Package ‘twoxtwo’

July 22, 2025

Type Package

Title Work with Two-by-Two Tables

Version 0.1.0

Maintainer VP Nagraj `<nagraj@nagraj.net>`

Description A collection of functions for data analysis with two-by-two contingency tables. The package provides tools to compute measures of effect (odds ratio, risk ratio, and risk difference), calculate impact numbers and attributable fractions, and perform hypothesis testing. Statistical analysis methods are oriented towards epidemiological investigation of relationships between exposures and outcomes.

Imports dplyr, tidyr, forcats, magrittr, rlang, knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat, lifecycle, rmarkdown, ggplot2, purrr

Depends R (>= 2.10)

VignetteBuilder knitr

NeedsCompilation no

Author VP Nagraj [aut, cre]

Repository CRAN

Date/Publication 2021-07-09 09:00:02 UTC

Contents

twoxtwo-package 2
af 2
bound 4
chisq 4
display 5
fisher 6

format_measure 7
impact 7
measures 9
print.twoxtwo 11
summary.twoxtwo 12
titanic 12
twoxtwo 13

Index 15

`twoxtwo-package` *twoxtwo*

Description

Provides a collection of functions for data analysis with two-by-two contingency tables.

`af` *Attributable fractions*

Description

In addition to measures of effect such as odds ratio, risk ratio, and risk difference, the twoxtwo framework allows for calculation of attributable fractions: attributable risk proportion in the exposed (ARP) and the population attributable risk proportion (PARP).

Estimates of the attributable fractions can be calculated with the `arp()` and `parp()` functions respectively. Each function takes an input dataset and arguments for outcome and exposure as bare, unquoted variable names. If the input has the twoxtwo class then the effect measures will be calculated using exposure and outcome information from that object. The functions all return a tidy `tibble` with the name of the measure, the point estimate, and lower/upper bounds of a confidence interval (CI) based on the SE.

Formulas used in point estimate and SE calculations are available in 'Details'.

Usage

```
arp(.data, exposure, outcome, alpha = 0.05, percent = FALSE, ...)

parp(
  .data,
  exposure,
  outcome,
  alpha = 0.05,
  percent = FALSE,
  prevalence = NULL,
  ...
)
```

Arguments

`.data`	Either a data frame with observation-level exposure and outcome data or a twoxtwo object
`exposure`	Name of exposure variable; ignored if input to `.data` is a `twoxtwo` object
`outcome`	Name of outcome variable; ignored if input to `.data` is a `twoxtwo` object
`alpha`	Significance level to be used for constructing confidence interval; default is `0.05`
`percent`	Logical as to whether or not the measure should be returned as a percentage; default is `FALSE`
`...`	Additional arguments passed to twoxtwo function; ignored if input to `.data` is a `twoxtwo` object
`prevalence`	Prevalence of exposure in the population; must be numeric between `0` and `1`; only used in `parp()`; default is `NULL` and will be ignored

Details

The formulas below denote cell values as A,B,C,D. For more on `twoxtwo` notation see the twoxtwo documentation.

Note that formulas for standard errors are not provided below but are based on forumlas described in Hildebrandt et al (2006).

Attributable Risk Proportion in the Exposed (ARP):

$$ARP = 1 - (1/((A/(A+B))/(C/(C+D))))$$

Population Attributable Risk Proportion (PARP):

$$PARP = (((A+C)/(A+B+C+D)) - (C/(C+D))))/((A+C)/(A+B+C+D))$$

If "prevalence" argument is not `NULL` then the formula uses the value specified for prevalence of exposure (p):

$$PARP = p*(((A/(A+B))/(C/(C+D))) - 1)/(p*(((A/(A+B))/(C/(C+D))) - 1) + 1)$$

Value

A `tibble` with the following columns:

- **measure**: Name of the measure calculated
- **estimate**: Point estimate for the effect measure
- **ci_lower**: The lower bound of the confidence interval for the estimate
- **ci_upper**: The upper bound of the confidence interval for the estimate
- **exposure**: Name of the exposure variable followed by +/- levels (e.g. smoking::yes/no)
- **outcome**: Name of the outcome variable followed by +/- levels (e.g. heart_disease::yes/no)

References

Hildebrandt, M., Bender, R., Gehrmann, U., & Blettner, M. (2006). Calculating confidence intervals for impact numbers. BMC medical research methodology, 6, 32. https://doi.org/10.1186/1471-2288-6-32

Szklo, M., & Nieto, F. J. (2007). Epidemiology: Beyond the basics. Sudbury, Massachussets: Jones and Bartlett.

Zapata-Diomedi, B., Barendregt, J. J., & Veerman, J. L. (2018). Population attributable fraction: names, types and issues with incorrect interpretation of relative risks. British journal of sports medicine, 52(4), 212–213. https://doi.org/10.1136/bjsports-2015-095531

`bound` *Bound a vector*

Description

This unexported helper function bounds a numeric vector on a minimum and maximum value.

Usage

```
bound(x, min = 0.01, max = 0.99)
```

Arguments

`x`	Numeric vector to be bounded
`min`	Minimum allowed value for vector "x"; default is `0.01`
`max`	Maximum allowed value for vector "x"; default is `0.99`

Value

Numeric vector of the same length as `x` with no values less than `minimum` nor greater than `maximum`.

`chisq` *Pearson's chi-squared test*

Description

This function conducts a Pearson's chi-squared test for a `twoxtwo` constructed using the specified exposure and outcome. Internally the function uses chisq.test. The output of the function includes the chi-squared test statistic, degrees of freedom, and the p-value from the test.

Usage

```
chisq(.data, exposure, outcome, correct = TRUE, ...)
```

Arguments

`.data`	Either a data frame with observation-level exposure and outcome data or a twoxtwo object
`exposure`	Name of exposure variable; ignored if input to `.data` is a `twoxtwo` object
`outcome`	Name of outcome variable; ignored if input to `.data` is a `twoxtwo` object
`correct`	Logical as to whether or not to apply continuity correction; default is `TRUE`
`...`	Additional arguments passed to twoxtwo function; ignored if input to `.data` is a `twoxtwo` object

Value

A `tibble` with the following columns:

- **test**: Name of the test conducted
- **estimate**: Point estimate from the test (`NA` for `chisq()`)
- **ci_lower**: The lower bound of the confidence interval for the estimate (`NA` for `chisq()`)
- **ci_upper**: The upper bound of the confidence interval for the estimate (`NA` for `chisq()`)
- **statistic**: Test statistic from the test
- **df**: Degrees of freedom parameter for the test statistic
- **pvalue**: P-value from the test
- **exposure**: Name of the exposure variable followed by +/- levels (e.g. smoking::yes/no)
- **outcome**: Name of the outcome variable followed by +/- levels (e.g. heart_disease::yes/no)

`display` *Display twoxtwo object*

Description

This is a helper to render a twoxtwo object as a kable. The function extracts `twoxtwo` cell counts and uses exposure levels as row names and outcome levels as column names.

Usage

```
display(.twoxtwo, ...)
```

Arguments

`.twoxtwo`	twoxtwo object
`...`	Additional arguments passed to kable

Value

A `knitr_kable` object with the `twoxtwo` cell counts, exposure levels as row names, and outcome levels as column names.

`fisher` *Fisher's exact test*

Description

This function conducts a Fisher's exact test using specified exposure and outcome. Internally the function uses fisher.test to test independence of `twoxtwo` rows and columns. The output of the function includes the odds ratio, the lower/upper bounds for the confidence interval around the estimate, and the p-value from the test.

Usage

```
fisher(
  .data,
  exposure,
  outcome,
  alternative = "two.sided",
  conf_level = 0.95,
  or = 1,
  ...
)
```

Arguments

`.data`	Either a data frame with observation-level exposure and outcome data or a twoxtwo object
`exposure`	Name of exposure variable; ignored if input to `.data` is a `twoxtwo` object
`outcome`	Name of outcome variable; ignored if input to `.data` is a `twoxtwo` object
`alternative`	Alternative hypothesis for test; must be one of "two.sided", "greater", or "less"; default is `"two.sided"`
`conf_level`	Confidence level for the confidence interval; default is `0.95`
`or`	Hypothesized odds ratio; default is `1`
`...`	Additional arguments passed to twoxtwo function; ignored if input to `.data` is a `twoxtwo` object

Value

A `tibble` with the following columns:

- **test**: Name of the test conducted
- **estimate**: Point estimate from the test
- **ci_lower**: The lower bound of the confidence interval for the estimate
- **ci_upper**: The upper bound of the confidence interval for the estimate
- **statistic**: Test statistic from the test (`NA` for `fisher()`)

- **df**: Degrees of freedom parameter for the test statistic (`NA` for `fisher()`)
- **pvalue**: P-value from the test
- **exposure**: Name of the exposure variable followed by +/- levels (e.g. smoking::yes/no)
- **outcome**: Name of the outcome variable followed by +/- levels (e.g. heart_disease::yes/no)

`format_measure` *Format measure*

Description

This helper takes the output from a `twoxtwo` effect measure function and formats the point estimate and lower/upper bounds of the computed confidence interval (CI) as a string.

Usage

```
format_measure(.data, digits = 3)
```

Arguments

`.data`	Output from a twoxtwo effect measure function (e.g. odds_ratio)
`digits`	Number of digits; default is `3`

Value

A character vector of length 1 with the effect measure formatted as point estimate (lower bound of CI, upper bound of CI). The point estimate and CI are rounded to precision specified in "digits" argument.

`impact` *Impact numbers*

Description

Impact numbers are designed to communicate how impactful interventions and/or exposures can be on a population. The twoxtwo framework allows for calculation of impact numbers: exposure impact number (EIN), case impact number (CIN), and the exposed cases impact number (ECIN).

The `ein()`, `cin()`, and `ecin()` functions provide interfaces for calculating impact number estimates. Each function takes an input dataset and arguments for outcome and exposure as bare, unquoted variable names. If the input has the twoxtwo class then the measures will be calculated using exposure and outcome information from that object. The functions all return a tidy `tibble` with the name of the measure, the point estimate, and lower/upper bounds of a confidence interval (CI) based on the SE.

Formulas used in point estimate and SE calculations are available in ’Details’.

Usage

```
ein(.data, exposure, outcome, alpha = 0.05, ...)

cin(.data, exposure, outcome, alpha = 0.05, prevalence = NULL, ...)

ecin(.data, exposure, outcome, alpha = 0.05, ...)
```

Arguments

`.data`	Either a data frame with observation-level exposure and outcome data or a twoxtwo object
`exposure`	Name of exposure variable; ignored if input to `.data` is a `twoxtwo` object
`outcome`	Name of outcome variable; ignored if input to `.data` is a `twoxtwo` object
`alpha`	Significance level to be used for constructing confidence interval; default is `0.05`
`...`	Additional arguments passed to twoxtwo function; ignored if input to `.data` is a `twoxtwo` object
`prevalence`	Prevalence of exposure in the population; must be numeric between `0` and `1`; only used in `cin()`; default is `NULL` and will be ignored

Details

The formulas below denote cell values as A,B,C,D. For more on `twoxtwo` notation see the twoxtwo documentation.

Note that formulas for standard errors are not provided below but are based on forumlas described in Hildebrandt et al (2006).

Exposure Impact Number (EIN):

$$EIN = 1/((A/(A+B)) - (C/(C+D)))$$

Case Impact Number (CIN):

$$CIN = 1/(((A+C)/(A+B+C+D)) - (C/(C+D))))/((A+C)/(A+B+C+D))$$

If "prevalence" argument is not `NULL` then the formula uses the value specified for prevalence of exposure (p):

$$CIN = 1/((p*(((A/(A+B))/(C/(C+D)))-1))/(p*(((A/(A+B))/(C/(C+D)))-1)+1))$$

Exposed Cases Impact Number (ECIN):

$$ECIN = 1/(1-(1/((A/(A+B))/(C/(C+D)))))$$

Value

A `tibble` with the following columns:

- **measure**: Name of the measure calculated
- **estimate**: Point estimate for the impact number
- **ci_lower**: The lower bound of the confidence interval for the estimate
- **ci_upper**: The upper bound of the confidence interval for the estimate
- **exposure**: Name of the exposure variable followed by +/- levels (e.g. smoking::yes/no)
- **outcome**: Name of the outcome variable followed by +/- levels (e.g. heart_disease::yes/no)

References

Hildebrandt, M., Bender, R., Gehrmann, U., & Blettner, M. (2006). Calculating confidence intervals for impact numbers. BMC medical research methodology, 6, 32. https://doi.org/10.1186/1471-2288-6-32

Heller, R. F., Dobson, A. J., Attia, J., & Page, J. (2002). Impact numbers: measures of risk factor impact on the whole population from case-control and cohort studies. Journal of epidemiology and community health, 56(8), 606–610. https://doi.org/10.1136/jech.56.8.606

`measures` *Measures of effect*

Description

The twoxtwo framework allows for estimation of the magnitude of association between an exposure and outcome. Measures of effect that can be calculated include odds ratio, risk ratio, and risk difference. Each measure can be calculated as a point estimate as well as the standard error (SE) around that value. It is critical to note that the interpretation of measures of effect depends on the study design and research question being investigated.

The `odds_ratio()`, `risk_ratio()`, and `risk_diff()` functions provide a standard interface for calculating measures of effect. Each function takes an input dataset and arguments for outcome and exposure as bare, unquoted variable names. If the input has the twoxtwo class then the effect measures will be calculated using exposure and outcome information from that object. The functions all return a tidy `tibble` with the name of the measure, the point estimate, and lower/upper bounds of a confidence interval (CI) based on the SE.

Formulas used in point estimate and SE calculations are available in 'Details'.

Usage

```
odds_ratio(.data, exposure, outcome, alpha = 0.05, ...)

risk_ratio(.data, exposure, outcome, alpha = 0.05, ...)

risk_diff(.data, exposure, outcome, alpha = 0.05, ...)
```

Arguments

`.data`	Either a data frame with observation-level exposure and outcome data or a twoxtwo object
`exposure`	Name of exposure variable; ignored if input to `.data` is a `twoxtwo` object
`outcome`	Name of outcome variable; ignored if input to `.data` is a `twoxtwo` object
`alpha`	Significance level to be used for constructing confidence interval; default is `0.05`
`...`	Additional arguments passed to twoxtwo function; ignored if input to `.data` is a `twoxtwo` object

Details

The formulas below denote cell values as A,B,C,D. For more on `twoxtwo` notation see the twoxtwo documentation.

Odds Ratio:

$$OR = (A * D)/(B * C)$$

$$seOR = sqrt(1/A + 1/B + 1/C + 1/D)$$

Risk Ratio:

$$RR = (A/(A + B))/(C/(C + D))$$

$$seRR = sqrt(((1-(A/(A+B)))/((A+B)*(A/(A+B))))+((1-(C/(C+D)))/((C+D)*(C/(C+D)))))$$

Risk Difference:

$$RD = (A/(A + B)) - (C/(C + D))$$

$$seRD = sqrt(((A * B)/((A + B)^3)) + ((C * D)/((C + D)^3)))$$

Value

A `tibble` with the following columns:

- **measure**: Name of the measure calculated
- **estimate**: Point estimate for the effect measure
- **ci_lower**: The lower bound of the confidence interval for the estimate
- **ci_upper**: The upper bound of the confidence interval for the estimate
- **exposure**: Name of the exposure variable followed by +/- levels (e.g. smoking::yes/no)
- **outcome**: Name of the outcome variable followed by +/- levels (e.g. heart_disease::yes/no)

References

Tripepi, G., Jager, K. J., Dekker, F. W., Wanner, C., & Zoccali, C. (2007). Measures of effect: relative risks, odds ratios, risk difference, and 'number needed to treat'. Kidney international, 72(7), 789–791. https://doi.org/10.1038/sj.ki.5002432

Walter S. D. (2000). Choice of effect measure for epidemiological data. Journal of clinical epidemiology, 53(9), 931–939. https://doi.org/10.1016/s0895-4356(00)00210-9

Szklo, M., & Nieto, F. J. (2007). Epidemiology: Beyond the basics. Sudbury, Massachussets: Jones and Bartlett.

Keyes, K.M, & Galea S. (2014). Epidemiology Matters: A new introduction to methodological foundations. New York, New York: Oxford University Press.

`print.twoxtwo` *Print twoxtwo object*

Description

The `print.twoxtwo()` function provides an S3 method for printing objects created with twoxtwo. The printed output formats the contents of the `twoxtwo` table as a kable.

Usage

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

Arguments

`x`	twoxtwo object
`...`	Additional arguments passed to kable

Value

A printed `knitr_kable` object with the `twoxtwo` cell counts, exposure levels as row names, and outcome levels as column names.

`summary.twoxtwo` *Summarize twoxtwo object*

Description

The `summary.twoxtwo()` function provides an S3 method for summarizing objects created with twoxtwo. The summary function prints the `twoxtwo` via print.twoxtwo along with characteristics of the contingency table such the number of missing observations and exposure/outcome variables and levels. The summary will also compute effect measures using odds_ratio, risk_ratio, and risk_diff and print the estimates and confidence interval for each.

Usage

```
## S3 method for class 'twoxtwo'
summary(object, alpha = 0.05, ...)
```

Arguments

`object`	twoxtwo object
`alpha`	Significance level to be used for constructing confidence interval; default is `0.05`
`...`	Additional arguments passed to print.twoxtwo

Value

Printed summary information including the outcome and exposure variables and levels, as well as the number of missing observations, the `twoxtwo` contingency table, and formatted effect measures (see "Description"). In addition to printed output, the function invisibly returns a named list with computed effect measures (i.e. the `tibble` outputs from odds_ratio, risk_ratio, and risk_diff respectively).

`titanic` *Expanded Titanic dataset*

Description

This data is based on the Titanic dataset. Unlike the version in the `datasets` package, the data here is expanded to the observation-level rather than cross-tabulated.

Usage

```
titanic
```

Format

A data frame with 2201 rows and 4 variables:

- **Class**: Passenger class ("1st", "2nd", "3rd") or crew status ("Crew")
- **Crew**: Logical as to whether or not a crew member (TRUE) or not (FALSE)
- **Sex**: Sex of individual ("Male" or "Female")
- **Age**: Categorized age ("Adult" or "Child")
- **Survived**: Whether or not individual survived ("Yes" or "No")

Examples

```
head(titanic)
```

`twoxtwo` *Create a twoxtwo table*

Description

The `twoxtwo` constructor function takes an input data frame and summarizes counts of the specified exposure and outcome variables as a two-by-two contingency table. This function is used internally in other functions, but can be used on its own as well. The returned object is given a `twoxtwo` class which allows dispatch of the `twoxtwo` S3 methods (see print.twoxtwo and summary.twoxtwo).

For more information on how the two-by-two table is created see ’Details’.

Usage

```
twoxtwo(.data, exposure, outcome, levels = NULL, na.rm = TRUE, retain = TRUE)
```

Arguments

`.data`	Data frame with observation-level exposure and outcome data
`exposure`	Name of exposure variable
`outcome`	Name of outcome variable
`levels`	Levels for the exposure and outcome as a named list; if supplied, then the contingency table will be oriented with respect to the sequence of levels specified; default is `NULL`
`na.rm`	Logical as to whether or not to remove `NA` values when constructing contingency table; default is `TRUE`
`retain`	Logical as to whether or not the original data passed to the ".data" argument should be retained; if `FALSE` the `summary.twoxtwo()` function will not compute effect measures; default is `TRUE`

Details

The two-by-two table covers four conditions that can be specified with A,B,C,D notation:

- **A**: Exposure "+" and Outcome "+"
- **B**: Exposure "+" and Outcome "-"
- **C**: Exposure "-" and Outcome "+"
- **D**: Exposure "-" and Outcome "-"

`twoxtwo()` requires that the exposure and outcome variables are binary. The columns can be character, numeric, or factor but must have only two levels. Each column will internally be coerced to a factor with levels reversed. The reversal results in exposures with `TRUE` and `FALSE` (or `1` and `0`) oriented in the two-by-two table with the `TRUE` as "+" (first row) and `FALSE` as "-" (second row). Likewise, `TRUE/FALSE` outcomes will be oriented with `TRUE` as "+" (first column) and `FALSE` as "-" (second column). Note that the user can also define the orientation of the table using the "levels" argument.

Value

A named list with the `twoxtwo` class. Elements include:

- **tbl**: The summarized two-by-two contingency table as a `tibble`.
- **cells**: Named list with the counts in each of the cells in the two-by-two contingency table (i.e. A,B,C,D)
- **exposure**: Named list of exposure information (name of variable and levels)
- **outcome**: Named list of outcome information (name of variable and levels)
- **n_missing**: The number of missing values (in either exposure or outcome variable) removed prior to computing counts for the two-by-two table
- **data**: The original data frame passed to the ".data" argument. If `retain=FALSE`, then this element will be `NULL`.

Index

∗ **datasets**
    titanic, 12

af, 2
arp *(*af*)*, 2

bound, 4

chisq, 4
chisq.test, *4*
cin *(*impact*)*, 7

display, 5

ecin *(*impact*)*, 7
ein *(*impact*)*, 7

fisher, 6
fisher.test, *6*
format_measure, 7

impact, 7

kable, *5*, *11*

measures, *2*, 9

odds_ratio, *7*, *12*
odds_ratio *(*measures*)*, 9

parp *(*af*)*, 2
print.twoxtwo, 11, *12, 13*

risk_diff, *12*
risk_diff *(*measures*)*, 9
risk_ratio, *12*
risk_ratio *(*measures*)*, 9

summary.twoxtwo, 12, *13*

Titanic, *12*
titanic, 12
twoxtwo, *2, 3*, *5–12*, 13
twoxtwo-package, 2