User Guide

for

spsurvey, version 2.2

Probability Survey Design and Analysis Functions

by

Thomas Kincaid

May 6, 2011

INTRODUCTION

The functions included in package **spsurvey** are intended for design and analysis of probability surveys. The functions were written for the U.S. Environmental Protection Agency's Environmental Monitoring and Assessment Program (EMAP; Messer et al., 1991) to design and analyze probability surveys of environmental resources (Diaz-Ramos *et al.*, 1995). The functions in **spsurvey** can select generalized random-tesselation stratified (GRTS) and independent random sample (IRS) survey designs. Although the function are applicable for a wide range of environmental survey designs, the **spsurvey** analysis functions were written to accommodate data generated by a GRTS sampling design. For further discussion of the GRTS design see Stevens and Olsen (2004). 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 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.

DESIGN FUNCTIONS

Descriptions

The design functions in **spsurvey** are organized in a hierarchical structure composed of three levels. The first level functions organize input and output for selection of GRTS and IRS survey designs. The first level functions call the second level function that is applicable for the type of environmental resource. In addition, three of the second level functions (**read.dbf**, **read.shape**, and **sp2shape**) are utilized for reading from or writing to Environmental Systems Research Institute, Inc. (ESRI) shapefiles (see http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf). A short description of each function in the top two levels is provided. Functions in the third level are not intended for access by the user. Further details regarding the functions are provided in subsequent sections and in the R help documentation for **spsurvey**.

First Level Functions

grts

This function organizes input and output for selection of a GRTS survey design. Options for the survey design include: stratification, selection probabilities that are specified by categories, selection probabilities that are proportional to an auxiliary variable, survey over time structures, and provision for an oversample.

irs

This function organizes input and output for selection of an IRS survey design. Options for the survey design include: stratification, selection probabilities that are specified by categories, selection probabilities that are proportional to an auxiliary variable, survey over time structures, and provision for an oversample.

Second Level Functions

grtspts

This function select a GRTS sample of a finite resource. This function creates a grid structure, uses hierarchical randomization to ensure that the sample will include no more than one point per grid cell, and picks points from randomly selected cells.

grtslin

This function select a GRTS sample of a linear resource. This function creates a grid structure, uses hierarchical randomization to ensure that the sample will include no more than one point per grid cell, and picks points from randomly selected cells.

grtsarea

This function select a GRTS sample of a finite resource. This function creates a grid structure, uses hierarchical randomization to ensure that the sample will include no more than one point per grid cell, and picks points from randomly selected cells.

irspts

This function select an IRS sample of a finite resource.

irslin

This function select an IRS sample of a linear resource.

irsarea

This function select an IRS sample of a finite resource.

read.dbf

This function reads the dbf (attributes) file of an ESRI shapefile. The function reads either a single dbf file or multiple dbf files. For multiple dbf files, all of the dbf files must have the same variable names.

read.shape

This function reads an ESRI shapefile. The function reads either a single shapefile or multiple shapefiles. For multiple shapefiles, all of the shapefiles must be the same shapefile type, i.e., point, polyline, or polygon.

sp2shape

This function creates an ESRI shapefile from an **sp** package object. The type of shapefile, i.e., point, polyline, or polygon, is determined by the class of the **sp** object, which must be either SpatialPointsDataFrame, SpatialLinesDataFrame, or SpatialPolygonsDataFrame. For further information regarding the **sp** object, see documentation for the **sp** package.

write.dbf

This function writes a data frame to the dbf file of an ESRI shapefile.

Data Input

Overview

Arguments to the first level functions provide information for the following categories: (1) design specifications, which includes specifications for each stratum for a stratified design; (2) information regarding the frame, which includes the type of frame and either the name of the shapefile containing the frame or coordinates for points in the frame (for a finite resource); (3) attributes associated with elements in the frame; and (4) additional information that controls the functions, including output from the functions. An extensive description follows regarding data entry for the first level functions. For the second level functions, information is provided for the functions that read and write ESRI shapefiles but not for the functions called by the first level functions. Data entry for the third level functions is not described.

First Level Functions (grts and irs)

Information regarding design specifications is provided by argument **design**, which is a named list that contains either a single element (for an unstratified design) or an element for each stratum (for a stratified design). Each element in the **design** argument is a list composed of four components: (1) **panel** - a named vector of sample sizes for each panel in stratum; (2) **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 in argument **mdcaty** (see below), "Continuous" - unequal probability selection proportional to the values in **mdcaty**; (3) **caty.n** - when **seltype** equals "Unequal", a named vector of sample sizes for each category specified by **mdcaty**, where the sum of the sample sizes in **caty.n** must equal the sum of the sample sizes provided in **panel**, and the names in **caty.n** must be equivalent to the values in **mdcaty** for the stratum; and (4) **over** - the number of replacement sites ("oversample" sites) for the entire design, which is set equal to 0 if none are required.

An example of **design** for a stratified sample follows:

```
design = list("Stratum 1"=list(panel=c(Panel=50), seltype="Equal" over=10),
    "Stratum 2"=list(panel=c("Panel One"=50, "Panel Two"=50), seltype="Unequal",
    caty.n=c(CatyOne=25, CatyTwo=25, CatyThree=25, CatyFour=25), over=75))
```

An example of **design** for an unstratified sample follows:

```
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))
```

Two argument provide information pertaining to ID values for sample sites. **DesignID** provides a name for the design, which is used to create a site ID for each sample site. The default value for **DesignID** is "Site". **SiteBegin** provides the number to use for the first site in the design. The default value for **SiteBegin** is 1.

Four arguments provide information pertaining to the frame. **type.frame** specifies the type of frame, which must be one of following: "finite", "linear", or "area". The default value for **type.frame** is "finite". **src.frame** specifies the source of the frame, which equals "shapefile" if the frame is to be read from an ESRI shapefile, "sp.object" if the frame is to be obtained from an **sp** package object, or "att.frame" if **type.frame** equals “finite” and coordinates for points in the frame are included in argument **att.frame** (see below). The default value for **src.frame** is "shapefile". **in.shape** specifies the name (without any extension) of the input shapefile. If **src.frame** equal "shapefile" and **in.shape** equals NULL, then the shapefile or shapefiles in the current directory are used. The default value for **in.shape** is NULL. **sp.object** specifies the name of the **sp** package object when **src.frame** equals "sp.object". The default value for **sp.object** is NULL.

Six arguments provide information pertaining to the attributes associated with elements in the frame. **att.frame** provides the data frame containing frame attributes, which must contain the columns used for arguments **stratum** (if the sample is stratified) and **mdcaty** (if **mdcaty** is required).

If **src.frame** equals "shapefile" and **att.frame** equals NULL, then **att.frame** is created from the ESRI dbf file(s) in the current directory. The default value for **att.frame** is NULL. **id** provides the name of the column from **att.frame** that identifies the ID value for each element in the frame. If **id** equals NULL, a column named "id" that contains the values 1 through the number of rows in **att.frame** is added to **att.frame**. The default value for **id** is NULL. When **src.frame** equals "att.frame", **xcoord** and **ycoord** provide the names of the columns from **att.frame** that identify x-coordinates and y-coordinates, respectively, for points in the frame. The default values for **xcoord** and **ycoord** are "x" and "y ", respectively. **stratum** provides 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 its elements equal to the name specified for the single element in **design**) is added to **att.frame**. The default value for **stratum** is NULL. **mdcaty** provides the name of the column from **att.frame** that identifies the unequal probability category for each element in the frame. The default value for **mdcaty** is NULL.

For function **grts**, three arguments provide information pertaining to the hierarchical grid that is employed for selecting the GRTS survey design.(see Stevens and Olsen, 2004). **startlev** provides the 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 value for **startlev** is NULL. **maxlev** provides the maximum number of hierarchical levels to use for the GRTS grid, which cannot be greater than 11. The default value for **maxlev** is 11. **shift.grid** controls the 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 value for **shift.grid** is TRUE.

Argument **maxtry** provides the maximum number of iterations for randomly generating a point within a grid cell (for function **grts**) or a point within the frame (for function **irs**) to select a site when **type.frame** equals "area". The default value for **maxtry** is 1000.

For function grts, argument **do.sample** controls the option to select a sample, where TRUE means select a sample and FALSE means return the entire sample frame in reverse hierarchical order. Note that FALSE can only be used when **type.frame** equals "points" and **seltype** equals "Equal". The default value for **do.sample** is TRUE.

Three argument provide information pertaining to output of an ESRI shapefile. **shapefile** controls the option to create an ESRI shapefile containing the survey design information, where TRUE equals create a shapefile and FALSE equals do not create a shapefile. The default value for **shapefile** is TRUE. **prjfilename** provides the name (without any extension) of the ESRI projection file for the input shapefile, which is use to name the projection file for the output shapefile. The default value for **prjfilename** is NULL. **out.shape** provides the name (without any extension) of the output shapefile containing the survey design information. The default value for **out.shape** is "sample".

Second Level Functions (read.dbf, read.shape, sp2shape, and write.dbf)

read.dbf

The sole argument for this function is **filename**, which provides the name of the ESRI dbf file without any extension. If **filename** equals a dbf file name, then that dbf file is read. If **filename** equals NULL, then all of the dbf files in the current directory are read. The default value for **filename** is NULL.

read.shape

The sole argument for this function is **filename**, which provides the name of the ESRI shapefile without any extension. If **filename** equals a shapefile name, then that shapefile is read. If **filename** equals NULL, then all of the shapefiles in the current directory are read. The default value for **filename** is NULL.

sp2shape

This function uses three arguments. **sp.obj** provides the sp package object. **shpfilename** provides the name (without any extension) of the output ESRI shapefile. The default value for **shpfilename** is "temp". **prjfilename** provides the name (without any extension) of the ESRI projection file for the output shapefile. The default value for **prjfilename** is NULL.

write.dbf

This function uses two arguments. **dframe** provides the data frame to be written to the dbf file. **filename** provides the name of the ESRI dbf file without any extension.

Function Output

First Level Functions (grts and irs)

These functions output an sp package object containing the survey design information and any additional attribute variables that were provided. The object is assigned class SpatialPointsDataFrame. For further information regarding the output object, see documentation for the sp package. Optionally, an ESRI shapefile of type point can be created that contains the survey design information.

Second Level Functions (read.dbf, read.shape, sp2shape, and write.dbf)

read.dbf

This function creates a data frame composed of either the contents of the single dbf file, when argument **filename** is provided, or the contents of the dbf file(s) in the current directory, when **filename** is NULL. The data frame is assigned class SurveyFrame.

read.shape

This function creates an **sp** package object containing information in the input ESRI shapefile (or shapefiles). The object is assigned class SpatialPointsDataFrame, SpatialLinesDataFrame, or SpatialPolygonsDataFrame corresponding to the shapefile type, i.e., point, polyline, or polygon, respectively. For further information regarding the output object, see documentation for the **sp** package.

sp2shape

This function creates an ESRI shapefile of type point, polyline, or polygon as determined by the class of argument **sp.obj**.

write.dbf

This function creates the dbf file of an ESRI shapefile.

ANALYSIS FUNCTIONS

Survey Design Options

As mentioned in the introduction, the **spsurvey** analysis functions can accommodate both stratified and unstratified survey designs. When the design is stratified, the stratum code for each site must be provided to the functions. The stratum codes are examined by the functions to ensure that more than one unique stratum code exists, otherwise the data is analyzed as an unstratified design. In addition, when removal of missing values results in a single unique stratum code, the data is analyzed as an unstratified design.

For a two-stage design, the stage one sampling unit (primary sampling unit or cluster) codes must be input to the functions. For a stratified design, the stage one sampling unit codes must identify the stratum code in addition to the stage one sampling unit code within a stratum.

Since **spsurvey** is intended for analysis of probability surveys, the design weight (i.e., inverse of the inclusion probability) for each site must be input to the functions. For a two-stage design, both stage one and stage two weights must be provided. A function (**adjwgt**) is included in **spsurvey** that

adjusts initial survey design weights when survey design implementation results in use of oversample sites or when it is desired to have final weights sum to a known size of the resource.

Since the default choice for variance estimation in **spsurvey** utilizes the x-coordinate and y-coordinate of each site for distance calculations, a function (**geodalbers**) is included in **spsurvey** that converts coordinates measured as latitude and longitude to a coordinate system appropriate for calculation of the distance metric used by the variance estimator, i.e., the Albers projection. The alternate choice for variance estimator does not require coordinates. In addition, for a two-stage design, coordinates are required for both the stage one and stage two sampling units.

Data Analysis Options

The primary functions in **spsurvey** for data analysis are **category.est**, **relrisk.est, attrisk.est, cdf.est**, **cdf.decon**, **cdf.plot**, **cdf.test**, and **total.est** (see the **Descriptions** section). Discussion regarding analysis options for these functions follows.

The user can provide the know size of a resource to the functions in **spsurvey**, which is used to adjust the estimate of a total, e.g., the size of a resource in a set of categories. For a stratified design, the known size of a resource can be provided for each stratum. In addition, the size of a resource, either a known value or an estimated value when a known value is not provided, are used as stratum weights for calculating estimates for a stratified design. The known size of a resource also is used for calculation of finite population and continuous population correction factors (see the discussion that follows).

For a finite resource, size-weighted analysis is accommodated in **spsurvey**. An example of a size-weights is the surface area of each lake in the Northeast region of the U.S., where the set of lakes in the region is treated as a finite resource. The user must provide the size-weight for each sampling unit. For a two-stage design, size-weights are required for the stage one and stage two sampling units. The size-weights are used to scale the design weights for calculation of estimates. The user can provide the known sum of the size-weights for the resource, which is used to adjust the estimate of a total. For a stratified design, the known sum of the size-weights for the resource can be provided for each stratum. In addition, the sum of the size-weights for a resource, either a known value or an estimated value when a known value is not provided, are used as stratum weights for calculating estimates for a stratified design.

Use of finite population and continuous population correction factors in variance estimation is accommodated in **spsurvey**. In order to calculate the factors for a single-stage design, the user must provide the known size of the resource and a support value for each sampling unit, where support is equal to one for a sampling unit from a finite resource and is equal to the size of the sampling unit for an extensive resource. If the single-stage design is stratified, then the known size of the resource must be provided for each stratum. For a two-stage design the user must provide the number of stage one sampling units in the resource, the known size of each stage one sampling unit, and a support value for each stage two sampling unit. If the two-stage design is stratified, then the number of stage one

sampling units in the resource must be provided for each stratum, and the known size of each stage one sampling unit must be identified both with a stratum code and the stage one sampling unit code.

The default choice for variance estimation in **spsurvey** is the local mean variance estimator. The alternate choice for variance estimation is the simple random sampling (SRS) variance estimator, which uses the independent random sample approximation to calculate joint inclusion probabilities. For additional information regarding the local mean variance estimator see Stevens and Olsen (2003).

Descriptions

Analysis functions in **spsurvey** are organized in a hierarchical structure composed of four levels. Functions in the first and second levels are intended for use with a set of response variables and indicators from a probability survey. The first level function creates an object that can be passed to the second level functions. The second level functions organize input and output for analysis and can be called by the user without use of the top level function. The second level functions call the third level functions, which implement data analysis algorithms with support of the fourth level functions. The third level functions can be called by the user for analysis of an individual response variable or indicator. In addition, two of the third level functions (**adjwgt** and **marinus**) are utilized to modify specific survey design variables prior to data analysis. A short description of each function in the top three levels is provided. Functions in the fourth level are not intended for access by the user. Further details regarding the functions are provided in subsequent sections and in the R help documentation for **spsurvey**.

First Level Function

spsurvey.analysis

This function creates an object of class spsurvey.analysis that contains all of the information necessary to use the functions in the **spsurvey** library to analyze data generated by a probability survey. Output from this functions can be passed directly to the second level functions.

Second Level Functions

attrisk.analysis

This function organizes input and output for attributable risk analysis of categorical data generated by a probability survey. Third level function **attrisk.est** is called by this function.

cat.analysis

This function organizes input and output for analysis of categorical data generated by a probability survey. Input can be either an object belonging to class spsurvey.analysis or through use of the other arguments to the function. Third level function **category.est** is called by this function.

cont.analysis

This function organizes input and output for analysis of continuous data generated by a probability survey. Input can be either an object belonging to class spsurvey.analysis or through use of the other arguments to this function. Third level functions **cdf.est**, **cdf.decon**, and **total.est** are called by this function.

cont.cdfplot

This function creates CDF plots using the output object created by function **cont.analysis**. Third level function **cdf.plot** is called by this function.

cont.cdftest

This function organizes input and output for inference regarding CDFs generated by a probability survey. Third level function **cdf.test** is called by this function.

relrisk.analysis

This function organizes input and output for relative risk analysis of categorical data generated by a probability survey. Third level function **relrisk.est** is called by this function.

Third Level Functions

adjwgt

This function adjusts initial survey design weights when implementation results in use of oversample sites or when it is desired to have final weights sum to a known size of the resource. Adjusted weights are equal to initial weight times the frame size divided by the sum of the initial weights. The adjustment is done separately for each weight adjustment category. This function is not called by a second level function.

attrisk.est

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. 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. The standard error of the log of the attributable risk estimate and confidence limits for the estimate also are calculated. This function is called by second level function **attrisk.analysis**.

category.est

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. Standard errors of the category estimates and confidence bounds are calculated. This function is called by second level function **cat.analysis**.

cdf.est

This function calculates an estimate of the 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, percentiles are estimated. Standard errors of the CDF and percentile estimates and confidence bounds are calculated. This function is called by second level function **cont.analysis**.

cdf.plot

This function creates a CDF plot. Input data for the plot 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. This function is called by second level function **cont.cdfplot**.

cdf.decon

This function calculates an estimate of the deconvoluted 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, percentiles are estimated. Standard errors of the CDF and percentile estimates and confidence bounds are calculated. This function is called by second level function **cont.analysis**.

cdf.test

This function calculates the Wald, Rao-Scott first order corrected (mean eigenvalue corrected), and Rao-Scott second order corrected (Satterthwaite corrected) statistics for categorical data to test for differences between two CDFs (Kincaid, 2004). The functions calculates both standard versions of those three statistics, which are distributed as chi-squared random variables, plus modified version of the statistics, which are distributed as F random variables. This function is not called by a second level function.

geodalbers

This function converts x-coordinates and y-coordinates measured in units of latitude and longitude, i.e., geographic coordinates measured in decimal degrees, to coordinates in an equal-area, conic map projection measured in units of kilometers. The map projection is named after Heinrich Albers. This function is not called by a second level function.

read.sas

This function reads either a SAS dataset or a SAS XPORT (transport) file and creates a data frame. This function is not called by a second level function.

relrisk.est

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 standard error of the log of the relative risk estimate and confidence limits for the estimate also are calculated. This function is called by second level function **relrisk.analysis**.

total.est

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, standard errors of the population estimates and confidence bounds are calculated. This function is called by second level function **cont.analysis**.

write.object

This function writes the contents of an object, which may be either a data frame or a matrix, to a plot. This function is not called by a second level function.

Data Input

Overview

Although the first level function provides the most flexibility, data entry is similar for the first and second level functions. Arguments to the first and second level functions provide information for the following categories: (1) sites to be included in the analysis, (2) identification of sets of populations

and subpopulations, (3) survey design variables, (4) response variables, and (5) additional variables specifying analytical options. As necessary, site IDs are used to connect the various arguments. An extensive description follows regarding data entry for the first level function. For the second level functions, differences in data entry from those described for the first level function are noted. Data entry for the third level functions is not described. For first, second, and third level functions, arguments are checked for errors and for compatibility of input values. In addition, for arguments indexed by site IDs, missing values are removed from the argument, and corresponding values are removed from all other arguments indexed by site IDs.

First Level Function (spsurvey.analysis)

Information regarding sites to be included in the analysis is provided by argument **sites**, which is a 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 it 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 value for **sites** is NULL.

Information identifying sets of populations and subpopulations for which estimates will be calculated is provided by argument **subpop**, which is a data frame. The first variable in **subpop** is site IDs, 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, when subpopulations are present, or provides a value that is identical for all sites, when subpopulations are not present. If this data frame is not provided, then it 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 value for **subpop** is NULL.

Information regarding survey design variables is provided by argument **design**, which is a data frame, or by the individual design variable arguments to the function. Individual design variables may be provided as a vector of values or as a formula, where the formulas are interpreted using the **design** data frame. If **design** is not provided, then it will be created from the values for the individual design variables in the argument list. The default value for **design** is NULL. If values for the individual variables are not provided, then the variables in **design** should be named as follows: (1) siteID – site IDs; (2) wgt – final adjusted weights, which are either the weights for a single-stage sample or the stage two weights for a two-stage sample; (3) xcoord – the 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; (4) ycoord – the 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; (5) stratum – the stratum codes; (6) cluster – the stage one sampling unit codes; (7) wgt1 – the final adjusted stage one weights; (8) xcoord1 – the stage one x-coordinates for location; and (9) ycoord1 – the stage one y-coordinates for location. Names of the nine individual design variable arguments are the same as the default names for the variables in the **design** data frame. Using formulas to input design variables allow the user to supply names for those

variables rather than using the default names. Values always are required for design variables **siteID** and **wgt**. Values for **xcoord** and **ycoord** are required when using the local mean variance estimator (see the discussion for argument **vartype**), but are not required for the SRS variance estimator. If a stratified sampling design is used, than values must be provided for design variable **stratum**. Similarly, if a two-stage sampling design was used, than values must be provided for design variables **cluster**, **wgt1**, **xcoord1**, and **ycoord1.** The default value for the **design** data frame and for the individual design variable arguments is NULL.

Information regarding categorical response variables is provided by **data.cat,** which is a data frame. The first variable in **data.cat** is site IDs, and subsequent variables are response variables. Missing data (NA) is allowed in **data.cat**. The default value for **data.cat** is NULL.

Information regarding continuous response variables is provided by **data.cont**, which is a data frame. The first variable in **data.cont** is site IDs, and subsequent variables are response variables. Missing data is allowed. The default value for **data.cat** is NULL.

Other arguments to the functions provide information required for optional analyses. Arguments **sigma** and **var.sigma** provide information for CDF deconvolution, where **sigma** is a vector of measurement error variance values, and **var.sigma** is a vector of variances for the measurement error variance values. When **sigma** is provided, it is not necessary to provide **var.sigma**, in which case **sigma** is treated as a known quantity, and variability of the deconvolution procedure that is due to estimating **sigma** is ignored. Both **sigma** and **var.sigma** must have the names attribute set to identify the continuous response variable names. Missing data is allowed. The default value for **sigma** and **var.sigma** is NULL.

Information regarding know size of the resource or know sum of size-weights for the resource is provided by **popsize**. This argument must be in the form of a list containing an entry for each Type of population in the **subpop** data frame, where NULL is a valid entry for a population Type. The list must be named using the column names for 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 for **popsize** is NULL.

An example of **popsize** for a stratified sample follows:

```
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)
```

An example of **popsize** for an unstratified sample follows:

```
popsize = list("Pop 1"=1500, "Pop 2"=list("SubPop 1"=750, "SubPop 2"=500, "SubPop 3"=375),
    "Pop 3"=NULL)}
```

Information required for calculation of finite and continuous population correction factors is provided by arguments **pcfsize**, **N.cluster**, **stage1size**, and **support**. Argument **pcfsize** is applicable to single-stage designs. Arguments **N.cluster** and **stage1size** are applicable to two-stage sampling designs. Argument **pcfsize** provides the resource size. For a stratified sample, **pcfsize** must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. **N.cluster** provides the number of stage one sampling units in the resource. For a stratified sample, **N.cluster** must be a vector containing a value for each stratum and must have the names attribute set to identify the stratum codes. Argument **stage1size** is a vector containing the known size of each stage one sampling unit and must have the names attribute set to identify the stage one sampling unit codes. For a stratified sample, the names attribute for **stage1size** 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". Argument **support** provides the support value for each site and is always required for calculation of population correction factors. For a sampling unit from a finite resource**, support** is a vector of ones; and for an extensive resource, it is a vector containing the size of the sampling unit associated with each site.

Argument **vartype** controls the choice of variance estimator, where "Local" indicates the local mean variance estimator and "SRS" indicates the SRS estimator. The default value for **vartype** is "Local".

Argument **conf** provides the confidence level that prescribes the Normal distribution multiplier used in calculating confidence bounds. The default value for **conf** is 95%.

Argument **pctval** provides the set of values at which percentiles are estimated by functions **cdf.est** and **cdf.decon**. The default set of values for **pctval** is: 5, 25, 50, 75, and 95.

Second Level Functions (attrisk.analysis, cat.analysis, cont.analysis, cont.cdfplot, cont.cdftest, and relrisk.analysis)

Data input for second levels functions **cat.analysis**, **cont.analysis**, and **cont.cdftest** can be either an object belonging to class spsurvey.analysis, i.e., output from the first level function **spsurvey.analysis**, or through use of the other arguments to these functions. For functions **attrisk.analysis and relrisk.analysis**, data input using an object belonging to class spsurvey.analysis is not available. When data input is not accomplished through use of an object belonging to class spsurvey.analysis, format for data entry is similar to the first level function. When an object of class spsurvey.analysis is not provided, then values must be supplied for the **sites**, **subpop**, and **design** data frames plus either the **data.ar** data frame for function **attrisk.analysis,** the **data.cat** data frame and the

type.cat vector for function **cat.analysis**, the **data.cont** data frame for functions **cont.analysis** and **cont.cdftest**, or the **data.rr** data frame for function **relrisk.analysis**,. Unlike the first level function, individual design variables cannot be input to the second level function, which means that only the default names are allowed in the **design** data frame, and design variable cannot be input using formulas. The following arguments that were discussed previously can be input to the second level functions: **popsize**, **pcfsize,N.cluster**, **stage1size**, **support**, **swgt**, **swgt1**, **vartype**, **conf**, and **pctval**. For function **cont.analysis**, values for arguments **sigma** and **var.sigma** can be input to the function. For function **relrisk.analysis**, values for arguments **response.var** and **stressor.var** must be input to the function, and values for arguments **response.levels** and **stressor.levels** can be provided.

Data input for function **cont.cdfplot** 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**. Input options include the ability to specify either continuous or ordinal data and the type of units in which the CDF is plotted (either percent, measurement units, or both).

Third Level Functions

Regarding data entry for the third level functions, consult the entry for each function in the R help documentation for **spsurvey**.

Analysis Algorithms

Data analysis algorithms are carried out by the third level functions with support of the fourth level functions. Descriptions follows for each type of data analysis carried out by the functions in **spsurvey**. In addition, discussion is provided regarding issues that are common to each type of data analysis.

Attributable Risk (attrisk.est)

Estimation of attributable risk is carried out by function **attrisk.est**. The attributable risk estimate is computed using marginal totals from a 2x2 table of cell counts defined by a categorical response variable and a categorical stressor variable. Marginal totals are estimated using the Horvitz-Thompson estimator. The standard error of the log of the attributable risk estimate is calculated using a first-order Taylor series linearization (Sarndal *et al.*, 1992).

Categorical Data Analysis (category.est)

Categorical data analysis is carried out by function **category.est**. 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 a category. The denominator of the ratio estimates the size of the resource. 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 size estimates, where that estimator is the product of the known value and the Horvitz-Thompson ratio estimator. 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 size estimates.

CDF and Percentiles Estimation (cdf.est and cdf.decon)

Function **cdf.est** carries out CDF and percentile estimation. Function **cdf.decon** carries out estimation of the deconvoluted CDF and percentile based on the deconvoluted CDF. The simulation extrapolation deconvolution method (Stefanski and Bay, 1996) is used to remove the effect of measurement error variance from the CDF of the response variable. When function **cdf.est** or **cdf.decon** is called directly, 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 is used to calculate the CDF estimate. 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. 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. In addition, the functions use the estimated CDF to calculate percentile estimates and approximate confidence bounds for the percentile estimates.

CDF Inference (cdf.test)

Function **cdf.test** carries out inference regarding the difference between two CDFs. The user supplies the set of upper bounds for defining the classes for the CDFs. The Horvitz-Thompson ratio estimator is used to calculate estimates of the class proportions for the CDFs. Note that function **cdf.test** currently is not written to handle either stratified designs or two-stage designs.

Population Total, Mean, Variance, and Standard Deviation Estimation (total.est)

Estimation of the population total, mean, variance, and standard deviation is carried out by function **total.est**. The Horvitz-Thompson estimator is used to calculate the total, variance, and standard deviation estimates. The Horvitz-Thompson ratio estimator is used to calculate the mean estimate.

Relative Risk (relrisk.est)

Estimation of relative risk is carried out by function **relrisk.est**. The relative risk estimate is computed using the ratio of a numerator probability to a denominator probability, which 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 is calculated using a first-order Taylor series linearization (Sarndal *et al.*, 1992).

Analysis of Stratified Designs

For a stratified design, 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 design, 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. For a stratified design 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.

Analysis of Two-Stage Designs

For a two-stage design, both stages must be accommodated for calculating estimates and standard errors. For calculation of estimates, the product of the stage one and stage two weights is utilized in the estimation process. For estimation of standard errors, the total and variance of the total is calculated for each stage one sampling unit, where the stage two weights are used in the estimation process. Next, variance of the stage one sampling unit totals is calculated using the stage one weights in the estimation process. Then the weighted sum of the estimated variance of the stage one sampling unit totals is calculated using the stage one weights. The standard error estimate is obtained by adding the estimated variance of the stage one sampling unit totals and the weighted sum of the estimated variance of the stage one sampling unit totals. Depending upon the quantity being estimated, e.g., a proportion estimate, the standard error estimate is scaled by an appropriate factor.

Function Output

First Level Function (spsurvey.analysis)

This function outputs a 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 list may be NULL. The output list is composed of the following elements: (1) the **sites** data frame; (2) the **subpop** data frame; (3) the **design** data frame; (4) the **data.cat** data frame;(5) the **data.cont** data frame; (6) **sigma** - measurement error variance; (7) **var.sigma** - variance of the estimated measurement error variance; (8) **stratum.ind** – a logical value that indicates whether the sample is stratified, where TRUE indicates a stratified sample and FALSE indicates not a stratified sample; (9) **cluster.ind** – a logical value that indicates whether the sample is a two-stage sample, where TRUE indicates a two-stage sample and FALSE indicates not a two-stage sample; (10) **popsize** – known size of the resource or known sum of size-weights of the resource for use in ratio adjustment of estimators;(11) **pcfactor.ind** – a logical

value that indicates whether the population correction factor will be used during variance estimation, where TRUE indicates use the population correction factor and FALSE indicates do not use the factor; (12) **pcfsize** –size of the resource for use in calculation of finite and continuous population correction factors; (13) **N.cluster** – the number of stage one sampling units in the resource; (14) **stage1size** – the known size of the stage one sampling units; (15) **swgt.ind** – a logical value that indicates whether the sample is a size-weighted sample, where TRUE indicates a size-weighted sample and FALSE indicates not a size-weighted sample; (16) **vartype** – the choice of variance estimator; (17) **conf** – the confidence level; and (18) **pctval** –the set of values at which percentiles are estimated.

Second Level Functions (attrisk.analysis, cat.analysis, cont.analysis, cont.cdfplot, cont.cdftest, and relrisk.analysis)

Function **attrisk.analysis** outputs 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.

Function **cat.analysis** outputs a data frame of population estimates for all combinations of subpopulation Types, subpopulations within Types, response variables, and categories within each response variable. The data frame provides estimates for proportion and size of the categories. Standard error estimates and confidence interval estimates also are included.

Function **cont.analysis** outputs 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 input values of measurement error variance are provided to the function. CDF and percentile estimates are calculated for both proportion and size of the population. Standard error estimates and confidence interval estimates also are calculated. The five data frames are: (1) **CDF** – a data frame containing the CDF estimates, (2) **Pct** – a data frame containing the percentile estimates plus the population total, mean, standard deviation, and variance estimates, (3) **CDF.D** – a data frame containing the deconvoluted CDF estimates, and (4) **Pct.D** – a data frame containing the deconvoluted percentile estimates.

Function **cont.cdfplot** creates plots for every combination of Type of population, subpopulation within Type, and response variable. Output from the function is placed in a PDF file.

Function **cont.cdftest** outputs a data frame containing the results of testing for differences between CDFs for every pair of subpopulations within each Type of population and each response variable. Degrees of freedom and p values also are provided.

Function **relrisk.analysis** outputs a 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.

Third Level Functions

Regarding output for the third level functions, consult the entry for each function in the R help documentation for **spsurvey**.

References

Diaz-Ramos, S., D.L. Stevens, Jr., and A.R. Olsen. 1995. EMAP Statistics Methods Manual. EPA/620/R-96/002, U.S. Environmental Protection Agency, National Health and Environmental Effects Research Laboratory, Corvallis, Oregon.

Kincaid, T.M. 2000. Testing for differences between cumulative distribution functions from complex environmental sampling surveys. *2000 Proceedings of the Section on Statistics and the Environment*, pp. 39 - 44. American Statistical Association, Alexandria, VA.

Messer, J.J ., R. A. Linthurst, and W. S. Overton. 1991. An EPA program for monitoring ecological status and trends. Environmental Monitoring and Assessment 17:67-78

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

Stefanski, L.A. and J.M. Bay. 1996. Simulation extrapolation deconvolution of finite population cumulative distribution function estimators. Biometrika 83: 496-517.

Stevens, D.L., Jr., and Olsen, A.R. 2003. Variance estimation for spatially balanced samples of environmental resources. Environmetrics 14: 593-610.

Stevens, D.L., Jr. and Olsen, A.R. 2004. Spatially-balanced sampling of natural resources. Journal of the American Statistical Association 99: 262-278.