Package 'metalite.ae'

August 29, 2026

Title Adverse Events Analysis Using 'metalite'

Version 0.1.4

Description Analyzes adverse events in clinical trials using the 'metalite' data structure. The package simplifies the workflow to create production-ready tables, listings, and figures discussed in the adverse events analysis chapters of ``R for Clinical Study Reports and Submission'' by Zhang et al. (2022) <https://r4csr.org/>.

License GPL-3

URL https://merck.github.io/metalite.ae/, https://github.com/Merck/metalite.ae

BugReports https://github.com/Merck/metalite.ae/issues

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports glue, metalite, r2rtf, stats, gt

Suggests DescTools, covr, dplyr, knitr, forestly, reactable, rmarkdown, testthat (>= 3.0.0), tibble, tidyr

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Yilong Zhang [aut], Yujie Zhao [aut, cre], Hiroaki Fukuda [aut], Benjamin Wang [aut], Nan Xiao [aut], Sarad Nepal [aut], Madhusudhan Ginnaram [aut], Venkatesh Burla [ctb],

Ruchitbhai Patel [aut],
Brian Lang [aut],
Xuan Deng [aut],
Bing Liu [aut],
Jeetender Chauhan [aut],
Li Ma [ctb],
Merck Sharp & Dohme Corp [cph]

Maintainer Yujie Zhao <yujie.zhao@merck.com>

Repository CRAN

Date/Publication 2026-08-29 12:10:22 UTC

Contents

extend_ae_specific_duration . . . 3
extend_ae_specific_events . . . 3
extend_ae_specific_inference . . . 4
extend_ae_specific_subgroup . . . 6
extend_ae_summary_eaer . . . 7
fmt_ci . . . 9
fmt_est . . . 10
fmt_pct . . . 11
fmt_pval . . . 11
format_ae_exp_adj . . . 12
format_ae_listing . . . 14
format_ae_specific . . . 16
format_ae_specific_subgroup . . . 19
format_ae_summary . . . 22
gt_ae_specific . . . 25
gt_ae_summary . . . 27
metalite_ae_adex . . . 29
metalite_ae_adexsum . . . 30
prepare_ae_listing . . . 31
prepare_ae_specific . . . 32
prepare_ae_specific_subgroup . . . 34
prepare_ae_summary . . . 37
rate_compare . . . 39
rate_compare_sum . . . 41
react_ae_listing . . . 42
tlf_ae_exp_adj . . . 44
tlf_ae_listing . . . 46
tlf_ae_specific . . . 47
tlf_ae_specific_subgroup . . . 50
tlf_ae_summary . . . 52

Index **56**

extend_ae_specific_duration

Add average duration information for AE specific analysis

Description

Add average duration information for AE specific analysis

Usage

```
extend_ae_specific_duration(outdata, duration_var, duration_unit = "Day")
```

Arguments

outdata	An outdata object created by prepare_ae_specific().
duration_var	A character value of variable name for adverse event duration.
duration_unit	A character value of adverse event duration unit.

Value

A list of analysis raw datasets.

Examples

```
meta <- metalite::meta_example()
tbl <- prepare_ae_specific(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel"
) |>
  extend_ae_specific_duration(duration_var = "ADURN") |>
  format_ae_specific(display = c("n", "prop", "dur"))
head(tbl$tbl)
```

extend_ae_specific_events

Add average number of events information for AE specific analysis

Description

Add average number of events information for AE specific analysis

Usage

```
extend_ae_specific_events(outdata)
```

Arguments

outdata	An outdata object created by prepare_ae_specific().

Value

A list of analysis raw datasets.

Examples

```
meta <- metalite::meta_example()
tbl <- prepare_ae_specific(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel"
) |>
  extend_ae_specific_events() |>
  format_ae_specific(display = c("n", "prop", "events_avg"))
head(tbl$tbl)
```

---

extend_ae_specific_inference

Add inference information for AE specific analysis

Description

Add inference information for AE specific analysis

Usage

```
extend_ae_specific_inference(outdata, ..., ci = 0.95)
```

Arguments

outdata	An outdata object created by prepare_ae_specific().
...	Other options passed on to rate_compare_sum()
ci	A numeric value for the percentile of confidence interval.

Value

A list of analysis raw datasets.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRT01A", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL == "RELATED",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
```

```
    name = "ae_specific",
    title = "Participants With {term1} Adverse Events {term2}"
  ) |>
  metalite::meta_build()

# Calculate AE specific analysis and format it
tbl <- prepare_ae_specific(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel"
) |>
  extend_ae_specific_inference(eps = 1e-6, bisection = 200) |>
  format_ae_specific(display = c("n", "prop", "diff", "diff_ci"))
head(tbl$tbl)
```

`extend_ae_specific_subgroup` *Add subgroup analysis in AE specific analysis*

Description

Add subgroup analysis in AE specific analysis

Usage

```
extend_ae_specific_subgroup(outdata, subgroup_var)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`subgroup_var`	a character string for subgroup variable name

Value

A list of analysis raw datasets.

Examples

```
meta <- metalite::meta_example()
tbl <- prepare_ae_specific(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel"
) |>
  extend_ae_specific_subgroup(subgroup_var = "SEX")
```

`extend_ae_summary_eaer` *Add exposure-adjusted rate information for AE summary analysis*

Description

Add exposure-adjusted rate information for AE summary analysis

Usage

```
extend_ae_summary_eaer(
  outdata,
  duration_var = "TRTDUR",
  adj_unit = c("year", "month", "week", "day")
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_summary()`.
`duration_var`	A character value of duration variable name. By default, `"TRTDUR"` is used.
`adj_unit`	A character value of exposure adjusted unit. It could be select from `"year"`, `"month"`, `"week"`, and `"day"`.

Value

A list of analysis raw datasets.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
```

```
  parameter = "any;rel;ser"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "any",
    term1 = "",
    term2 = "",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "All AEs"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_summary",
```

```
    title = "Adverse Event Summary"
  ) |>
  metalite::meta_build()

# Calculate exposure-adjusted event rates
prepare_ae_summary(
  meta,
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
) |>
  extend_ae_summary_eaer()
```

`fmt_ci` *Format confidence interval*

Description

Format confidence interval

Usage

```
fmt_ci(lower, upper, digits = 2, width = 3 + digits)
```

Arguments

`lower`	A numeric value of lower value of CI.
`upper`	A numeric value of upper value of CI.
`digits`	Digits of each column, i.e., format as (x.x, x.x).
`width`	Width of each column.

Value

A numeric vector with the expected format.

Examples

```
fmt_ci(0.2356, 0.3871)
```

`fmt_est` *Format model estimator*

Description

Formats mean sd/se to a format as x.x or x.x (x.xx) if both mean and sd/sd are defined.

Usage

```
fmt_est(
  mean,
  sd = rep(NA, length(mean)),
  digits = c(1, 1),
  width = c(4, 3) + digits
)
```

Arguments

`mean`	A numeric vector of mean value.
`sd`	A numeric vector of standard deviation value.
`digits`	Digits of each column, i.e., format as x.x (x.xx).
`width`	Width of each column.

Details

The function assumes 1 column or 2 columns:

- If there is only 1 column, only represent mean.
- If there are 2 columns, represent mean (sd) or mean(se). Decimals will understand the number will be formatted as x.x (x.xx).

Value

The same data frame with additional attributes for page features.

Specification

- Check all argument types and possible values.
- Add attributes into `tbl`.

Examples

```
fmt_est(mean(iris$Petal.Length), sd(iris$Petal.Length))
fmt_est(mean(iris$Petal.Length), sd(iris$Petal.Length), digits = c(2, 3))
```

fmt_pct *Format percentage*

Description

Format percentage

Usage

```
fmt_pct(x, digits = 1, pre = "(", post = ")")
```

Arguments

x	A numeric vector.
digits	Number of digits.
pre	Text before the number.
post	Text after the number.

Value

A numeric vector with the expected format.

Examples

```
fmt_pct(c(1, 1.52, 0.3, 100))
```

fmt_pval *Format p-value*

Description

Format p-value

Usage

```
fmt_pval(p, digits = 3, width = 3 + digits)
```

Arguments

p	A numeric vector of p-values.
digits	Digits of each column, i.e., format as x.xxx.
width	Width of each column.

Value

A numeric vector with the expected format.

Examples

```
fmt_pval(c(0.1234, 0.00002))
```

`format_ae_exp_adj` *Format exposure-adjusted AE summary*

Description

Format exposure-adjusted AE summary

Usage

```
format_ae_exp_adj(
  outdata,
  display = c("n", "total_exp", "events", "eaer", "total"),
  digits_total_exp = 2,
  digits_eaer = 2,
  mock = FALSE
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`display`	A character vector of measurement to be displayed: • `n`: Number of subjects exposed. • `total_exp`: Total exposure in person-time. • `events`: Number of AE. • `eaer`: Exposure adjusted event rate. • `total`: Total columns.
`digits_total_exp`	A numeric value of number of digits for total exposure value.
`digits_eaer`	A numeric value of number of digits for exposure-adjusted event rate.
`mock`	A boolean value to display mock table.

Value

A list of analysis raw datasets.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)

analysis_plan <- metalite::plan(
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)
meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
```

```
    name = "any",
    term1 = "",
    term2 = "",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "All AEs"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_summary",
    title = "Adverse Event Summary"
  ) |>
  metalite::meta_build()

# Calculate and format exposure-adjusted event rates
outdata <- meta |>
  prepare_ae_summary(
    population = "apat",
    observation = "wk12",
    parameter = "any;rel;ser"
  ) |>
  extend_ae_summary_eaer(adj_unit = "month")

tbl <- outdata |>
  format_ae_exp_adj()
head(tbl$tbl)
```

`format_ae_listing` *Format AE listing analysis*

Description

Format AE listing analysis

Usage

```
format_ae_listing(outdata, mock = FALSE)
```

Arguments

outdata	An outdata object created by prepare_ae_listing().
mock	A boolean value to display mock table.

Value

An outdata object (a structured list) for AE listing. Key elements include:

- meta: metadata used for analysis.
- population, observation, parameter: selected analysis terms.
- col_name: named vector of display labels for listing columns.
- tbl: formatted listing data frame used by tlf_ae_listing().
- prepare_call and extend_call: recorded function calls for reproducibility.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_listing",
  population = "apat",
  observation = "wk12",
  parameter = "ser"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
```

```
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_listing",
    var_name = c(
      "USUBJID", "ASTDY", "AEDECOD", "ADURN",
      "AESEV", "AESER", "AEREL", "AEOUT"
    ),
    group_by = c("USUBJID", "ASTDY"),
    page_by = "TRTA"
  ) |>
  metalite::meta_build()

outdata <- prepare_ae_listing(
  meta,
  analysis = "ae_listing",
  population = "apat",
  observation = "wk12",
  parameter = "ser"
)
tbl <- outdata |>
  format_ae_listing()
head(tbl$tbl)
```

`format_ae_specific` *Format AE specific analysis*

Description

Format AE specific analysis

Usage

```
format_ae_specific(
  outdata,
  display = c("n", "prop", "total"),
  hide_soc_stats = FALSE,
  digits_prop = 1,
  digits_ci = 1,
  digits_p = 3,
  digits_dur = c(1, 1),
  digits_events = c(1, 1),
  filter_method = c("percent", "count"),
  filter_criteria = 0,
  sort_order = c("alphabetical", "count_des", "count_asc"),
  sort_column = NULL,
  mock = FALSE
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`display`	A character vector of measurement to be displayed: • `n`: Number of subjects with adverse event. • `prop`: Proportion of subjects with adverse event. • `total`: Total columns. • `diff`: Risk difference. • `diff_ci`: 95% confidence interval of risk difference using M&N method. • `diff_p`: p-value of risk difference using M&N method. • `dur`: Average of adverse event duration. • `events_avg`: Average number of adverse event per subject. • `events_count`: Count number of adverse event per subject.
`hide_soc_stats`	A boolean value to hide stats for SOC rows.
`digits_prop`	A numeric value of number of digits for proportion value.
`digits_ci`	A numeric value of number of digits for confidence interval.
`digits_p`	A numeric value of number of digits for p-value.
`digits_dur`	A numeric value of number of digits for average duration of adverse event.
`digits_events`	A numeric value of number of digits for average of number of adverse events per subject.
`filter_method`	A character value to specify how to filter rows: • `count`: Filtered based on participant count. • `percent`: Filtered based percent incidence.
`filter_criteria`	A numeric value to display rows where at least one therapy group has a percent incidence or participant count greater than or equal to the specified value. If `filter_method` is `percent`, the value should be between 0 and 100. If `filter_method` is `count`, the value should be greater than 0.

sort_order	A character value to specify sorting order: • alphabetical: Sort by alphabetical order. • count_des: Sort by count in descending order. • count_asc: Sort by count in ascending order.
sort_column	A character value of group in outdata used to sort a table with.
mock	A boolean value to display mock table.

Value

A list of analysis raw datasets.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)
meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRT01A", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
```

```
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()

outdata <- prepare_ae_specific(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

# Basic example
tbl <- outdata |>
  format_ae_specific()
head(tbl$tbl)

# Filtering
tbl <- outdata |>
  format_ae_specific(
    filter_method = "percent",
    filter_criteria = 10
  )
head(tbl$tbl)

# Display different measurements
tbl <- outdata |>
  extend_ae_specific_events() |>
  format_ae_specific(display = c("n", "prop", "events_count"))
head(tbl$tbl)
```

`format_ae_specific_subgroup` *Format AE specific subgroup analysis*

Description

Format AE specific subgroup analysis

Usage

```
format_ae_specific_subgroup(
  outdata,
  display = c("n", "prop"),
  digits_prop = 1,
  digits_ci = 1,
  digits_p = 3,
  digits_dur = c(1, 1),
  digits_events = c(1, 1),
  mock = FALSE
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`display`	A character vector of measurement to be displayed. • `n`: Number of subjects with adverse event. • `prop`: Proportion of subjects with adverse event. • `total`: Total columns. • `dur`: Average of adverse event duration. • `events_avg`: Average number of adverse event per subject. • `events_count`: Count number of adverse event per subject.
`digits_prop`	A numeric value of number of digits for proportion value.
`digits_ci`	A numeric value of number of digits for confidence interval.
`digits_p`	A numeric value of number of digits for p-value.
`digits_dur`	A numeric value of number of digits for average duration of adverse event.
`digits_events`	A numeric value of number of digits for average of number of adverse event per subjects.
`mock`	Logical. Display mock table or not.

Value

A list of analysis raw datasets for subgroup analysis.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRTA <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRTA", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
```

```
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()

# Prepare and format subgroup analysis
prepare_ae_specific_subgroup(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel",
  subgroup_var = "SEX",
  display_subgroup_total = TRUE
) |>
  format_ae_specific_subgroup()
```

`format_ae_summary` *Format AE summary analysis*

Description

Format AE summary analysis

Usage

```
format_ae_summary(
  outdata,
  display = c("n", "prop", "total"),
  hide_soc_stats = FALSE,
  digits_prop = 1,
  digits_ci = 1,
  digits_p = 3,
  digits_dur = c(1, 1),
  digits_events = c(1, 1),
  filter_method = c("percent", "count"),
  filter_criteria = 0,
  sort_order = c("alphabetical", "count_des", "count_asc"),
  sort_column = NULL,
  mock = FALSE
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`display`	A character vector of measurement to be displayed: • `n`: Number of subjects with adverse event. • `prop`: Proportion of subjects with adverse event. • `total`: Total columns.

- diff: Risk difference.
- diff_ci: 95% confidence interval of risk difference using M&N method.
- diff_p: p-value of risk difference using M&N method.
- dur: Average of adverse event duration.
- events_avg: Average number of adverse event per subject.
- events_count: Count number of adverse event per subject.

hide_soc_stats A boolean value to hide stats for SOC rows.

digits_prop A numeric value of number of digits for proportion value.

digits_ci A numeric value of number of digits for confidence interval.

digits_p A numeric value of number of digits for p-value.

digits_dur A numeric value of number of digits for average duration of adverse event.

digits_events A numeric value of number of digits for average of number of adverse events per subject.

filter_method A character value to specify how to filter rows:

- count: Filtered based on participant count.
- percent: Filtered based percent incidence.

filter_criteria
A numeric value to display rows where at least one therapy group has a percent incidence or participant count greater than or equal to the specified value. If filter_method is percent, the value should be between 0 and 100. If filter_method is count, the value should be greater than 0.

sort_order A character value to specify sorting order:

- alphabetical: Sort by alphabetical order.
- count_des: Sort by count in descending order.
- count_asc: Sort by count in ascending order.

sort_column A character value of group in outdata used to sort a table with.

mock A boolean value to display mock table.

Value

A list of analysis raw datasets.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
```

```
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "any",
    term1 = "",
    term2 = "",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "All AEs"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
```

```
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_summary",
    title = "Adverse Event Summary"
  ) |>
  metalite::meta_build()

outdata <- prepare_ae_summary(meta,
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)
tbl <- outdata |>
  format_ae_summary()
head(tbl$tbl)
```

`gt_ae_specific` *Specific adverse events table in gt format*

Description

Specific adverse events table in gt format

Usage

```
gt_ae_specific(
  outdata,
  meddra_version,
  source,
  analysis,
  footnotes = NULL,
  title = c("analysis", "observation", "population")
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`meddra_version`	A character value of the MedDRA version for this dataset.
`source`	A character value of the data source.
`analysis`	One of analysis name existing at `outdata$meta$analysis`
`footnotes`	A character vector of table footnotes.
`title`	Term "analysis", "observation"and "population") for collecting title from metadata or a character vector of table titles.

Value

A `gt_tbl` object for an AE specific table.

Examples

```
library(gt)

# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRT01A", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
```

```
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()

meta |>
  prepare_ae_specific(
    population = "apat",
    observation = "wk12",
    parameter = "rel"
  ) |>
  format_ae_specific() |>
  gt_ae_specific(
    meddra_version = "24.0",
    source = "Source:  [CDISCpilot: adam-adsl; adae]",
    analysis = "ae_specific"
  )
```

`gt_ae_summary` *AE summary table*

Description

AE summary table

Usage

```
gt_ae_summary(
  outdata,
  source,
  analysis,
  title = c("analysis", "observation", "population"),
  footnotes = NULL
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`source`	A character value of the data source.
`analysis`	One of analysis name existing at `outdata$meta$analysis`
`title`	Term "analysis", "observation"and "population") for collecting title from metadata or a character vector of table titles.
`footnotes`	A character vector of table footnotes.

Value

RTF file and the source dataset for AE summary table.

Examples

```
library(gt)
#' # Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "any",
```

```
    term1 = "",
    term2 = "",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "All AEs"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_summary",
    title = "Adverse Event Summary"
  ) |>
  metalite::meta_build()

outdata <- prepare_ae_summary(meta,
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)
outdata |>
  format_ae_summary() |>
  gt_ae_summary(
    analysis = "ae_summary",
    source = "Source:  [CDISCpilot: adam-adsl; adae]"
  )
```

`metalite_ae_adex` *ADEX dataset*

Description

A dataset containing exposure details.

Usage

```
metalite_ae_adex
```

Format

A data frame with 591 rows and 41 variables.

Value

An analysis data frame.

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/sdtm/cdiscpilot01

`metalite_ae_adexsum` *ADEXSUM dataset*

Description

A dataset containing exposure details in Basic Data Structure (BDS).

Usage

```
metalite_ae_adexsum
```

Format

A data frame with 254 rows and 30 variables.

Value

An analysis data frame.

Source

https://github.com/phuse-org/phuse-scripts/tree/master/data/sdtm/cdiscpilot01

`prepare_ae_listing` *Prepare datasets for AE listing*

Description

Prepare datasets for AE listing

Usage

```
prepare_ae_listing(meta, analysis, population, observation, parameter)
```

Arguments

`meta`	A metadata object created by metalite.
`analysis`	Analysis name from `meta`.
`population`	A character value of population term name. The term name is used as key to link information.
`observation`	A character value of observation term name. The term name is used as key to link information.
`parameter`	A character value of parameter term name. The term name is used as key to link information.

Value

A list of analysis datasets needed for AE listing.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_listing", population = "apat",
  observation = "wk12", parameter = "ser"
)
```

```
meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRT01A", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_listing",
   var_name = c("USUBJID", "ASTDY", "AEDECOD", "ADURN", "AESEV", "AESER", "AEREL", "AEOUT"),
    group_by = c("USUBJID", "ASTDY"),
    page_by = "TRTA"
  ) |>
  metalite::meta_build()

str(prepare_ae_listing(meta, "ae_listing", "apat", "wk12", "ser"))
```

`prepare_ae_specific` *Prepare datasets for AE specific analysis*

Description

Prepare datasets for AE specific analysis

Usage

```
prepare_ae_specific(
  meta,
  population,
```

```
  observation,
  parameter,
  components = c("soc", "par"),
  reference_group = NULL
)
```

Arguments

meta	A metadata object created by metalite.
population	A character value of population term name. The term name is used as key to link information.
observation	A character value of observation term name. The term name is used as key to link information.
parameter	A character value of parameter term name. The term name is used as key to link information.
components	A character vector of components name.
reference_group	An integer to indicate reference group. Default is 2 if there are 2 groups, otherwise, the default is 1.

Value

A list of analysis datasets needed for AE specific analysis.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
```

```
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)
meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRT01A", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()

str(prepare_ae_specific(meta, "apat", "wk12", "rel"))

# Allow to extract each components
prepare_ae_specific(meta, "apat", "wk12", "rel", components = NULL)$data
prepare_ae_specific(meta, "apat", "wk12", "rel", components = "soc")$data
prepare_ae_specific(meta, "apat", "wk12", "rel", components = "par")$data
```

---

prepare_ae_specific_subgroup

Prepare datasets for AE specific subgroup analysis

Description

Prepare datasets for AE specific subgroup analysis

Usage

```
prepare_ae_specific_subgroup(
  meta,
  population,
  observation,
  parameter,
  subgroup_var,
  subgroup_header = c(meta$population[[population]]$group, subgroup_var),
  components = c("soc", "par"),
  display_subgroup_total = TRUE
)
```

Arguments

meta	A metadata object created by metalite.
population	A character value of population term name. The term name is used as key to link information.
observation	A character value of observation term name. The term name is used as key to link information.
parameter	A character value of parameter term name. The term name is used as key to link information.
subgroup_var	A character value of subgroup variable name in observation data saved in meta$data_observation.
subgroup_header	A character vector for column header hierarchy. The first element will be the first level header and the second element will be second level header.
components	A character vector of components name.
display_subgroup_total	Logical. Display total column for subgroup analysis or not.

Value

An outdata object containing analysis datasets needed for AE specific subgroup analysis. The subgroup structure is defined by subgroup_var, subgroup_header, and display_subgroup_total. Key values include:

- group: Treatment groups used to index the statistic columns.
- subgroup: Subgroup levels corresponding to the datasets in out_all.
- display_subgroup_total: Whether the subgroup total is displayed.
- out_all: A named list containing an AE-specific analysis result for each subgroup level and a Total result. Within each result, rows are indexed by order and name, and the commonly used statistics are:
  - n_pop: Number of participants in the population within the subgroup.

- n: Number of participants with an adverse event within the subgroup.
- prop: Proportion of participants with an adverse event within the subgroup.
- diff: Risk difference compared with the reference_group within the subgroup.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRTA <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRTA", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
```

```
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()

outdata <- prepare_ae_specific_subgroup(
  meta, "apat", "wk12", "rel",
  subgroup_var = "SEX"
)
names(outdata$out_all)
```

`prepare_ae_summary` *Prepare datasets for AE summary*

Description

Prepare datasets for AE summary

Usage

```
prepare_ae_summary(meta, population, observation, parameter, ...)
```

Arguments

`meta`	A metadata object created by metalite.
`population`	A character value of population term name. The term name is used as key to link information.
`observation`	A character value of observation term name. The term name is used as key to link information.
`parameter`	A character value of parameter term name. The term name is used as key to link information.
`...`	Additional arguments passed to `prepare_ae_specific()`.

Value

An `outdata` object containing analysis datasets needed for AE summary. Key values include:

- `group`: Treatment groups used to index the statistic columns.
- `order`: Numeric values defining the row order.
- `name`: Descriptive row labels corresponding to `order`.
- `n_pop`: Number of participants in the population.
- `n`: Number of participants with an adverse event.
- `prop`: Proportion of participants with an adverse event.
- `diff`: Risk difference compared with the `reference_group`.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "any",
    term1 = "",
    term2 = "",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "All AEs"
```

```
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_summary",
    title = "Adverse Event Summary"
  ) |>
  metalite::meta_build()

prepare_ae_summary(
  meta,
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)
```

`rate_compare` *Unstratified and stratified Miettinen and Nurminen test*

Description

Unstratified and stratified Miettinen and Nurminen test details can be found in `vignette("rate-compare")`.

Usage

```
rate_compare(
  formula,
  strata,
  data,
  delta = 0,
  weight = c("ss", "equal", "cmh"),
  test = c("one.sided", "two.sided"),
  bisection = 100,
```

```
  eps = 1e-06,
  alpha = 0.05
)
```

Arguments

formula	A symbolic description of the model to be fitted, which has the form y ~ x. Here, y is the numeric vector with values of 0 or 1. x is the group information.
strata	An optional vector of weights to be used in the analysis. If not specified, unstratified MN analysis is used. If specified, stratified MN analysis is conducted.
data	An optional data frame, list, or environment containing the variables in the model. If not found in data, the variables are taken from environment (formula), typically the environment from which rate_compare is called.
delta	A numeric value to set the difference of two group under the null.
weight	Weighting schema used in stratified MN method. Default is "ss": • "equal" for equal weighting. • "ss" for sample size weighting. • "cmh" for Cochran–Mantel–Haenszel's weights.
test	A character string specifying the side of p-value, must be one of "one.sided", or "two.sided".
bisection	The number of sections in the interval used in bisection method. Default is 100.
eps	The level of precision. Default is 1e-06.
alpha	Pre-defined alpha level for two-sided confidence interval.

Value

A data frame with the test results.

References

Miettinen, O. and Nurminen, M, Comparative Analysis of Two Rates. *Statistics in Medicine*, 4(2):213–226, 1985.

Examples

```
# Conduct the stratified MN analysis with sample size weights
treatment <- c(rep("pbo", 100), rep("exp", 100))
response <- c(rep(0, 80), rep(1, 20), rep(0, 40), rep(1, 60))
stratum <- c(rep(1:4, 12), 1, 3, 3, 1, rep(1:4, 12), rep(1:4, 25))
rate_compare(
  response ~ factor(treatment, levels = c("pbo", "exp")),
  strata = stratum,
  delta = 0,
  weight = "ss",
  test = "one.sided",
  alpha = 0.05
)
```

`rate_compare_sum` *Unstratified and stratified Miettinen and Nurminen test in aggregate data level*

Description

Unstratified and stratified Miettinen and Nurminen test in aggregate data level

Usage

```
rate_compare_sum(
  n0,
  n1,
  x0,
  x1,
  strata = NULL,
  delta = 0,
  weight = c("ss", "equal", "cmh"),
  test = c("one.sided", "two.sided"),
  bisection = 100,
  eps = 1e-06,
  alpha = 0.05
)
```

Arguments

`n0`, `n1`	The sample size in the control group and experimental group, separately. The length should be the same as the length for `x0/x1` and `strata`.
`x0`, `x1`	The number of events in the control group and experimental group, separately. The length should be the same as the length for `n0/n1` and `strata`.
`strata`	A vector of stratum indication to be used in the analysis. If `NULL` or the length of unique values of `strata` equals to 1, it is unstratified MN analysis. Otherwise, it is stratified MN analysis. The length of `strata` should be the same as the length for `x0/x1` and `n0/n1`.
`delta`	A numeric value to set the difference of two groups under the null.
`weight`	Weighting schema used in stratified MN method. Default is `"ss"`: • `"equal"` for equal weighting. • `"ss"` for sample size weighting. • `"cmh"` for Cochran-Mantel-Haenszel's weights.
`test`	A character string specifying the side of p-value, must be one of `"one.sided"`, or `"two.sided"`.
`bisection`	The number of sections in the interval used in bisection method. Default is 100.
`eps`	The level of precision. Default is 1e-06.
`alpha`	Pre-defined alpha level for two-sided confidence interval.

Value

A data frame with the test results.

References

Miettinen, O. and Nurminen, M, Comparative Analysis of Two Rates. *Statistics in Medicine*, 4(2):213–226, 1985.

Examples

```
# Conduct the stratified MN analysis with sample size weights
treatment <- c(rep("pbo", 100), rep("exp", 100))
response <- c(rep(0, 80), rep(1, 20), rep(0, 40), rep(1, 60))
stratum <- c(rep(1:4, 12), 1, 3, 3, 1, rep(1:4, 12), rep(1:4, 25))
n0 <- sapply(split(treatment[treatment == "pbo"], stratum[treatment == "pbo"]), length)
n1 <- sapply(split(treatment[treatment == "exp"], stratum[treatment == "exp"]), length)
x0 <- sapply(split(response[treatment == "pbo"], stratum[treatment == "pbo"]), sum)
x1 <- sapply(split(response[treatment == "exp"], stratum[treatment == "exp"]), sum)
strata <- c("a", "b", "c", "d")
rate_compare_sum(
  n0, n1, x0, x1,
  strata,
  delta = 0,
  weight = "ss",
  test = "one.sided",
  alpha = 0.05
)
```

---

react_ae_listing *Interactive AE listing table*

Description

Interactive AE listing table

Usage

```
react_ae_listing(
  outdata,
  default_page_size = 15,
  searchable = TRUE,
  striped = TRUE,
  highlight = TRUE,
  patient_folding = FALSE
)
```

Arguments

outdata	An outdata object created by format_ae_listing().
default_page_size	Number of rows to display per page.
searchable	A boolean value to enable global search. The default is TRUE.
striped	A boolean value to display striped rows. The default is TRUE.
highlight	A boolean value to highlight row on hover. The default is TRUE.
patient_folding	A boolean value to control patient-level folding. The default is FALSE. If TRUE, all rows are hidden by default and only the first column (patient ID column) is filterable; other column filters are disabled. In this mode, records are shown only when the entered value exactly matches a full patient ID in the first column. If FALSE, all rows are displayed and filters are available for all columns.

Value

A reactable htmlwidget object.

Examples

```
if (requireNamespace("reactable", quietly = TRUE) &&
  requireNamespace("forestly", quietly = TRUE)) {
  adsl <- forestly::forestly_adsl
  adae <- forestly::forestly_adae

  adsl$TRT01A <- factor(
    adsl$TRT01A,
    levels = c("Xanomeline Low Dose", "Placebo"),
    labels = c("Low Dose", "Placebo")
  )
  adae$TRTA <- factor(
    adae$TRTA,
    levels = c("Xanomeline Low Dose", "Placebo"),
    labels = c("Low Dose", "Placebo")
  )

  analysis_plan <- metalite::plan(
    analysis = "ae_listing",
    population = "apat",
    observation = "wk12",
    parameter = "rel"
  )

  meta <- metalite::meta_adam(observation = adae, population = adsl) |>
    metalite::define_plan(analysis_plan) |>
    metalite::define_population(
      name = "apat",
      var = c(
        "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
        "SITEID", "SEX", "RACE", "AGE"
```

```
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Related",
    term2 = "",
    subset = AEREL == "RELATED",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_listing",
    var_name = c(
      "USUBJID", "ASTDY", "AEDECOD", "ADURN",
      "AESEV", "AESER", "AEREL", "AEOUT"
    ),
    group_by = c("USUBJID", "ASTDY"),
    page_by = "TRTA"
  ) |>
  metalite::meta_build()

meta |>
  prepare_ae_listing(
    analysis = "ae_listing",
    population = "apat",
    observation = "wk12",
    parameter = "rel"
  ) |>
  format_ae_listing() |>
  react_ae_listing()
}
```

`tlf_ae_exp_adj` *Exposure-adjusted AE summary table*

Description

Exposure-adjusted AE summary table

Usage

```
tlf_ae_exp_adj(
  outdata,
  source,
  analysis,
  col_rel_width = NULL,
  text_font_size = 9,
  orientation = "portrait",
  title = c("analysis", "observation", "population"),
  footnotes = NULL,
  path_outdata = NULL,
  path_outtable = NULL
)
```

Arguments

outdata	An outdata object created by prepare_ae_specific().
source	A character value of the data source.
analysis	One of analysis name existing at outdata$meta$analysis
col_rel_width	Column relative width in a vector e.g. c(2,1,1) refers to 2:1:1. Default is NULL for equal column width.
text_font_size	Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
orientation	Orientation in 'portrait' or 'landscape'.
title	Term "analysis", "observation"and "population") for collecting title from metadata or a character vector of table titles.
footnotes	A character vector of table footnotes.
path_outdata	A character string of the outdata path.
path_outtable	A character string of the outtable path.

Value

RTF file and source dataset for exposure-adjusted AE summary table.

Examples

```
meta <- metalite::meta_example()
meta <- meta |>
  metalite::define_plan(
    plan = metalite::add_plan(meta$plan,
      analysis = "ae_exp_adj", population = "apat",
      observation = "wk12", parameter = "any"
    )
  ) |>
  metalite::define_analysis(
    name = "ae_exp_adj",
    label = "Exposure Adjusted Incident Rate",
```

```
    title = "Exposure-Adjusted Adverse Event Summary"
  ) |>
  metalite::meta_build()
outdata <- meta |>
  prepare_ae_summary(
    population = "apat",
    observation = "wk12",
    parameter = "any;rel;ser"
  ) |>
  extend_ae_summary_eaer(adj_unit = "month")
outdata |>
  format_ae_exp_adj() |>
  tlf_ae_exp_adj(
    source = "Source:  [CDISCpilot: adam-adsl; adae]",
    analysis = "ae_exp_adj",
    path_outdata = tempfile(fileext = ".Rdata"),
    path_outtable = tempfile(fileext = ".rtf")
  )
```

`tlf_ae_listing` *Generate AE listing*

Description

Generate AE listing

Usage

```
tlf_ae_listing(
  outdata,
  footnotes = NULL,
  source = NULL,
  analysis,
  col_rel_width = NULL,
  text_font_size = 9,
  orientation = "landscape",
  path_outdata = NULL,
  path_outtable = NULL
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_listing()`.
`footnotes`	A character vector of table footnotes.
`source`	A character value of the data source.
`analysis`	One of analysis name existing at `outdata$meta$analysis`
`col_rel_width`	Column relative width in a vector e.g. c(2,1,1) refers to 2:1:1. Default is NULL for equal column width.

text_font_size	Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
orientation	Orientation in 'portrait' or 'landscape'.
path_outdata	A character string of the outdata path.
path_outtable	A character string of the outtable path.

Value

RTF file and the source dataset for AE listing.

Examples

```
library(r2rtf)
library(metalite)

meta <- metalite::meta_example()
meta <- meta |>
  metalite::define_plan(
    plan = metalite::add_plan(meta$plan,
      analysis = "ae_listing", population = "apat",
      observation = "wk12", parameter = "ser"
    )
  ) |>
  metalite::define_analysis(
    name = "ae_listing",
   var_name = c("USUBJID", "ASTDY", "AEDECOD", "ADURN", "AESEV", "AESER", "AEREL", "AEOUT"),
    group_by = c("USUBJID", "ASTDY"),
    page_by = "TRTA"
  ) |>
  metalite::meta_build()
prepare_ae_listing(meta, "ae_listing", "apat", "wk12", "ser") |>
  tlf_ae_listing(
    footnotes = "footnote1",
    source = "Source:  [CDISCpilot: adam-adsl; adae]",
    analysis = "ae_listing",
    path_outdata = tempfile(fileext = ".Rdata"),
    path_outtable = tempfile(fileext = ".rtf")
  )
```

---

tlf_ae_specific *Specific adverse events table*

Description

Specific adverse events table

Usage

```
tlf_ae_specific(
  outdata,
  meddra_version,
  source,
  analysis,
  col_rel_width = NULL,
  text_font_size = 9,
  orientation = "portrait",
  footnotes = NULL,
  title = c("analysis", "observation", "population"),
  path_outdata = NULL,
  path_outtable = NULL
)
```

Arguments

outdata	An outdata object created by prepare_ae_specific().
meddra_version	A character value of the MedDRA version for this dataset.
source	A character value of the data source.
analysis	One of analysis name existing at outdata$meta$analysis
col_rel_width	Column relative width in a vector e.g. c(2,1,1) refers to 2:1:1. Default is NULL for equal column width.
text_font_size	Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
orientation	Orientation in 'portrait' or 'landscape'.
footnotes	A character vector of table footnotes.
title	Term "analysis", "observation"and "population") for collecting title from metadata or a character vector of table titles.
path_outdata	A character string of the outdata path.
path_outtable	A character string of the outtable path.

Value

RTF file and the source dataset for AE specific table.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
```

```
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)
meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRT01A", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()
```

```
meta |>
  prepare_ae_specific(
    population = "apat",
    observation = "wk12",
    parameter = "rel"
  ) |>
  format_ae_specific() |>
  tlf_ae_specific(
    source = "Source:  [CDISCpilot: adam-adsl; adae]",
    analysis = "ae_specific",
    meddra_version = "24.0",
    path_outdata = tempfile(fileext = ".Rdata"),
    path_outtable = tempfile(fileext = ".rtf")
  )
```

`tlf_ae_specific_subgroup` *Specific adverse events table for subgroup analysis*

Description

Specific adverse events table for subgroup analysis

Usage

```
tlf_ae_specific_subgroup(
  outdata,
  meddra_version,
  source,
  analysis,
  col_rel_width = NULL,
  text_font_size = 9,
  orientation = "landscape",
  footnotes = NULL,
  title = NULL,
  path_outdata = NULL,
  path_outtable = NULL
)
```

Arguments

`outdata`	An `outdata` object created by `prepare_ae_specific()`.
`meddra_version`	A character value of the MedDRA version for this dataset.
`source`	A character value of the data source.
`analysis`	One of analysis name existing at `outdata$meta$analysis`

col_rel_width	Column relative width in a vector e.g. c(2,1,1) refers to 2:1:1. Default is NULL for equal column width.
text_font_size	Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
orientation	Orientation in 'portrait' or 'landscape'.
footnotes	A character vector of table footnotes.
title	Term "analysis", "observation"and "population") for collecting title from metadata or a character vector of table titles.
path_outdata	A character string of the outdata path.
path_outtable	A character string of the outtable path.

Value

RTF file and the source dataset for AE specific subgroup analysis table.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRTA <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
  analysis = "ae_specific",
  population = "apat",
  observation = "wk12",
  parameter = "rel"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c("USUBJID", "SAFFL", "TRTA", "SITEID", "SEX", "RACE", "AGE"),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
```

```
    var = c(
      "USUBJID", "SAFFL", "TRTA", "SEX", "AEDECOD", "AEBODSYS",
      "AEREL", "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_analysis(
    name = "ae_specific",
    title = "Participants With Drug-Related Adverse Events"
  ) |>
  metalite::meta_build()

prepare_ae_specific_subgroup(meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel",
  subgroup_var = "SEX",
  display_subgroup_total = TRUE
) |>
  format_ae_specific_subgroup() |>
  tlf_ae_specific_subgroup(
    meddra_version = "24.0",
    source = "Source:  [CDISCpilot: adam-adsl; adae]",
    analysis = "ae_specific",
    path_outtable = tempfile(fileext = ".rtf")
  )
```

`tlf_ae_summary` *AE summary table*

Description

AE summary table

Usage

```
tlf_ae_summary(
  outdata,
  source,
```

```
  analysis,
  col_rel_width = NULL,
  text_font_size = 9,
  orientation = "portrait",
  title = c("analysis", "observation", "population"),
  footnotes = NULL,
  path_outdata = NULL,
  path_outtable = NULL
)
```

Arguments

outdata	An outdata object created by prepare_ae_specific().
source	A character value of the data source.
analysis	One of analysis name existing at outdata$meta$analysis
col_rel_width	Column relative width in a vector e.g. c(2,1,1) refers to 2:1:1. Default is NULL for equal column width.
text_font_size	Text font size. To vary text font size by column, use numeric vector with length of vector equal to number of columns displayed e.g. c(9,20,40).
orientation	Orientation in 'portrait' or 'landscape'.
title	Term "analysis", "observation"and "population") for collecting title from metadata or a character vector of table titles.
footnotes	A character vector of table footnotes.
path_outdata	A character string of the outdata path.
path_outtable	A character string of the outtable path.

Value

RTF file and the source dataset for AE summary table.

Examples

```
# Define metadata
adsl <- forestly::forestly_adsl
adae <- forestly::forestly_adae

adsl$TRT01A <- factor(
  adsl$TRT01A,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)
adae$TRTA <- factor(
  adae$TRTA,
  levels = c("Xanomeline Low Dose", "Placebo"),
  labels = c("Low Dose", "Placebo")
)

analysis_plan <- metalite::plan(
```

```
  analysis = "ae_summary",
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)

meta <- metalite::meta_adam(observation = adae, population = adsl) |>
  metalite::define_plan(analysis_plan) |>
  metalite::define_population(
    name = "apat",
    var = c(
      "USUBJID", "SAFFL", "TRT01A", "TRTDUR",
      "SITEID", "SEX", "RACE", "AGE"
    ),
    group = "TRT01A",
    subset = SAFFL == "Y",
    label = "All Participants as Treated"
  ) |>
  metalite::define_observation(
    name = "wk12",
    var = c(
      "USUBJID", "SAFFL", "TRTA", "AEDECOD", "AEBODSYS", "AEREL",
      "AESER", "AEOUT", "AEACN", "AESDTH", "ASTDT", "AENDT"
    ),
    group = "TRTA",
    subset = SAFFL == "Y",
    label = "Weeks 0 to 12"
  ) |>
  metalite::define_parameter(
    name = "any",
    term1 = "",
    term2 = "",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "All AEs"
  ) |>
  metalite::define_parameter(
    name = "rel",
    term1 = "Drug-Related",
    term2 = "",
    subset = AEREL %in% c("POSSIBLE", "PROBABLE"),
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Drug-related AEs"
  ) |>
  metalite::define_parameter(
    name = "ser",
    term1 = "Serious",
    term2 = "",
    subset = AESER == "Y",
    var = "AEDECOD",
    soc = "AEBODSYS",
    label = "Serious AEs"
```

```
  ) |>
  metalite::define_analysis(
    name = "ae_summary",
    title = "Adverse Event Summary"
  ) |>
  metalite::meta_build()

outdata <- prepare_ae_summary(meta,
  population = "apat",
  observation = "wk12",
  parameter = "any;rel;ser"
)
outdata |>
  format_ae_summary() |>
  tlf_ae_summary(
    source = "Source:  [CDISCpilot: adam-adsl; adae]",
    analysis = "ae_summary",
    path_outdata = tempfile(fileext = ".Rdata"),
    path_outtable = tempfile(fileext = ".rtf")
  )
```

Index

∗ **datasets**
    metalite_ae_adex, 29
    metalite_ae_adexsum, 30

extend_ae_specific_duration, 3
extend_ae_specific_events, 3
extend_ae_specific_inference, 4
extend_ae_specific_subgroup, 6
extend_ae_summary_eaer, 7

fmt_ci, 9
fmt_est, 10
fmt_pct, 11
fmt_pval, 11
format_ae_exp_adj, 12
format_ae_listing, 14
format_ae_listing(), *43*
format_ae_specific, 16
format_ae_specific_subgroup, 19
format_ae_summary, 22

gt_ae_specific, 25
gt_ae_summary, 27

metalite_ae_adex, 29
metalite_ae_adexsum, 30

prepare_ae_listing, 31
prepare_ae_listing(), *15*, *46*
prepare_ae_specific, 32
prepare_ae_specific(), *3, 4*, *6*, *12*, *17*, *20*, *22*, *25*, *27*, *37*, *45*, *48*, *50*, *53*
prepare_ae_specific_subgroup, 34
prepare_ae_summary, 37
prepare_ae_summary(), *7*

rate_compare, 39
rate_compare_sum, 41
rate_compare_sum(), *4*
react_ae_listing, 42

tlf_ae_exp_adj, 44
tlf_ae_listing, 46
tlf_ae_listing(), *15*
tlf_ae_specific, 47
tlf_ae_specific_subgroup, 50
tlf_ae_summary, 52