

Package ‘stepssurvey’

May 11, 2026

Title Analyse WHO STEPS Survey Data

Version 0.1.0

Description Provides a complete analysis pipeline for the WHO STEPwise Approach to NCD Risk Factor Surveillance (STEPS) as described in Riley et al. (2016) <[doi:10.2105/AJPH.2015.302962](https://doi.org/10.2105/AJPH.2015.302962)>. Imports raw survey data ('CSV', 'Excel', 'Stata', 'SPSS'), applies WHO-standard cleaning and recoding, sets up complex survey designs, computes all standard NCD indicators (tobacco, alcohol, diet, physical activity, anthropometry, blood pressure, biochemical), and generates publication-ready tables, visualisations, and 'Word'/'HTML' reports (fact sheet, data book, country report).

License MIT + file LICENSE

URL <https://github.com/drpakhare/stepssurvey>

BugReports <https://github.com/drpakhare/stepssurvey/issues>

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports bslib, dplyr, DT, flextable, ggplot2, glue, haven, janitor, patchwork, purrr, readr, readxl, rmarkdown, shiny, survey, tools

Suggests knitr, remotes, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Abhijit Pakhare [aut, cre],
Ankur Joshi [aut],
Lena Charlette [aut],
WHO STEPS R Pipeline Contributors [ctb]

Maintainer Abhijit Pakhare <abhijit.cfm@aiimsbhopal.edu.in>

Repository CRAN

Date/Publication 2026-05-11 19:10:03 UTC

Contents

stepssurvey-package	3
build_all_tables	3
build_forest_plot	4
build_radar_plot	4
build_steps_plots	5
build_steps_tables	6
build_table	7
clean_steps_data	7
compute_alcohol_indicators	9
compute_all_indicators	10
compute_all_tables	11
compute_anthropometry_indicators	11
compute_biochemical_indicators	12
compute_bp_indicators	14
compute_diet_pa_indicators	15
compute_engine	16
compute_table	16
compute_tobacco_indicators	17
detect_col	18
detect_steps_columns	18
generate_test_data	19
get_registry_by_section	20
get_registry_by_step	21
import_steps_data	21
list_registry_sections	22
plot_completeness	22
plot_digit_preference	23
plot_weights	23
read_column_mapping	24
render_country_report	25
render_data_book	26
render_fact_sheet	26
run_app	27
run_steps_pipeline	28
save_steps_plots	29
setup_survey_design	30
steps_colors	31
steps_config	31
steps_data_quality	33
steps_table_registry	34
svymn	35
svyprop	36
table_builder	36
theme_steps	37

stepssurvey-package *stepssurvey: Analyse WHO STEPS Survey Data*

Description

A complete analysis pipeline for the WHO STEPwise Approach to NCD Risk Factor Surveillance (STEPS).

Author(s)

Maintainer: Abhijit Pakhare <abhijit.cfm@aaimsbhopal.edu.in>

Authors:

- Ankur Joshi
- Lena Charlette

Other contributors:

- WHO STEPS R Pipeline Contributors [contributor]

See Also

Useful links:

- <https://github.com/drpakhare/stepssurvey>
- Report bugs at <https://github.com/drpakhare/stepssurvey/issues>

build_all_tables *Build all tables from computed results*

Description

Build all tables from computed results

Usage

```
build_all_tables(results)
```

Arguments

results A named list of results from `compute_all_tables()`.

Value

A named list of flextable objects. NULL entries excluded.

build_forest_plot	<i>Build forest plot of key indicators with 95% CIs</i>
-------------------	---

Description

Creates a horizontal point-and-CI plot (forest plot style) for all key indicators, grouped by domain.

Usage

```
build_forest_plot(key_indicators, country_name, survey_year)
```

Arguments

key_indicators A data frame with domain, indicator, estimate, lower, upper.
country_name Country name for title.
survey_year Survey year for title.

Value

A ggplot2 object.

build_radar_plot	<i>Build radar / spider chart of NCD risk factor profile</i>
------------------	--

Description

Creates a radar-style chart showing prevalence of key risk factors on a polar coordinate system for quick visual comparison.

Usage

```
build_radar_plot(key_indicators, country_name, survey_year)
```

Arguments

key_indicators A data frame with domain, indicator, estimate.
country_name Country name for title.
survey_year Survey year for title.

Value

A ggplot2 object.

build_steps_plots	<i>Build publication-ready STEPS visualizations</i>
-------------------	---

Description

Generates a list of ggplot2 plots showing key NCD risk factor prevalence with 95% confidence intervals, stratified by sex and age group.

Usage

```
build_steps_plots(indicators, key_indicators, country_name, survey_year)
```

Arguments

indicators	A list of indicator results from <code>compute_all_indicators()</code> .
key_indicators	A data frame with key indicators (domain, indicator, estimate, lower, upper).
country_name	Country name for plot titles.
survey_year	Survey year for plot titles.

Details

All plots use the WHO STEPS colour scheme and professional styling. Error bars represent 95% confidence intervals. Prevalence values are displayed on bars/points with light background text.

Value

A named list of ggplot2 objects:

- `overview`: Horizontal bar chart of key indicators
- `tobacco_by_sex`: Sex-stratified tobacco use
- `bp_by_sex`: Sex-stratified blood pressure
- `obesity_by_sex`: Sex-stratified overweight/obesity
- `glucose_by_sex`: Sex-stratified blood glucose
- `bp_by_age`: Age-stratified blood pressure with ribbon CI
- `obesity_by_age`: Age-stratified overweight/obesity with ribbon CI
- `sex_dashboard`: Combined 2x2 dashboard of sex-stratified charts (if ≥ 2 sex plots available)
NULL entries are preserved in the list.

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
all_ind <- compute_all_indicators(design)
plots <- build_steps_plots(all_ind$results, all_ind$key_indicators, "Test", 2023)
names(plots)
```

build_steps_tables *Build survey-weighted tables for STEPS indicators*

Description

Generates formatted flextable objects for all available STEPS indicators, with rows for age groups and columns for both sexes combined, males, and females. Tables include 95% confidence intervals.

Usage

```
build_steps_tables(indicators)
```

Arguments

indicators A list of indicator results from `compute_all_indicators()`, containing elements like tobacco, alcohol, diet_pa, anthropometry, blood_pressure, biochemical, etc. Each indicator list should contain `*_total`, `*_by_sex`, and `*_by_age` elements.

Details

Each table has age groups as rows and prevalence (with 95% CI) as a column. The last row shows the total (age-standardised) estimate. Column header styling uses WHO STEPS branding (dark blue background).

Value

A named list of flextable objects, one per indicator. Names correspond to indicators (e.g., `current_tobacco`, `raised_bp`). NULL entries are excluded. Prints count of tables generated.

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
all_ind <- compute_all_indicators(design)
```

```
tables <- build_steps_tables(all_ind$results)
names(tables)
```

build_table	<i>Build a formatted table from a computed result</i>
-------------	---

Description

Dispatches to the appropriate formatting method based on table type.

Usage

```
build_table(result)
```

Arguments

result A result list from [compute_table\(\)](#).

Value

A flextable object, or NULL if the table is not available.

clean_steps_data	<i>Clean and recode WHO STEPS data</i>
------------------	--

Description

Processes raw STEPS survey data: renames columns, coerces types, derives standard indicators, handles missing values, and applies plausibility checks.

Usage

```
clean_steps_data(  
  data,  
  cols,  
  age_min = 18,  
  age_max = 69,  
  bp_sbp_threshold = 140,  
  bp_dbp_threshold = 90,  
  bmi_overweight = 25,  
  bmi_obese = 30,  
  glucose_threshold = 7,  
  glucose_impaired_threshold = 6.1,  
  chol_threshold = 5  
)
```

Arguments

<code>data</code>	A data frame (typically from <code>import_steps_data()</code>).
<code>cols</code>	A named list of column names, as returned by <code>detect_steps_columns()</code> .
<code>age_min</code>	Minimum age for inclusion (default 18).
<code>age_max</code>	Maximum age for inclusion (default 69).
<code>bp_sbp_threshold</code>	SBP threshold for raised BP (default 140; Mongolia uses 130).
<code>bp_dbp_threshold</code>	DBP threshold for raised BP (default 90; Mongolia uses 80).
<code>bmi_overweight</code>	BMI threshold for overweight (default 25.0).
<code>bmi_obese</code>	BMI threshold for obesity (default 30.0).
<code>glucose_threshold</code>	Fasting glucose threshold for raised glucose / diabetes in mmol/L (default 7.0).
<code>glucose_impaired_threshold</code>	Fasting glucose threshold for impaired fasting glucose in mmol/L (default 6.1).
<code>chol_threshold</code>	Total cholesterol threshold for raised cholesterol in mmol/L (default 5.0).

Details

The function performs the following transformations:

- Renames columns to standard names (age, sex, wt_final, etc.)
- Converts numeric strings to appropriate types
- Restricts age to [age_min, age_max]
- Creates WHO standard age groups (18-24, 25-34, etc.)
- Harmonises sex coding to Male/Female
- Derives body mass index (BMI) and categories
- Averages blood pressure readings (last 2 of 3)
- Recodes yes/no variables to logical
- Creates derived risk indicators (raised BP, diabetes, etc.)
- Applies plausibility checks to measurements
- Drops records with missing age or sex

Value

A data frame with standardised and derived variables, ready for survey design setup.

`compute_alcohol_indicators`*Compute Alcohol Use Indicators*

Description

Calculates prevalence of alcohol use from a survey design object. Computes proportions of current alcohol use and heavy episodic drinking, stratified by sex and age group where available.

Usage

```
compute_alcohol_indicators(design)
```

Arguments

`design` A survey design object from [setup_survey_design\(\)](#).

Value

A named list of survey estimates. Each element contains proportion estimates (as tibble with columns: estimate, lower, upper, etc.) for:

- `current_alcohol_total`: current alcohol use, overall
- `current_alcohol_by_sex`: current alcohol use, by sex
- `current_alcohol_by_age`: current alcohol use, by age group
- `heavy_episodic_total`: heavy episodic drinking, overall
- `heavy_episodic_by_sex`: heavy episodic drinking, by sex
- `heavy_episodic_by_age`: heavy episodic drinking, by age group (if the corresponding variables exist in design)

See Also

[compute_all_indicators\(\)](#)

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
alcohol_results <- compute_alcohol_indicators(design)
```

`compute_all_indicators`*Compute All STEPS Indicators*

Description

Runs all indicator modules (tobacco, alcohol, diet & physical activity, anthropometry, blood pressure, and biochemical), using the appropriate step-specific survey design for each domain per WHO STEPS methodology:

- Step 1 (behavioural): tobacco, alcohol, diet & physical activity
- Step 2 (physical): anthropometry, blood pressure
- Step 3 (biochemical): biochemical measures

Usage

```
compute_all_indicators(design)
```

Arguments

`design` A `steps_designs` list from `setup_survey_design()` (with elements `$step1`, `$step2`, `$step3`), or a single `survey::svydesign` object for backward compatibility.

Value

A list with two elements:

- `results`: a named list containing indicator results grouped by domain (tobacco, alcohol, diet_pa, anthropometry, blood_pressure, biochemical)
- `key_indicators`: a tibble with columns `domain`, `indicator`, `estimate`, `lower`, and `upper`, summarising headline estimates across all domains

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
all_indicators <- compute_all_indicators(design)
names(all_indicators$results)
```

compute_all_tables	<i>Compute all tables from the registry</i>
--------------------	---

Description

Iterates through the full `steps_table_registry()` and computes every table that has available data. Returns a named list of results.

Usage

```
compute_all_tables(designs, data = NULL)
```

Arguments

designs	A list of survey designs, with elements step1, step2, step3 (as returned by <code>setup_survey_design()</code>).
data	The cleaned data frame.

Value

A named list of table results (from `compute_table()`). Only entries with `available == TRUE` are included.

compute_anthropometry_indicators	<i>Compute Anthropometry Indicators</i>
----------------------------------	---

Description

Calculates prevalence of overweight, obesity, and central obesity, plus mean BMI and waist circumference, from a survey design object.

Usage

```
compute_anthropometry_indicators(design)
```

Arguments

design	A survey design object from <code>setup_survey_design()</code> .
--------	--

Value

A named list of survey estimates. Each element contains estimates (as tibble with columns: estimate, lower, upper, etc.) for:

- `overweight_obese_total`: overweight or obese (BMI ≥ 25), overall
- `overweight_obese_by_sex`: overweight or obese, by sex
- `overweight_obese_by_age`: overweight or obese, by age group
- `obese_total`: obese (BMI ≥ 30), overall
- `obese_by_sex`: obese, by sex
- `obese_by_age`: obese, by age group
- `central_obesity_total`: central obesity, overall
- `central_obesity_by_sex`: central obesity, by sex
- `central_obesity_by_age`: central obesity, by age group
- `bmi_mean_total`: mean BMI, overall
- `bmi_mean_by_sex`: mean BMI, by sex
- `waist_cm_mean_total`: mean waist circumference, overall
- `waist_cm_mean_by_sex`: mean waist circumference, by sex (if the corresponding variables exist in design)

See Also

[compute_all_indicators\(\)](#)

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
anthropometry_results <- compute_anthropometry_indicators(design)
```

compute_biochemical_indicators

Compute Biochemical Indicators

Description

Calculates prevalence of raised glucose, diabetes, impaired glucose tolerance, and raised cholesterol, plus mean fasting glucose and total cholesterol from a survey design object.

Usage

```
compute_biochemical_indicators(design)
```

Arguments

design A survey design object from [setup_survey_design\(\)](#).

Value

A named list of survey estimates. Each element contains estimates (as tibble with columns: estimate, lower, upper, etc.) for:

- raised_glucose_total: raised fasting glucose, overall
- raised_glucose_by_sex: raised fasting glucose, by sex
- raised_glucose_by_age: raised fasting glucose, by age group
- diabetes_total: diabetes, overall
- diabetes_by_sex: diabetes, by sex
- diabetes_by_age: diabetes, by age group
- impaired_glucose_total: impaired fasting glucose, overall
- impaired_glucose_by_sex: impaired fasting glucose, by sex
- impaired_glucose_by_age: impaired fasting glucose, by age group
- raised_chol_total: raised total cholesterol, overall
- raised_chol_by_sex: raised total cholesterol, by sex
- raised_chol_by_age: raised total cholesterol, by age group
- fasting_glucose_mean_total: mean fasting glucose, overall
- fasting_glucose_mean_by_sex: mean fasting glucose, by sex
- total_chol_mean_total: mean total cholesterol, overall
- total_chol_mean_by_sex: mean total cholesterol, by sex (if the corresponding variables exist in design)

See Also

[compute_all_indicators\(\)](#)

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
biochemical_results <- compute_biochemical_indicators(design)
```

compute_bp_indicators *Compute Blood Pressure Indicators*

Description

Calculates prevalence of raised blood pressure and mean systolic and diastolic blood pressure from a survey design object.

Usage

```
compute_bp_indicators(design)
```

Arguments

design A survey design object from [setup_survey_design\(\)](#).

Value

A named list of survey estimates. Each element contains estimates (as tibble with columns: estimate, lower, upper, etc.) for:

- raised_bp_total: raised blood pressure, overall
- raised_bp_by_sex: raised blood pressure, by sex
- raised_bp_by_age: raised blood pressure, by age group
- mean_sbp_mean_total: mean systolic BP, overall
- mean_sbp_mean_by_sex: mean systolic BP, by sex
- mean_sbp_mean_by_age: mean systolic BP, by age group
- mean_dbp_mean_total: mean diastolic BP, overall
- mean_dbp_mean_by_sex: mean diastolic BP, by sex
- mean_dbp_mean_by_age: mean diastolic BP, by age group (if the corresponding variables exist in design)

See Also

[compute_all_indicators\(\)](#)

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
bp_results <- compute_bp_indicators(design)
```

`compute_diet_pa_indicators`*Compute Diet and Physical Activity Indicators*

Description

Calculates prevalence of insufficient physical activity and low fruit & vegetable intake, plus mean metabolic equivalent (MET) values, from a survey design object.

Usage

```
compute_diet_pa_indicators(design)
```

Arguments

`design` A survey design object from `setup_survey_design()`.

Value

A named list of survey estimates. Each element contains estimates (as tibble with columns: estimate, lower, upper, etc.) for:

- `insufficient_pa_total`: insufficient physical activity, overall
- `insufficient_pa_by_sex`: insufficient physical activity, by sex
- `insufficient_pa_by_age`: insufficient physical activity, by age group
- `low_fruit_veg_total`: low fruit & vegetable intake, overall
- `low_fruit_veg_by_sex`: low fruit & vegetable intake, by sex
- `low_fruit_veg_by_age`: low fruit & vegetable intake, by age group
- `met_mean_total`: mean MET (if available)
- `met_mean_by_sex`: mean MET by sex (if available) (if the corresponding variables exist in design)

See Also

[compute_all_indicators\(\)](#)

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
diet_pa_results <- compute_diet_pa_indicators(design)
```

compute_engine	<i>Generic Compute Engine for WHO STEPS Tables</i>
----------------	--

Description

Takes a table specification from `steps_table_registry()` and a survey design object, and produces the survey-weighted estimates needed to fill the standard WHO STEPS data book table.

compute_table	<i>Compute a single table from a registry entry</i>
---------------	---

Description

This is the main workhorse: given one registry entry and a survey design, it dispatches to the appropriate method based on `entry$type` and returns a standardised result list.

Usage

```
compute_table(entry, design, data = NULL)
```

Arguments

<code>entry</code>	A single list element from <code>steps_table_registry()</code> .
<code>design</code>	A survey design object (from <code>survey::svydesign()</code>).
<code>data</code>	The cleaned data frame (used for variable availability checks).

Value

A list with:

id Table identifier.

title Table title.

type Table type.

available Logical: TRUE if the required variable(s) exist.

results A list of data frames: For proportion: total, by_sex, by_age (each with estimate, lower, upper). For mean: total, by_sex, by_age (each with estimate, lower, upper). For category: total, by_sex, by_age (each with level, estimate, lower, upper). For cascade: named list of proportion results.

`compute_tobacco_indicators`*Compute Tobacco Use Indicators*

Description

Calculates prevalence of tobacco use from a survey design object. Computes proportions of current and daily tobacco use, stratified by sex and age group where available.

Usage

```
compute_tobacco_indicators(design)
```

Arguments

`design` A survey design object from [setup_survey_design\(\)](#).

Details

When both smoking and smokeless tobacco variables are present, `current_tobacco_any` (either smoking or smokeless) is preferred as the headline tobacco indicator. The function also reports `current_smoker` and `current_smokeless` separately if available.

Value

A named list of survey estimates. Each element contains proportion estimates (as tibble with columns: estimate, lower, upper, etc.) for:

- `current_tobacco_any_total/by_sex/by_age`: any current tobacco use (smoking or smokeless) – preferred headline variable
- `current_tobacco_total/by_sex/by_age`: current tobacco smoking
- `current_smoker_total/by_sex/by_age`: current smoker
- `current_smokeless_total/by_sex/by_age`: current smokeless tobacco
- `daily_tobacco_total/by_sex/by_age`: daily tobacco use (only elements for variables present in design are returned)

See Also

[compute_all_indicators\(\)](#)

Examples

```
test_data <- generate_test_data(n = 500, seed = 42)
cols <- detect_steps_columns(test_data)
clean <- clean_steps_data(test_data, cols)
design <- setup_survey_design(clean)
tobacco_results <- compute_tobacco_indicators(design)
```

detect_col	<i>Detect a STEPS column by alias</i>
------------	---------------------------------------

Description

Tries to find a column in the data matching one of several candidate names (case-insensitive).

Usage

```
detect_col(data, candidates, label = NULL)
```

Arguments

data	A data frame.
candidates	Character vector of possible column names.
label	Optional label for progress messages.

Value

The matched column name (character) or NULL.

detect_steps_columns	<i>Auto-detect all standard STEPS columns</i>
----------------------	---

Description

Scans a data frame for standard WHO STEPS variable names across versions 3.1 and 3.2. Aliases are listed in priority order: the first match wins, so put the most specific / unambiguous name first.

Usage

```
detect_steps_columns(data)
```

Arguments

data	A data frame (typically from <code>import_steps_data()</code>).
------	--

Details

WHO STEPS reorganised variable codes between v3.1 and v3.2:

v3.1 / Epi Info codes (still common in many country datasets): B1-B6 = blood-pressure readings, B7 = BP meds, C1 = fasting glucose, C5 = DM meds, C6 = total cholesterol, C10 = chol meds, M1 = height, M2 = weight, M3 = waist.

v3.2 instrument codes: M4a/M5a/M6a = SBP readings, M4b/M5b/M6b = DBP readings, M7 = BP meds, M11 = height, M12 = weight, M14 = waist, M15 = hip, B5 = fasting glucose, B6 = DM meds, B8 = total cholesterol, B9 = chol meds, B16 = triglycerides, B17 = HDL cholesterol, C1 = sex, C3 = age.

The function includes aliases for both versions so datasets from either instrument version are detected automatically.

Value

A named list of detected column names (or NULL for missing).

generate_test_data	<i>Generate simulated STEPS test data</i>
--------------------	---

Description

Creates a realistic simulated dataset matching WHO STEPS survey structure. Includes sampling design variables, demographics, and measures from all three steps (behavioural, physical, biochemical).

Usage

```
generate_test_data(n = 3000, seed = 42)
```

Arguments

n	Number of observations (default 3000).
seed	Random seed for reproducibility (default 42).

Details

Simulation parameters are realistic for low-middle income settings:

- Tobacco prevalence: 32% males, 8% females
- Alcohol current use: 55% males, 28% females
- Heavy episodic drinking: 35% of drinkers
- Physical activity: MET-minutes/week, mean 1800, SD 1200
- Diet: Fruit/veg days and servings per day (0-7, 1-5)
- BP increases with age; medication prevalence 12%

- Glucose: mean 5.2 mmol/L, increases with age
- Total cholesterol: mean 4.8 mmol/L

Use this function for:

- Testing the STEPS pipeline
- Developing reports before real data arrives
- Training analysts on the analysis system

Value

A data frame with n rows and the following columns:

- stratum: Strata identifier (S1-S5)
- psu: Primary sampling unit (PSU1-PSU40)
- wt_final: Final analysis weight
- sex: Sex (1=Male, 2=Female)
- age: Age in years (18-69)
- Step 1 (behavioural): t1, t2 (tobacco), a1, a5 (alcohol), met_total (physical activity), d1-d4 (diet)
- Step 2 (physical): m1 (height), m2 (weight), m3 (waist), b1-b6 (blood pressure), b7 (BP medication)
- Step 3 (biochemical): c1_mmol (glucose), c5 (DM meds), c6 (cholesterol), c10 (cholesterol meds)

Examples

```
# Generate smaller dataset for quick testing
test_data <- generate_test_data(n = 500, seed = 123)
head(test_data)
```

get_registry_by_section

Get table registry entries by section

Description

Get table registry entries by section

Usage

```
get_registry_by_section(section = NULL)
```

Arguments

section Section name (e.g., "Tobacco Use", "Blood Pressure"). If NULL, returns all entries.

Value

A filtered list of registry entries.

get_registry_by_step *Get table registry entries by step*

Description

Get table registry entries by step

Usage

```
get_registry_by_step(step)
```

Arguments

step STEPS step number (1, 2, or 3).

Value

A filtered list of registry entries.

import_steps_data *Import raw STEPS survey data*

Description

Reads a raw STEPS data file (CSV, Excel, Stata, or SPSS) and standardises column names to lowercase with underscores.

Usage

```
import_steps_data(path)
```

Arguments

path Character. Path to the data file.

Value

A data frame with cleaned column names.

Examples

```
## Not run:  
raw <- import_steps_data("data/raw/steps_data.csv")  
  
## End(Not run)
```

```
list_registry_sections
```

List all available sections in the registry

Description

List all available sections in the registry

Usage

```
list_registry_sections()
```

Value

Character vector of unique section names.

```
plot_completeness
```

Plot completeness heatmap across STEPS domains

Description

Creates a tile heatmap showing missingness percentage by variable, grouped by STEPS domain.

Usage

```
plot_completeness(dq)
```

Arguments

`dq` A `steps_quality` object from `steps_data_quality()`.

Value

A ggplot object.

plot_digit_preference *Plot digit preference histogram for a physical measurement*

Description

Creates a bar chart of terminal-digit frequencies with the expected uniform line at 10 %.

Usage

```
plot_digit_preference(dq, measure)
```

Arguments

dq	A steps_quality object from steps_data_quality() .
measure	Character: one of "SBP", "DBP", "Height", "Weight", "Waist".

Value

A ggplot object.

plot_weights *Plot sampling weight distribution*

Description

Creates a histogram of sampling weights with summary statistics.

Usage

```
plot_weights(dq, step = "weight_step1")
```

Arguments

dq	A steps_quality object from steps_data_quality() .
step	Character: which weight to plot ("weight_step1", "weight_step2", or "weight_step3"). Defaults to "weight_step1".

Value

A ggplot object.

read_column_mapping *Read a column mapping file*

Description

Reads a filled-in column mapping template (Excel or CSV) and returns a named list suitable for passing to `clean_steps_data()`. The mapping file should have at least two columns: one with the standard variable name (column A) and one with the user's column name (column C in the template, or the third column).

Usage

```
read_column_mapping(path, data = NULL)
```

Arguments

path	Path to the filled mapping file (.xlsx or .csv).
data	Optional data frame. If provided, the function validates that every mapped column actually exists in the data.

Details

This function is the manual alternative to `detect_steps_columns()`. Use it when your dataset has non-standard variable names that auto-detection cannot resolve.

A blank template can be obtained from `system.file("templates", "column_mapping_template.xlsx", package = "stepssurvey")` or downloaded from the Shiny app.

The function ignores domain-header rows (rows where column A is all-caps with no entry in column C) and skips any row where the user's column name is blank.

Value

A named list where names are standard variable identifiers (e.g. "age", "sbp1") and values are the corresponding column names in the user's dataset. Unmapped variables are set to NULL.

Examples

```
## Not run:  
cols <- read_column_mapping("my_mapping.xlsx")  
raw <- import_steps_data("survey.dta")  
clean <- clean_steps_data(raw, cols)  
  
## End(Not run)
```

render_country_report *Render STEPS Country Report*

Description

Generates a comprehensive Word document with executive summary, indicator-by-indicator analysis, and recommendations for public health action.

Usage

```
render_country_report(config, output_dir = tempdir())
```

Arguments

config	A list from <code>steps_config()</code> with survey metadata. Expected to have <code>country_name</code> , <code>survey_year</code> , <code>age_min</code> , <code>age_max</code> .
output_dir	Directory for output reports (default <code>tempdir()</code>).

Details

Sections include:

- Executive summary with key findings
- Tobacco use
- Physical activity
- Overweight and obesity
- Blood pressure
- Blood glucose and cholesterol
- Recommendations for public health action
- Methodology

Requires pre-computed indicators, tables, and plots in `data/processed/`.

Value

Path to generated Word document (invisibly). Prints message with output location.

render_data_book *Render STEPS Data Book report*

Description

Generates a Word document with detailed age-stratified prevalence tables for all available indicators, organized by STEPS step.

Usage

```
render_data_book(config, output_dir = tempdir())
```

Arguments

config	A list from <code>steps_config()</code> with survey metadata. Expected to have <code>country_name</code> , <code>survey_year</code> , <code>age_min</code> , <code>age_max</code> .
output_dir	Directory for output reports (default <code>tempdir()</code>).

Details

Sections correspond to STEPS steps:

- Step 1: Behavioural Risk Factors (tobacco, alcohol, diet, physical activity)
- Step 2: Physical Measurements (overweight/obesity, blood pressure)
- Step 3: Biochemical (glucose, cholesterol)

Requires pre-computed tables and plots in `data/processed/`.

Value

Path to generated Word document (invisibly). Prints message with output location.

render_fact_sheet *Render STEPS Fact Sheet report*

Description

Generates a Word document with an overview of key NCD risk factor prevalence, including summary table and sex-stratified charts.

Usage

```
render_fact_sheet(config, output_dir = tempdir(), format = c("html", "word"))
```

Arguments

config	A list from <code>steps_config()</code> with survey metadata and paths. Expected to have <code>country_name</code> , <code>survey_year</code> , <code>age_min</code> , <code>age_max</code> .
output_dir	Directory for output reports (default <code>tempdir()</code>).
format	Output format: "html" for self-contained HTML (default) or "word" for Word (.docx).

Details

The fact sheet template uses pre-computed indicators, `key_indicators`, and plots (via .rds files in `data/processed/`). Requires `rmarkdown`, `flextable`, `ggplot2`, `glue`, `patchwork` packages.

Value

Path to generated output file (invisibly). Prints message with output location.

run_app	<i>Launch the stepssurvey Shiny Application</i>
---------	---

Description

Starts the interactive STEPS survey analysis app in the user's browser. The app provides a guided workflow: upload data, clean, set survey design, compute indicators, visualise results, and generate Word reports.

Usage

```
run_app(...)
```

Arguments

... Additional arguments passed to `shiny::shinyApp()`.

Value

A Shiny app object (invisibly). Called for its side effect of launching the application.

Examples

```
## Not run:  
run_app()  
  
## End(Not run)
```

run_steps_pipeline *Run the complete STEPS analysis pipeline*

Description

Imports raw data, cleans it, sets up the survey design, computes all indicators, generates publication-ready tables and plots, and optionally renders Word reports.

Usage

```
run_steps_pipeline(  
  data_path,  
  country_name = "Country Name",  
  survey_year = 2024,  
  age_min = 18,  
  age_max = 69,  
  output_dir = tempdir(),  
  render_reports = TRUE,  
  mapping_file = NULL  
)
```

Arguments

data_path	Path to raw STEPS data file (CSV, Excel, Stata, or SPSS).
country_name	Country name for reports (default "Country Name").
survey_year	Survey year (default 2024).
age_min	Minimum age in years (default 18).
age_max	Maximum age in years (default 69).
output_dir	Directory for all outputs (default tempdir()).
render_reports	Logical; render Word documents? (default TRUE).
mapping_file	Optional path to a filled column mapping template (Excel or CSV). If provided, uses read_column_mapping() instead of auto-detection. See the template at <code>system.file("templates", "column_mapping_template.xlsx", package = "stepssurvey")</code> .

Details

This is the main entry point for end-to-end STEPS analysis.

Value

A list with elements:

raw_data Original imported data frame

clean_data Cleaned and recoded data

cols Detected column mapping from `detect_steps_columns()`

design `survey::svydesign` object

indicators List of all computed indicator results by domain

key_indicators Summary tibble of headline estimates

tables List of `flextable::flextable` objects

plots List of `ggplot2::ggplot` objects

config Configuration list from `steps_config()`

Examples

```
## Not run:
# Auto-detect columns
result <- run_steps_pipeline("data/raw/steps_data.csv",
                             country_name = "Senegal",
                             survey_year = 2023)

result$key_indicators
result$plots$overview

# Use a custom column mapping
result <- run_steps_pipeline("data/raw/steps_data.csv",
                             country_name = "Senegal",
                             survey_year = 2023,
                             mapping_file = "my_mapping.xlsx")

## End(Not run)
```

save_steps_plots	<i>Save STEPS plots to PNG files</i>
------------------	--------------------------------------

Description

Exports all plots in a list to PNG files in the specified directory.

Usage

```
save_steps_plots(plots, output_dir = tempdir())
```

Arguments

plots A named list of `ggplot2` objects (from `build_steps_plots()`).

output_dir Output directory path (default `tempdir()`).

Details

Files are named:

- 01_overview_indicators.png (12x8 in)
- 02_by_sex_dashboard.png (12x8 in)
- 03_bp_by_age.png (10x6 in)
- 04_obesity_by_age.png (10x6 in)

All saved at 150 dpi with white background.

Value

NULL (invisibly). Prints messages about saved files.

setup_survey_design *Set up survey designs for STEPS data (one per Step)*

Description

Creates up to three survey design objects — one per WHO STEPS Step — each using the appropriate step-specific weight column (`wt_step1`, `wt_step2`, `wt_step3`).

Usage

```
setup_survey_design(data)
```

Arguments

`data` A data frame (typically from `clean_steps_data()`).

Details

The returned object is a list of class "steps_designs" with elements `$step1`, `$step2`, `$step3`. For backward compatibility it can also be used directly as a single design (it delegates to `$step1`).

The function handles five design cases per step:

1. Full complex design: weights + strata + clusters
2. Weights + clusters, no strata
3. Weights + strata, no clusters
4. Weights only
5. Unweighted (simple random sampling)

Weights are used as-is without trimming, consistent with the WHO official STEPS analysis scripts.

Value

A list of class "steps_designs" with three `survey::svydesign` objects (`step1`, `step2`, `step3`).

steps_colors	<i>WHO STEPS colour palette</i>
--------------	---------------------------------

Description

A named list of colours used in WHO STEPS reports and visualisations.

Usage

```
steps_colors()
```

Value

A named list of hex colour codes.

Examples

```
steps_colors()$blue
```

steps_config	<i>Create STEPS analysis configuration</i>
--------------	--

Description

Builds a configuration list that specifies data paths, design variables, and report parameters for the STEPS pipeline.

Usage

```
steps_config(  
  data_path,  
  country_name = "Country Name",  
  survey_year = 2024,  
  age_min = 18,  
  age_max = 69,  
  weight_var = "wt_final",  
  strata_var = "stratum",  
  cluster_var = "psu",  
  bp_sbp_threshold = 140,  
  bp_dbp_threshold = 90,  
  bmi_overweight = 25,  
  bmi_obese = 30,  
  glucose_threshold = 7,  
  glucose_impaired_threshold = 6.1,  
  chol_threshold = 5  
)
```

Arguments

data_path	Path to raw STEPS data file (CSV or Excel).
country_name	Country name for reports (default "Country Name").
survey_year	Survey year (default 2024).
age_min	Minimum age (default 18).
age_max	Maximum age (default 69).
weight_var	Weight variable name (default "wt_final", set NULL if none).
strata_var	Strata variable name (default "stratum", set NULL if none).
cluster_var	Cluster variable name (default "psu", set NULL if none).
bp_sbp_threshold	SBP threshold for raised BP (default 140).
bp_dbp_threshold	DBP threshold for raised BP (default 90).
bmi_overweight	BMI threshold for overweight (default 25.0).
bmi_obese	BMI threshold for obesity (default 30.0).
glucose_threshold	Fasting glucose threshold in mmol/L (default 7.0).
glucose_impaired_threshold	Impaired fasting glucose threshold in mmol/L (default 6.1).
chol_threshold	Total cholesterol threshold in mmol/L (default 5.0).

Value

A list with elements:

- data_path: Input file path
- country_name: Country name
- survey_year: Survey year
- age_min, age_max: Age range
- weight_var, strata_var, cluster_var: Design variable names
- Threshold parameters for BP, BMI, glucose, cholesterol

Examples

```
## Not run:
cfg <- steps_config("data/steps_2023.csv", "Senegal", 2023)
cfg <- steps_config("data/steps.csv", "Mongolia", 2019,
                    bp_sbp_threshold = 130, bp_dbp_threshold = 80)

## End(Not run)
```

steps_data_quality *Data Quality Diagnostics for WHO STEPS Data*

Description

Produces a comprehensive data quality report covering digit preference, completeness, plausibility, and sampling weight diagnostics.

Usage

```
steps_data_quality(raw, cleaned, cols)
```

Arguments

raw	The raw (pre-cleaning) data frame, typically from <code>import_steps_data()</code> .
cleaned	The cleaned data frame from <code>clean_steps_data()</code> .
cols	Column mapping list from <code>detect_steps_columns()</code> .

Details

Digit preference / heaping is assessed using the Whipple-style heaping index: the ratio of observed frequency at a digit (0 or 5) to the expected frequency under uniform distribution. An index of 1.0 = no preference; >1.5 = moderate heaping; >2.0 = severe.

Completeness reports missing values for key STEPS variables grouped by Step (behavioural, physical, biochemical).

Plausibility counts values outside WHO-recommended ranges (e.g. height 100–250 cm, weight 20–300 kg, SBP 60–300 mmHg).

Weight diagnostics summarise the distribution of sampling weights and flag potential issues (high CV, zero/NA weights).

Value

A list of class "steps_quality" with elements:

digit_preference Terminal-digit tables and heaping indices for physical measurements (SBP, DBP, height, weight, waist).

completeness Per-variable missingness counts and percentages, grouped by STEPS domain.

plausibility Summary of values outside plausible ranges.

weights Sampling weight distribution statistics.

steps_table_registry *WHO STEPS Data Book Table Registry*

Description

Defines all standard tables from the WHO STEPS Epi Info report template. Each entry specifies the table metadata; generic compute and formatting functions use this registry to produce the full data book automatically.

Usage

```
steps_table_registry()
```

Value

A list of table specification lists.

Table types

proportion Single binary indicator: % (95% CI) by age × sex. Most common type. Example: "Current smokers among all respondents."

mean Continuous variable: mean (95% CI) by age × sex. Example: "Mean BMI (kg/m²)."

category Multi-level factor: % per level (95% CI) by age × sex. Example: "BMI classifications (Underweight / Normal / Overweight / Obese)."

cascade Diagnosis → treatment → control chain: multiple proportions with nested denominators. Example: "Raised BP diagnosis, treatment and control."

combined Summary of combined risk factors: 0, 1-2, 3-5 risk factors.

Registry fields

id Unique short identifier (e.g., "T_smoking_current").

section Data book section (e.g., "Tobacco Use", "Blood Pressure").

step STEPS step number (1, 2, or 3).

title Table title as shown in the data book.

description One-line description from the WHO template.

type One of: "proportion", "mean", "category", "cascade", "combined".

variable Column name(s) in the cleaned data frame to analyse. For proportion: single logical variable. For mean: single numeric variable. For category: single factor variable. For cascade: named list of logical variables.

denominator NULL (= all respondents) or column name for subsetting (e.g., "current_alcohol" to restrict to drinkers).

levels For category type: named character vector of level labels.

epi_info Epi Info program name(s) for reference.

unit Display unit (e.g., "%", "mmHg", "cm", "kg/m²", "mmol/L").

questions STEPS instrument question codes used.

sex_panels Logical. TRUE = 3 panels (Men/Women/Both); FALSE = 2 panels (Men/Women only, e.g., height/weight means). Default TRUE.

 svymn

Weighted mean estimation with 95% CI

Description

Calculates weighted means with 95% confidence intervals for a continuous variable, optionally stratified by a grouping variable.

Usage

```
svymn(formula, design, by = NULL, na.rm = TRUE)
```

Arguments

formula	A formula (e.g., ~age).
design	A survey design object (from setup_survey_design()).
by	Optional formula for stratification (e.g., ~sex).
na.rm	Logical; if TRUE (default), omit NA values.

Value

A data frame with columns:

- estimate: estimated mean
- lower: 95% CI lower bound
- upper: 95% CI upper bound
- se: standard error
- If by is specified: grouping column(s) prepended

svyprop	<i>Weighted proportion estimation with 95% CI</i>
---------	---

Description

Calculates weighted proportions (as percentages) with 95% confidence intervals for a yes/no variable, optionally stratified by a grouping variable.

Usage

```
svyprop(formula, design, by = NULL, na.rm = TRUE)
```

Arguments

formula	A formula (e.g., ~variable or using binary variables).
design	A survey design object (from setup_survey_design()).
by	Optional formula for stratification (e.g., ~sex).
na.rm	Logical; if TRUE (default), omit NA values.

Value

A data frame with columns:

- estimate: estimated proportion (%)
- lower: 95% CI lower bound (%)
- upper: 95% CI upper bound (%)
- se: standard error (%)
- If by is specified: grouping column(s) prepended

table_builder	<i>Generic Table Builder for WHO STEPS Data Book</i>
---------------	--

Description

Takes computed results from [compute_table\(\)](#) or [compute_all_tables\(\)](#) and produces formatted flextable objects in the standard WHO STEPS 3-panel format (Men / Women / Both Sexes).

theme_steps	<i>WHO STEPS ggplot2 theme</i>
-------------	--------------------------------

Description

A clean, minimal ggplot2 theme styled with WHO STEPS colours.

Usage

```
theme_steps(base_size = 11)
```

Arguments

base_size Base font size (default 11).

Value

A `ggplot2::theme` object.

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) + geom_point() + theme_steps()
```

Index

- * **compute**
 - compute_engine, 16
- * **tables**
 - table_builder, 36

- build_all_tables, 3
- build_forest_plot, 4
- build_radar_plot, 4
- build_steps_plots, 5
- build_steps_plots(), 29
- build_steps_tables, 6
- build_table, 7

- clean_steps_data, 7
- clean_steps_data(), 24, 30, 33
- compute_alcohol_indicators, 9
- compute_all_indicators, 10
- compute_all_indicators(), 5, 6, 9, 12–15, 17
- compute_all_tables, 11
- compute_all_tables(), 3, 36
- compute_anthropometry_indicators, 11
- compute_biochemical_indicators, 12
- compute_bp_indicators, 14
- compute_diet_pa_indicators, 15
- compute_engine, 16
- compute_table, 16
- compute_table(), 7, 11, 36
- compute_tobacco_indicators, 17

- detect_col, 18
- detect_steps_columns, 18
- detect_steps_columns(), 8, 24, 29, 33

- flextable::flextable, 29

- generate_test_data, 19
- get_registry_by_section, 20
- get_registry_by_step, 21
- ggplot2::ggplot, 29
- ggplot2::theme, 37

- import_steps_data, 21
- import_steps_data(), 8, 18, 33

- list_registry_sections, 22

- plot_completeness, 22
- plot_digit_preference, 23
- plot_weights, 23

- read_column_mapping, 24
- read_column_mapping(), 28
- render_country_report, 25
- render_data_book, 26
- render_fact_sheet, 26
- run_app, 27
- run_steps_pipeline, 28

- save_steps_plots, 29
- setup_survey_design, 30
- setup_survey_design(), 9–11, 13–15, 17, 35, 36
- shiny::shinyApp(), 27
- steps_colors, 31
- steps_config, 31
- steps_config(), 25–27, 29
- steps_data_quality, 33
- steps_data_quality(), 22, 23
- steps_table_registry, 34
- steps_table_registry(), 11, 16
- stepssurvey (stepssurvey-package), 3
- stepssurvey-package, 3
- survey::svydesign, 10, 29, 30
- survey::svydesign(), 16
- svymn, 35
- svyprop, 36

- table_builder, 36
- theme_steps, 37