

Package ‘AOUSDOHtools’

March 26, 2025

Type Package

Title Analyzing AOU SDOH Survey Data

Version 1.0.4

Description Functions for processing and analyzing survey data from the All of Us Social Determinants of Health (AOUSDOH) program, including tools for calculating health and well-being scores, recoding variables, and simplifying survey data analysis. For more details see - Koleck TA, Dreisbach C, Zhang C, Grayson S, Lor M, Deng Z, Conway A, Higgins PDR, Bakken S (2024) <[doi:10.1093/jamia/ocae214](https://doi.org/10.1093/jamia/ocae214)>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr, devtools, usethis

Config/testthat/edition 3

RoxygenNote 7.3.2

URL <https://github.com/zhd52/AOUSDOHtools>

BugReports <https://github.com/zhd52/AOUSDOHtools/issues>

NeedsCompilation no

Author Zhirui Deng [aut, cre],
Theresa A. Koleck [ctb],
Caitlin Dreisbach [ctb],
Chen Zhang [ctb],
Peter D.R. Higgins [ctb],
DreisbachLab [cph]

Maintainer Zhirui Deng <zhd52@pitt.edu>

Repository CRAN

Date/Publication 2025-03-26 10:50:02 UTC

Contents

calc_cohesion	2
calc_crime_safety	3
calc_density	4
calc_disorder	5
calc_edd_chronicity	6
calc_edd_frequency	7
calc_edd_situation	8
calc_emo_support	9
calc_english_level	9
calc_english_proficient	10
calc_food_insecurity	11
calc_hcd_count	12
calc_hcd_ever	13
calc_hcd_mean	14
calc_hcd_sum	15
calc_housing_insecurity	16
calc_housing_quality	16
calc_ins_support	17
calc_loneliness	18
calc_nei	19
calc_num_moves	20
calc_other_language	21
calc_physical_disorder	21
calc_religious_attendance	22
calc_social_disorder	23
calc_social_support	24
calc_spa	25
calc_spirit	26
calc_stress_category	27
calc_stress_sum	28
Index	29

calc_cohesion	<i>Calculate Neighborhood Cohesion Score</i>
---------------	--

Description

This function computes a neighborhood cohesion score ranging from 1 to 5 based on survey responses. The score is the mean of four specific item scores, where higher scores indicate greater neighborhood cohesion.

Usage

```
calc_cohesion(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'cohesion', where 'cohesion' is the calculated cohesion score for each participant. Participants who did not answer all four questions will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 1, 2, 2, 3, 3, 4, 4),
  question_concept_id = c(40192463, 40192411, 40192463, 40192411,
    40192499, 40192417, 40192499, 40192417),
  answer_concept_id = c(40192514, 40192455, 40192524, 40192408,
    40192514, 40192524, 40192408, 40192422)
)

# Compute neighborhood cohesion scores
cohesion_scores <- calc_cohesion(survey_df)
head(cohesion_scores)
```

calc_crime_safety *Calculate Crime Safety Score*

Description

This function computes a crime safety score ranging from 1 to 4 based on survey responses. The score is the mean of two specific item scores, where higher scores indicate a greater sense of crime safety in the neighborhood.

Usage

```
calc_crime_safety(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'crime_safety', where 'crime_safety' is the calculated crime safety score for each participant. Participants who did not answer both questions will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 1, 2, 2, 3, 3, 4, 4),
  question_concept_id = c(40192414, 40192492, 40192414, 40192492,
    40192414, 40192492, 40192414, 40192492),
  answer_concept_id = c(40192514, 40192478, 40192527, 40192422,
    40192514, 40192527, 40192422, 40192478)
)

# Compute crime safety scores
crime_safety_scores <- calc_crime_safety(survey_df)
head(crime_safety_scores)
```

 calc_density

Calculate Residential Density

Description

This function creates a binary categorical variable representing residential density based on survey responses. 'Low' denotes low residential density (detached single-family housing), while 'High' denotes high residential density.

Usage

```
calc_density(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'density', where 'density' is either "High" or "Low" based on the housing type in the neighborhood. Participants with non-answers will have an NA value for 'density'.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5),
  question_concept_id = c(40192458, 40192458, 40192458, 40192458, 40192458),
  answer_concept_id = c(40192407, 40192472, 40192418, 40192433, 40192409)
)

# Compute residential density categories
```

```
density_scores <- calc_density(survey_df)
head(density_scores)
```

`calc_disorder`*Calculate Neighborhood Disorder Score*

Description

This function computes a neighborhood disorder score ranging from 1 to 4 based on survey responses. The score is the mean of 13 specific item scores, where higher scores indicate a greater sense of disorder in the neighborhood, and lower scores indicate a sense of order. Some items are reverse-coded to ensure consistency in interpretation.

Usage

```
calc_disorder(survey_df)
```

Arguments

`survey_df` A data frame containing survey data with at least three columns: ‘person_id’, ‘question_concept_id’, and ‘answer_concept_id’.

Value

A data frame with two columns: ‘person_id’ and ‘disorder’, where ‘disorder’ is the calculated neighborhood disorder score for each participant. Participants who did not answer all 13 questions will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 13),
  question_concept_id = rep(c(40192420, 40192522, 40192412, 40192469, 40192456,
                              40192386, 40192500, 40192493, 40192457, 40192476,
                              40192404, 40192400, 40192384), times = 3),
  answer_concept_id = sample(c(40192514, 40192455, 40192408, 40192422),
                             39, replace = TRUE)
)

# Compute neighborhood disorder scores
disorder_scores <- calc_disorder(survey_df)
head(disorder_scores)
```

calc_edd_chronicity *Calculate Everyday Discrimination Chronicity Score*

Description

This function computes a chronicity score indicating the total number of perceived discrimination experiences in a year. The score ranges from 0 to 2340, where higher scores indicate more frequent perceived unfair treatment. The function also allows for filtering by a specific reason for discrimination.

Usage

```
calc_edd_chronicity(survey_df, reason)
```

Arguments

survey_df	A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.
reason	An optional argument specifying the reason for perceived discrimination, e.g., race or age. If provided, the function will limit the analysis to participants who reported this reason.

Value

A data frame with two columns: 'person_id' and 'edd_chronicity', where 'edd_chronicity' is the calculated chronicity score for each participant. Participants who did not answer all 9 questions or did not specify the given reason (if provided) will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 9),
  question_concept_id = rep(c(40192380, 40192395, 40192416, 40192451, 40192466,
                             40192489, 40192490, 40192496, 40192519), times = 3),
  answer_concept_id = sample(c(40192465, 40192464, 40192453, 40192461, 40192391,
                              40192421), 27, replace = TRUE)
)

# Compute everyday discrimination chronicity scores
edd_chronicity_scores <- calc_edd_chronicity(survey_df)
head(edd_chronicity_scores)
```

calc_edd_frequency	<i>Calculate Everyday Discrimination Frequency Score</i>
--------------------	--

Description

This function computes a frequency score indicating how often participants experience perceived discrimination based on 9 specific survey items. The score ranges from 9 to 54, where higher scores indicate more frequent perceived unfair treatment. The function also allows for filtering by a specific reason for discrimination.

Usage

```
calc_edd_frequency(survey_df, reason)
```

Arguments

survey_df	A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.
reason	An optional argument specifying the reason for perceived discrimination, e.g., race or age. If provided, the function will limit the analysis to participants who reported this reason.

Value

A data frame with two columns: 'person_id' and 'edd_frequency', where 'edd_frequency' is the calculated frequency score for each participant. Participants who did not answer all 9 questions or did not specify the given reason (if provided) will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 9),
  question_concept_id = rep(c(40192380, 40192395, 40192416, 40192451, 40192466,
                             40192489, 40192490, 40192496, 40192519), times = 3),
  answer_concept_id = sample(c(40192465, 40192464, 40192453, 40192461, 40192391,
                              40192421), 27, replace = TRUE)
)

# Compute everyday discrimination frequency scores
edd_frequency_scores <- calc_edd_frequency(survey_df)
head(edd_frequency_scores)
```

calc_edd_situation *Calculate Everyday Discrimination Situation Score*

Description

This function computes a situation score indicating how many specific questions participants responded to with something other than 'Never'. The score ranges from 0 to 9, with higher scores indicating more frequent perceived experiences of unfair treatment. The function also allows for filtering by a specific reason for discrimination.

Usage

```
calc_edd_situation(survey_df, reason)
```

Arguments

survey_df	A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.
reason	An optional argument specifying the reason for perceived discrimination, e.g., race or age. If provided, the function will limit the analysis to participants who reported this reason.

Value

A data frame with two columns: 'person_id' and 'edd_situation', where 'edd_situation' is the calculated situation score for each participant. Participants who did not answer all 9 questions or did not specify the given reason (if provided) will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 9),
  question_concept_id = rep(c(40192380, 40192395, 40192416, 40192451, 40192466,
                             40192489, 40192490, 40192496, 40192519), times = 3),
  answer_concept_id = sample(c(40192465, 40192464, 40192453, 40192461, 40192391,
                              40192421), 27, replace = TRUE)
)

# Compute everyday discrimination situation scores
edd_situation_scores <- calc_edd_situation(survey_df)
head(edd_situation_scores)
```

calc_emo_support	<i>Calculate Emotional Support Score</i>
------------------	--

Description

This function computes an emotional support score on a 0-100 scale based on survey responses. The score is the mean of four specific item scores, with higher scores indicating more emotional support.

Usage

```
calc_emo_support(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'emo_support', where 'emo_support' is the calculated emotional support score for each participant. Participants who did not answer all four questions will have an NA score.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 4),
  question_concept_id = rep(c(40192399, 40192439, 40192446, 40192528), times = 3),
  answer_concept_id = sample(c(40192454, 40192518, 40192486, 40192382, 40192521),
                             12, replace = TRUE)
)

# Compute emotional support scores
emo_support_scores <- calc_emo_support(survey_df)
head(emo_support_scores)
```

calc_english_level	<i>Calculate English Proficiency Level</i>
--------------------	--

Description

This function creates an ordinal categorical variable that describes the level of proficiency in English for participants who reported speaking a language other than English at home.

Usage

```
calc_english_level(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer'.

Value

A data frame with two columns: 'person_id' and 'english_level', where 'english_level' represents the participant's self-reported proficiency in English. Participants who did not respond or provided a "PMI: Skip" will have an NA value.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5),
  question_concept_id = c(40192529, 40192529, 40192529, 40192529, 40192529),
  answer = c("Very well", "Well", "Not well", "Not at all", "Skip")
)

# Compute English proficiency levels
english_level_scores <- calc_english_level(survey_df)
head(english_level_scores)
```

calc_english_proficient

Calculate English Proficiency Category

Description

This function creates a nominal categorical variable with values 'Proficient', 'Not proficient', or 'Unknown' for participants who endorsed speaking a language other than English at home. 'Proficient' refers to participants who reported speaking English 'Very well' or 'Well', while 'Not proficient' refers to participants who reported speaking English 'Not well' or 'Not at all'.

Usage

```
calc_english_proficient(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'english_proficient', where 'english_proficient' categorizes participants as 'Proficient', 'Not proficient', or 'Unknown'. Participants who did not respond will have an NA value for 'english_proficient'.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5, 6, 7),
  question_concept_id = rep(40192529, 7),
  answer_concept_id = c(40192435, 40192510, 40192405, 40192387,
                        903087, 903079, NA)
)

# Compute English proficiency categories
english_proficient_scores <- calc_english_proficient(survey_df)
head(english_proficient_scores)
```

calc_food_insecurity *Calculate Food Insecurity Risk*

Description

This function creates a binary categorical variable (TRUE/FALSE) indicating whether a participant is at risk or currently experiencing food insecurity based on their responses to two specific survey items.

Usage

```
calc_food_insecurity(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'food_insecurity', where 'food_insecurity' is TRUE if the participant reported experiencing food insecurity, and FALSE otherwise. Participants who did not respond to both questions will have an NA value for 'food_insecurity'.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5),
  question_concept_id = rep(c(40192426, 40192517), 5),
  answer_concept_id = c(40192508, 40192488, 40192508, 903096, 903096,
                        40192488, 40192488, 40192508, 40192508)
)

# Compute food insecurity risk scores
food_insecurity_scores <- calc_food_insecurity(survey_df)
head(food_insecurity_scores)
```

<code>calc_hcd_count</code>	<i>Calculate Health Care Discrimination Count</i>
-----------------------------	---

Description

This function creates a numeric score (range 0-7) indicating how many items the participant endorsed for perceived discrimination in health care. Higher scores indicate greater perceived discrimination in health care settings.

Usage

```
calc_hcd_count(survey_df)
```

Arguments

`survey_df` A data frame containing survey data with at least three columns: ‘person_id’, ‘question_concept_id’, and ‘answer_concept_id’.

Value

A data frame with two columns: ‘person_id’ and ‘hcd_count’, where ‘hcd_count’ represents the number of health care discrimination items endorsed by the participant. Participants who did not respond to all 7 items will have an NA value.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 7),
  question_concept_id = rep(c(40192383, 40192394, 40192423, 40192425,
                            40192497, 40192503, 40192505), times = 3),
  answer_concept_id = sample(c(40192465, 40192481, 40192429, 40192382, 40192515),
                            21, replace = TRUE)
)
```

```
# Compute health care discrimination count
hcd_count_scores <- calc_hcd_count(survey_df)
head(hcd_count_scores)
```

`calc_hcd_ever`*Calculate Ever Experienced Health Care Discrimination*

Description

This function creates a binary categorical variable (TRUE/FALSE) indicating whether a participant has ever endorsed perceived discrimination in health care based on responses to seven specific survey items. TRUE indicates that the participant has experienced at least one instance of perceived discrimination.

Usage

```
calc_hcd_ever(survey_df)
```

Arguments

`survey_df` A data frame containing survey data with at least three columns: ‘person_id’, ‘question_concept_id’, and ‘answer_concept_id’.

Value

A data frame with two columns: ‘person_id’ and ‘hcd_ever’, where ‘hcd_ever’ is TRUE if the participant endorsed any form of discrimination in health care and FALSE otherwise. Participants who did not respond to all 7 items will have an NA value for ‘hcd_ever’.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 7),
  question_concept_id = rep(c(40192383, 40192394, 40192423, 40192425,
    40192497, 40192503, 40192505), times = 3),
  answer_concept_id = sample(c(40192465, 40192481, 40192429, 40192382, 40192515),
    21, replace = TRUE)
)

# Compute whether participants have ever experienced health care discrimination
hcd_ever_scores <- calc_hcd_ever(survey_df)
head(hcd_ever_scores)
```

calc_hcd_mean	<i>Calculate Mean Perceived Health Care Discrimination Score</i>
---------------	--

Description

This function creates a numeric score (range 1-5) that reflects the mean level of perceived discrimination in health care based on responses to seven specific survey items. Higher scores indicate greater perceived discrimination in health care.

Usage

```
calc_hcd_mean(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'hcd_mean', where 'hcd_mean' represents the mean level of perceived discrimination in health care. Participants who did not respond to all 7 items will have an NA value for 'hcd_mean'.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 7),
  question_concept_id = rep(c(40192383, 40192394, 40192423, 40192425,
    40192497, 40192503, 40192505), times = 3),
  answer_concept_id = sample(c(40192465, 40192481, 40192429, 40192382, 40192515),
    21, replace = TRUE)
)

# Compute mean perceived health care discrimination scores
hcd_mean_scores <- calc_hcd_mean(survey_df)
head(hcd_mean_scores)
```

calc_hcd_sum	<i>Calculate Sum of Perceived Health Care Discrimination Score</i>
--------------	--

Description

This function creates a numeric score (range 7-35) that reflects the sum of perceived discrimination in health care based on responses to seven specific survey items. Higher scores indicate greater perceived discrimination in health care.

Usage

```
calc_hcd_sum(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'hcd_sum', where 'hcd_sum' represents the sum of perceived discrimination in health care. Participants who did not respond to all 7 items will have an NA value for 'hcd_sum'.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 7),
  question_concept_id = rep(c(40192383, 40192394, 40192423, 40192425,
                             40192497, 40192503, 40192505), times = 3),
  answer_concept_id = sample(c(40192465, 40192481, 40192429, 40192382, 40192515),
                             21, replace = TRUE)
)

# Compute sum of perceived health care discrimination scores
hcd_sum_scores <- calc_hcd_sum(survey_df)
head(hcd_sum_scores)
```

`calc_housing_insecurity`*Calculate Housing Insecurity*

Description

This function creates a binary categorical variable indicating whether a participant is at risk of, or is currently experiencing, housing insecurity. Housing insecurity is defined as having moved two or more times in the past 12 months.

Usage

```
calc_housing_insecurity(survey_df)
```

Arguments

`survey_df` A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer'.

Value

A data frame with two columns: 'person_id' and 'housing_insecurity', where 'housing_insecurity' is a TRUE or FALSE indicator for each participant. TRUE indicates housing insecurity (two or more moves in the past year), and FALSE otherwise. Participants without data will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5),
  question_concept_id = rep(40192441, 5),
  answer = c("0", "1", "2", "3", "Skip")
)

# Compute housing insecurity status
housing_insecurity_scores <- calc_housing_insecurity(survey_df)
head(housing_insecurity_scores)
```

`calc_housing_quality` *Calculate Housing Quality Needs*

Description

This function creates a binary categorical variable indicating whether a participant endorses having any housing-related problems (housing need). A value of TRUE indicates the participant selected at least one housing problem, while FALSE indicates no problems were reported.

Usage

```
calc_housing_quality(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'housing_quality', where 'housing_quality' is a TRUE or FALSE indicator for each participant. TRUE indicates the participant selected at least one housing problem, and FALSE indicates no housing problems. Participants without data will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5, 6, 7),
  question_concept_id = rep(40192402, 7),
  answer_concept_id = c(40192392, 40192479, 40192444, 40192460,
                        40192434, 40192468, 40192393)
)

# Compute housing quality needs
housing_quality_scores <- calc_housing_quality(survey_df)
head(housing_quality_scores)
```

calc_ins_support	<i>Calculate Instrumental Support Score</i>
------------------	---

Description

This function computes a numeric instrumental support score ranging from 0 to 100. The score is the mean of individual item scores transformed to a 0-100 scale, where higher scores indicate greater tangible support.

Usage

```
calc_ins_support(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'ins_support', where 'ins_support' is the calculated instrumental support score for each participant. The score is scaled from 0 to 100, with higher values indicating more tangible support. Participants who did not answer all four relevant questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 4),
  question_concept_id = rep(c(40192388, 40192442, 40192480, 40192511), times = 3),
  answer_concept_id = sample(c(40192454, 40192518, 40192486, 40192382, 40192521),
                             12, replace = TRUE)
)

# Compute instrumental support scores
ins_support_scores <- calc_ins_support(survey_df)
head(ins_support_scores)
```

calc_loneliness	<i>Calculate Loneliness Score</i>
-----------------	-----------------------------------

Description

This function computes a loneliness score based on responses to 8 specific items. The score ranges from 8 to 32, with higher scores indicating a greater degree of loneliness.

Usage

```
calc_loneliness(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'loneliness', where 'loneliness' is the calculated loneliness score for each participant. The score is the sum of the individual item scores, with higher values indicating a higher degree of loneliness. Participants who did not answer all 8 questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 8),
  question_concept_id = rep(c(40192390, 40192397, 40192398, 40192494,
                             40192501, 40192504, 40192507, 40192516), times = 3),
  answer_concept_id = sample(c(40192465, 40192481, 40192429, 40192482),
                             24, replace = TRUE)
)

# Compute loneliness scores
loneliness_scores <- calc_loneliness(survey_df)
head(loneliness_scores)
```

calc_nei

Calculate Neighborhood Environment Index (NEI)

Description

This function computes a Neighborhood Environment Index (NEI) score based on 6 specific items related to the built environment for physical activity. The score ranges from 0 to 6, with higher scores indicating a more favorable environment for physical activity.

Usage

```
calc_nei(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'nei', where 'nei' is the calculated Neighborhood Environment Index score for each participant. The score is the sum of individual item scores, where higher values indicate a more favorable built environment for physical activity. Participants who did not answer all 6 questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 6),
  question_concept_id = rep(c(40192410, 40192431, 40192436, 40192437,
                             40192440, 40192458), times = 3),
  answer_concept_id = sample(c(40192527, 40192422, 40192407, 903087,
                             903096, 40192520, 40192514, 40192455),
                             18, replace = TRUE)
)
```

```
      18, replace = TRUE)
)

# Compute Neighborhood Environment Index (NEI) scores
nei_scores <- calc_nei(survey_df)
head(nei_scores)
```

calc_num_moves	<i>Calculate Number of Moves in the Past Year</i>
----------------	---

Description

This function creates a numeric variable representing the number of times a participant has moved in the past 12 months.

Usage

```
calc_num_moves(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer'.

Value

A data frame with two columns: 'person_id' and 'num_moves', where 'num_moves' represents the number of moves in the past year for each participant. Participants without data or who skipped the question will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5),
  question_concept_id = rep(40192441, 5),
  answer = c("0", "1", "2", "3", "Skip")
)

# Compute number of moves in the past year
num_moves_scores <- calc_num_moves(survey_df)
head(num_moves_scores)
```

calc_other_language *Calculate Whether Participant Speaks a Language Other Than English at Home*

Description

This function creates a nominal categorical variable with values 'Yes', 'No', or 'PMI: Prefer Not To Answer', indicating whether the participant speaks a language other than English at home.

Usage

```
calc_other_language(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer'.

Value

A data frame with two columns: 'person_id' and 'other_language', where 'other_language' contains the values 'Yes', 'No', or 'PMI: Prefer Not To Answer'. Participants without data or who skipped the question will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5),
  question_concept_id = rep(40192526, 5),
  answer = c("Yes", "No", "Yes", "Prefer Not To Answer", "Skip")
)

# Compute whether participants speak a language other than English at home
other_language_scores <- calc_other_language(survey_df)
head(other_language_scores)
```

calc_physical_disorder

Calculate Neighborhood Physical Disorder Score

Description

This function computes a numeric score representing the level of physical disorder in a participant's neighborhood. The score ranges from 1 to 4, with higher scores indicating higher physical disorder, and lower scores indicating better physical order in the neighborhood.

Usage

```
calc_physical_disorder(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'physical_disorder', where 'physical_disorder' is the average score for neighborhood physical disorder for each participant. The score is calculated as the mean of six items, with higher values indicating more physical disorder. Participants who did not answer all six questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 6),
  question_concept_id = rep(c(40192420, 40192522, 40192412,
                             40192469, 40192456, 40192386), times = 3),
  answer_concept_id = sample(c(40192514, 40192455, 40192408, 40192422),
                             18, replace = TRUE)
)

# Compute neighborhood physical disorder scores
physical_disorder_scores <- calc_physical_disorder(survey_df)
head(physical_disorder_scores)
```

calc_religious_attendance

Calculate Religious Attendance Frequency

Description

This function creates an ordinal categorical variable indicating the frequency of attending religious meetings or services, based on participant responses.

Usage

```
calc_religious_attendance(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer'.

Value

A data frame with two columns: 'person_id' and 'religious_attendance', where 'religious_attendance' indicates how often the participant attends religious meetings or services. Participants without data or who skipped the question will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = c(1, 2, 3, 4, 5),
  question_concept_id = rep(40192470, 5),
  answer = c("Never", "Once a week", "More than once a week", "Never", "Skip")
)

# Compute religious attendance frequency
religious_attendance_scores <- calc_religious_attendance(survey_df)
head(religious_attendance_scores)
```

calc_social_disorder *Calculate Neighborhood Social Disorder Score*

Description

This function computes a numeric score representing the level of social disorder in a participant's neighborhood. The score ranges from 1 to 4, with higher scores indicating higher social disorder and lower scores indicating social order.

Usage

```
calc_social_disorder(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'social_disorder', where 'social_disorder' is the average score for neighborhood social disorder for each participant. The score is calculated as the mean of seven items, with higher values indicating more social disorder. Participants who did not answer all seven questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 7),
  question_concept_id = rep(c(40192500, 40192493, 40192457, 40192476,
                             40192404, 40192400, 40192384), times = 3),
  answer_concept_id = sample(c(40192514, 40192455, 40192408, 40192422),
                             21, replace = TRUE)
)

# Compute neighborhood social disorder scores
social_disorder_scores <- calc_social_disorder(survey_df)
head(social_disorder_scores)
```

calc_social_support *Calculate Social Support Score*

Description

This function computes a numeric social support score ranging from 0 to 100. The score is based on the mean of individual item scores, transformed to a 0-100 scale, with higher scores indicating more social support.

Usage

```
calc_social_support(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'social_support', where 'social_support' is the calculated social support score for each participant. The score is scaled from 0 to 100, with higher values indicating greater social support. Participants who did not answer all eight questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 8),
  question_concept_id = rep(c(40192388, 40192399, 40192439, 40192442,
                             40192446, 40192480, 40192511, 40192528), times = 3),
  answer_concept_id = sample(c(40192454, 40192518, 40192486, 40192382, 40192521),
                             24, replace = TRUE)
```



```
)  
  
# Compute social support scores  
social_support_scores <- calc_social_support(survey_df)  
head(social_support_scores)
```

calc_spa

Calculate Environmental Support for Physical Activity (SPA) Score

Description

This function computes a numeric score representing the environmental support for physical activity (SPA). The score ranges from 7 to 28, with higher scores indicating greater environmental support for physical activity.

Usage

```
calc_spa(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'spa', where 'spa' is the sum score for environmental support for physical activity. The score is based on responses to seven items, with higher values indicating more support for physical activity. Participants who did not answer all seven questions will have NA values.

Examples

```
# Create a sample survey data frame  
survey_df <- data.frame(  
  person_id = rep(1:3, each = 7),  
  question_concept_id = rep(c(40192436, 40192440, 40192437, 40192431,  
                             40192410, 40192492, 40192414), times = 3),  
  answer_concept_id = sample(c(40192514, 40192478, 40192527, 40192422),  
                             21, replace = TRUE)  
)  
  
# Compute environmental support for physical activity (SPA) scores  
spa_scores <- calc_spa(survey_df)  
head(spa_scores)
```

`calc_spirit`*Calculate Daily Religious or Spiritual Experiences (Spirit) Score*

Description

This function computes a numeric score representing the frequency of daily religious or spiritual experiences. The score ranges from 6 to 36, with higher scores indicating more frequent daily religious or spiritual experiences.

Usage

```
calc_spirit(survey_df)
```

Arguments

`survey_df` A data frame containing survey data with at least three columns: `'person_id'`, `'question_concept_id'`, and `'answer_concept_id'`.

Value

A data frame with two columns: `'person_id'` and `'spirit'`, where `'spirit'` is the sum score for daily religious or spiritual experiences. The score is based on responses to six items, with higher values indicating more frequent experiences. Participants who did not answer all six questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 6),
  question_concept_id = rep(c(40192401, 40192415, 40192443, 40192471,
    40192475, 40192498), times = 3),
  answer_concept_id = sample(c(40192487, 40192432, 40192509, 40192459,
    40192513, 40192484, 40192385, 40192403),
    18, replace = TRUE)
)

# Compute daily religious or spiritual experiences (Spirit) scores
spirit_scores <- calc_spirit(survey_df)
head(spirit_scores)
```

`calc_stress_category` *Calculate Perceived Stress Category*

Description

This function creates an ordinal categorical variable indicating perceived stress levels as 'Low', 'Moderate', or 'High' based on participants' total perceived stress score. The perceived stress score is calculated as the sum of individual item scores, with 'Low' for scores 0-13, 'Moderate' for scores 14-26, and 'High' for scores 27-40.

Usage

```
calc_stress_category(survey_df)
```

Arguments

`survey_df` A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'stress_category', where 'stress_category' represents the stress level ('Low', 'Moderate', or 'High') for each participant. Participants who did not answer all 10 questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 10),
  question_concept_id = rep(c(40192381, 40192396, 40192419, 40192445,
    40192449, 40192452, 40192462, 40192491,
    40192506, 40192525), times = 3),
  answer_concept_id = sample(c(40192465, 40192430, 40192429, 40192477,
    40192424), 30, replace = TRUE)
)

# Compute perceived stress categories
stress_category_scores <- calc_stress_category(survey_df)
head(stress_category_scores)
```

calc_stress_sum	<i>Calculate Perceived Stress Sum Score</i>
-----------------	---

Description

This function computes a numeric perceived stress score, which ranges from 0 to 40. The score is the sum of responses to 10 specific items, where higher scores indicate higher levels of perceived stress.

Usage

```
calc_stress_sum(survey_df)
```

Arguments

survey_df A data frame containing survey data with at least three columns: 'person_id', 'question_concept_id', and 'answer_concept_id'.

Value

A data frame with two columns: 'person_id' and 'stress_sum', where 'stress_sum' represents the total perceived stress score for each participant. Participants who did not answer all 10 questions will have NA values.

Examples

```
# Create a sample survey data frame
survey_df <- data.frame(
  person_id = rep(1:3, each = 10),
  question_concept_id = rep(c(40192381, 40192396, 40192419, 40192445,
                             40192449, 40192452, 40192462, 40192491,
                             40192506, 40192525), times = 3),
  answer_concept_id = sample(c(40192465, 40192430, 40192429, 40192477,
                              40192424), 30, replace = TRUE)
)

# Compute perceived stress sum scores
stress_sum_scores <- calc_stress_sum(survey_df)
head(stress_sum_scores)
```

Index

calc_cohesion, 2
calc_crime_safety, 3
calc_density, 4
calc_disorder, 5
calc_edd_chronicity, 6
calc_edd_frequency, 7
calc_edd_situation, 8
calc_emo_support, 9
calc_english_level, 9
calc_english_proficient, 10
calc_food_insecurity, 11
calc_hcd_count, 12
calc_hcd_ever, 13
calc_hcd_mean, 14
calc_hcd_sum, 15
calc_housing_insecurity, 16
calc_housing_quality, 16
calc_ins_support, 17
calc_loneliness, 18
calc_nei, 19
calc_num_moves, 20
calc_other_language, 21
calc_physical_disorder, 21
calc_religious_attendance, 22
calc_social_disorder, 23
calc_social_support, 24
calc_spa, 25
calc_spirit, 26
calc_stress_category, 27
calc_stress_sum, 28