

Package ‘ANOPA’

July 18, 2025

Type Package

Title Analyses of Proportions using Anscombe Transform

Version 0.1.3

Date 2024-03-21

Author Denis Cousineau [aut, ctb, cre],
Louis Laurencelle [aut, ctb]

Maintainer Denis Cousineau <denis.cousineau@uottawa.ca>

BugReports <https://github.com/dcousin3/ANOPA/issues/>

URL <https://dcousin3.github.io/ANOPA/>

Description Analyses of Proportions can be performed on the Anscombe (arcsine-related) transformed data. The 'ANOPA' package can analyze proportions obtained from up to four factors. The factors can be within-subject or between-subject or a mix of within- and between-subject. The main, omnibus analysis can be followed by additive decompositions into interaction effects, main effects, simple effects, contrast effects, etc., mimicking precisely the logic of ANOVA. For that reason, we call this set of tools 'ANOPA' (Analysis of Proportion using Anscombe transform) to highlight its similarities with ANOVA. The 'ANOPA' framework also allows plots of proportions easy to obtain along with confidence intervals. Finally, effect sizes and planning statistical power are easily done under this framework. Only particularity, the 'ANOPA' computes F statistics which have an infinite degree of freedom on the denominator. See Laurencelle and Cousineau (2023) <[doi:10.3389/fpsyg.2022.1045436](https://doi.org/10.3389/fpsyg.2022.1045436)>.

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

LazyData true

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

Imports superb (>= 0.95.0), Rdpack (>= 0.7), ggplot2 (>= 3.1.0),
scales (>= 1.2.1), stats, rraply, utils, plyr (>= 1.8.4)

Suggests rmarkdown, testthat, knitr

RdMacros Rdpack

NeedsCompilation no

Repository CRAN

Date/Publication 2024-03-22 19:40:05 UTC

Contents

A	2
anopa	4
anopaN2Power	6
anopa_asn_trans1	8
ArringtonEtAl2002	10
ArticleExample1	12
ArticleExample2	13
ArticleExample3	14
contrastProportions	15
conversion	16
corrected	18
emProportions	18
explain	20
minimalExamples	21
posthocProportions	22
rBernoulli	24
summarize	26
uncorrected	26
unitaryAlpha	27
Index	28

A	<i>transformation functions</i>
---	---------------------------------

Description

The transformation functions 'A()' performs the Anscombe transformation on a pair {number of success; number of trials} = {s; n} (where the symbol ";" is to be read "over". The function 'varA()' returns the theoretical variance from the pair {s; n}. Both functions are central to the ANOPA (Laurencelle and Cousineau 2023). It was originally proposed by (Zubin 1935) and formalized by (Anscombe 1948).

Usage

```
A(s, n)

varA(s, n)

Atrans(v)

SE.Atrans(v)

var.Atrans(v)

CI.Atrans(v, gamma)

prop(v)

CI.prop(v, gamma)
```

Arguments

s	a number of success;
n	a number of trials.
v	a vector of 0s and 1s.
gamma	a confidence level, default to .95 when omitted.

Details

The functions `A()` and `varA()` take as input two integers, `s` the number of success and `n` the number of observations. The functions `Atrans()`, `SE.Atrans()`, `var.Atrans()`, `CI.Atrans()`, `prop()` and `CI.prop()` take as input a single vector `v` of 0s and 1s from which the number of success and the number of observations are derived.

Value

`A()` returns a score between 0 and 1.57 where a `s` of zero results in $A(0, n)$ tending to zero when the number of trials is large, and where the maximum occurs when `s` equals `n` and are both very large, so that for example $A(1000, 1000) = 1.55$. The midpoint is always 0.786 irrespective of the number of trials $A(0.5 * n, n) = 0.786$. The function `varA()` returns the theoretical variance of an Anscombe transformed score. It is exact as `n` gets large, and overestimate variance when `n` is small. Therefore, a test based on this transform is either exact or conservative.

References

Anscombe FJ (1948). "The transformatin of poisson, binormal and negative-binomial data." *Biometrika*, **35**, 246–254. doi:10.1093/biomet/35.34.246.

Laurencelle L, Cousineau D (2023). "Analysis of proportions using arcsine transform with any experimental design." *Frontiers in Psychology*, **13**, 1045436. doi:10.3389/fpsyg.2022.1045436.

Zubin J (1935). “Note on a transformation function for proportions and percentages.” *Journal of Applied Psychology*, **19**, 213–220. doi:10.1037/h0057566.

Examples

```
# The transformations from number of 1s and total number of observations:
A(5, 10)

varA(5, 10)

# Same with a vector of observations:
Atrans( c(1,1,1,1,1,0,0,0,0,0) )

var.Atrans( c(1,1,1,1,1,0,0,0,0,0) )
```

anopa

ANOPA: analysis of proportions using Anscombe transform.

Description

The function 'anopa()' performs an ANOPA for designs with up to 4 factors according to the 'ANOPA' framework. See Laurencelle and Cousineau (2023) for more.

Usage

```
anopa(formula = NULL, data = NULL, WSFactors = NULL)
```

Arguments

formula	A formula with the factors on the left-hand side. See below for writing the formula to match the data format.
data	Dataframe in one of wide, long, or compiled format;
WSFactors	For within-subject designs, provide the factor names and their number of levels. This is expressed as a vector of strings such as "Moment(2)".

Details

Note the following limitations:

1. The main analysis performed by anopa() is currently restricted to four factors in total (between and/or within). Contact the author if you plan to analyse more complex designs.
2. If you have repeated-measure design, the data *must* be provided in wide or long format. The correlation between successes cannot be assessed once the data are in a compiled format.
3. The data can be given in three formats:

- wide: In the wide format, there is one line for each participant, and one column for each between-subject factors in the design. In the column(s), the level of the factor is given (as a number, a string, or a factor). For within-subject factors, the columns contains 0 or 1 based on the status of the measurement.
- long: In the long format, there is an identifier column for each participant, a factor column and a level number for that factor. If there are n participants and m factors, there will be in total n x m lines.
- compiled: In the compiled format, there are as many lines as there are cells in the design. If there are two factors, with two levels each, there will be 4 lines.

See the vignette [DataFormatsForProportions](#) for more on data format and how to write their formula.

Value

An omnibus analyses of the given proportions. Each factor's significance is assessed, as well as their interactions when there is more than one factor. For decomposition of the main analyses, follow the analysis with `emProportions()`, `contrastProportions()`, or `posthocProportions()`

References

Laurencelle L, Cousineau D (2023). "Analysis of proportions using arcsine transform with any experimental design." *Frontiers in Psychology*, **13**, 1045436. doi:10.3389/fpsyg.2022.1045436.

Examples

```
# -- FIRST EXAMPLE --
# Basic example using a single between-subject factor design with the data in compiled format.
# Fictitious data present success (1) or failure (0) of the observation according
# to the state of residency (three levels: Florida, Kentucky or Montana) for
# 3 possible cells. There are 175 observations (with unequal n, Montana having only)
# 45 observations).
minimalBSExample
# The data are in compiled format, consequently the data frame has only three lines.
# The complete data frame in wide format would be composed of 175 lines, one per participant.

# The following formula using curly braces is describing this data format
# (note the semicolon to separate the number of successes from the number of observations):
formula <- {s; n} ~ state

# The analysis is performed using the function `anopa()` with a formula and data:
w <- anopa(formula, minimalBSExample)
summary(w)
# As seen, the proportions of success do not differ across states.

# To see the proportions when the data is in compiled format, simply divide the
# number of success (s) by the total number of observations (n):
minimalBSExample$s / minimalBSExample$n

# A plot of the proportions with error bars (default 95% confidence intervals) is
# easily obtained with
```

```

anopaPlot(w)

# The data can be re-formatted into different formats with,
# e.g., `toRaw()`, `toLong()`, `toWide()`
head(toWide(w))
# In this format, only 1s and 0s are shown, one participant per line.
# See the vignette `DataFormatsForFrequencies` for more.

# -- SECOND EXAMPLE --
# Real-data example using a three-factor design with the data in compiled format:
ArringtonEtAl2002

# This dataset, shown in compiled format, has three cells missing
# (e.g., fishes whose location is African, are Detrivore, feeding Nocturnally)
w <- anopa( {s;n} ~ Location * Trophism * Diel, ArringtonEtAl2002 )

# The function `anopa()` generates the missing cells with 0 success over 0 observations.
# Afterwards, cells with missing values are imputed based on the option:
getOption("ANOPA.zeros")
# where 0.05 is 1/20 of a success over one observations (arcsine transforms allows
# fractions of success; it remains to be studied what imputation strategy is best...)

# The analysis suggests a main effect of Trophism (type of food ingested)
# but the interaction Trophism by Diel (moment of feeding) is not to be neglected...
summary(w) # or summarize(w)

# The above presents both the uncorrected statistics as well as the corrected
# ones for small samples [w76]. You can obtain only the uncorrected...
uncorrected(w)

#... or the corrected ones
corrected(w)

# You can also ask easier outputs with:
explain(w) # human-readable ouptut NOT YET DONE

```

Description

The function `'anopaN2Power()'` performs an analysis of statistical power according to the 'ANOPA' framework. See Laurencelle and Cousineau (2023) for more. `'anopaPower2N()'` computes the sample size to reach a given power. Finally, `'anopaProp2fsq()'` computes the f^2 effect size from a set of proportions.

Usage

```
anopaPower2N(power, P, f2, alpha)
```

```
anopaN2Power(N, P, f2, alpha)
```

```
anopaProp2fsq(props, ns, unitaryAlpha, method="approximation")
```

Arguments

N	sample size;
P	number of groups;
f2	effect size Cohen's f^2 ;
alpha	(default if omitted .05) the decision threshold.
power	target power to attain;
ns	sample size per group;
props	a set of expected proportions (if all between 0 and 1) or number of success per group.
method	for computing effect size f^2 is 'approximation' or 'exact' only.
unitaryAlpha	for within-subject design, the measure of correlation across measurements.

Details

Note that for `anopaProp2fsq()`, the expected effect size f^2 depends weakly on the sample sizes. Indeed, the Anscombe transform can reach more extreme scores when the sample sizes are larger, influencing the expected effect size.

Value

`anopaPower2N()` returns a sample size to reach a given power level. `anopaN2Power()` returns statistical power from a given sample size. `anopaProp2fsq()` returns f^2 the effect size from a set of proportions and sample sizes.

References

Laurencelle L, Cousineau D (2023). "Analysis of frequency tables: The ANOVA framework." *The Quantitative Methods for Psychology*, **19**, 173–193. doi:10.20982/tqmp.19.2.p173.

Examples

```
# 1- Example of the article:
# with expected frequencies .34 to .16, assuming as a first guess groups of 25 observations:
f2 <- anopaProp2fsq( c( 0.32, 0.64, 0.40, 0.16), c(25,25,25,25) );
f2
# f-square is 0.128.

# f-square can be converted to eta-square with
eta2 <- f2 / (1 + f2)
```

```
# With a total sample of 97 observations over four groups,
# statistical power is quite satisfactory (85%).
anopaN2Power(97, 4, f2)

# 2- Power planning.
# Suppose we plan a four-classification design with expected proportions of:
pred <- c(.35, .25, .25, .15)
# P is the number of classes (here 4)
P <- length(pred)
# We compute the predicted f2 as per Eq. 5
f2 <- 2 * sum(pred * log(P * pred) )
# the result, 0.0822, is a moderate effect size.

# Finally, aiming for a power of 80%, we run
anopaPower2N(0.80, P, f2)
# to find that a little more than 132 participants are enough.
```

anopa_asn_trans1

anopaPlot: Easy plotting of proportions.

Description

The function 'anopaPlot()' performs a plot of proportions for designs with up to 4 factors according to the 'ANOPA' framework. See Laurencelle and Cousineau (2023) for more. The plot is realized using the 'suber' library; see Cousineau et al. (2021). It uses the arc-sine transformation 'A()'.

Usage

```
anopaPlot(w, formula, confidenceLevel = .95, allowImputing = FALSE,
showPlotOnly = TRUE, plotStyle = "line",
errorbarParams = list( width =0.5, linewidth=0.75 ), ...)
```

Arguments

w	An ANOPA object obtained with anopa();
formula	(optional) Use formula to plot just specific terms of the omnibus test. For example, if your analysis stored in w has factors A, B and C, then anopaPlot(w, ~ A * B) will only plot the factors A and B.
confidenceLevel	Provide the confidence level for the confidence intervals (default is 0.95, i.e., 95%).
allowImputing	(default FALSE) if there are cells with no observations, can they be imputed? If imputed, the option "ANOPA.zeros" will be used to determine how many additional observations to add, and with how many successes. If for example,

	the option is (by default) $c(0.05, 1)$, then 20 cases will be added, only one being a success (respecting the .05 target). <i>Keep in mind that imputations has never been studies with regards to proportions so be mindfull that the default optin has never been tested nor validated.</i>
showPlotOnly	(optional, default True) shows only the plot or else shows the numbers needed to make the plot yourself.
plotStyle	(optional; default "line") How to plot the proportions; see superb for other layouts (e.g., "line").
errorbarParams	(optional; default <code>list(width =0.5, linewidth=0.75)</code>) is a list of attributes used to plot the error bars. See superb for more.
...	Other directives sent to superb(), typically 'plotStyle', 'errorbarParams', etc.

Details

The plot shows the proportions on the vertical axis as a function of the factors (the first on the horizontal axis, the second if any in a legend; and if a third or even a fourth factors are present, as distinct rows and columns). It also shows 95% confidence intervals of the proportions, adjusted for between-cells comparisons. The confidence intervals are based on a z distribution, which is adequate for large samples (Chen 1990; Lehman and Loh 1990). This "stand-alone" confidence interval is then adjusted for between-cell comparisons using the *superb* framework (Cousineau et al. 2021).

See the vignette [DataFormatsForProportions](#) for more on data formats and how to write their formula. See the vignette [ConfidenceIntervals](#) for details on the adjustment and its purpose.

Value

a ggplot2 object of the given proportions.

References

Chen H (1990). "The accuracy of approximate intervals for a binomial parameter." *Journal of the American Statistical Association*, **85**, 514–518. doi:10.1080/01621459.1990.10476229.

Cousineau D, Goulet M, Harding B (2021). "Summary plots with adjusted error bars: The superb framework with an implementation in R." *Advances in Methods and Practices in Psychological Science*, **4**, 1–18. doi:10.1177/25152459211035109.

Laurencelle L, Cousineau D (2023). "Analysis of proportions using arcsine transform with any experimental design." *Frontiers in Psychology*, **13**, 1045436. doi:10.3389/fpsyg.2022.1045436.

Lehman EL, Loh W (1990). "Pointwise versus uniform robustness of some large-sample tests and confidence intervals." *Scandinavian Journal of Statistics*, **17**, 177–187.

Examples

```
#
# The Arrington Et Al., 2002, data on fishes' stomach
ArringtonEtAl2002
```

```

# This examine the omnibus analysis, that is, a 3 x 2 x 4 ANOPA:
w <- anopa( {s;n} ~ Location * Trophism * Diel, ArringtonEtAl2002)

# Once processed into w, we can ask for a standard plot
anopaPlot(w)

# As you may notice, there are points missing because the data have
# three missing cells. The litterature is not clear what should be
# done with missing cells. In this package, we propose to impute
# the missing cells based on the option `getOption("ANOPA.zeros")`.
# Consider this option with care.
anopaPlot(w, allowImputing = TRUE)

# We can place the factor `Diel` on the x-axis (first):
anopaPlot(w, ~ Diel * Trophism * Location )

# Change the style for a plot with bars instead of lines
anopaPlot(w, plotStyle = "bar")

# Changing the error bar style
anopaPlot(w, plotStyle = "bar", errorbarParams = list( width =0.1, linewidth=0.1 ) )

# Illustrating the main effect of Location (not interacting with other factors)
# and the interaction Diel * Trophism separately
anopaPlot(w, ~ Location )
anopaPlot(w, ~ Diel * Trophism )

# All these plots are ggplot2 so they can be followed with additional directives, e.g.
library(ggplot2)
anopaPlot(w, ~ Location) + ylim(0.0, 1.0) + theme_classic()
anopaPlot(w, ~ Diel * Trophism) + ylim(0.0, 1.0) + theme_classic()

# etc. Any ggplot2 directive can be added to customize the plot to your liking.
# See the vignette `ArringtonExample`.

```

ArringtonEtAl2002

Arrington et al. (2002) dataset

Description

The data, taken from Arrington et al. (2002), is a dataset examining the distribution of fishes with empty stomachs, classified over three factors: 'Collection location' (3 levels: Africa, Central/South America, North America), 'Diel feeding behavior' (2 levels: diurnal, nocturnal), 'Trophic category' (4 levels: Detritivore, Invertivore, Omnivore, Piscivore). It is therefore a $3 \times 2 \times 4$ design with 24 cells. The original data set also contains Order, Family and Species of the observed fishes and can be obtained from https://figshare.com/collections/HOW_OFTEN_DO_FISHES_RUN_ON_EMPTY_/3297635. It was commented in Warton and Hui (2011).

Usage

ArringtonEtAl2002

Format

A data frame.

Source

[doi:10.1890/00129658\(2002\)083\[2145:HODFRO\]2.0.CO;2](https://doi.org/10.1890/00129658(2002)083[2145:HODFRO]2.0.CO;2)

References

Arrington DA, Winemiller KO, Loftus WF, Akin S (2002). “How often do fishes “run on empty”?” *Ecology*, **83**(8), 2145–2151. [doi:10.1890/00129658\(2002\)083\[2145:HODFRO\]2.0.CO;2](https://doi.org/10.1890/00129658(2002)083[2145:HODFRO]2.0.CO;2).

Warton DI, Hui FK (2011). “The arcsine is asinine: The analysis of proportions in ecology.” *Ecology*, **92**, 3–10. [doi:10.1890/100340.1](https://doi.org/10.1890/100340.1).

Examples

```
# see the dataset
ArringtonEtAl2002

# The columns s and n indicate the number of fishes with
# empty stomachs (the "success") and the total number
# of fishes observed, respectively. Thus s/n is the proportion.

# run the ANOPA analysis
w <- anopa( {s; n} ~ Location * Diel * Trophism, ArringtonEtAl2002)

# make a plot with all the factors
anopaPlot(w)

# ... or with a subset of factors, with
anopaPlot(w, ~ Location * Trophism)

# Because of the three-way interaction, extract simple effects for each Diel
e <- emProportions( w, {s;n} ~ Location * Trophism | Diel )

# As the two-way simple interaction for Nocturnal * Diel is close to significant,
# we extract the second-order simple effects for each Diel and each Location
e <- emProportions(w, {s;n} ~ Trophism | Location * Diel )
# As seen, the Trophism is significant for Nocturnal fishes of
# Central/South America.
```

`ArticleExample1`*ArticleExample1*

Description

These are the data from the first example reported in (Laurencelle and Cousineau 2023). It shows fictitious data with regards to the proportion of incubation as a function of the distracting task. The design is a between-subject design with 4 groups.

Usage`ArticleExample1`**Format**

An object of class `data.frame`.

Source

[doi:10.20982/tqmp.19.2.p173](https://doi.org/10.20982/tqmp.19.2.p173)

References

Laurencelle L, Cousineau D (2023). “Analysis of proportions using arcsine transform with any experimental design.” *Frontiers in Psychology*, **13**, 1045436. [doi:10.3389/fpsyg.2022.1045436](https://doi.org/10.3389/fpsyg.2022.1045436).

Examples

```
library(ANOPA)

# the ArticleExample1 data shows an effect of the type of distracting task
ArticleExample1

# We perform an anopa on this dataset
w <- anopa( {nSuccess; nParticipants} ~ DistractingTask, ArticleExample1)

# We finish with post-hoc Tukey test
e <- posthocProportions( w )

# a small plot is *always* a good idea
anopaPlot(w)
```

ArticleExample2ArticleExample2

Description

These are the data from the second example reported in (Laurencelle and Cousineau 2023). It shows fictitious data with regards to the proportion of graduation for persons with dyslexia as a function of the moment of diagnostic (early or late) and the socio-economic status (SES). The design is a between-subject design with $2 \times 3 = 6$ groups.

Usage

```
ArticleExample2
```

Format

An object of class `data.frame`.

Source

[doi:10.20982/tqmp.19.2.p173](https://doi.org/10.20982/tqmp.19.2.p173)

References

Laurencelle L, Cousineau D (2023). “Analysis of proportions using arcsine transform with any experimental design.” *Frontiers in Psychology*, **13**, 1045436. [doi:10.3389/fpsyg.2022.1045436](https://doi.org/10.3389/fpsyg.2022.1045436).

Examples

```
library(ANOPA)

# the ArticleExample2 data shows an effect on the success to graduate as a function of
# socioeconomic status and moment of diagnostic:
ArticleExample2

# perform an anopa on this dataset
w <- anopa( {s;n} ~ MofDiagnostic * SES, ArticleExample2)

# a small plot is *always* a good idea
anopaPlot(w)
# here the plot is only for the main effect of SES.
anopaPlot(w, ~ SES)
```

`ArticleExample3`*ArticleExample3*

Description

These are the data from the third example reported in (Laurencelle and Cousineau 2023). It shows fictitious data with regards to the proportion of patients suffering delirium tremens as a function of the drug administered (cBau, eaPoe, R&V, Placebo). The design is a within-subject design with 4 measurements (order of administration randomized).

Usage

`ArticleExample3`

Format

An object of class `data.frame`.

Source

[doi:10.20982/tqmp.19.2.p173](https://doi.org/10.20982/tqmp.19.2.p173)

References

Laurencelle L, Cousineau D (2023). “Analysis of proportions using arcsine transform with any experimental design.” *Frontiers in Psychology*, **13**, 1045436. [doi:10.3389/fpsyg.2022.1045436](https://doi.org/10.3389/fpsyg.2022.1045436).

Examples

```
library(ANOPA)

# the ArticleExample3 data shows an effect of the drug administered on the
# proportion of participants who had an episode of delirium tremens
ArticleExample3

# perform an anopa on this dataset
w <- anopa( cbind(cBau,eaPoe,RnV,Placebo) ~ ., ArticleExample3, WSFactors = "Drug(4)")

# We finish with post-hoc Tukey test
e <- posthocProportions( w )

# a small plot is *always* a good idea
anopaPlot(w)
```

contrastProportions	<i>contrastProportion: analysis of contrasts between proportions using Anscombe transform.</i>
---------------------	--

Description

The function 'contrastProportions()' performs contrasts analyses on proportion data after an omnibus analysis has been obtained with 'anopa()' according to the ANOPA framework. See Laurencelle and Cousineau (2023) for more.

Usage

```
contrastProportions(w = NULL, contrasts = NULL)
```

Arguments

w	An ANOPA object obtained from anopa() or emProportions();
contrasts	A list that gives the weights for the contrasts to analyze. The contrasts within the list can be given names to distinguish them. The contrast weights must sum to zero and their cross-products must equal 0 as well.

Details

contrastProportions() computes the F -s for the contrasts, testing the hypothesis that it equals zero. The contrasts are each 1 degree of freedom, and the sum of the contrasts' degrees of freedom totalize the effect-being-decomposed's degree of freedom.

Value

A table of significance of the different contrasts.

References

Laurencelle L, Cousineau D (2023). "Analysis of proportions using arcsine transform with any experimental design." *Frontiers in Psychology*, **13**, 1045436. doi:10.3389/fpsyg.2022.1045436.

Examples

```
# Basic example using a one between-subject factor design with the data in compiled format.
# Fictitious data present success or failure of observation classified according
# to the state of residency (three levels); 175 participants have been observed in total.

# The cells are unequal:
minimalBSExample

# First, perform the omnibus analysis :
w <- anopa( {s;n} ~ state, minimalBSExample)
summary(w)
```

```
# Compare the first two states jointly to the third, and
# compare the first to the second state:
cw <- contrastProportions( w, list(
  contrast1 = c(1, 1, -2)/2,
  contrast2 = c(1, -1, 0) )
)
summary(cw)
```

conversion

Converting between formats

Description

The functions 'toWide()', 'toLong()', and 'toCompiled()' converts the data into various formats.

Usage

```
toWide(w)
```

```
toLong(w)
```

```
toCompiled(w)
```

Arguments

w An instance of an ANOPA object.

Details

The proportions of success of a set of n participants can be given using many formats. In what follows, n is the number of participants, p is the number of between-subject factor(s), q is the number of repeated-measure factor(s).

- One basic format, called wide, has one line per participants, with a 1 if a "success" is observed or a 0 if no success is observed. What a success is is entirely arbitrary. The proportion of success is then the number of 1s divided by the number of participants in each group. The data frame has n lines and $p+q$ columns.
- A second format, called long, has, on a line, the factor name(s) and 1s or 0s to indicate success or not. The data frame has $n \times q$ lines and 4 columns (a Id column to identify the participant; p columns to identify the groups, one column to identify which within-subject measure is given and finally, a 1 or 0 for the score of that measurement).

- A third format, called compiled, is to have a list of all the between-subject factors and the number of success and the total number of participants. This format is more compact as if there are 6 groups, the data are all contained in six lines (one line per group). This format however is only valid for between-subject design as we cannot infer the correlation between successes/failure.

See the vignette [DataFormatsForProportions](#) for more.

Value

A data frame in the requested format.

Examples

```
# The minimalBSExample contains $n$ of 175 participants categorized according
# to one factor $f = 1$, namely `State of residency` (with three levels)
# for 3 possible cells.
minimalBSExample

# Lets incorporate the data in an ANOPA data structure
w <- anopa( {s;n} ~ state, minimalBSExample )

# The data presented using various formats looks like
toWide(w)
# ... has 175 lines, one per participants ($n$) and 2 columns (state, success or failure)

toLong(w)
# ... has 175 lines ($n \times f$) and 4 columns (participant's `Id`, state name, measure name,
# and success or failure)

toCompiled(w)
# ... has 3 lines and 3 columns ($f$ + 2: number of succes and number of participants).

# This second example is from a mixed-design. It indicates the
# state of a machine, grouped in three categories (the sole between-subject
# factor) and at four different moments.
# The four measurements times are before treatment, post-treatment,
# 1 week later, and finally, 5 weeks later.
minimalMxExample

# Lets incorporate the data in an ANOPA data structure
w <- anopa( cbind(bpre,bpost,b1week,b5week) ~ Status,
            minimalMxExample,
            WSFactors = "Moment(4)" )

# -- Wide format --
# Wide format is actually the format of minimalMxExample
# (27 lines with 8 subjects in the first group and 9 in the second)
toWide(w)

# -- Long format --
# (27 times 4 lines = 108 lines, 4 columns, that is Id, group, measurement, success or failure)
```

```
toLong(w)

# -- Compiled format --
# (three lines as there are three groups, 7 columns, that is,
# the group, the 4 measurements, the number of participants, and the
# correlation between measurements for each group measured by unitary alphas)
toCompiled(w)
```

corrected	<i>corrected</i>
-----------	------------------

Description

'corrected()' provides an ANOPA table with only the corrected statistics.

Usage

```
corrected(object, ...)
```

Arguments

object	an object to explain
...	ignored

Value

An ANOPA table with the corrected test statistics.

emProportions	<i>emProportions: simple effect analysis of proportions.</i>
---------------	--

Description

The function 'emProportions()' performs a *simple effect* analyses of proportions after an omnibus analysis has been obtained with 'anopa()' according to the ANOPA framework. Alternatively, it is also called an *expected marginal* analysis of proportions. See Laurencelle and Cousineau (2023) for more.

Usage

```
emProportions(w, formula)
```

Arguments

w	An ANOPA object obtained from <code>anopa()</code> ;
formula	A formula which indicates what simple effect to analyze. Only one simple effect formula at a time can be analyzed. The formula is given using a vertical bar, e.g., " <code>~ factorA factorB</code> " to obtain the effect of Factor A within every level of the Factor B.

Details

`emProportions()` computes expected marginal proportions and analyzes the hypothesis of equal proportion. The sum of the `_F_s` of the simple effects are equal to the interaction and main effect `_F_s`, as this is an additive decomposition of the effects.

Value

An ANOPA table of the various simple main effects and if relevant, of the simple interaction effects.

References

Laurencelle L, Cousineau D (2023). "Analysis of frequency tables: The ANOVA framework." *The Quantitative Methods for Psychology*, **19**, 173–193. doi:10.20982/tqmp.19.2.p173.

Examples

```
# -- FIRST EXAMPLE --
# This is a basic example using a two-factors design with the factors between
# subjects. Fictitious data present the number of success according
# to Class (three levels) and Difficulty (two levels) for 6 possible cells
# and 72 observations in total (equal cell sizes of 12 participants in each group).
twoWayExample

# As seen the data are provided in a compiled format (one line per group).
# Performs the omnibus analysis first (mandatory):
w <- anopa( {success;total} ~ Difficulty * Class, twoWayExample)
summary(w)

# The results shows an important interaction. You can visualize the data
# using anopaPlot:
anopaPlot(w)
# The interaction is overadditive, with a small differences between Difficulty
# levels in the first class, but important differences between Difficulty for
# the last class.

# Let's execute the simple effect of Difficulty for every levels of Class
e <- emProportions(w, ~ Difficulty | Class )
summary(e)

# -- SECOND EXAMPLE --
# Example using the Arrington et al. (2002) data, a 3 x 4 x 2 design involving
# Location (3 levels), Trophism (4 levels) and Diel (2 levels), all between subject.
```

```

ArringtonEtAl2002

# first, we perform the omnibus analysis (mandatory):
w <- anopa( {s;n} ~ Location * Trophism * Diel, ArringtonEtAl2002)
summary(w)

# There is a near-significant interaction of Trophism * Diel (if we consider
# the unadjusted p value, but you really should consider the adjusted p value...).
# If you generate the plot of the four factors, we don't see much:
anopaPlot(w)

#... but a plot specifically of the interaction helps:
anopaPlot(w, ~ Trophism * Diel )
# it seems that the most important difference is for omnivorous fishes
# (keep in mind that there were missing cells that were imputed but there does not
# exist to our knowledge agreed-upon common practices on how to impute proportions...
# Are you looking for a thesis topic?).

# Let's analyse the simple effect of Trophism for every levels of Diel and Location
e <- emProportions(w, ~ Trophism | Diel )
summary(e)

# You can ask easier outputs with
corrected(w) # or summary(w) for the ANOPA table only
explain(w)   # human-readable output ((pending))

```

explain

explain

Description

'explain()' provides a human-readable, exhaustive, description of the results. It also provides references to the key results.

Usage

```
explain(object, ...)
```

Arguments

object	an object to explain
...	ignored

Value

a human-readable output with details of computations.

minimalExamples	<i>A collection of minimal Examples from various designs with one or two factors.</i>
-----------------	---

Description

The datasets present minimal examples that are analyzed with an Analysis of Frequency Data method (described in Laurencelle and Cousineau (2023)). The five datasets are

- 'minimalBSExample': an example with a single factor (state of residency)
- 'twoWayExample': an example with two factors, Class and Difficulty
- 'minimalWSExample': an example with a within-subject design (three measurements)
- 'twoWayWithinExample': an example with two within-subject factors
- 'minimalMxExample': a mixed design having one within and one between-subject factors

Usage

```
minimalBSExample
```

```
twoWayExample
```

```
minimalWSExample
```

```
twoWayWithinExample
```

```
minimalMxExample
```

Format

Objects of class `data.frame`:

An object of class `data.frame` with 6 rows and 4 columns.

An object of class `data.frame` with 19 rows and 3 columns.

An object of class `data.frame` with 30 rows and 6 columns.

An object of class `data.frame` with 27 rows and 5 columns.

References

Laurencelle L, Cousineau D (2023). "Analysis of proportions using arcsine transform with any experimental design." *Frontiers in Psychology*, **13**, 1045436. doi:10.3389/fpsyg.2022.1045436.

Examples

```
library(ANOPA)

# the twoWayExample data with proportions per Classes and Difficulty levels
twoWayExample

# perform an anopa on this dataset
w <- anopa( {success;total} ~ Difficulty * Class, twoWayExample)

# We analyse the proportions by Difficulty for each Class
e <- emProportions(w, ~ Difficulty | Class)
```

posthocProportions	<i>posthocProportions: post-hoc analysis of proportions.</i>
--------------------	--

Description

The function 'posthocProportions()' performs post-hoc analyses of proportions after an omnibus analysis has been obtained with 'anopa()' according to the ANOPA framework. It is based on the tukey HSD test. See Laurencelle and Cousineau (2023) for more.

Usage

```
posthocProportions(w, formula)
```

Arguments

w	An ANOPA object obtained from anopa();
formula	A formula which indicates what post-hocs to analyze. only one simple effect formula at a time can be analyzed. The formula is given using a vertical bar, e.g., " ~ factorA factorB " to obtain the effect of Factor A within every level of the Factor B.

Details

posthocProportions() computes expected marginal proportions and analyzes the hypothesis of equal proportion. The sum of the F 's of the simple effects are equal to the interaction and main effect F 's, as this is an additive decomposition of the effects.

Value

a model fit of the simple effect.

References

Laurencelle L, Cousineau D (2023). "Analysis of frequency tables: The ANOVA framework." *The Quantitative Methods for Psychology*, **19**, 173–193. doi:10.20982/tqmp.19.2.p173.

Examples

```
# -- FIRST EXAMPLE --
# This is a basic example using a two-factors design with the factors between
# subjects. Fictitious data present the number of success according
# to Class (three levels) and Difficulty (two levels) for 6 possible cells
# and 72 observations in total (equal cell sizes of 12 participants in each group).
twoWayExample

# As seen the data are provided in a compiled format (one line per group).
# Performs the omnibus analysis first (mandatory):
w <- anopa( {success;total} ~ Class * Difficulty, twoWayExample)
summary(w)

# The results shows an important interaction. You can visualize the data
# using anopaPlot:
anopaPlot(w)
# The interaction is overadditive, with a small differences between Difficulty
# levels in the first class, but important differences between Difficulty for
# the last class.

# Let's execute the post-hoc tests
e <- posthocProportions(w, ~ Difficulty | Class )
summary(e)

# -- SECOND EXAMPLE --
# Example using the Arrington et al. (2002) data, a 3 x 4 x 2 design involving
# Location (3 levels), Trophism (4 levels) and Diel (2 levels), all between subject.
ArringtonEtAl2002

# first, we perform the omnibus analysis (mandatory):
w <- anopa( {s;n} ~ Location * Trophism * Diel, ArringtonEtAl2002)
summary(w)

# There is a near-significant interaction of Trophism * Diel (if we consider
# the unadjusted p value, but you really should consider the adjusted p value...).
# If you generate the plot of the four factors, we don't see much:
# anopaPlot(w)
#... but with a plot specifically of the interaction helps:
anopaPlot(w, ~ Trophism * Diel )
# it seems that the most important difference is for omnivorous fishes
# (keep in mind that there were missing cells that were imputed but there does not
# exist to our knowledge agreed-upon common practices on how to impute proportions...
# Are you looking for a thesis topic?).

# Let's analyse the simple effect of Trophism for every levels of Diel and Location
e <- posthocProportions(w, ~ Trophism | Diel )
summary(e)

# You can ask easier outputs with
summarize(w) # or summary(w) for the ANOVA table only
```

```
corrected(w)  # or uncorrected(w) for an abbreviated ANOVA table
explain(w)    # for a human-readable output ((pending))
```

rBernoulli

Generating random proportions with GRP

Description

The function 'GRP()' generates random proportions based on a design, i.e., a list giving the factors and the categories with each factor. The data are returned in the 'wide' format.

Usage

```
GRP( props, n, BSDesign=NULL, WSDesign=NULL, sname = "s" )

rBernoulli(n, p)
```

Arguments

n	How many simulated participants are in each between-subject group (can be a vector, one per group);
p	a proportion of success;
BSDesign	A list with the between-subject factor(s) and the categories within each;
WSDesign	A list with the within-subject factor(s) and the categories within each;
props	(optional) the proportion of success in each cell of the design. Default 0.50;
sname	(optional) the column name that will contain the success/failure;

Details

The name of the function GRP() is derived from GRD(), a general-purpose tool to generate random data (Calderini and Harding 2019) now bundled in the superb package (Cousineau et al. 2021). GRP() is actually a proxy for GRD().

Value

GRP() returns a data frame containing success (coded as 1) or failure (coded as 0) for n participants per cells of the design. Note that correlated scores cannot be generated by GRP(); see (Lunn and Davies 1998). rBernoulli() returns a sequence of n success (1) or failures (0)

References

Calderini M, Harding B (2019). “GRD for R: An intuitive tool for generating random data in R.” *The Quantitative Methods for Psychology*, **15**(1), 1–11. doi:10.20982/tqmp.15.1.p001.

Cousineau D, Goulet M, Harding B (2021). “Summary plots with adjusted error bars: The superb framework with an implementation in R.” *Advances in Methods and Practices in Psychological Science*, **4**, 1–18. doi:10.1177/25152459211035109.

Lunn AD, Davies SJ (1998). “A note on generating correlated binary variables.” *Biometrika*, **85**(2), 487–490. doi:10.1093/biomet/85.2.487.

Examples

```
# The first example generate score for 20 participants in one factor having
# two categories (low and high):
design <- list( A=c("low","high"))
GRP( design, props = c(0.1, 0.9), n = 20 )

# This example has two factors, with factor A having levels a, b, c
# and factor B having 2 levels, for a total of 6 conditions;
# with 40 participants per group, it represents 240 observations:
design <- list( A=letters[1:3], B = c("low","high"))
GRP( design, props = c(0.1, 0.15, 0.20, 0.80, 0.85, 0.90), n = 40 )

# groups can be unequal:
design <- list( A=c("low","high"))
GRP( design, props = c(0.1, 0.9), n = c(5, 35) )

# Finally, repeated-measures can be generated
# but note that correlated scores cannot be generated with `GRP()`
wsDesign = list( Moment = c("pre", "post") )
GRP( WSDesign=wsDesign, props = c(0.1, 0.9), n = 10 )

# This last one has three factors, for a total of 3 x 2 x 2 = 12 cells
design <- list( A=letters[1:3], B = c("low","high"), C = c("cat","dog"))
GRP( design, n = 30, props = rep(0.5,12) )

# To specify unequal probabilities, use
design <- list( A=letters[1:3], B = c("low","high"))
expProp <- c(.05, .05, .35, .35, .10, .10 )
GRP( design, n = 30, props=expProp )

# The name of the column containing the proportions can be changed
GRP( design, n=30, props=expProp, sname="patate")

# Examples of use of rBernoulli
t <- rBernoulli(50, 0.1)
mean(t)
```

summarize

summarize

Description

'summarize()' provides the statistics table an ANOPAobject. It is synonym of 'summary()' (but as actions are verbs, I used a verb).

Usage

```
summarize(object, ...)
```

Arguments

object	an object to summarize
...	ignored

Value

an ANOPA table as per articles.

uncorrected

uncorrected

Description

'uncorrected()' provides an ANOPA table with only the uncorrected statistics.

Usage

```
uncorrected(object, ...)
```

Arguments

object	an object to explain
...	ignored

Value

An ANOPA table with the un-corrected test statistics. That should be avoided, more so if your sample is rather small.

unitaryAlpha	<i>unitary alpha</i>
--------------	----------------------

Description

The function 'unitaryAlpha()' computes the unitary alpha ((Laurencelle and Cousineau 2023)). This quantity is a novel way to compute correlation in a matrix where each column is a measure and each line, a subject. This measure is based on Cronbach's alpha (which could be labeled a 'global alpha').

Usage

```
unitaryAlpha( m )
```

Arguments

m	A data matrix for a group of observations.
---	--

Details

This measure is derived from Cronbach' measure of reliability as shown by Laurencelle and Cousineau (2023).

Value

A measure of correlation between -1 and +1.

References

Laurencelle L, Cousineau D (2023). "Analysis of proportions using arcsine transform with any experimental design." *Frontiers in Psychology*, **13**, 1045436. doi:10.3389/fpsyg.2022.1045436.

Examples

```
# Generate a random matrix (here binary entries)
set.seed(42)
N <- M <- 10
m <- matrix( runif(N*M), N, M)

# compute the unitary alpha from that random matrix
unitaryAlpha(m)
```

Index

* datasets

- ArringtonEtAl2002, [10](#)
- ArticleExample1, [12](#)
- ArticleExample2, [13](#)
- ArticleExample3, [14](#)
- minimalExamples, [21](#)

A, [2](#)

anopa, [4](#)

anopa_asn_trans1, [8](#)

anopaN2Power, [6](#)

anopaPlot (anopa_asn_trans1), [8](#)

anopaPower2N (anopaN2Power), [6](#)

anopaProp2fsq (anopaN2Power), [6](#)

ArringtonEtAl2002, [10](#)

ArticleExample1, [12](#)

ArticleExample2, [13](#)

ArticleExample3, [14](#)

Atrans (A), [2](#)

CI.Atrans (A), [2](#)

CI.prop (A), [2](#)

contrastProportions, [15](#)

conversion, [16](#)

corrected, [18](#)

emProportions, [18](#)

explain, [20](#)

GRP (rBernoulli), [24](#)

gSwitch (rBernoulli), [24](#)

minimalBSEExample (minimalExamples), [21](#)

minimalExamples, [21](#)

minimalMxExample (minimalExamples), [21](#)

minimalWSEExample (minimalExamples), [21](#)

posthocProportions, [22](#)

prop (A), [2](#)

rBernoulli, [24](#)

SE.Atrans (A), [2](#)

summarize, [26](#)

toCompiled (conversion), [16](#)

toLong (conversion), [16](#)

toWide (conversion), [16](#)

twoWayExample (minimalExamples), [21](#)

twoWayWithinExample (minimalExamples),
[21](#)

uncorrected, [26](#)

unitaryAlpha, [27](#)

var.Atrans (A), [2](#)

varA (A), [2](#)