Data Analysis Protocols for the Status Network, Version 2.0

Guidelines for Florida's Probabilistic Monitoring Network

Division of Environmental Assessment and Restoration
Florida Department of Environmental Protection
August 2018

2600 Blair Stone Rd, MS 3560
Tallahassee, Florida 32399-2400
www.dep.state.fl.us

Table of Contents

Introduction

The probabilistic design of the Status Network utilizes an unbiased, rigorous data set for the purpose of answering water-resource questions with mathematical confidence. The Network allows DEP to ask specific questions regarding water quality and answer those questions within statistical confidence limits. The design is planned with specific questions in mind.

Questions addressed by the Status Network monitoring design comprise three different scales: (1) the state of Florida as a whole, (2) regions of the state, and (3) large drainage basins, or drainage basin complexes (i.e., reporting units). In 2009, the Network was augmented to answer questions on a statewide and a zone (regional) basis with fewer sampling events. The questions that pertain to the Status Network relate to water quality *on a statewide and regional basis*; they are *not waterbody specific*. The Network is not designed to address the questions related to small drainage basins, counties, or individual bodies of water. The Integrated Water Resource Monitoring Design[1] (Copeland, et al., 1999) has determined that questions for these smaller areas be addressed by other monitoring programs. The design of the Status Network is for addressing statewide and coarse-scale questions with a high degree of statistical confidence.

The Status Network design provides advantages not offered by other designs. For one, the statistical confidence provided by the methodologies and random sampling design provide the advantage of estimating the proportion of waters not meeting the threshold for an analyte within confidence bounds. Secondly, the random design generates proportion results that not only lend themselves to spatial comparisons (e.g., between one part of the state to another) but also temporal comparisons. Therefore, an objective statement with statistical confidence can be generated that tracks the water quality of the state. Hence, the Network will have a unique ability to provide long-term tracking of statewide water quality, with statistical confidence, for decades to come. Comparisons can be made for the entire state through time, for an individual basin, or between different basins given enough sampling events have been conducted. In summary, the Status Network allows us to ask the questions that both the Department of Environmental Protection, and the Florida public, regularly ask. Simply stated: are conditions getting better, remaining the same, or getting worse?

The document is divided into five sections. Part I outlines the operation of the Status Network and questions that may be addressed. Part II describes quality assurance guidelines. Part III describes the procedure used to extract the data that will be used in the analysis. Part IV describes procedures that are used when the questions being addressed require data from multiple years or resources to be combined. Part V takes an example resource, small lakes, and explains how the computer code used by the Status Network works.

[1] The Integrated Water Resource Monitoring Program is sometimes referred to by the acronym, "IWRM." http://publicfiles.dep.state.fl.us/dear/DEARweb/WMS/Reports_Docs_SOPs/Historic_Programs/IWRM.pdf

Part I – The Status Network

The Status Network has three components. First, standardized protocols are employed in data acquisition, e.g. using standardized sample collection methods to minimize error associated with sampling. Second, the data population must be characterized. This means that the population of resources from which the data were collected must be characterized to statistically describe the variability of the distributions of the indicators sampled. Third, the calculated distributions are used to make inferences on the overall condition of the resources (i.e., the original, overall population that was sampled).

For more information on the Status Monitoring Network please refer to https://floridadep.gov/dear/water-quality-assessment/content/reports-documents-sops-and-links for documentation on the design, sample collection methods, data management procedures, etc.

Questions Addressed by the Status Network

Each year the analyses are conducted on the data on a statewide basis. After a minimum of three years the data are combined and the analyses are performed on both the statewide data and the data in each of six reporting zones (based on the five water management districts with South split into an east and west section). For the state and each zone, a number of statistics are calculated. These are performed through extent and continuous variable estimate calculations. Appendix A gives example code for small streams. The following types of statistics can be assessed:

- What is the accessible proportion of a resource (surface or ground water)?
- How many samples were collected for each resource?
- What is the size of the resource?
- What proportion of the resource is dry?
- What proportion of flowing surface waters (i.e., rivers, streams, and canals) do not meet water quality thresholds?
- What proportion of lake surface waters (i.e., large lakes and small lakes) do not meet water quality thresholds?
- What proportion of ground water (i.e., as confined and unconfined aquifers) do not meet water quality thresholds?
- For flowing surface water resources, what proportion of dissolved oxygen, *Escherichia coli*, pH, unionized ammonia, chlorophyll a, and habitat metrics do not meet water quality thresholds?
- For ground water resources, what proportion of arsenic, cadmium, chromium, lead, nitrate-nitrite, sodium, fluoride, and total coliform are do not meet water quality thresholds?

Part II – Data Extraction

Water quality data collected for the Status Network are stored in the Generalized Water Information System (GWIS), an enterprise Oracle database. Once in GWIS, the data are reviewed by DEP Watershed Monitoring Section (WMS) staff before being deemed suitable for distribution and/or use in analysis. Data extractions from GWIS are performed by the WMS Data and Analysis Coordinator using SQL scripts and a PL/SQL package. Two types of data extractions are performed per water resource. One pulls all site reconnaissance information including the sites which are excluded from sample collection and the reason why each site was

excluded. The other pulls the field and analytical data generated from the sites which were sampled.

The file containing the reconnaissance information (from here on in referred to as the 'exclusion file') is created using either TONYS_EXCLUSION_SCRIPT.sql for surface water or TONYS_EXCLUSION_SCRIPT_WELL.sql for groundwater (located on DEP server \\fldep1\sol z\sql). These scripts create tables in the GWIS_ADMIN schema to hold the data for review before it is exported and given to the data analyst. For each resource, the exclusion file is checked to make sure that the targeted number of sites were sampled in each zone and that all sampled sites have a date sampled entered in GWIS. Excluded sites are checked to make sure an exclusion criterion and exclusion category were entered in GWIS. Any missing information is relayed to the appropriate project manager for updates. Once all missing information is filled in, any sites that are beyond the last sampled site are removed from the file. Once the data has been reviewed and updated if necessary, the data are exported as an Excel file and saved on a DEP server (\\fldep1\sol z\data analysis) in a folder for the current year and resource.

The file containing the field and analytical results (from here on in referred to as the 'results file') is created using a PL/SQL package called PKG_RESULT_PIVOT for each water resource. This package retrieves certain metadata elements (Sample ID, Station No, Station Name, Collection Date, Sample Type, Matrix) along with all the results (field and laboratory measurements and data qualifiers) for a resource. The retrieved data is exported as an Excel file and saved on a DEP server (\\fldep1\sol z\data analysis) in a folder for the current year and resource.

The data analyst is notified by the data manager when the exclusion and results files have been prepared. For each resource, if the number of samples collected differs from the predetermined number of samples for any zone, the project manager communicates this information to the data analyst. The exclusion file is used to provide information on the extent of waters determined to be non-target (not part of the water resource being assessed), and the extent of waters excluded due to other reasons. This information is used during data analysis to calculate weighting factors for the sites which were sampled.

Part III – Data Quality Assessments

Status Network data are collected and analyzed in accordance of the quality assurance protocols established for the Watershed Monitoring Section's Status Monitoring Network. Proper field and laboratory protocols are followed prior to their incorporation into the database. Data extracted from the database are then examined by the data analyst. This includes scanning data for outliers and erroneous values. After the analyses are complete, they are independently checked, using the quality assurance checklist in Appendix D as a guide.

The data analyst performs the following checks before data analyses.

Errors in the Data

The data should be screened for anomalies, outliers and questionable results based on proper ranges of values. Data that are orders of magnitude in variance from the central tendency are the most easily identified. Other values may fall outside of the logical range of values (e.g., negative values for NO_3). Continuous variable data (i.e., those not including raw count values) should have

values greater than zero. Unusual values may warrant further investigation, including examination of original field records.

Outliers

One of the most common types of data errors are outliers. Outliers should not be discarded or arbitrarily defined. Though arbitrary standards can be set, all data handling is ultimately a matter for best professional judgment. For example, one way of defining outliers is identification of points beyond what are referred to as the upper fence. In order to determine the upper fence, the Inter Quartile Range (IQR) (the value of the upper quartile minus the value of the lower quartile) is determined. The upper fence is then found by adding the quantity 1.5 times the IQR to the upper quartile. However, a point 1.5 times the upper quartile is often a valid data point, and erroneous data values may reside within this range. Ultimately, all decisions are subject to the analyst's best judgment given the data that is provided. Since most Status Network analyses are nonparametric, being based on the rank of the values as opposed to the values themselves, outliers are less problematic than they are for parametric analyses. Removal of data from the dataset should only occur under the most obvious of circumstances. These include measurements that are mathematical impossibilities (e.g., pH values greater than 14), conditions that reflect physical impossibilities of condition (e.g., temperatures of 2700 C), mislabeled data, and laboratory instrumentation errors.

Missing Values

Status Network sampling requires 15 collection points per zone for each surface water resource and 20 for each ground water resource. When less than these values are present in a specific zone, the reasons must be documented.

Detection Limits

Some reported data are at detection limits. The Status Network employs the laboratory's reported value for the method detection limit (MDL) for incorporation into continuous distribution functions (CDFs) for each indicator of water quality.

Qualifier Codes

Status Network qualifier codes are recorded in Appendix B and adhere to the 2017 QA Rule 62-160.700 Table 1 (Data Qualifier Codes). The type of qualifier, the reason for the particular qualifier, and the overall number of qualifiers should all be considered. A single qualifier can represent a variety of problems ranging from unimportant to serious. Additionally, the number of qualifiers may pose an additional problem; greater numbers are likely to have an influence on the CDFs. Because a CDF represents a nonparametric statistical distribution of underlying conditions, a single (or small number of) qualified data point will likely not affect the results. It is recommended that all qualified data remain in the analysis and that clear documentation of the reasons for the qualified data be presented alongside the results. CDFs constructed from heavily qualified data—or of much qualified data near threshold ranges—will have compromised accuracy. Qualified values exceeding thresholds pose the greatest need for additional examination and explanation.

Predetermined guidelines for how to handle various combinations of data qualifiers can be assigned. For example, analysis results could be flagged when the data includes 5 or more qualifiers. Whether, or not, such guidelines are established, data analysts and readers/editors of the reports must ultimately make decisions on a case-by-case basis as to whether, or not, to accept

the findings. In any case, readers should be given enough information to make informed decisions about the resource(s) in question.

Specific issues regarding data qualifiers include the following:

- Q qualified data – samples held beyond holding times. Bacteria data from the lab are submitted only when samples are analyzed within holding times or up to 30 hours outside holding times. All Q qualified bacteria data are currently used based on an in-house study of the Biology Section of the lab showing little degradation of *Escherichia coli* samples up to 30 hours outside holding times (*Holding Time Study for Water Quality Assessments,* July 5, 2013. Biology Section, Bureau of Laboratories, FDEP).
- J qualified data— J qualified data may have failed QA and QC protocols, whether in the field or the lab.
- V qualified data—data that registers detection in both samples and laboratory method blanks.
- G qualified data—data that registers detection in both samples and field blanks or equipment blanks.

Part IV – Combining Data

Combining Data from Multiple Years

Increased sample size is desirable because it generally has a positive effect on the confidence levels for the reported data, thereby increasing the confidence for statewide reporting. One way to increase the sample size is to combine data collected in different years. For status monitoring reporting, three years of annually collected status monitoring network data are used. The increase in sample size allows statewide reporting with 95% confidence levels at ± < 10%.

The process of analyzing three years of data requires several steps which are important to maintaining the statistical integrity of the analyses. First, for each water resource, the three yearly exclusion and result files are merged to provide a combined exclusion and a combined results file. These two combined files are checked to ensure that the columns in the files are the same for all three years. If there are columns in the exclusions file that do not exist in all three years, the exclusions file may need to be recreated by the Data and Analysis Coordinator. If there are columns in the results file that do not exist in all three years (usually due to an analyte being dropped from or added to the analyte list), a decision must be made to keep or delete the column. It is desirable to keep as many columns (i.e. analytes) as possible in the final three-year result file.

The next step is to make sure the sites in the combined exclusions file exist in the final year's water resource coverage. For example, only the water resource coverage for 2017 is used in the 2015-2017 analysis. The location of the sites sampled for 2015 and 2016 are checked to make sure they came from waterbodies present in the 2017 water resource coverage. Any sites that are not a part of the final year's water resource coverage are removed from the combined exclusions and results files before processing.

The extent of the water resources from the final year's water resource coverage (flowing water length in km or lake area in hectares) are derived during the site selection process from the shapefile used and are recorded in the accompanying design document (initially saved on DEP

server \\fldep1\sol z\Chris work in progress and then copied to the appropriate folder in \\fldep1\sol z\data analysis). The extent of each water resource in each zone is provided from an R software script run for the site selections. During data analysis these water resource extents in conjunction with the site exclusion information are used to calculate site weighting factors. Only the water resource extent for the final year is used in order to ensure that the statistics are not based on data generated from waterbodies, or waterbody segments, which do not meet the definition of the water resource's target population.

Combining Data from Multiple Resources

It is also desirable to report on the condition of combined water resources (e.g. flowing waters and lakes). This will increase the sample size and allow a more global look at the condition of the state's waters which also may be tracked over time When combining the flowing water resources, the lengths calculated from the coverages for all three resources (rivers, streams, and canals) are copied into an Excel spreadsheet. These lengths are summed to obtain the extent (total flowing waters length in kilometers) for the combined rivers and streams resource (LRSS) and the combined rivers, streams, and canals (LRSSCN) resource. These are the lengths used in the R software script as either targetsize or framesize.

Lakes require a different procedure as the large lake site selections are based on area while the small lake sites selections are represented by points and need to be recalculated based on area. An Excel spreadsheet with the final year's areal coverage is created and the ratio of each lake resource is calculated. In 2017 large lakes consisted of over 390,000 hectares or 97.1% of lake area. Small lakes were just under 12,000 hectares or 2.9% of lake area. The estimated percentages of each resource meeting a specific water quality threshold are calculated separately, and this ratio is used to calculate the combined result. For example, in 2017 the estimate for combined lakes (LLSL) meeting the threshold for total nitrogen (TN) was 79.0%. This result was calculated based on (94.8% meeting threshold)* (2.9%) for small lakes plus (78.5% meeting threshold)*(97.1%) for large lakes.

Part V – Analysis Procedures

The following is an example of data analysis on Status Network small lakes data from 2015-2017. An R script is used to generate the statistical output. The script was developed from original S-PLUS© code written and supplied by Tony Olsen, a US EPA statistician stationed at the US EPA facility in Corvallis, OR (the original script is saved on a DEP server \\fldep1\sol z\data analysis 2006_305B\Old SPLUS Codes\305b 2006-all codes\HomeDesktop\Suwannee River\Small Lakes). Analyses for other resources use similar script that has been adapted to the resource in question.

Running the R scripts

The following code runs in R Studio version 1.1.447, using R Statistical Software version 3.4.3.

Start R Studio. Create a new R project, saved to the DEP server data analysis folder (\\fldep1\sol z\data analysis) for the appropriate year and resource. Check that the R workspace is set to the same location as where the R project was created.

Locate the R script file from the previous year's analysis for the same resource, save a copy in the same location as where the R project was created, and rename it to reflect the year and resource being analyzed. Open the renamed R script using R Studio.

Load the sp and spsurvey packages. If these library folders are not installed, they can be downloaded from the Comprehensive R Archive Network (CRAN) repository using the install package from repository tool in the R Studio tools menu.

Next load the data files (exclusion and result file). Data files are initially loaded by one set of procedures but later saved within the R workspace to be called by other commands.

Use the import dataset tool in the R Studio file menu for the initial loading of the data files. Select the two files to be loaded. Check that all columns will be loaded using the correct data type, and use the drop-down menus above the individual columns to change the data type if needed. All columns containing numeric data should be loaded as type "double". See Appendix C for a list of data formats for other columns.

If data have already been loaded into the current R workspace then they are stored within R and are ready to be used in other commands.

After the data files have been loaded, the R script can be run. To run a portion of the script, highlight relevant portions of text code in the script window, and hit the run button (small green arrow near upper right of script window). The console window will show the output, including any error messages received. Run sections of code step-by-step in order to make sure each step runs properly (e.g., no errors). Sections of code are divided by comment lines. (In R these are specified by # symbols preceding text and will not execute with other script. They are descriptive only).

Explanations for Each Portion of Script:

The following illustrates the script and provides commentary. Portions of script code are listed below in indented paragraphs (full code example for streams is in Appendix A), and explanations are beneath in italics.

```
# Read in data using File: import
data SL.EXCLUSIONS<-SL_EXC_2015.17DO
names(SL.EXCLUSIONS)
```

Read column headers into the console window; this insures data are being read properly by the script.

```
# do equal area projection for variance estimation
tmp <- marinus(SL.EXCLUSIONS$latdd, SL.EXCLUSIONS$londd)
SL.EXCLUSIONS$xmarinus <- tmp[,'x']
SL.EXCLUSIONS$ymarinus <- tmp[,'y']
```

In order to calculate variance, an equal area map projection must be created (marinus). This set of commands creates a map projection similar to those used by ArcMap, creates a matrix, x and y, and assigns this to the site info data frame. Latitude and longitude should be provided in decimal degree format. If they are provided in a different format, additional commands will be needed to convert to decimal degrees prior to running these commands.

```
# check design variables
table(SL.EXCLUSIONS$CAN_BE_SAMPLED)
table(SL.EXCLUSIONS$EXCLUSION_CATEGORY)
table(SL.EXCLUSIONS$EXCLUSION_CRITERIA)
table(SL.EXCLUSIONS$EXCLUSION_CATEGORY, SL.EXCLUSIONS$CAN_BE_SAMPLED)

# create status and TNT variables
SL.EXCLUSIONS$EXCLUSION_CATEGORY<-
   as.character(SL.EXCLUSIONS$EXCLUSION_CATEGORY)
SL.EXCLUSIONS$EXCLUSION_CATEGORY[SL.EXCLUSIONS$CAN_BE_SAMPLED == 'Y'] <-
   'SAMPLED'
SL.EXCLUSIONS$EXCLUSION_CATEGORY <- as.factor(SL.EXCLUSIONS$EXCLUSION_CATEGORY)
levels(SL.EXCLUSIONS$EXCLUSION_CATEGORY)
SL.EXCLUSIONS$TNT <- SL.EXCLUSIONS$EXCLUSION_CATEGORY
levels(SL.EXCLUSIONS$TNT) <- list(T=c('SAMPLED', 'NO PERMISSION FROM OWNER',
   'UNABLE TO ACCESS','OTHERWISE UNSAMPLEABLE','DRY'),
NT=c('WRONG RESOURCE/NOT PART OF TARGET POPULATION') )

table(SL.EXCLUSIONS$EXCLUSION_CATEGORY, SL.EXCLUSIONS$TNT)
```

Before proceeding with the remainder of the code, it is helpful to examine the loaded data. Rather than scanning through a data frame, it is easier to look at summary information. This allows familiarity with the data.

Variable TNT represents Target and Non-target samples. The exclusion categories are given but this is an additional simplification. The exclusion categories are subsumed into two categories: T and NT. Non-target (NT) samples are those sites that were excluded as wrong resource / not part of target population. Target (T) samples are sites that were either sampled or excluded for other reasons that are unlikely to be permanent (e.g. unable to access, dry, or no permission from owner).

```
##### adjust weights for use of over sample sites
framesize=c('ZONE 1'=2132.19332,'ZONE 2'=423.83411,
'ZONE 3'=5019.79487,'ZONE 4'=3620.17302,
'ZONE 5'=487.82495,'ZONE 6'=28.51382)
tst <- !is.na(SL.EXCLUSIONS$TNT)
SL.EXCLUSIONS <- SL.EXCLUSIONS[tst,]

nr <- nrow(SL.EXCLUSIONS)
SL.EXCLUSIONS$wgt <- adjwgt(rep(TRUE,nr),
SL.EXCLUSIONS$NEST1_WT,
SL.EXCLUSIONS$REPORTING_UNIT,
framesize=framesize)    ### change to match stratum
```

The values entered in the framesize command are the total size of the resource. These values are copied from the documentation for the design of each resource and pasted into the R script.

The adjust weights step accounts for oversampling. Ideally a selection of 15 lakes would result in 15 samples: complete accessibility. But what if 15 samples required 20 attempts? Sample size changes (by 20/15). This is the ratio adjustment to account for oversamples. [Entering "help(adjwgt)" in the R command window will display the function description].

Analysis of a different resource will require changing the framesize values here, as well as data columns names, in order for the code to work. Otherwise, calculations for other resources employ nearly identical code.

```
##### Example Extent estimation
sites <- data.frame(siteID=SL.EXCLUSIONS$PK_RANDOM_SAMPLE_LOCATION,
   Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=SL.EXCLUSIONS$PK_RANDOM_SAMPLE_LOCATION,
Combined=rep("All Basins",nr),
basin=SL.EXCLUSIONS$REPORTING_UNIT )

dsgn <- data.frame(siteID=SL.EXCLUSIONS$PK_RANDOM_SAMPLE_LOCATION,
wgt=SL.EXCLUSIONS$wgt,
xcoord=SL.EXCLUSIONS$xmarinus ,
ycoord=SL.EXCLUSIONS$ymarinus ,
stratum=SL.EXCLUSIONS$REPORTING_UNIT )

data.cat <- data.frame(siteID=SL.EXCLUSIONS$PK_RANDOM_SAMPLE_LOCATION,
EXCLUSION.CATEGORY=SL.EXCLUSIONS$EXCLUSION_CATEGORY,
TNTStatus=SL.EXCLUSIONS$TNT )

software.program <- 'R Studio'
ExtentEst <- cat.analysis(sites, subpop, dsgn, data.cat,
popsize=list(Combined=list("All Basins"=framesize),
basin=list("ZONE 1"=framesize,
          “ZONE 2"=framesize,
          "ZONE 3"=framesize,
          "ZONE 4"=framesize,
          "ZONE 5"=framesize,
          "ZONE 6"=framesize)),,
          conf=95 )
#write out or export results
write.table(ExtentEst,file='ExtentEst.csv',sep=','
```

There are different types of data: categorical and continuous. Examples of categorical data are exclusion categories, or whether something was sampled or not. The "cat. analysis" function is used for these types of data. [Entering “help(cat.analysis)” in the R command window will display the function description]. Running the cat.analysis function requires data to be setup in 4 different dataframes: sites, subpop, dsgn, and data.cat.

All sites included in the analysis are represented by the dataframe “sites”.

The part of the population examined is represented by the dataframe “subpop”. “Subpop” recombines all the data statewide and can group populations into smaller units (e.g., Reporting Units, Basins, etc.).

The dataframe "Dsgn" gives the survey design used: it takes sites, weights, and stratification. Though x and y coordinates are called here, ultimately these are not used in making estimates for CDFs. No geographic information goes into a CDF, though geographic information is employed within confidence limits or variance calculations (a local neighborhood variance calculation is found in cat.analysis).

The dataframe "Data.cat" gives the data used in the analysis, which is either 1) an exclusion categorical variable, or 2) a TNT variable.

The size of the population can be specified and is part of the included documentation in "Cat.analysis". (e.g. 402860.9 ha of *lakes, 1500 km of streams, etc.)*

The dataframe "ExtentEst" provides an estimated extent. For example, an estimate of the total number of dry lakes in Zone 2. If 10% of lakes sampled are dry, there would be 40286.1 ha in a sample size of 402860.9 ha of lakes.

```
###### Do summaries for data
# import data
SLcomb<-SL_RESULTS_2015.2017
names(SLcomb)

# Note that original data spreadsheet may have blank, primary and bottom data.
# Need to subset before doing analyses.
# Keep Primary data only.
keep <- SLcomb$Sample.Type == 'PRIMARY'

# Merge subset of data with exclusion file
comba <- merge(SL.EXCLUSIONS, SLcomb[keep,], by.x='PK_RANDOM_SAMPLE_LOCATION',
by.y='Station.Name')

# Check that merged data includes only PRIMARY data
summary(comba$Sample.Type)
# Check that merged data includes only water data. If a mix of water and
# sediment data is found, subset the merged data so that only water data are
# retained.
summary(comba$Matrix)
keep1 <- comba$Matrix == 'WATER'
comb<-(comba[keep1,])
summary(comb$Matrix)
names(comb)
```

It is necessary to subset the results data file because the SAMPLE.TYPE column often contains three types of data: BLANK, PRIMARY, and BOTTOM.

File merging must happen before the final results are produced. In the example code, we must merge the exclusion file and the subset of primary sample data from the results data file.

```
##### Example continuous indicator estimation
nr <- nrow(comb)
sites <- data.frame(siteID=comb$PK_RANDOM_SAMPLE_LOCATION, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb$PK_RANDOM_SAMPLE_LOCATION,
Combined=rep("All Basins",nr),
basin=comb$REPORTING_UNIT )

dsgn <- data.frame(siteID=comb$PK_RANDOM_SAMPLE_LOCATION,
wgt=comb$wgt,
xcoord=comb$xmarinus ,
ycoord=comb$ymarinus ,
stratum=rep('comb',nr) )

data.cont <-data.frame(siteID=comb$PK_RANDOM_SAMPLE_LOCATION,comb[, seq(91,
   125, by=2)] )
ExampleEst <- cont.analysis(sites, subpop, dsgn, data.cont)

write.table(ExampleEst$CDF,file='ExampleEstCDF.csv',sep=',')
write.table(ExampleEst$Pct,file='ExampleEstPCT.csv',sep=',')
write.table(ExampleEst$Tot,file='ExampleEstTOT.csv',sep=',')
```

The last portion of the code creates percent, total, and CDF calculations and writes them to tables. "Cont.analysis" is for use with the continuous variables (e.g., pH, nitrate, etc.). [Entering "help(cont.analysis)" in the R command window will display the function description]. CDFs

can only be created using continuous data; count data such as bacteria can theoretically be used, however tied data may generate problems. This is set up similarly to categorical analysis, with four different dataframes: sites, subpop, design, etc.

The function above creates CDF estimates for a range of variables (columns 91 through 125 of the combined data file) and subpopulations at the same time. The CDFs generated are estimated population CDFs. These are what the CDF would look like if all lakes in the listframe were sampled.

The program generates four output files which are saved in the folder designated as the current R workspace. The file names are: ExampleEstCDF, ExampleEstTot, ExampleEstPCT, and ExtentEst. These are comma delimited text files that can be opened in Excel.

Analysis for Indicator Thresholds Dependent on Geography and other Variables.

Some water quality indicators applicable to Florida's surface waters have complex thresholds, where the threshold value may vary based on geographic location or based on result values of other parameters. Examples of these indicators include total nitrogen (TN), total phosphorous (TP), dissolved oxygen (DO), and total ammonia. To evaluate whether the result data from each sampled site meets these thresholds, the nutrient region and dissolved oxygen region names and associated threshold values (if applicable) must be added to the data file with site information about sampled and excluded sites. If these columns are not included in the file provided by the Data and Analysis Coordinator, then they must be added using ArcMap prior to loading the data file in R Studio.

Prior to working with the data in ArcMap, the data files are opened in Excel and any empty cells are replaced with 'NaN' to avoid having empty cells convert to a zero. If the latitude and longitude are not provided in decimal degrees, they must be converted to decimal degrees before proceeding. In ArcMap, open the data file, and display the data as points (make sure the coordinate system is WGS84) and export the points in the combined spreadsheet as a shapefile. Add this shapefile as a layer in ArcMap. Add the nutrient regions (NWR101211.shp) and SCI bioregion (SCI_2012_Bioregions.shp) shapefiles from the DEP server (\\fldep1\SOL_Z\Chris work in progress\GIS_Coverages) as layers in ArcMap. Using ArcToolbox (Analysis Tools\Overlay\Identity) join the data shapefile and the nutrient regions shapefile so that each point has an associated NNC region. Repeat this process for the SCI bioregions using the previously joined file and the SCI bioregions layer. The final file will have both NNC and SCI (i.e. DO) regions associated with each site. If there are any sites that did not join to a nutrient or DO region, they must be examined manually by zooming in to see the region to which they fall closest. Manually update the regions that did not automatically populate, and export this file as a text file.

For flowing waters (rivers, streams, and canals), only geographic information is needed to determine the TN, TP, and DO thresholds. The text file that was exported by ArcMap is opened in Excel and the appropriate TN, TP, and DO thresholds are assigned to each site based on the NNC Region (F.A.C.62-302.531) and SCI 2012 Bioregion (F.A.C. 62-302.533).

For lakes (large lakes and small lakes), only geographic information is needed to determine the DO thresholds. The DO threshold values are added as described above for flowing waters.

Additional information, specifically the result values for true color and alkalinity, is needed to determine the TN and TP thresholds that apply to lake samples. The data are loaded into R and each sample is categorized according to its color and alkalinity result values.

```
# True color > 40 PCU assigned 0; Ture color <= 40 PCU assigned 1.
comb3$Colcat<- ifelse((comb3$Color..true > 40) ,0,1)
#Alkalinity > 20 mg/L CaCO3 assigned 0; Alkalinity <= 20 mg/L CaCO3 assigned 1.
comb3$Alkcat<- ifelse((comb3$Alkalinity..Total..as.CaCO3 > 20) ,0,1)
# Export out and amend file with NNC
write.table(comb3,file='comb3.csv',sep=',')
```

The data are then exported as a text file, opened in Excel, and the appropriate TN, TP, and DO thresholds are assigned to each site based on the color category, alkalinity category, and NNC Region (F.A.C.62-302.531). TN and TP have two applicable thresholds, based on additional criteria not currently being used in WMS reporting. For WMS reporting purposes, the less stringent ("maximum") threshold is currently being used.

For all surface water resources, after TN, TP, and DO thresholds have been assigned, the data are then loaded into R and the result values for each sample are compared to their respective thresholds. To determine the combined total, it is determined whether all three values (TN, TP, and DO) are meeting their thresholds.

```
#Pass = 1, Fail = 0
#TN
#Pass if TN threshold value is >= TN result value.
#Fail if TN threshold value is < TN result value.
comb3$TNcat<-ifelse((comb3$TN.min >=
comb3$TN),1,0)
#TP
#Pass if TP threshold value is >= TP result value.
#Fail if TP threshold value is < TP result value.
comb3$TPcat<-ifelse((comb3$TP.min >= comb3$Phosphorus..Total..as.P),1,0)
#DO
#Fail if DO threshold value is > DO result value.
#Pass if DO threshold value is =< DO result
value. comb3$DOcat<-ifelse((comb3$DO.Conc >
comb3$Oxygen..Dissolved.Percent.Saturation),0,1)
#Total
comb3$Tot<-(comb3$TNcat+comb3$TPcat+comb3$DOcat)
```

The percent of the resource meeting the thresholds for TN, TP, and DO are then estimated using the cat.analysis function. For the combined total, the percentage of samples with all three values (TN, TP, and DO) meeting the threshold is reported.

```
#Example categorical estimation for
TN comb3$TNicat<-comb3$TNcat
TNicat <- cut(comb3$TNicat, breaks=c(0,0.99,1), include.lowest=TRUE)
comb3$TNicat <- TNicat
comb3$TNicat <- as.factor(comb3$TNicat)
nr <- nrow(comb3)

sites <- data.frame(siteID=comb3$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb3$PK_RANDOM_,
CombinedBasins=rep("All Basins",nr),
Basin=comb3$REPORTING_)
```

```
dsgn <- data.frame(siteID=comb3$PK_RANDOM_,
wgt=comb3$wgt,
xcoord=comb3$xmarinus ,
ycoord=comb3$ymarinus ,
stratum=comb3$REPORTING_)

data.catTN <- data.frame(siteID=comb3$PK_RANDOM_,
TNcategory=comb3$TNicat)
CategoryEstTN <- cat.analysis(sites, subpop, dsgn, data.catTN,
popsize=list(CombinedBasins=list("All
Basins"=framesize),
Basin=list("ZONE 1"=framesize,
"ZONE 2"=framesize,
"ZONE 3"=framesize,
"ZONE 4"=framesize,
"ZONE 5"=framesize,
"ZONE 6"=framesize )))

# write out the results
write.table(CategoryEstTN,file='CategoryExampleTNmin_SL.csv',sep=',')
```

References

Copeland, R., Upchurch, S., Summers, K., Janicki, P., Hansard, P., Paulic, P., Maddox, G., Silvanima, J., and Craig, P., 1999, Overview of the Florida Department of Environmental Protection's Integrated Water Resource Monitoring Efforts and the Design Plan of the Status Network: Florida Department of Environmental Protection, Ambient Monitoring Section, 41 p.

Florida Department of Environmental Protection, Bureau of Laboratories, Biology Section. 2013. Holding Time Study for Water Quality Assessments. 44 p.

Appendix A – Example Streams Code

```
# File: 2015-2017 SS analyses with NNC script.R
# Purpose: Illustrate estimation for Florida Streams using R
# (Combined Statewide assessment 2015-2017)
# updated by C. Sedlacek on 6 May 2014
# originally written in S-Plus, transposed to R by C. Sedlacek
# Load psurvey.analysis library using menu
# Packages: Load Packages sp and spsurvey
# edited by C. Sedlacek
# on 7-24-2018

SS.SITES<-SS_EXC_2015_17DONNC
names(SS.SITES)

# convert to Decimal degrees and do map projection
deg <- floor(SS.SITES$RANDOM_LAT/10000)
min <- floor((SS.SITES$RANDOM_LAT - deg*10000)/100)
sec <- SS.SITES$RANDOM_LAT - deg*10000 - min*100
SS.SITES$latdd <- deg + min/60 + sec/3600
deg <- floor(SS.SITES$RANDOM_LON/10000)
min <- floor((SS.SITES$RANDOM_LON - deg*10000)/100)
sec <- SS.SITES$RANDOM_LON - deg*10000 - min*100
SS.SITES$londd <- deg + min/60 + sec/3600

# do equal area projection for variance estimation
tmp <- marinus(SS.SITES$latdd, SS.SITES$londd)
SS.SITES$xmarinus <- tmp[,'x']
SS.SITES$ymarinus <- tmp[,'y']

# check design variables
table(SS.SITES$CAN_BE_SAM)
table(SS.SITES$EXCLUSION_)
table(SS.SITES$EXCLUSIO_1)
table(SS.SITES$EXCLUSION_, SS.SITES$CAN_BE_SAM)
table(SS.SITES$EXCLUSIO_1, SS.SITES$EXCLUSION_)
```

```
# create Status variable
SS.SITES$EXCLUSION_ <- as.character(SS.SITES$EXCLUSION_)
SS.SITES$EXCLUSION_[SS.SITES$CAN_BE_SAM == 'Y'] <- 'SAMPLED'
SS.SITES$EXCLUSION_ <- as.factor(SS.SITES$EXCLUSION_)
levels(SS.SITES$EXCLUSION_)
SS.SITES$STATUS <- SS.SITES$EXCLUSION_
levels(SS.SITES$STATUS) <- list(T=c('SAMPLED', 'NO PERMISSION FROM OWNER', 'UNABLE TO ACCESS','OTHERWISE
UNSAMPLEABLE','OTHERWISE UNSAMPELABLE','DRY'),
                                NT=c('WRONG RESOURCE/NOT PART OF TARGET POPULATION') )

table(SS.SITES$EXCLUSION_, SS.SITES$STATUS)
summary(SS.SITES$STATUS)

##### adjust weights for use of over sample sites
# use only small stream sites that were evaluated: drop non-evaluated sites

# create basin variable
SS.SITES$basin <-  SS.SITES$REPORTING_
table(SS.SITES$basin)

framesize <- c("ZONE 1"=12914.2,"ZONE 2"=2344.1,
               "ZONE 3"=4658.5,"ZONE 4"=4479.1,"ZONE 5"=1104.7,
               "ZONE 6"=197.1) ### change to match combined resource stratum

nr <- nrow(SS.SITES)
SS.SITES$wgt <- adjwgt(!is.nan(SS.SITES$STATUS),
                       SS.SITES$NEST1_WT,
                       SS.SITES$REPORTING_,
                       framesize=framesize )

tapply(SS.SITES$wgt, SS.SITES$REPORTING_, sum)

##### Estimate stream km in population and by SS.SITES category
sites <- data.frame(siteID=SS.SITES$PK_RANDOM_,
                    Use=rep(TRUE,nr))

subpop <- data.frame(siteID=SS.SITES$PK_RANDOM_,
                     Combined=rep("All Basins",nr),
                     basin=SS.SITES$REPORTING_ )

dsgn <- data.frame(siteID=SS.SITES$PK_RANDOM_,
                   wgt=SS.SITES$wgt,
                   xcoord=SS.SITES$xmarinus ,
                   ycoord=SS.SITES$ymarinus ,
                   stratum=SS.SITES$REPORTING )
```

```
data.cat <- data.frame(siteID=SS.SITES$PK_RANDOM_,
                       EXCLUSION.CATEGORY=SS.SITES$EXCLUSION_,
                       TNTStatus=SS.SITES$STATUS)

software.program<-'R Studio'

ExtentEst <- cat.analysis(sites, subpop, dsgn, data.cat,
                          popsize=list(Combined=list("All Basins"=framesize),
                                       basin=list("ZONE 1"=framesize,"ZONE 2"=framesize,
                                                  "ZONE 3"=framesize,"ZONE 4"=framesize,"ZONE 5"=framesize,
                                                  "ZONE 6"=framesize )),
                          conf=95 )

###write out or export results
write.table(ExtentEst,file='ExtentEst.csv',sep=',')

###### Do summaries for data
# import data
SS<-SS_RESULTS_2015_17
names(SS)

# look at sample type
summary(SS$Sample.Type)

# check for duplicate data
table(SS$Station.Name, SS$Sample.Type)

# Note that have BLANK, BOTTOM and PRIMARY data in original spreadsheet.
# hence need to subset before doing analyses
keep <- SS$Sample.Type == 'PRIMARY'

# merge with SS.SITES
comb <- merge(SS.SITES, SS[keep,], by.x='PK_RANDOM_',
              by.y='Station.Name')

# check that have only PRIMARY data
summary(comb$Sample.Type)

####
names(comb)

##### Continuous indicator estimation, generates CDFs
##### for 'Oxygen..Dissolved..Field','pH..Field', ‘Escherichia.Coli.Quanti.Tray’, Chlorophyll.A..Monochromatic','UIA'
nr <- nrow(comb)
levels(comb$STATUS)
```

```
sites <- data.frame(siteID=comb$PK_RANDOM_,
                    Use=comb$STATUS=='T' )

subpop <- data.frame(siteID=comb$PK_RANDOM_,
                     CombinedBasins=rep("All Basins", nr),
                     Basin=comb$REPORTING_ )

dsgn <- data.frame(siteID=comb$PK_RANDOM_,
                   wgt=comb$wgt,
                   xcoord=comb$xmarinus ,
                   ycoord=comb$ymarinus ,
                   stratum=comb$REPORTING_)

data.cont <- data.frame(siteID=comb$PK_RANDOM_,
                   comb[,c('Oxygen..Dissolved..Field','pH..Field','Escherichia.Coli.Quanti.Tray','Chlorophyll.A..Monoch
                   romatic','UIA')])

ExampleEst <- cont.analysis(sites, subpop, dsgn, data.cont,
                            popsize=list(CombinedBasins=list("All Basins"=framesize),
                                         Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                    "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                    "ZONE 5"=framesize, "ZONE 6"=framesize)))
# write out the results
write.table(ExampleEst$CDF,file='ExampleEstCDF.csv',sep=',')
write.table(ExampleEst$Pct,file='ExampleEstPCT.csv',sep=',')
write.table(ExampleEst$Tot,file='ExampleEstTOT.csv',sep=',')

##############################################################################################
################### 305b categorical estimation (streams)#####################################
##############################################################################################

###_____________________________________________________________________________
### Script to do categorical estimation for E.coli
X31616cat <- cut(comb$Escherichia.Coli.Quanti.Tray, breaks=c(0,410,10000000), include.lowest=TRUE)
comb$X31616cat <- X31616cat
comb$X31616cat <- as.factor(comb$X31616cat)

nr <- nrow(comb)
sites <- data.frame(siteID=comb$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb$REPORTING_)
```

```
dsgn <- data.frame(siteID=comb$PK_RANDOM_,
                   wgt=comb$wgt,
                   xcoord=comb$xmarinus ,
                   ycoord=comb$ymarinus ,
                   stratum=comb$REPORTING_)

data.cat31616 <- data.frame(siteID=comb$PK_RANDOM_,
                            X31616category=comb$X31616cat)

CategoryEst31616 <- cat.analysis(sites, subpop, dsgn, data.cat31616,
                                 popsize=list("CombinedBasins"=list("All Basins"=framesize),
                                              Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                         "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                         "ZONE 5"=framesize, "ZONE 6"=framesize)))

# write out the results
write.table(CategoryEst31616,file='CategoryExampleE_coli.csv',sep=',')

###_____________________________________________________________________________
### Script to do categorical estimation for pH (param code 406)
X406cat <- cut(comb$pH..Field, breaks=c(0,5.999,8.5,14), include.lowest=TRUE)
comb$X406cat <- X406cat
comb$X406cat <- as.factor(comb$X406cat)

nr <- nrow(comb)

sites <- data.frame(siteID=comb$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb$REPORTING_)

dsgn <- data.frame(siteID=comb$PK_RANDOM_,
                   wgt=comb$wgt,
                   xcoord=comb$xmarinus ,
                   ycoord=comb$ymarinus ,
                   stratum=comb$REPORTING_)

data.cat406 <- data.frame(siteID=comb$PK_RANDOM_,
                          X406category=comb$X406cat)
```

```
CategoryEst406 <- cat.analysis(sites, subpop, dsgn, data.cat406,
                               popsize=list("CombinedBasins"=list("All Basins"=framesize),
                                            Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                       "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                       "ZONE 5"=framesize, "ZONE 6"=framesize)) )

# write out the results
write.table(CategoryEst406,file='CategoryExample406pH.csv',sep=',')

###_____________________________________________________________________________
#Compare Chl values to threshold
### Pass=1 AND Fail=0
#Pass if chl result value is >= 20 ug/L.
#Fail if chl result value is < 20 ug/L.

comb2$Chlpf<- ifelse((comb2$Chlorophyll.A..Monochromatic >= 20) ,0,1)

###_____________________________________________________________________________
### Script to do categorical estimation for chl
X32211cat <- cut(comb2$Chlpf, breaks=c(0,0.99,1), include.lowest=TRUE)
comb2$X32211cat <- X32211cat
comb2$X32211cat <- as.factor(comb2$X32211cat)

# comb2$basin<-comb2$REPORTING.UNIT
nr <- nrow(comb2)

sites <- data.frame(siteID=comb2$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb2$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb2$basin)

dsgn <- data.frame(siteID=comb2$PK_RANDOM_,
                   wgt=comb2$wgt,
                   xcoord=comb2$xmarinus ,
                   ycoord=comb2$ymarinus ,
                   stratum=comb2$basin)

data.cat32211 <- data.frame(siteID=comb2$PK_RANDOM_,
                            X32211category=comb2$X32211cat)
```

```
CategoryEst32211 <- cat.analysis(sites, subpop, dsgn, data.cat32211,
                                 popsize=list(CombinedBasins=list("All Basins"=framesize),
                                              Basin=list("ZONE 1"=framesize ,"ZONE 2"=framesize ,
                                                         "ZONE 3"=framesize ,"ZONE 4"=framesize ,
                                                         "ZONE 5"=framesize ,"ZONE 6"=framesize  ) ) )

# write out the results
write.table(CategoryEst32211,file='Chlorophylla_pass_fail.csv',sep=',')

###
##Start here to process the data for NNC results!

## Perform analysis with a copy of the data file to ensure that you do not corrupt original merged data
comb2<-comb
names(comb2)

#Calculate TN from TKN and NOx
comb2$TN<-(comb2$Kjeldahl.Nitrogen..Total..as.N+comb2$Nitrate.Nitrite..Total..as.N)

#Compare NNC and DO values to thresholds
### Pass=1 AND Fail=0
#TN
#Pass if TN threshold value is >= TN result value.
#Fail if TN threshold value is < TN result value.
comb2$TNcat<-ifelse((comb2$TN_NNC >= comb2$TN),1,0)
#TP
#Pass if TP threshold value is >= TP result value.
#Fail if TP threshold value is < TP result value.
comb2$TPcat<-ifelse((comb2$TP_NNC >= comb2$Phosphorus..Total..as.P),1,0)
#DO
#Fail if DO threshold value is > DO result value.
#Pass if DO threshold value is =< DO result value.

comb2$DOcat<-ifelse((comb2$Oxygen..Dissolved.Percent.Saturation >= comb2$DO_conc ),1,0)

# Total
comb2$Tot<-(comb2$TNcat+comb2$TPcat+comb2$DOcat)

##### Example continuous indicator estimation
nr <- nrow(comb2)
levels(comb2$STATUS)

sites <- data.frame(siteID=comb2$PK_RANDOM_,
                    Use=comb2$STATUS=='T' )
```

```
subpop <- data.frame(siteID=comb2$PK_RANDOM_,
                     CombinedBasins=rep("All Basins", nr),
                     Basin=comb2$basin )

dsgn <- data.frame(siteID=comb2$PK_RANDOM_,
                   wgt=comb2$wgt,
                   xcoord=comb2$xmarinus ,
                   ycoord=comb2$ymarinus ,
                   stratum=comb2$basin)

data.cont <- data.frame(siteID=comb2$PK_RANDOM_,
                        comb2[,c('TNcat','TPcat','DOcat','Tot')])

ExampleEst2 <- cont.analysis(sites, subpop, dsgn, data.cont,
                             popsize=list(CombinedBasins=list("All Basins"=framesize),
                                          Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                     "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                     "ZONE 5"=framesize, "ZONE 6"=framesize )))
# write out the results
write.table(ExampleEst2$CDF,file='ExampleEst2CDFWR_SS.csv',sep=',')
write.table(ExampleEst2$Pct,file='ExampleEst2PCTWR_SS.csv',sep=',')
write.table(ExampleEst2$Tot,file='ExampleEst2TOTWR_SS.csv',sep=',')

##############################################################################################
################### 305b categorical estimation (streams)#####################################
##############################################################################################

###_____________________________________________________________________________
### Script to do categorical estimation for Total Nitrogen
comb2$TNicat<-comb2$TNcat
TNicat <- cut(comb2$TNicat, breaks=c(0,0.99,1), include.lowest=TRUE)
comb2$TNcat <- TNicat
comb2$TNcat <- as.factor(comb2$TNcat)

nr <- nrow(comb2)

sites <- data.frame(siteID=comb2$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb2$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb2$basin)
```

```
dsgn <- data.frame(siteID=comb2$PK_RANDOM_,
                   wgt=comb2$wgt,
                   xcoord=comb2$xmarinus ,
                   ycoord=comb2$ymarinus ,
                   stratum=comb2$basin)

data.catTN <- data.frame(siteID=comb2$PK_RANDOM_,
                         TNcategory=comb2$TNcat)

CategoryEstTN <-  cat.analysis(sites, subpop, dsgn, data.catTN,
                               popsize=list(CombinedBasins=list("All Basins"=framesize),
                                            Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                       "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                       "ZONE 5"=framesize, "ZONE 6"=framesize)))
# write out the results
write.table(CategoryEstTN,file='CategoryExampleTNicatSS.csv',sep=',')

###_____________________________________________________________________________
### Script to do categorical estimation for TP
comb2$TPicat<-comb2$TPcat
TPicat <- cut(comb2$TPicat, breaks=c(0,0.99,1), include.lowest=TRUE)
comb2$TPcat <- TPicat
comb2$TPcat <- as.factor(comb2$TPcat)

nr <- nrow(comb2)

sites <- data.frame(siteID=comb2$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb2$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb2$basin)

dsgn <- data.frame(siteID=comb2$PK_RANDOM_,
                   wgt=comb2$wgt,
                   xcoord=comb2$xmarinus ,
                   ycoord=comb2$ymarinus ,
                   stratum=comb2$basin)

data.catTP <- data.frame(siteID=comb2$PK_RANDOM_,
                         TPcategory=comb2$TPcat)
```

```
CategoryEstTP <-  cat.analysis(sites, subpop, dsgn, data.catTP,
                               popsize=list(CombinedBasins=list("All Basins"=framesize),
                                            Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                       "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                       "ZONE 5"=framesize, "ZONE 6"=framesize )))

# write out the results
write.table(CategoryEstTP,file='CategoryExampleTPicatSS.csv',sep=',')

###
### Script to do categorical estimation for DO
comb2$DOicat<-comb2$DOcat
DOicat <- cut(comb2$DOicat, breaks=c(0,0.99,1), include.lowest=TRUE)
comb2$DOcat <- DOicat
comb2$DOcat <- as.factor(comb2$DOcat)

nr <- nrow(comb2)

sites <- data.frame(siteID=comb2$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb2$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb2$basin)

dsgn <- data.frame(siteID=comb2$PK_RANDOM_,
                   wgt=comb2$wgt,
                   xcoord=comb2$xmarinus ,
                   ycoord=comb2$ymarinus ,
                   stratum=comb2$basin)

data.catDO <- data.frame(siteID=comb2$PK_RANDOM_,
                         DOcategory=comb2$DOcat)

CategoryEstDO <-  cat.analysis(sites, subpop, dsgn, data.catDO,
                               popsize=list(CombinedBasins=list("All Basins"=framesize),
                                            Basin=list("ZONE 1"=framesize, "ZONE 2"=framesize,
                                                       "ZONE 3"=framesize, "ZONE 4"=framesize,
                                                       "ZONE 5"=framesize, "ZONE 6"=framesize )))

# write out the results
write.table(CategoryEstDO,file='CategoryExampleDOicatSS.csv',sep=',')
```

```
###_____________________________________________________________________________
### Script to do categorical estimation on overall total water quality (TN + TP + DO)
comb2$Totcat<-comb2$Tot
Totcat <- cut(comb2$Totcat, breaks=c(0,0.99,1.99,2.99,5), include.lowest=TRUE)
comb2$Totcat <- Totcat
comb2$Totcat <- as.factor(comb2$Totcat)

nr <- nrow(comb2)

sites <- data.frame(siteID=comb2$PK_RANDOM_, Use=rep(TRUE,nr) )

subpop <- data.frame(siteID=comb2$PK_RANDOM_,
                     CombinedBasins=rep("All Basins",nr),
                     Basin=comb2$basin)

dsgn <- data.frame(siteID=comb2$PK_RANDOM_,
                   wgt=comb2$wgt,
                   xcoord=comb2$xmarinus ,
                   ycoord=comb2$ymarinus ,
                   stratum=comb2$basin)

data.catTot <- data.frame(siteID=comb2$PK_RANDOM_,
                          Totcategory=comb2$Totcat)

CategoryEstTot <-  cat.analysis(sites, subpop, dsgn, data.catTot,
                                popsize=list(CombinedBasins=list("All Basins"=framesize),
                                             Basin=list("ZONE 1"=framesize,
                                                        "ZONE 2"=framesize,
                                                        "ZONE 3"=framesize,
                                                        "ZONE 4"=framesize,
                                                        "ZONE 5"=framesize,
                                                        "ZONE 6"=framesize )))

# write out the results
write.table(CategoryEstTot,file='CategoryExampleToticat_SS.csv',sep=',')

#########
```

Appendix B - Data Qualifiers

Value Qualifiers [from 2017 QA Rule 62-160.700 Table 1 (Data Qualifier Codes)]

Symbol	*Meaning*
A	*Value reported is the arithmetic mean (average) of two or more determinations.*
B	*Results based upon colony counts outside the acceptable range. This code applies to microbiological tests and specifically to membrane filter colony counts. The code is to be used if the colony count is generated from a plate in which the total number of coliform colonies is outside the method indicated ideal range.*
G	*Indicates that the analyte was detected at or above the method detection limit in both the sample and the associated field collected blank, and the value of the blank is greater than 10% of the associated sample value.*
I	*The reported value is greater than or equal to the laboratory method detection limit but less than the laboratory practical quantification limit.*
J	*Estimate value. Shall be accompanied by a detailed explanation to justify the reason(s) for designating the value as estimated. Examples of situations in which a "J" code must be reported include: instances where a quality control item associated with the reported value failed to meet the established quality control criteria (the specific failure must be identified); instances when the sample matrix interfered with the ability to make any accurate determination; instances when data are questionable because of improper laboratory or field protocols (e.g., composite sample was collected instead of a grab sample); instances when the analyte was detected at or above the method detection limit in an analytical laboratory blank other than the method blank (such as calibration blank) and the value the blank is greater than 10% of the associated sample value; or instances when the field or laboratory calibrations or calibration verifications did not meet calibration acceptance criteria.*
K	*Off-scale low. The actual value is known to be less than the value given. (Only used for lab analyses.)*
L	*Off-scale high. The actual value is known to be greater than the value given. (Only used for lab analyses.)*
N	*Presumptive evidence of presence of material; component tentatively identified based on mass spectral library search or there is an indication that the analyte is present, but quality control requirements for the confirmation were not met.*
O	*Sampled but analysis lost or not performed.*
Q	*Sample held beyond the accepted holding time. Value is derived from a sample that was prepared or analyzed after the approved holding time restrictions for sample preparation or analysis.*
R	*Significant rain (typically in excess of ½ inch) in the past 48 hours, which might contribute to a lower or higher than normal value.*
S	*Secchi disk visible to bottom of waterbody. The value reported is the depth of the waterbody at the location of the Secchi disk measurement.*
T	*Value reported is less than the laboratory method detection limit. Value reported for informational purposes only and shall not be used in statistical analysis.*
U	*Indicates that the compound was analyzed for but not detected. The reported value shall be the method detection limit.*

Symbol	*Meaning*
V	*Indicates that the analyte was detected at or above the method detection limit in both the sample and the associated method blank and the blank value was greater than 10% of the associated sample value.*
X	*Indicates, when reporting results from a Stream Condition Index Analysis (LT 7200 and FS 7420), that insufficient individuals were present in the sample to achieve a minimum of 280 organisms for identification (the method calls for two aliquots of 140-160 organisms), suggesting either extreme environmental stress or a sampling error.*
Y	*The laboratory analysis was from an unpreserved or improperly preserved sample. The data may not be accurate.*
Z	*Too many colonies were present for accurate counting. Historically, this condition has been reported as "too numerous to count" (TNTC). The "Z" qualifier code shall be reported when the total number of colonies of all types is more than 200 in all dilutions of the sample tested using a membrane filter technique. When applicable to the observed test results, a numeric value for the colony count for the microorganism tested may be estimated from the highest dilution factor (smallest sample volume) used for the test and reported with the qualifier code.*
!	*Indicates that the reported value deviates from historically established concentration ranges.*
?	*Data are rejected and should not be used. Some or all of the quality control data for the analyte were outside criteria, and the presence or absence of the analyte cannot be determined from the data.*

Note: italicized descriptions deviate from EPA and/or DEP QAS descriptions.

Missing Values: blank

*Notes: Historically, the W qualifier was used in the following ways. 1) If turbidity is greater than 100 NTU, all analytes coming from that well will be qualified with a W. 2) If the well currently has, or historically had, a water level recording device employing a lead weight, all lead values coming from that well will be qualified with a W. 3) All VOC's will be qualified for each glued PVC well. 4) The following detections of analytes coming from galvanized steel wells will be qualified with a W: iron, manganese, zinc, cadmium. 5) The following detections of analytes coming from stainless steel wells will be qualified with a W: nickel, chromium. 6) All detections of trace metals coming from any type of iron well will be qualified with a W.

Appendix C – Example Surface Water Site Info File

Column No.	*Column Label*	*Correct Format*
1	Pk.random.sample.location	factor
2	Fk.epa.random	factor
3	Nest1.Wt	double
4	Epa.oversample	double
5	Epa.division	double
6	Resource.type	factor
7	Reporting.unit	factor
8	Random.latitude	double
9	Random.longitude	double
10	Hydrologic.unit.code	factor
11	Hydrologic.unit.name	factor
12	Can.be.sampled	factor
13	Exclusion.category	factor
14	Exclusion.criteria	factor
15	Sampled.date	factor
16	Hectares	double
17	Analysis.suite	factor
18	Locational.datum	factor
19	Fk.project	factor
20	Status*	factor
21	lat_dec	double
22	long_dec	double
23	xmarinus*	double
24	ymarinus*	double
25	TNT*	factor
26	wgt*	double

*These columns were added by the FLLakesAnalysis execution rather than being part of the original data.

Appendix D – Quality Assurance Checklist

This checklist is used to independently review the results of the Status Network analysis. The example checklist below is for the 2015-2017 combined analyses. The reviewer must fill in the resource and date that the checklist is being completed. Any items of concern noted during the independent review are communicated to the data analyst. The data analyst investigates the items noted, applies any necessary corrections, and performs the analyses again if needed.

2015-2017 Status Network Analysis NNC / DO – QA Checks – (List Water Resource) – (List Date Checklist Last Updated)

Original data files - Exclusions

- Check that weights seem reasonable (same order of magnitude, etc.).
- Check that all "Can be sampled = N" have exclusion category and exclusion criteria listed.
- Check that all "Can be sampled = Y" do not have exclusion category or exclusion criteria.
- Check that there are no sites missing (e.g. "Can be sampled = NA") up through highest selection sampled in each zone.

Combined data files - Exclusions

- Check that exclusion data from individual years was completely transferred to combined exclusion file.

Resource	# Sites 2015	# Sites 2016	# Sites 2017	Sum of # Sites from 3 Individual Years	# Sites 2015-2017 in Combined File

- Use GIS to check that all sites in combined file are located in waterbodies present in most recent year's coverage that was used for site selections. Use spatial query: for lakes site must be within the lake polygon, for flowing waters site must be within 50m of the line representing the waterbody. Manually inspect any sites that do not match spatial query requirements before flagging sites for removal.
- If there are sites present in individual years' files, but missing from combined file, investigate why these were removed. If removed because they were located in waterbodies not present in most recent year's coverage, check if these removed sites seemed like permanent exclusions.
- Check that TN, TP, & DO thresholds in combined file match thresholds in WMS Design Document.

Original data files – Results

- Check that data was provided for all sampled sites.

Year	Total Sites Sampled from Combined Exclusion File	Total Sites with Data in Combined Result File
2015		
2016		
2017		

Combined data files - Results

- Check that result data from individual years was completely transferred to combined result file.

Resource	# Sampled Sites 2015	# Sampled Sites 2016	# Sampled Sites 2017	Sum of # Sampled Sites from 3 Individual Years	# Sampled Sites in 2015-2017 Combined File

Final Results

- Check that number of values (N) used in analysis matches N in combined data file.
- Check that thresholds in R script / output file match thresholds in WMS Design Document.
- Check that weights used in analysis seem reasonable (same order of magnitude, etc.)
- Check that total N in R output file (ExtentExt.csv) matches combined exclusion data file.

Zone	Total N from combined exclusion data file	Total N from ExtentExt.csv output file
All Zones (Statewide Analysis)		

- Check that R output results match results in final report (Word document).