SEACAR Discrete Water Quality Analysis: Field Bottom pH

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)
```

```r
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: 36555, Number Passed Filter: 36555
## Program 476 H Codes: 0 (0%)
## I Codes: 0 (0%)
## Q Codes: 0 (0%)
## U Codes: 0 (0%)
```

```r
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),
```

```r
             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
```

```r
      }
   }
   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)
```

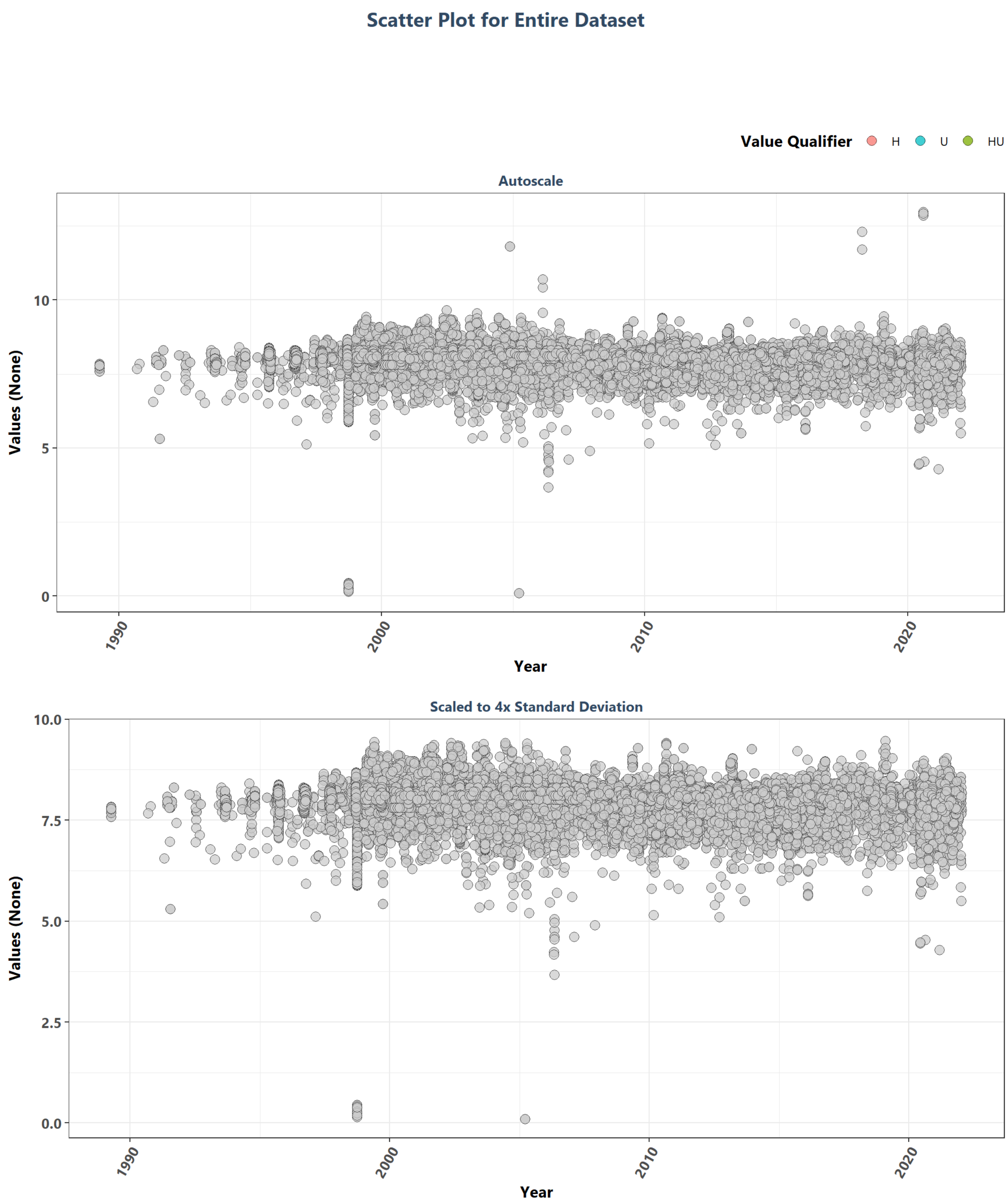
```
KT.Stats2$MonitoringID <- NULL
fwrite(KT.Stats2, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                        "_KendallTau_Stats.txt"), sep="|")
rm(KT.Stats2)
data <- data[!is.na(data$ResultValue),]
```

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",
```

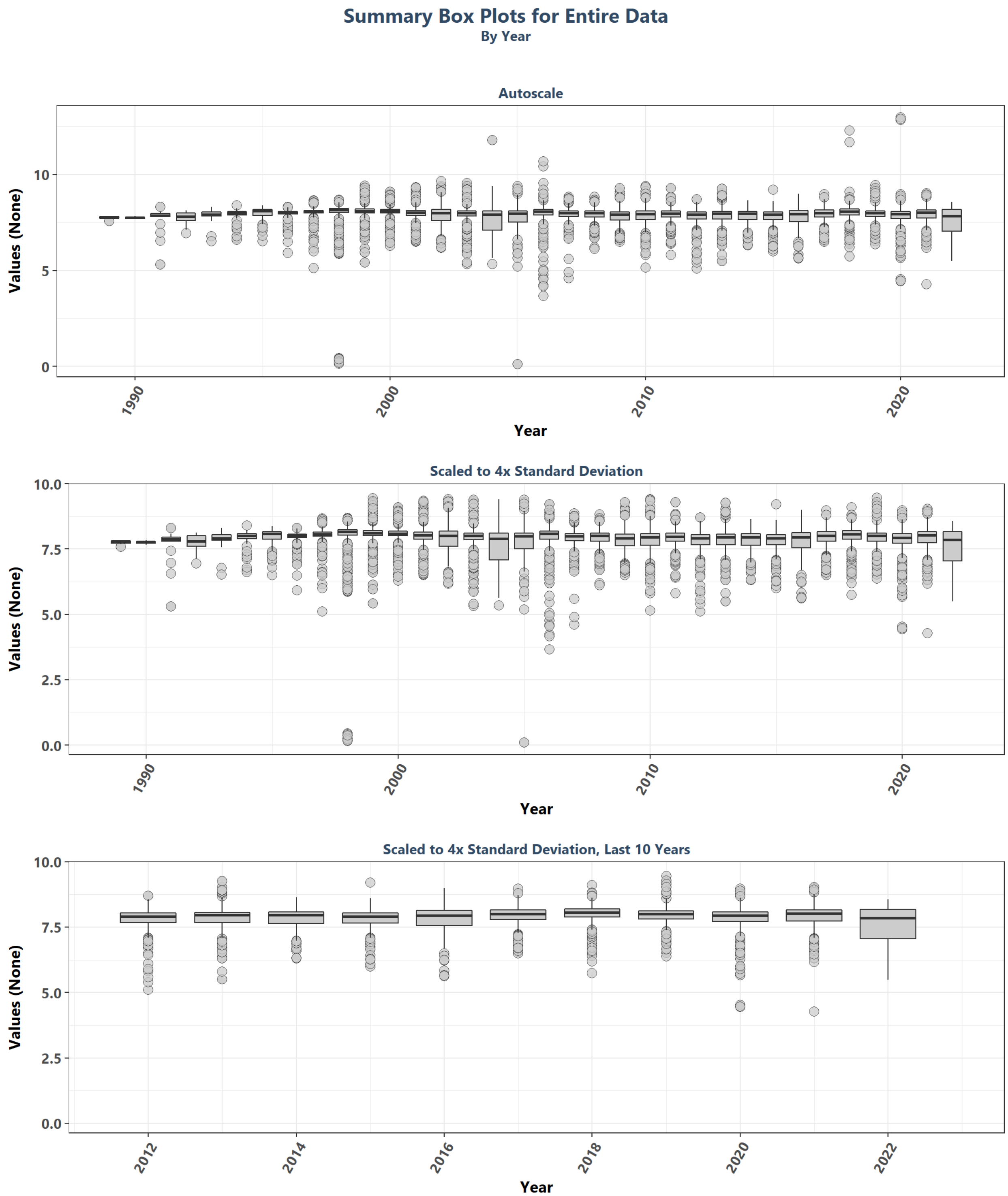
```
                       subtitle="By Year") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

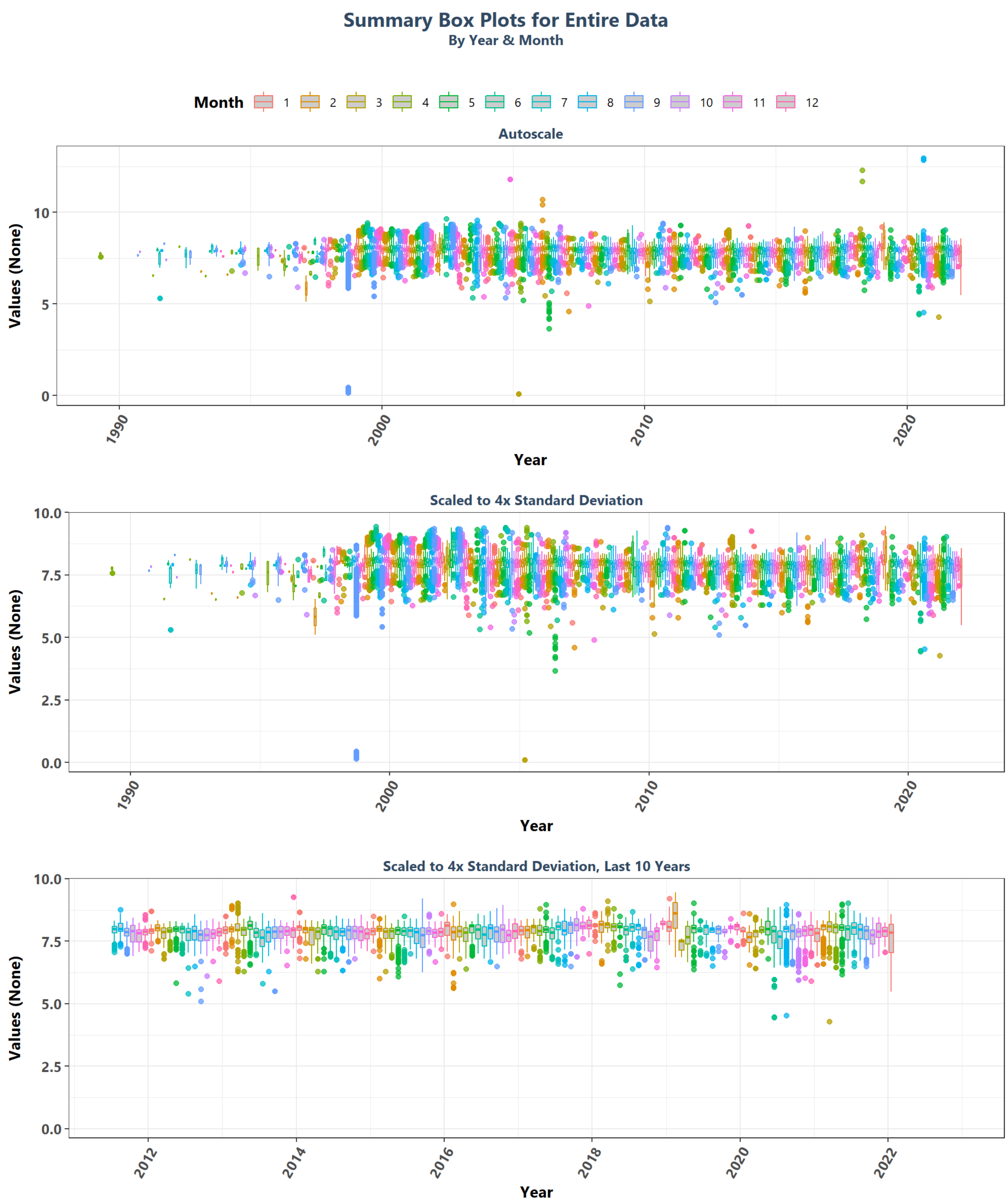
This set of box plots are grouped by year and month with the color being related to the month.

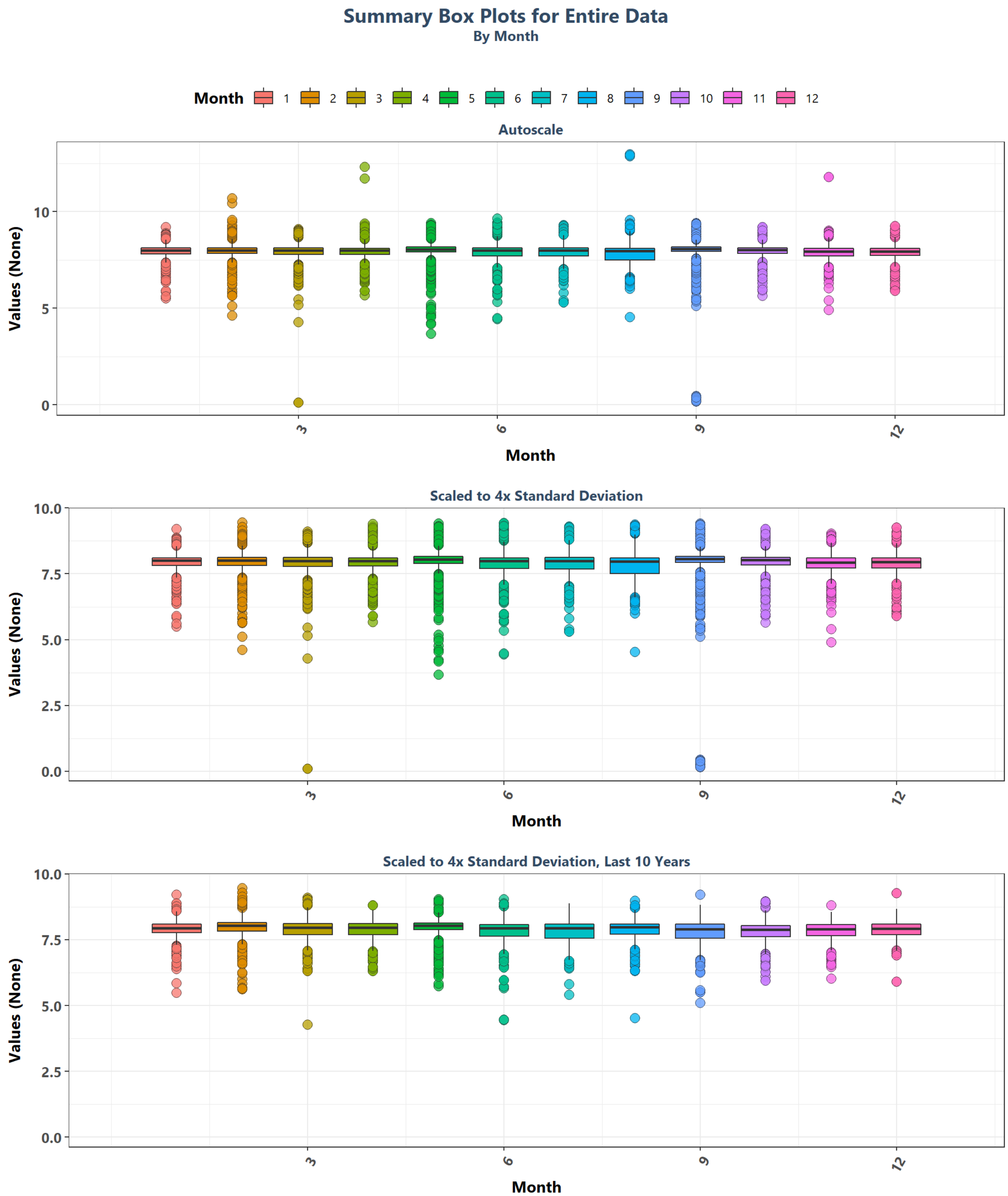
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit, ")"), 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.

```r
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 (None)
Year
Scaled to 4x Standard Deviation
Values (None)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year

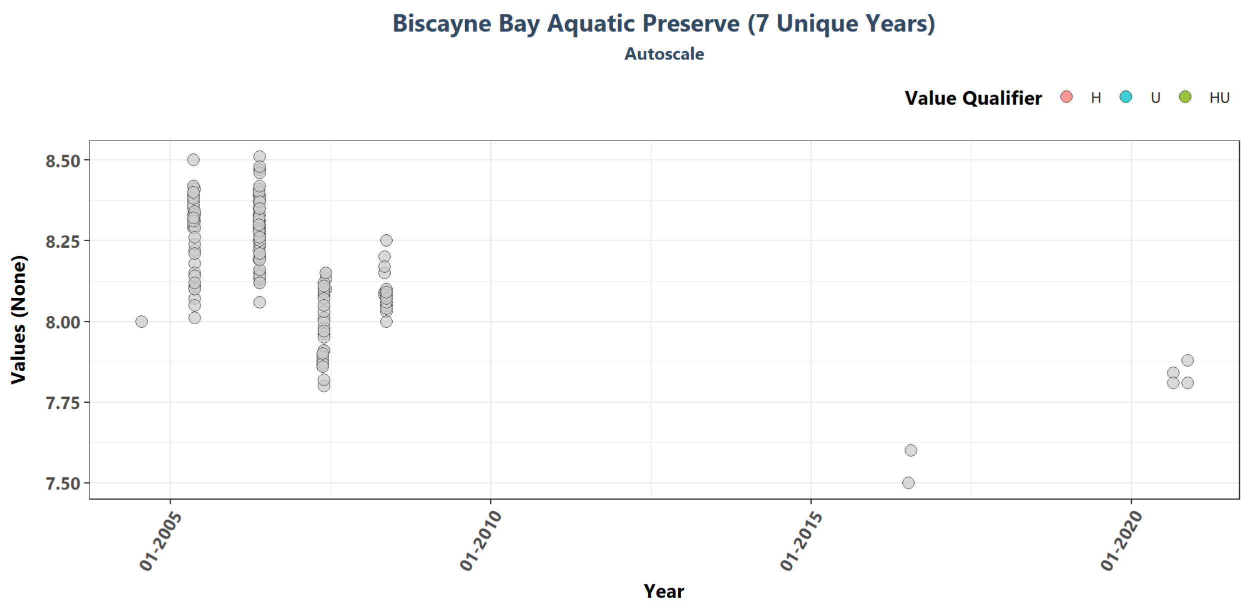

Summary Box Plots for Entire Data
By Year & Month
Month
1
2
3
4
5
6
7
8
9
10
11
12
Autoscale
Values (None)
10
5
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (None)
10.0
7.5
5.0
2.5
0.0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
10.0
7.5
5.0
2.5
0.0
2012
2014
2016
2018
2020
2022
Year

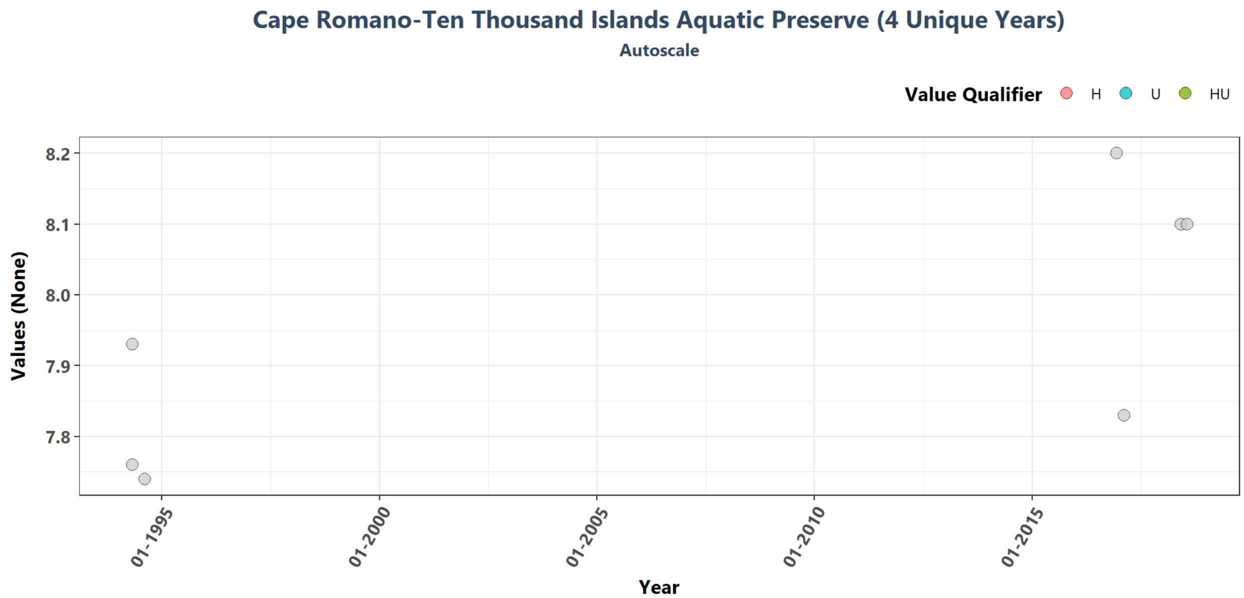


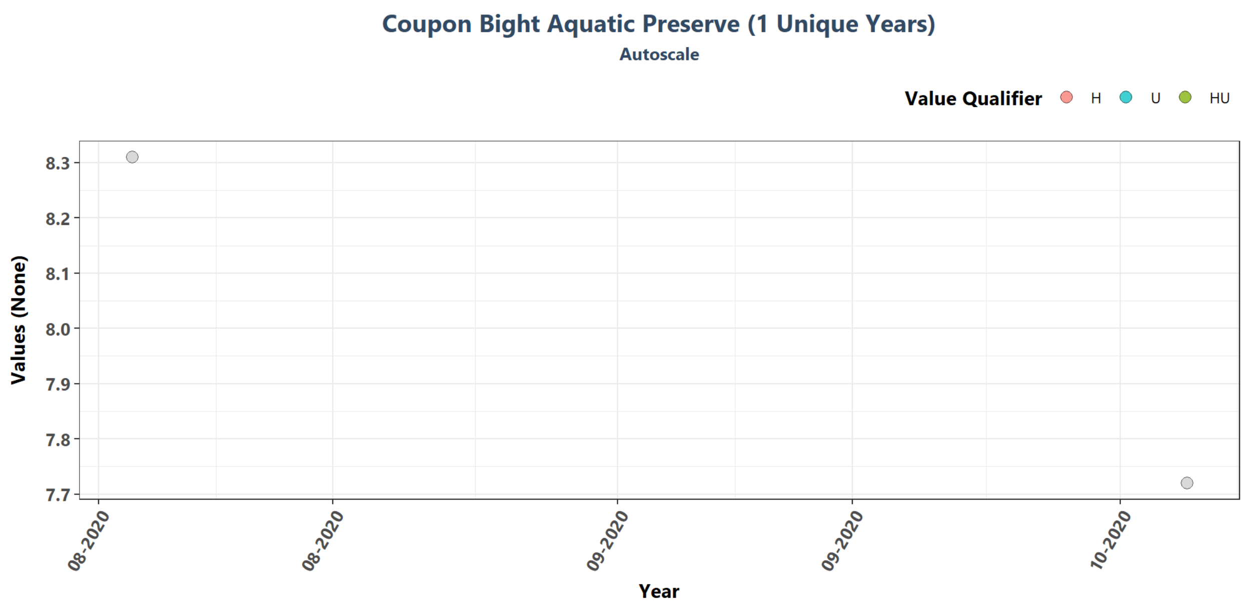
Appendix III: Excluded Managed Areas

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

```r
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)
   }
}
```

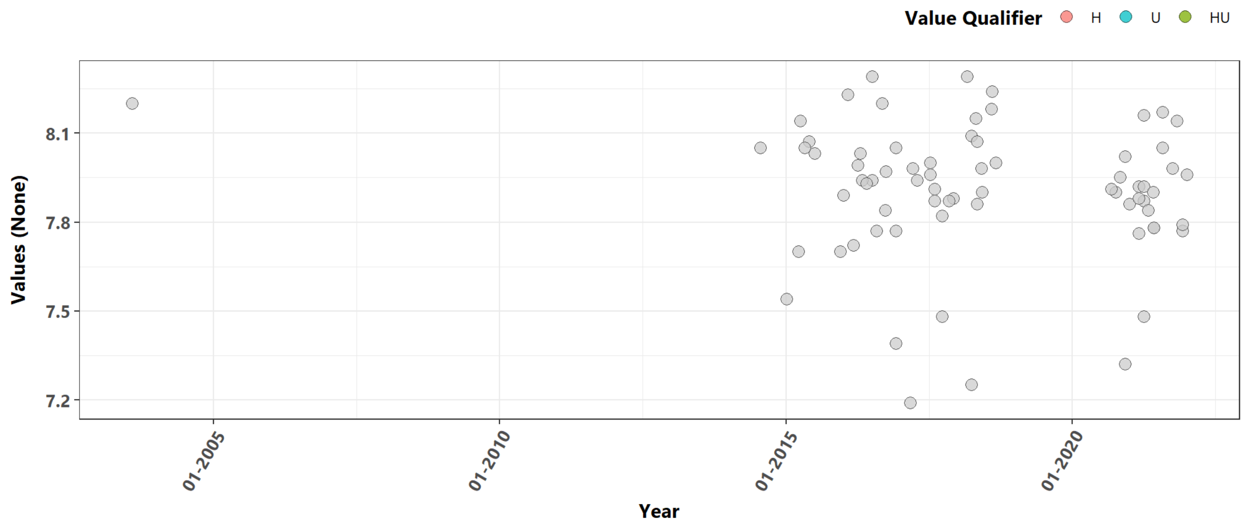



Cape Romano-Ten Thousand Islands Aquatic Preserve (4 Unique Years)
Autoscale
Value Qualifier
H
U
HU
8.2
8.1
8.0
7.9
7.8
Values (None)
01-1995
01-2000
01-2005
01-2010
01-2015
Year


Coupon Bight Aquatic Preserve (1 Unique Years)
Autoscale
Value Qualifier
H
U
HU
8.3
8.2
8.1
8.0
7.9
7.8
7.7
Values (None)
08-2020
08-2020
09-2020
09-2020
10-2020
Year

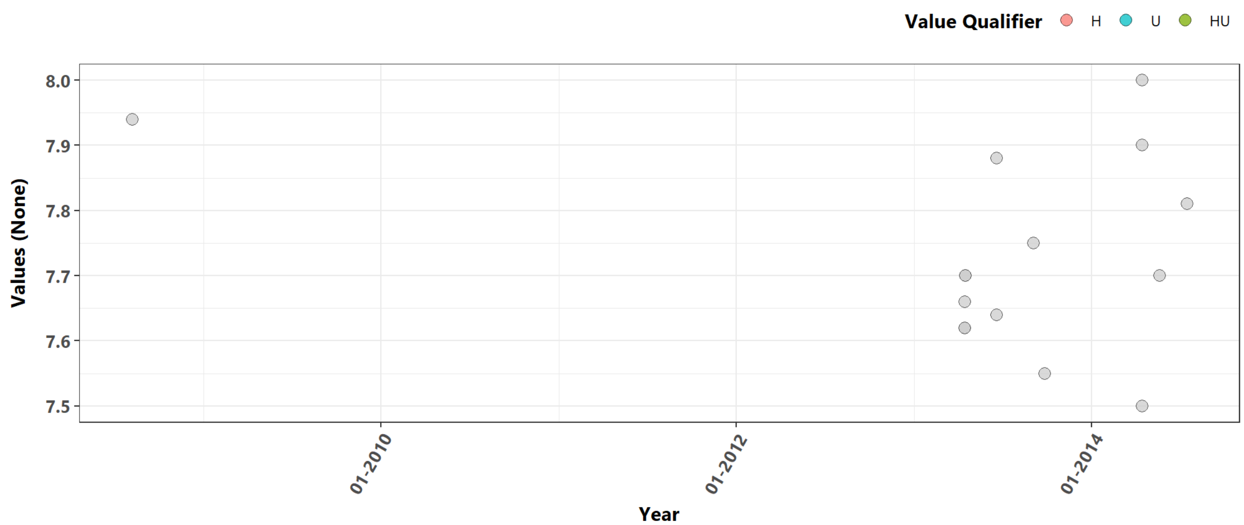
Estero Bay Aquatic Preserve (9 Unique Years)

Autoscale



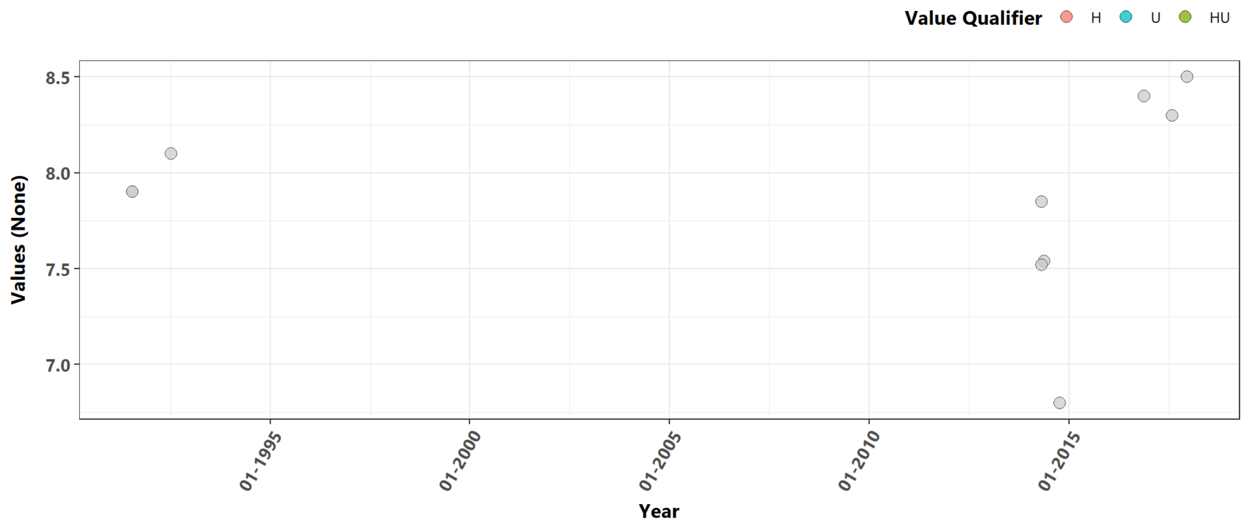
Fort Clinch State Park Aquatic Preserve (3 Unique Years)

Autoscale



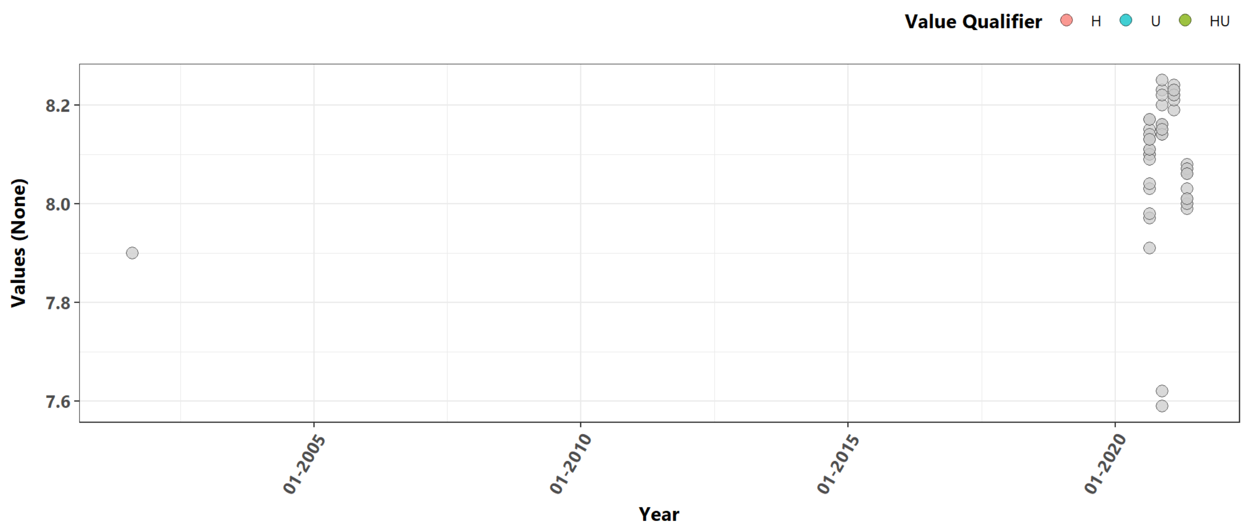
Fort Pickens State Park Aquatic Preserve (5 Unique Years)

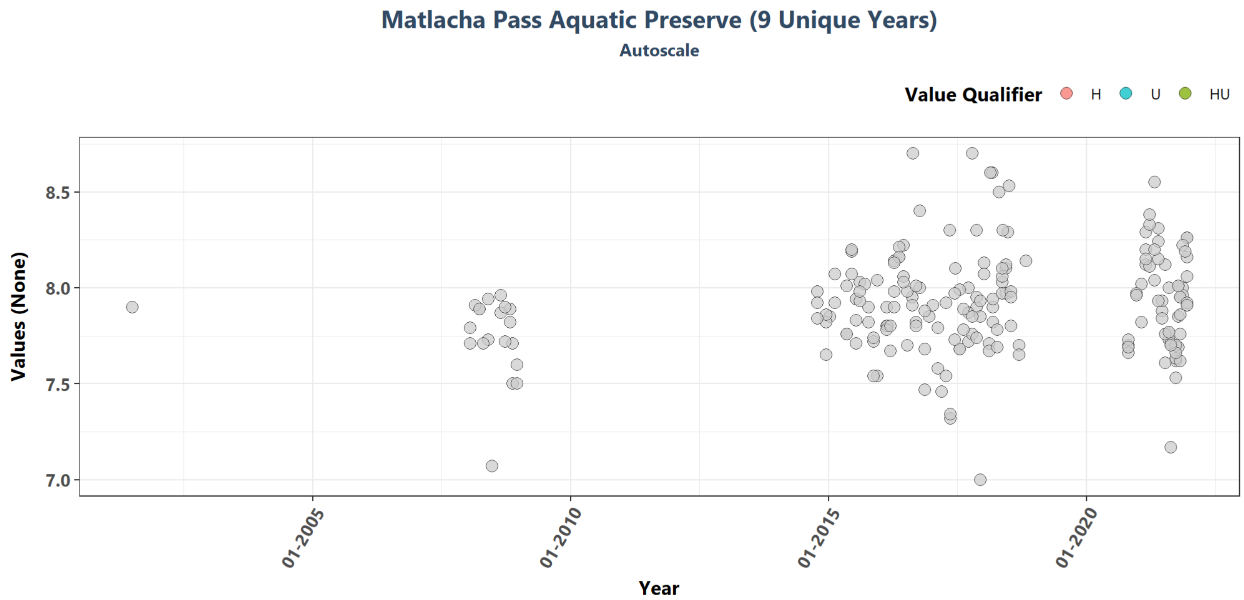
Autoscale

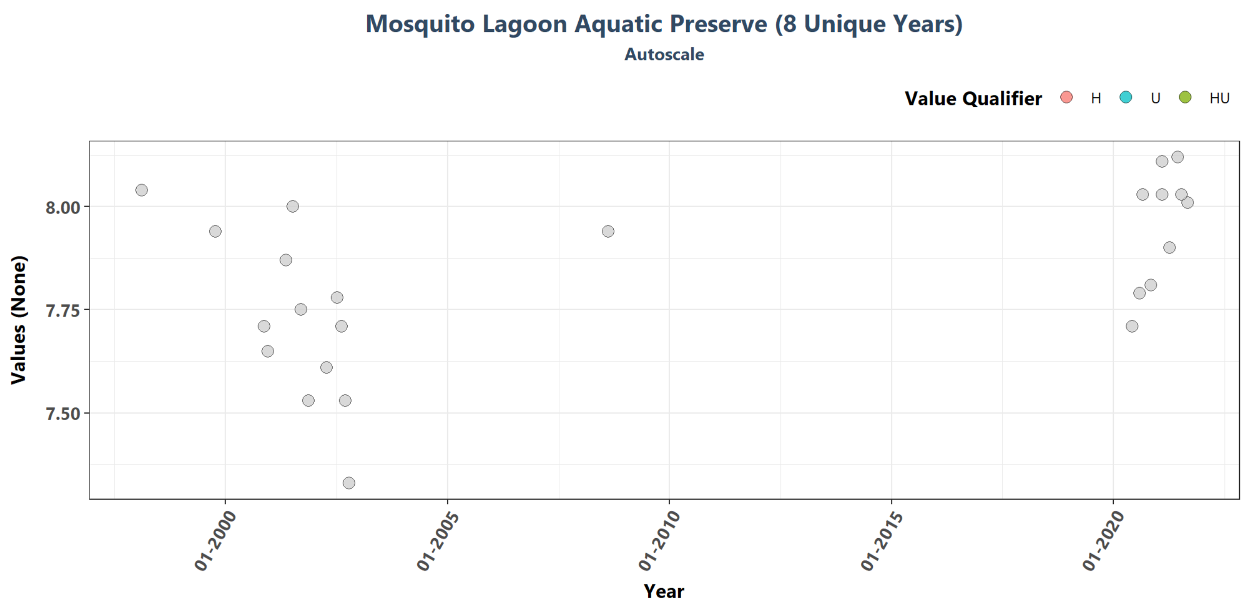


Lignumvitae Key Aquatic Preserve (3 Unique Years)

Autoscale

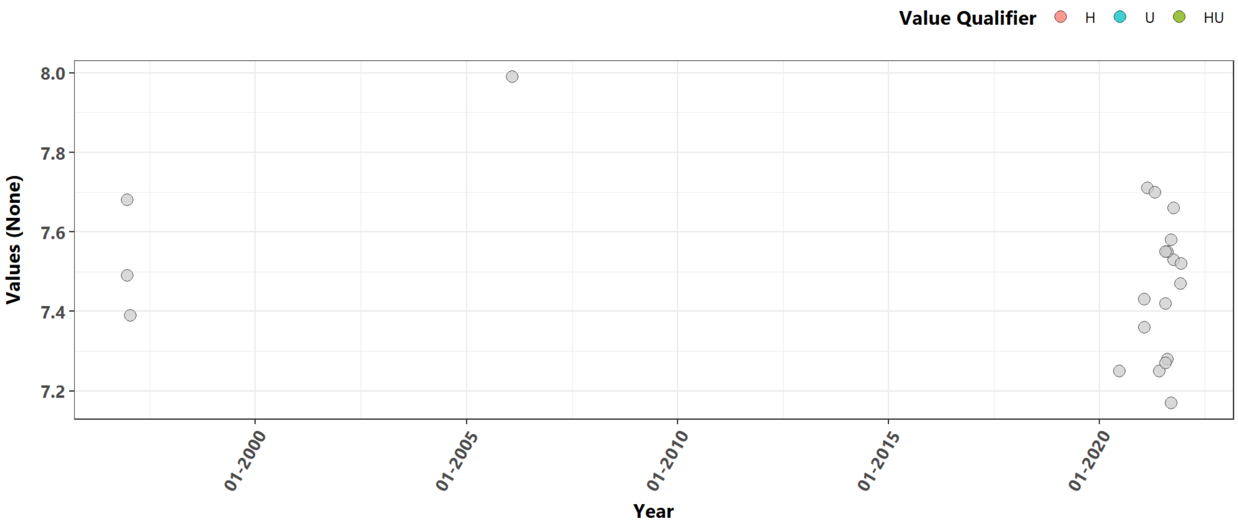



Matlacha Pass Aquatic Preserve (9 Unique Years)
Autoscale
Value Qualifier
H
U
HU
Values (None)
8.5
8.0
7.5
7.0
01-2005
01-2010
01-2015
01-2020
Year


Mosquito Lagoon Aquatic Preserve (8 Unique Years)
Autoscale
Value Qualifier
H
U
HU
Values (None)
8.00
7.75
7.50
01-2000
01-2005
01-2010
01-2015
01-2020
Year

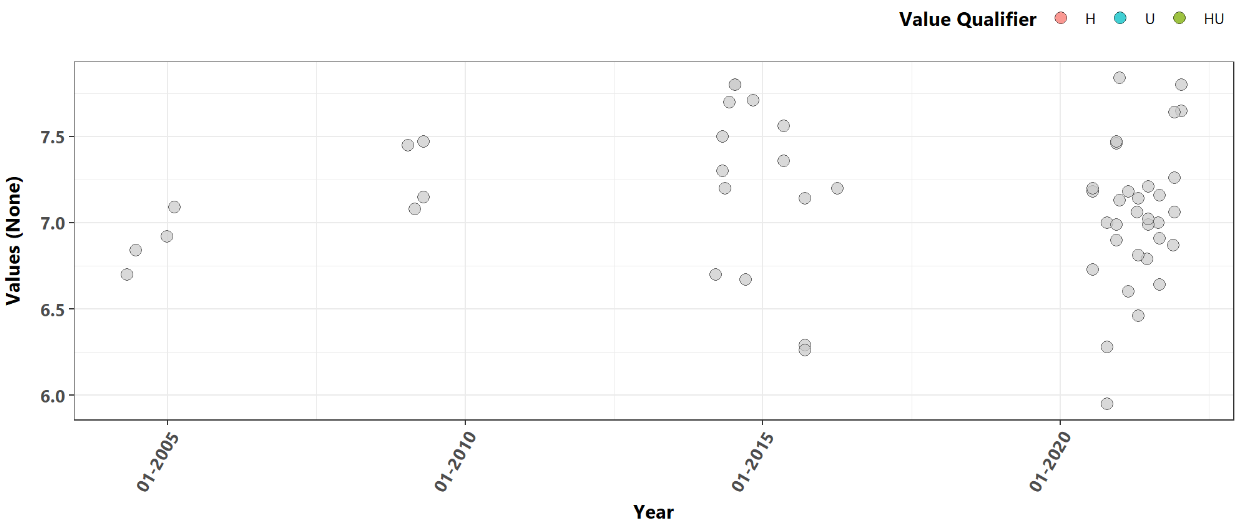
North Fork St. Lucie Aquatic Preserve (5 Unique Years)

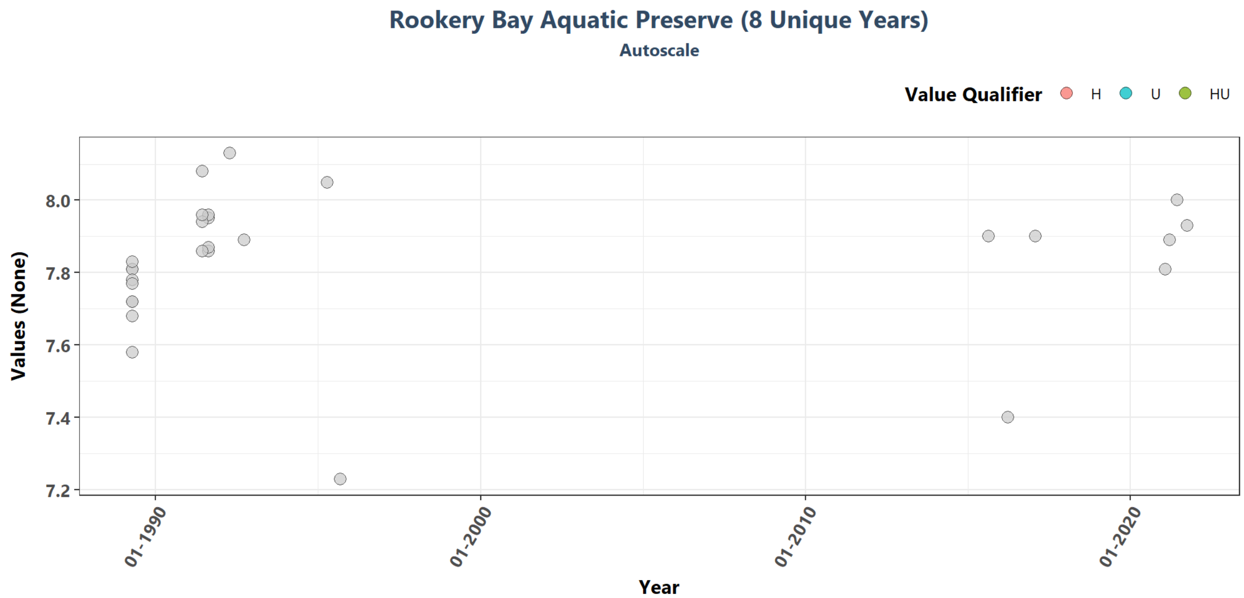
Autoscale

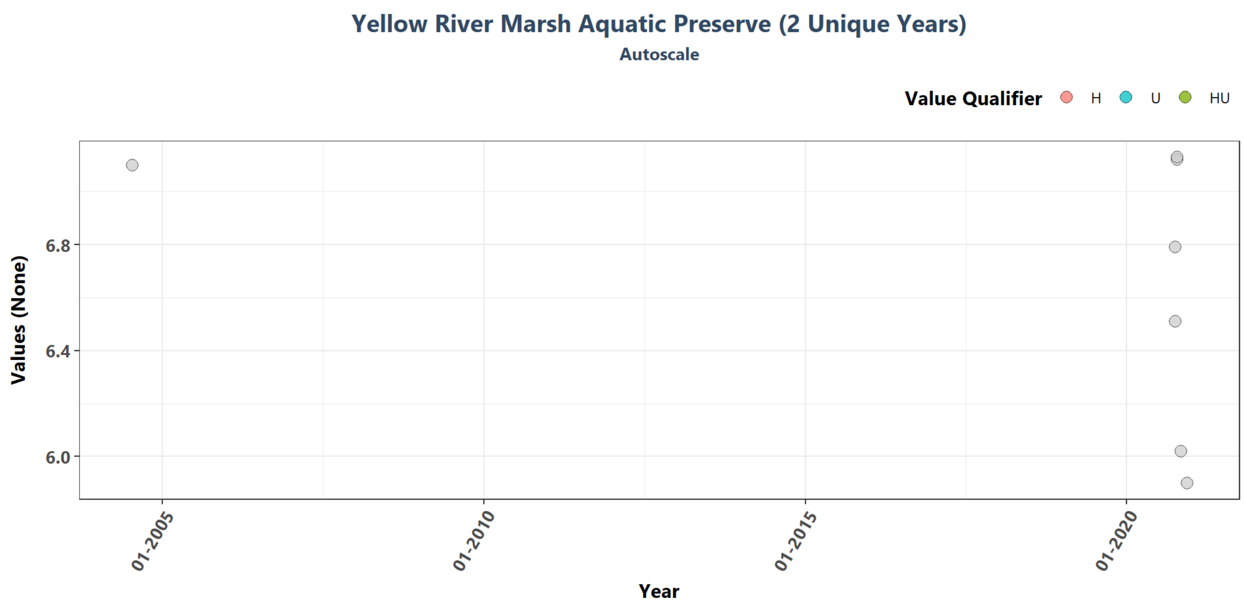


Pellicer Creek Aquatic Preserve (9 Unique Years)

Autoscale







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

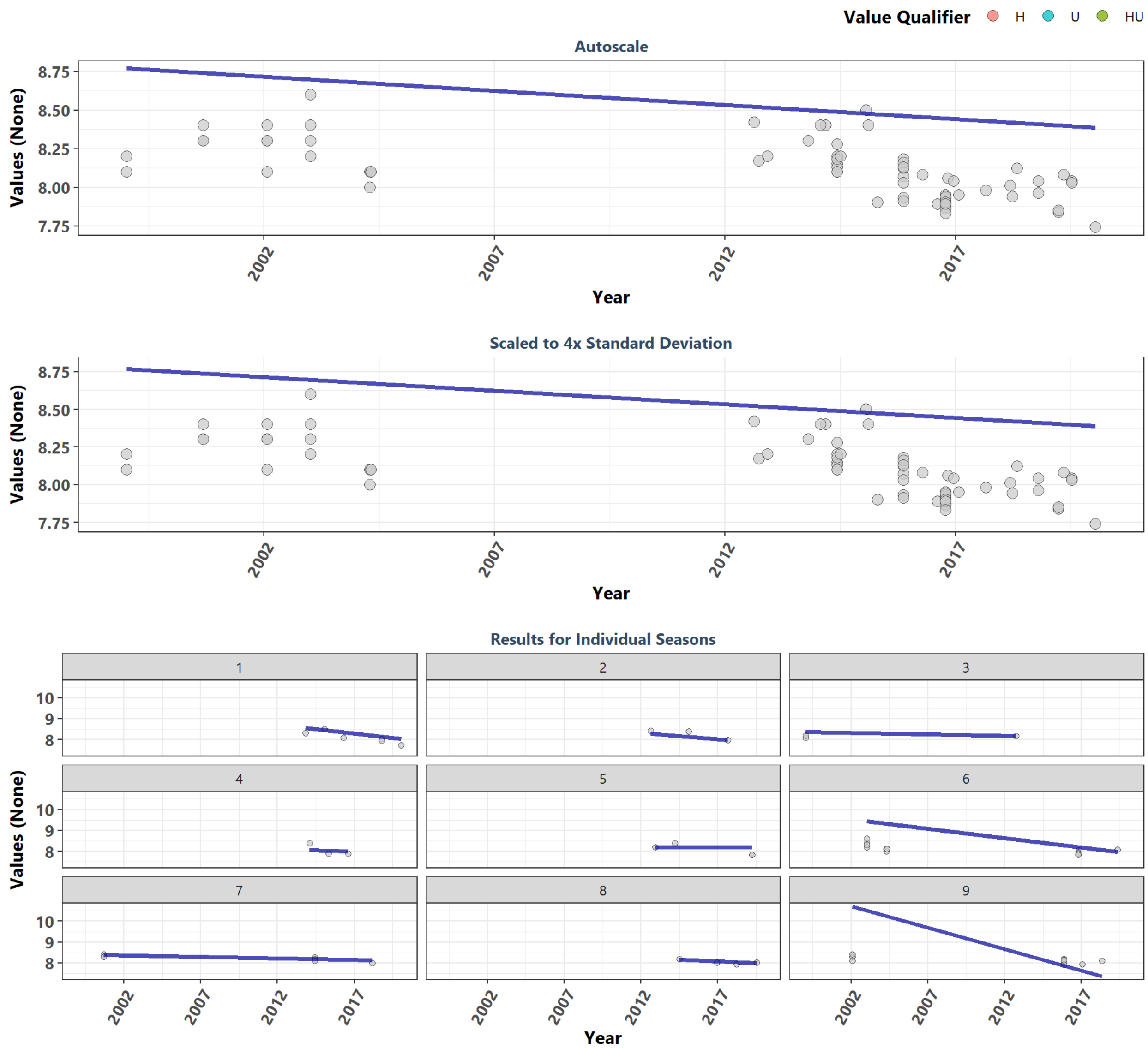
5. Set the plot type as a point plot with the size of the points
6. Add the linear trend
7. Create the title, x-axis, y-axis, and color fill labels
8. Set the y and x limits
9. Make the axis labels bold
10. Plot the arrangement as a set of panels

```
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$relyear_dd), 5, ceiling) - round_any(min(plot_data$rel

      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_dd

      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)) +
```

```r
    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 = relye
                                                                                        y = relyear_
                                                                                        xend = relyea
                                                                                        yend = relyea
                 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 = relye
                                                                                        y = relyear_
                                                                                        xend = relyea
                                                                                        yend = relyea
                 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 = rely
                                                                                             y = relyear_
                                                                                             xend = relye
                                                                                             yend = relye
                      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)
   }
}
```

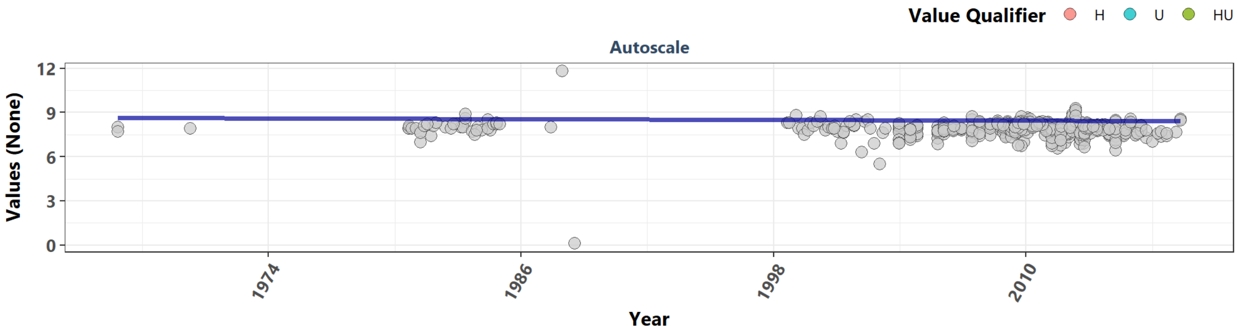
Alligator Harbor Aquatic Preserve

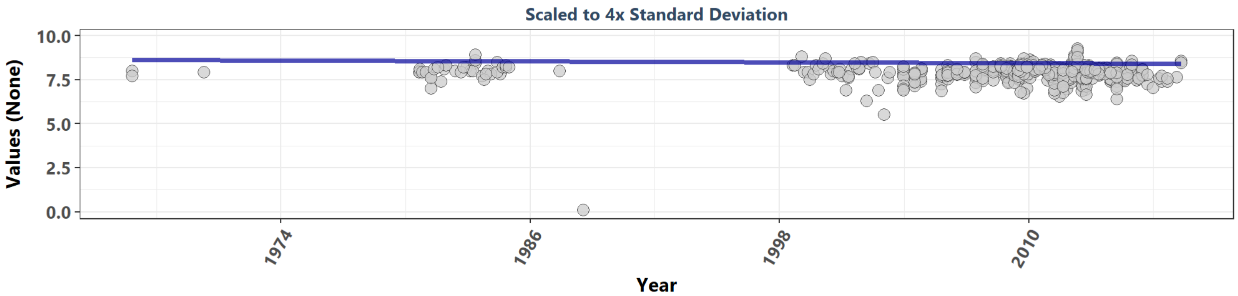


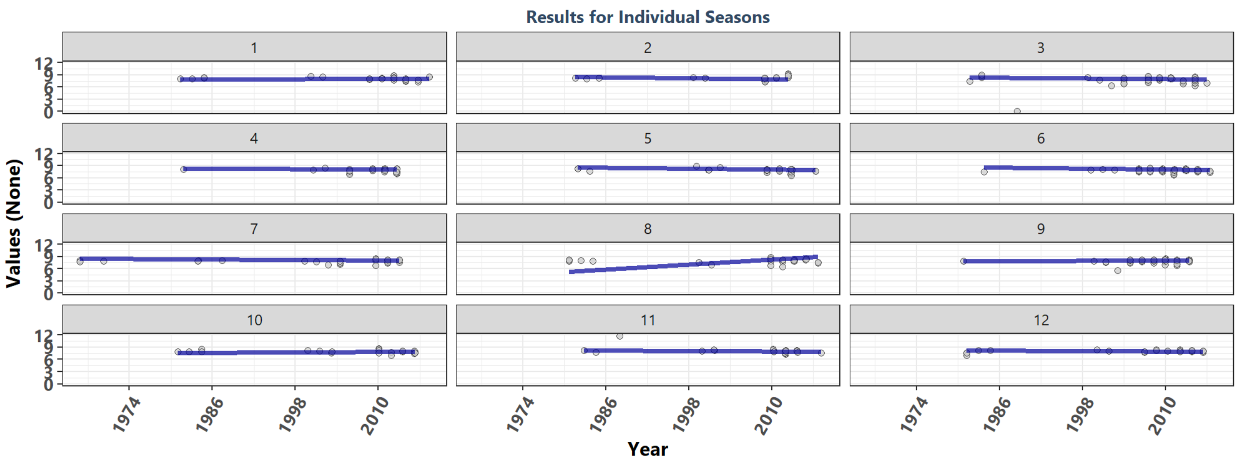
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	72	8.10	-0.5343	-0.0227	8.9200	-5.6	0.0000	10.9	0.2092	-1
1	6	8.06	-1.0000	-0.1100	10.6000	-2.1	0.0354	NA	NA	-1
2	3	8.40	-0.5000	-0.0720	9.5370	-1.0	0.2963	NA	NA	-1
3	3	NA	-0.3516	-0.0173	8.4655	NA	NA	NA	NA	NA
4	3	7.90	-0.5455	-0.0230	8.4830	-1.0	0.2963	NA	NA	-1
5	4	8.02	0.0000	0.0018	8.1573	-0.7	0.4701	NA	NA	1
6	23	8.00	-0.8000	-0.1127	10.5387	-4.0	0.0001	NA	NA	-1
7	11	8.18	-0.5636	-0.0182	8.5255	-2.8	0.0053	NA	NA	-1
8	5	8.04	-0.4000	-0.0400	8.9200	-0.8	0.4350	NA	NA	-1
9	14	8.12	-1.0000	-0.2550	13.0000	-1.9	0.0538	NA	NA	-1

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

Apalachicola Bay Aquatic Preserve



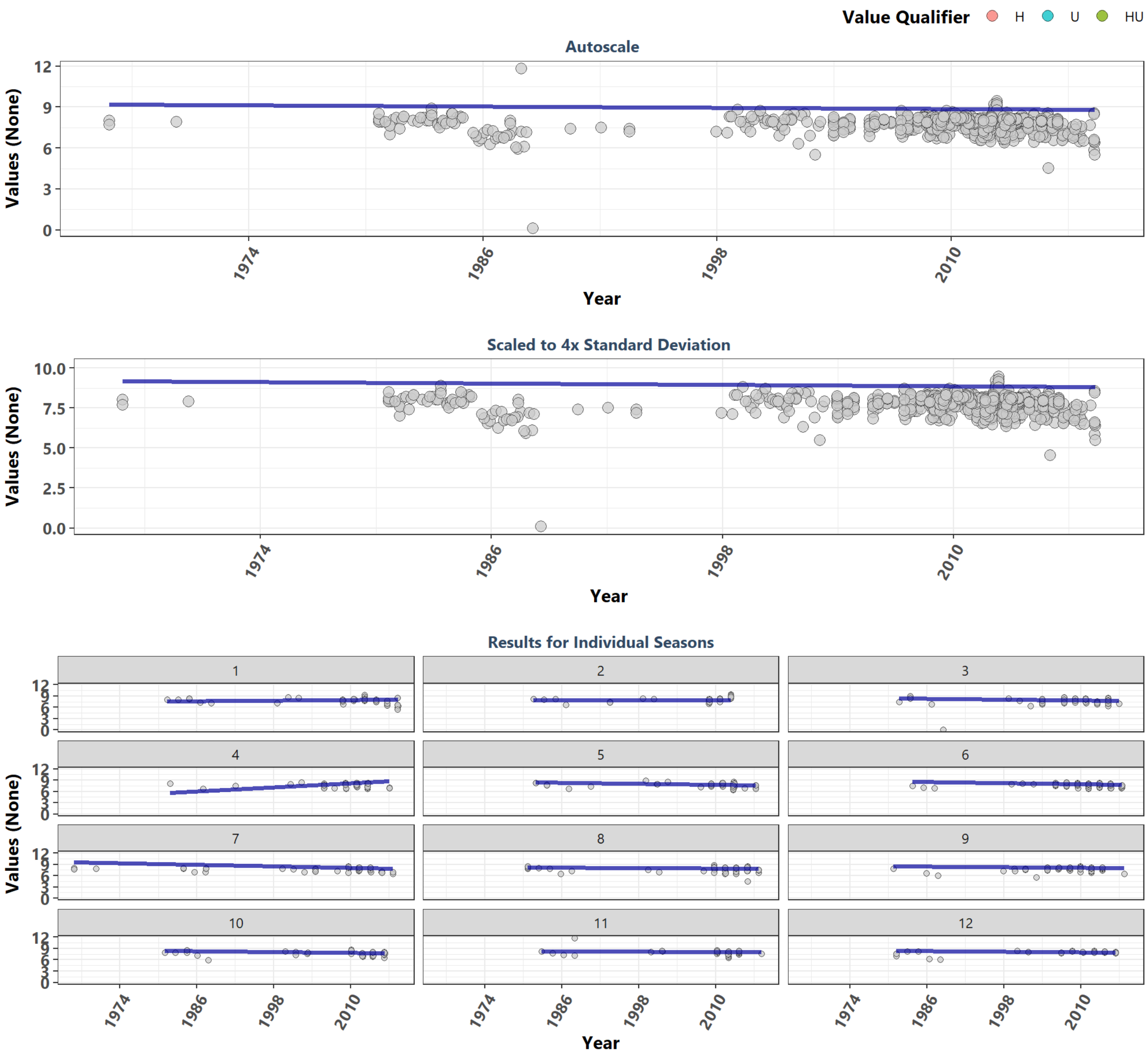




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	599	8.01	-0.0501	-0.0067	8.8025	-1.7	0.0889	45.3	0	0
1	41	8.08	0.0725	0.0100	7.4650	-1.7	0.0809	NA	NA	0
2	33	8.20	-0.3220	-0.0328	9.6721	3.9	0.0001	NA	NA	0
3	58	7.90	-0.1174	-0.0260	9.2830	-1.3	0.1910	NA	NA	0
4	49	8.00	-0.1173	-0.0100	8.6600	-0.6	0.5487	NA	NA	0
5	44	7.98	-0.2483	-0.0300	9.6800	-2.3	0.0222	NA	NA	0
6	85	7.96	-0.2294	-0.0378	10.0200	-1.0	0.3219	NA	NA	0
7	41	7.93	-0.1878	-0.0167	8.9800	2.0	0.0406	NA	NA	0
8	37	8.00	0.4564	0.1700	-0.8100	0.5	0.5951	NA	NA	0
9	80	8.00	0.0616	0.0049	7.7371	1.0	0.3358	NA	NA	0
10	33	7.90	0.2195	0.0150	7.1350	-2.7	0.0067	NA	NA	0
11	49	8.06	-0.0578	-0.0175	8.9450	-2.6	0.0086	NA	NA	0
12	49	8.12	-0.0723	-0.0125	8.6350	-1.2	0.2315	NA	NA	0

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

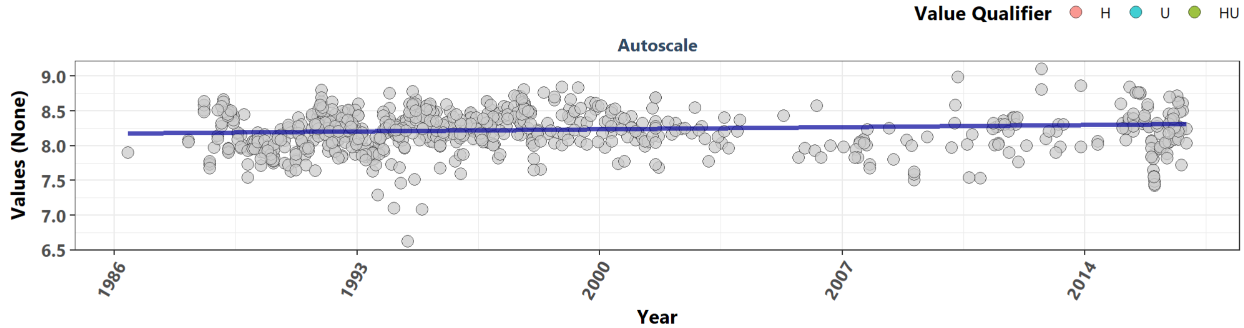
Apalachicola National Estuarine Research Reserve

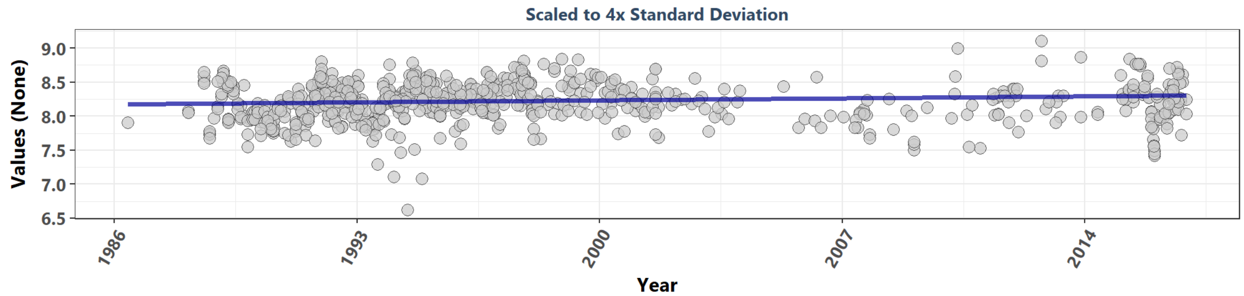


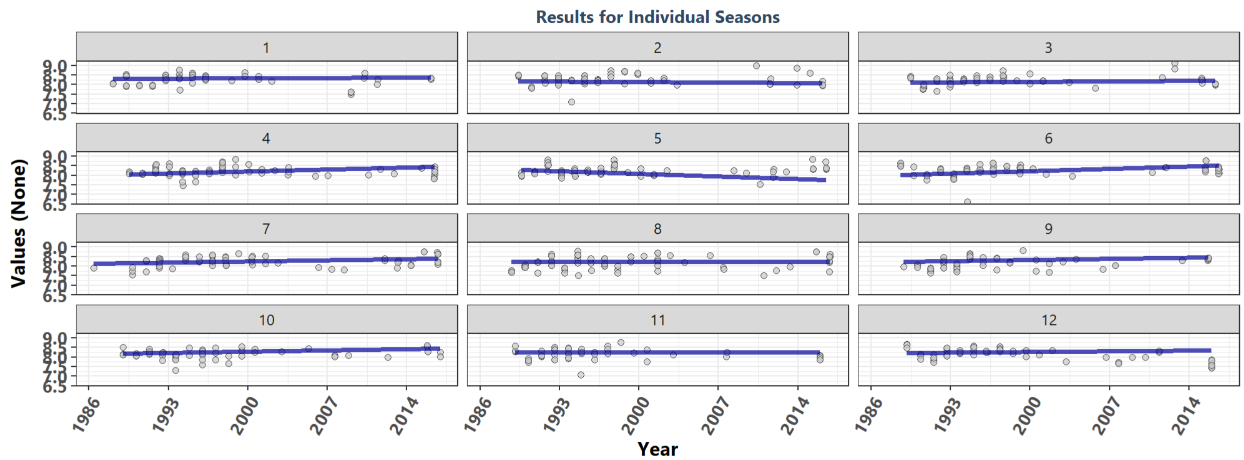
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	812	7.95	-0.0626	-0.0125	9.5313	-2.8	0.0058	49.3	0	-1
1	66	8.00	0.1158	0.0200	6.9200	-3.2	0.0012	NA	NA	1
2	45	8.12	-0.0235	-0.0025	8.0225	4.8	0.0000	NA	NA	-1
3	67	7.80	-0.1321	-0.0335	9.5777	-1.6	0.1116	NA	NA	-1
4	66	7.78	0.4687	0.1500	0.1700	-1.0	0.3181	NA	NA	1
5	72	7.92	-0.0821	-0.0324	9.5225	-1.8	0.0694	NA	NA	-1
6	109	7.92	-0.1397	-0.0364	9.9018	-1.9	0.0550	NA	NA	-1
7	61	7.89	-0.2699	-0.0571	11.0907	-0.3	0.7915	NA	NA	-1
8	59	7.90	-0.0836	-0.0147	8.6941	-0.9	0.3456	NA	NA	-1
9	95	7.98	-0.1475	-0.0280	9.5470	1.7	0.0926	NA	NA	-1
10	49	7.86	-0.1956	-0.0312	9.5475	-2.0	0.0439	NA	NA	-1
11	68	8.04	-0.0263	-0.0025	8.2425	-1.9	0.0633	NA	NA	-1
12	55	8.11	-0.1228	-0.0300	9.5400	-0.3	0.7803	NA	NA	-1

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

Banana River Aquatic Preserve



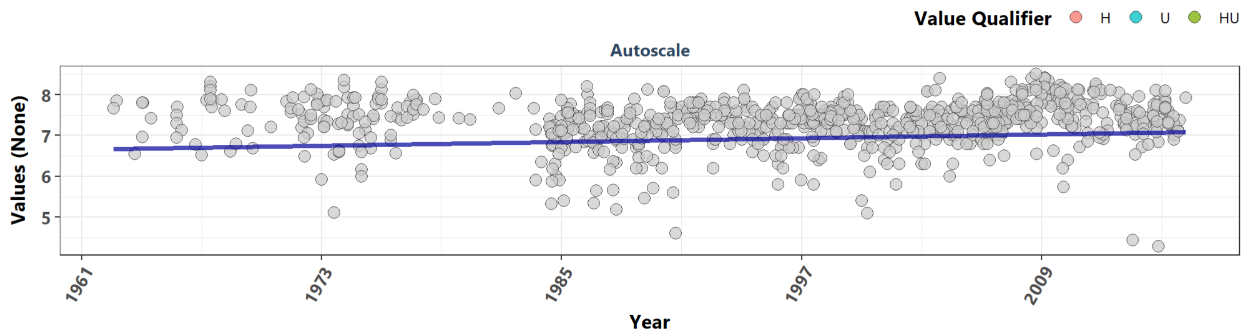


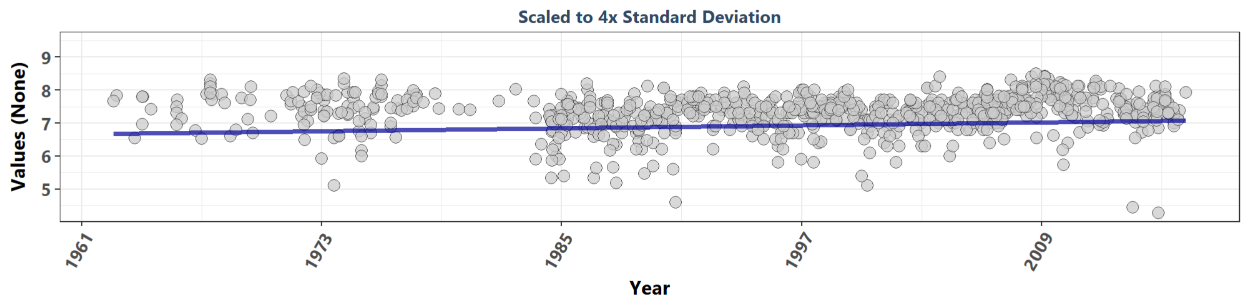


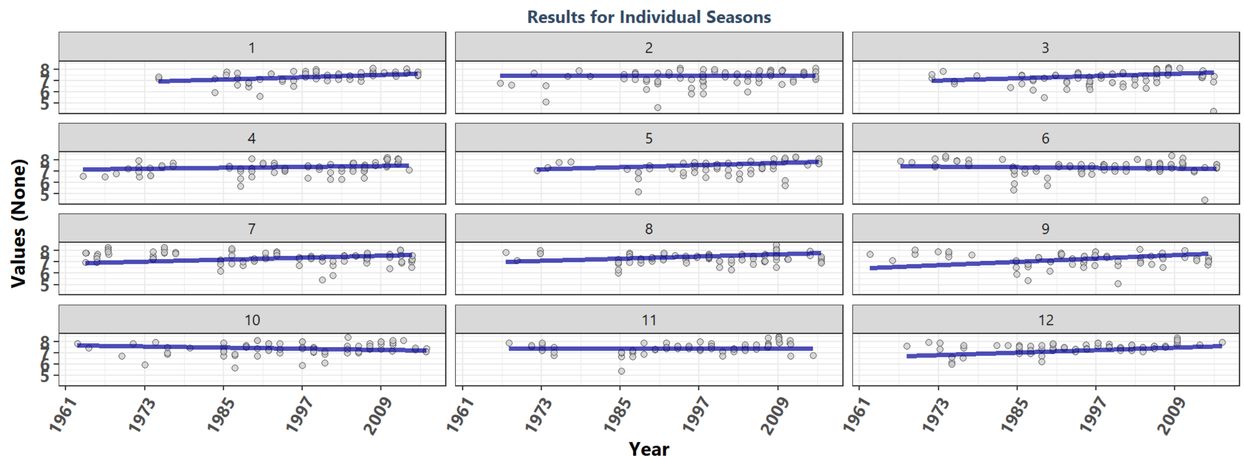
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	726	8.22	0.0727	0.0050	8.1509	2.9	0.0034	29.8	0.0017	1
1	58	8.27	0.0870	0.0031	8.2762	1.4	0.1650	NA	NA	1
2	59	8.23	-0.0609	-0.0044	8.2174	0.1	0.9058	NA	NA	-1
3	57	8.22	0.0734	0.0038	8.0903	1.6	0.1132	NA	NA	1
4	81	8.23	0.2028	0.0178	7.9011	0.0	0.9673	NA	NA	1
5	61	8.25	-0.3078	-0.0238	8.4900	1.5	0.1430	NA	NA	-1
6	65	8.32	0.2544	0.0200	7.8800	1.0	0.3058	NA	NA	1
7	66	8.25	0.1441	0.0100	8.0900	2.0	0.0413	NA	NA	1
8	68	8.15	0.0111	0.0000	8.2300	2.4	0.0144	NA	NA	-1
9	54	8.12	0.1252	0.0100	8.1550	2.7	0.0065	NA	NA	1
10	55	8.14	0.1716	0.0110	8.0900	0.8	0.4308	NA	NA	1
11	46	8.16	0.0034	0.0000	8.2300	-0.6	0.5547	NA	NA	-1
12	56	8.18	0.1284	0.0069	8.1469	-3.4	0.0008	NA	NA	1

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

Big Bend Seagrasses Aquatic Preserve



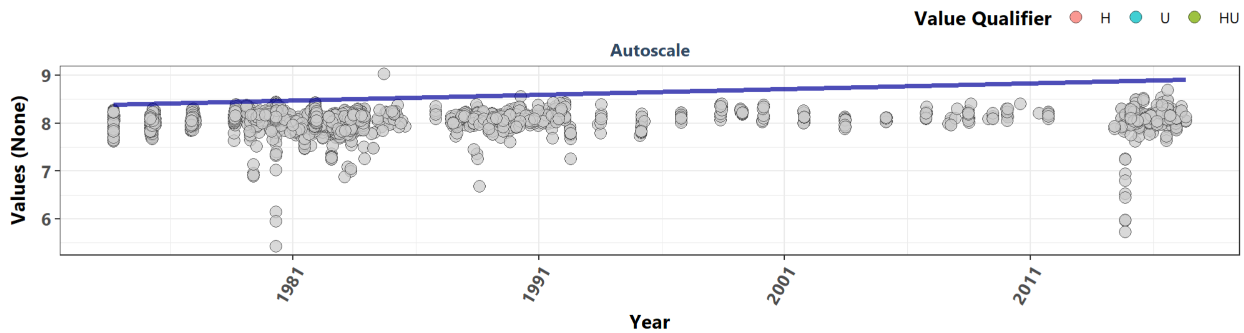


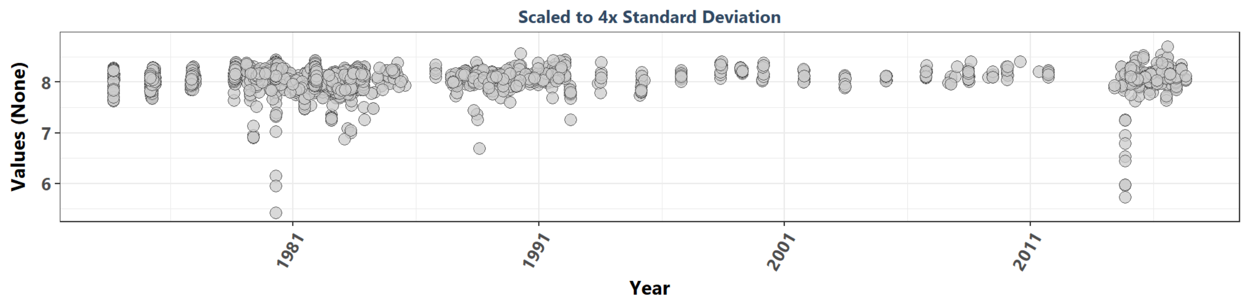


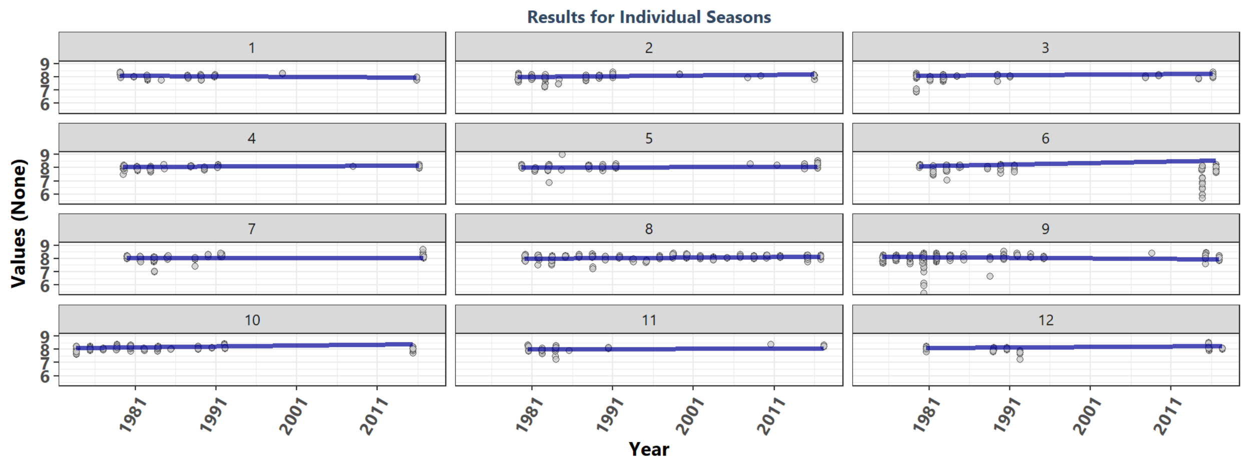
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	885	7.40	0.1355	0.0130	6.3386	5.5	0.0000	51.1	0	1
1	62	7.44	0.3069	0.0300	5.9100	3.4	0.0007	NA	NA	1
2	80	7.40	-0.0044	0.0000	7.4000	2.8	0.0059	NA	NA	-1
3	73	7.30	0.2946	0.0288	6.0600	2.9	0.0034	NA	NA	1
4	74	7.32	0.1052	0.0110	6.8940	3.9	0.0001	NA	NA	1
5	64	7.50	0.2566	0.0250	6.4000	3.8	0.0001	NA	NA	1
6	91	7.40	-0.0764	-0.0081	7.6893	-0.1	0.9534	NA	NA	-1
7	83	7.40	0.2095	0.0239	6.2772	-2.5	0.0137	NA	NA	1
8	80	7.36	0.2795	0.0277	6.2107	-0.2	0.8511	NA	NA	1
9	64	7.32	0.3269	0.0427	5.3636	-0.9	0.3739	NA	NA	1
10	70	7.40	-0.1840	-0.0148	8.0519	1.3	0.1985	NA	NA	-1
11	70	7.50	-0.0146	0.0000	7.3650	3.4	0.0006	NA	NA	-1
12	74	7.50	0.2336	0.0308	5.8200	3.2	0.0012	NA	NA	1

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

Boca Ciega Bay Aquatic Preserve



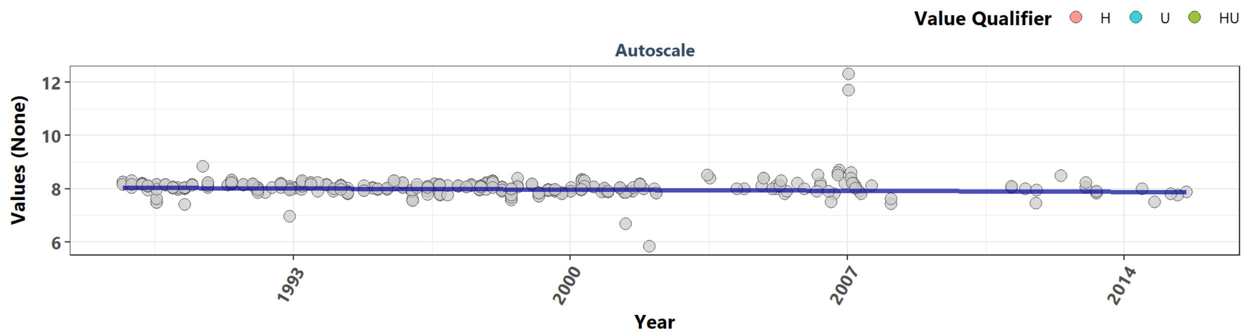


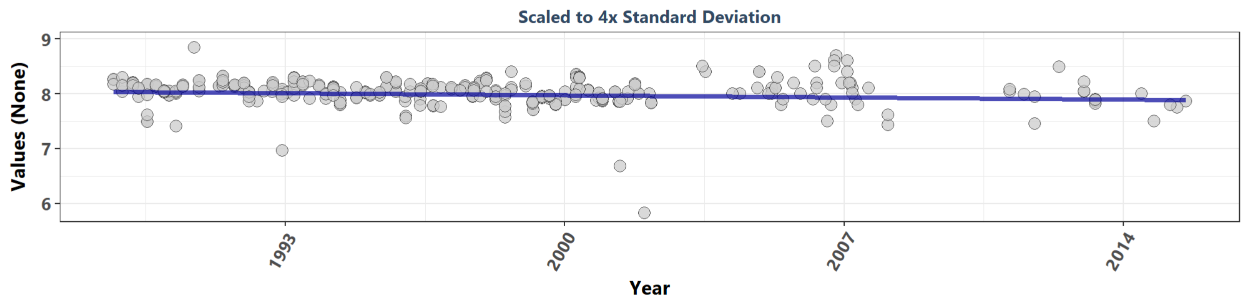


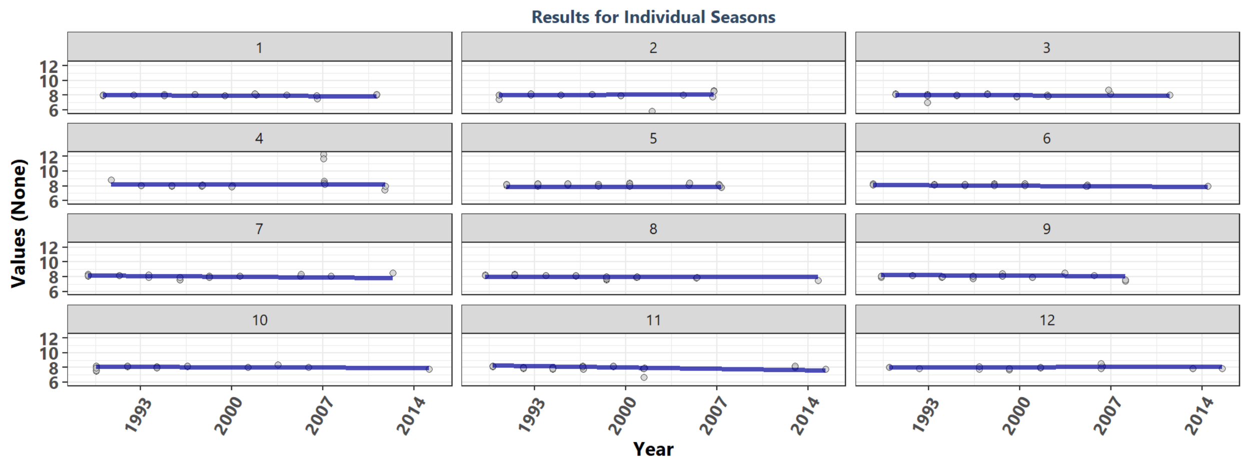
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4062	8.09	0.1721	0.0200	7.9490	17.0	0.0000	279	0	1
1	89	8.07	-0.1996	-0.0059	8.2473	-2.4	0.0163	NA	NA	-1
2	127	8.01	0.1950	0.0080	7.7940	3.3	0.0010	NA	NA	1
3	96	8.00	0.1689	0.0053	7.9710	3.5	0.0005	NA	NA	1
4	97	8.04	0.1510	0.0050	7.9350	0.7	0.4775	NA	NA	1
5	122	8.12	0.0483	0.0020	7.9860	3.7	0.0002	NA	NA	1
6	136	8.07	0.2193	0.0200	7.6000	-3.5	0.0005	NA	NA	1
7	100	8.11	0.0021	0.0000	8.0500	2.5	0.0113	NA	NA	-1
8	466	8.07	0.2388	0.0076	7.7935	5.3	0.0000	NA	NA	1
9	2216	8.10	-0.1713	-0.0075	8.2950	16.0	0.0000	NA	NA	-1
10	454	8.09	0.1903	0.0120	7.8260	6.2	0.0000	NA	NA	1
11	82	8.02	0.0229	0.0024	7.9630	0.3	0.7551	NA	NA	1
12	77	8.05	0.2244	0.0073	7.9018	0.0	0.9820	NA	NA	1

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

Cape Haze Aquatic Preserve



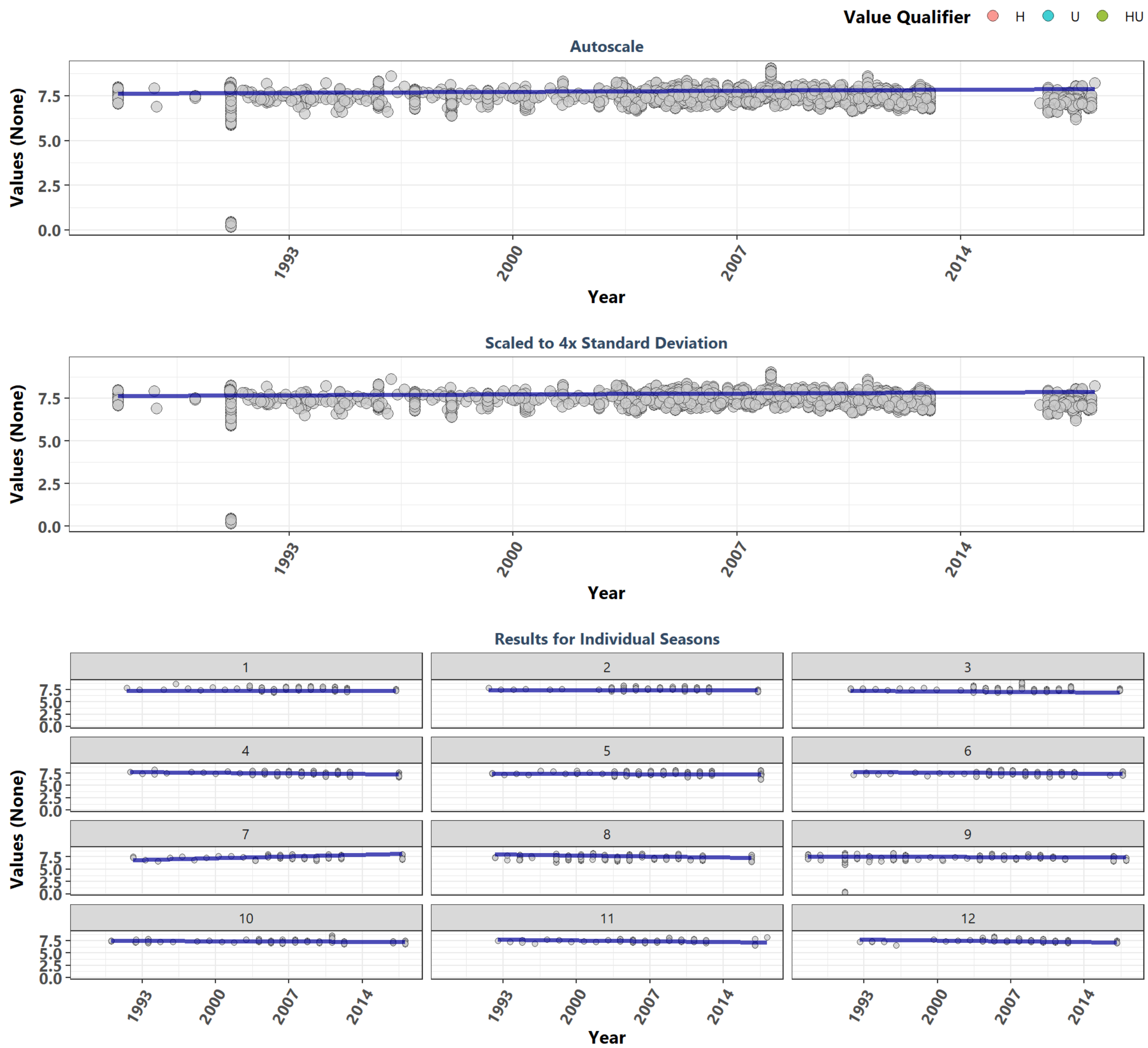




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	336	8.05	-0.1144	-0.0133	8.3175	-3.1	0.0020	19.7	0.0503	-1
1	28	8.00	-0.1077	-0.0194	8.4761	0.1	0.8882	NA	NA	-1
2	29	8.02	0.0533	0.0110	7.7550	-1.3	0.1941	NA	NA	1
3	26	7.99	-0.1149	-0.0130	8.2920	-0.8	0.4459	NA	NA	-1
4	25	8.03	-0.0185	-0.0007	8.2079	0.4	0.7198	NA	NA	-1
5	35	8.19	-0.0462	-0.0033	7.9800	-0.1	0.8851	NA	NA	-1
6	26	8.18	-0.1700	-0.0200	8.5200	-1.0	0.3339	NA	NA	-1
7	24	8.07	-0.2222	-0.0300	8.7700	0.4	0.6870	NA	NA	-1
8	30	8.14	0.0212	0.0017	7.9567	-4.5	0.0000	NA	NA	1
9	27	8.05	-0.1354	-0.0140	8.5350	-1.6	0.1037	NA	NA	-1
10	30	8.09	-0.0667	-0.0110	8.3430	-0.5	0.6090	NA	NA	-1
11	30	7.98	-0.5770	-0.0600	9.5500	-0.9	0.3745	NA	NA	-1
12	26	7.89	0.0616	0.0093	7.8375	-0.3	0.7532	NA	NA	1

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

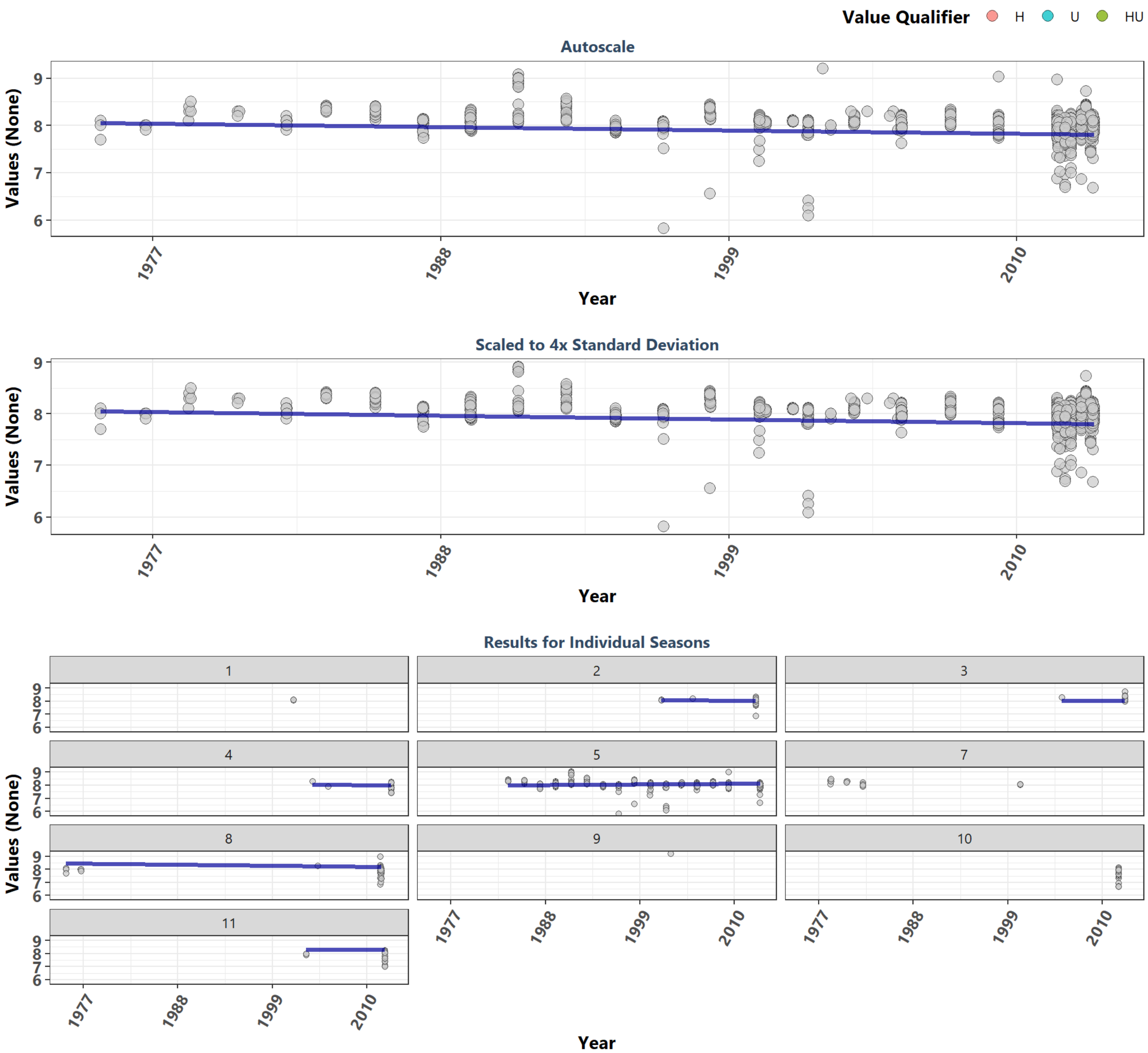
Cockroach Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3391	7.28	-0.0288	0.0100	7.5595	4.5	0.0000	237.8	0	1
1	151	7.53	-0.0049	0.0000	7.2600	-2.0	0.0470	NA	NA	-1
2	148	7.57	0.0027	0.0000	7.3700	-5.3	0.0000	NA	NA	-1
3	159	7.43	-0.1390	-0.0158	7.4003	-1.2	0.2473	NA	NA	-1
4	160	7.37	-0.2939	-0.0233	7.9900	-4.8	0.0000	NA	NA	-1
5	168	7.37	-0.0801	-0.0100	7.5450	0.1	0.9593	NA	NA	-1
6	162	7.30	-0.1085	-0.0133	7.8633	-1.5	0.1287	NA	NA	-1
7	149	7.26	0.2249	0.0600	6.2300	-0.1	0.9303	NA	NA	1
8	1027	7.13	-0.2919	-0.0333	8.4033	11.5	0.0000	NA	NA	-1
9	798	7.20	-0.0616	-0.0060	7.5740	-6.0	0.0000	NA	NA	-1
10	173	7.30	-0.1582	-0.0100	7.5400	-3.1	0.0019	NA	NA	-1
11	153	7.38	-0.2541	-0.0300	8.0900	-3.0	0.0025	NA	NA	-1
12	143	7.43	-0.1641	-0.0275	8.0400	-5.2	0.0000	NA	NA	-1

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

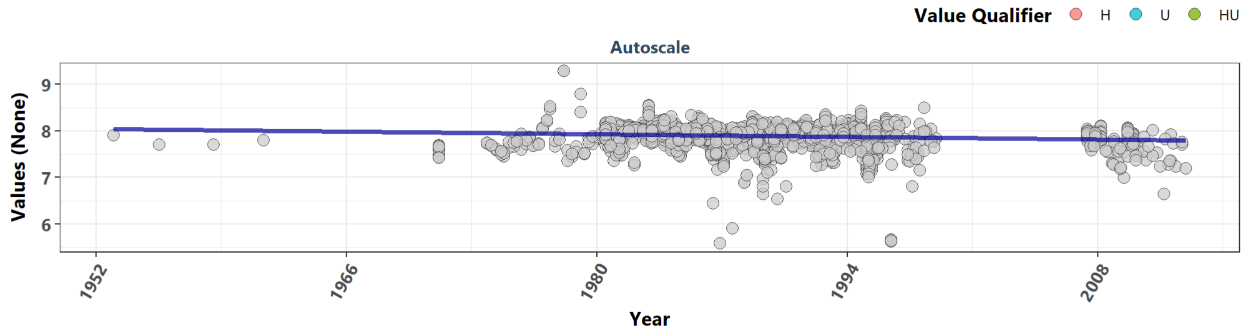
Florida Keys National Marine Sanctuary

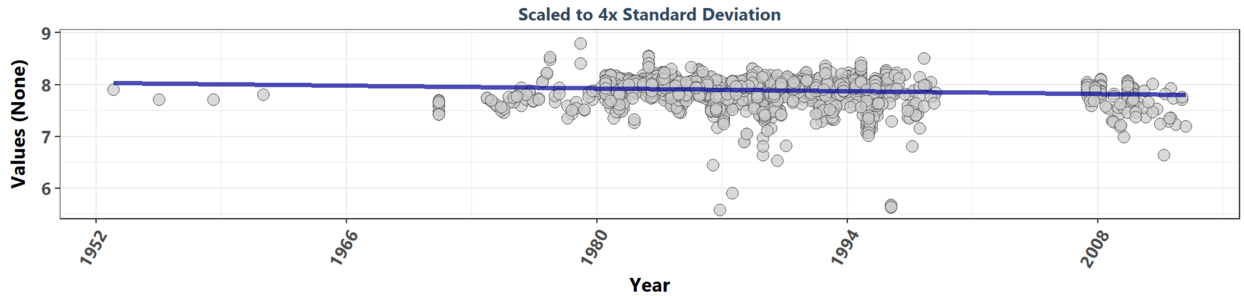


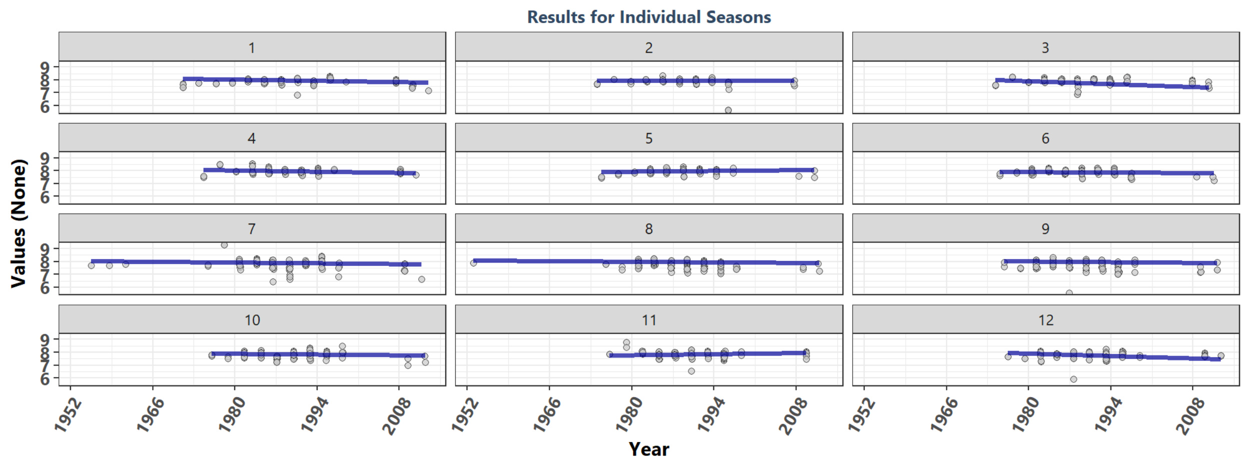
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1825	8.07	-0.1204	-0.0120	8.4100	-10.2	0.0000	114.6	0	-1
1	18	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	175	8.13	-0.2376	-0.0125	8.6400	3.5	0.0005	NA	NA	-1
3	63	NA	0.0031	0.0010	7.9750	NA	NA	NA	NA	NA
4	121	8.17	-0.2036	-0.0126	8.6255	-0.3	0.7755	NA	NA	-1
5	841	8.09	0.1054	0.0073	7.7560	-10.3	0.0000	NA	NA	1
7	62	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	276	8.02	-0.0040	-0.0140	8.8840	0.3	0.7789	NA	NA	-1
9	1	NA	-0.0161	-0.0100	8.4100	NA	NA	NA	NA	NA
10	191	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	77	NA	0.0067	0.0025	8.1825	NA	NA	NA	NA	NA

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

Gasparilla Sound-Charlotte Harbor Aquatic Preserve



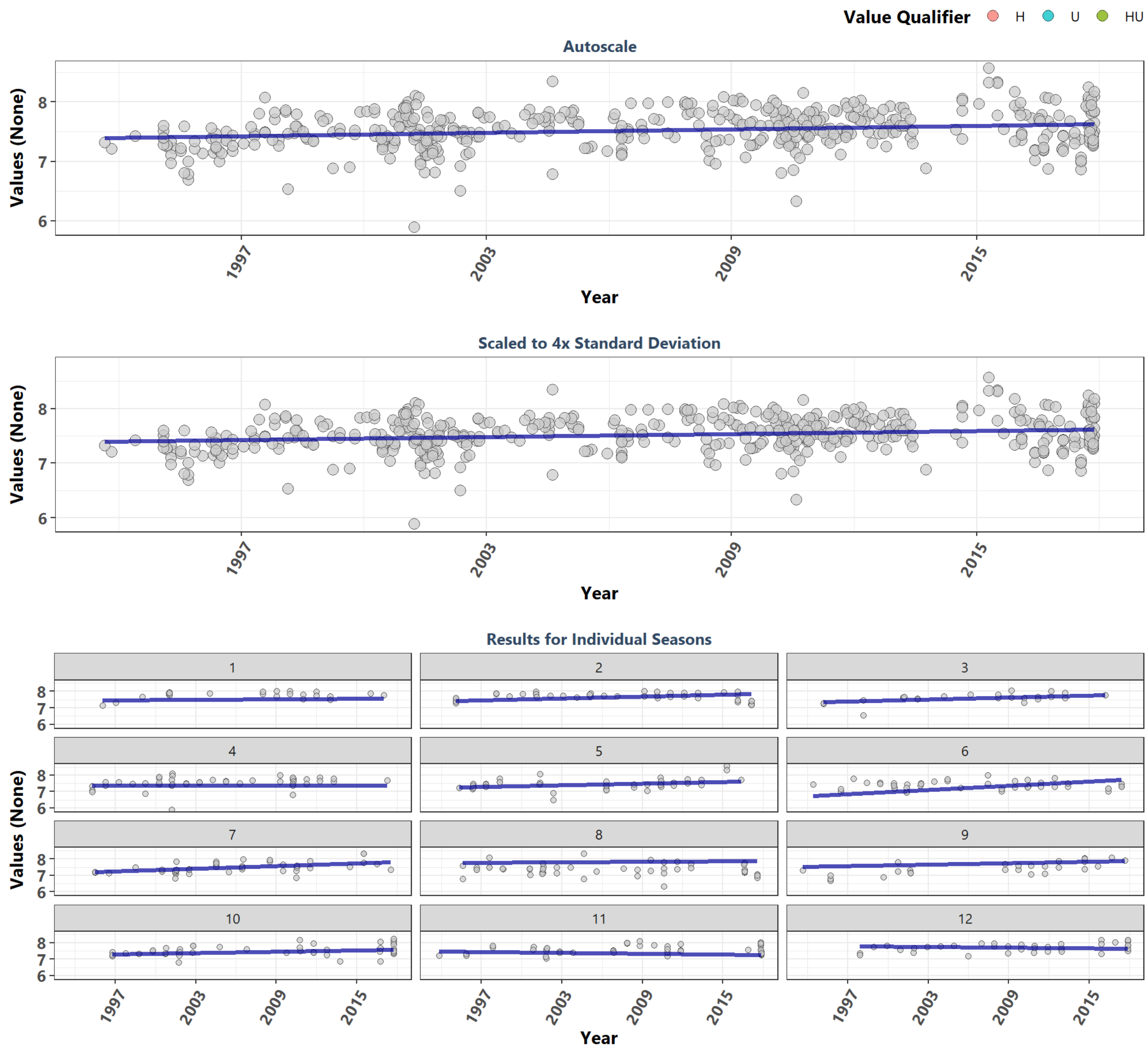




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2048	7.91	-0.1057	-0.0111	8.5400	-7.3	0.0000	46.7	0	-1
1	191	7.94	-0.2253	-0.0200	9.1300	-0.4	0.6671	NA	NA	-1
2	143	7.97	-0.0208	-0.0020	8.0560	-4.1	0.0001	NA	NA	-1
3	168	7.94	-0.2347	-0.0467	10.4867	-3.7	0.0002	NA	NA	-1
4	158	7.97	-0.1292	-0.0167	8.9367	-1.6	0.1145	NA	NA	-1
5	130	7.97	0.0671	0.0100	7.3950	1.2	0.2496	NA	NA	1
6	170	7.97	-0.0358	-0.0075	8.2975	-2.5	0.0113	NA	NA	-1
7	165	7.87	-0.1656	-0.0141	8.7053	-0.7	0.4901	NA	NA	-1
8	175	7.78	-0.1917	-0.0100	8.5300	-4.7	0.0000	NA	NA	-1
9	179	7.78	-0.0834	-0.0100	8.5500	-3.3	0.0011	NA	NA	-1
10	170	7.84	-0.1348	-0.0113	8.5225	1.6	0.1030	NA	NA	-1
11	194	7.88	0.0831	0.0150	6.9650	-3.5	0.0005	NA	NA	1
12	205	7.87	-0.1621	-0.0367	9.9067	-2.9	0.0037	NA	NA	-1

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

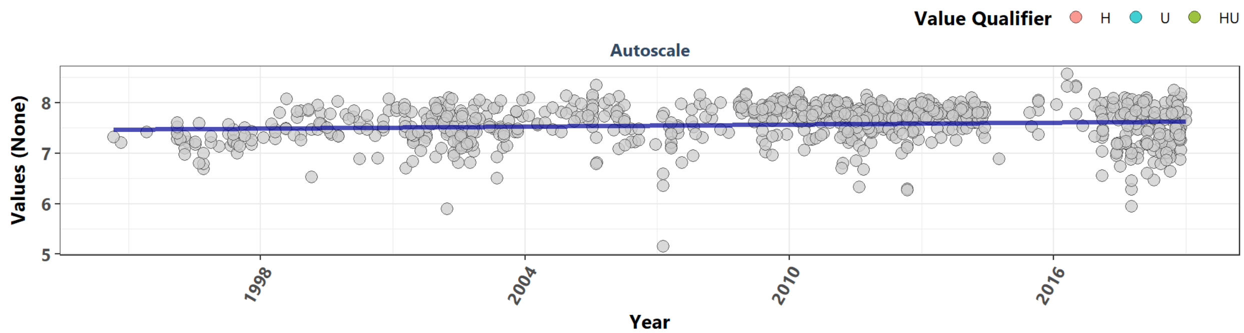
Guana River Marsh Aquatic Preserve

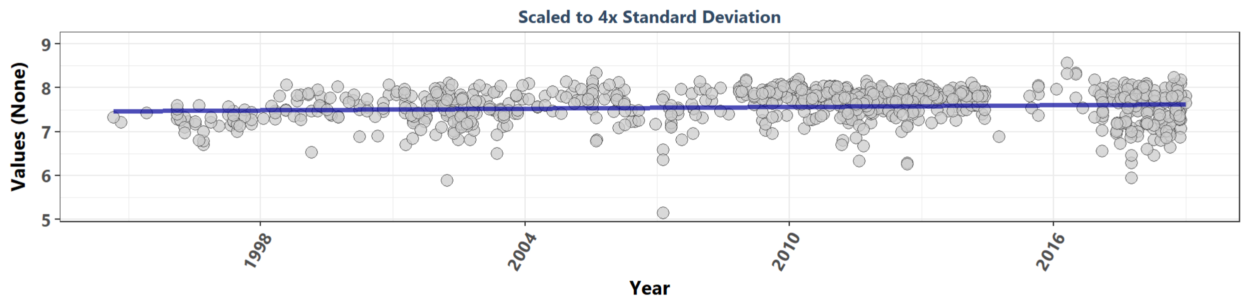


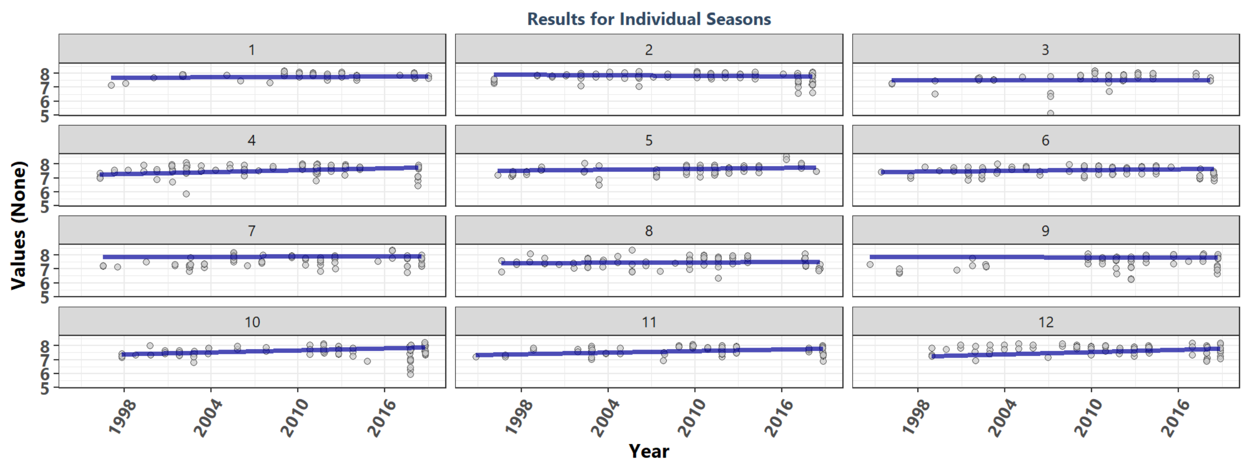
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	485	7.51	0.1677	0.0095	7.3650	4.7	0.0000	36.5	0.0001	1
1	25	7.83	0.1135	0.0059	7.4176	0.3	0.7759	NA	NA	1
2	46	7.71	0.2979	0.0183	7.3600	-1.4	0.1508	NA	NA	1
3	24	7.62	0.3986	0.0212	7.2549	2.7	0.0066	NA	NA	1
4	47	7.58	0.0011	0.0000	7.3700	3.0	0.0031	NA	NA	-1
5	46	7.43	0.2667	0.0169	7.1769	2.6	0.0088	NA	NA	1
6	43	7.37	0.5328	0.0425	6.6050	0.0	1.0000	NA	NA	1
7	44	7.47	0.3721	0.0267	7.0883	3.6	0.0003	NA	NA	1
8	52	7.36	0.0433	0.0040	7.7620	-1.2	0.2247	NA	NA	1
9	27	7.37	0.2402	0.0143	7.4886	3.9	0.0001	NA	NA	1
10	46	7.45	0.2628	0.0133	7.1967	2.6	0.0096	NA	NA	1
11	48	7.53	-0.1161	-0.0076	7.4682	1.1	0.2511	NA	NA	-1
12	37	7.76	-0.1469	-0.0082	7.8491	2.1	0.0365	NA	NA	-1

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

Guana Tolomato Matanzas National Estuarine Research Reserve



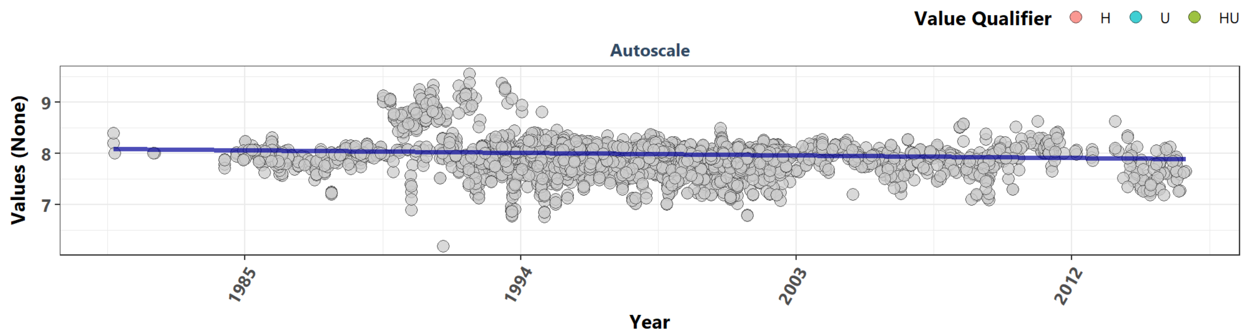


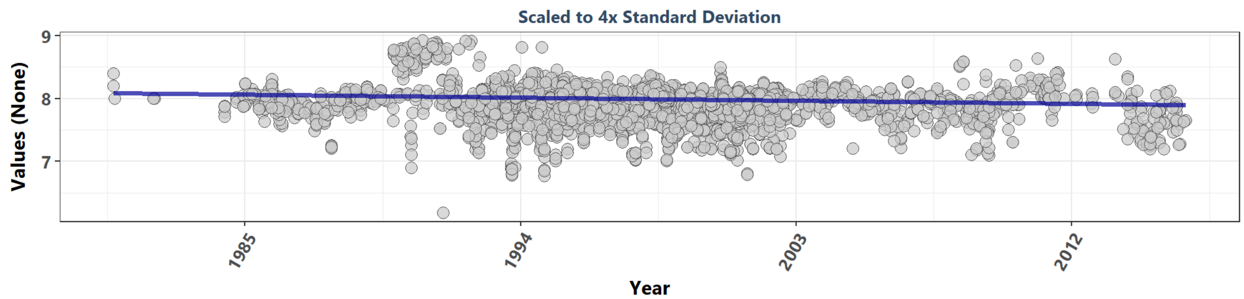


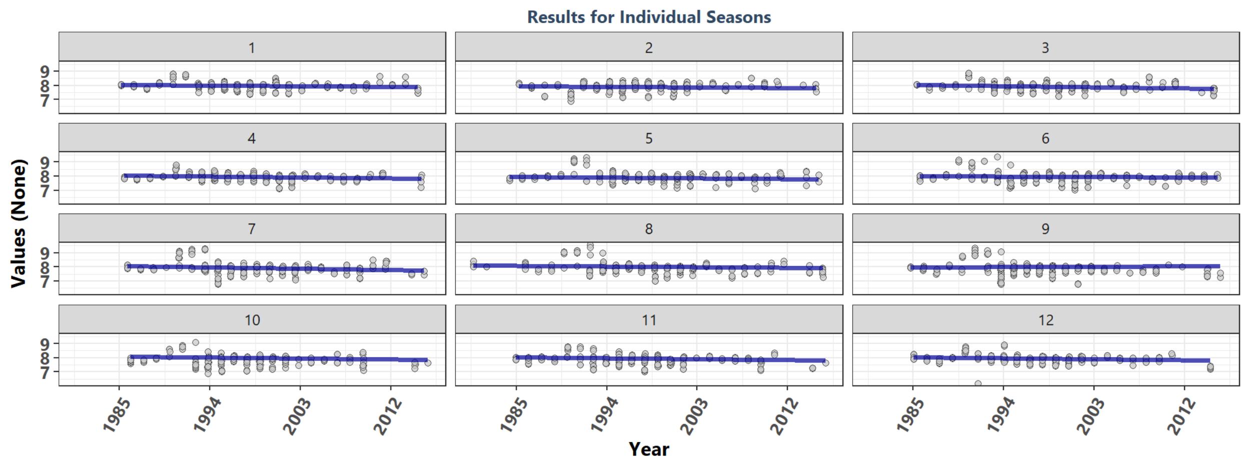
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	889	7.70	0.0917	0.0067	7.4467	3.6	0.0003	29.8	0.0017	1
1	53	7.87	0.0564	0.0040	7.6640	0.3	0.7751	NA	NA	1
2	90	7.82	-0.1006	-0.0069	7.9618	-0.6	0.5512	NA	NA	-1
3	52	7.70	-0.0178	-0.0010	7.5322	3.1	0.0019	NA	NA	-1
4	88	7.66	0.1857	0.0221	7.1579	1.8	0.0746	NA	NA	1
5	75	7.56	0.1288	0.0100	7.4850	4.5	0.0000	NA	NA	1
6	82	7.52	0.1268	0.0100	7.4000	-0.2	0.8158	NA	NA	1
7	77	7.62	0.0276	0.0014	7.8429	2.1	0.0385	NA	NA	1
8	82	7.46	0.0388	0.0033	7.4083	0.5	0.6068	NA	NA	1
9	58	7.60	-0.0427	-0.0025	7.8675	2.1	0.0388	NA	NA	-1
10	75	7.60	0.2956	0.0250	7.2200	1.6	0.1053	NA	NA	1
11	77	7.74	0.1603	0.0171	7.3114	0.7	0.4666	NA	NA	1
12	80	7.83	0.3535	0.0275	7.0650	-1.3	0.1858	NA	NA	1

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

Indian River-Malabar to Vero Beach Aquatic Preserve



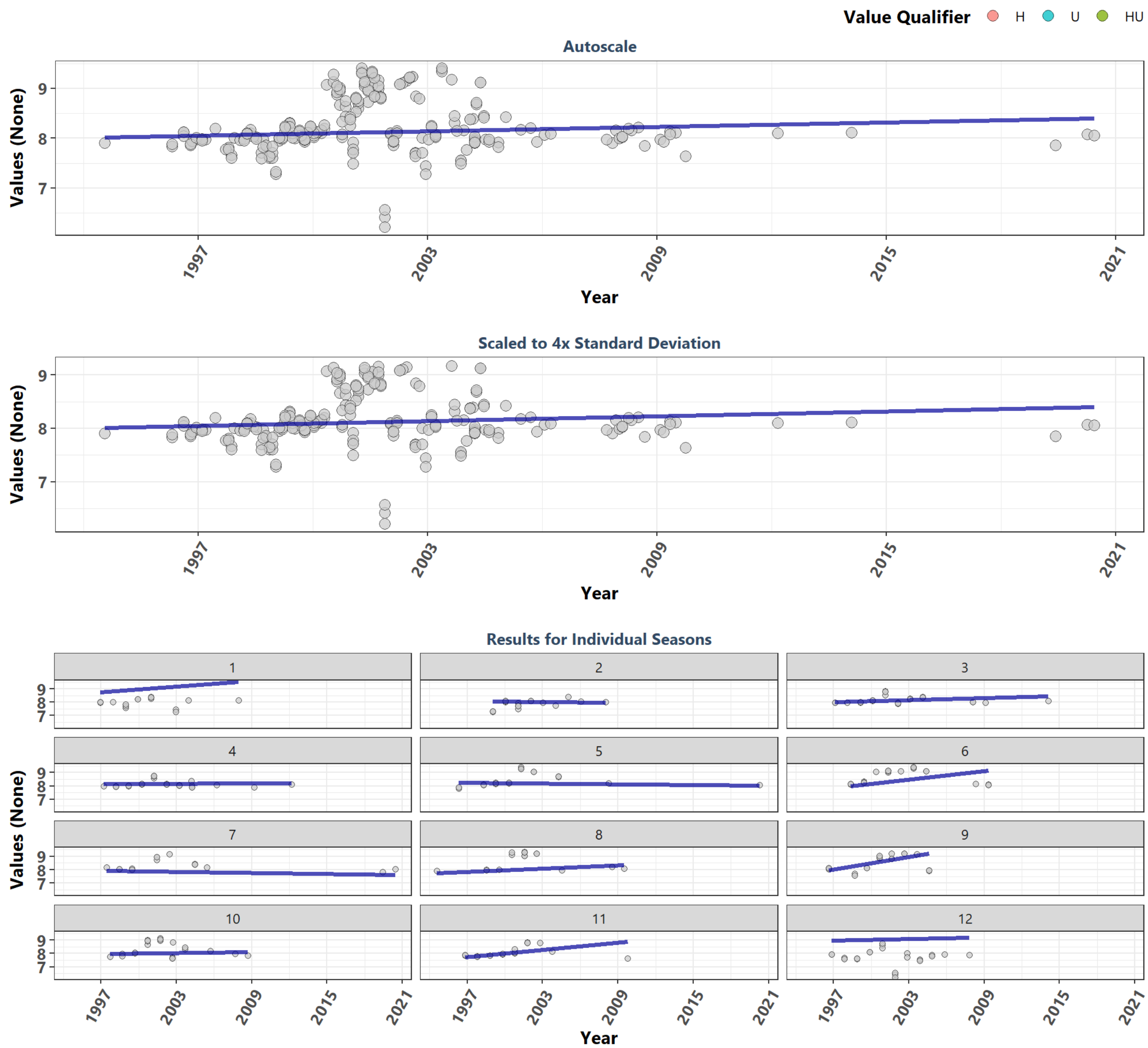




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2994	7.94	-0.0903	-0.0070	8.1833	-7.5	0.0000	23.4	0.0156	-1
1	232	8.00	-0.0733	-0.0050	8.1000	-2.4	0.0162	NA	NA	-1
2	231	8.01	-0.0650	-0.0050	8.0100	1.7	0.0916	NA	NA	-1
3	249	7.98	-0.1366	-0.0100	8.1800	-3.1	0.0019	NA	NA	-1
4	272	7.96	-0.1358	-0.0100	8.2200	-3.0	0.0023	NA	NA	-1
5	247	7.96	-0.0566	-0.0067	8.0600	-1.7	0.0851	NA	NA	-1
6	265	7.95	-0.0570	-0.0040	8.0620	-1.4	0.1655	NA	NA	-1
7	252	7.94	-0.1599	-0.0133	8.3033	-2.3	0.0188	NA	NA	-1
8	254	7.93	-0.1057	-0.0067	8.1867	-3.8	0.0001	NA	NA	-1
9	261	7.91	0.0743	0.0050	7.8700	-3.3	0.0010	NA	NA	1
10	247	7.88	-0.1318	-0.0100	8.2600	-1.3	0.1834	NA	NA	-1
11	247	7.88	-0.1236	-0.0100	8.2250	-1.5	0.1267	NA	NA	-1
12	237	7.95	-0.0990	-0.0100	8.2200	-3.1	0.0018	NA	NA	-1

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

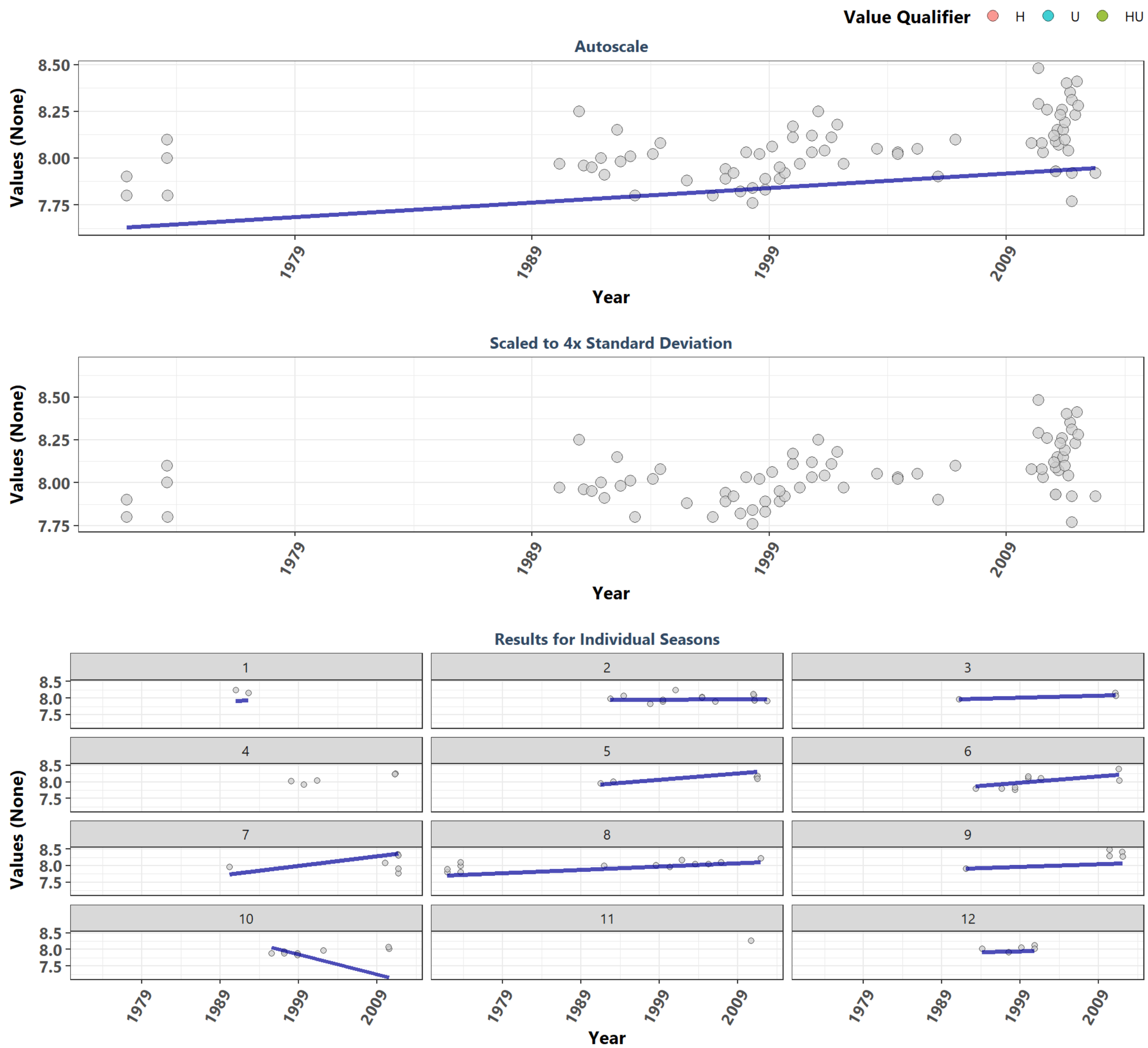
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	240	8.10	0.1369	0.0150	7.9583	2.4	0.0173	17.4	0.0956	1
1	16	8.08	0.2026	0.0700	8.3150	0.7	0.4929	NA	NA	1
2	20	8.00	-0.1773	-0.0100	8.1500	1.4	0.1732	NA	NA	-1
3	26	8.02	0.1333	0.0250	7.8550	1.1	0.2755	NA	NA	1
4	29	8.04	0.0110	0.0009	8.1709	-1.4	0.1751	NA	NA	1
5	20	8.23	-0.0143	-0.0088	8.2875	2.4	0.0165	NA	NA	-1
6	18	9.05	0.3895	0.1035	7.2414	1.1	0.2507	NA	NA	1
7	14	8.18	-0.0797	-0.0120	7.9740	0.0	1.0000	NA	NA	-1
8	16	9.06	0.2211	0.0400	7.6050	0.8	0.4078	NA	NA	1
9	18	8.49	0.3529	0.1607	7.0436	2.0	0.0423	NA	NA	1
10	21	8.20	0.1538	0.0082	7.9426	-0.1	0.9513	NA	NA	1
11	18	7.96	0.5098	0.0900	7.2000	2.9	0.0032	NA	NA	1
12	24	7.86	0.1583	0.0200	8.8600	-0.5	0.5997	NA	NA	1

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

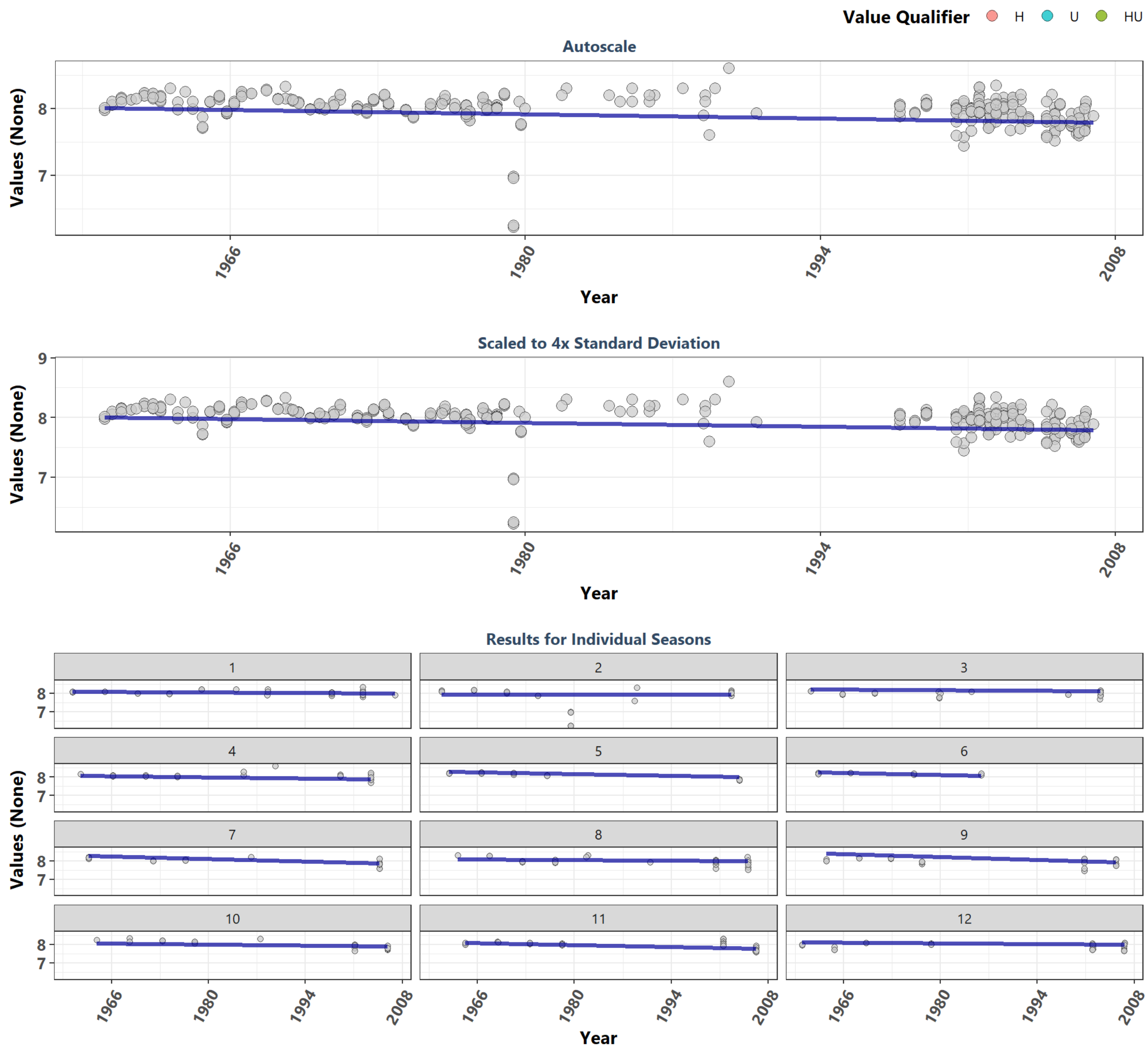
Jensen Beach to Jupiter Inlet Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	76	8.03	0.3780	0.0129	7.3473	3.7	0.0002	12.3	0.264	1
1	2	NA	0.5000	0.0250	7.0800	NA	NA	NA	NA	NA
2	14	7.96	0.0330	0.0025	7.8600	0.1	0.9113	NA	NA	1
3	3	NA	0.5897	0.0106	7.6169	NA	NA	NA	NA	NA
4	5	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	5	8.10	0.2000	0.0327	6.8190	1.7	0.0961	NA	NA	1
6	9	8.04	0.7000	0.0317	6.7733	1.9	0.0539	NA	NA	1
7	6	8.03	0.5278	0.0483	6.1550	0.0	1.0000	NA	NA	1
8	13	8.02	0.7000	0.0164	7.3473	2.8	0.0053	NA	NA	1
9	5	8.29	0.6667	0.0125	7.4950	0.3	0.7940	NA	NA	1
10	8	7.92	-1.0000	-0.1000	11.6500	1.9	0.0557	NA	NA	-1
11	1	NA	0.5714	0.0175	7.2500	NA	NA	NA	NA	NA
12	5	8.03	0.0667	0.0092	7.6033	1.0	0.3122	NA	NA	1

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

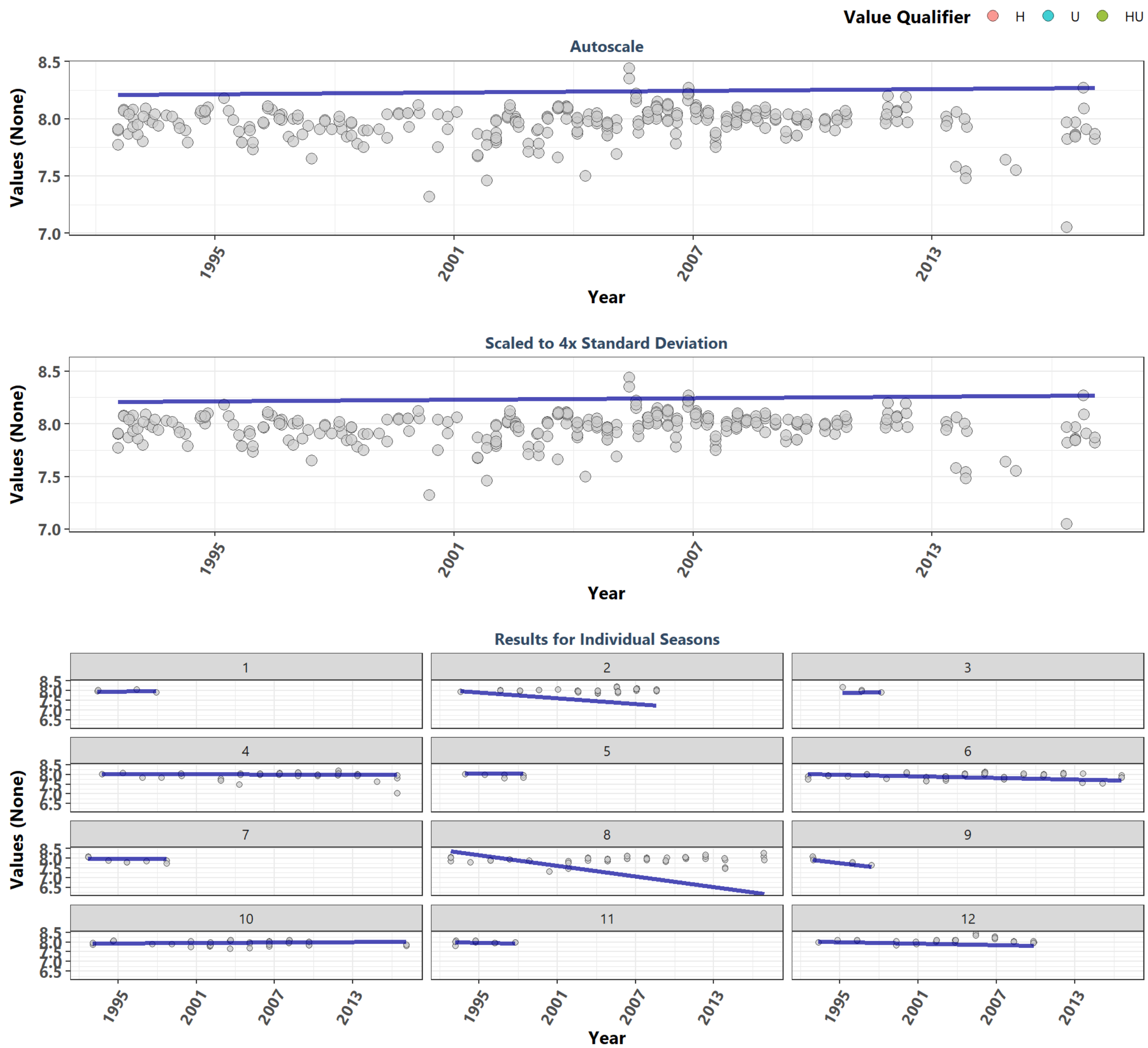
Lemon Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	360	8.02	-0.3272	-0.0213	9.2100	-8.3	0.0000	22.9	0.0181	-1
1	41	8.04	-0.1256	-0.0071	8.4757	-1.2	0.2445	NA	NA	-1
2	33	8.02	0.0129	0.0000	7.9400	-2.6	0.0081	NA	NA	-1
3	31	7.94	-0.3636	-0.0100	8.7650	0.1	0.9304	NA	NA	-1
4	31	8.05	-0.3768	-0.0200	9.2000	-1.7	0.0815	NA	NA	-1
5	20	8.18	-0.4283	-0.0300	9.9900	-4.0	0.0001	NA	NA	-1
6	12	8.18	-0.3118	-0.0329	10.1257	-1.7	0.0917	NA	NA	-1
7	22	8.02	-0.6368	-0.0433	10.7417	-3.6	0.0003	NA	NA	-1
8	31	7.99	-0.2404	-0.0117	8.7514	-2.5	0.0121	NA	NA	-1
9	29	8.00	-0.6695	-0.0500	11.2600	-2.9	0.0036	NA	NA	-1
10	27	8.01	-0.3182	-0.0200	9.2200	-5.0	0.0000	NA	NA	-1
11	45	8.04	-0.5281	-0.0350	10.1200	-4.3	0.0000	NA	NA	-1
12	38	7.99	-0.2172	-0.0114	8.7586	-2.2	0.0292	NA	NA	-1

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

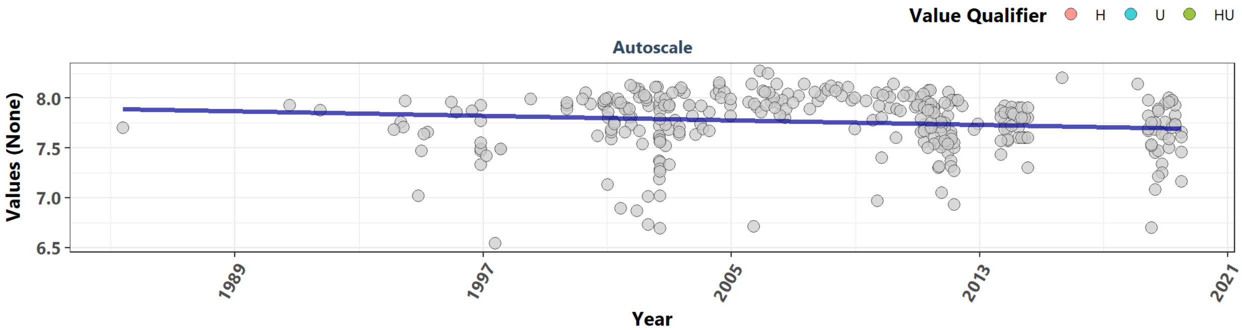
Loxahatchee River-Lake Worth Creek Aquatic Preserve

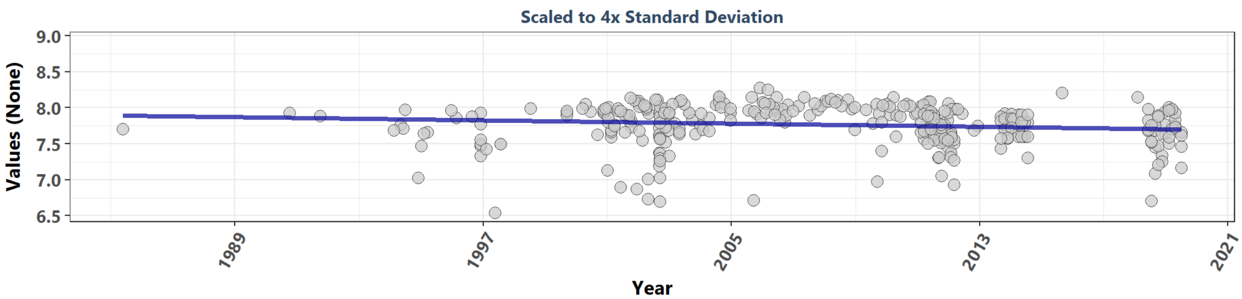


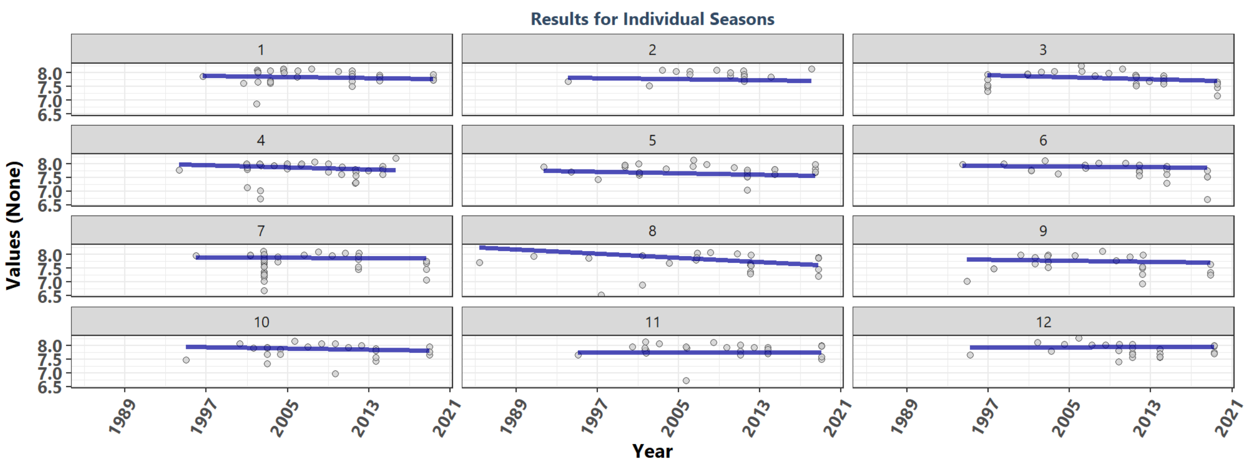
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	299	7.99	0.0035	0.0036	8.1642	1.7	0.0941	22.7	0.0195	0
1	4	8.00	0.1361	0.0067	7.8667	0.0	1.0000	NA	NA	0
2	35	8.00	-0.5556	-0.0733	8.9267	1.2	0.2493	NA	NA	0
3	4	8.02	0.2227	0.0111	7.7367	-1.4	0.1486	NA	NA	0
4	42	8.00	-0.0755	-0.0033	8.0733	-0.7	0.4848	NA	NA	0
5	7	7.97	0.0321	0.0018	8.0252	-1.3	0.2040	NA	NA	0
6	54	7.97	-0.4286	-0.0200	8.2700	0.9	0.3450	NA	NA	0
7	9	7.90	0.0244	0.0016	7.9489	-2.1	0.0379	NA	NA	0
8	51	7.97	-0.8333	-0.1350	10.0400	2.3	0.0208	NA	NA	0
9	6	7.86	-0.7333	-0.1167	9.3767	-2.1	0.0398	NA	NA	0
10	43	7.98	0.0887	0.0044	7.8831	0.2	0.8244	NA	NA	0
11	10	8.00	-0.2444	-0.0200	8.2550	-0.9	0.3525	NA	NA	0
12	34	8.06	-0.1667	-0.0200	8.2850	0.3	0.7988	NA	NA	0

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

Nassau River-St. Johns River Marshes Aquatic Preserve



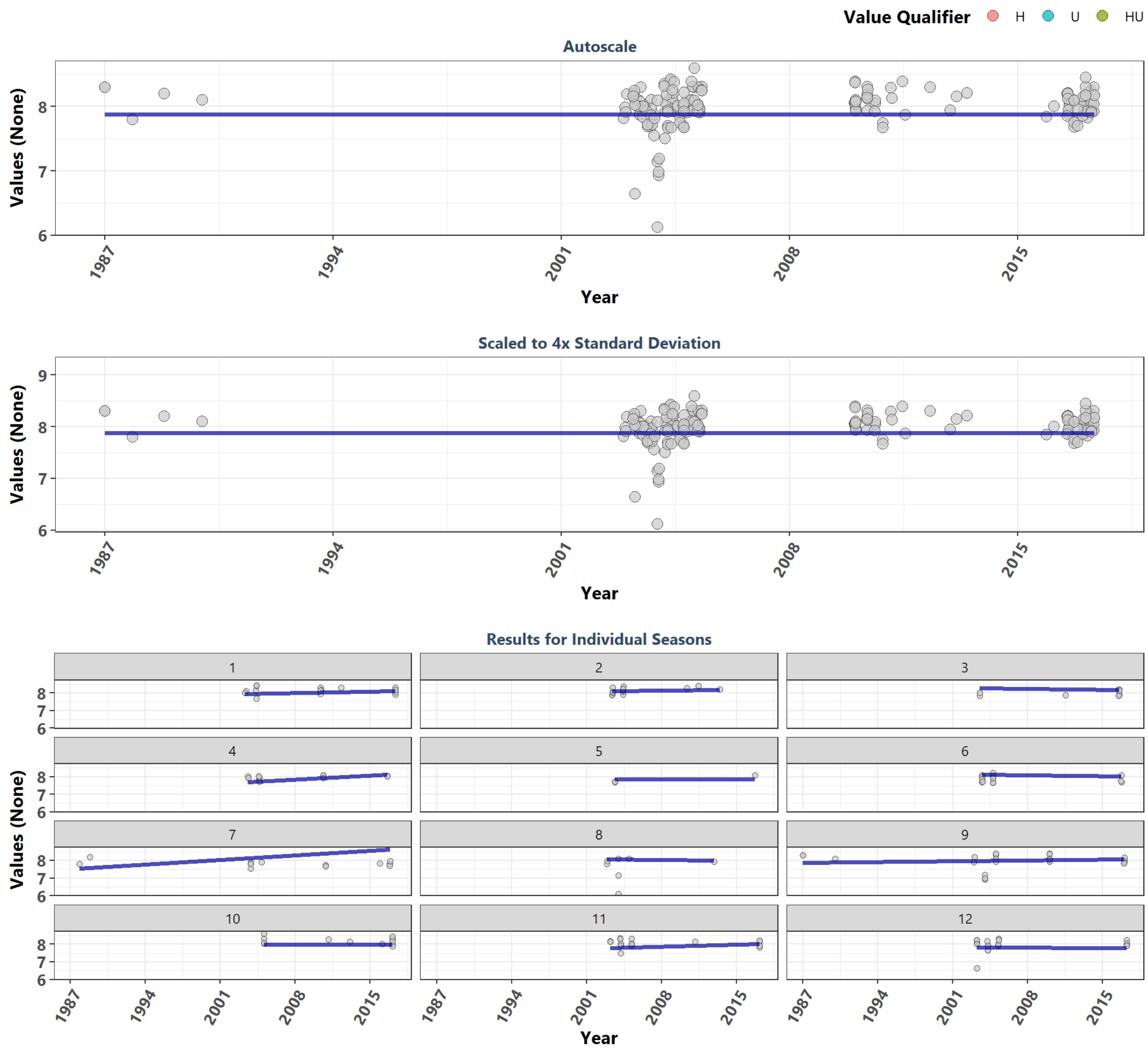




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	315	7.82	-0.1031	-0.0075	7.9883	-2.4	0.0175	6.9	0.8095	-1
1	31	7.88	-0.0695	-0.0065	8.0289	-0.2	0.8369	NA	NA	-1
2	16	7.94	-0.1108	-0.0062	7.9600	0.1	0.9268	NA	NA	-1
3	30	7.82	-0.1976	-0.0115	8.1667	-1.2	0.2267	NA	NA	-1
4	34	7.84	-0.1563	-0.0121	8.2069	-0.6	0.5696	NA	NA	-1
5	29	7.76	-0.0996	-0.0097	7.9309	-0.8	0.4039	NA	NA	-1
6	22	7.76	-0.0733	-0.0041	8.0166	-2.8	0.0047	NA	NA	-1
7	37	7.74	-0.0280	-0.0017	7.9283	0.0	0.9887	NA	NA	-1
8	22	7.75	-0.4329	-0.0250	8.5800	-0.6	0.5280	NA	NA	-1
9	22	7.65	-0.0996	-0.0064	7.9487	-0.6	0.5296	NA	NA	-1
10	24	7.88	-0.0652	-0.0071	8.0957	-0.4	0.6699	NA	NA	-1
11	25	7.89	0.0030	0.0000	7.7400	-0.5	0.6193	NA	NA	-1
12	23	7.80	0.0250	0.0020	7.8704	-1.3	0.1894	NA	NA	1

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

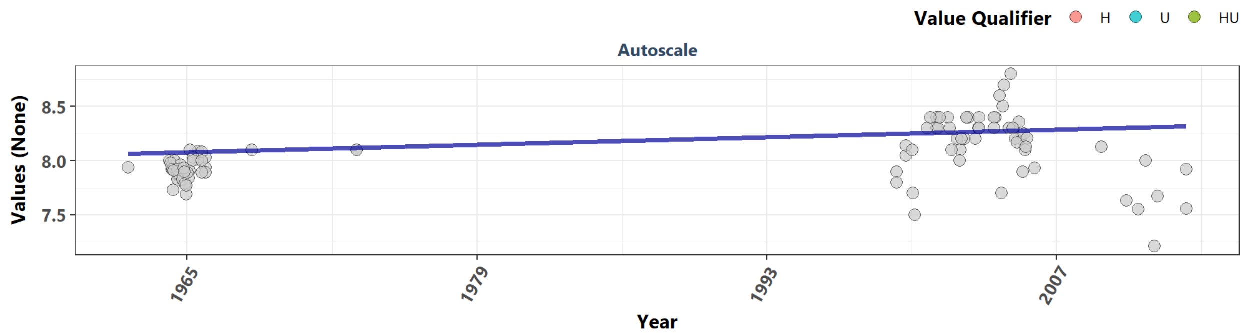
Nature Coast Aquatic Preserve

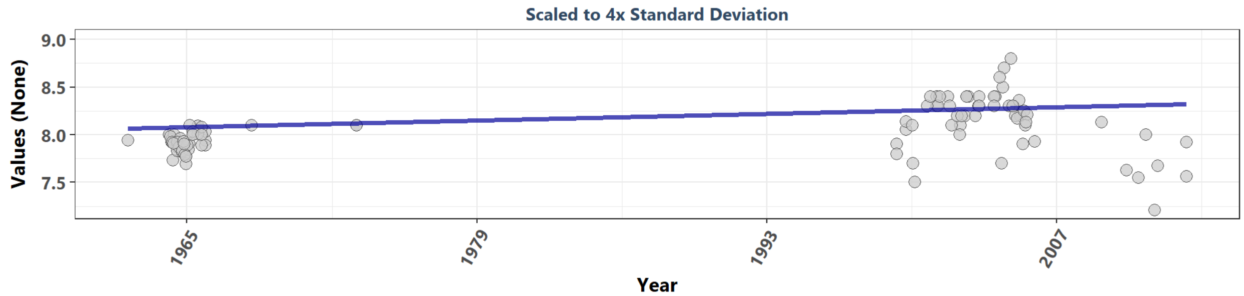


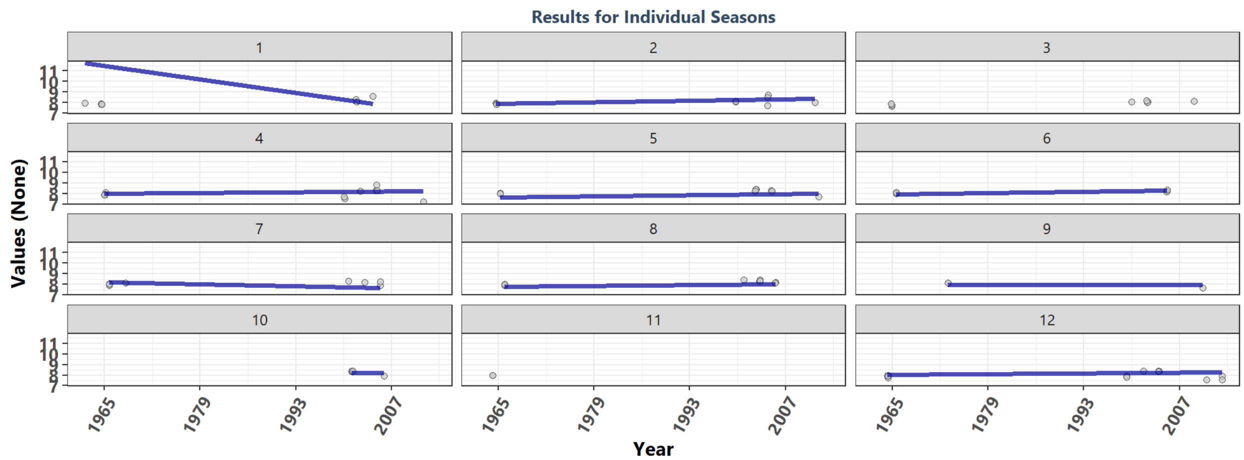
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	173	8.03	0.0490	0.0000	7.8744	0.1	0.9092	10.6	0.4776	0
1	18	8.14	0.0952	0.0130	7.7190	0.3	0.7518	NA	NA	0
2	14	8.17	0.0588	0.0033	8.0650	2.0	0.0448	NA	NA	0
3	12	8.00	-0.0897	-0.0067	8.3633	1.3	0.1803	NA	NA	0
4	12	8.00	0.6667	0.0304	7.2035	1.4	0.1651	NA	NA	0
5	3	NA	-0.0190	-0.0008	7.8738	NA	NA	NA	NA	NA
6	15	7.86	-0.1281	-0.0060	8.2080	-0.1	0.9569	NA	NA	0
7	12	7.80	0.3846	0.0371	7.5014	-0.2	0.8334	NA	NA	0
8	7	7.94	-0.1397	-0.0083	8.2100	0.2	0.8725	NA	NA	0
9	29	8.07	0.3030	0.0067	7.8750	-1.0	0.3162	NA	NA	0
10	13	8.17	0.0429	0.0015	7.9423	-0.4	0.6930	NA	NA	0
11	17	8.06	0.2576	0.0146	7.5615	-0.8	0.4402	NA	NA	0
12	21	7.97	-0.0606	-0.0025	7.8575	0.3	0.8017	NA	NA	0

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

Pine Island Sound Aquatic Preserve



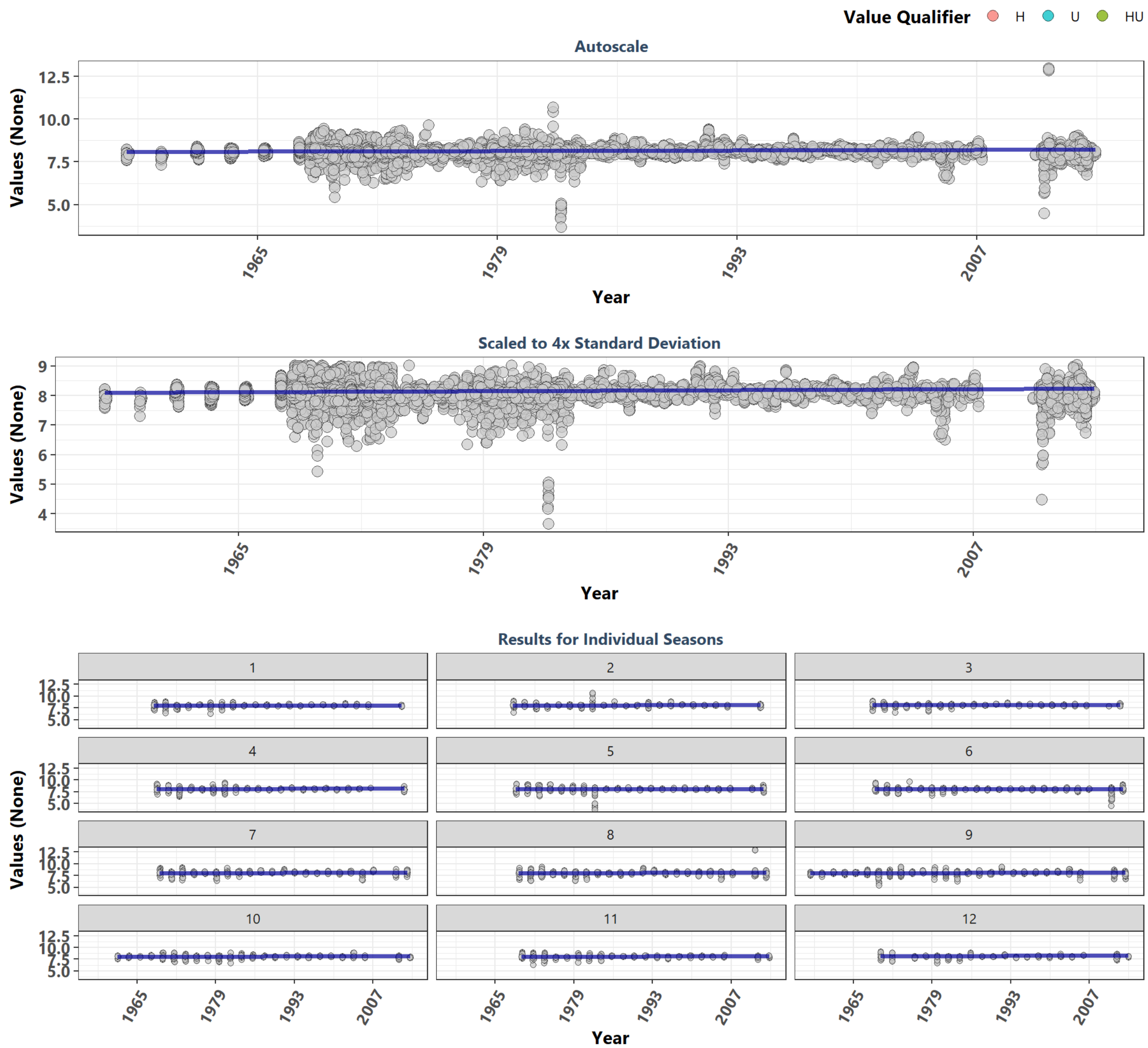




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	98	8.04	0.1965	0.0115	7.5500	2.6	0.0095	18.4	0.0482	1
1	8	7.93	-0.4000	-0.2100	21.1100	1.6	0.1054	NA	NA	-1
2	10	7.98	0.4643	0.0238	6.8375	1.3	0.1852	NA	NA	1
3	11	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	11	7.91	0.3571	0.0112	7.4825	0.7	0.5148	NA	NA	1
5	9	8.20	0.1636	0.0194	6.7281	0.0	1.0000	NA	NA	1
6	6	NA	0.5818	0.0187	7.0675	NA	NA	NA	NA	NA
7	8	8.09	-0.6667	-0.0294	9.5394	1.2	0.2479	NA	NA	-1
8	10	8.17	0.3333	0.0147	7.0853	0.6	0.5714	NA	NA	1
9	3	NA	-0.0095	0.0000	7.9200	NA	NA	NA	NA	NA
10	5	NA	-0.0278	-0.0083	8.7115	NA	NA	NA	NA	NA
11	2	NA	0.1556	0.0100	7.5500	NA	NA	NA	NA	NA
12	15	7.92	0.6000	0.0094	7.6171	0.0	1.0000	NA	NA	1

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

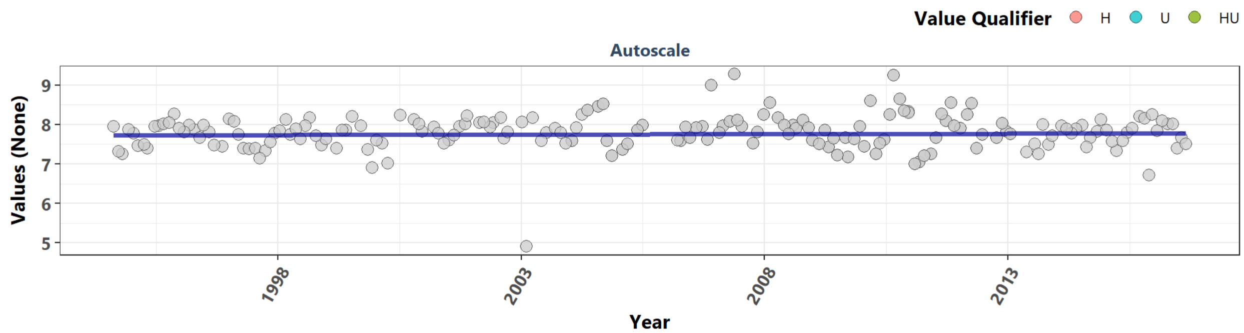
Pinellas County Aquatic Preserve

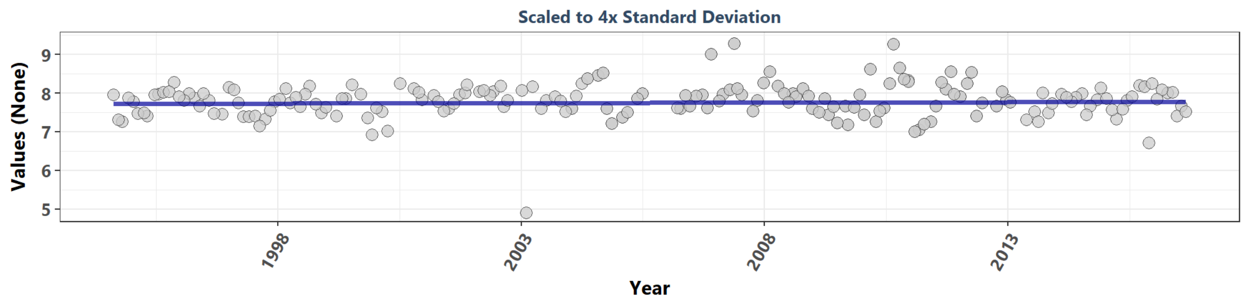


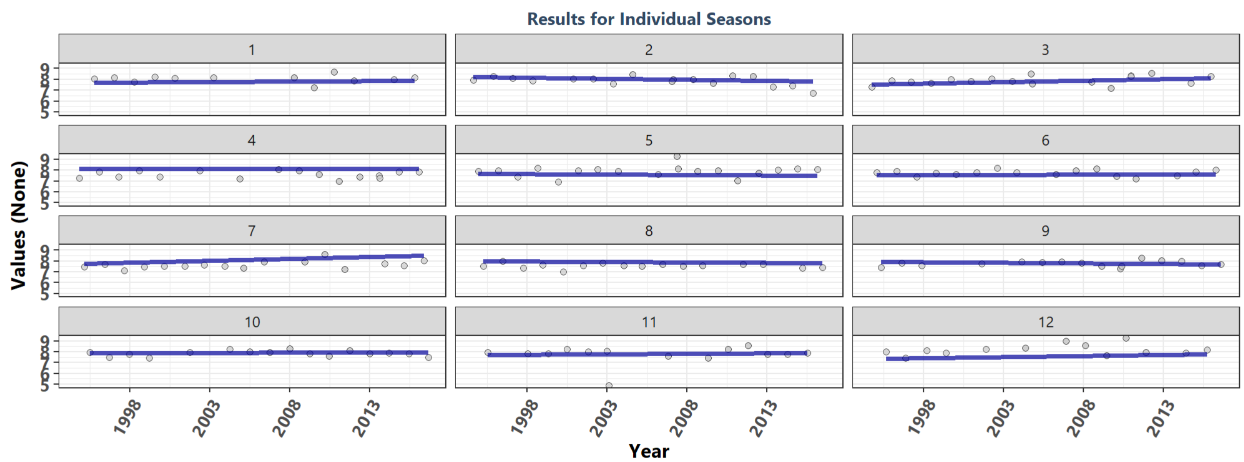
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	12409	8.09	0.0704	0.0050	7.9084	11.2	0.0000	121.5	0	1
1	502	8.07	-0.0104	0.0000	8.0800	2.6	0.0090	NA	NA	-1
2	566	8.07	0.0528	0.0025	7.9575	5.8	0.0000	NA	NA	1
3	497	8.09	0.0477	0.0024	7.9824	3.7	0.0002	NA	NA	1
4	803	8.04	0.1591	0.0067	7.7367	5.9	0.0000	NA	NA	1
5	821	8.08	0.0183	0.0006	8.0318	2.3	0.0222	NA	NA	1
6	650	8.07	-0.0628	-0.0022	8.1833	-2.4	0.0162	NA	NA	-1
7	561	8.10	0.1351	0.0067	7.7200	1.7	0.0896	NA	NA	1
8	1646	8.06	0.1634	0.0060	7.7720	1.1	0.2600	NA	NA	1
9	4247	8.11	0.0776	0.0037	7.8775	10.2	0.0000	NA	NA	1
10	1243	8.08	0.1095	0.0043	7.8757	-0.6	0.5814	NA	NA	1
11	454	8.07	0.0625	0.0027	7.9393	2.0	0.0457	NA	NA	1
12	419	8.09	0.1031	0.0067	7.8167	4.9	0.0000	NA	NA	1

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

Rocky Bayou State Park Aquatic Preserve



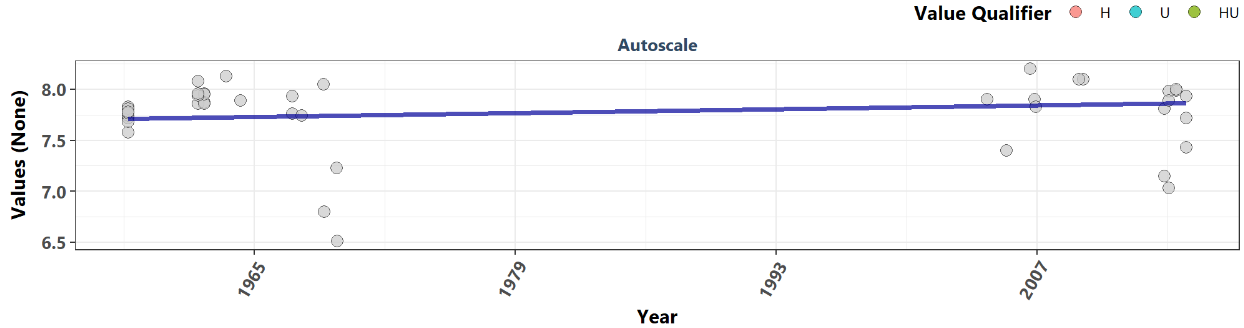


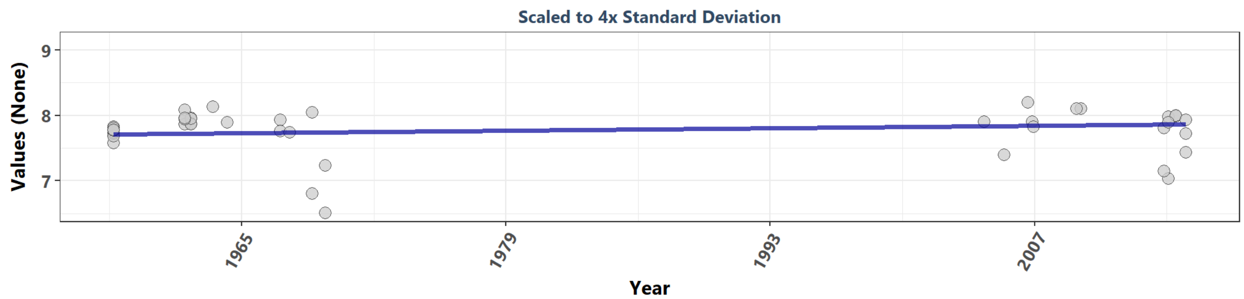


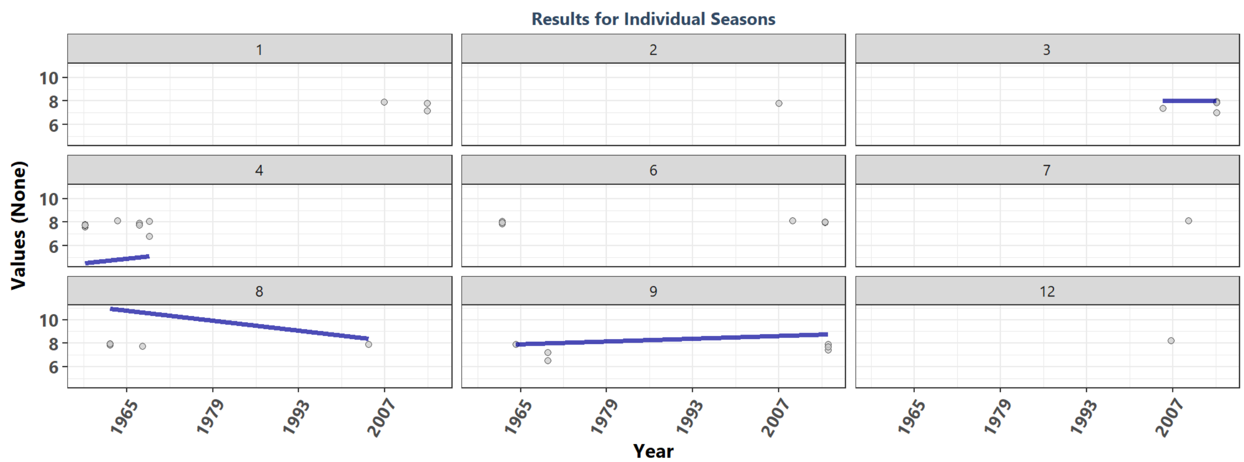
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	258	7.84	0.0399	0.0033	7.7018	0.8	0.4283	8.1	0.7069	0
1	16	8.13	0.1014	0.0113	7.6059	0.1	0.8917	NA	NA	0
2	24	7.97	-0.2065	-0.0244	8.3906	-1.4	0.1622	NA	NA	0
3	22	7.81	0.1861	0.0343	7.3007	1.2	0.2330	NA	NA	0
4	22	7.57	0.0333	0.0025	8.0838	-0.2	0.8649	NA	NA	0
5	24	7.94	-0.0303	-0.0091	7.7336	0.3	0.7836	NA	NA	0
6	21	7.78	0.0751	0.0027	7.5560	-0.3	0.7385	NA	NA	0
7	21	7.57	0.1345	0.0433	7.4633	1.7	0.0951	NA	NA	0
8	23	7.60	-0.1107	-0.0097	8.0547	0.5	0.6316	NA	NA	0
9	24	7.81	-0.0571	-0.0140	8.0180	0.7	0.4994	NA	NA	0
10	23	7.89	0.0435	0.0033	7.8883	-0.7	0.4733	NA	NA	0
11	19	7.85	0.0877	0.0100	7.6700	0.5	0.6220	NA	NA	0
12	19	8.20	0.2667	0.0250	7.1700	0.8	0.4382	NA	NA	0

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

Rookery Bay National Estuarine Research Reserve



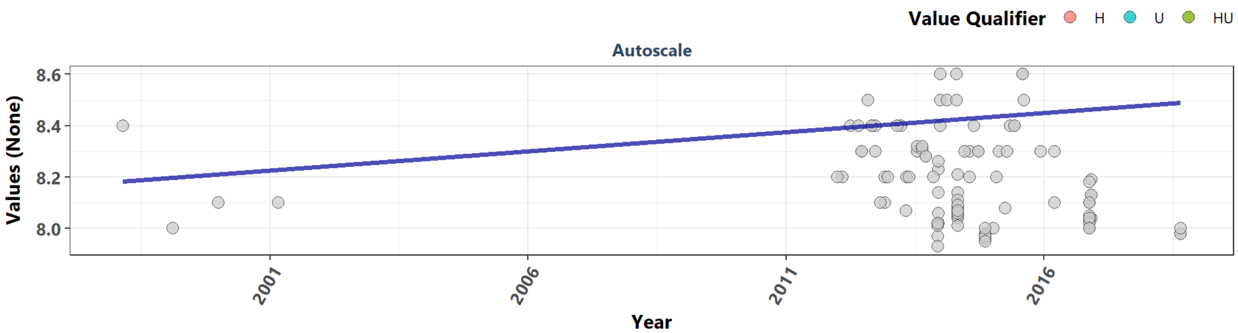


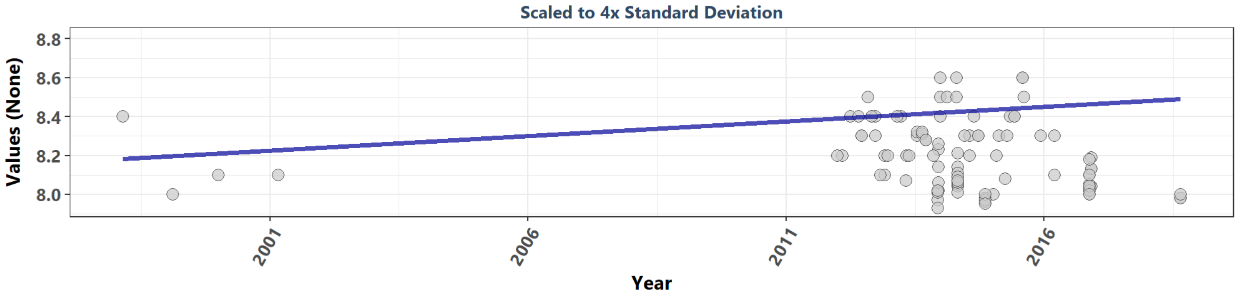


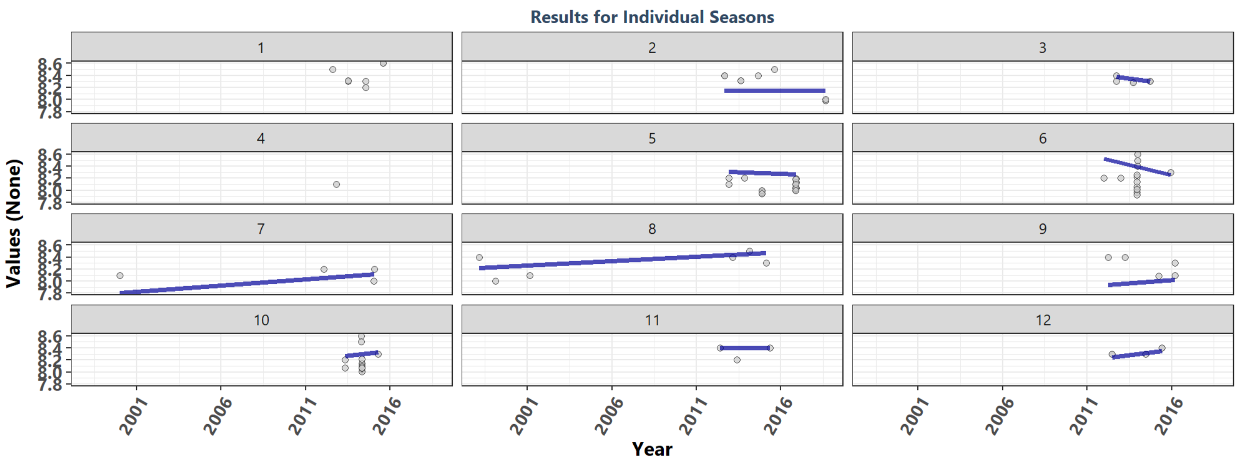
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	44	7.85	0.0720	0.0047	7.5517	1.0	0.2958	4.3	0.5106	0
1	3	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	4	NA	0.2857	0.0015	7.9360	NA	NA	NA	NA	NA
4	15	7.77	0.1667	0.0980	1.1770	0.9	0.3702	NA	NA	0
6	7	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	1	NA	-0.2000	-0.0025	7.9750	NA	NA	NA	NA	NA
8	6	7.88	-0.6667	-0.1050	14.7400	-0.5	0.6520	NA	NA	0
9	6	7.57	0.1524	0.0310	6.7160	0.4	0.6810	NA	NA	0
12	1	NA	0.2000	0.0077	7.1673	NA	NA	NA	NA	NA

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

St. Andrews State Park Aquatic Preserve



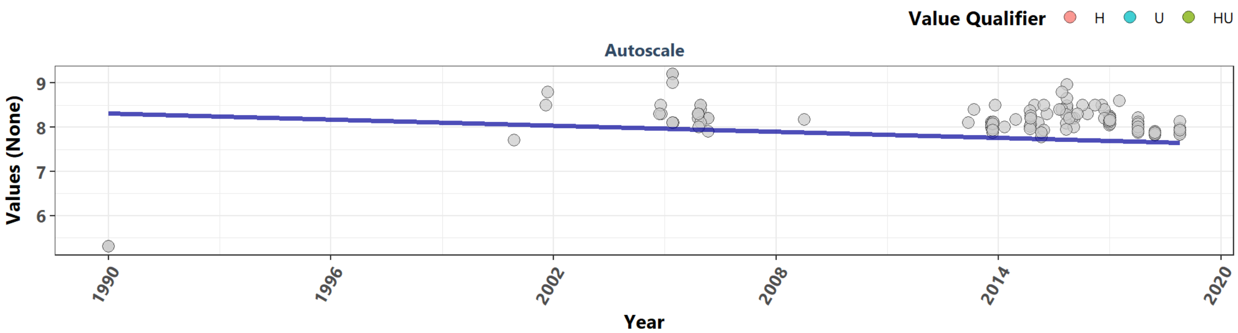


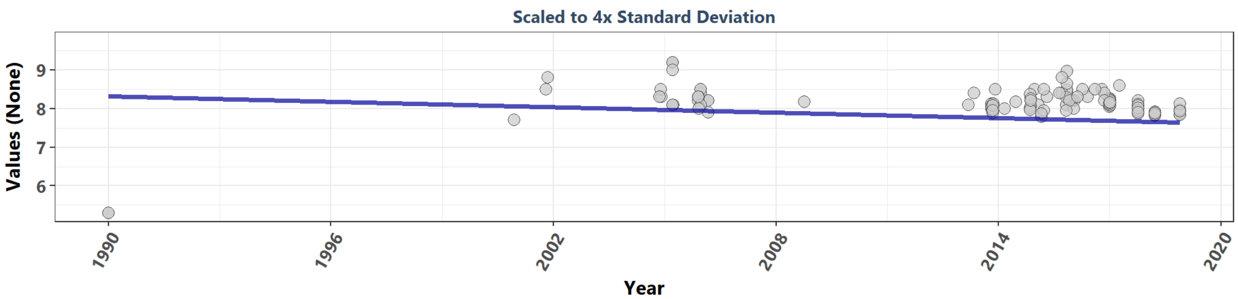


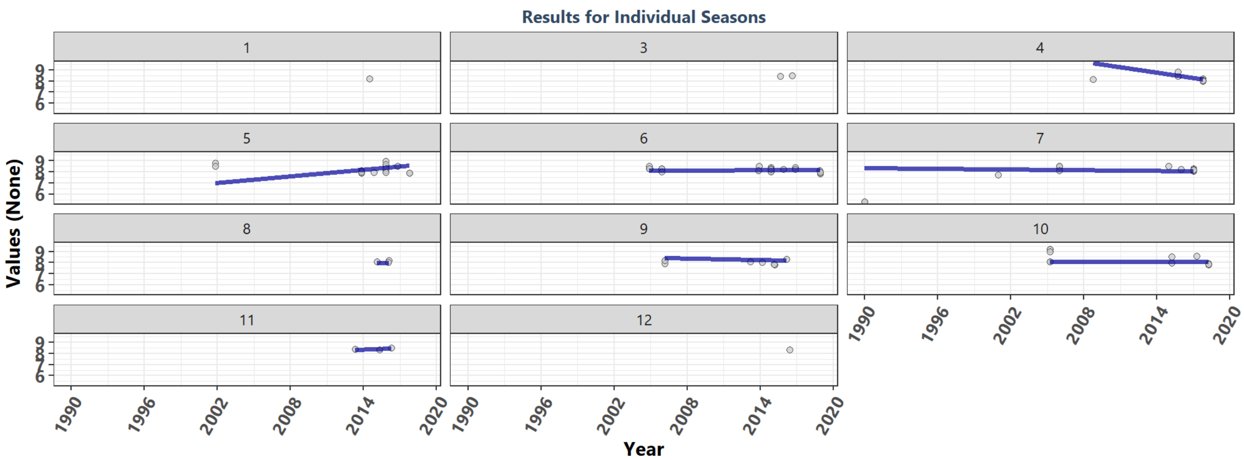
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	106	8.16	0.0517	0.0150	8.1500	1.3	0.1804	10.4	0.4079	0
1	7	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	9	8.32	0.0000	0.0000	8.1500	-1.5	0.1226	NA	NA	0
3	6	8.30	-0.4000	-0.0333	8.9333	-0.2	0.8291	NA	NA	0
4	1	NA	0.0250	0.0175	7.8550	NA	NA	NA	NA	NA
5	26	8.04	-0.1333	-0.0100	8.4800	1.5	0.1363	NA	NA	0
6	16	8.17	-0.4167	-0.0680	9.6120	0.1	0.8936	NA	NA	0
7	4	8.15	0.0654	0.0200	7.7300	0.0	1.0000	NA	NA	0
8	6	8.35	0.2667	0.0143	8.1929	0.6	0.5661	NA	NA	0
9	5	8.30	0.1846	0.0200	7.6150	-0.8	0.4350	NA	NA	0
10	18	8.09	0.1429	0.0333	7.6867	0.5	0.5935	NA	NA	0
11	3	8.40	0.0000	0.0000	8.4000	0.0	1.0000	NA	NA	0
12	5	8.30	0.6000	0.0333	7.7000	1.5	0.1281	NA	NA	0

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

St. Joseph Bay Aquatic Preserve



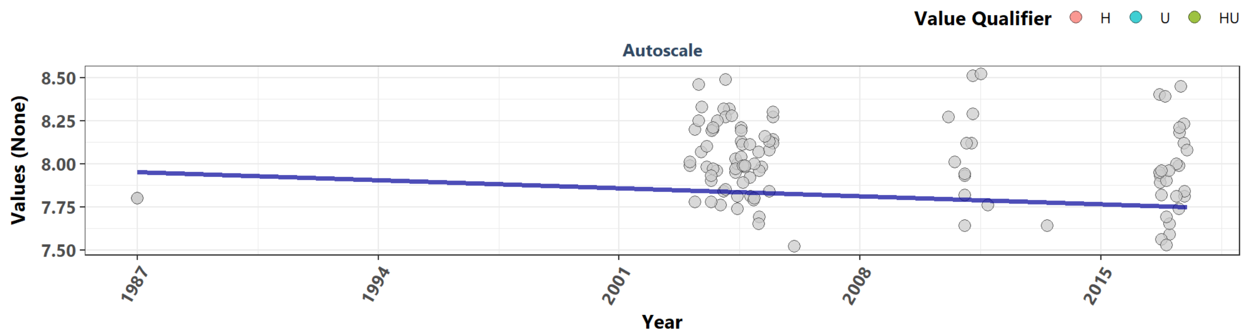


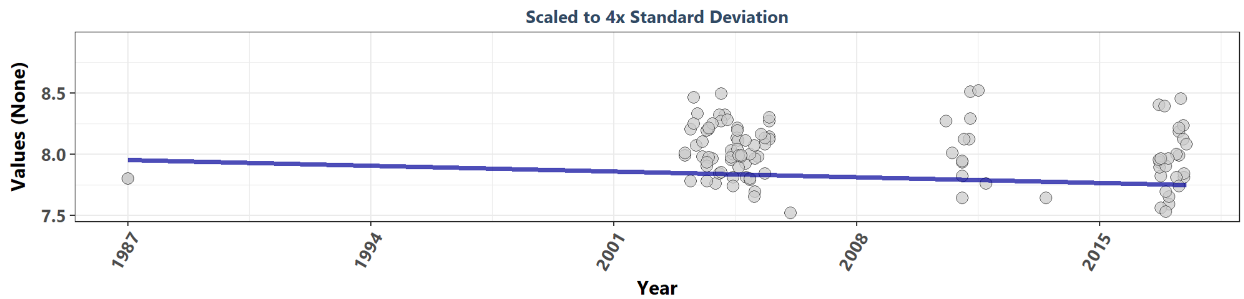


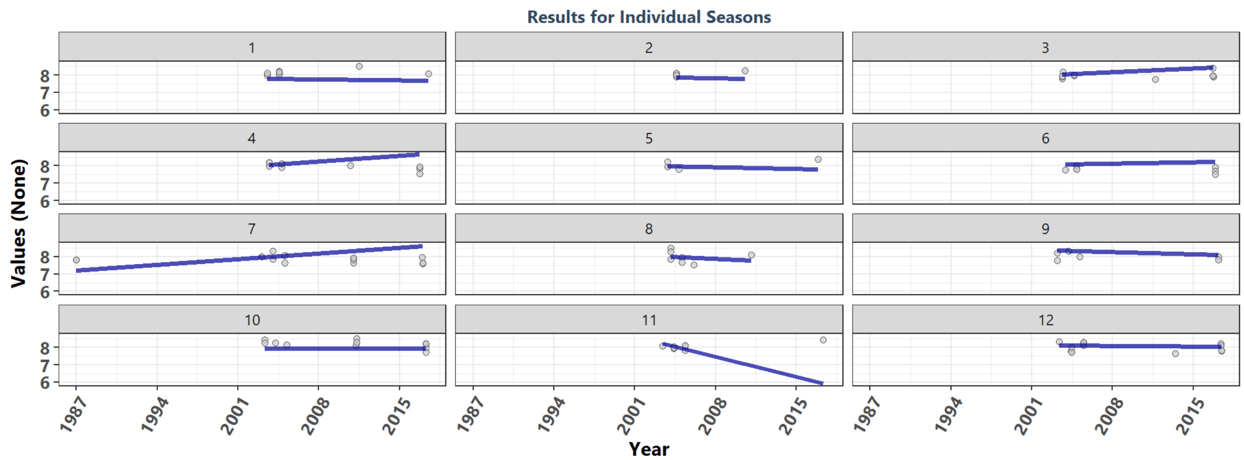
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	123	8.11	-0.2360	-0.0233	8.3200	-4.4	0.0000	26.9	0.0007	-1
1	1	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	2	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	10	8.09	-0.3778	-0.1600	12.5750	-1.8	0.0721	NA	NA	-1
5	23	8.08	1.0000	0.1000	5.8000	-0.2	0.8437	NA	NA	1
6	27	8.20	0.0476	0.0045	8.0373	-3.6	0.0003	NA	NA	1
7	21	8.16	-0.0316	-0.0100	8.3200	0.3	0.7603	NA	NA	-1
8	3	NA	-0.2778	-0.0196	8.4714	NA	NA	NA	NA	NA
9	9	8.00	-0.4815	-0.0264	8.8607	-1.0	0.3232	NA	NA	-1
10	23	7.91	0.0000	0.0000	8.1000	-4.0	0.0001	NA	NA	-1
11	3	8.40	0.3333	0.0333	7.5667	0.0	1.0000	NA	NA	1
12	1	NA	-0.5455	-0.0600	9.5900	NA	NA	NA	NA	NA

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

St. Martins Marsh Aquatic Preserve



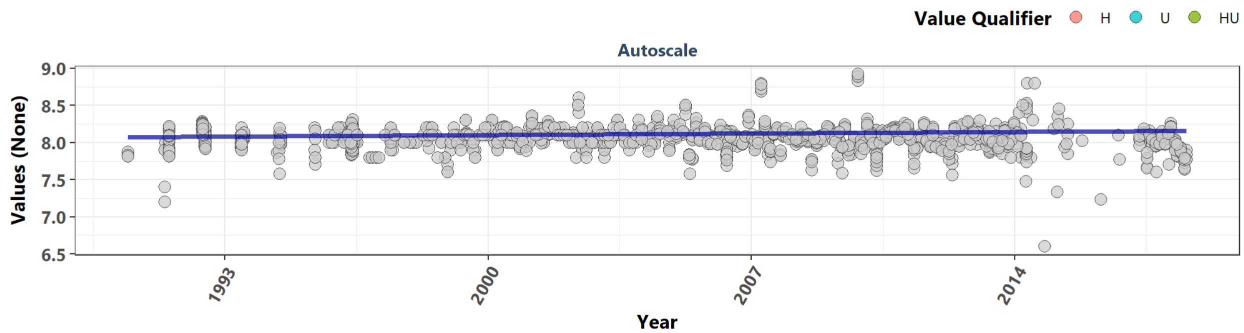


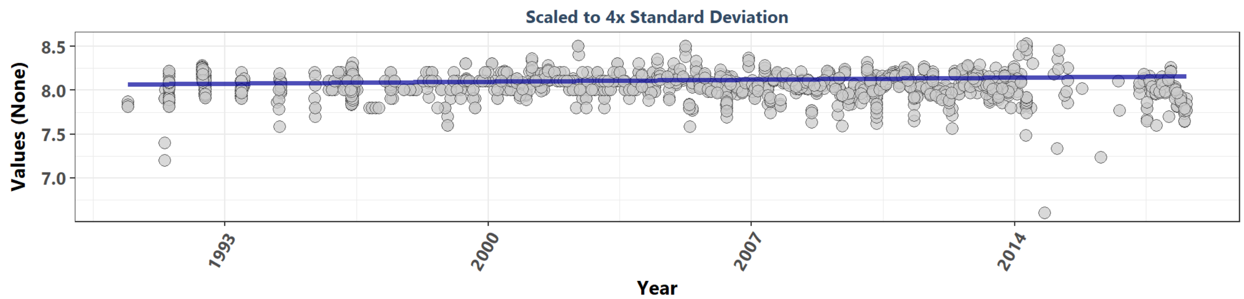


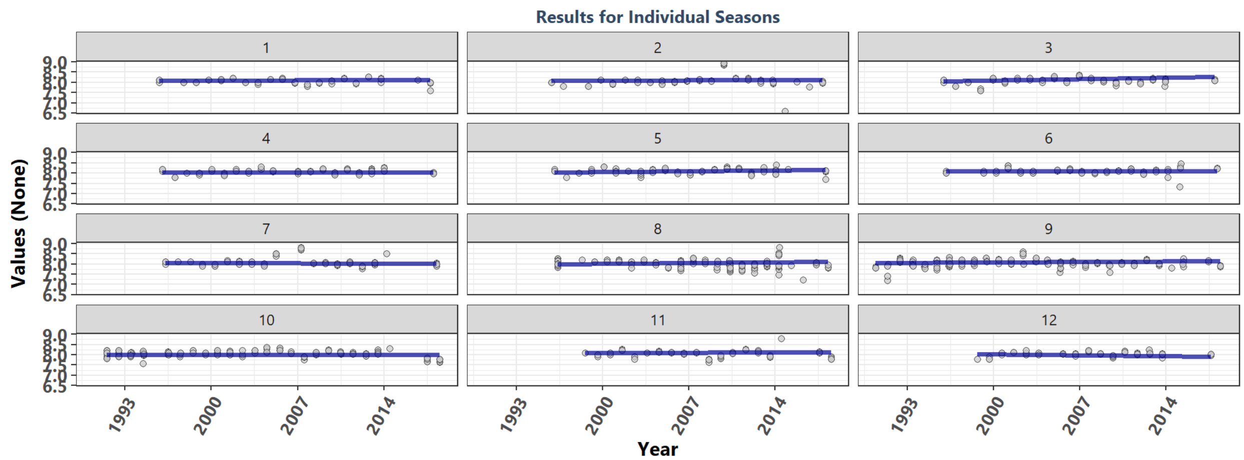
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	103	7.99	-0.0896	-0.0067	7.9515	-1.6	0.1128	14.8	0.1931	0
1	8	8.12	-0.1429	-0.0083	7.9400	1.1	0.2836	NA	NA	0
2	4	NA	-0.2000	-0.0068	7.9630	NA	NA	NA	NA	NA
3	11	7.93	0.2857	0.0304	7.5177	0.2	0.8685	NA	NA	0
4	9	7.97	0.3214	0.0471	7.2664	-2.2	0.0288	NA	NA	0
5	4	8.11	-0.0667	-0.0142	8.2379	0.0	1.0000	NA	NA	0
6	7	7.79	0.1667	0.0108	7.9165	-0.3	0.7423	NA	NA	0
7	15	7.84	0.5000	0.0467	7.2100	-1.0	0.3113	NA	NA	0
8	7	7.96	-0.5833	-0.0286	8.4843	-1.0	0.3408	NA	NA	0
9	6	7.99	-0.3818	-0.0186	8.6557	0.0	1.0000	NA	NA	0
10	11	8.21	0.0545	0.0019	7.8954	-1.6	0.1044	NA	NA	0
11	8	8.05	-0.3333	-0.1650	10.9300	0.9	0.3580	NA	NA	0
12	13	8.12	-0.0897	-0.0058	8.2250	-0.4	0.7021	NA	NA	0

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

Terra Ceia Aquatic Preserve



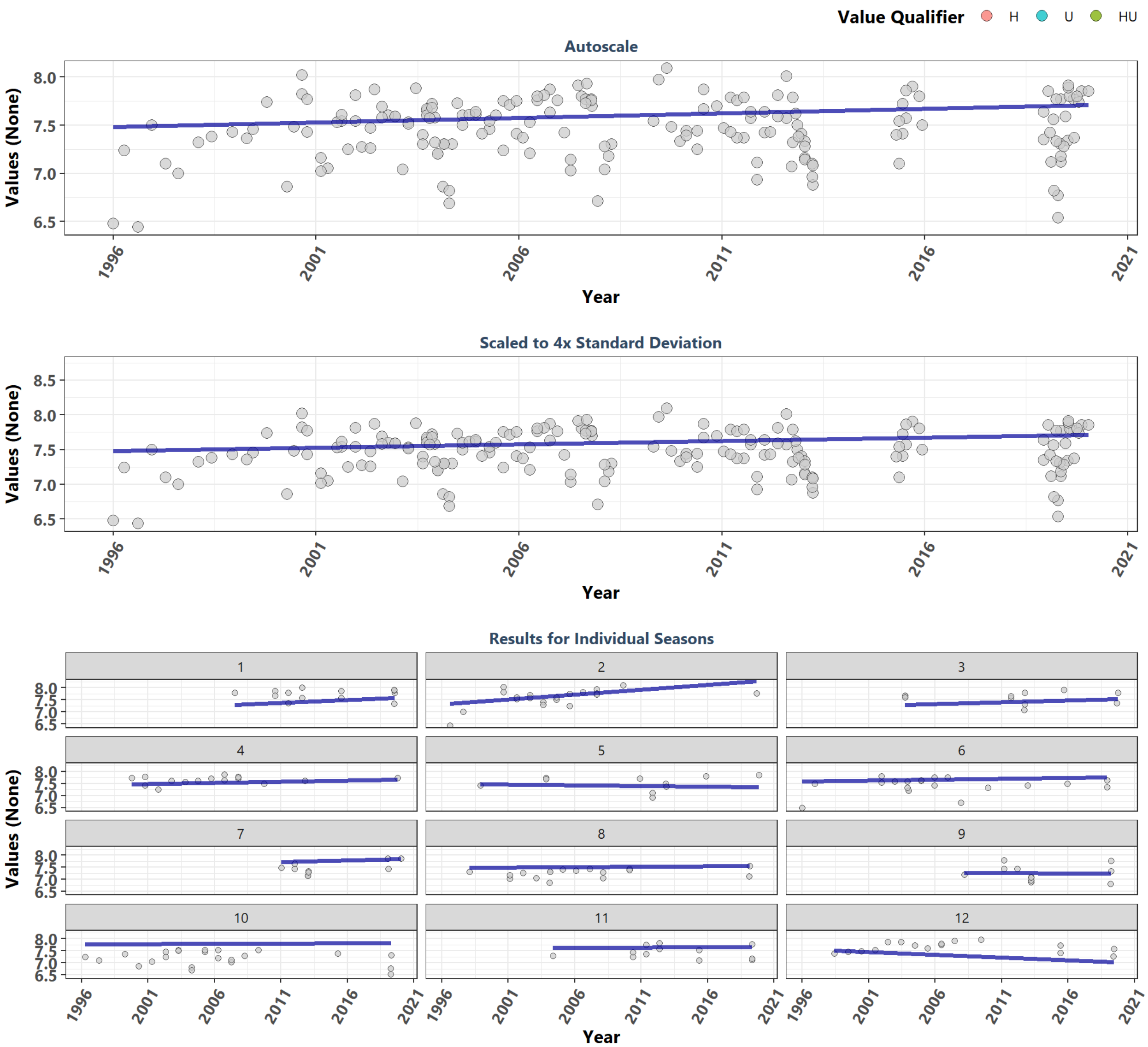




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1951	8.06	0.0442	0.0031	8.0567	4.3	0.0000	93.9	0	1
1	76	8.00	0.0569	0.0017	8.0658	-0.6	0.5678	NA	NA	1
2	75	8.05	0.0287	0.0008	8.0750	1.9	0.0563	NA	NA	1
3	83	8.10	0.1987	0.0100	7.9400	1.4	0.1579	NA	NA	1
4	69	8.10	-0.0143	0.0000	8.0400	0.9	0.3670	NA	NA	-1
5	78	8.10	0.1656	0.0044	8.0022	0.7	0.4614	NA	NA	1
6	70	8.10	-0.0490	-0.0007	8.1150	2.0	0.0417	NA	NA	-1
7	66	8.03	-0.0965	-0.0020	8.0720	-1.1	0.2514	NA	NA	-1
8	326	8.05	0.1499	0.0050	7.9450	-5.7	0.0000	NA	NA	1
9	546	8.04	0.1052	0.0025	8.0475	7.2	0.0000	NA	NA	1
10	431	8.08	-0.0446	0.0000	8.0000	0.9	0.3588	NA	NA	-1
11	69	8.10	0.0742	0.0014	8.0700	-0.6	0.5529	NA	NA	1
12	62	8.04	-0.1893	-0.0070	8.1200	-0.2	0.8739	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	182	7.50	0.1126	0.0094	7.4842	2.2	0.0272	9	0.6259	1
1	13	7.80	0.1111	0.0233	7.0233	0.2	0.8508	NA	NA	1
2	20	7.65	0.3368	0.0400	7.3100	2.1	0.0398	NA	NA	1
3	11	7.64	0.3392	0.0150	7.1800	0.0	1.0000	NA	NA	1
4	16	7.66	0.2222	0.0103	7.4250	0.9	0.3642	NA	NA	1
5	10	7.59	-0.0152	-0.0057	7.4964	0.8	0.4152	NA	NA	-1
6	21	7.50	0.1750	0.0075	7.5862	0.3	0.7376	NA	NA	1
7	10	7.42	0.1417	0.0120	7.5460	0.4	0.7063	NA	NA	1
8	19	7.30	0.0571	0.0031	7.4719	2.0	0.0422	NA	NA	1
9	11	7.18	-0.0079	-0.0013	7.2712	-1.1	0.2898	NA	NA	-1
10	23	7.26	0.0513	0.0017	7.7705	0.0	0.9788	NA	NA	1
11	12	7.40	0.0182	0.0014	7.6157	0.0	1.0000	NA	NA	1
12	16	7.66	-0.2545	-0.0233	7.5767	0.7	0.4699	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

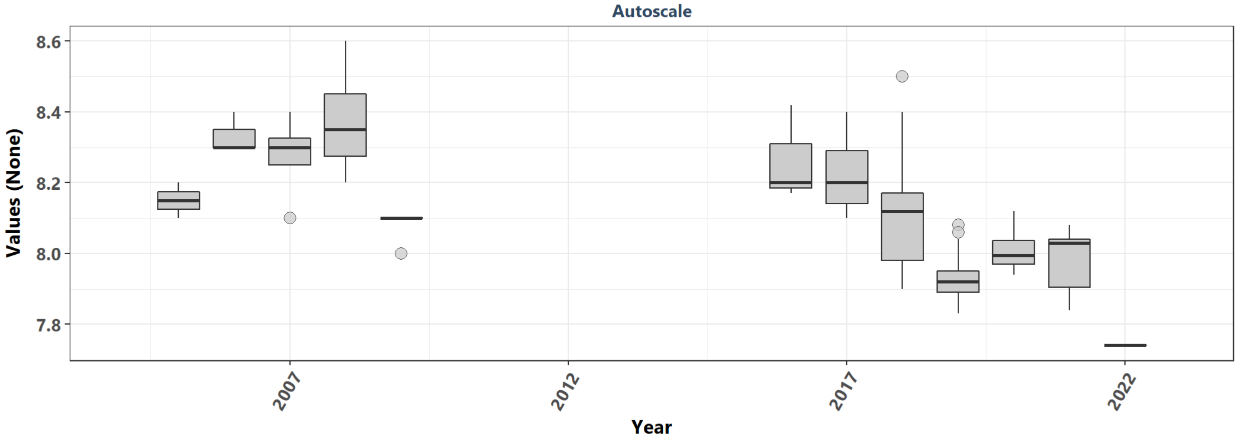
```r
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$SufficientData==TRUE &
                           data$ManagedAreaName==MA_Include[i],]
      year_lower <- min(plot_data$Year)
      year_upper <- max(plot_data$Year)
      min_RV <- 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
```

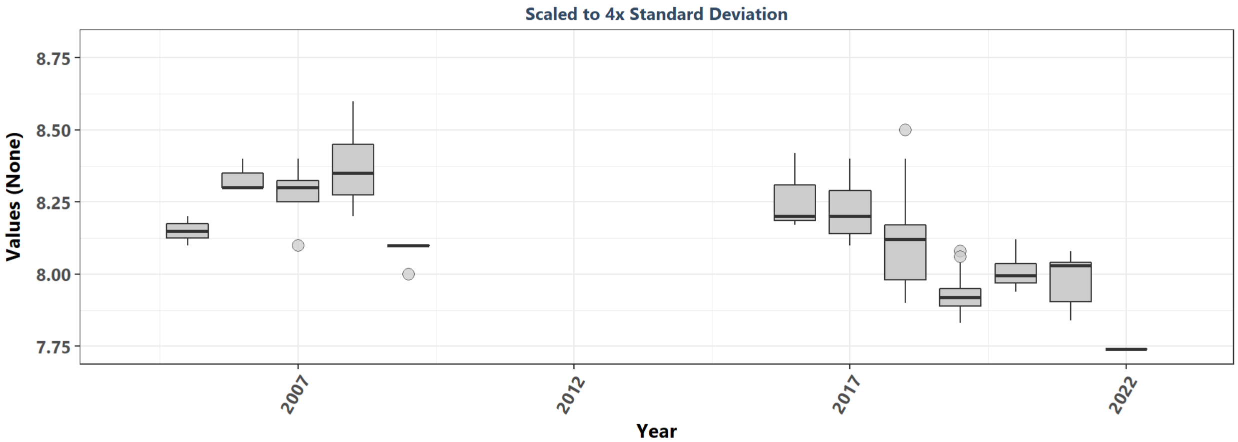
```r
p2 <- ggplot(data=plot_data,
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to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

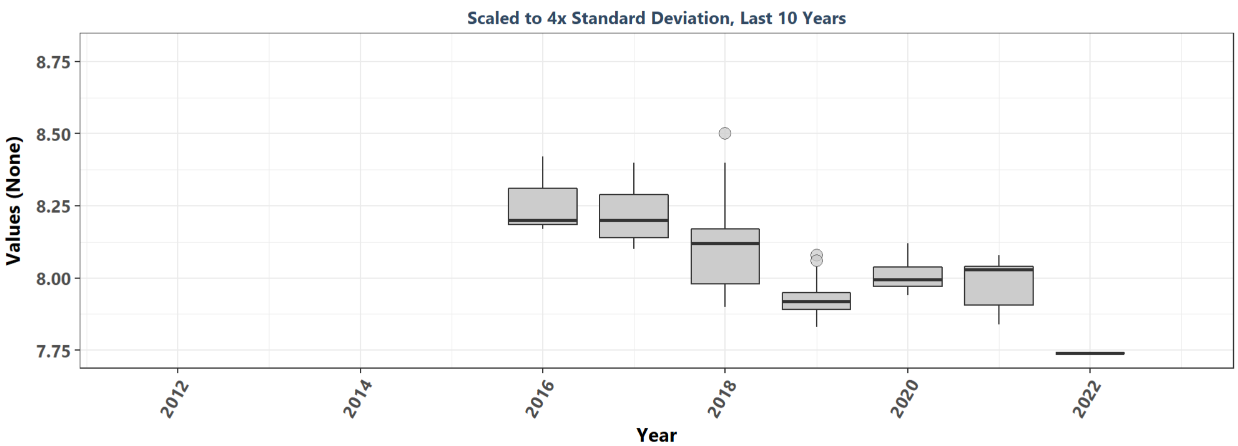
```r
    ylim(min_RV, y_scale) +
    scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
                       breaks=rev(seq(year_upper,
                                      year_lower, -x_scale))) +
    plot_theme +
    theme(legend.position="top", legend.box="horizontal") +
    guides(color=guide_legend(nrow=1))

  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

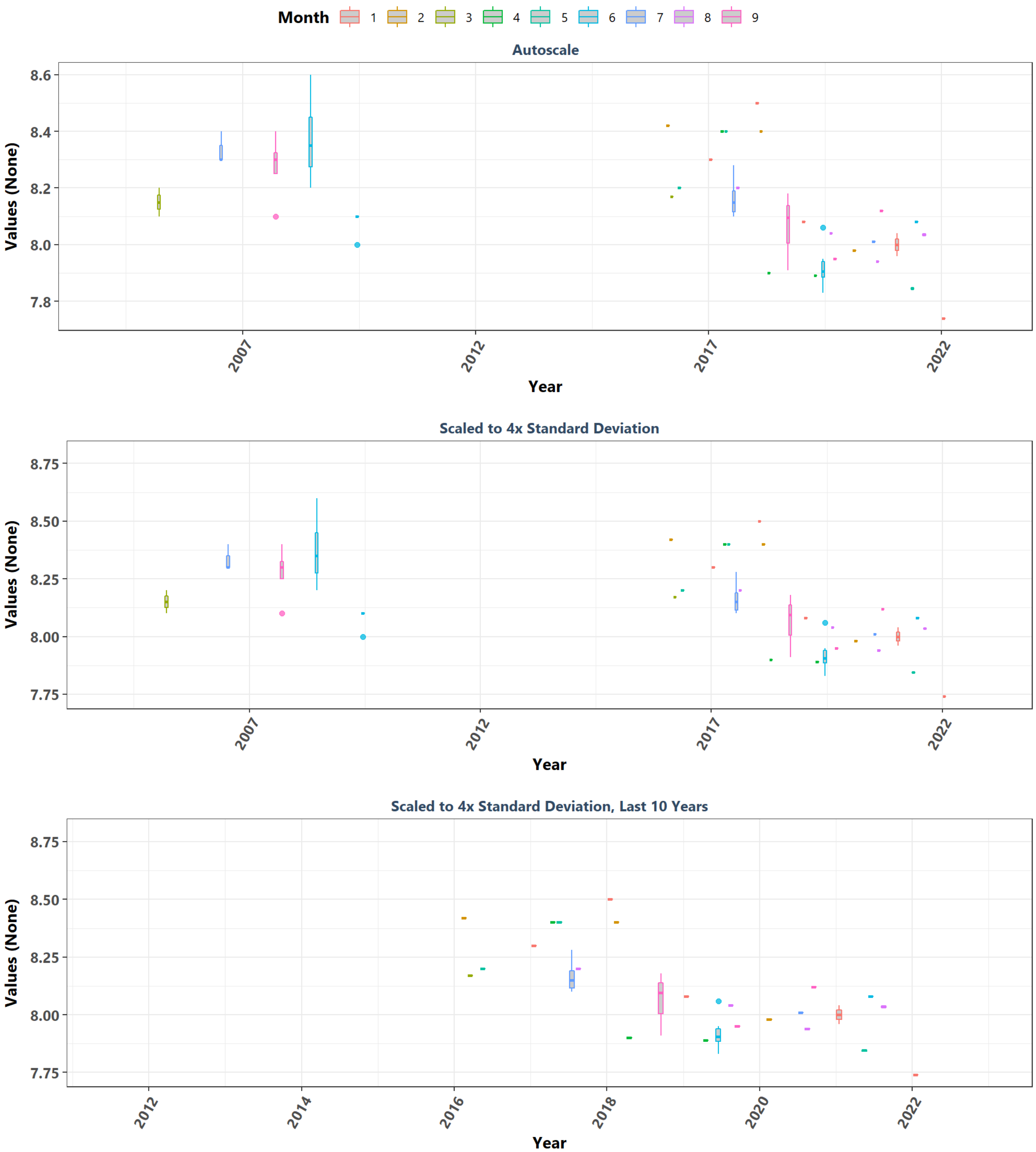


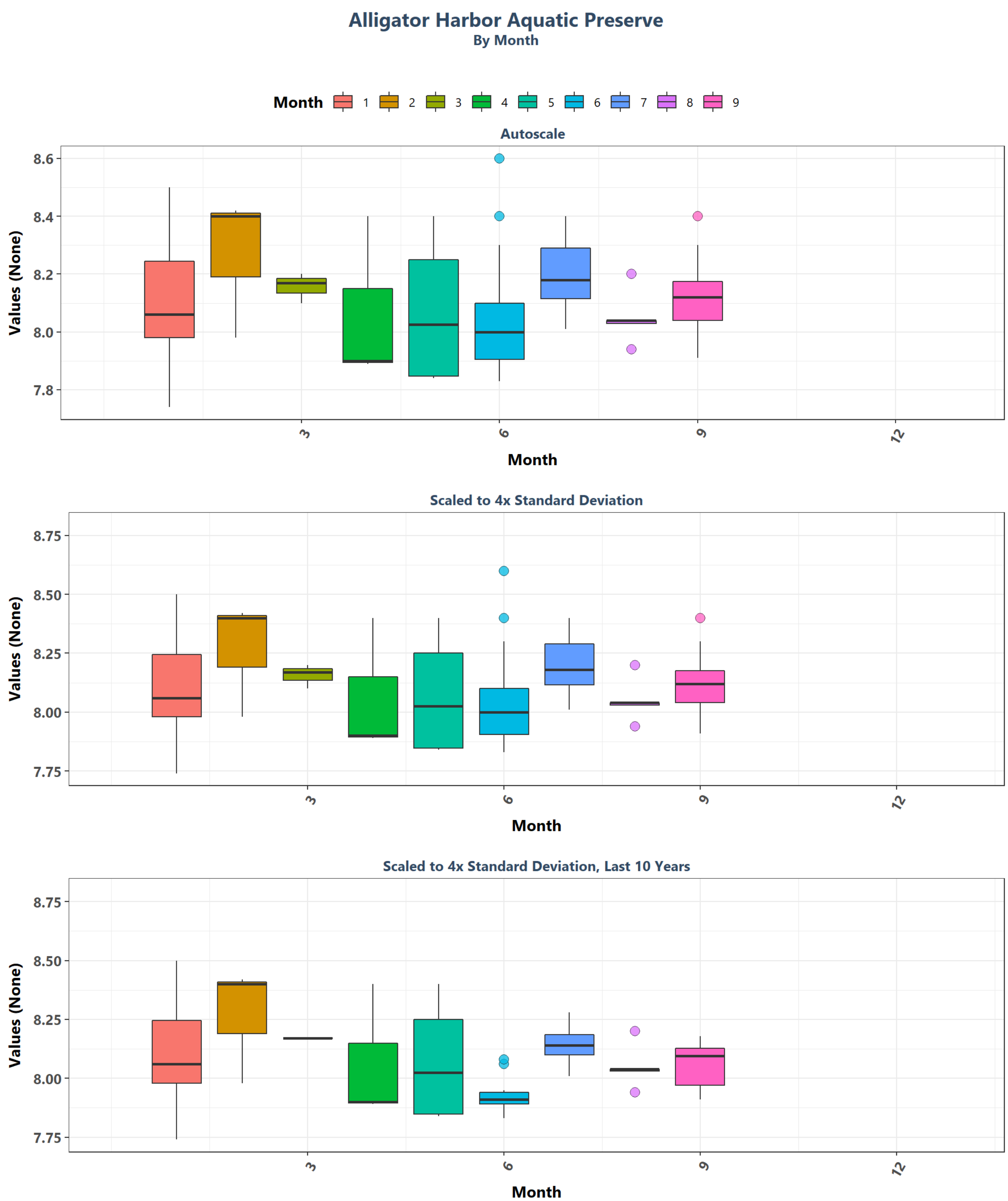




Alligator Harbor Aquatic Preserve

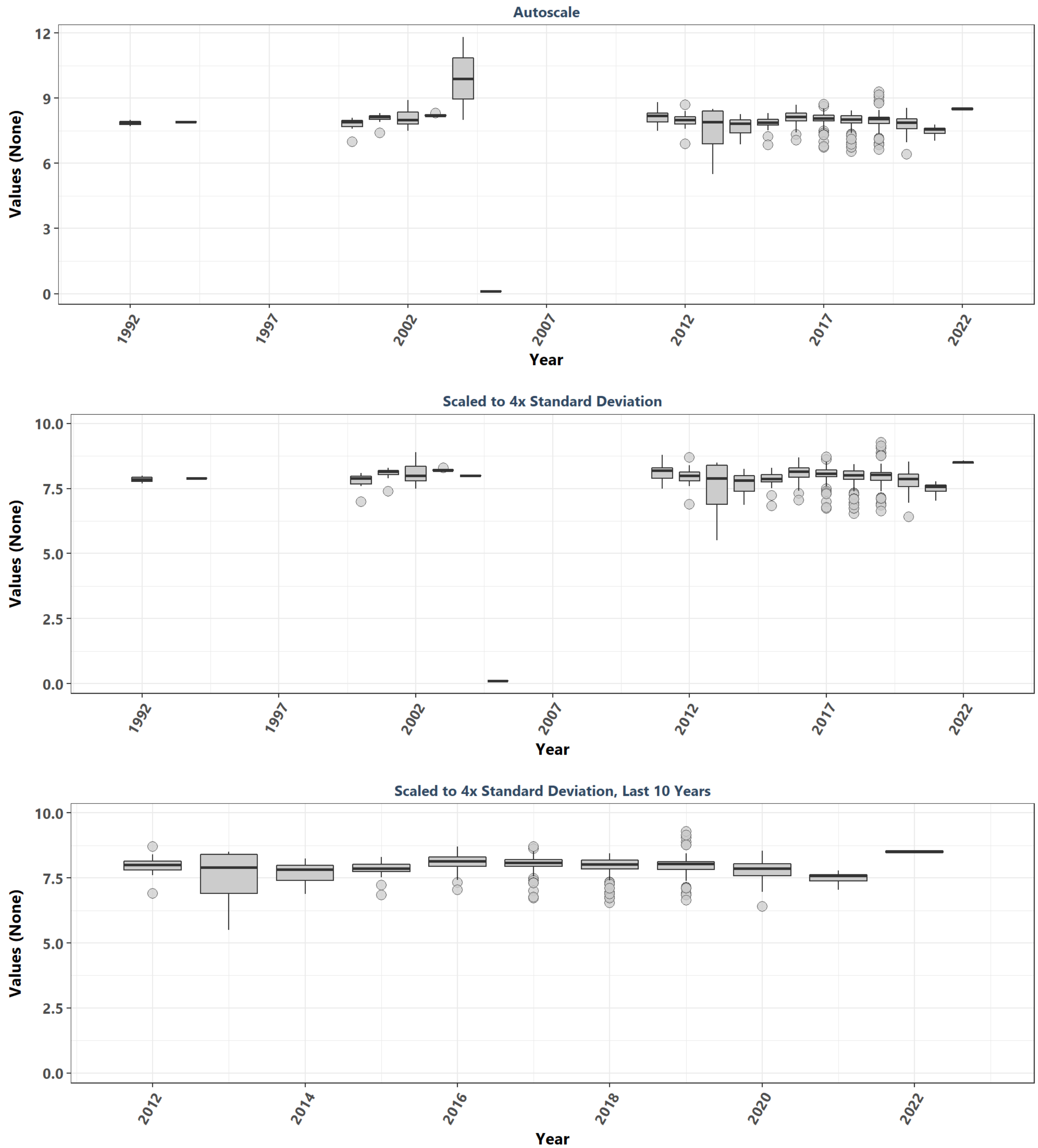
By Year & Month

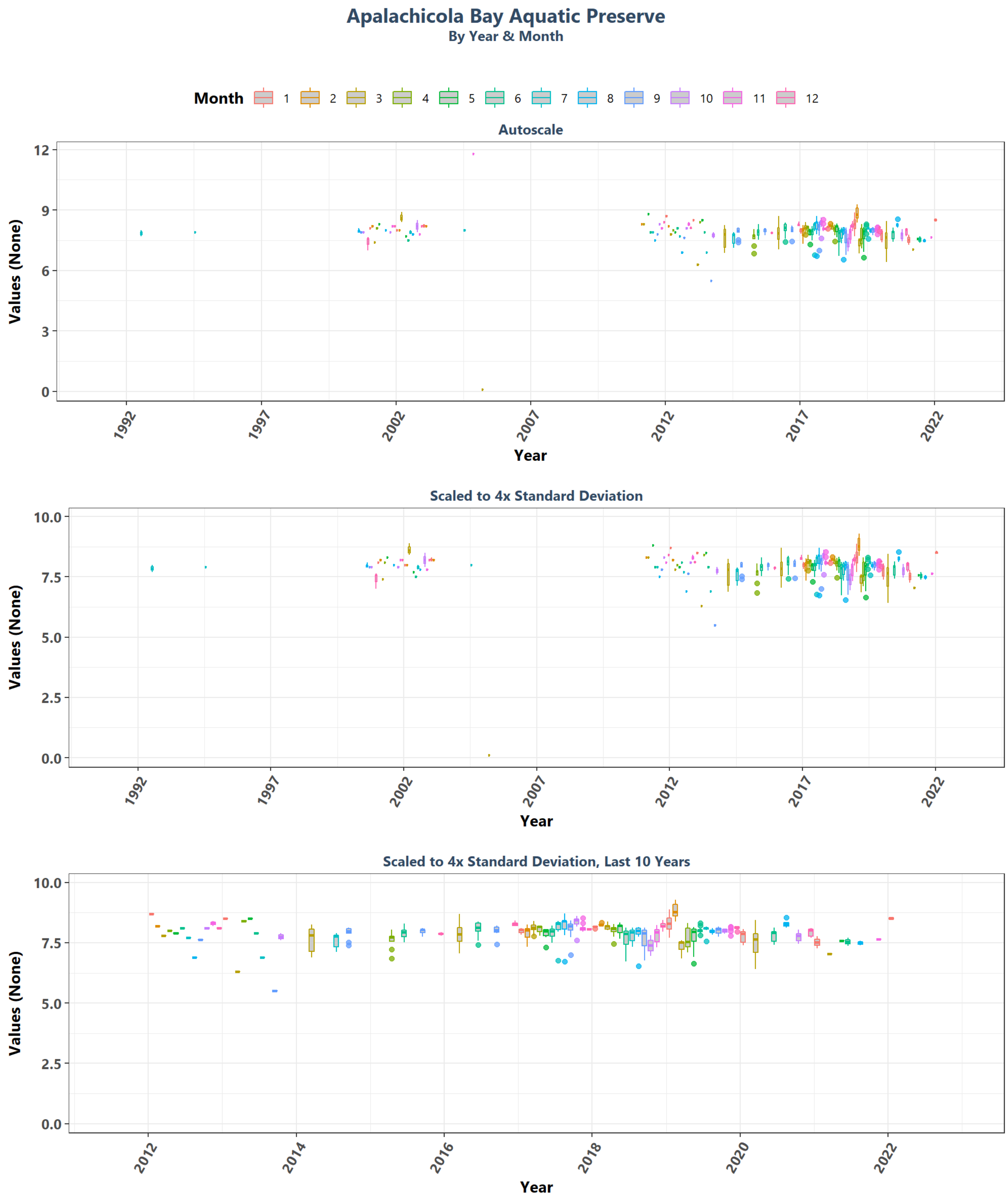




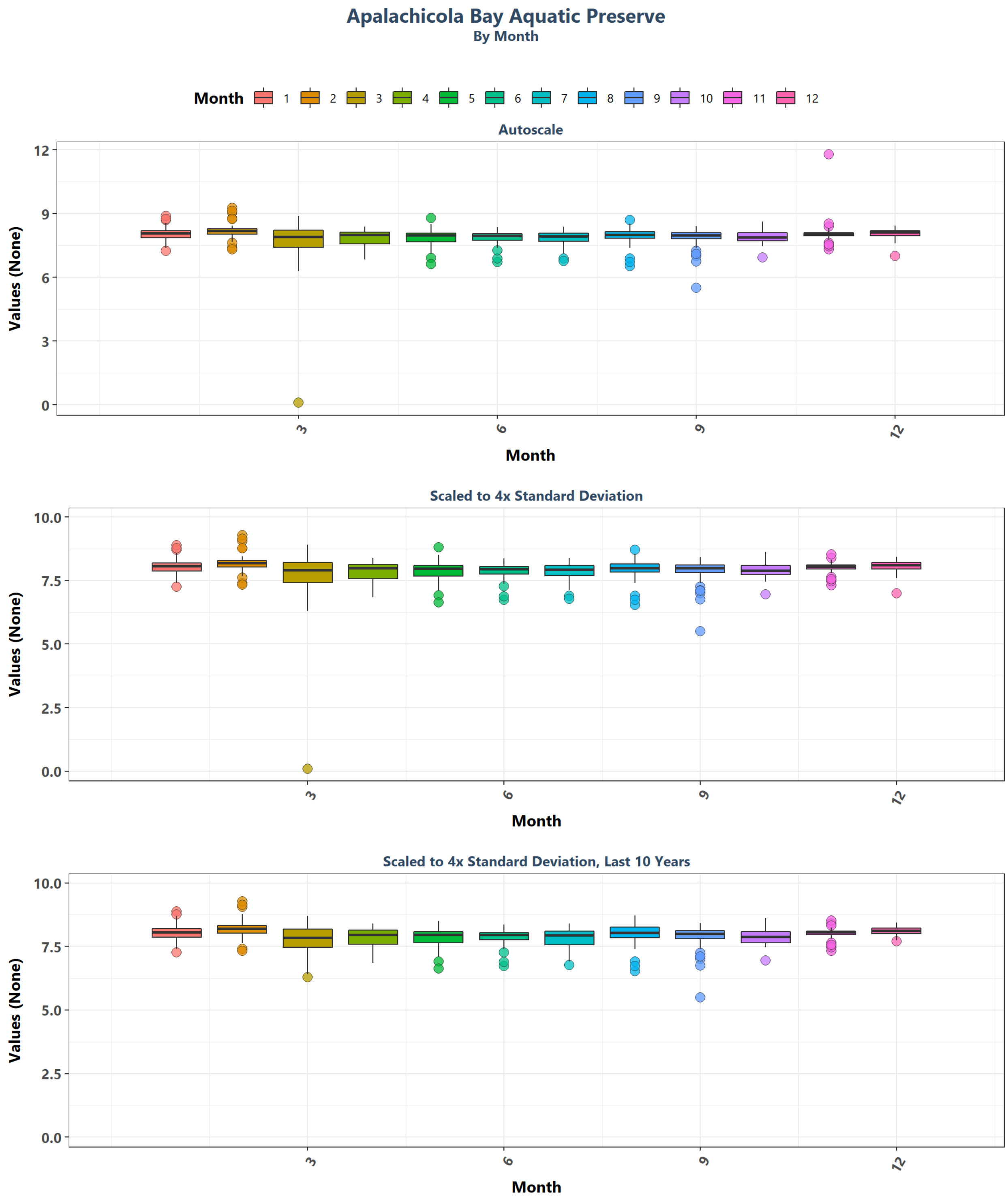
Alligator Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9
Autoscale
Values (None)
8.6
8.4
8.2
8.0
7.8
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
8.75
8.50
8.25
8.00
7.75
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8.75
8.50
8.25
8.00
7.75
3
6
9
12
Month

Apalachicola Bay Aquatic Preserve
By Year



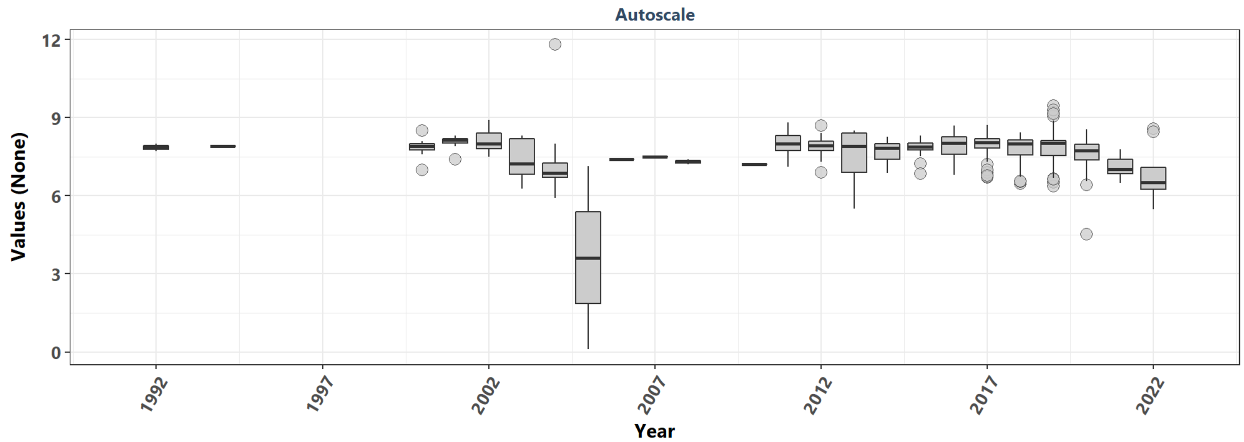


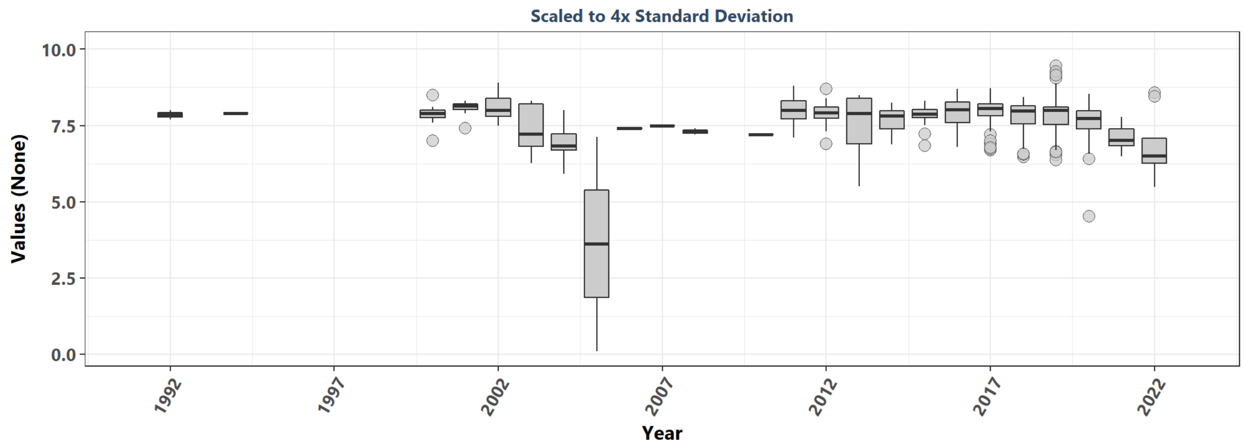
Apalachicola Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
12
9
6
3
0
1992 1997 2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation
Values (None)
10.0
7.5
5.0
2.5
0.0
1992 1997 2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
10.0
7.5
5.0
2.5
0.0
2012 2014 2016 2018 2020 2022
Year

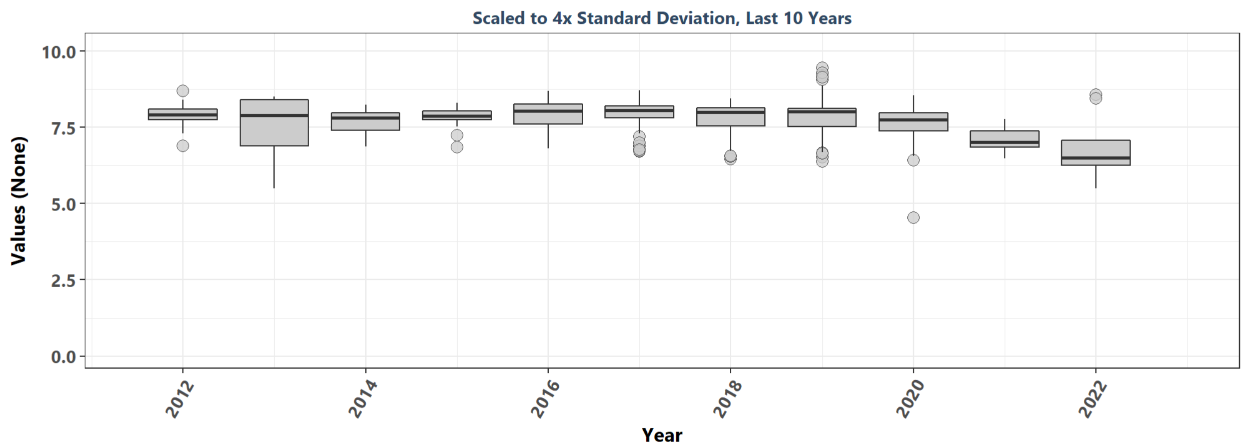

Apalachicola Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
12
9
6
3
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month

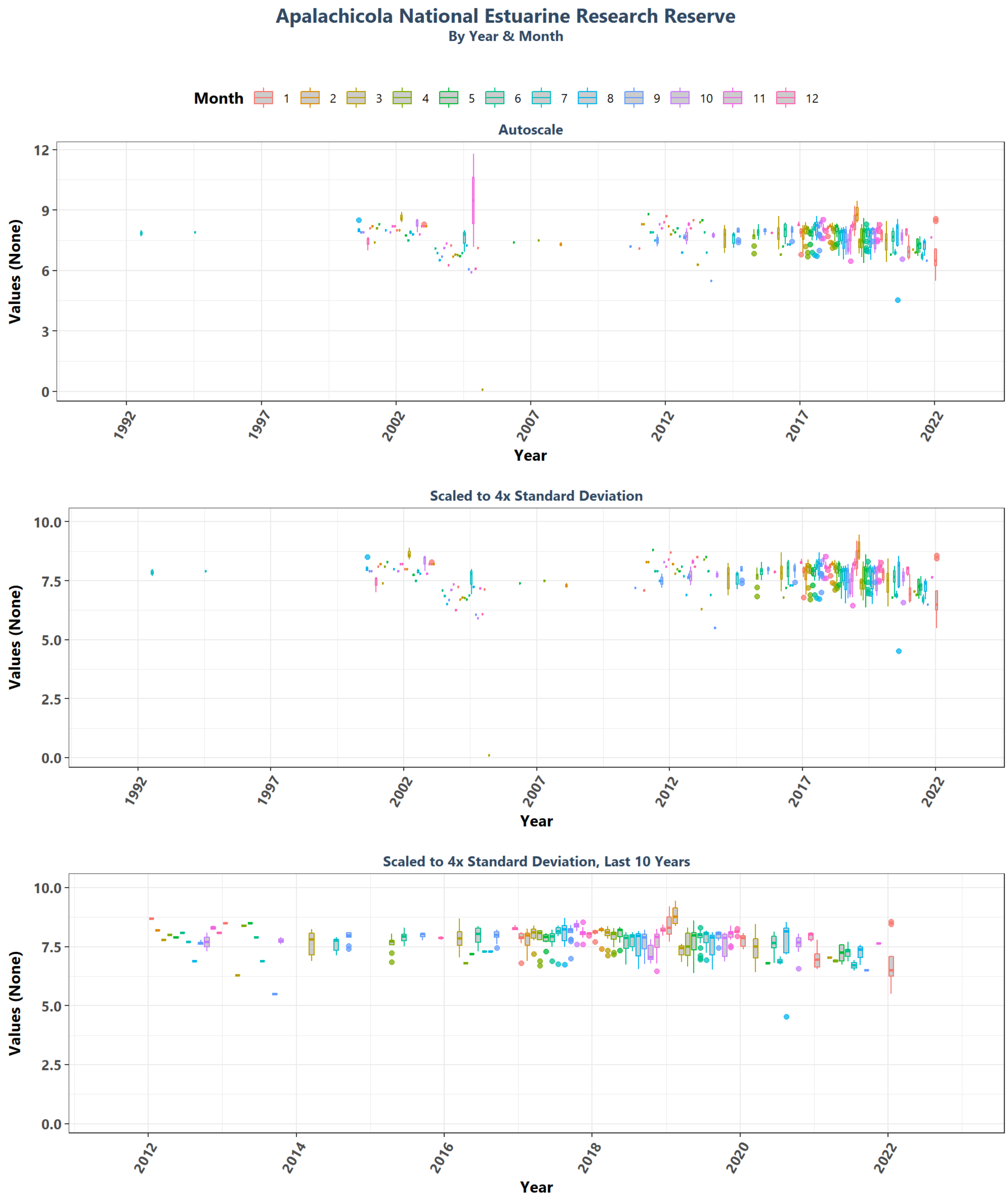
Apalachicola National Estuarine Research Reserve

By Year

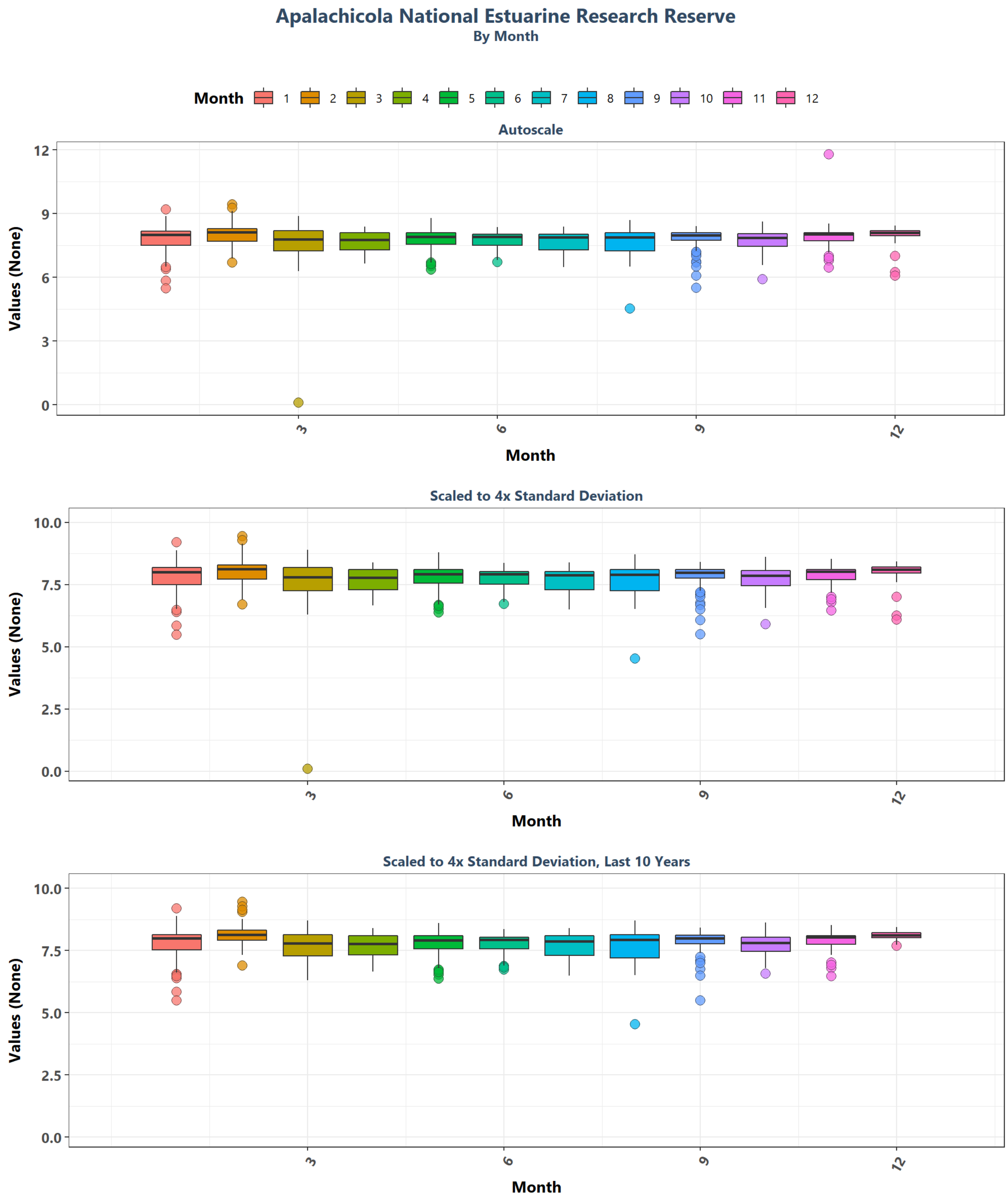




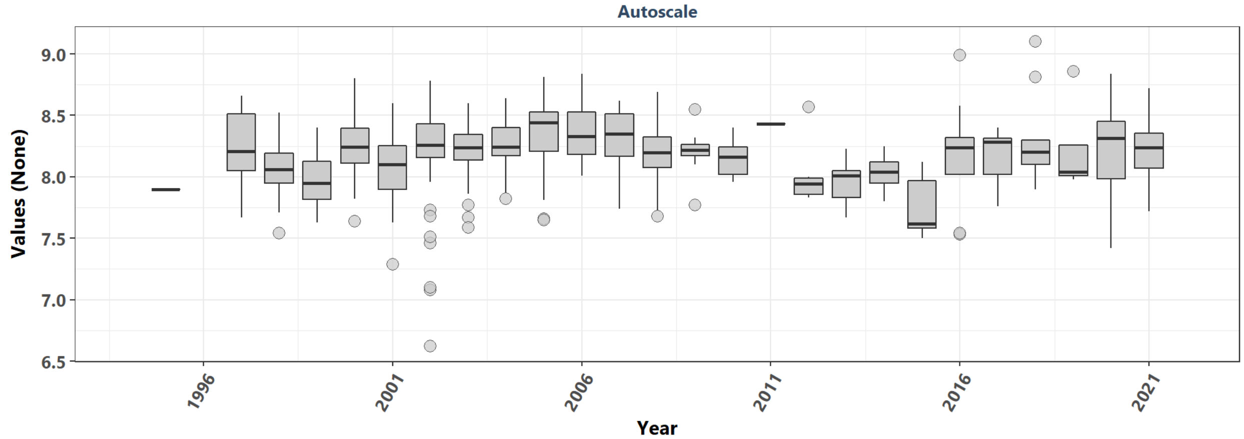


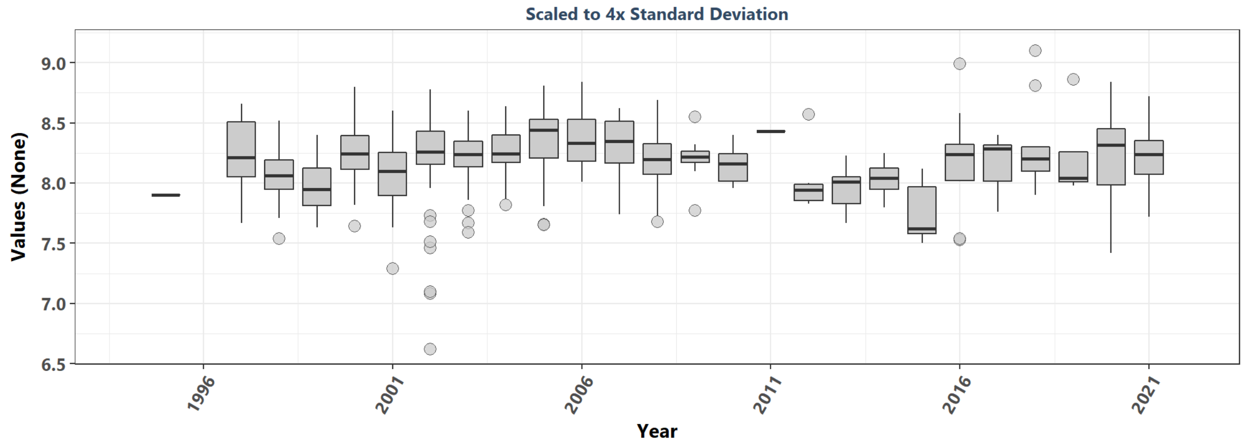


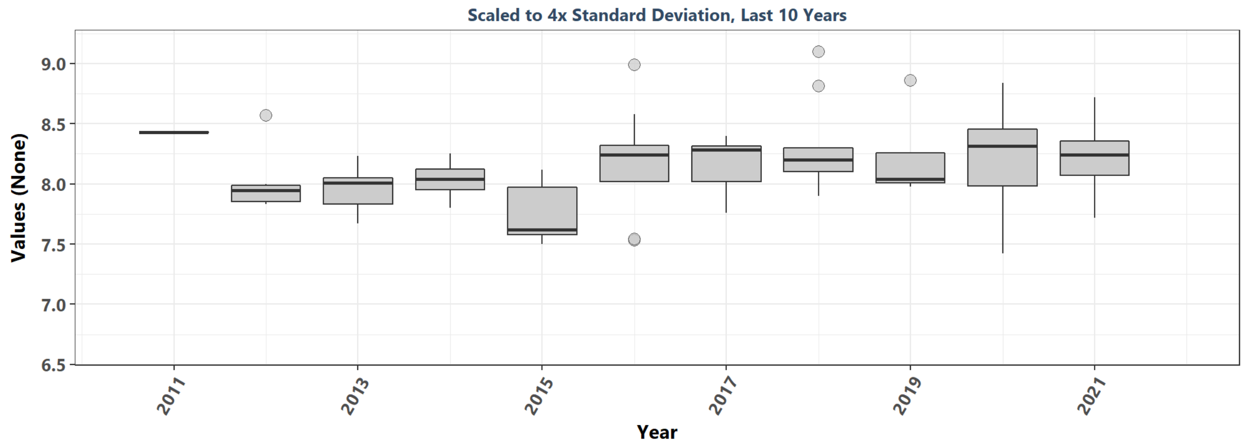
Apalachicola National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
Year
Scaled to 4x Standard Deviation
Values (None)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year

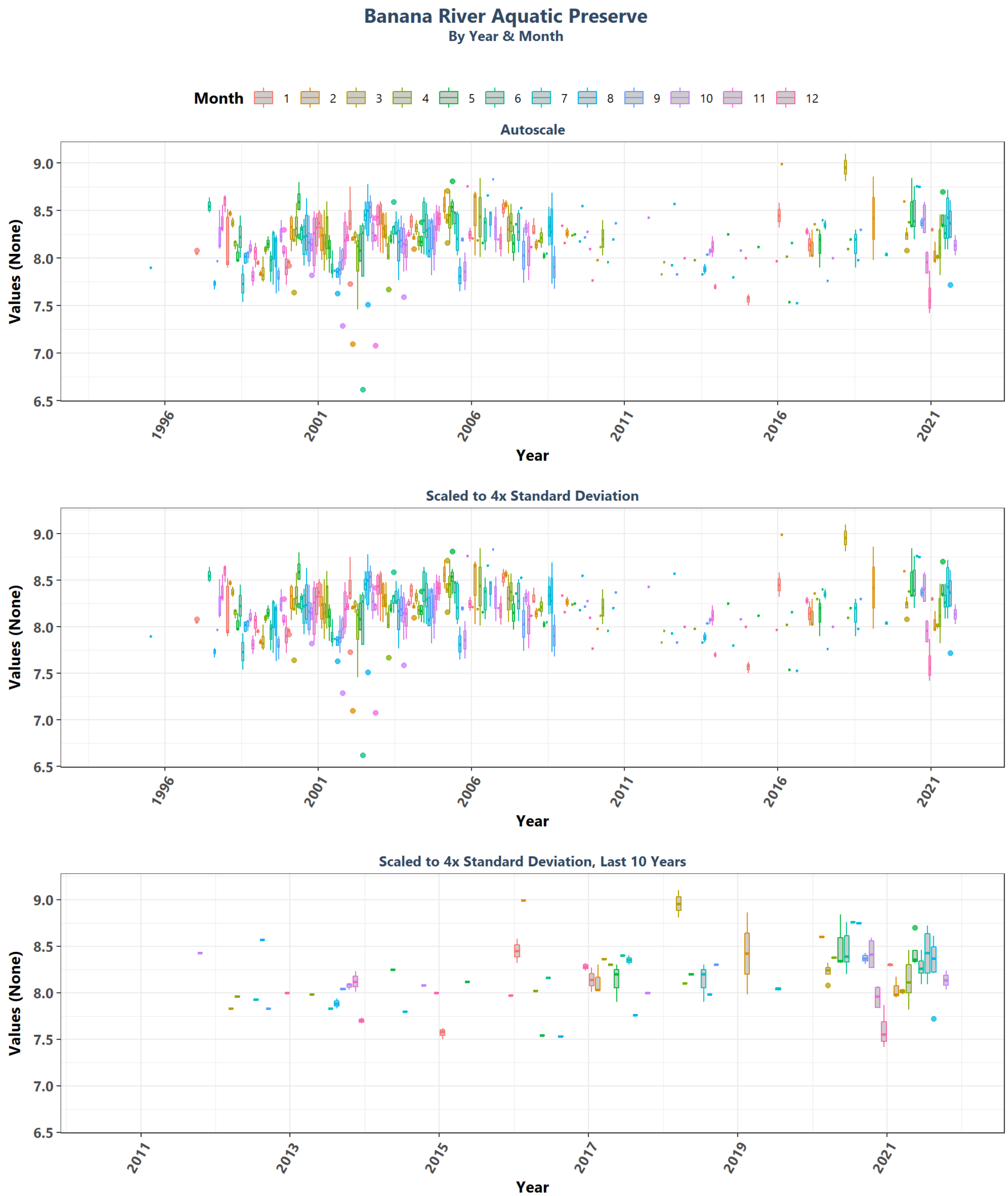

Apalachicola National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
12
9
6
3
0
Month
3
6
9
12
Scaled to 4x Standard Deviation
10.0
7.5
5.0
2.5
0.0
Scaled to 4x Standard Deviation, Last 10 Years

Banana River Aquatic Preserve
By Year

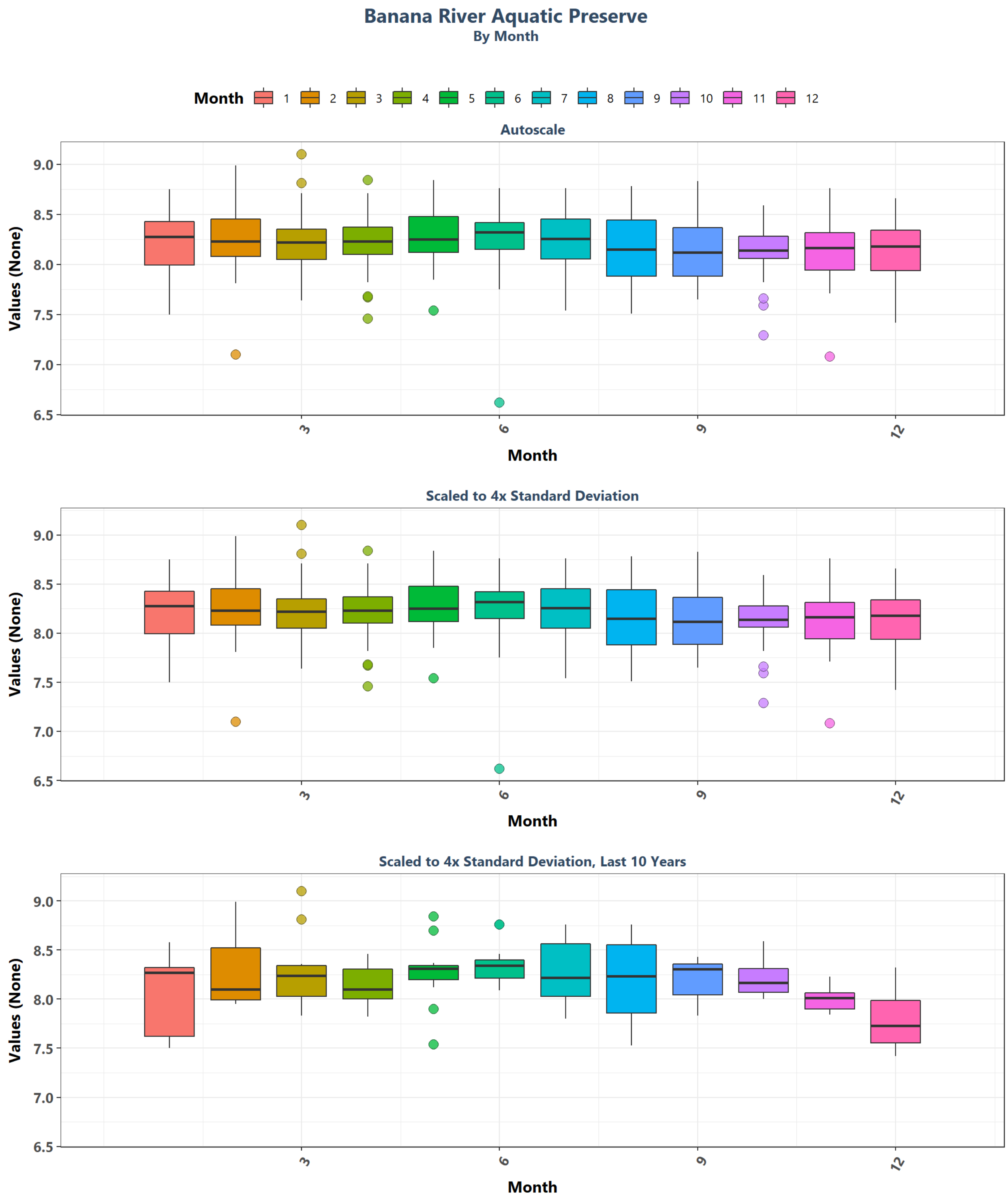




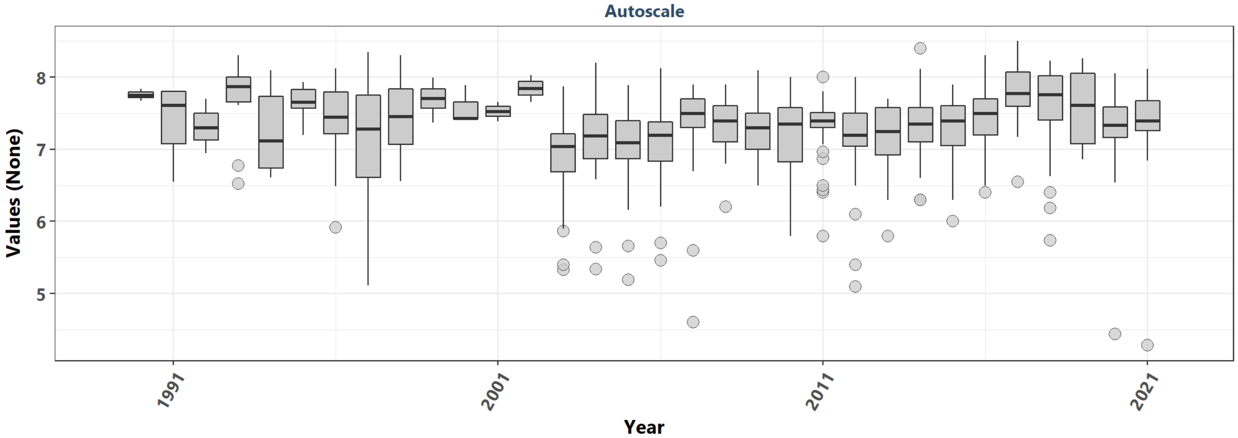


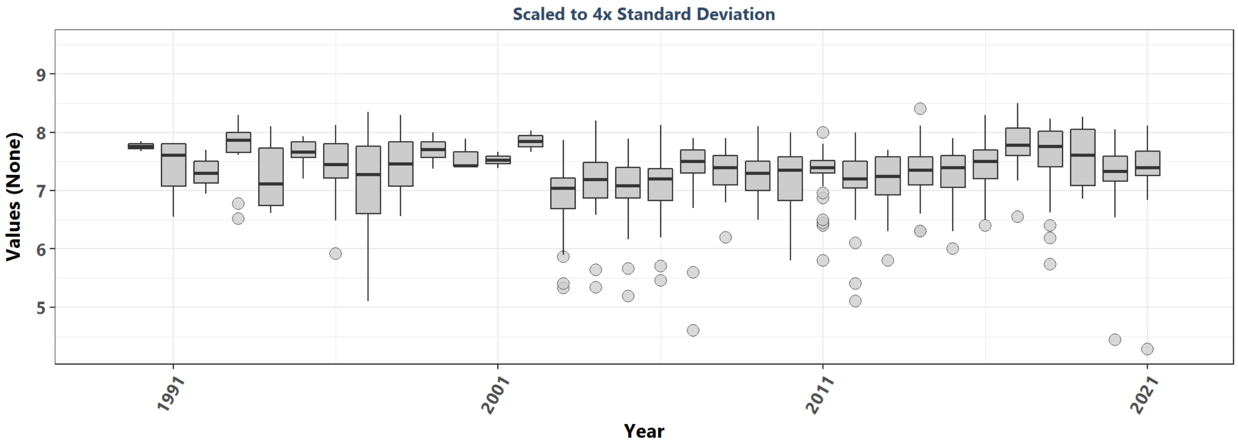


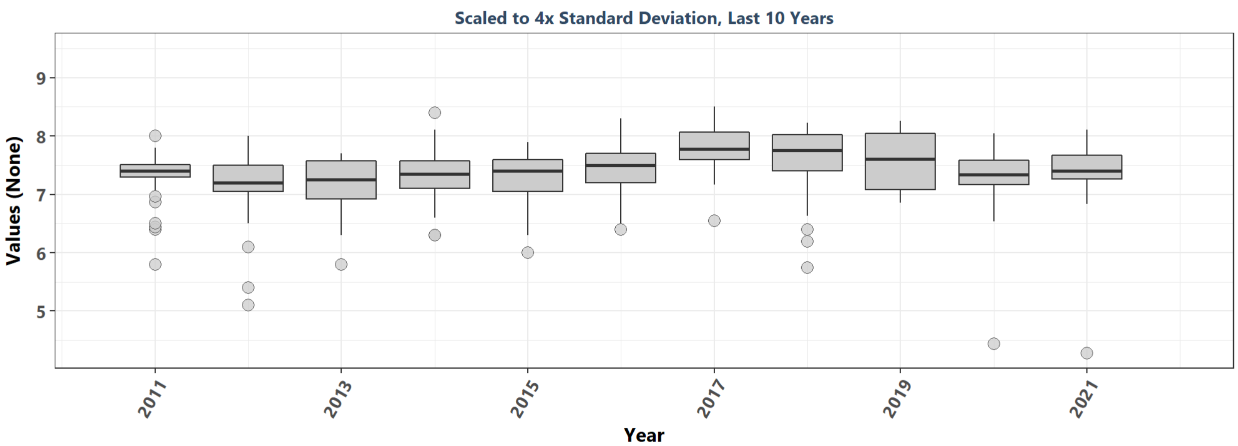
Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
2011
2013
2015
2017
2019
2021
Year

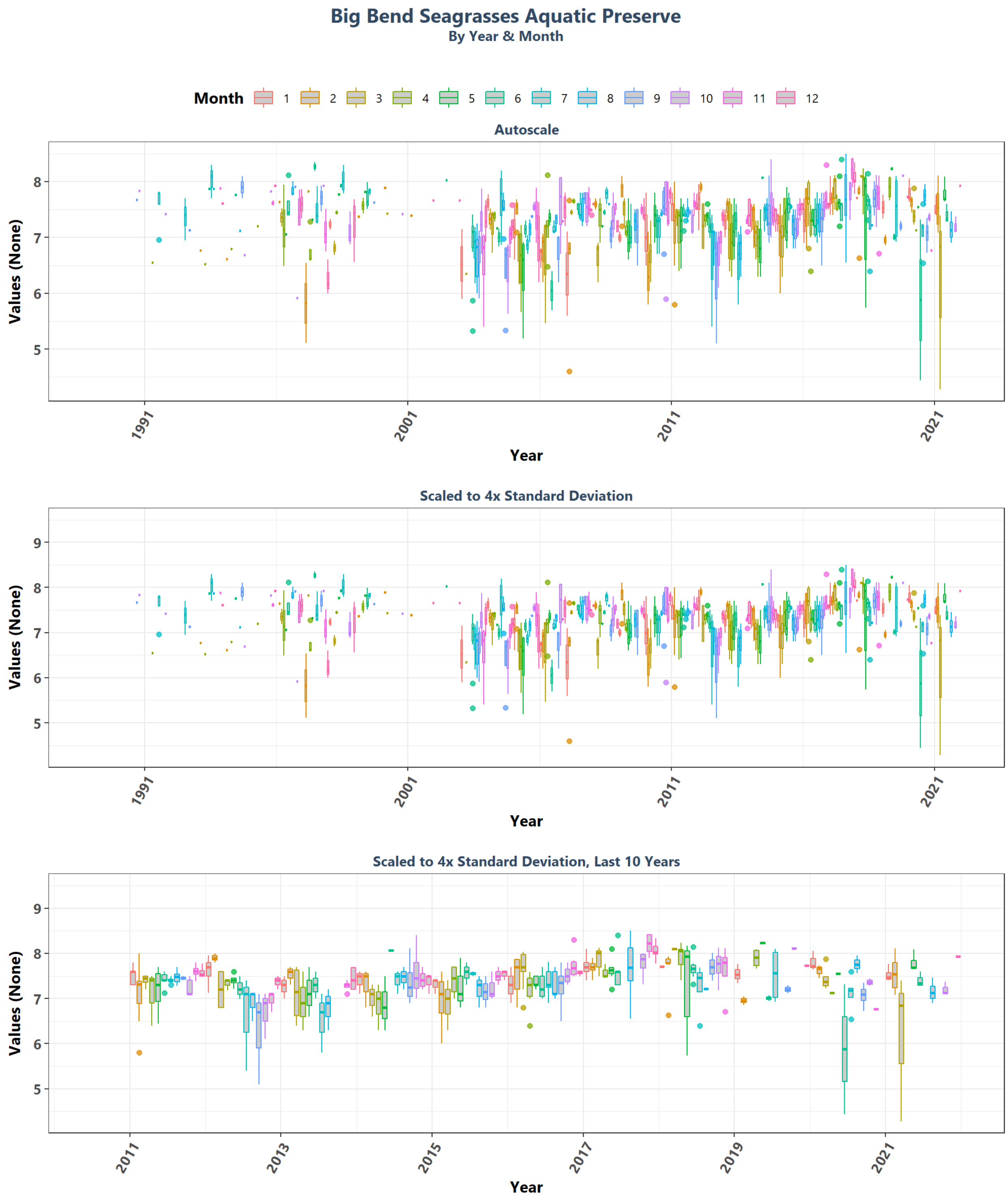

Banana River Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years

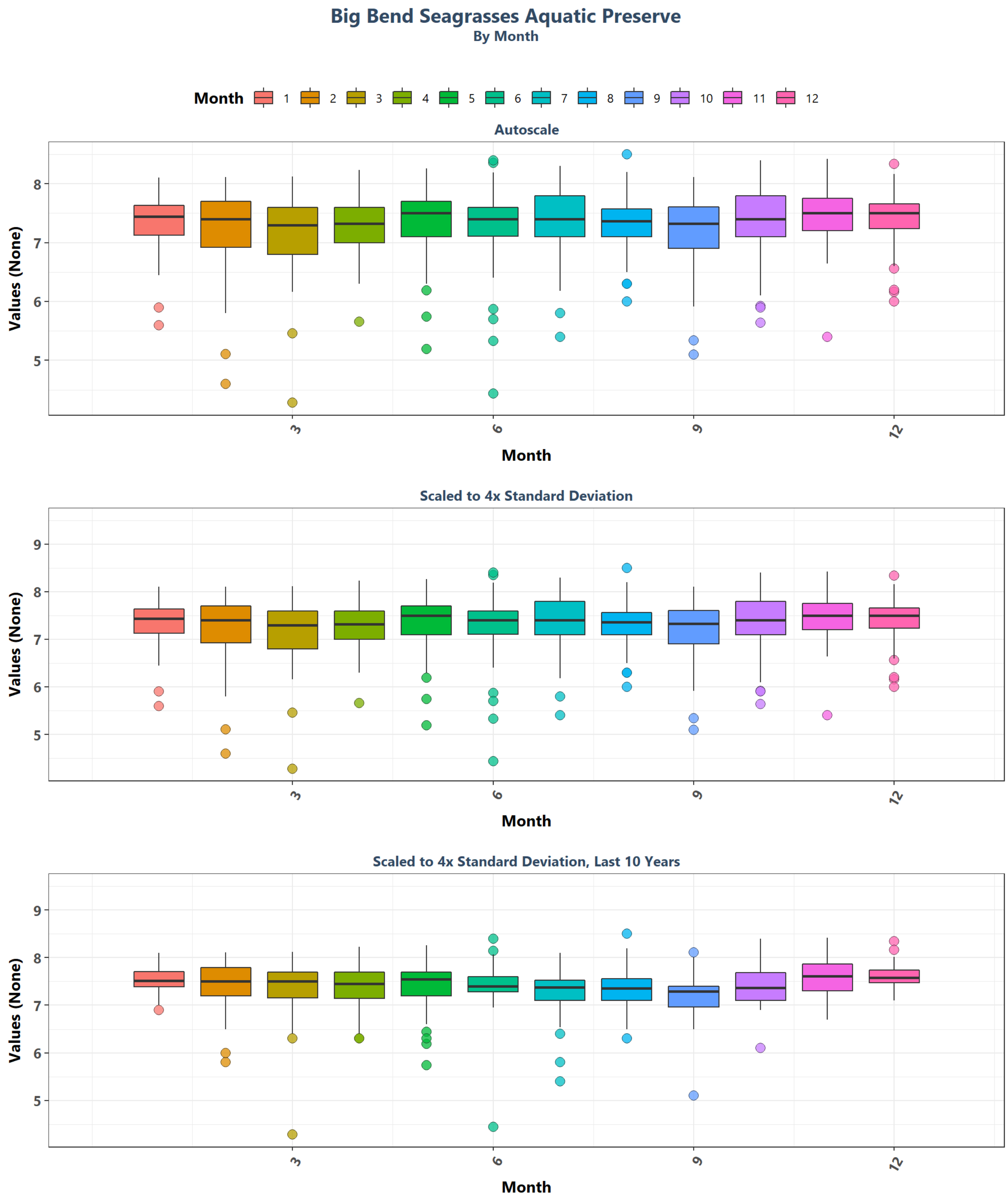
Big Bend Seagrasses Aquatic Preserve
By Year







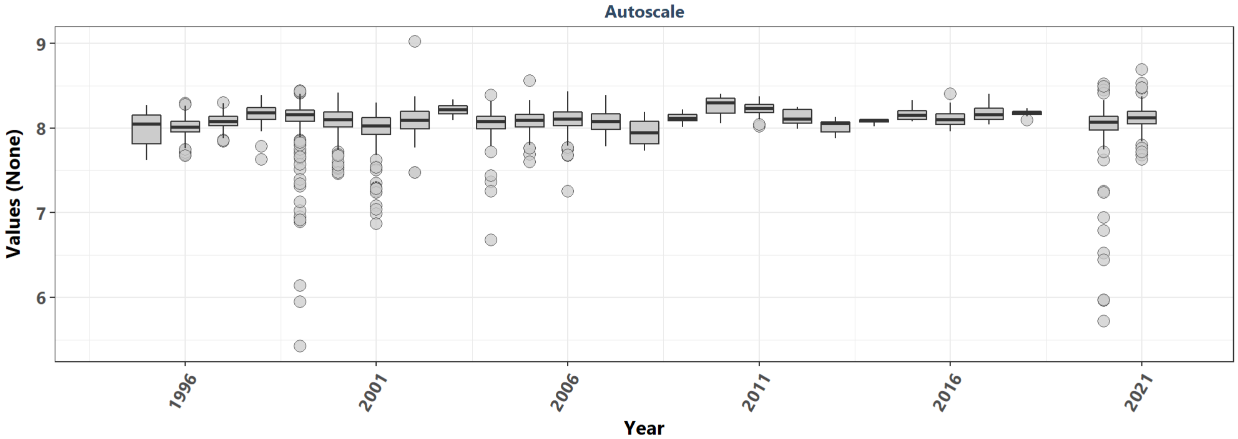

Big Bend Seagrasses Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
Year
1991
2001
2011
2021
Scaled to 4x Standard Deviation
Values (None)
Year
1991
2001
2011
2021
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year
2011
2013
2015
2017
2019
2021

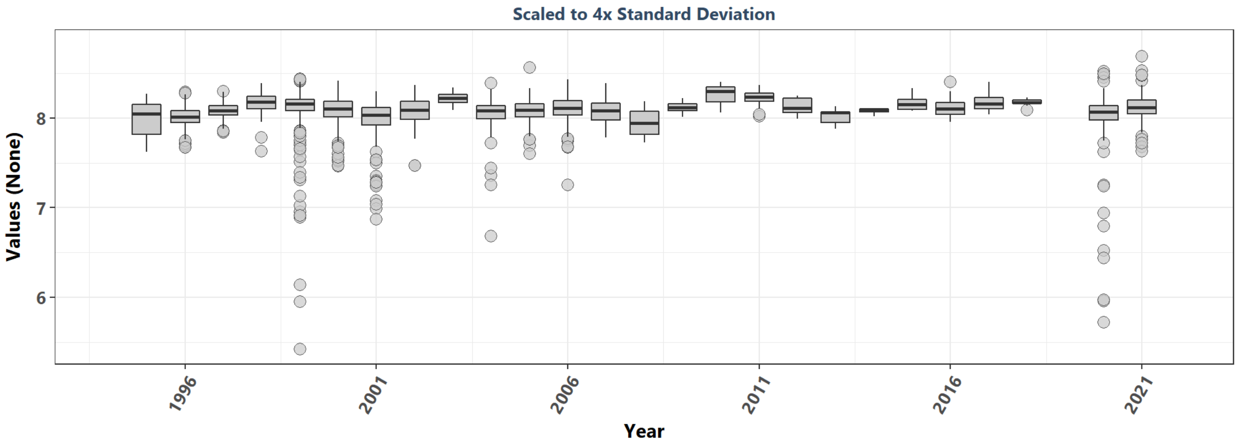


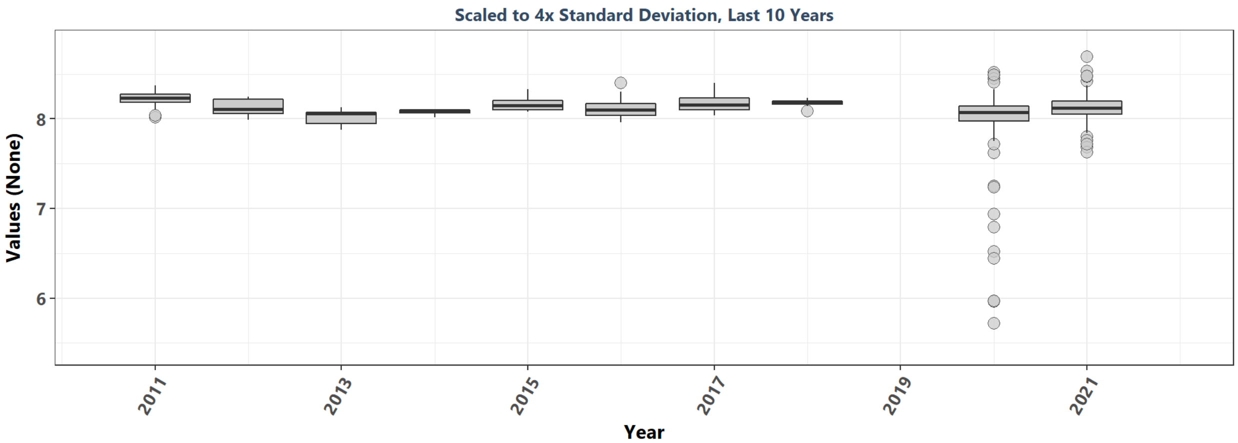
Big Bend Seagrasses Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
6
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
5
3
6
9
12
Month

Boca Ciega Bay Aquatic Preserve

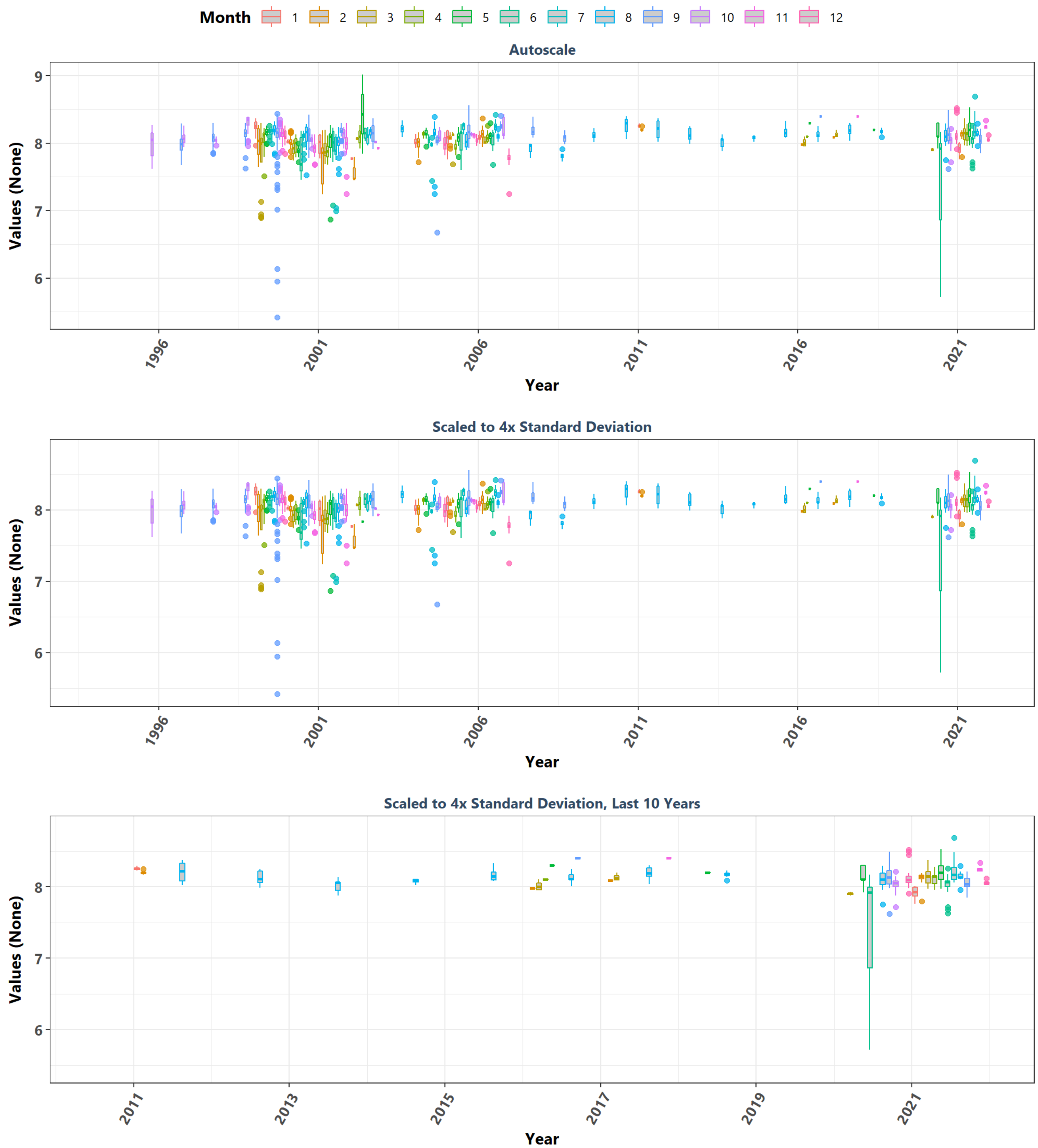
By Year

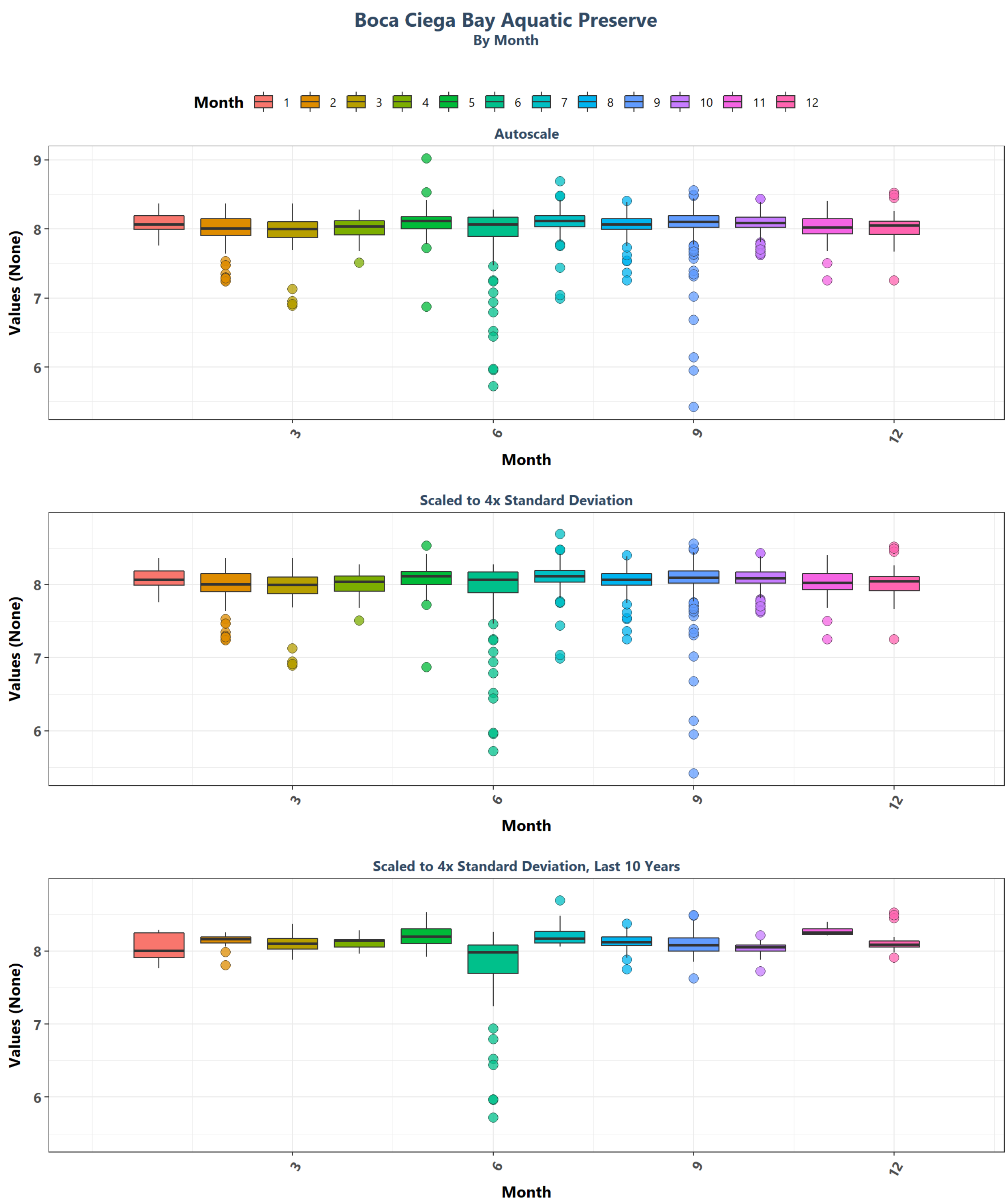






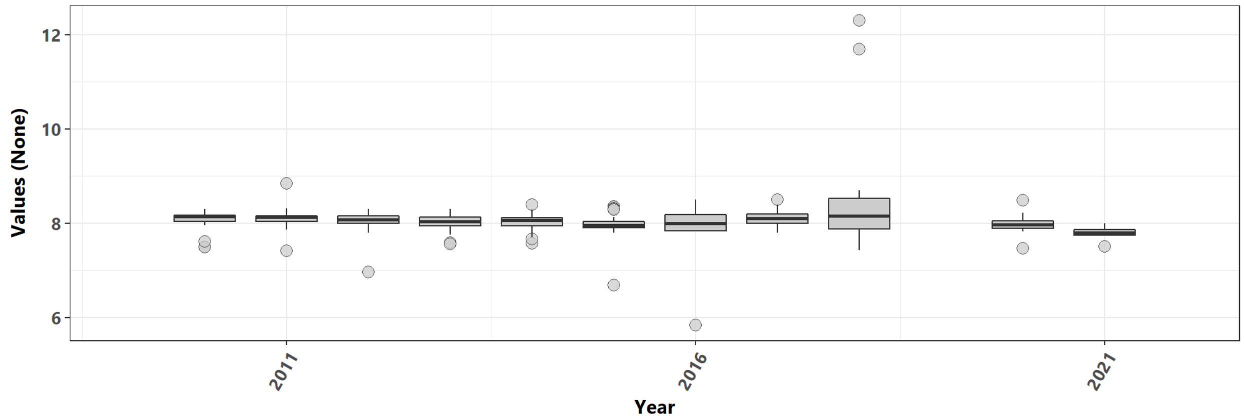
Boca Ciega Bay Aquatic Preserve
By Year & Month



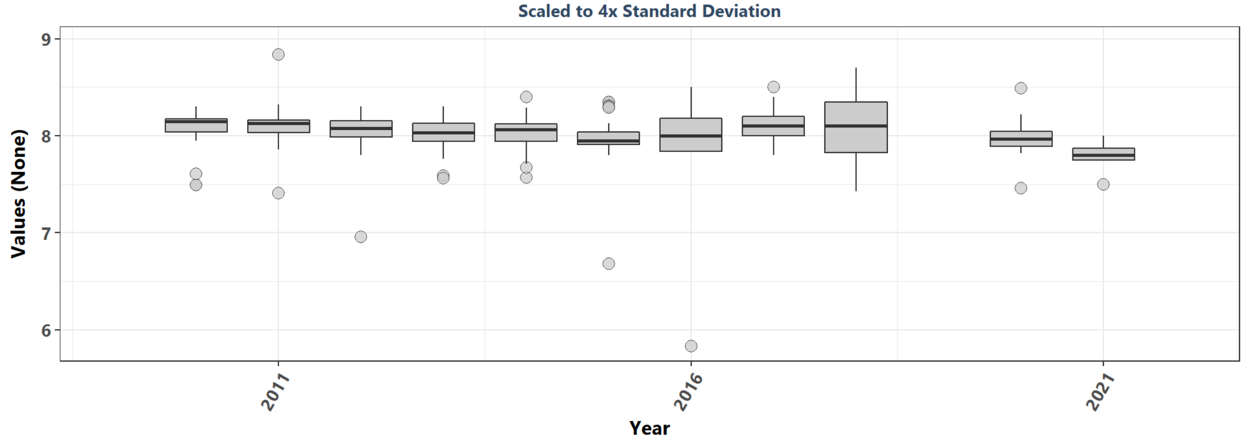

Boca Ciega Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
9
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8
7
6
3
6
9
12
Month

Cape Haze Aquatic Preserve
By Year

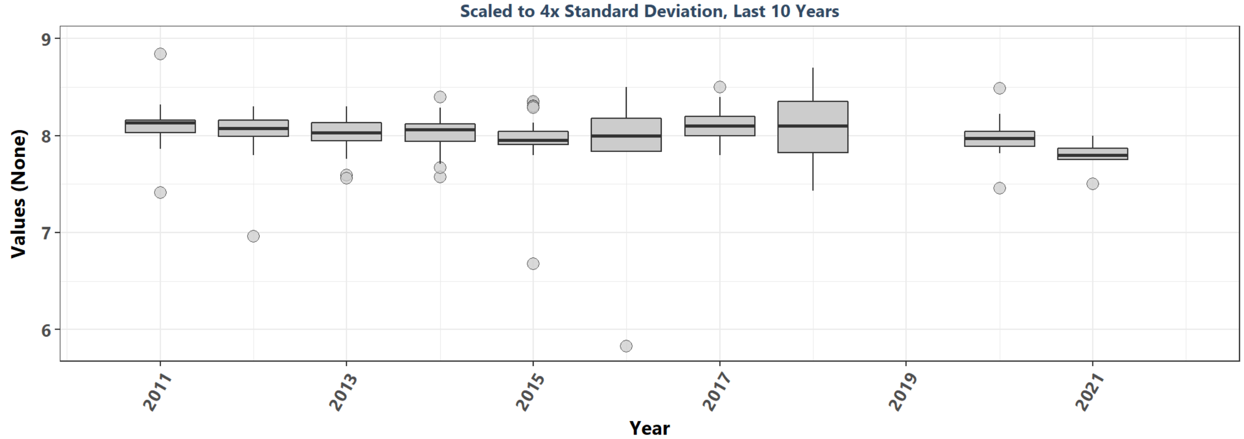
Autoscale



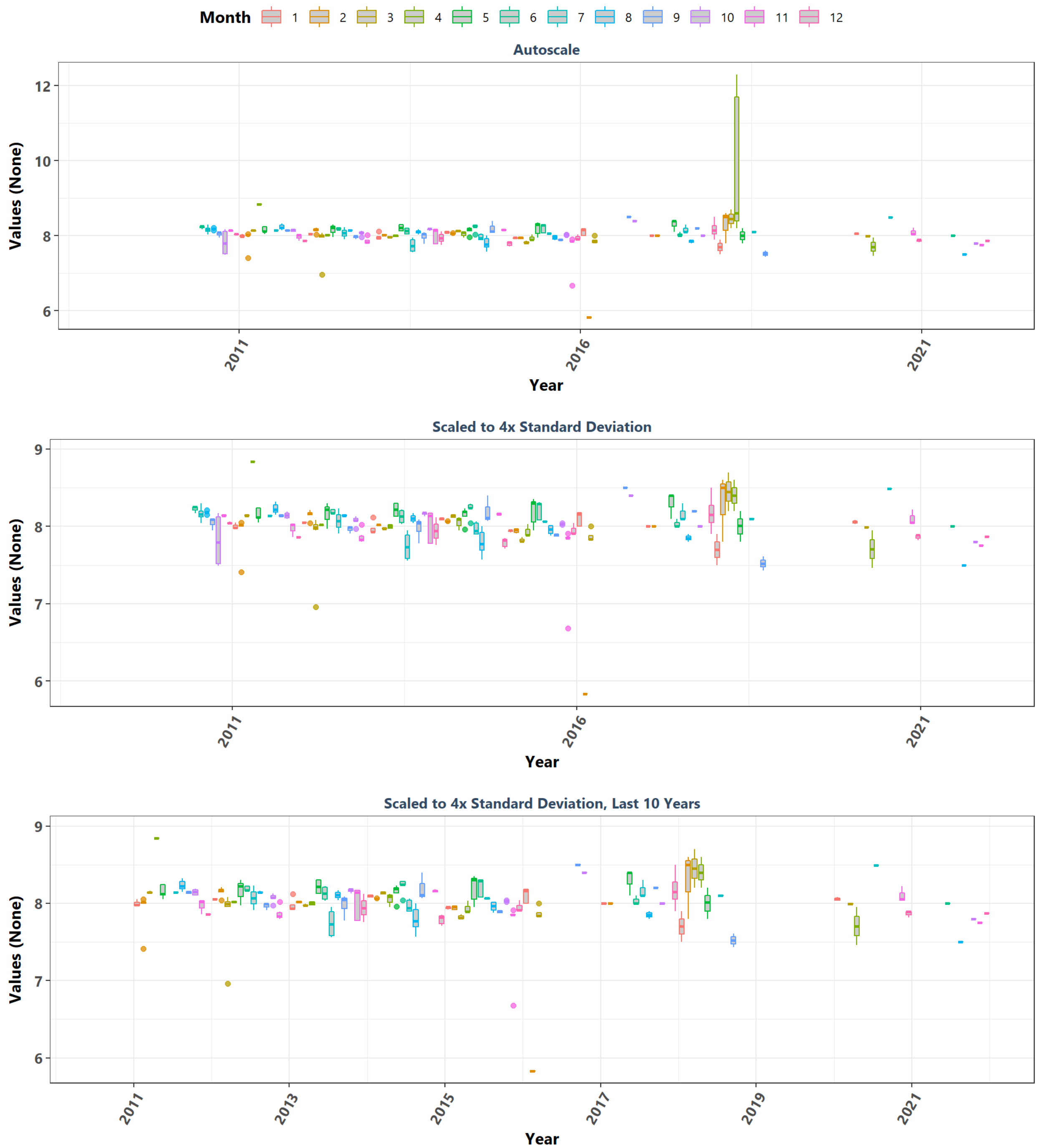
Scaled to 4x Standard Deviation



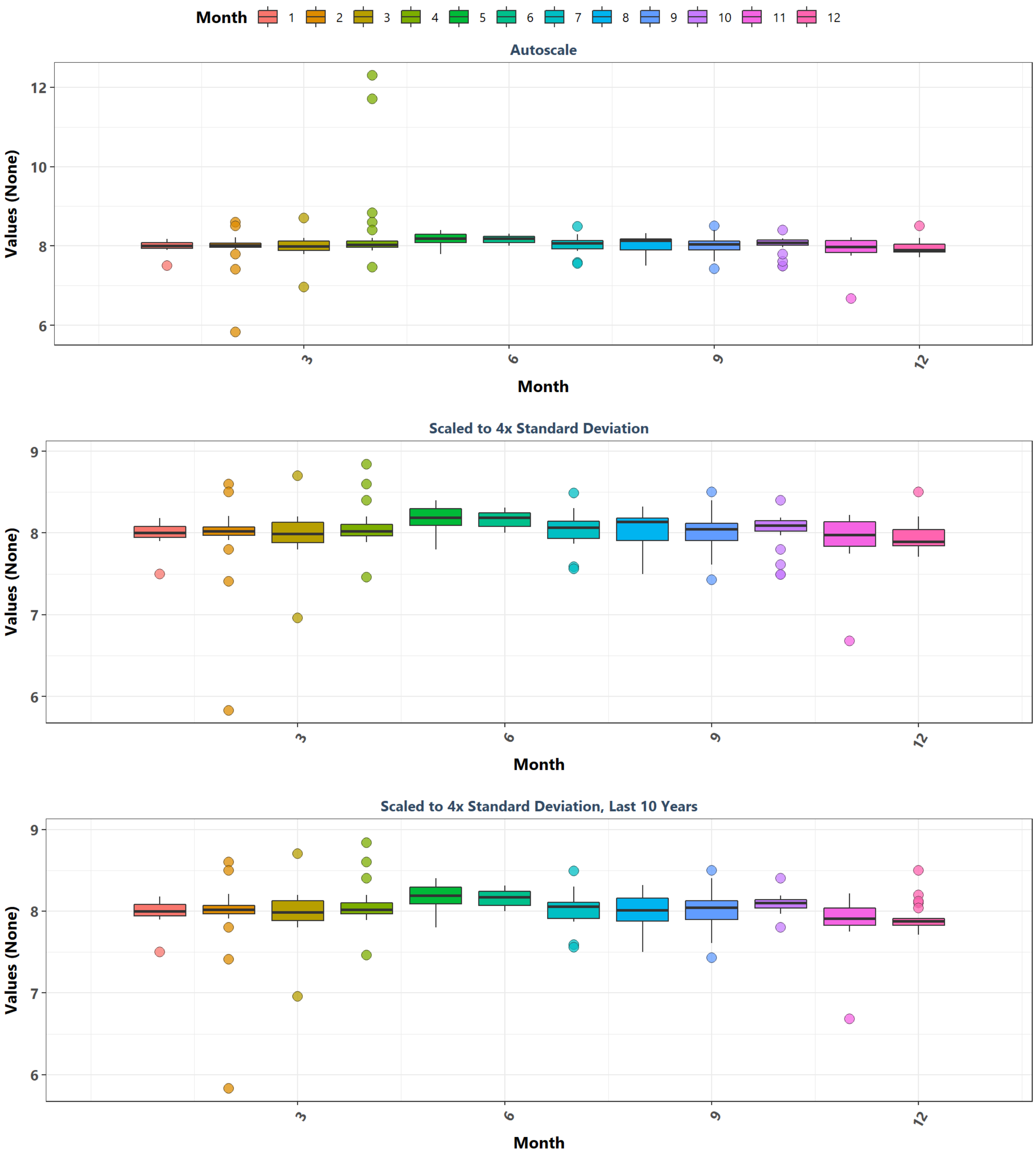
Scaled to 4x Standard Deviation, Last 10 Years

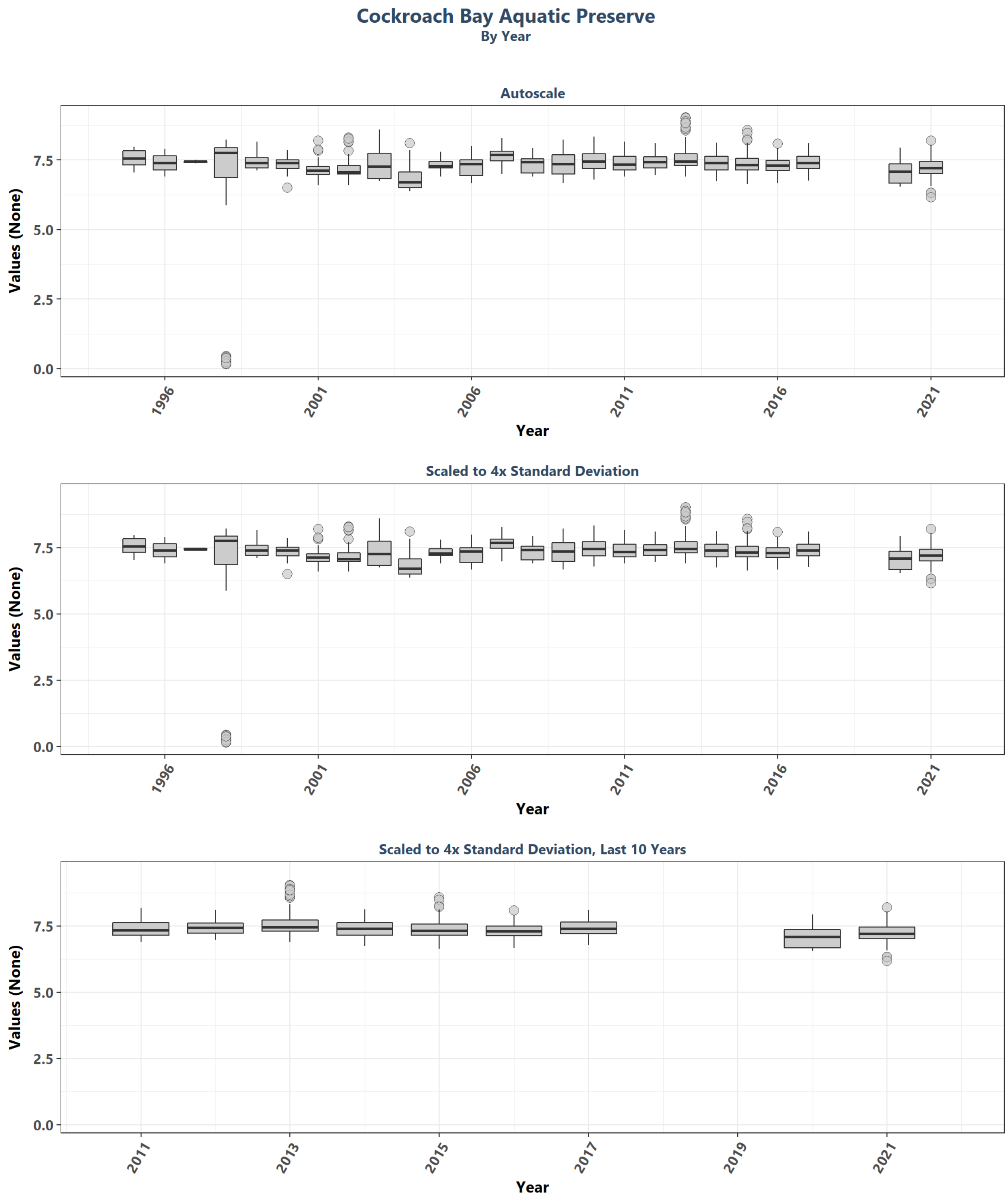


Cape Haze Aquatic Preserve
By Year & Month

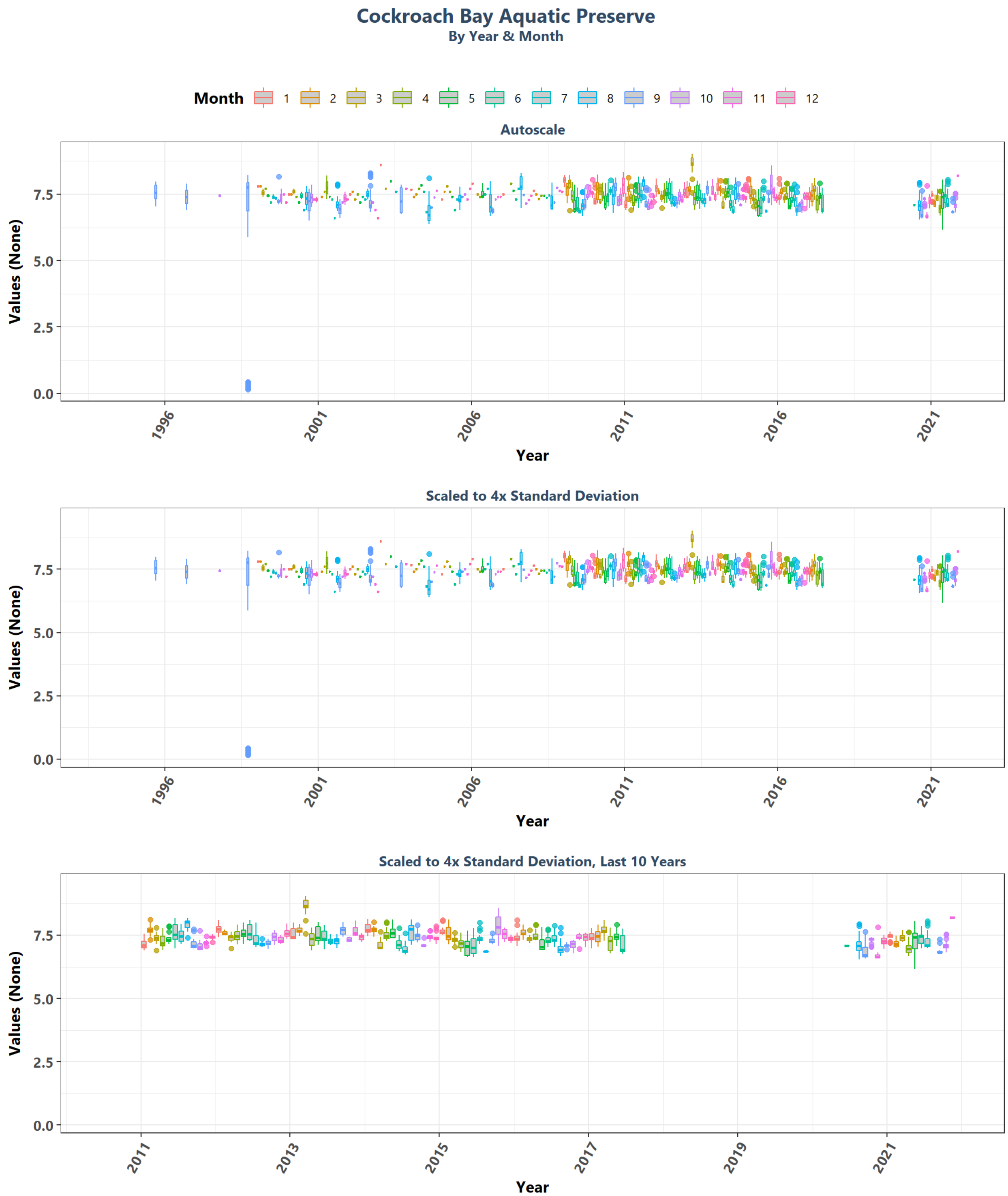


Cape Haze Aquatic Preserve
By Month

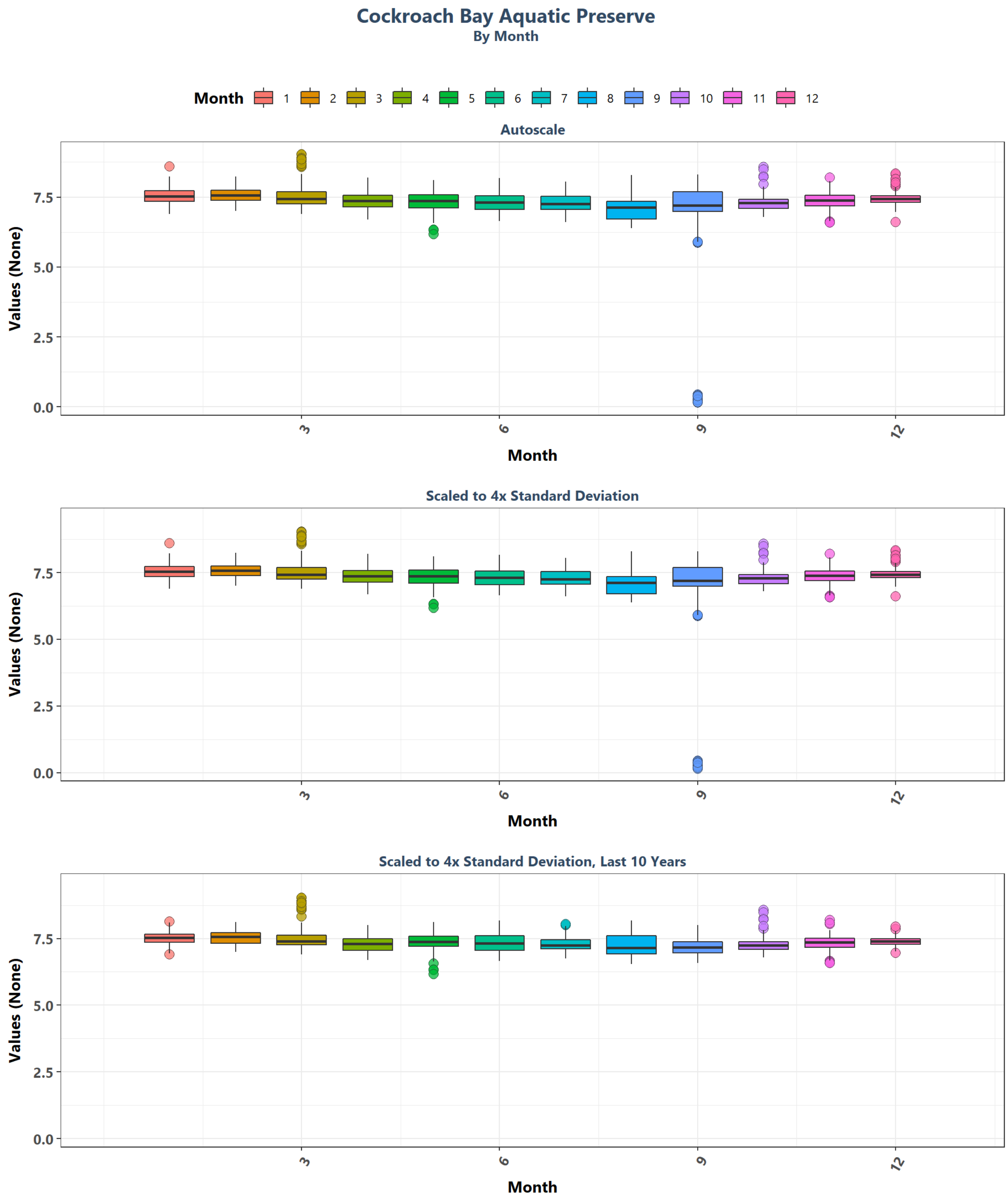




Cockroach Bay Aquatic Preserve
By Year
Autoscale
Values (None)
0.0
2.5
5.0
7.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
0.0
2.5
5.0
7.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
0.0
2.5
5.0
7.5
2011
2013
2015
2017
2019
2021
Year

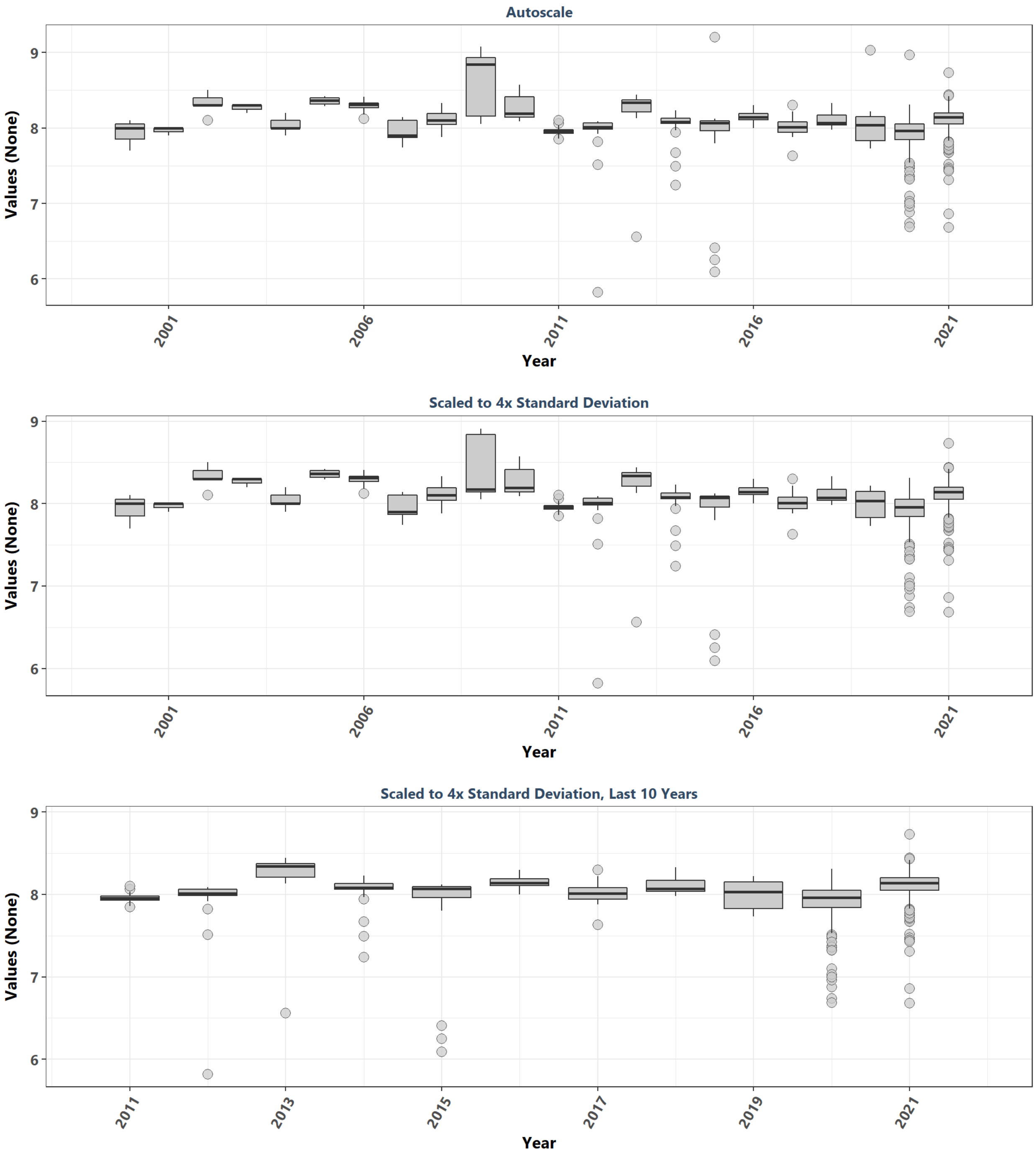


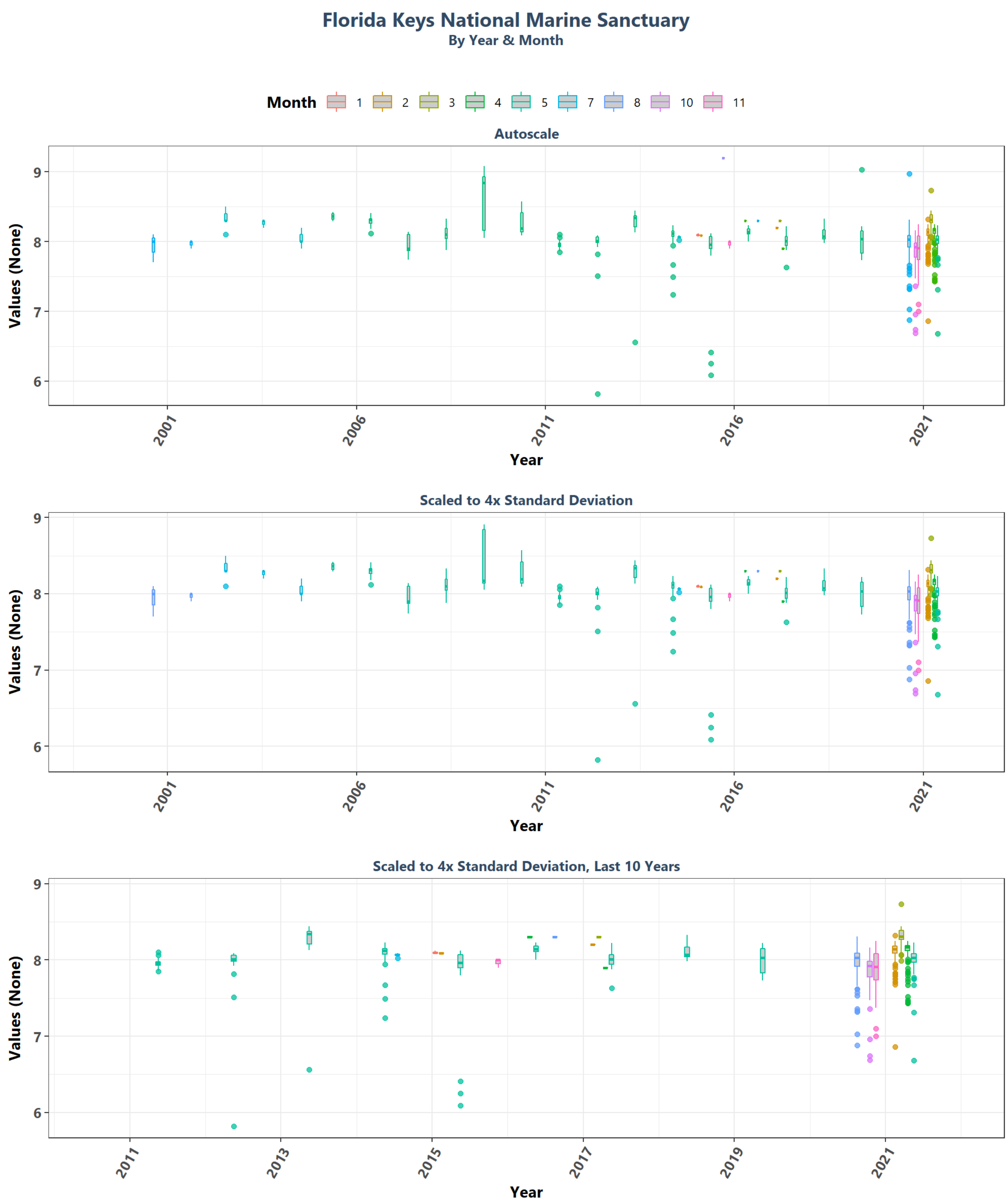
Cockroach Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
7.5
5.0
2.5
0.0
2011
2013
2015
2017
2019
2021
Year

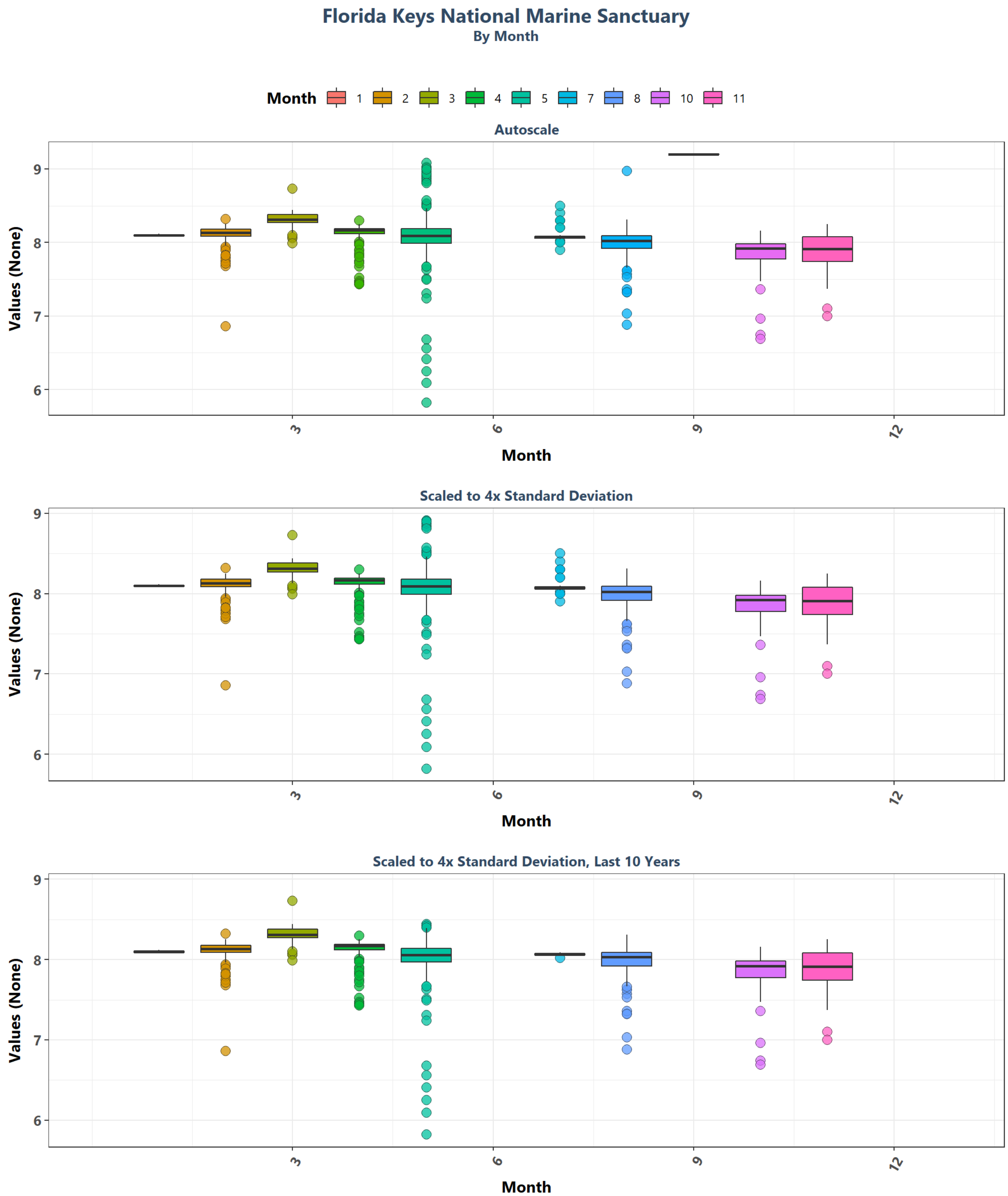

Cockroach Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
7.5
5.0
2.5
0.0
3
6
9
12
Month

Florida Keys National Marine Sanctuary

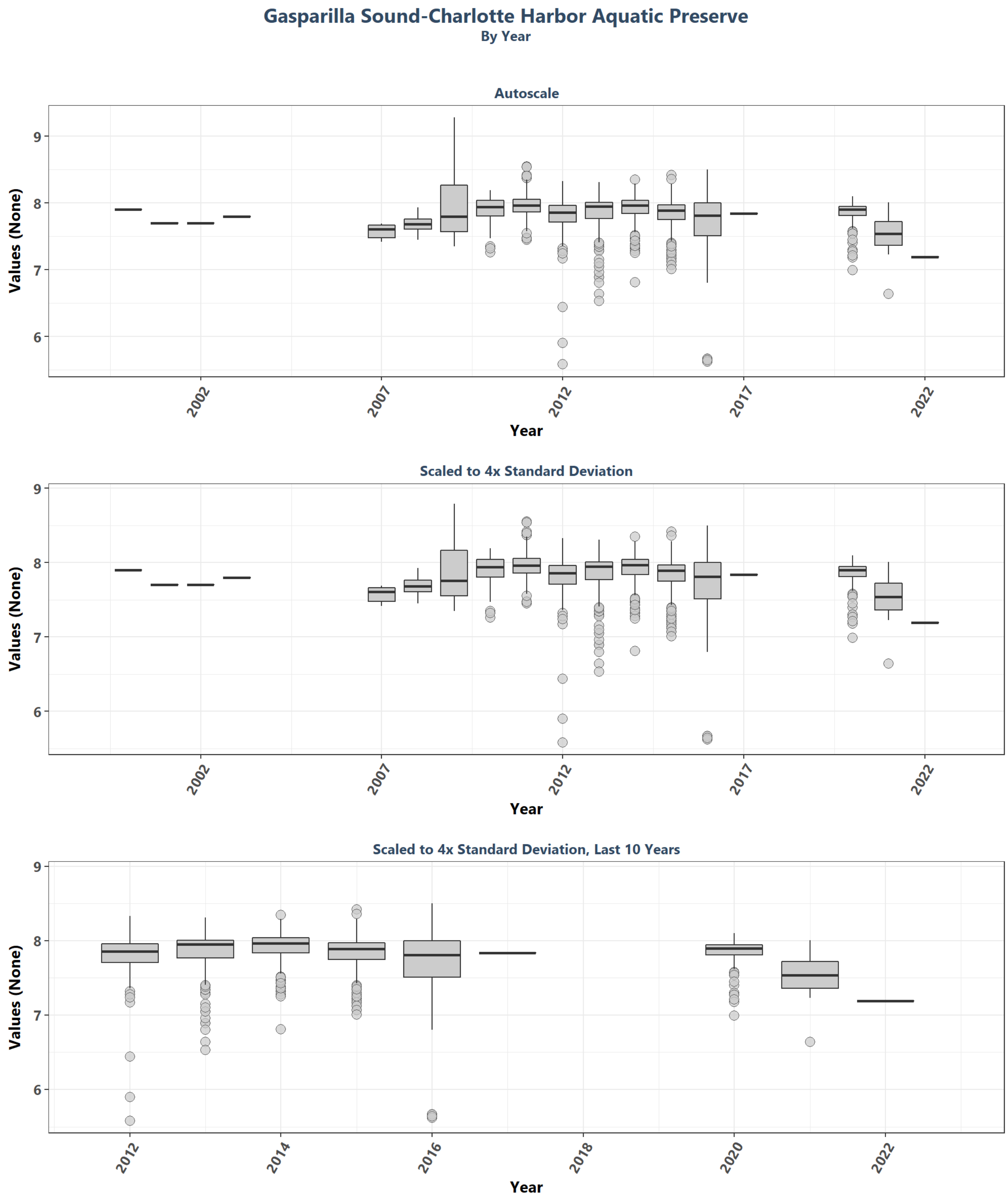
By Year



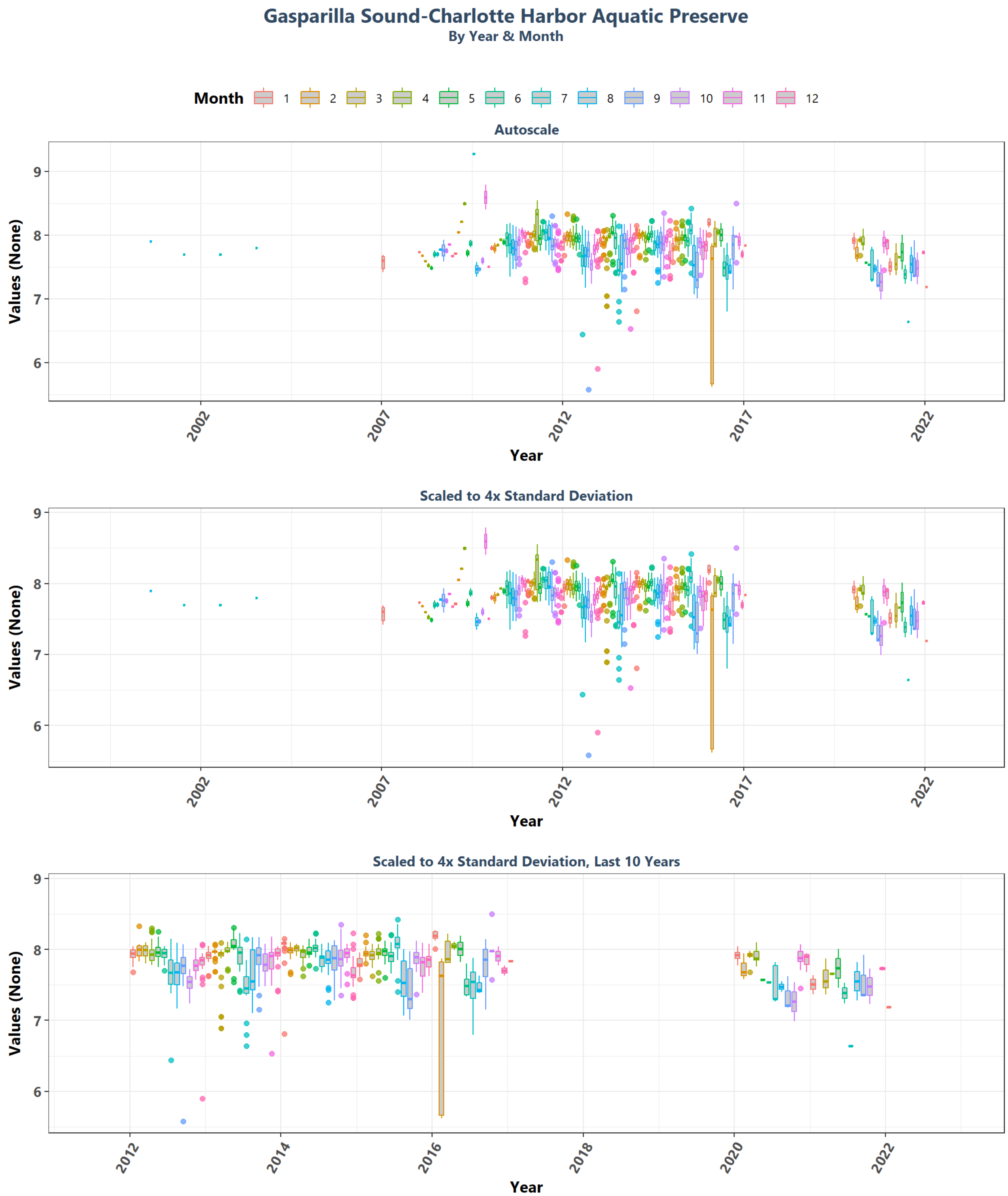

Florida Keys National Marine Sanctuary
By Year & Month
Month 1 2 3 4 5 7 8 10 11
Autoscale
Values (None)
9
8
7
6
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
2011
2013
2015
2017
2019
2021
Year



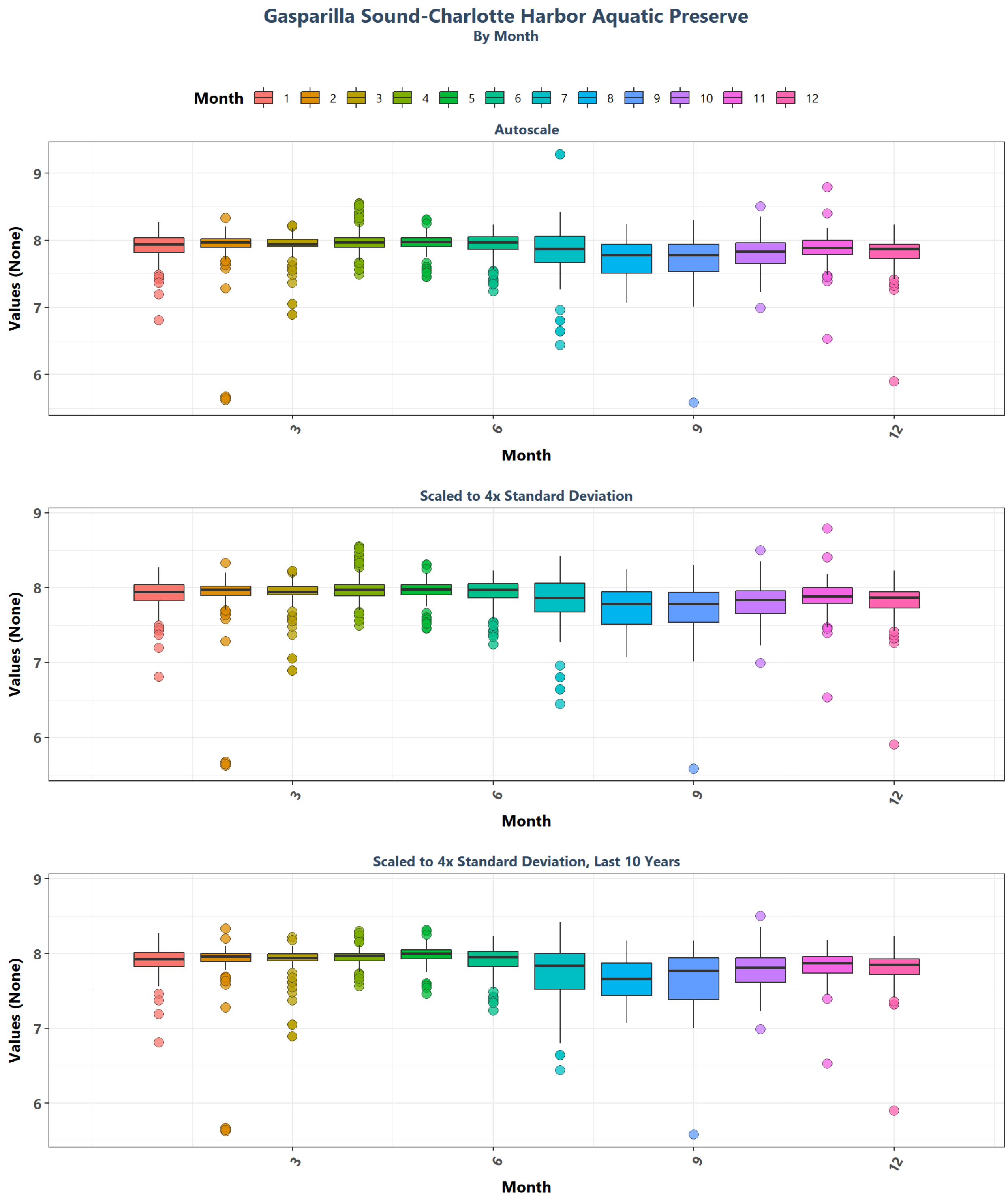
Florida Keys National Marine Sanctuary
By Month
Month 1 2 3 4 5 7 8 10 11
Autoscale
Values (None)
9
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
3
6
9
12
Month



Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year
Autoscale
Values (None)
9
8
7
6
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
2012
2014
2016
2018
2020
2022
Year

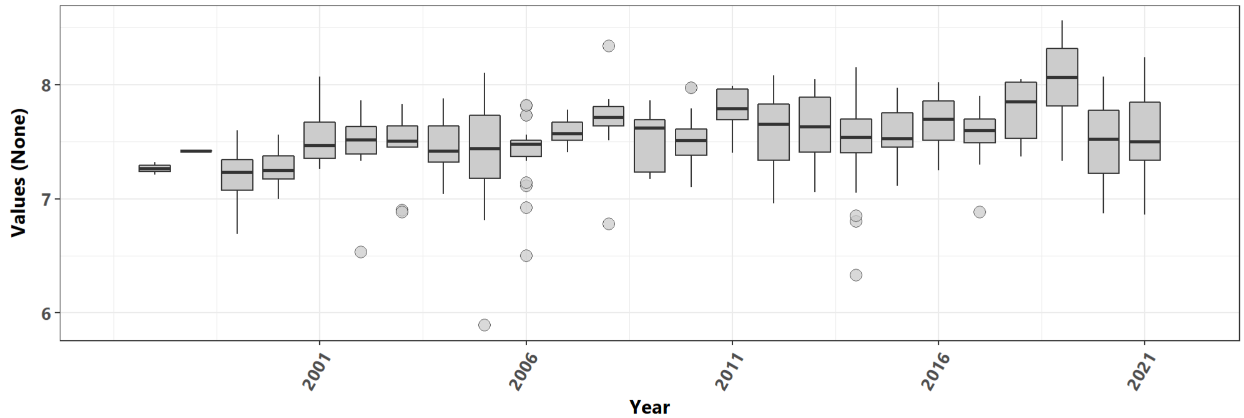


Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
Year
Scaled to 4x Standard Deviation
Values (None)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year

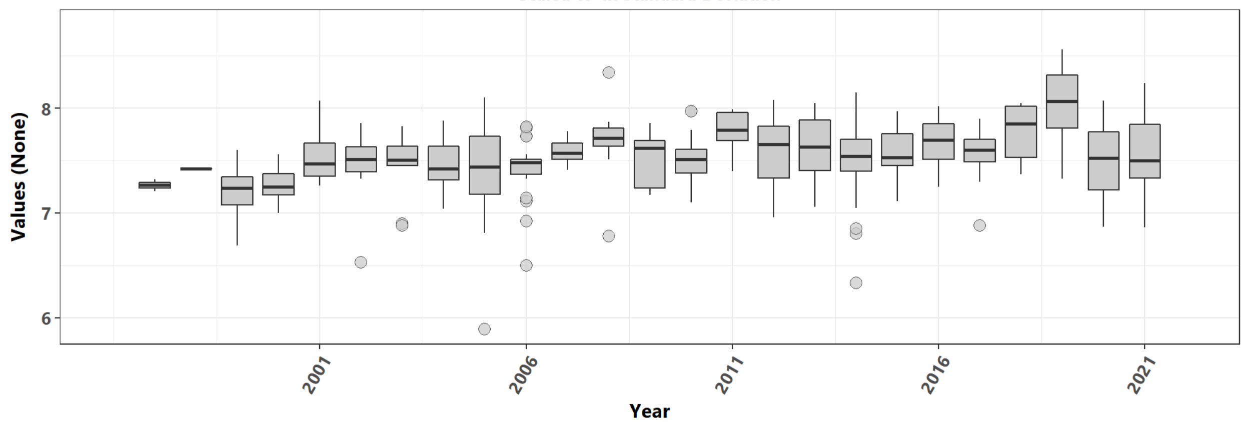

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Month
9
8
7
6
3
6
9
12

Guana River Marsh Aquatic Preserve
By Year

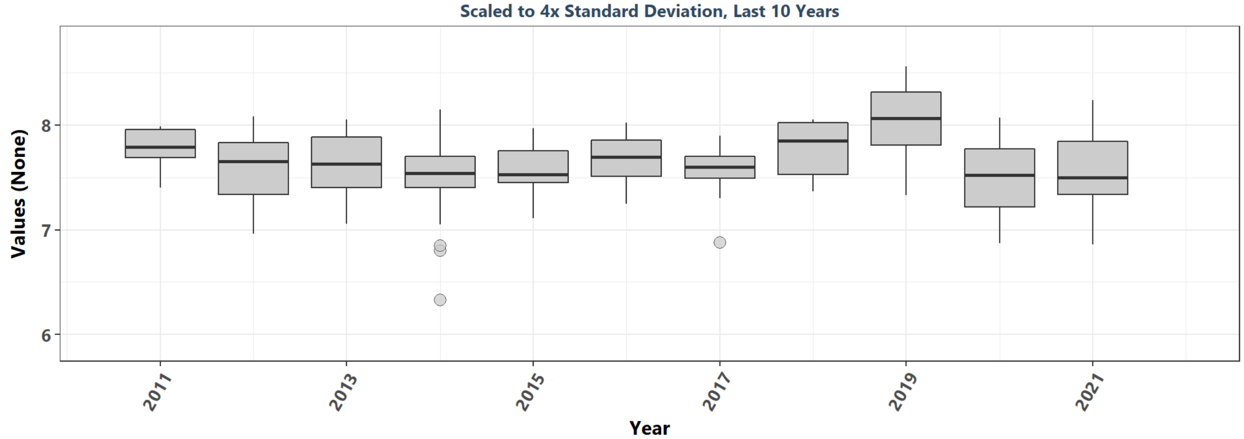
Autoscale

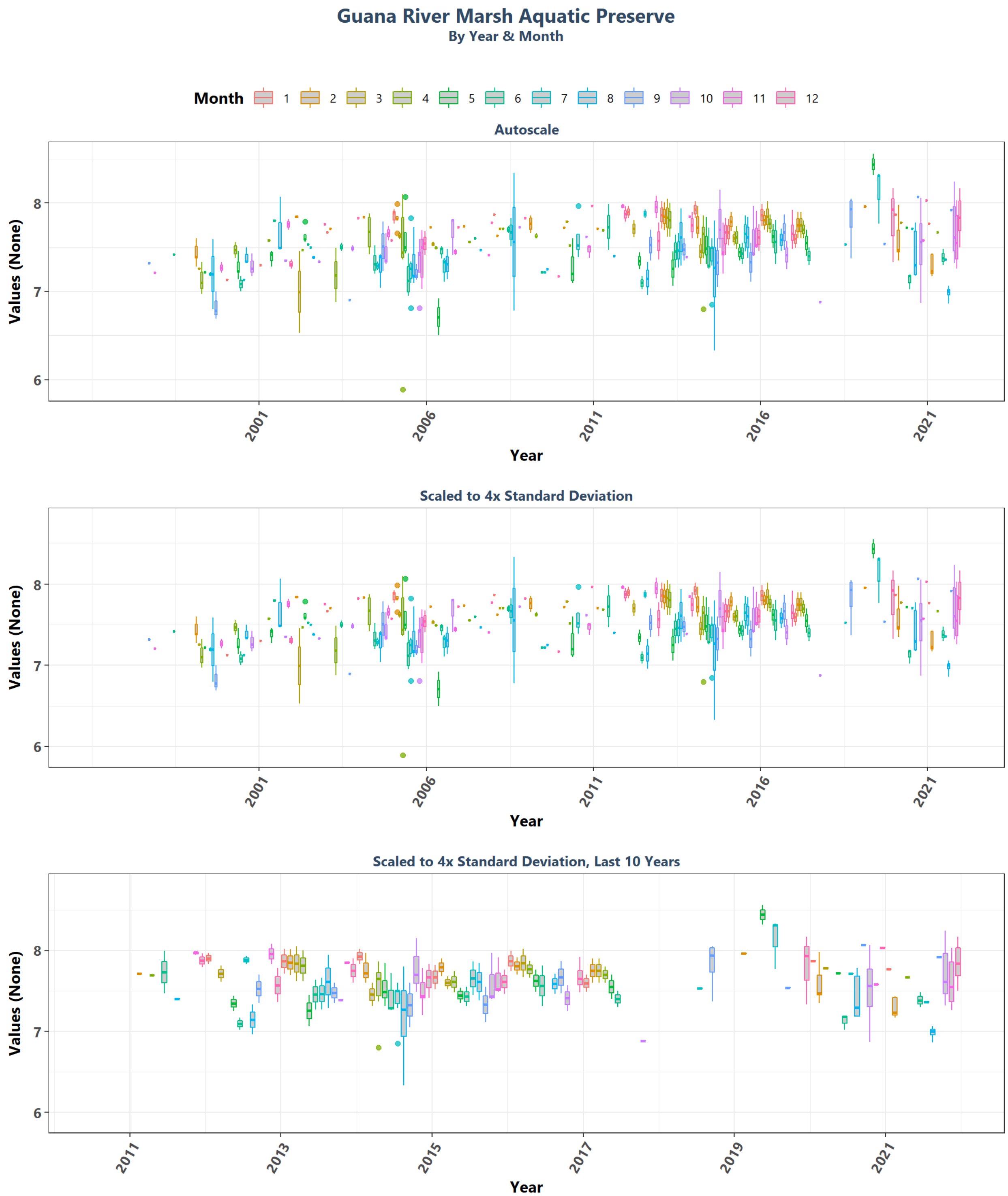


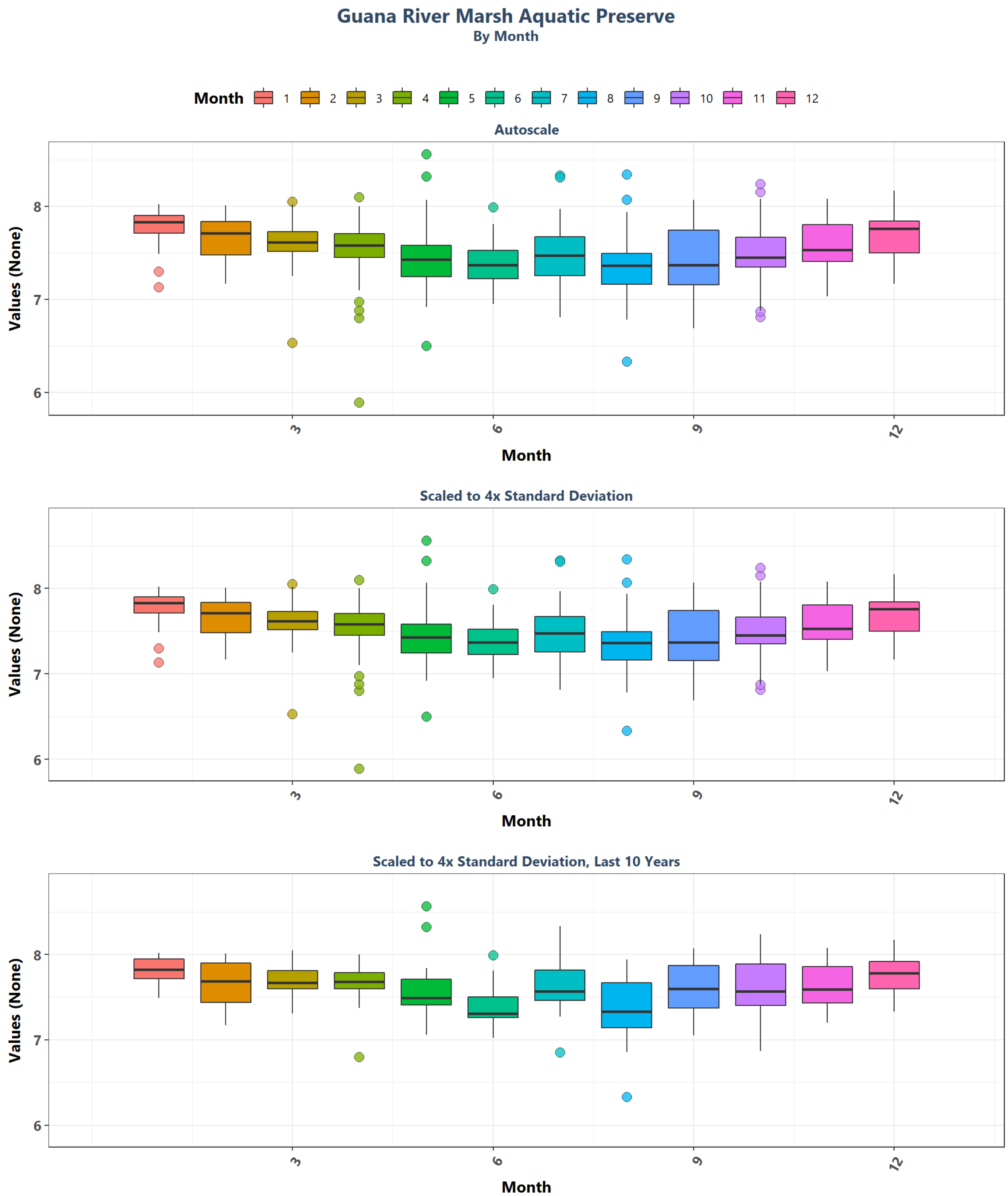
Scaled to 4x Standard Deviation



Scaled to 4x Standard Deviation, Last 10 Years



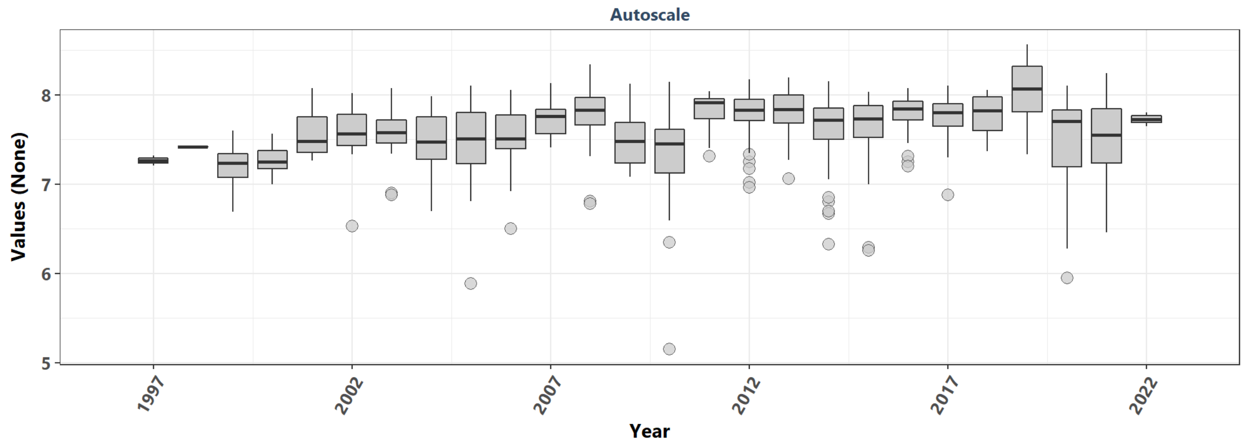

Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
6
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
8
7
6
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8
7
6
2011
2013
2015
2017
2019
2021
Year

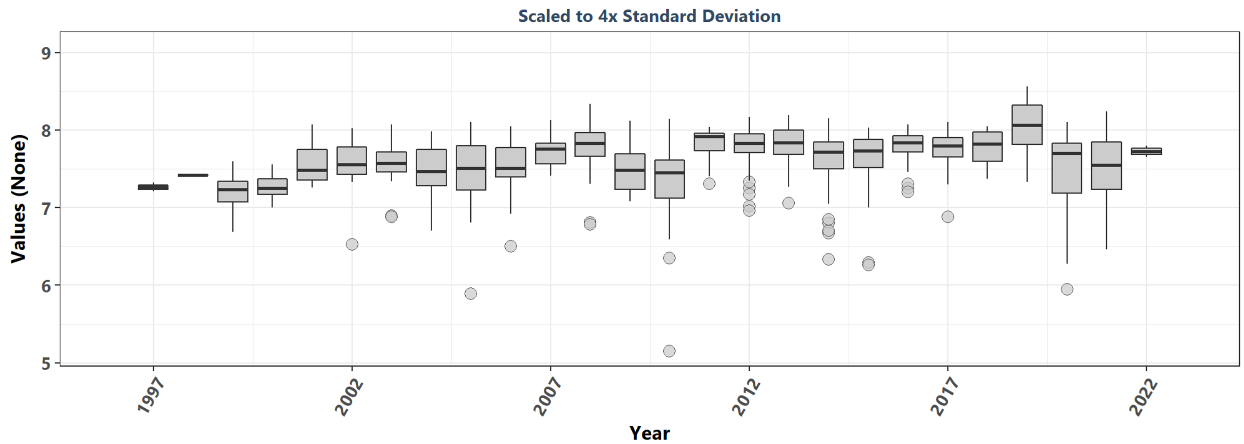


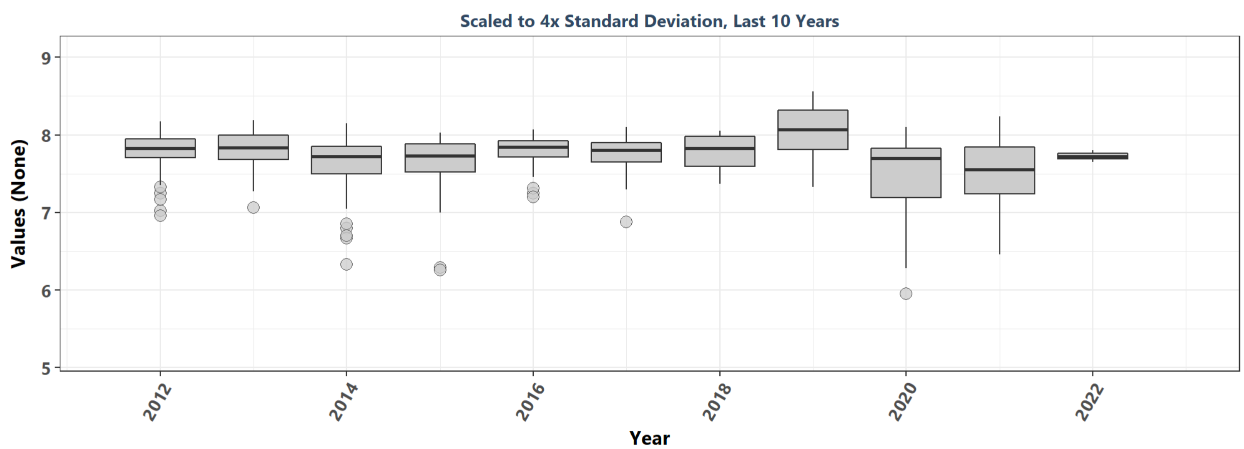
Guana River Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8
7
6
3
6
9
12
Month

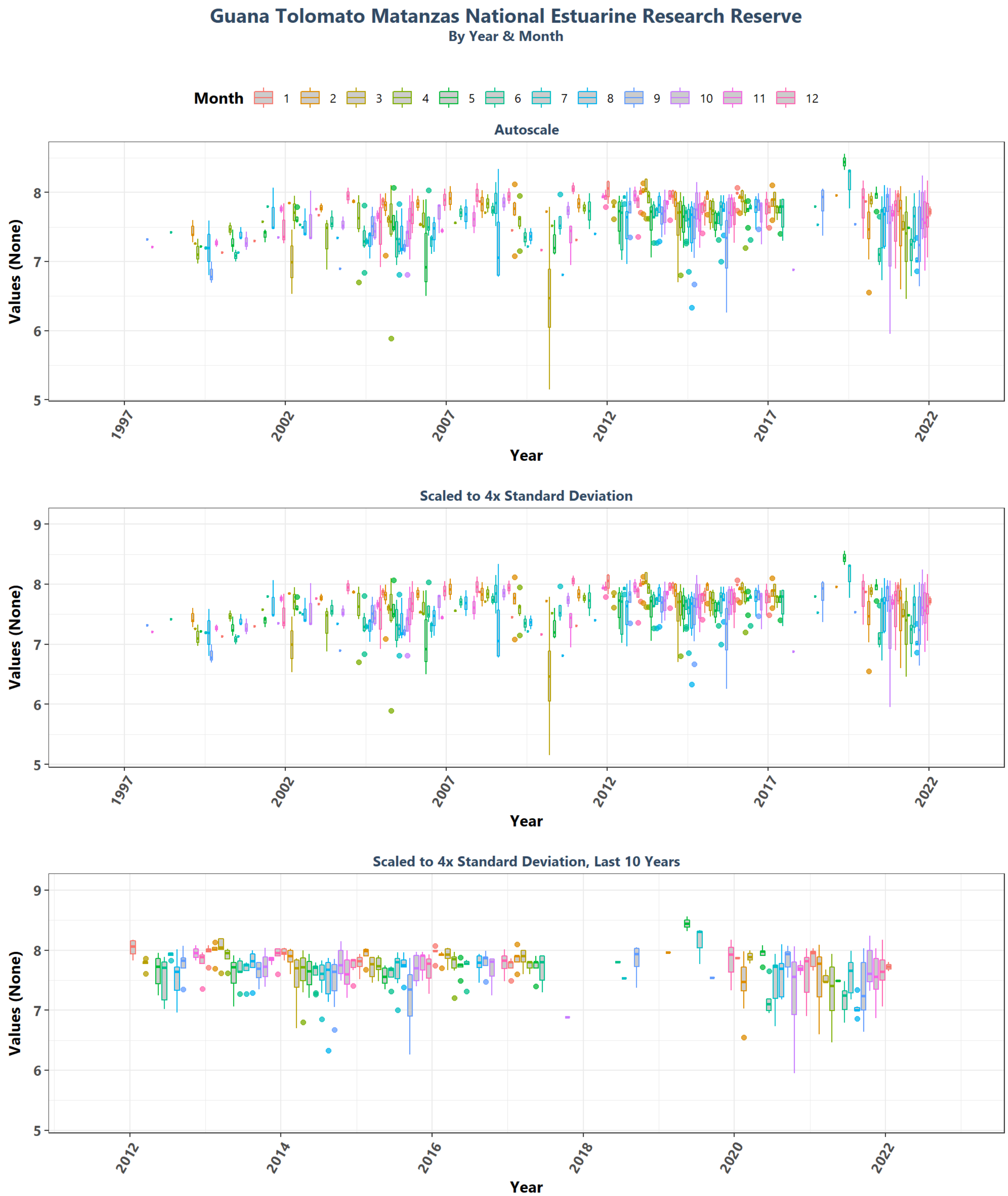
Guana Tolomato Matanzas National Estuarine Research Reserve

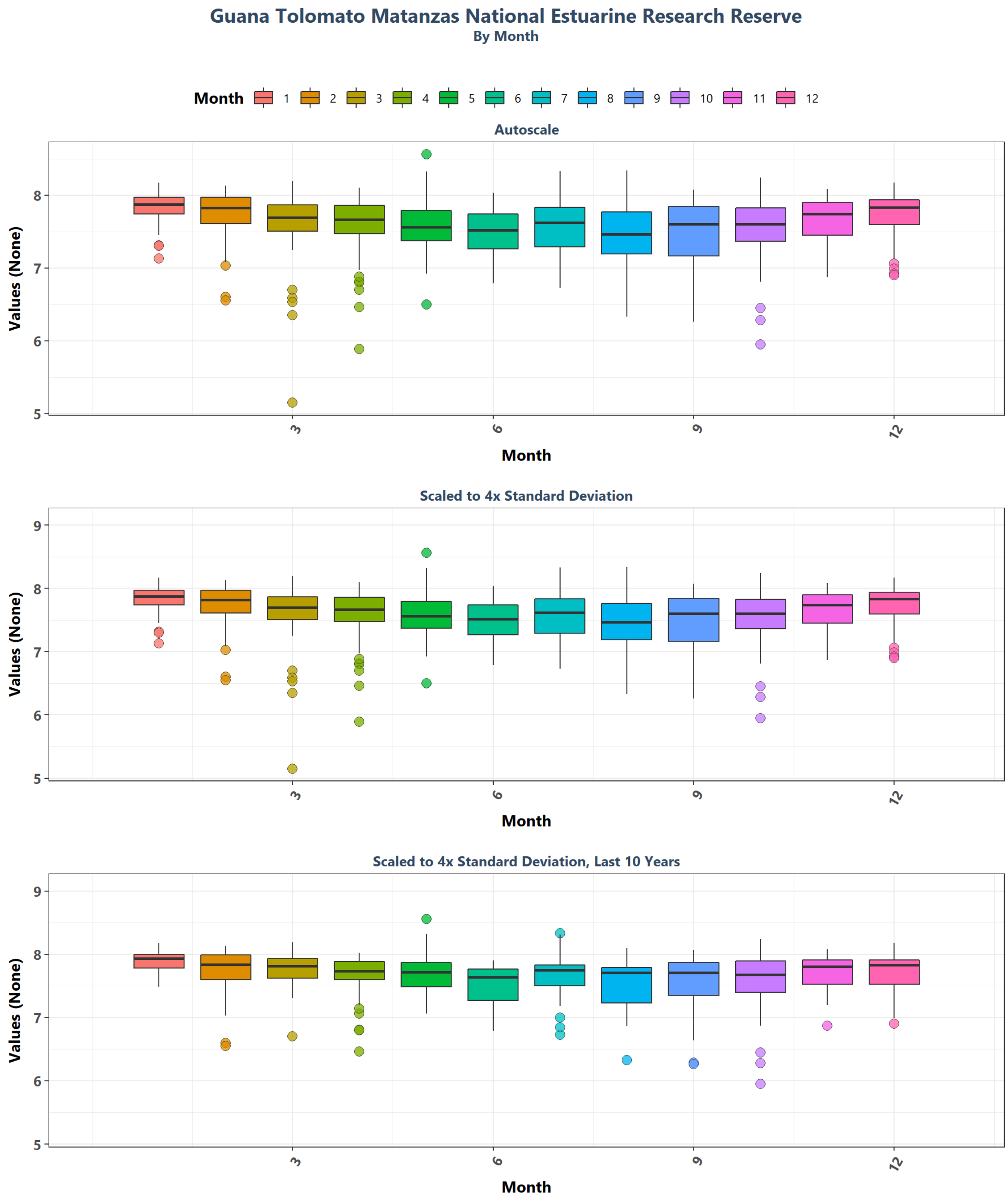
By Year



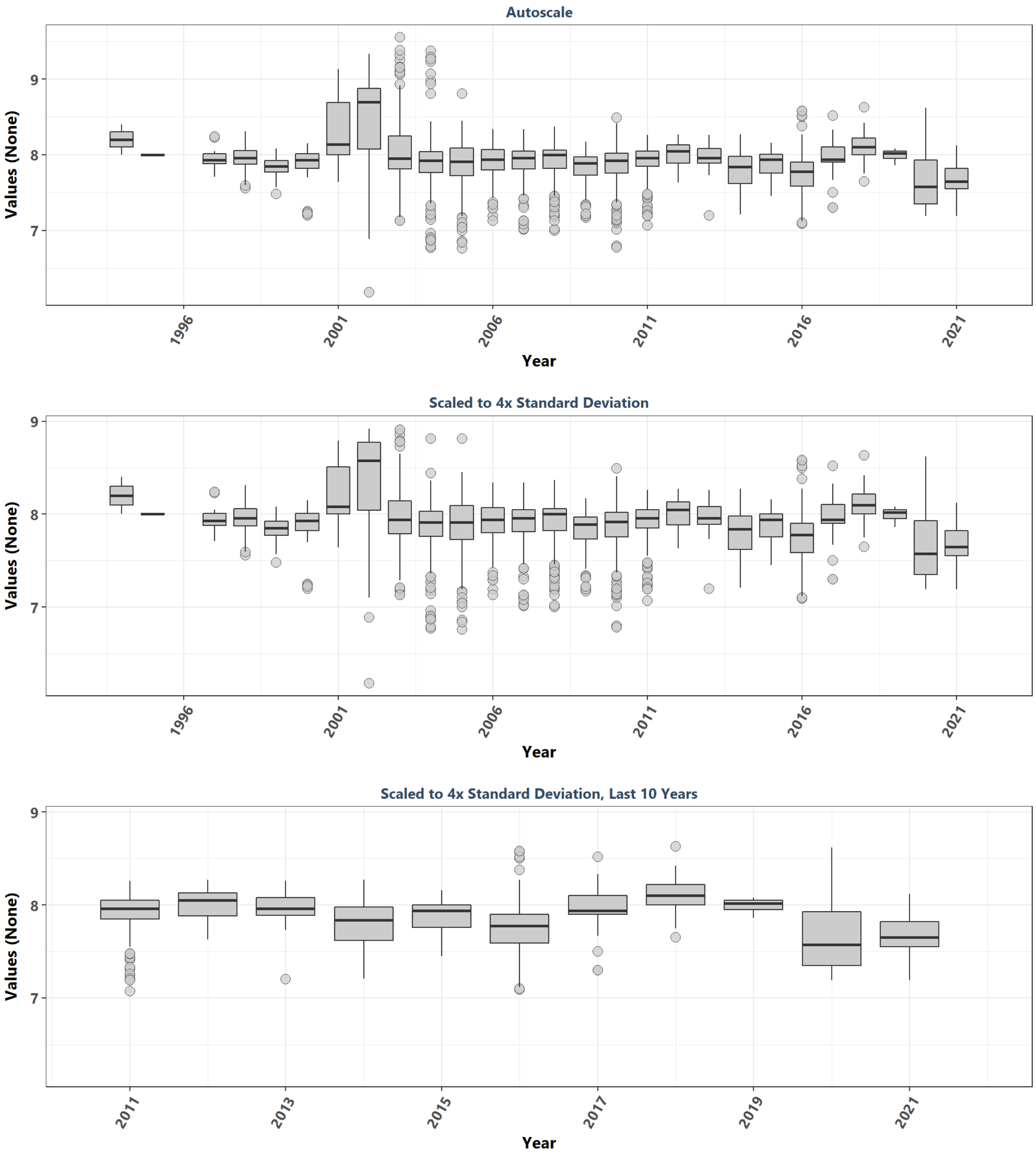


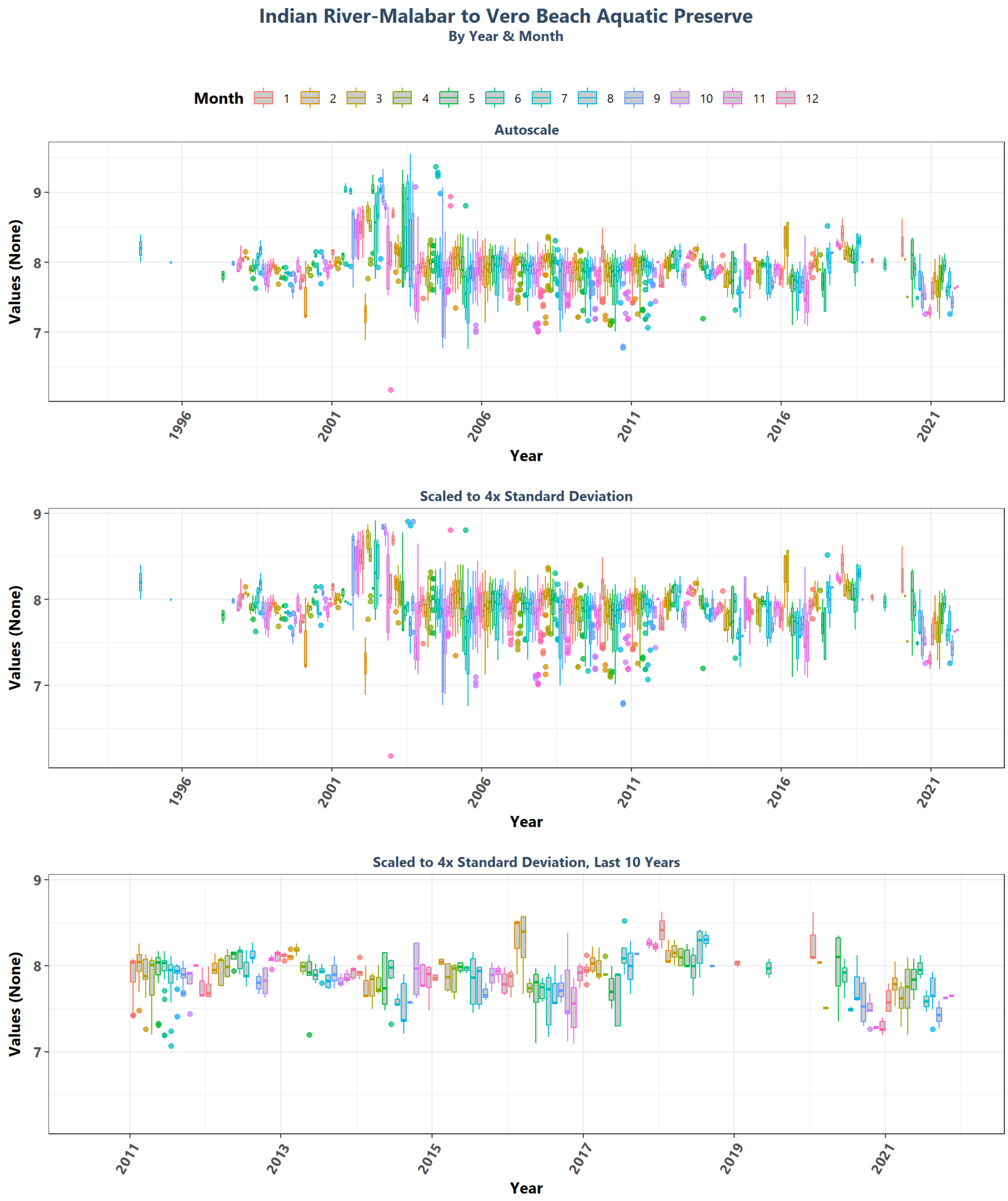



Guana Tolomato Matanzas National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
Year
Scaled to 4x Standard Deviation
Values (None)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year


Guana Tolomato Matanzas National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
6
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
5
3
6
9
12
Month

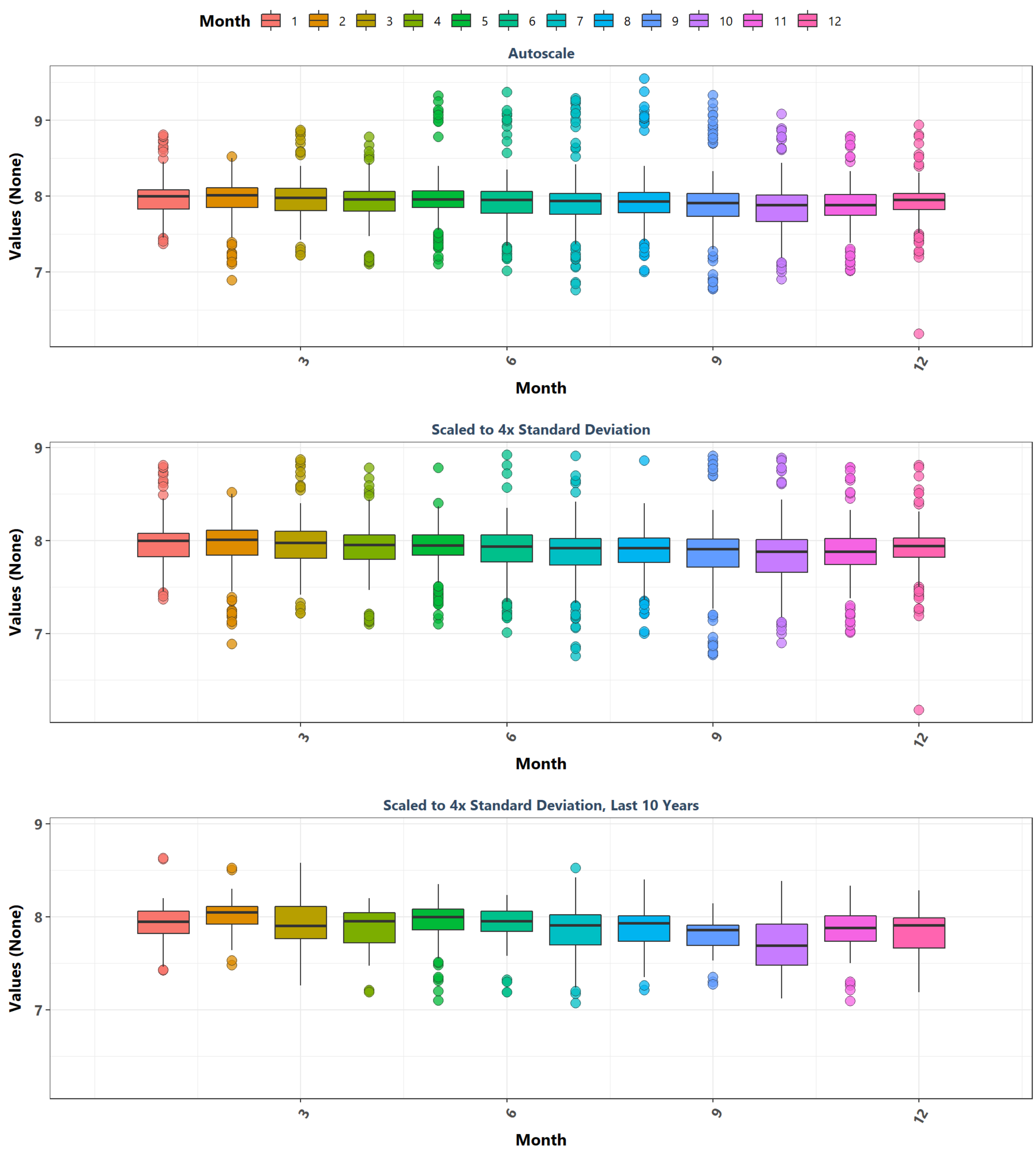
Indian River-Malabar to Vero Beach Aquatic Preserve
By Year




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 (None)
9
8
7
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
2011
2013
2015
2017
2019
2021
Year

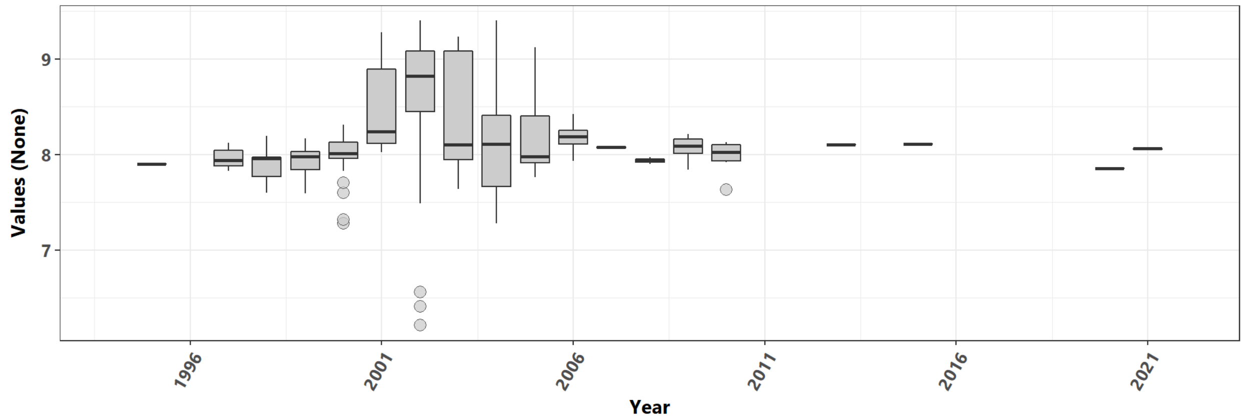
Indian River-Malabar to Vero Beach Aquatic Preserve

By Month

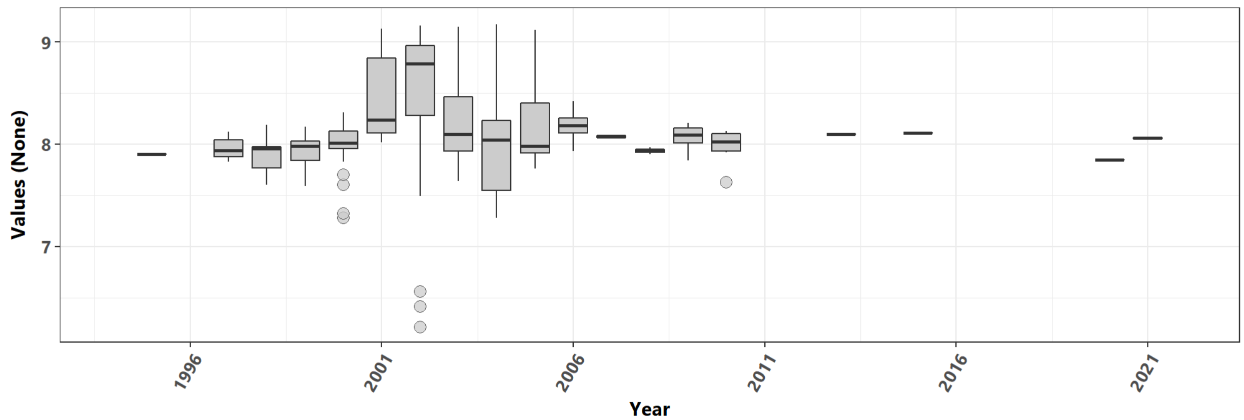


Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year

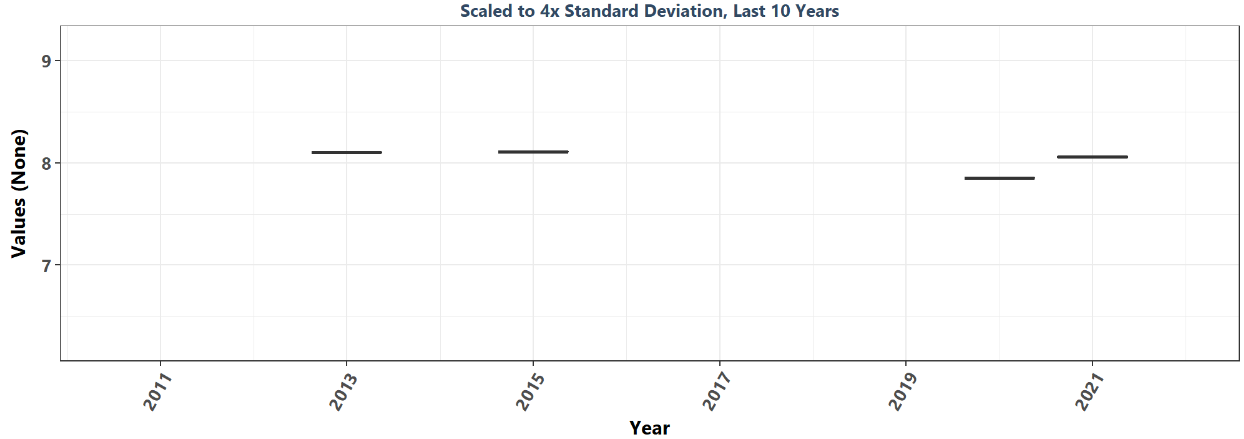
Autoscale



Scaled to 4x Standard Deviation

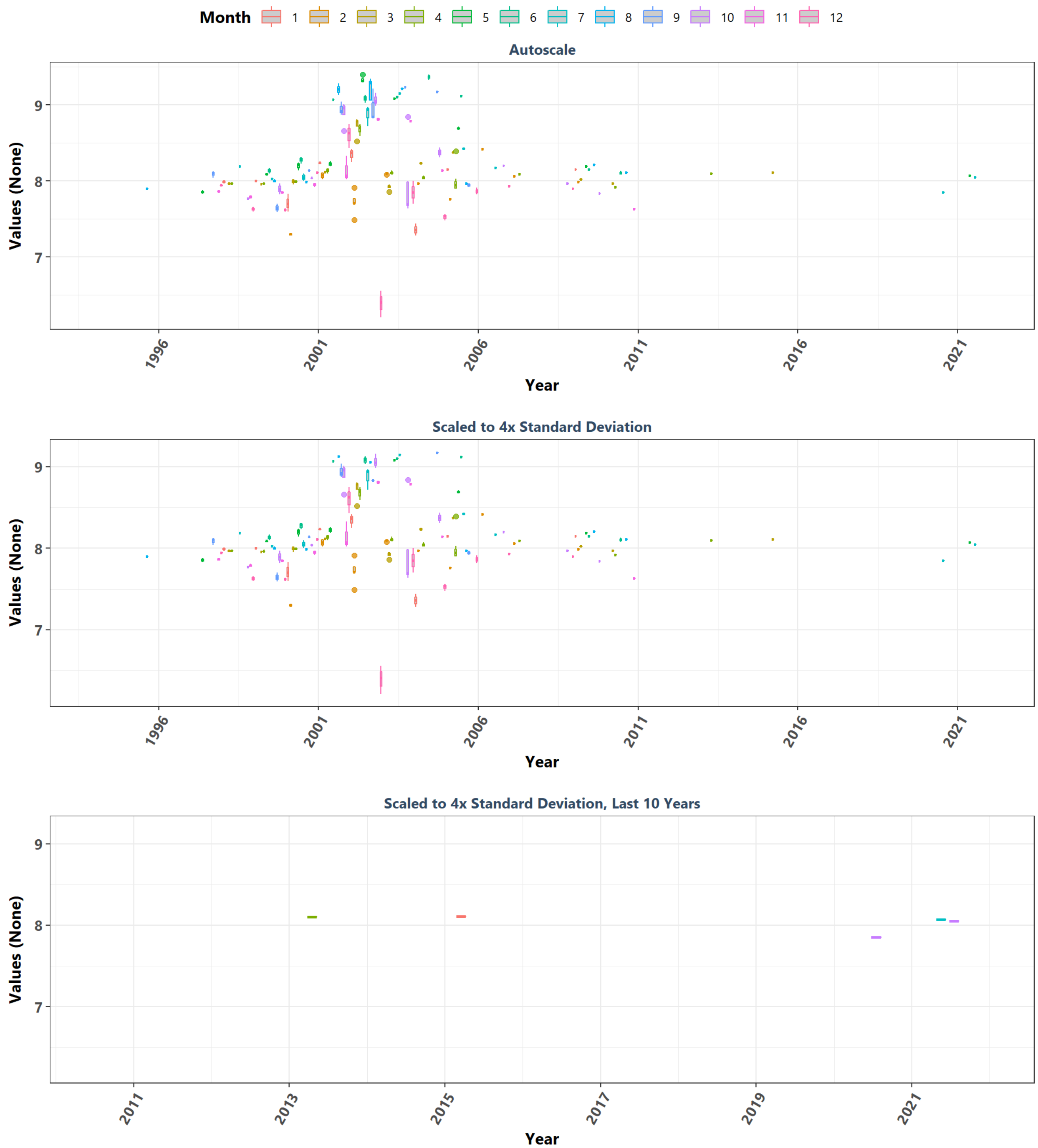


Scaled to 4x Standard Deviation, Last 10 Years



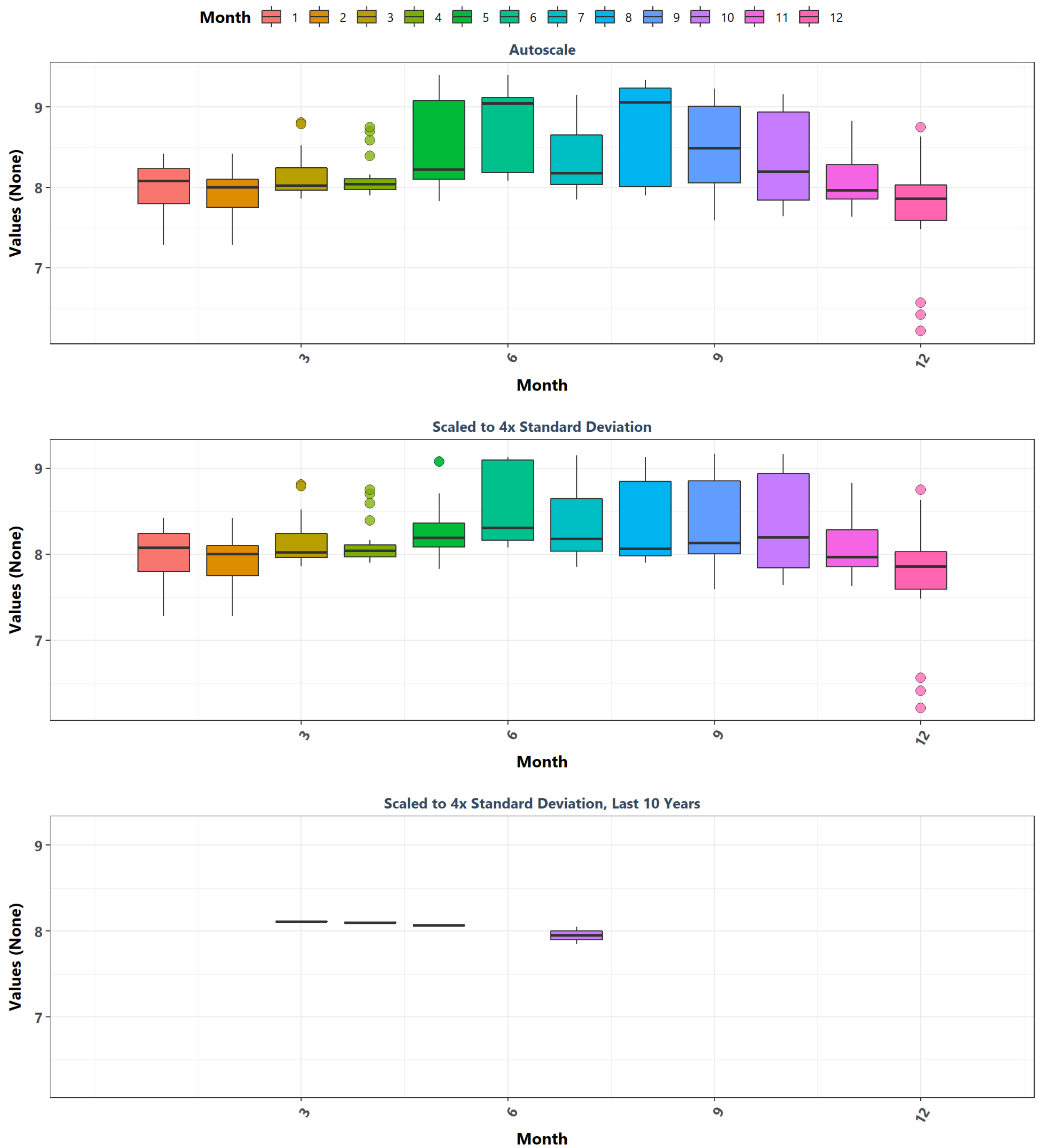
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

By Year & Month



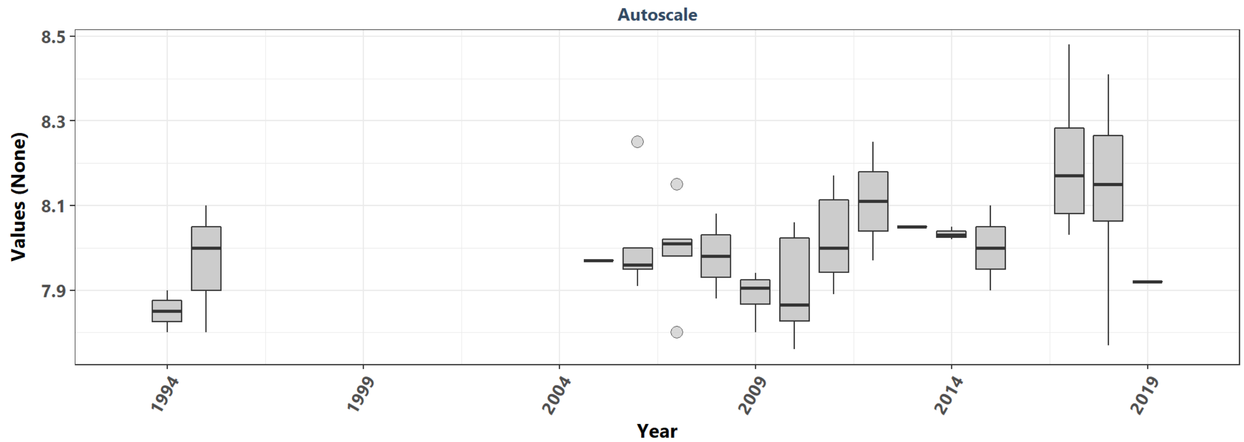
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

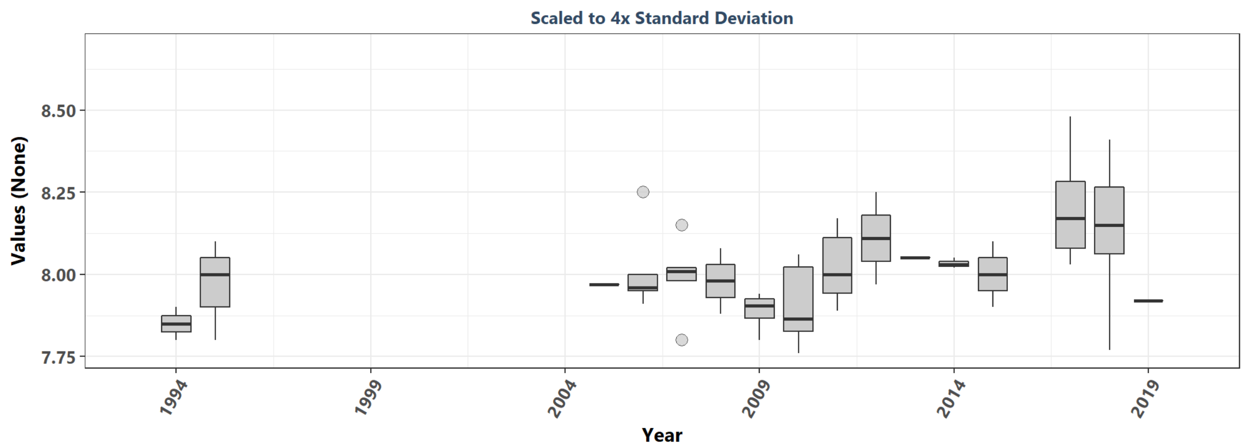
By Month

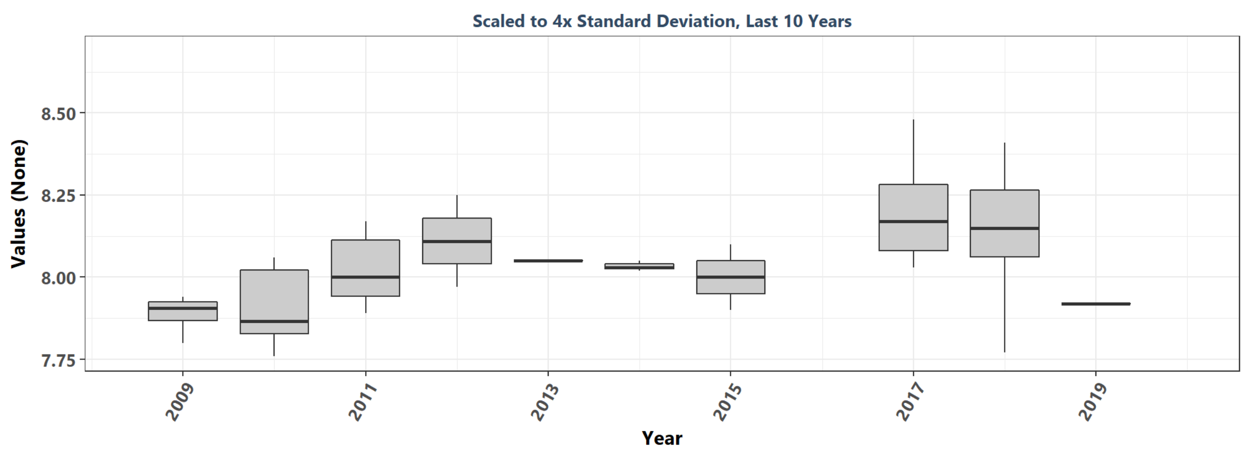


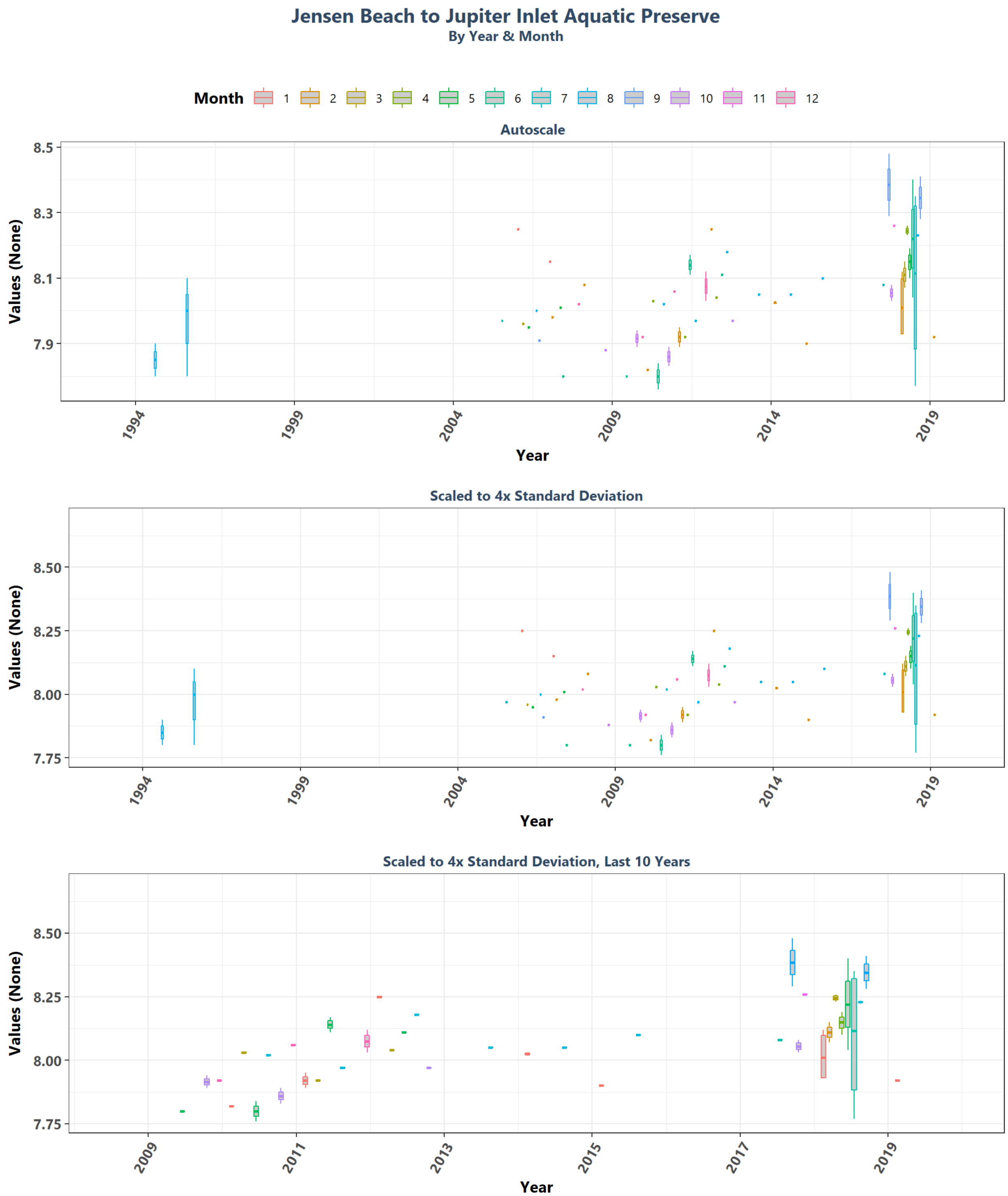
Jensen Beach to Jupiter Inlet Aquatic Preserve

By Year





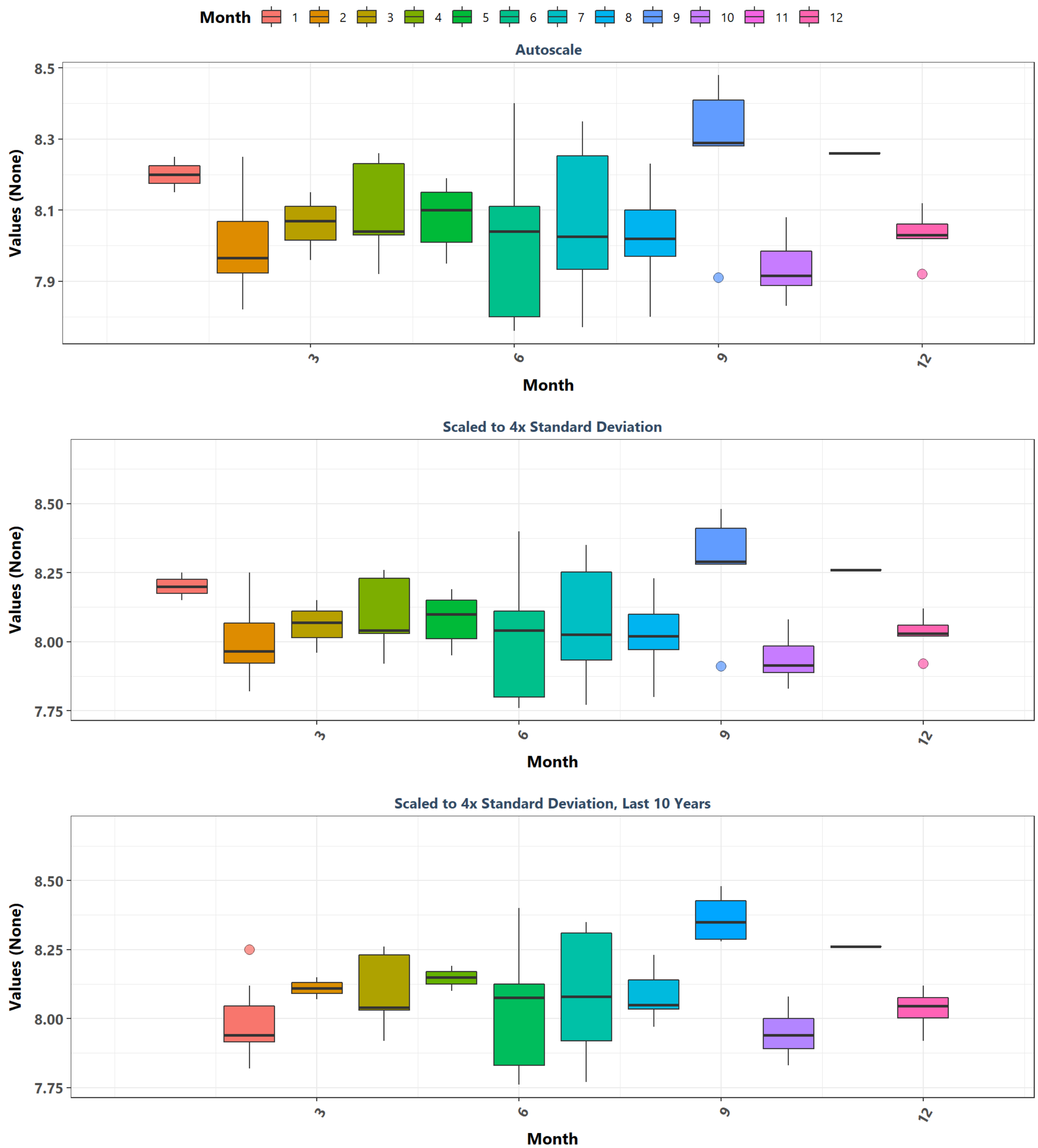




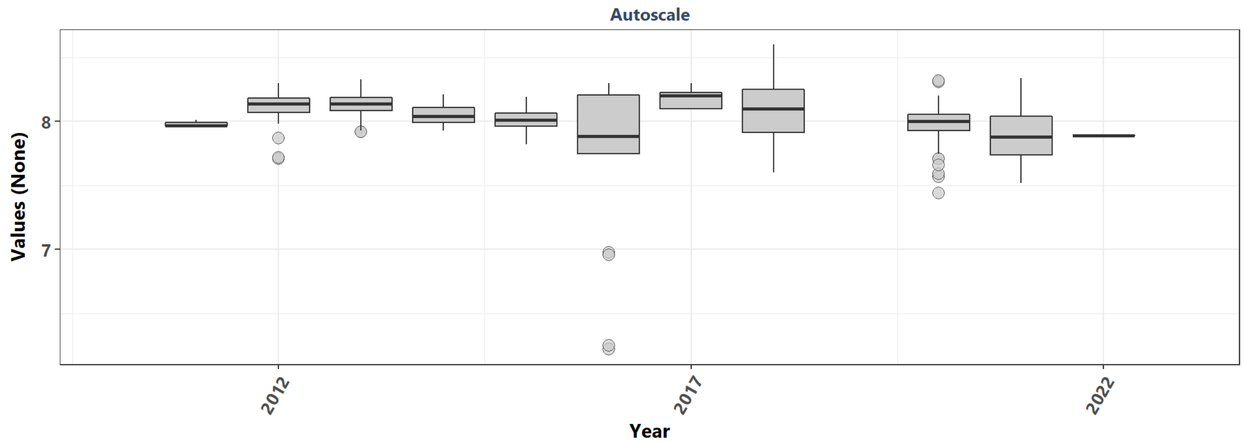
Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.5
8.3
8.1
7.9
1994
1999
2004
2009
2014
2019
Year
Scaled to 4x Standard Deviation
Values (None)
8.50
8.25
8.00
7.75
1994
1999
2004
2009
2014
2019
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8.50
8.25
8.00
7.75
2009
2011
2013
2015
2017
2019
Year

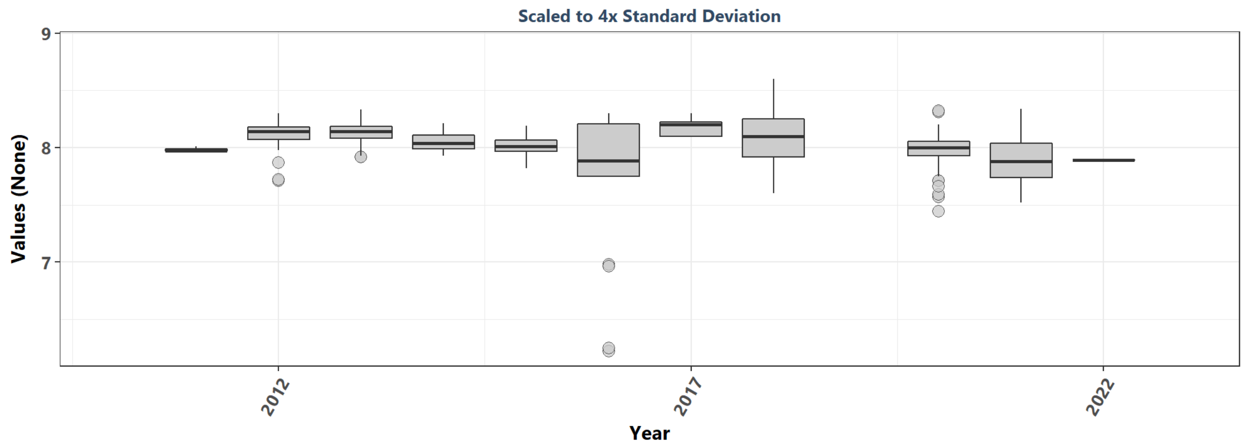
Jensen Beach to Jupiter Inlet Aquatic Preserve

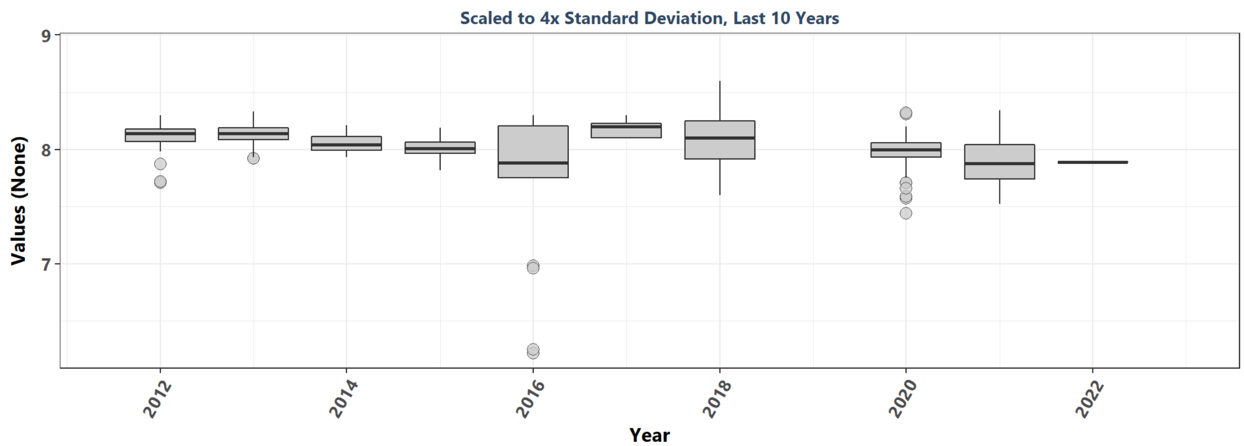
By Month

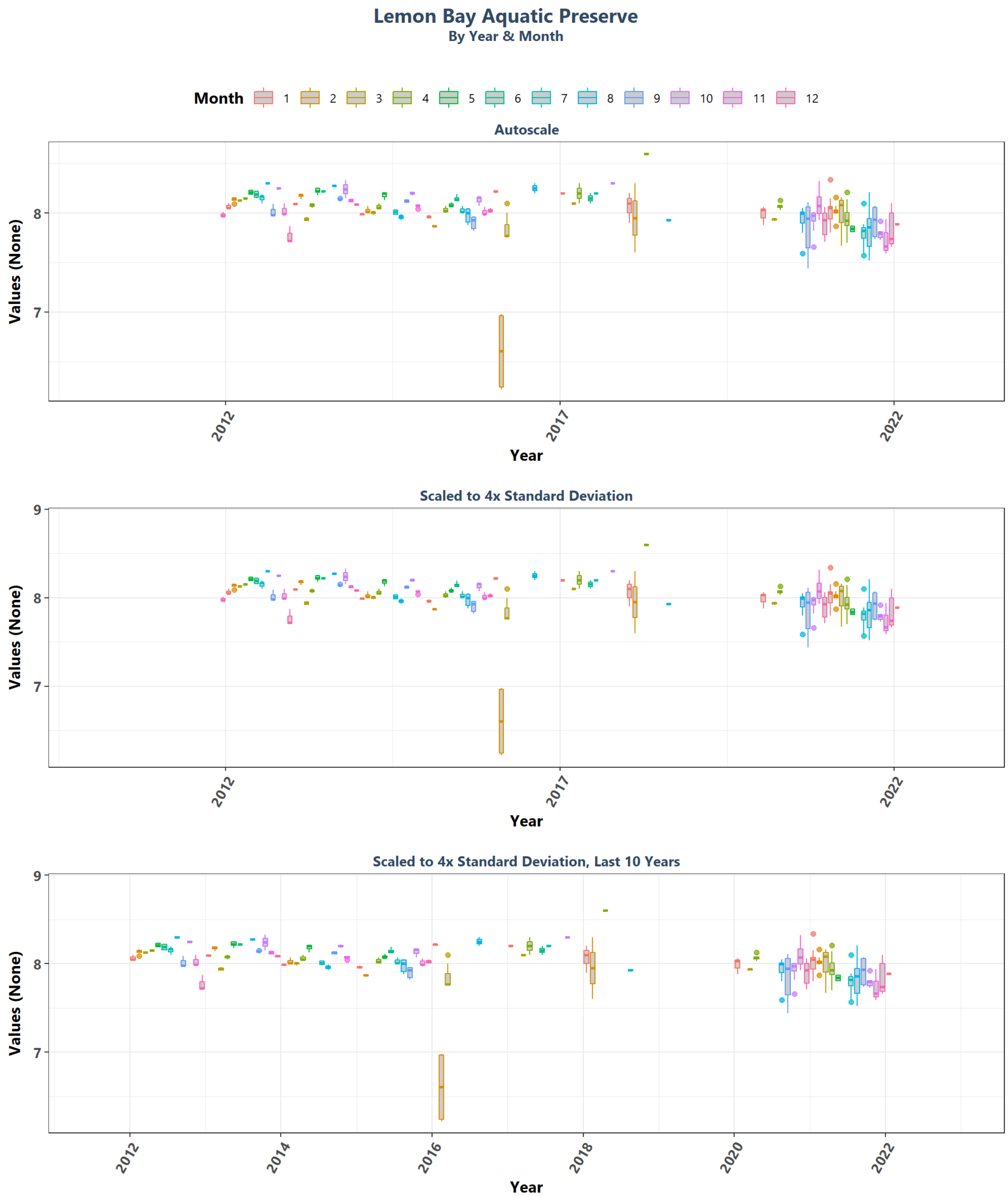


Lemon Bay Aquatic Preserve
By Year

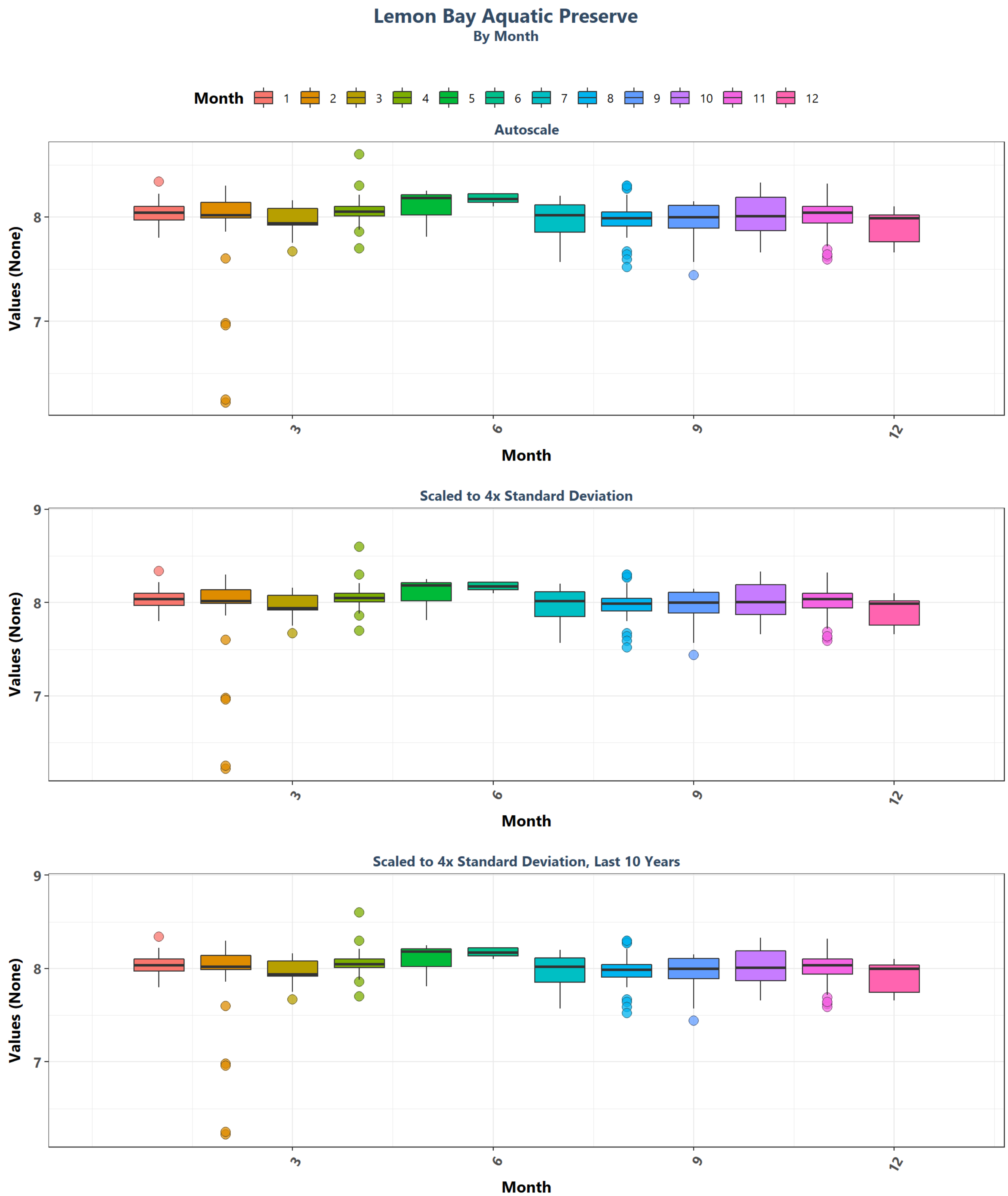






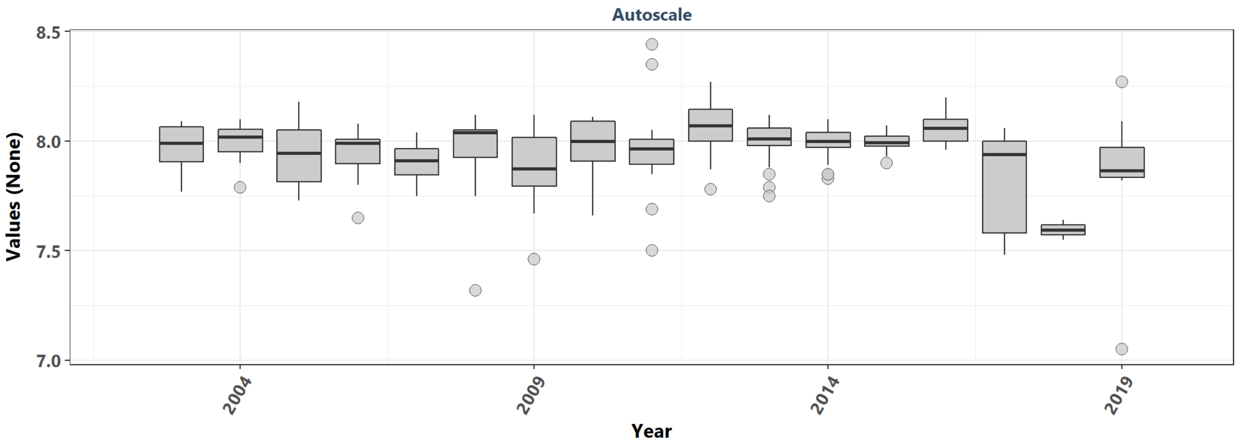


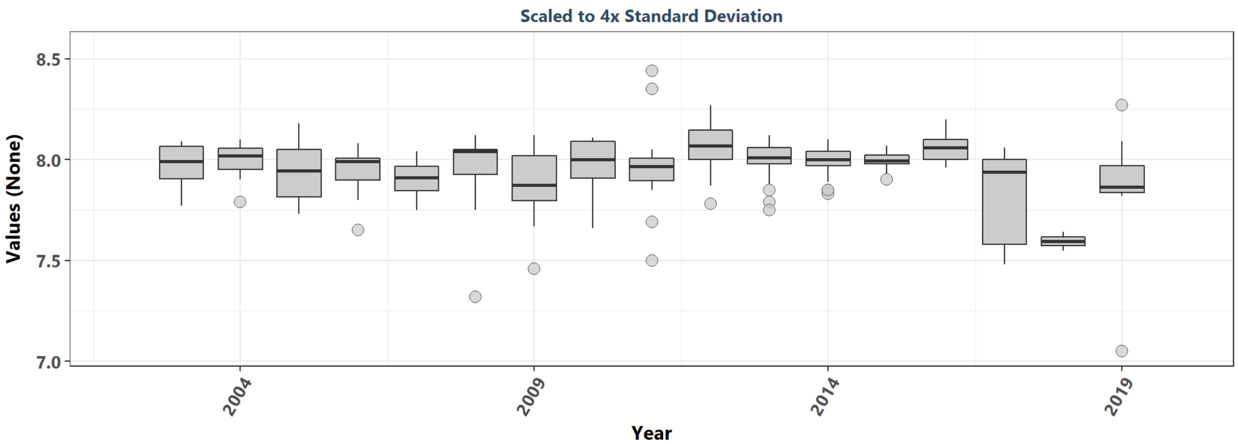
Lemon Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
2012
2014
2016
2018
2020
2022
Year

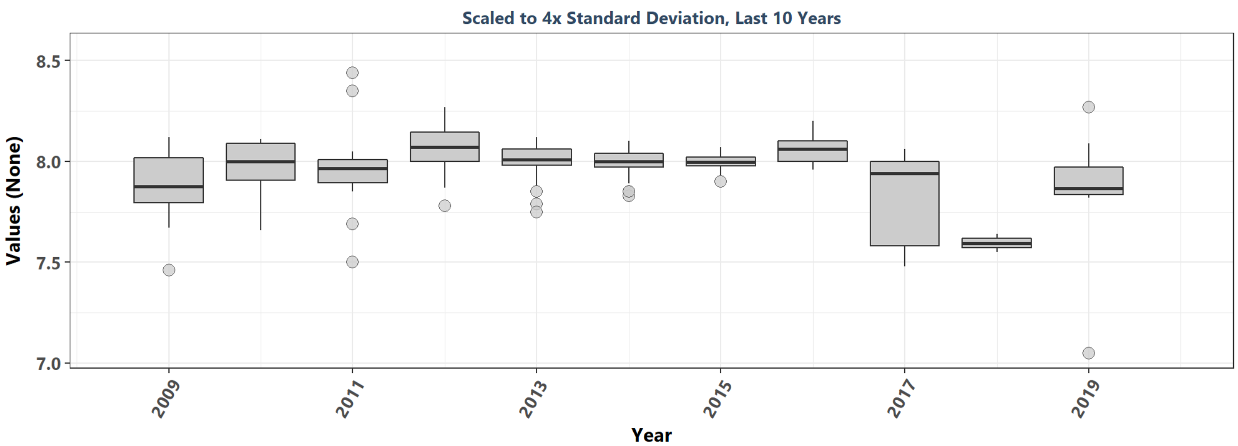

Lemon Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9
8
7
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
3
6
9
12
Month

Loxahatchee River-Lake Worth Creek Aquatic Preserve

By Year

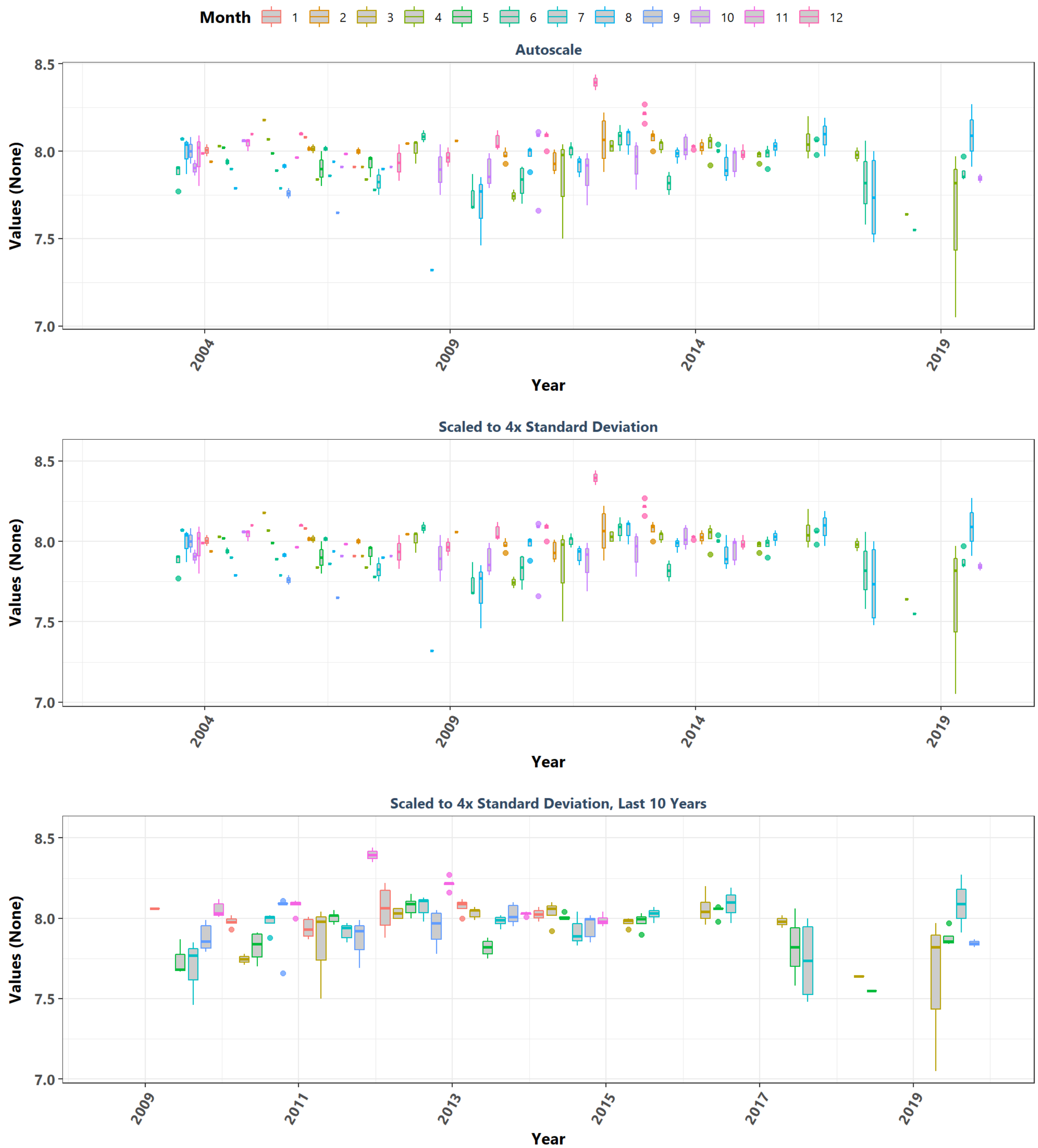


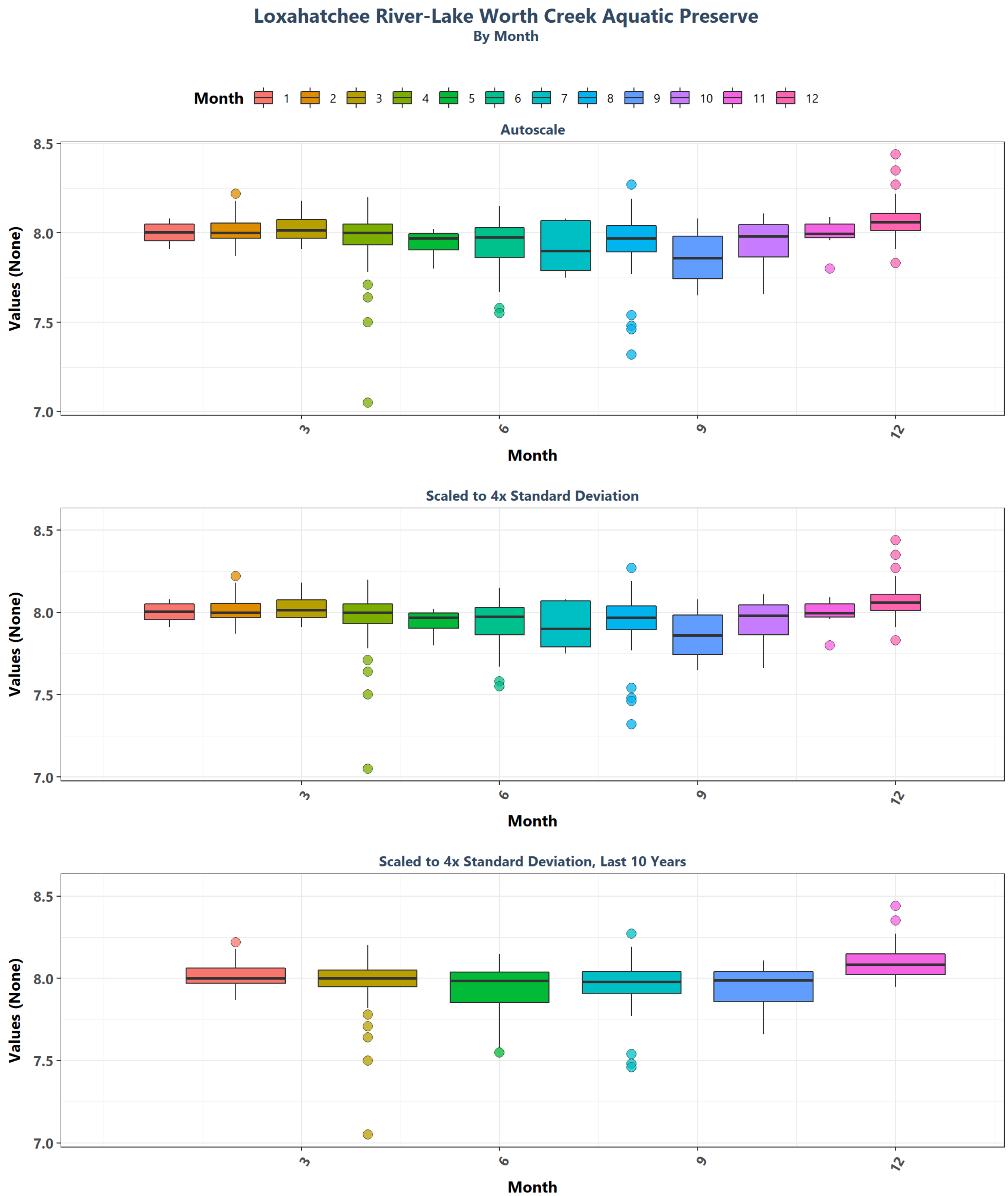


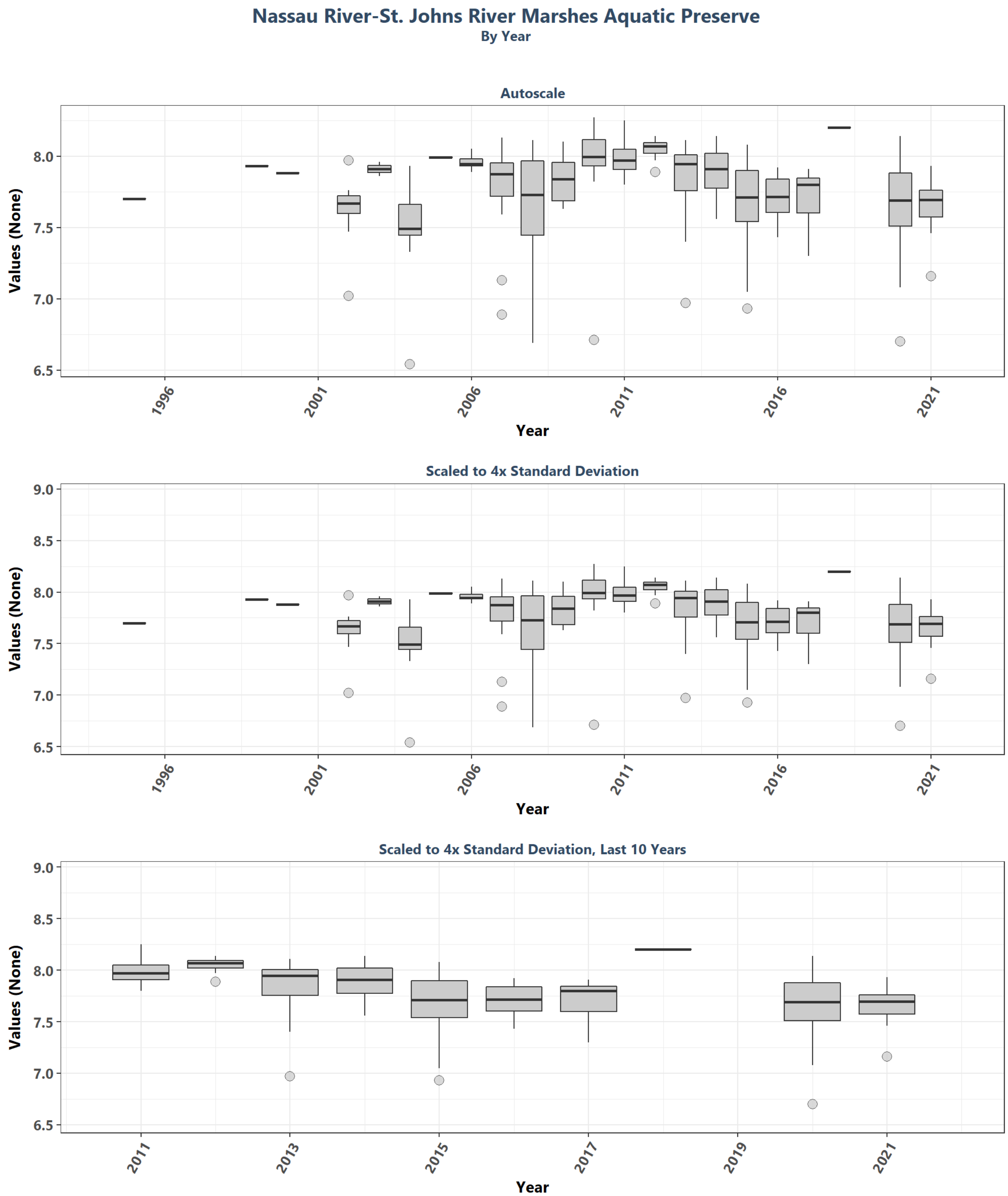


Loxahatchee River-Lake Worth Creek Aquatic Preserve

By Year & Month

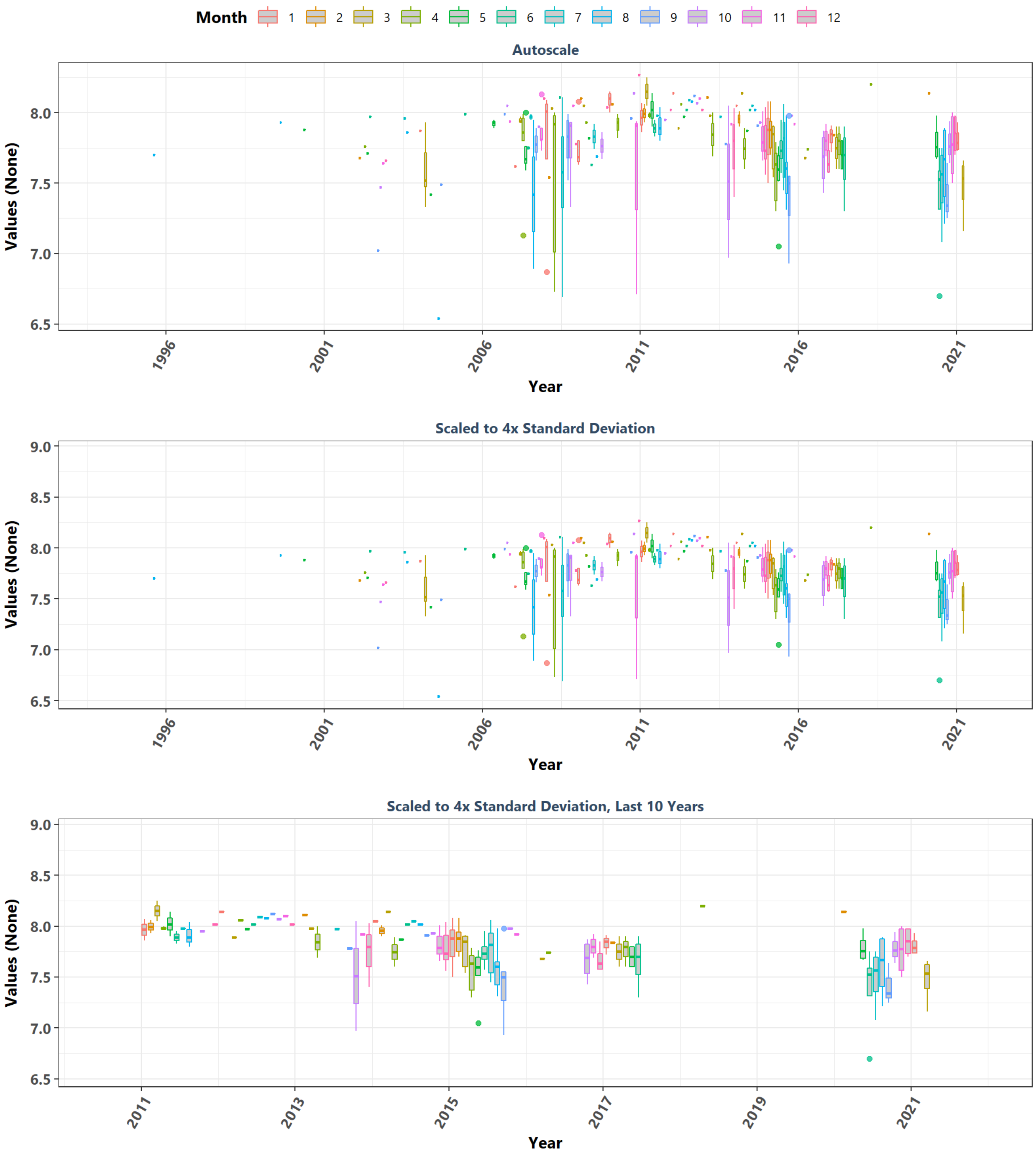


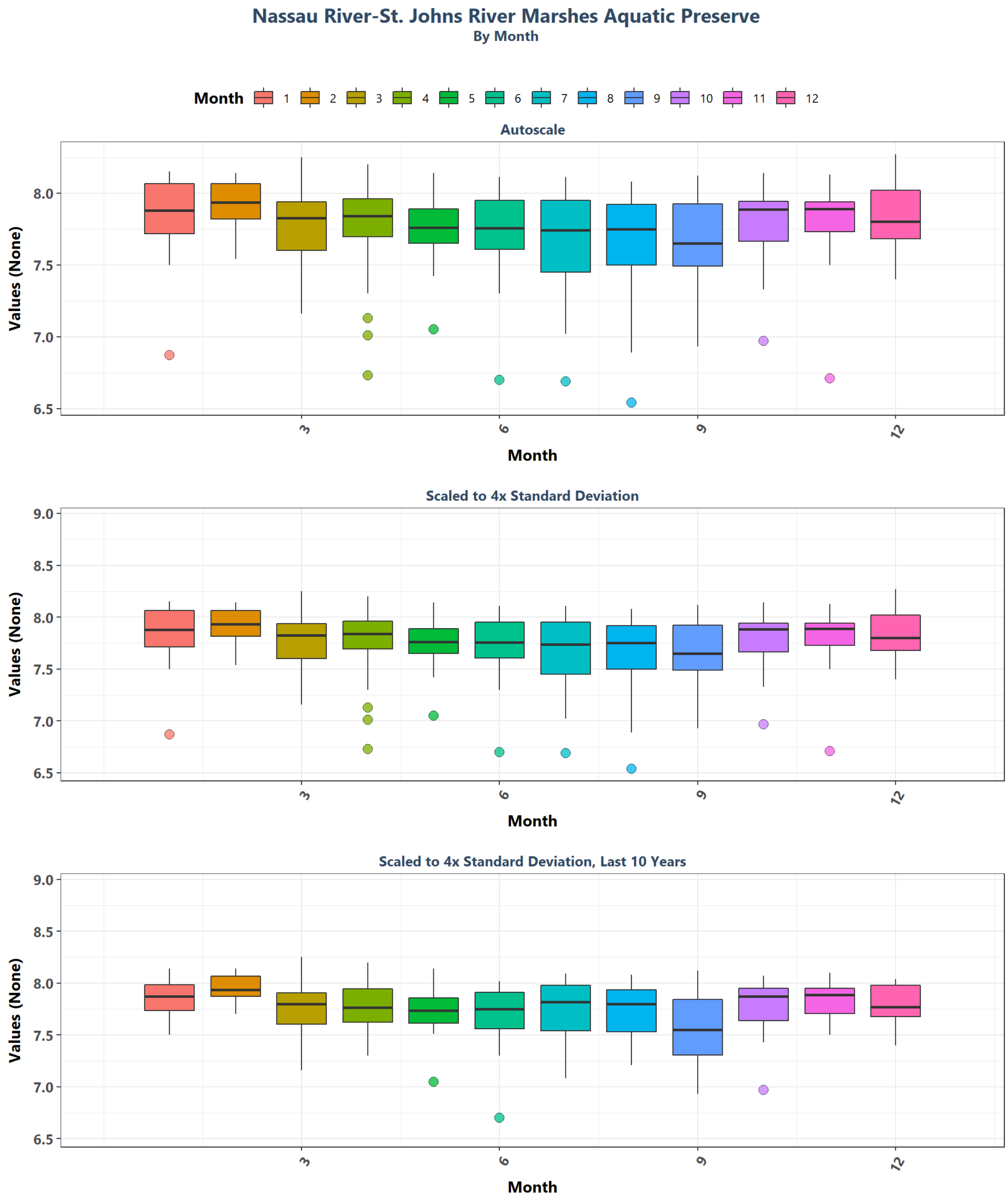

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Month
8.5
8.0
7.5
7.0
3
6
9
12


Nassau River-St. Johns River Marshes Aquatic Preserve
By Year
Autoscale
Values (None)
8.0
7.5
7.0
6.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
2011
2013
2015
2017
2019
2021
Year

Nassau River-St. Johns River Marshes Aquatic Preserve

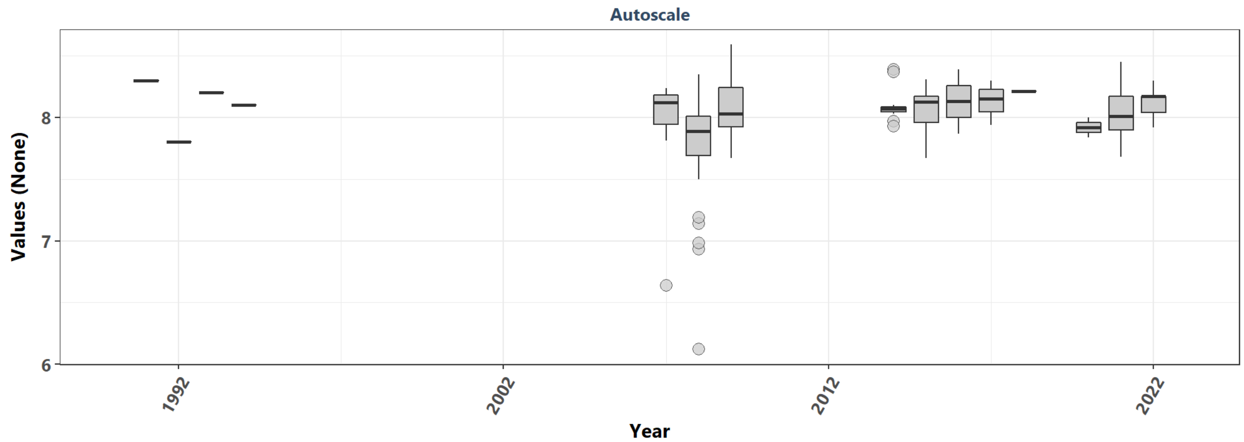
By Year & Month

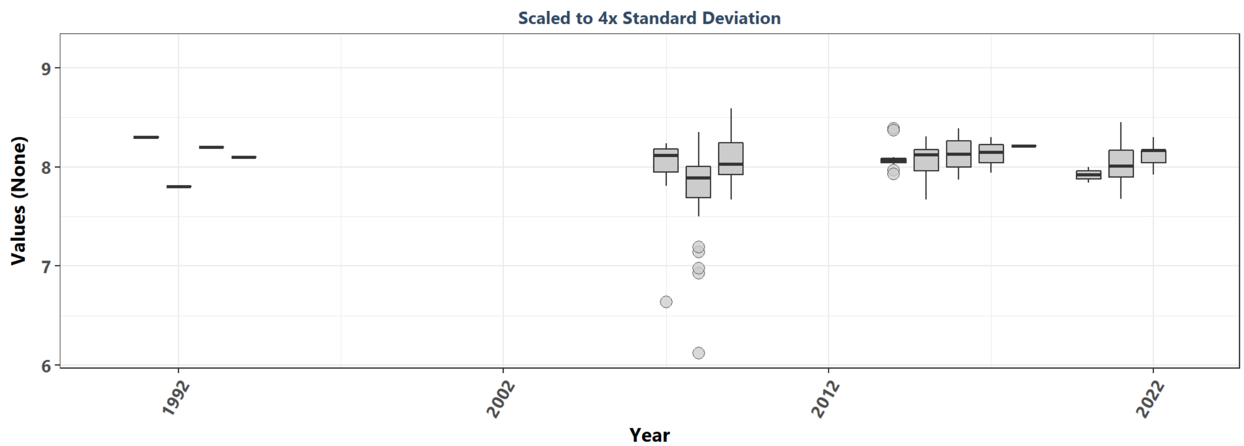


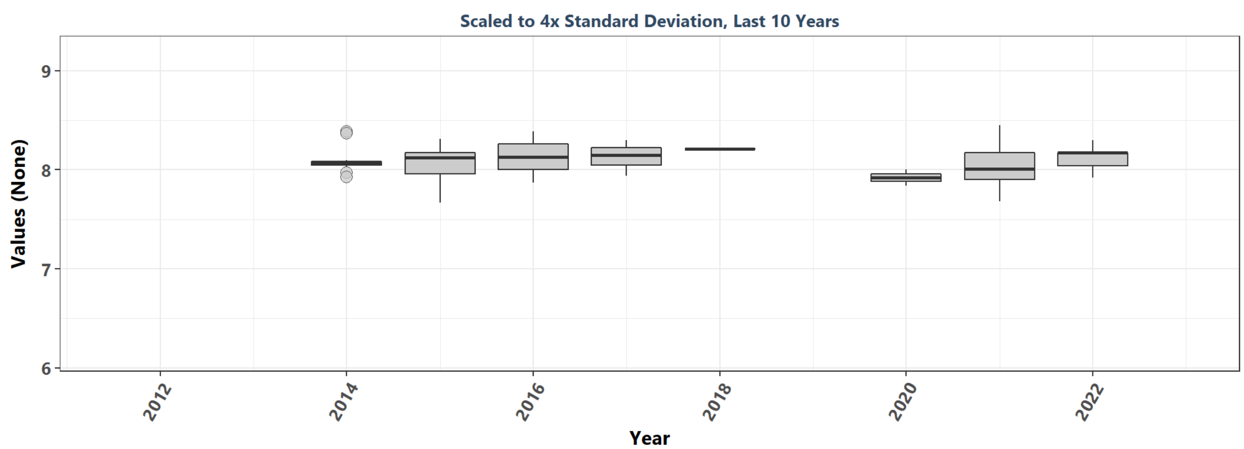


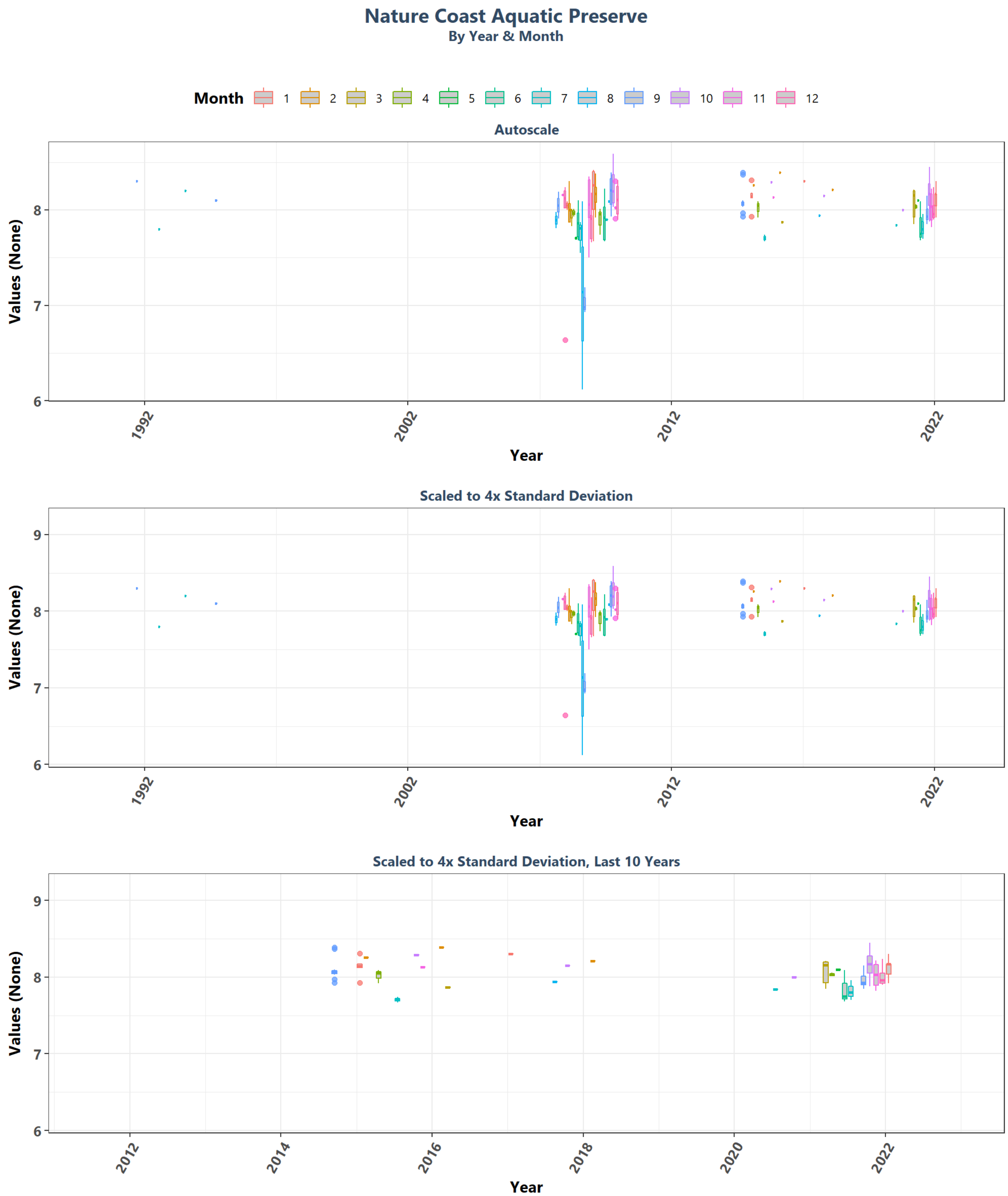
Nassau River-St. Johns River Marshes Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.0
7.5
7.0
6.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
3
6
9
12
Month

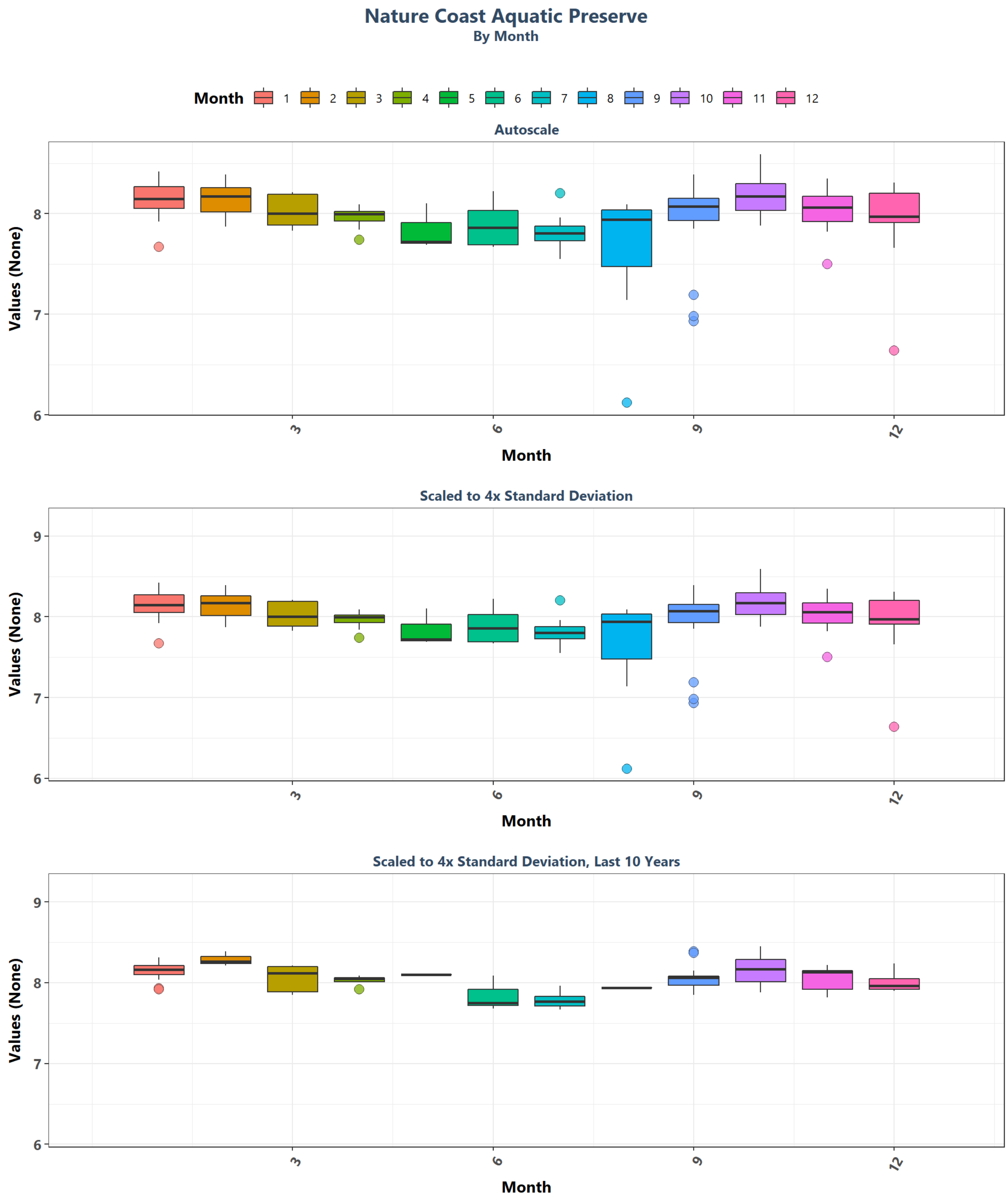
Nature Coast Aquatic Preserve
By Year





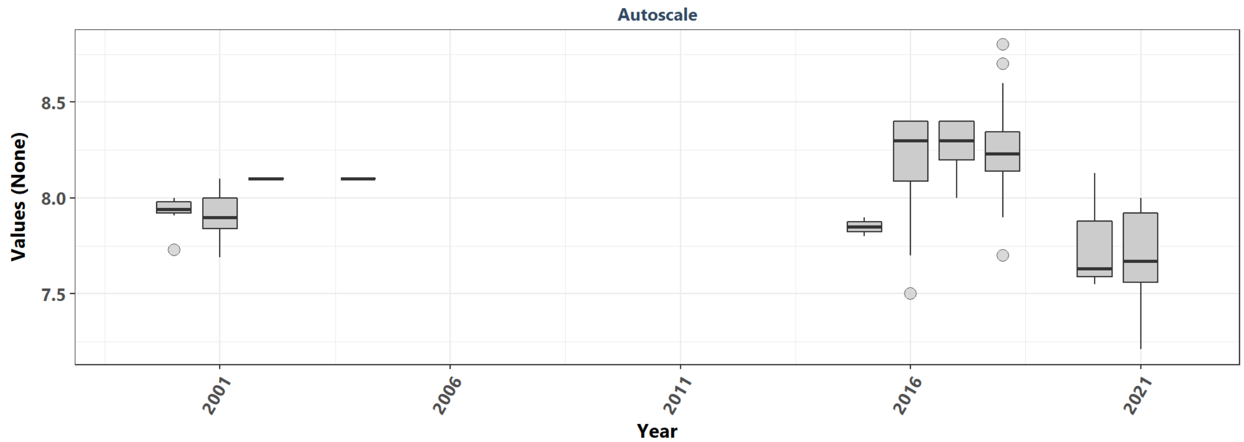


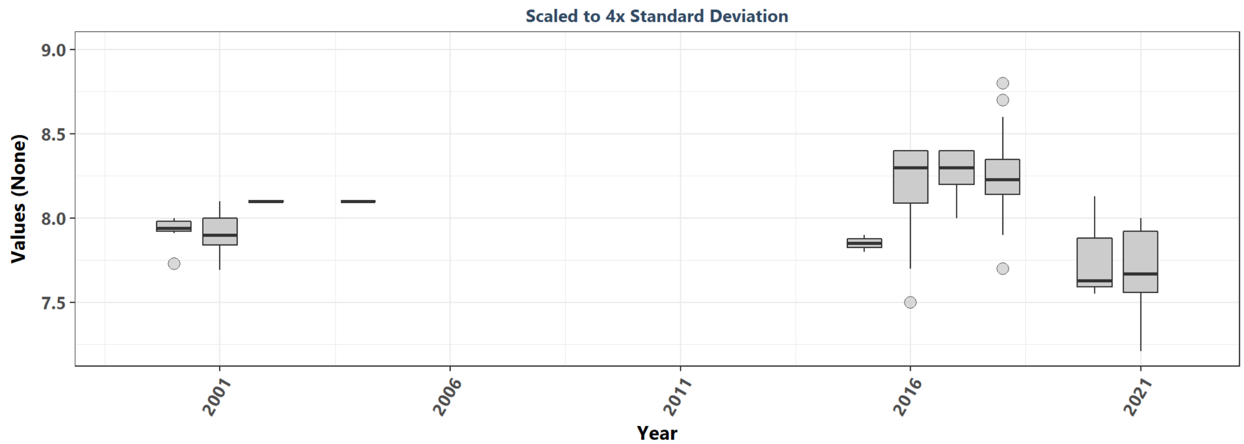

Nature Coast Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
6
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
2012
2014
2016
2018
2020
2022
Year

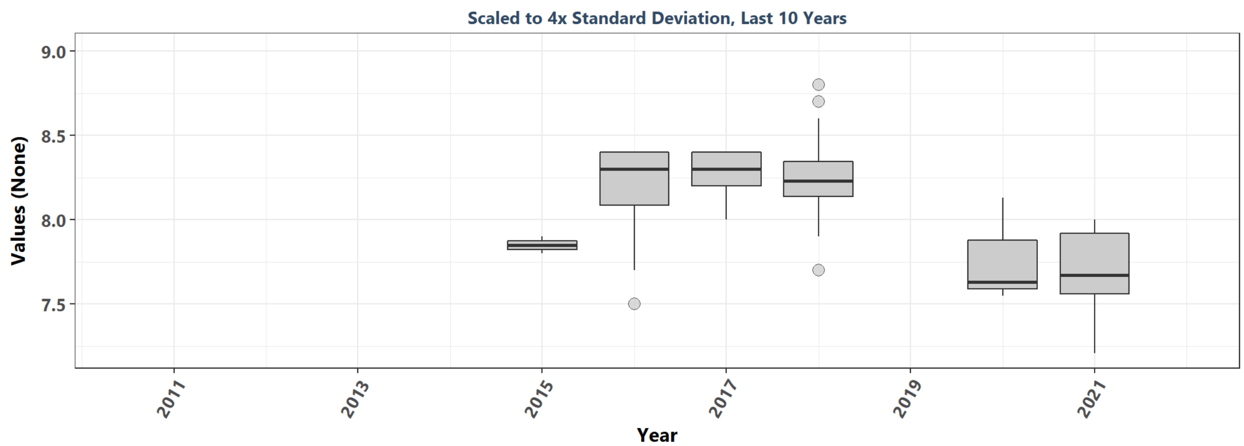


Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
3
6
9
12
Month

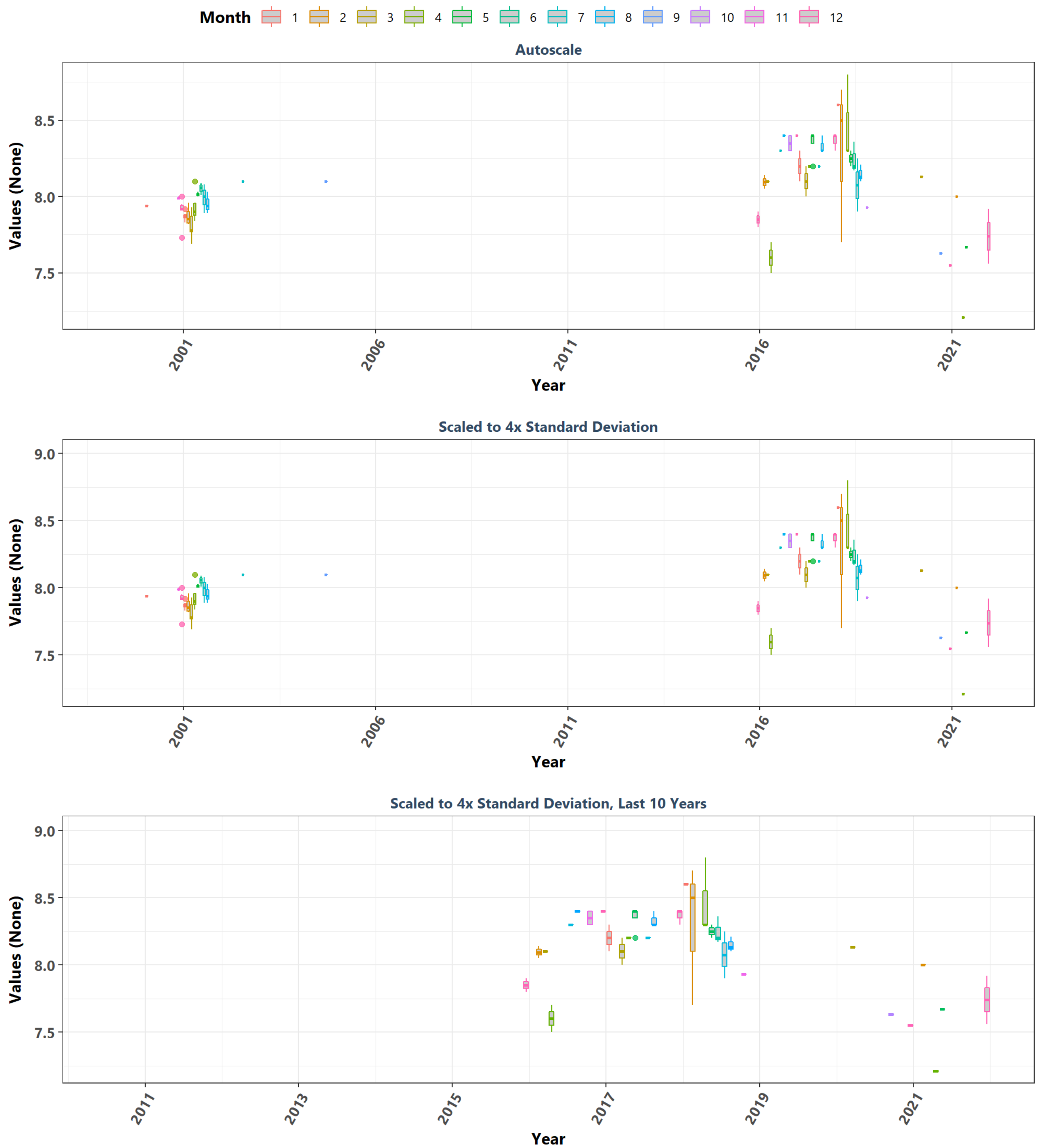
Pine Island Sound Aquatic Preserve
By Year

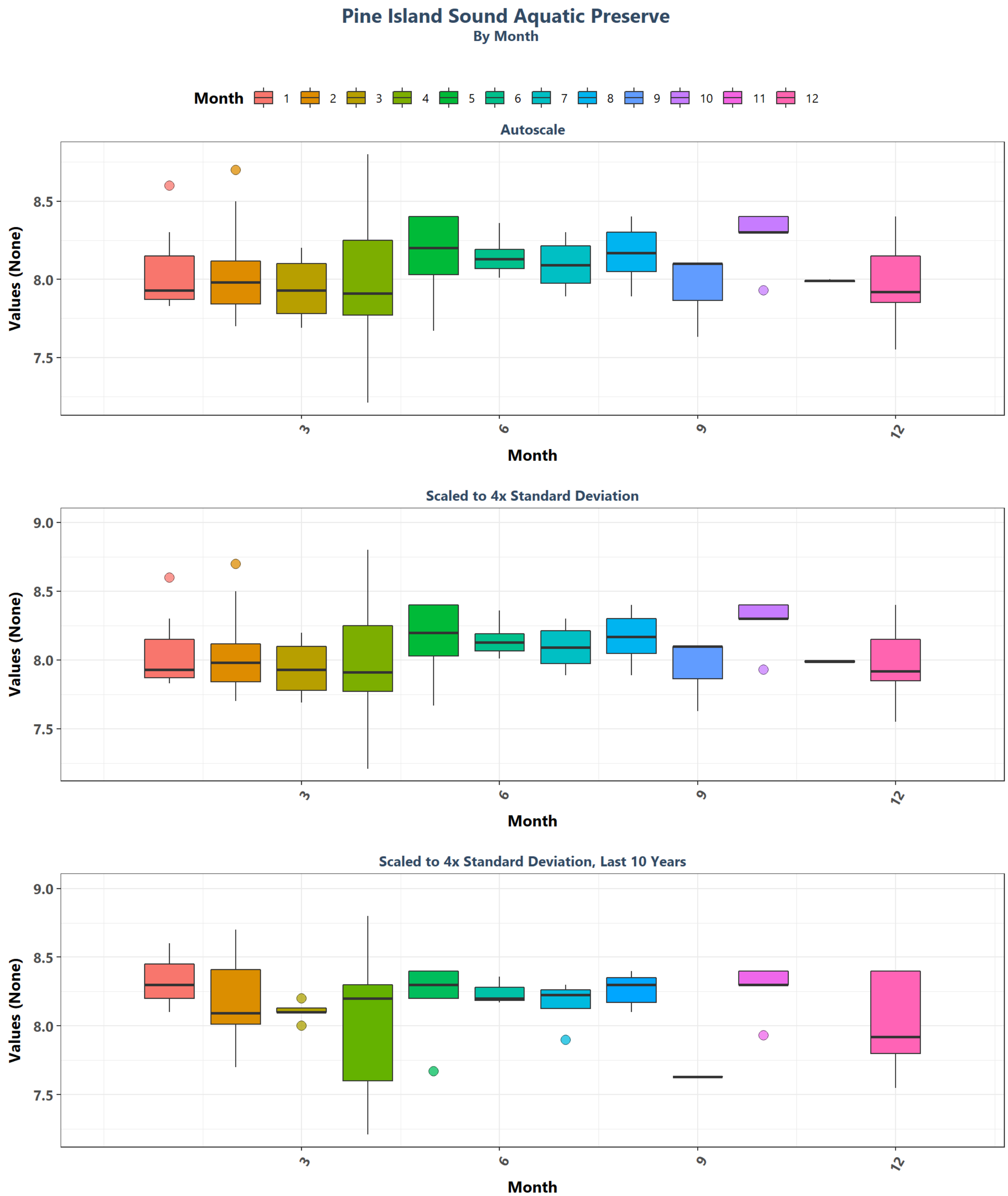






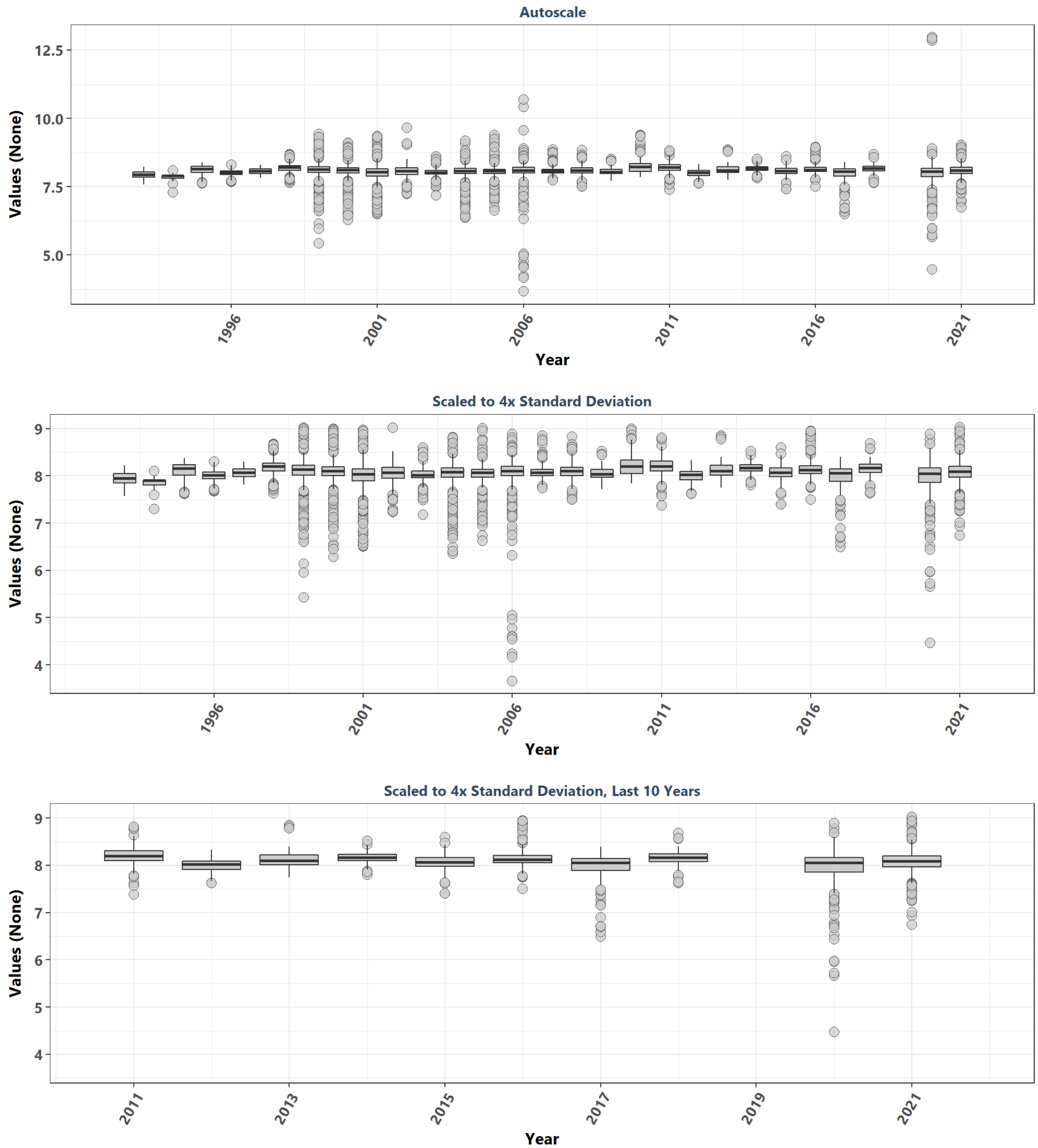
Pine Island Sound Aquatic Preserve
By Year & Month

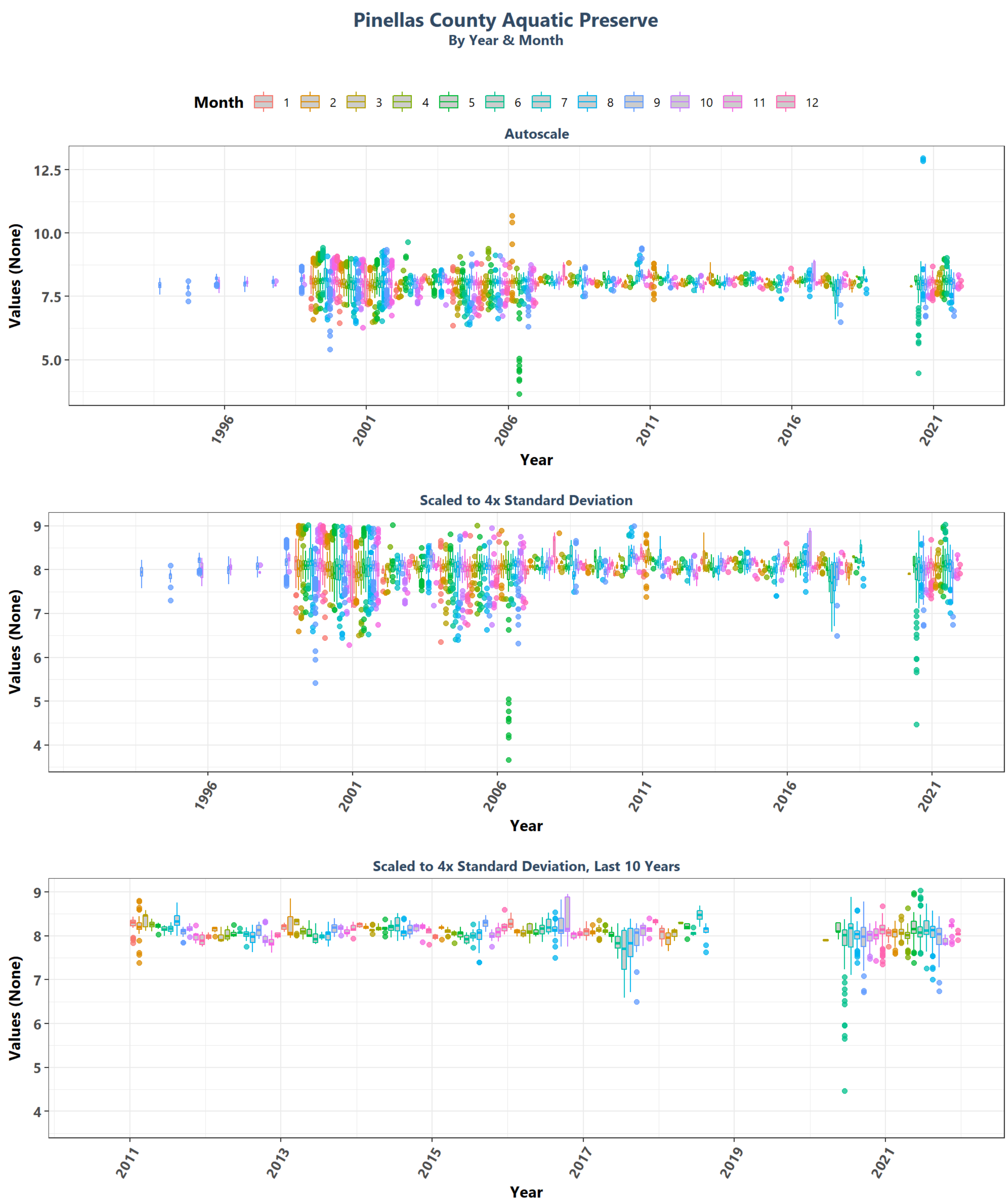




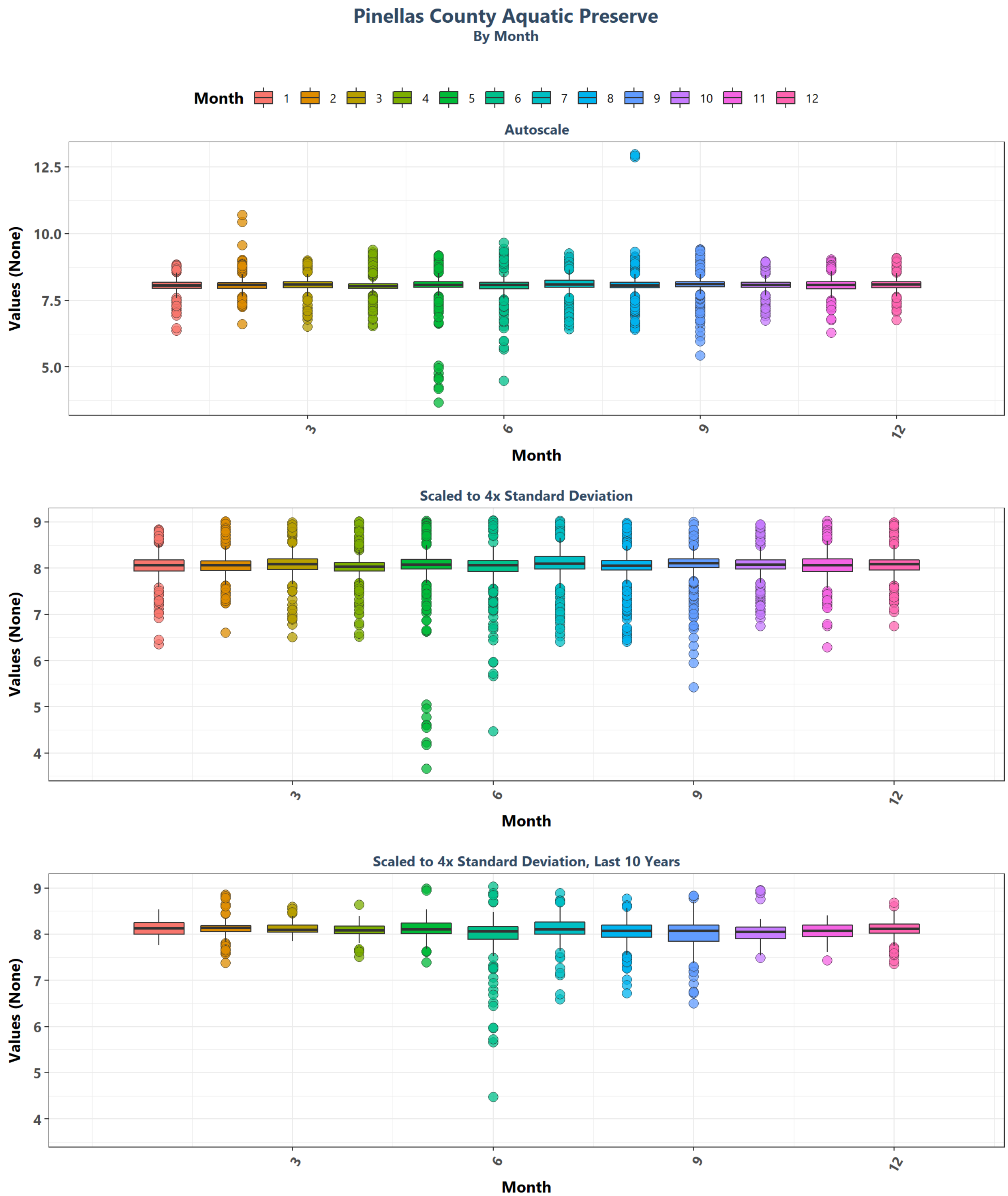
Pine Island Sound Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.5
8.0
7.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9.0
8.5
8.0
7.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9.0
8.5
8.0
7.5
3
6
9
12
Month

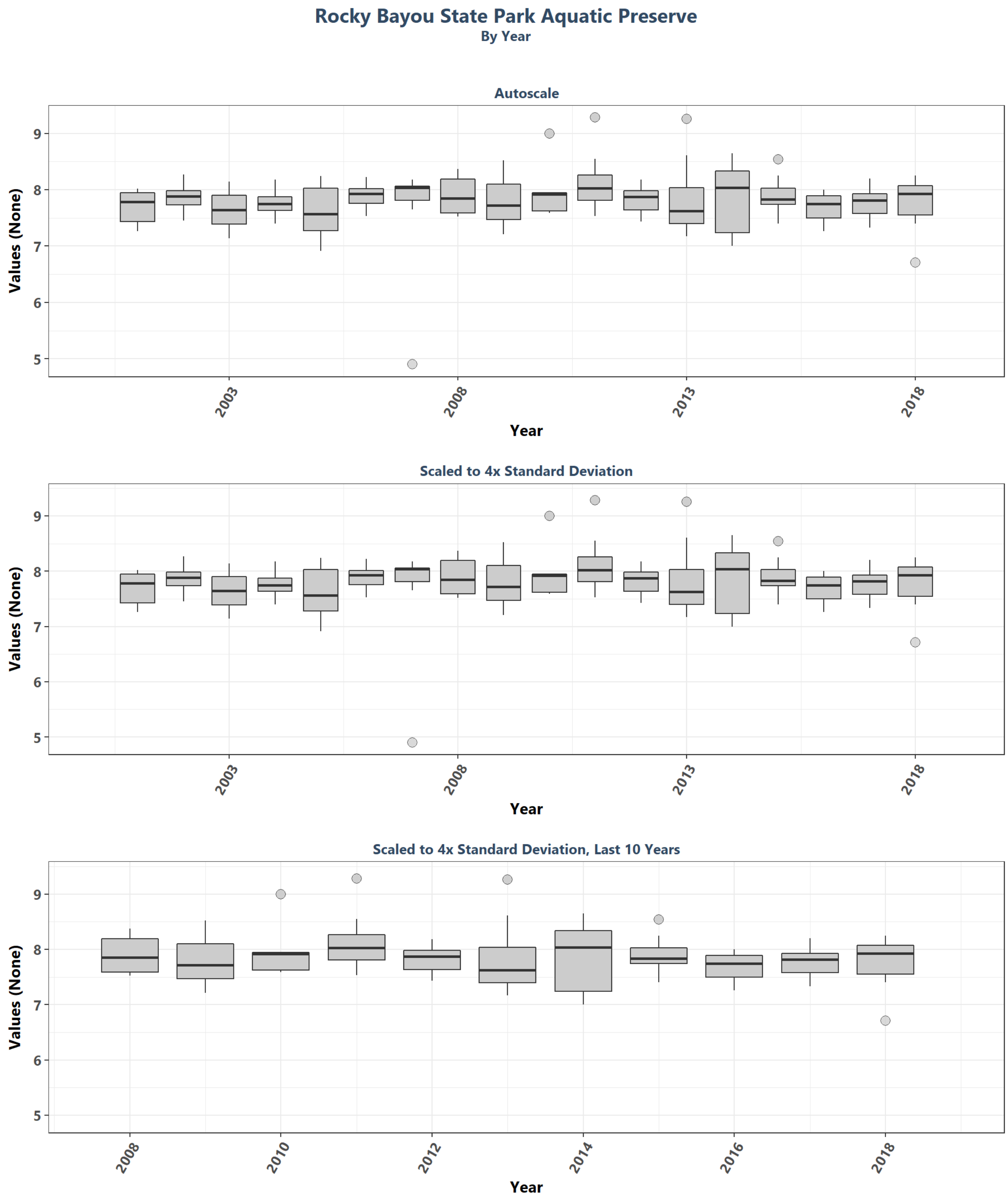
Pinellas County Aquatic Preserve
By Year



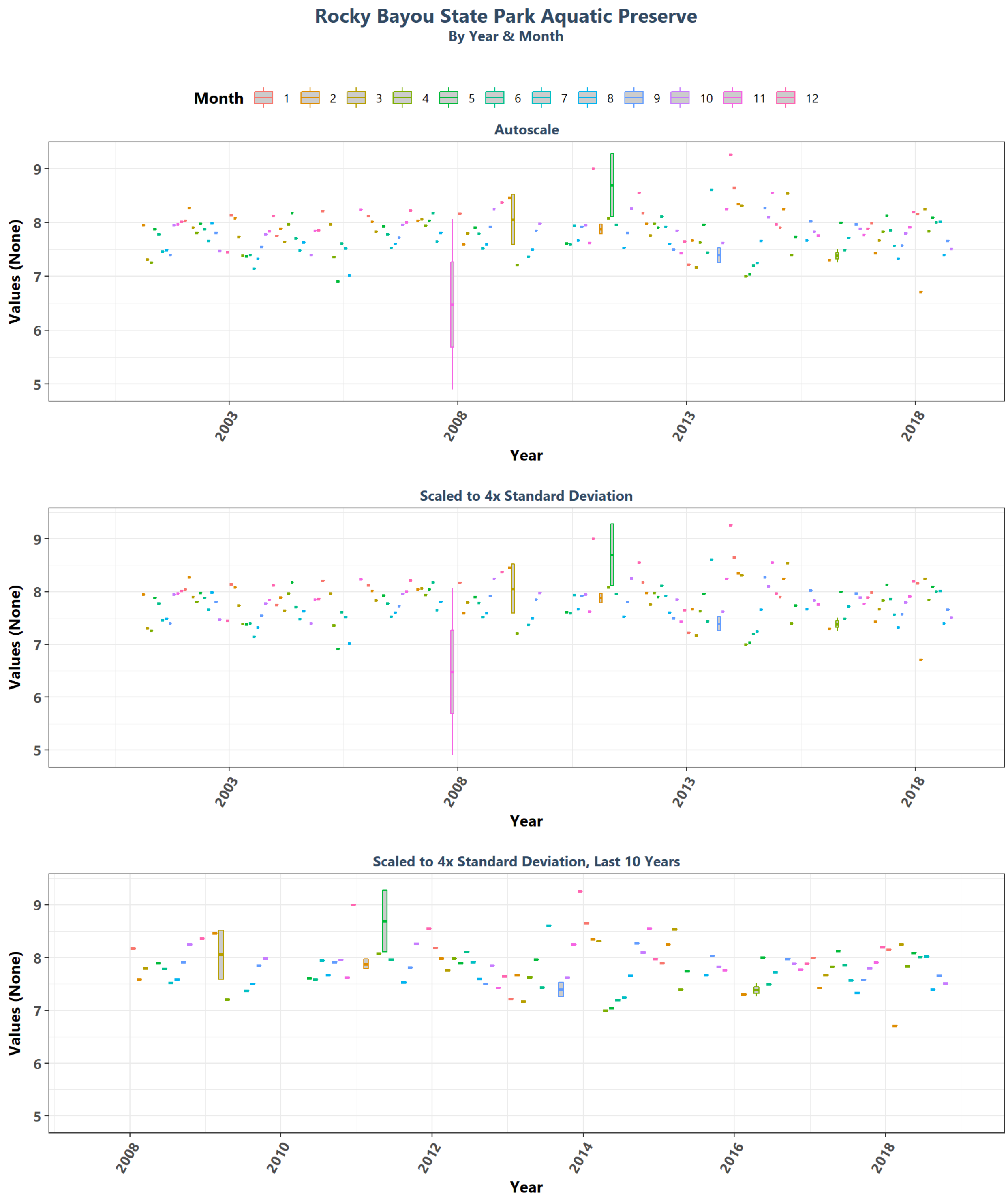


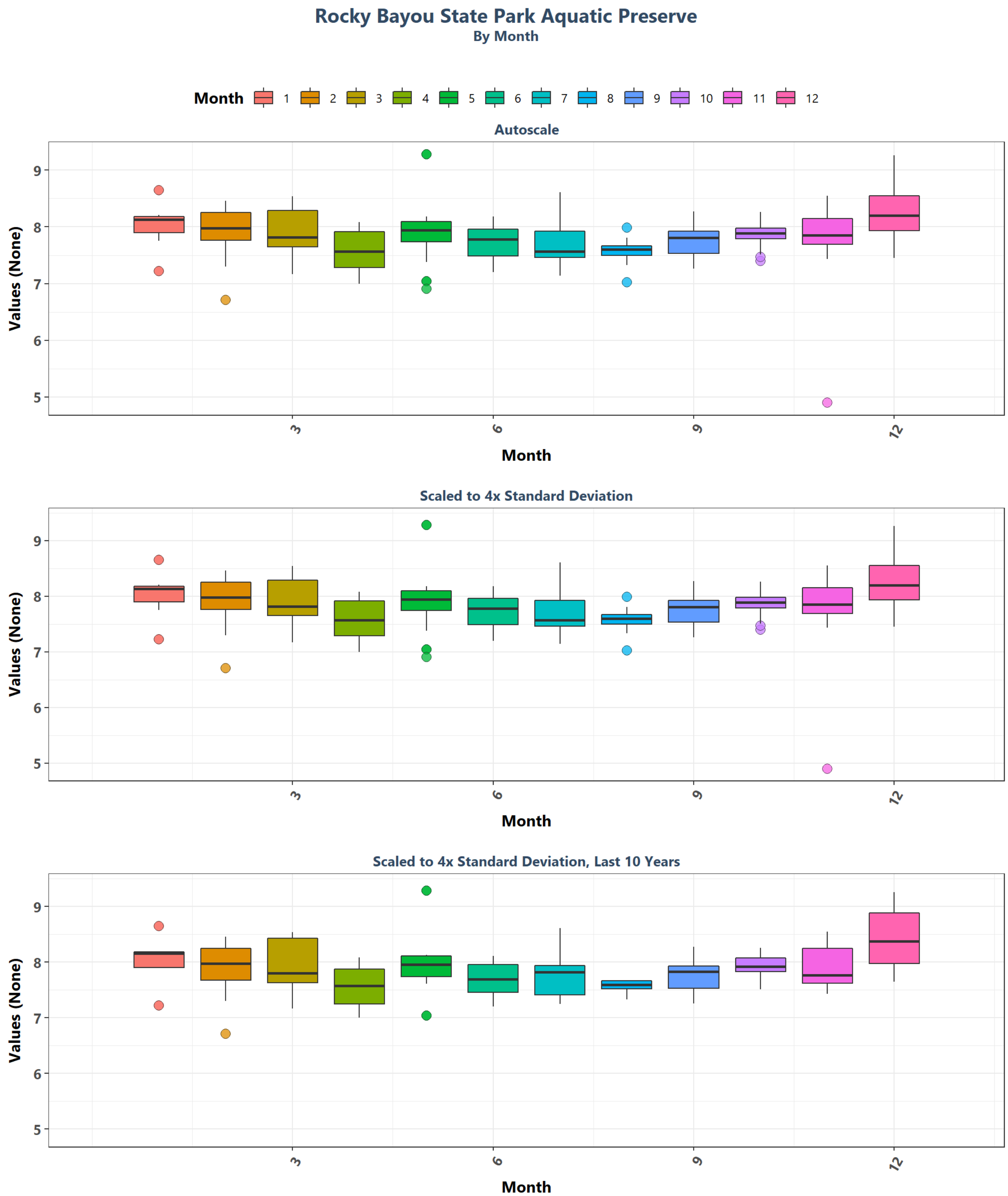
Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
12.5
10.0
7.5
5.0
Year
1996
2001
2006
2011
2016
2021
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
5
4
Year
1996
2001
2006
2011
2016
2021
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
5
4
Year
2011
2013
2015
2017
2019
2021


Pinellas County Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
12.5
10.0
7.5
5.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
5
4
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
5
4
3
6
9
12
Month



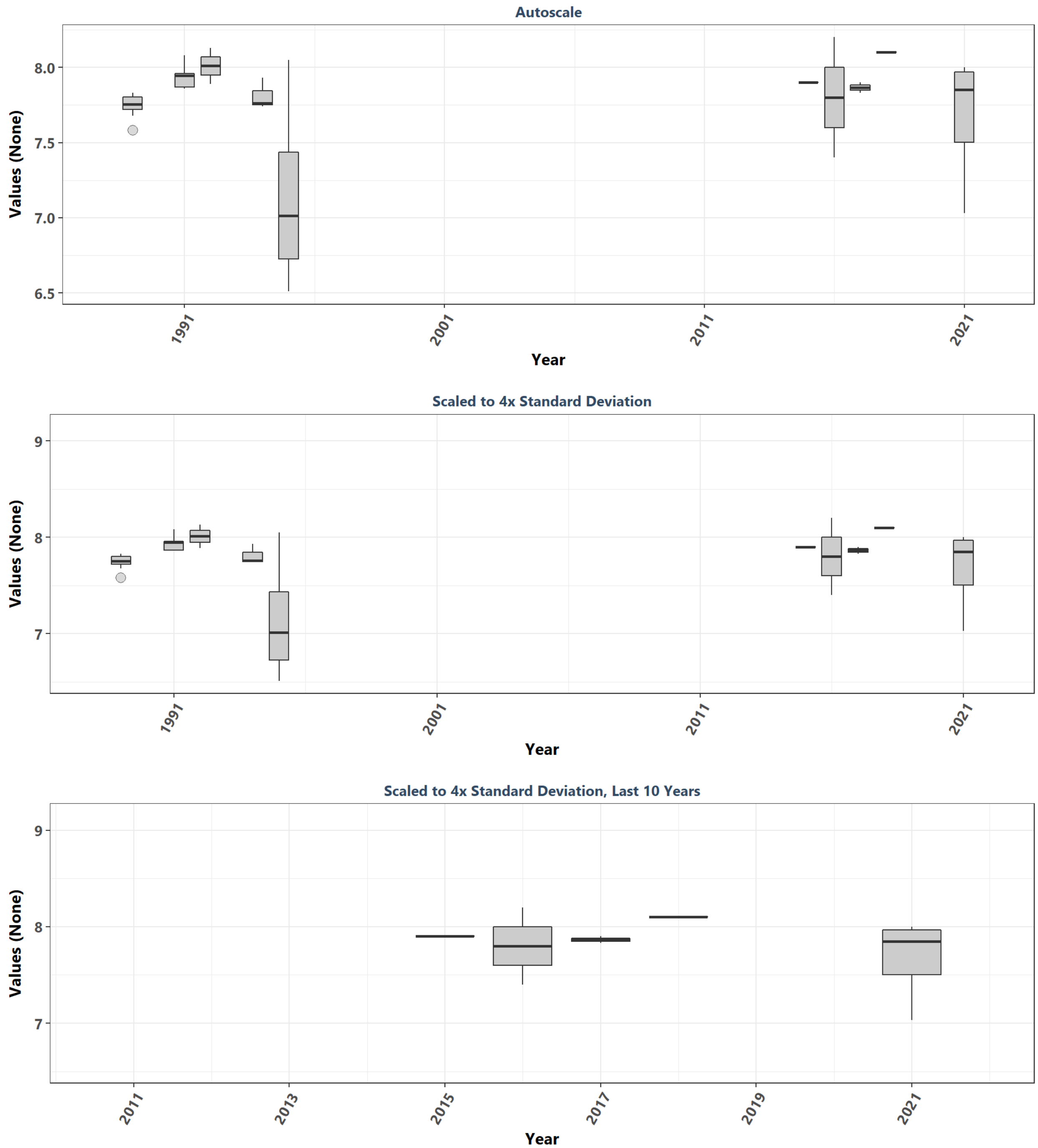
Rocky Bayou State Park Aquatic Preserve
By Year
Autoscale
Values (None)
9
8
7
6
5
2003
2008
2013
2018
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
6
5
2003
2008
2013
2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
6
5
2008
2010
2012
2014
2016
2018
Year

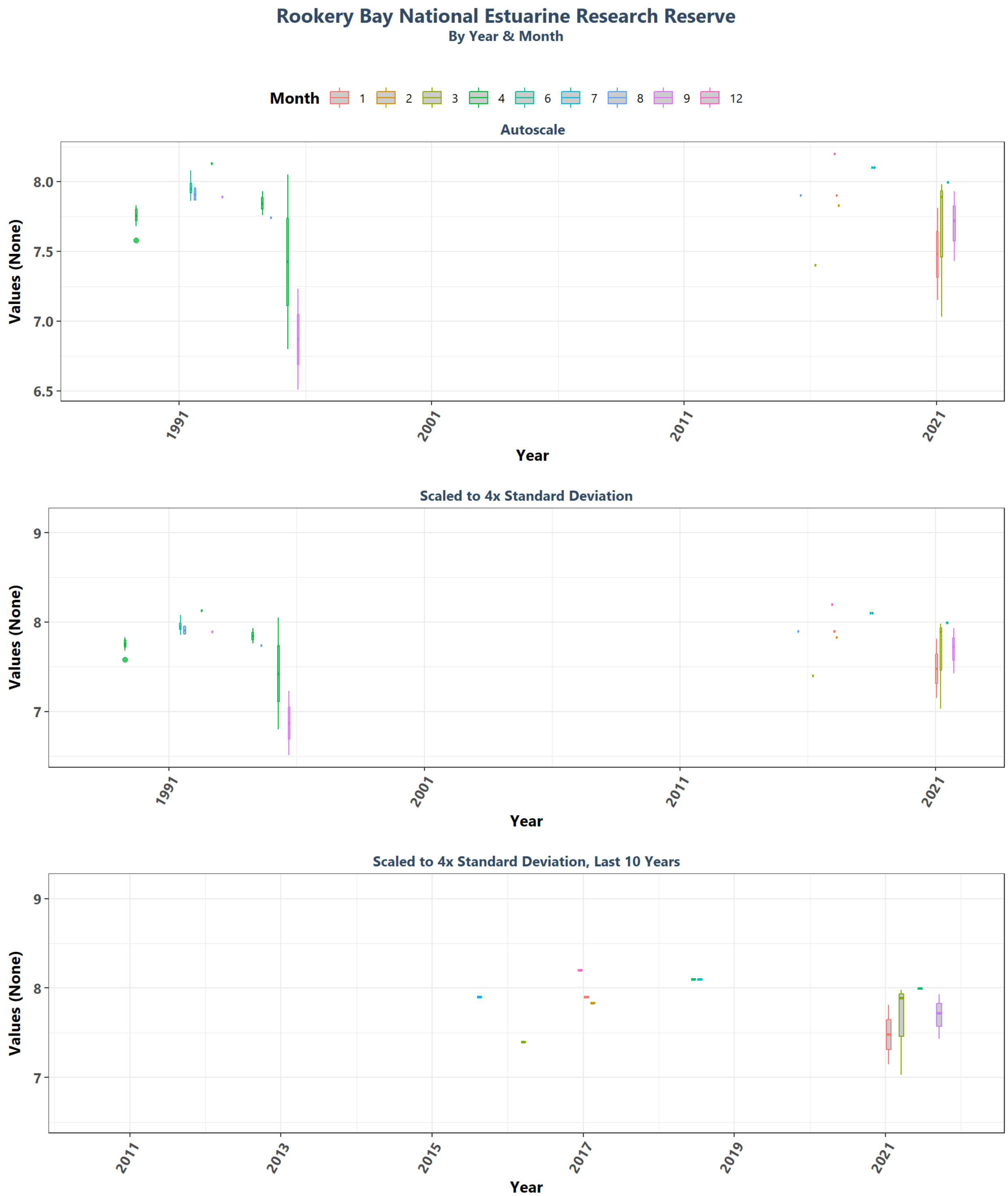

Rocky Bayou State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year
2003 2008 2013 2018
2008 2010 2012 2014 2016 2018
5 6 7 8 9


Rocky Bayou State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
9
8
7
6
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Month

Rookery Bay National Estuarine Research Reserve

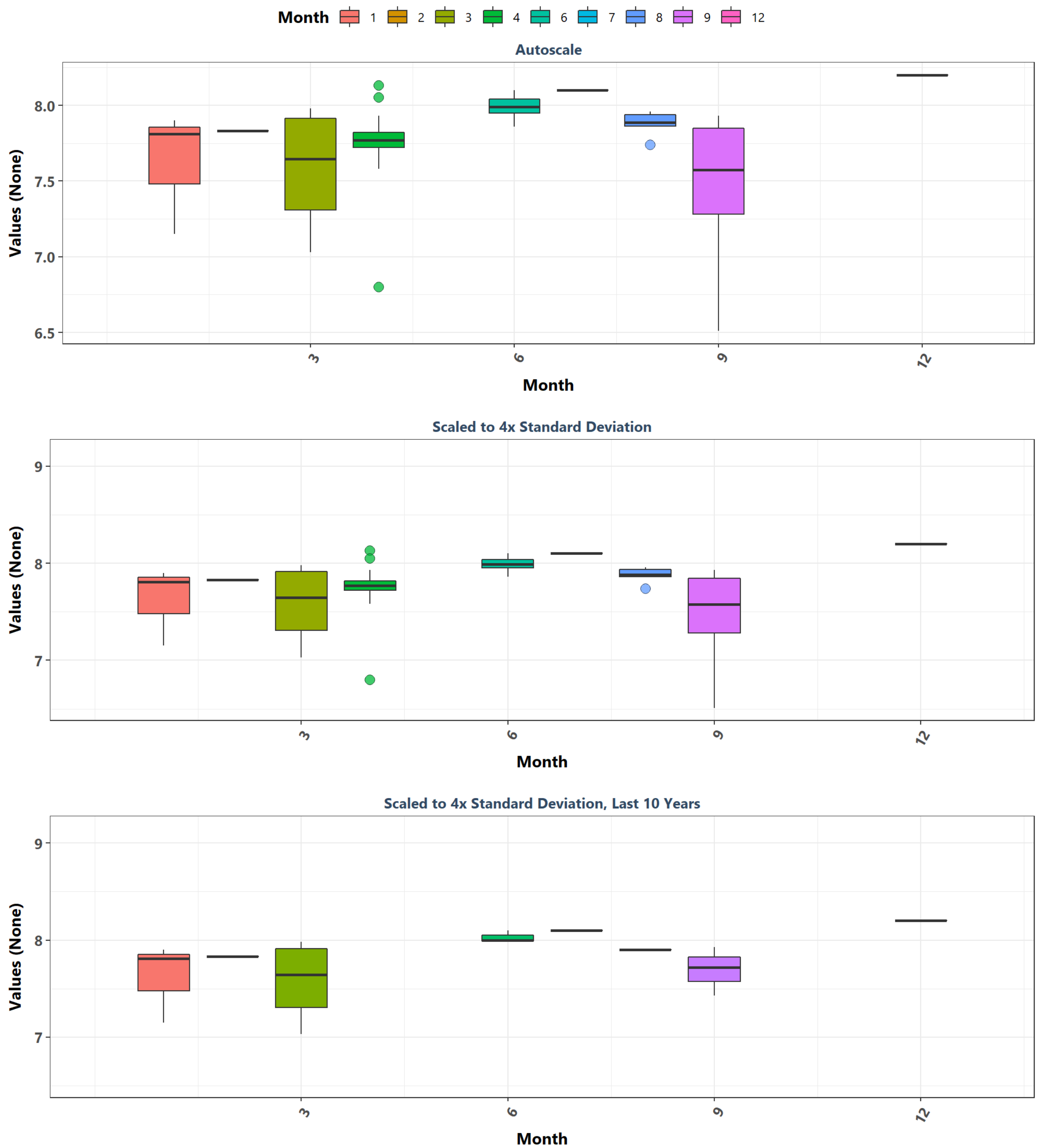
By Year



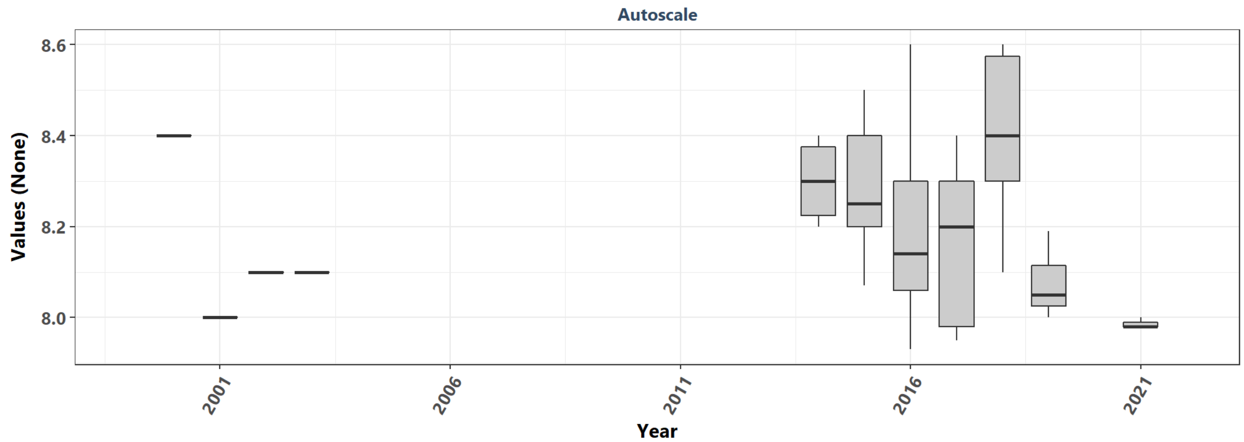


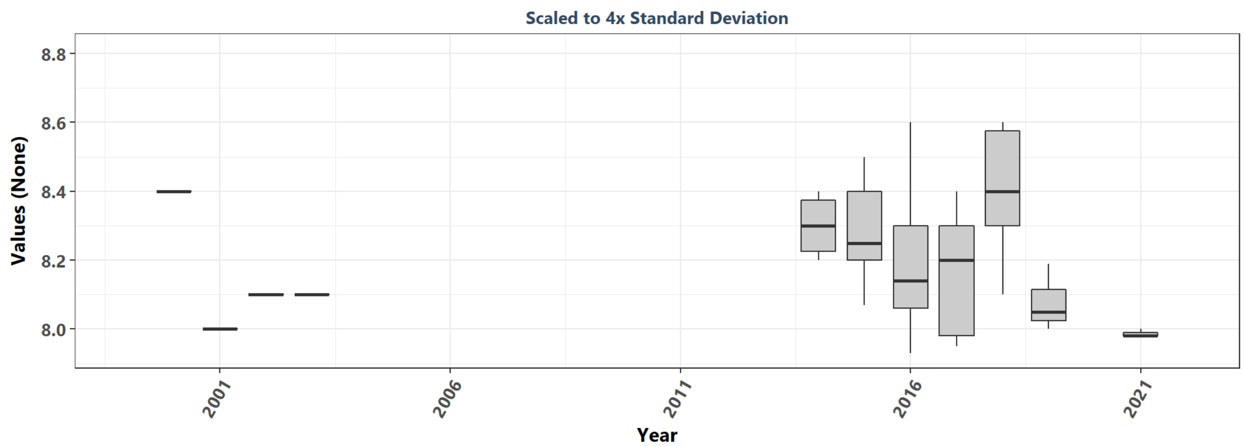
Rookery Bay National Estuarine Research Reserve
By Year & Month
Month
1
2
3
4
6
7
8
9
12
Autoscale
Values (None)
8.0
7.5
7.0
6.5
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (None)
9
8
7
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
9
8
7
2011
2013
2015
2017
2019
2021
Year

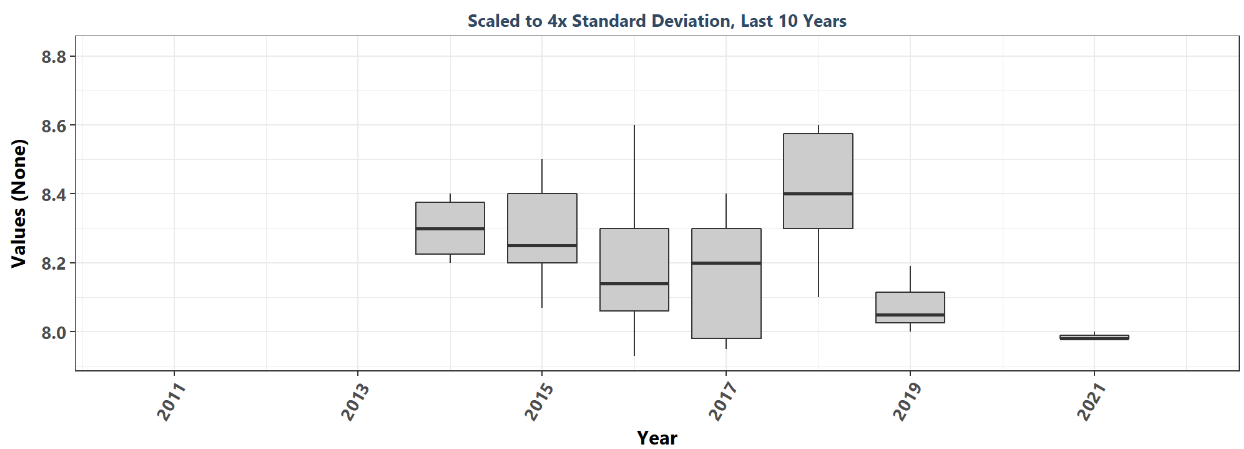
Rookery Bay National Estuarine Research Reserve
By Month

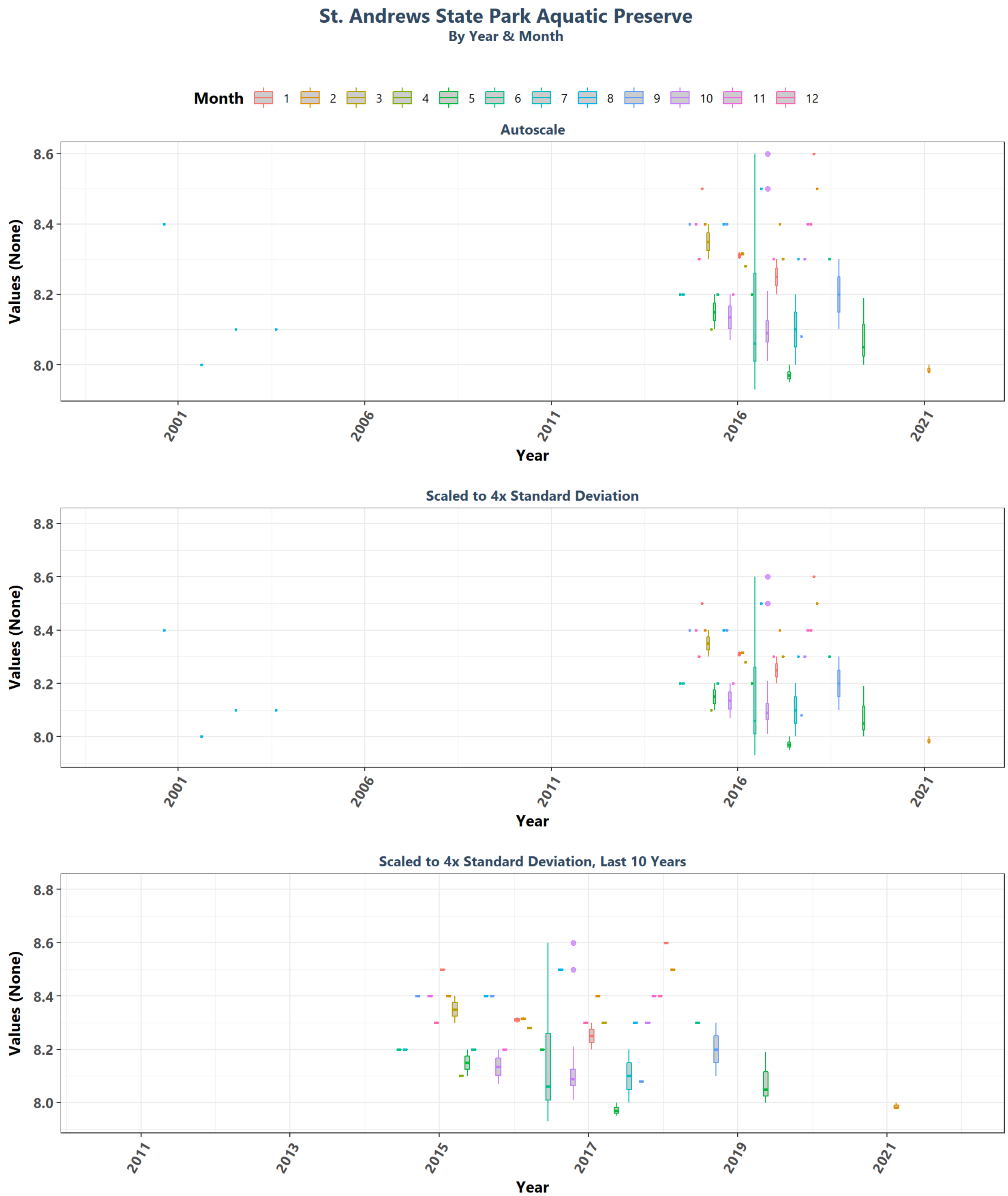


St. Andrews State Park Aquatic Preserve
By Year

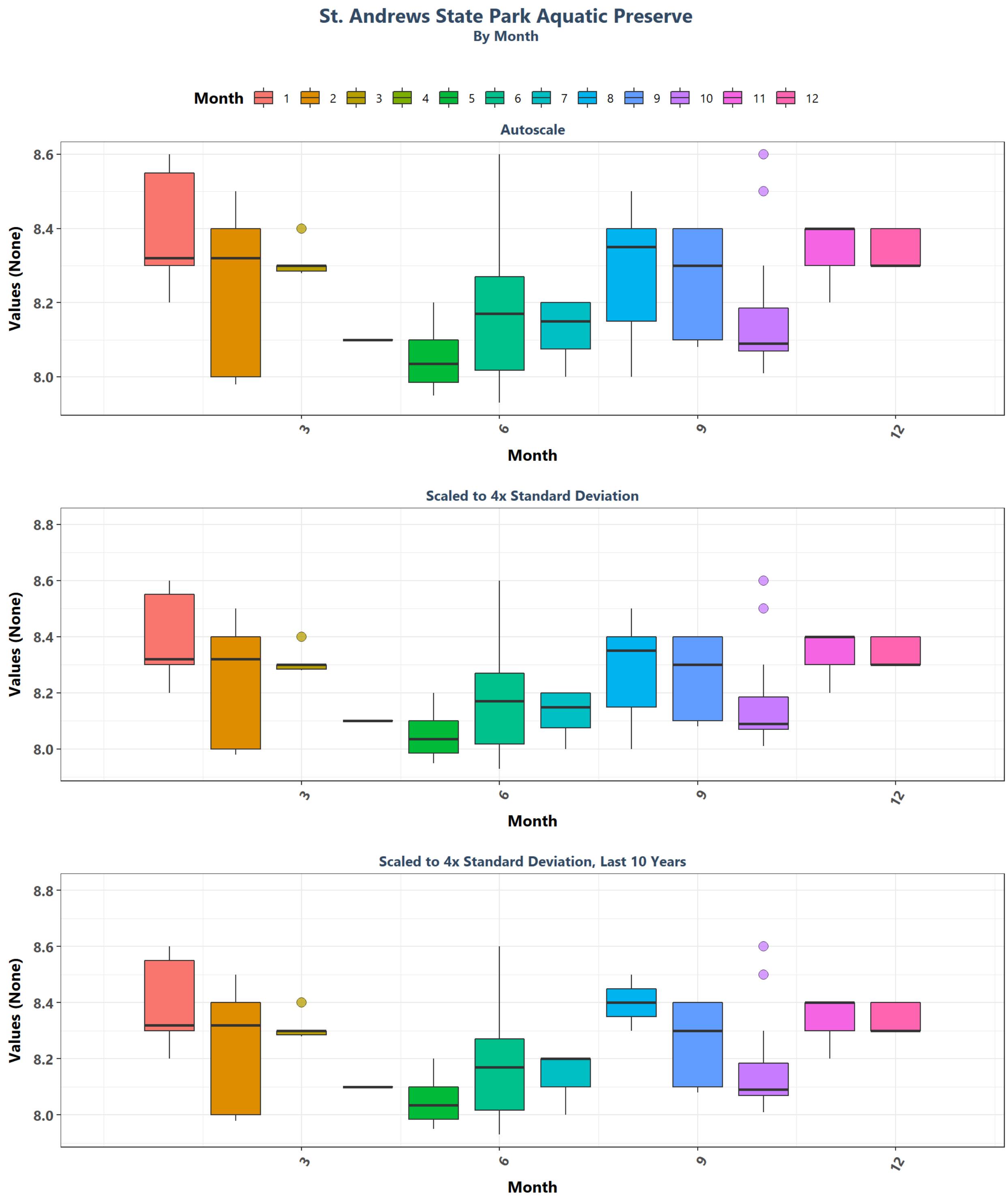




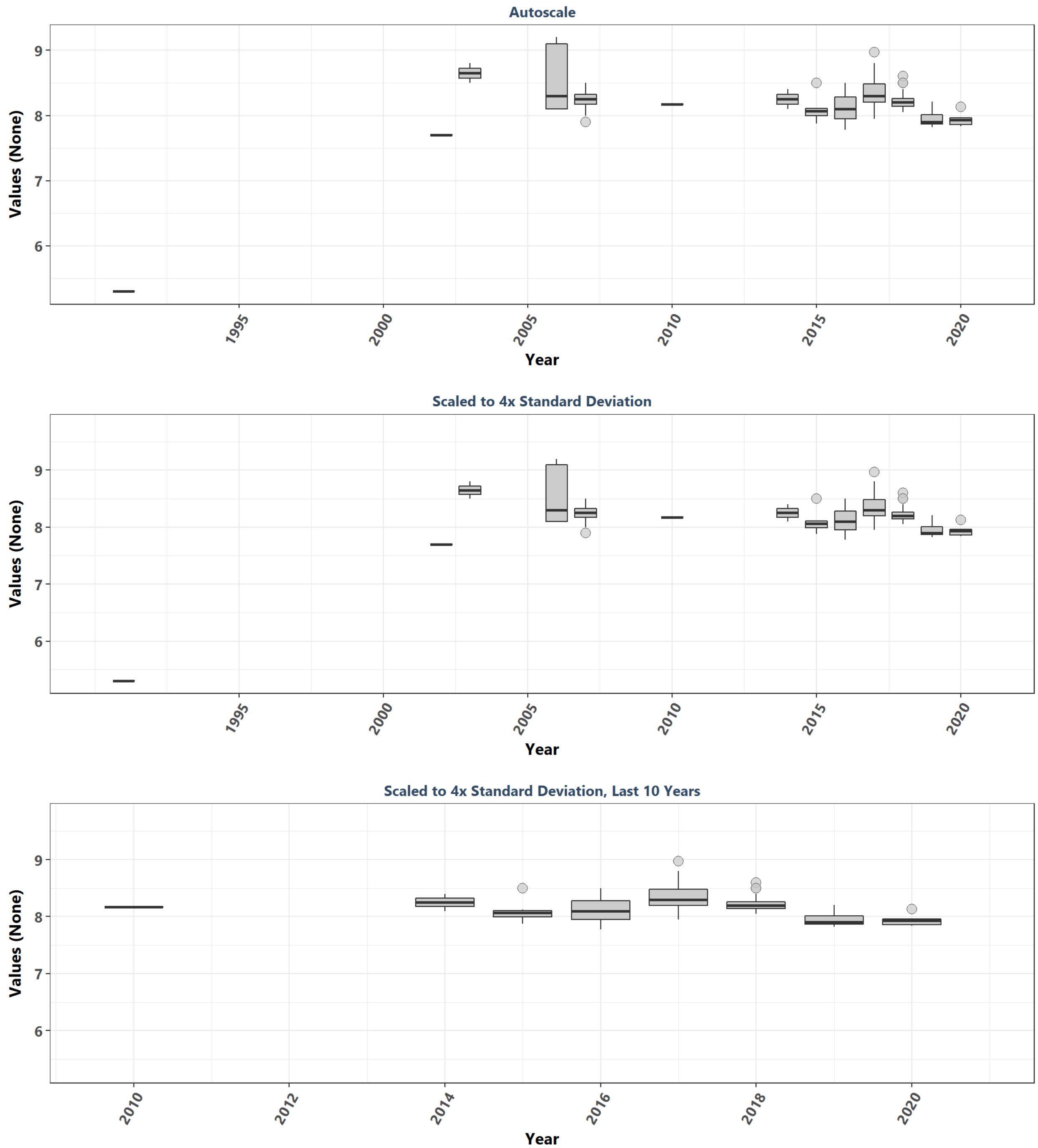


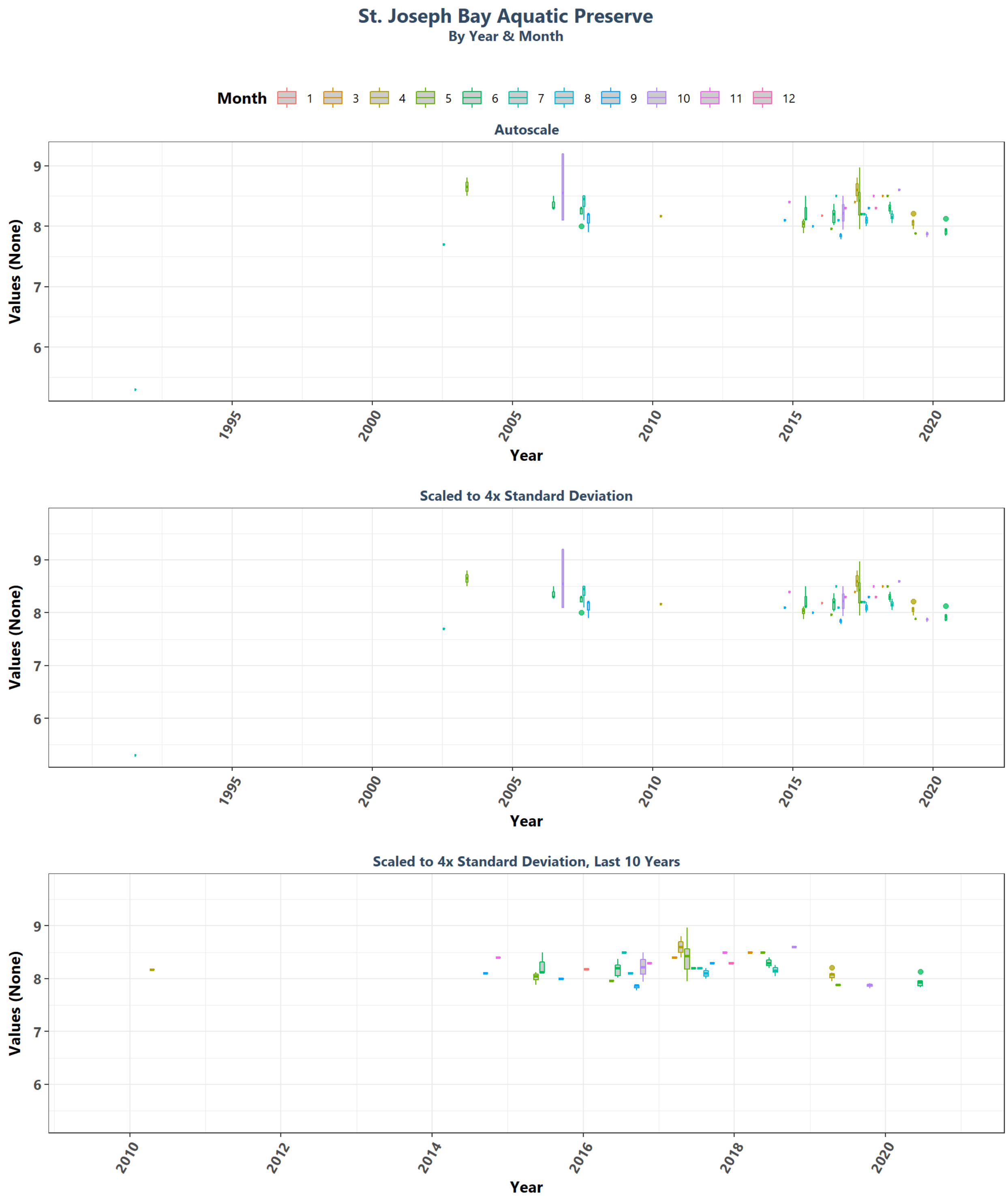


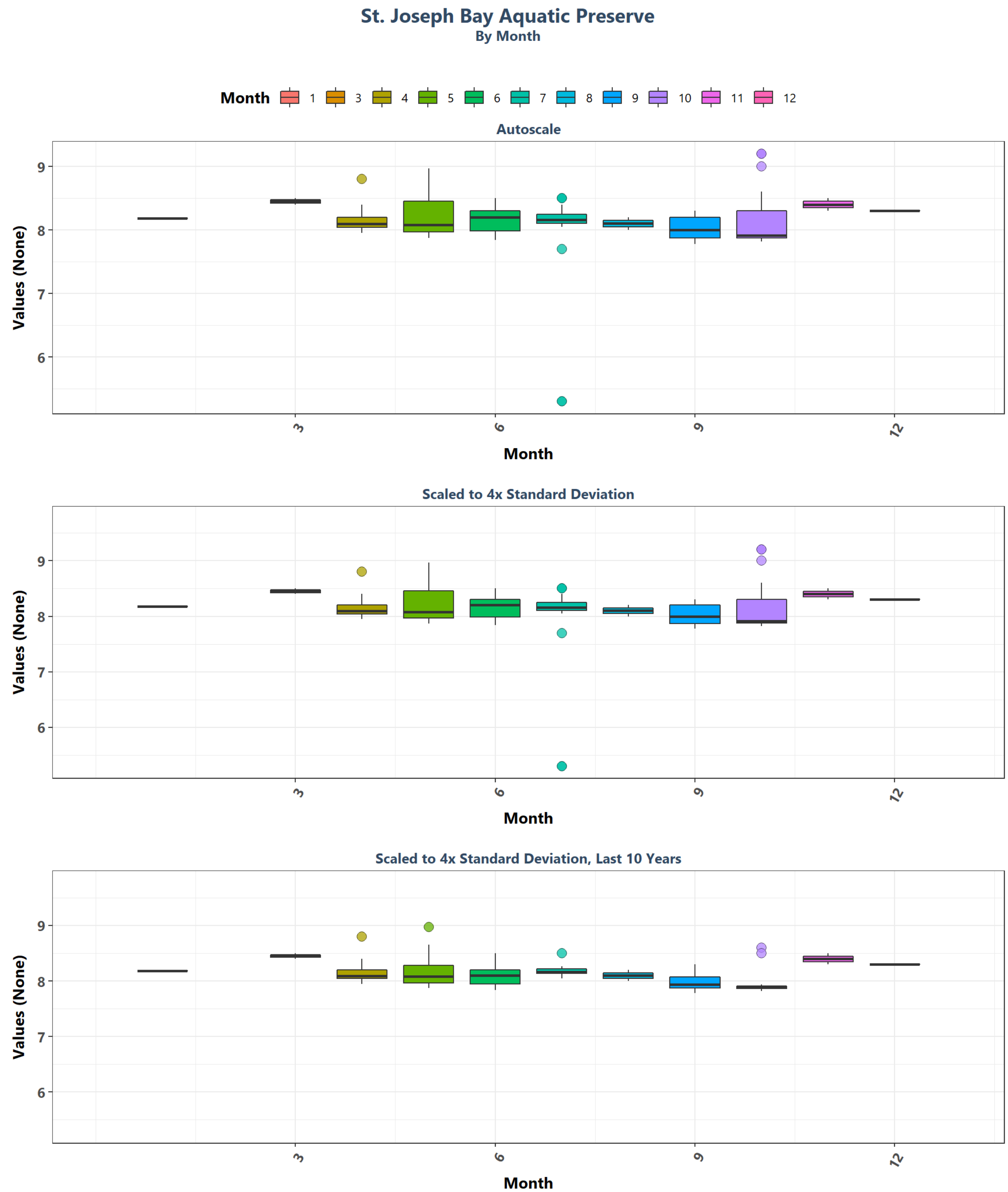
St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.6
8.4
8.2
8.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
8.8
8.6
8.4
8.2
8.0
Scaled to 4x Standard Deviation, Last 10 Years
2011
2013
2015
2017
2019
2021


St. Andrews State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.6
8.4
8.2
8.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
8.8
8.6
8.4
8.2
8.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8.8
8.6
8.4
8.2
8.0
3
6
9
12
Month

St. Joseph Bay Aquatic Preserve
By Year

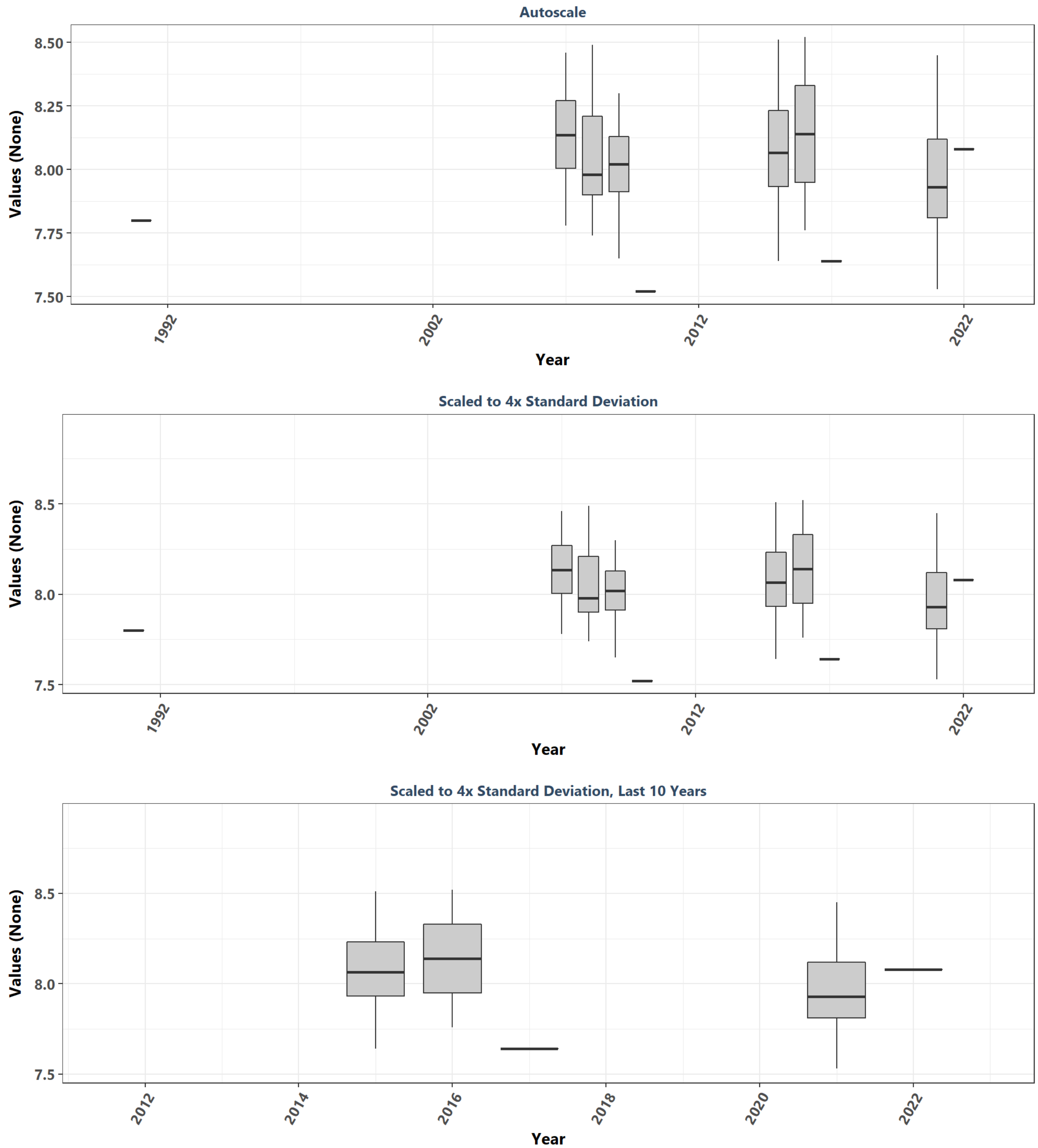


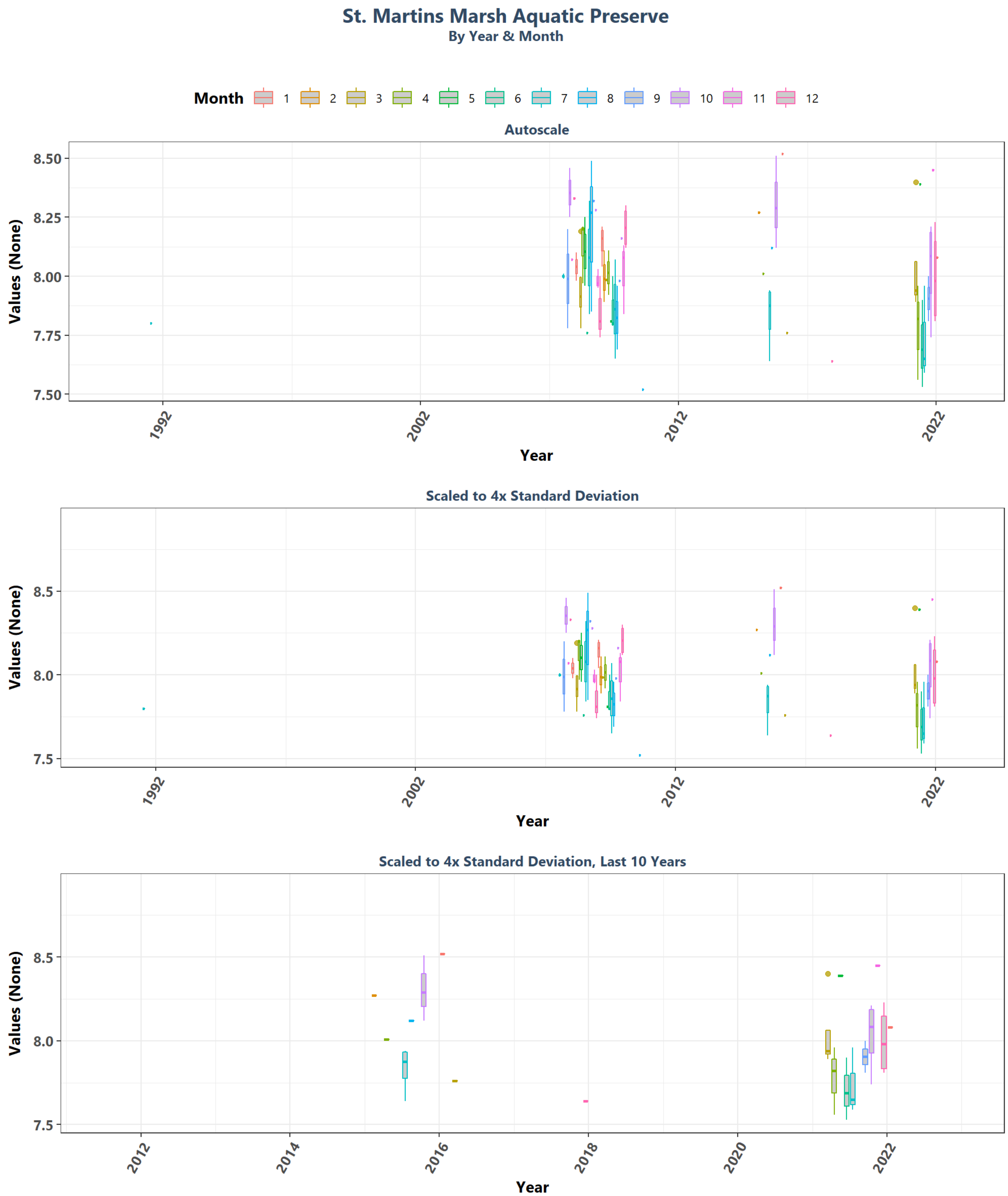

St. Joseph Bay Aquatic Preserve
By Year & Month
Month 1 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
Year
Scaled to 4x Standard Deviation
Values (None)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Year



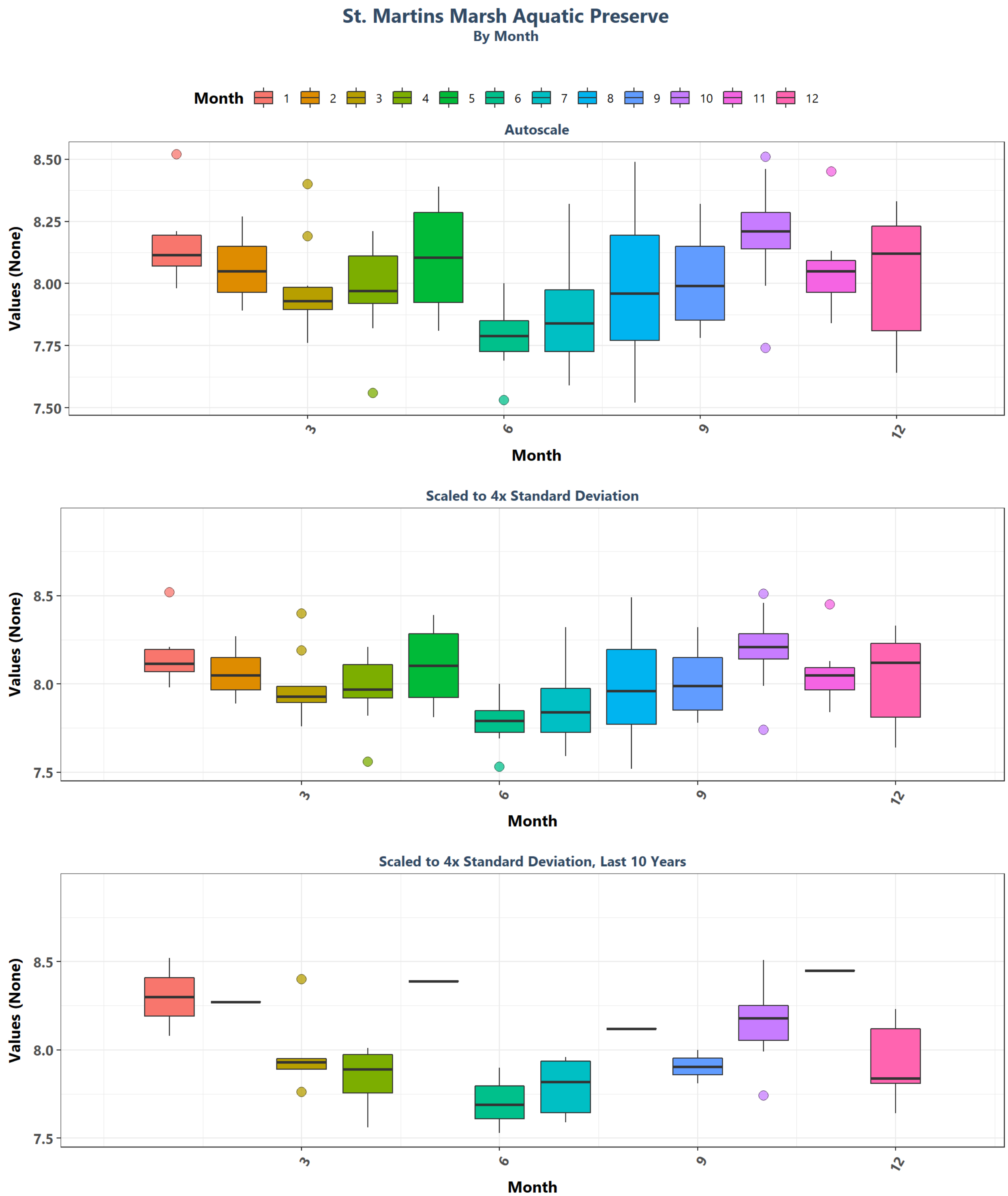
St. Joseph Bay Aquatic Preserve
By Month
Month 1 3 4 5 6 7 8 9 10 11 12
Autoscale
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
Month
3
6
9
12

St. Martins Marsh Aquatic Preserve
By Year

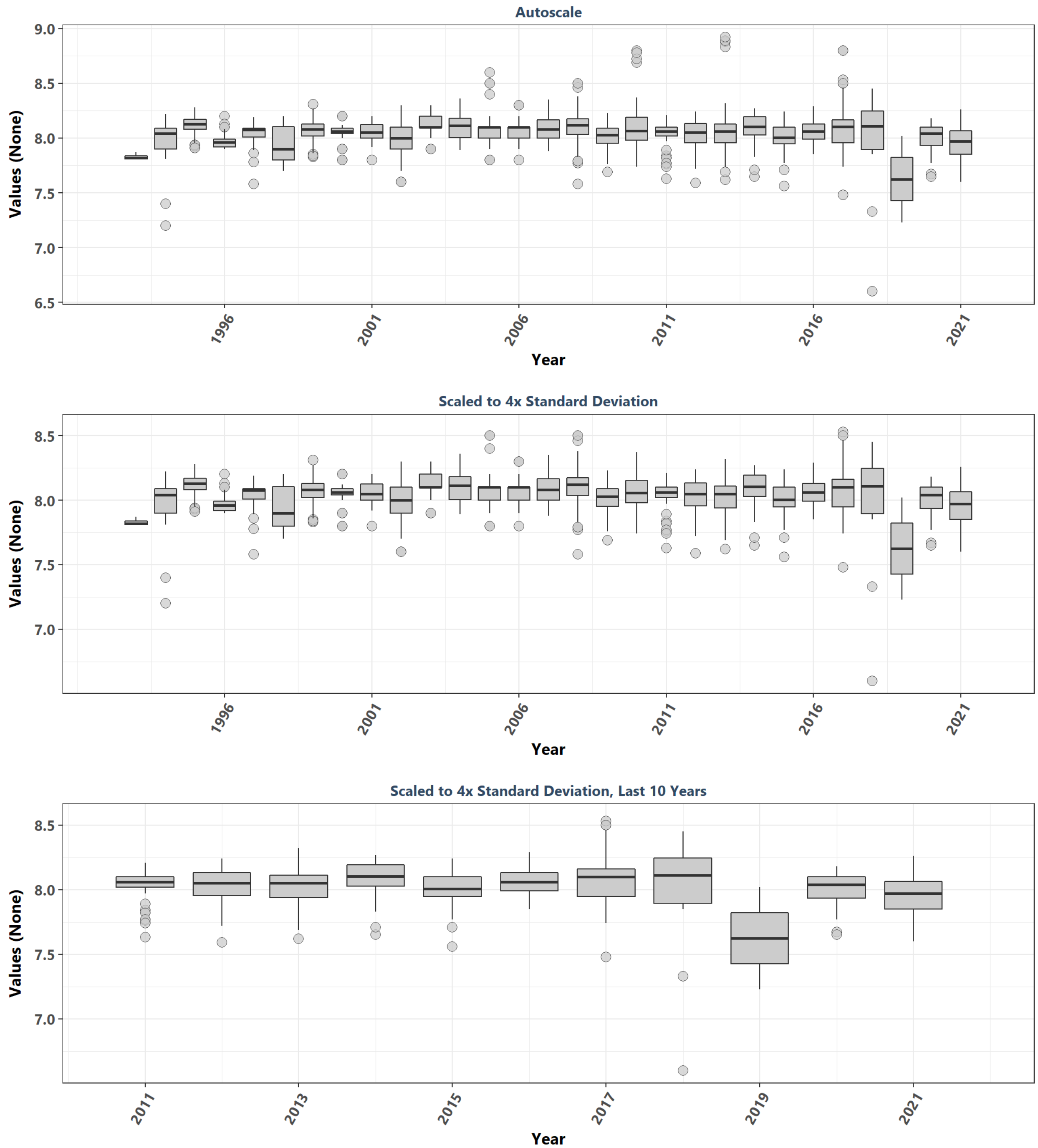


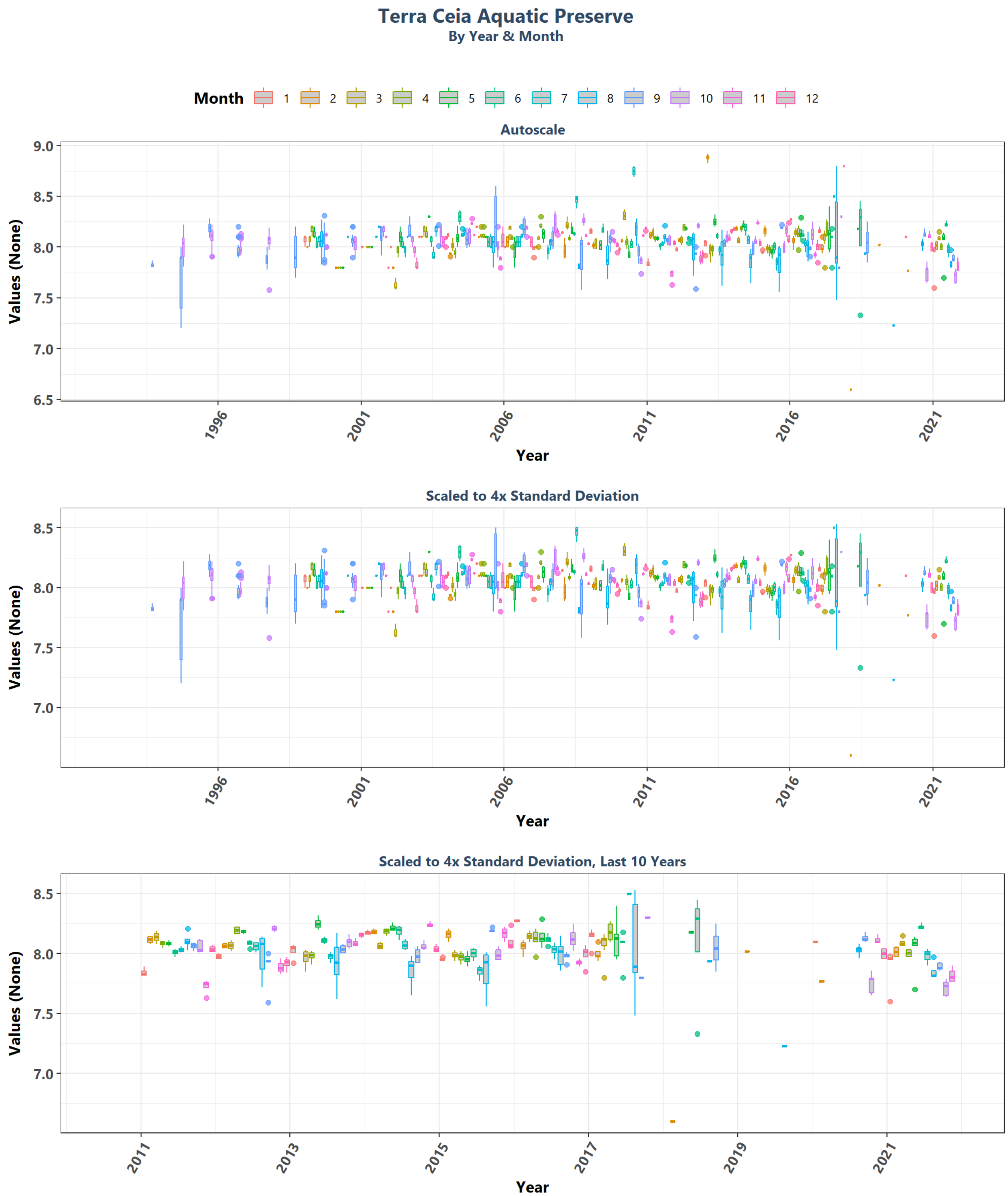


St. Martins Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.50
8.25
8.00
7.75
7.50
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (None)
8.5
8.0
7.5
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8.5
8.0
7.5
2012
2014
2016
2018
2020
2022
Year

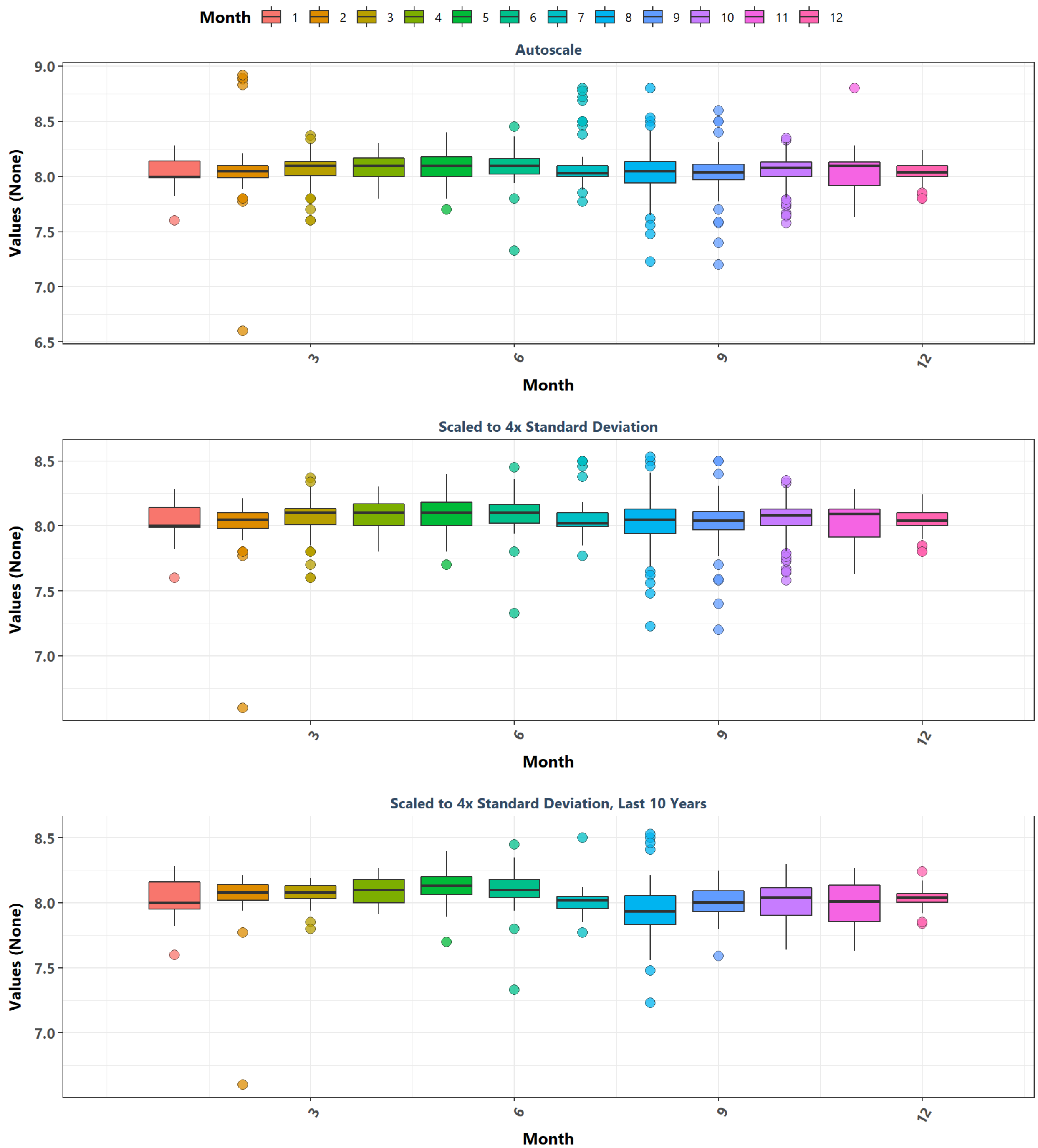

St. Martins Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
8.50
8.25
8.00
7.75
7.50
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (None)
8.5
8.0
7.5
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8.5
8.0
7.5
3
6
9
12
Month

Terra Ceia Aquatic Preserve
By Year

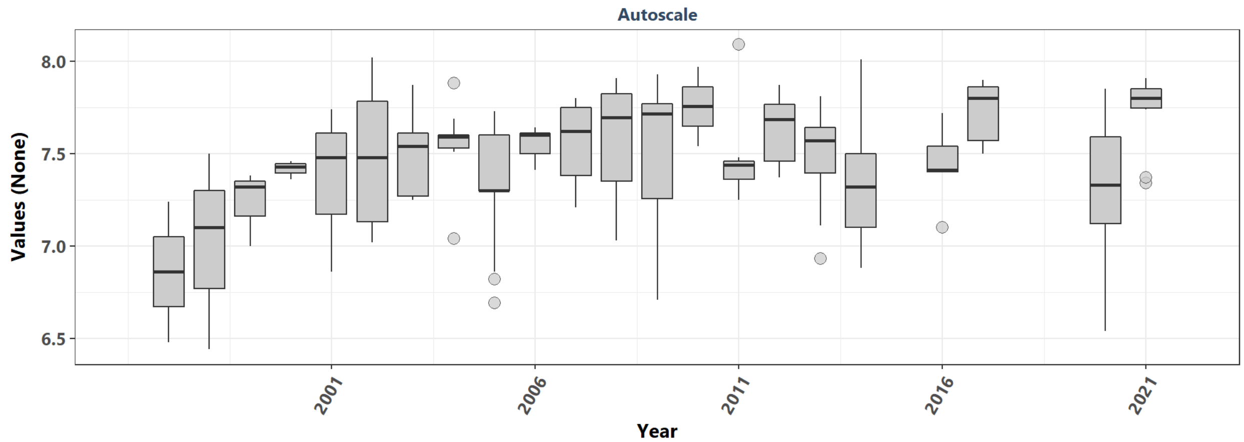


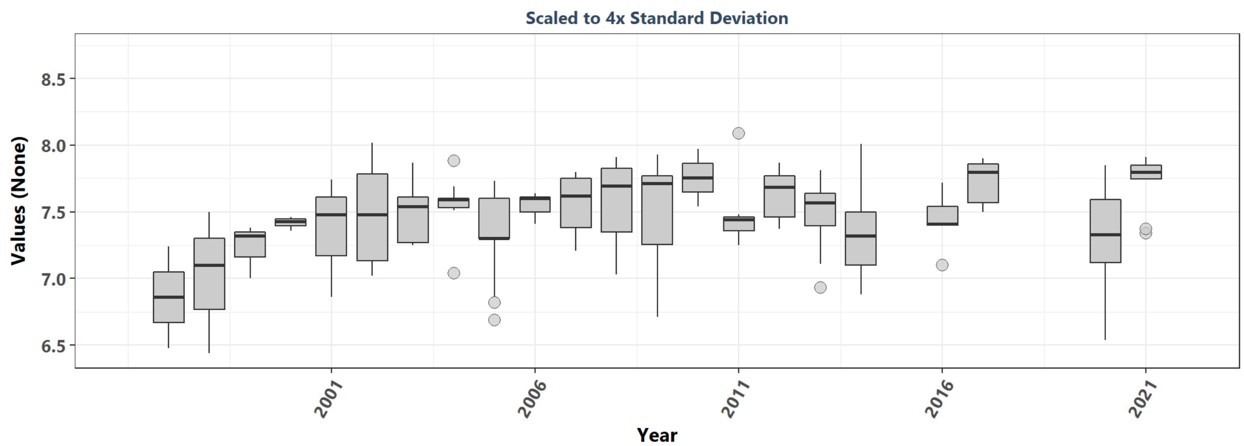

Terra Ceia Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (None)
9.0
8.5
8.0
7.5
7.0
6.5
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (None)
8.5
8.0
7.5
7.0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (None)
8.5
8.0
7.5
7.0
2011
2013
2015
2017
2019
2021
Year

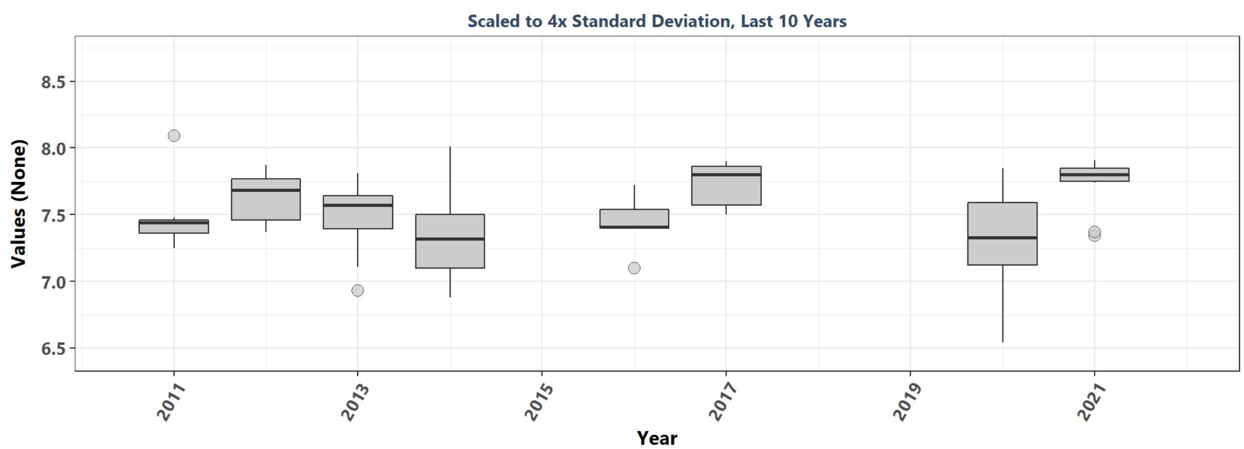
Terra Ceia Aquatic Preserve
By Month



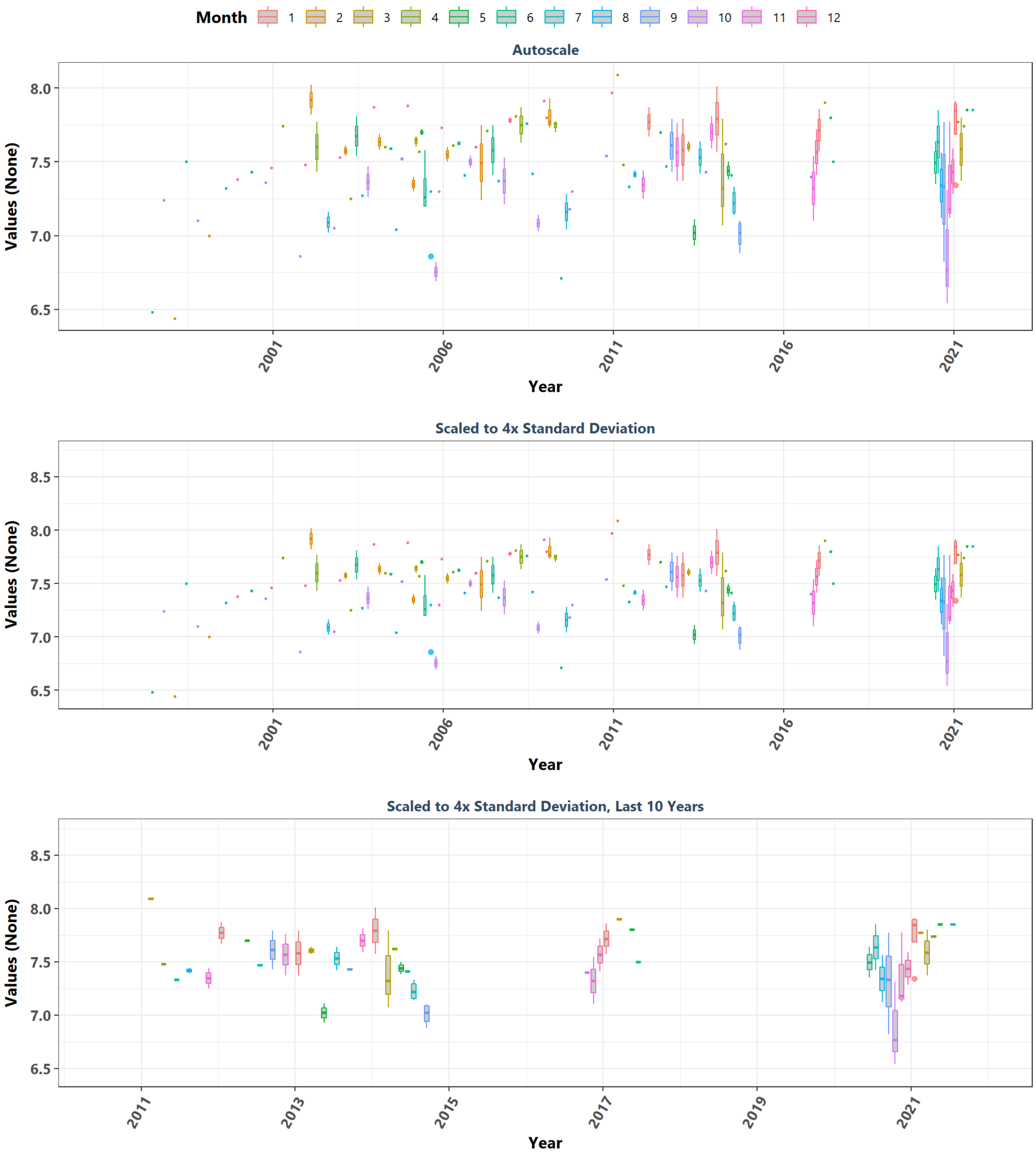
Tomoka Marsh Aquatic Preserve
By Year

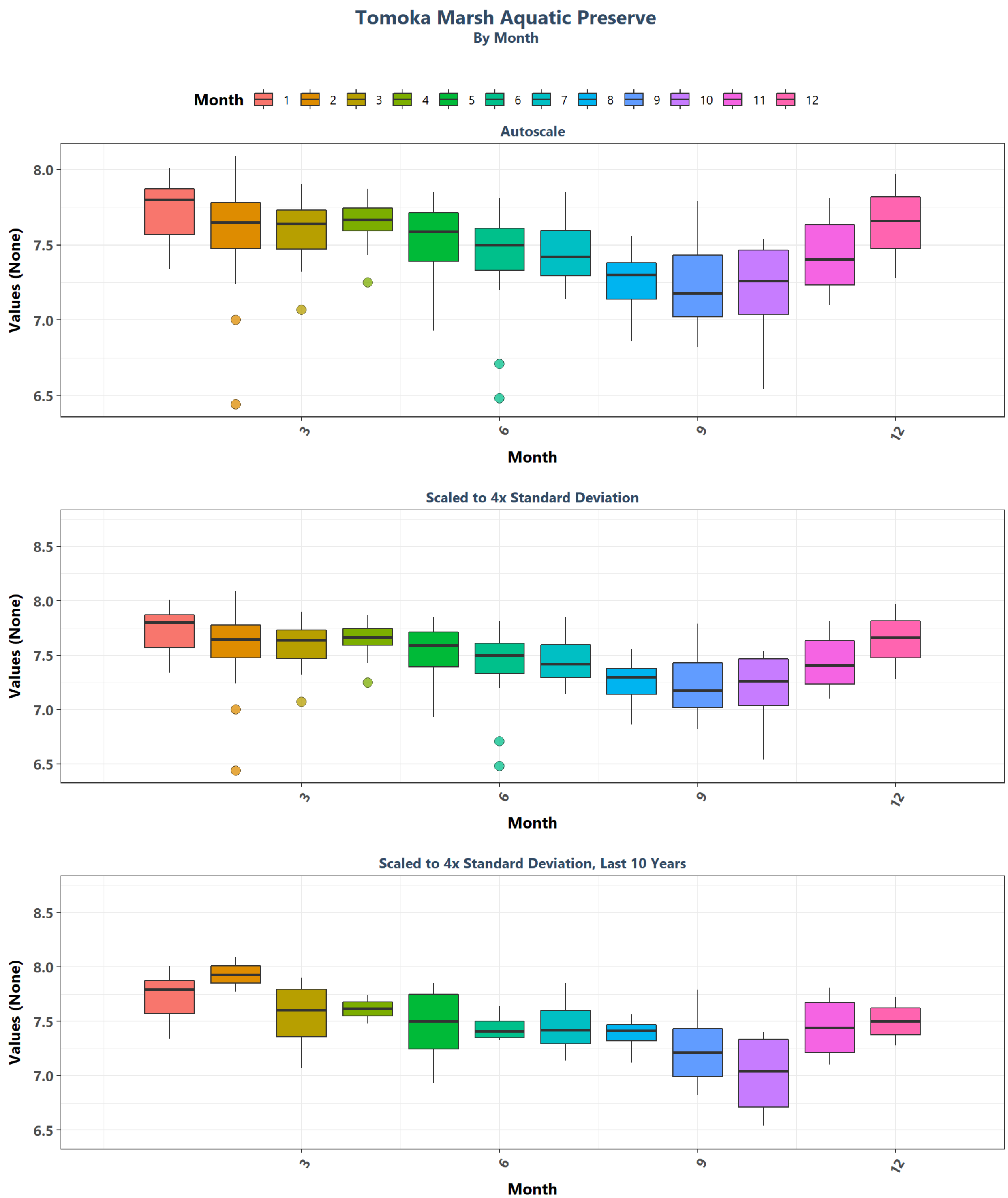






Tomoka Marsh Aquatic Preserve
By Year & Month






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