Package ‘metalite.ae’

February 24, 2023

Title Adverse Events Analysis Using 'metalite'

Version 0.1.1

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


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

Encoding UTF-8

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports dplyr, glue, metalite, r2rtf, stats, tidyr

Suggests DescTools, covr, knitr, kableExtra, rmarkdown, readxl, rprojroot, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

NeedsCompilation no

Author Yilong Zhang [aut],
Yujie Zhao [aut, cre],
Benjamin Wang [aut],
Nan Xiao [aut],
Sarad Nepal [aut],
Madhusudhan Ginnaram [aut],
Venkatesh Bural [ctb],
Ruchithbhai Patel [aut],
Brian Lang [aut],
Xuan Deng [aut],
Merck Sharp & Dohme Corp [cph]
extend_ae_specific_duration

Add average duration information for AE specific analysis

Description

Add average duration information for AE specific analysis

Usage

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

Arguments

- **outdata**: A `outdata` object created by `prepare_ae_specific()`.
- **duration_var**: A character value of variable name for AE duration.
- **duration_unit**: A character value of AE duration unit.
Value

A list of analysis raw datasets.

Examples

```r
meta <- meta_ae_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

```r
extend_ae_specific_events(outdata)
```

Arguments

- `outdata`: A `outdata` object created by `prepare_ae_specific()`.

Value

A list of analysis raw datasets.

Examples

```r
meta <- meta_ae_example()
tbl <- prepare_ae_specific(meta,
    population = "apat",
    observation = "wk12",
    parameter = "rel"
) |> 
    extend_ae_specific_events() |> 
    format_ae_specific(display = c("n", "prop", "events"))
head(tbl$tbl)
```
**extend_ae_specific_inference**

*Add inference information for AE specific analysis*

**Description**

Add inference information for AE specific analysis

**Usage**

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

**Arguments**

- `outdata` A `outdata` object created by `prepare_ae_specific()`.
- `ci` A numeric value for the percentile of confidence interval.

**Value**

A list of analysis raw datasets.

**Examples**

```r
meta <- meta_ae_example()
tbl <- prepare_ae_specific(meta,
    population = "apat",
    observation = "wk12",
    parameter = "rel"
  ) |> extend_ae_specific_inference() |> format_ae_specific(display = c("n", "prop", "diff", "diff_ci"))
head(tbl$tbl)
```

---

**fmt_ci**

*Format confidence interval*

**Description**

Format confidence interval

**Usage**

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

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 sd 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

library(dplyr)

x <- datasets::iris |>
  summarise(
    mean = mean(Petal.Length),
    n = n(),
    sd = sd(Petal.Length)
  )
fmt_est(x$mean, x$sd)

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(0.1234)

format_ae_specific  
Format AE specific analysis

Description
Format AE specific analysis

Usage
format_ae_specific(
  outdata,
  display = c("n", "prop", "total"),
  digits_prop = 1,
  digits_ci = 1,
  digits_p = 3,
  digits_dur = c(1, 1),
  digits_events = c(1, 1),
  mock = FALSE
)
Arguments

- **outdata**: A outdata object created by `prepare_ae_specific()`.
- **display**: A character vector of measurement to be displayed:
  - `n`: Number of subjects with AE.
  - `prop`: Proportion of subjects with AE.
  - `total`: Total columns.
  - `diff`: Risk difference.
  - `diff_ci`: 95% confidence interval of risk difference using M&N method.
  - `dur`: Average of AE duration.
  - `events`: Average number of AE 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 AE.
- **digits_events**: A numeric value of number of digits for average of number of AE per subjects.
- **mock**: A boolean value to display mock table.

Value

A list of analysis raw datasets.

Examples

```r
meta <- meta_ae_example()

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

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

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

Description
Format AE summary analysis

Usage

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

Arguments

outdata A outdata object created by prepare_ae_specific().
display A character vector of measurement to be displayed:
  • n: Number of subjects with AE.
  • prop: Proportion of subjects with AE.
  • 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 AE duration.
  • events: Average number of AE 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 AE.digits_events A numeric value of number of digits for average of number of AE per subjects.mock A boolean value to display mock table.

Value

A list of analysis raw datasets.
**Examples**

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

---

**meta_ae_example**  
*Create an example meta_adam object*

**Description**

This function is only for illustration purpose. r2rtf is required.

**Usage**

```r
meta_ae_example()
```

**Value**

A metadata object.

**Examples**

```r
meta_ae_example()
```

---

**prepare_ae_listing**  
*Prepare datasets for AE specific analysis*

**Description**

Prepare datasets for AE specific analysis

**Usage**

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>A metadata object created by metalite.</td>
</tr>
<tr>
<td>analysis</td>
<td>Analysis name from meta.</td>
</tr>
<tr>
<td>population</td>
<td>A character value of population term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
<tr>
<td>observation</td>
<td>A character value of observation term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
<tr>
<td>parameter</td>
<td>A character value of parameter term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
</tbody>
</table>

Value

A list of analysis raw datasets.

Examples

```r
meta <- meta_ae_example()
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

```r
prepare_ae_specific(
  meta,
  population,
  observation,
  parameter,
  components = c("soc", "par"),
  reference_group = NULL
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>A metadata object created by metalite.</td>
</tr>
<tr>
<td>population</td>
<td>A character value of population term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
<tr>
<td>observation</td>
<td>A character value of observation term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
</tbody>
</table>
prepare_ae_specific_subgroup

Prepares datasets for AE specific analysis.

**Description**

Prepare datasets for AE specific analysis.

**Usage**

```r
prepare_ae_specific_subgroup(
  meta, population, observation, parameter, subgroup_var,
  subgroup_header = 1, components = c("soc", "par"),
  display_total = TRUE, display_subgroup_total = TRUE
)
```

**Parameters**

- `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 raw datasets.

**Examples**

```r
meta <- meta_ae_example()
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
```
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**: An integer value of column header for subgroup.
- **components**: A character vector of components name.
- **display_total**: A character vector of components name.
- **display_subgroup_total**: A character vector of components name.

Value

A list of analysis raw datasets.

Examples

```r
meta <- meta_ae_example()
prepare_ae_specific_subgroup(
  meta,
  population = "apat",
  observation = "wk12",
  parameter = "rel",
  subgroup_var = "RACE"
)
```

Description

Prepare datasets for AE specific analysis

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>A metadata object created by metalite.</td>
</tr>
<tr>
<td>population</td>
<td>A character value of population term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
<tr>
<td>observation</td>
<td>A character value of observation term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
<tr>
<td>parameter</td>
<td>A character value of parameter term name. The term name is used as key to</td>
</tr>
<tr>
<td></td>
<td>link information.</td>
</tr>
<tr>
<td>...</td>
<td>Additional arguments passed to <code>prepare_ae_specific()</code>.</td>
</tr>
</tbody>
</table>

Value

A list of analysis raw datasets.

Examples

```r
meta <- meta_ae_example()
preserve_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

```r
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 \( y \sim x \). 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 number of sections in the interval used in bisection method. Default is 1e-06.

alpha Pre-defined alpha level for two-sided confidence interval.

Value

A data frame with the test results.

References


Examples

```
# Conduct the stratified MN analysis with sample size weights
# treatment <- c(rep("pbo", 100), rep("exp", 100))
treatment <- c(rep(0, 80), rep(1, 20), rep(0, 40), rep(1, 60))
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


Examples

```r
# 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
)
```

---

tlf_ae_listing

**Specific adverse events table**

Description

Specific adverse events table

Usage

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

outdata A outdata list created by `prepare_ae_listing()`.
footnotes A character vector of table footnotes.
source A character value of the data source.
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 source dataset for AE listing.

Examples

```r
library(r2rtf)
library(metalite)

meta <- meta_ae_example()
prepare_ae_listing(meta, "ae_listing", "apat", "wk12", "ser") |> 
tlf_ae_listing(
  footnotes = "footnote1",
  source = "Source: [CDISCpilot: adam-adsl; adae]",
  path_outdata = tempfile(fileext = ".Rdata"),
  path_outtable = tempfile(fileext = ".rtf")
)
```

---

**tlf_ae_specific**

*Specific adverse events table*

Description

Specific adverse events table

Usage

```r
tlf_ae_specific(
  outdata, 
  meddra_version, 
  source, 
  col_rel_width = NULL, 
  text_font_size = 9, 
  orientation = "portrait",
)```
Arguments

outdata A outdata list created from `prepare_ae_specific()`.
meddra_version A character value of the MedDRA version for this dataset.
source A character value of the data source.
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 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 source dataset for AE specific table.

Examples

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

AE summary table

Usage

```r
tlf_ae_summary(
  outdata,
  source,
  col_rel_width = NULL,
  text_font_size = 9,
  orientation = "portrait",
  footnotes = NULL,
  path_outdata = NULL,
  path_outtable = NULL
)
```

Arguments

- **outdata**: A outdata list created from `prepare_ae_specific()`.
- **source**: A character value of the data source.
- **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.
- **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 AE listing.

Examples

```r
meta <- meta_ae_example()
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]",
  path_outdata = tempfile(fileext = "Rdata"),
  path_outtable = tempfile(fileext = "rtf")
)
Index

extend_ae_specific_duration, 2
extend_ae_specific_events, 3
extend_ae_specific_inference, 4

fmt_ci, 4
fmt_est, 5
fmt_pct, 6
fmt_pval, 7
format_ae_specific, 7
format_ae_summary, 9

meta_ae_example, 10

prepare_ae_listing, 10
prepare_ae_listing(), 18
prepare_ae_specific, 11
prepare_ae_specific(), 2–4, 8, 9, 14, 19, 20
prepare_ae_specific_subgroup, 12
prepare_ae_summary, 13

rate_compare, 14
rate_compare_sum, 16

tlf_ae_listing, 17
tlf_ae_specific, 18
tlf_ae_summary, 20