Package 'pvLRT'

January 21, 2023

Title Likelihood Ratio Test-Based Approaches to Pharmacovigilance

Version 0.5

Maintainer Saptarshi Chakraborty <chakra.saptarshi@gmail.com>

Date 2023-01-18

Description A suite of likelihood ratio test based methods to use in pharmacovigilance. Contains various testing and post-processing functions.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports stats, methods, utils, magrittr (>= 2.0.0), progress, data.table, bit64, glue, RColorBrewer, ggplot2, ggfittext

Depends R (>= 3.6.0)

Suggests furrr, plotly, patchwork

BugReports https://github.com/c7rishi/pvLRT/issues

NeedsCompilation no

Author Saptarshi Chakraborty [aut, cre],
Marianthi Markatou [aut],
Anran Liu [ctb]

Repository CRAN

Date/Publication 2023-01-21 15:50:02 UTC

R topics documented:

`pvLRT-package` *pvLRT: An R package implementing various Likelihood Ratio Test-based approaches to pharmacovigilance*

Description

pvLRT is an R package that implements a suite of likelihood ratio test based methods to use in pharmacovigilance. It can handle adverse events data on several simultaneous drugs, with possibly zero inflated report counts. Several testing and post-processing functions are implemented.

`all.equal.pvlrt` *Test if two pvlrt objects are (Nearly) Equal*

Description

Test if two pvlrt objects are (Nearly) Equal

Usage

```
## S3 method for class 'pvlrt'
all.equal(target, current, ...)
```

Arguments

`target`	First pvlrt object (output of pvlrt).
`current`	Second pvlrt object.
`...`	Arguments passed to all.equal.default.

Details

Compares all values and attributes of target and current `pvlrt` objects except running times. See all.equal.default for details on the generic function.

See Also

all.equal.default

`as.matrix.pvlrt` *Casting a* `pvlrt` *object as a matrix of log LR statistics*

Description

Casting a `pvlrt` object as a matrix of log LR statistics

Usage

```
## S3 method for class 'pvlrt'
as.matrix(x, ...)
```

Arguments

`x`	a `pvlrt` object; an output of function `pvlrt`().
`...`	other input parameters. Currently unused.

Value

Returns a matrix with the same dimensions as the input contingency table in the original `pvlrt` call, with each cell providing the corresponding value of the observed log-likelihood ratio test statistic.

See Also

pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

test1 <- pvlrt(statin46, nsim = 500)
as.matrix(test1)
```

`convert_raw_to_contin_table` *Convert raw AE-Drug incidence data into a contingency table*

Description

Convert raw AE-Drug incidence data into a contingency table

Usage

```
convert_raw_to_contin_table(
  rawdata,
  Drug_col_name = "DRUG",
  AE_col_name = "AE",
  id_col_name = "CASEID",
  count_col_name = "COUNT",
  aggregated = FALSE,
  create_other_Drug_col = FALSE,
  other_Drug_excludes = NULL,
  other_Drug_colname = "Other_Drug",
  create_other_AE_row = FALSE,
  other_AE_excludes = NULL,
  other_AE_rowname = "Other_AE",
  ...
)
```

Arguments

`rawdata`	a `data.frame` or an object that can be converted to a `data.frame`. Must contain 3 columns (i) DRUG: the drug names/labels, (ii) AE: the AE names, and either (iii) CASEID: case ids corresponding to each combination of AE and DRUG, (if `aggregated` is `FALSE`), or (iii') COUNT: the total number of incidences of each AE and DRUG combination (if `aggregated` is `TRUE`). If these columns are named differently in `rawdata`, supply the appropriate column names through `Drug_col_name`, `AE_col_name`, `id_col_name` and `count_col_name`.
`Drug_col_name, AE_col_name, id_col_name, count_col_name`	Drug, AE, case id and count column names in `rawdata`. Defaults to `DRUG`, `AE`, `CASEID` and `COUNT`.

`aggregated` logical. Has the incidences been already aggregated/summarized into counts in `rawdata`? If `TRUE` then then `COUNT` column is used to produce the output contingency table. If `FALSE` (default) incidences are first aggregated into counts before converting to contingency tables.

`create_other_Drug_col` logical. Add a column in the contingency table for "Other Drugs"? This column plays the role of a "baseline" group of drugs that typically do not indicate an adverse event association with the signal of interest. Care should be taken while determining which drugs to include in this group; See Ding et al (2020) for guidance.

`other_Drug_excludes` character vector cataloging Drugs that are NOT to be included in the column for Other Drugs. If NULL (default) then then no Drugs are included in Other Drugs (i.e., `other_Drug_excludes` contains all Drugs in the raw data). Ignored if `create_other_Drug_col = FALSE`.

`other_Drug_colname` character. Row name for the "Other Drug" column created. Ignored if `create_other_Drug_col = FALSE`.

`create_other_AE_row` logical. Add a row in the contingency table for "Other AEs"? This can aid computation in situations where there are certain AEs of primary interest. See `other_AE_excludes` for details on how to specify the "Other AE" row.

`other_AE_excludes` character vector cataloging AEs that are NOT to be included in the row for Other AEs. If NULL (default) then then no AEs are included in Other AEs (i.e., `other_AE_excludes` contains all AEs in the raw data). Ignored if `create_other_AE_row = FALSE`.

`other_AE_rowname` character. Row name for the "Other AE" row created. Defaults to "Other AE". Ignored if `create_other_AE_row = FALSE`.

`...` unused.

Details

This is a convenience function that creates a contingency table cataloging counts of AE-Drug incidences from a raw Drug/AE incidence data frame. It accepts both raw incidence data (each row is one incidence of a Drug-AE combination, indexed by case ids) and summarized count data (each row catalogs the total counts of incidences of each Drug-AE pair). The output is a matrix (contingency table) enumerating total count of cases for each pair of AE (along the rows) and drug (along the columns) with appropriately specified row and column names, and can be passed to a pvlrt() call. See the examples for more details.

The output can be fed into pvlrt or its wrappers as `contin_table`

References

Ding, Y., Markatou, M. and Ball, R., 2020. An evaluation of statistical approaches to postmarketing surveillance. Statistics in Medicine, 39(7), pp.845-874.

Chakraborty, S., Liu, A., Ball, R. and Markatou, M., 2022. On the use of the likelihood ratio test methodology in pharmacovigilance. Statistics in Medicine, 41(27), pp.5395-5420.

Examples

```
# convert to contingency table form incidence (non-aggregated) raw data
# AE subset = AEs in statin46
# Durg subset = union of statin46 and gbca drugs
tab1 <- convert_raw_to_contin_table(
  rawdata = faers22q3raw,
  Drug_col_name = "DRUG",
  AE_col_name = "AE",
  id_col_name = "CASEID",
  aggregated = FALSE,
  other_AE_excludes = rownames(statin46),
  other_Drug_excludes = union(colnames(gbca), colnames(statin)),
  create_other_Drug_col = TRUE,
  create_other_AE_row = FALSE
)

# convert to contingency table AFTER aggregating and counting
# the total number of incidences of each (AE, Drug) pair
## Same AE and Drug subsets as before
## aggregation (counting) done using data.table dt[i, j, by] syntax
## uses magrittr %>% pipe
tab2 <- data.table::as.data.table(
  faers22q3raw
)[
  ,
  .(COUNT = length(unique(CASEID))),
  by = .(DRUG, AE)
] %>%
  convert_raw_to_contin_table(
    Drug_col_name = "DRUG",
    AE_col_name = "AE",
    count_col_name = "COUNT",
    aggregated = TRUE,
    other_AE_excludes = rownames(statin46),
    other_Drug_excludes = union(colnames(gbca), colnames(statin)),
    create_other_Drug_col = TRUE,
    create_other_AE_row = FALSE
  )

all.equal(tab1, tab2)

# use the contingency table produced above in pvlrt()
## 500 bootstrap iterations (nsim) in the example below
## is for quick demonstration only --
## we recommended setting nsim to 10000 (default) or bigger
test1 <- pvlrt(tab1, nsim = 500)
```

`extract_AE_names` *Extracting and setting AE and Drug names from a pvlrt object*

Description

Extracting and setting AE and Drug names from a pvlrt object

Usage

```
extract_AE_names(object)

extract_Drug_names(object)

set_AE_names(object, old, new)

set_Drug_names(object, old, new)
```

Arguments

`object`	a `pvlrt` object, which is the output of the function pvlrt or one of its wrappers such as lrt_zi_poisson, lrt_poisson and `lrt_vanilla_poisson`.
`old`	character vector containing the old names
`new`	character vector containing the new names

Value

- `extract_AE_names` returns a character vector of the names of the AEs in the input `pvlrt` object
- `extract_Drug_names` returns a character vector of the names of the Drugs in the input `pvlrt` object
- `set_AE_names` returns a new `pvlrt` object with updated AE names as specified through the arguments `old` and `new`.
- `set_Drug_names` returns a new `pvlrt` object with updated Drug names as specified through the arguments `old` and `new`.

Note

Because a `pvlrt` object is simply a reclassified matrix, the AE (rows) and Drug (columns) names can also be extracted/modified through rownames and colnames respectively.

See Also

pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

test1 <- pvlrt(statin46, test_zi = TRUE, nsim = 500)
extract_AE_names(test1)
extract_Drug_names(test1)

set_AE_names(test1, old = "Rhabdomyolysis", new = "Rhabdo")
set_Drug_names(test1, old = "Other", new = "Other-Drugs")

## can be chained with pipes `%>%`:
test2 <- test1 %>%
  set_AE_names(old = "Rhabdomyolysis", new = "Rhabdo") %>%
  set_Drug_names(old = "Other", new = "Other-Drugs")

# see the summary for changed labels
summary(test2)
```

`extract_lrstat_matrix` *Extract various summary measures from a pvlrt object*

Description

Extract various summary measures from a pvlrt object

Usage

```
extract_lrstat_matrix(object, ...)

extract_p_value_matrix(object, ...)

extract_zi_probability(object, ...)

extract_n_matrix(object, ...)

extract_significant_pairs(object, significance_level = 0.05, ...)

extract_run_time(object, ...)
```

Arguments

`object`	a `pvlrt` object, which is the output of the function pvlrt or one of its wrappers such as lrt_zi_poisson, lrt_poisson and `lrt_vanilla_poisson`.
`...`	other input parameters. Currently unused.

`significance_level`
: numeric. Level of significance.

Value

- `extract_lrstat_matrix` returns the matrix of the computed log-likelihood ratio test statistics for signals. This produces a result identical to applying `as.matrix`.
- `extract_p_value_matrix` returns the matrix of computed p-values associated with the likelihood ratio tests.
- `extract_zi_probability` returns a vector of (estimated) zero-inflation probabilities.
- `extract_n_matrix` returns the original contingency table (matrix) used.
- `extract_significant_pairs` returns a data.table listing the AE/drug pairs determined to be significant at the provided significance level. This is essentially a subset of the data.table obtained through summary.pvlrt() that satisfies the provided significance threshold.
- `extract_run_time` returns a difftime object measuring the total CPU time needed to run the original pvlrt call.

See Also

pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

test1 <- pvlrt(statin46, test_zi = TRUE, nsim = 500)
extract_lrstat_matrix(test1)
extract_p_value_matrix(test1)
extract_zi_probability(test1)
extract_n_matrix(test1)
extract_significant_pairs(test1)
```

---

faers22q3raw *FDA FAERS dataset for 2022 Q3*

Description

The raw FDA FAERS dataset for 2022 Q3; downloaded from FDA's website and then subsetted to inlcude incidences with `ROLE_COD == "PS"`.

Usage

```
faers22q3raw
```

Format

An object of class `data.table` (inherits from `data.frame`) with 496312 rows and 3 columns.

Details

obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4

This is a raw incidence data stored as a data.table with each row corresponding to a specific incidence index by case ids. It contains the following columns:

- `CASEID` - case ids for each incidence
- `DRUG` - names of the drugs
- `AE` - names of the adverse events

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

---

gbca *FDA GBCA dataset with all observed 1707 adverse events*

Description

A Drug-Adverse event (AE) count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4

Usage

```
gbca
```

Format

An object of class `matrix` (inherits from `array`) with 1707 rows and 10 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 1707 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that drug/AE pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset contains 6 Gadolinium-Based Contrast Agents (GBCAs) as drugs:

gadobenate, gadobutrol, gadodiamide, gadofosveset, gadopentetate, gadoterate, gadoteridol, gadoversetamide, gadoxetate

Corresponding to all 1707 observed adverse events (AEs) as curated in FAERS database.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

heatmap_pvlrt *Heatmap, barplot and bubbleplot displaying likelihood ratio test results in a* pvlrt *object*

Description

Heatmap, barplot and bubbleplot displaying likelihood ratio test results in a pvlrt object

Usage

```
heatmap_pvlrt(
  object,
  AE = NULL,
  Drug = NULL,
  grep = FALSE,
  fill_measure = "p_value",
  show_n = FALSE,
  show_lrstat = FALSE,
  show_p_value = FALSE,
  p_value_lower = 0,
  p_value_upper = 1,
  lrstat_lower = 0,
  lrstat_upper = Inf,
  n_lower = 0,
  n_upper = Inf,
  arrange_alphabetical = FALSE,
  remove_outside = FALSE,
  digits = 2,
  border_color = "black",
  fill_gradient_range = c("darkred", "white"),
  ...
)

## S3 method for class 'pvlrt'
barplot(
  height,
  AE = NULL,
  Drug = NULL,
  grep = FALSE,
  x_axis_measure = "lrstat",
  fill_measure = "p_value",
  show_n = FALSE,
  arrange_alphabetical = FALSE,
  show_p_value = FALSE,
  show_lrstat = FALSE,
  p_value_lower = 0,
  p_value_upper = 1,
```

```
  lrstat_lower = 0,
  lrstat_upper = Inf,
  n_lower = 0,
  n_upper = Inf,
  remove_outside = FALSE,
  digits = 2,
  Drug_nrow = 1,
  border_color = "black",
  x_axis_logscale = TRUE,
  fill_gradient_range = c("darkred", "white"),
  ...
)

bubbleplot_pvlrt(
  object,
  AE = NULL,
  Drug = NULL,
  grep = FALSE,
  x_axis_measure = "lrstat",
  fill_measure = "p_value",
  size_measure = "n",
  show_n = FALSE,
  arrange_alphabetical = FALSE,
  show_p_value = FALSE,
  show_lrstat = FALSE,
  p_value_lower = 0,
  p_value_upper = 1,
  lrstat_lower = 0,
  lrstat_upper = Inf,
  n_lower = 0,
  n_upper = Inf,
  remove_outside = FALSE,
  digits = 2,
  Drug_nrow = 1,
  border_color = "black",
  x_axis_logscale = TRUE,
  size_logscale = TRUE,
  fill_gradient_range = c("darkred", "white"),
  ...
)
```

Arguments

`object, height`	pvlrt object; output of `pvlrt()`
`AE`	input parameter determining which adverse events to show in the plot. This can be a numeric scalar specifying the number of *top* (in terms of computed LRT values) adverse events to show. Alternatively, it can be a character vector, specifying the exact adverse events to show. It can also be a vector of patterns

to match (ignores cases) against the full names of all available adverse events, provided grep is set to TRUE. Defaults to adverse events corresponding to the top M pairs where M = max(number of possible pairs, 50). Set AE = Inf to force display of all adverse events.

Drug — input parameter determining which drugs to show in the plot. This can be a numeric scalar specifying the number of *top* (in terms of computed LRT values) drugs to show. Alternatively, it can be a character vector, specifying the exact drugs to show. It can also be a vector of patterns to match (ignores cases) against the full names of all available drugs, provided grep is set to TRUE. Defaults to drugs corresponding to the top M pairs where M = max(number of possible pairs, 50). Set Drug = Inf to force display all drugs.

grep — logical. Match patterns against the supplied AE or Drug names? Ignores if neither AE nor Drug is a character vector.

fill_measure — Measure to govern the filling color in each cell (in heatmap) or bar (in barplot) or circle/bubble (in bubbleplot) for each drug/AE combination. Defaults to "p_value". Available choices are: "p.value", "lrstat", and "n".

show_n — logical. show the sample size as inscribed text on each cell?

show_lrstat — logical. show the computed LRT statistic (on log-scale) inscribed text on each cell?

show_p_value — logical. show the computed p-value as inscribed text on each cell?

p_value_lower, p_value_upper — lower and upper limits on the computed p-values to display on the plot.

lrstat_lower, lrstat_upper — lower and upper limits on the computed LRT values to display on the plot.

n_lower, n_upper — lower and upper limits on the computed sample sizes to display on the plot.

arrange_alphabetical — logical. should the rows (AEs) and columns (Drugs) be arranged in alphabetical orders? Defaults to FALSE, in which case the orderings of the computed LRT values are used.

remove_outside — logical. Should the values for pairs with p-value, LRT statistics or sample sizes falling outside of the provided ranges through p_value_lower, p_value_upper etc., be replaced with NA? Defaults to FALSE. Setting this to TRUE may help distinguish drugs or AEs which has some pairs falling within and some pairs falling outside of the provided ranges better.

digits — numeric. Number of decimal places to show on the inscribed texts on the plot.

border_color — character string. Specifies the border color of cells/bars.

fill_gradient_range — character vector. Specifies the range of gradient colors used for fill_measure. Passed into the colours argument of scale_fill_gradientn from ggplot2.

... — Other arguments. Currently ignored

x_axis_measure — measure to show on the x-axis of the (horizontal) bar plots. Defaults to "lrstat" available choices are "lrstat", "p_value" and "n".

Drug_nrow — Number of rows in the panels for Drugs for the barplots.

`x_axis_logscale`
: logical. Should the x axis measure in the bar plot or the bubble plot be log transformed (more precisely, "log1p" transformed with the function f(x) = log(1 + x))? Defaults to TRUE.

`size_measure`
: measure to govern sizes of the circles in the bubble plot. Defaults to "n". Available choices are "lrstat", "p_value" and "n".

`size_logscale`
: logical. Should the circle size measure in the the bubble plot be log transformed (more precisely, "log1p" transformed with the function f(x) = log(1 + x)). Defaults to TRUE.

Value

A ggplot object.

See Also

pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger
test1 <- pvlrt(statin46, nsim = 500)
bubbleplot_pvlrt(test1)
heatmap_pvlrt(test1)
barplot(test1)
```

---

logLik.pvlrt *Overall Log-likelihood for a pvlrt object*

Description

Overall Log-likelihood for a pvlrt object

Usage

```
## S3 method for class 'pvlrt'
logLik(object, type = "full-zip", ...)
```

Arguments

`object`
: a `pvlrt` object, which is the output of the function pvlrt or one of its wrappers such as lrt_zi_poisson, lrt_poisson and `lrt_vanilla_poisson`.

`type`
: Type of model and hypothesis combination. Available choices are "full-poisson", "null-poisson", "full-zip" (default), and "null-zip". See details.

`...`
: other input parameters. Currently unused.

Details

The function extracts the overall log-likelihood and degrees of freedom (the number of estimated parameters) from a `pvlrt` object. There are four possible choices of distribution and hypothesis combinations, supplied through the argument `type`, with the default being `type = "full-zip"`. In a "full" model the signal parameters lambdas are estimated for all cells in the contingency table from the data (subject to the condition lambda >= 1), whereas under a "null" model each lambda is fixed at 1 for each cell. In a "zip" model (type = "full-zip" and type = "null-zip") the log-likelihood under a zero-inflated Poisson model with estimated or supplied zero inflation parameters (through `zi_prob` in pvlrt) is returned. The degrees of freedom reflects whether the zero-inflation parameters are estimated. Note that if an ordinary Poisson LRT is run either by setting `zi_prob = 0` in pvlrt or equivalently through lrt_poisson then "full-zip" and "null-zip" refer to zero-inflated poisson models with presepecified zero-inflation probabilities equal to 0. Thus, in such cases the results with type = "full-zip" and type = "null-zip" are identical to type = "full-poisson" and type = "null-poisson" respectively. See examples.

Value

An object of class logLik. See Details.

Note

The overall log likelihood must be computed during the original pvlrt run. This is ensured by setting `return_overall_loglik = TRUE`, and `parametrization = "lambda"` (or `parametrization = "rrr"`) while running pvlrt().

See Also

pvlrt; AIC

Examples

```
# 500 bootstrap iterations (nsim) in each example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

set.seed(100)
# estimates zero inflation probabilities
test1 <- pvlrt(statin46, nsim = 500)
logLik(test1)
AIC(test1)
BIC(test1)

# compare with and without zero inflation
BIC(logLik(test1, type = "full-zip"))
BIC(logLik(test1, type = "full-poisson"))

# ordinary poisson model
## equivalent to pvlrt(statin46, zi_prob = 0, nsim = 500)
test2 <- lrt_poisson(statin46, nsim = 500)
```

```
all.equal(logLik(test2, "full-zip"), logLik(test2, "full-poisson"))
```

`lovastatin` *FDA lovastatin dataset*

Description

A Drug-Adverse event (AE) count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4

Usage

```
lovastatin
```

Format

An object of class `matrix` (inherits from `array`) with 47 rows and 3 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 46 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that drug/AE pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset contains 1 column for the lovastatin drug, and one column for all other drugs combined.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

---

`lrt_poisson` *Likelihood Ratio Test under the (vanilla, non-zero-inflated) Poisson model*

Description

Likelihood Ratio Test under the (vanilla, non-zero-inflated) Poisson model

Usage

```
lrt_poisson(contin_table, nsim = 10000, parametrization = "rrr", ...)

lrt_vanilla_poisson(contin_table, nsim = 10000, parametrization = "rrr", ...)
```

Arguments

contin_table	IxJ contingency table showing pairwise counts of adverse events for I AE (along the rows) and J Drugs (along the columns)
nsim	Number of simulated null contingency table to use for computing the p-value of the test
parametrization	Type of parametrization to use in the LR test. Available choices are "rrr", "lambda", "rr", and "p-q". The relative reporting ratio (default) parametrization of the test is used when when `parametrization %in% c("rrr", "lambda")`, and the reporting rate parametrization is used otherwise. NOTE: zero inflation can be handled only for the relative reporting ratio parametrization.
...	additional arguments. Currently unused.

Value

Returns a `pvlrt` object. See pvlrt for more details.

Note

`lrt_poisson()` and `lrt_vanilla_poisson()` are both wrappers for `pvlrt()` with `omega_vec = rep(0, ncol(contin_table))`

See Also

pvlrt

Examples

```
data("statin46")

# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

# no grouping -- each drug forms its own class
test1 <- lrt_poisson(lovastatin, nsim = 500)
```

---

lrt_zi_poisson *Pseudo Likelihood Ratio Test under the zero-inflated Poisson model with relative reporting rate parametrization*

Description

Pseudo Likelihood Ratio Test under the zero-inflated Poisson model with relative reporting rate parametrization

Usage

```
lrt_zi_poisson(contin_table, nsim = 10000, ...)
```

Arguments

contin_table	IxJ contingency table showing pairwise counts of adverse events for I AE (along the rows) and J Drugs (along the columns)
nsim	Number of simulated null contingency table to use for computing the p-value of the test
...	additional arguments passed to pvlrt

Value

Returns a pvlrt object. See pvlrt for more details.

Note

lrt_zi_poisson() is a wrapper for pvlrt() with parametrization = "rrr".

See Also

pvlrt

Examples

```
data("statin46")

# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger
test1 <- lrt_zi_poisson(statin46, nsim = 500)
test1
```

---

plot.pvlrt *Plotting method for a pvlrt object*

Description

Plotting method for a pvlrt object

Usage

```
## S3 method for class 'pvlrt'
plot(x, type = "bubbleplot", ...)
```

Arguments

x	a pvlrt object; an output of function pvlrt().
type	character string determining the type of plot to show. Available choices are "bubbleplot" which calls bubbleplot_pvlrt, "heatmap" which calls heatmap_pvlrt, and "barplot" which calls barplot.pvlrt, with the additional arguments supplied in ...
...	additional arguments passed to heatmap_pvlrt or barplot.pvlrt depending on type.

Value

A ggplot object.

See Also

pvlrt; bubbleplot_pvlrt; heatmap_pvlrt; barplot.pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

test1 <- pvlrt(statin46, nsim = 500)
plot(test1, type = "bubbleplot")
plot(test1, type = "barplot")
plot(test1, type = "heatmap")
```

---

print.pvlrt *Print method for pvlrt objects*

Description

Print method for pvlrt objects

Usage

```
## S3 method for class 'pvlrt'
print(
  x,
  significance_level = 0.05,
  topn = 12,
  digits = 2,
  show_test_summary = FALSE,
  ...
)
```

Arguments

x	a pvlrt object; an output of function pvlrt().
significance_level	numeric. Level of significance.
topn	number of top (with respect to likelihood ratio statistic value) pairs to show at the given significance level.
digits	number of digits to show after the decimal place.
show_test_summary	logical. Should a brief summary showing the top few test results be displayed? defaults to FALSE.
...	other input parameters. Currently unused.

Value

Invisibly returns the input pvlrt object.

See Also

pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

test1 <- pvlrt(statin46, nsim = 500)
print(test1)
```

---

pvlrt *Pseudo Likelihood Ratio Test for determining significant AE-Drug pairs under Poisson and zero-inflated Poisson models for pharmacovigilance*

Description

Pseudo Likelihood Ratio Test for determining significant AE-Drug pairs under Poisson and zero-inflated Poisson models for pharmacovigilance

Usage

```
pvlrt(
  contin_table,
  nsim = 10000,
  omega_vec = NULL,
  zi_prob = NULL,
  no_zi_idx = NULL,
  test_drug_idx = seq_len(max(ncol(contin_table) - 1, 0)),
  drug_class_idx = list(test_drug_idx),
  grouped_omega_est = FALSE,
  test_zi = NULL,
  test_omega = NULL,
  pval_ineq_strict = FALSE,
  parametrization = "rrr",
  null_boot_type = "parametric",
  is_zi_structural = TRUE,
  return_overall_loglik = TRUE,
  ...
)
```

Arguments

`contin_table`	IxJ contingency table showing pairwise counts of adverse events for I AE (along the rows) and J Drugs (along the columns)
`nsim`	Number of simulated null contingency table to use for computing the p-value of the test
`zi_prob, omega_vec`	Alias, determining zero inflation probabilities in the model. Can be a vector, providing different zero inflation probabilities for different drugs, or a scalar, providing the common zero inflation probability for all drugs. If NULL (default), then is estimated from the data. See also the description of the argument `grouped_omega_est`. If `omega_vec = rep(0, ncol(contin_table))`, then test reduces to an ordinary (non-zero inflated) Poisson test. NOTE: `zi_prob` and `omega_vec` are alias.
`no_zi_idx`	List of pairs (i, j) where zero inflation is not allowed. To specify the entirety i-th row (or j-th column) use c(i, 0) (or c(0, j)). See examples.
`test_drug_idx`	integer (position) or character (names) vector indicating the columns (drugs indices or drug labels) of contin_table to be tested for signal using LRT. Defaults to all except the last columns (which is typically the column for "Other Drugs").
`drug_class_idx`	a list, with the h-th entry providing the h-th group/class of drugs. Relevant only for drugs used for testing (supplied through `test_drug_idx`). By default all drugs provided in `test_drug_idx` are included in the same class, which is ensured by supplying `drug_class_idx = list(test_drug_idx)`. If more than one drug is present in a class, an extended LRT is performed for the class (which ensures the correct Type I error rate is preserved). If `drug_class_idx` excludes any drug present in `test_drug_idx`, each remaining drug is made to form its own class. See examples.

`grouped_omega_est`
: Logical. When performing a test with grouped drug classes (extended LRT), should the estimated zero-inflation parameter "omega" reflect the corresponding grouping? If TRUE, then the estimated omegas are obtained by combining columns from the same group, and if FALSE (default), then omegas are estimated separately for each drug (column) irrespective of the groups specified through `drug_class_idx`. Ignored if `omega_vec` is supplied/non-`NULL` (i.e., not estimated).

`test_zi, test_omega`
: logical indicators specifying whether to perform a pseudo likelihood ratio test for zero inflation. Defaults to FALSE. Ignored if `omega_vec` is supplied (is non-NULL). Defaults to FALSE. NOTE: `test_omega` and `test_zi` are aliases.

`pval_ineq_strict`
: logical. Use a strict inequality in the definition of the p-values? Defaults to FALSE.

`parametrization`
: Type of parametrization to use in the LR test. Available choices are "rrr", "lambda", "rr", and "p-q". The relative reporting ratio (default) parametrization of the test is used when when `parametrization %in% c("rrr", "lambda")`, and the reporting rate parametrization is used otherwise. NOTE: zero inflation can be handled only for the relative reporting ratio parametrization.

`null_boot_type`
: Type of bootstrap sampling to perform for generating null resamples. Available choices are "parametric" (default) and "non-parametric". NOTE: zero inflation is not handled properly in a non-parametric bootstrap resampling.

`is_zi_structural`
: logical. Do the inflated zeros correspond to structural zeros (indicating impossible AE-Drug combination)? This determines how the bootstrap null zero-inflation indicators are generated. If TRUE (default), then then the null zero-inflation random indicators are randomly generated using the (estimated) *conditional* probabilities of zero inflation given observed counts. If FALSE, then they are generated using the *marginal* (drug-specific) estimated probabilities of zero-inflation.

`return_overall_loglik`
: logical. Return overall log-likelihood for the table? This is needed if `logLik` method is to be used.

`...`
: additional arguments. Currently unused.

Value

The function returns an S3 object of class `pvlrt` containing test results. A `pvlrt` object is simply a re-classified matrix containing log likelihood ratio test statistics for cells in the input contingency table, with various other test and input data information (including p-values, estimated zero inflation parameters, overall log-likelihood etc.) embedded as attributes. The following S3 methods and functions are available for an `pvlrt` object:

Various post processing methods for `pvlrt` objects are available. This includes:

- bubbleplot_pvlrt
- extract_AE_names

- extract_Drug_names
- extract_lrstat_matrix
- extract_n_matrix
- extract_p_value_matrix
- extract_significant_pairs
- extract_zi_probability
- heatmap_pvlrt
- lrt_poisson
- lrt_vanilla_poisson
- lrt_zi_poisson
- r_contin_table_zip
- set_AE_names
- set_Drug_names
- print.pvlrt
- plot.pvlrt
- summary.pvlrt
- logLik.pvlrt
- as.matrix.pvlrt

Examples

```
data("statin46")

# 500 bootstrap iterations (nsim) in each example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

# no grouping -- each drug forms its own class,
# default "rrr" (lambda) parametrization, possible zi,
# null distribution evaluated using parametric bootstrap
test1 <- pvlrt(statin46, nsim = 500)
test1
## extract the observed LRT statistic
extract_lrstat_matrix(test1)
## extract the estimated omegas
extract_zi_probability(test1)

# grouped drugs --
# group 1: drug 1, drug 2
# group 2: drug 3, drug 4
# drug 5, 6, in their own groups
## 7 is not tested, so excluded from test_drug_idx (default)
## if needed, include 7 in test_drug_idx
drug_groups <- list(c(1, 2), c(3, 4))
```

```
## 5, 6 not present in drug_groups, so each will form their own groups
set.seed(50)
##
test2 <- pvlrt(statin46, drug_class_idx = drug_groups, nsim = 500)
test2

# instead of column positions column names can also be supplied
# in drug_class_idx and/or test_drug_idx
## column name version of drug_groups
drug_groups_colnames <- lapply(drug_groups, function(i) colnames(statin46)[i])
test_drug_colnames <- head(colnames(statin46), -1)
set.seed(50)
test20 <- pvlrt(
  statin46,
  test_drug_idx = test_drug_colnames,
  drug_class_idx = drug_groups_colnames,
  nsim = 500
)
test20
all.equal(test2, test20)


# specify no zero inflation at the entirety of the last row and the last column
no_zi_idx <- list(c(nrow(statin46), 0), c(0, ncol(statin46)))
test3 <- pvlrt(statin46, no_zi_idx = no_zi_idx, nsim = 500)
test3


# use non-parametric bootstrap to evaluate the null distribution
# takes longer, due to computational costs with non-parametric
# contigency table generation
test4 <- pvlrt(statin46, null_boot_type = "non-parametric", nsim = 500)
test4


# test zi probabilities (omegas)
test5 <- pvlrt(statin46, test_omega = TRUE, nsim = 500)
test5
```

`rv` *FDA rotavirus vaccine dataset with 794 adverse events observed among combined old (age >= 1 year) and young (age < 1 year) individuals*

Description

A vaccine-Adverse event (AE) count dataset (contingency table) obtained from the FDA VAERS database for the year 1999

Usage

```
rv
```

Format

An object of class `matrix` (inherits from `array`) with 794 rows and 2 columns.

Details

Data are stored in the forms of contingency table, with the vaccines listed across the columns and the 794 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that vaccine/AE pair and detected in the FDA VAERS database for the year 1999.

The dataset contains two columns – one for the rotavirus vaccine, and another for other (37 vaccines combined).

Source

https://vaers.hhs.gov/data/datasets.html

---

rvold *FDA rotavirus vaccine dataset with 727 adverse events observed among "old" (non-infant; age >= 1 year) individuals*

Description

A vaccine-Adverse event (AE) count dataset (contingency table) obtained from the FDA VAERS database for the year 1999

Usage

```
rvold
```

Format

An object of class `matrix` (inherits from `array`) with 727 rows and 2 columns.

Details

Data are stored in the forms of contingency table, with the vaccines listed across the columns and the 727 AEs presented across the rows. Each cell in the contingency table represents the total report counts (from "old" individuals with age >= 1 year) associated with that vaccine/AE pair and detected in the FDA VAERS database for the year 1999.

The dataset contains two columns – one for the rotavirus vaccine, and another for other (37 vaccines combined).

Source

https://vaers.hhs.gov/data/datasets.html

`rvyoung` *FDA rotavirus vaccine dataset with 346 adverse events observed among young (infant – 1 year) individuals*

Description

A vaccine-Adverse event (AE) count dataset (contingency table) obtained from the FDA VAERS database for the year 1999

Usage

```
rvyoung
```

Format

An object of class `matrix` (inherits from `array`) with 346 rows and 2 columns.

Details

Data are stored in the forms of contingency table, with the vaccines listed across the columns and the 346 AEs presented across the rows. Each cell in the contingency table represents the total report counts from young individuals with age < 1 year associated with that vaccine/AE pair and detected in the FDA VAERS database for the year 1999.

The dataset contains two columns – one for the rotavirus vaccine, and another for other (37 vaccines combined).

Source

https://vaers.hhs.gov/data/datasets.html

`r_contin_table_zip` *Generate random contingency tables for adverse event (across rows) and drug (across columns) combinations given row and column marginal totals, embedded signals, and possibly with zero inflation*

Description

Generate random contingency tables for adverse event (across rows) and drug (across columns) combinations given row and column marginal totals, embedded signals, and possibly with zero inflation

Usage

```
r_contin_table_zip(
  n = 1,
  row_marginals,
  col_marginals,
  signal_mat = matrix(1, length(row_marginals), length(col_marginals)),
  omega_vec = rep(0, length(col_marginals)),
  no_zi_idx = NULL,
  ...
)
```

Arguments

n	number of random matrices to generate.
row_marginals, col_marginals	(possibly named) vector of row and column marginal totals. Must add up to the same total. If named, the names are passed on to the randomly generated matrix/matrices.
signal_mat	numeric matrix of dimension length(row_marginals) x length(col_marginals). The (i, j)-th entry of signal_mat determines the signal strength of the i-th adverse event and j-th drug pair. The default is 1 for each pair, which means no signal for the pair.
omega_vec	vector of zero inflation probabilities. Must be of the same length as col_marginals.
no_zi_idx	List of pairs (i, j) where zero inflation is not allowed. To specify the entirety i-th row (or j-th column) use c(i, 0) (or c(0, j)). See examples.
...	Additional arguments. Currently unused.

Value

A list of length n, with each entry being a length(row_marginal) by length(col_marginal) matrix.

Examples

```
set.seed(42)

# first load the 46 statin data
data(statin46)
# zero inflation probabilities
omega_tru <- c(rep(0.15, ncol(statin46) - 1), 0)

# signal matrix
signal_mat <- matrix(1, nrow(statin46), ncol(statin46))

# "positive" signal at the (1, 1) entry
# the first column
signal_mat[1, 1] <- 10
```

```
# Now simulate data with the same row/column marginals
# as in statin46, with embedded signals as specified in
# the above signal_mat

# no zero inflation at (1, 1)
# (where signal is elicited) and the last row ("Other PT")
# and at the last column ("Other drugs") of the simulated matrix

sim_statin <- r_contin_table_zip(
  n = 1,
  row_marginals = rowSums(statin46),
  col_marginals = colSums(statin46),
  signal_mat = signal_mat,
  omega_vec = omega_tru,
  no_zi_idx = list(
    c(1, 1),
    c(nrow(statin46), 0), # the entire last row
    c(0, ncol(statin46)) # the entire last column
  )
)[[1]]

# now analyze the above simulated data

# using a pseudo LRT with a ZIP model
test1 <- pvlrt(
  contin_table = sim_statin,
  nsim = 500
  # set to 500 for demonstration purposes only,
  # we recommend the default 10000 or bigger
)
extract_zi_probability(test1)
extract_significant_pairs(test1)

# LRT with a poisson model
test2 <- lrt_poisson(
  contin_table = sim_statin,
  nsim = 500
  # set to 500 for demonstration purposes only,
  # we recommend the default 10000 or bigger
)
extract_zi_probability(test2)
extract_significant_pairs(test2)

# LRT with true omega supplied
test3 <- pvlrt(
  contin_table = sim_statin,
  zi_prob = omega_tru,
  nsim = 500
  # set to 500 for demonstration purposes only,
  # we recommend the default 10000 or bigger
)
extract_zi_probability(test3)
```

```
extract_significant_pairs(test3)
```

`statin` *FDA Statin dataset with 6039 adverse events*

Description

A Drug-Adverse event (AE) count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4

Usage

```
statin
```

Format

An object of class `matrix` (inherits from `array`) with 6039 rows and 7 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 6039 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that drug/AE pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset catalogs 6 statin drugs (across columns):

Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin, Simvastatin

Corresponding to all 6039 observed adverse events (AEs) observed in statins

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

---

`statin1491` *FDA Statin dataset with 1491 adverse events*

Description

A Drug-Adverse event (AE) count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4

Usage

```
statin1491
```

Format

An object of class `matrix` (inherits from `array`) with 1491 rows and 7 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 1491 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that drug/AE pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset catalogs 6 statin drugs (across columns):

Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin, Simvastatin

The 1491 AEs stored in the dataset represent the intersection of adverse events of the statin class of drugs and the GBCA drugs

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

See Also

statin46, statin, gbca

---

statin46 *FDA Statin dataset with 46 adverse events*

Description

A Drug-Adverse event (AE) count dataset (contingency table) obtained from the FDA FAERS database for the quarters 2014Q3 - 2020Q4

Usage

```
statin46
```

Format

An object of class `matrix` (inherits from `array`) with 47 rows and 7 columns.

Details

Data are stored in the form of a contingency table, with drugs listed across the columns and the 46 AEs presented across the rows. Each cell in the contingency table represents the total report counts associated with that drug/AE pair and detected in the FDA FAERS database during 2014Q3 - 2020Q4.

The dataset catalogs 6 statin drugs (across columns):

Atorvastatin, Fluvastatin, Lovastatin, Pravastatin, Rosuvastatin, Simvastatin

The 46 adverse events presented across the rows are considered significant by FDA.

Source

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

See Also

statin, statin1491, gbca

`summary.pvlrt` *Summary method for a pvlrt object*

Description

Summary method for a pvlrt object

Usage

```
## S3 method for class 'pvlrt'
summary(object, show_zi = FALSE, ...)
```

Arguments

`object`	a `pvlrt` object, which is the output of the function pvlrt or one of its wrappers such as lrt_zi_poisson, lrt_poisson and `lrt_vanilla_poisson`.
`show_zi`	logical. Should summary of the estimates and tests (if performed) of the zero inflation parameters be returned? Defaults to FALSE. If TRUE, then the zero inflation summary is included as an attribute with name "zi". See examples.
`...`	other input parameters. Currently unused.

Value

Returns a data.table with rows corresponding to all possible AE/Drug pairs as obtained from the input contingency table, and columns titled "AE", "Drug", "n", "lrstat" (log-likelihood ratio test statistic) and "p_value". Additionally, if `show_zi` is set to `TRUE`, then as an attribute named "zi" a data.table with rows corresponding to Drugs (columns in the input contingency table), and columns titled "AE", "zi", "lrstat" (log-likelihood ratio test statistic for zero-inflation), "p_value" and "q_value" (Benjamini-Hochberg adjusted p-values, as obtained through p.adjust) is returned.

See Also

pvlrt

Examples

```
# 500 bootstrap iterations (nsim) in the example below
# are for quick demonstration only --
# we recommended setting nsim to 10000 (default) or bigger

test1 <- pvlrt(statin46, test_zi = TRUE, nsim = 500)
summary(test1)
tmp <- summary(test1, show_zi = TRUE)
print(tmp)
tmp_zi <- attr(tmp, "zi")
print(tmp_zi)
```

Index