Package: numGen (via r-universe)

October 12, 2024

Type Package

Title Number Series Generator

Version 0.1.1

Date 2017-07-04

Maintainer Bao Sheng Loe (Aiden) <bs128@cam.ac.uk>

Description A number series generator that creates number series items based on cognitive models.

License GPL-3

LazyData TRUE

RoxygenNote 6.0.1

Repository https://aidenloe.r-universe.dev

RemoteUrl https://github.com/aidenloe/numgen

RemoteRef HEAD

RemoteSha b142e683a0d7d3216e1f513a2b923d73554139ac

Contents

imEight *Item Model 8*

Description

This uses item model 8 to create number series items.

Usage

```
imEight(cat, n, items, arith)
```

Arguments

cat	Number of categorical groups per question.
n	The differences between the pair of objects
items	The number of items you want to generate.
arith	The arithmetic operator of your choice ("add","substr","multi","div").

Details

This is based on the categorical / pattern recognition rule. Neighbouring pairs or triads of objects are related, includes arithmetic operations.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imEight(cat=2,n=4,items=2, arith="add")
```

imEleven *Item Model 11*

Description

This uses item model 11 to create number series items - Identification of alternating coefficients of change.

Usage

```
imEleven(items = 1, fun1 = "add", fun2 = "add")
```

Arguments

items	Generate a random mix of items.
fun1	The argument decides the arithmetic to be employed for Neighbouring objects. There are only two arithmetic: add, substr.
fun2	The argument decides the arithmetic to be employed for the two values between the grouped objects. There are two arithmetic: add, substr.

Details

This function creates number series that is a combination of Neighbouring objects and 2-sequence coefficient. Multiplication and Division are removed since the calculated value is too big. Example: A sequence whose coefficient of change alternates between (add 6) and (multiply by 2). 1 7 14 20 40 46 (92) (98).

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
#Draws 5 items randomly.
imEleven(items=5, fun1 = "add", fun2= "add")
```

---

imFive *Item Model 5*

Description

This uses item model 5 to create number series items - Identification of co-occurring relationships between elements (with use of arithmetic skills)

Usage

```
imFive(arithOne = "add", arithTwo = "substr", n = 2, items = 4)
```

Arguments

arithOne	Select the arithmetic operator of choice ("add","multi", "sub", "div").
arithTwo	Select the arithmetic operator of choice ("add","multi", "sub", "div").
n	Value you want use the arithmetic operator on.
items	Generate a random mix of items.

Details

Logic analogous to the Item Model 4, but at least one sub-sequence involves the basic arithmetic operations. Sequences combine items from Item Families 1 and 3. The arithmetic operations change but the differences in value remains the name. Example: Odd elements of the sequence increase by 2 and even elements of the sequence are multiplied by 2. (2 12 4 24 6 48 8 (96) (10))

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imFive(arithOne="add",arithTwo="add",n=2,items=5)
```

`imFour` *Item Model 4*

Description

This uses item model 4 to create number series items - Identification of co-occurring relationships between elements (without use of arithmetic skills)

Usage

```
imFour(items = 5, seed = 1)
```

Arguments

`items`	Number of items to generate.
`seed`	This gives you the same result again.

Details

Sequences which consist of regularly alternating parallel sub-sequences. Understanding of succession does not require use of algebraic skill. Sub-sequences involve items from Item Model 1. Example: Odd elements of the sequence are multiples of 1 and even elements of the sequence are multiples of 10. (1 10 2 20 3 30 (4) (40)) 2 simple linear (without arithmetic) 1 2 3 / 10 20 30 combine to form a number series item.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
## Not run:

imFour(items=5, seed=5)


## End(Not run)
```

`imNine` *Item Model 9*

Description

This uses item model 10 to create number series items - Identification of relationships within a chain of elements.

Usage

```
imNine(items)
```

Arguments

`items`	Number of items to generate.

Details

Progressive sequences which involve relationships between multiple preceding objects (e.g. Fibonacci sequence). Example: Each element of the sequence is a result of addition of its two preceding elements (1 1 2 3 5 8 (13)). The maximum number to be generated is 15 items.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imNine(items=3)
```

imOne *Item Model 1*

Description

This uses item model 1 to create number series items - Elementary understanding of sequence succession.

Usage

```
imOne(items = 5, seed = 1)
```

Arguments

items	Number of items to generate.
seed	Setting the seed returns the same items on the local computer.

Details

Simple linear sequences which do not require use of advanced arithmetic operations, such as ordered multiples of 1, 10, or 100. Example: A sequence of ordered multiples of 10. (10 20 30 40 (50)).

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imOne(items=5, seed=5)
```

imSeven *Item Model 7*

Description

This uses item model 7 to create number series items - Identification of complex coefficients of change

Usage

```
imSeven(vOne = 1, vTwo = 3, items, seed = 1, logic = "one",
  random = FALSE)
```

Arguments

vOne	The first value in the complex coefficient (x). Can be a sequence of values or a specific value.
vTwo	The second value in the complex coefficient (y). Can be a sequence of values or a specific value.
items	Generate a random mix of items.
seed	To get the same random sampling of items
logic	"one" or "two"
random	If random=FALSE, the items will follow in sequential order.

Details

This function creates number series that is a combination of Arithmetic, Linear and Complex coefficient. Ability to identify complex coefficients; the coefficient of change involves a combination of arithmetic operations (e.g. addition and multiplication) applied serially.
There are two logic to calculate the number series. First logic of complex coefficient = i*x+y.
Second logic of complex coefficient = (i+x)*y.
. Example: Each element in the sequence is derived from the preceding by adding two and multiplying the result by two. (2 8 20 44 92 (188)).

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
#Draws 5 items randomly.
imSeven(vOne=1,vTwo=3,items=5,seed=2,logic="one",random=TRUE)

# Calculates all combinations
# Items and seed arg is ignored.
imSeven(vOne=1:2,vTwo=1:3,items=5,seed=2,logic="one",random=FALSE)
```

---

imSix *Item Model 6*

Description

This uses item model 6 to create number series items - Identification of progressively evolving coefficients of change.

Usage

```
imSix(items)
```

Arguments

items	Number of items to generate.

Details

Non-linear progressive sequences which require a higher level of abstraction; the coefficient of change between two neighbouring elements is not invariable and its elements form a sequence. The coefficient sequences correspond to items from Item Families 1 and 3. Example: The coefficient of change between each pair of neighbouring elements in the sequence increases by 1. (2 4 7 11 16 (22))

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imSix(items=3)
```

---

imTen *Item Model 10*

Description

This uses item model 10 to create number series items - Combined identification of parallel subsequences and progressively evolving coefficients of change.

Usage

```
imTen(items, logic = "one", n = 2, arith = "add")
```

Arguments

items	Generate a random mix of items.
logic	The combination of sequences follow two logic ("one" or "two").
n	The value that the arithmetic operator uses to calculate the next value
arith	The arithmetic operator of your choice ("add","substr","multi","div").

Details

The number series items are a combination of Arithmetic, linear sequence and progressive coefficient.
First logic is combining sequences x y x y x y x y = one simple (cannot be controlled), one progressive .
Second logic is combining sequences x y x y x y x y = two progressive. The minimum number of items that will be generated is 2.

Logic analogous to the Item Model 5, but at least one sub-sequence involves a progressively evolving coefficient. Sub-sequences involve items from Item Families 1, 3, and 7. Example: The coefficient of change between odd elements in the sequence increases by 1. The coefficient of change between even elements increases by -1. (2 8 4 7 7 5 11 2 16 (-2) (22)).

When using the first logic, n corresponds to the change in the progressive pattern. However, the simple pattern is fixed and hence drawn randomly.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
#Draws 10 items randomly.
imTen(10,logic="one", n=2,arith="add")
```

`imThirteen` *Item Model 13*

Description

This uses item model 13 to create number series items - Combined identification of unevenly ordered sub-sequences and non-successive relationships between elements.

Usage

```
imThirteen(items)
```

Arguments

`items`	Generate a random mix of items.

Details

This function creates number series creates a combination of sequences and ratios. TLogic analogous to the Item Model 13, but the second sequence belongs to the Item Model 9. As a result, pairs of elements following certain rule are embedded into a progressive sequence. Example: Sequence with coefficient of (+ 1) is interposed with pairs of elements which differ by 3. 1 5 8 2 209 212 3 41 (44) (4). Only the addition and substraction arimethic operators are used to generate the number series items.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
#Draws 10 items randomly.
imThirteen(10)
```

---

imThree *Item Model 3*

Description

This uses item model 3 to create number series items - Use of basic algebraic skills.

Usage

```
imThree(items, n, arith = "add")
```

Arguments

items	The number of items to generate
n	Value to use the arithmetic operator on
arith	Use either 'add', 'substr', 'multi', 'div'.

Details

Each element in the sequence is derived from the preceding by applying one of four basic arithmetic operations - addition, subtraction, multiplication, or division. Coefficient of change is invariant across the sequence. 20 18 16 14 (12). Currently it only displays up to a series of 9.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imThree(items=4,n=2,arith="add")
```

imTwelve *Item Model 12*

Description

This uses item model 12 to create number series items - Identification of unevenly ordered subsequences

Usage

```
imTwelve(items)
```

Arguments

items	Generate a random mix of items.

Details

This function creates number series that is a irregular combination of sequences a b b a b b a ... Only the addition and substraction arithmetic operators are used to create the number series items.

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
#Draws 10 items randomly.
imTwelve(10)
```

imTwo *Item Model 2*

Description

This uses item model 2 to create number series items - Understanding of object categorisation.

Usage

```
imTwo(cat = 2, items = 4, random = FALSE)
```

Arguments

cat	Length of categorical groups per question.
items	The number of items you want to generate.
random	To randomise the position of the numeric values.

Details

Sequences consist of elements belonging to two homogeneous groups with equal number of elements. Missing element belongs to the group with fewer elements present in the sequence. For example, 1 1 1 5 5 (5).

Author(s)

Aiden Loe and Filip Simonfy

Examples

```
imTwo(cat=2,items=4,random=FALSE)
```

---

numGen *numGen: A package for generating number series items.*

Description

The numGen package provides 14 item models for generating number series items.

Item model 1

This number series generates simple linear sequences with a magnitude of up to 5000.
imOne

Item model 2

This number series generate sequences consist of elements belonging to two homogeneous groups with equal number of elements.
imTwo

Item model 3

This function allows you to select one of the four arithmetic operators following a sequence succession rule.
imThree

Item model 4

This create items that relates to comprehension of abstract object representation (Item model 5) and Identification of co-occurring relationships between elements (Item model 4).
imFour

Item model 5

Generate items with two sequences combined into one number series.
imFive

Item model 6

This model uses the addition and substraction (Arithmetic) operator, Linear pattern and Progressive coefficient to create the number series.
imSix

Item model 7

This function creates number series that is a combination of Arithmetic, Linear and Complex coefficient.
First logic of complex coefficient = i*x+y.
Second logic of complex coefficient = (i+x)*y.
imSeven

Item model 8

This is based on the categorical / pattern recognition rule. Neighbouring pairs or triads of objects are related, includes arithmetic operations.
imEight

Item model 9

This function creates Fibonacci sequences. The maximum number to be generated is 15 items.
imNine

Item model 10

The number series is a combination of Arithmetic, linear sequence and progressive coefficient.
First logic is combining sequences x y x y x y x y = one simple (cannot be controlled), one progressive.
Second logic is combining sequences x y x y x y x y = two progressive.
imTen

Item model 11

Neighbouring objects + 2-sequence coefficient.
This function creates number series that is a combination of Neighbouring objects + 2-sequence coefficient.
Multiplication and Division is removed since the calculated value is too big.
imEleven

Item model 12

This function creates number series that is a irregular combination of sequences a b b a b b a ...
Only the addition and substraction arithmetic operators are used to create the number series items.
imTwelve

Item model 13

Combination of sequences and ratios.
imThirteen

References

LeFevre, J. A., & Bisanz, J. (1986). A cognitive analysis of number-series problems: Sources of individual differences in performance. *Memory & Cognition*, 14(4), 287-298.

Holzman, T. G., Pellegrino, J. W., & Glaser, R. (1983). Cognitive variables in series completion. *Journal of Educational Psychology*, 75(4), 603.

Simon, H. A., & Kotovsky, K. (1963). Human acquisition of concepts for sequential patterns. *Psychological Review*, 70(6), 534.

Index