SEACAR Discrete Water Quality Analysis: Sample Surface Total Phosphorus

Last compiled on 19 May, 2022

Contents

Libraries

Loads libraries used in the script. The inclusion of `scipen` option limits how frequently R defaults to scientific notation.

```
library(knitr)
library(data.table)
library(dplyr)
library(lubridate)
library(ggplot2)
```

```
library(ggpubr)
library(scales)
library(EnvStats)
library(tidyr)
options(scipen = 999)
```

File Import

Imports file that is determined in the WC_Discrete_parameter_ReportCompile.R script.

The command `fread` is used because of its improved speed while handling large data files. Only columns that are used by the script are imported from the file, and are designated in the `select` input.

The script then gets the name of the parameter as it appears in the data file, units of the parameter, sets the `SampleDate` as a date object, and creates various scales of the date to be used by plotting functions.

```
data <- fread(file_in, sep = "|", header = TRUE, stringsAsFactors = FALSE,
              select = c("ManagedAreaName", "ProgramID", "ProgramName",
                         "ProgramLocationID", "SampleDate", "Year", "Month",
                         "RelativeDepth", "ActivityType", "ParameterName",
                         "ResultValue", "ParameterUnits", "ValueQualifier",
                         "SEACAR_QAQCFlagCode", "Include"), na.strings = "")

data <- merge.data.frame(MA_All[,c("AreaID", "ManagedAreaName")],
                          data, by = "ManagedAreaName")

parameter <- unique(data$ParameterName)
unit <- unique(data$ParameterUnits)
data$SampleDate <- as.Date(data$SampleDate)
data$YearMonth <- paste0(data$Month, "-", data$Year)
data$YearMonthDec <- data$Year + ((data$Month-0.5) / 12)
data$DecDate <- decimal_date(data$SampleDate)
```

Data Summary Before Filtering

This part is to create a data summary file that looks at the total amount of data, how many pass the initial database filtering, and how many are impacted by various ValueQualifiers. The ValueQualifiers of interest are H, I, Q, S, and U.

```
data_summ <- data %>%
   group_by(AreaID, ManagedAreaName) %>%
   summarize(ParameterName = parameter,
             RelativeDepth = depth,
             ActivityType = activity,
             N_Total = length(ResultValue),
             N_Include = length(ResultValue[Include==1]),
             N_H = length(grep("H", data$ValueQualifier[data$ProgramID==476])),
             perc_H = 100*N_H/length(data$ValueQualifier),
             N_I = length(grep("I", data$ValueQualifier)),
             perc_I = 100*N_I/length(data$ValueQualifier),
             N_Q = length(grep("Q", data$ValueQualifier)),
```

```
              perc_Q = 100*N_Q/length(data$ValueQualifier),
              N_S = length(grep("S", data$ValueQualifier)),
              perc_S = 100*N_S/length(data$ValueQualifier),
              N_U = length(grep("U", data$ValueQualifier)),
              perc_U = 100*N_U/length(data$ValueQualifier))
```

```
## `summarise()` has grouped output by 'AreaID'. You can override using the `.groups` argument.
```

```
data_summ <- as.data.table(data_summ[order(data_summ$ManagedAreaName), ])
fwrite(data_summ, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                          "_DataSummary_BeforeFilter.csv"), sep = ",")
```

Data Filtering and Data Impacted by Specific Value Qualifiers

Most data filtering is performed on export from the database, and is indicated by the `Include` variable. `Include` values of 1 indicate the data should be used for analysis, values of 0 indicate the data should not be used for analysis. Documentation on the database filtering is provided here: SEACAR Documentation-Analysis Filters and Calculations.docx

The filtering that is performed by the script at this point removes rows that are missing values for `ResultValue`, and only keeps data that is measured at the relative depth (surface, bottom, etc.) and activity type (field or sample) of interest. This is partly handled on export with the `RelativeDepth` variable, but there are some measurements that are considered both surface and bottom based on measurement depth and total depth. By default, these are marked as `Surface` for `RelativeDepth` and receive a `SEACAR_QAQCFlag` indicator of `12Q`. Data passes the filtering the process if it is from the correct depth and has an `Include` value of 1. The script also only looks at data of the desired `ActivityType` which indicates whether it was measured in the field (`Field`) or in the lab (`Sample`).

After the initial filtering, a second filter variable is created to determine whether enough time is represented in the managed area, which is that each managed area has 10 year or more of unique year entries for observation that pass the initial filter. If data passes the first set of filtering criteria and the time criteria, they are used in the analysis.

After filtering, the amount of data impacted by the H (for dissolved oxygen & pH in program 476), I, Q, S (for Secchi depth), and U value qualifiers. A variable is also created that determines if scatter plot points should be a different color based on value qualifiers of interest.

```
if(depth=="Bottom"){
   data$RelativeDepth[grep("12Q", data$SEACAR_QAQCFlagCode[
      data$RelativeDepth == "Surface"])] <- "Bottom"
}

data$Include <- as.logical(data$Include)
data <- data[!is.na(data$ResultValue),]
if(param_name!="Secchi_Depth"){
   data <- data[!is.na(data$RelativeDepth),]
data <- data[data$RelativeDepth==depth,]
}

if(length(grep("Blank", data$ActivityType)) >0){
      data <- data[-grep("Blank", data$ActivityType),]
}
```

```r
if(param_name == "Chlorophyll_a_uncorrected_for_pheophytin" |
   param_name == "Salinity" | param_name == "Turbidity"){
   data <- data[grep(activity, data$ActivityType[!is.na(data$ActivityType)]),]
}

if(param_name == "Water_Temperature"){
   data <- data[data$ResultValue>=-2,]
} else{
   data <- data[data$ResultValue>=0,]
}
data$Include[grep("H", data$ValueQualifier[data$ProgramID==476])] <- TRUE

MA_Years <- data[data$Include == TRUE, ] %>%
   group_by(ManagedAreaName) %>%
   summarize(N = length(unique(Year)))
MA_Years <- as.data.table(MA_Years[order(MA_Years$ManagedAreaName), ])
MA_Years$Enough_Time <- ifelse(MA_Years$N < 10, FALSE, TRUE)
data <- merge.data.frame(data, MA_Years[,c("ManagedAreaName", "Enough_Time")],
                         by = "ManagedAreaName")

data$Use_In_Analysis <- ifelse(data$Include == TRUE &
                                  data$Enough_Time == TRUE,
                               TRUE, FALSE)

total <- length(data$Include)
pass_filter <- length(data$Include[data$Include==TRUE])

count_H <- length(grep("H", data$ValueQualifier[data$ProgramID==476]))
perc_H <- 100*count_H/length(data$ValueQualifier)

count_I <- length(grep("I", data$ValueQualifier))
perc_I <- 100*count_I/length(data$ValueQualifier)

count_Q <- length(grep("Q", data$ValueQualifier))
perc_Q <- 100*count_Q/length(data$ValueQualifier)

count_S <- length(grep("S", data$ValueQualifier))
perc_S <- 100*count_S/length(data$ValueQualifier)

count_U <- length(grep("U", data$ValueQualifier))
perc_U <- 100*count_U/length(data$ValueQualifier)


data$VQ_Plot <- data$ValueQualifier

inc_H <- ifelse(param_name=="pH" | param_name=="Dissolved_Oxygen" |
                   param_name=="Dissolved_Oxygen_Saturation", TRUE, FALSE)

if (inc_H==TRUE){
   data$VQ_Plot <- gsub("[^HU]+", "", data$VQ_Plot)
   data$VQ_Plot <- gsub("UH", "HU", data$VQ_Plot)
   data$VQ_Plot[data$ProgramID!=476] <- gsub("[^U]+", "",
                                             data$VQ_Plot[data$ProgramID!=476])
```

```r
  data$VQ_Plot[data$VQ_Plot==""] <- NA


  cat(paste0("Number of Measurements: ", total,
             ", Number Passed Filter: ", pass_filter, "\n",
             "Program 476 H Codes: ", count_H, " (", round(perc_H, 6), "%)\n",
             "I Codes: ", count_I, " (", round(perc_I, 6), "%)\n",
             "Q Codes: ", count_Q, " (", round(perc_Q, 6), "%)\n",
             "U Codes: ", count_U, " (", round(perc_U, 6), "%)"))
} else if (param_name=="Secchi_Depth") {
  count_S <- length(grep("S", data$ValueQualifier))
  perc_S <- 100*count_S/length(data$ValueQualifier)
  data$VQ_Plot <- gsub("[^SU]+", "", data$VQ_Plot)
  data$VQ_Plot <- gsub("US", "SU", data$VQ_Plot)
  data$VQ_Plot[data$VQ_Plot==""] <- NA
  cat(paste0("Number of Measurements: ", total,
             ", Number Passed Filter: ", pass_filter, "\n",
             "I Codes: ", count_I, " (", round(perc_I, 6), "%)\n",
             "Q Codes: ", count_Q, " (", round(perc_Q, 6), "%)\n",
             "S Codes: ", count_S, " (", round(perc_S, 6), "%)\n",
             "U Codes: ", count_U, " (", round(perc_U, 6), "%)"))
} else{
  data$VQ_Plot <- gsub("[^U]+", "", data$VQ_Plot)
  data$VQ_Plot[data$VQ_Plot==""] <- NA
  cat(paste0("Number of Measurements: ", total,
             ", Number Passed Filter: ", pass_filter, "\n",
             "I Codes: ", count_I, " (", round(perc_I, 6), "%)\n",
             "Q Codes: ", count_Q, " (", round(perc_Q, 6), "%)\n",
             "U Codes: ", count_U, " (", round(perc_U, 6), "%)"))
}
```

```
## Number of Measurements: 96289, Number Passed Filter: 95446
## I Codes: 11435 (11.875708%)
## Q Codes: 1227 (1.274289%)
## U Codes: 2881 (2.992034%)
```

```r
data_summ <- data %>%
  group_by(AreaID, ManagedAreaName) %>%
  summarize(ParameterName = parameter,
            RelativeDepth = depth,
            ActivityType = activity,
            N_Total = length(ResultValue),
            N_AnalysisUse = length(ResultValue[Use_In_Analysis==TRUE]),
            N_H = length(grep("H", data$ValueQualifier[data$ProgramID==476])),
            perc_H = 100*N_H/length(data$ValueQualifier),
            N_I = length(grep("I", data$ValueQualifier)),
            perc_I = 100*N_I/length(data$ValueQualifier),
            N_Q = length(grep("Q", data$ValueQualifier)),
            perc_Q = 100*N_Q/length(data$ValueQualifier),
            N_S = length(grep("S", data$ValueQualifier)),
            perc_S = 100*N_S/length(data$ValueQualifier),
            N_U = length(grep("U", data$ValueQualifier)),
            perc_U = 100*N_U/length(data$ValueQualifier))
```

```
## ‘summarise()‘ has grouped output by ’AreaID’. You can override using the ‘.groups‘ argument.
```

```
data_summ <- as.data.table(data_summ[order(data_summ$ManagedAreaName), ])
fwrite(data_summ, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                         "_DataSummary_AfterFilter.csv"), sep = ",")
```

Managed Area Statistics

Gets summary statistics for each managed area. Excluded managed areas are not included into whether the data should be used or not. Uses piping from dplyr package to feed into subsequent steps. The following steps are performed:

1. Take the `data` variable and only include rows that have a `Use_In_Analysis` value of TRUE
2. Group data that have the same `ManagedAreaName`, `Year`, and `Month`.
   - Second summary statistics do not use the `Month` grouping and are only for `ManagedAreaName` and `Year`.
   - Third summary statistics do not use `Year` grouping and are only for `ManagedAreaName` and `Month`
3. For each group, provide the following information: Number of Entries (N), Lowest Value (Min), Largest Value (Max), Median, Mean, Standard Deviation, and a list of all Program IDs included in these measurements.
4. Sort the data in ascending (A to Z and 0 to 9) order based on `ManagedAreaName` then `Year` then `Month`
5. Write summary stats to a pipe-delimited .txt file in the output directory
   - Click this text to open Git directory with output files

```
MA_names <- unique(data$ManagedAreaName[data$Use_In_Analysis == TRUE])
MA_names <- MA_names[order(MA_names)]
n <- length(MA_names)

MA_YM_Stats <- data[data$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, Year, Month) %>%
   summarize(ParameterName = parameter,
             RelativeDepth = depth,
             ActivityType = activity,
             N = length(ResultValue),
             Min = min(ResultValue),
             Max = max(ResultValue),
             Median = median(ResultValue),
             Mean = mean(ResultValue),
             StandardDeviation = sd(ResultValue),
             ProgramIDs = paste(sort(unique(ProgramID), decreasing = FALSE),
                                collapse = ', '))
MA_YM_Stats <- as.data.table(MA_YM_Stats[order(MA_YM_Stats$ManagedAreaName,
                                               MA_YM_Stats$Year,
                                               MA_YM_Stats$Month), ])
fwrite(MA_YM_Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                           "_ManagedArea_YearMonth_Stats.txt"), sep = "|")

MA_Y_Stats <- data[data$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, Year) %>%
   summarize(ParameterName = parameter,
```

```
            RelativeDepth = depth,
            ActivityType = activity,
            N = length(ResultValue),
            Min = min(ResultValue),
            Max = max(ResultValue),
            Median = median(ResultValue),
            Mean = mean(ResultValue),
            StandardDeviation = sd(ResultValue),
            ProgramIDs = paste(sort(unique(ProgramID), decreasing = FALSE),
                               collapse = ', '))
MA_Y_Stats <- as.data.table(MA_Y_Stats[order(MA_Y_Stats$ManagedAreaName,
                                             MA_Y_Stats$Year), ])
fwrite(MA_Y_Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                          "_ManagedArea_Year_Stats.txt"), sep = "|")

MA_M_Stats <- data[data$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, Month) %>%
   summarize(ParameterName = parameter,
            RelativeDepth = depth,
            ActivityType = activity,
            N = length(ResultValue),
            Min = min(ResultValue),
            Max = max(ResultValue),
            Median = median(ResultValue),
            Mean = mean(ResultValue),
            StandardDeviation = sd(ResultValue),
            ProgramIDs = paste(sort(unique(ProgramID), decreasing = FALSE),
                               collapse = ', '))
MA_M_Stats <- as.data.table(MA_M_Stats[order(MA_M_Stats$ManagedAreaName,
                                             MA_M_Stats$Month), ])
fwrite(MA_M_Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                          "_ManagedArea_Month_Stats.txt"), sep = "|")
```

Monitoring Location Statistics

Gets monitoring location statistics, which is defined as a unique combination of `ManagedAreaName`, `ProgramID`, `ProgramAreaName`, and `ProgramLocationID`, using piping from dplyr package. The following steps are performed:

1. Take the `data` variable and only include rows that have a `Use_In_Analysis` value of `TRUE`
2. Group data that have the same `ManagedAreaName`, `ProgramID`, `ProgramName`, and `ProgramLocationID`.
3. For each group, provide the following information: Earliest Sample Date (EarliestSampleDate), Latest Sample Date (LastSampleDate), Number of Entries (N), Lowest Value (Min), Largest Value (Max), Median, Mean, and Standard Deviation.
4. Sort the data in ascending (A to Z and 0 to 9) order based on `ManagedAreaName` then `ProgramName` then `ProgramID` then `ProgramLocationID`
5. Write summary stats to a pipe-delimited .txt file in the output directory

   - Click this text to open Git directory with output files

```
Mon_Stats <- data[data$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID) %>%
   summarize(ParameterName = parameter,
             RelativeDepth = depth,
             ActivityType = activity,
             EarliestSampleDate = min(SampleDate),
             LastSampleDate = max(SampleDate),
             N = length(ResultValue),
             Min = min(ResultValue),
             Max = max(ResultValue),
             Median = median(ResultValue),
             Mean = mean(ResultValue),
             StandardDeviation = sd(ResultValue))

Mon_Stats <- as.data.table(Mon_Stats[order(Mon_Stats$ManagedAreaName,
                                              Mon_Stats$ProgramName,
                                              Mon_Stats$ProgramID,
                                              Mon_Stats$ProgramLocationID), ])
fwrite(Mon_Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                          "_MonitoringLoc_Stats.txt"), sep = "|")
```

Seasonal Kendall Tau Analysis

Gets seasonal Kendall Tau statistics using the `kendallSeasonalTrendTest` from the `EnvStats` package. The `Trend` parameter is determined from a user-defined function based on the median, Senn slope, and p values from the data. Analysis modified from code created by Jason Scolaro that performed at The Water Atlas: https://sarasota.wateratlas.usf.edu/water-quality-trends/#analysis-overview

The following steps are performed:

1. Define the functions used in the analysis
2. Check to see if there are any groups to run analysis on.
3. Take the `data` variable and only include rows that have a `Use_In_Analysis` value of `TRUE`
4. Group data that have the same `ManagedAreaName`.
5. For each group, provides the following information: Earliest Sample Date (EarliestSampleDate), Latest Sample Date (LastSampleDate), Number of Entries (N), Lowest Value (Min), Largest Value (Max), Median, Mean, Standard Deviation, tau, Senn Slope (SennSlope), Senn Intercept (SennIntercept), and p.
   - The analysis is run with the `kendallSeasonalTrendTest` function using the `Year` values for year, and `Month` as the seasonal qualifier, and Trend.
   - An independent.obs value of `TRUE` indicates that the data should be treated as not being serially auto-correlated. An independent.obs value of `FALSE` indicates that it is treated as being serially auto-correlated, but also requires one observation per season per year for the full time of observation.
6. Reformat columns in the data frame from export.
7. Write summary stats to a pipe-delimited .txt file in the output directory
   - Click this text to open Git directory with output files

```
tauSeasonal <- function(data, independent, stats.median, stats.minYear,
                        stats.maxYear) {
```

```
  tau <- NULL
  tryCatch({
    ken <-
      kendallSeasonalTrendTest(
        y = data$ResultValue,
        season = data$Month,
        year = data$Year,
        independent.obs = independent
      )
    tau <- ken$estimate[1]
    p <- ken$p.value[2]
    slope <- ken$estimate[2]
    intercept <- ken$estimate[3]
    trend <- trend_calculator(slope, stats.median, p)
  }, warning = function(w) {
    print(w)
  }, error = function(e) {
    print(e)
  }, finally = {
    if (!exists("tau")) {
      tau <- NULL
    }
    if (!exists("p")) {
      p <- NULL
    }
    if (!exists("slope")) {
      slope <- NULL
    }
    if (!exists("intercept")) {
      intercept <- NULL
    }
    if (!exists("trend")) {
      trend <- NULL
    }
  })
  KT <-c(unique(data$AreaID),
         unique(data$ManagedAreaName),
         parameter,
         depth,
         activity,
         independent,
         stats.median,
         nrow(data),
         stats.minYear,
         stats.maxYear,
         tau,
         p,
         slope,
         intercept,
         trend)
  return(KT)
}
runStats <- function(data) {
```

```
    data$Index <- as.Date(data$SampleDate) # , "%Y-%m-%d")
    data$ResultValue <- as.numeric(data$ResultValue)
    # Calculate basic stats
    stats.median <- median(data$ResultValue, na.rm = TRUE)
    stats.minYear <- min(data$Year, na.rm = TRUE)
    stats.maxYear <- max(data$Year, na.rm = TRUE)
    # Calculate Kendall Tau and Slope stats, then update appropriate columns and table
    KT <- tauSeasonal(data, TRUE, stats.median,
                      stats.minYear, stats.maxYear)
    if (is.null(KT[11])) {
       KT <- tauSeasonal(data, FALSE, stats.median,
                         stats.minYear, stats.maxYear)
    }
    if (is.null(KT.Stats) == TRUE) {
       KT.Stats <- KT
    } else{
       KT.Stats <- rbind(KT.Stats, KT)
    }
    return(KT.Stats)
 }
 trend_calculator <- function(slope, median_value, p) {
    trend <-
       if (p < .05 & abs(slope) > abs(median_value) / 10.) {
          if (slope > 0) {
             2
          }
          else {
             -2
          }
       }
    else if (p < .05 & abs(slope) < abs(median_value) / 10.) {
       if (slope > 0) {
          1
       }
       else {
          -1
       }
    }
    else
       0
    return(trend)
 }
 KT.Stats <- NULL
 # Loop that goes through each managed area. List of managed areas stored in MA_Years$ManagedAreaName
 c_names <- c("AreaID", "ManagedAreaName", "ParameterName", "RelativeDepth",
              "ActivityType", "Independent", "Median", "N", "EarliestYear",
              "LatestYear", "tau", "p", "SennSlope", "SennIntercept", "Trend")
 if(n==0){
    KT.Stats <- data.frame(matrix(ncol=15, nrow=0))
    colnames(KT.Stats) <- c_names
    fwrite(KT.Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                            "_KendallTau_Stats.txt"), sep = "|")
 } else{
```

```
   for (i in 1:n) {
      values <- data[data$Use_In_Analysis == TRUE &
                        data$ManagedAreaName == MA_names[i], ]
      if (nrow(values) > 0) {
         KT.Stats <- runStats(values)
      }
   }
   KT.Stats <- as.data.frame(KT.Stats)
   c_names <- c("AreaID", "ManagedAreaName", "ParameterName", "RelativeDepth",
                "ActivityType", "Independent", "Median", "N", "EarliestYear",
                "LatestYear", "tau", "p", "SennSlope", "SennIntercept", "Trend")
   if(dim(KT.Stats)[2]==1){
      KT.Stats <- as.data.frame(t(KT.Stats))
   }
   colnames(KT.Stats) <- c_names
   rownames(KT.Stats) <- seq(1:nrow(KT.Stats))
   KT.Stats$Median <- as.numeric(KT.Stats$Median)
   KT.Stats$N <- as.integer(KT.Stats$N)
   KT.Stats$EarliestYear <- as.integer(KT.Stats$EarliestYear)
   KT.Stats$LatestYear <- as.integer(KT.Stats$LatestYear)
   KT.Stats$tau <- round(as.numeric(KT.Stats$tau), digits=4)
   KT.Stats$p <- round(as.numeric(KT.Stats$p), digits=4)
   KT.Stats$SennSlope <- as.numeric(KT.Stats$SennSlope)
   KT.Stats$SennIntercept <- as.numeric(KT.Stats$SennIntercept)
   KT.Stats$Trend <- as.integer(KT.Stats$Trend)
   fwrite(KT.Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                           "_KendallTau_Stats.txt"), sep = "|")
}
```

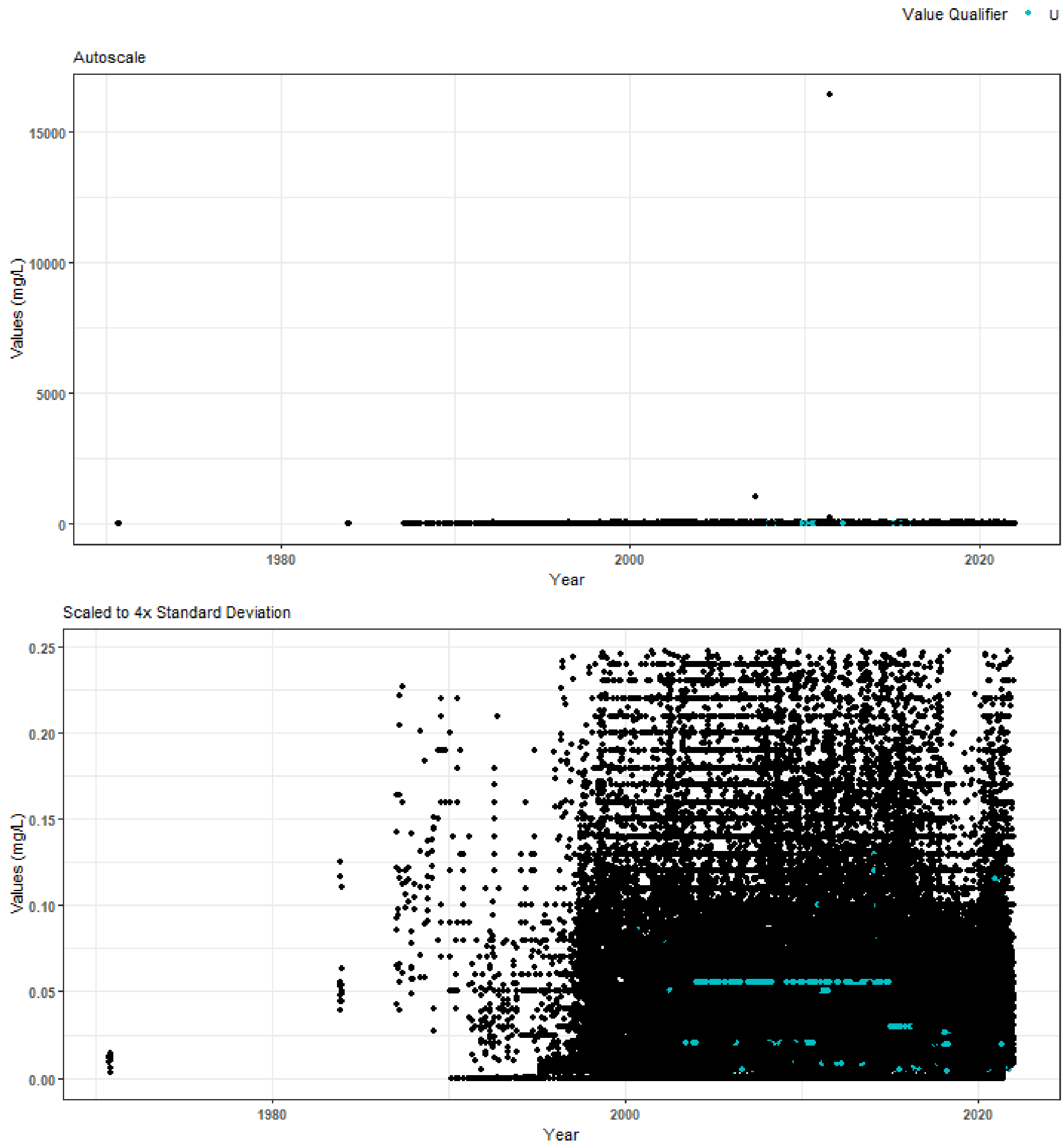
Appendix I: Scatter Plot of Entire Dataset

This part will create a scatter plot of the all data that passed initial filtering criteria with points colored based on specific value qualifiers. The values determined at the beginning (`year_lower`, `year_upper`, `min_RV`, `mn_RV`, `x_scale`, and `y_scale`) are solely for use by the plotting functions and are not output as part of the computed statistics.

```
year_lower <- min(data$Year)
year_upper <- max(data$Year)
min_RV <- min(data$ResultValue)
mn_RV <- mean(data$ResultValue[data$ResultValue <
                                  quantile(data$ResultValue, 0.98)])
sd_RV <- sd(data$ResultValue[data$ResultValue <
                                quantile(data$ResultValue, 0.98)])
x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
y_scale <- mn_RV + 4 * sd_RV


p1 <- ggplot(data = data[data$Include==TRUE,],
             aes(x = SampleDate, y = ResultValue,
                 color=VQ_Plot)) +
   geom_point(size = 1.5) +
   labs(subtitle = "Autoscale",
```

```
        x = "Year", y = paste0("Values (", unit, ")"),
        color="Value Qualifier") +
   theme_bw() +
   theme(legend.position = "top", legend.box = "horizontal",
         legend.justification = "right",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face="bold")) +
   scale_x_date(labels = date_format("%Y")) +
   {if(inc_H==TRUE){
      scale_color_manual(values = c("H"= "#F8766D", "U"= "#00BFC4",
                                   "HU" = "#7CAE00"), na.value="black")
   } else {
      scale_color_manual(values = c("U"= "#00BFC4"), na.value="black")
   }}


p2 <- ggplot(data = data[data$Include==TRUE,],
             aes(x = SampleDate, y = ResultValue,
                 color=VQ_Plot)) +
   geom_point(size = 1.5) +
   ylim(min_RV, y_scale) +
   labs(subtitle = "Scaled to 4x Standard Deviation",
        x = "Year", y = paste0("Values (", unit, ")")) +
   theme_bw() +
   theme(legend.position = "none",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold")) +
   scale_x_date(labels = date_format("%Y")) +
   {if(inc_H==TRUE){
      scale_color_manual(values = c("H"= "#F8766D", "U"= "#00BFC4",
                                   "HU" = "#7CAE00"), na.value="black")
   } else {
      scale_color_manual(values = c("U"= "#00BFC4"), na.value="black")
   }}

leg <- get_legend(p1)
pset <- ggarrange(leg, p1 + theme(legend.position = "none"), p2,
                  ncol = 1, heights = c(0.1, 1, 1))

p0 <- ggplot() + labs(title = "Scatter Plot for Entire Dataset") +
   theme_bw() + theme(plot.title = element_text(face="bold", hjust = 0.5),
                      panel.border = element_blank(),
                      panel.grid.major = element_blank(),
                      panel.grid.minor = element_blank(),
                      axis.line = element_blank())

ggarrange(p0, pset, ncol = 1, heights = c(0.1, 1))
```

Scatter Plot for Entire Dataset



Appendix II: Dataset Summary Box Plots

Box plots are created by using the entire data set and excludes any data that has been previously filtered out. The scripts that create plots follow this format

1. Use the data set that only has `Use_In_Analysis` of `TRUE`
2. Set what values are to be used for the x-axis, y-axis, and the variable that should determine groups for the box plots
3. Set the plot type as a box plot with the size of the outlier points
4. Create the title, x-axis, y-axis, and color fill labels
5. Set the y and x limits
6. Make the axis labels bold
7. Plot the arrangement as a set of panels

This set of box plots are grouped by year.

```
min_RV <- min(data$ResultValue[data$Include == TRUE])
mn_RV <- mean(data$ResultValue[data$Include == TRUE &
                                  data$ResultValue <
                                  quantile(data$ResultValue, 0.98)])
sd_RV <- sd(data$ResultValue[data$Include == TRUE &
                                data$ResultValue <
                                quantile(data$ResultValue, 0.98)])
y_scale <- mn_RV + 4 * sd_RV

p1 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = Year, y = ResultValue, group = Year)) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Autoscale", x = "Year",
        y = paste0("Values (", unit, ")")) +
   theme_bw() +
   theme(axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

p2 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = Year, y = ResultValue, group = Year)) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation", x = "Year",
        y = paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   theme_bw() +
   theme(axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

p3 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = as.integer(Year), y = ResultValue, group = Year)) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation, Last 10 Years",
        x = "Year", y = paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits = c(max(data$Year) - 10.5, max(data$Year)+1),
                      breaks = seq(max(data$Year) - 10, max(data$Year), 2)) +
   theme_bw() +
   theme(axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

set <- ggarrange(p1, p2, p3, ncol = 1)

p0 <- ggplot() + labs(title = "Summary Box Plots for Entire Data",
```

```
                         subtitle = "By Year") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust = 0.5),
         panel.border = element_blank(), panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(), axis.line = element_blank())

Yset <- ggarrange(p0, set, ncol=1, heights = c(0.07, 1))
```

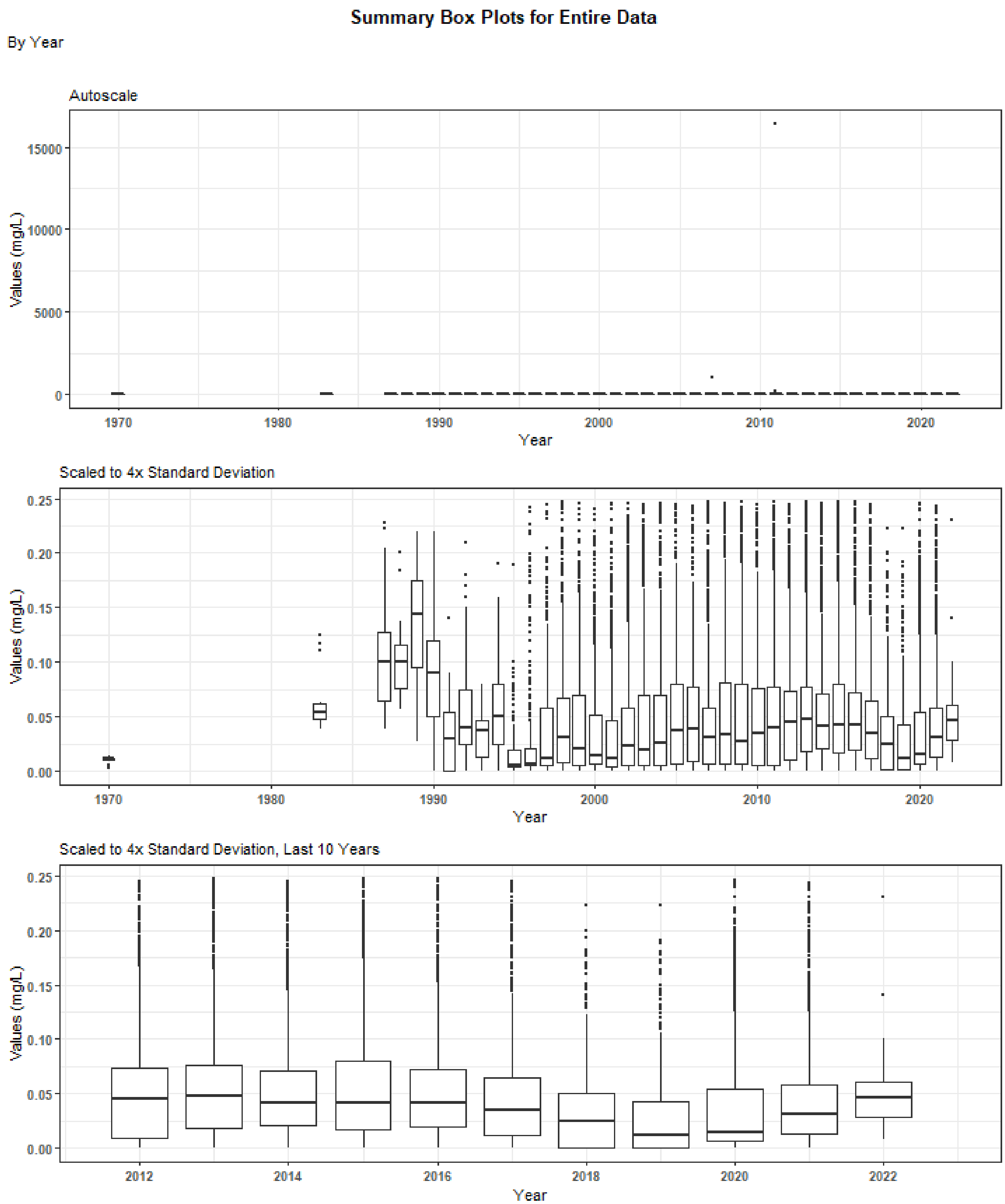
This set of box plots are grouped by year and month with the color being related to the month.

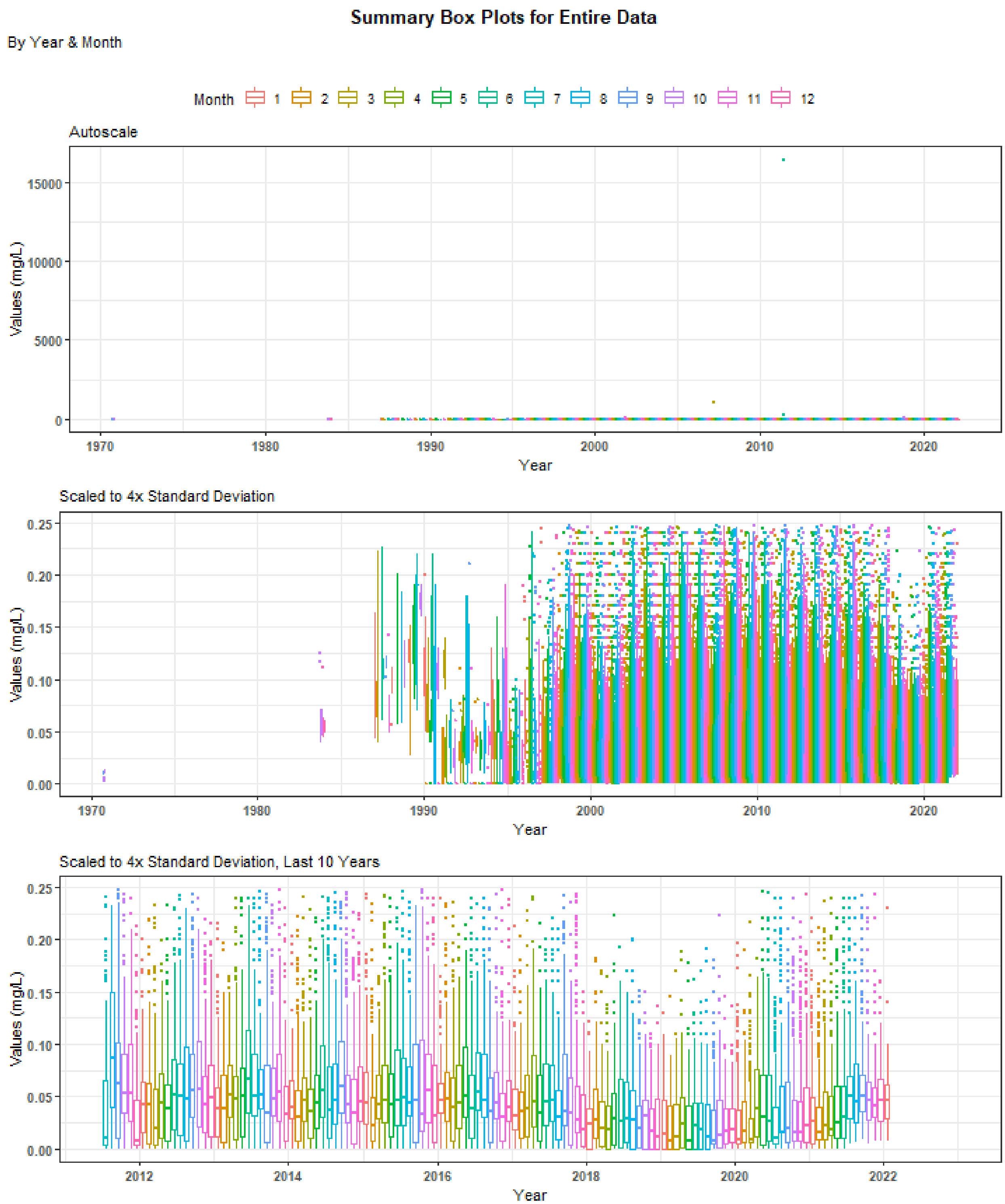
```
p1 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = YearMonthDec, y = ResultValue,
                 group = YearMonth, color = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Autoscale", x = "Year",
        y = paste0("Values (", unit, ")"), color="Month") +
   theme_bw() +
   theme(legend.position = "top", legend.box = "horizontal",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold")) +
   guides(color = guide_legend(nrow = 1))

p2 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = YearMonthDec, y = ResultValue,
                 group = YearMonth, color = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation",
        x = "Year", y = paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   theme_bw() +
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

p3 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = YearMonthDec, y = ResultValue,
                 group = YearMonth, color = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation, Last 10 Years",
        x = "Year", y = paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits = c(max(data$Year) - 10.5, max(data$Year)+1),
                      breaks = seq(max(data$Year) - 10, max(data$Year), 2)) +
   theme_bw() +
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))
leg <- get_legend(p1)
set <- ggarrange(leg, p1 + theme(legend.position = "none"), p2, p3, ncol = 1,
                 heights = c(0.1, 1, 1, 1))

p0 <- ggplot() + labs(title = "Summary Box Plots for Entire Data",
                      subtitle = "By Year & Month") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust = 0.5),
         panel.border = element_blank(), panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(), axis.line = element_blank())
```

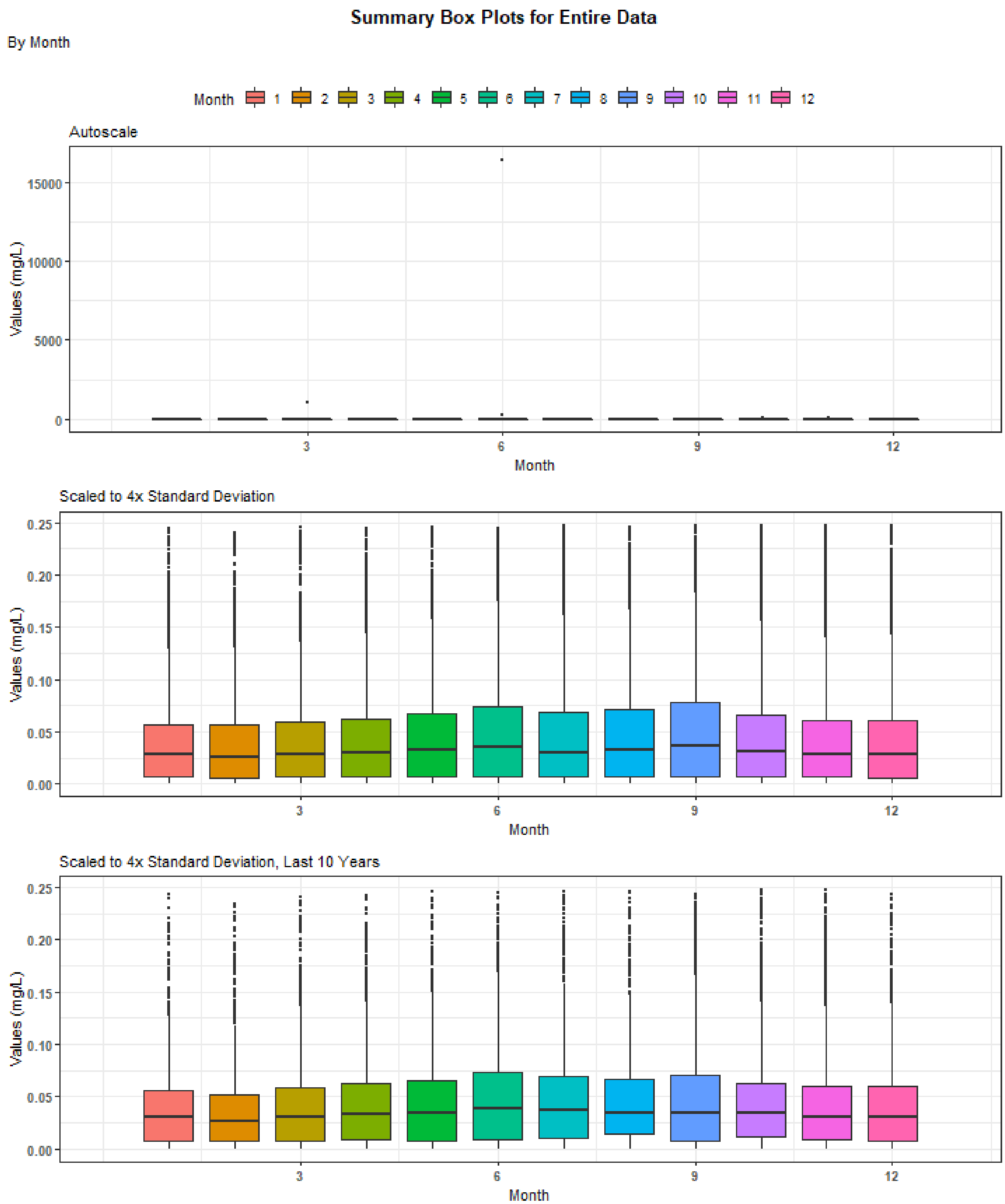
```r
YMset <- ggarrange(p0, set, ncol=1, heights = c(0.07, 1))
```

The following box plots are grouped by month with fill color being related to the month. This is designed to view potential seasonal trends.

```r
p1 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Autoscale", x = "Month",
        y = paste0("Values (", unit, ")"), fill="Month") +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "top", legend.box = "horizontal",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold")) +
   guides(fill = guide_legend(nrow = 1))

p2 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation",
        x = "Month", y = paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

p3 <- ggplot(data = data[data$Include == TRUE &
                           data$Year >= max(data$Year) - 10, ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation, Last 10 Years",
        x = "Month", y = paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))
leg <- get_legend(p1)
set <- ggarrange(leg, p1 + theme(legend.position = "none"), p2, p3, ncol = 1,
                 heights = c(0.1, 1, 1, 1))

p0 <- ggplot() + labs(title = "Summary Box Plots for Entire Data",
                      subtitle = "By Month") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust = 0.5),
         panel.border = element_blank(), panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(), axis.line = element_blank())

Mset <- ggarrange(p0, set, ncol=1, heights = c(0.07, 1))
```


Summary Box Plots for Entire Data
By Year
Autoscale
Values (mg/L)
15000
10000
5000
0
1970
1980
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.25
0.20
0.15
0.10
0.05
0.00
1970
1980
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.25
0.20
0.15
0.10
0.05
0.00
2012
2014
2016
2018
2020
2022
Year

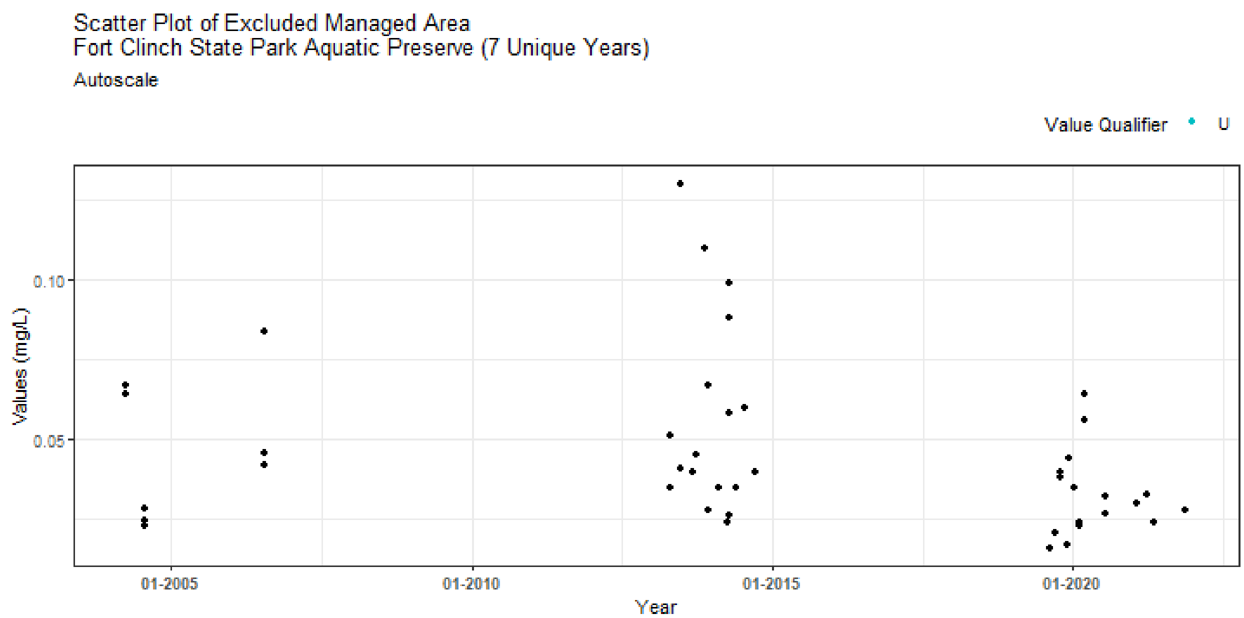

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15000
10000
5000
0
1970
1980
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.25
0.20
0.15
0.10
0.05
0.00
1970
1980
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.25
0.20
0.15
0.10
0.05
0.00
2012
2014
2016
2018
2020
2022
Year

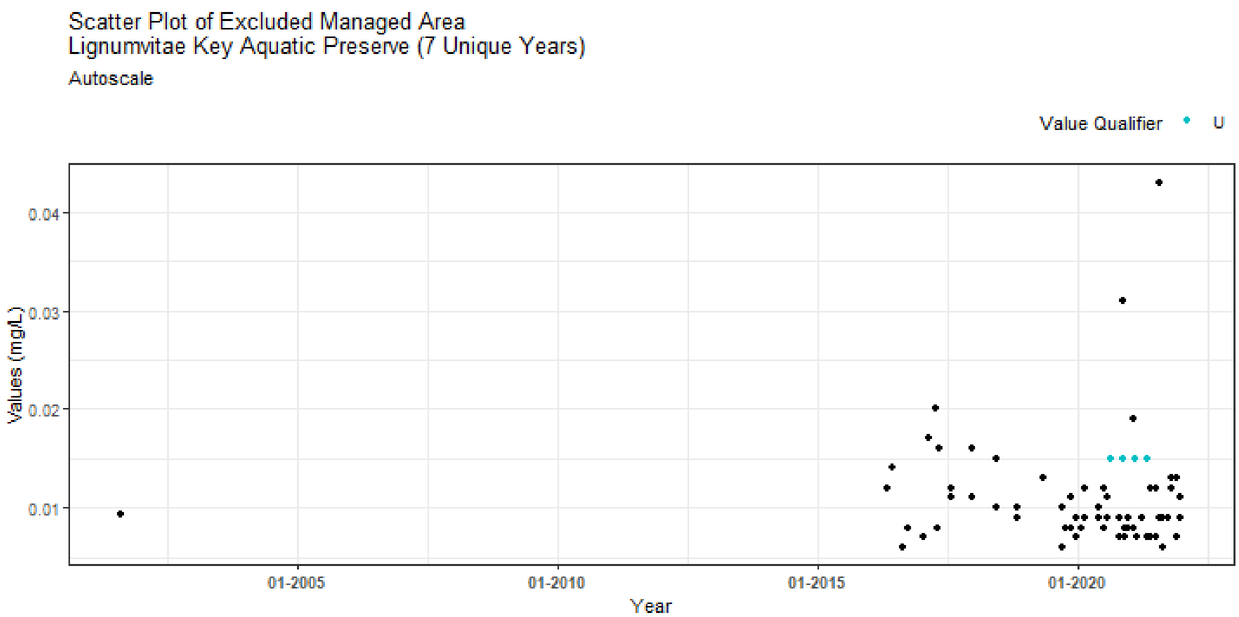


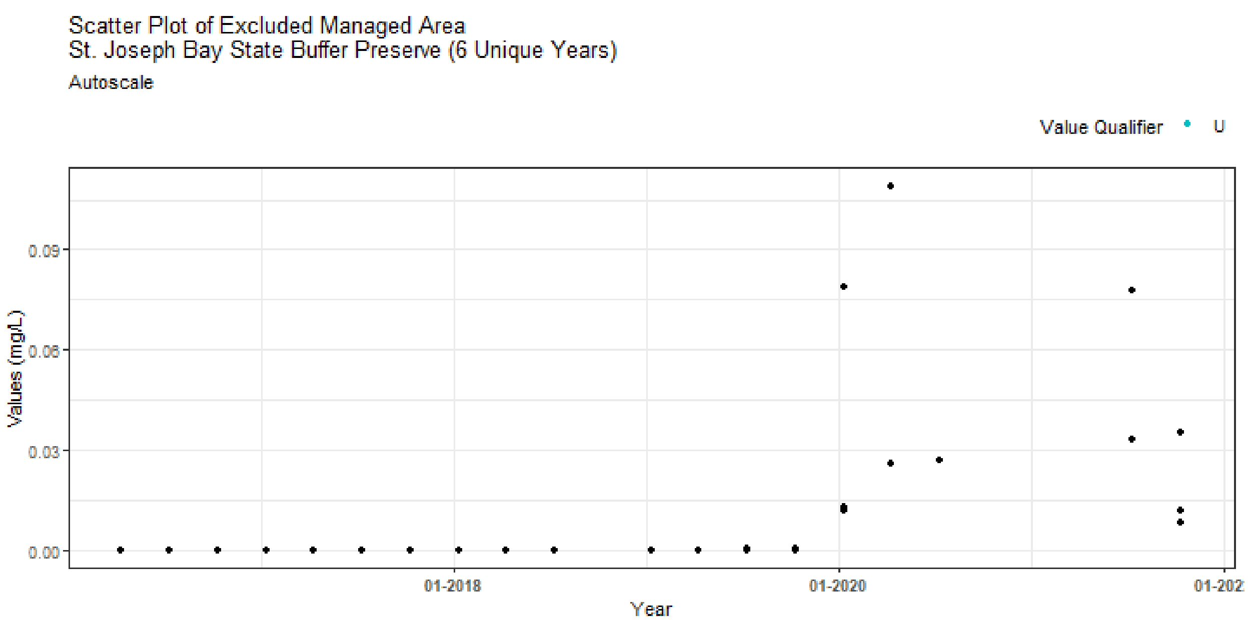
Appendix III: Excluded Managed Areas

Scatter plots of data values are created for managed areas that have fewer than 10 separate years of data entries. Data points are colored based on specific value qualifiers of interest.

```r
MA_Exclude <- MA_Years[MA_Years$Enough_Time==FALSE,]
MA_Exclude <- MA_Exclude[order(MA_Exclude$ManagedAreaName),]
z=length(MA_Exclude$ManagedAreaName)

if(z==0){
   print("There are no managed areas that qualify.")
} else {
   for(i in 1:z){

      p1<-ggplot(data=data[data$ManagedAreaName==MA_Exclude$ManagedAreaName[i]&
                              data$Include == TRUE, ],
                 aes(x = SampleDate, y = ResultValue, color=VQ_Plot)) +
         geom_point() +
         labs(title = paste0("Scatter Plot of Excluded Managed Area\n",
                             MA_Exclude$ManagedAreaName[i], " (",
                             MA_Exclude$N[i], " Unique Years)"),
              subtitle="Autoscale", x = "Year",
              y = paste0("Values (", unit, ")"), color="Value Qualifier") +
         theme_bw() +
         theme(legend.position = "top", legend.box = "horizontal",
               legend.justification = "right",

               axis.text.x = element_text(face = "bold")) +
         scale_x_date(labels = date_format("%m-%Y")) +
         {if(inc_H==TRUE){
            scale_color_manual(values = c("H"= "#F8766D", "U"= "#00BFC4",
                                          "HU" = "#7CAE00"), na.value="black")
         } else {
            scale_color_manual(values = c("U"= "#00BFC4"), na.value="black")
         }}
      print(p1)
   }
}
```







Appendix IV: Managed Area Trendlines

The plots created in this section are designed to show the general trend of the data. Data is taken and grouped by `ManagedAreaName`. The trendlines on the plots are created using the Senn slope and intercept from the seasonal Kendall Tau analysis. The scripts that create plots follow this format

1. Use the data set that only has `Use_In_Analysis` of `TRUE` for the desired managed area
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Determine the minimum, mean, and standard deviation for the data to be used for y-axis scales
    - Excludes the top 2% of values to reduce the impact of extreme outliers on the y-axis scale
4. Set what values are to be used for the x-axis, y-axis, and the variable that should determine groups for the plots

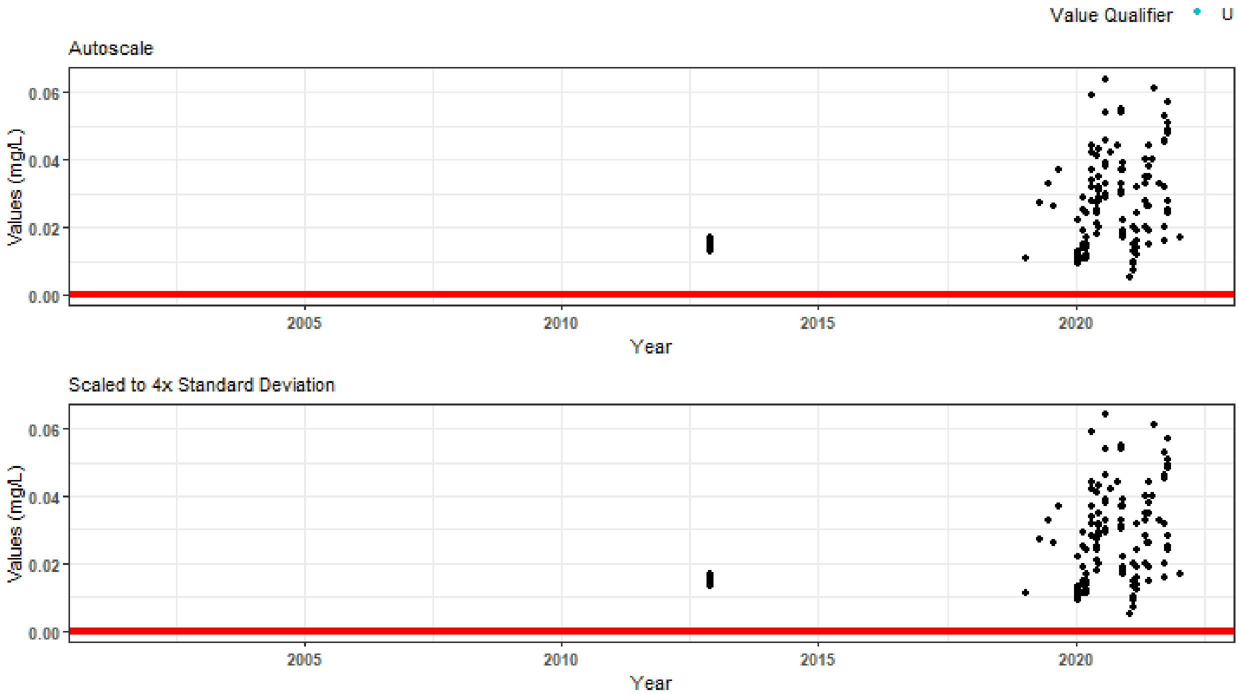
5. Set the plot type as a point plot with the size of the points
6. Add the linear trend
7. Create the title, x-axis, y-axis, and color fill labels
8. Set the y and x limits
9. Make the axis labels bold
10. Plot the arrangement as a set of panels

```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$Use_In_Analysis == TRUE &
                              data$ManagedAreaName == MA_names[i],]
      year_lower <- min(plot_data$Year)
      year_upper <- max(plot_data$Year)
      min_RV <- min(plot_data$ResultValue)
      mn_RV <- mean(plot_data$ResultValue[data$ResultValue <
                                  quantile(data$ResultValue, 0.98)])
      sd_RV <- sd(plot_data$ResultValue[data$ResultValue <
                                quantile(data$ResultValue, 0.98)])
      x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
      y_scale <- mn_RV + 4 * sd_RV

      tau <- KT.Stats$tau[KT.Stats$ManagedAreaName==MA_names[i]]
      s_slope <- KT.Stats$SennSlope[KT.Stats$ManagedAreaName==MA_names[i]]
      s_int <- KT.Stats$SennIntercept[KT.Stats$ManagedAreaName==MA_names[i]]
      trend <- KT.Stats$Trend[KT.Stats$ManagedAreaName==MA_names[i]]
      p <- KT.Stats$p[KT.Stats$ManagedAreaName==MA_names[i]]

      model <- lm(ResultValue ~ DecDate,
                  data = plot_data)
      m_int <- coef(model)[[1]]
      m_slope <- coef(model)[[2]]
      p1 <- ggplot(data = plot_data,
                   aes(x = DecDate, y = ResultValue,
                       color=VQ_Plot)) +
         geom_point(size = 1.5) +
         geom_abline(aes(slope=s_slope, intercept=s_int),
                     color="red", size=1.5) +
         labs(subtitle = "Autoscale",
              x = "Year", y = paste0("Values (", unit, ")"),
              color="Value Qualifier") +
         theme_bw() +
         theme(legend.position = "top", legend.box = "horizontal",
               legend.justification = "right",
               axis.text.x = element_text(face = "bold"),
               axis.text.y = element_text(face="bold")) +
         {if(inc_H==TRUE){
            scale_color_manual(values = c("H"= "#F8766D", "U"= "#00BFC4",
                                          "HU" = "#7CAE00"), na.value="black")
         } else {
            scale_color_manual(values = c("U"= "#00BFC4"), na.value="black")
         }}
```

```
    p2 <- ggplot(data = plot_data,
                 aes(x = DecDate, y = ResultValue,
                     color=VQ_Plot)) +
      geom_point(size = 1.5) +
      geom_abline(aes(slope=s_slope, intercept=s_int),
                  color="red", size=1.5) +
      ylim(min_RV, y_scale) +
      labs(subtitle = "Scaled to 4x Standard Deviation",
           x = "Year", y = paste0("Values (", unit, ")")) +
      theme_bw() +
      theme(legend.position = "none",
            axis.text.x = element_text(face = "bold"),
            axis.text.y = element_text(face="bold")) +
      {if(inc_H==TRUE){
        scale_color_manual(values = c("H"= "#F8766D", "U"= "#00BFC4",
                                      "HU" = "#7CAE00"), na.value="black")
      } else {
        scale_color_manual(values = c("U"= "#00BFC4"), na.value="black")
      }}
    leg <- get_legend(p1)
    KTset <- ggarrange(leg, p1 + theme(legend.position = "none"), p2,
                       ncol = 1, heights = c(0.1, 1, 1))

    p0 <- ggplot() + labs(title = paste0("Data Points with Trendlines for ",
                                         MA_names[i]),
                          subtitle =paste0("Senn Slope = ", s_slope,
                                           ",     Senn Intercept = ", s_int,
                                           "\nTrend = ", trend,
                                           ",     tau = ", tau,
                                           ",     p = ", p,
                                           "\nLinear Trendline: ",
                                           "y = ", m_slope,"x + ",m_int)) +
      theme_bw() + theme(plot.title = element_text(face="bold", hjust = 0.5),
                          panel.border = element_blank(),
                          panel.grid.major = element_blank(),
                          panel.grid.minor = element_blank(),
                          axis.line = element_blank())

    print(ggarrange(p0, KTset, ncol = 1, heights = c(0.15, 1)))
    rm(plot_data)
  }
}
```

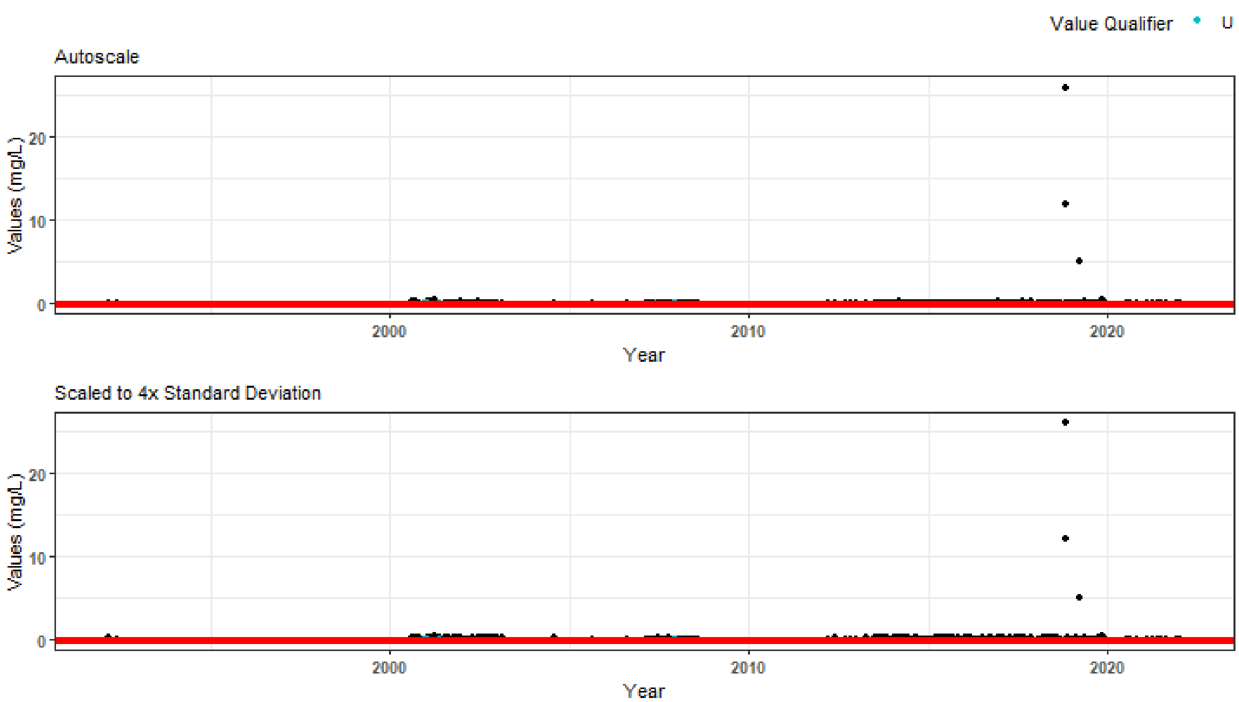
Data Points with Trendlines for Alligator Harbor Aquatic Preserve

Senn Slope = 0.0000016666666666667,   Senn Intercept = -0.00340129195804196
Trend = 1,   tau = 0.2947,   p = 0
Linear Trendline: y = 0.000840248496192683x + -1.68694658364648



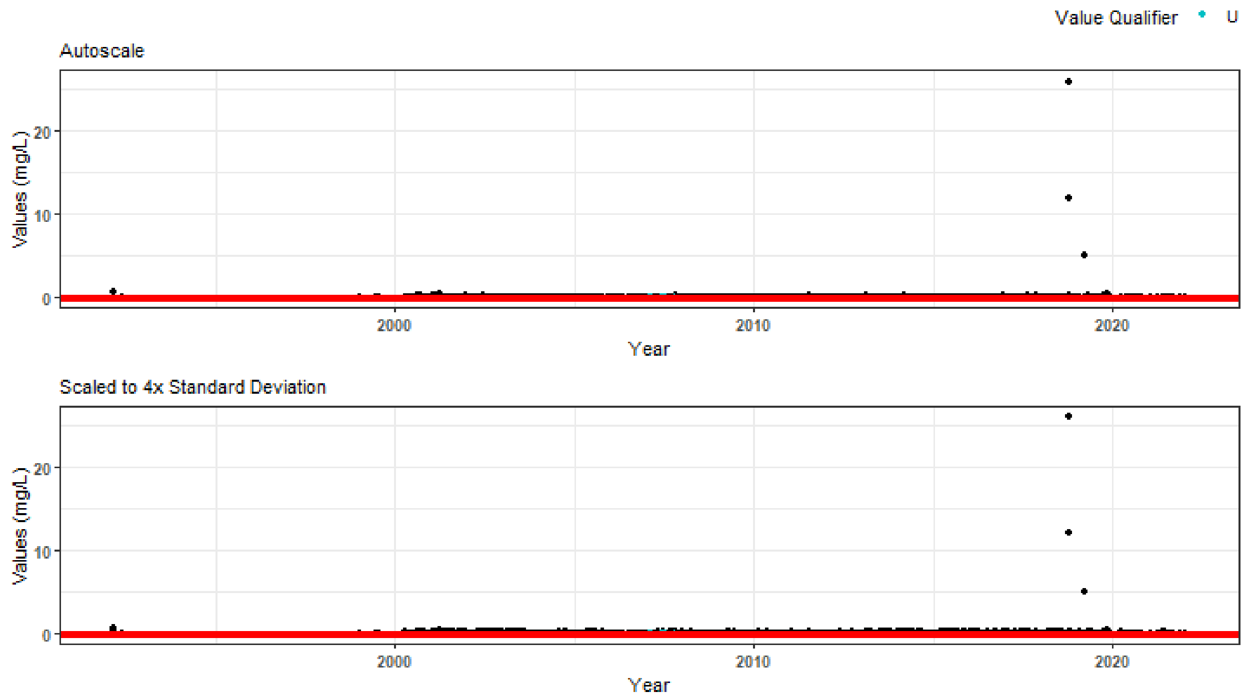
Data Points with Trendlines for Apalachicola Bay Aquatic Preserve

Senn Slope = -0.000249999999999999,   Senn Intercept = 0.265369047619048
Trend = -1,   tau = -0.057,   p = 0.0001
Linear Trendline: y = 0.00295532425923597x + -5.89572313828038



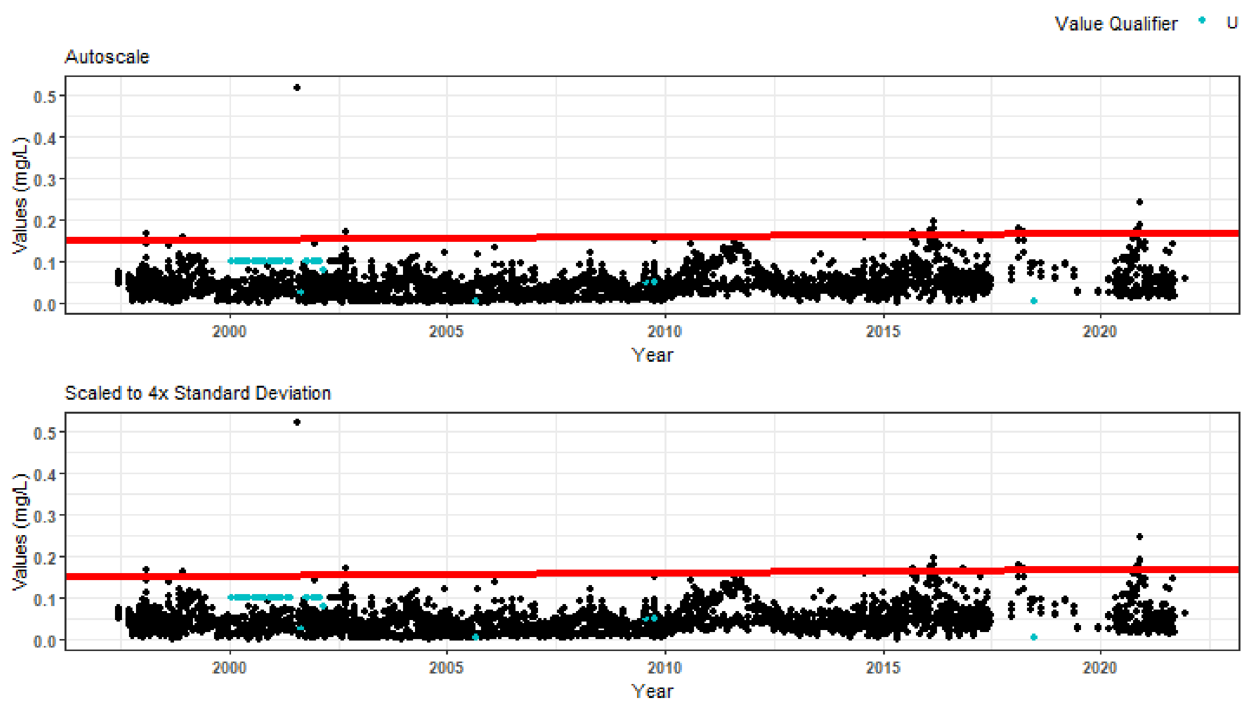
Data Points with Trendlines for Apalachicola National Estuarine Research Reserve

Senn Slope = -0.000166666666666666, Senn Intercept = 0.102678571428571
Trend = -1, tau = -0.0477, p = 0.0002
Linear Trendline: y = 0.00361799889958276x + -7.22401580283591



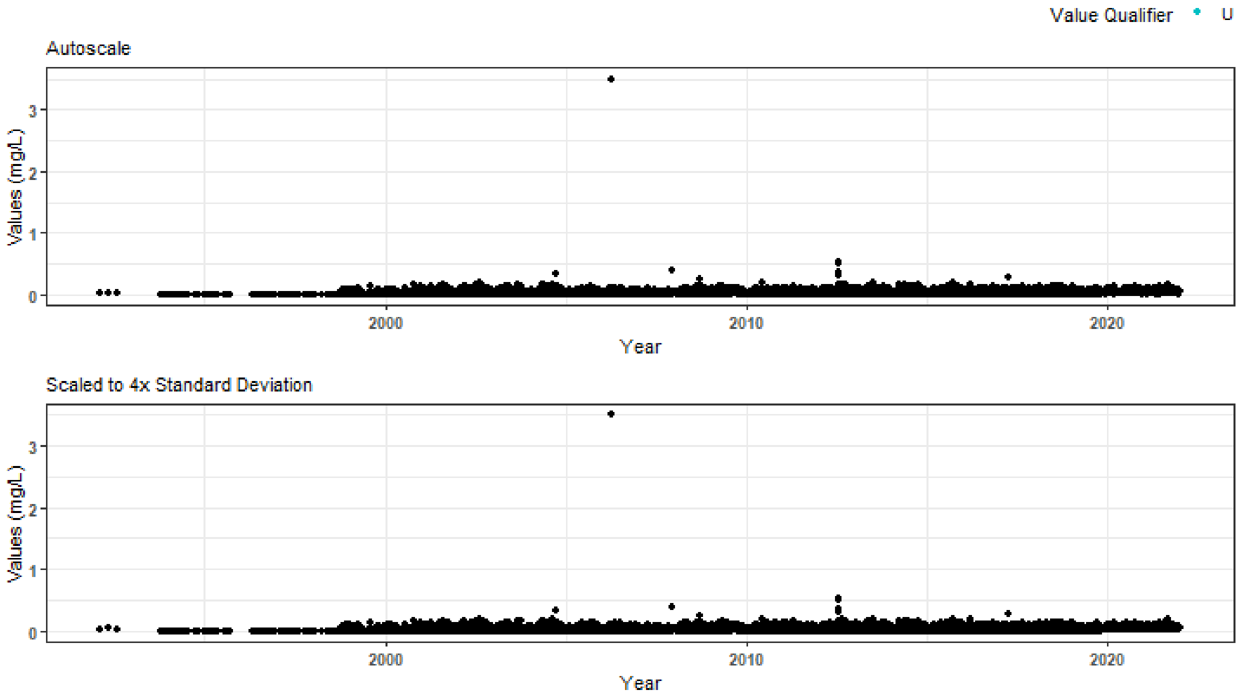
Data Points with Trendlines for Banana River Aquatic Preserve

Senn Slope = 0.000781818181818181, Senn Intercept = -1.41066561919505
Trend = 1, tau = 0.1313, p = 0
Linear Trendline: y = 0.000985478517938677x + -1.93572697210911



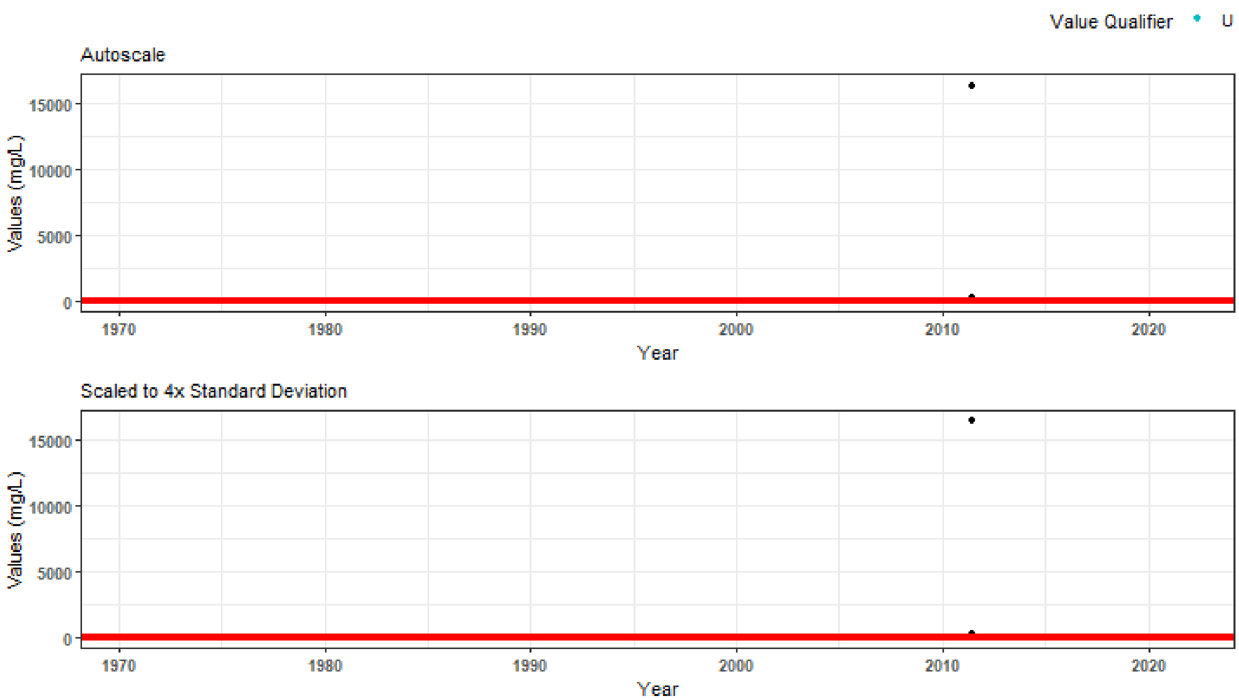
Data Points with Trendlines for Big Bend Seagrasses Aquatic Preserve

Senn Slope = 0.000881,   Senn Intercept = -2.051667
Trend = 1,   tau = 0.1886,   p = 0
Linear Trendline: y = 0.0012705786211641x + -2.52197637320058



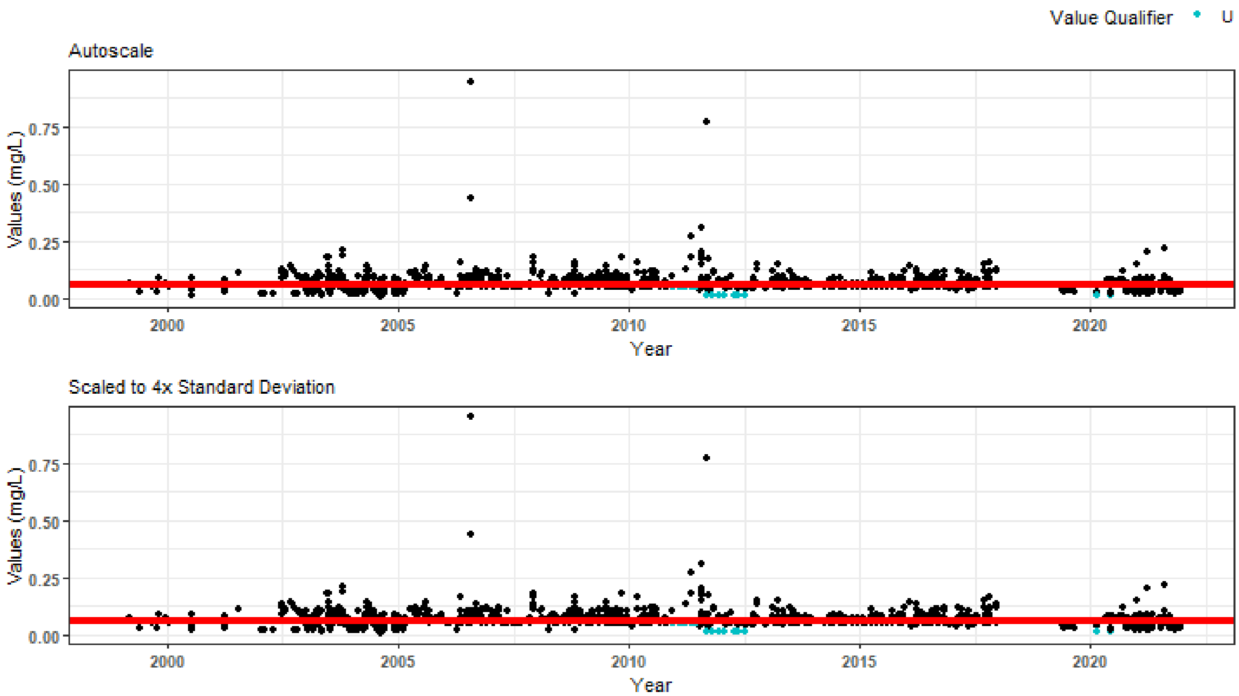
Data Points with Trendlines for Biscayne Bay Aquatic Preserve

Senn Slope = 0.0006,   Senn Intercept = -1.09027495035461
Trend = 1,   tau = 0.1524,   p = 0
Linear Trendline: y = -0.714874396662105x + 1448.10912195826



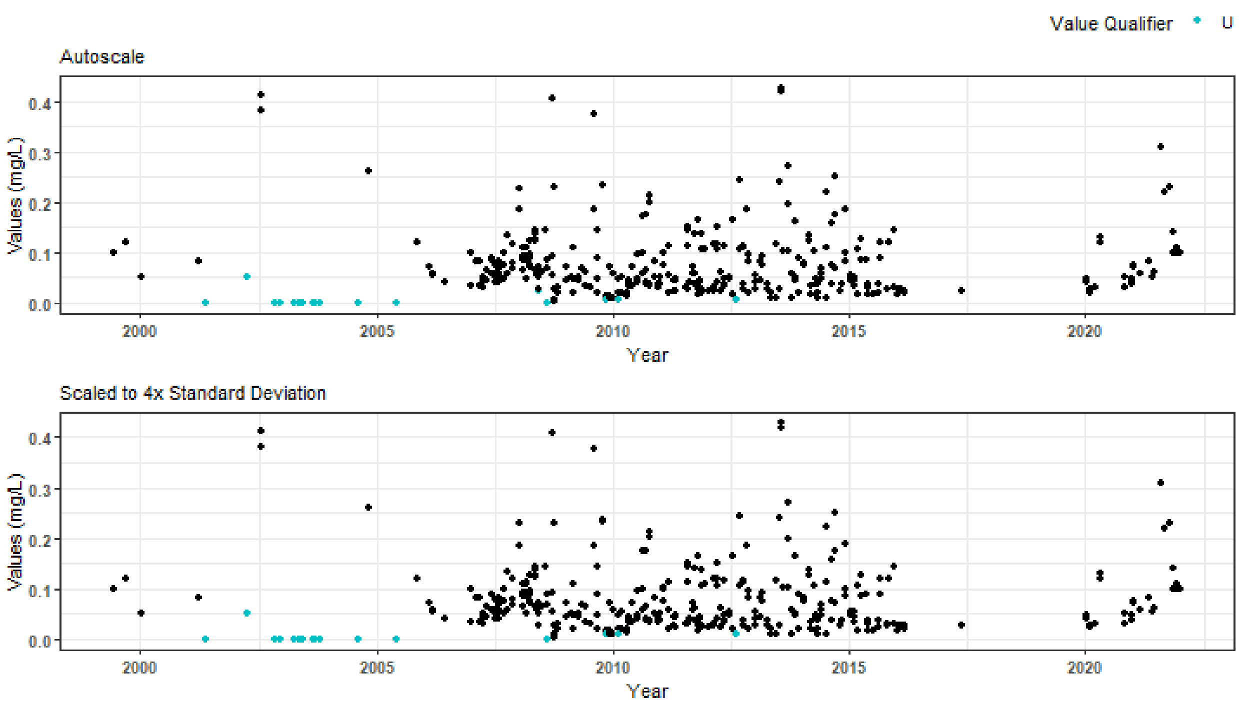
Data Points with Trendlines for Boca Ciega Bay Aquatic Preserve

Senn Slope = 0, Senn Intercept = 0.06
Trend = -1, tau = -0.0678, p = 0
Linear Trendline: y = -0.000465006249073396x + 1.00013869751421



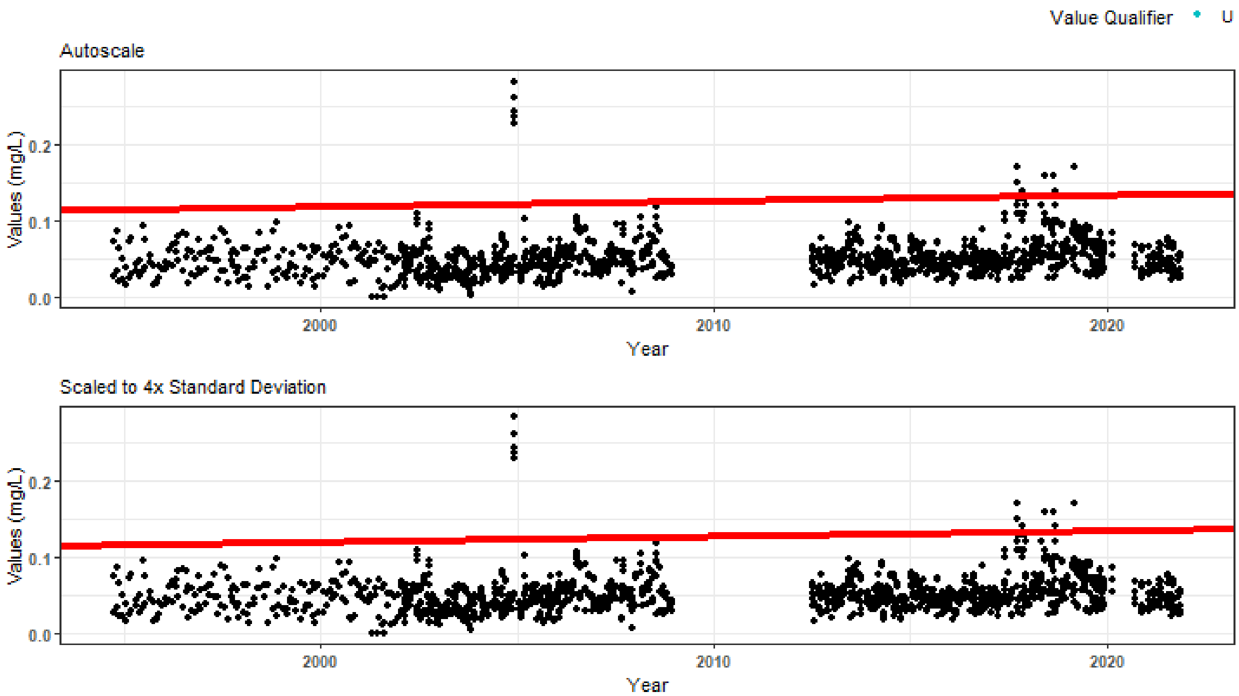
Data Points with Trendlines for Cape Haze Aquatic Preserve

Senn Slope = 0.0008, Senn Intercept = -0.0715059523809525
Trend = 0, tau = 0.0427, p = 0.2715
Linear Trendline: y = 0.000868338965259799x + -1.67280512393152



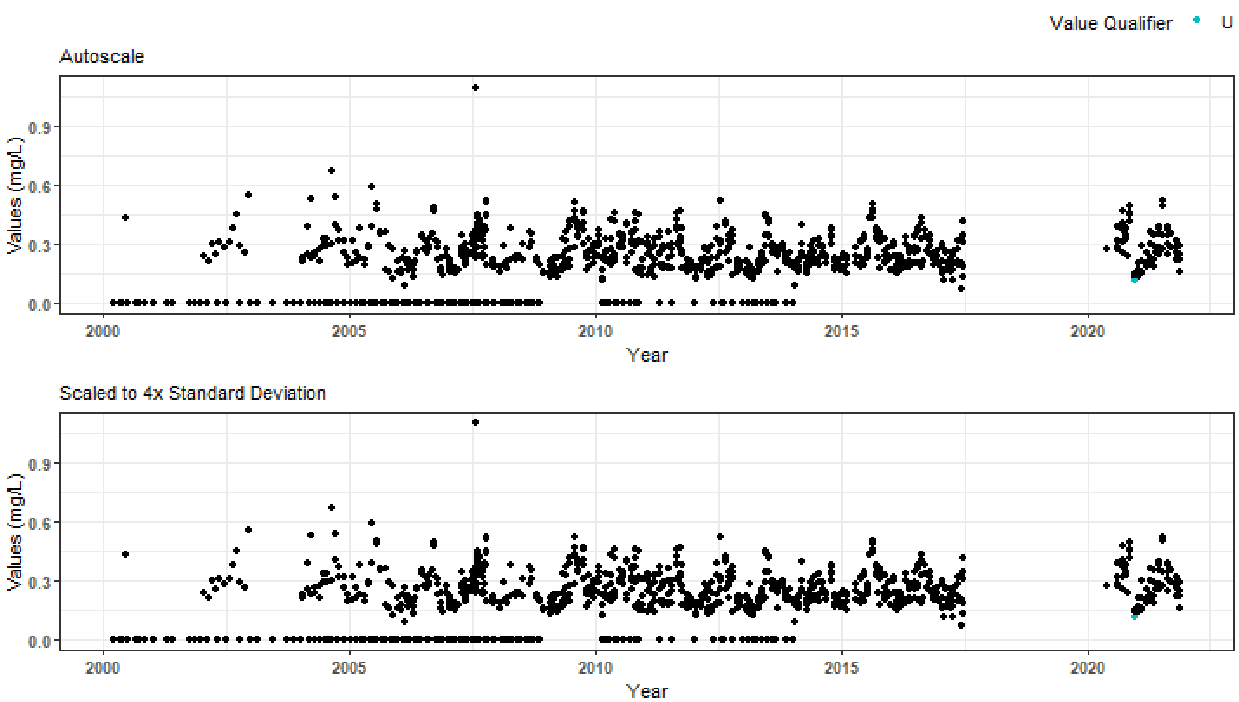
Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve

Senn Slope = 0.000744097222222222,   Senn Intercept = -1.36910666666667
Trend = 1,   tau = 0.1807,   p = 0
Linear Trendline: y = 0.000722151239013868x + -1.4020319429712



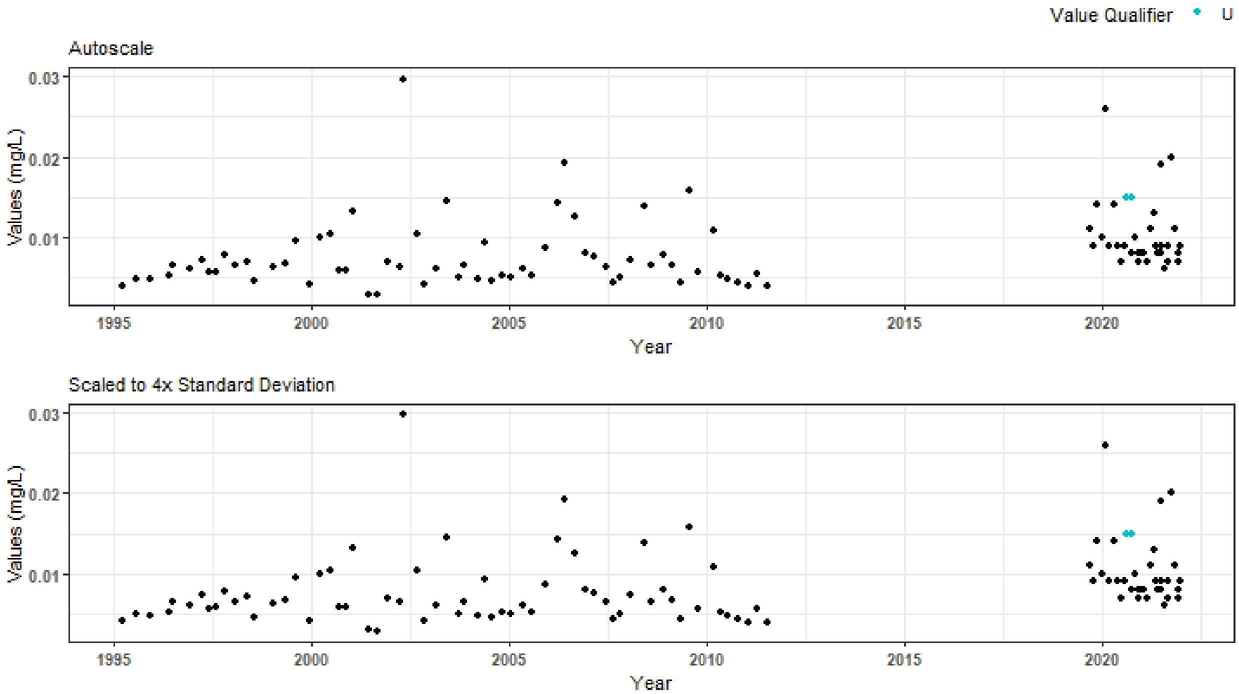
Data Points with Trendlines for Cockroach Bay Aquatic Preserve

Senn Slope = 0.00699999999999999,   Senn Intercept = -11.9089642857143
Trend = 1,   tau = 0.2032,   p = 0
Linear Trendline: y = 0.0102580318194668x + -20.4353228445522



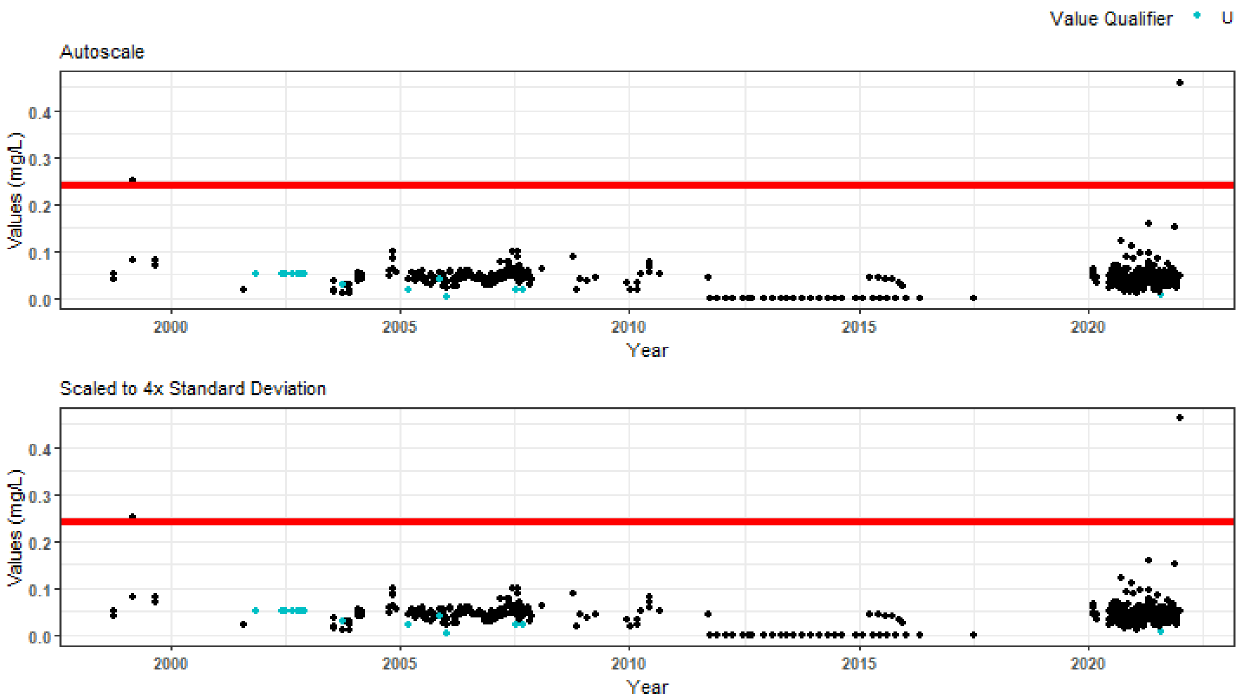
Data Points with Trendlines for Coupon Bight Aquatic Preserve

Senn Slope = 0.0001184048,   Senn Intercept = -0.277262343149038
Trend = 1,   tau = 0.3185,   p = 0.0001
Linear Trendline: y = 0.000152775760675 91x + -0.298559072713593



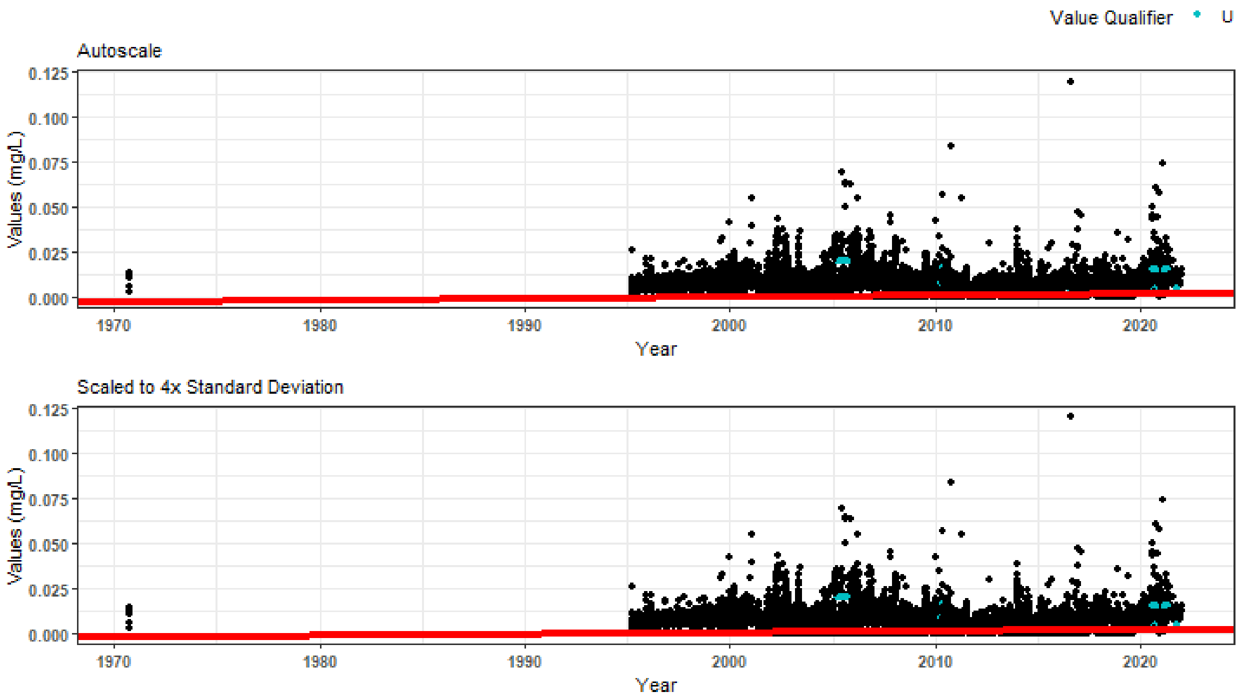
Data Points with Trendlines for Estero Bay Aquatic Preserve

Senn Slope = -0.000006,   Senn Intercept = 0.252482142857143
Trend = 0,   tau = -0.0151,   p = 0.5053
Linear Trendline: y = -0.000188490399480071x + 0.418320159932362



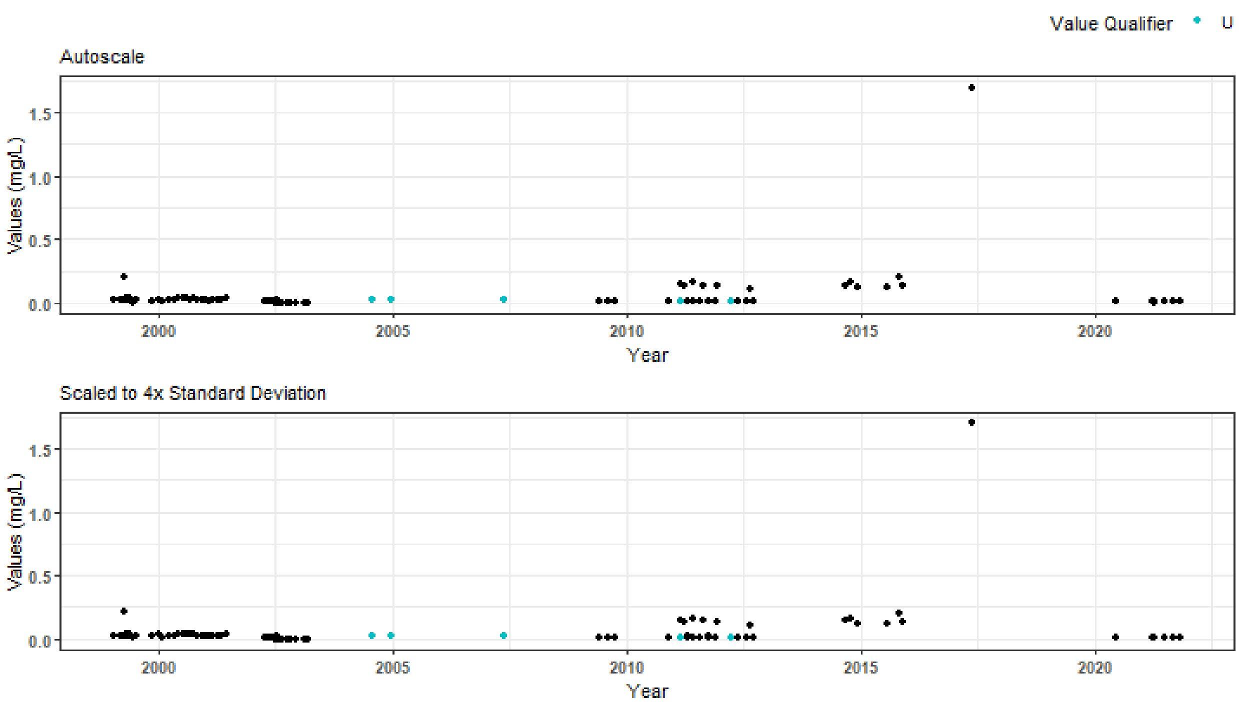
Data Points with Trendlines for Florida Keys National Marine Sanctuary

Senn Slope = 0.00008775,   Senn Intercept = -0.175069143971292
Trend = 1,   tau = 0.1291,   p = 0
Linear Trendline: y = 0.000114008539299939x + -0.222285882646983



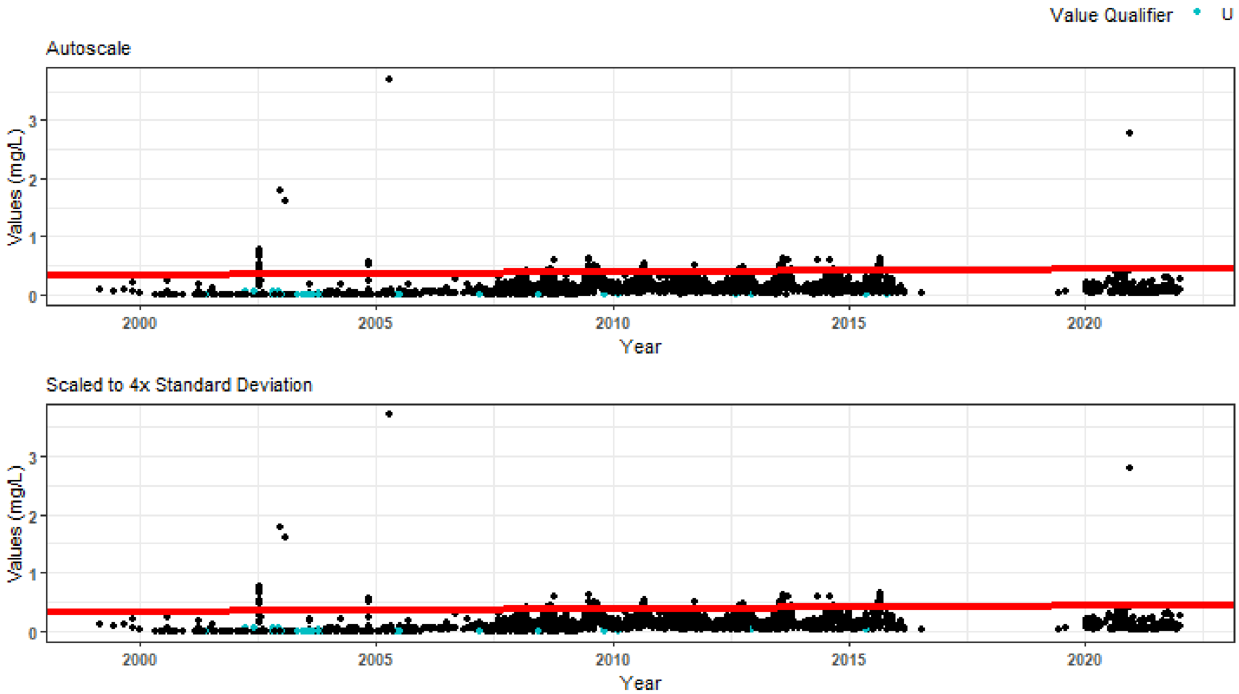
Data Points with Trendlines for Fort Pickens State Park Aquatic Preserve

Senn Slope = -0.000591666666666667,   Senn Intercept = -0.0432847222222222
Trend = 0,   tau = -0.0982,   p = 0.0583
Linear Trendline: y = 0.00358627873433203x + -7.15122406661616



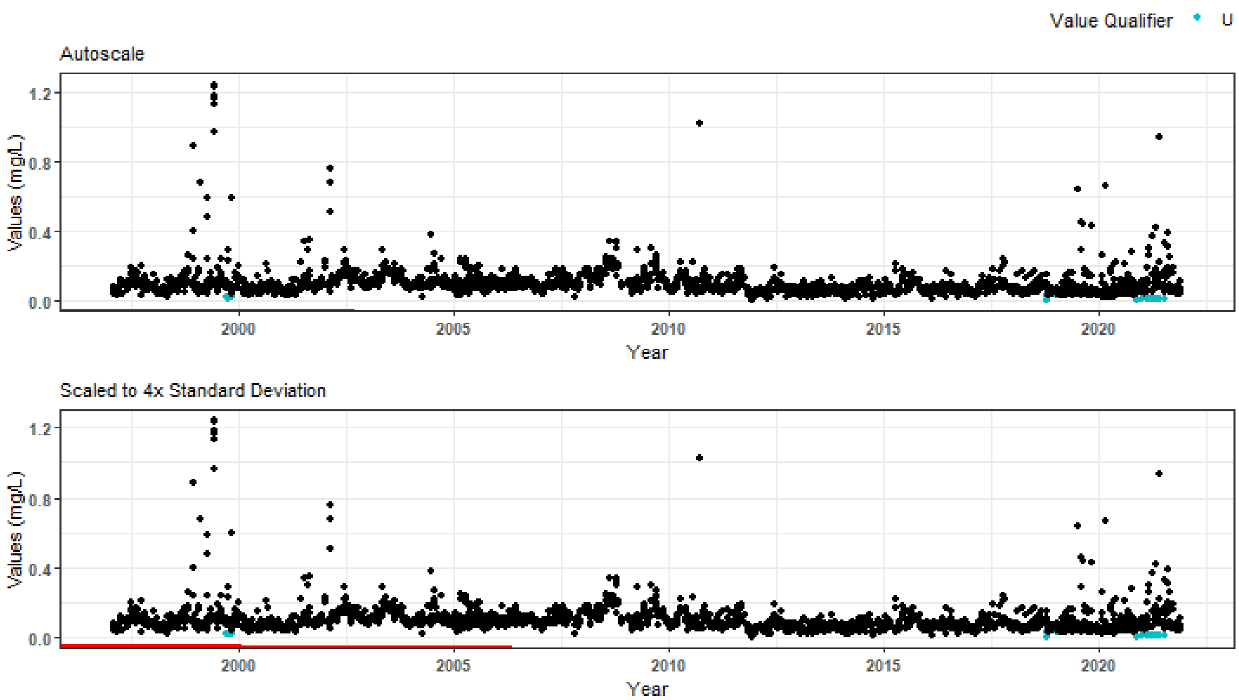
Data Points with Trendlines for Gasparilla Sound-Charlotte Harbor Aquatic Preserve

Senn Slope = 0.005,   Senn Intercept = -9.65770534583333
Trend = 1,   tau = 0.1904,   p = 0
Linear Trendline: y = 0.00375384295941343x + -7.42276334670218



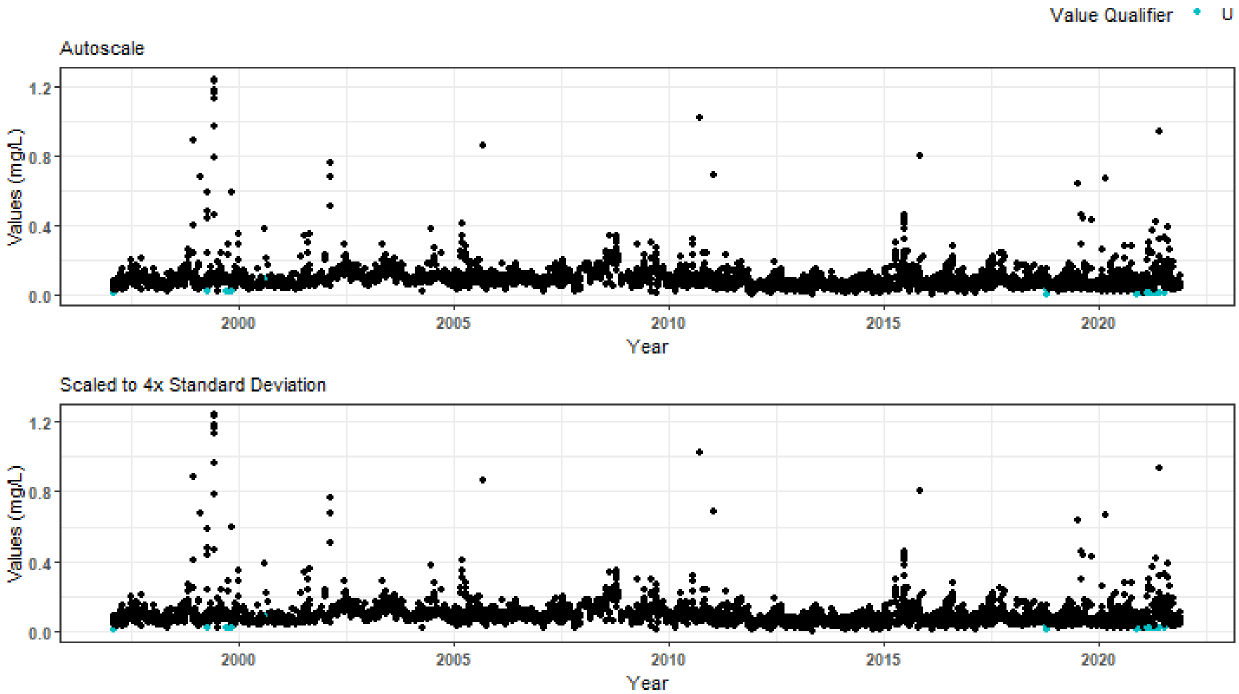
Data Points with Trendlines for Guana River Marsh Aquatic Preserve

Senn Slope = -0.00151666666666667,   Senn Intercept = 2.97211535714286
Trend = -1,   tau = -0.1962,   p = 0
Linear Trendline: y = -0.00196700850739608x + 4.05382318224379



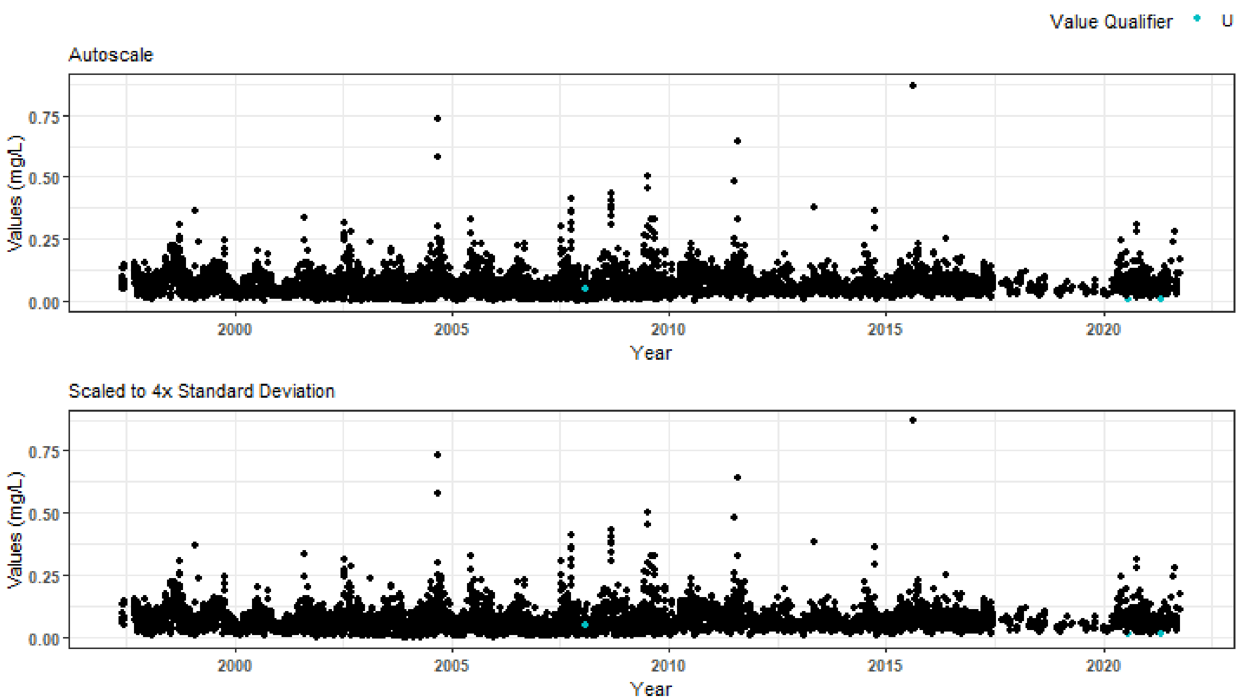
Data Points with Trendlines for Guana Tolomato Matanzas National Estuarine Research Reserve

Senn Slope = -0.00108181818181818, Senn Intercept = 2.08657941176471
Trend = -1, tau = -0.1505, p = 0
Linear Trendline: y = -0.00164902863834335x + 3.41011852595448



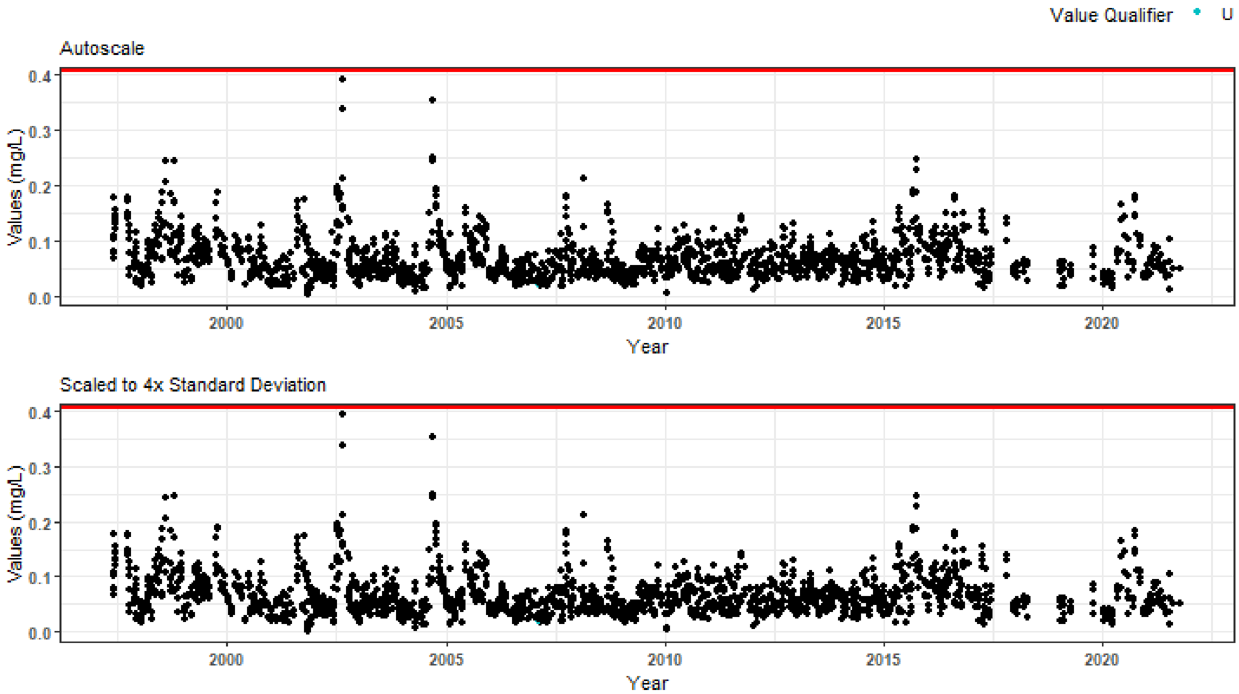
Data Points with Trendlines for Indian River-Malabar to Vero Beach Aquatic Preserve

Senn Slope = 0.000035294117647059, Senn Intercept = -0.275743097527472
Trend = 0, tau = 0.0043, p = 0.4549
Linear Trendline: y = 0.0000552789166114777x + -0.0448887270183835



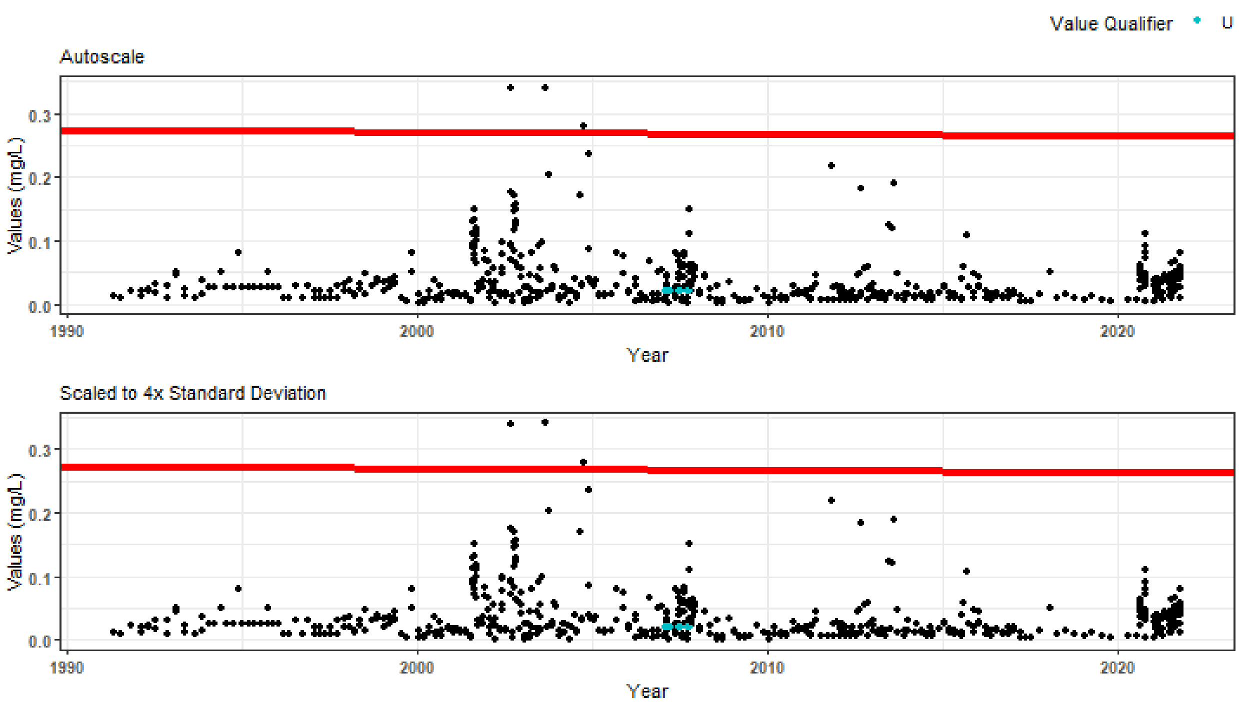
Data Points with Trendlines for Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

Senn Slope = 0,   Senn Intercept = 0.410010775401069
Trend = 0,   tau = -0.0091,   p = 0.7572
Linear Trendline: y = -0.000349405924908135x + 0.769425359652614



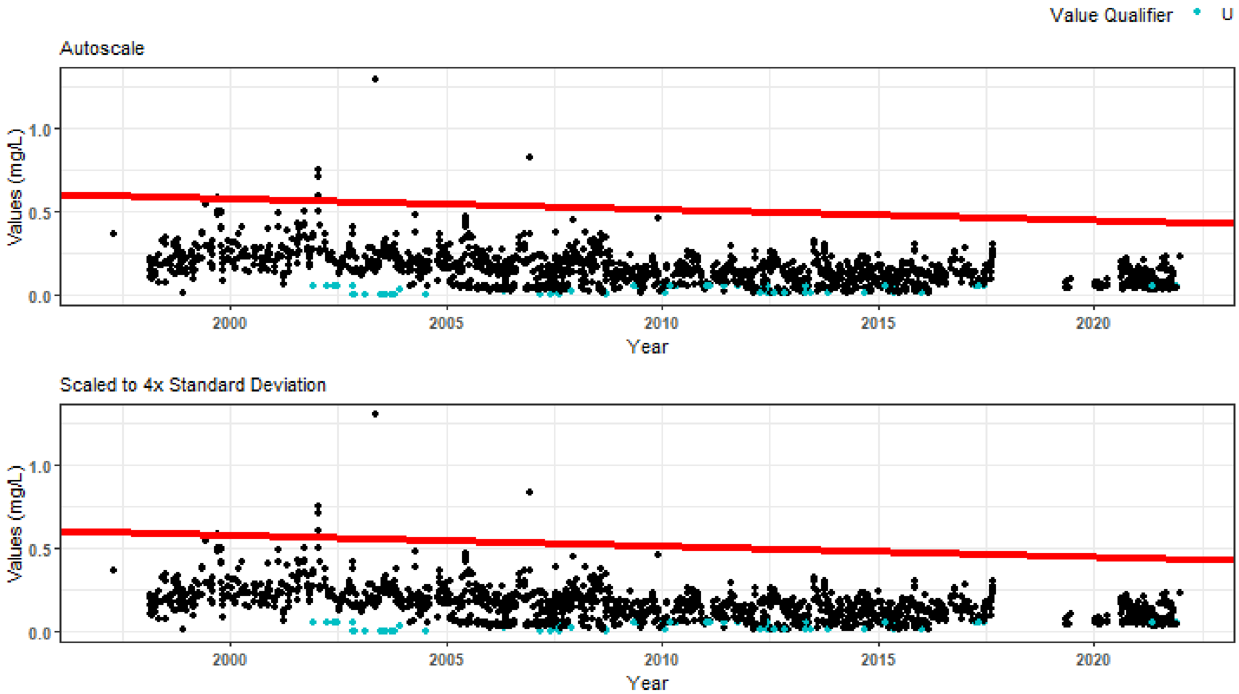
Data Points with Trendlines for Jensen Beach to Jupiter Inlet Aquatic Preserve

Senn Slope = -0.000333333333333334,   Senn Intercept = 0.937545454545454
Trend = -1,   tau = -0.1253,   p = 0.0001
Linear Trendline: y = -0.000278309393087511x + 0.595246244794624



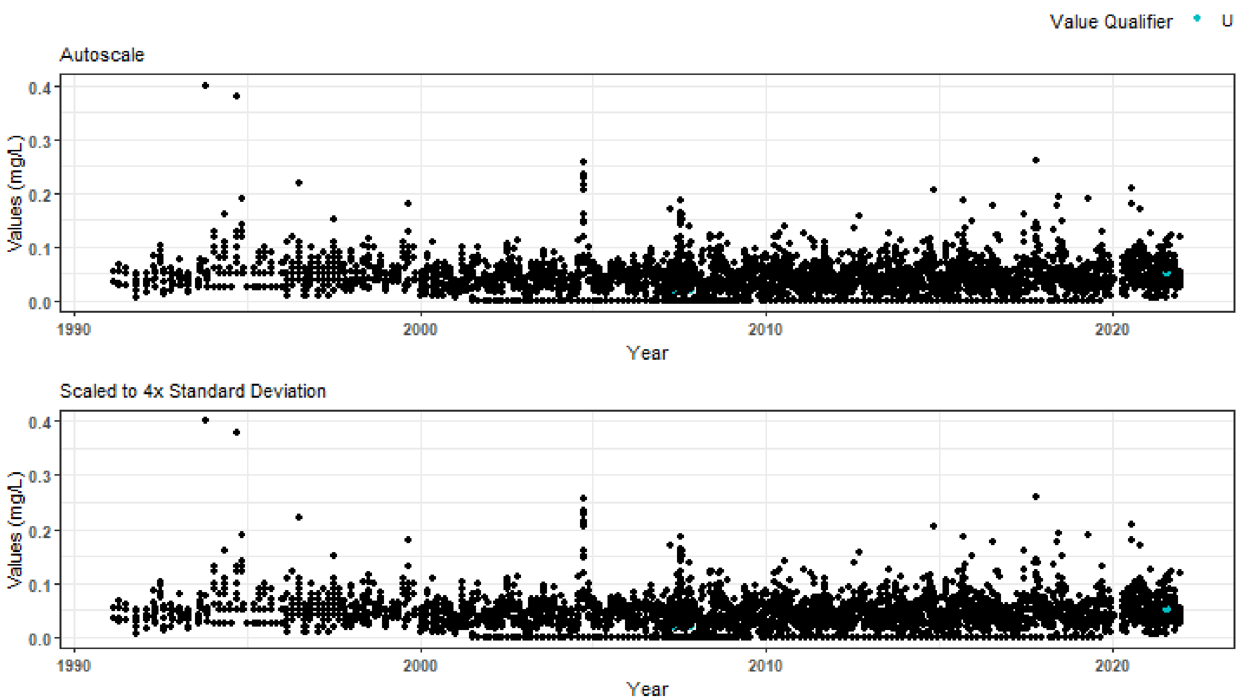
Data Points with Trendlines for Lemon Bay Aquatic Preserve

Senn Slope = -0.00646153846153846, Senn Intercept = 13.5014848484848
Trend = -1, tau = -0.299, p = 0
Linear Trendline: y = -0.00721884082863484x + 14.6533287095524



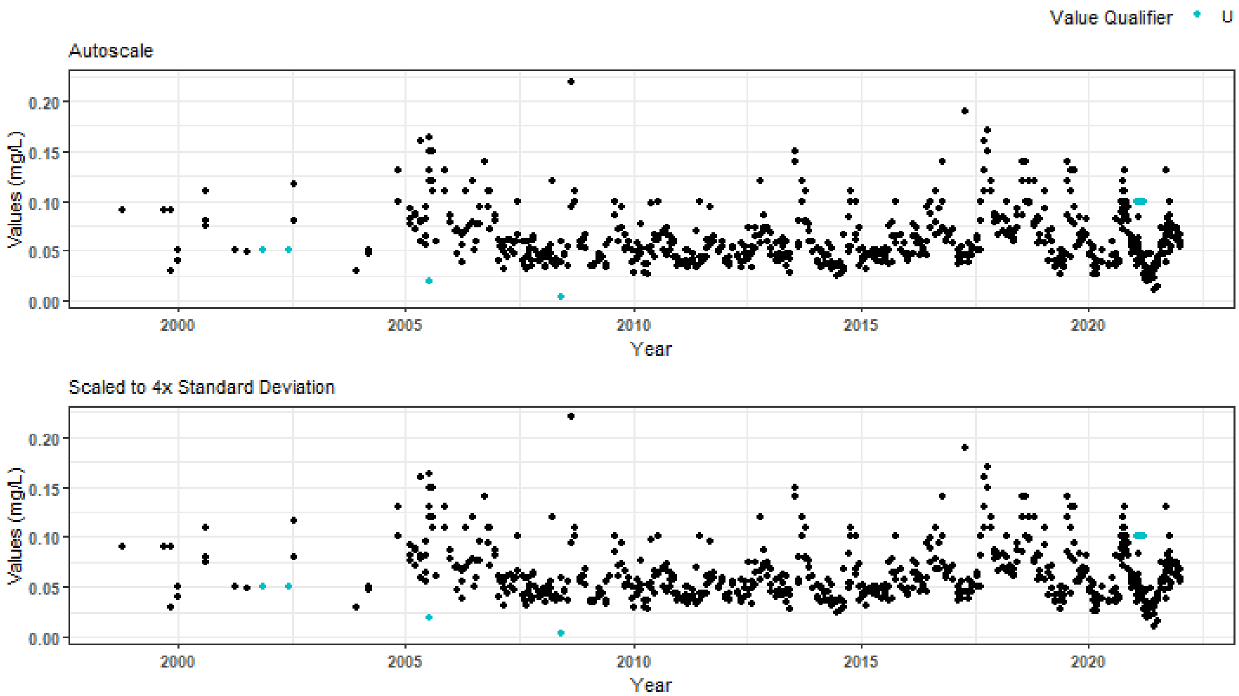
Data Points with Trendlines for Loxahatchee River-Lake Worth Creek Aquatic Preserve

Senn Slope = -0.0000105, Senn Intercept = -0.165116071428572
Trend = 0, tau = -0.0028, p = 0.1632
Linear Trendline: y = -0.000151271235709536x + 0.349391914400247



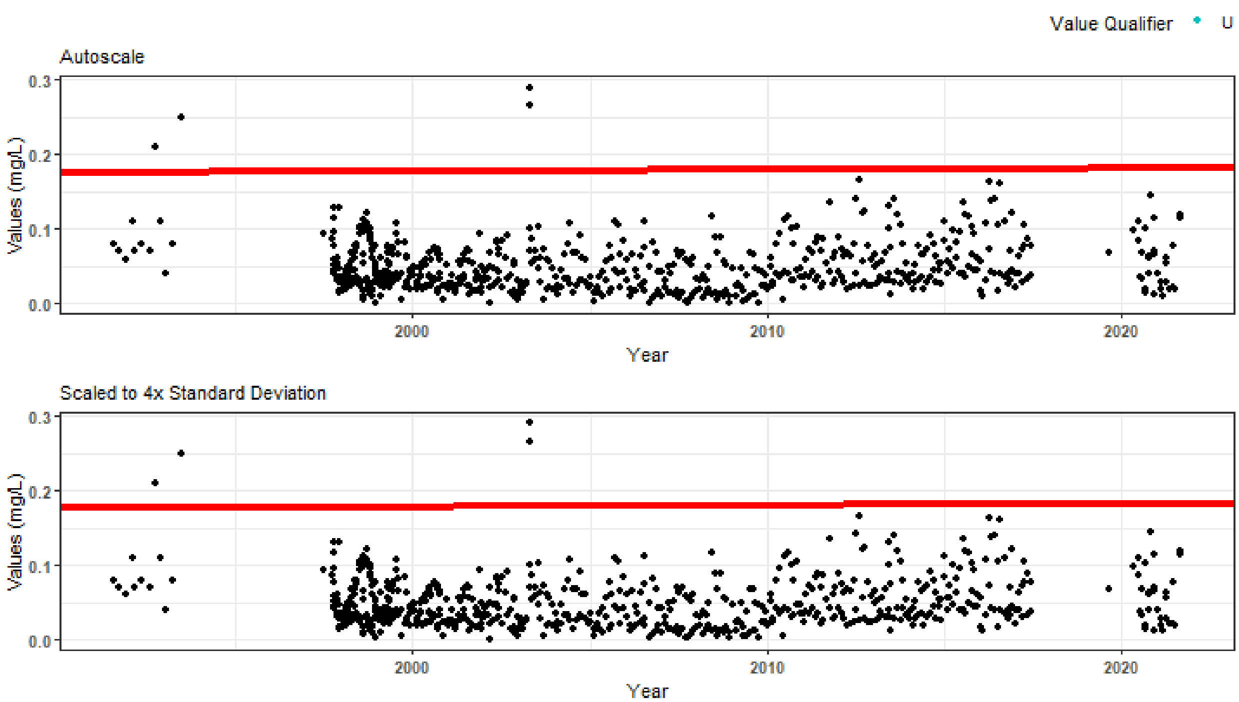
Data Points with Trendlines for Matlacha Pass Aquatic Preserve

Senn Slope = 0,   Senn Intercept = 0.558590909090909
Trend = 0,   tau = -0.0129,   p = 0.8637
Linear Trendline: y = -0.0000844442690101868x + 0.233675290509491



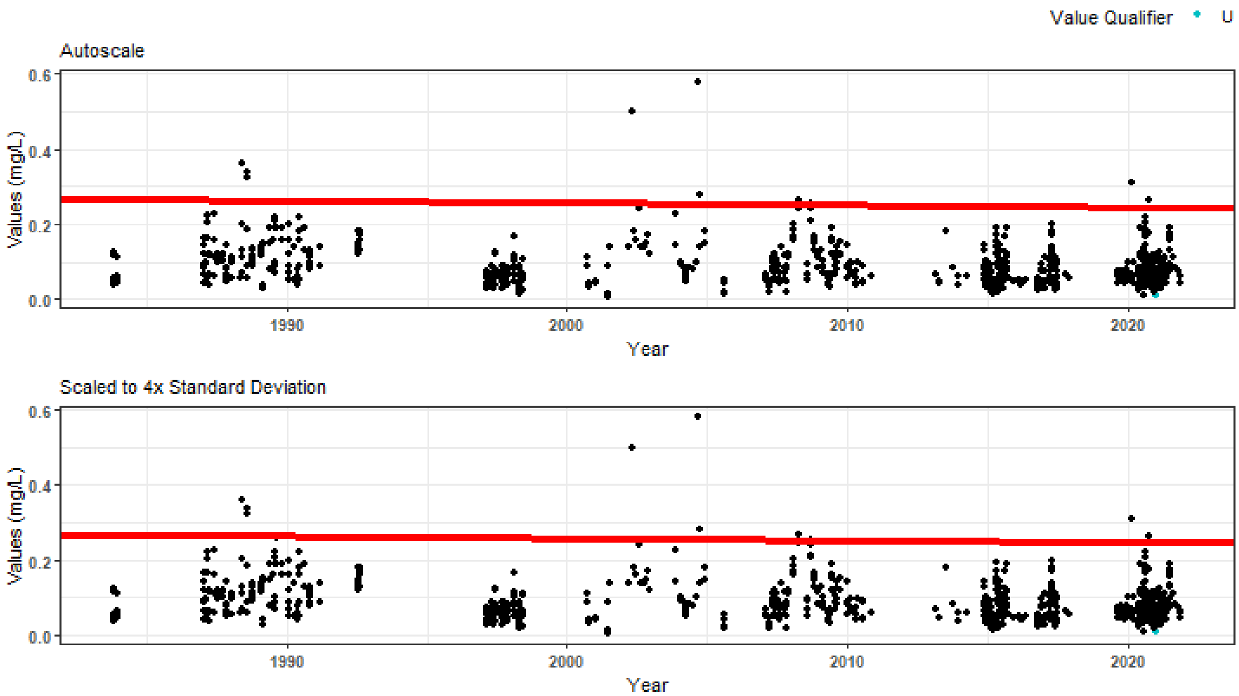
Data Points with Trendlines for Mosquito Lagoon Aquatic Preserve

Senn Slope = 0.000202105263157895,   Senn Intercept = -0.225527980889725
Trend = 0,   tau = 0.0387,   p = 0.134
Linear Trendline: y = 0.000468982840005037x + -0.891610036218806



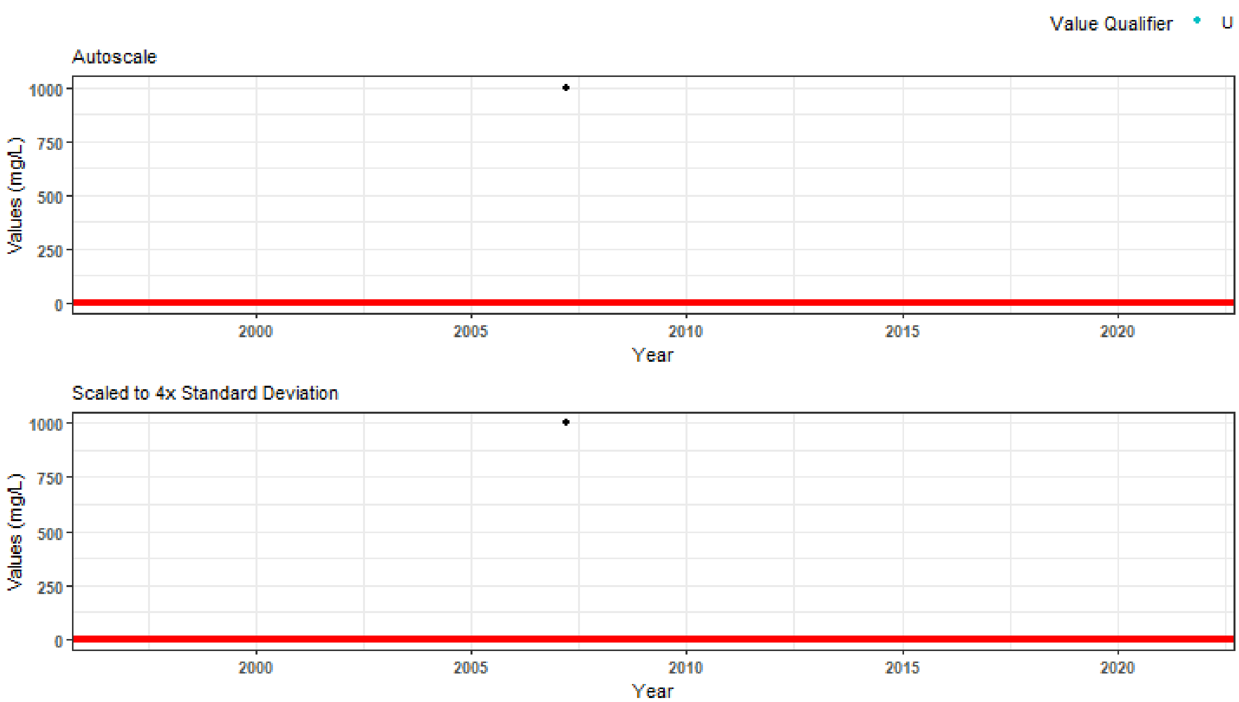
Data Points with Trendlines for Nassau River-St. Johns River Marshes Aquatic Preserve

Senn Slope = -0.00058125, Senn Intercept = 1.42003409090909
Trend = -1, tau = -0.1205, p = 0
Linear Trendline: y = -0.000879070010731271x + 1.85295319323624



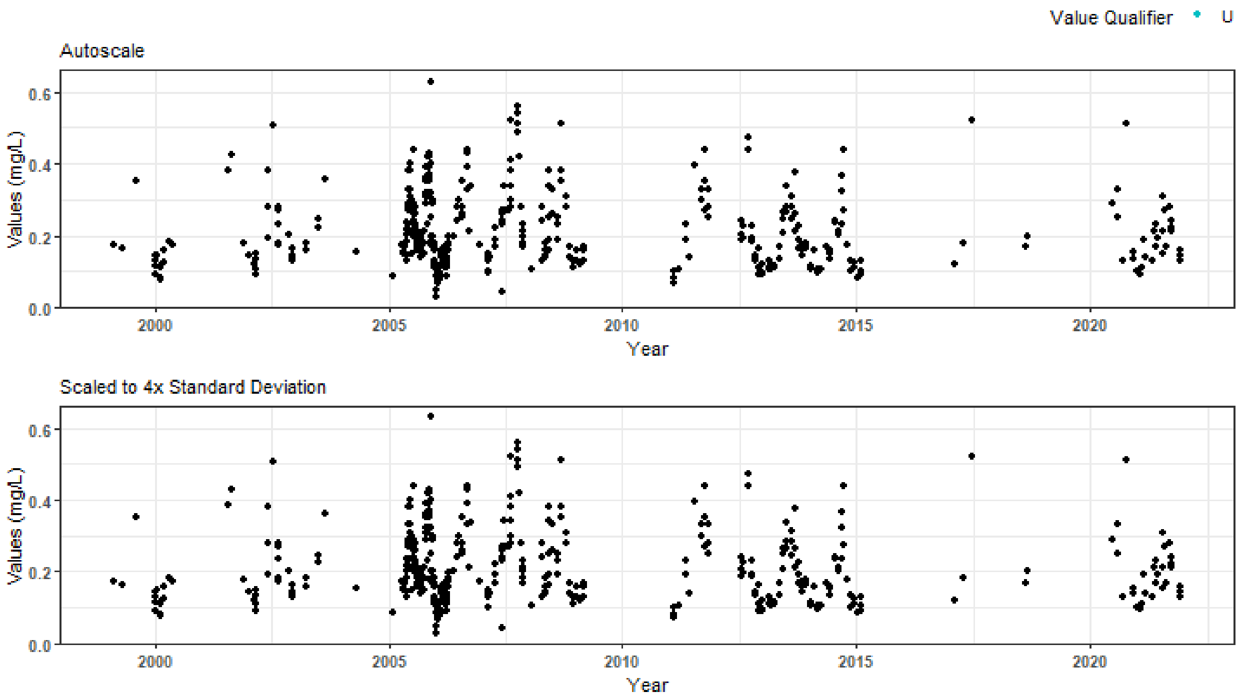
Data Points with Trendlines for Nature Coast Aquatic Preserve

Senn Slope = -0.000125, Senn Intercept = 0.25946875
Trend = -1, tau = -0.1016, p = 0
Linear Trendline: y = 0.0176982909927329x + -35.2395580859207



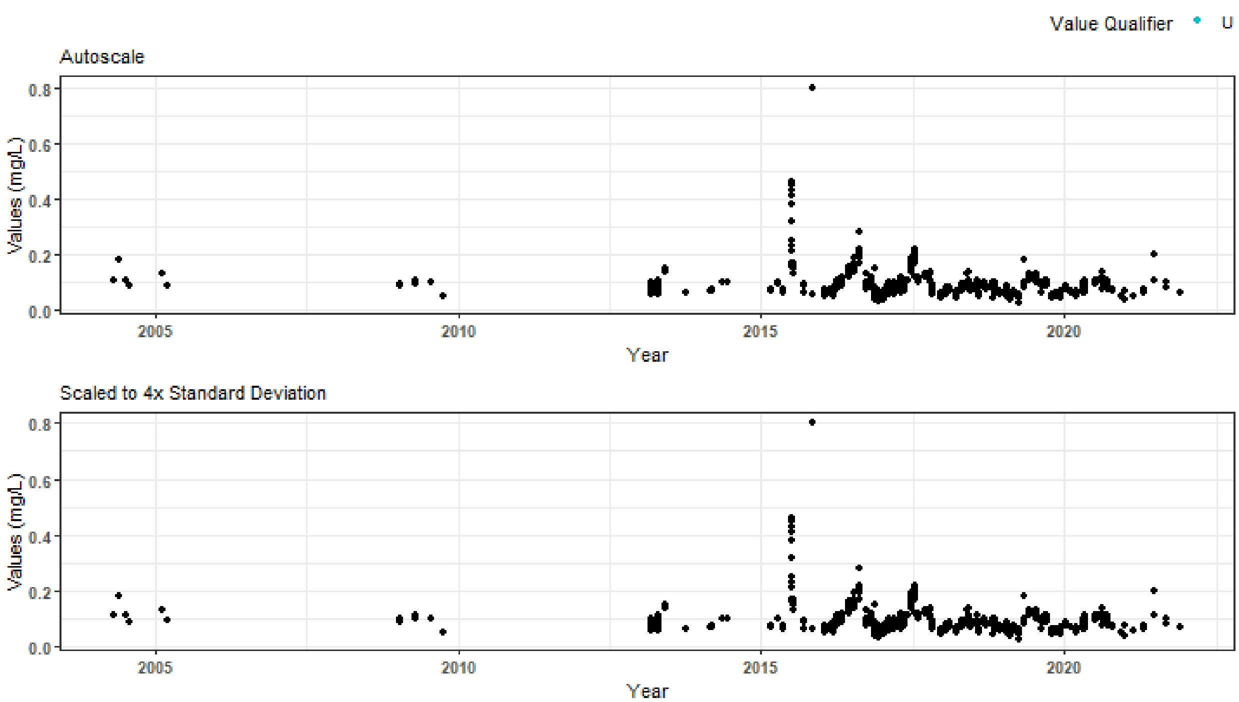
Data Points with Trendlines for North Fork St. Lucie Aquatic Preserve

Senn Slope = -0.00175,   Senn Intercept = 3.23631904761905
Trend = -1,   tau = -0.111,   p = 0.0016
Linear Trendline: y = -0.000811600450080053x + 1.84340608576465



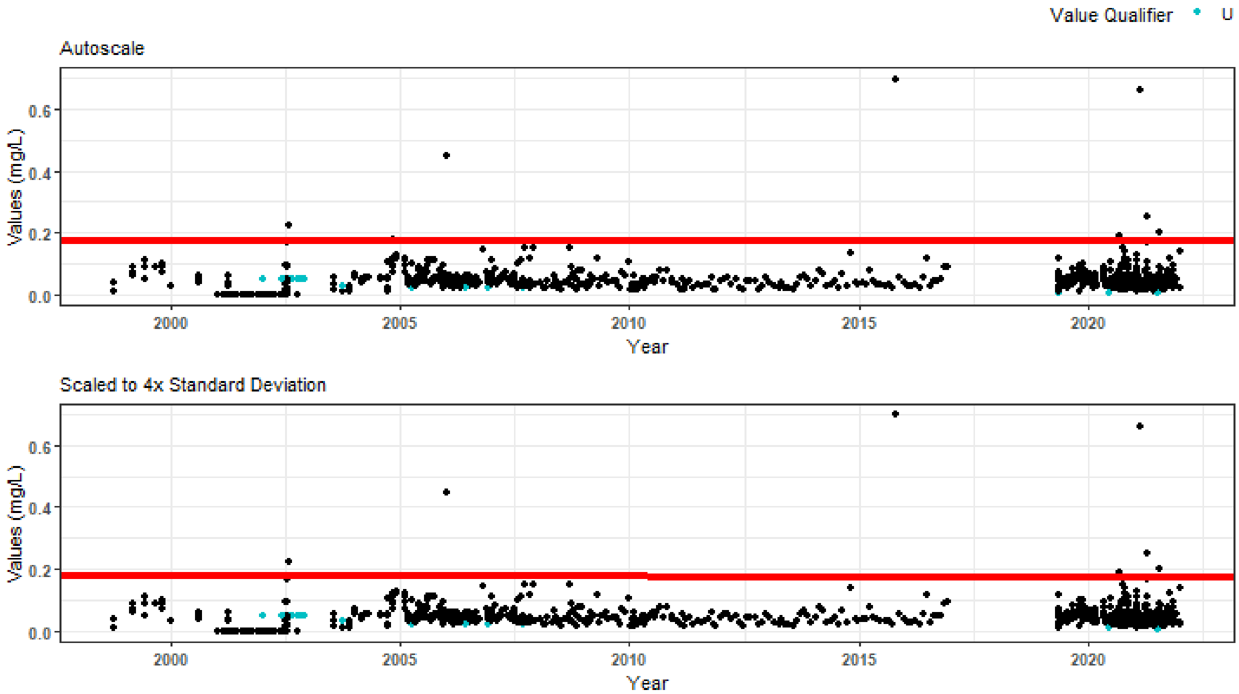
Data Points with Trendlines for Pellicer Creek Aquatic Preserve

Senn Slope = -0.00233333333333334,   Senn Intercept = 6.12775000000001
Trend = -1,   tau = -0.1472,   p = 0
Linear Trendline: y = -0.00490462496663977x + 9.99040737076144



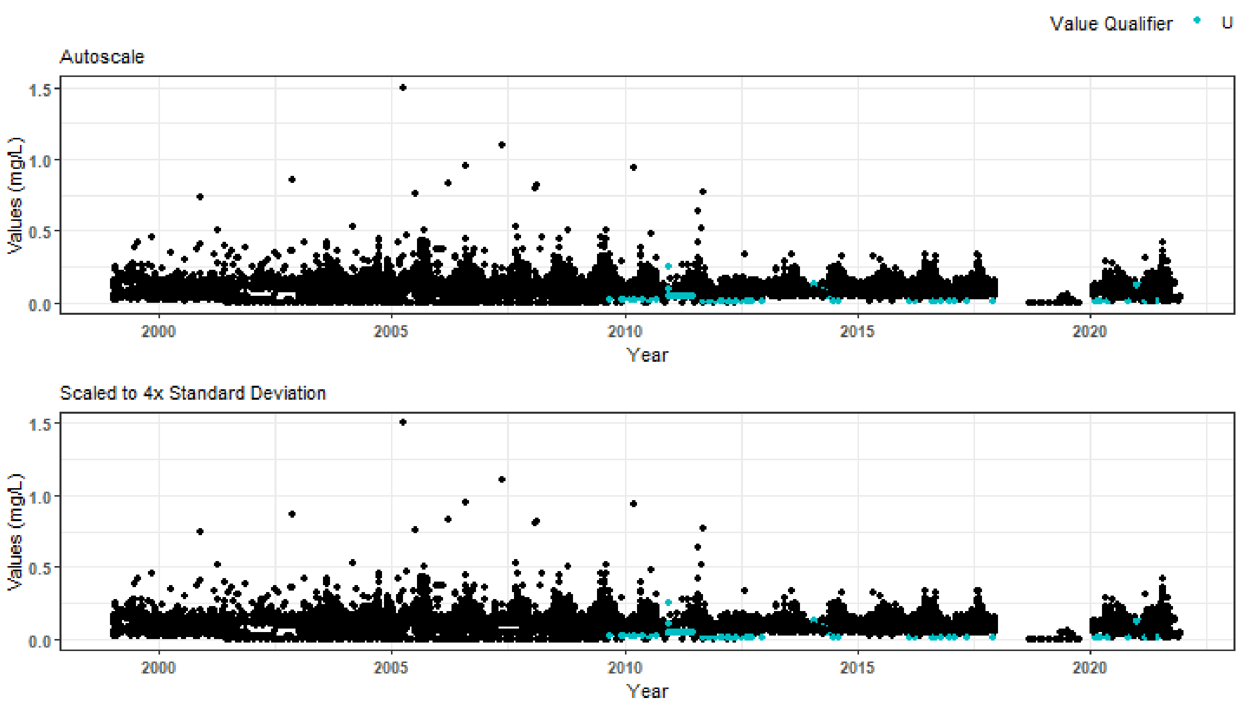
Data Points with Trendlines for Pine Island Sound Aquatic Preserve

Senn Slope = -0.0002, Senn Intercept = 0.580257575757576
Trend = -1, tau = -0.0378, p = 0.0308
Linear Trendline: y = 0.000152640265798155x + -0.262238367661858



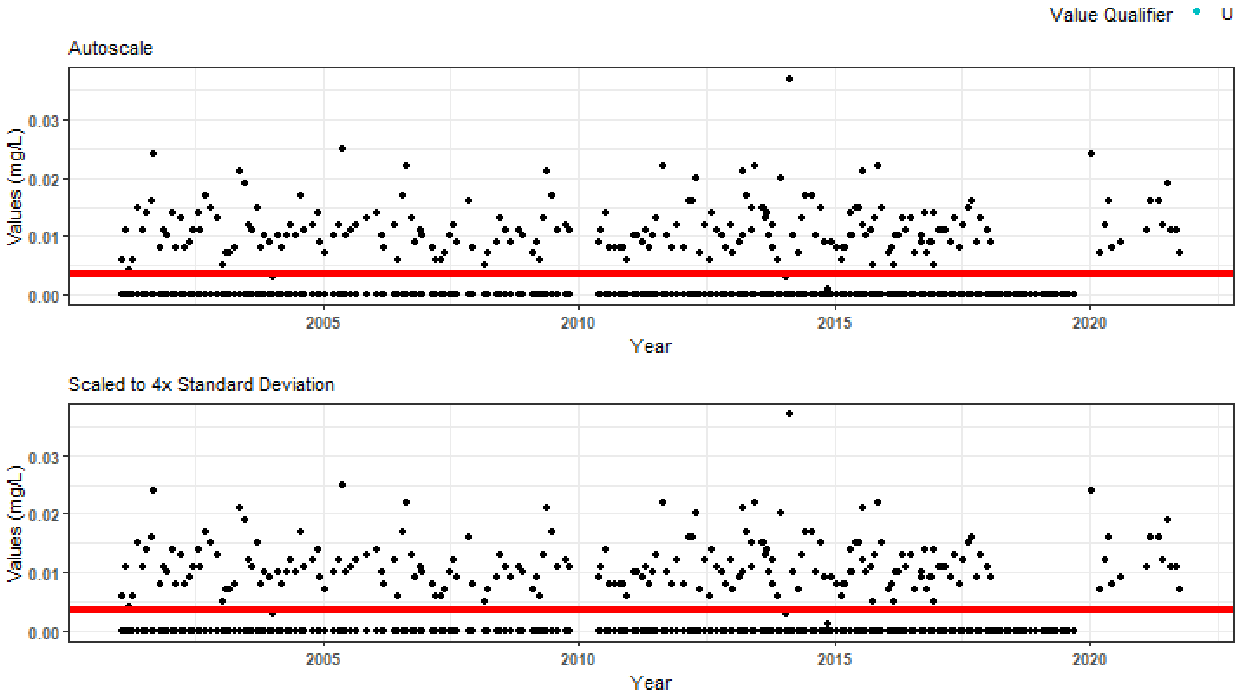
Data Points with Trendlines for Pinellas County Aquatic Preserve

Senn Slope = -0.000857142857142858, Senn Intercept = 1.41121328671329
Trend = -1, tau = -0.0587, p = 0
Linear Trendline: y = -0.00110856131814632x + 2.32880863627535



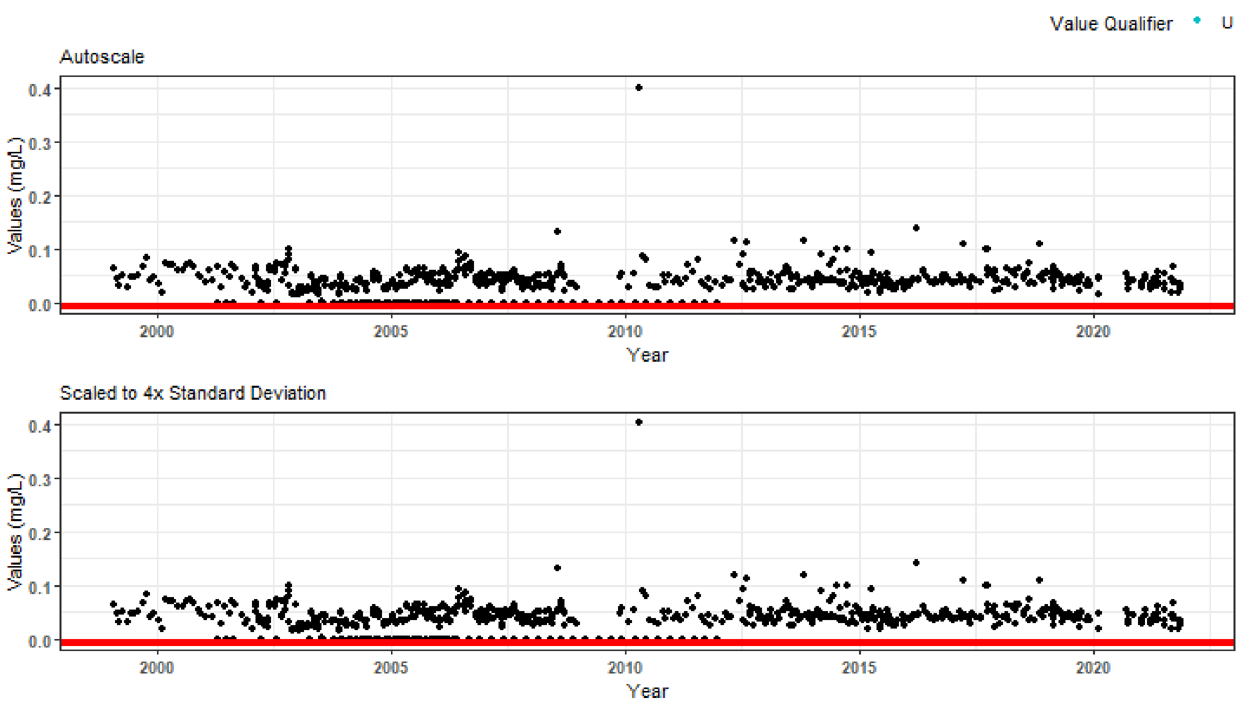
Data Points with Trendlines for Rocky Bayou State Park Aquatic Preserve

Senn Slope = 0.000000909090909090908,   Senn Intercept = 0.00343125
Trend = 0,   tau = 0.0453,   p = 0.1777
Linear Trendline: y = 0.0000410562204289646x + -0.0767363726072159



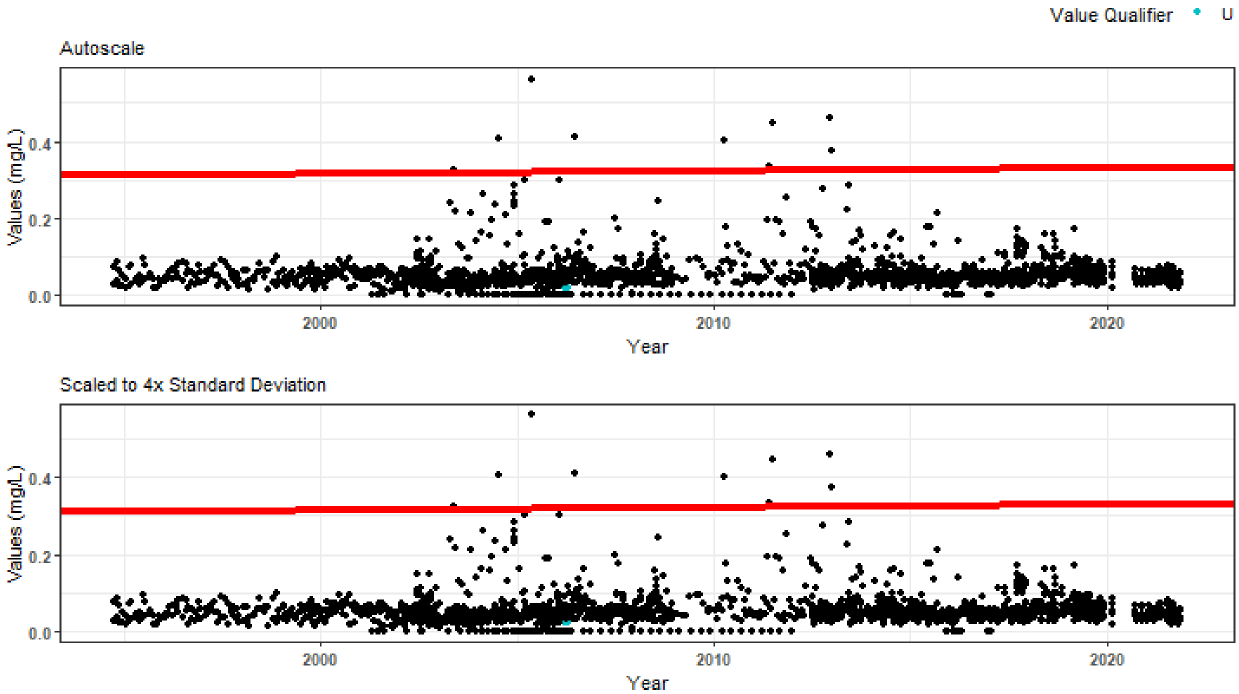
Data Points with Trendlines for Rookery Bay Aquatic Preserve

Senn Slope = 0.0000081,   Senn Intercept = -0.023356
Trend = 1,   tau = 0.1281,   p = 0
Linear Trendline: y = 0.00138164230224144x + -2.7501868155804



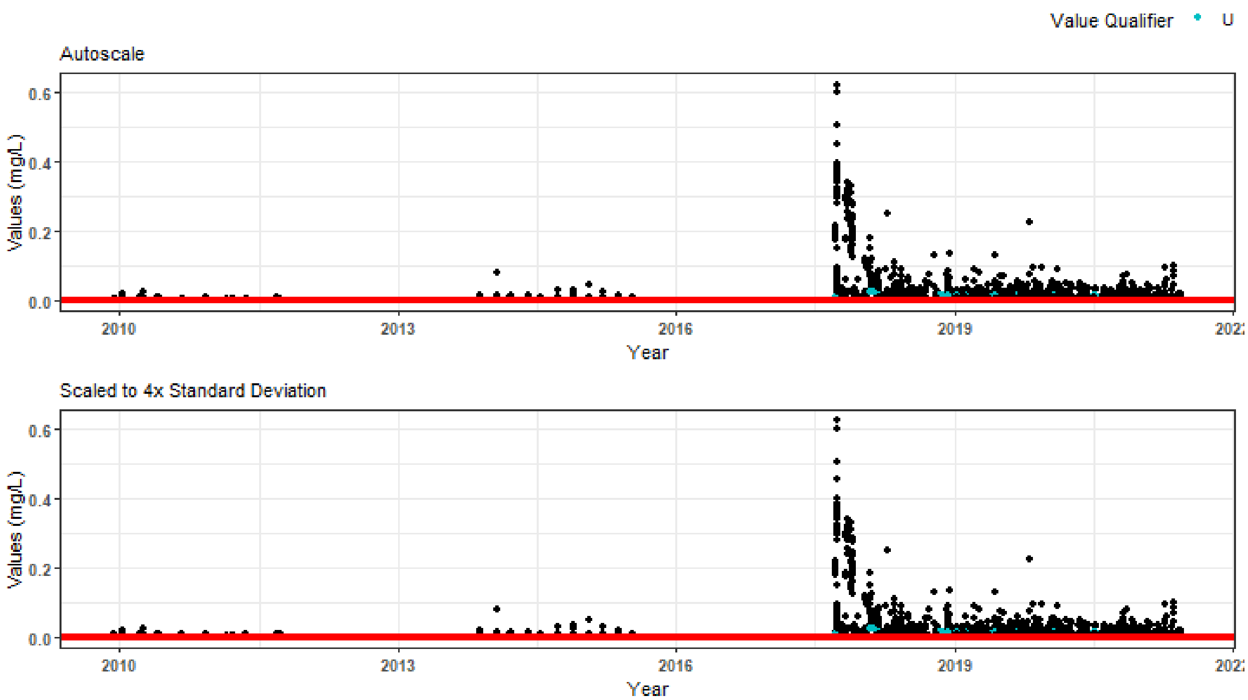
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve

Senn Slope = 0.000752086956521739, Senn Intercept = -1.18879318181818
Trend = 1, tau = 0.1403, p = 0
Linear Trendline: y = 0.000913596187679696x + -1.79249597932887



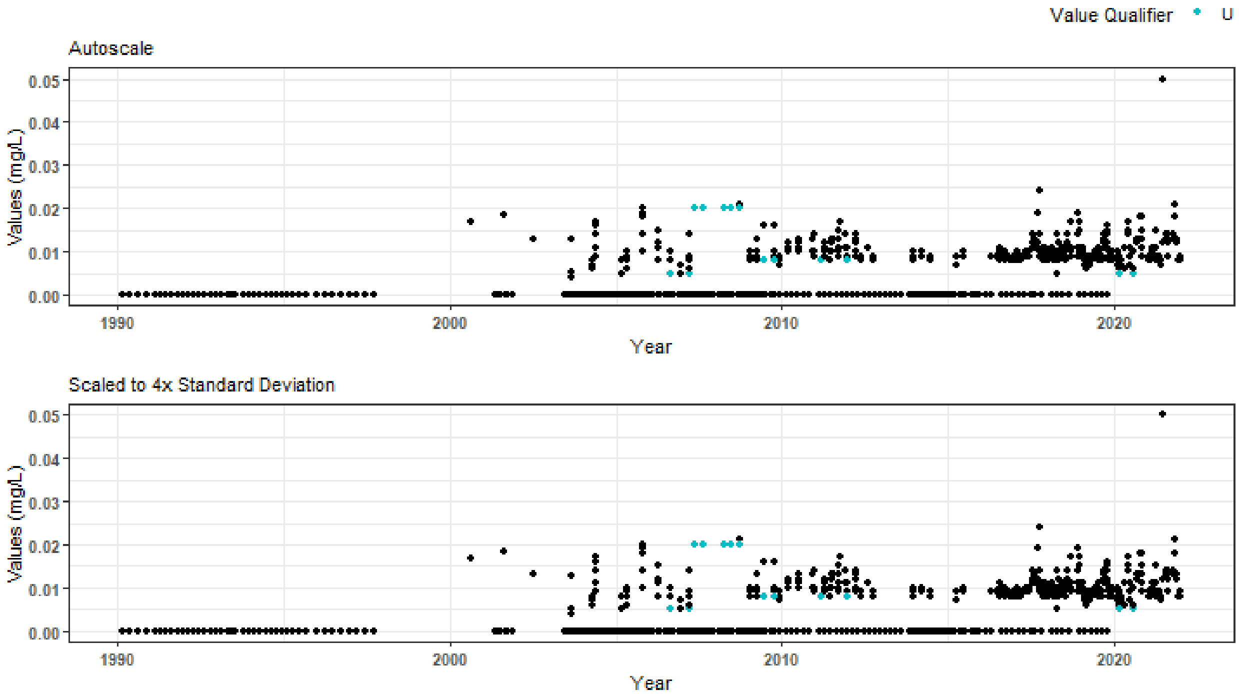
Data Points with Trendlines for Southeast Florida Coral Reef Ecosystem Conservation Area

Senn Slope = 0, Senn Intercept = 0.001
Trend = 0, tau = -0.0583, p = 0
Linear Trendline: y = -0.00261099074539683x + 5.28337639808238



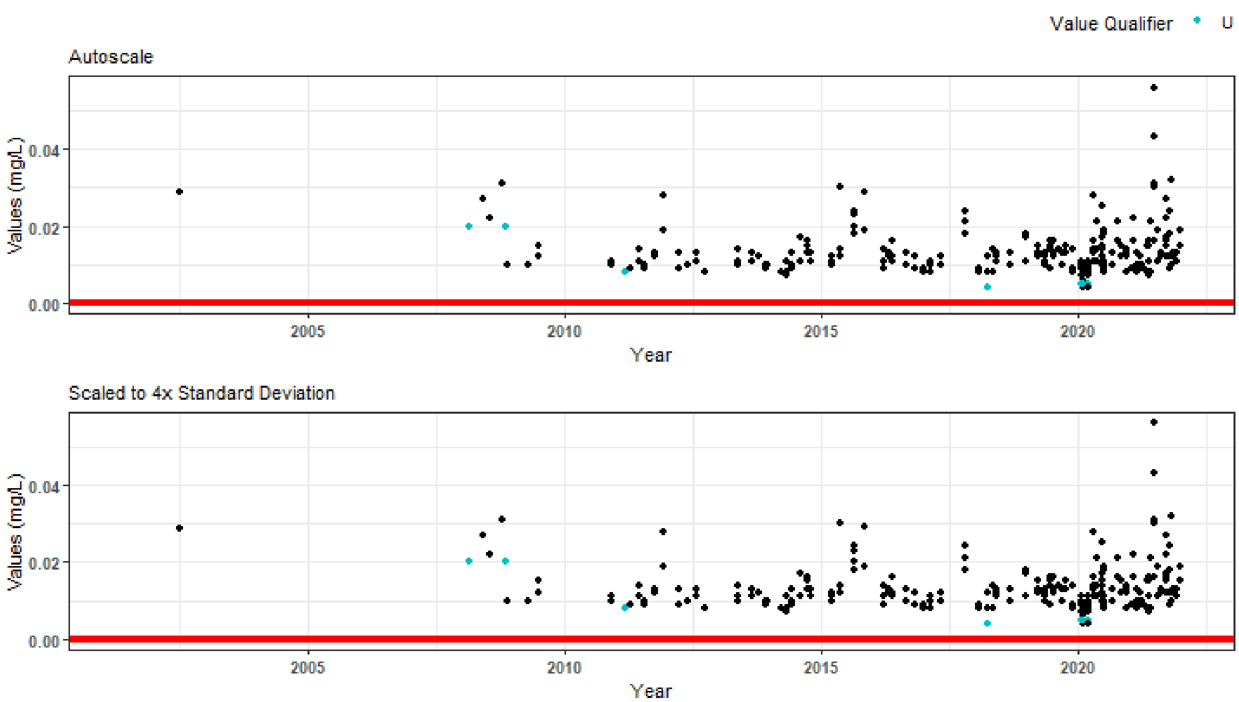
Data Points with Trendlines for St. Andrews State Park Aquatic Preserve

Senn Slope = 0.000245336822660099, Senn Intercept = -0.583565865384615
Trend = 1, tau = 0.3213, p = 0
Linear Trendline: y = 0.000342543785068996x + -0.683047177623877



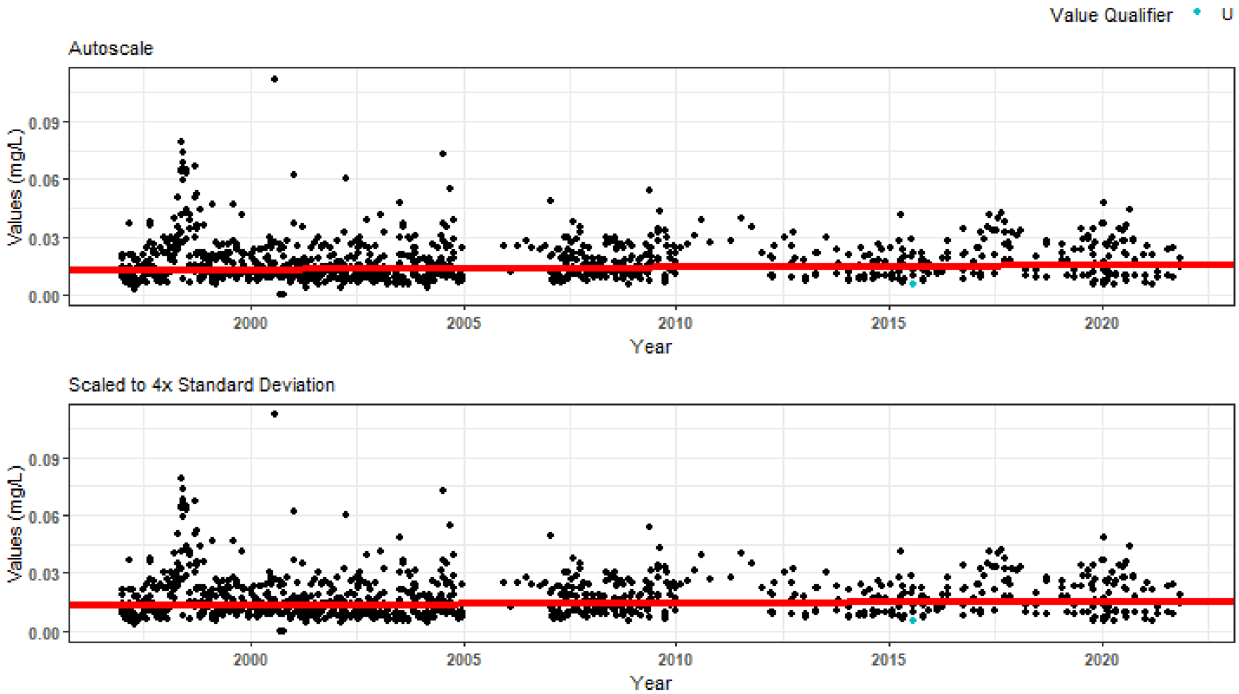
Data Points with Trendlines for St. Joseph Bay Aquatic Preserve

Senn Slope = 0.00000130769230769231, Senn Intercept = -0.0026018
Trend = 1, tau = 0.3574, p = 0
Linear Trendline: y = 0.000470959071655565x + -0.944238817834334



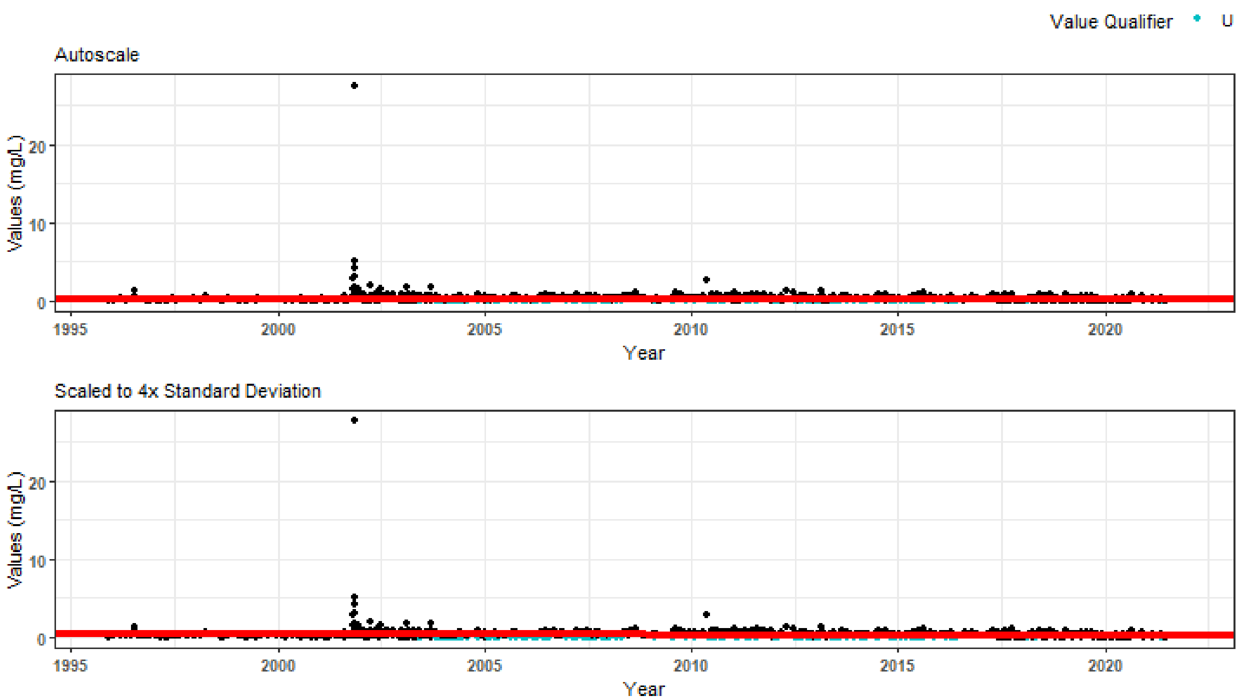
Data Points with Trendlines for St. Martins Marsh Aquatic Preserve

Senn Slope = 0.0001, Senn Intercept = -0.1866
Trend = 1, tau = 0.0605, p = 0.0047
Linear Trendline: y = 0.0000604529762928757x + -0.103786326274541



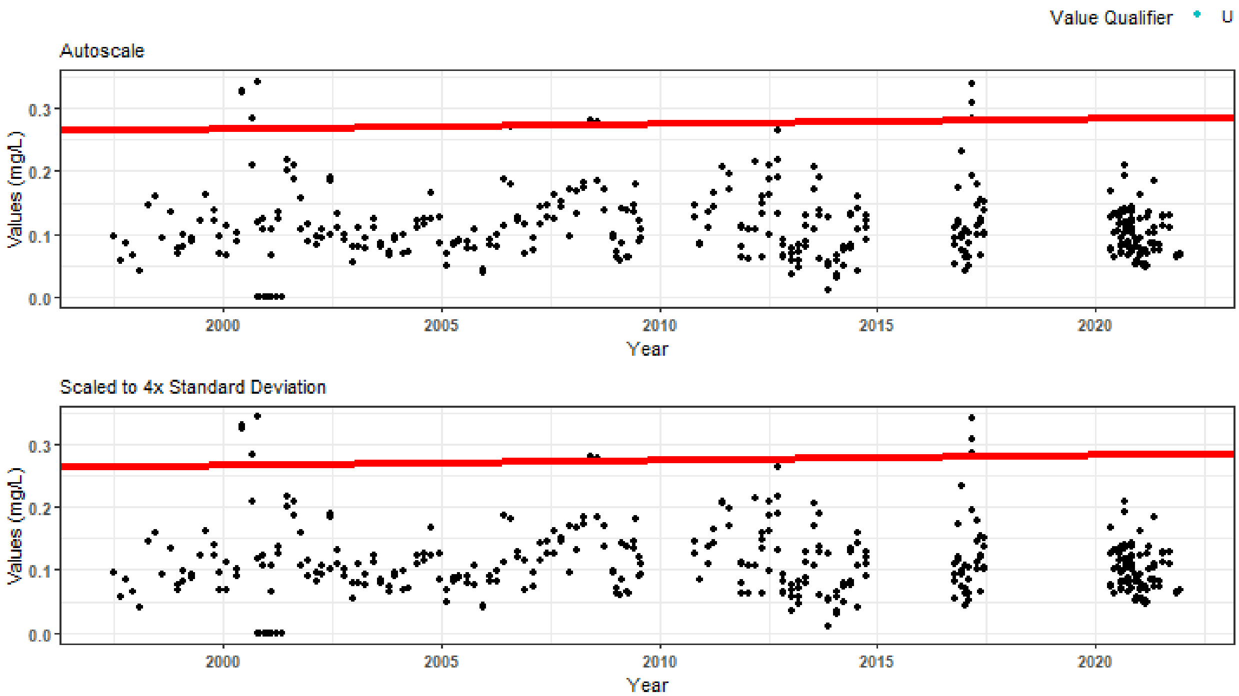
Data Points with Trendlines for Terra Ceia Aquatic Preserve

Senn Slope = -0.003375, Senn Intercept = 7.1575
Trend = -1, tau = -0.1787, p = 0
Linear Trendline: y = -0.00529972161396573x + 10.8566245965828



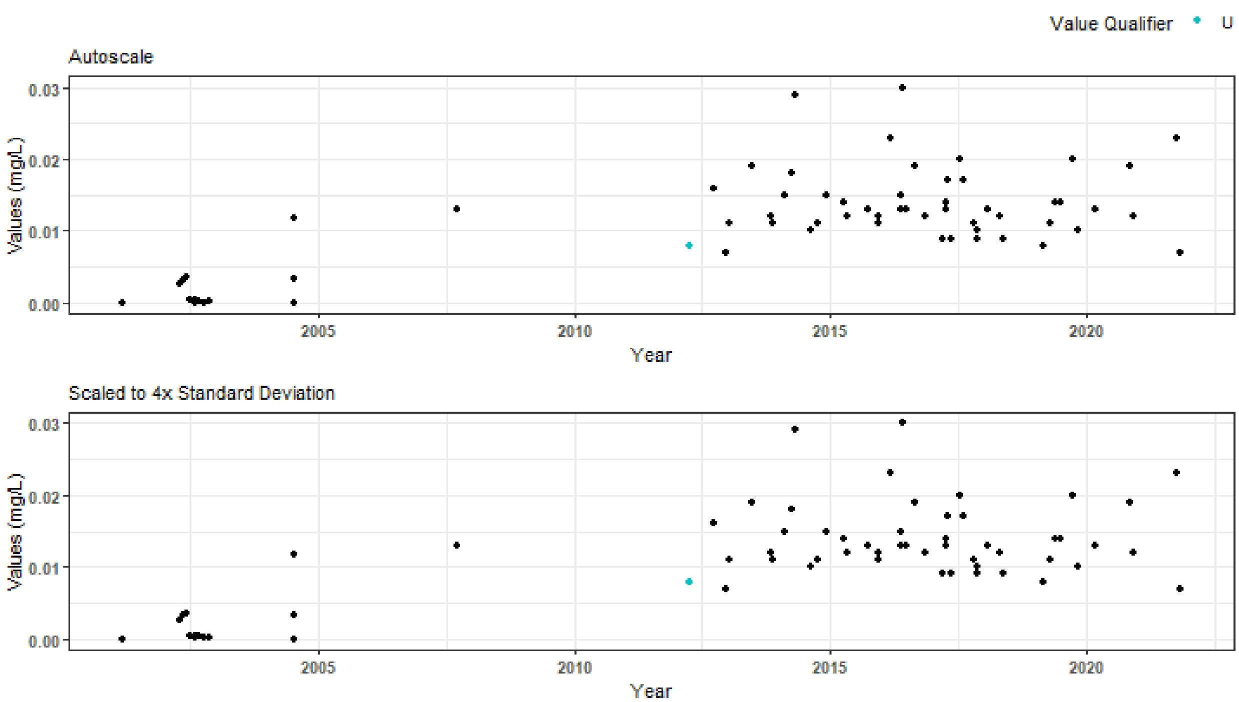
Data Points with Trendlines for Tomoka Marsh Aquatic Preserve

Senn Slope = 0.000811111111111111,   Senn Intercept = -1.355421875
Trend = 1,   tau = 0.0794,   p = 0.0326
Linear Trendline: y = 0.000956261779558498x + -1.81883238260791



Data Points with Trendlines for Yellow River Marsh Aquatic Preserve

Senn Slope = 0.000497058823529412,   Senn Intercept = -0.8064
Trend = 0,   tau = 0.2749,   p = 0.0698
Linear Trendline: y = 0.000754529133563516x + -1.50808071066374



Appendix V: Managed Area Summary Box Plots

Data is taken and grouped by `ManagedAreaName`. The scripts that create plots follow this format

1. Use the data set that only has `Use_In_Analysis` of `TRUE` for the desired managed area
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Determine the minimum, mean, and standard deviation for the data to be used for y-axis scales
    - Excludes the top 2% of values to reduce the impact of extreme outliers on the y-axis scale
4. Set what values are to be used for the x-axis, y-axis, and the variable that should determine groups for the box plots
5. Set the plot type as a box plot with the size of the outlier points
6. Create the title, x-axis, y-axis, and color fill labels
7. Set the y and x limits
8. Make the axis labels bold
9. Plot the arrangement as a set of panels

The following plots are arranged by `ManagedAreaName` with data grouped by `Year`, then `Year` and `Month`, then finally `Month` only. Each managed area will have 3 sets of plots, each with 3 panels in them. Each panel goes as follows:

1. Y-axis autoscaled
2. Y-axis set to be mean + 4 times the standard deviation
3. Y-axis set to be mean + 4 times the standard deviation for most recent 10 years of data

```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      year_lower <- min(data$Year[data$Use_In_Analysis == TRUE &
                                  data$ManagedAreaName == MA_names[i]])
      year_upper <- max(data$Year[data$Use_In_Analysis == TRUE &
                                  data$ManagedAreaName == MA_names[i]])
      min_RV <- min(data$ResultValue[data$Use_In_Analysis == TRUE &
                                     data$ManagedAreaName == MA_names[i]])
      mn_RV <- mean(data$ResultValue[data$Use_In_Analysis == TRUE &
                                     data$ManagedAreaName == MA_names[i] &
                                     data$ResultValue <
                                     quantile(data$ResultValue, 0.98)])
      sd_RV <- sd(data$ResultValue[data$Use_In_Analysis == TRUE &
                                   data$ManagedAreaName == MA_names[i] &
                                   data$ResultValue <
                                   quantile(data$ResultValue, 0.98)])
      x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
      y_scale <- mn_RV + 4 * sd_RV

      ##Year plots
      p1 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                               data$ManagedAreaName == MA_names[i], ],
                   aes(x = Year, y = ResultValue, group = Year)) +
         geom_boxplot(outlier.size = 0.5) +
         labs(subtitle = "Autoscale",
              x = "Year", y = paste0("Values (", unit, ")")) +
```

```
    scale_x_continuous(limits = c(year_lower - 1, year_upper + 1),
                       breaks = rev(seq(year_upper,
                                        year_lower, -x_scale))) +
    theme_bw() +
    theme(axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

 p2 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                              data$ManagedAreaName == MA_names[i], ],
              aes(x = Year, y = ResultValue, group = Year)) +
    geom_boxplot(outlier.size = 0.5) +
    labs(subtitle = "Scaled to 4x Standard Deviation",
         x = "Year", y = paste0("Values (", unit, ")")) +
    ylim(min_RV, y_scale) +
    scale_x_continuous(limits = c(year_lower - 1, year_upper + 1),
                       breaks = rev(seq(year_upper,
                                        year_lower, -x_scale))) +
    theme_bw() +
    theme(axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

 p3 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                              data$ManagedAreaName == MA_names[i] &
                              data$Year>=year_upper-10, ],
              aes(x = Year, y = ResultValue, group = Year)) +
    geom_boxplot(outlier.size = 0.5) +
    labs(subtitle = "Scaled to 4x Standard Deviation, Last 10 Years",
         x = "Year", y = paste0("Values (", unit, ")")) +
    ylim(min_RV, y_scale) +
    scale_x_continuous(limits = c(year_upper - 10.5, year_upper + 1),
                       breaks = rev(seq(year_upper, year_upper - 10,-2))) +
    theme_bw() +
    theme(axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

 Yset <- ggarrange(p1, p2, p3, ncol = 1)

 p0 <- ggplot() + labs(title = paste0("Summary Box Plots for ",
                                      MA_names[i]), subtitle = "By Year") +
    theme_bw() + theme(plot.title = element_text(face="bold", hjust = 0.5),
                       panel.border = element_blank(),
                       panel.grid.major = element_blank(),
                       panel.grid.minor = element_blank(), axis.line = element_blank())


 ## Year & Month Plots
 p4 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                              data$ManagedAreaName == MA_names[i], ],
              aes(x = YearMonthDec, y = ResultValue,
                  group = YearMonth, color = as.factor(Month))) +
    geom_boxplot(outlier.size = 0.5) +
    labs(subtitle = "Autoscale",
         x = "Year", y = paste0("Values (", unit, ")"), color = "Month") +
```

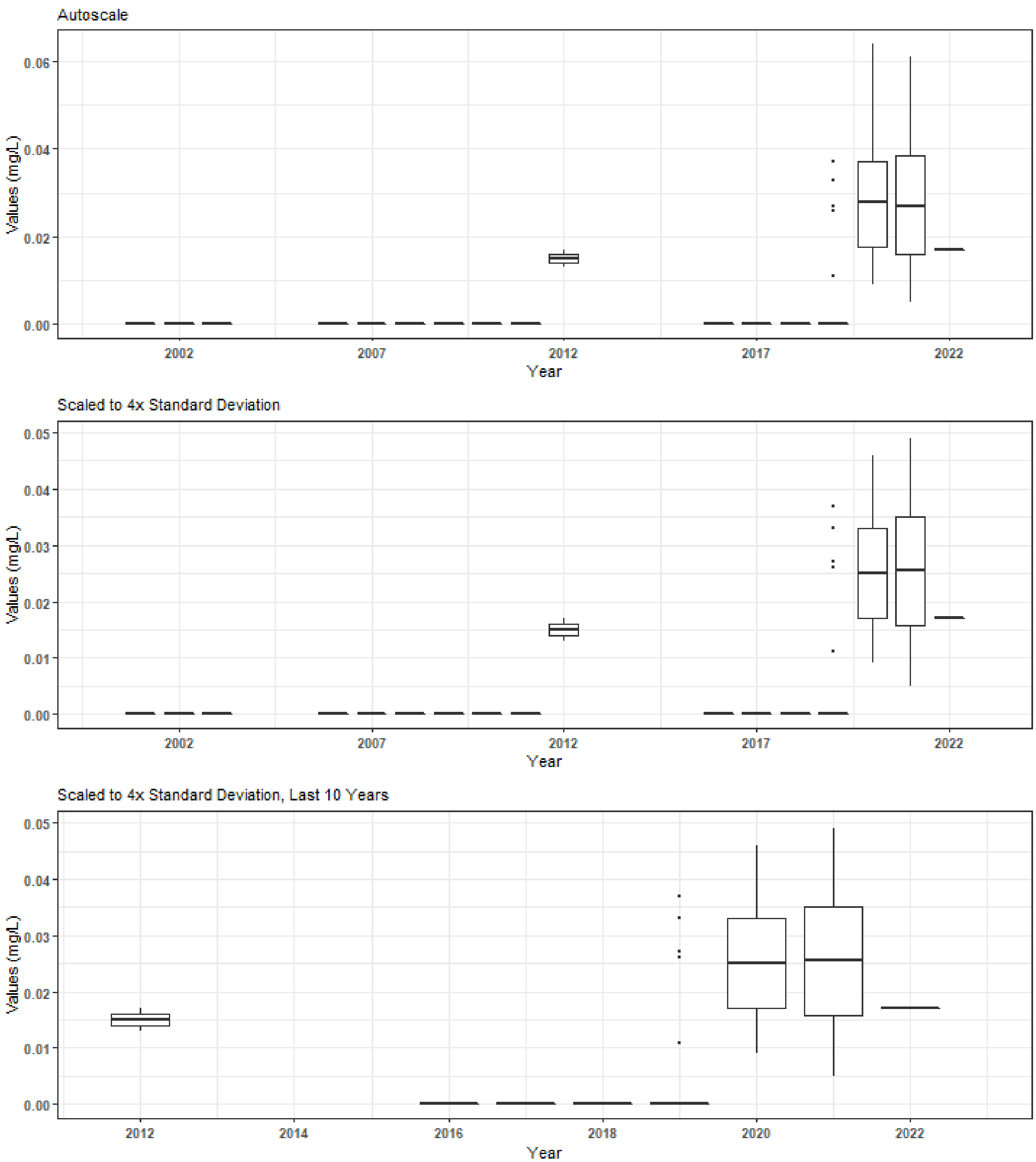
```
    scale_x_continuous(limits = c(year_lower - 1, year_upper + 1),
                       breaks = rev(seq(year_upper,
                                        year_lower, -x_scale))) +
    theme_bw() +
    theme(legend.position = "none",
          axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

p5 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                          data$ManagedAreaName == MA_names[i], ],
             aes(x = YearMonthDec, y = ResultValue,
                 group = YearMonth, color = as.factor(Month))) +
    geom_boxplot(outlier.size = 0.5) +
    labs(subtitle = "Scaled to 4x Standard Deviation",
         x = "Year", y = paste0("Values (", unit, ")"), color = "Month") +
    ylim(min_RV, y_scale) +
    scale_x_continuous(limits = c(year_lower - 1, year_upper + 1),
                       breaks = rev(seq(year_upper,
                                        year_lower, -x_scale))) +
    theme_bw() +
    theme(legend.position = "top", legend.box = "horizontal",
          axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold")) +
    guides(color = guide_legend(nrow = 1))

p6 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                          data$ManagedAreaName == MA_names[i], ],
             aes(x = YearMonthDec, y = ResultValue,
                 group = YearMonth, color = as.factor(Month)
             )) +
    geom_boxplot(outlier.size = 0.5) +
    labs(subtitle = "Scaled to 4x Standard Deviation, Last 10 Years",
         x = "Year", y = paste0("Values (", unit, ")"), color = "Month") +
    ylim(min_RV, y_scale) +
    scale_x_continuous(limits = c(year_upper - 10.5, year_upper + 1),
                       breaks = rev(seq(year_upper, year_upper - 10,-2))) +
    theme_bw() +
    theme(legend.position = "none",
          axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

leg1 <- get_legend(p5)
YMset <- ggarrange(leg1, p4, p5 + theme(legend.position = "none"), p6,
                   ncol = 1, heights = c(0.1, 1, 1, 1))

p00 <- ggplot() + labs(title = paste0("Summary Box Plots for ",
                                      MA_names[i]),
                       subtitle = "By Year & Month") + theme_bw() +
    theme(plot.title = element_text(face="bold", hjust = 0.5),
          panel.border = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(), axis.line = element_blank())
```

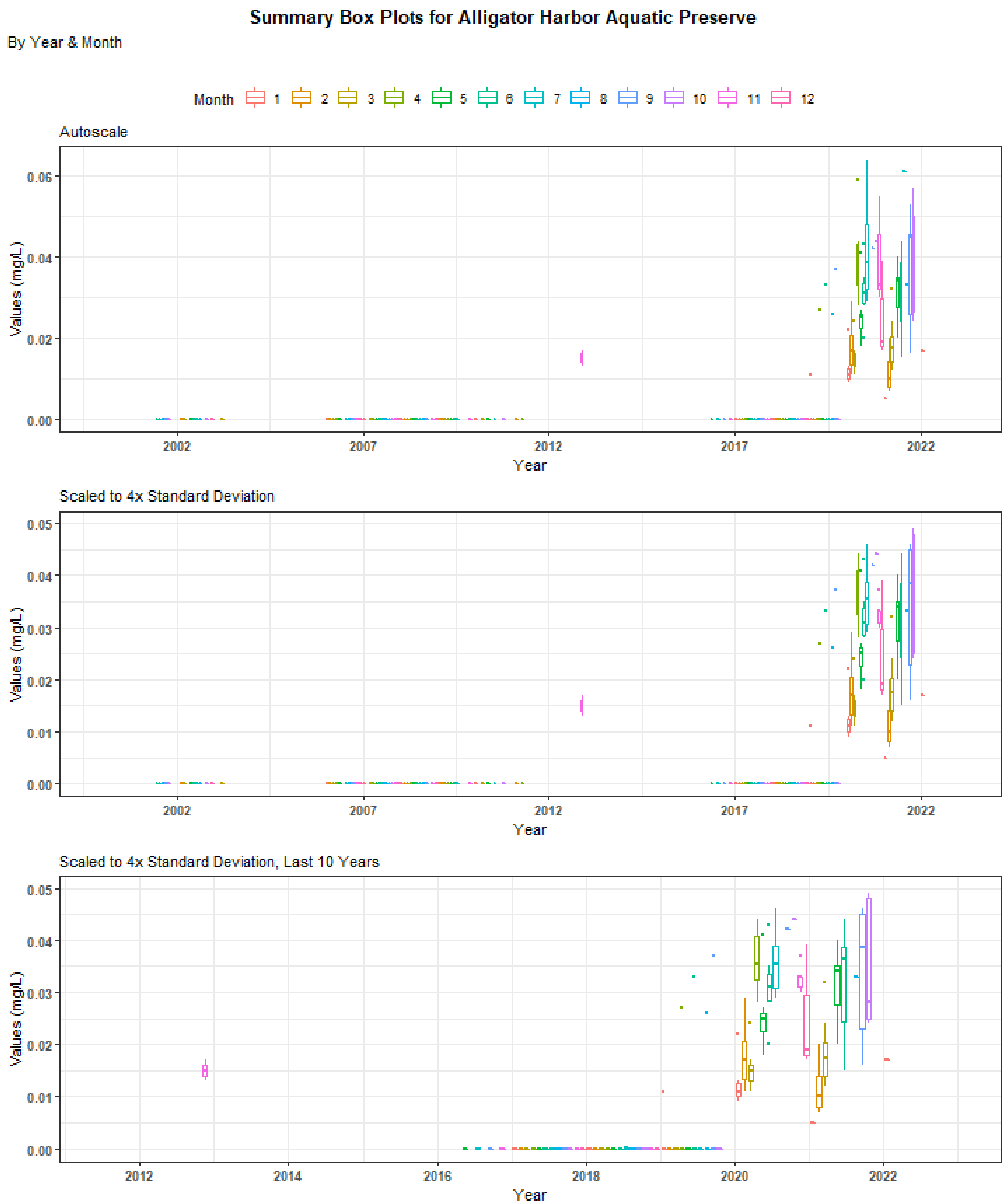
```r
## Month Plots
p7 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                            data$ManagedAreaName == MA_names[i], ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Autoscale",
        x = "Month", y = paste0("Values (", unit, ")"), fill = "Month") +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "none",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

p8 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                            data$ManagedAreaName == MA_names[i], ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation",
        x = "Month", y = paste0("Values (", unit, ")"), fill = "Month") +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "top", legend.box = "horizontal",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold")) +
   guides(fill = guide_legend(nrow = 1))

p9 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                            data$ManagedAreaName == MA_names[i] &
                            data$Year >= year_upper - 10, ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Scaled to 4x Standard Deviation, Last 10 Years",
        x = "Month", y = paste0("Values (", unit, ")"), fill = "Month") +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "none",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

leg2 <- get_legend(p8)
Mset <- ggarrange(leg2, p7, p8 + theme(legend.position = "none"), p9,
                  ncol = 1, heights = c(0.1, 1, 1, 1))

p000 <- ggplot() + labs(title = paste0("Summary Box Plots for ",
                                       MA_names[i]),
                        subtitle = "By Month") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust = 0.5),
         panel.border = element_blank(),
```

```
              panel.grid.major = element_blank(),
              panel.grid.minor = element_blank(), axis.line = element_blank())

      print(ggarrange(p0, Yset, ncol = 1, heights = c(0.07, 1)))
      print(ggarrange(p00, YMset, ncol = 1, heights = c(0.07, 1)))
      print(ggarrange(p000, Mset, ncol = 1, heights = c(0.07, 1, 0.7)))
   }
}
```

Summary Box Plots for Alligator Harbor Aquatic Preserve

By Year

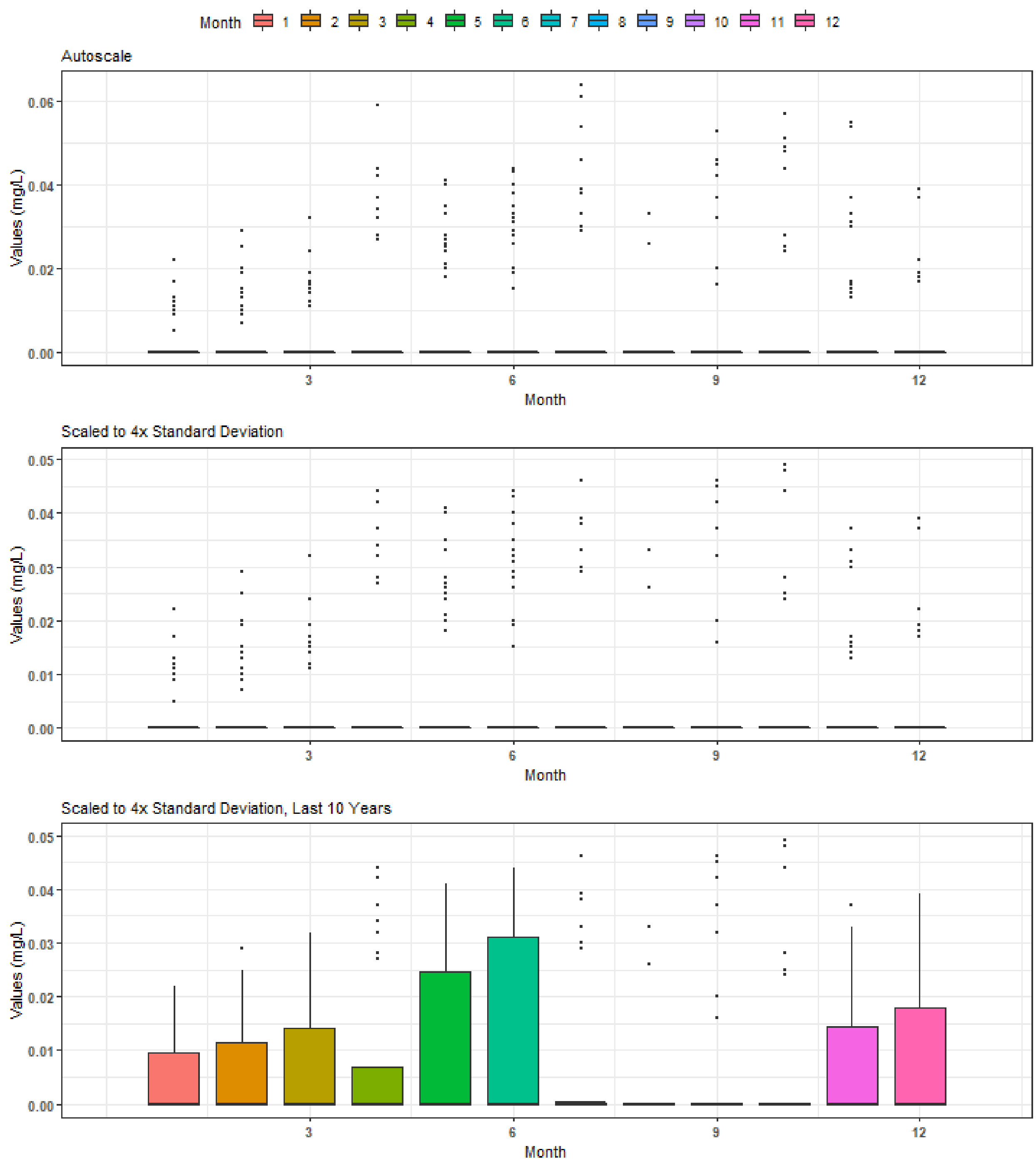




Summary Box Plots for Alligator Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.00
0.02
0.04
0.06
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.01
0.02
0.03
0.04
0.05
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.01
0.02
0.03
0.04
0.05
2012
2014
2016
2018
2020
2022
Year

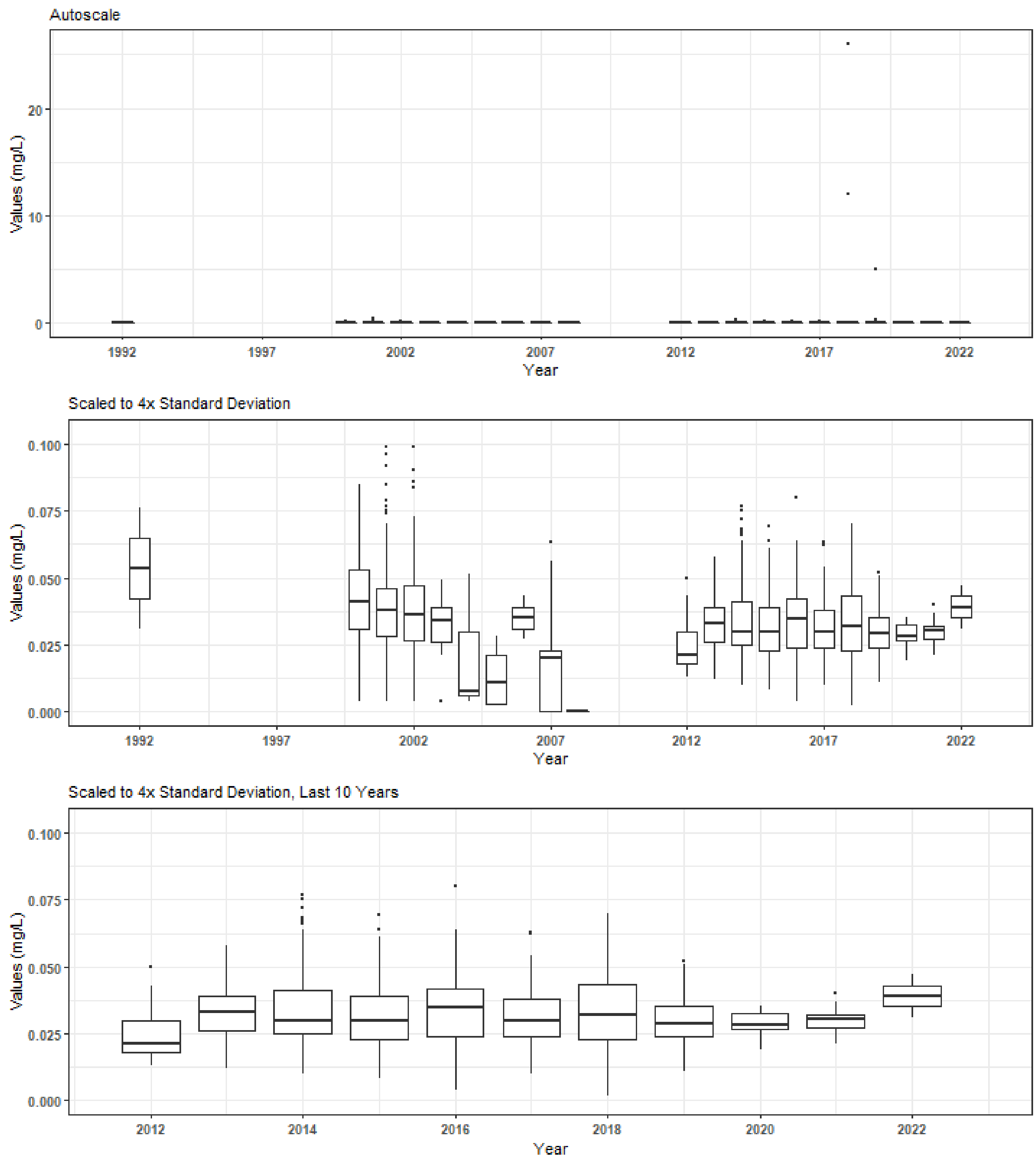
Summary Box Plots for Alligator Harbor Aquatic Preserve

By Month



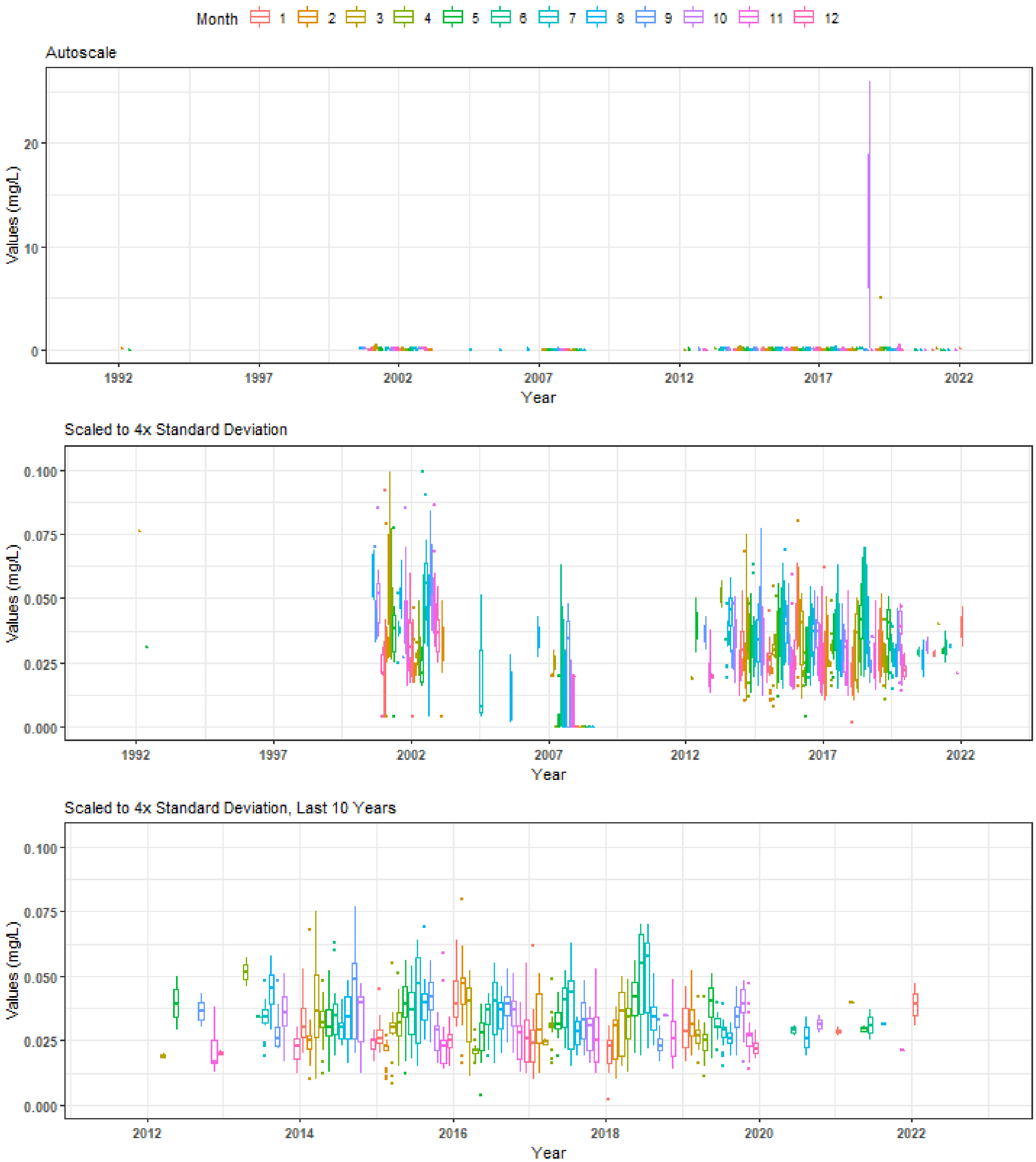
Summary Box Plots for Apalachicola Bay Aquatic Preserve

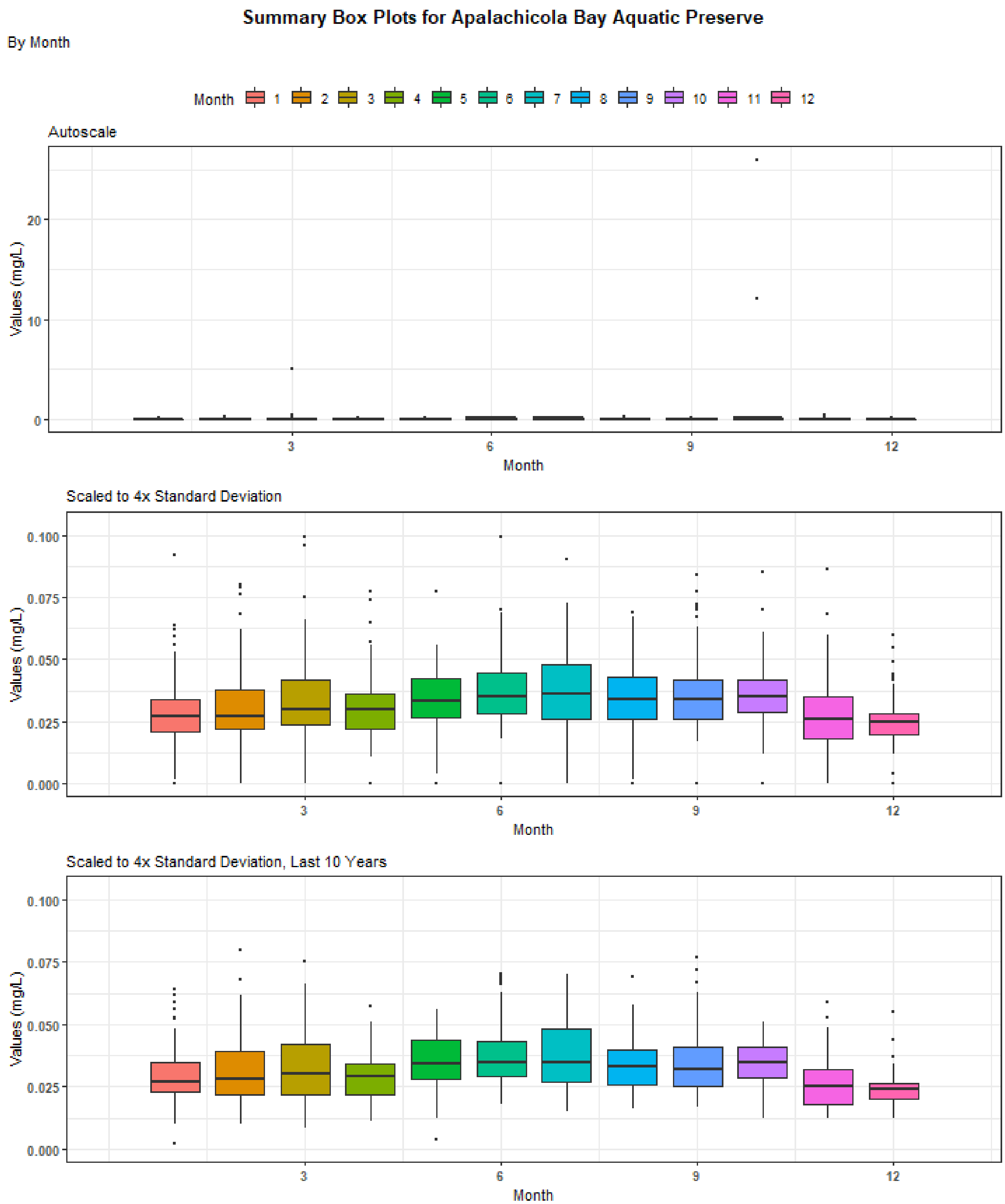
By Year



Summary Box Plots for Apalachicola Bay Aquatic Preserve

By Year & Month

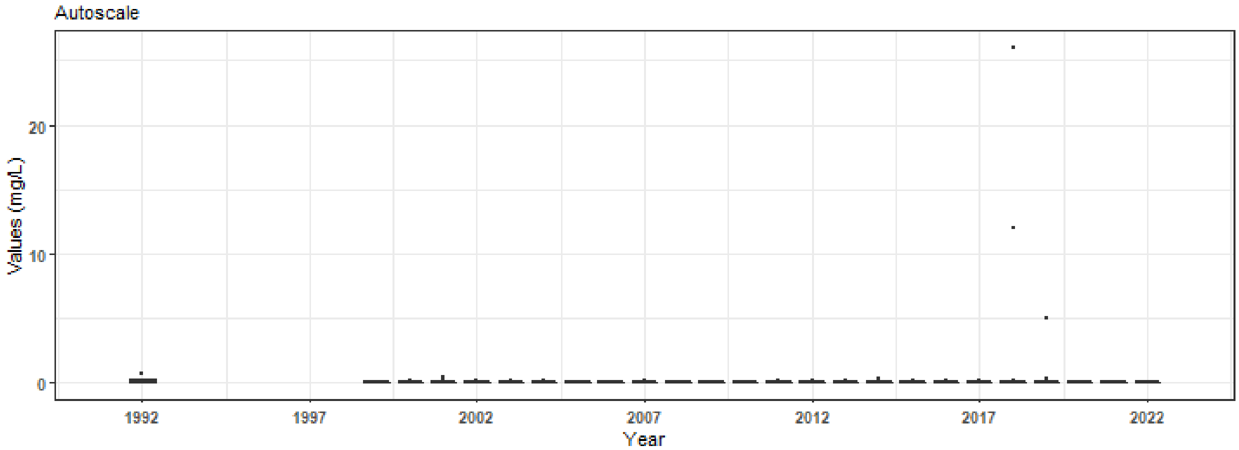


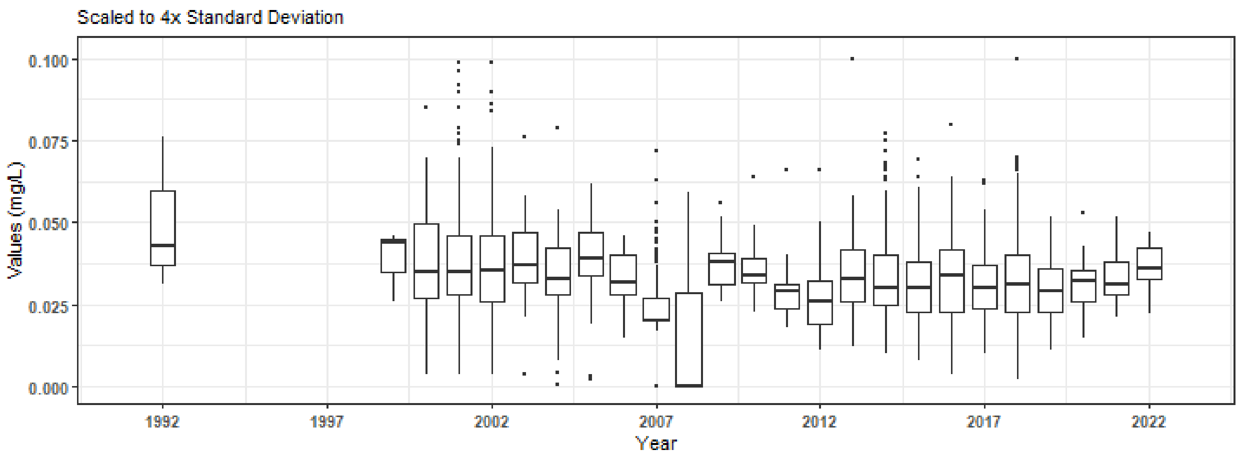


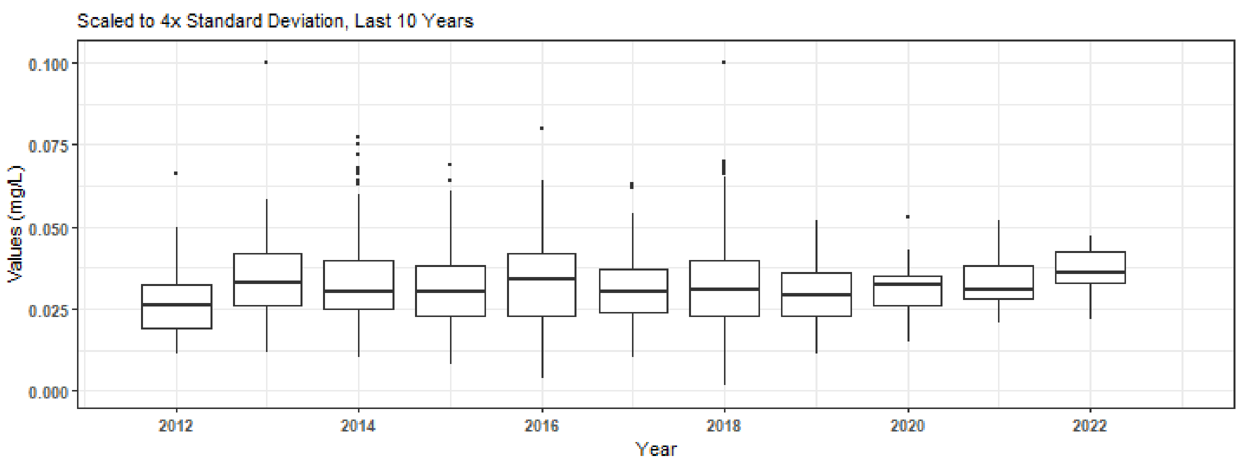
Summary Box Plots for Apalachicola Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
10
20
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.000
0.025
0.050
0.075
0.100
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.000
0.025
0.050
0.075
0.100
3
6
9
12
Month

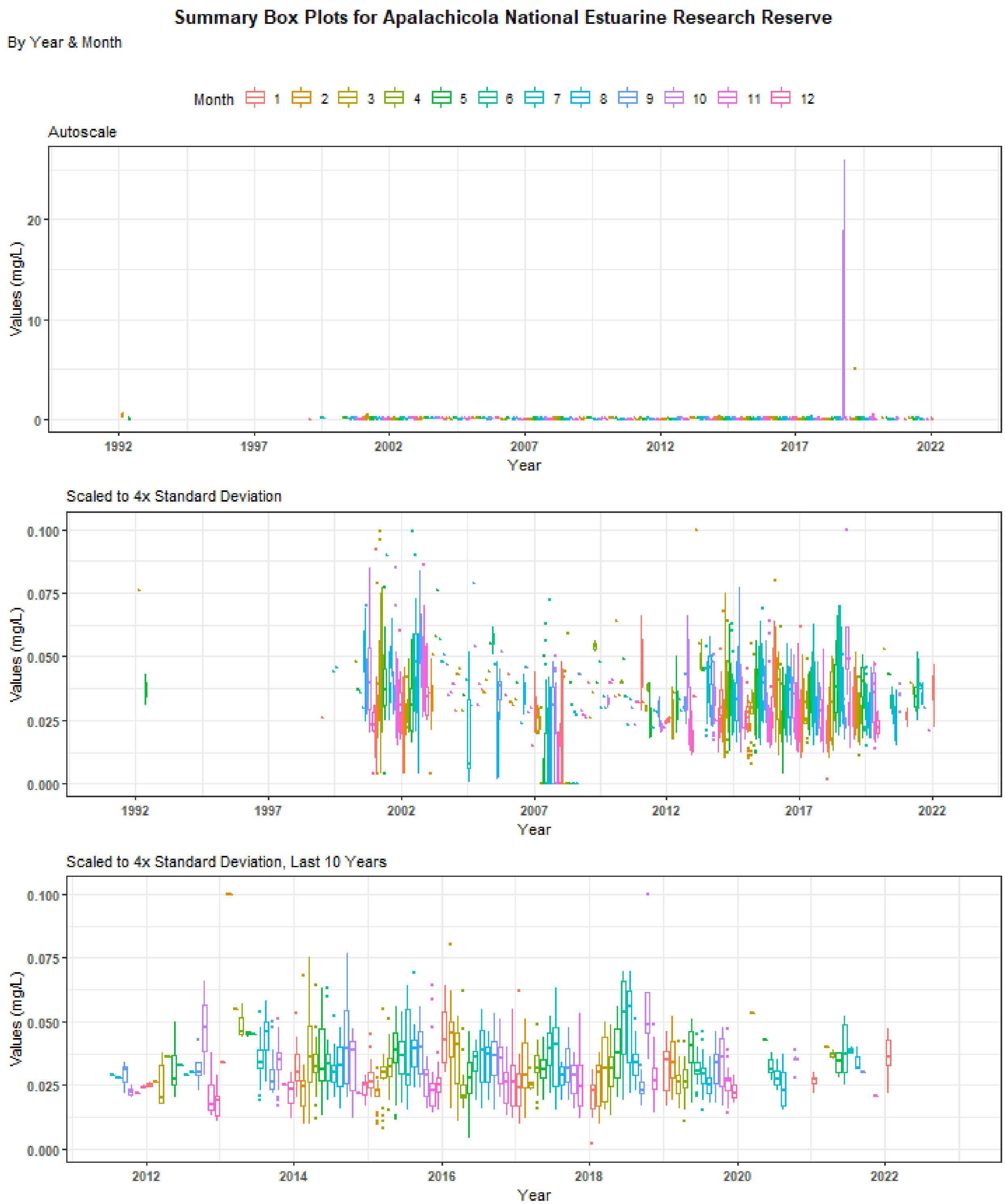
Summary Box Plots for Apalachicola National Estuarine Research Reserve

By Year



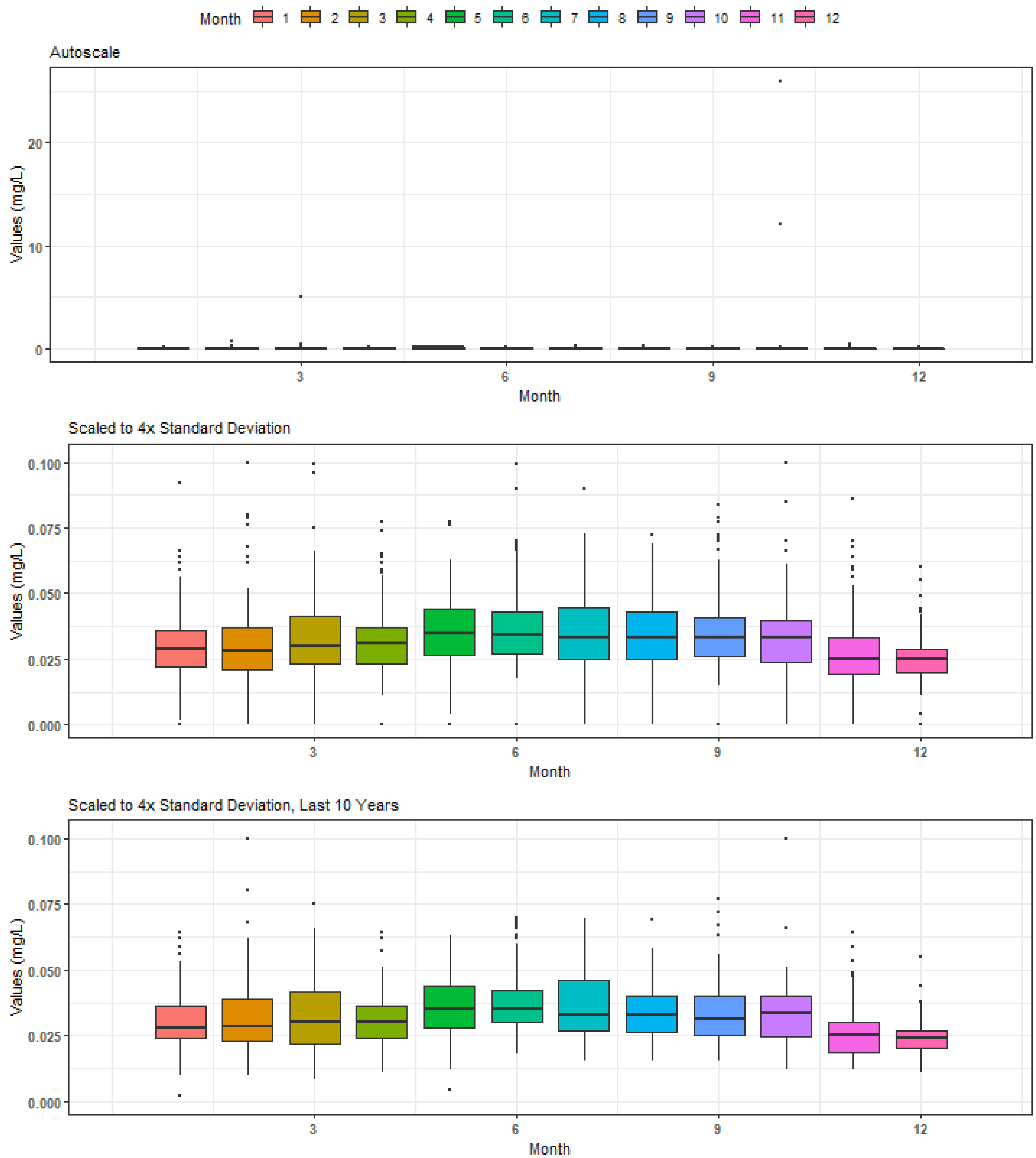





Summary Box Plots for Apalachicola National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
10
20
1992 1997 2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.000 0.025 0.050 0.075 0.100
1992 1997 2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.000 0.025 0.050 0.075 0.100
2012 2014 2016 2018 2020 2022
Year

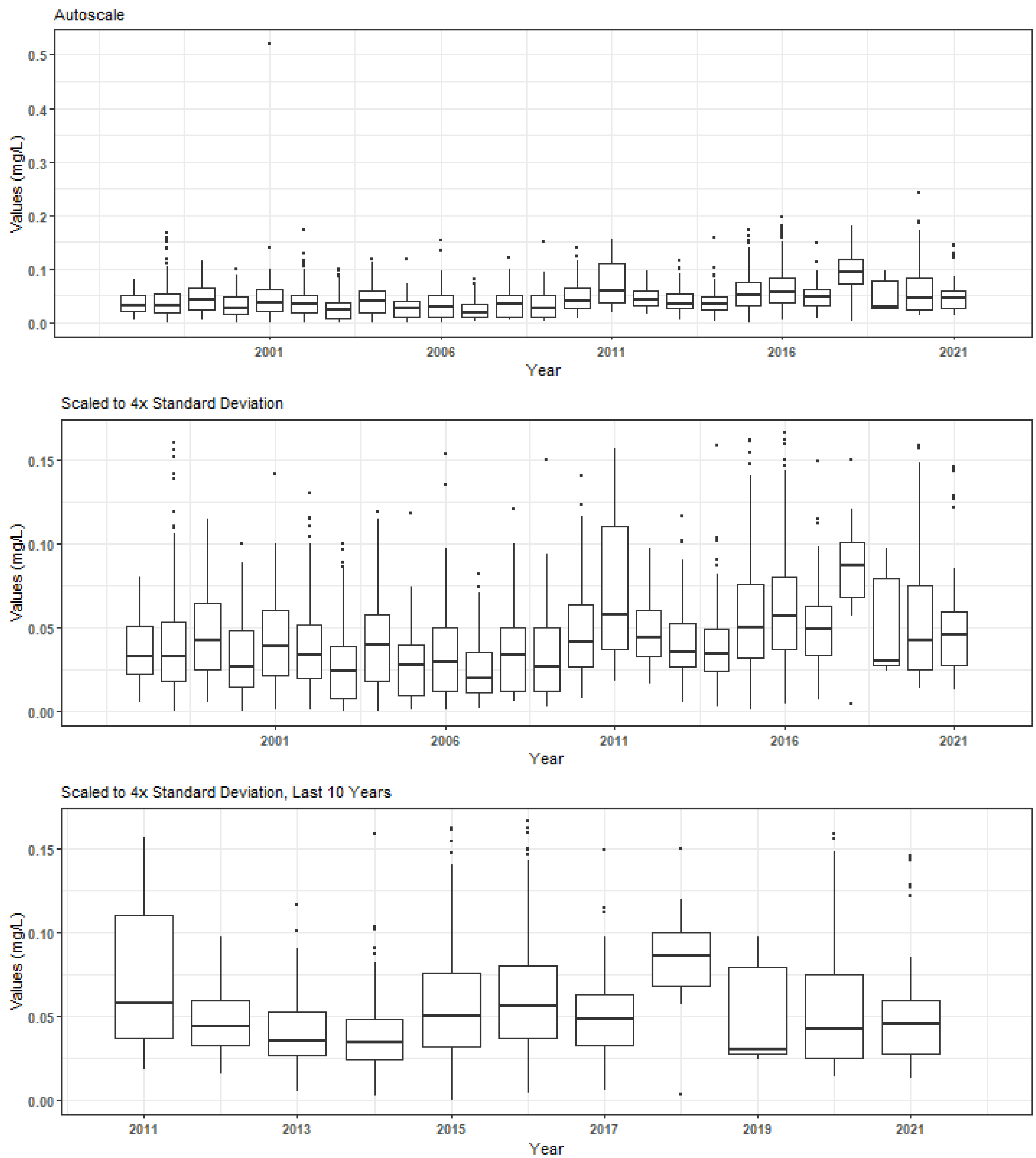
Summary Box Plots for Apalachicola National Estuarine Research Reserve

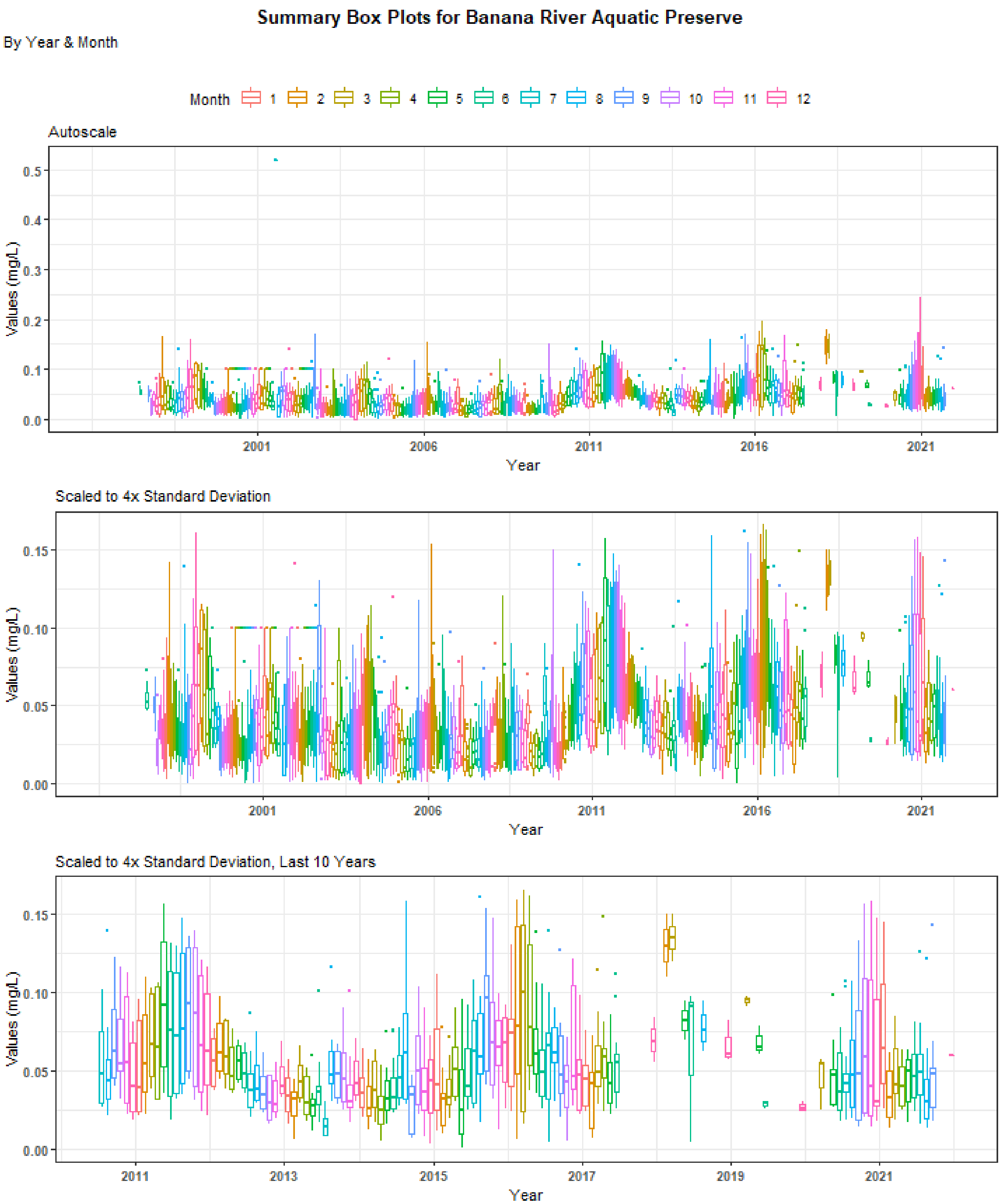
By Month



Summary Box Plots for Banana River Aquatic Preserve

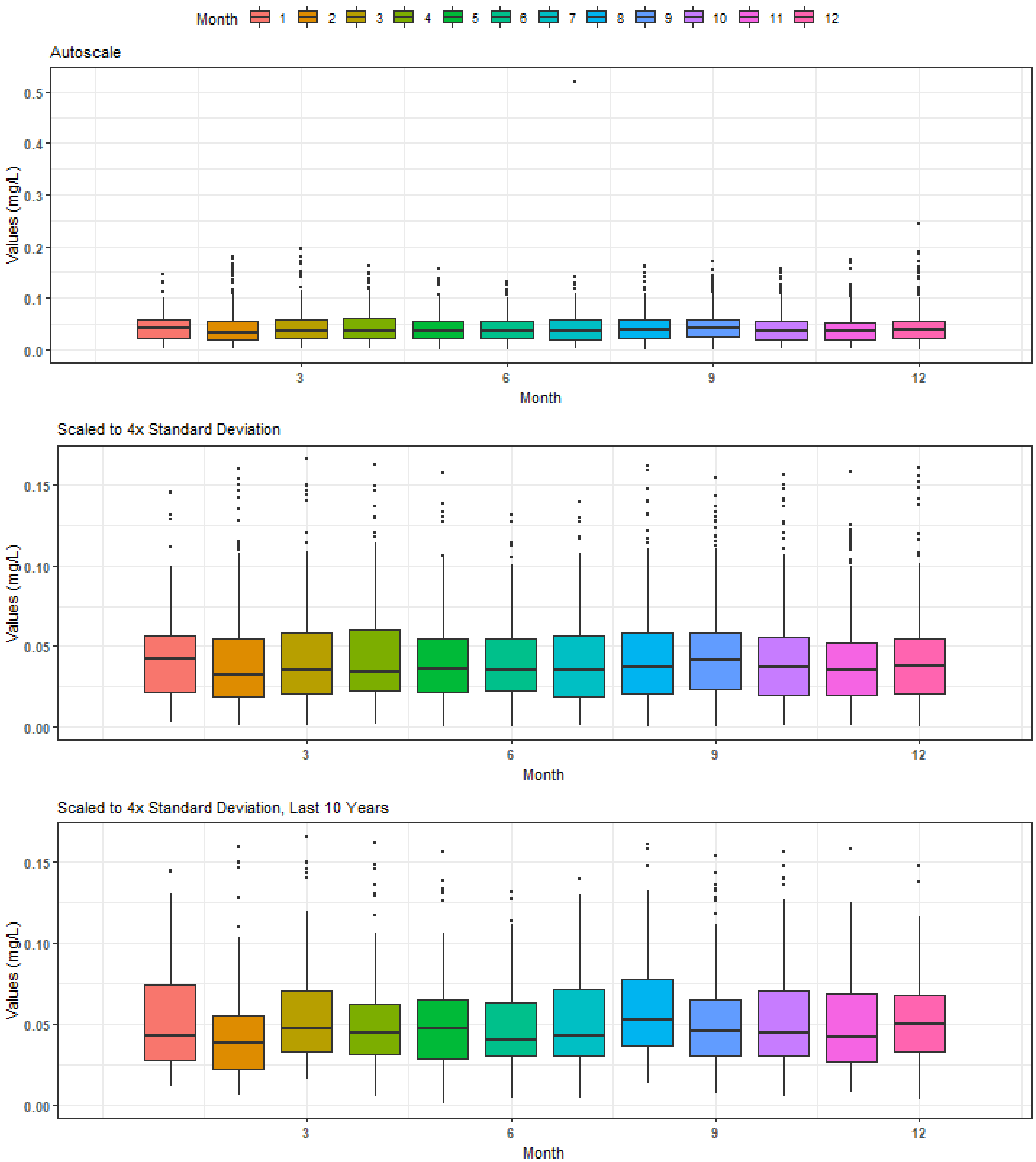
By Year

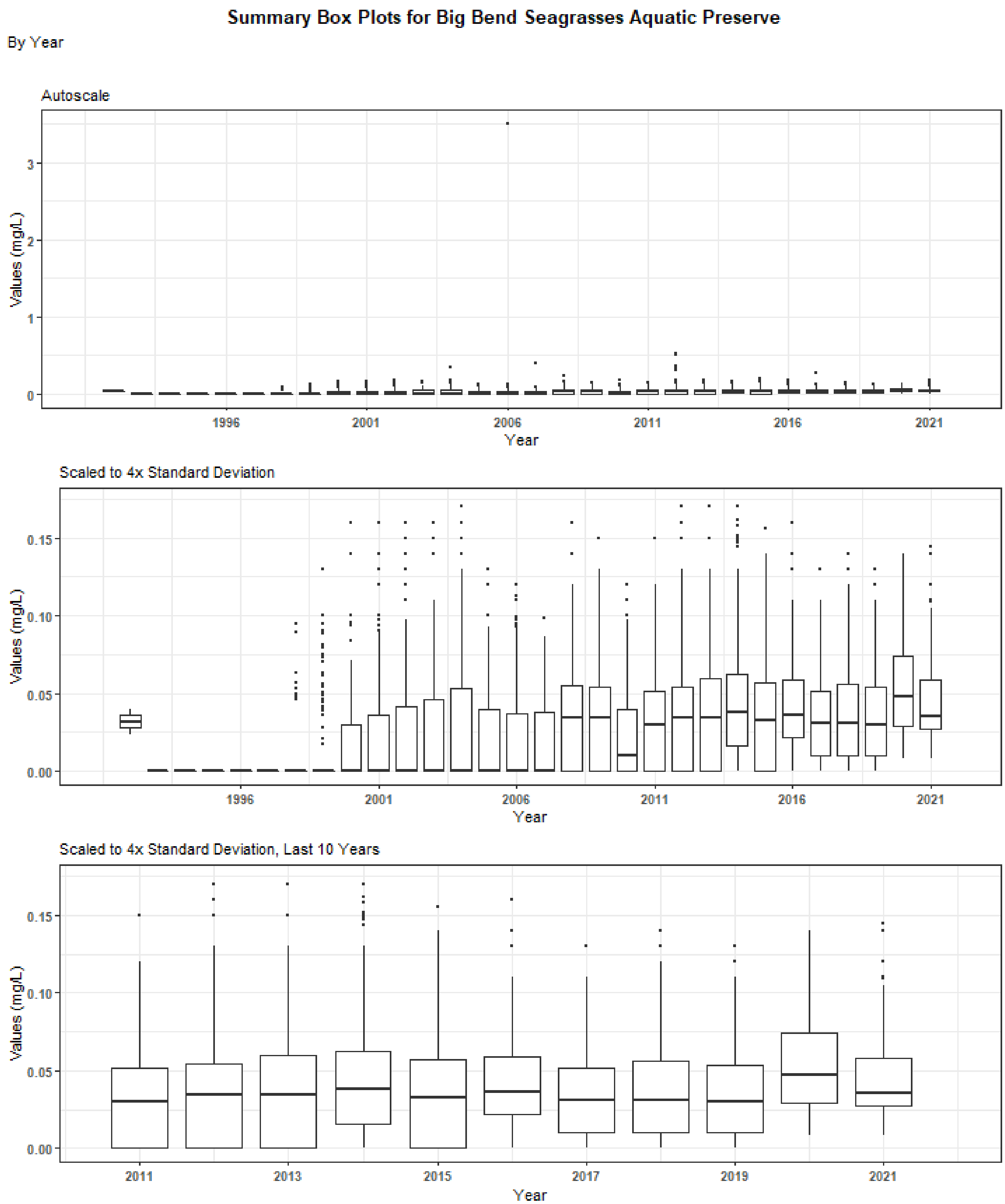



Summary Box Plots for Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

Summary Box Plots for Banana River Aquatic Preserve

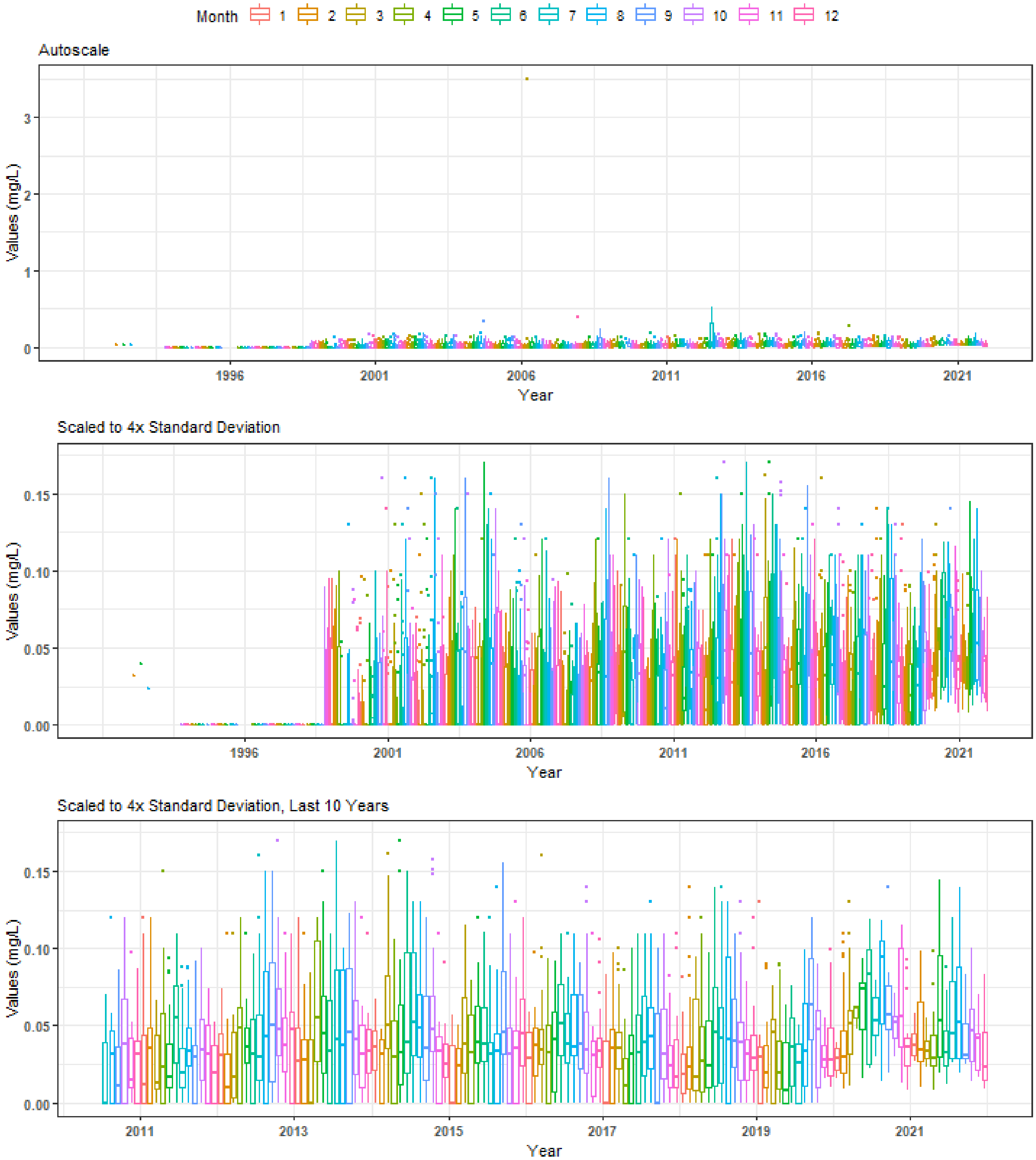
By Month




Summary Box Plots for Big Bend Seagrasses Aquatic Preserve
By Year
Autoscale
Values (mg/L)
0
1
2
3
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
2011
2013
2015
2017
2019
2021
Year

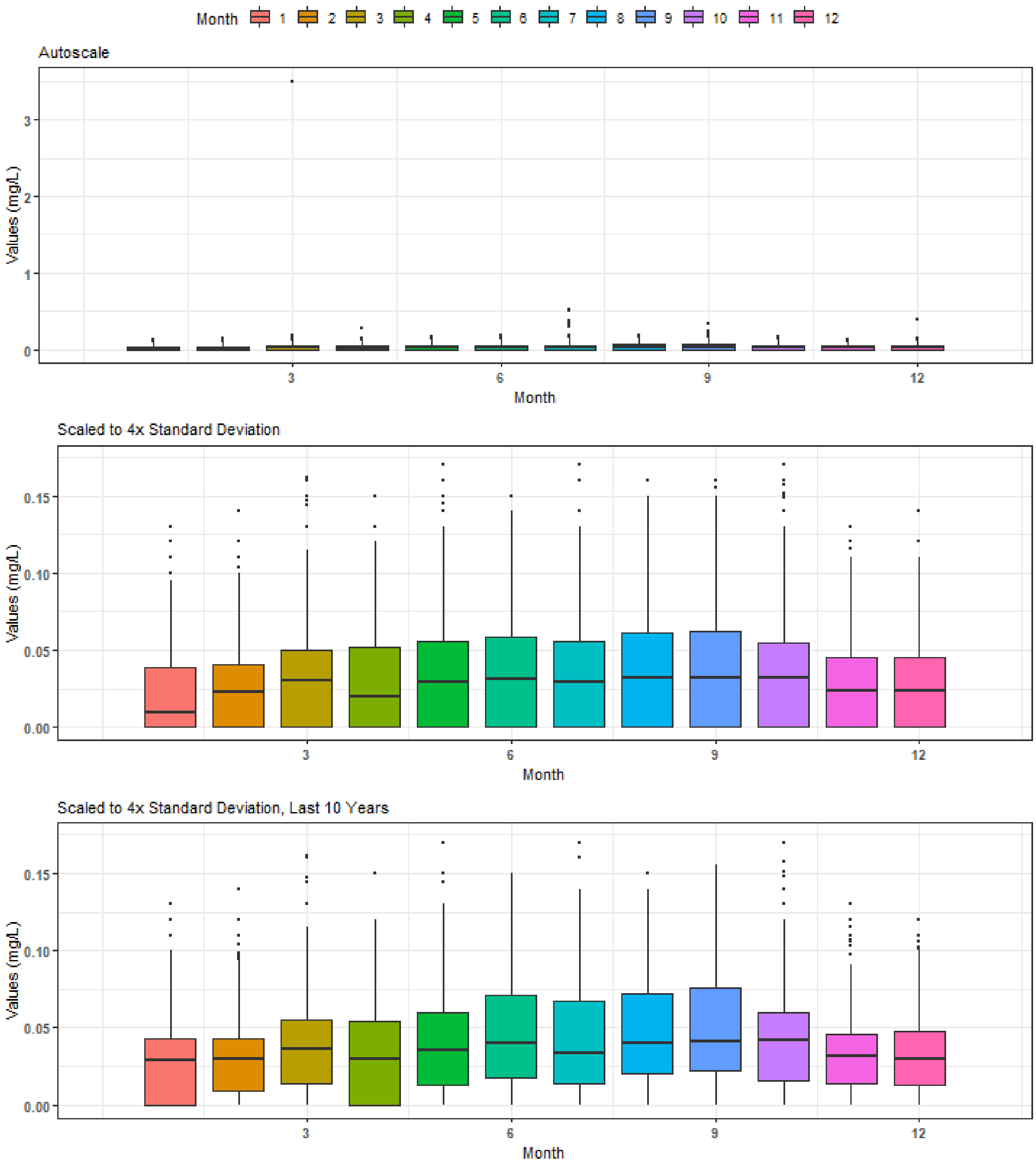
Summary Box Plots for Big Bend Seagrasses Aquatic Preserve

By Year & Month



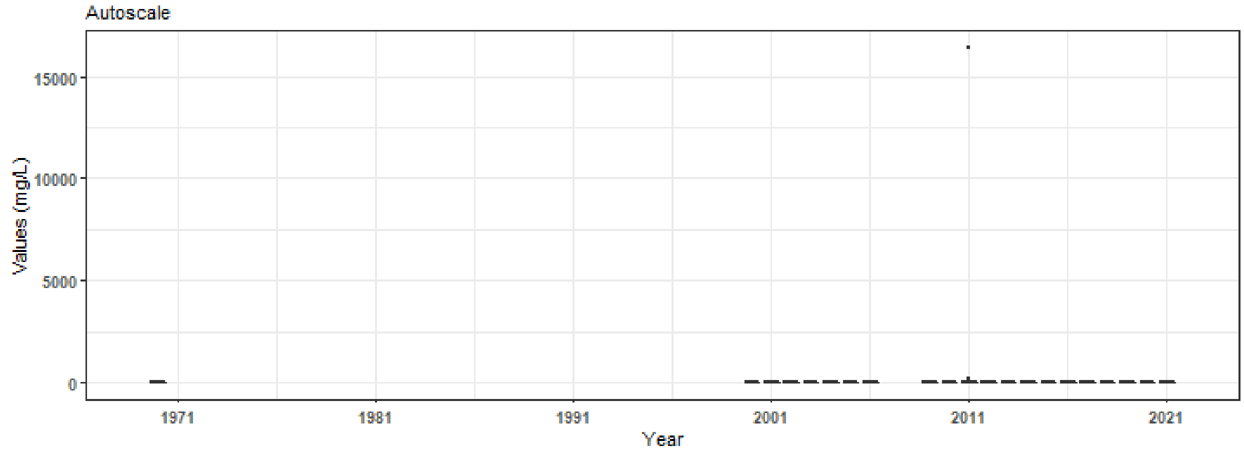
Summary Box Plots for Big Bend Seagrasses Aquatic Preserve

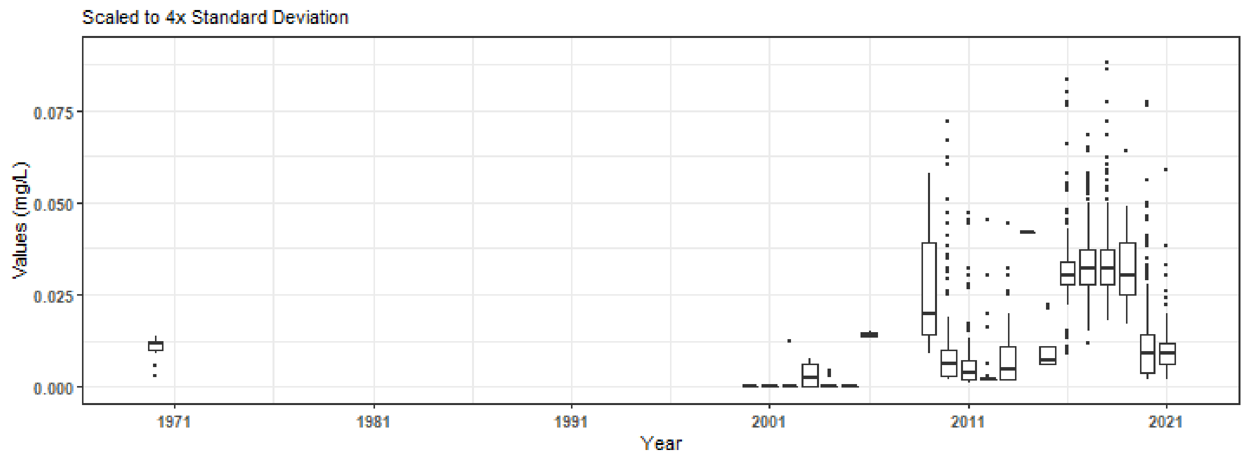
By Month

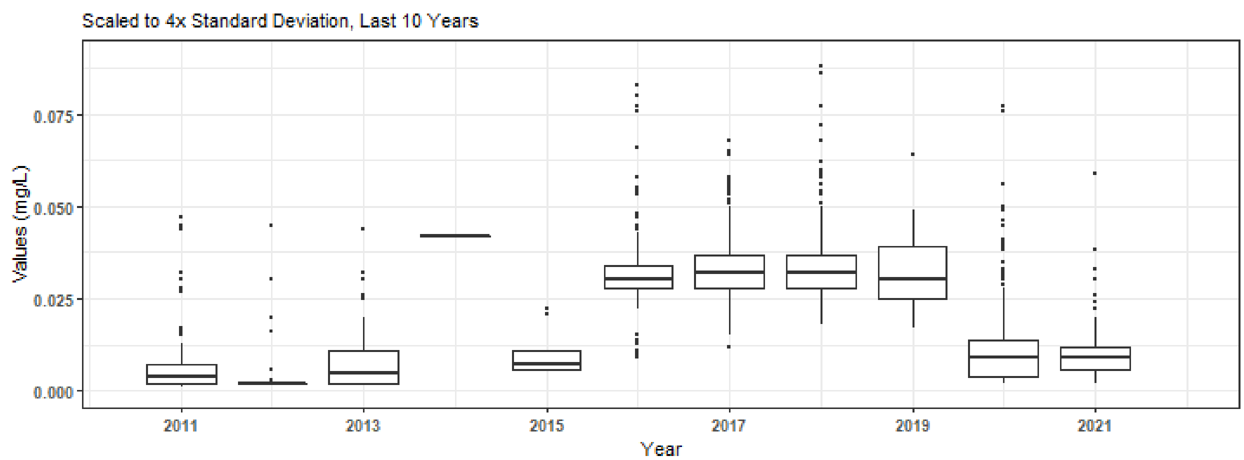


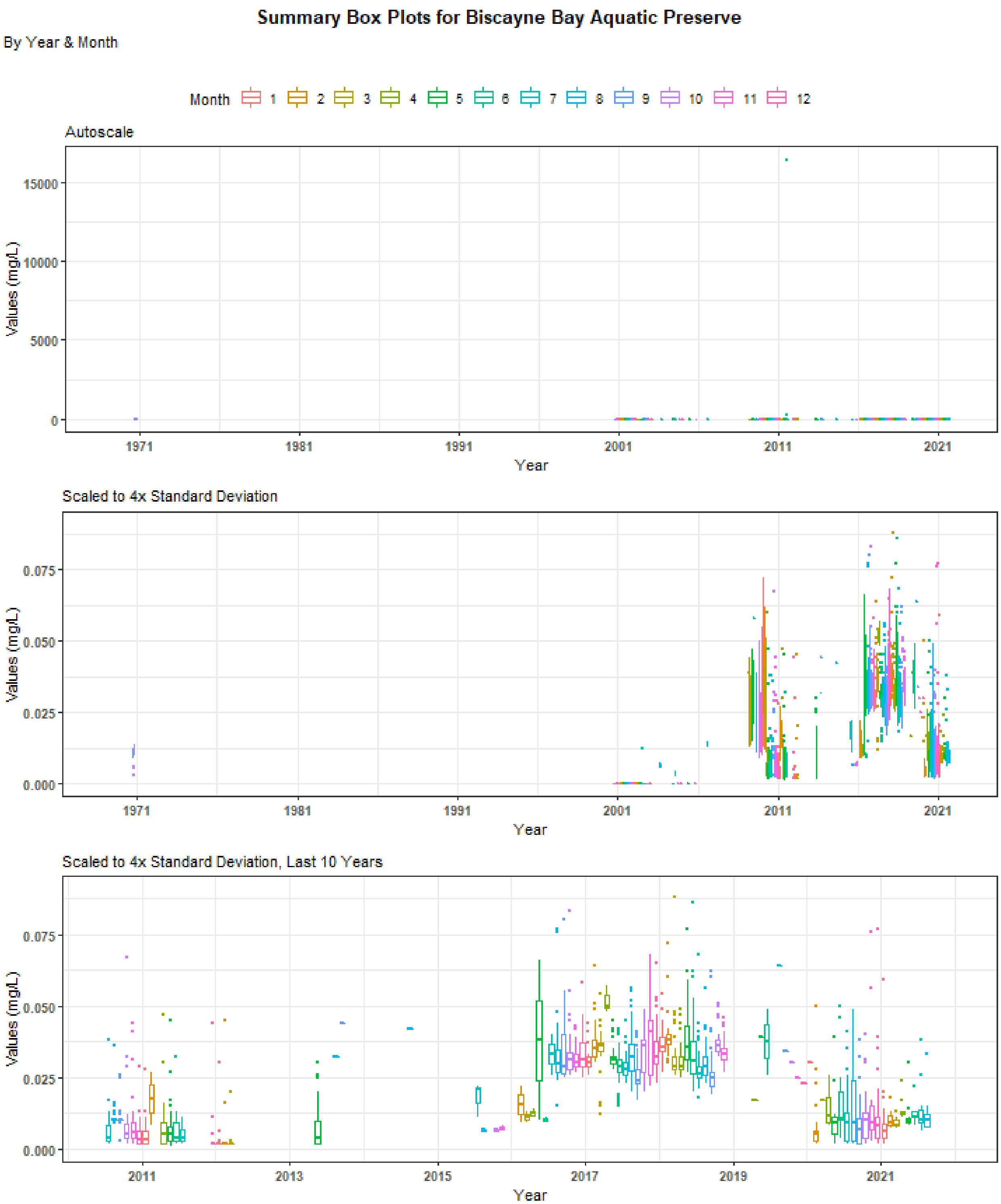
Summary Box Plots for Biscayne Bay Aquatic Preserve

By Year





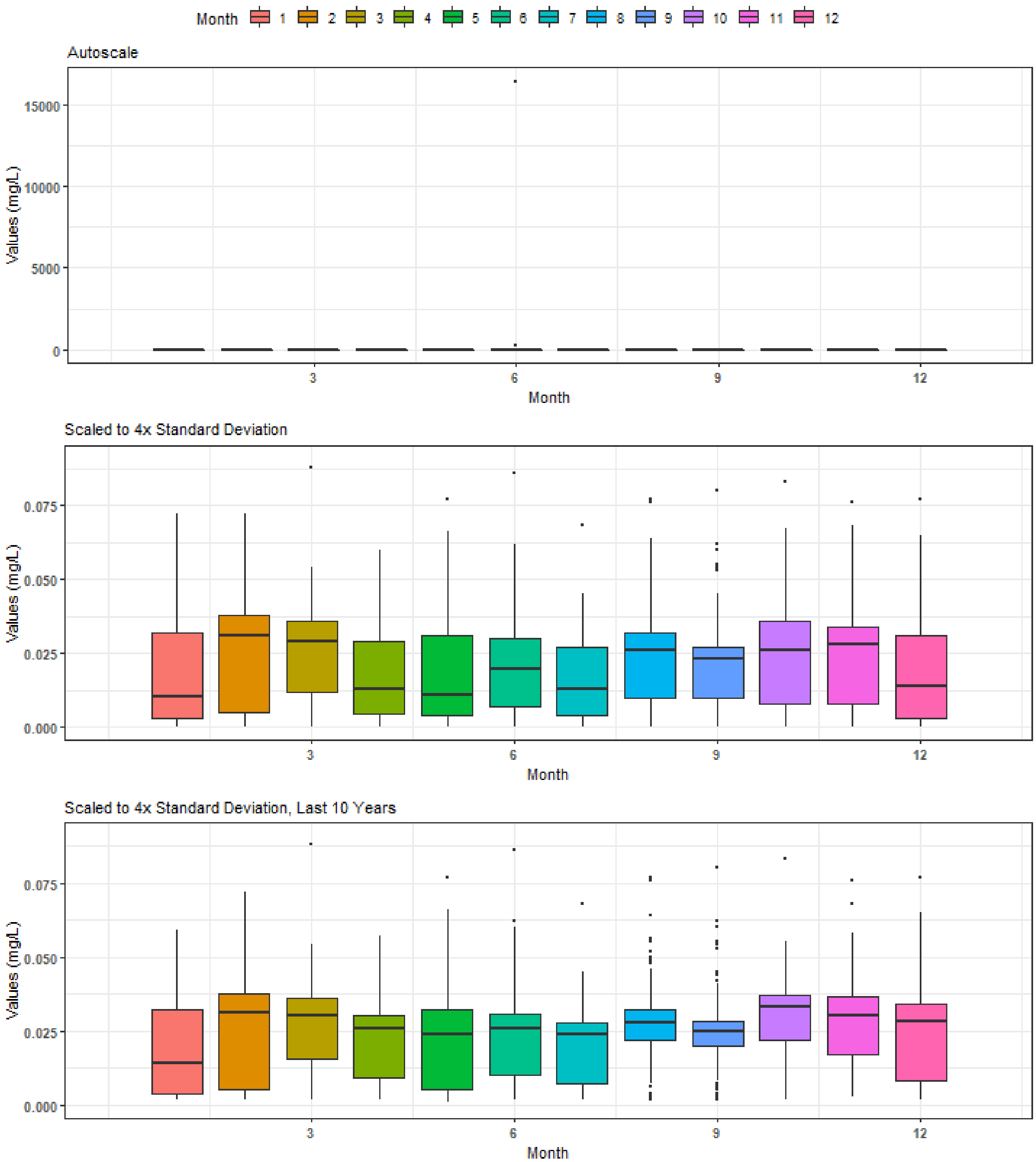




Summary Box Plots for Biscayne Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
15000
10000
5000
0
Values (mg/L)
1971 1981 1991 2001 2011 2021
Year
Scaled to 4x Standard Deviation
0.075
0.050
0.025
0.000
Values (mg/L)
1971 1981 1991 2001 2011 2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
0.075
0.050
0.025
0.000
Values (mg/L)
2011 2013 2015 2017 2019 2021
Year

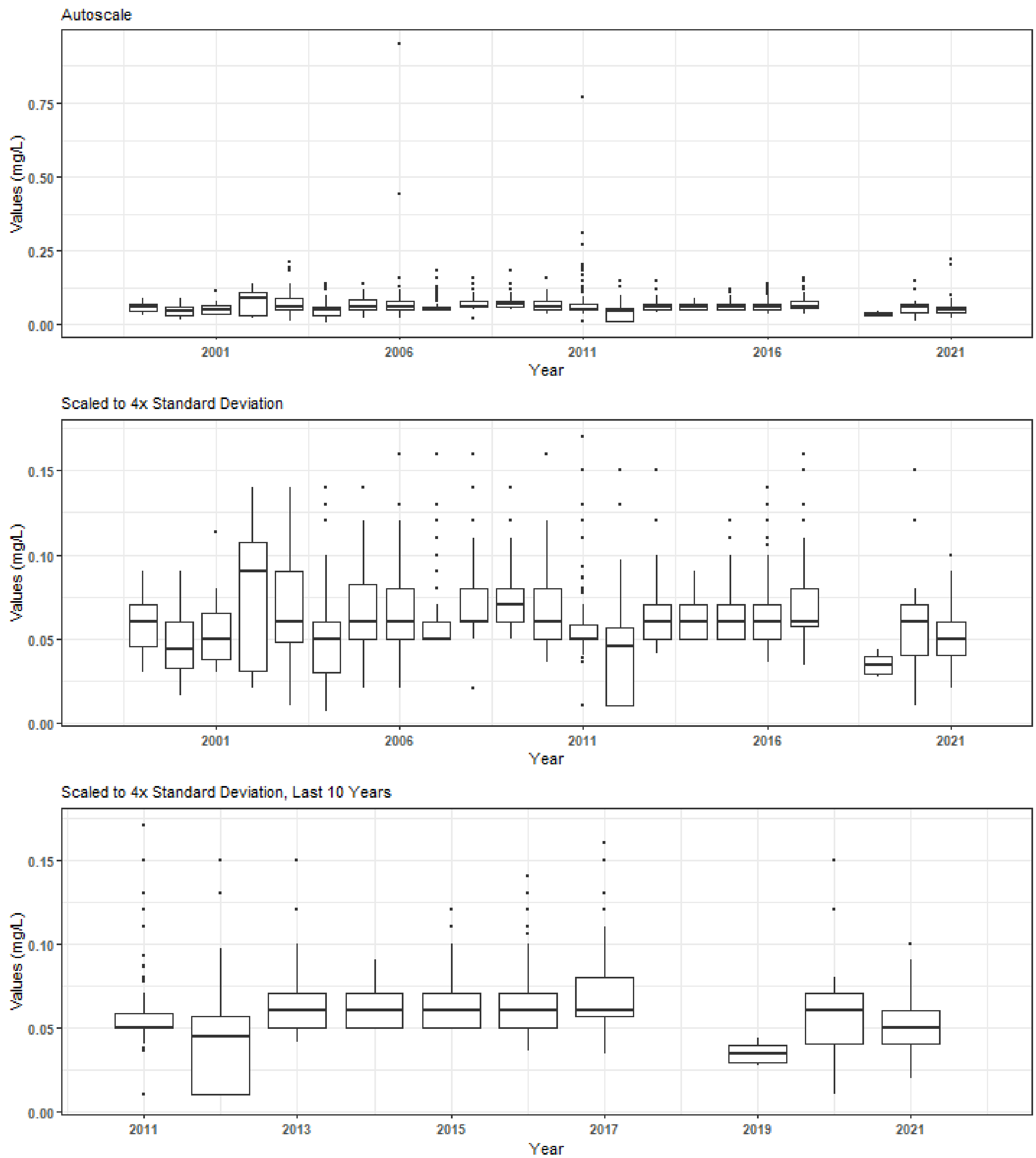
Summary Box Plots for Biscayne Bay Aquatic Preserve

By Month



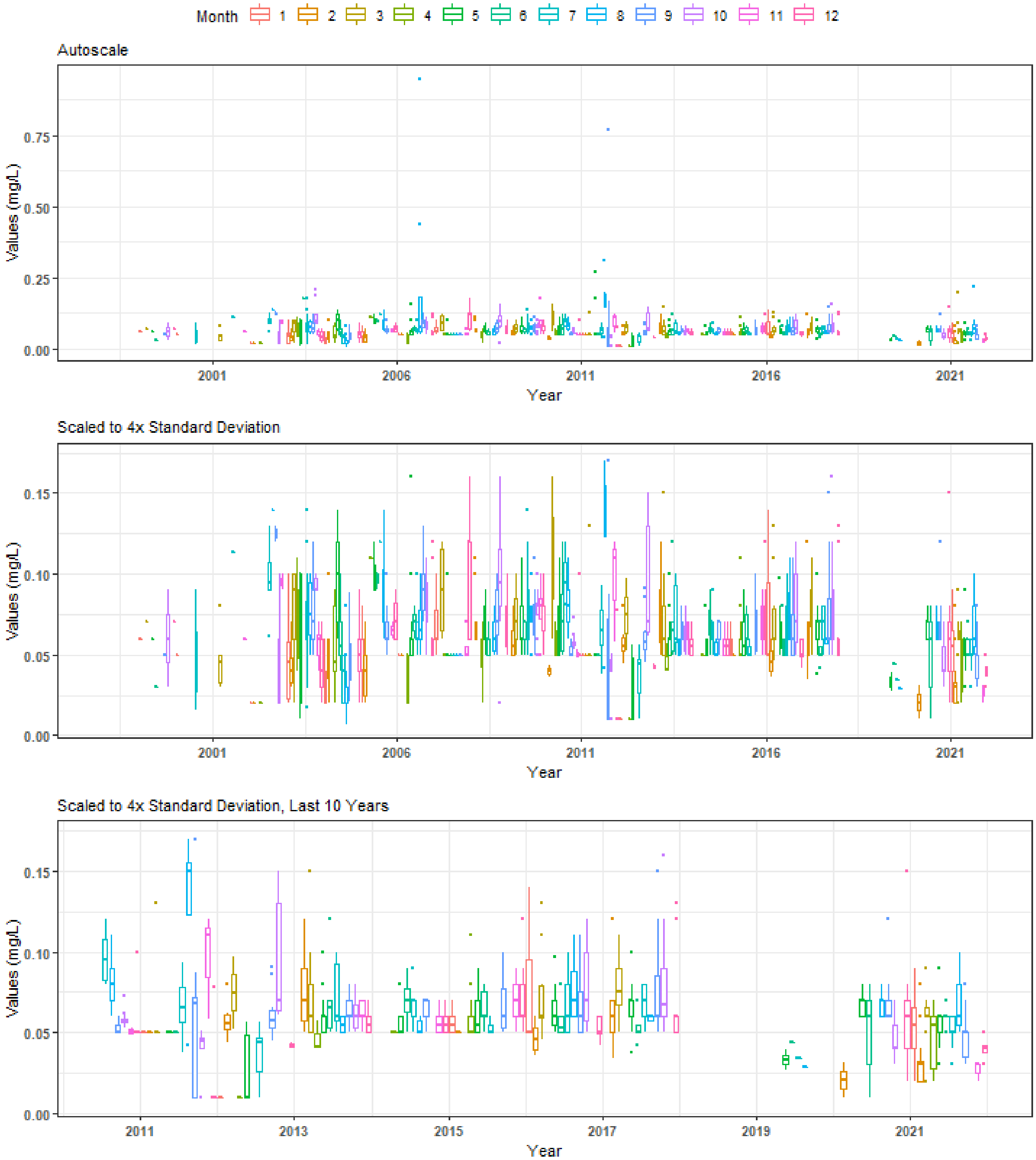
Summary Box Plots for Boca Ciega Bay Aquatic Preserve

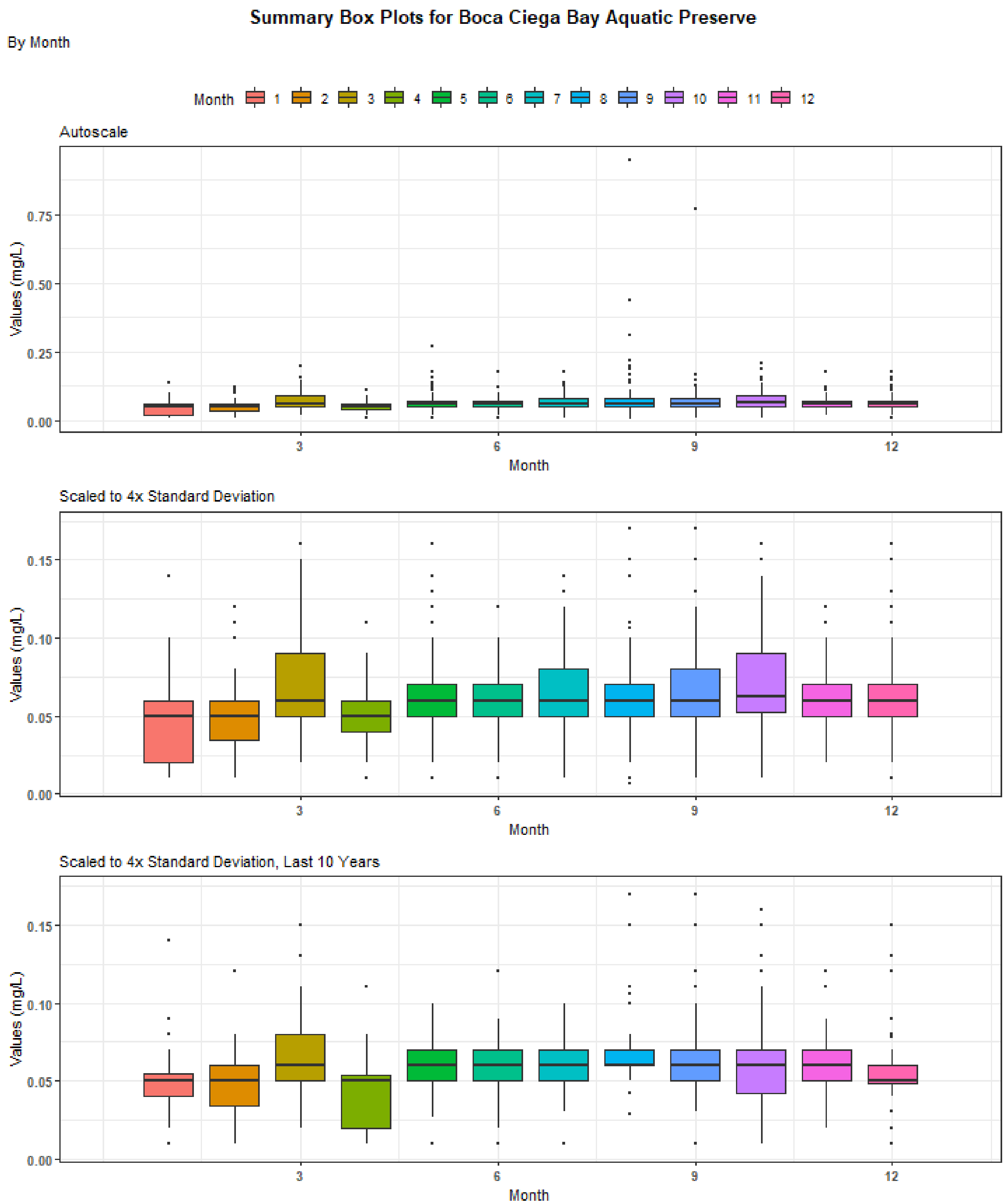
By Year



Summary Box Plots for Boca Ciega Bay Aquatic Preserve

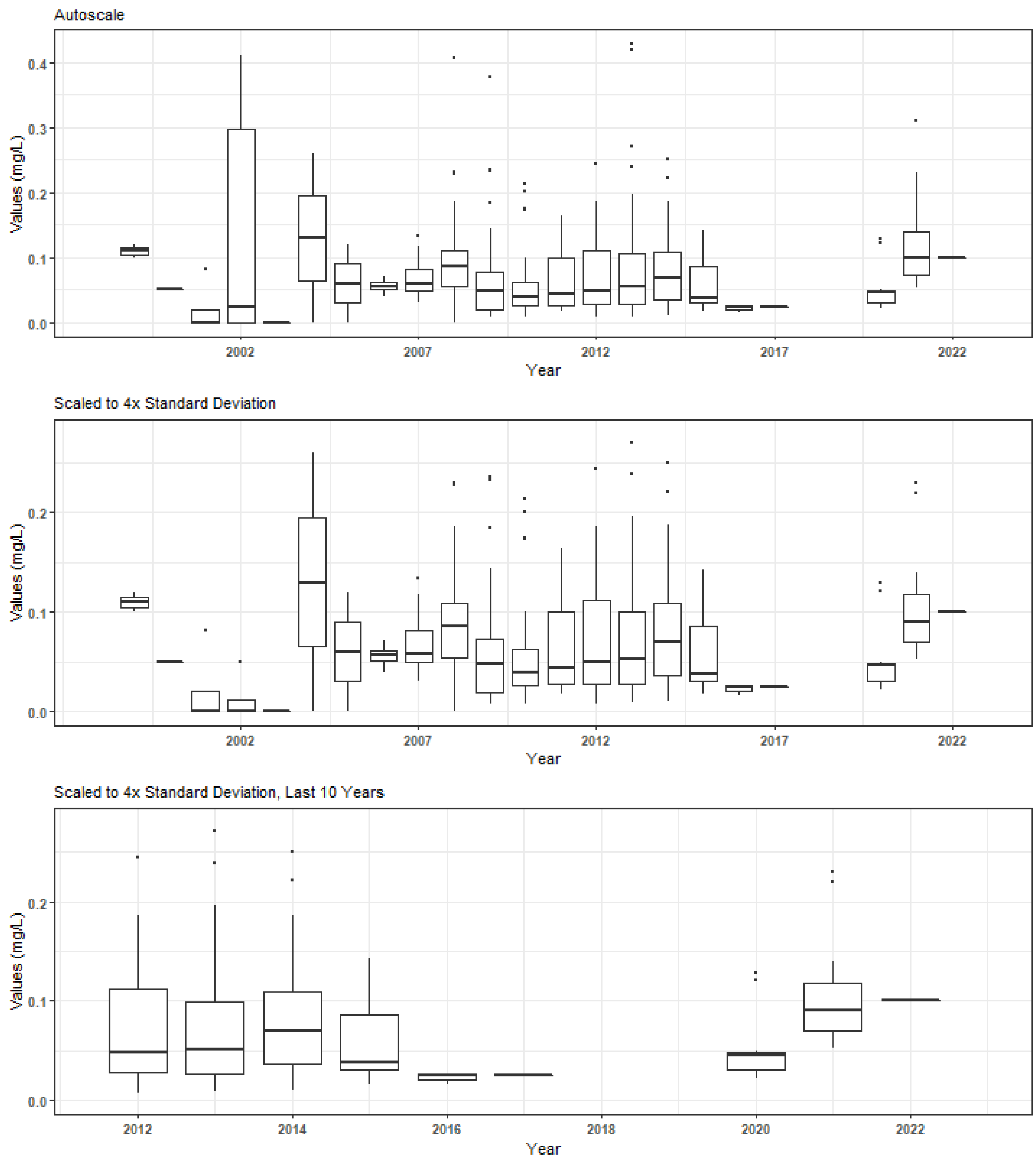
By Year & Month

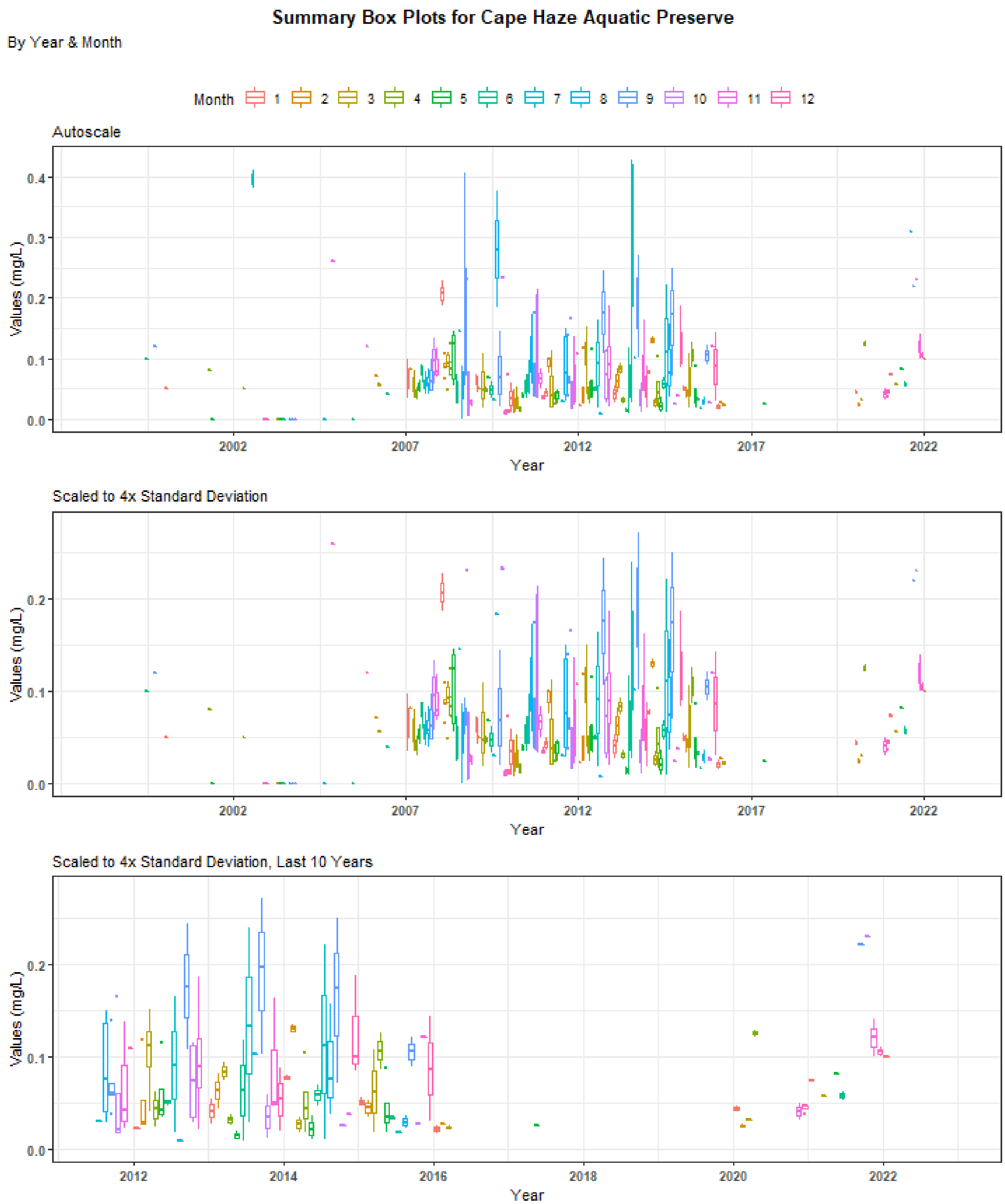


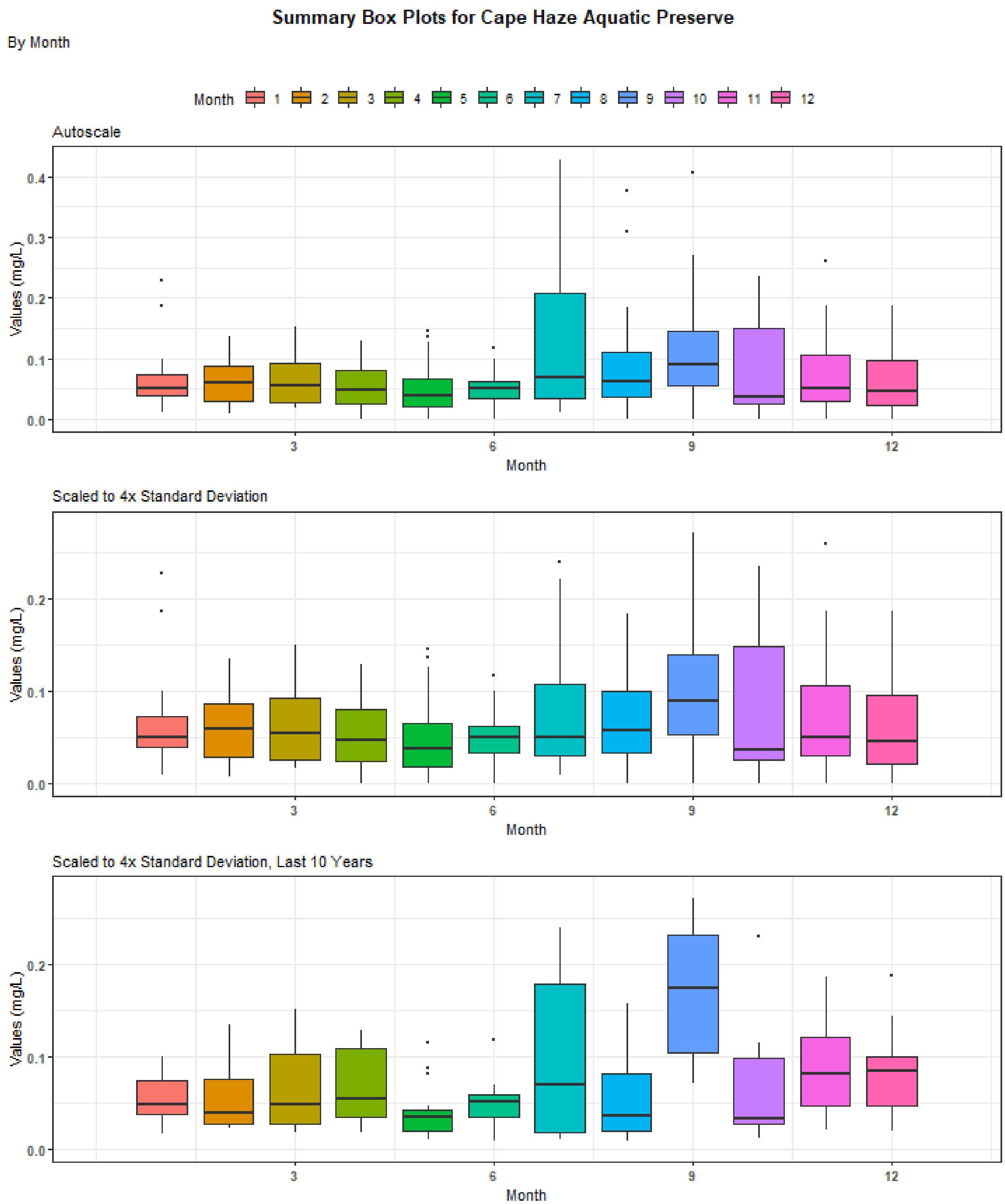

Summary Box Plots for Boca Ciega Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.00
0.25
0.50
0.75
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
3
6
9
12
Month

Summary Box Plots for Cape Haze Aquatic Preserve

By Year



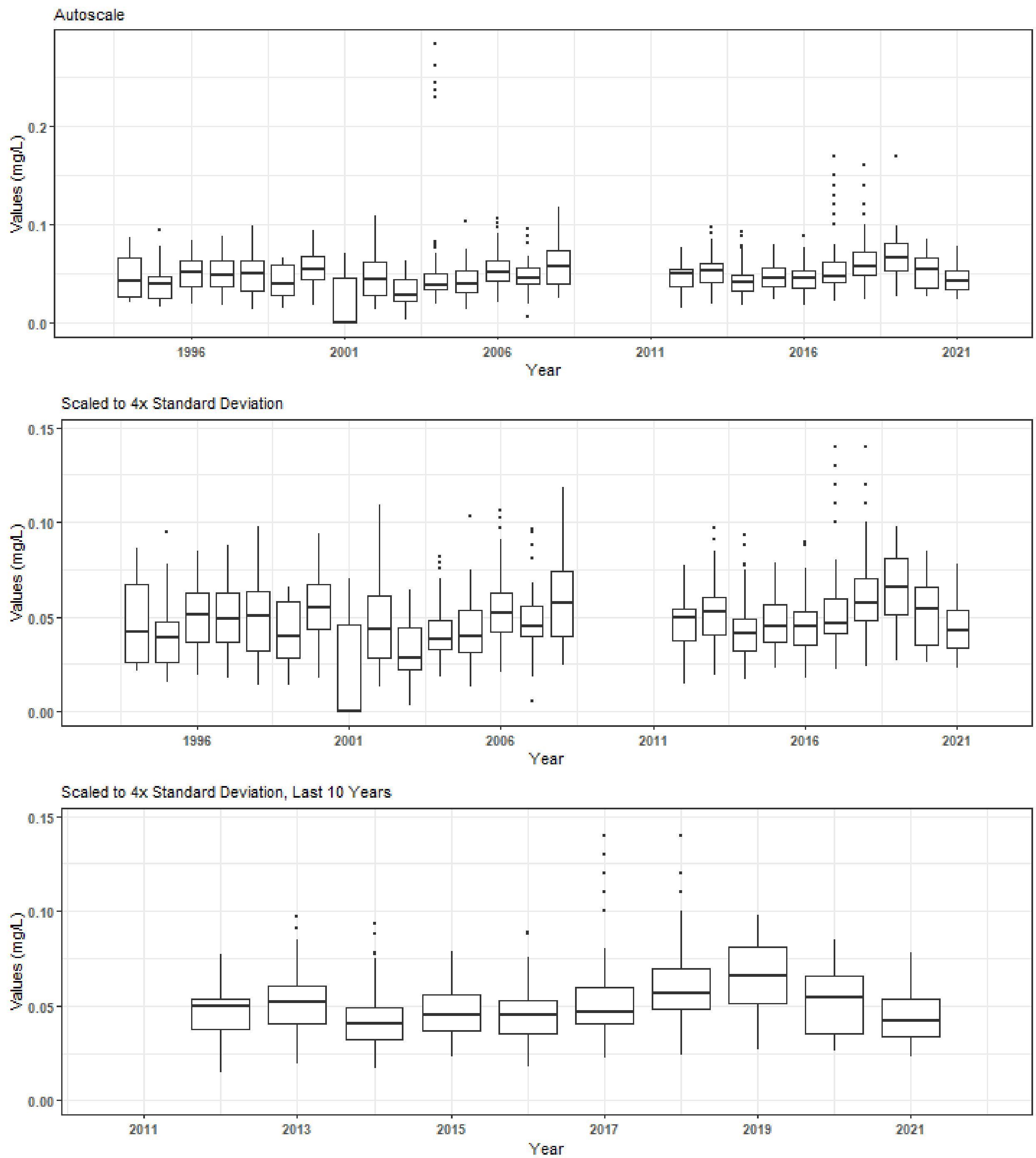

Summary Box Plots for Cape Haze Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
2012
2014
2016
2018
2020
2022
Year

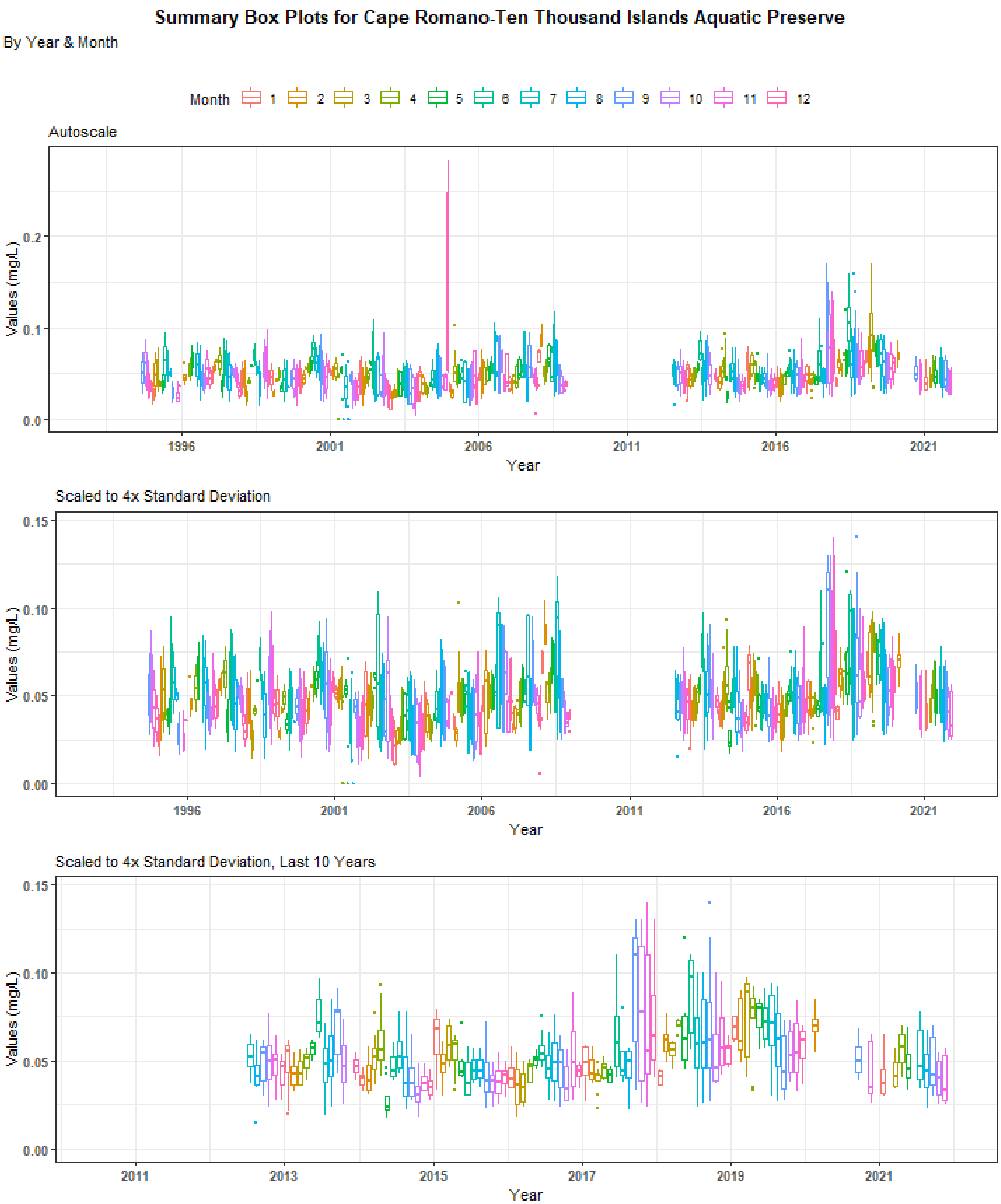


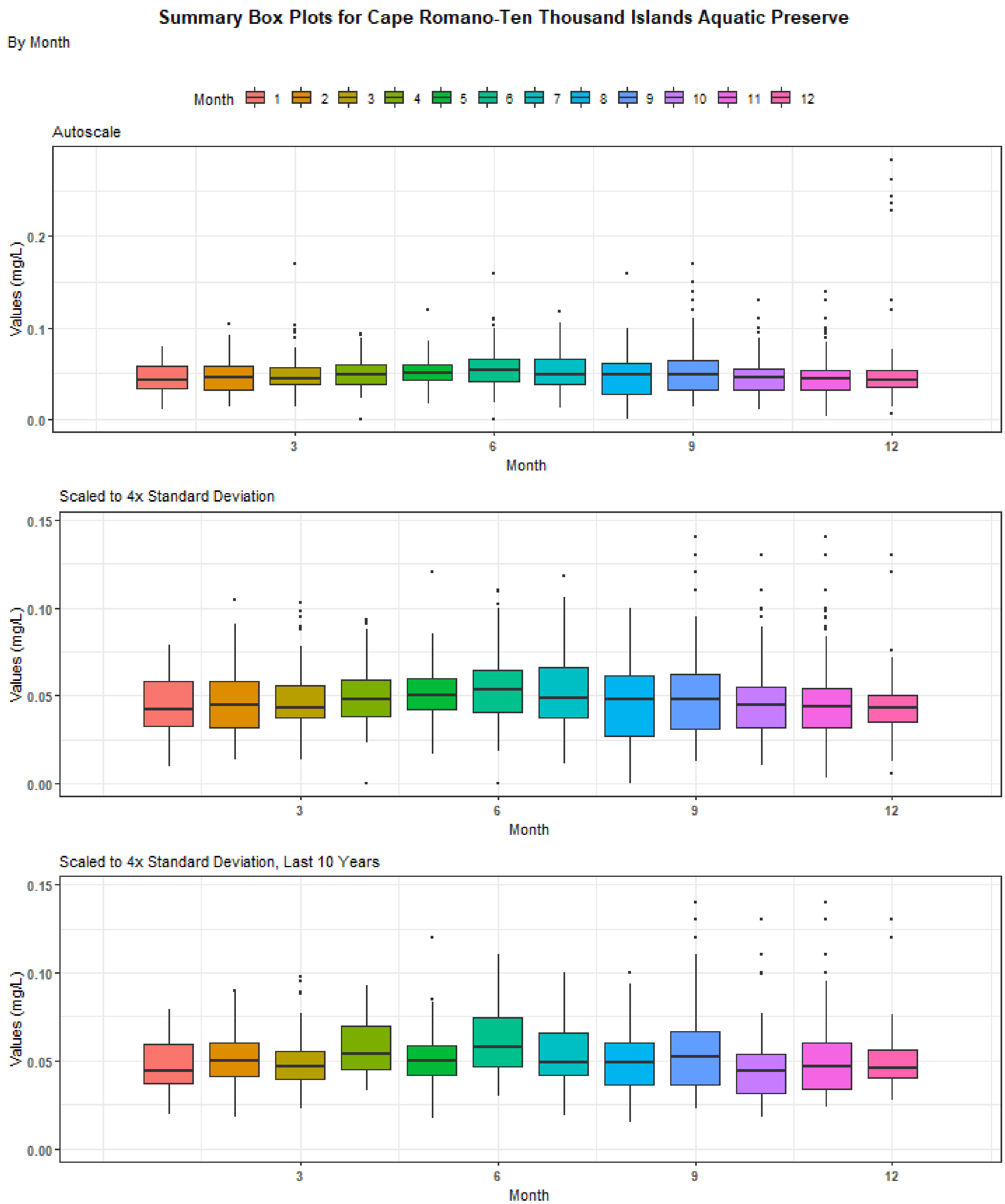
Summary Box Plots for Cape Haze Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
3
6
9
12
Month

Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve

By Year

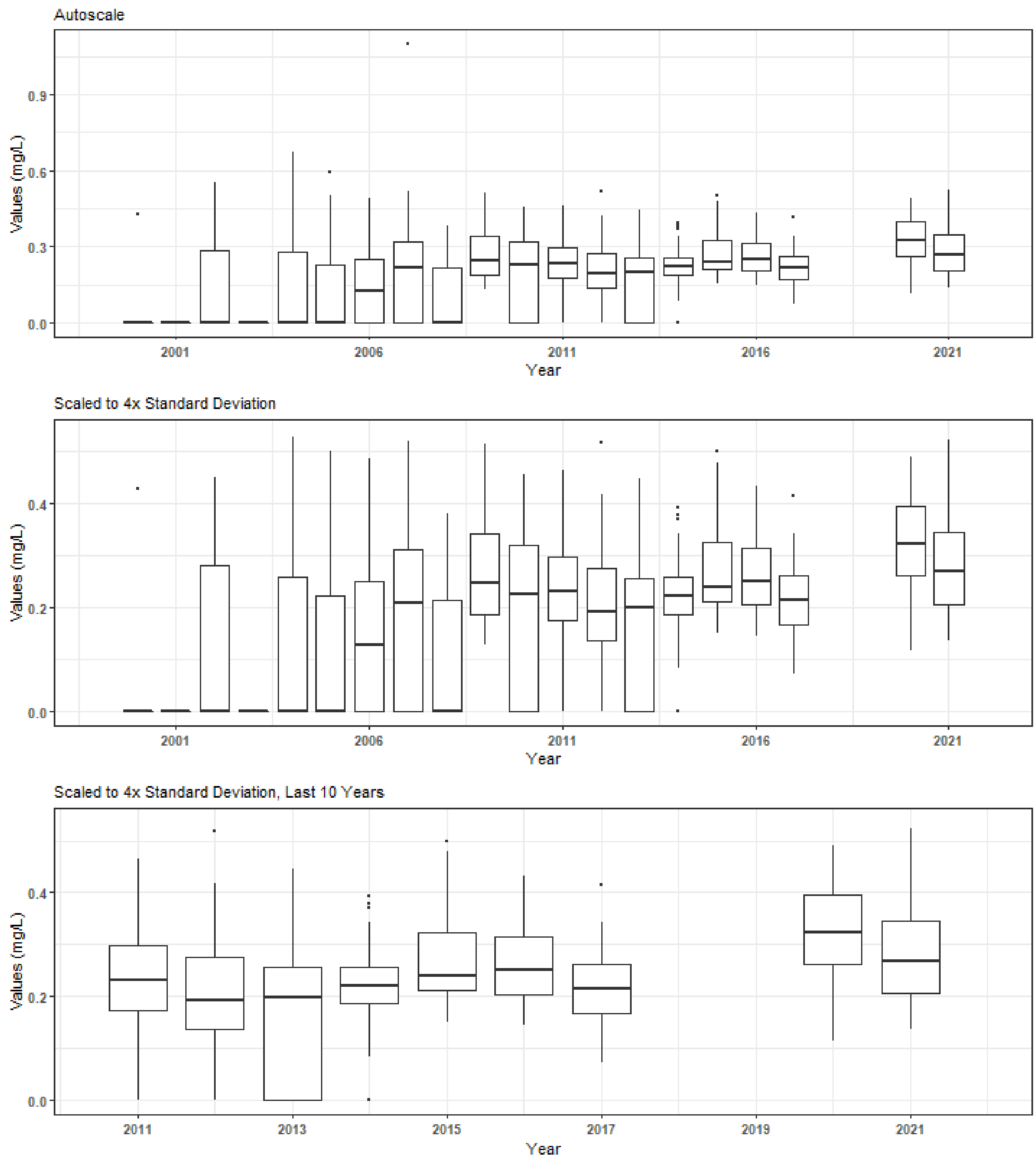


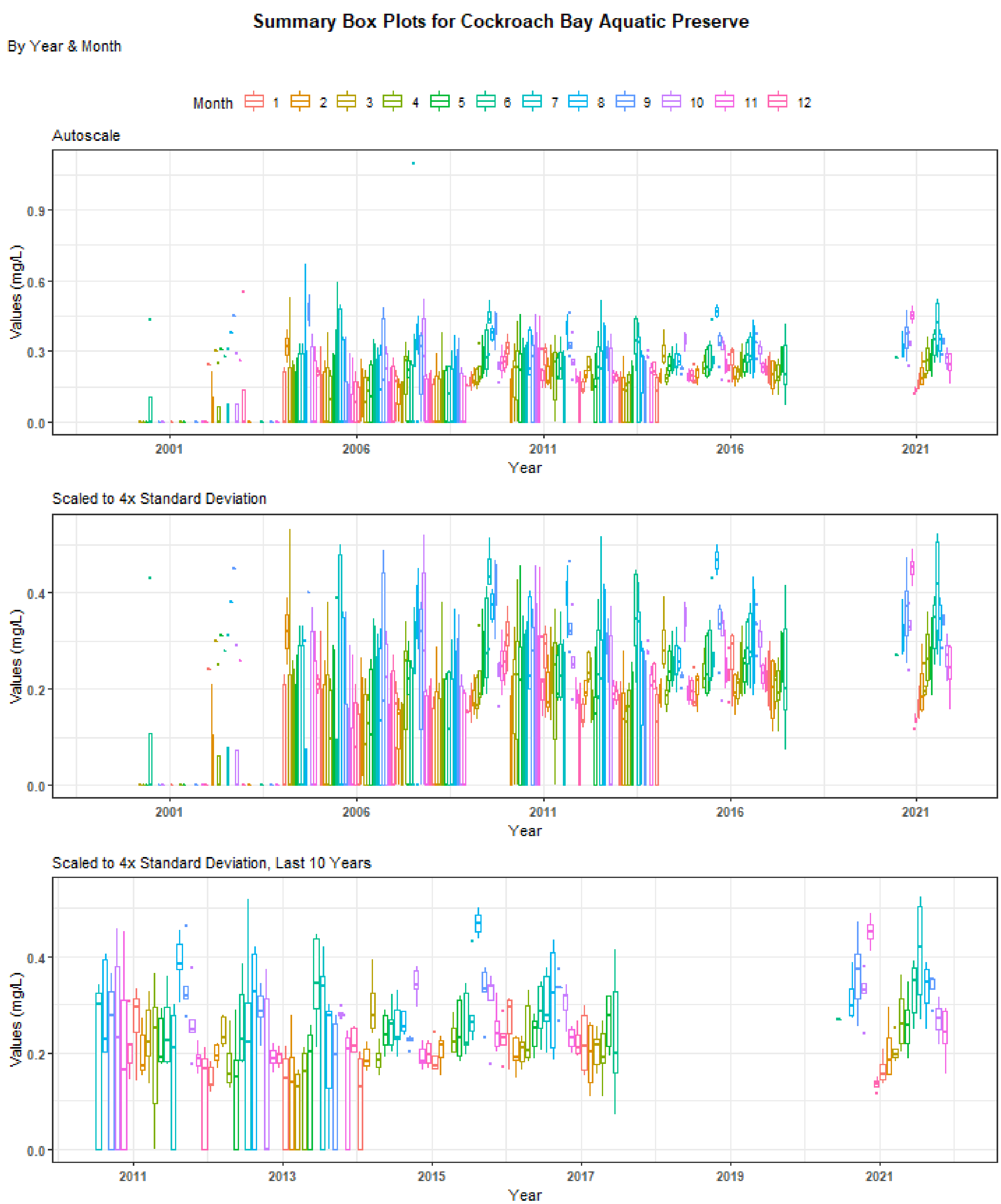

Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
2011
2013
2015
2017
2019
2021
Year


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
3
6
9
12
Month

Summary Box Plots for Cockroach Bay Aquatic Preserve

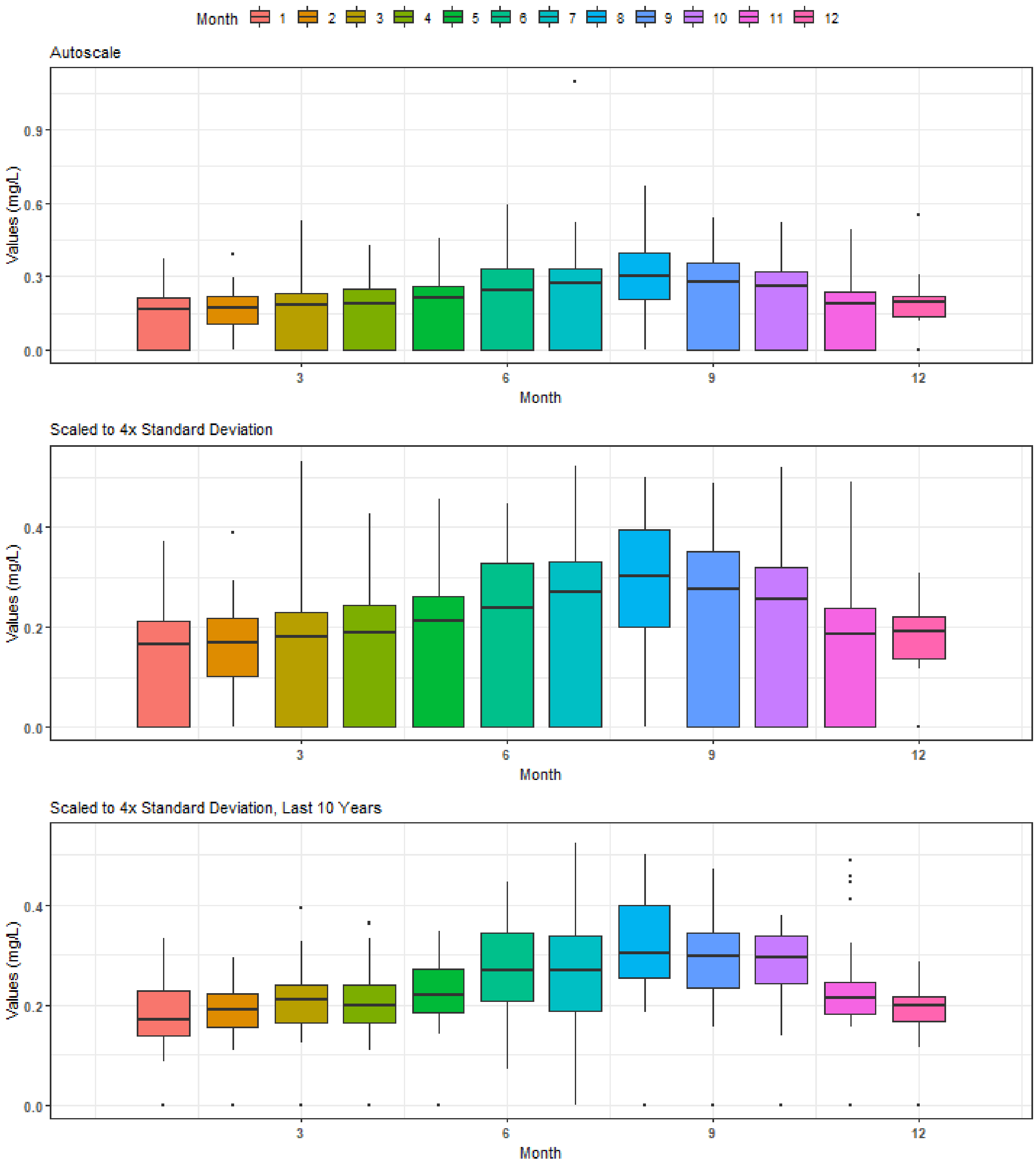
By Year




Summary Box Plots for Cockroach Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.3
0.6
0.9
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.2
0.4
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.2
0.4
2011
2013
2015
2017
2019
2021
Year

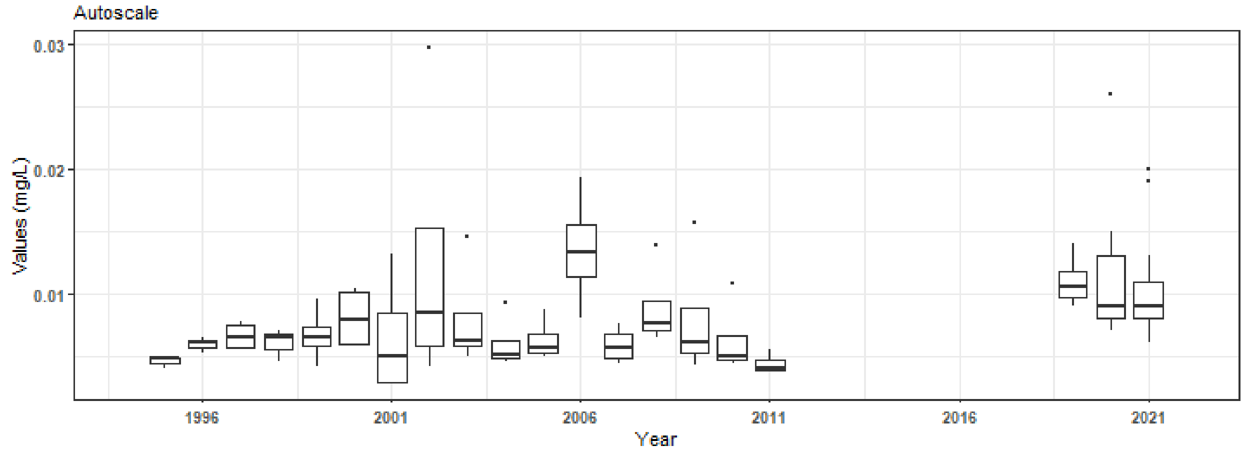
Summary Box Plots for Cockroach Bay Aquatic Preserve

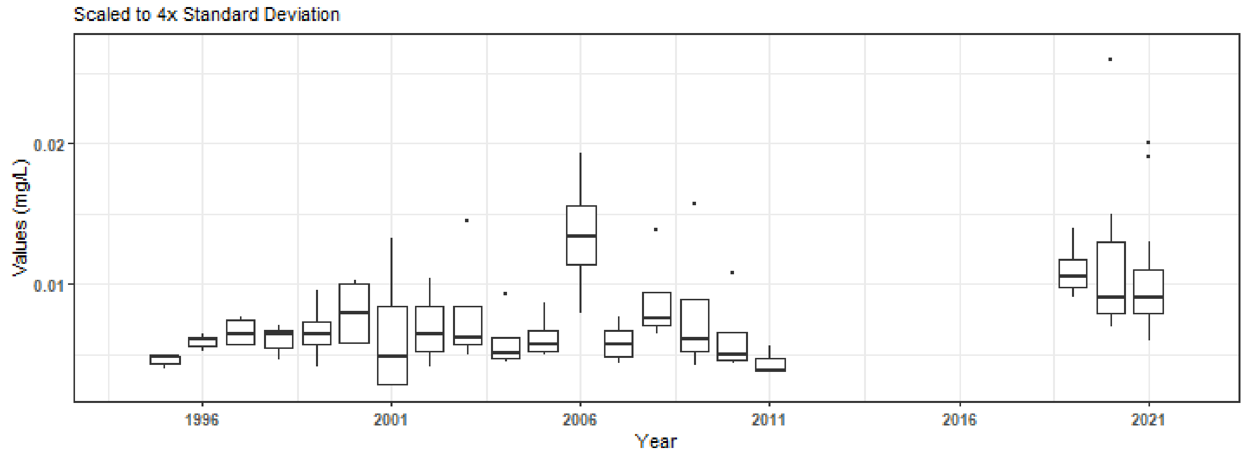
By Month

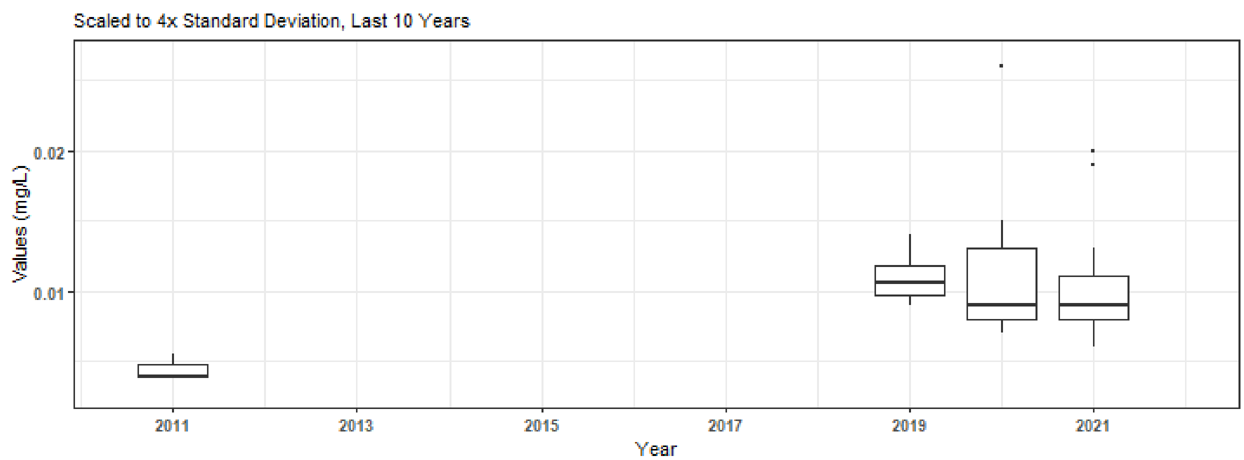


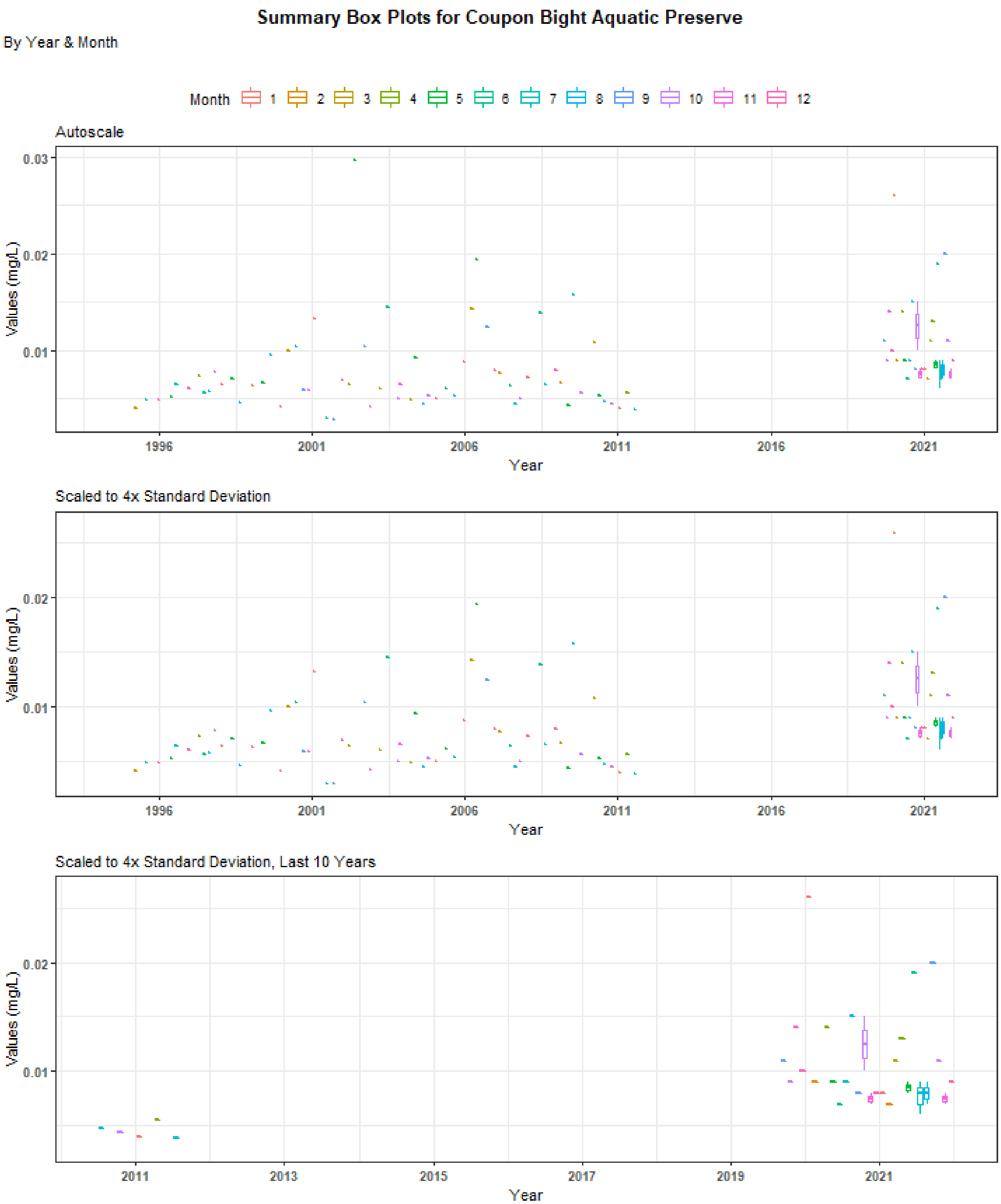
Summary Box Plots for Coupon Bight Aquatic Preserve

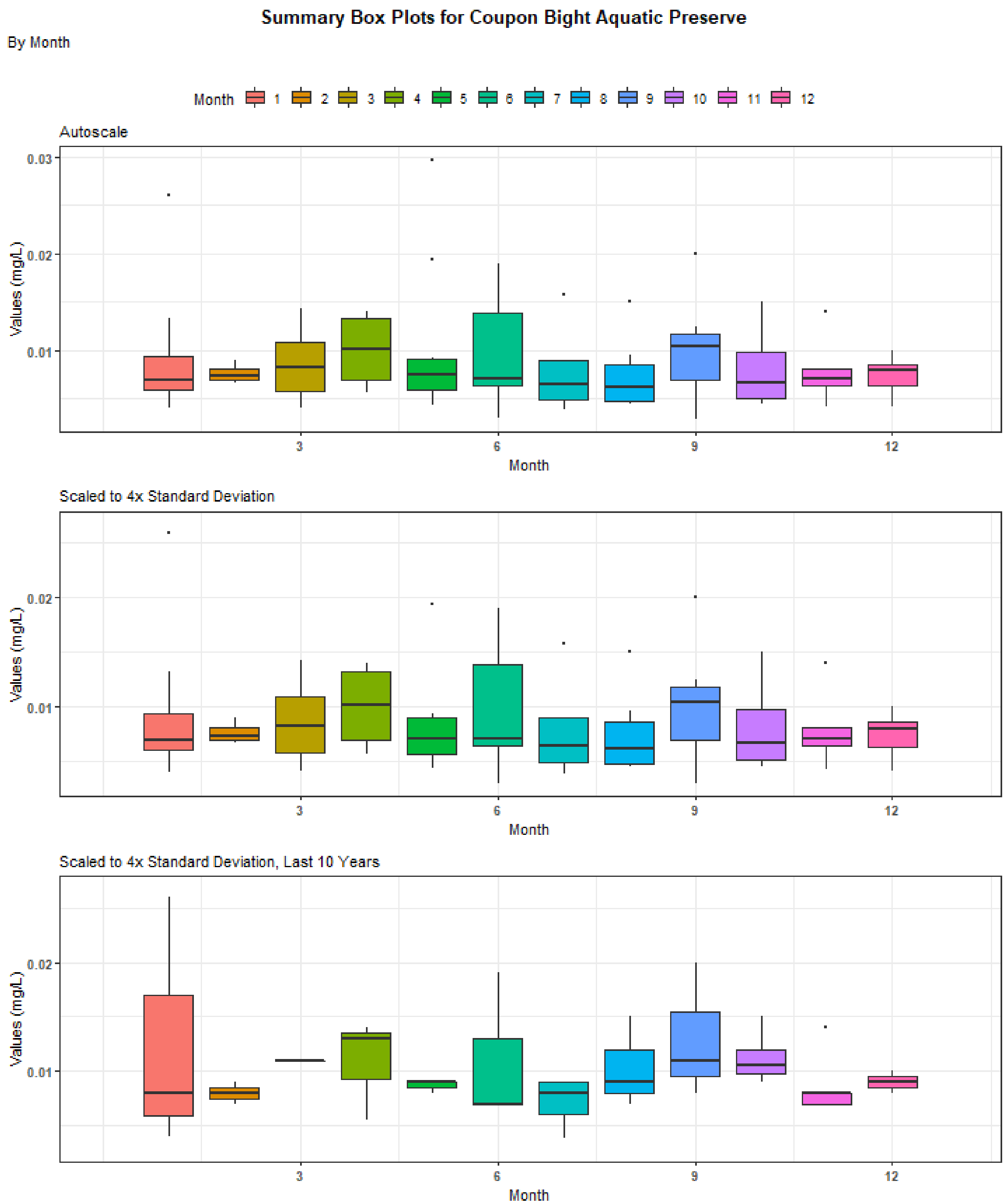
By Year








Summary Box Plots for Coupon Bight Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.03
0.02
0.01
1996 2001 2006 2011 2016 2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.02
0.01
1996 2001 2006 2011 2016 2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.02
0.01
2011 2013 2015 2017 2019 2021
Year

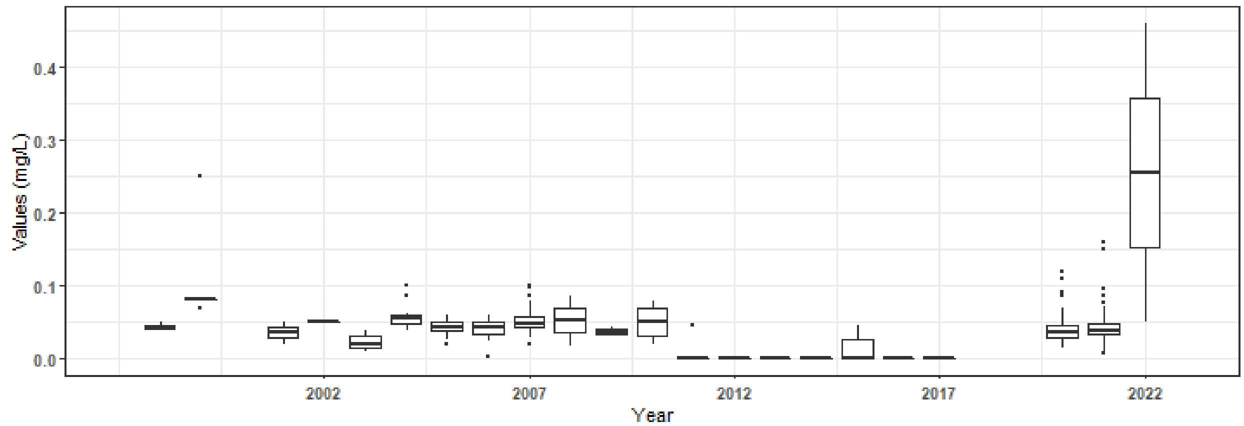


Summary Box Plots for Coupon Bight Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.03
0.02
0.01
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.02
0.01
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.02
0.01
3
6
9
12
Month

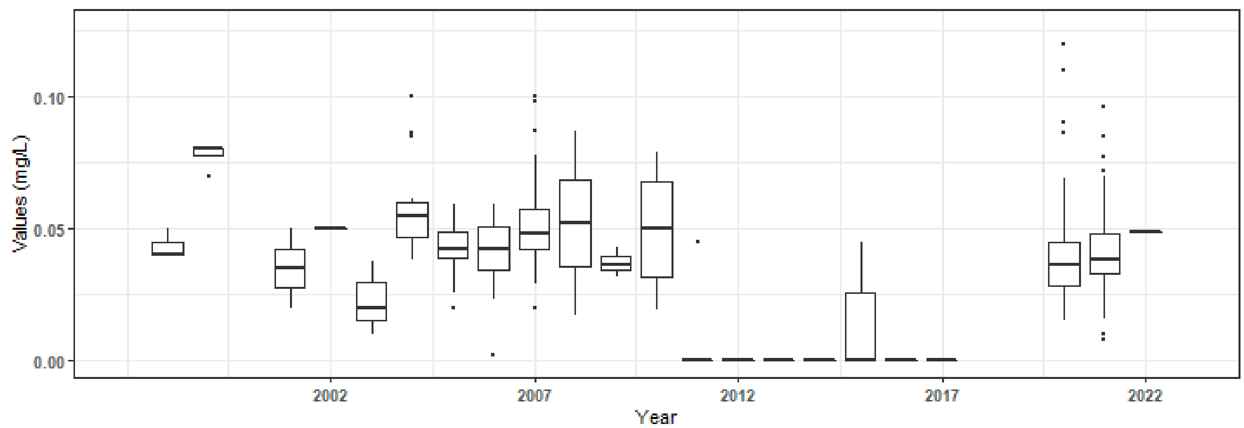
Summary Box Plots for Estero Bay Aquatic Preserve

By Year

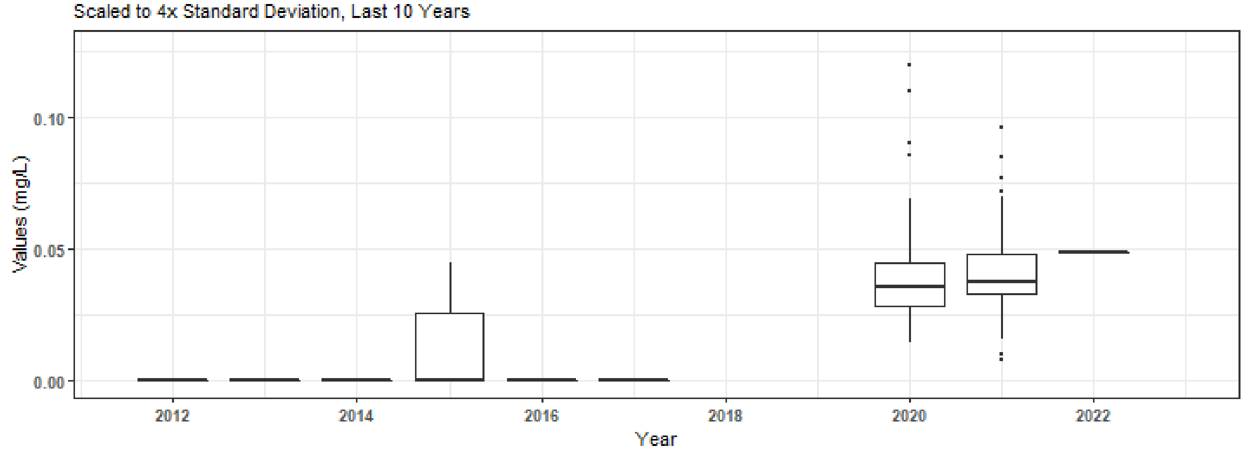
Autoscale



Scaled to 4x Standard Deviation

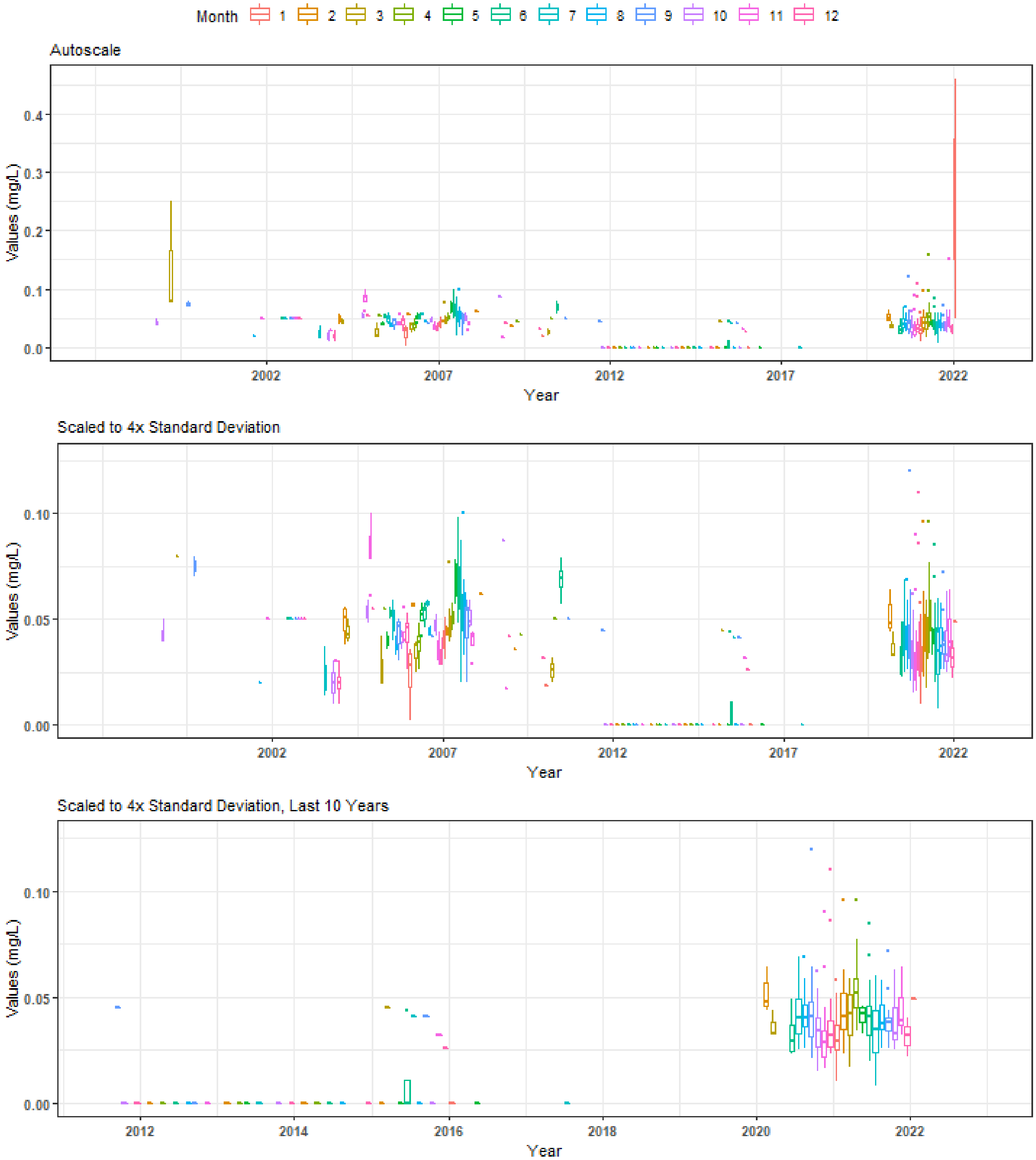


Scaled to 4x Standard Deviation, Last 10 Years



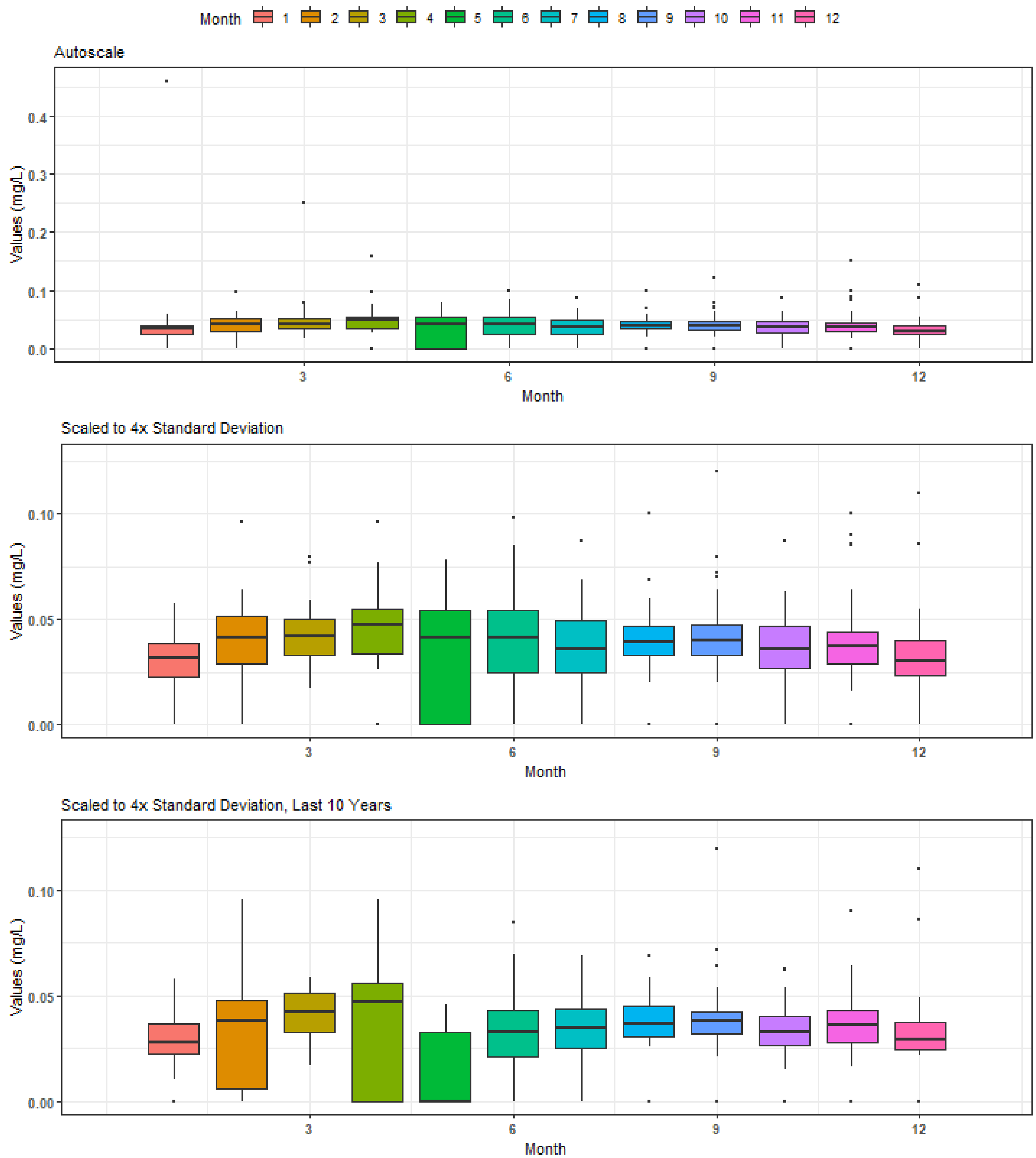
Summary Box Plots for Estero Bay Aquatic Preserve

By Year & Month



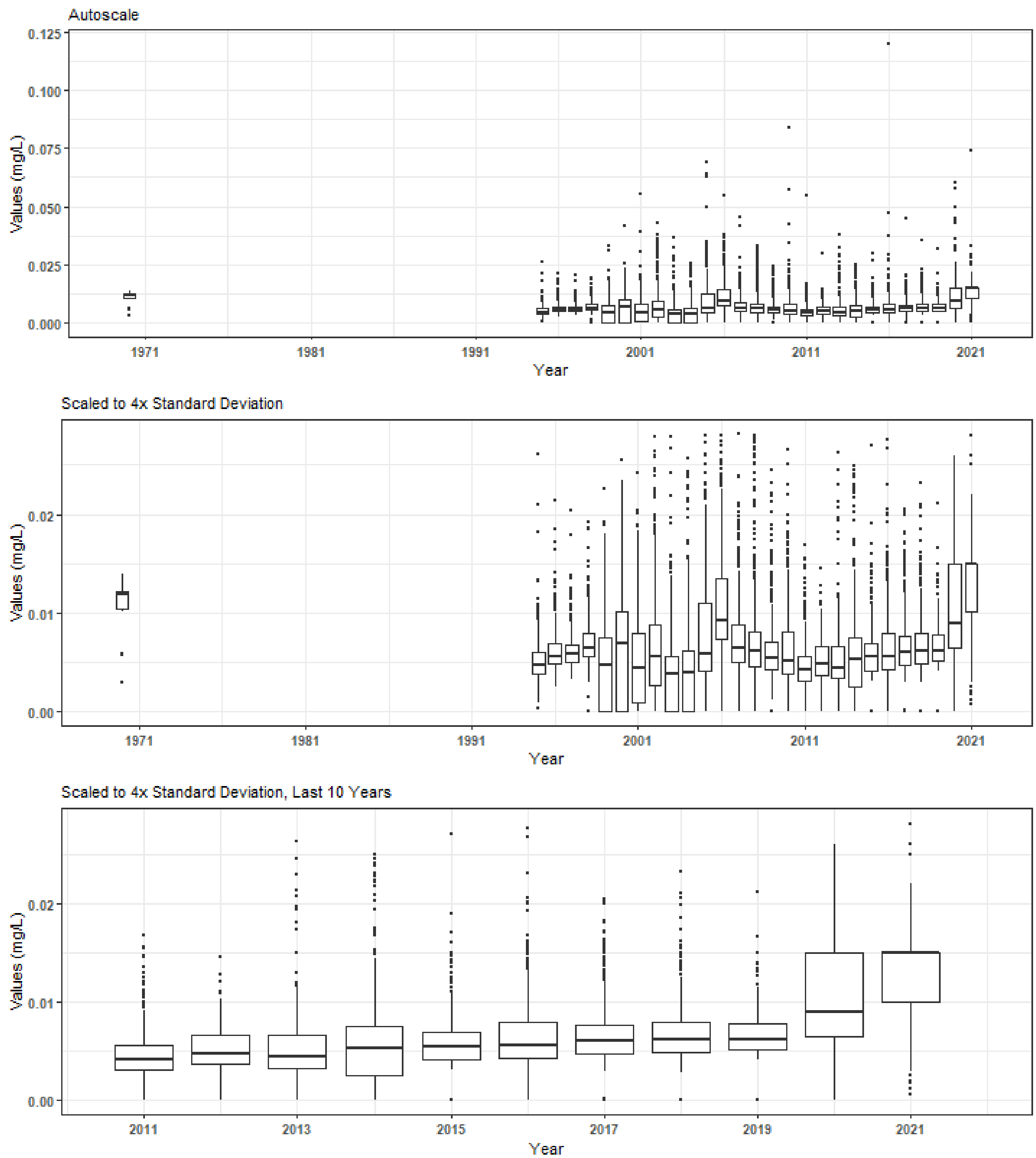
Summary Box Plots for Estero Bay Aquatic Preserve

By Month



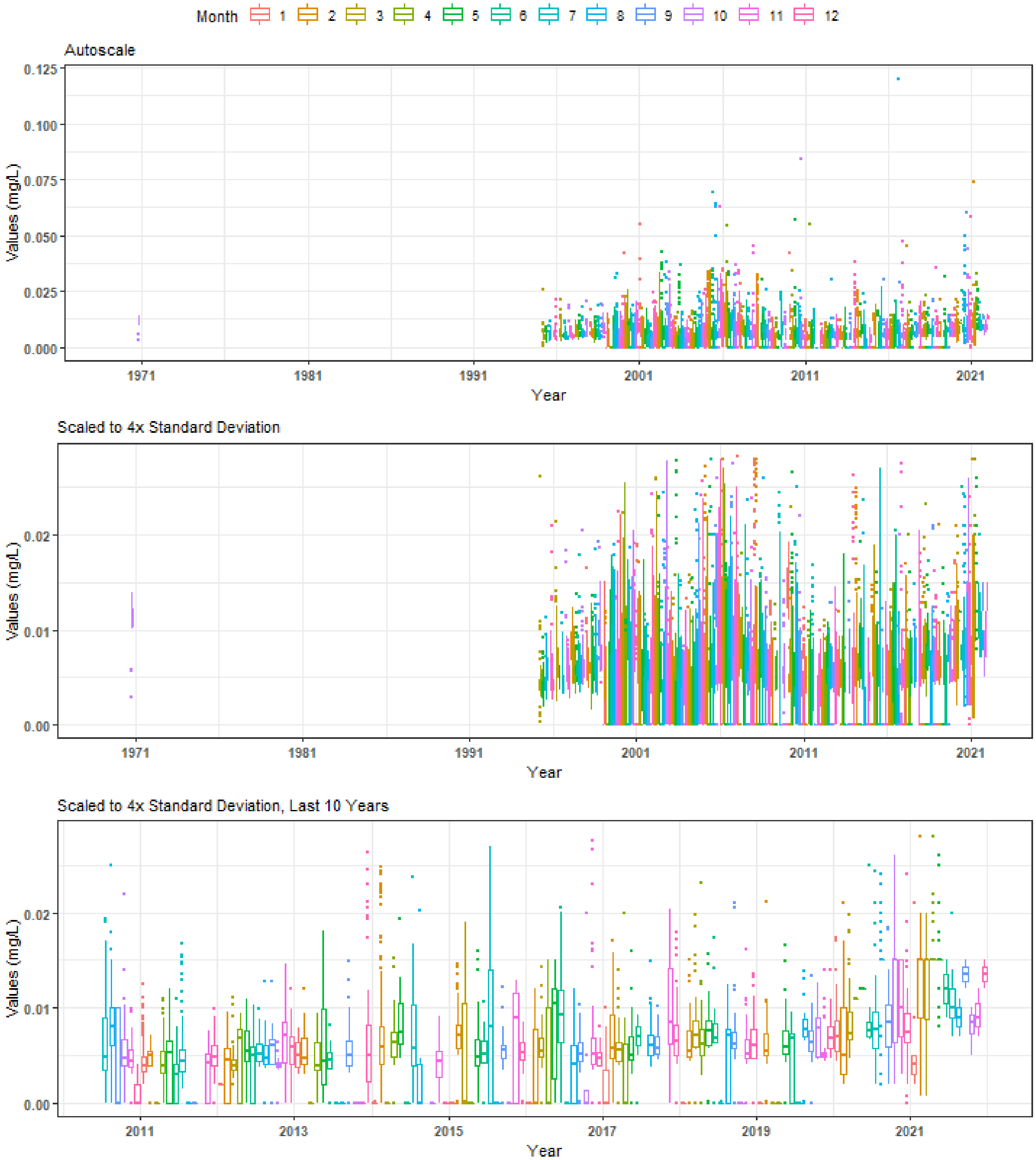
Summary Box Plots for Florida Keys National Marine Sanctuary

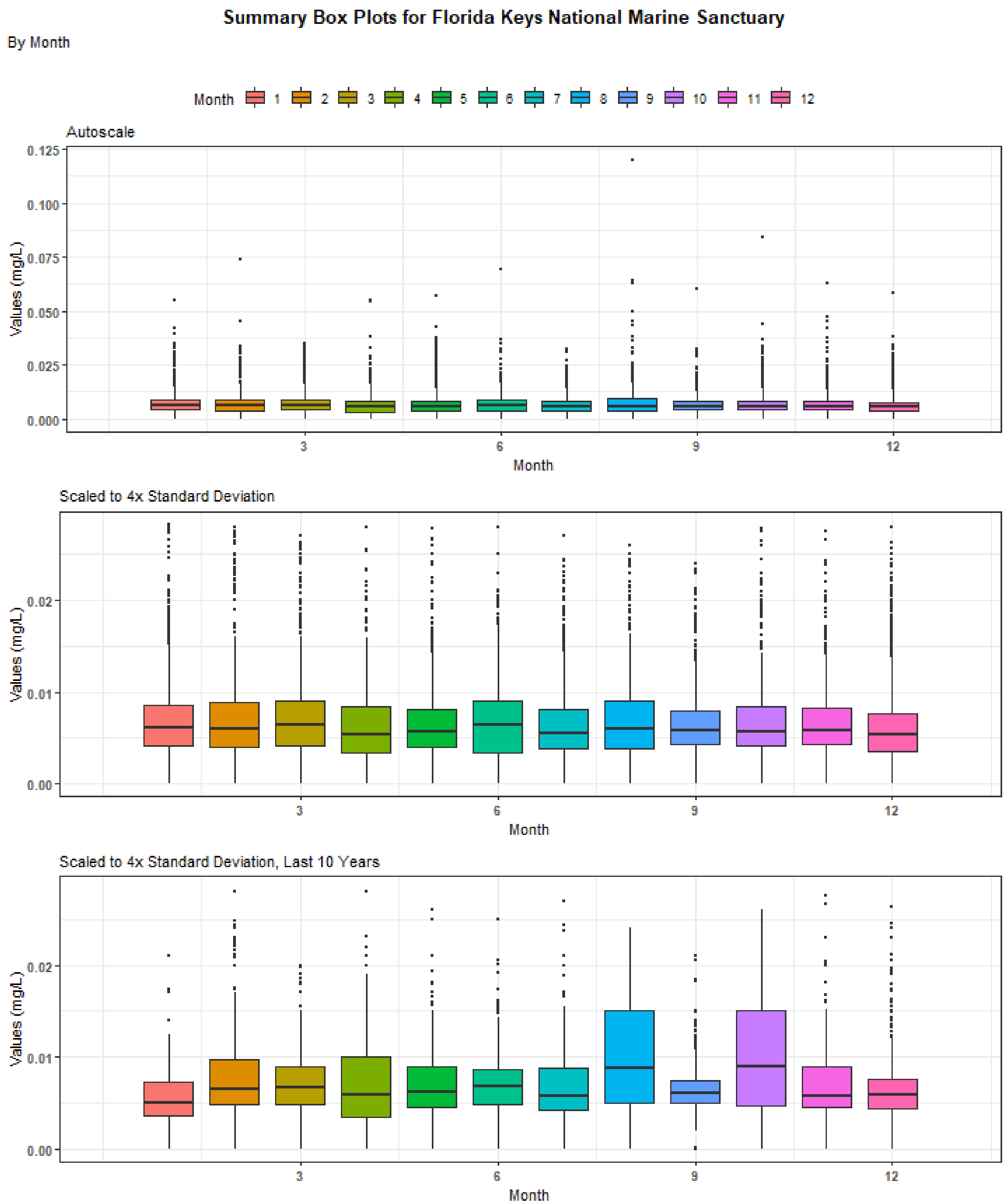
By Year



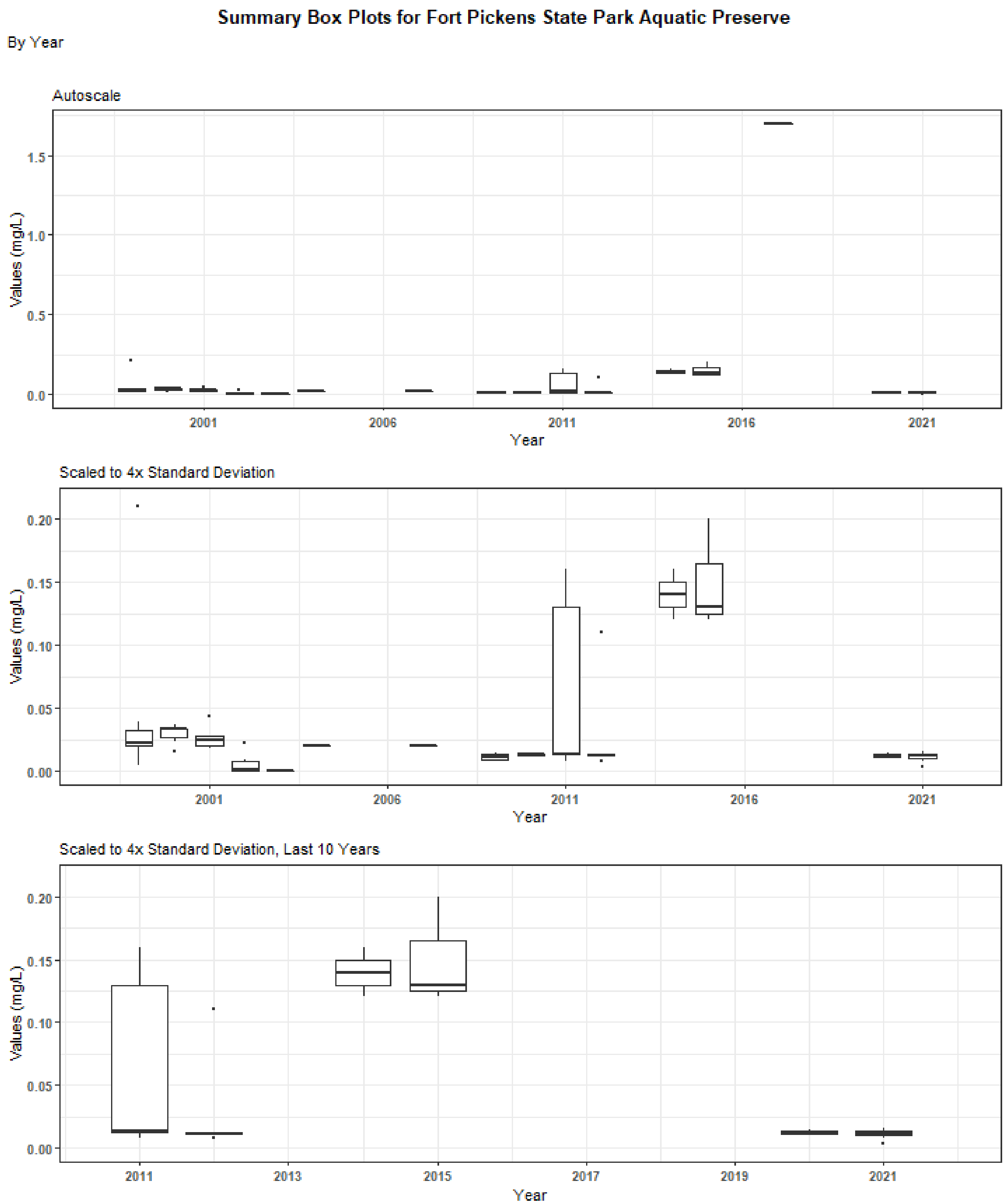
Summary Box Plots for Florida Keys National Marine Sanctuary

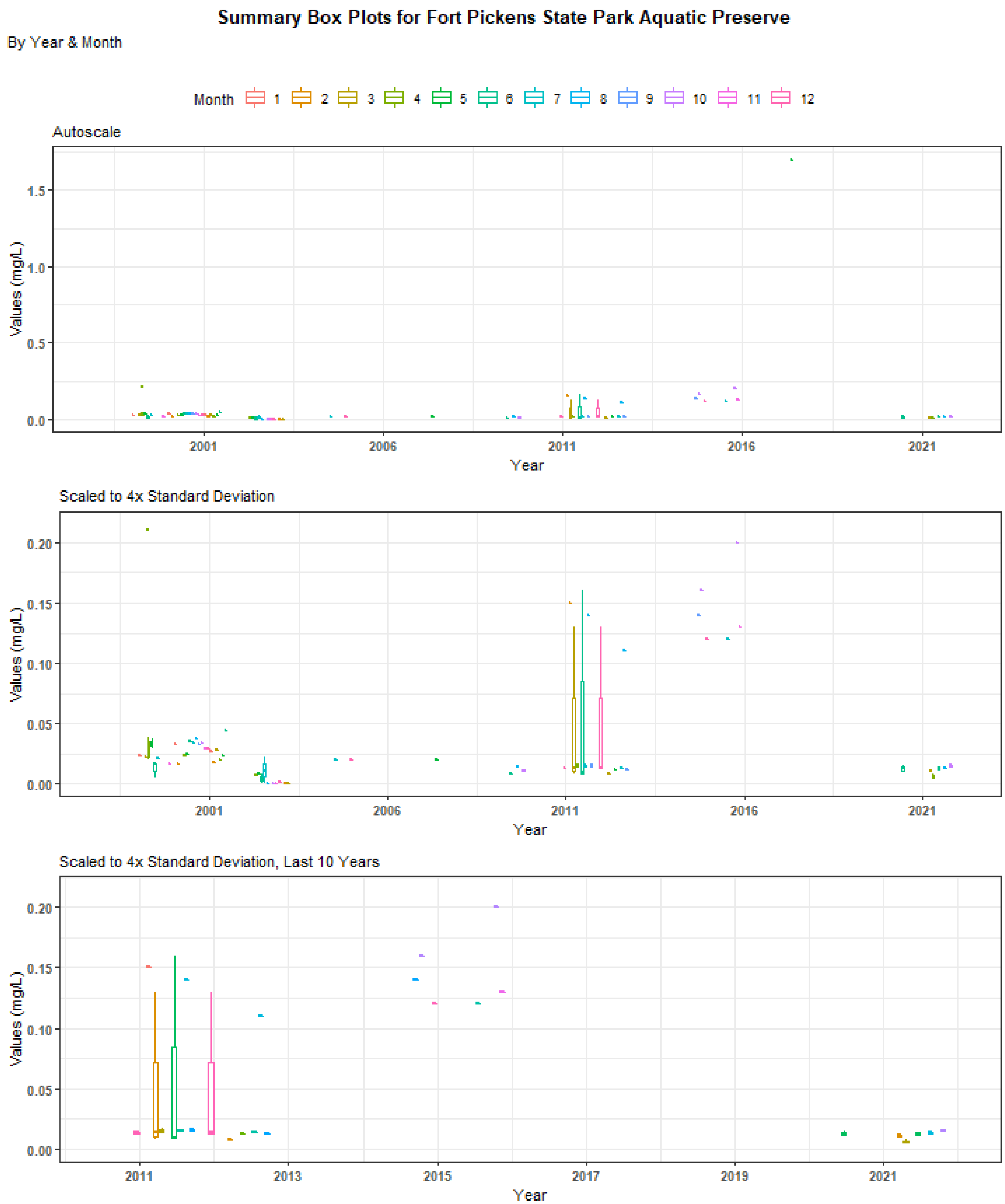
By Year & Month

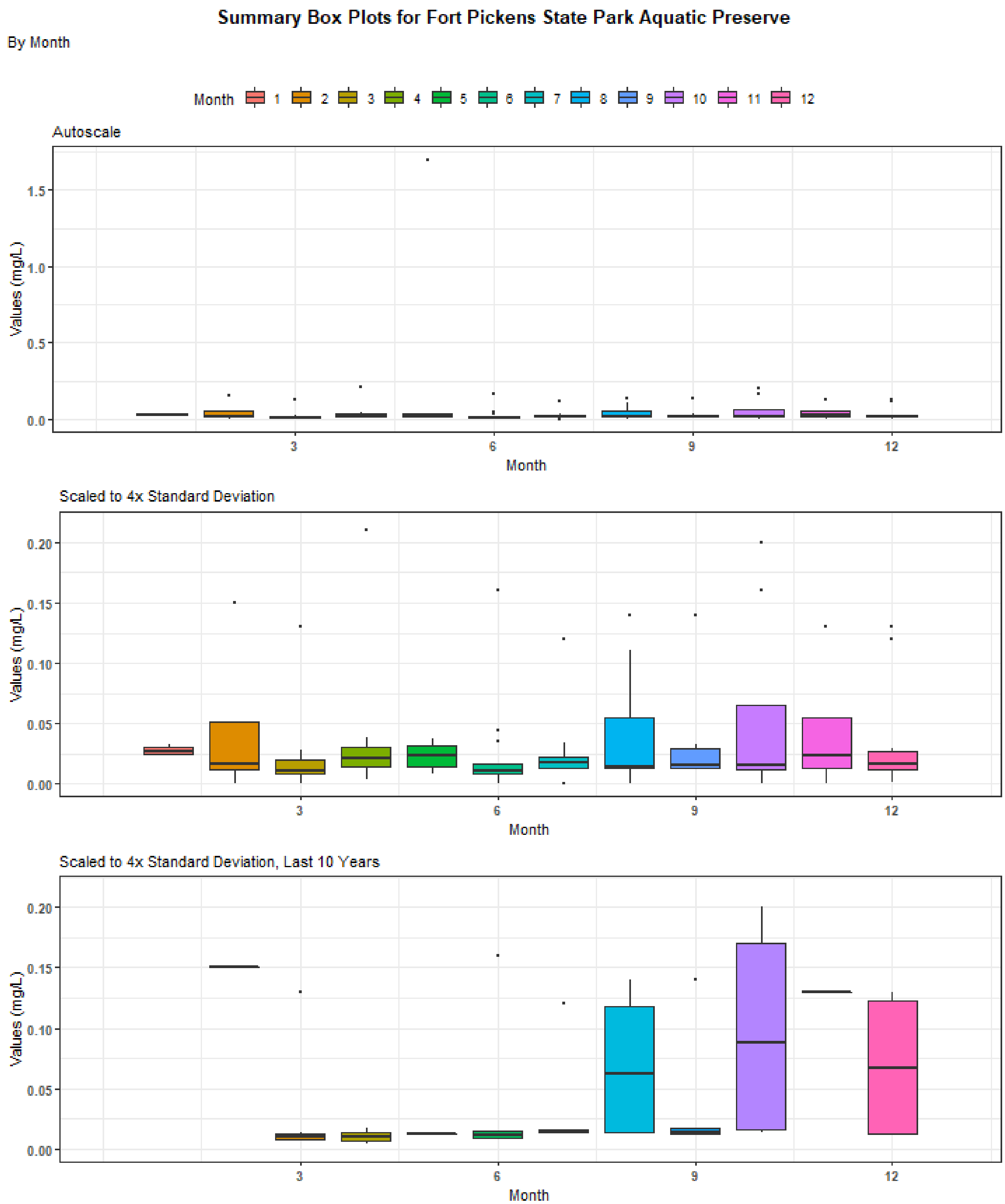




Summary Box Plots for Florida Keys National Marine Sanctuary
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.125
0.100
0.075
0.050
0.025
0.000
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.02
0.01
0.00
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.02
0.01
0.00
3
6
9
12
Month


Summary Box Plots for Fort Pickens State Park Aquatic Preserve
By Year
Autoscale
Values (mg/L)
1.5
1.0
0.5
0.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.20
0.15
0.10
0.05
0.00
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.20
0.15
0.10
0.05
0.00
2011
2013
2015
2017
2019
2021
Year

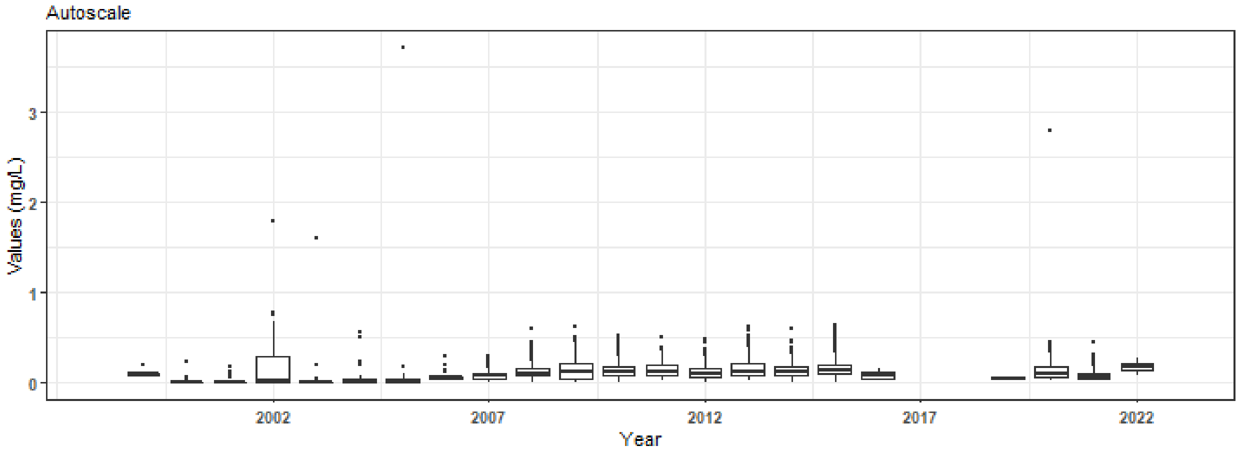

Summary Box Plots for Fort Pickens State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
1.5
1.0
0.5
0.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.20
0.15
0.10
0.05
0.00
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.20
0.15
0.10
0.05
0.00
2011
2013
2015
2017
2019
2021
Year

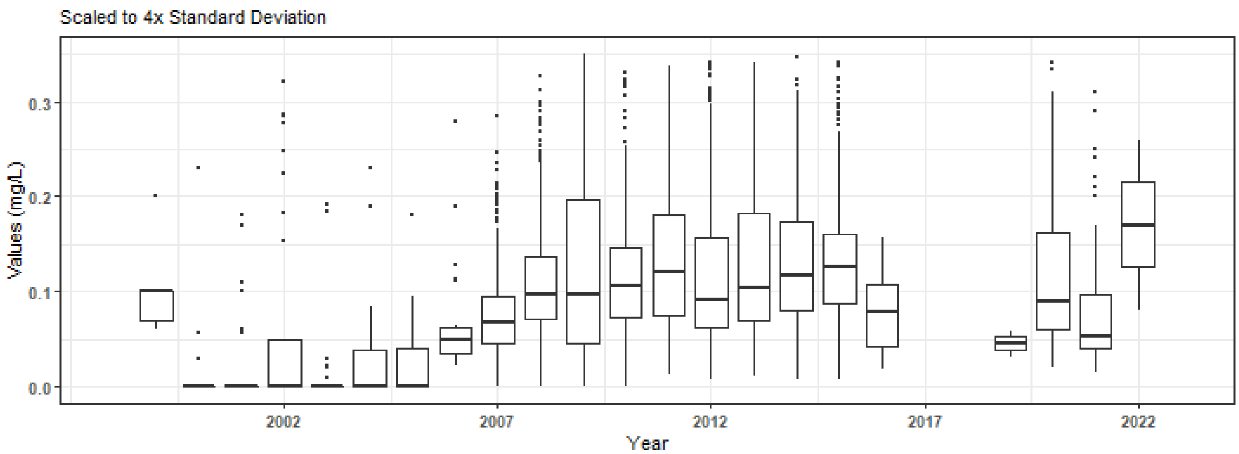


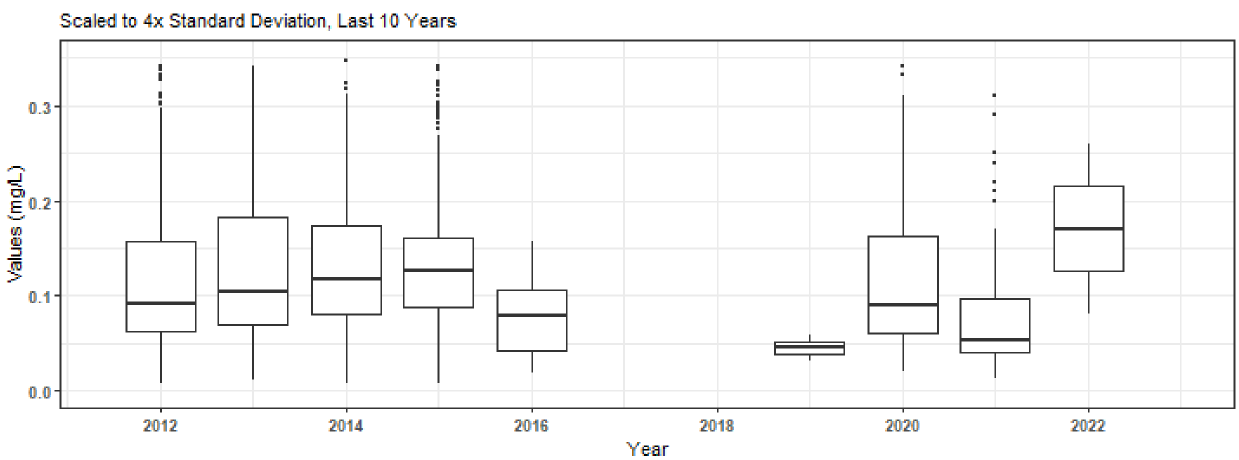
Summary Box Plots for Fort Pickens State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.5
1.0
1.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
3
6
9
12
Month

Summary Box Plots for Gasparilla Sound-Charlotte Harbor Aquatic Preserve

By Year

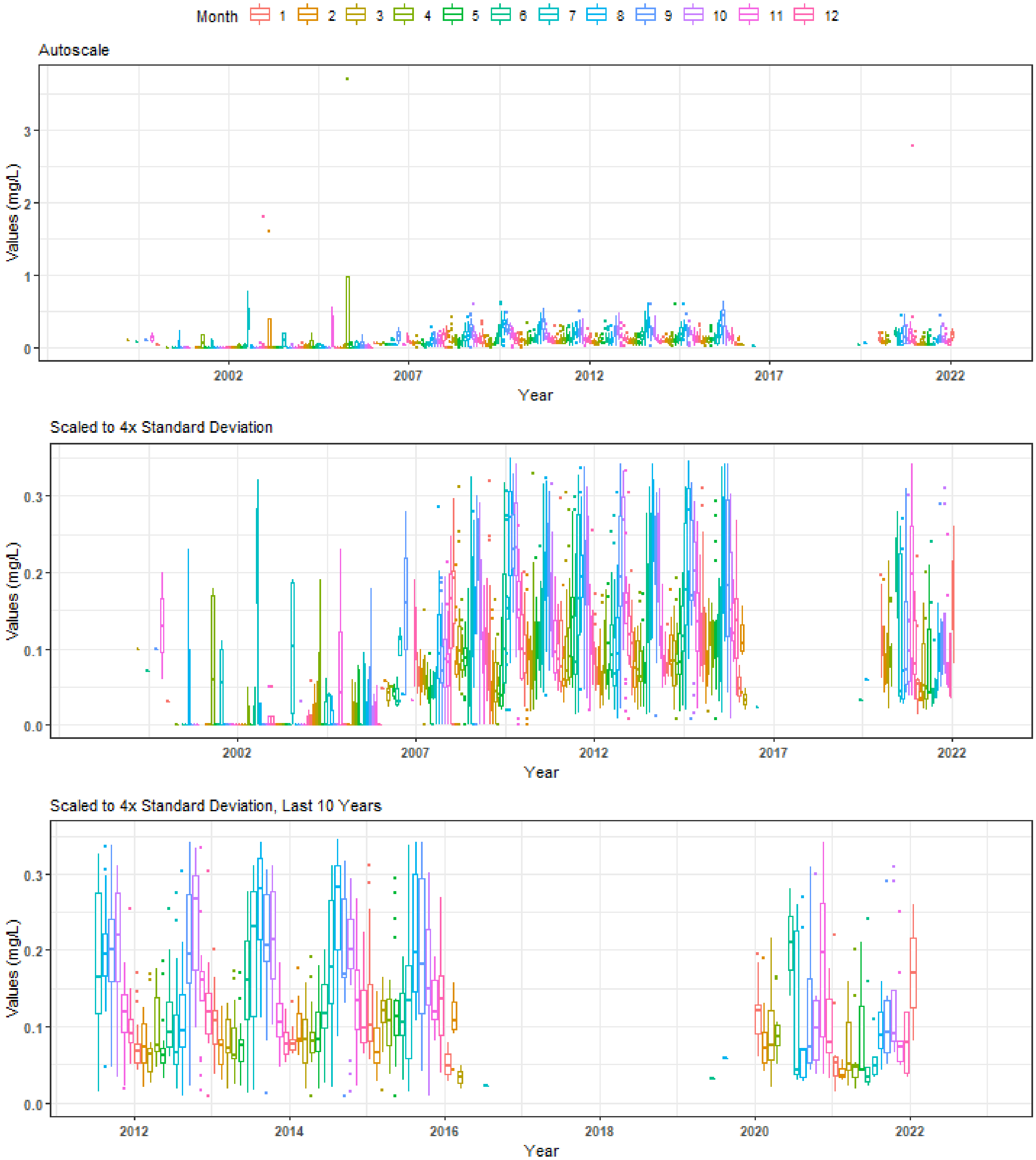






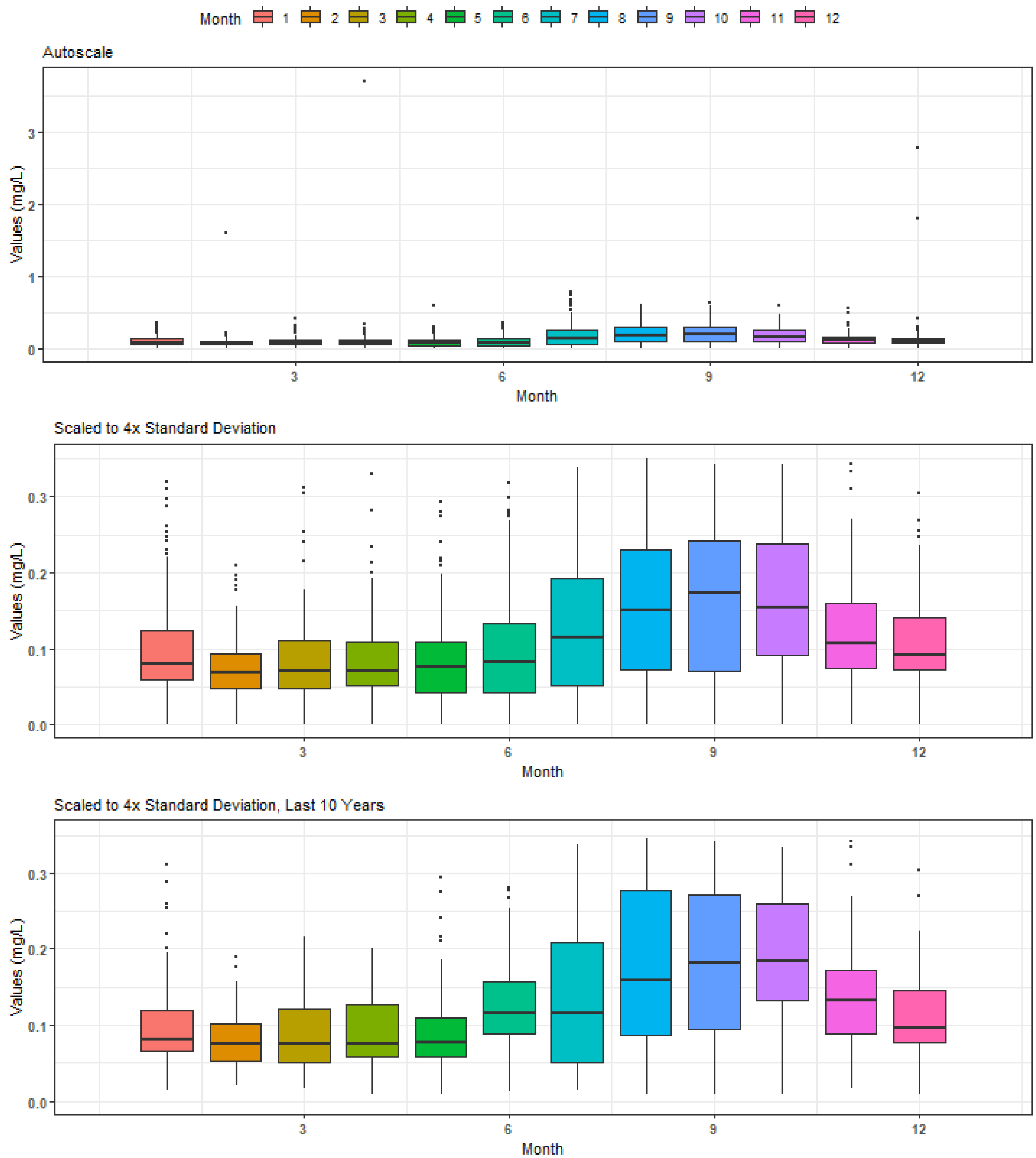
Summary Box Plots for Gasparilla Sound-Charlotte Harbor Aquatic Preserve

By Year & Month



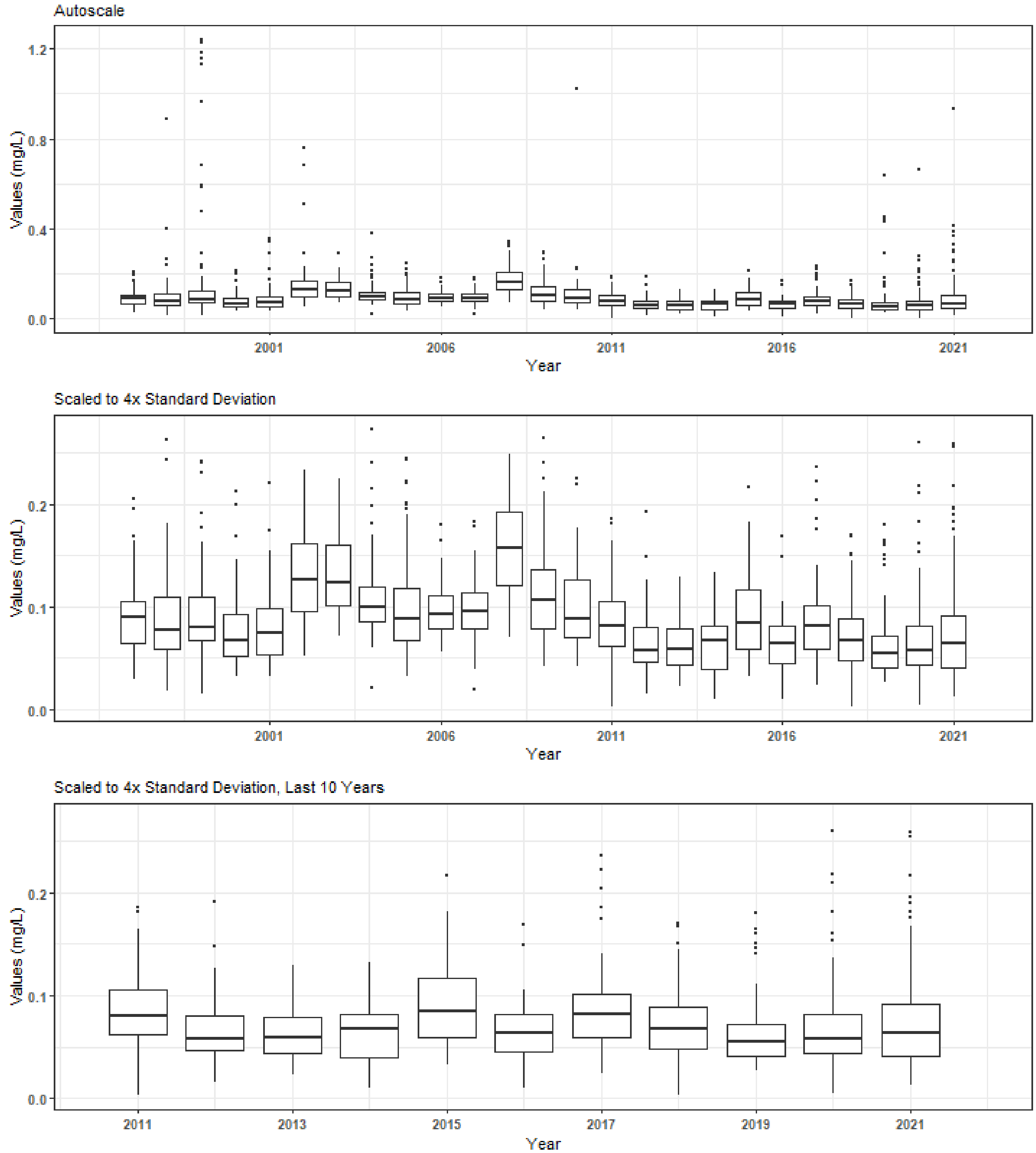
Summary Box Plots for Gasparilla Sound-Charlotte Harbor Aquatic Preserve

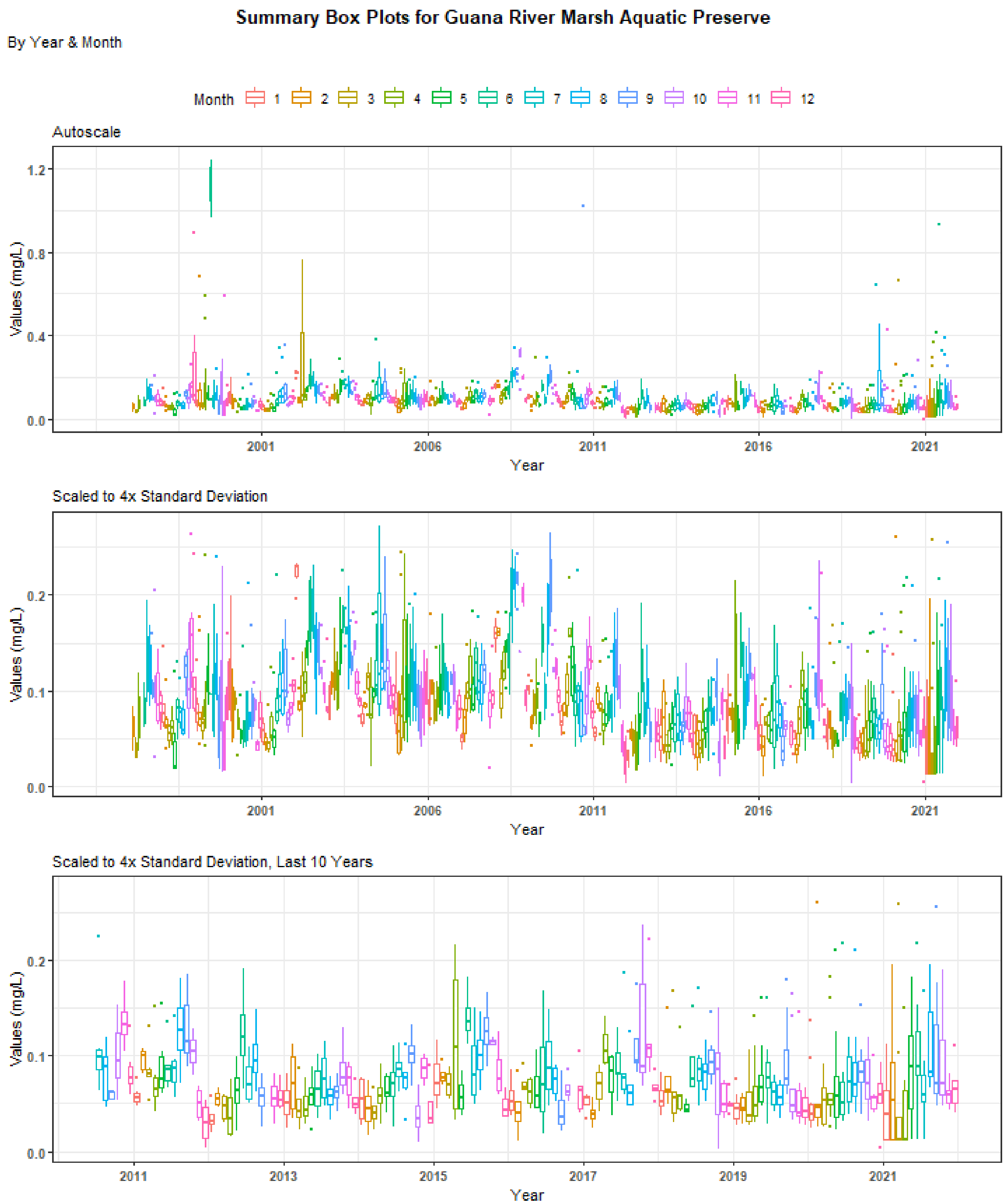
By Month

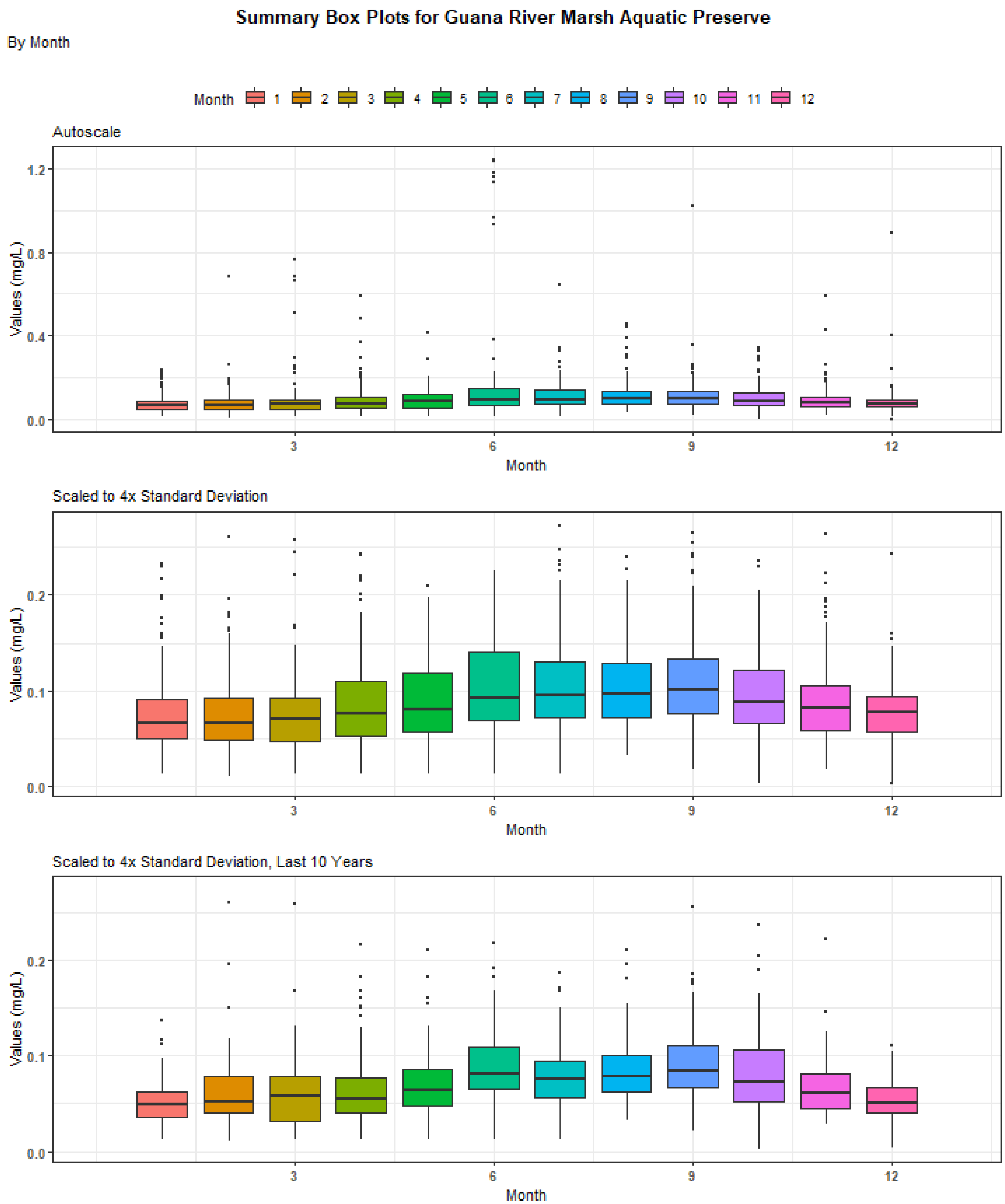


Summary Box Plots for Guana River Marsh Aquatic Preserve

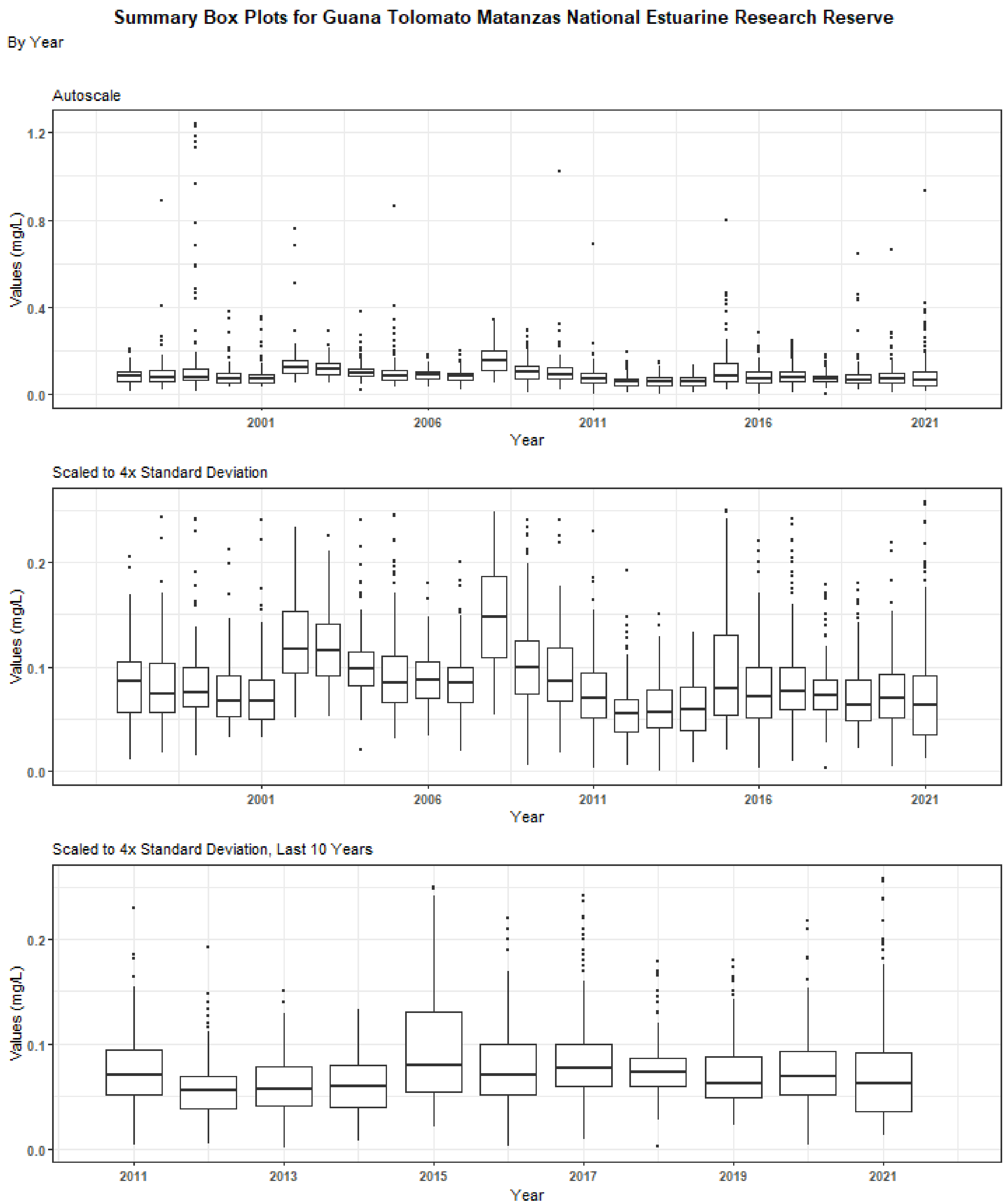
By Year

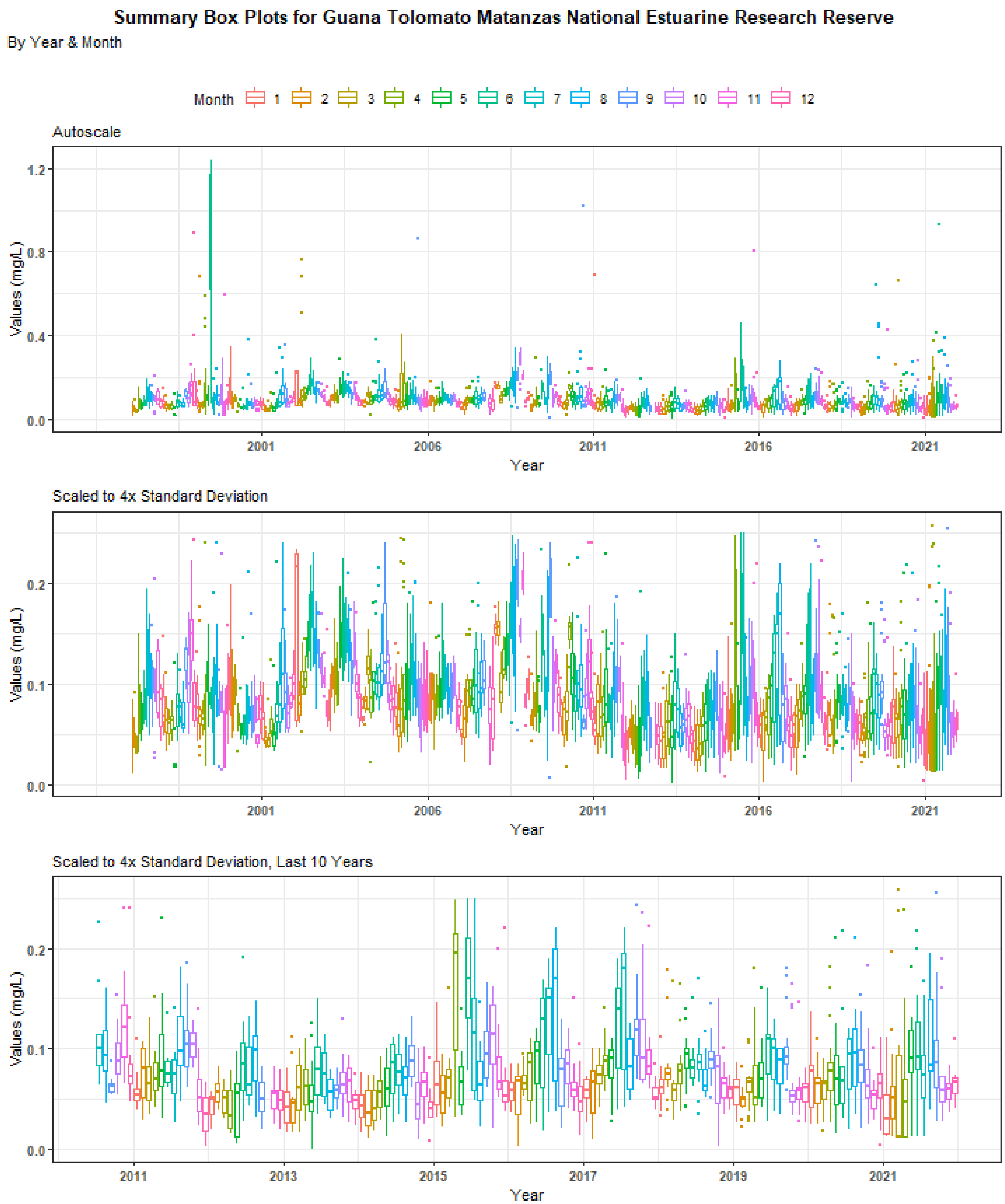


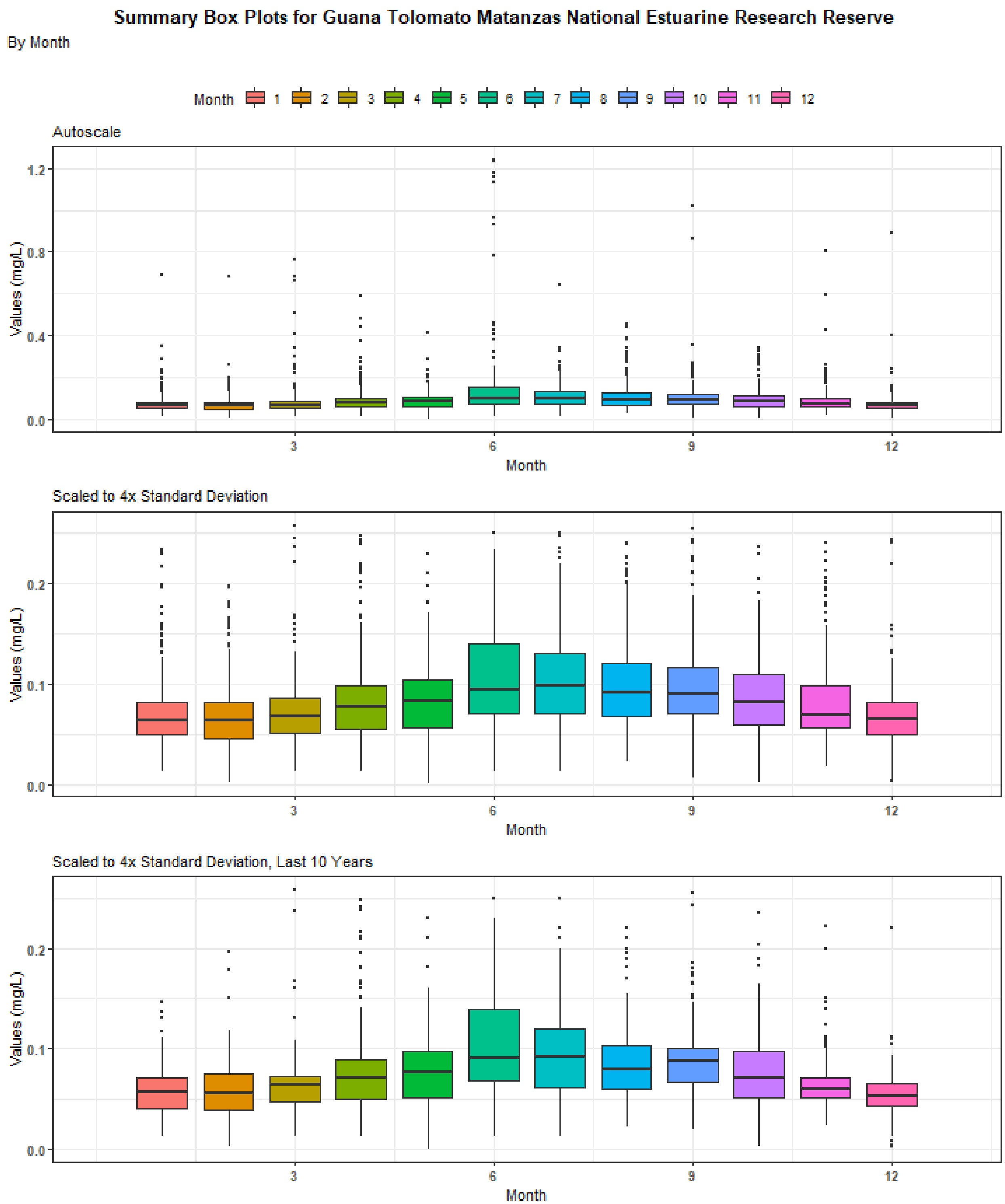

Summary Box Plots for Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.4
0.8
1.2
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
2011
2013
2015
2017
2019
2021
Year



Summary Box Plots for Guana River Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
1.2
0.8
0.4
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.2
0.1
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.2
0.1
0.0
3
6
9
12
Month

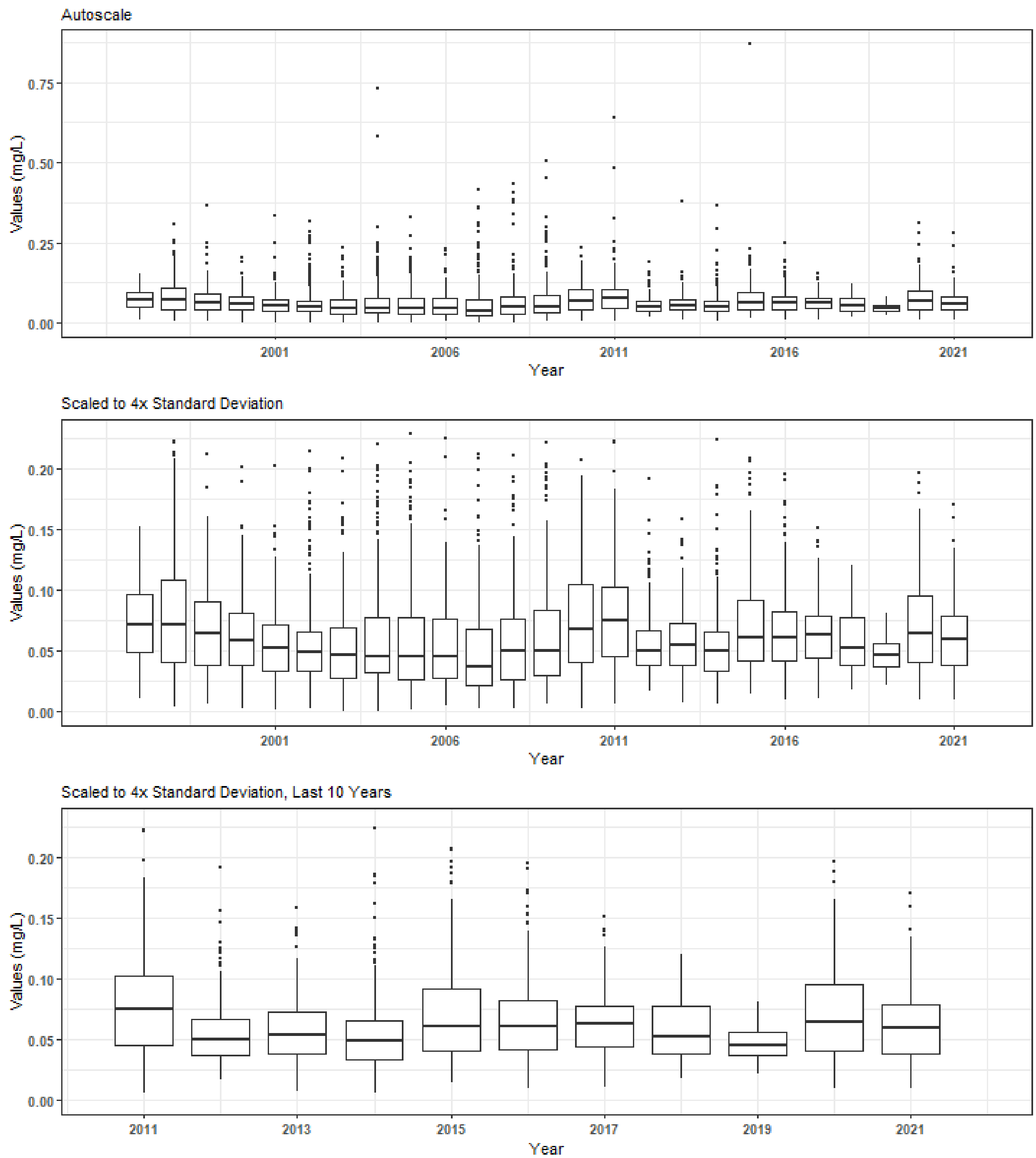

Summary Box Plots for Guana Tolomato Matanzas National Estuarine Research Reserve
By Year
Autoscale
Values (mg/L)
0.0
0.4
0.8
1.2
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
2011
2013
2015
2017
2019
2021
Year


Summary Box Plots for Guana Tolomato Matanzas National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
1.2
0.8
0.4
0.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.2
0.1
0.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.2
0.1
0.0
2011
2013
2015
2017
2019
2021
Year


Summary Box Plots for Guana Tolomato Matanzas National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
1.2
0.8
0.4
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.2
0.1
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.2
0.1
0.0
3
6
9
12
Month

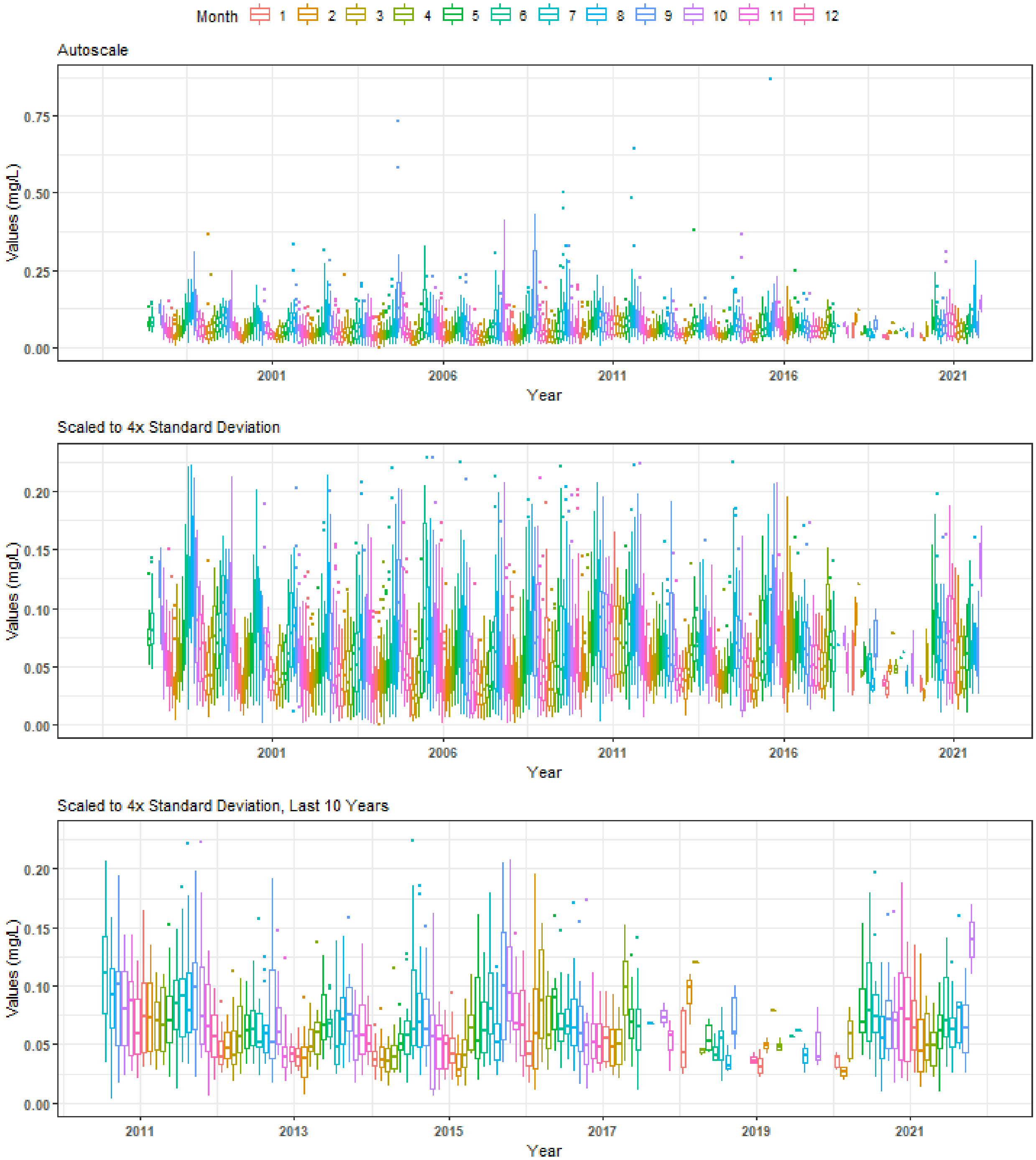
Summary Box Plots for Indian River-Malabar to Vero Beach Aquatic Preserve

By Year



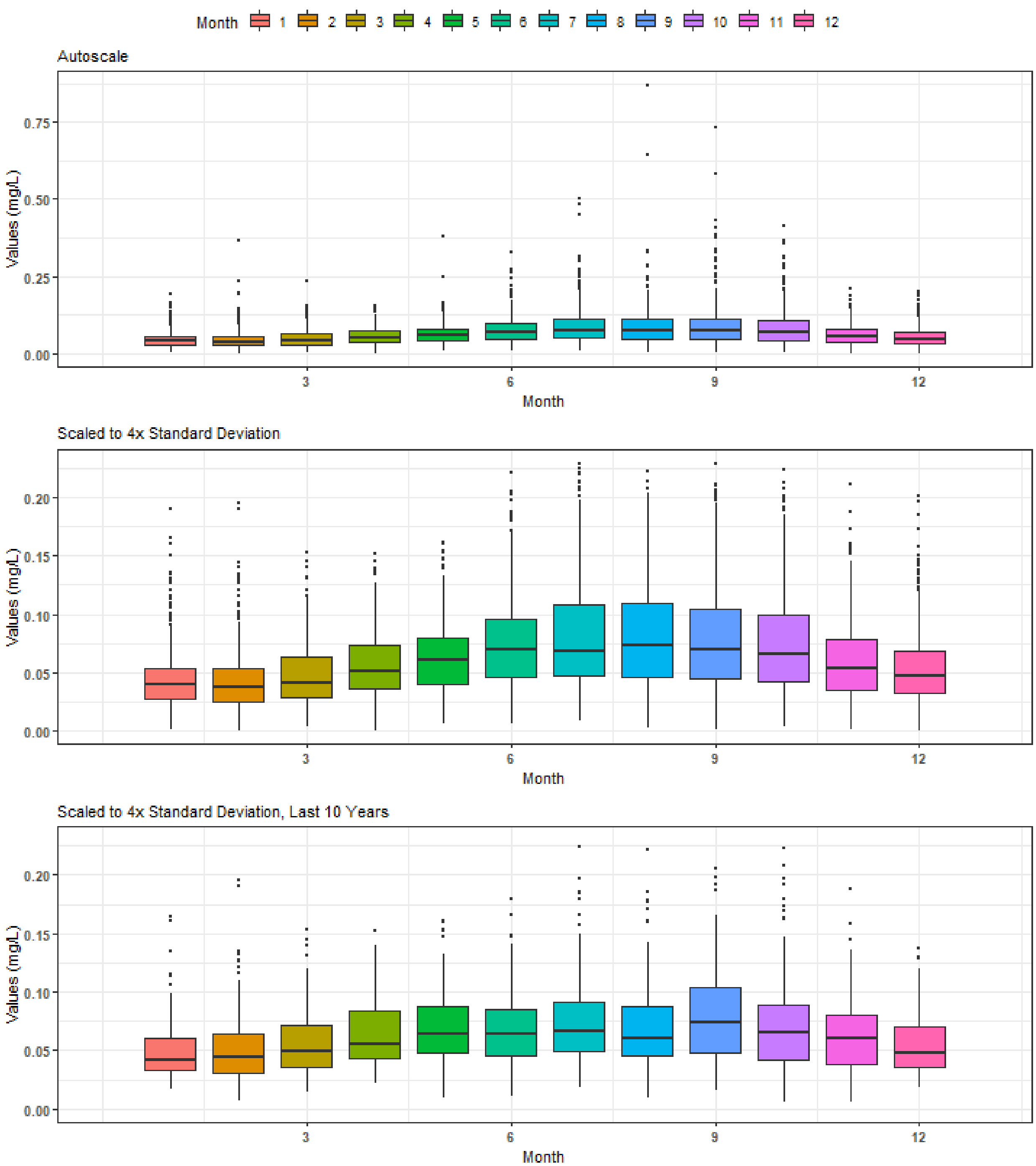
Summary Box Plots for Indian River-Malabar to Vero Beach Aquatic Preserve

By Year & Month



Summary Box Plots for Indian River-Malabar to Vero Beach Aquatic Preserve

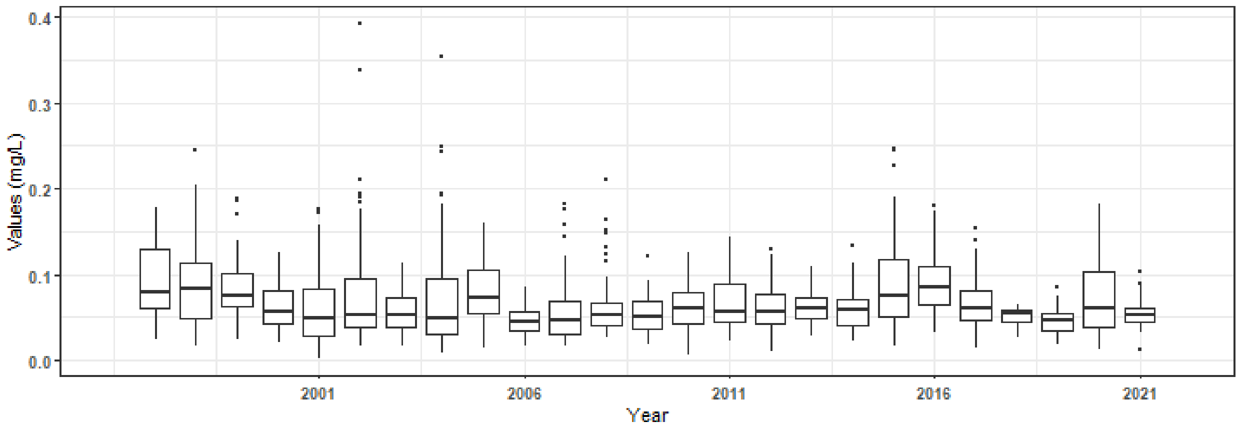
By Month



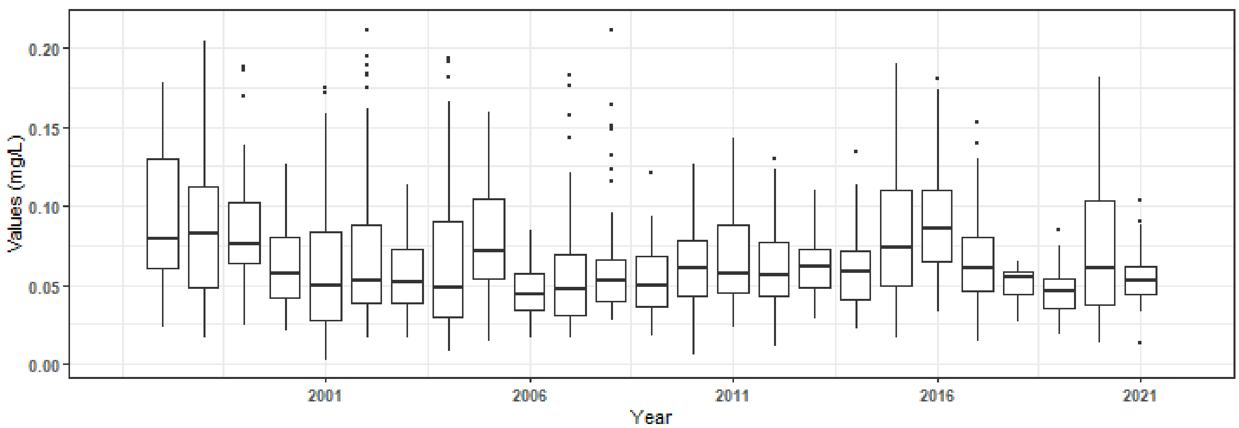
Summary Box Plots for Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

By Year

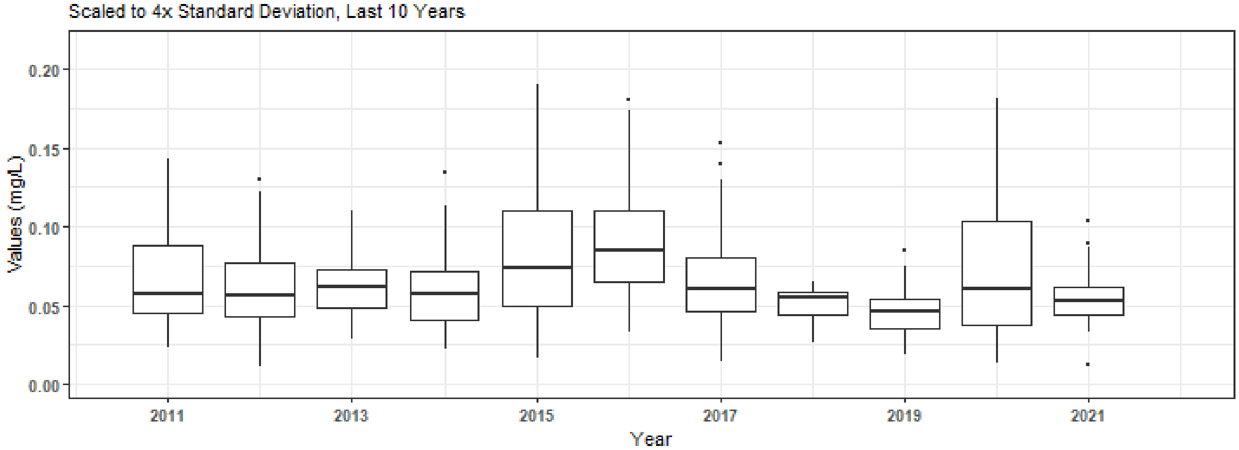
Autoscale

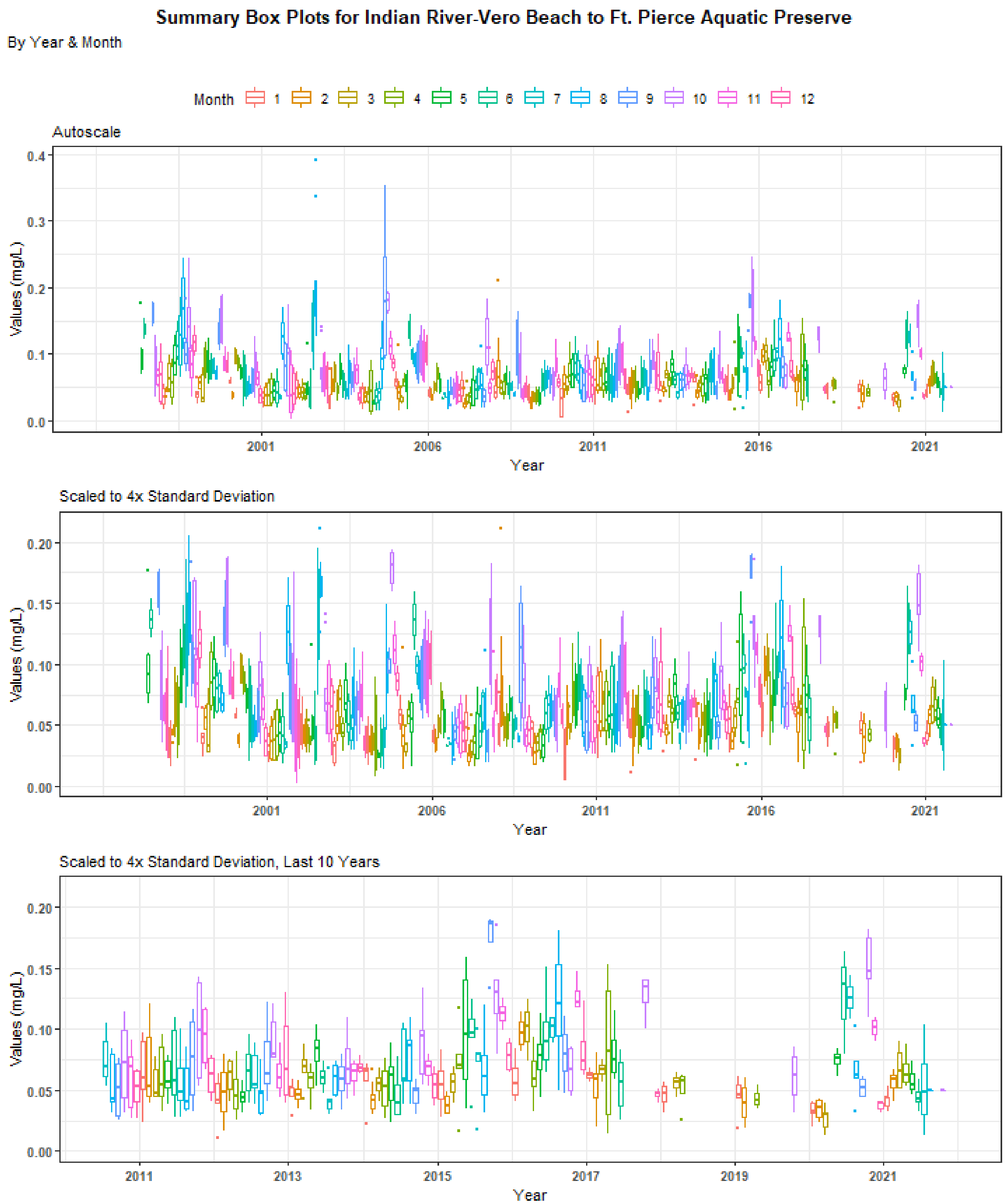


Scaled to 4x Standard Deviation

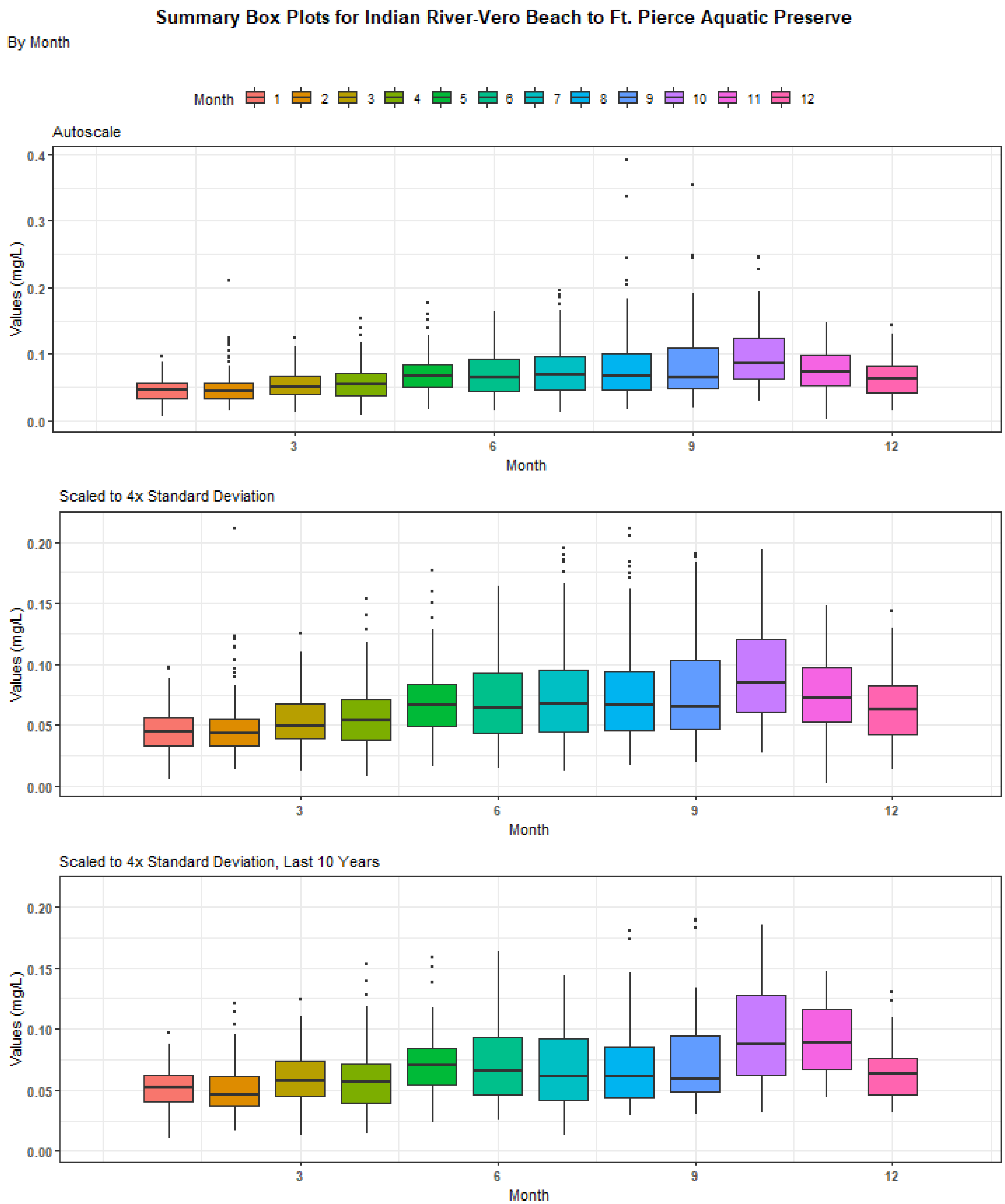


Scaled to 4x Standard Deviation, Last 10 Years





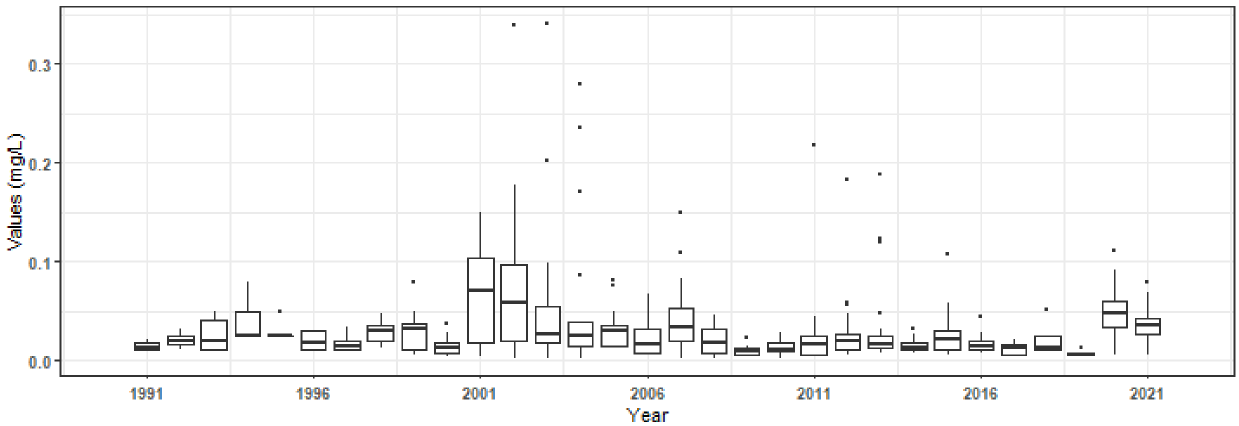
Summary Box Plots for Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
2011
2013
2015
2017
2019
2021
Year


Summary Box Plots for Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
3
6
9
12
Month

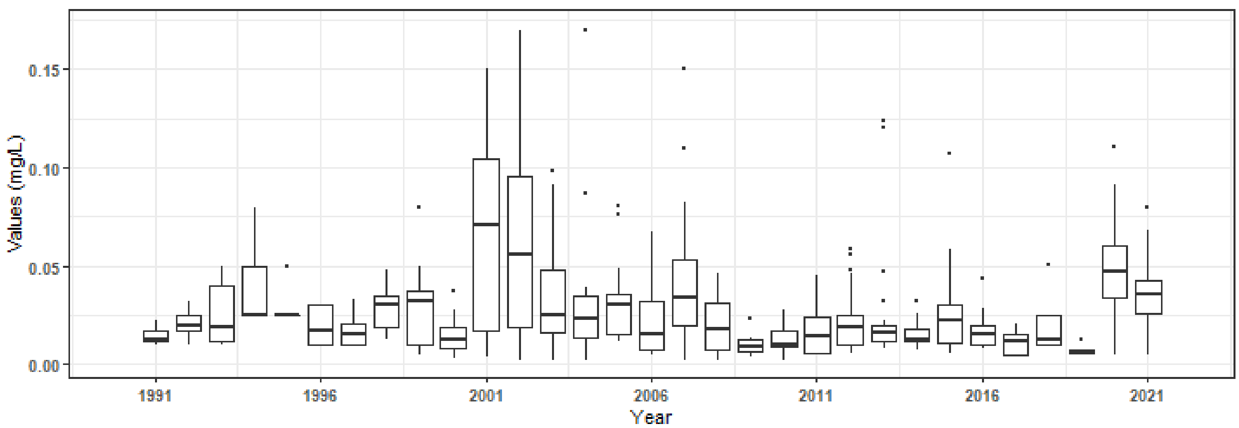
Summary Box Plots for Jensen Beach to Jupiter Inlet Aquatic Preserve

By Year

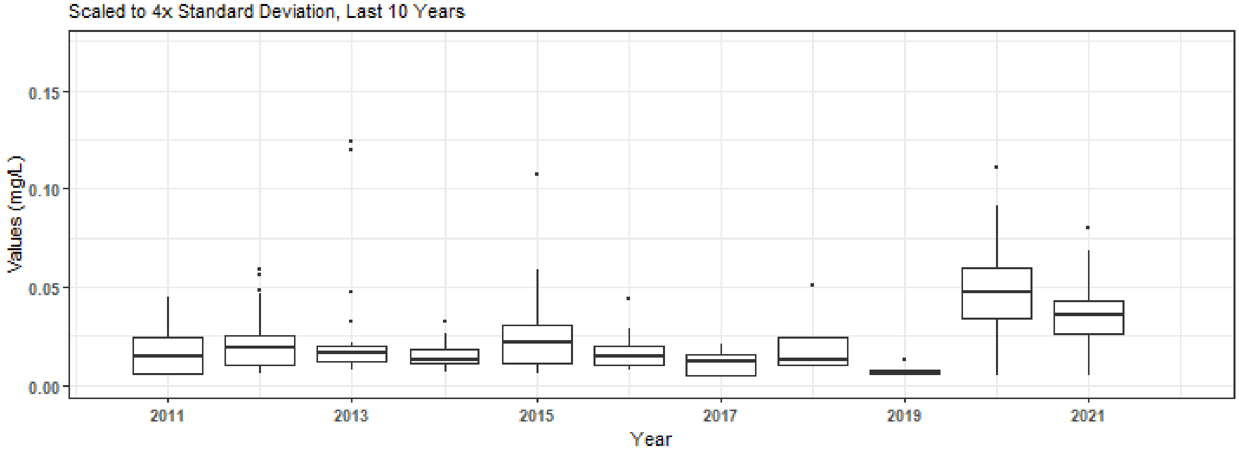
Autoscale

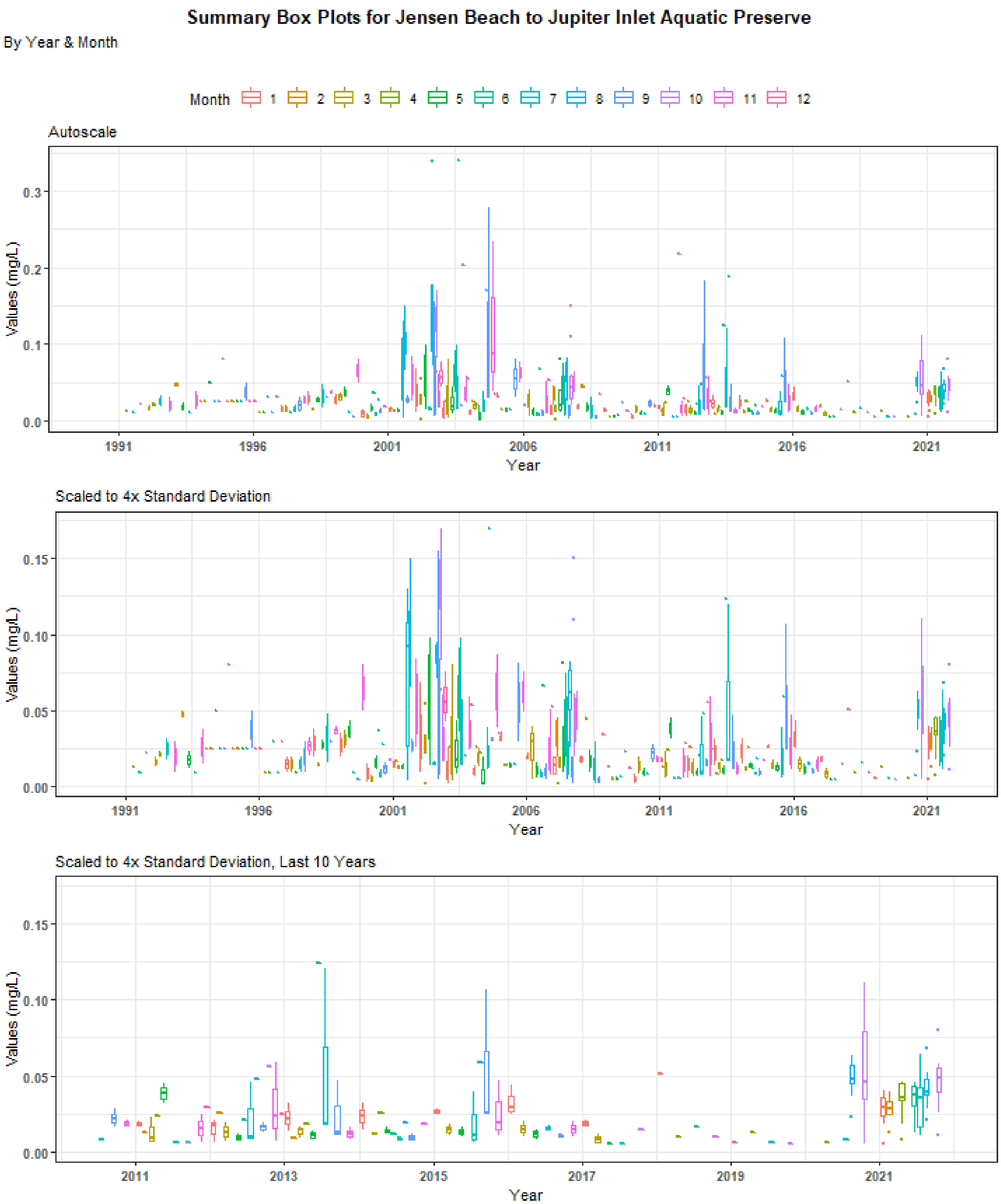


Scaled to 4x Standard Deviation



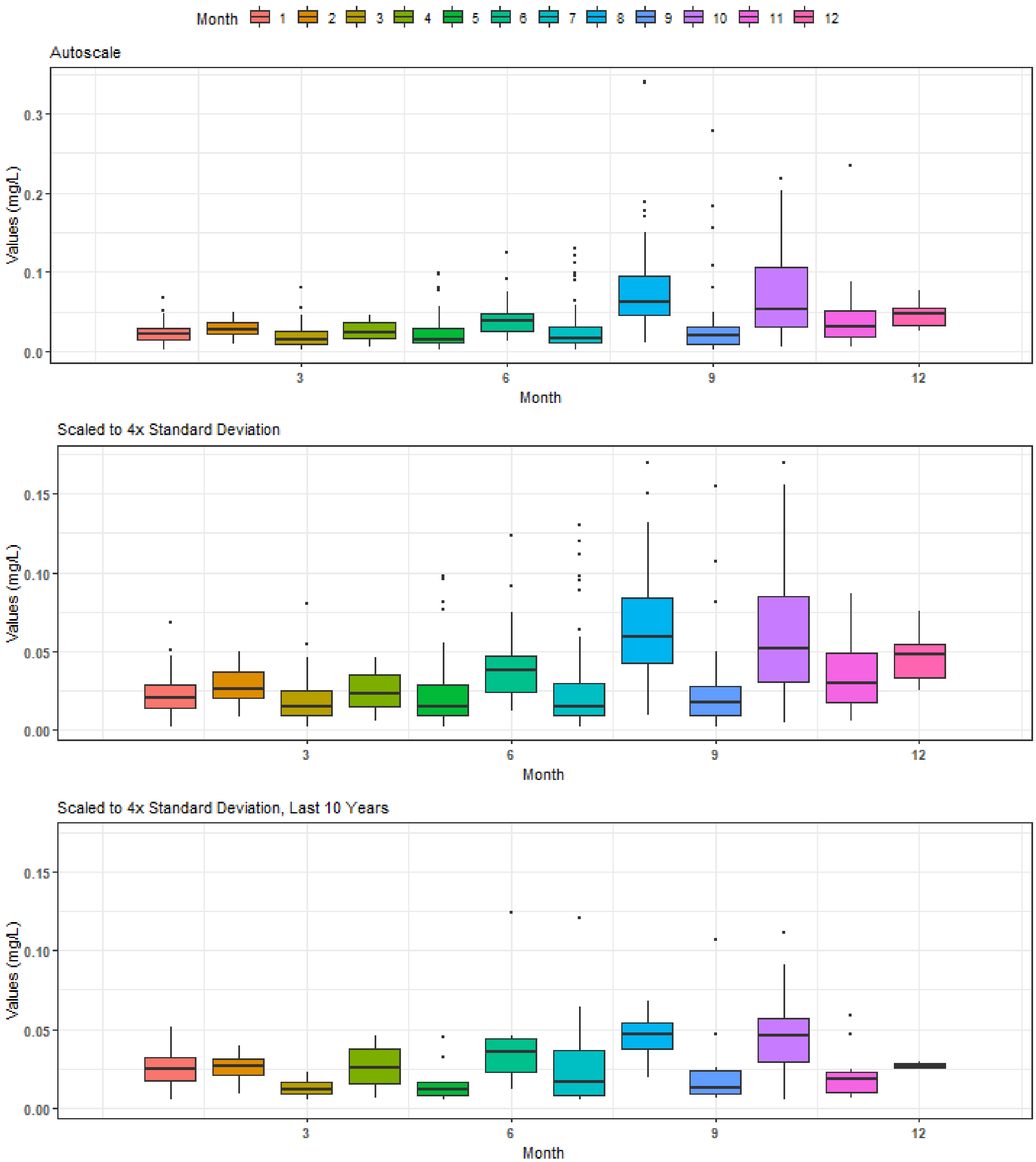
Scaled to 4x Standard Deviation, Last 10 Years




Summary Box Plots for Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0 0.1 0.2 0.3
1991 1996 2001 2006 2011 2016 2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00 0.05 0.10 0.15
1991 1996 2001 2006 2011 2016 2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00 0.05 0.10 0.15
2011 2013 2015 2017 2019 2021
Year

Summary Box Plots for Jensen Beach to Jupiter Inlet Aquatic Preserve

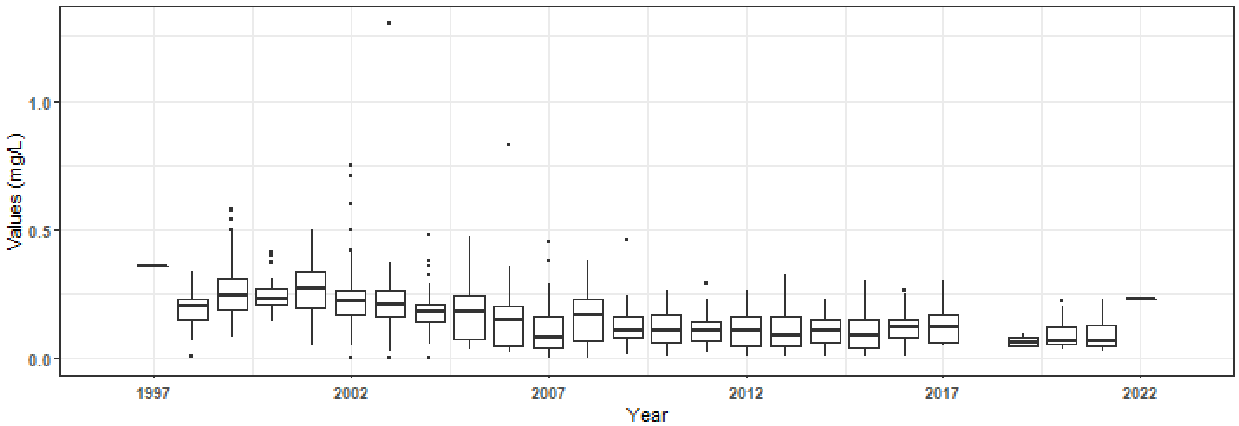
By Month



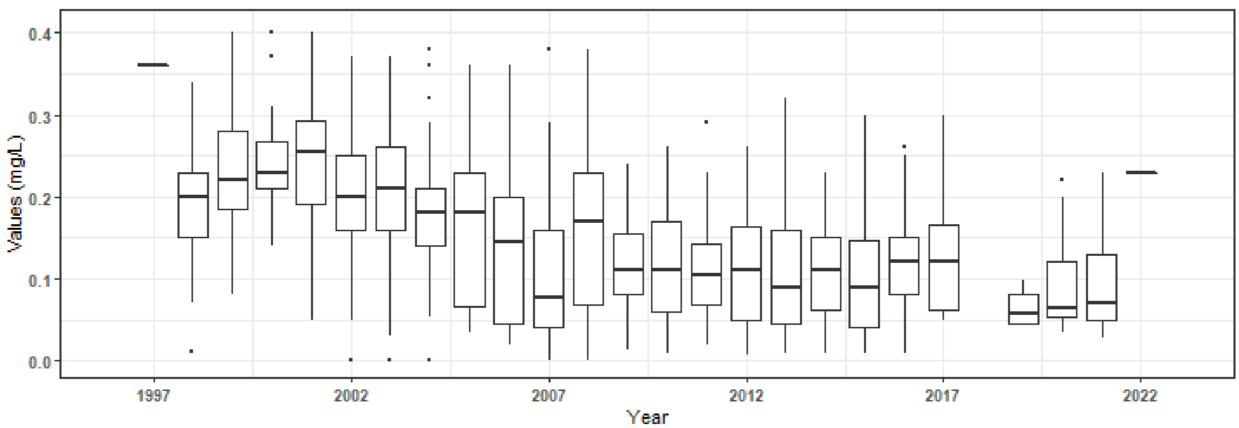
Summary Box Plots for Lemon Bay Aquatic Preserve

By Year

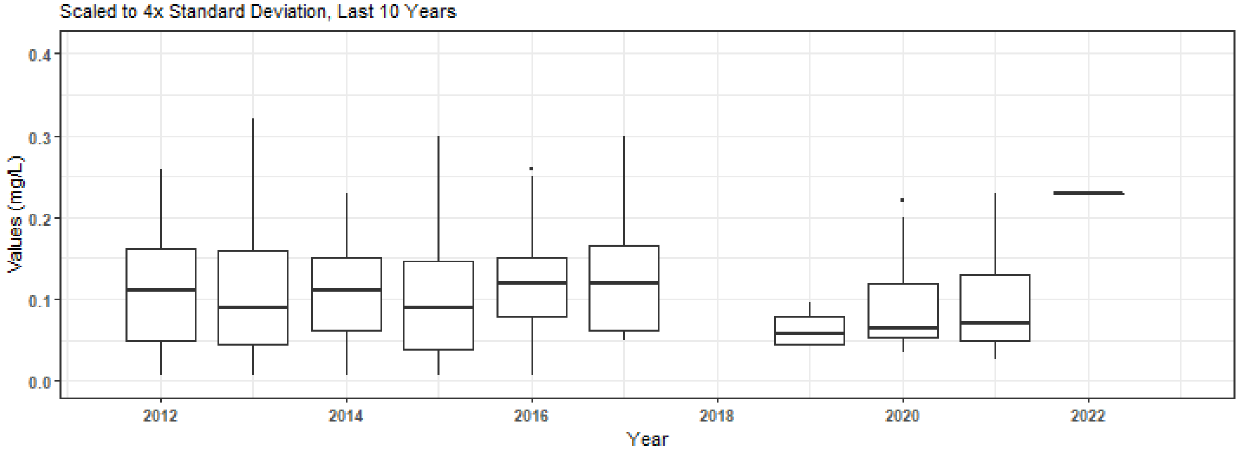
Autoscale



Scaled to 4x Standard Deviation

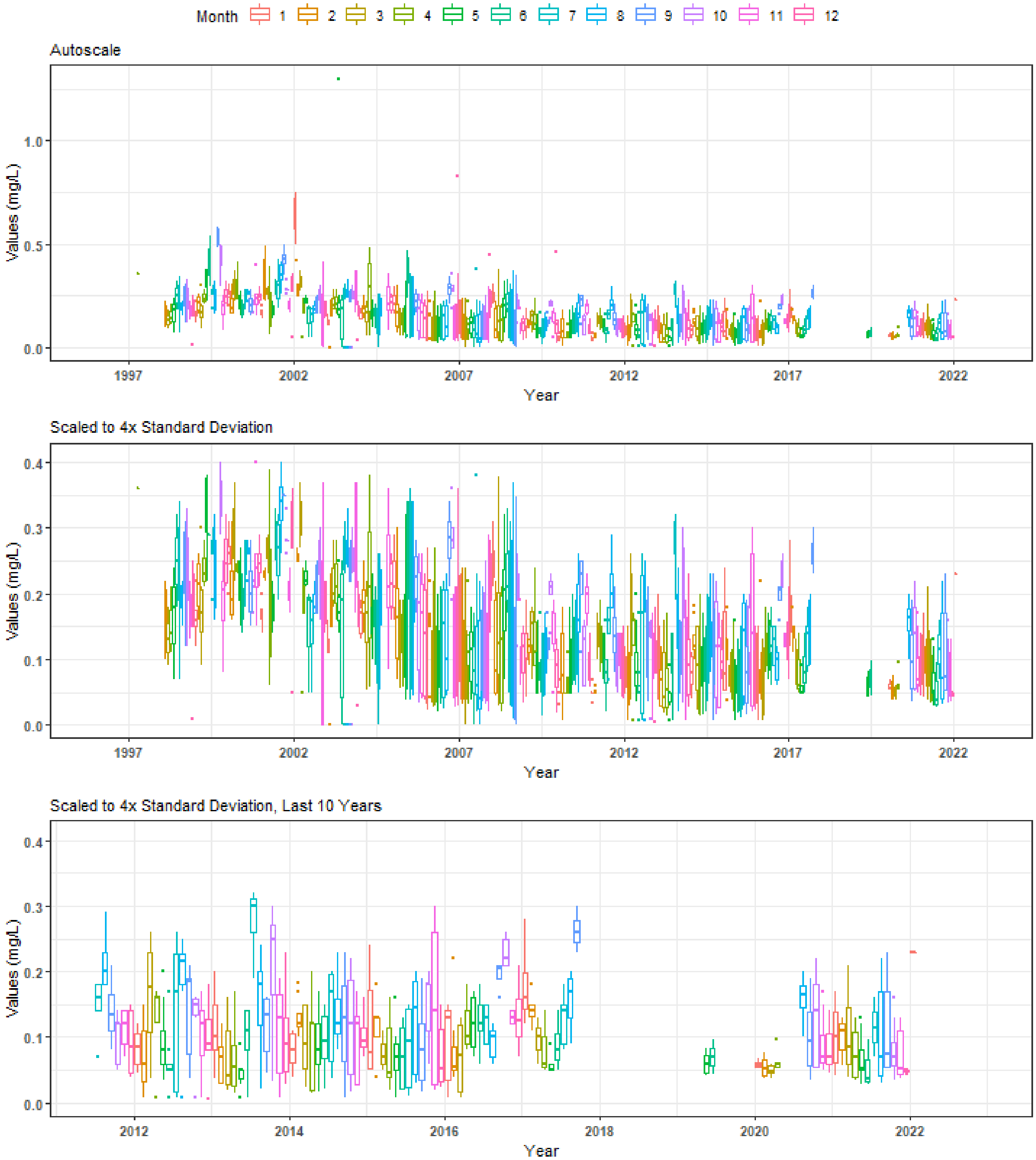


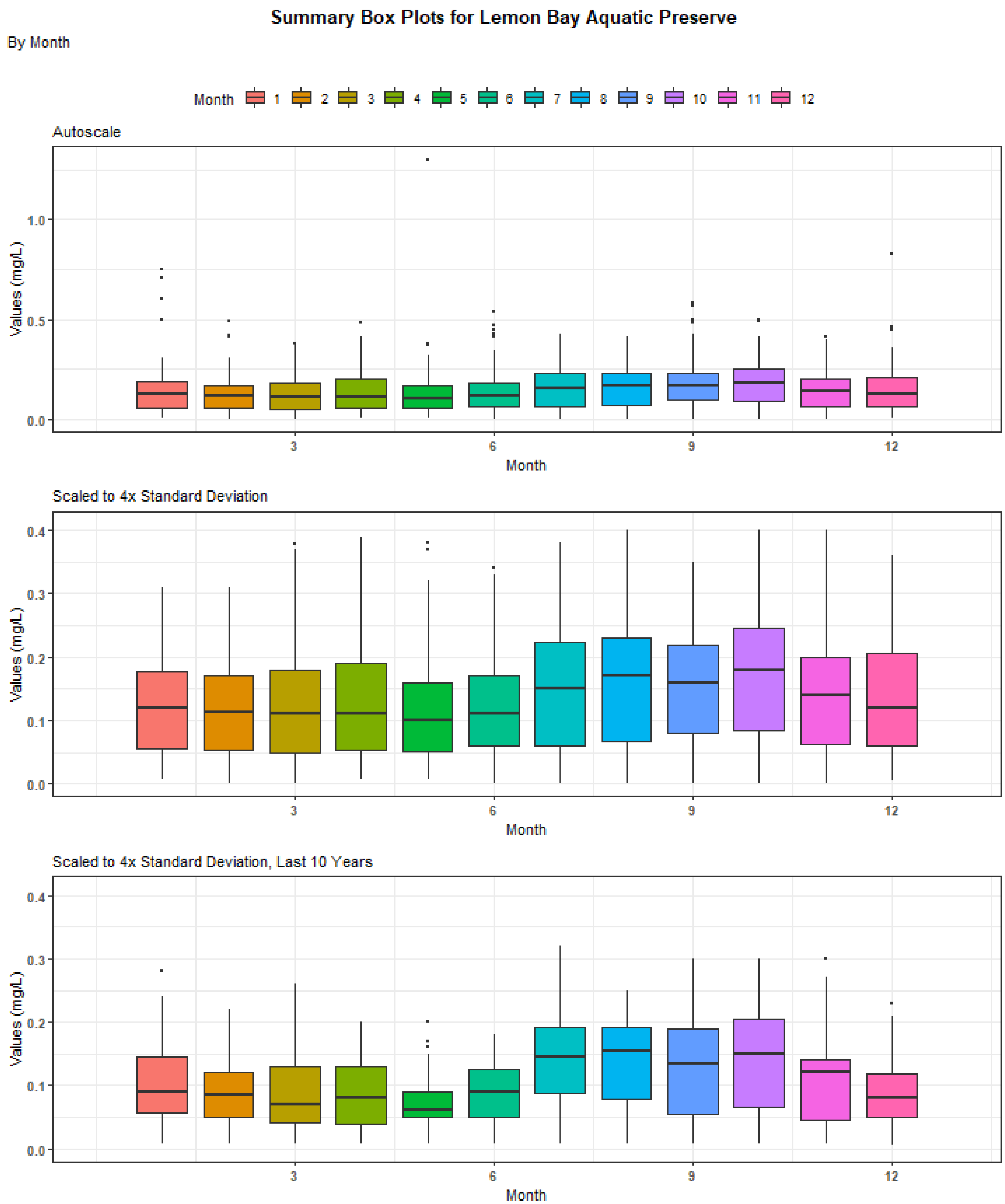
Scaled to 4x Standard Deviation, Last 10 Years



Summary Box Plots for Lemon Bay Aquatic Preserve

By Year & Month

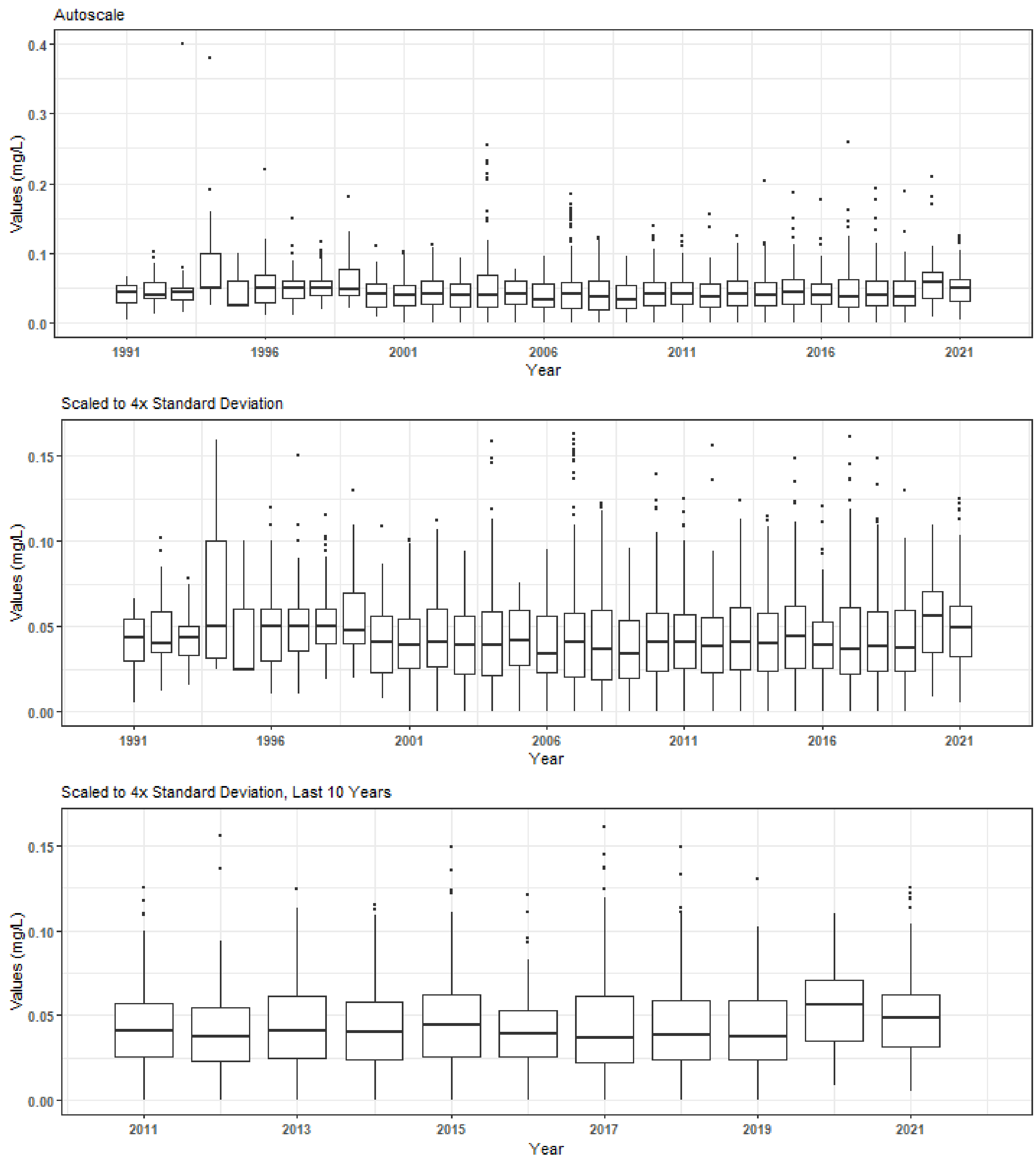




Summary Box Plots for Lemon Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
1.0
0.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.4
0.3
0.2
0.1
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.4
0.3
0.2
0.1
0.0
3
6
9
12
Month

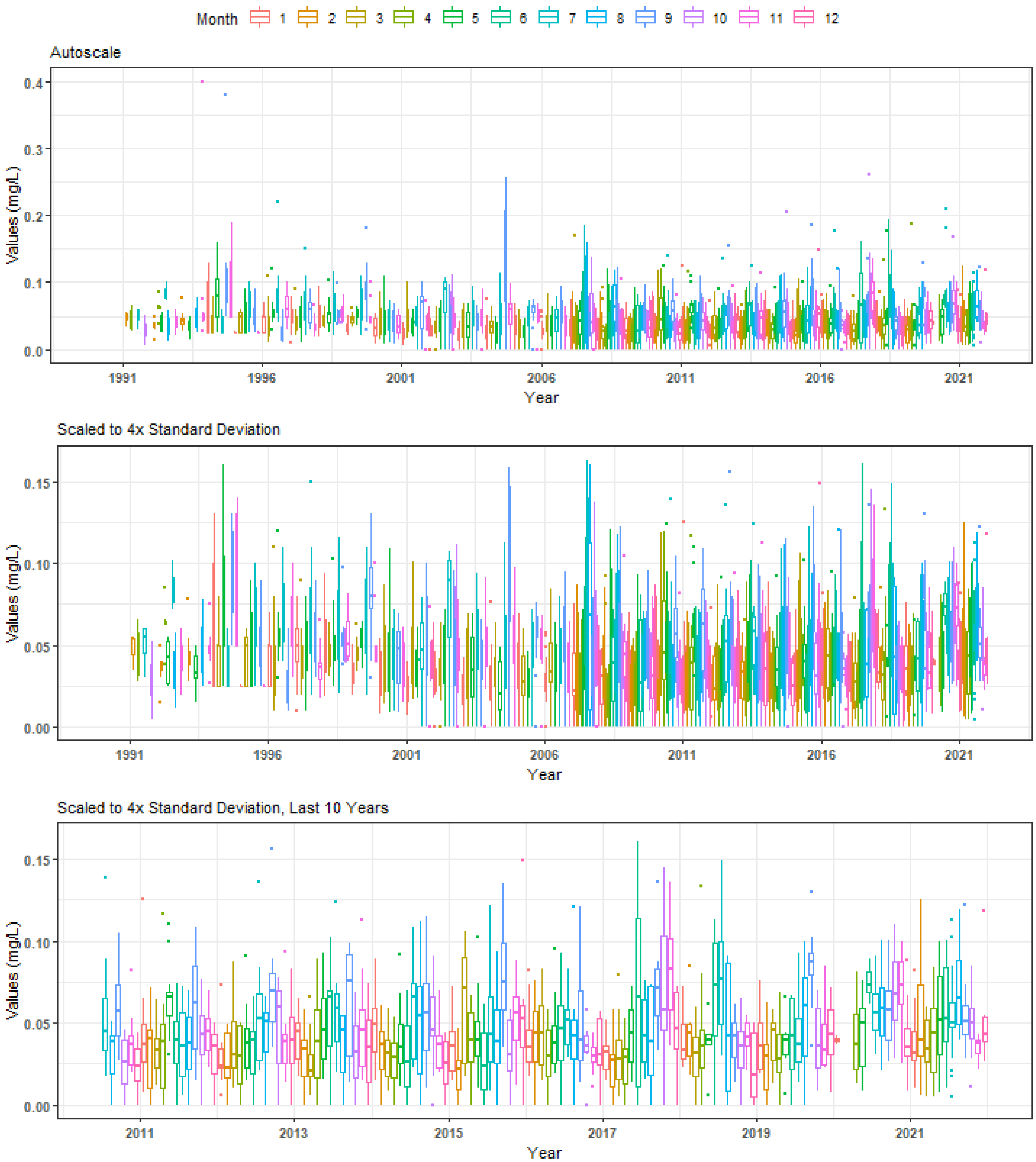
Summary Box Plots for Loxahatchee River-Lake Worth Creek Aquatic Preserve

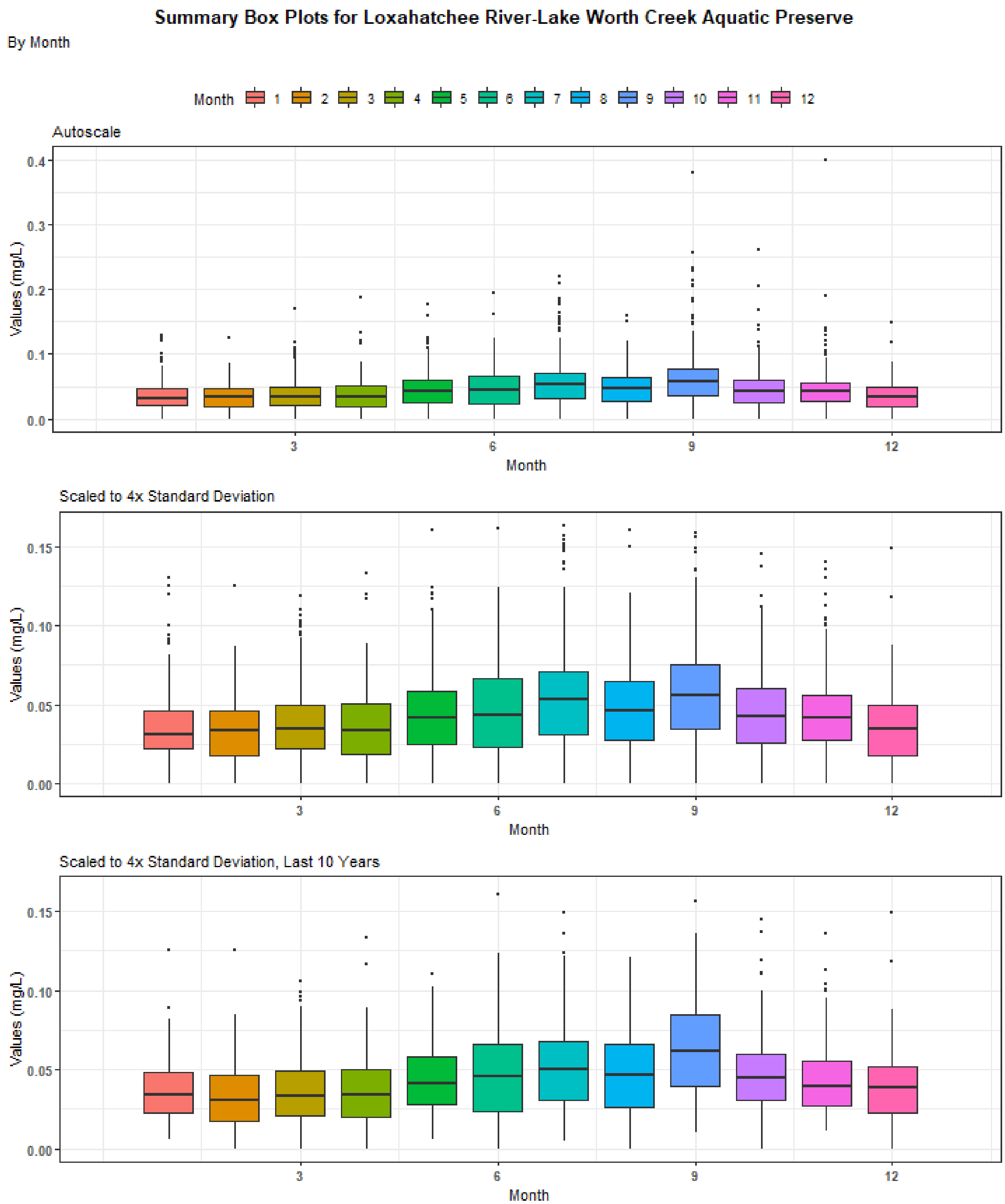
By Year



Summary Box Plots for Loxahatchee River-Lake Worth Creek Aquatic Preserve

By Year & Month



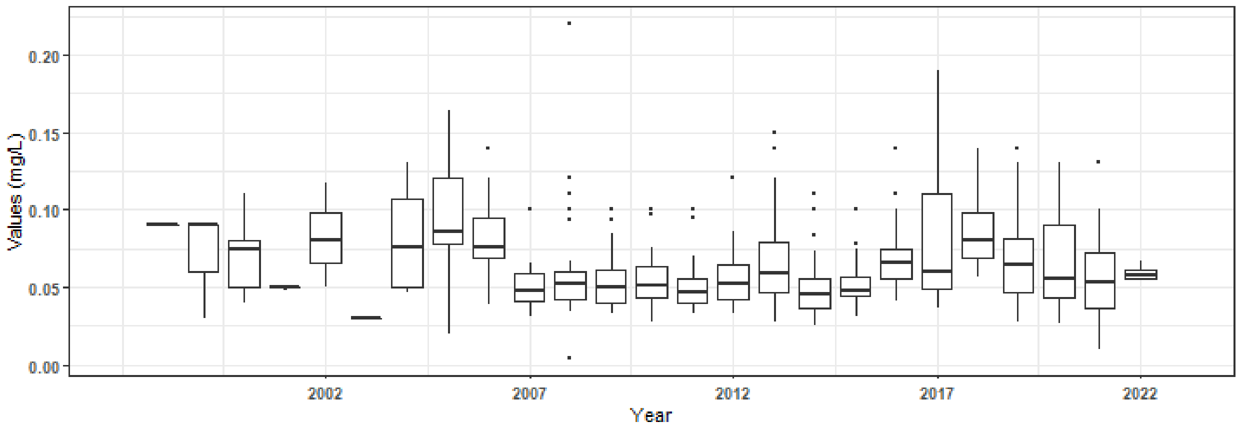


Summary Box Plots for Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
3
6
9
12
Month

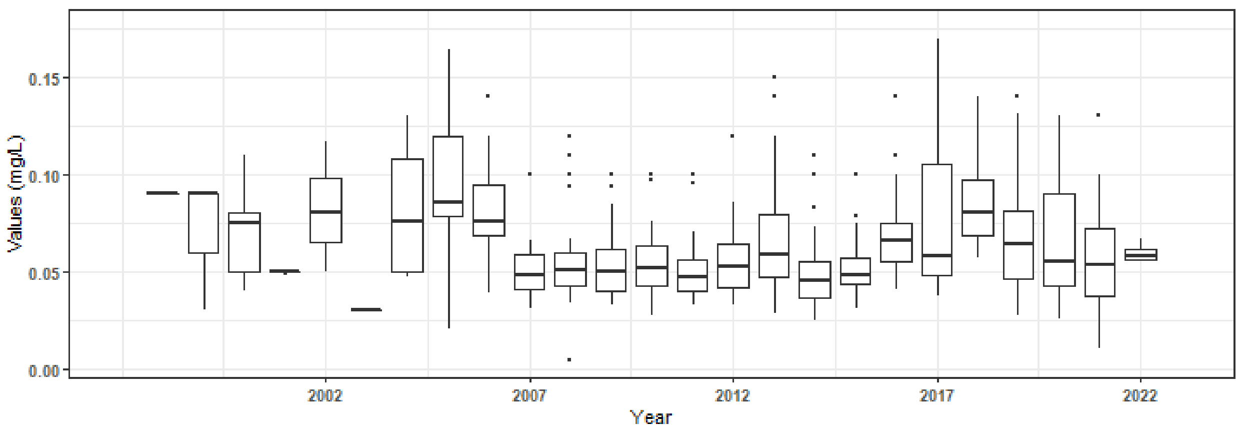
Summary Box Plots for Matlacha Pass Aquatic Preserve

By Year

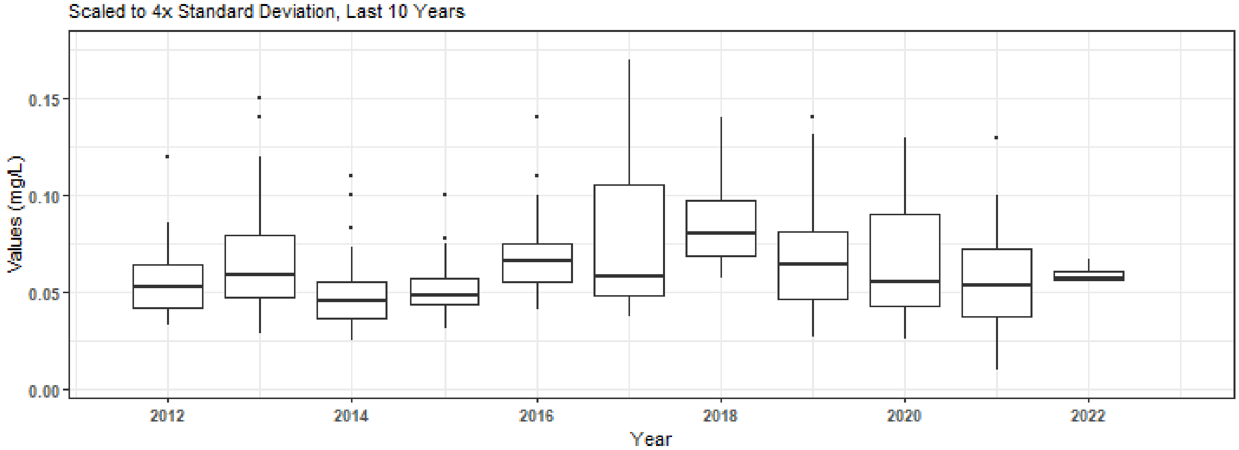
Autoscale

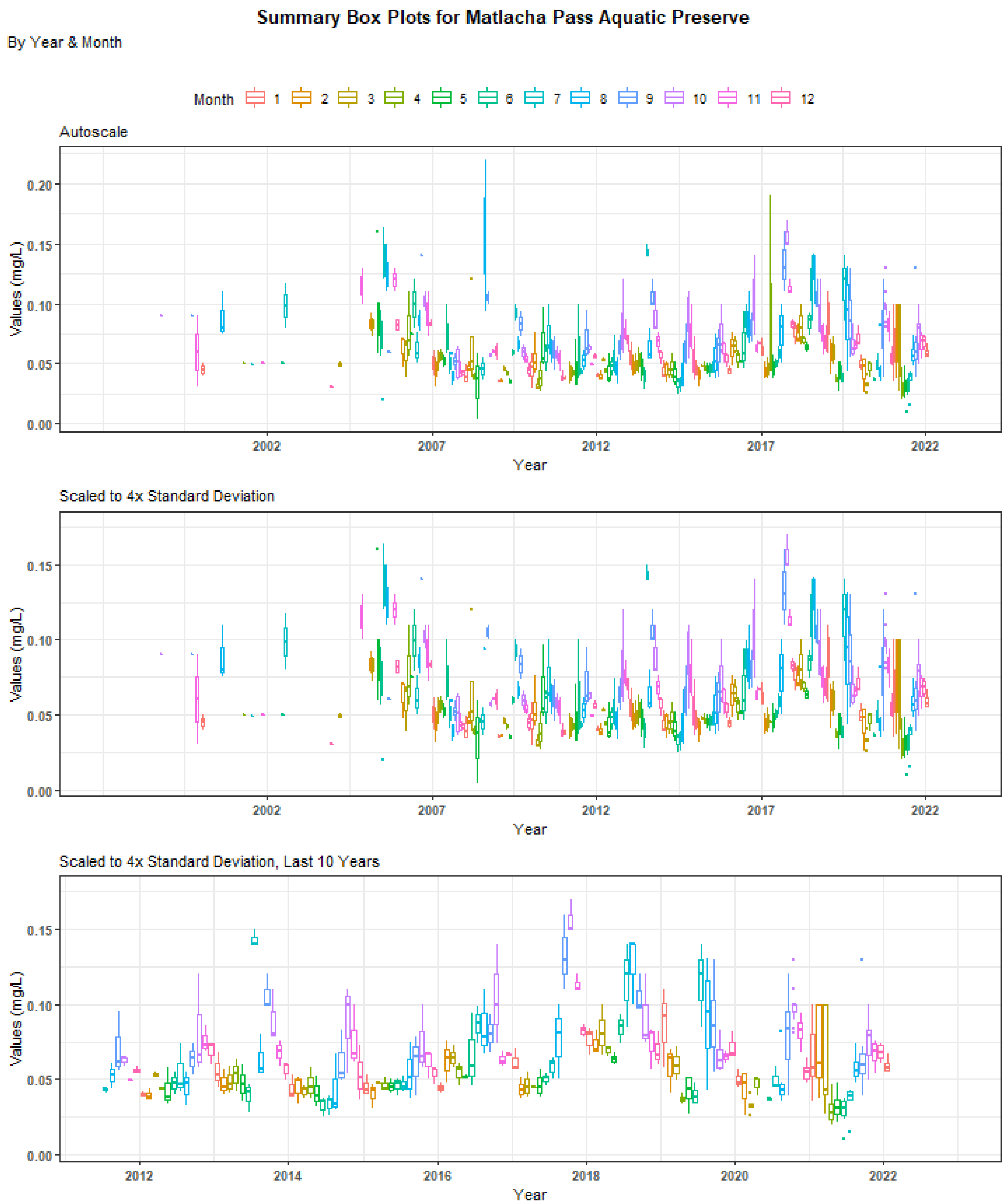


Scaled to 4x Standard Deviation



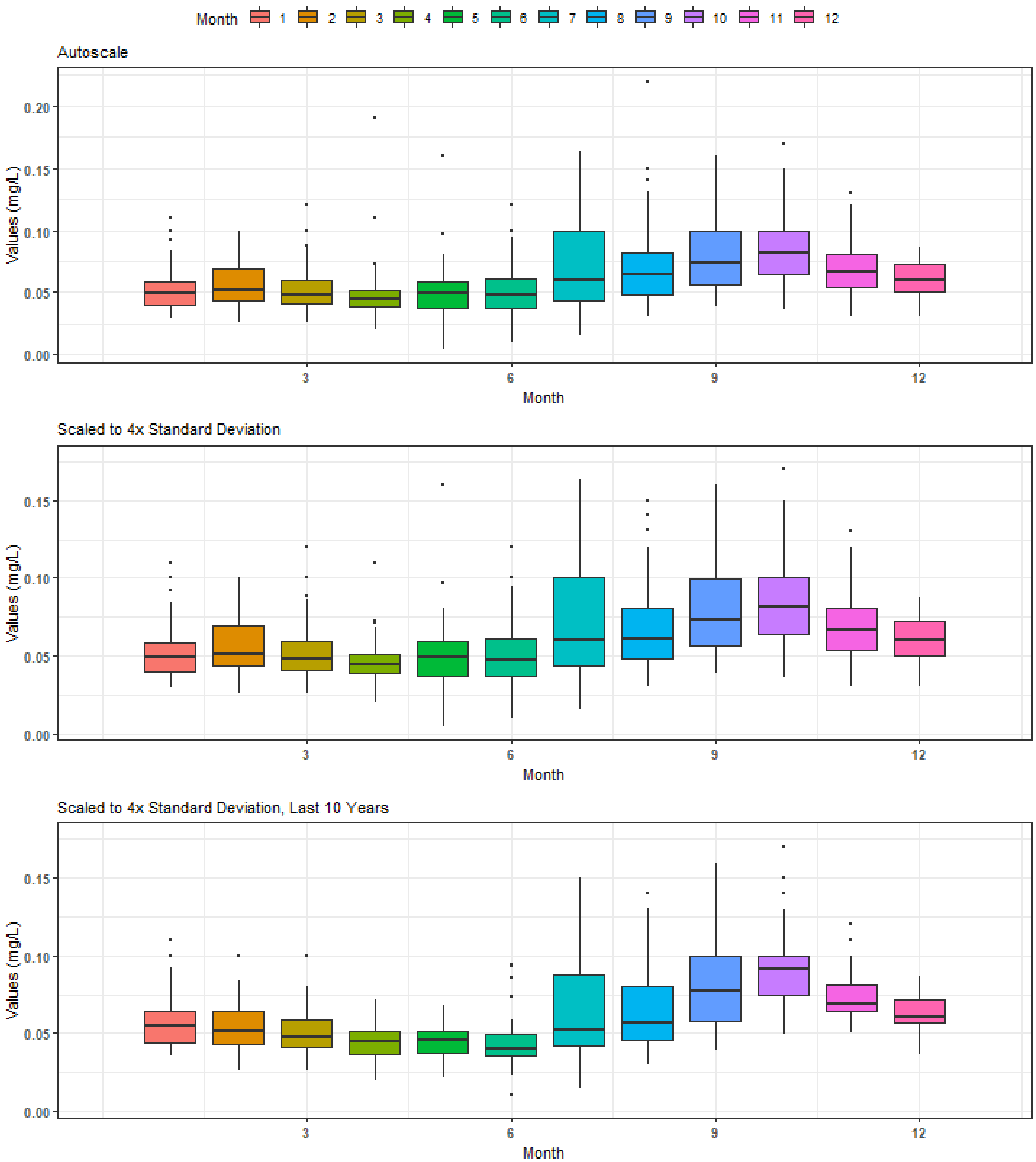
Scaled to 4x Standard Deviation, Last 10 Years




Summary Box Plots for Matlacha Pass Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.00
0.05
0.10
0.15
0.20
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
2012
2014
2016
2018
2020
2022
Year

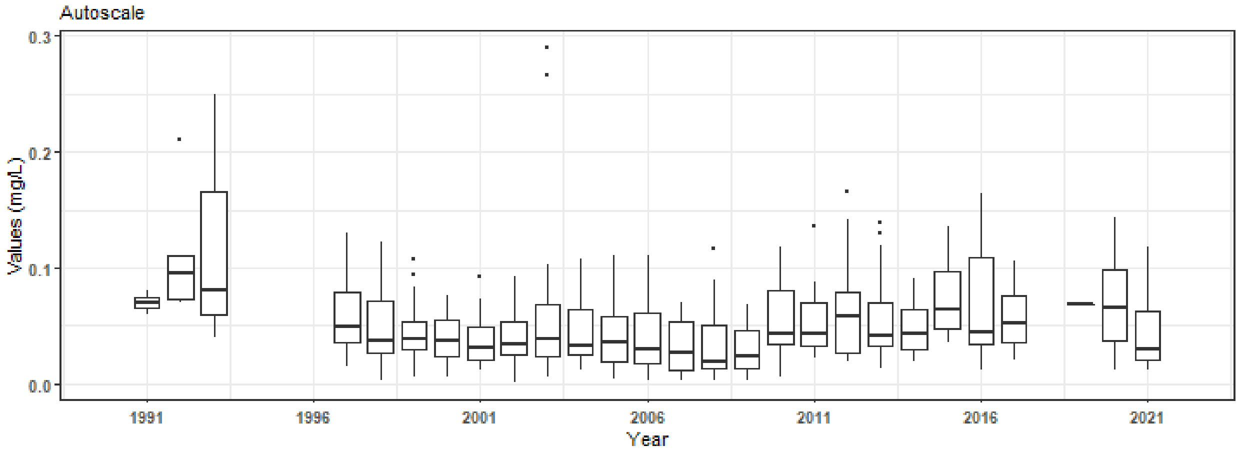
Summary Box Plots for Matlacha Pass Aquatic Preserve

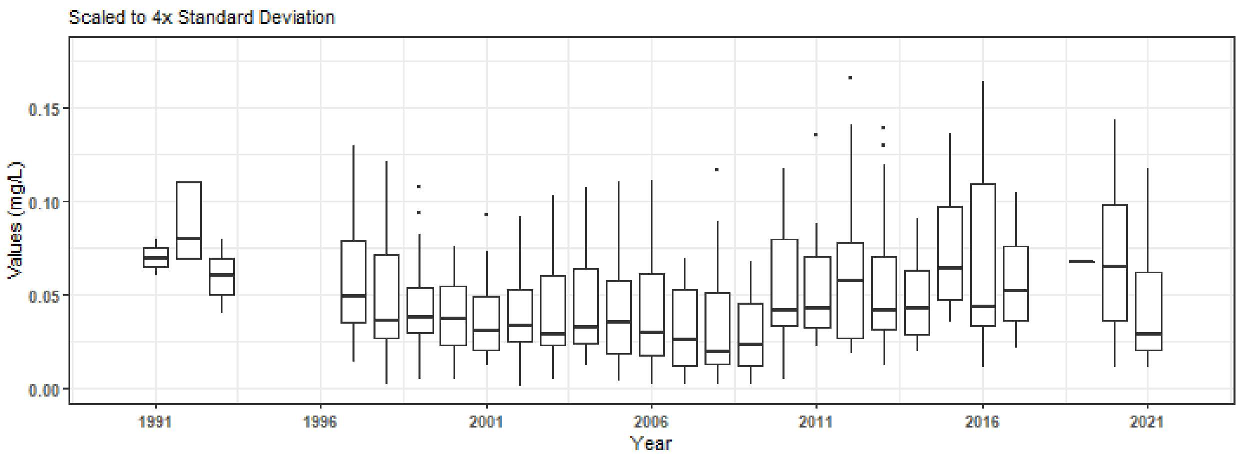
By Month

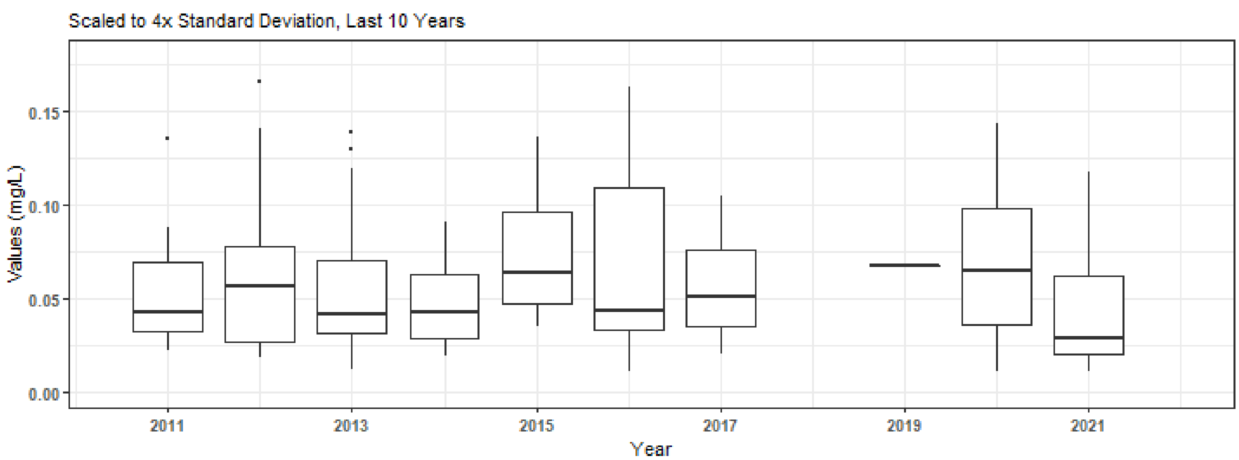


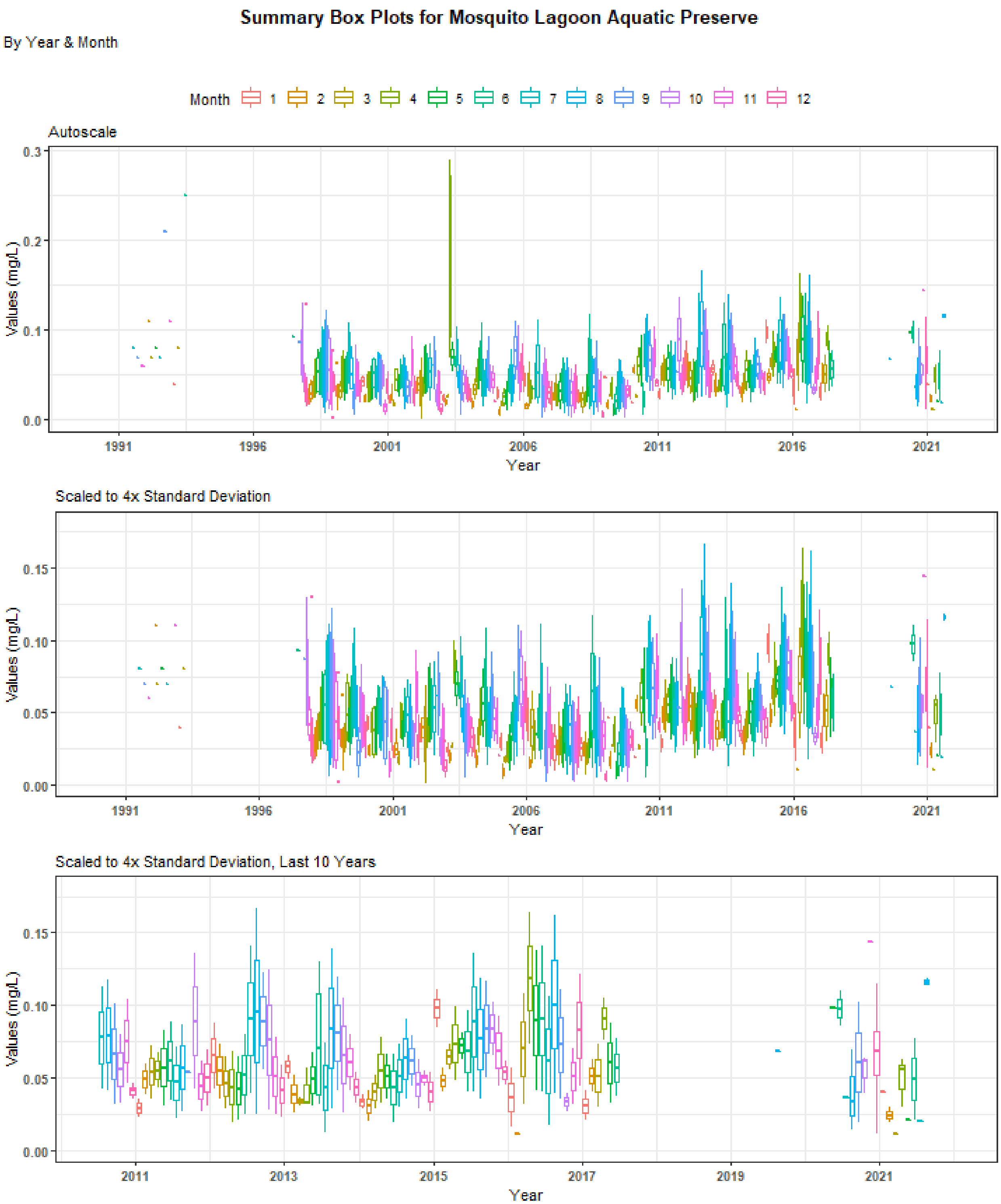
Summary Box Plots for Mosquito Lagoon Aquatic Preserve

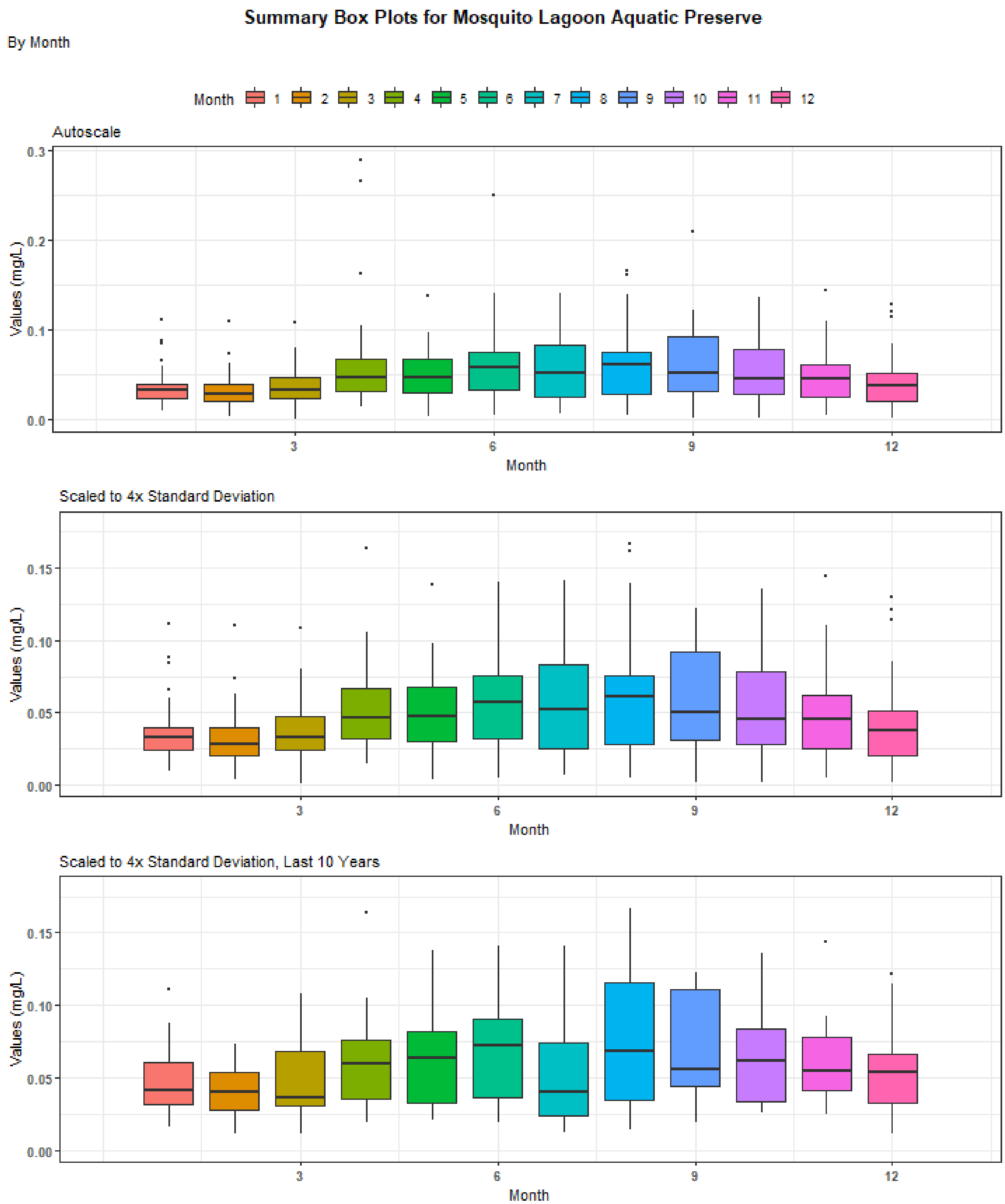
By Year







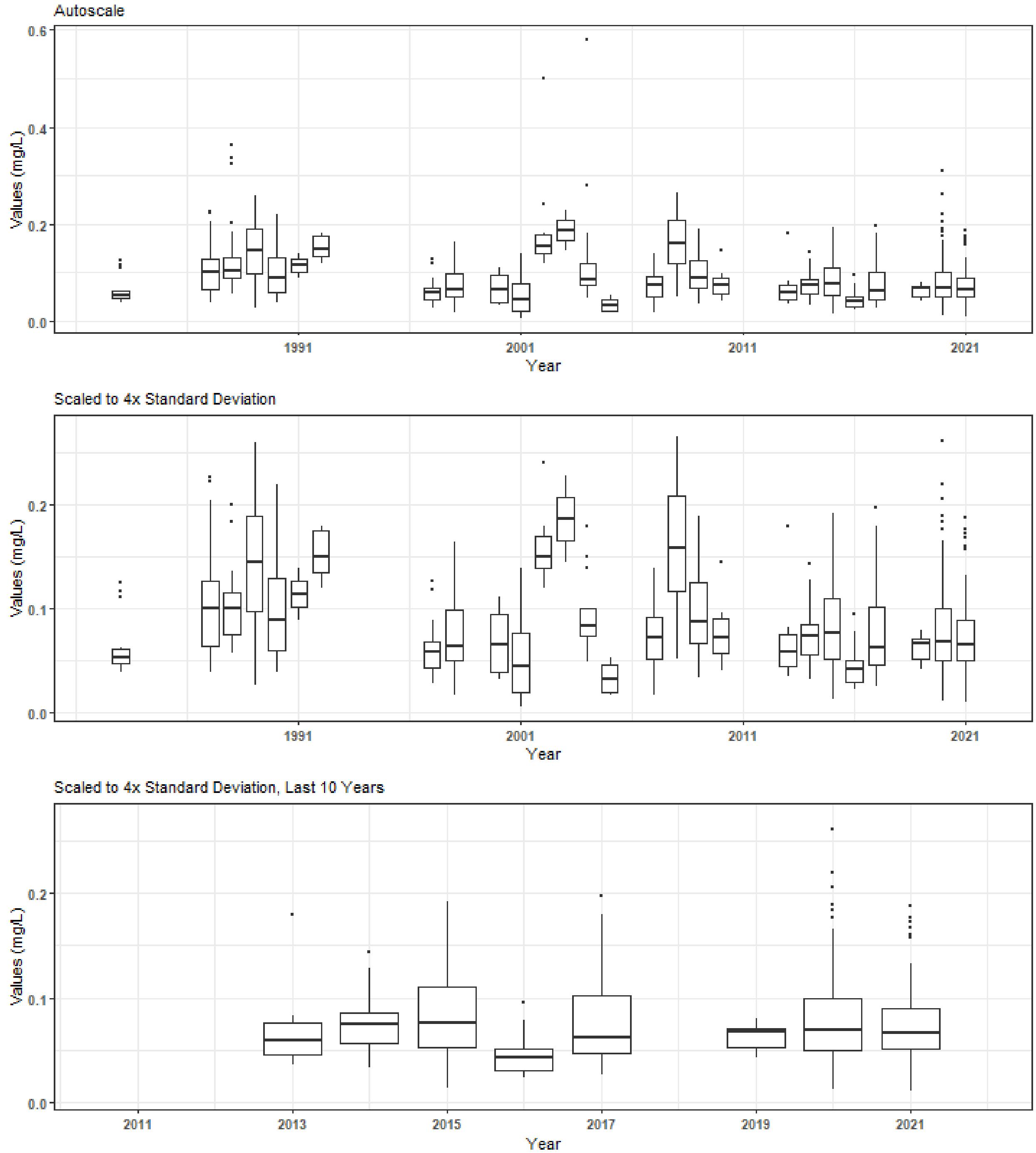

Summary Box Plots for Mosquito Lagoon Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
2011
2013
2015
2017
2019
2021
Year

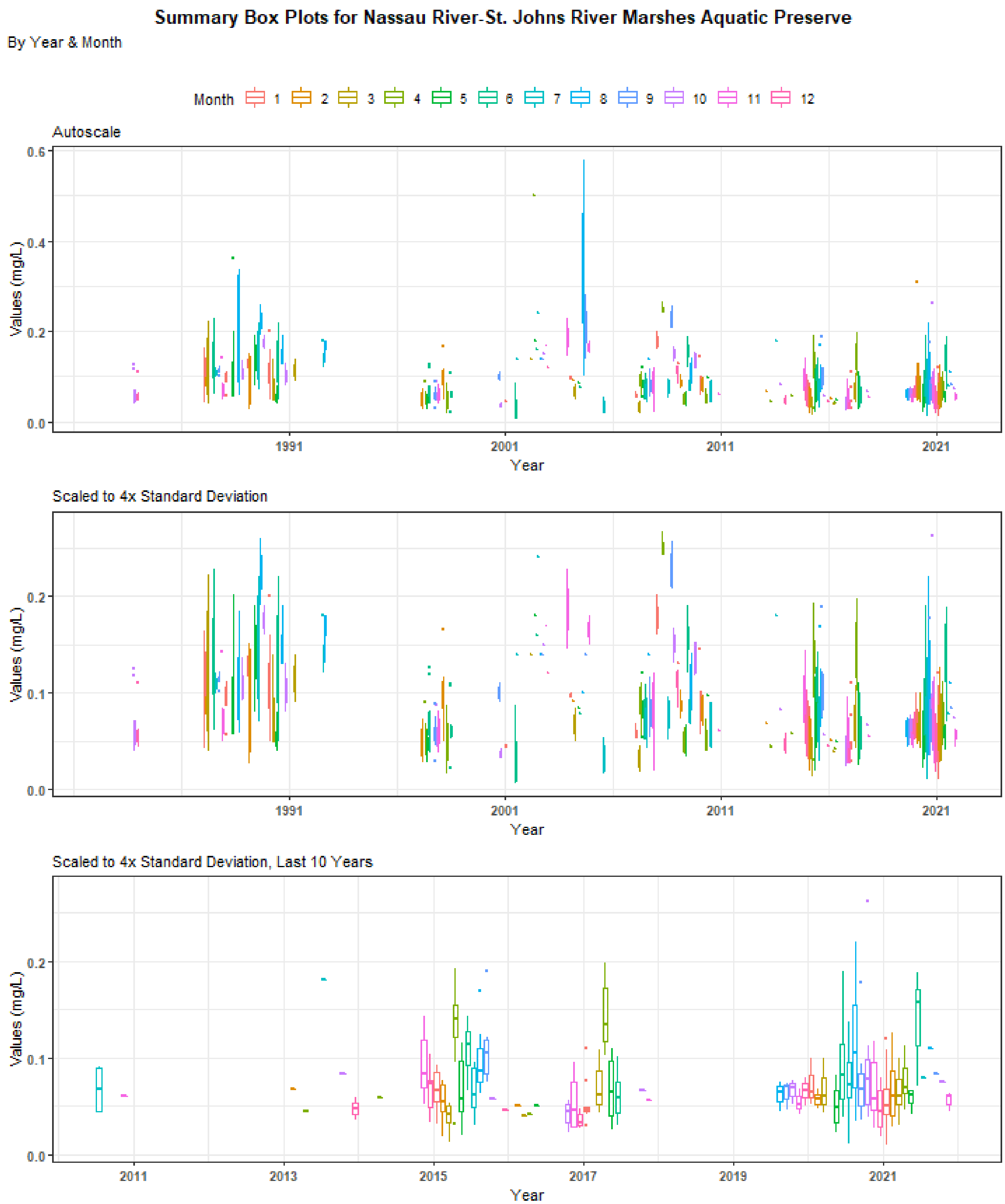


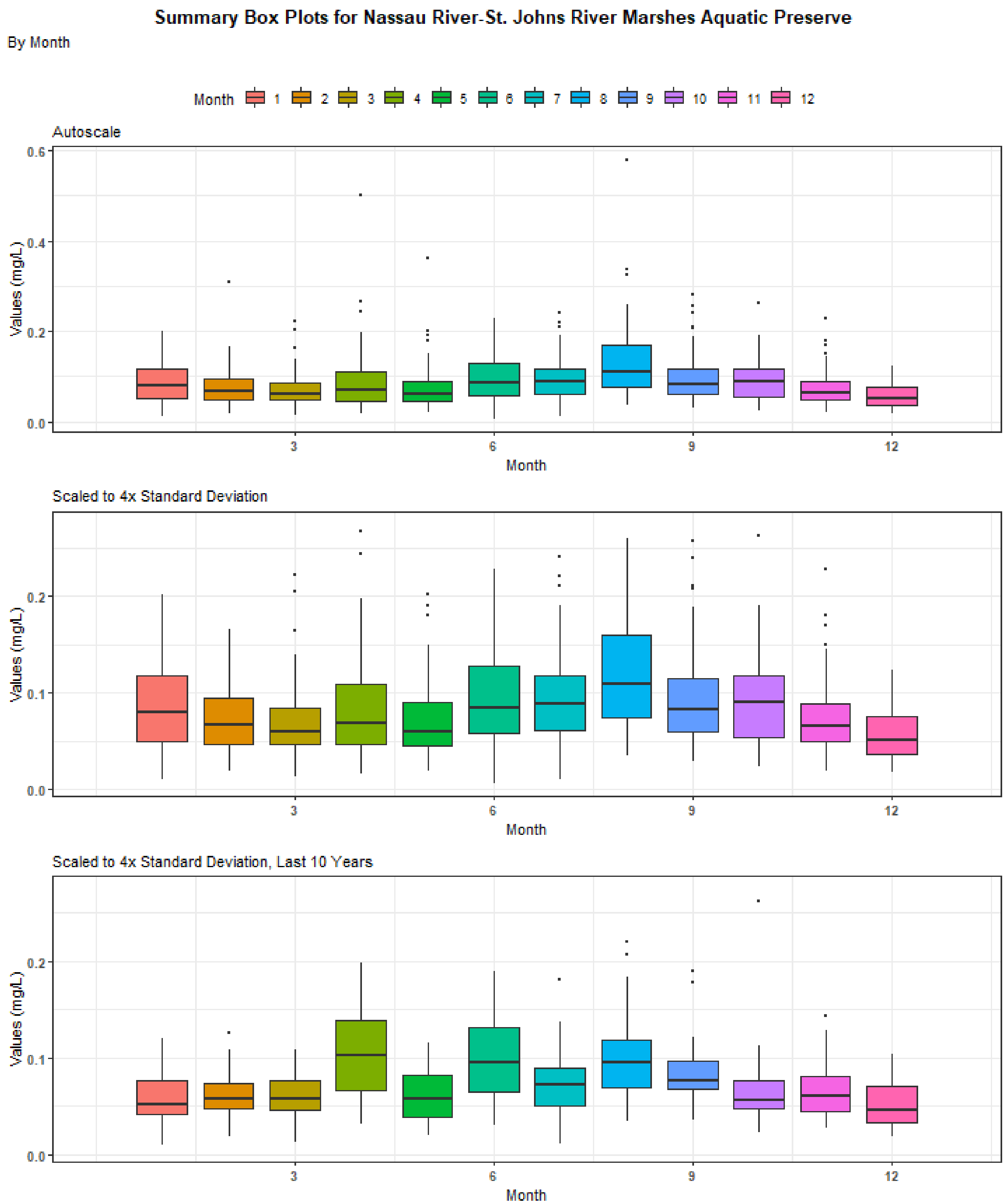
Summary Box Plots for Mosquito Lagoon Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

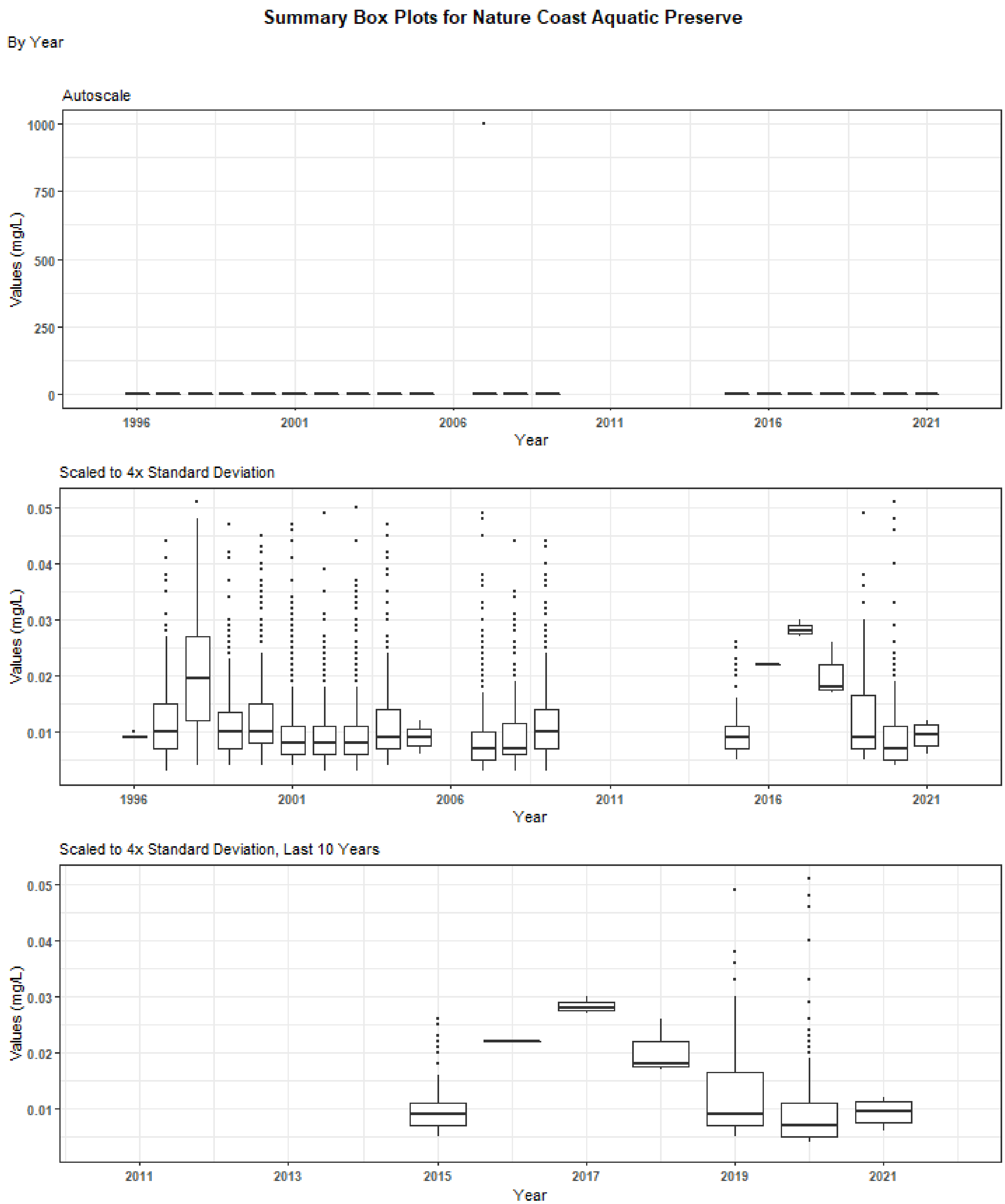
Summary Box Plots for Nassau River-St. Johns River Marshes Aquatic Preserve

By Year



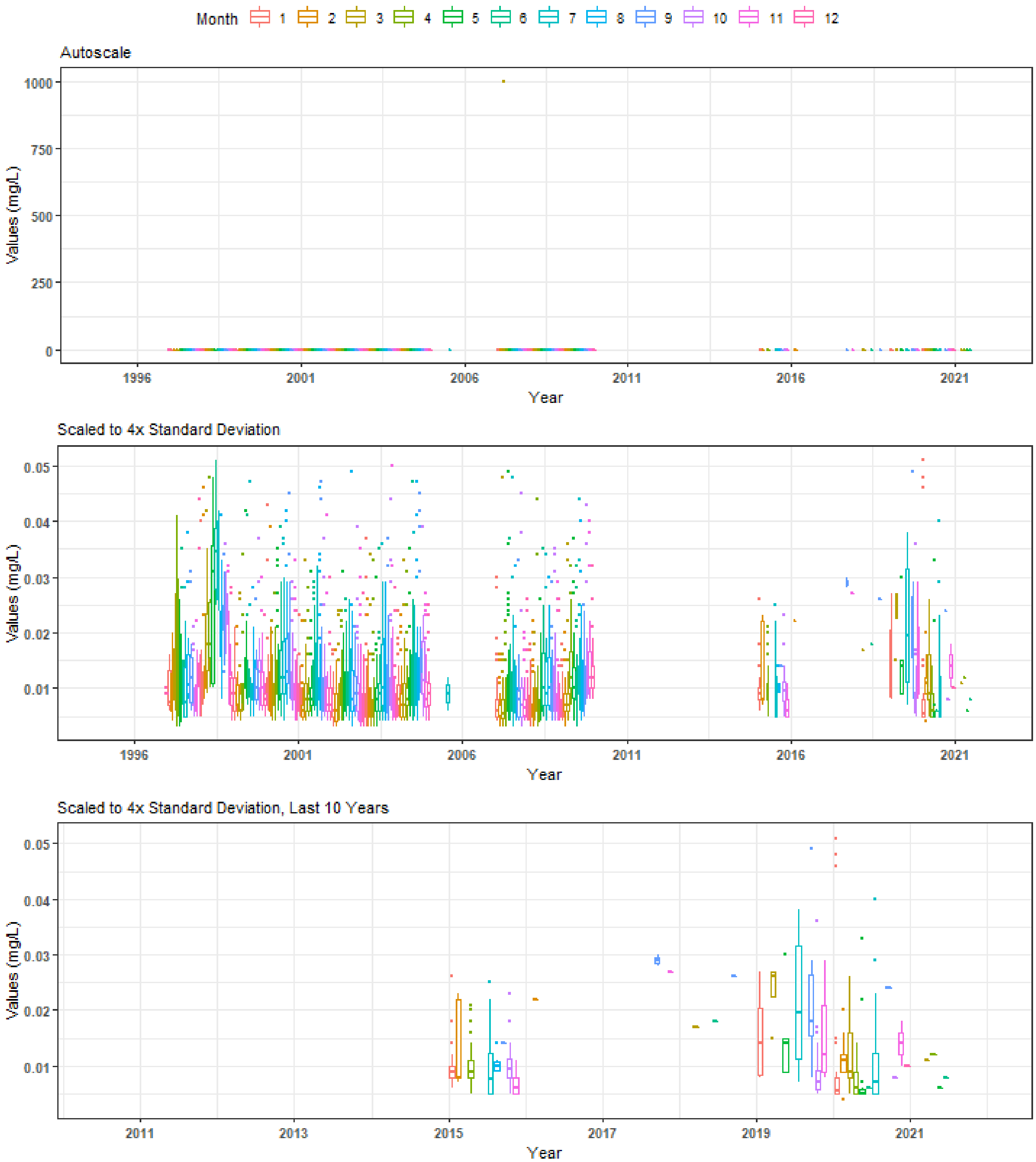

Summary Box Plots for Nassau River-St. Johns River Marshes Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.2
0.4
0.6
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
2011
2013
2015
2017
2019
2021
Year

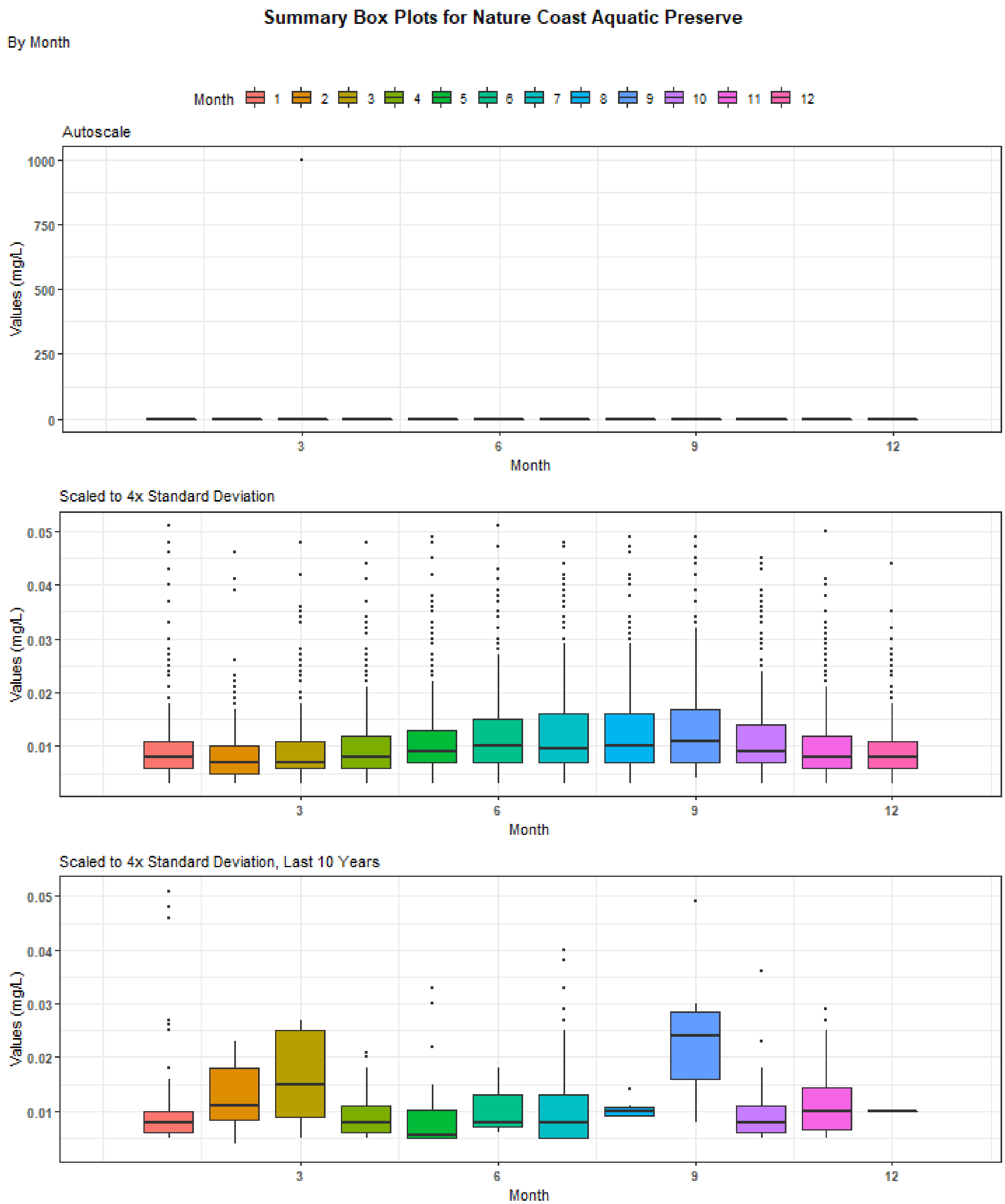

Summary Box Plots for Nassau River-St. Johns River Marshes Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.2
0.4
0.6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
3
6
9
12
Month


Summary Box Plots for Nature Coast Aquatic Preserve
By Year
Autoscale
Values (mg/L)
1000
750
500
250
0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.05
0.04
0.03
0.02
0.01
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.05
0.04
0.03
0.02
0.01
2011
2013
2015
2017
2019
2021
Year

Summary Box Plots for Nature Coast Aquatic Preserve

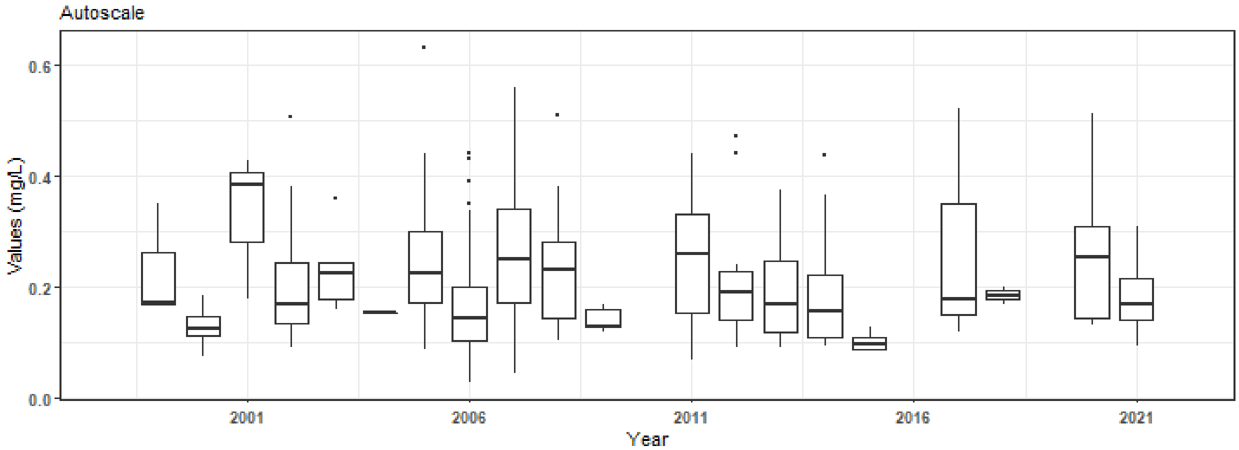
By Year & Month

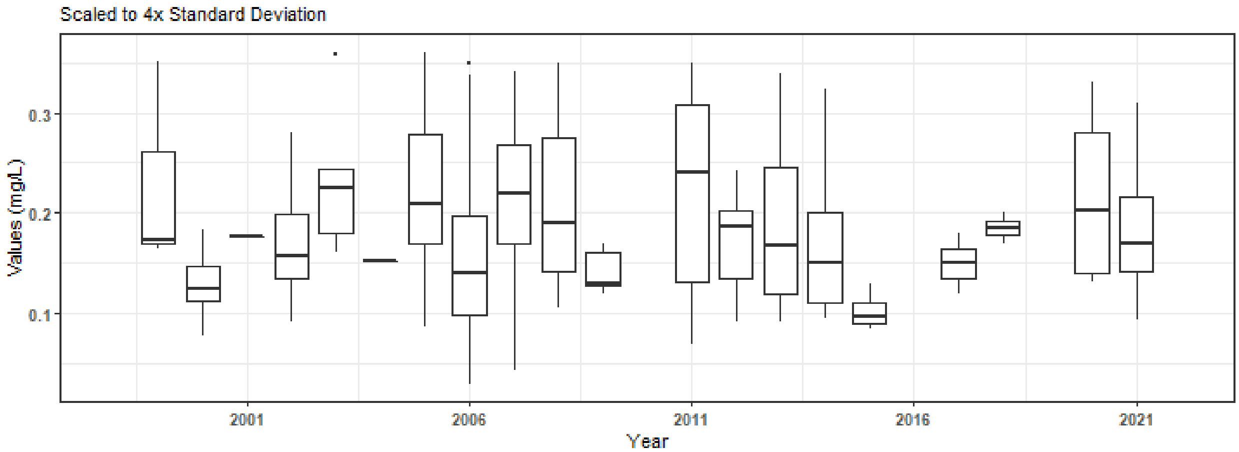


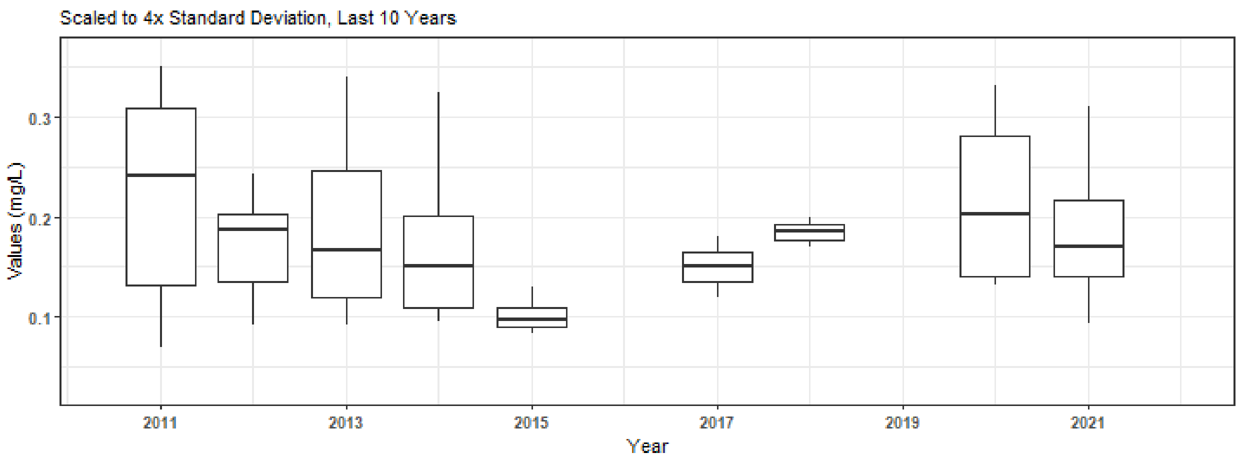

Summary Box Plots for Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
1000
750
500
250
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.05
0.04
0.03
0.02
0.01
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.05
0.04
0.03
0.02
0.01
3
6
9
12
Month

Summary Box Plots for North Fork St. Lucie Aquatic Preserve

By Year

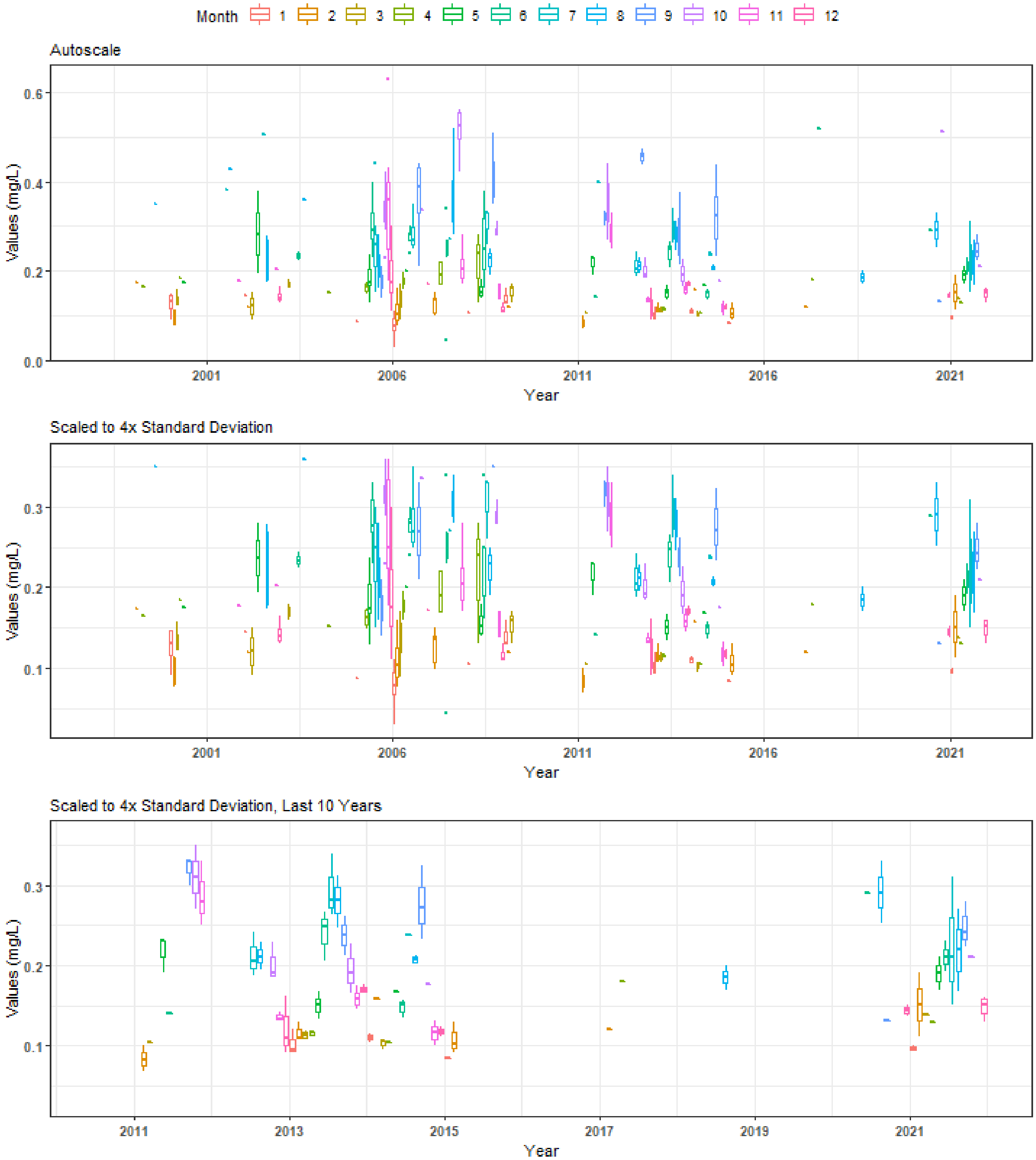


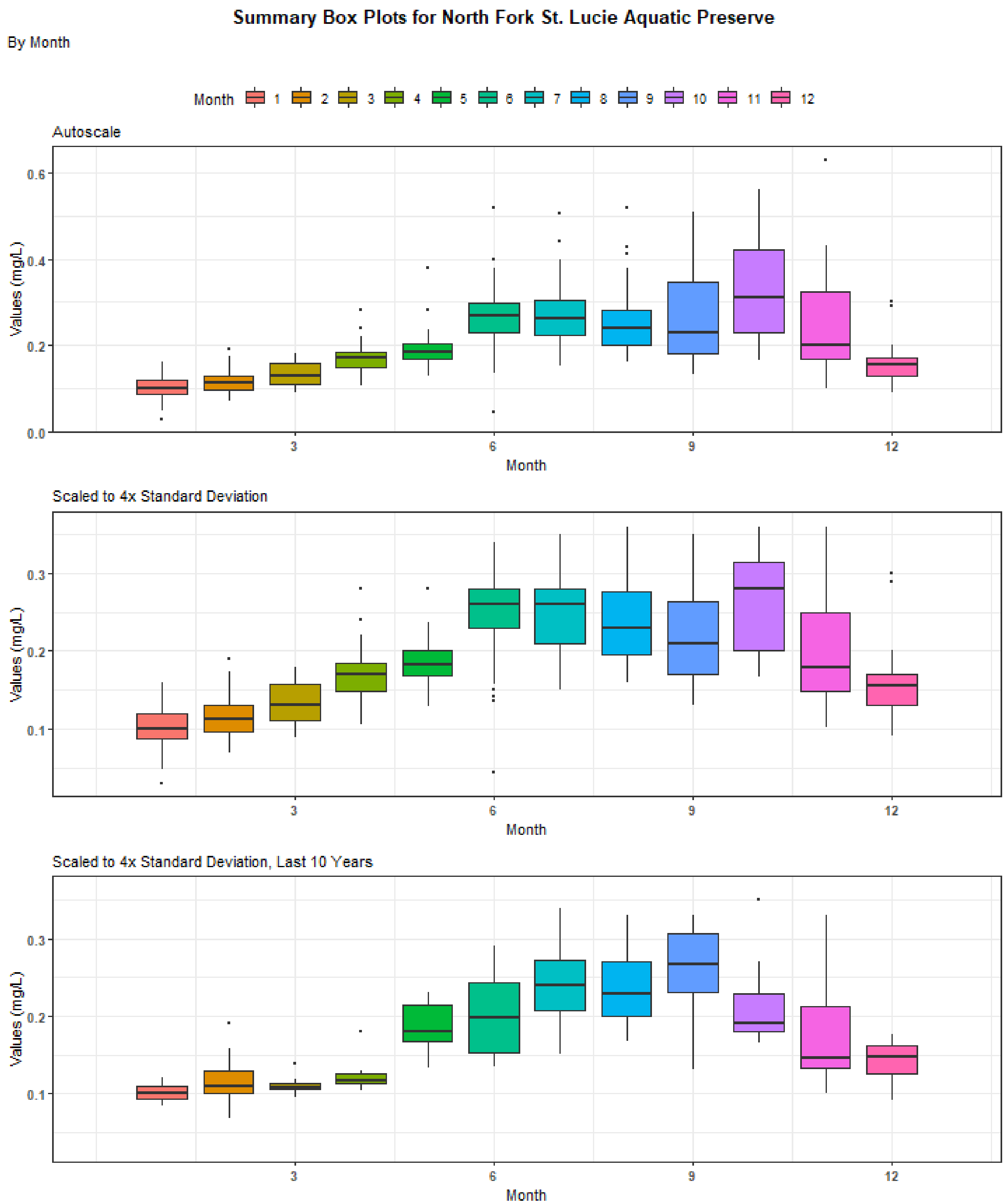




Summary Box Plots for North Fork St. Lucie Aquatic Preserve

By Year & Month



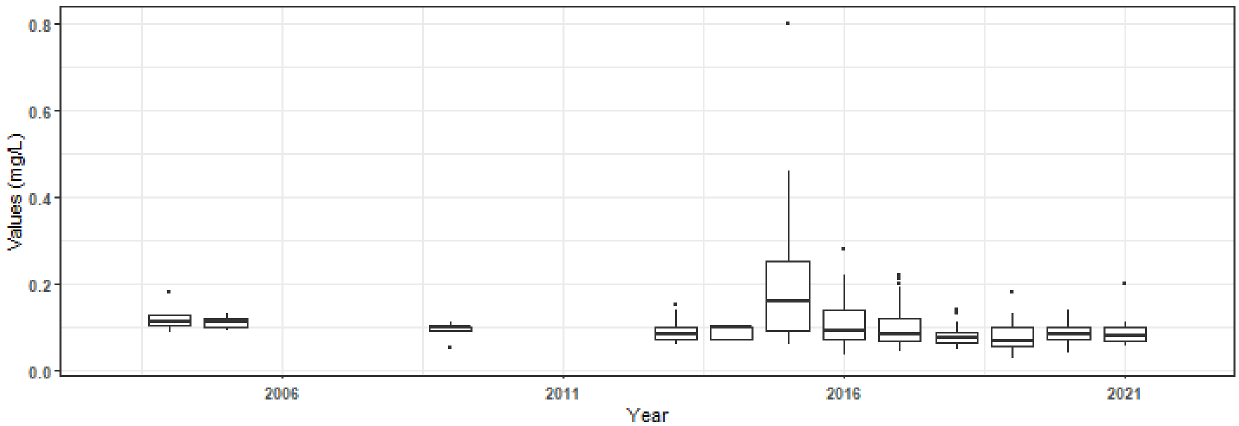


Summary Box Plots for North Fork St. Lucie Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.2
0.4
0.6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.1
0.2
0.3
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.1
0.2
0.3
3
6
9
12
Month

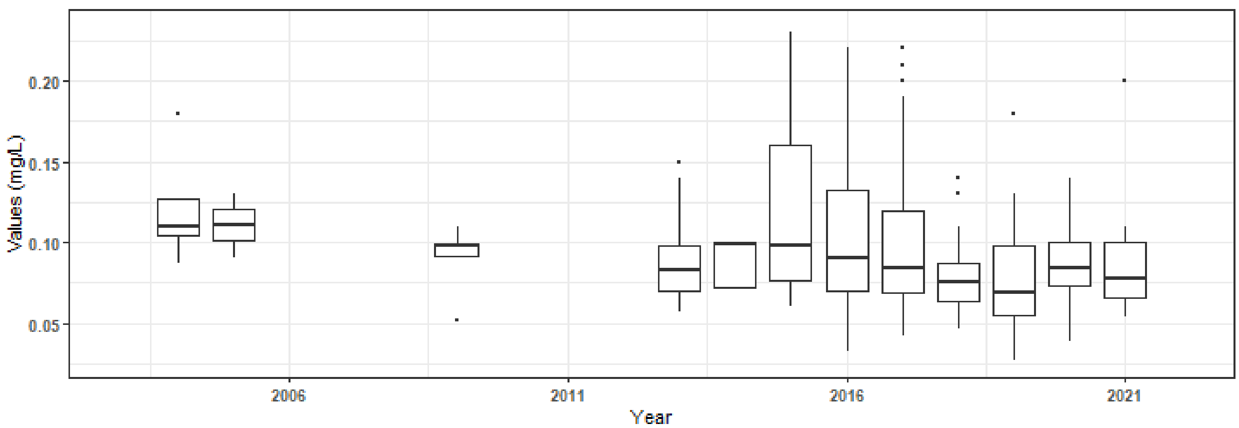
Summary Box Plots for Pellicer Creek Aquatic Preserve

By Year

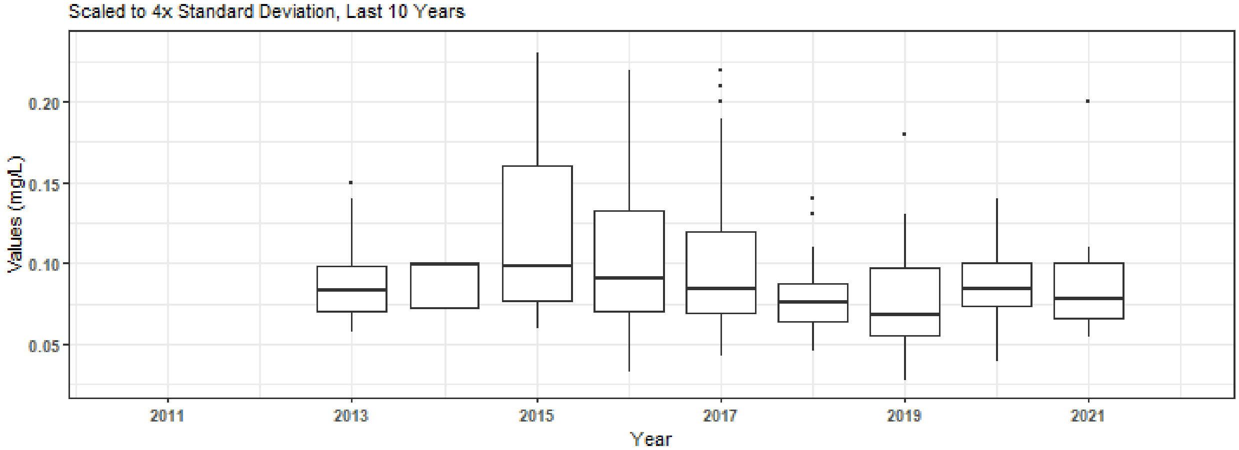
Autoscale



Scaled to 4x Standard Deviation

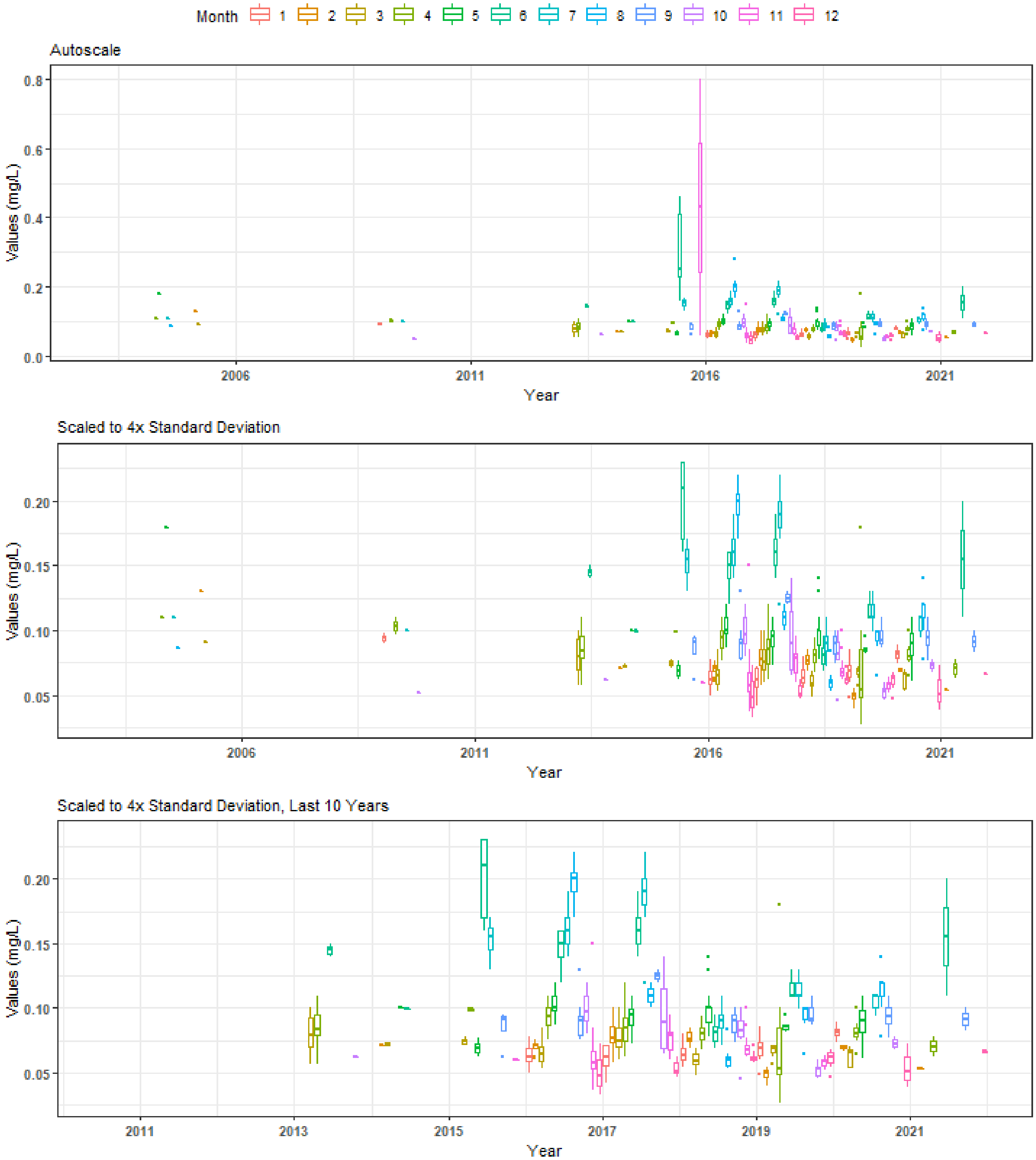


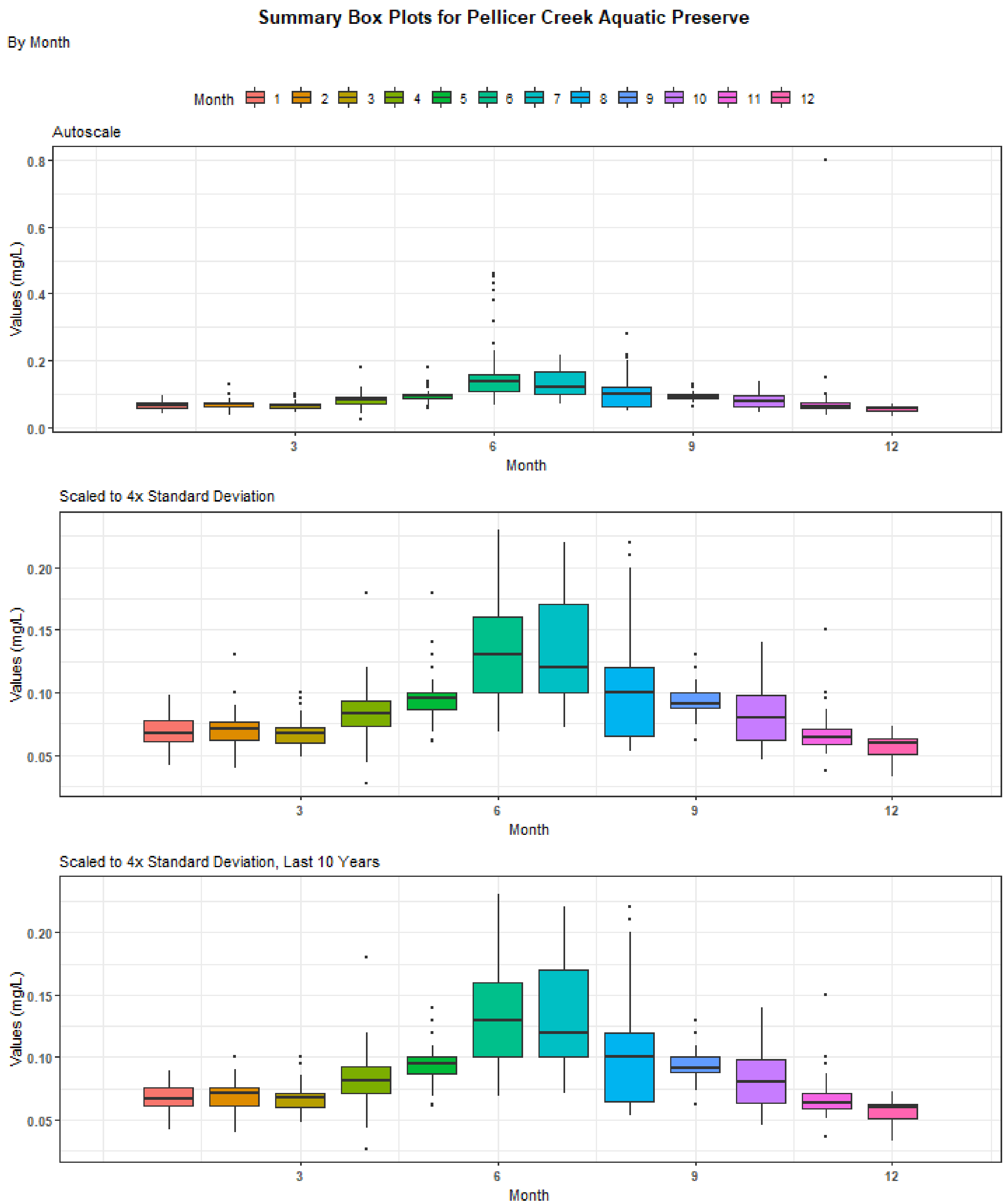
Scaled to 4x Standard Deviation, Last 10 Years



Summary Box Plots for Pellicer Creek Aquatic Preserve

By Year & Month



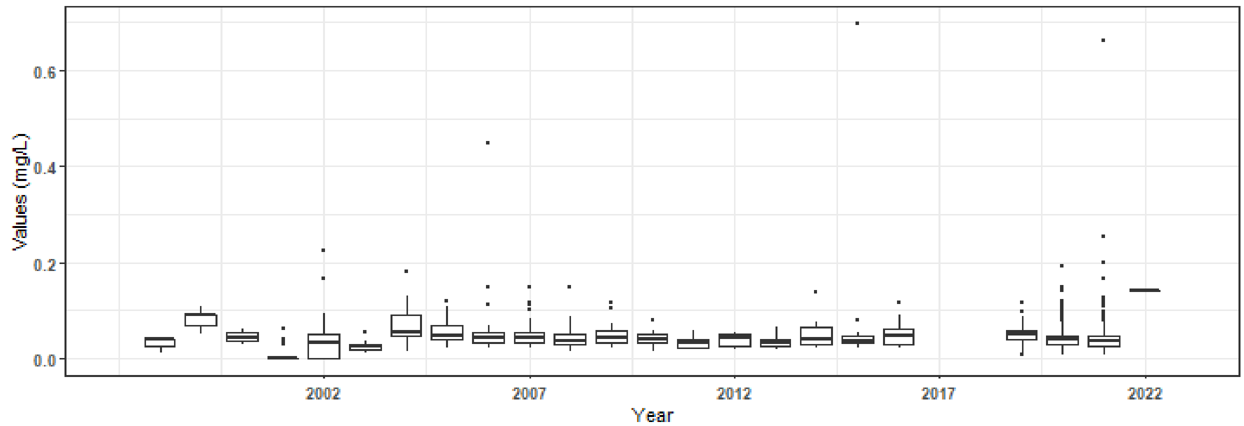


Summary Box Plots for Pellicer Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.2
0.4
0.6
0.8
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.05
0.10
0.15
0.20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.05
0.10
0.15
0.20
3
6
9
12
Month

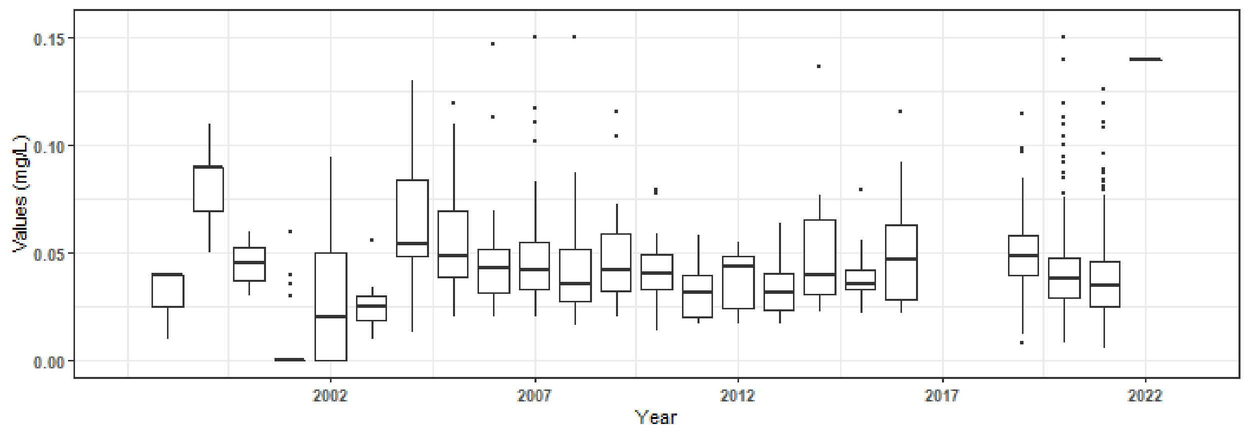
Summary Box Plots for Pine Island Sound Aquatic Preserve

By Year

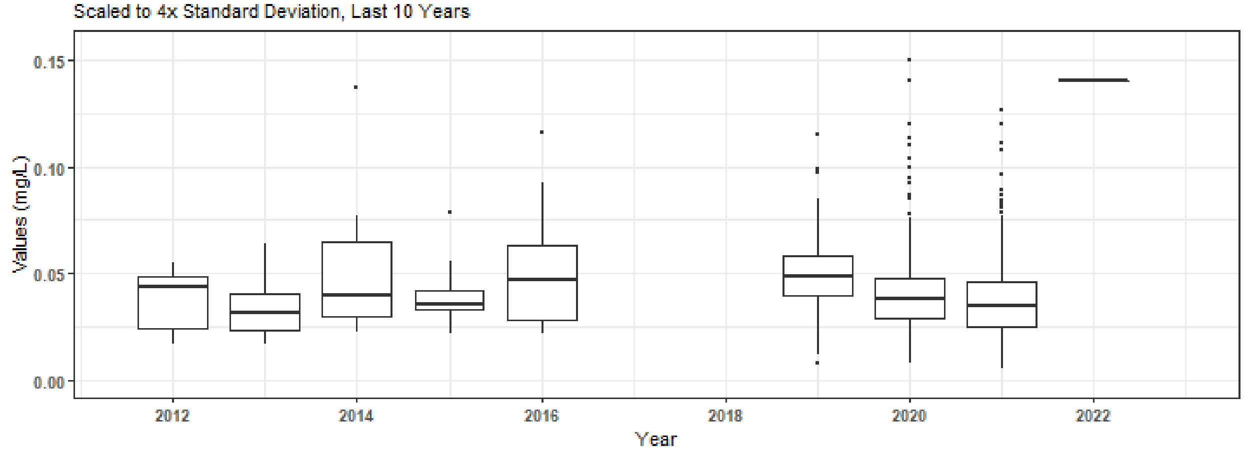
Autoscale



Scaled to 4x Standard Deviation

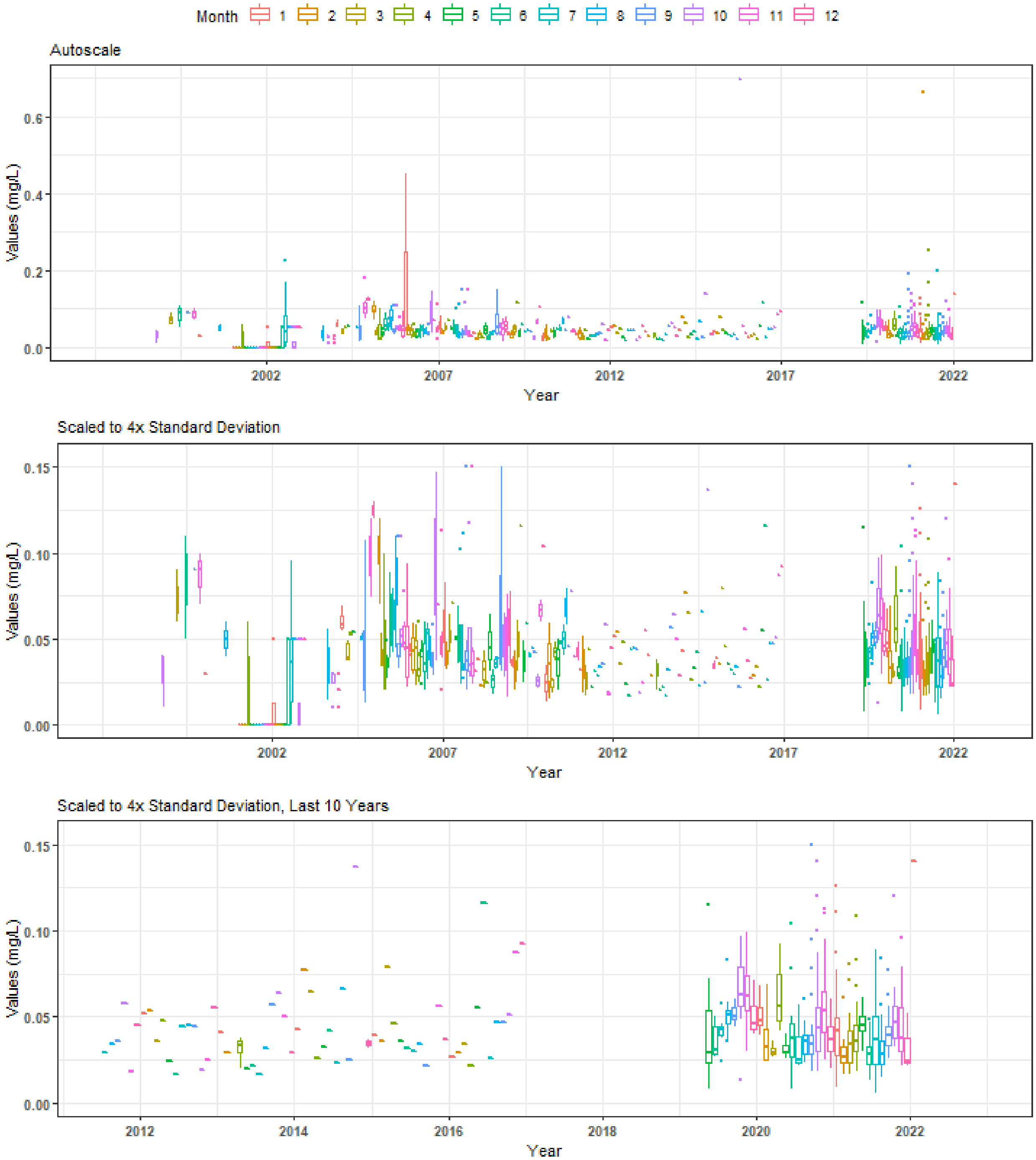


Scaled to 4x Standard Deviation, Last 10 Years



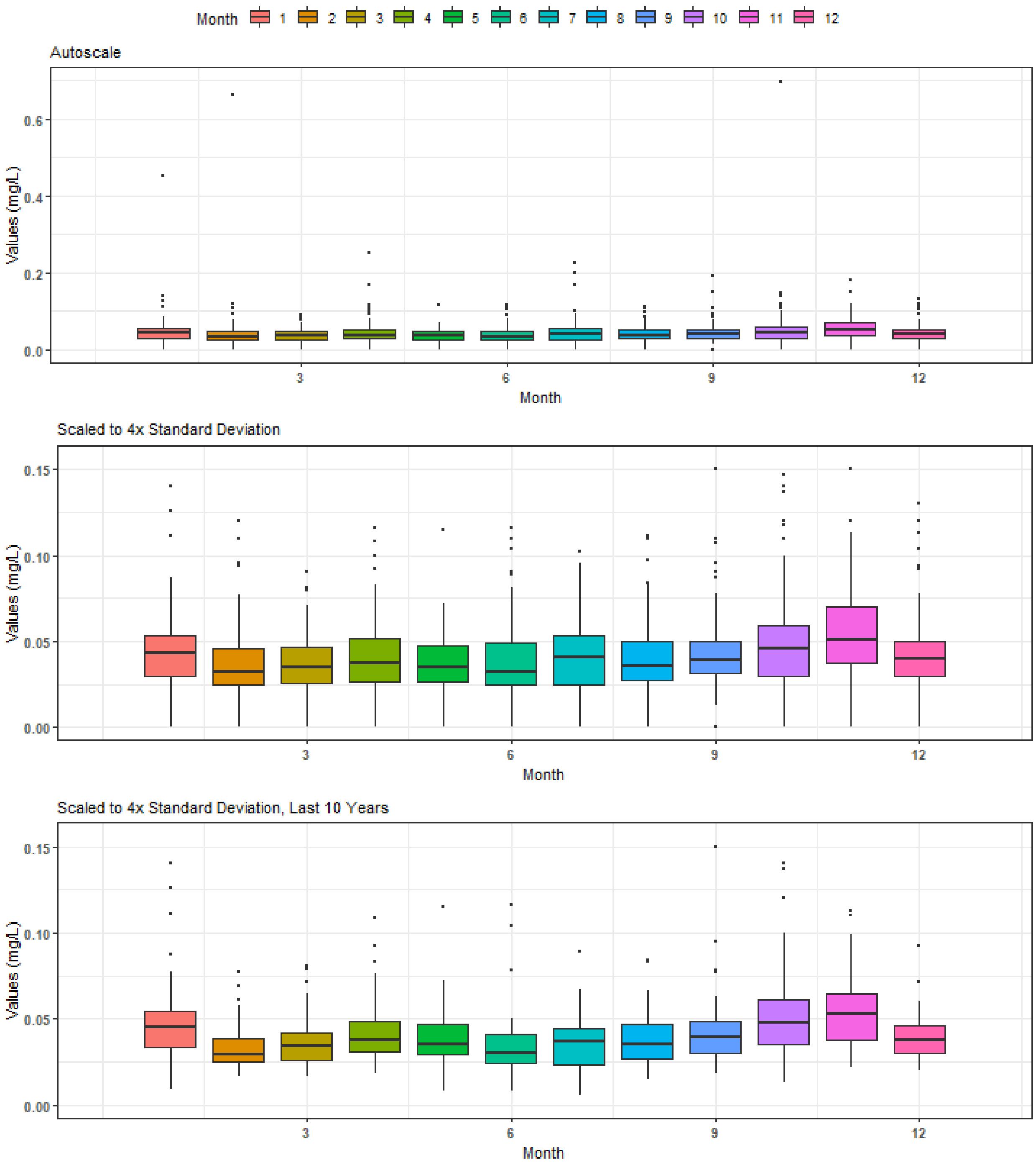
Summary Box Plots for Pine Island Sound Aquatic Preserve

By Year & Month



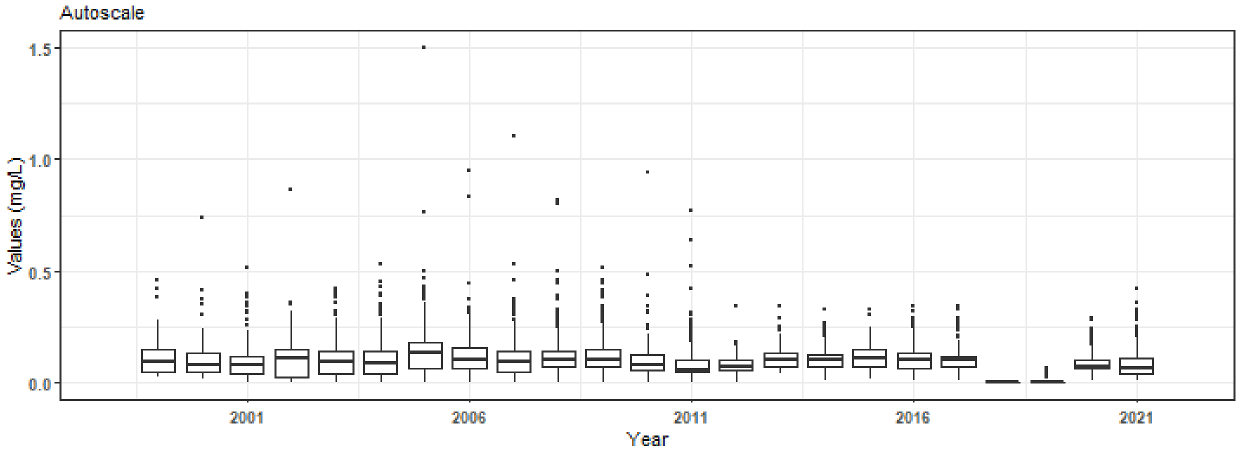
Summary Box Plots for Pine Island Sound Aquatic Preserve

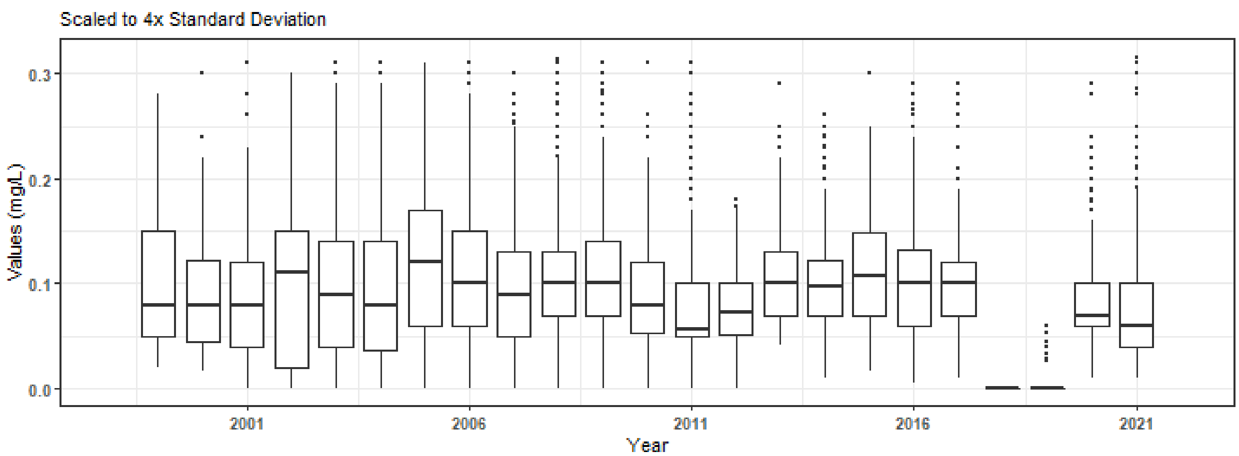
By Month

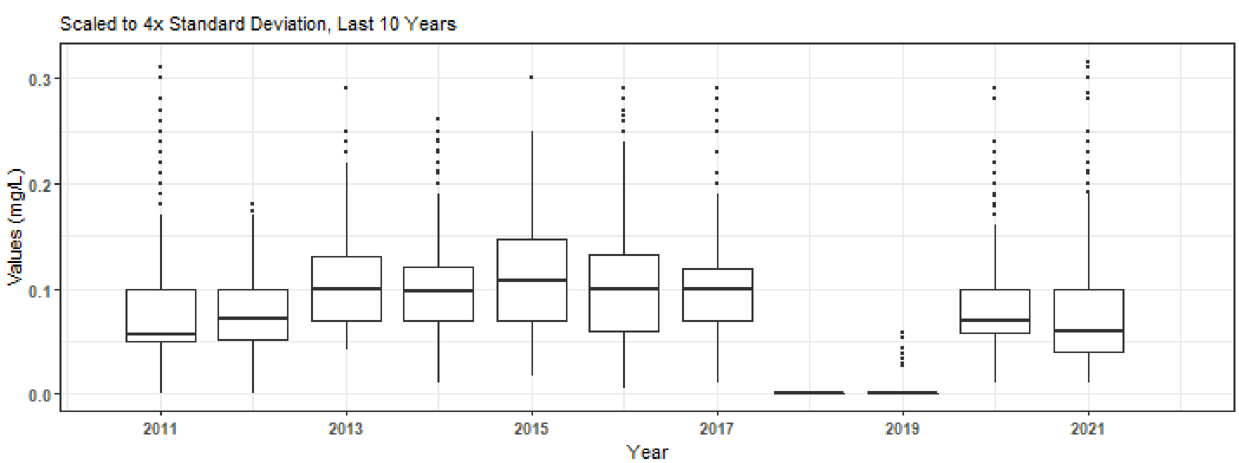


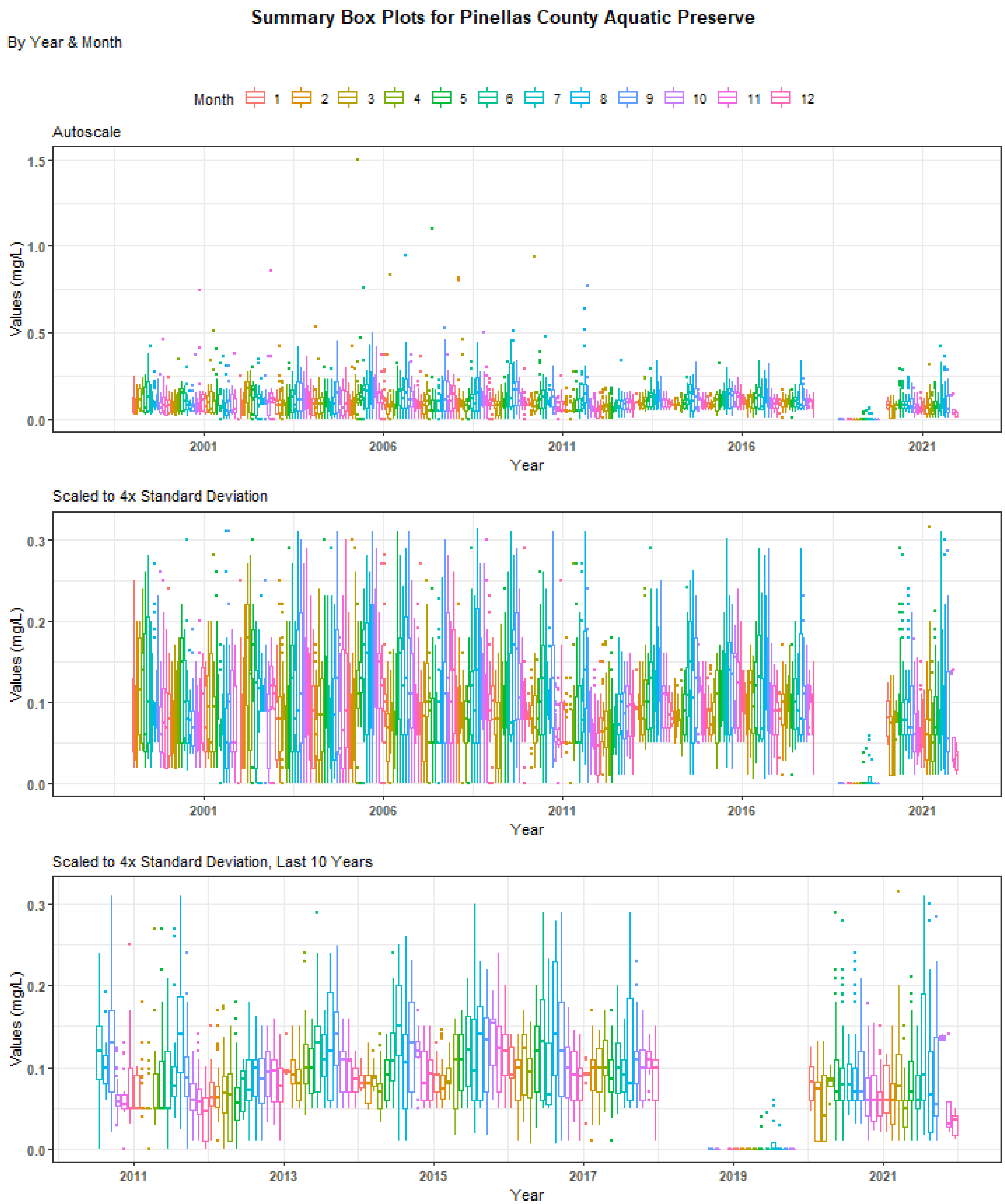
Summary Box Plots for Pinellas County Aquatic Preserve

By Year



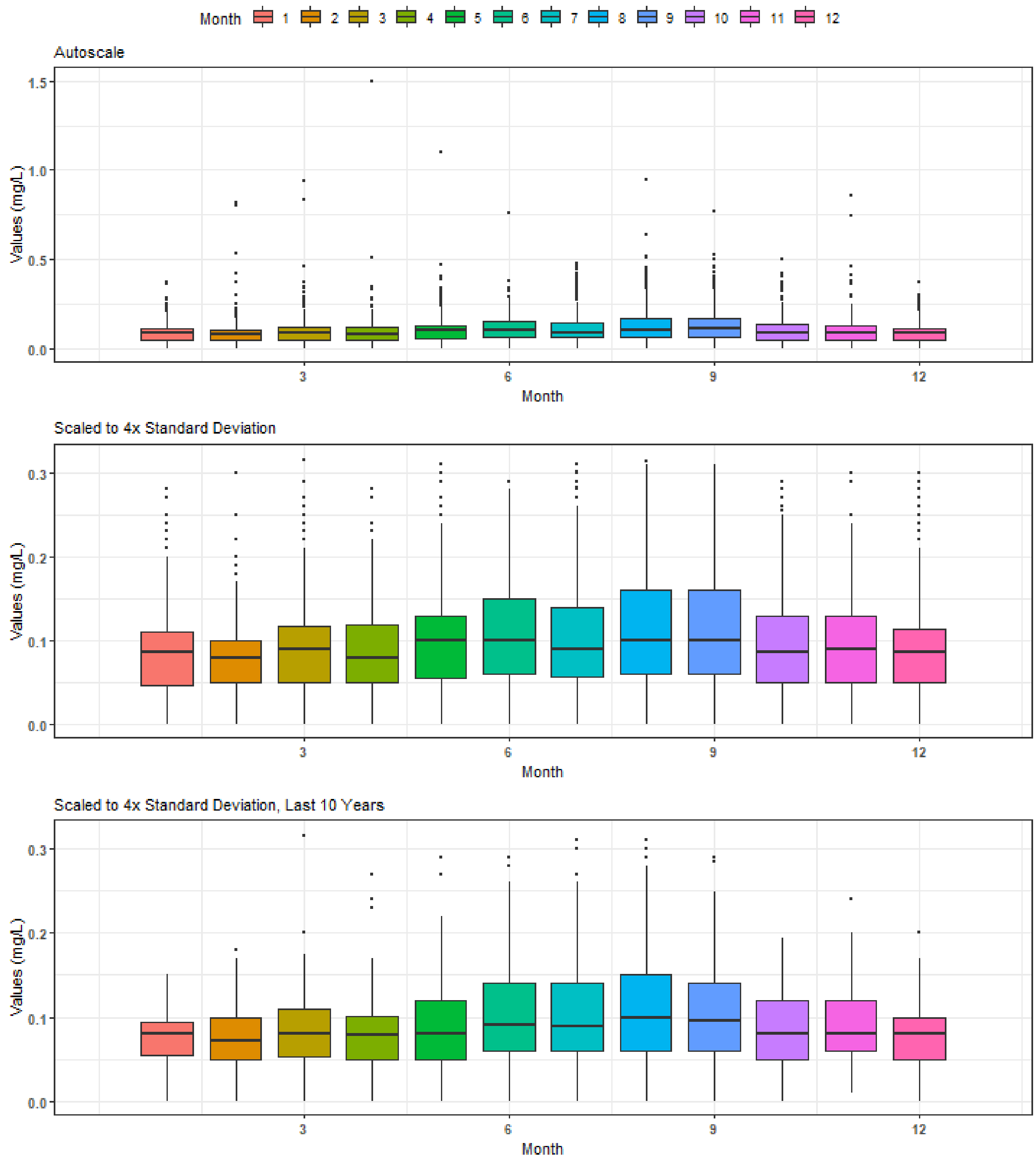





Summary Box Plots for Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

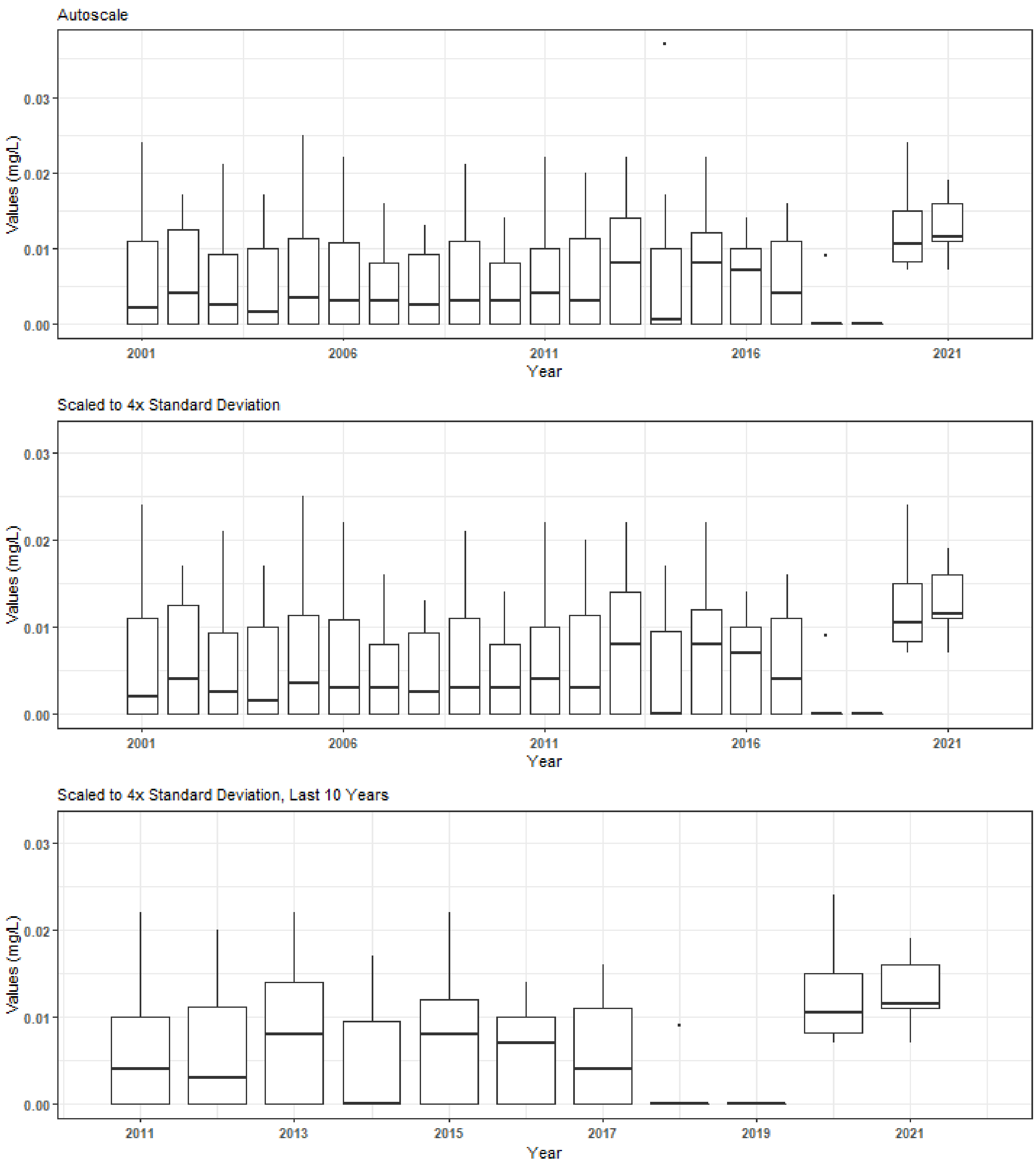
Summary Box Plots for Pinellas County Aquatic Preserve

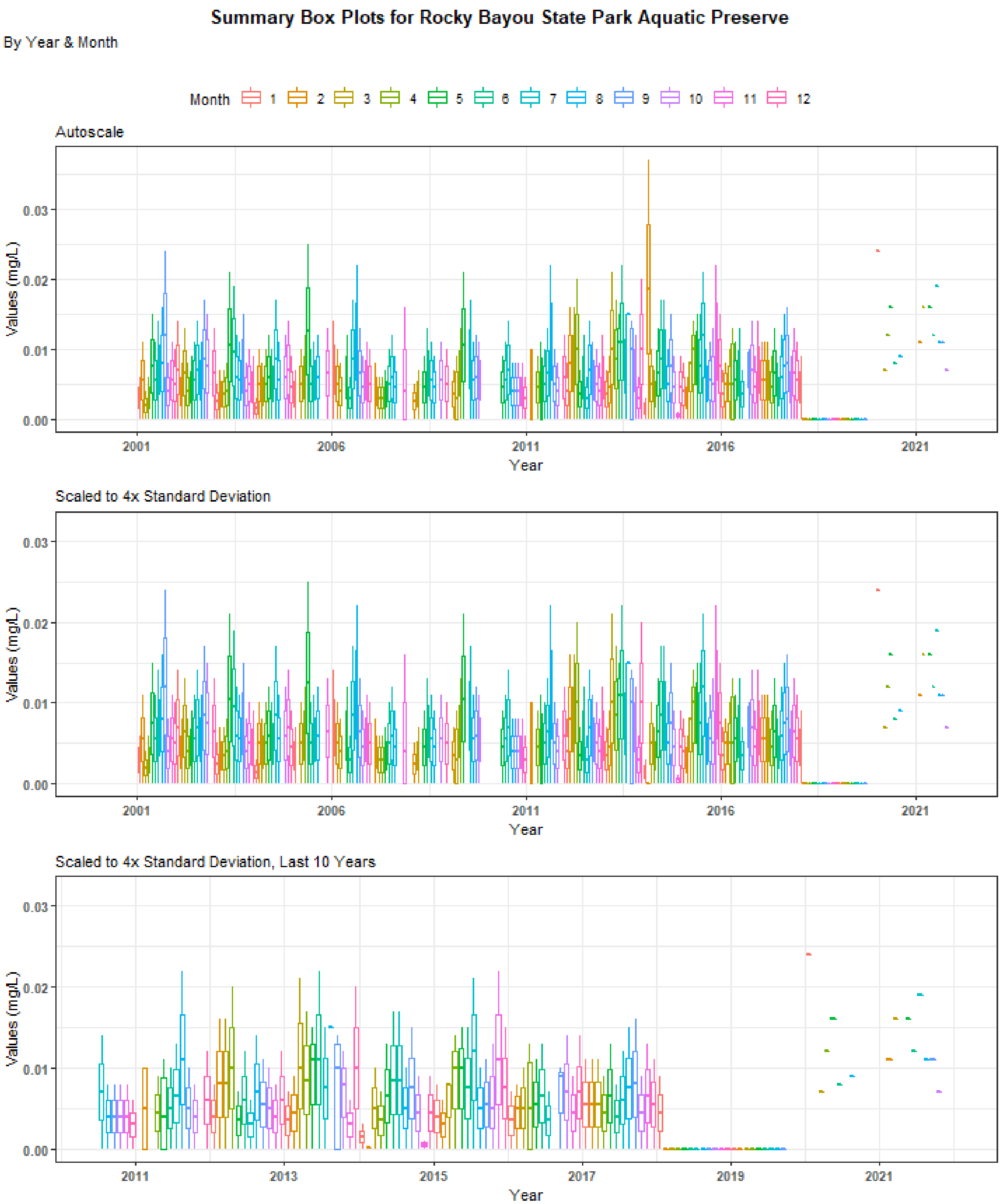
By Month



Summary Box Plots for Rocky Bayou State Park Aquatic Preserve

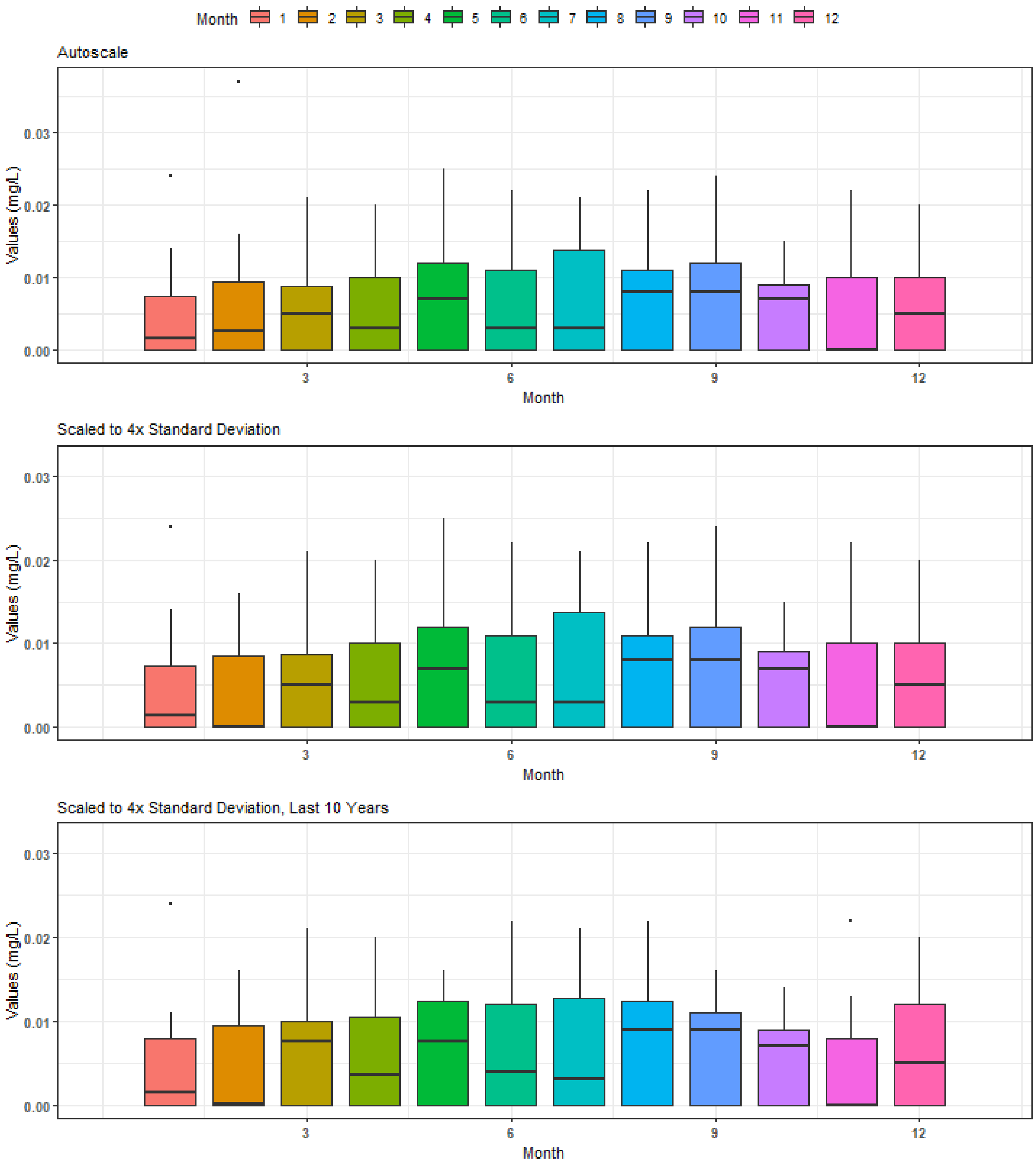
By Year




Summary Box Plots for Rocky Bayou State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.03
0.02
0.01
0.00
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.03
0.02
0.01
0.00
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.03
0.02
0.01
0.00
2011
2013
2015
2017
2019
2021
Year

Summary Box Plots for Rocky Bayou State Park Aquatic Preserve

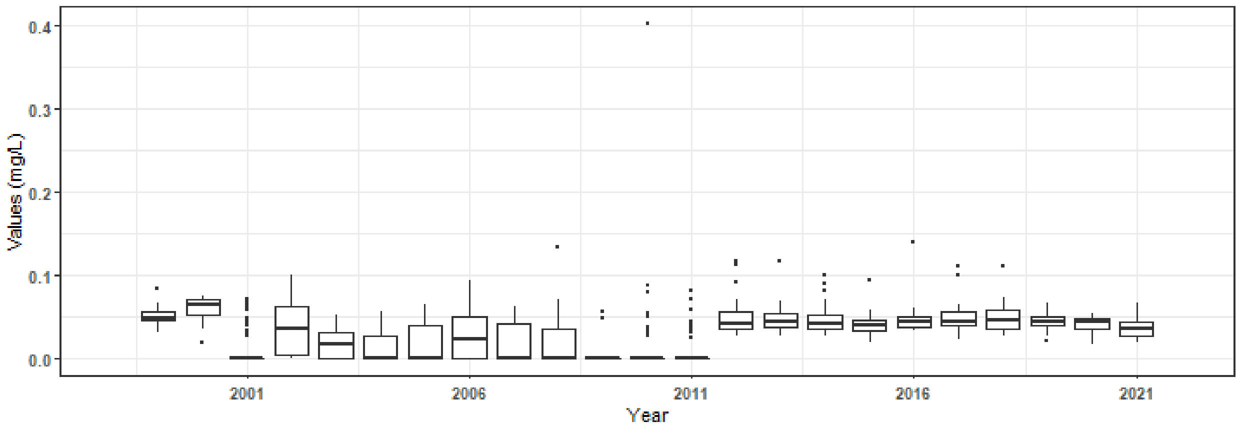
By Month



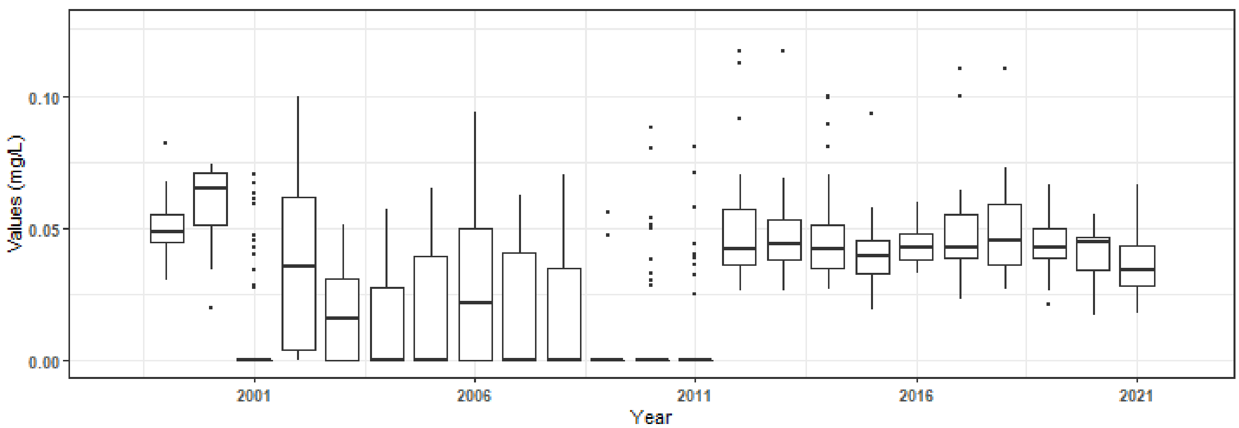
Summary Box Plots for Rookery Bay Aquatic Preserve

By Year

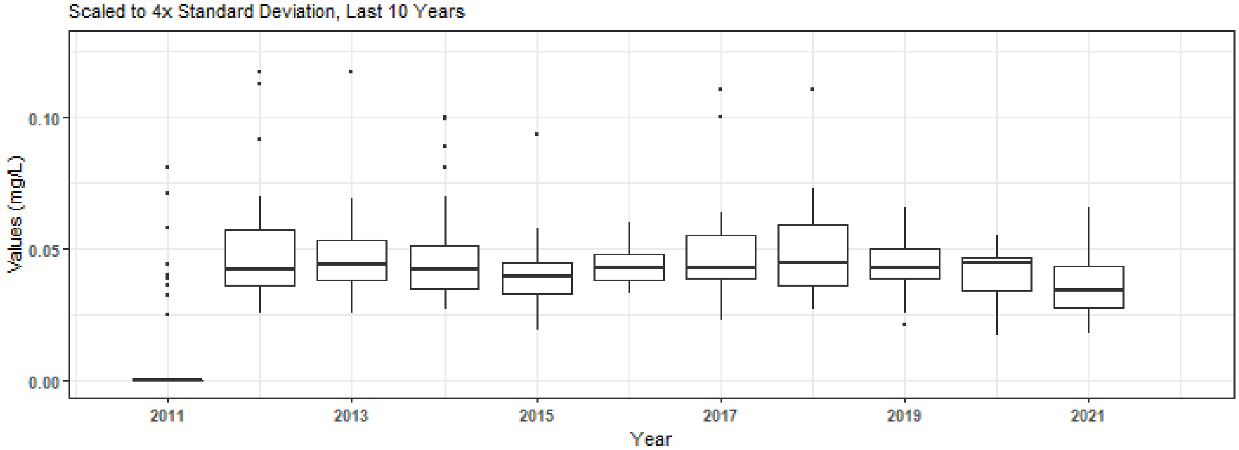
Autoscale

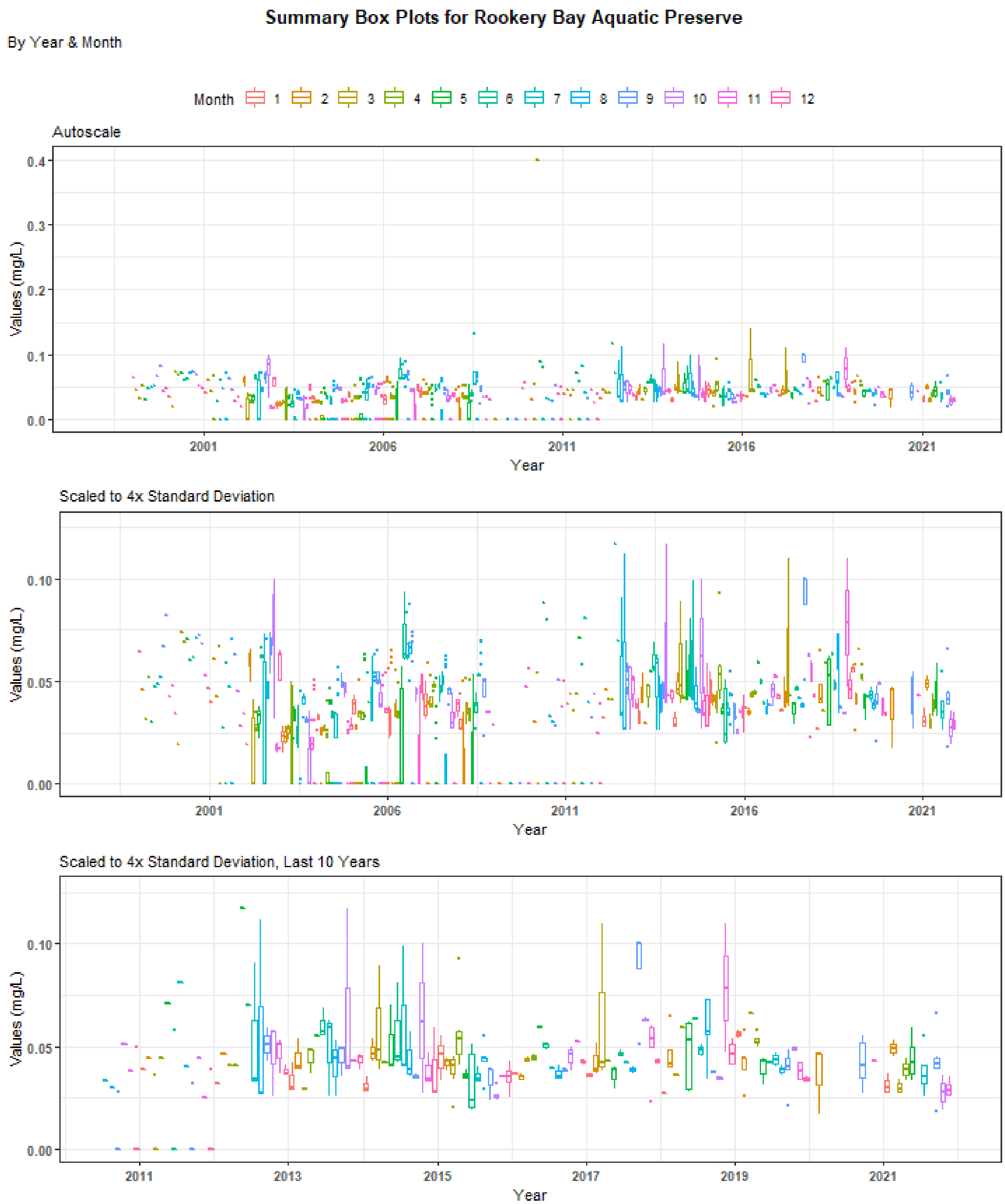


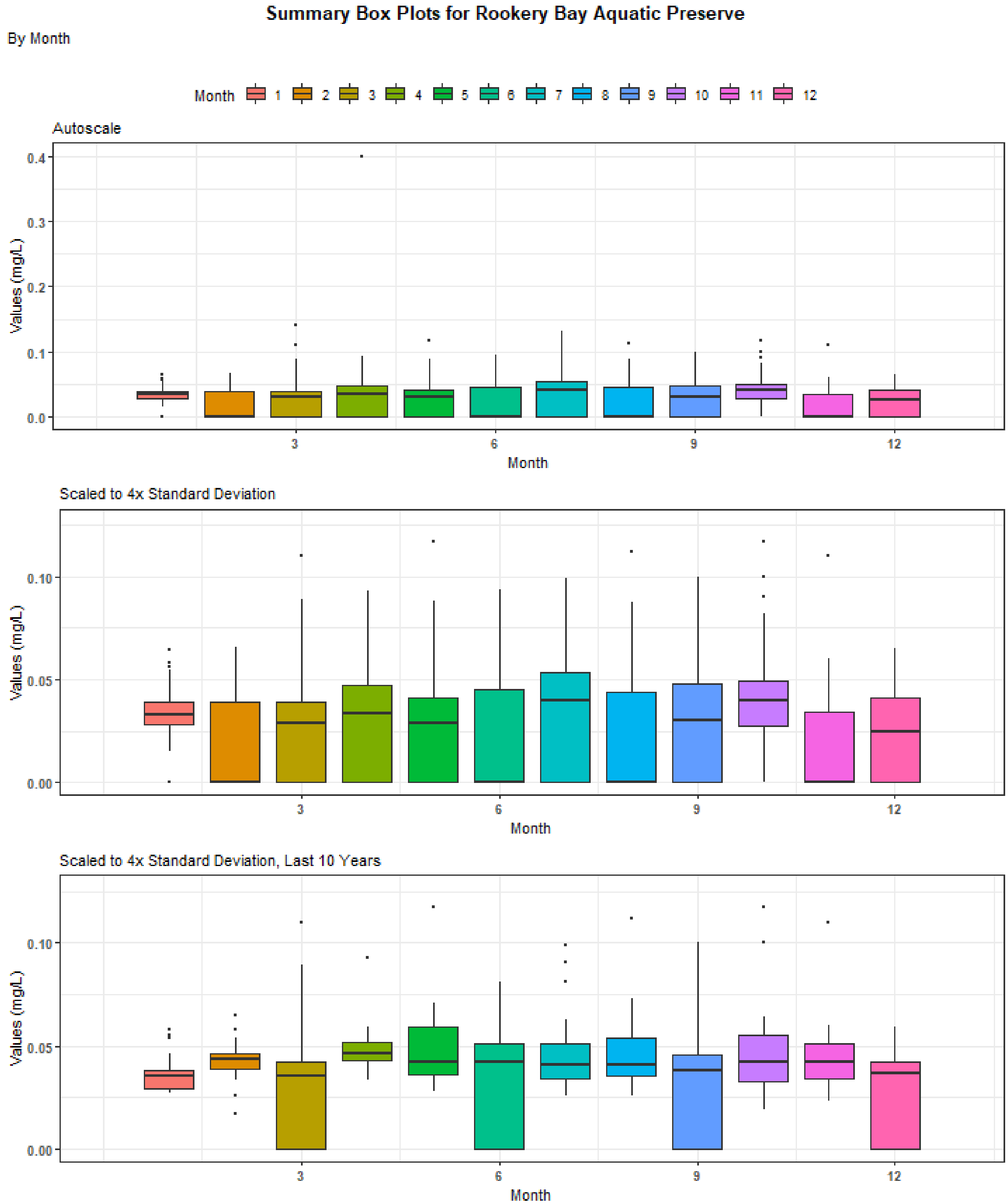
Scaled to 4x Standard Deviation



Scaled to 4x Standard Deviation, Last 10 Years

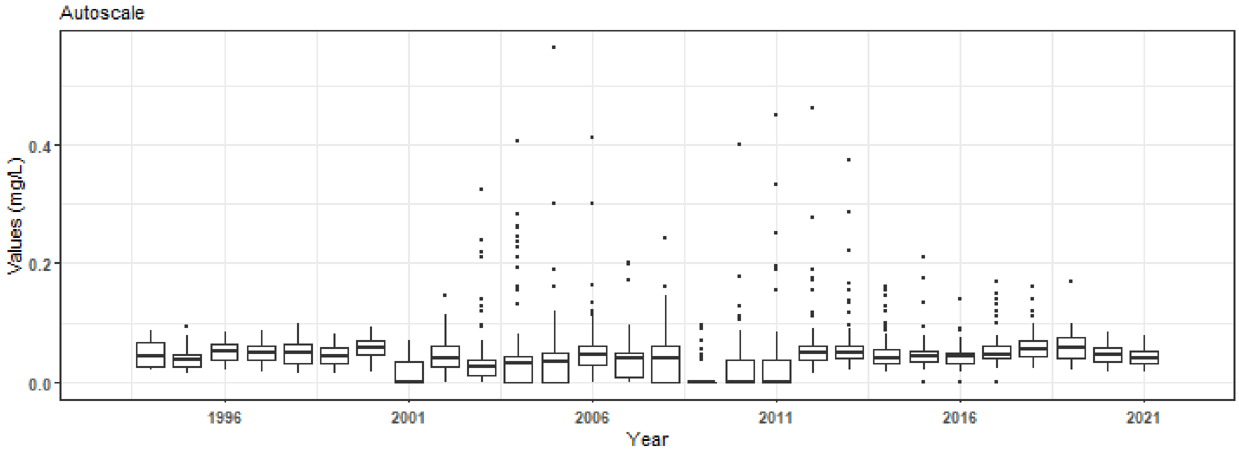


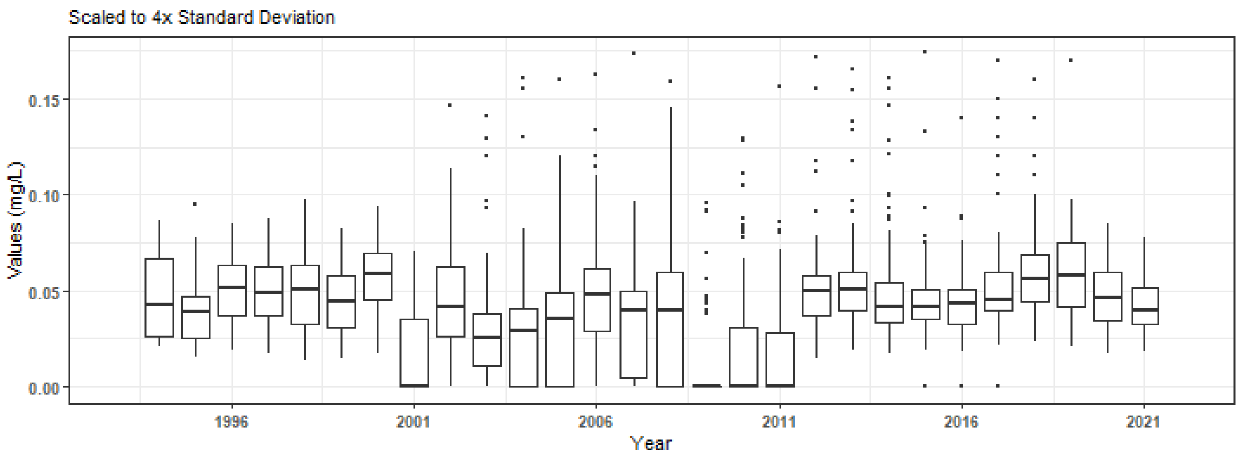

Summary Box Plots for Rookery Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
2011
2013
2015
2017
2019
2021
Year

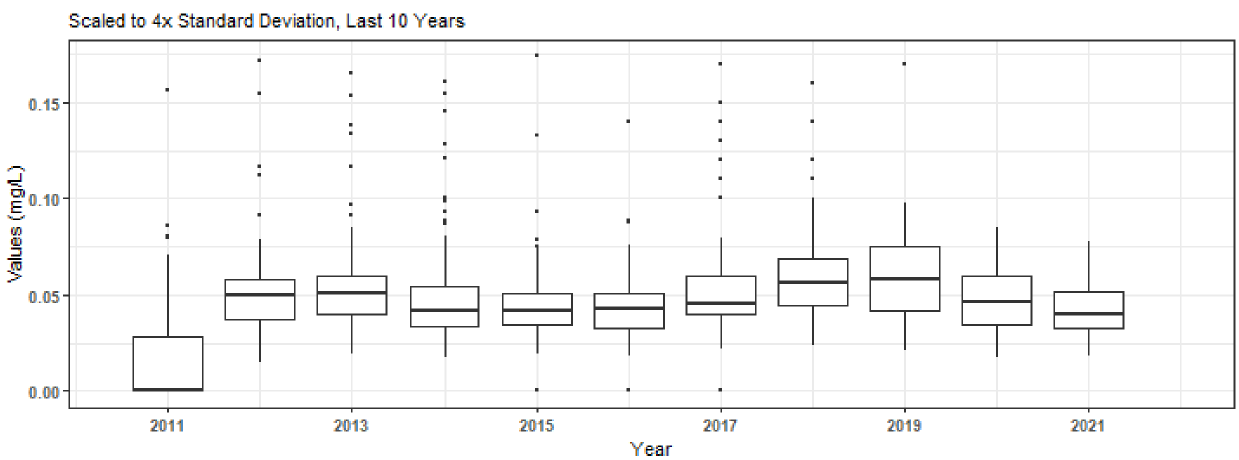

Summary Box Plots for Rookery Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
0.4
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
3
6
9
12
Month

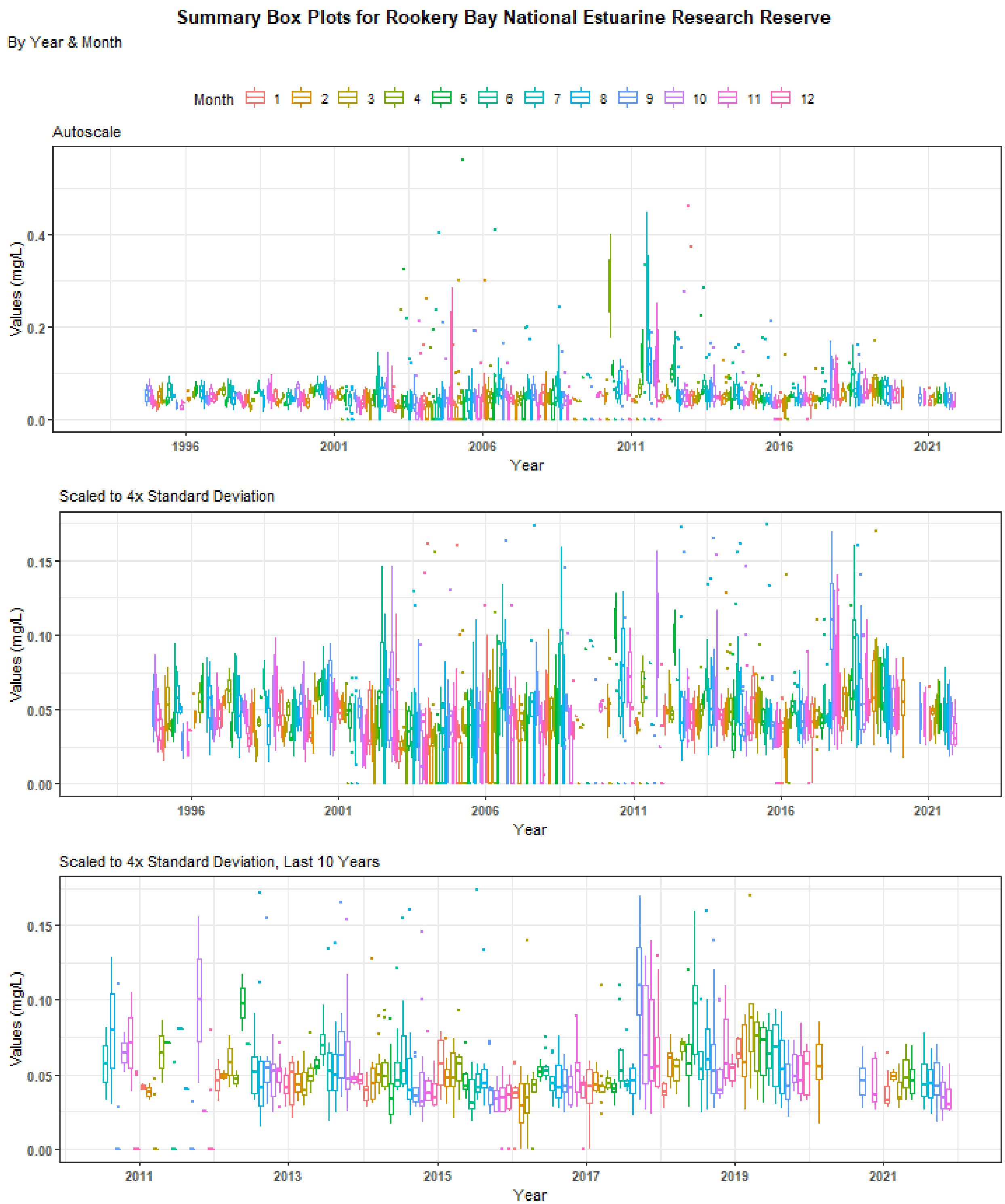
Summary Box Plots for Rookery Bay National Estuarine Research Reserve

By Year



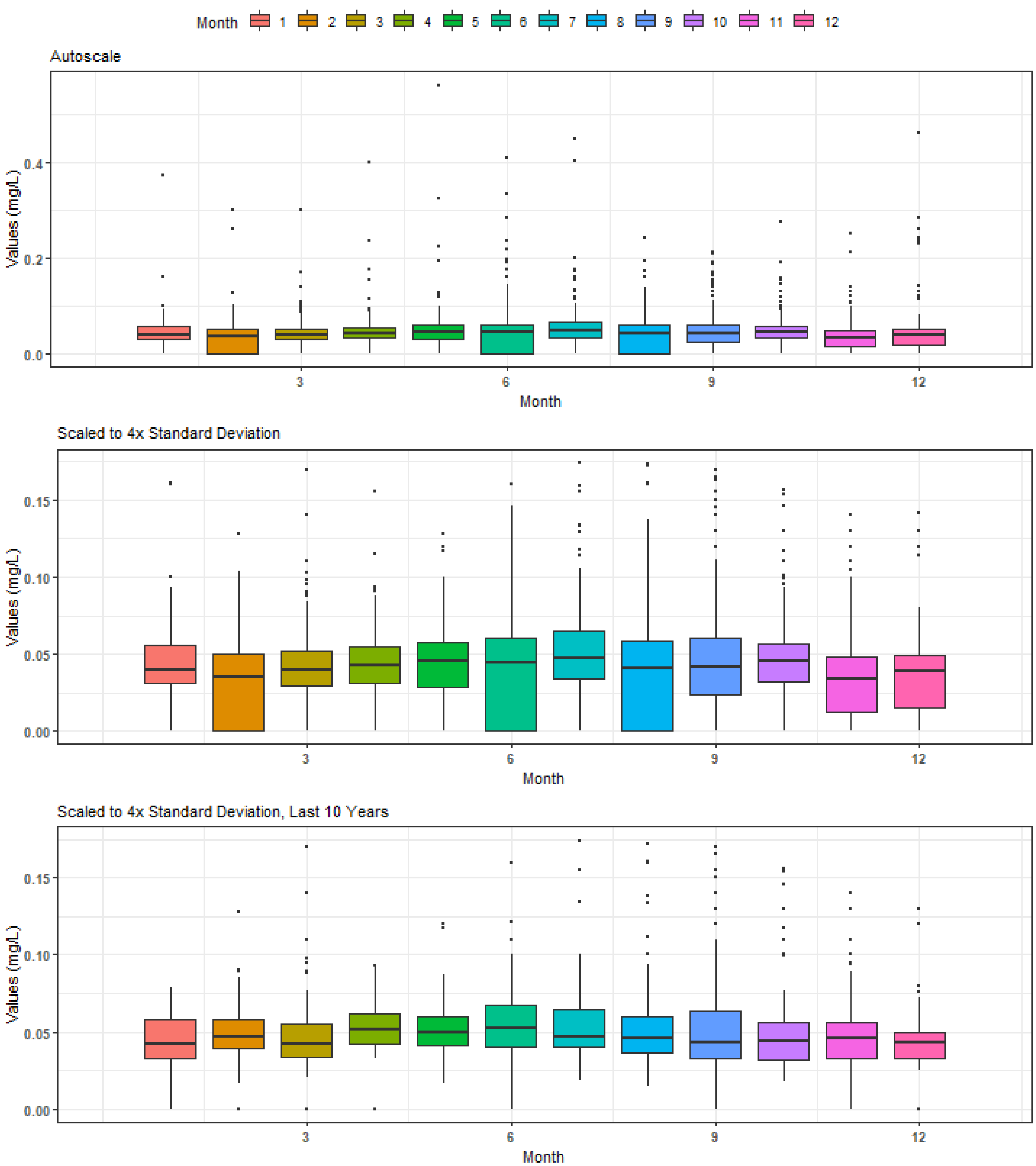


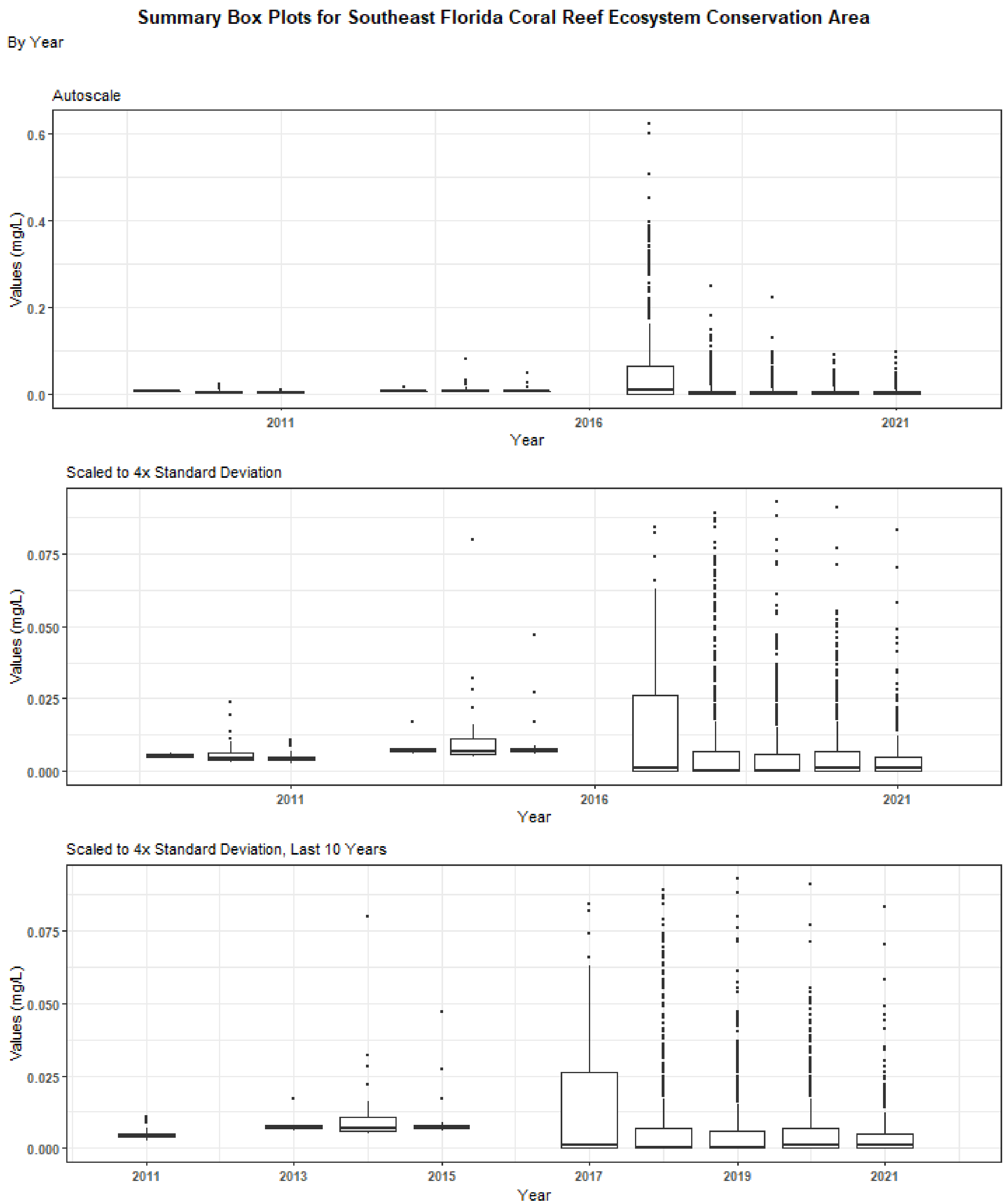



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.2
0.4
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.05
0.10
0.15
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.05
0.10
0.15
2011
2013
2015
2017
2019
2021
Year

Summary Box Plots for Rookery Bay National Estuarine Research Reserve

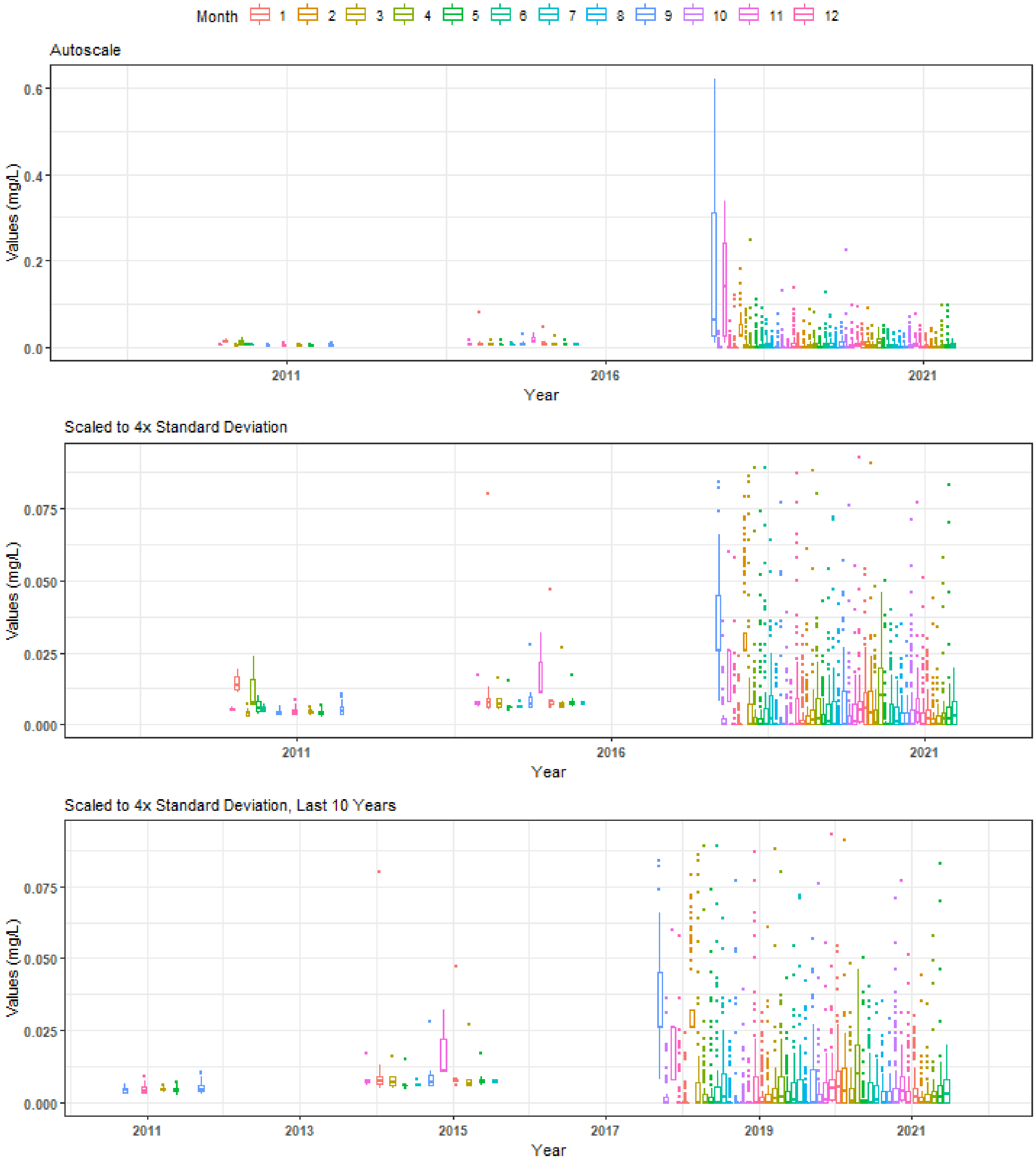
By Month

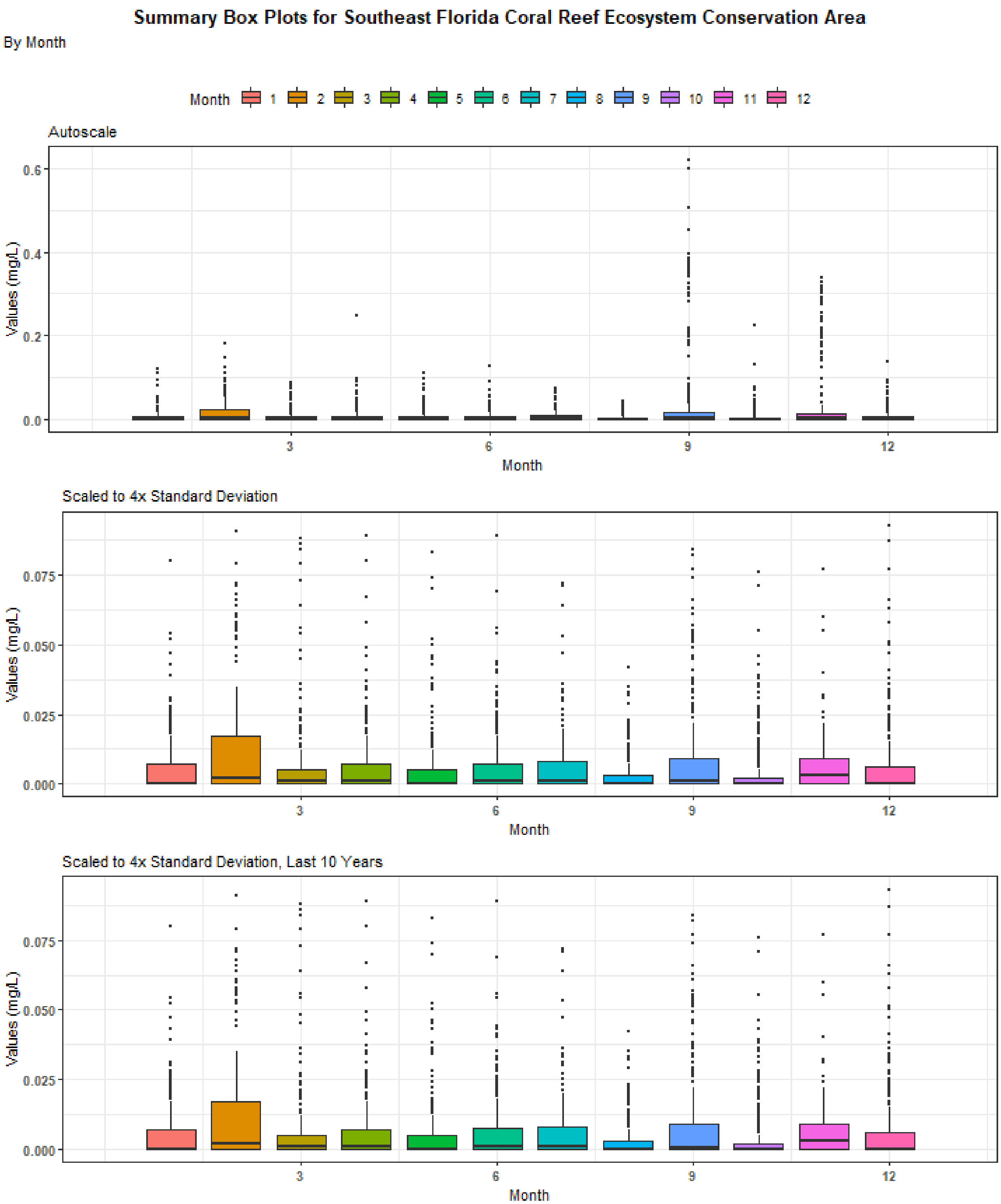



Summary Box Plots for Southeast Florida Coral Reef Ecosystem Conservation Area
By Year
Autoscale
Values (mg/L)
0.6
0.4
0.2
0.0
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.075
0.050
0.025
0.000
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.075
0.050
0.025
0.000
2011
2013
2015
2017
2019
2021
Year

Summary Box Plots for Southeast Florida Coral Reef Ecosystem Conservation Area

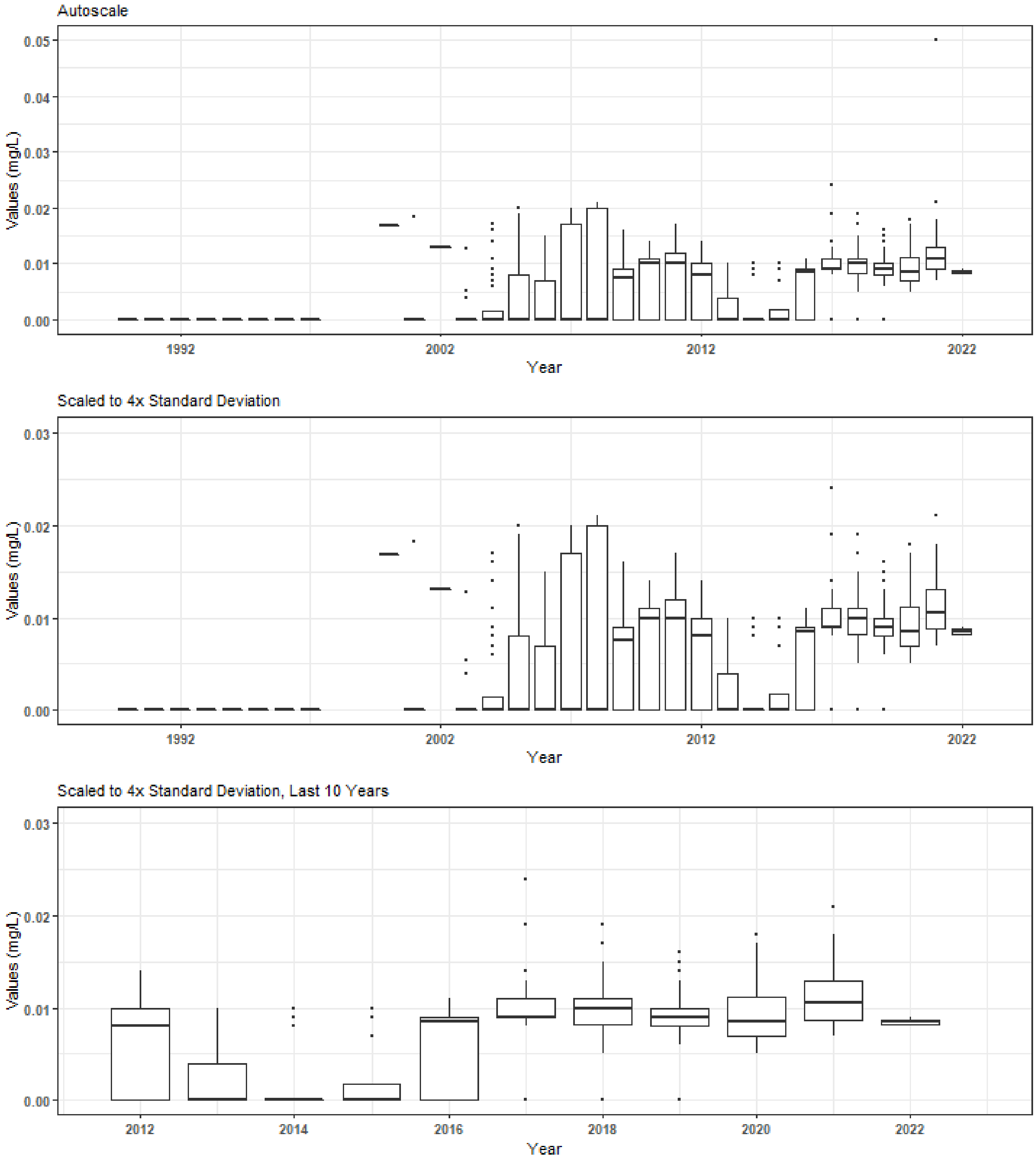
By Year & Month

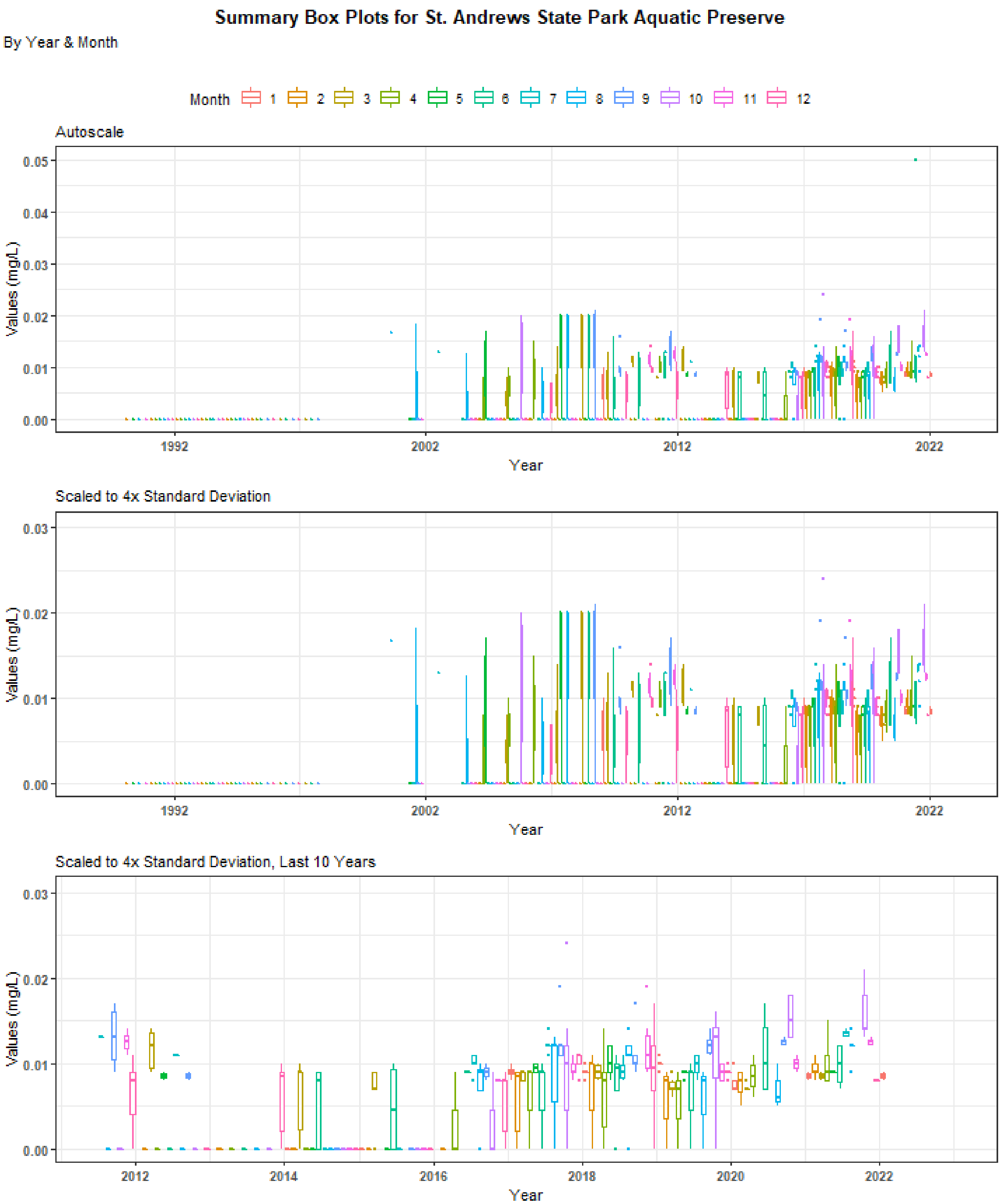



Summary Box Plots for Southeast Florida Coral Reef Ecosystem Conservation Area
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.2
0.4
0.6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.000
0.025
0.050
0.075
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.000
0.025
0.050
0.075
3
6
9
12
Month

Summary Box Plots for St. Andrews State Park Aquatic Preserve

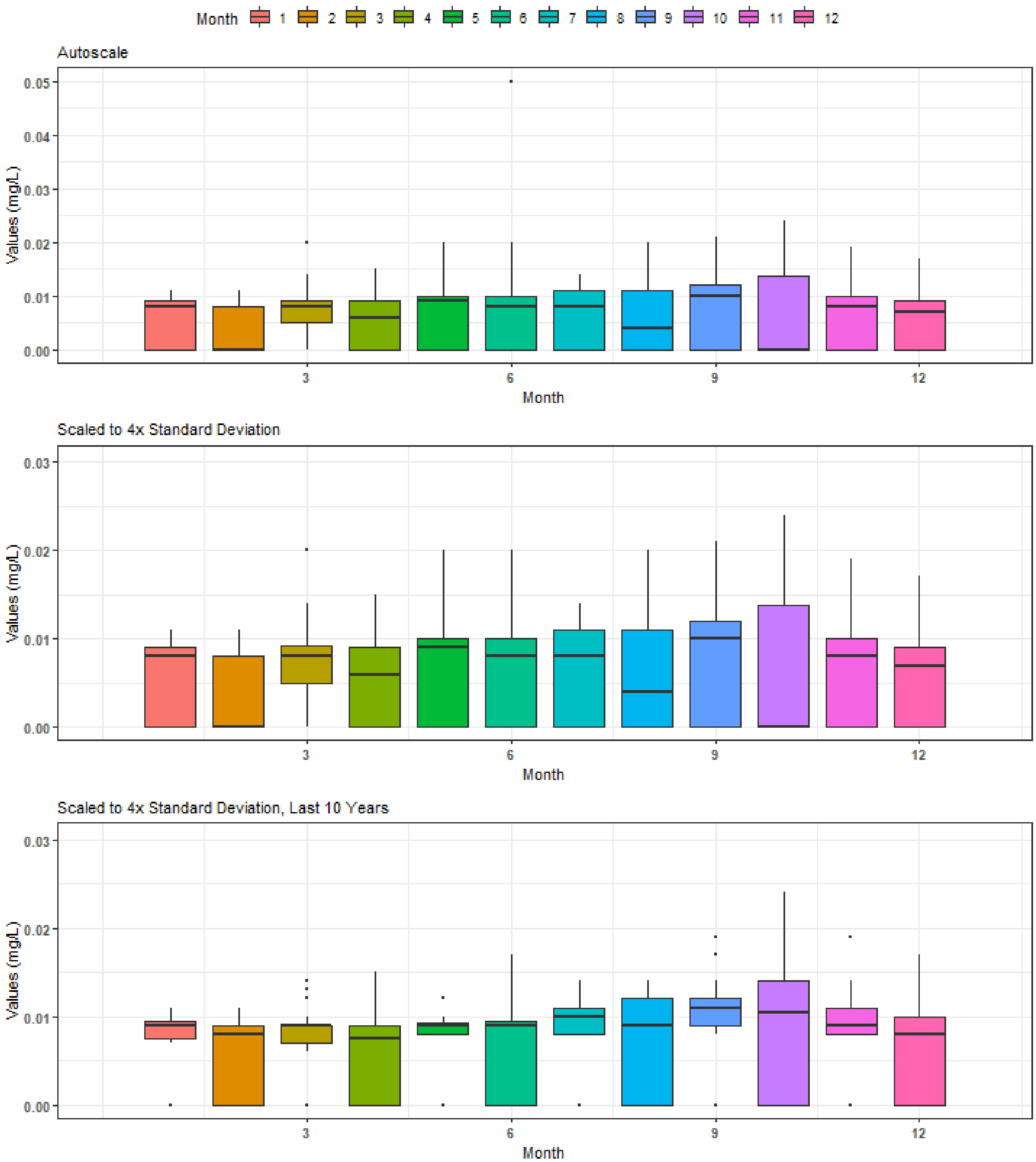
By Year




Summary Box Plots for St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.05
0.04
0.03
0.02
0.01
0.00
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.03
0.02
0.01
0.00
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.03
0.02
0.01
0.00
2012
2014
2016
2018
2020
2022
Year

Summary Box Plots for St. Andrews State Park Aquatic Preserve

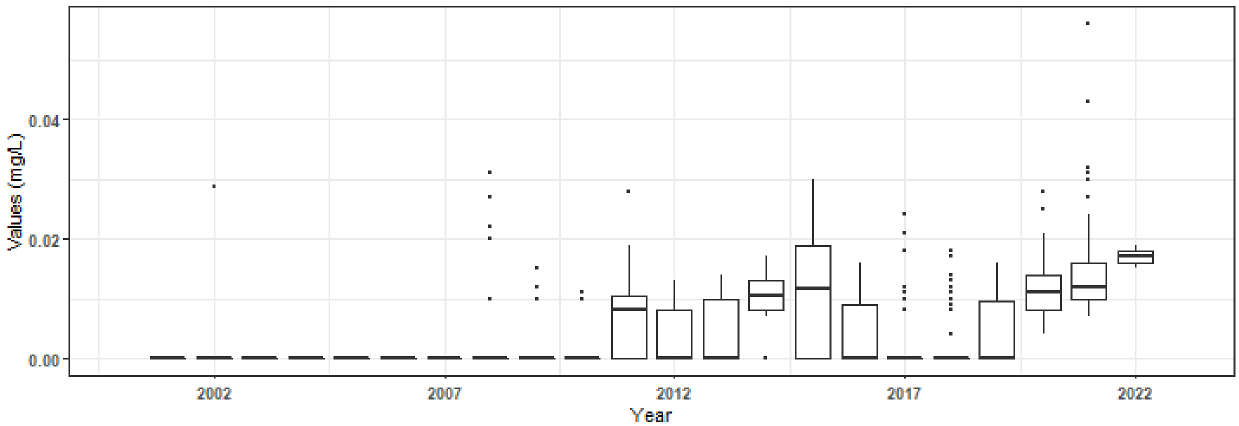
By Month



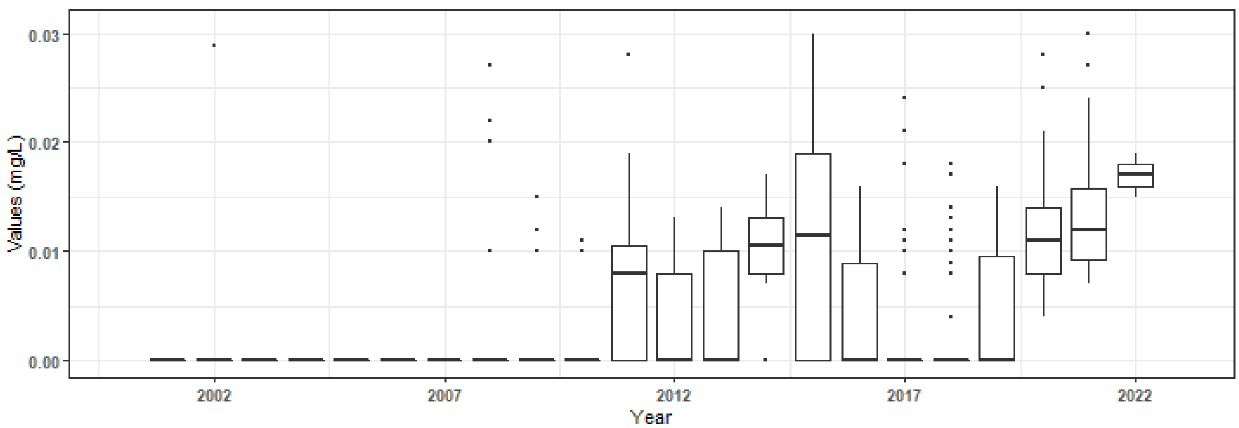
Summary Box Plots for St. Joseph Bay Aquatic Preserve

By Year

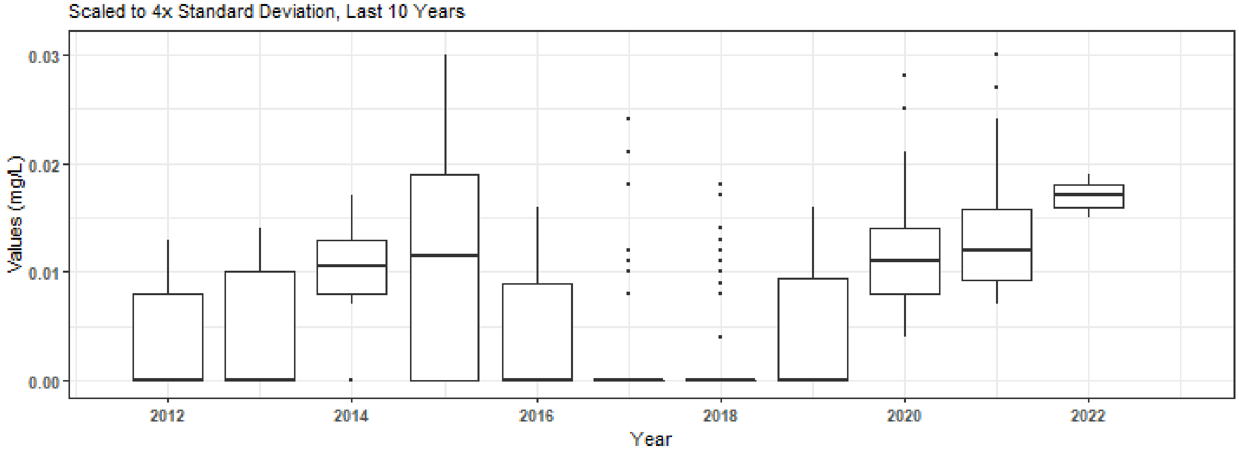
Autoscale

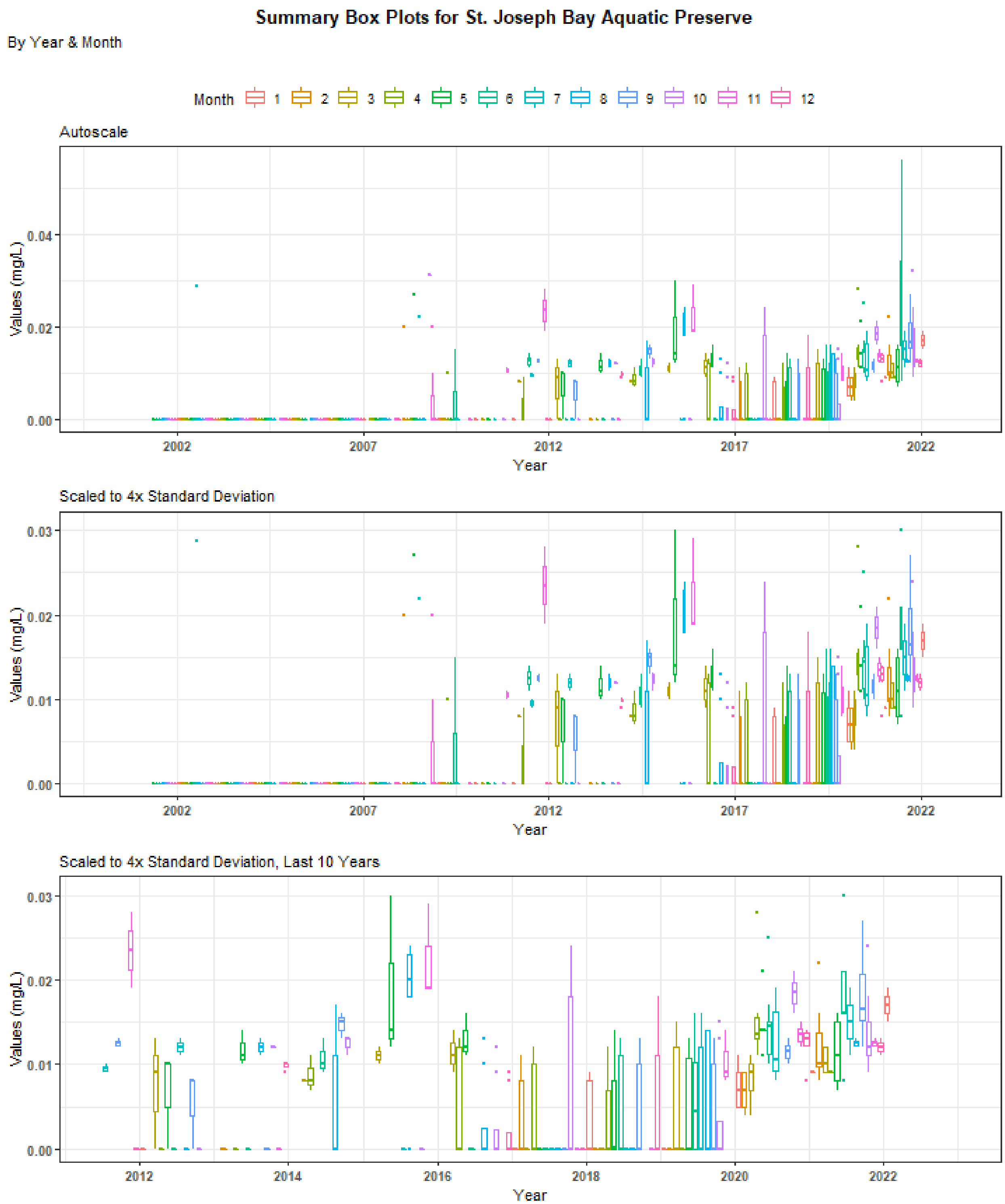


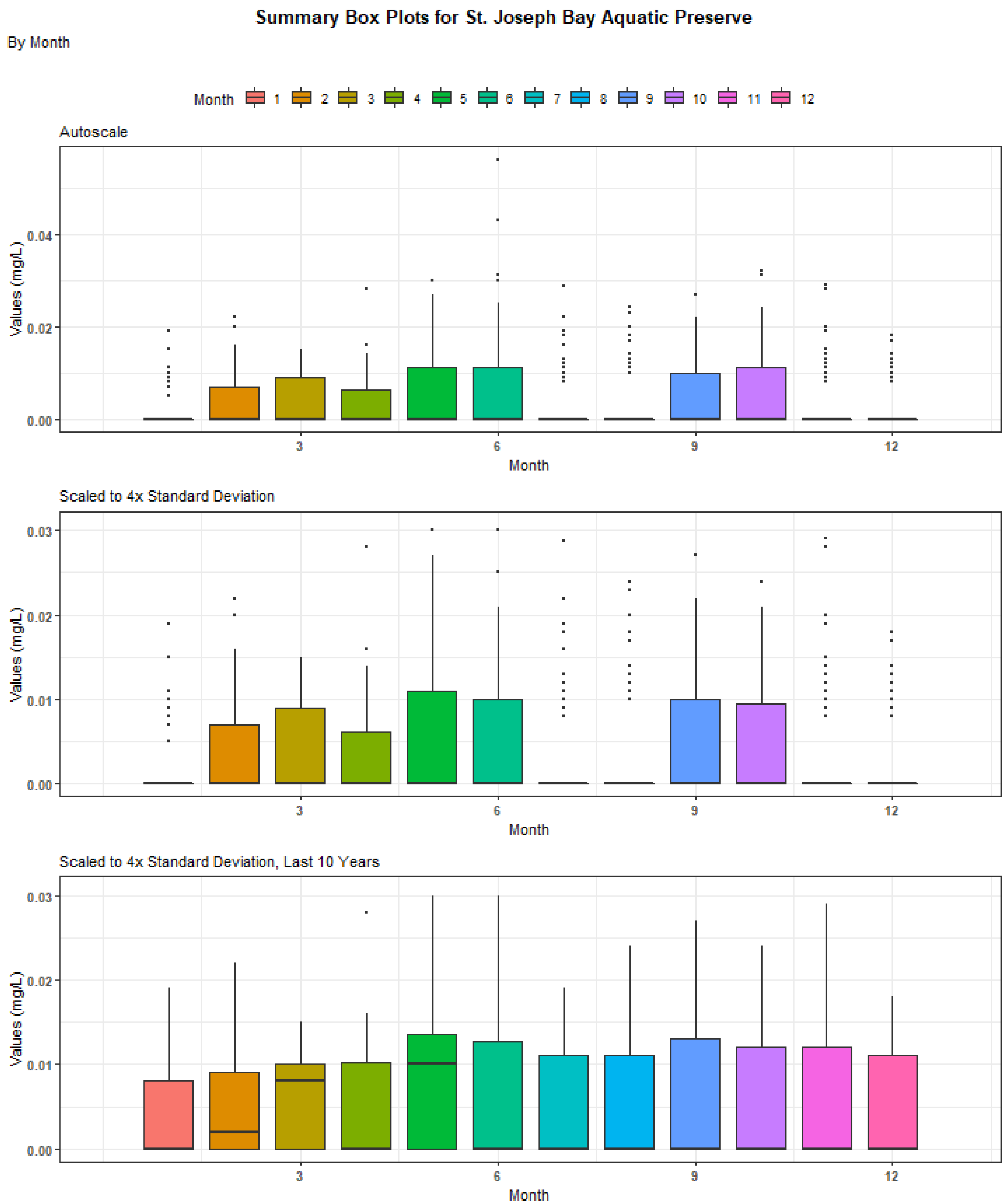
Scaled to 4x Standard Deviation



Scaled to 4x Standard Deviation, Last 10 Years



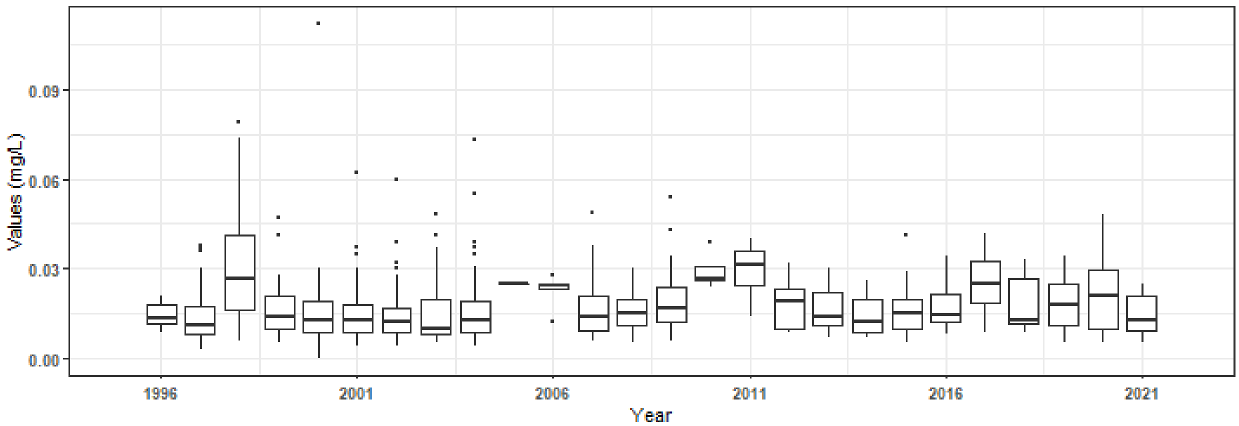

Summary Box Plots for St. Joseph Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.00
0.02
0.04
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.01
0.02
0.03
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.01
0.02
0.03
2012
2014
2016
2018
2020
2022
Year


Summary Box Plots for St. Joseph Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.00
0.02
0.04
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.01
0.02
0.03
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.01
0.02
0.03
3
6
9
12
Month

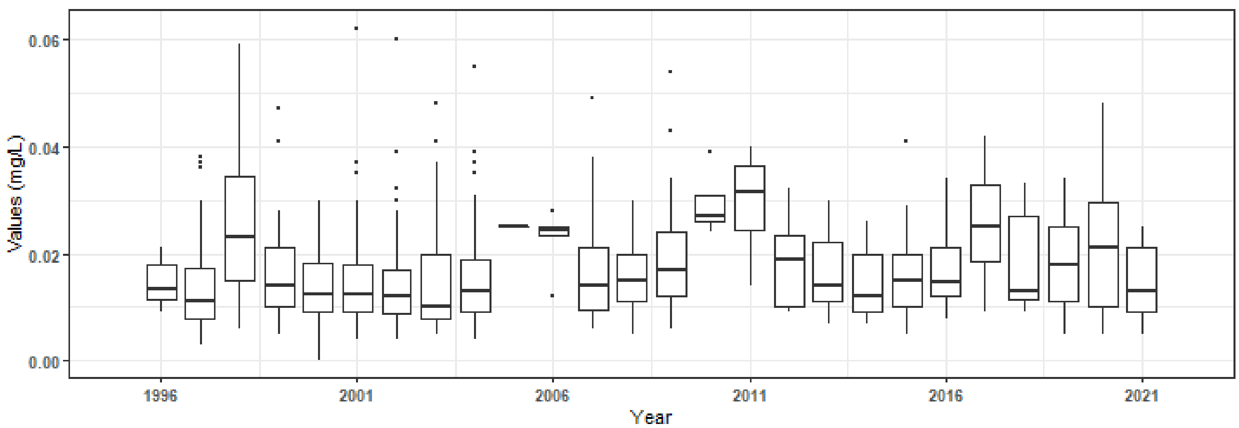
Summary Box Plots for St. Martins Marsh Aquatic Preserve

By Year

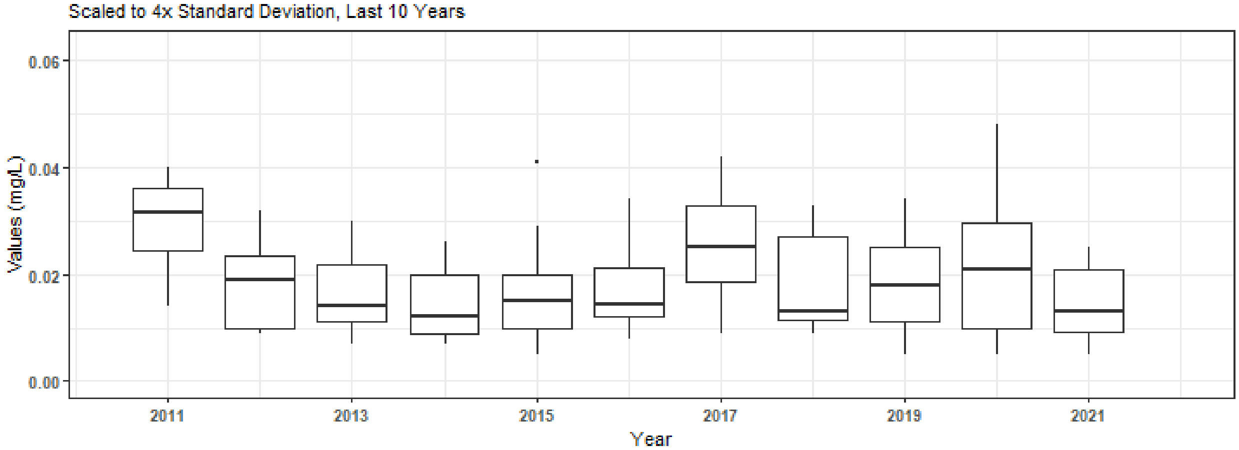
Autoscale



Scaled to 4x Standard Deviation

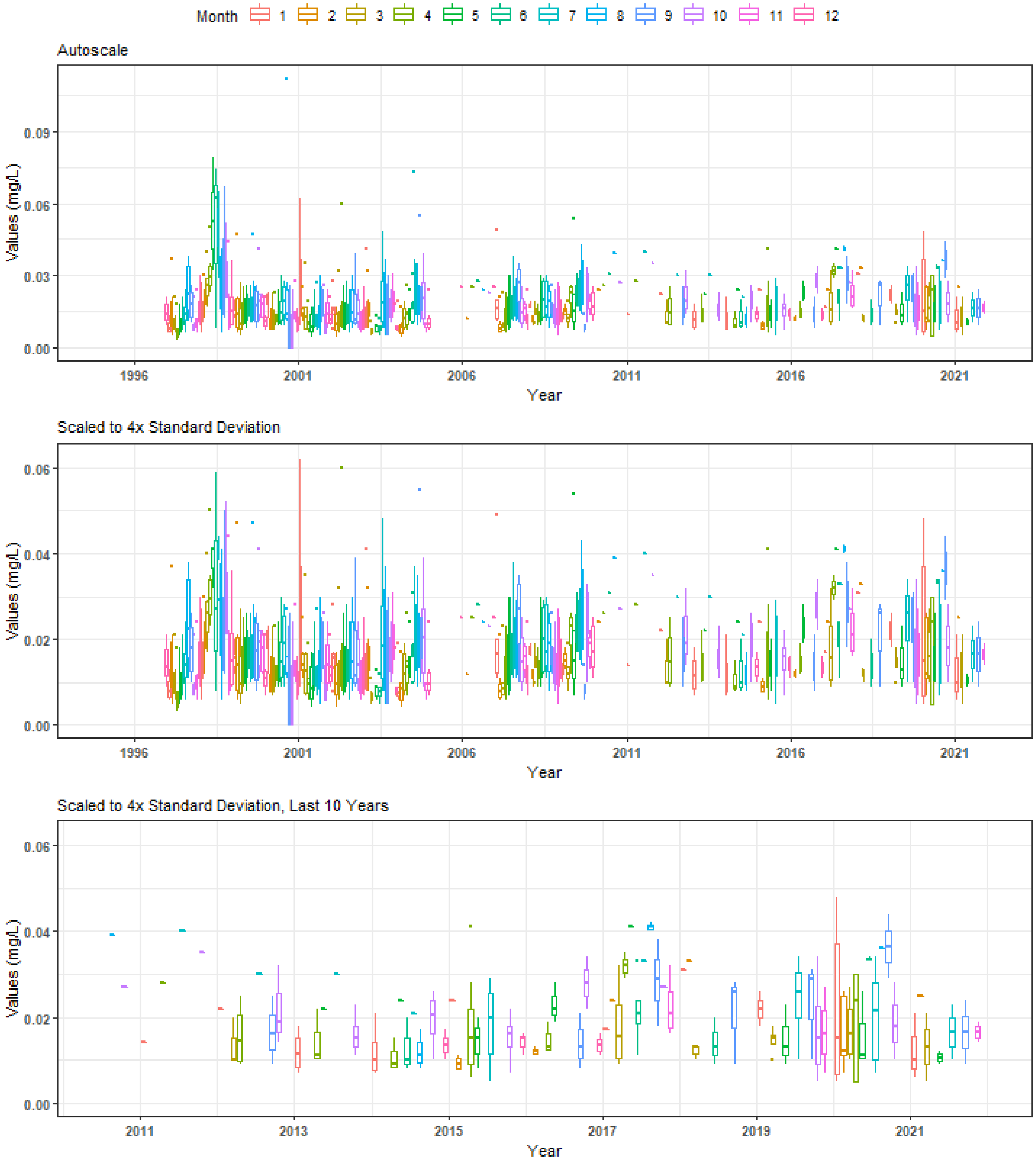


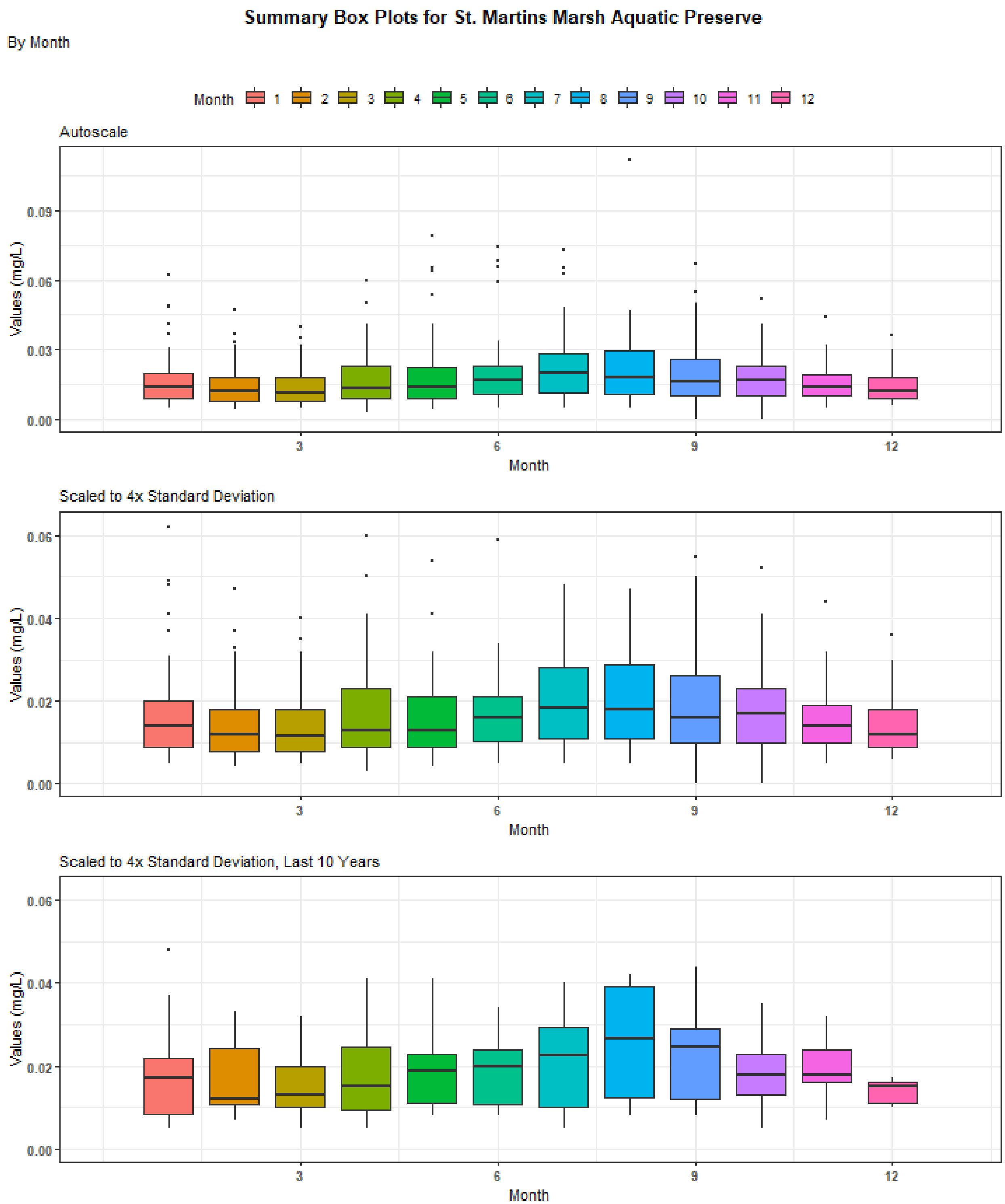
Scaled to 4x Standard Deviation, Last 10 Years



Summary Box Plots for St. Martins Marsh Aquatic Preserve

By Year & Month

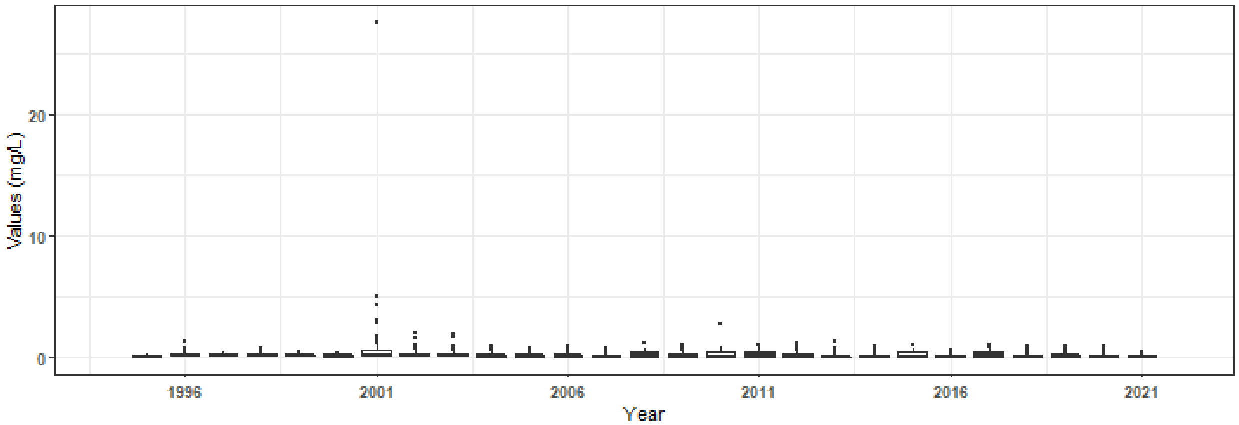



Summary Box Plots for St. Martins Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.00
0.03
0.06
0.09
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0.00
0.02
0.04
0.06
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.00
0.02
0.04
0.06
3
6
9
12
Month

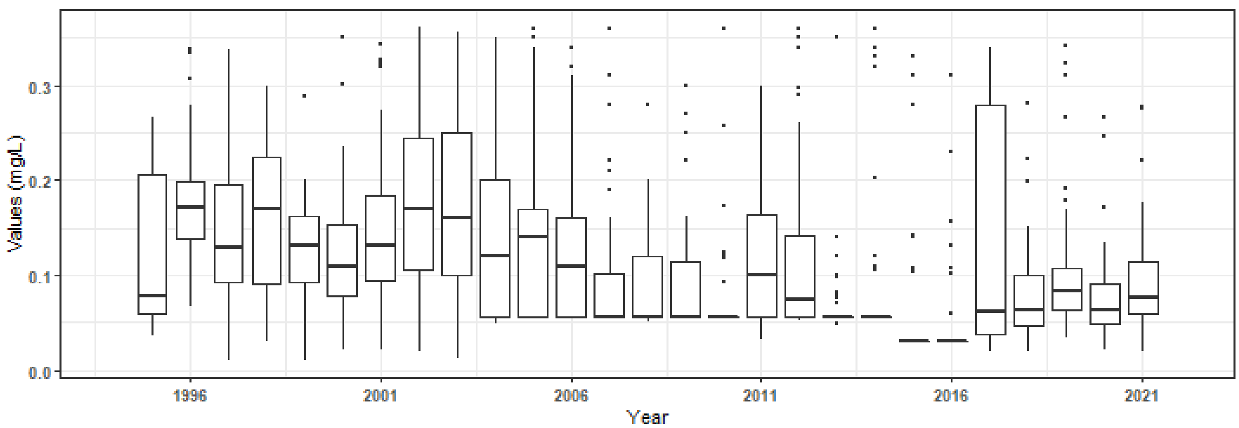
Summary Box Plots for Terra Ceia Aquatic Preserve

By Year

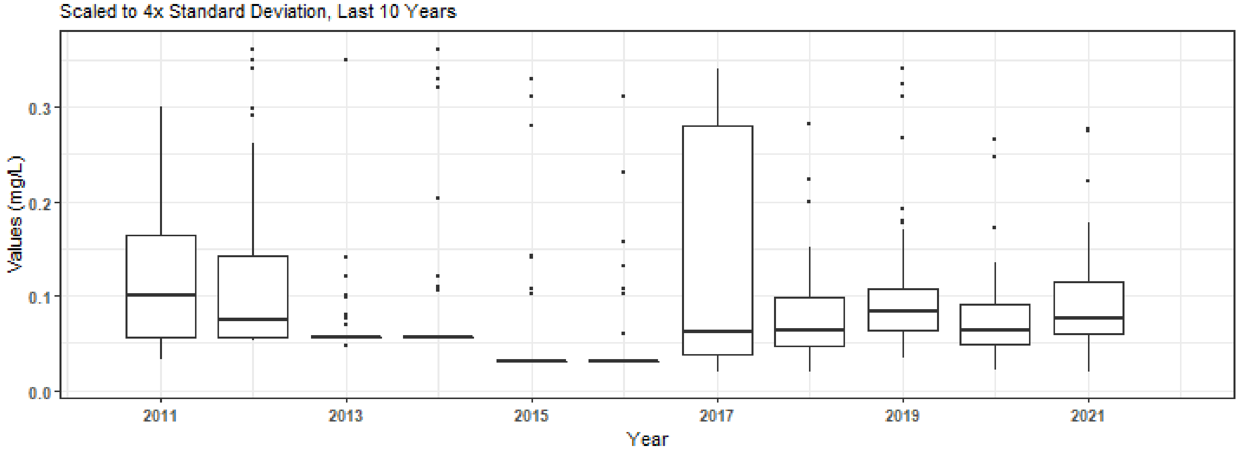
Autoscale



Scaled to 4x Standard Deviation

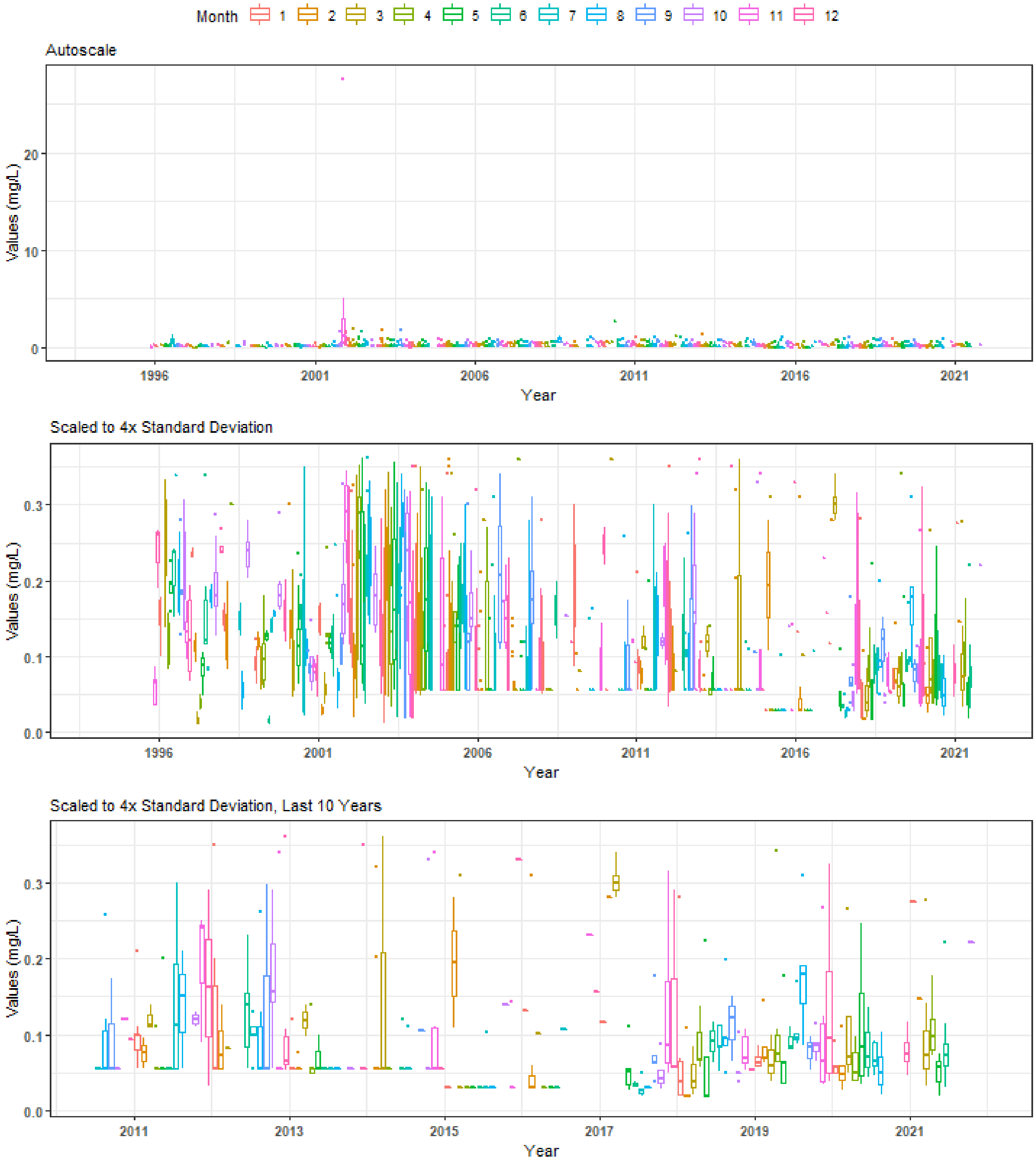


Scaled to 4x Standard Deviation, Last 10 Years



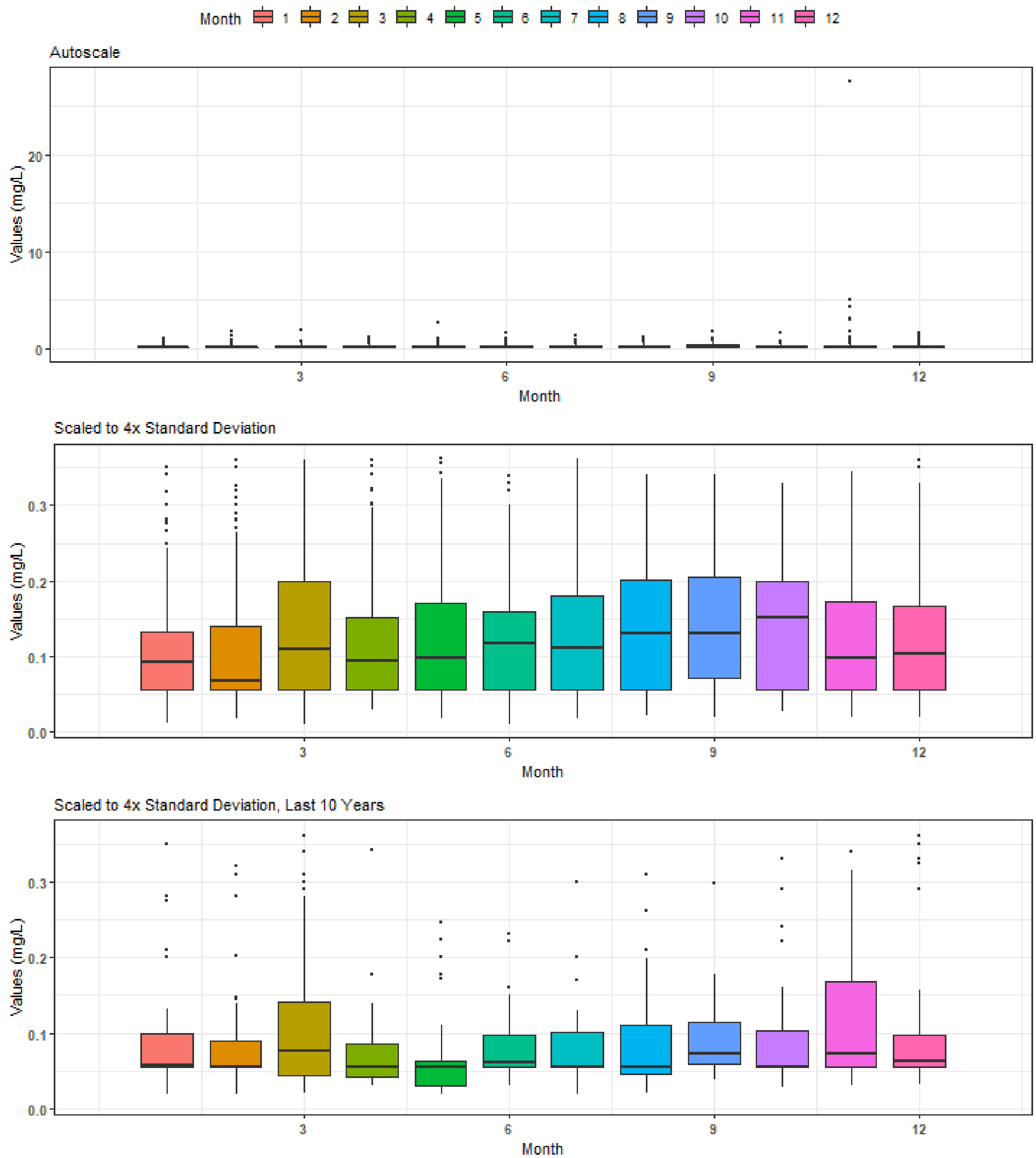
Summary Box Plots for Terra Ceia Aquatic Preserve

By Year & Month



Summary Box Plots for Terra Ceia Aquatic Preserve

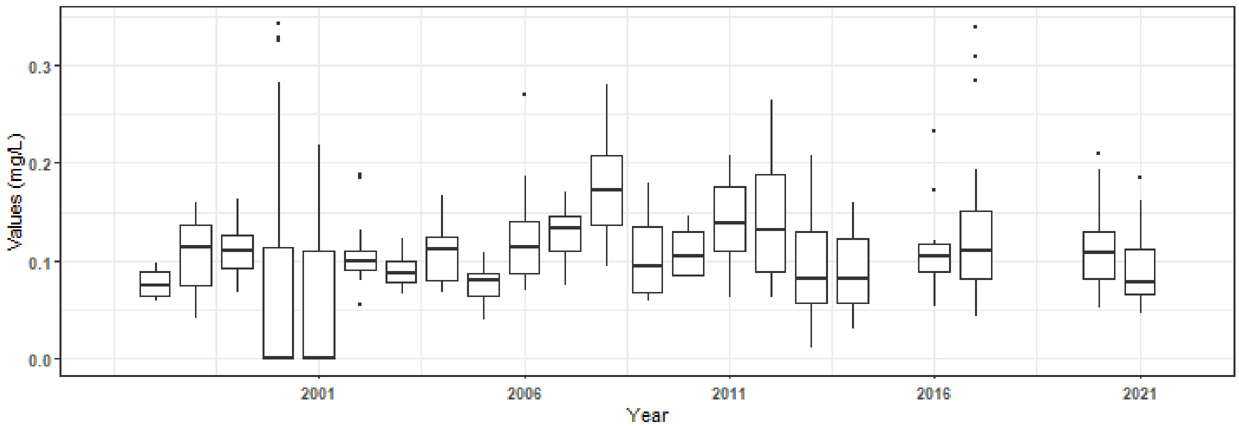
By Month



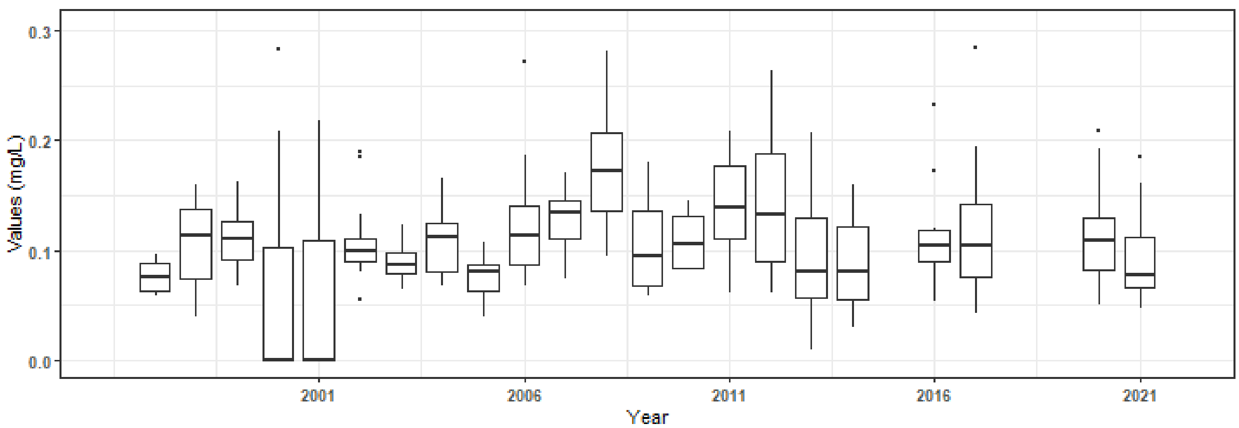
Summary Box Plots for Tomoka Marsh Aquatic Preserve

By Year

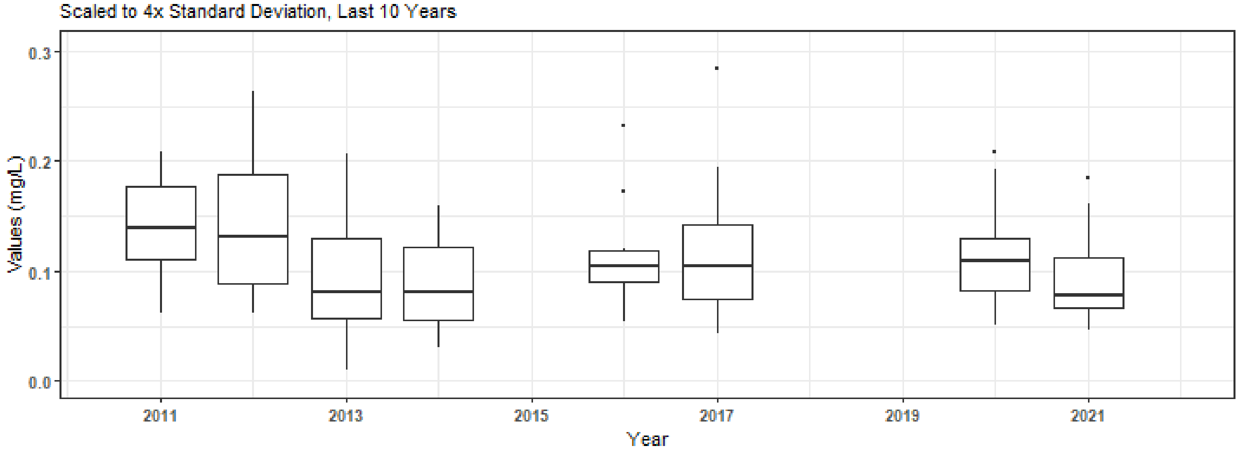
Autoscale

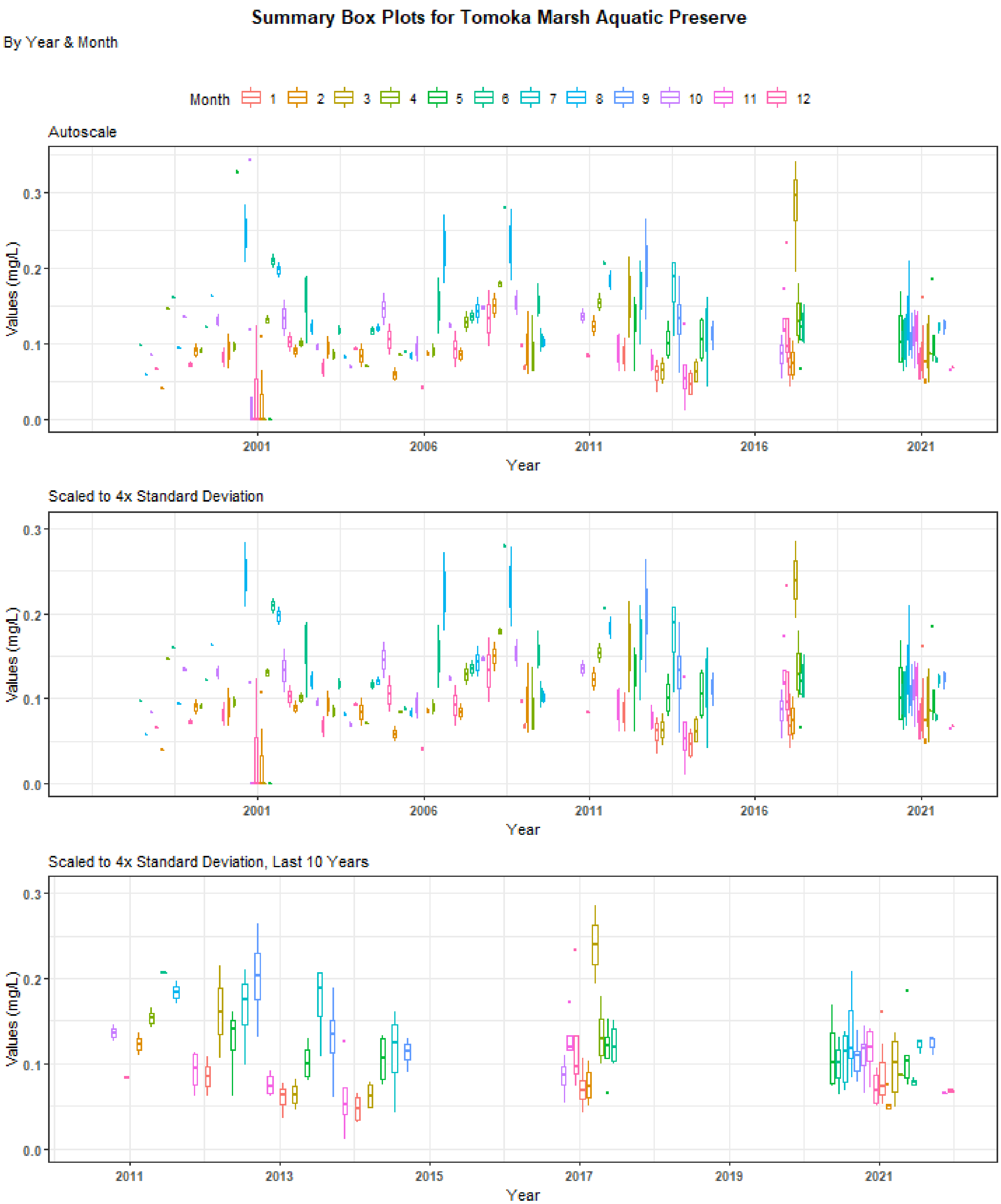


Scaled to 4x Standard Deviation



Scaled to 4x Standard Deviation, Last 10 Years

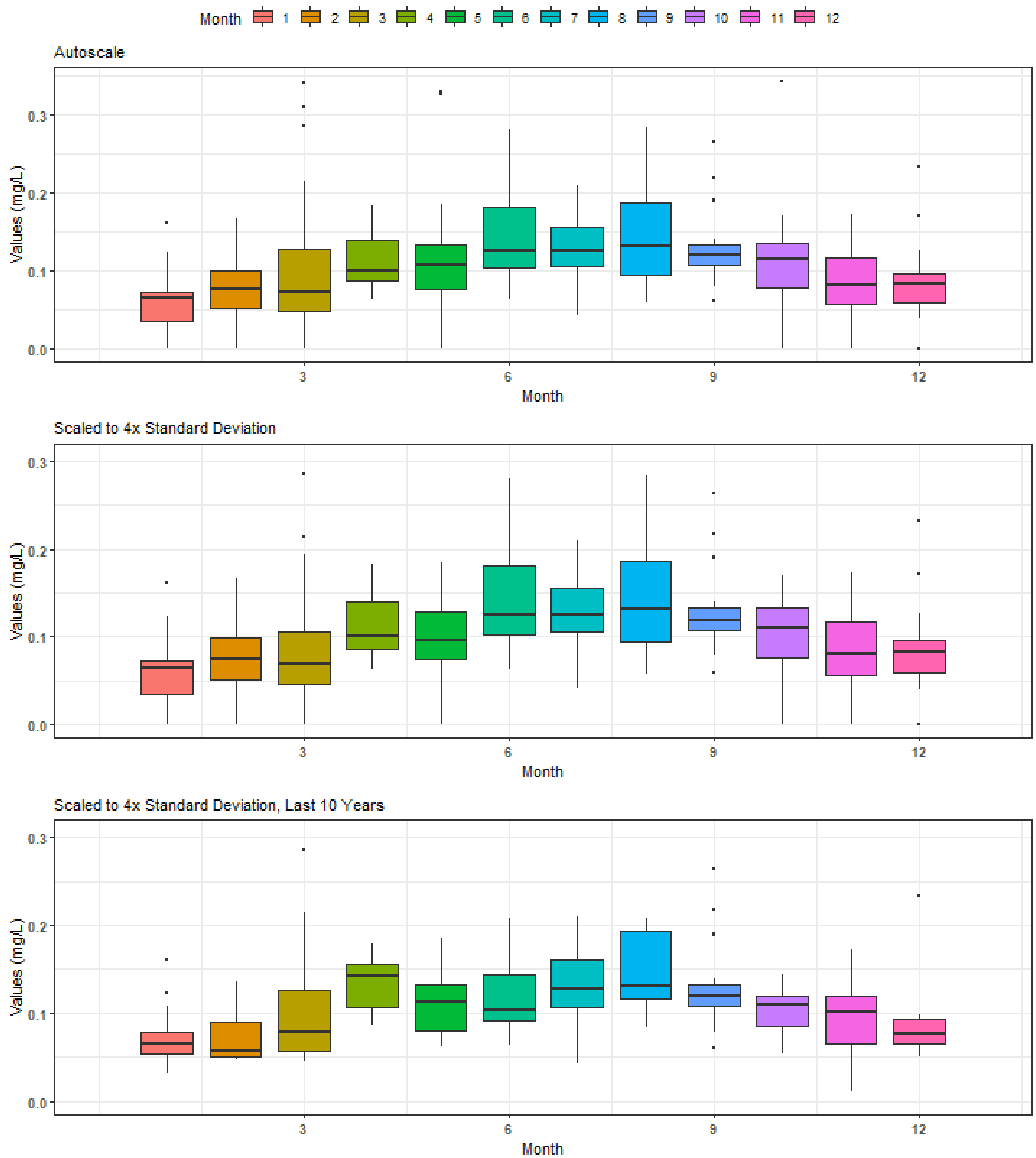




Summary Box Plots for Tomoka Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0.0
0.1
0.2
0.3
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0.0
0.1
0.2
0.3
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0.0
0.1
0.2
0.3
2011
2013
2015
2017
2019
2021
Year

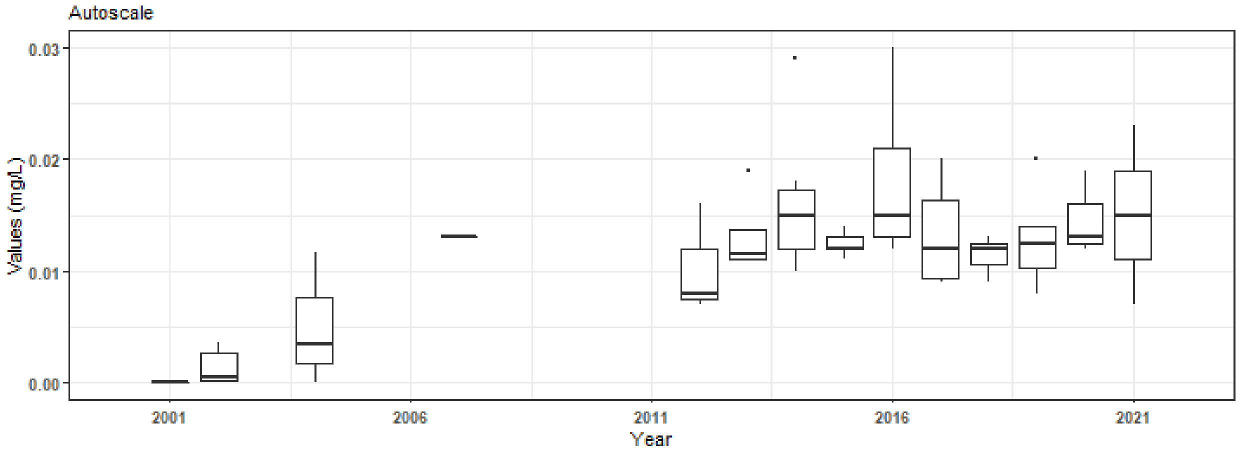
Summary Box Plots for Tomoka Marsh Aquatic Preserve

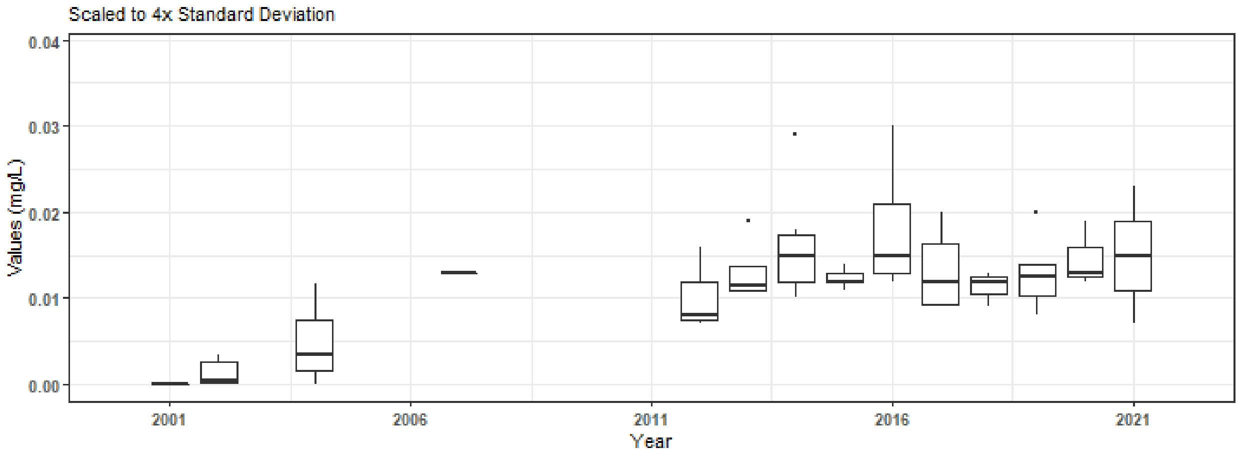
By Month

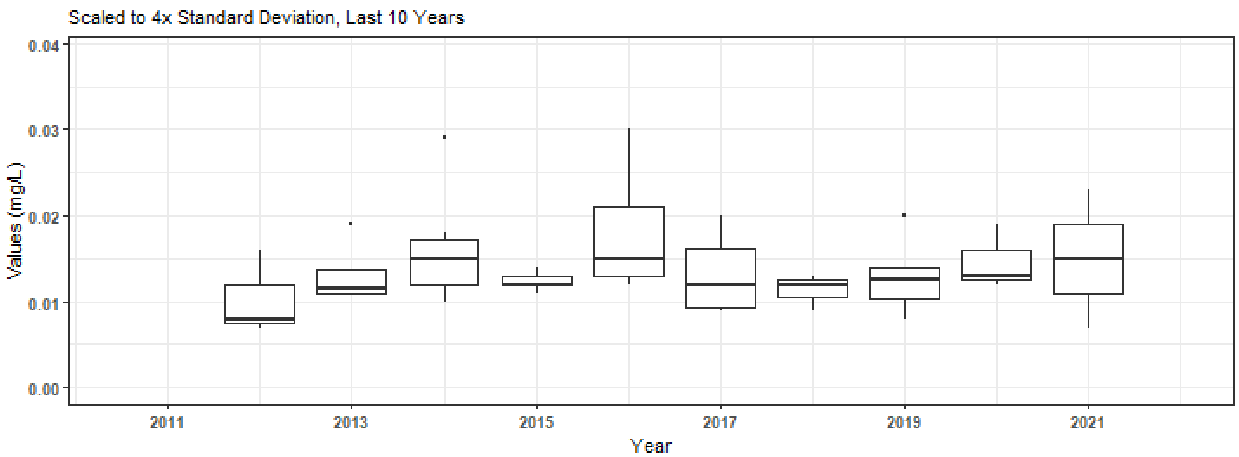


Summary Box Plots for Yellow River Marsh Aquatic Preserve

By Year

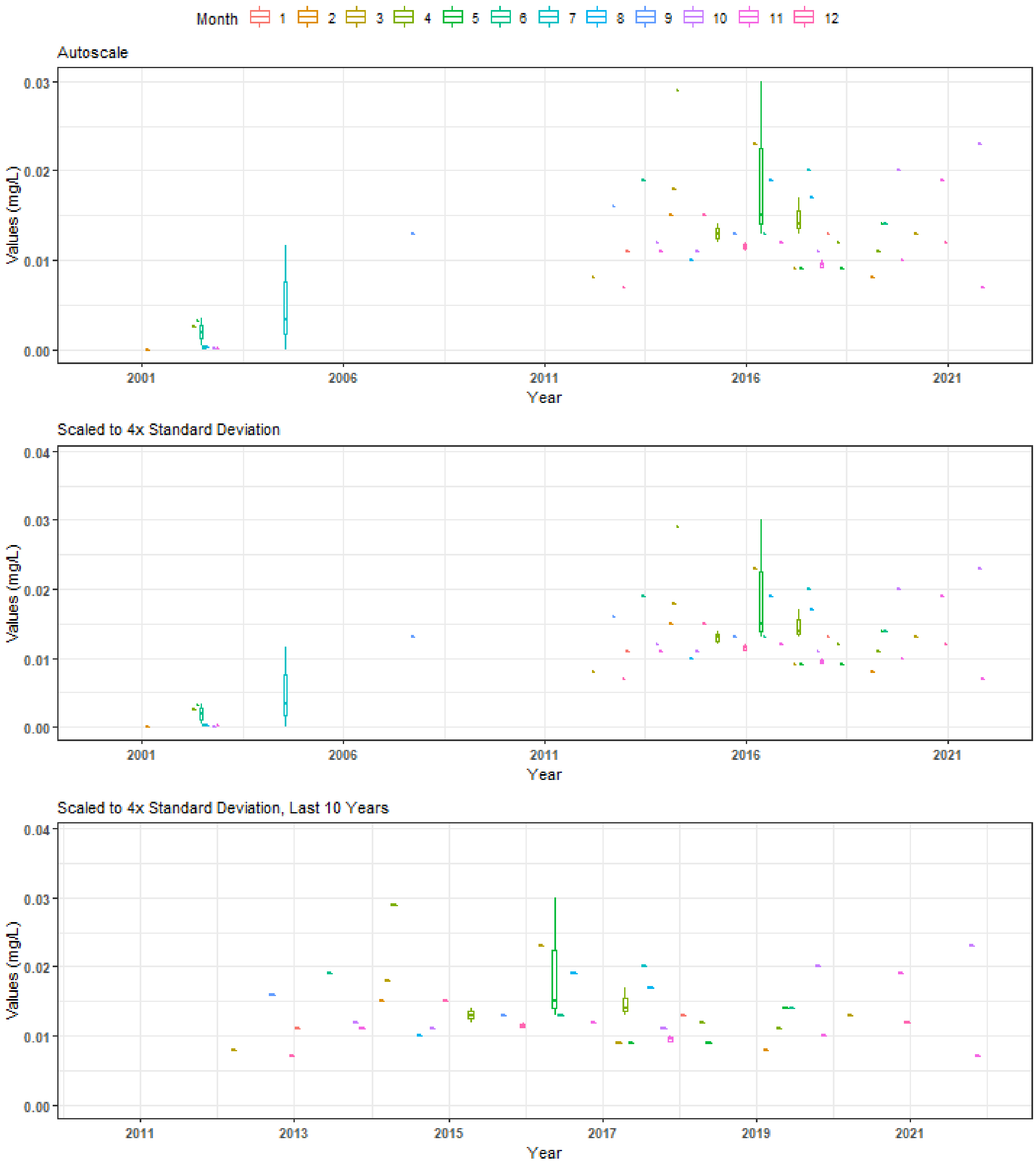






Summary Box Plots for Yellow River Marsh Aquatic Preserve

By Year & Month



Summary Box Plots for Yellow River Marsh Aquatic Preserve

By Month