

Package ‘specificity’

July 2, 2014

Type Package

Title Specificity of personality trait-outcome (or trait-trait) associations

Version 0.1.1

Date 2013-08-19

Author Kenn Konstabel, Rene Mottus

Maintainer Rene Mottus <rene.mottus@ed.ac.uk>

Depends car

Description The package helps to test the specificity of personality trait-outcome (or trait-trait) associations by comparing the observed associations to those obtained using randomly created personality scales.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-08-21 07:01:59

R topics documented:

eta2	2
randomCorrelations	3
specificityLm, specificityEta2, specificityGlm	5

Index	9
--------------	----------

eta2	<i>Partial eta-squares</i>
------	----------------------------

Description

Obtain partial eta-squares from an `lm` object.

Usage

```
eta2(x)
```

Arguments

`x` An object of class `lm`.

Details

Based on type III sums of squares.

Value

A data frame containing partial eta-squares and p-values.

Author(s)

Rene Mottus <rene.mottus@ed.ac.uk>

See Also

[specificity](#)

Examples

```
x = rnorm(1:100)
y = as.factor(rep(1:2, each=50))
a = lm(x ~ y)
eta2(a)
```

randomCorrelations *Correlations between randomly generated scales*

Description

In social sciences, everything tends to correlate with everything, possibly due to theoretically uninteresting reasons. In questionnaire data, this general tendency of everything being correlated with everything is supplemented by the spurious intercorrelations resulting from all sorts of common method artefacts. This means that the null hypotheses (e.g., $r = 0$) is almost never correct, even when there are no meaningful and substantively interpretable associations between variables and this is probably especially true for questionnaire-based research. Therefore, the correct baseline hypotheses against which researchers can compare their associations of interest is often not the null-hypotheses. One way to guesstimate the true baseline hypotheses is to calculate associations between randomly generated variables; in questionnaires such variables can be created by randomly aggregating items into scales. This function is suitable when associations between two randomly generated scales might provide a proper baseline hypothesis against which substantive hypotheses can be tested (the `specificity` function is suitable for guesstimating associations of random scales with external phenomena, which are not based on questionnaires).

Usage

```
randomCorrelations(Data1, Data2 = NULL, n.items, R = 1000,  
complete.overlap = FALSE, item.overlap = FALSE, trait.overlap = TRUE,  
Key1 = NULL, Key2 = NULL)
```

Arguments

<code>Data1</code>	The data.frame from which random scales are drawn.
<code>Data2</code>	An optional data.frame. Used when the random scales are drawn from independent data (e.g., self-ratings and informant-ratings, measurements at two different time-points, or parallel forms of a questionnaire).
<code>n.items</code>	The number of items in random scales.
<code>R</code>	The number of random scale correlations to be calculated.
<code>complete.overlap</code>	Logical. If TRUE, there will be complete overlap in the items included in the two random scales (overrides other "overlap" arguments).
<code>item.overlap</code>	Logical. If TRUE, item overlap between random scales is allowed (but not granted). If FALSE, item overlap between random scales is precluded.
<code>trait.overlap</code>	Logical. If TRUE, two random scales to be correlated may have items from the same trait. If FALSE the two random scales are formed from the items belonging to different traits (the pool of traits is randomly split into two subpools and items from the two random scales are drawn from the two independent subpools). Requires the scoring key (<code>Key1</code>) to be provided.

Key1	The scoring key, indicating the trait-belonging of the items provided in Data1 (and also in Data2 if this was provided and Key2 was not provided). Its length equals the number of items in Data1 (and also in Data2 if this was provided and Key2 was not provided) and each number corresponds to one trait: for example <code>scale=c(1,2,3,4,5,1,2,3,4,5)</code> corresponds to a 10-item questionnaire that measures five traits, with items being ordered as shown).
Key2	Another scoring key, indicating the trait-belonging of the items provided in Data2. Only necessary when the scoring keys differ for Data1 and Data2 (e.g., these are different questionnaires measuring the same traits). Note that Key1 and Key2 should contain the same elements (i.e., the datasets should contain items of the same traits), otherwise trait overlaps should be allowed and no scoring keys should be passed to the function.

Details

This function can be used to guesstimate random associations between variables of the same dataset. Likewise, and perhaps more interestingly, it can also be used to guesstimate random associations between different datasets. For example, the unspecific associations (that are not bound to any substantive trait) between self-reported and informant-reported variables can be estimated, or between data from two different time-points or from parallel questionnaires. Overlaps in item content can be allowed or ruled out. Likewise, overlaps in trait content can be allowed or prohibited. Note that when the two dataset reflect different traits, item and trait overlaps can be allowed and there is no point in passing Key1 and Key2 to the function.

Value

A vector containing the requested random correlations.

Author(s)

Rene Mottus <rene.mottus@ed.ac.uk>

See Also

[specificity](#)

Examples

```
# Create random data.frames

selfratings <- as.data.frame(matrix(ncol=60, nrow=100,
  sample(1:5, size=600, replace=TRUE)))
informantratings <- as.data.frame(matrix(ncol=60, nrow=100,
  sample(1:5, size=600, replace=TRUE)))
colnames(selfratings) <- colnames(informantratings) <- c(paste("Per", 1:60, sep=""))
other.inventory <- as.data.frame(matrix(ncol=100, nrow=100,
  sample(1:5, size=1000, replace=TRUE)))

# Create key (optional)
```

```
key1 <- rep(1:5, each=12)
key2 <- rep(1:5, each=20)

# Analyses

rcAcrossRaters = randomCorrelations(Data1 = selfratings, Data2 = informantratings,
n.items = 12, R=100, item.overlap = FALSE, trait.overlap = TRUE)

rcWithinRaters = randomCorrelations(Data1 = selfratings, Data2 = informantratings,
n.items = 12, R=100, item.overlap = FALSE, trait.overlap = FALSE, Key1 = key1)

rcAcrossQuestionnaire = randomCorrelations(Data1 = selfratings, Data2 = other.inventory,
n.items = 12, R=100, item.overlap = FALSE, trait.overlap = FALSE, Key1 = key1, Key2 = key2)

rcCompleteOverlap = randomCorrelations(Data1 = selfratings, Data2 = informantratings,
n.items = 12, R=100, complete.overlap = TRUE)

# Look at the results

summary(rcAcrossRaters)
summary(rcWithinRaters)
summary(rcAcrossQuestionnaire)
summary(rcCompleteOverlap)
```

specificityLm, specificityEta2, specificityGlm
Specificity of personality trait-outcome associations

Description

These functions test the specificity of personality trait-outcome associations by comparing the observed associations to those obtained using randomly created personality scales. Running the function requires item-level data from multi-trait personality questionnaires and the scoring key for the questionnaire.

The functions are based on `lm` (or on `glm`, if `specificityGlm` is used).

Usage

```
specificityLm(Formula, Data, Key, Shuffle = "exclusive", R = 1000)
```

```
specificityEta2(Formula, Data, Key, Shuffle = "exclusive", R = 1000)
```

```
specificityGlm(Formula, Data, Key, Shuffle = "exclusive", Family="binomial",
R = 1000)
```

Arguments

Formula	A formula specifying the associations to be tested. It should be something along these lines: <code>Outcome ~ .allTraits + Covar1 + Covar2</code> , with <code>.allTraits</code> being a constant that denotes all personality traits measured by the questionnaire (this argument has to be included in all formulas!), <code>Outcome</code> and <code>Covar*</code> denoting the variables in the <code>data.frame</code> that are to be treated as the dependent variable and covariates, respectively. Alternatively, if personality traits are considered as dependent variables, the order of variables can be changed as follows: <code>.allTraits ~ Outcome + Covar1 + Covar2</code> . Note that in either case, covariates must be listed after both <code>.allTraits</code> and <code>Outcome</code> . If personality trait scores need to be standardized, use <code>scale(.allTraits)</code> ; if <code>Outcome</code> needs to be standardized, use <code>scale(Outcome)</code> .
Data	The <code>data.frame</code> to be used.
Key	A list, which has (at least) three elements: <code>start</code> and <code>end</code> denote the positions of the first and the last personality test item in the <code>data.frame</code> , respectively; <code>scale</code> is a vector that denotes the scoring key of the questionnaire (its length equals the number of items in test and each number corresponds to one trait: for example <code>scale=c(1,2,3,4,5,1,2,3,4,5)</code> corresponds to a 10-item questionnaire that measures five traits, with items being ordered as shown). The list may also have a fourth element (<code>names</code>), which is a vector containing the personality trait names (if <code>names</code> is not specified, trait names will be automatically generated). See the example below.
Shuffle	This specifies the way random scales are created: "none" means that no random scales are created, "inclusive" means that items to be aggregated as random scales are randomly selected from among all items, "exclusive" means that different random scales are created for each personality trait such that the items of particular traits are not included in any random scale.
R	The number of simulations.
Family	NB! Only to be specified, when <code>specificityGlm</code> is used. Specifies the family of the <code>glm</code> . See the example below.

Details

The Five-Factor Model personality traits tend to be associated with many outcomes (non-personality variables) in a fairly similar manner and the same outcomes often have similar associations with most personality traits. In particular, socially favourable personality trait levels correlate among themselves and with desirable outcomes; therefore, we call this the 'positive-things-go-together' pattern. There are several possible reasons for this pattern. It could reflect veridical and meaningful ways in which people differ from each other, along the lines of general factor of personality or yet more general fitness factors. Alternatively, it could result from complex multivariate causal mechanisms (e.g., the crud factor) or from methodological artifacts (e.g., rating biases, poor questionnaire design). This ambiguity may limit the informativeness of particular personality trait-outcome associations. In order to quantify the level of uniqueness in particular personality trait-outcome associations, over and above the 'positive-things-go-together' pattern, we have put forward the concept of specificity, operationally defined as the probability that an observed trait-outcome association is stronger than the association of any random combination of personality questionnaire items with

the outcome. This definition provides a simple specificity metric that can be used as a post hoc tool to evaluate the substantive importance of specific trait-outcome associations.

These functions facilitate carrying out such specificity analyses.

`specificityLm` is suitable, when the Outcome is a normally distributed continuous variable, whereas `specificityEta2` is suitable, when the Outcome is the independent variable and is a factor; `specificityGlm` is suitable, when the Outcome is the dependent variable and has a non-normal distribution.

Value

...	The result will be an object of S3 class. Typing <code>summary(...)</code> will display observed associations between personality trait and the outcome (the associations are calculated for all personality traits separately including all covariates as specified in the Formula), specificity estimates and adjusted effects sizes (observed effect size minus average random effect).
observed	The observed associations of personality traits with the outcome. For a nicer output use: <code>summary(...)\$true.results</code>
random	Random associations of personality traits with the outcome. For a nicer output use: <code>summary(...)\$rand.results.all.traits</code>
time	The time used for running the random-scale analyses

Author(s)

Kenn Konstabel <kenn.konstabel@tai.ee>, Rene Mottus <rene.mottus@ed.ac.uk>

References

Mottus, R., Johnson, W., Booth, T., Deary, I. J., Konstabel, K., Realo, A., & Allik, J. (2013). Generality and specificity of personality trait scores in their associations with other phenomena. Submitted for publication

See Also

[randomCorrelations](#)

Examples

```
# Create random data.frame
data = as.data.frame(matrix(ncol=63, nrow=100, sample(1:5, size=6300, replace=TRUE)))
colnames(data) = c(paste("Per", 1:60, sep=""), c("Outcome", "Covar1", "Covar2"))

# Make the Outcome categorical
data$OutcomeCat = as.factor(data$Outcome)

# Create Keys
keyWithNames <- list(start=1, end=60, scale=rep(1:5, each=12), names=c("N","E","O","A","C"))
keyWithoutNames <- list(start=1, end=60, scale = rep(1:5, each=12))

# Run analyses
```

```
resultLm <- specificityLm(scale(.allTraits) ~ scale(Outcome) + Covar1 + Covar2,  
  Data=data, Key = keyWithNames, R=100)  
resultEta2 <- specificityEta2(.allTraits ~ OutcomeCat + Covar1 + Covar2,  
  Data=data, Key = keyWithNames, R=100)  
resultGlm <- specificityGlm(Outcome ~ .allTraits + Covar1 + Covar2,  
  Data=data, Family="gaussian", Key = keyWithoutNames, R=100)  
  
# See results  
summary(resultLm)  
summary(resultEta2)$true.results  
summary(resultGlm)$rand.results.all.traits
```

Index

`dum (specificityLm, specificityEta2, specificityGlm)`, 5

`eta2`, 2

`family`, 6

`glm`, 5

`lm`, 5

`print.summary.specificity (specificityLm, specificityEta2, specificityGlm)`, 5

`randomCorrelations`, 3, 7

`scramble (specificityLm, specificityEta2, specificityGlm)`, 5

`specEta2 (specificityLm, specificityEta2, specificityGlm)`, 5

`specGlm (specificityLm, specificityEta2, specificityGlm)`, 5

`specificity`, 2–4

`specificity (specificityLm, specificityEta2, specificityGlm)`, 5

`specificity-package (specificityLm, specificityEta2, specificityGlm)`, 5

`specificityEta2 (specificityLm, specificityEta2, specificityGlm)`, 5

`specificityGlm (specificityLm, specificityEta2, specificityGlm)`, 5

`specificityLm (specificityLm, specificityEta2, specificityGlm)`, 5

`specificityLm, specificityEta2, specificityGlm`, 5

`specLm (specificityLm, specificityEta2, specificityGlm)`, 5

`summary.specificity (specificityLm, specificityEta2, specificityGlm)`, 5