Analysis of a GRTS Survey Design for an Area Resource

Thomas Kincaid

May 23, 2011

Contents

This document presents analysis of a GRTS survey design for an area resource. The area resource used in the analysis is estuaries in South Carolina. Although a stratified survey design was used to sample estuaries, analyses will be conducted as if the design was unstratified. Instead, strata will be used to define subpopulations for analysis. The strata employed in the survey were: (1) open water and (2) tidal creeks. The analysis will include calculation of three types of population estimates: (1) estimation of proportion and size (area of estuaries) for site evaluation status categorical variables; (2) estimation of proportion and size for estuary condition categorical variables; and (3) estimation of the cumulative distribution function (CDF) and percentiles for quantitative variables. Testing for difference between CDFs from subpopulations also will be presented.

1 Preliminaries

The initial step is to use the library function to load the spsurvey package. After the package is loaded, a message is printed to the R console indicating that the spsurvey package was loaded successfully.

Load the spsurvey package.

```
> library(spsurvey)
```

Version 2.2 of the spsurvey package was loaded successfully.

Version 2.2 of the spsurvey package was loaded successfully.

2   Read the survey design and analytical variables data file

The next step is to read the data file, which includes both survey design variables and analytical variables. The read.delim function is used to read the tab-delimited file and assign it to a data frame named SC_estuaries. The nrow function is used to determine the number of rows in the SC_estuaries data frame, and the resulting value is assigned to an object named nr. Finally, the initial six lines and the final six lines in the SC_estuaries data frame are printed using the head and tail functions, respectively.

Read the survey design and analytical variables data file.

```
> SC_estuaries <- read.delim("SC_estuaries.tab")
> nr <- nrow(SC_estuaries)
```

Display the initial six lines in the data file.

```
> head(SC_estuaries)
```

```
      siteID  xcoord  ycoord      wgt    stratum  status IBI_status IBI_score
1 EEOW00-001 1549320 1263020 10.47515 Open Water Sampled       Good       3.5
2 EEOW00-002 1487548 1226750 10.47515 Open Water Sampled       Good       4.0
3 EEOW00-003 1442831 1159768 10.47515 Open Water Sampled       Good       4.0
4 EEOW00-004 1425151 1148860 10.47515 Open Water Sampled       Good       4.5
5 EEOW00-005 1432172 1140588 10.47515 Open Water Sampled       Good       4.5
6 EEOW00-006 1540551 1280557 10.47515 Open Water Sampled   Marginal       2.5
  WQ_status WQ_score
1      Good      4.3
2      Good      4.6
3      Good      5.0
4      Good      5.0
5      Good      5.0
6      Good      4.2
```

Display the final six lines in the data file.

```
> tail(SC_estuaries)
```

```
        siteID  xcoord  ycoord     wgt     stratum    status IBI_status
130 EETC99-035 1441835 1151395 1.41106 Tidal Creek NonTarget       <NA>
131 EETC99-036 1535449 1247775 1.41106 Tidal Creek   Sampled       Good
132 EETC99-037 1500880 1225190 1.41106 Tidal Creek   Sampled   Marginal
133 EETC99-038 1440733 1147398 1.41106 Tidal Creek   Sampled       Good
134 EETC99-039 1468504 1179279 1.41106 Tidal Creek   Sampled       Good
135 EETC99-040 1430671 1151686 1.41106 Tidal Creek   Sampled   Marginal
    IBI_score WQ_status WQ_score
130        NA      <NA>       NA
131       3.5      Good      4.3
132       2.5      Poor      3.0
133       3.0  Marginal      3.7
134       4.0      Good      4.3
135       2.5  Marginal      3.7
```

The sample of estuaries in South Carolina is displayed in Figure 1. The sample sites for each stratum are displayed using a unique color. First, the levels function is used to extract the set of unique strata values, and the result is assigned to object stratum. Next, the rainbow function is called to select a set of two colors, and the result is assigned to object cols. The plot function is then used to produce the basic figure, but plotting of sample points is suppressed. The for function is used to loop through the set of two unique strata values and plot the color-coded points for each stratum using the points function. Finally, the legend function is used to add a legend to the figure, and the title function is used to create a figure title.

```
> stratum <- levels(SC_estuaries$stratum)
> cols <- rainbow(2)
> plot(SC_estuaries$xcoord, SC_estuaries$ycoord, type="n", xlab="x-coordinate",
+       ylab="y-coordinate")
> for(i in 1:2) {
+     ind <- SC_estuaries$stratum == stratum[i]
+     points(SC_estuaries$xcoord[ind], SC_estuaries$ycoord[ind], pch=20,
+           col=cols[i])
+     }
> legend(x="topright", inset=0.05, legend=stratum, pch=20, cex=0.8, col=cols)
> title("Plot of South Carolina Estuary Sites Color-Coded by Stratum")
```

3   Analysis of site status evaluation variables

The first analysis that will be examined is calculation of extent estimates for a site status evaluation variable. Extent is measured both by the proportion of the resource in status evaluation categories and by size of the resource in each category. For an area resource like estuaries, size refers to the area of estuaries in a category. For calculating extent estimates (and for all of the analyses we will consider), the survey design weights are incorporated into the calculation process. The site status variable named status will be examined, which classifies estuaries into two evaluation categories: "Sampled" and "NonTarget". The table and addmargins functions are used to create tables displaying the count for each code (level) of the status variable.

```
> addmargins(table(SC_estuaries$status))
```

A table displaying the number of values for each level of the status
variable follows:

```
NonTarget   Sampled      Sum
       19       116      135
```

The cat.analysis function in the spsurvey package will be used to calculate extent estimates. Four data frames constitute the primary input to the cat.analysis function. The first column (variable) in the four data frames provides the unique identifier (site ID) for each sample site and is used to connect records among the data frames. The siteID variable in the SC_estuaries data frame is assigned to the siteID variable in the data frames. The four data frames that will be created are named as follows: sites, subpop, design, and data.cat. The sites data frame identifies sites to use in the analysis and contains two variables: (1) siteID - site ID values and (2) Use - a logical vector indicating which sites to use in the analysis. The rep (repeat) function is used to
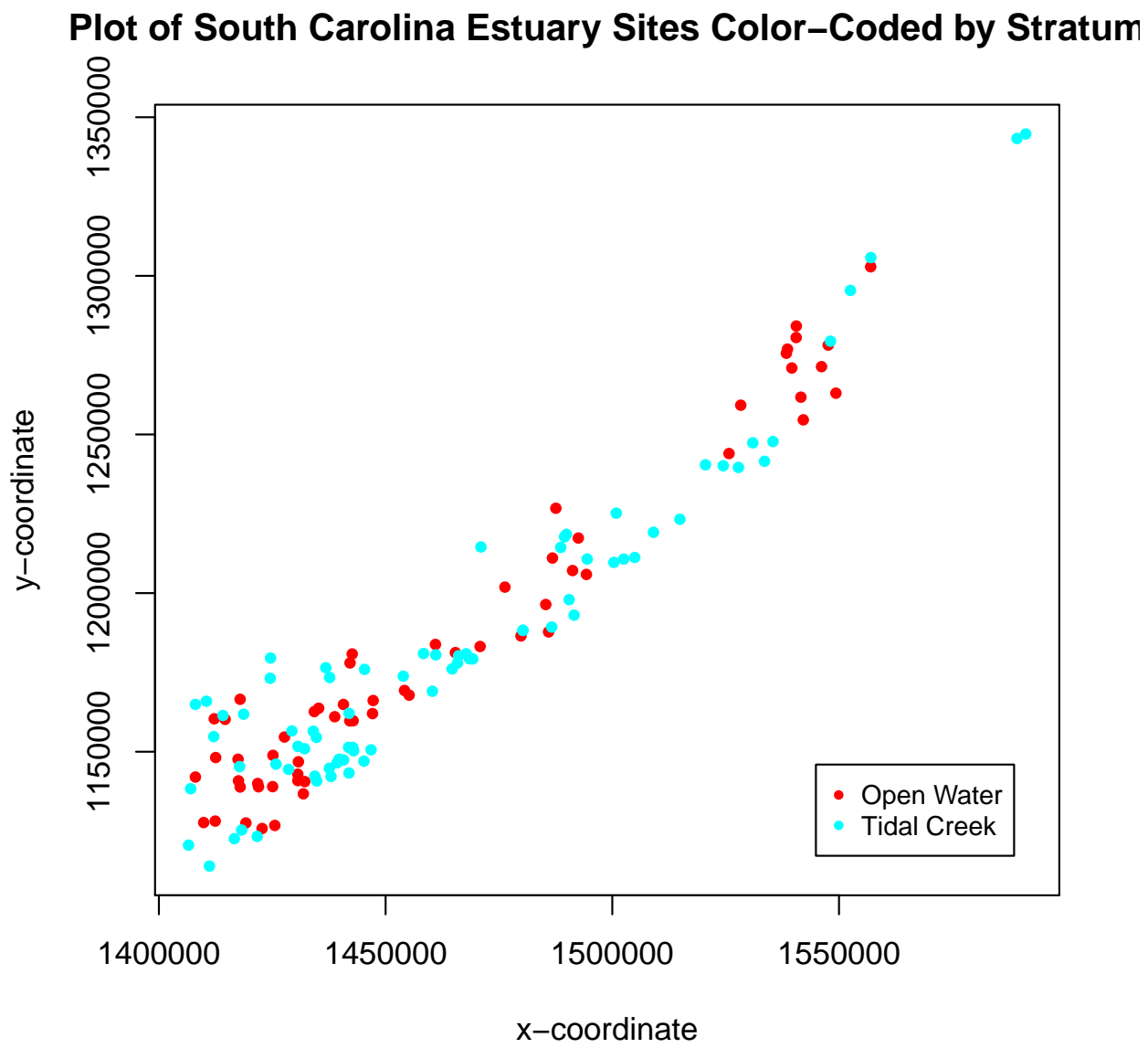
3

Figure 1: South Carolina Estuary Sample Sites.

assign the value TRUE to each element of the Use variable. Recall that nr is an object containing the number of rows in the SC_estuaries data frame. The subpop data frame defines populations and, optionally, subpopulations for which estimates are desired. Unlike the sites and design data frames, the subpop data frame can contain an arbitrary number of columns. The first variable in the subpop data frame identifies site ID values 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 a character value. If the number of unique values for a type variable is greater than one, then the set of values represent subpopulations of that type. When a type variable consists of a single unique value, then the type does not contain subpopulations. For this analysis, the subpop data frame contains three variables: (1) siteID - site ID values, (2) All_Estuaries - which will be used to calculate estimates for all of the sample sites combined, and (3) Estuary_Type - which will be used to calculate estimates for each stratum individually. The stratum variable in the SC_estuaries data frame is assigned to the Estuary_Type variable in the subpop data frame. The design data frame consists of survey design variables. For the analysis under consideration, the design data frame contains the following variables: (1) siteID - site ID values; (2) wgt - final, adjusted, survey design weights; (3) xcoord - x-coordinates for location; and (4) ycoord - y-coordinates for location. The wgt, xcoord, and ycoord variables in the design data frame are assigned values using variables with the same names in the SC_estuaries data frame. Like the subpop data frame, the data.cat data frame can contain an arbitrary number of columns. The first variable in the data.cat data frame identifies site ID values and each subsequent variable identifies a response variable. The response variable is Status, which is assigned the status variable in the SC_estuaries data frame. Missing data (NA) is allowed for the response variables, which are the only variables in the input data frames for which NA values are allowed.

Create the sites data frame.

```
> sites <- data.frame(siteID=SC_estuaries$siteID,
+                     Use=rep(TRUE, nr))
```

Create the subpop data frame.

```
> subpop <- data.frame(siteID=SC_estuaries$siteID,
+                      All_Estuaries=rep("All Estuaries", nr),
+                      Estuary_Type=SC_estuaries$stratum)
```

Create the design data frame.

```
> design <- data.frame(siteID=SC_estuaries$siteID,
+                      wgt=SC_estuaries$wgt,
+                      xcoord=SC_estuaries$xcoord,
+                      ycoord=SC_estuaries$ycoord)
```

Create the data.cat data frame.

```
> data.cat <- data.frame(siteID=SC_estuaries$siteID,
+                        Status=SC_estuaries$status)
```

Use the cat.analysis function to calculate extent estimates for the site status evaluation variables.

```
> Extent_Estimates <- cat.analysis(sites, subpop, design, data.cat)
```

The extent estimates are displayed using the print function. The object produced by cat.analysis is a data frame containing thirteen columns. The first five columns identify the population (Type), subpopulation (Subpopulation), response variable (Indicator), levels of the response variable (Category), and number of values in a category (NResp). A category labeled "Total" is included for each combination of population, subpopulation, and response variable. The next four columns in cat.analysis provide results for the proportion estimates: the proportion estimate ((Estimate.P), standard error of the estimate (StdError.P), lower confidence bound (LCB95Pct.P), and upper confidence bound (UCB95Pct.P). Argument conf for cat.analysis allows control of the confidence bound level. The default value for conf is 95, hence the column names for confidence bounds contain the value 95. Supplying a different value to the conf argument will be reflected in the confidence bound names. Confidence bounds are obtained using the standard error and the Normal distribution multiplier corresponding to the confidence level. The final four columns in cat.analysis provide results for the size (units) estimates: the units estimate (Estimate.U), standard error of the estimate (StdError.U), lower confidence bound (LCB95Pct.U), and upper confidence bound (UCB95Pct.U).

```
> print(Extent_Estimates)
```

	Type	Subpopulation	Indicator	Category	NResp	Estimate.P	StdError.P
1	All_Estuaries	All Estuaries	Status	NonTarget	19	4.885244	1.335908
2	All_Estuaries	All Estuaries	Status	Sampled	116	95.114756	1.335908
3	All_Estuaries	All Estuaries	Status	Total	135	100.000000	0.000000
4	Estuary_Type	Open Water	Status	NonTarget	1	1.666667	1.443963
5	Estuary_Type	Open Water	Status	Sampled	59	98.333333	1.443963
6	Estuary_Type	Open Water	Status	Total	60	100.000000	0.000000
7	Estuary_Type	Tidal Creek	Status	NonTarget	18	24.000000	3.904808
8	Estuary_Type	Tidal Creek	Status	Sampled	57	76.000000	3.904808
9	Estuary_Type	Tidal Creek	Status	Total	75	100.000000	0.000000

	LCB95Pct.P	UCB95Pct.P	Estimate.U	StdError.U	LCB95Pct.U	UCB95Pct.U
1	2.266912	7.503575	35.87424	9.445802e+00	17.36081	54.38767
2	92.496425	97.733088	698.46458	4.380900e+01	612.60052	784.32864
3	100.000000	100.000000	734.33882	4.210738e+01	651.80986	816.86778
4	0.000000	4.496782	10.47515	9.075442e+00	0.00000	28.26270
5	95.503218	100.000000	618.03414	9.075442e+00	600.24660	635.82168
6	100.000000	100.000000	628.50930	2.902591e-14	628.50930	628.50930
7	16.346718	31.653282	25.39909	4.132439e+00	17.29965	33.49852
8	68.346718	83.653282	80.43044	4.132439e+00	72.33100	88.52987
9	100.000000	100.000000	105.82952	4.517939e-15	105.82952	105.82952

The write.table function is used to store the extent estimates as a comma-separated value (csv) file. Files in csv format can be read by programs such as Microsoft Excel.

```
> write.table(Extent_Estimates, file="Extent_Estimates.csv", sep=",",
+             row.names=FALSE)
```

4   Analysis of estuary condition variables

The second analysis that will be examined is estimating resource proportion and size for estuary condition variables. Two estuary condition variables will be examined: (1) IBI_Status, which

classifies estuaries by benthic IBI (index of biotic integrity) status categories and (2) WQ_Status, which classifies estuaries by WQ (water quality) status categories. The table and addmargins functions are used to create tables displaying the count for each level of the two estuary condition variables.

```
> addmargins(table(SC_estuaries$IBI_status))
```

A table displaying the number of values for each level of the IBI status variable follows:

```
    Good Marginal     Poor      Sum
      99       14        3      116
```

```
> addmargins(table(SC_estuaries$WQ_status))
```

A table displaying the number of values for each level of the WQ status variable follows:

```
    Good Marginal     Poor      Sum
      83       29        4      116
```

As for extent estimates, the cat.analysis function will be used to calculate condition estimates. The sites data frame for this analysis differs from the one used to calculate extent estimates. The Use logical variables in sites is set equal to the value "Sampled", so that only sampled sites are used in the analysis. The subpop and design data frames created in the prior analysis can be reused for this analysis. The data.cat data frame contains the two estuary condition variables: IBI_Status and WQ_Status. Variables IBI_status and WQ_status in the SC_estuaries data frame are assigned to IBI_Status and WQ_Status, respectively.

Create the sites data frame.

```
> sites <- data.frame(siteID=SC_estuaries$siteID,
+                     Use=SC_estuaries$status == "Sampled")
```

Create the data.cat data frame.

```
> data.cat <- data.frame(siteID=SC_estuaries$siteID,
+                     IBI_Status=SC_estuaries$IBI_status,
+                     WQ_Status=SC_estuaries$WQ_status)
```

Use the cat.analysis function to calculate estimates for the estuary condition variables.

```
> Condition_Estimates <- cat.analysis(sites, subpop, design, data.cat)
```

Print the estuary condition estimates for all sites combined.

```
> print(Condition_Estimates[c(1:3, 16:19), ])
```

```
           Type Subpopulation  Indicator Category NResp Estimate.P StdError.P
1  All_Estuaries All Estuaries IBI_Status     Good    99  86.183869   3.063318
2  All_Estuaries All Estuaries IBI_Status Marginal    14  11.912344   3.111665
3  All_Estuaries All Estuaries IBI_Status     Poor     3   1.903787   1.349921
16 All_Estuaries All Estuaries  WQ_Status    Total   116 100.000000   0.000000
```

7

```
17   Estuary_Type    Open Water  WQ_Status      Good   51  86.440678   4.011639
18   Estuary_Type    Open Water  WQ_Status Marginal    8  13.559322   4.011639
19   Estuary_Type    Open Water  WQ_Status     Total   59 100.000000   0.000000
   LCB95Pct.P UCB95Pct.P Estimate.U   StdError.U LCB95Pct.U UCB95Pct.U
1    80.179876  92.187861  601.96380 4.147219e+01  520.67980  683.24780
2     5.813592  18.011097   83.20351 2.217377e+01   39.74371  126.66330
3     0.000000   4.549583   13.29728 9.383494e+00    0.00000   31.68858
16 100.000000 100.000000  698.46458 3.977348e+01  620.50999  776.41917
17   78.578010  94.303346  534.23290 2.479330e+01  485.63893  582.82688
18    5.696654  21.421990   83.80124 2.479330e+01   35.20726  132.39521
19 100.000000 100.000000  618.03414 3.657743e-14  618.03414  618.03414
```

Use the write.table function to write the condition estimates as a csv file.

```
> write.table(Condition_Estimates, file="Condition_Estimates.csv", sep=",",
+             row.names=FALSE)
```

5   Analysis of estuary condition variables correcting for population size

The frame is a data structure containing spatial location data in addition to other attributes regarding a resource of interest and is used to create a survey design. A frame often takes the form of a shapefile. The frame can be used to obtain size values (e.g., area of estuaries) for the populations and subpopulations examined in an analysis. Examination of the Estimates.U column in the Condition_Estimates data frame produced by cat.analysis reveals that the estimated Total value for both condition variables and each combination of population value and subpopulation value does not sum to the corresponding frame size value. For example, the Total entry in the Estimate.U column for the IBI_status variable, population "All_Estuaries" and subpopulation "All Estuaries" is 698 square kilometers (rounded to a whole number). The corresponding frame size value is 734 square kilometers. The popsize (population size) argument to cat.analysis provides a mechanism for forcing the Total category to equal a desired value. First, the c (combine) function is used to create a named vector of frame size values for each basin. Output from the c function is assigned to an object named framesize. The popsize argument is a list, which is a particular type of R object. The popsize list must include an entry for each population type included in the subpop data frame, i.e., All_Estuaries and Estuary_Type for this analysis. The sum function applied to framesize is assigned to the All_Estuaries entry in the popsize list. Recall that the Estuary_Type population contains subpopulations, i.e., stratum categories. When a population type contains subpopulations, the entry in the popsize list also is a list. The as.list function is applied to framesize, and the result is assigned to the Estuary_Type entry in the popsize list.

Assign frame size values.

```
> framesize <- c("Open Water"=628.509298, "Tidal Creek"=105.829522)
```

Use the cat.analysis function to calculate estimates for the estuary condition variables.

```
> Condition_Estimates_popsize <- cat.analysis(sites, subpop, design, data.cat,
+   popsize=list(All_Estuaries=sum(framesize),
+             Estuary_Type=as.list(framesize)))
```

Print the estuary condition estimates for IBI status.

```
> print(Condition_Estimates_popsize[1:12, ])
```

	Type	Subpopulation	Indicator	Category	NResp	Estimate.P	StdError.P
1	All_Estuaries	All Estuaries	IBI_Status	Good	99	86.183869	3.063318
2	All_Estuaries	All Estuaries	IBI_Status	Marginal	14	11.912344	3.111665
3	All_Estuaries	All Estuaries	IBI_Status	Poor	3	1.903787	1.349921
4	All_Estuaries	All Estuaries	IBI_Status	Total	116	100.000000	NA
5	Estuary_Type	Open Water	IBI_Status	Good	51	86.440678	3.461229
6	Estuary_Type	Open Water	IBI_Status	Marginal	7	11.864407	3.526629
7	Estuary_Type	Open Water	IBI_Status	Poor	1	1.694915	1.528238
8	Estuary_Type	Open Water	IBI_Status	Total	59	100.000000	NA
9	Estuary_Type	Tidal Creek	IBI_Status	Good	48	84.210526	4.344105
10	Estuary_Type	Tidal Creek	IBI_Status	Marginal	7	12.280702	3.959455
11	Estuary_Type	Tidal Creek	IBI_Status	Poor	2	3.508772	2.090309
12	Estuary_Type	Tidal Creek	IBI_Status	Total	57	100.000000	NA

	LCB95Pct.P	UCB95Pct.P	Estimate.U	StdError.U	LCB95Pct.U	UCB95Pct.U
1	80.179876	92.187861	632.881606	22.495132	588.791958	676.971253
2	5.813592	18.011097	87.476970	22.850167	42.691465	132.262475
3	0.000000	4.549583	13.980245	9.912994	0.000000	33.409355
4	NA	NA	734.338820	NA	NA	NA
5	79.656794	93.224562	543.287698	21.754147	500.650354	585.925042
6	4.952341	18.776473	74.568900	22.165192	31.125922	118.011878
7	0.000000	4.690206	10.652700	9.605117	0.000000	29.478383
8	NA	NA	628.509298	NA	NA	NA
9	75.696237	92.724816	89.119597	4.597346	80.108966	98.130229
10	4.520312	20.041092	12.996608	4.190273	4.783824	21.209392
11	0.000000	7.605702	3.713317	2.212164	0.000000	8.049078
12	NA	NA	105.829522	NA	NA	NA

Use the write.table function to write the condition estimates as a csv file.

```
> write.table(Condition_Estimates_popsize, file="Condition_Estimates_popsize.csv",
+             sep=",", row.names=FALSE)
```

6 Analysis of quantitative variables

The third analysis that will be examined is estimating the CDF and percentiles for quantitative variables. Two quantitative variables will be examined: (1) IBI_score - IBI score and (2) WQ_score - WQ score. The summary function is used to summarize the data structure of the two quantitative variables.

```
> summary(SC_estuaries$IBI_score)
```

Summarize the data structure of the IBI score variable:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 1.000   3.000   3.500   3.612   4.125   5.000  19.000
```

```
> summary(SC_estuaries$WQ_score)
```

Summarize the data structure of the WQ score variable:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 2.700   4.000   4.600   4.407   5.000   5.000  19.000
```

The cont.analysis function will be used to calculate estimates for quantitative variables. Input to the cont.analysis function is the same as input for the cat.analysis function except that the data frame containing response variables is named cont.data rather than cat.data. The sites, subpop, and design data frames created in the analysis of estuary condition variables can be reused for this analysis. The data.cont data frame contains the two quantitative variables: IBI_Score and WQ_Score, which contain the numeric scores for the IBI and WQ variables, respectively. Variables IBI_score and WQ_score in the SC_estuaries data frame are assigned to IBI_Score and WQ_Score, respectively. The popsize argument is included in the call to cont.analysis.

Create the data.cont data frame.

```
> data.cont <- data.frame(siteID=SC_estuaries$siteID,
+                         IBI_Score=SC_estuaries$IBI_score,
+                         WQ_Score=SC_estuaries$WQ_score)
```

Use the cont.analysis function to calculate CDF and percentile estimates for the quantitative variables.

```
> CDF_Estimates <- cont.analysis(sites, subpop, design, data.cont,
+   popsize=list(All_Estuaries=sum(framesize),
+               Estuary_Type=as.list(framesize)))
```

The object produced by cont.analysis is a list containing two objects: (1) CDF, a data frame containing the CDF estimates and (2) Pct, a data frame containing percentile estimates plus estimates of population values for mean, variance, and standard deviation. Format for the CDF data frame is analogous to the data frame produced by cat.analysis. For the CDF data frame, however, the fourth column is labeled Value and contains the value at which the CDF was evaluated. Unlike the data frames produced by the other analysis functions we have examined, the Pct data frame contains only nine columns since there is a single set of estimates rather than two sets of estimates. In addition, the fourth column is labeled Statistic and identifies either a percentile or the mean, variance, or standard deviation. Finally, since percentile estimates are obtained by inverting the CDF estimate, the percentile estimates do not have a standard error value associated with them.

Use the write.table function to write the CDF estimates as a csv file.

```
> write.table(CDF_Estimates$CDF, file="CDF_Estimates.csv", sep=",",
+             row.names=FALSE)
```

The cont.cdfplot function in spsurvey can be used to produce a PDF file containing plots of the CDF estimates. The primary arguments to cont.cdfplot are a character string containing a name for the PDF file and the CDF data frame in the CDF_Estimates object.

Produce a PDF file containing plots of the CDF estimates.

```
> cont.cdfplot("CDF_Estimates.pdf", CDF_Estimates$CDF)
```

Print the percentile estimates for IBI score for all sites combined.

```
> print(CDF_Estimates$Pct[1:10, ])
```

```
           Type Subpopulation Indicator      Statistic NResp  Estimate
1  All_Estuaries All Estuaries IBI_Score          5Pct     3 1.9835561
2  All_Estuaries All Estuaries IBI_Score         10Pct     6 2.2809551
3  All_Estuaries All Estuaries IBI_Score         25Pct    17 2.8823748
4  All_Estuaries All Estuaries IBI_Score         50Pct    60 3.5875846
5  All_Estuaries All Estuaries IBI_Score         75Pct    87 4.1208723
6  All_Estuaries All Estuaries IBI_Score         90Pct    87 4.4496095
7  All_Estuaries All Estuaries IBI_Score         95Pct   110 4.6753552
21 All_Estuaries All Estuaries IBI_Score          Mean   116 3.7144320
31 All_Estuaries All Estuaries IBI_Score      Variance   116 0.6908874
41 All_Estuaries All Estuaries IBI_Score Std. Deviation  116 0.8311964
             StdError  LCB95Pct  UCB95Pct
1                    1.5615314 2.1490635
2                    2.0393930 2.5134122
3                    2.6702423 3.0793319
4                    3.3695953 3.7713425
5                    3.9325709 4.3115620
6                    4.2517257 4.9369713
7                    4.4424379 5.0000000
21 0.0721515473688744 3.5730176 3.8558464
31 0.0866221699439855 0.5211111 0.8606637
41 0.0521069224449442 0.7290687 0.9333241
```

Use the write.table function to write the percentile estimates as a csv file.

```
> write.table(CDF_Estimates$Pct, file="Percentile_Estimates.csv", sep=",",
+             row.names=FALSE)
```

The cont.cdftest function in spsurvey can be used to test for statistical difference between the CDFs from subpopulations. For this analysis we will use the cont.cdftest function to test for statistical difference between the CDFs from the two strata. Arguments to cont.cdftest are the same as arguments to cont.analysis. Since we are interested only in testing among strata, the subpop data frame is subsetted to include only the siteID and Estuary_Type variables. Note that the popsize argument was modified from prior examples to include only the entry for Estuary_Type.

```
> CDF_Tests <- cont.cdftest(sites, subpop[,c(1,3)], design, data.cont,
+     popsize=list(Estuary_Type=as.list(framesize)))
```

The print function is used to display results for IBI score of the statistical tests for difference between CDFs for strata. The object produced by cont.cdftest is a data frame containing eight columns. The first column (Type) identifies the population. The second and third columns (Subpopulation_1 and Subpopulation_2) identify the subpopulations. The fourth column (Indicator) identifies the response variable. Column five contains values of the test statistic. Six test statistics are available, and the default statistic is an F-distribution version of the Wald statistic,

which is identified in the data frame as "Wald-F". The default statistic is used in this analysis. For further information about the test statistics see the help file for the cdf.test function in spsurvey, which includes a reference for the test for differences in CDFs. Columns six and seven (Degrees_of_Freedom_1 and Degrees_of_Freedom_2) provide the numerator and denominator degrees of freedom for the Wald test. The final column (p_Value) provides the p-value for the test.

```
> print(CDF_Tests)

          Type Subpopulation_1 Subpopulation_2 Indicator     Wald_F
1 Estuary_Type      Open Water     Tidal Creek IBI_Score   2.981072
2 Estuary_Type      Open Water     Tidal Creek  WQ_Score  14.585353
  Degrees_of_Freedom_1 Degrees_of_Freedom_2      p_Value
1                    2                  109 5.489145e-02
2                    2                  109 2.438223e-06
```

Use the write.table function to write CDF test results as a csv file.

```
> write.table(CDF_Tests, file = "CDF_Tests.csv", sep = ",", row.names = FALSE)
```