Package 'TEIscoring'

June 6, 2018

Title Apply a set of Scoring Algorithms to Technology Enhanced Item Data

Version 0.0.0.9000

Description Generates scores from data generated from technology enhanced item types. Currently, the package contains scoring functions.

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

`Balanced` *Balanced Scoring Algorithm*

Description

An adaptation of the Ripkey method found in Tarasowa and Auer (2013).

Usage

```
Balanced(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

`keyNum` The number of correct response options in the key.

`correctCheckNum` The number of correct response options marked by the examinee

`distractorNum` The number of innocrrect response options in the key.

`incrrectCheckNum` The number of incorrect response options marked by the examinee

Examples

```
Balanced(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

`DichoScoring` *Dichotomous Scoring*

Description

Dichotomous Scoring for TEIs. Produces a value of 1 the all correct choices were marked and zero incorrect choices were marked.

Usage

```
DichoScoring(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

`keyNum` The number of correct response options in the key.

`correctCheckNum` The number of correct response options marked by the examinee

`distractorNum` The number of innocrrect response options in the key.

`incrrectCheckNum` The number of incorrect response options marked by the examinee

Examples

```
DichoScoring(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

MCC *Matthews correlation coefficient*

Description

Calculates the Matthews correlation coefficient as a scoring algorithm.

Usage

```
MCC(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

`keyNum` The number of correct response options in the key.

`correctCheckNum` The number of correct response options marked by the examinee

`distractorNum` The number of innocrrect response options in the key.

`incrrectCheckNum` The number of incorrect response options marked by the examinee

Examples

```
MCC(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

Morgan *The Morgan scoring algorithm*

Description

Implements the scoring algorithm described in Morgan (1979). Penalizes incorrect responses and rewards correctly marked responses. Also known as the 'Middlesex Scoring Scheme', discussed in detail in Buckly-Sharp and Harris (1971).

Usage

```
Morgan(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

`keyNum` The number of correct response options in the key.

`correctCheckNum` The number of correct response options marked by the examinee

`distractorNum` The number of innocrrect response options in the key.

`incrrectCheckNum` The number of incorrect response options marked by the examinee

Examples

```
Morgan(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

plusMinus *Plus/Minus Scoring Algorithm*

Description

SA 2 from Domnich et al (2015). Recieves the addition of

Usage

```
plusMinus(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

`keyNum` The number of correct response options in the key.

`correctCheckNum` The number of correct response options marked by the examinee

`distractorNum` The number of innocrrect response options in the key.

`incrrectCheckNum` The number of incorrect response options marked by the examinee

Examples

```
plusMinus(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

Ripkey *Ripkey Partial Credit Scoring*

Description

Partial credit scoring algorithm described in Ripkey et al (1996) for TEIs. Evaluates to the number of correctly marked response options divided by the total number of keyed-correct options. If more than the keyed number of options are marked, the resulting score is zero.

Usage

```
Ripkey(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

`keyNum` The number of correct response options in the key.

`correctCheckNum` The number of correct response options marked by the examinee

`distractorNum` The number of innocrrect response options in the key.

`incrrectCheckNum` The number of incorrect response options marked by the examinee

Examples

```
Ripkey(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

TrueFalse *True-False Scoring Algorithm*

Description

Formula 4 (Husa et al, 1984) or SA 3 (Domnich et al, 2015). Simply the total number of options appropriately marked divided by the number of options. Therefore, has no penalty marking an incorrect item. Equivalent to the proportion of correct responses if each choices was scored individually as a True-False item. Hence, the name.

Usage

```
TrueFalse(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

keyNum	The number of correct response options in the key.
correctCheckNum	The number of correct response options marked by the examinee
distractorNum	The number of innocrrect response options in the key.
incrrectCheckNum	The number of incorrect response options marked by the examinee

Examples

```
TrueFalse(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

TrueFalseP *True-False with Penalty Scoring Algorithm*

Description

SA 4 from Domnich et al (2015).

Usage

```
TrueFalseP(keyNum, correctCheckNum, incorrectCheckNum, distractorNum)
```

Arguments

keyNum	The number of correct response options in the key.
correctCheckNum	The number of correct response options marked by the examinee
distractorNum	The number of innocrrect response options in the key.
incrrectCheckNum	The number of incorrect response options marked by the examinee

Examples

```
TrueFalseP(keyNum = 4, correctCheckNum = 4, incorrectCheckNum = 0,distractorNum = 4)
```

Index