Package: NRMSampling (via r-universe)

May 22, 2026

Type Package

Title Sampling Design and Estimation Methods for Natural Resource Management

Version 0.2.2

Description Provides functions for probability and non-probability sampling design, sample selection, and population estimation tailored to natural resource management. Probability methods include simple random sampling, stratified sampling, systematic sampling, cluster sampling, and probability-proportional-to-size sampling. Non-probability methods include convenience, judgement-based, and quota sampling. Estimation functions cover means, totals, ratio estimators, regression estimators, and the unequal-probability estimator of Horvitz and Thompson (1952, <doi:10.2307/2280784>) for unequal-probability designs. Utilities support biomass, soil-loss, and carbon-stock estimation from field plots. Spatial extensions provide random, systematic, stratified, and raster-weighted sampling within geographic polygons using the 'sf' and 'terra' packages, with extraction of remote-sensing covariates at sample locations. Applications include forest inventory, soil erosion monitoring, watershed studies, and ecological field surveys.

License GPL (>= 3)

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports stats, utils

Suggests sf (>= 1.0.0), terra (>= 1.7.0), ggplot2 (>= 3.4.0), dplyr (>= 1.1.0), testthat (>= 3.0.0), knitr (>= 1.40), rmarkdown (>= 2.18), covr, spelling

VignetteBuilder knitr

RoxygenNote 7.3.3

Config/testthat/edition 3

LazyData true

NeedsCompilation no

Author Sadikul Islam [aut, cre] (ORCID: <https://orcid.org/0000-0003-2924-7122>)

Maintainer Sadikul Islam <`sadikul.islamiasri@gmail.com`>

Repository https://cran.r-universe.dev

Date/Publication 2026-04-22 14:13:04 UTC

RemoteUrl https://github.com/cran/NRMSampling

RemoteRef HEAD

RemoteSha e32b325c78f7d48a7db2b795077e2497aa6bfe9b

Contents

`NRMSampling-package` *NRMSampling: Sampling Design and Estimation for Natural Resource Management*

Description

Provides functions for probability and non-probability sampling design, sample selection, and estimation tailored to natural resource management.

Probability Sampling

`srs_sample`, `stratified_sample`, `systematic_sample`, `cluster_sample`, `pps_sample`

Non-Probability Sampling

`convenience_sample`, `purposive_sample`, `quota_sample`

Estimation

`estimate_mean`, `estimate_total`, `estimate_variance`, `estimate_se`, `estimate_ci`, `ratio_estimator`, `regression_estimator`, `ht_estimator`, `ht_variance`, `stratified_estimator`

NRM Utilities

`biomass_estimate`, `soil_loss_estimate`, `carbon_stock_estimate`, `plot_summary`, `sampling_efficiency`

Spatial Sampling

`to_sf_points`, `spatial_random_sample`, `spatial_systematic_sample`, `spatial_stratified_sample`, `spatial_cluster_sample`, `raster_stratified_sample`, `raster_pps_sample`, `extract_raster_values`, `spatial_biomass_estimate`, `plot_sampling`, `plot_sampling_gg`

Datasets

`sample_nrm`, `sample_spatial`

Note

`sample()`, `runif()`, `setNames()`, and `mean()` are `base::` functions and must NOT be listed in `importFrom(stats, ...)`. They are always available without any import declaration.

Author(s)

Maintainer: Sadikul Islam `<sadikul.islamiasri@gmail.com>` (ORCID)

References

Cochran, W.G. (1977). *Sampling Techniques*, 3rd ed. Wiley. Lohr, S.L. (2022). *Sampling: Design and Analysis*, 3rd ed. CRC Press.

`biomass_estimate` *Biomass Estimation from Field Plots*

Description

Estimates total standing biomass over a landscape or management unit from plot-level measurements.

Usage

```
biomass_estimate(df, biomass_var, area)
```

Arguments

`df`	A data frame of sampled plots.
`biomass_var`	Character. Name of the column containing plot-level biomass density (e.g., Mg/ha or kg/plot).
`area`	Numeric. Total area of the management unit (same units as the denominator of `biomass_var`). E.g., if `biomass_var` is in Mg/ha, `area` should be in ha.

Value

A named list with elements:

`mean_biomass` Mean biomass density across sampled plots.

`total_biomass` Estimated total biomass over the study area.

`se` Standard error of the mean biomass density.

`n` Number of plots used.

References

Avery, T.E. and Burkhart, H.E. (2002). *Forest Measurements*, 5th ed. McGraw-Hill, New York.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
biomass_estimate(srs, biomass_var = "biomass", area = 1000)
```

`carbon_stock_estimate` *Carbon Stock Estimation*

Description

Converts a biomass estimate to carbon stock using a biomass-to-carbon conversion factor.

Usage

```
carbon_stock_estimate(df, biomass_var, area, carbon_fraction = 0.47)
```

Arguments

`df`	A data frame of sampled plots.
`biomass_var`	Character. Name of the biomass density column.
`area`	Numeric. Total study area.
`carbon_fraction`	Numeric. Fraction of biomass that is carbon. Default 0.47 (IPCC default for tropical forests).

Value

A named list with elements `total_biomass`, `total_carbon`, `carbon_fraction`, and `n`.

References

IPCC (2006). *IPCC Guidelines for National Greenhouse Gas Inventories, Volume 4: Agriculture, Forestry and Other Land Use*. IGES, Hayama, Japan.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
carbon_stock_estimate(srs, biomass_var = "biomass", area = 1000)
```

`cluster_sample` *Cluster Sampling*

Description

Performs single-stage cluster sampling.

Usage

```
cluster_sample(data, cluster_var, n_clusters)
```

Arguments

`data`	A data frame.
`cluster_var`	Character. Cluster column.
`n_clusters`	Integer. Number of clusters.

Value

A data frame of sampled clusters.

`convenience_sample` *Convenience Sampling*

Description

Returns the first `n` rows of a data frame as a convenience sample. This is the simplest non-probability method; results are generally not representative of the population.

Usage

```
convenience_sample(data, n)
```

Arguments

`data`	A data frame.
`n`	Integer. Number of units to select.

Details

Convenience sampling is fast but prone to selection bias. It may be appropriate for pilot studies or logistical constraints, but population inference requires strong assumptions. See Lohr (2022) for discussion.

Value

A data frame containing the first `n` rows.

References

Lohr, S.L. (2022). *Sampling: Design and Analysis*, 3rd ed. CRC Press, Boca Raton, FL.

Examples

```
data(sample_nrm)
cs <- convenience_sample(sample_nrm, n = 10)
nrow(cs)
```

`estimate_ci` *Confidence Interval for the Population Mean*

Description

Computes a confidence interval for the population mean based on a simple random sample, using the t-distribution.

Usage

```
estimate_ci(y, N = NULL, conf_level = 0.95)
```

Arguments

`y`	Numeric vector. Sample observations.
`N`	Integer. Population size. Used for the finite-population correction. Set to `NULL` to omit.
`conf_level`	Numeric. Confidence level in (0, 1). Default 0.95.

Value

A named numeric vector with elements `mean`, `lower`, and `upper`.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
estimate_ci(srs$biomass, N = nrow(sample_nrm))
```

`estimate_mean` *Sample Mean*

Description

Computes the sample mean of a numeric vector, ignoring missing values.

Usage

```
estimate_mean(y)
```

Arguments

`y`	Numeric vector. Sample observations.

Value

Numeric scalar. The sample mean.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
estimate_mean(srs$biomass)
```

`estimate_se` *Standard Error of the Sample Mean*

Description

Computes the estimated standard error of the sample mean under simple random sampling without replacement (SRSWOR).

Usage

```
estimate_se(y, N = NULL)
```

Arguments

`y`	Numeric vector. Sample observations.
`N`	Integer. Population size. If `NULL` (default), the finite-population correction (fpc) is omitted.

Details

With fpc:

$$SE(\bar{y}) = \sqrt{\frac{s^2}{n}\left(1 - \frac{n}{N}\right)}$$

Without fpc:

$$SE(\bar{y}) = \sqrt{\frac{s^2}{n}}$$

Value

Numeric scalar. Estimated standard error.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
estimate_se(srs$biomass, N = nrow(sample_nrm))
```

`estimate_total` *Population Total Estimator*

Description

Estimates the population total by expanding the sample mean to the full population size.

Usage

```
estimate_total(y, N)
```

Arguments

`y`	Numeric vector. Sample observations.
`N`	Integer. Known population size (number of units).

Details

$$\hat{Y} = N\bar{y}$$

Value

Numeric scalar. Estimated population total.

References

Cochran, W.G. (1977). *Sampling Techniques*, 3rd ed. John Wiley & Sons, New York.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
estimate_total(srs$biomass, N = nrow(sample_nrm))
```

`estimate_variance` *Sample Variance*

Description

Computes the unbiased sample variance of a numeric vector.

Usage

```
estimate_variance(y)
```

Arguments

`y`	Numeric vector. Sample observations.

Value

Numeric scalar. The unbiased sample variance s^2.

Examples

```
data(sample_nrm)
estimate_variance(sample_nrm$biomass)
```

`extract_raster_values` *Extract Raster Values at Sample Points*

Description

Extracts cell values from a `SpatRaster` at the locations of `sf` point features, returning the results as a data frame.

Usage

```
extract_raster_values(raster, points_sf)
```

Arguments

`raster`	A `SpatRaster` object.
`points_sf`	An `sf` POINT object. Must have a compatible CRS with `raster`.

Details

Requires both the **terra** and **sf** packages. The CRS of `points_sf` is reprojected to match `raster` if needed.

Value

A data frame with one row per point and one column per raster layer.

Examples

```
if (requireNamespace("terra", quietly = TRUE) &&
    requireNamespace("sf", quietly = TRUE)) {
  r    <- terra::rast(nrows=20, ncols=20, vals=runif(400))
  bbox <- sf::st_as_sfc(sf::st_bbox(terra::ext(r)))
  pts  <- spatial_random_sample(bbox, n = 10)
  sf::st_crs(pts) <- sf::st_crs(terra::crs(r))
  extract_raster_values(r, pts)
}
```

`ht_estimator` *Horvitz-Thompson (HT) Estimator*

Description

Provides a design-unbiased estimate of the population total for unequal-probability sampling designs.

Usage

```
ht_estimator(y, pi)
```

Arguments

`y`	Numeric vector. Sample values of the study variable.
`pi`	Numeric vector. First-order inclusion probabilities (same length as `y`). All values must be in (0, 1].

Details

$$\hat{Y}_{HT} = \sum_{i \in s} \frac{y_i}{\pi_i}$$

Value

Numeric scalar. HT estimate of the population total.

References

Horvitz, D.G. and Thompson, D.J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260), 663–685.

Examples

```
data(sample_nrm)
pps <- pps_sample(sample_nrm, size_var = "size", n = 20)
pi_i <- pps$.inclusion_prob
ht_estimator(pps$biomass, pi = pi_i)
```

`ht_variance` *Variance of the Horvitz-Thompson Estimator*

Description

Estimates the variance of the Horvitz-Thompson total estimator using the Sen-Yates-Grundy approximation for with-replacement PPS designs.

Usage

```
ht_variance(y, pi)
```

Arguments

`y`	Numeric vector. Sample values of the study variable.
`pi`	Numeric vector. First-order inclusion probabilities.

Value

Numeric scalar. Estimated variance of the HT total.

References

Yates, F. and Grundy, P.M. (1953). Selection without replacement from within strata with probability proportional to size. *Journal of the Royal Statistical Society B*, 15, 253–261.

Examples

```
data(sample_nrm)
pps <- pps_sample(sample_nrm, size_var = "size", n = 20)
ht_variance(pps$biomass, pps$.inclusion_prob)
```

`plot_sampling` *Plot Sample Points (Base Graphics)*

Description

Produces a simple plot of sample point locations using base graphics.

Usage

```
plot_sampling(
  sf_points,
  col = "steelblue",
  pch = 16,
  main = "Sample Locations"
)
```

Arguments

`sf_points`	An `sf` POINT object.
`col`	Character. Point colour. Default `"steelblue"`.
`pch`	Integer. Point character. Default `16`.
`main`	Character. Plot title. Default `"Sample Locations"`.

Details

Requires the **sf** package.

Value

`NULL` invisibly. Called for its side effect (a plot).

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  bbox <- sf::st_as_sfc(sf::st_bbox(c(xmin=77, xmax=78,
                                      ymin=30, ymax=31),
                         crs = sf::st_crs(4326)))
  pts <- spatial_random_sample(bbox, n = 20)
  plot_sampling(pts)
}
```

`plot_sampling_gg` *Plot Sample Points with ggplot2*

Description

Produces a publication-quality map of sample point locations using **ggplot2**.

Usage

```
plot_sampling_gg(
  sf_points,
  colour = "tomato",
  size = 2,
  title = "Spatial Sample Locations"
)
```

Arguments

`sf_points`	An `sf` POINT object.
`colour`	Character. Point colour. Default `"tomato"`.
`size`	Numeric. Point size. Default `2`.
`title`	Character. Plot title.

Details

Requires the **sf** and **ggplot2** packages.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("sf", quietly = TRUE) &&
    requireNamespace("ggplot2", quietly = TRUE)) {
  bbox <- sf::st_as_sfc(sf::st_bbox(c(xmin=77, xmax=78,
                                     ymin=30, ymax=31),
                          crs = sf::st_crs(4326)))
  pts <- spatial_random_sample(bbox, n = 25)
  plot_sampling_gg(pts)
}
```

`plot_summary` *Summary Statistics for a Sampled NRM Plot Dataset*

Description

Returns a concise summary table of key variables from a sample, including means, standard deviations, and sample sizes.

Usage

```
plot_summary(df, vars = NULL)
```

Arguments

`df`	A data frame (sample or population).
`vars`	Character vector. Names of numeric columns to summarise. If `NULL` (default), all numeric columns are included.

Value

A data frame with columns `variable`, `n`, `mean`, `sd`, `min`, and `max`.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
plot_summary(srs, vars = c("biomass", "soil_loss"))
```

`pps_sample` *PPS Sampling*

Description

Performs probability proportional to size sampling.

Usage

```
pps_sample(data, size_var, n)
```

Arguments

`data`	A data frame.
`size_var`	Character. Size variable.
`n`	Integer. Sample size.

Value

A data frame with inclusion probabilities.

`purposive_sample` *Purposive (Judgement) Sampling*

Description

Selects units from a data frame that satisfy a logical condition supplied as a character string. This is a non-probability method in which units are selected based on expert judgement or predetermined criteria.

Usage

```
purposive_sample(data, condition)
```

Arguments

`data`	A data frame.
`condition`	Character string. A valid R logical expression referring to column names in `data`. Example: `"biomass > 30 & strata == 'forest'"`.

Details

The `condition` is parsed and evaluated in the context of `data` using `with()`. Only columns present in `data` may be referenced.

Value

A data frame of rows satisfying `condition`.

Examples

```
data(sample_nrm)
# Select high-biomass forest plots
ps <- purposive_sample(sample_nrm,
                       condition = "biomass > 30 & strata == 'forest'")
nrow(ps)
```

`quota_sample` *Quota Sampling*

Description

Selects a fixed number of units from the top of each stratum. This non-probability method resembles stratified sampling but does not use random selection within strata.

Usage

```
quota_sample(data, strata_var, quota)
```

Arguments

`data`	A data frame.
`strata_var`	Character. Name of the column defining quota groups.
`quota`	Integer or named integer vector. • A single integer: the same quota applied to every stratum. • A named integer vector: stratum-specific quotas; names must match unique values of `strata_var`.

Value

A data frame containing the selected rows from each stratum.

References

Lohr, S.L. (2022). *Sampling: Design and Analysis*, 3rd ed. CRC Press, Boca Raton, FL.

Examples

```
data(sample_nrm)
qs <- quota_sample(sample_nrm, strata_var = "strata", quota = 5)
table(qs$strata)

# Variable quotas
q <- c(forest = 6, agriculture = 3, grassland = 4)
qs2 <- quota_sample(sample_nrm, "strata", quota = q)
```

`raster_pps_sample` *Raster-Weighted PPS Spatial Sampling*

Description

Selects sample points from a `SpatRaster` with probability proportional to cell values (e.g., vegetation density, erosion risk).

Usage

```
raster_pps_sample(raster, n)
```

Arguments

`raster`	A `SpatRaster` object.
`n`	Integer. Number of sample points.

Details

Requires both the **terra** and **sf** packages. Identical in behaviour to `raster_stratified_sample`; exposed as a separate function to match PPS nomenclature.

Value

An `sf` POINT object.

Examples

```
if (requireNamespace("terra", quietly = TRUE) &&
    requireNamespace("sf", quietly = TRUE)) {
  r <- terra::rast(nrows = 20, ncols = 20, vals = runif(400, 0, 1))
  pts <- raster_pps_sample(r, n = 10)
}
```

`raster_stratified_sample`

Raster-Stratified Spatial Sampling

Description

Samples a specified number of cells from a `SpatRaster` with probability proportional to cell values, and returns their coordinates as an `sf` point object.

Usage

```
raster_stratified_sample(raster, n)
```

Arguments

`raster`	A `SpatRaster` object (from the **terra** package).
`n`	Integer. Number of sample points.

Details

Requires both the **terra** and **sf** packages. Cells with `NA` values are excluded from sampling.

Value

An `sf` POINT object with CRS taken from `raster`.

Examples

```
if (requireNamespace("terra", quietly = TRUE) &&
    requireNamespace("sf", quietly = TRUE)) {
  r <- terra::rast(nrows = 20, ncols = 20, vals = runif(400, 1, 100))
  pts <- raster_stratified_sample(r, n = 15)
  plot(sf::st_geometry(pts))
}
```

`ratio_estimator` *Ratio Estimator*

Description

Estimates the population total or mean of a study variable y using a correlated auxiliary variable x with a known population total X.

Usage

```
ratio_estimator(y, x, X_total)
```

Arguments

`y`	Numeric vector. Sample values of the study variable.
`x`	Numeric vector. Sample values of the auxiliary variable (same length as `y`).
`X_total`	Numeric. Known population total of the auxiliary variable.

Details

$$\hat{R} = \frac{\sum y_i}{\sum x_i}, \qquad \hat{Y}_R = \hat{R} \cdot X$$

Value

Numeric scalar. Ratio estimate of the population total of y.

References

Cochran, W.G. (1977). *Sampling Techniques*, 3rd ed. John Wiley & Sons, New York.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
X_total <- sum(sample_nrm$size)
ratio_estimator(y = srs$biomass, x = srs$size, X_total = X_total)
```

`regression_estimator` *Regression Estimator*

Description

Provides a model-assisted estimate of the population mean of y using a known population mean of the auxiliary variable x.

Usage

```
regression_estimator(y, x, X_mean)
```

Arguments

`y`	Numeric vector. Sample values of the study variable.
`x`	Numeric vector. Sample values of the auxiliary variable.
`X_mean`	Numeric. Known population mean of x.

Details

$$\hat{\bar{Y}}_{\text{reg}} = \bar{y} + \hat{\beta}(\bar{X} - \bar{x})$$

where $\hat{\beta}$ is the ordinary least-squares slope from regressing y on x.

Value

Numeric scalar. Regression estimate of the population mean.

References

Sarndal, C.E., Swensson, B., and Wretman, J. (2003). *Model Assisted Survey Sampling*. Springer, New York.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 30)
X_mean <- mean(sample_nrm$size)
regression_estimator(y = srs$biomass, x = srs$size, X_mean = X_mean)
```

`sample_nrm` *Simulated NRM Plot Dataset*

Description

A synthetic dataset of 100 field plots representing a heterogeneous natural resource management landscape with three strata and ten spatial clusters.

Format

A data frame with 100 rows and 7 variables:

`plot_id` Integer. Unique plot identifier (1–100).

`biomass` Numeric. Aboveground biomass density (Mg/ha), drawn from Uniform(5, 50).

`soil_loss` Numeric. Annual soil loss (Mg/ha/yr), drawn from Uniform(0.1, 10).

`strata` Character. Land-use stratum: one of `"forest"`, `"agriculture"`, or `"grassland"`.

`cluster` Integer. Spatial cluster identifier (1–10).

`size` Numeric. Plot size measure used for PPS sampling (e.g., stand basal area), drawn from Uniform(1, 100).

`carbon` Numeric. Estimated carbon stock (Mg C/ha), derived as `0.47 * biomass`.

Source

Synthetic data generated in `data-raw/generate_datasets.R`.

Examples

```
data(sample_nrm)
head(sample_nrm)
table(sample_nrm$strata)
summary(sample_nrm[, c("biomass", "soil_loss")])
```

`sample_spatial` *Simulated Spatial NRM Dataset*

Description

A synthetic dataset of 100 geo-referenced field observations within a one-degree tile (77–78 degrees E, 30–31 degrees N) representing a Himalayan watershed zone.

Format

A data frame with 100 rows and 8 variables:

- `id` Integer. Observation identifier.
- `lon` Numeric. Longitude (decimal degrees, WGS 84).
- `lat` Numeric. Latitude (decimal degrees, WGS 84).
- `biomass` Numeric. Aboveground biomass (Mg/ha).
- `soil_loss` Numeric. Annual soil loss (Mg/ha/yr).
- `strata` Character. Land-use class: `"forest"` or `"agriculture"`.
- `cluster` Integer. Spatial cluster (1–5).
- `ndvi` Numeric. Synthetic NDVI value (0–1), correlated with biomass.

Source

Synthetic data generated in `data-raw/generate_datasets.R`.

Examples

```
data(sample_spatial)
head(sample_spatial)

if (requireNamespace("sf", quietly = TRUE)) {
  pts <- to_sf_points(sample_spatial, lon = "lon", lat = "lat")
  plot(sf::st_geometry(pts))
}
```

`sampling_efficiency` *Sampling Efficiency Comparison*

Description

Computes the relative efficiency (RE) of two sampling designs by comparing their estimated variances of the mean. RE > 1 indicates Design 2 is more efficient than Design 1.

Usage

```
sampling_efficiency(y1, y2, N = NULL)
```

Arguments

`y1`	Numeric vector. Sample from Design 1.
`y2`	Numeric vector. Sample from Design 2.
`N`	Integer. Population size (for fpc). Set `NULL` to omit.

Value

A named numeric vector with `var_design1`, `var_design2`, and `relative_efficiency` (var1 / var2).

Examples

```
data(sample_nrm)
srs1 <- srs_sample(sample_nrm, n = 20)
srs2 <- srs_sample(sample_nrm, n = 30)
sampling_efficiency(srs1$biomass, srs2$biomass, N = 100)
```

`soil_loss_estimate` *Soil Loss Estimation from Sample Plots*

Description

Estimates mean and total soil loss from a set of erosion measurement plots, with a standard error.

Usage

```
soil_loss_estimate(df, loss_var, area)
```

Arguments

`df`	A data frame of sampled erosion plots.
`loss_var`	Character. Name of the column containing plot-level soil loss measurements (e.g., Mg/ha/year).
`area`	Numeric. Total catchment or management area.

Value

A named list with elements `mean_loss`, `total_loss`, `se`, and `n`.

References

Wischmeier, W.H. and Smith, D.D. (1978). *Predicting Rainfall Erosion Losses*. USDA Agriculture Handbook 537.

Examples

```
data(sample_nrm)
srs <- srs_sample(sample_nrm, n = 25)
soil_loss_estimate(srs, loss_var = "soil_loss", area = 500)
```

`spatial_biomass_estimate`

Spatial Biomass Estimation from sf Points

Description

Estimates total biomass over a study area from field measurements stored in an `sf` point object.

Usage

```
spatial_biomass_estimate(sf_data, biomass_var, area)
```

Arguments

`sf_data`	An `sf` object with a biomass attribute column.
`biomass_var`	Character. Name of the biomass density column.
`area`	Numeric. Total study area (in consistent units).

Details

Requires the **sf** package.

Value

A named list with `mean_biomass`, `total_biomass`, `se`, and `n`.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  data(sample_spatial)
  pts_sf <- to_sf_points(sample_spatial, lon = "lon", lat = "lat")
  spatial_biomass_estimate(pts_sf, biomass_var = "biomass", area = 1000)
}
```

`spatial_cluster_sample` *Spatial Cluster Sampling*

Description

Randomly selects a number of spatial clusters (e.g., sub-watersheds, administrative units) and returns all features within selected clusters.

Usage

```
spatial_cluster_sample(sf_data, cluster_var, n_clusters)
```

Arguments

`sf_data`	An `sf` object.
`cluster_var`	Character. Name of the column identifying clusters.
`n_clusters`	Integer. Number of clusters to select.

Details

Requires the **sf** package.

Value

An `sf` object containing features from the selected clusters.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  data(sample_spatial)
  pts_sf <- to_sf_points(sample_spatial, lon = "lon", lat = "lat")
  cl_sp  <- spatial_cluster_sample(pts_sf, "cluster", n_clusters = 3)
  length(unique(cl_sp$cluster))
}
```

`spatial_random_sample` *Spatial Random Sampling within a Polygon*

Description

Draws a random sample of points uniformly distributed within an `sf` polygon or multipolygon geometry.

Usage

```
spatial_random_sample(polygon, n)
```

Arguments

`polygon`	An `sf` or `sfc` object representing the sampling frame (polygon boundary).
`n`	Integer. Number of random points to generate.

Details

Requires the **sf** package. Points are drawn using `sf::st_sample(..., type = "random")`.

Value

An `sf` object of POINT geometries within `polygon`.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  # Create a simple rectangular polygon for illustration
  bbox <- sf::st_as_sfc(sf::st_bbox(c(xmin=77, xmax=78,
                                       ymin=30, ymax=31),
                        crs = sf::st_crs(4326)))
  pts <- spatial_random_sample(bbox, n = 20)
  plot(sf::st_geometry(pts))
}
```

`spatial_stratified_sample`

Stratified Spatial Sampling

Description

Selects a random sample of `n_per_stratum` features from each stratum of an `sf` object.

Usage

```
spatial_stratified_sample(sf_data, strata_var, n_per_stratum)
```

Arguments

`sf_data`	An `sf` object with a stratum attribute column.
`strata_var`	Character. Name of the column defining strata.
`n_per_stratum`	Integer or named integer vector. Sample size per stratum (see `stratified_sample` for details).

Details

Requires the **sf** package.

Value

An `sf` object with the selected features.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  data(sample_spatial)
  pts_sf <- to_sf_points(sample_spatial, lon = "lon", lat = "lat")
  st_sp  <- spatial_stratified_sample(pts_sf, "strata", n_per_stratum = 5)
  table(st_sp$strata)
}
```

`spatial_systematic_sample` *Systematic Grid Sampling within a Polygon*

Description

Generates a systematic grid of points at a specified spacing within an `sf` polygon, retaining only points that fall inside the boundary.

Usage

```
spatial_systematic_sample(polygon, spacing)
```

Arguments

`polygon`	An `sf` or `sfc` polygon object.
`spacing`	Numeric. Grid cell size in the units of the CRS (degrees for EPSG:4326, metres for projected CRS).

Details

Requires the **sf** package.

Value

An `sf` object of POINT geometries inside `polygon`.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  bbox <- sf::st_as_sfc(sf::st_bbox(c(xmin=77, xmax=78,
                                       ymin=30, ymax=31),
                         crs = sf::st_crs(4326)))
  grid_pts <- spatial_systematic_sample(bbox, spacing = 0.1)
  plot(sf::st_geometry(grid_pts))
}
```

`srs_sample` *Simple Random Sampling*

Description

Draws a simple random sample from a data frame.

Usage

```
srs_sample(data, n, replace = FALSE)
```

Arguments

`data`	A data frame representing the population.
`n`	Integer. Number of units to sample.
`replace`	Logical. Sample with replacement? Default FALSE.

Value

A data frame with sampled rows and a ".sample_id" column.

`stratified_estimator` *Stratified Mean Estimator*

Description

Estimates the population mean from a stratified sample using stratum weights (proportional to stratum size).

Usage

```
stratified_estimator(y, strata, N_h)
```

Arguments

`y`	Numeric vector. Sample values of the study variable.
`strata`	Character or factor vector. Stratum labels (same length as `y`).
`N_h`	Named numeric vector. Population stratum sizes; names must match unique values of `strata`.

Details

$$\hat{\bar{Y}}_{st} = \sum_h W_h \bar{y}_h, \qquad W_h = N_h/N$$

Value

Numeric scalar. Estimated population mean.

References

Cochran, W.G. (1977). *Sampling Techniques*, 3rd ed. John Wiley & Sons, New York.

Examples

```
data(sample_nrm)
st <- stratified_sample(sample_nrm, "strata", n_per_stratum = 5)
N_h <- table(sample_nrm$strata)
stratified_estimator(st$biomass, st$strata, N_h)
```

`stratified_sample` *Stratified Sampling*

Description

Performs stratified random sampling.

Usage

```
stratified_sample(data, strata_var, n_per_stratum, replace = FALSE)
```

Arguments

`data`	A data frame.
`strata_var`	Character. Column defining strata.
`n_per_stratum`	Integer or named vector.
`replace`	Logical. Sample with replacement?

Value

A data frame with sampled rows.

`systematic_sample` *Systematic Sampling*

Description

Performs systematic sampling using interval k.

Usage

```
systematic_sample(data, k)
```

Arguments

`data`	A data frame.
`k`	Integer. Sampling interval.

Value

A data frame of sampled rows.

`to_sf_points` *Convert a Data Frame to an sf Point Object*

Description

Creates an `sf` simple-features point object from longitude and latitude columns in a data frame.

Usage

```
to_sf_points(data, lon, lat, crs = 4326)
```

Arguments

`data`	A data frame containing coordinate columns.
`lon`	Character. Name of the longitude column.
`lat`	Character. Name of the latitude column.
`crs`	Integer or character. Coordinate reference system as an EPSG code or PROJ string. Default `4326` (WGS 84 geographic).

Details

Requires the **sf** package.

Value

An `sf` object with a POINT geometry column.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  data(sample_spatial)
  pts <- to_sf_points(sample_spatial, lon = "lon", lat = "lat")
  print(pts)
}
```

Index