Package: chevron (via r-universe)

October 13, 2026

Type Package

Title Standard TLGs for Clinical Trials Reporting

Version 0.2.15

Date 2026-09-30

Description Provide standard tables, listings, and graphs (TLGs) libraries used in clinical trials. This package implements a structure to reformat the data with 'dunlin', create reporting tables using 'rtables' and 'tern' with standardized input arguments to enable quick generation of standard outputs. In addition, it also provides comprehensive data checks and script generation functionality.

License Apache License 2.0

URL https://insightsengineering.github.io/chevron/, https://github.com/insightsengineering/chevron/

BugReports https://github.com/insightsengineering/chevron/issues

Depends R (>= 4.4.0)

Imports checkmate (>= 2.1.0), dplyr (>= 1.1.0), dunlin (>= 0.1.10), forcats (>= 1.0.0), formatters (>= 0.5.12), ggplot2 (>= 3.4.0), glue (>= 1.0.0), grid, lifecycle (>= 0.2.0), lubridate (>= 1.9.4), magrittr (>= 1.5), methods, nestcolor (>= 0.1.1), purrr (>= 0.3.0), rlang (>= 1.0.0), rlistings (>= 0.2.13), rtables (>= 0.6.15), stringr (>= 1.4.1), tern (>= 0.9.11), tibble (>= 2.0.0), utils

Suggests knitr (>= 1.42), rmarkdown (>= 2.28), testthat (>= 3.3.0), tidyr (>= 0.8.3), vdiffr (>= 1.0.7), withr (>= 2.1.0)

VignetteBuilder knitr, rmarkdown

Config/Needs/verdepcheck mllg/checkmate, tidyverse/dplyr, insightsengineering/dunlin, tidyverse/forcats, insightsengineering/formatters, tidyverse/ggplot2, tidyverse/glue, r-lib/lifecycle, tidyverse/lubridate, tidyverse/magrittr, insightsengineering/nestcolor,

tidyverse/purrr, r-lib/rlang, insightsengineering/rlistings, insightsengineering/rtables, tidyverse/stringr, insightsengineering/tern, tidyverse/tibble, yihui/knitr, rstudio/rmarkdown, r-lib/testthat, tidyverse/tidyr, r-lib/vdiffr, r-lib/withr

Config/Needs/website insightsengineering/nesttemplate

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

Collate 'report_null.R' 'utils.R' 'chevron_tlg-S4class.R' 'ael01_nollt.R' 'ael02.R' 'ael03.R' 'aet01.R' 'aet01_aesi.R' 'aet02.R' 'aet03.R' 'aet04.R' 'aet05.R' 'aet05_all.R' 'aet10.R' 'assertions.R' 'cfbt01.R' 'checks.R' 'chevron_tlg-S4methods.R' 'cml02a_gl.R' 'cmt01a.R' 'cmt02_pt.R' 'coxt01.R' 'coxt02.R' 'data.R' 'dmt01.R' 'dst01.R' 'dtht01.R' 'dummy_template.R' 'egt01.R' 'egt02.R' 'egt03.R' 'egt05_qtcat.R' 'ext01.R' 'fstg01.R' 'fstg02.R' 'gen_args.R' 'kmg01.R' 'lbt01.R' 'lbt04.R' 'lbt05.R' 'lbt06.R' 'lbt07.R' 'lbt14.R' 'lbt15.R' 'mht01.R' 'mng01.R' 'package.R' 'pdt01.R' 'pdt02.R' 'reexports.R' 'rmpt01.R' 'rmpt03.R' 'rmpt04.R' 'rmpt05.R' 'rmpt06.R' 'rspt01.R' 'rtables_utils.R' 'standard_rules.R' 'ttet01.R' 'unwrap.R' 'vst01.R' 'vst02.R' 'zzz.R'

Config/pak/sysreqs cmake make libicu-dev

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

Date/Publication 2026-09-30 05:49:41 UTC

RemoteUrl https://github.com/insightsengineering/chevron

RemoteRef HEAD

RemoteSha d0dfb4b9019263ee5acc275ce9c7a5e4257fa45b

Contents

chevron-package 4
ael01_nollt_main 5
ael02_main 7
ael03_main 8
aet01_aesi_main 9
aet01_main 11
aet02_label 12
aet03_main 14
aet04_main 15

aet05_all_pre . . . 17
aet05_main . . . 18
aet10_main . . . 19
args_ls . . . 21
assert_single_value . . . 21
assert_valid_type . . . 22
assert_valid_var . . . 22
assert_valid_var_pair . . . 24
assert_valid_variable . . . 25
cfbt01_main . . . 25
chevron_tlg-class . . . 28
cml02a_gl_main . . . 30
cmt01_label . . . 31
cmt02_pt_main . . . 33
convert_to_month . . . 35
coxt01_main . . . 35
coxt02_main . . . 37
create_id_listings . . . 39
ctcv4_dir . . . 39
ctcv5_dir . . . 40
dataset . . . 40
dmt01_main . . . 41
dose_change_rule . . . 42
dst01_main . . . 43
dtht01_main . . . 45
dummy_template . . . 46
egt01_main . . . 47
egt02_1_main . . . 48
egt02_2_main . . . 50
egt03_main . . . 51
egt05_qtcat_main . . . 53
empty_rule . . . 54
ext01_main . . . 55
format_date . . . 57
fstg01_main . . . 58
fstg02_main . . . 59
gen_args . . . 61
get_grade_rule . . . 63
get_section_div . . . 63
gg_list . . . 64
gg_theme_chevron . . . 64
grob_list . . . 65
h_format_dec . . . 65
kmg01_main . . . 66
lbt01_main . . . 67
lbt04_main . . . 69
lbt05_main . . . 71
lbt06_main . . . 72

lbt07_main . . . 74
lbt14_main . . . 75
lbt15_pre . . . 77
listing_format_chevron . . . 78
lvls . . . 78
main . . . 79
mht01_label . . . 79
missing_rule . . . 81
mla_dir . . . 81
mng01_main . . . 82
nocoding . . . 84
outcome_rule . . . 84
pdt01_main . . . 85
pdt02_main . . . 86
postprocess . . . 88
preprocess . . . 89
report_null . . . 89
rmpt01_main . . . 90
rmpt03_main . . . 92
rmpt04_main . . . 93
rmpt05_main . . . 94
rmpt06_main . . . 95
rspt01_main . . . 97
run . . . 99
script . . . 100
set_section_div . . . 101
smart_prune . . . 102
std_postprocessing . . . 102
syn_data . . . 103
ttet01_main . . . 103
var_labels_for . . . 105
vst01_main . . . 105
vst02_1_main . . . 107
vst02_2_main . . . 109
yes_no_rule . . . 110

Index **111**

`chevron-package` `chevron` *package*

Description

Provide standard tables, listings, and graphs (TLGs) libraries used in clinical trials. This package implements a structure to reformat the data with 'dunlin', create reporting tables using 'rtables' and 'tern' with standardized input arguments to enable quick generation of standard outputs. In addition, it also provides comprehensive data checks and script generation functionality.

Author(s)

Maintainer: Joe Zhu <joe.zhu@roche.com> (ORCID)

Authors:

- Liming Li <liming.li@roche.com> (ORCID) (Original creator of the package)
- Benoit Falquet <benoit.falquet@roche.com> (ORCID) (Original creator of the package)
- Xiaoli Duan <xiaoli.duan@roche.com>

Other contributors:

- Adrian Waddell <adrian.waddell@gmail.com> [contributor]
- Chenkai Lv <chenkai.lv@roche.com> [contributor]
- Pawel Rucki <pawel.rucki@roche.com> [contributor]
- Tim Barnett <timothy.barnett@roche.com> [contributor]
- Tian Fang <tian.fang@roche.com> [contributor]
- F. Hoffmann-La Roche AG [copyright holder, funder]

See Also

Useful links:

- https://insightsengineering.github.io/chevron/
- https://github.com/insightsengineering/chevron/
- Report bugs at https://github.com/insightsengineering/chevron/issues

---

ael01_nollt_main — AEL01_NOLLT *Listing 1 (Default) Glossary of Preferred Terms and Investigator-Specified Terms.*

Description

AEL01_NOLLT Listing 1 (Default) Glossary of Preferred Terms and Investigator-Specified Terms.

Usage

```
ael01_nollt_main(
  adam_db,
  dataset = "adae",
  key_cols = c("AEBODSYS", "AEDECOD"),
  disp_cols = "AETERM",
  split_into_pages_by_var = NULL,
  unique_rows = TRUE,
  ...
)
```

```
ael01_nollt_pre(
  adam_db,
  dataset = "adae",
  key_cols = c("AEBODSYS", "AEDECOD"),
  disp_cols = "AETERM",
  ...
)

ael01_nollt
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
key_cols	(character) names of columns that should be treated as key columns when rendering the listing. Key columns allow you to group repeat occurrences.
disp_cols	(character) names of non-key columns which should be displayed when the listing is rendered.
split_into_pages_by_var	(character or NULL) the name of the variable to split the listing by.
unique_rows	(flag) whether to keep only unique rows in listing.
...	additional arguments passed to rlistings::as_listing.

Details

- Removes duplicate rows.
- By default, uses dataset adae, sorting by key columns AEBODSYS and AEDECOD.
- If using with a dataset other than adae, be sure to specify the desired labels for variables in key_cols and disp_cols, and pre-process missing data.

Value

the main function returns an rlistings or a list object.

the preprocessing function returns a list of data.frame.

Functions

- ael01_nollt_main(): Main TLG function
- ael01_nollt_pre(): Preprocessing

Note

- adam_db object must contain the dataset table with columns specified by key_cols and disp_cols.

Examples

```
run(ael01_nollt, syn_data)
```

ael02_main AEL02 *Listing 1 (Default) Listing of Adverse Events.*

Description

AEL02 Listing 1 (Default) Listing of Adverse Events.

Usage

```
ael02_main(
  adam_db,
  dataset = "adae",
  key_cols = c("ID", "ASR"),
 disp_cols = c("AEDECOD", "TRTSDTM", "ASTDY", "ADURN", "AESER", "ASEV", "AREL", "AEOUT",
    "AECONTRT", "AEACN"),
  split_into_pages_by_var = "ACTARM",
  unique_rows = FALSE,
  ...
)

ael02_pre(adam_db, dataset = "adae", arm_var = "ACTARM", ...)

ael02
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
key_cols	(character) names of columns that should be treated as key columns when rendering the listing. Key columns allow you to group repeat occurrences.
disp_cols	(character) names of non-key columns which should be displayed when the listing is rendered.
split_into_pages_by_var	(character or NULL) the name of the variable to split the listing by.
unique_rows	(flag) whether to keep only unique rows in listing.
...	not used.
arm_var	(string) variable used for column splitting

Value

the main function returns an rlistings or a list object.

the preprocessing function returns a list of data.frame.

Functions

- `ael02_main()`: Main TLG function
- `ael02_pre()`: Preprocessing

Examples

```
res <- run(ael02, syn_data)
```

---

ael03_main — AEL03 *Listing 1 (Default) Listing of Serious Adverse Events.*

Description

AEL03 Listing 1 (Default) Listing of Serious Adverse Events.

Usage

```
ael03_main(
  adam_db,
  dataset = "adae",
  key_cols = c("ID", "ASR"),
  disp_cols = c("AEDECOD", "TRTSDTM", "ASTDY", "ADURN", "ASEV", "AREL", "AEOUT",
    "AECONTRT", "AEACN", "SERREAS"),
  split_into_pages_by_var = "ACTARM",
  unique_rows = FALSE,
  ...
)

ael03_pre(adam_db, dataset = "adae", arm_var = "ACTARM", ...)

ael03
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
key_cols	(character) names of columns that should be treated as key columns when rendering the listing. Key columns allow you to group repeat occurrences.
disp_cols	(character) names of non-key columns which should be displayed when the listing is rendered.
split_into_pages_by_var	(character or NULL) the name of the variable to split the listing by.
unique_rows	(flag) whether to keep only unique rows in listing.
...	not used.
arm_var	(string) variable used for column splitting

Value

the main function returns an `rlistings` or a `list` object.

Functions

- `ael03_main()`: Main TLG function
- `ael03_pre()`: Preprocessing

Examples

```
res <- run(ael03, syn_data)
```

---

aet01_aesi_main — AET01_AESI *Table 1 (Default) Adverse Event of Special Interest Summary Table.*

Description

`AET01_AESI` Table 1 (Default) Adverse Event of Special Interest Summary Table.

Usage

```
aet01_aesi_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  aesi_vars = NULL,
  grade_groups = NULL,
  ...
)

aet01_aesi_pre(adam_db, ...)

aet01_aesi_post(tlg, prune_0 = FALSE, ...)

aet01_aesi
```

Arguments

`adam_db`	(`list` of `data.frames`) object containing the `ADaM` datasets
`arm_var`	(`string`) variable used for column splitting
`lbl_overall`	(`string`) label used for overall column, if set to `NULL` the overall column is omitted
`aesi_vars`	(`character`) the `AESI` variables to be included in the summary. Defaults to `NA`.
`grade_groups`	(`list`) the grade groups to be displayed.

...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Does not remove rows with zero counts by default.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- aet01_aesi_main(): Main TLG function
- aet01_aesi_pre(): Preprocessing
- aet01_aesi_post(): Postprocessing

Note

- adam_db object must contain an adae table with columns "AEOUT", "AEACN", "AECONTRT", "AESER", "AREL", and the column specified by arm_var.
- aesi_vars may contain any/all of the following variables to display: "ALLRESWD", "ALLRESDSM", "ALLRESCONTRT", "NOTRESWD", "NOTRESDSM", "NOTRESCONTRT", "SERWD", "SERDSM", "SERCONTRT", "RELWD", "RELDSM", "RELCONTRT", "RELSER".
- aesi_vars variable prefixes are defined as follows:
  - "ALLRES" = "all non-fatal adverse events resolved"
  - "NOTRES" = "at least one unresolved or ongoing non-fatal adverse event"
  - "SER" = "serious adverse event"
  - "REL" = "related adverse event"
- aesi_vars variable suffixes are defined as follows:
  - "WD" = "patients with study drug withdrawn"
  - "DSM" = "patients with dose modified/interrupted"
  - "CONTRT" = "patients with treatment received"
- Several aesi_vars can be added to the table at once:
  - aesi_vars = "ALL" will include all possible aesi_vars.
  - Including "ALL_XXX" in aesi_vars where XXX is one of the prefixes listed above will include all aesi_vars with that prefix.

Examples

```
run(aet01_aesi, syn_data)
```

aet01_main — AET01 *Table 1 (Default) Overview of Deaths and Adverse Events Summary Table 1.*

Description

AET01 Table 1 (Default) Overview of Deaths and Adverse Events Summary Table 1.

Usage

```
aet01_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
 anl_vars = list(safety_var = c("FATAL", "SER", "SERWD", "SERDSM", "RELSER", "WD",
    "DSM", "REL", "RELWD", "RELDSM", "SEV")),
  anl_lbls = "Total number of {patient_label} with at least one",
  show_wd = TRUE,
  ...
)

aet01_pre(adam_db, ...)

aet01_post(tlg, prune_0 = FALSE, ...)

aet01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
anl_vars	Named (list) of (character) variables the safety variables to be summarized.
anl_lbls	(character) of analysis labels.
show_wd	(flag) whether to display the number of patients withdrawn from study due to an adverse event and the number of death.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Does not remove rows with zero counts by default.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `aet01_main()`: Main TLG function
- `aet01_pre()`: Preprocessing
- `aet01_post()`: Postprocessing

Note

- `adam_db` object must contain an `adsl` table with the `"DTHFL"` and `"DCSREAS"` columns.
- `adam_db` object must contain an `adae` table with the columns passed to `anl_vars`.

Examples

```
run(aet01, syn_data, arm_var = "ARM")
```

---

aet02_label — `AET02` *Table 1 (Default) Adverse Events by System Organ Class and Preferred Term Table 1.*

Description

The `AET02` table provides an overview of the number of subjects experiencing adverse events and the number of advert events categorized by Body System and Dictionary-Derived Term.

Usage

```
aet02_label

aet02_main(
  adam_db,
  arm_var = "ACTARM",
  row_split_var = "AEBODSYS",
  lbl_overall = NULL,
  summary_labels = list(all = aet02_label, TOTAL = c(nonunique =
    "Overall total number of events")),
  ...
)

aet02_pre(adam_db, row_split_var = "AEBODSYS", ...)

aet02_post(tlg, row_split_var = "AEBODSYS", prune_0 = TRUE, ...)

aet02
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
row_split_var	(character) additional row split variables.
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
summary_labels	(list) of summarize labels. See details.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Numbers represent absolute numbers of subject and fraction of N, or absolute number of event when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort Dictionary-Derived Code (AEDECOD) by highest overall frequencies.
- Missing values in AEBODSYS, and AEDECOD are labeled by No Coding Available. summary_labels is used to control the summary for each level. If "all" is used, then each split will have that summary statistic with the labels. One special case is "TOTAL", this is for the overall population.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- aet02_label: Default labels
- aet02_main(): Main TLG function
- aet02_pre(): Preprocessing
- aet02_post(): Postprocessing

Note

- adam_db object must contain an adae table with the columns "AEBODSYS" and "AEDECOD".

Examples

```
run(aet02, syn_data)
```

aet03_main AET03 *Table 1 (Default) Advert Events by Greatest Intensity Table 1.*

Description

An adverse events table categorized by System Organ Class, Dictionary-Derived Term and Greatest intensity.

Usage

```
aet03_main(adam_db, arm_var = "ACTARM", lbl_overall = NULL, ...)

aet03_pre(adam_db, ...)

aet03_post(tlg, prune_0 = TRUE, ...)

aet03
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Default Adverse Events by Greatest Intensity table.
- Numbers represent absolute numbers of patients and fraction of N.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort by Body System or Organ Class (SOC) and Dictionary-Derived Term (PT).

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- `aet03_main()`: Main TLG function
- `aet03_pre()`: Preprocessing
- `aet03_post()`: Postprocessing

Note

- adam_db object must contain an adae table with the columns `"AEBODSYS"`, `"AEDECOD"` and `"ASEV"`.

Examples

```
run(aet03, syn_data)
```

---

aet04_main — AET04 *Table 1 (Default) Adverse Events by Highest* NCI CTACAE AE *Grade Table 1.*

Description

The `AET04` table provides an overview of adverse event with the highest `NCI CTCAE` grade per individual.

Usage

```
aet04_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  grade_groups = NULL,
  ...
)

aet04_pre(adam_db, ...)

aet04_post(tlg, prune_0 = TRUE, ...)

aet04
```

Arguments

adam_db	(`list` of `data.frames`) object containing the `ADaM` datasets
arm_var	(`string`) variable used for column splitting
lbl_overall	(`string`) label used for overall column, if set to `NULL` the overall column is omitted
grade_groups	(`list`) putting in correspondence toxicity grades and labels.

...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Numbers represent absolute numbers of patients and fraction of N, or absolute number of event when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Events with missing grading values are excluded.
- Split columns by arm, typically ACTARM.
- Does not include a total column by default.
- Sort Body System or Organ Class and Dictionary-Derived Term by highest overall frequencies. Analysis Toxicity Grade is sorted by severity.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- aet04_main(): Main TLG function
- aet04_pre(): Preprocessing
- aet04_post(): Postprocessing

Note

- adam_db object must contain an adae table with the columns "AEBODSYS", "AEDECOD" and "ATOXGR".

Examples

```
grade_groups <- list(
  "Grade 1-2" = c("1", "2"),
  "Grade 3-4" = c("3", "4"),
  "Grade 5" = c("5")
)
proc_data <- dunlin::log_filter(syn_data, AEBODSYS == "cl A.1", "adae")
run(aet04, proc_data, grade_groups = grade_groups)
```

aet05_all_pre — AET05_ALL *Table 1 (Default) Adverse Event Rate Adjusted for Patient-Years at Risk - All Occurrences.*

Description

The AET05_ALL table produces the standard adverse event rate adjusted for patient-years at risk summary considering all occurrences.

Usage

```
aet05_all_pre(adam_db, dataset = "adsaftte", ...)

aet05_all
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
...	not used.

Value

the preprocessing function returns a list of data.frame.

Functions

- aet05_all_pre(): Preprocessing

Examples

```
library(dplyr)
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "AETOT1" | PARAMCD == "AEREPTTE", "adsaftte")

run(aet05_all, proc_data)

run(aet05_all, proc_data, conf_level = 0.90, conf_type = "exact")
```

aet05_main — AET05 *Table 1 (Default) Adverse Event Rate Adjusted for Patient-Years at Risk - First Occurrence.*

Description

The AET05 table produces the standard adverse event rate adjusted for patient-years at risk summary considering first occurrence.

Usage

```
aet05_main(
  adam_db,
  dataset = "adsaftte",
  arm_var = "ACTARM",
  lbl_overall = NULL,
  ...
)

aet05_pre(adam_db, dataset = "adsaftte", ...)

aet05_post(tlg, prune_0 = FALSE, ...)

aet05
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) the arm variable used for arm splitting.
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
...	Further arguments passed to tern::control_incidence_rate().
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Total patient-years at risk is the sum over all patients of the time intervals (in years).
- Split columns by arm, typically ACTARM.
- Split rows by parameter code.
- AVAL is patient-years at risk.
- N_EVENTS is the number of adverse events observed.
- The table allows confidence level to be adjusted, default is 95%.
- Keep zero count rows by default.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `aet05_main()`: Main TLG function
- `aet05_pre()`: Preprocessing
- `aet05_post()`: Postprocessing

Note

- `adam_db` object must contain table named as `dataset` with the columns `"PARAMCD"`, `"PARAM"`, `"AVAL"`, and `"CNSR"`.

Examples

```
library(dplyr)
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "AETTE1", "adsaftte")

run(aet05, proc_data)

run(aet05, proc_data, conf_level = 0.90, conf_type = "exact")
```

---

aet10_main — `AET10` *Table 1 (Default) Most Common (xx%) Adverse Events Preferred Terms Table 1.*

Description

The `AET10` table Include Adverse Events occurring with user-specified threshold X% in at least one of the treatment groups. Standard table summarized by preferred term (PT). Order the data by total column frequency from most to least frequently reported PT (regardless of SOC).

Usage

```
aet10_main(adam_db, arm_var = "ACTARM", lbl_overall = NULL, ...)

aet10_pre(adam_db, ...)

aet10_post(tlg, atleast = 0.05, ...)

aet10
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
atleast	given cut-off in numeric format, default is 0.05

Details

- Numbers represent absolute numbers of subject and fraction of N, or absolute number of event when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort Dictionary-Derived Code (AEDECOD) by highest overall frequencies.
- Missing values in AEDECOD are labeled by No Coding Available.

Value

the main function returns an rtables object

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- aet10_main(): Main TLG function
- aet10_pre(): Preprocessing
- aet10_post(): Postprocessing

Note

- adam_db object must contain an adae table with the columns "AEDECOD".

Examples

```
run(aet10, syn_data)
```

args_ls — *Get Arguments List*

Description

Get Arguments List

Usage

```
args_ls(x, simplify = FALSE, omit = NULL)

## S4 method for signature 'chevron_tlg'
args_ls(x, simplify = FALSE, omit = NULL)
```

Arguments

x	(chevron_tlg) input.
simplify	(flag) whether to simplify the output, coalescing the values of the parameters. The order of priority for the value of the parameters is: main, preprocess and postprocess.
omit	(character) the names of the argument to omit from the output.

Value

a list of the formal arguments with their default for the functions stored in the chevron_tlg object passed a x argument.

Examples

```
args_ls(aet01, simplify = TRUE)
```

assert_single_value — *Check variable only has one unique value.*

Description

Check variable only has one unique value.

Usage

```
assert_single_value(x, label = deparse(substitute(x)))
```

Arguments

x	value vector.
label	(string) label of input.

Value

invisible NULL or an error message if the criteria are not fulfilled.

---

assert_valid_type *Check variable is of correct type*

Description

Check variable is of correct type

Usage

```
assert_valid_type(x, types, label = deparse(substitute(x)))
```

Arguments

x	Object to check the type.
types	(character) possible types to check.
label	(string) label.

Value

invisible NULL or an error message if the criteria are not fulfilled.

---

assert_valid_var *Check whether var is valid*

Description

Check whether var is valid

Usage

```
assert_valid_var(x, label, na_ok, empty_ok, ...)

## S3 method for class 'character'
assert_valid_var(
  x,
  label = deparse(substitute(x)),
  na_ok = FALSE,
  empty_ok = FALSE,
  min_chars = 1L,
  ...
)
```

```
## S3 method for class 'factor'
assert_valid_var(
  x,
  label = deparse(substitute(x)),
  na_ok = FALSE,
  empty_ok = FALSE,
  min_chars = 1L,
  ...
)

## S3 method for class 'logical'
assert_valid_var(
  x,
  label = deparse(substitute(x)),
  na_ok = TRUE,
  empty_ok = FALSE,
  ...
)

## S3 method for class 'numeric'
assert_valid_var(
  x,
  label = deparse(substitute(x)),
  na_ok = TRUE,
  empty_ok = FALSE,
  integerish = FALSE,
  ...
)

## S3 method for class 'POSIXct'
assert_valid_var(
  x,
  label = deparse(substitute(x)),
  na_ok = TRUE,
  empty_ok = FALSE,
  tzs = OlsonNames(),
  ...
)

## Default S3 method:
assert_valid_var(
  x,
  label = deparse(substitute(x)),
  na_ok = FALSE,
  empty_ok = FALSE,
  ...
)
```

Arguments

x	value of col_split variable
label	(string) hints.
na_ok	(flag) whether NA value is allowed
empty_ok	(flag) whether length 0 value is allowed.
...	Further arguments to methods.
min_chars	(integer) the minimum length of the characters.
integerish	(flag) whether the number should be treated as integerish.
tzs	(character) time zones.

Details

This function checks the variable values are valid or not.

Value

invisible NULL or an error message if the criteria are not fulfilled.

---

assert_valid_var_pair *Check variables are of same levels*

Description

Check variables are of same levels

Usage

```
assert_valid_var_pair(
  df1,
  df2,
  var,
  lab1 = deparse(substitute(df1)),
  lab2 = deparse(substitute(df2))
)
```

Arguments

df1	(data.frame) input.
df2	(data.frame) input.
var	(string) variable to check.
lab1	(string) label hint for df1.
lab2	(string) label hint for df2.

Value

invisible NULL or an error message if the criteria are not fulfilled.

assert_valid_variable *Check variables in a data frame are valid character or factor.*

Description

Check variables in a data frame are valid character or factor.

Usage

```
assert_valid_variable(
  df,
  vars,
  label = deparse(substitute(df)),
  types = NULL,
  ...
)
```

Arguments

df	(data.frame) input dataset.
vars	(character) variables to check.
label	(string) labels of the data frame.
types	Named (list) of type of the input.
...	further arguments for assert_valid_var. Please note that different methods have different arguments so if provided make sure the variables to check is of the same class.

Value

invisible TRUE or an error message if the criteria are not fulfilled.

cfbt01_main CFBT01 *Change from Baseline By Visit Table.*

Description

The CFBT01 table provides an overview of the actual values and its change from baseline of each respective arm over the course of the trial.

Usage

```
cfbt01_main(
  adam_db,
  dataset,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  row_split_var = NULL,
  summaryvars = c("AVAL", "CHG"),
  visitvar = "AVISIT",
  precision = list(default = 2L),
  page_var = "PARAMCD",
  .stats = c("n", "mean_sd", "median", "range"),
  skip = list(CHG = "BASELINE"),
  ...
)

cfbt01_pre(adam_db, dataset, ...)

cfbt01_post(tlg, prune_0 = TRUE, ...)

cfbt01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
row_split_var	(character) additional row split variables.
summaryvars	(character) variables to be analyzed. The label attribute of the corresponding column in table of adam_db is used as label.
visitvar	(string) typically one of "AVISIT" or user-defined visit incorporating "ATPT".
precision	(named list of integer) where names are values found in the PARAMCD column and the values indicate the number of digits in statistics. If default is set, and parameter precision not specified, the value for default will be used. If default is not set, the default precision is 2.
page_var	(string) variable name prior to which the row split is by page.
.stats	(character) statistics names, see tern::analyze_vars().
skip	Named (list) of visit values that need to be inhibited.
...	additional arguments like .indent_mods, .labels.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- The `Analysis Value` column, displays the number of patients, the mean, standard deviation, median and range of the analysis value for each visit.
- The `Change from Baseline` column, displays the number of patient and the mean, standard deviation, median and range of changes relative to the baseline.
- Remove zero-count rows unless overridden with `prune_0 = FALSE`.
- Split columns by arm, typically `ACTARM`.
- Does not include a total column by default.
- Sorted based on factor level; first by `PARAM` labels in alphabetic order then by chronological time point given by `AVISIT`. Re-level to customize order

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `cfbt01_main()`: Main TLG function
- `cfbt01_pre()`: Preprocessing
- `cfbt01_post()`: Postprocessing

Note

- `adam_db` object must contain table named as `dataset` with the columns specified in `summaryvars`.

Examples

```
library(dunlin)

proc_data <- log_filter(
  syn_data,
  PARAMCD %in% c("DIABP", "SYSBP"), "advs"
)
run(cfbt01, proc_data, dataset = "advs")
```

chevron_tlg-class chevron_tlg *class*

Description

The chevron_tlg S4 class associates a preprocess function, a main tlg function and a postprocess function.

chevron_t, a subclass of chevron_tlg with specific validation criteria to handle table creation

chevron_l, a subclass of chevron_tlg with specific validation criteria to handle listing creation

chevron_g, a subclass of chevron_tlg with specific validation criteria to handle graph creation

chevron_simple, a subclass of chevron_tlg, where main function is a simple call

Usage

```
chevron_t(
  main = function(adam_db, ...) build_table(basic_table(), adam_db[[1]]),
  preprocess = function(adam_db, ...) adam_db,
  postprocess = std_postprocessing,
  dataset = NULL,
  ...
)

chevron_l(
  main = function(adam_db, ...) data.frame(),
  preprocess = function(adam_db, ...) adam_db,
  postprocess = std_postprocessing,
  dataset = NULL,
  ...
)

chevron_g(
  main = function(adam_db, ...) ggplot2::ggplot(),
  preprocess = function(adam_db, ...) adam_db,
  postprocess = std_postprocessing,
  dataset = NULL,
  ...
)

chevron_simple()
```

Arguments

main	(function) returning a tlg, with adam_db as first argument. Typically one of the _main function of chevron.
preprocess	(function) returning a pre-processed list of data.frames, with adam_db as first argument. Typically one of the _pre function of chevron.

postprocess	(function) returning a post-processed tlg, with tlg as first argument.
dataset	(string) the name of a table in the adam_db object.
...	not used

Value

a chevron_t class object.

a chevron_l class object.

a chevron_g class object.

a chevron_simple class object.

Slots

main (function) returning a tlg. Typically one of the *_main function from chevron.

preprocess (function) returning a pre-processed list of data.frames amenable to tlg creation. Typically one of the *_pre function from chevron.

postprocess (function) returning a post-processed tlg. Typically one of the *_post function from chevron.

dataset (character) the names of the data sets used in the chevron_tlg object. If NULL, all data sets are possibly used.

Note

To ensure the correct execution of the workflow, additional validation criteria are:

- the first argument of the main function must be adam_db, the input list of data.frames to pre-process. The ... argument is mandatory.
- the first argument of the preprocess function must be adam_db, the input list of data.frames to create tlg output. The ... argument is mandatory.
- the first argument of the postprocess function must be tlg, the input TableTree object to post-process. The ... argument is mandatory.

Examples

```
chevron_t_obj <- chevron_t()
chevron_t_obj <- chevron_t(postprocess = function(tlg, indent, ...) {
  rtables::table_inset(tlg) <- indent
  tlg
})

chevron_l_obj <- chevron_l()

chevron_g_obj <- chevron_g()
chevron_g_obj <- chevron_g(
  postprocess = function(tlg, title, ...) tlg + ggplot2::labs(main = title)
)

chevron_simple_obj <- chevron_simple()
```

cml02a_gl_main — CML02A_GL *Listing 1 (Default) Concomitant Medication Class Level 2, Preferred Name, and Investigator-Specified Terms.*

Description

CML02A_GL Listing 1 (Default) Concomitant Medication Class Level 2, Preferred Name, and Investigator-Specified Terms.

Usage

```
cml02a_gl_main(
  adam_db,
  dataset = "adcm",
  key_cols = c("ATC2", "CMDECOD"),
  disp_cols = c("ATC2", "CMDECOD", "CMTRT"),
  split_into_pages_by_var = NULL,
  unique_rows = TRUE,
  ...
)

cml02a_gl_pre(
  adam_db,
  dataset = "adcm",
  disp_cols = c("ATC2", "CMDECOD", "CMTRT"),
  ...
)

cml02a_gl
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
key_cols	(character) names of columns that should be treated as key columns when rendering the listing. Key columns allow you to group repeat occurrences.
disp_cols	(character) names of non-key columns which should be displayed when the listing is rendered.
split_into_pages_by_var	(character or NULL) the name of the variable to split the listing by.
unique_rows	(flag) whether to keep only unique rows in listing.
...	not used.

Value

the main function returns an rlistings or a list object.

the preprocessing function returns a list of data.frame.

Functions

- `cml02a_gl_main()`: Main TLG function
- `cml02a_gl_pre()`: Preprocessing

Examples

```
run(cml02a_gl, syn_data)
```

---

cmt01_label — CMT01A *Concomitant Medication by Medication Class and Preferred Name.*

Description

A concomitant medication table with the number of subjects and the total number of treatments by medication class.

Usage

```
cmt01_label

cmt01a_main(
  adam_db,
  arm_var = "ARM",
  lbl_overall = NULL,
  row_split_var = "ATC2",
  medname_var = "CMDECOD",
  summary_labels = setNames(rep(list(cmt01_label), length(row_split_var) + 1L),
    c("TOTAL", row_split_var)),
  ...
)

cmt01a_pre(adam_db, ...)

cmt01a_post(
  tlg,
  prune_0 = TRUE,
  sort_by_freq = FALSE,
  row_split_var = "ATC2",
  medname_var = "CMDECOD",
  ...
)

cmt01a
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
row_split_var	(character) the variable defining the medication category. By default ATC2.
medname_var	(string) variable name of medical treatment name.
summary_labels	(list) of summarize labels. See details.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows
sort_by_freq	(flag) whether to sort medication class by frequency.

Details

- Data should be filtered for concomitant medication. (ATIREL == "CONCOMITANT").
- Numbers represent absolute numbers of subjects and fraction of N, or absolute numbers when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort by medication class alphabetically and within medication class by decreasing total number of patients with the specific medication. summary_labels is used to control the summary for each level. If "all" is used, then each split will have that summary statistic with the labels. One special case is "TOTAL", this is for the overall population.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- cmt01_label: Default labels
- cmt01a_main(): Main TLG function
- cmt01a_pre(): Preprocessing
- cmt01a_post(): Postprocessing

Note

- adam_db object must contain an adcm table with the columns specified in row_split_var and medname_var as well as "CMSEQ".

Examples

```
library(dplyr)

proc_data <- syn_data
proc_data$adcm <- proc_data$adcm %>%
  filter(ATIREL == "CONCOMITANT")

run(cmt01a, proc_data)
```

`cmt02_pt_main` `CMT02_PT` *Table 1 (Default) Concomitant Medications by Preferred Name.*

Description

A concomitant medication table with the number of subjects and the total number of treatments by medication name sorted by frequencies.

Usage

```
cmt02_pt_main(
  adam_db,
  arm_var = "ARM",
  lbl_overall = NULL,
  row_split_var = NULL,
  medname_var = "CMDECOD",
  summary_labels = list(TOTAL = cmt01_label),
  ...
)

cmt02_pt_pre(adam_db, ...)

cmt02_pt_post(
  tlg,
  prune_0 = TRUE,
  sort_by_freq = FALSE,
  row_split_var = NULL,
  medname_var = "CMDECOD",
  ...
)

cmt02_pt
```

Arguments

`adam_db`	(`list` of `data.frames`) object containing the `ADaM` datasets
`arm_var`	(`string`) variable used for column splitting

lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
row_split_var	(character) the variable defining the medication category. By default ATC2.
medname_var	(string) variable name of medical treatment name.
summary_labels	(list) of summarize labels. See details.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows
sort_by_freq	(flag) whether to sort medication class by frequency.

Details

- Data should be filtered for concomitant medication. (ATIREL == "CONCOMITANT").
- Numbers represent absolute numbers of subjects and fraction of N, or absolute numbers when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort by medication class alphabetically and within medication class by decreasing total number of patients with the specific medication. summary_labels is used to control the summary for each level. If "all" is used, then each split will have that summary statistic with the labels. One special case is "TOTAL", this is for the overall population.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- cmt02_pt_main(): Main TLG function
- cmt02_pt_pre(): Preprocessing
- cmt02_pt_post(): Postprocessing

Note

- adam_db object must contain an adcm table with the columns specified in row_split_var and medname_var as well as "CMSEQ".

Examples

```
run(cmt02_pt, syn_data)
```

convert_to_month *Helper function to convert to months if needed*

Description

Helper function to convert to months if needed

Usage

```
convert_to_month(x, unit)
```

Arguments

x	(numeric) time.
unit	(character) or (factor) time unit.

Value

A numeric vector with the time in months.

coxt01_main COXT01 *(Default) Cox Regression Model Table.*

Description

Cox models are the most commonly used methods to estimate the magnitude of the effect in survival analyses. It assumes proportional hazards; that is, it assumes that the ratio of the hazards of the two groups (e.g. two arms) is constant over time. This ratio is referred to as the "hazard ratio" and is one of the most commonly reported metrics to describe the effect size in survival analysis.

Usage

```
coxt01_main(
  adam_db,
  arm_var = "ARM",
  time_var = "AVAL",
  event_var = "EVENT",
  covariates = c("SEX", "RACE", "AAGE"),
  strata = NULL,
  lbl_vars = "Effect/Covariate Included in the Model",
  multivar = FALSE,
  ...
)

coxt01_pre(adam_db, arm_var = "ARM", ...)
```

```
coxt01_post(tlg, prune_0 = FALSE, ...)

coxt01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) the arm variable used for arm splitting.
time_var	(string) the time variable in a Cox proportional hazards regression model.
event_var	(string) the event variable in a Cox proportional hazards regression model.
covariates	(character) will be fitted and the corresponding effect will be estimated.
strata	(character) will be fitted for the stratified analysis.
lbl_vars	(string) text label for the a Cox regression model variables.
multivar	(flag) indicator of whether multivariate cox regression is conducted.
...	Further arguments passed to tern::control_coxreg().
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- The reference arm will always the first level of arm_var. Please change the level if you want to change the reference arms.
- The table allows confidence level to be adjusted, default is two-sided 95%.
- The stratified analysis is with DISCRETE tie handling (equivalent to tern::control_coxreg(ties = "exact") in R).
- Model includes treatment plus specified covariate(s) as factor(s) or numeric(s), with "SEX", "RACE" and "AAGE" as default candidates.
- The selection of the covariates and whether or not there is a selection process (vs. a fixed, pre-specified list) needs to be pre-specified.
- For pairwise comparisons using the hazard ratio, the value for the control group is the denominator.
- Keep zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- coxt01_main(): Main TLG function
- coxt01_pre(): Preprocessing
- coxt01_post(): Postprocessing

Note

- adam_db object must contain an adtte table with "PARAMCD", "ARM", "AVAL", "CNSR, and the columns specified by "covariates" which is denoted as c("SEX", "RACE", "AAGE") by default.

Examples

```
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "CRSD", "adtte")
proc_data <- log_filter(proc_data, ARMCD != "ARM C", "adsl")
run(coxt01, proc_data)

run(coxt01, proc_data, covariates = c("SEX", "AAGE"), strata = c("RACE"), conf_level = 0.90)
```

---

coxt02_main COXT02 *Multi-Variable Cox Regression Model Table.*

Description

The COXT02 table follows the same principles as the general Cox model analysis and produces the estimates for each of the covariates included in the model (usually the main effects without interaction terms).

Usage

```
coxt02_main(
  adam_db,
  arm_var = "ARM",
  time_var = "AVAL",
  event_var = "EVENT",
  covariates = c("SEX", "RACE", "AAGE"),
  strata = NULL,
  lbl_vars = "Effect/Covariate Included in the Model",
  multivar = TRUE,
  ...
)

coxt02
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) the arm variable used for arm splitting.
time_var	(string) the time variable in a Cox proportional hazards regression model.
event_var	(string) the event variable in a Cox proportional hazards regression model.

covariates	(character) will be fitted and the corresponding effect will be estimated.
strata	(character) will be fitted for the stratified analysis.
lbl_vars	(string) text label for the a Cox regression model variables.
multivar	(flag) indicator of whether multivariate cox regression is conducted.
...	Further arguments passed to tern::control_coxreg().

Details

- The reference arm will always the first level of arm_var. Please change the level if you want to change the reference arms.
- The table allows confidence level to be adjusted, default is two-sided 95%.
- The stratified analysis is with DISCRETE tie handling (equivalent to tern::control_coxreg(ties = "exact") in R).
- Model includes treatment plus specified covariate(s) as factor(s) or numeric(s), with "SEX", "RACE" and "AAGE" as default candidates.
- The selection of the covariates and whether or not there is a selection process (vs. a fixed, pre-specified list) needs to be pre-specified.
- For pairwise comparisons using the hazard ratio, the value for the control group is the denominator.
- Keep zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object.

Functions

- coxt02_main(): Main TLG function

Note

- adam_db object must contain an adtte table with "PARAMCD", "ARM", "AVAL", "CNSR, and the columns specified by "covariates" which is denoted as c("SEX", "RACE", "AAGE") by default.

Examples

```
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "CRSD", "adtte")

run(coxt02, proc_data)

run(coxt02, proc_data, covariates = c("SEX", "AAGE"), strata = c("RACE"), conf_level = 0.90)
```

create_id_listings *Concatenate Site and Subject ID*

Description

Concatenate Site and Subject ID

Usage

```
create_id_listings(site, subject, sep = "/")
```

Arguments

site	(string)
subject	(string)
sep	(string)

Note

the {Patient_label} whisker placeholder will be used in the label.

Examples

```
create_id_listings("BRA-1", "xxx-1234")
```

ctcv4_dir *CTC version 4 Grade Direction Data*

Description

CTC version 4 Grade Direction Data

Usage

```
ctcv4_dir
```

Format

An object of class data.frame with 35 rows and 3 columns.

ctcv5_dir — *CTC version 5 Grade Direction Data*

Description

CTC version 5 Grade Direction Data

Usage

```
ctcv5_dir
```

Format

An object of class `data.frame` with 35 rows and 3 columns.

dataset — *Data Set*

Description

retrieve or set `dataset` slot.

Usage

```
dataset(x)

## S4 method for signature 'chevron_tlg'
dataset(x)

dataset(x) <- value

## S4 replacement method for signature 'chevron_tlg'
dataset(x) <- value
```

Arguments

`x`	(`chevron_tlg`) input.
`value`	(`character` or `NULL`) defining the dataset slot.

Value

the `character` stored in the `dataset` slot of the `x` argument.

dmt01_main DMT01 *Table 1 (Default) Demographics and Baseline Characteristics Table 1.*

Description

For each variable, summary statistics are by default based on the number of patients in the corresponding n row.

Usage

```
dmt01_main(
  adam_db,
  arm_var = "ARM",
  lbl_overall = "All {Patient_label}",
  summaryvars = c("AAGE", "AGEGR1", "SEX", "ETHNIC", "RACE"),
  stats = list(default = c("n", "mean_sd", "median", "range", "count_fraction")),
  precision = list(),
  ...
)

dmt01_pre(adam_db, ...)

dmt01_post(tlg, prune_0 = TRUE, ...)

dmt01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
summaryvars	(character) variables summarized in demographic table. The label attribute of the corresponding column in adsl table of adam_db is used as label.
stats	(named list of character) where names are strings found in summaryvars and the values indicate the statistical analysis to perform. If default is set, and parameter precision not specified, the value for default will be used.
precision	(named list of integer) where names are strings found in summaryvars and the values indicate the number of digits in statistics for numeric variables. If default is set, and parameter precision not specified, the value for default will be used. If neither are provided, auto determination is used. See tern::format_auto.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Information from ADSUB are generally included into ADSL before analysis.
- Default demographic and characteristics table
- If not specified otherwise, numbers represent absolute numbers of patients and fraction of N
- Remove zero-count rows
- Split columns by arm (planned or actual / code or description)
- Include a total column by default

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `dmt01_main()`: Main TLG function
- `dmt01_pre()`: Preprocessing
- `dmt01_post()`: Postprocessing

Note

- `adam_db` object must contain an `adsl` table with the columns specified in `summaryvars`.

Examples

```
run(dmt01, syn_data)
```

---

dose_change_rule *Dose Change Rule*

Description

Dose Change Rule

Usage

```
dose_change_rule
```

dst01_main *DST01 Table 1 (Default) Patient Disposition Table 1.*

Description

The DST01 Disposition Table provides an overview of patients study completion. For patients who discontinued the study a reason is provided.

Usage

```
dst01_main(
  adam_db,
  arm_var = "ARM",
  lbl_overall = "All {Patient_label}",
  study_status_var = "EOSSTT",
  detail_vars = list(Discontinued = c("DCSREAS")),
  trt_status_var = NULL,
  ...
)

dst01_pre(adam_db, ...)

dst01_post(tlg, prune_0 = TRUE, ...)

dst01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable. Usually one of ARM, ACTARM, TRT01A, or TRT01A.
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
study_status_var	(string) variable used to define patient status. Default is EOSSTT, however can also be a variable name with the pattern EOPxxSTT where xx must be substituted by 2 digits referring to the analysis period.
detail_vars	Named (list) of grouped display of study_status_var. The names must be subset of unique levels of study_status_var.
trt_status_var	(string) variable of treatment status.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Default patient disposition table summarizing the reasons for patients withdrawal.
- Numbers represent absolute numbers of patients and fraction of `N`.
- Remove zero-count rows.
- Split columns by arm.
- Include a total column by default.
- Sort withdrawal reasons by alphabetic order.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `dst01_main()`: Main TLG function
- `dst01_pre()`: Preprocessing
- `dst01_post()`: Postprocessing

Note

- `adam_db` object must contain an `adsl` table with the columns specified by `status_var` and `disc_reason_var`.

Examples

```
run(dst01, syn_data, detail_vars = list(Ongoing = "STDONS"))

run(dst01, syn_data, detail_vars = list(Discontinued = "DCSREAS", Ongoing = "STDONS"))

run(
  dst01, syn_data,
  detail_vars = list(
    Discontinued = c("DCSREASGP", "DCSREAS"),
    Ongoing = "STDONS"
  )
)
```

dtht01_main DTHT01 *Table 1 (Default) Death Table.*

Description

A description of the causes of death optionally with the breakdown of the OTHER category and/or post-study reporting of death.

Usage

```
dtht01_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  other_category = FALSE,
  time_since_last_dose = FALSE,
  ...
)

dtht01_pre(adam_db, ...)

dtht01_post(tlg, prune_0 = TRUE, ...)

dtht01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
other_category	(flag) should the breakdown of the OTHER category be displayed.
time_since_last_dose	(flag) should the time to event information be displayed.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Numbers represent absolute numbers of subjects and fraction of N, or absolute numbers when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Does not include a total column by default.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `dtht01_main()`: Main TLG function
- `dtht01_pre()`: Preprocessing
- `dtht01_post()`: Postprocessing

Note

- `adam_db` object must contain an `adsl` table with the columns `"DTHFL"`, `"DTHCAT"` as well as `LDDTHGR1` if `time_since_last_dose` is `TRUE`.

Examples

```
run(dtht01, syn_data)

run(dtht01, syn_data, other_category = TRUE, time_since_last_dose = TRUE)
```

---

dummy_template *Dummy template.*

Description

This template creates a dummy output.

Usage

```
dummy_template
```

Examples

```
run(dummy_template, syn_data)
```

egt01_main EGT01 *ECG Parameters and Change from Baseline By Visit Table.*

Description

The EGT01 table provides an overview of the ECG values and its change from baseline of each respective arm over the course of the trial.

Usage

```
egt01_main(
  adam_db,
  dataset = "adeg",
  arm_var = "ACTARM",
  lbl_overall = NULL,
  row_split_var = NULL,
  summaryvars = c("AVAL", "CHG"),
  visitvar = "AVISIT",
  precision = list(default = 0L),
  page_var = "PARAMCD",
  .stats = c("n", "mean_sd", "median", "range"),
  skip = list(CHG = "BASELINE"),
  ...
)

egt01_pre(adam_db, dataset = "adeg", ...)

egt01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
row_split_var	(character) additional row split variables.
summaryvars	(character) variables to be analyzed. The label attribute of the corresponding column in table of adam_db is used as label.
visitvar	(string) typically one of "AVISIT" or user-defined visit incorporating "ATPT".
precision	(named list of integer) where names are values found in the PARAMCD column and the values indicate the number of digits in statistics. If default is set, and parameter precision not specified, the value for default will be used. If default is not set, the default precision is 2.
page_var	(string) variable name prior to which the row split is by page.

`.stats`	(`character`) statistics names, see `tern::analyze_vars()`.
`skip`	Named (`list`) of visit values that need to be inhibited.
`...`	additional arguments like `.indent_mods`, `.labels`.

Details

- The `Analysis Value` column, displays the number of patients, the mean, standard deviation, median and range of the analysis value for each visit.
- The `Change from Baseline` column, displays the number of patient and the mean, standard deviation, median and range of changes relative to the baseline.
- Remove zero-count rows unless overridden with `prune_0 = FALSE`.
- Split columns by arm, typically `ACTARM`.
- Does not include a total column by default.
- Sorted based on factor level; first by `PARAM` labels in alphabetic order then by chronological time point given by `AVISIT`. Re-level to customize order

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

Functions

- `egt01_main()`: Main TLG function
- `egt01_pre()`: Preprocessing

Note

default precision is 0.

Examples

```
run(egt01, syn_data)
```

`egt02_1_main` `EGT02` *ECG Abnormalities Table.*

Description

ECG Parameters outside Normal Limits Regardless of Abnormality at Baseline Table.

Usage

```
egt02_1_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  exclude_base_abn = FALSE,
  ...
)

egt02_pre(adam_db, ...)

egt02_post(tlg, ...)

egt02_1
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
exclude_base_abn	(flag) whether baseline abnormality should be excluded.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.

Details

- Only count LOW or HIGH values.
- Results of "LOW LOW" are treated as the same as "LOW", and "HIGH HIGH" the same as "HIGH".
- Does not include a total column by default.
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- egt02_1_main(): Main TLG function
- egt02_pre(): Preprocessing
- egt02_post(): Postprocessing

Note

- adam_db object must contain an adeg table with the "PARAM", "ANRIND" and "BNRIND" columns.

Examples

```
run(egt02_1, syn_data)
```

---

egt02_2_main EGT02_2 *ECG Abnormalities Table.*

Description

ECG Parameters outside Normal Limits Among Patients without Abnormality at Baseline Table.

Usage

```
egt02_2_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  exclude_base_abn = TRUE,
  ...
)

egt02_2
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
exclude_base_abn	(flag) whether baseline abnormality should be excluded.
...	not used.

Details

- Only count LOW or HIGH values.
- Results of "LOW LOW" are treated as the same as "LOW", and "HIGH HIGH" the same as "HIGH".
- Does not include a total column by default.
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an `rtables` object

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `egt02_2_main()`: Main TLG function

Note

- `adam_db` object must contain an `adeg` table with the `"PARAM"`, `"ANRIND"` and `"BNRIND"` columns.

Examples

```
run(egt02_2, syn_data)
```

---

egt03_main — EGT03 *Shift Table of ECG Interval Data - Baseline versus Minimum or Maximum Post-Baseline.*

Description

The `EGT03` Table entries provide the number of patients by baseline assessment and minimum or maximum post-baseline assessment. Percentages are based on the total number of patients in a treatment group. Baseline is the patient's last observation prior to initiation of study drug.

Usage

```
egt03_main(
  adam_db,
  arm_var = "ACTARMCD",
  summaryvar = "BNRIND",
  splitvar = "ANRIND",
  visitvar = "AVISIT",
  page_var = "PARAMCD",
  ...
)

egt03_pre(adam_db, ...)

egt03_post(tlg, prune_0 = FALSE, ...)

egt03
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(character) the arm variables used for row split, typically "ACTARMCD".
summaryvar	(character) variables to be analyzed, typically "BNRIND". Labels of the corresponding columns are used as subtitles.
splitvar	(character) variables to be analyzed, typically "ANRIND". Labels of the corresponding columns are used as subtitles.
visitvar	(string) typically "AVISIT" or user-defined visit incorporating "ATPT".
page_var	(string) variable name prior to which the row split is by page.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- ADEG data are subsetted to contain only "POST-BASELINE MINIMUM"/"POST-BASELINE MAXIMUM" visit according to the preprocessing.
- Percentages are based on the total number of patients in a treatment group.
- Split columns by Analysis Reference Range Indicator, typically ANRIND.
- Does not include a total column by default.
- Sorted based on factor level.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- egt03_main(): Main TLG function
- egt03_pre(): Preprocessing
- egt03_post(): Postprocessing

Note

- adam_db object must contain an adeg table with a "ACTARMCD" column as well as columns specified in summaryvar and splitvar.

Examples

```
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "HR", "adeg")
run(egt03, proc_data)
```

egt05_qtcat_main EGT05_QTCAT *ECG Actual Values and Changes from Baseline by Visit Table.*

Description

The EGT05_QTCAT table summarizes several electrocardiogram parameters and their evolution throughout the study.

Usage

```
egt05_qtcat_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  summaryvars = c("AVALCAT1", "CHGCAT1"),
  row_split_var = NULL,
  visitvar = "AVISIT",
  page_var = NULL,
  ...
)

egt05_qtcat_pre(adam_db, ...)

egt05_qtcat_post(tlg, prune_0 = TRUE, ...)

egt05_qtcat
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
summaryvars	(character) variables to be analyzed. The label attribute of the corresponding column in adeg table of adam_db is used as name.
row_split_var	(character) additional row split variables.
visitvar	(string) typically "AVISIT" or user-defined visit incorporating "ATPT".
page_var	(string) variable name prior to which the row split is by page.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- The `Value at Visit` column, displays the categories of the specific `"PARAMCD"` value for patients.
- The `Change from Baseline` column, displays the categories of the specific `"PARAMCD"` value change from baseline for patients.
- Remove zero-count rows unless overridden with `prune_0 = FALSE`.
- Split columns by arm, typically `"ACTARM"`.
- Does not include a total column by default.
- Sorted based on factor level; by chronological time point given by `"AVISIT"` or user-defined visit incorporating `"ATPT"`. Re-level to customize order.
- Please note that it is preferable to convert `summaryvars` to factor.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `egt05_qtcat_main()`: Main TLG function
- `egt05_qtcat_pre()`: Preprocessing
- `egt05_qtcat_post()`: Postprocessing

Note

- `adam_db` object must contain an `adeg` table with column specified in `visitvar`. For `summaryvars`, please make sure `AVALCAT1` and `CHGCAT1` columns existed in input data sets.

Examples

```
run(egt05_qtcat, syn_data)
```

---

empty_rule *Empty rule*

Description

Empty rule

Usage

```
empty_rule
```

ext01_main EXT01 *Exposure Summary Table.*

Description

The EXT01 table provides an overview of the of the exposure of the patients in terms of Total dose administered or missed, and treatment duration.

Usage

```
ext01_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  summaryvars = "AVAL",
  row_split_var = "PARCAT2",
  page_var = NULL,
  map = NULL,
  stats = list(default = c("n", "mean_sd", "median", "range", "count_fraction")),
  precision = list(default = 0),
  ...
)

ext01_pre(adam_db, ...)

ext01_post(tlg, prune_0 = TRUE, ...)

ext01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
summaryvars	(character) variables to be analyzed. The label attribute of the corresponding column in adex table of adam_db is used as label.
row_split_var	(character) additional row split variables.
page_var	(string) variable name prior to which the row split is by page.
map	(data.frame) of mapping for split rows.
stats	(named list of character) where names are values found in the PARAMCD column and the values indicate the statistical analysis to perform. If default is set, and parameter precision not specified, the value for default will be used.
precision	(named list of integer) where names are values found in the PARAMCD column and the values

...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Default Exposure table
- The n row provides the number of non-missing values. The percentages for categorical variables is based on n. The percentages for Total number of patients with at least one dose modification are based on the number of patients in the corresponding analysis population given by N.
- Split columns by arm, typically ACTARM.
- Does not include a total column by default.
- Sorted by alphabetic order of the PARAM value. Transform to factor and re-level for custom order.
- ANL01FL is not relevant subset.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- ext01_main(): Main TLG function
- ext01_pre(): Preprocessing
- ext01_post(): Postprocessing

Note

- adam_db object must contain an adex table with columns specified in summaryvars.

Examples

```
run(ext01, syn_data)

run(ext01, syn_data, summaryvars = c("AVAL", "AVALCAT1"), prune_0 = FALSE)

levels(syn_data$adex$AVALCAT1) <- c(levels(syn_data$adex$AVALCAT1), "12 months")
map <- data.frame(
  PARAMCD = "TDURD",
  AVALCAT1 = c("< 1 month", "1 to <3 months", ">=6 months", "3 to <6 months", "12 months")
)
run(
  ext01,
  syn_data,
  summaryvars = c("AVAL", "AVALCAT1"),
```

```
  prune_0 = FALSE,
  map = map,
  precision = list(TDOSE = 4, default = 4),
  stats = list(TDURD = "n", default = c("n", "count_fraction"))
)
```

`format_date` *Formatting of date*

Description

Formatting of date

Usage

```
format_date(date_format = "%d%b%Y")
```

Arguments

`date_format`	(`string`) the output format.

Value

a `function` converting a date into `string`.

Note

The date is extracted at the location of the measure, not at the location of the system.

Examples

```
format_date("%d%b%Y")(as.Date("2021-01-01"))
if ("NZ" %in% OlsonNames()) {
  format_date("%d%b%Y")(as.POSIXct("2021-01-01 00:00:01", tz = "NZ"))
}
if ("US/Pacific" %in% OlsonNames()) {
  format_date("%d%b%Y")(as.POSIXct("2021-01-01 00:00:01", tz = "US/Pacific"))
}
```

fstg01_main FSTG01 *Subgroup Analysis of Best Overall Response.*

Description

The template produces the subgroup analysis of best overall response graphic.

Usage

```
fstg01_main(
  adam_db,
  dataset = "adrs",
  arm_var = "ARM",
  rsp_var = "IS_RSP",
  subgroups = c("SEX", "AGEGR1", "RACE"),
  strata_var = NULL,
  stat_var = c("n_tot", "n", "n_rsp", "prop", "or", "ci"),
  ...
)

fstg01_pre(adam_db, ...)

fstg01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) the arm variable name used for group splitting.
rsp_var	(string) the response variable name to flag whether each subject is a binary response or not.
subgroups	(character) the subgroups variable name to list baseline risk factors.
strata_var	(character) required if stratified analysis is performed.
stat_var	(character) the names of statistics to be reported in tabulate_rsp_subgroups.
...	Further arguments passed to g_forest and extract_rsp_subgroups (a wrapper for h_odds_ratio_subgroups_df and h_proportion_subgroups_df). For details, see the documentation in tern. Commonly used arguments include col_symbol_size, col, vline, groups_lists, conf_level, method, label_all, etc.

Details

- No overall value.
- Keep zero count rows by default.

Value

the main function returns a grob object.

a gTree object.

the preprocessing function returns a list of data.frame.

Functions

- fstg01_main(): Main TLG Function
- fstg01_pre(): Preprocessing

Note

- adam_db object must contain the table specified by dataset with "PARAMCD", "ARM", "AVALC", and the columns specified by subgroups which is denoted as c("SEX", "AGEGR1", "RACE") by default.
- If the plot is too large to be rendered in the output, please provide gp, width_row_names, width_columns and width_forest manually to make it fit. See tern::g_forest for more details.

Examples

```
library(dplyr)
library(dunlin)

proc_data <- log_filter(
  syn_data,
  PARAMCD == "BESRSPI" & ARM %in% c("A: Drug X", "B: Placebo"), "adrs"
)
run(fstg01, proc_data,
  subgroups = c("SEX", "AGEGR1", "RACE"),
  conf_level = 0.90, dataset = "adrs"
)
```

---

fstg02_main FSTG02 *Subgroup Analysis of Survival Duration.*

Description

The template produces the subgroup analysis of survival duration graphic.

Usage

```
fstg02_main(
  adam_db,
  dataset = "adtte",
  arm_var = "ARM",
  subgroups = c("SEX", "AGEGR1", "RACE"),
```

```
  strata_var = NULL,
  stat_var = c("n_tot", "n", "median", "hr", "ci"),
  ...
)

fstg02_pre(adam_db, ...)

fstg02
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) the arm variable name used for group splitting.
subgroups	(character) the subgroups variable name to list baseline risk factors.
strata_var	(character) required if stratified analysis is performed.
stat_var	(character) the names of statistics to be reported in tabulate_survival_subgroups.
...	Further arguments passed to g_forest and extract_rsp_subgroups (a wrapper for h_odds_ratio_subgroups_df and h_proportion_subgroups_df). For details, see the documentation in tern. Commonly used arguments include gp, col_symbol_size, col, vline, groups_lists, conf_level, method, label_all, etc.

Details

- No overall value.
- Keep zero count rows by default.

Value

the main function returns a gTree object.

a gTree object.

the preprocessing function returns a list of data.frame.

Functions

- fstg02_main(): Main TLG Function
- fstg02_pre(): Preprocessing

Note

- adam_db object must contain the table specified by dataset with "PARAMCD", "ARM", "AVAL", "AVALU", "CNSR", and the columns specified by subgroups which is denoted as c("SEX", "AGEGR1", "RACE") by default.
- If the plot is too large to be rendered in the output, please refer to FSTG01.

Examples

```
library(dplyr)
library(dunlin)

proc_data <- log_filter(
  syn_data,
  PARAMCD == "OS" & ARM %in% c("A: Drug X", "B: Placebo"), "adtte"
)
run(fstg02, proc_data,
  subgroups = c("SEX", "AGEGR1", "RACE"),
  conf_level = 0.90, dataset = "adtte"
)
```

---

gen_args *General Argument Name Convention*

Description

General Argument Name Convention

Usage

```
gen_args(
  adam_db,
  main,
  preprocess,
  postprocess,
  dataset,
  type,
  arm_var,
  lbl_overall,
  prune_0,
  req_tables,
  deco,
  group,
  tlg,
  visitvar,
  visit_value,
  paramcd_value,
  key_cols,
  disp_cols,
  row_split_var,
  split_into_pages_by_var,
  page_var,
  unique_rows,
  ...
)
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
main	(function) returning a tlg, with adam_db as first argument. Typically one of the _main function of chevron.
preprocess	(function) returning a pre-processed list of data.frames, with adam_db as first argument. Typically one of the _pre function of chevron.
postprocess	(function) returning a post-processed tlg, with tlg as first argument.
dataset	(string) the name of a table in the adam_db object.
type	(string) indicating the subclass.
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
prune_0	(flag) remove 0 count rows
req_tables	(character) names of the required tables.
deco	(character) decoration with title, subtitles and main_footer content
group	(list of lists) for group-dependent data binning
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
visitvar	(string) typically "AVISIT" or user-defined visit incorporating "ATPT".
visit_value	Value of visit variable.
paramcd_value	Value of PARAMCD variable.
key_cols	(character) names of columns that should be treated as key columns when rendering the listing. Key columns allow you to group repeat occurrences.
disp_cols	(character) names of non-key columns which should be displayed when the listing is rendered.
row_split_var	(character) additional row split variables.
split_into_pages_by_var	(character or NULL) the name of the variable to split the listing by.
page_var	(string) variable name prior to which the row split is by page.
unique_rows	(flag) whether to keep only unique rows in listing.
...	not used.

Details

the following arguments are better provided through the study object: lbl_overall, arm_var.

Value

invisible NULL. This function is for documentation purpose only.

get_grade_rule *Get grade rule*

Description

Get grade rule

Usage

```
get_grade_rule(direction = "high", missing = "incl")
```

Arguments

direction	(string) of abnormality direction.
missing	(string) method to deal with missing

Value

a rule object.

get_section_div *Get Section dividers*

Description

Get Section dividers

Usage

```
get_section_div()
```

Value

(character) value with section dividers at corresponding section.

gg_list *List of* gg *object*

Description

[Deprecated]

Usage

```
gg_list(...)
```

Arguments

...	(ggplot) objects.

Value

a gg_list object.

gg_theme_chevron *Theme for Chevron Plot*

Description

Theme for Chevron Plot

Usage

```
gg_theme_chevron(
  grid_y = TRUE,
  grid_x = FALSE,
  legend_position = "top",
  text_axis_x_rot = 45
)
```

Arguments

grid_y	(flag) should horizontal grid be displayed.
grid_x	(flag) should vertical grid be displayed.
legend_position	(string) the position of the legend.
text_axis_x_rot	(numeric) the x axis text rotation angle.

Value

a theme object.

grob_list — *List of* grob *object*

Description

[Deprecated]

Usage

```
grob_list(...)
```

Arguments

...	(grob) objects.

Value

a grob_list object.

h_format_dec — *Decimal formatting*

Description

Decimal formatting

Usage

```
h_format_dec(digits, format, ne = NULL)
```

Arguments

digits	(integer) number of digits.
format	(string) describing how the numbers should be formatted following the sprintf syntax.
ne	(string) that should replace actual value. If NULL, no replacement is performed.

Value

function formatting numbers with the defined format.

Examples

```
fun <- h_format_dec(c(1, 1), "%s - %s")
fun(c(123, 567.89))
```

kmg01_main KMG01 *Kaplan-Meier Plot 1.*

Description

KMG01 Kaplan-Meier Plot 1.

Usage

```
kmg01_main(
  adam_db,
  dataset = "adtte",
  arm_var = "ARM",
  strata = NULL,
  strat = lifecycle::deprecated(),
  ...
)

kmg01_pre(adam_db, dataset = "adtte", ...)

kmg01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) variable used for column splitting
strata	(character) the variable name of stratification variables.
strat	(character) **[Deprecated]**; for backwards compatibility only. Use strata instead.
...	Further arguments passed to g_km and control_coxph. For details, see the documentation in tern. Commonly used arguments include col, pval_method, ties, conf_level, conf_type, annot_coxph, annot_stats, etc.

Details

- No overall value.

Value

the main function returns a gTree object.

a gTree object.

the preprocessing function returns a list of data.frame.

Functions

- kmg01_main(): Main TLG Function
- kmg01_pre(): Preprocessing

Note

- adam_db object must contain the table specified by dataset with the columns specified by arm_var.

Examples

```
library(dplyr)
library(dunlin)

col <- c(
  "A: Drug X" = "black",
  "B: Placebo" = "blue",
  "C: Combination" = "gray"
)

pre_data <- log_filter(syn_data, PARAMCD == "OS", "adtte")
run(kmg01, pre_data, dataset = "adtte", col = col)
```

---

lbt01_main LBT01 *Lab Results and Change from Baseline by Visit Table.*

Description

The LBT01 table provides an overview of the Lab values and its change from baseline of each respective arm over the course of the trial.

Usage

```
lbt01_main(
  adam_db,
  dataset = "adlb",
  arm_var = "ACTARM",
  lbl_overall = NULL,
  row_split_var = NULL,
  summaryvars = c("AVAL", "CHG"),
  visitvar = "AVISIT",
  precision = lab_paramcd_precision(),
  page_var = "PARAMCD",
  .stats = c("n", "mean_sd", "median", "range"),
  skip = list(CHG = "BASELINE"),
  ...
)
```

```
lbt01_pre(adam_db, dataset = "adlb", ...)

lbt01

lab_paramcd_precision()
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
row_split_var	(character) additional row split variables.
summaryvars	(character) variables to be analyzed. The label attribute of the corresponding column in table of adam_db is used as label.
visitvar	(string) typically one of "AVISIT" or user-defined visit incorporating "ATPT".
precision	(named list of integer) where names are values found in the PARAMCD column and the values indicate the number of digits in statistics. If default is set, and parameter precision not specified, the value for default will be used. If default is not set, the default precision is 2.
page_var	(string) variable name prior to which the row split is by page.
.stats	(character) statistics names, see tern::analyze_vars().
skip	Named (list) of visit values that need to be inhibited.
...	additional arguments like .indent_mods, .labels.

Details

- The Analysis Value column, displays the number of patients, the mean, standard deviation, median and range of the analysis value for each visit.
- The Change from Baseline column, displays the number of patient and the mean, standard deviation, median and range of changes relative to the baseline.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm, typically ACTARM.
- Does not include a total column by default.
- Sorted based on factor level; first by PARAM labels in alphabetic order then by chronological time point given by AVISIT. Re-level to customize order

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

a named list with the precision of each lab parameter code (default is 2).

Functions

- `lbt01_main()`: Main TLG function
- `lbt01_pre()`: Preprocessing
- `lab_paramcd_precision()`: Default lab parameter precision

Note

- `adam_db` object must contain table named as `dataset` with the columns specified in `summaryvars`.

Examples

```
run(lbt01, syn_data)
# example code
head(lab_paramcd_precision())
```

---

lbt04_main — LBT04 *Laboratory Abnormalities Not Present at Baseline Table.*

Description

The `LBT04` table provides an overview of laboratory abnormalities not present at baseline.

Usage

```
lbt04_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  analysis_abn_var = "ANRIND",
  baseline_abn_var = "BNRIND",
  row_split_var = "PARCAT1",
  page_var = tail(row_split_var, 1L),
  ...
)

lbt04_pre(adam_db, ...)

lbt04_post(tlg, ...)

lbt04
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
analysis_abn_var	(string) column describing anomaly magnitude
baseline_abn_var	(string) column describing anomaly at baseline.
row_split_var	(character) additional row split variables.
page_var	(string) variable name prior to which the row split is by page.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.

Details

- Only count LOW or HIGH values.
- Lab test results with missing analysis_abn_var values are excluded.
- Split columns by arm, typically ACTARM.
- Does not include a total column by default.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- lbt04_main(): Main TLG function
- lbt04_pre(): Preprocessing
- lbt04_post(): Postprocessing

Note

- adam_db object must contain an adlb table with columns "PARCAT1", "PARCAT2", "PARAM", "ANRIND", and column specified by arm_var.

Examples

```
run(lbt04, syn_data)
```

lbt05_main — LBT05 *Table 1 (Default) Laboratory Abnormalities with Single and Replicated Marked.*

Description

LBT05 Table 1 (Default) Laboratory Abnormalities with Single and Replicated Marked.

Usage

```
lbt05_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  map = lab_paramcd_abn_dir(),
  ...
)

lbt05_pre(adam_db, ...)

lbt05_post(tlg, prune_0 = FALSE, keep = "Any Abnormality", ...)

lbt05

lab_paramcd_abn_dir()
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
map	(data.frame) with two columns mapping of parameters code (PARAMCD) to directions of abnormality (ABN_DIR). If a parameter is not in the map or if map is NULL, both directions are analyzed.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows
keep	(character) the levels to keep in the table even if they are empty. If NULL, all levels are pruned.

Details

- Does not remove rows with zero counts by default.
- Lab test results with missing AVAL values are excluded.
- Split columns by arm, typically ACTARM.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

a `data.frame` with the direction of abnormality of each lab parameter code.

Functions

- `lbt05_main()`: Main TLG function
- `lbt05_pre()`: Preprocessing
- `lbt05_post()`: Postprocessing
- `lab_paramcd_abn_dir()`: Default lab parameter abnormality direction

Note

- `adam_db` object must contain an `adlb` table with columns `"ONTRTFL"`, `"PARCAT2"`, `"PARAM"`, `"ANRIND"`, `"AVALCAT1"`, and column specified by `arm_var`.

Examples

```
run(lbt05, syn_data)

map <- data.frame(PARAMCD = c("ALT", "ALT", "CRP", "CRP", "IGA", "XXX"), ABN_DIR = c("Low", "High"))
run(lbt05, syn_data, map = map)
# example code
head(lab_paramcd_abn_dir())
```

---

lbt06_main — LBT06 *Table 1 (Default) Laboratory Abnormalities by Visit and Baseline Status Table 1.*

Description

The `LBT06` table produces the standard laboratory abnormalities by visit and baseline status summary.

Usage

```
lbt06_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  page_var = "PARAMCD",
  ...
)
```

```
lbt06_pre(adam_db, ...)

lbt06_post(tlg, prune_0 = FALSE, ...)

lbt06
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) the arm variable used for arm splitting.
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
page_var	(string) variable name prior to which the row split is by page.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Only count "LOW" or "HIGH" values for ANRIND and BNRIND.
- Lab test results with missing ANRIND values are excluded.
- Split columns by arm, typically ACTARM.
- Keep zero count rows by default.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- lbt06_main(): Main TLG function
- lbt06_pre(): Preprocessing
- lbt06_post(): Postprocessing

Note

- adam_db object must contain an adlb table with columns "AVISIT", "ANRIND", "BNRIND", "ONTRTFL", and "PARCAT2", and column specified by arm_var.

Examples

```
run(lbt06, syn_data)
```

lbt07_main — LBT07 *Table 1 (Default) Laboratory Test Results and Change from Baseline by Visit.*

Description

The LBT07 table provides an overview of the analysis values and its change from baseline of each respective arm over the course of the trial.

Usage

```
lbt07_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  param_var = "PARAM",
  grad_dir_var = "GRADE_DIR",
  grad_anl_var = "GRADE_ANL",
  ...
)

lbt07_pre(adam_db, ...)

lbt07_post(tlg, prune_0 = TRUE, keep = "Any", ...)

lbt07
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
param_var	(string) the name of the column storing the parameters name.
grad_dir_var	(string) the name of the column storing the grade direction variable which is required in order to obtain the correct denominators when building the layout as it is used to define row splitting.
grad_anl_var	(string) the name of the column storing toxicity grade variable where all negative values from ATOXGR are replaced by their absolute values.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows
keep	(character) the levels to keep in the table even if they are empty. If NULL, all levels are pruned.

Details

- Split columns by arm, typically `ACTARM`.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `lbt07_main()`: Main TLG function
- `lbt07_pre()`: Preprocessing
- `lbt07_post()`: Postprocessing

Note

- `adam_db` object must contain an `adlb` table with columns `"USUBJID"`, `"ATOXGR"`, `"ONTRTFL"` and column specified by `arm_var`.

Examples

```
run(lbt07, syn_data)
```

---

lbt14_main — `LBT14` *Laboratory Test Results Shift Table – Highest* `NCI-CTCAE` *Grade Post-Baseline by Baseline Grade (Low or High Direction).*

Description

`LBT14` Laboratory Test Results Shift Table – Highest `NCI-CTCAE` Grade Post-Baseline by Baseline Grade (Low or High Direction).

Usage

```
lbt14_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  gr_missing = "incl",
  page_var = "PARAMCD",
  ...
)

lbt14_pre(adam_db, gr_missing = "incl", direction = "low", ...)
```

```
lbt14_post(tlg, prune_0 = TRUE, ...)

lbt14
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
gr_missing	(string) how missing baseline grades should be handled. Defaults to "incl" to include the "Missing" level. Other options are "excl" to exclude patients with missing baseline grades and "gr_0" to convert missing baseline grades to grade 0.
page_var	(string) variable name prior to which the row split is by page.
...	not used.
direction	(string) one of "high" or "low" indicating which shift direction should be detailed.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- This table follows ADaMIG v1.1.
- Only the worst grade recorded for each patient is included in the table.
- If no missing baseline lab results, the "Missing" level of BTOXGR is excluded.
- Grading takes value from -4 to 4, negative value means the abnormality direction is low, positive value means the abnormality direction is high.
- Grades 0, 1, 2, 3, and 4 are counted as "Not Low" when direction = "low". Conversely, when direction = "high", Grades 0, -1, -2, -3, and -4 are counted as '"Not High".
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm, typically ACTARM.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- lbt14_main(): Main TLG function
- lbt14_pre(): Preprocessing
- lbt14_post(): Postprocessing

Note

- adam_db object must contain an adlb table with columns "USUBJID", "PARAM", "BTOXGR", "ATOXGR", and the column specified by arm_var.

Examples

```
run(lbt14, syn_data)
```

---

lbt15_pre — LBT15 *Laboratory Test Shifts to* NCI-CTCAE *Grade 3-4 Post-Baseline Table.*

Description

LBT15 Laboratory Test Shifts to NCI-CTCAE Grade 3-4 Post-Baseline Table.

Usage

```
lbt15_pre(adam_db, ...)

lbt15
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
...	not used.

Value

the preprocessing function returns a list of data.frame.

Functions

- lbt15_pre(): Preprocessing

Source

lbt04.R

Examples

```
run(lbt15, syn_data)
```

listing_format_chevron *Format for Chevron Listings*

Description

Format for Chevron Listings

Usage

```
listing_format_chevron()
```

Value

a `list` of `fmt_config`.

lvls *Obtain levels from vector*

Description

Obtain levels from vector

Usage

```
lvls(x)
```

Arguments

`x`	(`character`) or (`factor`) object to obtain levels.

Details

For factors, the levels will be returned. For characters, the sorted unique values will be returned.

Value

`character` with unique values.

main *Main*

Description

retrieve or set main function.

Usage

```
main(x)

## S4 method for signature 'chevron_tlg'
main(x)

main(x) <- value

## S4 replacement method for signature 'chevron_tlg'
main(x) <- value
```

Arguments

x	(chevron_tlg) input.
value	(function) returning a tlg. Typically one of the _main function of chevron.

Value

the function stored in the main slot of the x argument.

mht01_label MHT01 *Medical History Table.*

Description

The MHT01 table provides an overview of the subjects medical history by SOC and Preferred Term.

Usage

```
mht01_label

mht01_main(
  adam_db,
  arm_var = "ARM",
  row_split_var = "MHBODSYS",
  lbl_overall = NULL,
  summary_labels = list(all = mht01_label),
```

```
  ...
)

mht01_pre(adam_db, ...)

mht01_post(tlg, row_split_var = "MHBODSYS", prune_0 = TRUE, ...)

mht01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
row_split_var	(character) additional row split variables.
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
summary_labels	(list) of summarize labels. See details.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Numbers represent absolute numbers of patients and fraction of N, or absolute number of event when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Order by row_split_var alphabetically and medical condition by decreasing total number of patients with the specific condition. summary_labels is used to control the summary for each level. If "all" is used, then each split will have that summary statistic with the labels. One special case is "TOTAL", this is for the overall population.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- mht01_label: Default labels
- mht01_main(): Main TLG function
- mht01_pre(): Preprocessing
- mht01_post(): Postprocessing

Note

- adam_db object must contain an admh table with columns "MHBODSYS" and "MHDECOD".

Examples

```
run(mht01, syn_data)
```

---

missing_rule *Missing rule*

Description

Missing rule

Usage

```
missing_rule
```

---

mla_dir *MLA Grade Direction Data*

Description

MLA Grade Direction Data

Usage

```
mla_dir
```

Format

An object of class data.frame with 76 rows and 2 columns.

mng01_main MNG01 *Mean Plot Graph.*

Description

Overview of a summary statistics across time and arm for a selected data set.

Usage

```
mng01_main(
  adam_db,
  dataset = "adlb",
  x_var = "AVISIT",
  y_var = "AVAL",
  y_name = "PARAM",
  y_unit = NULL,
  arm_var = "ACTARM",
  center_fun = "mean",
  interval_fun = "mean_ci",
  jitter = 0.3,
  line_col = nestcolor::color_palette(),
  line_type = NULL,
  ggtheme = gg_theme_chevron(),
  table = c("n", center_fun, interval_fun),
  ...
)

mng01_pre(adam_db, dataset, x_var = "AVISIT", ...)

mng01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
x_var	(string) the name of a column in the dataset to represent on the x-axis.
y_var	(string) the name of the variable to be represented on the y-axis.
y_name	(string) the variable name for y. Used for plot's subtitle.
y_unit	(string) the name of the variable with the units of y. Used for plot's subtitle. if NULL, only y_name is displayed as subtitle.
arm_var	(string) variable used for column splitting
center_fun	(string) the function to compute the estimate value.
interval_fun	(string) the function defining the crossbar range. If NULL, no crossbar is displayed.

jitter	(numeric) the width of spread for data points on the x-axis; a number from 0 (no jitter) to 1 (high jitter), with a default of 0.3 (slight jitter).
line_col	(character) describing the colors to use for the lines or a named character associating values of arm_var with color names.
line_type	(character) describing the line type to use for the lines or a named character associating values of arm_var with line types.
ggtheme	(theme) passed to tern::g_lineplot().
table	(character) names of the statistics to be displayed in the table. If NULL, no table is displayed.
...	passed to tern::g_lineplot().

Details

- No overall value.
- Preprocessing filters for ANL01FL in the selected data set.

Value

the main function returns a list of ggplot objects.

a list of ggplot objects.

the preprocessing function returns a list of data.frame.

Functions

- mng01_main(): Main TLG Function
- mng01_pre(): Preprocessing

Note

- adam_db object must contain the table specified by dataset with the columns specified by x_var, y_var, y_name, y_unit and arm_var.

See Also

gg_theme_chevron(), tern::g_lineplot().

Examples

```
col <- c(
  "A: Drug X" = "black",
  "B: Placebo" = "blue",
  "C: Combination" = "gray"
)

lt <- c(
  "A: Drug X" = "29",
  "B: Placebo" = "99",
  "C: Combination" = "solid"
```

```
)

run(
  mng01,
  syn_data,
  dataset = "adlb",
  x_var = c("AVISIT", "AVISITN"),
  line_col = col,
  line_type = lt
)
```

---

nocoding *No Coding Available rule*

Description

No Coding Available rule

Usage

```
nocoding
```

---

outcome_rule *Outcome Rule*

Description

Outcome Rule

Usage

```
outcome_rule
```

pdt01_main pdt01 *Major Protocol Deviations Table.*

Description

A major protocol deviations table with the number of subjects and the total number of treatments by medication class sorted alphabetically and medication name sorted by frequencies.

Usage

```
pdt01_main(
  adam_db,
  arm_var = "ARM",
  lbl_overall = NULL,
  dvcode_var = "DVDECOD",
  dvterm_var = "DVTERM",
  ...
)

pdt01_pre(adam_db, ...)

pdt01_post(
  tlg,
  prune_0 = TRUE,
  dvcode_var = "DVDECOD",
  dvterm_var = "DVTERM",
  ...
)

pdt01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
dvcode_var	(string) the variable defining the protocol deviation coded term. By default DVDECOD.
dvterm_var	(string) the variable defining the protocol deviation term. By default DVTERM.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Data should be filtered for major protocol deviations. (DVCAT == "MAJOR").
- Numbers represent absolute numbers of subjects and fraction of N, or absolute numbers when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort by medication class alphabetically and within medication class by decreasing total number of patients with the specific medication.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- pdt01_main(): Main TLG function
- pdt01_pre(): Preprocessing
- pdt01_post(): Postprocessing

Note

- adam_db object must contain an addv table with the columns specified in dvcode_var and dvterm_var as well as "DVSEQ".

Examples

```
run(pdt01, syn_data)
```

---

pdt02_main — pdt02 *Major Protocol Deviations Related to Epidemic/Pandemic Table.*

Description

A major protocol deviations table with the number of subjects and the total number of Major Protocol Deviations Related to Epidemic/Pandemic sorted alphabetically and deviations name sorted by frequencies.

Usage

```
pdt02_main(
  adam_db,
  arm_var = "ARM",
  lbl_overall = NULL,
  dvreas_var = "DVREAS",
  dvterm_var = "DVTERM",
  ...
)

pdt02_pre(adam_db, ...)

pdt02_post(
  tlg,
  prune_0 = TRUE,
  dvreas_var = "DVREAS",
  dvterm_var = "DVTERM",
  ...
)

pdt02
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
dvreas_var	(string) the variable defining the reason for deviation. By default DVREAS.
dvterm_var	(string) the variable defining the protocol deviation term. By default DVTERM.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Data should be filtered for major protocol deviations related to epidemic/pandemic. (AEPRELFL == "Y" & DVCAT == "MAJOR").
- Numbers represent absolute numbers of subjects and fraction of N, or absolute numbers when specified.
- Remove zero-count rows unless overridden with prune_0 = FALSE.
- Split columns by arm.
- Does not include a total column by default.
- Sort by deviation reason alphabetically and within deviation reason by decreasing total number of patients with the specific deviation term.

Value

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `pdt02_main()`: Main TLG function
- `pdt02_pre()`: Preprocessing
- `pdt02_post()`: Postprocessing

Note

- `adam_db` object must contain an `addv` table with the columns specified in `dvreas_var` and `dvterm_var`.

Examples

```
run(pdt02, syn_data)
```

---

postprocess *Post process*

Description

retrieve or set `postprocess` function.

Usage

```
postprocess(x)

## S4 method for signature 'chevron_tlg'
postprocess(x)

postprocess(x) <- value

## S4 replacement method for signature 'chevron_tlg'
postprocess(x) <- value
```

Arguments

`x`	(`chevron_tlg`) input.
`value`	(`function`) returning a post-processed `tlg`.

Value

the `function` stored in the `postprocess` slot of the `x` argument.

preprocess *Pre process*

Description

retrieve or set `preprocess` function.

Usage

```
preprocess(x)

## S4 method for signature 'chevron_tlg'
preprocess(x)

preprocess(x) <- value

## S4 replacement method for signature 'chevron_tlg'
preprocess(x) <- value
```

Arguments

`x`	(`chevron_tlg`) input.
`value`	(`function`) returning a pre-processed `list` of `data.frames` amenable to `tlg` creation. Typically one of the `_pre` function of `chevron`.

Value

the `function` stored in the `preprocess` slot of the `x` argument.

report_null *Creates* `NULL` *Report*

Description

Creates `NULL` Report

Usage

```
report_null(tlg, ...)

## S4 method for signature 'NULL'
report_null(tlg, ind = 2L, ...)

## S4 method for signature 'VTableTree'
report_null(tlg, ind = 2L, ...)
```

```
## S4 method for signature 'listing_df'
report_null(tlg, ind = 2L, ...)

## S4 method for signature 'list'
report_null(tlg, ind = 2L, ...)

## S4 method for signature 'ANY'
report_null(tlg, ...)

standard_null_report()
```

Arguments

tlg	to convert to null report.
...	not used.
ind	(integer) indentation for the outputs of class VTableTree.

Value

the tlg object or a NULL report if the tlg is NULL, is a TableTree with 0 rows, is a listing_df with 0 rows or is a list with 0 elements.

Examples

```
report_null(NULL)
```

---

rmpt01_main RMPT01*Duration of Exposure for Risk Management Plan Table.*

Description

The RMPT01 table provides an overview of duration of exposure.

Usage

```
rmpt01_main(
  adam_db,
  summaryvars = "AVALCAT1",
  show_tot = TRUE,
  row_split_var = NULL,
  col_split_var = NULL,
  overall_col_lbl = NULL,
  ...
)

rmpt01_pre(adam_db, summaryvars = "AVALCAT1", ...)
```

```
rmpt01_post(tlg, prune_0 = FALSE, ...)
```

```
rmpt01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
summaryvars	(string) variables to be analyzed. The label attribute of the corresponding columns in adex table of adam_db is used as label.
show_tot	(flag) whether to display the cumulative total.
row_split_var	(string) the name of the column that containing variable to split exposure by.
col_split_var	(string) additional column splitting variable.
overall_col_lbl	(string) name of the overall column. If NULL, no overall level is added.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Person time is the sum of exposure across all patients.
- Summary statistics are by default based on the number of patients in the corresponding N row (number of non-missing values).
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- rmpt01_main(): Main TLG function
- rmpt01_pre(): Preprocessing
- rmpt01_post(): Postprocessing

Note

- adam_db object must contain an adex table with "AVAL" and the columns specified by summaryvars.

Examples

```
run(rmpt01, syn_data, col_split_var = "SEX")
```

rmpt03_main rmpt03*Duration of Exposure for Risk Management Plan Table.*

Description

The rmpt03 table provides an overview of duration of exposure.

Usage

```
rmpt03_main(
  adam_db,
  summaryvars = "AGEGR1",
  show_tot = TRUE,
  row_split_var = NULL,
  col_split_var = "SEX",
  overall_col_lbl = "All Genders",
  ...
)

rmpt03_pre(adam_db, summaryvars = "AGEGR1", ...)

rmpt03
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
summaryvars	(string) variables to be analyzed. The label attribute of the corresponding columns in adex table of adam_db is used as label.
show_tot	(flag) whether to display the cumulative total.
row_split_var	(string) the name of the column that containing variable to split exposure by.
col_split_var	(string) additional column splitting variable.
overall_col_lbl	(string) name of the overall column. If NULL, no overall level is added.
...	not used.

Details

- Person time is the sum of exposure across all patients.
- Summary statistics are by default based on the number of patients in the corresponding N row (number of non-missing values).
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

Functions

- `rmpt03_main()`: Main TLG function
- `rmpt03_pre()`: Preprocessing

Examples

```
pre_data <- dunlin::propagate(syn_data, "adsl", "AGEGR1", "USUBJID")
run(rmpt03, pre_data)
```

---

rmpt04_main — RMPT04*Extent of Exposure by Ethnic Origin for Risk Management Plan Table.*

Description

The `RMPT04` table provides an overview of duration of exposure extent.

Usage

```
rmpt04_main(
  adam_db,
  summaryvars = "ETHNIC",
  show_tot = TRUE,
  row_split_var = NULL,
  col_split_var = NULL,
  overall_col_lbl = NULL,
  ...
)

rmpt04_pre(adam_db, summaryvars = "ETHNIC", ...)

rmpt04
```

Arguments

`adam_db`	(`list` of `data.frames`) object containing the `ADaM` datasets
`summaryvars`	(`string`) variables to be analyzed. The label attribute of the corresponding columns in `adex` table of `adam_db` is used as label.
`show_tot`	(`flag`) whether to display the cumulative total.
`row_split_var`	(`character`) additional row split variables.
`col_split_var`	(`string`) additional column splitting variable.
`overall_col_lbl`	(`string`) name of the overall column. If `NULL`, no overall level is added.
`...`	not used.

Details

- Person time is the sum of exposure across all patients.
- Summary statistics are by default based on the number of patients in the corresponding `N` row (number of non-missing values).
- Does not remove zero-count rows unless overridden with `prune_0 = TRUE`.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

Functions

- `rmpt04_main()`: Main TLG function
- `rmpt04_pre()`: Preprocessing

Examples

```
run(rmpt04, syn_data)
```

---

rmpt05_main — RMPT05 *Extent of Exposure by Race for Risk Management Plan Table.*

Description

The `RMPT05` table provides an overview of duration of exposure extent.

Usage

```
rmpt05_main(
  adam_db,
  summaryvars = "RACE",
  show_tot = TRUE,
  row_split_var = NULL,
  col_split_var = NULL,
  overall_col_lbl = NULL,
  ...
)

rmpt05_pre(adam_db, summaryvars = "RACE", ...)

rmpt05
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
summaryvars	(string) variables to be analyzed. The label attribute of the corresponding columns in adex table of adam_db is used as label.
show_tot	(flag) whether to display the cumulative total.
row_split_var	(character) additional row split variables.
col_split_var	(string) additional column splitting variable.
overall_col_lbl	(string) name of the overall column. If NULL, no overall level is added.
...	not used.

Details

- Person time is the sum of exposure across all patients.
- Summary statistics are by default based on the number of patients in the corresponding N row (number of non-missing values).
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

Functions

- rmpt05_main(): Main TLG function
- rmpt05_pre(): Preprocessing

Examples

```
run(rmpt05, syn_data)
```

---

rmpt06_main — RMPT06 *Table 1 (Default) Seriousness, Outcomes, Severity, Frequency with 95% CI for Risk Management Plan.*

Description

RMPT06 Table 1 (Default) Seriousness, Outcomes, Severity, Frequency with 95% CI for Risk Management Plan.

Usage

```
rmpt06_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  method = "clopper-pearson",
  conf_level = 0.95,
  show_diff = FALSE,
  ref_group = NULL,
  method_diff = "wald",
  conf_level_diff = 0.95,
  grade_groups = NULL,
  ...
)

rmpt06_pre(adam_db, ...)

rmpt06_post(tlg, prune_0 = FALSE, ...)

rmpt06
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
method	(string) the method used to construct the confidence interval. See tern::estimate_proportion.
conf_level	(proportion) the confidence level of the interval. See tern::estimate_proportion.
show_diff	(flag) whether to show the difference of patient with at least one adverse event between groups.
ref_group	(string) the reference group for the difference.
method_diff	(string) the method used to construct the confidence interval for the difference between groups.
conf_level_diff	(proportion) the confidence level of the interval for the difference between groups.
grade_groups	(list) the grade groups to be displayed.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `rmpt06_main()`: Main TLG function
- `rmpt06_pre()`: Preprocessing
- `rmpt06_post()`: Postprocessing

Examples

```
run(rmpt06, syn_data)
```

---

rspt01_main RSPT01 *Binary Outcomes Summary.*

Description

`RSPT01` template may be used to summarize any binary outcome or response variable at a single time point. Typical application for oncology

Usage

```
rspt01_main(
  adam_db,
  dataset = "adrs",
  arm_var = "ARM",
  ref_group = NULL,
  odds_ratio = TRUE,
  perform_analysis = "unstrat",
  strata = NULL,
  conf_level = 0.95,
  methods = list(),
  ...
)

rspt01_pre(adam_db, ...)

rspt01_post(tlg, prune_0 = TRUE, ...)

rspt01
```

Arguments

- `adam_db` (`list` of `data.frames`) object containing the `ADaM` datasets
- `dataset` (`string`) the name of a table in the `adam_db` object.
- `arm_var` (`string`) variable used for column splitting
- `ref_group` (`string`) The name of the reference group, the value should be identical to the values in `arm_var`, if not specified, it will by default use the first level or value of `arm_var`.
- `odds_ratio` (`flag`) should the odds ratio be calculated, default is `TRUE`
- `perform_analysis` (`string`) option to display statistical comparisons using stratified analyses, or unstratified analyses, or both, e.g. `c("unstrat", "strat")`. Only unstratified will be displayed by default
- `strata` (`string`) stratification factors, e.g. `strata = c("STRATA1", "STRATA2")`, by default as NULL
- `conf_level` (`numeric`) the level of confidence interval, default is 0.95.
- `methods` (`list`) a named list, use a named list to control, for example: `methods = list(prop_conf_method = "wald", diff_conf_method = "wald", strat_diff_conf_method = "ha", diff_pval_method = "fisher", strat_diff_pval_method = "schouten")` `prop_conf_method` controls the methods of calculating proportion confidence interval, `diff_conf_method` controls the methods of calculating unstratified difference confidence interval, `strat_diff_conf_method` controls the methods of calculating stratified difference confidence interval, `diff_pval_method` controls the methods of calculating unstratified p-value for odds ratio, `strat_diff_pval_method` controls the methods of calculating stratified p-value for odds ratio, see more details in `tern`
- `...` not used.
- `tlg` (`TableTree`, `Listing` or `ggplot`) object typically produced by a `main` function.
- `prune_0` (`flag`) remove 0 count rows

Details

- No overall value.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `rspt01_main()`: Main TLG function
- `rspt01_pre()`: Preprocessing
- `rspt01_post()`: Postprocessing

Examples

```
library(dplyr)
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "BESRSPI", "adrs")

run(rspt01, proc_data)

run(rspt01, proc_data,
  odds_ratio = FALSE, perform_analysis = c("unstrat", "strat"),
  strata = c("STRATA1", "STRATA2"), methods = list(diff_pval_method = "fisher")
)
```

---

run *Run the TLG-generating pipeline*

Description

Run the TLG-generating pipeline

Usage

```
run(
  object,
  adam_db,
  auto_pre = TRUE,
  verbose = FALSE,
  unwrap = FALSE,
  ...,
  user_args = list(...)
)

## S4 method for signature 'chevron_tlg'
run(
  object,
  adam_db,
  auto_pre = TRUE,
  verbose = get_arg("chevron.run.verbose", "R_CHEVRON_RUN_VERBOSE", FALSE),
  unwrap = get_arg("chevron.run.unwrap", "R_CHEVRON_RUN_UNWRAP", verbose),
  ...,
  user_args = list(...)
)
```

Arguments

object	(chevron_tlg) input.
adam_db	(list of data.frames) object containing the ADaM datasets

auto_pre	(flag) whether to perform the default pre processing step.
verbose	(flag) whether to print argument information.
unwrap	(flag) whether to print the preprocessing postprocessing and main function together with the associated layout function.
...	extra arguments to pass to the pre-processing, main and post-processing functions.
user_args	(list) arguments from ....

Details

The functions stored in the preprocess, main and postprocess slots of the chevron_tlg objects are called respectively, preprocessing, main and postprocessing functions.

When executing the run method on a chevron_tlg object, if auto_pre is TRUE, the adam_bd list is first passed to the preprocessing function. The resulting list is then passed to the main function which produces a table, graph or listings or a list of these objects. This output is then passed to the postprocessing function which performed the final modifications before returning the output. Additional arguments specified in ... or user_args are passed to each of the three functions.

Value

an rtables (for chevron_t), rlistings (for chevron_l), grob (for chevron_g) or ElementaryTable (null report) depending on the class of chevron_tlg object passed as object argument.

Examples

```
run(mng01, syn_data, auto_pre = TRUE, dataset = "adlb")
```

---

script *Create Script for* TLG *Generation*

Description

Create Script for TLG Generation

Usage

```
script_funs(x, adam_db, args, name = deparse(substitute(x)))

## S4 method for signature 'chevron_tlg'
script_funs(x, adam_db, args, name = deparse(substitute(x)))

## S4 method for signature 'chevron_simple'
script_funs(x, adam_db, args, name = deparse(substitute(x)))
```

Arguments

x	(chevron_tlg) input.
adam_db	(string) the name of the dataset.
args	(string) the name of argument list.
name	(string) name of the template.

Value

character that can be integrated into an executable script.

Examples

```
script_funs(aet04, adam_db = "syn_data", args = "args")
```

---

set_section_div *Set Section Dividers*

Description

Set Section Dividers

Usage

```
set_section_div(x)
```

Arguments

x	(integerish) value of at which the section divider should be added.

Details

Section dividers are empty lines between sections in tables. E.g. if 1 is used then for the first row split an empty line is added. Currently it only works for aet02, cmt01a and mht01 template.

Value

invisible NULL. Set the chevron.section_div option.

smart_prune *Prune table up to an* ElementaryTable

Description

Avoid returning NULL when the table is empty.

Usage

```
smart_prune(tlg)
```

Arguments

tlg	(TableTree) object.

Value

pruned TableTree.

std_postprocessing *Standard Post Processing*

Description

Standard Post Processing

Usage

```
std_postprocessing(tlg, ...)
```

Arguments

tlg	to post process.
...	additional arguments passed to report_null.

Value

a processed tlg or a null report.

Examples

```
library(rtables)
std_postprocessing(build_table(basic_table() |> analyze("Species"), iris), ind = 10L)
```

syn_data — *Example* adam *Synthetic Data*

Description

Example adam Synthetic Data

Usage

```
syn_data
```

Format

A named list of 13 data.frames: - adsl - adae - adsaftte - adcm - addv - adeg - adex - adlb - admh - adrs - adsub - adtte - advs

Source

based on package random.cdisc.data

ttet01_main — TTET01 *Binary Outcomes Summary.*

Description

TTET01 template may be used to summarize any binary outcome or response variable at a single time point. Typical application for oncology

Usage

```
ttet01_main(
  adam_db,
  dataset = "adtte",
  arm_var = "ARM",
  ref_group = NULL,
  summarize_event = TRUE,
  perform_analysis = "unstrat",
  strata = NULL,
  ...
)

ttet01_pre(adam_db, dataset = "adtte", ...)

ttet01_post(tlg, prune_0 = TRUE, ...)

ttet01
```

Arguments

- `adam_db` (`list` of `data.frames`) object containing the `ADaM` datasets
- `dataset` (`string`) the name of a table in the `adam_db` object.
- `arm_var` (`string`) variable used for column splitting
- `ref_group` (`string`) The name of the reference group, the value should be identical to the values in `arm_var`, if not specified, it will by default use the first level or value of `arm_var`.
- `summarize_event` (`flag`) should the event description be displayed, default is TRUE
- `perform_analysis` (`string`) option to display statistical comparisons using stratified analyses, or unstratified analyses, or both, e.g. `c("unstrat", "strat")`. Only unstratified will be displayed by default
- `strata` (`string`) stratification factors, e.g. `strata = c("STRATA1", "STRATA2")`, by default as NULL
- `...` Further arguments passed to `control_surv_time()`, `control_coxph()`, `control_survtp()`, and `surv_timepoint()`. For details, see the documentation in `tern`. Commonly used arguments include `pval_method`, `conf_level`, `conf_type`, `quantiles`, `ties`, `time_point`, `method`, etc.
- `tlg` (`TableTree`, `Listing` or `ggplot`) object typically produced by a `main` function.
- `prune_0` (`flag`) remove 0 count rows

Details

- No overall value.

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

the postprocessing function returns an `rtables` object or an `ElementaryTable` (null report).

Functions

- `ttet01_main()`: Main TLG function
- `ttet01_pre()`: Preprocessing
- `ttet01_post()`: Postprocessing

Examples

```
library(dplyr)
library(dunlin)

proc_data <- log_filter(syn_data, PARAMCD == "PFS", "adtte")
run(ttet01, proc_data)
```

```
run(ttet01, proc_data,
  summarize_event = FALSE, perform_analysis = c("unstrat", "strat"),
  strata = c("STRATA1", "STRATA2"),
  conf_type = "log-log",
  time_point = c(6, 12),
  method = "both"
)
```

---

var_labels_for *Retrieve labels for certain variables*

Description

Retrieve labels for certain variables

Usage

```
var_labels_for(df, vars)
```

Arguments

df	(data.frame) containing columns with label attribute.
vars	(character) variable names in df.

Details

The labels will be returned if the column has label attribute, otherwise the column name will be returned. Any values between brackets will be replaced with dunlin::render_safe.

Value

a character with replaced placeholders and a label attribute.

---

vst01_main VST01 *Vital Sign Results and change from Baseline By Visit Table.*

Description

The VST01 table provides an overview of the Vital Sign values and its change from baseline of each respective arm over the course of the trial.

Usage

```
vst01_main(
  adam_db,
  dataset = "advs",
  arm_var = "ACTARM",
  lbl_overall = NULL,
  row_split_var = NULL,
  summaryvars = c("AVAL", "CHG"),
  visitvar = "AVISIT",
  precision = list(DIABP = 0, PUL = 0, SYSBP = 0, default = 2L),
  page_var = "PARAMCD",
  .stats = c("n", "mean_sd", "median", "range"),
  skip = list(CHG = "BASELINE"),
  ...
)

vst01_pre(adam_db, dataset = "advs", ...)

vst01
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
dataset	(string) the name of a table in the adam_db object.
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
row_split_var	(character) additional row split variables.
summaryvars	(character) variables to be analyzed. The label attribute of the corresponding column in table of adam_db is used as label.
visitvar	(string) typically one of "AVISIT" or user-defined visit incorporating "ATPT".
precision	(named list of integer) where names are values found in the PARAMCD column and the values indicate the number of digits in statistics. If default is set, and parameter precision not specified, the value for default will be used. If default is not set, the default precision is 2.
page_var	(string) variable name prior to which the row split is by page.
.stats	(character) statistics names, see tern::analyze_vars().
skip	Named (list) of visit values that need to be inhibited.
...	additional arguments like .indent_mods, .labels.

Details

- The Analysis Value column, displays the number of patients, the mean, standard deviation, median and range of the analysis value for each visit.

- The `Change from Baseline` column, displays the number of patient and the mean, standard deviation, median and range of changes relative to the baseline.
- Remove zero-count rows unless overridden with `prune_0 = FALSE`.
- Split columns by arm, typically `ACTARM`.
- Does not include a total column by default.
- Sorted based on factor level; first by `PARAM` labels in alphabetic order then by chronological time point given by `AVISIT`. Re-level to customize order

Value

the main function returns an `rtables` object.

the preprocessing function returns a `list` of `data.frame`.

Functions

- `vst01_main()`: Main TLG function
- `vst01_pre()`: Preprocessing

Note

- `adam_db` object must contain table named as `dataset` with the columns specified in `summaryvars`.

Examples

```
library(dunlin)

proc_data <- log_filter(
  syn_data,
  PARAMCD %in% c("DIABP", "SYSBP"), "advs"
)
run(vst01, proc_data)
```

`vst02_1_main` `VST02` *Vital Sign Abnormalities Table.*

Description

Vital Sign Parameters outside Normal Limits Regardless of Abnormality at Baseline.

Usage

```
vst02_1_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  exclude_base_abn = FALSE,
  ...
```

```
)

vst02_pre(adam_db, ...)

vst02_post(tlg, prune_0 = FALSE, ...)

vst02_1
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
exclude_base_abn	(flag) whether baseline abnormality should be excluded.
...	not used.
tlg	(TableTree, Listing or ggplot) object typically produced by a main function.
prune_0	(flag) remove 0 count rows

Details

- Only count LOW or HIGH values.
- Results of "LOW LOW" are treated as the same as "LOW", and "HIGH HIGH" the same as "HIGH".
- Does not include a total column by default.
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- vst02_1_main(): Main TLG function
- vst02_pre(): Preprocessing
- vst02_post(): Postprocessing

Note

- adam_db object must contain an advs table with the "PARAM", "ANRIND" and "BNRIND" columns.

Examples

```
run(vst02_1, syn_data)
```

vst02_2_main VST02 *Vital Sign Abnormalities Table.*

Description

Vital Sign Parameters outside Normal Limits Among Patients without Abnormality at Baseline.

Usage

```
vst02_2_main(
  adam_db,
  arm_var = "ACTARM",
  lbl_overall = NULL,
  exclude_base_abn = TRUE,
  ...
)

vst02_2
```

Arguments

adam_db	(list of data.frames) object containing the ADaM datasets
arm_var	(string) variable used for column splitting
lbl_overall	(string) label used for overall column, if set to NULL the overall column is omitted
exclude_base_abn	(flag) whether baseline abnormality should be excluded.
...	not used.

Details

- Only count LOW or HIGH values.
- Results of "LOW LOW" are treated as the same as "LOW", and "HIGH HIGH" the same as "HIGH".
- Does not include a total column by default.
- Does not remove zero-count rows unless overridden with prune_0 = TRUE.

Value

the main function returns an rtables object.

the preprocessing function returns a list of data.frame.

the postprocessing function returns an rtables object or an ElementaryTable (null report).

Functions

- vst02_2_main(): Main TLG function

Note

- adam_db object must contain an advs table with the "PARAM", "ANRIND" and "BNRIND" columns.

Examples

```
run(vst02_2, syn_data)
```

---

yes_no_rule *Yes/No rule in title case*

Description

Yes/No rule in title case

Usage

```
yes_no_rule
```

Index

∗ **datasets**
ctcv4_dir, 39
ctcv5_dir, 40
mla_dir, 81
syn_data, 103
.chevron_g *(chevron_tlg-class)*, 28
.chevron_l *(chevron_tlg-class)*, 28
.chevron_simple *(chevron_tlg-class)*, 28
.chevron_t *(chevron_tlg-class)*, 28
.chevron_tlg *(chevron_tlg-class)*, 28

ael01_nollt *(ael01_nollt_main)*, 5
ael01_nollt_main, 5
ael01_nollt_pre *(ael01_nollt_main)*, 5
ael02 *(ael02_main)*, 7
ael02_main, 7
ael02_pre *(ael02_main)*, 7
ael03 *(ael03_main)*, 8
ael03_main, 8
ael03_pre *(ael03_main)*, 8
aet01 *(aet01_main)*, 11
aet01_aesi *(aet01_aesi_main)*, 9
aet01_aesi_main, 9
aet01_aesi_post *(aet01_aesi_main)*, 9
aet01_aesi_pre *(aet01_aesi_main)*, 9
aet01_main, 11
aet01_post *(aet01_main)*, 11
aet01_pre *(aet01_main)*, 11
aet02 *(aet02_label)*, 12
aet02_label, 12
aet02_main *(aet02_label)*, 12
aet02_post *(aet02_label)*, 12
aet02_pre *(aet02_label)*, 12
aet03 *(aet03_main)*, 14
aet03_main, 14
aet03_post *(aet03_main)*, 14
aet03_pre *(aet03_main)*, 14
aet04 *(aet04_main)*, 15
aet04_main, 15
aet04_post *(aet04_main)*, 15
aet04_pre *(aet04_main)*, 15
aet05 *(aet05_main)*, 18
aet05_all *(aet05_all_pre)*, 17
aet05_all_pre, 17
aet05_main, 18
aet05_post *(aet05_main)*, 18
aet05_pre *(aet05_main)*, 18
aet10 *(aet10_main)*, 19
aet10_main, 19
aet10_post *(aet10_main)*, 19
aet10_pre *(aet10_main)*, 19
args_ls, 21
args_ls,chevron_tlg-method *(args_ls)*, 21
assert_single_value, 21
assert_valid_type, 22
assert_valid_var, 22
assert_valid_var_pair, 24
assert_valid_variable, 25

cfbt01 *(cfbt01_main)*, 25
cfbt01_main, 25
cfbt01_post *(cfbt01_main)*, 25
cfbt01_pre *(cfbt01_main)*, 25
chevron *(chevron-package)*, 4
chevron-package, 4
chevron_g *(chevron_tlg-class)*, 28
chevron_g-class *(chevron_tlg-class)*, 28
chevron_graph *(chevron_tlg-class)*, 28
chevron_l *(chevron_tlg-class)*, 28
chevron_l-class *(chevron_tlg-class)*, 28
chevron_listing *(chevron_tlg-class)*, 28
chevron_simple *(chevron_tlg-class)*, 28
chevron_simple-class *(chevron_tlg-class)*, 28
chevron_t *(chevron_tlg-class)*, 28
chevron_t-class *(chevron_tlg-class)*, 28
chevron_table *(chevron_tlg-class)*, 28
chevron_tlg, *28*
chevron_tlg-class, 28
cml02a_gl *(cml02a_gl_main)*, 30

cml02a_gl_main, 30
cml02a_gl_pre (cml02a_gl_main), 30
cmt01_label, 31
cmt01a (cmt01_label), 31
cmt01a_main (cmt01_label), 31
cmt01a_post (cmt01_label), 31
cmt01a_pre (cmt01_label), 31
cmt02_pt (cmt02_pt_main), 33
cmt02_pt_main, 33
cmt02_pt_post (cmt02_pt_main), 33
cmt02_pt_pre (cmt02_pt_main), 33
convert_to_month, 35
coxt01 (coxt01_main), 35
coxt01_main, 35
coxt01_post (coxt01_main), 35
coxt01_pre (coxt01_main), 35
coxt02 (coxt02_main), 37
coxt02_main, 37
create_id_listings, 39
ctcv4_dir, 39
ctcv5_dir, 40

dataset, 40
dataset,chevron_tlg-method (dataset), 40
dataset<- (dataset), 40
dataset<-,chevron_tlg-method (dataset), 40
dmt01 (dmt01_main), 41
dmt01_main, 41
dmt01_post (dmt01_main), 41
dmt01_pre (dmt01_main), 41
dose_change_rule, 42
dst01 (dst01_main), 43
dst01_main, 43
dst01_post (dst01_main), 43
dst01_pre (dst01_main), 43
dtht01 (dtht01_main), 45
dtht01_main, 45
dtht01_post (dtht01_main), 45
dtht01_pre (dtht01_main), 45
dummy_template, 46

egt01 (egt01_main), 47
egt01_main, 47
egt01_pre (egt01_main), 47
egt02_1 (egt02_1_main), 48
egt02_1_main, 48
egt02_2 (egt02_2_main), 50
egt02_2_main, 50
egt02_post (egt02_1_main), 48
egt02_pre (egt02_1_main), 48
egt03 (egt03_main), 51
egt03_main, 51
egt03_post (egt03_main), 51
egt03_pre (egt03_main), 51
egt05_qtcat (egt05_qtcat_main), 53
egt05_qtcat_main, 53
egt05_qtcat_post (egt05_qtcat_main), 53
egt05_qtcat_pre (egt05_qtcat_main), 53
empty_rule, 54
ext01 (ext01_main), 55
ext01_main, 55
ext01_post (ext01_main), 55
ext01_pre (ext01_main), 55

format_date, 57
fstg01 (fstg01_main), 58
fstg01_main, 58
fstg01_pre (fstg01_main), 58
fstg02 (fstg02_main), 59
fstg02_main, 59
fstg02_pre (fstg02_main), 59

gen_args, 61
get_grade_rule, 63
get_section_div, 63
gg_list, 64
gg_theme_chevron, 64
gg_theme_chevron(), *83*
grob_list, 65

h_format_dec, 65

kmg01 (kmg01_main), 66
kmg01_main, 66
kmg01_pre (kmg01_main), 66

lab_paramcd_abn_dir (lbt05_main), 71
lab_paramcd_precision (lbt01_main), 67
lbt01 (lbt01_main), 67
lbt01_main, 67
lbt01_pre (lbt01_main), 67
lbt04 (lbt04_main), 69
lbt04_main, 69
lbt04_post (lbt04_main), 69
lbt04_pre (lbt04_main), 69
lbt05 (lbt05_main), 71
lbt05_main, 71

lbt05_post (lbt05_main), 71
lbt05_pre (lbt05_main), 71
lbt06 (lbt06_main), 72
lbt06_main, 72
lbt06_post (lbt06_main), 72
lbt06_pre (lbt06_main), 72
lbt07 (lbt07_main), 74
lbt07_main, 74
lbt07_post (lbt07_main), 74
lbt07_pre (lbt07_main), 74
lbt14 (lbt14_main), 75
lbt14_main, 75
lbt14_post (lbt14_main), 75
lbt14_pre (lbt14_main), 75
lbt15 (lbt15_pre), 77
lbt15_pre, 77
listing_format_chevron, 78
lvls, 78

main, 79
main,chevron_tlg-method (main), 79
main<- (main), 79
main<-,chevron_tlg-method (main), 79
mht01 (mht01_label), 79
mht01_label, 79
mht01_main (mht01_label), 79
mht01_post (mht01_label), 79
mht01_pre (mht01_label), 79
missing_rule, 81
mla_dir, 81
mng01 (mng01_main), 82
mng01_main, 82
mng01_pre (mng01_main), 82

nocoding, 84

outcome_rule, 84

pdt01 (pdt01_main), 85
pdt01_main, 85
pdt01_post (pdt01_main), 85
pdt01_pre (pdt01_main), 85
pdt02 (pdt02_main), 86
pdt02_main, 86
pdt02_post (pdt02_main), 86
pdt02_pre (pdt02_main), 86
postprocess, 88
postprocess,chevron_tlg-method (postprocess), 88
postprocess<- (postprocess), 88
postprocess<-,chevron_tlg-method (postprocess), 88
preprocess, 89
preprocess,chevron_tlg-method (preprocess), 89
preprocess<- (preprocess), 89
preprocess<-,chevron_tlg-method (preprocess), 89

report_null, 89, *102*
report_null,ANY-method (report_null), 89
report_null,list-method (report_null), 89
report_null,listing_df-method (report_null), 89
report_null,NULL-method (report_null), 89
report_null,VTableTree-method (report_null), 89
rlistings::as_listing, *6*
rmpt01 (rmpt01_main), 90
rmpt01_main, 90
rmpt01_post (rmpt01_main), 90
rmpt01_pre (rmpt01_main), 90
rmpt03 (rmpt03_main), 92
rmpt03_main, 92
rmpt03_pre (rmpt03_main), 92
rmpt04 (rmpt04_main), 93
rmpt04_main, 93
rmpt04_pre (rmpt04_main), 93
rmpt05 (rmpt05_main), 94
rmpt05_main, 94
rmpt05_pre (rmpt05_main), 94
rmpt06 (rmpt06_main), 95
rmpt06_main, 95
rmpt06_post (rmpt06_main), 95
rmpt06_pre (rmpt06_main), 95
rspt01 (rspt01_main), 97
rspt01_main, 97
rspt01_post (rspt01_main), 97
rspt01_pre (rspt01_main), 97
run, 99
run,chevron_tlg-method (run), 99

script, 100
script_funs (script), 100
script_funs,chevron_simple-method (script), 100

script_funs,chevron_tlg-method
(script), 100
set_section_div, 101
smart_prune, 102
standard_null_report (report_null), 89
std_postprocessing, 102
syn_data, 103

tern::estimate_proportion, *96*
tern::format_auto, *41*
tern::g_lineplot(), *83*
ttet01 (ttet01_main), 103
ttet01_main, 103
ttet01_post (ttet01_main), 103
ttet01_pre (ttet01_main), 103

var_labels_for, 105
vst01 (vst01_main), 105
vst01_main, 105
vst01_pre (vst01_main), 105
vst02_1 (vst02_1_main), 107
vst02_1_main, 107
vst02_2 (vst02_2_main), 109
vst02_2_main, 109
vst02_post (vst02_1_main), 107
vst02_pre (vst02_1_main), 107

yes_no_rule, 110