Data Analysis Procedures for Status Assessments, Version 6.0

Guidelines for Florida's Probabilistic Monitoring Network

Division of Environmental Assessment and Restoration Florida Department of Environmental Protection May 2025

2600 Blair Stone Rd, MS 3560 Tallahassee, Florida 32399-2400 floridadep.gov



Table of Contents

Introduction
Part I – The Status Network
Questions Addressed by the Status Network4
Part II – Data Extraction
Part III – Data Quality Assessments
Errors in the Data5
Outliers6
Missing Values
Detection Limits
Qualifier Codes
Part IV – Combining Data7
Combining Data from Multiple Years7
Combining Data from Multiple Resources8
Part V – Analysis Procedures
Running the R scripts9
Explanations for Each Portion of Script:9
Analysis for Indicator Thresholds Dependent on Geography and other Variables21
References
Appendix A – Example Streams Code25
Appendix B - Data Qualifiers
Appendix C – Quality Assurance Checklist

Introduction

The probabilistic design of the Status Network utilizes an unbiased data set for the purpose of answering water-resource questions with known mathematical confidence. The Network allows DEP to address 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: 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). 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 questions related to small drainage basins, counties, or individual bodies of water. The Integrated Water Resource Monitoring Design¹ (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 current design allows for annual assessments of seven water resource types: Rivers, Streams, Canals, Large and Small Lakes, Confined and Unconfined Aquifers (Florida Department of Environmental Protection. 2024).

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?

This 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, large lakes, and explains how the computer code used for Status Network analysis works.

¹ The Integrated Water Resource Monitoring Program is sometimes referred to by the acronym, "IWRM" (Copeland et. al. 1999)

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 <u>https://floridadep.gov/dear/water-quality-assessment/content/reports-documents-sops-and-links</u> for documentation on the design, sample collection methods, data management procedures, etc.

Questions Addressed by the Status Network

Each year the results are reported for analyses conducted on the data on a statewide basis. A minimum of three years the data are combined. The results are reported for analyses performed on: (1) the statewide data and (2) the data in each of six reporting zones (based on the five water management districts with South Florida 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. The minimum sample size needed to report is 23. Appendix A gives example code (in R) for the streams resource. The following types of statistics can be assessed:

- What is the accessible proportion of a resource (surface or groundwater)?
- 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 groundwater (i.e., as confined and unconfined aquifers) do not meet water quality thresholds?
- For flowing surface water resources, what proportion of dissolved oxygen, total nitrogen, total phosphorus, *Escherichia coli*, pH, total ammonia nitrogen, chlorophyll a, and habitat metrics do not meet water quality thresholds?
- For groundwater resources, what proportion of arsenic, cadmium, chromium, lead, nitrate+ nitrite, sodium, fluoride, and total coliform 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 using the FDEPgetdata R package. This R package (version 2.02 developed 2024-10-11) was developed *Page 4 of 39* by WMS staff and is saved on a DEP server (\\floridadep\data\dear\wqap\sol_z\R software\Packages). 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 reconnaissance information (also referred to as the site evaluation data) is created using one of the exclusion functions from the FDEPgetdata package (getdata_fw_exclusions for flowing surface waters; getdata_lake_exclusions for lake surface waters; getdata_aq_exclusions or getdata_aq_exclusions_multi_yr for groundwater). These functions create an R data frame via an oracle data pull from the SITE_EVALUATIONS table of GWIS. For surface waters, the data retrieved from the SITE_EVALUATIONS table include applicable total nitrogen (F.A.C. 62-302.531), total phosphorus (F.A.C. 62-302.531), and / or dissolved oxygen (F.A.C. 62-302.533) criteria based on the corresponding nutrient watershed region and bioregion.

The file containing the field and analytical results (from here on in referred to as the 'results file') is created using the getdata_results function from the FDEPgetdata package. This function creates an R data frame via an oracle data pull from GWIS for the water resource(s) and year(s) specified by the user. The data retrieved includes 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. In addition to retrieving the data, this function replaces measurement values in the R data frame with 'NA' for those measurements having any of the following fatal data qualifiers: '?,O,N,T,X'. Definitions for these codes may be found in FS 62-160.700 Table 1 (Data Qualifier Codes).

The Analysis and Reporting Coordinator notifies the data analyst when all data for each water resource and year are complete in GWIS and ready to be extracted using the functions described above. The site evaluation data are 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 with 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 C 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 total phosphorus). 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 anomalies 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 of a data distribution, 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

The Status Network sampling design for one cycle (one calendar year) calls for 15 collection points per zone for each surface water resource and 20 for each groundwater 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. For indicators with water quality thresholds, measurement values are replaced with 'NA' during Status Network data analysis if the reported values are both above the threshold value (e.g. > 4 MPN/100 mL for total coliform) and are listed as below the method detection limit ('U' qualifier).

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. It is recommended that qualified data remain in the analysis, except for data with fatal qualifiers (?,O,N,T,X) that are removed during data extraction as described above (Part II – Data Extraction). Clear documentation of the reasons for the 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 DEP 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 inhouse study of the Biology Section of the lab showing little degradation of *Escherichia coli* samples up to 30 hours outside holding times (Florida Department of Environmental Protection. 2013).
- 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 $\pm < 10\%$.

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, site evaluation data and results files for each of the three years are merged to provide a combined site evaluation R data frame and a combined results R data frame. The FDEPgetdata R package functions described in Part II – Data Extraction perform the data merging as part of the data retrieval process when the user specifies multiple water resources or years. These two combined data sets are checked to ensure that all data fields of interest are populated for all three years, the site evaluations data may need to be reextracted after the GWIS SITE_EVALUATIONS table is updated by the Data Coordinator. If there are data fields in the results file that are missing values for one or more years (usually due to an analyte being dropped from or added to the analyte list), a decision must be made regarding whether to include those analytes in the analysis. It is desirable to keep as many analytes as possible in the final three-year result file.

The next step is to make sure the wells or waterbody segments in the combined site evaluations data 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 wells or 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. WMS staff have developed several tools to accomplish these checks. The getdata fw site removals and

getdata_lake_site_removals functions in the FDEPdetdata package are used to identify sites to be removed when performing multi-year surface water analyses. The getdata_aq_exclusions_multi_yr function in the FDEPgetdata package is used to identify wells to be removed when performing multi-year groundwater analyses.

The extent of the water resources from the final year's water resource coverage (flowing water length in kilometers (km) or lake area in hectares (ha)) are derived during the site selection process from the shapefile used and are recorded in the accompanying design document (initially saved on DEP server \\floridadep\data\dear\wqap\sol z\Status Sample Survey Designs and then copied to the appropriate folder in \\floridadep\data\dear\wqap\sol z\Status Sample Survey Designs and then copied to the appropriate folder in \\floridadep\data\dear\wqap\sol z\Status Sample Survey Designs and then copied to the appropriate folder in \\floridadep\data\dear\wqap\sol z\Status Sample Survey Designs and then resource in each zone is imported from a .CSV file, created 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 - rivers, streams, and canals, and lakes - large and small). 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 in zones 1-6, streams in zones 1-6, and canals in zones 3-6) 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 the framesize to recalculate the weightings.

When combining the lakes water resources, the areas calculated from the coverages for both resources (large lakes and small lakes) are summed to obtain the extent (total lakes area in hectares) for the combined lakes resource (LLSL). These are the areas used in the R software script as the framesize to recalculate the weightings.

Part V – Analysis Procedures

The following is an example of data analysis on Status Network large lakes data from 2022-2024. An R script is used to generate 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 \\floridadep\data\dear\wqap\sol z\data analysis\2006_305B\Old SPLUS Codes\305b 2006-all codes\HomeDesktop\Suwannee River\Large 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 2024.4.2 build 764, using R Statistical Software version 4.4.1, 64-bit platform.

Start R Studio. Create a new R project, saved to the DEP server data analysis folder (\\floridadep\data\dear\wqap\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. Read through the script and make minor edits as needed to reflect the year and resource being analyzed.

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, comment lines are specified by # symbols preceding text. They will not execute when the script is run. They are descriptive only). Comments are updated each time that the R script is run, to reflect the data analyst's notes about the data or analysis results.

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.

```
# Code developed using R version 4.4.1 (2024-06-14), spsurvey version 5.5.1,
# FDEPgetdata version 2.02.
# Load libraries for the data analyses
library(FDEPgetdata)
library(spsurvey)
library(sqldf)
```

Determine if the required packages are installed. The FDEPgetdata package can be installed using the install package from Package Archive File tool in the R studio tools menu. The spsurvey and sqldf packages can be downloaded from the Comprehensive R Archive Network (CRAN) contributed packages repository (<u>https://cran.r-project.org/web/packages/index.html</u>) or archives (<u>https://cran.r-project.org/web/packages/index.html</u>) or archives (<u>https://cran.r-project.org/web/packages/index.html</u>). After confirming that the required packages are installed load them for analysis.

Run function of FDEPgetdata package which will pull exclusion data. # Insert variable name between parentheses in function call below. FDEPgetdata::getdata_lake_exclusions("'LL22','LL23','LL24'")

Extract the site evaluation data from GWIS (contact WMS Data Coordinator for access credentials). Function getdata_lake_exclusions creates a data frame named 'Exclusions' from the information provided. Corresponding Nutrient watershed region and bioregion are incorporated for each site.

Create a copy of LL_SITES_NOT_INTERSECT and drop sf object formatting. LL_sitesNOTintersects <- st_drop_geometry(LL_SITES_NOT_INTERSECTS)</pre>

Determine if any of the sites fall on lake polygons which are no longer included in the target population. Function getdata_lake_site_removals creates a sf object data frame containing the sites that should be removed from this analysis because they no longer are in the target population. Save a copy of these data with the sf object formatting removed. Name the reformatted data frame "LL sitesNOTintersects".

Note that once data have been loaded into the current R workspace they are then stored within R and are ready to be used in other commands. Once the site removals data (LL_sitesNOTintersects) have been created, a new site evaluations data frame can be created by removing the sites contained in LL_sitesNOTintersects.

```
# Create a copy of the SiteEvaluations data frame for use in next steps of
analysis. New data frame is named LL.SITES.
LL.SITES<-SiteEvaluations
names(LL.SITES)
# Convert to Decimal degrees
deg <- floor(LL.SITES$RANDOM LATITUDE/10000)</pre>
min <- floor((LL.SITES$RANDOM LATITUDE - deg*10000)/100)</pre>
sec <- LL.SITES$RANDOM_LATITUDE - deg*10000 - min*100</pre>
LL.SITES$latdd <- deg + min/60 + sec/3600
deg <- floor(LL.SITES$RANDOM LONGITUDE/10000)</pre>
min <- floor((LL.SITES$RANDOM LONGITUDE - deg*10000)/100)</pre>
sec <- LL.SITES$RANDOM LONGITUDE - deg*10000 - min*100</pre>
LL.SITES$londd <- deg + min/60 + sec/3600
# Change londd to negative for correct use in sf.
LL.SITES$londd <- -LL.SITES$londd
# Create sf object and transform to Albers projection for analysis
  This codes utilizes Coordinate Reference System (CRS/EPSG) Codes.
  The first crs code (4269) below is for NAD 83 coordinate system the
  second crs code (3087) is for Florida albers projection.
# More information on these codes is found here:
# https://www.nceas.ucsb.edu/sites/default/files/2020-
# 04/OverviewCoordinateReferenceSystems.pdf.
dsgn LL <- st as sf(LL.SITES,</pre>
coords = c("londd", "latdd"), remove = FALSE, crs = 4269)
dsgn sf <- st transform(dsgn LL, crs = 3087)
```

```
# keep xy coordinatess as variables
tmp <- st_coordinates(dsgn_sf)
dsgn_sf$xcoord <- tmp[, "X"]
dsgn_sf$ycoord <- tmp[, "Y"]</pre>
```

In order to calculate variance, an equal area map projection must be used. This set of commands transforms the map projection from NAD83 to Florida Albers, creates a matrix, x and y, and assigns this to the site information data frame. Latitude and longitude should be provided in compact degrees, minutes, seconds format (DDMMSS.SSS). If they are provided in a different format, additional commands will be needed to reformat the location data prior to running these commands.

```
# Create a simple features object from a shapefile of polygon features
representing the Zones
# Change projection for Zones sf object to Florida Albers HARN(CRS code 3087).
wms c3 reporting units <- st read(dsn=".",
     layer="Watershed Monitoring Section (WMS) Cycle 3 Reporting Units")
wms c3 reporting units <- st transform(wms c3 reporting units, crs = 3087)
wms_c3_reporting_units
# Use sf to plot the Zone polygons and sites that were evaluated.
jpeg('2022 2024 LL Evaluated Sites.jpg', units = 'in',
     width = 7, height = 7, res = 300)
     plot(st geometry(wms c3 reporting units),
    main= '2022-2024 Large Lake Evaluated Sites')
     plot(st geometry(dsgn sf), pch = 21, bg = 'red', add = TRUE)
     legend(120000, 400000, legend='Zones', col='black', lty=1)
    legend(120000, 300000, legend='Evaluated Sites', col='red',pch=16)
dev.off()
```

A visual inspection of the site location data is performed by creating a simple map that shows the site locations and the reporting units (Zones).

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

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 design as implemented
# Save framesize CSV file to R project workspace, then load framesize data into
# R project.
framesize.df <- read.csv('2024 Large Lake Framesize.csv')</pre>
# Reduce framesize to remove total row.
# Change Zone name values to all capital letters.
framesize.df <- framesize.df[framesize.df$Zones !='Sum',]</pre>
framesize.df$Zones <- toupper(framesize.df$Zones)</pre>
# Convert framesize data frame to named vector
framesize <- as.vector(framesize.df$hectares)</pre>
names(framesize) <- framesize.df[,'Zones']</pre>
# View framesize and paste results in comments.
print(framesize)
   ZONE 1 ZONE 2 ZONE 3 ZONE 4 ZONE 5
                                                  ZONE 6
# 18064.8 7581.9 121935.8 43092.9 60545.3 129968.4
sum(framesize)
# Framesize in hectares = 381189.1 for entire data frame.
```

The framesize values are imported from a .CSV file, created from an R software script run for the site selections. The framesize values are the total size of the resource in each reporting unit. The framesize data format is changed to a named vector. This reformatting ensures that the reporting unit names can be used to link the framesize data to the data in the dsgn_sf data frame.

The adjust weights step accounts for oversampling and NT designations. Ideally a selection of 15 lake sites 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 name of the imported framesize file, as well as data columns names, in order for the code to work. Otherwise, calculations for other resources employ nearly identical code.

```
ExtentEst <- cat analysis(dsgn,</pre>
                          vars = c('TNTStatus', 'EXCLUSION.CATEGORY'),
                          subpops = c('Combined', 'Basin'),
                          siteID = 'PK RANDOM_SAMPLE_LOCATION',
                          weight = 'wgt',
                          xcoord = 'xcoord',
                          vcoord = 'ycoord',
                          stratumID = 'Basin',
                          vartype = "Local",
                          conf=95)
# Add data columns to ExtentEst results. These will be used when loading
analysis results to GWIS tables for Status Network analysis results.
analysis date <- as.character(Sys.Date())</pre>
                              ExtentEst <- cbind(ExtentEst,
                              SAMPLE YEAR = '2022-2024',
                              REPORTING CYCLE = '16-18',
                              WATER_RESOURCE = 'LL',
                              ANALYSIS DATE = analysis date,
                              MATRIX = 'WATER',
                              ESTIMATEU UNITS = 'HECTARES')
# export results
write.csv(ExtentEst, file='2022 2024 LL ExtentEst.csv',
         row.names = FALSE)
```

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].

All sites included in the analysis are represented by the data frame "dsgn". The part of the population examined is represented by the cat_analysis function argument "subpops". "Subpops" recombines all the data statewide and can group populations into smaller units (e.g., Reporting Units, Basins, etc.).

Additional cat_analysis function arguments give the survey design used (sites, weights, and stratification). Though x and y coordinates are also specified in the function arguments, 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 (use of a local neighborhood variance calculation in cat_analysis, is specified by the "vartype" argument).

The cat_analysis function argument "vars" gives the data used in the analysis, which is either 1) an exclusion categorical variable, or 2) a TNT variable.

The data frame "ExtentEst" provides an estimated extent. For example, an estimate of the total area of dry lakes in Zone 2. If 1.5% of lake area is dry, there would be 114.9 ha of dry lake area in a sample size of 7581.9 ha of lakes.

```
# To estimate the percent of the target population
# that could be sampled, requires that the analysis be restricted to just sites
# in the target population, i.e., TNT = "T"
dsgn <- subset(dsgn, dsgn$TNT == "T")</pre>
```

The analysis is rerun with only sites in the target population included in the analysis. The data frame "ExtentEst Target" provides an estimated extent of the target population.

```
###### Water Quality Data Analysis
# Run function of FDEPgetdata package which will pull result data.
# Insert variable name between parentheses in function call below.
FDEPgetdata::getdata results("'LL22','LL23','LL24'")
```

Extract the result data from GWIS. Function getdata_results creates a data frame named 'Results' from the information provided. Replaces measurement values with 'NA' for those measurements having fatal data qualifiers as described in Part II – Data Extraction.

```
# Create new data frame from the one just created.
LL RSLTS<-Results
names(LL RSLTS)
# Determine sample types in file.
addmargins(table(LL RSLTS$SAMPLE TYPE, LL RSLTS$MATRIX, useNA = 'ifany'))
# Note that original data spreadsheet may have blank, primary and bottom sample
# types and water and sediment matrices.
# Need to subset before doing water quality analyses.
# Keep Primary Water data only.
keep <- LL RSLTS$SAMPLE TYPE == 'PRIMARY' & LL RSLTS$MATRIX == 'WATER'</pre>
# Merge subset of result data with site evaluation data
LL WQ <- merge(as.data.frame(dsgn sf)[, c("PK RANDOM SAMPLE LOCATION",
                 "REPORTING_UNIT", "EXCLUSION_CATEGORY","TNT", "wgt",
"londd", "latdd", "xcoord", "ycoord",
"NUTRIENT_WATERSHED_REGION", "DO_Conc")], LL_RSLTS[keep,],
                 by.x = 'PK RANDOM SAMPLE_LOCATION',
                 by.y = 'FK RANDOM SAMPLE LOCATION')
# Check that merged data includes only PRIMARY WATER data
addmargins(table(LL WQ$SAMPLE TYPE, LL WQ$MATRIX, useNA = 'ifany'))
# Create a second data subset for sediment analyses.
# Keep Primary Sediment data only.
keep2 <- LL RSLTS$SAMPLE TYPE == 'PRIMARY' & LL RSLTS$MATRIX == 'SEDIMENT'</pre>
# Merge sediment result data with site evaluation data
LL SED <- merge(as.data.frame(dsgn sf)[, c("PK RANDOM SAMPLE LOCATION",
                 "REPORTING_UNIT", "EXCLUSION_CATEGORY", "TNT", "wgt",
"londd", "latdd", "xcoord", "ycoord",
                 "NUTRIENT WATERSHED_REGION",
                 "DO Conc")], LL RSLTS[keep2,],
                 by.x = 'PK RANDOM SAMPLE LOCATION',
                 by.y = 'FK RANDOM SAMPLE LOCATION')
```

Check that merged data includes only PRIMARY SEDIMENT data addmargins(table(LL_SED\$SAMPLE_TYPE, LL_SED\$MATRIX, useNA = 'ifany'))

It is necessary to subset the results data file because the SAMPLE_TYPE column often contains three types of data: BLANK, PRIMARY, and BOTTOM. For lakes, the MATRIX column often contains two types of data: WATER and SEDIMENT.

Merging the site evaluations data and results data must happen before water quality analyses can be performed. This ensures that result data from sites that are not part of the target population, and result data that have been flagged as inappropriate for analysis due to quality assurance concerns are removed from the data set.

```
##### Example continuous water quality indicator population estimation
nr <- nrow(LL WQ)</pre>
levels(LL WQ$TNT)
# Data frame LL_WQ is the merged dsgn_sf and LL_RESULTS data.
# Add Combined category with name "All Basins" and added Basin category
# with reporting unit data.
data cont WQ <- data.frame(LL WQ,</pre>
                                  Combined = rep("All Basins", nr),
                                  Basin = LL WQ$REPORTING UNIT)
# List all variables for continuous analysis
ContVars <- c('Water Temperature', 'pH Field', 'Oxygen Dissolved Field',
                  'Oxygen Dissolved Percent Saturation',
                 'Specific Conductance Field', 'TN', 'Phosphorus Total as P',
                 'Ammonia Total as N', Escherichia Coli Quanti Tray',
                 'NitrateNitrite Total as N', 'Kjeldahl Nitrogen Total as N',
                 'Chlorophyll_A_Monochromatic','Alkalinity_Total_as_CaCO3',
'Total_Suspended_Solids_TSS', 'Organic_Carbon_Total',
                 'Turbidity_Lab','Chloride_Total','Sodium_Total','Aluminum_Total',
'Antimony_Total','Arsenic_Total','Barium_Total','Beryllium_Total',
'Cadmium_Total','Chromium_Total','Copper_Total','Iron_Total',
'Lead_Total','Manganese_Total', 'Molybdenum_total','Nickel_Total',
'Selenium_Total','Silver_Total', 'Thallium_Total','Zinc_Total',
                 'Fluoride Total', 'Anatoxin a', 'Cylindrospermopsin',
                 'Desmethyl_microcystin_LR', 'Microcystin_HilR', 'Microcystin_HtyR',
                 'Microcystin LA', 'Microcystin LF', 'Microcystin LR',
                 'Microcystin LW', 'Microcystin LY', 'Microcystin RR',
                 'Microcystin WR', 'Microcystin YR', 'Nodularin R')
# Run Continuous analysis for all analytes in ContVars.
Water quality Cont <- cont analysis(data cont WQ,
                                            vars = ContVars,
                                            subpops = c('Combined', 'Basin'),
                                            siteID = 'PK RANDOM SAMPLE_LOCATION',
                                            weight = 'wgt',
                                            xcoord = 'xcoord',
                                            ycoord = 'ycoord',
                                            stratumID = 'Basin',
                                            vartype = 'Local',
                                            statistics = c('CDF', 'Pct', 'Mean', 'Total'),
                                            conf=95,
                                            popsize = list(Basin = framesize))
```

```
# Add data columns to all Water quality Cont results data frames. These will be
# used when loading analysis results to GWIS tables for Status Network analysis
# results.
analysis date <- as.character(Sys.Date())</pre>
Water quality Cont <- lapply(Water quality Cont, function(df)
                      cbind(df,
                            SAMPLE YEAR = '2022 - 2024',
                             REPORTING CYCLE = '16-18',
                            WATER RESOURCE = 'LARGE LAKE',
                            ANALYSIS DATE = analysis date,
                            MATRIX = 'WATER'))
# Merge Pct and Mean Results into single data frame, for consistency with
# format of previous years' results.
Water quality Cont[["Mean"]] <- cbind(Water quality Cont[["Mean"]],
                                      Statistic="Mean")
Water quality Cont[["Pct"]] <- rbind(Water quality Cont[["Pct"]],
                                      Water_quality_Cont[["Mean"]])
```

The continuous analysis portion of the code creates percentile and CDF calculations and writes them to tables. The "cont_analysis" function 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 sites listed in the "dsgn" dataframe, and arguments to the cont_analysis function to specify the , subpop, weight, stratumID, etc.

The function above creates CDF estimates and percentile estimates for a range of variables 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.

```
##### Example categorical water quality indicator population estimation
# Set up threshold category variables for specified water quality analytes
# E Coli category
LL WQ$E Coli Category <- as.factor(cut(LL WQ$Escherichia Coli Quanti Tray,
breaks=c(0,410,1000000), include.lowest=TRUE))
# pH Category
LL WQ$pH Category <- as.factor(cut(LL WQ$pH Field, breaks=c(0,5.999,8.5,14),
include.lowest=TRUE))
# Fluoride class III water quality standard exceedances.
LL WQ$Fluoride Category <- as.factor(cut(LL WQ$Fluoride Total, breaks=c(0,10,100000),
include.lowest=TRUE))
#### Metals exceeding class III water quality standards.
# Antimony Category
LL WQ$Antimony Category <- as.factor(cut(LL WQ$Antimony Total,
breaks=c(0,4300,100000), include.lowest=TRUE))
# Arsenic Category
LL WQ$Arsenic Category <- as.factor(cut(LL WQ$Arsenic Total, breaks=c(0,50,100000),
include.lowest=TRUE))
# Beryllium Category
LL WQ$Beryllium Category <- as.factor(cut(LL WQ$Beryllium Total,
breaks=c(0, 0.13, 100000), include.lowest=TRUE))
# Iron Category
LL WQ$Iron Category <- as.factor(cut(LL WQ$Iron Total, breaks=c(0,1000,100000),
```

```
include.lowest=TRUE))
```

```
# Selenium Category
LL WQ$Selenium Category <- as.factor(cut(LL WQ$Selenium Total, breaks=c(0,5,100000),
include.lowest=TRUE))
# Silver Category
LL WQ$Silver Category <- as.factor(cut(LL WQ$Silver Total, breaks=c(0,0.07,100000),
include.lowest=TRUE))
# Thallium Category
LL WQ$Thallium Category <- as.factor(cut(LL WQ$Thallium Total,
breaks=c(0, 6.299, 100000), include.lowest=TRUE))
# categorical water guality estimates
# Data frame LL WQ is the merged dsgn sf and LL RESULTS data.
# Add Combined category with name "All Basins" and add Basin category
# with reporting unit data.
data cat WQ <- data.frame(LL WQ,</pre>
                           Combined = rep("All Basins", nr),
                          Basin = LL WQ$REPORTING UNIT)
# List of all variables for categorical analysis.
CatVars <- subset(names(LL WQ),
                  grepl(" Category$", names(LL WQ)))
# Run categorical analysis for all analytes in CatVars.
Water Quality Cat <- cat analysis(data cat WQ,
                                   vars = CatVars,
                                   subpops = c('Combined', 'Basin'),
                                   siteID = 'PK RANDOM SAMPLE LOCATION',
                                   weight = 'wgt',
                                   xcoord = 'xcoord',
                                  ycoord = 'ycoord',
                                  stratumID = 'Basin',
                                   vartype = 'Local',
                                   conf=95,
                                   popsize = list(Basin = framesize))
# Add data columns to Water Quality Cat results. These will be
# used when loading analysis results to GWIS tables for Status Network analysis
# results.
analysis date <- as.character(Sys.Date())</pre>
Water Quality Cat <- cbind(Water Quality Cat,
                            SAMPLE YEAR = '2022-2024',
                           REPORTING CYCLE = '16-18',
                           WATER_RESOURCE = 'LL',
                           ANALYSIS DATE = analysis_date,
                           MATRIX = 'WATER',
                           ANALYSIS TYPE = ' TARGET POPULATION',
                           ESTIMATEU UNITS = 'HECTARES')
```

The categorical analysis portion of the code creates estimates of the percent of the resource in each of the user-generated categories (i.e. meeting water quality threshold, not meeting water quality threshold) and writes them to a table. The cat_analysis function is used for categorical variables.[Entering "help(cat.analysis)" in the R command window will display the function description].

The program generates four output files which are saved in the folder designated as the current *R* workspace. These are comma delimited text files that can be opened in Excel.

```
###### Set up threshold category columns for specified sediment analytes
# Create categories using Probable Effects Concentration (PEC) for each indicator
# Create categories for combined analysis of all indicators
LL_SED$AsPECcat <- ifelse(LL_SED$Arsenic_Sediments > 33, 1, 0)
LL_SED$AsPECcat[is.na(LL_SED$AsPECcat)] <- 0</pre>
LL SED$CdPECcat <- ifelse(LL SED$Cadmium Sediments > 5, 1, 0)
LL_SED$CdPECcat[is.na(LL_SED$CdPECcat)] <- 0
LL SED$CrPECcat <- ifelse(LL SED$Chromium Sediments > 110, 1, 0)
LL SED$CrPECcat[is.na(LL SED$CrPECcat)] <- 0
LL_SED$CuPECcat <- ifelse(LL_SED$Copper_Sediments > 150, 1, 0)
LL SED$CuPECcat[is.na(LL SED$CuPECcat)] <- 0
LL SED$AgPECcat <- ifelse(LL SED$Silver Sediments > 2.2, 1, 0)
LL SED$AgPECcat[is.na(LL SED$AgPECcat)] <- 0</pre>
LL SED$NiPECcat <- ifelse(LL SED$Nickel Sediments > 49, 1, 0)
LL_SED$NiPECcat[is.na(LL_SED$NiPECcat)] <- 0</pre>
LL SED$PbPECcat <- ifelse(LL SED$Lead Sediments > 130, 1, 0)
LL_SED$PbPECcat[is.na(LL_SED$PbPECcat)] <- 0
LL_SED$HgPECcat <- ifelse(LL_SED$Mercury_Sediments > 1.1, 1, 0)
LL_SED$HgPECcat[is.na(LL_SED$HgPECcat)] <- 0</pre>
LL SED$ZnPECcat <- ifelse(LL SED$Zinc Sediments > 460, 1, 0)
LL SED$ZnPECcat[is.na(LL SED$ZnPECcat]] <- 0
# Combined total PEC categories
LL SED$NumExceedPECcat <- 0
LL_SED$NumExceedPECcat <- ifelse(LL SED$AsPECcat == 0 | is.na(LL SED$AsPECcat),
                          LL SED$NumExceedPECcat, (LL SED$NumExceedPECcat+1))
LL SED$NumExceedPECcat <- ifelse(LL SED$CdPECcat == 0 | is.na(LL SED$CdPECcat),
                          LL SED$NumExceedPECcat, (LL SED$NumExceedPECcat+1))
LL SED$NumExceedPECcat <- ifelse(LL SED$CrPECcat == 0 | is.na(LL SED$CrPECcat),
                          LL_SED$NumExceedPECcat, (LL_SED$NumExceedPECcat+1))
LL_SED$NumExceedPECcat <- ifelse(LL_SED$CuPECcat == 0 | is.na(LL_SED$CuPECcat),
                          LL_SED$NumExceedPECcat, (LL_SED$NumExceedPECcat+1))
LL_SED$NumExceedPECcat <- ifelse(LL_SED$AgPECcat == 0 | is.na(LL_SED$AgPECcat),</pre>
                          LL_SED$NumExceedPECcat, (LL_SED$NumExceedPECcat+1))
LL SED$NumExceedPECcat <- ifelse(LL SED$NiPECcat == 0 | is.na(LL SED$NiPECcat),
                          LL_SED$NumExceedPECcat, (LL_SED$NumExceedPECcat+1))
LL_SED$NumExceedPECcat <- ifelse(LL_SED$PbPECcat == 0 | is.na(LL_SED$PbPECcat),
                          LL_SED$NumExceedPECcat, (LL_SED$NumExceedPECcat+1))
LL SED$NumExceedPECcat <- ifelse(LL SED$HgPECcat == 0 | is.na(LL SED$HgPECcat),
                          LL SED$NumExceedPECcat, (LL SED$NumExceedPECcat+1))
LL SED$NumExceedPECcat <- ifelse(LL SED$ZnPECcat == 0 | is.na(LL SED$ZnPECcat),
                          LL SED$NumExceedPECcat, (LL SED$NumExceedPECcat+1))
LL SED$NumExceedPECcat <- ifelse(is.na(LL SED$AsPECcat) & is.na(LL SED$CdPECcat)
                                & is.na(LL_SED$CrPECcat) & is.na(LL_SED$CuPECcat)
                                & is.na(LL_SED$AgPECcat) & is.na(LL_SED$NiPECcat)
                                & is.na(LL_SED$PbPECcat) & is.na(LL_SED$HgPECcat)
                                & is.na(LL_SED$ZnPECcat), NA,
                                LL SED$NumExceedPECcat)
# Sites that exceed at least one PEC threshold
LL_SED$Exceed1_PECcat <- LL_SED$NumExceedPECcat
LL SED$Exceed1 PECcat[LL SED$Exceed1 PECcat >= 1] <- 1
addmargins(table(NumExceedPECcat = LL SED$NumExceedPECcat,
           Exceed1 PECcat = LL SED$Exceed1 PECcat, useNA = 'ifany'))
##### Sediment Category Population Estimation
# Data frame LL WQ is the merged dsgn sf and LL RESULTS data.
# Add Combined category with name "All Basins" and add Basin category
```

```
# with reporting unit data.
# Rename sediment category variables as needed.
data cat sed <- data.frame(PK RANDOM SAMPLE LOCATION =
                             data cont sed$PK RANDOM SAMPLE LOCATION,
                             Combined = data cont sed$Combined,
                             Basin = data_cont_sed$Basin,
                             wgt = data cont sed$wgt,
                             xcoord = data cont sed$xcoord,
                             ycoord = data cont sed$ycoord,
                             Num Exceed PEC Category = LL SED$NumExceedPECcat,
                             Exceed_1_PEC_Category = LL_SED$Exceed1_PECcat,
                             Arsenic_PEC_Category = LL_SED$AsPECcat,
                             Cadmium_PEC_Category = LL_SED$CdPECcat,
                             Chromium_PEC_Category = LL_SED$CrPECcat,
                            Copper_PEC_Category = LL_SED$CuPECcat,
Silver_PEC_Category = LL_SED$AgPECcat,
Nickel_PEC_Category = LL_SED$NiPECcat,
                             Lead_PEC_Category = LL_SED$PbPECcat,
                             Mercury PEC Category = LL SED$HgPECcat,
                             Zinc PEC Category = LL SEDSIPECcat)
# List all variables for categorical analysis here.
CatVars <- c('Num Exceed PEC Category',
              'Exceed 1 PEC Category',
              'Arsenic_PEC_Category',
              'Cadmium_PEC_Category',
              'Chromium PEC Category',
              'Copper_PEC_Category',
              'Silver_PEC_Category',
'Nickel_PEC_Category',
              'Lead PEC Category',
              'Mercury_PEC_Category',
              'Zinc PEC Category')
# Run categorical analysis for all analytes in CatVars.
Sediment Quality Cat <- cat analysis (data cat sed,
                                 vars = CatVars,
                                 subpops = c('Combined', 'Basin'),
                                 siteID = 'PK RANDOM SAMPLE LOCATION',
                                 weight = 'wgt',
                                 xcoord = 'xcoord',
                                 ycoord = 'ycoord',
                                 stratumID = 'Basin',
                                 vartype = 'Local',
                                 conf=95,
                                 popsize = list(Basin = framesize))
# Add data columns to Sediment Quality Cat results. These will be
# used when loading analysis results to GWIS tables for Status Network analysis
# results.
analysis date <- as.character(Sys.Date())</pre>
Sediment Quality Cat <- cbind(Sediment Quality Cat,
                                SAMPLE YEAR = '2022-2024',
                                REPORTING CYCLE = '16-18',
                                WATER RESOURCE = 'LL',
                                ANALYSIS DATE = analysis date,
                                MATRIX = 'SEDIMENT',
                                ANALYSIS TYPE = ' TARGET POPULATION',
                                ESTIMATEU UNITS = 'HECTARES')
```

```
##### Sediment continuous distribution estimation
# LL SED is the merged dsgn sf and LR RESULTS data.
\# Add Combined category with name "All Basins" and added Basin category
# with reporting unit data.
nr <- nrow(LL SED)</pre>
data cont sed <- data.frame(LL SED,</pre>
                            Combined = rep("All Basins", nr),
                            Basin = LL SED$REPORTING UNIT)
# Create data frame with only sediment analytes to be used for CDFs
ContVars_Sed <- c('Arsenic_Sediments', 'Cadmium_Sediments', 'Chromium_Sediments',</pre>
                 'Aluminum Sediments', 'Antimony Sediments', 'Beryllium Sediments',
                 'Iron_Sediments', 'Manganese_Sediments', 'Molybdenum_Sediments',
                 'Selenium Sediments', 'Copper Sediments', 'Silver Sediments',
                 'Nickel Sediments', 'Lead Sediments', 'Mercury_Sediments',
                 'Zinc Sediments')
# Run Continuous analysis for all analytes in ContVars Sed.
Sediment quality Cont <- cont analysis (data cont sed,
                                         vars = ContVars Sed,
                                         subpops = c('Combined', 'Basin'),
                                         siteID = 'PK RANDOM SAMPLE LOCATION',
                                         weight = 'wg\overline{t}',
                                         xcoord = 'xcoord',
                                         ycoord = 'ycoord',
                                         stratumID = 'Basin',
                                         vartype = 'Local',
                                         statistics =
                                          c('CDF', 'Pct', 'Mean', 'Total'),
                                         conf=95,
                                         popsize = list(Basin = framesize))
# Add data columns to all Sediment_quality_Cont results data frames. These will
# be used when loading analysis results to GWIS tables for Status Network
# analysis results.
analysis date <- as.character(Sys.Date())</pre>
Sediment quality Cont <- lapply(Sediment quality Cont, function(df)</pre>
                                 cbind(df,
                                 SAMPLE YEAR = '2022 - 2024',
                                 REPORTING CYCLE = '16-18',
                                 WATER RESOURCE = 'LARGE LAKE',
                                 ANALYSIS_DATE = analysis_date,
                                MATRIX = 'SEDIMENT'))
# Merge Pct and Mean Results into single data frame, for consistency with
# format of previous years' results.
Sediment quality Cont[["Mean"]] <- cbind(Sediment quality Cont[["Mean"]],</pre>
                                          Statistic="Mean")
Sediment quality Cont[["Pct"]] <- rbind(Sediment quality Cont[["Pct"]],</pre>
                                          Sediment quality Cont[["Mean"]])
# Export the results
write.csv(Sediment Quality Cat, "2022-2024 LL Sed Cat.csv", row.names = FALSE)
write.csv(Sediment quality Cont$CDF, file = '2022-2024 LL Sed Cont EstCDF.csv',
          row.names = FALSE)
write.csv(Sediment quality Cont$Pct, file = '2022-2024 LL Sed Cont EstPCT.csv',
          row.names = FALSE)
write.csv(Sediment quality Cont$Total,
          file = '2022-2024 LL Sed Cont EstTotal.csv', row.names = FALSE)
```

Continuous and categorical analyses can be performed for sediment analytes using the same procedures that were used for the water quality analyses. The program generates four output files which are saved in the folder designated as the current R workspace. 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), chlorophyll (lakes only), total ammonia nitrogen, and several metals standards for class III surface waters (cadmium, chromium, copper, lead, nickel, zinc). To evaluate whether the resultant data from each sampled site meets thresholds that vary based on geographic location, the nutrient region and dissolved oxygen region names and associated threshold values (if applicable) must be included in the site evaluation data.

For flowing waters (rivers, streams, and canals), only geographic information is needed to determine the TN, TP, and DO thresholds. For lakes (large lakes and small lakes), only geographic information is needed to determine the DO thresholds. Additional information, specifically the result values for true color and alkalinity, is needed to determine the TN, TP, and chlorophyll thresholds that apply to lake samples. The result data for each sample is categorized according to its color and alkalinity result values.

```
# True color > 40 PCU assigned 0;
# Ture color <= 40 PCU assigned 1.
LL_WQ$Color_cat<- ifelse((LL_WQ$Color_true > 40) ,0,1)
# Alkalinity > 20 mg/L CaCO3 assigned 0;
# Alkalinity <= 20 mg/L CaCO3 assigned 1.
LL_WQ$Alkalinity_cat<- ifelse((LL_WQ$Alkalinity_Total_as_CaCO3 > 20) ,0,1)
# Combine color and alkalinity categories into single character string
# variable.
LL WQ$Col Alk cat<-paste(LL WQ$Color cat, LL WQ$Alkalinity_cat)</pre>
```

A conditional statement is then used to assign the appropriate TN, TP, DO, and chlorophyll thresholds 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 and reported.

```
## Use new Col Alk cat character variable to assign thresholds for TN, TP, and
# chlorophyll
LL WQ$TN Max<- ifelse(LL WQ$Col Alk cat=="0 0",2.23,
                ifelse(LL WQ$Col Alk cat=="0 1", 2.23,
                ifelse(LL WQ$Col Alk cat== "1 0", 1.91,
                ifelse(LL_WQ$Col_Alk_cat=="1 1",0.93,
               NA))))
LL WQ$TP Max<- ifelse((LL WQ$Color cat==0 &
                   LL WQ$NUTRIENT WATERSHED REGION=="WEST CENTRAL"),0.49,
                   ifelse(LL_WQ$Col_Alk_cat=="0 0",0.16,
                   ifelse(LL_WQ$Col_Alk_cat=="0 1",0.16,
ifelse(LL_WQ$Col_Alk_cat=="1 0", 0.09,
                   ifelse(LL WQ$Col Alk cat=="1 1",0.03,
                   NA)))))
LL WQ$Chlorophyll conc<- ifelse(LL WQ$Col Alk cat=="0 0",20,
                           ifelse(LL WQ$Col Alk cat=="0 1", 20,
                           ifelse(LL WQ$Col Alk cat== "1 0", 20,
                           ifelse(LL WQ$Col Alk cat=="1 1", 6,
                            NA))))
```

After TN, TP, DO, and chlorophyll thresholds have been assigned, the result values for each sample are compared to their respective thresholds. To determine the combined result, it is determined whether all three values (TN, TP, and DO) are meeting their thresholds.

```
# Pass = 1, Fail = 0
# TN
# Calculate TN
LL WQ$TN<-(LL WQ$Kjeldahl Nitrogen Total as N+LL WQ$NitrateNitrite Total as N
# Pass if TN threshold value is >= TN result value.
# Fail if TN threshold value is < TN result value.
LL WQ$TN cat<-ifelse((LL WQ$TN Max >= LL WQ$TN),1,0)
# TP
# Pass if TP threshold value is >= TP result value.
# Fail if TP threshold value is < TP result value.
LL WQ$TP cat<-ifelse((LL WQ$TP Max >= LL WQ$Phosphorus Total as P),1,0)
# DO
# Fail if DO threshold value is > DO result value.
# Pass if DO threshold value is =< DO result</pre>
LL WQ$DO cat<-ifelse((LL WQ$DO Conc >
LL WQ$0xygen Dissolved Percent Saturation),0,1)
# Chlorophyll
LL WQ$Chlorophyll cat<-ifelse((LL WQ$Chlorophyll conc >=
                               LL WQ$Chlorophyll A Monochromatic),1,0)
# Total (TN, TP, DO)
# Note for combined NNc and DO, Pass = 3, Fail < 3
LL WQ$NNCDO cat<-(LL WQ$TN cat+LL WQ$TP cat+LL WQ$DO cat)
```

Total Ammonia Nitrogen (TAN) thresholds are calculated using the single sample criteria equation from the DEP TAN calculator (https://floridadep.gov/dear/water-quality-standards-program/documents/total-ammonia-nitrogen-calculator%C2%A0). Thresholds for cadmium, chromium, copper, lead, nickel, and zinc are calculated using the equations for hardness-based metals criteria from the DEP metals criteria calculator (https://floridadep.gov/dear/water-quality-standards/content/surface-water-quality-support-documents). The result values for pH and water temperature are needed to determine the TAN thresholds, and the result values for total hardness (mg/L as CaCO₃) are needed to determine the metals thresholds. A conditional statement is used to assign the appropriate thresholds to each site. After the threshold has been assigned, the resultant values for each sample are compared to their respective threshold.

```
# TAN pH = pH used in TAN threshold calc.
# If measured pH < 6.5, value used is 6.5.
# If measured pH > 9.0, value used is 9.0.
LL WQ$TAN pH <- ifelse(LL WQ$pH Field < 6.5, 6.5,
                ifelse(LL WQ$pH Field > 9.0, 9.0, LL WQ$pH Field))
# TAN temp = water temperature used in TAN threshold calc.
# If measured temp < 7 degrees C, value used is 7.
LL WQ$TAN temp <- ifelse(LL WQ$Water Temperature < 7, 7,
LL WQ$Water Temperature)
# calculate single sample Total Ammonia Criteria using TAN_pH and
# LL WQ$TAN temp
LL WQ$TAN Crit SingleSamp <- ifelse(is.na(LL WQ$TAN pH), NA,
    ifelse(is.na(LL_WQ$TAN_temp), NA,
    (2.5*(0.8876*((0.0278/(1+10^(7.688-LL WQ$TAN pH)))+
    (1.1994/(1+10^(LL WQ$TAN pH-7.688))))* 2.126*10^(0.028*
    (20-(LL WQ$TAN temp)))))))
# Round result to two decimal places, for consistency with thresholds generated
# by TAN calculator spreadsheet
LL WQ$TAN Crit SingleSamp <- round(LL WQ$TAN Crit SingleSamp, digits=2)
# Pass=1 AND Fail=0
# Pass if TAN threshold value is >= TAN result value.
```

```
# Fail if TAN threshold value is < TAN result value.
LL WQ$TAN Cat<-ifelse((LL WQ$TAN Crit SingleSamp >=
LL_WQ$Ammonia_Total_as_N),1,0)
## Metals Hardness = Harness used in metals threshold calculation. If measured
   Hardness calculated as CACO3 < 25, value used is 25. If measured
#
  Hardness calculated as CACO3 > 400, value used is 400.
#
LL WQ$Metals Hardness <- ifelse(LL WQ$Hardness calculated as CACO3 < 25, 25,
    ifelse(LL WQ$Hardness calculated as CACO3 > 400, 400,
    LL WQ$Hardness calculated as CACO3))
## For cadmium, chromium, copper, lead, nickel, and zinc categories:
# Pass=1 AND Fail=0
# Pass if threshold value is >= result value.
# Fail if threshold value is < result value.</pre>
## Calculate single sample Cadmium Total Criteria using Metals Hardness
LL WQ$TotCadmiumCrit SingleSamp <- ifelse(is.na(LL WQ$Metals Hardness), NA,
    (exp((0.7409*(log(LL WQ$Metals Hardness)))-4.719)))
# Round result to four decimal places
LL WQ$TotCadmiumCrit SingleSamp <- round(LL WQ$TotCadmiumCrit SingleSamp,
digits=4)
# Assign values for Pass (1) and Fail (0).
LL WQ$Cadmium Category<-ifelse((LL WQ$TotCadmiumCrit SingleSamp >=
LL WQ$Cadmium Total),1,0)
## Calculate single sample Chromium_Total Criteria using Metals_Hardness
LL_WQ$TotChromiumCrit_SingleSamp <- ifelse(is.na(LL_WQ$Metals_Hardness), NA,
     (exp((0.819*(log(LL WQ$Metals Hardness)))+0.6848)))
# Round result to four decimal places
LL WQ$TotChromiumCrit SingleSamp <- round(LL WQ$TotChromiumCrit SingleSamp,
digits=4)
# Assign values for Pass (1) and Fail (0).
LL WQ$Chromium Category<-ifelse((LL WQ$TotChromiumCrit SingleSamp >=
LL WQ$Chromium Total),1,0)
## Calculate single sample Copper Total Criteria using Metals Hardness
LL WQ$TotCopperCrit SingleSamp <- ifelse(is.na(LL WQ$Metals Hardness), NA,
     (\exp((0.8545*(\log(LL_WQ$Metals_Hardness)))-1.702)))
# Round result to four decimal places
LL WQ$TotCopperCrit SingleSamp <- round(LL WQ$TotCopperCrit SingleSamp,
digits=4)
# Assign values for Pass (1) and Fail (0).
LL WQ$Copper Category<-ifelse((LL WQ$TotCopperCrit SingleSamp >=
LL_WQ$Copper_Total),1,0)
## Calculate single sample Lead Total Criteria using Metals Hardness
LL WQ$TotLeadCrit SingleSamp <- ifelse(is.na(LL_WQ$Metals_Hardness), NA,
     (exp((1.273*(log(LL WQ$Metals Hardness)))-4.705)))
# Round result to four decimal places
LL WQ$TotLeadCrit SingleSamp <- round(LL WQ$TotLeadCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
LL WQ$Lead Category<-ifelse((LL WQ$TotLeadCrit SingleSamp >=
LL WQ$Lead Total),1,0)
## Calculate single sample Nickel_Total Criteria using Metals_Hardness
LL WQ$TotNickelCrit SingleSamp <- ifelse(is.na(LL WQ$Metals Hardness), NA,
     (exp((0.846*(log(LL_WQ$Metals_Hardness)))+0.0584)))
# Round result to four decimal places
LL WQ$TotNickelCrit SingleSamp <- round(LL WQ$TotNickelCrit SingleSamp,
digits=4)
# Assign values for Pass (1) and Fail (0).
LL_WQ$Nickel_Category<-ifelse((LL_WQ$TotNickelCrit_SingleSamp >=
LL WQ$Nickel Total),1,0)
```

```
## Calculate single sample Zinc_Total Criteria using Metals_Hardness
LL_WQ$TotZincCrit_SingleSamp <- ifelse(is.na(LL_WQ$Metals_Hardness), NA,
        (exp((0.8473*(log(LL_WQ$Metals_Hardness)))+0.884)))
# Round result to four decimal places
LL_WQ$TotZincCrit_SingleSamp <- round(LL_WQ$TotZincCrit_SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
LL_WQ$Zinc_Category<-ifelse((LL_WQ$TotZincCrit_SingleSamp >=
LL_WQ$Zinc_Total),1,0)
```

The percent of the resource meeting the thresholds for TN, TP, DO, chlorophyll, TAN, chromium, copper, lead, nickel, and zinc are then estimated using the cat_analysis function, as described in the previous subsection of this document. For the combined total, the percentage of samples with TN, TP, and DO values meeting their three respective thresholds is reported.

References

- Copeland, R., Upchurch, S., Summers, K., Janicki, P., Hansard, P., Paulic, P., Maddox, G., Silvanima, J., and Craig, P.. 1999. <u>Overview of the Florida Department of Environmental</u> <u>Protection's Integrated Water Resource Monitoring Efforts and the Design Plan of the</u> <u>Status Network</u>: Florida Department of Environmental Protection, Ambient Monitoring Section.
- Florida Department of Environmental Protection. 2013. <u>Holding Time Study for Water Quality</u> <u>Assessments</u>. Tallahassee, FL: Bureau of Laboratories, Biology Section. 2013.
- Florida Department of Environmental Protection. 2024. <u>Florida Watershed Monitoring Status and</u> <u>Trend Program design document</u>. Tallahassee, FL: Division of Environmental Assessment and Restoration, Watershed Monitoring Program.

Appendix A – Example Streams Code

File: 2022-2024 SS Script.R # Purpose: Analysis of Florida Stream data generated by the Status # Monitoring Program. Created by Jay Silvanima using code developed # by Tony Olsen, Jay Silvanima, Chris Sedlacek, Stephanie Sunderman-Barnes, # and Liz Miller # Code developed using R version 4.4.1 (20204-06-14), spsurvey version 5.5.1, and FDEPgetdata version 2.02. # Check working directory location. getwd() # Load libraries for the data analyses library(FDEPgetdata) library(spsurvey) library(sqldf) # Run function of FDEPgetdata package which pulls exclusion data. # Insert variable name between parentheses in function call below. # Insert the string of characters which apply to the names for all stream projects. # For instance for 2018-2020 small stream projects enter "'SS18','SS19','SS20'". # This will pull the exclusion data for projects Z1SS1805, Z2SS1805, Z3SS1805, # Z4SS1805, Z5SS1805, Z6SS1805, Z1SS1905, Z2S1905, Z3SS1905, Z4SS1905, Z5SS1905, # Z6SS1905, Z1SS2005, Z2SS2005, Z3SS2005, Z4S2005, Z5SS2005, and Z6SS2005 # from the FDEP Oracle GWIS database table SITE EVALUATIONs. # Be sure to enclose character string in double and then single quotes. # Example "'SS18','SS19','SS20'" FDEPgetdata::getdata fw exclusions("'SS22','SS23','SS24'") # Function getdata fw exclusions creates a dataframe named 'Exclusions' from the # information provided. Total nitrogen (F.A.C. 62-302.531), total phosphorus (F.A.C. 62-302.531), # and dissolved oxygen (F.A.C. 62-302.533) criteria are added for each record # based on the corresponding nutrient watershed region and bioregion. # Determine if any of the sites fall on stream segments which are no # longer included in the target population. Function getdata fw site removals creates a sf object data # frame containing the sites that should be removed from this analysis because they no # longer are in the target population. This function will produce sf object format data frames with # sites to be retained for analysis (SITES WITHIN 50M) and sites to removed (SITES NOT WITHIN 50M). # Save a copy of SITES NOT WITHIN 50M data with the sf object formatting removed. getdata fw site removals (site evaluations = Exclusions, listframe directory = "./listframe shapefiles 2022-2024 streams", listframe = "Cycle18 streams coverage 2024") SS sitesNOTwithin50meters <- st drop geometry(SS SITES NOT WITHIN 50M) View(SS sitesNOTwithin50meters)

```
# Create a new data frame, named SiteEvaluations, by taking all sites from the Exclusions data frame that do
# not match sites in the site removals data frame.
SiteEvaluations <- sqldf('select * from Exclusions
                          where PK RANDOM SAMPLE LOCATION
                          not in (select PK RANDOM SAMPLE LOCATION
                                  SS sitesNOTwithin50meters)')
View(SiteEvaluations)
# Create new data frame from the one just created.
SS.SITES<-SiteEvaluations
names(SS.SITES)
# Convert to Decimal degrees
deg <- floor(SS.SITES$RANDOM LATITUDE/10000)</pre>
min <- floor((SS.SITES$RANDOM LATITUDE - deg*10000)/100)</pre>
sec <- SS.SITES$RANDOM LATITUDE - deg*10000 - min*100</pre>
SS.SITES$latdd <- deg + min/60 + sec/3600
deg <- floor(SS.SITES$RANDOM LONGITUDE/10000)</pre>
min <- floor((SS.SITES$RANDOM LONGITUDE - deg*10000)/100)</pre>
sec <- SS.SITES$RANDOM LONGITUDE - deg*10000 - min*100</pre>
SS.SITES$londd <- deg + min/60 + sec/3600
# Change londd to negative for correct use in sf.
SS.SITES$londd <- -SS.SITES$londd</pre>
# Create sf object and transform to Albers projection for analysis. This code utilizes Coordinate
# Reference System (CRS/EPSG) Codes. The first crs code (4269) below is for NAD 83 coordinate system
# the second crs code (3087) is for Florida albers projection. More information on these codes is found
# here: https://www.nceas.ucsb.edu/sites/default/files/2020-04/OverviewCoordinateReferenceSystems.pdf.
dsqn SS <- st as sf(SS.SITES, coords = c("londd", "latdd"), remove = FALSE, crs = 4269)
dsgn sf <- st transform(dsgn SS, crs = 3087)
# keep xy coords as variables
tmp <- st coordinates(dsgn sf)</pre>
dsgn sf$xcoord <- tmp[, "X"]</pre>
dsqn sf$ycoord <- tmp[, "Y"]</pre>
# Inspect the site location data by plotting them on a map.
# Create simple features objects from shapefiles of polygon features representing the Zones
# (Watershed Monitoring Section (WMS) Cycle 3 Reporting Units).
# Change projection for Zones sf object to Florida Albers HARN(CRS code 3087).
wms c3 reporting units <-
st read(dsn=".",layer="Watershed Monitoring Section (WMS) Cycle 3 Reporting Units")
wms c3 reporting units <- st transform(wms c3 reporting units, crs = 3087)
wms c3 reporting units
```

```
# Use sf to plot the Zone polygons and sites that were evaluated.
jpeg('2022 2024 SS Evaluated Sites.jpg', units = 'in', width = 7, height = 7, res = 300)
    plot(st geometry(wms c3 reporting units), main= '2022-2024 Steams Evaluated Sites')
    plot(st geometry(dsgn sf), pch = 21, bg = 'red', add = TRUE)
    legend(120000, 400000, legend='Zones', col='black',lty=1)
    legend(120000, 300000, legend='Evaluated Sites', col='red',pch=16)
dev.off()
# Site Evaluation
# The variables CAN BE SAMPLED, EXCLUSION CATEGORY and EXCLUSION CRITERIA provide information on the
# site evaluation results for each site. Review the information and create target/nontarget (TNT)
# variable.
addmargins(table(dsqn sf$EXCLUSION CATEGORY, dsqn sf$CAN BE SAMPLED, useNA = 'ifany'))
addmargins(table(dsgn sf$EXCLUSION CRITERIA, useNA = 'ifany'))
# create sampled and target (T) / nontarget (NT) variables
dsgn sf$EXCLUSION CATEGORY <- as.character(dsgn sf$EXCLUSION CATEGORY)
dsqn sf$EXCLUSION CATEGORY[dsqn sf$CAN BE SAMPLED == 'Y'] <- 'SAMPLED'
dsqn sf$EXCLUSION CATEGORY <- as.factor(dsqn sf$EXCLUSION CATEGORY)
     levels(dsgn sf$EXCLUSION CATEGORY)
dsqn sf$TNT <- dsqn sf$EXCLUSION CATEGORY
     levels(dsqn sf$TNT) <- list(T=c('SAMPLED', 'NO PERMISSION FROM OWNER', 'UNABLE TO
     ACCESS', 'OTHERWISE UNSAMPLEABLE', 'DRY'), NT=c ('WRONG RESOURCE/NOT PART OF TARGET POPULATION') )
addmargins(table(dsgn sf$EXCLUSION CATEGORY, dsgn sf$TNT, useNA = 'ifany'))
# Adjust weights for design as implemented
# Note need frame size here found in design doc for stream site selections
# Copy "2024 Stream Framesize.csv" from
# Z:\Status Sample Survey Designs\2024\Small Streams.
# Save CSV file to R project workspace, then load framesize data into R project.
framesize.df <- read.csv('2024 Stream Framesize.csv')</pre>
# Reduce framesize to remove total row. Change Zone name values to all capital letters.
framesize.df <- framesize.df[framesize.df$Zones !='Sum',]</pre>
framesize.df$Zones <- toupper(framesize.df$Zones)</pre>
# Convert framesize data frame to named vector
framesize <- as.vector(framesize.df$length km)</pre>
names(framesize) <- framesize.df[,'Zones']</pre>
# View framesize and paste results in comments here.
print(framesize)
# ZONE 1 ZONE 2 ZONE 3 ZONE 4 ZONE 5 ZONE 6
# 12320.7 2168.9 4456.9 4130.4 883.9 176.4
sum(framesize)
# Framesize in kilometers = 24137.2 for entire data frame.
```

```
# use all evaluated sites to adjust weights
nr <- nrow(dsgn sf)</pre>
dsqn sf$wqt <- adjwqt(rep(TRUE,nr), wqt=dsqn sf$NEST1 WT,</pre>
                      wgtcat=dsgn sf$REPORTING UNIT, framesize=framesize)
# check sum of weights for each reporting unit/basin
addmargins(tapply(dsgn sf$wgt, dsgn sf$REPORTING UNIT, sum))
# This gives the weights for the stream design as implemented in 2024.
# It must include all evaluated sites as some sites are not in the target population.
# Estimate Extent Stream Area.
# Since the sample frame includes portions of stream object line segments that do not meet the
# definition of a stream, the site evaluation information is used to estimate the stream kilometers in
# the target population for entire state and for each of the reporting units/basins.
dsgn sf$Combined = rep("All Basins", nr)
dsgn <- data.frame(PK RANDOM SAMPLE LOCATION = dsgn sf$PK RANDOM SAMPLE LOCATION,
                   wqt = dsqn sf$wqt,
                   xcoord = dsqn sf$xcoord,
                   ycoord = dsqn sf$ycoord,
                   Basin = dsgn sf$REPORTING UNIT,
                   Combined = dsgn sf$Combined,
                   TNTStatus = dsgn sf$TNT,
                   EXCLUSION.CATEGORY = dsqn sf$EXCLUSION CATEGORY)
ExtentEst <- cat analysis(dsgn,</pre>
                          vars = c('TNTStatus', 'EXCLUSION.CATEGORY'),
                          subpops = c('Combined', 'Basin'),
                          siteID = 'PK RANDOM SAMPLE LOCATION',
                          weight = 'wgt',
                          xcoord = 'xcoord',
                          ycoord = 'ycoord',
                          stratumID = 'Basin',
                          vartype = "Local",
                          conf=95)
# Add data columns to ExtentEst results. These will be used when loading analysis results to GWIS tables
# for Status Network analysis results.
analysis date <- as.character(Sys.Date())</pre>
ExtentEst <- cbind(ExtentEst,</pre>
                   SAMPLE YEAR = '2022-2024',
                   REPORTING CYCLE = '16-18',
                   WATER RESOURCE = 'SS',
                   ANALYSIS DATE = analysis date,
                   MATRIX = 'WATER',
                   ESTIMATEU UNITS = 'KILOMETERS')
# Export results
write.csv(ExtentEst,file = '2022 2024 SS ExtentEst.csv', row.names = FALSE)
```

```
# To estimate the percent of the target population
# that could be sampled, requires that the analysis be restricted to just sites
# in the target population, i.e., TNT = "T"
dsgn <- subset(dsgn, dsgn$TNT == "T")</pre>
```

ExtentEst_Target <- cat_analysis(dsgn,</pre>

```
vars = c('TNTStatus','EXCLUSION.CATEGORY'),
subpops = c('Combined','Basin'),
siteID = 'PK_RANDOM_SAMPLE_LOCATION',
weight = 'wgt',
xcoord = 'xcoord',
ycoord = 'ycoord',
stratumID = 'Basin',
vartype = "Local",
conf=95)
```

Export results
write.csv(ExtentEst_Target, file = '2022_2024_SS_ExtentEst_Target.csv')

Water Quality Data Analysis

Run function of FDEPgetdata package to pull result data. # Insert variable name between parentheses in function call below. # The function will pull the water resource for the water resource by year. For example canal projects # during year 2018 the entry would be "'CN18'" Entering "'SS18','SS19','SS20'" for the variable will # produce a data frame for FDEP Status Streams sampled 2018 - 2020. # Be sure to enclose in double and single quotes. FDEPgetdata::getdata results("'SS22','SS23','SS24'") # Function getdata results creates the table 'Results'. # Examine the results data frame. If more than two columns are present for each parameter, the data set # includes samples with multiple results for at least one parameter. Need to locate the affected # samples and investigate further. Type c(" in the R Studio search bar to search the results data frame # for the affected samples. # Warnings are generated regarding using type.convert.default. Examine the data to ensure # that result values are data type number. Redefine data types if needed, and # proceed if data types are correct. # Create new data frame from the one just created. SS RSLTS<-Results names(SS RSLTS) # Determine sample types and matrices present in results data. addmargins(table(SS_RSLTS\$SAMPLE_TYPE, SS_RSLTS\$MATRIX, useNA = 'ifany'))

Check that only PRIMARY sample type and WATER matrix data are present addmargins(table(SS WQ\$SAMPLE TYPE, SS WQ\$MATRIX, useNA = 'ifany'))


```
# Total Ammonia Nitrogen (TAN)
# Calculator for total ammonia nitrogen (TAN) single sample criteria.
# Created using TAN calculator spreadsheet as a guide
# (accessed 1/3/2020, https://floridadep.gov/dear/water-guality-standards-program/documents/total-
# ammonia-nitrogen-calculator%C2%A0).
# TAN pH = pH used in TAN threshold calc.
# If measured pH < 6.5, value used is 6.5.
# If meas. pH > 9.0, value used is 9.0.
SS WQ$TAN pH <- ifelse(SS WQ$pH Field < 6.5, 6.5,
               ifelse(SS WQ$pH Field > 9.0, 9.0, SS WQ$pH Field))
# TAN temp = water temperature used in TAN threshold calc.
# If measured temp < 7 degrees C, value used is 7.
SS WQ$TAN temp <- ifelse(SS WQ$Water Temperature < 7, 7, SS WQ$Water Temperature )
# calculate single sample TAN criteria using TAN pH and TAN temp
SS WQ$TAN Crit SingleSamp <- ifelse(is.na(SS WQ$TAN pH), NA,
                        ifelse(is.na(SS WQ$TAN temp), NA,
                        (2.5*(0.8876*((0.0278/(1+10^(7.688-SS WQ$TAN pH)))+(1.1994/(1+10^(SS WQ$TAN pH-
                        7.688))))*2.126*10^(0.028*(20-(SS WQ$TAN temp)))))))
# Round result to two decimal places, for consistency with thresholds generated using TAN calculator
# spreadsheet
```

```
SS_WQ$TAN_Crit_SingleSamp <- round(SS_WQ$TAN_Crit_SingleSamp, digits=2)</pre>
```

```
# Compare TAN thresholds to TAN result value for each sample
# Pass=1 AND Fail=0
SS WQ$TAN Cat<-ifelse((SS WQ$TAN Crit SingleSamp >= SS WQ$Ammonia Total as N),1,0)
###### Set up threshold category columns for other analytes
# E Coli category
SS WQ$E Coli Category <- as.factor(cut(SS WQ$Escherichia Coli Quanti Tray, breaks=c(0,410,10000000),
                                   include.lowest=TRUE))
# Old Dissolved Oxygen Category
SS WQ$DO cat old <- as.factor(cut(SS WQ$Oxygen Dissolved Field, breaks=c(0,4.999,1000000),
                                   include.lowest=TRUE))
# pH Category
SS WQ$pH Category <- as.factor(cut(SS WQ$pH Field, breaks=c(0,5.999,8.5,14), include.lowest=TRUE))
# Chlorophyll Category
SS WQ$Chlorophyll Category <- as.factor(cut(SS WQ$Chlorophyll A Monochromatic, breaks=c(0,20,10000),
                                   include.lowest=TRUE))
# Fluoride class III water quality standard exceedances.
SS WQ$Fluoride Category <- as.factor(cut(SS WQ$Fluoride Total, breaks=c(0,10,100000),
                                   include.lowest=TRUE))
##### Numeric Nutrient and DO Categories
# Note Pass=1 AND Fail=0
# Total NNC category = sum of category results for TN, TP, and DO.
# If Tot cat = 3, sample passed criteria for these 3 parameters.
# Calculate Total Nitrogen (TN) as sum of TKN & NO3NO2.
SS WQ$TN<-(SS WQ$Kjeldahl Nitrogen Total as N+SS WQ$NitrateNitrite Total as N)
# TN NNC, TP NNC, and DO Conc are thresholds included in site evaluations data, based on nutrient
# watershed regions and bioregions.
SS WQ$TN cat<-ifelse((SS WQ$TN NNC >= SS WQ$TN),1,0)
SS WQ$TP cat<-ifelse((SS WQ$TP NNC >= SS WQ$Phosphorus Total as P),1,0)
SS WQ$DO cat<-ifelse((SS WQ$Oxygen Dissolved Percent Saturation >= SS WQ$DO Conc ),1,0)
SS WQ$Tot cat<-(SS WQ$TN cat+SS WQ$TP cat+SS WQ$DO cat)
#### Categories for metals exceeding class III water quality standards.
# Antimony Category
SS WQ$Antimony Category <- as.factor(cut(SS WQ$Antimony Total, breaks=c(0,4300,100000),
include.lowest=TRUE))
# Arsenic Category
SS WQ$Arsenic Category <- as.factor(cut(SS WQ$Arsenic Total, breaks=c(0,50,100000),
include.lowest=TRUE))
```

```
# Beryllium Category
SS WQ$Beryllium Category <- as.factor(cut(SS WQ$Beryllium Total, breaks=c(0,0.13,100000),
include.lowest=TRUE))
# Iron Category
SS WQ$Iron Category <- as.factor(cut(SS WQ$Iron Total, breaks=c(0,1000,100000), include.lowest=TRUE))
# Selenium Category
SS WQ$Selenium Category <- as.factor(cut(SS WQ$Selenium Total, breaks=c(0,5,100000),
include.lowest=TRUE))
# Silver Category
SS WQ$Silver Category <- as.factor(cut(SS WQ$Silver Total, breaks=c(0,0.07,100000),
include.lowest=TRUE))
# Thallium Category
SS WQ$Thallium Category <- as.factor(cut(SS WQ$Thallium Total, breaks=c(0,6.299,100000),
include.lowest=TRUE))
### Metals thresholds based on hardness (cadmium, chromium, copper, lead, nickel, zinc) are calculating
   using the calculator spreadsheet as a guide (accessed 10/13/2023,
# https://floridadep.gov/sites/default/files/MetalsCriteriaCalculator.xlsm)
## Metals Hardness = Harness used in metals threshold calculation.
  If measured hardness calculated as CACO3 < 25, value used is 25.
   If measured hardness calculated as CACO3 > 400, value used is 400.
                         SS WQ$Metals Hardness <- ifelse(SS WQ$Hardness calculated as CACO3 < 25, 25,
                         ifelse(SS WQ$Hardness calculated as CACO3 > 400, 400,
                         SS WQ$Hardness calculated as CACO3))
## Calculate single sample Cadmium Total Criteria using Metals Hardness
SS WQ$TotCadmiumCrit SingleSamp <- ifelse(is.na(SS WQ$Metals Hardness), NA,
      (exp((0.7409*(log(SS WQ$Metals Hardness)))-4.719)))
# Round result to four decimal places
SS WQ$TotCadmiumCrit SingleSamp <- round(SS WQ$TotCadmiumCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
SS WQ$Cadmium Category<-ifelse((SS WQ$TotCadmiumCrit SingleSamp >= SS WQ$Cadmium Total),1,0)
## Calculate single sample Chromium Total Criteria using Metals Hardness
SS WQ$TotChromiumCrit SingleSamp <- ifelse(is.na(SS WQ$Metals Hardness), NA,
      (exp((0.819*(log(SS WQ$Metals Hardness)))+0.6848)))
# Round result to four decimal places
SS WQ$TotChromiumCrit SingleSamp <- round(SS WQ$TotChromiumCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
SS WQ$Chromium Category<-ifelse((SS WQ$TotChromiumCrit SingleSamp >= SS WQ$Chromium Total),1,0)
```

```
## Calculate single sample Copper Total Criteria using Metals Hardness
SS WQ$TotCopperCrit SingleSamp <- ifelse(is.na(SS WQ$Metals Hardness), NA,
      (exp((0.8545*(log(SS WQ$Metals Hardness)))-1.702)))
# Round result to four decimal places
SS WQ$TotCopperCrit SingleSamp <- round(SS WQ$TotCopperCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
SS WQ$Copper Category<-ifelse((SS WQ$TotCopperCrit SingleSamp >= SS WQ$Copper Total),1,0)
## Calculate single sample Lead Total Criteria using Metals Hardness
SS WQ$TotLeadCrit SingleSamp <- ifelse(is.na(SS WQ$Metals Hardness), NA,
      (exp((1.273*(log(SS WQ$Metals Hardness)))-4.705)))
# Round result to four decimal places
SS WQ$TotLeadCrit SingleSamp <- round(SS WQ$TotLeadCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
SS WQ$Lead Category<-ifelse((SS WQ$TotLeadCrit SingleSamp >= SS WQ$Lead Total),1,0)
## Calculate single sample Nickel Total Criteria using Metals Hardness
SS WQ$TotNickelCrit SingleSamp <- ifelse(is.na(SS WQ$Metals Hardness), NA,
      (exp((0.846*(log(SS WQ$Metals Hardness)))+0.0584)))
# Round result to four decimal places
SS WQ$TotNickelCrit SingleSamp <- round(SS WQ$TotNickelCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
SS WQ$Nickel Category<-ifelse((SS WQ$TotNickelCrit SingleSamp >= SS WQ$Nickel Total),1,0)
## Calculate single sample Zinc Total Criteria using Metals Hardness
SS WQ$TotZincCrit SingleSamp <- ifelse(is.na(SS WQ$Metals Hardness), NA,
      (exp((0.8473*(log(SS WQ$Metals Hardness)))+0.884)))
# Round result to four decimal places
SS WQ$TotZincCrit SingleSamp <- round(SS WQ$TotZincCrit SingleSamp, digits=4)
# Assign values for Pass (1) and Fail (0).
SS WQ$Zinc Category<-ifelse((SS WQ$TotZincCrit SingleSamp >= SS WQ$Zinc Total),1,0)
## Export dataframe SS WQ to .csv file format
write.csv(SS WQ, "SS WQ.csv", row.names = FALSE)
*****
# Continuous and categorical water quality indicator population estimation
#### continuous WO estimates
# SS WQ is the merged dsgn sf and SS RESULTS data.
# Add Combined category with name "All Basins" and added Basin category with reporting unit data.
nr <- nrow(SS WO)</pre>
levels(SS WQ$TNT)
```

data_cont_WQ <- data.frame(SS_WQ,</pre>

Combined = rep("All Basins", nr), Basin = SS_WQ\$REPORTING_UNIT) # List all variables for continuous analysis here.

```
ContVars <- c('Water_Temperature','pH_Field','Oxygen_Dissolved_Field',
    'Oxygen_Dissolved_Percent_Saturation',
    'Specific_Conductance_Field','TN','Phosphorus_Total_as_P','Ammonia_Total_as_N',
    'Escherichia_Coli_Quanti_Tray','NitrateNitrite_Total_as_N','Kjeldahl_Nitrogen_Total_as_N',
    'Chlorophyll_A_Monochromatic','Alkalinity_Total_as_CaCO3','Total_Suspended_Solids_TSS',
    'Organic_Carbon_Total','Turbidity_Lab','Chloride_Total','Sodium_Total','Aluminum_Total',
    'Antimony_Total','Arsenic_Total','Barium_Total','Beryllium_Total','Cadmium_Total',
    'Chromium_Total','Copper_Total','Iron_Total','Lead_Total','Manganese_Total',
    'Molybdenum_total','Nickel_Total','Selenium_Total','Silver_Total','Thallium_Total',
    'Zinc_Total', 'Fluoride_Total', 'Anatoxin_a','Cylindrospermopsin',
    'Desmethyl_microcystin_LR', 'Microcystin_HilR','Microcystin_LY',
    'Microcystin_LF', 'Microcystin_UR','Microcystin_LY',
    'Microcystin_RR','Microcystin_WR','Microcystin_YR','Nodularin R')
```

Run Continuous analysis for all analytes in ContVars.

```
Water_quality_Cont <- cont_analysis(data_cont_WQ,</pre>
```

```
vars = ContVars,
subpops = c('Combined','Basin'),
siteID = 'PK_RANDOM_SAMPLE_LOCATION',
weight = 'wgt',
xcoord = 'xcoord',
ycoord = 'ycoord',
stratumID = 'Basin',
vartype = 'Local',
statistics = c('CDF','Pct','Mean','Total'),
conf=95,
popsize = list(Basin = framesize))
```

Add data columns to all Water_quality_Cont results data frames. These will be # used when loading analysis results to GWIS tables for Status Network analysis results. analysis_date <- as.character(Sys.Date())</pre>

```
Water_quality_Cont <- lapply(Water_quality_Cont,</pre>
```

```
function(df)
cbind(df,
    SAMPLE_YEAR = '2022-2024',
    REPORTING_CYCLE = '16-18',
    WATER_RESOURCE = 'STREAM',
    ANALYSIS_DATE = analysis_date,
    MATRIX = 'WATER'))
```

Merge Pct and Mean Results into single data frame, for consistency with format of previous years'
results.
Water_quality_Cont[["Mean"]] <- cbind(Water_quality_Cont[["Mean"]], Statistic="Mean")
Water_quality_Cont[["Pct"]] <- rbind(Water_quality_Cont[["Pct"]],Water_quality_Cont[["Mean"]])</pre>

categorical WQ estimates

Data frame SS_WQ is the merged dsgn_sf and SS_RESULTS data.

Add Combined category with name "All Basins" and add Basin category

```
# with reporting unit data.
data cat WQ <- data.frame(SS WQ,</pre>
                          Combined = rep("All Basins", nr),
                          Basin = SS WQ$REPORTING UNIT)
# List all variables for categorical analysis here.
CatVars <- subset(names(SS WQ),
                  grepl(" Category$", names(SS WQ)))
# Run categorical analysis for all analytes in CatVars.
Water Quality Cat <- cat analysis(data cat WQ,
                                  vars = CatVars,
                                  subpops = c('Combined', 'Basin'),
                                  siteID = 'PK RANDOM SAMPLE LOCATION',
                                  weight = 'wqt',
                                  xcoord = 'xcoord',
                                  ycoord = 'ycoord',
                                  stratumID = 'Basin',
                                  vartype = 'Local',
                                  conf=95,
                                  popsize = list(Basin = framesize))
# Add data columns to Water Quality Cat results. These will be
# used when loading analysis results to GWIS tables for Status Network analysis results.
analysis date <- as.character(Sys.Date())</pre>
Water Quality Cat <- cbind (Water Quality Cat,
                            SAMPLE YEAR = '2022-2024',
                            REPORTING CYCLE = '16-18',
                           WATER RESOURCE = 'SS',
                           ANALYSIS DATE = analysis date,
                           MATRIX = 'WATER',
                            ANALYSIS TYPE = 'TARGET POPULATION',
                            ESTIMATEU UNITS = 'KILOMETERS')
# Export the results
write.csv(Water Quality Cat, "2022 2024 SS Cat.csv", row.names = FALSE)
write.csv(Water quality Cont$CDF, file = '2022 2024 SS Cont EstCDF.csv', row.names = FALSE)
write.csv(Water quality Cont$Pct, file = '2022 2024 SS Cont EstPCT.csv', row.names = FALSE)
write.csv(Water quality Cont$Total, file = '2022 2024 SS Cont EstTotal.csv', row.names = FALSE)
```

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.				
В	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.				
Ι	The reported value is greater than or equal to the laboratory method detection				
	limit but less than the laboratory practical quantification limit.				
J	<i>Estimate value. Shall be accompanied by a detailed explanation to justify the</i>				
	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	<i>Off-scale low. The actual value is known to be less than the value given. (Only</i>				
_	used for lab analyses.)				
L	<i>Off-scale high. The actual value is known to be greater than the value given.</i>				
	(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.				
0	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 $\frac{1}{2}$ 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 (anglest sample volume) used for the tot and upported with the				
	aualifier 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 – Quality Assurance Checklist

This checklist is used to independently review the results of Status Network analyses. The example checklist below is for the 2022-2024 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.

2022-2024 Status Network Analysis– QA Checks – (List Water Resource) – (List Date Checklist Last Updated)

Exclusion / Site Evaluation Data

- Check that weights seem reasonable (same order of magnitude, etc.).
- Check that lat / long are in DDMMSS.THM format.
- Check that all "Can be sampled = N" have exclusion category & criteria listed.
- Check that all "Can be sampled = Y" do not have exclusion category or criteria.
- Check that there are no sites missing (e.g. "Can be sampled = NA") up through highest selection sampled in each zone.
- Check that site evaluation data from individual years were completely transferred to combined site evaluation file.

Resource	# Sites	# Sites	# Sites	Sum of # Sites from	# Sites 2022-2024 in
	2022	2023	2024	3 Individual Years	Combined File

- Check GIS analysis and confirm that sites were removed from analysis if they were located on portions of features not included in coverage used for most recent year's site selections. Manually inspect sites that were removed to understand reason for permanent removal from coverage used for site selections. (Applies to multi-year analysis only.)
- Check that TN, TP, & DO thresholds in site evaluations data match thresholds in WMS Design Document. (Applies to surface water resources only.)

Result Data

• Check that data was provided for all sampled sites.

Year	Total Sites Sampled from Combined Site Evaluations Data	Total Sites in Combined Result Data
2022		
2023		
2024		

• Check that result data from individual years were completely transferred to combined result file.

Resource	# Sampled Sites 2022	# Sampled Sites 2023	# Sampled Sites 2024	Sum of # Sampled Sites from 3 Individual Years	# Sampled Sites in 2022-2024 Combined File

• Check that merged site evaluation and result data has correct number of samples (# sites from result data minus samples marked as inappropriate for analysis (random_site_location ends in "B") minus samples from sites removed in GIS analysis step).

R Script and Analysis Results

- Check that framesize information was correctly transferred from site selection documentation to analysis R script.
- Check that weights used in analysis seem reasonable (same order of magnitude, etc.)
- Check that fatal qualifiers (O, ?, N, X, T) and non-detects with MDL above water quality threshold are handled correctly in analysis.
- Check that thresholds in R script & output files match thresholds in WMS Design Document.
- Check that list of parameters in R script & output files for continuous analysis matches reference list (\\floridadep\data\dear\wqap\sol_z\data analysis\Data Protocols\Parameters for Status Network Continuous Analysis.docx).
- Check that list of parameters in R script & output files for categorical analysis matches reference list (\\floridadep\data\dear\wqap\sol_z\data analysis\Data Protocols\Parameters for Status Network Categorical Analysis.docx).
- Check that number of values (N) used in analysis matches N in merged data file.
- Check that population size for categorical and continuous analysis results matches total extent of resource (framesize total).
- Check that total N in ExtentExt.csv match N from site evaluation data.

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

• Check that analysis is reproducible. Confirm that R script can be rerun without encountering errors. Inspect any warning encountered and ensure that warnings do not affect analysis results.

Misc. Notes

Recommend performing script edits starting at bottom of list and working towards top, to preserve references to line numbers.