Package 'sleacr'

May 9, 2026

Type Package

Title Simplified Lot Quality Assurance Sampling Evaluation of Access and Coverage (SLEAC) Tools

Version 0.1.3

Description In the recent past, measurement of coverage has been mainly through two-stage cluster sampled surveys either as part of a nutrition assessment or through a specific coverage survey known as Centric Systematic Area Sampling (CSAS). However, such methods are resource intensive and often only used for final programme evaluation meaning results arrive too late for programme adaptation. SLEAC, which stands for Simplified Lot Quality Assurance Sampling Evaluation of Access and Coverage, is a low resource method designed specifically to address this limitation and is used regularly for monitoring, planning and importantly, timely improvement to programme quality, both for agency and Ministry of Health (MoH) led programmes. SLEAC is designed to complement the Semi-quantitative Evaluation of Access and Coverage (SQUEAC) method. This package provides functions for use in conducting a SLEAC assessment.

License GPL (>= 3)

Depends R (>= 4.2.0)

Imports cli, methods, parallel, parallelly, stats

Suggests covr, knitr, rmarkdown, spelling, testthat (>= 3.0.0), tibble

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.3

URL https://nutriverse.io/sleacr/, https://github.com/nutriverse/sleacr

BugReports https://github.com/nutriverse/sleacr/issues

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Ernest Guevarra [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-4887-4415>),
Mark Myatt [aut, cph] (ORCID: <https://orcid.org/0000-0003-1119-1474>)

Maintainer Ernest Guevarra <ernest@guevarra.io>

Repository CRAN

Date/Publication 2026-02-03 14:30:02 UTC

Contents

`check_coverage_homogeneity`
Check coverage distribution

Description

Check coverage distribution

Usage

```
check_coverage_homogeneity(cov_df, k = 3, p = 0.05)
```

Arguments

`cov_df`	A `data.frame()` of stratified coverage survey data to get overall coverage estimates of. `cov_df` should have a variable named `cases_in` for number of SAM or MAM cases in the programme found during the survey, `cases_out` for number SAM or MAM cases not in the programme found during the survey, and `rec_in` for children recovering from SAM or MAM who are in the programme

found during the survey. A final required variable should be one that contains identifying geographical information corresponding to the location from which each row of the survey data was collected from.

k — Correction factor. Ratio of the mean length of an untreated episode to the mean length of a CMAM treatment episode

p — Minimum p-value to test statistic. Default is 0.05.

Value

A named list of 2 lists: one for case-finding effectiveness (*cf*) and the second for treatment coverage (*tc*). For each list, the following values are provided:

- **statistic** - calculated chi-square statistic
- **df** - degrees of freedom
- **p** - p-value of chi-square statistic

Examples

```
check_coverage_homogeneity(survey_data)
```

---

estimate_coverage_overall

Weighted post-stratification estimation of coverage over several service delivery units

---

Description

Weighted post-stratification estimation of coverage over several service delivery units

Usage

```
estimate_coverage_overall(cov_df, pop_df, strata, u5, p, k = 3)

estimate_coverage(cov_df, cov_type = c("cf", "tc"), k = 3)
```

Arguments

cov_df — A `data.frame()` of stratified coverage survey data to get overall coverage estimates of. `cov_df` should have a variable named `cases_in` for number of SAM or MAM cases in the programme found during the survey, `cases_out` for number SAM or MAM cases not in the programme found during the survey, and `rec_in` for children recovering from SAM or MAM who are in the programme found during the survey. A final required variable should be one that contains identifying geographical information corresponding to the location from which each row of the survey data was collected from.

pop_df	A data.frame() with at least two variables: strata for the stratification/grouping information that matches the grouping information in cov_df and pop for information on population for the given grouping information.
strata	A character value of the variable name in cov_df that corresponds to the strata values to match with values in pop_df.
u5	A numeric value for the proportion of the population that is under years old.
p	Prevalence of SAM or MAM in the given population.
k	Correction factor. Ratio of the mean length of an untreated episode to the mean length of a CMAM treatment episode
cov_type	Coverage estimator to report. Either *"cf"* for *case-finding effectiveness* or *"tc"* for *treatment coverage*. Default is *"cf"*.

Value

A list of overall coverage estimates with corresponding 95% confidence intervals for case-finding effectiveness and treatment coverage.

Examples

```
cov_df <- survey_data

pop_df <- pop_data |>
  setNames(nm = c("strata", "pop"))

estimate_coverage_overall(
  cov_df, pop_df, strata = "district", u5 = 0.177, p = 0.01
)
```

---

get_n_cases *Calculate estimated number of cases for a condition affecting children under 5 years old in a specified survey area*

Description

Calculate estimated number of cases for a condition affecting children under 5 years old in a specified survey area

Usage

```
get_n_cases(N, u5, p)
```

Arguments

N	Population for all ages in the specified survey area.
u5	Proportion (value from 0 to 1) of population that are aged 6-59 months.
p	Prevalence (value from 0 to 1) of condition that is to be assessed.

Value

A numeric value of the estimated number of cases in the specified survey area

Examples

```
## Calculate number of SAM cases in a population of 100000 persons of all
## ages with an under-5 population of 17% and a prevalence of 2%
get_n_cases(N = 100000, u5 = 0.17, p = 0.02)
```

`get_n_clusters` *Calculate number of clusters to sample to reach target sample size*

Description

Calculate number of clusters to sample to reach target sample size

Usage

```
get_n_clusters(n, n_cluster, u5, p)
```

Arguments

`n`	Target sample size of cases for the coverage survey.
`n_cluster`	Average cluster population for all ages in the specified survey area.
`u5`	Proportion (value from 0 to 1) of population that are aged 6-59 months.
`p`	Prevalence (value from 0 to 1) of condition that is to be assessed.

Value

A numeric value of the estimated number of clusters to sample to reach target sample size.

Examples

```
## Calculate number of villages to sample given an average village population
## of 600 persons of all ages with an under-5 population of 17% and a
## prevalence of SAM of 2% if the target sample size is 40
get_n_clusters(n = 40, n_cluster = 600, u5 = 0.17, p = 0.02)
```

get_sample_n *Calculate sample size and decision rule for a specified LQAS sampling plan*

Description

Calculate sample size and decision rule for a specified LQAS sampling plan

Usage

```
get_sample_n(N, dLower, dUpper, alpha = 0.1, beta = 0.1)

get_sample_d(N, n, dLower, dUpper, alpha = 0.1, beta = 0.1)
```

Arguments

N	Total population size of cases in the specified survey area
dLower	Lower triage threshold. Values from 0 to 1.
dUpper	Upper triage threshold. Values from 0 to 1.
alpha	Maximum tolerable alpha error. Values from 0 to 1. Default is 0.1
beta	Maximum tolerable beta error. Values from 0 to 1. Default is 0.1
n	Sample size

Value

A list of values providing the LQAS sampling plan for the specified parameters. The list includes sample size, decision rule, alpha error and beta error for the specified classification scheme

Examples

```
get_sample_n(N = 600, dLower = 0.7, dUpper = 0.9)
get_sample_d(N = 600, n = 40, dLower = 0.7, dUpper = 0.9)
```

get_sampling_clusters *Select sampling clusters using systematic sampling*

Description

Select sampling clusters using systematic sampling

Usage

```
get_sampling_clusters(N_clusters, n_clusters)

get_sampling_list(cluster_list, n_clusters)
```

Arguments

N_clusters	Total number of clusters in survey area.
n_clusters	Number of sampling clusters to be selected.
cluster_list	A data.frame containing at least the name or any other identifier for the entire set of clusters to sample from.

Value

An integer vector for get_sampling_clusters() giving the row index for selected clusters. A data.frame for [get_sampling_list()] which is a subset of cluster_list.

Examples

```
get_sampling_clusters(N_clusters = 211, n_clusters = 35)
get_sampling_list(cluster_list = village_list, n_clusters = 70)
```

---

lqas_classify_ *LQAS classifier*

Description

LQAS classifier

Usage

```
lqas_classify_(
  cases_in,
  cases_out,
  rec_in = NULL,
  k = 3,
  threshold = c(0.2, 0.5),
  label = FALSE
)

lqas_classify(
  cases_in,
  cases_out,
  rec_in = NULL,
  k = 3,
  threshold = c(0.2, 0.5),
  label = FALSE
)

lqas_classify_cf(cases_in, cases_out, threshold = c(0.2, 0.5), label = FALSE)
```

```
lqas_classify_tc(
  cases_in,
  cases_out,
  rec_in,
  k,
  threshold = c(0.2, 0.5),
  label = FALSE
)
```

Arguments

cases_in	Number of SAM and/or MAM cases found during the survey who are in the CMAM programme.
cases_out	Number of SAM and/or MAM cases found during the survey who are in the CMAM programme.
rec_in	Number of children recovering from SAM or MAM found during the survey who are in the programme.
k	Correction factor. Ratio of the mean length of an untreated episode to the mean length of a CMAM treatment episode
threshold	Decision rule threshold/s. Should be between 0 and 1. At least one threshold should be provided for a two-tier classifier. Two thresholds should be provided for a three-tier classifier. Default is a three-tier classifier with rule set at 0.2 and 0.5.
label	Logical. Should the output results be classification labels? If TRUE, output classification are character labels else they are integer values. Default is FALSE.

Value

A `data.frame()` of coverage classifications for case-finding effectiveness and for treatment coverage.

Author(s)

Ernest Guevarra

Examples

```
lqas_classify(cases_in = 6, cases_out = 34, rec_in = 6)

with(
  survey_data,
  lqas_classify(
    cases_in = cases_in, cases_out = cases_out, rec_in = rec_in
  )
)
```

lqas_get_class_prob *Produce misclassification probabilities*

Description

Produce misclassification probabilities

Usage

```
lqas_get_class_prob(x)
```

Arguments

x	Simulated results data produced by `lqas_simulate_test()`

Value

A list object of class `lqasClass` for LQAS misclassification probabilities results

Examples

```
sim <- lqas_simulate_test(
  pop = 10000, n = 40, dLower = 0.6, dUpper = 0.9, replicates = 5, runs = 5
)

lqas_get_class_prob(x = sim)
```

lqas_simulate_population *Simulate survey data of covered/cases and non-covered/non-cases given a coverage/prevalence proportion*

Description

Simulate survey data of covered/cases and non-covered/non-cases given a coverage/prevalence proportion

Perform LQAS on simulated data based on specified decision rules

Usage

```
lqas_simulate_population(proportion, pop)

lqas_simulate_run(proportion, pop, n, dLower, dUpper)

lqas_simulate_runs(
  pop,
  n,
  dLower,
  dUpper,
  pLower = 0,
  pUpper = 1,
  fine = 0.01,
  runs = 50,
  cores = parallelly::availableCores(omit = 1)
)

lqas_simulate_test(
  pop,
  n,
  dLower,
  dUpper,
  pLower = 0,
  pUpper = 1,
  fine = 0.01,
  runs = 50,
  replicates = 20,
  cores = parallelly::availableCores(omit = 1)
)
```

Arguments

proportion	A numeric value of a coverage/prevalence proportion to simulate on. Values should be between 0 and 1.
pop	Population size from which simulated coverage survey data is to be taken from.
n	Sample size of actual or test coverage data.
dLower	A numeric value for the lower classification threshold proportion. Value should be between 0 and 1.
dUpper	A numeric value for the upper classification threshold proportion. Value should be between 0 and 1.
pLower	Starting proportion for simulations. Default is 0.
pUpper	Ending proportion for simulations. Default is 1.
fine	Granularity of simulated proportions. Default is 0.01.
runs	Number of simulation runs to perform per coverage proportion. Default is 50 runs.

cores	The number of computer cores to use/number of child processes will be run simultaneously.
replicates	Number of replicate LQAS simulations to perform. Default is set to 20 replicates.

Value

A data.frame with 2 variables: id for unique identifier and case for numeric vector of cases and non-cases (1s and 0s)

A data.frame with variable cases for total number of covered/cases, outcome for LQAS outcome, and proportion for the coverage/prevalence proportion being simulated on. For [lqas_simulate_test()], a 'sleac" class object.

Examples

```
lqas_simulate_population(proportion = 0.3, pop = 10000)

lqas_simulate_run(
  proportion = 0.3, pop = 10000, n = 40, dLower = 0.6, dUpper = 0.9
)

lqas_simulate_runs(
  pop = 10000, n = 40, dLower = 0.6, dUpper = 0.9, runs = 10
)

lqas_simulate_test(
  pop = 10000, n = 40, dLower = 0.6, dUpper = 0.9, runs = 5, replicates = 5
)
```

---

plot.lqasSim plot *helper function for* lqas_simulate_test() *function*

Description

plot helper function for lqas_simulate_test() function

Usage

```
## S3 method for class 'lqasSim'
plot(x, ...)
```

Arguments

x	An object of class lqasSim produced by lqas_simulate_test() function
...	Additional plot parameters

Value

An LQAS probability of classification plot

Examples

```
x <- lqas_simulate_test(
  pop = 10000, n = 40, dLower = 0.6, dUpper = 0.9, replicates = 5, runs = 5
)
plot(x)
```

---

pop_data *Population data of districts of Sierra Leone based on 2015 census*

Description

Population data of districts of Sierra Leone based on 2015 census

Usage

```
pop_data
```

Format

A tibble with 14 rows and 2 columns

Variable	Description
district	District name
pop	Population

Source

https://sierraleone.unfpa.org/sites/default/files/pub-pdf/Population%20structure%20Report_1.pdf

Examples

```
pop_data
```

print.lqasClass — print *helper function for* lqas_get_class_prob() *function*

Description

print helper function for lqas_get_class_prob() function

Usage

```
## S3 method for class 'lqasClass'
print(x, ...)
```

Arguments

x	An object resulting from applying the lqas_get_class_prob() function.
...	Additional print parameters

Value

Printed output of lqas_get_class_prob() function

Examples

```
sim <- lqas_simulate_test(
  pop = 10000, n = 40, dLower = 0.6, dUpper = 0.9, replicates = 5, runs = 5
)

x <- lqas_get_class_prob(x = sim)
print(x)
```

survey_data — *SLEAC survey data from Sierra Leone*

Description

SLEAC survey data from Sierra Leone

Usage

```
survey_data
```

Format

A tibble with 14 rows and 6 columns

Variable	Description
country	Country
province	Province
district	District
cases_in	SAM cases found who are in the programme
cases_out	SAM cases found who are not in the programme
rec_in	Recovering SAM cases in the programme
cases_total	Total number SAM cases found

Source

Ministry of Health, Sierra Leone

Examples

```
survey_data
```

`village_list` *List of villages in Bo District, Sierra Leone*

Description

List of villages in Bo District, Sierra Leone

Usage

```
village_list
```

Format

A tibble with 1001 rows and 4 columns

Variable	Description
id	Unique identifier
chiefdom	Chiefdom
section	Section
village	Village

Source

Ministry of Health, Sierra Leone

Examples

```
village_list
```

Index