (a character vector of length one), Default: 'PDF'  
 digits\_1L\_dbl    Digits (a double vector of length one), Default: 2  
 big\_mark\_1L\_chr            Big mark (a character vector of length one), Default: ' '

### Value

Bayesian regression models model print (a list)

---

 make\_brms\_md1\_smry\_tbl

*Make bayesian regression models model summary table*


---

### Description

make\_brms\_md1\_smry\_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make bayesian regression models model summary table. The function returns Bayesian regression models model summary (a tibble).

**Usage**

```
make_brms_md1_smry_tbl(smry_md1_ls, grp_1L_chr, popl_1L_chr, fam_1L_chr)
```

**Arguments**

```
smry_md1_ls      Summary (a list of models)
grp_1L_chr       Group (a character vector of length one)
popl_1L_chr      Population (a character vector of length one)
fam_1L_chr       Fam (a character vector of length one)
```

**Value**

Bayesian regression models model summary (a tibble)

---

```
make_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\_cohort\_ls        *Make cohort*

---

**Description**

make\_cohort\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make cohort list. The function returns Cohort (a list).

**Usage**

```
make_cohort_ls(  
  descv_tbls_ls,  
  ctgl_vars_regrouping_ls = NULL,  
  nbr_of_digits_1L_int = 2L  
)
```

**Arguments**

descv\_tbls\_ls    Descriptive tables (a list)  
ctgl\_vars\_regrouping\_ls  
                  Ctgl variables regrouping (a list), Default: NULL  
nbr\_of\_digits\_1L\_int  
                  Number of digits (an integer vector of length one), Default: 2

**Value**

Cohort (a list)

---

make_coi_text	<i>Make coi text</i>
---------------	----------------------

---

**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	<i>Make correlation text</i>
-----------------------	------------------------------

---

**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

<code>results_ls</code>	Results (a list)
<code>ref_1L_int</code>	Reference (an integer vector of length one), Default: 1

### Value

Title (a character vector of length one)

---

```
make_cs_ts_ratios_tb Make cs time series ratios
```

---

### Description

`make_cs_ts_ratios_tb()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make cs time series ratios tibble. The function returns Cs time series ratios (a tibble).

### Usage

```
make_cs_ts_ratios_tb(  
  predr_ctgs_ls,  
  mdl_coef_ratios_ls,  
  candidate_predrs_chr = NULL,  
  nbr_of_digits_1L_int = 2L,  
  fn_ls = NULL  
)
```

### Arguments

<code>predr_ctgs_ls</code>	Predictor category categories (a list)
<code>mdl_coef_ratios_ls</code>	Model coefficient ratios (a list)
<code>candidate_predrs_chr</code>	Candidate predictors (a character vector), Default: NULL
<code>nbr_of_digits_1L_int</code>	Number of digits (an integer vector of length one), Default: 2
<code>fn_ls</code>	Function list (a list of functions), Default: NULL

**Value**

Cs time series ratios (a tibble)

---

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	<i>Make dataset summary</i>
-----------------	-----------------------------

---

**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	<i>Make eq5d dataset dictionary</i>
-------------------	-------------------------------------

---

**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	<i>Make ethics text</i>
------------------	-------------------------

---

**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	<i>Make fake eq5d dataset</i>
-------------------	-------------------------------

---

### 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	<i>Make fake time series data</i>
-------------------	-----------------------------------

---

### 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_folds_ls	<i>Make folds</i>
---------------	-------------------

---

**Description**

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

**Usage**

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

**Arguments**

data\_tb            Data (a tibble)  
 depnt\_var\_nm\_1L\_chr  
                   Dependent variable name (a character vector of length one), Default: 'utl\_total\_w'  
 folds\_1L\_int      Folds (an integer vector of length one), Default: 10

**Value**

Folds (a list)

---

make_funding_text	<i>Make funding text</i>
-------------------	--------------------------

---

**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**

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

**Value**

Header yaml arguments (a list)

---

`make_hlth_utl_and_predrs_ls`*Make health utility and predictors*

---

**Description**

`make_hlth_utl_and_predrs_ls()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make health utility and predictors list. The function returns Health utility and predictors (a list).

**Usage**

```
make_hlth_utl_and_predrs_ls(  
  outp_smry_ls,  
  descv_tbls_ls,  
  nbr_of_digits_1L_int = 2L,  
  old_nms_chr = NULL,  
  new_nms_chr = NULL  
)
```

**Arguments**

outp\_smry\_ls    Output summary (a list)  
descv\_tbls\_ls    Descriptive tables (a list)  
nbr\_of\_digits\_1L\_int  
                  Number of digits (an integer vector of length one), Default: 2  
old\_nms\_chr     Old names (a character vector), Default: NULL  
new\_nms\_chr     New names (a character vector), Default: NULL

**Value**

Health utility and predictors (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	<i>Make</i>
----------	-------------

---

**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\_mdls\_ls      *Make models list*

---

**Description**

make\_mdls\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make models list. The function returns Models (a list).

**Usage**

```
make_mdls_ls(outp_smry_ls, mdls_tb)
```

**Arguments**

outp\_smry\_ls      Output summary (a list)  
 mdls\_tb          Models (a tibble)

**Value**

Models (a list)

---

make\_mdls\_smry\_tbls\_ls      *Make models summary tables list*

---

**Description**

make\_mdls\_smry\_tbls\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make models summary tables list. The function returns Models summary tables (a list).

**Usage**

```
make_mdls_smry_tbls_ls(outp_smry_ls, nbr_of_digits_1L_int = 2L)
```

**Arguments**

outp\_smry\_ls      Output summary (a list)  
 nbr\_of\_digits\_1L\_int      Number of digits (an integer vector of length one), Default: 2

**Value**

Models summary tables (a list)

---

make\_md1\_coef\_range\_text

*Make model coefficient range text*

---

**Description**

make\_md1\_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_md1_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\_md1\_coef\_ratio\_ls

*Make model coefficient ratio*

---

**Description**

make\_md1\_coef\_ratio\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model coefficient ratio list. The function returns Model coefficient ratios (a list).

**Usage**

```
make_md1_coef_ratio_ls(md1_ingredients_ls, predr_ctgs_ls = NULL)
```

**Arguments**

md1\_ingredients\_ls

Model ingredients (a list)

predr\_ctgs\_ls

Predictor category categories (a list), Default: NULL

**Value**

Model coefficient ratios (a list)

---

make\_md1\_desc\_lines     *Make model description lines*

---

### Description

make\_md1\_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_md1_desc_lines(otp_smry_ls, md1_nm_1L_chr, output_type_1L_chr = "PDF")
```

### Arguments

otp\_smry\_ls     Output summary (a list)  
 md1\_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\_md1\_nms\_ls     *Make model names*

---

### Description

make\_md1\_nms\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model names list. The function returns Model names (a list).

### Usage

```
make_md1_nms_ls(predr_vars_nms_ls, md1_types_chr)
```

### Arguments

predr\_vars\_nms\_ls     Predictor variables names (a list)  
 md1\_types\_chr     Model types (a character vector)

### Value

Model names (a list)

---

 make\_mdl\_smry\_elmt\_tbl

*Make model summary element table*


---

### Description

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

### Usage

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

### Arguments

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

### Value

Model element sum (a tibble)

---

make\_mdl\_smry\_ls

*Make model summary*


---

### Description

make\_mdl\_smry\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model summary list. The function returns Model summary (a list).

### Usage

```
make_mdl_smry_ls(
  mdl_types_lup = get_cndts_for_mxd_mdls(),
  mdl_types_chr = NULL,
  choose_from_pfx_chr = NULL,
  folds_1L_int = 10L,
  max_nbr_of_boruta_mdl_runs_int = 300L
)
```

### Arguments

mdl_types_lup	Model types (a lookup table), Default: get_cndts_for_mxd_mdls()
mdl_types_chr	Model types (a character vector), Default: NULL
choose_from_pfx_chr	Choose from prefix (a character vector), Default: NULL
folds_1L_int	Folds (an integer vector of length one), Default: 10
max_nbr_of_boruta_mdl_runs_int	Maximum number of boruta model runs (an integer vector), Default: 300

**Value**

Model summary (a list)

---

```
make_md1_type_smry_tbl
```

*Make model type summary table*

---

**Description**

make\_md1\_type\_smry\_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make model type summary table. The function returns Model type summary table (a tibble).

**Usage**

```
make_md1_type_smry_tbl(
  mdl_s_tb,
  mdl_nms_chr,
  mdl_type_1L_chr,
  add_md1_nm_sfx_1L_lgl = T
)
```

**Arguments**

mdl\_s\_tb           Models (a tibble)  
 mdl\_nms\_chr       Model names (a character vector)  
 mdl\_type\_1L\_chr   Model type (a character vector of length one)  
 add\_md1\_nm\_sfx\_1L\_lgl   Add model name suffix (a logical vector of length one), Default: T

**Value**

Model type summary table (a tibble)

---

```
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\_paths\_to\_ss\_plts\_ls  
*Make paths to ss plots*

---

### Description

make\_paths\_to\_ss\_plts\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make paths to ss plots list. The function returns Paths to ss plots (a list).

### Usage

```
make_paths_to_ss_plts_ls(
  output_data_dir_1L_chr,
  outp_smry_ls,
  additional_paths_chr = "/dens_and_sctr.png"
)
```

**Arguments**

output\_data\_dir\_1L\_chr  
Output data directory (a character vector of length one)

outp\_smry\_ls Output summary (a list)

additional\_paths\_chr  
Additional paths (a character vector), Default: `'/dens_and_sctr.png'`

**Value**

Paths to ss plots (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 = "aaaaaaaaa.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: `'aaaaaaaaa.txt'`

**Value**

Path params (a list)

---

```
make_predn_ds_with_one_predr
```

*Make prediction dataset with one predictor*

---

### Description

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

### Usage

```
make_predn_ds_with_one_predr(
  model_md1,
  depnt_var_nm_1L_chr = "utl_total_w",
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  predr_vals_dbl,
  predn_type_1L_chr = NULL
)
```

### Arguments

```
model_md1      Model (a model)
depnt_var_nm_1L_chr
                Dependent variable name (a character vector of length one), Default: 'utl_total_w'
tfmn_1L_chr    Transformation (a character vector of length one), Default: 'NTF'
predr_var_nm_1L_chr
                Predictor variable name (a character vector of length one)
predr_vals_dbl Predictor values (a double vector)
predn_type_1L_chr
                Prediction type (a character vector of length one), Default: NULL
```

### Value

Prediction dataset (a tibble)

---

```
make_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_ctgs_ls    Make predictor category categoriess
```

---

**Description**

make\_predr\_ctgs\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make predictor category categoriess list. The function returns Predictor category categoriess (a list).

**Usage**

```
make_predr_ctgs_ls(outp_smry_ls, include_idx_int = NULL)
```

**Arguments**

```
outp_smry_ls    Output summary (a list)
include_idx_int Include index (an integer vector), Default: NULL
```

**Value**

Predictor category categoriess (a list)

---

```
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_prmry_analysis_params_ls
    Make prmry analysis params
```

---

**Description**

make\_prmry\_analysis\_params\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make prmry analysis params list. The function returns Prmry analysis params (a list).

**Usage**

```
make_prmry_analysis_params_ls(
  analysis_core_params_ls,
  candidate_covar_nms_chr = NA_character_,
  ds_tb,
  path_params_ls,
  maui_params_ls,
  prefd_covars_chr = NULL,
  prefd_mdl_types_chr = NULL,
  raw_ds_tfmn_fn = NULL,
  subtitle_1L_chr = "Methods Report 1: Analysis Program (Primary Analysis)",
  utl_class_fn_1L_chr = "as.numeric"
)
```

**Arguments**

```
analysis_core_params_ls
    Analysis core params (a list)
candidate_covar_nms_chr
    Candidate covariate names (a character vector), Default: 'NA'
```

ds\_tb Dataset (a tibble)  
 path\_params\_ls Path params (a list)  
 maui\_params\_ls Maui params (a list)  
 prefd\_covars\_chr Preferred covariates (a character vector), Default: NULL  
 prefd\_md1\_types\_chr Preferred model types (a character vector), Default: NULL  
 raw\_ds\_tfmn\_fn Raw dataset transformation (a function), Default: NULL  
 subtitle\_1L\_chr Subtitle (a character vector of length one), Default: 'Methods Report 1: Analysis Program (Primary Analysis)'  
 utl\_class\_fn\_1L\_chr Utility class function (a character vector of length one), Default: 'as.numeric'

**Value**

Pmry analysis params (a list)

---

make\_psych\_predrs\_lup *Make psych predictors*

---

**Description**

make\_psych\_predrs\_lup() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make psych predictors lookup table. The function returns Predictors (a lookup table).

**Usage**

make\_psych\_predrs\_lup()

**Value**

Predictors (a lookup table)

---

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\_ranked\_predrs\_ls    *Make ranked predictors*

---

**Description**

make\_ranked\_predrs\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ranked predictors list. The function returns Ranked predictors (a list).

**Usage**

```
make_ranked_predrs_ls(descv_tbls_ls, old_nms_chr = NULL, new_nms_chr = NULL)
```

**Arguments**

descv\_tbls\_ls    Descriptive tables (a list)  
 old\_nms\_chr      Old names (a character vector), Default: NULL  
 new\_nms\_chr      New names (a character vector), Default: NULL

**Value**

Ranked predictors (a list)

---

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\_results\_ls\_spine *Make results list spine*

---

### Description

make\_results\_ls\_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make results list spine. The function returns Spine of results (a list).

### Usage

```
make_results_ls_spine(  
  output_format_ls = NULL,  
  params_ls_ls = NULL,  
  path_params_ls = NULL,  
  study_descs_ls,  
  fn_ls = NULL,  
  include_idx_int = NULL,  
  nbr_of_digits_1L_int = NULL,  
  output_data_dir_1L_chr = NULL,  
  var_nm_change_lup = NULL  
)
```

### Arguments

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)

fn\_ls      Function list (a list of functions), Default: NULL

include\_idx\_int      Include index (an integer vector), Default: NULL

nbr\_of\_digits\_1L\_int      Number of digits (an integer vector of length one), Default: NULL

output\_data\_dir\_1L\_chr      Output data directory (a character vector of length one), Default: NULL

var\_nm\_change\_lup      Variable name change (a lookup table), Default: NULL

### Value

Spine of results (a list)

---

make_scaling_text	<i>Make scaling text</i>
-------------------	--------------------------

---

**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 = "cfsc1")
```

**Arguments**

results_ls	Results (a list)
table_1L_chr	Table (a character vector of length one), Default: 'cfsc1'

**Value**

Text (a character vector of length one)

---

make_scndry_anlys_params	<i>Make scndry anlys params</i>
--------------------------	---------------------------------

---

**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\_md1\_text  
*Make selected model text*

---

**Description**

make\_selected\_md1\_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_md1_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_md1      Make shareable
```

---

**Description**

make\_shareable\_md1() 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_md1(
  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\_smry\_of\_brm\_md1 *Make summary of bayesian regression model*

---

**Description**

make\_smry\_of\_brm\_md1() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make summary of bayesian regression model model. The function returns Summary of bayesian regression model model (a tibble).

**Usage**

```
make_smry_of_brm_md1(  
  mdl_ls,  
  data_tb,  
  depnt_var_nm_1L_chr = "utl_total_w",  
  predr_vars_nms_chr,  
  mdl_nm_1L_chr = NA_character_,  
  seed_1L_dbl = 23456,  
  tfmn_1L_chr  
)
```

**Arguments**

mdl_ls	Model list (a list of models)
data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
predr_vars_nms_chr	Predictor variables names (a character vector)
mdl_nm_1L_chr	Model name (a character vector of length one), Default: 'NA'
seed_1L_dbl	Seed (a double vector of length one), Default: 23456
tfmn_1L_chr	Transformation (a character vector of length one)

**Value**

Summary of bayesian regression model model (a tibble)

---

make\_smry\_of\_md1\_outp *Make summary of model output*

---

### Description

make\_smry\_of\_md1\_outp() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make summary of model output. The function returns Summary of one predictor model (a tibble).

### Usage

```
make_smry_of_md1_outp(
  data_tb,
  model_md1 = NULL,
  folds_1L_int = 10,
  depnt_var_nm_1L_chr = "utl_total_w",
  start_1L_chr = NULL,
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  covar_var_nms_chr = NA_character_,
  md1_type_1L_chr = "OLS_NTF",
  md1_types_lup = NULL,
  predn_type_1L_chr = NULL
)
```

### Arguments

data_tb	Data (a tibble)
model_md1	Model (a model), Default: NULL
folds_1L_int	Folds (an integer vector of length one), Default: 10
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
start_1L_chr	Start (a character vector of length one), Default: NULL
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
md1_type_1L_chr	Model type (a character vector of length one), Default: 'OLS_NTF'
md1_types_lup	Model types (a lookup table), Default: NULL
predn_type_1L_chr	Prediction type (a character vector of length one), Default: NULL

### Value

Summary of one predictor model (a tibble)

---

 make\_smry\_of\_ts\_mdl\_outp

*Make summary of time series model output*


---

### Description

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

### Usage

```
make_smry_of_ts_mdl_outp(
  data_tb,
  predr_vars_nms_chr,
  mdl_nm_1L_chr,
  path_to_write_to_1L_chr = NA_character_,
  depnt_var_nm_1L_chr = "utl_total_w",
  id_var_nm_1L_chr = "fkClientID",
  round_var_nm_1L_chr = "round",
  round_bl_val_1L_chr = "Baseline",
  predictors_lup,
  utl_min_val_1L_dbl = -1,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  iters_1L_int = 4000L,
  mdl_types_lup,
  seed_1L_int = 1000L,
  prior_ls = NULL,
  control_ls = NULL
)
```

### Arguments

data_tb	Data (a tibble)
predr_vars_nms_chr	Predictor variables names (a character vector)
mdl_nm_1L_chr	Model name (a character vector of length one)
path_to_write_to_1L_chr	Path to write to (a character vector of length one), Default: 'NA'
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
round_bl_val_1L_chr	Round baseline value (a character vector of length one), Default: 'Baseline'
predictors_lup	Predictors (a lookup table)

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: <code>getOption("brms.backend", "rstan")</code>
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
mdl_types_lup	Model types (a lookup table)
seed_1L_int	Seed (an integer vector of length one), Default: 1000
prior_ls	Prior (a list), Default: NULL
control_ls	Control (a list), Default: NULL

**Value**

Summary of time series (a list of models)

---

`make_sngl_md1_smry_tb` *Make single model summary*

---

**Description**

`make_sngl_md1_smry_tb()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make single model summary tibble. The function returns New (a tibble).

**Usage**

```
make_sngl_md1_smry_tb(
  mdl_s_tb,
  mdl_nm_1L_chr,
  mdl_type_1L_chr,
  add_md1_nm_sfx_1L_lgl = T
)
```

**Arguments**

<code>mdl_s_tb</code>	Models (a tibble)
<code>mdl_nm_1L_chr</code>	Model name (a character vector of length one)
<code>mdl_type_1L_chr</code>	Model type (a character vector of length one)
<code>add_md1_nm_sfx_1L_lgl</code>	Add model name suffix (a logical vector of length one), Default: T

**Value**

New (a tibble)

---

make\_ss\_tbls\_ls      *Make ss tables list*

---

### Description

make\_ss\_tbls\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ss tables list. The function returns Ss tables (a list).

### Usage

```
make_ss_tbls_ls(
  outp_smry_ls,
  mdl_s_mry_tbls_ls,
  covars_mdls_ls,
  descv_tbls_ls,
  nbr_of_digits_1L_int = 2L
)
```

### Arguments

outp\_smry\_ls      Output summary (a list)  
 mdl\_s\_mry\_tbls\_ls      Models summary tables (a list)  
 covars\_mdls\_ls      Covariates models (a list)  
 descv\_tbls\_ls      Descriptive tables (a list)  
 nbr\_of\_digits\_1L\_int      Number of digits (an integer vector of length one), Default: 2

### Value

Ss tables (a list)

---

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\_tfd\_sngl\_predr\_mdls\_tb  
                                   *Make transformed single predictor models*

---

**Description**

make\_tfd\_sngl\_predr\_mdls\_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make transformed single predictor models tibble. The function returns Transformed single predictor models (a tibble).

**Usage**

```
make_tfd_sngl_predr_mdls_tb(  
  outp_smry_ls,  
  nbr_of_digits_1L_int = 2L,  
  mdl_pfx_ls = list(OLS = "Ordinary Least Squares ", GLM =  
    c("Generalised Linear Model with ", "Beta Regression Model with Binomial "))  
)
```

**Arguments**

outp\_smry\_ls    Output summary (a list)  
 nbr\_of\_digits\_1L\_int  
                   Number of digits (an integer vector of length one), Default: 2  
 mdl\_pfx\_ls      Model prefix (a list), Default: list(OLS = "Ordinary Least Squares ", GLM =  
                   c("Generalised Linear Model with ", "Beta Regression Model with Binomial "))

**Value**

Transformed single predictor models (a tibble)

---

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\_ttu\_cs\_ls            *Make ttu cs*

---

**Description**

make\_ttu\_cs\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ttu cs list. The function returns Ttu cs (a list).

**Usage**

```
make_ttu_cs_ls(  
  outp_smry_ls,  
  sig_covars_some_predrs_mdls_tb,  
  sig_thresh_covars_1L_chr  
)
```

**Arguments**

outp\_smry\_ls      Output summary (a list)  
 sig\_covars\_some\_predrs\_mdls\_tb  
                     Sig covariates some predictors models (a tibble)  
 sig\_thresh\_covars\_1L\_chr  
                     Sig thresh covariates (a character vector of length one)

**Value**

Ttu cs (a list)

---

make\_uid\_rename\_lup      *Make unique identifier rename*

---

**Description**

make\_uid\_rename\_lup() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make unique identifier rename lookup table. The function returns Unique identifier rename lookup table (a tibble).

**Usage**

```
make_uid_rename_lup(data_tb, id_var_nm_1L_chr = "UID")
```

**Arguments**

data\_tb              Data (a tibble)  
 id\_var\_nm\_1L\_chr  
                     Identity variable name (a character vector of length one), Default: 'UID'

**Value**

Unique identifier rename lookup table (a tibble)

---

make\_unique\_ls\_elmt\_idx\_int  
                             *Make unique list element index*

---

**Description**

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

**Usage**

```
make_unique_ls_elmt_idx_int(data_ls)
```

**Arguments**

data\_ls            Data (a list)

**Value**

Unique list element index (an integer vector)

---

make\_valid\_params\_ls\_ls

*Make valid params*

---

**Description**

make\_valid\_params\_ls\_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make valid params list list. The function returns Valid params (a list of lists).

**Usage**

```
make_valid_params_ls_ls(
  analysis_core_params_ls,
  ds_tb,
  path_params_ls,
  maui_params_ls,
  candidate_covar_nms_chr = NA_character_,
  prefd_covars_chr = NULL,
  prefd_md1_types_chr = NULL,
  raw_ds_tfmn_fn = NULL,
  scndry_analysis_extra_vars_chr = NA_character_,
  subtitle_1L_chr = "Methods Report 1: Analysis Program (Primary Analysis)",
  utl_class_fn_1L_chr = "as.numeric"
)
```

**Arguments**

analysis\_core\_params\_ls            Analysis core params (a list)

ds\_tb                                Dataset (a tibble)

path\_params\_ls                      Path params (a list)

maui\_params\_ls                      Maui params (a list)

candidate\_covar\_nms\_chr            Candidate covariate names (a character vector), Default: 'NA'

prefd\_covars\_chr                    Preferred covariates (a character vector), Default: NULL

prefd\_md1\_types\_chr                Preferred model types (a character vector), Default: NULL

raw\_ds\_tfmn\_fn                      Raw dataset transformation (a function), Default: NULL

scndry\_analysis\_extra\_vars\_chr    Scndry analysis extra variables (a character vector), Default: 'NA'

subtitle\_1L\_chr

Subtitle (a character vector of length one), Default: 'Methods Report 1: Analysis Program (Primary Analysis)'

utl\_class\_fn\_1L\_chr

Utility class function (a character vector of length one), Default: 'as.numeric'

### Value

Valid params (a list of lists)

---

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\_bnm1\_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)

---

plot\_auto\_lm

*Plot automatic linear model*

---

**Description**

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

**Usage**

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

**Arguments**

mdl                    Model (a model)

which\_dbl            Which (a double vector), Default: 1:6

ncol\_1L\_int        Ncol (an integer vector of length one), Default: 3

label\_size\_1L\_int

Label size (an integer vector of length one), Default: 3

---

plot_lnr_cmprsn	<i>Plot linear comparison</i>
-----------------	-------------------------------

---

### Description

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

### Usage

```
plot_lnr_cmprsn(
  data_tb,
  predn_ds_tb,
  predr_var_nm_1L_chr,
  predr_var_desc_1L_chr,
  depnt_var_nm_1L_chr = "utl_total_w",
  depnt_var_desc_1L_chr = "Total weighted utility score"
)
```

### Arguments

data_tb	Data (a tibble)
predn_ds_tb	Prediction dataset (a tibble)
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)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'Total weighted utility score'

---

plot_obsd_predd_dnst	<i>Plot observed predicted density</i>
----------------------	--

---

### Description

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

**Usage**

```
plot_obsd_predd_dnst(
  tfd_data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  depnt_var_desc_1L_chr = "Total weighted utility score",
  predd_val_var_nm_1L_chr = "Predicted",
  cmprsn_predd_var_nm_1L_chr = NA_character_
)
```

**Arguments**

tfd\_data\_tb      Transformed data (a tibble)

depnt\_var\_nm\_1L\_chr  
                  Dependent variable name (a character vector of length one), Default: 'utl\_total\_w'

depnt\_var\_desc\_1L\_chr  
                  Dependent variable description (a character vector of length one), Default: 'Total weighted utility score'

predd\_val\_var\_nm\_1L\_chr  
                  Predicted value variable name (a character vector of length one), Default: 'Predicted'

cmprsn\_predd\_var\_nm\_1L\_chr  
                  Comparison predicted variable name (a character vector of length one), Default: 'NA'

---

```
plot_obsd_predd_sctr_cmprsn
```

*Plot observed predicted scatter comparison*

---

**Description**

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

**Usage**

```
plot_obsd_predd_sctr_cmprsn(
  tfd_data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  depnt_var_desc_1L_chr = "Total weighted utility score",
  round_var_nm_1L_chr = "round",
  args_ls = NULL,
  predd_val_var_nm_1L_chr = "Predicted"
)
```

**Arguments**

tfd\_data\_tb      Transformed data (a tibble)

depnt\_var\_nm\_1L\_chr  
                  Dependent variable name (a character vector of length one), Default: 'utl\_total\_w'

depnt\_var\_desc\_1L\_chr  
 Dependent variable description (a character vector of length one), Default: 'Total weighted utility score'

round\_var\_nm\_1L\_chr  
 Round variable name (a character vector of length one), Default: 'round'

args\_ls  
 Arguments (a list), Default: NULL

predd\_val\_var\_nm\_1L\_chr  
 Predicted value variable name (a character vector of length one), Default: 'Predicted'

plot\_sctr\_plt\_cmprsn *Plot scatter plot comparison*

### Description

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

### Usage

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

### Arguments

tfd\_data\_tb  
 Transformed data (a tibble)

depnt\_var\_nm\_1L\_chr  
 Dependent variable name (a character vector of length one), Default: 'utl\_total\_w'

predd\_val\_var\_nm\_1L\_chr  
 Predicted value variable name (a character vector of length one), Default: 'Predicted'

plt\_types\_lup *Model plot types lookup table*

### Description

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

### Usage

```
plt_types_lup
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 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_from_shareable_md1
      Predict from shareable
```

---

**Description**

`predict_from_shareable_md1()` is a Predict function that makes predictions from data using a specified statistical model. Specifically, this function implements an algorithm to predict from shareable model. The function returns New data (a double vector).

**Usage**

```
predict_from_shareable_md1(
  model_md1,
  data_tb,
  predn_type_1L_chr = "response",
  sd_dbl,
  deterministic_1L_lgl = T
)
```

**Arguments**

<code>model_md1</code>	Model (a model)
<code>data_tb</code>	Data (a tibble)
<code>predn_type_1L_chr</code>	Prediction type (a character vector of length one), Default: 'response'
<code>sd_dbl</code>	Standard deviation (a double vector)
<code>deterministic_1L_lgl</code>	Deterministic (a logical vector of length one), Default: T

**Value**

New data (a double vector)

---

predict\_shrble\_betareg

*Predict shrble betareg*

---

### Description

predict\_shrble\_betareg() is a Predict function that makes predictions from data using a specified statistical model. Specifically, this function implements an algorithm to predict shrble betareg. The function is called for its side effects and does not return a value.

### Usage

```
predict_shrble_betareg(
  object,
  newdata = NULL,
  type = c("response", "link", "precision", "variance", "quantile"),
  na.action = na.pass,
  at = 0.5,
  sd_1L_dbl,
  ...
)
```

### Arguments

object	PARAM_DESCRIPTION
newdata	PARAM_DESCRIPTION, Default: NULL
type	PARAM_DESCRIPTION, Default: c("response", "link", "precision", "variance", "quantile")
na.action	PARAM_DESCRIPTION, Default: na.pass
at	PARAM_DESCRIPTION, Default: 0.5
sd_1L_dbl	Standard deviation (a double vector of length one)
...	Additional arguments

---

predict\_shrble\_glm

*Predict shrble glm*

---

### Description

predict\_shrble\_glm() is a Predict function that makes predictions from data using a specified statistical model. Specifically, this function implements an algorithm to predict shrble glm. The function is called for its side effects and does not return a value.

**Usage**

```

predict_shrble_glm(
  object,
  newdata = NULL,
  type = c("link", "response", "terms"),
  se.fit = FALSE,
  dispersion = NULL,
  terms = NULL,
  na.action = na.pass,
  sd_1L_dbl,
  ...
)

```

**Arguments**

object	PARAM_DESCRIPTION
newdata	PARAM_DESCRIPTION, Default: NULL
type	PARAM_DESCRIPTION, Default: c("link", "response", "terms")
se.fit	PARAM_DESCRIPTION, Default: FALSE
dispersion	PARAM_DESCRIPTION, Default: NULL
terms	PARAM_DESCRIPTION, Default: NULL
na.action	PARAM_DESCRIPTION, Default: na.pass
sd_1L_dbl	Standard deviation (a double vector of length one)
...	Additional arguments

---

predict_shrble_lm	<i>Predict shrble linear model</i>
-------------------	------------------------------------

---

**Description**

predict\_shrble\_lm() is a Predict function that makes predictions from data using a specified statistical model. Specifically, this function implements an algorithm to predict shrble linear model. The function is called for its side effects and does not return a value.

**Usage**

```

predict_shrble_lm(
  object,
  newdata,
  se.fit = FALSE,
  scale = NULL,
  df = Inf,
  interval = c("none", "confidence", "prediction"),
  level = 0.95,
  type = c("response", "terms"),
  terms = NULL,
  na.action = na.pass,
  pred.var = res.var/weights,
)

```

```

    weights = 1,
    sd_1L_dbl,
    ...
)

```

### Arguments

object	PARAM_DESCRIPTION
newdata	PARAM_DESCRIPTION
se.fit	PARAM_DESCRIPTION, Default: FALSE
scale	PARAM_DESCRIPTION, Default: NULL
df	Data.frame (a data.frame), Default: Inf
interval	PARAM_DESCRIPTION, Default: c("none", "confidence", "prediction")
level	PARAM_DESCRIPTION, Default: 0.95
type	PARAM_DESCRIPTION, Default: c("response", "terms")
terms	PARAM_DESCRIPTION, Default: NULL
na.action	PARAM_DESCRIPTION, Default: na.pass
pred.var	PARAM_DESCRIPTION, Default: res.var/weights
weights	PARAM_DESCRIPTION, Default: 1
sd_1L_dbl	Standard deviation (a double vector of length one)
...	Additional arguments

---

predict\_uncnstrd\_util *Predict uncnstrd utility*

---

### Description

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

### Usage

```

predict_uncnstrd_util(
  data_tb,
  model_md1,
  new_data_is_1L_chr = "Predicted",
  predn_type_1L_chr = NULL,
  tfmn_for_bnm1_1L_lgl = F,
  deterministic_1L_lgl = T,
  family_1L_chr = NA_character_,
  tfmn_1L_chr = "NTF",
  is_brms_md1_1L_lgl = F,
  force_new_data_1L_lgl = F,
  sd_dbl = NA_real_
)

```

**Arguments**

data_tb	Data (a tibble)
model_md1	Model (a model)
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
tfmn_for_bnm1_1L_lgl	Transformation for binomial (a logical vector of length one), Default: F
deterministic_1L_lgl	Deterministic (a logical vector of length one), Default: T
family_1L_chr	Family (a character vector of length one), Default: 'NA'
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
is_brms_md1_1L_lgl	Is bayesian regression models model (a logical vector of length one), Default: F
force_new_data_1L_lgl	Force new data (a logical vector of length one), Default: F
sd_dbl	Standard deviation (a double vector), Default: NA

**Value**

New data (a double vector)

---

predict_utility	<i>Predict utility</i>
-----------------	------------------------

---

**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_md1,
  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_bnm1_1L_lgl = F,
  family_1L_chr = NA_character_,
  is_brms_md1_1L_lgl = T
)
```

**Arguments**

data_tb	Data (a tibble)
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
model_md1	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
util_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
util_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_bnm1_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_md1_1L_lgl	Is bayesian regression models model (a logical vector of length one), Default: T

**Value**

Predicted utility (a double vector)

---

predict\_util\_from\_k10 *Predict utility from k10*

---

**Description**

predict\_util\_from\_k10() is a Predict function that makes predictions from data using a specified statistical model. Specifically, this function implements an algorithm to predict utility from k10. The function is called for its side effects and does not return a value.

**Usage**

```
predict_util_from_k10(
  k10_1L_dbl,
  b0_aqol_md1_1L_dbl = 0.204665,
  b1_aqol_md1_1L_dbl = -3.617134,
  b0_eq5d_md1_1L_dbl = 0.8644649,
  b1_eq5d_md1_1L_dbl = -2.926161,
  aqol_error_1L_dbl = 0,
  eq5d_error_1L_dbl = 0
)
```

**Arguments**

k10_1L_db1	K10 (a double vector of length one)
b0_aqol_md1_1L_db1	B0 Assessment of Quality of Life model (a double vector of length one), Default: 0.204665
b1_aqol_md1_1L_db1	B1 Assessment of Quality of Life model (a double vector of length one), Default: -3.617134
b0_eq5d_md1_1L_db1	B0 eq5d model (a double vector of length one), Default: 0.8644649
b1_eq5d_md1_1L_db1	B1 eq5d model (a double vector of length one), Default: -2.926161
aqol_error_1L_db1	Assessment of Quality of Life error (a double vector of length one), Default: 0
eq5d_error_1L_db1	Eq5d error (a double vector of length one), Default: 0

**Value**

NA ()

---

print\_all\_plts\_for\_md1\_set

*Print all plots for model set*


---

**Description**

print\_all\_plts\_for\_md1\_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_md1_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\_md1\_plts     *Print time series model plots*

---

**Description**

print\_ts\_md1\_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_md1_plts(paths_to_plts_chr, title_1L_chr, label_refs_chr, mdl_smry_ls)
```

**Arguments**

paths\_to\_plts\_chr Paths to plots (a character vector)  
 title\_1L\_chr Title (a character vector of length one)  
 label\_refs\_chr Label references (a character vector)  
 mdl\_smry\_ls Model summary (a list)

---

prototype\_lup *Class prototype lookup table*

---

**Description**

Metadata on classes used in readyforwhatsnext suite

**Usage**

prototype\_lup

**Format**

An object of class ready4\_class\_pt\_lup (inherits from ready4\_class\_pt\_lup, tbl\_df, tbl, data.frame) with 28 rows and 6 columns.

**Details**

A tibble

**type\_chr** Type (a character vector)  
**val\_chr** Value (a character vector)  
**pt\_ns\_chr** Prototype namespace (a character vector)  
**fn\_to\_call\_chr** Function to call (a character vector)  
**default\_val\_chr** Default value (a character vector)  
**old\_class\_lgl** Old class (a logical vector)

---

randomise\_changes\_in\_fct\_levs  
*Randomise changes in factor vector levels*

---

**Description**

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

**Usage**

randomise\_changes\_in\_fct\_levs(vector\_fct, prob\_unchanged\_dbl)

**Arguments**

vector\_fct      Vector (a factor vector)  
 prob\_unchanged\_dbl  
                   Probability unchanged (a double vector)

---

rename\_from\_nmd\_vec      *Rename from named vector*

---

**Description**

rename\_from\_nmd\_vec() is a Rename function that renames elements of an object based on a pre-specified schema. Specifically, this function implements an algorithm to rename from named vector. The function returns Renamed data (a tibble).

**Usage**

```
rename_from_nmd_vec(data_tb, nmd_vec_chr, vec_nms_as_new_1L_lgl = T)
```

**Arguments**

data\_tb            Data (a tibble)  
 nmd\_vec\_chr      Named vector (a character vector)  
 vec\_nms\_as\_new\_1L\_lgl  
                   Vector names as new (a logical vector of length one), Default: T

**Value**

Renamed data (a tibble)

---

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)

---

reorder\_tbs\_for\_target\_cors  
*Reorder tibbles for target correlations*

---

**Description**

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

**Usage**

```

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

```

**Arguments**

tbs\_ls                      Tibbles (a list)  
 cor\_dbl                      Correlation (a double vector)  
 cor\_var\_chr                Correlation variable (a character vector)  
 id\_var\_to\_rm\_1L\_chr        Identity variable to rm (a character vector of length one), Default: 'NA'

**Value**

Tibbles (a list)

---

rprt_lup	<i>Report types lookup table</i>
----------	----------------------------------

---

**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)

---

scramble_xx	<i>Scramble</i>
-------------	-----------------

---

**Description**

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

**Usage**

```
scramble_xx(vector_xx)
```

**Arguments**

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

**Value**

Scrambled vector (an output object of multiple potential types)

---

 transform\_chr\_digit\_pairs

*Transform character vector digit pairs*


---

### Description

transform\_chr\_digit\_pairs() 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 character vector digit pairs. Function argument digit\_pairs\_chr specifies the object to be updated. Argument nbr\_of\_digits\_1L\_int provides the object to be updated. The function returns Transformed digit pairs (a character vector).

### Usage

```
transform_chr_digit_pairs(digit_pairs_chr, nbr_of_digits_1L_int = 2L)
```

### Arguments

digit\_pairs\_chr

Digit pairs (a character vector)

nbr\_of\_digits\_1L\_int

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

### Value

Transformed digit pairs (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\_md1 provides the object to be updated. The function returns Transformed data (a tibble).

### Usage

```
transform_data_tb_for_cmprsn(
  data_tb,
  model_md1,
  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_md1_1L_lgl = F,
  sd_dbl = NA_real_,
  sfx_1L_chr = "",
  tfmn_for_bnm1_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_md1	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_md1_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_bnm1_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\_depnt\_var\_nm

*Transform dependent variable name*

---

### Description

transform\_depnt\_var\_nm() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform dependent variable name. Function argument depnt\_var\_nm\_1L\_chr specifies the object to be updated. Argument tfmn\_1L\_chr provides the object to be updated. The function returns Transformed dependent variable name (a character vector of length one).

**Usage**

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

**Arguments**

depnt\_var\_nm\_1L\_chr  
Dependent variable name (a character vector of length one)

tfmn\_1L\_chr Transformation (a character vector of length one), Default: 'NTF'

**Value**

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

---

transform\_dict\_with\_rename\_lup  
*Transform dictionary with rename*

---

**Description**

transform\_dict\_with\_rename\_lup() 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 dictionary with rename lookup table. Function argument dictionary\_tb specifies the object to be updated. Argument rename\_lup provides the object to be updated. The function returns Tfmd dictionary (a tibble).

**Usage**

```
transform_dict_with_rename_lup(dictionary_tb, rename_lup)
```

**Arguments**

dictionary\_tb Dictionary (a tibble)

rename\_lup Rename (a lookup table)

**Value**

Tfmd dictionary (a tibble)

---

`transform_ds_for_all_cmprsn_plts`*Transform dataset for all comparison plots*

---

### Description

`transform_ds_for_all_cmprsn_plts()` 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 all comparison plots. Function argument `tfd_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_ds_for_all_cmprsn_plts(  
  tfd_data_tb,  
  model_mdl,  
  depnt_var_nm_1L_chr,  
  is_brms_mdl_1L_lgl,  
  predn_type_1L_chr,  
  sd_dbl,  
  sfx_1L_chr = "",  
  tfmn_1L_chr,  
  utl_min_val_1L_dbl = -1  
)
```

### Arguments

<code>tfd_data_tb</code>	Transformed data (a tibble)
<code>model_mdl</code>	Model (a model)
<code>depnt_var_nm_1L_chr</code>	Dependent variable name (a character vector of length one)
<code>is_brms_mdl_1L_lgl</code>	Is bayesian regression models model (a logical vector of length one)
<code>predn_type_1L_chr</code>	Prediction type (a character vector of length one)
<code>sd_dbl</code>	Standard deviation (a double vector)
<code>sfx_1L_chr</code>	Suffix (a character vector of length one), Default: ""
<code>tfmn_1L_chr</code>	Transformation (a character vector of length one)
<code>utl_min_val_1L_dbl</code>	Utility minimum value (a double vector of length one), Default: -1

### 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\_cls

*Transform model variables with classes*

---

**Description**

transform\_mdl\_vars\_with\_cls() 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_md1_vars_with_cls(
  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_names	<i>Transform names</i>
-----------------	------------------------

---

**Description**

transform\_names() 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 names. Function argument names\_chr specifies the object to be updated. Argument rename\_lup provides the object to be updated. The function returns New names (a character vector).

**Usage**

```
transform_names(names_chr, rename_lup, invert_1L_lgl = F)
```

**Arguments**

names\_chr            Names (a character vector)

rename\_lup            Rename (a lookup table)

invert\_1L\_lgl        Invert (a logical vector of length one), Default: F

**Value**

New names (a character vector)

---

`transform_params_ls_from_lup`*Transform params list from*

---

**Description**

`transform_params_ls_from_lup()` 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 params list from lookup table. Function argument `params_ls` specifies the object to be updated. Argument `rename_lup` provides the object to be updated. The function returns Params (a list).

**Usage**

```
transform_params_ls_from_lup(params_ls, rename_lup)
```

**Arguments**

<code>params_ls</code>	Params (a list)
<code>rename_lup</code>	Rename (a lookup table)

**Value**

Params (a list)

---

`transform_params_ls_to_valid`*Transform params list to valid*

---

**Description**

`transform_params_ls_to_valid()` 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 params list to valid. Function argument `params_ls` specifies the object to be updated. Argument `scndry_analysis_extra_vars_chr` provides the object to be updated. The function returns Valid params (a list of lists).

**Usage**

```
transform_params_ls_to_valid(  
  params_ls,  
  scndry_analysis_extra_vars_chr = NA_character_  
)
```

**Arguments**

<code>params_ls</code>	Params (a list)
<code>scndry_analysis_extra_vars_chr</code>	Scndry analysis extra variables (a character vector), Default: 'NA'

**Value**

Valid params (a list of lists)

---

```
transform_paths_ls_for_scndry
```

*Transform paths list for scndry*

---

**Description**

transform\_paths\_ls\_for\_scndry() 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 paths list for scndry. Function argument paths\_ls specifies the object to be updated. Argument reference\_1L\_int provides the object to be updated. The function returns Paths (a list).

**Usage**

```
transform_paths_ls_for_scndry(
  paths_ls,
  reference_1L_int = 1,
  remove_prmry_1L_lgl = F,
  remove_mkdn_1L_lgl = F
)
```

**Arguments**

```
paths_ls          Paths (a list)
reference_1L_int   Reference (an integer vector of length one), Default: 1
remove_prmry_1L_lgl Remove prmry (a logical vector of length one), Default: F
remove_mkdn_1L_lgl Remove markdown (a logical vector of length one), Default: F
```

**Value**

Paths (a list)

---

```
transform_predd_var_nm
```

*Transform predicted variable name*

---

**Description**

transform\_predd\_var\_nm() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform predicted variable name. Function argument new\_data\_is\_1L\_chr specifies the object to be updated. Argument sfx\_1L\_chr provides the object to be updated. The function returns Tfmd predicted variable name (a character vector of length one).

**Usage**

```
transform_predd_var_nm(
  new_data_is_1L_chr,
  sfx_1L_chr = "",
  utl_min_val_1L_dbl = NA_real_
)
```

**Arguments**

`new_data_is_1L_chr`      New data is (a character vector of length one)

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

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

**Value**

Tfmd predicted variable name (a character vector of length one)

---

`transform_predr_nm_part_of_phrases`  
*Transform predictor name part of phrases*

---

**Description**

`transform_predr_nm_part_of_phrases()` 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 predictor name part of phrases. Function argument `phrases_chr` specifies the object to be updated. Argument `old_nms_chr` provides the object to be updated. The function returns Transformed phrases (a character vector).

**Usage**

```
transform_predr_nm_part_of_phrases(
  phrases_chr,
  old_nms_chr = NULL,
  new_nms_chr = NULL
)
```

**Arguments**

`phrases_chr`              Phrases (a character vector)

`old_nms_chr`              Old names (a character vector), Default: NULL

`new_nms_chr`              New names (a character vector), Default: NULL

**Value**

Transformed phrases (a character vector)

---

transform\_rpvt\_lup      *Transform report*

---

### Description

transform\_rpvt\_lup() 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 report lookup table. Function argument rpvt\_lup specifies the object to be updated. Argument add\_suplry\_rpvt\_1L\_lgl provides the object to be updated. The function returns Report (a lookup table).

### Usage

```
transform_rpvt_lup(
  rpvt_lup,
  add_suplry_rpvt_1L_lgl = T,
  add_sharing_rpvt_1L_lgl = F,
  start_at_int = NULL,
  reference_1L_int = NULL
)
```

### Arguments

rpvt\_lup            Report (a lookup table)

add\_suplry\_rpvt\_1L\_lgl  
                  Add suplry report (a logical vector of length one), Default: T

add\_sharing\_rpvt\_1L\_lgl  
                  Add sharing report (a logical vector of length one), Default: F

start\_at\_int      Start at (an integer vector), Default: NULL

reference\_1L\_int  
                  Reference (an integer vector of length one), Default: NULL

### Value

Report (a lookup table)

---

transform\_tbl\_to\_rnd\_vars  
                          *Transform table to rnd variables*

---

### Description

transform\_tbl\_to\_rnd\_vars() 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 table to rnd variables. Function argument ds\_tb specifies the object to be updated. Argument nbr\_of\_digits\_1L\_int provides the object to be updated. The function returns Transformed dataset (a tibble).

**Usage**

```
transform_tbl_to_rnd_vars(ds_tb, nbr_of_digits_1L_int = 2L)
```

**Arguments**

ds\_tb                    Dataset (a tibble)  
 nbr\_of\_digits\_1L\_int  
                           Number of digits (an integer vector of length one), Default: 2

**Value**

Transformed dataset (a tibble)

---

```
transform_tb_to_md1_inp
```

*Transform tibble to model input*

---

**Description**

transform\_tb\_to\_md1\_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_md1_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)

---

`transform_timepoint_vals`  
*Transform timepoint values*

---

**Description**

`transform_timepoint_vals()` 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 timepoint values. Function argument `timepoint_vals_chr` specifies the object to be updated. Argument `timepoint_levels_chr` provides the object to be updated. The function returns Timepoint values (a character vector).

**Usage**

```
transform_timepoint_vals(
  timepoint_vals_chr,
  timepoint_levels_chr,
  bl_val_1L_chr
)
```

**Arguments**

`timepoint_vals_chr` Timepoint values (a character vector)  
`timepoint_levels_chr` Timepoint levels (a character vector)  
`bl_val_1L_chr` Baseline value (a character vector of length one)

**Value**

Timepoint values (a character vector)

---

transform\_ts\_md1\_data *Transform time series model data*

---

### Description

transform\_ts\_md1\_data() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform time series model data. Function argument md1\_ls specifies the object to be updated. Argument data\_tb provides the object to be updated. The function returns Cnfdl (a list of models).

### Usage

```
transform_ts_md1_data(
  md1_ls,
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_vars_nms_chr,
  id_var_nm_1L_chr = "fkClientID",
  md1_nm_1L_chr
)
```

### Arguments

md1_ls	Model list (a list of models)
data_tb	Data (a tibble)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
predr_vars_nms_chr	Predictor variables names (a character vector)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
md1_nm_1L_chr	Model name (a character vector of length one)

### Value

Cnfdl (a list of models)

---

transform\_uid\_var *Transform unique identifier variable*

---

### Description

transform\_uid\_var() 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 unique identifier variable. Function argument data\_tb specifies the object to be updated. Argument id\_var\_nm\_1L\_chr provides the object to be updated. The function returns Tfmd data (a tibble).

**Usage**

```
transform_uid_var(
  data_tb,
  id_var_nm_1L_chr,
  rename_tb = NULL,
  old_new_chr = c("old_id_xx", "new_id_int")
)
```

**Arguments**

data_tb	Data (a tibble)
id_var_nm_1L_chr	Identity variable name (a character vector of length one)
rename_tb	Rename (a tibble), Default: NULL
old_new_chr	Old new (a character vector), Default: c("old_id_xx", "new_id_int")

**Value**

Tfmd data (a tibble)

---

TTU_predictors_lup	<i>TTU S3 class for candidate predictors lookup table</i>
--------------------	---

---

**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 prototype 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\_box\_cox\_tfmn      *Write box cox transformation*

---

### Description

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

### Usage

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

### Arguments

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

### Value

Path to plot (a character vector of length one)

---

write_csp_output	<i>Write csp output</i>
------------------	-------------------------

---

**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	<i>Write main oupt directory</i>
---------------------	----------------------------------

---

**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 = "aaaaaaaaa.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: 'aaaaaaaaa.txt'

**Value**

Paths (a list)

---

write_manuscript	<i>Write manuscript</i>
------------------	-------------------------

---

### 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_plts	<i>Write model plots</i>
----------------	--------------------------

---

**Description**

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

**Usage**

```
write_mdl_plts(
  data_tb,
  model_mdl,
  mdl_fl_nm_1L_chr = "OLS_NTF",
  depnt_var_nm_1L_chr = "utl_total_w",
  depnt_var_desc_1L_chr = "Utility score",
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  predr_var_desc_1L_chr,
  predr_vals_dbl,
  covar_var_nms_chr = NA_character_,
  path_to_write_to_1L_chr,
  predn_type_1L_chr = NULL,
  tfmn_for_bnm1_1L_lgl = F,
  family_1L_chr = NA_character_,
  plt_idx_int = 1:5
)
```

**Arguments**

data_tb	Data (a tibble)
model_md1	Model (a model)
mdl_fl_nm_1L_chr	Model file name (a character vector of length one), Default: 'OLS_NTF'
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'Utility score'
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
predr_var_nm_1L_chr	Predictor variable name (a character vector of length one)
predr_var_desc_1L_chr	Predictor variable description (a character vector of length one)
predr_vals_dbl	Predictor values (a double vector)
covar_var_nms_chr	Covariate variable names (a character vector), Default: 'NA'
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
predn_type_1L_chr	Prediction type (a character vector of length one), Default: NULL
tfmn_for_bnml_1L_lgl	Transformation for binomial (a logical vector of length one), Default: F
family_1L_chr	Family (a character vector of length one), Default: 'NA'
plt_idx_s_int	Plot indices (an integer vector), Default: 1:5

---

write\_md1\_smry\_rprt     *Write model summary report*

---

**Description**

write\_md1\_smry\_rprt() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model summary report. The function returns Input params (a list).

**Usage**

```
write_md1_smry_rprt(
  input_params_ls = NULL,
  header_yaml_args_ls = NULL,
  path_params_ls = NULL,
  use_fake_data_1L_lgl = FALSE,
  output_format_ls = NULL,
  abstract_args_ls = NULL,
  dv_ds_nm_and_url_chr = NULL,
  reference_int = 0,
```

```

  rp1t_lup = NULL,
  rcrd_nm_1L_chr = "AAA_RPRT_WRTNG_MTH",
  rp1t_nm_1L_chr = "AAA_TTU_MD1_CTG",
  start_at_int = c(2, 1),
  use_shareable_mdls_1L_lgl = F
)

```

### Arguments

input\_params\_ls  
Input params (a list), Default: NULL

header\_yaml\_args\_ls  
Header yaml arguments (a list), Default: NULL

path\_params\_ls Path params (a list), Default: NULL

use\_fake\_data\_1L\_lgl  
Use fake data (a logical vector of length one), Default: FALSE

output\_format\_ls  
Output format (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

reference\_int Reference (an integer vector), Default: 0

rp1t\_lup Report (a lookup table), Default: NULL

rcrd\_nm\_1L\_chr Rcrd name (a character vector of length one), Default: 'AAA\_RPRT\_WRTNG\_MTH'

rp1t\_nm\_1L\_chr Report name (a character vector of length one), Default: 'AAA\_TTU\_MD1\_CTG'

start\_at\_int Start at (an integer vector), Default: c(2, 1)

use\_shareable\_mdls\_1L\_lgl  
Use shareable models (a logical vector of length one), Default: F

### Value

Input params (a list)

---

write\_md1\_type\_covars\_mdls

*Write model type covariates models*

---

### Description

write\_md1\_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_md1_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_idx_s_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_idx_s_int	Plot indices (an integer vector), Default: c(3, 5)

**Value**

Summary of model single predictors (a tibble)

---

```
write_md1_type_sngl_outps
```

*Write model type single outputs*

---

## Description

write\_md1\_type\_sngl\_outps() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write model type single outputs. The function returns Summary of one predictor model (a tibble).

## Usage

```
write_md1_type_sngl_outps(
  data_tb,
  folds_1L_int = 10,
  depnt_var_nm_1L_chr = "utl_total_w",
  start_1L_chr = NULL,
  tfmn_1L_chr = "NTF",
  predr_var_nm_1L_chr,
  predr_var_desc_1L_chr,
  predr_vals_dbl,
  covar_var_nms_chr = NA_character_,
  mdl_type_1L_chr = "OLS_NTF",
  mdl_types_lup = NULL,
  path_to_write_to_1L_chr,
  mdl_fl_nm_1L_chr,
  plt_idx_int = NA_integer_
)
```

## Arguments

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

path\_to\_write\_to\_1L\_chr  
 Path to write to (a character vector of length one)

mdl\_fl\_nm\_1L\_chr  
 Model file name (a character vector of length one)

plt\_idxes\_int Plot indices (an integer vector), Default: NA

**Value**

Summary of one predictor model (a tibble)

---

write\_new\_outp\_dir *Write new output directory*

---

**Description**

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

**Usage**

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

**Arguments**

path\_to\_write\_to\_1L\_chr  
 Path to write to (a character vector of length one)

new\_dir\_nm\_1L\_chr  
 New directory name (a character vector of length one)

**Value**

Output directory (a character vector of length one)

---

write\_predr\_and\_covars\_cmprsn  
*Write predictor and covariates comparison*

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

Predictor and covariates comparison (a list)

---

write\_predr\_and\_md1\_tstng\_results

*Write predictor and model testing results*

---

**Description**

write\_predr\_and\_md1\_tstng\_results() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write predictor and model testing results. The function returns Output summary (a list).

**Usage**

```
write_predr_and_md1_tstng_results(  
  scored_data_tb,  
  ds_smry_ls,  
  mdl_smry_ls,  
  session_data_ls,  
  output_data_dir_1L_chr,  
  seed_1L_int = 1234  
)
```

**Arguments**

scored\_data\_tb    Scored data (a tibble)  
 ds\_smry\_ls        Dataset summary (a list)  
 mdl\_smry\_ls       Model summary (a list)  
 session\_data\_ls    Session data (a list)  
 output\_data\_dir\_1L\_chr    Output data directory (a character vector of length one)  
 seed\_1L\_int        Seed (an integer vector of length one), Default: 1234

**Value**

Output summary (a list)

---

write\_predr\_cmprsn\_outps  
*Write predictor comparison outputs*

---

**Description**

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

**Usage**

```

write_predr_cmprsn_outps(
  data_tb,
  path_to_write_to_1L_chr,
  new_dir_nm_1L_chr = "B_Candidate_Predrs_Cmprsn",
  depnt_var_nm_1L_chr = "utl_total_w",
  candidate_predrs_chr,
  max_nbr_of_boruta mdl_runs_int = 300L
)

```

**Arguments**

data\_tb            Data (a tibble)  
 path\_to\_write\_to\_1L\_chr    Path to write to (a character vector of length one)  
 new\_dir\_nm\_1L\_chr    New directory name (a character vector of length one), Default: 'B\_Candidate\_Predrs\_Cmprsn'  
 depnt\_var\_nm\_1L\_chr    Dependent variable name (a character vector of length one), Default: 'utl\_total\_w'  
 candidate\_predrs\_chr    Candidate predictors (a character vector)  
 max\_nbr\_of\_boruta mdl\_runs\_int    Maximum number of boruta model runs (an integer vector), Default: 300

**Value**

Confirmed predictors (a tibble)

---

write_report	<i>Write report</i>
--------------	---------------------

---

**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,
  rpvt_nm_1L_chr,
  abstract_args_ls = NULL,
  header_yaml_args_ls = NULL,
  rpvt_lup = NULL
)
```

**Arguments**

params_ls	Params (a list)
paths_ls	Paths (a list)
rpvt_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
rpvt_lup	Report (a lookup table), Default: NULL

---

write_reporting_dir	<i>Write reporting directory</i>
---------------------	----------------------------------

---

**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\_rpvt\_with\_rcrd *Write report with rcrd*

---

**Description**

write\_rpvt\_with\_rcrd() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write report with rcrd. 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_rpvt_with_rcrd(
  path_to_outp_fl_1L_chr,
  paths_ls,
  header_yaml_args_ls = NULL,
  rpvt_lup = NULL,
  use_fake_data_1L_lgl = F,
  rpvt_nm_1L_chr = "AAA_TTU_MDL_CTG",
  rcrd_nm_1L_chr = "AAA_RPVT_WRTNG_MTH",
  reference_1L_int = NULL,
  start_at_int = c(2, 1),
  output_type_1L_chr = "PDF",
  rpvt_output_type_1L_chr = "PDF",
  nbr_of_digits_1L_int = 2L,
  abstract_args_ls = NULL,
  main_rpvt_append_ls = NULL,
  rcrd_rpvt_append_ls = NULL
)
```

**Arguments**

path_to_outp_fl_1L_chr	Path to output file (a character vector of length one)
paths_ls	Paths (a list)
header_yaml_args_ls	Header yaml arguments (a list), Default: NULL
rprrt_lup	Report (a lookup table), Default: NULL
use_fake_data_1L_lgl	Use fake data (a logical vector of length one), Default: F
rprrt_nm_1L_chr	Report name (a character vector of length one), Default: 'AAA_TTU_MDL_CTG'
rcrd_nm_1L_chr	Rcrd name (a character vector of length one), Default: 'AAA_RPRT_WRTNG_MTH'
reference_1L_int	Reference (an integer vector of length one), Default: NULL
start_at_int	Start at (an integer vector), Default: c(2, 1)
output_type_1L_chr	Output type (a character vector of length one), Default: 'PDF'
rprrt_output_type_1L_chr	Report output type (a character vector of length one), Default: 'PDF'
nbr_of_digits_1L_int	Number of digits (an integer vector of length one), Default: 2
abstract_args_ls	Abstract arguments (a list), Default: NULL
main_rprrt_append_ls	Main report append (a list), Default: NULL
rcrd_rprrt_append_ls	Rcrd report append (a list), Default: NULL

---

write\_scndry\_analysis *Write scndry analysis*

---

**Description**

write\_scndry\_analysis() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write scndry analysis. 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_scndry_analysis(
  predictors_lup = NULL,
  valid_params_ls_ls,
  candidate_covar_nms_chr,
  candidate_predrs_chr = NULL,
  header_yaml_args_ls,
  path_params_ls,
  prefd_covars_chr = NA_character_,
```

```

reference_1L_int,
start_at_int = c(2, 1),
rprt_nm_1L_chr = "AAA_SUPLRY_ANLYS_MTH",
abstract_args_ls = NULL
)

```

### Arguments

**predictors\_lup** Predictors (a lookup table), Default: NULL  
**valid\_params\_ls\_ls**  
Valid params (a list of lists)  
**candidate\_covar\_nms\_chr**  
Candidate covariate names (a character vector)  
**candidate\_predrs\_chr**  
Candidate predictors (a character vector), Default: NULL  
**header\_yaml\_args\_ls**  
Header yaml arguments (a list)  
**path\_params\_ls** Path params (a list)  
**prefd\_covars\_chr**  
Preferred covariates (a character vector), Default: 'NA'  
**reference\_1L\_int**  
Reference (an integer vector of length one)  
**start\_at\_int** Start at (an integer vector), Default: c(2, 1)  
**rprt\_nm\_1L\_chr** Report name (a character vector of length one), Default: 'AAA\_SUPLRY\_ANLYS\_MTH'  
**abstract\_args\_ls**  
Abstract arguments (a list), Default: NULL

---

write\_scndry\_analysis\_dir

*Write scndry analysis directory*

---

### Description

write\_scndry\_analysis\_dir() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write scndry analysis directory. The function returns Paths (a list).

### Usage

```
write_scndry_analysis_dir(paths_ls, reference_1L_int = 1)
```

### Arguments

**paths\_ls** Paths (a list)  
**reference\_1L\_int**  
Reference (an integer vector of length one), Default: 1

### Value

Paths (a list)

---

write\_shareable\_dir    *Write shareable directory*

---

### Description

write\_shareable\_dir() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write shareable directory. The function returns Output directory (a character vector).

### Usage

```
write_shareable_dir(
  outp_smry_ls,
  new_dir_nm_1L_chr = "G_Shareable",
  sub_dirs_chr = c("Ingredients", "Models", "Table_Predn_Tools")
)
```

### 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'  
 sub\_dirs\_chr    Sub directories (a character vector), Default: c("Ingredients", "Models", "Table\_Predn\_Tools")

### Value

Output directory (a character vector)

---

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\_shareable\_mdls\_to\_dv  
                             *Write shareable models to dataverse*

---

**Description**

write\_shareable\_mdls\_to\_dv() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write shareable models to dataverse. The function returns Shareable models (a tibble).

**Usage**

```

write_shareable_mdls_to_dv(
  outp_smry_ls,
  new_dir_nm_1L_chr = "G_Shareable",
  shareable_title_detail_1L_chr = "",
  share_ingredients_1L_lgl = T,
  output_dir_chr = NA_character_
)

```

**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: ""  
 share\_ingredients\_1L\_lgl  
                 Share ingredients (a logical vector of length one), Default: T  
 output\_dir\_chr    Output directory (a character vector), Default: 'NA'

**Value**

Shareable models (a tibble)

---

```
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_idx_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\_idxes\_int Plot indices (an integer vector), Default: NA  
 dictionary\_tb Dictionary (a tibble)

**Value**

Summary of single predictor models (a tibble)

---

write\_study\_outp\_ds     *Write study output dataset*

---

**Description**

write\_study\_outp\_ds() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write study output dataset. The function returns Dataverse dataset name and url (a character vector).

**Usage**

```
write_study_outp_ds(
  input_params_ls,
  dv_ds_nm_and_url_chr = NULL,
  rpt_lups_ls = NULL,
  output_format_ls = NULL,
  path_params_ls = NULL,
  abstract_args_ls = NULL,

  dv_md1_desc_1L_chr = "This is a longitudinal transfer to utility model designed for use with the y
  header_yaml_args_ls = NULL,
  inc_fl_types_chr = ".pdf",
  purge_data_1L_lgl = FALSE,
  start_at_int = c(2, 1),
  use_fake_data_1L_lgl = NULL
)
```

**Arguments**

input\_params\_ls  
                   Input params (a list)  
 dv\_ds\_nm\_and\_url\_chr  
                   Dataverse dataset name and url (a character vector), Default: NULL  
 rpt\_lups\_ls     Report lups (a list), Default: NULL  
 output\_format\_ls  
                   Output format (a list), Default: NULL  
 path\_params\_ls Path params (a list), Default: NULL  
 abstract\_args\_ls  
                   Abstract arguments (a list), Default: NULL

dv_md1_desc_1L_chr	Dataverse model description (a character vector of length one), Default: 'This is a longitudinal transfer to utility model designed for use with the youthu R package.'
header_yaml_args_ls	Header yaml arguments (a list), Default: NULL
inc_fl_types_chr	Include file types (a character vector), Default: '.pdf'
purge_data_1L_lgl	Purge data (a logical vector of length one), Default: FALSE
start_at_int	Start at (an integer vector), Default: c(2, 1)
use_fake_data_1L_lgl	Use fake data (a logical vector of length one), Default: NULL

**Value**

Dataverse dataset name and url (a character vector)

---

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_mdls
```

*Write to delete model files*

---

### Description

`write_to_delete_mdls()` 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_mdls(otp_smry_ls)
```

### Arguments

`otp_smry_ls` Output summary (a list)

---

```
write_ts_mdls
```

*Write time series models*

---

### Description

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

### Usage

```
write_ts_mdls(
  data_tb,
  depnt_var_nm_1L_chr = "utl_total_w",
  predr_vars_nms_ls,
  id_var_nm_1L_chr = "fkClientID",
  round_var_nm_1L_chr = "round",
  round_bl_val_1L_chr = "Baseline",
  utl_min_val_1L_dbl = -1,
  backend_1L_chr = getOption("brms.backend", "rstan"),
  mdl_nms_ls,
  mdl_smry_dir_1L_chr,
  predictors_lup,
  iters_1L_int = 4000L,
  mdl_types_lup,
  seed_1L_int = 1000L,
  prior_ls = NULL,
  control_ls = 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'
predr_vars_nms_ls	Predictor variables names (a list)
id_var_nm_1L_chr	Identity variable name (a character vector of length one), Default: 'fkClientID'
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
round_bl_val_1L_chr	Round baseline value (a character vector of length one), Default: 'Baseline'
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")
mdl_nms_ls	Model names (a list)
mdl_smry_dir_1L_chr	Model summary directory (a character vector of length one)
predictors_lup	Predictors (a lookup table)
iters_1L_int	Iterations (an integer vector of length one), Default: 4000
mdl_types_lup	Model types (a lookup table)
seed_1L_int	Seed (an integer vector of length one), Default: 1000
prior_ls	Prior (a list), Default: NULL
control_ls	Control (a list), Default: NULL

**Value**

Models summary (a tibble)

---

write\_ts\_mdls\_from\_alg\_outp

*Write time series models from algorithm output*

---

**Description**

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

**Usage**

```
write_ts_mdls_from_alg_outp(
  outp_smry_ls,
  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)

---

<code>write_ts_mdl_plts</code>	<i>Write time series model plots</i>
--------------------------------	--------------------------------------

---

**Description**

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

**Usage**

```
write_ts_mdl_plts(
  brms_mdl,
  table_predn_mdl = NULL,
  tfd_data_tb,
  mdl_nm_1L_chr,
  path_to_write_to_1L_chr,
  depnt_var_nm_1L_chr = "utl_total_w",
  depnt_var_desc_1L_chr = "Utility score",
```

```

predn_type_1L_chr = NULL,
round_var_nm_1L_chr = "round",
sd_dbl = NA_real_,
sfx_1L_chr = " from table",
tfmn_1L_chr = "NTF",
units_1L_chr = "in",
height_dbl = c(rep(6, 2), rep(5, 8)),
width_dbl = c(rep(6, 2), rep(6, 8)),
rsl_dbl = rep(300, 10),
args_ls = NULL,
seed_1L_dbl = 23456,
utl_min_val_1L_dbl = -1
)

```

### Arguments

brms_mdl	Bayesian regression models (a model)
table_predn_mdl	Table prediction (a model), Default: NULL
tfd_data_tb	Transformed data (a tibble)
mdl_nm_1L_chr	Model name (a character vector of length one)
path_to_write_to_1L_chr	Path to write to (a character vector of length one)
depnt_var_nm_1L_chr	Dependent variable name (a character vector of length one), Default: 'utl_total_w'
depnt_var_desc_1L_chr	Dependent variable description (a character vector of length one), Default: 'Utility score'
predn_type_1L_chr	Prediction type (a character vector of length one), Default: NULL
round_var_nm_1L_chr	Round variable name (a character vector of length one), Default: 'round'
sd_dbl	Standard deviation (a double vector), Default: NA
sfx_1L_chr	Suffix (a character vector of length one), Default: ' from table'
tfmn_1L_chr	Transformation (a character vector of length one), Default: 'NTF'
units_1L_chr	Units (a character vector of length one), Default: 'in'
height_dbl	Height (a double vector), Default: c(rep(6, 2), rep(5, 8))
width_dbl	Width (a double vector), Default: c(rep(6, 2), rep(6, 8))
rsl_dbl	Resolution (a double vector), Default: rep(300, 10)
args_ls	Arguments (a list), Default: NULL
seed_1L_dbl	Seed (a double vector of length one), Default: 23456
utl_min_val_1L_dbl	Utility minimum value (a double vector of length one), Default: -1

### Value

Model plots paths (a list)

# Index

## \* datasets

- abbreviations\_lup, 7
  - fn\_type\_lup\_tb, 15
  - fns\_dmt\_tb, 14
  - mdl\_types\_lup, 82
  - plt\_types\_lup, 86
  - prototype\_lup, 98
  - rppt\_lup, 101
- abbreviations\_lup, 7
- add\_prefd\_predr\_var\_to\_mdl\_smry\_ls, 8
- add\_tfmd\_var\_to\_ds, 8
- add\_uids\_to\_tbs\_ls, 9
- add\_utility\_predn\_to\_ds, 10
- add\_utl\_predn\_to\_new\_ds, 11
- calculate\_dpnt\_var\_tfmn, 12
- calculate\_rmse, 13
- fit\_ts\_model\_with\_brm, 13
- fn\_type\_lup\_tb, 15
- fns\_dmt\_tb, 14
- get\_background\_text, 15
- get\_cndts\_for\_mxd\_mdls, 16
- get\_conclusion\_text, 16
- get\_covar\_ctgs, 17
- get\_covars\_by\_ctg, 17
- get\_hlth\_utl\_nm, 18
- get\_hlth\_utl\_stat, 18
- get\_link\_from\_tfmn, 19
- get\_lngl\_ttu\_types, 19
- get\_mdl\_cmprsns, 20
- get\_mdl\_type\_from\_nm, 21
- get\_mdls\_with\_signft\_covars, 20
- get\_nbr\_of\_predrs, 21
- get\_nbr\_of\_predrs\_by\_ctg, 22
- get\_nbr\_of\_scndry\_analyses, 22
- get\_ordered\_sngl\_csnl\_mdls, 23
- get\_popl\_descvs, 23
- get\_predr\_ctgs, 24
- get\_predrs\_by\_ctg, 24
- get\_prefd\_mdl\_predrs, 25
- get\_random\_intercept, 25
- get\_scndry\_anlys\_descs, 26
- get\_selected\_mixed\_mdls, 26
- get\_signft\_covars, 27
- get\_table\_predn\_mdl, 27
- is\_TTU\_predictors\_lup, 28
- knit\_from\_tmpl, 28
- knit\_mdl\_rppt, 29
- make\_abstract\_args\_ls, 29
- make\_all\_mdl\_types\_smry\_tbl, 30
- make\_analysis\_core\_params\_ls, 30
- make\_analysis\_ds\_smry\_ls, 31
- make\_bl\_fup\_add\_to\_row\_ls, 32
- make\_brms\_mdl\_print\_ls, 32
- make\_brms\_mdl\_smry\_tbl, 33
- make\_cmpst\_sctr\_and\_dnsty\_plt, 34
- make\_cndt\_predr\_text, 34
- make\_cohort\_ls, 35
- make\_coi\_text, 35
- make\_correlation\_text, 36
- make\_covar\_ttu\_tbl\_refs, 37
- make\_covar\_ttu\_tbl\_title, 37
- make\_covariates\_text, 36
- make\_cs\_ts\_ratios\_tb, 38
- make\_data\_availability\_text, 38
- make\_dnsty\_and\_sctr\_plt\_title, 39
- make\_ds\_descvs\_ls, 39
- make\_ds\_smry\_ls, 40
- make\_eq5d\_ds\_dict, 41
- make\_ethics\_text, 42
- make\_fake\_eq5d\_ds, 42
- make\_fake\_ts\_data, 43
- make\_folds\_ls, 44
- make\_funding\_text, 44
- make\_header\_yaml\_args\_ls, 45
- make\_hlth\_utl\_and\_predrs\_ls, 45
- make\_indpnt\_predrs\_lngl\_tbl\_title, 47
- make\_indpnt\_predrs\_lngl\_tbls\_ref, 46
- make\_input\_params, 47
- make\_knit\_pars\_ls, 48
- make\_lngl\_ttu\_r2\_text, 49
- make\_lngl\_ttu\_with\_covars\_text, 50

- make\_mau\_i\_params\_ls, 50
- make\_md1, 51
- make\_md1\_coef\_range\_text, 53
- make\_md1\_coef\_ratio\_ls, 53
- make\_md1\_desc\_lines, 54
- make\_md1\_nms\_ls, 54
- make\_md1\_smry\_elmt\_tbl, 55
- make\_md1\_smry\_ls, 55
- make\_md1\_type\_smry\_tbl, 56
- make\_mdls\_ls, 52
- make\_mdls\_smry\_tbls\_ls, 52
- make\_nbr\_at\_fup\_text, 56
- make\_nbr\_included\_text, 57
- make\_new\_TTU\_predictors\_lup, 57
- make\_output\_format\_ls, 58
- make\_path\_params\_ls, 59
- make\_paths\_to\_ss\_plts\_ls, 58
- make\_predn\_ds\_with\_one\_predr, 60
- make\_predr\_ctgs\_ls, 61
- make\_predr\_vals, 61
- make\_predr\_vars\_nms\_ls, 62
- make\_predrs\_for\_best\_mdls, 60
- make\_prefd\_mdls\_vec, 62
- make\_prmry\_analysis\_params\_ls, 63
- make\_psych\_predrs\_lup, 64
- make\_pt\_TTU\_predictors\_lup, 64
- make\_random\_forest\_text, 65
- make\_ranked\_predrs\_ls, 66
- make\_results\_ls, 66
- make\_results\_ls\_spine, 68
- make\_scaling\_text, 69
- make\_scndry\_anlys\_params, 69
- make\_scndry\_anlys\_text, 70
- make\_selected\_md1\_text, 70
- make\_shareable\_md1, 71
- make\_smry\_of\_brm\_md1, 72
- make\_smry\_of\_md1\_outp, 73
- make\_smry\_of\_ts\_md1\_outp, 74
- make\_sngl\_md1\_smry\_tb, 75
- make\_ss\_tbls\_ls, 76
- make\_study\_descs\_ls, 76
- make\_ten\_fold\_text, 78
- make\_ten\_folds\_tbl\_title, 77
- make\_tfd\_sngl\_predr\_mdls\_tb, 78
- make\_tfmn\_cmprsn\_plt, 79
- make\_ttu\_cs\_ls, 79
- make\_uid\_rename\_lup, 80
- make\_unique\_ls\_elmt\_idx\_int, 80
- make\_valid\_params\_ls\_ls, 81
- make\_within\_between\_ratios\_text, 82
- mdl\_types\_lup, 82
- plot\_auto\_lm, 83
- plot\_lnr\_cmprsn, 84
- plot\_obsd\_predd\_dnst, 84
- plot\_obsd\_predd\_sctr\_cmprsn, 85
- plot\_sctr\_plt\_cmprsn, 86
- plt\_types\_lup, 86
- predict\_from\_shareable\_md1, 87
- predict\_shrble\_betareg, 88
- predict\_shrble\_glm, 88
- predict\_shrble\_lm, 89
- predict\_uncnstrd\_utl, 90
- predict\_utility, 91
- predict\_utl\_from\_k10, 92
- print\_all\_plts\_for\_md1\_set, 93
- print\_cohort\_table, 94
- print\_corls\_tbl, 94
- print\_covar\_ttu\_tbls, 95
- print\_indpnt\_predrs\_coefs\_tbl, 95
- print\_indpnt\_predrs\_lngl\_md1\_coefs, 96
- print\_lngl\_ttu\_tbls, 96
- print\_ten\_folds\_tbl, 97
- print\_ts\_md1\_plts, 97
- prototype\_lup, 98
- randomise\_changes\_in\_fct\_levs, 98
- rename\_from\_nmd\_vec, 99
- reorder\_cndt\_predrs\_chr, 99
- reorder\_tbs\_for\_target\_cors, 100
- rprr\_lup, 101
- scramble\_xx, 101
- transform\_chr\_digit\_pairs, 102
- transform\_data\_tb\_for\_cmprsn, 102
- transform\_depnt\_var\_nm, 103
- transform\_dict\_with\_rename\_lup, 104
- transform\_ds\_for\_all\_cmprsn\_plts, 105
- transform\_ds\_for\_md1ng, 106
- transform\_ds\_to\_predn\_ds, 106
- transform\_md1\_vars\_with\_clss, 107
- transform\_names, 108
- transform\_params\_ls\_from\_lup, 109
- transform\_params\_ls\_to\_valid, 109
- transform\_paths\_ls\_for\_scndry, 110
- transform\_predd\_var\_nm, 110
- transform\_predr\_nm\_part\_of\_phrases, 111
- transform\_rprr\_lup, 112
- transform\_tb\_to\_md1\_inp, 113
- transform\_tbl\_to\_rnd\_vars, 112
- transform\_timepoint\_vals, 114
- transform\_ts\_md1\_data, 115
- transform\_uid\_var, 115
- TTU (TTU-package), 6

TTU-package, 6  
TTU\_predictors\_lup, 116  
  
validate\_TTU\_predictors\_lup, 117  
  
write\_analyses, 117  
write\_box\_cox\_tfmn, 118  
write\_csp\_output, 119  
write\_main\_oupt\_dir, 119  
write\_manuscript, 120  
write\_mdl\_cmprsn, 121  
write\_mdl\_plts, 122  
write\_mdl\_smry\_rprt, 123  
write\_mdl\_type\_covars\_mdls, 124  
write\_mdl\_type\_multi\_outps, 125  
write\_mdl\_type\_sngl\_outps, 127  
write\_mdls\_with\_covars\_cmprsn, 121  
write\_new\_outp\_dir, 128  
write\_predr\_and\_covars\_cmprsn, 128  
write\_predr\_and\_mdl\_tstng\_results, 129  
write\_predr\_cmprsn\_outps, 130  
write\_report, 131  
write\_reporting\_dir, 131  
write\_rprt\_with\_rcrd, 132  
write\_scndry\_analysis, 133  
write\_scndry\_analysis\_dir, 134  
write\_shareable\_dir, 135  
write\_shareable\_mdls, 135  
write\_shareable\_mdls\_to\_dv, 136  
write\_sngl\_predr\_multi\_mdls\_outps, 137  
write\_study\_outp\_ds, 138  
write\_to\_delete\_ds\_copies, 139  
write\_to\_delete\_mdl\_fls, 140  
write\_ts\_mdl\_plts, 142  
write\_ts\_mdls, 140  
write\_ts\_mdls\_from\_alg\_outp, 141