Package 'spsurvey'

September 18, 2020

Title Spatial Survey Design and Analysis

Version 4.1.4

Date 2020-9-15

Maintainer Marc Weber <Weber.Marc@epa.gov>

Depends R (>= 3.5.0), methods

Suggests knitr, testthat, rmarkdown, lwgeom (>= 0.2-1)

Description These functions provide procedures for selecting sites for spatial surveys using spatially balanced algorithms applied to discrete points, linear networks, or polygons. The probability survey designs available include independent random samples, stratified random samples, and unequal probability random samples (categorical or probability proportional to size). Design-based estimation based on the results from surveys is available for estimating totals, means, quantiles, CDFs, and linear models. The analyses rely on package survey for most results. Variance estimation options include a local neighborhood variance estimator that is appropriate for spatially-balanced survey designs. A reference for the survey design portion of the package is: D. L. Stevens, Jr. and A. R. Olsen (2004), ``Spatially-balanced sampling of natural resources.", Journal of the American Statistical Association 99(465): 262-278, <DOI:10.1198/016214504000000250>. Additional helpful references for this package are A. R. Olsen, T. M. Kincaid, and Q. Payton (2012) and T. M. Kincaid and A. R. Olsen (2012), both of which are chapters in the book ``Design and Analysis of Long-Term Ecological Monitoring Studies" (R. A. Gitzen, J. J. Millspaugh, A. B. Cooper, and D. S. Licht (eds.), Cambridge University Press, New York, <Online ISBN:9781139022422>).

License GPL (>= 3)

Imports crossdes, deldir, foreign, graphics, grDevices, Hmisc, MASS, parallel, rgeos, stats, sf, sp

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Tom Kincaid [aut],
Tony Olsen [aut],
Marc Weber [aut, cre],
Don Stevens [ctb],
Christian Platt [ctb],
Denis White [ctb],
Richard Remington [ctb]

Repository CRAN

Date/Publication 2020-09-18 08:20:03 UTC

R topics documented:

spsurvey-package *spsurvey: Spatial Survey Design and Analysis*

Description

This package provides functions for design and analysis of probability surveys. The functions in spsurvey can select generalized random-tesselation stratified (GRTS) and independent random sample (IRS) survey designs. Although the function can be used with a wide range of environmental survey designs, the spsurvey analysis functions were written to accommodate data generated by a GRTS sampling design. The functions in spsurvey are applicable to finite (discrete units, zero-dimensional), linear (one-dimensional), and areal (two-dimensional) resources. Examples of these resource are lakes in the United States (a finite resource), rivers and streams in Oregon (a linear resource), and Chesapeake Bay (an areal resource). The design functions can select stratified and unstratified sampling designs. The analysis functions can accommodate stratified and unstratified designs, both of which can utilize single-stage or two-stage sampling. Analytical capabilities accommodate both categorical and continuous data. For categorical data, estimates of proportion and size of each category (class) can be obtained. For a finite resource, size is the number of units in the resource. For an extensive (linear or areal) resource, size is the measure (extent) of the resource, i.e., length, area, or volume. In addition, for categorical data that contains bivariate (two categories) response variables and bivariate explanatory (stressor) variables, relative risk estimates and attributable risk estimates can be calculated. For continuous data, estimates of the cumulative distribution function (CDF) and percentiles can be obtained in addition to estimation of the population mean, total, variance, and standard deviation. Optionally, for continuous data, estimation of the deconvoluted CDF and estimation of percentiles using the deconvoluted CDF are available.

Author(s)

Maintainer: Marc Weber <Weber.Marc@epa.gov>

Authors:

- Tom Kincaid <Kincaid.Tom@epa.gov>
- Tony Olsen <Olsen.Tony@epa.gov>

Other contributors:

- Don Stevens [contributor]
- Christian Platt [contributor]
- Denis White [contributor]
- Richard Remington [contributor]

adjwgt *Adjust Survey Design Weights by Categories*

Description

Purpose of this function is to adjust initial survey design weights when implementation results in use of oversample sites or when it is desired to have final weights sum to known frame size. Adjusted weights are equal to initial weight * framesize/sum(initial weights). The adjustment is done separately for each category specified in wtcat.

Usage

```
adjwgt(sites, wgt, wtcat, framesize)
```

Arguments

sites	Vector of the logical value for each site, where TRUE = include the site and FALSE = do not include the site.
wgt	Vector of the initial weight (inverse of the sample inclusion probability) for each site.
wtcat	Vector of the weight adjustment category name for each site.
framesize	Vector of the known size of the frame for each category name in wtcat, which must have the names attribute set to match the category names used in wtcat.

Value

Vector of adjusted weights, where the adjusted weight is set to zero for sites that have the logical value in the sites argument set to FALSE.

Author(s)

Tony Olsen <olsen.tony@epa.gov>

Examples

```
sites <- as.logical(rep(rep(c("TRUE","FALSE"), c(9,1)), 5))
wgt <- runif(50, 10, 100)
wtcat <- rep(c("A","B"), c(30, 20))
framesize <- c(15, 10)
names(framesize) <- c("A","B")
adjwgt(sites, wgt, wtcat, framesize)
```

`albersgeod` *Project Albers Projection in Plane to Latitude and Longitude (Spheroid)*

Description

Convert x-coordinates and y-coordinates given in Albers projection to latitude and longitude in Clarke1866, GRS80 or WGS84 spheriod with specified parameters.

Usage

```
albersgeod(x, y, sph = "GRS80", clon = -96, clat = 23, sp1 = 29.5,
  sp2 = 45.5)
```

Arguments

`x`	Vector of Albers x-coordinates to be projected to latitude/longitude.
`y`	Vector of Albers y-coordinates to be projected to latitude/longitude.
`sph`	Spheroid options: Clarke1866, GRS80, WGS84. The default is GRS80.
`clon`	Center longitude (decimal degrees). The default is -96.
`clat`	Origin latitude (decimal degrees). The default is 23.
`sp1`	Standard parallel 1 (decimal degrees). The default is 29.5.
`sp2`	Standard parallel 2 (decimal degrees). The default is 45.5.

Value

A data frame of latitude and longitude projections for Albers x-coordinates and y-coordinates

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

References

J. Snyder, USGS Professional Paper 1395.

ash1.wgt *Compute the Average Shifted Histogram (ASH) for Weighted Data*

Description

Calculate the average shifted histogram estimate of a density based on data from a survey design with weights.

Usage

```
ash1.wgt(x, wgt = rep(1, length(x)), m = 5, nbin = 50, ab = NULL,
  support = "Continuous")
```

Arguments

x	Vector of data used to estimate the density. NAs are allowed
wgt	Vector of Weights for each observation from a probability sample. The default is equal weights (equal probability).
m	The number of empty bins to add to the ends when the range is not completely specified. The default is 5.
nbin	The number of bins for density estimation. The default is 50.
ab	Optional range for support associated with the density. Both values may be equal to NA. If equal to NA, then corresponding limit will be based on nicerange(). The default is NULL.
support	The type of support. If equal to "Continuous", then data are from a continuous distribution. If equal to "Ordinal", then data are from a discrete distribution defined for integers only. The default is "Continuous".

Value

A list containing the ASH density estimate. List consists of

- tcen - x-coordinate for center of bin
- f - y-coordinate for density estimate height

Author(s)

Tony Olsen <Olsen.tony@epa.gov>

References

Scott, D. W. (1985). "Averaged shifted histograms: effective nonparametric density estimators in several dimensions." The Annals of Statistics 13(3): 1024-1040.

Examples

```
x <- rnorm(100, 10, sqrt(10))
wgt <- runif(100, 10, 100)
rslt <- ash1.wgt(x, wgt)
plot(rslt)
```

`attrisk.analysis` *Attributable Risk Analysis for Probability Survey Data*

Description

This function organizes input and output for attributable risk analysis of categorical data generated by a probability survey.

Usage

```
attrisk.analysis(sites = NULL, subpop = NULL, design, data.ar,
  response.var, stressor.var, response.levels = rep(list(c("Poor",
  "Good")), length(response.var)), stressor.levels = rep(list(c("Poor",
  "Good")), length(stressor.var)), popcorrect = FALSE, pcfsize = NULL,
  N.cluster = NULL, stage1size = NULL, sizeweight = FALSE,
  vartype = "Local", conf = 95)
```

Arguments

`sites` Data frame consisting of two variables: the first variable is site IDs, and the second variable is a logical vector indicating which sites to use in the analysis. The default is NULL.

`subpop` Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is site IDs. Each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. The default is NULL.

`design` Data frame consisting of design variables. Variables should be named as follows:

siteID Vector of site IDs

wgt Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample

xcoord Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample

ycoord Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample

stratum Vector of the stratum codes for each site

cluster Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site

wgt1 Vector of stage one weights in a two-stage design

xcoord1 Vector of the stage one x-coordinates for location in a two-stage design

ycoord1 Vector of the stage one y-coordinates for location in a two-stage design

support Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors.

swgt Vector of size-weights, which is the stage two size-weight for a two-stage design.

swgt1 Vector of stage one size-weights for a two-stage design.

`data.ar` Data frame of categorical response and stressor variables, where each variable consists of two categories. If response or stressor variables include more than two categories, occurrences of those categories must be removed or replaced with missing values. The first column of this argument is site IDs. Subsequent columns are response and stressor variables. Missing data (NA) is allowed.

`response.var` Character vector providing names of columns in argument data.ar that contain a response variable, where names may be repeated. Each name in this argument is matched with the corresponding value in the stressor.var argument.

`stressor.var` Character vector providing names of columns in argument data.ar that contain a stressor variable, where names may be repeated. Each name in this argument is matched with the corresponding value in the response.var argument. This argument must be the same length as argument response.var.

`response.levels`
List providing the category values (levels) for each element in the response.var argument. This argument must be the same length as argument response.var. The default is a list containing the values "Poor" and "Good" for the first and second levels, respectively, of each element in the response.var argument.

`stressor.levels`
List providing the category values (levels) for each element in the stressor.var argument. This argument must be the same length as argument response.var. The default is a list containing the values "Poor" and "Good" for the first and second levels, respectively, of each element in the stressor.var argument.

`popcorrect` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster and stage1size, and for the support variable of the design argument.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample

this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`sizeweight` Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

`vartype` The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

`conf` Numeric value for the confidence level. The default is 95.

Value

A data frame of attributable risk estimates for all combinations of population Types, subpopulations within Types, and response variables. Standard error and confidence interval estimates also are provided.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

References

Sarndal, C.E., B. Swensson, and J. Wretman. (1992). *Model Assisted Survey Sampling*. Springer-Verlag, New York.

See Also

`attrisk.est` computes the attributable risk estimate

`dframe.check` check site IDs, the sites data frame, the subpop data frame, and the data.ar data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

`uniqueID` creates unique site IDs by appending a unique number to each occurrence of a site ID

`input.check` check input values for errors, consistency, and compatibility with psurvey.analysis analytical functions

Examples

```
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(TRUE, 100))
mysubpop <- data.frame(
  siteID=mysiteID,
  All.Sites=rep("All Sites", 100),
  Resource.Class=rep(c("Agr", "Forest"), c(55,45)))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1", "Stratum2"), 50))
mydata.ar <- data.frame(
  siteID=mysiteID,
  RespVar1=sample(c("Poor", "Good"), 100, replace=TRUE),
  RespVar2=sample(c("Poor", "Good"), 100, replace=TRUE),
  StressVar=sample(c("Poor", "Good"), 100, replace=TRUE))
attrisk.analysis(sites=mysites, subpop=mysubpop, design=mydesign,
  data.ar=mydata.ar, response.var=c("RespVar1", "RespVar2"),
  stressor.var=rep("StressVar", 2))
```

---

attrisk.est *Compute the Attributable Risk Estimate*

Description

This function calculates the attributable risk estimate for a 2x2 table of cell counts defined by a categorical response variable and a categorical explanatory (stressor) variable for an unequal probability design (Van Sickle and Paulsen, 2008). The attributable risk of the stressor variable is the percent reduction in the first level of the response variable that would result from elimination of the stressor variable. Cell totals are estimated using the Horvitz-Thompson estimator. The standard error of the log of the attributable risk estimate and confidence limits for the estimate also are calculated. The standard error is calculated using a first-order Taylor series linearization (Sarndal et al, 1992).

Usage

```
attrisk.est(response, stressor, response.levels = c("Poor", "Good"),
  stressor.levels = c("Poor", "Good"), wgt, xcoord = NULL,
  ycoord = NULL, stratum = NULL, cluster = NULL, wgt1 = NULL,
  xcoord1 = NULL, ycoord1 = NULL, popcorrect = FALSE,
  pcfsize = NULL, N.cluster = NULL, stage1size = NULL,
  support = NULL, sizeweight = FALSE, swgt = NULL, swgt1 = NULL,
  vartype = "Local", conf = 95, check.ind = TRUE, warn.ind = NULL,
  warn.df = NULL, warn.vec = NULL)
```

Arguments

response	Vector of the categorical response variable values.
stressor	Vector of the categorical explanatory (stressor) variable values.
response.levels	Vector of category values (levels) for the categorical response variable, where the first level is used for calculating the numerator and the denominator of the attributable risk estimate. If response.levels is not supplied, then values "Poor" and "Good" are used for the first level and second level of the response variable, respectively. The default is c("Poor", "Good").
stressor.levels	Vector of category values (levels) for the categorical stressor variable, where the first level is used for calculating the numerator of the attributable risk estimate and the second level is used for calculating the denominator of the estimate. If stressor.levels is not supplied, then values "Poor" and "Good" are used for the first level and second level of the stressor variable, respectively. The default is c("Poor", "Good").
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
xcoord	Vector of x-coordinates for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
ycoord	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
stratum	Vector of the stratum values for each site. The default is NULL.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site. The default is NULL.
wgt1	Vector of the final adjusted stage one weight for each site. The default is NULL.
xcoord1	Vector of the stage one x-coordinate for location for each site. The default is NULL.
ycoord1	Vector of the stage one y-coordinate for location for each site. The default is NULL.
popcorrect	Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	The size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.
stage1size	The size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.
support	The support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.
sizeweight	Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.
swgt	The size-weight for each site, which is the stage two size-weight for two-stage sample. The default is NULL.
swgt1	The stage one size-weight for each site. The default is NULL.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
conf	Numeric value for the confidence level. The default is 95.
check.ind	A logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	A logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	A data frame for storing warning messages. The default is NULL.
warn.vec	A vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called directly, then value is the Results list, which contains the following components:

- AttRisk - the attributable risk estimate
- ARlog.se - standard error for the log of the attributable risk estimate
- ConfLimits - confidence limits for the attributable risk estimate
- WeightTotal - sum of the final adjusted weights
- CellCounts - cell and margin counts for the 2x2 table

- `CellProportions` - estimated cell proportions for the 2x2 table

If the function was called by the attrisk.analysis function, then the result is a list containing the following components:

- `Results` - Results list. See above for list contents
- `warn.ind` - a logical value indicating whether warning messages were generated
- `warn.df` - a data frame containing warning messages

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

References

Van Sickle, J. and S. G. Paulsen. (2008). Assessing the attributable risks, relative risks, and regional extent of aquatic stressors. *Journal of the North American Benthological Society* **27**, 920-931.

Sarndal, C.E., B. Swensson, and J. Wretman. (1992). *Model Assisted Survey Sampling*. Springer-Verlag, New York.

Examples

```
response <- sample(c("Poor", "Good"), 100, replace=TRUE)
stressor <- sample(c("Poor", "Good"), 100, replace=TRUE)
wgt <- runif(100, 10, 100)
attrisk.est(response, stressor, wgt=wgt, vartype="SRS")

xcoord <- runif(100)
ycoord <- runif(100)
attrisk.est(response, stressor, wgt=wgt, xcoord=xcoord, ycoord=ycoord)
```

---

attrisk.var *Compute the Variance Estimate for Attributable Risk*

Description

This function calculates the variance-covariance estimate for the cell totals used to calculate the attributable risk estimate. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples.

Usage

```
attrisk.var(response, stressor, response.levels, stressor.levels, wgt, x,
  y, stratum.ind, stratum.level, cluster.ind, cluster, wgt1, x1, y1,
  pcfactor.ind, pcfsize, N.cluster, stage1size, support, vartype, warn.ind,
  warn.df, warn.vec)
```

Arguments

response	Vector of the categorical response variable.
stressor	Vector of the categorical stressor variable.
response.levels	Vector of category values (levels) for the categorical response variable. If response.levels equals NULL, then values "Poor" and "Good" are used for the first level and second level of the response variable, respectively. The default is NULL.
stressor.levels	Vector of category values (levels) for the categorical stressor variable. If stressor.levels equals NULL, then values "Poor" and "Good" are used for the first level and second level of the stressor variable, respectively. The default is NULL.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	Vector of the stratum levels for the sites.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	A data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains the variance-covariance estimate, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.cov calculate the variance/covariance matrix using the local mean estimator

localmean.weight calculate the weighting matrix for the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

See Also

attrisk.est for estimating single relative risk and attrisk.analysis for estimating relative risk for multiple variables or subpopulations.

`cat.analysis` *Categorical Data Analysis for Probability Survey Data*

Description

This function organizes input and output for analysis of categorical data generated by a probability survey. Input can be either an object of class spsurvey.analysis (see the documentation for function spsurvey.analysis) or through use of the other arguments to this function.

Usage

```
cat.analysis(sites = NULL, subpop = NULL, design = NULL,
  data.cat = NULL, popsize = NULL, popcorrect = FALSE,
  pcfsize = NULL, N.cluster = NULL, stage1size = NULL,
  sizeweight = FALSE, vartype = "Local", conf = 95,
  spsurvey.obj = NULL)
```

Arguments

`sites` Data frame consisting of two variables: the first variable is site IDs, and the second variable is a logical vector indicating which sites to use in the analysis. If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`subpop` Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is site IDs. Each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`design` Data frame consisting of design variables. If spsurvey.obj is not provided, then this argument is required. The default is NULL. Variables should be named as follows:

siteID Vector of site IDs

wgt Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample

xcoord Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample

ycoord Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample

stratum Vector of the stratum codes for each site

cluster Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site

wgt1 Vector of stage one weights in a two-stage design

xcoord1 Vector of the stage one x-coordinates for location in a two-stage design

ycoord1 Vector of the stage one y-coordinates for location in a two-stage design

support Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors.

swgt Vector of size-weights, which is the stage two size-weight for a two-stage design.

swgt1 Vector of stage one size-weights for a two-stage design.

`data.cat` Data frame of categorical response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. If sp-survey.obj is not provided, then this argument is required. The default is NULL.

`popsize` Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource and to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. The argument must be in the form of a list containing an element for each population Type in the subpop data frame, where NULL is a valid choice for a population Type. The list must be named using the column names for the population Types in subpop. If a population Type doesn't contain subpopulations, then each element of the list is either a single value for an unstratified sample or a vector containing a value for each stratum for a stratified sample, where elements of the vector are named using the stratum codes. If a population Type contains subpopulations, then each element of the list is a list containing an element for each subpopulation, where the list is named using the subpopulation names. The element for each subpopulation will be either a single value for an unstratified sample or a named vector of values for a stratified sample. The default is NULL.

Example popsize for a stratified sample:
popsize = list("Pop 1"=c("Stratum 1"=750,
"Stratum 2"=500,
"Stratum 3"=250),
"Pop 2"=list("SubPop 1"=c("Stratum 1"=350,
"Stratum 2"=250,
"Stratum 3"=150),
"SubPop 2"=c("Stratum 1"=250,
"Stratum 2"=150,
"Stratum 3"=100),
"SubPop 3"=c("Stratum 1"=150,
"Stratum 2"=150,
"Stratum 3"=75)),
"Pop 3"=NULL)

Example popsize for an unstratified sample:
popsize = list("Pop 1"=1500,

"Pop 2"=list("SubPop 1"=750,
"SubPop 2"=500,
"SubPop 3"=375),
"Pop 3"=NULL)

popcorrect — Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster and stage1size, and for the support variable of the design argument.

pcfsize — Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

N.cluster — Number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

stage1size — Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

sizeweight — Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

vartype — The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

conf — Numeric value for the confidence level. The default is 95.

spsurvey.obj — List of class spsurvey.analysis that was produced by the function spsurvey.analysis. Depending on input to that function, some elements of the list may be NULL. The default is NULL.

Value

A data frame of population estimates for all combinations of population Types, subpopulations within Types, response variables, and categories within each response variable. Estimates are provided for proportion and size of the population plus standard error and confidence interval estimates.

Other Functions Required

`dframe.check` check site IDs, the sites data frame, the subpop data frame, and the data.cat data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

`vecprint` takes an input vector and outputs a character string with line breaks inserted

`uniqueID` creates unique site IDs by appending a unique number to each occurrence of a site ID

`input.check` check input values for errors, consistency, and compatibility with analytical functions

`category.est` estimate proportion (expressed as percent) and size of a resource in each of a set of categories

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>
Tom Kincaid <Kincaid.Tom@epa.gov>

References

Diaz-Ramos, S., D.L. Stevens, Jr., and A.R. Olsen. (1996). *EMAP Statistical Methods Manual.* EPA/620/R-96/XXX. Corvallis, OR: U.S. Environmental Protection Agency, Office of Research and Development, National Health Effects and Environmental Research Laboratory, Western Ecology Division.

See Also

`category.est`

Examples

```
# Categorical variable example for two resource classes:
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(TRUE, 100))
mysubpop <- data.frame(
  siteID=mysiteID,
  All.Sites=rep("All Sites", 100),
  Resource.Class=rep(c("Good","Poor"), c(55,45)))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1", "Stratum2"), 50))
mydata.cat <- data.frame(
  siteID=mysiteID,
  CatVar=rep(c("north", "south", "east", "west"), 25))
mypopsize <- list(
  All.Sites=c(Stratum1=3500, Stratum2=2000),
```

```
  Resource.Class=list(Good=c(Stratum1=2500, Stratum2=1500),
                      Poor=c(Stratum1=1000, Stratum2=500)))
cat.analysis(sites=mysites, subpop=mysubpop, design=mydesign,
  data.cat=mydata.cat, popsize=mypopsize)

# Exclude category "south" from the analysis:
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(c(TRUE, FALSE, TRUE, TRUE), 25))
cat.analysis(sites=mysites, subpop=mysubpop, design=mydesign,
  data.cat=mydata.cat, popsize=mypopsize)
```

`category.est` *Category Proportion and Size Estimates*

Description

This function estimates proportion (expressed as percent) and size of a resource in each of a set of categories and can also be used to estimate proportion and size for site status categories. Upper and lower confidence bounds also are estimated. Proportion estimates are calculated using the Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators. The numerator of the ratio estimates the size of the category. The denominator of the ratio estimates the size of the resource. Variance estimates for the proportion estimates are calculated using either the local mean variance estimator or the simple random sampling (SRS) variance estimator. The choice of variance estimator is subject to user control. The local mean variance estimator requires the x-coordinate and the y-coordinate of each site. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. Confidence bounds are calculated using a Normal distribution multiplier. For a finite resource size is the number of units in the resource. For an extensive resource size is the measure (extent) of the resource, i.e., length, area, or volume. Size estimates are calculated using the Horvitz-Thompson estimator. Variance estimates for the size estimates are calculated using either the local mean variance estimator or the SRS variance estimator. The function can accommodate a stratified sample. For a stratified sample, separate estimates and standard errors are calculated for each stratum, which are used to produce estimates and standard errors for all strata combined. Strata that contain a single value are removed. For a stratified sample, when either the size of the resource or the sum of the size-weights for the resource is provided for each stratum, those values are used as stratum weights for calculating the estimates and standard errors for all strata combined. In addition, when either of those known values is provided for each stratum, size estimates are obtained by multiplying the proportion estimate, i.e., the Horvitz- Thompson ratio estimator, by the known value for the stratum. For a stratified sample when neither the size of the resource nor the sum of the size-weights of the resource is provided for each stratum, estimated values are used as stratum weights for calculating the estimates and standard errors for all strata combined. The function can accommodate single-stage and two-stage samples for both stratified and unstratified sampling designs. Finite population and continuous population correction factors can be utilized in variance estimation. The function checks for compatibility of input values and removes missing values.

Usage

```
category.est(catvar, wgt, x = NULL, y = NULL, stratum = NULL,
  cluster = NULL, wgt1 = NULL, x1 = NULL, y1 = NULL,
  popsize = NULL, popcorrect = FALSE, pcfsize = NULL,
  N.cluster = NULL, stage1size = NULL, support = NULL,
  sizeweight = FALSE, swgt = NULL, swgt1 = NULL, vartype = "Local",
  conf = 95, check.ind = TRUE, warn.ind = NULL, warn.df = NULL,
  warn.vec = NULL)
```

Arguments

catvar	Vector of the value of the categorical response variable or the site status for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
stratum	Vector of the stratum for each site. The default is NULL.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site. The default is NULL.
wgt1	Vector of the final adjusted stage one weight for each site. The default is NULL.
x1	Vector of the stage one x-coordinate for location for each site. The default is NULL.
y1	Vector of the stage one y-coordinate for location for each site. The default is NULL.
popsize	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource and to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.
popcorrect	Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.
N.cluster	Number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.
sizeweight	Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample. The default is NULL.
swgt1	Vector of the stage one size-weight for each site. The default is NULL.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
conf	Numeric value for the confidence level. The default is 95.
check.ind	Logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	Data frame for storing warning messages. The default is NULL.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called by the cat.analysis function, then output is an object in list format composed of the Results data frame, which contains estimates and confidence bounds, and the warn.df data frame, which contains warning messages. If the function was called directly, then output is the Results data frame.

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

catvar.prop calculate variance of the proportion estimates

catvar.size calculate variance of the size estimates

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

References

Diaz-Ramos, S., D.L. Stevens, Jr., and A.R. Olsen. (1996). *EMAP Statistical Methods Manual.* EPA/620/R-96/XXX. Corvallis, OR: U.S. Environmental Protection Agency, Office of Research and Development, National Health Effects and Environmental Research Laboratory, Western Ecology Division.

Examples

```
catvar <- rep(c("north", "south", "east", "west"), rep(25, 4))
wgt <- runif(100, 10, 100)
category.est(catvar, wgt, vartype="SRS")

x <- runif(100)
y <- runif(100)
category.est(catvar, wgt, x, y)
```

---

catvar.prop *Variance Estimates of Estimated Proportions*

Description

This function calculates variance estimates of the estimated proportion in each of a set of categories. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples.

Usage

```
catvar.prop(z, wgt, x, y, prop, stratum.ind, stratum.level, cluster.ind,
  cluster, wgt1, x1, y1, pcfactor.ind, pcfsize, N.cluster, stage1size,
  support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

z	Vector of the value of the categorical response variable or the site status for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of the x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of the y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
prop	Vector of the proportion estimates.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two-stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Lgical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".

support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <kincaid.tom@epa.gov>

---

catvar.size *Variance Estimates of the Estimated Size for Categorical Data*

Description

This function calculates variance estimates of the estimated size in each of a set of categories. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples.

Usage

```
catvar.size(z, wgt, x, y, size, stratum.ind, stratum.level, cluster.ind,
  cluster, wgt1, x1, y1, pcfactor.ind, pcfsize, N.cluster, stage1size,
  support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

z	Vector of the values of the categorical response variable or the site status for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vecteor of the x-coordinates for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of the y-coordinates for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
size	Vector of the size values for each site.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	Vector of the stratum for each site.
cluster.ind	Logical value that indicates whether the sample is a two-stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling units (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinates for location for each site.
y1	Vector of the stage one y-coordinates for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".

support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	A data frame for storing warning messages.
warn.vec	A vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <kincaid.tom@epa.gov>

---

cdf.decon *Deconvolution Estimate of the Cumulative Distribution Function*

Description

This function calculates an estimate of the deconvoluted cumulative distribution function (CDF) for the proportion (expressed as percent) and the total of a response variable, where the response variable may be defined for either a finite or an extensive resource. Optionally, for a finite resource, the size-weighted CDF can be calculated. In addition the standard error of the estimated CDF and confidence bounds are calculated. The simulation extrapolation deconvolution method (Stefanski and Bay, 1996) is used to deconvolute measurement error variance from the response. The user can supply the set of values at which the CDF is estimated. For the CDF of a proportion, the Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the CDF estimate. For the CDF of a total, the user can supply the known size of the resource or the known sum of the size-weights of the resource, as appropriate. For the CDF of a total when either the size of the resource or the sum of the size- weights of the resource is provided, the classic ratio estimator is used to calculate the CDF estimate, where that estimator is the product of the known value and the Horvitz-Thompson ratio estimator. For the CDF of a total when neither the size

of the resource nor the sum of the size-weights of the resource is provided, the Horvitz-Thompson estimator is used to calculate the CDF estimate. Variance estimates for the estimated CDF are calculated using either the local mean variance estimator or the simple random sampling (SRS) variance estimator. The choice of variance estimator is subject to user control. The local mean variance estimator requires the x-coordinate and the y-coordinate of each site. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. Confidence bounds are calculated using a Normal distribution multiplier. In addition the function uses the estimated CDF to calculate percentile estimates. Estimated confidence bounds for the percentile estimates are calculated. The user can supply the set of values for which percentiles estimates are desired. Optionally, the user can use the default set of percentiles. The function can accommodate a stratified sample. For a stratified sample, separate estimates and standard errors are calculated for each stratum, which are used to produce estimates and standard errors for all strata combined. Strata that contain a single value are removed. For a stratified sample, when either the size of the resource or the sum of the size-weights of the resource is provided for each stratum, those values are used as stratum weights for calculating the estimates and standard errors for all strata combined. For a stratified sample when neither the size of the resource nor the sum of the size-weights of the resource is provided for each stratum, estimated values are used as stratum weights for calculating the estimates and standard errors for all strata combined. The function can accommodate single-stage and two-stage samples for both stratified and unstratified sampling designs. Finite population and continuous population correction factors can be utilized in variance estimation. The function checks for compatibility of input values and removes missing values.

Usage

```
cdf.decon(z, wgt, sigma, var.sigma = NULL, x = NULL, y = NULL,
  stratum = NULL, cluster = NULL, wgt1 = NULL, x1 = NULL,
  y1 = NULL, popsize = NULL, popcorrect = FALSE, pcfsize = NULL,
  N.cluster = NULL, stage1size = NULL, support = NULL,
  sizeweight = FALSE, swgt = NULL, swgt1 = NULL, vartype = "Local",
  conf = 95, cdfval = NULL, pctval = c(5, 10, 25, 50, 75, 90, 95),
  check.ind = TRUE, warn.ind = NULL, warn.df = NULL,
  warn.vec = NULL)
```

Arguments

z	Vector of the response values for each site. for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
sigma	Measurement error variance.
var.sigma	Variance of the measurement error variance. The default is NULL.
x	Vector of x-coordinates for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y	Vector of y-coordinates for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.

`stratum` Vector of the stratum value for each site. The default is NULL.

`cluster` Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site. The default is NULL.

`wgt1` Vector of the final adjusted stage one weight for each site. The default is NULL.

`x1` Vector of the stage one x-coordinates for location for each site. The default is NULL.

`y1` Vector of the stage one y-coordinates for location for each site. The default is NULL.

`popsize` Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource and to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`popcorrect` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`support` Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.

sizeweight	Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample. The default is NULL.
swgt1	Vector of the stage one size-weight for each site. The default is NULL.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
conf	The confidence level. The default is 95.
cdfval	The set of values at which the CDF is estimated. If a set of values is not provided, then the sorted set of unique values of the response variable is used. The default is NULL.
pctval	The set of values at which percentiles are estimated. The default set is: 5, 10, 25, 50, 75, 90, 95.
check.ind	Logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	A data frame for storing warning messages. The default is NULL.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called by the cont.analysis function, then output is an object in list format composed of a list named Results, which contains estimates and confidence bounds, and a data frame named warn.df, which contains warning messages. The Results list is composed of two data frames: one data frame named CDF, which contains the CDF estimates, and a second data frame named Pct, which contains the percentile estimates. If the function was called directly, then output is the Results list.

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

cdf.nresp calculate the number of response values less than or equal to each of the set of values at which the CDF is estimated

simex perform deconvolution of the response values

dcdf.prop calculate the deconvoluted CDF for the proportion

dcdf.total calculate the deconvoluted CDF for the total

dcdf.size.prop calculate the size-weighted deconvoluted CDF for the proportion

dcdf.size.total calculate the size-weighted deconvoluted CDF for the total

dcdfvar.prop calculate variance of the deconvoluted CDF for the proportion

dcdfvar.total calculate variance of the deconvoluted CDF for the total

dcdfvar.size.prop calculate variance of the size-weighted deconvoluted CDF for the proportion

dcdfvar.size.total calculate variance of the size-weighted deconvoluted CDF for the total

isotonic perform isotonic regression

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
z <- rnorm(100, 10, 1)
wgt <- runif(100, 10, 100)
cdfval <- seq(min(z), max(z), length=20)
cdf.decon(z, wgt, sigma=0.25, var.sigma=0.1, vartype="SRS", cdfval=cdfval)

x <- runif(100)
y <- runif(100)
cdf.decon(z, wgt, sigma=0.25, var.sigma=0.1, x, y, cdfval=cdfval)
```

---

cdf.est *Cumulative Distribution Function Estimate for Survey Design*

Description

This function calculates an estimate of the cumulative distribution function (CDF) for the proportion (expressed as percent) and the total of a response variable, where the response variable may be defined for either a finite or an extensive resource. Optionally, for a finite resource, the size- weighted CDF can be calculated. In addition the standard error of the estimated CDF and confidence bounds are calculated. The user can supply the set of values at which the CDF is estimated. For the CDF of a proportion, the Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the CDF estimate. For the CDF of a total, the user can supply the known size of the resource or the known sum of the size-weights of the resource, as appropriate. For the CDF of a total when either the size of the resource or the sum of the size-weights of the resource is provided, the classic ratio estimator is used to calculate the CDF estimate, where that estimator is the product of the known value and the Horvitz-Thompson ratio estimator. For the CDF of a total when neither the size of the resource nor the sum of the size-weights of the resource is provided, the Horvitz-Thompson estimator is used to calculate the CDF estimate. Variance estimates for the estimated CDF are calculated using either the local mean variance estimator or the simple random sampling (SRS) variance estimator. The choice of variance estimator is subject to user control. The local mean variance estimator requires the x-coordinate and the y-coordinate of each site. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion

probabilities. Confidence bounds are calculated using a Normal distribution multiplier. In addition the function uses the estimated CDF to calculate percentile estimates. Estimated confidence bounds for the percentile estimates are calculated. The user can supply the set of values for which percentiles estimates are desired. Optionally, the user can use the default set of percentiles. The function can accommodate a stratified sample. For a stratified sample, separate estimates and standard errors are calculated for each stratum, which are used to produce estimates and standard errors for all strata combined. Strata that contain a single value are removed. For a stratified sample, when either the size of the resource or the sum of the size-weights of the resource is provided for each stratum, those values are used as stratum weights for calculating the estimates and standard errors for all strata combined. For a stratified sample when neither the size of the resource nor the sum of the size-weights of the resource is provided for each stratum, estimated values are used as stratum weights for calculating the estimates and standard errors for all strata combined. The function can accommodate single-stage and two-stage samples for both stratified and unstratified sampling designs. Finite population and continuous population correction factors can be utilized in variance estimation. The function checks for compatibility of input values and removes missing values.

Usage

```
cdf.est(z, wgt, x = NULL, y = NULL, stratum = NULL, cluster = NULL,
  wgt1 = NULL, x1 = NULL, y1 = NULL, popsize = NULL,
  popcorrect = FALSE, pcfsize = NULL, N.cluster = NULL,
  stage1size = NULL, support = NULL, sizeweight = FALSE,
  swgt = NULL, swgt1 = NULL, vartype = "Local", conf = 95,
  cdfval = NULL, pctval = c(5, 10, 25, 50, 75, 90, 95),
  check.ind = TRUE, warn.ind = NULL, warn.df = NULL,
  warn.vec = NULL)
```

Arguments

z	Vector of the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of the x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y	Vector y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
stratum	Vector of the stratum for each site. The default is NULL.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site. The default is NULL.
wgt1	Vector of the final adjusted stage one weight for each site. The default is NULL.
x1	Vector of the stage one x-coordinate for location for each site. The default is NULL.
y1	Vector of the stage one y-coordinate for location for each site. The default is NULL.

popsize — Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource and to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

popcorrect — Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

pcfsize — Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

N.cluster — The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

stage1size — Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

support — The support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.

sizeweight — Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

swgt — The size-weight for each site, which is the stage two size-weight for two-stage sample. The default is NULL.

swgt1 — Vector of the stage one size-weight for each site. The default is NULL.

vartype — The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

conf — Numeric value for the confidence level. The default is 95.

cdfval	The set of values at which the CDF is estimated. If a set of values is not provided, then the sorted set of unique values of the response variable is used. The default is NULL.
pctval	The set of values at which percentiles are estimated. The default set is: 5, 10, 25, 50, 75, 90, 95.
check.ind	Logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	A data frame for storing warning messages. The default is NULL.
warn.vec	A vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called by the cont.analysis function, then output is an object in list format composed of a list named Results, which contains estimates and confidence bounds, and a data frame named warn.df, which contains warning messages. The Results list is composed of two data frames: one data frame named CDF, which contains the CDF estimates, and a second data frame named Pct, which contains the percentile estimates. If the function was called directly, then output is the Results list.

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

cdf.nresp calculate the number of response values less than or equal to each of the set of values at which the CDF is estimated

cdf.prop calculate the CDF for the proportion

cdf.total calculate the CDF for the total

cdf.size.prop calculate the size-weighted CDF for the proportion

cdf.size.total calculate the size-weighted CDF for the total

cdfvar.prop calculate variance of the CDF for the proportion

cdfvar.total calculate variance of the CDF for the total

cdfvar.size.prop calculate variance of the size-weighted CDF for the proportion

cdfvar.size.total calculate variance of the size-weighted CDF for the total

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
z <- rnorm(100, 10, 1)
wgt <- runif(100, 10, 100)
cdfval <- seq(min(z), max(z), length=20)
cdf.est(z, wgt, vartype="SRS", cdfval=cdfval)

x <- runif(100)
y <- runif(100)
cdf.est(z, wgt, x, y, cdfval=cdfval)
```

---

cdf.nresp *Internal Function: Count Number of Responses Less Than a Set of Values*

Description

This function calculates the number of response values less than or equal to each of the set of values at which the cumulative distribution function (CDF) is estimated.

Usage

```
cdf.nresp(z, val)
```

Arguments

z	Vector of the response values.
val	Vector of the set of values at which the CDF is estimated.

Value

Output is the number of response values for each CDF estimation value.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdf.plot` *Plot a Cumulatiave Distribution Function*

Description

This function creates a CDF plot. Input data for the plots is provided by a data frame utilizing the same structure as the data frame named "CDF" that is included in the output object produced by function cont.analysis, but the data frame includes only the values for a single CDF. Confidence limits for the CDF also are plotted.

Usage

```
cdf.plot(cdfest, units.cdf = "Percent", type.cdf = "Continuous",
  logx = "", xlbl = NULL, ylbl = "Percent", ylbl.r = NULL,
  figlab = NULL, legloc = "BR", confcut = 5, conflev = 95,
  cex.main = 1.2, ...)
```

Arguments

`cdfest`	Data frame utilizing the same structure as the data frame named "CDF" that is included in the output object produced by function cont.analysis. The data frame must contain only a single cdf estimate.
`units.cdf`	Indicator for the type of units in which the CDF is plotted, where "Percent" means the plot is in terms of percent of the population, and "Units" means the plot is in terms of units of the population. The default is "Percent".
`type.cdf`	Character string consisting of the value "Continuous" or "Ordinal" that controls the type of CDF plot for each indicator. The default is "Continuous".
`logx`	Character string consisting of the value "" or "x" that controls whether the x axis uses the original scale ("") or the base 10 logarithmic scale ("x"). The default is "".
`xlbl`	Character string providing the x-axis label. If this argument equals NULL, then the indicator name is used as the label. The default is NULL.
`ylbl`	Character string providing the y-axis label. The default is "Percent".
`ylbl.r`	Character string providing the label for the right side y-axis, where NULL means a label is not created, and "Same" means the label is the same as the left side label (i.e., argument ylbl). The default is NULL.
`figlab`	Character string providing the plot title. The default is NULL.
`legloc`	Indicator for location of the plot legend, where "BR" means bottom right, "BL" means bottom left, "TR" means top right, and "TL" means top left. The default is "BR".
`confcut`	Numeric value that controls plotting confidence limits at the CDF extremes. Confidence limits for CDF values (percent scale) less than confcut or greater than 100 minus confcut are not plotted. A value of zero means confidence limits are plotted for the complete range of the CDF. The default is 5.

conflev	Numeric value of the confidence level used for confidence limits. The default is 95.
cex.main	Expansion factor for the plot title. The default is 1.2.
...	Additional arguments passed to the plot function.

Value

A plot of the CDF and its associated confidence limits.

Other Functions Required

interp.cdf interpolate CDF values at a set of percentiles

interp.axis create right side y-axis labels for a CDF plot

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>
Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(siteID=mysiteID, Active=rep(TRUE, 100))
mysubpop <- data.frame(siteID=mysiteID, All.Sites=rep("All Sites",100),
   Resource.Class=rep(c("Good","Poor"), c(55,45)))
mydesign <- data.frame(siteID=mysiteID, wgt=runif(100, 10, 100),
   xcoord=runif(100), ycoord=runif(100), stratum=rep(c("Stratum1",
   "Stratum2"), 50))
ContVar <- rnorm(100, 10, 1)
mydata.cont <- data.frame(siteID=mysiteID, ContVar=ContVar)
mypopsize <- list(All.Sites=c(Stratum1=3500, Stratum2=2000),
   Resource.Class=list(Good=c(Stratum1=2500, Stratum2=1500),
   Poor=c(Stratum1=1000, Stratum2=500)))
myanalysis <- cont.analysis(sites=mysites, subpop=mysubpop,
   design=mydesign, data.cont=mydata.cont, popsize=mypopsize)
keep <- myanalysis$CDF$Type == "Resource.Class" &
   myanalysis$CDF$Subpopulation == "Good"
par(mfrow=c(2,1))
cdf.plot(myanalysis$CDF[keep,], xlbl="ContVar",
   ylbl="Percent of Stream Length", ylbl.r="Stream Length (km)",
   figlab="Estimates for Resource Class: Good")
cdf.plot(myanalysis$CDF[keep,], xlbl="ContVar",
   ylbl="Percent of Stream Length", ylbl.r="Same)",
   figlab="Estimates for Resource Class: Good")
```

cdf.prop *Estimate of Cumulative Distribution Function for a Proportion*

Description

This function calculates an estimate of the cumulative distribution function (CDF) for the proportion of a finite or an extensive resource. The set of values at which the CDF is estimated is supplied to the function. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the estimate. The numerator of the ratio estimates the total of the resource equal to or less than a specified value. The denominator of the ratio estimates the size of the resource. For a finite resource size is the number of units in the resource. For an extensive resource size is the extent (measure) of the resource, i.e., length, area, or volume. The function can accomodate single-stage and two-stage samples.

Usage

```
cdf.prop(z, wgt, val, cluster.ind, cluster, wgt1)
```

Arguments

z	Vector of the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
val	Vector of the set of values at which the CDF is estimated.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.

Value

The CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdf.size.prop` *Size-weighted Cumulative Distribution Function Estimate*

Description

This function calculates an estimate of the size-weighted cumulative distribution function (CDF) for the proportion of a finite resource. The set of values at which the CDF is estimated is supplied to the function. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the estimate. The numerator of the ratio estimates the size-weighted total of the resource equal to or less than a specified value. The denominator of the ratio estimates the sum of the size-weights for the resource. The function can accomodate single-stage and two-stage samples.

Usage

```
cdf.size.prop(z, wgt, val, cluster.ind, cluster, wgt1, swgt, swgt1)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`val`	Vector of the set of values at which the CDF is estimated.
`cluster.ind`	Lloical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.
`swgt`	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
`swgt1`	Vector of the stage one size-weight for each site.

Value

The size-weighted CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdf.size.total` *Size-weighted Cumulative Distribution Funcition Estimate for Total*

Description

This function calculates an estimate of the size-weighted cumulative distribution function (CDF) for the total of a finite resource. The set of values at which the CDF is estimated is supplied to the function. If the known sum of the size-weights of the resource is provided, the classic ratio estimator is used to calculate the estimate. That estimator is the product of the known sum of the size-weights of the resource and the Horvitz- Thompson ratio estimator, where the latter is the ratio of two Horvitz- Thompson estimators. The numerator of the ratio estimates the size-weighted total of the resource equal to or less than a specified value. The denominator of the ratio estimates the sum of the size-weights of the resource. If the known sum of the size-weights of the resource is not provided, the Horvitz-Thompson estimator of the size-weighted total of the resource equal to or less than a specified value is used to calculate the estimate. The function can accomodate single-stage and two-stage samples.

Usage

```
cdf.size.total(z, wgt, val, cluster.ind, cluster, wgt1, popsize, swgt,
  swgt1)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`val`	Vector of the set of values at which the CDF is estimated.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.
`popsize`	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
`swgt`	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
`swgt1`	Vector of the stage one size-weight for each site.

Value

The size-weighted CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdf.test` *Test for Difference Between Two Estimated Cumulative Distribution Functions*

Description

This function tests for differences between cumulative distribution functions (CDFs) generated by probability surveys. The function returns a variety of test statistics along with their degrees of freedom and p values. The inferential procedures divide the CDFs into a discrete set of intervals (classes) and then utilize procedures that have been developed for analysis of categorical data from probability surveys. The function calculates the Wald, Rao-Scott first order corrected (mean eigenvalue corrected), and Rao-Scott second order corrected (Satterthwaite corrected) test statistics. Both standard versions of the three statistics, which are distributed as Chi-squared random variables, and alternate version of the statistics, which are distributed as F random variables, are available. The default test statistic is the F distribution version of the Wald statistic. The user supplies the set of upper bounds that define the intervals (classes) into which the CDFs are divided (binned). The minimum number of classes is three. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz- Thompson estimators, is used to calculate estimates of the class proportions for the CDFs. Variance estimates for the test statistics are calculated using either the local mean variance estimator or the simple random sampling (SRS) variance estimator. The choice of variance estimator is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accommodate a stratified sample. For a stratified sample, separate class proportion estimates and associated covariance estimates are calculated for each stratum, which are used to produce estimates for all strata combined. Strata that contain a single value are removed. For a stratified sample, when either the size of the resource or the sum of the size-weights of the resource is provided for each stratum, those values are used as stratum weights for calculating the estimates for all strata combined. For a stratified sample when neither the size of the resource nor the sum of the size-weights of the resource is provided for each stratum, estimated values are used as stratum weights for calculating the estimates for all strata combined. The function can accommodate single-stage and two-stage samples for both stratified and unstratified sampling designs. Finite population and continuous population correction factors can be utilized in variance estimation. The function checks for compatability of input values and removes missing values.

Usage

```
cdf.test(bounds, z_1, wgt_1, x_1 = NULL, y_1 = NULL, z_2, wgt_2,
  x_2 = NULL, y_2 = NULL, stratum_1 = NULL, stratum_2 = NULL,
  cluster_1 = NULL, cluster_2 = NULL, wgt1_1 = NULL, x1_1 = NULL,
  y1_1 = NULL, wgt1_2 = NULL, x1_2 = NULL, y1_2 = NULL,
```

```
popsize_1 = NULL, popsize_2 = NULL, popcorrect_1 = FALSE,
pcfsize_1 = NULL, N.cluster_1 = NULL, stage1size_1 = NULL,
support_1 = NULL, popcorrect_2 = FALSE, pcfsize_2 = NULL,
N.cluster_2 = NULL, stage1size_2 = NULL, support_2 = NULL,
sizeweight_1 = FALSE, swgt_1 = NULL, swgt1_1 = NULL,
sizeweight_2 = FALSE, swgt_2 = NULL, swgt1_2 = NULL,
vartype_1 = "Local", vartype_2 = "Local", check.ind = TRUE,
warn.ind = NULL, warn.df = NULL, warn.vec = NULL)
```

Arguments

bounds	Vector of upper bounds that define classes for the CDFs, which must contain at least two values.
z_1	Vector of response value for each sample one site.
wgt_1	Vector of final adjusted weight (inverse of the sample inclusion probability) for each sample one site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
x_1	Vector of x-coordinate for location for each sample one site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y_1	Vector of y-coordinate for location for each sample one site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
z_2	Vector of response value for each sample two site.
wgt_2	Vector of final adjusted weight (inverse of the sample inclusion probability) for each sample two site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
x_2	Vector of x-coordinate for location for each sample two site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y_2	Vector of y-coordinate for location for each sample two site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
stratum_1	Vector of the stratum for each sample one site. The default is NULL.
stratum_2	Vector of the stratum for each sample two site. The default is NULL.
cluster_1	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each sample one site. The default is NULL.
cluster_2	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each sample two site. The default is NULL.
wgt1_1	Vector of the final adjusted stage one weight for each sample one site. The default is NULL.
x1_1	Vector of the stage one x-coordinate for location for each sample one site. The default is NULL.
y1_1	Vector of the stage one y-coordinate for location for each sample one site. The default is NULL.

wgt1_2 Vector of the final adjusted stage one weight for each sample two site. The default is NULL.

x1_2 Vector of the stage one x-coordinate for location for each sample two site. The default is NULL.

y1_2 Vector of the stage one y-coordinate for location for each sample two site. The default is NULL.

popsize_1 The known size of the sample one resource - the total number of sampling units of a finite resource or the measure of an extensive resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample, this variable also is used to calculate strata weights. For a stratified sample, this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

popsize_2 The known size of the sample two resource - the total number of sampling units of a finite resource or the measure of an extensive resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample, this variable also is used to calculate strata weights. For a stratified sample, this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

popcorrect_1 Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation for sample one, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize_1 and support_1. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster_1, stage1size_1, and support_1.

pcfsize_1 Size of the sample one resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

N.cluster_1 The number of stage one sampling units in the sample one resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

stage1size_1 Size of the stage one sampling units of a two-stage sample for sample one, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

support_1 Vector of the support value for each sample one site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with

a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.

`popcorrect_2` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation for sample two, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize_2 and support_2. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster_2, stage1size_2, and support_2.

`pcfsize_2` Size of the sample two resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster_2` The number of stage one sampling units in the sample two resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size_2` Vector of the size of the stage one sampling units of a two-stage sample for sample two, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`support_2` Vector of the support value for each sample two site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.

`sizeweight_1` Logical value that indicates whether size-weights should be used in the analysis for sample one, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

`swgt_1` Vector of the size-weight for each sample one site, which is the stage two size-weight for a two-stage sample. The default is NULL.

`swgt1_1` Vector of the stage one size-weight for each sample one site. The default is NULL.

`sizeweight_2` Logical value that indicates whether size-weights should be used in the analysis for sample two, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

`swgt_2` Vector of the size-weight for each sample two site, which is the stage two size-weight for a two-stage sample. The default is NULL.

`swgt1_2` Vector of the stage one size-weight for each sample two site. The default is NULL.

`vartype_1` The choice of variance estimator for sample one, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

vartype_2	The choice of variance estimator for sample two, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
check.ind	Logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	A data frame for storing warning messages. The default is NULL.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

An object in data frame format containing the test statistic, degrees of freedom (two values labeled Degrees of Freedom_1 and Degrees of Freedom_2), and p value for the Wald, mean eigenvalue, and Satterthwaite test procedures, which includes both Chi-squared distribution and F distribution versions of the procedures. For the Chi-squared versions of the test procedures, Degrees of Freedom_1 contains the relevant value and Degrees of Freedom_2 is set to missing (NA). For the F-based versions of the test procedures Degrees of Freedom_1 contains the numerator degrees of freedom and Degrees of Freedom_2 contains the denominator degrees of freedom.

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

cdf.test.prop calculates an estimate of the population proportions in the set of classes

cdf.test.size.prop calculates a size-weighted estimate of the population proportions in the set of classes

cdfvar.test calculates estimates of the variance-covariance matrix of the population proportions in the set of classes

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
n <- 100
resp <- rnorm(n, 10, 1)
wgt <- runif(n, 10, 100)
sample1 <- list(z=resp, wgt=wgt)
sample2 <- list(z=resp+0.5, wgt=wgt)
bounds <- sort(c(sample1$z, sample2$z))[floor(seq((2*n)/3, (2*n),
   length=3))]
```

```
cdf.test(bounds=bounds, z_1=sample1$z, wgt_1=sample1$wgt, z_2=sample2$z,
   wgt_2=sample2$wgt, vartype_1="SRS", vartype_2="SRS")

xcoord <- runif(n)
ycoord <- runif(n)
sample1 <- list(z=resp, wgt=wgt, x=xcoord, y=ycoord)
sample2 <- list(z=1.05*resp, wgt=wgt, x=xcoord, y=ycoord)
cdf.test(bounds=bounds, z_1=sample1$z, wgt_1=sample1$wgt, x_1=sample1$x,
   y_1=sample1$y, z_2=sample2$z, wgt_2=sample2$wgt, x_2=sample2$x,
   y_2=sample2$y)
```

`cdf.test.prop` *Estimate of Population Proportion for Classes*

Description

This function calculates an estimate of the population proportion in a set of intervals (classes). The set of values defining the upper bound of each class is supplied to the function. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the estimate. The numerator of the ratio estimates the total of the resource within a class. The denominator of the ratio estimates the size of the resource. For a finite resource size is the number of units in the resource. For an extensive resource size is the extent (measure) of the resource, i.e., length, area, or volume. The function can accomodate single stage and two-stage samples.

Usage

```
cdf.test.prop(z, wgt, bounds, cluster.ind, cluster, wgt1)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
`bounds`	Upper bounds for calculating classes for the CDF.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.

Value

The class proportion estimates.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdf.test.size.prop` *Size-Weighted Estimate of Population Proportion for Classes*

Description

This function calculates a size-weighted estimate of the population proportions in a set of intervals (classes). The set of values defining the upper bound of each class is supplied to the function. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the estimate. The numerator of the ratio estimates the total of the resource within a class. The denominator of the ratio estimates the size of the resource. For a finite resource size is the number of units in the resource. For an extensive resource size is the extent (measure) of the resource, i.e., length, area, or volume. The function can accomodate single stage and two-stage samples.

Usage

```
cdf.test.size.prop(z, wgt, bounds, cluster.ind, cluster, wgt1, swgt, swgt1)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
`bounds`	Upper bounds for calculating classes for the CDF.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site. a two-stage sample.
`swgt`	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
`swgt1`	Vector of the stage one size-weight for each site.

Value

The class proportion estimates.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdf.total` *Estimate of Cumulative Distribution Function for a Total*

Description

This function calculates an estimate of the cumulative distribution function (CDF) for the total of a finite or an extensive resource. The set of values at which the CDF is estimated is supplied to the function. If the known extent of the resource is provided, the classic ratio estimator is used to calculate the estimate. That estimator is the product of the known extent of the resource and the Horvitz-Thompson ratio estimator, where the latter is the ratio of two Horvitz-Thompson estimators. The numerator of the ratio estimates the total of the resource equal to or less than a specified value. The denominator of the ratio estimates the extent of the resource. If the known extent of the resource is not provided, the Horvitz- Thompson estimator of the total of the resource equal to or less than a specified value is used to calculate the estimate. For a finite resource, size is the number of units in the resource. For an extensive resource, size is the measure of the resource, i.e., length, area, or volume. The function can accomodate single-stage and two-stage samples.

Usage

```
cdf.total(z, wgt, val, cluster.ind, cluster, wgt1, popsize)
```

Arguments

`z`	= Vector of the response value for each site.
`wgt`	= Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`val`	Vector of values at which the CDF is estimated.
`cluster.ind`	= Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	= Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	= Vector of the final adjusted stage one weight for each site.
`popsize`	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

Value

The CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdfvar.prop` *Variance Estimate for Cumulative Distribution Function for a Proportion*

Description

This function calculates variance estimates of the estimated cumulative distribution function (CDF) for the proportion of a finite or a continuous resource. The set of values at which the CDF is estimated is supplied to the function. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
cdfvar.prop(z, wgt, x, y, val, cdfest, stratum.ind, stratum.level,
  cluster.ind, cluster, wgt1, x1, y1, pcfactor.ind, pcfsize, N.cluster,
  stage1size, support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`x`	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
`y`	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
`val`	Vector of the set of values at which the CDF is estimated.
`cdfest`	The CDF estimate.
`stratum.ind`	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
`stratum.level`	Vector of the stratum level.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.

x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	The support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	A data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdfvar.size.prop` *Variance Estimate for Size-Weighted Cumulative Distribution Function for a Proportion*

Description

This function calculates variance estimates of the estimated size-weighted cumulative distribution function (CDF) for the proportion of a finite resource. The set of values at which the CDF is estimated is supplied to the function. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
cdfvar.size.prop(z, wgt, x, y, val, cdfest, stratum.ind, stratum.level,
  cluster.ind, cluster, wgt1, x1, y1, pcfactor.ind, pcfsize, N.cluster,
  stage1size, support, swgt, swgt1, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`x`	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
`y`	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
`val`	Vector of the set of values at which the CDF is estimated.
`cdfest`	The CDF estimate.
`stratum.ind`	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
`stratum.level`	Vector of the stratum level.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.

x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
swgt1	Vector of the stage one size-weight for each site.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

cdfvar.size.total *Variance Estimate for Size-weighted Cumulative Distribution Function for a Total*

Description

This function calculates variance estimates of the estimated size-weighted cumulative distribution function (CDF) for the total of a finite resource. The set of values at which the CDF is estimated is supplied to the function. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples.

Usage

```
cdfvar.size.total(z, wgt, x, y, val, cdfest, stratum.ind, stratum.level,
  cluster.ind, cluster, wgt1, x1, y1, popsize, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, swgt, swgt1, vartype, warn.ind, warn.df,
  warn.vec)
```

Arguments

z	Vector of the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
val	Vector of the set of values at which the CDF is estimated.
cdfest	The CDF estimate.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	Vector of the stratum level.

cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
popsize	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
swgt1	Vector of the stage one size-weight for each site.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.

warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

cdfvar.test *Variance-Covariance Matrix Estimate for Estimated Population Proportions*

Description

This function calculates estimates of the variance-covariance matrix of the population proportions in a set of intervals (classes). The set of values defining upper bounds for the classes is supplied to the function. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The simple random sampling variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
cdfvar.test(z, wgt, x, y, bounds, phat, stratum.ind, stratum.level,
  cluster.ind, cluster, wgt1, x1, y1, popsize, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, swgt.ind, swgt, swgt1, vartype, warn.ind,
  warn.df, warn.vec)
```

Arguments

z	Vector of the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
bounds	Vector of upper bounds for calculating classes for the CDF.
phat	The class proportions estimate.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	Vector of = the stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
popsize	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
swgt.ind	Logical value that indicates whether the sample includes size-weights, where TRUE = the sample includes size-weights and FALSE = the sample does not include size-weights.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
swgt1	Vector of the stage one size-weight for each site.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	dat A frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

Object in list format composed of a vector named nbin, which contains the number of response values in each class, a vector named varest, which contains variance estimates, a numeric value named df, which contain degrees of freedom of the variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.cov calculate the variance/covariance matrix using the local mean estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`cdfvar.total` *Variance Estimate for Cumulative Distribution Function for a Total*

Description

This function calculates variance estimates of the estimated cumulative distribution function (CDF) for the total of a finite or a continuous resource. The set of values at which the CDF is estimated is supplied to the function. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
cdfvar.total(z, wgt, x, y, val, cdfest, stratum.ind, stratum.level,
  cluster.ind, cluster, wgt1, x1, y1, popsize, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`x`	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
`y`	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
`val`	Vector of the set of values at which the CDF is estimated.
`cdfest`	The CDF estimate.
`stratum.ind`	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
`stratum.level`	The stratum level.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.
`x1`	Vector of the stage one x-coordinate for location for each site.
`y1`	Vector of the stage one y-coordinate for location for each site.

popsize	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	A data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

cell.wt *Last Revised: September 24, 2018*

Description

Total Inclusion Probablity for Matrix of Cells

Usage

```
cell.wt(cel, xc, yc, dx, dy, pts)
```

Arguments

cel	= the index value for a cell.
xc	= x-coordinates that define the cells.
yc	= y-coordinates that define the cells.
dx	= width of the cells along the x-axis.
dy	= width of the cells along the y-axis.
pts	= an 'sf' data frame containing x-coordinates and y-coordinates in the geometry list-column, and mdm values.

Details

Calculates the total inclusion probability for a cell. Used to evaluate spatial balance of a survey design realization.

Value

The total inclusion probability for a cell.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

Marc Weber <Weber.Marc@epa.gov>

cellWeight *Total Inclusion Probablity for a Grid Cell*

Description

Calculates the total inclusion probability for each grid cell from a GRTS survey design.

Usage

```
cellWeight(xc, yc, dx, dy, sfobject)
```

Arguments

xc	x-coordinates that define the cells.
yc	y-coordinates that define the cells.
dx	The x-axis grid cell dimension.
dy	The y-axis grid cell dimension.
sfobject	the sf object containing the survey frame.

Value

Vector containing the total inclusion probability for each cell.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

change.analysis *Estimation of Change between Two Time Periods in a Probability Survey*

Description

This function organizes input and output for estimation of change between two probability surveys.

Usage

```
change.analysis(sites, repeats = NULL, subpop = NULL, design,
  data.cat = NULL, data.cont = NULL, revisitwgt = FALSE,
  test = "mean", popsize_1 = NULL, popsize_2 = NULL,
  popcorrect_1 = FALSE, popcorrect_2 = FALSE, pcfsize_1 = NULL,
  pcfsize_2 = NULL, N.cluster_1 = NULL, N.cluster_2 = NULL,
  stage1size_1 = NULL, stage1size_2 = NULL, sizeweight_1 = FALSE,
  sizeweight_2 = FALSE, vartype_1 = "Local", vartype_2 = "Local",
  conf = 95)
```

Arguments

sites	Data frame consisting of three variables: the first variable is site IDs, and the other variables are logical vectors indicating which sites to use in the analysis. The first logical vector indicates the complete set of sites for the first survey. The second logical vector indicates the complete set of sites for the second survey.
repeats	Data frame that identifies site IDs for repeat visit sites from the two surveys. The first variable is site IDs for survey one. The second variable is site IDs for survey two. For each row of the data frame, the two site IDs must correspond to the same site. This argument should equal NULL when repeat visit sites are not present. The default is NULL.
subpop	Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is site IDs. Each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. The default is NULL.
design	Data frame consisting of design variables. If spsurvey.obj is not provided, then this argument is required. The default is NULL. Variables should be named as follows: **siteID** Vector of site IDs **wgt** Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample **xcoord** Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample **ycoord** Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample **stratum** Vector of the stratum codes for each site **cluster** Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site **wgt1** Vector of stage one weights in a two-stage design **xcoord1** Vector of the stage one x-coordinates for location in a two-stage design **ycoord1** Vector of the stage one y-coordinates for location in a two-stage design **support** Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors. **swgt** Vector of size-weights, which is the stage two size-weight for a two-stage design. **swgt1** Vector of stage one size-weights for a two-stage design.
data.cat	Data frame of categorical response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. The default is NULL.
data.cont	Data frame of continuous response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. The default is NULL.

`revisitwgt` Logical value that indicates whether each repeat visit site has the same survey design weight in the two surveys, where TRUE = the weight for each repeat visit site is the same and FALSE = the weight for each repeat visit site is not the same. When this argument is FALSE, all of the repeat visit sites are assigned equal weights when calculating the covariance component of the change estimate standard error. The default is FALSE.

`test` Character string or character vector providing the location measure(s) to use for change estimation for continuous variables. The choices are "mean", "median", or c("mean", "median"). The default is "mean".

`popsize_1` Known size of the resource for survey one, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource and to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size- weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. The argument must be in the form of a list containing an element for each population Type in the subpop data frame, where NULL is a valid choice for a population Type. The list must be named using the column names for the population Types in subpop. If a population Type doesn't contain subpopulations, then each element of the list is either a single value for an unstratified sample or a vector containing a value for each stratum for a stratified sample, where elements of the vector are named using the stratum codes. If a population Type contains subpopulations, then each element of the list is a list containing an element for each subpopulation, where the list is named using the subpopulation names. The element for each subpopulation will be either a single value for an unstratified sample or a named vector of values for a stratified sample. The default is NULL.

Example popsize for a stratified sample:
popsize = list("Pop 1"=c("Stratum 1"=750,
"Stratum 2"=500,
"Stratum 3"=250),
"Pop 2"=list("SubPop 1"=c("Stratum 1"=350,
"Stratum 2"=250,
"Stratum 3"=150),
"SubPop 2"=c("Stratum 1"=250,
"Stratum 2"=150,
"Stratum 3"=100),
"SubPop 3"=c("Stratum 1"=150,
"Stratum 2"=150,
"Stratum 3"=75)),
"Pop 3"=NULL)

Example popsize for an unstratified sample:
popsize = list("Pop 1"=1500,
"Pop 2"=list("SubPop 1"=750,
"SubPop 2"=500,
"SubPop 3"=375),

"Pop 3"=NULL)

popsize_2 Known size of the resource for survey two. The default is NULL.

popcorrect_1 Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation for survey one, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize_1 and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster_1 and stage1size_1 and for the support variable of the design argument.

popcorrect_2 Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation for survey two, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize_2 and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster_2 and stage1size_2 and for the support variable of the design argument.

pcfsize_1 Size of the resource for survey one, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

pcfsize_2 Size of the resource for survey two. The default is NULL.

N.cluster_1 Number of stage one sampling units in the resource for survey one, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

N.cluster_2 Number of stage one sampling units in the resource for survey two. The default is NULL.

stage1size_1 Size of the stage one sampling units of a two-stage sample for survey one, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

stage1size_2 Size of the stage one sampling units of a two-stage sample for survey two. The default is NULL.

sizeweight_1 Logical value that indicates whether size-weights should be used in the analysis of survey one, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

sizeweight_2 Logical value that indicates whether size-weights should be used in the analysis of survey two. The default is FALSE.

vartype_1	The choice of variance estimator for survey one, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
vartype_2	The choice of variance estimator for survey two. The default is "Local".
conf	Numeric value for the confidence level. The default is 95.

Value

List of change estimates composed of three items: (1) catsum contains change estimates for categorical variables, (2) contsum_mean contains estimates for continuous variables using the mean, and (3) contsum_median contains estimates for continuous variables using the median. The items in the list will contain NULL for estimates that were not calculated. Each data frame includes estimates for all combinations of population Types, subpopulations within Types, response variables, and categories within each response variable (for categorical variables and continuous variables using the median). Change estimates are provided plus standard error estimates and confidence interval estimates.

Other Functions Required

dframe.check check site IDs, the sites data frame, the subpop data frame, and the data.cat data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

vecprint takes an input vector and outputs a character string with line breaks inserted

uniqueID creates unique site IDs by appending a unique number to each occurrence of a site ID

input.check check input values for errors, consistency, and compatibility with analytical functions

change.est estimate change between two surveys

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
# Categorical variable example for three resource classes
mysiteID <- paste("Site", 1:200, sep="")
mysites <- data.frame(
  siteID=mysiteID,
  Survey1=rep(c(TRUE, FALSE), c(100,100)),
  Survey2=rep(c(FALSE, TRUE), c(100,100)))
myrepeats <- data.frame(
  siteID_1=paste("Site", 1:40, sep=""),
  siteID_2=paste("Site", 101:140, sep=""))
mysubpop <- data.frame(
  siteID=mysiteID,
  All_Sites=rep("All Sites", 200),
  Region=rep(c("North","South"), 100))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(200, 10, 100),
```

```
  xcoord=runif(200),
  ycoord=runif(200),
  stratum=rep(rep(c("Stratum1", "Stratum2"), c(2,2)), 50))
mydata.cat <- data.frame(
  siteID=mysiteID,
  Resource_Class=sample(c("Good","Fair","Poor"), 200, replace=TRUE))
change.analysis(sites=mysites, repeats=myrepeats, subpop=mysubpop,
  design=mydesign, data.cat=mydata.cat, data.cont=NULL)
```

`change.est` *Estimate Change between Two Surveys*

Description

This function estimates change between two probability surveys. The function can accommodate both categorical and continuous response variables. For a categorical response variable, change is estimated by the difference in category estimates for the two surveys, where a category estimate is the estimated proportion of values in a category. Note that a separate change estimate is calculated for each category of a categorical response variable. For a continuous response variable, change can be estimated for the mean, the median, or for both the mean and median. For a continuous response variable using the mean, change is estimated by the difference in estimated mean values for the two surveys. For change estimates using the median, the first step is to calculate an estimate of the median for the first survey. The estimated median from the first survey is then used to define two categories: (1) values that are less than or equal to the estimated median and (2) values that are greater than the estimated median. Once the categories are defined, change analysis for the median is identical to change analysis for a categorical variable, i.e., change is estimated by the difference in category estimates for the two surveys. In addition to change estimates, the standard error of the change estimates and confidence bounds are calculated. Variance estimates are calculated using either the local mean variance estimator or the simple random sampling (SRS) variance estimator. The choice of variance estimator is subject to user control. The local mean variance estimator requires the x-coordinate and y-coordinate of each site. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. Confidence bounds are calculated using a Normal distribution multiplier. The function can accommodate a stratified sample. For a stratified sample, separate estimates and standard errors are calculated for each stratum, which are used to produce estimates and standard errors for all strata combined. Strata that contain a single value are removed. For a stratified sample, when either the size of the resource or the sum of the size-weights of the resource is provided for each stratum, those values are used as stratum weights for calculating the estimates and standard errors for all strata combined. For a stratified sample when neither the size of the resource nor the sum of the size-weights of the resource is provided for each stratum, estimated values are used as stratum weights for calculating the estimates and standard errors for all strata combined. The function can accommodate single-stage and two-stage samples for both stratified and unstratified sampling designs. It is assumed that both surveys employ the same type of survey design. Finite population and continuous population correction factors can be utilized in variance estimation. The function checks for compatibility of input values and removes missing values.

Usage

```
change.est(resp.ind, z_1, wgt_1, x_1 = NULL, y_1 = NULL, repeat_1, z_2,
  wgt_2, x_2 = NULL, y_2 = NULL, repeat_2, revisitwgt = FALSE,
  test = "mean", stratum_1 = NULL, stratum_2 = NULL,
  cluster_1 = NULL, cluster_2 = NULL, wgt1_1 = NULL, x1_1 = NULL,
  y1_1 = NULL, wgt1_2 = NULL, x1_2 = NULL, y1_2 = NULL,
  popsize_1 = NULL, popsize_2 = NULL, popcorrect_1 = FALSE,
  pcfsize_1 = NULL, N.cluster_1 = NULL, stage1size_1 = NULL,
  support_1 = NULL, popcorrect_2 = FALSE, pcfsize_2 = NULL,
  N.cluster_2 = NULL, stage1size_2 = NULL, support_2 = NULL,
  sizeweight_1 = FALSE, swgt_1 = NULL, swgt1_1 = NULL,
  sizeweight_2 = FALSE, swgt_2 = NULL, swgt1_2 = NULL,
  vartype_1 = "Local", vartype_2 = "Local", conf = 95,
  check.ind = TRUE, warn.ind = NULL, warn.df = NULL,
  warn.vec = NULL)
```

Arguments

resp.ind	A character value that indicates the type of response variable, where "cat" indicates a categorical variable and "cont" indicates a continuous variable.
z_1	Vector of response value for each survey one site.
wgt_1	Vector of final adjusted weight (inverse of the sample inclusion probability) for each survey one site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
x_1	Vector of x-coordinate for location for each survey one site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y_1	Vector of y-coordinate for location for each survey one site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
repeat_1	Logical variable that identifies repeat visit sites for survey one.
z_2	Vector of response value for each survey two site.
wgt_2	Vector of final adjusted weight for each survey two site.
x_2	Vector of x-coordinate for location for each survey two site. The default is NULL.
y_2	Vector of y-coordinate for location for each survey two site. The default is NULL.
repeat_2	Logical variable that identifies repeat visit sites for survey two.
revisitwgt	Logical value that indicates whether each repeat visit site has the same survey design weight in the two surveys, where TRUE = the weight for each repeat visit site is the same and FALSE = the weight for each repeat visit site is not the same. When this argument is FALSE, all of the repeat visit sites are assigned equal weights when calculating the covariance component of the change estimate standard error. The default is FALSE.

test	A character string or character vector providing the location measure(s) to use for change estimation for continuous variables. The choices are "mean", "median", or c("mean", "median"). The default is "mean".
stratum_1	Vector of the stratum for each survey one site. The default is NULL.
stratum_2	Vector of the stratum for each survey two site. The default is NULL.
cluster_1	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each survey one site. The default is NULL.
cluster_2	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each survey two site. The default is NULL.
wgt1_1	Vector of the final adjusted stage one weight for each survey one site. The default is NULL.
x1_1	Vector of the stage one x-coordinate for location for each survey one site. The default is NULL.
y1_1	Vector of the stage one y-coordinate for location for each survey one site. The default is NULL.
wgt1_2	Vector of the final adjusted stage one weight for each survey two site. The default is NULL.
x1_2	Vector of the stage one x-coordinate for location for each survey two site. The default is NULL.
y1_2	Vector of the stage one y-coordinate for location for each survey two site. The default is NULL.
popsize_1	The known size of the survey one resource - the total number of sampling units of a finite resource or the measure of an extensive resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample, this variable also is used to calculate strata weights. For a stratified sample, this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.
popsize_2	The known size of the survey two resource. The default is NULL.
popcorrect_1	= a logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation for survey one, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize_1 and support_1. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster_1, stage1size_1, and support_1.
pcfsize_1	Size of the survey one resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.
N.cluster_1	The number of stage one sampling units in the survey one resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector

containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size_1` Size of the stage one sampling units of a two-stage sample for survey one, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`support_1` Vector of the support value for each survey one site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.

`popcorrect_2` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation for survey two, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize_2 and support_2. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster_2, stage1size_2, and support_2.

`pcfsize_2` Size of the survey two resource. The default is NULL.

`N.cluster_2` The number of stage one sampling units in the survey two resource. The default is NULL.

`stage1size_2` Size of the stage one sampling units of a two-stage survey for survey two. The default is NULL.

`support_2` Vector of the support value for each survey two site. The default is NULL.

`sizeweight_1` Logical value that indicates whether size-weights should be used in the analysis for survey one, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

`swgt_1` Vector of size-weight for each survey one site, which is the stage two size-weight for a two-stage sample. The default is NULL.

`swgt1_1` Vector of the stage one size-weight for each survey one site. The default is NULL.

`sizeweight_2` Logical value that indicates whether size-weights should be used in the analysis for survey two. The default is FALSE.

`swgt_2` Vector of the size-weight for each survey two site. The default is NULL.

`swgt1_2` Vector of the stage one size-weight for each survey two site. The default is NULL.

`vartype_1` The choice of variance estimator for survey one, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

`vartype_2` The choice of variance estimator for survey two. The default is "Local".

`conf` Numeric value for the confidence level. The default is 95.

check.ind	Logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	Data frame for storing warning messages. The default is NULL.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called by the change.analysis function, then output is an object in list format composed of the Results data frame, which contains estimates and confidence bounds, and the warn.df data frame, which contains warning messages. If the function was called directly, then output is the Results data frame.

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

category.est estimate proportion (expressed as percent) and size of a resource in each of a set of categories

changevar.prop calculate covariance or correlation estimates of the estimated change in class proportion estimates between two probability surveys

cdf.est estimate the cumulative distribution function (CDF) for the proportion (expressed as percent) and the total of a response variable

total.est estimate the population total, mean, variance, and standard deviation of a response variable

changevar.mean calculate the covariance or correlation estimate of the estimated change in means between two probability surveys

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
z_1 <- sample(c("Good","Fair","Poor"), 100, replace=TRUE)
z_2 <- sample(c("Good","Fair","Poor"), 100, replace=TRUE)
wgt_1 <- runif(100, 10, 100)
wgt_2 <- runif(100, 10, 100)
repeat_1 <- rep(c(TRUE,FALSE), c(20,80))
repeat_2 <- rep(c(TRUE,FALSE), c(20,80))
change.est(resp.ind="cat", z_1=z_1, wgt_1=wgt_1, repeat_1=repeat_1,
```

```
  z_2=z_2, wgt_2=wgt_2, repeat_2=repeat_2, vartype_1="SRS", vartype_2="SRS")

z_1 <- rnorm(100, 10,10)
z_2 <- rnorm(100, 12, 10)
change.est(resp.ind="cont", z_1=z_1, wgt_1=wgt_1, repeat_1=repeat_1,
  z_2=z_2, wgt_2=wgt_2, repeat_2=repeat_2, vartype_1="SRS", vartype_2="SRS")
```

`changevar.mean` *Covariance or Correlation Matrix Estimate of Change in Means between Two Surveys*

Description

This function calculates estimates of the variance-covariance matrix of the population proportions in a set of intervals (classes). The set of values defining upper bounds for the classes is supplied to the function. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The simple random sampling variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
changevar.mean(z1, z2, wgt, x, y, revisitwgt, mean1, mean2, stratum.ind,
  stratum.level, cluster.ind, cluster, wgt1, x1, y1, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

`z1`	Vector of the response value for each site for survey one.
`z2`	Vector of the response value for each site for survey two.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
`x`	Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
`y`	Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
`revisitwgt`	Logical value that indicates whether each repeat visit site has the same survey design weight in the two surveys, where TRUE = the weight for each repeat visit site is the same and FALSE = the weight for each repeat visit site is not the same. When this argument is FALSE, all of the repeat visit sites are assigned equal weights when calculating the covariance component of the change estimate standard error.
`mean1`	The estimated mean for survey one.

mean2	The estimated mean for survey two.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named rslt, which contains the covariance or correlation estimate, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

`localmean.weight` calculate the weighting matrix for the local mean variance estimator

`localmean.cov` calculate the variance/covariance matrix using the local mean estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

changevar.prop — *Covariance or Correlation Matrix Estimate of Change in Class Proportions between Two Surveys*

Description

This function uses the repeat visit sites for two probability surveys to calculate either covariance or correlation estimates of estimated change in the proportion in each of a set of categories. Covariance estimates are calculated when the resivit sites have the same survey design weight in both surveys. Correlation estimates are calculated when the revisit sites do not have the same weight in both surveys, in which case the sites are assigned equal weights. The revisitwgt argument controls whether covariance or correlation estimates are calculated. Either the simple random sampling (SRS) variance/covariance estimator or the local mean variance/covariance estimator is calculated, which is subject to user control. The simple random sampling variance/covariance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
changevar.prop(catvar.levels, catvar1, catvar2, wgt, x, y, revisitwgt,
  prop1, prop2, stratum.ind, stratum.level, cluster.ind, cluster, wgt1, x1,
  y1, pcfactor.ind, pcfsize, N.cluster, stage1size, support, vartype,
  warn.ind, warn.df, warn.vec)
```

Arguments

`catvar.levels`	Vector of the set of categorical response values.
`catvar1`	Vector of the response value for each site for survey one.
`catvar2`	Vector of the response value for each site for survey two.

wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
revisitwgt	Logical value that indicates whether each repeat visit site has the same survey design weight in the two surveys, where TRUE = the weight for each repeat visit site is the same and FALSE = the weight for each repeat visit site is not the same. When this argument is FALSE, all of the repeat visit sites are assigned equal weights when calculating the covariance component of the change estimate standard error.
prop1	The set of category proportion estimates for survey one.
prop2	The set of category proportion estimates for survey two.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify

	both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named rslt, which contains the covariance or correlation estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.cov calculate the variance/covariance matrix using the local mean estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

changevar.size *Covariance or Correlation Matrix Estimate of Change in Class Resource Sizes between Two Surveys*

Description

This function uses the repeat visit sites for two probability surveys to calculate either covariance or correlation estimates of estimated change in resource size in each of a set of categories. Covariance estimates are calculated when the resivit sites have the same survey design weight in both surveys. Correlation estimates are calculated when the revisit sites do not have the same weight in both surveys, in which case the sites are assigned equal weights. The revisitwgt argument controls whether covariance or correlation estimates are calculated. Either the simple random sampling (SRS) variance/covariance estimator or the local mean variance/covariance estimator is calculated, which is subject to user control. The simple random sampling variance/covariance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
changevar.size(catvar.levels, catvar1, catvar2, wgt, x, y, revisitwgt,
  size1, size2, stratum.ind, stratum.level, cluster.ind, cluster, wgt1, x1,
  y1, pcfactor.ind, pcfsize, N.cluster, stage1size, support, vartype,
  warn.ind, warn.df, warn.vec)
```

Arguments

catvar.levels	Vector of the set of categorical response values.
catvar1	Vector of the response value for each site for survey one.
catvar2	Vector of the response value for each site for survey two.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
revisitwgt	Logical value that indicates whether each repeat visit site has the same survey design weight in the two surveys, where TRUE = the weight for each repeat visit site is the same and FALSE = the weight for each repeat visit site is not the same. When this argument is FALSE, all of the repeat visit sites are assigned equal weights when calculating the covariance component of the change estimate standard error.
size1	The set of category size estimates for survey one.
size2	The set of category size estimates for survey two.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named rslt, which contains the covariance or correlation estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.cov calculate the variance/covariance matrix using the local mean estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

constructAddr — *Construct the Hierarchical Addresses for a Generalized Random-Tesselation Stratified (GRTS) Survey Design*

Description

This function constructS the hierarchical addresses for a GRTS survey design.

Usage

```
constructAddr(xc, yc, dx, dy, nlev)
```

Arguments

xc	Vector of x-coordinates for the grid cells.
yc	Vector of y-coordinates for the grid cells.
dx	The x-axis grid cell dimension.
dy	The y-axis grid cell dimension.
nlev	Number of hierarchical levels for the grid.

Value

Vector of hierarchical addresses.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

cont.analysis — *Continuous Variable Data Analysis for Probability Survey Data*

Description

This function organizes input and output for analysis of continuous data generated by a probability survey. Input can be either an object of class spsurvey.analysis (see the documentation for function spsurvey.analysis) or through use of the other arguments to this function.

Usage

```
cont.analysis(sites = NULL, subpop = NULL, design = NULL,
  data.cont = NULL, sigma = NULL, var.sigma = NULL, popsize = NULL,
  popcorrect = FALSE, pcfsize = NULL, N.cluster = NULL,
  stage1size = NULL, sizeweight = FALSE, total = FALSE,
  vartype = "Local", conf = 95, pctval = c(5, 10, 25, 50, 75, 90,
  95), spsurvey.obj = NULL)
```

Arguments

`sites` Data frame consisting of two variables: the first variable is site IDs, and the second variable is a logical vector indicating which sites to use in the analysis. If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`subpop` Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is site IDs. Each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`design` Data frame consisting of design variables. If spsurvey.obj is not provided, then this argument is required. The default is NULL. Variables should be named as follows:

siteID Vector of site IDs

wgt Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample

xcoord Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample

ycoord Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample

stratum Vector of the stratum codes for each site

cluster Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site

wgt1 Vector of stage one weights in a two-stage design

xcoord1 Vector of the stage one x-coordinates for location in a two-stage design

ycoord1 Vector of the stage one y-coordinates for location in a two-stage design

support Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors.

swgt Vector of size-weights, which is the stage two size-weight for a two-stage design.

swgt1 Vector of stage one size-weights for a two-stage design.

`data.cont` Data frame of continuous response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. The default is NULL.

`sigma` Measurement error variance. This variable must be a vector containing a value for each response variable and must have the names attribute set to identify the response variable names. Missing data (NA) is allowed. The default is NULL.

`var.sigma` Variance of the measurement error variance. This variable must be a vector containing a value for each response variable and must have the names attribute

set to identify the response variable names. Missing data (NA) is allowed. The default is NULL.

popsize

Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource and to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. The argument must be in the form of a list containing an element for each population Type in the subpop data frame, where NULL is a valid choice for a population Type. The list must be named using the column names for the population Types in subpop. If a population Type doesn't contain subpopulations, then each element of the list is either a single value for an unstratified sample or a vector containing a value for each stratum for a stratified sample, where elements of the vector are named using the stratum codes. If a population Type contains subpopulations, then each element of the list is a list containing an element for each subpopulation, where the list is named using the subpopulation names. The element for each subpopulation will be either a single value for an unstratified sample or a named vector of values for a stratified sample. The default is NULL.

Example popsize for a stratified sample:
popsize = list("Pop 1"=c("Stratum 1"=750,
"Stratum 2"=500,
"Stratum 3"=250),
"Pop 2"=list("SubPop 1"=c("Stratum 1"=350,
"Stratum 2"=250,
"Stratum 3"=150),
"SubPop 2"=c("Stratum 1"=250,
"Stratum 2"=150,
"Stratum 3"=100),
"SubPop 3"=c("Stratum 1"=150,
"Stratum 2"=150,
"Stratum 3"=75)),
"Pop 3"=NULL)

Example popsize for an unstratified sample:
popsize = list("Pop 1"=1500,
"Pop 2"=list("SubPop 1"=750,
"SubPop 2"=500,
"SubPop 3"=375),
"Pop 3"=NULL)

popcorrect

Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize and for the support variable of the design

argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster and stage1size, and for the support variable of the design argument.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`sizeweight` Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

`total` Logical value that indicates whether the population total estimate should be included in the output Pct data frame, where TRUE = include the total estimate and FALSE = do not include the estimate. The default is FALSE.

`vartype` The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

`conf` Numeric value for the confidence level. The default is 95.

`pctval` Vector of the set of values at which percentiles are estimated. The default set is: 5, 10, 25, 50, 75, 90, 95.

`spsurvey.obj` List of class spsurvey.analysis that was produced by the function spsurvey.analysis. Depending on input to that function, some elements of the list may be NULL. The default is NULL.

Value

A list containing either two or four data frames of population estimates for all combinations of population Types, subpopulations within Types, and response variables. The data frames containing deconvoluted CDF estimates and deconvoluted percentile estimates are only included in the output list when an input value for measurement error variance is provided to the function. CDF estimates are calculated for both proportion and size of the population. Standard error estimates and confidence interval estimates also are calculated.

The four data frames are:

`CDF` data frame containing the CDF estimates

`Pct` data frame containing the percentile estimates plus population mean, standard deviation, and variance estimates

`CDF.D` data frame containing the deconvoluted CDF estimates

`Pct.D` data frame containing the deconvoluted percentile estimates

If an input value for measurement error variance is not provided to the function, then CDF.D and Pct.D are assigned the value NULL.

Other Functions Required

`dframe.check` check site IDs, the sites data frame, the subpop data frame, and the data.cat data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

`vecprint` takes an input vector and outputs a character string with line breaks inserted

`uniqueID` creates unique site IDs by appending a unique number to each occurrence of a site ID

`input.check` check input values for errors, consistency, and compatibility with analytical functions

`cdf.est` estimate the cumulative distribution function (CDF) for the proportion (expressed as percent) and the total of a response variable

`cdf.decon` estimate the deconvoluted CDF for the proportion and the total of a response variable

`total.est` estimate the population total, mean, variance, and standard deviation of a response variable

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>
Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
# Continuous variable example:
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(TRUE, 100))
mysubpop <- data.frame(
  siteID=mysiteID,
  All.Sites=rep("All Sites",100),
  Resource.Class=rep(c("Good","Poor"), c(55,45)))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1", "Stratum2"), 50))
ContVar <- rnorm(100, 10, 1)
mydata.cont <- data.frame(
  siteID=mysiteID,
  ContVar=ContVar)
```

```
mypopsize <- list(
  All.Sites=c(Stratum1=3500, Stratum2=2000),
  Resource.Class=list(Good=c(Stratum1=2500, Stratum2=1500),
                      Poor=c(Stratum1=1000, Stratum2=500)))
cont.analysis(sites=mysites, subpop=mysubpop, design=mydesign,
  data.cont=mydata.cont, popsize=mypopsize)

# Include deconvolution estimates:
mydata.cont <- data.frame(
  siteID=mysiteID,
  ContVar=ContVar,
  ContVar1=ContVar + rnorm(100, 0, sqrt(0.25)),
  ContVar2=ContVar + rnorm(100, 0, sqrt(0.50)))
mysigma <- c(ContVar=NA, ContVar1=0.25, ContVar2=0.50)
cont.analysis(sites=mysites, subpop=mysubpop[,1:2], design=mydesign,
  data.cont=mydata.cont, sigma=mysigma, popsize=mypopsize[1])
```

`cont.cdfplot` *Plot Multiple Cumulative Distribution Functions*

Description

This function creates CDF plots. Input data for the plots is provided by a data frame utilizing the same structure as the data frame named "CDF" that is included in the output object produced by function cont.analysis. Plots are produced for every combination of Type of population, subpopulation within Type, and indicator. Output from the function is placed in a PDF file.

Usage

```
cont.cdfplot(pdffile = "cdf2x2.pdf", cdfest, units.cdf = "Percent",
  ind.type = rep("Continuous", nind), logx = rep("", nind),
  xlbl = NULL, ylbl = "Percent", ylbl.r = NULL, legloc = "BR",
  cdf.page = 4, width = 10, height = 8, confcut = 5,
  cex.main = 1.2, ...)
```

Arguments

`pdffile`	Name of the PDF file. The default is "cdf2x2.pdf".
`cdfest`	Data frame utilizing the same structure as the data frame named "CDF" that is included in the output object produced by function cont.analysis.
`units.cdf`	Indicator for the type of units in which the CDF is plotted, where "Percent" means the plot is in terms of percent of the population, and "Units" means the plot is in terms of units of the population. The default is "Percent".
`ind.type`	Character vector consisting of the values "Continuous" or "Ordinal" that controls the type of CDF plot for each indicator. The default is "Continuous" for every indicator.

logx	Character vector consisting of the values "" or "x" that controls whether the x axis uses the original scale ("") or the base 10 logarithmic scale ("x") for each indicator. The default is "" for every indicator.
xlbl	Character vector consisting of the x-axis label for each indicator. If this argument equals NULL, then indicator names are used as the labels. The default is NULL.
ylbl	Character string providing the y-axis label. The default is "Percent".
ylbl.r	Character string providing the label for the right side y-axis, where NULL means a label is not created, and "Same" means the label is the same as the left side label (i.e., argument ylbl). The default is NULL.
legloc	Indicator for location of the plot legend, where "BR" means bottom right, "BL" means bottom left, "TR" means top right, and "TL" means top left. The default is "BR".
cdf.page	Number of CDF plots on each page, which must be chosen from the values: 1, 2, 4, or 6. The default is 4.
width	Width of the graphic region in inches. The default is 10.
height	Height of the graphic region in inches. The default is 8.
confcut	Numeric value that controls plotting confidence limits at the CDF extremes. Confidence limits for CDF values (percent scale) less than confcut or greater than 100 minus confcut are not plotted. A value of zero means confidence limits are plotted for the complete range of the CDF. The default is 5.
cex.main	Expansion factor for the plot title. The default is 1.2.
...	Additional arguments passed to the cdf.plot function.

Value

A PDF file containing the CDF plots.

Other Functions Required

cdf.plot plot the CDF and associated confidence limits

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>
Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
## Not run:
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(siteID=mysiteID, Active=rep(TRUE, 100))
mysubpop <- data.frame(
  siteID=mysiteID,
  All.Sites=rep("All Sites",100),
  Resource.Class=rep(c("Good","Poor"), c(55,45)))
mydesign <- data.frame(
```

```
  siteID=mysiteID, wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1", "Stratum2"), 50))
ContVar <- rnorm(100, 10, 1)
mydata.cont <- data.frame(
  siteID=mysiteID, ContVar=ContVar)
mypopsize <- list(
  All.Sites=c(Stratum1=3500, Stratum2=2000),
  Resource.Class=list(Good=c(Stratum1=2500, Stratum2=1500),
                      Poor=c(Stratum1=1000, Stratum2=500)))
myanalysis <- cont.analysis(sites=mysites, subpop=mysubpop, design=mydesign,
     data.cont=mydata.cont, popsize=mypopsize)
cont.cdfplot("myanalysis.pdf", myanalysis$CDF, ylbl.r="Stream Length (km)")

## End(Not run)
```

`cont.cdftest` *Cumulative Distribution Function Inference for a Probability Survey*

Description

This function organizes input and output for conducting inference regarding cumulative distribution functions (CDFs) generated by a probability survey. Input can be either an object of class spsurvey.analysis (see the documentation for function spsurvey.analysis) or through use of the other arguments to this function. For every response variable and every population Type, differences between CDFs are tested for every pair of subpopulations within a Type. The inferential procedures divide the CDFs into a discrete set of intervals (classes) and then utilize procedures that have been developed for analysis of categorical data from probability surveys. Choices for inference are the Wald, Rao-Scott first order corrected (mean eigenvalue corrected), and Rao-Scott second order corrected (Satterthwaite corrected) test statistics. Both standard versions of the three statistics, which are distributed as Chi-squared random variables, and alternate version of the statistics, which are distributed as F random variables, are available. The default test statistic is the F distribution version of the Wald statistic.

Usage

```
cont.cdftest(sites = NULL, subpop = NULL, design = NULL,
  data.cont = NULL, popsize = NULL, popcorrect = FALSE,
  pcfsize = NULL, N.cluster = NULL, stage1size = NULL,
  sizeweight = FALSE, vartype = "Local", testname = "Wald_F",
  nclass = 3, spsurvey.obj = NULL)
```

Arguments

`sites`	Data frame consisting of two variables: the first variable is site IDs, and the second variable is a logical vector indicating which sites to use in the analysis.

If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`subpop` Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is site IDs. Each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`design` Data frame consisting of design variables. If spsurvey.obj is not provided, then this argument is required. The default is NULL. Variables should be named as follows:

siteID Vector of site IDs

wgt Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample

xcoord Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample

ycoord Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample

stratum Vector of the stratum codes for each site

cluster Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site

wgt1 Vector of stage one weights in a two-stage design

xcoord1 Vector of the stage one x-coordinates for location in a two-stage design

ycoord1 Vector of the stage one y-coordinates for location in a two-stage design

support Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors.

swgt Vector of size-weights, which is the stage two size-weight for a two-stage design.

swgt1 Vector of stage one size-weights for a two-stage design.

`data.cont` Data frame of continuous response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. If spsurvey.obj is not provided, then this argument is required. The default is NULL.

`popsize` Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. The argument must be in the form of a list containing an element for each population Type in the subpop data frame, where NULL is a valid choice for a population Type. The list must be named using the column names for the population Types in subpop. If a population Type doesn't contain

subpopulations, then each element of the list is either a single value for an unstratified sample or a vector containing a value for each stratum for a stratified sample, where elements of the vector are named using the stratum codes. If a population Type contains subpopulations, then each element of the list is a list containing an element for each subpopulation, where the list is named using the subpopulation names. The element for each subpopulation will be either a single value for an unstratified sample or a named vector of values for a stratified sample. The default is NULL.

Example popsize for a stratified sample:
popsize = list("Pop 1"=c("Stratum 1"=750,
"Stratum 2"=500,
"Stratum 3"=250),
"Pop 2"=list("SubPop 1"=c("Stratum 1"=350,
"Stratum 2"=250,
"Stratum 3"=150),
"SubPop 2"=c("Stratum 1"=250,
"Stratum 2"=150,
"Stratum 3"=100),
"SubPop 3"=c("Stratum 1"=150,
"Stratum 2"=150,
"Stratum 3"=75)),
"Pop 3"=NULL)

Example popsize for an unstratified sample:
popsize = list("Pop 1"=1500,
"Pop 2"=list("SubPop 1"=750,
"SubPop 2"=500,
"SubPop 3"=375),
"Pop 3"=NULL)

`popcorrect` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster and stage1size, and for the support variable of the design argument.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.
sizeweight	Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
testname	Name of the test statistic to be reported in the output data frame. Choices for the name are: "Wald", "Wald_F", "Mean_Eigenvalue", "Mean_Eigenvalue_F", "Satterthwaite", and "Satterthwaite_F". The default is "Wald_F".
nclass	Number of classes into which the CDFs will be divided (binned), which must equal at least two. The default is 3.
spsurvey.obj	A list of class spsurvey.analysis that was produced by the function spsurvey.analysis. Depending on input to that function, some elements of the list may be NULL. The default is NULL.

Value

Data frame of CDF test results for all pairs of subpopulations within each population Type for every response variable. The data frame includes the test statistic specified by argument testname plus its degrees of freedom and p-value.

Other Functions Required

dframe.check check site IDs, the sites data frame, the subpop data frame, and the data.cat data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

vecprint takes an input vector and outputs a character string with line breaks inserted

uniqueID creates unique site IDs by appending a unique number to each occurrence of a site ID

input.check check input values for errors, consistency, and compatibility with analytical functions

cdf.test test for differences between cumulative distribution functions (CDFs)

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
mysiteID <- paste("Site", 1:100, sep="")
n <- 200
mysiteID <- paste("Site", 1:n, sep="")
```

```
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(TRUE, n))
mysubpop <- data.frame(
  siteID=mysiteID,
  Resource_Class=sample(c("Agr", "Forest", "Urban"), n, replace=TRUE))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(n, 10, 100),
  xcoord=runif(n),
  ycoord=runif(n),
  stratum=rep(c("Stratum1", "Stratum2"), n/2))
mypopsize <- list(
  Resource_Class=list(Agr=c(Stratum1=2500, Stratum2=1500),
                      Forest=c(Stratum1=1000, Stratum2=500),
                      Urban=c(Stratum1=600, Stratum2=450)))
ContVar <- numeric(n)
tst <- mysubpop$Resource_Class == "Agr"
ContVar[tst] <- rnorm(sum(tst), 10, 1)
tst <- mysubpop$Resource_Class == "Forest"
ContVar[tst] <- rnorm(sum(tst), 10.1, 1)
tst <- mysubpop$Resource_Class == "Urban"
ContVar[tst] <- rnorm(sum(tst), 10.5, 1)
mydata.cont <- data.frame(
  siteID=mysiteID,
  ContVar=ContVar)
cont.cdftest(sites=mysites, subpop=mysubpop, design=mydesign,
  data.cont=mydata.cont, popsize=mypopsize, testname="Mean_Eigenvalue")
```

`cov.panel.dsgn` *Covariance Matrix for a Panel Design*

Description

Covariance structure accounts for the panel design and the four variance components: unit variation, period variation, unit by period interaction variation and index (or residual) variation. The model incorporates unit, period, unit by period, and index variance components. It also includes a provision for unit correlation and period autocorrelation.

Usage

```
cov.panel.dsgn(paneldsgn = matrix(50, 1, 10), nrepeats = 1,
  unit.var = NULL, period.var = NULL, unitperiod.var = NULL,
  index.var = NULL, unit.rho = 1, period.rho = 0)
```

Arguments

`paneldsgn`	A matrix (dimensions: number of panels (rows) by number of periods (columns)) containing the number of units visited for each combination of panel and period.

	Default is matrix(50, 1, 10) which is a single panel of 50 units visited 10 times, typical time is a period.
nrepeats	Either NULL or a list of matrices the same length as paneldsgn specifying the number of revisits made to units in a panel in the same period for each design. Specifying NULL indicates that number of revisits to units is the same for all panels and for all periods and for all panel designs. The default is NULL, a single visit. Names must match list names in paneldsgn.
unit.var	The variance component estimate for unit (the default is Null)
period.var	The variance component estimate for period (the default is Null)
unitperiod.var	The variance component estimate for unit by period interaction (the default is Null)
index.var	The variance component estimate for index error (the def0ult is Null)
unit.rho	unit correlation across periods (the default is 1)
period.rho	period autocorrelation (the default is 0)

Details

Covariance structure accounts for the panel design and the four variance components: unit variation, period variation, unit by period interaction variation and index (or residual) variation. Uses the model structure defined by Urquhart.

If nrepeats is NULL, then no units sampled more than once in a specific panel, period combination) aned then unit by period and index variances are added together or user may have only estimated unit, period and unit by period variance components so that index component is zero It calculates the covariance matrix for the simple linear regression. The standard error for a linear trend coeficient is the square root of the variance.

Value

A list containing the covariance matrix (cov) for the panel design, the input panel design (paneldsgn), the input nrepeats design (nrepeats.dsgn) and the function call.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

References

Urquhart, N. S., W. S. Overton, et al. (1993) Comparing sampling designs for monitoring ecological status and trends: impact of temporal patterns. In: *Statistics for the Environment.* V. Barnett and K. F. Turkman. John Wiley & Sons, New York, pp. 71-86.

Urquhart, N. S. and T. M. Kincaid (1999). Designs for detecting trends from repeated surveys of ecological resources. *Journal of Agricultural, Biological, and Environmental Statistics*, **4(4)**, 404-414.

Urquhart, N. S. (2012). The role of monitoring design in detecting trend in long-term ecological monitoring studies. In: *Design and Analysis of Long-term Ecological Monitoring Studies.* R. A.

Gitzen, J. J. Millspaugh, A. B. Cooper, and D. S. Licht (eds.). Cambridge University Press, New York, pp. 151-173.

See Also

revisit_dsgn create a panel revisit design

revisit_bibd create a balanced incomplete block panel revisit design

revisit_rand create a revisit design with random assignment to panels and time periods

panel_summary summarize characteristics of a revisit panel design

power.dsgn power calculation for multiple panel designs

plot_powerpaneldesign plot power curves for panel designs

---

dcdf.prop *Deconvoluted Cumulative Distribution Function Estimate for Proportion*

Description

This function calculates an estimate of the deconvoluted cumulative distribution function (CDF) for the proportion of a discrete or an extensive resource. The simulation extrapolation deconvolution method (Stefanski and Bay, 1996) is use to deconvolute measurement error variance from the response. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the estimate. The numerator of the ratio estimates the total of the resource equal to or less than a specified value. The denominator of the ratio estimates the size of the resource. For a discrete resource size is the number of units in the resource. For an extensive resource size is the extent (measure) of the resource, i.e., length, area, or volume. The function can accomodate single-stage and two-stage samples.

Usage

```
dcdf.prop(g, wgt, cluster.ind, cluster, wgt1)
```

Arguments

g	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.

Value

The deconvoluted CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`dcdf.size.prop` *Deconvoluted Size-Weighted Cumulative Distribution Function for Proportion*

Description

This function calculates an estimate of the size-weighted, deconvoluted cumulative distribution function (CDF) for the proportion of a discrete resource. The simulation extrapolation deconvolution method (Stefanski and Bay, 1996) is use to deconvolute measurement error variance from the response. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the estimate. The numerator of the ratio estimates the size-weighted total of the resource equal to or less than a specified value. The denominator of the ratio estimates the sum of the size-weights for the resource. The function can accomodate single-stage and two-stage samples.

Usage

```
dcdf.size.prop(g, wgt, cluster.ind, cluster, wgt1, swgt, swgt1)
```

Arguments

`g`	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.
`swgt`	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
`swgt1`	Vector of the stage one size-weight for each site.

Value

The deconvoluted CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`dcdf.size.total` *Deconvoluted Size-weighted Cumulative Distribution function for Total*

Description

This function calculates an estimate of the size-weighted, deconvoluted cumulative distribution function (CDF) for the total of a discrete resource. The simulation extrapolation deconvolution method (Stefanski and Bay, 1996) is use to deconvolute measurement error variance from the response. If the known sum of the size-weights of the resource is provided, the classic ratio estimator is used to calculate the estimate. That estimator is the product of the known sum of the size-weights of the resource and the Horvitz- Thompson ratio estimator, where the latter is the ratio of two Horvitz- Thompson estimators. The numerator of the ratio estimates the size-weighted total of the resource equal to or less than a specified value. The denominator of the ratio estimates the sum of the size-weights of the resource. If the known sum of the size-weights of the resource is not provided, the Horvitz-Thompson estimator of the size-weighted total of the resource equal to or less than a specified value is used to calculate the estimate. The function can accomodate single-stage and two-stage samples.

Usage

```
dcdf.size.total(g, wgt, cluster.ind, cluster, wgt1, popsize, swgt, swgt1)
```

Arguments

`g`	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.
`popsize`	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
swgt1	Vector of the stage one size-weight for each site.

Value

The deconvoluted CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

dcdf.total *Deconvoluted Cumulative Distribution Function for Total*

Description

This function calculates an estimate of the deconvoluted cumulative distribution function (CDF) for the total of a discrete or an extensive resource. The simulation extrapolation deconvolution method (Stefanski and Bay, 1996) is use to deconvolute measurement error variance from the response. If the known extent of the resource is provided, the classic ratio estimator is used to calculate the estimate. That estimator is the product of the known extent of the resource and the Horvitz-Thompson ratio estimator, where the latter is the ratio of two Horvitz-Thompson estimators. The numerator of the ratio estimates the total of the resource equal to or less than a specified value. The denominator of the ratio estimates the extent of the resource. If the known extent of the resource is not provided, the Horvitz-Thompson estimator of the total of the resource equal to or less than a specified value is used to calculate the estimate. For a discrete resource, size is the number of units in the resource. For an extensive resource, size is the measure of the resource, i.e., length, area, or volume. The function can accomodate single-stage and two-stage samples.

Usage

```
dcdf.total(g, wgt, cluster.ind, cluster, wgt1, popsize)
```

Arguments

g	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.

popsize — Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

Value

The deconvoluted CDF estimate.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

dcdfvar.prop *Variance Estimate for Deconvoluted CDF for Proportion*

Description

This function calculates variance estimates of the estimated, deconvoluted cumulative distribution function (CDF) for the proportion of a discrete or a continuous resource. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. When variance of the estimated measurement error variance is nonzero, a correction factor is added to the estimated variance of the CDF. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
dcdfvar.prop(g, dg, var.sigma, wgt, x, y, cdfest, stratum.ind,
  stratum.level, cluster.ind, cluster, wgt1, x1, y1, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

g — Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.

dg — Vector of the values of the derivative of the deconvolution function g(.) evaluated at val for the response value for each site.

var.sigma — Variance of the measurement error variance.

wgt — Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.

x	Vector of the x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of the y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
cdfest	The CDF estimate.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.

warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

dcdfvar.size.prop *Variance Estimate for Deconvoluted Size-weighed CDF for Proportion*

Description

This function calculates variance estimates of the estimated, deconvoluted, size-weighted cumulative distribution function (CDF) for the proportion of a discrete resource. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. When variance of the estimated measurement error variance is nonzero, a correction factor is added to the estimated variance of the CDF. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
dcdfvar.size.prop(g, dg, var.sigma, wgt, x, y, cdfest, stratum.ind,
  stratum.level, cluster.ind, cluster, wgt1, x1, y1, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, swgt, swgt1, vartype, warn.ind, warn.df,
  warn.vec)
```

Arguments

g	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
dg	Vector of the values of the derivative of the deconvolution function g(.) evaluated at val for the response value for each site.
var.sigma	Variance of the measurement error variance.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
cdfest	The CDF estimate.
stratum.ind	Logical value that indicates whether the sample is stratied, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify

	both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.
swgt1	Vector of the stage one size-weight for each site.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	LogicaL value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

dcdfvar.size.total *Variance Estimate of Deconvoluted Size-weighted CDF for Total*

Description

This function calculates variance estimates of the estimated, deconvoluted, size-weighted cumulative distribution function (CDF) for the total of a discrete resource. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. When variance of the estimated measurement error variance is nonzero, a correction factor is added to the estimated variance of the CDF. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
dcdfvar.size.total(g, dg, var.sigma, wgt, x, y, cdfest, stratum.ind,
  stratum.level, cluster.ind, cluster, wgt1, x1, y1, popsize, pcfactor.ind,
  pcfsize, N.cluster, stage1size, support, swgt, swgt1, vartype, warn.ind,
  warn.df, warn.vec)
```

Arguments

g	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
dg	Vector of the values of the derivative of the deconvolution function g(.) evaluated at val for the response value for each site.
var.sigma	Variance of the measurement error variance.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
x	Vector of the x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
y	Vector of the y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
cdfest	The CDF estimate.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum.level	The stratum level.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
popsize	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor

and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".

`support` Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.

`swgt` Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample.

`swgt1` Vector of the stage one size-weight for each site.

`vartype` The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.

`warn.ind` Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.

`warn.df` Data frame for storing warning messages.

`warn.vec` Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which

Other Functions Required

`localmean.weight` calculate the weighting matrix for the local mean variance estimator

`localmean.var` calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`dcdfvar.total` *Variance Estimate of Deconvoluted CDF for Total*

Description

This function calculates variance estimates of the estimated, deconvoluted cumulative distribution function (CDF) for the total of a discrete or a continuous resource. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. When variance of the estimated measurement error variance is nonzero, a correction factor is added to the estimated variance of the CDF. The function can accomodate single-stage and two-stage samples. Finite population and continuous population correction factors can be utilized in variance estimation.

Usage

```
dcdfvar.total(g, dg, var.sigma, wgt, x, y, cdfest, stratum.ind,
  stratum.level, cluster.ind, cluster, wgt1, x1, y1, popsize, pcfactor.ind,
  pcfsize, N.cluster, stage1size, support, vartype, warn.ind, warn.df,
  warn.vec)
```

Arguments

`g`	Vector of the values of the deconvolution function g(.) evaluated at a specified value for the response value for each site.
`dg`	Vector of the derivative of the deconvolution function g(.) evaluated at val for the response value for each site.
`var.sigma`	Variance of the measurement error variance.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single- stage sample or the stage two weight for a two-stage sample.
`x`	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
`y`	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
`cdfest`	The CDF estimate.
`stratum.ind`	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
`stratum.level`	The stratum level.
`cluster.ind`	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.

cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
wgt1	Vector of the final adjusted stage one weight for each site.
x1	Vector of the stage one x-coordinate for location for each site.
y1	Vector of the stage one y-coordinate for location for each site.
popsize	Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

An object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which

Other Functions Required

`localmean.weight` calculate the weighting matrix for the local mean variance estimator

`localmean.var` calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

decon_data *Small Lakes in Florida for Examining CDF Deconvolution*

Description

A dataset containing attributes for small lakes in Florida that can be used to examine CDF deconvolution.

Usage

```
decon_data
```

Format

A data frame with 930 rows and 6 attributes:

xcoord simulated x-coordinate value.

ycoord simulated y-coordinate value.

Richness simulated species richness value.

Richness_25 species richness value plus 25% measurement error variance.

Richness_50 species richness value plus 50% measurement error variance.

Richness_100 species richness value plus 100% measurement error variance.

dframe.check *Internal Check on Survey Analysis Data Frames*

Description

This function checks site IDs, the sites data frame, the subpop data frame, the data.cat data frame, the data.cont data frame, the data.ar data frame, and the data.rr data frame to assure valid contents. If they do not exist, then the sites data frame and the subpop data frame are created.

Usage

```
dframe.check(sites, design, subpop, data.cat, data.cont, data.risk,
  design.names)
```

Arguments

sites	The sites data frame.
design	The design data frame.
subpop	The subpop data frame.
data.cat	The data.cat data frame of categorical response variables.
data.cont	The data.cont data frame of continuous response variables.
data.risk	The data.ar or data.rr data frame of categorical response and stressor variables.
design.names	The names for the design data frame.

Value

List consisting of the sites data frame, design data frame, subpop data frame, data.cat data frame, and data.cont data frame.

Other Functions Required

vecprint takes an input vector and outputs a character string with line breaks inserted

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

dsgnsum *Summary of a Survey Design*

Description

This function summarizes the sites selected for a survey design by producing contingency tables containing the cross-tabluation of number of sites for survey design variables and, optionally, for auxiliary variables.

Usage

```
dsgnsum(spsample, auxvar = NULL)
```

Arguments

spsample	An object of class SpatialDesign produced by either the grts or irs functions that contains survey design information and additional attribute (auxiliary) variables.
auxvar	Vector containing the names of columns in the data slot of the

Value

A list containing two components named DesignSum and AuxVarSum. DesignSum is a list of contingency tables containing the cross-tabulation of number of sites for the following combinations of survey design variables:

- multidensity category (mdcaty) and stratum
- stratum and panel
- mdcaty, panel, and stratum

AuxVarSum is a list of contingency tables containing the cross-tabulation of number of sites for each auxiliary variable and the design variables mdcaty, panel, and stratum.

In addition the output list plus labeling information is printed to the console.

Other Functions Required

vecprint takes an input vector and outputs a character string with line breaks inserted

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
## Not run:
design <- list(
  Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10),
  Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal",
    caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
samp <- grts(design=design, DesignID="Test.Site", type.frame="area",
  src.frame="shapefile", in.shape="shapefile", stratum="stratum",
  mdcaty="mdcaty", shapefile=TRUE, out.shape="sample.shp")
dsgnsum(samp, auxvar=c("ecoregion", "state"))

## End(Not run)
```

---

examine *Examine Variables in a Data Frame*

Description

This function examines variables in a data frame by printing either a table or the results of a call to the describe function in the Hmisc package.

Usage

```
examine(dframe, subpop = NULL, ord = TRUE, cmax = 50)
```

Arguments

dframe	Data frame.
subpop	Character string identifying a variable in dframe that is used to group output. A separate table or call to the describe function is printed for each unique value in the variable. The default value is NULL.
ord	Logical value that controls the order in which the variables in dframe are processed. TRUE mean that variables are processed in sorted order. FALSE means that variables are processed in the order in which they occur in dframe. The default value is TRUE.
cmax	Numeric value that controls whether a call to table or a call to describe is used to process variables in dframe. If the number of unique values in a variable is less than or equal to cmax, then table is called. If the number of unique values in a variable is greater than cmax, then describe is called. The default value is 50.

Value

Tables and/or the output from calls to describe are printed. The function returns NULL invisibly.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
df <- data.frame(
  SiteID = 1:100,
  Catvar = sample(LETTERS[1:5], 100, TRUE),
  Contvar = rnorm(100, 10, 1),
  Gender = rep(c("Male", "Female"), rep(50, 2)))
examine(df, ord = FALSE)
examine(df, "Gender", FALSE)
```

FL_lakes *Small Lakes in Florida*

Description

A dataset containing attributes for small lakes in Florida.

Usage

```
FL_lakes
```

Format

A data frame with 930 rows and 11 attributes:

siteID site ID value.

xcoord Albers projection x-coordinate.

ycoord Albers projection y-coordinate.

wgt survey design weight.

Basin stream basin code.

Status site evaluation status code.

TNT target or nontarget category for the site evaluation status code.

pH_Cat pH category.

Coliform_Cat fecal coliform count category.

Oxygen dissolved oxygen value.

Turbidity turbidity value.

framesum *Summary of the Sample Frame for a Survey Design*

Description

This function summarizes the frame for a survey design. When type.frame equals "finite", summary is a count of number of units in att.frame for cross-tabulation of stratum, mdcaty, and auxvar. When type.frame equals "linear" or "area", summary is the sum of length or area for units for cross-tabulation of stratum, mdcaty, and auxvar. If argument mdcaty or argument stratum equals NULL or if both arguments equal NULL, then the cross-tabulation is performed without use of the design variable(s).

Usage

```
framesum(att.frame, design, type.frame = "finite", stratum = NULL,
  mdcaty = NULL, auxvar = NULL, units.in = "Number", scale = 1,
  units.out = "Number")
```

Arguments

att.frame	Data frame composed of attributes associated with elements in the frame, which must contain the columns used for stratum and mdcaty (if required by the survey design).
design	Named list of stratum design specifications which are also lists. Stratum names must be subset of values in stratum argument. Each stratum list has four components: **panel** named vector of sample sizes for each panel in stratum **seltype** the type of random selection, which must be one of following: "Equal" - equal probability selection, "Unequal" - unequal probability selection by the categories specified in caty.n and mdcaty, or "Continuous" - unequal probability selection proportional to auxiliary variable mdcaty **caty.n** if seltype equals "Unequal", a named vector of sample sizes for each category specified by mdcaty, where sum of the sample sizes must equal sum of the panel sample sizes, and names must be a subset of values in mdcaty **over** number of replacement sites ("oversample" sites) for the entire design, which is set equal to 0 if none are required) Example design: design=list(Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10), Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal", caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
type.frame	The type of frame, which must be one of following: "finite", "linear", or "area". The default is "finite".
stratum	Name of the column from att.frame that identifies stratum membership for each element in the frame. If stratum equals NULL, the design is unstratified. The default is NULL.

mdcaty	Name of the column from att.frame that identifies the unequal probability category for each element in the frame. The default is NULL.
auxvar	Vector containing the names of columns from att.frame that identify auxiliary variables to be used to summarize frame size. The default is NULL.
units.in	Character string giving the name of units used to measure size in the frame. The default is "Number".
scale	The scale factor used to change units.in to units.out. For example, use 1000 to change "Meters" to "Kilometers". The default is 1.
units.out	Character string giving the name of units used to measure size in the results. The default is "Number".

Value

A list containing two components named DesignSize and AuxVarSize. DesignSize summarizes the frame for survey design variables, and AuxVarSize summarizes the frame for auxiliary variables. DesignSize is either a table (for type.frame equals "finite") or an array (for type.frame equals "linear" or "area") that contains the cross-tabulation of frame extent for design variables multidensity category (mdcaty) and stratum, where extent of the frame is the number of sites for type.frame equals "finite", the sum of resource length for type.frame equals "linear", or the sum of resource area for type.frame equals "area". AuxVarSize is a list containing a component for each auxiliary variable, where each component of the list is one of the following: (1) if the type of random selection does not equal "Continuous" for any stratum, each component is either a table (for type.frame equals "finite") or an array (for type.frame equals "linear" or "area") that contains the cross-tabulation of frame extent for mdcaty, stratum, and the auxiliary variable or (2) if type of random selection equals "Continuous" for all strata, each component is either a table (finite frame) or an array (linear or area frame) containing the cross-tabulation of frame extent for stratum and the auxiliary variable. In addition the output list plus labeling information is printed to the console.

Other Functions Required

vecprint takes an input vector and outputs a character string with line breaks inserted

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
## Not run:
attframe <- read.dbf("shapefile.shp")
design <- list(
  Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10),
  Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal",
    caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
framesum(att.frame=attframe, design=design, type.frame="area",
  stratum="stratum", mdcaty="mdcaty", auxvar=c("ecoregion",
  "state"), units.in="Meters", scale=1000, units.out="Kilometers")

## End(Not run)
```

geodalbers *Project Latitude/Longitude to Albers Projection*

Description

Project Latitude/Longitude to Albers Projection

Usage

```
geodalbers(lon, lat, sph = "GRS80", clon = -96, clat = 23,
  sp1 = 29.5, sp2 = 45.5)
```

Arguments

lon	Vector of longitude (decimal degrees) values to be projected to Albers.
lat	Vector of latitude (decimal degrees) values to be projected to Albers.
sph	Spheroid options: Clarke1866, GRS80, WGS84. The default is GRS80.
clon	Center longitude (decimal degrees). The default is -96.
clat	Origin latitude (decimal degrees). The default is 23.
sp1	Standard parallel 1 (decimal degrees). The default is 29.5.
sp2	Standard parallel 2 (decimal degrees). The default is 45.5.

Value

Data frame of Albers x-coordinate and y-coordinate projections for latitude and longitude.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

grts *Select a Generalized Random-Tesselation Stratified (GRTS) Sample*

Description

This function select a GRTS sample of a finite, linear, or area resource. Frame elements must be located in 1- or 2-dimensional coordinate system. Sample may be equal probability or unequal probability (either categorical or proportional to auxiliary variable). May designate panels of sites for surveys over time.

Usage

```
grts(design, DesignID = "Site", SiteBegin = 1, type.frame = NULL,
  src.frame = "shapefile", in.shape = NULL, sf.object = NULL,
  sp.object = NULL, att.frame = NULL, id = NULL, xcoord = NULL,
  ycoord = NULL, stratum = NULL, mdcaty = NULL, startlev = NULL,
  maxlev = 11, maxtry = NULL, shift.grid = TRUE,
  do.sample = rep(TRUE, length(design)), shapefile = TRUE,
  prjfilename = NULL, out.shape = "sample.shp")
```

Arguments

design	Named list of stratum design specifications which are also lists. Stratum names must be subset of values in stratum argument. Each stratum list has four components: **panel** named vector of sample sizes for each panel in stratum **seltype** the type of random selection, which must be one of following: "Equal" - equal probability selection, "Unequal" - unequal probability selection by the categories specified in caty.n and mdcaty, or "Continuous" - unequal probability selection proportional to auxiliary variable mdcaty **caty.n** if seltype equals "Unequal", a named vector of sample sizes for each category specified by mdcaty, where sum of the sample sizes must equal sum of the panel sample sizes, and names must be a subset of values in mdcaty **over** number of replacement sites ("oversample" sites) for the entire design, which is set equal to 0 if none are required) Example design for a stratified sample: design=list(Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10), Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal", caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75)) Example design for an unstratified sample: design <- list(None=list(panel=c(Panel1=50, Panel2=100, Panel3=50), seltype="Unequal", caty.n=c("Caty 1"=50, "Caty 2"=25, "Caty 3"=25, "Caty 4"=25, "Caty 5"=75), over=100))
DesignID	Name for the design, which is used to create a site ID for each site. The default is "Site".
SiteBegin	Number to use for first site in the design. The default is 1.
type.frame	The type of frame, which must be one of following: "finite", "linear", or "area". The default is NULL.
src.frame	Source of the frame, which equals "sf.object" if the frame is contained in an sf package object, "shapefile" if the frame is to be read from a shapefile, "sp.object" if the frame is obtained from an sp package object, or "att.frame" if type.frame equals "finite" and the frame is included in att.frame. The default is "shapefile".
in.shape	Name of a shapefile containing the frame, which is required when src.frame equals "shapefile". The shapefile name should include the ".shp" extension. If the name does not include that extension, it will be added. The default is NULL.

sf.object	An sf package object containing the frame, which is required when src.frame equals "sf.object". The default is NULL.
sp.object	Name of the sp package object when src.frame equals "sp.object". The default is NULL.
att.frame	Data frame composed of attributes associated with elements in the frame. If src.frame equals "att.frame", then att.frame must include columns that contain x-coordinates and y-coordinates for each element in the frame. If src.frame does not equal "att.frame" and att.frame is not equal to NULL, then an sf object is created from att.frame and the geometry column from the object named "sf.object" that is created by the function. The default is NULL.
id	This argument is depricated.
xcoord	Character string containing the name of the column from att.frame that identifies x-coordinates when src.frame equals "att.frame". If xcoord equals NULL, then xcoord is given the value "x". The default is NULL.
ycoord	Character string containing the name of the column from att.frame that identifies y-coordinates when src.frame equals "att.frame". If ycoord equals NULL, then ycoord is given the value "y". The default is NULL.
stratum	Character string containing the name of the column from att.frame that identifies stratum membership for each element in the frame. If stratum equals NULL, the design is unstratified, and a column named "stratum" (with all of its elements equal to the stratum name specified in design) is added to att.frame. The default is NULL.
mdcaty	Character string containing the name of the column from att.frame that identifies the unequal probability category for each element in the frame. The default is NULL.
startlev	Initial number of hierarchical levels to use for the GRTS grid, which must be less than or equal to maxlev (if maxlev is specified) and cannot be greater than 11. The default is NULL.
maxlev	Maxmum number of hierarchical levels to use for the GRTS grid, which cannot be greater than 11. The default is 11.
maxtry	This argument is depricated.
shift.grid	Option to randomly shift the hierarchical grid, where TRUE means shift the grid and FALSE means do not shift the grid, which is useful if one desires strict spatial stratification by hierarchical grid cells. The default is TRUE.
do.sample	Named vector that provides the option controlling sample selection for each stratum, where TRUE means select a sample from a stratum and FALSE means return the sample frame for a stratum in reverse hierarchical order. Note that FALSE can only be used when type.frame equals "points" and seltype equals "Equal". Names for the vector must match the names in design. If the vector is not named, then the names in design are used. The default is TRUE for each stratum.
shapefile	Option to create a shapefile containing the survey design information, where TRUE equals create a shapefile and FALSE equals do not create a shapefile. The default is TRUE.
prjfilename	This argument is depricated.
out.shape	Name of the output shapefile. The default is "sample.shp".

Value

An object of class SpatialDesign containing the survey design information and any additional attribute variables that were provided. Optionally, a shapefile can be created that contains the survey design information.

Other Functions Required

grtsarea select a GRTS sample of an area resource

grtslin select a GRTS sample of a linear resource

grtspts select a GRTS sample of a finite resource

SpatialPoints sp package function to create an object of class SpatialPoints

SpatialPointsDataFrame sp package function to create an object of class SpatialPointsDataFrame

Author(s)

Tom Kincaid emailKincaid.Tom@epa.gov

Examples

```
## Not run:
  test_design <- list(
    Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10),
    Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal",
      caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
  test.sample <- grts(design=test_design, DesignID="TestSite",
    type.frame="area", src.frame="shapefile", in.shape="test_shapefile.shp",
    stratum="test_stratum", mdcaty="test_mdcaty", shapefile=TRUE,
    out.shape="test_sample.shp")

## End(Not run)
```

---

grtsarea — *Select a Generalized Random-Tesselation Stratified (GRTS) Sample of an Area Resource*

Description

This function select a GRTS sample of an area resource. The function uses hierarchical randomization to ensure that the sample will include no more than one point per cell and then picks a point in selected cells.

Usage

```
grtsarea(areaframe, samplesize = 100, SiteBegin = 1,
  shift.grid = TRUE, startlev = NULL, maxlev = 1, maxtry = NULL)
```

Arguments

areaframe	The sf object containing attributes: id, mdcaty and mdm.
samplesize	Number of points to select in the sample. The default is 100.
SiteBegin	First number to start siteID numbering. The default is 1.
shift.grid	Option to randomly shift the hierarchical grid. The default is TRUE.
startlev	Initial number of hierarchical levels to use for the GRTS grid, which must be less than or equal to maxlev (if maxlev is specified) and cannot be greater than 11. The default is NULL.
maxlev	Maximum number of hierarchical levels to use for the GRTS grid, which cannot be greater than 11. The default is 11.
maxtry	This argument is depricated.

Value

Data frame of sample points containing: siteID, id, x, y, mdcaty, and weight.

Other Functions Required

numLevels determines the number of levels for hierarchical randomization

constructAddr constructs the hierarchical address for sample points

ranho constructs the randomized hierarchical address for sample points

pickGridCells selects grid cells that get a sample point

insideAreaGridCell determines feature ID value and clipped polygon area for each feature contained in a selected grid cell

selectFeatureID identifies a feature ID from which to select a sample point

pickSamplePoints selects sample points from an sf object

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

---

grtslin *Select a Generalized Random-Tesselation Stratified (GRTS) Sample of a Linear Resource*

Description

This function select a GRTS sample of a linear resource. The function uses hierarchical randomization to ensure that the sample will include no more than one point per cell and then picks a point in selected cells.

Usage

```
grtslin(linframe, samplesize = 100, SiteBegin = 1, shift.grid = TRUE,
  startlev = NULL, maxlev = 1)
```

Arguments

linframe	The sf object containing attributes: id, mdcaty, and mdm.
samplesize	Number of points to select in the sample. The default is 100.
SiteBegin	First number to start siteID numbering. The default is 1.
shift.grid	Option to randomly shift the hierarchical grid. The default is TRUE.
startlev	Initial number of hierarchical levels to use for the GRTS grid, which must be less than or equal to maxlev (if maxlev is specified) and cannot be greater than 11. The default is NULL.
maxlev	Maximum number of hierarchical levels to use for the GRTS grid, which cannot be greater than 11. The default is 11.

Value

Data frame of sample points containing: siteID, id, x, y, mdcaty, and weight.

Other Functions Required

numLevels determines the number of levels for hierarchical randomization

constructAddr constructs the hierarchical address for sample points

ranho constructs the randomized hierarchical address for sample points

pickGridCells selects grid cells that get a sample point

insideLinearGridCell determines feature ID value and clipped linestring length for each feature contained in a selected grid cell

selectFeatureID identifies a feature ID from which to select a sample point

pickSamplePoints selects sample points from an sf object

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov> Tony Olsen <Olsen.Tony@epa.gov>

---

grtspts — *Select a Generalized Random-Tesselation Stratified (GRTS) Sample of a Finite Resource*

Description

This function select a GRTS sample of a finite resource. This function uses hierarchical randomization to ensure that the sample will include no more than one point per grid cell and then picks a point in selected cells.

Usage

```
grtspts(ptsframe, samplesize = 100, SiteBegin = 1, shift.grid = TRUE,
  do.sample = TRUE, startlev = NULL, maxlev = 11)
```

Arguments

ptsframe	The sf object containing attributes: id, x, y, mdcaty, and mdm.
samplesize	Number of points to select in the sample. The default is 100.
SiteBegin	First number to start siteID numbering. The default is 1.
shift.grid	Option to randomly shift the hierarchical grid. The default is TRUE.
do.sample	Option to select a sample, where TRUE means select a sample and FALSE means return the entire sample frame in reverse hierarchical order. The default is TRUE.
startlev	Initial number of hierarchical levels to use for the GRTS grid, which must be less than or equal to maxlev (if maxlev is specified) and cannot be greater than 11. The default is NULL.
maxlev	Maximum number of hierarchical levels to use for the GRTS grid, which cannot be greater than 11. The default is 11.

Value

Data frame of sample points containing: siteID, id, x, y, mdcaty, and weight.

Other Functions Required

numLevels determines the number of levels for hierarchical randomization

constructAddr constructs the hierarchical address for sample points

ranho constructs the randomized hierarchical address for sample points

pickGridCells selects grid cells that get a sample point

pickFiniteSamplePoints pick sample point(s) from selected cells

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov> Tony Olsen <Olsen.Tony@epa.gov>

---

input.check *Check Input Values for Analytical Functions*

Description

This function checks input values for errors, consistency, and compatibility with analytical functions.

Usage

```
input.check(nresp, wgt, sigma, var.sigma, xcoord, ycoord, stratum.ind,
  stratum, stratum.levels, nstrata, cluster.ind, cluster, cluster.levels,
  ncluster, wgt1, xcoord1, ycoord1, popsize, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, swgt.ind, swgt, swgt1, vartype, conf,
  cdfval = NULL, pctval = NULL, subpop = NULL)
```

Arguments

nresp	The number of response values.
wgt	Vector of the final adjusted weights.
sigma	Measurement error variance.
var.sigma	Variance of the measurement error variance.
xcoord	Vector of the x-coordinates for location.
ycoord	Vector of the y-coordinates for location.
stratum.ind	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
stratum	Vector of the stratum codes.
stratum.levels	Levels of the stratum variable.
nstrata	Number of strata.
cluster.ind	Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two- stage sample.
cluster	Vector of the stage one sampling unit codes.
cluster.levels	Factor levels of the stage one sampling unit codes.
ncluster	The number of stage one sampling units in the sample.
wgt1	Vector of the final adjusted stage one weights.
xcoord1	Vector of the stage one x-coordinates for location.
ycoord1	Vector of the stage one y-coordinates for location.
popsize	Known size of the resource.
pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	Number of stage one sampling units in the resource.
stage1size	Known size of the stage one sampling units.
support	Vector of the support for each sampling unit.
swgt.ind	Logical value that indicates whether the sample is a size- weighted sample, where TRUE = a size-weighted sample and FALSE = not a size-weighted sample.
swgt	Vector of the size-weight for each site.
swgt1	Vector of the stage one size-weight for each site.

vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
conf	The confidence level.
cdfval	Vector of the set of values at which the CDF is estimated.
pctval	Vector of the set of values at which percentiles are estimated.
subpop	Data frame describing sets of populations and subpopulations for which estimates will be calculated.

Value

A list consisting of popsize, pcfsize, N.cluster, and stage1size.

Other Functions Required

vecprint takes an input vector and outputs a character string with line breaks inserted

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

input.format *Format an Input Value*

Description

This function formats an input value of class numeric, character, or factor. For a numeric value, the number of digits after the decimal point can be specified. A factor value is converted to character. Missing values are allowed.

Usage

```
input.format(x, n.digits = 2, miss = "NA")
```

Arguments

x	The input value.
n.digits	The number of digits after the decimal point, which can be zero. The default is 2.
miss	The missing value code expressed as a character string. The default is "NA".

Value

A value of mode character that is one of the following, as appropriate: (1) character representation of a real number with the specified number of digits after the decimal point when the input numeric value is a real number, (2) character representation of an integer when the input numeric value is an integer, (3) the original value when the input value is class character or factor, or (4) the missing value code when the input value is missing.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

insideAreaGridCell *Calculate Clipped Feature Areas in a Set of Grid Cells*

Description

For each grid cell, this function calculates the clipped area of each polygon feature contained in the cell.

Usage

```
insideAreaGridCell(sfobject, rdx.u, xc, yc, dx, dy)
```

Arguments

sfobject	The sf polygon object.
rdx.u	Vector of cell IDs.
xc	Vector of x-coordinates for the grid cells.
yc	Vector of y-coordinates for the grid cells.
dx	The x-axis grid cell dimension.
dy	The y-axis grid cell dimension.

Value

Data frame containing the following variables: cellID, featureArea, and featureID.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

insideLinearGridCell *Calculate Clipped Feature Lengths in a Set of Grid Cells*

Description

For each grid cell, this function calculates the clipped length of each linestring feature contained in the cell.

Usage

```
insideLinearGridCell(sfobject, rdx.u, xc, yc, dx, dy)
```

Arguments

sfobject	The sf linestring object.
rdx.u	Vector of cell IDs.
xc	Vector of x-coordinates for the grid cells.
yc	Vector of y-coordinates for the grid cells.
dx	The x-axis grid cell dimension.
dy	The y-axis grid cell dimension.

Value

Data frame containing the following variables: cellID, featureLength, and featureID.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

interp.axis *Create Right-Side y-Axis Lables for a CDF Plot*

Description

This function creates right side y-axis labels for a CDF plot. It assumes that arguments cdfest.l and cdfest.r are strictly increasing. If argument yl.lab is less than the first cdfest.l vlaue, then the funciton assumes 0 for both cdfest.l and cdfest.r.

Usage

```
interp.axis(yl.lab, cdfest.l, cdfest.r)
```

Arguments

yl.lab	Vector of left side y-axis labels, which are the basis for interpolating cdfest.r values.
cdfest.l	Vector of CDF estimates corresponding to the left side y-axis.
cdfest.r	Vector of CDF estimates corresponding to the right side y-axis.

Value

A numeric vector consisting of the right side y-axis labels.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>
Tom Kincaid <Kincaid.Tom@epa.gov>

interp.cdf *Interpolate CDF Values at a Set of Percentiles*

Description

This function interpolates CDF values at a set of percentiles. The CDF values can be CDF estimates, CDF confidence bound estimates, or values at which the CDF is estimated (i.e., x-axis values). It is assumed that arguments cdfest.p and cdf.value are strictly increasing.

Usage

```
interp.cdf(pctval, cdfest.p, cdf.value)
```

Arguments

pctval	Vector of percentiles (expressed as percents) at which the CDF values are to be interpolated.
cdfest.p	Vector of CDF estimates in terms of proportions.
cdf.value	Vector of CDF values to be interpolated.

Value

A numeric vector consisting of the interpolated CDF values.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>
Tom Kincaid <Kincaid.Tom@epa.gov>

IN_streams *Streams in the Upper Wabash Basin in Indiana*

Description

A dataset containing attributes for streams in the Upper Wabash Basin in Indiana.

Usage

```
IN_streams
```

Format

A data frame with 100 rows and 11 attributes:

siteID site ID value.

xcoord Albers projection x-coordinate.

ycoord Albers projection y-coordinate.

wgt survey design weight.

Strahler_Cat Strahler order category.

Status site evaluation status code.

TNT target or nontarget category for the site evaluation status code.

IBI_Score IBI (index of biotic integrity) score.

IBI_Status status category of the IBI score.

QHEI_Score QHEI ((qualitative habitat evaluation index) score.

QHEI_Status status category of the QHEI score.

---

irs *Select an Independent Random Sample (IRS)*

Description

Select an independent random sample from a point, linear, or areal frame. Frame features must be located in a 1-dimensional or 2-dimensional coordinate system. Sample may be equal probability or unequal probability (either categorical or proportional to an auxiliary variable). May designate panels of sites for surveys over time.

Usage

```
irs(design, DesignID = "Site", SiteBegin = 1, type.frame = "finite",
  src.frame = "shapefile", in.shape = NULL, sf.object = NULL,
  sp.object = NULL, att.frame = NULL, id = NULL, xcoord = NULL,
  ycoord = NULL, stratum = NULL, mdcaty = NULL, maxtry = NULL,
  shapefile = TRUE, prjfilename = NULL, out.shape = "sample.shp")
```

Arguments

`design` Named list of stratum design specifications which are also lists. Stratum names must be subset of values in stratum argument. Each stratum list has four components:

panel named vector of sample sizes for each panel in stratum

seltype the type of random selection, which must be one of following: "Equal" - equal probability selection, "Unequal" - unequal probability selection by the categories specified in caty.n and mdcaty, or "Continuous" - unequal probability selection proportional to auxiliary variable mdcaty

caty.n if seltype equals "Unequal", a named vector of sample sizes for each category specified by mdcaty, where sum of the sample sizes must equal sum of the panel sample sizes, and names must be a subset of values in mdcaty

over number of replacement sites ("oversample" sites) for the entire design, which is set equal to 0 if none are required)

Example design for a stratified sample:
design=list(Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10), Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal", caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75)) Example design for an unstratified sample:
design <- list(None=list(panel=c(Panel1=50, Panel2=100, Panel3=50), seltype="Unequal", caty.n=c("Caty 1"=50, "Caty 2"=25, "Caty 3"=25, "Caty 4"=25, "Caty 5"=75), over=100))

`DesignID` Name for the design, which is used to create a site ID for each site. The default is "Site".

`SiteBegin` Number to use for first site in the design. The default is 1.

`type.frame` The type of frame, which must be one of following: "finite", "linear", or "area". The default is "finite".

`src.frame` Source of the frame, which equals "sf.object" if the frame is contained in an sf package object, "shapefile" if the frame is to be read from a shapefile, "sp.object" if the frame is obtained from an sp package object, or "att.frame" if type.frame equals "finite" and the frame is included in att.frame. The default is "shapefile".

`in.shape` Name of a shapefile containing the frame, which is required when src.frame equals "shapefile". The shapefile name should include the ".shp" extension. If the name does not include that extension, it will be added. The default is NULL.

`sf.object` An sf package object containing the frame, which is required when src.frame equals "sf.object". The default is NULL.

`sp.object` An sp package object containing the frame, which is required when src.frame equals "sp.object". The default is NULL.

`att.frame` Data frame composed of attributes associated with elements in the frame. If src.frame equals "att.frame", then att.frame must include columns that contain x-coordinates and y-coordinates for each element in the frame. If src.frame does not equal "att.frame" and att.frame is not equal to NULL, then an sf object is created from att.frame and the geometry column from the object named "sf.object" that is created by the function. The default is NULL.

`id` This argument is depricated.

`xcoord` Character string containing the name of the column from att.frame that identifies x-coordinates, which is required when src.frame equals "att.frame". The default is NULL.

`ycoord` Character string containing the name of the column from att.frame that identifies y-coordinates, which is required when src.frame equals "att.frame". The default is NULL.

stratum	Character string containing the name of the attribute in sf.object that identifies stratum membership for each feature. If stratum equals NULL, the design is unstratified, and an attribute named "stratum" (with all of its elements equal to the stratum name specified in design) is added to sf.object. The default is NULL.
mdcaty	Character string containing the name of the attribute in sf.object that identifies the unequal probability category for each feature. The default is NULL.
maxtry	This argument is depricated.
shapefile	Option to create a shapefile containing the survey design information, where TRUE equals create a shapefile and FALSE equals do not create a shapefile. The default is TRUE.
prjfilename	This argument is depricated.
out.shape	Name of the output shapefile. The default is "sample.shp".

Value

An object of class SpatialDesign containing the survey design information and any additional attribute variables that were provided. Optionally, a shapefile can be created that contains the survey design information.

Other Functions Required

irsarea select an IRS sample of an area resource

irslin select an IRS sample of a linear resource

irspts select an IRS sample of a finite resource

mdmarea calculate multidensity-density multipliers for an area resource

mdmlin calculate multidensity-density multipliers for a linear resource

mdmpts calculate multidensity-density multipliers for a finite resource

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
## Not run:
  test_design <- list(
    Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10),
    Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal",
      caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
  test_sfobject <- st_read("test_shapefile.shp")
  test_sample <- irs(design=test_design, DesignID="TestSite",
    type.frame="area", src.frame = "sf.object", sf.object=test_sfobject,
    stratum="test_stratum", mdcaty="test_mdcaty")

## End(Not run)
```

irsarea *Select an Independent Random Sample (IRS) of an Area Resource*

Description

This function selects an IRS of an area resource.

Usage

```
irsarea(areaframe, samplesize = 100, SiteBegin = 1)
```

Arguments

areaframe	The sf object containing attributes: id, mdcaty, area_mdm, and mdm.
samplesize	Number of points to select in the sample. The default is 100.
SiteBegin	First number to start siteID numbering. The default is 1.

Value

An sf object of sample points containing attributes: siteID, id, mdcaty, and wgt.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

irslin *Select an Independent Random Sample (IRS) of a Linear Resource*

Description

This function selects an IRS of a linear resource.

Usage

```
irslin(linframe, samplesize = 100, SiteBegin = 1)
```

Arguments

linframe	The sf object containing attributes: id, mdcaty, length_mdm, and mdm.
samplesize	Number of points to select in the sample. The default is 100.
SiteBegin	First number to start siteID numbering. The default is 1.

Value

An sf object of sample points containing attributes: siteID, id, mdcaty, and wgt.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

irspts *Select an Independent Random Sample (IRS) of a Finite Resource*

Description

This function selects an IRS of a finite resource (discrete points).

Usage

```
irspts(ptsframe, samplesize = 100, SiteBegin = 1)
```

Arguments

ptsframe	The sf object containing attributes: id, mdcaty, and mdm.
samplesize	Number of points to select in the sample. The default is 100.
SiteBegin	First number to start siteID numbering. The default is 1.

Value

An sf object of sample points containing attributes: siteID, id, mdcaty, and wgt.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

isotonic *Internal Function: Isotonic Regression*

Description

This function performs isotonic regression of an input set of values so that the output set of values is a nondecreasing sequence. The output set of values is truncated to the range: [minval, maxval].

Usage

```
isotonic(y, minval, maxval)
```

Arguments

y	Vector of the set of values on which to perform isotonic regression.
minval	Minimum value for the output set of values.
maxval	Maximum value for the output set of values.

Value

The revised set of input values

Other Functions Required: sorted - determine whether a set of values is a nondecreasing sequence

Other Functions Required

`sorted` determines whether a set of values is a nondecreasing sequence

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

localmean.cov *Internal Function: Variance-Covariance Matrix Based on Local Mean Estimator*

Description

This function calculates the variance-covariance matrix using the local mean estimator.

Usage

```
localmean.cov(zmat, weight.lst)
```

Arguments

zmat	Matrix of weighted response values or weighted residual values for the sample points.
weight.lst	List from the local mean weight function containing two elements: a matrix named ij composed of the index values of neighboring points and a vector named gwt composed of weights.

Value

The local mean estimator of the variance-covariance matrix.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

localmean.df *Internal Function: Degrees of Freedom for Local Mean Variance-Covariance Estimator*

Description

This function calculates the degrees of freedom of the local mean variance- covariance estimator.

Usage

```
localmean.df(weight.lst)
```

Arguments

weight.lst	List from the local mean weight function containing two elements: a matrix named ij composed of the index values of neighboring points and a vector named gwt composed of weights.

Value

The degrees of freedom of the local mean variance-covariance estimator.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

localmean.var *Internal Function: Local Mean Variance Estimator*

Description

This function calculates the local mean variance estimator.

Usage

```
localmean.var(z, weight.lst)
```

Arguments

z	Vector of weighted response values or weighted residual values for the sample points.
weight.lst	List from the local mean weight function containing two elements: a matrix named ij composed of the index values of neighboring points and a vector named gwt composed of weights.

Value

The local mean estimator of the variance.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`localmean.weight` *Internal Function: Local Mean Variance Neighbors and Weights*

Description

This function calculates the index values of neighboring points and associated weights required by the local mean variance estimator.

Usage

```
localmean.weight(x, y, prb, nbh = 4, vincr = 1e-05 * abs(mean(y)))
```

Arguments

x	Vector of x-coordinates for location of the sample points.
y	Vector of y-coordinates for location of the sample points.
prb	Vector of inclusion probabilities for the sample points.
nbh	Number of neighboring points to use in the calculations.
vincr	The variance increment for correcting an La.svd error. The default is 0.00001*abs(mean(y)).

Value

List containing two elements: a matrix named ij composed of the index values of neighboring points and a vector named gwt composed of weights.

Author(s)

Don Stevens <Kincaid.Tom@epa.gov>

See Also

localmean.weight2

localmean.weight2 *Internal Function: Recovery from a Singular Value Decomposition Error*

Description

This function calculates the initial section of the localmean.weight function and serves to allow recovery from an error in the singular value decomposition function (La.svd) that is called by the generalized inverse function (ginv) in the MASS package.

Usage

```
localmean.weight2(x, y, prb, nbh)
```

Arguments

x	Vector of x-coordinates for location of the sample points.
y	Vector of y-coordinates for location of the sample points.
prb	Vector of inclusion probabilities for the sample points.
nbh	Number of neighboring points to use in the calculations.

Value

Either an object of class "try-error" when the ginv function terminates with an error or a generalized inverse matrix when the ginv function terminates normally.

Author(s)

Don Stevens Tom Kincaid <Kincaid.Tom@epa.gov>

Luck_Ash_streams *Streams in the Luckiamute Watershed in Oregon*

Description

A dataset containing attributes for streams in the Luckiamute Watershed in Oregon.

Usage

```
Luck_Ash_streams
```

Format

An object of class sf (simple features) containing 429 features and 3 attributes:

Per_Int stream type for the line segment.

Level3_Nam Strahler order category for the line segment.

Length_km length of the line segment in kilometers.

`make_grid` *Create the Grid for a GRTS Survey Design*

Description

Creates the grid for a GRTS suvey design.

Usage

```
make_grid(xc, yc, dx, dy, sfobject)
```

Arguments

`xc`	x-coordinates that define the cells.
`yc`	y-coordinates that define the cells.
`dx`	The x-axis grid cell dimension.
`dy`	The y-axis grid cell dimension.
`sfobject`	the sf object containing the survey frame.

Value

An sf object containing the grid.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

marinus *Marinus Cylindrical Map Projection*

Description

This function converts x,y coordinates measured in units of latitude and longitude, i.e., geographic coordinates measured in decimal degrees, to coordinates in the equidistant, cylindric map projection measured in units of kilometers. The projection center is defined as the midpoint in latitude-longitude space. The map projection is here named after Marinus of Tyre (see J.P. Snyder. USGS Prof Paper 1395, p. 90).

Usage

```
marinus(lat, lon)
```

Arguments

lat	Vector of latitudes.
lon	Vector of longitudes.

Value

Matrix with column names "x" and "y" containing the x and y coordinates in the equidistant, cylindric map projection measured in units of kilometers.

Author(s)

Denis White

Examples

```
lat <- 45 + runif(100, -5, 5)
lon <- 120 + runif(100, -10, 10)
marinus(lat, lon)
```

mdmarea *Internal Function: GRTS Multipliers for Multi-Density Categories for Areas*

Description

GRTS Multiplieers for Multi-density Categories for areas

Usage

```
mdmarea(area, mdcaty, n.desired)
```

Arguments

area	Vector of polygon areas for each polygon in the sample frame.
mdcaty	Vector of multi-density category names for each polygon in the sample frame.
n.desired	Expected sample size for each category. Row names must match category names in mdcaty.

Value

Numeric vector of multipliers that is same length as area and mdcaty.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

mdmlin *Internal Function: GRTS Multipliers for Multi-Density Categories for Linear Network*

Description

Internal Function: GRTS Multipliers for Multi-Density Categories for Linear Network

Usage

```
mdmlin(len, mdcaty, n.desired)
```

Arguments

len	Vector of segment lengths for each segment in sample frame.
mdcaty	Vector of multi-density category groups for each segment in sample frame.
n.desired	Expected sample size for each category. Row names must match category names in mdcaty.

Value

Numeric vector of multipliers that is same length as len and mdcaty.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

mdmpts — *Internal Function: GRTS Multipliers for Multi-Density Categories for Points*

Description

Internal Function: GRTS Multipliers for Multi-Density Categories for Points

Usage

```
mdmpts(mdcaty, n.desired)
```

Arguments

mdcaty	Vector of multi-density category groups for each element in sample frame.
n.desired	Expected sample size for each category. Row names must match category names in mdcaty.

Value

Numeric vector of multipliers that is same length as mdcaty.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

NE_lakes — *Lakes in the Southern New England Region of the U.S.*

Description

A dataset containing attributes for Lakes in the Southern New England Region of the U.S.

A simple features (sf) point data.frame of 6121 lakes that was created from a shapefile of lakes in the southern New England region of the U.S. containing the area category in hectares and coordinates. The variables are as follows:

Usage

```
NE_lakes
```

```
NE_lakes
```

Format

An object of class sf (simple features) containing 6,121 features and 4 attributes:

xcoord Albers projection x-coordinate of the lake centroid.

ycoord Albers projection y-coordinate of the lake centroid.

State state code for the lake.

Area_Cat lake surface area category in hectares.

NLA_2007 *Western Mountains Ecoregion Lakes*

Description

A dataset containing attributes for Western Mountains Ecoregion lakes sampled by the U.S. Environmental Protection Agency (EPA) during the National Lakes Assessment (NLA) survey for 2007.

Usage

```
NLA_2007
```

Format

A data frame with 236 rows and 15 attributes:

siteID site ID value.

xcoord Albers projection x-coordinate.

ycoord Albers projection y-coordinate.

wgt survey design weight.

Lake_Origin lake origin category.

Chla chlorophyll-a concentration.

OE5 value of the index of macroinvertebrate taxa loss.

PTL total phosphorus concentration.

NTL total nitrogen concentration.

Turbidity turbidity value.

Chla_cond condition class category ("Good", "Fair", or "Poor") of the chlorophyll-a value.

OE5_cond condition class category of the macroinvertebrate taxa loss index.

PTL_cond condition class category of the total phosphorus value.

NTL_cond condition class category of the total nitrogen value.

Turbidity_cond condition class category of the turbidity value.

NRSA_2009 *Western Mountains Ecoregion Rivers and Streams*

Description

A dataset containing attributes for Western Mountains Ecoregion rivers and streams sampled by the U.S. Environmental Protection Agency (EPA) during the National Rivers and Streams Assessment (NRSA) surveys for 2004-2006 and 2008-2009.

Usage

```
NRSA_2009
```

Format

A data frame with 668 rows and 13 attributes:

siteID site ID value.

xcoord Albers projection x-coordinate.

ycoord Albers projection y-coordinate.

wgt survey design weight.

Survey survey identifier, which is either WSA or NRSA.

Revisit_Site identifier of revisit sites for the two surveys, where Y = a revisit site and N = not a revisit site.

Stream_Size stream size category, which is either large or small.

NTL total nitrogen concentration.

PTL total phosphorus concentration.

Benthic_MMI value of the benthic macroinvertebrate multimetric index (MMI).

NTL_Cond condition class category of the total nitrogen value.

PTL_Cond condition class category of the total phosphorus value.

Benthic_MMI_Cond condition class category of the benthic MMI value.

numLevels *Determine the Number of Levels for Hierarchical Randomization for a Generalized Random-Tesselation Stratified (GRTS) Survey Design*

Description

This function determine the bumber of levels of hierarchical randomization for a GRTS survey design.

Usage

```
numLevels(samplesize, shift.grid, startlev, maxlev, sfobject)
```

Arguments

samplesize	The desired smaple size.
shift.grid	Logical value indicating whether the GRTS grid should be randomly shifted.
startlev	The initial number of levels for the GRTS grid.
maxlev	The maximum number of levels for the GRTS grid.
sfobject	The sf object containing the survey frame.

Value

A list containing the number of levels, x-coordinates, y-coordinates, x-axis grid cell dimension, y-axis grid cell dimension, cell total weights, and sampling interval.

Other Functions Required

cellWeight calculates total inclusion probability for each cell in a grid

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

panel_summary *Summary Characteristics of a Revisit Panel Design*

Description

Revisit panel design characteristics are summarized: number of panels, number of time periods, total number of sample events for the revisit design, total number of sample events for each panel, total number of sample events for each time period and cumulative number of unique units sampled by time periods.

Usage

```
panel_summary(paneldsgn, visitdsgn = NULL)
```

Arguments

paneldsgn	Two-dimensional array with dimnames specifying revisit panel design. Typically, array is output from revisit_dsgn, revisit_bibd or revisit_rand functions.
visitdsgn	Two-dimensional array with same dimensions as paneldsgn specifying the number of times a sample unit is sampled at each time period. Default is visitdsgn=NULL, where default assumes that a sample unit will be sampled only once at each time period.

Details

The revisit panel design and the visit design (if present) are summarized. Summaries can be useful to know the effort required to complete the survey design. See the values returned for the summaries that are produced.

Value

List of six elements.

- n.panel - number of panels in revisit design
- n.period - number of time periods in revisit design
- n.total - total number of sample events across all panels and all time periods, accounting for visitdsgn, that will be sampled in the revisit design
- n.periodunit - Vector of the number of time periods a unit will be sampled in each panel
- n.unitpnl - Vector of the number of sample units, accounting for visitdsgn, that will be sampled in each panel
- n.unitperiod - Vector of the number of sample units, accounting for visitdsgn, that will be sampled during each time period
- ncum.unit - Vector of the cumulative number of unique units that will be sampled in time periods up to and including the current time period.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

See Also

revisit_dsgn create a panel revisit design

revisit_bibd create a balanced incomplete block panel revisit design

revisit_rand create a revisit design with random assignment to panels and time periods

power.dsgn power calculation for multiple panel designs

cov.panel.dsgn covariance matrix for a panel design

plot_powerpaneldesign plot power curves for panel designs

Examples

```
# Serially alternating panel revisit design summary
sa.dsgn <- revisit_dsgn(20, panels=list(SA60N=list(n=60, pnl_dsgn = c(1, 4),
                        pnl_n=NA, start_option="None")), begin=1 )
panel_summary(sa.dsgn)

# Add visit design where first panel is sampled twice at every time period
sa.visit <- sa.dsgn
sa.visit [sa.visit > 0] <- 1
sa.visit [1, sa.visit[1,] > 0] <- 2
panel_summary(sa.dsgn, sa.visit)
```

pickFiniteSamplePoints *Select Sample Points from a Simple Features Point Object*

Description

This function selects sample points from an sf object of geometry type point.

Usage

```
pickFiniteSamplePoints(rdx, xc, yc, dx, dy, sfobject)
```

Arguments

rdx	Vector of cell IDs.
xc	Vector of x-coordinates for the grid cells.
yc	Vector of y-coordinates for the grid cells.
dx	The x-axis grid cell dimension.
dy	The y-axis grid cell dimension.
sfobject	The sf point object.

Value

Vector containing feature IDs of the selected sample points.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

pickGridCells *Identify Grid Cells from which Sample Points Will Be Selected*

Description

This function identifies grid cells from which sample points will be selected.

Usage

```
pickGridCells(samplesize, idx)
```

Arguments

samplesize	The desired smaple size.
idx	Vector of values for identifying cell IDs.

Value

Vector of grid cells IDs.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

pickSamplePoints *Select Sample Points from a Simple Features Object*

Description

This function selects sample points from an sf object of geometry type polygon or linestring.

Usage

```
pickSamplePoints(sfobject, featureID, xc, yc, dx, dy)
```

Arguments

sfobject	The sf polygon object.
featureID	Vector identifying features in sfobject from which a sample point will be selected.
xc	Vector of x-coordinates for the grid cells.
yc	Vector of y-coordinates for the grid cells.
dx	The x-axis grid cell dimension.
dy	The y-axis grid cell dimension.

Value

Data frame containing x-coordinates and y-coordinates for sample points.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`plot_powerpaneldesign` *Plot Power Curves for Panel Designs*

Description

Plot power curves and relative power curves for trend detection for set of panel designs, time periods, indicators, signifance levels and trend. Trend may be based on percent change per period in mean or percent change in proportion of cumulative distribution function above or below a fixed cut point. Types of plots are combinations of standard/relative, mean/percent, period/change and design/indicator. Input must be be of class powerpaneldesign and is normally the output of function power.dsgn.

Usage

```
plot_powerpaneldesign(dsgnpower, plot.type = "standard",
  trend.type = "mean", xaxis.type = "period", comp.type = "design",
  dsgns = NULL, indicator = NULL, trend = NULL, period = NULL,
  alpha = NULL)
```

Arguments

`dsgnpower`	List object of class powerpaneldesign. Object provides power calculated for a set of panel designs, set of indicators, set of trend values, and set of alpha values. Expect input as list as output from function power.dsgn.
`plot.type`	Default is "standard" which plots standard power curve. If equal to "relative", then plot power of one panel design compared to one or more other panel designs.
`trend.type`	Character value for trend in mean ("mean") or or percent change in proportion ("percent") of cumulative distribution function above or below a fixed cut point. Default is "mean".
`xaxis.type`	Character value equal to "period" or "change" which designates the type of x-axis for power plot where power is plotted on y-axis. For xaxis.type = "period", x-axis is periods in dsgnpower. If xaxis.type = "change", then x-axis is percent per period with secondary x-axises for total percent per period and associated change in mean. Default is "period". Note that xaxis.type controls how the input for "period" and "trend" paramenters is used.
`comp.type`	Character value equal to "design" or "Indicator" which designates the type of power curve comparison that will occur on a single plot. If comp.type = "design", then on a single plot of power curves all panel designs specified in "dsgns" are plotted for a single indicator, single trend value and single alpha. If comp.type = "indicator", then on a single plot of power curves all indicators specified in "indicator" are plotted for a single panel design, single trend value and single alpha. Default is "design".
`dsgns`	Vector of names of panel designs that are to be plotted. Names must be all, or a subset of, names of designs in dsgnpower. Default is NULL which results in only the first panel design in dsgnpower being used.

`indicator` Vector of indicator names contained in dsgnpower that are to be plotted. Indicator names must be all, or a subset of, indicator names in dsgnpower. Default is NULL which results in only the first indicator in dsgnpower being used.

`trend` NULL, a single value or vector of values contained in dsgnpower that will be plotted. Values must be all, or a subset of, trend values in dsgnpower. If xaxis.type is equal to "period", then NULL results in maximum trend value being used and a single value or vector of values results in a separate plot for each value specified. If xaxis.type is equal to "change", then NULL results in all trend values in dsgnpower being plotted on x-axis and a vector of values results in all trend values in dsgnpower from minimum value to maximum value specified being plotted on x-axis.

`period` NULL, a single value or vector of values contained in dsgnpower that will be plotted. Values must be all, or a subset of, period values in dsgnpower. If xaxis.type is equal to "period", then NULL results in all time periods in dsgnpower being plotted on x-axis and a vector of values results in all period values in dsgnpower from minimum value to maximum value specified being plotted on x-axis. If xaxis.type is equal to "change", then NULL results in all time periods in dsgnpower being plotted in separate plots and a vector of values results in time periods specified being plotted in separate plots.

`alpha` A single value or vector of significance levels (as proportion, e.g. 0.05) contained in dsgnpower to used for power plots. Specifying more than a single value results in multiple plots. Default is NULL which results in the minimum significance level in dsgnpower being used.

Details

By default the plot function produces a standard power curve at end of each time period on the x-axis with y-axis as power. When more than one panel design is in dsgnpower, the first panel design is used When more than one indicatoris in dsgnpower, the first indicator is used When more than one trend value is in dsgnpower, the maximum trend value is used. When more than one significance level, alpha, is in dsgnpower, the minimum significance level is used.

Control of the type of plot produced is governed by plot.type, trend.type, xaxis.type and comp.type. The number of plots produced is governed by the number of panel designs (dsgn) specified, the number of indicators (indicator) specified, the number of time periods (period) specifice, the number of trend values (trend) specified and the number of significance levels (alpha) specified.

When the comparison type ("comp.type") is equal to "design", all power curves specified by dsgn are plotted on the same plot. When comp.type is equal to "indicator", all power curves specified by "indicator" are plotted on the same plot. Typically, no more than 4-5 power curves should be plotted on same plot.

Value

One or more power curve plots are created and plotted. User must specify output graphical device if more than one plot is created. See Devices for graphical output options.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

See Also

revisit_dsgn create a panel revisit design

revisit_bibd create a balanced incomplete block panel revisit design

revisit_rand create a revisit design with random assignment to panels and time periods

panel_summary summarize characteristics of a revisit panel design

power.dsgn power calculation for multiple panel designs

cov.panel.dsgn covariance matrix for a panel design

Examples

```
# Construct a rotating panel design with sample size of 60
R60N <- revisit_dsgn (20, panels=list(R60N=list(n=60, pnl_dsgn = c(1, NA),
                      pnl_n=NA, start_option="None")), begin=1 )

# Construct a fixed panel design with sample size of 60
F60 <- revisit_dsgn (20, panels=list(F60=list(n=60, pnl_dsgn = c(1, 0),
                     pnl_n=NA, start_option="None")), begin=1 )

# Power for rotating panel with sample size 60
Power.tst <- power.dsgn("Variable_Name", ind.values = 43, unit.var = 280,
                        period.var = 4, unitperiod.var = 40, index.var = 90,
                        unit.rho = 1, period.rho = 0, paneldsgn = list(
                          R60N=R60N, F60=F60), nrepeats = NULL,
                        trend.type = "mean", trend= c(1.0, 2.0), alpha=0.05 )
plot_powerpaneldesign(Power.tst)
plot_powerpaneldesign(Power.tst, dsgns = c("F60", "R60N"))
plot_powerpaneldesign(Power.tst, dsgns = c("F60", "R60N"), trend = 1.0)
## Not run:
pdf("Power.tst.pdf")
plot_powerpaneldesign(Power.tst, plot.type = "relative", comp.type = "design",
     trend.type = "mean", trend = c(1, 2), dsgns = c("R60N", "F60"),
     indicator="Variable_Name")
graphics.off()

## End(Not run)
```

---

power.dsgn *Power Calculation for Multiple Panel Designs*

Description

Calculates the power for trend detection for one or more variables, for one or more panel designs, for one or more linear trends, and for one or more signficance levels. The panel designs create a covarance model where the model includes variance components for units, periods, the interaction of units and periods, and the residual (or index) variance.

Usage

```
power.dsgn(ind.names, ind.values, unit.var, period.var, unitperiod.var,
  index.var, unit.rho = 1, period.rho = 0, paneldsgn,
  nrepeats = NULL, trend.type = "mean", ind.pct = NULL,
  ind.tail = NULL, trend = 2, alpha = 0.05)
```

Arguments

ind.names	Vector of indicator names
ind.values	Vector of indicator mean values
unit.var	Vector of variance component estimates for unit variability for the indicators
period.var	Vector of variance component estimates for period variability for the indicators
unitperiod.var	Vector of variance component estimates for unit by period interaction variability for the indicators
index.var	Vector of variance component estimates for index (residual) error for the indicators
unit.rho	Correlation across units. Default is 1
period.rho	Correlation across periods. Default is 0
paneldsgn	A list of panel designs each as a matrix. Each element of the list is a matrix with dimnames (dimensions: number of panels (rows) by number of periods (columns)) containing the number of units visited for each combination of panel and period. Dimnames for columns must be convertable to an integer (e.g., 2016). All designs must span the same number of periods. Typically, the panel designs are the output of the function revisit_dsgn.
nrepeats	Either NULL or a list of matrices the same length as paneldsgn specifying the number of revisits made to units in a panel in the same period for each design. Specifying NULL indicates that number of revisits to units is the same for all panels and for all periods and for all panel designs. The default is NULL, a single visit. Names must match list names in paneldsgn.
trend.type	Trend type is either "mean" where trend is applied as percent trend in the indicator mean or "percent" where the trend is applied as percent trend in the proportion (percent) of the distribution that is below or above a fixed value. Default is trend.type="mean"
ind.pct	When trend.type is equal to "percent", a vector of the values of the indicator fixed value that defines the percent. Default is NULL
ind.tail	When trend.type is equal to "percent", a character vector with values of either "lower" or "upper" for each indicator. "lower" states that the percent is associated with the lower tail of the distribution and "upper" states that the percent is associated with the upper tail of the distribution. Default is NULL.
trend	Single value or vector of assumed percent change from initial value in the indicator for each period. Assumes the trend is expressed as percent per period. Note that the trend may be either positive or negative. The default is trend=2.
alpha	Single value or vector of significance level for linear trend test, alpha, Type I error, level. The defualt is 0.05.

Details

Calculates the power for detecting a change in the mean for different panel design structures. The model incorporates unit, period, unit by period, and index variance components as well as correlation across units and across periods. See references for methods.

Value

A list with components trend.type, ind.pct, ind.tail, trend values across periods, periods (all periods included in one or more panel designs), significance levels, a five-dimensional array of power calculations (dimensions: panel design names, periods, indicator names, trend names, alpha.names), an array of indicator mean values for each trend and the function call.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

References

Urquhart, N. S., W. S. Overton, et al. (1993) Comparing sampling designs for monitoring ecological status and trends: impact of temporal patterns. In: *Statistics for the Environment.* V. Barnett and K. F. Turkman. John Wiley & Sons, New York, pp. 71-86.

Urquhart, N. S. and T. M. Kincaid (1999). Designs for detecting trends from repeated surveys of ecological resources. *Journal of Agricultural, Biological, and Environmental Statistics*, **4(4)**, 404-414.

Urquhart, N. S. (2012). The role of monitoring design in detecting trend in long-term ecological monitoring studies. In: *Design and Analysis of Long-term Ecological Monitoring Studies.* R. A. Gitzen, J. J. Millspaugh, A. B. Cooper, and D. S. Licht (eds.). Cambridge University Press, New York, pp. 151-173.

See Also

revisit_dsgn create a panel revisit design

revisit_bibd create a balanced incomplete block panel revisit design

revisit_rand create a revisit design with random assignment to panels and time periods

panel_summary summarize characteristics of a revisit panel design

cov.panel.dsgn covariance matrix for a panel design

plot_powerpaneldesign plot power curves for panel designs

Examples

```
# Power for rotating panel with sample size 60
power.dsgn("Variable_Name", ind.values = 43, unit.var = 280, period.var = 4,
           unitperiod.var = 40, index.var = 90, unit.rho = 1, period.rho = 0,
           paneldsgn = list(NoR60=revisit_dsgn(20,
                            panels=list(NoR60=list(n=60, pnl_dsgn = c(1, NA),
```

```
                    pnl_n = NA, start_option = "None")), begin = 1)),
nrepeats = NULL, trend.type = "mean", trend= 1.0, alpha=0.05)
```

---

ranho *Construct Randomized Hierarchical Addresses for a Generalized Random-Tesselation Stratified (GRTS) Survey Design*

Description

This function constructs randomized hierarchical addresses for a GRTS survey design.

Usage

```
ranho(hadr)
```

Arguments

hadr	Vector hierarchical addresses.

Value

Vector of randomized hierarchical addresses.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

read.dbf *Read the dbf File of an ESRI Shapefile*

Description

This function reads the dbf file of an ESRI shapefile and creates a data frame.

Usage

```
read.dbf(filename)
```

Arguments

filename	Character string containing the name of the shapefile.

Value

Data frame containing contents of the dbf file.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Marc Weber <Weber.Marc@epa.gov>

`read.sas` *Read a SAS dataset or a SAS XPORT File*

Description

This function reads either a SAS dataset or a SAS XPORT (transport) file and creates a data frame.

Usage

```
read.sas(filename, libname = NULL, xport = FALSE,
  sascmd = "C:/Program Files/SASHome/SASFoundation/9.4/sas.exe")
```

Arguments

`filename`	If xport equals TRUE, a character string giving the full path to the SAS XPORT file, which must include the file extension. If xport equals FALSE, either a character string giving the the name of a dataset in the SAS library or a vector of character strings giving the names of datasets in the SAS library, where the dataset names cannot exceed eight characters in length and do not include the file extension.
`libname`	Character string defining the SAS library, which is usually a directory reference. If xport equals FALSE and the dataset(s) named in argument filename do not reside in the working directory, then this argument is required. The default value is NULL.
`xport`	Logical value indicating whether the input file is a SAS XPORT file. The default value is FALSE.
`sascmd`	Character string giving the full path to SAS executable. This argument is required only when xport equals FALSE. The default value is "C:/Program Files/SAS/SAS 9.1/sas.exe".

Value

Either a single data frame or a list of data frames.

Other Functions Required

`read.ssd` function in the foreign package that reads a SAS dataset and creates a data frame

`read.xport` function in the foreign package that reads a SAS XPORT file and creates a data frame

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
## Not run:
MySasFile <- read.sas("mysasfil", "C:/Documents and Settings/auser/My Project")

## End(Not run)
```

`read.shape` *Read an ESRI Shapefile*

Description

This function reads an ESRI shapefile and creates a simple features (sf) object.

Usage

```
read.shape(filename)
```

Arguments

`filename`	Character string containing the name of the shapefile. The shapefile name should include the ".shp" extension. If the name does not include that extension, it will be added.

Value

An object belonging to class sf.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Marc Weber <Weber.Marc@epa.gov>

`relrisk.analysis` *Relative Risk Analysis for Probability Survey Data*

Description

This function organizes input and output for relative risk analysis of categorical data generated by a probability survey.

Usage

```
relrisk.analysis(sites = NULL, subpop = NULL, design, data.rr,
  response.var, stressor.var, response.levels = rep(list(c("Poor",
  "Good")), length(response.var)), stressor.levels = rep(list(c("Poor",
  "Good")), length(stressor.var)), popcorrect = FALSE, pcfsize = NULL,
  N.cluster = NULL, stage1size = NULL, sizeweight = FALSE,
  vartype = "Local", conf = 95)
```

Arguments

`sites` Data frame consisting of two variables: the first variable is site IDs, and the second variable is a logical vector indicating which sites to use in the analysis. The default is NULL.

`subpop` Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is site IDs. Each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. The default is NULL.

`design` Data frame consisting of design variables. If spsurvey.obj is not provided, then this argument is required. The default is NULL. Variables should be named as follows:

siteID Vector of site IDs

wgt Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample

xcoord Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample

ycoord Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample

stratum Vector of the stratum codes for each site

cluster Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site

wgt1 Vector of stage one weights in a two-stage design

xcoord1 Vector of the stage one x-coordinates for location in a two-stage design

ycoord1 Vector of the stage one y-coordinates for location in a two-stage design

support Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors.

swgt Vector of size-weights, which is the stage two size-weight for a two-stage design.

swgt1 Vector of stage one size-weights for a two-stage design.

`data.rr` Data frame of categorical response and stressor variables, where each variable consists of two categories. If response or stressor variables include more than

two categories, occurrences of those categories must be removed or replaced with missing values. The first column of this argument is site IDs. Subsequent columns are response and stressor variables. Missing data (NA) is allowed.

`response.var` Character vector providing names of columns in argument data.rr that contain a response variable, where names may be repeated. Each name in this argument is matched with the corresponding value in the stressor.var argument.

`stressor.var` Character vector providing names of columns in argument data.rr that contain a stressor variable, where names may be repeated. Each name in this argument is matched with the corresponding value in the response.var argument. This argument must be the same length as argument response.var.

`response.levels`

List providing the category values (levels) for each element in the response.var argument. This argument must be the same length as argument response.var. The first level for each element in the list is used for calculating the numerator and the denominator of the relative risk estimate. The default is a list containing the values "Poor" and "Good" for the first and second levels, respectively, of each element in the response.var argument.

`stressor.levels`

List providing the category values (levels) for each element in the stressor.var argument. This argument must be the same length as argument response.var. The first level for each element in the list is used for calculating the numerator of the relative risk estimate, and the second level for each element in the list is used for calculating the denominator of the estimate. The default is a list containing the values "Poor" and "Good" for the first and second levels, respectively, of each element in the stressor.var argument.

`popcorrect` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster and stage1size, and for the support variable of the design argument.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify

	both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.
sizeweight	Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
conf	Numeric value for the confidence level. The default is 95.

Value

Data frame of relative risk estimates for all combinations of population Types, subpopulations within Types, and response variables. Standard error and confidence interval estimates also are provided.

Other Functions Required

dframe.check check site IDs, the sites data frame, the subpop data frame, and the data.rr data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

vecprint takes an input vector and outputs a character string with line breaks inserted

uniqueID creates unique site IDs by appending a unique number to each occurrence of a site ID

input.check check input values for errors, consistency, and compatibility with analytical functions

relrisk.est compute the relative risk estimate

Author(s)

Tom Kincaid <Kincaid.Tom@epa.Gov>

Examples

```
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(TRUE, 100))
mysubpop <- data.frame(
  siteID=mysiteID,
  All.Sites=rep("All Sites", 100),
  Resource.Class=rep(c("Agr", "Forest"), c(55,45)))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1",
  "Stratum2"), 50))
mydata.rr <- data.frame(
```

```
  siteID=mysiteID,
  RespVar1=sample(c("Poor", "Good"), 100, replace=TRUE),
  RespVar2=sample(c("Poor", "Good"), 100, replace=TRUE),
  StressVar=sample(c("Poor", "Good"), 100, replace=TRUE))
relrisk.analysis(sites=mysites, subpop=mysubpop, design=mydesign,
  data.rr=mydata.rr, response.var=c("RespVar1", "RespVar2"),
  stressor.var=rep("StressVar", 2))
```

`relrisk.est` *Relative Risk Estimate for 2x2 Table*

Description

This function calculates the relative risk estimate for a 2x2 table of cell counts defined by a categorical response variable and a categorical explanatory (stressor) variable for an unequal probability design. Relative risk is the ratio of two probabilities: the numerator is the probability that the first level of the response variable is observed given occurrence of the first level of the stressor variable, and the denominator is the probability that the first level of the response variable is observed given occurrence of the second level of the stressor variable. The numerator probability and denominator probability are estimated using cell and marginal totals from a 2x2 table of cell counts defined by a categorical response variable and a categorical stressor variable. An estimate of the numerator probability is provided by the ratio of the cell total defined by the first level of response variable and the first level of the stressor variable to the marginal total for the first level of the stressor variable. An estimate of the denominator probability is provided by the ratio of the cell total defined by the first level of response variable and the second level of the stressor variable to the marginal total for the second level of the stressor variable. Cell and marginal totals are estimated using the Horvitz-Thompson estimator. The standard error of the log of the relative risk estimate and confidence limits for the estimate also are calculated. The standard error is calculated using a first-order Taylor series linearization (Sarndal et al., 1992).

Usage

```
relrisk.est(response, stressor, response.levels = c("Poor", "Good"),
  stressor.levels = c("Poor", "Good"), wgt, xcoord = NULL,
  ycoord = NULL, stratum = NULL, cluster = NULL, wgt1 = NULL,
  xcoord1 = NULL, ycoord1 = NULL, popcorrect = FALSE,
  pcfsize = NULL, N.cluster = NULL, stage1size = NULL,
  support = NULL, sizeweight = FALSE, swgt = NULL, swgt1 = NULL,
  vartype = "Local", conf = 95, check.ind = TRUE, warn.ind = NULL,
  warn.df = NULL, warn.vec = NULL)
```

Arguments

`response`	Vector of the categorical response variable values.
`stressor`	Vector of the categorical explanatory (stressor) variable values.

response.levels
: Vector of category values (levels) for the categorical response variable, where the first level is used for calculating the numerator and the denominator of the relative risk estimate. If response.levels is not supplied, then values "Poor" and "Good" are used for the first level and second level of the response variable, respectively. The default is c("Poor", "Good").

stressor.levels
: Vector of category values (levels) for the categorical stressor variable, where the first level is used for calculating the numerator of the relative risk estimate and the second level is used for calculating the denominator of the estimate. If stressor.levels is not supplied, then values "Poor" and "Good" are used for the first level and second level of the stressor variable, respectively. The default is c("Poor", "Good").

wgt
: Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.

xcoord
: Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.

ycoord
: Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.

stratum
: Vector of the stratum for each site. The default is NULL.

cluster
: Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site. The default is NULL.

wgt1
: Vector of the final adjusted stage one weight for each site. The default is NULL.

xcoord1
: Vector of the stage one x-coordinate for location for each site. The default is NULL.

ycoord1
: Vector of the stage one y-coordinate for location for each site. The default is NULL.

popcorrect
: Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

pcfsize
: Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

N.cluster
: The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.
sizeweight	Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.
swgt	Vector of the size-weight for each site, which is the stage two size-weight for two-stage sample. The default is NULL.
swgt1	Vector of the stage one size-weight for each site. The default is NULL.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
conf	Numeric value for the confidence level. The default is 95.
check.ind	= a logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	Data frame for storing warning messages. The default is NULL.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called by the relrisk.analysis function, then output is an object in list format composed of the Results list, which contains estimates and confidence bounds, the warn.ind logical value, which indicates whether warning messages were generated, and the warn.df data frame, which contains warning messages. If the function was called directly, then output is the Results list, which contains the following components:

RelRisk the relative risk estimate

RRnum numerator ("elevated" risk) of the relative risk estimate

RRdenom denominator ("baseline" risk) of the relative risk estimate

RRlog.se standard error for the log of the relative risk estimate

ConfLimits confidence limits for the relative risk estimate

WeightTotal sum of the final adjusted weights

CellCounts cell and margin counts for the 2x2 table

CellProportions estimated cell proportions for the 2x2 table

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

relrisk.var calculate values required for estimating variance of the relative risk estimate

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
response <- sample(c("Poor", "Good"), 100, replace=TRUE)
stressor <- sample(c("Poor", "Good"), 100, replace=TRUE)
wgt <- runif(100, 10, 100)
relrisk.est(response, stressor, wgt=wgt, vartype="SRS")

xcoord <- runif(100)
ycoord <- runif(100)
relrisk.est(response, stressor, wgt=wgt, xcoord=xcoord, ycoord=ycoord)
```

---

relrisk.var *Variance-Covariance Estimate for the Relative Risk Estimator*

Description

This function calculates the variance-covariance estimate for the cell and marginal totals used to calculate the relative risk estimate. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples.

Usage

```
relrisk.var(response, stressor, response.levels, stressor.levels, wgt, x,
  y, stratum.ind, stratum.level, cluster.ind, cluster, wgt1, x1, y1,
  pcfactor.ind, pcfsize, N.cluster, stage1size, support, vartype, warn.ind,
  warn.df, warn.vec)
```

Arguments

response	Vector of the categorical response variable.
stressor	Vector of the categorical stressor variable.

`response.levels`
: Vector of category values (levels) for the categorical response variable, where the first level is used for calculating the relative risk estimate. If response.levels equals NULL, then values "Poor" and "Good" are used for the first level and second level of the response variable, respectively. The default is NULL.

`stressor.levels`
: Vector of category values (levels) for the categorical stressor variable, where the first level is used for calculating the numerator of the relative risk estimate and the second level is used for calculating the denominator of the estimate. If stressor.levels equals NULL, then values "Poor" and "Good" are used for the first level and second level of the stressor variable, respectively. The default is NULL.

`wgt`
: Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.

`x`
: Vector of x-coordinate for location for each site, which is either the x- coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.

`y`
: Vector of y-coordinate for location for each site, which is either the y- coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.

`stratum.ind`
: Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.

`stratum.level`
: The stratum level.

`cluster.ind`
: Logical value that indicates whether the sample is a two- stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.

`cluster`
: Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.

`wgt1`
: Vector of the final adjusted stage one weight for each site.

`x1`
: Vector of the stage one x-coordinate for location for each site.

`y1`
: Vector of the stage one y-coordinate for location for each site.

`pcfactor.ind`
: Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

`pcfsize`
: Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

`N.cluster`
: The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.

stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

Object in list format composed of a vector named varest, which contains the variance-covariance estimate, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.cov calculate the variance/covariance matrix using the local mean estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

revisit_bibd *Create a Balanced Incomplete Block Panel Revisit Design*

Description

Create a revisit design for panels in a survey that specifies the time periods for the units of each panel to be sampled based on searching for a D-optimal block design that is a member of the class of generalized Youden designs. The resulting design need not be a balanced incomplete block design. Based on algorithmic idea by Cook and Nachtsheim (1989) and implemented by Robert Wheeler.

Usage

```
revisit_bibd(n.period, n.pnl, n.visit, nsamp, panel_name = "BIB",
  begin = 1, skip = 1, iter = 30)
```

Arguments

n.period	Number of time periods for the survey design. Typically, number of periods if sampling occurs once per period or number of months if sampling occurs once per month. (v, number of varieties/treatments in BIBD terms)
n.pnl	Number of panels (b, number of blocks in BIBD terms)
n.visit	Number of time periods to be visited in a panel (k, block size in BIBD terms)
nsamp	Number of samples in each panel.
panel_name	Prefix for name of each panel
begin	Numeric name of first sampling occasion, e.g. a specific period.
skip	Number of sampling occasions to skip between planned sampling periods, e.g., sampling will occur only every 5 periods if skip = 5.
iter	Maximum number of iterations in search for D-optimal Generallized Youden Design.

Details

The function uses find.BIB function from crossdes package to search for a D-optimal block design. crossdes uses package AlgDesign to search balanced incomplete block designs.

Value

A two-dimensional array of sample sizes to be sampled for each panel and each sampling occasion.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

References

Cook R. D. and C. Nachtsheim. (1989). Computer-aided blocking of factorial and response-surface designs. *Technometrics* **31(3)**, 339-346.

See Also

revisit_dsgn create a panel revisit design

revisit_rand create a revisit design with random assignment to panels and time periods

panel_summary summarize characteristics of a revisit panel design

power.dsgn power calculation for multiple panel designs

cov.panel.dsgn covariance matrix for a panel design

plot_powerpaneldesign plot power curves for panel designs

Examples

```
# Balanced incomplete block design with 20 sample occasions, 20 panels,
# 3 visits to each unit, and 20 units in each panel.
revisit_bibd(n.period = 20, n.pnl = 20, n.visit = 3, nsamp = 20)
```

`revisit_dsgn` *Create a Panel Revisit Design*

Description

Create a revisit design for panels in a survey that specifies the time periods that members of each panel will be sampled. Three basic panel design structures may be created: always revisit panel, serially alternating panels, or rotating panels.

Usage

```
revisit_dsgn(n.period, panels, begin = 1, skip = 1)
```

Arguments

`n.period`	Number of time periods for the panel design. For example, number of periods if sampling occurs once per period or number of months if sampling occurs once per month.
`panels`	List of lists where each list specifies a revisit panel structure. Each sublist consists of four components: n - sample size for each panel in the sublist, pnl_dsgn - a vector with an even number of elements specifying the panel revisit schedule in terms of the number of consecutive time periods sample units will be sampled, followed by number of consecutive time periods skipped, and then repeated as necessary. pnl_n - number of panels in the sublist, and start_option - option for starting the pnl_dsgn (None, Partial_Begin, or Partial_End). Three basic panel structures are possible: a) if pnl_dsgn ends in 0, then the sample units are visited on all subsequent time periods, b) if pnl_dsgn ends in NA, then panel follows a rotating panel structure, and c) if pnl_dsgn ends in any number > 0, then panel follows a serially alternating panel structure. See details for further information.
`begin`	Numeric name of first sampling occasion, e.g. a specific period.
`skip`	Number of time periods to skip between planned sampling periods, e.g., sampling will occur only every 5 periods if skip = 5.

Details

The function creates revisit designs using the concepts in McDonald (2003) to specify the revisit pattern across time periods for each panel. The panel revisit schedule is specificed by a vector. Odd positions in vector specify the number of consecutive time periods when panel units are sampled. Even positions in vector specify the number of consecutive time periods when panel units are not sampled.

If last even position is a "0", then a single panel follows an always revisit panel structure. After satisfying the initial revisit schedule specified prior to the "0", units in a panel are always visited for rest of the time periods. The simplest always revisit panel design is to revisit every sample unit on every time period, specified as pnl_dsgn = c(1,0) or using McDonald's notation [1-0].

If the last even position is NA, the panels follow a rotating panel structure. For example, pnl_dsgn = c(1, NA) designates that sample units in a panel will be visited once and then never again, [1-n] in McDonald's notation. pnl_dsgn = c(1, 4, 1, NA) designates that sample units in a panel will be visited once, then not sampled on next four time periods, then sampled again once at the next time period and then never sampled again, [1-4-1-n] in McDonald/s notation.

If the last even position is > 0, the panels follow a serially alternating panel structure. For example, pnl_dsgn = c(1, 4) designates that sample units in a panel will be visited once, then not sampled during the next four time periods, then sampled once and not sampled for next four time periods, and that cycle repeated until end of the number of time periods, [1-4] in McDonald's notation. pnl_dsgn = c(2, 3, 1, 4) designates that the cycle has sample units in a panel being visited during two consecutive time periods and then not sampled on next four time periods, and the cycle is repeated until end of the number of time periods, [2-3-1-4] in McDonald's notation.

The number of panels in a single panel design is specified by pnl_n. For an always revisit panel structure, a single panel is created and pnl_n is ignored. For a rotating panel structure, when pnl_n = NA, the number of panels is equal to n.period. Note that this should only be used when the rotating panel structure is the only panel design, i.e., no split panel design (see below for split panel details). If pnl_n = m is specified for a rotating panel design, then then number of panels will be m. For example, pnl_dsgn = c(1, 4, 1, NA) and and pnl_n = 5 means that only 5 panels will be constructed and the last time period to be sampled will be time period 10. In McDonald's notation the panel design structure is [(1-4-1-n)^5]. If the number of time periods, n.period, is 20 and no other panel design structure is specified, then the last 10 time periods will not be sampled. For serially alternating panels, when pnl_n = NA, the number of panels will be the sum of the elements in pan_dsgn (ignoring NA). If pnl_n is specified as m, then m panels will be created. For example, pnl_dsgn = c(1, 4, 1, 4) and pnl_n =3, [(1-4-1-4)^3] in McDonald's notation, will create first three panels of the 510 serially alternating panels specified by pnl_dsgn.

A serially alternating or rotating panel revisit design may not result in the same number of units being sampled during each time period, particularly during the initial start up period. The default is to not specify a startup option ("None"). Start up option "Partial_Begin" initiates the revisit design at the last time period scheduled for sampling in the first panel. For example, a [2-3-1-4] design starts at time period 6 instead of time period 1 under the Partial_Begin option. For a serially alternating panel structure, start up option "Partial_End" initiates the revisit design at the time period that begins the second serially alternating pattern. For example, a [2-3-1-4] design starts at time period 11 instead of time period 1. For a rotating panel structure design, use of Partial_End makes the assumption that the number of panels equals the number of time periods and adds units to the last "m" panels for time periods 1 to "m" as if number of time periods was extended by "m" where "m" is one less than then the sum of the panel design. For example, a [1-4-1-4-1-n] design would result in m = 10. Note that some designs with pnl_n not equal to the number of sample occasions can produce unespected panel designs. See examples.

Different types of panel structures can be combined, these are termed split panels by many authors, by specifying more than one list for the panels parameter. The total number of panels is the sum of the number of panels in each of the panel structures specified by the split panel design.

Value

A two-dimensional array of sample sizes to be sampled at each combination of panel and time period.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

References

McDonald, T. (2003). Review of environmental monitoring methods: survey designs. *Environmental Monitoring and Assessment* **85**, 277-292.

See Also

revisit_bibd create a balanced incomplete block panel revisit design

revisit_rand create a revisit design with random assignment to panels and time periods

panel_summary summarize characteristics of a revisit panel design

power.dsgn power calculation for multiple panel designs

cov.panel.dsgn covariance matrix for a panel design

plot_powerpaneldesign plot power curves for panel designs

Examples

```
# One panel of  60 sample units sampled at every time period: [1-0]
revisit_dsgn(20, panels = list(
  Annual=list(n = 60, pnl_dsgn = c(1, 0), pnl.n = NA,
              start_option = "None")), begin=1)

# Rotating panels of 60 units sampled once and never again: [1-n].  Number
# of panels equal n.period.
revisit_dsgn(20, panels=list(
  R60N=list(n=60, pnl_dsgn = c(1, NA), pnl_n=NA, start_option="None")),
           begin=1 )

# Serially alternating panel with three visits to sample unit then skip
# next two time periods: [3-2]
revisit_dsgn(20, panels=list(
  SA60PE=list(n=20, pnl_dsgn = c(3, 2), pnl_n=NA,
             start_option="Partial_End")), begin=1 )

# Split panel of sample units combining above two panel designs: [1-0, 1-n]
revisit_dsgn(n.period=20, begin=2017,  panels = list(
  Annual=list(n = 60, pnl_dsgn = c(1, 0), pnl.n = NA,
              start_option = "None"),
  R60N=list(n=60, pnl_dsgn = c(1, NA), pnl_n=NA, start_option="None") ))
```

`revisit_rand` *Create a Revisit Design with Random Assignment to Panels and Time Periods*

Description

Create a revisit design for a survey that specifies the panels and time periods that will be sampled by random selection of panels and time periods. Three options for random assignments are "period" where the number of time periods to be sampled in a panel is fixed, "panel" where the number panels to be sampled in a time period is fixed, and "none" where the number of panel-period combinations is fixed.

Usage

```
revisit_rand(n.period, n.pnl, rand.control = "period", n.visit, nsamp,
  panel_name = "Random", begin = 1, skip = 1)
```

Arguments

`n.period`	Number of time periods for the survey design. Typically, number of periods if sampling occurs once per period or number of months if sampling occurs once per month. (v, number of varieties (or treatments) in BIBD terms)
`n.pnl`	Number of panels
`rand.control`	Character value must be "none", "panel", or "period". Specifies whether the number of sample events will be fixed for each panel ("panel"), for each sample occasion ("occasion"), or for total panel-period combinations ("none"). Default is "panel".
`n.visit`	If rand_control is "panel", this is the number of panels that will be sampled in each time period. If rand_control is "period", this is the number of time periods to be sampled in each panel. If rand_control is "none", this is the total number of panel-period combinations that will have units sampled in the revisit design.
`nsamp`	Number of samples in each panel.
`panel_name`	Prefix for name of each panel
`begin`	Numeric name of first sampling occasion, e.g. a specific period.
`skip`	Number of sampling occasions to skip between planned sampling periods, e.g., sampling will occur only every 5 periods if skip = 5.

Details

The revisit design for a survey is created by random selection of panels and time periods that will have sample events. The number of sample occasions that will be visited by a panel is random.

Value

A two-dimensional array of sample sizes to be sampled for each panel and each time period.

Author(s)

Tony Olsen <Olsen.Tony@epa.gov>

See Also

revisit_dsgn create a panel revisit design

revisit_bibd create a balanced incomplete block panel revisit design

panel_summary summarize characteristics of a revisit panel design

power.dsgn power calculation for multiple panel designs

cov.panel.dsgn covariance matrix for a panel design

plot_powerpaneldesign plot power curves for panel designs

Examples

```
revisit_rand(n.period = 20, n.pnl = 10, rand.control = "none", n.visit = 50,
             nsamp = 20)
revisit_rand(n.period = 20, n.pnl = 10, rand.control = "panel", n.visit = 5,
             nsamp = 10)
revisit_rand(n.period = 20, n.pnl = 10, rand.control = "period",
              n.visit = 5, nsamp = 10)
```

---

sbcframe *Calculate Spatial Balance Grid Cell Extent and Proportion for a Survey Frame*

Description

This function calculates spatial balance grid cell extent and proportion for the sample frame.

Usage

```
sbcframe(sfobject, nrows = 5, dxdy = TRUE)
```

Arguments

sfobject	An object of class sf that contains the survey frame.
nrows	Number of rows (and columns) for the grid of cells. The default is 5.
dxdy	Indicator for equal x-coordinate and y-coordinate grid cell increments, where TRUE means the increments are equal and FALSE means the increments are not equal. The default is TRUE.

Value

List containing the following components:

extent the frame extent for each grid cell

prop the frame proportion for each grid cell

xmin the grid x-coordinate minimum value

xmax the grid x-coordinate maximum value

ymin the grid y-coordinate minimum value

ymax the grid y-coordinate maximum value

dx the grid cell x-coordinate increment value

dy the grid cell y-coordinate increment value

xc the vector of grid cell x-coordinates

yc the vector of grid cell y-coordinates

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`sbcsamp` *Calculate Spatial Balance Grid Cell Extent and Proportions for a Survey Design*

Description

This function calculates spatial balance grid cell extent and proportions for a survey design. The user must provide either sbc.frame or values for dx, dy, xc, and yc.

Usage

```
sbcsamp(spsample, sbc.frame = NULL, dx = NULL, dy = NULL,
  xc = NULL, yc = NULL)
```

Arguments

`spsample`	Object of class SpatialDesign produced by either the grts or irs functions that contains survey design information and additional attribute (auxiliary) variables.
`sbc.frame`	The object created by the sbcframe function. The default is NULL.
`dx`	Grid cell x-coordinate increment value. The default is NULL.
`dy`	Grid cell y-coordinate increment value. The default is NULL.
`xc`	Vector of grid cell x-coordinates. The default is NULL.
`yc`	Vector of grid cell y-coordinates. The default is NULL.

Value

List containing the following components:

extent the sample extent for each grid cell

prop the sample proportion for each grid cell

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

`SC_estuaries` *Estuaries in South Carolina*

Description

A dataset containing attributes for estuaries in South Carolina.

Usage

```
SC_estuaries
```

Format

A data frame with 135 rows and 10 attributes:

siteID site ID value.

xcoord Albers projection x-coordinate.

ycoord Albers projection y-coordinate.

wgt survey design weight.

Stratum stratum code.

Status site evaluation status code.

IBI_score IBI (index of biotic integrity) score.

IBI_status status category of the IBI score.

WQ_score WQ (water quality) score.

WQ_status status category of the WQ score.

selectFeatureID *Identify a Feature in an sf Object for Selecting a Sample Point*

Description

This function determines the ID value of the feature in an sf object from which a sample point will be selected.

Usage

```
selectFeatureID(rdx, cellID, featureMeasure, featureID, mdm, id)
```

Arguments

rdx	Value of the randomized hierarchical address identifying a grid cell that will get a sample point.
cellID	Vector of grid cell IDs.
featureMeasure	Vector of grid cell sf feature lengths for linestring objects or sf feature areas for polygon objects.
featureID	Vector of grid cell sf feature IDs.
mdm	Vector of multidensity multipliers for the shapefile features.
id	Vector of sf feature IDs.

Value

The ID of an sf feature.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

selectframe *Internal Function: Select All Points in Survey Frame*

Description

Internal Function: Select All Points in Frame

Usage

```
selectframe(rord, xc, yc, dx, dy, pts)
```

Arguments

rord	Vector of the index value for all cells.
xc	Vector of x-coordinates that define the cells.
yc	Vector of y-coordinates that define the cells.
dx	Width of the cells along the x-axis.
dy	Width of the cells along the y-axis.
pts	Data frame containing id values, x-coordinates, y-coordinates, and mdm values.

Value

The id value for all points in the frame.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

selectpts *Internal Function: Select Probability Sample from a Set of Cells*

Description

Selects a sample of size one or larger from a set of cells based on inclusion probabilities.

Usage

```
selectpts(rdx, xc, yc, dx, dy, pts)
```

Arguments

rdx	Vector of the index value for selected cells.
xc	Vector of x-coordinates that define the cells.
yc	Vector of y-coordinates that define the cells.
dx	Width of the cells along the x-axis.
dy	Width of the cells along the y-axis.
pts	Data frame containing id values, x-coordinates, y-coordinates, and mdm values.

Value

The id value for the sample points.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov> Tony Olsen <olsen.tony@epa.gov>

simex — *Internal Function: Extrapolation for Simulation-Extrapolation Function*

Description

This function executes the extrapolation step of the simulation extrapolation deconvolution method (Stefanski and Bay, 1996). The function can accomodate single-stage and two-stage samples.

Usage

```
simex(z, val, sigma, var.sigma, cluster.ind, cluster)
```

Arguments

z	Vector of the response value for each site.
val	Vector of the set of values at which the CDF is estimated.
sigma	Measurement error variance.
var.sigma	Variance of the estimated measurement error variance.
cluster.ind	Logical value that indicates whether the survey design utilizes two stages.
cluster	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.

Value

Output is a list containing the following matrices:

g values of the function g(.) evaluated at val for each value of z

dg values of the derivative of the function g(.)

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

sorted — *Internal Function: Determine if a Vector is Sorted*

Description

This function determines whether the input set of values is a nondecreasing sequence.

Usage

```
sorted(x)
```

Arguments

x	Vector of values.

Value

Logical variable, where TRUE = sorted and FALSE = not sorted.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

sp2shape *Create an ESRI shapefile from an sp package object*

Description

This function creates an ESRI shapefile from an sp package object. The function can also accommodate an object created by the grts or irs functions in spsurvey. The type of shapefile, i.e., point, polyline, or polygon, is determined by the class of the sp object.

Usage

```
sp2shape(sp.obj, shpfilename)
```

Arguments

sp.obj	the sp package object or object created by either the grts or irs functions.
shpfilename	Character string containing the name of the shapefile. The shapefile name should include the ".shp" extension. If the name does not include that extension, it will be added.

Value

An ESRI shapefile.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

SpatialDesign-class *Class SpatialDesign*

Description

Create the definition for class SpatialDesign, a class for spatial survey designs.

Create the wrapper function for class SpatialDesign.

Define S3 and S4 methods for summary and plot for class SpatialDesign.

Usage

```
SpatialDesign(design, sp_obj)

## S3 method for class 'SpatialDesign'
summary(object, ..., auxvar = NULL,
  sfframe = NULL, tess_ind = TRUE, sbc_ind = FALSE, nrows = 5,
  dxdy = TRUE)

## S4 method for signature 'SpatialDesign'
summary(object, ..., auxvar = NULL,
  sfframe = NULL, tess_ind = TRUE, sbc_ind = FALSE, nrows = 5,
  dxdy = TRUE)

## S3 method for class 'SpatialDesign'
plot(x, y, ..., sfframe = NULL, stratum = NULL,
  mdcaty = NULL, auxvar = NULL, pdffile = NULL, width = 8,
  height = 10)

## S4 method for signature 'SpatialDesign,missing'
plot(x, y, ..., sfframe = NULL,
  stratum = NULL, mdcaty = NULL, auxvar = NULL, pdffile = NULL,
  width = 8, height = 10)
```

Arguments

design	Object of class list containing specifications for the survey design.
sp_obj	Object of class SpatialPointsDataFrame containing spatial attributes that have spatial point locations.
object	SpatialDesign object.
...	Arguments passed through.
auxvar	Vector containing the names of variables in the data slot of the SpatialDesign object that identify auxiliary variables to be used to summarize the survey design or create plots of the survey design. The default is NULL.
sfframe	Object of class sf that contains the survey design frame. The default is NULL.

tess_ind	a logical variable indicating whether spatial balance metrics are calculated using proportions obtained from the intersection of Dirichlet tesselation polygons for the sample points with the frame object. TRUE means calculate the metrics. FALSE means do not calculate the metrics. The default is TRUE.
sbc_ind	a logical variable indicating whether spatial balance metrics are calculated using proportions obtained from a rectangular grid superimposed on the sample points and the frame. TRUE means calculate the metrics. FALSE means do not calculate the metrics. The default is FALSE.
nrows	number of rows (and columns) for the grid of cells. The default is 5.
dxdy	indicator for equal x-coordinate and y-coordinate grid cell increments, where TRUE means the increments are equal and FALSE means the increments are not equal. The default is TRUE.
x	SpatialDesign object.
y	Missing - this argument is not used.
stratum	name of the attribute from the sfframe object that identifies stratum membership for each feature in the frame. If stratum equals NULL, the design is unstratified, and an attribute named "stratum" (with all its elements equal to the stratum name specified in design) is added to the sfframe object. The default is NULL.
mdcaty	name of the attribute from the sfframe object that identifies the unequal probability category for each feature in the survey frame. The default is NULL.
pdffile	a character variable containing the name of the pdf file to which output is written. If a value is not provided, output is written to the graphics window. The default is NULL.
width	width of the graphic region in inches. The default is 8.
height	height of the graphic region in inches. The default is 10.

Value

Object of class SpatialDesign.

A summary or plot depending on the method called.

Slots

design Object of class list containing specifications for the survey design

data Object of class data.frame containing the attribute data

coords.nrs Numeric object that records the column positions in data from which the coordinates were obtained

coords Object of class matrix containing the coordinates matrix, where points are rows in the matrix

bbox Object of class matrix containing the bounding box

proj4string Object of class CRS containing the projection string

Extends

Class "SpatialPointsDataFrame", directly.

Class "SpatialPoints", by class "SpatialPointsDataFrame".

Class "Spatial", by class "SpatialPoints".

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

spbalance *Calculate Spatial Balance Metrics for a Survey Design*

Description

This function calculates spatial balance metrics for a survey design. Two options for calculation of spatial balance metrics are available: (1) use proportions obtained from the intersection of Dirichlet tesselation polygons for the sample points with the frame object and (2) use proportions obtained from a rectangular grid superimposed on the sample points and the frame object. In both cases the proportions are used to calculate the spatial balance metrics. Two metrics are calculated: (1) the Pielou evenness measure and (2) the chi-square statistic.

Usage

```
spbalance(spsample, sfframe, tess_ind = TRUE, sbc_ind = FALSE,
  nrows = 5, dxdy = TRUE)
```

Arguments

spsample	Object of class SpatialDesign produced by either the grts or irs functions that contains survey design information and additional attribute (auxiliary) variables.
sfframe	An object of class sf that contains the survey frame.
tess_ind	Logical variable indicating whether spatial balance metrics are calculated using proportions obtained from the intersection of Dirichlet tesselation polygons for the sample points with the frame object. TRUE means calculate the metrics. FALSE means do not calculate the metrics. The default is TRUE.
sbc_ind	Logical variable indicating whether spatial balance metrics are calculated using proportions obtained from a rectangular grid superimposed on the sample points and the frame. TRUE means calculate the metrics. FALSE means do not calculate the metrics. The default is FALSE.
nrows	Number of rows (and columns) for the grid of cells. The default is 5.
dxdy	Indicator for equal x-coordinate and y-coordinate grid cell increments, where TRUE means the increments are equal and FALSE means the increments are not equal. The default is TRUE.

Value

List containing the following components:

`tess` results for spatial balance metrics using tesselation polygons

`sbc` results for spatial balance metrics using a rectangular grid

If either the tess_ind or sbc_ind arguments are set to FALSE, the corresponding component in the list is set to NULL. Otherwise, each components of the list is a lists that contains the following components:

`J_subp` Pielou evenness measure

`chi_sq` Chi-square statistic

`extent` frame extent for each Dirichlet tesselation polygon or rectangular grid cell

`prop` frame proportion for each Dirichlet tesselation polygon or rectangular grid cell

Other Functions Required

`deldir` deldir package function that computes the Delaunay triangulation and Dirichlet tesselation of a set of points

`tile.list` deldir package function that extracts coordinates of the Dirichlet tesselation polygons from the object produced by the deldir function

`gIntersection` rgeos package function that determines the intersection between two sp package objects

`LinesLength` sp package function that determines length of the line segemnts in a class Lines object

`sbcframe` function to calculate spatial balance grid cell extent and proportions for a sample frame

`sbcsamp` function to calculate spatial balance grid cell extent and proportions for a survey design

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
## Not run:
design <- list(
  Stratum1=list(panel=c(PanelOne=50), seltype="Equal", over=10),
  Stratum2=list(panel=c(PanelOne=50, PanelTwo=50), seltype="Unequal",
    caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
samp <- grts(design=design, DesignID="Test.Site", type.frame="area",
  src.frame="shapefile", in.shape="shapefile.shp", stratum="stratum",
  mdcaty="mdcaty", shapefile=TRUE, out.shape="sample.shp")
sframe <- read.shp("shapefile.shp")
spbalance(samp, sframe, sbc_ind = TRUE)

## End(Not run)
```

spsurvey.analysis *Create an Object of Class spsurvey.analysis*

Description

This function creates an object of class spsurvey.analysis that contains all of the information necessary to use the analysis functions in the spsurvey package.

Usage

```
spsurvey.analysis(sites = NULL, subpop = NULL, design = NULL,
  data.cat = NULL, data.cont = NULL, siteID = NULL, wgt = NULL,
  sigma = NULL, var.sigma = NULL, xcoord = NULL, ycoord = NULL,
  stratum = NULL, cluster = NULL, wgt1 = NULL, xcoord1 = NULL,
  ycoord1 = NULL, popsize = NULL, popcorrect = FALSE,
  pcfsize = NULL, N.cluster = NULL, stage1size = NULL,
  support = NULL, sizeweight = FALSE, swgt = NULL, swgt1 = NULL,
  vartype = "Local", conf = 95, pctval = c(5, 10, 25, 50, 75, 90,
  95))
```

Arguments

sites	Data frame consisting of two variables: the first variable is site IDs and the second variable is a logical vector indicating which sites to use in the analysis. If this data frame is not provided, then the data frame will be created, where (1) site IDs are obtained either from the design argument, the siteID argument, or both (when siteID is a formula); and (2) a variable named use.sites that contains the value TRUE for all sites is created. The default is NULL.
subpop	Data frame describing sets of populations and subpopulations for which estimates will be calculated. The first variable is siteIDs and each subsequent variable identifies a Type of population, where the variable name is used to identify Type. A Type variable identifies each site with one of the subpopulations of that Type. If this data frame is not provided, then the data frame will be created, where (1) site IDs are obtained either from the design argument, the siteID argument, or both (when siteID is a formula); and (2) a single Type variable named all.sites that contains the value "All Sites" for all sites is created. The default is NULL.
design	Data frame consisting of design variables. If variable names are provided as formulas in the corresponding arguments, then the formulas are interpreted using this data frame. If this data frame is not provided, then the data frame will be created from inputs to the design variables in the argument list. The default is NULL. If variable names are not provided as formulas, then variables should be named as follows: **siteID** Vector of site IDs **wgt** Vector of weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample

xcoord Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample

ycoord Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample

stratum Vector of the stratum codes for each site

cluster Vector of the stage one sampling unit (primary sampling unit or cluster) codes for each site

wgt1 Vector of stage one weights in a two-stage design

xcoord1 Vector of the stage one x-coordinates for location in a two-stage design

ycoord1 Vector of the stage one y-coordinates for location in a two-stage design

support Vector of support values - for a finite resource, the value one (1) for a for site. For an extensive resource, the measure of the sampling unit associated with a site. Required for calculation of finite and continuous population correction factors.

swgt Vector of size-weights, which is the stage two size-weight for a two-stage design.

swgt1 Vector of stage one size-weights for a two-stage design.

`data.cat` Data frame of categorical response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. If psurvey.obj is not provided, then this argument is required. The default is NULL.

`data.cont` Data frame of continuous response variables. The first variable is site IDs. Subsequent variables are response variables. Missing data (NA) is allowed. The default is NULL.

`siteID` site IDs. This variable can be input directly or as a formula and must be supplied either as this argument or in the design data frame. The default is NULL.

`wgt` Vector of final adjusted weights, which are either the weights for a single- stage sample or the stage two weights for a two-stage sample. This variable can be input directly or as a formula and must be supplied either as this argument or in the design data frame. The default is NULL.

`sigma` Measurement error variance. This variable must be a vector containing a value for each response variable and must have the names attribute set to identify the response variable names. Missing data (NA) is allowed. The default is NULL.

`var.sigma` Variance of the measurement error variance. This variable must be a vector containing a value for each response variable and must have the names attribute set to identify the response variable names. Missing data (NA) is allowed. The default is NULL.

`xcoord` Vector of x-coordinates for location, which are either the x-coordinates for a single-stage sample or the stage two x-coordinates for a two-stage sample. This variable can be input directly or as a formula and must be supplied either as this argument or in the design data frame when argument vartype is set to "Local". The default is NULL.

`ycoord` Vector of y-coordinates for location, which are either the y-coordinates for a single-stage sample or the stage two y-coordinates for a two-stage sample. This

variable can be input directly or as a formula and must be supplied either as this argument or in the design data frame when argument vartype is set to "Local". The default is NULL.

`stratum` The stratum codes. This variable can be input directly or as a formula. The default is NULL.

`cluster` Vector of the stage one sampling unit (primary sampling unit or cluster) codes. This variable can be input directly or as a formula. The default is NULL.

`wgt1` Vector of final adjusted stage one weights. This variable can be input directly or as a formula. The default is NULL.

`xcoord1` Vector of the stage one x-coordinates for location. This variable can be input directly or as a formula. The default is NULL.

`ycoord1` Vector of the stage one y-coordinates for location. This variable can be input directly or as a formula. The default is NULL.

`popsize` Known size of the resource, which is used to perform ratio adjustment to estimators expressed using measurement units for the resource. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. The argument must be in the form of a list containing an element for each population Type in the subpop data frame, where NULL is a valid choice for a population Type. The list must be named using the column names for the population Types in subpop. If a population Type doesn't contain subpopulations, then each element of the list is either a single value for an unstratified sample or a vector containing a value for each stratum for a stratified sample, where elements of the vector are named using the stratum codes. If a population Type contains subpopulations, then each element of the list is a list containing an element for each subpopulation, where the list is named using the subpopulation names. The element for each subpopulation will be either a single value for an unstratified sample or a named vector of values for a stratified sample. The default is NULL.

Example popsize for a stratified sample:

```
popsize = list("Pop 1"=c("Stratum 1"=750,
"Stratum 2"=500,
"Stratum 3"=250),
"Pop 2"=list("SubPop 1"=c("Stratum 1"=350,
"Stratum 2"=250,
"Stratum 3"=150),
"SubPop 2"=c("Stratum 1"=250,
"Stratum 2"=150,
"Stratum 3"=100),
"SubPop 3"=c("Stratum 1"=150,
"Stratum 2"=150,
"Stratum 3"=75)),
"Pop 3"=NULL)
```

Example popsize for an unstratified sample:

```
popsize = list("Pop 1"=1500,
"Pop 2"=list("SubPop 1"=750,
"SubPop 2"=500,
"SubPop 3"=375),
"Pop 3"=NULL)
```

`popcorrect` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for argument pcfsize and for the support variable of the design argument. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster and stage1size, and for the support variable of the design argument.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`support` Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. This variable can be input directly or as a formula. The default is NULL.

`sizeweight` Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

`swgt` Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample. This variable can be input directly or as a formula. The default is NULL.

`swgt1` Vector of the stage one size-weight for each site. This variable can be input directly or as a formula. The default is NULL.

`vartype` The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".

`conf` Numeric value for the confidence level. The default is 95.

`pctval` The set of values at which percentiles are estimated. The default set is: 5, 10, 25, 50, 75, 90, 95.

Value

List of class spsurvey.analysis. Only those sites indicated by the logical variable in the sites data frame are retained in the output. The sites, subpop, and design data frames will always exist in the output. At least one of the data.cat and data.cont data frames will exist. Depending upon values of the input variables, other elements in the output may be NULL. The output list is composed of the following elements:

sites the sites data frame.

subpop the subpop data frame.

design the design data frame.

data.cat the data.cat data frame.

data.cont the data.cont data frame.

sigma measurement error variance.

var.sigma variance of the estimated measurement error variance.

stratum.ind a logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.

cluster.ind a logical value that indicates whether the sample is a two-stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.

popsize the known size of the resource.

pcfactor.ind a logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor.

pcfsize size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample.

N.cluster the number of stage one sampling units in the resource stage1sizesize of the stage one sampling units of a two-stage sample.

swgt.ind a logical value that indicates whether the sample is a size-weighted sample, where TRUE = a size-weighted sample and FALSE = not a size-weighted sample.

vartype the choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.

conf the confidence level.

pctval the set of values at which percentiles are estimated.

Other Functions Required

`dframe.check` check site IDs, the sites data frame, the subpop data frame, and the data.cat data frame to assure valid contents and, as necessary, create the sites data frame and the subpop data frame

`vecprint` takes an input vector and outputs a character string with line breaks inserted

`uniqueID` creates unique site IDs by appending a unique number to each occurrence of a site ID

`input.check` check input values for errors, consistency, and compatibility with analytical functions

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
#Categorical variable example:
mysiteID <- paste("Site", 1:100, sep="")
mysites <- data.frame(
  siteID=mysiteID,
  Active=rep(TRUE, 100))
mysubpop <- data.frame(
  siteID=mysiteID,
  All.Sites=rep("All Sites", 100),
  Resource.Class=rep(c("Good","Poor"), c(55,45)))
mydesign <- data.frame(
  siteID=mysiteID,
  wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1", "Stratum2"), 50))
mydata.cat <- data.frame(
  siteID=mysiteID,
  CatVar=rep(c("north", "south", "east", "west"), 25))
mypopsize <- list(
  All.Sites=c(Stratum1=3500, Stratum2=2000),
  Resource.Class=list(Good=c(Stratum1=2500, Stratum2=1500),
                      Poor=c(Stratum1=1000, Stratum2=500)))

# Continuous variable example - including deconvolution estimates:
mydesign <- data.frame(
  ID=mysiteID,
  wgt=runif(100, 10, 100),
  xcoord=runif(100),
  ycoord=runif(100),
  stratum=rep(c("Stratum1", "Stratum2"), 50))
ContVar <- rnorm(100, 10, 1)
mydata.cont <- data.frame(
  siteID=mysiteID,
  ContVar=ContVar,
  ContVar.1=ContVar + rnorm(100, 0, sqrt(0.25)),
  ContVar.2=ContVar + rnorm(100, 0, sqrt(0.50)))
mysigma <- c(ContVar=NA, ContVar.1=0.25, ContVar.2=0.50)
spsurvey.analysis(sites=mysites, subpop=mysubpop[,1:2], design=mydesign,
  data.cont=mydata.cont, siteID=~ID, sigma=mysigma, popsize=mypopsize[1])
```

---

total.est *Estimators for Population Total, Mean, Variance, and Standard Deviation*

Description

This function calculates estimates of the population total, mean, variance, and standard deviation of a response variable, where the response variable may be defined for either a finite or an extensive resource. In addition the standard error of the population estimates and confidence bounds are calculated. The Horvitz-Thompson estimator is used to calculate the total, variance, and standard deviation estimates. The Horvitz-Thompson ratio estimator, i.e., the ratio of two Horvitz-Thompson estimators, is used to calculate the mean estimate. Variance estimates are calculated using either the local mean variance estimator or the simple random sampling (SRS) variance estimator. The choice of variance estimator is subject to user control. The local mean variance estimator requires the x-coordinate and the y-coordinate of each site. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. Confidence bounds are calculated using a Normal distribution multiplier. The function can accommodate a stratified sample. For a stratified sample, separate estimates and standard errors are calculated for each stratum, which are used to produce estimates and standard errors for all strata combined. Strata that contain a single value are removed. For a stratified sample, when either the size of the resource or the sum of the size-weights of the resource is provided for each stratum, those values are used as stratum weights for calculating the estimates and standard errors for all strata combined. For a stratified sample when neither the size of the resource nor the sum of the size-weights of the resource is provided for each stratum, estimated values are used as stratum weights for calculating the estimates and standard errors for all strata combined. The function can accommodate single-stage and two-stage samples for both stratified and unstratified sampling designs. Finite population and continuous population correction factors can be utilized in variance estimation. The function checks for compatibility of input values and removes missing values.

Usage

```
total.est(z, wgt, x = NULL, y = NULL, stratum = NULL,
  cluster = NULL, wgt1 = NULL, x1 = NULL, y1 = NULL,
  popsize = NULL, popcorrect = FALSE, pcfsize = NULL,
  N.cluster = NULL, stage1size = NULL, support = NULL,
  sizeweight = FALSE, swgt = NULL, swgt1 = NULL, vartype = "Local",
  conf = 95, check.ind = TRUE, warn.ind = NULL, warn.df = NULL,
  warn.vec = NULL)
```

Arguments

z	Vector of the response value for each site.
wgt	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
x	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample. The default is NULL.
y	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample. The default is NULL.
stratum	Vector of the stratum for each site. The default is NULL.

`cluster` Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site. The default is NULL.

`wgt1` Vector of the final adjusted stage one weight for each site. The default is NULL.

`x1` Vector of the stage one x-coordinate for location for each site. The default is NULL.

`y1` Vector of the stage one y-coordinate for location for each site. The default is NULL.

`popsize` Known size of the resource, which is used to calculate strata proportions for calculating estimates for a stratified sample. For a finite resource, this argument is either the total number of sampling units or the known sum of size-weights. For an extensive resource, this argument is the measure of the resource, i.e., either known total length for a linear resource or known total area for an areal resource. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`popcorrect` Logical value that indicates whether finite or continuous population correction factors should be employed during variance estimation, where TRUE = use the correction factor and FALSE = do not use the correction factor. The default is FALSE. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.

`pcfsize` Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`N.cluster` The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. The default is NULL.

`stage1size` Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1". The default is NULL.

`support` Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from an extensive resource, which is required for calculation of finite and continuous population correction factors. The default is NULL.

`sizeweight` Logical value that indicates whether size-weights should be used in the analysis, where TRUE = use the size-weights and FALSE = do not use the size-weights. The default is FALSE.

swgt	Vector of the size-weight for each site, which is the stage two size-weight for a two-stage sample. The default is NULL.
swgt1	Vector of the stage one size-weight for each site. The default is NULL.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator. The default is "Local".
conf	Numeric value for the confidence level. The default is 95.
check.ind	Logical value that indicates whether compatability checking of the input values is conducted, where TRUE = conduct compatibility checking and FALSE = do not conduct compatibility checking. The default is TRUE.
warn.ind	Logical value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated. The default is NULL.
warn.df	Data frame for storing warning messages. The default is NULL.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator. The default is NULL.

Value

If the function was called by the cont.analysis function, then output is an object in list format composed of the Results data frame, which contains estimates and confidence bounds, and the warn.df data frame, which contains warning messages. If the function was called directly, then output is the Results data frame.

Other Functions Required

input.check check input values for errors, consistency, and compatibility with analytical functions

wnas remove missing values

vecprint takes an input vector and outputs a character string with line breaks inserted

total.var calculate variance of the total, mean, variance, and standard deviation estimates

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
z <- rnorm(100, 10, 1)
wgt <- runif(100, 10, 100)
total.est(z, wgt, vartype="SRS")

x <- runif(100)
y <- runif(100)
total.est(z, wgt, x, y)
```

`total.var` *Variance Estimate for Population Total, Mean, Variance, and Standard Deviation*

Description

This function calculates variance estimates of the estimated population total, mean, variance, and standard deviation of a response variable. Either the simple random sampling (SRS) variance estimator or the local mean variance estimator is calculated, which is subject to user control. The SRS variance estimator uses the independent random sample approximation to calculate joint inclusion probabilities. The function can accomodate single-stage and two-stage samples.

Usage

```
total.var(z, wgt, x, y, mean.est, var.est, sd.est, stratum.ind,
  stratum.level, cluster.ind, cluster, wgt1, x1, y1, pcfactor.ind, pcfsize,
  N.cluster, stage1size, support, vartype, warn.ind, warn.df, warn.vec)
```

Arguments

`z`	Vector of the response value for each site.
`wgt`	Vector of the final adjusted weight (inverse of the sample inclusion probability) for each site, which is either the weight for a single-stage sample or the stage two weight for a two-stage sample.
`x`	Vector of x-coordinate for location for each site, which is either the x-coordinate for a single-stage sample or the stage two x-coordinate for a two-stage sample.
`y`	Vector of y-coordinate for location for each site, which is either the y-coordinate for a single-stage sample or the stage two y-coordinate for a two-stage sample.
`mean.est`	The mean estimate.
`var.est`	The variance estimate.
`sd.est`	The standard deviation estimate.
`stratum.ind`	Logical value that indicates whether the sample is stratified, where TRUE = a stratified sample and FALSE = not a stratified sample.
`stratum.level`	The stratum level.
`cluster.ind`	Logical value that indicates whether the sample is a two-stage sample, where TRUE = a two-stage sample and FALSE = not a two-stage sample.
`cluster`	Vector of the stage one sampling unit (primary sampling unit or cluster) code for each site.
`wgt1`	Vector of the final adjusted stage one weight for each site.
`x1`	Vector of the stage one x-coordinate for location for each site.
`y1`	Vector of the stage one y-coordinate for location for each site.

pcfactor.ind	Logical value that indicates whether the population correction factor is used during variance estimation, where TRUE = use the population correction factor and FALSE = do not use the factor. To employ the correction factor for a single-stage sample, values must be supplied for arguments pcfsize and support. To employ the correction factor for a two-stage sample, values must be supplied for arguments N.cluster, stage1size, and support.
pcfsize	Size of the resource, which is required for calculation of finite and continuous population correction factors for a single-stage sample. For a stratified sample this argument must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
N.cluster	The number of stage one sampling units in the resource, which is required for calculation of finite and continuous population correction factors for a two-stage sample. For a stratified sample this variable must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes.
stage1size	Size of the stage one sampling units of a two-stage sample, which is required for calculation of finite and continuous population correction factors for a two-stage sample and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute must be set to identify both stratum codes and stage one sampling unit codes using a convention where the two codes are separated by the & symbol, e.g., "Stratum 1&Cluster 1".
support	Vector of the support value for each site - the value one (1) for a site from a finite resource or the measure of the sampling unit associated with a site from a continuous resource, which is required for calculation of finite and continuous population correction factors.
vartype	The choice of variance estimator, where "Local" = local mean estimator and "SRS" = SRS estimator.
warn.ind	LogicLal value that indicates whether warning messages were generated, where TRUE = warning messages were generated and FALSE = warning messages were not generated.
warn.df	Data frame for storing warning messages.
warn.vec	Vector that contains names of the population type, the subpopulation, and an indicator.

Value

Object in list format composed of a vector named varest, which contains variance estimates, a logical variable named warn,ind, which is the indicator for warning messges, and a data frame named warn.df, which contains warning messages.

Other Functions Required

localmean.weight calculate the weighting matrix for the local mean variance estimator

localmean.var calculate the local mean variance estimator

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

uniqueID *Internal Function: Create Unique IDs for a Survey Design*

Description

This function creates unique site IDs by appending a unique number to each occurrence of a site ID. It is intended for survey designs that have repeat visits to sites.

Usage

```
uniqueID(siteID)
```

Arguments

siteID	Vector of site IDs.

Value

Vector of unique site IDs.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

UT_ecoregions *Ecoregions in Utah*

Description

A dataset containing attributes for ecoregions in Utah.

Usage

```
UT_ecoregions
```

Format

An object of class sf (simple features) containing 10 features and 3 attributes:

Level3 Level 3 ecoregion code for the polygon.

Level3_Nam Level 3 ecoregion name for the polygon.

Area_ha area of the polygon in hectares.

vecprint *Internal Function: Create Vector to Print*

Description

This function takes an input vector and outputs a character string with line breaks inserted so that, whenever possible, no line in the string exceeds the input value n.char, which is set to 78 characters by default. The input vector is coereced to mode character. When an element of the input vector is greater than n.char characters in length, then that element is inserted in the output character string as an individual line.

Usage

```
vecprint(x, n.char = 78)
```

Arguments

x	Character vector.
n.char	The maximum number of characters per line. The default is 78.

Value

Character string that is suitable for printing by the functions: stop, warning, or cat.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
sites <- paste("Site Number", 1:50)
sites.str <- vecprint(sites)
cat(sites.str)

temp <- c(1, 5, 21:25, 33:37)
sites.str <- vecprint(sites[temp])
warning(paste("\nThe following site ID values were removed from the
  analysis:\n", sites.str, sep=""))
```

warnprnt *Internal Function: Print the Warnings Data Frame*

Description

This function prints the warnings data frame.

Usage

```
warnprnt(warn.df = get("warn.df", envir = .GlobalEnv),
  m = 1:nrow(warn.df))
```

Arguments

warn.df	Data frame that contains warning messages. The default is "warn.df", which is the name given to the warnings data frame created by functions in the spsurvey package.
m	Vector of indices for warning messages that are to be printed. The default is a vector containing the integers from 1 through the number of rows in warn.df, which will print all warning messages in the data frame.

Value

Invisible return. Prints warnings.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

wnas *Internal Function: Remove NAs from Data*

Description

This function removes missing values from data for which the mode is one of the following: numeric, logical, or character. Data that is not one of those modes will cause the function to terminate with an error message. For numeric data this function removes values that are not finite, i.e., missing value (NA), not a number (NaN), infinity (Inf), and minus infinity (-Inf). For logical data this function removes missing values (NA). For character data the following values are removed: "", "NA", NA (R only), "NaN", "Inf", and "-Inf". For a factor this function removes the following values: NA, NaN, Inf, and -Inf. For a vector this function returns the vector with the indicated values removed. For a data frame this function returns the data frame with rows removed that contain at least one indicated value. For a list with components that are the same length, the list is converted to a data frame. For a list with components that are not the same length, the function prints an error message and terminates. When the process of removing missing values produces an object that no longer contains any elements (vector) or rows (data frame or list), then a NULL object is returned.

Usage

```
wnas(data)
```

Arguments

data	Object of type numeric, logical, or character.

Value

Object with NAs removed.

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

---

write.object *Write an Object to a Plot*

Description

This function writes the contents of an object to a plot. The object may be either a data frame or a matrix. Values in the input data frame or matrix must be of class numeric, character, or factor.

Usage

```
write.object(obj, n.digits = 2, r.names = TRUE, c.names = TRUE,
  r.cex = 1, c.cex = 1, miss = "NA")
```

Arguments

obj	The object (either a data frame or a matrix).
n.digits	Number of digits after the decimal point for numeric values. The default is 2.
r.names	Logical value that indicates whether to print the row names, where TRUE = print the row names and FALSE = do not print the row names. The default is TRUE.
c.names	Logical value that indicates whether to print the column names, where TRUE = print the column names and FALSE = do not print the column names. The default is TRUE.
r.cex	Character expansion parameter for the row labels. The default is 1.
c.cex	Character expansion parameter for the column labels. The default is 1.
miss	The missing value code expressed as a character string. The default is "NA".

Value

The function returns NULL. Side effect of the function is to write contents of the input object to a plot.

Other Functions Required

`input.format` format an input value

Author(s)

Tom Kincaid <Kincaid.Tom@epa.gov>

Examples

```
z <- rnorm(100)
z.mean <- c(tapply(z, rep(1:4, rep(25,4)), mean), mean(z))
z.sd <- sqrt(c(tapply(z, rep(1:4, rep(25,4)), var), var(z)))
z.upper <- z.mean+1.96*z.sd
z.lower <- z.mean-1.96*z.sd
obj <- data.frame(rbind(z.mean, z.sd, z.upper, z.lower))
dimnames(obj) <- list(c("Mean Estimate", "Standard Deviation", "Lower 95\%
  Conf. Bound", "Upper 95\% Conf. Bound"), c(paste("Stratum", 1:4, sep=""),
  "AllStrata"))
write.object(obj, n.digits=3, r.cex=0.75)

obj <- data.frame(matrix(round(5 + runif(30), 1), nrow=6))
colnames(obj) <- c("United States", "Russia", "Germany", "Japan", "France")
write.object(obj, n.digits=1, r.names=FALSE)
```

Index