SEACAR Discrete Water Quality Analysis: Field Bottom Dissolved Oxygen

Last compiled on 24 June, 2022

Contents

Important Notes

All scripts and outputs can be found on the SEACAR GitHub repository:

https://github.com/FloridaSEACAR/SEACAR_Panzik

Note: The top 2% of data is excluded when computing mean and standard deviations in plotting sections solely for the purpose of getting y-axis scales. The exclusion of the top 2% is not used in any statistics that are exported.

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(plyr)
library(dplyr)
library(lubridate)
library(ggplot2)
library(ggpubr)
library(scales)
library(EnvStats)
library(tidyr)
library(stringr)
library(kableExtra)

windowsFonts(`Segoe UI` = windowsFont('Segoe UI'))
options(scipen=999)
opts_chunk$set(warning=FALSE, message=FALSE, dpi=200)
```

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.

```
#MA_All <- fread(here::here("WQ_Discrete/data/ManagedArea.csv"), sep = ",", header = TRUE, stringsAsFac
                #na.strings = "")

#file_in <- "C:/Users/steph/Dropbox/SEACAR_Panzik/SEACAR_Panzik/WQ_Discrete/data/Combined_WQ_WC_NUT_Wat
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="")

activity <- activity
depth <- depth
parameter <- unique(data$ParameterName)
unit <- unique(data$ParameterUnits)
# activity <- unique(data$ActivityType)
# depth <- unique(data$RelativeDepth)
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[, `:=` (relyear = Year - min(Year), relyear_dd = DecDate - min(DecDate)), by = "ManagedAreaName"]
data <- data[ParameterName == parameter & str_detect(ActivityType, activity) & RelativeDepth == depth &
```

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.

```
# param_name <- "Water_Temperature"
# out_dir <- here::here("WQ_Discrete/output/by_parameter/")
# APP_Plots <- TRUE

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

data$Include <- as.logical(data$Include)
data$Include[grep("H", data$ValueQualifier[data$ProgramID==476])] <- TRUE
data <- data[!is.na(data$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),]
}

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

MA_Summ <- data %>%
   group_by(AreaID, ManagedAreaName) %>%
   dplyr::summarize(ParameterName=parameter,
             RelativeDepth=depth,
             ActivityType=activity,
             N_Data=length(ResultValue[Include==TRUE & !is.na(ResultValue)]),
             N_Years=length(unique(Year[Include==TRUE & !is.na(Year)])),
             EarliestYear=min(Year[Include==TRUE]),
             LatestYear=max(Year[Include==TRUE]),
             SufficientData=ifelse(N_Data>0 & N_Years>=10, TRUE, FALSE))

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

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

MA_Summ <- MA_Summ %>%
   select(AreaID, ManagedAreaName, ParameterName, RelativeDepth, ActivityType,
          SufficientData, everything())
MA_Summ <- as.data.frame(MA_Summ[order(MA_Summ$ManagedAreaName), ])


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$VQ_Plot[na.omit(data$ProgramID!=476)] <- gsub("[^U]+", "",
                                        data$VQ_Plot[na.omit(data$ProgramID!=476)])
   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: 164352, Number Passed Filter: 164352
## Program 476 H Codes: 0 (0%)
## I Codes: 0 (0%)
## Q Codes: 0 (0%)
## U Codes: 0 (0%)
```

```
data_summ <- data %>%
   group_by(AreaID, ManagedAreaName) %>%
   dplyr::summarize(ParameterName=parameter,
             RelativeDepth=depth,
             ActivityType=activity,
             N_Total=length(ResultValue),
```

```
            N_AnalysisUse=length(ResultValue[SufficientData==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))

data_summ <- as.data.table(data_summ[order(data_summ$ManagedAreaName), ])
fwrite(data_summ, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                         "_DataSummary.csv"), sep=",")
rm(data_summ)
MA_Include <- MA_Summ$ManagedAreaName[MA_Summ$SufficientData==TRUE &
                                      MA_Summ$N_Data<2000000]
n <- length(MA_Include)
MA_Exclude <- MA_Summ[MA_Summ$N_Years<10 & MA_Summ$N_Years>0,]
MA_Exclude <- MA_Exclude[,c("ManagedAreaName", "N_Years")]
z <- nrow(MA_Exclude)
setDT(data)
```

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 `SufficientData` 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_YM_Stats <- data[data$Use_In_Analysis==TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, Year, Month) %>%
   dplyr::summarize(ParameterName=parameter,
            RelativeDepth=depth,
            ActivityType=activity,
            N_Data=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="|")
rm(MA_YM_Stats)

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

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

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 `SufficientData` 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) %>%
   dplyr::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="|")
rm(Mon_Stats)
```

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 `SufficientData` 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, seasondata = MA_M_Stats[MA_M_Stats$ManagedAreaName == MA_Include
  setDT(data)
  tau <- NULL
   tryCatch({ken <- kendallSeasonalTrendTest(
      y = data$ResultValue,
      season = data$Month,
      year = data$relyear,
      independent.obs = independent)

   tau <- ken$estimate[1]
   z <- ken$statistic[2]
   p_z <- ken$p.value[2]
   chi_sq <- ken$statistic[1]
   p_chi_sq <- ken$p.value[1]
   slope <- ken$estimate[2]
   intercept <- ken$estimate[3]
   trend <- trend_calculator(slope, stats.median, p_z)

   seasonresults <- as.data.table(ken$seasonal.estimates)
   rm(ken)
   }, warning = function(w) {
      print(w)
   }, error = function(e) {
      print(e)
   }, finally = {
      if (!exists("tau")) {
         tau <- NA
      }
     if (!exists("z")) {
         z <- NA
      }
      if (!exists("p_z")) {
         p_z <- NA
      }
      if (!exists("chi_sq")) {
         chi_sq <- NA
      }
```

```
    if (!exists("p_chi_sq")) {
       p_chi_sq <- NA
    }
    if (!exists("slope")) {
       slope <- NA
    }
    if (!exists("intercept")) {
       intercept <- NA
    }
    if (!exists("trend")) {
       trend <- NA
    }
})
KT <-data.table(AreaID = unique(data$AreaID),
                ManagedAreaName = unique(data$ManagedAreaName),
                season = "All",
                stats.median = stats.median,
                independent = independent,
                tau = tau,
                z = z,
                p_z = p_z,
                chi_sq = chi_sq,
                p_chi_sq = p_chi_sq,
                slope = slope,
                intercept = intercept,
                trend = trend)

seasonresults[, `:=` (AreaID = unique(data$AreaID),
                      ManagedAreaName = unique(data$ManagedAreaName),
                      season = unique(data$Month),
                      stats.median = as.numeric(NA),
                      independent = independent,
                      z = as.numeric(NA),
                      p_z = as.numeric(NA),
                      chi_sq = as.numeric(NA),
                      p_chi_sq = as.numeric(NA),
                      trend = as.integer(NA))]

for(s in as.integer(unique(seasonresults$season))){
  seasondat_s <- data[Month == s, ]

  if(nrow(seasondat_s) < 3 | length(unique(seasondat_s$Year)) < 3 | is.na(seasonresults[season == s,
    next

  } else{
    if(!is.na(unique(seasondat_s$Month))){
    trend_s <- trend_calculator(seasonresults[season == s, slope], seasondata[Month == s, Median], p
    ken_s <- kendallTrendTest(ResultValue ~ relyear, data = seasondat_s)
    seasonresults[season == s, `:=` (stats.median = unique(seasondata[Month == s, Median]),
                                     z = ken_s$statistic,
                                     p_z = ken_s$p.value,
                                     chi_sq = NA,
                                     p_chi_sq = NA,
```

```
                                        trend = trend_s)]
    } else{
      next
    }
   }
  }

  seasonresults[, season := as.character(season)]

  KT <- rbind(KT, seasonresults)
  KT[, season := factor(season, levels = c("All", seq(1:12)), ordered = TRUE)]

  return(KT)
}
runStats <- function(data, MA_M_Stats) {
  data$Index <- as.Date(data$SampleDate) # , "%Y-%m-%d")
  data$ResultValue <- as.numeric(data$ResultValue)
  # Calculate basic stats
  stats.median <- median(data$ResultValue, na.rm = TRUE)
  stats.minYear <- min(data$relyear, na.rm = TRUE)
  stats.maxYear <- max(data$relyear, na.rm = TRUE)
  # Calculate Kendall Tau and Slope stats, then update appropriate columns and table
  seasondata <- MA_M_Stats[MA_M_Stats$ManagedAreaName == MA_Include[i]]
  KT <- tauSeasonal(data, TRUE, stats.median,
                    stats.minYear, stats.maxYear, seasondata)
  # if (is.null(KT[9])) {
  if (is.na(KT[season == "All", trend])) {
    KT <- tauSeasonal(data, FALSE, stats.median,
                      stats.minYear, stats.maxYear, seasondata)
  }
  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$ManagedAreaName
c_names <- c("AreaID", "ManagedAreaName", "Season", "Median", "Independent",
             "tau", "z", "p_z", "chi_sq", "p_chi_sq", "SennSlope", "SennIntercept", "Trend")
if(n==0){
   KT.Stats <- data.frame(matrix(ncol=length(c_names),
                                 nrow=length(MA_Summ$ManagedAreaName)))
   colnames(KT.Stats) <- c_names
   # KT.Stats[, c("AreaID", "ManagedAreaName")] <-
      # MA_Summ[, c("AreaID", "ManagedAreaName")]
} else{
   for (i in 1:n) {
      x <- nrow(data[data$Use_In_Analysis == TRUE &
                        data$ManagedAreaName == MA_Include[i], ])
      if (x>0) {
         KT.Stats <- runStats(data[data$Use_In_Analysis == TRUE &
                                      data$ManagedAreaName ==
                                      MA_Include[i], ], MA_M_Stats)
      }
   }
   KT.Stats <- as.data.frame(KT.Stats)
   # c_names <- c("AreaID", "ManagedAreaName", "Season", "Median", "Independent",
   #              "tau", "z", "p_z", "chi_sq", "p_chi_sq", "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$tau <- round(as.numeric(KT.Stats$tau), digits=4)
   KT.Stats$z <- round(as.numeric(KT.Stats$z), digits=4)
   KT.Stats$p_z <- round(as.numeric(KT.Stats$p_z), digits=4)
   KT.Stats$chi_sq <- round(as.numeric(KT.Stats$chi_sq), digits=4)
   KT.Stats$p_chi_sq <- round(as.numeric(KT.Stats$p_chi_sq), 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)
}

KT.Stats <-  merge.data.frame(MA_Summ, KT.Stats,
                              by=c("AreaID", "ManagedAreaName"), all=TRUE)

KT.Stats <- as.data.table(KT.Stats[order(KT.Stats$ManagedAreaName, KT.Stats$Season), ])
KT.Stats2 <- copy(KT.Stats)
KT.Stats[, `:=` (RelativeDepth = depth, Units = unit)]
KT.Stats_all <- rbind(KT.Stats_all, KT.Stats)
```

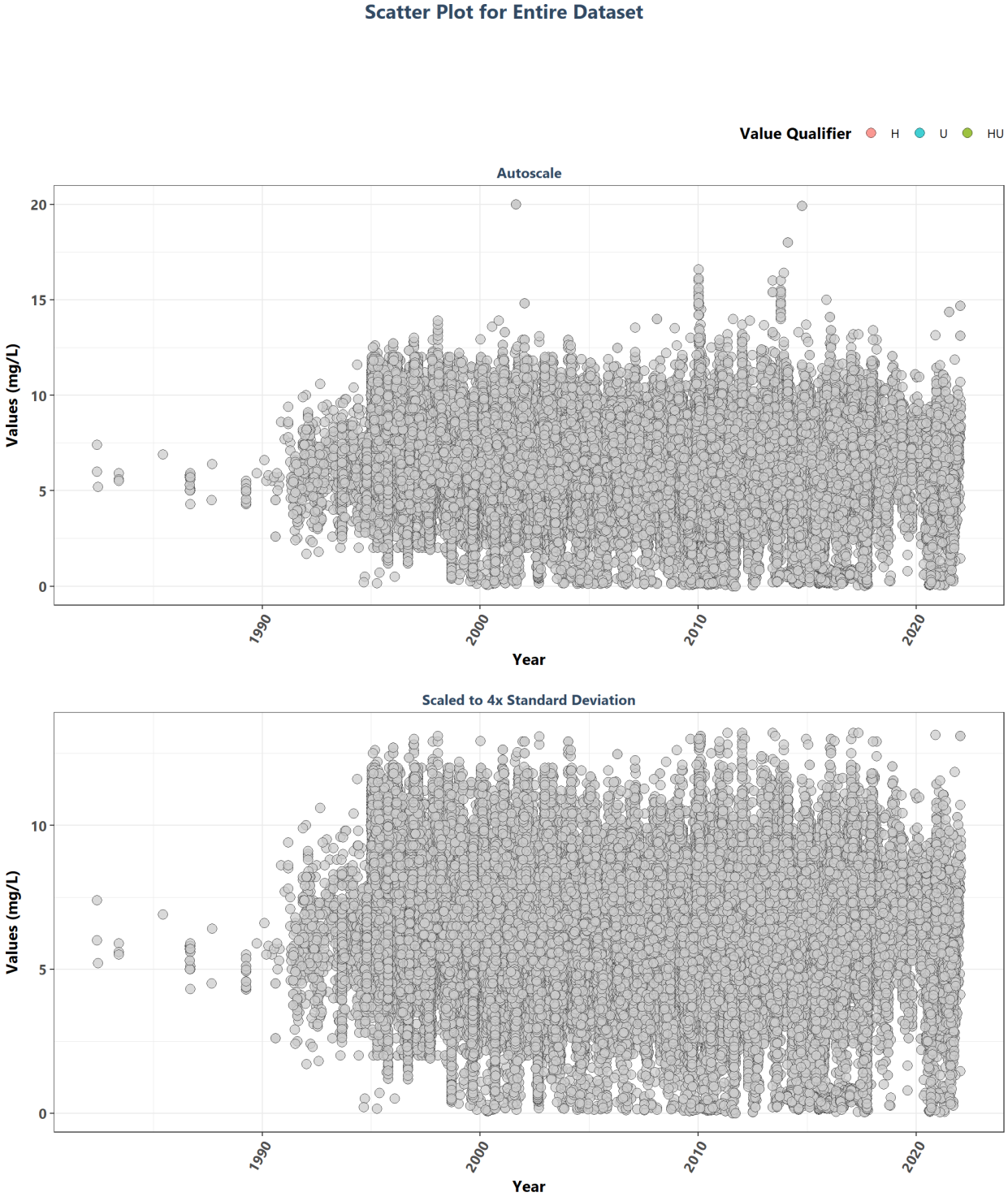
```
KT.Stats2$MonitoringID <- NULL
fwrite(KT.Stats2, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                         "_KendallTau_Stats.txt"), sep="|")
rm(KT.Stats2)
data <- data[!is.na(data$ResultValue),]
```

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.

```
plot_theme <- theme_bw() +
   theme(text=element_text(family="Segoe UI"),
         title=element_text(face="bold"),
         plot.title=element_text(hjust=0.5, size=14, color="#314963"),
         plot.subtitle=element_text(hjust=0.5, size=10, color="#314963"),
         axis.title.x = element_text(margin = margin(t = 5, r = 0,
                                                     b = 10, l = 0)),
         axis.title.y = element_text(margin = margin(t = 0, r = 10,
                                                     b = 0, l = 0)),
         axis.text=element_text(size=10),
         axis.text.x=element_text(face="bold", angle = 60, hjust = 1),
         axis.text.y=element_text(face="bold"))

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, fill=VQ_Plot)) +
   geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")"),
        fill="Value Qualifier") +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal",
         legend.justification="right") +
   scale_x_date(labels=date_format("%Y")) +
   {if(inc_H==TRUE){
      scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                 "HU"="#7CAE00"), na.value="#cccccc")
   } else if(param_name=="Secchi_Depth"){
      scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
```

```
                                    "SU"="#7CAE00"), na.value="#cccccc")
   } else {
      scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
   }}


p2 <- ggplot(data=data[data$Include==TRUE,],
             aes(x=SampleDate, y=ResultValue, fill=VQ_Plot)) +
   geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
   ylim(min_RV, y_scale) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
   plot_theme +
   theme(legend.position="none") +
   scale_x_date(labels=date_format("%Y")) +
   {if(inc_H==TRUE){
      scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                    "HU"="#7CAE00"), na.value="#cccccc")
   } else if(param_name=="Secchi_Depth"){
      scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                    "SU"="#7CAE00"), na.value="#cccccc")
   } else {
      scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
   }}

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") +
   plot_theme + theme(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))
```



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 `SufficientData` 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(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit, ")")) +
   plot_theme

p2 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation", x="Year",
        y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   plot_theme

p3 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=as.integer(Year), y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(max(data$Year) - 10.5, max(data$Year)+1),
                      breaks=seq(max(data$Year) - 10, max(data$Year), 2)) +
   plot_theme

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

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

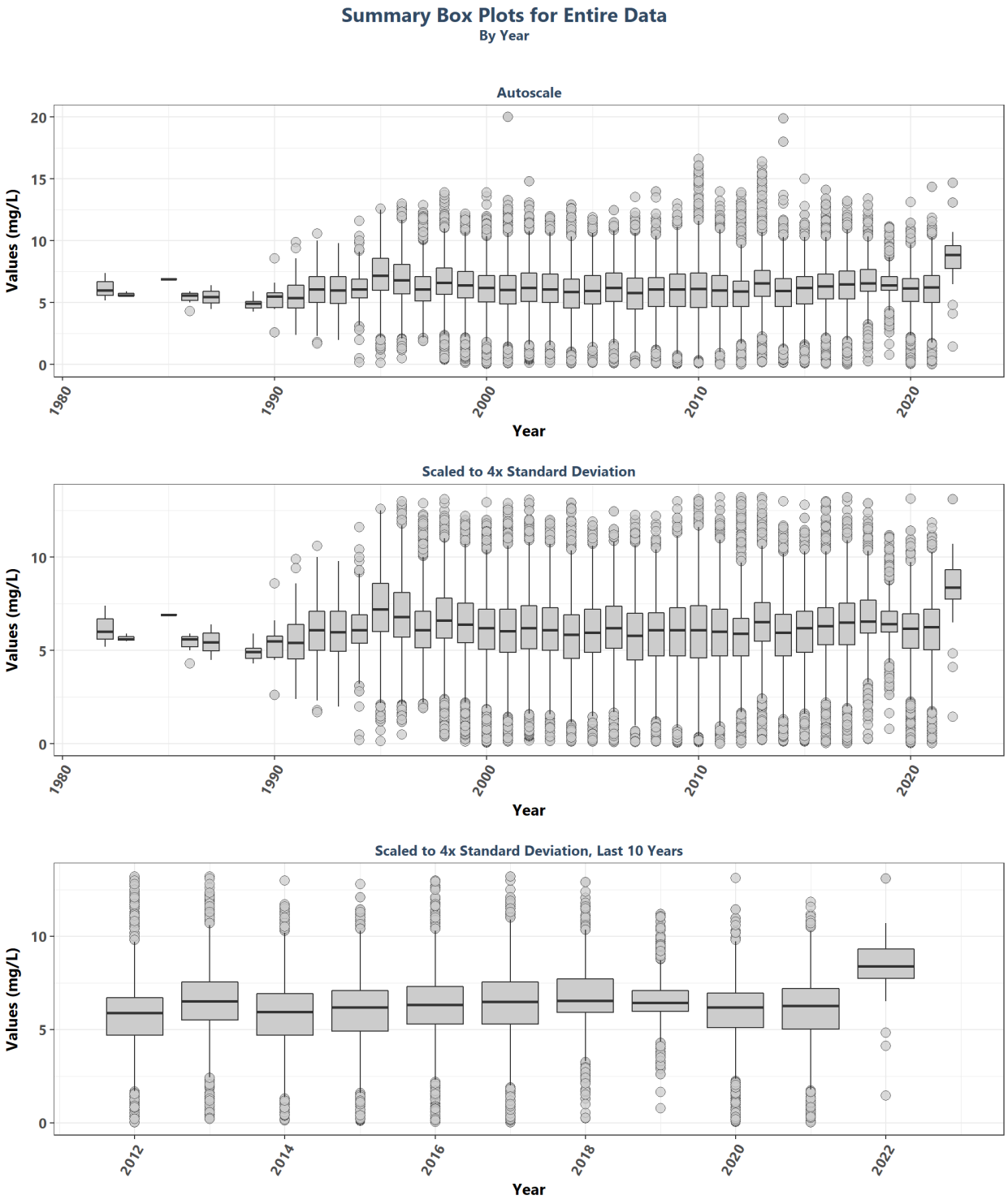
```
                      subtitle="By Year") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

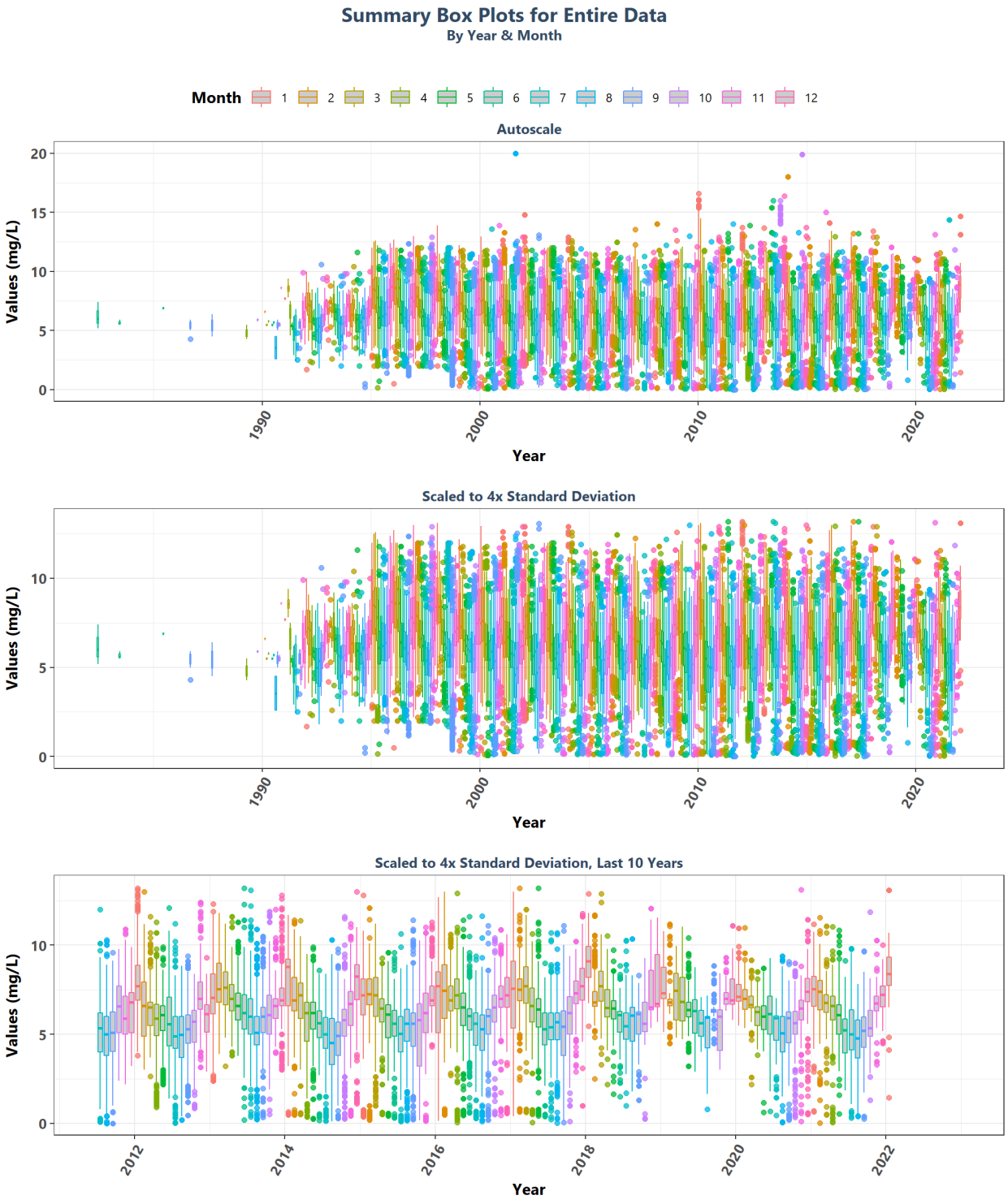
This set of box plots are grouped by year and month with the color being related to the month.

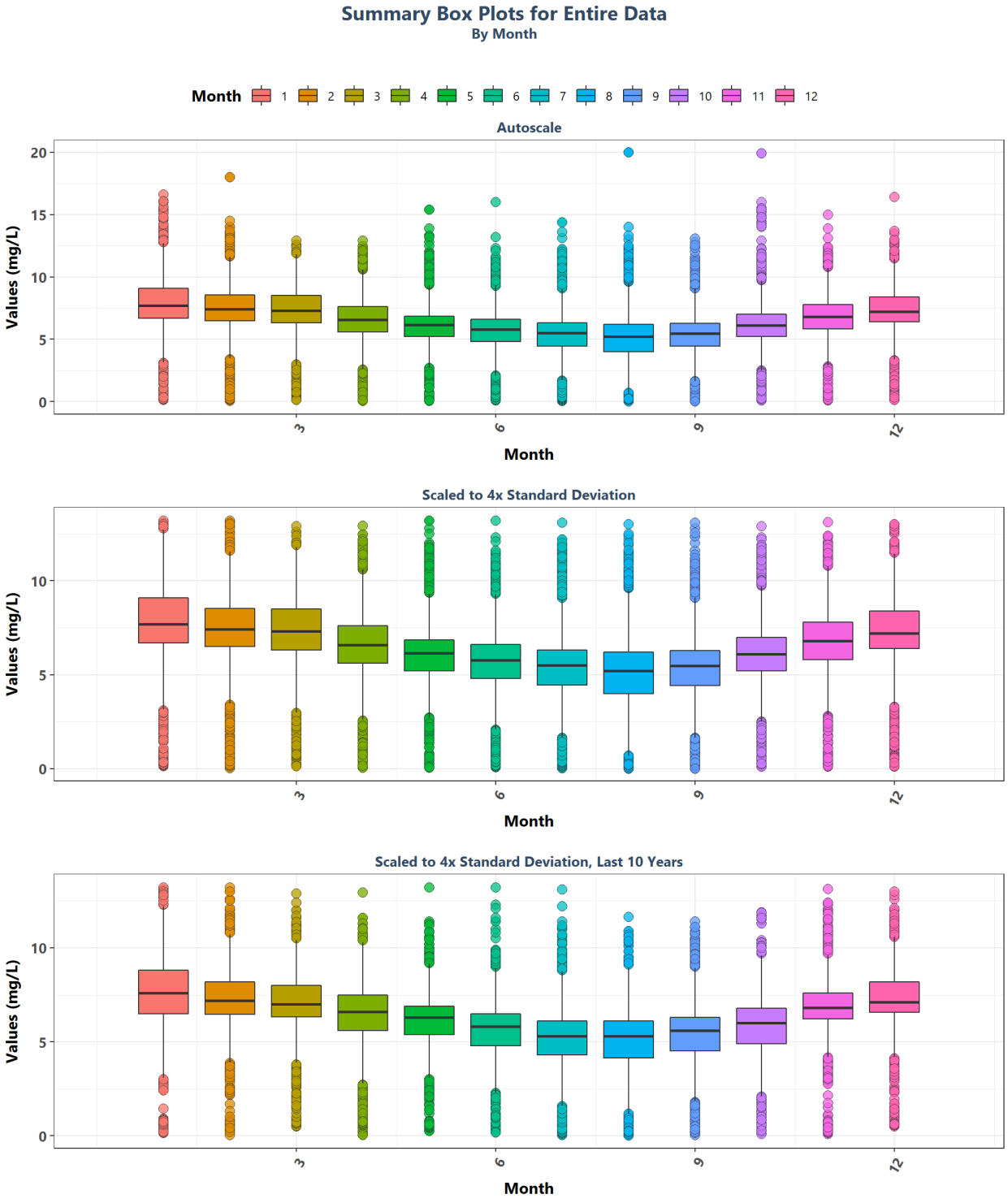
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit, ")"), color="Month") +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
   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(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   plot_theme +
   theme(legend.position="none", axis.text.x=element_text(face="bold"),
         axis.text.y=element_text(face="bold"))

p3 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(max(data$Year) - 10.5, max(data$Year)+1),
                      breaks=seq(max(data$Year) - 10, max(data$Year), 2)) +
   plot_theme +
   theme(legend.position="none")
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") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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.

```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Month",
        y=paste0("Values (", unit, ")"), fill="Month") +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
   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(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   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)) +
   plot_theme +
   theme(legend.position="none")

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(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   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)) +
   plot_theme +
   theme(legend.position="none")
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") + plot_theme +
   theme(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)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

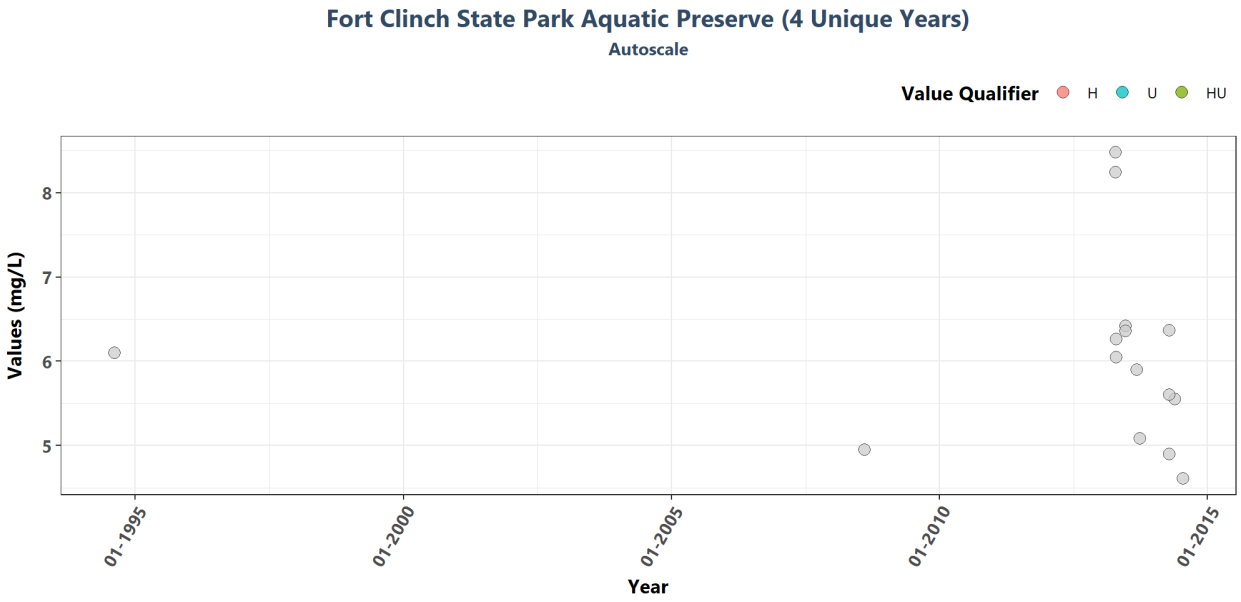

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

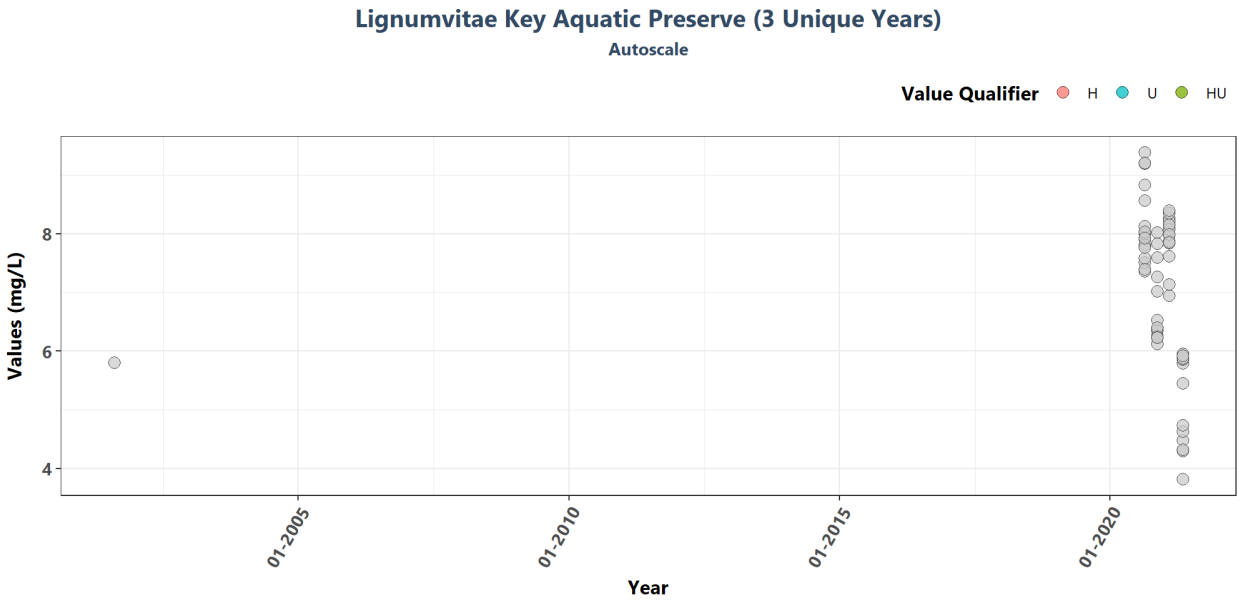


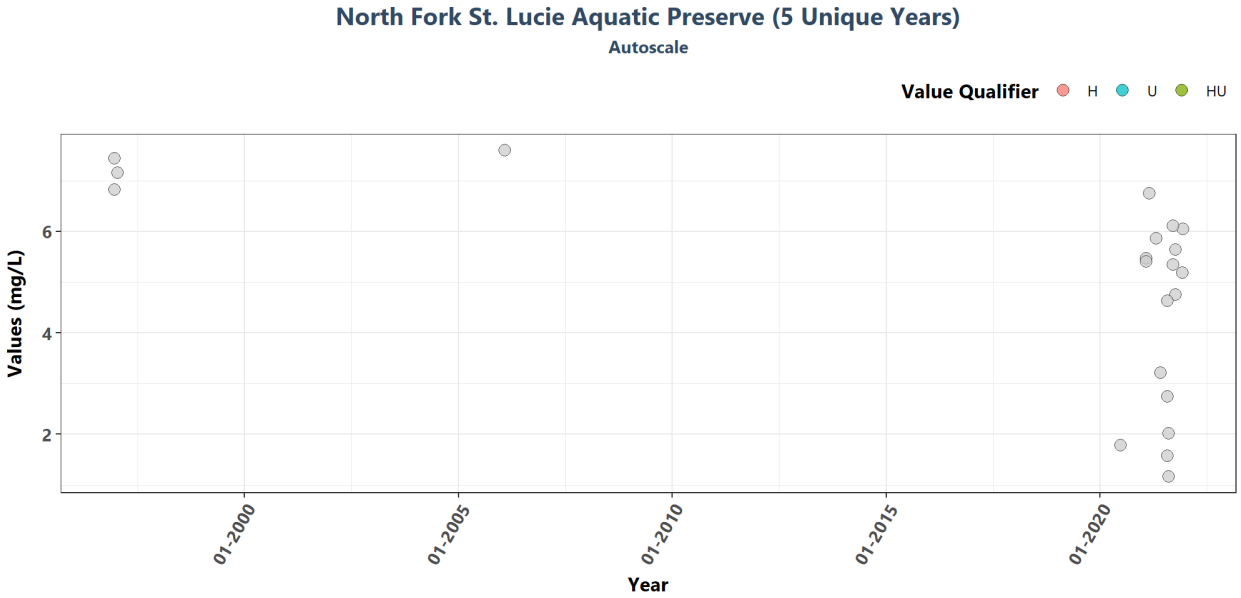
Appendix III: Excluded Managed Areas

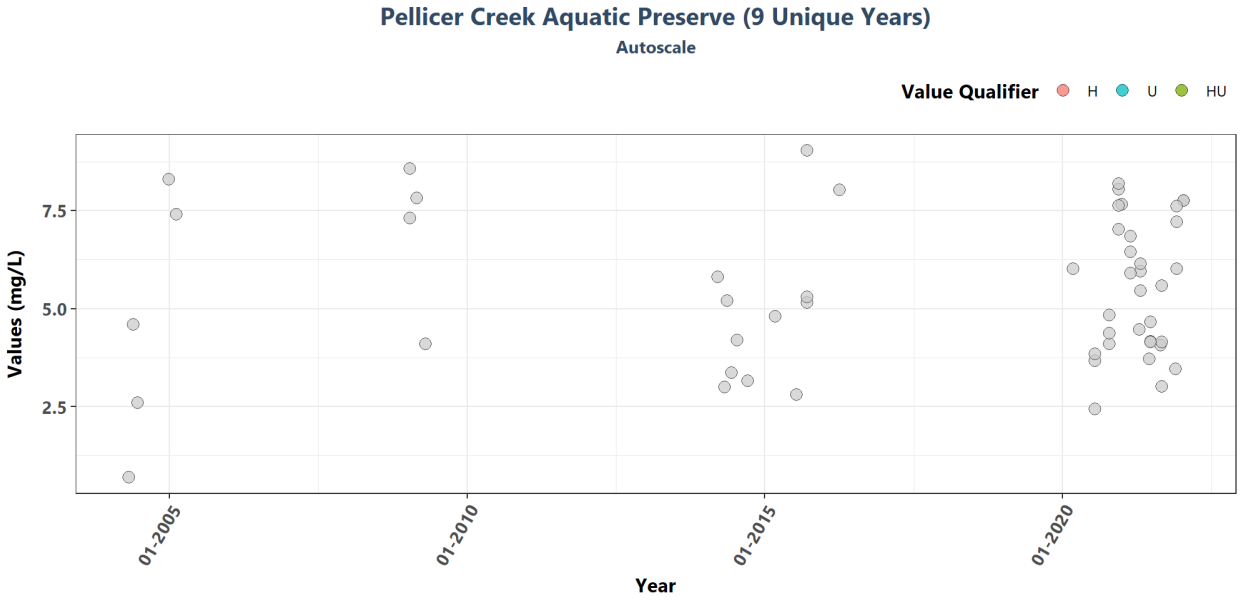
Scatter plots of data values are created for managed areas that have fewer than 10 separate years of data entries. Data points are colored based on specific value qualifiers of interest.

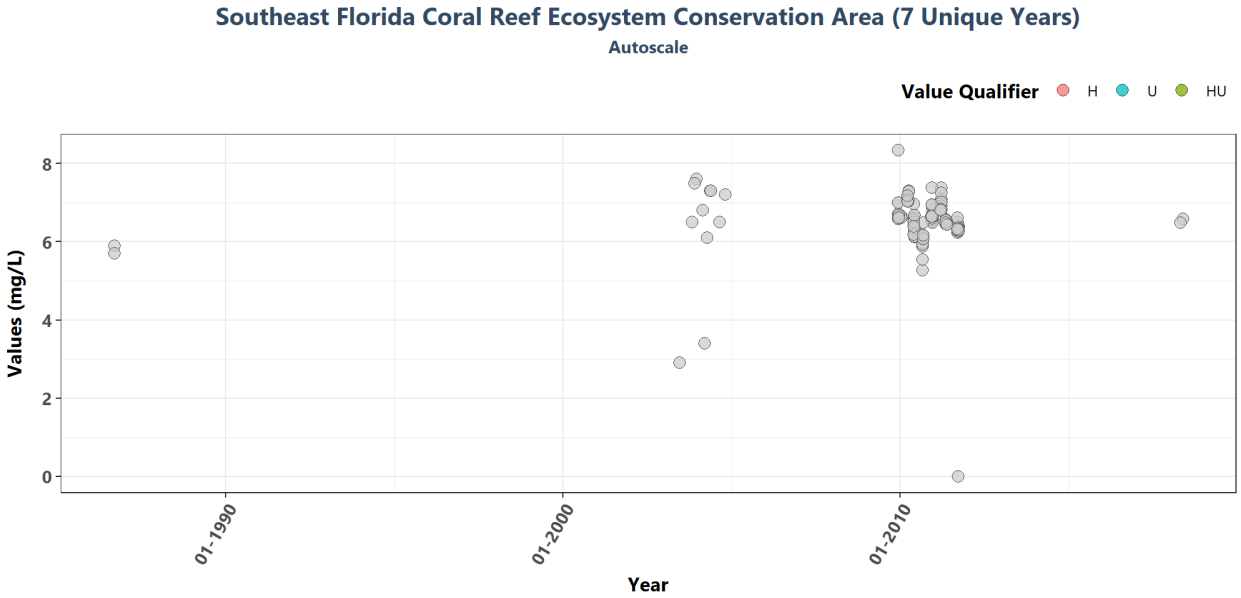
```
if(z==0){
   print("There are no managed areas that qualify.")
} else {
   for(i in 1:z){
      p1<-ggplot(data=data[data$ManagedAreaName==MA_Exclude$ManagedAreaName[i]&
                           data$Include==TRUE, ],
                 aes(x=SampleDate, y=ResultValue, fill=VQ_Plot)) +
         geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
         labs(title=paste0(MA_Exclude$ManagedAreaName[i], " (",
                           MA_Exclude$N_Years[i], " Unique Years)"),
              subtitle="Autoscale", x="Year",
              y=paste0("Values (", unit, ")"), fill="Value Qualifier") +
         plot_theme +
         theme(legend.position="top", legend.box="horizontal",
               legend.justification="right") +
         scale_x_date(labels=date_format("%m-%Y")) +
         {if(inc_H==TRUE){
            scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                       "HU"="#7CAE00"), na.value="#cccccc")
         } else if(param_name=="Secchi_Depth"){
            scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                       "SU"="#7CAE00"), na.value="#cccccc")
         } else {
            scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
         }}
      print(p1)
   }
}
```




Lignumvitae Key Aquatic Preserve (3 Unique Years)
Autoscale
Value Qualifier
H
U
HU
Values (mg/L)
8
6
4
01-2005
01-2010
01-2015
01-2020
Year


North Fork St. Lucie Aquatic Preserve (5 Unique Years)
Autoscale
Value Qualifier
H
U
HU
Values (mg/L)
6
4
2
01-2000
01-2005
01-2010
01-2015
01-2020
Year





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 `SufficientData` 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

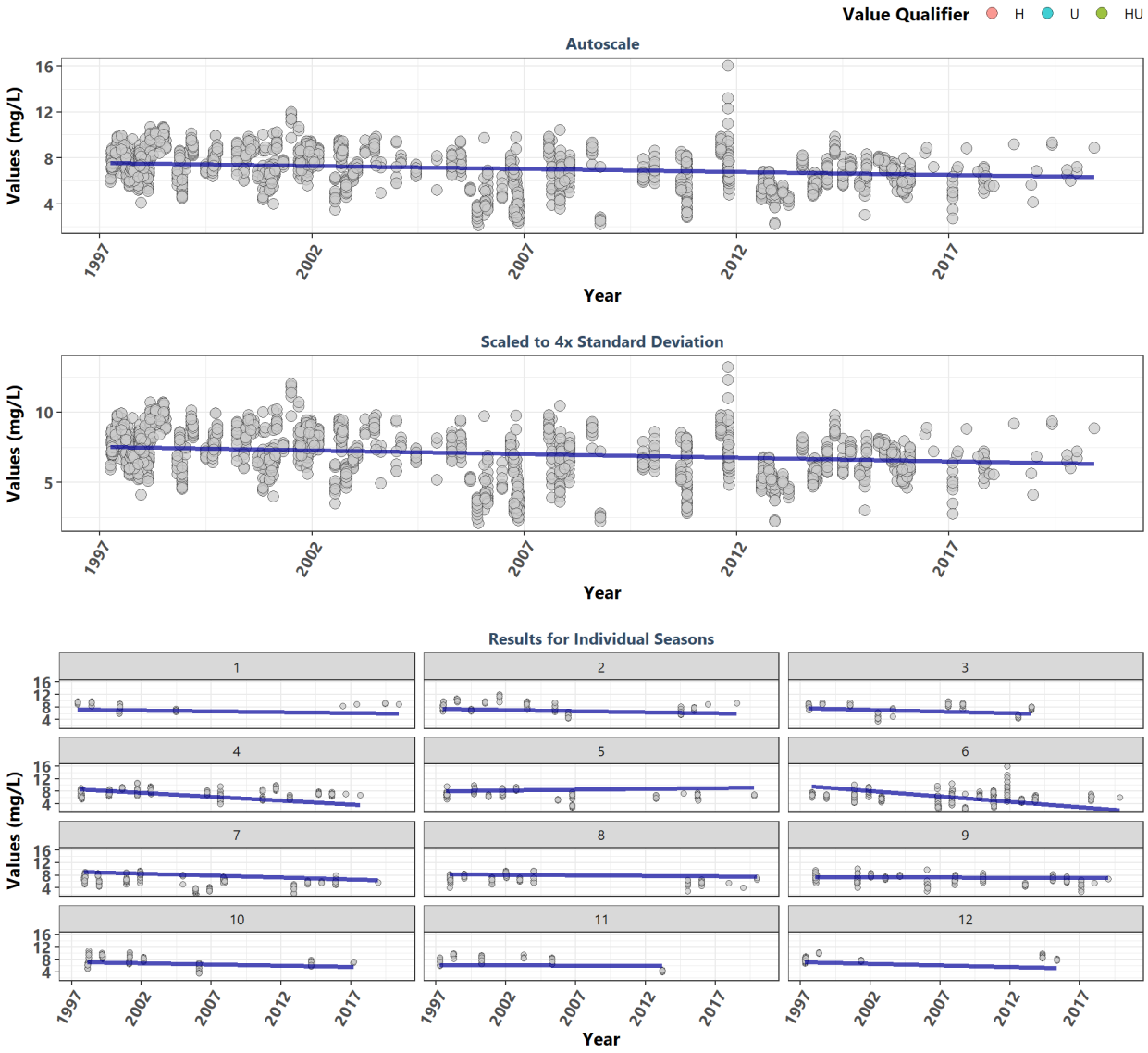
5. Set the plot type as a point plot with the size of the points
6. Add the linear trend
7. Create the title, x-axis, y-axis, and color fill labels
8. Set the y and x limits
9. Make the axis labels bold
10. Plot the arrangement as a set of panels

```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$SufficientData==TRUE &
                           data$ManagedAreaName==MA_Include[i],]
      plot_data$Season <- factor(plot_data$Month, levels = c("All", seq(1, 12)), ordered = TRUE)
      year_lower <- min(plot_data$relyear)
      year_upper <- max(plot_data$relyear)
      min_RV <- min(plot_data$ResultValue)
      mn_RV <- mean(plot_data$ResultValue[plot_data$ResultValue <
                                             quantile(data$ResultValue, 0.98)])
      sd_RV <- sd(plot_data$ResultValue[plot_data$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_Include[i]]
      s_slope <- KT.Stats$SennSlope[KT.Stats$ManagedAreaName==MA_Include[i]]
      s_int <- KT.Stats$SennIntercept[KT.Stats$ManagedAreaName==MA_Include[i]]
      trend <- KT.Stats$Trend[KT.Stats$ManagedAreaName==MA_Include[i]]
      z <- KT.Stats$z[KT.Stats$ManagedAreaName==MA_Include[i]]
      p_z <- KT.Stats$p_z[KT.Stats$ManagedAreaName==MA_Include[i]]
      chi_sq <- KT.Stats$chi_sq[KT.Stats$ManagedAreaName==MA_Include[i]]
      p_chi_sq <- KT.Stats$p_chi_sq[KT.Stats$ManagedAreaName==MA_Include[i]]

      # model <- lm(ResultValue ~ relyear_dd,
      #             data=plot_data)
      # m_int <- coef(model)[[1]]
      # m_slope <- coef(model)[[2]]
      # rm(model)

      xbrks <- seq(round_any(min(plot_data$relyear_dd), 5, floor), round_any(max(plot_data$relyear_dd),
                   by = (round_any(max(plot_data$relyear_dd), 5, ceiling) - round_any(min(plot_data$rely

      xlabs <- seq(max(plot_data$Year) - round_any(max(plot_data$relyear_dd), 5, ceiling),
                   max(plot_data$Year),
                   by = (max(plot_data$Year) - (max(plot_data$Year) - round_any(max(plot_data$relyear_d

      KT.Stats[, season := Season]
      KT.Stats[ManagedAreaName==MA_Include[i] & season != "All", `:=` (N_Data = nrow(plot_data[Season ==
      KT.Stats[ManagedAreaName==MA_Include[i] & season == "All", `:=` (relyear_dd_lower = min(plot_data
      KT.Stats[, season := NULL]

      # plot_data[is.na(VQ_Plot), VQ_Plot := "None"]
      p1 <- ggplot(data=plot_data,
                   aes(x=relyear_dd, y=ResultValue, fill = VQ_Plot)) +
```

```
    geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
    # geom_abline(aes(slope=s_slope, intercept=s_int),
    #             color="#000099", size=1.2, alpha=0.7) +
    geom_segment(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", ], aes(x = relyear_dd,
                                                                                          y = relyear_dd,
                                                                                          xend = relyear_dd,
                                                                                          yend = relyear_dd),
                 color="#000099", size=1.2, alpha=0.7, inherit.aes = FALSE) +
    labs(subtitle="Autoscale",
         x="Year", y=paste0("Values (", unit, ")"),
         fill="Value Qualifier") +
    plot_theme +
    theme(legend.position="top", legend.box="horizontal",
          legend.justification="right") +
    {if(inc_H==TRUE){
       scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                  "HU"="#7CAE00"), na.value="#cccccc")
    } else if(param_name=="Secchi_Depth"){
       scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                  "SU"="#7CAE00"), na.value="#cccccc")
    } else {
       scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
    }} +
    scale_x_continuous(breaks = xbrks,
                       labels = xlabs)

p2 <- ggplot(data=plot_data,
             aes(x=relyear_dd, y=ResultValue, fill=VQ_Plot)) +
    geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
    # geom_abline(aes(slope=s_slope, intercept=s_int),
    #             color="#000099", size=1.2, alpha=0.7) +
    geom_segment(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", ], aes(x = relyear_dd,
                                                                                          y = relyear_dd,
                                                                                          xend = relyear_dd,
                                                                                          yend = relyear_dd),
                 color="#000099", size=1.2, alpha=0.7, inherit.aes = FALSE) +
    ylim(min_RV, y_scale) +
    labs(subtitle="Scaled to 4x Standard Deviation",
         x="Year", y=paste0("Values (", unit, ")")) +
    plot_theme +
    theme(legend.position="none") +
    {if(inc_H==TRUE){
       scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                  "HU"="#7CAE00"), na.value="#cccccc")
    } else if(param_name=="Secchi_Depth"){
       scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                  "SU"="#7CAE00"), na.value="#cccccc")
    } else {
       scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
    }} +
    scale_x_continuous(breaks = xbrks,
                       labels = xlabs)
```

```
      splot <- ggplot(plot_data, aes(x = relyear_dd, y = ResultValue)) +
         geom_point(shape = 21, size = 1.5, color="#333333", fill="#cccccc", alpha=0.75) +
         geom_segment(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season != "All", ], aes(x = relyear_dd,
                                                                                              y = relyear_dd,
                                                                                              xend = relyear_dd,
                                                                                              yend = relyear_dd,
                      color="#000099", size=1.2, alpha=0.7) +
         #ylim(min_RV-0.1*y_scale, y_scale) +
         scale_x_continuous(breaks = xbrks,
                            labels = xlabs) +
         labs(y = paste0("Values (", unit, ")"), x = "Year", subtitle = "Results for Individual Seasons",
         facet_wrap(~Season, ncol = 3) +
         plot_theme

      leg <- get_legend(p1)
      KTset <- ggarrange(leg, p1 + theme(legend.position="none"), p2,
                         splot, ncol=1, heights=c(0.1, 1, 1, 1.5))

      p0 <- ggplot() + labs(title=paste0(MA_Include[i])) +
         plot_theme + theme(panel.border=element_blank(),
                            panel.grid.major=element_blank(),
                            panel.grid.minor=element_blank(),
                            axis.line=element_blank())

      KT.Stats[ManagedAreaName==MA_Include[i], `:=` (N = N_Data,
                                                     Median = round(Median, 2),
                                                     Slope = round(SennSlope, 4),
                                                     Int. = round(SennIntercept, 4),
                                                     z = round(z, 1),
                                                     chi_sq = round(chi_sq, 1))]

      print(ggarrange(p0, KTset, ncol=1, heights=c(0.1, 1.25)))
      cat('\n')
      print(KT.Stats[KT.Stats$ManagedAreaName==MA_Include[i], ] %>%
         select(Season, N, Median, tau, Slope, Int., z, p_z, chi_sq, p_chi_sq, Trend) %>%
         kable(format="latex") %>%
            row_spec(0,bold=TRUE) %>%
            kable_styling(latex_options = "HOLD_position",
                          font_size = 7) %>%
            add_footnote(
            "p < 0.00005 appear as 0 due to rounding"))
      cat('\n')
      rm(plot_data)
      rm(KTset, leg)
   }
}
```

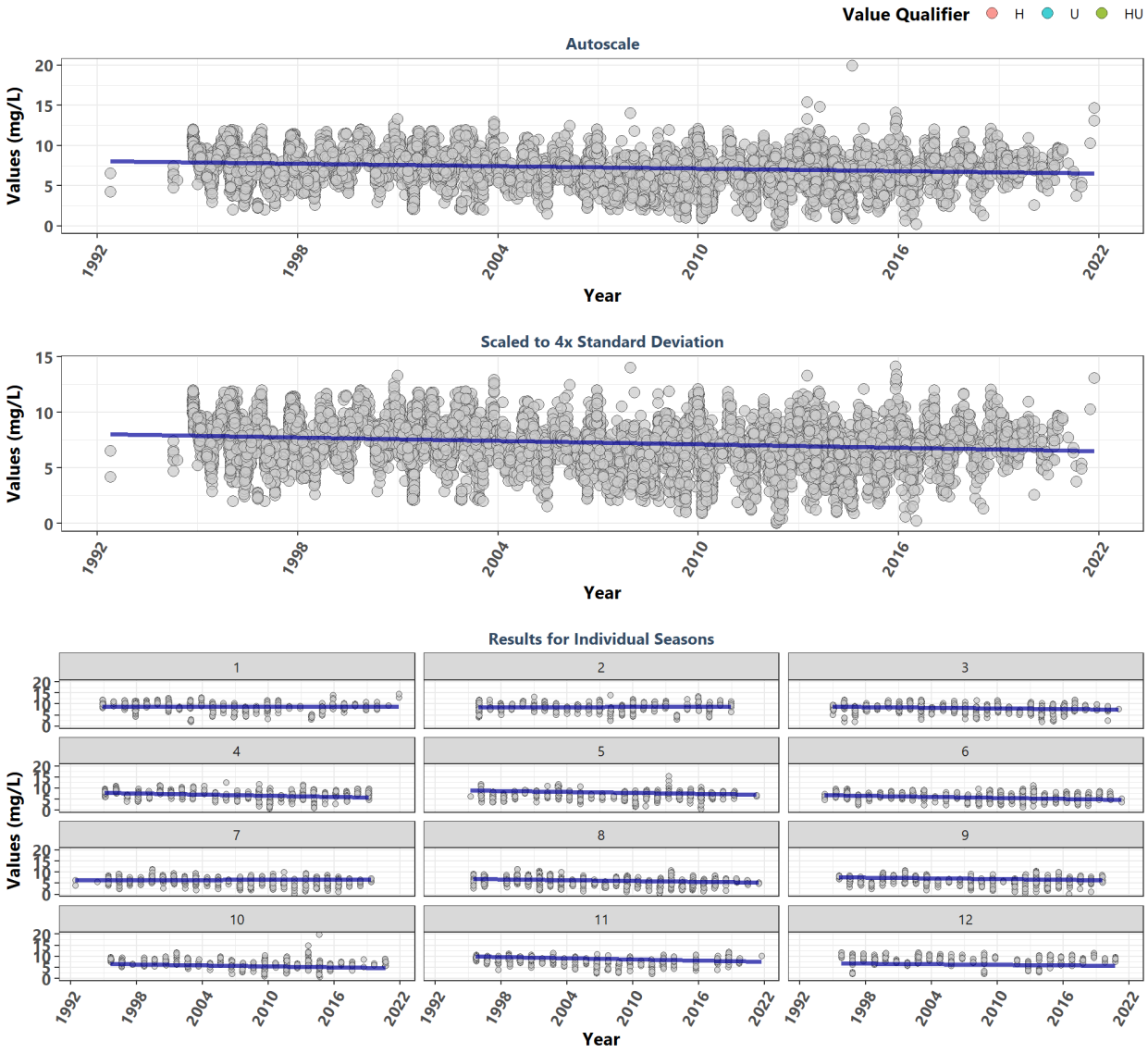
Alligator Harbor Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2394	7.10	-0.1575	-0.0516	7.5467	-9.3	0.0000	108.2	0	-1
1	69	8.39	-0.1829	-0.0600	7.3000	-6.0	0.0000	NA	NA	-1
2	189	8.30	-0.2238	-0.0667	7.3667	-6.6	0.0000	NA	NA	-1
3	158	8.10	-0.2008	-0.1067	7.7267	-2.0	0.0468	NA	NA	-1
4	277	7.30	-0.4329	-0.2500	8.8250	-0.6	0.5784	NA	NA	-1
5	234	6.80	0.2387	0.0500	7.9500	-4.8	0.0000	NA	NA	1
6	418	6.00	-0.4753	-0.3429	9.7614	-1.4	0.1696	NA	NA	-1
7	275	6.30	-0.3216	-0.1182	9.1273	-6.1	0.0000	NA	NA	-1
8	165	7.00	-0.1038	-0.0313	8.2875	-3.6	0.0004	NA	NA	-1
9	241	6.90	-0.0221	-0.0061	7.3364	-5.2	0.0000	NA	NA	-1
10	162	7.30	-0.2086	-0.0733	7.1667	-3.9	0.0001	NA	NA	-1
11	88	7.70	-0.0447	-0.0200	6.2500	-6.1	0.0000	NA	NA	-1
12	118	7.95	-0.2432	-0.1111	7.1889	4.2	0.0000	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

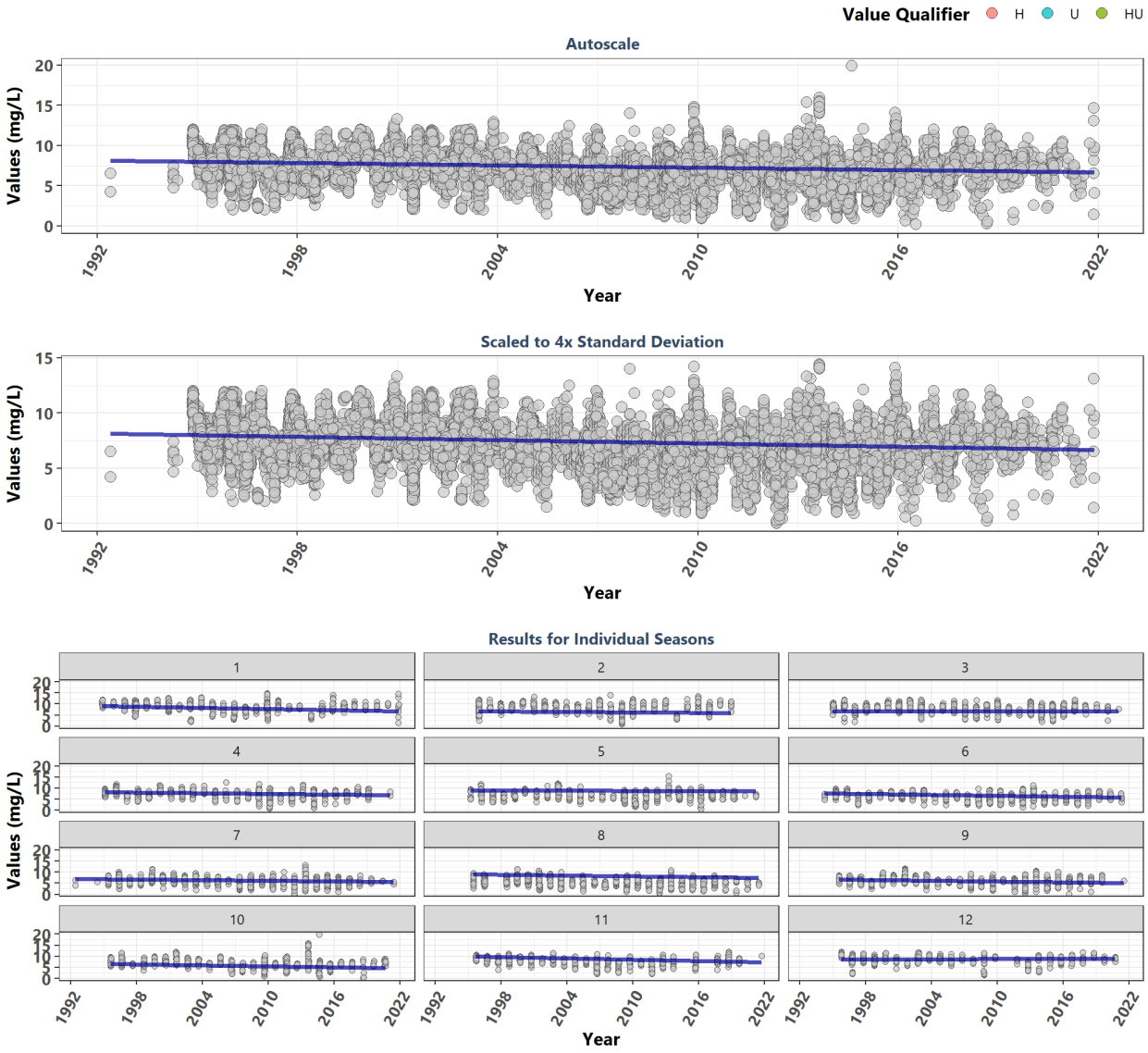
Apalachicola Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	11543	7.00	-0.1458	-0.0522	8.0583	-24.1	0.0000	293.7	0	-1
1	780	9.50	0.0045	0.0000	8.8000	-10.7	0.0000	NA	NA	-1
2	766	8.60	0.0382	0.0124	8.4762	1.6	0.1122	NA	NA	1
3	1233	8.50	-0.1733	-0.0526	9.0789	-9.1	0.0000	NA	NA	-1
4	958	7.27	-0.2470	-0.0833	8.0667	-6.8	0.0000	NA	NA	-1
5	958	6.60	-0.2160	-0.0667	9.0333	2.1	0.0354	NA	NA	-1
6	1177	6.20	-0.2351	-0.0714	6.9714	-5.5	0.0000	NA	NA	-1
7	1339	5.90	0.0453	0.0143	6.3429	-12.9	0.0000	NA	NA	1
8	1458	5.90	-0.1570	-0.0545	7.0373	-11.7	0.0000	NA	NA	-1
9	928	6.11	-0.1470	-0.0600	8.0500	-7.2	0.0000	NA	NA	-1
10	722	6.90	-0.2040	-0.0750	6.7250	-10.0	0.0000	NA	NA	-1
11	665	8.10	-0.2542	-0.0900	10.2200	-8.4	0.0000	NA	NA	-1
12	559	8.80	-0.1069	-0.0357	6.8071	0.2	0.8733	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

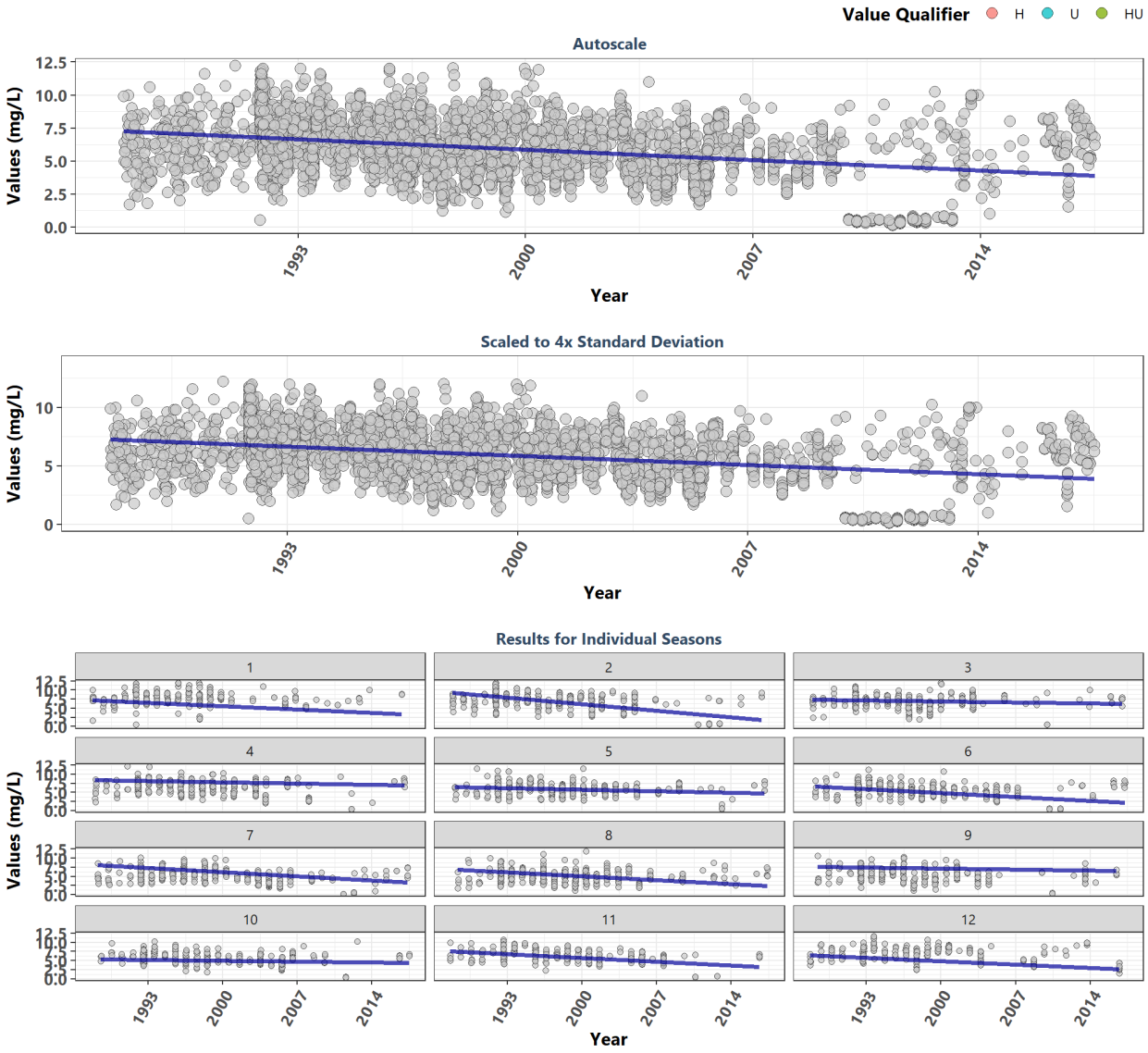
Apalachicola National Estuarine Research Reserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	16348	7.20	-0.1479	-0.0500	8.1487	-29.3	0.0000	366.9	0	-1
1	1218	9.40	-0.2431	-0.0789	9.2053	-14.5	0.0000	NA	NA	-1
2	1136	8.66	-0.1054	-0.0312	6.7312	2.6	0.0084	NA	NA	-1
3	1864	8.50	0.0009	0.0000	6.7000	-13.1	0.0000	NA	NA	-1
4	1555	7.70	-0.1478	-0.0545	8.4091	-8.8	0.0000	NA	NA	-1
5	1317	6.70	-0.0258	-0.0086	8.8600	0.1	0.9601	NA	NA	-1
6	1373	6.23	-0.2089	-0.0706	7.8882	-6.8	0.0000	NA	NA	-1
7	1688	5.90	-0.1229	-0.0400	6.9100	-11.8	0.0000	NA	NA	-1
8	1810	5.90	-0.2019	-0.0625	9.1875	-13.3	0.0000	NA	NA	-1
9	1253	6.20	-0.1914	-0.0600	6.8600	-5.6	0.0000	NA	NA	-1
10	1272	6.90	-0.2074	-0.0721	6.6936	-11.2	0.0000	NA	NA	-1
11	1037	8.10	-0.2744	-0.1000	10.2000	-11.8	0.0000	NA	NA	-1
12	825	8.80	0.0519	0.0167	8.4933	-1.1	0.2651	NA	NA	1

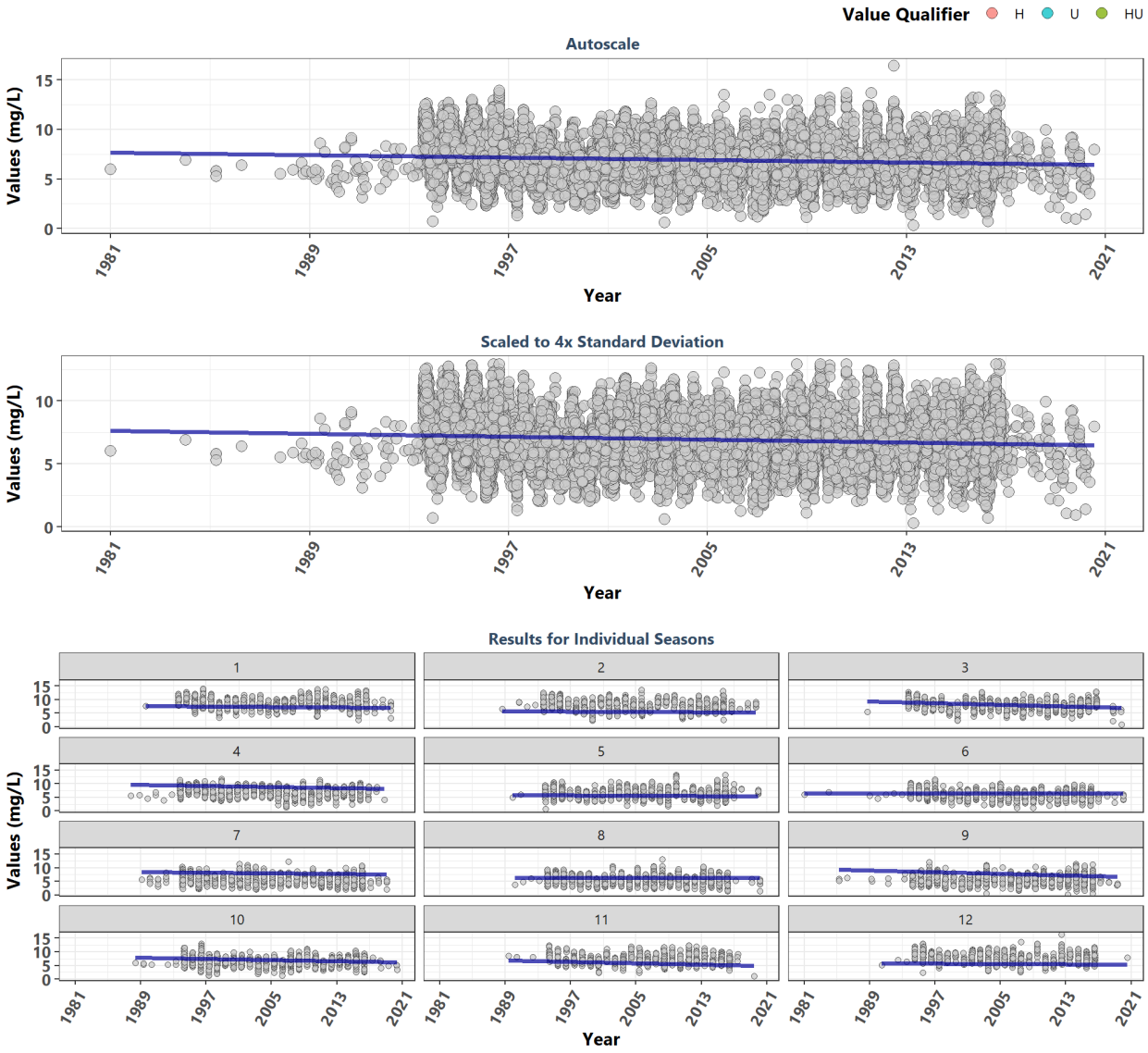
[a] $p < 0.00005$ appear as 0 due to rounding

Banana River Aquatic Preserve



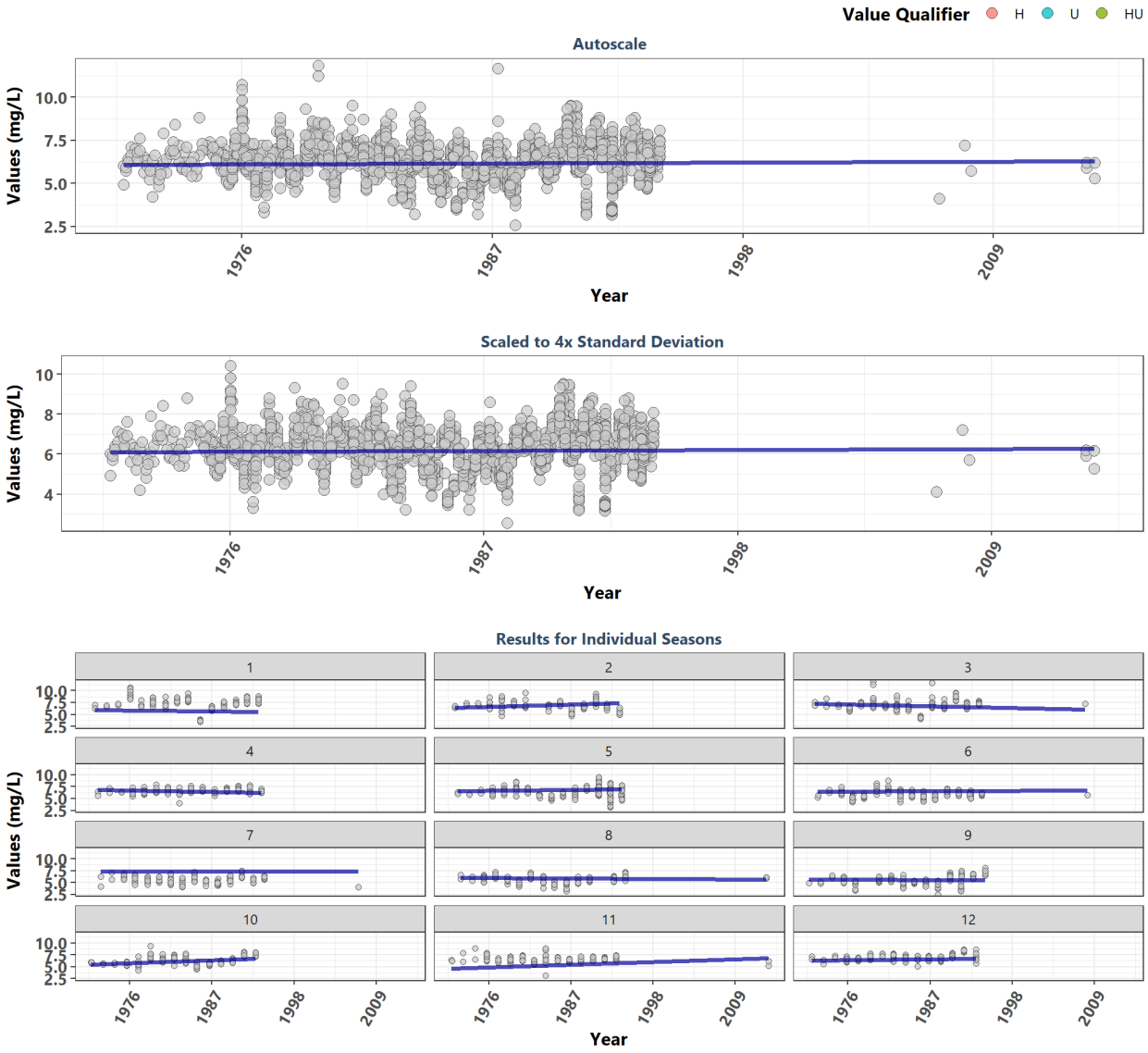
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5719	6.10	-0.2201	-0.1133	7.4700	-25.1	0.0000	167.4	0	-1
1	330	8.00	-0.3106	-0.1333	7.5000	-2.7	0.0073	NA	NA	-1
2	546	7.48	-0.3536	-0.2573	9.7960	-12.6	0.0000	NA	NA	-1
3	426	7.00	-0.0881	-0.0400	7.4400	-2.7	0.0063	NA	NA	-1
4	544	6.50	-0.0981	-0.0500	8.4500	-8.3	0.0000	NA	NA	-1
5	611	6.00	-0.1477	-0.0612	6.6738	-5.5	0.0000	NA	NA	-1
6	520	5.50	-0.3173	-0.1500	6.9000	-10.4	0.0000	NA	NA	-1
7	546	5.10	-0.2379	-0.1600	8.4200	-11.1	0.0000	NA	NA	-1
8	574	5.04	-0.3044	-0.1500	7.1500	-2.4	0.0149	NA	NA	-1
9	414	5.50	-0.0680	-0.0400	7.6900	-7.8	0.0000	NA	NA	-1
10	488	5.90	-0.0676	-0.0300	5.3950	-10.3	0.0000	NA	NA	-1
11	397	6.20	-0.3216	-0.1500	7.8500	-9.7	0.0000	NA	NA	-1
12	323	7.29	-0.2524	-0.1286	6.6571	-1.8	0.0652	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	31316	6.5	-0.1031	-0.0300	7.6458	-25.8	0.000	379.8	0	-1
1	1818	8.8	-0.0766	-0.0250	7.9250	-8.7	0.000	NA	NA	-1
2	1931	8.1	-0.0535	-0.0143	5.8571	-13.8	0.000	NA	NA	-1
3	2786	7.7	-0.2089	-0.0750	9.9000	-15.8	0.000	NA	NA	-1
4	2986	6.9	-0.1361	-0.0500	10.0000	-13.8	0.000	NA	NA	-1
5	2829	6.3	-0.0544	-0.0154	5.9846	-0.8	0.410	NA	NA	-1
6	2633	5.9	-0.0093	0.0000	6.4000	-18.0	0.000	NA	NA	-1
7	2894	5.5	-0.0923	-0.0308	8.7385	-4.4	0.000	NA	NA	-1
8	3362	5.6	-0.0103	0.0000	6.3000	-4.7	0.000	NA	NA	-1
9	3031	5.5	-0.1998	-0.0778	9.6444	-4.4	0.000	NA	NA	-1
10	2932	6.4	-0.1685	-0.0500	8.1500	-0.8	0.449	NA	NA	-1
11	2148	7.3	-0.2334	-0.0611	7.3667	-5.3	0.000	NA	NA	-1
12	1966	8.0	-0.0544	-0.0154	5.8692	-6.1	0.000	NA	NA	-1

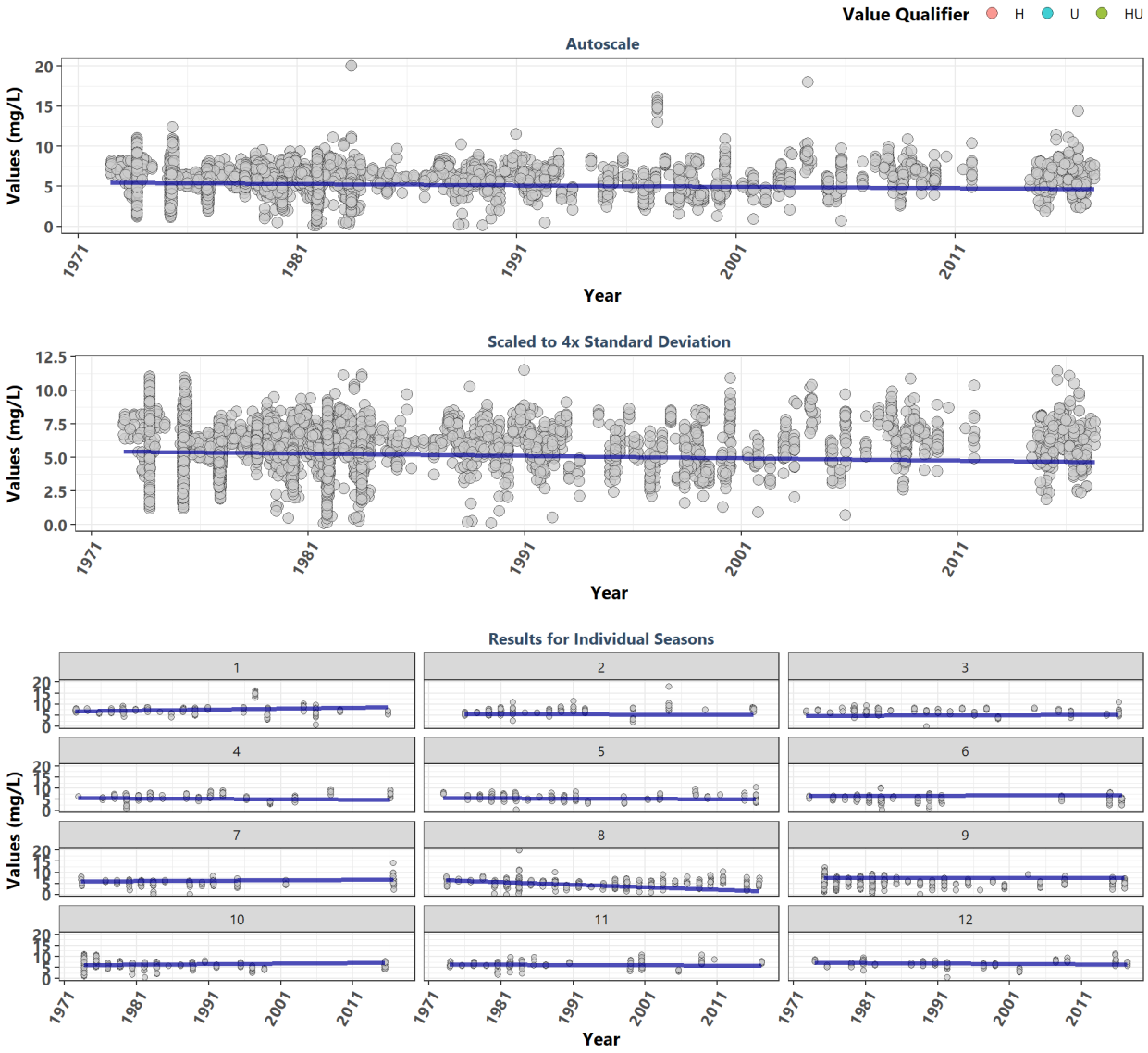
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1802	6.39	0.0374	0.0063	5.9486	1.4	0.1531	61.3	0	0
1	136	7.38	-0.0670	-0.0200	6.3400	-0.1	0.9323	NA	NA	0
2	125	6.80	0.2289	0.0675	4.8075	-2.9	0.0034	NA	NA	0
3	148	6.80	-0.1766	-0.0571	8.6857	1.7	0.0860	NA	NA	0
4	136	6.50	-0.0839	-0.0436	7.8709	2.1	0.0379	NA	NA	0
5	268	6.30	0.0949	0.0300	5.8400	-2.1	0.0374	NA	NA	0
6	150	5.88	0.0771	0.0143	6.0571	-0.7	0.4612	NA	NA	0
7	147	5.70	-0.0050	0.0000	7.3800	-1.2	0.2276	NA	NA	0
8	136	5.62	-0.0405	-0.0117	6.2583	-0.5	0.6504	NA	NA	0
9	146	5.38	-0.0263	-0.0046	5.7565	3.7	0.0002	NA	NA	0
10	137	6.10	0.2220	0.0817	3.5683	3.9	0.0001	NA	NA	0
11	138	6.50	0.2043	0.0812	2.6988	1.3	0.1785	NA	NA	0
12	135	6.90	0.1198	0.0214	5.8143	4.0	0.0001	NA	NA	0

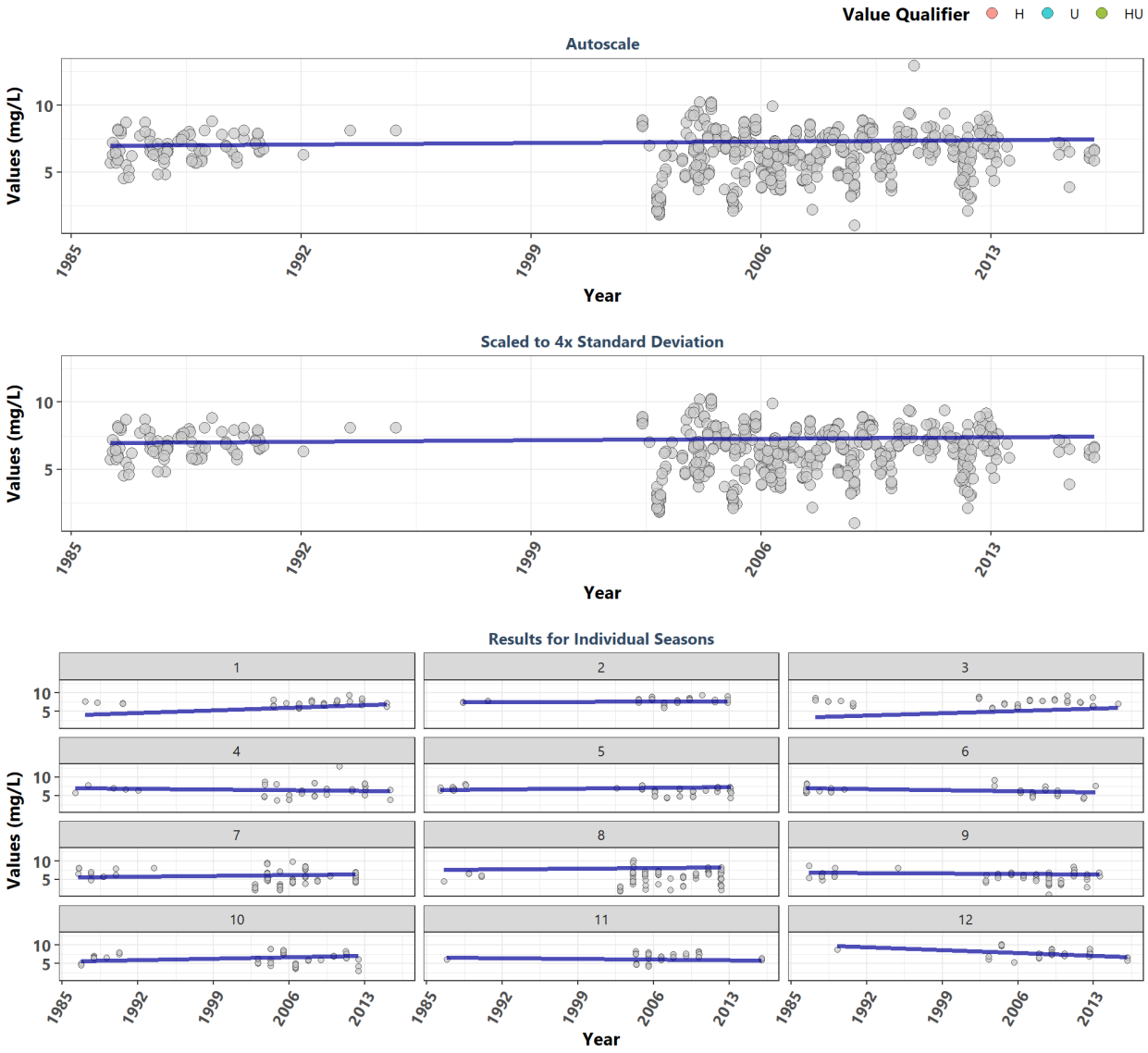
[a] $p < 0.00005$ appear as 0 due to rounding

Boca Ciega Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6160	5.76	-0.0445	-0.0293	6.0494	-3.9	0.0001	222.5	0	-1
1	216	7.50	0.2490	0.0667	5.4800	0.2	0.8110	NA	NA	1
2	174	7.28	-0.0377	-0.0082	5.7406	4.9	0.0000	NA	NA	-1
3	185	6.78	0.0585	0.0227	4.2014	-2.3	0.0194	NA	NA	1
4	175	6.44	-0.0296	-0.0217	5.9500	1.9	0.0579	NA	NA	-1
5	212	6.14	-0.0789	-0.0235	6.1488	-2.0	0.0456	NA	NA	-1
6	175	5.51	0.0350	0.0100	6.4800	-0.7	0.4567	NA	NA	1
7	149	5.49	0.0882	0.0333	5.3000	-1.4	0.1509	NA	NA	1
8	562	4.81	-0.3082	-0.1931	10.6977	2.2	0.0284	NA	NA	-1
9	3156	5.43	0.0109	0.0029	7.4009	-2.6	0.0108	NA	NA	1
10	783	6.45	0.0961	0.0376	5.3114	-13.3	0.0000	NA	NA	1
11	207	6.50	-0.0919	-0.0247	6.8985	1.9	0.0559	NA	NA	-1
12	166	6.80	-0.1152	-0.0320	7.7720	0.7	0.5009	NA	NA	-1

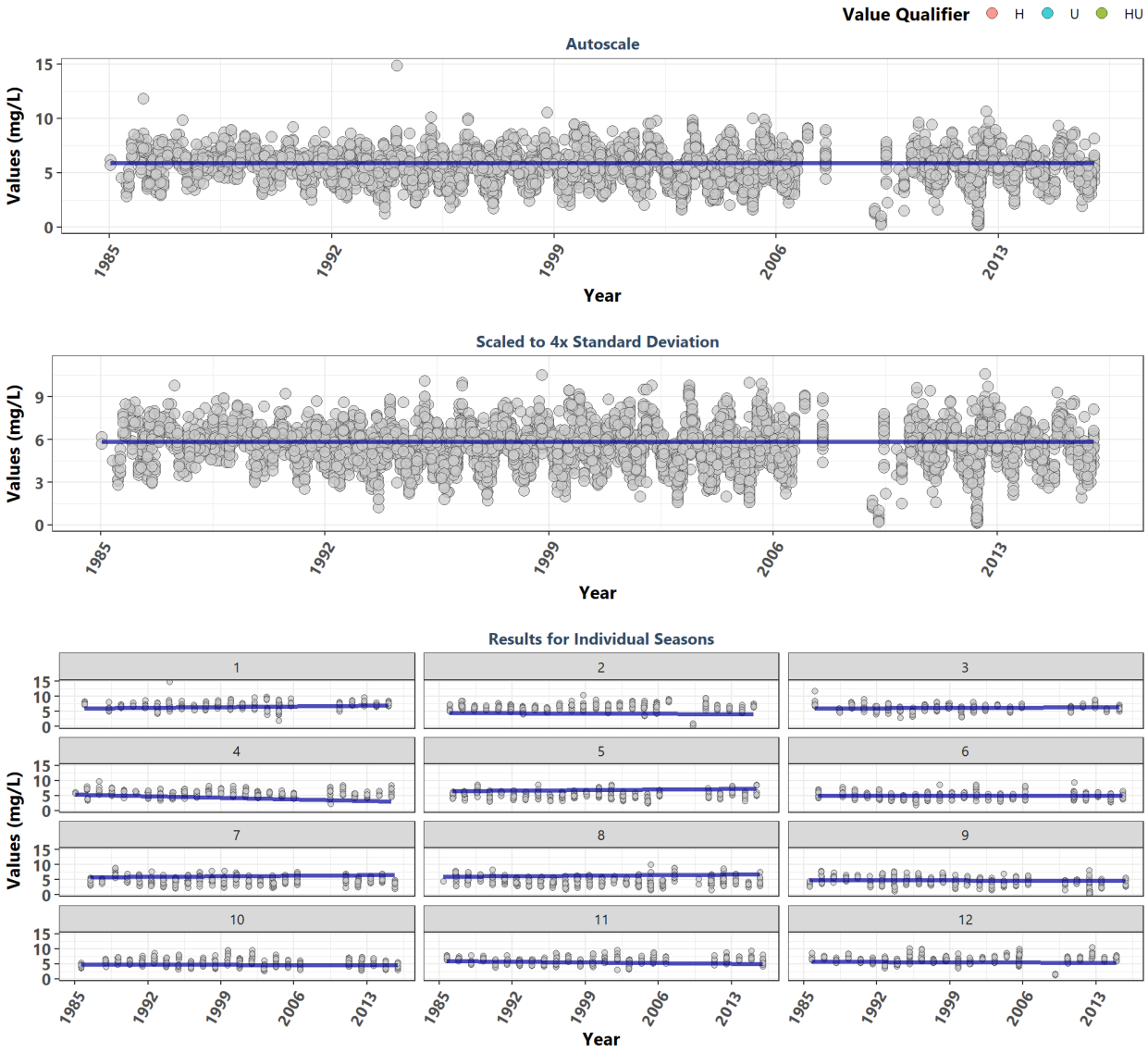
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	605	6.47	0.0252	0.0187	6.8575	1.7	0.0911	25.3	0.0083	0
1	36	7.20	0.1886	0.1192	3.2283	1.4	0.1736	NA	NA	0
2	35	8.10	0.0315	0.0068	7.5425	1.3	0.2064	NA	NA	0
3	47	7.70	0.1165	0.1060	2.6680	0.3	0.7612	NA	NA	0
4	42	6.22	-0.1600	-0.0300	7.1200	0.7	0.4914	NA	NA	0
5	51	6.40	0.1587	0.0343	6.3207	-1.7	0.0969	NA	NA	0
6	37	6.20	-0.2417	-0.0453	7.2427	-2.1	0.0343	NA	NA	0
7	99	5.00	0.0743	0.0315	5.4494	1.7	0.0855	NA	NA	0
8	80	5.85	0.1496	0.0250	7.5000	2.5	0.0125	NA	NA	0
9	60	6.02	-0.0398	-0.0200	6.9400	-1.1	0.2561	NA	NA	0
10	47	6.50	0.0952	0.0610	5.3060	-0.4	0.6982	NA	NA	0
11	43	6.71	-0.1006	-0.0315	6.7750	0.9	0.3653	NA	NA	0
12	28	7.62	-0.2143	-0.1270	10.8007	-1.6	0.1095	NA	NA	0

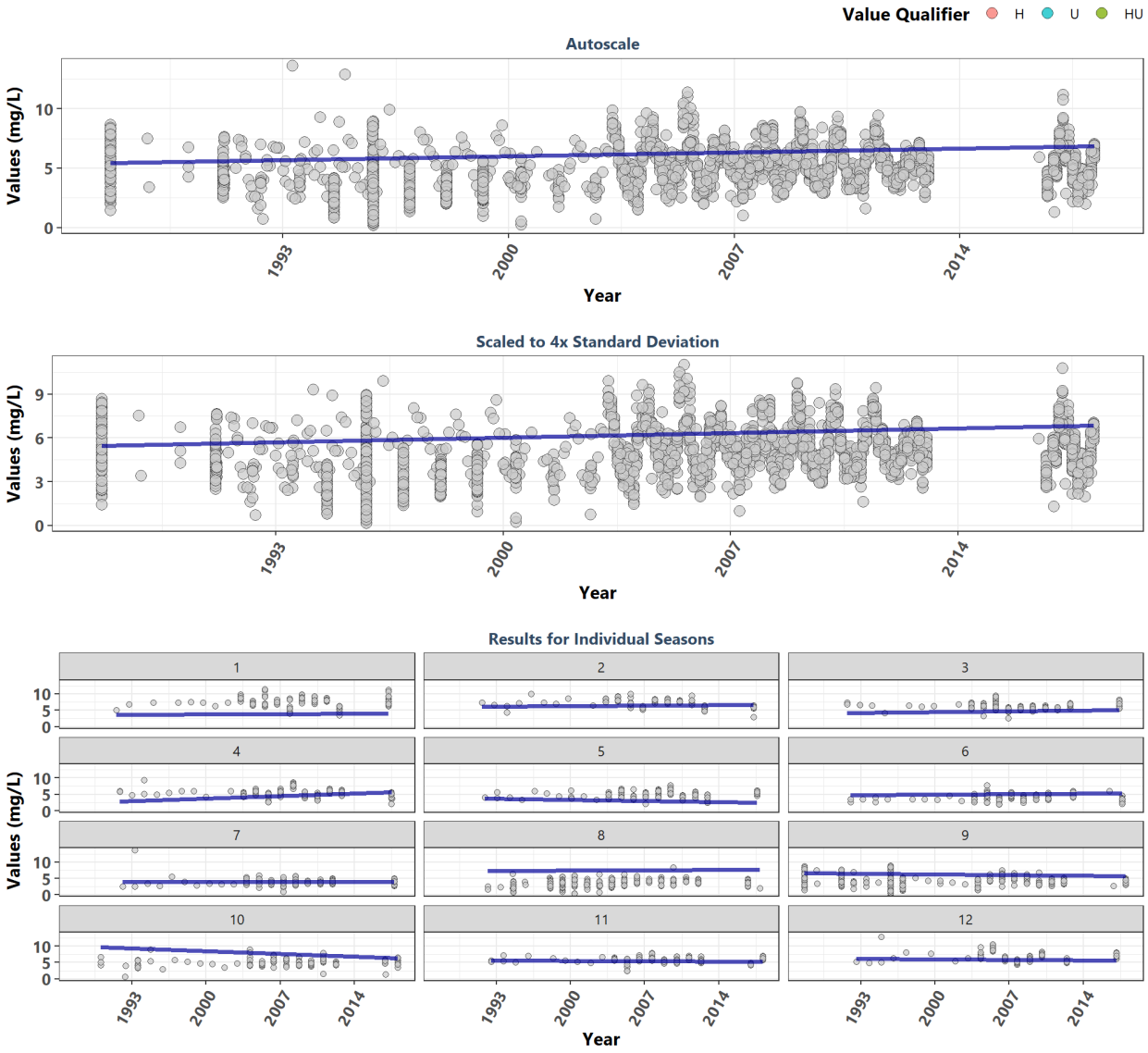
[a] $p < 0.00005$ appear as 0 due to rounding

Cape Romano-Ten Thousand Islands Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4715	5.50	-0.0078	0.0000	5.8625	-1.4	0.1634	149	0	0
1	342	6.90	0.1400	0.0375	5.8850	2.8	0.0055	NA	NA	0
2	397	6.40	-0.0816	-0.0222	4.7100	3.8	0.0001	NA	NA	0
3	314	6.19	0.0726	0.0200	5.8700	1.9	0.0545	NA	NA	0
4	398	5.62	-0.2692	-0.0925	5.8650	-1.9	0.0602	NA	NA	0
5	421	5.00	0.1003	0.0278	6.4278	0.1	0.8875	NA	NA	0
6	377	4.80	0.0046	0.0000	5.0000	-1.2	0.2343	NA	NA	0
7	414	4.68	0.1363	0.0308	5.5769	-1.0	0.3168	NA	NA	0
8	449	4.31	0.1278	0.0300	5.8600	-2.6	0.0097	NA	NA	0
9	402	4.20	-0.0409	-0.0091	4.9545	-8.1	0.0000	NA	NA	0
10	448	5.40	-0.0329	-0.0091	4.8386	-3.6	0.0003	NA	NA	0
11	383	6.10	-0.1150	-0.0333	6.0333	4.0	0.0001	NA	NA	0
12	370	6.48	-0.0629	-0.0154	5.8765	4.0	0.0001	NA	NA	0

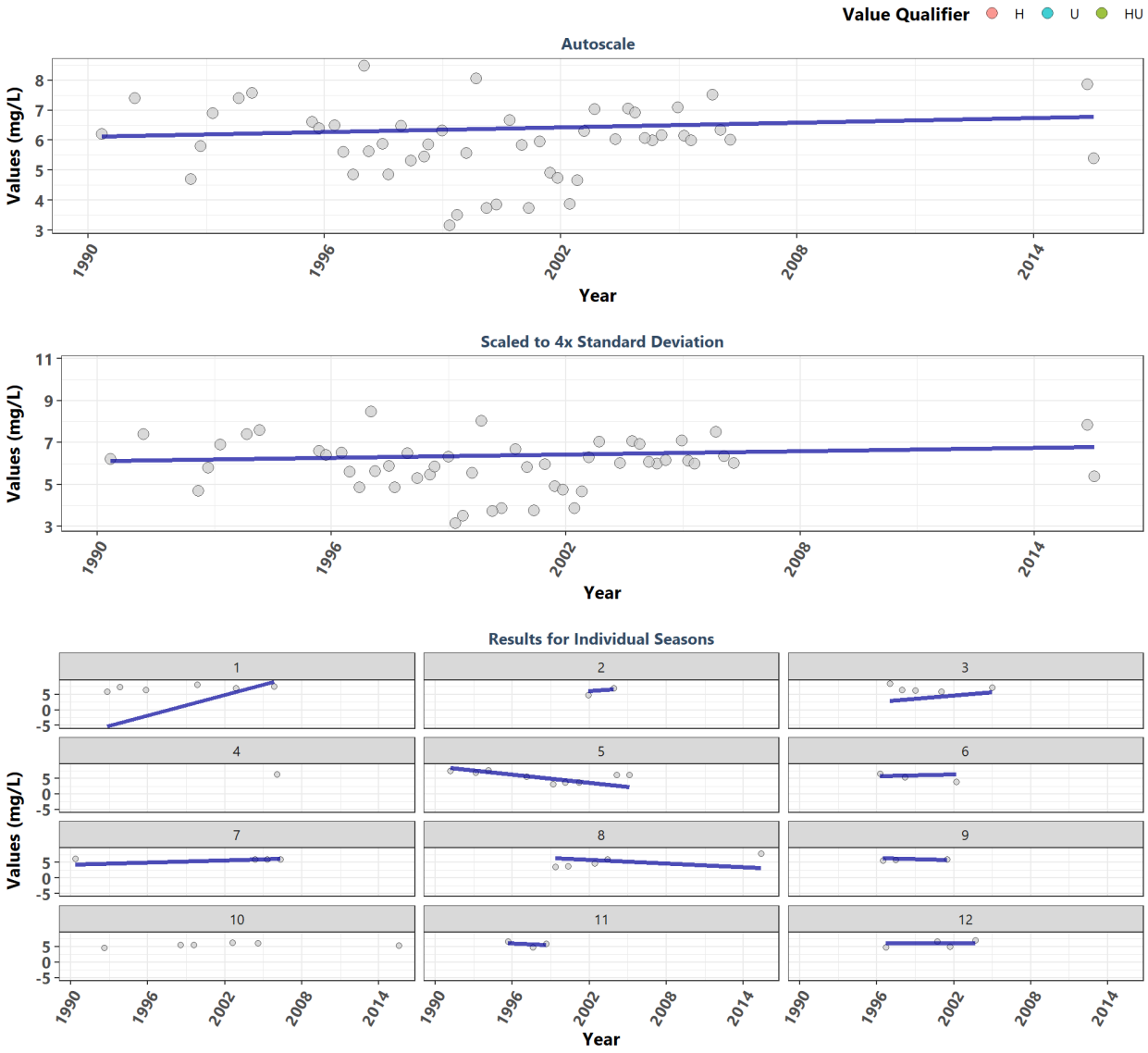
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3854	4.01	0.0519	0.0533	5.1160	9.0	0.0000	275.4	0	1
1	146	7.58	0.0515	0.0157	3.4829	0.4	0.6632	NA	NA	1
2	149	7.38	0.0423	0.0218	5.9364	-5.6	0.0000	NA	NA	1
3	153	6.00	0.1163	0.0425	3.7200	-1.8	0.0774	NA	NA	1
4	162	5.37	0.2506	0.1225	1.6825	-0.6	0.5430	NA	NA	1
5	167	4.74	-0.1037	-0.0564	4.3636	2.2	0.0252	NA	NA	-1
6	149	3.86	0.0857	0.0222	4.5667	0.9	0.3494	NA	NA	1
7	147	3.99	0.0079	0.0017	3.9500	0.1	0.8875	NA	NA	1
8	1319	3.52	0.0243	0.0127	7.2583	14.3	0.0000	NA	NA	1
9	993	3.63	-0.0959	-0.0309	6.7418	-5.1	0.0000	NA	NA	-1
10	173	5.10	-0.3064	-0.1374	10.8147	1.7	0.0931	NA	NA	-1
11	170	5.84	-0.0321	-0.0123	5.6654	-1.6	0.1110	NA	NA	-1
12	126	6.46	-0.0820	-0.0260	6.4850	0.7	0.4804	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

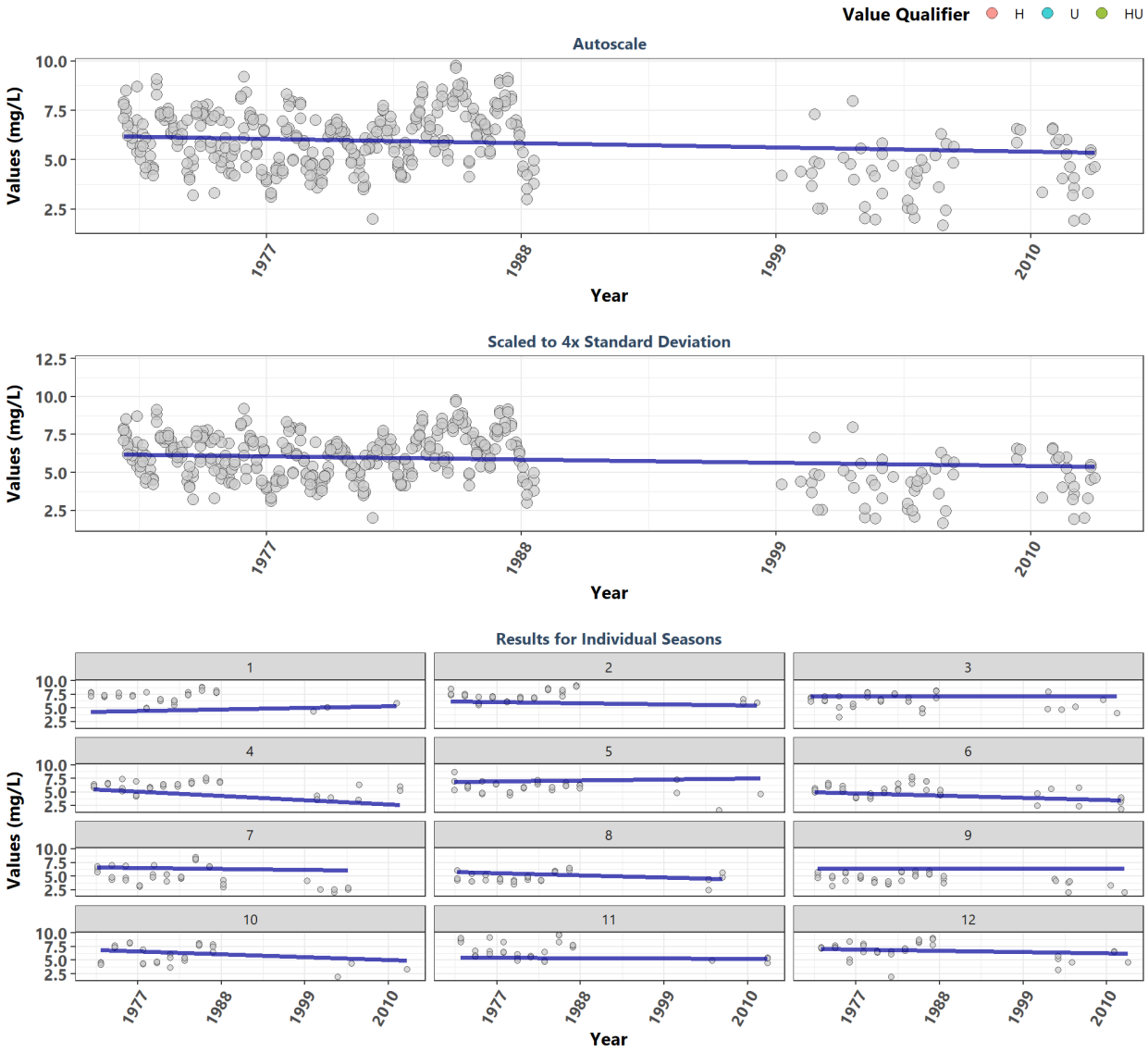
Coupon Bight Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	51	6.01	0.1627	0.0257	6.1214	0.8	0.4492	17.3	0.0686	0
1	6	7.21	1.0000	1.0921	-8.3649	1.1	0.2597	NA	NA	0
2	2	NA	0.6667	0.2223	3.4554	NA	NA	NA	NA	NA
3	5	6.48	1.0000	0.3643	0.2942	-0.7	0.4624	NA	NA	0
4	1	NA	0.4667	0.1117	6.3198	NA	NA	NA	NA	NA
5	9	6.07	-1.0000	-0.4380	8.8142	-0.7	0.4655	NA	NA	0
6	3	5.31	1.0000	0.0710	5.3733	-1.0	0.2963	NA	NA	0
7	4	6.00	0.3333	0.1191	4.2533	-0.3	0.7341	NA	NA	0
8	5	4.67	-0.4000	-0.1950	8.2352	2.2	0.0275	NA	NA	0
9	3	5.87	-0.2222	-0.1169	7.1196	1.0	0.2963	NA	NA	0
10	6	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	3	5.86	-0.3333	-0.2467	7.5867	0.0	1.0000	NA	NA	0
12	4	5.79	-0.3333	-0.0081	6.1214	1.0	0.3082	NA	NA	0

[a] $p < 0.00005$ appear as 0 due to rounding

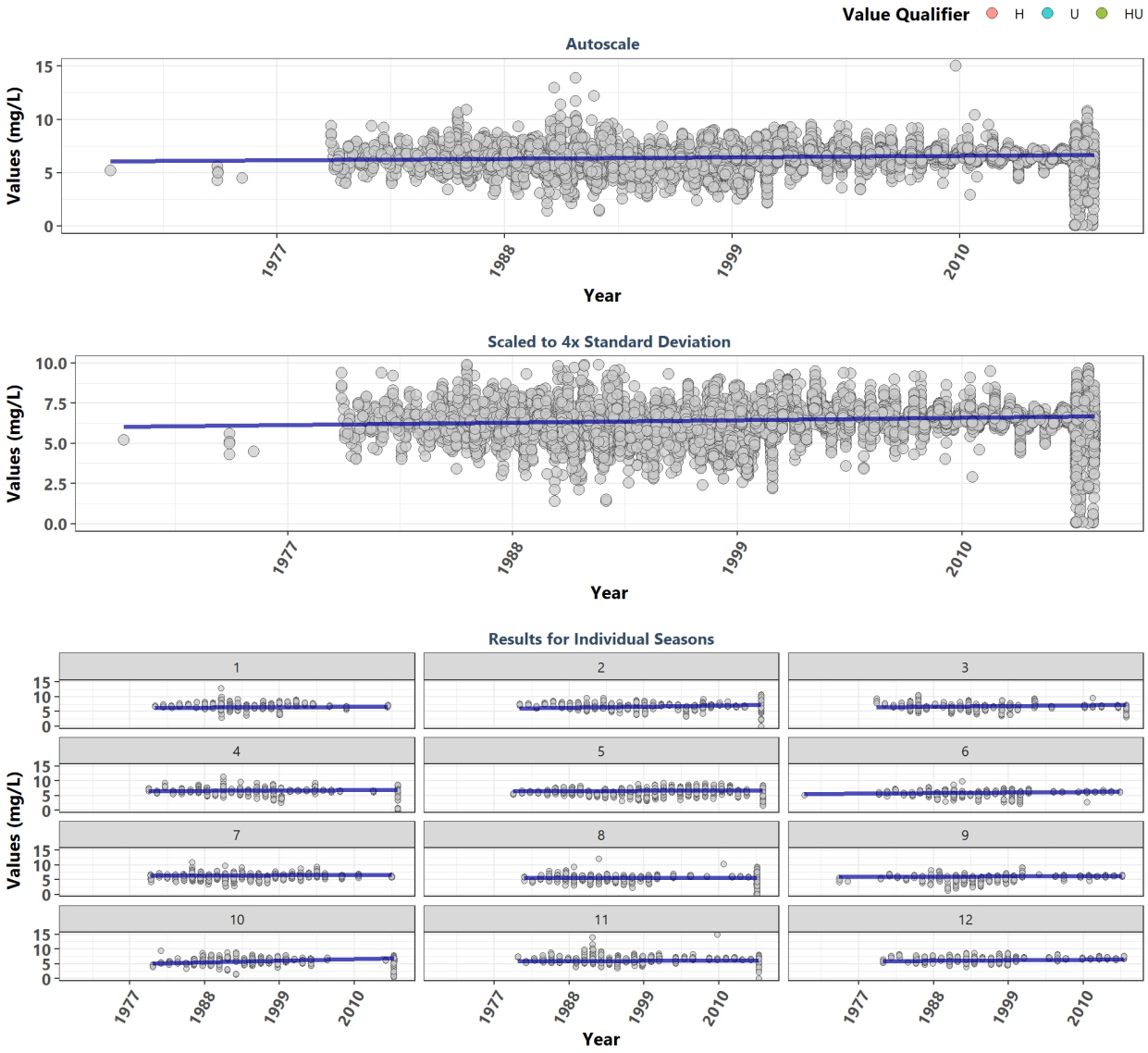
Estero Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	412	5.97	-0.0903	-0.0351	7.1455	-2.8	0.0056	13.8	0.2424	-1
1	33	7.20	0.1634	0.0453	3.0640	-0.1	0.9379	NA	NA	1
2	33	7.03	-0.0784	-0.0337	7.0910	0.4	0.6742	NA	NA	-1
3	36	6.40	-0.0114	0.0000	7.2000	-0.7	0.5030	NA	NA	-1
4	37	6.36	-0.2763	-0.1267	8.9700	0.1	0.9581	NA	NA	-1
5	34	5.98	0.0530	0.0300	6.0400	-0.6	0.5220	NA	NA	1
6	39	5.35	-0.2342	-0.0651	6.7540	-1.8	0.0686	NA	NA	-1
7	37	4.79	-0.0794	-0.0283	7.3542	-2.4	0.0163	NA	NA	-1
8	31	4.56	-0.2038	-0.0667	7.6167	1.3	0.1999	NA	NA	-1
9	37	4.54	0.0075	0.0038	6.2325	-2.0	0.0419	NA	NA	1
10	30	5.48	-0.2280	-0.0843	9.2014	-0.2	0.8717	NA	NA	-1
11	31	6.42	-0.0230	-0.0100	5.8100	-1.8	0.0727	NA	NA	-1
12	34	6.90	-0.1123	-0.0405	8.2407	-0.9	0.3559	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

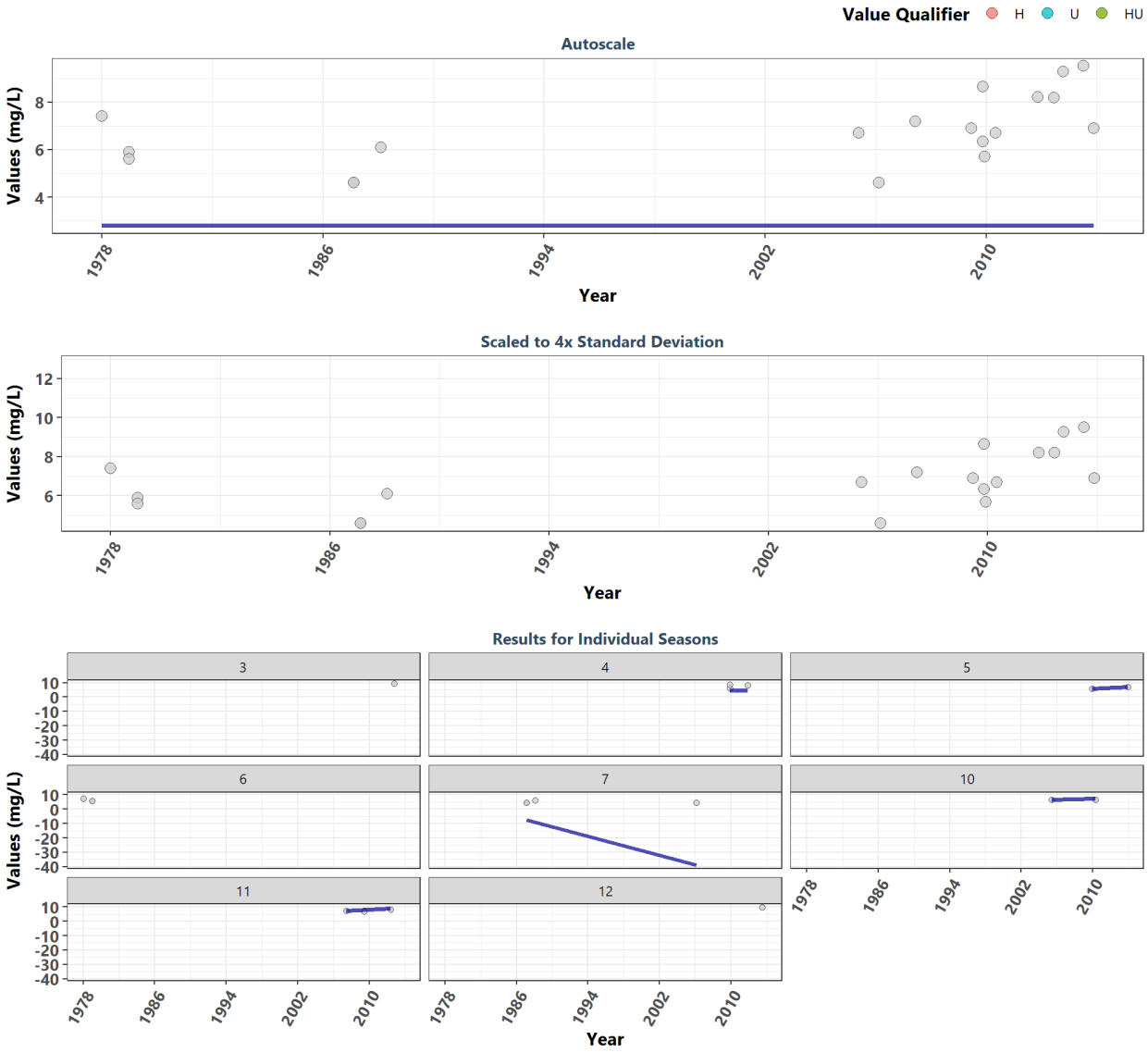
Florida Keys National Marine Sanctuary



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	15592	6.27	0.1267	0.0165	5.8405	22.4	0.0000	446.6	0	1
1	1134	6.85	0.0931	0.0150	5.9411	7.8	0.0000	NA	NA	1
2	1289	6.82	0.2272	0.0400	5.1493	4.5	0.0000	NA	NA	1
3	1223	6.70	0.1536	0.0254	5.9641	2.0	0.0454	NA	NA	1
4	1049	6.50	0.0833	0.0109	6.3606	3.7	0.0002	NA	NA	1
5	2255	6.30	0.0381	0.0048	6.5237	9.5	0.0000	NA	NA	1
6	992	6.07	0.1703	0.0215	5.3362	8.1	0.0000	NA	NA	1
7	1640	6.01	0.0761	0.0089	6.1689	6.8	0.0000	NA	NA	1
8	1241	5.68	-0.0297	-0.0044	5.8281	-1.6	0.1152	NA	NA	-1
9	1187	6.10	0.0442	0.0071	5.8528	23.8	0.0000	NA	NA	1
10	1312	6.09	0.4605	0.0565	3.9548	2.4	0.0160	NA	NA	1
11	1259	6.48	0.1118	0.0110	5.6163	5.0	0.0000	NA	NA	1
12	1011	6.67	0.1332	0.0175	5.5983	10.8	0.0000	NA	NA	1

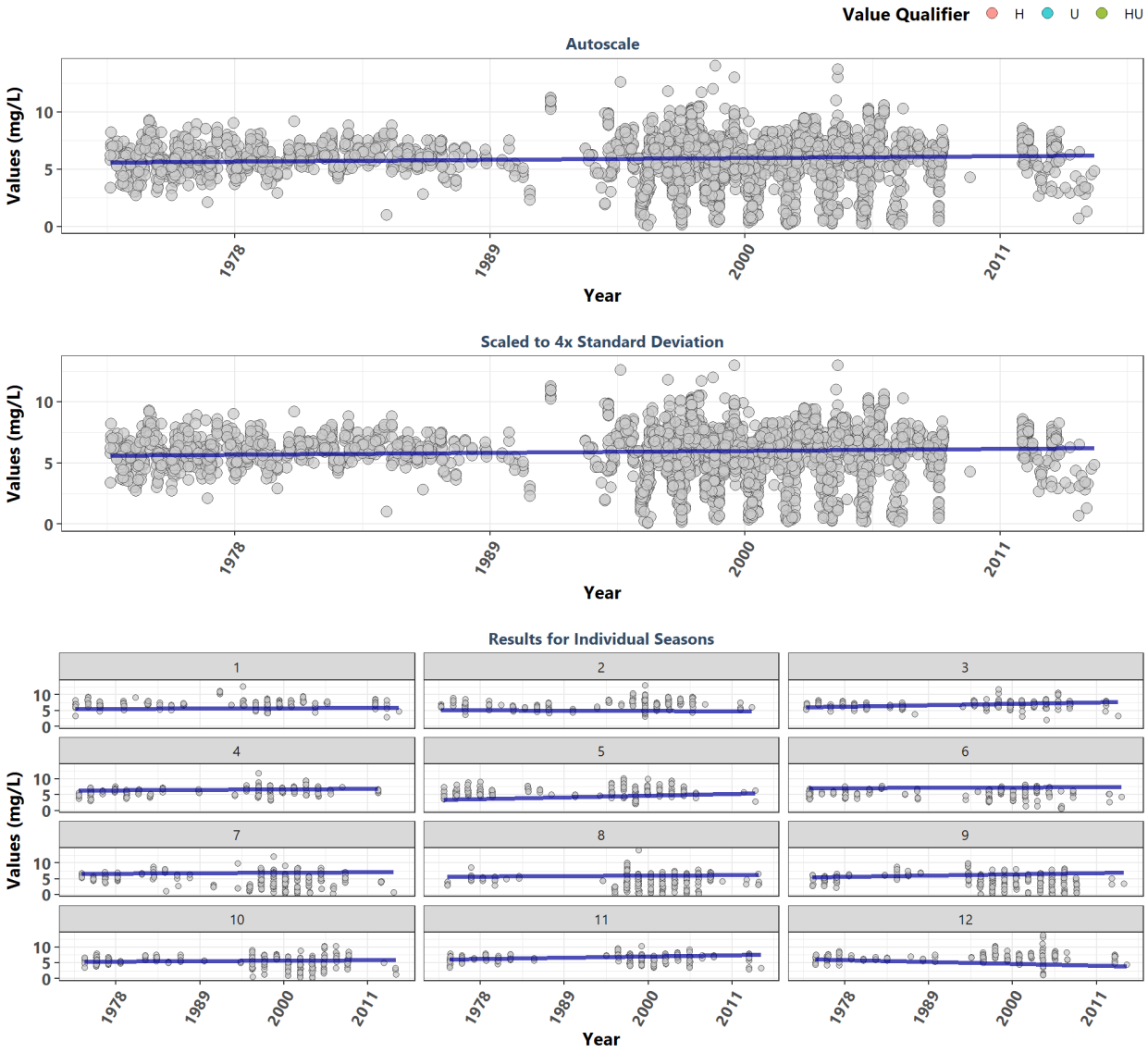
[a] $p < 0.00005$ appear as 0 due to rounding

Fort Pickens State Park Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	19	6.7	0.0980	0.0000	2.80	0	1	NaN	NaN	0
3	1	NA	0.0000	0.0000	6.70	NA	NA	NA	NA	NA
4	3	NA	0.1667	0.0000	4.60	NA	NA	NA	NA	NA
5	2	NA	1.0000	0.3000	-3.90	NA	NA	NA	NA	NA
6	3	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	4	4.6	-0.6667	-1.6500	7.55	0	1	NA	NA	0
10	2	NA	0.3333	0.2000	1.00	NA	NA	NA	NA	NA
11	3	7.2	0.0000	0.3575	-3.22	0	1	NA	NA	0
12	1	NA	NA	NA	NA	NA	NA	NA	NA	NA

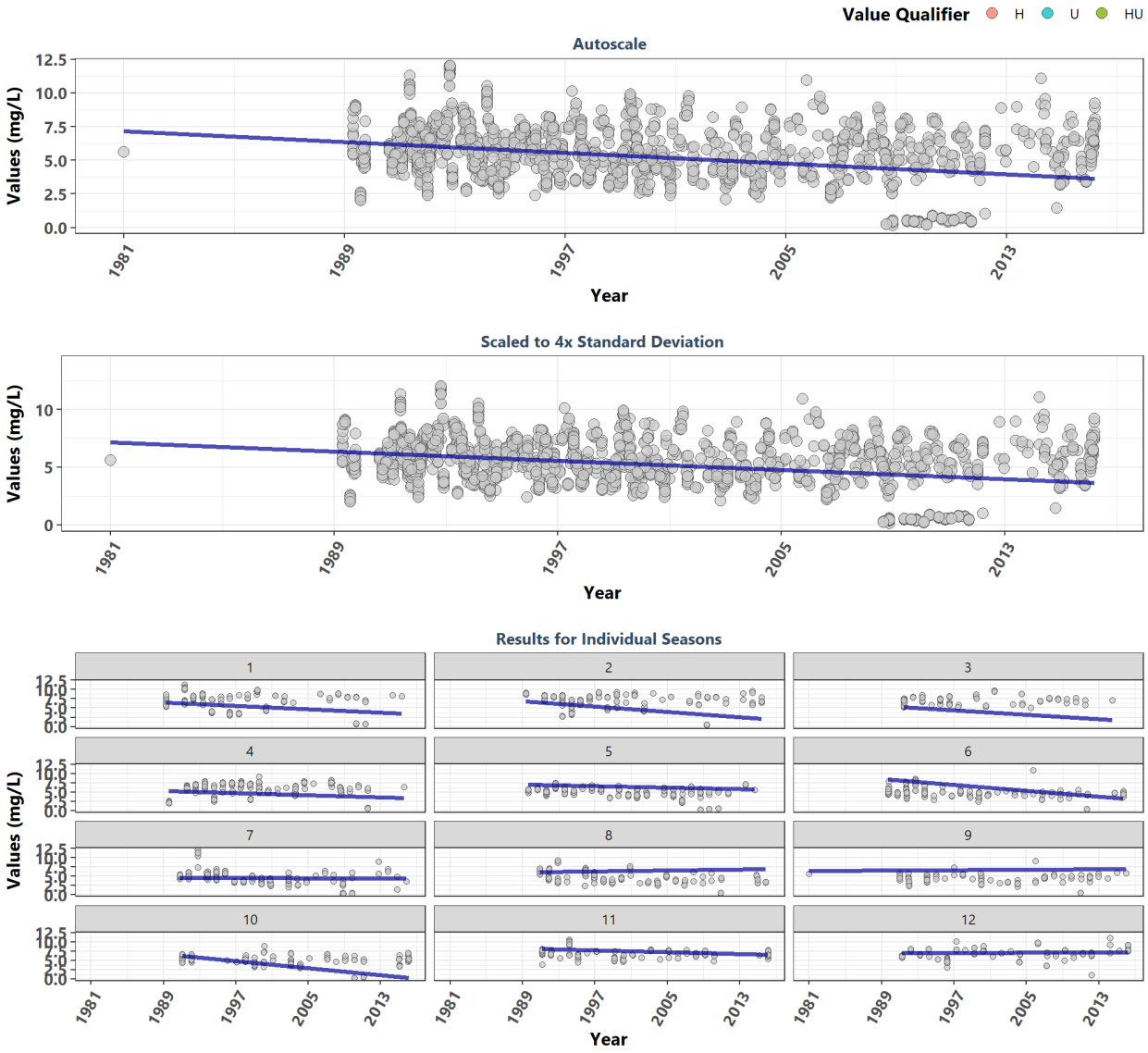
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3978	6.20	0.0827	0.0231	5.0330	7.3	0.0000	121.4	0	1
1	302	7.25	0.0837	0.0171	5.0143	1.1	0.2884	NA	NA	1
2	307	7.10	-0.0282	-0.0150	5.5000	5.0	0.0000	NA	NA	-1
3	336	7.00	0.1911	0.0611	4.5944	7.0	0.0000	NA	NA	1
4	328	6.30	0.0984	0.0247	5.6418	6.7	0.0000	NA	NA	1
5	296	5.98	0.1263	0.0781	1.5169	2.3	0.0228	NA	NA	1
6	323	5.70	0.0408	0.0094	6.8722	2.3	0.0242	NA	NA	1
7	385	4.90	0.0822	0.0225	6.0775	-0.8	0.4058	NA	NA	1
8	319	4.72	0.0884	0.0233	5.0517	3.4	0.0007	NA	NA	1
9	371	4.80	0.2488	0.0507	4.3171	-5.4	0.0000	NA	NA	1
10	373	5.74	0.0545	0.0211	4.8979	1.6	0.1147	NA	NA	1
11	308	6.63	0.2545	0.0500	4.9500	2.6	0.0096	NA	NA	1
12	330	7.00	-0.1869	-0.0825	8.1000	2.2	0.0253	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

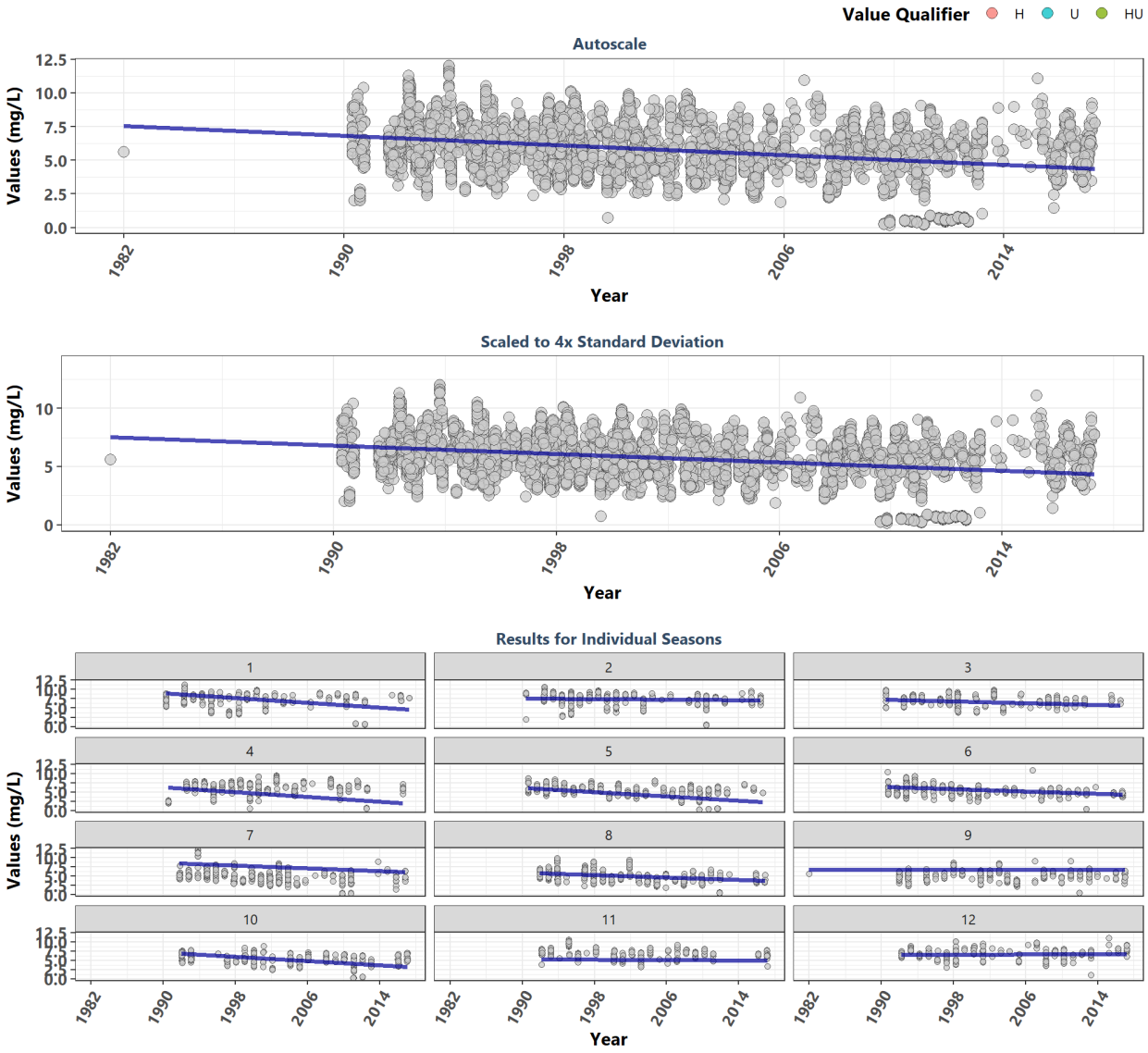
Guana River Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2162	5.50	-0.2345	-0.1000	7.1423	-17.5	0.0000	209.2	0	-1
1	202	6.80	-0.2910	-0.1125	7.3750	-6.0	0.0000	NA	NA	-1
2	192	7.69	-0.4365	-0.1818	8.3364	-3.3	0.0010	NA	NA	-1
3	100	7.03	-0.4153	-0.1500	6.8550	0.5	0.5960	NA	NA	-1
4	222	6.40	-0.2808	-0.0683	5.8567	2.0	0.0481	NA	NA	-1
5	259	4.70	-0.2638	-0.0514	7.4200	-10.5	0.0000	NA	NA	-1
6	177	4.90	-0.2817	-0.2000	10.2000	-5.6	0.0000	NA	NA	-1
7	229	4.50	-0.0259	-0.0050	4.5700	-10.5	0.0000	NA	NA	-1
8	190	4.38	0.0889	0.0333	5.8000	-8.5	0.0000	NA	NA	1
9	125	4.50	0.0613	0.0165	6.3971	-0.4	0.6667	NA	NA	1
10	187	4.90	-0.4656	-0.2333	8.4667	-5.9	0.0000	NA	NA	-1
11	166	6.70	-0.1593	-0.0585	8.6254	-5.1	0.0000	NA	NA	-1
12	113	6.71	0.0358	0.0077	6.9096	1.0	0.3349	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding

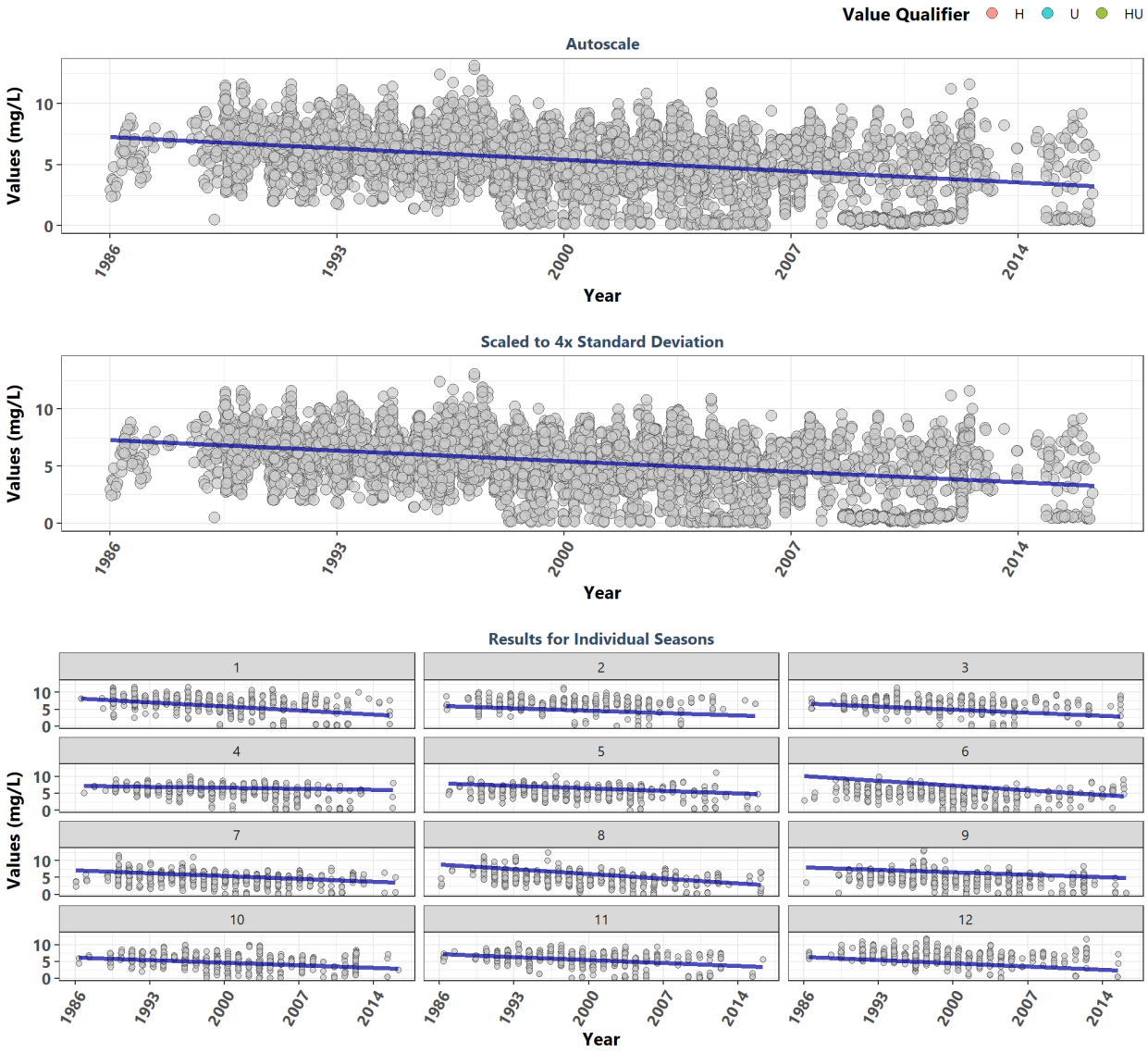
Guana Tolomato Matanzas National Estuarine Research Reserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	5490	5.80	-0.2314	-0.0900	7.5250	-27.3	0.0000	305.8	0	-1
1	482	7.53	-0.2829	-0.1636	10.3118	-9.3	0.0000	NA	NA	-1
2	449	7.86	-0.0574	-0.0182	7.6909	-9.2	0.0000	NA	NA	-1
3	363	7.40	-0.2619	-0.0625	7.8250	-1.6	0.1008	NA	NA	-1
4	477	6.60	-0.4024	-0.1617	7.6700	0.6	0.5437	NA	NA	-1
5	557	5.40	-0.3635	-0.1433	7.3800	-12.4	0.0000	NA	NA	-1
6	435	5.23	-0.2468	-0.0780	7.0160	-8.5	0.0000	NA	NA	-1
7	573	4.80	-0.2890	-0.0933	9.3533	-13.0	0.0000	NA	NA	-1
8	568	4.76	-0.2704	-0.0769	6.5377	-14.4	0.0000	NA	NA	-1
9	387	5.12	0.0113	0.0006	6.7063	-1.4	0.1495	NA	NA	1
10	486	5.30	-0.3521	-0.1417	8.2333	-8.2	0.0000	NA	NA	-1
11	407	6.70	-0.0488	-0.0125	5.3575	-7.9	0.0000	NA	NA	-1
12	306	6.72	0.0186	0.0067	6.4800	0.3	0.7676	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding

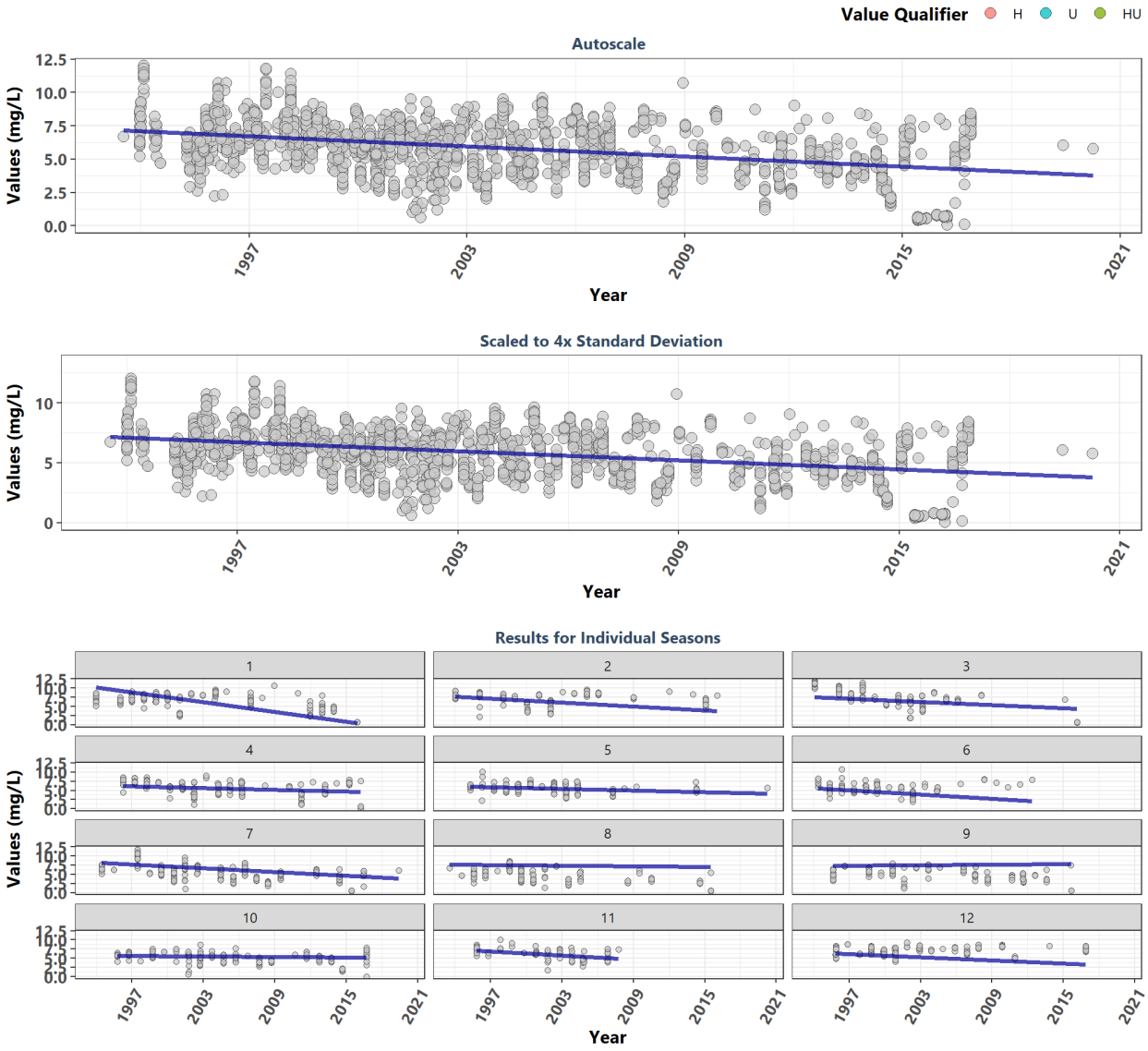
Indian River-Malabar to Vero Beach Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	9367	5.58	-0.2777	-0.1325	7.2800	-40.6	0.0000	152.3	0	-1
1	721	7.08	-0.3193	-0.1714	8.2900	-13.0	0.0000	NA	NA	-1
2	631	6.80	-0.2320	-0.1000	6.0300	-3.7	0.0002	NA	NA	-1
3	591	6.63	-0.3250	-0.1333	6.7467	-8.7	0.0000	NA	NA	-1
4	928	5.80	-0.0978	-0.0425	7.3100	-18.9	0.0000	NA	NA	-1
5	831	5.70	-0.2384	-0.1105	8.0668	-11.1	0.0000	NA	NA	-1
6	722	5.28	-0.3236	-0.2000	10.0800	-13.1	0.0000	NA	NA	-1
7	1016	4.50	-0.2559	-0.1214	7.1571	-14.7	0.0000	NA	NA	-1
8	905	4.80	-0.4143	-0.2000	8.8000	-11.9	0.0000	NA	NA	-1
9	813	4.73	-0.2297	-0.1071	8.0857	-9.9	0.0000	NA	NA	-1
10	838	5.50	-0.2645	-0.1100	6.2300	-11.0	0.0000	NA	NA	-1
11	732	5.89	-0.2530	-0.1250	7.2500	-13.0	0.0000	NA	NA	-1
12	639	6.80	-0.3077	-0.1333	6.3667	-8.7	0.0000	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

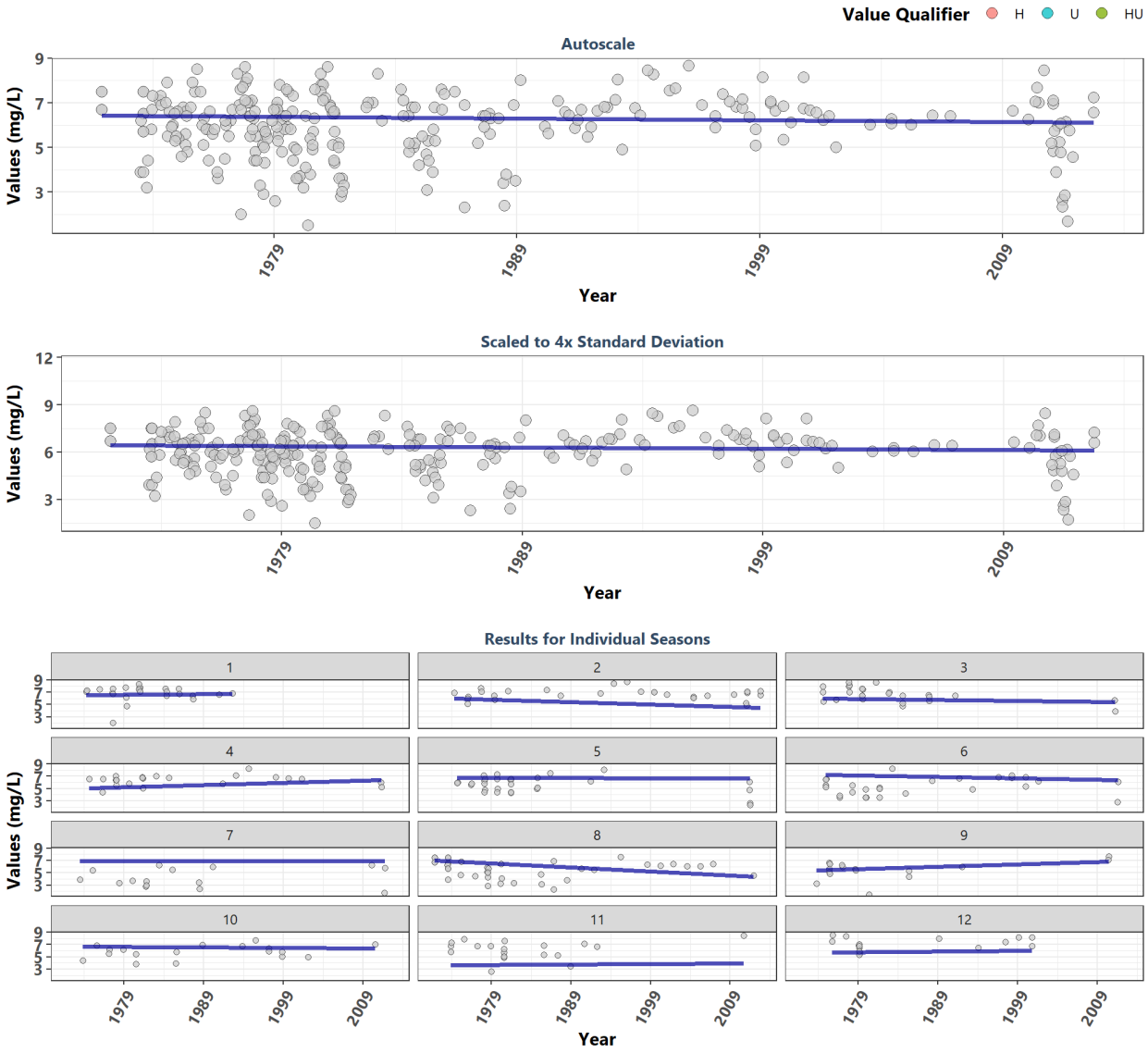
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1973	5.97	-0.2610	-0.1267	7.4700	-17.5	0.0000	126.6	0	-1
1	211	6.80	-0.5429	-0.4450	11.5500	-6.3	0.0000	NA	NA	-1
2	133	7.40	-0.3353	-0.1837	8.2876	-1.1	0.2570	NA	NA	-1
3	137	7.10	-0.2684	-0.1417	7.9171	-9.5	0.0000	NA	NA	-1
4	218	5.90	-0.2154	-0.0768	6.5452	-7.4	0.0000	NA	NA	-1
5	154	5.70	-0.2524	-0.0727	6.3182	-4.0	0.0001	NA	NA	-1
6	112	5.60	-0.3683	-0.2000	6.3000	-0.8	0.4485	NA	NA	-1
7	249	5.40	-0.2894	-0.1667	8.6333	-8.9	0.0000	NA	NA	-1
8	137	4.70	-0.0661	-0.0333	7.7667	-6.4	0.0000	NA	NA	-1
9	133	4.90	0.0986	0.0252	7.1733	-5.7	0.0000	NA	NA	1
10	231	5.30	-0.0483	-0.0250	5.7750	-5.7	0.0000	NA	NA	-1
11	103	6.50	-0.3761	-0.1850	7.9900	-4.1	0.0000	NA	NA	-1
12	155	7.40	-0.3323	-0.1390	6.9850	1.8	0.0671	NA	NA	-1

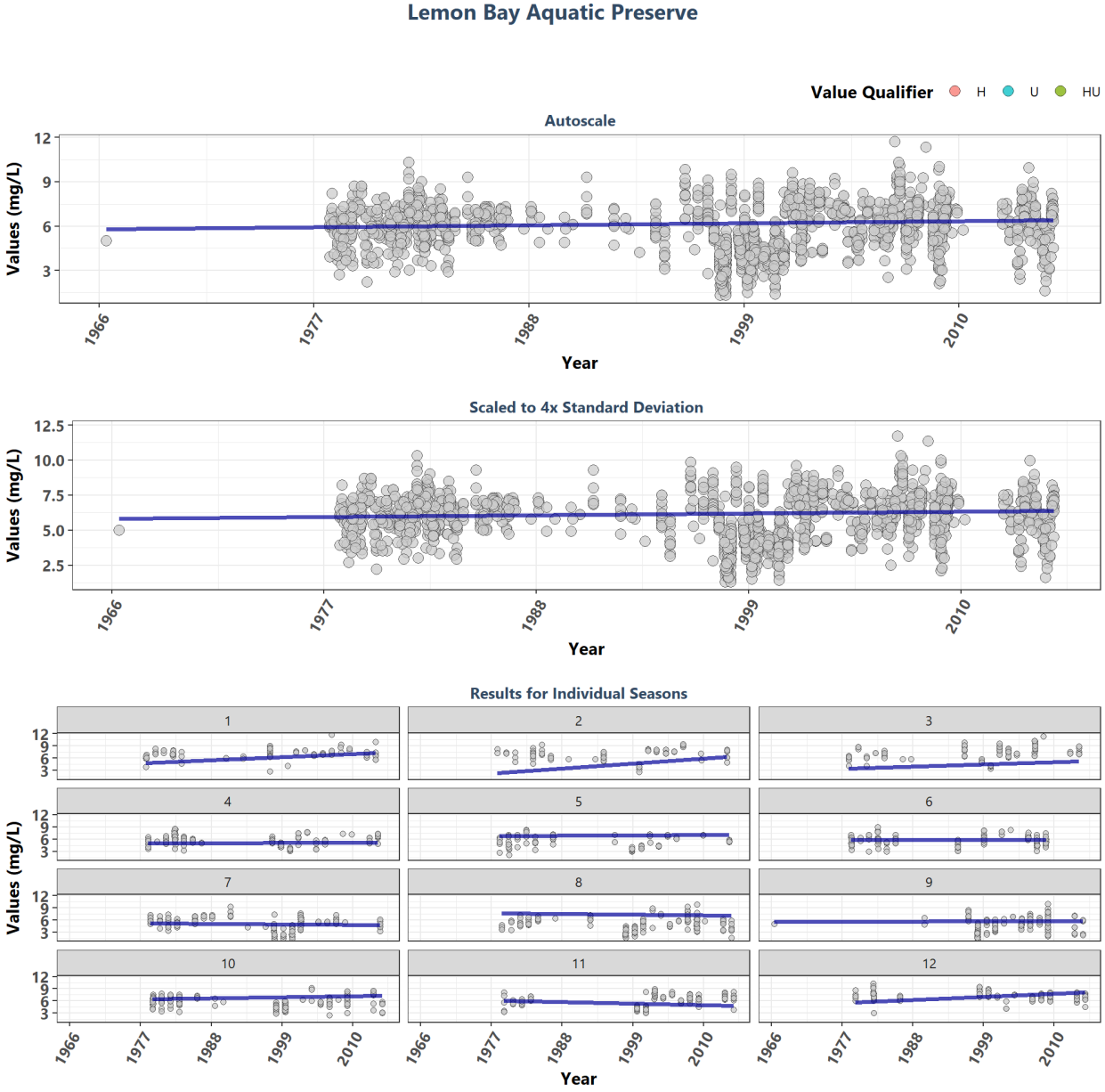
[a] $p < 0.00005$ appear as 0 due to rounding

Jensen Beach to Jupiter Inlet Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	282	6.30	-0.0302	-0.0131	6.7148	-1.2	0.2265	10.7	0.4692	0
1	24	7.10	0.1186	0.0157	6.1600	-0.6	0.5624	NA	NA	0
2	28	6.72	-0.1936	-0.0641	7.4671	-0.1	0.9210	NA	NA	0
3	26	6.42	-0.0598	-0.0264	6.5885	-2.0	0.0471	NA	NA	0
4	23	6.60	0.1538	0.0588	3.6850	0.8	0.4383	NA	NA	0
5	30	5.85	-0.0159	-0.0031	6.8411	-0.5	0.6509	NA	NA	0
6	27	5.33	-0.0870	-0.0388	8.1850	1.1	0.2660	NA	NA	0
7	15	3.70	0.0000	0.0000	6.9000	0.1	0.8812	NA	NA	0
8	40	5.80	-0.2769	-0.1080	9.3360	-1.8	0.0780	NA	NA	0
9	16	5.60	0.1583	0.0627	4.0002	0.8	0.4073	NA	NA	0
10	18	5.84	-0.0526	-0.0154	7.0451	0.5	0.5943	NA	NA	0
11	19	6.63	0.0381	0.0158	3.2428	-0.3	0.7761	NA	NA	0
12	16	6.90	0.0980	0.0210	5.1936	0.0	1.0000	NA	NA	0

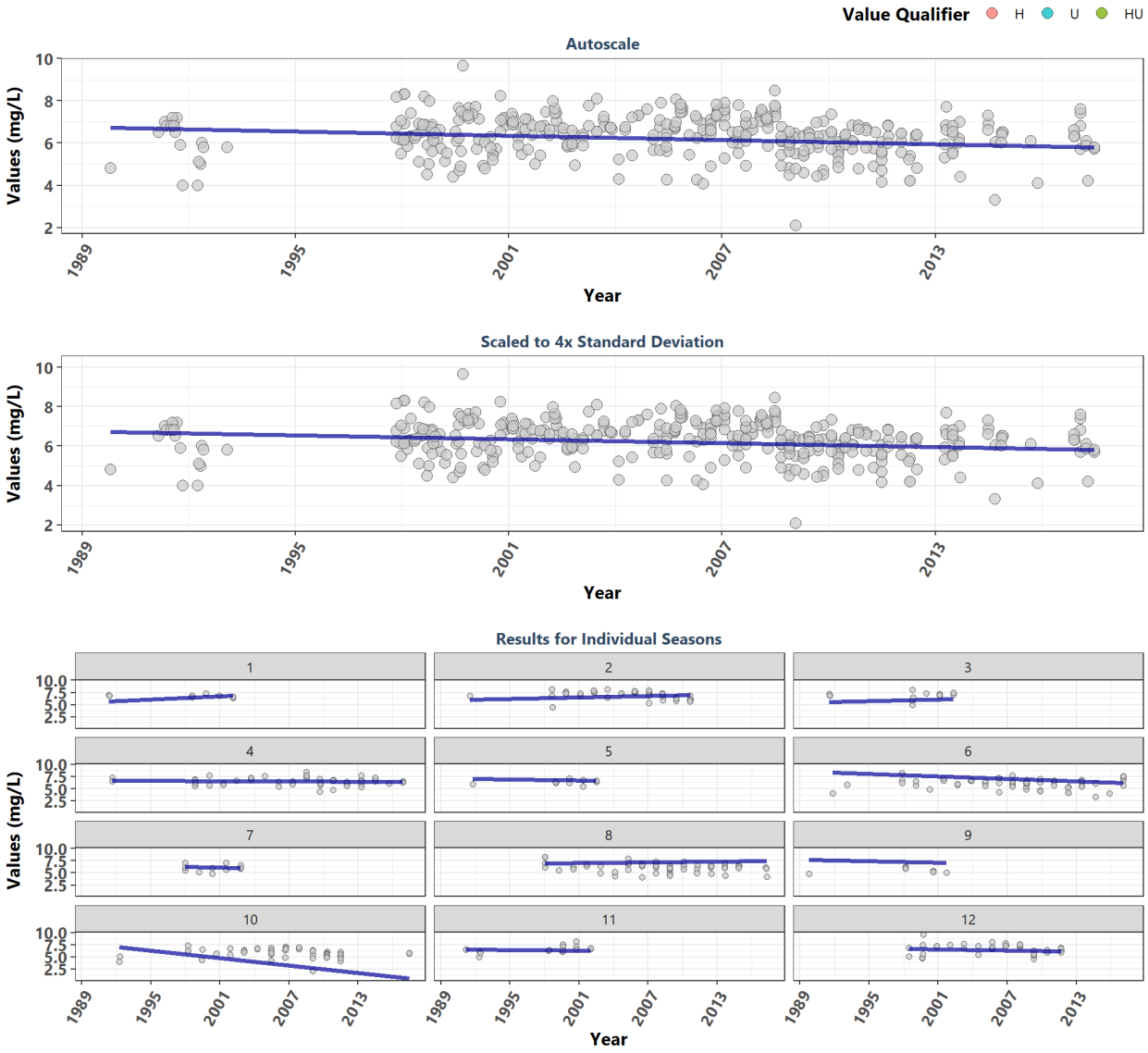
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1825	5.89	0.0569	0.0162	5.5614	3.3	0.0009	72.9	0	1
1	88	6.97	0.2589	0.0937	2.6589	1.1	0.2507	NA	NA	1
2	92	6.95	0.1805	0.1500	-1.2000	1.5	0.1398	NA	NA	1
3	142	7.08	0.1407	0.0650	1.9650	4.2	0.0000	NA	NA	1
4	185	5.40	0.0240	0.0071	4.8214	-3.2	0.0013	NA	NA	1
5	95	5.10	0.0831	0.0117	6.5239	0.3	0.7316	NA	NA	1
6	134	5.80	0.0263	0.0014	5.7447	0.5	0.6517	NA	NA	1
7	195	5.00	-0.0414	-0.0154	5.6000	-0.9	0.3870	NA	NA	-1
8	177	4.50	-0.1406	-0.0250	8.3000	2.8	0.0051	NA	NA	-1
9	188	4.80	0.0161	0.0045	5.5227	3.7	0.0002	NA	NA	1
10	204	5.70	0.1042	0.0284	5.8139	0.3	0.7312	NA	NA	1
11	172	6.50	-0.1572	-0.0439	7.1126	5.1	0.0000	NA	NA	-1
12	153	7.20	0.2340	0.0933	3.3517	-2.6	0.0093	NA	NA	1

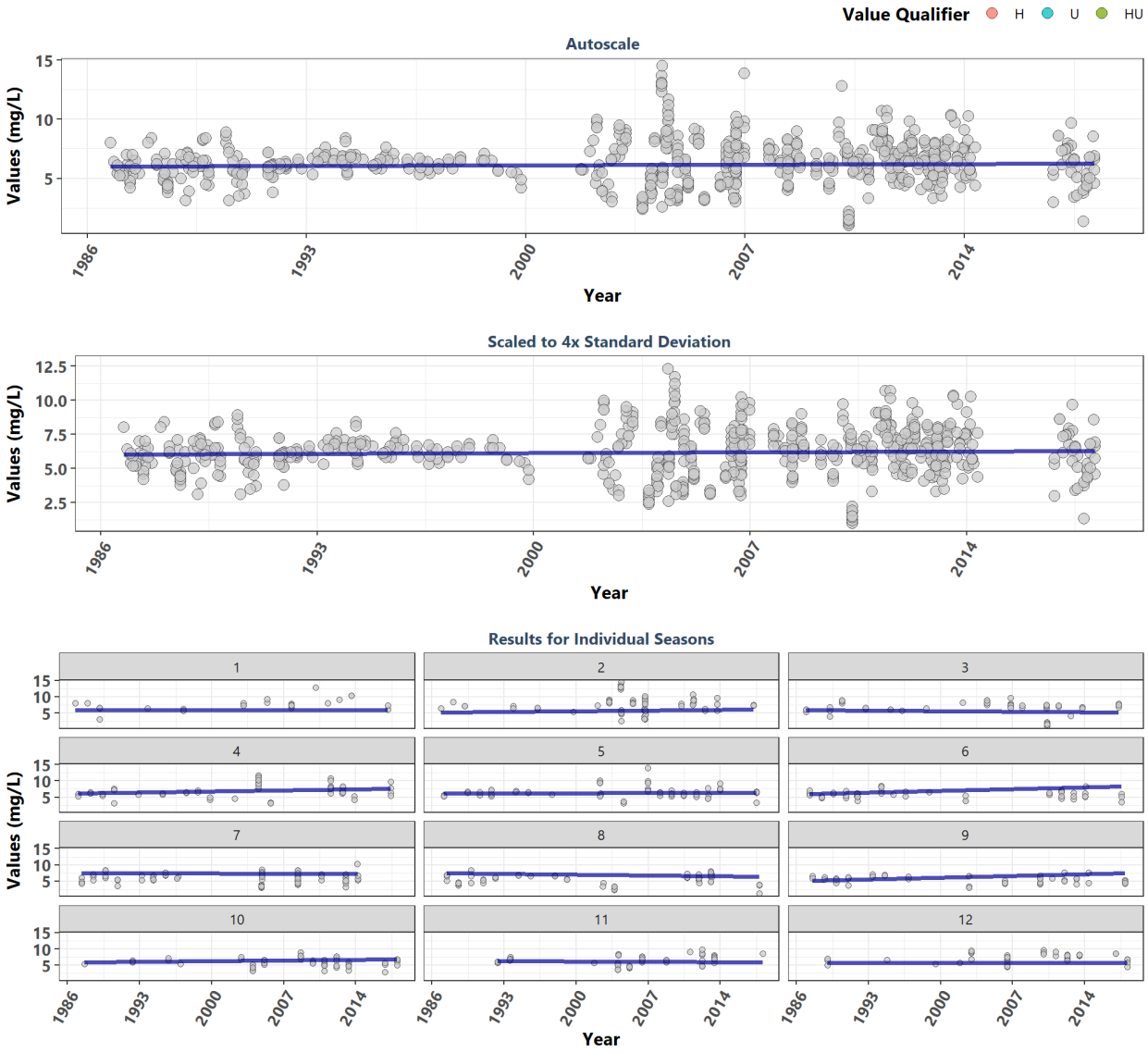
[a] $p < 0.00005$ appear as 0 due to rounding

Loxahatchee River-Lake Worth Creek Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	370	6.44	-0.0983	-0.0400	6.9511	-3.3	0.0009	22.1	0.0236	-1
1	9	6.80	0.4381	0.1367	4.7233	-0.9	0.3892	NA	NA	1
2	47	7.16	0.1667	0.0589	5.5656	-2.9	0.0033	NA	NA	1
3	14	7.00	0.0545	0.0633	5.0633	1.0	0.3343	NA	NA	1
4	50	6.55	-0.0555	-0.0110	6.7865	-0.6	0.5737	NA	NA	-1
5	9	6.39	-0.2500	-0.0367	7.2767	0.5	0.5940	NA	NA	-1
6	57	6.22	-0.2960	-0.1033	9.1233	-0.9	0.3621	NA	NA	-1
7	11	5.95	-0.1804	-0.0671	7.1157	0.2	0.8726	NA	NA	-1
8	57	6.29	0.1978	0.0262	6.6456	-1.5	0.1318	NA	NA	1
9	8	5.42	-0.1163	-0.0647	8.0697	-0.9	0.3580	NA	NA	-1
10	50	5.84	-0.2857	-0.3067	9.4117	-1.9	0.0641	NA	NA	-1
11	15	6.50	-0.0833	-0.0239	6.7214	2.3	0.0226	NA	NA	-1
12	43	6.84	-0.1372	-0.0414	7.1600	-1.1	0.2732	NA	NA	-1

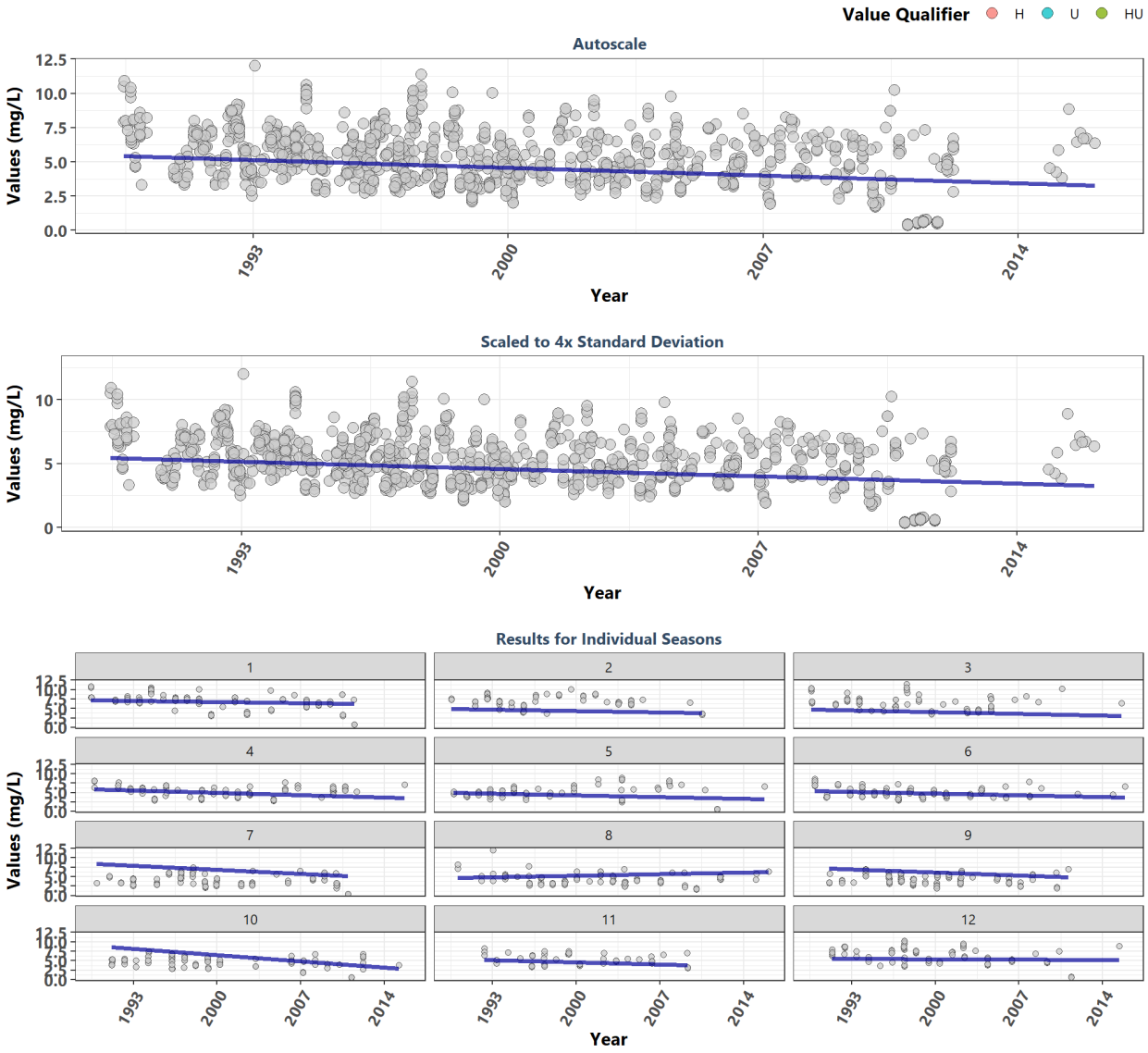
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	720	6.24	0.0398	0.0095	5.9525	1.2	0.2334	14.3	0.2147	0
1	27	7.40	0.0068	0.0000	5.9750	2.1	0.0331	NA	NA	0
2	85	7.41	0.1298	0.0368	4.9632	-0.1	0.8950	NA	NA	0
3	54	6.76	-0.1267	-0.0242	6.0317	-0.7	0.4759	NA	NA	0
4	56	6.45	0.1173	0.0500	5.9300	1.2	0.2239	NA	NA	0
5	70	6.30	0.0480	0.0128	6.0049	0.6	0.5584	NA	NA	0
6	52	5.97	0.2906	0.0892	5.4387	0.1	0.9493	NA	NA	0
7	87	5.80	-0.0098	-0.0060	7.5480	0.0	1.0000	NA	NA	0
8	65	5.70	-0.0664	-0.0400	7.7050	1.5	0.1245	NA	NA	0
9	52	5.50	0.1935	0.0833	4.6850	-1.3	0.1844	NA	NA	0
10	67	6.00	0.1117	0.0381	5.6500	-0.6	0.5707	NA	NA	0
11	56	6.44	-0.0475	-0.0154	6.4003	2.1	0.0342	NA	NA	0
12	49	7.23	-0.0003	0.0000	5.8000	1.2	0.2291	NA	NA	0

[a] $p < 0.00005$ appear as 0 due to rounding

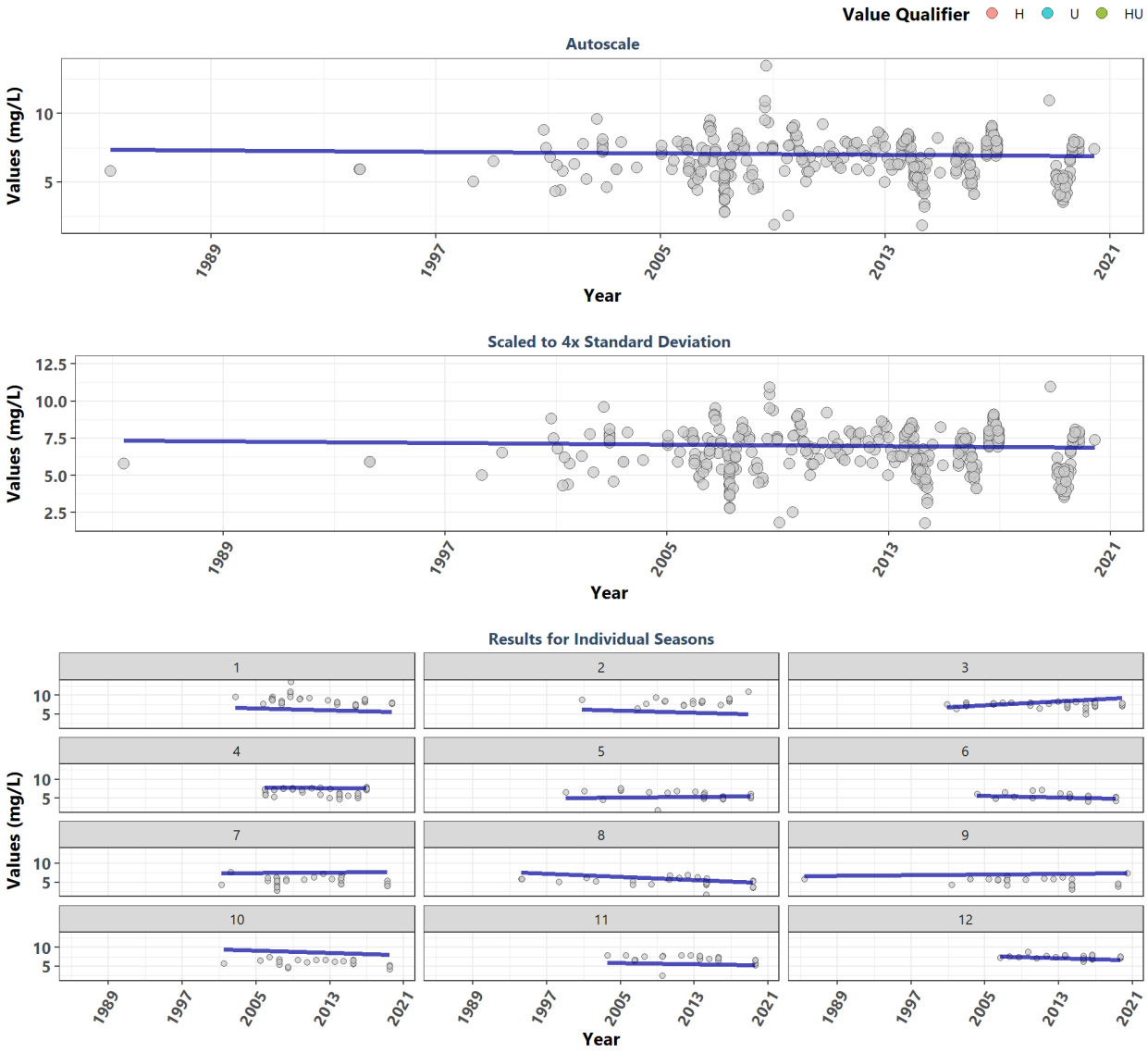
Mosquito Lagoon Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1093	5.20	-0.1676	-0.0800	5.6938	-8.5	0.0000	39.7	0	-1
1	115	7.00	-0.0835	-0.0449	7.3487	-5.4	0.0000	NA	NA	-1
2	69	6.90	-0.1330	-0.0500	5.0150	-1.0	0.3096	NA	NA	-1
3	77	6.42	-0.1653	-0.0667	4.9667	-2.3	0.0199	NA	NA	-1
4	98	5.40	-0.1867	-0.0889	6.2444	-0.7	0.4882	NA	NA	-1
5	84	5.05	-0.1495	-0.0667	5.2000	1.7	0.0885	NA	NA	-1
6	96	4.83	-0.1915	-0.0643	5.6357	-3.1	0.0019	NA	NA	-1
7	95	4.10	-0.3414	-0.1500	8.9500	-2.4	0.0173	NA	NA	-1
8	88	4.22	0.1262	0.0600	4.4200	-1.8	0.0663	NA	NA	1
9	106	4.40	-0.1805	-0.1154	7.8046	-2.3	0.0226	NA	NA	-1
10	98	4.80	-0.3429	-0.2333	9.7600	-2.8	0.0051	NA	NA	-1
11	62	5.00	-0.2151	-0.0825	5.7375	-2.2	0.0313	NA	NA	-1
12	105	6.96	-0.0475	-0.0167	5.6500	-5.2	0.0000	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

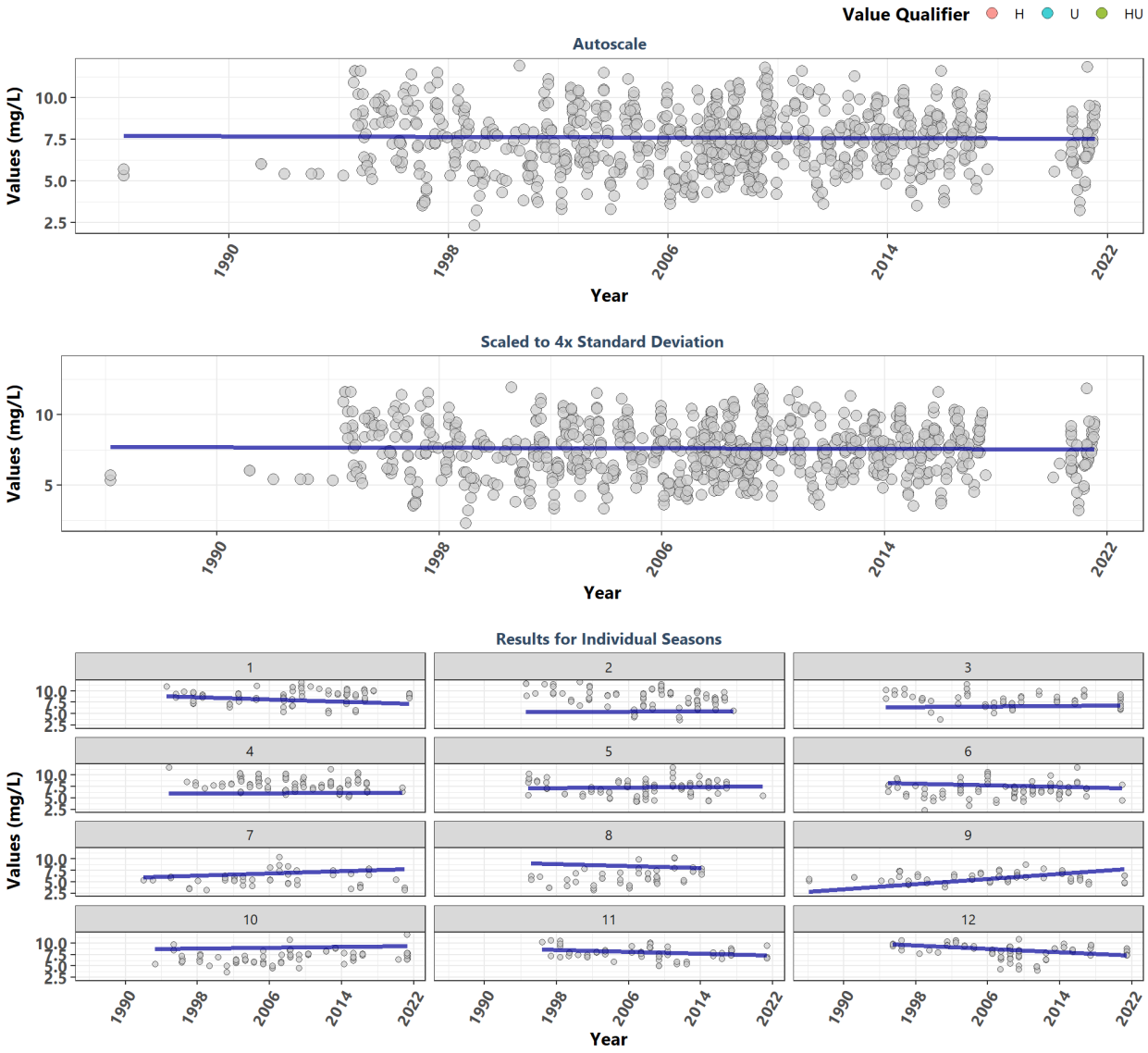
Nassau River-St. Johns River Marshes Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	372	6.89	-0.0816	-0.0133	7.3963	-1.5	0.1283	29.8	0.0017	0
1	41	8.50	-0.2381	-0.0657	8.1536	-2.6	0.0093	NA	NA	0
2	25	8.48	-0.2000	-0.0692	7.5425	2.8	0.0051	NA	NA	0
3	53	7.60	0.3933	0.1300	4.1900	0.7	0.4860	NA	NA	0
4	42	7.20	-0.1452	-0.0267	8.5200	0.9	0.3807	NA	NA	0
5	26	5.54	0.0756	0.0225	4.6150	-2.5	0.0137	NA	NA	0
6	21	5.26	-0.2253	-0.0589	7.0556	-1.3	0.2096	NA	NA	0
7	39	5.20	0.0646	0.0100	7.2500	0.7	0.4722	NA	NA	0
8	23	5.23	-0.3415	-0.1011	8.8817	-1.5	0.1335	NA	NA	0
9	24	5.64	0.0941	0.0200	6.5450	-1.1	0.2922	NA	NA	0
10	22	6.22	-0.2805	-0.0720	10.8760	-1.5	0.1245	NA	NA	0
11	24	7.15	-0.1558	-0.0418	6.8512	-2.0	0.0423	NA	NA	0
12	32	7.60	-0.2971	-0.0804	9.7283	-1.2	0.2334	NA	NA	0

[a] $p < 0.00005$ appear as 0 due to rounding

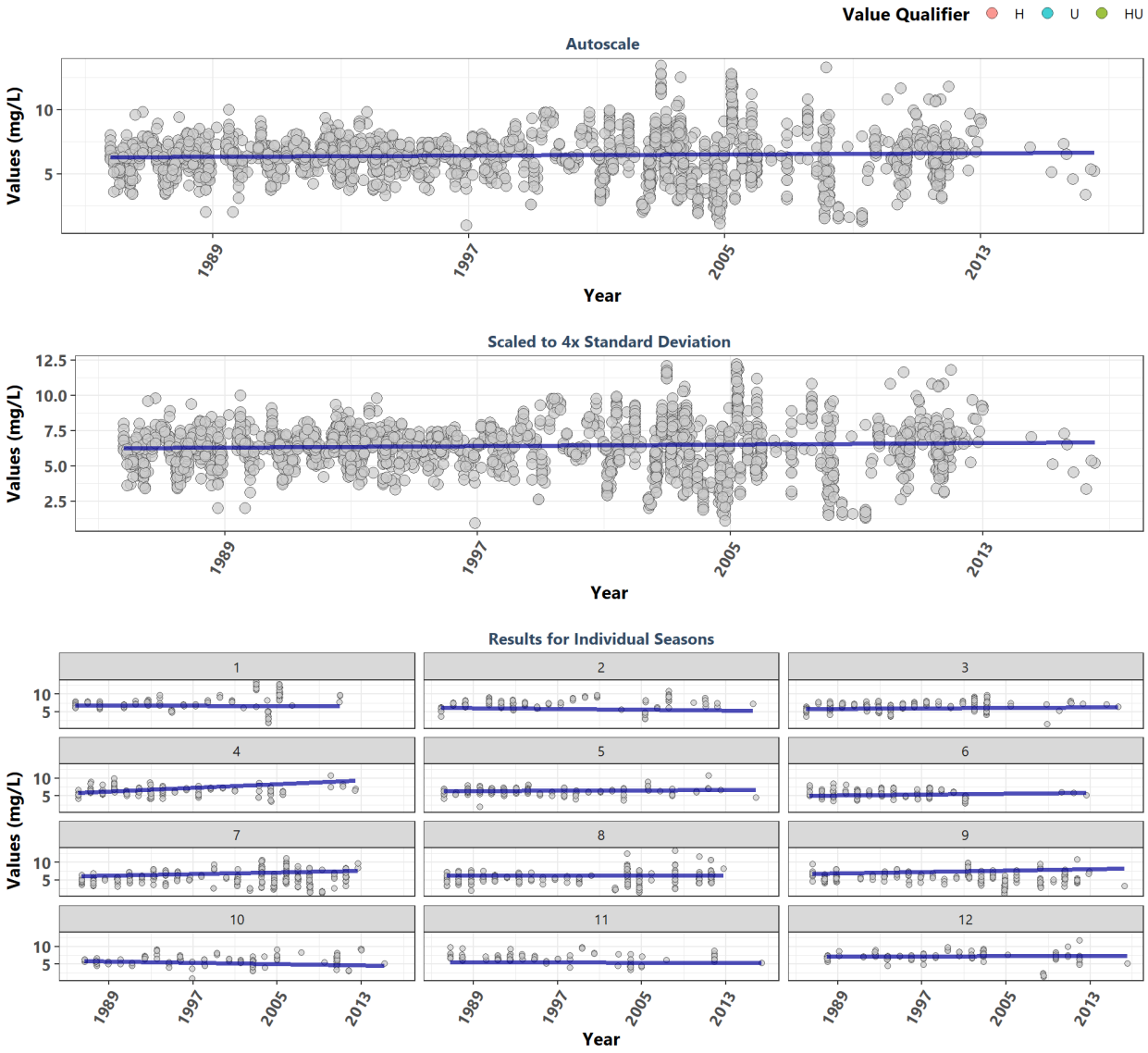
Nature Coast Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	850	7.40	-0.0120	-0.0050	7.7146	-0.7	0.4580	45	0	0
1	75	9.10	-0.1458	-0.0600	9.5600	0.9	0.3444	NA	NA	0
2	83	8.50	0.0485	0.0083	5.2929	-2.6	0.0102	NA	NA	0
3	62	8.00	0.0467	0.0167	6.2000	-1.9	0.0635	NA	NA	0
4	91	8.00	0.0317	0.0055	5.8624	-2.1	0.0402	NA	NA	0
5	86	7.30	0.0315	0.0133	6.9533	0.4	0.6698	NA	NA	0
6	90	6.65	-0.1541	-0.0429	8.8143	0.7	0.5157	NA	NA	0
7	45	5.50	0.2519	0.0600	5.4000	0.5	0.6446	NA	NA	0
8	48	5.80	-0.1861	-0.0533	9.7467	3.1	0.0019	NA	NA	0
9	61	6.00	0.3094	0.1415	2.2613	0.4	0.7222	NA	NA	0
10	79	6.90	0.0746	0.0223	8.4759	3.3	0.0010	NA	NA	0
11	57	7.70	-0.1613	-0.0500	9.3450	-1.7	0.0906	NA	NA	0
12	73	8.36	-0.1916	-0.0923	10.9923	-2.3	0.0196	NA	NA	0

[a] $p < 0.00005$ appear as 0 due to rounding

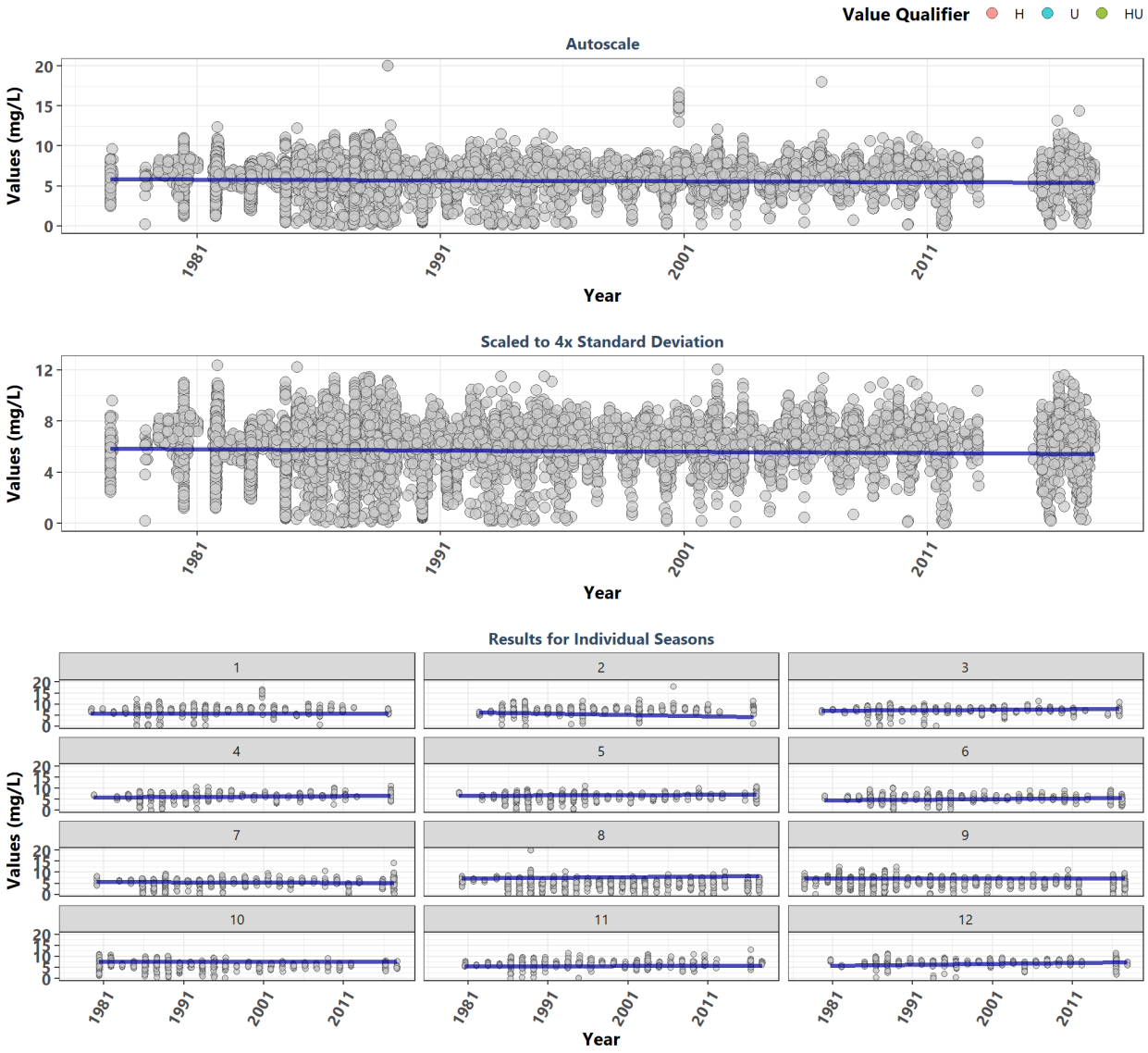
Pine Island Sound Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2345	6.30	0.0499	0.0143	6.1452	3.1	0.0022	69.8	0	1
1	174	7.64	-0.0620	-0.0118	6.9235	5.7	0.0000	NA	NA	-1
2	136	7.07	-0.1134	-0.0333	6.4100	3.1	0.0016	NA	NA	-1
3	244	6.65	0.0587	0.0154	5.7308	4.1	0.0000	NA	NA	1
4	190	6.20	0.2862	0.1500	4.4950	0.4	0.7136	NA	NA	1
5	162	6.40	0.0562	0.0111	6.2222	1.1	0.2869	NA	NA	1
6	150	5.50	0.1197	0.0333	4.7667	-2.1	0.0366	NA	NA	1
7	331	5.37	0.1761	0.0647	5.3882	-0.5	0.6193	NA	NA	1
8	250	5.60	0.0179	0.0000	6.2000	2.8	0.0046	NA	NA	-1
9	251	5.61	0.1816	0.0615	6.0904	-2.7	0.0073	NA	NA	1
10	173	6.10	-0.1144	-0.0487	6.2786	1.2	0.2474	NA	NA	-1
11	142	6.70	-0.0183	-0.0067	5.5367	-1.1	0.2689	NA	NA	-1
12	142	7.20	0.0522	0.0105	6.9579	0.9	0.3535	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding

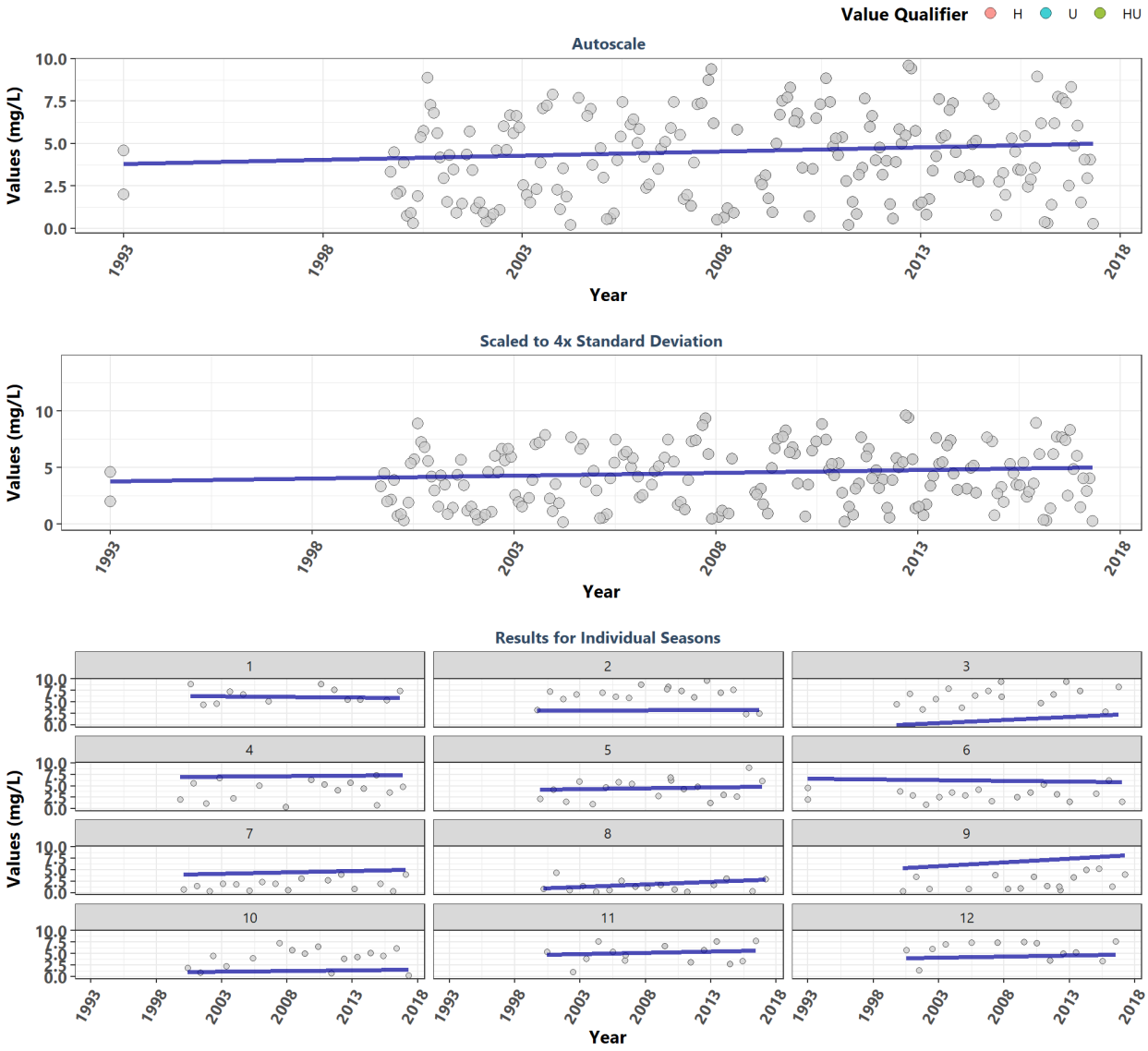
Pinellas County Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	16244	5.89	-0.0039	-0.0147	6.1325	-4.7	0.0000	488.4	0	-1
1	693	7.70	0.0344	0.0075	5.4675	-0.5	0.6492	NA	NA	1
2	641	7.50	-0.2774	-0.0927	8.4917	6.3	0.0000	NA	NA	-1
3	607	7.20	0.1278	0.0300	6.2800	-0.2	0.8451	NA	NA	1
4	949	6.13	0.0858	0.0271	5.2729	7.0	0.0000	NA	NA	1
5	968	6.06	0.0642	0.0162	6.3431	4.0	0.0001	NA	NA	1
6	727	5.70	0.0937	0.0393	3.5793	1.4	0.1638	NA	NA	1
7	635	5.52	-0.0368	-0.0200	6.0050	1.0	0.3319	NA	NA	-1
8	1847	4.72	0.1647	0.0400	6.2600	6.1	0.0000	NA	NA	1
9	6216	5.53	-0.0053	-0.0008	7.2231	-4.4	0.0000	NA	NA	-1
10	1736	6.18	-0.0115	-0.0033	7.8013	-17.5	0.0000	NA	NA	-1
11	660	6.86	0.0256	0.0075	5.2875	2.5	0.0134	NA	NA	1
12	565	7.27	0.1481	0.0588	4.4241	4.6	0.0000	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding

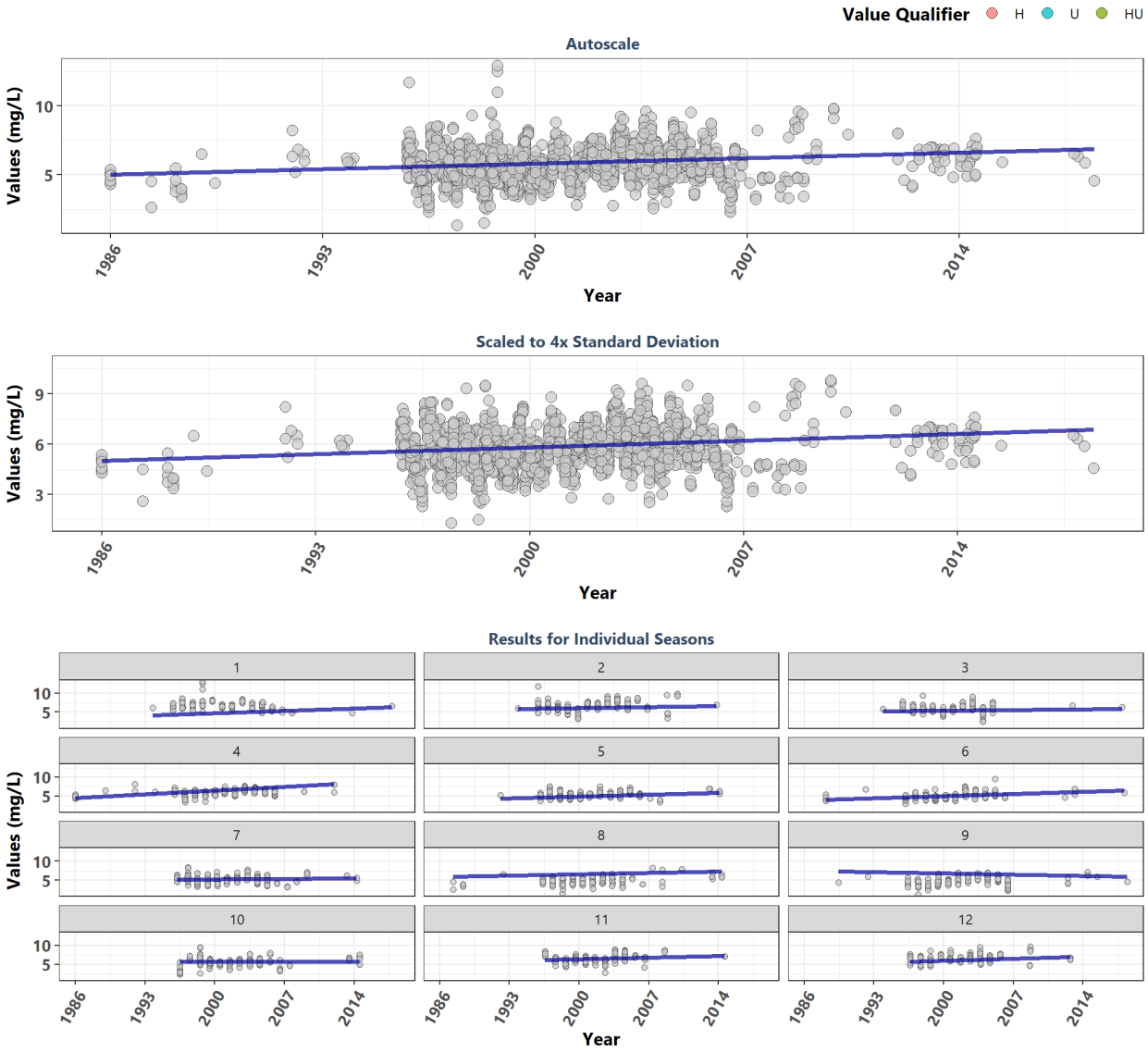
Rocky Bayou State Park Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	260	4.25	0.0997	0.0500	3.7882	2.4	0.0185	9.5	0.5715	1
1	16	6.05	-0.1000	-0.0200	6.4200	-0.5	0.6176	NA	NA	-1
2	24	7.16	0.0198	0.0075	3.0500	0.1	0.9205	NA	NA	1
3	22	6.63	0.3768	0.1282	-0.8279	1.6	0.1117	NA	NA	1
4	22	4.66	0.0181	0.0160	6.8880	0.3	0.7552	NA	NA	1
5	24	4.51	0.0652	0.0380	3.8640	0.4	0.6715	NA	NA	1
6	23	3.17	-0.0760	-0.0323	6.5692	0.1	0.9155	NA	NA	-1
7	21	1.95	0.0519	0.0580	3.6210	1.8	0.0734	NA	NA	1
8	23	1.30	0.2857	0.1138	0.1300	0.6	0.5416	NA	NA	1
9	24	1.48	0.2468	0.1609	4.2164	2.6	0.0101	NA	NA	2
10	23	4.52	0.0949	0.0339	0.7582	0.3	0.7301	NA	NA	1
11	19	5.38	0.0760	0.0600	4.3000	0.4	0.6726	NA	NA	1
12	19	6.02	0.0553	0.0475	3.7125	-0.4	0.6724	NA	NA	1

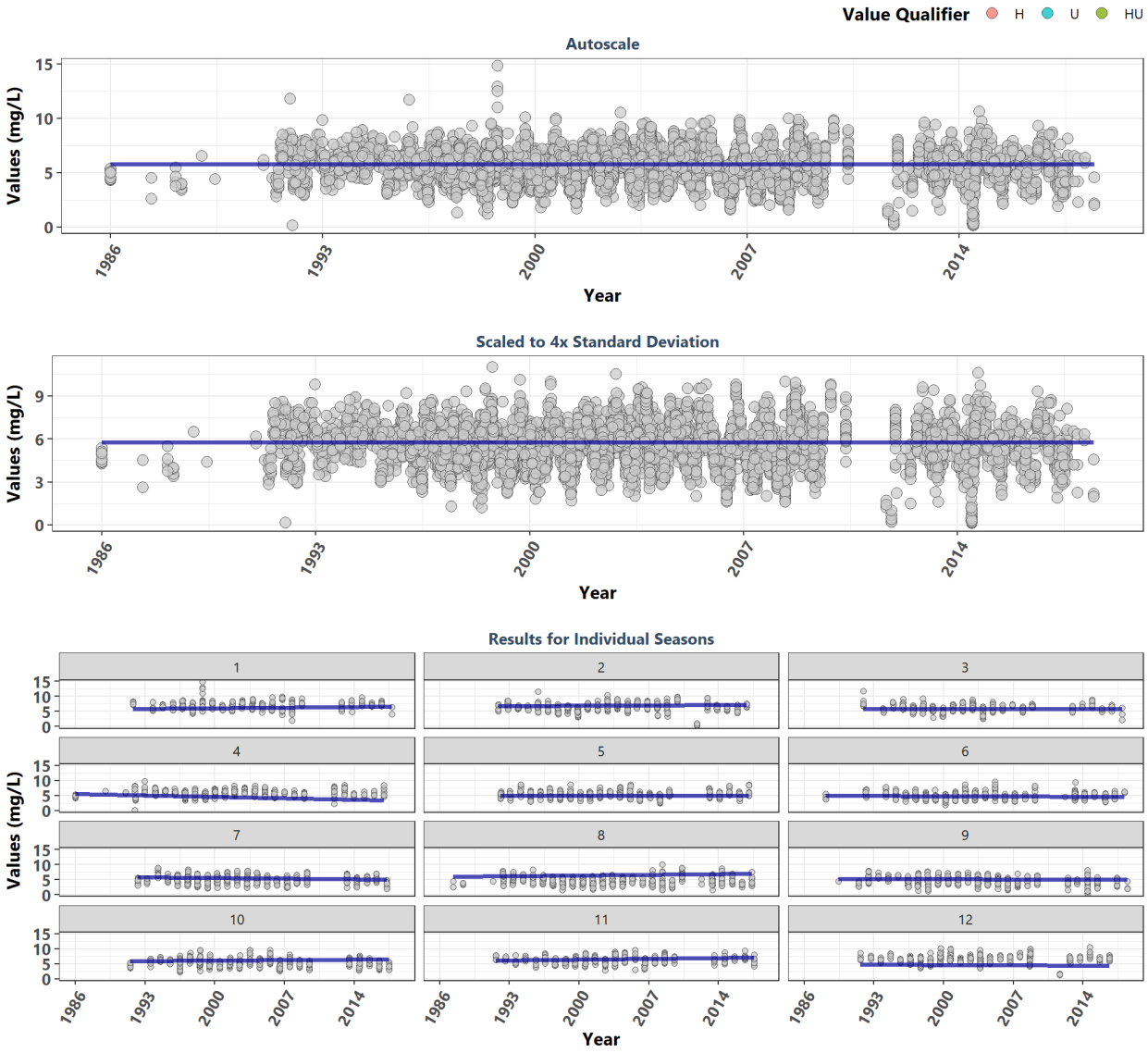
[a] $p < 0.00005$ appear as 0 due to rounding

Rookery Bay Aquatic Preserve



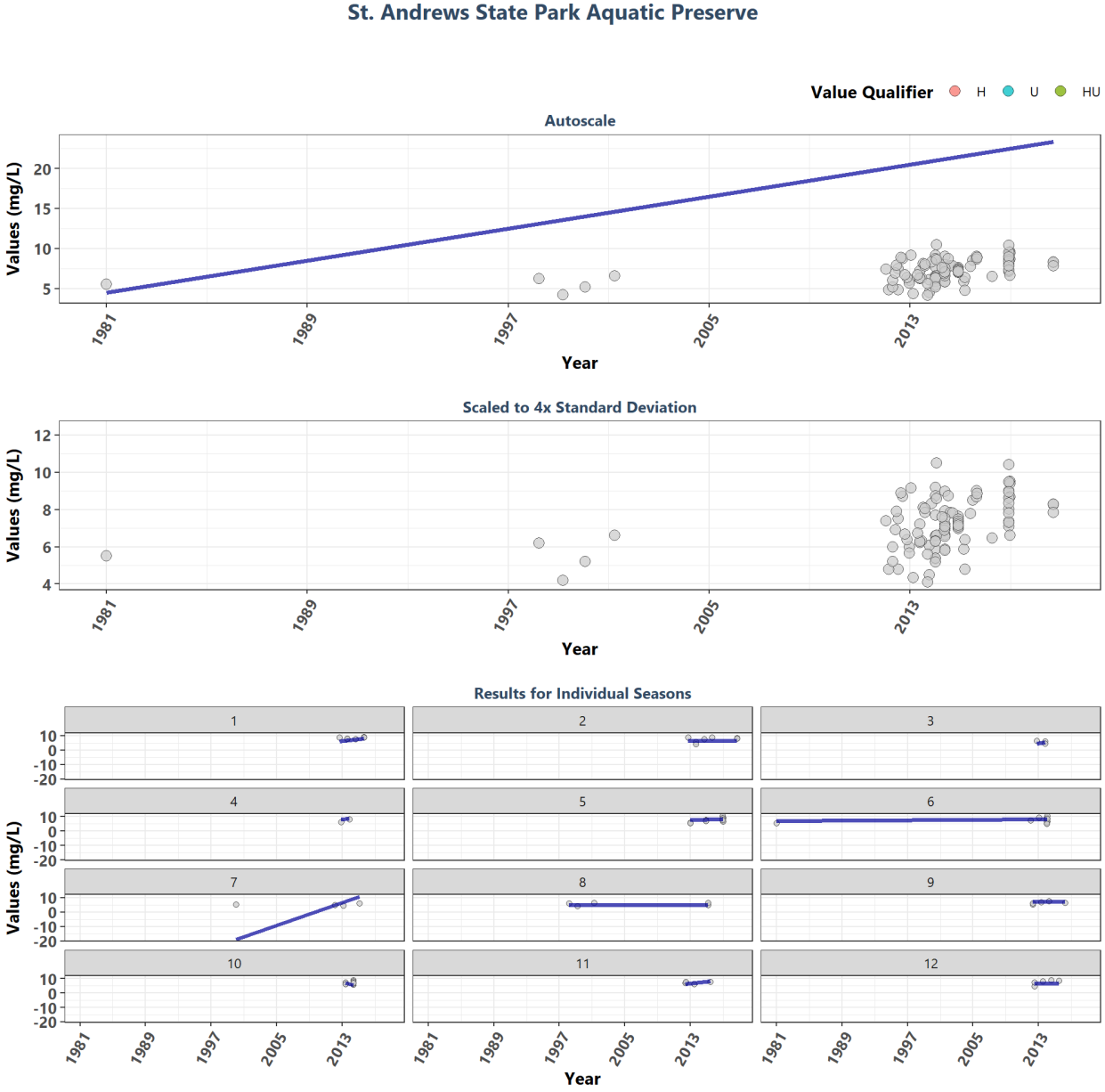
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1643	5.80	0.1326	0.0575	5.0026	8.3	0.0000	46.7	0	1
1	122	6.76	0.2287	0.0875	3.5425	-1.6	0.1046	NA	NA	1
2	142	6.65	0.0842	0.0400	5.5000	4.2	0.0000	NA	NA	1
3	131	6.10	0.0836	0.0288	4.9677	1.4	0.1520	NA	NA	1
4	156	6.00	0.2354	0.1444	4.4833	5.2	0.0000	NA	NA	1
5	142	5.40	0.2229	0.0748	3.8532	1.5	0.1386	NA	NA	1
6	143	5.10	0.2423	0.0825	3.8625	4.3	0.0000	NA	NA	1
7	145	5.30	0.0349	0.0175	5.0375	0.6	0.5331	NA	NA	1
8	138	4.90	0.0949	0.0500	5.9350	3.9	0.0001	NA	NA	1
9	142	4.86	-0.0990	-0.0450	7.4400	4.1	0.0001	NA	NA	-1
10	131	5.64	0.0105	0.0025	5.6025	0.2	0.8604	NA	NA	1
11	127	6.40	0.1077	0.0600	5.5000	1.8	0.0712	NA	NA	1
12	124	6.69	0.2814	0.0774	4.8772	1.6	0.1164	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6248	5.60	-0.0011	0.0000	5.7760	-0.6	0.5397	122.3	0	0
1	457	6.90	0.1113	0.0280	5.7520	1.6	0.1156	NA	NA	0
2	530	6.44	0.0492	0.0150	6.6450	4.4	0.0000	NA	NA	0
3	441	6.12	-0.0104	0.0000	5.8000	2.0	0.0451	NA	NA	0
4	547	5.80	-0.1904	-0.0667	5.5333	-0.4	0.7148	NA	NA	0
5	556	5.10	0.0040	0.0000	4.9000	-0.8	0.4247	NA	NA	0
6	511	4.90	-0.0579	-0.0167	5.0833	0.1	0.8935	NA	NA	0
7	549	4.80	-0.1032	-0.0328	6.0079	-2.0	0.0423	NA	NA	0
8	573	4.50	0.1230	0.0357	5.9286	-2.1	0.0395	NA	NA	0
9	533	4.40	-0.0226	-0.0050	5.1850	-6.6	0.0000	NA	NA	0
10	565	5.45	0.0637	0.0200	5.8200	-3.7	0.0002	NA	NA	0
11	503	6.20	0.1279	0.0364	5.8218	3.7	0.0002	NA	NA	0
12	483	6.50	-0.0574	-0.0167	4.7833	4.0	0.0001	NA	NA	0

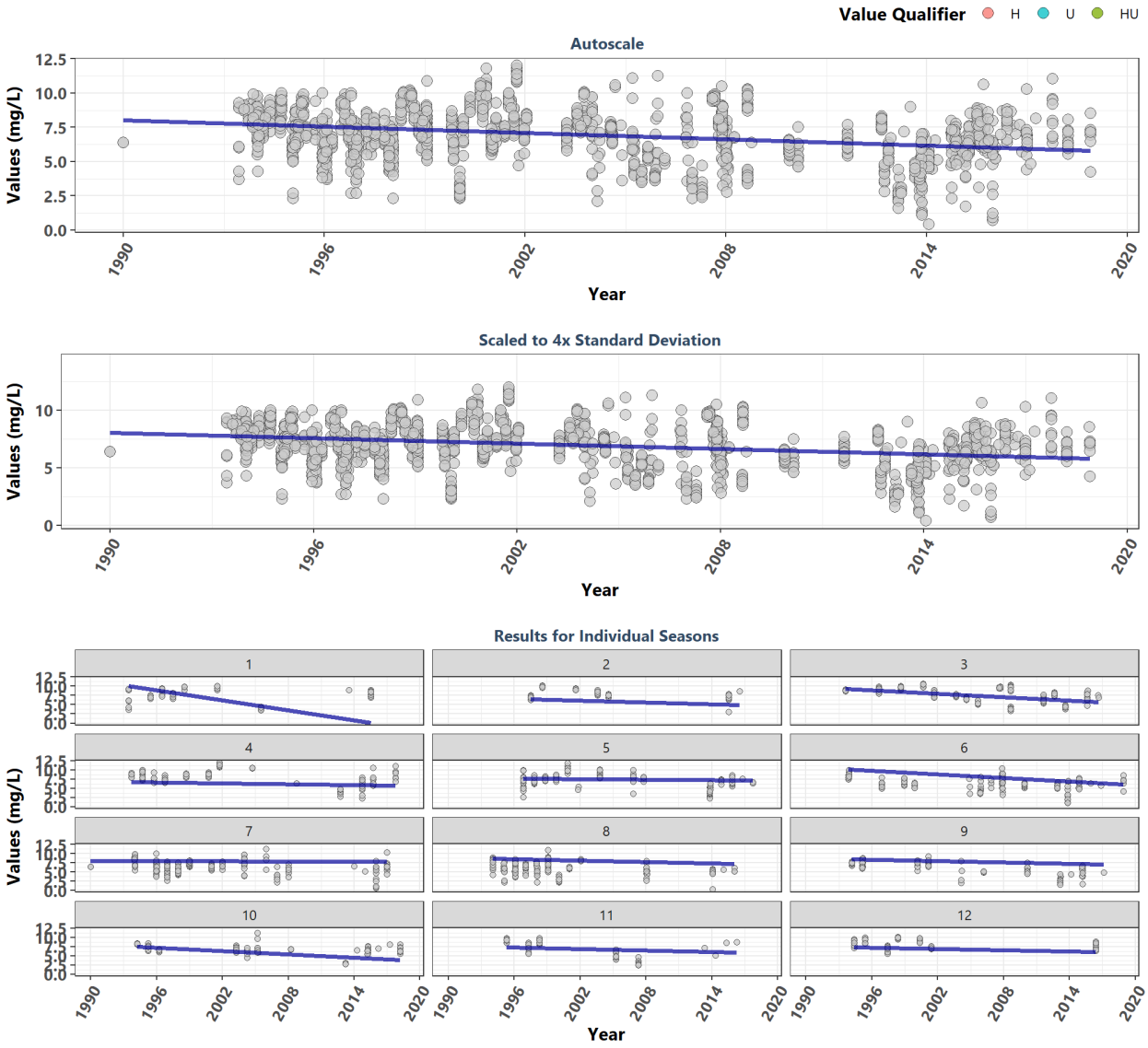
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	101	7.16	0.2094	0.5000	4.4683	3.6	0.0003	15.6	0.1557	1
1	6	8.36	0.0956	0.5900	-12.4600	0.4	0.6967	NA	NA	1
2	8	8.05	0.1000	0.0112	5.9765	0.6	0.5209	NA	NA	1
3	3	NA	0.4246	0.6750	-16.8850	NA	NA	NA	NA	NA
4	2	NA	0.7000	0.6150	-11.5800	NA	NA	NA	NA	NA
5	26	7.42	0.2143	0.1138	4.1256	3.5	0.0005	NA	NA	1
6	16	6.58	0.2000	0.0400	7.0150	-0.1	0.9468	NA	NA	1
7	4	5.00	1.0000	1.9500	-56.0300	0.0	1.0000	NA	NA	2
8	5	6.20	0.0000	0.0060	4.8110	0.0	1.0000	NA	NA	1
9	5	6.47	-0.1667	-0.0433	8.7000	1.0	0.3122	NA	NA	-1
10	17	NA	-0.6667	-1.4000	52.2800	NA	NA	NA	NA	NA
11	4	7.34	0.5000	0.7200	-16.5700	0.0	1.0000	NA	NA	1
12	5	8.10	-0.0167	-0.0036	6.7050	1.5	0.1296	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

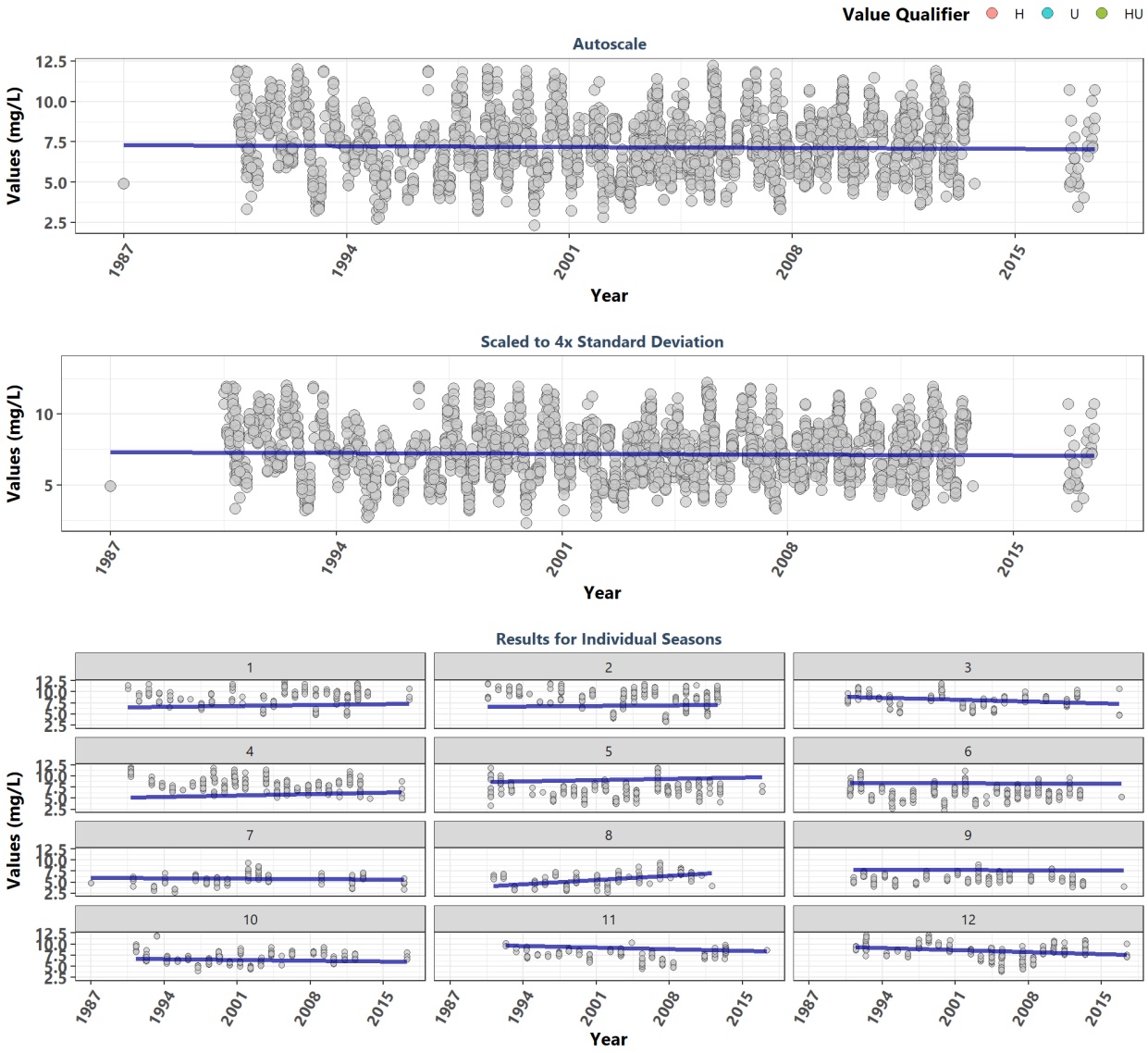
St. Joseph Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1782	7.10	-0.2080	-0.0767	8.0250	-11.4	0.0000	50.8	0	-1
1	85	7.90	-0.5545	-0.4500	11.6000	-0.2	0.8180	NA	NA	-1
2	104	7.70	-0.1632	-0.0800	7.0400	-4.8	0.0000	NA	NA	-1
3	214	7.80	-0.3097	-0.1586	9.9201	-6.3	0.0000	NA	NA	-1
4	146	8.20	-0.1012	-0.0400	6.8600	-2.9	0.0043	NA	NA	-1
5	174	7.45	-0.0726	-0.0250	7.8000	-1.4	0.1526	NA	NA	-1
6	218	6.47	-0.2883	-0.1583	10.6500	-4.3	0.0000	NA	NA	-1
7	235	6.50	-0.0171	-0.0056	7.9500	-2.3	0.0203	NA	NA	-1
8	239	6.40	-0.1907	-0.0619	8.8952	-3.8	0.0002	NA	NA	-1
9	106	6.45	-0.1585	-0.0619	8.7571	-7.0	0.0000	NA	NA	-1
10	107	6.70	-0.4544	-0.1500	8.1000	-4.4	0.0000	NA	NA	-1
11	71	8.00	-0.1953	-0.0666	7.6735	-7.0	0.0000	NA	NA	-1
12	83	8.40	-0.2873	-0.0550	7.4700	-2.6	0.0099	NA	NA	-1

[a] $p < 0.00005$ appear as 0 due to rounding

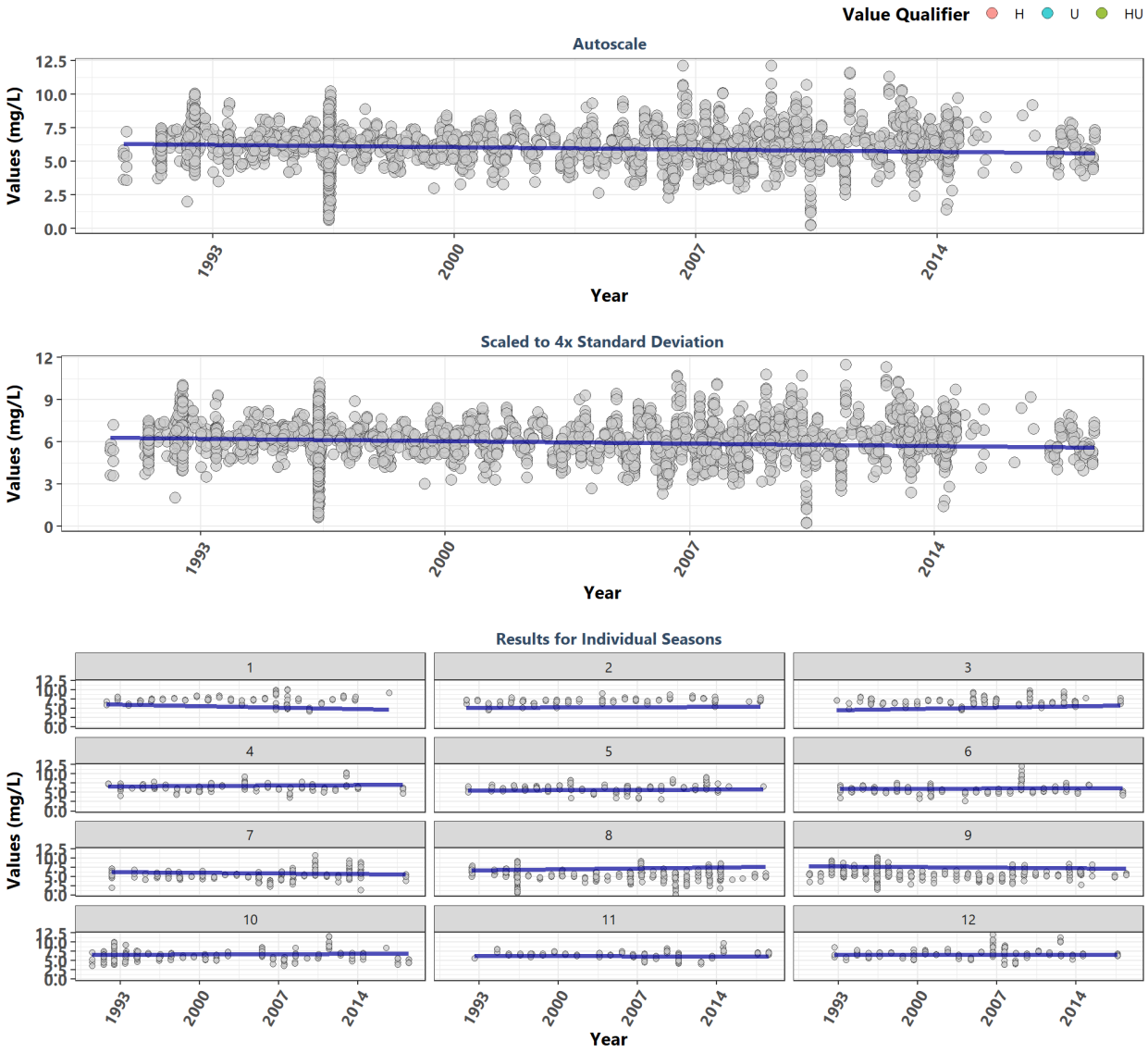
St. Martins Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	2667	7.20	-0.0074	-0.0083	7.3099	-1.9	0.0518	113.2	0	0
1	230	9.30	0.0840	0.0282	6.4047	2.3	0.0214	NA	NA	0
2	296	8.90	0.0355	0.0143	6.6571	-3.2	0.0014	NA	NA	0
3	165	8.40	-0.1593	-0.0600	9.0500	-0.3	0.7601	NA	NA	0
4	274	8.00	0.1857	0.0444	5.0333	-3.9	0.0001	NA	NA	0
5	282	6.90	0.1017	0.0400	8.5000	0.9	0.3738	NA	NA	0
6	303	6.40	-0.0160	-0.0048	8.4810	-1.9	0.0589	NA	NA	0
7	145	5.70	-0.0725	-0.0167	6.0667	3.3	0.0009	NA	NA	0
8	155	5.70	0.3778	0.1375	3.6375	7.0	0.0000	NA	NA	0
9	186	5.80	-0.0202	-0.0053	7.8447	-1.5	0.1412	NA	NA	0
10	229	6.80	-0.0726	-0.0250	6.7750	1.9	0.0578	NA	NA	0
11	154	7.75	-0.1239	-0.0500	9.9000	-0.4	0.7100	NA	NA	0
12	248	8.40	-0.1459	-0.0625	9.5250	-3.4	0.0006	NA	NA	0

[a] $p < 0.00005$ appear as 0 due to rounding

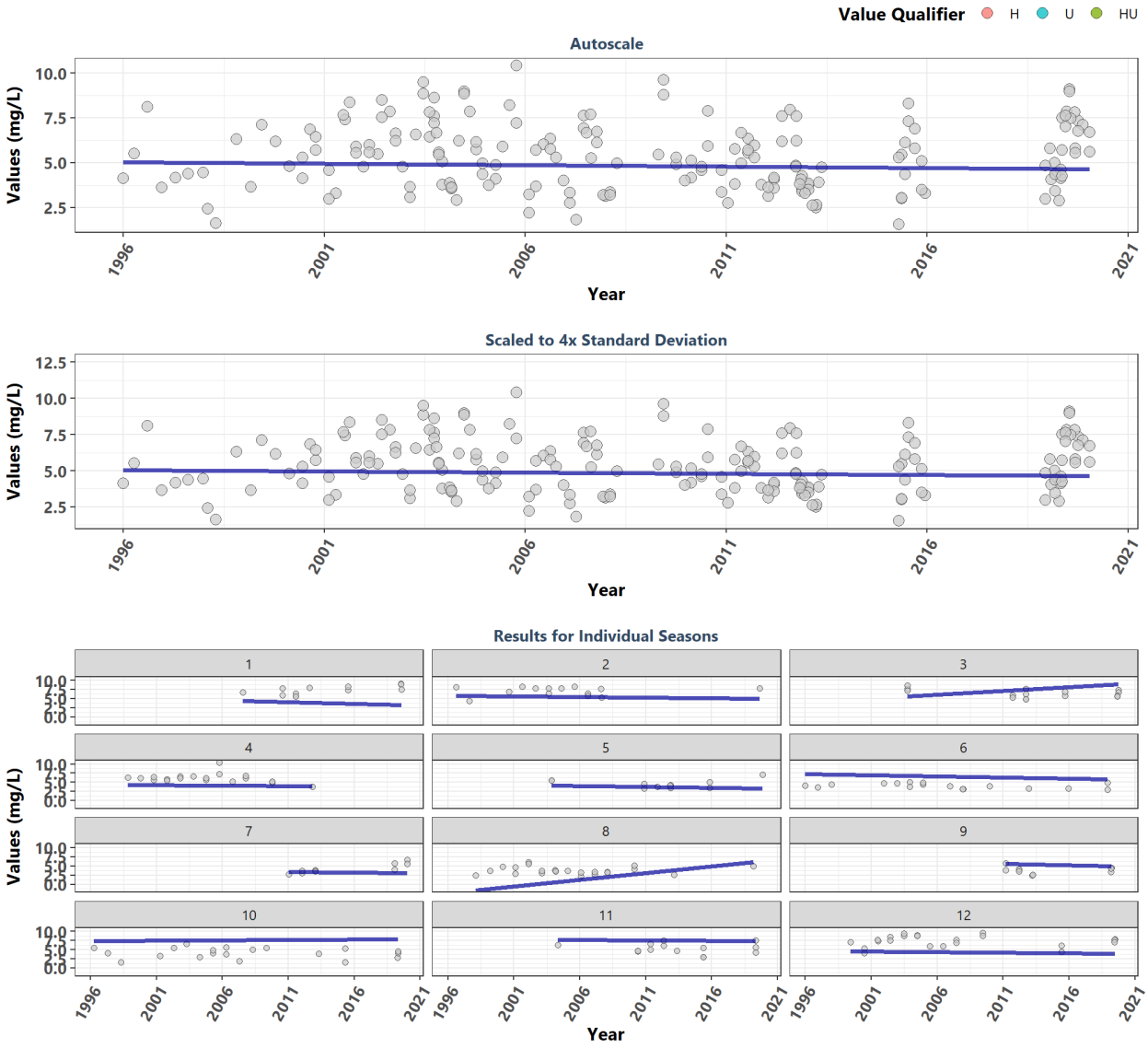
Terra Ceia Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3227	6.10	-0.0296	-0.0250	6.4287	-6.1	0.0000	101.9	0	-1
1	121	7.35	-0.1770	-0.0550	6.4500	-1.2	0.2385	NA	NA	-1
2	113	7.10	0.0245	0.0106	5.1417	4.4	0.0000	NA	NA	1
3	154	6.80	0.1477	0.0468	4.2765	0.9	0.3438	NA	NA	1
4	150	6.23	0.0449	0.0127	6.6155	-1.1	0.2541	NA	NA	1
5	139	6.06	0.0540	0.0100	5.5100	0.7	0.4607	NA	NA	1
6	139	5.70	0.0422	0.0092	5.8846	0.9	0.3446	NA	NA	1
7	215	5.26	-0.0721	-0.0210	6.2822	3.2	0.0012	NA	NA	-1
8	528	5.30	0.2807	0.0375	6.4250	0.9	0.3851	NA	NA	1
9	797	5.90	-0.0722	-0.0225	7.8225	-7.6	0.0000	NA	NA	-1
10	594	6.14	0.0514	0.0100	6.6000	-2.7	0.0075	NA	NA	1
11	123	6.62	-0.0627	-0.0112	6.4325	0.3	0.8008	NA	NA	-1
12	154	6.87	0.0155	0.0020	6.5780	0.8	0.4078	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding

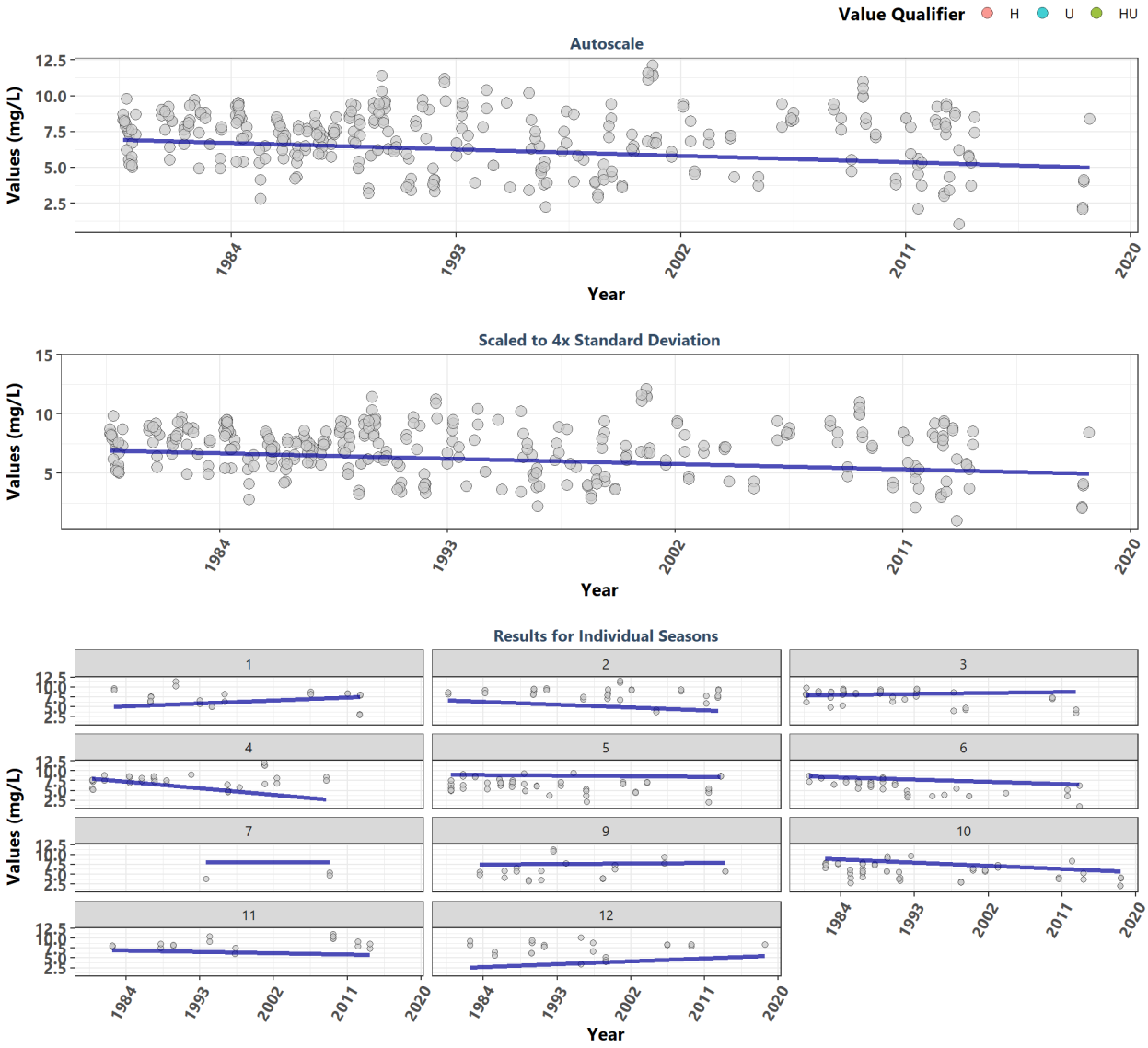
Tomoka Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	189	5.28	-0.0286	-0.0159	5.0400	-0.9	0.3527	22	0.0247	0
1	12	7.38	-0.1111	-0.0900	5.4150	2.0	0.0420	NA	NA	0
2	14	7.76	-0.0879	-0.0347	5.7770	-0.8	0.4081	NA	NA	0
3	15	6.23	0.4545	0.2012	4.0644	-1.0	0.3077	NA	NA	0
4	19	6.15	-0.1158	-0.0209	4.3537	-1.1	0.2606	NA	NA	0
5	12	4.15	-0.2794	-0.0489	4.4501	-0.1	0.8878	NA	NA	0
6	17	4.01	-0.2000	-0.0565	7.1898	-1.5	0.1256	NA	NA	0
7	10	3.79	-0.0652	-0.0341	3.9029	3.1	0.0018	NA	NA	0
8	24	3.63	0.7778	0.3600	-2.3300	-0.4	0.6717	NA	NA	0
9	10	3.93	-0.1930	-0.0787	6.8588	-0.4	0.7099	NA	NA	0
10	20	4.14	0.0909	0.0234	7.3807	-0.7	0.4941	NA	NA	0
11	14	5.20	-0.1758	-0.0300	8.0150	-0.4	0.6971	NA	NA	0
12	22	7.58	-0.0455	-0.0300	4.6650	0.6	0.5708	NA	NA	0

[a] $p < 0.00005$ appear as 0 due to rounding

Yellow River Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	328	7.30	-0.1419	-0.0750	8.2750	-4.4	0.0000	36.4	0.0001	-1
1	22	7.80	0.2238	0.1208	2.6583	-1.1	0.2542	NA	NA	1
2	33	8.70	-0.2850	-0.1182	8.7864	-1.1	0.2735	NA	NA	-1
3	42	8.30	0.1307	0.0400	7.2400	-1.9	0.0628	NA	NA	1
4	33	7.50	-0.5646	-0.2739	12.9727	0.5	0.6280	NA	NA	-1
5	51	6.70	-0.1345	-0.0270	9.5361	-1.3	0.2059	NA	NA	-1
6	37	6.40	-0.1986	-0.0889	10.2111	-4.9	0.0000	NA	NA	-1
7	3	NA	-0.0303	-0.0063	8.2750	NA	NA	NA	NA	NA
9	21	5.80	0.0606	0.0250	6.9250	1.4	0.1623	NA	NA	1
10	46	5.95	-0.1775	-0.1333	11.5333	-2.8	0.0052	NA	NA	-1
11	18	8.40	-0.1224	-0.0500	7.9000	0.7	0.4661	NA	NA	-1
12	22	8.10	0.6667	0.1200	0.2600	-0.2	0.8645	NA	NA	1

[a] $p < 0.00005$ appear as 0 due to rounding

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 `SufficientData` 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:

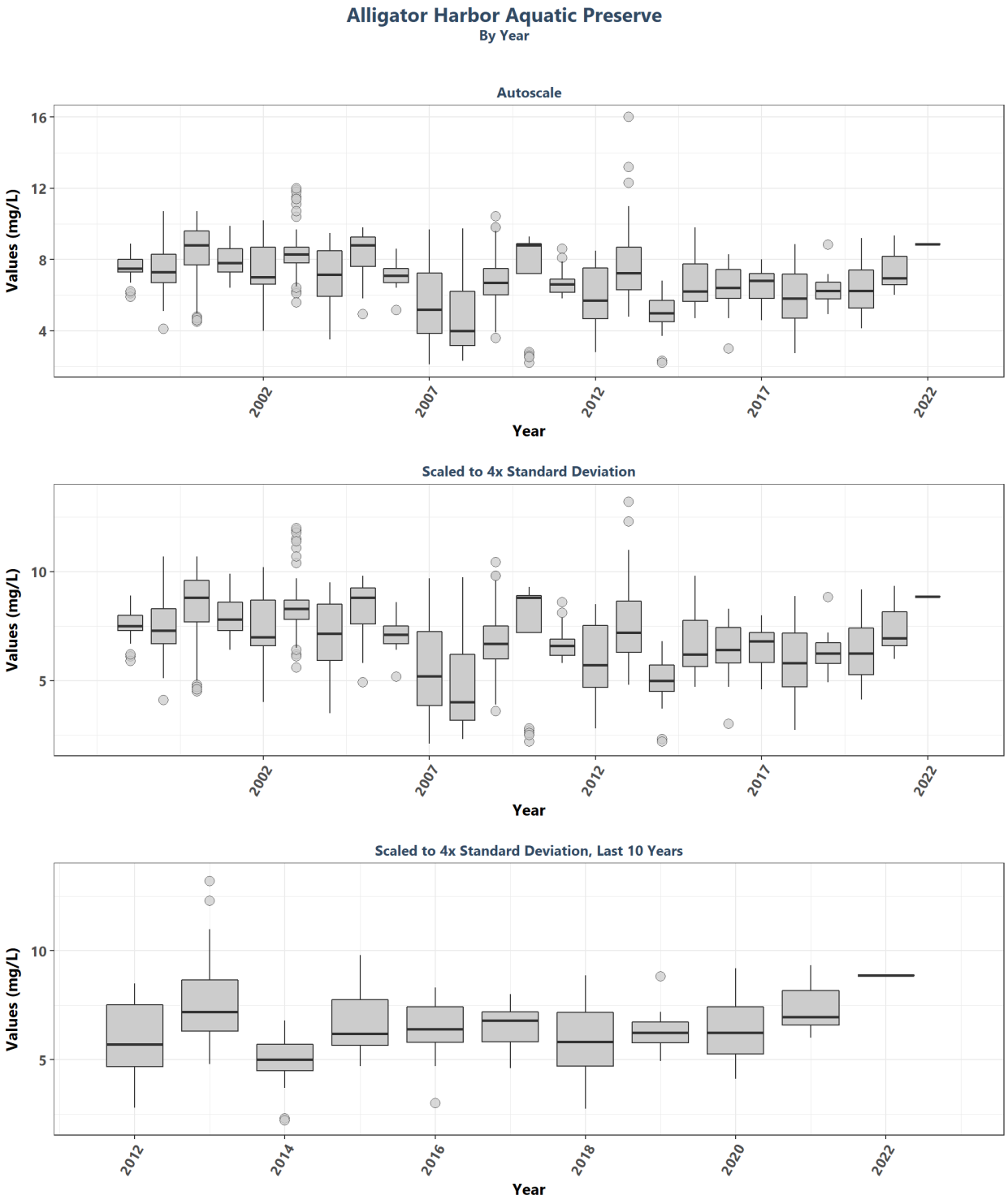
1. Y-axis autoscaled
2. Y-axis set to be mean + 4 times the standard deviation
3. Y-axis set to be mean + 4 times the standard deviation for most recent 10 years of data

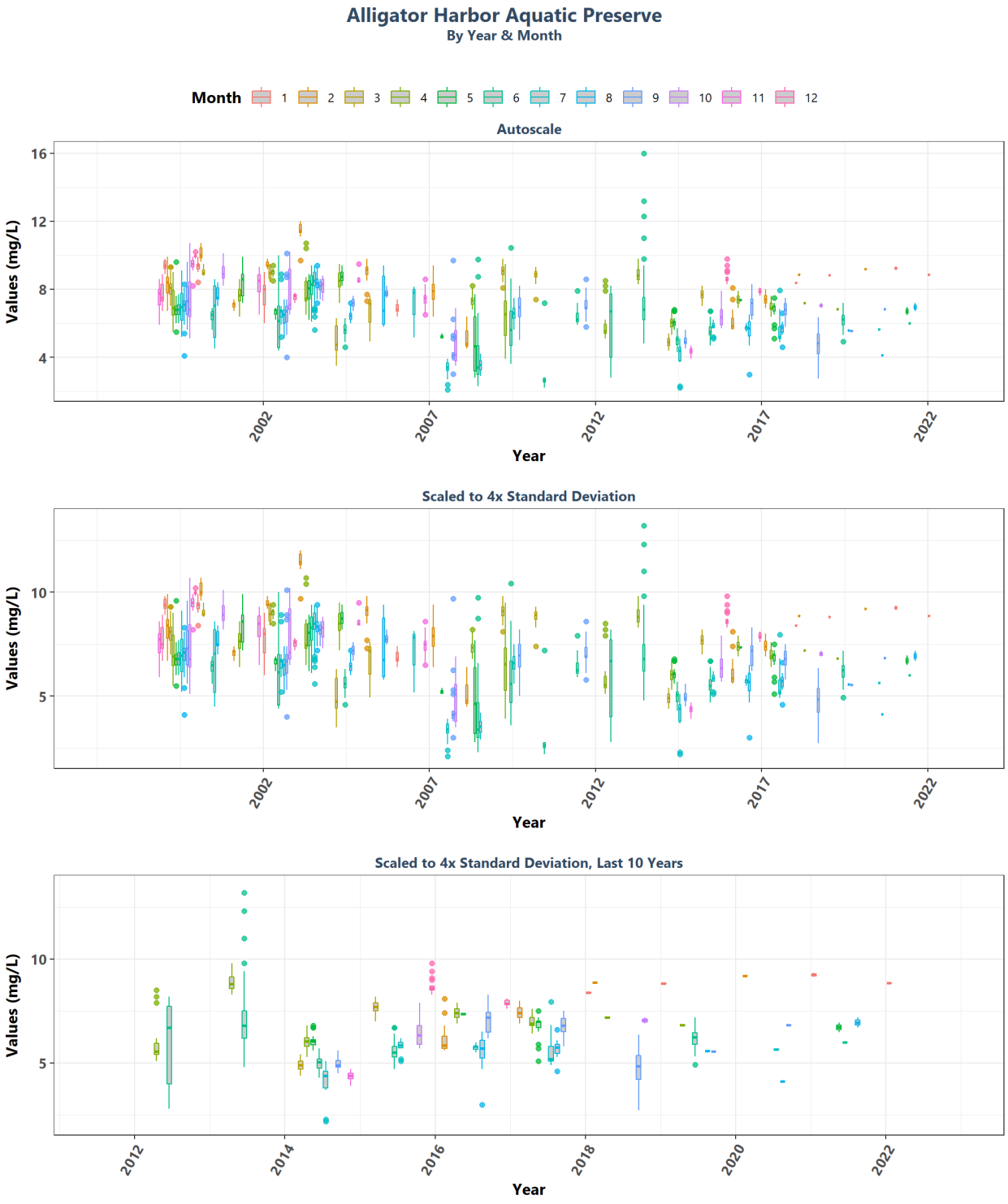
```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$SufficientData==TRUE &
                           data$ManagedAreaName==MA_Include[i],]
      year_lower <- min(plot_data$Year)
      year_upper <- max(plot_data$Year)
      min_RV <- min(plot_data$ResultValue)
      mn_RV <- mean(plot_data$ResultValue[plot_data$ResultValue <
                                             quantile(data$ResultValue, 0.98)])
      sd_RV <- sd(plot_data$ResultValue[plot_data$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=plot_data,
                   aes(x=Year, y=ResultValue, group=Year)) +
         geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                      outlier.size=3, outlier.color="#333333",
                      outlier.fill="#cccccc", outlier.alpha=0.75) +
         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))) +
         plot_theme
```

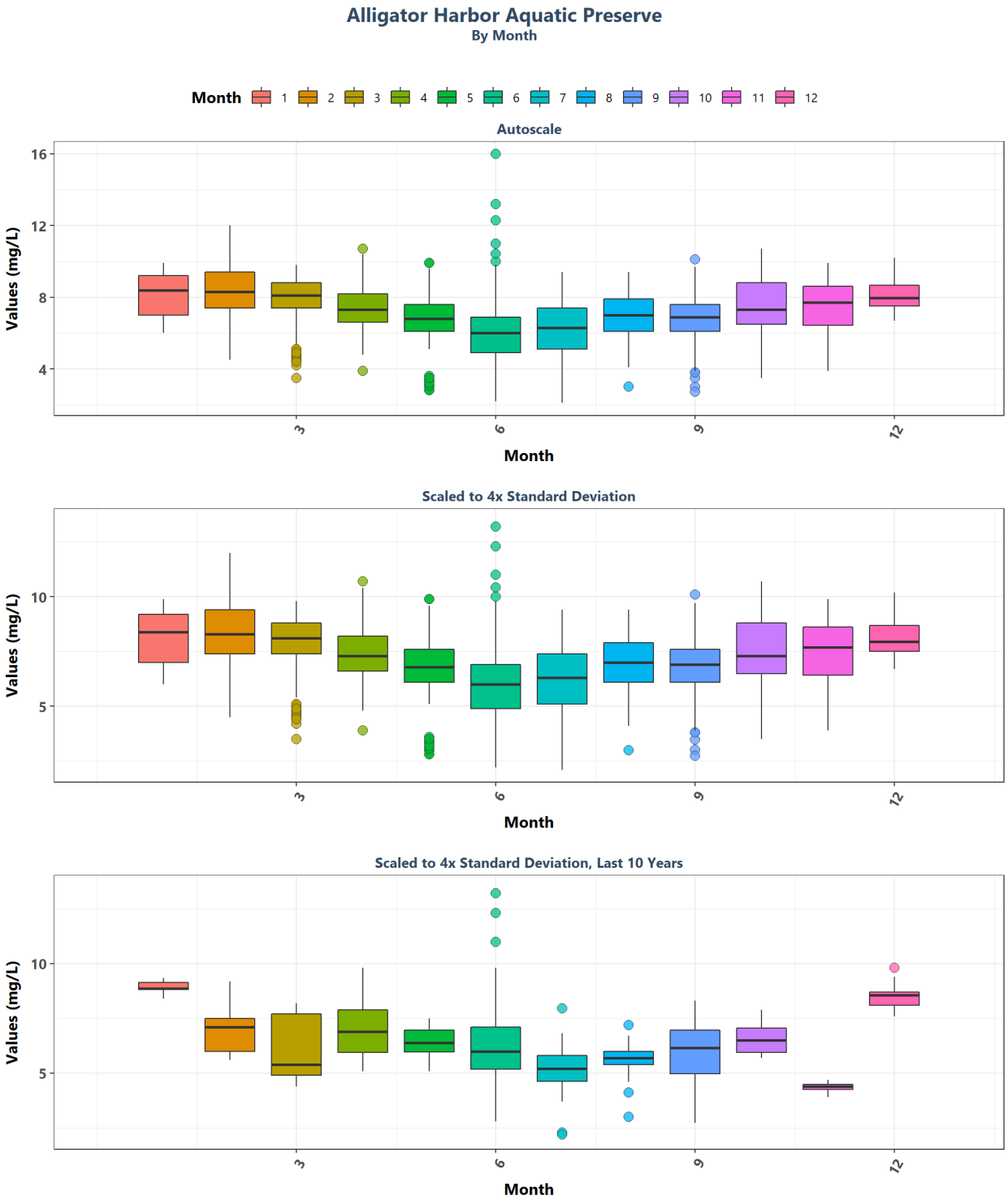
```
p2 <- ggplot(data=plot_data,
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 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))) +
   plot_theme

p3 <- ggplot(data=plot_data[plot_data$Year >= year_upper - 10, ],
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   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))) +
   plot_theme

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

p0 <- ggplot() + labs(title=paste0(MA_Include[i]),
                      subtitle="By Year") +
   plot_theme + theme(panel.border=element_blank(),
                      panel.grid.major=element_blank(),
                      panel.grid.minor=element_blank(),
                      axis.line=element_blank())

## Year & Month Plots
p4 <- ggplot(data=plot_data,
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   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))) +
   plot_theme +
   theme(legend.position="none")

p5 <- ggplot(data=plot_data,
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   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))) +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
   guides(color=guide_legend(nrow=1))

p6 <- ggplot(data=plot_data[plot_data$Year >= year_upper - 10, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   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))) +
   plot_theme +
   theme(legend.position="none")

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(MA_Include[i]),
                       subtitle="By Year & Month") + plot_theme +
   theme(panel.border=element_blank(),
         panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

## Month Plots
p7 <- ggplot(data=plot_data,
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="none")

p8 <- ggplot(data=plot_data,
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   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)) +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
```

```
        guides(fill=guide_legend(nrow=1))

    p9 <- ggplot(data=plot_data[plot_data$Year >= year_upper - 10, ],
                 aes(x=Month, y=ResultValue,
                     group=Month, fill=as.factor(Month))) +
        geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                     outlier.color="#333333", outlier.alpha=0.75) +
        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)) +
        plot_theme +
        theme(legend.position="none")

    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(MA_Include[i]),
                            subtitle="By Month") + plot_theme +
        theme(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)))

    rm(plot_data)
    rm(p1, p2, p3, p4, p5, p6, p7, p8, p9, p0, p00, p000, leg1, leg2,
       Yset, YMset, Mset)
  }
}
```

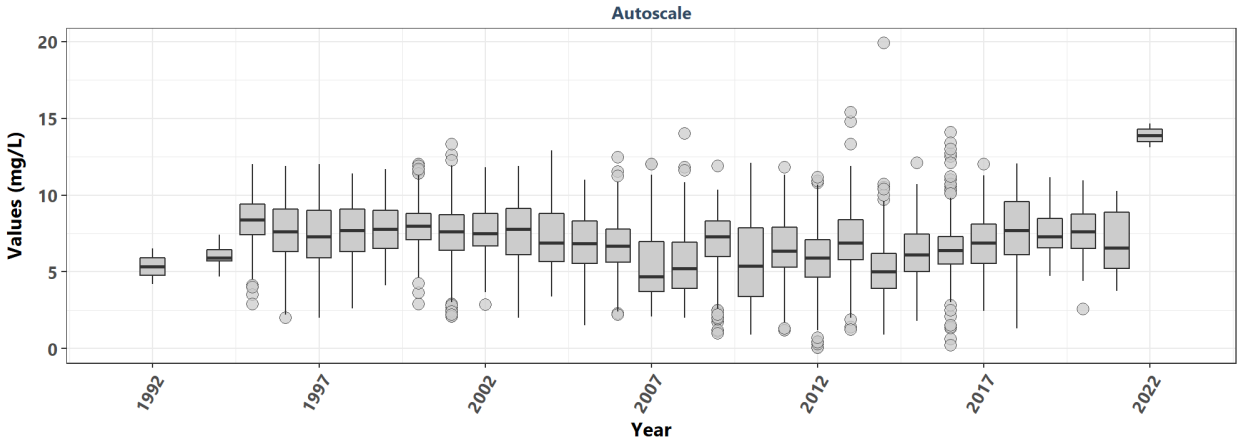

Alligator Harbor Aquatic Preserve
By Year
Autoscale
Values (mg/L)
16
12
8
4
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
2012
2014
2016
2018
2020
2022
Year

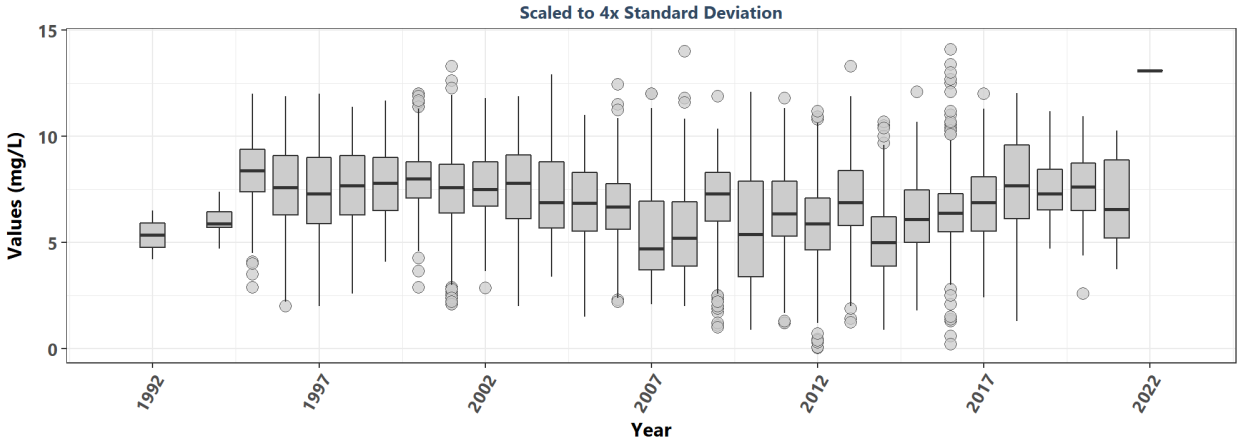

Alligator Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
16
12
8
4
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
2012
2014
2016
2018
2020
2022
Year

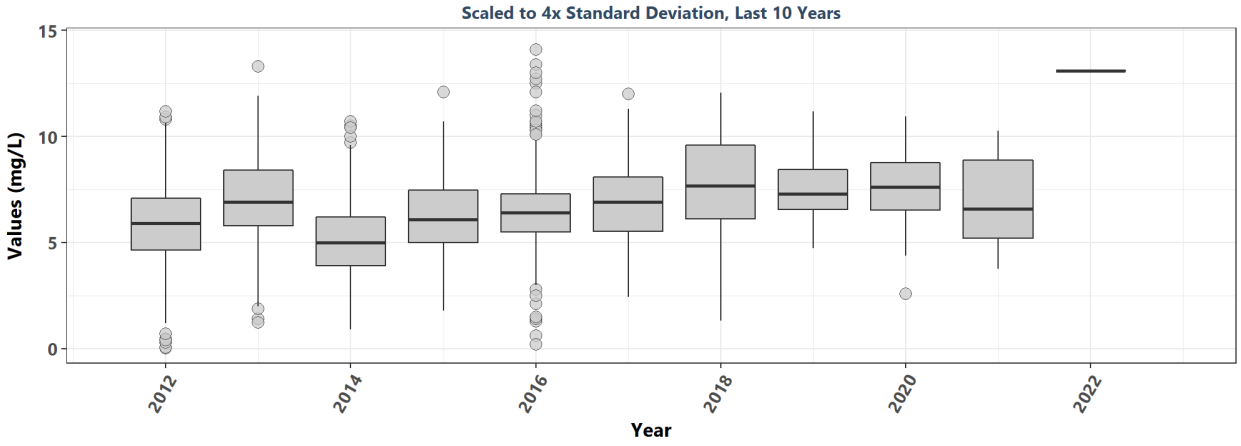

Alligator Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
16
12
8
4
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
3
6
9
12
Month

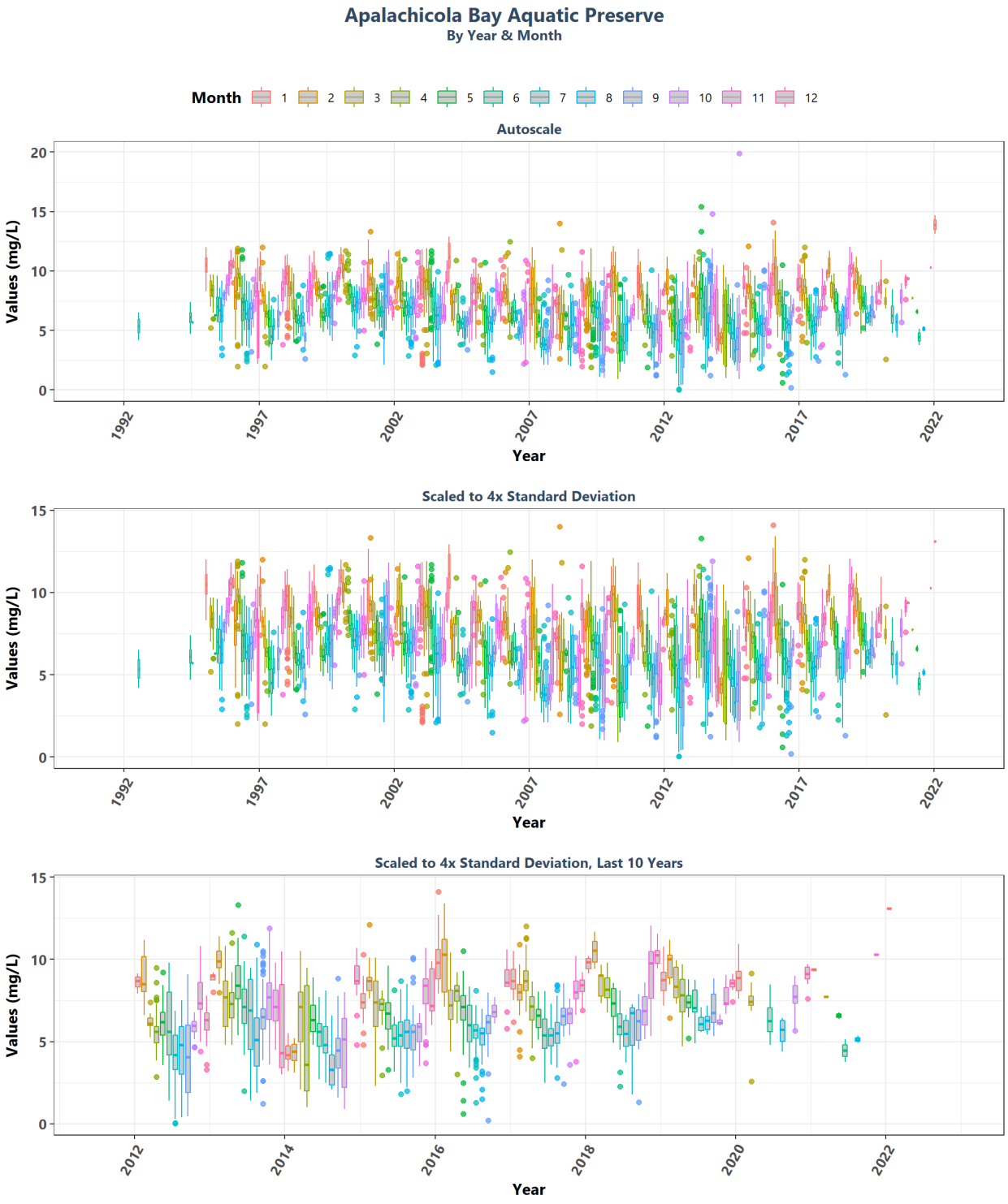
Apalachicola Bay Aquatic Preserve

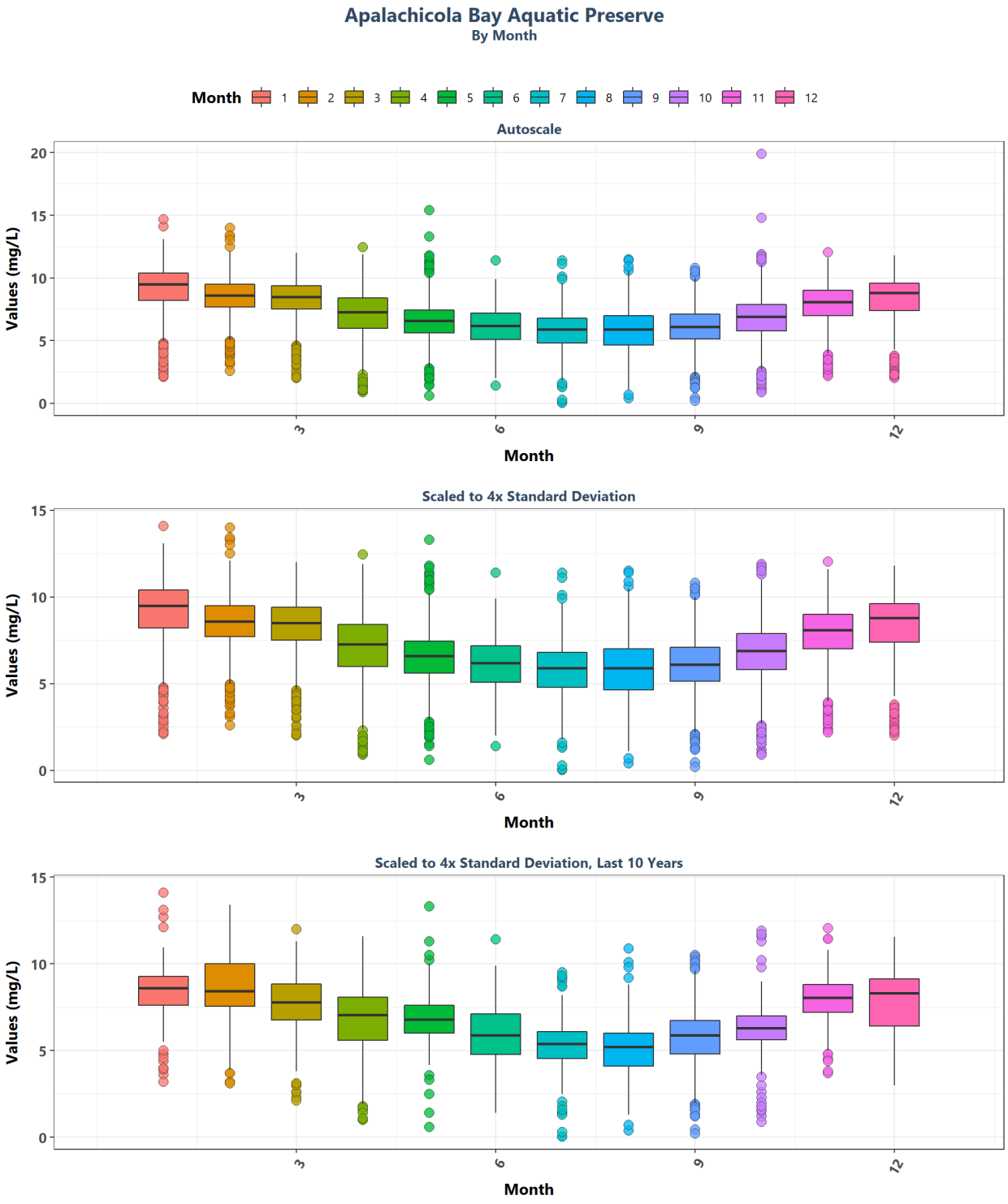
By Year





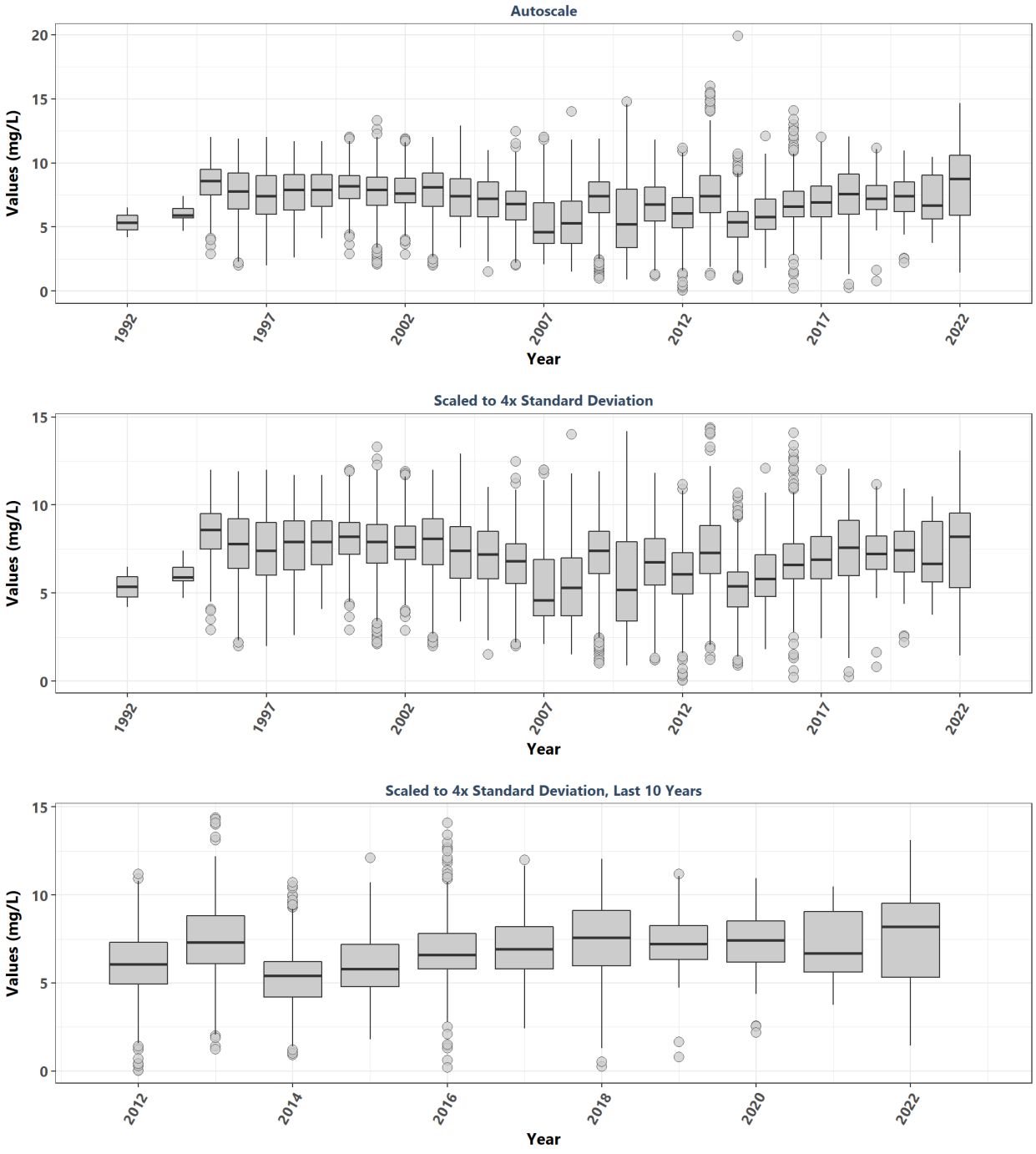


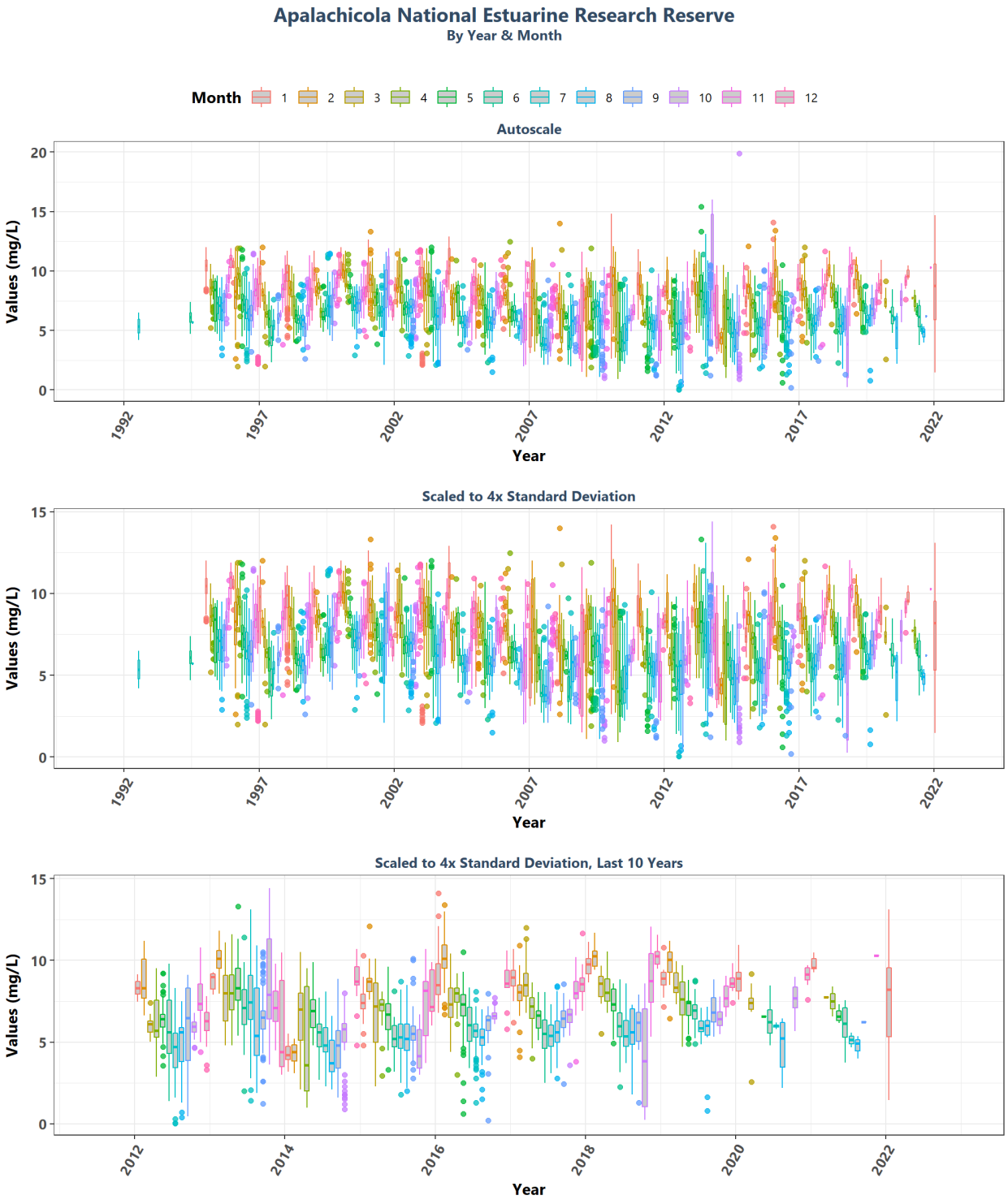

Apalachicola Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

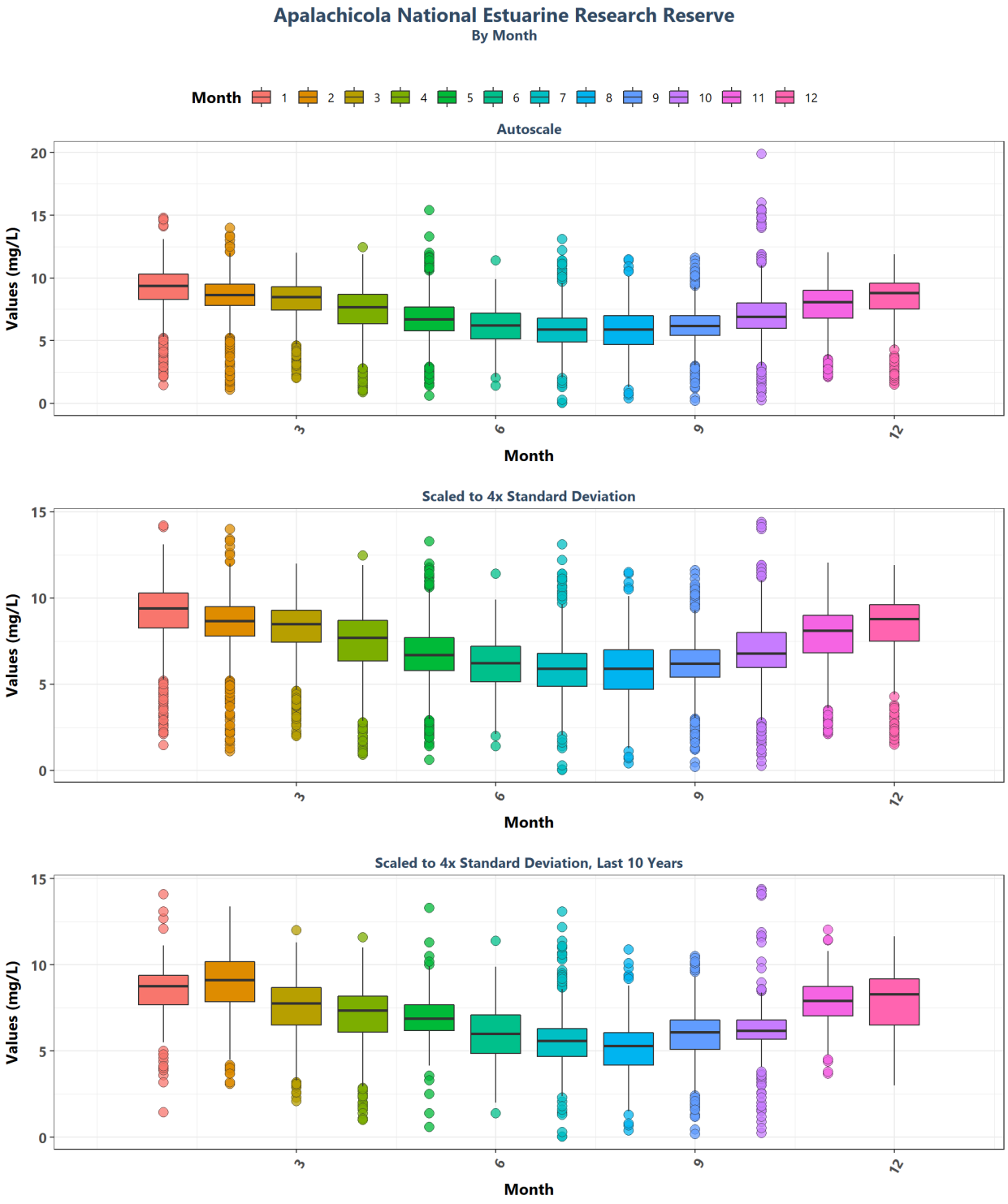

Apalachicola Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

Apalachicola National Estuarine Research Reserve

By Year

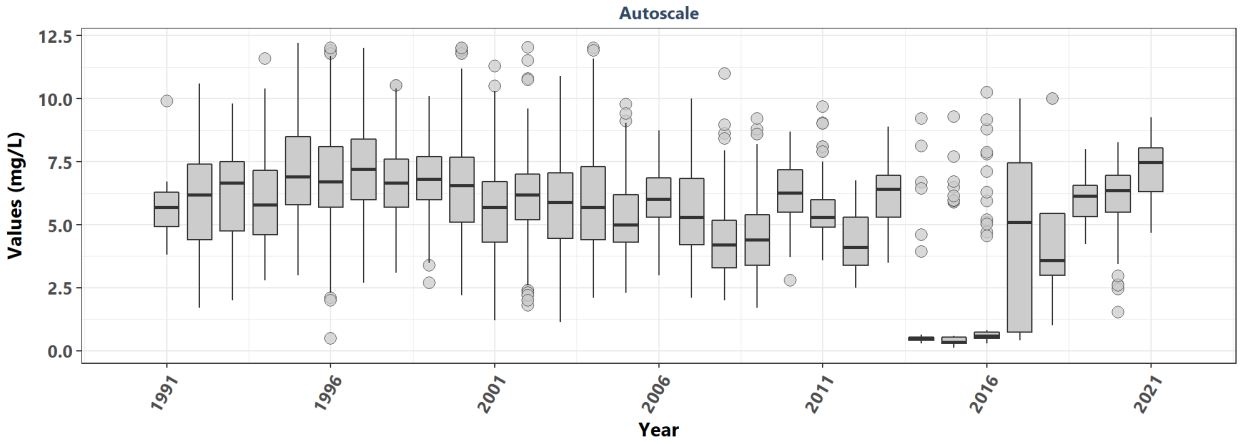


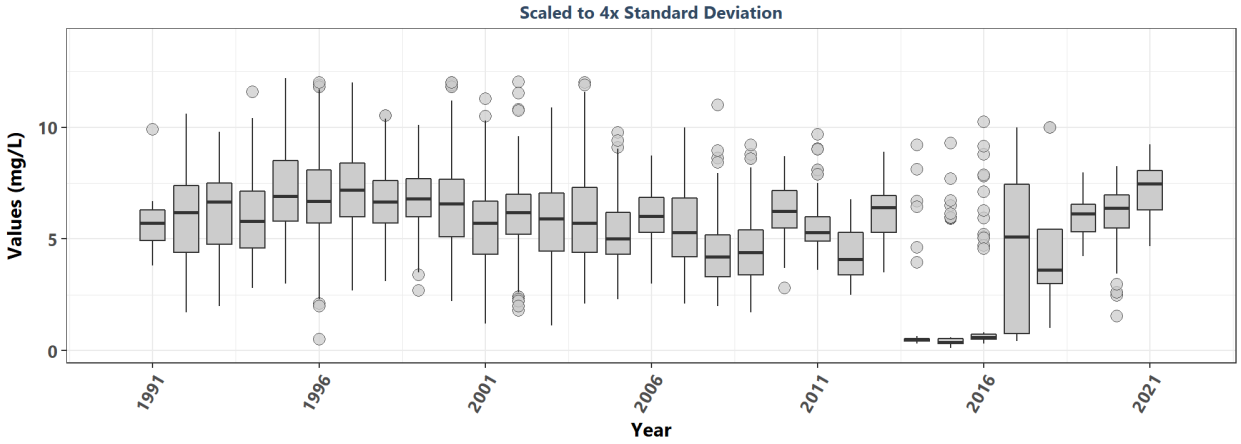

Apalachicola National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

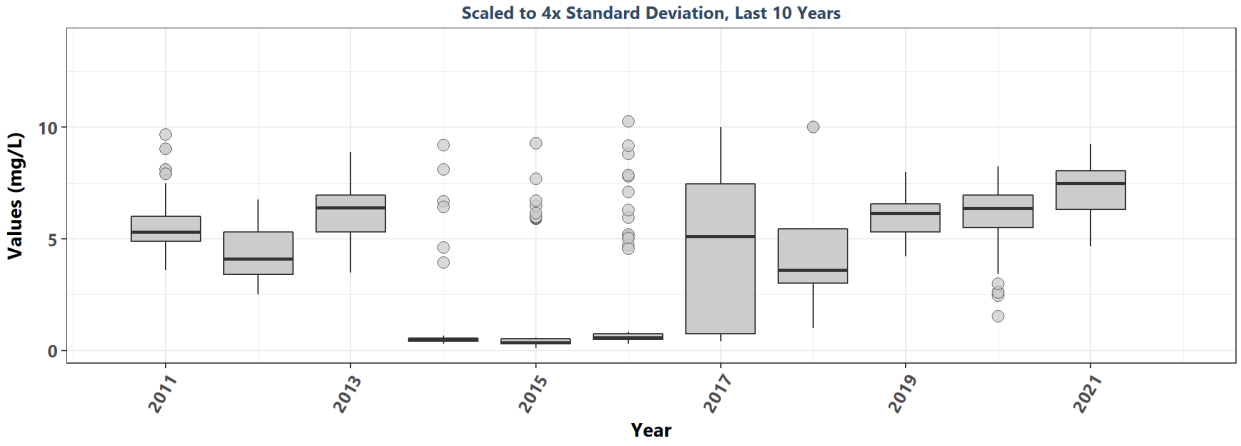

Apalachicola National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

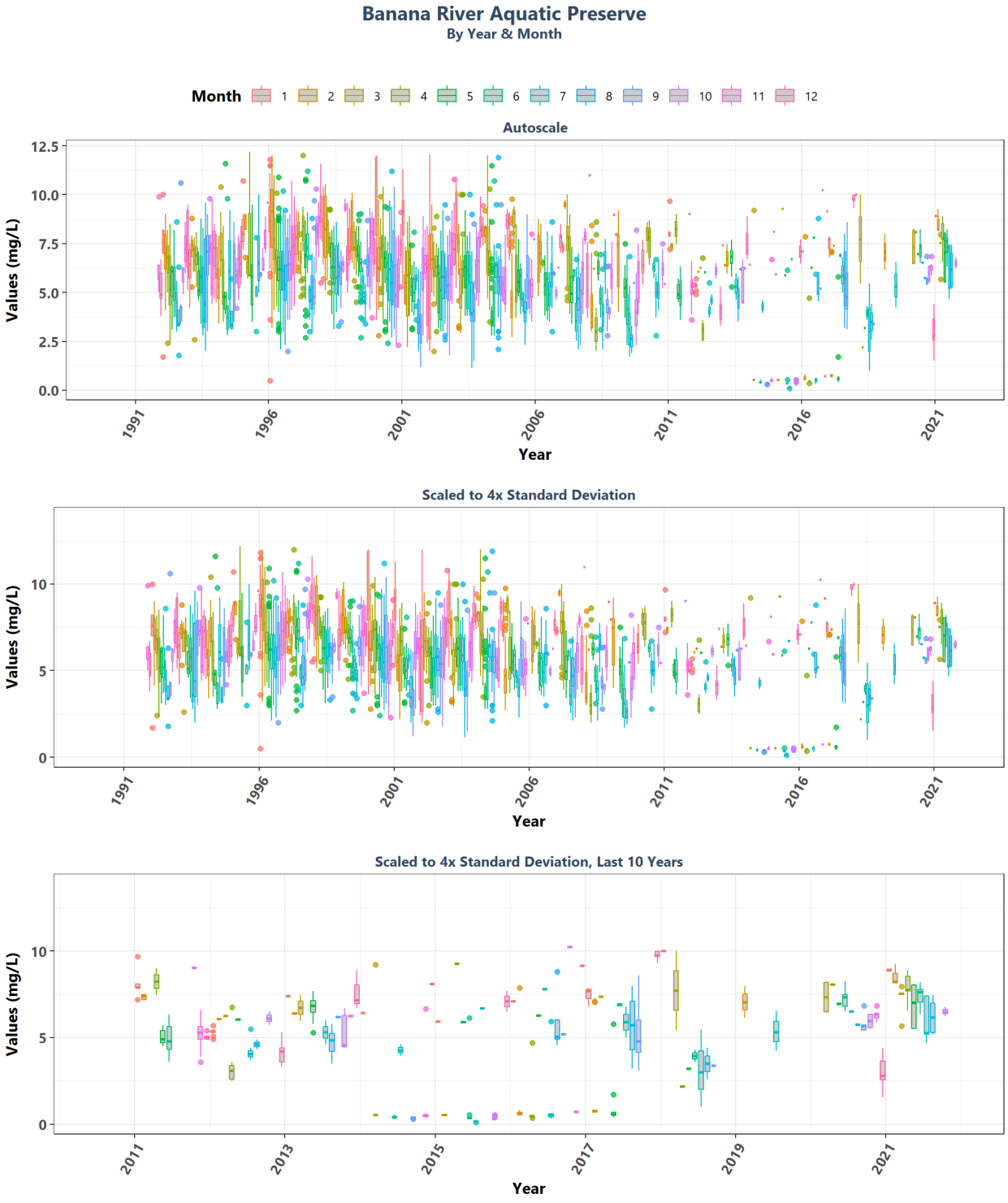
Banana River Aquatic Preserve

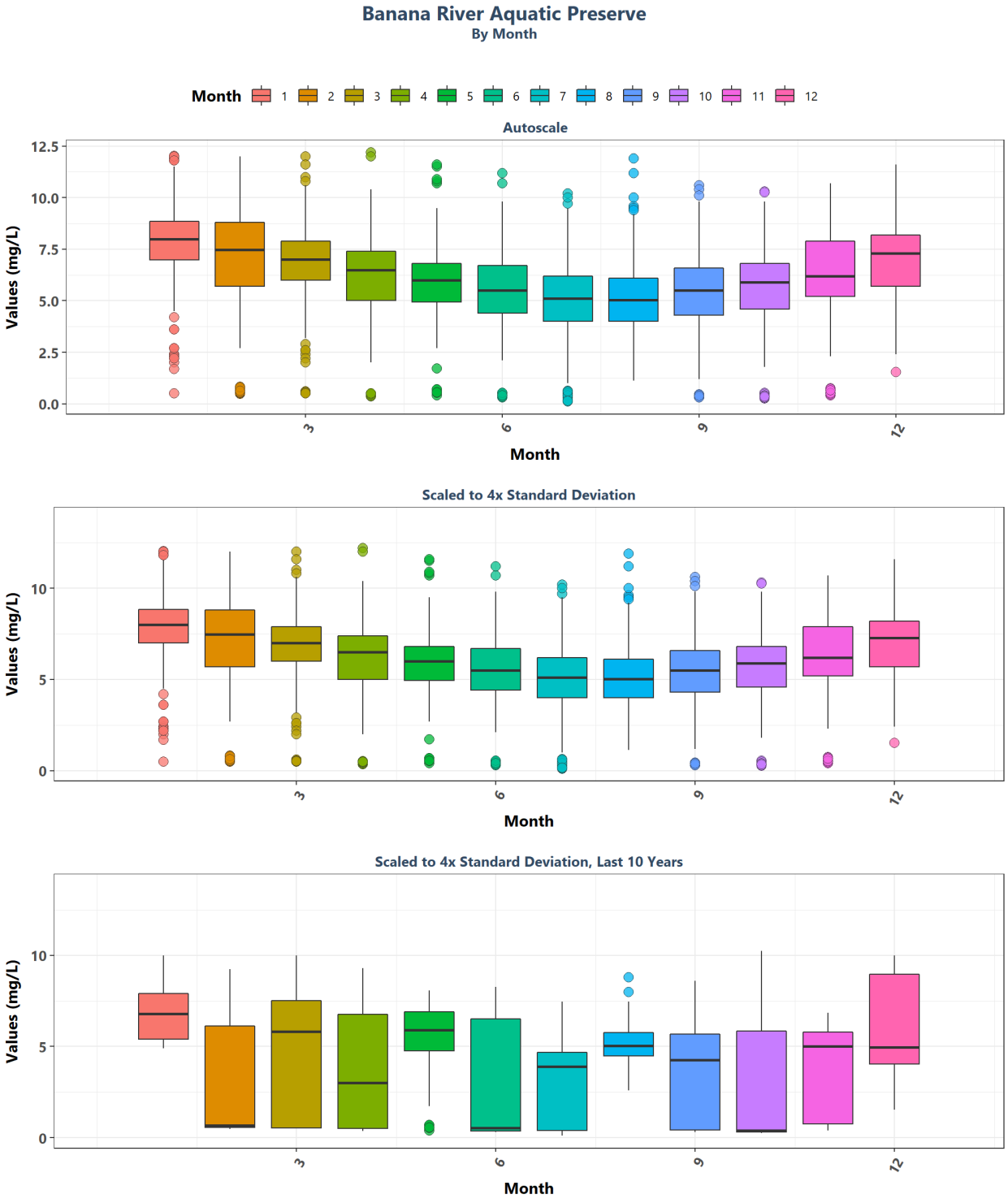
By Year

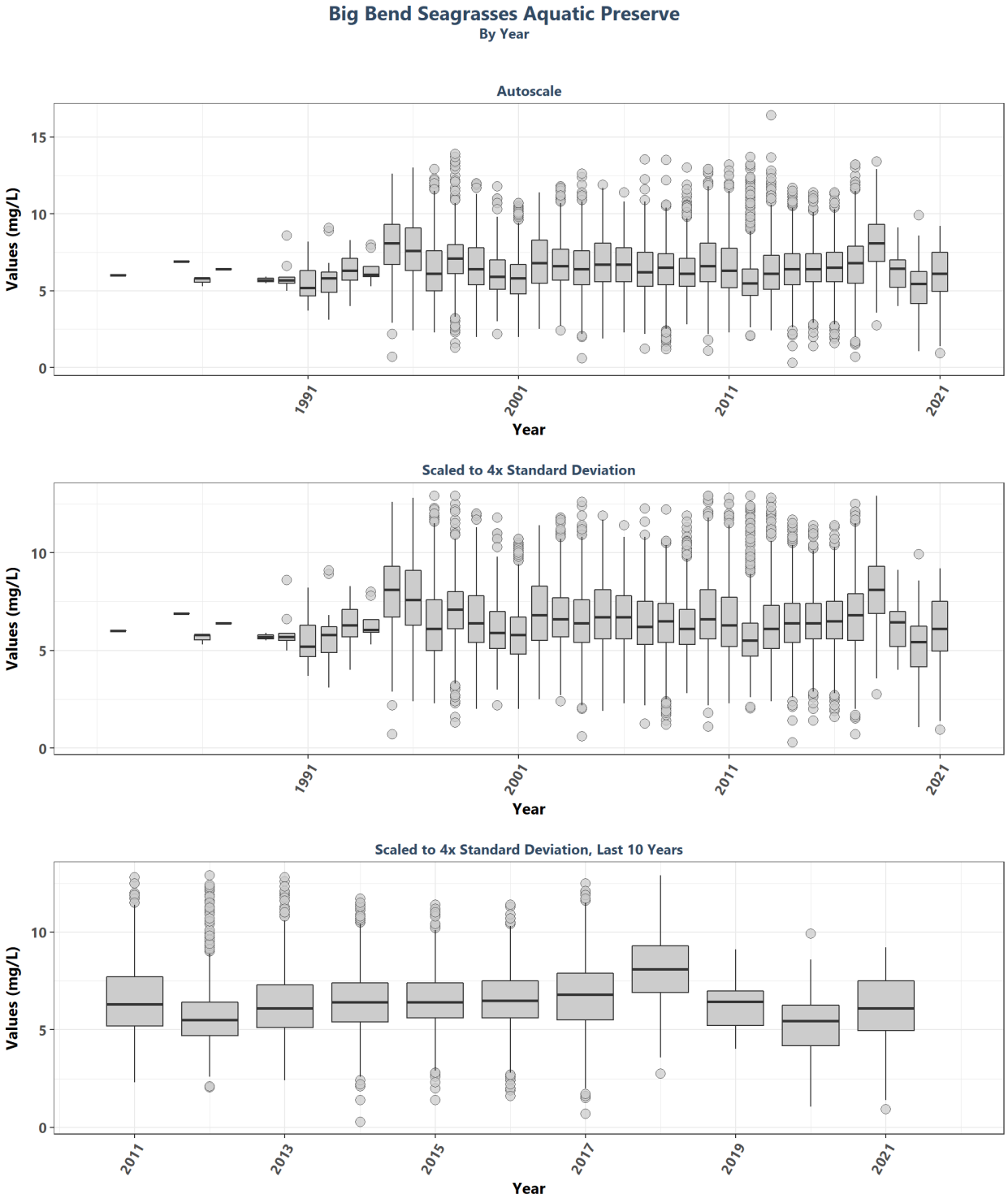


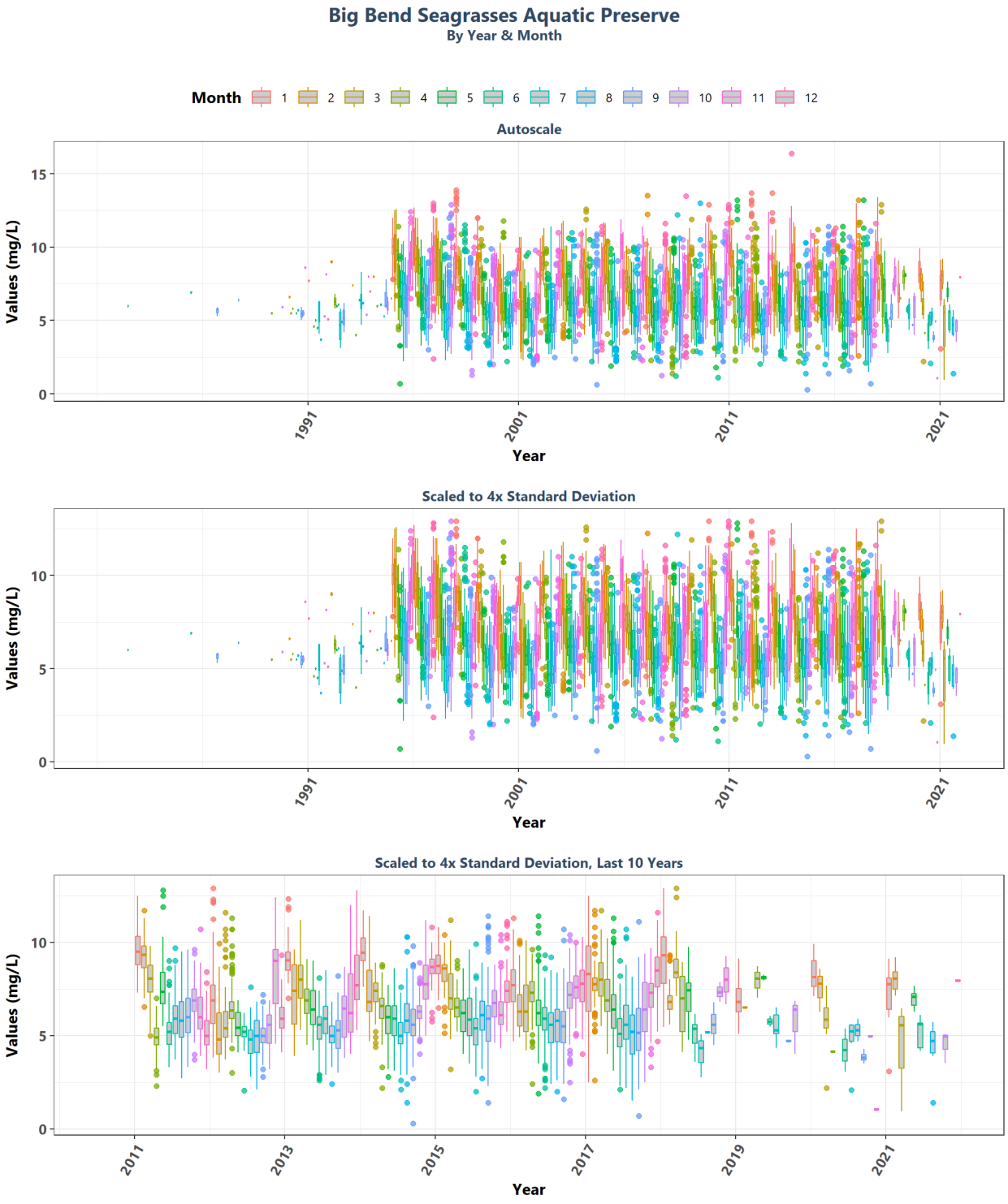


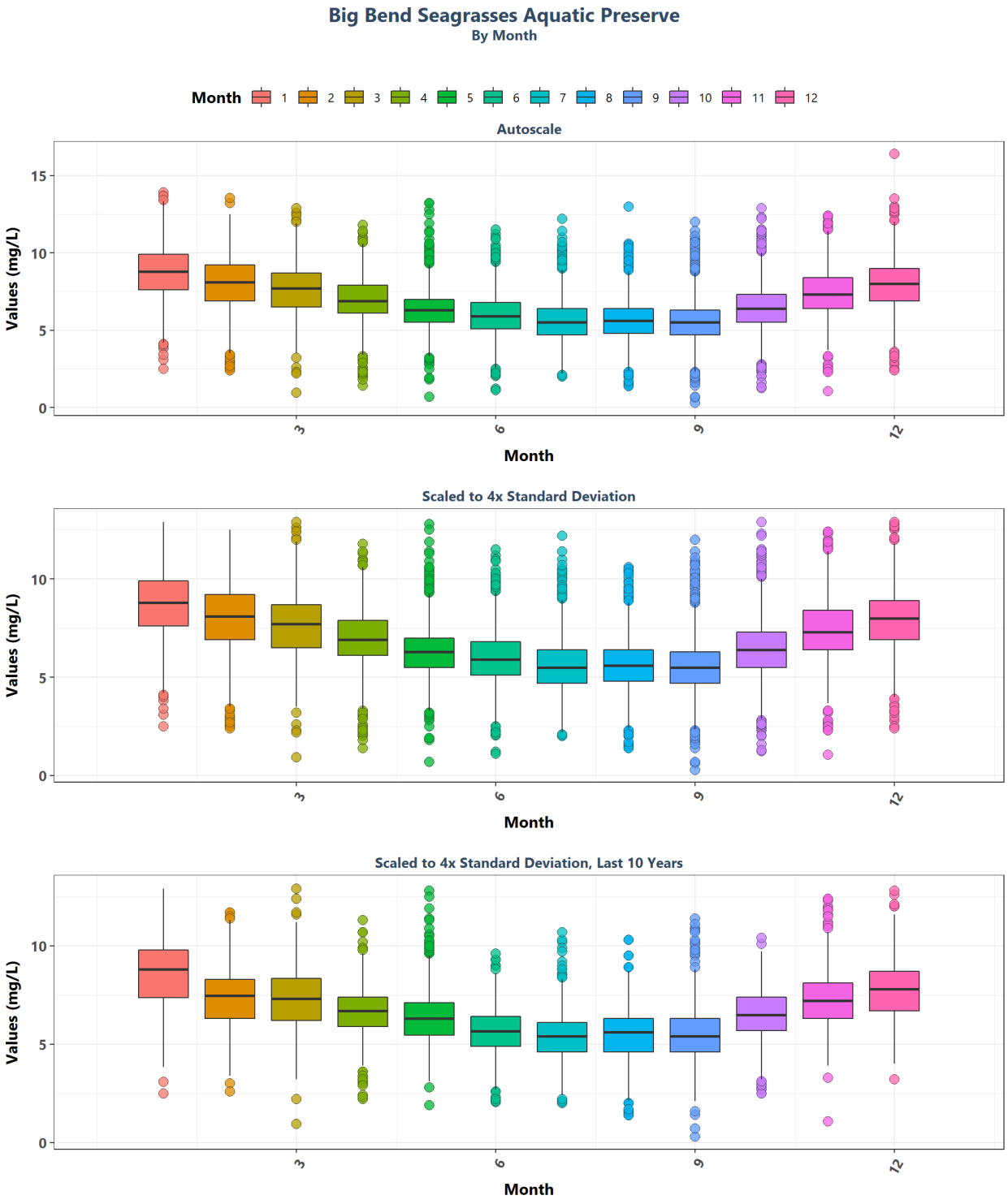



Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2011
2013
2015
2017
2019
2021
Year


Banana River Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

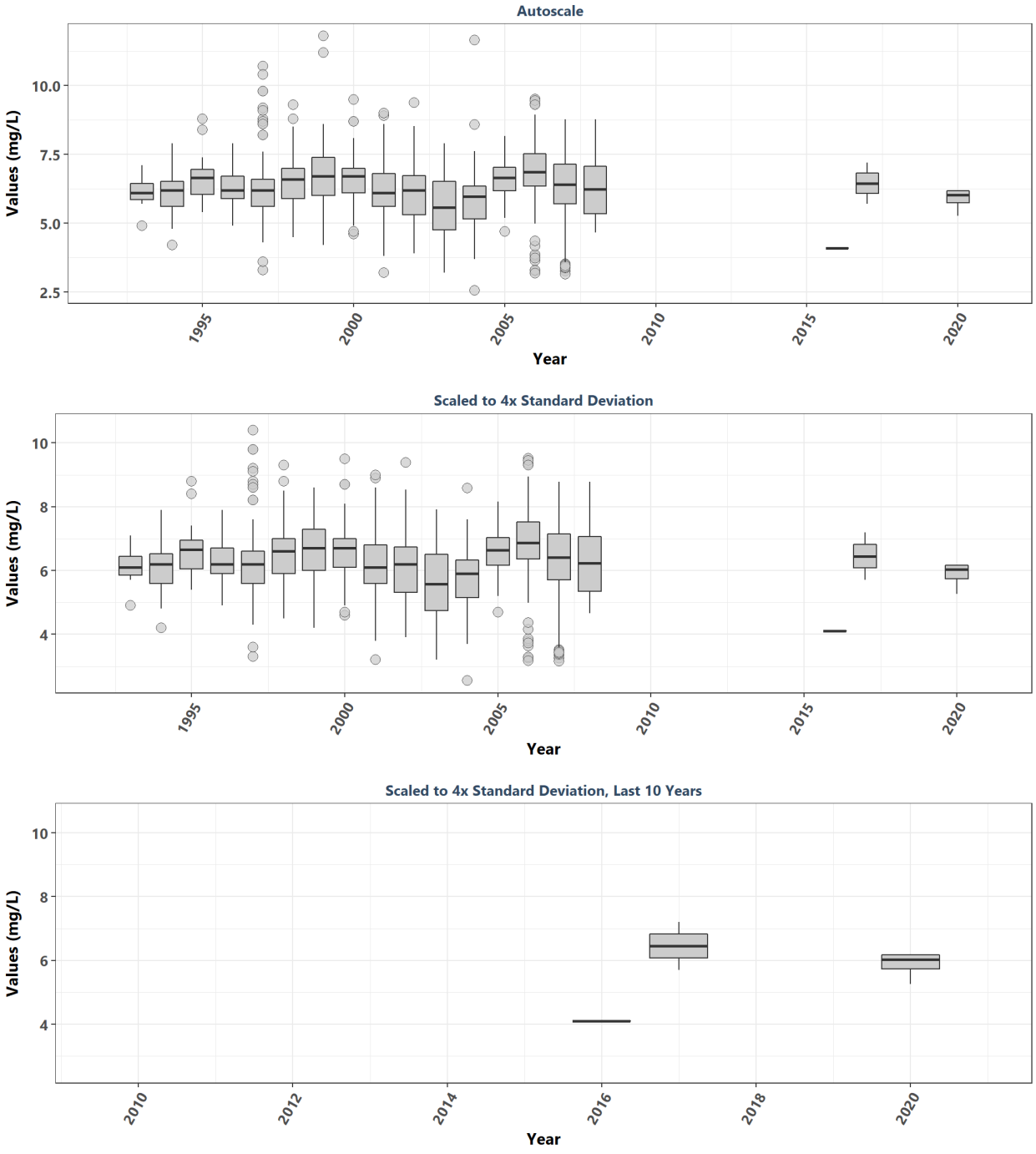

Big Bend Seagrasses Aquatic Preserve
By Year
Autoscale
Values (mg/L)
0
5
10
15
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
2011
2013
2015
2017
2019
2021
Year

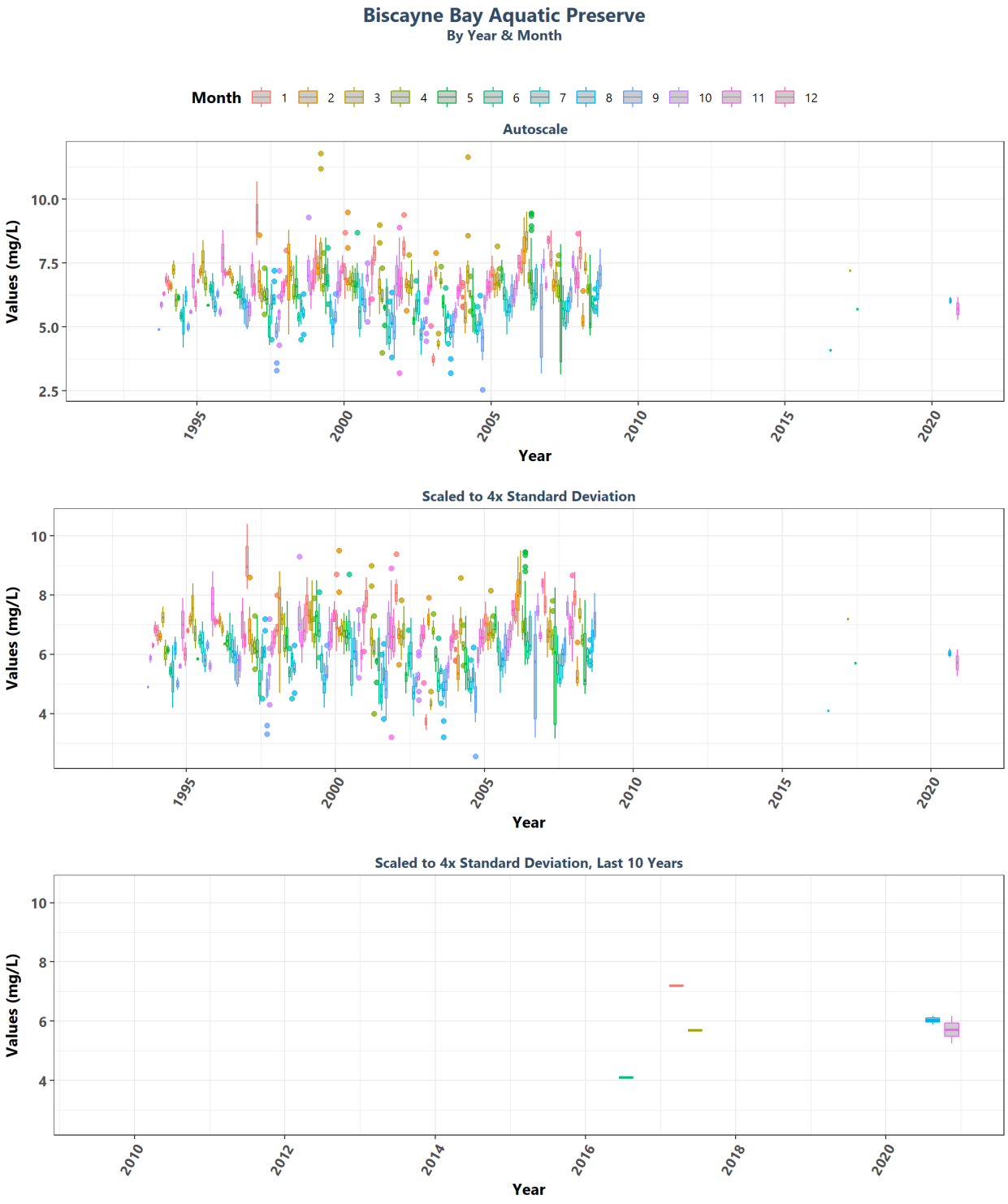

Big Bend Seagrasses Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
5
10
15
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
2011
2013
2015
2017
2019
2021
Year

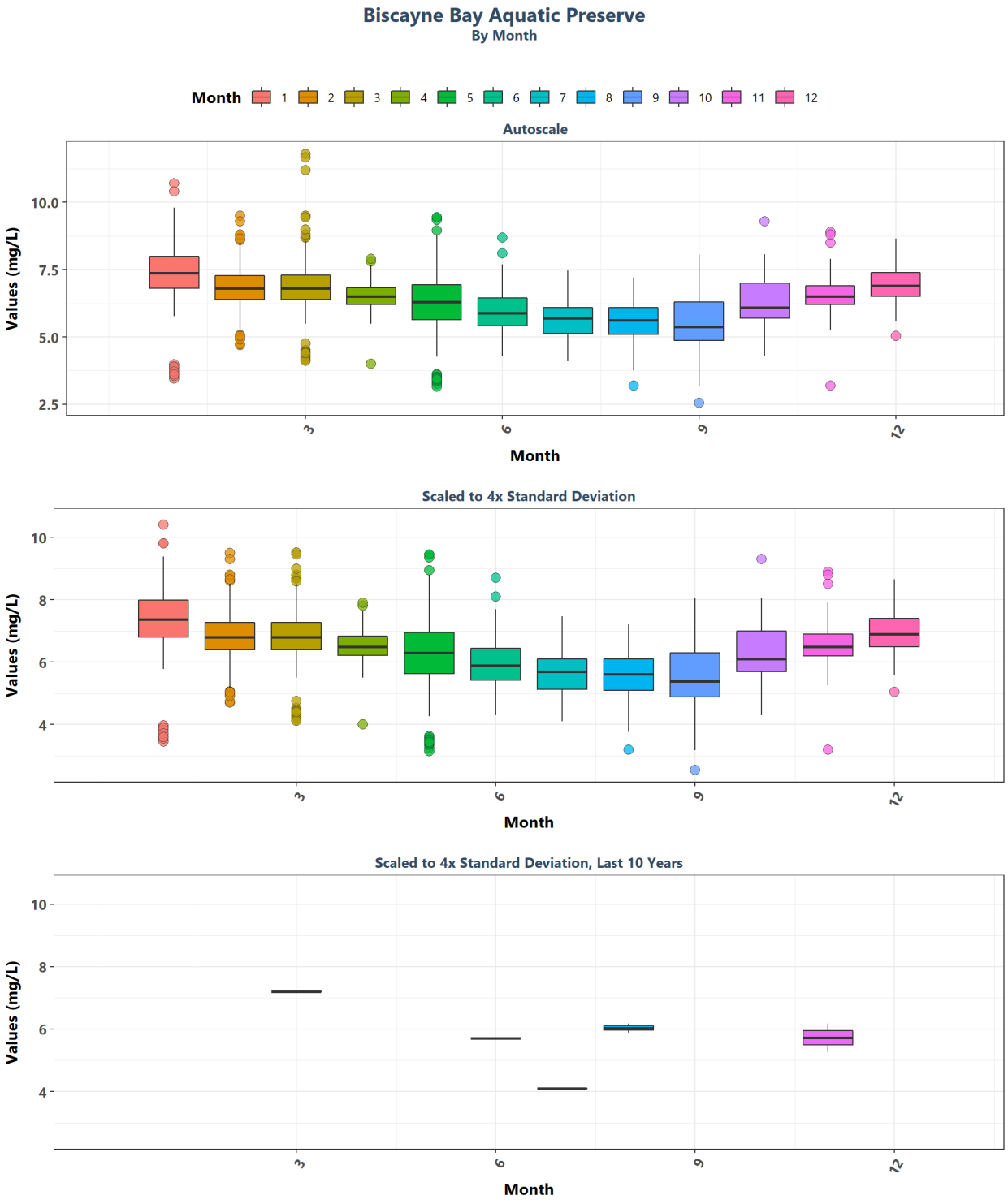

Big Bend Seagrasses Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

Biscayne Bay Aquatic Preserve

By Year

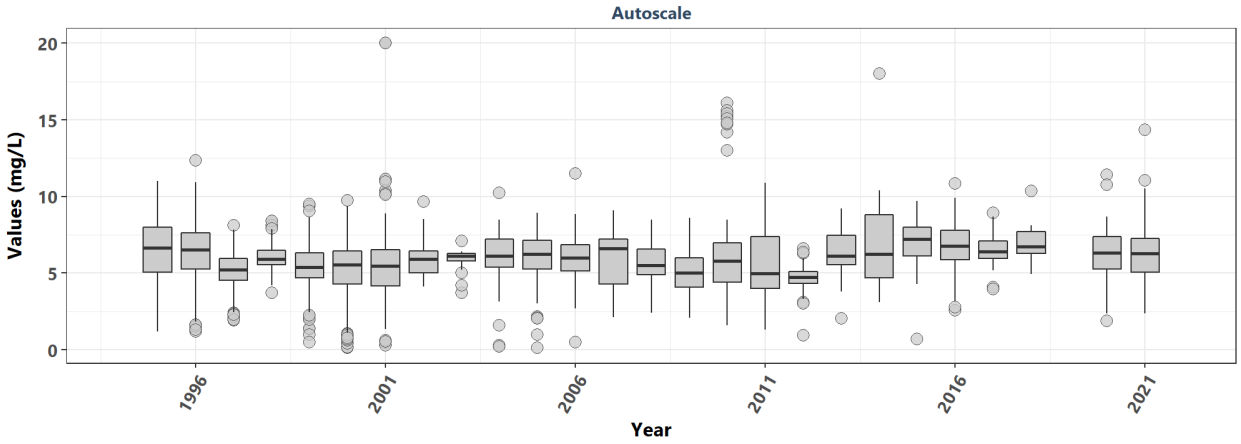


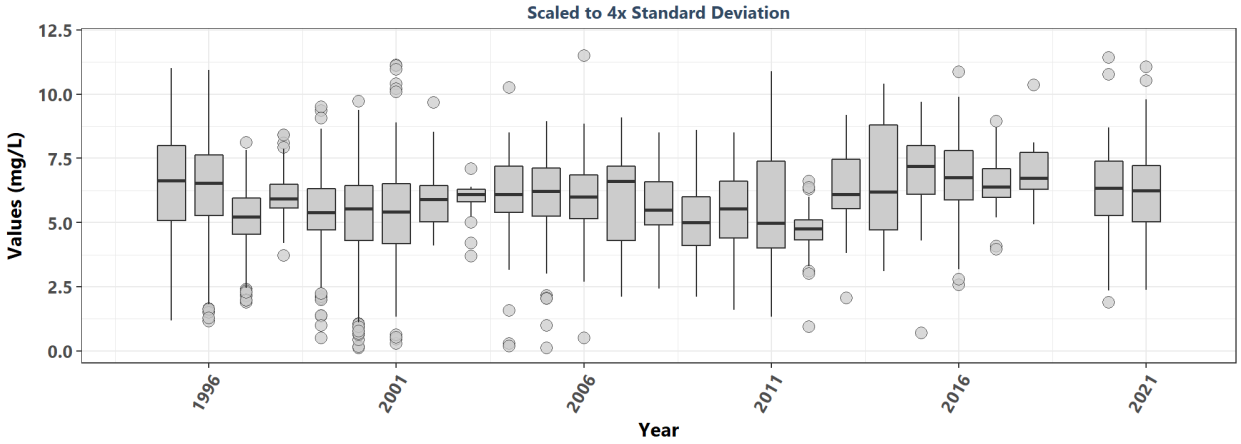

Biscayne Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
8
6
4
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
8
6
4
2010
2012
2014
2016
2018
2020
Year

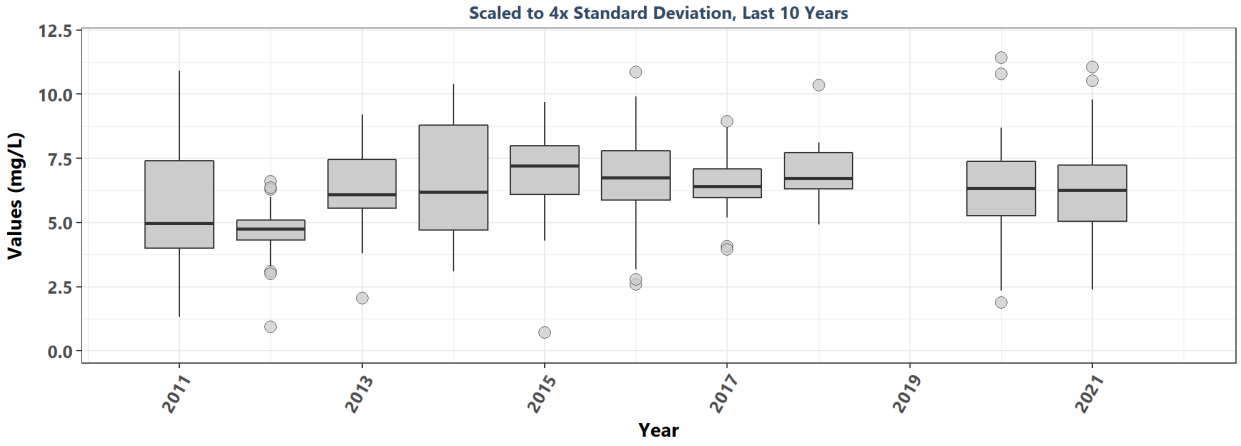

Biscayne Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
8
6
4
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
8
6
4
3
6
9
12
Month

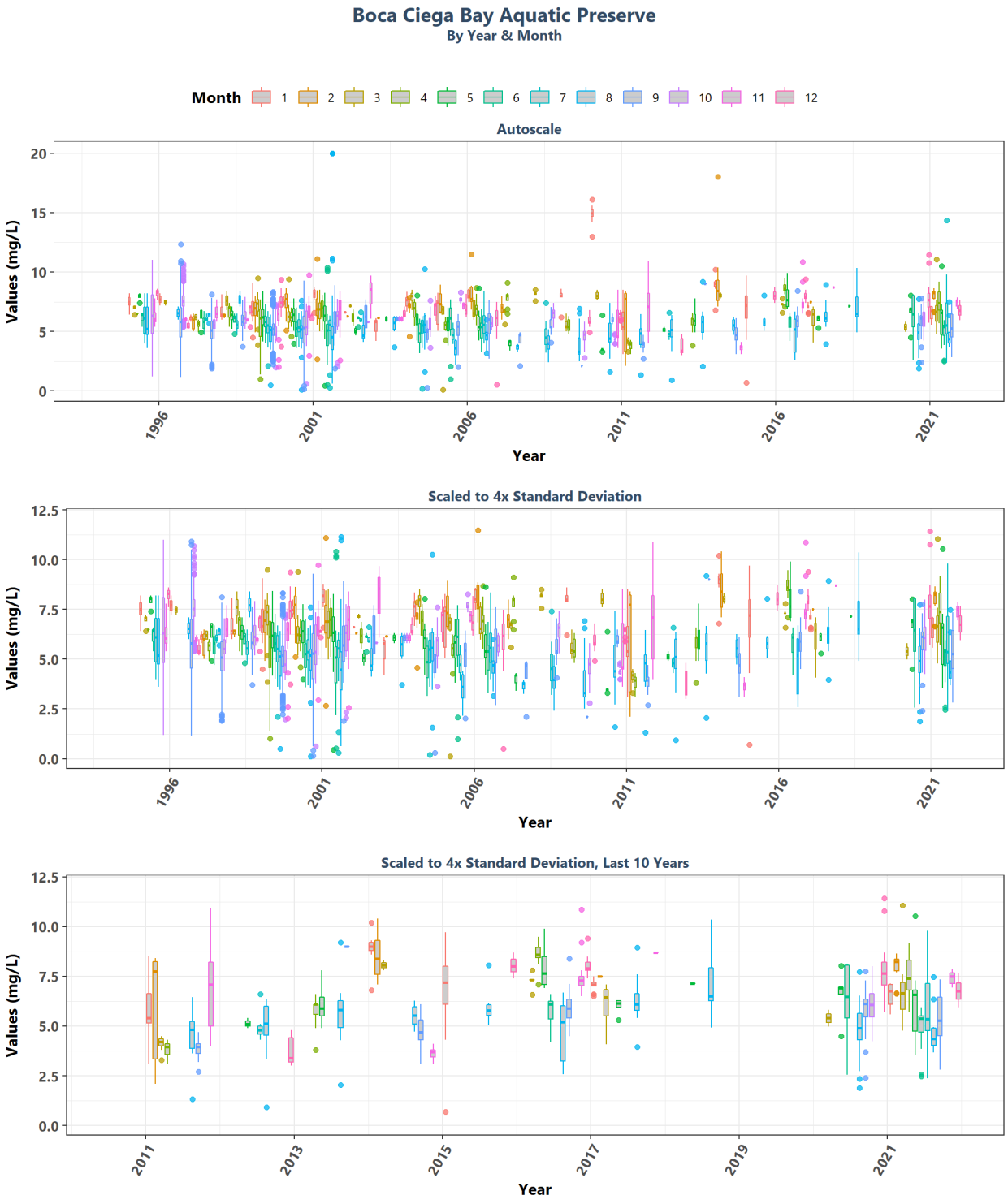
Boca Ciega Bay Aquatic Preserve

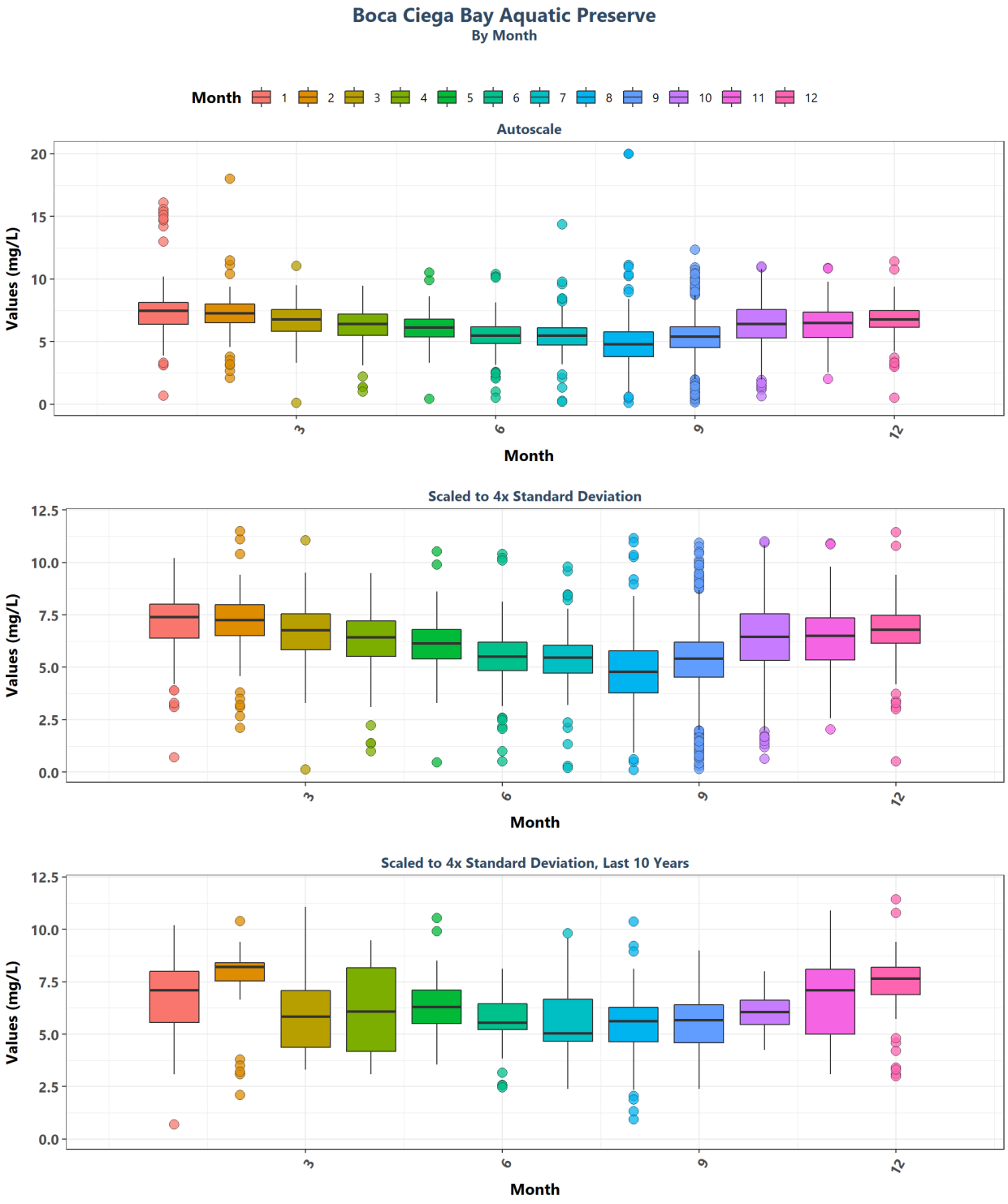
By Year



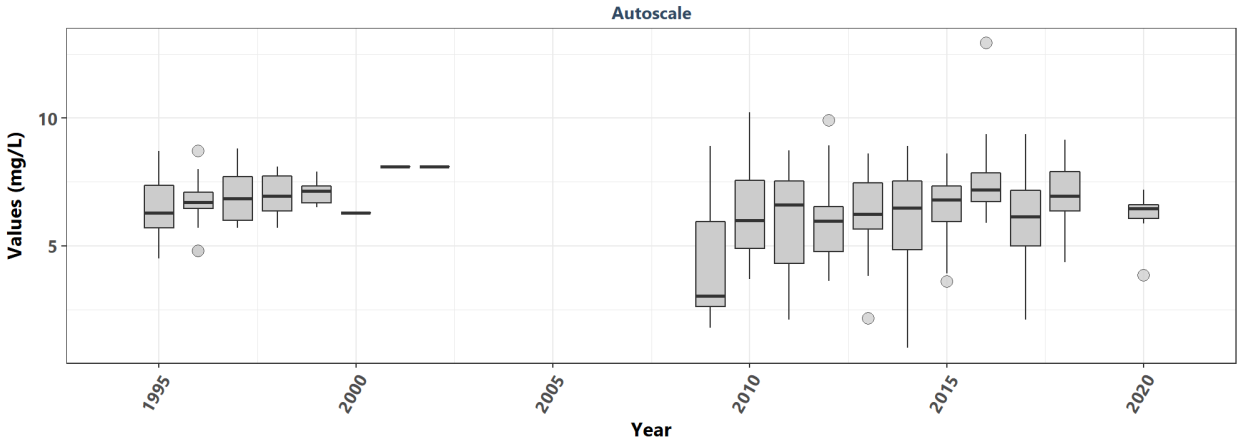


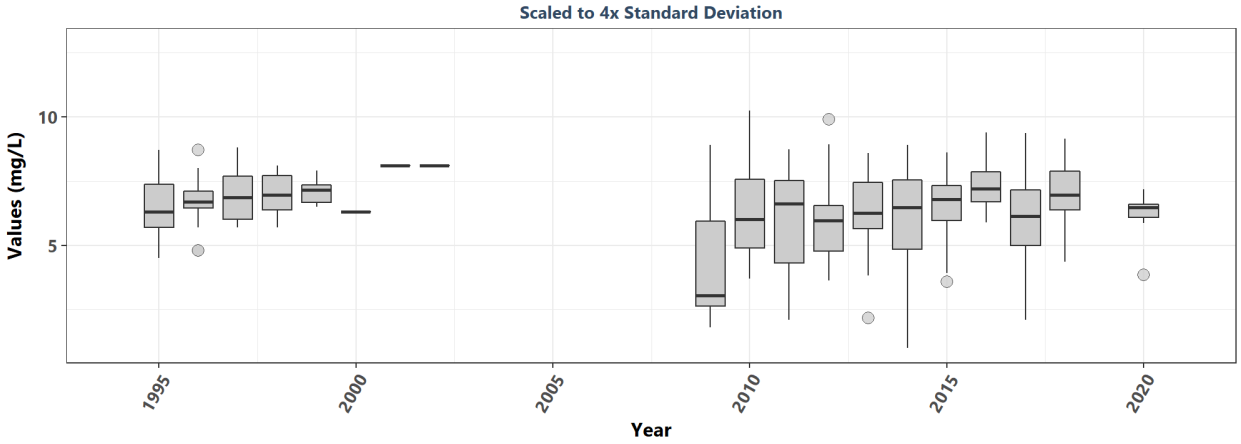


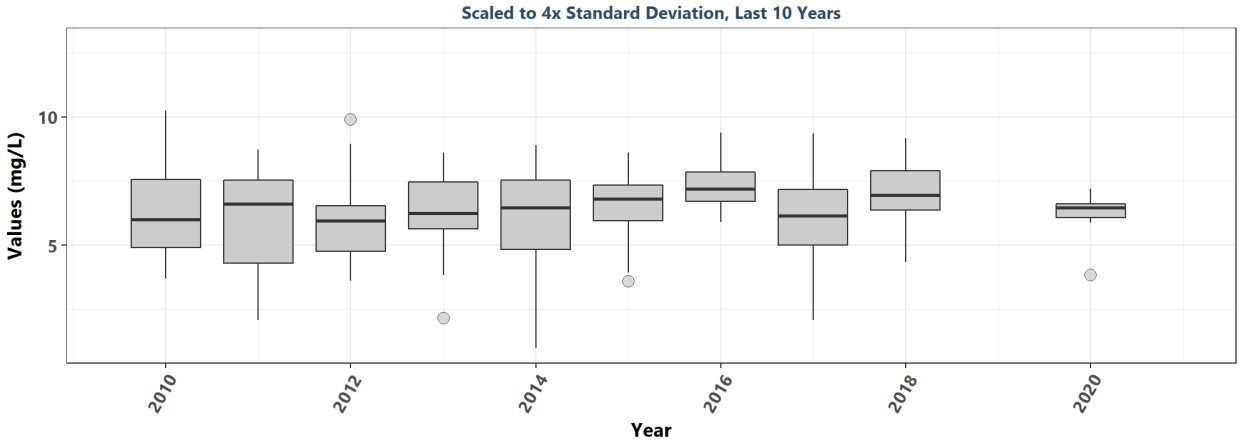

Boca Ciega Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

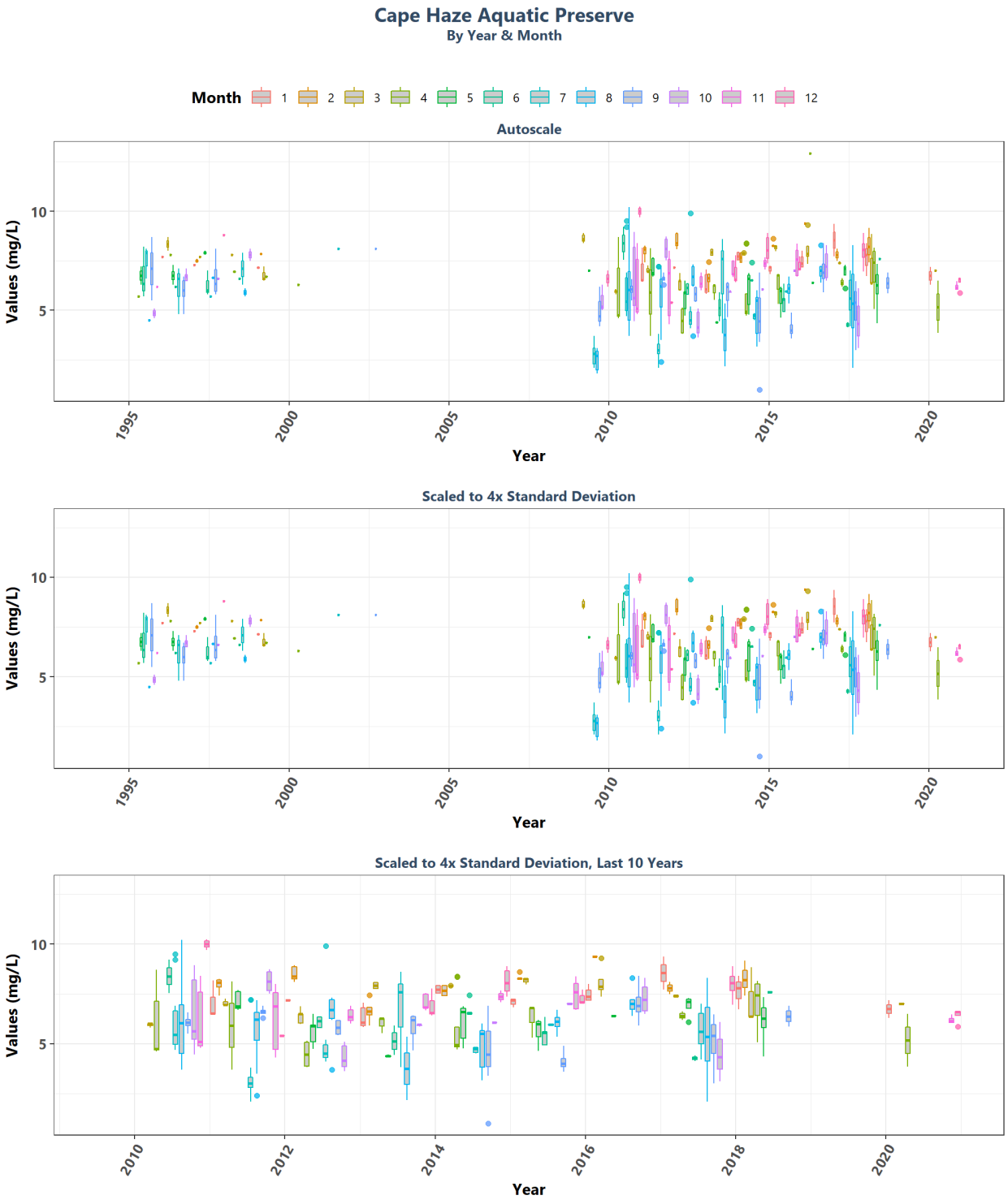

Boca Ciega Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

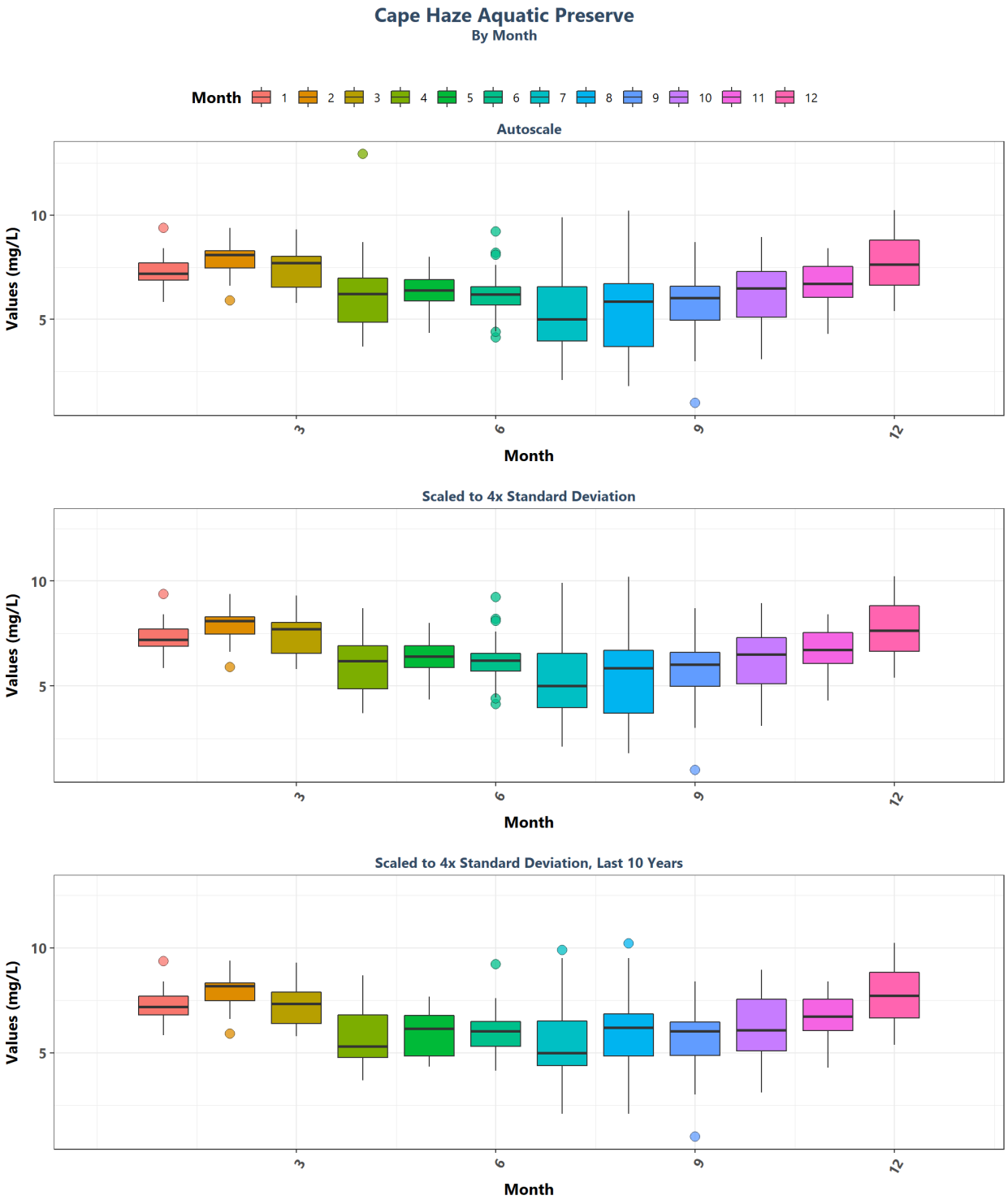
Cape Haze Aquatic Preserve
By Year





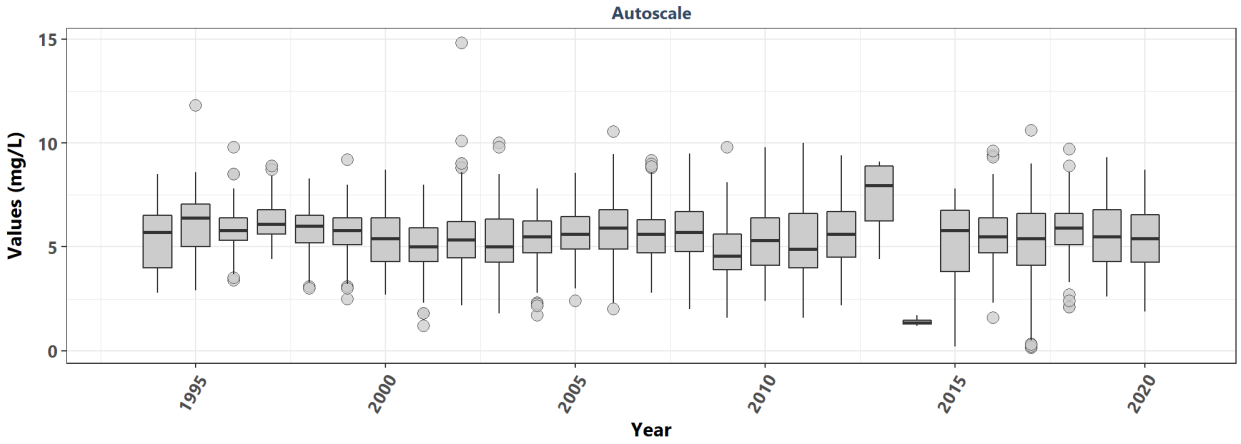


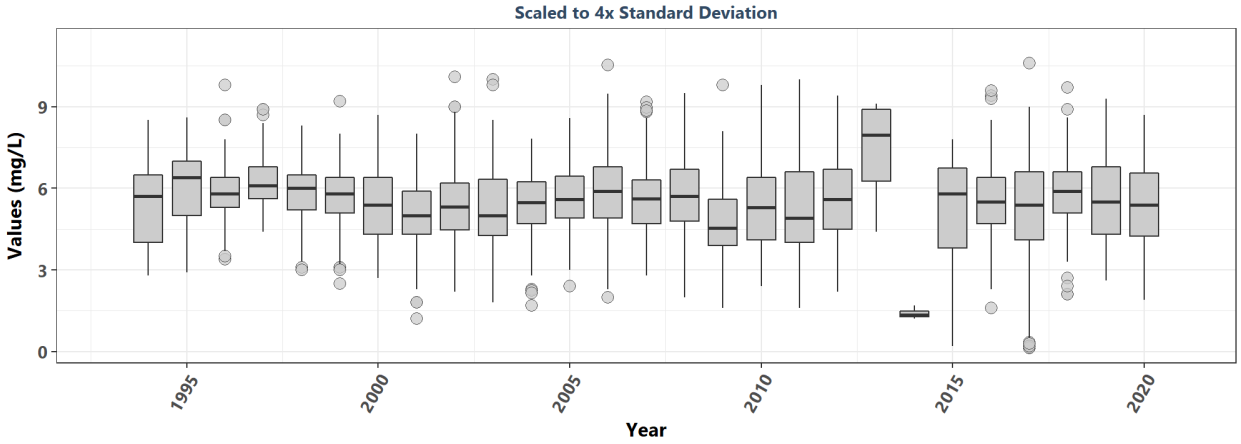

Cape Haze Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
2010
2012
2014
2016
2018
2020
Year

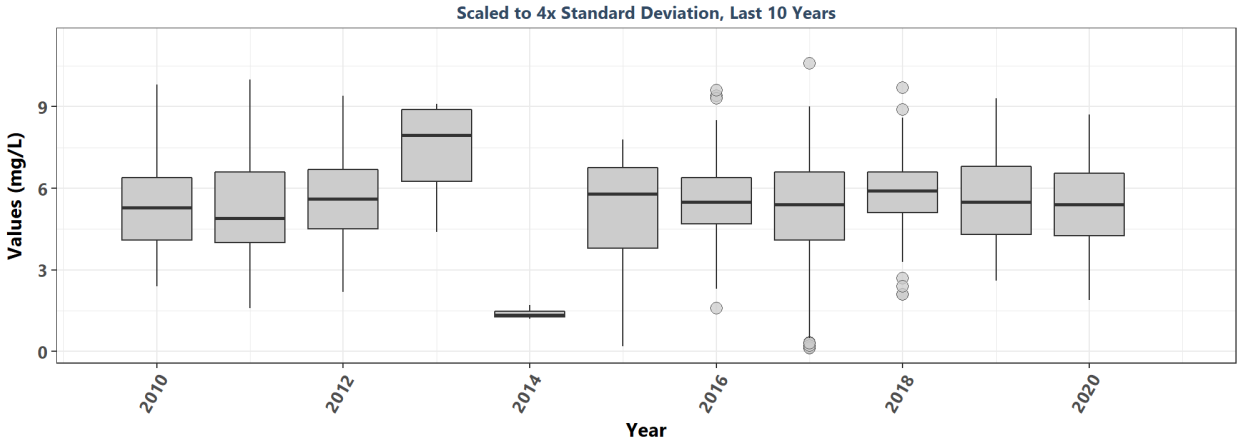

Cape Haze Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
3
6
9
12
Month

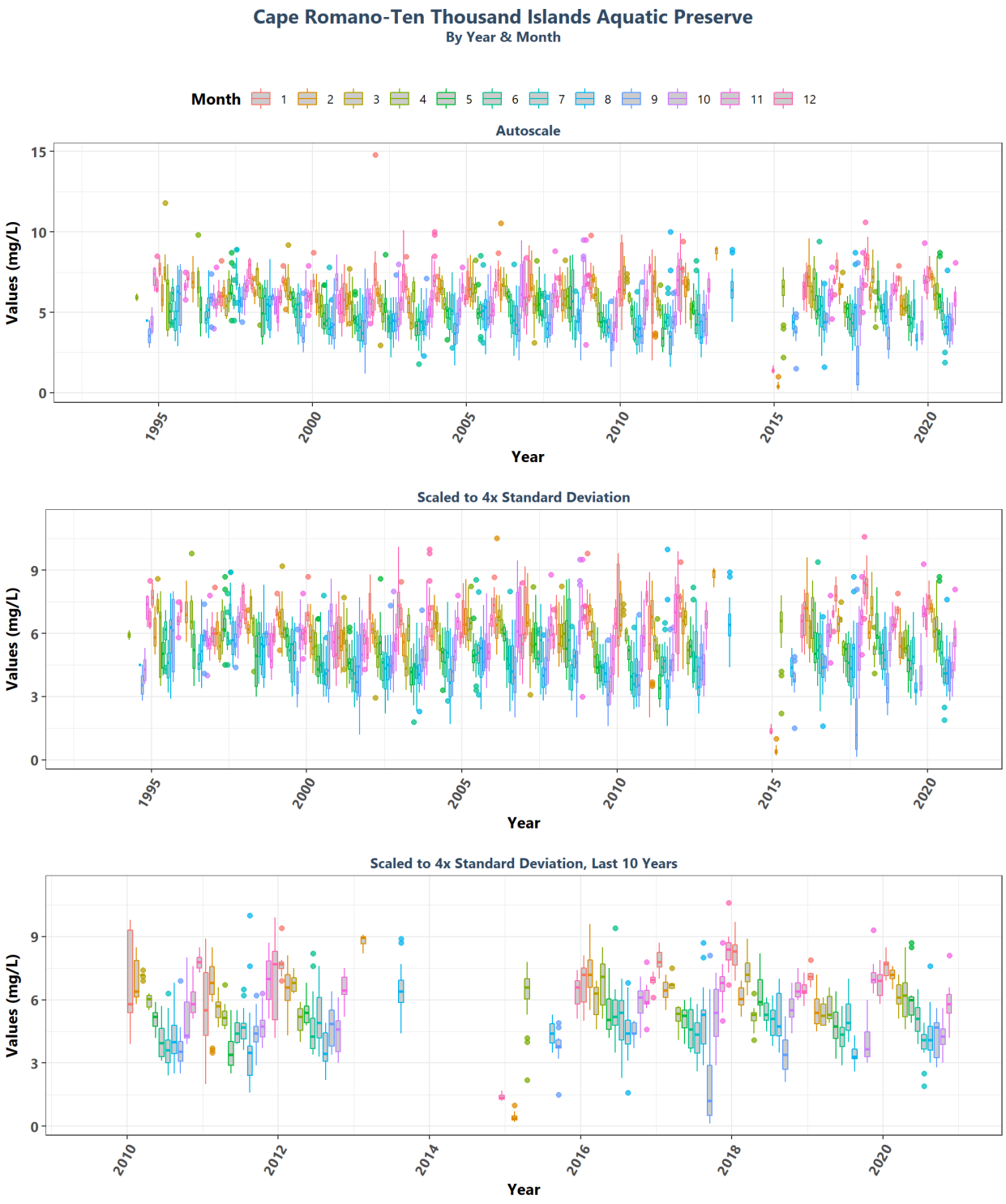
Cape Romano-Ten Thousand Islands Aquatic Preserve

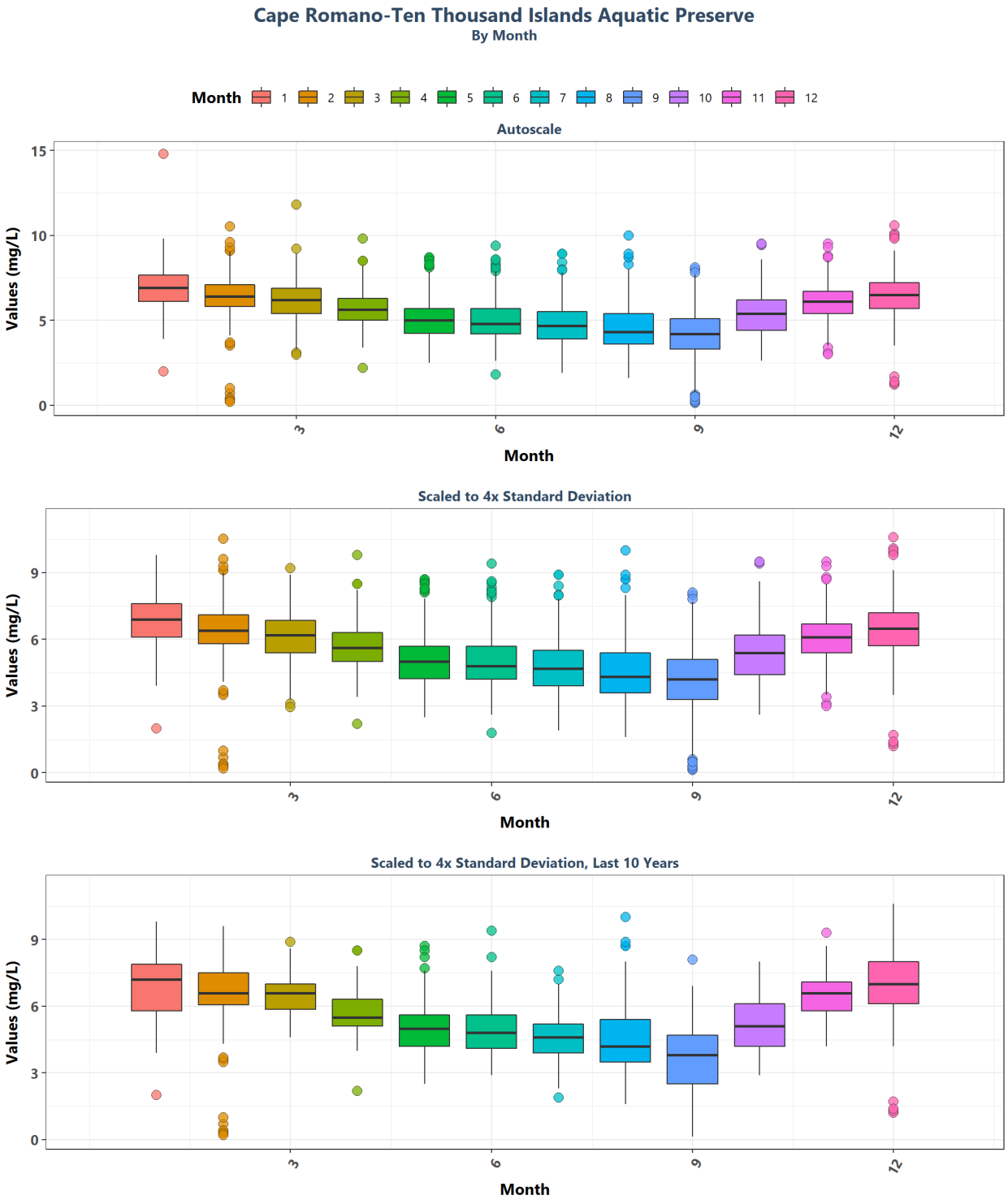
By Year





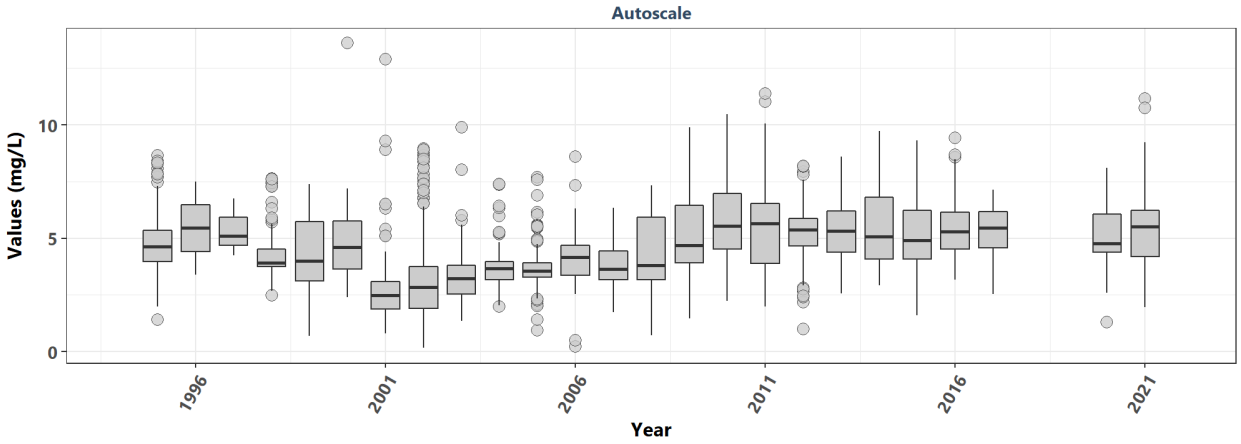


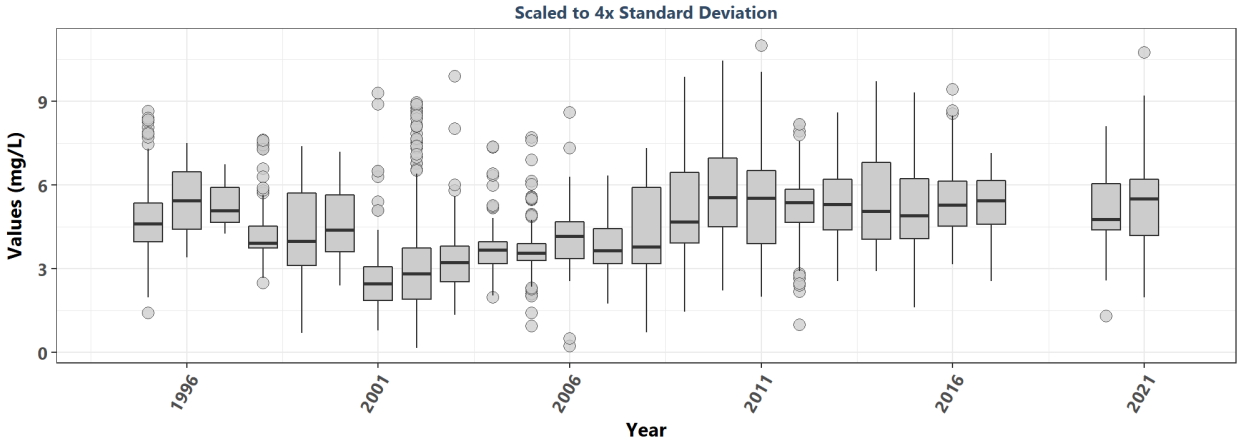

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

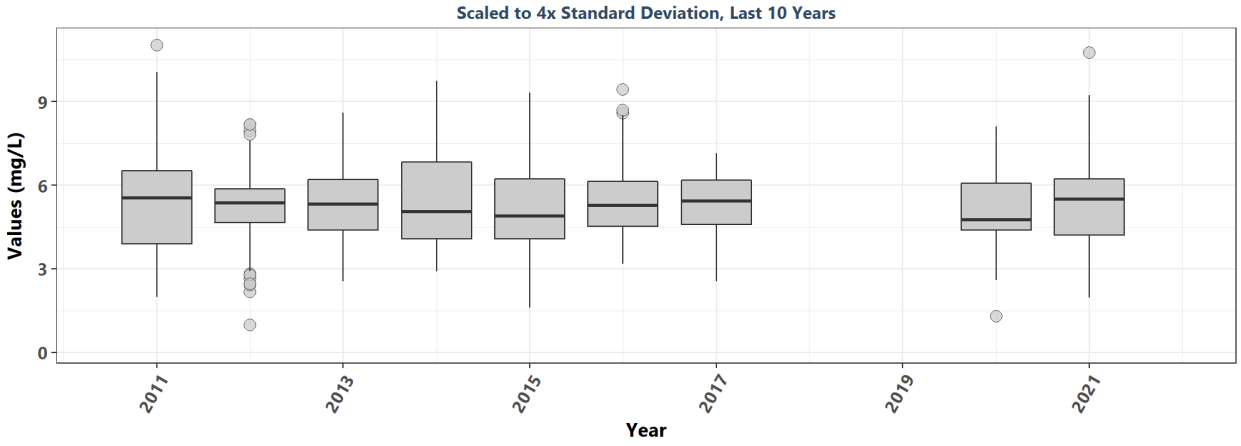

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
9
6
3
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
9
6
3
0
3
6
9
12
Month

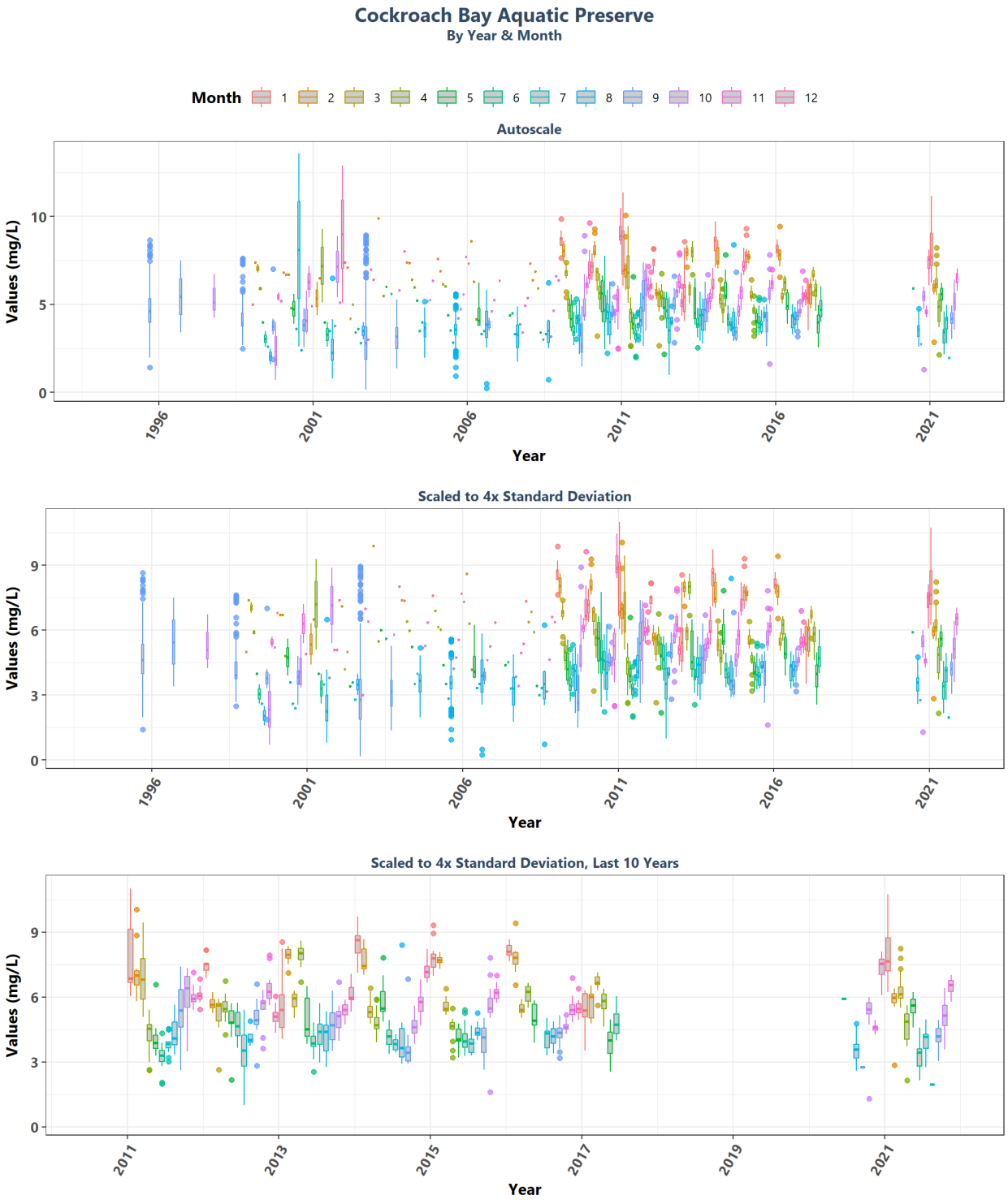
Cockroach Bay Aquatic Preserve

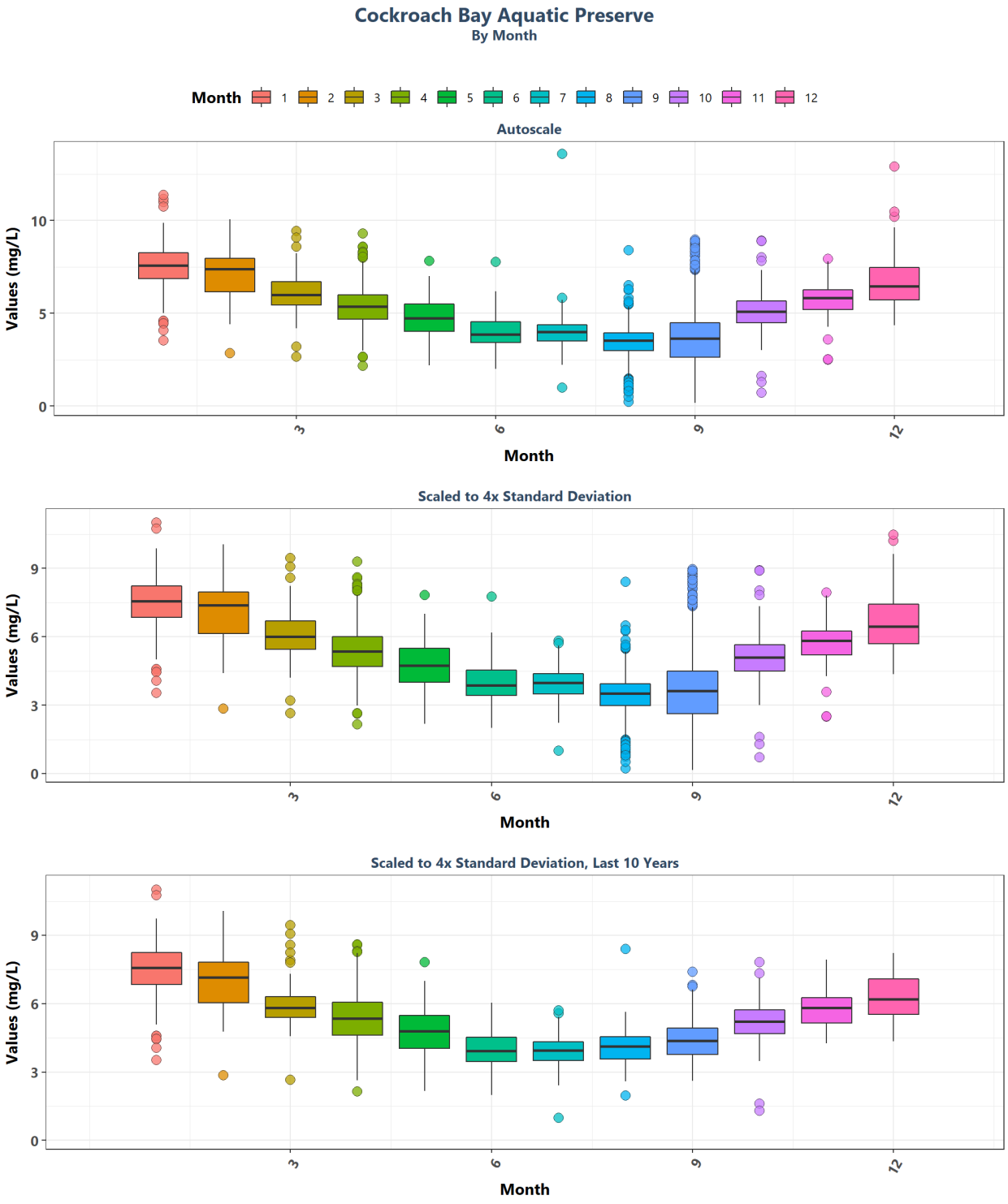
By Year





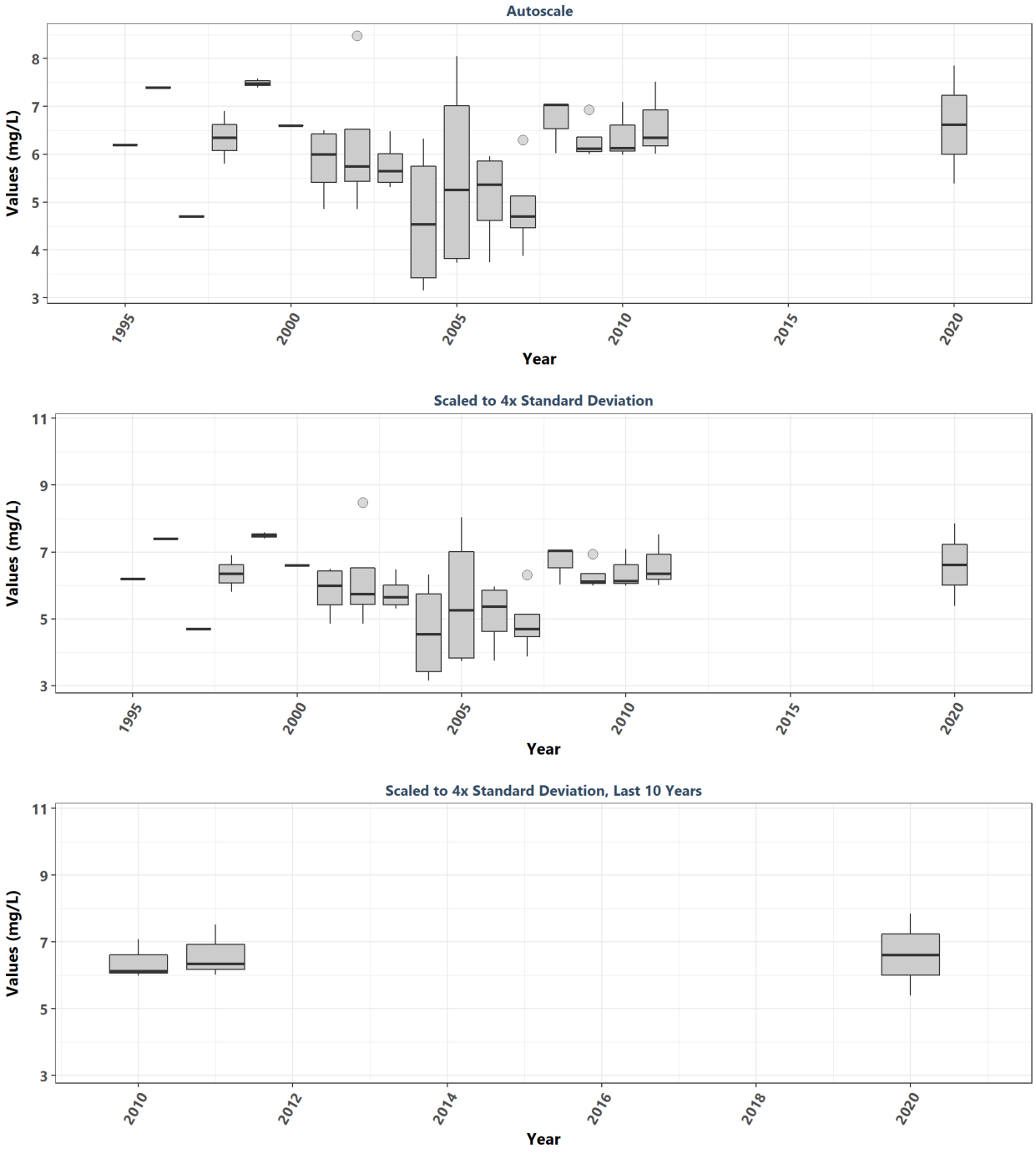


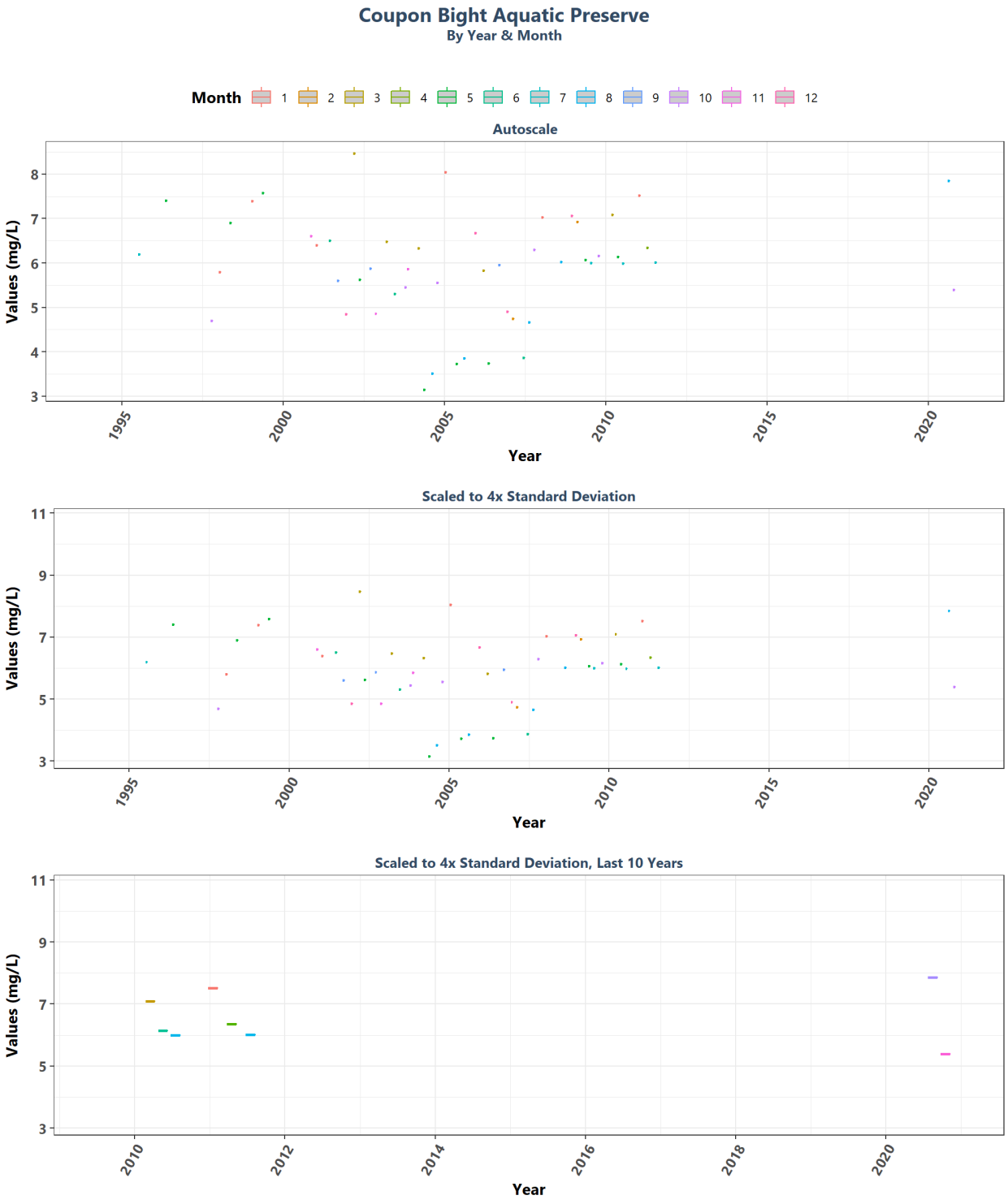

Cockroach Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
5
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0
3
6
9
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
3
6
9
2011
2013
2015
2017
2019
2021
Year

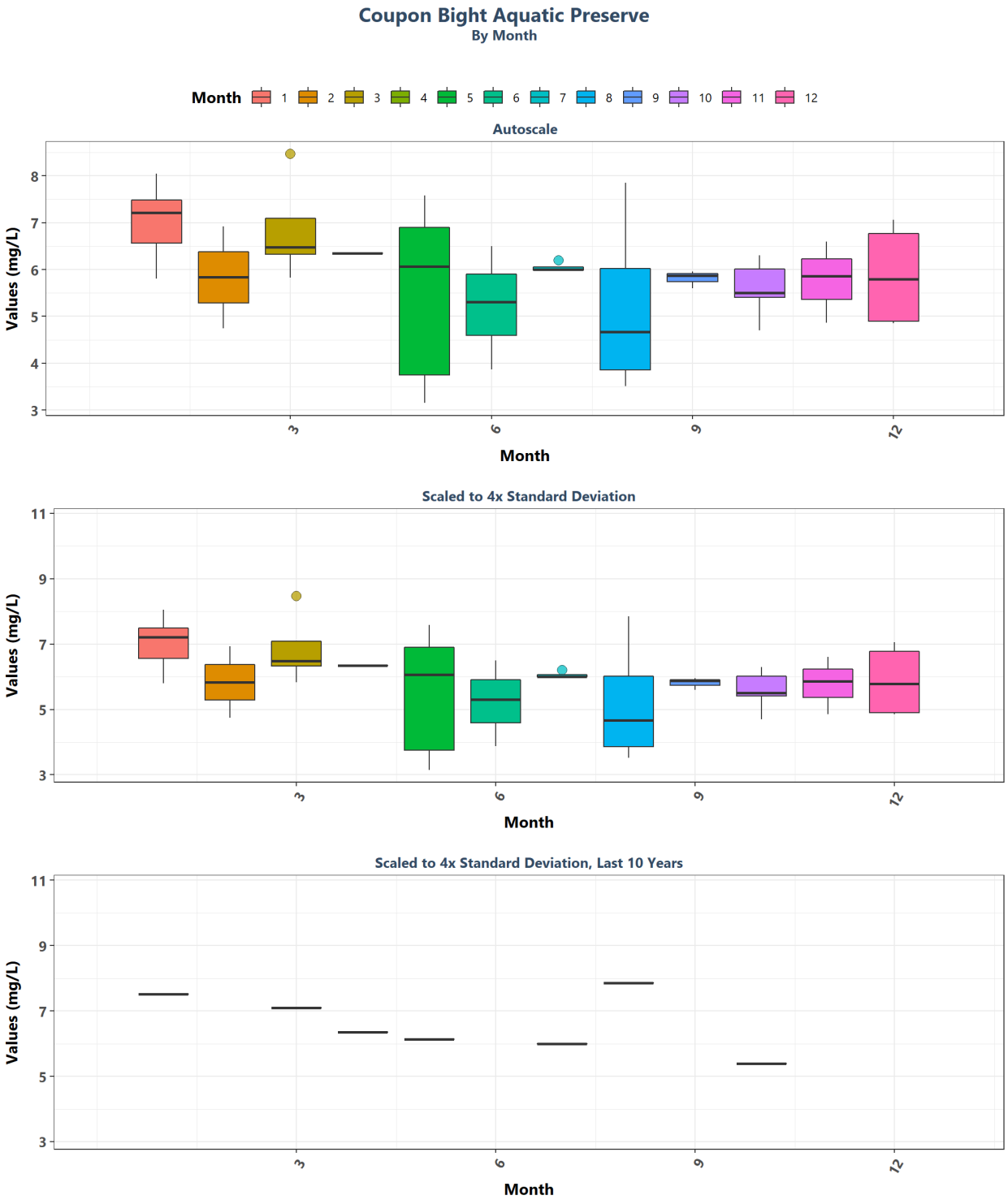

Cockroach Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

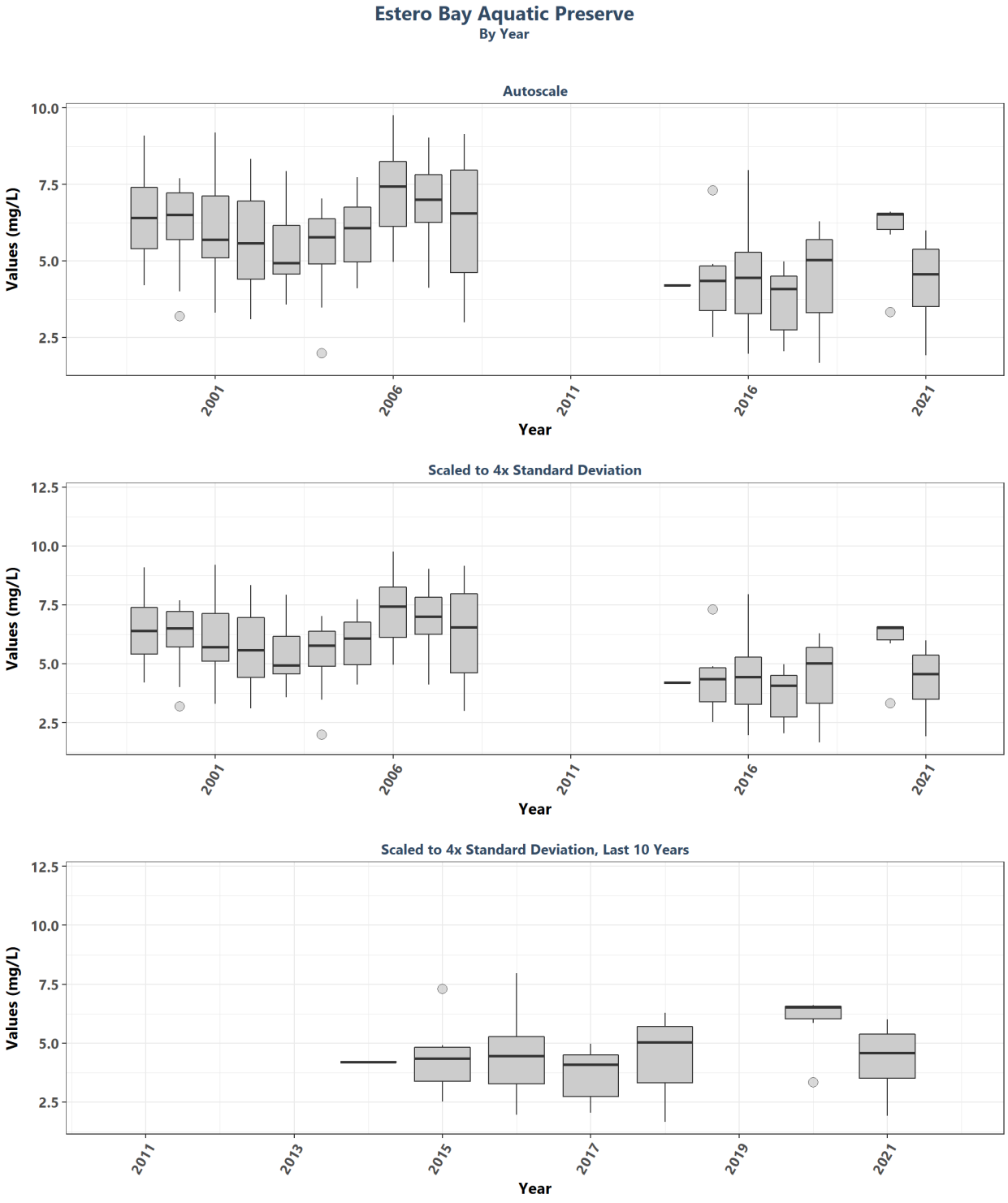
Coupon Bight Aquatic Preserve

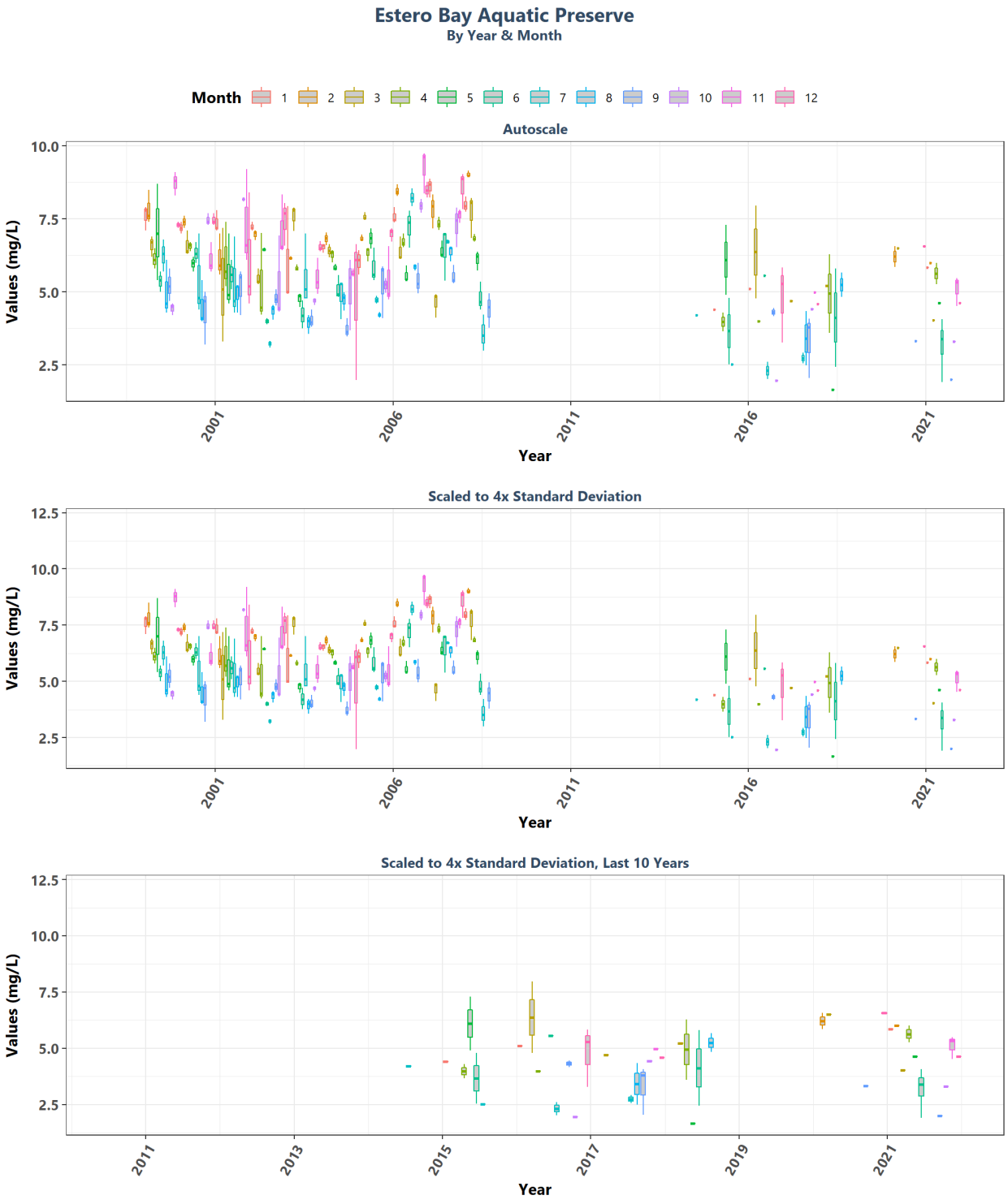
By Year

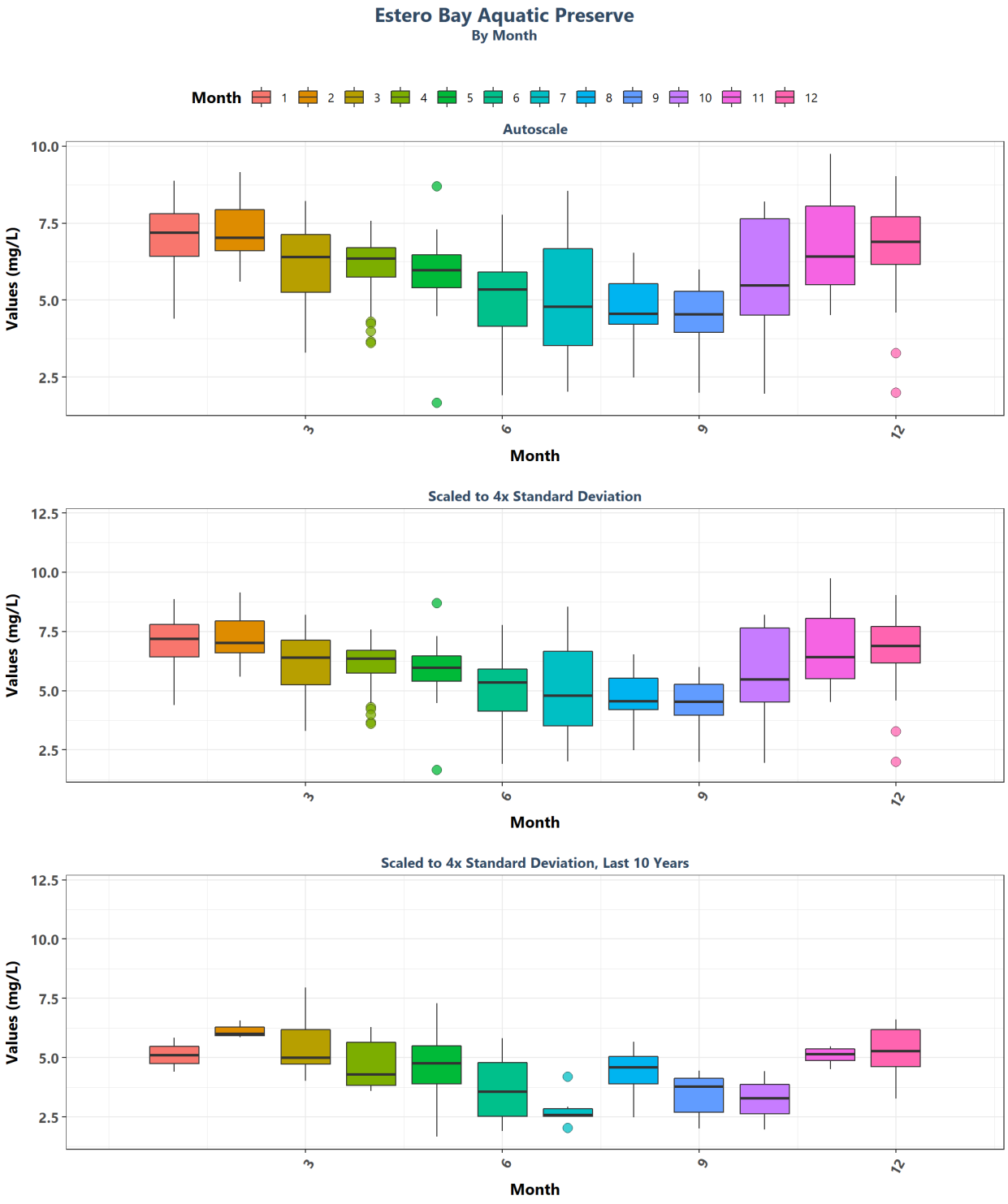



Coupon Bight Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year


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

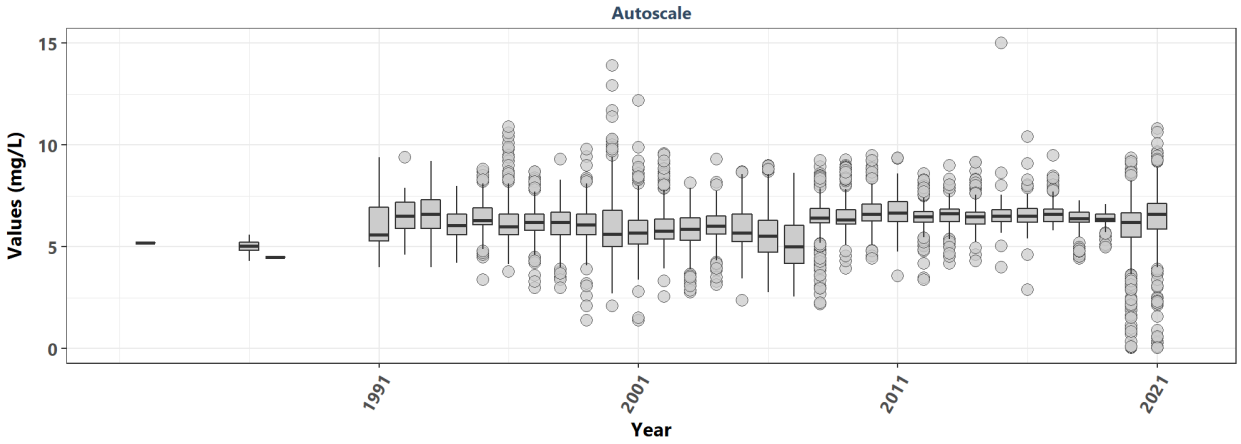

Estero Bay Aquatic Preserve
By Year
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year

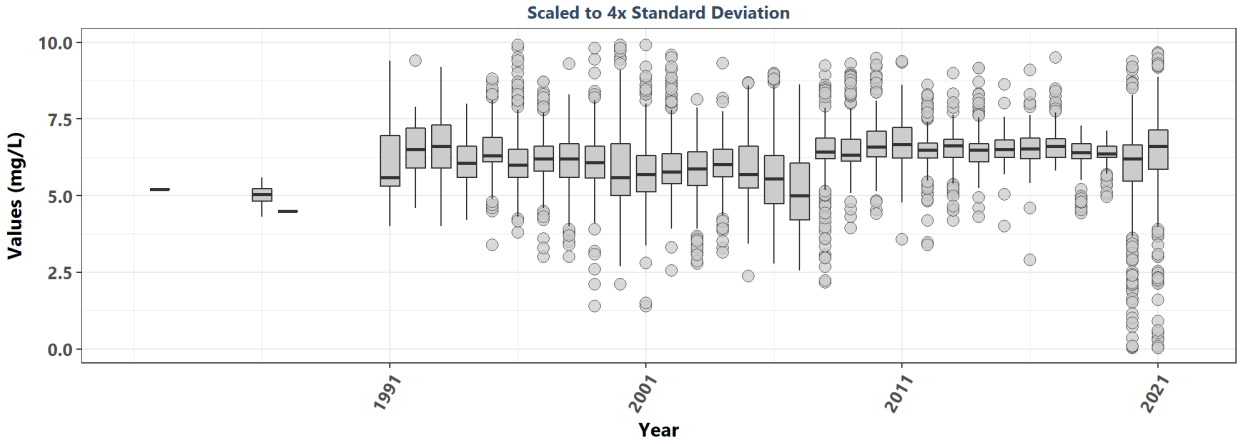

Estero Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year

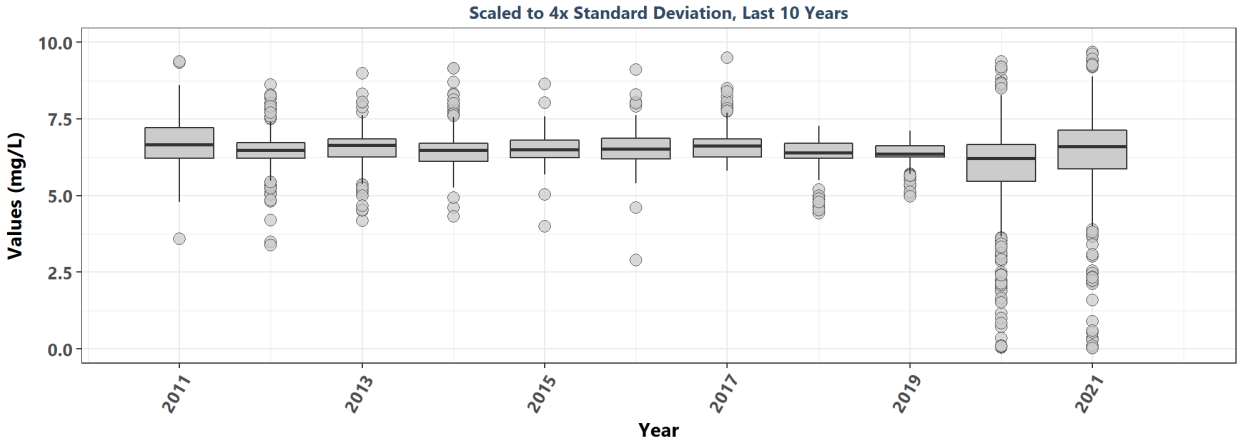

Estero Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month

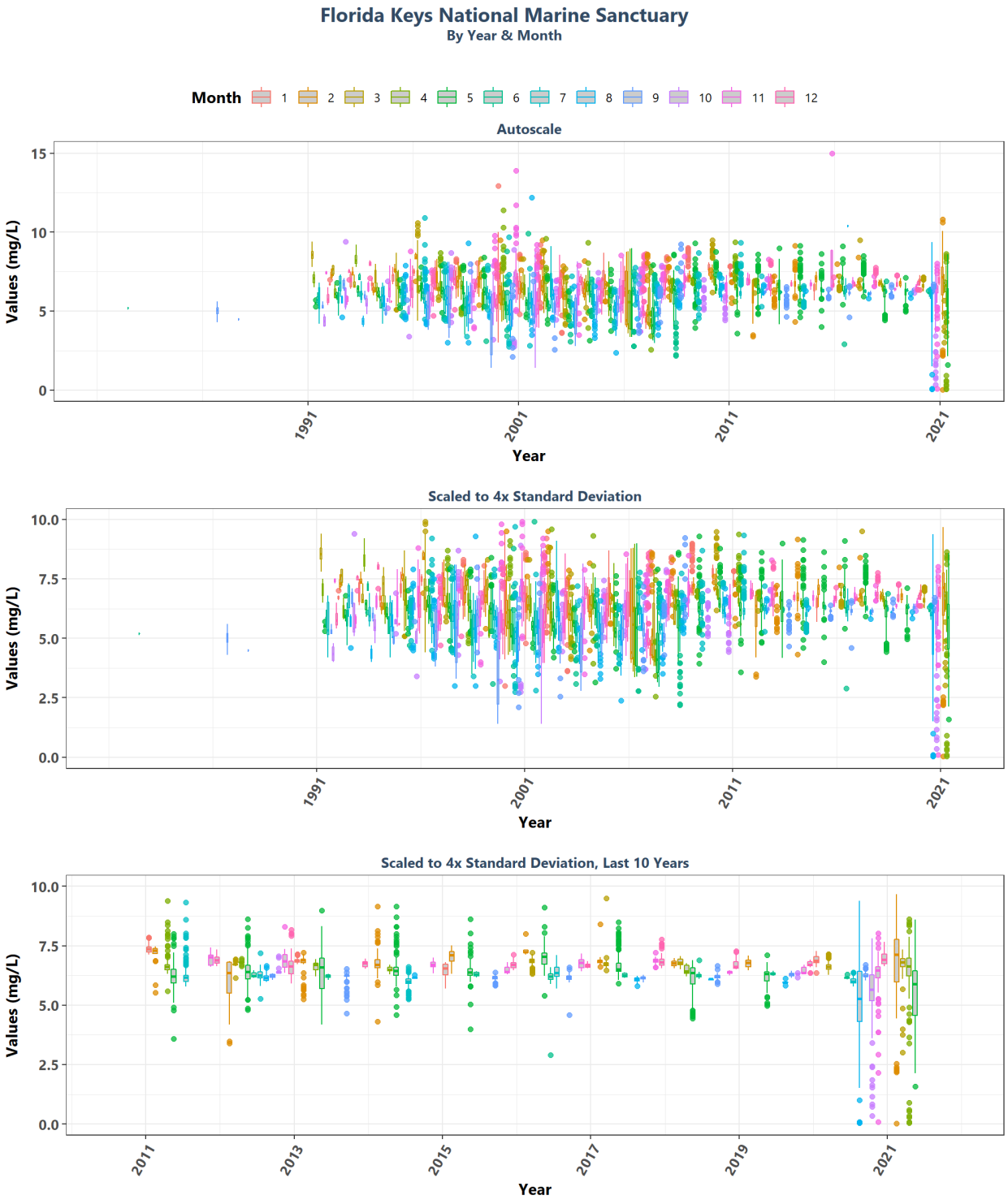
Florida Keys National Marine Sanctuary

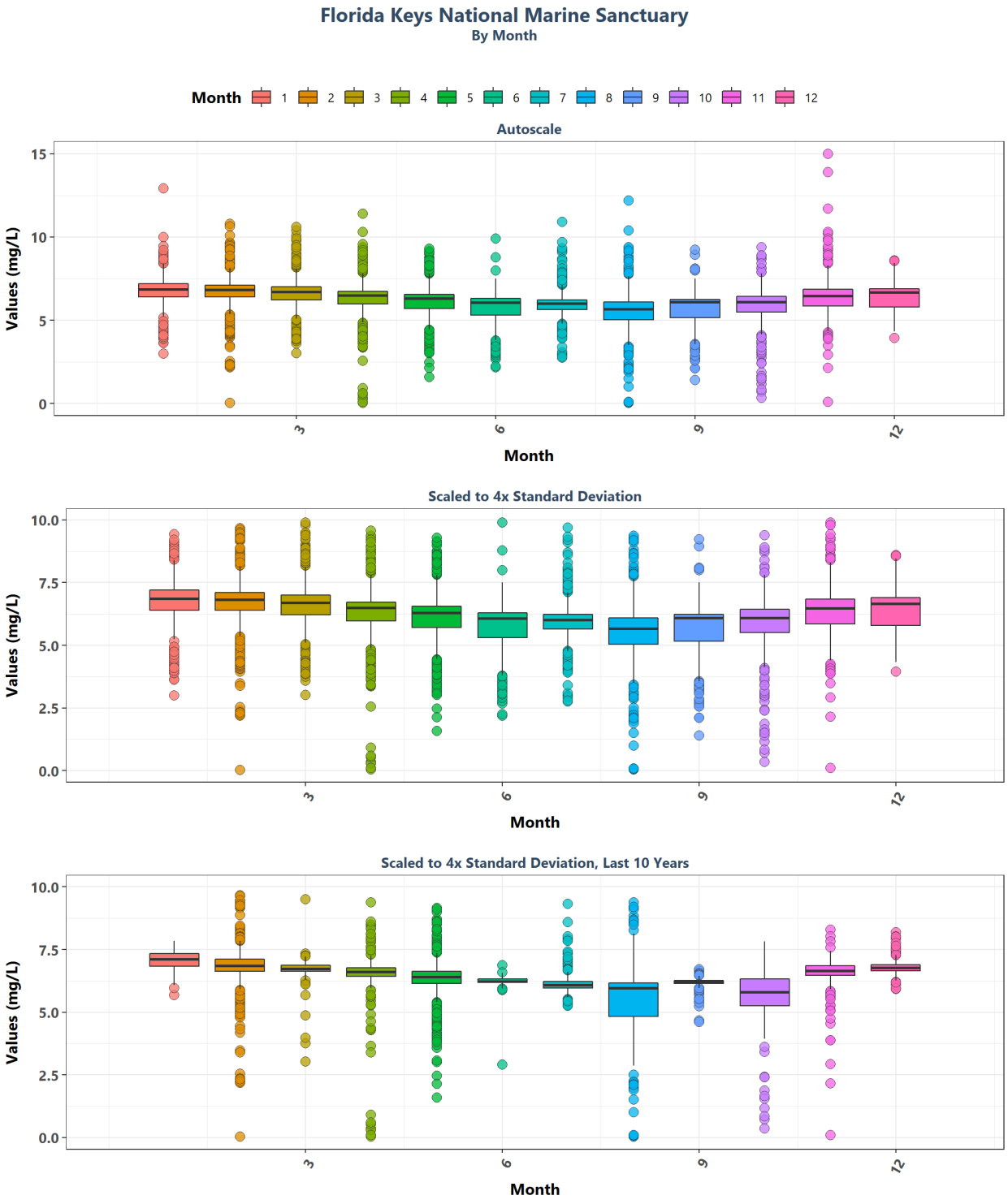
By Year





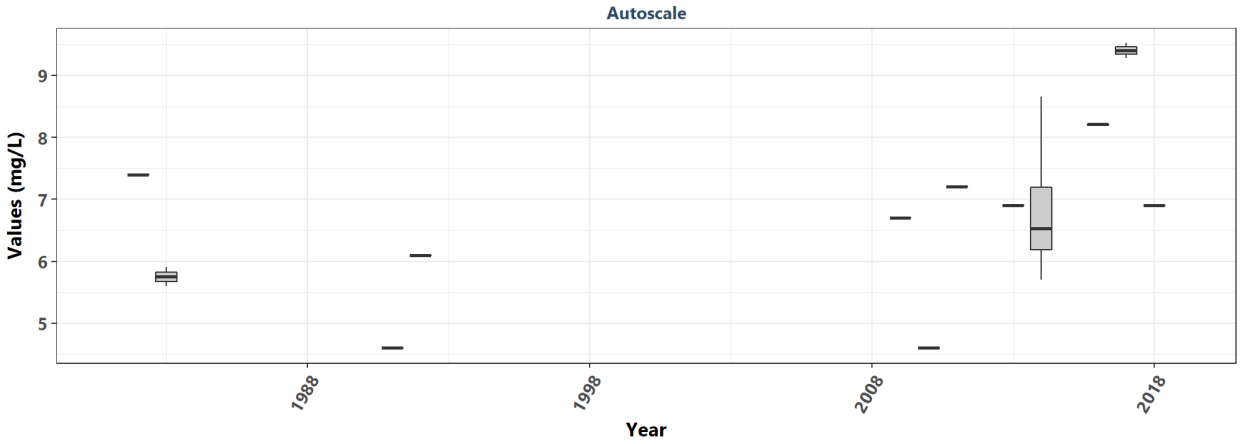


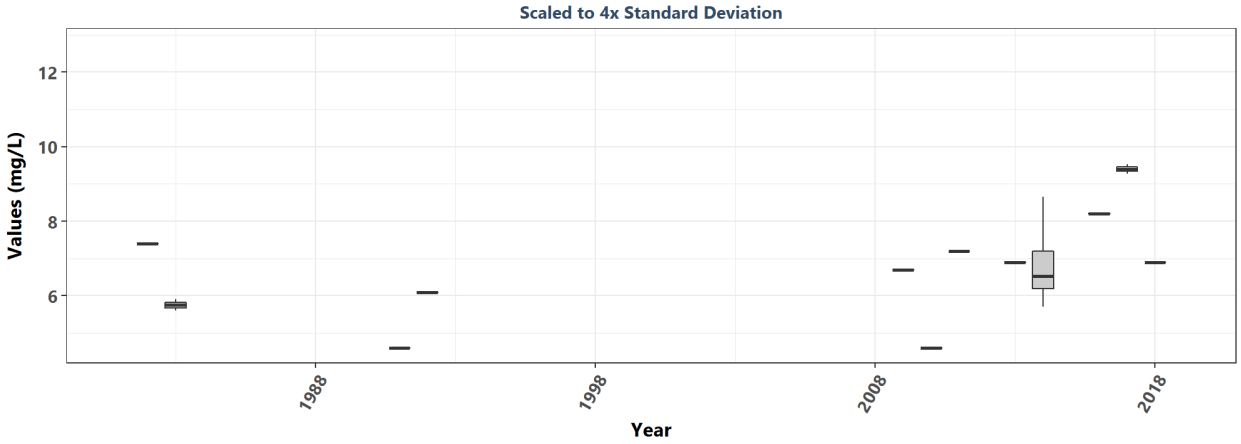

Florida Keys National Marine Sanctuary
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15
10
5
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
2011
2013
2015
2017
2019
2021
Year

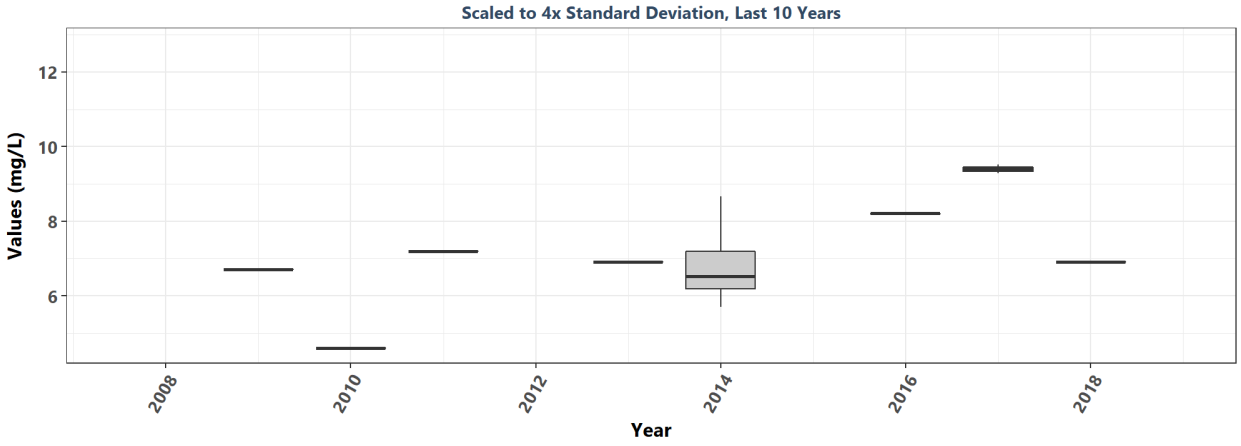

Florida Keys National Marine Sanctuary
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month

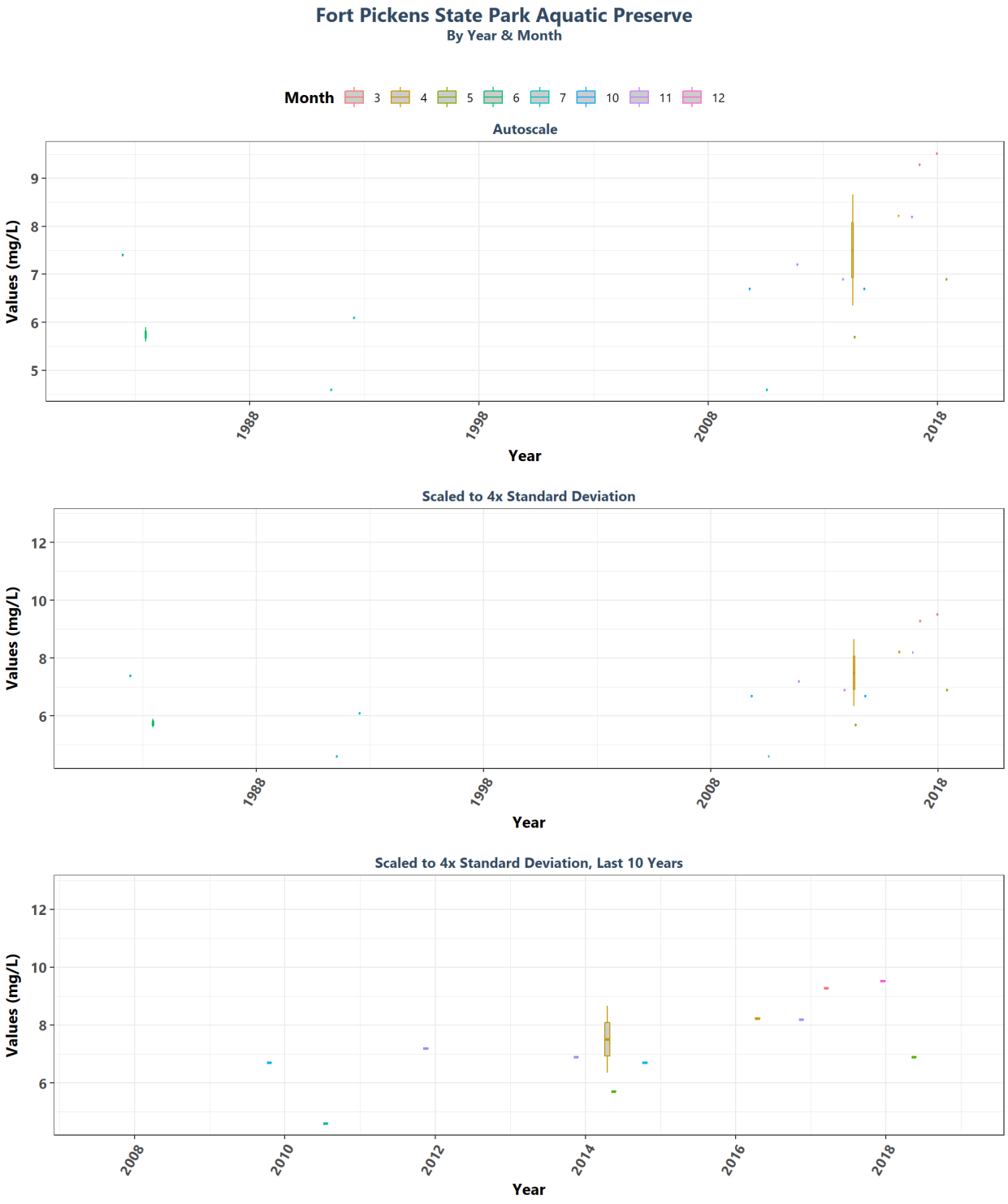
Fort Pickens State Park Aquatic Preserve

By Year



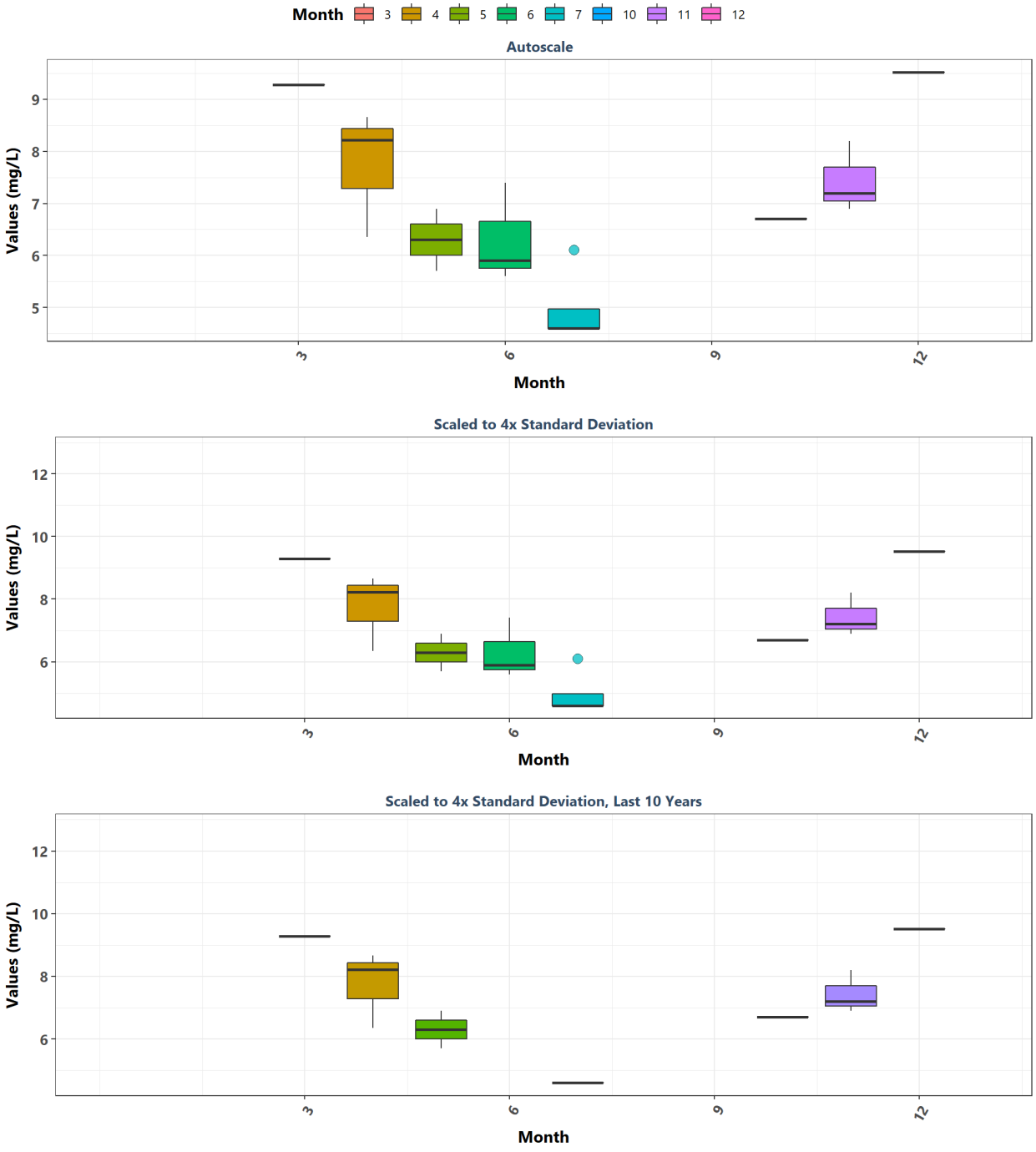


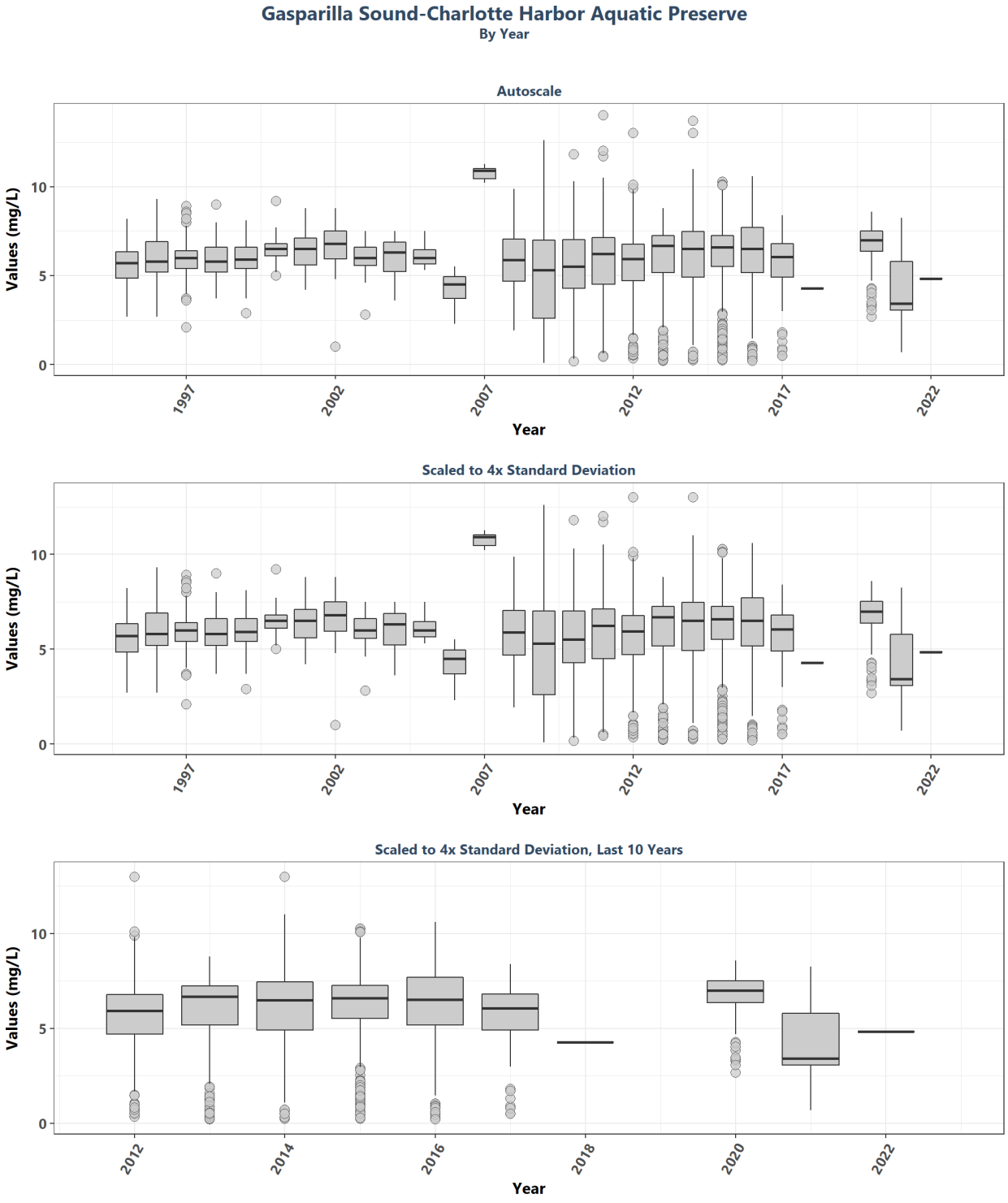


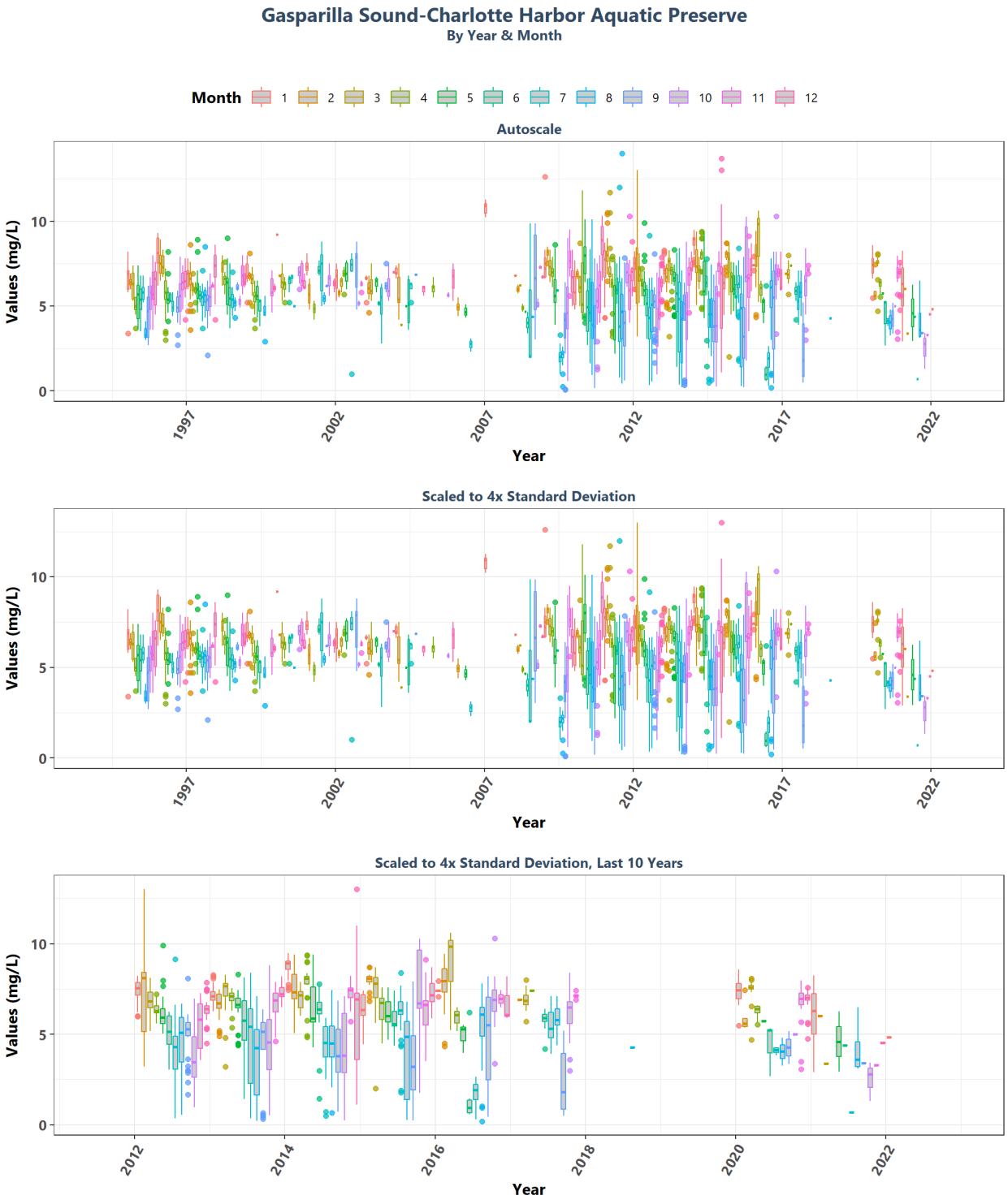

Fort Pickens State Park Aquatic Preserve
By Year & Month
Month 3 4 5 6 7 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

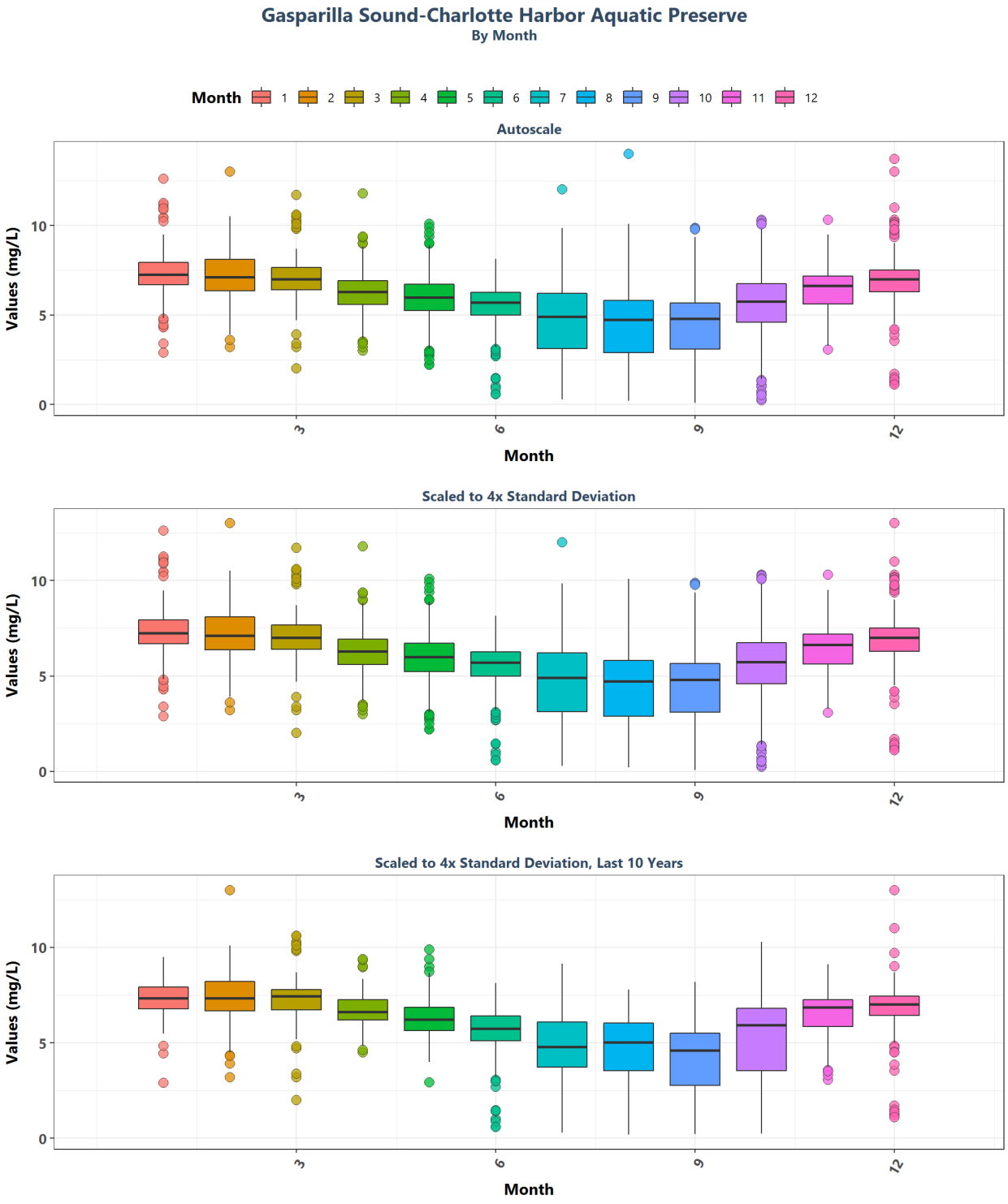
Fort Pickens State Park Aquatic Preserve

By Month

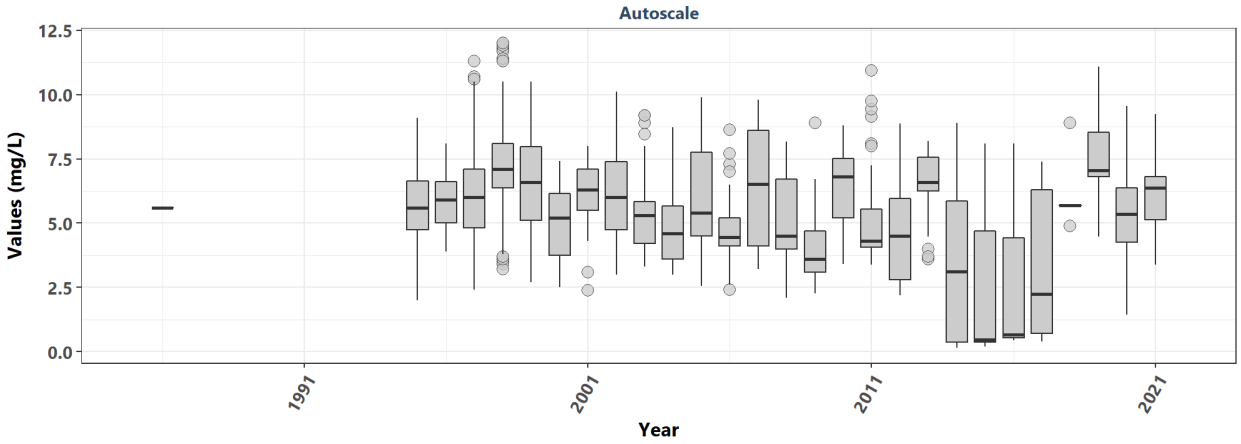


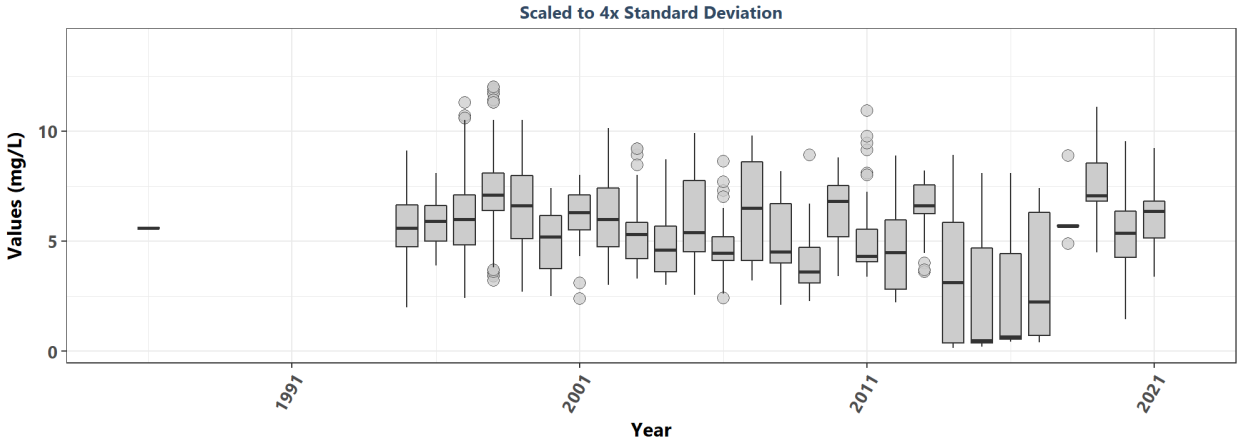

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year
Autoscale
Values (mg/L)
0
5
10
1997
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
1997
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
2012
2014
2016
2018
2020
2022
Year

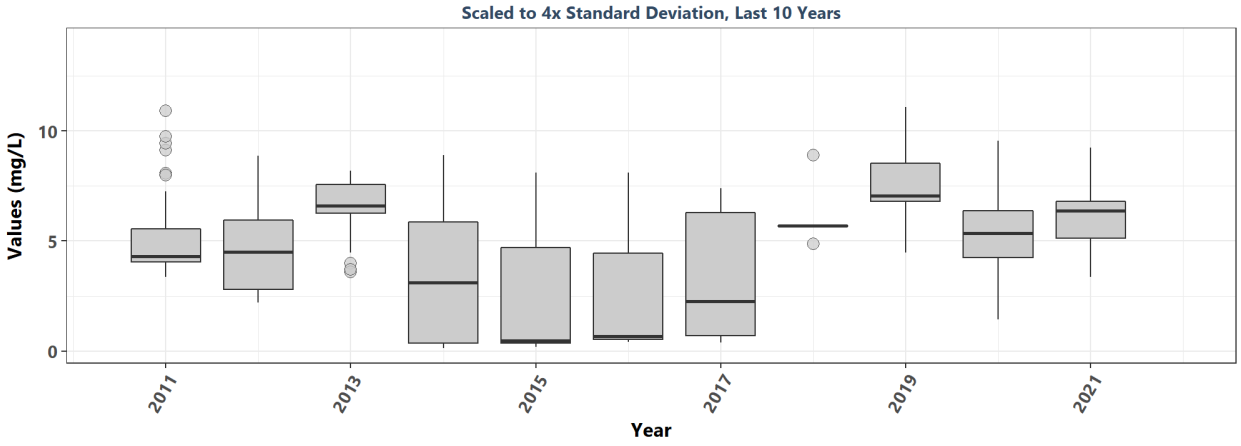

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
5
10
1997
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
1997
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
2012
2014
2016
2018
2020
2022
Year

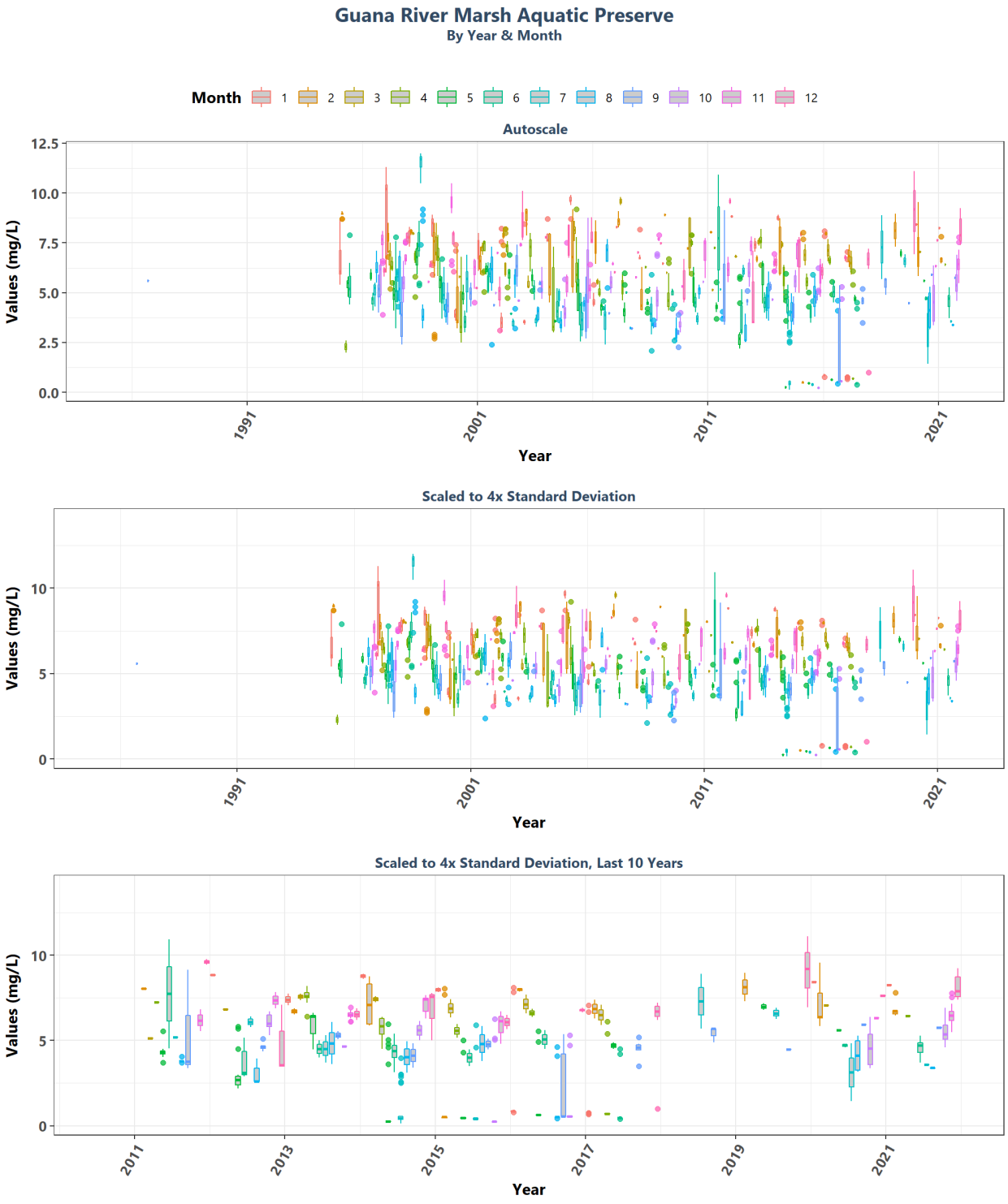

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
5
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
3
6
9
12
Month

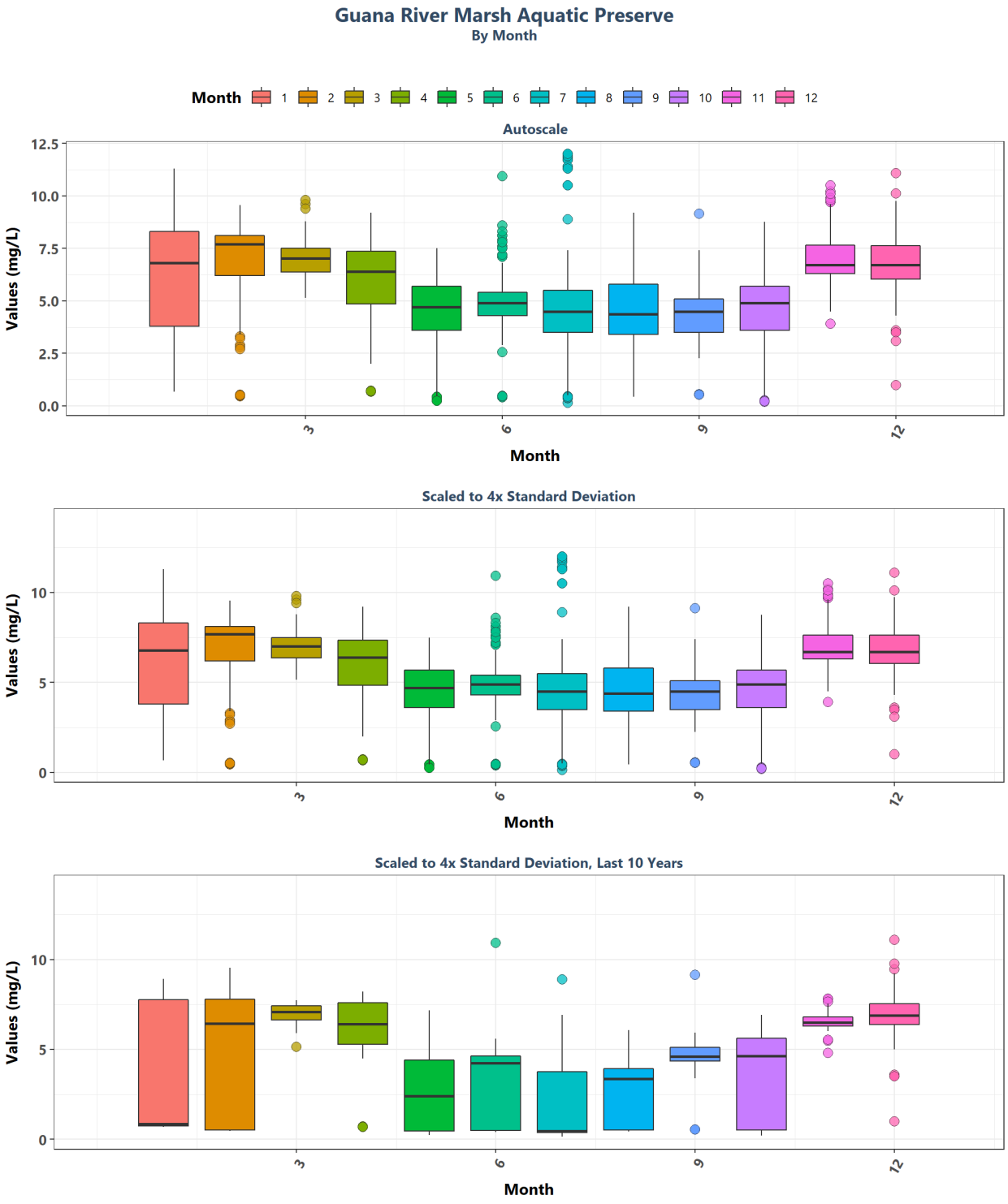
Guana River Marsh Aquatic Preserve
By Year





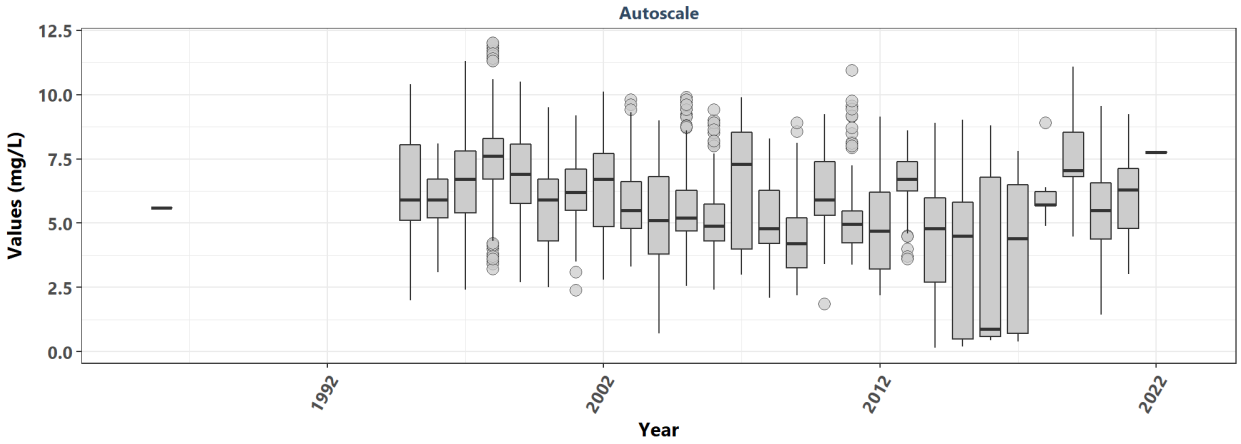


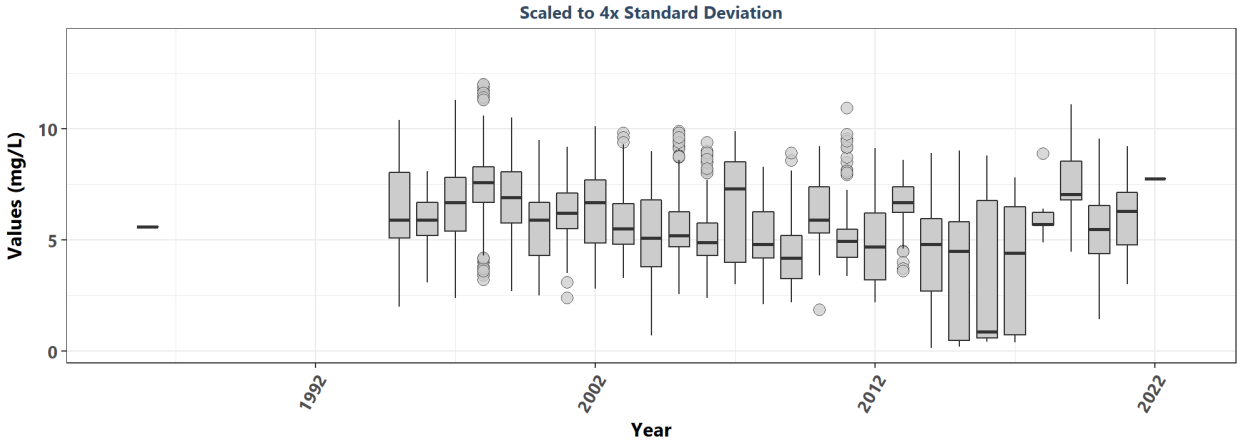

Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2011
2013
2015
2017
2019
2021
Year

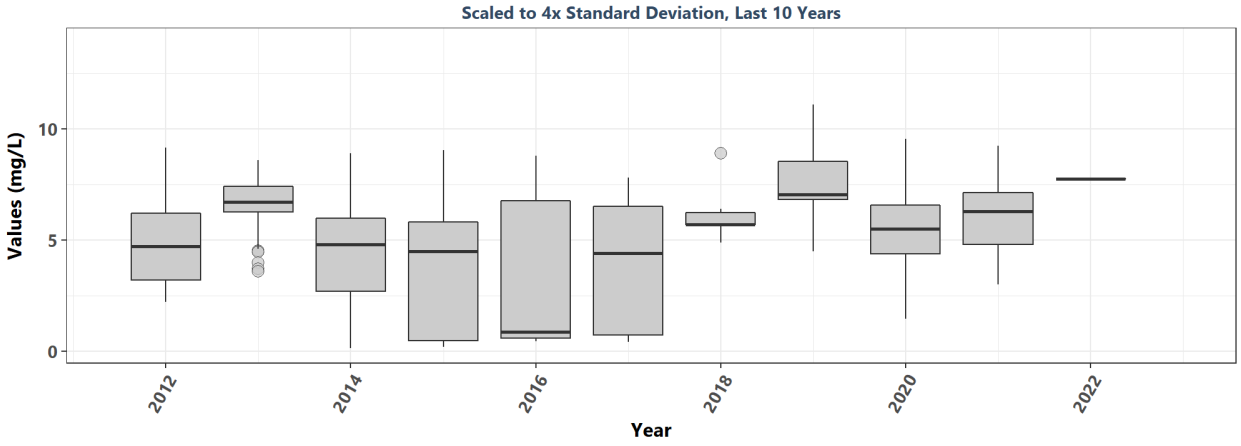

Guana River Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

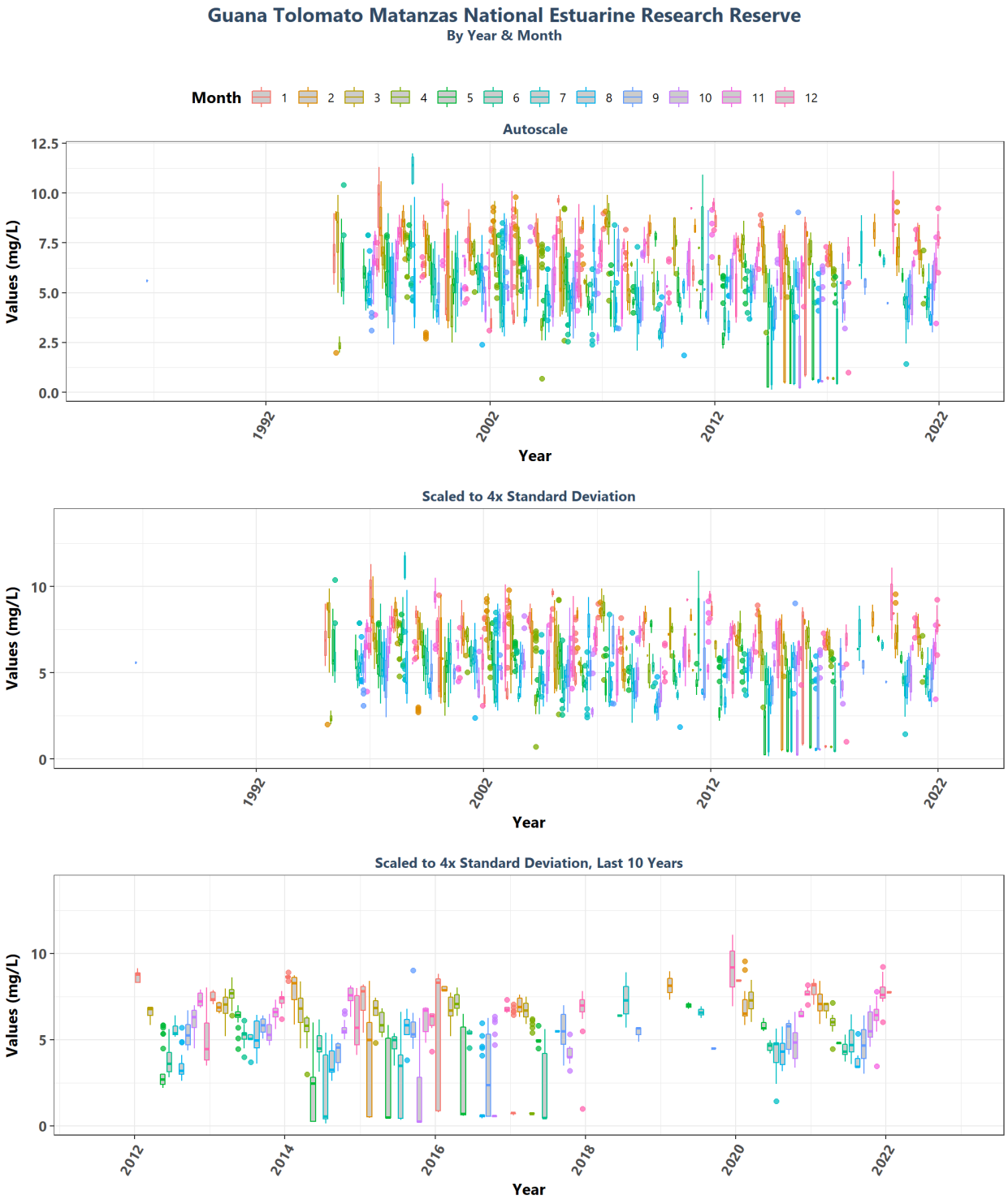
Guana Tolomato Matanzas National Estuarine Research Reserve

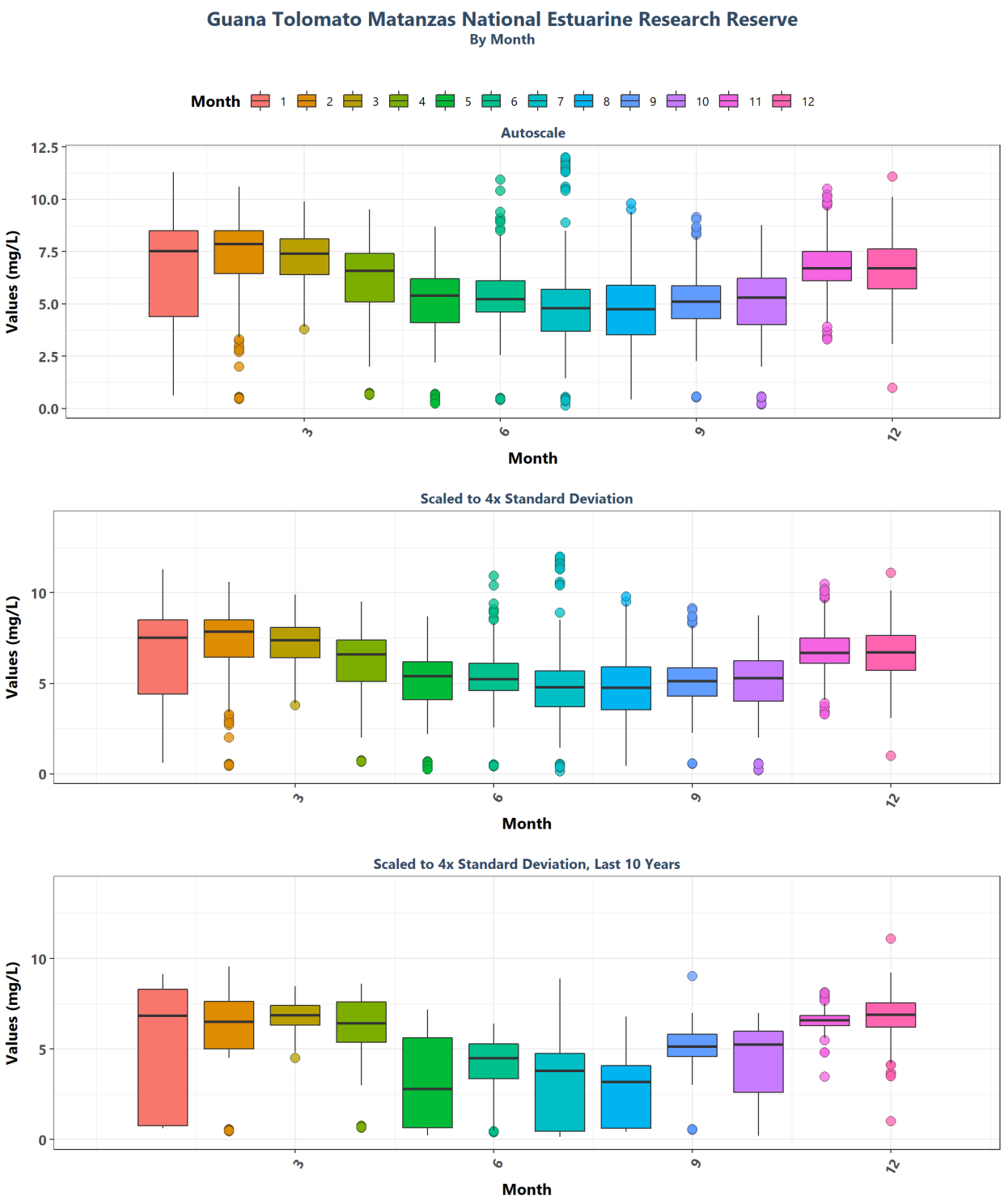
By Year

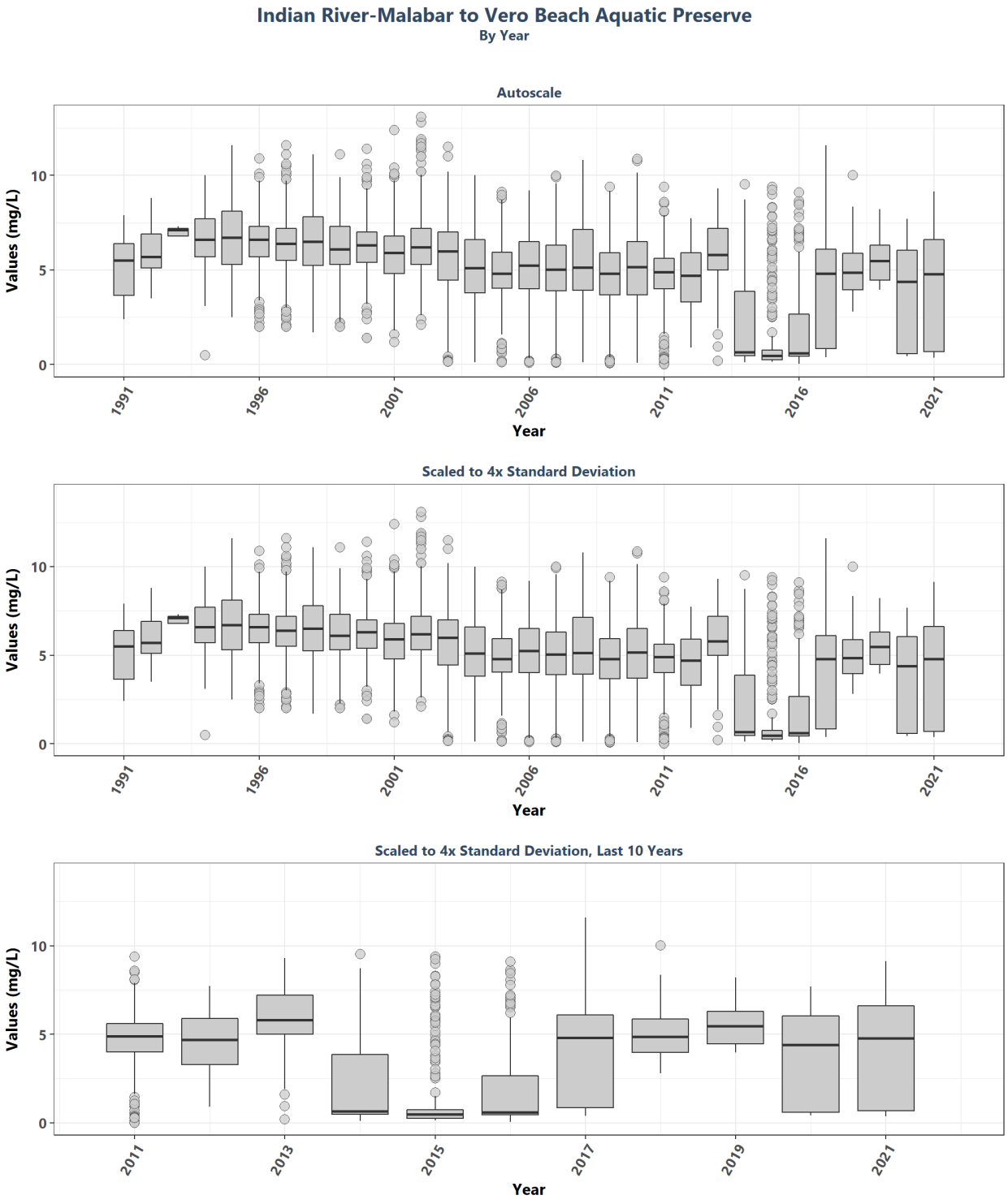


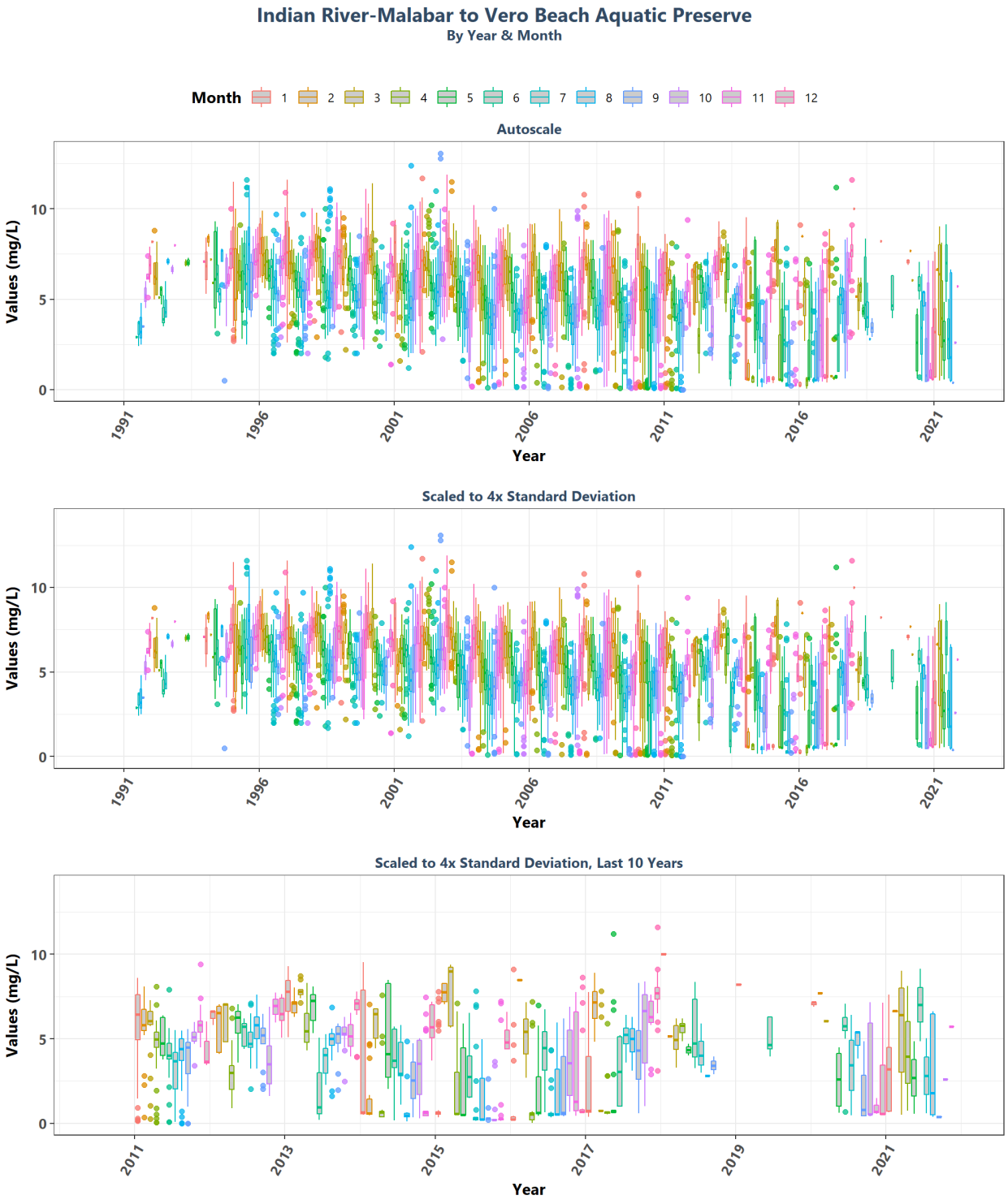


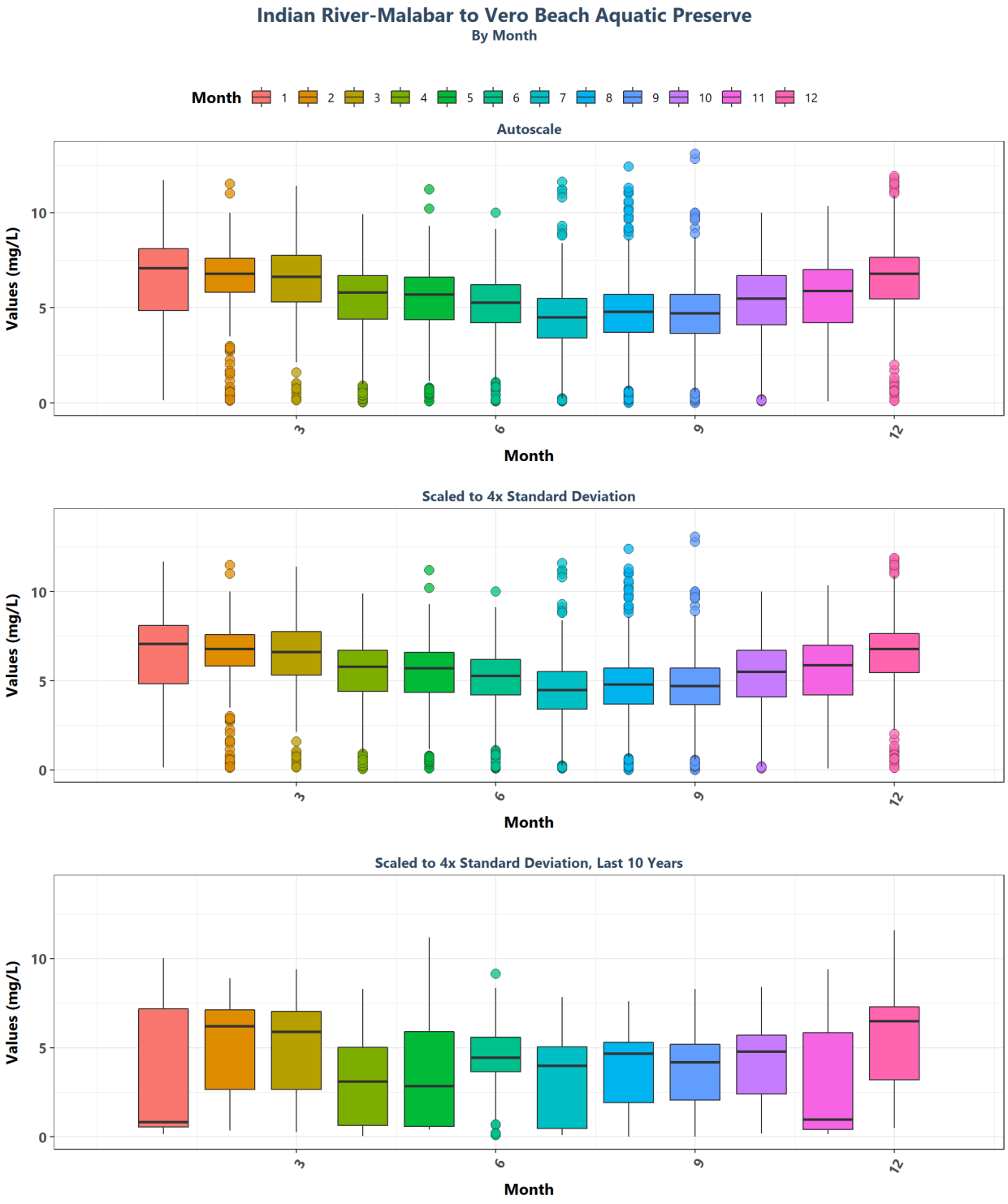



Guana Tolomato Matanzas National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2012
2014
2016
2018
2020
2022
Year


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)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

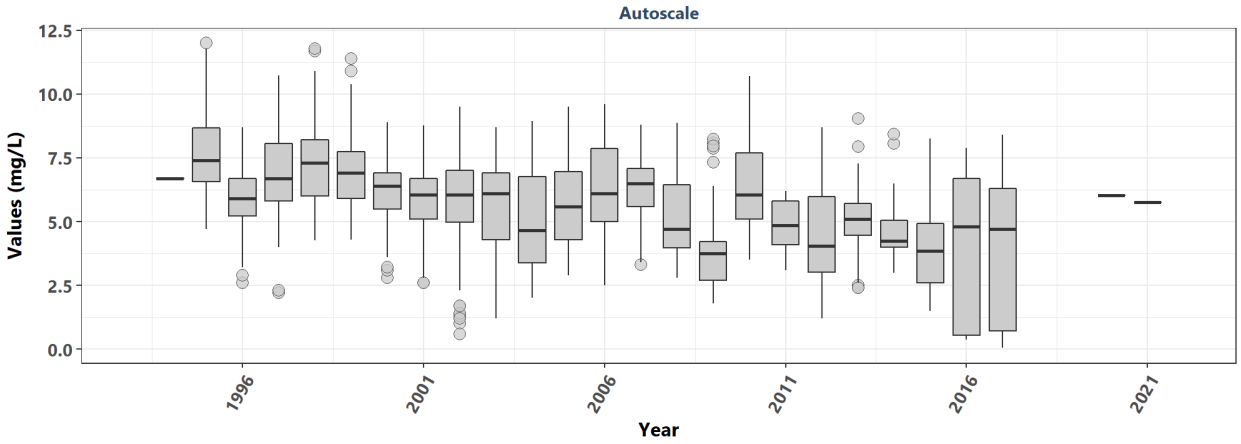

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year
Autoscale
Values (mg/L)
0
5
10
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
2011
2013
2015
2017
2019
2021
Year

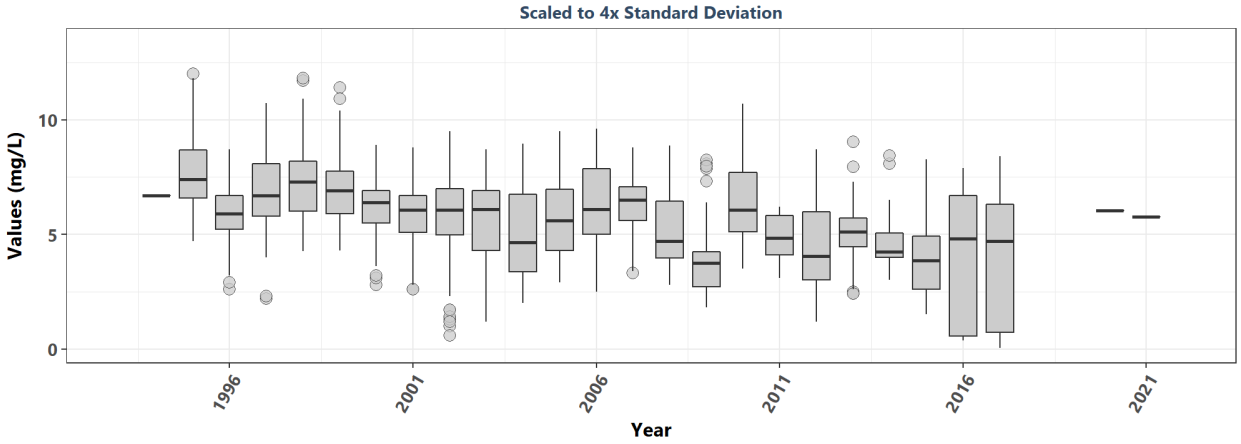

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

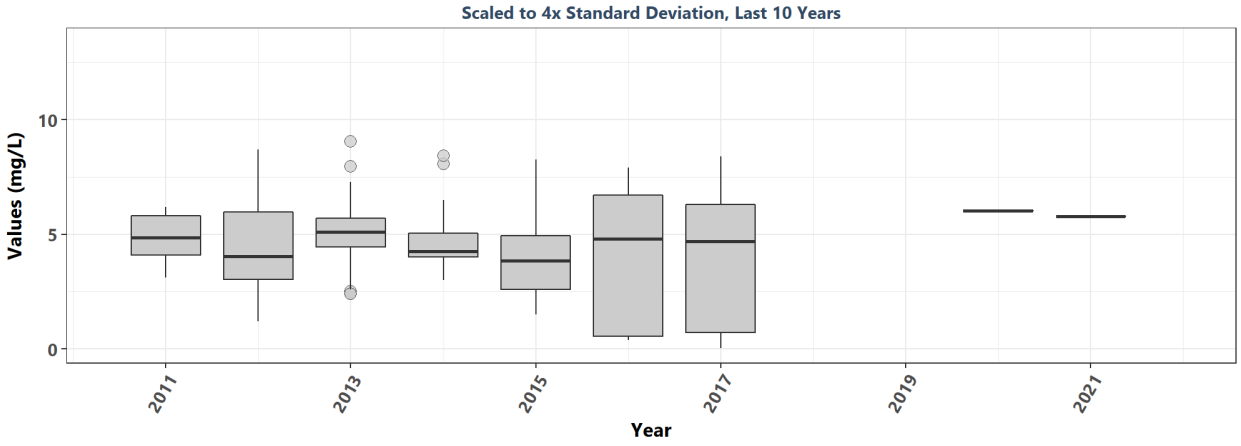

Indian River-Malabar to Vero Beach Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
0
5
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
0
5
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
0
5
10
3
6
9
12
Month

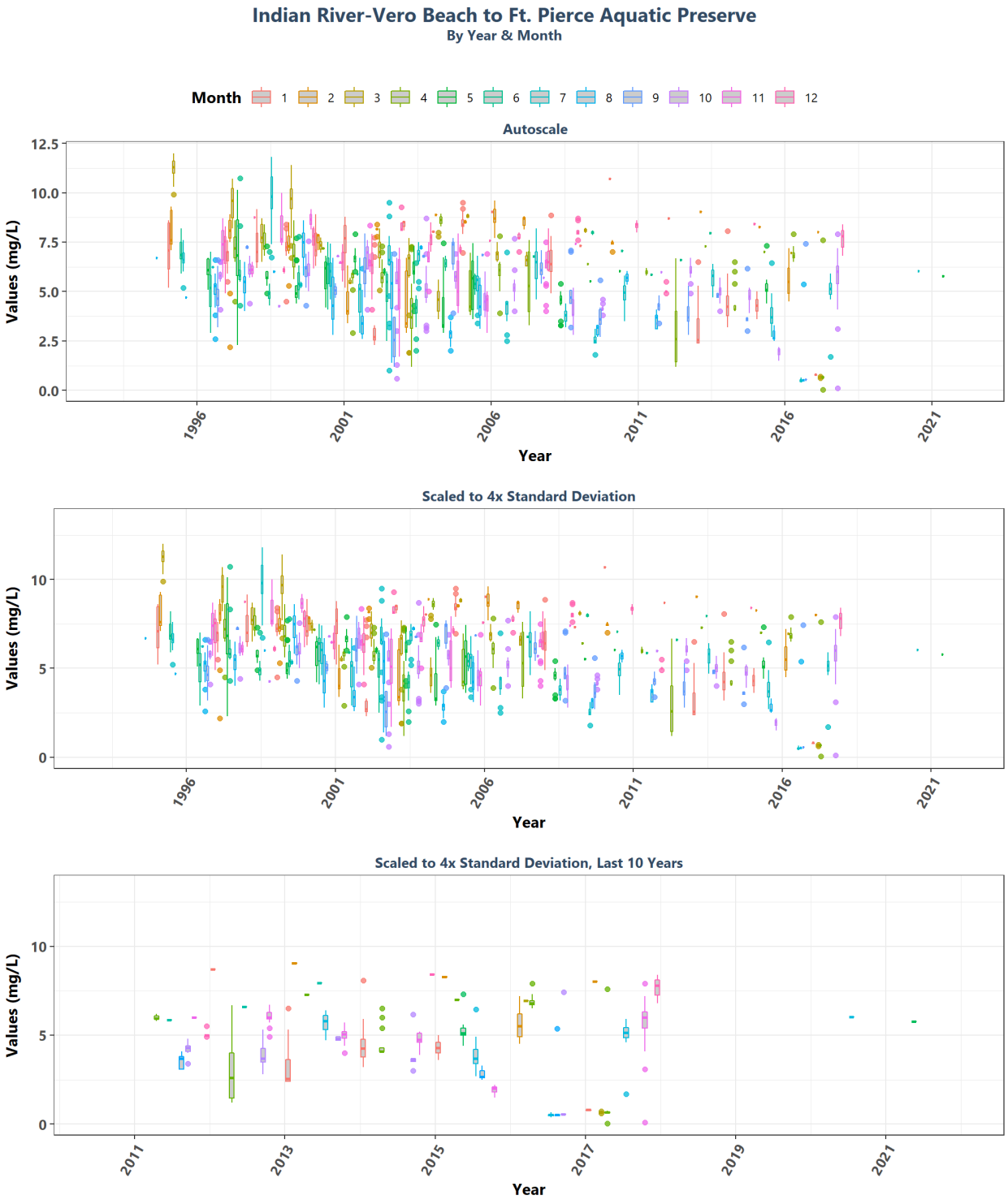
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

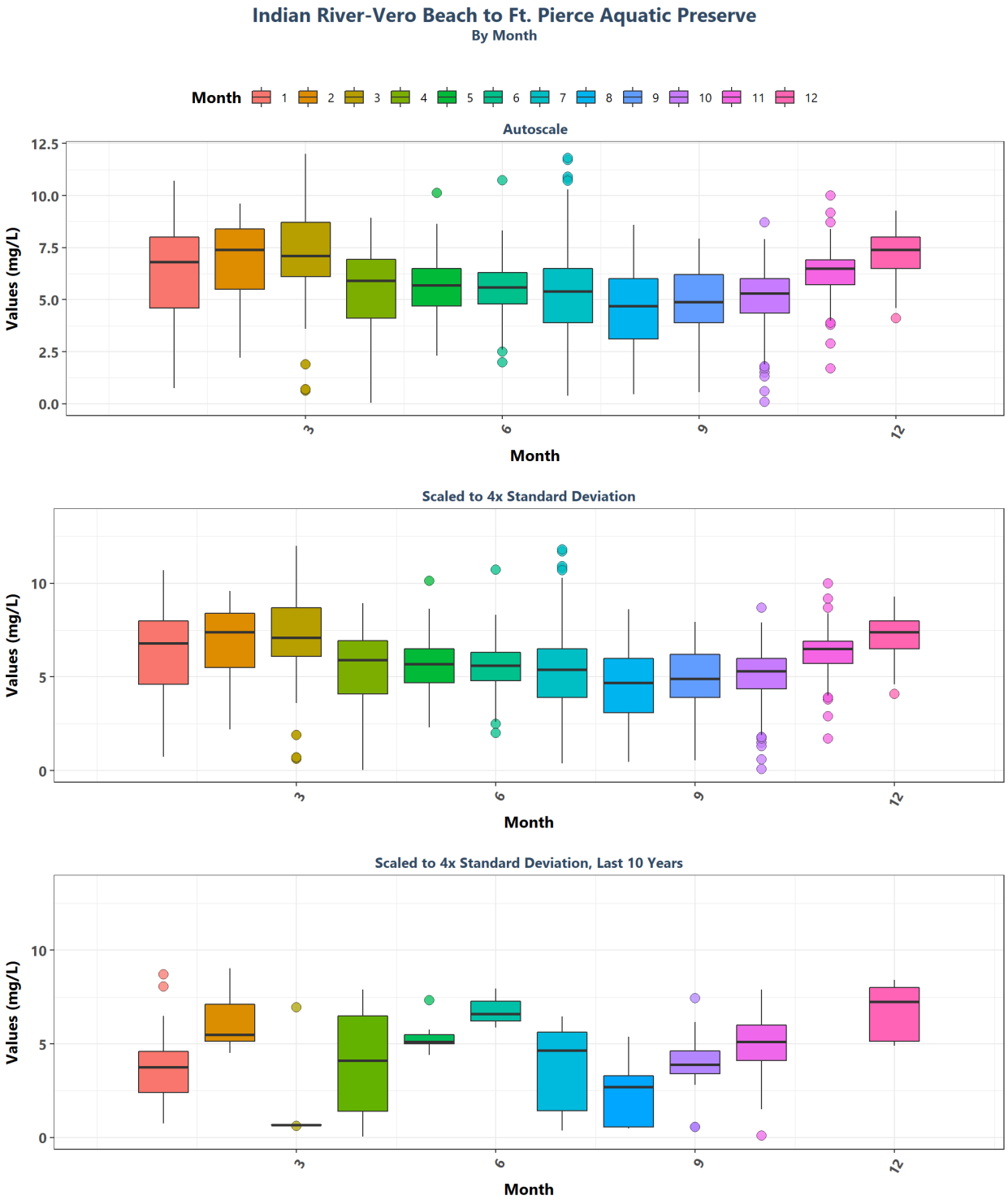
By Year





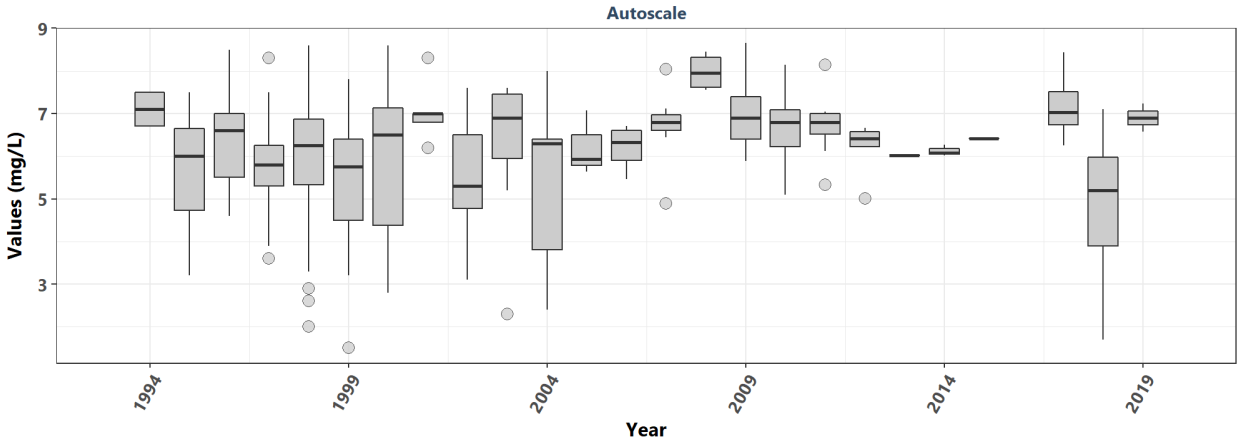


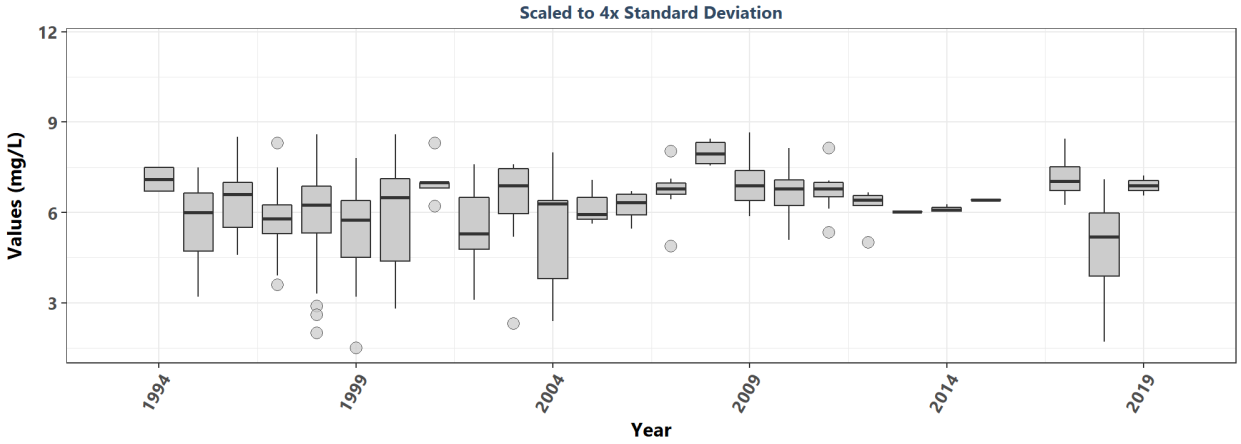

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2011
2013
2015
2017
2019
2021
Year

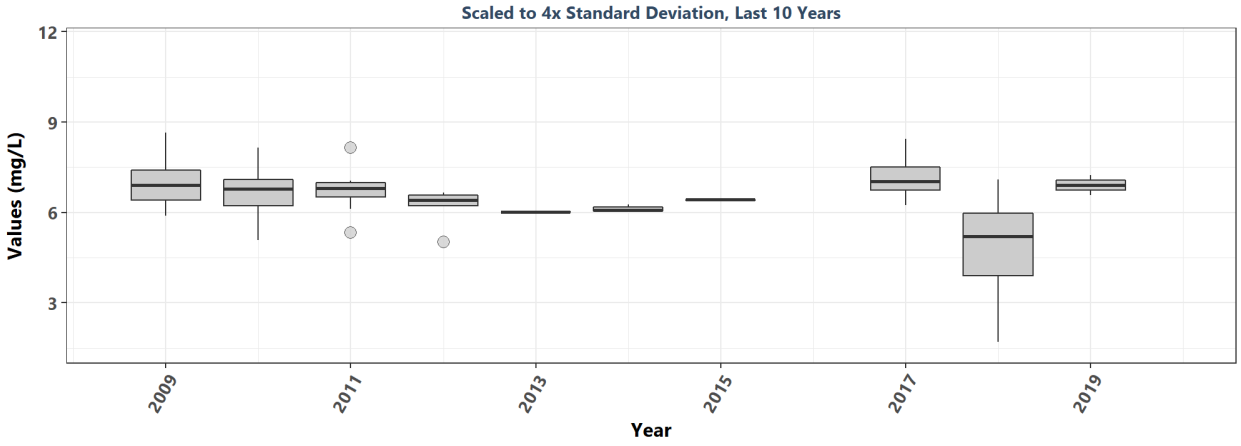

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

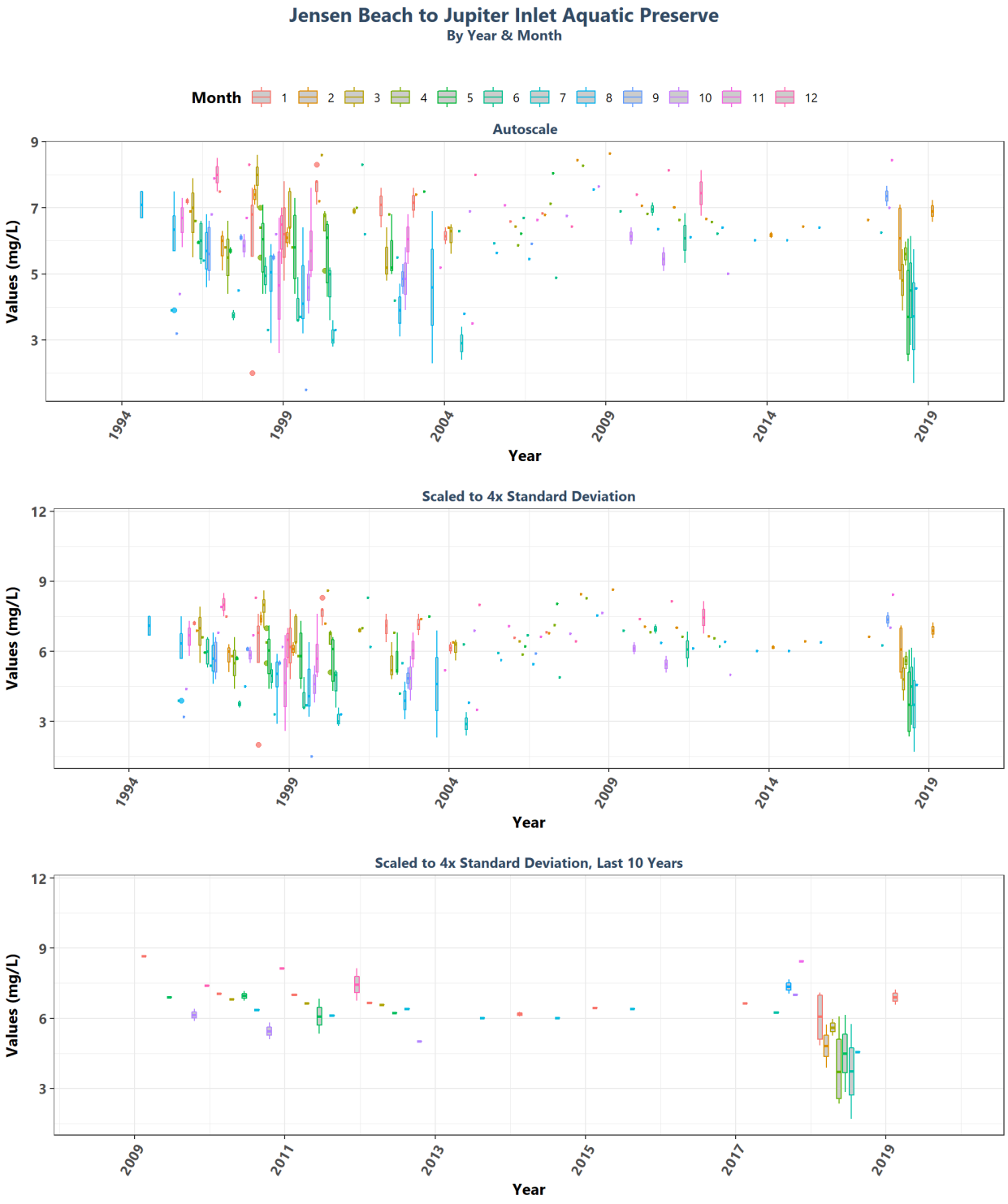
Jensen Beach to Jupiter Inlet Aquatic Preserve

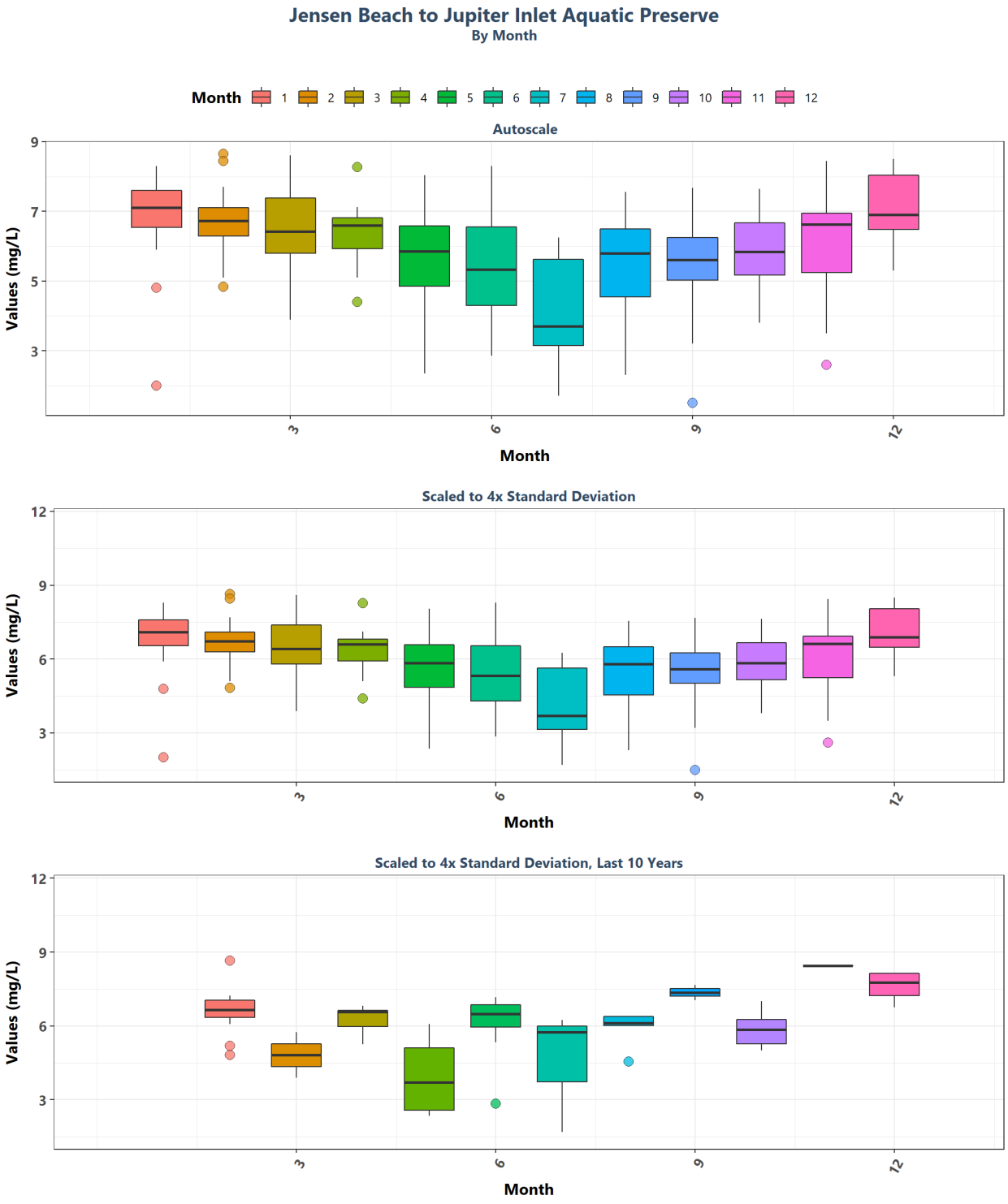
By Year

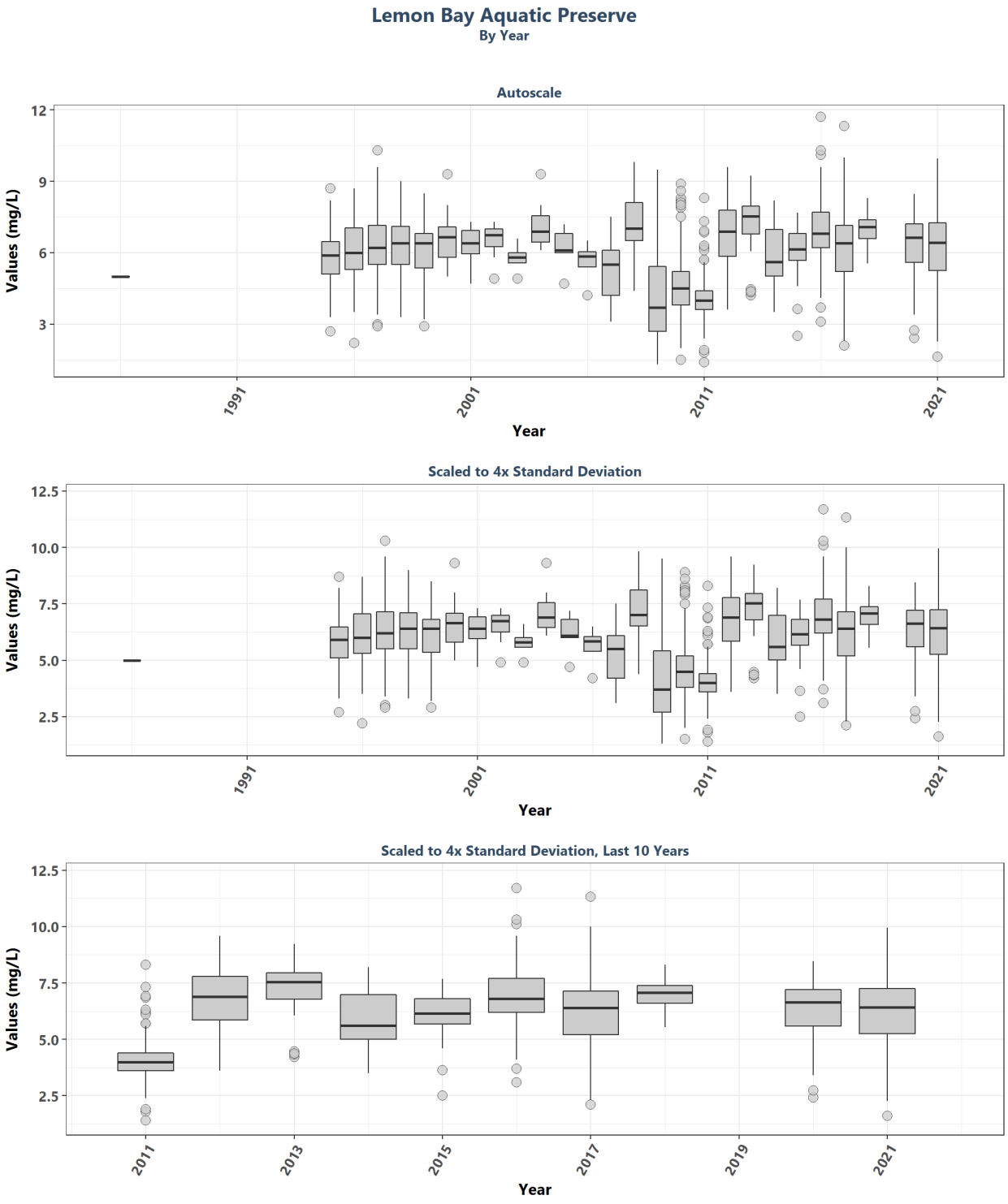


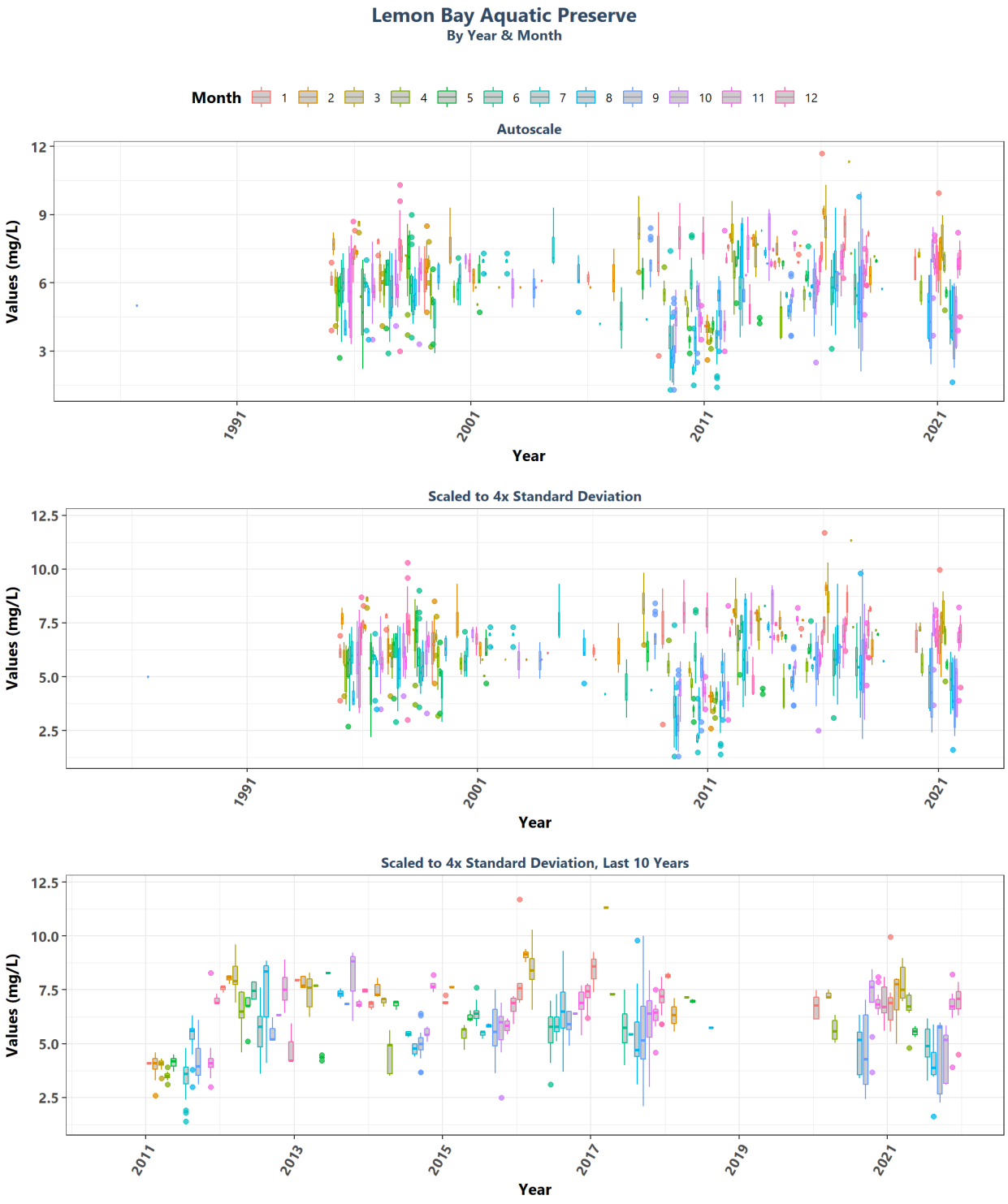


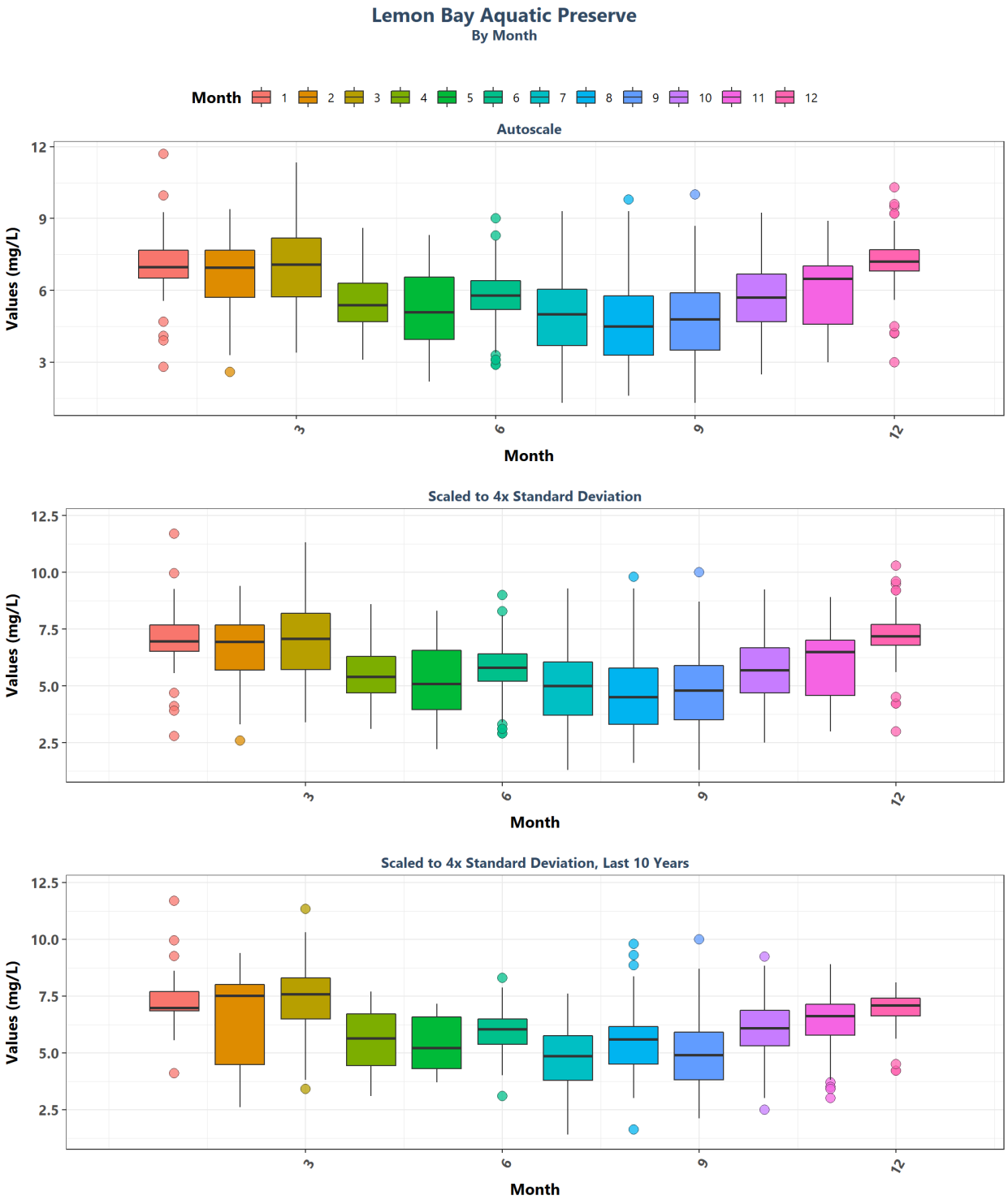



Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year


Jensen Beach to Jupiter Inlet Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
9
7
5
3
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12
9
6
3
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12
9
6
3
3
6
9
12
Month

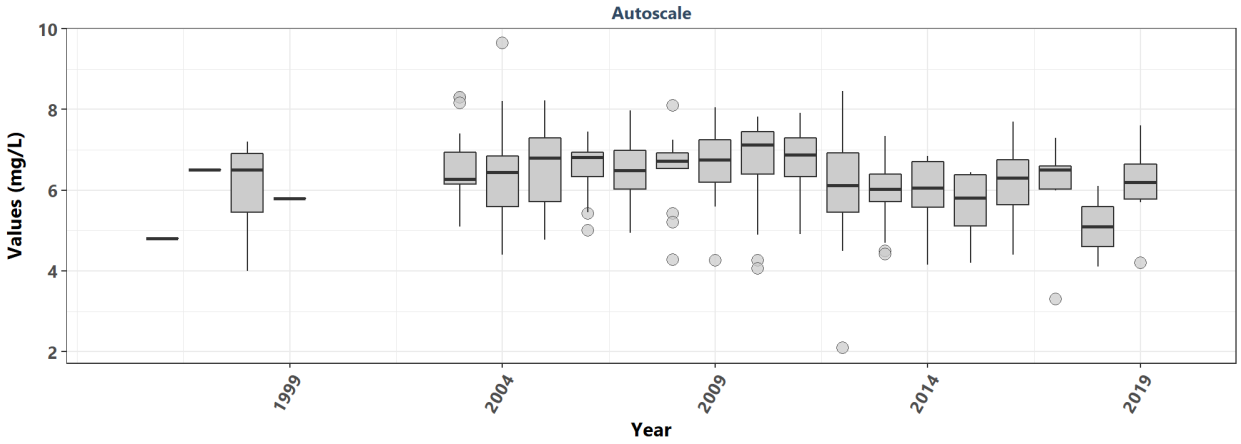

Lemon Bay Aquatic Preserve
By Year
Autoscale
Values (mg/L)
12
9
6
3
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year

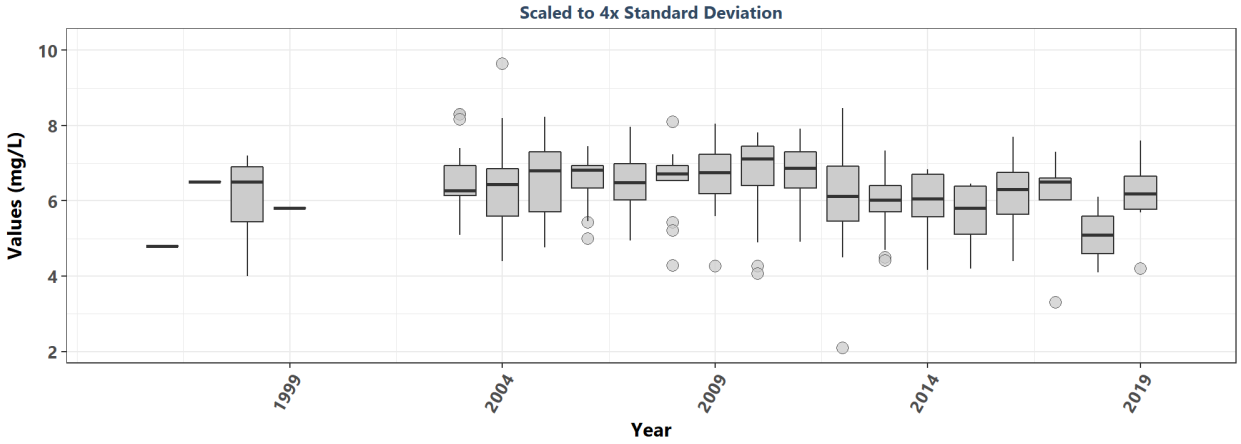

Lemon Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

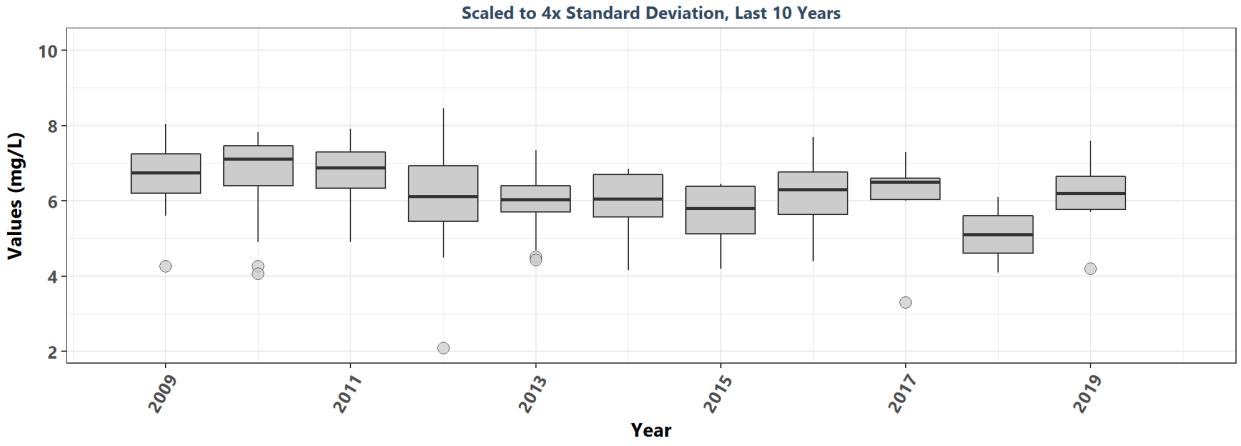

Lemon Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12
9
6
3
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month

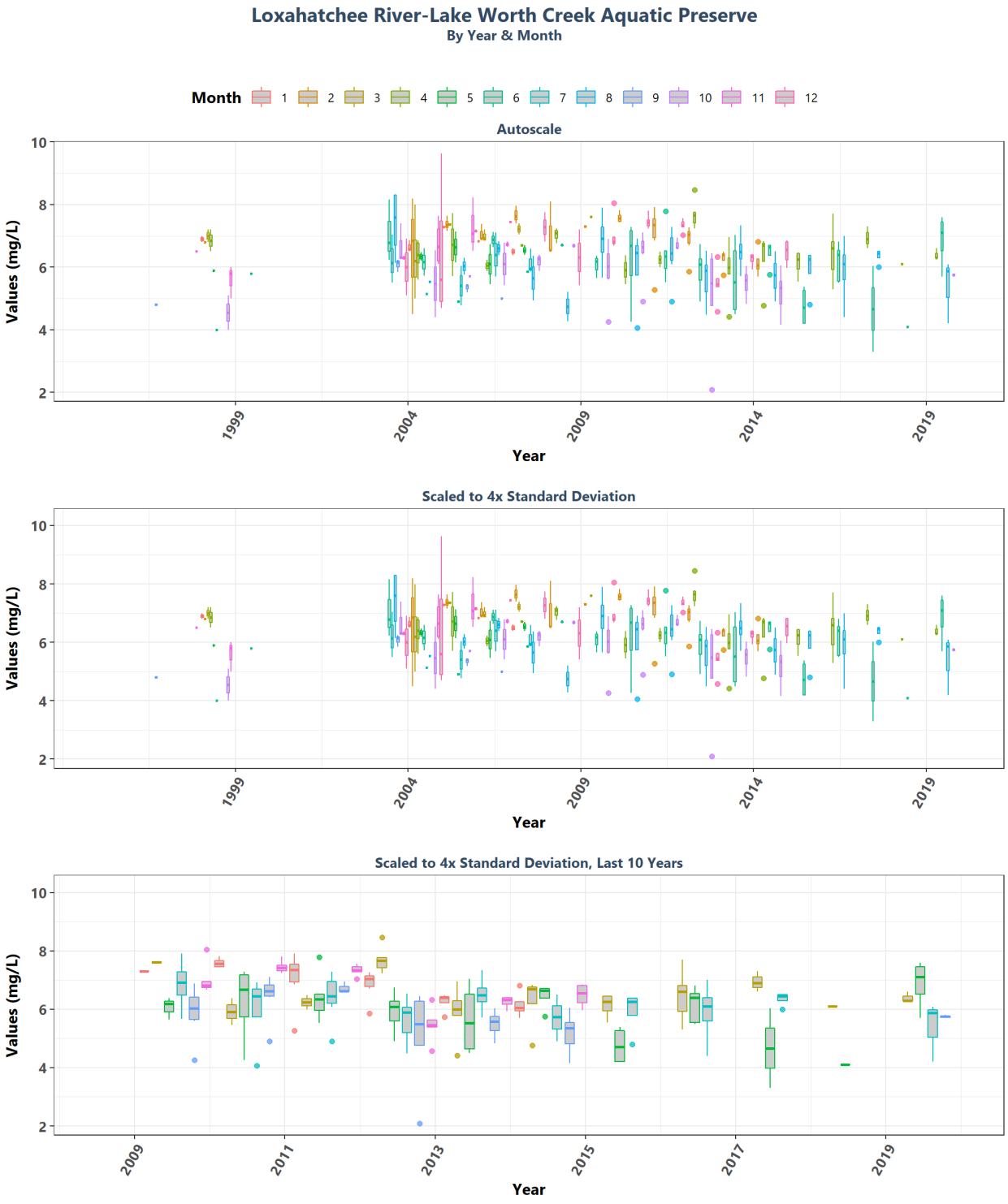
Loxahatchee River-Lake Worth Creek Aquatic Preserve

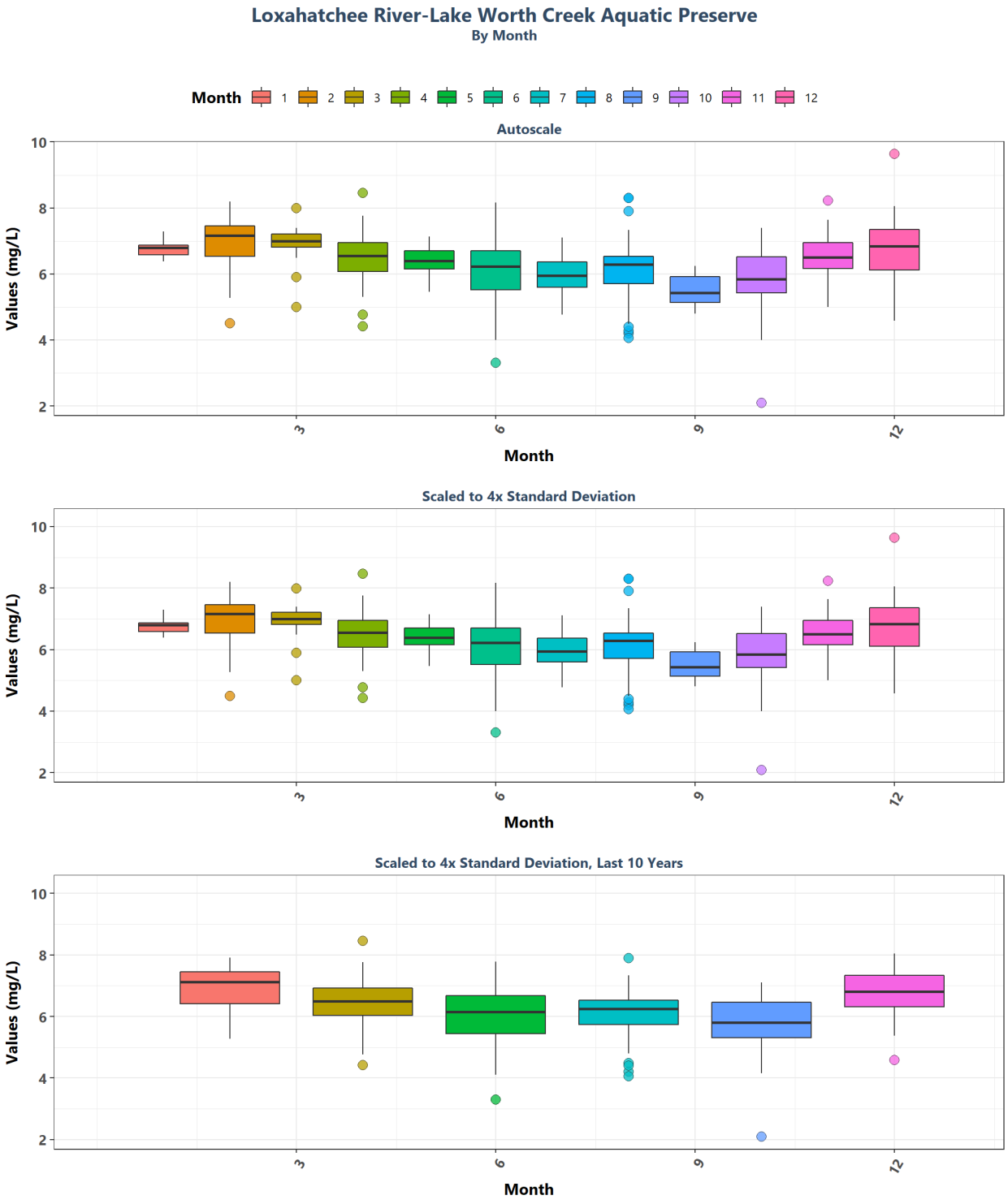
By Year





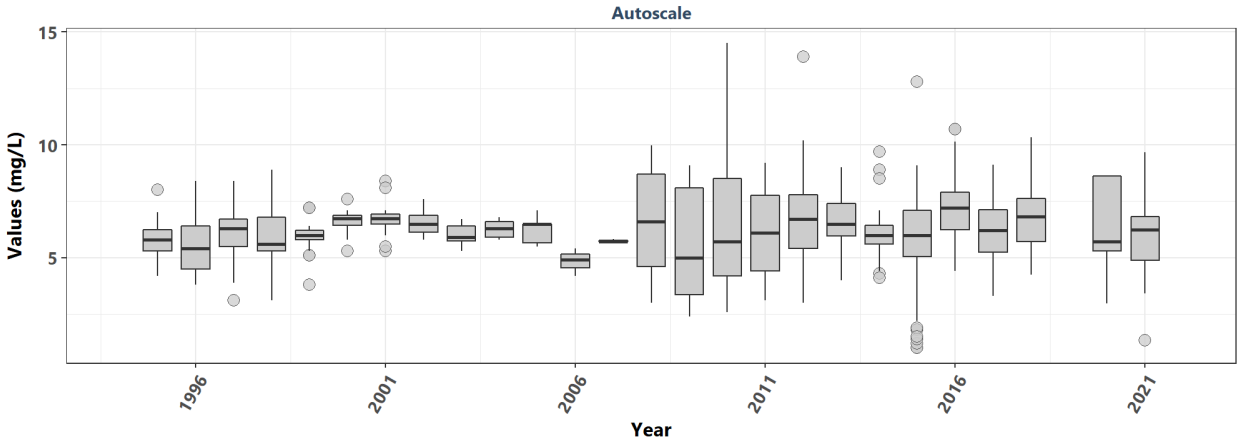


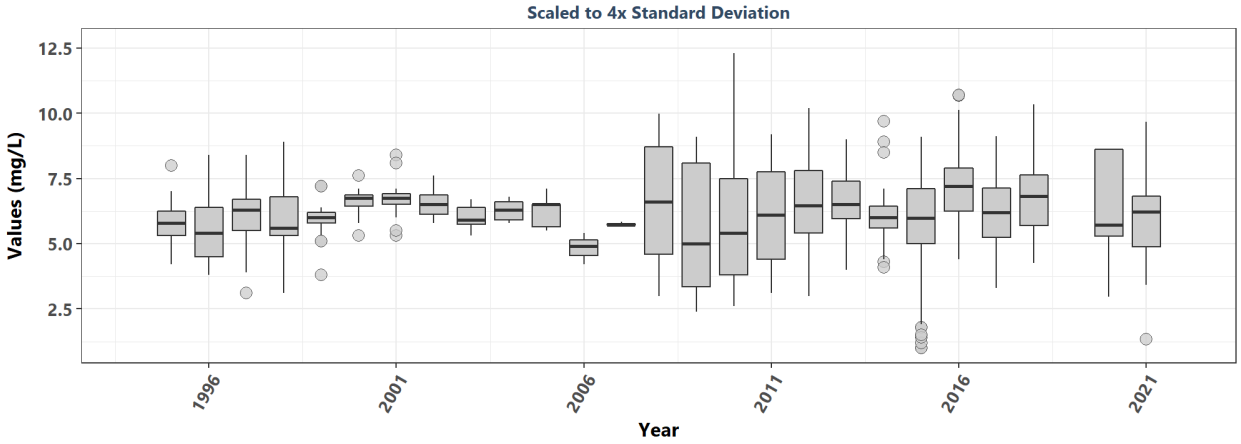

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

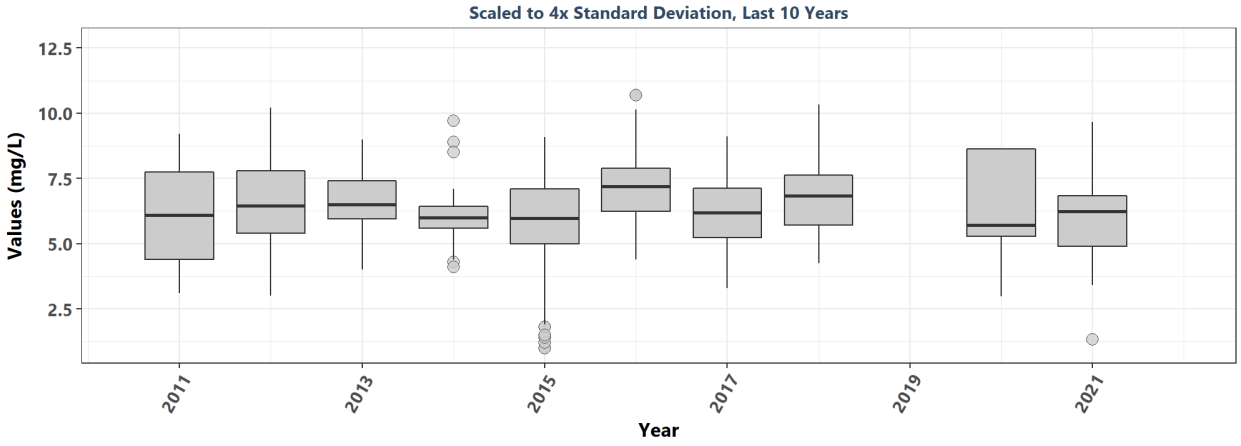

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

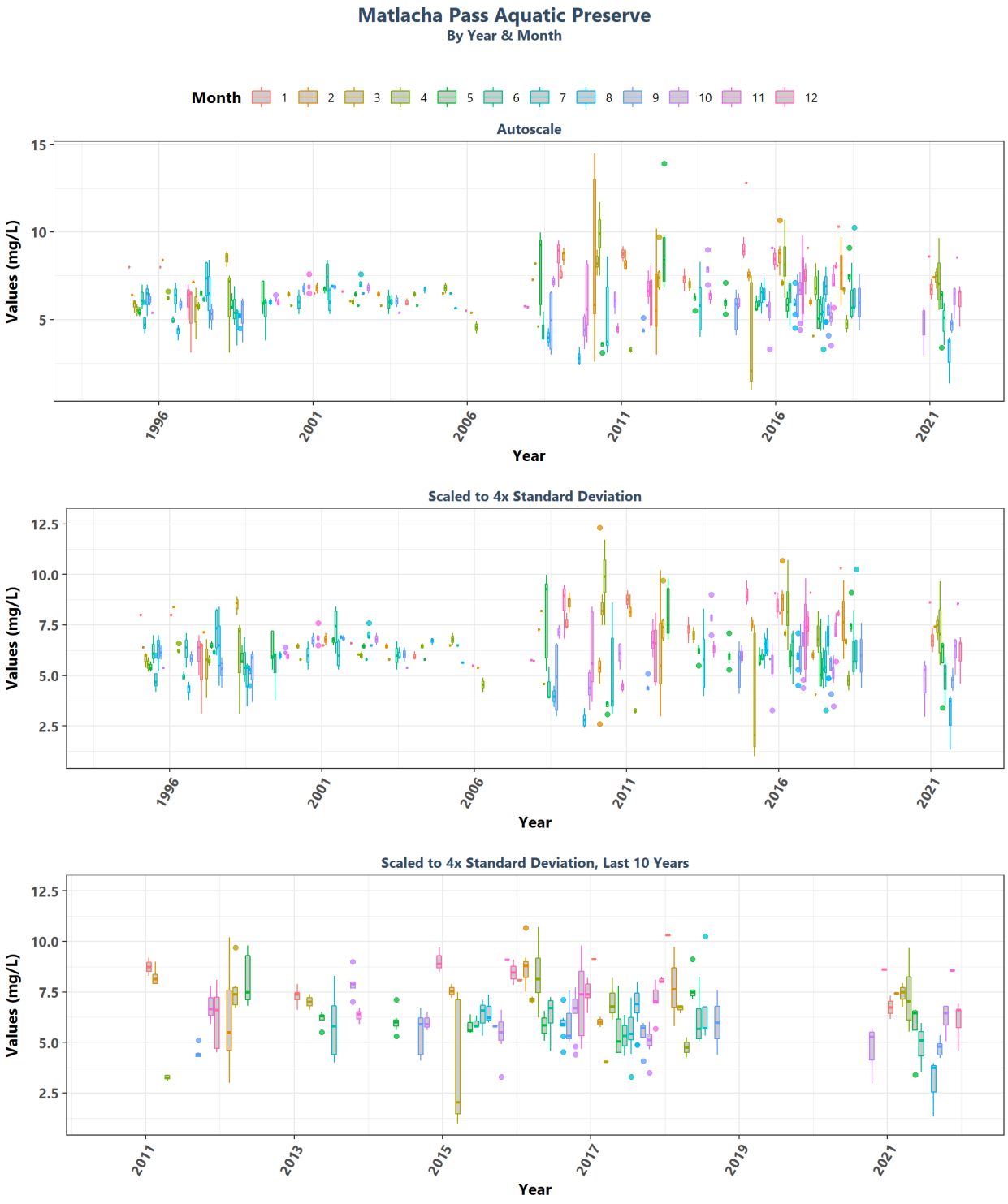
Matlacha Pass Aquatic Preserve

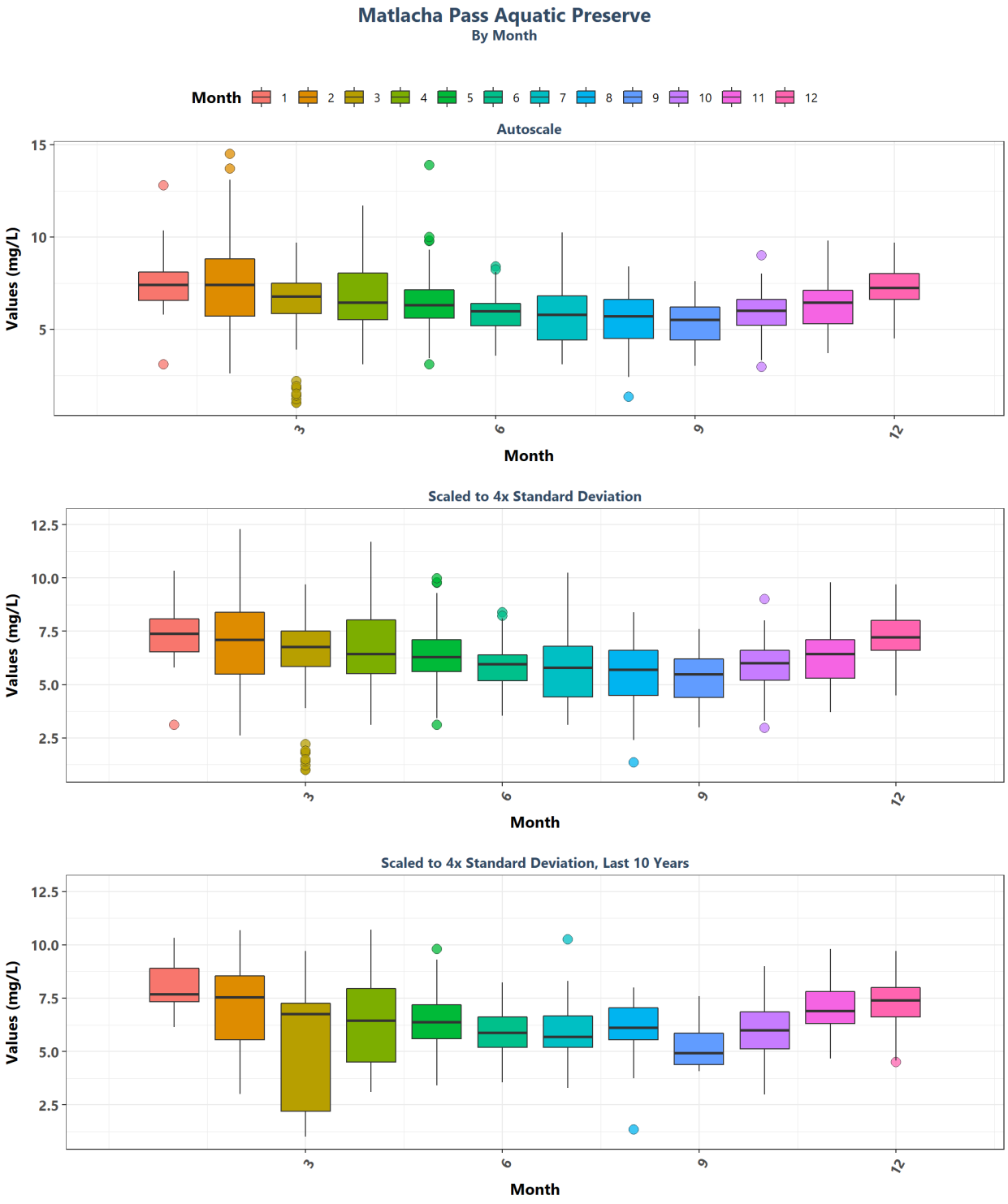
By Year

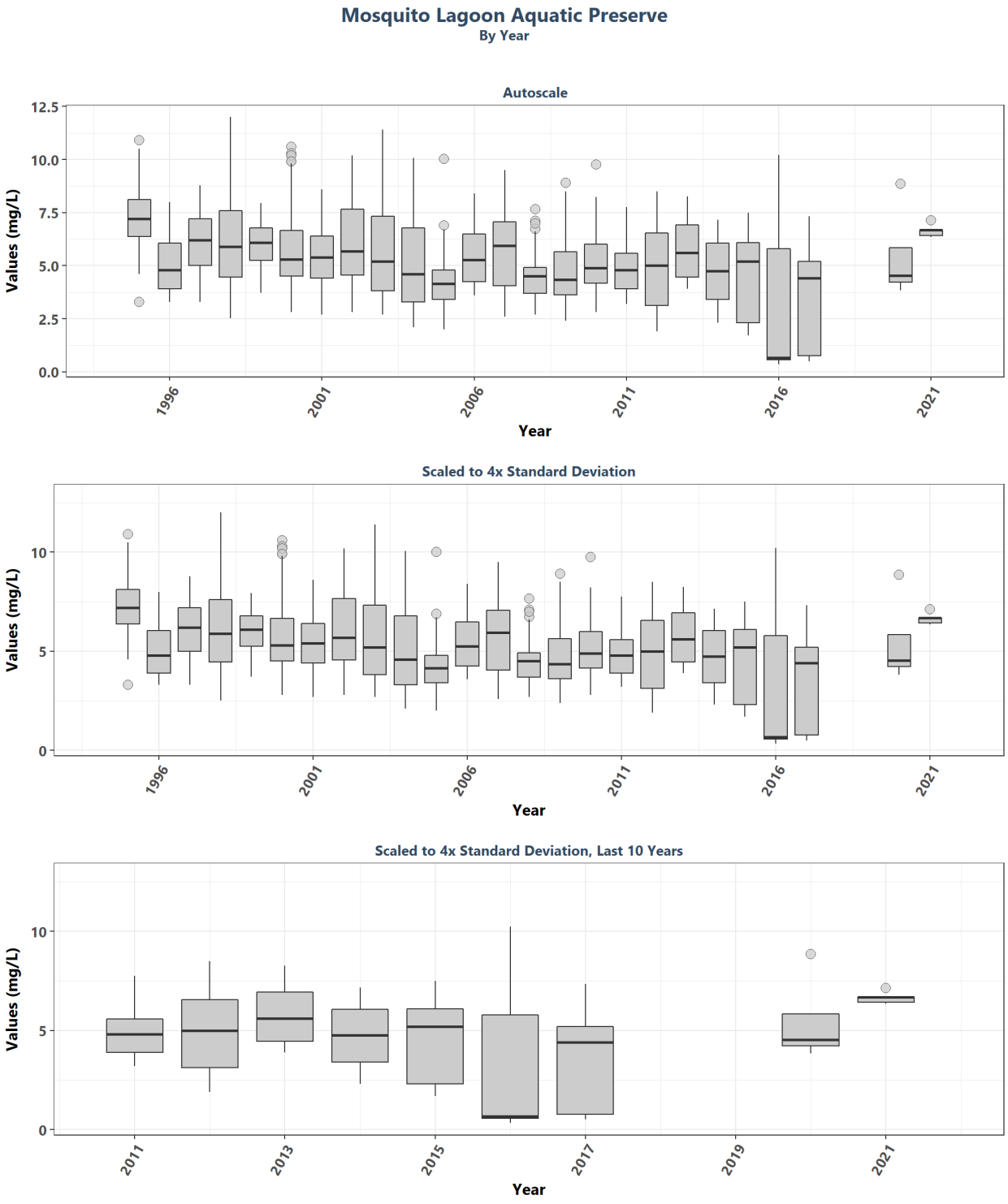


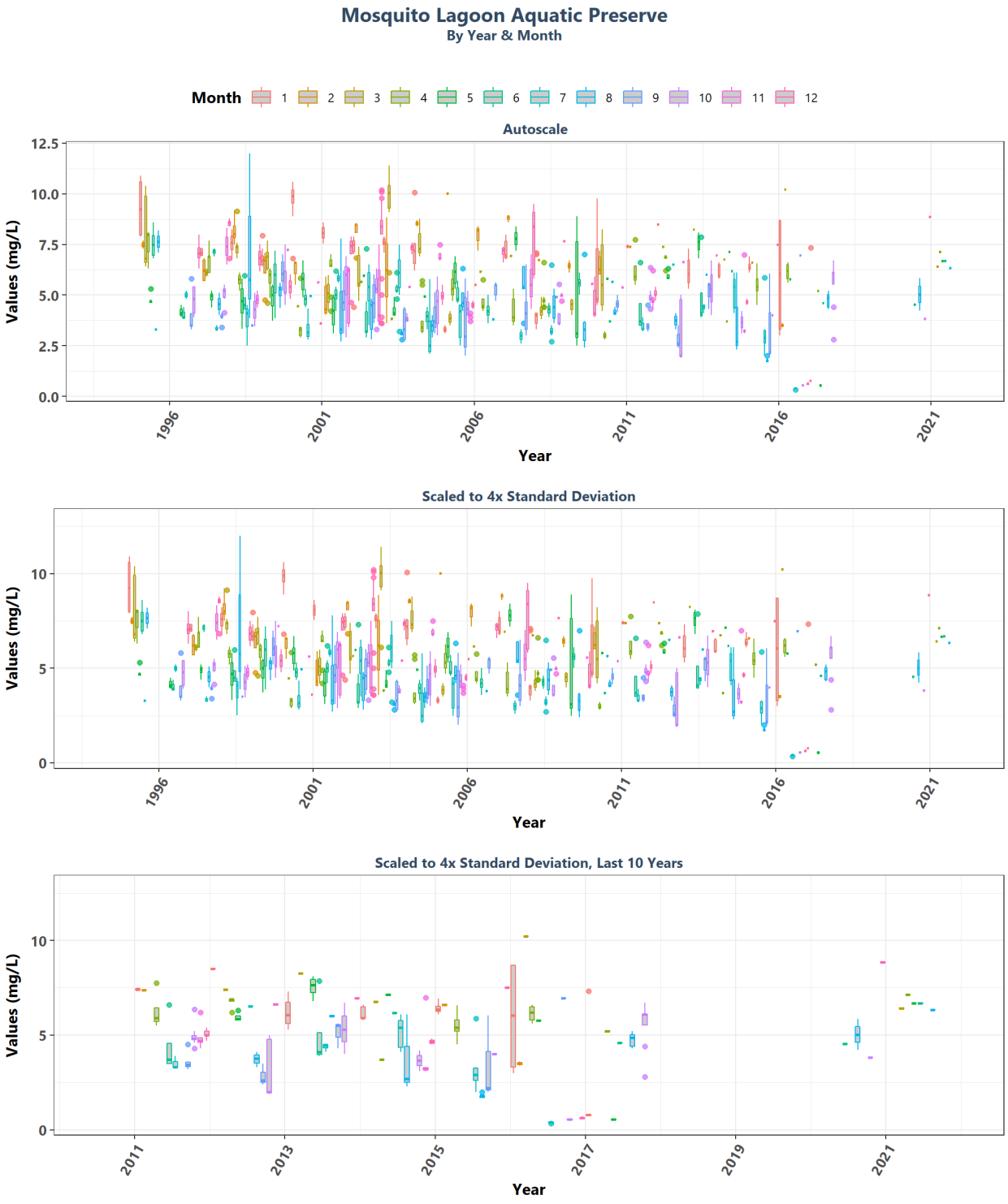


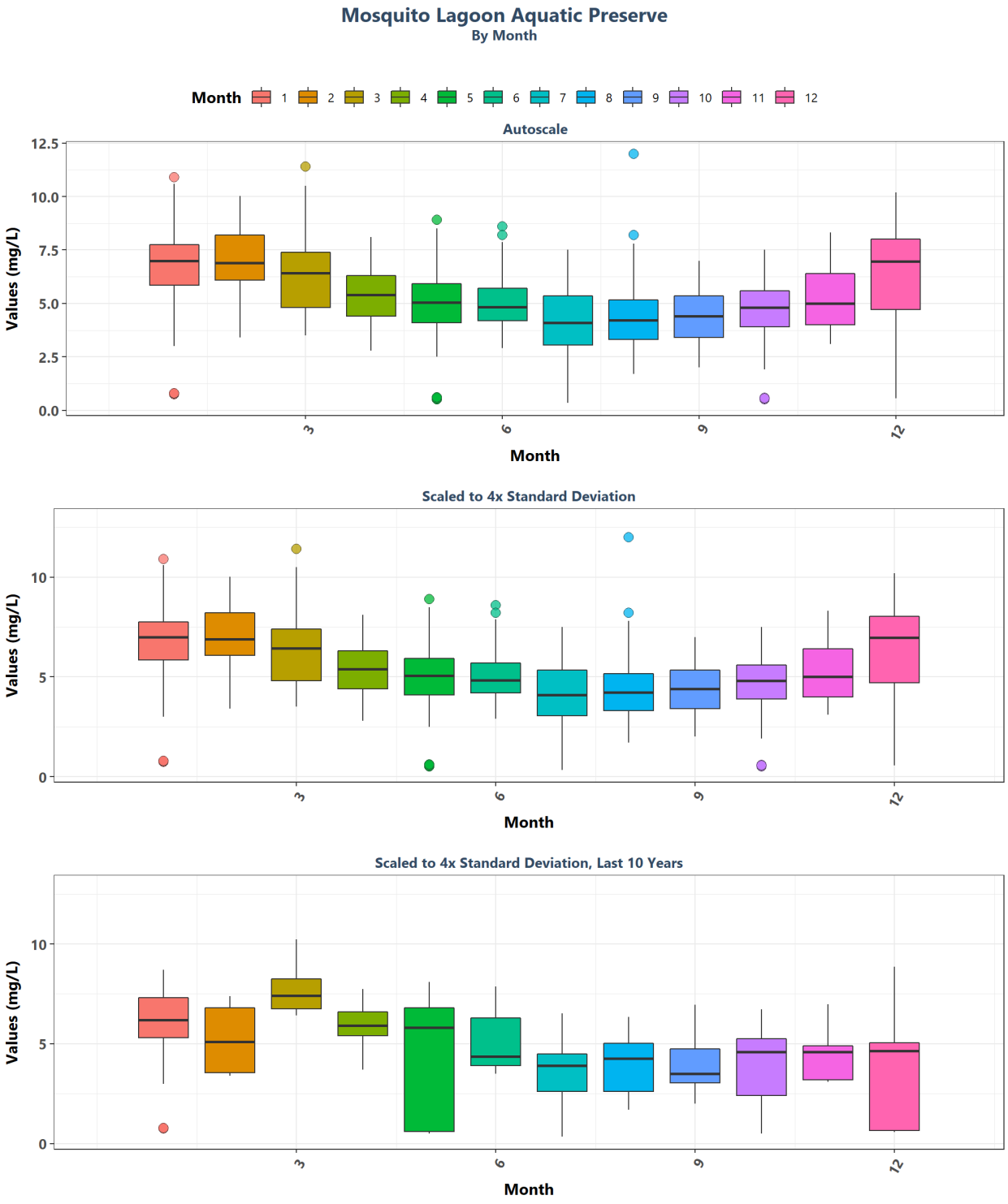



Matlacha Pass Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15
10
5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year


Matlacha Pass Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month

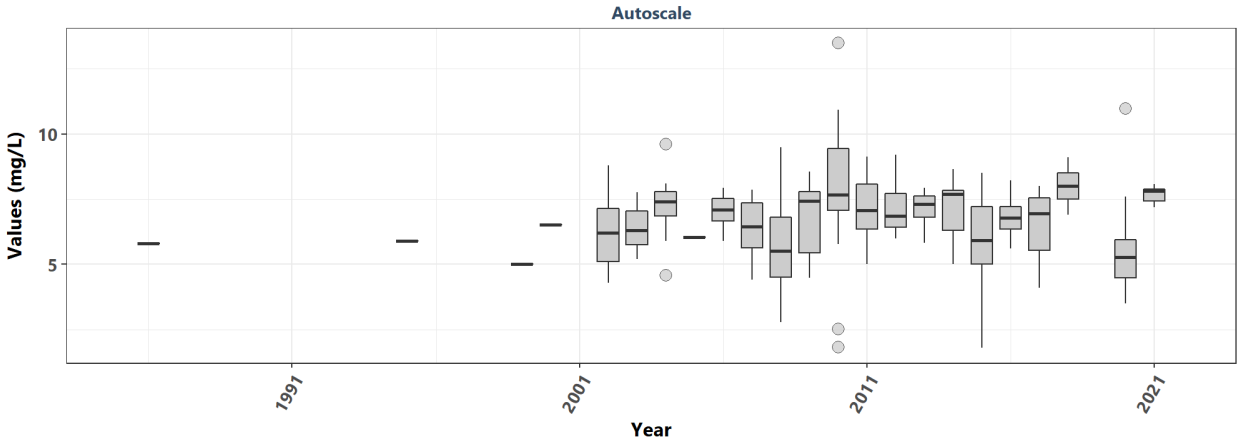

Mosquito Lagoon Aquatic Preserve
By Year
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2011
2013
2015
2017
2019
2021
Year

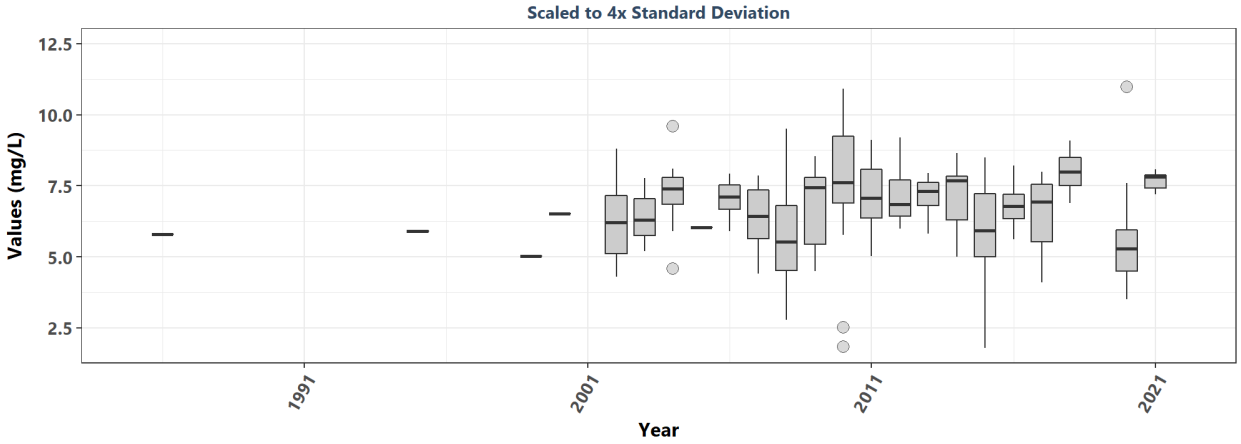

Mosquito Lagoon Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2011
2013
2015
2017
2019
2021
Year

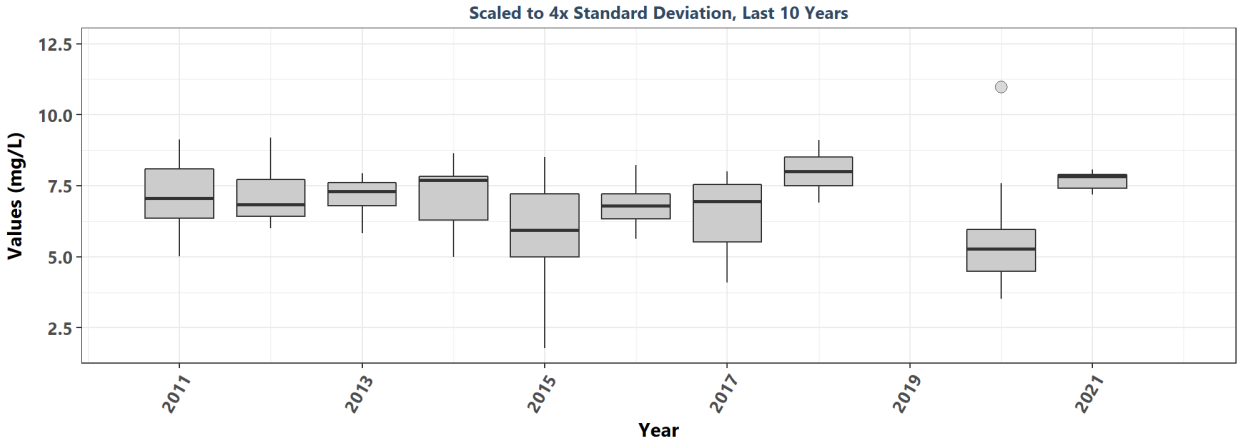

Mosquito Lagoon Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

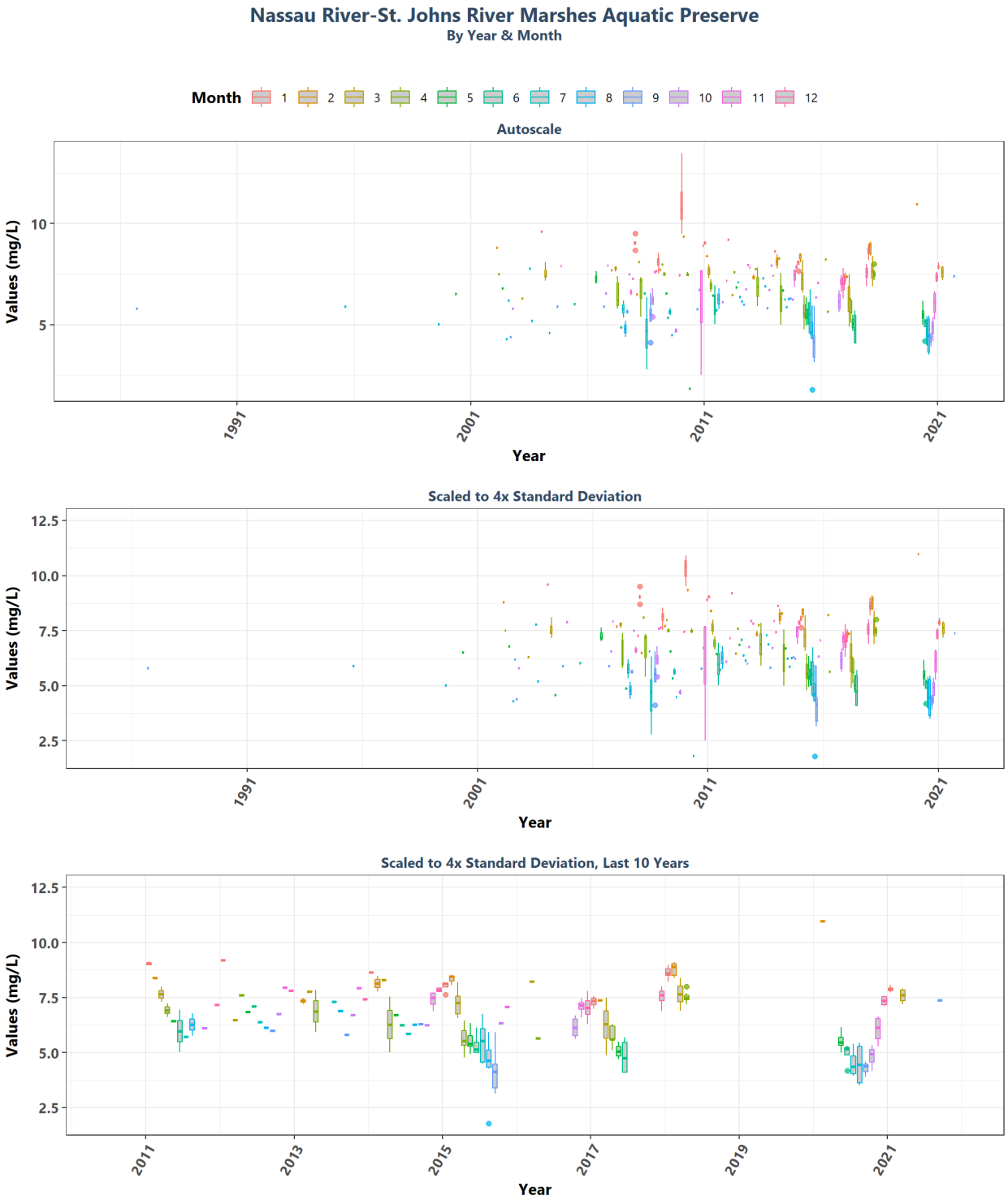
Nassau River-St. Johns River Marshes Aquatic Preserve

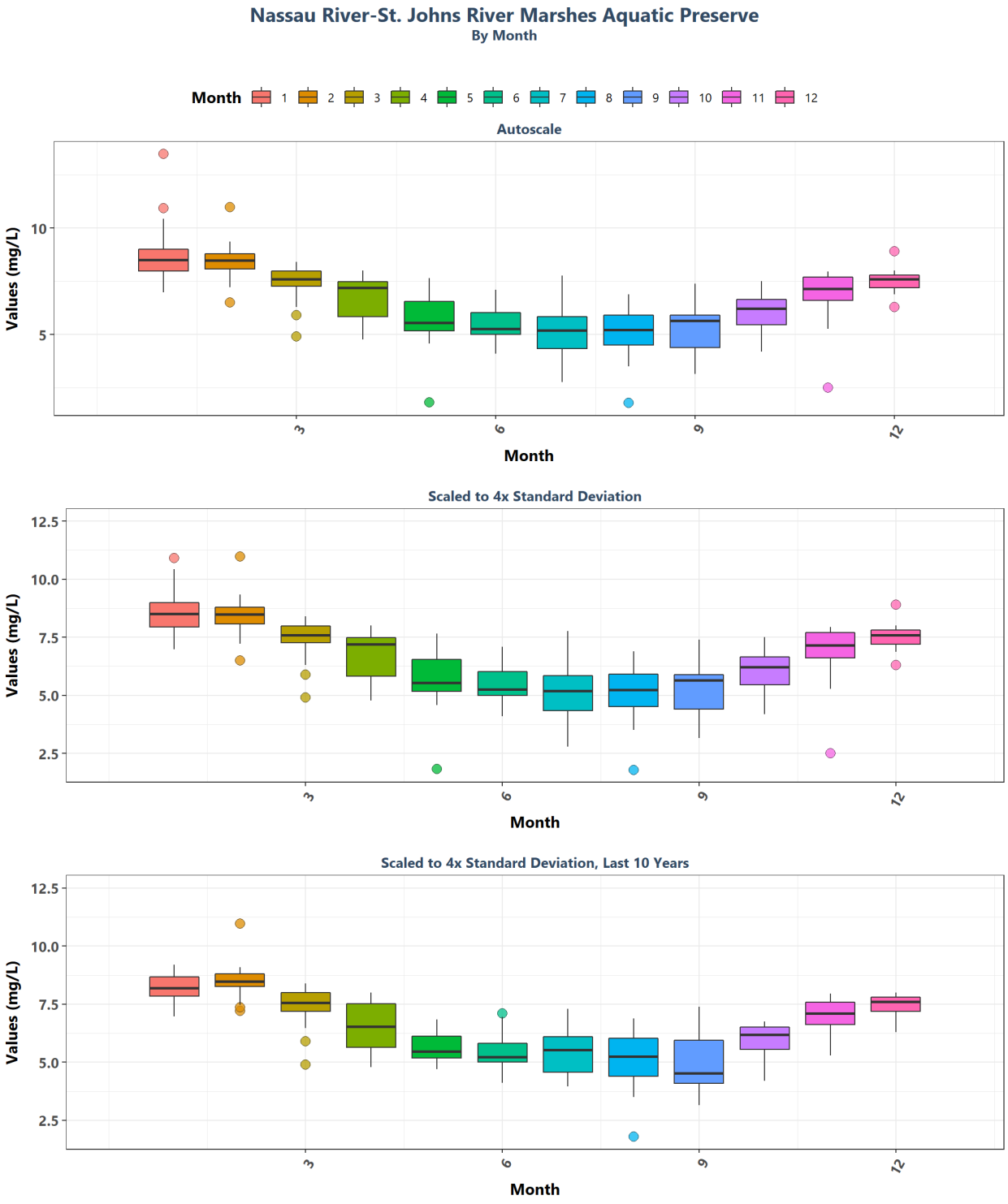
By Year





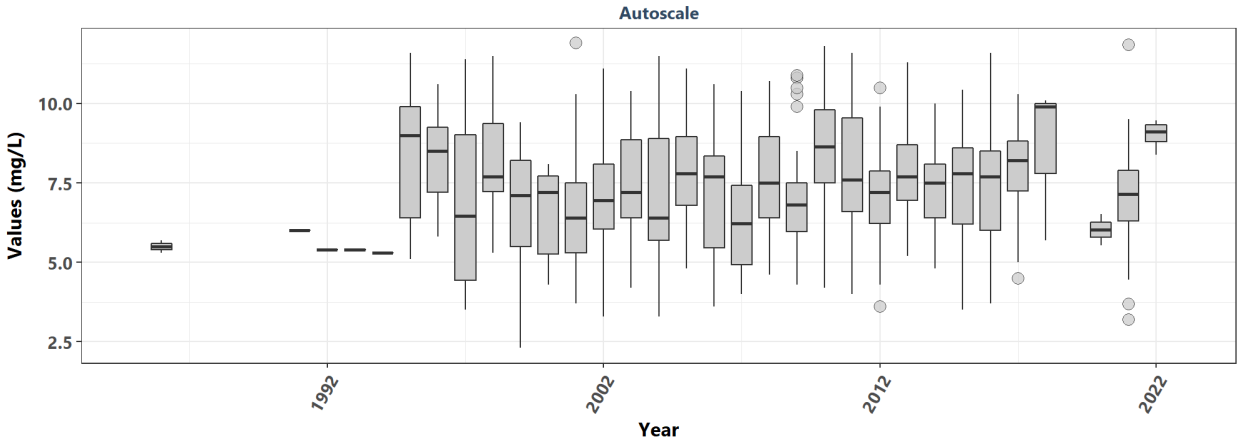


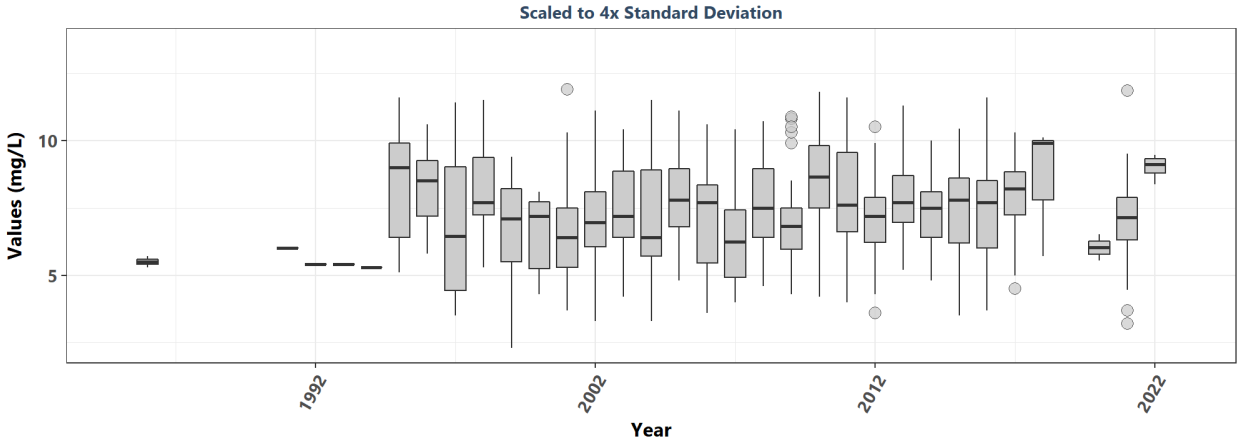

Nassau River-St. Johns River Marshes Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year

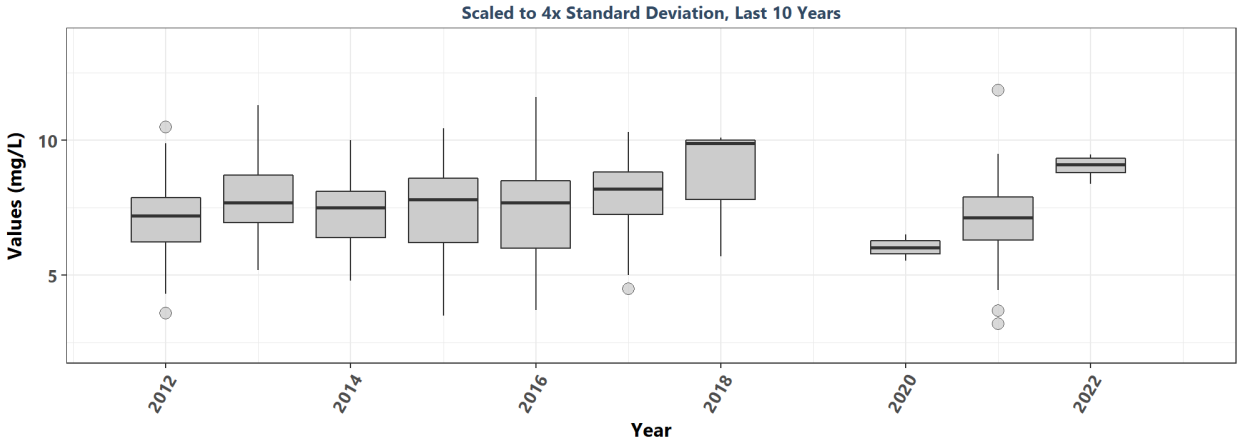

Nassau River-St. Johns River Marshes Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month

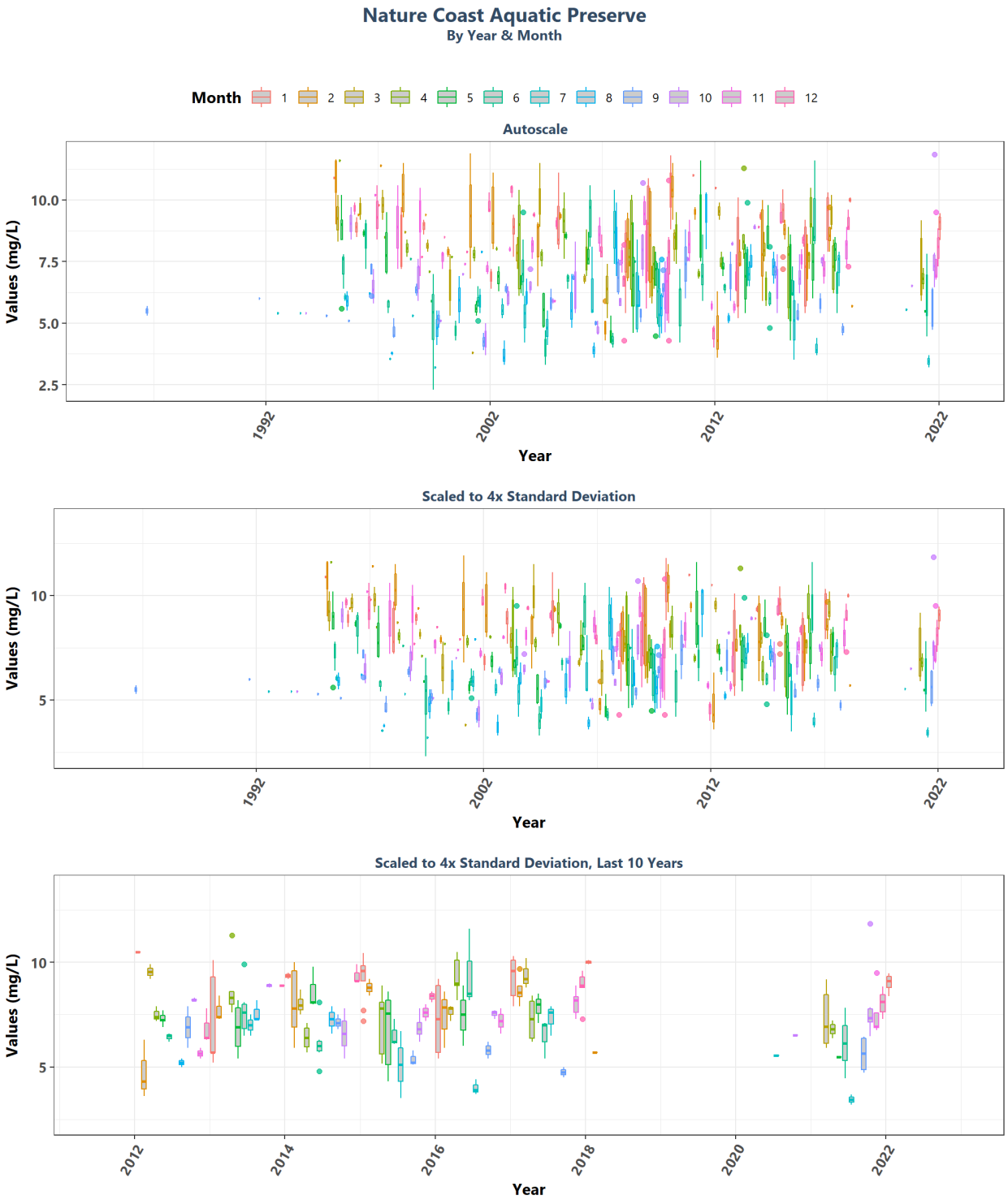
Nature Coast Aquatic Preserve

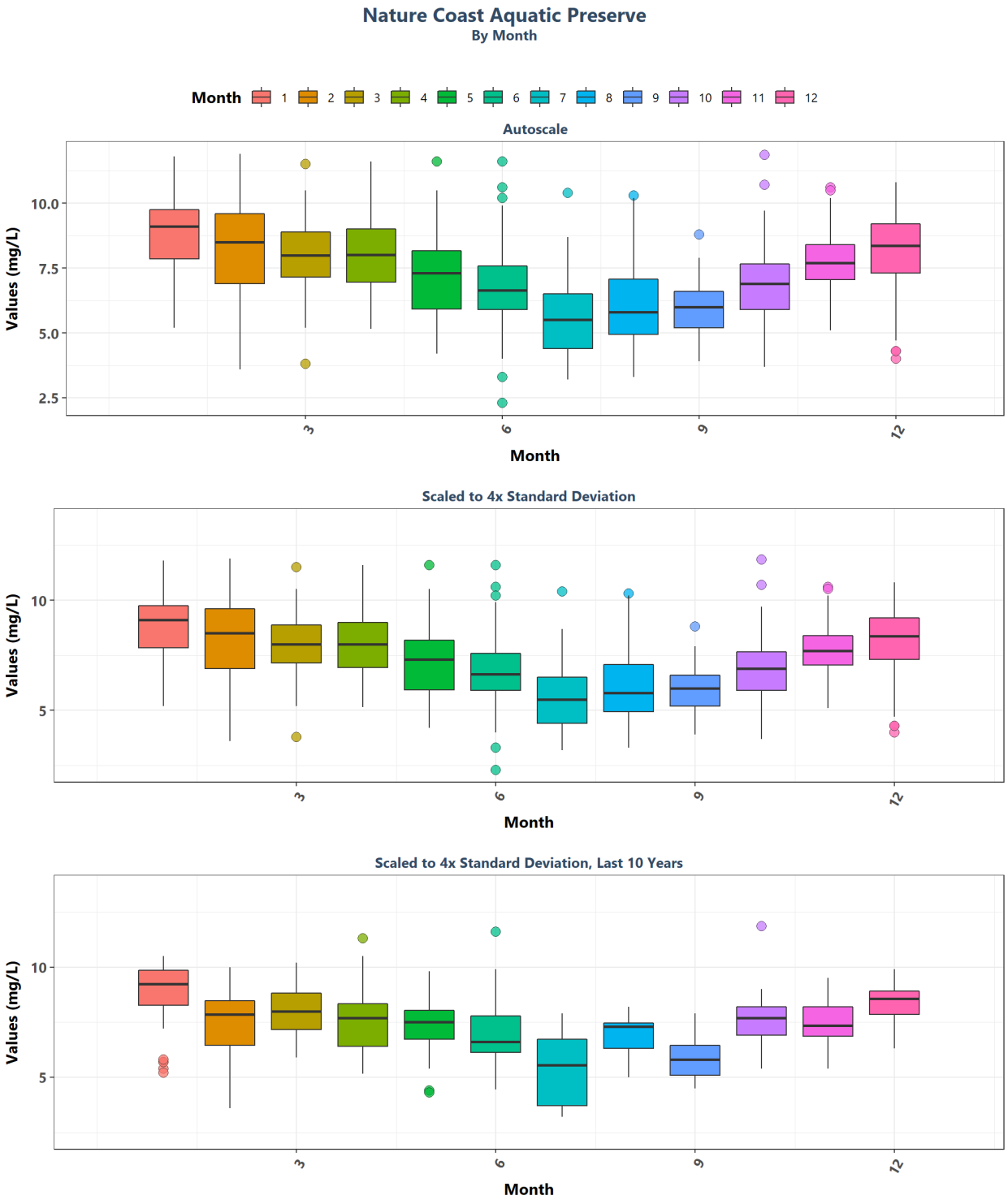
By Year

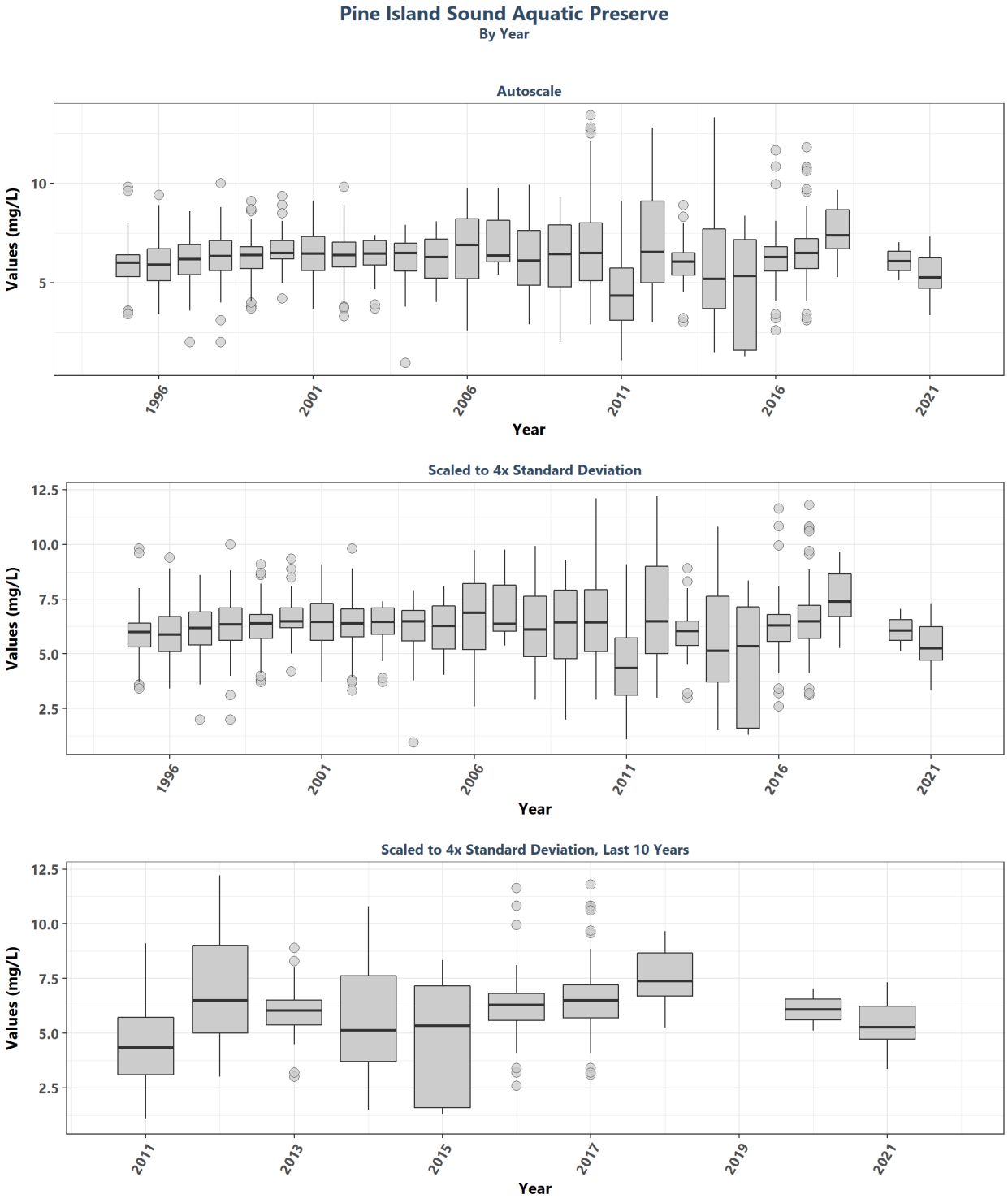


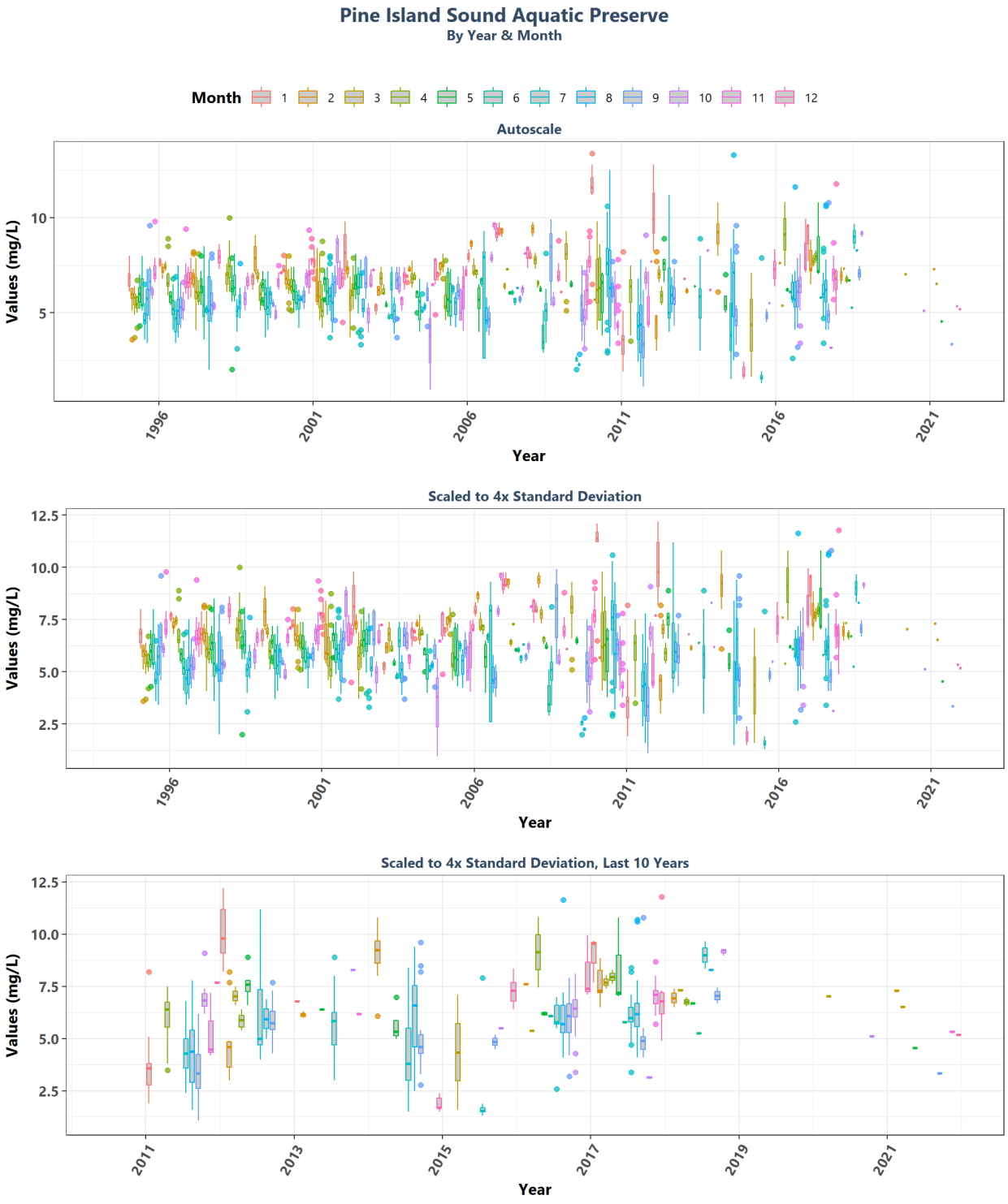


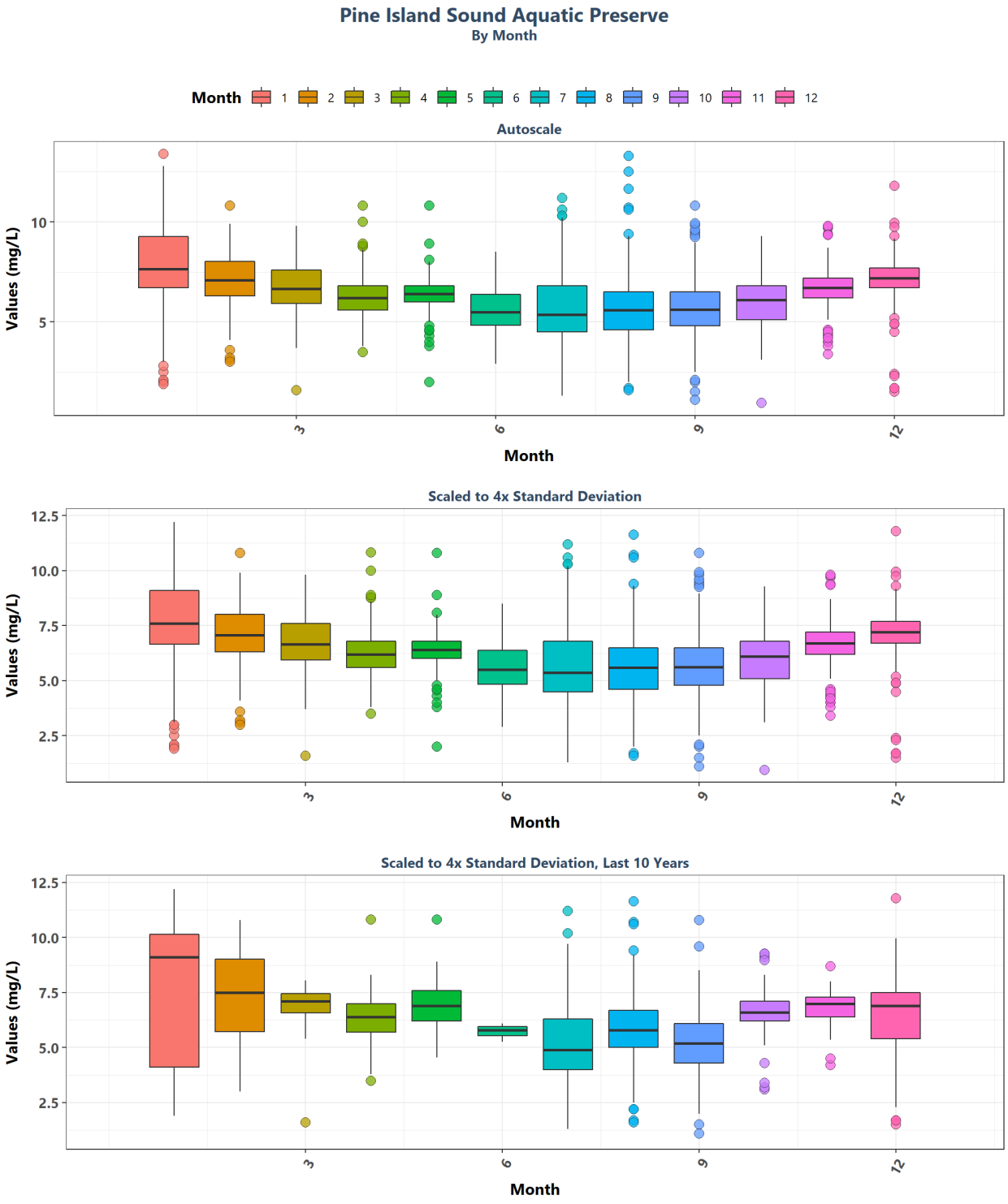


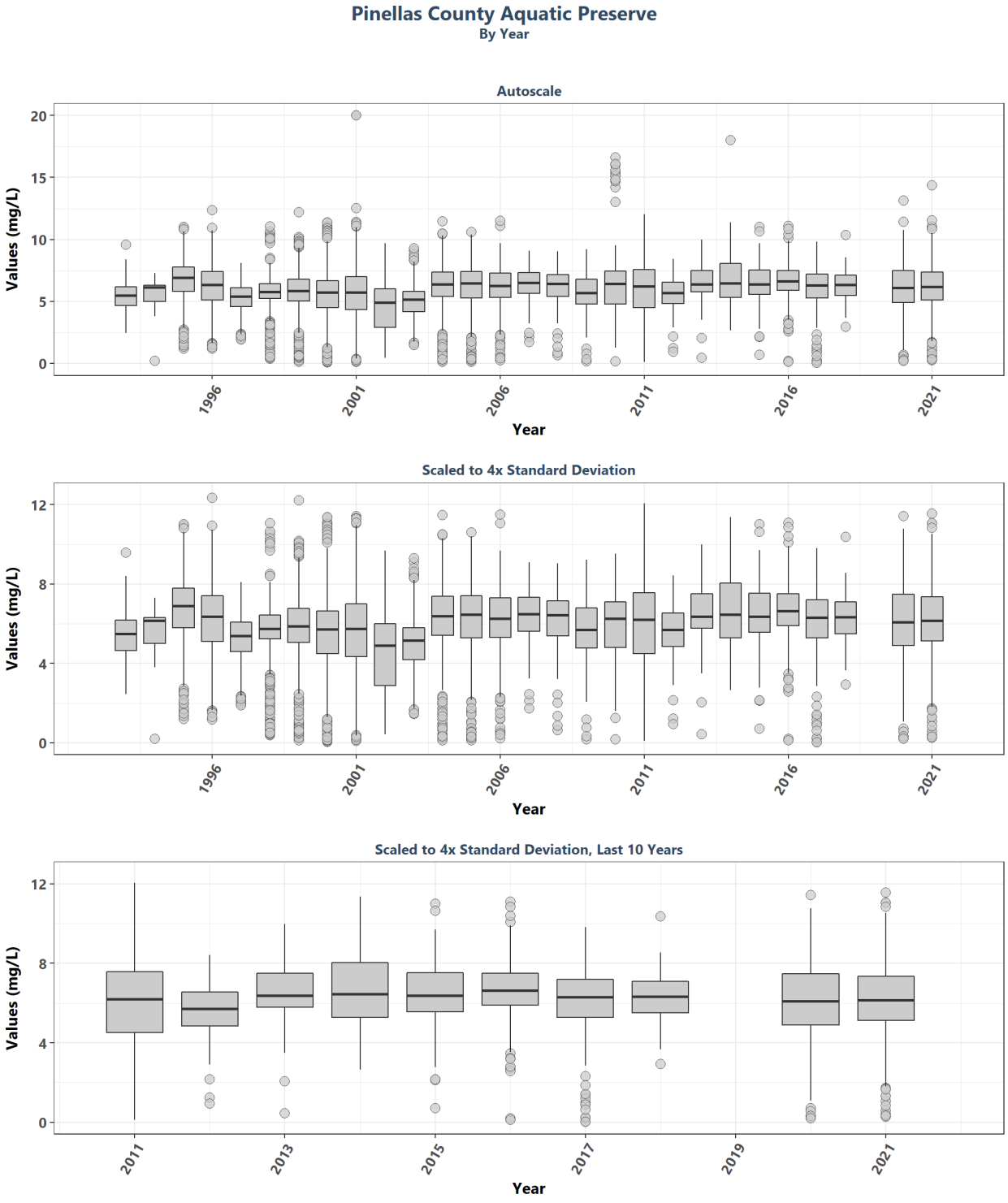

Nature Coast Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
2012
2014
2016
2018
2020
2022
Year

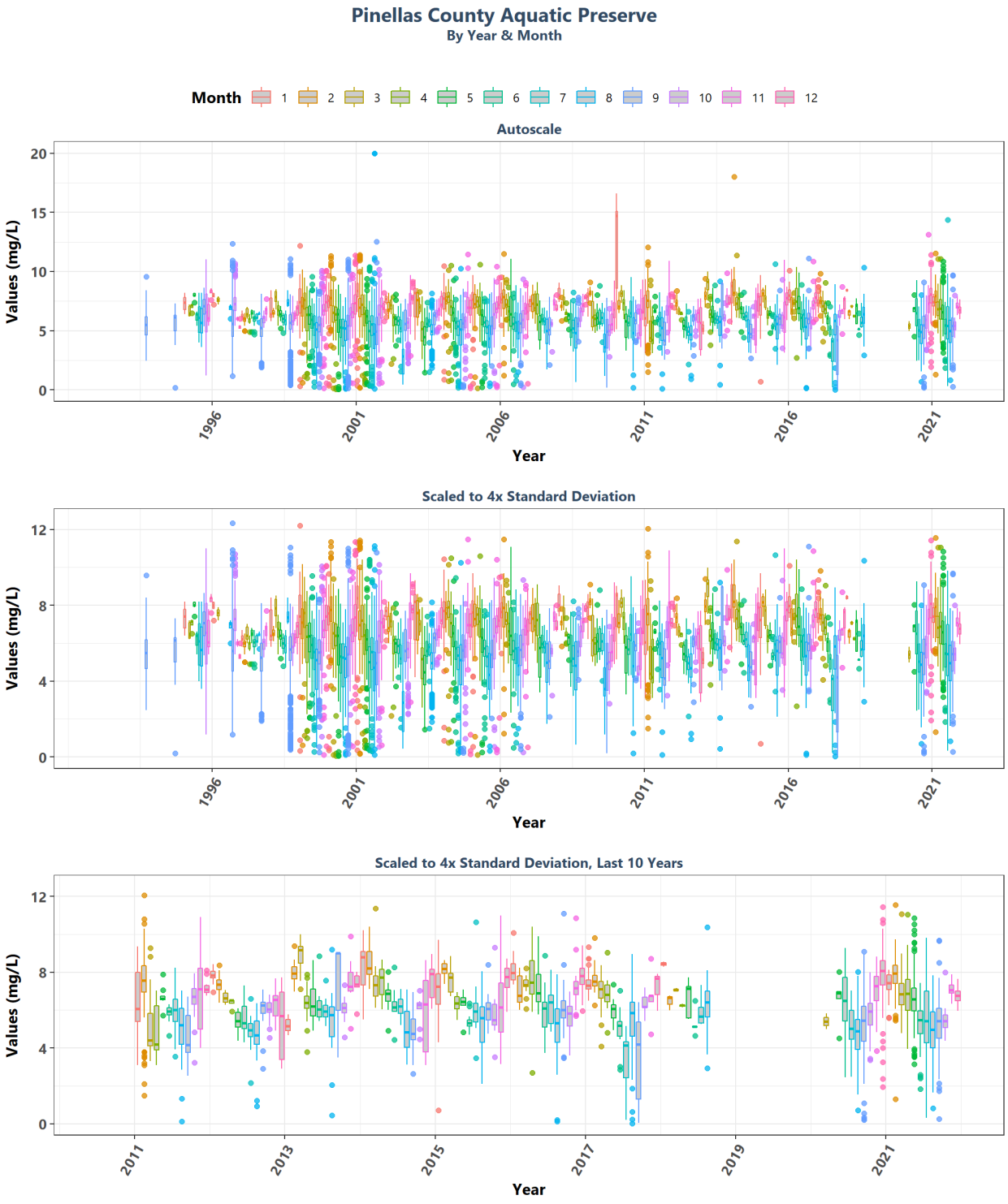

Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
3
6
9
12
Month

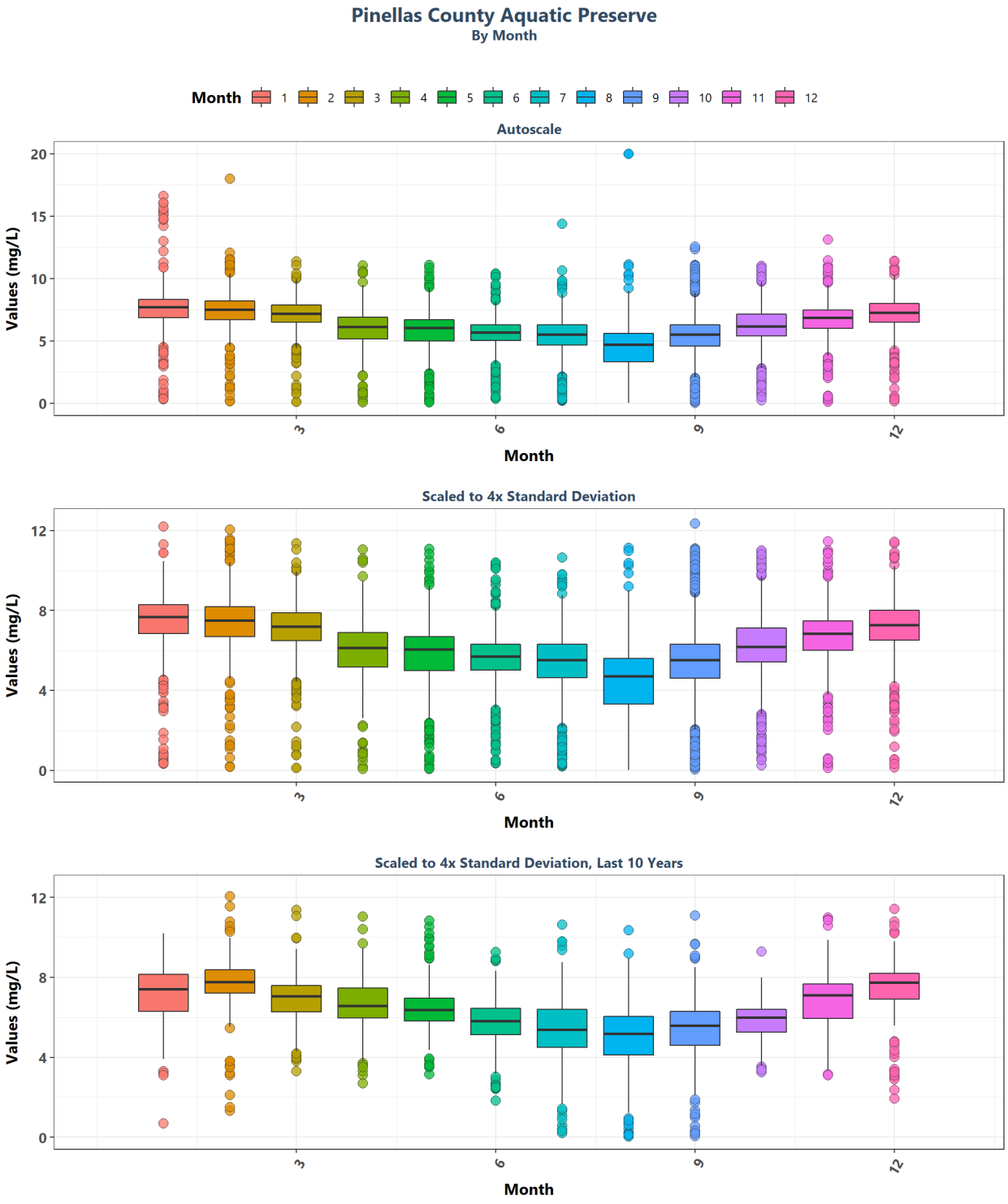

Pine Island Sound Aquatic Preserve
By Year
Autoscale
Values (mg/L)
10
5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year

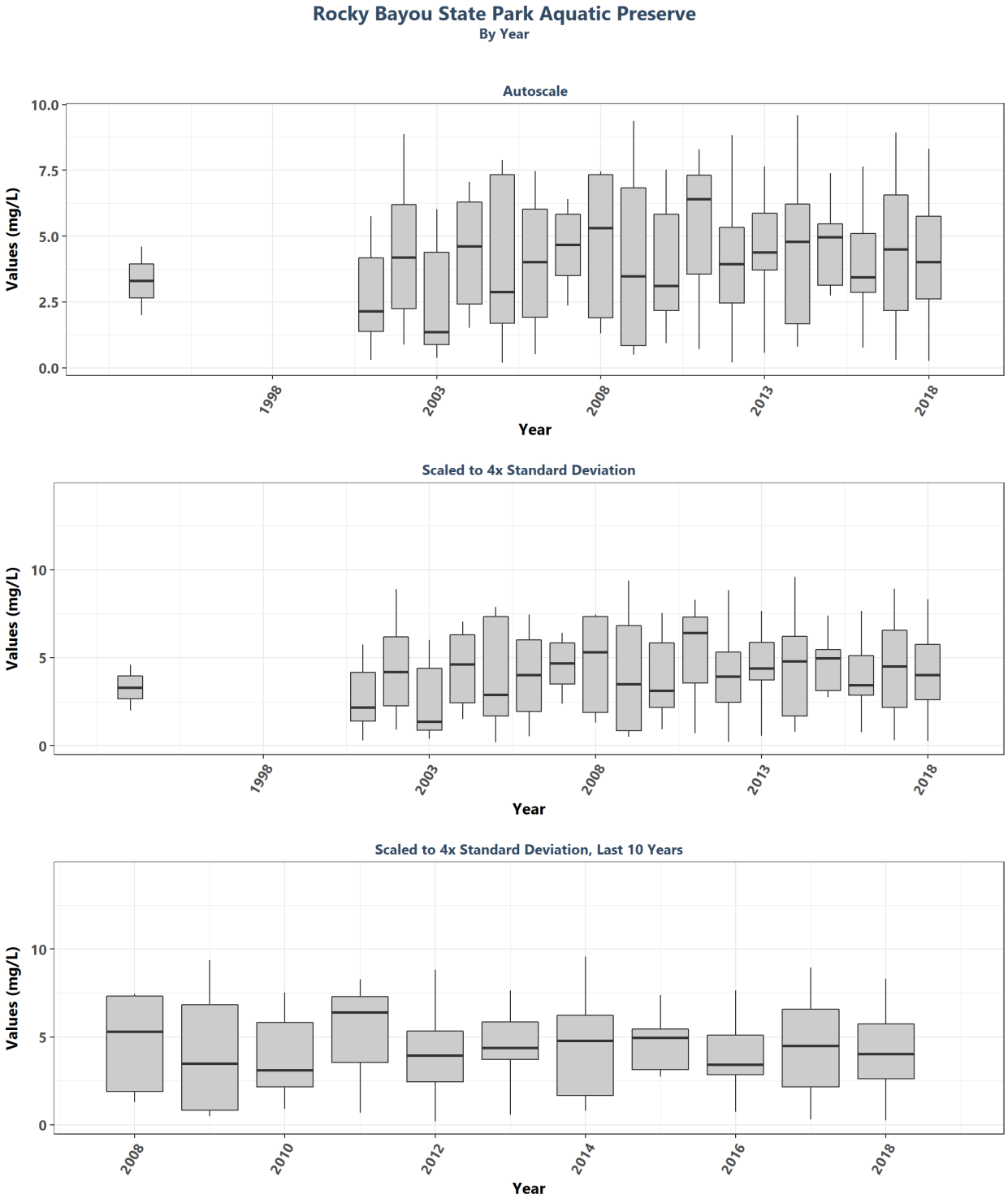

Pine Island Sound Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year

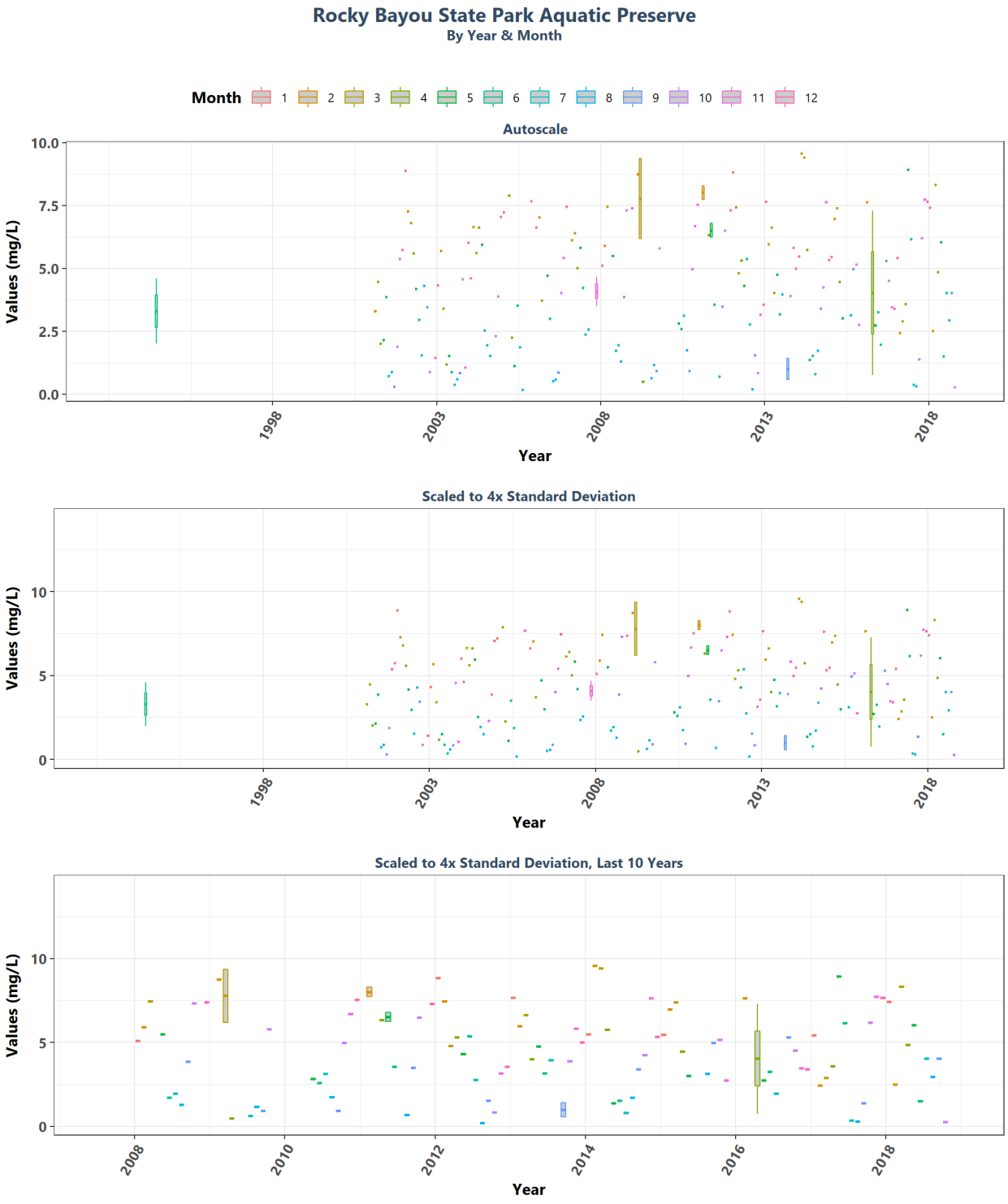

Pine Island Sound Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month

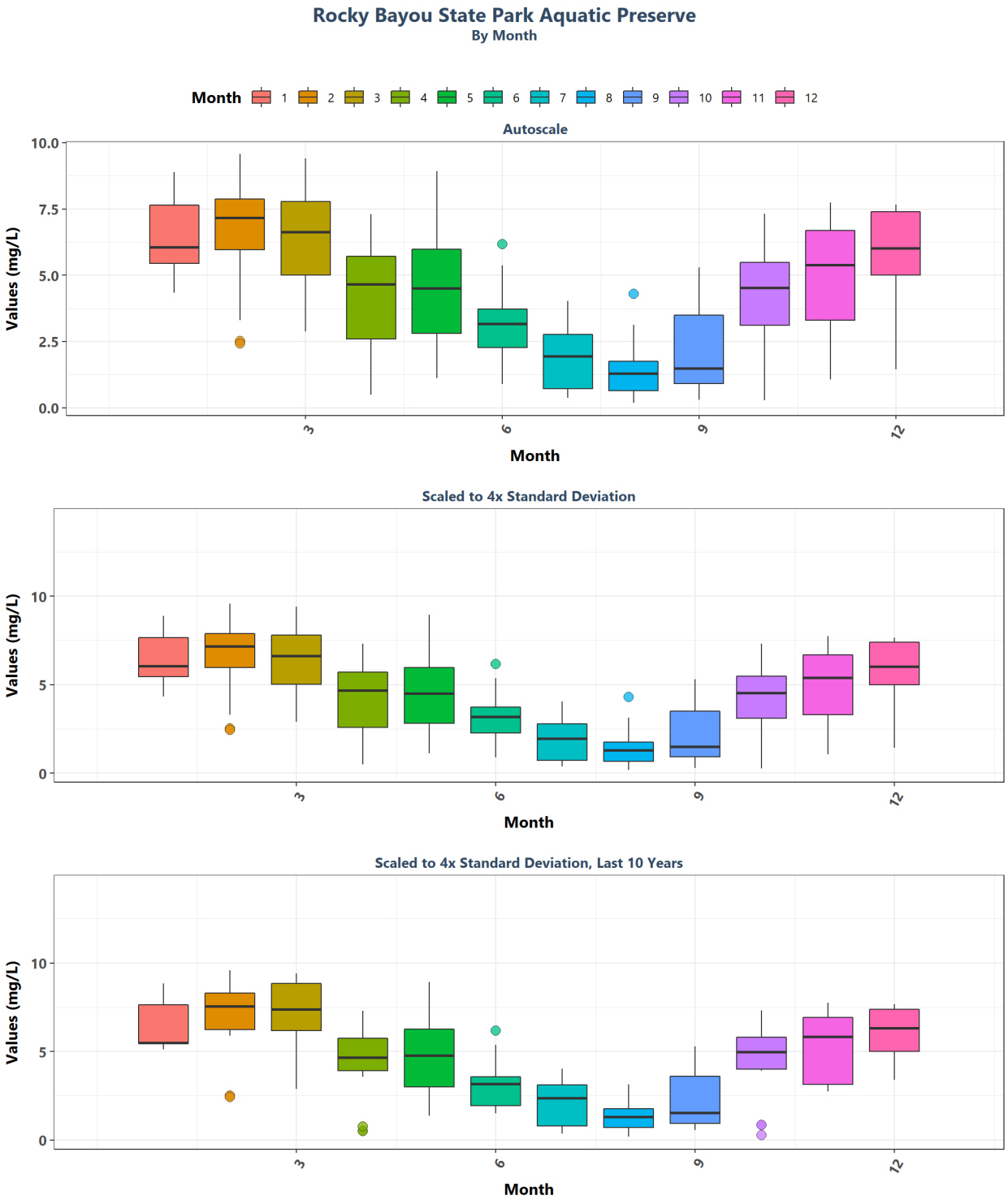

Pinellas County Aquatic Preserve
By Year
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

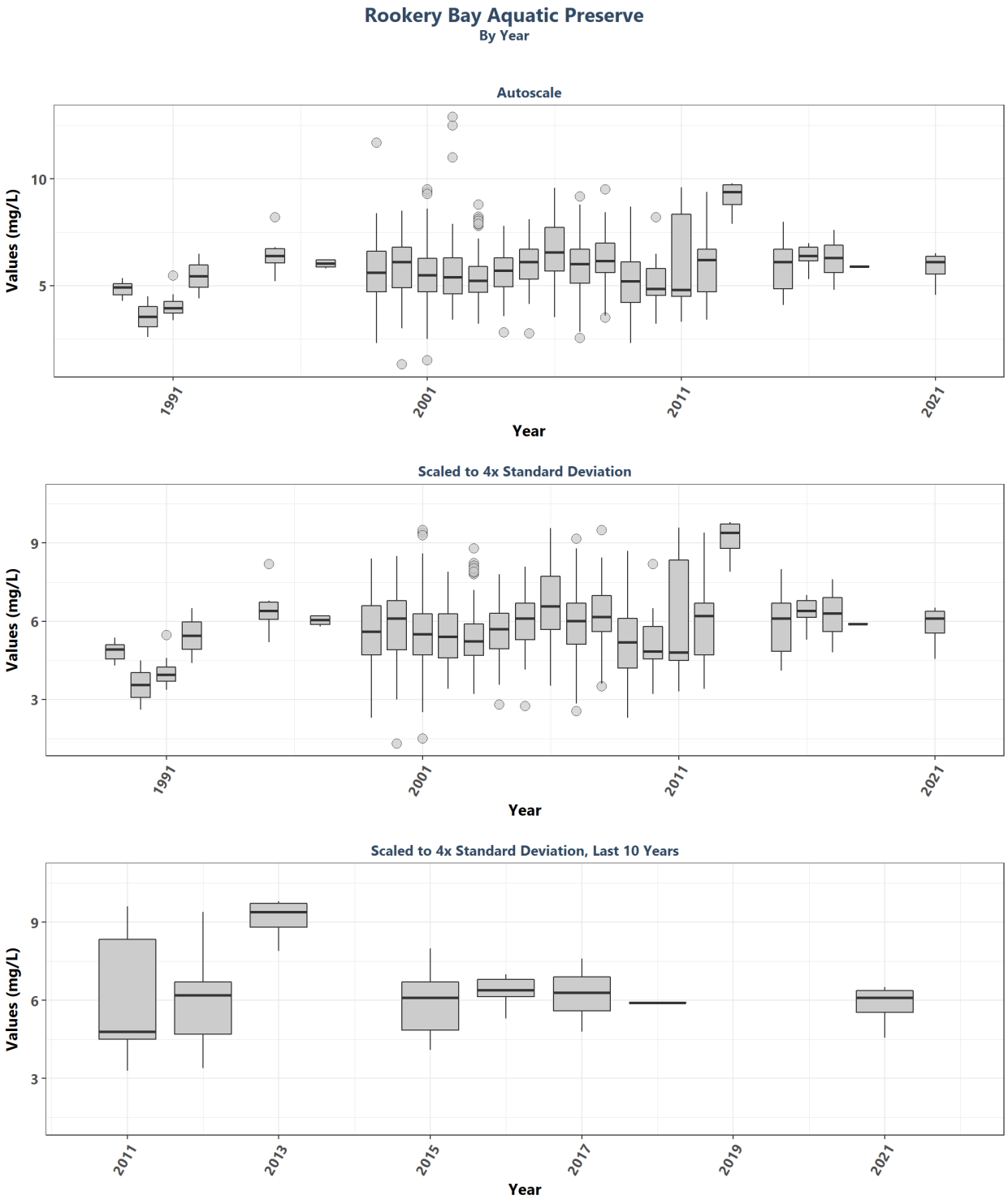

Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

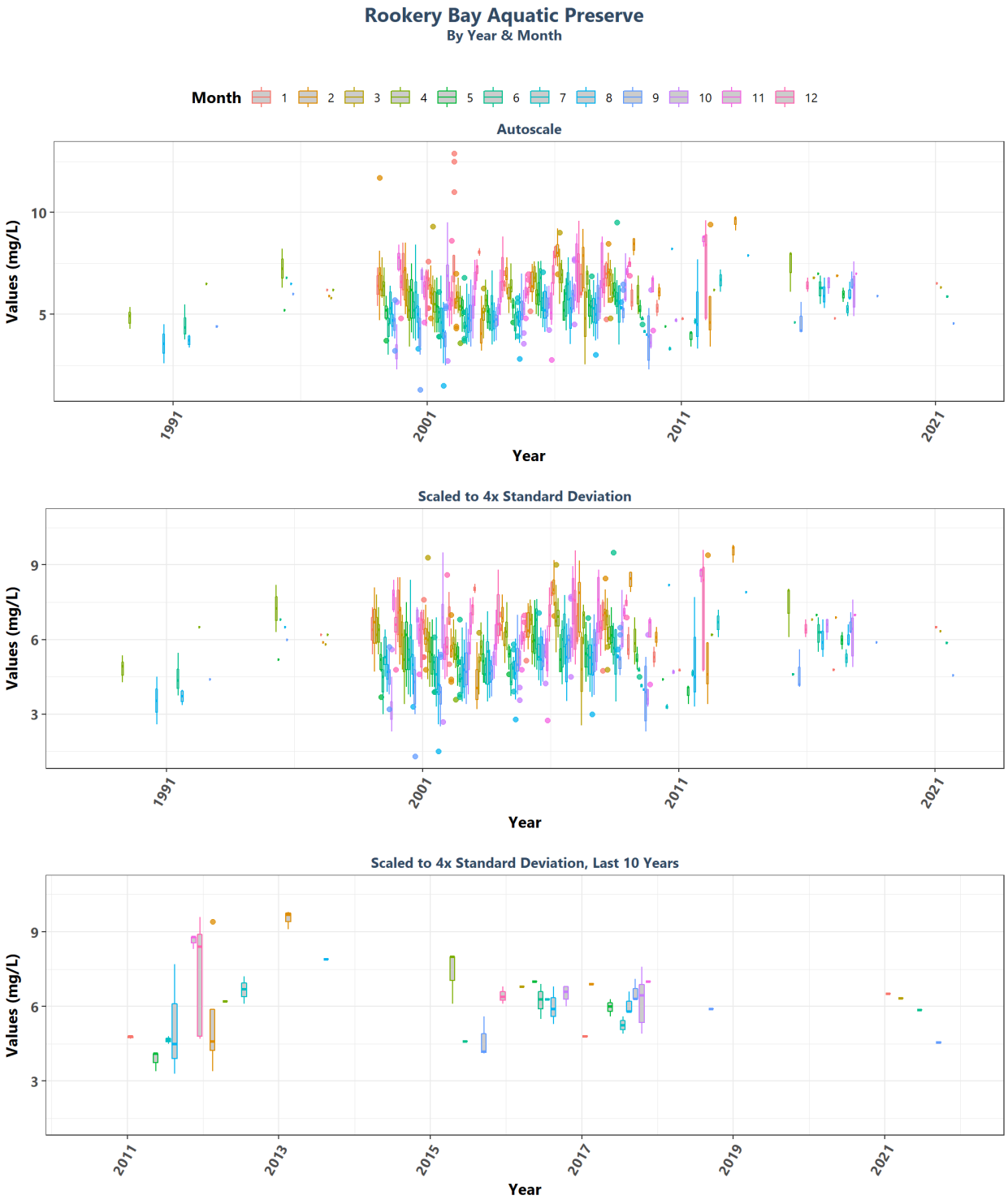

Pinellas County Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

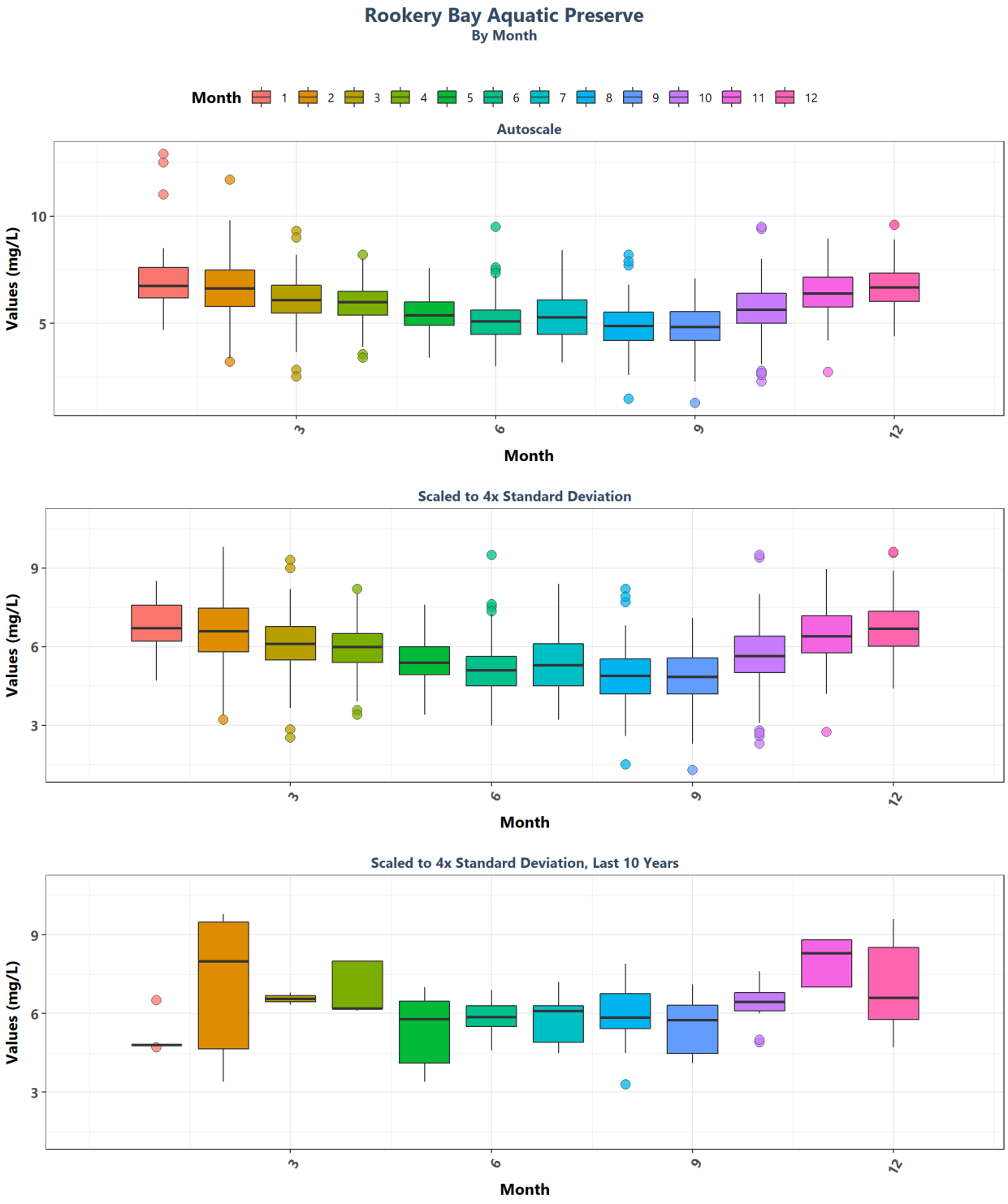

Rocky Bayou State Park Aquatic Preserve
By Year
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
1998
2003
2008
2013
2018
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1998
2003
2008
2013
2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2008
2010
2012
2014
2016
2018
Year

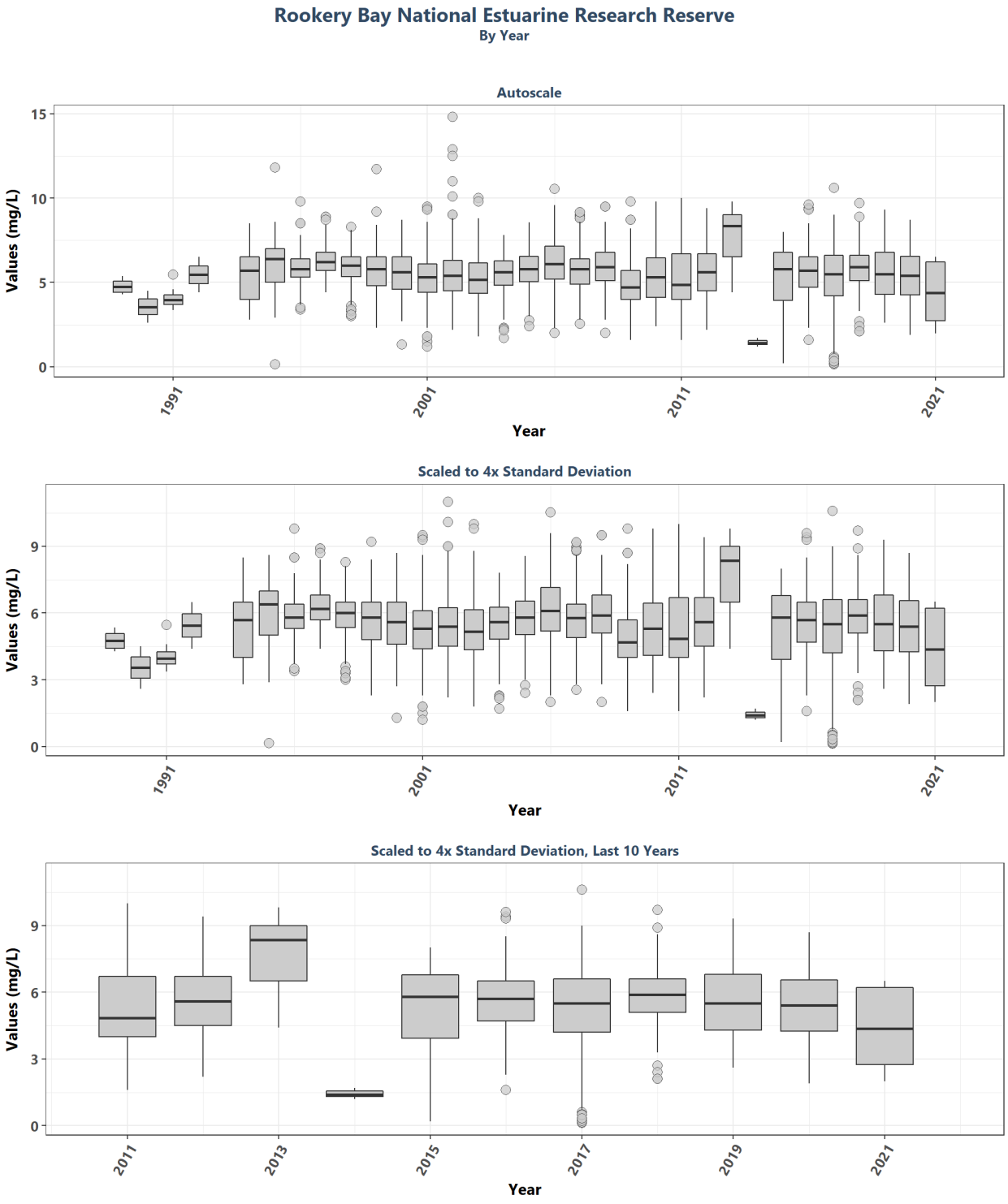

Rocky Bayou State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
1998
2003
2008
2013
2018
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1998
2003
2008
2013
2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2008
2010
2012
2014
2016
2018
Year

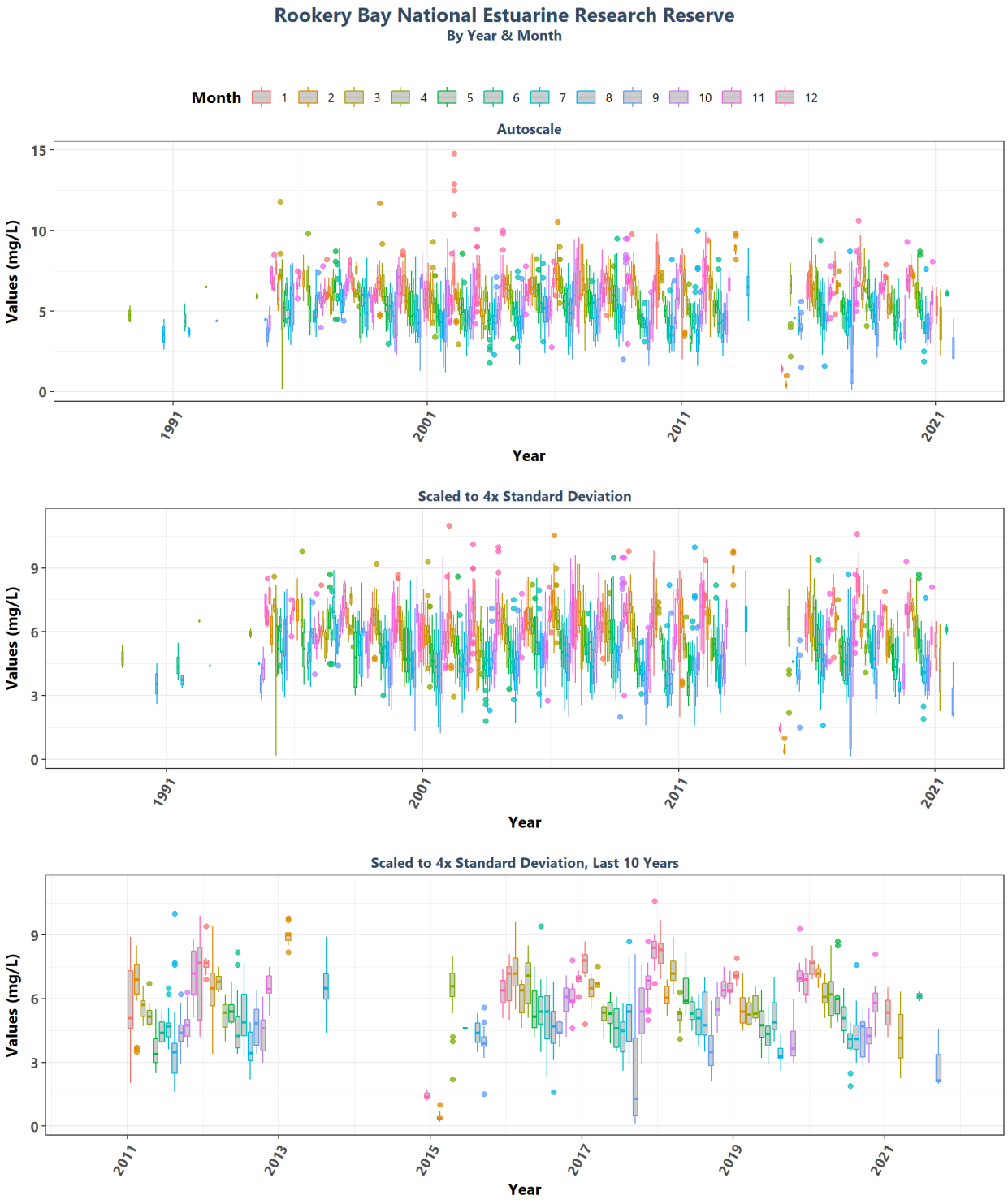

Rocky Bayou State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

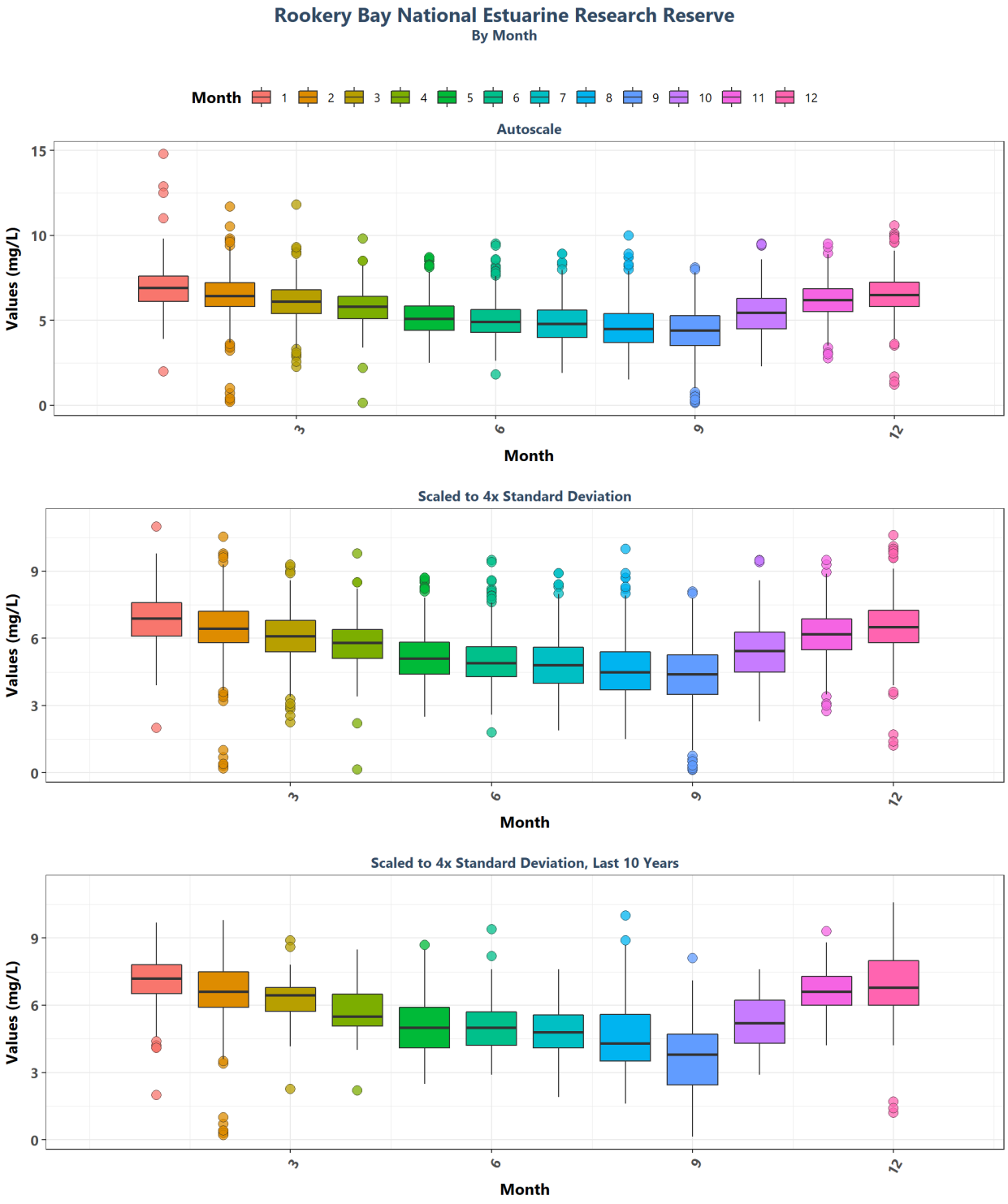

Rookery Bay Aquatic Preserve
By Year
Autoscale
Values (mg/L)
10
5
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
9
6
3
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
9
6
3
2011
2013
2015
2017
2019
2021
Year

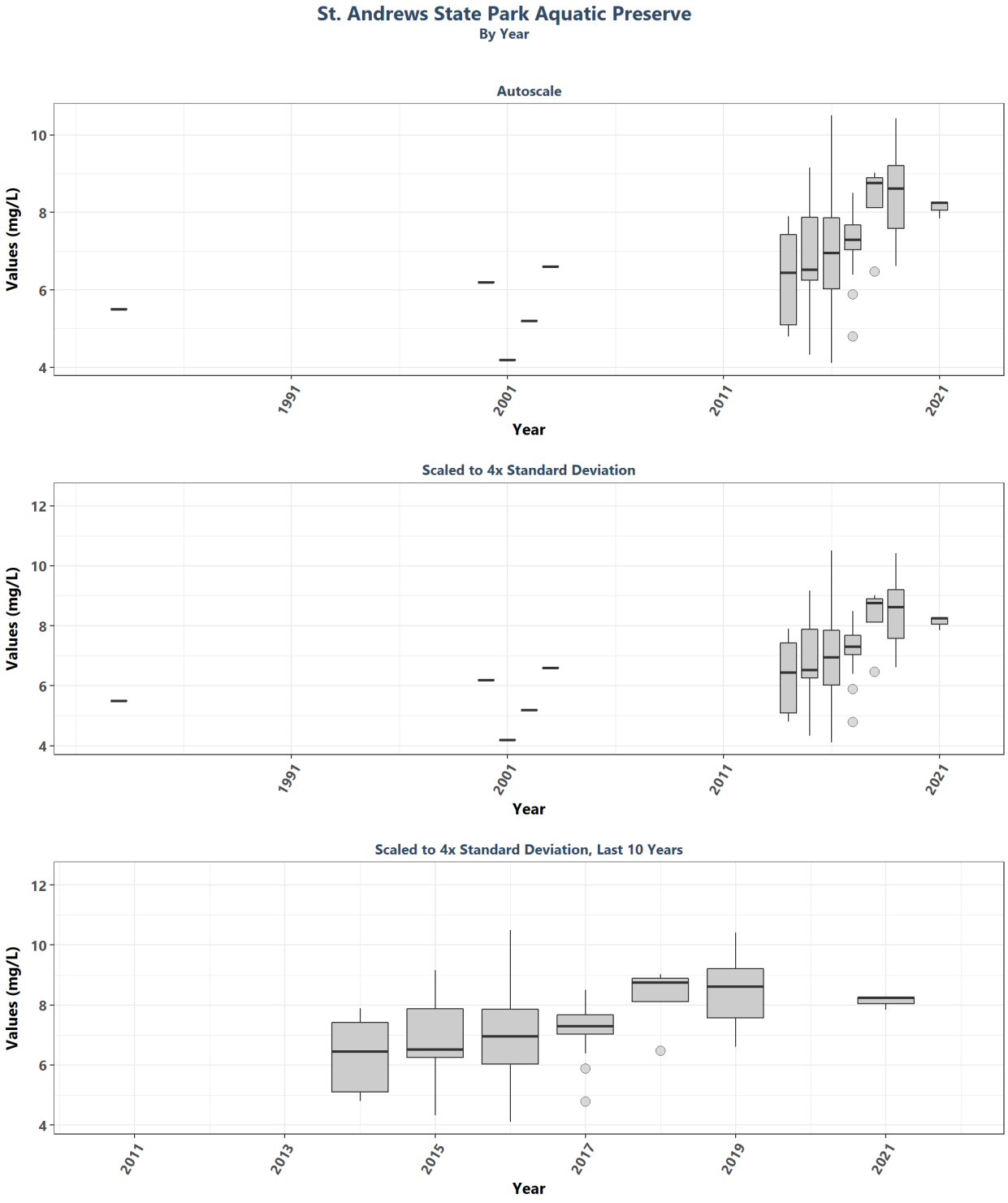

Rookery Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
9
6
3
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
9
6
3
2011
2013
2015
2017
2019
2021
Year

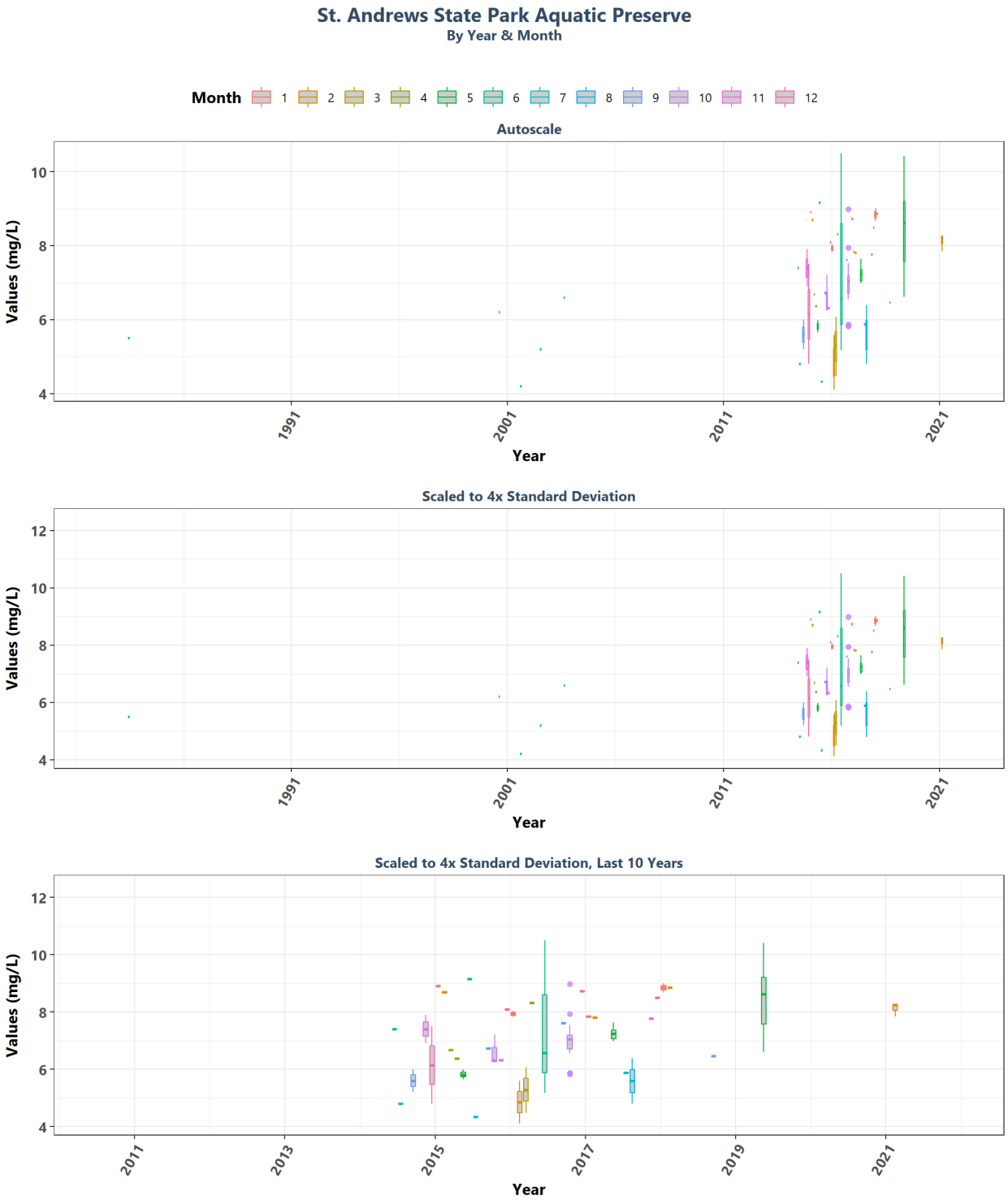

Rookery Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
9
6
3
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
9
6
3
Month

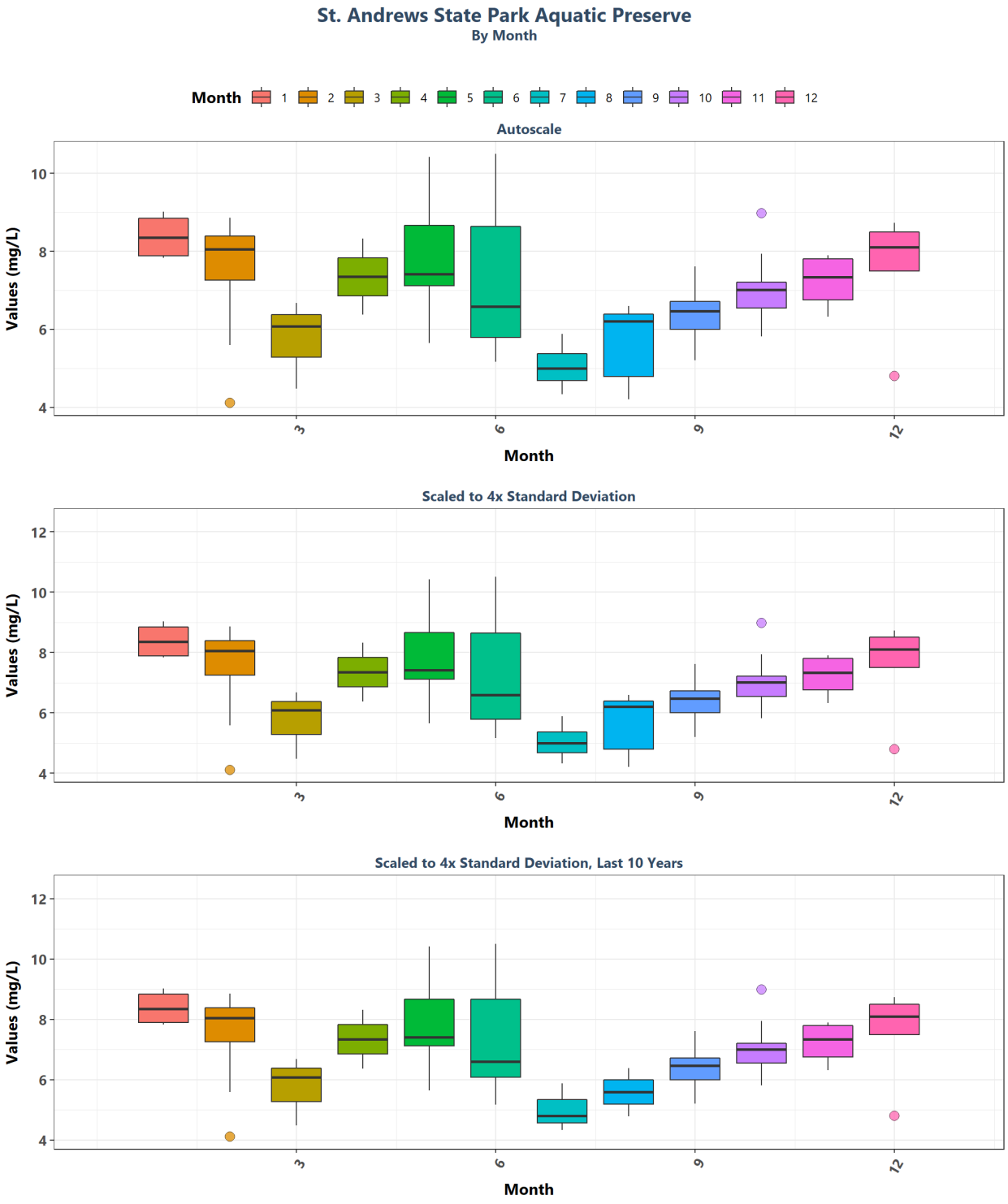

Rookery Bay National Estuarine Research Reserve
By Year
Autoscale
Values (mg/L)
15
10
5
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
9
6
3
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
9
6
3
0
2011
2013
2015
2017
2019
2021
Year

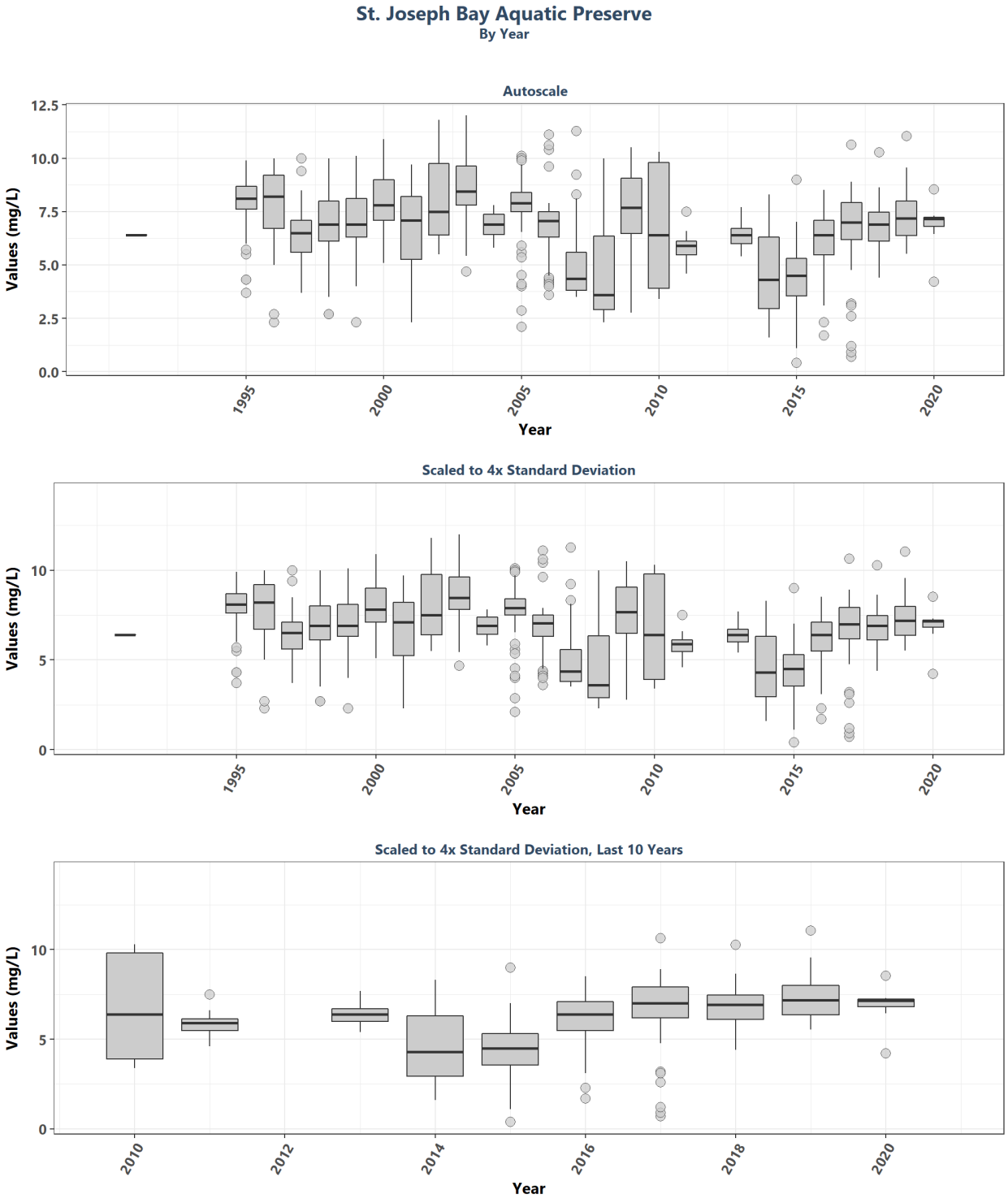

Rookery Bay National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

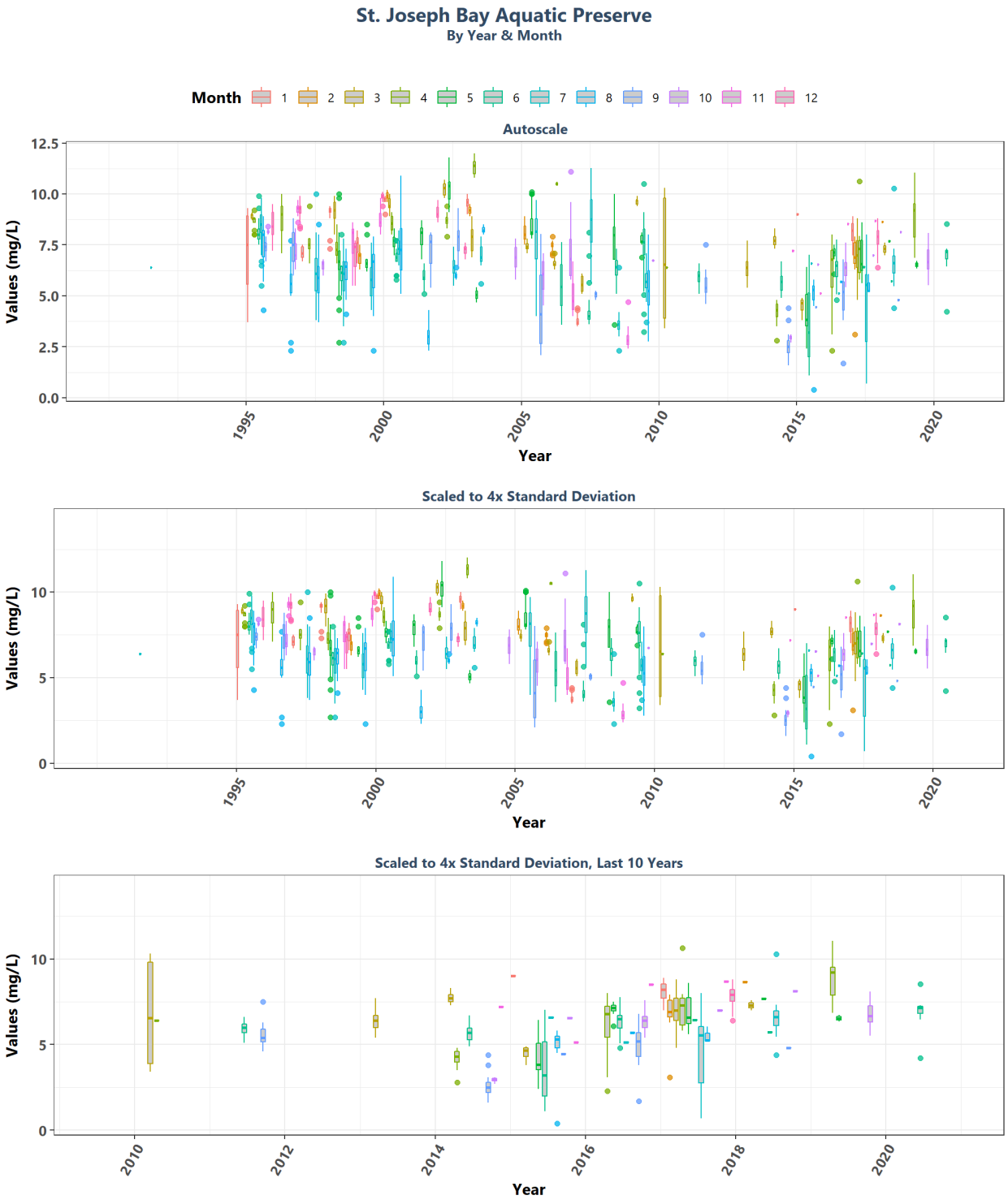

Rookery Bay National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
9
6
3
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
9
6
3
0
3
6
9
12
Month

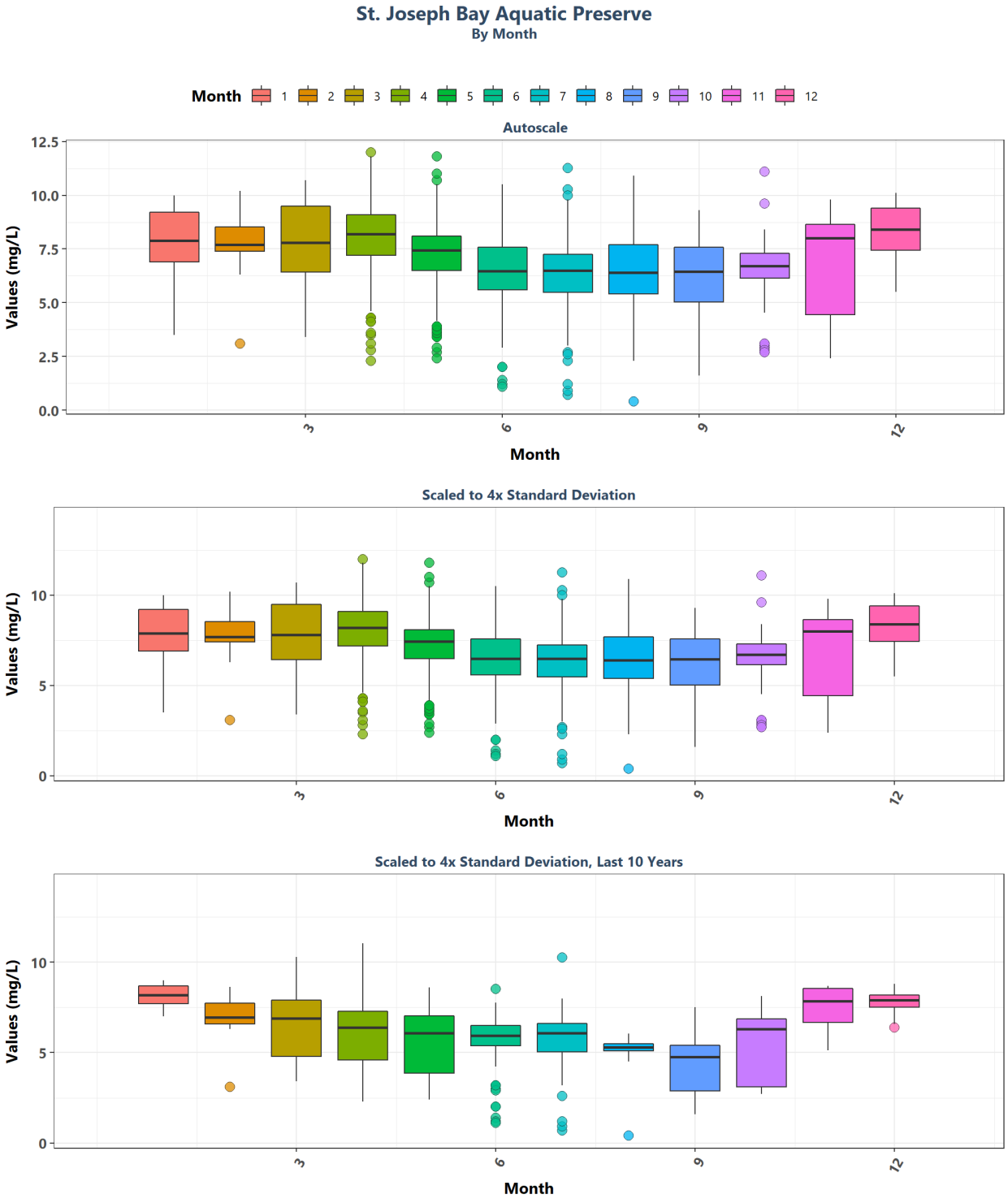

St. Andrews State Park Aquatic Preserve
By Year
Autoscale
Values (mg/L)
10
8
6
4
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12
10
8
6
4
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12
10
8
6
4
2011
2013
2015
2017
2019
2021
Year

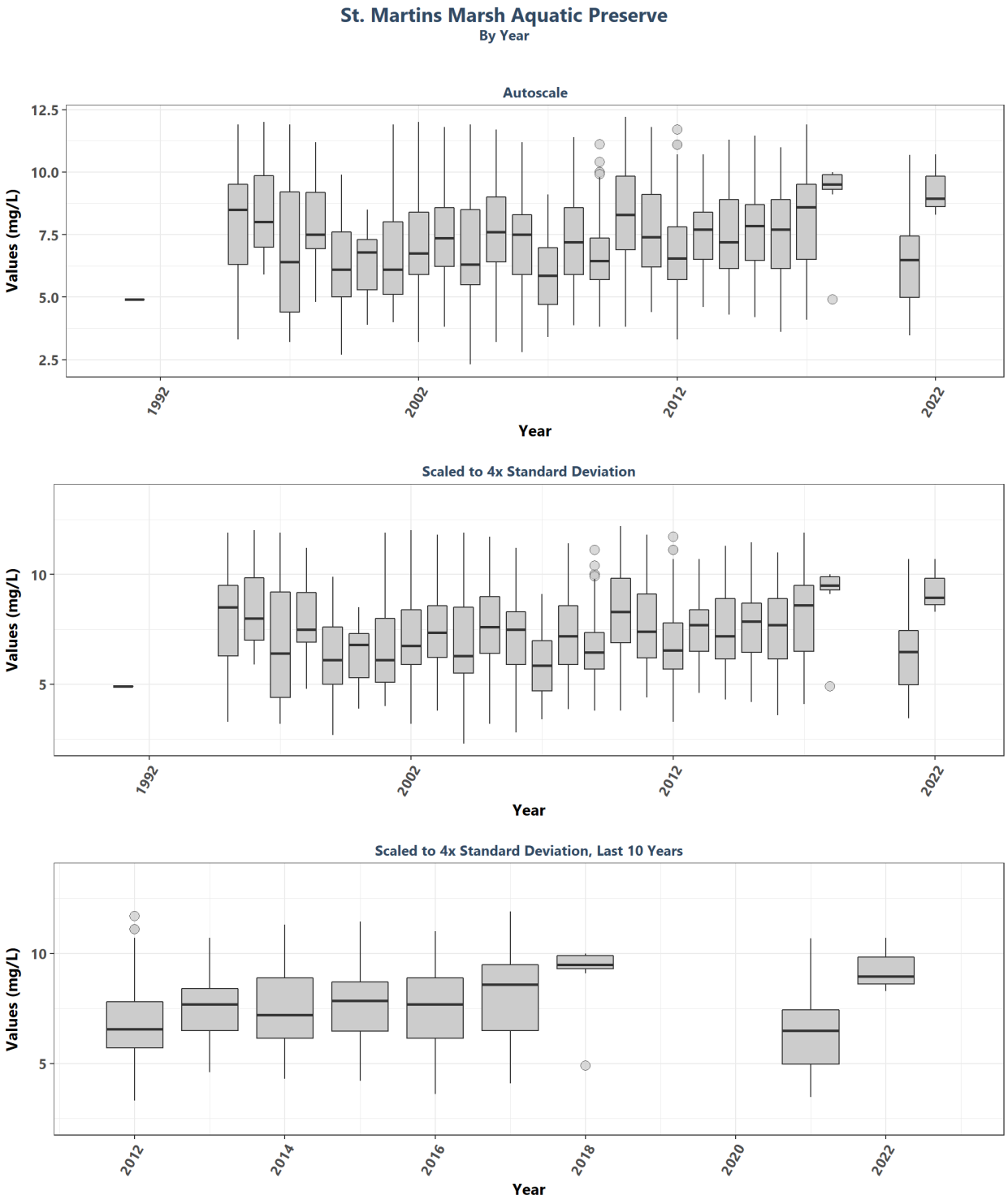

St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Year
Scaled to 4x Standard Deviation
Values (mg/L)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Year

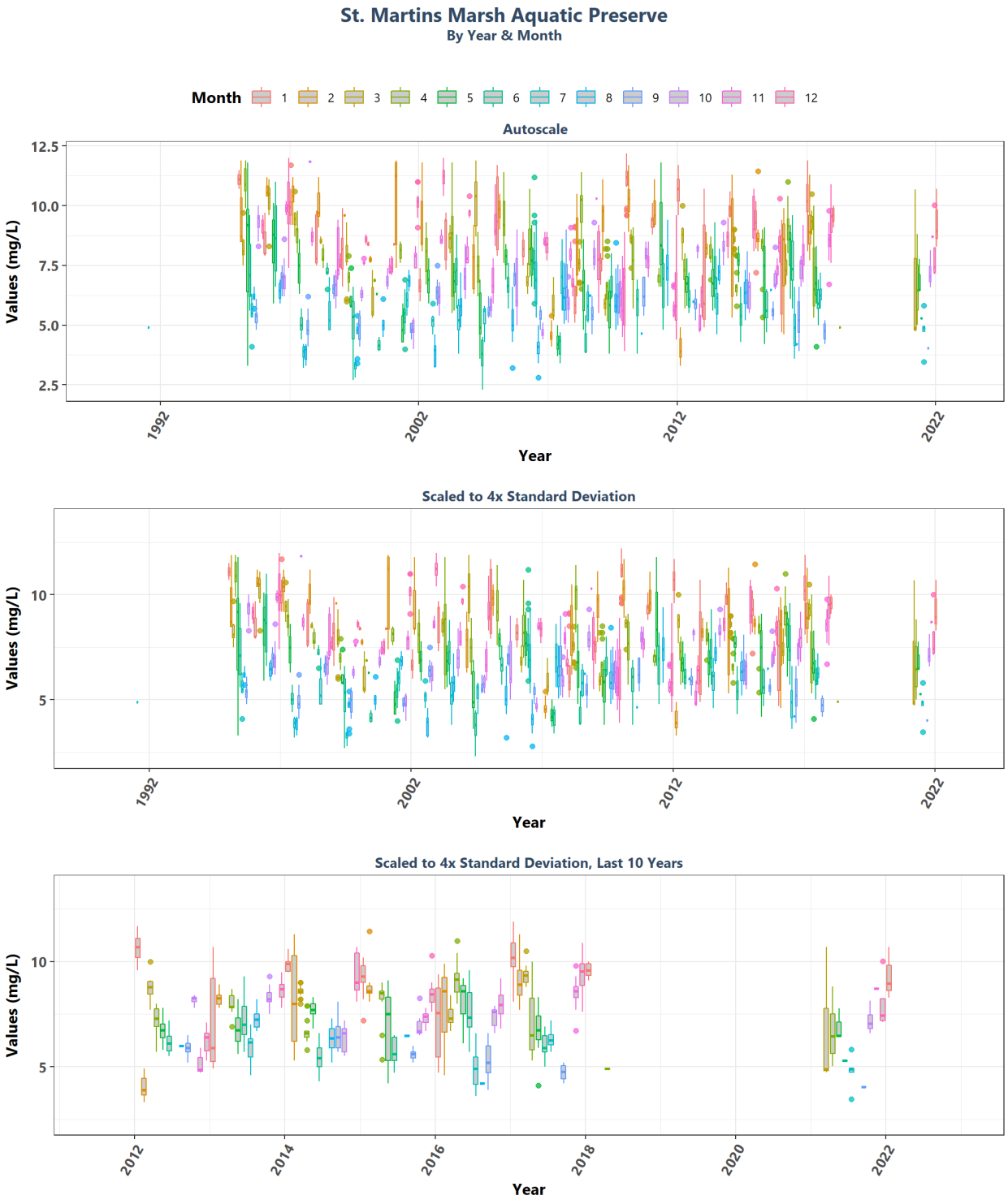

St. Andrews State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
Month
Scaled to 4x Standard Deviation
Values (mg/L)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
Month

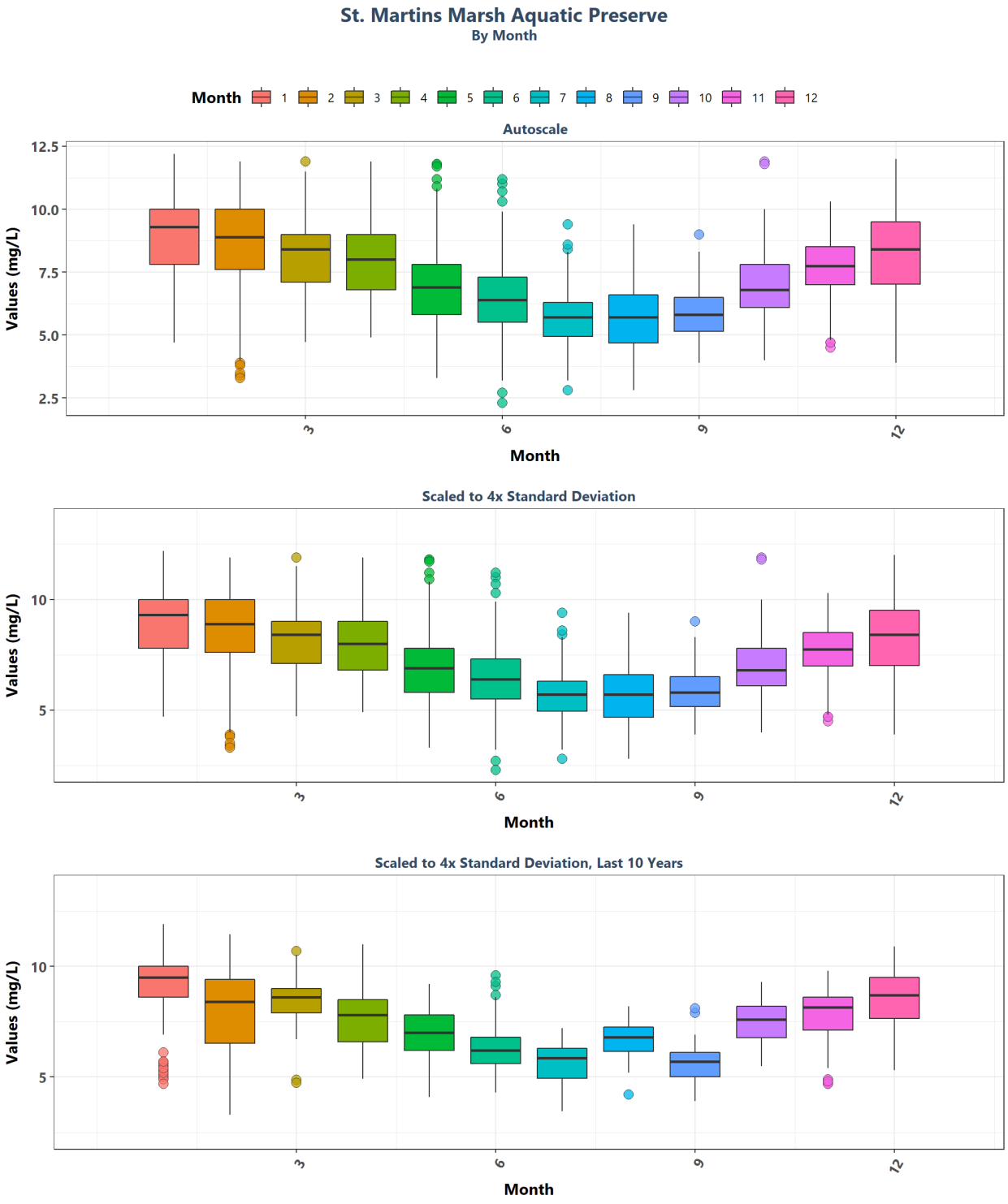

St. Joseph Bay Aquatic Preserve
By Year
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2010
2012
2014
2016
2018
2020
Year


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)
12.5
10.0
7.5
5.0
2.5
0.0
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
2010
2012
2014
2016
2018
2020
Year


St. Joseph Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
0
3
6
9
12
Month

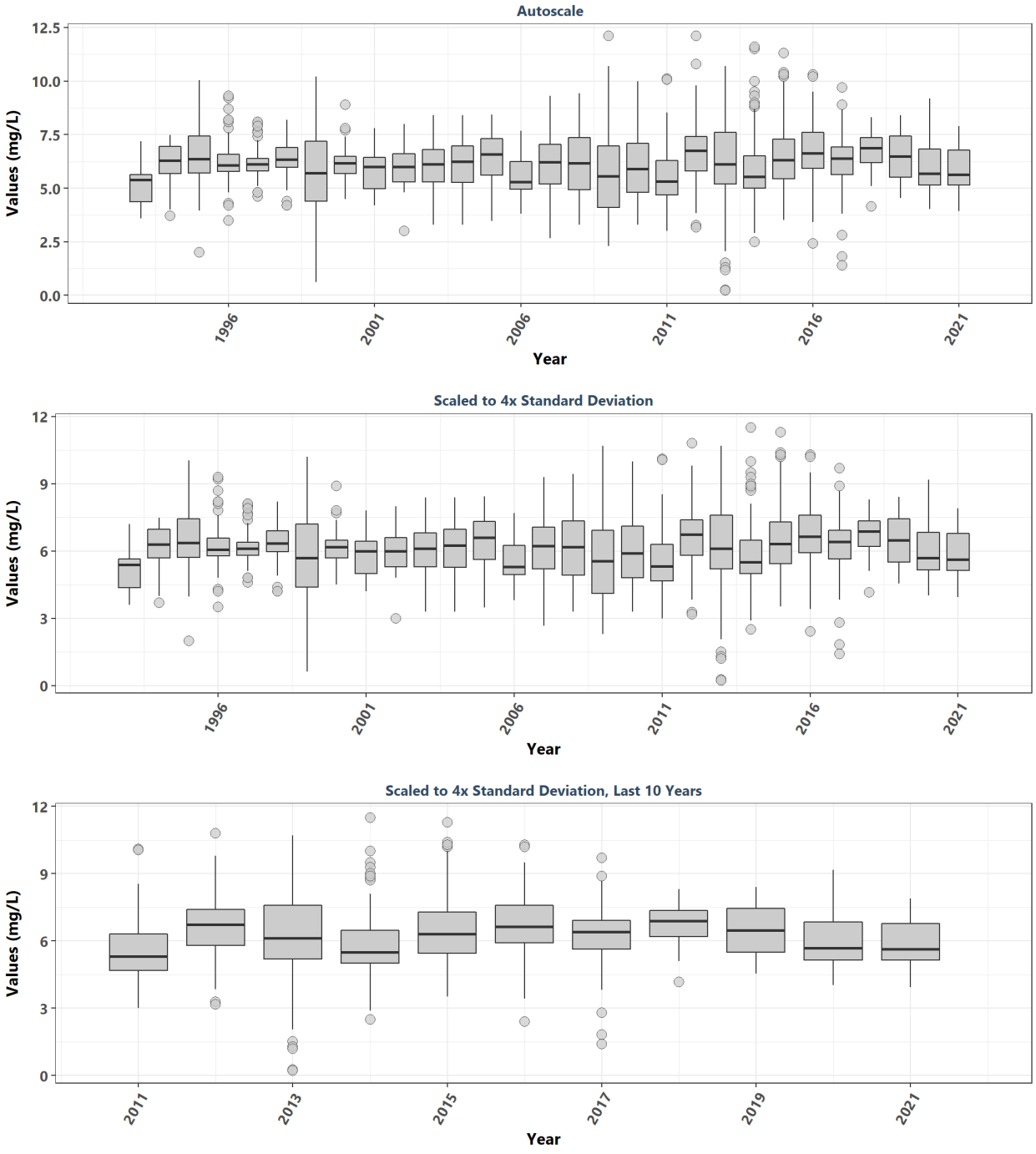

St. Martins Marsh Aquatic Preserve
By Year
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
2012
2014
2016
2018
2020
2022
Year

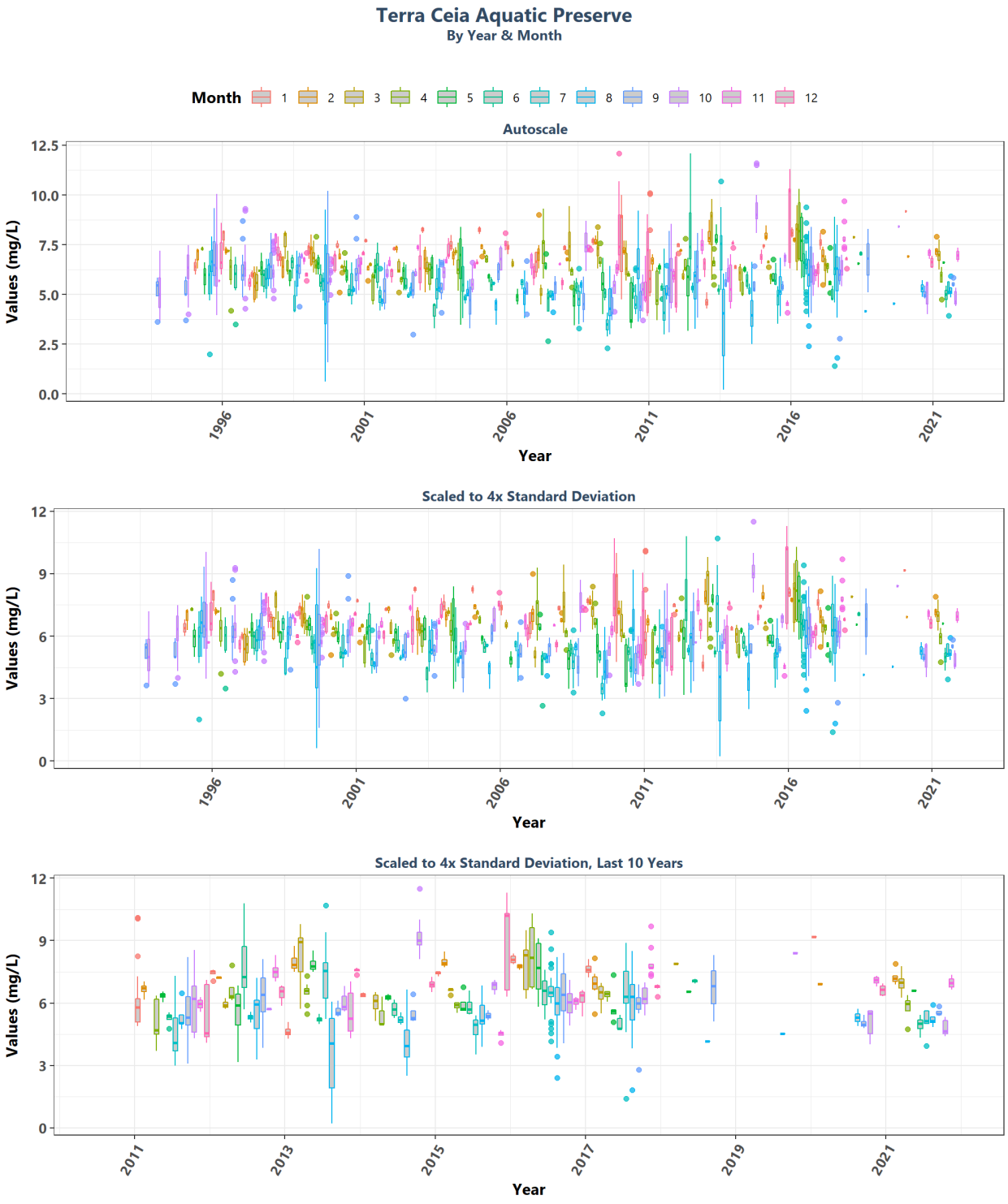

St. Martins Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
2012
2014
2016
2018
2020
2022
Year

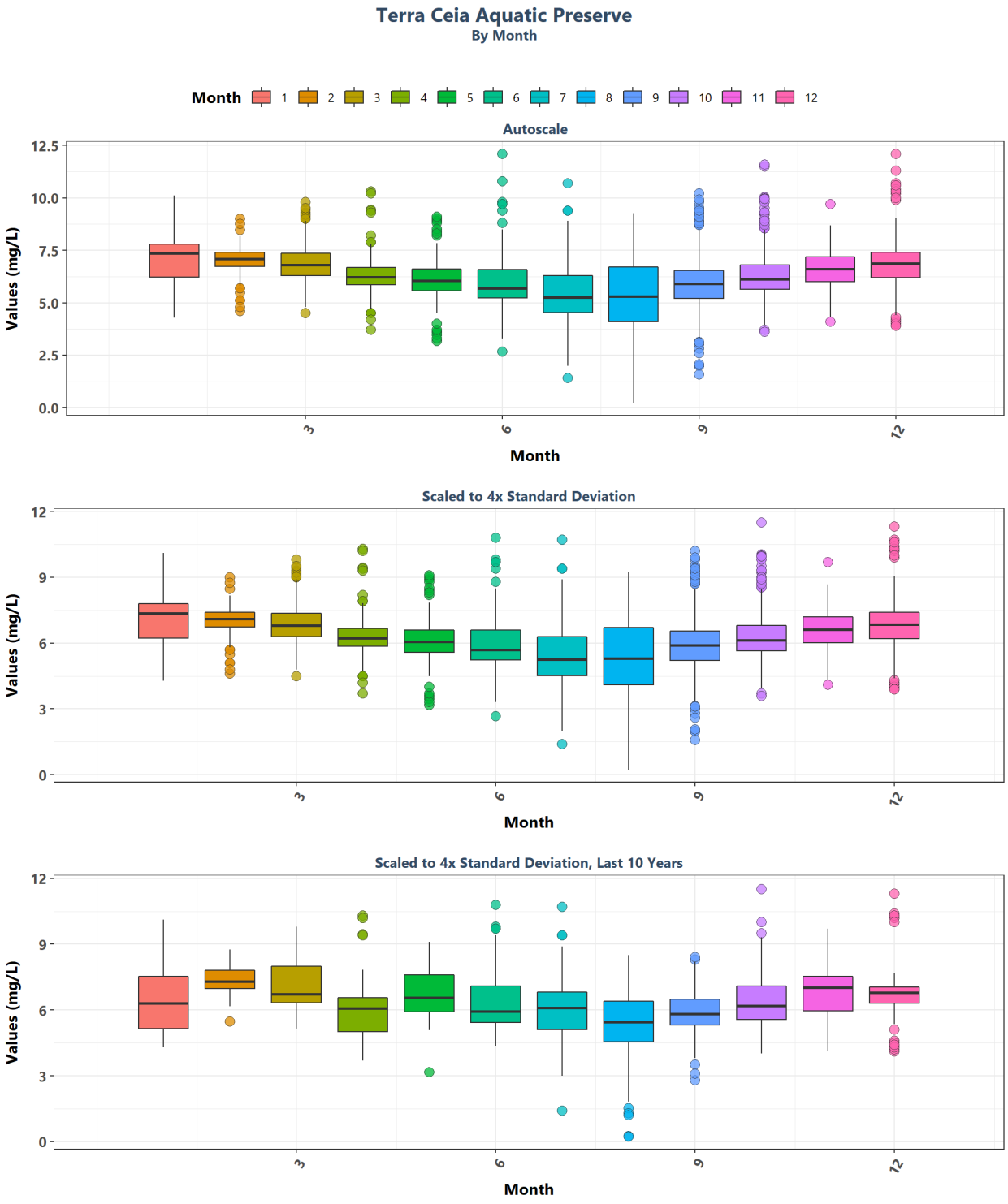

St. Martins Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
10
5
3
6
9
12
Month

Terra Ceia Aquatic Preserve

By Year

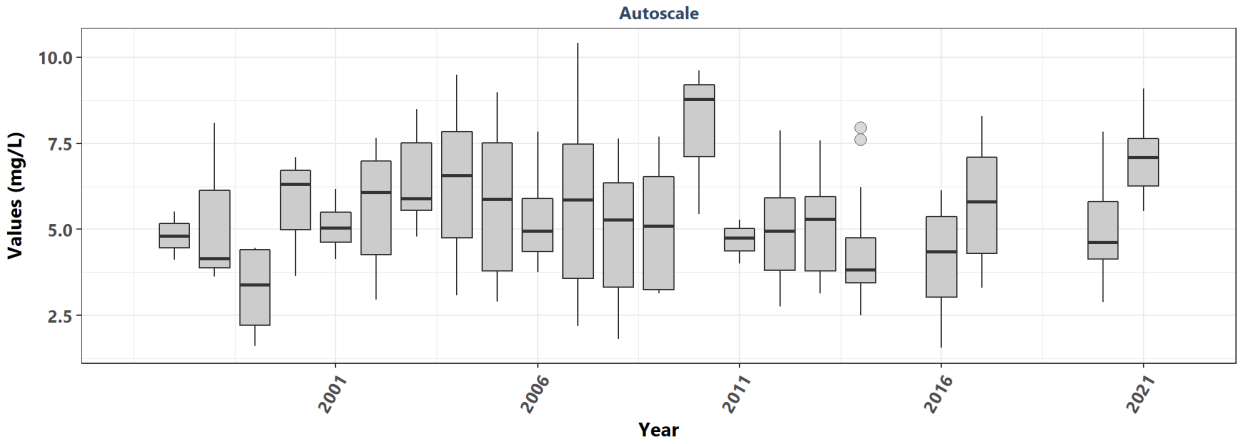


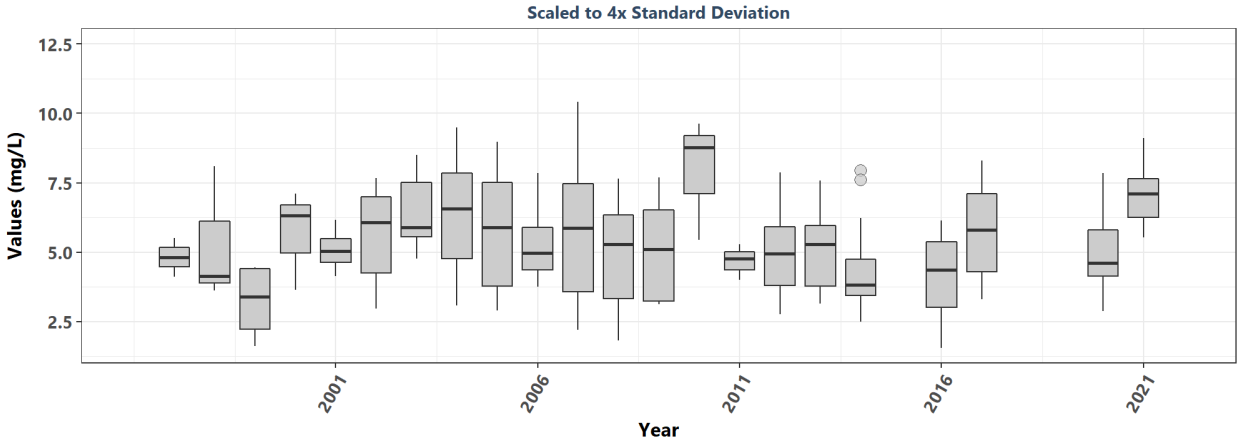

Terra Ceia Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12
9
6
3
0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12
9
6
3
0
2011
2013
2015
2017
2019
2021
Year

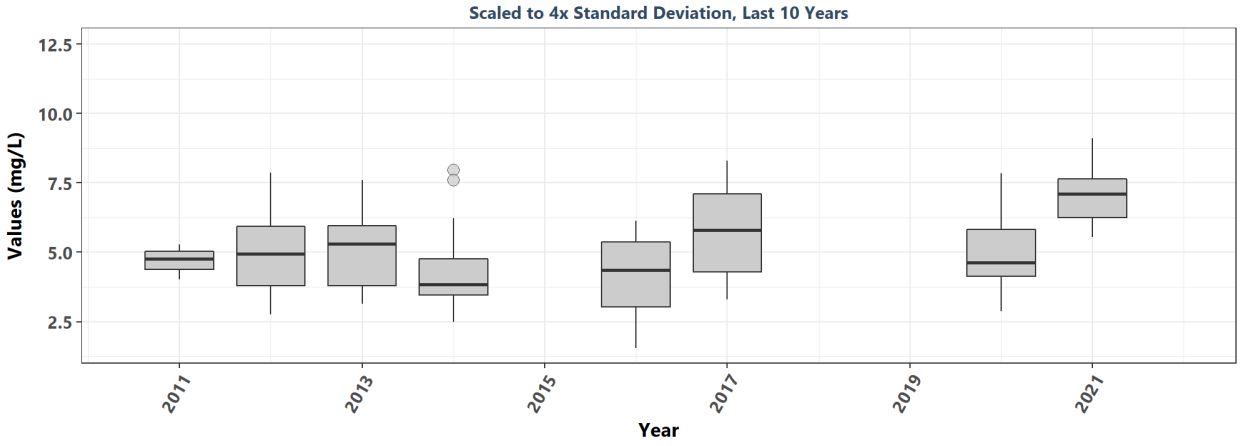

Terra Ceia Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12
9
6
3
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12
9
6
3
0
3
6
9
12
Month

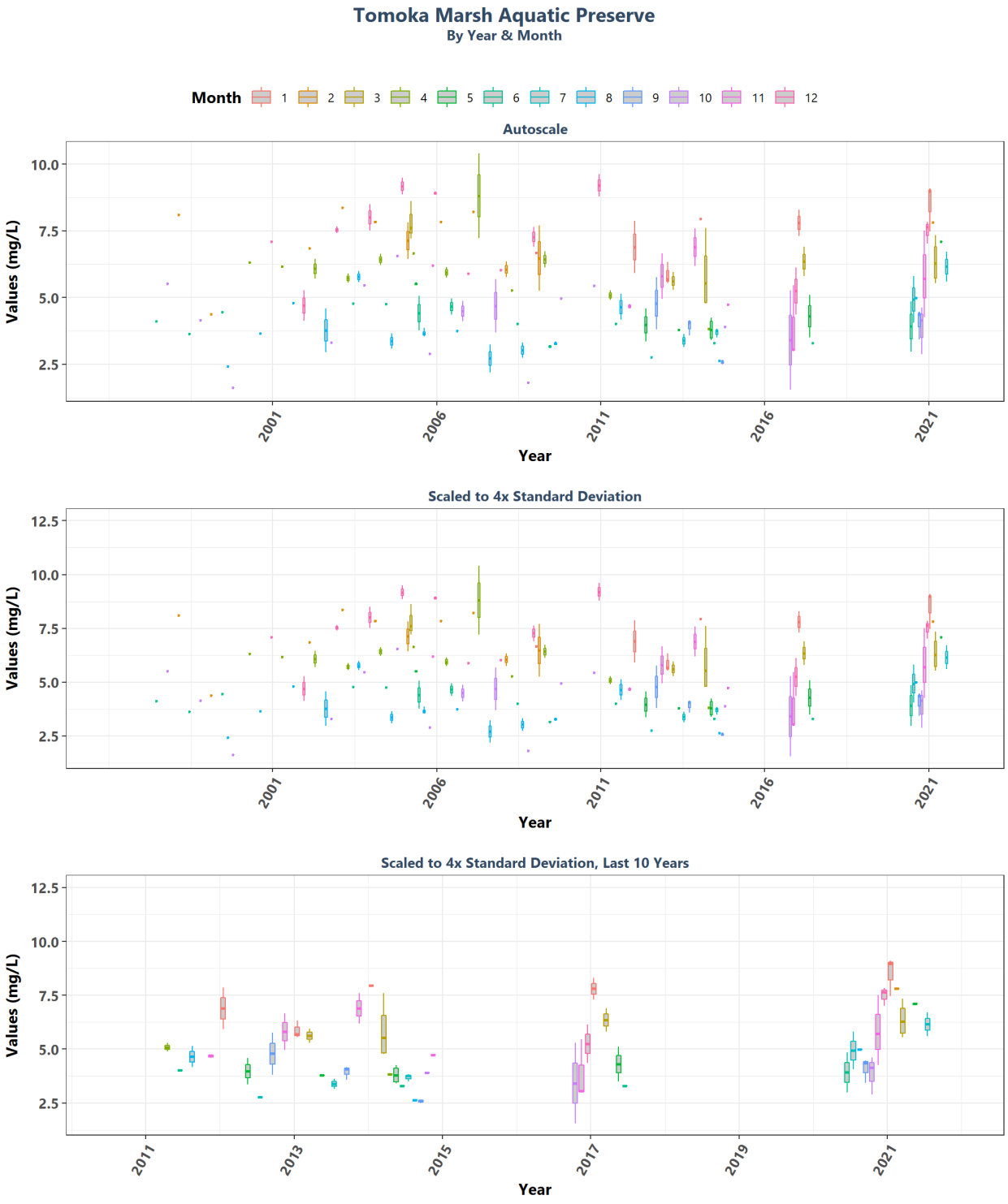
Tomoka Marsh Aquatic Preserve

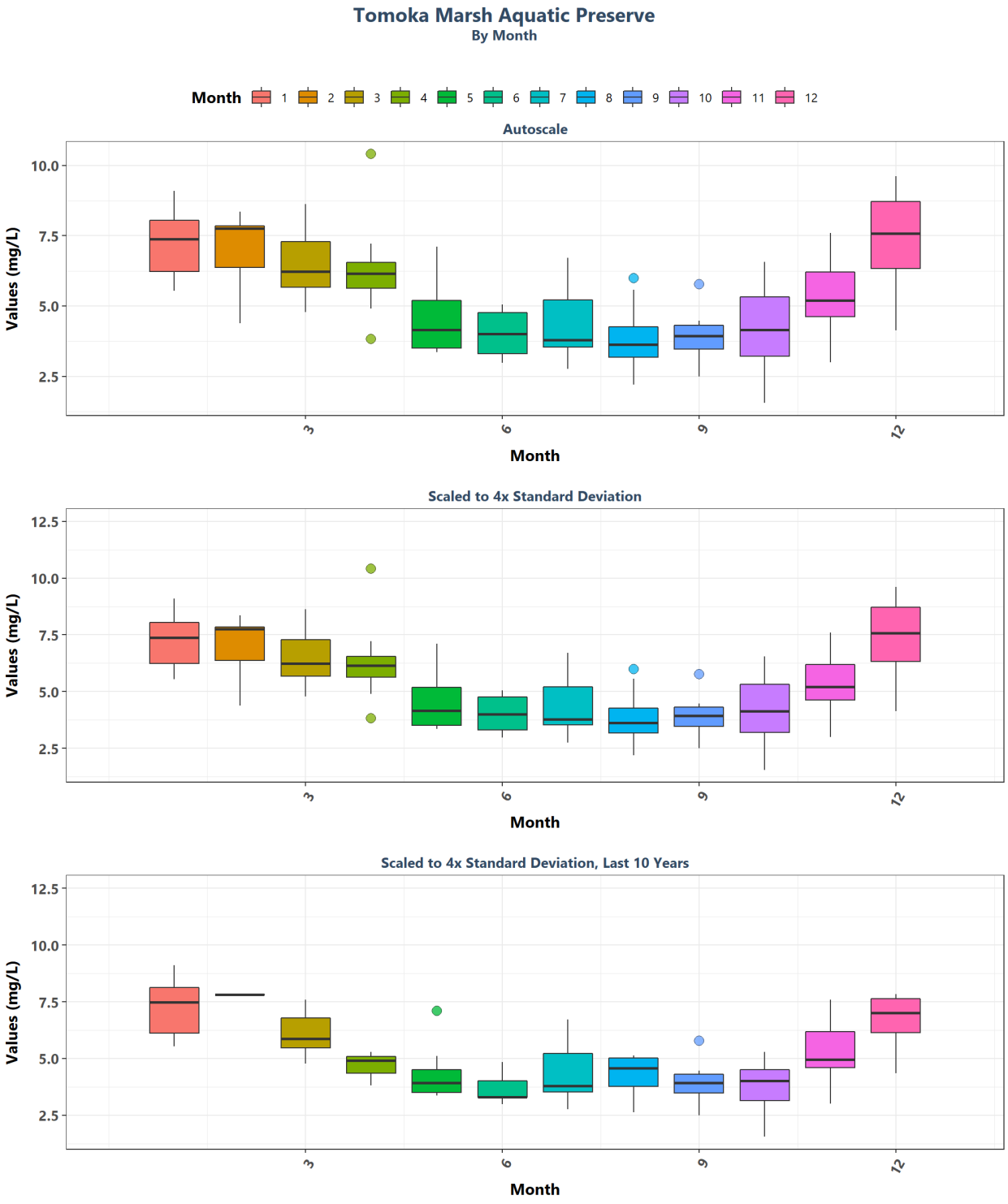
By Year

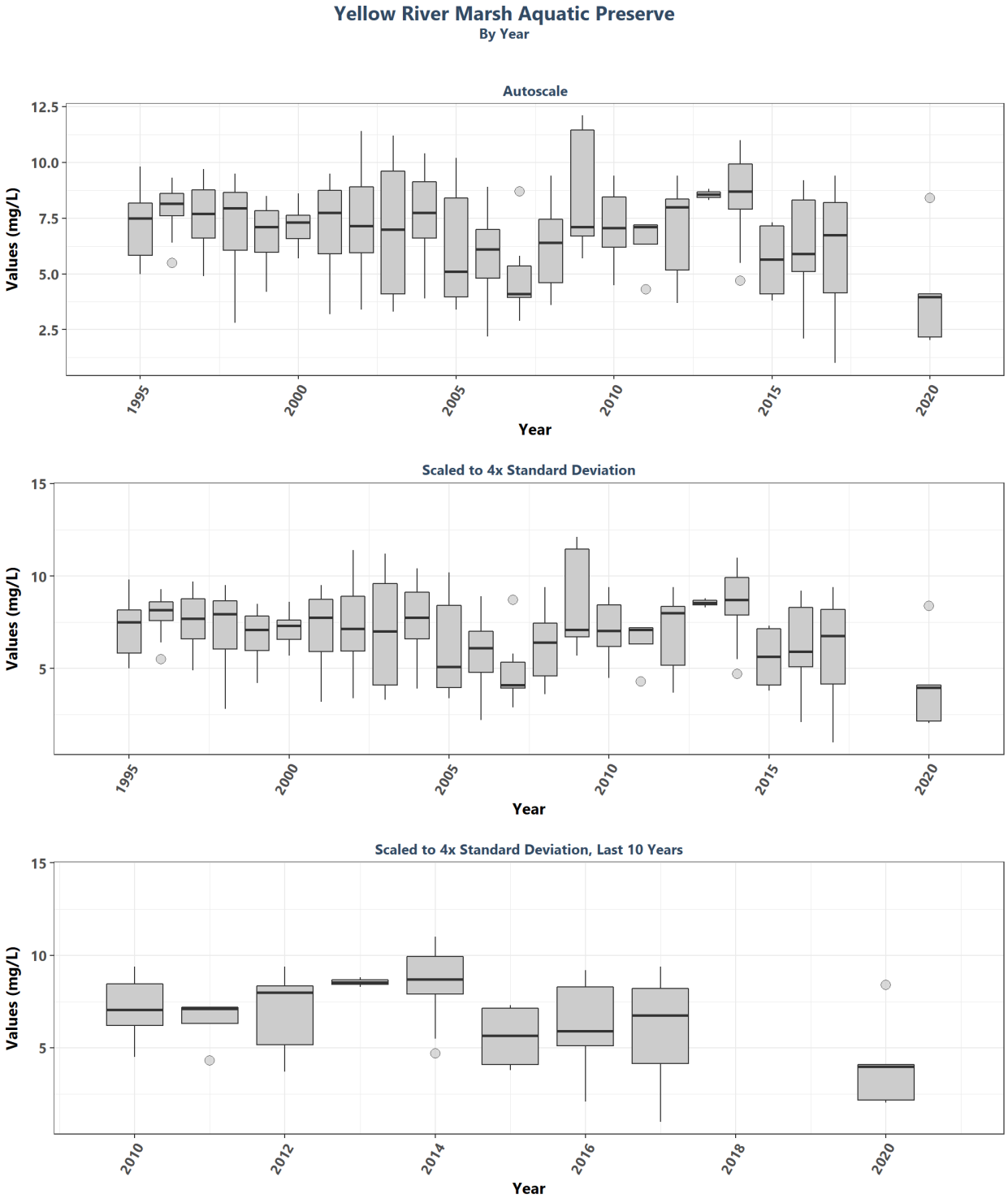


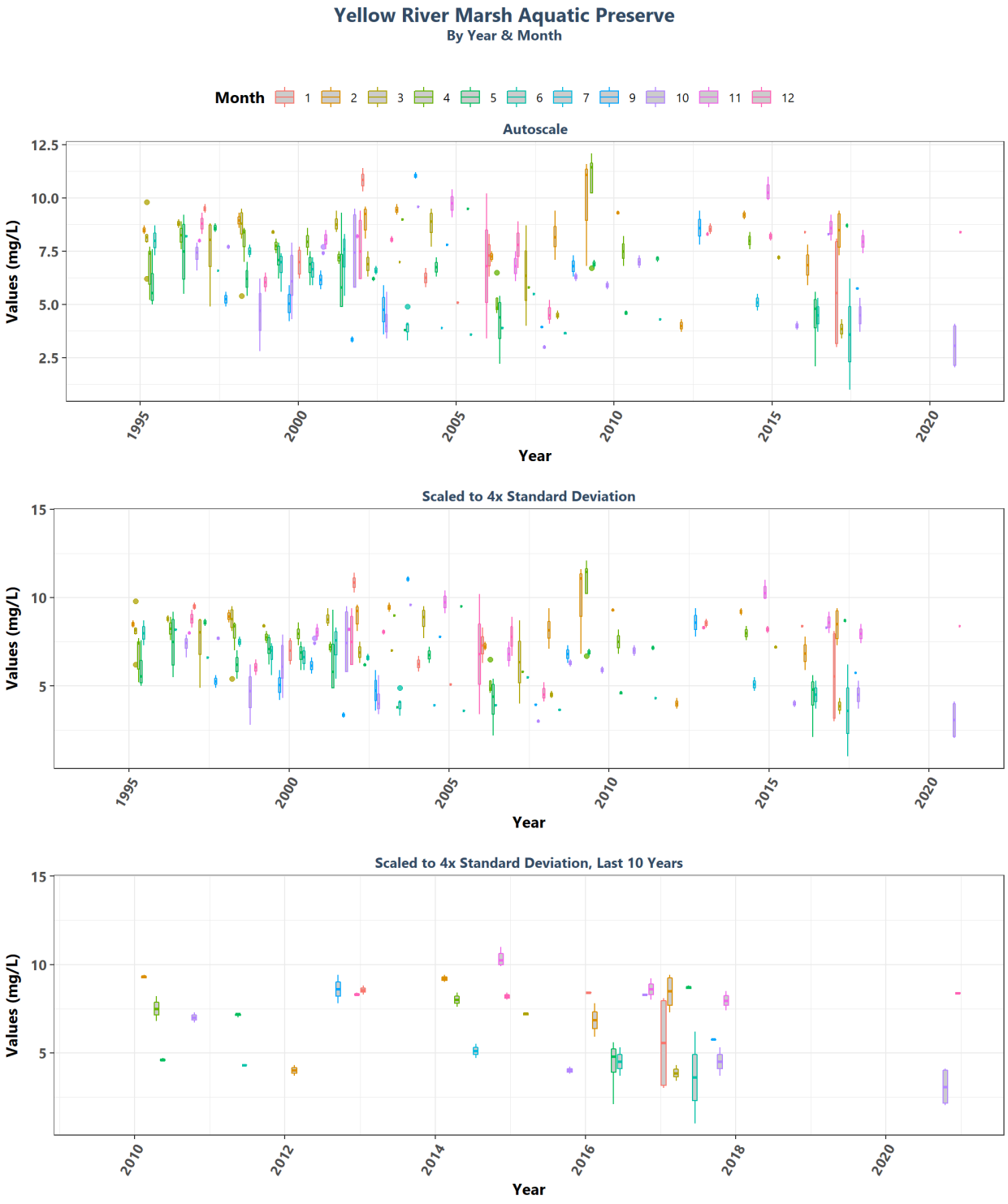


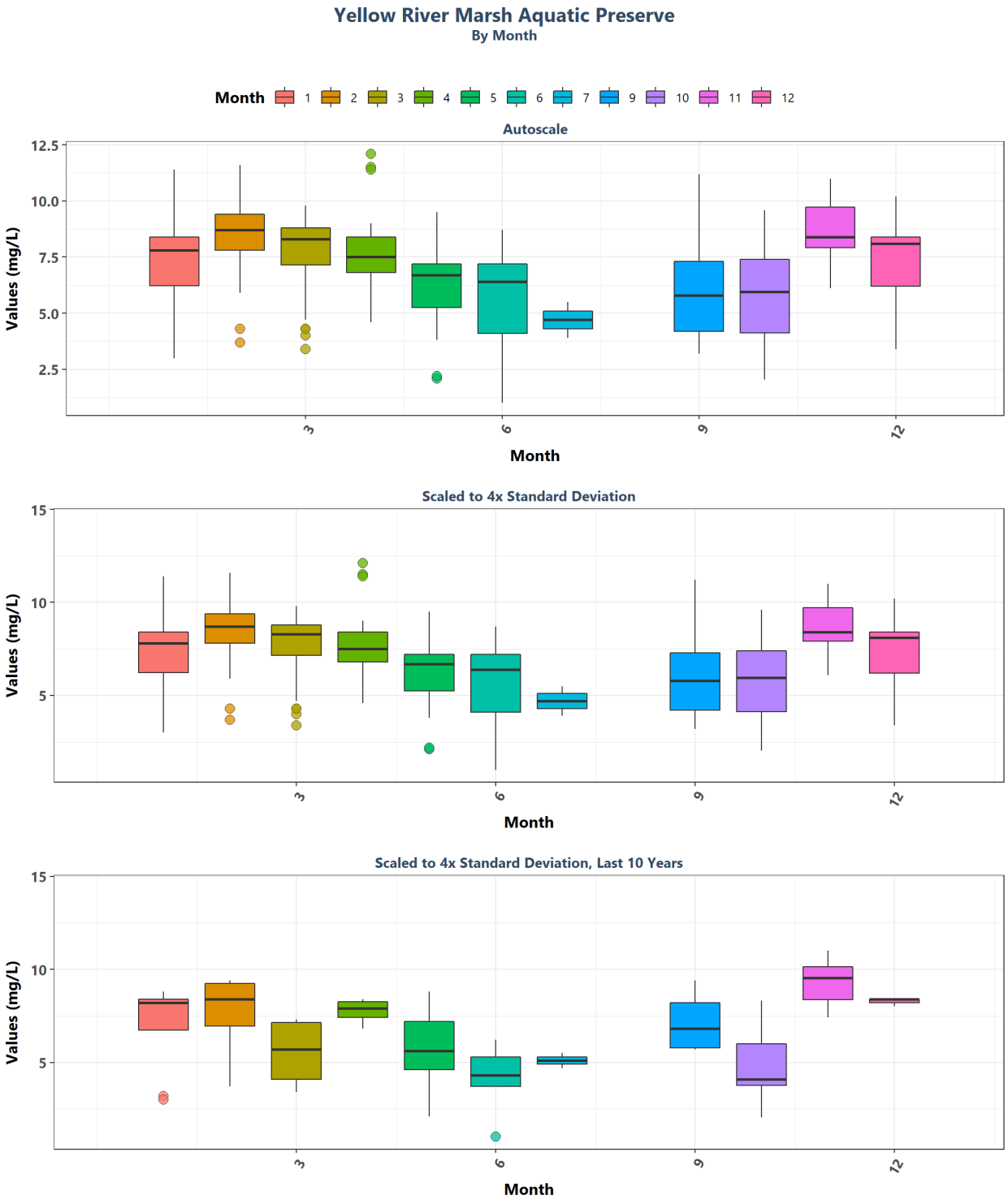



Tomoka Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
2011
2013
2015
2017
2019
2021
Year


Tomoka Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (mg/L)
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
3
6
9
12
Month


Yellow River Marsh Aquatic Preserve
By Year
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
15
10
5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
15
10
5
2010
2012
2014
2016
2018
2020
Year


Yellow River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 9 10 11 12
Autoscale
Values (mg/L)
12.5
10.0
7.5
5.0
2.5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (mg/L)
15
10
5
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (mg/L)
15
10
5
2010
2012
2014
2016
2018
2020
Year




```
rm(list = setdiff(ls(), c("all_params", "all_depths", "all_activity", "param_name", "depth", "activity"
```