Package: iRepro (via r-universe)

September 8, 2024

Type Package

Title Reproducibility for Interval-Censored Data

Version 1.2

Date 2023-07-13

Author Jelena Kovacic

Maintainer Jelena Kovacic <jkovacic@imi.hr>

Description Calculates intraclass correlation coefficient (ICC) for
assessing reproducibility of interval-censored data with two
repeated measurements (Kovacic and Varnai (2014)
<doi:10.1097/EDE.0000000000000139>). ICC is estimated by
maximum likelihood from model with one fixed and one random
effect (both intercepts). Help in model checking (normality of
subjects' means and residuals) is provided.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-08 10:30:10 UTC

Contents

---

iRepro-package   *Reproducibility for Interval-Censored Data*

Description

Calculates intraclass correlation coefficient (ICC) for assessing reproducibility of interval-censored data with two repeated measurements (Kovacic and Varnai (2014) <doi:10.1097/EDE.0000000000000139>). ICC is estimated by maximum likelihood from model with one fixed and one random effect (both intercepts). Help in model checking (normality of subjects' means and residuals) is provided.

Details

Package:	iRepro
Type:	Package
Version:	1.2
Date:	2023-07-13
License:	GPL-3

Author(s)

Jelena Kovacic

Maintainer: Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

Examples

```
# Data generation (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,100,replace=TRUE) # first measurement
r2 <- sample(classes,100,replace=TRUE) # second measurement

summary(intervalICC(r1,r2,predefined.classes=TRUE,classes,class.limits)) # ICC estimation
```

| intervalICC | *Intraclass Correlation Coefficient for Interval-Censored Data* |

Description

The function calculates intraclass correlation coefficient (ICC) for interval-censored data with two repeated measurements. ICC is estimated by maximum likelihood from model with one fixed and one random effect (both intercepts).

Usage

```
intervalICC(r1, r2, predefined.classes=FALSE, classes, c.limits, optim.method=1)
```

Arguments

r1
: data corresponding to the first measurement. If `predefined.classes=TRUE` (appropriate for grouped data), this is a vector of length n, where each observation is one of the labels given in `classes`. Otherwise, if `predefined.classes=FALSE`, r1 is a matrix or a data frame with n rows and 2 columns, with columns representing lower and upper bounds of censoring intervals (e.g., if i-th observation lies in the interval $[a, b]$, then `r1[i,]=c(a,b)`).

r2
: data corresponding to the second measurement. If `predefined.classes=TRUE` (appropriate for grouped data), this is a vector of length n, where each observation is one of the labels given in `classes`. Otherwise, if `predefined.classes=FALSE`, r2 is a matrix or a data frame with n rows and 2 columns, with columns representing lower and upper bounds of censoring intervals (e.g., if i-th observation lies in the interval $[a, b]$, then `r2[i,]=c(a,b)`).

predefined.classes
: logical, indicating whether observations belong to predefined classes (e.g. grouped data in questionnaires) or each observation has its own lower and upper limit (default; FALSE).

classes
: a vector with unique labels for the *k* predefined classes. Required if `predefined.classes=TRUE`.

c.limits
: a matrix or a data frame with *k* rows and 2 columns, corresponding to lower and upper bounds of censoring intervals for classes. Required if `predefined.classes=TRUE`.

optim.method
: an integer (1 or 2) specifying the optimization method to be used in maximum likelihood estimation (default is 1). Details are given below.

Details

ICC is estimated by maximum likelihood from random effects model

$$Y_{ij} = \mu + b_i + e_{ij},$$

where b_i and e_{ij} are independent and normally distributed with means 0 and variances σ_b^2 and σ^2, respectively. If data were uncensored, this would be analogous to

```
lme(ratings~1, random=~1|id, method="ML", data=observed)
```

in `nlme` package, where

`observed=as.data.frame(rbind(cbind(r1,1:n), cbind(r2,1:n)))`

and `colnames(observed)=c("ratings","id")`. To maximize log-likelihood, `constrOptim` from `stats` package is used (method=BFGS).

Two available optimization methods, specified by `optim.method`, correspond to two mathematically equivalent expressions for log-likelihood. The option `optim.method=1` resulted in slightly more accurate estimates in simulations with grouped data, but `optim.method=2` was more numerically stable. See the reference for more details.

Value

An object of class "ICCfit". The object is a list with the components:

`icc`	maximum likelihood estimate (MLE) of ICC
`sigma2.b`	MLE of between-class variance σ_b^2
`sigma2.w`	MLE of within-class variance σ^2
`mu`	MLE of mean μ
`loglikelihood`	log-likelihood evaluated at MLE parameters

Note

If there are many observations with same values (i.e. with the same lower and upper bounds), it is advisable to group all observations into `classes` and use option `predefined.classes=TRUE`; this will reduce computation time.

Subjects with only one measurement are omitted from ICC calculation.

Author(s)

Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

See Also

summary.ICCfit

Examples

```
# Example with 6 predefined classes (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,30,replace=TRUE)
r2 <- sample(classes,30,replace=TRUE)

intervalICC(r1,r2,predefined.classes=TRUE,classes,class.limits)
```

```
# The same result can be obtained with predefined.classes=FALSE option,
# although with slower computation time
rtg1 <- matrix(nrow=30,ncol=2)
rtg2 <- matrix(nrow=30,ncol=2)
# when predefined.classes=FALSE, ratings must be given with lower and upper bounds
# for each observation:
for(i in 1:length(classes)){
  rtg1[r1==classes[i],1] <- class.limits[i,1]
  rtg1[r1==classes[i],2] <- class.limits[i,2]
  rtg2[r2==classes[i],1] <- class.limits[i,1]
  rtg2[r2==classes[i],2] <- class.limits[i,2]
}

intervalICC(rtg1,rtg2,predefined.classes=FALSE)
```

---

ntest.means	*Normality Check for Interval-Censored Data with Repeated Measurements - Means*

Description

The function checks whether interval-censored data with two repeated measurements meet the normality assumption for subjects' means. This is a prerequisite for the random effects model used in ICC calculation.

Usage

```
ntest.means(r1, r2, predefined.classes=FALSE, classes, c.limits, optim.method=1, bins=10)
```

Arguments

r1	argument passed to intervalICC; see documentation for that function.
r2	argument passed to intervalICC; see documentation for that function.
predefined.classes	
	argument passed to intervalICC; see documentation for that function.
classes	argument passed to intervalICC; see documentation for that function.
c.limits	argument passed to intervalICC; see documentation for that function.
optim.method	argument passed to intervalICC; see documentation for that function.
bins	number of categories in chi-square test; see details below (default is 10).

Details

For ICC estimation the random effects data model

$$Y_{ij} = \mu + b_i + e_{ij},$$

is used, where b_i and e_{ij} are normally distributed with means 0 and variances σ_b^2 and σ^2, respectively. This function assesses the assumption that the subjects' means $0.5(Y_{i1} + Y_{i2})$ are normally distributed with mean μ and variance $\sigma_b^2 + 0.5\sigma^2$, as is expected under the specified model.

To test normality, chi-square goodness-of-fit test with bins subsequent data categories is used (call to chisq.test from package stats). The categories (bins) are determined using the equidistant quantiles of expected normal distribution, with corresponding maximum likelihood parameters. Maximum likelihood estimates for parameters μ, σ_b^2 and σ^2 are obtained by calling the function intervalICC. The probability corresponding to each bin is 1/bins (expected relative frequencies; this corresponds to p = rep(1/bins,bins) in chisq.test function). Since means are interval-censored and censoring intervals overlap, the observed relative frequencies are calculated in the following way. If one of the original intervals representing subjects mean spans multiple bins, each bin receives a share of votes from the original interval. This share is calculated using the expected normal density function and it is proportional to the probability of data falling within the intersection of the original interval and bin.

Value

An object of class "ntestMeans". The object is a list with the components:

statistic	value of chi-squared statistic; statistic in the output of chisq.test
parameter	number of degrees of freedom for chi-squared distribution; parameter in the output of chisq.test
p.value	p-value of test; p.value in the output of chisq.test
data	character string with value "means"
mu	mean of the expected normal distribution for subjects' means; equal to maximum likelihood estimate for μ from intervalICC
var	variance of the expected normal distribution for subjects' means; equal to maximum likelihood estimate for $\sigma_b^2 + 0.5\sigma^2$ from intervalICC
bins	number of categories in chi-square test

Note

This function was designed as a help in assessing goodness of model fit. However, it has not been tested in simulations nor in any other way. It is the responsibility of the user to provide appropriate number of bins; the function checks only if bins is a positive integer. Testing normality with low number of bins is unreliable. On the other hand, if the number of bins is too large, chisq.test will complain since the expected frequencies will be too low.

Author(s)

Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

See Also

summary.ntestMeans, intervalICC, chisq.test

Examples

```
# Example with 6 predefined classes (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,30,replace=TRUE)
r2 <- sample(classes,30,replace=TRUE)
ntest.means(r1,r2,predefined.classes=TRUE,classes,class.limits,bins=10)
```

---

ntest.res	*Normality Check for Interval-Censored Data with Repeated Measurements - Residuals*

Description

The function checks whether interval-censored data with two repeated measurements meet the normality assumption for subjects' residuals. This is a prerequisite for the random effects model used in ICC calculation.

Usage

```
ntest.res(r1, r2, predefined.classes=FALSE, classes, c.limits, optim.method=1, bins=10)
```

Arguments

r1	argument passed to intervalICC; see documentation for that function.
r2	argument passed to intervalICC; see documentation for that function.
predefined.classes	
	argument passed to intervalICC; see documentation for that function.
classes	argument passed to intervalICC; see documentation for that function.
c.limits	argument passed to intervalICC; see documentation for that function.
optim.method	argument passed to intervalICC; see documentation for that function.
bins	number of categories in chi-square test; see details below (default is 10).

Details

For ICC estimation the random effects data model

$$Y_{ij} = \mu + b_i + e_{ij},$$

is used, where b_i and e_{ij} are normally distributed with means 0 and variances σ_b^2 and σ^2, respectively. This function assesses the assumption that the subjects' "residuals" $Y_{i1} - 0.5(Y_{i1} + Y_{i2})$ and $Y_{i2} - 0.5(Y_{i1} + Y_{i2})$ are normally distributed with mean 0 and variance $0.5\sigma^2$, as is expected under the specified model.

To test normality, chi-square goodness-of-fit test with `bins` subsequent data categories is used (call to `chisq.test` from package `stats`). The categories (bins) are determined using the equidistant quantiles of expected normal distribution, with corresponding maximum likelihood parameters. Maximum likelihood estimates for parameters μ, σ_b^2 and σ^2 are obtained by calling the function [intervalICC](#). The probability corresponding to each bin is 1/bins (expected relative frequencies; this corresponds to `p = rep(1/bins,bins)` in `chisq.test` function). Since residuals are interval-censored and censoring intervals overlap, the observed relative frequencies are calculated in the following way. If one of the original intervals representing subjects residual spans multiple bins, each bin receives a share of votes from the original interval. This share is calculated using the expected normal density function and it is proportional to the probability of data falling within the intersection of the original interval and bin.

Residuals for the first time point $(Y_{i1} - 0.5(Y_{i1} + Y_{i2}))$ and residuals for the second $(Y_{i2} - 0.5(Y_{i1} + Y_{i2}))$ are tested separately; therefore two test results in the output are given.

Value

An object of class "ntestRes". The object is a list with the components:

statistic.res1	value of chi-squared statistic corresponding to the first residual; `statistic` in the output of `chisq.test`
p.value.res1	p-value of test corresponding to the first residual; `p.value` in the output of `chisq.test`
statistic.res2	value of chi-squared statistic corresponding to the second residual; `statistic` in the output of `chisq.test`
p.value.res2	p-value of test corresponding to the second residual; `p.value` in the output of `chisq.test`
parameter	number of degrees of freedom for chi-squared distribution (the same for both residuals); `parameter` in the output of `chisq.test`
data	character string with value „residuals"
mu	mean of the expected normal distribution for subjects' residuals; equal to 0
var	variance of the expected normal distribution for subjects' residuals; equal to maximum likelihood estimate for $0.5\sigma^2$ from `intervalICC`
bins	number of categories in chi-square test

Note

This function was designed as a help in assessing goodness of model fit. However, it has not been tested in simulations nor in any other way. It is the responsibility of the user to provide appropriate number of bins; the function checks only if `bins` is a positive integer. Testing normality with low number of bins is unreliable. On the other hand, if the number of bins is too large, `chisq.test` will complain since the expected frequencies will be too low.

Author(s)

Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

See Also

[summary.ntestRes](#), [intervalICC](#), [chisq.test](#)

Examples

```
# Example with 6 predefined classes (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,30,replace=TRUE)
r2 <- sample(classes,30,replace=TRUE)
ntest.res(r1,r2,predefined.classes=TRUE,classes,class.limits,bins=10)
```

---

summary.ICCfit   *Summary for ICCfit Objects*

Description

The function summarizes the results of ICC estimation.

Usage

```
## S3 method for class 'ICCfit'
summary(object, ...)
```

Arguments

object	object of the class ICCfit (output of the [intervalICC](#) function)
...	additional arguments passed to the function (they do not affect the summary produced)

Details

For more details about ICC estimation and output values shortly described below, please refer to the documentation for [intervalICC](#).

Value

An object of class "summary.ICCfit". The object is a list with the components:

estimates	a data frame containing maximum likelihood estimates for ICC, mean and variance components
loglikelihood	log-likelihood evaluated at maximum likelihood estimates

Author(s)

Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

See Also

[intervalICC](#)

Examples

```
# Example with 6 predefined classes (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,30,replace=TRUE)
r2 <- sample(classes,30,replace=TRUE)
icc.est <- intervalICC(r1,r2,predefined.classes=TRUE,classes,class.limits)
summary(icc.est)
```

---

summary.ntestMeans   *Summary for ntestMeans Objects*

Description

The function summarizes the results of normality check for means.

Usage

```
## S3 method for class 'ntestMeans'
summary(object, ...)
```

Arguments

object	object of the class ntestMeans (output of the [ntest.means](#) function)
...	additional arguments passed to the function (they do not affect the summary produced)

Details

For more details about normality check and output values shortly described below, please refer to the documentation for [ntest.means](#).

Value

An object of class "summary.ntestMeans". The object is a list with the components:

test.res	a data frame containing the chi-squared statistic and p-value for normality test
mu	mean of the expected normal distribution for means
stdev	standard deviation of the expected normal distribution for means
bins	number of categories in chi-squared normality test
df	number of degrees of freedom in chi-squared normality test

Author(s)

Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

See Also

[ntest.means](#)

Examples

```
# Example with 6 predefined classes (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,30,replace=TRUE)
r2 <- sample(classes,30,replace=TRUE)
nm <- ntest.means(r1,r2,predefined.classes=TRUE,classes,class.limits,bins=10)
summary(nm)
```

---

summary.ntestRes	*Summary for ntestRes Objects*

Description

The function summarizes the results of normality check for residuals.

Usage

```
## S3 method for class 'ntestRes'
summary(object, ...)
```

Arguments

`object`	object of the class ntestRes (output of the ntest.res function)
`...`	additional arguments passed to the function (they do not affect the summary produced)

Details

For more details about normality check and output values shortly described below, please refer to the documentation for ntest.res.

Value

An object of class "summary.ntestRes". The object is a list with the components:

`test.res`	a data frame containing the chi-squared statistics and p-values for normality tests
`mu`	mean of the expected normal distribution for residuals
`stdev`	standard deviation of the expected normal distribution for residuals
`bins`	number of categories in chi-squared normality test
`df`	number of degrees of freedom in chi-squared normality test

Author(s)

Jelena Kovacic <jkovacic@imi.hr>

References

Kovacic J, Varnai VM. Intraclass correlation coefficient for grouped data. *Epidemiology* 2014;25(5):769–770.

See Also

ntest.res

Examples

```
# Example with 6 predefined classes (grouped data)
classes <- 1:6
class.limits <- cbind(classes-0.5,classes+0.5)
r1 <- sample(classes,30,replace=TRUE)
r2 <- sample(classes,30,replace=TRUE)
nr <- ntest.res(r1,r2,predefined.classes=TRUE,classes,class.limits,bins=10)
summary(nr)
```

Index