

# Analysis of a GRTS Survey Design for an Area Resource

Thomas Kincaid

August 16, 2016

## Contents

<b>1 Preliminaries</b>	<b>1</b>
<b>2 Load the survey design and analytical variables data set</b>	<b>2</b>
<b>3 Analysis of site status evaluation variables</b>	<b>5</b>
<b>4 Analysis of estuary condition variables</b>	<b>8</b>
<b>5 Analysis of estuary condition variables correcting for population size</b>	<b>10</b>
<b>6 Analysis of quantitative variables</b>	<b>11</b>

## 1 Preliminaries

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.

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.

```
> # Load the spsurvey package
> library(spsurvey)
>
```

Version 3.2 of the spsurvey package was loaded successfully.

## 2 Load the survey design and analytical variables data set

The next step is to load the data set, which includes both survey design variables and analytical variables. The `data` function is used to load the data set 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.

Load the survey design and analytical variables data set

```
> # Load the data set and determine the number of rows in the data frame
> data(SC_estuaries)
> nr <- nrow(SC_estuaries)
>
```

Display the initial six lines in the data file.

```
> # Display the initial six lines in the data file
> head(SC_estuaries)
```

	siteID	xcoord	ycoord	wgt	Stratum	Status	IBI_score	IBI_status
1	EEOW00-001	1549286	1263060	10.47516	Open Water	Sampled	3.5	Good
2	EEOW00-002	1487515	1226790	10.47516	Open Water	Sampled	4.0	Good
3	EEOW00-003	1442800	1159806	10.47516	Open Water	Sampled	4.0	Good
4	EEOW00-004	1425120	1148898	10.47516	Open Water	Sampled	4.5	Good
5	EEOW00-005	1432141	1140626	10.47516	Open Water	Sampled	4.5	Good
6	EEOW00-006	1540516	1280598	10.47516	Open Water	Sampled	2.5	Mrgn
	WQ_score	WQ_status						
1	4.3	Good						
2	4.6	Good						

```

3      5.0      Good
4      5.0      Good
5      5.0      Good
6      4.2      Good

```

```
>
```

Display the final six lines in the data file.

```

> # Display the final six lines in the data file
> tail(SC_estuaries)

```

```

      siteID  xcoord  ycoord    wgt    Stratum    Status  IBI_score
130 EETC99-035 1441803 1151432 1.41106 Tidal Creek NonTarget      NA
131 EETC99-036 1535415 1247815 1.41106 Tidal Creek   Sampled    3.5
132 EETC99-037 1500847 1225230 1.41106 Tidal Creek   Sampled    2.5
133 EETC99-038 1440701 1147436 1.41106 Tidal Creek   Sampled    3.0
134 EETC99-039 1468472 1179318 1.41106 Tidal Creek   Sampled    4.0
135 EETC99-040 1430639 1151724 1.41106 Tidal Creek   Sampled    2.5
      IBI_status  WQ_score  WQ_status
130      <NA>      NA      <NA>
131      Good      4.3      Good
132      Mrgn      3.0      Poor
133      Good      3.7      Mrgn
134      Good      4.3      Good
135      Mrgn      3.7      Mrgn

```

```
>
```

The location of sample sites in South Carolina estuaries is displayed in Figure 1. The sites for each stratum are displayed using a unique color.

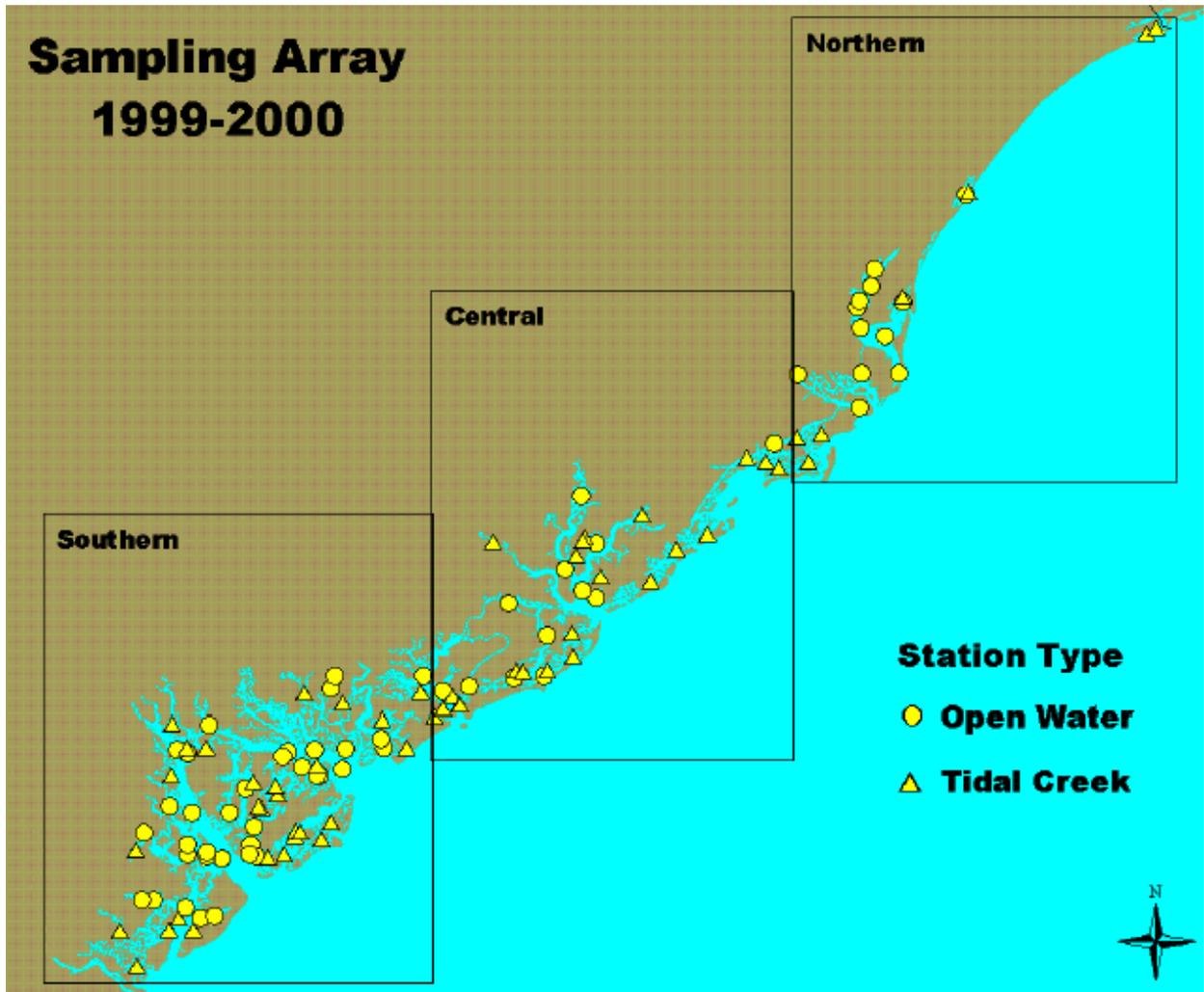


Figure 1: Location of estuaries that were sampled in South Carolina by the U.S. Environmental Protection Agency during the National Lakes Assessment (NLA) conducted in 1999 and 2000.

### 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), survey design weights are incorporated into the calculation process. Weights used in the analyses were modified from the original survey design weights to ensure that the weights sum to the known size of the resource. Further information regarding weight adjustment is provided in the help page for the `adjwgt` (weight adjustment) function. 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 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) `AllEstuaries` - 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.

```
> # 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 the data frame 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 the data frame 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). Note that the size estimate for the Total category will be equal to the sum of the survey design weights.

```
> # Print the extent estimates
> print(Extent_Estimates)
```

	Type	Subpopulation	Indicator	Category	NResp	Estimate.P	StdError.P				
1	All_Estuaries	All Estuaries	Status	NonTarget	19	4.885243	1.333584				
2	All_Estuaries	All Estuaries	Status	Sampled	116	95.114757	1.333584				
3	All_Estuaries	All Estuaries	Status	Total	135	100.000000	0.000000				
4	Estuary_Type	Open Water	Status	NonTarget	1	1.666667	1.443982				
5	Estuary_Type	Open Water	Status	Sampled	59	98.333333	1.443982				
6	Estuary_Type	Open Water	Status	Total	60	100.000000	0.000000				
7	Estuary_Type	Tidal Creek	Status	NonTarget	18	24.000000	3.913876				
8	Estuary_Type	Tidal Creek	Status	Sampled	57	76.000000	3.913876				
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	2.271467	7.499019	35.87423	9.425042e+00	17.40149	54.34698					
2	92.500981	97.728533	698.46457	4.378451e+01	612.64850	784.28063					
3	100.000000	100.000000	734.33880	4.207087e+01	651.88141	816.79619					
4	0.000000	4.496819	10.47516	9.075561e+00	0.000000	28.26293					
5	95.503181	100.000000	618.03415	9.075561e+00	600.24637	635.82192					
6	100.000000	100.000000	628.50930	5.163671e-14	628.50930	628.50930					
7	16.328944	31.671056	25.39908	4.142035e+00	17.28084	33.51732					
8	68.328944	83.671056	80.43042	4.142035e+00	72.31218	88.54866					
9	100.000000	100.000000	105.82950	4.854608e-15	105.82950	105.82950					

```
>
```

The `write.csv` 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.csv(Extent_Estimates, file="Extent_Estimates.csv")
```

## 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) `IBL_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 Mrgn	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 Mrgn	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: `IBL_Status` and `WQ_Status`. Variables `IBL_status` and `WQ_status` in the `SC_estuaries` data frame are assigned to `IBL_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.

```
> # Calculate estimates for the categorical variables  
> Condition_Estimates <- cat.analysis(sites, subpop, design, data.cat)  
>
```

Print the estuary condition estimates for all sites combined.

```
> # Print the condition estimates for all basins combined  
> print(Condition_Estimates[c(1:4, 13:16),])
```

	Type	Subpopulation	Indicator	Category	NResp	Estimate.P	StdError.P				
1	All_Estuaries	All Estuaries	IBI_Status	Good	99	86.1838689	3.054977				
2	All_Estuaries	All Estuaries	IBI_Status	Mrgn	14	11.9123445	3.101725				
3	All_Estuaries	All Estuaries	IBI_Status	Poor	3	1.9037866	1.349820				
4	All_Estuaries	All Estuaries	IBI_Status	Total	116	100.0000000	0.000000				
13	All_Estuaries	All Estuaries	WQ_Status	Good	83	82.9514988	3.454680				
14	All_Estuaries	All Estuaries	WQ_Status	Mrgn	29	16.2404087	3.443841				
15	All_Estuaries	All Estuaries	WQ_Status	Poor	4	0.8080925	0.364253				
16	All_Estuaries	All Estuaries	WQ_Status	Total	116	100.0000000	0.000000				
	LCB95Pct.P	UCB95Pct.P	Estimate.U	StdError.U	LCB95Pct.U	UCB95Pct.U					
1	80.19622370	92.171514	601.96379	41.403906	520.8136211	683.11395					
2	5.83307547	17.991613	83.20351	22.101079	39.8861870	126.52082					
3	0.00000000	4.549386	13.29728	9.383106	0.0000000	31.68783					
4	100.00000000	100.000000	698.46457	39.709481	620.6354124	776.29372					
13	76.18045106	89.722546	579.38683	43.296215	494.5278028	664.24585					
14	9.49060367	22.990214	113.43350	24.285825	65.8341570	161.03284					
15	0.09416982	1.522015	5.64424	2.463364	0.8161351	10.47234					
16	100.00000000	100.000000	698.46457	39.709481	620.6354124	776.29372					

```
>
```

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

```
> write.csv(Condition_Estimates, file="Condition_Estimates.csv")
```

## 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 `IBLstatus` variable, population "All\_Estuaries" and subpopulation "All Estuaries" is 698 square kilometers (rounded to a whole number). This value is an estimate of the size of the sampled resource. The corresponding frame size value is 734 square kilometers. The `popsize` (population size) argument to `cat.analysis` provides a mechanism for forcing the size estimates to sum to a desired value, e.g., the frame size value. Note that including `popsize` as an argument results in assigning the `popsize` value to the Total category of the size estimates. Use of the `popsize` argument assumes that sites which were evaluated but not sampled were missing at random. The missing at random assumption may not be a valid assumption, e.g., sites for which access was denied by the landowner may not be the same as sites that were sampled. For the current analysis, we will assume that the assumption is valid. As a first step for use of the `popsize` argument, 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 all sites combined.

```
> # Print the estuary condition estimates for all sites combined
> print(Condition_Estimates_popsize[c(1:4, 13:16),])
```

	Type	Subpopulation	Indicator	Category	NResp	Estimate.P	StdError.P
1	All_Estuaries	All Estuaries	IBI_Status	Good	99	86.1838689	3.054977
2	All_Estuaries	All Estuaries	IBI_Status	Mrgn	14	11.9123445	3.101725
3	All_Estuaries	All Estuaries	IBI_Status	Poor	3	1.9037866	1.349820
4	All_Estuaries	All Estuaries	IBI_Status	Total	116	100.0000000	NA
13	All_Estuaries	All Estuaries	WQ_Status	Good	83	82.9514988	3.454680
14	All_Estuaries	All Estuaries	WQ_Status	Mrgn	29	16.2404087	3.443841
15	All_Estuaries	All Estuaries	WQ_Status	Poor	4	0.8080925	0.364253
16	All_Estuaries	All Estuaries	WQ_Status	Total	116	100.0000000	NA
	LCB95Pct.P	UCB95Pct.P	Estimate.U	StdError.U	LCB95Pct.U	UCB95Pct.U	
1	80.19622370	92.171514	632.881606	22.433883	588.9120028	676.85121	
2	5.83307547	17.991613	87.476970	22.777170	42.8345376	132.11940	
3	0.00000000	4.549386	13.980244	9.912255	0.0000000	33.40791	
4	NA	NA	734.338820	NA	NA	NA	
13	76.18045106	89.722546	609.145057	25.369054	559.4226254	658.86749	
14	9.49060367	22.990214	119.259626	25.289464	69.6931870	168.82606	
15	0.09416982	1.522015	5.934137	2.674851	0.6915255	11.17675	
16	NA	NA	734.338820	NA	NA	NA	

```
>
```

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

```
> write.csv(Condition_Estimates_popsize, file="Condition_Estimates_popsize.csv")
```

## 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

```
> 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

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: `IBLScore` and `WQ_Score`, which contain the numeric scores for the IBI and WQ variables, respectively. Variables `IBLscore` and `WQ_score` in the `SC_estuaries` data frame are assigned to `IBLScore` 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.csv` function to write the CDF estimates as a csv file.

```
> write.csv(CDF_Estimates$CDF, file="CDF_Estimates.csv")
```

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 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
8	All_Estuaries	All Estuaries	IBI_Score	Mean	116	3.7144320
9	All_Estuaries	All Estuaries	IBI_Score	Variance	116	0.6908874
10	All_Estuaries	All Estuaries	IBI_Score	Std. Deviation	116	0.8311964
		StdError	LCB95Pct	UCB95Pct		
1			1.5616016	2.1490377		
2			2.0408270	2.5125581		
3			2.6707797	3.0788808		
4			3.3703757	3.7707670		
5			3.9331037	4.3110103		
6			4.2521696	4.9356565		
7			4.4428288	5.0000000		
8	0.0718189615220736		3.5736694	3.8551946		
9	0.0865446486737272		0.5212630	0.8605118		
10	0.0520602901032027		0.7291601	0.9332327		

```
>
```

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

```
> write.csv(CDF_Estimates$Pct, file="Percentile_Estimates.csv")
```

The `cont.cdfctest` function in `spsurvey` can be used to test for statistical difference between the CDFs from subpopulations. For this analysis we will use the `cont.cdfctest` function to test for statistical difference between the CDFs from the two strata. Arguments to `cont.cdfctest` 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.cdfctest(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.cdfctest` 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 results of the statistical tests for difference between strata CDFs for
> # IBI score and WQ score
> print(CDF_Tests, digits=3)
```

	Type	Subpopulation_1	Subpopulation_2	Indicator	Wald_F
1	Estuary_Type	Open Water	Tidal Creek	IBI_Score	2.98
2	Estuary_Type	Open Water	Tidal Creek	WQ_Score	14.61
	Degrees_of_Freedom_1	Degrees_of_Freedom_2	p_Value		
1	2	109	5.50e-02		
2	2	109	2.39e-06		

```
>
```

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

```
> # Write CDF test results as a csv file
> write.csv(CDF_Tests, file="CDF_Tests.csv", row.names=FALSE)
>
```