Package: GAD (via r-universe)

August 31, 2024

Title Analysis of Variance from General Principles

Version 2.0

Depends R (>= 3.0)

Imports matrixStats

Description Analysis of complex ANOVA models with any combination of orthogonal/nested and fixed/random factors, as described by Underwood (1997). There are two restrictions: (i) data must be balanced; (ii) fixed nested factors are not allowed. Homogeneity of variances is checked using Cochran's C test and 'a posteriori' comparisons of means are done using Student-Newman-Keuls (SNK) procedure. For those terms with no denominator in the F-ratio calculation, pooled mean squares and quasi F-ratios are provided. Magnitute of effects are assessed by components of variation.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Leonardo Sandrini-Neto [aut, cre] (<`https://orcid.org/0000-0003-4086-6741`>), Eliandro Gilbert [aut], Maurício Camargo [aut] (<`https://orcid.org/0000-0002-8827-1176`>)

Maintainer Leonardo Sandrini-Neto <`leonardosandrini@ufpr.br`>

Repository CRAN

Date/Publication 2024-05-01 23:24:57 UTC

Contents

`as.fixed` *Encodes a vector as a "fixed factor"*

Description

Assigns a class `"fixed"` to a vector

Usage

```
as.fixed(x)
```

Arguments

`x`	a vector of data, usually a nominal variable.

Details

The function works the same way as `as.factor`, but assigns an additional class informing that it is a fixed factor.

Value

Function `as.fixed` returns an object of class `"factor"` and `"fixed"`.

Author(s)

Leonardo Sandrini-Neto (`<leonardosandrini@ufpr.br>`)

See Also

`as.random`

Examples

```
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
class(CG)
class(MQ)
```

`as.random` *Encodes a vector as a "random factor"*

Description

Assigns a class `"random"` to a vector

Usage

```
as.random(x)
```

Arguments

`x`	a vector of data, usually a nominal variable.

Details

The function works the same way as `as.factor`, but assigns an additional class informing that it is a random factor.

Value

Function `as.random` returns an object of class `"factor"` and `"random"`.

Author(s)

Leonardo Sandrini-Neto (`<leonardosandrini@ufpr.br>`)

See Also

`as.fixed`

Examples

```
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
class(CG)
class(MQ)
```

`C.test` *Cochran's C test of homogeneity of variances*

Description

Performs a Cochran's test of the null hypothesis that the largest variance in several sampled variances are the same.

Usage

```
C.test(object)
```

Arguments

`object`	an object of class "`lm`", containing the specified design.

Details

The test statistic is a ratio that relates the largest variance to the sum of the sampled variances.

Value

A list of class htest containing the following components:

`statistic`	Cochran's C test statistic
`p-value`	The p-value of the test
`alternative`	A character string describing the alternative hypothesis
`method`	The character string Cochran test of homogeneity of variances
`data.name`	A character string giving the name of the lm object
`estimate`	Sample estimates of variances

Author(s)

Leonardo Sandrini-Neto (<`leonardosandrini@ufpr.br`>)

See Also

`gad`

Examples

```
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
model <- lm(wing ~ CG + MQ%in%CG, data = rohlf95)
C.test(model)
```

`comp.var` *Components of variation*

Description

This function calculates components of variation for fixed and random factors.

Usage

```
comp.var(object, anova.tab = NULL)
```

Arguments

`object`	an object of class "`lm`", containing the specified design with random and/or fixed factors.
`anova.tab`	an object containing the results returned by `gad` or `pooling` functions. See examples below.

Details

This function calculates components of variation for any combination of orthogonal/nested and fixed/random factors. For pooled terms, `comp.var` function seeks for the denominator of the removed term and keep its type (fixed or random).

Note that there are differences on the interpretation of the results between fixed and random factors. For fixed factors, the components of variation calculated are just the sum of squares divided by the degrees of freedom and the hypothesis only concern those levels that were included in the model. On the other hand, for random factors the components of variation calculated are truly variance components and are interpretable as measures of variability of a population of levels, which are randomly sampled. However, for most of studies that aims to compare the amount of variation attributed to each term, the estimates of components of variation for both fixed and random factors are important and directly comparable (Anderson et al., 2008).

Eventually, the estimates of components of variation for some terms in the model can be negative. The term in question generally has a large p-value and its estimative is usually set to zero, since a negative estimate is illogical. An alternative to this problem is remove the term using `pooling` function and re-analyse the model (Fletcher and Underwood, 2002).

Value

A list of length 2, containing the mean squares estimates (`$mse`) and the table of components of variation (`$comp.var`) for the model.

The `$comp.var` table contains four columns: “Type” shows whether terms are fixed or random; “Estimate” shows the estimate of component of variation (in squared units); “Square root” shows the square root of the “Estimate” column (original unit); and “Percentage” shows the percentage of variability attributable to each term (both fixed and random factors).

Author(s)

Eliandro Gilbert (`<eliandrogilbert@gmail.com>`)

References

Anderson, M.J., Gorley, R.N., Clarke, K.R. 2008. *PERMANOVA+ for PRIMER*: Guide to Software and Statistical Methods. PRIMER-E: Plymouth, UK.

Fletcher, D.J., Underwood, A.J. 2002. How to cope with negative estimates of components of variance in ecological field studies. Journal of Experimental Marine Biology and Ecology 273, 89-95.

See Also

`gad`, `pooling`, `estimates`

Examples

```
library(GAD)
data(crabs)
head(crabs)

Re <- as.random(crabs$Region)   # random factor
Lo <- as.random(crabs$Location) # random factor nested in Region
Si <- as.random(crabs$Site)     # random factor nested in Location

model <- lm(Density ~ Re + Lo%in%Re + Si%in%Lo%in%Re, data = crabs)

C.test(model)                     # Checking homogeneity of variances
estimates(model)                  # Checking model structure
model.tab <- gad(model)           # Storing the result of ANOVA on a new object
model.tab                         # Checking ANOVA results
comp.var(model, anova.tab = model.tab) # Calculating the components of variations
```

`crabs` *Distribution patterns of the crab* Ucides cordatus *at different spatial scales*

Description

Distribution patterns of the mangrove crab *Ucides cordatus* were assessed at four levels of spatial hierarchy (from meters to tens of km) using a nested ANOVA and variance components measures. The design included four spatial scales of variation: regions (10s km), locations (km), sites (10s m), and quadrats (m). Three regions tens of kilometers apart were sampled across the Paranagua Bay salinity-energy gradient. Within each region, three locations (1.5 - 3.5 km apart from each other) were randomly chosen. In each location, three sites of 10 x 10 m were randomized, and within these, five quadrats of 2 x 2 m were sampled (Sandrini-Neto and Lana, 2012).

Usage

```
data(crabs)
```

Format

A data frame with 135 rows and 6 variables

Details

- Region: a random factor with three levels (`R1, R2, R3`)
- Location: a random factor with three levels (`L1, L2, L3`) nested in Region
- Site: a random factor with three levels (`S1, S2, S3`) nested in Location
- Quadrat: sample size
- Crabs: response variable; number of crabs per quadrat
- Density: response variable; number of crabs per square meter

References

Sandrini-Neto, L., Lana, P.C. 2012. Distribution patterns of the crab *Ucides cordatus* (Brachyura, Ucididae) at different spatial scales in subtropical mangroves of Paranagua Bay (southern Brazil). Helgoland Marine Research 66, 167-174.

Examples

```
library(GAD)
data(crabs)
crabs
```

`estimates` *Estimates of an analysis of variance design*

Description

This function is used to construct the mean squares estimates of an ANOVA design, considering the complications imposed by nested/orthogonal and fixed/random factors.

Usage

```
estimates(object, quasi.f = FALSE)
```

Arguments

`object`	an object of class "`lm`", containing the specified design with random and/or fixed factors.
`quasi.f`	logical, indicating whether to use quasi F-ratio when there is no single error term appropriate in the analysis. Default to `FALSE`.

Details

Determines what each mean square estimates in an ANOVA design by a set of procedures originally described by Cornfield and Tukey (1956). This version is a modification proposed by Underwood (1997), which does not allow for the use of fixed nested factors. The steps involve the construction of a table of multipliers with a row for each source of variation and a column for each term in the model that is not an interaction. The mean square estimates for each source of variation is obtained by determining which components belong to each mean square and what is their magnitude. This enables the recognition of appropriate F-ratios.

Value

A list containing the table of multipliers (`$tm`), the mean squares estimates (`$mse`) and the F-ratio versus (`$f.versus`) for the model.

Author(s)

Leonardo Sandrini-Neto (<`leonardosandrini@ufpr.br`>)

Eliandro Gilbert (<`eliandrogilbert@gmail.com`>)

References

Cornfield, J., Tukey, J.W. 1956. Average values of mean squares in factorials. Annals of Mathematical Statistics 27, 907-949.

Underwood, A.J. 1997. *Experiments in Ecology*: Their Logical Design and Interpretation Using Analysis of Variance. Cambridge University Press, Cambridge.

See Also

`gad`

Examples

```
# Example 1
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
model <- lm(wing ~ CG + MQ%in%CG, data = rohlf95)
estimates(model)

# Example 2
data(rats)
names(rats)
TR <- as.fixed(rats$treat)
RA <- as.random(rats$rat)
LI <- as.random(rats$liver)
model2 <- lm(glycog ~ TR + RA%in%TR + LI%in%RA%in%TR, data = rats)
estimates(model2)

# Example 3
```

```
data(snails)
O <- as.random(snails$origin)
S <- as.random(snails$shore)
B <- as.random(snails$boulder)
C <- as.random(snails$cage)
model3 <- lm(growth ~ O + S + O*S + B%in%S + O*(B%in%S) + C%in%(O*(B%in%S)), data = snails)
estimates(model3) # 'no test' for shore
estimates(model3, quasi.f = TRUE) # suitable test for shore
```

`gad` *General analysis of variance design*

Description

Fits a general ANOVA design with any combination of orthogonal/nested and fixed/random factors through function `estimates`.

Usage

```
gad(object, quasi.f = FALSE)
```

Arguments

`object`	an object of class lm, containing the specified design with random and/or fixed factors.
`quasi.f`	logical, indicating whether to use quasi F-ratio when there is no single error term appropriate in the analysis. Default to `FALSE`.

Details

Function gad returns an analysis of variance table using `estimates` to identify the appropriate F-ratios and consequently p-values for any complex model of orthogonal or nested, fixed or random factors as described by Underwood(1997).

Value

A "`list`" containing an object of class `"anova"` inheriting from class `"data.frame"`.

Author(s)

Leonardo Sandrini-Neto (`<leonardosandrini@ufpr.br>`)

References

Underwood, A.J. 1997. *Experiments in Ecology*: Their Logical Design and Interpretation Using Analysis of Variance. Cambridge University Press, Cambridge.

See Also

`estimates`

Examples

```
# Example 1
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
model <- lm(wing ~ CG + MQ%in%CG, data = rohlf95)
model.tab <- gad(model)
model.tab

# Example 2
data(rats)
TR <- as.fixed(rats$treat)
RA <- as.random(rats$rat)
LI <- as.random(rats$liver)
model2 <- lm(glycog ~ TR + RA%in%TR + LI%in%RA%in%TR, data = rats)
model2.tab <- gad(model2)
model2.tab

# Example 3
data(snails)
O <- as.random(snails$origin)
S <- as.random(snails$shore)
B <- as.random(snails$boulder)
C <- as.random(snails$cage)
model3 <- lm(growth ~ O + S + O*S + B%in%S + O*(B%in%S) + C%in%(O*(B%in%S)), data = snails)
model3.tab <- gad(model3)
model3.tab # 'no test' for shore
model3.tab2 <- gad(model3, quasi.f = TRUE)
model3.tab2 # suitable test for shore
```

`is.fixed` *Check if a factor is fixed*

Description

This function works the same way as `is.factor`.

Usage

```
is.fixed(x)
```

Arguments

x	a vector of data, usually a nominal variable encoded as a "`fixed factor`" using `as.fixed`.

Value

Function `is.fixed` returns `TRUE` or `FALSE` depending on whether its argument is a fixed factor or not.

Author(s)

Leonardo Sandrini-Neto (`<leonardosandrini@ufpr.br>`)

See Also

`is.random`, `as.fixed`

Examples

```
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
is.fixed(CG)
is.random(MQ)
```

`is.random` *Check if a factor is random*

Description

This function works the same way as `is.factor`.

Usage

```
is.random(x)
```

Arguments

`x`	a vector of data, usually a nominal variable encoded as a `"random factor"` using `as.random`.

Value

Function `is.random` returns `TRUE` or `FALSE` depending on whether its argument is a random factor or not.

Author(s)

Leonardo Sandrini-Neto (`<leonardosandrini@ufpr.br>`)

See Also

`is.fixed`, `as.random`

Examples

```
library(GAD)
data(rohlf95)
CG <- as.fixed(rohlf95$cages)
MQ <- as.random(rohlf95$mosquito)
is.fixed(CG)
is.random(MQ)
```

`pooling` *Post-hoc pooling*

Description

Performs a *post-hoc* pooling by combining or completely excluding terms from linear models

Usage

```
pooling(object, term = NULL, method = "pool", anova.tab = NULL)
```

Arguments

`object`	an object of class "`lm`", containing the specified design with random and/or fixed factors.
`term`	the term which will be removed from model.
`method`	method for removing a term from the model. Could be `method = "eliminate"` for completely exclude the term from model, or `method = "pool"` for pool the selected term with its appropriated F-ratio. Default to `method = "pool"`.
`anova.tab`	an object containing the results returned by `gad` or `pooling` functions. See examples below.

Details

Post-hoc pooling is a procedure to remove terms from a model. It might be done by several reasons: (i) lack of evidence against the null hypothesis of that term; (ii) a negative estimate of that term's component of variation (Fletcher and Underwood, 2002); (iii) the hypothesis of interest can not be tested until some terms are excluded from the model (Anderson et al., 2008). According to literature the term's p-value should exceed 0.25 before removing it (Underwood, 1997).

There are two different methods to remove a term from the model, determinated by `method` argument. When `method = "eliminate"` the chosen term is completely excluded from the model and its sum of squares and degrees of freedom are pooled with the residual sum of squares and degrees of freedom, as if the selected term had never been part of the model. When `method = "pool"` the chosen term's sum of squares and degrees of freedom are pooled with its denominator's sum of squares and degrees of freedom. The removal of terms using `method = "pool"` will be appropriated for most of situations (Anderson et al., 2008).

Note that removing a term has consequences for the construction of F-ratios (or quasi F-ratios), p-values and the estimation of components for the remaining terms, so should be done wisely. When there is more than one term which might be removed from the model (which p-value exceed 0.25), it is recommended to begin with the one having the smallest mean square (Anderson et. al, 2008).

Function `pooling` removes one term at once. After the removal of the term of interest, one should re-assess whether or not more terms should be removed. If it is the case, the output of `pooling` function should be stored in a new object and the function should be run again, using this new object in the `data` argument. This can be done successively. The way of `pooling` function does the analysis, step-by-step and storing the result in a new object at each step, gives the user total control of what happens and makes it easier return to the previous results.

Value

A list of length 4, containing the table of pooled terms (`$pool.table`), the mean squares estimates (`$mse`), the F-ratio versus (`$f.versus`) and the result of the analysis of variance (`$anova`).

Author(s)

Eliandro Gilbert (<`eliandrogilbert@gmail.com`>)

References

Anderson, M.J., Gorley, R.N., Clarke, K.R. 2008. *PERMANOVA+ for PRIMER*: Guide to Software and Statistical Methods. PRIMER-E: Plymouth, UK.

Fletcher, D.J., Underwood, A.J. 2002. How to cope with negative estimates of components of variance in ecological field studies. Journal of Experimental Marine Biology and Ecology 273, 89-95.

Underwood, A.J. 1997. *Experiments in Ecology*: Their Logical Design and Interpretation Using Analysis of Variance. Cambridge University Press, Cambridge.

See Also

`estimates`, `gad`, `comp.var`

Examples

```
library(GAD)
data(snails)
O <- as.random(snails$origin)   # a random factor
S <- as.random(snails$shore)    # a random factor orthogonal to origin
B <- as.random(snails$boulder)  # a random factor nested in shore
C <- as.random(snails$cage)    # a random factor nested in the combination of boulder and origin

model <- lm(growth ~ O + S + O*S + B%in%S + O*(B%in%S) + C%in%(O*(B%in%S)),data = snails)
estimates(model, quasi.f = FALSE)  # 'no test' for shore
gad(model, quasi.f = FALSE)        # no results for shore term
estimates(model, quasi.f = TRUE)   # suitable test for shore
gad(model, quasi.f = TRUE)         # test result for shore
```

```
# An alternative of using linear combinations of mean squares is the pooling function.
model.tab <- gad(model, quasi.f = FALSE) # stores the result of ANOVA on a new object
pooling(model, term = "S:B", method = "pool", anova.tab = model.tab)      # pooling terms
pooling(model, term = "S:B", method = "eliminate", anova.tab = model.tab) # or eliminating terms
```

`rats` *Glycogen content of rat livers*

Description

Duplicate readings were made on each of three preparations of rat livers from each of two rats for three different treatments (Sokal & Rohlf, 1995).

Usage

```
data(rats)
```

Format

A data frame with 36 rows and 4 variables

Details

- treat: a fixed factor with three levels
- rat: a random factor with two levels nested in treat
- liver: sample size
- glycog: response variable

References

Sokal, R.R., Rohlf, F.J. 1995. *Biometry*: The Principles and Practice of Statistics in Biological Research. 3rd Edition. W.H. Freeman and Co., New York.

Examples

```
library(GAD)
data(rats)
rats
```

`rohlf95` *Mosquitos' wing data colleted by Rohlf and cited in Sokal & Rohlf (1995)*

Description

Three different types of cage are tested on the growth of *Aedes intrudens*, a kind of mosquito pupae. In each one, four mosquitos are added and its wings measured twice. There are 24 observations (3 cages x 4 jars x 2 measures).

Usage

```
data(rohlf95)
```

Format

A data frame with 24 rows and 4 variables

Details

- cages: a fixed factor with three levels (`cage 1`, `cage2`, `cage3`)
- mosquito: a random factor with four levels (`m1`, `m2`, `m3`, `m4`) nested in cages
- measure: sample size
- wing: response variable

References

Sokal, R.R., Rohlf, F.J. 1995. *Biometry*: The Principles and Practice of Statistics in Biological Research. 3rd Edition. W.H. Freeman and Co., New York.

Examples

```
library(GAD)
data(rohlf95)
rohlf95
```

`snails` *Growth rates of snails on large boulders on different rock shores*

Description

This design was extracted from Underwood (1997), but data are artificial. Snails were transplanted from origin to different shores. Several boulders were used on each shore. Cages with snails of each origin on each boulder were replicated. All factors (origin, shore, boulder and cage) are random.

Usage

```
data(snails)
```

Format

A data frame with 240 rows and 6 variables

Details

- origin: a random factor with two levels (`O1, O2`)
- shore: a random factor with four levels (`S1, S2, S3, S4`) orthogonal to origin
- boulder: a random factor with three levels (`B1, B2, B3`) nested in shore
- cage: a random factor with two levels (`C1, C2`) nested in the combination of boulder and origin
- replicate: sample size
- growth: response variable

References

Underwood, A.J. 1997. *Experiments in Ecology*: Their Logical Design and Interpretation Using Analysis of Variance. Cambridge University Press, Cambridge.

Examples

```
library(GAD)
data(snails)
snails
```

`snk.test` *Student-Newman-Keuls (SNK) procedure*

Description

This function performs a SNK post-hoc test of means on the factors of a chosen term of the model, comparing among levels of one factor within each level of other factor or combination of factors.

Usage

```
snk.test(object, term = NULL, among = NULL, within = NULL, anova.tab = NULL)
```

Arguments

`object`	an object of class "`lm`" containing the specified design.
`term`	term of the model to be analysed. Argument `term` can be a main effect, an interaction or a nested factor, which must be specified using "quotes". Use `estimates` to see the right form to inform it. See examples below.
`among`	specifies the factor which levels will be compared among. Need to be specified if the term to be analysed envolves more than one factor.
`within`	specifies the factor or combination of factors that will be compared within level among.
`anova.tab`	an object containing the results returned by `gad` or `pooling` functions. See examples below.

Details

SNK is a stepwise procedure for hypothesis testing. First the sample means are sorted, then the pairwise studentized range (q) is calculated by dividing the differences between means by the standard error, which is based upon the average variance of the two samples.

Value

A list containing the standard error, the degrees of freedom and pairwise comparisons among levels of one factor within each level of other(s) factor(s).

Author(s)

Maurício Camargo (<`mauricio.camargo@furg.br`>)

Eliandro Gilbert (<`eliandrogilbert@gmail.com`>)

Leonardo Sandrini-Neto (<`leonardosandrini@ufpr.br`>)

See Also

`gad`, `estimates`

Examples

```
library(GAD)

# Example 1
data(rohlf95)
CG <- as.fixed(rohlf95$cages)     # a fixed factor
MQ <- as.random(rohlf95$mosquito) # a random factor nested in cages

model <- lm(wing ~ CG + CG%in%MQ, data = rohlf95)
model.tab <- gad(model) # storing ANOVA table in an object
model.tab               # checking ANOVA results
estimates(model)        # checking model structure

# Comparison among levels of mosquito ("MQ") within each level of cage ("CG")
snk.test(model, term = "CG:MQ", among = "CG", within = "MQ", anova.tab = model.tab)

# Example 2
data(snails)
O <- as.random(snails$origin)   # a random factor
S <- as.random(snails$shore)    # a random factor orthogonal to origin
B <- as.random(snails$boulder)  # a random factor nested in shore
C <- as.random(snails$cage)    # a random factor nested in the combination of boulder and origin
model2 <- lm(growth ~ O + S + O*S + B%in%S + O*(B%in%S) + C%in%(O*(B%in%S)), data = snails)
model2.tab <- gad(model2, quasi.f = FALSE) # storing ANOVA table in an object
model2.tab                                 # checking ANOVA results
estimates(model2, quasi.f = FALSE)         # checking model structure

# Comparison among levels of "origin"
snk.test(model2, term = "O", anova.tab = model2.tab)
# Comparison among levels of "shore" within each level of "origin"
snk.test(model2, term = "O:S", among = "S", within = "O", anova.tab = model2.tab)
# If term "O:S:B" were significant, we could try
snk.test(model2, term = "O:S:B", among = "B", within = "O:S", anova.tab = model2.tab)
```

Index