# Package 'mdsstat'

September 26, 2018

**Type** Package

**Title** Statistical Trending for Medical Devices Surveillance

**Version** 0.2.0

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

**Description** A collection of common statistical algorithms used in active surveillance of medical device events. Context includes post-market surveillance, pharmacovigilance, signal detection and trending, and regulatory reporting. Primary inputs are device-event time series. Outputs include trending results with the ability to run multiple algorithms at once. This package works well with the 'mds' package, but does not require it.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0)

**Imports** utils, stats, graphics, mds, lubridate

**Suggests** testthat, knitr, rmarkdown

**RoxygenNote** 6.1.0

**VignetteBuilder** knitr

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2018-09-26 17:00:02 UTC

## R topics documented:

---

`define_algos` *Set List of Algorithms to Run*

---

## Description

Define any number of algorithms with various parameter settings and save as a reusable set of instructions.

## Usage

```
define_algos(algos)
```

## Arguments

| | |
|---|---|
| `algos` | Required named list of `mdsstat` algorithms to run. Each named list element must be a single list of parameter values for the algorithm named. The list of parameters may be an empty list (indicating the default values) and must not contain the first parameter `df`. See details and examples for more. |

## Details

Valid list names (algorithm function names in `mdsstat`) are currently `shewhart`, `prr`, and `poisson_rare`. Each algorithm may be named multiple times (to allow running of multiple parameter settings). Do not specify the `df` parameter.

## Value

Validated list of instructions that may be used in the `run_algos` function.

## Examples

```
x <- list(prr=list(),
  shewhart=list(),
  shewhart=list(ts_event=c(Rate="rate"), we_rule=2L),
  poisson_rare=list(p_rate=0.3))
define_algos(x)
```

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 *Sample List of* mds_ts *Time Series*

### Description

A list of three mds_ts time series generated using the mds package function time_series(). Its underlying data were queried from the [FDA MAUDE API](https://open.fda.gov/api/). Furthermore, a simulated exposure dataset was generated to provide exposure data.

### Usage

```
mds_ts
```

### Format

A list with 3 elements, with each element being a data frame of class

### Source

maude and sales datasets, run through mds package functions.

## poisson_rare *Poisson for Rare Events*

### Description

Test on rare events using an exact test on the Poisson distribution rate parameter (stats::poisson.test()).

### Usage

```
poisson_rare(df, ...)

## S3 method for class 'mds_ts'
poisson_rare(df, ts_event = c(Count = "nA"),
  analysis_of = NA, ...)

## Default S3 method:
poisson_rare(df, analysis_of = NA,
  eval_period = NULL, zero_rate = 2/3, p_rate = 0.2, p_crit = 0.05,
  ...)
```

## Arguments

| | |
|---|---|
| df | Required input data frame of class mds_ts or, for generic usage, any data frame with the following columns:<br>**time** Unique times of class Date<br>**event** Either the event count or rate of class numeric |
| ... | Further arguments passed onto poisson_rare methods |
| ts_event | Required if df is of class mds_ts. Named string indicating the variable corresponding to the event count or rate. Rate must be calculated in a separate column in df as it is not calculated by default. The name of the string is an English description of what was analyzed.<br>Default: c("Count"="nA") corresponding to the event count column in mds_ts objects. Name is generated from mds_ts metadata.<br>Example: c("Rate of Bone Filler Events in Canada"="rate") |
| analysis_of | Optional string indicating the English description of what was analyzed. If specified, this will override the name of the ts_event string parameter.<br>Default: NA indicates no English description for plain df data frames, or ts_event English description for df data frames of class mds_ts.<br>Example: "Rate of bone cement leakage" |
| eval_period | Optional positive integer indicating the number of unique times counting in reverse chronological order to assess.<br>Default: NULL considers all times in df. |
| zero_rate | Required minimum proportion of events in df (constrained by eval_period) containing zeroes for this algorithm to run.<br>Default: 2/3 requires a minimum of 2/3 zeros in events in df. |
| p_rate | Hypothesized Poisson rate parameter null value at which the Poisson test is performed (null vs. greater). See details for more.<br>Default: 0.2 |
| p_crit | Critical p-value for the Poisson test..<br>Default: 0.05 |

## Details

p_rate default of 0.2 is a suggested null value for the Poisson rate parameter. However this value is highly advised to be set based on known priors and/or your specific application.

## Value

A named list of class mdsstat_test object, as follows:

**test_name** Name of the test run

**analysis_of** English description of what was analyzed

**status** Named boolean of whether the test was run. The name contains the run status.

**result** A standardized list of test run results: statistic for the test statistic, lcl and ucl for the 95 confidence bounds, p for the p-value, signal status, and signal_threshold.

**params** The test parameters

**data** The data on which the test was run

### Methods (by class)

- `mds_ts`: Poisson on mds_ts data
- `default`: Poisson on general data

### Examples

```
# Basic Example
data <- data.frame(time=c(1:8), event=c(rep(0, 6), stats::rpois(2, 4)))
a1 <- poisson_rare(data)
# Example using an mds_ts object
a2 <- poisson_rare(mds_ts[[1]])
# Example using a derived rate as the "event"
data <- mds_ts[[1]]
data$rate <- ifelse(is.na(data$nA), 0, data$nA) / data$exposure
a3 <- poisson_rare(data, c("Rate"="rate"))
```

---

## prr *Proportional Reporting Ratio*

---

### Description

Test on device-events using the proportional reporting ratio (PRR). From the family of disproportionality analyses (DPA) used to generate signals of disproportionate reporting (SDRs).

### Usage

```
prr(df, ...)

## S3 method for class 'mds_ts'
prr(df, ts_event = c(Count = "nA"), analysis_of = NA,
  ...)

## Default S3 method:
prr(df, analysis_of = NA, eval_period = 1L,
  null_ratio = 1, alpha = 0.05, ...)
```

### Arguments

`df` Required input data frame of class `mds_ts` or, for generic usage, any data frame with the following columns:

**time** Unique times of class `Date`

**nA** Cell A count (class `numeric`) of the 2x2 table: device/event of interest.

**nB** Cell B count (class `numeric`) of the 2x2 table: device/non-event of interest.

**nC** Cell C count (class `numeric`) of the 2x2 table: non-device/event of interest.

**nD** Cell D count (class `numeric`) of the 2x2 table: non-device/non-event of interest.

| | |
|---|---|
| ... | Further arguments passed onto prr methods |
| ts_event | Required if df is of class mds_ts. Named string indicating the variable corresponding to the event count (cell A in the 2x2 contingency table). In most cases, the default is the appropriate setting. See details for alternative options.<br>Default: c("Count"="nA") corresponding to the event count column in mds_ts objects. Name is generated from mds_ts metadata. |
| analysis_of | Optional string indicating the English description of what was analyzed. If specified, this will override the name of the ts_event string parameter.<br>Default: NA indicates no English description for plain df data frames, or ts_event English description for df data frames of class mds_ts.<br>Example: "Count of bone cement leakages" |
| eval_period | Required positive integer indicating the number of unique times counting in reverse chronological order to sum over to create the 2x2 contingency table.<br>Default: 1L considers only the most recent time in df.<br>Example: 12L sums over the last 12 time periods to create the 2x2 contingency table. |
| null_ratio | Numeric PRR value representing the null hypothesis, used with alpha to establish the signal status and the p-value.<br>Default: 1 indicates a null hypothesis of PRR=1 and tests if the actual PRR is greater than 1. |
| alpha | Numeric value representing the statistical alpha used to establish the signal status.<br>Default: 0.05 corresponds to the standard alpha value of 5%. |

## Details

For parameter ts_event, in the uncommon case where the device-event count (Cell A) variable is not "nA", the name of the variable may be specified here. Note that the remaining 3 cells of the 2x2 contingency table (Cells B, C, D) must be the variables "nB", "nC", and "nD" respectively in df. A named character vector may be used where the name is the English description of what was analyzed. Note that if the parameter analysis_of is specified, it will override this name. Example: ts_event=c("Count of Bone Cement Leakages"="event_count")

## Value

A named list of class mdsstat_test object, as follows:

**test_name** Name of the test run

**analysis_of** English description of what was analyzed

**status** Named boolean of whether the test was run. The name contains the run status.

**result** A standardized list of test run results: statistic for the test statistic, lcl and ucl for the set confidence bounds, p for the p-value, signal status, and signal_threshold.

**params** The test parameters

**data** The data on which the test was run

### Methods (by class)

- mds_ts: PRR on mds_ts data
- default: PRR on general data

### References

Evans, S. J. W., Waller, P. C., & Davis, S. (2001). Use of proportional reporting ratios (PRRs) for signal generation from spontaneous adverse drug reaction reports. Pharmacoepidemiology and Drug Safety, 10(6), 483-486. https://doi.org/10.1002/pds.677

### Examples

```
# Basic Example
data <- data.frame(time=c(1:25),
                   nA=as.integer(stats::rnorm(25, 25, 5)),
                   nB=as.integer(stats::rnorm(25, 50, 5)),
                   nC=as.integer(stats::rnorm(25, 100, 25)),
                   nD=as.integer(stats::rnorm(25, 200, 25)))
a1 <- prr(data)
# Example using an mds_ts object
a2 <- prr(mds_ts[[3]])
```

---

## run_algos *Run Multiple Algorithms*

---

### Description

Run a multiple number of mdsstat algorithms on a single input dataset.

### Usage

```
run_algos(data, algos, dataframe = T, non_dpa = "skip", ...)

## S3 method for class 'list'
run_algos(data, algos, dataframe = T, non_dpa = "skip",
  ...)

## Default S3 method:
run_algos(data, algos, dataframe = T,
  non_dpa = "skip", ...)
```

### Arguments

| | |
|---|---|
| data | Required input dataset. Note that the dataset must satisfy the dataset requirements for each algorithm specified (parameter df). An mds times series object (class mds_ts) is a natural fit. |

| | |
|---|---|
| algos | Input list of algorithms to run. Must be a list generated by define_algos. |
| dataframe | Logical on whether to output results as a mdsstat_tests data frame. If FALSE, will output as a list of mdsstat_test lists.<br>Default: TRUE |
| non_dpa | What to do when input data is not prepared for disproportionality analysis (DPA) data. Three values are accepted: "skip", "warn", and "stop". "skip" skips the DPA test without warnings or errors. "warn" outputs a warning and then skips the DPA test. "stop" stops the function call.<br>Default: "skip" |
| ... | Further arguments for future work. |

## Value

A mdsstat_tests data frame or list of mdsstat_test lists with the results of the algorithm runs.

## Methods (by class)

- list: Run algorithms on a list of time series
- default: Run algorithms on a single time series

## Examples

```
data <- mds_ts[[3]]
data$rate <- data$nA / data$exposure
x <- list(prr=list(),
  shewhart=list(),
  shewhart=list(ts_event=c(Rate="rate"), we_rule=2L),
  poisson_rare=list(p_rate=0.3))
algos <- define_algos(x)
run_algos(data, algos)
run_algos(data, algos, FALSE)
```

---

# 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().

---

shewhart *Shewhart x-bar Control Chart*

---

## Description

Test on device-events using the Shewhart x-bar control chart. Includes the first 4 Western Electric rules common to statistical process control.

## Usage

```
shewhart(df, ...)

## S3 method for class 'mds_ts'
shewhart(df, ts_event = c(Count = "nA"),
  analysis_of = NA, ...)

## Default S3 method:
shewhart(df, analysis_of = NA, eval_period = NULL,
  zero_rate = 1/3, we_rule = 1L, ...)
```

## Arguments

| | |
|---|---|
| df | Required input data frame of class mds_ts or, for generic usage, any data frame with the following columns:<br>**time** Unique times of class Date<br>**event** Either the event count or rate of class numeric |
| ... | Further arguments passed onto shewhart methods |
| ts_event | Required if df is of class mds_ts. Named string indicating the variable corresponding to the event count or rate. Rate must be calculated in a separate column in df as it is not calculated by default. The name of the string is an English description of what was analyzed.<br>Default: c("Count"="nA") corresponding to the event count column in mds_ts objects. Name is generated from mds_ts metadata.<br>Example: c("Rate of Bone Filler Events in Canada"="rate") |

| | |
|---|---|
| analysis_of | Optional string indicating the English description of what was analyzed. If specified, this will override the name of the ts_event string parameter.<br>Default: NA indicates no English description for plain df data frames, or ts_event English description for df data frames of class mds_ts.<br>Example: "Rate of bone cement leakage" |
| eval_period | Optional positive integer indicating the number of unique times counting in reverse chronological order to assess. This will be used to establish the process mean and moving range.<br>Default: NULL considers all times in df. |
| zero_rate | Required maximum proportion of events in df (constrained by eval_period) containing zeroes for this algorithm to run. Because Shewhart does not perform well on rare events, a value >0 is recommended.<br>Default: 1/3 requires no more than 1/3 zeros in events in df in order to run. |
| we_rule | Required integer from 1 to 4 representing the Western Electric rule to use. See details for descriptions.<br>Default: 1 represents the first Western Electric rule of one point over the 3-sigma limit. |

## Details

Function shewhart() is a standard implementation of the x-bar Control Chart test from the family of statistical process control tests originally proposed by Walter Shewhart.

we_rule has four possible values: 1 is one point over the 3-sigma limit. 2 is two out of three consecutive points over the 2-sigma limit. 3 is four of five consecutive points over the 1-sigma limit. 4 is nine consecutive points over the process mean.

## Value

A named list of class mdsstat_test object, as follows:

**test_name** Name of the test run

**analysis_of** English description of what was analyzed

**status** Named boolean of whether the test was run. The name contains the run status.

**result** A standardized list of test run results: statistic for the test statistic, lcl and ucl for the 95 confidence bounds, p for the p-value, signal status, and signal_threshold.

**params** The test parameters

**data** The data on which the test was run

## Methods (by class)

- mds_ts: Shewhart on mds_ts data
- default: Shewhart on general data

## References

Montgomery, Douglas C. Introduction to Statistical Quality Control by Douglas C. Montgomery, 5th Edition: Study Guide. Cram101, 2013.

## Examples

```
# Basic Example
data <- data.frame(time=c(1:25), event=as.integer(stats::rnorm(25, 100, 25)))
a1 <- shewhart(data)
# Example using an mds_ts object
a2 <- shewhart(mds_ts[[3]])
# Example using a derived rate as the "event"
data <- mds_ts[[3]]
data$rate <- ifelse(is.na(data$nA), 0, data$nA) / data$exposure
a3 <- shewhart(data, c(Rate="rate"))
```

---

# test_as_row *Coerce mdsstat Test to 1-Row Data Frame*

---

## Description

Coerce an mdsstat test (class `mdsstat_test`) to a 1-row data frame.

## Usage

```
test_as_row(df)
```

## Arguments

| | |
|---|---|
| `df` | Required input object of class `mdsstat_test` |

## Value

1-row data frame (class `mdsstat_df`) summarizing the test.

## Examples

```
test_as_row(prr(mds_ts[[3]]))
```

# Index