

# Package ‘mds’

September 26, 2018

**Type** Package

**Title** Medical Devices Surveillance

**Version** 0.2.0

**Maintainer** Gary Chung <gchung05@gmail.com>

**Description** A set of core functions for handling medical device event data in the context of post-market surveillance, pharmacovigilance, signal detection and trending, and regulatory reporting. Primary inputs are data on events by device and data on exposures by device. Outputs include: standardized device-event and exposure datasets, defined analyses, and time series.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.0

**Depends** R (>= 2.10)

**Imports** stats, graphics, parsedate, lubridate

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Gary Chung [aut, cre]

**Repository** CRAN

**Date/Publication** 2018-09-26 17:00:08 UTC

## R topics documented:

define_analyses . . . . .	2
define_analyses_dataframe . . . . .	4
deviceevent . . . . .	4
exposure . . . . .	6
maude . . . . .	8
mds_ts . . . . .	9

plot.mds_ts . . . . .	9
sales . . . . .	10
summary.mds_das . . . . .	11
time_series . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

define_analyses	<i>Assess &amp; Save MD-PMS Analyses Definitions</i>
-----------------	--

---

### Description

Define analyses based on an MD-PMS device-event data frame and (optionally) an MD-PMS exposure data frame.

### Usage

```
define_analyses(deviceevents, device_level, event_level = NULL,
  exposure = NULL, date_level = "months", date_level_n = 1,
  covariates = "_none_", times_to_calc = NULL, prior = NULL)
```

### Arguments

deviceevents	A device-events object of class mds_de
device_level	String value indicating the source device variable name to analyze by. Example: If the deviceevents variable column is device_1 where the source variable name for device_1 is 'Device Code', specify device_level='Device Code'.
event_level	String value indicating the source event variable name to analyze by. Note that event_level is not matched to exposure. Default: NULL will not analyze by event. Example: If the deviceevents variable column is event_1 where the source variable name for event_1 is 'Event Code', specify event_level='Event Code'.
exposure	Optional exposure object of class mds_e. See details for how exposure analyses definitions are handled. Default: NULL will not consider inclusion of exposure.
date_level	String value for the primary date unit to analyze by. Can be either 'months' or 'days'. Default: 'months'
date_level_n	Numeric value indicating the number of date_levels to analyze by. Default: 1 Example: date_level='months' and date_level_n=3 indicates analysis on a quarterly level.

covariates	Character vector specifying names of covariates to also define analyses for. Acceptable names are covariate variable names from deviceevents. Analyses will be defined for each unique level of each covariate. "_none_" specifies no covariates, while "_all_" specifies all covariates from deviceevents. Default: "_none_" specifies no covariates. Example: "Country"
times_to_calc	Integer value indicating the number of date units counting backwards from the latest date to define analyses for. If prior is specified, times_to_calc will be ignored. Default: NULL will define analyses across all available time. Example 1: times_to_calc=12 with date_level="months" and date_level_n=1 defines analyses for the last year by month. Example 2: times_to_calc=8 with date_level="months" and date_level_n=3 defines analyses for the 2 years by quarter.
prior	Future placeholder, currently not used.

### Details

The analyses definitions will always include rollup levels for each of device\_level, event\_level (if specified), and covariates. These rollup analyses will be indicated by the keyword 'All', while the rollup of all covariates will be called 'Data'.

If exposure is specified, any available match\_levels will be used to calculate the appropriate timeframe for analyses. The exception are the special rollup analyses (see prior paragraph).

### Value

A list of defined analyses of class mds\_das. Each list item, indexed by a numeric key, defines a set of analyses for a unique combination of device, event, and covariate level. Each list item is of the class mds\_da. Attributes are as follows:

- date\_level** Defined value for date\_level
- date\_level\_n** Defined value for date\_level\_n
- device\_level** Defined value for device\_level
- event\_level** Defined value for event\_level
- times\_to\_calc** Defined value for times\_to\_calc
- prior\_used** Boolean for whether prior was specified.
- timestamp** System time when the analyses were defined.

### Examples

```
# Device-Events
de <- deviceevent(
  data_frame=maude,
  time="date_received",
  device_hierarchy=c("device_name", "device_class"),
  event_hierarchy=c("event_type", "medical_specialty_description"),
```

```

    key="report_number",
    covariates=c("region"),
    descriptors="_all_")
# Exposures
ex <- exposure(
  data_frame=sales,
  time="sales_month",
  device_hierarchy="device_name",
  match_levels="region",
  count="sales_volume")
# Defined Analyses - Simple example
da <- define_analyses(de, "device_name")
# Defined Analyses - Simple example with a quarterly analysis
da <- define_analyses(de, "device_name", date_level_n=3)
# Defined Analyses - Example with event type, exposures, and covariates
da <- define_analyses(de, "device_name", "event_type", ex, covariates="region")

```

---

define\_analyses\_dataframe

*Create Data Frame from Analyses Definitions*

---

### Description

Returns a data frame summarizing all defined analyses from the `mds_das` object.

### Usage

```
define_analyses_dataframe(inlist)
```

### Arguments

`inlist`            Object of class `mds_das`

### Value

A data frame with each row representing an analysis.

---

deviceevent

*MD-PMS Device Event Data Frame*

---

### Description

Converts a data frame into a MD-PMS Device Event data frame.

**Usage**

```
deviceevent(data_frame, time, device_hierarchy, event_hierarchy,
            key = NULL, covariates = NULL, descriptors = NULL,
            implant_days = NULL)
```

**Arguments**

- |                  |  |
|------------------|--|
| data_frame       | The input data frame requiring components specified in the remaining arguments.  |
| time             | Character name of date variable in data_frame corresponding to the event. Class must be Date, POSIXt, or character.<br>Example: "event_date"   |
| device_hierarchy | Vector of character variable names representing the device hierarchy in data_frame. Vector ordering is lowest level first, most general level last.<br>Example: c("Version", "Device", "ProductLine")  |
| event_hierarchy  | Vector of character variable names representing the event hierarchy in data_frame. Vector ordering is most specific event category first, most broad event category last.<br>Example: c("Event Code", "Event Group")   |
| key              | Character name of (uniquely identifying) primary key variable in data_frame. Class must be character or numeric.<br>Example: "key_ID"<br>Default: NULL will create a key variable.   |
| covariates       | Vector of character variable names representing the desired covariates to retain "_all_" includes all covariates, assumed to be remaining variables in data_frame not already specified in key, time, device_hierarchy, or event_hierarchy. It is recommended that covariates are categorical.<br>Example: c("Reporter", "City", "Country")<br>Default: NULL includes no covariates.   |
| descriptors      | Vector of character variable names representing additional descriptive variables that will not be used in any analyses but may be recalled or displayed later during individual device-event review. "_all_" includes all remaining variables in data_frame not already specified in key, time, device_hierarchy, event_hierarchy, or covariates. Typical descriptors are free text or high-dimensional categoricals.<br>Example: c("Description", "Unique Device Identifier")<br>Default: NULL includes no descriptors. |
| implant_days     | Character name of integer variable in data_frame representing the days in vivo of the device at the time of the event (time). More generally, this represents days of exposure of the device at the time of the event.<br>Example: "Implant Days". For example, a value of 45 indicates that the implant was in vivo for 45 days at the time of the event.<br>Default: NULL indicates this variable will not be used.  |

**Value**

A standardized MD-PMS data frame of class `mds_de`. Rows are deduplicated. Attributes are as follows:

**key** Original variable name for key

**time** Original variable name for time

**device\_hierarchy** Vector of original variable names for `device_hierarchy` with converted variable names correspondingly named.

**event\_hierarchy** Vector of original variable names for `event_hierarchy` with converted variable names correspondingly named.

**covariates** Vector of original variable names for `covariates` with converted variable names correspondingly named.

**descriptors** Vector of original variable names for `descriptors` with converted variable names correspondingly named.

**Examples**

```
# A barebones dataset
de <- deviceevent(maude, "date_received", "device_name", "event_type")
# With more variables and variable types
de <- deviceevent(
  data_frame=maude,
  time="date_received",
  device_hierarchy=c("device_name", "device_class"),
  event_hierarchy=c("event_type", "medical_specialty_description"),
  key="report_number",
  covariates=c("region"),
  descriptors="_all_")
```

---

exposure

*MD-PMS Exposure Data Frame*

---

**Description**

Converts a data frame into a MD-PMS Exposure data frame.

**Usage**

```
exposure(data_frame, time, device_hierarchy, event_hierarchy = NULL,
  key = NULL, match_levels = NULL, count = NULL)
```

**Arguments**

<code>data_frame</code>	The input data frame requiring components specified in the remaining arguments.
<code>time</code>	Character name of date variable in <code>data_frame</code> . Class must be Date, POSIXt, or character. Example: "event_date"
<code>device_hierarchy</code>	Vector of character variable names representing the device hierarchy in <code>data_frame</code> . Vector ordering is lowest level first, most general level last. Example: <code>c("Version", "Device", "ProductLine")</code>
<code>event_hierarchy</code>	Vector of character variable names representing the event hierarchy in <code>data_frame</code> . Vector ordering is most specific event category first, most broad event category last. Example: <code>c("Family", "Device", "ProductCode")</code> Default: NULL will not include any event hierarchy.
<code>key</code>	Character name of (uniquely identifying) primary key variable in <code>data_frame</code> . Class must be character or numeric. Example: "key_ID" Default: NULL will create a key variable.
<code>match_levels</code>	Vector of character variable names in <code>data_frame</code> representing additional grouping factors for exposure. Specified variables will be implicitly matched to equivalently named variables contained in the <code>mds_de</code> object class. Example: <code>c("Country", "Region")</code> Default: NULL will not include any additional grouping factors.
<code>count</code>	Character name of exposure count variable in <code>data_frame</code> . Class must be numeric. Example: "Units Sold" Default: NULL will assume each row represents one exposure.

**Value**

A standardized MD-PMS data frame of class `mds_e`. Rows are deduplicated. Attributes are as follows:

**key** Original variable name for key

**time** Original variable name for time

**device\_hierarchy** Vector of original variable names for `device_hierarchy` with converted variable names correspondingly named.

**event\_hierarchy** Vector of original variable names for `event_hierarchy` with converted variable names correspondingly named.

**match\_levels** Vector of variable names for grouping factors

**count** Original variable name for count

## Examples

```
# A barebones dataset
ex <- exposure(sales, "sales_month", "device_name")
# With more variables and variable types
ex <- exposure(
  data_frame=sales,
  time="sales_month",
  device_hierarchy="device_name",
  match_levels="region",
  count="sales_volume")
```

---

 maude

*Bone Cement MAUDE Events in 2017*


---

## Description

A dataset containing 535 events reported into the FDA MAUDE database on bone cement in 2017. Data were obtained via the openFDA API (<https://open.fda.gov>).

## Usage

```
maude
```

## Format

A data frame with 535 rows and 15 variables. Full variable descriptions may be found on the FDA Device Reference Guide (<https://open.fda.gov>). Note that `region` is a simulated variable not present in MAUDE. Descriptions as follows:

**report\_number** Identifying number for the adverse event report.

**event\_type** Outcomes associated with the adverse event.

**date\_received** Date the report was received by the FDA.

**product\_problem\_flag** Indicates whether or not a report was about the quality, performance or safety of a device.

**adverse\_event\_flag** Whether the report is about an incident where the use of the device is suspected to have resulted in an adverse outcome in a patient.

**report\_source\_code** Source of the adverse event report.

**lot\_number** The lot number found on the label or packaging material.

**model\_number** The exact model number found on the device label or accompanying packaging.

**manufacturer\_d\_name** Device manufacturer name.

**manufacturer\_d\_country** Device manufacturer country.

**brand\_name** The trade or proprietary name of the suspect medical device as used in product labeling or in the catalog.

**device\_name** This is the proprietary name, or trade name, of the cleared device.



**medical\_specialty\_description** Regulation Medical Specialty is assigned based on the regulation (e.g. 21 CFR Part 888 is Orthopedic Devices).

**device\_class** A risk based classification system for all medical devices ((Federal Food, Drug, and Cosmetic Act, section 513)

**region** A simulated, randomly assigned geographical region for package example purposes.

### Source

<https://open.fda.gov/data/maude/>

---

mds_ts	<i>Example List of mds_ts Time Series Objects</i>
--------	---

---

### Description

An example list of time series objects (class mds\_ts) generated using the mds package.

### Usage

```
mds_ts
```

### Format

A list of 3 elements each of class mds\_ts

### Source

See ?maude and ?sales for source device-event and exposure data. See ?mds::time\_series for how to generate mds\_ts time series.

---

plot.mds_ts	<i>Plot MD-PMS Time Series</i>
-------------	--------------------------------

---

### Description

Quickly visualizes an MD-PMS times series of class mds\_ts.

### Usage

```
## S3 method for class 'mds_ts'
plot(x, mode = "nA", xlab = "Time", ylab = "Count",
     main = NULL, ...)
```

**Arguments**

<code>x</code>	An object of class <code>mds_ts</code> .
<code>mode</code>	Series to plot. Valid values are: 'nA', 'nB', 'nC', 'nD', 'exposure', 'rate'. 'rate' is simply 'nA' / 'exposure'. See details for more. Default: 'nA'
<code>xlab</code>	x-axis label #' Default: 'Time'
<code>ylab</code>	y-axis label Default: 'Count'
<code>main</code>	Plot title Default: NULL infers the title from <code>x</code> and <code>mode</code> .
<code>...</code>	Further arguments to pass onto <code>plot()</code> generic.

**Details**

mode values defined as follows. Note: The following definitions use a device-event pair as a working example, however it may also be a covariate-device pair.

'nA' Counts of the device-event pair.

'nB' Counts of the device for all other events.

'nC' Counts of all other devices for the event.

'nD' Counts of all other devices for all other events.

'exposure' Counts of exposure for the device-event pair.

'rate' A crude rate, calculated as the device-event counts pair divided by the exposure counts.

---

sales

*Simulated Device Sales Data*

---

**Description**

A dataset containing simulated monthly sales by device and country for devices reported in the maude dataset. For package usage examples, this data serves as a proxy for exposures. Data were generated using a random normal distribution weighted by the number of reported events by device and country.

**Usage**

sales

**Format**

A data frame with 360 rows and 4 variables:

**device\_name** Name of the device mapped from the maude dataset.

**region** Geographical region mapped from the maude dataset.

**sales\_month** The month of sales.

**sales\_volume** The volume of sales.

**Source**

Random normal distribution using `rnorm()`.

---

summary.mds_das	<i>Summarize a Collection of MD-PMS Defined Analyses Prints basic counts and date ranges by various analysis factors as defined in the original <code>define_analyses()</code> call.</i>
-----------------	--

---

**Description**

Summarize a Collection of MD-PMS Defined Analyses Prints basic counts and date ranges by various analysis factors as defined in the original `define_analyses()` call.

**Usage**

```
## S3 method for class 'mds_das'
summary(object, ...)
```

**Arguments**

object	A MD-PMS Defined Analyses object of class <code>mds_das</code>
...	Additional arguments affecting the summary produced

**Value**

List of analyses counts and date ranges.

time\_series

*Generate Time Series from Defined Analysis or Analyses***Description**

Converts defined analysis (class `mds_da`) to a time series object or a list of defined analyses (class `mds_das` or `list`) to a list of time series objects.

**Usage**

```
time_series(analysis, ...)

## S3 method for class 'list'
time_series(analysis, ...)

## S3 method for class 'mds_das'
time_series(analysis, ...)

## S3 method for class 'mds_da'
time_series(analysis, deviceevents, exposure = NULL,
            use_hierarchy = T, ...)
```

**Arguments**

<code>analysis</code>	A defined analysis object of class <code>mds_da</code> , list of class <code>mds_das</code> , or a list of objects each of class <code>mds_da</code> .
<code>...</code>	Further arguments for future work.
<code>deviceevents</code>	A device-events object of class <code>mds_de</code> . Typically, this will be the same <code>mds_de</code> object used to generate analysis.
<code>exposure</code>	Optional exposure object of class <code>mds_e</code> . Typically, this will be the same <code>mds_e</code> object used to generate analysis, if an exposure was used. Default: <code>NULL</code> will not consider exposure data.
<code>use_hierarchy</code>	Logical value indicating whether device and event hierarchies should be used in counting contingency tables for disproportionality analysis. See details for more.

**Details**

When `use_hierarchy=T`, the B, C, and D cells of the 2x2 contingency table count at the next level up in the device and/or event hierarchies. For example, if the A cell is counting `device_1="Apple"` and "Apple" is the child of the parent `device_2="Fruits"`, the B cell counts all "Fruits" not equal to "Apple".

When `use_hierarchy=F`, the B, C, and D cells of the 2x2 contingency table simply count all devices/events/covariate levels not equal to the reference cell A.

**Value**

A standardized MD-PMS time series data frame of class `mds_ts`. The data frame contains, by defined date levels, the following:

**nA** Count of the device & event level of interest. If covariate analysis is indicated, this will be at the covariate & device level of interest.

**nB** Count of the device & non-event, or if covariate analysis, covariate & non-device. `nB` will be missing if this is an 'All' level analysis.

**nC** Count of the non-device & event, or if covariate analysis, non-covariate & device. `nC` will be missing if this is an 'All' level analysis.

**nD** Count of the non-device & non-event, or if covariate analysis, non-covariate & non-device. `nD` will be missing if this is an 'All' level analysis.

**ids** List of all keys from `deviceevents` constituting `nA`.

**exposure** Count of exposures applicable to `nA`. This counts at the device and covariate levels but not at the event level. If a matching device and/or covariate level is not found, then exposure will be `NA`. The exception is an 'All' level analysis, which counts exposures across all levels.

**ids\_exposure** List of all exposure keys from exposure applicable to `nA`.

Attributes are as follows:

**nA** Variable name(s) and level(s) (key-value) pair(s) for `nA`.

**nABCD** Variable names and levels (key-value) pairs for the entire 2x2 contingency table, if applicable to the analysis.

**nLabels** Plain language labels for `nA` and, if applicable, the rows and columns of the 2x2 contingency table.

**exposure** Boolean of whether exposure counts are present.

**dpa** Boolean of whether 2x2 contingency table counts are present (presumably for disproportionality analysis or 'DPA').

**Methods (by class)**

- `list`: Generate time series from a list
- `mds_das`: Generate time series from a list of defined analyses
- `mds_da`: Generate time series using defined analysis

**Examples**

```
de <- deviceevent(maude, "date_received", "device_name", "event_type")
ex <- exposure(sales, "sales_month", "device_name", count="sales_volume")
da <- define_analyses(de, "device_name", exposure=ex)
# Time series on one analysis
time_series(da, de, ex)
# Time series on multiple analyses
time_series(da[1:3], de, ex)
```

# Index

## \*Topic **datasets**

maude, [8](#)

mds\_ts, [9](#)

sales, [10](#)

define\_analyses, [2](#)

define\_analyses\_dataframe, [4](#)

deviceevent, [4](#)

exposure, [6](#)

maude, [8](#)

mds\_ts, [9](#)

plot.mds\_ts, [9](#)

sales, [10](#)

summary.mds\_das, [11](#)

time\_series, [12](#)