

Package: analysistools (via r-universe)

October 9, 2026

Title Tools to perform some analysis on survey data collected with ODK

Version 0.0.0.903

Description The package provides a couple of wrappers around srvyr and tidyverse to perform some analysis primarily based on dataset collected through ODK format.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports addindicators, cleaningtools, dplyr, glue, magrittr, purrr, rlang, srvyr (>= 1.3.0), stringr, tidyr, tidyselect

Suggests covr, survey, testthat (>= 3.0.0), tibble

Remotes impact-initiatives/addindicators, impact-initiatives/cleaningtools

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL <https://impact-initiatives.github.io/analysistools/>

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libnode-dev

Repository <https://impact-initiatives.r-universe.dev>

Date/Publication 2024-11-27 13:28:20 UTC

RemoteUrl <https://github.com/impact-initiatives/analysistools>

RemoteRef HEAD

RemoteSha d748f320c41eeaa5619be45457414ffb4f53e78a

Contents

add_weights	2
analysistools_MSNA_template_choices	3

analysistools_MSNA_template_data	4
analysistools_MSNA_template_loa	4
analysistools_MSNA_template_loa_with_ratio	4
analysistools_MSNA_template_no_ratio_results_table	5
analysistools_MSNA_template_survey	5
analysistools_MSNA_template_with_ratio_results_table	5
check_loa	6
create_analysis	6
create_analysis_key_table	8
create_analysis_mean	9
create_analysis_median	10
create_analysis_prop_select_multiple	11
create_analysis_prop_select_one	12
create_analysis_ratio	13
create_loa	15
create_loa_from_results	16
review_analysis	17
unite_variables	18
verify_analysis_key	19
Index	20

add_weights	<i>Add a weight variable using the sample frame</i>
-------------	---

Description

Add a weight variable using the sample frame

Usage

```
add_weights(  
  dataset,  
  sample_data,  
  strata_column_dataset = NULL,  
  strata_column_sample = NULL,  
  population_column = NULL,  
  weight_column = "weights"  
)
```

Arguments

- dataset the clean dataframe
- sample_data sample dataframe including poplution numbers and the strata
- strata_column_dataset
 name of strata column in the clean dataframe
- strata_column_sample
 name of strata column in the sample dataframe

population_column name of population column in the sample dataframe
weight_column name of the added weight column. By default "weights"

Value

The clean dataset with 1 new column: weight

Examples

```
clean_data <- data.frame(
  uuid = c(1, 2, 3, 4, 5, 6, 7, 8),
  strata = c(
    "strata1", "strata2", "strata1",
    "strata2", "strata1", "strata2",
    "strata1", "strata1"
  )
)
sample <- data.frame(
  strata = c("strata1", "strata2"),
  population = c(30000, 50000)
)

clean_data_weighted <- clean_data %>%
  add_weights(sample,
    strata_column_dataset = "strata",
    strata_column_sample = "strata",
    population_column = "population"
  )
```

analysistools_MSNA_template_choices

MSNA template kobo choices (example)

Description

MSNA template kobo choices (example)

Usage

analysistools_MSNA_template_choices

analysistools_MSNA_template_data
MSNA template dataset (example)

Description

MSNA template dataset (example)

Usage

analysistools_MSNA_template_data

analysistools_MSNA_template_loa
Example of list of analysis

Description

Example of list of analysis

Usage

analysistools_MSNA_template_loa

analysistools_MSNA_template_loa_with_ratio
Example of list of analysis with ratio

Description

Clean data and Sample Frame

Usage

analysistools_MSNA_template_loa_with_ratio

Details

These data sets include HH data template

Examples

analysistools_MSNA_template_data

analysistools_MSNA_template_no_ratio_results_table
Analysis results example (no ratio)

Description

Analysis results example (no ratio)

Usage

analysistools_MSNA_template_no_ratio_results_table

analysistools_MSNA_template_survey
MSNA template kobo survey (example)

Description

MSNA template kobo survey (example)

Usage

analysistools_MSNA_template_survey

analysistools_MSNA_template_with_ratio_results_table
Analysis results example (with ratio)

Description

Analysis results example (with ratio)

Usage

analysistools_MSNA_template_with_ratio_results_table

check_loa	<i>Helper to check the list of analysis format</i>
-----------	--

Description

Helper to check the list of analysis format

Usage

```
check_loa(loa, design)
```

Arguments

loa	<i>a list of analysis</i>
design	Survey design object created with <code>srvyr::as_survey</code> or <code>as_survey_design</code>

Value

The loa with default values added if missing or an error if something is not correct.

Examples

```
check_loa(
  design = srvyr::as_survey(analysistools_MSNA_template_data),
  loa = analysistools_MSNA_template_loa
)
```

create_analysis	<i>Create analysis from a design</i>
-----------------	--------------------------------------

Description

If not *list of analysis (loa)* is provided, the analysis will run on the overall dataset, and all the grouping variables set.

Usage

```
create_analysis(design, loa = NULL, group_var = NULL, sm_separator = ".")
```

Arguments

design	Survey design object created with <code>srvyr::as_survey</code> or <code>as_survey_design</code>
loa	list of analysis: Default is NULL. If provided it will be used to create the analysis.
group_var	Default is NULL. If provided, it will first create a list of analysis and then will run the analysis. It should be a vector.
sm_separator	Separator for choice multiple questions. The default is "."

Details

The *loa* should contains the following columns :

- **analysis_type**: analysis type to be perform. At the moment mean, median, prop_select_one, and ratio are available.
- **analysis_var**: analysis variable to be used as string.
- **group_var**: The grouping variable as string. NA if there is no grouping variable. If a combination of grouping variable should be used together it should be 1 string character separated with a ",". i.e. c("admin1", "admin2") and "admin1, admin2" are different.
 - c("admin1", "admin2") : will perform the analysis grouping once by admin1, and once by admin2
 - "admin1, admin2" : will perform the analysis grouping once by admin1 and admin2
- **level**: confidence level to be used. If the column does not exists, .95 will be used. It can also include a column **level**, if not provided .95 will be set as default.

If ratios have to be performed, the *loa* should include the following columns as well:

- **analysis_var_numerator** **analysis_var_denominator** **numerator_NA_to_0** **filter_denominator_0**

Value

A list with 3 items:

- The results table in a long format with the analysis key
- The dataset that was used
- The list of analysis that was used

Examples

```
create_analysis(
  design = srvyr::as_survey(analysisistools_MSNA_template_data),
  loa = analysisistools_MSNA_template_loa,
  sm_separator = "/"
)
create_analysis(
  design = srvyr::as_survey(analysisistools_MSNA_template_data),
  loa = analysisistools_MSNA_template_loa_with_ratio,
  sm_separator = "/"
)
shorter_df <- analysisistools_MSNA_template_data[, c(
  "admin1",
  "admin2",
  "expenditure_debt",
  "wash_drinkingwatersource"
)]
create_analysis(
  design = srvyr::as_survey(shorter_df),
  group_var = "admin1"
)
```

```

create_analysis(
  design = srvyr::as_survey(shorter_df),
  group_var = "admin1, admin2"
)
create_analysis(
  design = srvyr::as_survey(shorter_df),
  group_var = c("admin1", "admin2")
)

```

```
create_analysis_key_table
```

Create a table from the analysis_key and results object

Description

Helper function to turn a analysis_key to a table. It will return a dataframe with analysis_var_x, analysis_var_value_x, group_var_x, group_var_value_x, nb_analysis_var, nb_group_var:

- Where YY_var_x are the variable names for the dependent and independent variables.
- Where YY_var_value_x are the values for the dependent and independent variables.
- Where x is the number of the variable by order of grouping.
- nb_xx is the number of variable used in the grouping.

Usage

```
create_analysis_key_table(results_table, analysis_key_column = "analysis_key")
```

Arguments

```

results_table  result object with an analysis key
analysis_key_column
                character vector with the name of the analysis key column. Default is "analysis
                key"

```

Value

a dataframe with the analysis key and analysis_var_x, analysis_var_value_x, group_var_x, group_var_value_x, nb_analysis_var and nb_group_var

Examples

```
create_analysis_key_table(analysisistools_MSNA_template_with_ratio_results_table$results_table)
```

create_analysis_mean *Calculate a mean from a survey*

Description

Calculate a mean from a survey

Usage

```
create_analysis_mean(design, group_var = NA, analysis_var, level = 0.95)
```

Arguments

design	design survey
group_var	dependent variable(s), variable to group by. If no dependent variable, it should be NA or empty string. If more than one variable, it should be one string with each variable separated by comma, e.g. "groupa, groupb" to group for groupa and groupb. NA is default for no grouping.
analysis_var	the independent variable, variable to summarise
level	the confidence level. 0.95 is default

Value

a data frame with the mean for each group

Examples

```
somedata <- data.frame(
  aa = 1:10,
  bb = rep(c("a", "b"), 5),
  weights = rep(c(.5, 1.5), 5)
)
me_design <- srvyr::as_survey(somedata)
create_analysis_mean(me_design, analysis_var = "aa")
create_analysis_mean(me_design, group_var = "bb", analysis_var = "aa")

me_design_w <- srvyr::as_survey(somedata, weights = weights)
create_analysis_mean(me_design_w, analysis_var = "aa")
create_analysis_mean(me_design_w, group_var = "bb", analysis_var = "aa")
```

```
create_analysis_median
```

Calculate a median from a survey

Description

Calculate a median from a survey

Usage

```
create_analysis_median(design, group_var = NA, analysis_var, level = 0.95)
```

Arguments

design	design survey
group_var	dependent variable(s), variable to group by. If no dependent variable, it should be NA or empty string. If more than one variable, it should be one string with each variable separated by comma, e.g. "groupa, groupb" to group for groupa and groupb. NA is default for no grouping.
analysis_var	the independent variable, variable to summarise
level	the confidence level. 0.95 is default

Value

a data frame with the median for each group

Note

Default from *survey/srvyr* is "math" methodology. In case of odds number, it will return the lower value. Default for *stats::median* will calculate the mean between the two points. If there is a set of *c(1,2)*, *median(1,2)* will return 1.5; *survey_mean()* will return 1 by default. *create_analysis_median* has the "school" methodology set as default, the results will match the default results from *stats::median*, *pandas.median* If want to calculate with the "math" methodology, you should run your own analysis with *survey_median*.

Examples

```
somedata <- data.frame(
  aa = 1:10,
  bb = rep(c("a", "b"), 5),
  weights = rep(c(.5, 1.5), 5)
)
me_design <- srvyr::as_survey(somedata)
create_analysis_median(me_design, analysis_var = "aa")
create_analysis_median(me_design, group_var = "bb", analysis_var = "aa")

me_design_w <- srvyr::as_survey(somedata, weights = weights)
```

```
create_analysis_median(me_design_w, analysis_var = "aa")
create_analysis_median(me_design_w, group_var = "bb", analysis_var = "aa")
```

```
create_analysis_prop_select_multiple
```

Calculate proportion from a survey design

Description

Calculate proportion from a survey design

Usage

```
create_analysis_prop_select_multiple(
  design,
  group_var = NA,
  analysis_var,
  level = 0.95,
  sm_separator = "."
)
```

Arguments

design	design survey
group_var	dependent variable(s), variable to group by. If no dependent variable, it should be NA or empty string. If more than one variable, it should be one string with each variable separated by comma, e.g. "groupa, groupb" to group for groupa and groupb. NA is default for no grouping.
analysis_var	the independent variable, variable to summarise. It has to be the parent select multiple
level	the confidence level. 0.95 is default
sm_separator	Select multiple separator. Default is "."

Value

a data frame with the proportion for each group and the analysis index.

Examples

```
somedata <- data.frame(
  groups = sample(c("group_a", "group_b"), size = 100, replace = TRUE),
  smvar = rep(NA_character_, 100),
  smvar.option1 = sample(c(TRUE, FALSE), size = 100, replace = TRUE, prob = c(.7, .3)),
  smvar.option2 = sample(c(TRUE, FALSE), size = 100, replace = TRUE, prob = c(.6, .4)),
  smvar.option3 = sample(c(TRUE, FALSE), size = 100, replace = TRUE, prob = c(.1, .9)),
  smvar.option4 = sample(c(TRUE, FALSE), size = 100, replace = TRUE, prob = c(.8, .2)),
```

```

  uuid = 1:100 %>% as.character()
) %>%
  cleaningtools::recreate_parent_column(uuid = "uuid", sm_separator = ".")

somedata <- somedata$data_with_fix_concat

create_analysis_prop_select_multiple(srvyr::as_survey(somedata),
  group_var = NA,
  analysis_var = "smvar",
  level = 0.95
)

create_analysis_prop_select_multiple(srvyr::as_survey(somedata),
  group_var = "groups",
  analysis_var = "smvar",
  level = 0.95
)

```

```
create_analysis_prop_select_one
```

Calculate proportion from a survey

Description

Calculate proportion from a survey

Usage

```

create_analysis_prop_select_one(
  design,
  group_var = NA,
  analysis_var,
  level = 0.95
)

```

Arguments

design	design survey
group_var	dependent variable(s), variable to group by. If no dependent variable, it should be NA or empty string. If more than one variable, it should be one string with each variable separated by comma, e.g. "groupa, groupb" to group for groupa and groupb. NA is default for no grouping.
analysis_var	the independent variable, variable to summarise
level	the confidence level. 0.95 is default

Value

a data frame with the proportion for each group and th analysis index.

Examples

```

somedata <- data.frame(
  groups = sample(c("group_a", "group_b"),
    size = 100,
    replace = TRUE
  ),
  value = sample(c("a", "b", "c"),
    size = 100, replace = TRUE,
    prob = c(.6, .4, .1)
  )
)
create_analysis_prop_select_one(srvyr::as_survey(somedata, strata = groups),
  group_var = NA,
  analysis_var = "value",
  level = .95
)
create_analysis_prop_select_one(srvyr::as_survey(somedata, strata = groups),
  group_var = "groups",
  analysis_var = "value",
  level = .95
)

```

create_analysis_ratio *Calculate a ratio from a survey*

Description

The function will calculate the ratio between 2 variables, the numerator and denominator. The numerator : By default, it will change all NAs to 0. If numerator_NA_to_0 is set to FALSE, rows with missing values will be filtered out.

The denominator : All rows with missing value will be filtered out (cannot be changed). In addition, by default, all rows with a value equal to 0 will be filtered out, if filter_denominator_0 is set to TRUE, they will be kept.

Usage

```

create_analysis_ratio(
  design,
  group_var = NA,
  analysis_var_numerator,
  analysis_var_denominator,
  numerator_NA_to_0 = TRUE,
  filter_denominator_0 = TRUE,
  level = 0.95
)

```

Arguments

design	design survey
group_var	dependent variable(s), variable to group by. If no dependent variable, it should be NA or empty string. If more than one variable, it should be one string with each variable separated by comma, e.g. "groupa, groupb" to group for groupa and groupb. NA is default for no grouping.
analysis_var_numerator	character string with the numerator column name.
analysis_var_denominator	character string with the denominator column name.
numerator_NA_to_0	Will turn all NA of the numerator into 0's, default TRUE.
filter_denominator_0	Will remove all rows with 0's in the denominator, default TRUE.
level	the confidence level. 0.95 is default

Value

a data frame with the ratio for each group

Examples

```

school_ex <- data.frame(
  hh = c("hh1", "hh2", "hh3", "hh4"),
  num_children = c(3, 0, 2, NA),
  num_enrolled = c(3, NA, 0, NA),
  num_attending = c(1, NA, NA, NA),
  group = c("a", "a", "b", "b")
)
me_design <- srvyr::as_survey(school_ex)
# Default value will give a ratio of 0.2 as there are 1 child out of 5 attending school.
# numerator: 1 child from hh1 and 0 from hh3. denominator: 3 from hh1 and 2 from hh3. In the
# hh3, the num_attending is NA because there is a skip logic, there cannot be a child attending as
# none are enrolled. By default, the function has the argument numerator_NA_to_0 set to TRUE to
# turn that NA into a 0.
# n and n_total are 2 as 2 households were included in the calculation. hh2 was not included in
# the calculation of totals. The argument filter_denominator_0 set to TRUE removes that row.
create_analysis_ratio(me_design,
  analysis_var_numerator = "num_attending",
  analysis_var_denominator = "num_children"
)
# If numerator_NA_to_0 is set to FALSE, ratio will be 1/3, as hh3 with 2 children and NA for
# attending will be removed with the na.rm = T inside the survey_ratio calculation.
# n and n_total is 1 as only 1 household was used.
create_analysis_ratio(me_design,
  analysis_var_numerator = "num_attending",
  analysis_var_denominator = "num_children",
  numerator_NA_to_0 = FALSE
)

```

```

# If filter_denominator_0 is set to FALSE, ratio will be 0.2 as there are 1 child out of 5
# attending school.
# The number of household counted, n and n_total, is equal to 3 instead 2. The household with 0
# child is counted in the totals.
# (01 + 0 + 0) / (3 + 0 + 2)
create_analysis_ratio(me_design,
                      analysis_var_numerator = "num_attending",
                      analysis_var_denominator = "num_children",
                      filter_denominator_0 = FALSE
)
# For weights and group:
set.seed(8988)
somedata <- data.frame(
  groups = rep(c("a", "b"), 50),
  children_518 = sample(0:5, 100, replace = TRUE),
  children_enrolled = sample(0:5, 100, replace = TRUE)
) %>%
  dplyr::mutate(children_enrolled = ifelse(children_enrolled > children_518,
                                          children_518,
                                          children_enrolled
))
somedata[["weights"]] <- ifelse(somedata$groups == "a", 1.33, .67)
create_analysis_ratio(srvyr::as_survey(somedata, weights = weights, strata = groups),
                      group_var = NA,
                      analysis_var_numerator = "children_enrolled",
                      analysis_var_denominator = "children_518",
                      level = 0.95
)
create_analysis_ratio(srvyr::as_survey(somedata, weights = weights, strata = groups),
                      group_var = "groups",
                      analysis_var_numerator = "children_enrolled",
                      analysis_var_denominator = "children_518",
                      level = 0.95
)

```

create_loa

Create a list of analysis from design and a grouping variable

Description

Helper for the create_analysis. It will take all the columns that are character, logical, double or integers.

Usage

```
create_loa(design, group_var = NULL, sm_separator = ".")
```

Arguments

- | | |
|--------------|---|
| design | Survey design object created with <code>srvyr::as_survey</code> or <code>as_survey_design</code> |
| group_var | The grouping variable as string. If a combination of grouping variable should be used together it should be 1 string character separated with a ",". i.e. <code>c("admin1", "population")</code> and <code>"admin1, population"</code> are different: <ul style="list-style-type: none"> • <code>c("admin1", "population")</code> : will perform the analysis grouping once by admin1, and once by population • <code>"admin1, population"</code> : will perform the analysis grouping once by admin1 and admin 2 |
| sm_separator | Separator for choice multiple questions. The default is "." |

Value

a list of analysis.

Examples

```
shorter_df <- analysistools_MSNA_template_data[, c(
  "admin1",
  "expenditure_debt",
  "wash_drinkingwatersource"
)]

create_loa(
  design = srvyr::as_survey(shorter_df),
  group_var = "admin1"
)
```

```
create_loa_from_results
```

Create a list of analysis from a results table

Description

Create a list of analysis from a results table

Usage

```
create_loa_from_results(results_table, analysis_key_column = "analysis_key")
```

Arguments

- | | |
|---------------------|--|
| results_table | Results table with an analysis key |
| analysis_key_column | character vector with the name of the analysis key column. Default is "analysis key" |

Value

a list of analysis

Examples

```
create_loa_from_results(analysis_tools_MSNA_template_with_ratio_results_table$results_table)
```

review_analysis	<i>Review columns comparing it to another set of columns and spots differences</i>
-----------------	--

Description

Wrapper around `addindicators::review_variables` specific to review analysis.

Usage

```
review_analysis(
  results_table,
  stat_columns_to_review = "stat.x",
  stat_columns_to_compare_with = "stat.y",
  analysis_key_column = "analysis_key",
  prefix = "review"
)
```

Arguments

`results_table` Results table with an analysis key

`stat_columns_to_review` Vectors of columns to review (should be paired with `columns_to_compare_with`). Default is "stat.x".

`stat_columns_to_compare_with` Vectors of columns to compare with (should be paired with `columns_to_review`). Default is "stat.y"

`analysis_key_column` character vector with the name of the analysis key column. Default is "analysis key"

`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

```
analysis_key_column <- c("mean @/ @ income %/% NA @/ @ NA %/% NA",
  "prop_select_one @/ @ water_source %/% tap_water @/ @ district %/% district_a",
  "prop_select_one @/ @ water_source %/% tap_water @/ @ district %/% district_a -/- population %/% di",
  "prop_select_multiple @/ @ source_information %/% relatives @/ @ NA %/% NA",
  "ratio @/ @ food_expenses %/% NA -/- total_expenses %/% NA @/ @ NA %/% NA",
  "prop_select_one @/ @ water_source %/% tap_water @/ @ population %/% displaced",
  "ratio @/ @ food_expenses %/% NA -/- total_expenses %/% NA @/ @ district %/% district_a",
  "prop_select_one @/ @ water_source %/% tap_water @/ @ population %/% returnees")
test_analysis_results <- 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.x = c(0, 1, 2, NA, 0.00019, 0.0002, 0.00035, 0.0003),
  upp_col.x = c(0, 1, 2, NA, 0.00019, 0.0002, 0.00035, 0.0003),
  stat_col.y = c(0, 2, NA, 3, 0.0002, 0.00019, 0.0003, 0.00035),
  upp_col.y = c(0, 2, NA, 3, 0.0002, 0.00019, 0.0003, 0.00035),
  analysis_key = analysis_key_column
)

review_analysis(test_analysis_results,
  stat_columns_to_review = "stat_col.x",
  stat_columns_to_compare_with = "stat_col.y")
```

unite_variables

Unite variable from the key_table

Description

Function will unite all analysis_var, analysis_var_value, group_var, and group_var_value from an key table resulting from create_analysis_key_table.

Usage

```
unite_variables(key_table)
```

Arguments

key_table a key table built with create_analysis_key_table

Value

a table with analysis_var, analysis_var_value, group_var, and group_var_value united and with a %/% as separator

Examples

```
## Not run:  
unite_variables(key_table)  
  
## End(Not run)
```

verify_analysis_key *Helper to check the format of the analysis key*

Description

Helper to check the format of the analysis key

Usage

```
verify_analysis_key(analysis_key)
```

Arguments

analysis_key vector of analysis key (not the name of the column)

Value

It will stop the function if the analysis is not in the correct format

Examples

```
verify_analysis_key(analysistools_MSNA_template_no_ratio_results_table$results_table$analysis_key)
```

Index

- * **datasets**
 - analysistools_MSNA_template_choices,
 - [3](#)
 - analysistools_MSNA_template_data,
 - [4](#)
 - analysistools_MSNA_template_loa, [4](#)
 - analysistools_MSNA_template_loa_with_ratio,
 - [4](#)
 - analysistools_MSNA_template_no_ratio_results_table,
 - [5](#)
 - analysistools_MSNA_template_survey,
 - [5](#)
 - analysistools_MSNA_template_with_ratio_results_table,
 - [5](#)
- unite_variables, [18](#)
- verify_analysis_key, [19](#)
- add_weights, [2](#)
- analysistools_MSNA_template_choices, [3](#)
- analysistools_MSNA_template_data, [4](#)
- analysistools_MSNA_template_loa, [4](#)
- analysistools_MSNA_template_loa_with_ratio,
 - [4](#)
- analysistools_MSNA_template_no_ratio_results_table,
 - [5](#)
- analysistools_MSNA_template_survey, [5](#)
- analysistools_MSNA_template_with_ratio_results_table,
 - [5](#)
- check_loa, [6](#)
- create_analysis, [6](#)
- create_analysis_key_table, [8](#)
- create_analysis_mean, [9](#)
- create_analysis_median, [10](#)
- create_analysis_prop_select_multiple,
 - [11](#)
- create_analysis_prop_select_one, [12](#)
- create_analysis_ratio, [13](#)
- create_loa, [15](#)
- create_loa_from_results, [16](#)
- review_analysis, [17](#)