Package: addindicators (via r-universe)

October 15, 2024

Title The addindicators Package Focuses on Adding Indicators to a Dataset and Reviews the Added Indicator

Version 0.1.1

Description The addindicators package focuses on adding indicators such as Food Consumption Score (FCS), Household Hunger Score(HHS) etc to a dataset and reviews the added indicator.

License MIT + file LICENSE

URL https://impact-initiatives.github.io/addindicators/

Depends R (>= 2.10)

Imports dplyr, glue, magrittr, purrr, rlang, tidyr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Repository https://humanitarian-user-group.r-universe.dev

RemoteUrl https://github.com/impact-initiatives/addindicators

RemoteRef HEAD

RemoteSha dfaaee4b75566b5c1617c62b4c101a764dc22772

Contents

addindicators_analysis_by_group *Analysis by population group*

Description

Survey data

Usage

```
addindicators_analysis_by_group
```

Details

These data sets include HH data (raw and clean data) and analysis along with cleaning log.

Examples

```
addindicators_analysis_by_group
addindicators_raw_data
addindicators_cleaning_log
addindicators_clean_data
addindicators_survey
addindicators_choices
addindicators_overall_analysis
```

addindicators_choices *Choices tab of kobo tool*

Description

Choices tab of kobo tool

Usage

```
addindicators_choices
```

`addindicators_cleaning_log` *Cleaning log*

Description

Cleaning log

Usage

```
addindicators_cleaning_log
```

`addindicators_clean_data` *Clean data*

Description

Clean data

Usage

```
addindicators_clean_data
```

`addindicators_food_consumption_df` *Dataset with food consumption, household hunger Score component*

Description

Dataset with food consumption, household hunger Score component

Usage

```
addindicators_food_consumption_df
```

`addindicators_MSNA_template_data` *MSNA template dataset (example)*

Description

MSNA template dataset (example)

Usage

```
addindicators_MSNA_template_data
```

`addindicators_overall_analysis` *Nation/all population level analysis.*

Description

Nation/all population level analysis.

Usage

```
addindicators_overall_analysis
```

`addindicators_raw_data` *Raw data*

Description

Raw data

Usage

```
addindicators_raw_data
```

`addindicators_survey` *Survey tab of kobo tool*

Description

Survey tab of kobo tool

Usage

```
addindicators_survey
```

`add_eg_fclcm_phase` *Calculating FEWSNET Food Consumption-Livelihood Coping Matrix*

Description

Calculating FEWSNET Food Consumption-Livelihood Coping Matrix

Usage

```
add_eg_fclcm_phase(
  dataset,
  fc_phase_var = "fc_phase",
  fc_phase_1 = "Phase 1 FC",
  fc_phase_2 = "Phase 2 FC",
  fc_phase_3 = "Phase 3 FC",
  fc_phase_4 = "Phase 4 FC",
  fc_phase_5 = "Phase 5 FC",
  lcs_cat_var = "lcsi_cat",
  lcs_cat_none = "None",
  lcs_cat_stress = "Stress",
  lcs_cat_crisis = "Crisis",
  lcs_cat_emergency = "Emergency",
  fclcm_phase_var = "fclcm_phase"
)
```

Arguments

`dataset`	Dataset
`fc_phase_var`	Column name containing food consumption phase.
`fc_phase_1`	The name of the value "Phase 1 FC" (by default) in the food consumption phase.
`fc_phase_2`	The name of the value "Phase 2 FC" (by default) in the food consumption phase.
`fc_phase_3`	The name of the value "Phase 3 FC" (by default) in the food consumption phase.
`fc_phase_4`	The name of the value "Phase 4 FC" (by default) in the food consumption phase.

`fc_phase_5` The name of the value "Phase 5 FC" (by default) in the food consumption phase.

`lcs_cat_var` Column name containing livelihood coping category.

`lcs_cat_none` The name of the value "None" (by default) in the livelihood coping category.

`lcs_cat_stress` The name of the value "Stress" (by default) in the livelihood coping category.

`lcs_cat_crisis` The name of the value "Crisis" (by default) in the livelihood coping category.

`lcs_cat_emergency` The name of the value "Emergency" (by default) in the livelihood coping category.

`fclcm_phase_var` A character vector which will be the column name for FSLC phase.

Value

Returns a dataframe with a additional column for FCLC phase.

Examples

```
test_df <- data.frame(
  lcsi_cat = c("None", "Stress"),
  fc_phase = c("Phase 1 FC", "Phase 2 FC")
)
test_df |> add_eg_fclcm_phase()
```

`add_eg_fcm_phase` *Add the food consumption matrix to the dataset*

Description

Add the food consumption matrix to the dataset

Usage

```
add_eg_fcm_phase(
  dataset,
  fcs_column_name = "fcs_cat",
  rcsi_column_name = "rcsi_cat",
  hhs_column_name = "hhs_cat",
  fcs_categories_acceptable = "Acceptable",
  fcs_categories_poor = "Poor",
  fcs_categories_borderline = "Borderline",
  rcsi_categories_low = "No to Low",
  rcsi_categories_medium = "Medium",
  rcsi_categories_high = "High",
  hhs_categories_none = "None",
  hhs_categories_little = "Little",
  hhs_categories_moderate = "Moderate",
  hhs_categories_severe = "Severe",
  hhs_categories_very_severe = "Very Severe"
)
```

Arguments

`dataset` A dataframe

`fcs_column_name`
A string specifying the column name of the food consumption score in the dataset

`rcsi_column_name`
A string specifying the column name of the reduced coping strategy index in the dataset

`hhs_column_name`
A string specifying the column name of the household hunger scale in the dataset

`fcs_categories_acceptable`
The name of the value "Acceptable" (by default) in the fcs categories

`fcs_categories_poor`
The name of the value "Poor" (by default) in the fcs categories

`fcs_categories_borderline`
The name of the value "Borderline" (by default) in the fcs categories

`rcsi_categories_low`
The name of the value "No to Low" (by default) in the rcsi categories

`rcsi_categories_medium`
The name of the value "Medium" (by default) in the rcsi categories

`rcsi_categories_high`
The name of the value "High" (by default) in the rcsi categories

`hhs_categories_none`
The name of the value "None" (by default) in the hhs categories

`hhs_categories_little`
The name of the value "Little" (by default) in the hhs categories

`hhs_categories_moderate`
The name of the value "Moderate" (by default) in the hhs categories

`hhs_categories_severe`
The name of the value "Severe" (by default) in the hhs categories

`hhs_categories_very_severe`
The name of the value "Very Severe" (by default) in the hhs categories

Value

this function returns a dataframe with a column called fc_cell that includes values from 1 to 45 representing the Food Consumption Score Matrix and the fc_phase column that includes the different 5 phases of food consumption

Examples

```
test_data <- data.frame(
  fcs_cat = c("Acceptable", "Poor", "Borderline", "Acceptable"),
  rcsi_cat = c("No to Low", "Medium", "No to Low", "High"),
  hhs_cat = c("None", "Little", "Severe", "Very Severe")
)
add_eg_fcm_phase(test_data,
  fcs_column_name = "fcs_cat",
  rcsi_column_name = "rcsi_cat",
  hhs_column_name = "hhs_cat",
  fcs_categories_acceptable = "Acceptable",
  fcs_categories_poor = "Poor",
  fcs_categories_borderline = "Borderline",
  rcsi_categories_low = "No to Low",
  rcsi_categories_medium = "Medium",
  rcsi_categories_high = "High",
  hhs_categories_none = "None",
  hhs_categories_little = "Little",
  hhs_categories_moderate = "Moderate",
  hhs_categories_severe = "Severe",
  hhs_categories_very_severe = "Very Severe"
)
```

add_eg_fcs *add_eg_fcs*

Description

add_eg_fcs

Usage

```
add_eg_fcs(
  .dataset,
  cutoffs = c("normal", "alternative"),
  fsl_fcs_cereal = "fsl_fcs_cereal",
  fsl_fcs_legumes = "fsl_fcs_legumes",
  fsl_fcs_veg = "fsl_fcs_veg",
  fsl_fcs_fruit = "fsl_fcs_fruit",
  fsl_fcs_meat = "fsl_fcs_meat",
  fsl_fcs_dairy = "fsl_fcs_dairy",
  fsl_fcs_sugar = "fsl_fcs_sugar",
  fsl_fcs_oil = "fsl_fcs_oil"
)
```

Arguments

`.dataset` the clean dataset

`cutoffs` either "normal", or "alternative". The default is set to normal

`fsl_fcs_cereal` the name of the variable that indicates the number of days cereals were consumed

`fsl_fcs_legumes` the name of the variable that indicates the number of days legumes were consumed

`fsl_fcs_veg` the name of the variable that indicates the number of days vegetables were consumed

`fsl_fcs_fruit` the name of the variable that indicates the number of days fruits were consumed

`fsl_fcs_meat` the name of the variable that indicates the number of days meat/fish were consumed

`fsl_fcs_dairy` the name of the variable that indicates the number of days dairy were consumed

`fsl_fcs_sugar` the name of the variable that indicates the number of days cereals was consumed

`fsl_fcs_oil` the name of the variable that indicates the number of days oild were consumed

Value

the dataset with fsl_fcs_score and fsl_fcs_cat computed, as well as the 8 weighted food groups

Examples

```
df1 <- data.frame(
  fsl_fcs_cereal = c(1, 2, 3, 2, 5, 6, 7),
  fsl_fcs_legumes = c(3, 4, 5, 6, 1, 6, 5),
  fsl_fcs_veg = c(3, 2, 1, 6, 5, 4, 3),
  fsl_fcs_fruit = c(1, 4, 6, 2, 2, 2, 4),
  fsl_fcs_meat = c(5, 4, 3, 2, 7, 4, 5),
  fsl_fcs_dairy = c(1, 2, 6, 7, 3, 4, 2),
  fsl_fcs_sugar = c(1, 7, 6, 5, 2, 3, 4),
  fsl_fcs_oil = c(2, 3, 6, 5, 1, 7, 4)
)
add_eg_fcs(.dataset = df1,
  cutoffs = "normal"
)
```

`add_eg_hhs` *Add the household hunger scale to the dataset*

Description

Add the household hunger scale to the dataset

Usage

```
add_eg_hhs(
  .dataset,
  hhs_nofoodhh_1 = "fs_hhs_nofood_yn",
  hhs_nofoodhh_1a = "fs_hhs_nofood_freq",
  hhs_sleephungry_2 = "fs_hhs_sleephungry_yn",
  hhs_sleephungry_2a = "fs_hhs_sleephungry_freq",
  hhs_alldaynight_3 = "fs_hhs_daynoteating_yn",
  hhs_alldaynight_3a = "fs_hhs_daynoteating_freq",
  yes_answer = "yes",
  no_answer = "no",
  rarely_answer = "rarely_1_2",
  sometimes_answer = "sometimes_3_10",
  often_answer = "often_10_times"
)
```

Arguments

`.dataset` Dataset

`hhs_nofoodhh_1` The name of the column "In the past 4 weeks (30 days), was there ever no food to eat of any kind in your house because of lack of resources to get food?". It has to be a string.

`hhs_nofoodhh_1a` The name of the column "How often did this happen in the past (4 weeks/30 days)?". It has to be a string.

`hhs_sleephungry_2` The name of the column "In the past 4 weeks (30 days), did you or any household member go to sleep at night hungry because there was not enough food?". It has to be a string.

`hhs_sleephungry_2a` The name of the column "How often did this happen in the past (4 weeks/30 days)?". It has to be a string.

`hhs_alldaynight_3` The name of the column "In the past 4 weeks (30 days), did you or any household member go a whole day and night without eating anything at all because there was not enough food?". It has to be a string.

`hhs_alldaynight_3a` The name of the column "How often did this happen in the past (4 weeks/30 days)?". It has to be a string.

`yes_answer` Value used for "Yes"

`no_answer` Value used for the "No"

`rarely_answer` Value used for "Rarely (1-2)"

`sometimes_answer` Value used for "Sometimes (3-10)"

`often_answer` Value used for "Often (10+ times)

Value

It returns the dataframe with 12 extras columns: recoded hhs questions, score for the 3 sets of questions (from 0 to 2), the HHS score (from 0 to 6), the HHS category and the HHS IPC category

Examples

```
{
  input_data <- data.frame(
    fs_hhs_nofood_yn = c("no", "yes", "no", "no", "no"),
    fs_hhs_nofood_freq = c(NA_character_, "rarely_1_2",
                           NA_character_, NA_character_, NA_character_),
    fs_hhs_sleephungry_yn = c("no", "no", "yes", "no", "no"),
    fs_hhs_sleephungry_freq = c(NA_character_, NA_character_,
                                "often_10_times", NA_character_, NA_character_),
    fs_hhs_daynoteating_yn = c("no", "no", "yes", "yes", "yes"),
    fs_hhs_daynoteating_freq = c(NA_character_, NA_character_, "often_10_times",
                                 "rarely_1_2", "sometimes_3_10")
  )

  add_eg_hhs(
    .dataset = input_data,
    hhs_nofoodhh_1 = "fs_hhs_nofood_yn",
    hhs_nofoodhh_1a = "fs_hhs_nofood_freq",
    hhs_sleephungry_2 = "fs_hhs_sleephungry_yn",
    hhs_sleephungry_2a = "fs_hhs_sleephungry_freq",
    hhs_alldaynight_3 = "fs_hhs_daynoteating_yn",
    hhs_alldaynight_3a = "fs_hhs_daynoteating_freq",
    yes_answer = "yes",
    no_answer = "no",
    rarely_answer = "rarely_1_2",
    sometimes_answer = "sometimes_3_10",
    often_answer = "often_10_times"
  )
}
```

add_eg_lcsi *Add LCSI*

Description

Function to calculate Livelihood Coping Strategy Index (LCSI)

Usage

```
add_eg_lcsi(
  .dataset,
  lcsi_stress_vars,
  lcsi_crisis_vars,
  lcsi_emergency_vars,
  yes_val = NULL,
  no_val = NULL,
  exhausted_val = NULL,
  not_applicable_val = NULL,
  ignore_NA = FALSE
)
```

Arguments

`.dataset` A dataframe with the ten LCSI variables needed for analysis.

`lcsi_stress_vars` A vector of character values that are the column names for the four stress LCSI variables.

`lcsi_crisis_vars` A vector of character values that are the column names for the three crisis LCSI variables.

`lcsi_emergency_vars` A vector of character values that are the column names for the thre emergency LCSI variables.

`yes_val` A character value in the dataset associated with "Yes, used this coping strategy in the last 30 days."

`no_val` A character value in the dataset associated with "No, have not used this coping strategy in the last 30 days."

`exhausted_val` A character value in the dataset associated with "No, haven't used in the last 30 days because I've exhausted this coping strategy in the last 6 or 12 months."

`not_applicable_val` A character value in the dataset associated with "This coping strategy is not applicable for the household.

`ignore_NA` Default is FALSE. If set to TRUE, the missing values will be ignored.

Value

Returns a dataframe with added columns for LCSI indicators. - lcsi_x_yes : 1 means one of the of the x strategies was used (*yes_val*) - lcsi_x_exhaust: 1 means one of the x strategies was exhausted and could not be used (*exhausted_val*) - lcsi_x: 1 means one of the x strategies was if either used (*yes_val*) or exhausted (*exhausted_val*) Where x is stress, crisis or emergency - lcsi_cat_yes : the highest category between the lcsi_x_yes - lcsi_cat_exhast: the highest category between the lcsi_x_exhaust - lcsi_cat: the highest category between the lcsi_x

Examples

```
{
  input_data1 <- data.frame(
stress1 =            c("No", "No", "Exhausted", "Not Applicable", "No"),
stress2 =            c("No", "Yes", "Not Applicable", "No", "No"),
stress3 =            c("Not Applicable", "Not Applicable", "Yes", "No", "No"),
stress4 =            c("Not Applicable", "No", "Yes", "Yes", "No"),
crisis1 =            c("No", "Not Applicable", "Yes", "Exhausted", "No"),
crisis2 =             c("No", "No", "No", "No", "No"),
crisis3 =             c("No", "No", "Yes", "Not Applicable", "No"),
emergency1 =          c("No", "Not Applicable", "Not Applicable", "No", "No"),
emergency2 =          c("No", "Not Applicable", "Yes", "Not Applicable", "No"),
emergency3 =          c("Not Applicable", "No", "Not Applicable", "No", "Exhausted"))

add_eg_lcsi(.dataset = input_data1,
lcsi_stress_vars = c("stress1", "stress2", "stress3", "stress4"),
lcsi_crisis_vars = c("crisis1", "crisis2", "crisis3"),
lcsi_emergency_vars = c("emergency1", "emergency2", "emergency3"),
yes_val = "Yes",
no_val = "No",
exhausted_val = "Exhausted",
not_applicable_val = "Not Applicable")

}
```

`add_eg_rcsi` *Add indicator for reduced Household CSI Score(rcsi)*

Description

Add indicator for reduced Household CSI Score(rcsi)

Usage

```
add_eg_rcsi(
  data,
  rCSILessQlty = "rCSILessQlty",
  rCSIBorrow = "rCSIBorrow",
  rCSIMealSize = "rCSIMealSize",
  rCSIMealAdult = "rCSIMealAdult",
  rCSIMealNb = "rCSIMealNb",
  new_colname = "rcsi"
)
```

Arguments

`data` dataset

`rCSILessQlty` Column representing question- During the last 7 days, were there days (and, if so, how many) when your household had to rely on less preferred and less expensive food to cope with a lack of food or money to buy it?

`rCSIBorrow` Column representing question- During the last 7 days, were there days (and, if so, how many) when your household had to borrow food or rely on help from a relative or friend to cope with a lack of food or money to buy it?

`rCSIMealSize` Column representing question- During the last 7 days, were there days (and, if so, how many) when your household had to limit portion size of meals at meal times to cope with a lack of food or money to buy it?

`rCSIMealAdult` Column representing question- During the last 7 days, were there days (and, if so, how many) when your household had to restrict consumption by adults in order for small children to eat to cope with a lack of food or money to buy it?

`rCSIMealNb` Column representing question - During the last 7 days, were there days (and, if so, how many) when your household had to reduce number of meals eaten in a day to cope with a lack of food or money to buy it?

`new_colname` The prefix for the new columns. It has to be a string.

Value

A dataset with one additional column.

Examples

```
test_data <- data.frame(
  rCSILessQlty = c(1, 2, 3, 1),
  rCSIBorrow = c(0, 0, 3, 0),
  rCSIMealSize = c(4, 2, 6, 1),
  rCSIMealAdult = c(4, 3, 5, 0),
  rCSIMealNb = c(2, 5, NA_integer_, 1)
)
add_eg_rcsi(test_data)
```

`review_one_variable` *Review 1 column comparing it to another one and spots differences*

Description

Review 1 column comparing it to another one and spots differences

Usage

```
review_one_variable(
  dataset,
  column_to_review,
  column_to_compare_with,
  uuid_column = "uuid",
  prefix = "review",
  return_dataset = FALSE
)
```

Arguments

`dataset` A dataset to be check.

`column_to_review`
Name of the column to review.

`column_to_compare_with`
Name of the column to compare with.

`uuid_column` uuid column in the dataset. Default is uuid.

`prefix` Prefix to be used for the review and comment column. Default is "review".

`return_dataset` Logical, if the result table should be returned. Default is "FALSE".

Value

The review table, or the review table added to the results.

Examples

```
test_numeric <- data.frame(
  test = c(
    "test equality",
    "test difference",
    "test Missing in y",
    "test Missing in x",
    "test equality rounding in x",
    "test equality rounding in y",
    "test difference rounding in x",
    "test difference rounding in y"
  ),
  var_x = c(0, 1, 2, NA, 0.00019, 0.0002, 0.00035, 0.0003),
  var_y = c(0, 2, NA, 3, 0.0002, 0.00019, 0.0003, 0.00035),
  uuid = letters[1:8]
)
review_one_variable(test_numeric,
  column_to_review = "var_x",
  column_to_compare_with = "var_y"
)
```

review_variables *Review columns comparing it to another set of columns and spots differences*

Description

review_variables is a wrapper around review_one_variable

Usage

```
review_variables(
  dataset,
  columns_to_review,
  columns_to_compare_with,
  uuid_column = "uuid",
  prefix = "review"
)
```

Arguments

`dataset` A dataset to be check.

`columns_to_review`
Vectors of columns to review (should be paired with columns_to_compare_with).

`columns_to_compare_with`
Vectors of columns to compare with (should be paired with columns_to_review).

`uuid_column` uuid column in the dataset. Default is uuid.

`prefix` Prefix to be used for the review and comment column. Default is "review"

Value

A list with two objects: - the result table the review and comment columns - the review table

Examples

```
test_numeric_2_var <- data.frame(
  test = c(
    "test equality",
    "test difference",
    "test Missing in y",
    "test Missing in x",
    "test equality rounding in x",
    "test equality rounding in y",
    "test difference rounding in x",
    "test difference rounding in y"
  ),
  stat_col_one.x = c(0, 1, 2, NA, 0.00019, 0.0002, 0.00035, 0.0003),
  stat_col_two.x = c(0, 1, 2, NA, 0.00019, 0.0002, 0.00035, 0.0003),
  stat_col_one.y = c(0, 2, NA, 3, 0.0002, 0.00019, 0.0003, 0.00035),
  stat_col_two.y = c(0, 2, NA, 3, 0.0002, 0.00019, 0.0003, 0.00035),
  uuid = letters[1:8]
)

actual_results <- review_variables(test_numeric_2_var,
  columns_to_review = c("stat_col_one.x", "stat_col_two.x"),
  columns_to_compare_with = c("stat_col_one.y", "stat_col_two.y")
)
```

Index