

# Package ‘pvda’

January 17, 2025

**Type** Package

**Title** Disproportionality Functions for Pharmacovigilance

**Version** 0.0.4

**Description** Tools for performing disproportionality analysis using the information component, proportional reporting rate and the reporting odds ratio. The anticipated use is passing data to the `da()` function, which executes the disproportionality analysis. See Norén et al (2011) <[doi:10.1177/0962280211403604](https://doi.org/10.1177/0962280211403604)> and Montastruc et al (2011) <[doi:10.1111/j.1365-2125.2011.04037.x](https://doi.org/10.1111/j.1365-2125.2011.04037.x)> for further details.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr (>= 1.43), rmarkdown (>= 2.24), testthat (>= 3.1.10), writexl (>= 1.4.2)

**Config/testthat/edition** 3

**BuildVignettes** true

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**Imports** checkmate (>= 2.1.0), cli (>= 3.6.3), data.table (>= 1.14.6), dplyr (>= 1.0.10), dtplyr (>= 1.2.2), glue (>= 1.6.2), purrr (>= 0.3.5), Rdpack (>= 2.4), rlang (>= 1.0.6), stats (>= 4.1.3), stringr (>= 1.5.0), tibble (>= 3.1.8), tidyr (>= 1.3.0), tidyselect (>= 1.2.0), utils (>= 4.1.3)

**Depends** R (>= 2.10)

**URL** <https://oskargauffin.github.io/pvda/>

**BugReports** <https://github.com/OskarGauffin/pvda/issues>

**RdMacros** Rdpack

**NeedsCompilation** no

**Author** Oskar Gauffin [aut] (<<https://orcid.org/0000-0003-1593-356X>>), Michele Fusaroli [cre] (<<https://orcid.org/0000-0002-0254-2212>>)

**Maintainer** Michele Fusaroli <[michele.fusaroli@who-umc.org](mailto:michele.fusaroli@who-umc.org)>

**Repository** CRAN

**Date/Publication** 2025-01-17 09:10:14 UTC

## Contents

add_disproportionality	2
add_expected_counts	3
apply_rule_of_N	4
build_colnames_da	5
ci_for_ic	5
ci_for_prr	6
ci_for_ror	7
conf_lvl_to_quantile_prob	8
count_expected_prr	8
count_expected_ror	9
count_expected_rrr	9
da	10
drug_event_df	12
grouped_da	13
ic	14
print.da	15
prr	16
ror	18
round_and_sort_by_lower_da_limit	19
round_columns_with_many_decimals	20
summary.da	21
tiny_dataset	21
write_to_excel	22

**Index** **23**

---

add\_disproportionality

*Add disproportionality estimates to data frame with expected counts*

---

### Description

Add disproportionality estimates to data frame with expected counts

### Usage

```
add_disproportionality(
  df = NULL,
  df_syms = NULL,
  da_estimators = c("ic", "prr", "ror"),
  rule_of_N = 3,
  conf_lvl = 0.95
)
```

**Arguments**

df	Intended use is on the output tibble from add_expected_counts.
df_syms	A list built from df_colnames through conversion to symbols.
da_estimators	Character vector specifying which disproportionality estimators to use, in case you don't need all implemented options. Defaults to c("ic", "pr", "ror").
rule_of_N	Numeric value. Sets estimates for ROR and PRR to NA when observed counts are strictly less than the passed value of rule_of_N. Default value is 3, 5 is sometimes used as a more liberal alternative. Set to NULL if you don't want to apply any such rule.
conf_lvl	Confidence level of confidence or credibility intervals. Default is 0.95 (i.e. 95 % confidence interval).

**Value**

The passed data frame with disproportionality point and interval estimates.

---

add\_expected\_counts     *Produces expected counts*

---

**Description**

Produces various counts used in disproportionality analysis.

**Usage**

```
add_expected_counts(
  df = NULL,
  df_colnames = NULL,
  df_syms = NULL,
  expected_count_estimators = c("rrr", "pr", "ror")
)
```

**Arguments**

df	An object possible to convert to a data table, e.g. a tibble or data.frame, containing patient level reported drug-event-pairs. See header 'The df object' below for further details.
df_colnames	A list of column names to use in df. That is, point da to the 'report id'-column (report_id), the 'drug name'-column (drug), the 'adverse event'-column (event) and optionally a grouping column group_by to calculate disproportionality across. See the vignette for further details.
df_syms	A list built from df_colnames through conversion to symbols.
expected_count_estimators	A character vector containing the desired expected count estimators. Defaults to c("rrr", "pr", "ror").

**Value**

A tibble containing the various counts.

**The df object**

The passed df should be (convertible to) a data table and at least contain three columns: report\_id, drug and event. The data table should contain one row per reported drug-event-combination, i.e. receiving a single additional report for drug X and event Y would add one row to the table. If the single report contained drug X for event Y and event Z, two rows would be added, with the same report\_id and drug on both rows. Column report\_id must be of type numeric or character. Columns drug and event must be of type character. If column group\_by is provided, it can be either numeric or character. You can use a df with column names of your choosing, as long as you connect role and name in the df\_colnames-parameter.

---

apply_rule_of_N	<i>apply_rule_of_N</i>
-----------------	------------------------

---

**Description**

Internal function to set disproportionality cells for ROR and PRR to NA when observed count < 3

**Usage**

```
apply_rule_of_N(
  da_df = NULL,
  da_estimators = c("ic", "pr", "ror"),
  rule_of_N = NULL
)
```

**Arguments**

da_df	See the intermediate object da_df in add_disproportionality
da_estimators	Default is c("ic", "pr", "ror").
rule_of_N	An length one integer between 0 and 10.

**Details**

Sometimes, you want to protect yourself from spurious findings based on small observed counts combined with infinitesimal expected counts.

**Value**

The input data frame (da\_df) with potentially some cells set to NA.

---

build_colnames_da	<i>An internal function creating colnames for da confidence/credibility bounds</i>
-------------------	--

---

### Description

Given the output from `quantile_prob`, and a `da_name` string, create column names such as PRR025, ROR025 and IC025

### Usage

```
build_colnames_da(
  quantile_prob = list(lower = 0.025, upper = 0.975),
  da_name = NULL
)
```

### Arguments

<code>quantile_prob</code>	A list with two parameters, lower and upper. Default: <code>list(lower = 0.025, upper = 0.975)</code>
<code>da_name</code>	A string, such as "ic", "prr" or "ror". Default: NULL

### Value

A list with two symbols, to be inserted in the `dtplyr-chain`

---

ci_for_ic	<i>Confidence intervals for Information Component (IC)</i>
-----------	--

---

### Description

Mainly used in function `ic`. Produces quantiles of the posterior gamma distribution. Called twice in `ic` to create credibility intervals.

### Usage

```
ci_for_ic(obs, exp, conf_lvl_probs, shrinkage)
```

### Arguments

<code>obs</code>	A numeric vector with observed counts, i.e. number of reports for the selected drug-event-combination. Note that shrinkage (e.g. +0.5) is added inside the function and should not be included here.
<code>exp</code>	A numeric vector with expected counts, i.e. number of reports to be expected given a comparator or <i>background</i> . Note that shrinkage (e.g. +0.5) is added inside the function and should not be included here.

conf_lv1_probs	The probabilities of the posterior, based on a passed confidence level (conf_lv1) in <a href="#">ic</a> . For instance, if sgn_lv1 = .95 in <a href="#">ic</a> is used, quantiles will be extracted at sgn_lv1_probs 0.025 and 0.975.
shrinkage	A non-negative numeric value, to be added to observed and expected count. Default is 0.5.

**Value**

The credibility interval specified by input parameters.

**See Also**

[ic](#)

---

ci\_for\_prr

*Confidence intervals for Proportional Reporting Rate*

---

**Description**

Mainly for use in [prr](#). Produces (symmetric, normality based) confidence bounds for the PRR, for a passed probability. Called twice in [prr](#) to create confidence intervals.

**Usage**

```
ci_for_prr(
  obs = NULL,
  n_drug = NULL,
  n_event_prr = NULL,
  n_tot_prr = NULL,
  conf_lv1_probs = 0.95
)
```

**Arguments**

obs	Number of reports for the specific drug and event (i.e. the observed count).
n_drug	Number of reports with the drug of interest.
n_event_prr	Number of reports with the event in the background.
n_tot_prr	Number of reports in the background.
conf_lv1_probs	The probabilities of the normal distribution, based on a passed confidence level (conf_lv1) in <a href="#">prr</a> . If sgn_lv1 = .95 in <a href="#">prr</a> , quantiles of the normal distribution will be extracted at sgn_lv1_probs of 0.025 and 0.975.

**Value**

The confidence interval specified by input parameters.

**See Also**[prp](#)

---

`ci_for_ror`*Confidence intervals for Reporting Odds Ratio*

---

**Description**

Mainly for use in [ror](#). Produces (symmetric, normality based) confidence bounds for the ROR, for a passed probability. Called twice in [ror](#) to create confidence intervals.

**Usage**

```
ci_for_ror(a, b, c, d, conf_lvl_probs)
```

**Arguments**

- |                             |   |
|-----------------------------|---|
| <code>a</code>              | Number of reports for the specific drug and event (i.e. the observed count).  |
| <code>b</code>              | Number of reports with the drug, without the event  |
| <code>c</code>              | Number of reports without the drug, with the event  |
| <code>d</code>              | Number of reports without the drug, without the event   |
| <code>conf_lvl_probs</code> | The probabilities of the normal distribution, based on a passed confidence level ( <code>conf_lvl</code> ) in <a href="#">ror</a> . If <code>sgn_lvl = .95</code> in <a href="#">ror</a> , quantiles of the normal distribution will be extracted at <code>sgn_lvl_probs</code> of 0.025 and 0.975. |

**Value**

The credibility interval specified by input parameters.

**See Also**[ror](#)

---

`conf_lvl_to_quantile_prob`*Quantile probabilities from confidence level*

---

**Description**

Calculates equi-tailed quantile probabilities from a confidence level

**Usage**

```
conf_lvl_to_quantile_prob(conf_lvl = 0.95)
```

**Arguments**

`conf_lvl` Confidence level of confidence or credibility intervals. Default is 0.95 (i.e. 95 % confidence interval).

**Value**

A list with two numerical vectors, "lower" and "upper".

**Examples**

```
conf_lvl_to_quantile_prob(0.95)
```

---

`count_expected_prr`*Count expected for Proportional Reporting Rate*

---

**Description**

Internal function to provide expected counts related to the PRR

**Usage**

```
count_expected_prr(count_dt)
```

**Arguments**

`count_dt` A data table, output from `count_expected_rrr`

**Value**

A data table with added columns for `n_event_prr` `n_tot_prr` and `expected_prr @export`



---

count_expected_ror	<i>Count expected for Reporting Odds Ratio</i>
--------------------	--

---

**Description**

Internal function to provide expected counts related to the ROR

**Usage**

```
count_expected_ror(count_dt)
```

**Arguments**

count_dt	A data table, output from count_expected_rrr
----------	--

**Details**

DETAILS

**Value**

A data table with added columns for n\_event\_prr, n\_tot\_prr and expected\_prr  
OUTPUT\_DESCRIPTION

**See Also**

[mutate](#), [select everything](#)

---

count_expected_rrr	<i>Count Expected for Relative Reporting Rate</i>
--------------------	---

---

**Description**

Internal function to provide expected counts related to the RRR

**Usage**

```
count_expected_rrr(df, df_colnames, df_syms)
```

**Arguments**

df	See documentation for add_expected_counts
df_colnames	See documentation for da
df_syms	A list built from df_colnames through conversion to symbols.

**Value**

A data frame with columns for obs, n\_drug, n\_event, n\_tot and (RRR) expected

---

 da *Disproportionality Analysis*


---

**Description**

The function `da` executes disproportionality analyses, i.e. compares the proportion of reports with a specific adverse event for a drug, against an event proportion from a comparator based on the passed data frame. See the vignette for a brief introduction to disproportionality analysis. Furthermore, `da` supports three estimators: Information Component (IC), Proportional Reporting Rate (PRR) and the Reporting Odds Ratio (ROR).

**Usage**

```
da(
  df = NULL,
  df_colnames = list(report_id = "report_id", drug = "drug", event = "event", group_by =
    NULL),
  da_estimators = c("ic", "pr", "ror"),
  sort_by = "ic",
  number_of_digits = 2,
  rule_of_N = 3,
  conf_lvl = 0.95,
  excel_path = NULL
)
```

**Arguments**

<code>df</code>	An object possible to convert to a data table, e.g. a tibble or <code>data.frame</code> , containing patient level reported drug-event-pairs. See header 'The df object' below for further details.
<code>df_colnames</code>	A list of column names to use in <code>df</code> . That is, point <code>da</code> to the 'report id'-column ( <code>report_id</code> ), the 'drug name'-column ( <code>drug</code> ), the 'adverse event'-column ( <code>event</code> ) and optionally a grouping column <code>group_by</code> to calculate disproportionality across. See the vignette for further details.
<code>da_estimators</code>	Character vector specifying which disproportionality estimators to use, in case you don't need all implemented options. Defaults to <code>c("ic", "pr", "ror")</code> .
<code>sort_by</code>	The output is sorted in descending order of the lower bound of the confidence/credibility interval for a passed <code>da</code> estimator. Any of the passed strings in " <code>da_estimators</code> " is accepted, the default is "ic". If a grouping variable is passed, sorting is made by the sample average across each drug-event-combination (ignoring NAs).
<code>number_of_digits</code>	Round decimal columns to specified precision, default is two decimals.
<code>rule_of_N</code>	Numeric value. Sets estimates for ROR and PRR to NA when observed counts are strictly less than the passed value of <code>rule_of_N</code> . Default value is 3, 5 is sometimes used as a more liberal alternative. Set to NULL if you don't want to apply any such rule.

conf_lvl	Confidence level of confidence or credibility intervals. Default is 0.95 (i.e. 95 % confidence interval).
excel_path	Intended for users who prefer to work in excel with minimal work in R. To write the output of da to an excel file, provide a path to a folder. For instance, to write to your current working directory, pass <code>getwd()</code> . The excel file will by default be named <code>da.xlsx</code> . To control the excel file name, pass a path ending with the desired filename suffixed with <code>.xlsx</code> . If you do not want to export the output to an excel file, pass <code>NULL</code> (the default).

### Value

da returns a data frame (invisibly) containing counts and estimates related to supported disproportionality estimators. Each row corresponds to a drug-event pair.

### The df object

The passed df should be (convertible to) a data table and at least contain three columns: `report_id`, `drug` and `event`. The data table should contain one row per reported drug-event-combination, i.e. receiving a single additional report for drug X and event Y would add one row to the table. If the single report contained drug X for event Y and event Z, two rows would be added, with the same `report_id` and `drug` on both rows. Column `report_id` must be of type numeric or character. Columns `drug` and `event` must be of type character. If column `group_by` is provided, it can be either numeric or character. You can use a df with column names of your choosing, as long as you connect role and name in the `df_colnames`-parameter.

### Examples

```
### Run a disproportionality analysis

da_1 <-
  tiny_dataset |>
  da()

### Run a disproportionality across subgroups
list_of_colnames <-
  list(
    report_id = "report_id",
    drug = "drug",
    event = "event",
    group_by = "group"
  )

da_2 <-
  tiny_dataset |>
  da(df_colnames = list_of_colnames)

# If columns in your df have different names than the default ones,
# you can specify the column names in the df_colnames parameter list:

renamed_df <-
  tiny_dataset |>
```

```
dplyr::rename(ReportID = report_id)

list_of_colnames$report_id <- "ReportID"

da_3 <-
  renamed_df |>
  da(df_colnames = list_of_colnames)
```

---

drug\_event\_df

*A simulated ICSR database*

---

## Description

drug\_event\_df is a simulated dataset, slightly larger than the "tiny\_dataset" which is also contained in this package.

## Usage

```
drug_event_df
```

## Format

‘drug\_event\_df’ A data frame with 3,971 rows and 3 columns. In total 1000 unique report\_ids, i.e. the same report\_id can have several drugs and events.

Number of drugs per report\_id is sampled as  $1 + \text{Pois}(3)$ , with increasing probability as the drug letter closes in on Z. Every drug is assigned an event, with decreasing probability as the event index number increases towards 1000. See the DATASET.R file in the data-raw folder for details.

**report\_id** A patient or report identifier

**drug** One of 26 fake drugs (Drug\_A - Drug\_Z)

**event** Sampled events (Event\_1 - Event\_1000)

## Source

Simulated data.

**Description**

A package internal wrapper for executing da across subgroups

**Usage**

```
grouped_da(  
  df = NULL,  
  df_colnames = NULL,  
  df_syms = NULL,  
  expected_count_estimators = NULL,  
  da_estimators = NULL,  
  sort_by = NULL,  
  conf_lvl = NULL,  
  rule_of_N = NULL,  
  number_of_digits = NULL  
)
```

**Arguments**

df	See the da function
df_colnames	See the da function
df_syms	A list built from df_colnames through conversion to symbols.
expected_count_estimators	See the da function
da_estimators	See the da function
sort_by	See the da function
conf_lvl	See the da function
rule_of_N	See the da function
number_of_digits	See the da function

**Details**

See the da documentation

**Value**

See the da function

---

ic *Information component*

---

### Description

Calculates the information component ("IC") and credibility interval, used in disproportionality analysis.

### Usage

```
ic(obs = NULL, exp = NULL, shrinkage = 0.5, conf_lvl = 0.95)
```

### Arguments

obs	A numeric vector with observed counts, i.e. number of reports for the selected drug-event-combination. Note that shrinkage (e.g. +0.5) is added inside the function and should not be included here.
exp	A numeric vector with expected counts, i.e. number of reports to be expected given a comparator or <i>background</i> . Note that shrinkage (e.g. +0.5) is added inside the function and should not be included here.
shrinkage	A non-negative numeric value, to be added to observed and expected count. Default is 0.5.
conf_lvl	Confidence level of confidence or credibility intervals. Default is 0.95 (i.e. 95 % confidence interval).

### Details

The IC is a log<sub>2</sub>-transformed observed-to-expected ratio, based on the relative reporting rate (RRR) for counts, but modified with an addition of "shrinkage" to protect against spurious associations.

$$\hat{IC} = \log_2\left(\frac{\hat{O} + k}{\hat{E} + k}\right)$$

where  $\hat{O}$  = observed number of reports,  $k$  is the shrinkage (typically +0.5), and expected  $\hat{E}$  is (for RRR, and using the entire database as comparator or *background*) estimated as

$$\hat{E} = \frac{\hat{N}_{drug} \times \hat{N}_{event}}{\hat{N}_{TOT}}$$

where  $\hat{N}_{drug}$ ,  $\hat{N}_{event}$  and  $\hat{N}_{TOT}$  are the number of reports with the drug, the event, and in the whole database respectively.

The credibility interval is created from the quantiles of the posterior gamma distribution with shape ( $\hat{S}$ ) and rate ( $\hat{R}$ ) parameters as

$$\hat{S} = \hat{O} + k$$

$$\hat{R} = \hat{E} + k$$

using the `stats::qgamma` function. Parameter  $k$  is the shrinkage defined earlier. For completeness, a credibility interval of the gamma distributed  $X$  (i.e.  $X \sim \Gamma(\hat{S}, \hat{R})$  where  $\hat{S}$  and  $\hat{R}$  are shape and rate parameters) with associated quantile function  $Q_X(p)$  for a significance level  $\alpha$  is constructed as

$$[Q_X(\alpha/2), Q_X(1 - \alpha/2)]$$

### Value

A tibble with three columns (point estimate and credibility bounds).

### Further details

From a bayesian point-of-view, the credibility interval of the IC is constructed from the poisson-gamma conjugacy. The shrinkage constitutes a prior of observed and expected of 0.5. A shrinkage of +0.5 with a gamma-quantile based 95 % credibility interval cannot have lower bound above 0 unless the observed count exceeds 3. One benefit of  $\log_2$  is to provide a log-scale for convenient plotting of multiple IC values side-by-side.

### References

Norén GN, Hopstadius J, Bate A (2011). “Shrinkage observed-to-expected ratios for robust and transparent large-scale pattern discovery.” *Statistical Methods in Medical Research*, **22**(1), 57–69. doi:10.1177/0962280211403604, <https://doi.org/10.1177/0962280211403604>.

### Examples

```
ic(obs = 20, exp = 10)
# Note that obs and exp can be vectors (of equal length, no recycling allowed)
ic(obs = c(20, 30), exp = c(10, 10))
```

---

print.da

*print function for da objects*

---

### Description

print function for da objects

### Usage

```
## S3 method for class 'da'
print(x, n = 10, ...)
```

**Arguments**

x	A S3 obj of class "da", output from <code>pvda::da()</code> .
n	Control the number of rows to print.
...	For passing additional parameters to extended classes.

**Value**

Nothing, but prints the tibble `da_df` in the `da` object.

**Examples**

```
da_1 <-
  tiny_dataset |>
  da()
print(da_1)
```

---

pr

*Proportional Reporting Rate*

---

**Description**

Calculates Proportional Reporting Rate ("PRR") with confidence intervals, used in disproportional-ity analysis.

**Usage**

```
pr(
  obs = NULL,
  n_drug = NULL,
  n_event_prr = NULL,
  n_tot_prr = NULL,
  conf_lvl = 0.95
)
```

**Arguments**

obs	Number of reports for the specific drug and event (i.e. the observed count).
n_drug	Number of reports with the drug of interest.
n_event_prr	Number of reports with the event in the background.
n_tot_prr	Number of reports in the background.
conf_lvl	Confidence level of confidence or credibility intervals. Default is 0.95 (i.e. 95 % confidence interval).



## Details

The PRR is the proportion of reports with an event in set of exposed cases, divided with the proportion of reports with the event in a background or comparator, which does not include the exposed.

The PRR is estimated from a observed-to-expected ratio, based on similar to the RRR and IC, but excludes the exposure of interest from the comparator.

$$P\hat{R}R = \frac{\hat{O}}{\hat{E}}$$

where  $\hat{O}$  is the observed number of reports, and expected  $\hat{E}$  is estimated as

$$\hat{E} = \frac{\hat{N}_{drug} \times (\hat{N}_{event} - \hat{O})}{\hat{N}_{TOT} - \hat{N}_{drug}}$$

where  $\hat{N}_{drug}$ ,  $\hat{N}_{event}$ ,  $\hat{O}$  and  $\hat{N}_{TOT}$  are the number of reports with the drug, the event, the drug and event, and in the whole database respectively.

A confidence interval is derived in Gravel (2009) using the delta method:

$$\hat{s} = \sqrt{1/\hat{O} - 1/(\hat{N}_{drug}) + 1/(\hat{N}_{event} - \hat{O}) - 1/(\hat{N}_{TOT} - \hat{N}_{drug})}$$

and

$$[\hat{C}I_{\alpha/2}, \hat{C}I_{1-\alpha/2}] = \left[ \frac{\hat{O}}{\hat{E}} \times \exp(Q_{\alpha/2} \times \hat{s}), \frac{\hat{O}}{\hat{E}} \times \exp(Q_{1-\alpha/2} \times \hat{s}) \right]$$

where  $Q_{\alpha}$  denotes the quantile function of a standard Normal distribution at significance level  $\alpha$ .

Note: For historical reasons, another version of this standard deviation is sometimes used where the last fraction under the square root is added rather than subtracted, with negligible practical implications in large databases. This function uses the version declared above, i.e. with subtraction.

## Value

A tibble with three columns (point estimate and credibility bounds). Number of rows equals length of inputs obs, n\_drug, n\_event\_prr and n\_tot\_prr.

## References

- Montastruc J, Sommet A, Bagheri H, Lapeyre-Mestre M (2011). "Benefits and strengths of the disproportionality analysis for identification of adverse drug reactions in a pharmacovigilance database." *British Journal of Clinical Pharmacology*, **72**(6), 905–908. doi:10.1111/j.13652125.2011.04037.x, <https://doi.org/10.1111/j.1365-2125.2011.04037.x>.
- Gravel C (2009). "Statistical Methods for Signal Detection in Pharmacovigilance." <https://repository.library.carleton.ca/downloads/jd472x08w>.

**Examples**

```

prc(
  obs = 5,
  n_drug = 10,
  n_event_prr = 20,
  n_tot_prr = 10000
)

# Note that input parameters can be vectors (of equal length, no recycling)
pvda::prc(
  obs = c(5, 10),
  n_drug = c(10, 20),
  n_event_prr = c(15, 30),
  n_tot_prr = c(10000, 10000)
)

```

ror

*Reporting Odds Ratio***Description**

Calculates Reporting Odds Ratio ("ROR") and confidence intervals, used in disproportionality analysis.

**Usage**

```
ror(a = NULL, b = NULL, c = NULL, d = NULL, conf_lvl = 0.95)
```

**Arguments**

a	Number of reports for the specific drug and event (i.e. the observed count).
b	Number of reports with the drug, without the event
c	Number of reports without the drug, with the event
d	Number of reports without the drug, without the event
conf_lvl	Confidence level of confidence or credibility intervals. Default is 0.95 (i.e. 95 % confidence interval).

**Details**

The ROR is an odds ratio calculated from reporting counts. The R for Reporting in ROR is meant to emphasize an interpretation of reporting, as the ROR is calculated from a reporting database. Note: the function is vectorized, i.e. a, b, c and d can be vectors, see the examples.

A reporting odds ratio is simply an odds ratio based on adverse event reports.

$$R\hat{O}R = \frac{a/b}{c/d}$$

where  $a$  = observed count (i.e. number of reports with exposure and outcome),  $b$  = number of reports with the drug and without the event,  $c$  = number of reports without the drug with the event and  $d$  = number of reports with neither of the drug and the event.

A confidence interval for the ROR can be derived through the delta method, with a standard deviation:

$$\hat{s} = \sqrt{1/a + 1/b + 1/c + 1/d}$$

with the resulting confidence interval for significance level  $\alpha$

$$[\hat{ROR} \times \exp(\Phi_{\alpha/2} \times \hat{s}), \hat{ROR} \times \exp(\Phi_{1-\alpha/2} \times \hat{s})]$$

### Value

A tibble with three columns (point estimate and credibility bounds). Number of rows equals length of inputs a, b, c, d.

### References

Montastruc J, Sommet A, Bagheri H, Lapeyre-Mestre M (2011). “Benefits and strengths of the disproportionality analysis for identification of adverse drug reactions in a pharmacovigilance database.” *British Journal of Clinical Pharmacology*, **72**(6), 905–908. doi:10.1111/j.13652125.2011.04037.x, <https://doi.org/10.1111/j.1365-2125.2011.04037.x>.

### Examples

```
ror(
  a = 5,
  b = 10,
  c = 20,
  d = 10000
)

# Note that a, b, c and d can be vectors (of equal length, no recycling)
pvda::ror(
  a = c(5, 10),
  b = c(10, 20),
  c = c(15, 30),
  d = c(10000, 10000)
)
```

---

round\_and\_sort\_by\_lower\_da\_limit

*Sort a disproportionality analysis by the lower da conf. or cred. limit*

---

### Description

Sorts the output by the mean lower limit of a passed da estimator

**Usage**

```
round_and_sort_by_lower_da_limit(
  df = NULL,
  df_colnames = NULL,
  df_syms = NULL,
  conf_lvl = NULL,
  sort_by = NULL,
  da_estimators = NULL,
  number_of_digits = 2
)
```

**Arguments**

df	See add_disproportionality
df_colnames	See add_disproportionality
df_syms	See add_disproportionality
conf_lvl	See add_disproportionality
sort_by	See add_disproportionality
da_estimators	See add_disproportionality
number_of_digits	Numeric value. Set the number of digits to show in output by passing an integer. Default value is 2 digits. Set to NULL to avoid rounding.

**Value**

The df object, sorted.

---

round\_columns\_with\_many\_decimals

*Rounds columns in da\_df with many decimals*

---

**Description**

Internal function containing a mutate + across

**Usage**

```
round_columns_with_many_decimals(
  da_df = NULL,
  da_estimators = NULL,
  number_of_digits = NULL
)
```

**Arguments**

da\_df            See add\_disproportionality  
 da\_estimators   See add\_disproportionality  
 number\_of\_digits  
                   See add\_disproportionality

**Value**

A df with rounded columns

---

summary.da	<i>Summary function for disproportionality objects</i>
------------	--

---

**Description**

Provides summary counts of SDRs and shows the top five DECs

**Usage**

```
## S3 method for class 'da'
summary(object, print = TRUE, ...)
```

**Arguments**

object            A S3 obj of class "da", output from pvda::da().  
 print            Do you want to print the output to the console. Defaults to TRUE.  
 ...               For passing additional parameters to extended classes.

**Value**

Passes a tibble with the SDR counts invisibly.

---

tiny_dataset	<i>A 110 reports big, simulated ICSR database</i>
--------------	---

---

**Description**

The dataframe tiny\_dataset is used to demonstrate the functionality of the package in examples. The larger drug\_event\_df-dataset can also be used.

**Usage**

```
tiny_dataset
```

**Format**

'tiny\_dataset' A data frame with 110 rows and 3 columns. In total 110 unique report\_ids. In particular, for Drug A and Event 1 the observed count will be 4 and exp\_rrr = 1.1

**report\_id** A report identifier, 1-110.

**drug** Drugs named as Drug\_A - Drug\_Z.

**event** Events named as Event\_1 - Event\_97)

**group** In this example, sex of the patient, i.e. Male or Female.

**Source**

Simulated data.

---

write\_to\_excel

*Write to excel*

---

**Description**

Writes output from a disproportionality analysis to an excel file

**Usage**

```
write_to_excel(df, write_path = NULL)
```

**Arguments**

**df** The data frame to export. See '?da' for details.

**write\_path** A string giving the file path

**Value**

Nothing.

# Index

## \* datasets

drug\_event\_df, 12

tiny\_dataset, 21

add\_disproportionality, 2

add\_expected\_counts, 3

apply\_rule\_of\_N, 4

build\_colnames\_da, 5

ci\_for\_ic, 5

ci\_for\_prr, 6

ci\_for\_ror, 7

conf\_lvl\_to\_quantile\_prob, 8

count\_expected\_prr, 8

count\_expected\_ror, 9

count\_expected\_rrr, 9

da, 10

drug\_event\_df, 12

everything, 9

grouped\_da, 13

ic, 5, 6, 14

mutate, 9

print.da, 15

prr, 6, 7, 16

ror, 7, 18

round\_and\_sort\_by\_lower\_da\_limit, 19

round\_columns\_with\_many\_decimals, 20

select, 9

summary.da, 21

tiny\_dataset, 21

write\_to\_excel, 22