# Package 'ConjointChecks'

August 29, 2016

**Title** A package to check the cancellation axioms of conjoint measurement.

**Version** 0.0.9

**Date** 2012-12-14

**Author** Ben Domingue <ben.domingue@gmail.com>

**Maintainer** Ben Domingue <ben.domingue@gmail.com>

**Depends** R (>= 2.14.0), parallel, methods

**Suggests** snow, Rmpi

**Description** Implementation of a procedure (Domingue, 2012; see also Karabatsos, 2001 and Kyngdon, 2011) to test the single and double cancellation axioms of conjoint measure in data that is dichotomously coded and measured with error.

**LazyData** Yes

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2012-12-15 07:27:55

**NeedsCompilation** no

## R topics documented:

## ConjointChecs-package *ConjointChecks: A package to check the cancellation axioms of conjoint measurement.*

### Description

Implementation of a procedure (Domingue, 2012; see also Karabatsos, 2001 and Kyngdon, 2011) to test the single and double cancellation axioms of conjoint measure in data that is dichotomously coded and measured with error.

### Author(s)

Ben Domingue <ben.domingue@gmail.com>

### References

Domingue, B. (2012). Evaluating the Equal-Interval Hypothesis with Test Score Scales. Doctoral Dissertation, University of Colorado Boulder, May 2012.

Karabatsos, G. (2001). The rasch model, additive conjoint measurement, and new models of probabilistic measurement theory. Journal of Applied Measurement, 2(4), 389-423.

Kyngdon, A. (2011). Plausible measurement analogies to some psychometric models of test performance. British Journal of Mathematical and Statistical Psychology, 64(3), 478-497.

Perline, R., Wright, B. D., & Wainer, H. (1979). The Rasch model as additive conjoint measurement. Applied Psychological Measurement, 3(2), 237-255.

## checks-class *Class "checks"*

### Description

The formal S4 class for checks. This class contains transformed version of the raw response data as well as summaries of the checks.

### Details

Objects of class `checks` contains all information returned by `ConjointChecks`.

### Objects from the Class

Object created by a call to function `ConjointChecks`.

### Slots

N: matrix containing the number of respondents at each item/ability intersection

n: matrix containing the number of correct responses at each item/ability intersection

Checks: List containing information about each checked 3-matrix

tab: matrix containing information about the detected violations at each item/ability intersection

means: vector containing weighted and unweighted means for the detected violations (where weights are the number of individuals at each ability level)

check.counts: matrix giving the number of times a item/ability cell was sampled

### Author(s)

Ben Domingue <ben.domingue@gmail.com>

### See Also

ConjointChecks, summary.checks, plot.checks

---

## ConjointChecks *Check Single and Double Cancellation in a sample of 3-matrices*

---

### Description

Given two matrices, n and N (which contain the number of correct responses and the number of total responses for each cell), a check of single and double cancellation is performed in n.3mat matrices. To check large numbers of 3-matrices (to see why, see Domingue (2012)), parallel options help.

### Usage

```
ConjointChecks(N,n,n.3mat=1,par.options=NULL,CR=c(.025,.975),seed=NULL)
```

### Arguments

| | |
|---|---|
| N | Matrix containing the total number of responses. |
| n | Matrix containing the number of correct responses. |
| n.3mat | Number of 3-matrices to sample or the string "adjacent" if all adjacently formed 3-matrices are to be checked. |
| par.options | A named list indicating "n.workers" and "type". The first defaults to unity and the latter to PSOCK. |
| seed | Random number seed. |
| CR | Width of the credible region taken from the posterior. Defaults to a 95% credible region (c(.025,.975)). |

### Author(s)

Ben Domingue <ben.domingue@gmail.com>

## References

Perline, R., Wright, B. D., & Wainer, H. (1979). The Rasch model as additive conjoint measurement. Applied Psychological Measurement, 3(2), 237-255.

## Examples

```
#####################################################
#parole data
#page 244 (table 2) of Perline, Wright, and Wainer
#about 9% were bad in perline
matrix(c(15,47,61,84,82,86,60,47,8),9,9,byrow=FALSE)->N
per <-structure(c(0, 0.06, 0.07, 0.18, 0.13, 0.13, 0.17, 0.17,
 1, 0, 0.04, 0.15, 0.24, 0.33, 0.28, 0.47, 0.85, 1, 0, 0.04, 0.08,
 0.12, 0.3, 0.64, 0.85, 1, 1, 0, 0.19, 0.39, 0.4, 0.51, 0.58,
 0.82, 0.98, 1, 0, 0.06, 0.18, 0.52, 0.73, 0.95, 1, 1, 1, 0,
 0.23, 0.33, 0.51, 0.68, 0.91, 0.93, 1, 1, 0.27, 0.51, 0.61,
 0.64, 0.68, 0.77, 0.9, 1, 1, 0, 0.21, 0.52, 0.68, 0.84, 0.97,
 0.97, 1, 1, 0.73, 0.64, 0.67, 0.7, 0.78, 0.78, 0.9, 1, 1),
 .Dim = c(9L, 9L) )
round(per*N)->n
ConjointChecks(N,n,n.3mat=1)->out

#####################################################
#Data from Rasch (1960) data
#page 250 (table 5) of Perline, Wright, and Wainer
#about 4% showed violations
matrix(c(49,112,32,76,82,102,119,133,123,94,61,17,10),13,7,byrow=FALSE)->N
per <-structure(c(0, 0, 0, 0, 0.02, 0.01, 0.02, 0.03, 0.06, 0.09,
 0.23, 0.35, 0.7, 0.01, 0, 0.04, 0.05, 0.09, 0.09, 0.16, 0.28, 0.39,
 0.66, 0.8, 0.91, 0.85, 0, 0.02, 0.07, 0.07, 0.24, 0.28, 0.45, 0.59,
 0.76, 0.87, 0.9, 1, 0.85, 0.01, 0.04, 0.12, 0.21, 0.42, 0.62, 0.73,
 0.83, 0.9, 0.93, 0.98, 1, 1, 0.06, 0.11, 0.4, 0.7, 0.7, 0.79, 0.84,
 0.88, 0.94, 0.95, 0.98, 1, 1, 0.48, 0.84, 0.84, 0.86, 0.86, 0.9,
 0.95, 0.96, 0.98, 0.99, 0.99, 1, 1, 0.92, 0.98, 0.98, 0.99, 0.98,
 0.99, 0.99, 1, 1, 1, 1, 1, 1), .Dim = c(13L, 7L))
round(per*N)->n
ConjointChecks(N,n,n.3mat=1)->out

############
#simulated rasch example
n.3mat<-5000
n.items<-20
n.respondents<-2000
#simulate data
rnorm(n.items)->diff
rnorm(n.respondents)->abil
matrix(abil,n.respondents,n.items,byrow=FALSE)->m1
matrix(diff,n.respondents,n.items,byrow=TRUE)->m2
m1-m2 -> kern
exp(kern)/(1+exp(kern))->pv
runif(n.items*n.respondents)->test
ifelse(pv>test,1,0)->resp
```

```
##now check
PrepareChecks(resp)->tmp
#not run
#detectCores()->n.workers
#ConjointChecks(tmp$N,tmp$n,n.3mat=5000,
#  par.options=list(n.workers=n.workers,type="PSOCK"))->rasch5000
```

---

plot.checks *Plot checks produced by* ConjointChecks.

---

## Description

Takes output from ConjointChecks and produces a matplot showing the percentage of reported violations at each cell.

## Usage

```
## S3 method for class 'checks'
plot(x,items=NULL,item.labels=TRUE,...)
```

## Arguments

| | |
|---|---|
| x | Object returned by ConjointChecks of class checks. |
| items | Vector of item numbers to include in a single plot. Defaults to all, but this is less helpful for diagnostic purposes. |
| item.labels | Should item numbers be included? Defaults to TRUE. If length of items is unity (perhaps if the small multiple format of Tufte, 2001 is going to be used), then the item number gets printed below the x-axis. If the length of items is more than unity, the item number gets plotted in the figure above the largest proportion of violations for each item. |
| ... | further arguments passed to or from other methods |

## References

Tufte, E. R. (2001). The visual display of quantitative information (2nd ed.). Chesire, CT: Graphics Press.

## Examples

```
par(mfrow=c(3,2))
plot(rasch5000)
plot(rasch5000,items=c(5,10,15))
for (i in c(3,9,13,18)) plot(rasch5000,items=i)
```

## PrepareChecks *Prepare raw response data for* ConjointChecks.

### Description

Takes output from ConjointChecks and produces a matrix showing the percentage of reported violations at each cell.

### Usage

```
PrepareChecks(resp,ss.lower=10)
```

### Arguments

| | |
|---|---|
| resp | Raw dichotomously coded response data. Columns represent items and rows represent individuals. |
| ss.lower | Only sum scores that have at least this many distinct individuals with that sum score will be used. |

### Examples

```
#simulated Rasch example
n.items<-20
n.respondents<-2000
#simulate data
rnorm(n.items)->diff
rnorm(n.respondents)->abil
matrix(abil,n.respondents,n.items,byrow=FALSE)->m1
matrix(diff,n.respondents,n.items,byrow=TRUE)->m2
m1-m2 -> kern
exp(kern)/(1+exp(kern))->pv
runif(n.items*n.respondents)->test
ifelse(pv>test,1,0)->resp
#now check
PrepareChecks(resp)->obj
```

## rasch5000 *5000 sampled 3-matrices from simulated Rasch data.*

### Description

Object created by first generating Rasch data and then running ConjointChecks on 5000 sampled 3 matrices

### Usage

```
rasch5000
```

## Format

An object of class checks.

## Source

Simulated via Rasch model.

---

summary.checks *Summarize checks produced by* ConjointChecks.

---

## Description

Takes output from ConjointChecks and produces a matrix showing the percentage of reported violations at each cell.

## Usage

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

## Arguments

| | |
|---|---|
| object | Object returned by ConjointChecks of class checks. |
| ... | further arguments passed to or from other methods |

## Examples

```
summary(rasch5000)
```

# Index