Package 'lwc2022'

November 20, 2024

Title Langa-Weir Classification of Cognitive Function for 2022 HRS Data

Version 1.0.0

URL https://github.com/C-Monaghan/lwc2022, https://c-monaghan.github.io/lwc2022/

BugReports https://github.com/C-Monaghan/lwc2022/issues

Language en-US

Description Generates the Langa-Weir classification of cognitive function for the 2022 Health and Retirement Study (HRS) cognition data. It is particularly useful for researchers studying cognitive aging who wish to work with the most recent release of HRS data. The package provides user-friendly functions for data preprocessing, scoring, and classification allowing users to easily apply the Langa-Weir classification system. For details regarding the; HRS <https://hrsdata.isr.umich.edu/> and Langa-Weir classifications <https://hrsdata.isr.umich.edu/data-products/langa-weir-classification-cognitive-function-1995-2020>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports dplyr

Suggests knitr

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Cormac Monaghan [cph, aut, cre] (<https://orcid.org/0000-0001-9012-3060>), Rafael de Andrade Moral [aut] (<https://orcid.org/0000-0002-0875-3563>), Joanna McHugh Power [aut] (<https://orcid.org/0000-0002-7387-3107>)

Maintainer Cormac Monaghan <cormacmonaghan@protonmail.com>

Repository CRAN

Date/Publication 2024-11-20 10:40:07 UTC

Contents

`classify` *Classify Cognitive Function Based on Total Scores*

Description

This function classifies individuals into cognitive function groups based on their total cognition score, which is calculated from immediate word recall, delayed word recall, serial subtraction, and backwards counting scores. The classification creates three categories of cognitive function.

Usage

```
classify(data)
```

Arguments

`data`	A dataframe containing cognitive test scores, including total immediate word recall, delayed word recall, serial subtraction, and backwards counting scores.

Details

The function creates a total cognitive score by summing the scores for immediate word recall, delayed word recall, serial subtraction, and backwards counting. It then classifies the cognitive function into three levels:

- Class 1: Normal (total score >= 12).
- Class 2: Cognitive impairment no dementia (total score between 7 and 11).
- Class 3: Demented (total score <= 6).

Value

A dataframe with:

- `Total_cog_score`: Total cognitive score (sum of all individual task scores).
- `Class`: Cognitive function classification (1 = Normal, 2 = Cognitive impairment no dementia, 3 = Demented).
- Renamed columns with updated labels for 2022 data: `imrc_imp2022`, `dlrc_imp2022`, `ser7_imp2022`, `bwc20_imp2022`, `cogtot27_imp2022`, and `cogfunction2022`.

Examples

```
# Assuming `cog_data` is a dataframe with the relevant columns
classified_data <- classify(cog_data_score)
```

---

cog_data *Cognition Data*

Description

A simulated dataset with cognition test scores, following the same methodology as the Health and Retirement Study (HRS). The dataset includes immediate word recall, delayed word recall, serial subtraction, backwards counting tasks, and mouse click clicking with scores representing cognitive performance on these tests.

Usage

```
cog_data
```

Format

A dataframe with 10 rows and 35 variables:

HHID Household identifier, a unique 6-digit integer.

PN Person number, a unique 1- or 2-digit integer within each household.

SD182M1 Immediate word recall test score for the first word.

SD182M2 Immediate word recall test score for the second word.

SD182M3 Immediate word recall test score for the third word.

SD182M4 Immediate word recall test score for the fourth word.

SD182M5 Immediate word recall test score for the fifth word.

SD182M6 Immediate word recall test score for the sixth word.

SD182M7 Immediate word recall test score for the seventh word.

SD182M8 Immediate word recall test score for the eight word.

SD182M9 Immediate word recall test score for the ninth word.

SD182M10 Immediate word recall test score for the tenth word.

SD183M1 Delayed word recall test score for the first word.

SD183M2 Delayed word recall test score for the second word.

SD183M3 Delayed word recall test score for the third word.

SD183M4 Delayed word recall test score for the fourth word.

SD183M5 Delayed word recall test score for the fifth word.

SD183M6 Delayed word recall test score for the sixth word.

SD183M7 Delayed word recall test score for the seventh word.

SD183M8 Delayed word recall test score for the eight word.

SD183M9 Delayed word recall test score for the ninth word.

SD183M10 Delayed word recall test score for the tenth word.

SD142 Serial subtraction, result of subtracting 7 from 100.

SD143 Serial subtraction, result of subtracting 7 from the previous number.

SD144 Serial subtraction, result of subtracting 7 from the previous number.

SD145 Serial subtraction, result of subtracting 7 from the previous number.

SD146 Serial subtraction, result of subtracting 7 from the previous number.

SD124 Backwards counting test, success on the first attempt (1 = success, 0 = fail).

SD129 Backwards counting test, success on the second attempt (1 = success, 0 = fail).

SD237WA Mouse clicking test: accuracy result (first attemp)

SD237WC Mouse clicking test: total click count (first attemp)

SD237WT Mouse clicking test: total time spent (in seconds; first attempt)

SD238WA Mouse clicking test: accuracy result (second attemp)

SD238WC Mouse clicking test: total click count (second attemp)

SD238WT Mouse clicking test: total time spent (in seconds; second attempt)

Examples

```
# Load the data
data(cog_data)

# View the first few rows
head(cog_data)
```

cog_data_score *Scored Cognition Data*

Description

A simulated dataset with scored cognition test results. This dataset contains calculated total scores for immediate recall, delayed recall, serial subtraction.

Usage

```
cog_data_score
```

Format

A dataframe with 10 rows and 6 variables:

HHID Household identifier, a unique 6-digit integer.

PN Person number, a unique 1- or 2-digit integer within each household.

Total_I Total immediate word recall score, ranging from 0 to 5 (sum of 5 items from the immediate recall test).

Total_D Total delayed word recall score, ranging from 0 to 5 (sum of 5 items from the delayed recall test).

Total_Sub Total serial subtraction score, ranging from 0 to 5 (sum of successful subtractions from the serial subtraction test).

Total_Count Total backwards counting score, ranging from 0 to 2 (2 points for success on the first try, 1 point for success on the second try, and 0 for failure).

Examples

```
# Load the data
data(cog_data_score)

# View the first few rows
head(cog_data_score)
```

extract *Extract Key Cognitive Measures from Dataset*

Description

This function extracts specific cognitive measures from a dataset, including immediate and delayed word recall, serial subtraction, and backwards counting, along with household and person identifiers.

Usage

```
extract(data)
```

Arguments

data	A dataframe containing the full dataset from which specific variables will be selected.

Details

The function selects key cognitive test results and identifiers from the dataset. It uses `dplyr::select()` to retrieve:

- Immediate and delayed word recall variables (those starting with "SD182M" and "SD183M").
- Serial subtraction results (`SD142` to `SD146`).
- Backwards counting variables (`SD124`, `SD129`).

Value

A dataframe with the following variables:

- `HHID`: Household ID.
- `PN`: Person number (individual identifier).
- Immediate and delayed word recall variables (columns starting with `"SD182M"` and `"SD183M"`).
- Serial subtraction variables (`SD142` to `SD146`).
- Backwards counting variables (`SD124`, `SD129`).

Examples

```
# Assuming `cog_data` is a dataframe with the relevant columns
extract(cog_data)
```

---

score *Calculate Cognitive Test Scores*

Description

This function calculates various cognitive test scores from a dataset, including word recall, serial subtraction, and backwards counting. It computes total scores for immediate and delayed word recall, scores for serial subtraction tasks, and a total score for backwards counting.

Usage

```
score(data)
```

Arguments

`data`	A dataframe containing the cognitive test data, including columns for word recall, serial subtraction, and backwards counting tasks.

Details

The function applies scoring functions to the cognitive test data:

- Word recall: Scores immediate and delayed recall using the `score_recall` function, and computes total scores.
- Serial subtraction: Applies the `score_subtraction` function to calculate scores for each subtraction step, and computes the total score.
- Backwards counting: Assigns 2 points for correct counting on the first try, 1 point for correct counting on the second try, and 0 for incorrect counting.

Value

A dataframe with the following computed scores:

- `Total_I`: Total score for immediate word recall.
- `Total_D`: Total score for delayed word recall.
- `Total_Sub`: Total score for serial subtraction.
- `Total_Count`: Total score for backwards counting.

Examples

```
# Assuming `cog_data` is a dataframe with the relevant columns
scored_data <- score(cog_data)
```

`score_recall` *Score Word Recall Task*

Description

This function scores a word recall task where respondents are given 1 for a correct recall and 0 for an incorrect recall. Missing values (NA) are retained as NA in the output.

Usage

```
score_recall(x)
```

Arguments

`x`	A numeric vector representing respondents' word recall responses. Specific numeric codes are used to define incorrect responses.

Details

The function assigns a score of 1 for a correct word recall. Incorrect recall is determined by specific numeric codes (51 to 67, 96, 98, and 99) and assigned a score of 0. Any NA values in the input will remain NA in the output.

Value

A numeric vector where:

- `1`: Correct recall.
- `0`: Incorrect recall (based on specific codes).
- `NA`: If the original value was missing, it remains NA.

Examples

```
responses <- c(53, 62, 100, NA, 66)
score_recall(responses)
```

`score_subtraction` *Score Serial Subtraction Task*

Description

This function scores a serial subtraction task where respondents are scored based on their ability to successfully subtract a specific value (e.g., 7) from the previous value. A score of 1 is given for correct subtraction, and a score of 0 is given for incorrect subtraction. However, a respondent can still receive a score of 1 if they recover from an initial mistake by correctly subtracting later.

Usage

```
score_subtraction(val, diff)
```

Arguments

`val`	A numeric vector representing the respondent's current answer.
`diff`	A numeric vector representing the correct difference (e.g., expected subtraction value).

Details

The function checks if the respondent's answer (`val`) is equal to the correct subtraction difference (`diff`). If so, they are awarded a score of 1. If they make a mistake, they get 0. However, if they correct their mistake in the next step, they can receive a score of 1 for that step. Missing values (NA) in the input remain as NA in the output.

Value

A numeric vector where:

- `1`: Correct subtraction.
- `0`: Incorrect subtraction.
- `NA`: If the original value is missing (NA), it remains NA.

Examples

```
responses <- c(93, 86, 79, 72, NA)
correct_diffs <- c(93, 86, 79, 72, 65) - 7
score_subtraction(responses, correct_diffs)
```

Index