Package ‘metalite’
August 10, 2023

Title ADaM Metadata Structure
Version 0.1.3
Description A metadata structure for clinical data analysis and reporting based on Analysis Data Model (ADaM) datasets.
The package simplifies clinical analysis and reporting tool development by defining standardized inputs, outputs, and workflow.
The package can be used to create analysis and reporting planning grid, mock table, and validated analysis and reporting results based on consistent inputs.
License GPL-3
BugReports https://github.com/Merck/metalite/issues
Encoding UTF-8
VignetteBuilder knitr
Depends R (>= 4.1.0)
Imports glue, rlang, utils, stats
Suggests ggplot2, covr, dplyr, gt, knitr, r2rtf, rmarkdown, testthat
(>= 3.0.0), tibble
Config/testthat/edition 3
RoxygenNote 7.2.3
NeedsCompilation no
Author Yilong Zhang [aut],
Yujie Zhao [aut, cre],
Nan Xiao [aut],
Benjamin Wang [ctb],
Brian Lang [ctb],
Howard Baek [ctb],
Ruchitbhai Patel [ctb],
Madhusudhan Ginnaram [ctb],
Sarad Nepal [ctb],
Venkatesh Burla [ctb],
Merck Sharp & Dohme Corp [cph]

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

Repository CRAN

Date/Publication 2023-08-10 17:00:02 UTC

---

**R topics documented:**

- `adam_mapping` ................................................................. 3
- `add_plan` ................................................................. 4
- `assign_label` ............................................................... 5
- `collect_adam_mapping` .................................................. 6
- `collect_dataname` .......................................................... 6
- `collect_n_subject` .......................................................... 7
- `collect_observation_index` .............................................. 8
- `collect_observation_record` ............................................. 9
- `collect_population` ....................................................... 10
- `collect_population_id` .................................................. 10
- `collect_population_index` .............................................. 11
- `collect_population_record` ............................................. 12
- `collect_title` .............................................................. 12
- `default_apply` ............................................................. 13
- `define_analysis` ........................................................... 14
- `define_observation` ....................................................... 15
- `define_parameter` .......................................................... 16
- `define_plan` ................................................................. 17
- `define_population` ........................................................ 17
- `get_label` ................................................................. 19
- `meta_adam` ................................................................. 19
- `meta_add_total` ............................................................. 20
- `meta_build` ................................................................. 20
- `meta_example` ............................................................... 21
- `meta_example_exploration` .............................................. 22
- `meta_run` ................................................................. 23
- `meta_split` ................................................................. 23
- `n_subject` ................................................................. 24
- `outdata` ................................................................. 25
- `plan` ................................................................. 26
- `print.meta_adam` ........................................................... 27
- `spec_analysis_population` .............................................. 28
- `spec_call_program` ........................................................ 28
- `spec_filename` ............................................................. 29
- `spec_title` ................................................................. 29
- `update_adam_mapping` .................................................... 30

**Index** 31
adam_mapping

Construct ADaM mappings

Description

ADaM mappings describe how variables and meta information in the ADaM data are mapped to standardized term.

Usage

adam_mapping(
  name,
  id = NULL,
  group = NULL,
  var = NULL,
  subset = NULL,
  label = NULL,
  ...
)

Arguments

- name: A character value of term name. The term name is used as key to link information.
- id: A character value of subject identifier variable name in an ADaM dataset.
- group: A character vector of group variable names in an ADaM dataset.
- var: A character vector of useful variable names in an ADaM dataset.
- subset: An expression to identify analysis records. See `base::subset()`.
- label: A character value of analysis label.
- ... Additional variables.

Details

The design is inspired by `ggplot2::aes()`.

Value

A list with class `adam_mapping`. Components of the list are either quosures or constants.

Examples

```r
adam_mapping(
  name = "apat",
  id = "USUBJID",
  group = "TRT01A",
  subset = TRTFL == "Y",
  label = "All Participants as Treated"
)
```
add_plan

---

**add_plan**

*Add additional analysis plan*

**Description**

Add additional analysis plan

**Usage**

`add_plan(plan, analysis, population, observation, parameter, ...)`

**Arguments**

- `plan` A `meta_plan` object.
- `analysis` A character value of analysis term name. The term name is used as key to link information.
- `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.

**Value**

A data frame containing analysis plans with new plans added.

**Examples**

```r
plan("ae_summary",
    population = "apat",
    observation = c("wk12", "wk24"),
    parameter = "any;rel"
) |> add_plan("ae_specific",
    population = "apat",
    observation = c("wk12", "wk24"),
    parameter = c("any", "rel")
)```

assign_label

A function to assign labels to a data frame

Description

A function to assign labels to a data frame

Usage

assign_label(data, var = names(data), label = names(data))

Arguments

data A data frame.
var The variables to assign labels.
label The labels to be assigned.

Details

• Case 1: If the variable’s label is already define in the original data frame but not redefined in assign_label(...), its original labels will be kept.
• Case 2: If the variable’s label is already define in the original data frame but re-defined by assign_label(...), its labels will be re-defined.
• Case 3: If the variable’s label is not define in the original data frame but it is defined by assign_label(...), its labels will added.
• Case 4: If the variable’s label is not define in the original data frame, neither was it defined by assign_label(...), its labels will be the variable name itself.

Value

A data frame with labels updated.

Examples

assign_label(r2rtf::r2rtf_adae) |> head()
assign_label(
  r2rtf::r2rtf_adae,
  var = "USUBJID",
  label = "Unique subject identifier"
) |> head()
**collect_adam_mapping**  
*Collect adam_mapping from meta_adam by name*

**Description**  
Collect adam_mapping from meta_adam by name

**Usage**  
`collect_adam_mapping(meta, name)`

**Arguments**
- `meta`: A `meta_adam` object.
- `name`: A keyword value.

**Value**
An `adam_mapping` class object containing the definition of the search variable in `name`.

**Examples**
```r
meta <- meta_example()
collect_adam_mapping(meta, "apat")
```

**collect_dataname**  
*Collect specification for dataset name*

**Description**  
Collect specification for dataset name

**Usage**
`collect_dataname(meta)`

**Arguments**
- `meta`: A `meta_adam` object.

**Value**
A vector of character strings containing the name of the population/observation.

**Examples**
```r
meta <- meta_example()
collect_dataname(meta)
```
**collect_n_subject**  
*Collect number of subjects and its subset condition*

**Description**

Collect number of subjects and its subset condition

**Usage**

```r
collect_n_subject(
  meta, 
  population, 
  parameter, 
  listing = FALSE, 
  histogram = FALSE, 
  var_listing = NULL, 
  remove_blank_group = FALSE, 
  type = "Subjects", 
  use_na = c("ifany", "no", "always"), 
  display_total = TRUE 
)
```

**Arguments**

- `meta`  
  A `meta_adam` object.

- `population`  
  A character value of population 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.

- `listing`  
  A logical value to display drill down listing per row.

- `histogram`  
  A logical value to display histogram by group.

- `var_listing`  
  A character vector of additional variables included in the listing.

- `remove_blank_group`  
  A logical value to remove a group with all missing value of a parameter.

- `type`  
  A character value to control title name, e.g., Subjects or Records.

- `use_na`  
  A character value for whether to include NA values in the table. See the `useNA()` argument in `base::table()` for more details.

- `display_total`  
  A logical value to display total column.

**Value**

A list containing number of subjects and its subset condition.
Examples

```r
suppressWarnings(
  meta <- meta_example() |>  
    define_parameter(name = "sex", var = "SEX", label = "Sex")  
)  
collect_n_subject(meta, "apat", "sex")
```

---

**collect_observation_index**

Collect observation record index from observation dataset

**Description**

Collect observation record index from observation dataset

**Usage**

`collect_observation_index(meta, population, observation, parameter)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>meta</code></td>
<td>A meta_adam object.</td>
</tr>
<tr>
<td><code>population</code></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><code>observation</code></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><code>parameter</code></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 vector of patient index within the observation group.

**Examples**

```r
meta <- meta_example()
collect_observation_index(meta, "apat", "wk12", "ser")
```
collect_observation_record

Collect observation record from observation dataset

Description

The key variables used in id, group, and subset are displayed by default.

Usage

collect_observation_record(
  meta, 
  population, 
  observation, 
  parameter, 
  var = NULL
)

Arguments

meta A meta_adam object.
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.
var A character vector of additional variables to be displayed in the output.

Value

A data frame of the observation dataset.

Examples

meta <- meta_example()
collect_observation_record(meta, "apat", "wk12", "ser")
collect_observation_record(meta, "apat", "wk12", "ser", var = "AEDECD0")
**collect_population**  
*Collect specification for population definition*

**Description**
Collect specification for population definition

**Usage**
```r
collect_population(meta, population, observation = NULL, parameter = NULL)
```

**Arguments**
- **meta**  
  A `meta_adam` object.
- **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 covering the filter of population, observation (if given) and parameter (if given).

**Examples**
```r
meta <- meta_example()
collect_population(meta, "apat")
collect_population(meta, "apat", "wk12")
collect_population(meta, "apat", "wk12", "ser")
```

---

**collect_population_id**  
*Collect subject identifier information from population dataset*

**Description**
Collect subject identifier information from population dataset

**Usage**
```r
collect_population_id(meta, population)
```
**Arguments**

- **meta**: A `meta_adam` object.
- **population**: A character value of population term name. The term name is used as key to link information.

**Value**

A vector of patient ID within the population group.

**Examples**

```r
meta <- meta_example()
head(collect_population_id(meta, "apat"))
```

---

**Description**

Collect population record index from population dataset

**Usage**

```r
collect_population_index(meta, population)
```

**Arguments**

- **meta**: A `meta_adam` object.
- **population**: A character value of population term name. The term name is used as key to link information.

**Value**

A vector of patient index within the population group.

**Examples**

```r
meta <- meta_example()
head(collect_population_index(meta, "apat"))
```
**collect_population_record**

*Collect population record from population dataset*

**Description**

The key variables used in id, group, and subset are displayed by default.

**Usage**

```r
collect_population_record(meta, population, var = NULL)
```

**Arguments**

- `meta` A `meta_adam` object.
- `population` A character value of population term name. The term name is used as key to link information.
- `var` A character vector of additional variables to be displayed in the output.

**Value**

A data frame containing the variables in the population dataset.

**Examples**

```r
meta <- meta_example()
head(collect_population_record(meta, "apat"))
head(collect_population_record(meta, "apat", var = "AGE"))
```

**collect_title**

*Collect specification for title*

**Description**

Collect specification for title

**Usage**

```r
collect_title(
  meta,
  population,
  observation,
  parameter,
  analysis,
  title_order = c("analysis", "observation", "population")
)
```
Arguments

- **meta**
  A meta_adam object.

- **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.

- **analysis**
  A character value of analysis term name. The term name is used as key to link information.

- **title_order**
  A character vector to define the order of title from each component.

Value

A vector of strings to compose the table captions.

Examples

```r
meta <- meta_example()
collect_title(meta, "apat", "wk12", "ser", "ae_summary")
collect_title(meta, "apat", "wk12", "ser", "ae_specific")
```

---

**default_apply**  Apply default values to ADaM mappings

**Description**

Apply default values to ADaM mappings

**Usage**

default_apply(x)

**Arguments**

- **x**
  An adam_mapping object.

**Value**

Similar to the input, but with the missing values updated to the default values.

**Examples**

default_apply(adam_mapping(name = "apat"))
define_analysis

Define analysis function meta information for ADaM dataset

Description

Define analysis function meta information for ADaM dataset

Usage

define_analysis(meta, name, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>A meta_adam object.</td>
</tr>
<tr>
<td>name</td>
<td>A character value of term name. The term name is used as key to link information.</td>
</tr>
<tr>
<td>...</td>
<td>Additional variables.</td>
</tr>
</tbody>
</table>

Value

A metadata object with analysis details defined.

Examples

```r
plan <- plan(
  analysis = "ae_summary", population = "apat",
  observation = c("wk12", "wk24"), parameter = "any;rel;ser"
)

meta_adam(
  population = r2rtf::r2rtf_adsl,
  observation = r2rtf::r2rtf_adae
) |> define_plan(plan = plan) |> define_analysis(
  name = "ae_summary",
  title = "Summary of Adverse Events"
)
```
define_observation

Define analysis observation meta information for ADaM dataset

Description

Define analysis observation meta information for ADaM dataset

Usage

```r
define_observation(
    meta,
    name,
    id = "USUBJID",
    group = NULL,
    var = NULL,
    subset = NULL,
    label = NULL,
    ...
)
```

Arguments

- **meta**: A `meta_adam` object.
- **name**: A character value of term name. The term name is used as key to link information.
- **id**: A character value of subject identifier variable name in an ADaM dataset.
- **group**: A character vector of group variable names in an ADaM dataset.
- **var**: A character vector of useful variable names in an ADaM dataset.
- **subset**: An expression to identify analysis records. See `base::subset()`.
- **label**: A character value of analysis label.
- **...**: Additional variables.

Value

A metadata object with observation defined.

Examples

```r
plan <- plan(
    analysis = "ae_summary", population = "apat",
    observation = c("wk12", "wk24"), parameter = "any;rel;ser"
)

meta_adam(
    population = r2rtf::r2rtf_ads1,
    observation = r2rtf::r2rtf_adae
)
define_parameter

Define analysis parameter meta information for ADaM dataset

Description
Define analysis parameter meta information for ADaM dataset

Usage
define_parameter(meta, name, subset = NULL, ...)

Arguments

- **meta**: A meta_adam object.
- **name**: A character value of term name. The term name is used as key to link information.
- **subset**: An expression to identify analysis records. See `base::subset()`.
- **...**: Additional variables.

Value
A metadata object with parameters defined.

Examples

```r
plan <- plan(
  analysis = "ae_summary", population = "apat",
  observation = c("wk12", "wk24"), parameter = "any;rel;ser"
)

meta_adam(
  population = r2rtf::r2rtf_adsl,
  observation = r2rtf::r2rtf_adae
) |> define_plan(plan = plan) |> define_parameter(
  name = "rel",
  subset = AEREL %in% c("POSSIBLE", "PROBABLE")
)
```
define_plan

Define analysis plan meta information for ADaM dataset

Description

Define analysis plan meta information for ADaM dataset

Usage

define_plan(meta, plan)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>A meta_adam object.</td>
</tr>
<tr>
<td>plan</td>
<td>A data frame for analysis plan.</td>
</tr>
</tbody>
</table>

Value

A metadata object with plans defined.

Examples

```r
plan <- plan(
  analysis = "ae_summary", population = "apat",
  observation = c("wk12", "wk24"), parameter = "any;rel;ser"
)

meta_adam(
  population = r2rtf::r2rtf_adsl,
  observation = r2rtf::r2rtf_adae
) |> define_plan(plan)
```

define_population

Define analysis population meta information for ADaM dataset

Description

Define analysis population meta information for ADaM dataset
define_population

Usage

```r
define_population(
  meta,
  name,
  id = "USUBJID",
  group = NULL,
  var = NULL,
  subset = NULL,
  label = NULL,
  ...
)
```

Arguments

- `meta`: A `meta_adam` object.
- `name`: A character value of term name. The term name is used as key to link information.
- `id`: A character value of subject identifier variable name in an ADaM dataset.
- `group`: A character vector of group variable names in an ADaM dataset.
- `var`: A character vector of useful variable names in an ADaM dataset.
- `subset`: An expression to identify analysis records. See `base::subset()`.
- `label`: A character value of analysis label.
- `...`: Additional variables.

Value

A metadata object with population defined.

Examples

```r
plan <- plan(
  analysis = "ae_summary", population = "apat",
  observation = c("wk12", "wk24"), parameter = "any;rel;ser"
)

meta_adam(
  population = r2rtf::r2rtf_ads1,
  observation = r2rtf::r2rtf_aadae
) |> 
define_plan(plan) |> 
define_population(name = "apat")
```
get_label

A function to get the labels of data frame columns

Description
A function to get the labels of data frame columns

Usage
get_label(data)

Arguments
data A data frame.

Value
Labels of the input data frame.

Examples
get_label(r2rtf::r2rtf_adae)

meta_adam
Create a metadata representation for ADaM data analysis

Description
Create a metadata representation for ADaM data analysis

Usage
meta_adam(observation, population = observation)

Arguments
observation A data frame for observation level data.
population A data frame for population level data. Default is the same as observation.

Value
An initialized metadata object with observation and population defined.

Examples
meta_adam(observation = r2rtf::r2rtf_adae, population = r2rtf::r2rtf_adae)
**meta_add_total**  
*Add duplicate data to enable a total group*

**Description**
Add duplicate data to enable a total group

**Usage**
```
meta_add_total(meta, total = "Total")
```

**Arguments**
- `meta`: A metalite object.
- `total`: A character value of total group name.

**Value**
A metadata object with a total group added.

**Examples**
```
x <- meta_add_total(meta_example())
# A `Total` group is added
table(x$data_population$TRTA)
```

---

**meta_build**  
*Build complete meta information*

**Description**
Build complete meta information

**Usage**
```
meta_build(meta)
```

**Arguments**
- `meta`: A meta_adam object.

**Value**
A composed metadata object.
Examples

```r
meta_adam(
  observation = r2rtf::r2rtf_adae,
  population = r2rtf::r2rtf_adsl
) |> # define analysis plan
define_plan(
  plan(
    analysis = "ae_summary",
    population = "apat",
    observation = c("wk12"),
    parameter = "any;rel"
  )
) |> # define population
define_population(
  name = "apat",
  group = "TRT01A",
  subset = SAFFL == "Y"
) |> # define observation
define_observation(
  name = "wk12",
  group = "TRTA",
  subset = SAFFL == "Y",
  label = "Weeks 0 to 12"
) |> # define parameter - rel
define_parameter(
  name = "rel",
  subset = AEREL %in% c("POSSIBLE", "PROBABLE")
) |> # define analysis
define_parameter(
  name = "rel",
  subset = AEREL %in% c("POSSIBLE", "PROBABLE")
) |> meta_build()
```

Description

This function is only for illustration purposes. The r2rtf package is required.

Usage

```r
meta_example()
```
meta_example_exploration

Value
A metadata object.

Examples
meta_example()

meta_example_exploration

Create a data exploration meta_adam object

Description
Create a data exploration meta_adam object

Usage
meta_example_exploration(
data,
group,
name = "ase",
subset = NULL,
label = "All Subjects Enrolled"
)

Arguments
data A data frame.
group A character vector of group variable names in an ADaM dataset.
name A character value of term name. The term name is used as key to link information.
subset An expression to identify analysis records. See base::subset().
label A character value of analysis label.

Value
A metadata object.

Examples
meta <- meta_example_exploration(r2rtf::r2rtf_ads1, group = "TRT01A")
collect_n_subject(meta, "ase", "AGE")
collect_n_subject(meta, "ase", "SEX")
## meta_run

Execute analysis based on the analysis plan

### Description

Execute analysis based on the analysis plan

### Usage

`meta_run(meta, i = NULL, ...)`

### Arguments

- `meta` A `meta_adam` object.
- `i` A vector of integers to indicate i-th analysis in `meta$plan`.
- `...` Additional arguments passed to `spec_call_program()`.

### Value

Executed analysis based on the analysis plan.

### Examples

```r
meta <- meta_example()
ae_summary <- function(...) {
  paste("results of", deparse(match.call(), nlines = 1))
}
ae_specific <- function(...) {
  paste("results of", deparse(match.call(), nlines = 1))
}
meta_run(meta)
meta_run(meta, i = 2)
```

## meta_split

Split metadata into groups

### Description

Split metadata into groups

### Usage

`meta_split(meta, by)`
**Arguments**

- **meta**: A meta_adam object.
- **by**: A character variable name both in population level and observation level data of a metadata object.

**Value**

A metadata object split by the input variable.

**Examples**

```r
meta_example() |> meta_split("RACE")
```

<table>
<thead>
<tr>
<th>n_subject</th>
<th>Count number of unique subjects</th>
</tr>
</thead>
</table>

**Description**

Count number of unique subjects

**Usage**

```r
n_subject(
  id,
  group,
  par = NULL,
  na = "Missing",
  use_na = c("ifany", "no", "always")
)
```

**Arguments**

- **id**: A character vector of subject identifier.
- **group**: A factor vector of group name.
- **par**: A character vector of parameter name.
- **na**: A character string used to label missing values. Defaults to "Missing".
- **use_na**: A character value for whether to include NA values in the table. See the useNA argument in `base::table()` for more details.

**Value**

A data frame summarizing the number of unique subjects in different arms.
Examples

```r
library(r2rtf)

r2rtf_adae$TRTA <- factor(r2rtf_adae$TRTA)
r2rtf_adae$SEX[1:5] <- NA

n_subject(r2rtf_adae$USUBJID, r2rtf_adae$TRTA)

n_subject(r2rtf_adae$USUBJID, r2rtf_adae$TRTA, r2rtf_adae$SEX)
n_subject(r2rtf_adae$USUBJID, r2rtf_adae$TRTA, r2rtf_adae$SEX, use_na = "always")
n_subject(r2rtf_adae$USUBJID, r2rtf_adae$TRTA, r2rtf_adae$SEX, na = "Null")
```

Description

The `outdata` class defines a standard output format for analysis and reporting.

Usage

```r
outdata(
  meta,
  population, 
  observation,
  parameter,
  n, 
  order, 
  group, 
  reference_group,
  ...
)
```

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.
- `n` A data frame for number of subjects in each criteria.
- `order` A numeric vector of row display order.
- `group` A character vector of group variable names in an ADaM dataset.
- `reference_group` A numeric value to indicate reference group in levels of group.
- `...` Additional variables to save to `outdata`. 
Details

The design is inspired by ggplot2::aes().

Value

A list with class outdata. Components of the list are either quosures or constants.

Examples

outdata(
  meta = meta_example(),
  population = "apat",
  observation = "wk12",
  parameter = "rel",
  n = data.frame(
    TRTA = c("Placebo", "Xanomeline Low Dose", "Xanomeline High Dose"),
    n = c(86, 84, 84)
  ),
  group = "TRTA",
  reference_group = 1,
  order = 1:3
)

plan

Create a analysis plan from all combination of variables

Description

This function is a wrapper of base::expand.grid().

Usage

plan(analysis, population, observation, parameter, mock = 1, ...)

Arguments

- **analysis**: A character value of analysis term name. The term name is used as key to link information.
- **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.
- **mock**: A numeric value of mock table number.
- **...**: Additional arguments.
Value

A data frame containing the analysis plan.

Examples

# Example 1
# Create an analysis plan of AE summary
# with any AE, drug-related AE, and serious AE
plan(
  analysis = "ae_summary",
  population = "apat",
  observation = c("wk12", "wk24"),
  parameter = "any;rel;ser"
)

# Example 2
# Create an analysis plan of AE specific
# with any AE, drug-related AE, and serious AE
plan(
  analysis = "ae_specific",
  population = "apat",
  observation = c("wk12", "wk24"),
  parameter = c("any", "rel", "ser")
)

print.meta_adam

Print a metadata object with its population, observation, and analysis plans

Description

Print a metadata object with its population, observation, and analysis plans

Usage

## S3 method for class 'meta_adam'
print(x, ...)

Arguments

x An object returned by meta_adam().

... Additional parameters for print() (not used).

Value

A printed summary of the metadata.

Examples

meta_adam(observation = r2rtf::r2rtf_aeae, population = r2rtf::r2rtf_aeae) |> print()
spec_analysis_population

Specification for population definition

Description

Specification for population definition

Usage

spec_analysis_population(meta)

Arguments

meta A meta_adam object.

Value

A vector of character strings containing the populations used in the order of the analysis plans.

Examples

meta <- meta_example()
spec_analysis_population(meta)

---

spec_call_program

Specification for analysis call program

Description

Specification for analysis call program

Usage

spec_call_program(meta, ...)

Arguments

meta A meta_adam object.
... Additional arguments used in all call programs.

Value

A vector of character strings containing the call program in the order of the analysis plans.
### spec_filename

**Description**

Specification for analysis output filename

**Usage**

```r
spec_filename(meta)
```

**Arguments**

- `meta` A `meta_adam` object.

**Value**

A vector of character strings containing the RTF file names.

**Examples**

```r
meta <- meta_example()
spec_filename(meta)
```

---

### spec_title

**Description**

Specification for analysis title

**Usage**

```r
spec_title(meta)
```

**Arguments**

- `meta` A `meta_adam` object.

**Value**

A vector of character strings containing the table captions in the order of the analysis plans.
update_adam_mapping

**Description**

Update mapping rule in `adam_mapping`

**Usage**

`update_adam_mapping(meta, name, ...)`

**Arguments**

- `meta`: A `meta_adam` object.
- `name`: A vector of keywords.
- `...`: Additional variables to be added in the mapping rule among those keywords.

**Value**

A metadata object with the input updated.

**Examples**

```r
meta <- meta_example()
meta <- update_adam_mapping(meta, names(meta$parameter), start_date = "ASTDT")
collect_adam_mapping(meta, "ser")
```
Index

adam_mapping, 3
add_plan, 4
assign_label, 5
base::expand.grid(), 26
base::subset(), 3, 15, 16, 18, 22
base::table(), 7, 24
collect_adam_mapping, 6
collect_dataname, 6
collect_n_subject, 7
collect_observation_index, 8
collect_observation_record, 9
collect_population, 10
collect_population_id, 10
collect_population_index, 11
collect_population_record, 12
collect_title, 12
default_apply, 13
define_analysis, 14
define_observation, 15
define_parameter, 16
define_plan, 17
define_population, 17
get_label, 19
meta_adam, 19
meta_adam(), 27
meta_add_total, 20
meta_build, 20
meta_example, 21
meta_example_exploration, 22
meta_run, 23
meta_split, 23
n_subject, 24
outdata, 25
plan, 26
print(), 27
print.meta_adam, 27
spec_analysis_population, 28
spec_call_program, 28
spec_filename, 29
spec_title, 29
update_adam_mapping, 30