SEACAR Discrete Water Quality Analysis: Sample Surface Turbidity

Last compiled on 24 June, 2022

Contents

Important Notes

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

https://github.com/FloridaSEACAR/SEACAR_Panzik

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

Libraries

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

```
library(knitr)
library(data.table)
library(plyr)
library(dplyr)
library(lubridate)
library(ggplot2)
library(ggpubr)
library(scales)
library(EnvStats)
library(tidyr)
library(stringr)
library(kableExtra)

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

File Import

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

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

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

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

#file_in <- "C:/Users/steph/Dropbox/SEACAR_Panzik/SEACAR_Panzik/WQ_Discrete/data/Combined_WQ_WC_NUT_Wat
data <- fread(file_in, sep="|", header=TRUE, stringsAsFactors=FALSE,
              select=c("ManagedAreaName", "ProgramID", "ProgramName",
                       "ProgramLocationID", "SampleDate", "Year", "Month",
                       "RelativeDepth", "ActivityType", "ParameterName",
                       "ResultValue", "ParameterUnits", "ValueQualifier",
                       "SEACAR_QAQCFlagCode", "Include"), na.strings="")

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

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

Data Filtering and Data Impacted by Specific Value Qualifiers

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

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

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

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

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

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

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

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

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

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

```
}

if(param_name=="Water_Temperature"){
   data <- data[data$ResultValue>=-2,]
} else{
   data <- data[data$ResultValue>=0,]
}


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

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

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

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

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


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

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

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

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

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

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

```
data$VQ_Plot <- data$ValueQualifier

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

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


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

```
## Number of Measurements: 222630, Number Passed Filter: 222630
## I Codes: 3549 (1.594125%)
## Q Codes: 520 (0.233571%)
## U Codes: 238 (0.106904%)
```

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

```
            N_H=length(grep("H", data$ValueQualifier[data$ProgramID==476])),
            perc_H=100*N_H/length(data$ValueQualifier),
            N_I=length(grep("I", data$ValueQualifier)),
            perc_I=100*N_I/length(data$ValueQualifier),
            N_Q=length(grep("Q", data$ValueQualifier)),
            perc_Q=100*N_Q/length(data$ValueQualifier),
            N_S=length(grep("S", data$ValueQualifier)),
            perc_S=100*N_S/length(data$ValueQualifier),
            N_U=length(grep("U", data$ValueQualifier)),
            perc_U=100*N_U/length(data$ValueQualifier))

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

Managed Area Statistics

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

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

```
MA_YM_Stats <- data[data$Use_In_Analysis==TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, Year, Month) %>%
   dplyr::summarize(ParameterName=parameter,
            RelativeDepth=depth,
            ActivityType=activity,
            N_Data=length(ResultValue),
            Min=min(ResultValue),
            Max=max(ResultValue),
            Median=median(ResultValue),
```

```
            Mean=mean(ResultValue),
            StandardDeviation=sd(ResultValue),
            ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),
                             collapse=', '))
MA_YM_Stats <- as.data.table(MA_YM_Stats[order(MA_YM_Stats$ManagedAreaName,
                                               MA_YM_Stats$Year,
                                               MA_YM_Stats$Month), ])
fwrite(MA_YM_Stats, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                           "_ManagedArea_YearMonth_Stats.txt"), sep="|")
rm(MA_YM_Stats)

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

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

Monitoring Location Statistics

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

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

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

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

Seasonal Kendall Tau Analysis

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

The following steps are performed:

1. Define the functions used in the analysis
2. Check to see if there are any groups to run analysis on.
3. Take the `data` variable and only include rows that have a `SufficientData` value of `TRUE`
4. Group data that have the same `ManagedAreaName`.

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

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

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

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

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

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

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

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

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

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

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

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

  return(KT)
}
runStats <- function(data, MA_M_Stats) {
  data$Index <- as.Date(data$SampleDate) # , "%Y-%m-%d")
  data$ResultValue <- as.numeric(data$ResultValue)
  # Calculate basic stats
  stats.median <- median(data$ResultValue, na.rm = TRUE)
  stats.minYear <- min(data$relyear, na.rm = TRUE)
  stats.maxYear <- max(data$relyear, na.rm = TRUE)
  # Calculate Kendall Tau and Slope stats, then update appropriate columns and table
  seasondata <- MA_M_Stats[MA_M_Stats$ManagedAreaName == MA_Include[i]]
  KT <- tauSeasonal(data, TRUE, stats.median,
                    stats.minYear, stats.maxYear, seasondata)
  # if (is.null(KT[9])) {
  if (is.na(KT[season == "All", trend])) {
    KT <- tauSeasonal(data, FALSE, stats.median,
                      stats.minYear, stats.maxYear, seasondata)
  }
  if (is.null(KT.Stats) == TRUE) {
    KT.Stats <- KT
  } else{
    KT.Stats <- rbind(KT.Stats, KT)
  }
  return(KT.Stats)
}
trend_calculator <- function(slope, median_value, p) {
  trend <-
    if (p < .05 & abs(slope) > abs(median_value) / 10.) {
      if (slope > 0) {
        2
      }
      else {
        -2
      }
    }
  else if (p < .05 & abs(slope) < abs(median_value) / 10.) {
    if (slope > 0) {
      1
    }
    else {
      -1
```

```
      }
   }
   else
      0
   return(trend)
}
KT.Stats <- NULL
# Loop that goes through each managed area.
# List of managed areas stored in MA_Years$ManagedAreaName
c_names <- c("AreaID", "ManagedAreaName", "Season", "Median", "Independent",
             "tau", "z", "p_z", "chi_sq", "p_chi_sq", "SennSlope", "SennIntercept", "Trend")
if(n==0){
   KT.Stats <- data.frame(matrix(ncol=length(c_names),
                                 nrow=length(MA_Summ$ManagedAreaName)))
   colnames(KT.Stats) <- c_names
   # KT.Stats[, c("AreaID", "ManagedAreaName")] <-
      # MA_Summ[, c("AreaID", "ManagedAreaName")]
} else{
   for (i in 1:n) {
      x <- nrow(data[data$Use_In_Analysis == TRUE &
                        data$ManagedAreaName == MA_Include[i], ])
      if (x>0) {
         KT.Stats <- runStats(data[data$Use_In_Analysis == TRUE &
                                      data$ManagedAreaName ==
                                      MA_Include[i], ], MA_M_Stats)
      }
   }
   KT.Stats <- as.data.frame(KT.Stats)
   # c_names <- c("AreaID", "ManagedAreaName", "Season", "Median", "Independent",
   #              "tau", "z", "p_z", "chi_sq", "p_chi_sq", "SennSlope", "SennIntercept", "Trend")
   if(dim(KT.Stats)[2]==1){
      KT.Stats <- as.data.frame(t(KT.Stats))
   }
   colnames(KT.Stats) <- c_names
   rownames(KT.Stats) <- seq(1:nrow(KT.Stats))
   KT.Stats$tau <- round(as.numeric(KT.Stats$tau), digits=4)
   KT.Stats$z <- round(as.numeric(KT.Stats$z), digits=4)
   KT.Stats$p_z <- round(as.numeric(KT.Stats$p_z), digits=4)
   KT.Stats$chi_sq <- round(as.numeric(KT.Stats$chi_sq), digits=4)
   KT.Stats$p_chi_sq <- round(as.numeric(KT.Stats$p_chi_sq), digits=4)
   KT.Stats$SennSlope <- as.numeric(KT.Stats$SennSlope)
   KT.Stats$SennIntercept <- as.numeric(KT.Stats$SennIntercept)
   KT.Stats$Trend <- as.integer(KT.Stats$Trend)
}

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

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

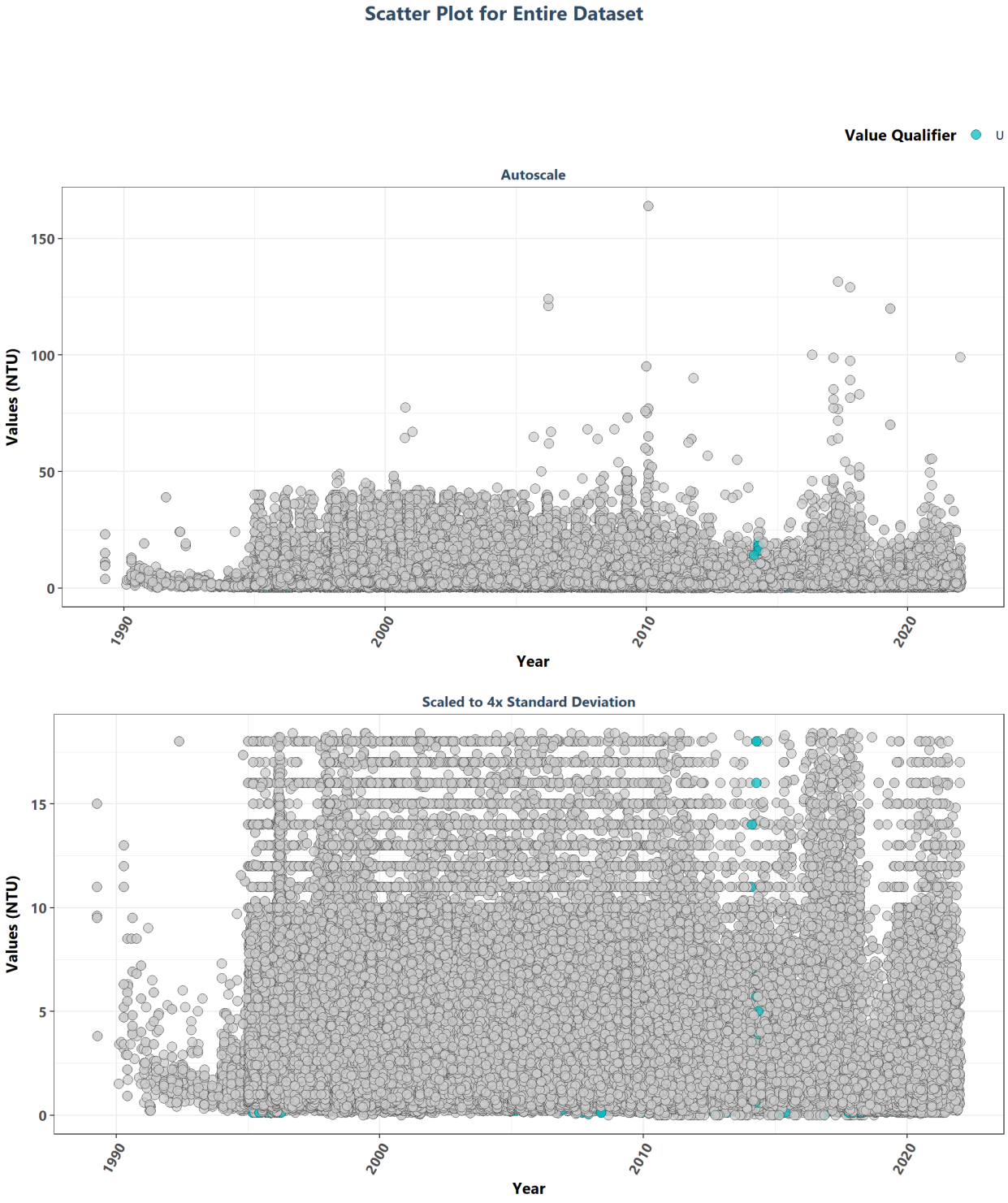
```
KT.Stats2$MonitoringID <- NULL
fwrite(KT.Stats2, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                         "_KendallTau_Stats.txt"), sep="|")
rm(KT.Stats2)
data <- data[!is.na(data$ResultValue),]
```

Appendix I: Scatter Plot of Entire Dataset

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

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

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


p1 <- ggplot(data=data[data$Include==TRUE,],
             aes(x=SampleDate, y=ResultValue, fill=VQ_Plot)) +
   geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")"),
        fill="Value Qualifier") +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal",
         legend.justification="right") +
   scale_x_date(labels=date_format("%Y")) +
   {if(inc_H==TRUE){
      scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                 "HU"="#7CAE00"), na.value="#cccccc")
   } else if(param_name=="Secchi_Depth"){
      scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
```

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


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

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

p0 <- ggplot() + labs(title="Scatter Plot for Entire Dataset") +
   plot_theme + theme(panel.border=element_blank(),
                      panel.grid.major=element_blank(),
                      panel.grid.minor=element_blank(),
                      axis.line=element_blank())

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



Appendix II: Dataset Summary Box Plots

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

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

This set of box plots are grouped by year.

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

p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit, ")")) +
   plot_theme

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

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

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

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

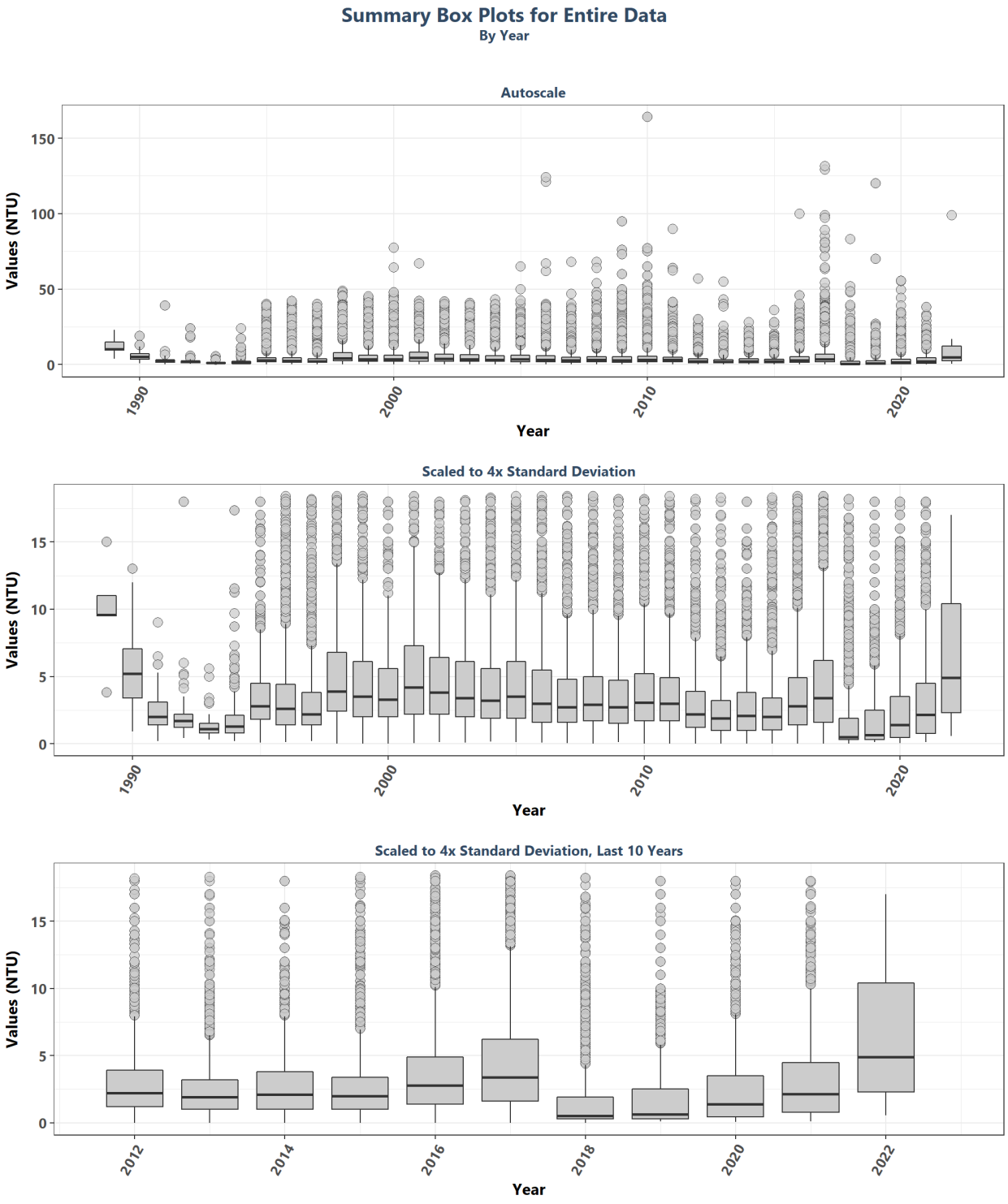
```
                      subtitle="By Year") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

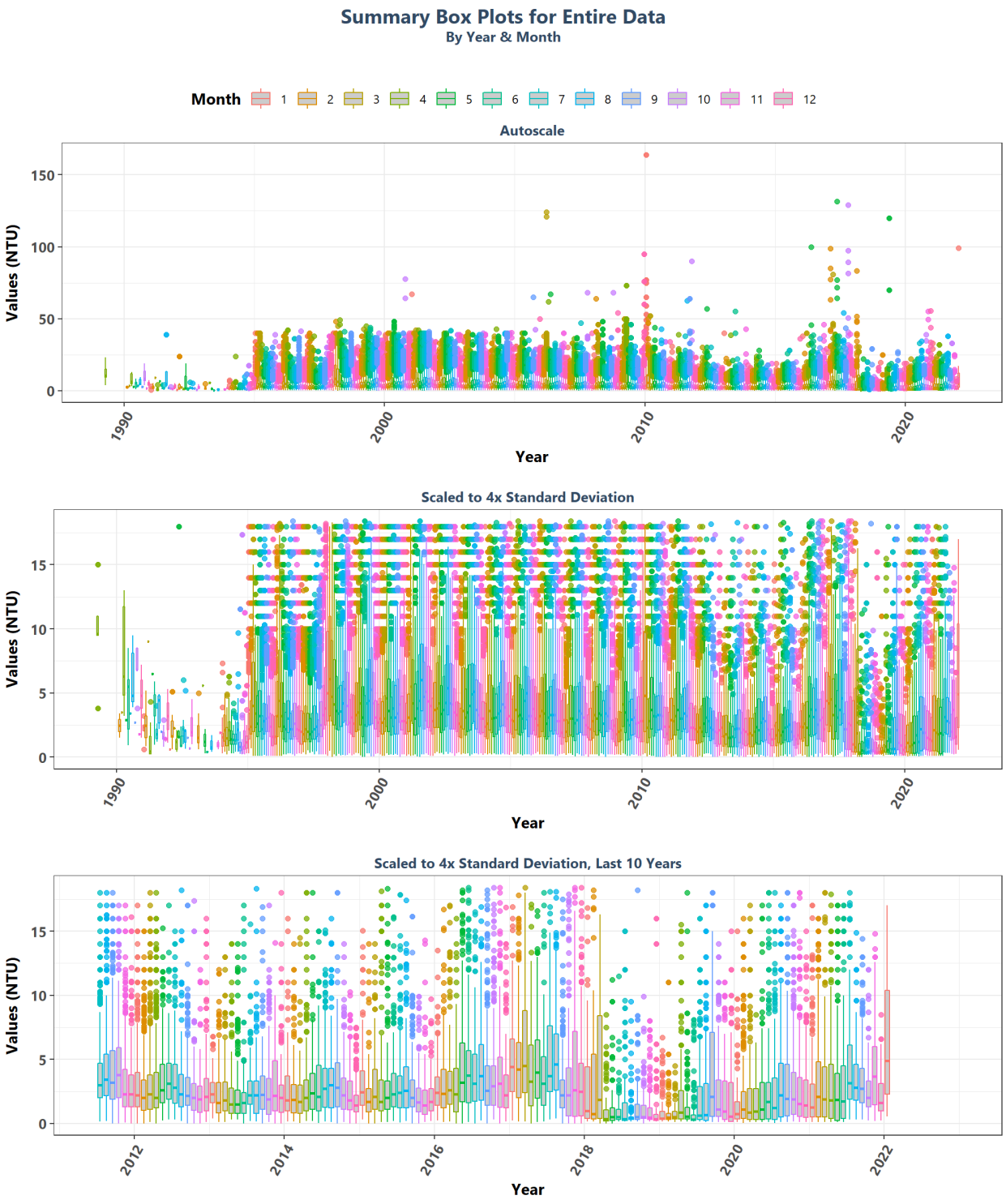
This set of box plots are grouped by year and month with the color being related to the month.

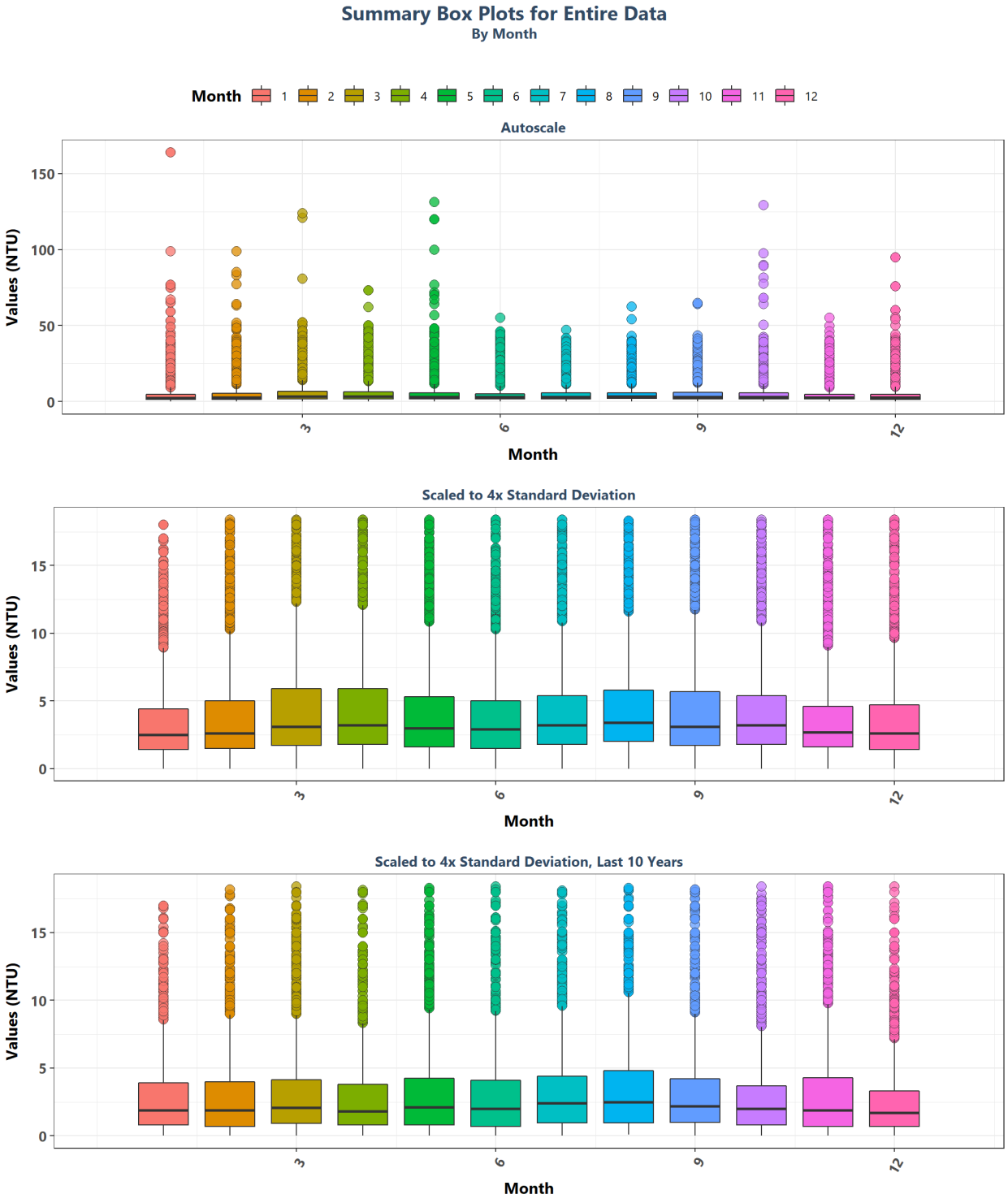
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit, ")"), color="Month") +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
   guides(color=guide_legend(nrow=1))

p2 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   plot_theme +
   theme(legend.position="none", axis.text.x=element_text(face="bold"),
         axis.text.y=element_text(face="bold"))

p3 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(max(data$Year) - 10.5, max(data$Year)+1),
                      breaks=seq(max(data$Year) - 10, max(data$Year), 2)) +
   plot_theme +
   theme(legend.position="none")
leg <- get_legend(p1)
set <- ggarrange(leg, p1 + theme(legend.position="none"), p2, p3, ncol=1,
                 heights=c(0.1, 1, 1, 1))

p0 <- ggplot() + labs(title="Summary Box Plots for Entire Data",
                      subtitle="By Year & Month") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

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

```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Month",
        y=paste0("Values (", unit, ")"), fill="Month") +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
   guides(fill=guide_legend(nrow=1))

p2 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Month", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="none")

p3 <- ggplot(data=data[data$Include==TRUE &
                       data$Year >= max(data$Year) - 10, ],
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
        x="Month", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="none")
leg <- get_legend(p1)
set <- ggarrange(leg, p1 + theme(legend.position="none"), p2, p3, ncol=1,
                 heights=c(0.1, 1, 1, 1))

p0 <- ggplot() + labs(title="Summary Box Plots for Entire Data",
                      subtitle="By Month") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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


Summary Box Plots for Entire Data
By Year
Autoscale
Values (NTU)
150
100
50
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (NTU)
15
10
5
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
15
10
5
0
2012
2014
2016
2018
2020
2022
Year

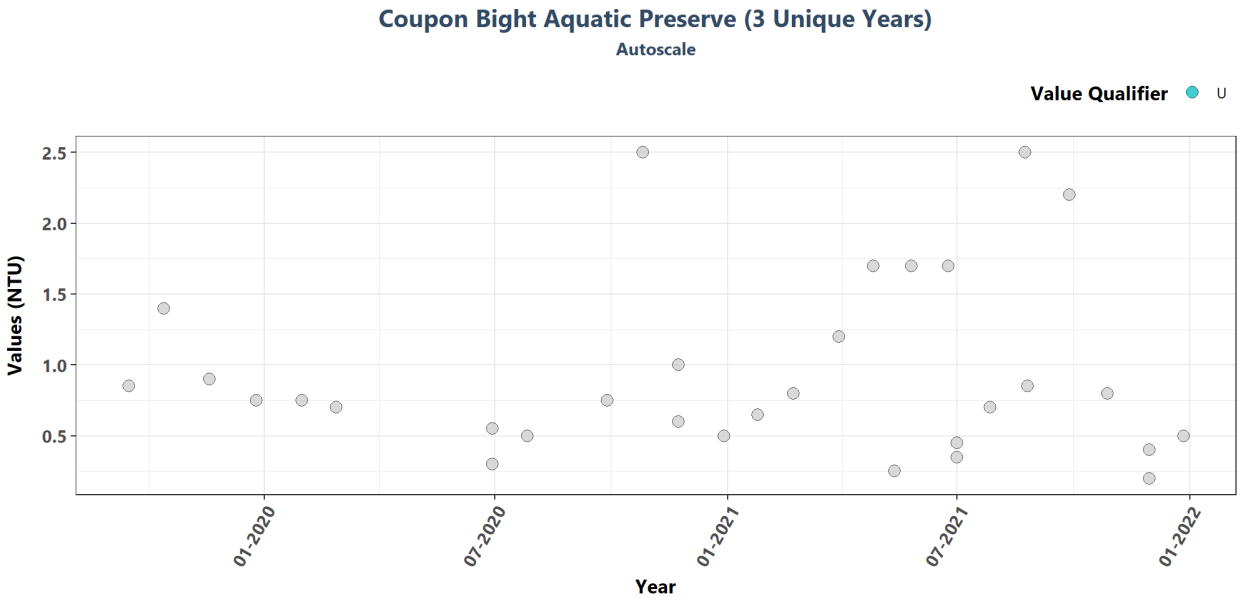

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
150
100
50
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (NTU)
15
10
5
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
15
10
5
0
2012
2014
2016
2018
2020
2022
Year

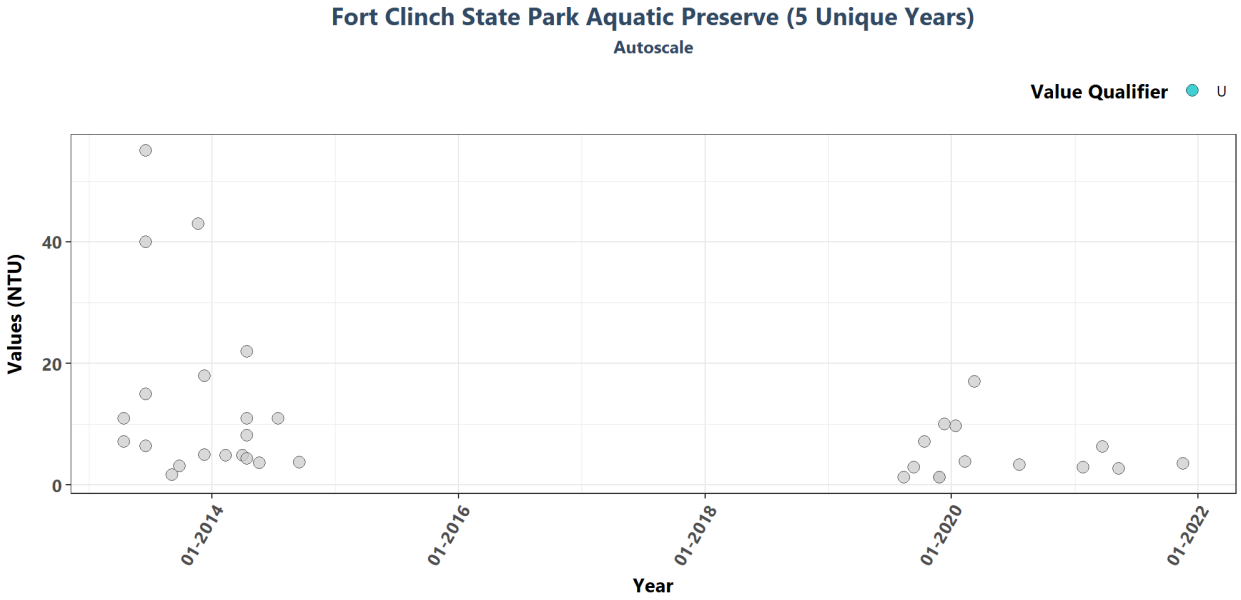


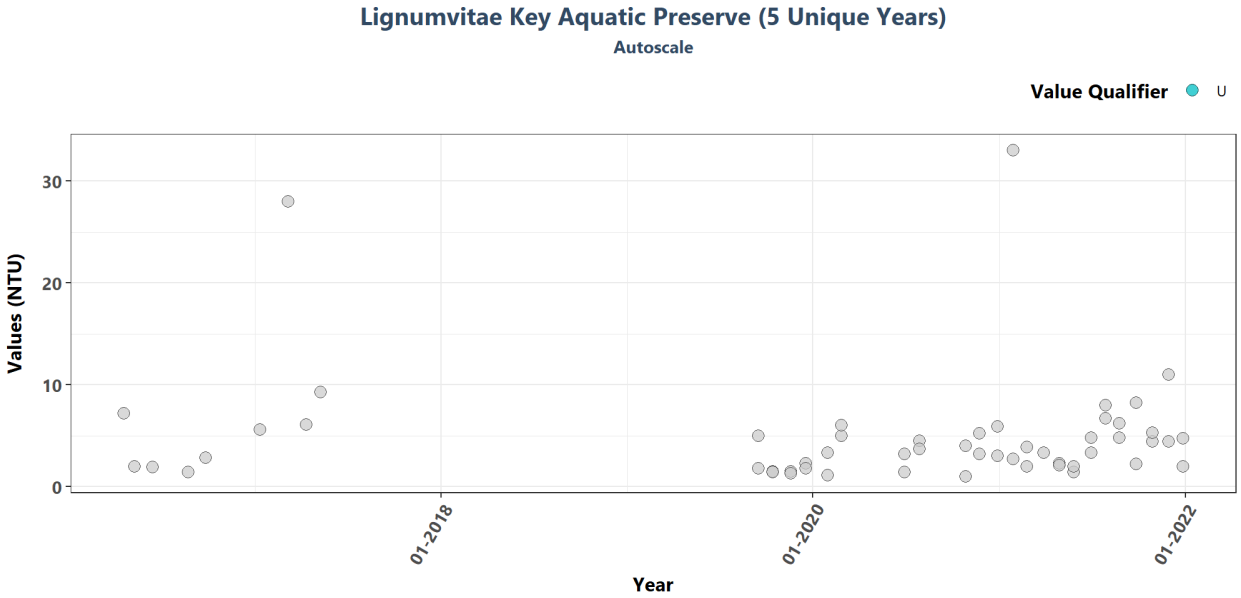
Appendix III: Excluded Managed Areas

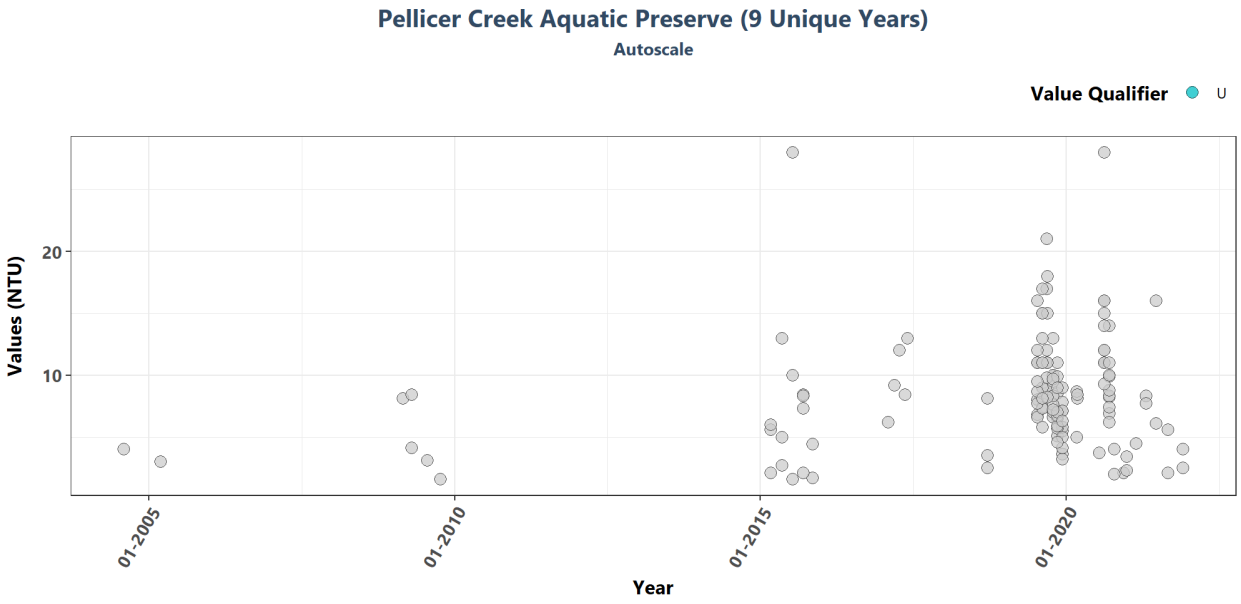
Scatter plots of data values are created for managed areas that have fewer than 10 separate years of data entries. Data points are colored based on specific value qualifiers of interest.

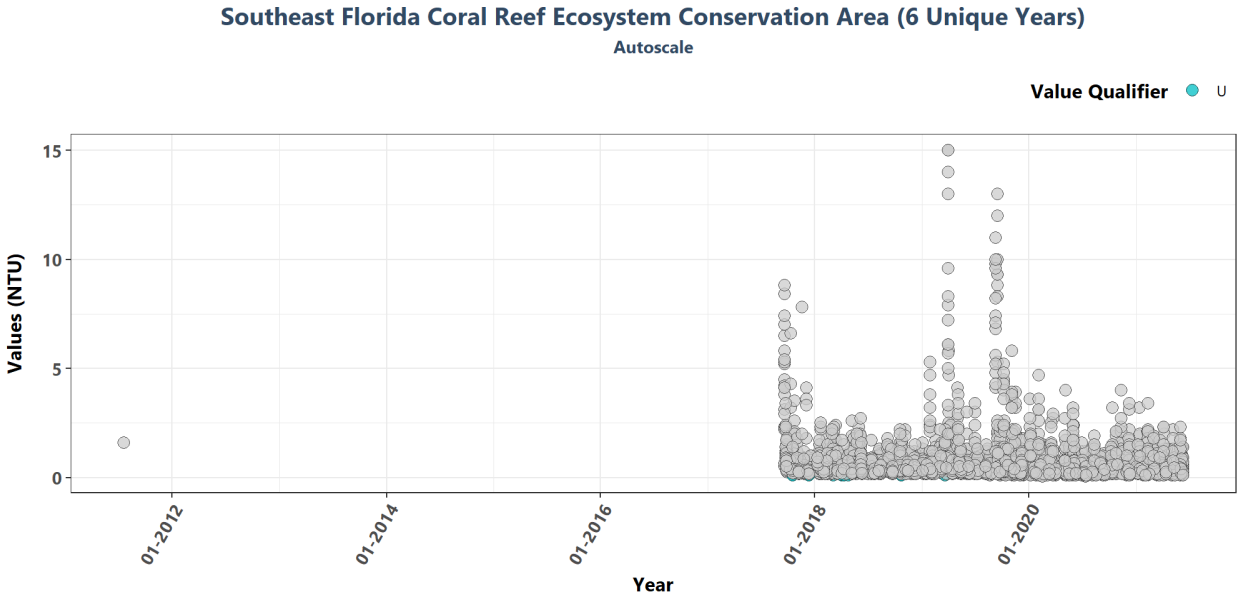
```
if(z==0){
   print("There are no managed areas that qualify.")
} else {
   for(i in 1:z){
      p1<-ggplot(data=data[data$ManagedAreaName==MA_Exclude$ManagedAreaName[i]&
                           data$Include==TRUE, ],
                 aes(x=SampleDate, y=ResultValue, fill=VQ_Plot)) +
         geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
         labs(title=paste0(MA_Exclude$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)
   }
}
```




Fort Clinch State Park Aquatic Preserve (5 Unique Years)
Autoscale
Value Qualifier
U
Values (NTU)
40
20
0
01-2014
01-2016
01-2018
01-2020
01-2022
Year


Lignumvitae Key Aquatic Preserve (5 Unique Years)
Autoscale
Value Qualifier
U
Values (NTU)
30
20
10
0
01-2018
01-2020
01-2022
Year





Appendix IV: Managed Area Trendlines

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

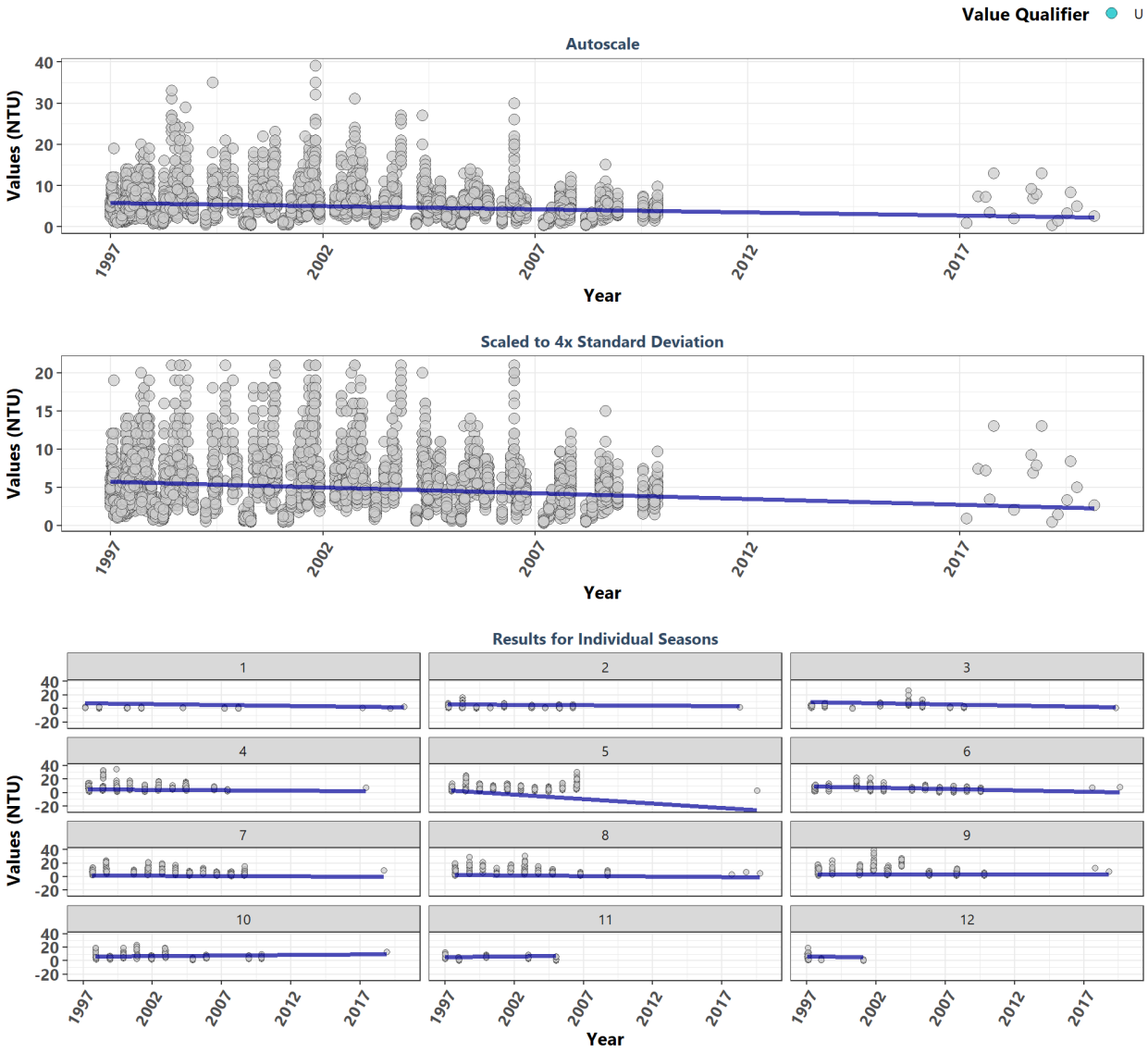
1. Use the data set that only has `SufficientData` of `TRUE` for the desired managed area
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Determine the minimum, mean, and standard deviation for the data to be used for y-axis scales
    - Excludes the top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

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

```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$SufficientData==TRUE &
                           data$ManagedAreaName==MA_Include[i],]
      plot_data$Season <- factor(plot_data$Month, levels = c("All", seq(1, 12)), ordered = TRUE)
      year_lower <- min(plot_data$relyear)
      year_upper <- max(plot_data$relyear)
      min_RV <- min(plot_data$ResultValue)
      mn_RV <- mean(plot_data$ResultValue[plot_data$ResultValue <
                                             quantile(data$ResultValue, 0.98)])
      sd_RV <- sd(plot_data$ResultValue[plot_data$ResultValue <
                                          quantile(data$ResultValue, 0.98)])
      x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
      y_scale <- mn_RV + 4 * sd_RV

      tau <- KT.Stats$tau[KT.Stats$ManagedAreaName==MA_Include[i]]
      s_slope <- KT.Stats$SennSlope[KT.Stats$ManagedAreaName==MA_Include[i]]
      s_int <- KT.Stats$SennIntercept[KT.Stats$ManagedAreaName==MA_Include[i]]
      trend <- KT.Stats$Trend[KT.Stats$ManagedAreaName==MA_Include[i]]
      z <- KT.Stats$z[KT.Stats$ManagedAreaName==MA_Include[i]]
      p_z <- KT.Stats$p_z[KT.Stats$ManagedAreaName==MA_Include[i]]
      chi_sq <- KT.Stats$chi_sq[KT.Stats$ManagedAreaName==MA_Include[i]]
      p_chi_sq <- KT.Stats$p_chi_sq[KT.Stats$ManagedAreaName==MA_Include[i]]

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

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

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

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

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

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

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

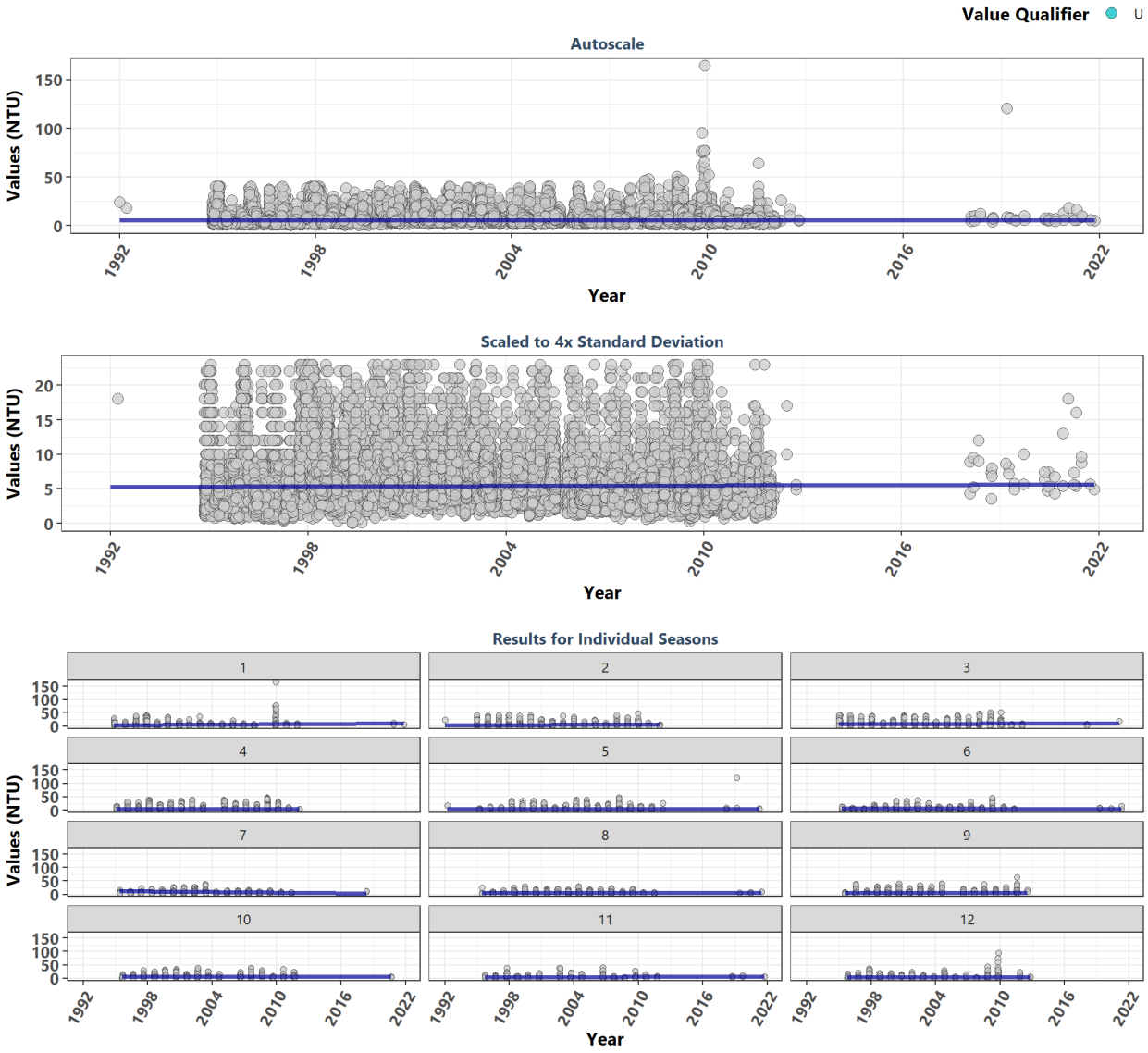
```
      splot <- ggplot(plot_data, aes(x = relyear_dd, y = ResultValue)) +
         geom_point(shape = 21, size = 1.5, color="#333333", fill="#cccccc", alpha=0.75) +
         geom_segment(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season != "All", ], aes(x = relyear_dd,
                     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

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

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

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

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



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3540	5.30	-0.1436	-0.1500	5.7653	-11.9	0.0000	203.7	0	-1
1	149	1.00	-0.2371	-0.2556	7.7889	-8.3	0.0000	NA	NA	-2
2	291	2.00	-0.1085	-0.1250	6.1750	-6.3	0.0000	NA	NA	-1
3	240	3.40	-0.1928	-0.3333	9.4667	-0.1	0.9412	NA	NA	-1
4	370	7.00	-0.0689	-0.1000	4.4000	2.7	0.0063	NA	NA	-1
5	356	6.30	-0.3788	-1.3250	3.8000	3.4	0.0007	NA	NA	-2
6	431	5.30	-0.2453	-0.4000	9.4000	-6.8	0.0000	NA	NA	-1
7	440	6.00	-0.4534	-0.0833	1.4167	-7.5	0.0000	NA	NA	-1
8	296	7.80	-0.2446	-0.1350	2.5400	-5.0	0.0000	NA	NA	-1
9	375	7.00	-0.0032	0.0000	3.4000	-7.2	0.0000	NA	NA	-1
10	318	5.55	0.0935	0.1667	6.3333	-2.9	0.0036	NA	NA	1
11	129	4.30	0.1193	0.1889	5.3556	-1.2	0.2341	NA	NA	1
12	145	3.80	-0.2184	-0.2333	6.7000	-8.9	0.0000	NA	NA	-1

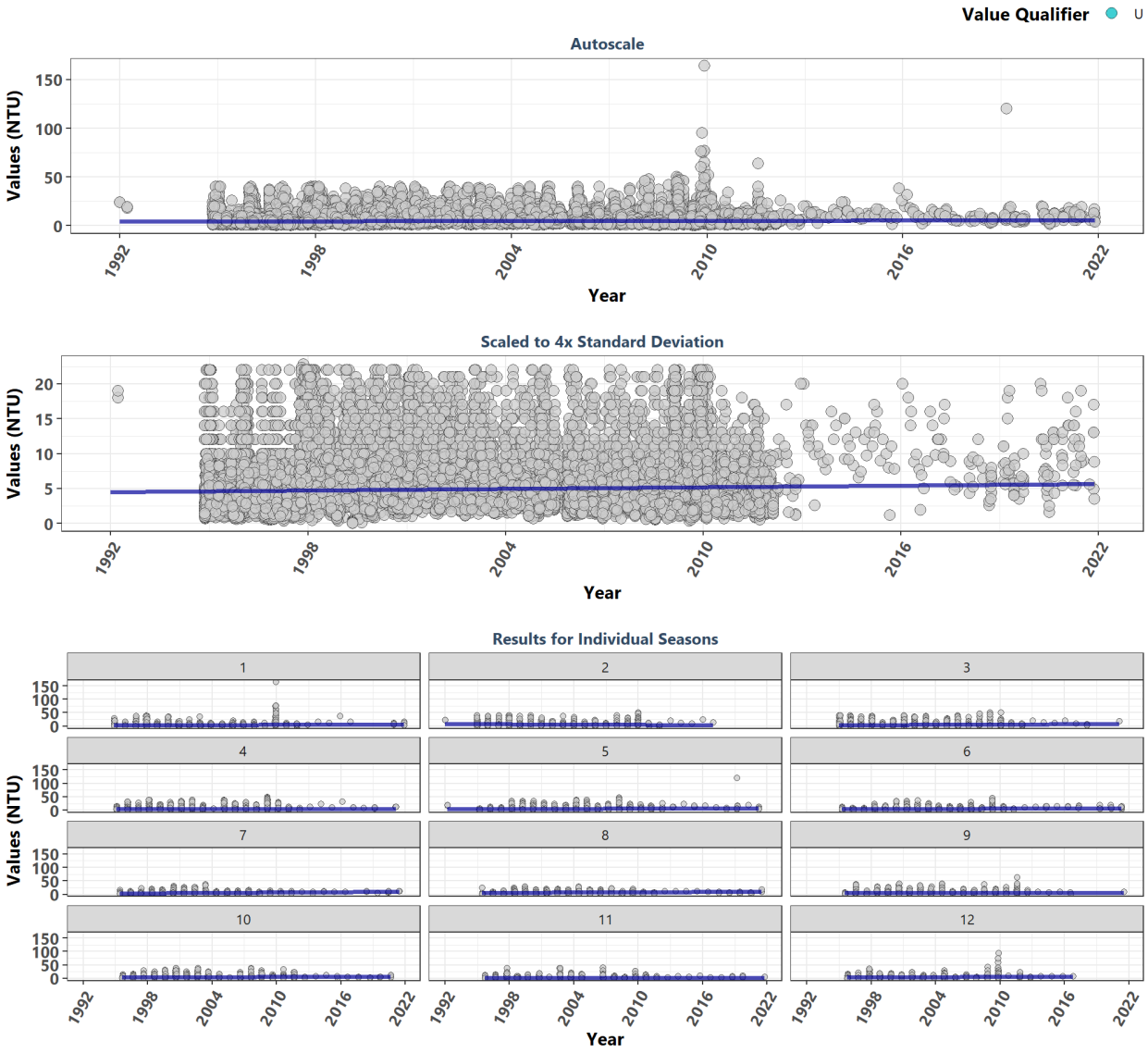
[a] $p < 0.00005$ appear as 0 due to rounding

Apalachicola Bay Aquatic Preserve



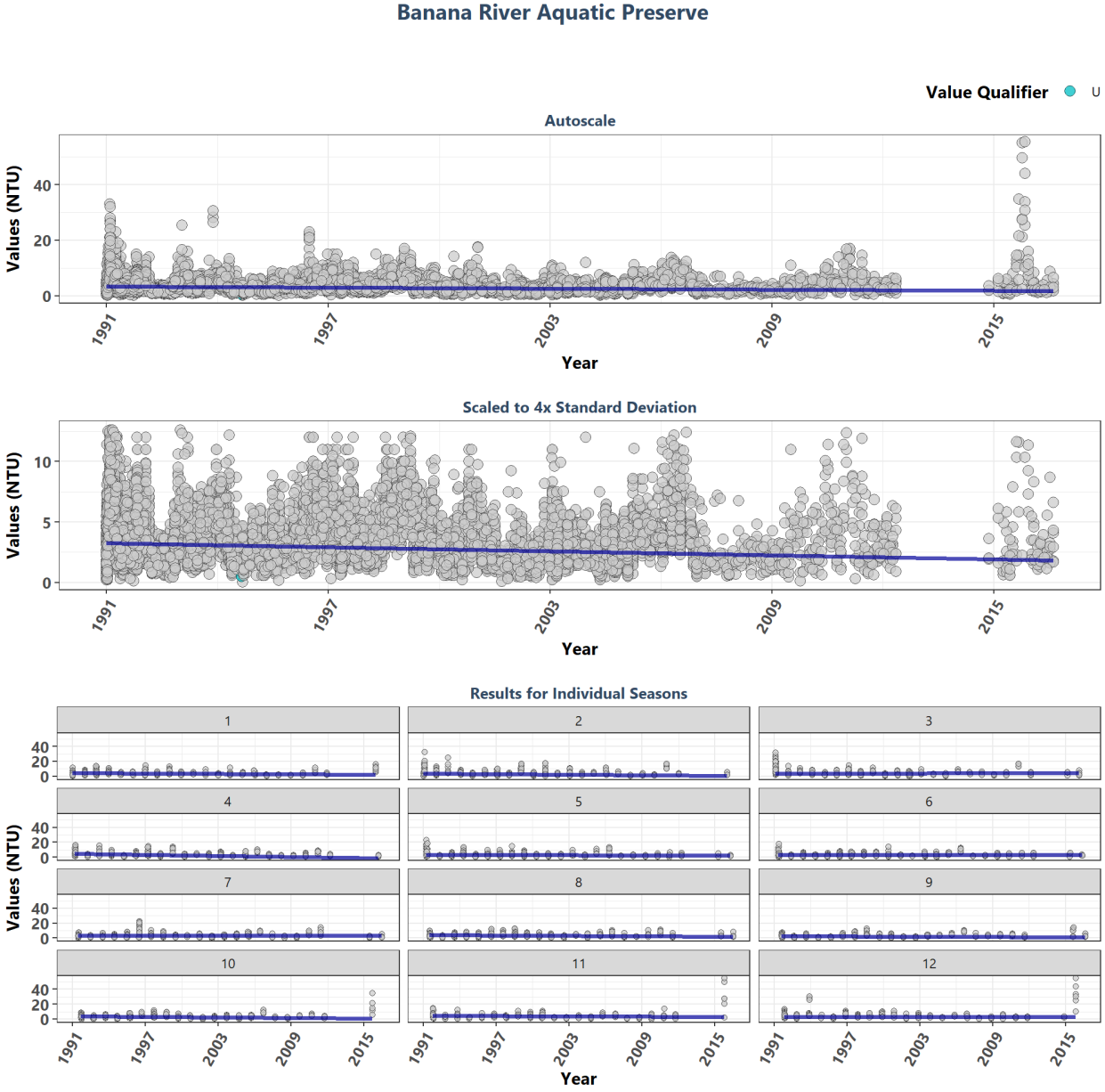
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	15504	5.6	0.0155	0.0111	5.2838	2.8	0.0047	299.1	0	1
1	1234	5.4	0.2207	0.2364	3.0727	1.1	0.2669	NA	NA	1
2	1246	7.2	0.0514	0.0455	4.3000	-3.1	0.0020	NA	NA	1
3	1567	9.1	0.0379	0.0600	8.0600	-10.9	0.0000	NA	NA	1
4	1098	8.6	-0.0034	0.0000	5.4000	1.9	0.0586	NA	NA	-1
5	1211	5.4	0.0121	0.0077	5.3077	0.6	0.5265	NA	NA	1
6	1276	4.8	-0.0582	-0.0818	7.7727	2.8	0.0058	NA	NA	-1
7	1628	5.4	-0.1825	-0.3364	12.1273	-0.2	0.8373	NA	NA	-1
8	2079	5.1	0.0210	0.0200	5.2600	7.2	0.0000	NA	NA	1
9	1253	5.8	0.1056	0.0900	4.2900	1.8	0.0652	NA	NA	1
10	935	5.3	0.0346	0.0400	5.4800	0.9	0.3580	NA	NA	1
11	1000	3.4	0.0200	0.0167	5.1500	-1.9	0.0554	NA	NA	1
12	977	5.2	-0.0402	-0.0200	3.6200	10.4	0.0000	NA	NA	-1

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



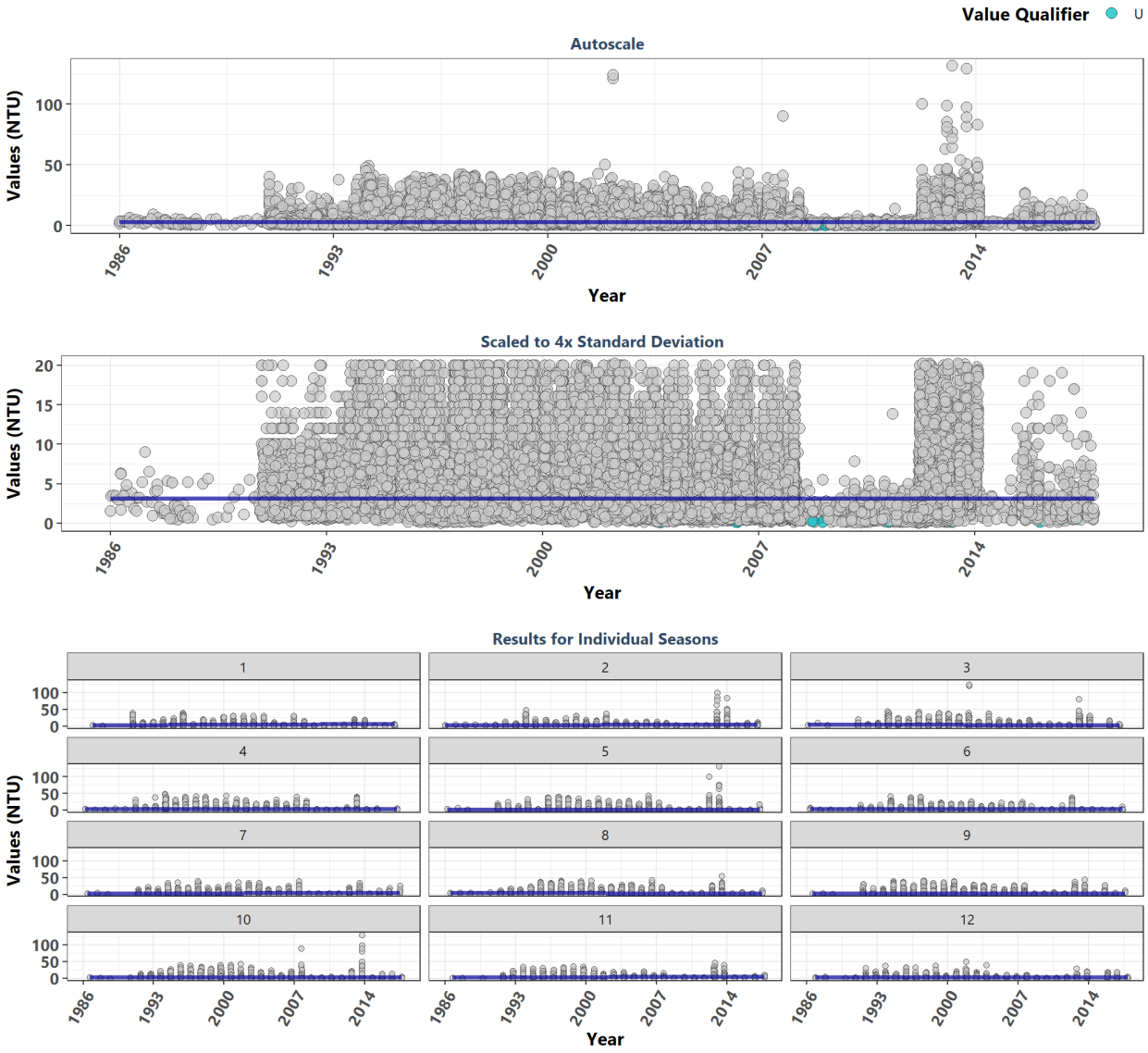
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	22861	5.1	0.0494	0.0385	4.5194	9.5	0.0000	405.7	0	1
1	2108	4.9	0.1219	0.1000	3.5000	7.3	0.0000	NA	NA	1
2	2123	6.4	-0.1255	-0.1800	8.6200	-0.8	0.4180	NA	NA	-1
3	2514	7.0	0.1053	0.1074	4.0407	-9.5	0.0000	NA	NA	1
4	1912	6.9	0.0167	0.0111	4.4889	5.8	0.0000	NA	NA	1
5	1767	5.1	0.0930	0.0727	4.2455	3.7	0.0002	NA	NA	1
6	1624	4.6	0.0593	0.0500	4.5500	7.4	0.0000	NA	NA	1
7	1891	5.3	0.2654	0.2769	2.2462	2.2	0.0285	NA	NA	1
8	2526	4.9	0.0882	0.1250	5.7750	7.0	0.0000	NA	NA	1
9	1748	5.2	-0.0117	-0.0083	6.4667	0.7	0.4729	NA	NA	-1
10	1597	4.6	0.0114	0.0077	5.1308	1.0	0.3146	NA	NA	1
11	1511	3.0	0.0149	0.0000	3.0000	0.9	0.3829	NA	NA	-1
12	1540	4.6	0.0335	0.0273	5.0273	15.7	0.0000	NA	NA	1

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



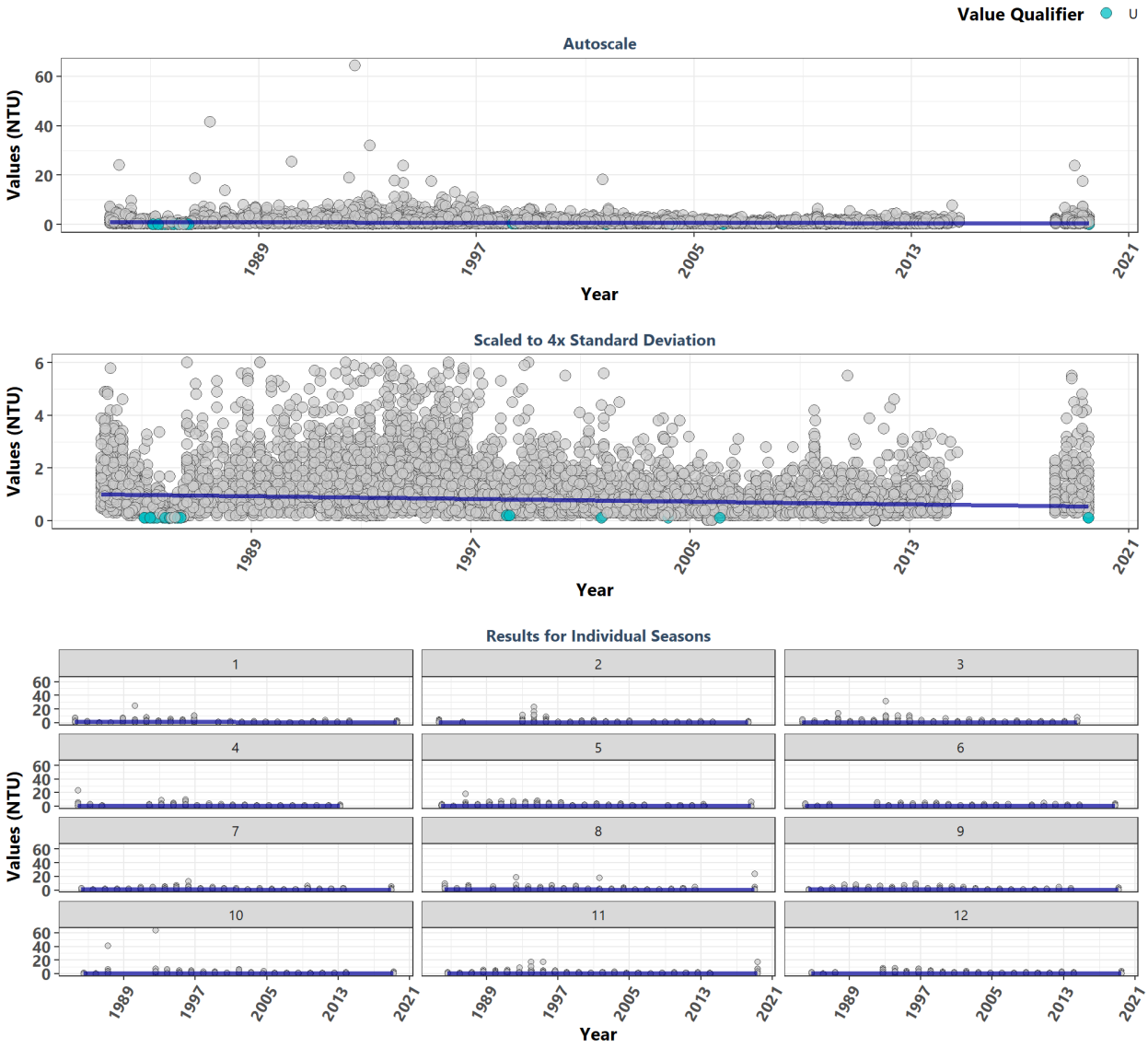
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	12853	3.20	-0.1118	-0.0571	3.2727	-18.9	0.0000	426.4	0	-1
1	861	3.90	-0.1378	-0.0923	4.1692	-3.2	0.0012	NA	NA	-1
2	1267	3.80	-0.2251	-0.1091	3.3455	-7.6	0.0000	NA	NA	-1
3	990	3.40	0.0710	0.0357	3.1857	-18.3	0.0000	NA	NA	1
4	1157	2.90	-0.3869	-0.2500	4.7750	-9.7	0.0000	NA	NA	-1
5	1417	2.80	-0.1057	-0.0429	3.0571	-12.8	0.0000	NA	NA	-1
6	1136	2.56	0.0216	0.0111	2.8778	-7.2	0.0000	NA	NA	1
7	1107	2.80	-0.0176	-0.0057	3.1286	-5.3	0.0000	NA	NA	-1
8	1350	3.40	-0.1309	-0.0800	4.1600	3.9	0.0001	NA	NA	-1
9	775	2.90	-0.1421	-0.0600	2.8035	0.9	0.3564	NA	NA	-1
10	1262	3.10	-0.1899	-0.1000	3.5000	-0.9	0.3461	NA	NA	-1
11	906	3.20	-0.0736	-0.0488	4.1925	-0.3	0.7321	NA	NA	-1
12	625	4.00	-0.0075	0.0000	3.2000	-5.0	0.0000	NA	NA	-1

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



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	41216	3.5	0.0098	0.0000	3.1200	3.3	0.0008	421.9	0	-1
1	2997	2.6	0.1364	0.1200	3.0400	-0.8	0.4135	NA	NA	1
2	3118	2.9	0.0333	0.0214	3.7214	-0.5	0.6155	NA	NA	1
3	3851	3.7	-0.0875	-0.0667	5.0333	-9.4	0.0000	NA	NA	-1
4	3553	4.1	0.0498	0.0286	3.0286	-7.8	0.0000	NA	NA	1
5	3475	4.2	-0.0099	0.0000	2.6000	4.6	0.0000	NA	NA	-1
6	2968	3.6	-0.0060	0.0000	2.9000	0.3	0.7482	NA	NA	-1
7	3642	3.4	0.0766	0.0500	3.2000	4.5	0.0000	NA	NA	1
8	4072	3.9	-0.1012	-0.0800	4.9800	7.3	0.0000	NA	NA	-1
9	3461	4.6	-0.0617	-0.0286	2.8714	12.1	0.0000	NA	NA	-1
10	3876	4.0	0.0164	0.0071	2.8000	3.1	0.0019	NA	NA	1
11	3194	2.9	0.0520	0.0400	3.6000	1.4	0.1648	NA	NA	1
12	3009	2.5	0.0039	0.0000	3.6000	-5.1	0.0000	NA	NA	-1

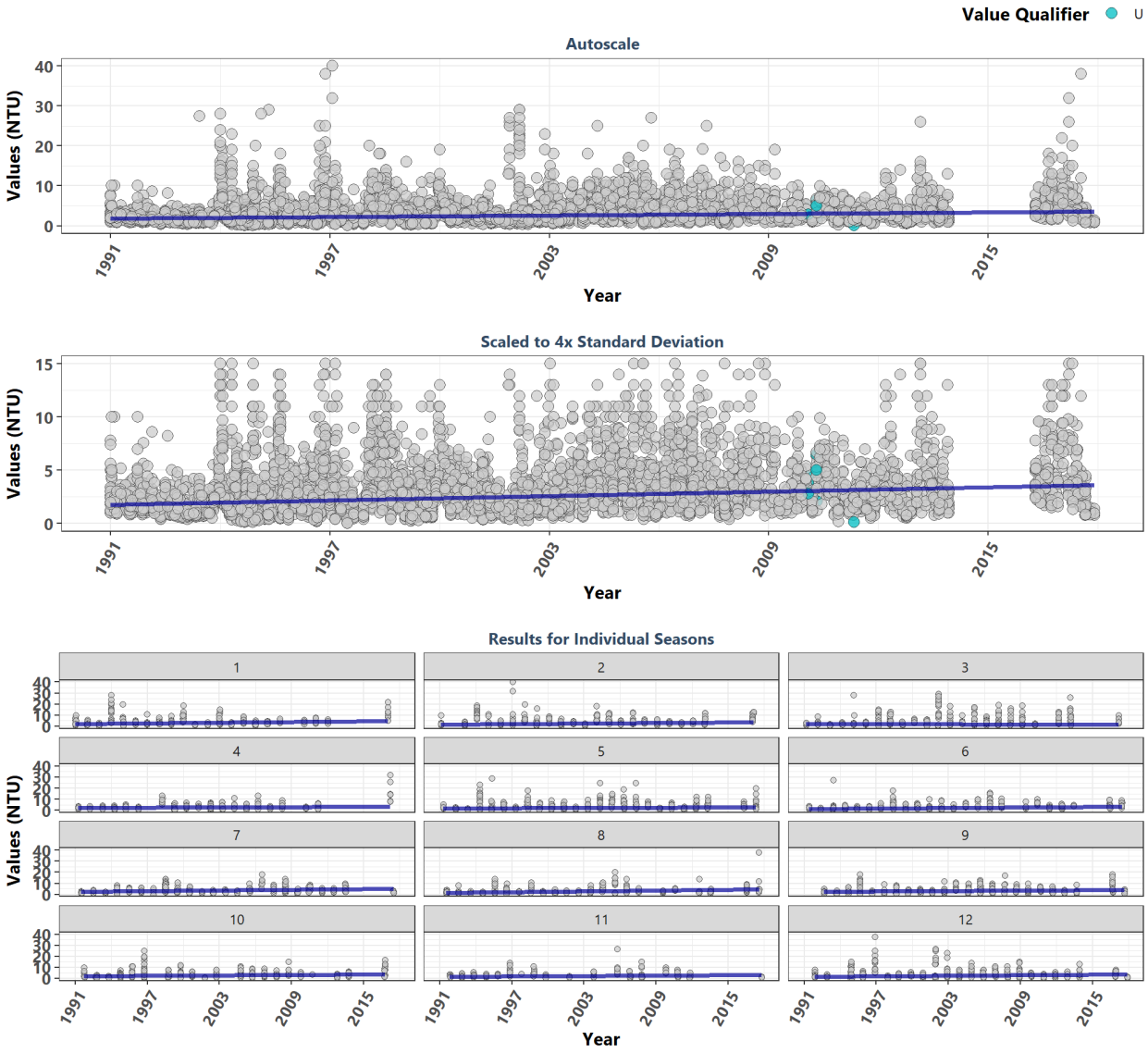
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	8583	0.80	-0.1249	-0.0167	1.1933	-17.2	0.0000	26.7	0.0052	-1
1	720	0.90	-0.1996	-0.0286	1.5857	-8.0	0.0000	NA	NA	-1
2	607	0.70	-0.1018	-0.0143	1.0429	-6.0	0.0000	NA	NA	-1
3	746	0.80	-0.1080	-0.0111	0.9556	-5.0	0.0000	NA	NA	-1
4	668	0.70	-0.0918	-0.0118	0.9836	-5.9	0.0000	NA	NA	-1
5	760	0.83	-0.0760	-0.0079	0.8886	-5.6	0.0000	NA	NA	-1
6	664	0.70	-0.0869	-0.0107	1.1357	-3.6	0.0004	NA	NA	-1
7	761	0.90	-0.1615	-0.0250	1.3250	-3.6	0.0003	NA	NA	-1
8	699	0.80	-0.1810	-0.0250	1.4000	-7.2	0.0000	NA	NA	-1
9	809	0.70	-0.0925	-0.0133	1.2067	-4.6	0.0000	NA	NA	-1
10	686	0.70	-0.1213	-0.0203	1.2677	-3.0	0.0028	NA	NA	-1
11	787	0.90	-0.1530	-0.0200	1.1800	-3.9	0.0001	NA	NA	-1
12	676	0.70	-0.1354	-0.0222	1.3411	-4.0	0.0001	NA	NA	-1

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

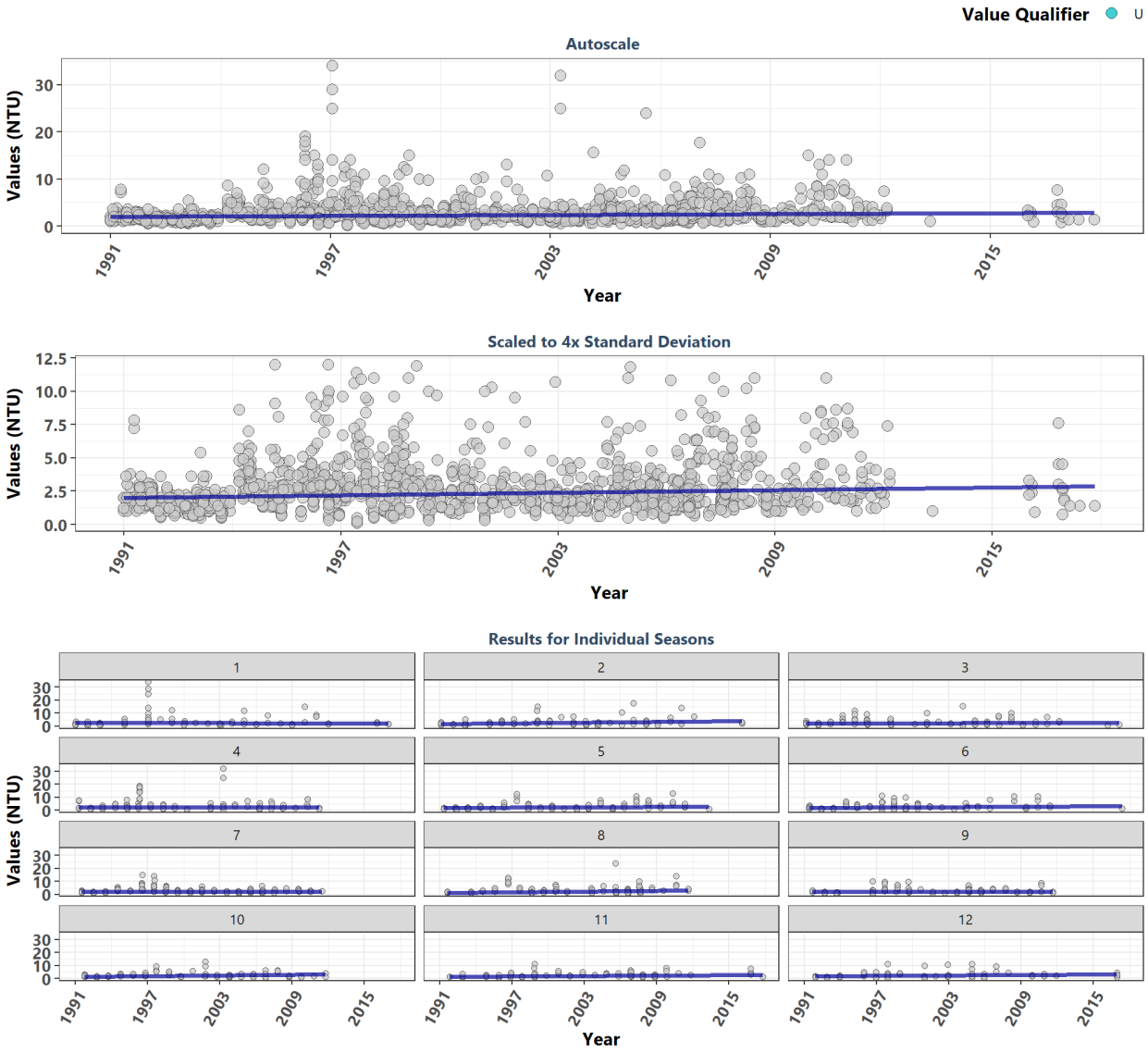
Boca Ciega Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	5826	2.4	0.1792	0.0688	1.7700	20.8	0.0000	113.7	0	1
1	435	2.8	0.2037	0.1000	1.8000	4.5	0.0000	NA	NA	1
2	379	3.0	0.2807	0.1000	1.2000	6.2	0.0000	NA	NA	1
3	511	2.9	-0.0450	-0.0111	1.9889	6.9	0.0000	NA	NA	-1
4	385	2.0	0.0656	0.0273	2.4000	4.6	0.0000	NA	NA	1
5	588	2.1	0.1555	0.0556	1.5000	10.2	0.0000	NA	NA	1
6	489	2.4	0.2153	0.0714	1.2857	10.0	0.0000	NA	NA	1
7	468	2.7	0.2118	0.1125	2.1000	4.8	0.0000	NA	NA	1
8	449	2.3	0.3008	0.1067	1.5467	7.2	0.0000	NA	NA	1
9	640	2.0	0.1440	0.0643	2.3500	8.2	0.0000	NA	NA	1
10	527	2.1	0.1475	0.0500	2.2500	6.5	0.0000	NA	NA	1
11	407	1.9	0.1876	0.0636	1.6545	-1.4	0.1727	NA	NA	1
12	548	2.7	0.2263	0.0800	1.7400	2.3	0.0213	NA	NA	1

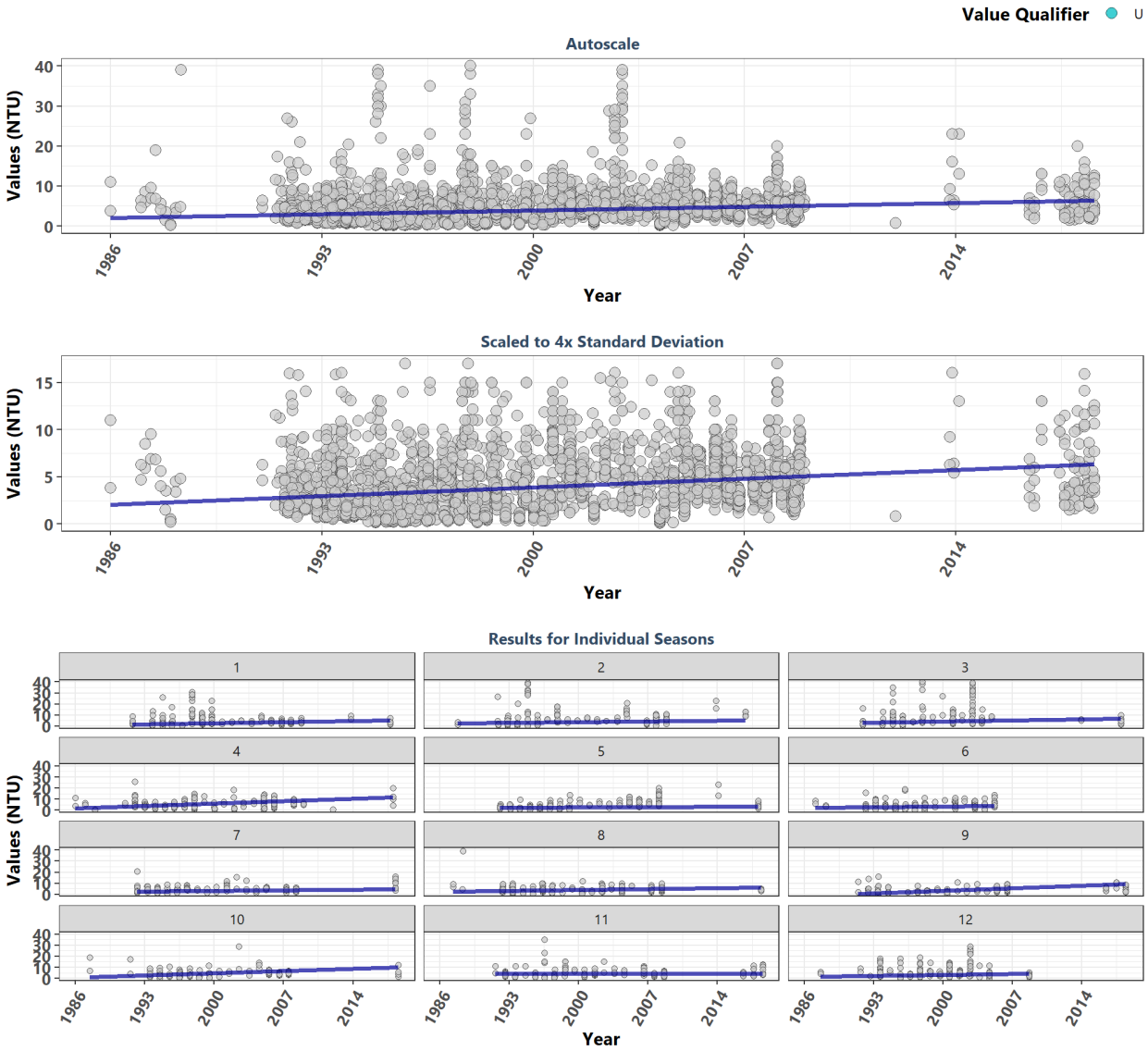
[a] $p < 0.00005$ appear as 0 due to rounding

Cape Haze Aquatic Preserve



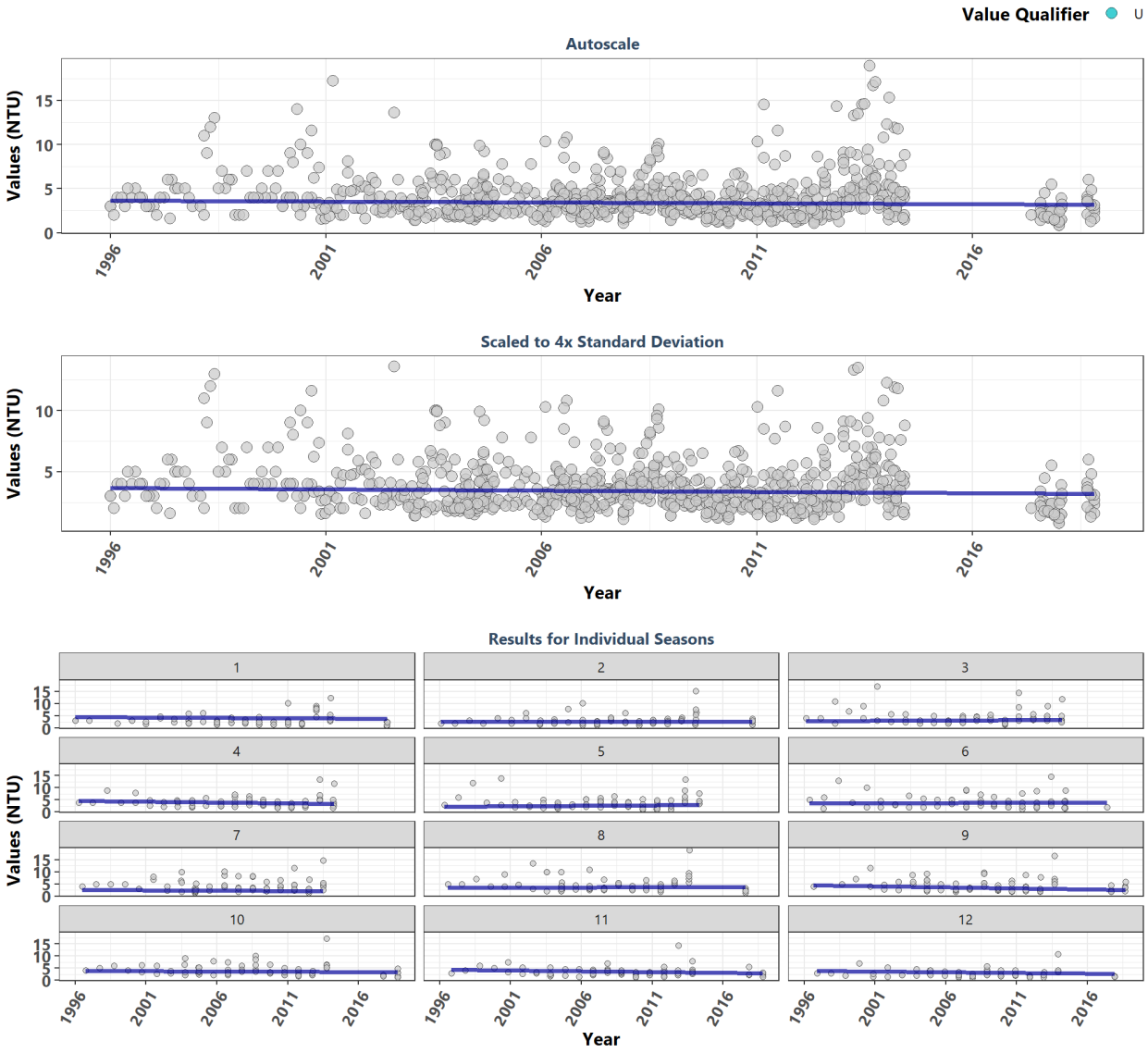
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1597	2.20	0.1127	0.0333	1.9817	6.0	0.0000	46.7	0	1
1	104	2.15	-0.0357	-0.0077	2.4615	0.8	0.4326	NA	NA	-1
2	111	2.00	0.2805	0.0917	1.5383	3.7	0.0002	NA	NA	1
3	113	2.20	0.0521	0.0152	2.0285	-0.6	0.5349	NA	NA	1
4	136	2.00	0.0096	0.0000	2.2000	0.6	0.5364	NA	NA	-1
5	147	2.00	0.0790	0.0253	2.0580	5.3	0.0000	NA	NA	1
6	148	2.40	0.1306	0.0520	2.0880	2.4	0.0180	NA	NA	1
7	209	2.40	-0.0395	-0.0100	2.2900	-0.8	0.4413	NA	NA	-1
8	144	2.40	0.2367	0.0824	1.3412	1.4	0.1580	NA	NA	1
9	143	2.18	0.0357	0.0100	1.9350	5.0	0.0000	NA	NA	1
10	127	2.20	0.2949	0.1000	1.3000	0.2	0.8737	NA	NA	1
11	111	2.24	0.1538	0.0462	1.3838	3.4	0.0007	NA	NA	1
12	104	1.80	0.2161	0.0615	1.6246	2.3	0.0202	NA	NA	1

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



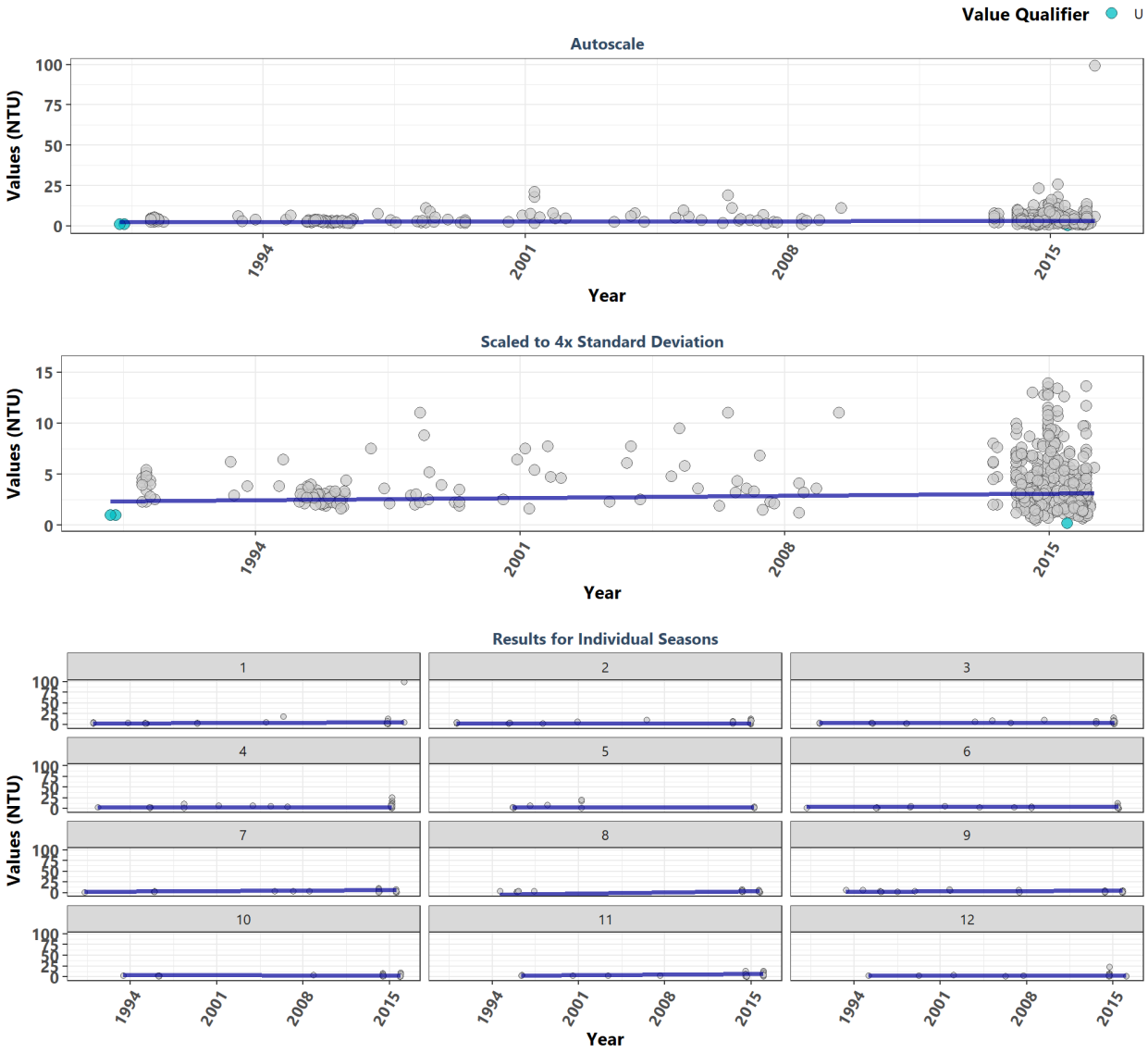
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3390	3.20	0.1862	0.1308	2.0471	15.5	0.0000	110.9	0	1
1	301	3.00	0.2572	0.1333	0.7900	3.6	0.0003	NA	NA	1
2	249	3.50	0.1192	0.0800	2.6000	8.8	0.0000	NA	NA	1
3	177	4.60	0.2404	0.1455	2.3000	5.1	0.0000	NA	NA	1
4	334	3.80	0.2578	0.3100	1.5000	3.3	0.0010	NA	NA	1
5	333	2.40	0.0262	0.0222	2.3556	11.3	0.0000	NA	NA	1
6	339	2.60	0.1110	0.0846	2.0692	0.7	0.4700	NA	NA	1
7	221	3.90	0.1384	0.0750	2.0250	5.3	0.0000	NA	NA	1
8	323	3.40	0.1205	0.1091	2.4909	3.2	0.0013	NA	NA	1
9	243	2.39	0.4145	0.3286	-1.2143	6.0	0.0000	NA	NA	2
10	289	3.00	0.3706	0.3000	0.5000	2.8	0.0046	NA	NA	2
11	280	2.80	0.0117	0.0111	4.1344	6.7	0.0000	NA	NA	1
12	301	4.29	0.2670	0.1273	1.2727	0.3	0.7609	NA	NA	1

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



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	829	3.20	-0.0426	-0.0200	3.6688	-1.8	0.0766	15.4	0.1653	0
1	71	2.70	-0.0377	-0.0250	4.4750	-0.2	0.8264	NA	NA	0
2	76	2.70	-0.0181	0.0000	2.7000	1.6	0.1002	NA	NA	0
3	62	3.80	0.0505	0.0333	2.8500	0.4	0.6964	NA	NA	0
4	71	3.60	-0.1444	-0.0667	4.6500	-0.4	0.7086	NA	NA	0
5	72	3.25	0.1284	0.0455	2.1545	0.6	0.5321	NA	NA	0
6	65	3.60	0.0344	0.0186	3.5769	-1.2	0.2240	NA	NA	0
7	68	3.60	-0.0754	-0.0250	2.6000	0.2	0.8066	NA	NA	0
8	70	4.20	0.0206	0.0143	3.4429	-0.5	0.6465	NA	NA	0
9	72	3.85	-0.2025	-0.0846	4.5150	-1.8	0.0726	NA	NA	0
10	75	3.50	-0.0306	-0.0134	3.7607	-2.6	0.0100	NA	NA	0
11	66	3.10	-0.1034	-0.0600	4.3800	-1.7	0.0958	NA	NA	0
12	61	2.30	-0.1403	-0.0567	3.8087	-0.9	0.3919	NA	NA	0

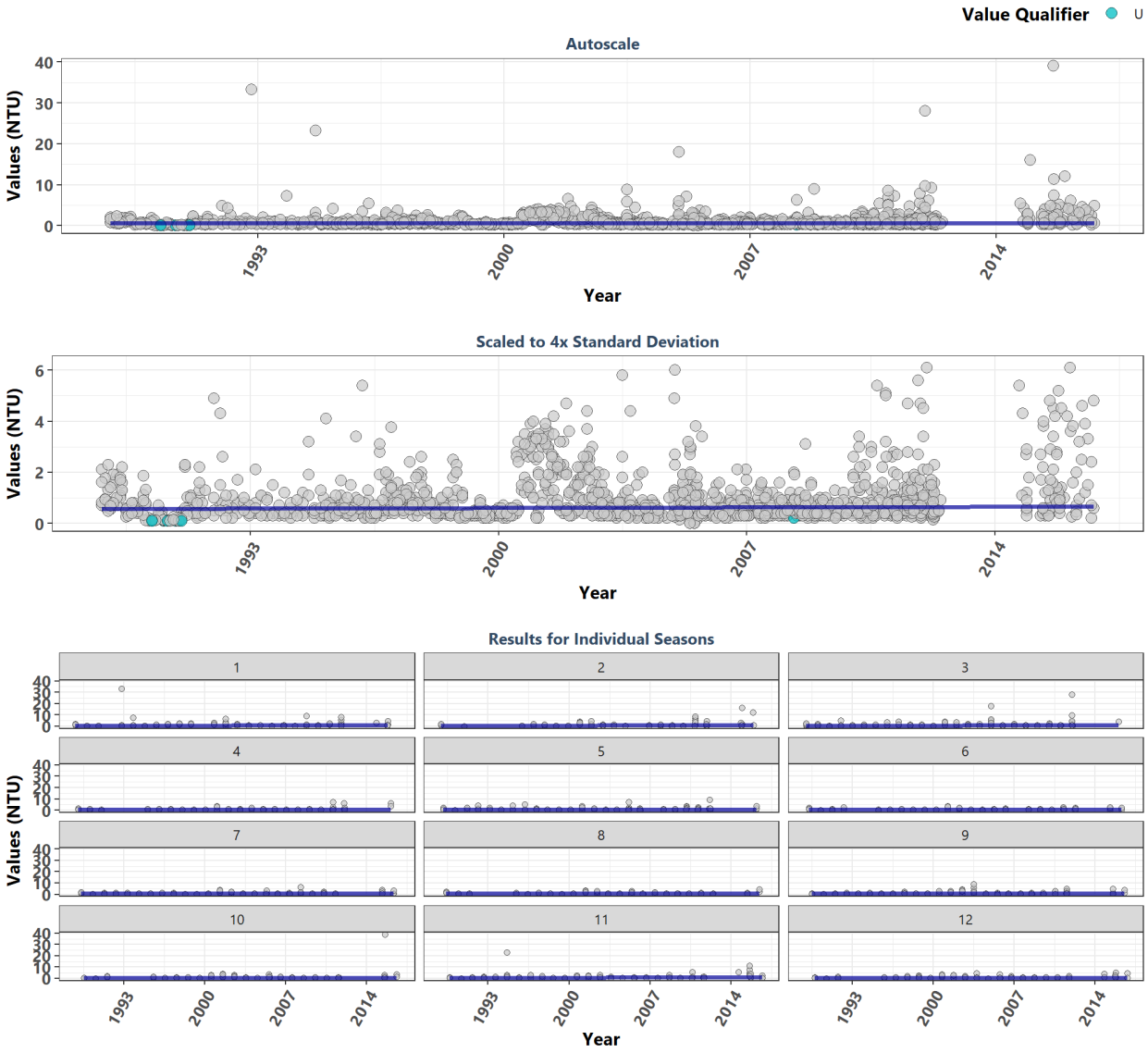
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	490	3.41	0.0694	0.0429	1.7996	2.1	0.0373	29.4	0.0019	1
1	40	3.41	0.2508	0.1807	-0.4064	0.8	0.4271	NA	NA	1
2	42	6.15	0.1216	0.0662	1.1143	3.1	0.0021	NA	NA	1
3	36	5.02	-0.0065	-0.0023	3.6669	2.4	0.0182	NA	NA	-1
4	36	6.08	0.0795	0.0291	2.5373	2.1	0.0316	NA	NA	1
5	13	3.00	0.0138	0.0350	2.4850	0.4	0.6540	NA	NA	1
6	37	3.10	-0.0275	-0.0251	4.4841	1.3	0.2059	NA	NA	-1
7	48	3.76	0.2048	0.2125	-0.2950	-0.3	0.7723	NA	NA	1
8	53	3.60	0.2357	0.5377	-12.2135	-0.1	0.9463	NA	NA	2
9	52	2.55	0.1026	0.1107	1.0071	-1.5	0.1464	NA	NA	1
10	48	2.29	-0.1826	-0.0883	4.8458	-2.0	0.0455	NA	NA	-1
11	55	3.38	0.2962	0.2667	-1.8502	2.8	0.0047	NA	NA	1
12	30	3.50	-0.1305	-0.0636	4.3986	0.1	0.9037	NA	NA	-1

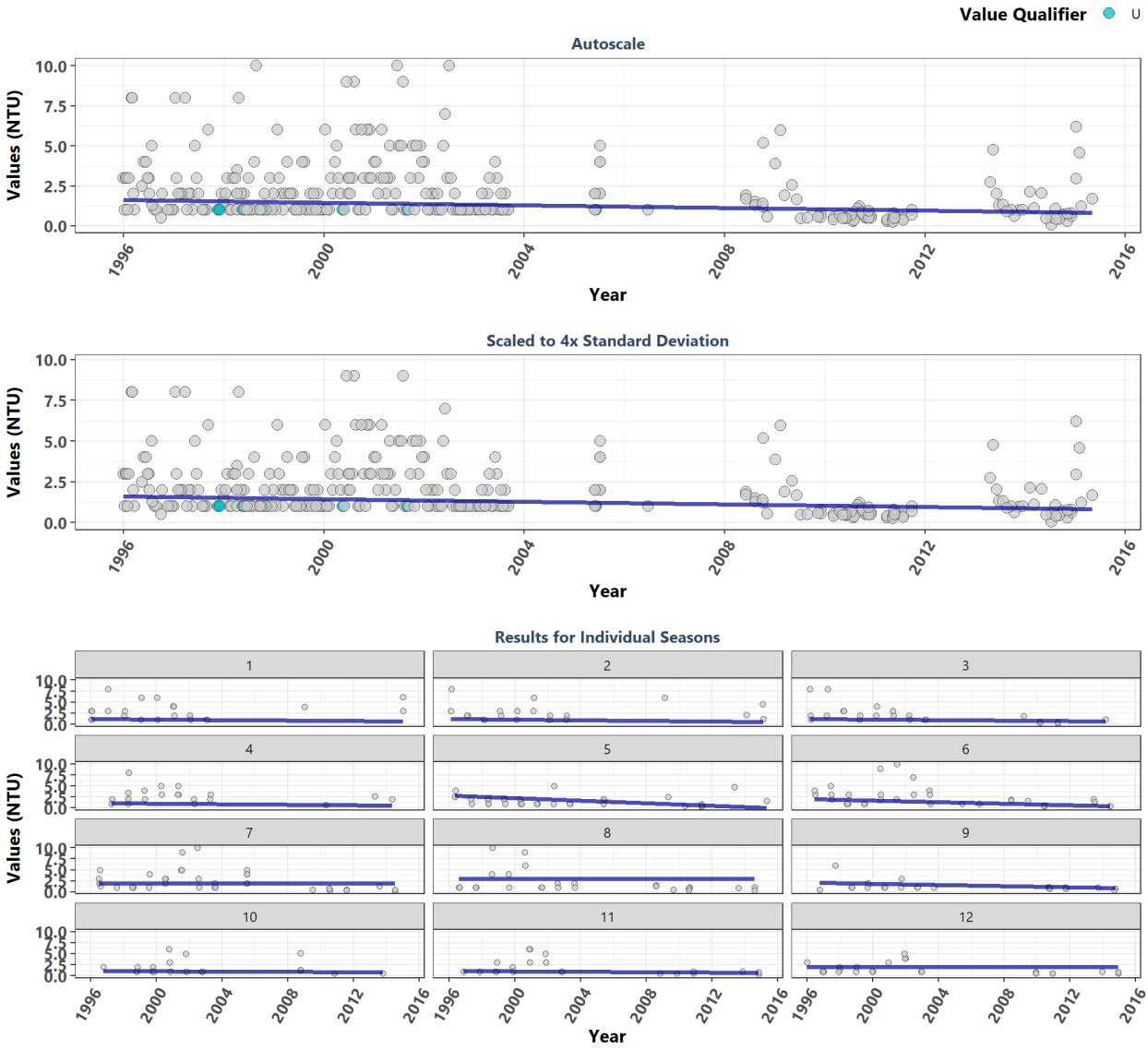
[a] $p < 0.00005$ appear as 0 due to rounding

Florida Keys National Marine Sanctuary



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1590	0.60	0.0680	0.0036	0.5500	3.9	0.0001	7.5	0.753	1
1	115	0.70	0.1189	0.0100	0.4100	1.5	0.1329	NA	NA	1
2	116	0.55	0.0688	0.0040	0.5279	1.7	0.0877	NA	NA	1
3	138	0.76	0.1658	0.0127	0.3243	0.9	0.3611	NA	NA	1
4	126	0.56	0.0683	0.0000	0.6000	2.8	0.0056	NA	NA	-1
5	160	0.60	0.0237	0.0000	0.5500	0.6	0.5336	NA	NA	-1
6	150	0.55	0.0138	0.0000	0.6000	0.4	0.6652	NA	NA	-1
7	129	0.60	0.0330	0.0000	0.6000	1.2	0.2486	NA	NA	-1
8	138	0.60	-0.0077	0.0000	0.6000	-0.1	0.8938	NA	NA	-1
9	143	0.60	0.0946	0.0083	0.5500	1.2	0.2207	NA	NA	1
10	124	0.60	0.1066	0.0100	0.3400	0.2	0.8208	NA	NA	1
11	137	0.60	0.0524	0.0030	0.7015	1.9	0.0549	NA	NA	1
12	114	0.60	0.1102	0.0100	0.4100	1.9	0.0592	NA	NA	1

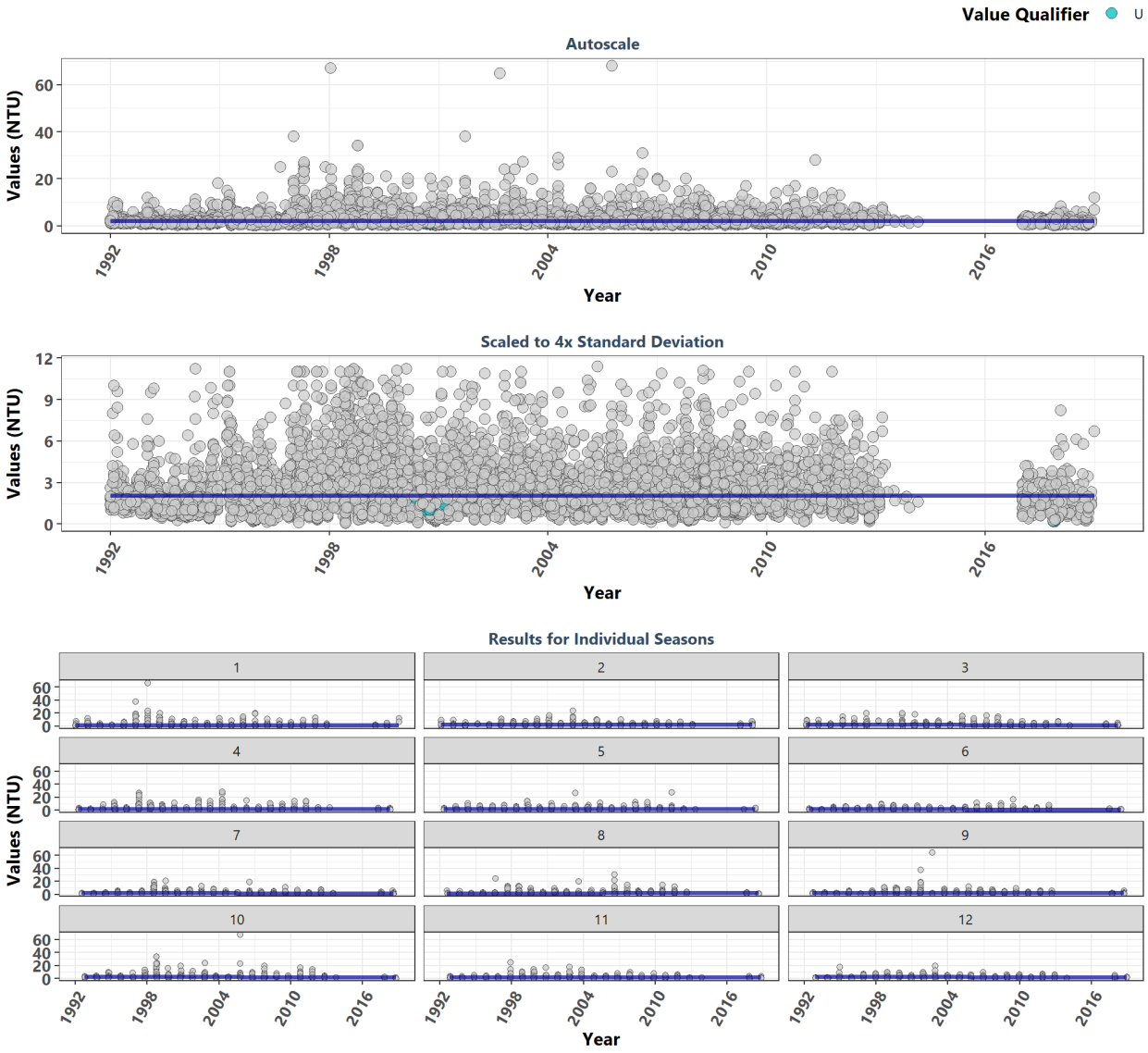
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	309	1.07	-0.2587	-0.0409	1.6127	-6.7	0.0000	14.2	0.2207	-1
1	22	2.97	-0.2714	-0.0311	1.2180	0.0	1.0000	NA	NA	-1
2	20	2.00	-0.3667	-0.0373	1.2236	-0.1	0.9469	NA	NA	-1
3	24	1.94	-0.2954	-0.0322	1.2254	-3.6	0.0004	NA	NA	-1
4	22	2.03	-0.1842	-0.0280	1.1540	-0.6	0.5622	NA	NA	-1
5	36	1.00	-0.5109	-0.1429	2.8021	-2.4	0.0152	NA	NA	-2
6	35	1.70	-0.3393	-0.0857	2.0000	-2.4	0.0167	NA	NA	-1
7	37	1.31	-0.0158	0.0000	2.0000	-3.0	0.0029	NA	NA	-1
8	26	1.00	0.0043	0.0000	2.9749	-2.2	0.0313	NA	NA	-1
9	22	1.00	-0.2790	-0.0643	2.0857	-1.9	0.0520	NA	NA	-1
10	20	1.00	-0.2857	-0.0157	1.1102	-1.2	0.2479	NA	NA	-1
11	21	1.00	-0.3333	-0.0232	1.1275	-2.4	0.0183	NA	NA	-1
12	24	1.00	-0.0909	0.0000	2.0298	-2.3	0.0194	NA	NA	-1

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

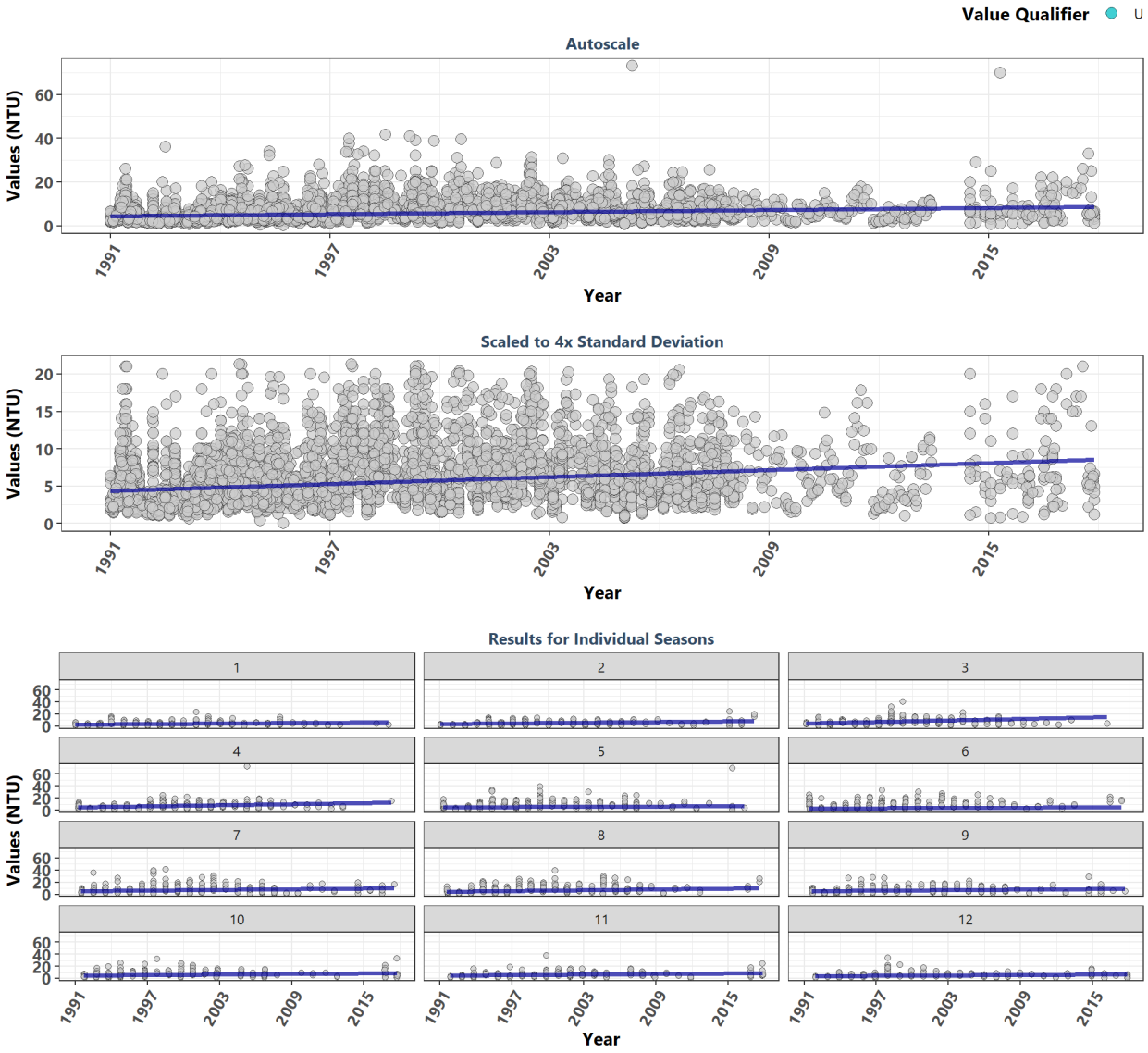
Gasparilla Sound-Charlotte Harbor Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	8786	2.00	-0.0149	0.0000	2.0728	-2.0	0.0469	88.4	0	-1
1	657	1.90	-0.0151	0.0000	2.0000	-2.4	0.0186	NA	NA	-1
2	774	1.90	0.0222	0.0067	2.3200	-4.0	0.0001	NA	NA	1
3	697	2.20	-0.0955	-0.0250	2.1750	-4.0	0.0001	NA	NA	-1
4	843	2.00	0.0237	0.0056	1.7556	-0.7	0.5113	NA	NA	1
5	801	1.80	0.0740	0.0231	1.5923	3.1	0.0017	NA	NA	1
6	830	1.80	-0.1244	-0.0316	2.0789	1.0	0.3049	NA	NA	-1
7	720	2.20	-0.0613	-0.0167	2.0667	-0.6	0.5291	NA	NA	-1
8	571	2.40	0.0590	0.0160	1.5600	0.8	0.4262	NA	NA	1
9	669	2.20	0.0871	0.0250	1.9500	3.4	0.0007	NA	NA	1
10	837	2.20	-0.0380	-0.0125	2.3125	-1.6	0.0992	NA	NA	-1
11	665	1.70	-0.0157	0.0000	2.2000	-4.8	0.0000	NA	NA	-1
12	722	1.72	-0.1015	-0.0333	2.5333	2.4	0.0175	NA	NA	-1

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

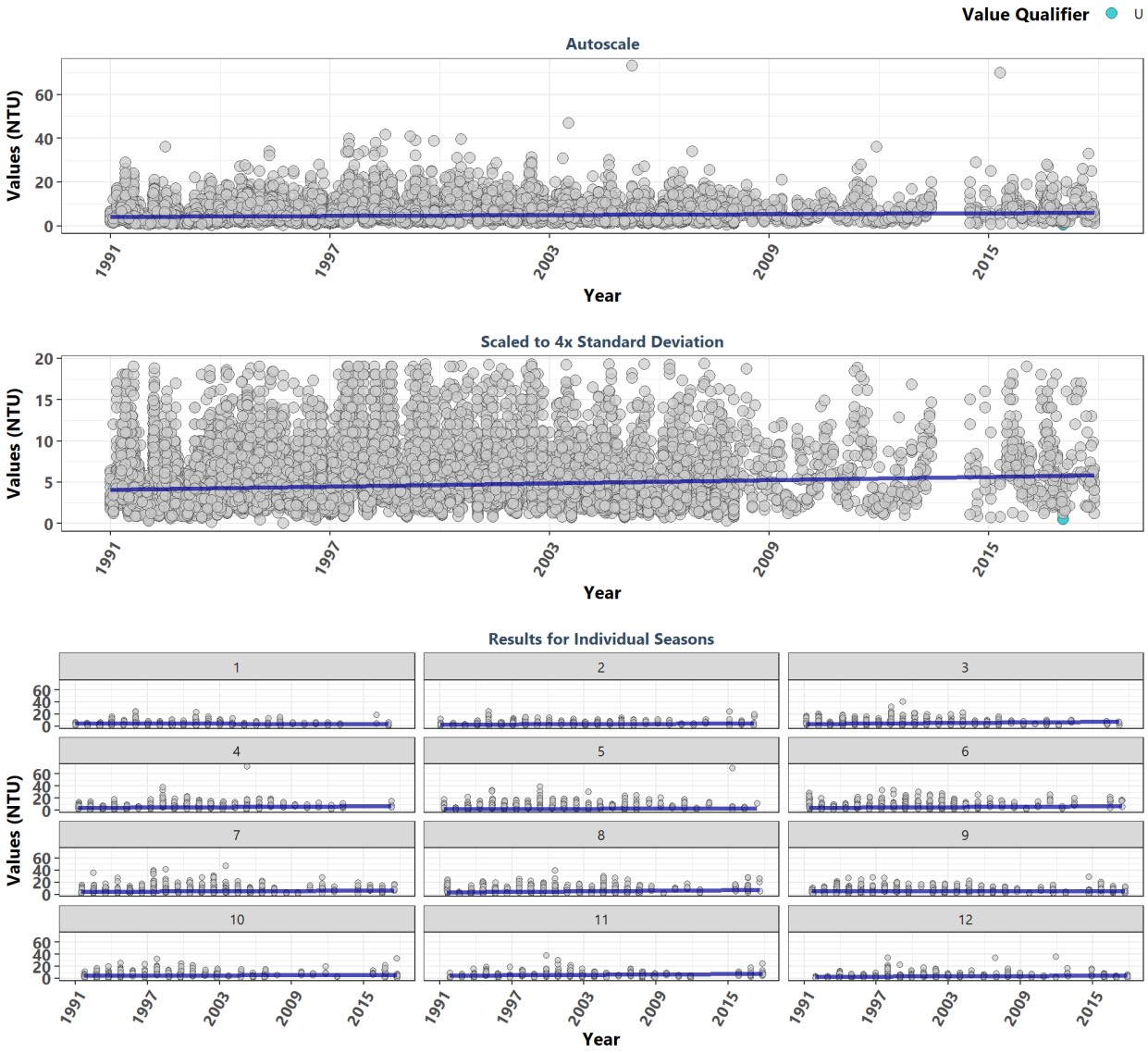
Guana River Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4769	5.10	0.1941	0.1533	4.3938	19.4	0.0000	54.4	0	1
1	453	3.30	0.2322	0.1222	2.9667	4.1	0.0001	NA	NA	1
2	375	3.70	0.3393	0.1964	3.5218	6.7	0.0000	NA	NA	1
3	316	5.15	0.3227	0.4325	4.5375	4.2	0.0000	NA	NA	1
4	480	5.50	0.2280	0.2766	4.9172	5.6	0.0000	NA	NA	1
5	436	6.30	0.0792	0.0571	5.3485	5.6	0.0000	NA	NA	1
6	373	6.30	0.1272	0.0750	2.7750	6.6	0.0000	NA	NA	1
7	504	6.16	0.1796	0.2000	5.1000	4.5	0.0000	NA	NA	1
8	443	6.70	0.2314	0.2129	4.2264	10.2	0.0000	NA	NA	1
9	326	4.86	0.1347	0.1362	5.2064	6.3	0.0000	NA	NA	1
10	451	5.75	0.1693	0.1364	4.5455	2.5	0.0117	NA	NA	1
11	345	4.70	0.1582	0.1500	4.2500	9.4	0.0000	NA	NA	1
12	267	4.60	0.1769	0.1100	3.9409	4.3	0.0000	NA	NA	1

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

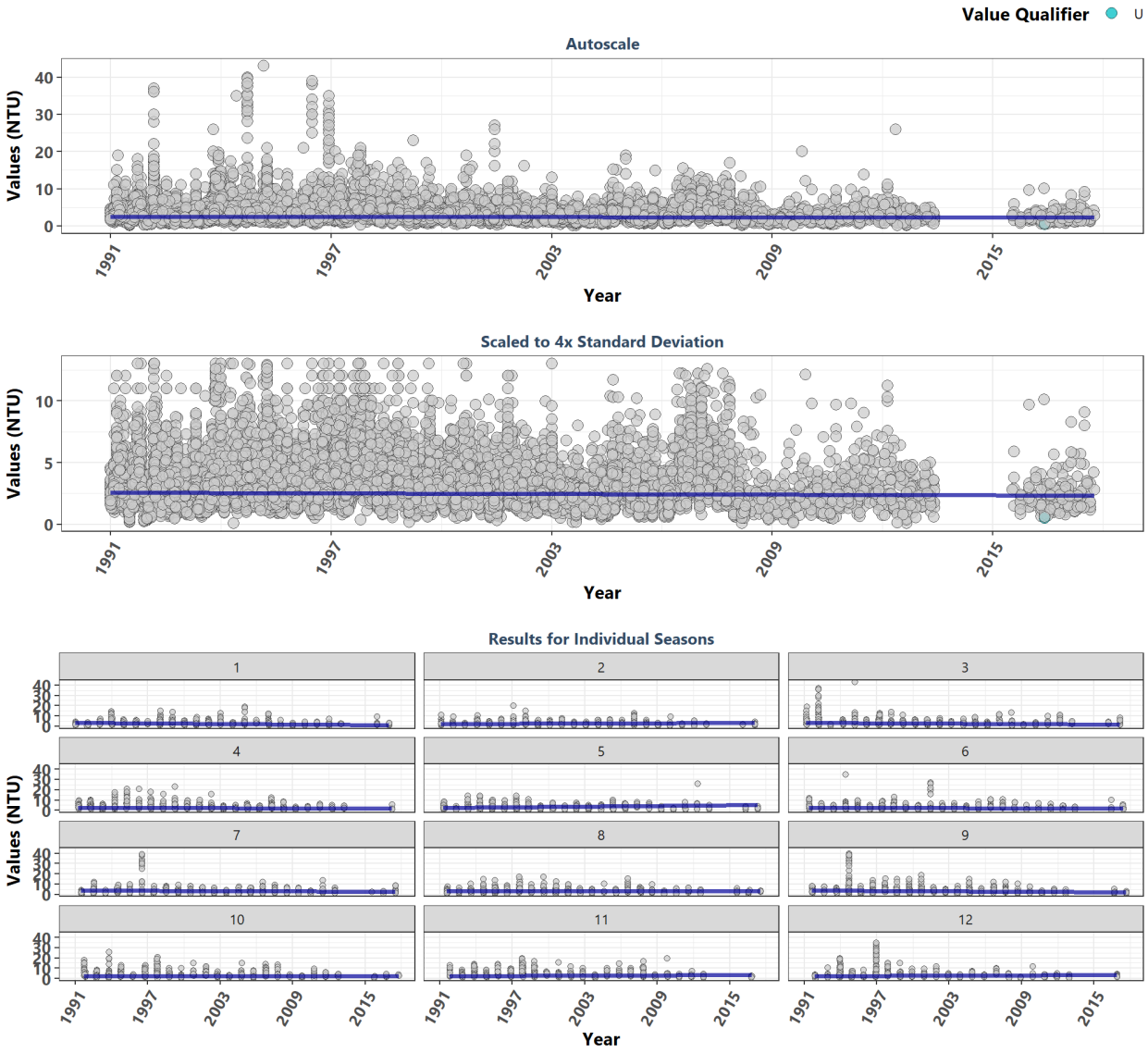
Guana Tolomato Matanzas National Estuarine Research Reserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	14227	4.40	0.1021	0.0667	4.0554	17.2	0.0000	125.1	0	1
1	1278	2.70	-0.0456	-0.0364	4.7182	4.5	0.0000	NA	NA	-1
2	1061	3.10	0.1577	0.0800	2.5400	7.7	0.0000	NA	NA	1
3	1247	4.50	0.1888	0.1333	3.9000	-2.4	0.0154	NA	NA	1
4	1154	4.80	0.1129	0.0985	4.1108	6.5	0.0000	NA	NA	1
5	1126	4.80	0.0839	0.0357	2.4143	5.7	0.0000	NA	NA	1
6	1173	5.30	0.1276	0.1000	4.0000	5.5	0.0000	NA	NA	1
7	1581	5.00	0.1025	0.0951	4.2294	2.5	0.0135	NA	NA	1
8	1375	4.80	0.1964	0.1167	3.6833	5.7	0.0000	NA	NA	1
9	1234	4.70	0.0502	0.0333	4.9000	10.0	0.0000	NA	NA	1
10	1162	5.10	0.0414	0.0333	4.7667	2.6	0.0101	NA	NA	1
11	995	4.50	0.1076	0.1010	4.6942	9.3	0.0000	NA	NA	1
12	841	3.48	0.1680	0.0833	2.9011	7.3	0.0000	NA	NA	1

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

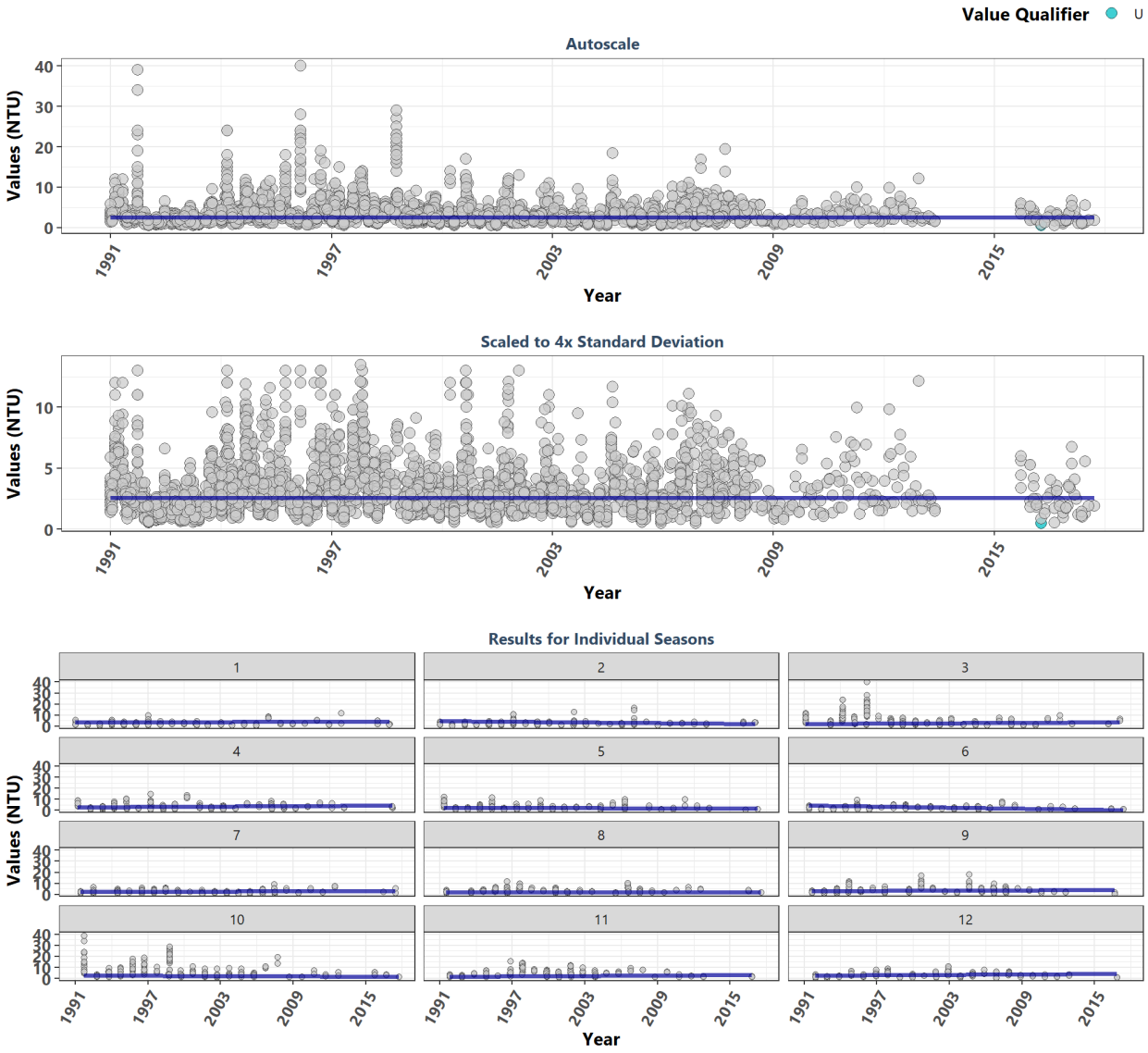
Indian River-Malabar to Vero Beach Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	13798	2.80	-0.0207	-0.0083	2.5591	-4.2	0.0000	407.3	0	-1
1	1098	2.50	-0.2202	-0.1000	3.5400	5.1	0.0000	NA	NA	-1
2	1161	2.40	0.1060	0.0400	2.1200	5.4	0.0000	NA	NA	1
3	1009	2.94	-0.1556	-0.0625	3.4375	-10.5	0.0000	NA	NA	-1
4	1408	3.00	-0.0534	-0.0195	2.3974	-8.8	0.0000	NA	NA	-1
5	1051	2.40	0.1605	0.1000	2.5000	-5.0	0.0000	NA	NA	1
6	1075	2.30	-0.1028	-0.0364	2.6182	-2.6	0.0085	NA	NA	-1
7	1322	2.38	-0.0809	-0.0364	3.6182	1.3	0.2015	NA	NA	-1
8	1097	2.70	0.0175	0.0067	3.0600	5.7	0.0000	NA	NA	1
9	1309	3.10	-0.1254	-0.0600	3.8200	1.0	0.3418	NA	NA	-1
10	1360	3.40	0.0234	0.0057	2.3373	-4.5	0.0000	NA	NA	1
11	1022	3.40	0.1033	0.0388	2.2287	-6.0	0.0000	NA	NA	1
12	886	3.00	0.1154	0.0462	2.4692	7.2	0.0000	NA	NA	1

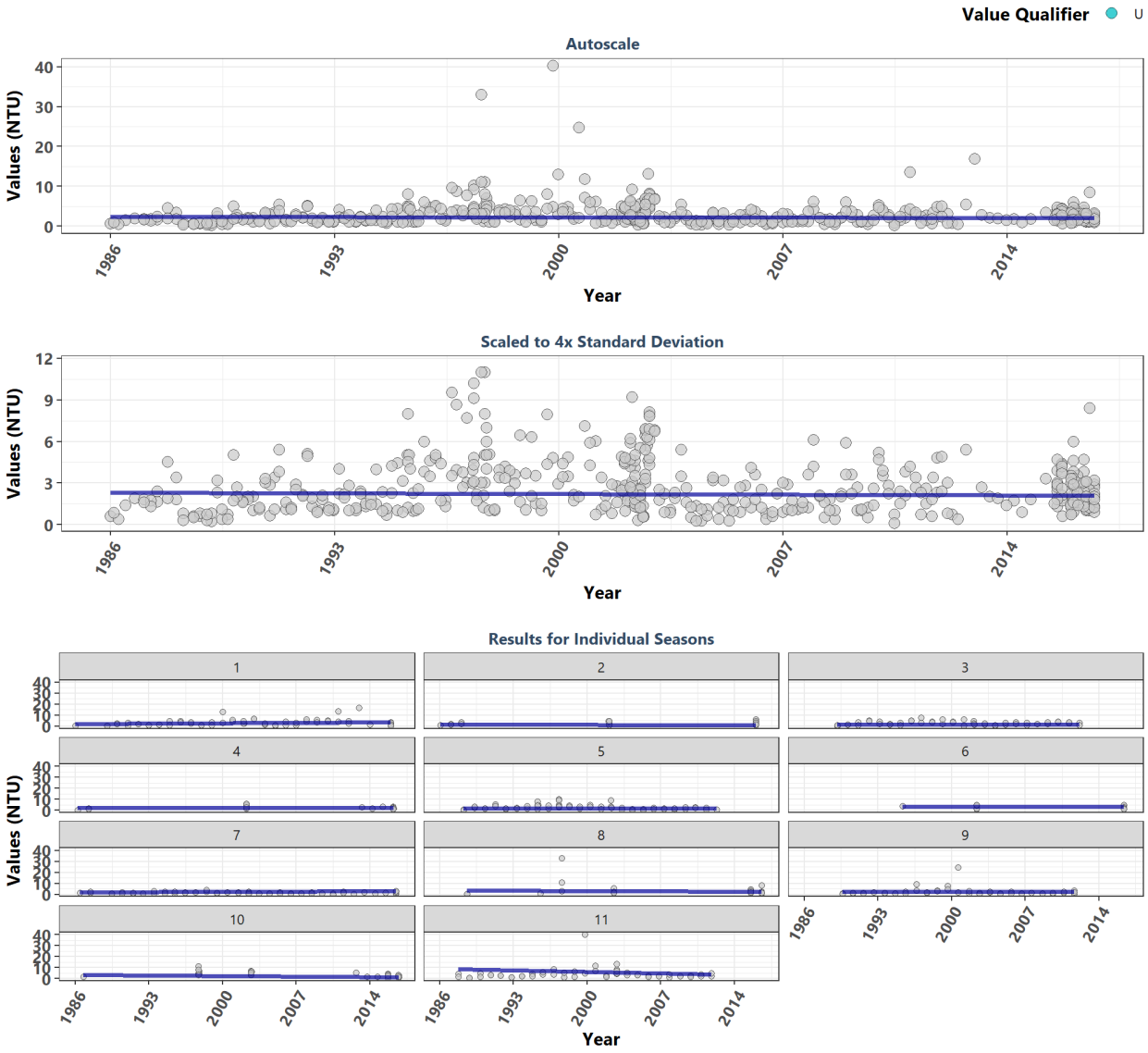
[a] $p < 0.00005$ appear as 0 due to rounding

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve



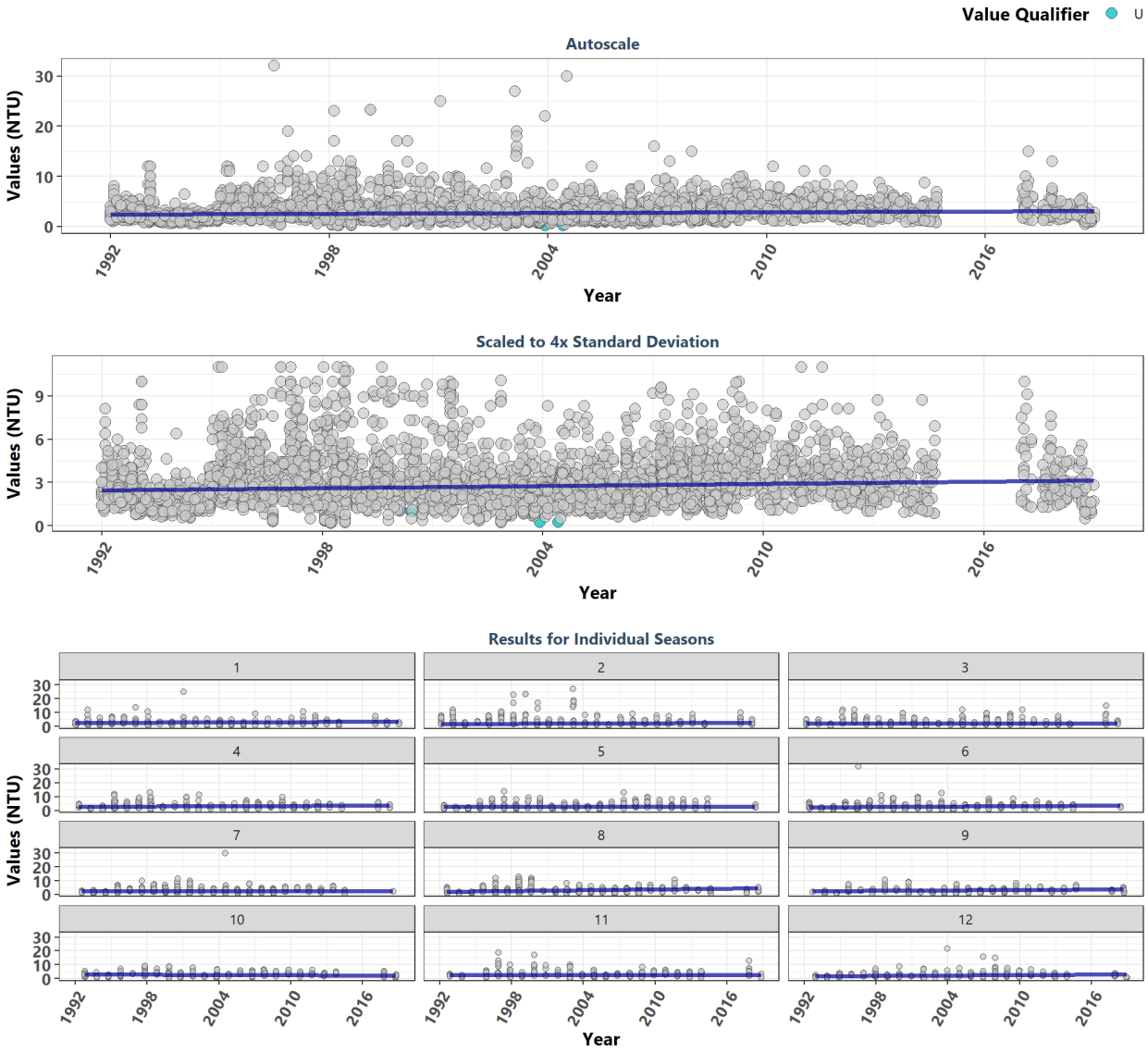
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3744	2.6	-0.0009	0.0000	2.5767	-1.1	0.2552	158	0	0
1	343	2.0	0.0396	0.0222	3.5222	-2.6	0.0093	NA	NA	0
2	289	2.5	-0.1464	-0.1000	4.7000	4.9	0.0000	NA	NA	0
3	313	3.1	0.1920	0.0591	2.1457	-7.2	0.0000	NA	NA	0
4	388	2.5	0.1167	0.0500	2.8000	1.8	0.0709	NA	NA	0
5	330	1.9	-0.0937	-0.0231	2.1385	-0.4	0.6842	NA	NA	0
6	246	2.0	-0.2729	-0.1625	4.4000	4.1	0.0000	NA	NA	0
7	377	2.2	0.0612	0.0200	2.3400	-3.8	0.0002	NA	NA	0
8	258	3.1	-0.0150	0.0000	1.9000	2.1	0.0329	NA	NA	0
9	271	3.1	0.0888	0.0386	2.9072	2.9	0.0041	NA	NA	0
10	372	3.9	-0.1290	-0.0438	2.6375	-4.2	0.0000	NA	NA	0
11	277	3.7	0.1733	0.0556	1.7222	1.0	0.3241	NA	NA	0
12	280	2.9	0.1244	0.0640	2.5159	3.1	0.0019	NA	NA	0

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



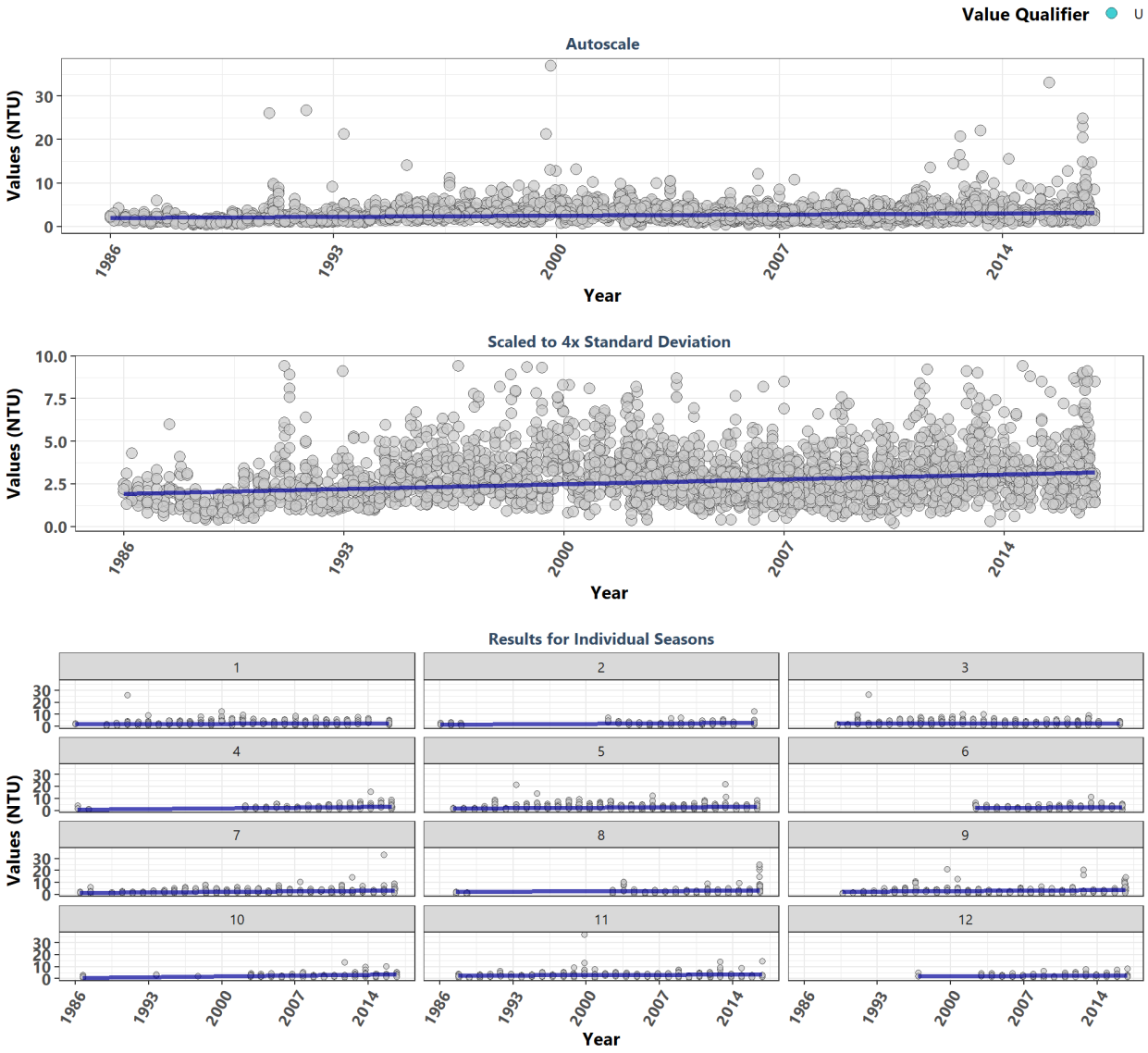
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	474	2.20	-0.0412	-0.0071	2.3231	-1.0	0.3248	28.4	0.0016	0
1	63	2.60	0.2048	0.0414	2.1379	1.7	0.0934	NA	NA	0
2	21	2.80	-0.0281	-0.0050	1.2750	1.4	0.1741	NA	NA	0
3	50	2.61	0.0231	0.0000	1.4000	0.1	0.9133	NA	NA	0
4	22	2.00	-0.0366	-0.0077	2.3231	-0.1	0.8809	NA	NA	0
5	59	2.30	-0.0260	0.0000	2.0000	-2.4	0.0151	NA	NA	0
6	16	2.70	0.0338	0.0158	3.0632	-0.9	0.3570	NA	NA	0
7	65	1.40	0.1449	0.0370	1.9710	0.3	0.7893	NA	NA	0
8	31	2.10	-0.1583	-0.0357	3.5214	-0.3	0.7781	NA	NA	0
9	49	1.20	0.0114	0.0039	2.5518	-0.3	0.7823	NA	NA	0
10	41	3.30	-0.2174	-0.0750	3.2000	-4.3	0.0000	NA	NA	0
11	57	3.30	-0.4512	-0.1929	8.7000	0.4	0.7144	NA	NA	0

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



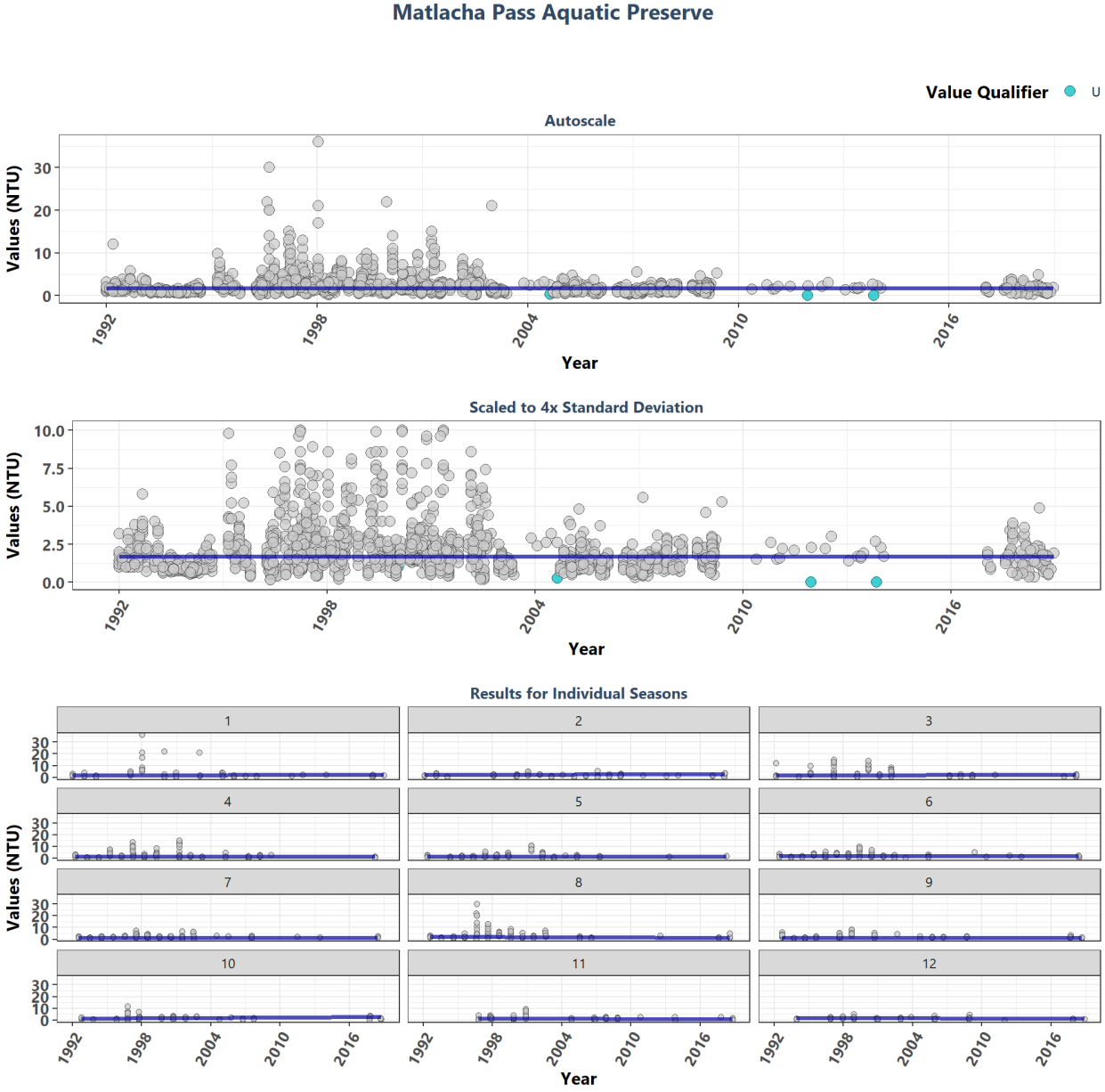
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4406	2.60	0.0758	0.0250	2.4471	7.1	0.0000	74	0	1
1	346	2.10	0.1043	0.0357	2.5143	0.2	0.8480	NA	NA	1
2	323	2.80	0.1261	0.0400	1.6400	-2.0	0.0471	NA	NA	1
3	302	2.70	0.0069	0.0000	2.1000	2.8	0.0051	NA	NA	-1
4	410	2.90	0.1005	0.0368	2.5684	3.1	0.0023	NA	NA	1
5	340	3.00	0.0253	0.0083	2.6667	7.1	0.0000	NA	NA	1
6	425	2.80	0.1077	0.0421	2.2789	3.2	0.0013	NA	NA	1
7	476	2.70	0.0313	0.0100	2.3800	-1.3	0.1883	NA	NA	1
8	433	2.75	0.2563	0.0933	2.0667	0.8	0.4299	NA	NA	1
9	344	2.20	0.1289	0.0400	2.5800	3.5	0.0004	NA	NA	1
10	419	2.90	-0.0739	-0.0286	3.0571	4.0	0.0001	NA	NA	-1
11	297	2.50	-0.0403	-0.0143	2.8286	0.8	0.4203	NA	NA	-1
12	291	2.20	0.1784	0.0400	1.8000	4.6	0.0000	NA	NA	1

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



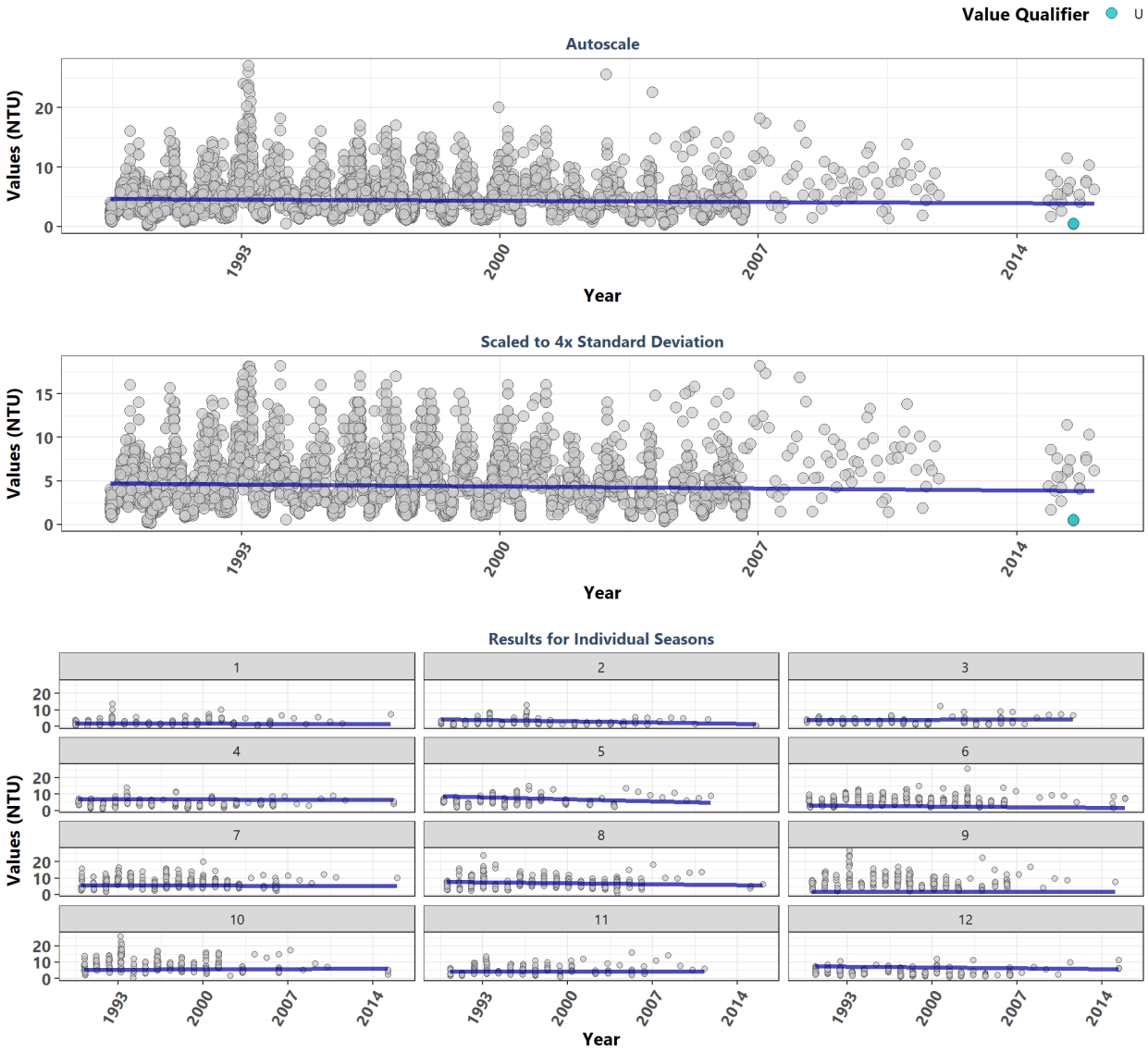
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3716	2.60	0.1690	0.0400	1.9260	14.7	0.0000	75	0	1
1	464	3.00	0.0996	0.0200	1.9400	6.0	0.0000	NA	NA	1
2	137	2.50	0.1954	0.0389	1.6889	2.3	0.0214	NA	NA	1
3	453	2.90	0.0641	0.0136	2.2818	3.4	0.0007	NA	NA	1
4	163	3.04	0.2867	0.0708	0.8708	6.3	0.0000	NA	NA	1
5	475	2.80	0.2846	0.0522	1.5530	5.1	0.0000	NA	NA	1
6	135	2.70	0.0309	0.0082	2.2618	2.0	0.0409	NA	NA	1
7	504	2.44	0.1185	0.0558	1.4729	9.6	0.0000	NA	NA	1
8	151	2.50	0.1569	0.0414	2.1371	5.2	0.0000	NA	NA	1
9	469	2.35	0.1873	0.0500	2.1750	6.3	0.0000	NA	NA	1
10	188	2.40	0.3334	0.1050	0.5200	2.0	0.0418	NA	NA	1
11	454	2.50	0.1062	0.0314	2.3971	2.0	0.0410	NA	NA	1
12	123	2.45	0.1325	0.0280	1.9120	0.5	0.6124	NA	NA	1

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



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1827	1.60	0.0015	0.0000	1.6980	0.2	0.8667	24.6	0.0106	0
1	116	1.30	0.0616	0.0222	1.8667	-0.8	0.4146	NA	NA	0
2	175	1.40	0.0610	0.0185	2.0077	-0.4	0.7009	NA	NA	0
3	141	1.80	0.0650	0.0167	1.9067	0.2	0.8076	NA	NA	0
4	181	1.60	0.0431	0.0063	1.4559	-1.0	0.3320	NA	NA	0
5	185	1.80	-0.0194	0.0000	1.4000	3.0	0.0026	NA	NA	0
6	184	2.00	0.0138	0.0000	1.8000	1.2	0.2116	NA	NA	0
7	208	1.80	-0.0511	-0.0080	1.3720	-2.1	0.0318	NA	NA	0
8	158	2.10	-0.0995	-0.0250	1.9750	1.1	0.2543	NA	NA	0
9	111	1.99	-0.0851	-0.0141	1.4406	1.0	0.3085	NA	NA	0
10	133	1.50	0.1482	0.0500	1.5000	0.7	0.4610	NA	NA	0
11	120	1.50	-0.1001	-0.0273	1.7182	-1.6	0.1001	NA	NA	0
12	115	1.30	-0.0483	-0.0111	1.6778	-1.4	0.1731	NA	NA	0

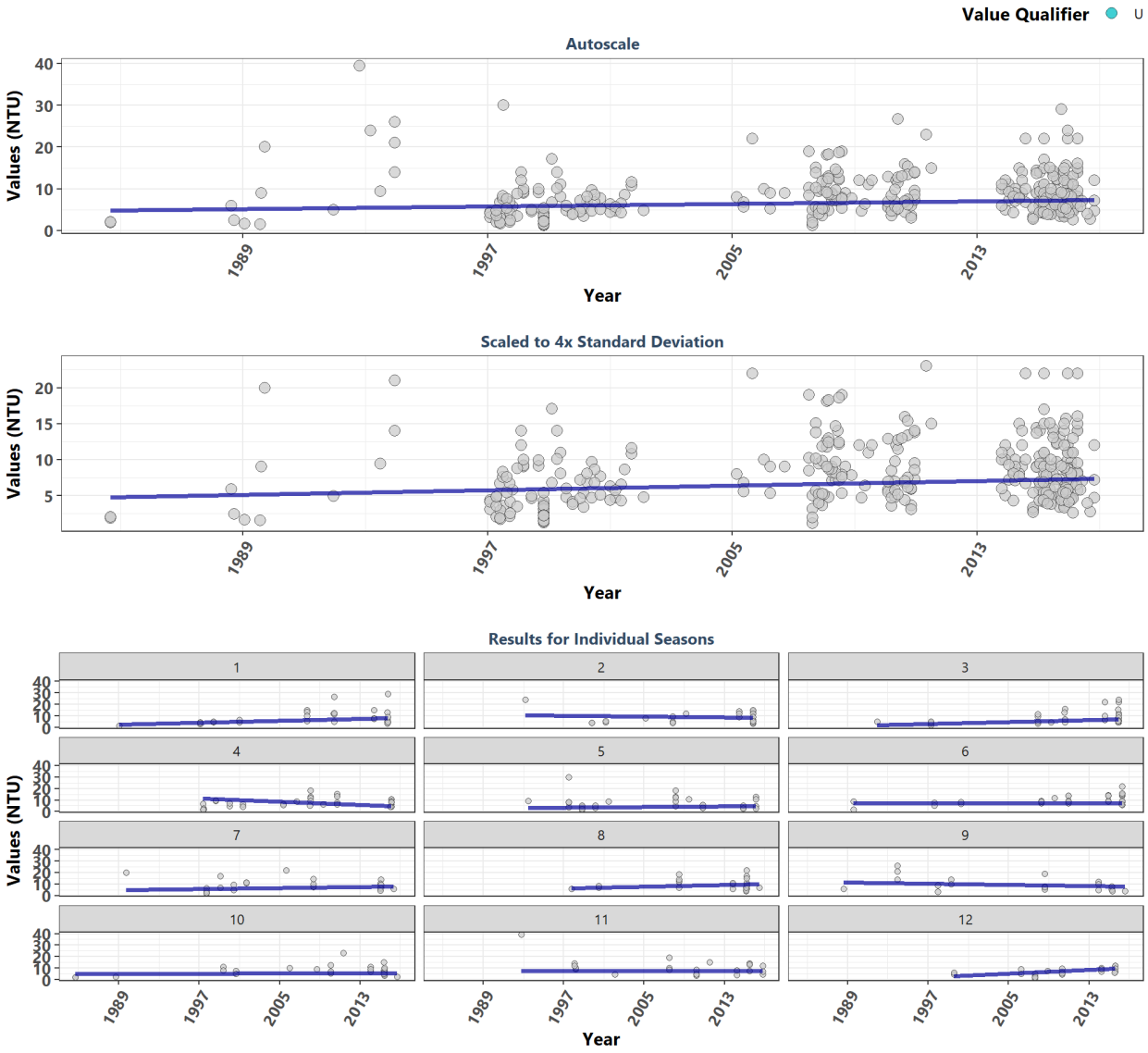
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3400	4.60	-0.0585	-0.0308	4.7881	-5.0	0.0000	48.4	0	-1
1	298	1.80	-0.0622	-0.0143	1.9571	-0.4	0.6831	NA	NA	-1
2	240	1.80	-0.1896	-0.1000	4.4800	-1.4	0.1502	NA	NA	-1
3	289	2.60	0.0388	0.0167	3.8000	-4.9	0.0000	NA	NA	1
4	267	4.00	-0.0115	-0.0077	6.7346	0.9	0.3441	NA	NA	-1
5	199	5.40	-0.1502	-0.1667	9.1667	-0.3	0.7411	NA	NA	-1
6	346	5.50	-0.1913	-0.0538	3.1923	1.4	0.1740	NA	NA	-1
7	333	7.10	-0.0157	-0.0125	5.5125	-2.1	0.0340	NA	NA	-1
8	326	6.65	-0.0777	-0.0750	8.0000	-0.3	0.7561	NA	NA	-1
9	329	7.10	-0.0158	0.0000	1.8000	-2.0	0.0462	NA	NA	-1
10	289	7.50	0.0489	0.0367	5.0961	-3.8	0.0001	NA	NA	1
11	209	4.40	0.0059	0.0000	4.4000	0.1	0.8997	NA	NA	-1
12	275	3.38	-0.0734	-0.0733	7.9067	-4.7	0.0000	NA	NA	-1

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

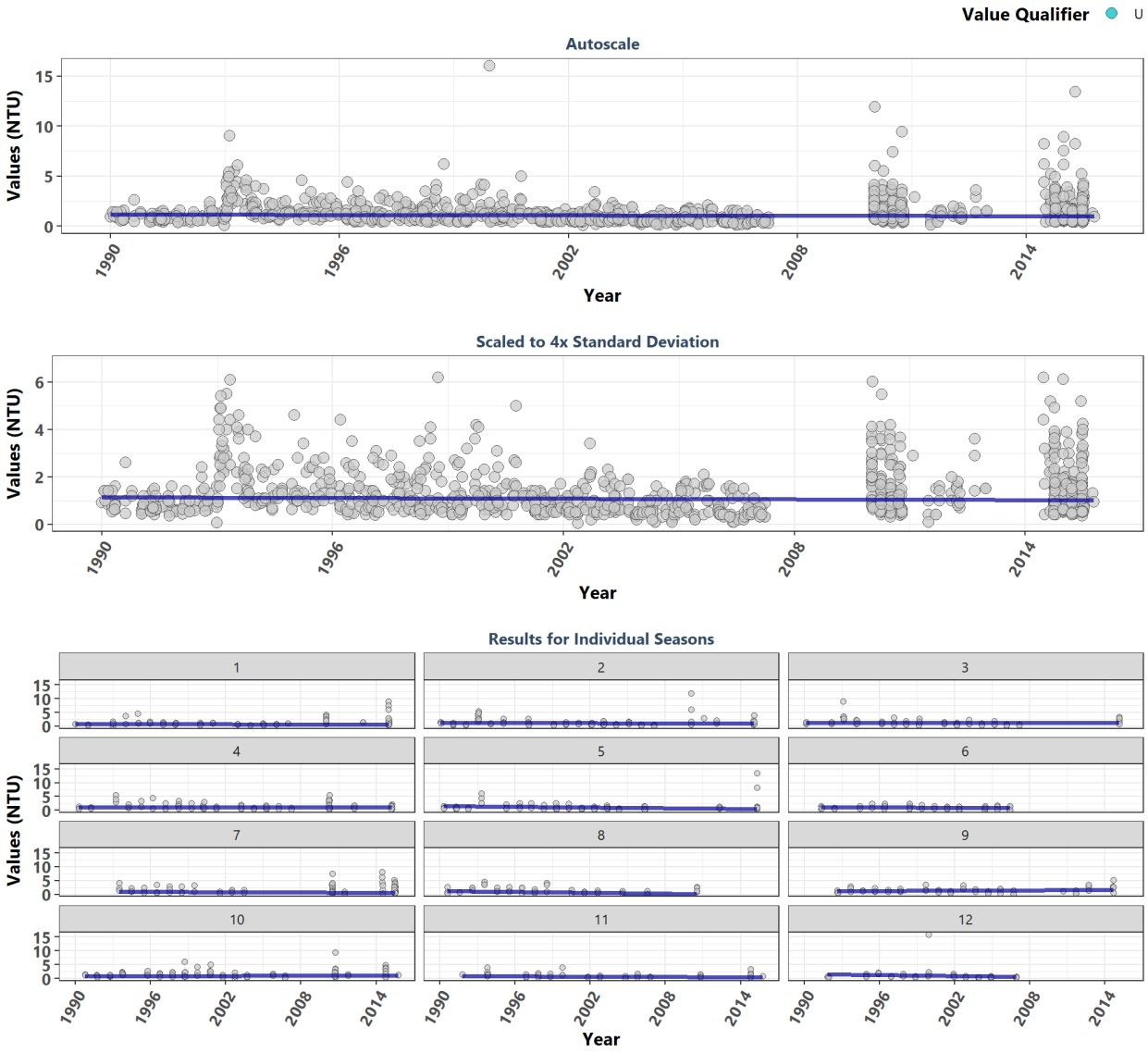
Nassau River-St. Johns River Marshes Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	344	7.56	0.0908	0.1064	3.4207	2.8	0.0054	30	0.0016	1
1	28	6.79	0.2698	0.2663	-1.7274	2.0	0.0416	NA	NA	1
2	25	5.70	-0.0615	-0.1178	12.9247	0.3	0.7303	NA	NA	-1
3	36	6.85	0.2460	0.2690	-2.6976	2.3	0.0240	NA	NA	1
4	34	7.00	-0.3857	-0.4664	21.8163	1.3	0.1819	NA	NA	-1
5	41	3.80	0.0988	0.0843	1.7779	0.9	0.3544	NA	NA	1
6	31	8.92	-0.0121	-0.0044	7.2980	2.6	0.0100	NA	NA	-1
7	24	7.67	0.1586	0.1638	2.0857	0.1	0.8993	NA	NA	1
8	26	8.80	0.3183	0.2500	0.9187	-0.5	0.6448	NA	NA	1
9	21	7.82	-0.2067	-0.1612	13.9972	-2.5	0.0141	NA	NA	-1
10	32	7.15	0.0500	0.0270	4.7557	-0.1	0.9321	NA	NA	1
11	25	9.00	0.0217	0.0110	7.3528	-1.4	0.1491	NA	NA	1
12	21	6.96	0.4286	0.5110	-8.8812	2.7	0.0062	NA	NA	1

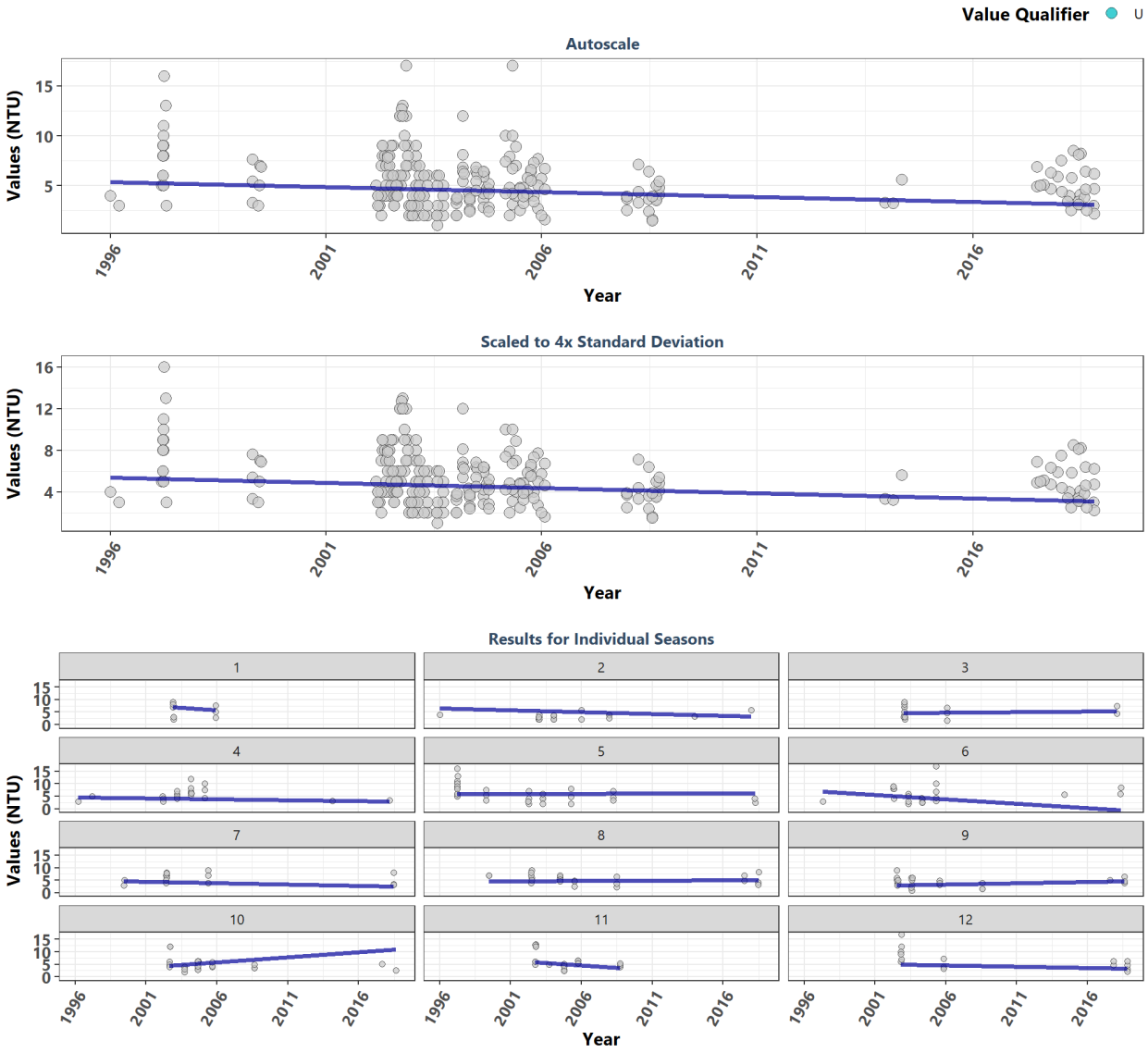
[a] $p < 0.00005$ appear as 0 due to rounding

Nature Coast Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1023	1.00	-0.0687	-0.0050	1.1427	-2.2	0.0303	29.9	0.0016	-1
1	116	1.02	-0.0832	-0.0050	0.8150	1.9	0.0627	NA	NA	-1
2	105	0.90	-0.0686	-0.0079	1.1454	-2.2	0.0256	NA	NA	-1
3	63	1.10	0.0078	0.0000	1.2000	-2.8	0.0050	NA	NA	-1
4	134	1.00	0.0098	0.0016	1.0684	-1.2	0.2337	NA	NA	1
5	89	0.76	-0.2417	-0.0400	1.4600	-1.2	0.2467	NA	NA	-1
6	67	0.76	-0.1740	-0.0195	1.1638	-1.5	0.1256	NA	NA	-1
7	111	1.10	-0.1473	-0.0200	1.1400	0.2	0.8754	NA	NA	-1
8	45	1.20	-0.3631	-0.0500	1.2500	-1.8	0.0772	NA	NA	-1
9	58	1.35	0.1077	0.0253	1.0719	1.2	0.2334	NA	NA	1
10	130	1.20	0.1147	0.0126	0.7637	0.1	0.8958	NA	NA	1
11	54	0.91	-0.1275	-0.0230	0.9667	-1.9	0.0617	NA	NA	-1
12	51	0.70	-0.1828	-0.0500	1.5500	-3.8	0.0002	NA	NA	-1

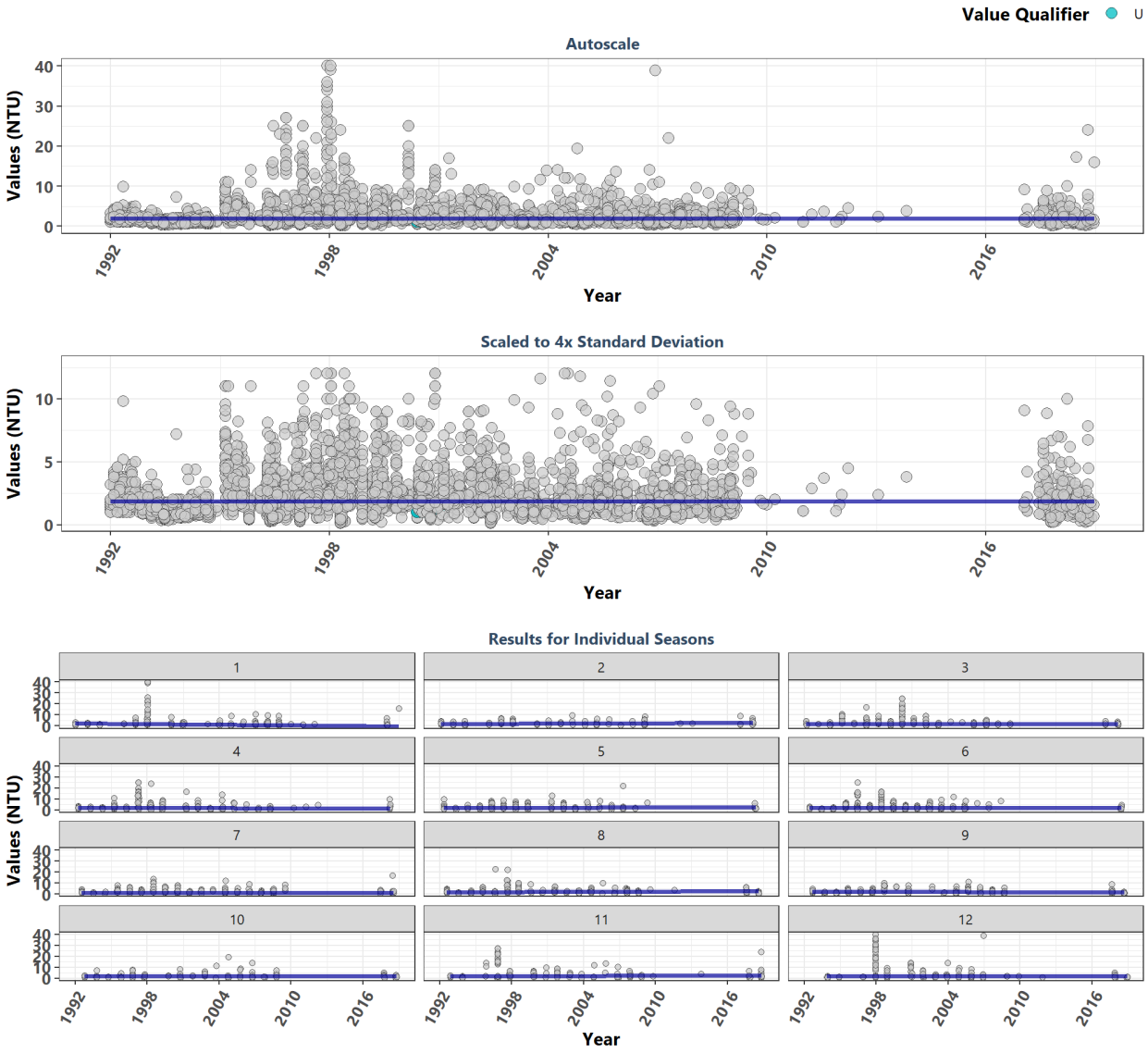
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	254	5.00	-0.1248	-0.1000	5.3900	-3.3	0.0009	29.4	0.002	-1
1	9	NA	-0.5048	-0.3310	9.1791	NA	NA	NA	NA	NA
2	20	3.10	-0.2867	-0.1462	6.5692	1.2	0.2292	NA	NA	-1
3	19	4.60	0.0308	0.0406	4.3250	0.0	0.9631	NA	NA	1
4	21	5.00	-0.1140	-0.0667	4.4667	1.8	0.0771	NA	NA	-1
5	34	5.20	0.0667	0.0062	5.9625	-3.6	0.0003	NA	NA	1
6	26	4.65	-0.4189	-0.3551	7.3303	0.2	0.8386	NA	NA	-1
7	15	6.00	-0.2208	-0.1100	5.0800	0.3	0.7489	NA	NA	-1
8	25	5.40	0.0117	0.0300	4.3900	-2.0	0.0428	NA	NA	1
9	27	4.00	0.1895	0.1000	2.3000	-0.8	0.3956	NA	NA	1
10	22	4.20	0.2762	0.4000	1.8000	-1.4	0.1485	NA	NA	1
11	21	5.40	-0.2810	-0.4000	8.6000	-1.8	0.0678	NA	NA	-1
12	15	6.20	-0.0556	-0.1000	5.7000	-2.7	0.0066	NA	NA	-1

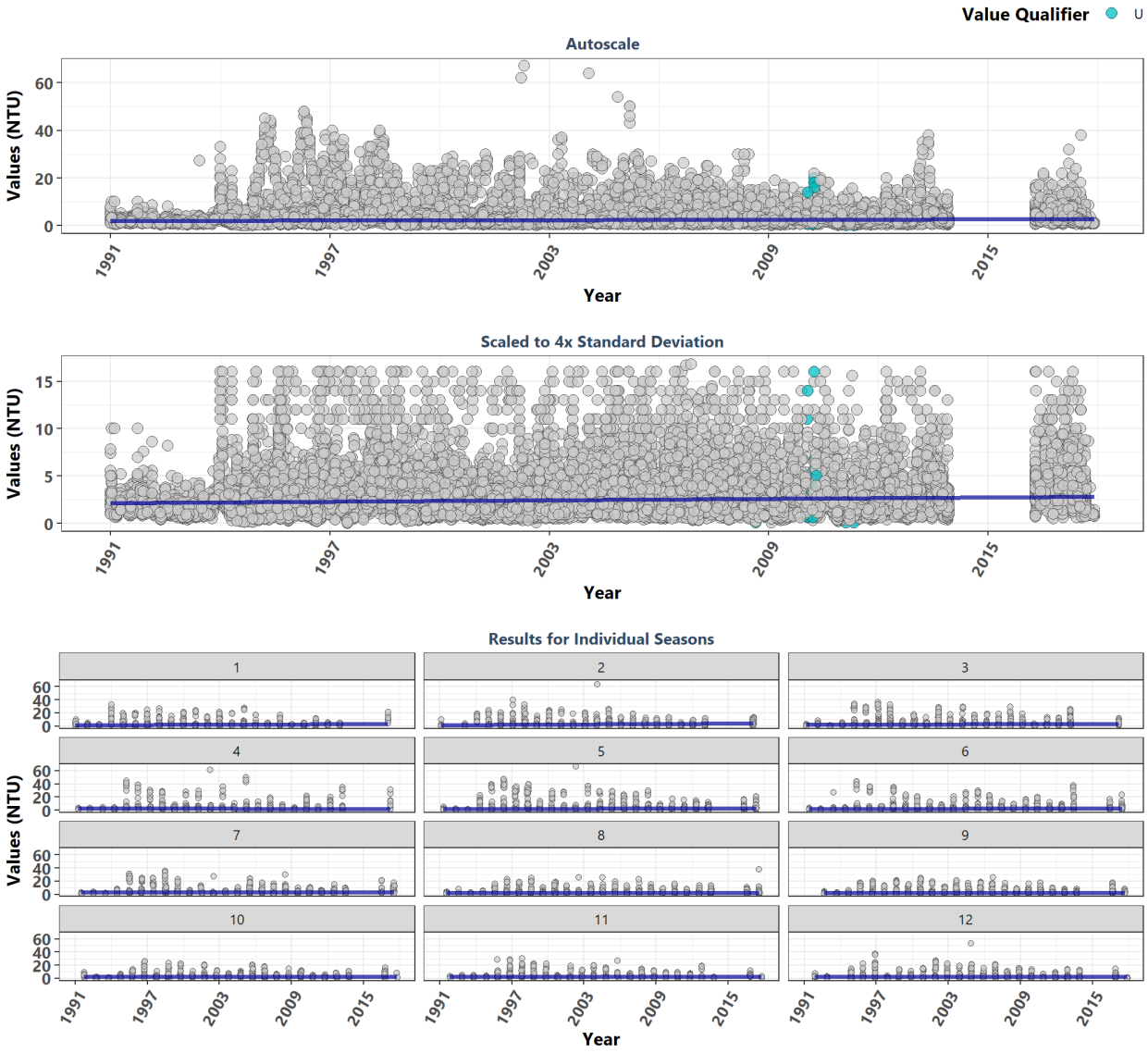
[a] $p < 0.00005$ appear as 0 due to rounding

Pine Island Sound Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4252	1.9	-0.0081	0.0000	1.8875	-0.7	0.4879	67.1	0	0
1	311	1.8	-0.2134	-0.0967	2.4700	0.3	0.7623	NA	NA	0
2	223	2.0	0.1582	0.0350	1.7900	3.5	0.0004	NA	NA	0
3	442	2.0	0.0062	0.0000	1.9000	-3.2	0.0012	NA	NA	0
4	404	1.9	-0.1022	-0.0375	2.3000	-0.8	0.4424	NA	NA	0
5	329	2.0	0.0858	0.0364	1.7818	1.6	0.1140	NA	NA	0
6	517	2.0	0.0115	0.0000	1.8000	2.9	0.0033	NA	NA	0
7	418	2.2	-0.0328	-0.0056	1.3444	-2.0	0.0418	NA	NA	0
8	437	1.9	0.0581	0.0250	1.8750	0.2	0.8453	NA	NA	0
9	366	1.8	-0.0664	-0.0200	2.3600	0.2	0.8705	NA	NA	0
10	290	1.3	0.0057	0.0000	1.8000	-0.8	0.4016	NA	NA	0
11	247	2.0	0.0337	0.0113	1.9206	0.8	0.4277	NA	NA	0
12	268	1.6	-0.0255	-0.0074	1.9446	-5.2	0.0000	NA	NA	0

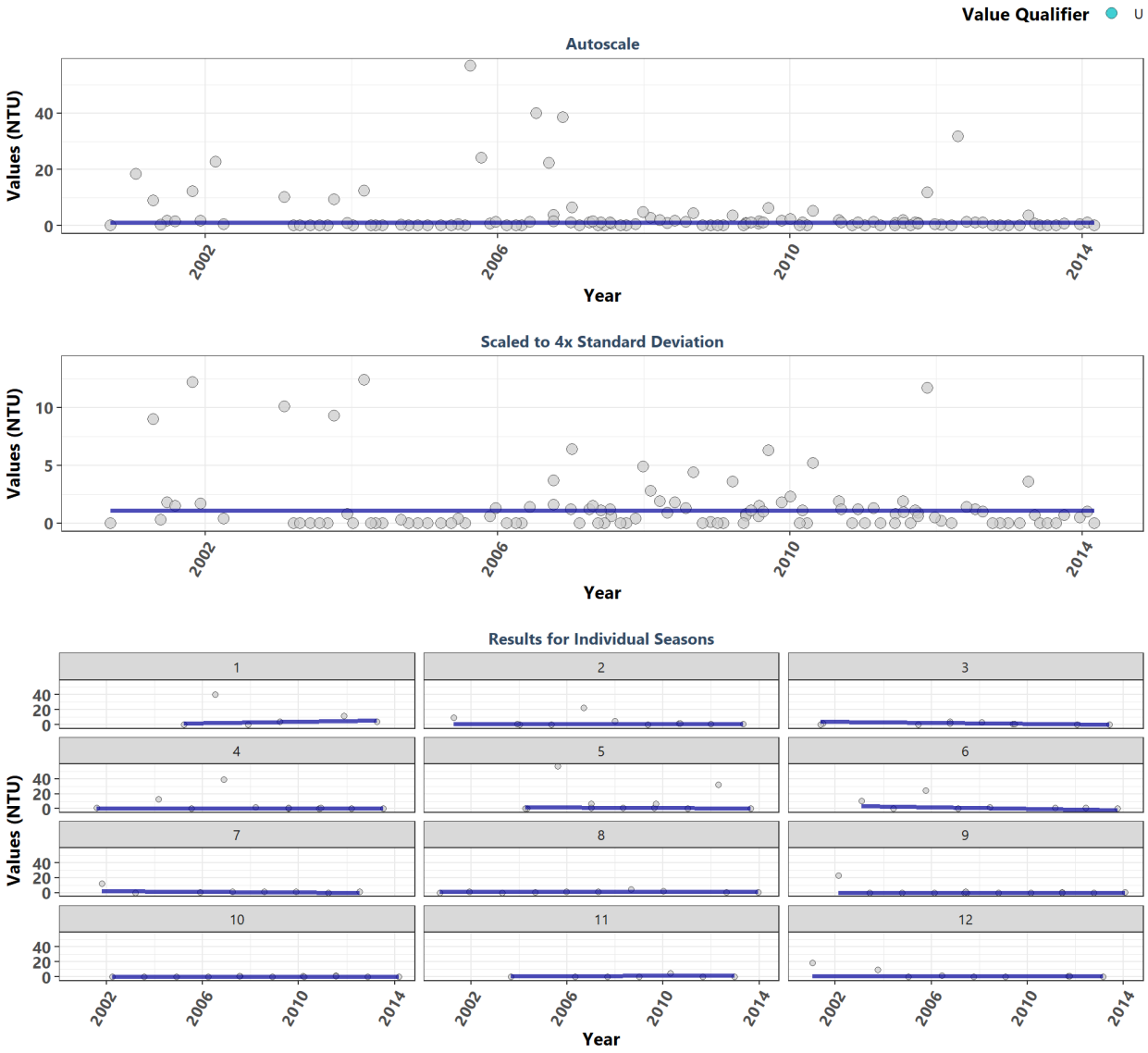
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	15980	2.4	0.0626	0.0260	2.1333	12.7	0.0000	191.4	0	1
1	1153	2.3	0.1165	0.0429	1.8429	0.0	0.9753	NA	NA	1
2	1159	2.3	0.1738	0.0727	1.9273	1.5	0.1243	NA	NA	1
3	1250	2.6	0.1431	0.0545	2.0455	2.5	0.0129	NA	NA	1
4	1116	2.2	-0.0516	-0.0200	2.3600	2.8	0.0044	NA	NA	-1
5	1716	2.4	0.1125	0.0453	1.9107	7.2	0.0000	NA	NA	1
6	1448	2.8	0.0800	0.0273	2.1000	9.9	0.0000	NA	NA	1
7	1152	3.0	0.0389	0.0167	2.8000	2.0	0.0479	NA	NA	1
8	1284	2.7	-0.0631	-0.0200	2.4400	7.7	0.0000	NA	NA	-1
9	1778	2.5	0.0568	0.0222	1.9556	7.1	0.0000	NA	NA	1
10	1352	2.4	-0.0006	0.0000	2.3000	4.4	0.0000	NA	NA	-1
11	1003	2.2	0.0469	0.0200	2.3600	-3.0	0.0027	NA	NA	1
12	1569	2.1	0.0301	0.0111	2.1667	-3.1	0.0022	NA	NA	1

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

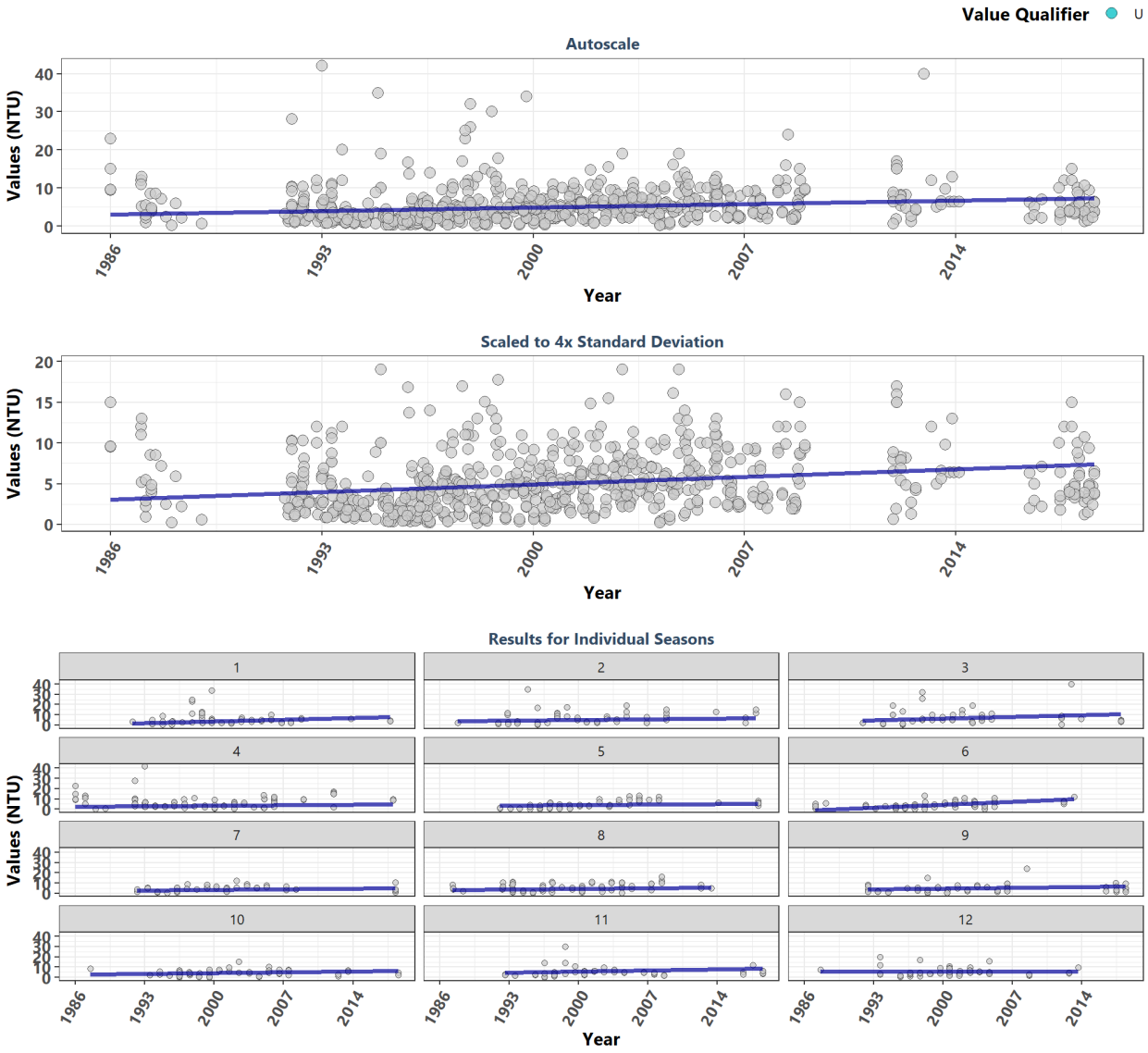
Rocky Bayou State Park Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	119	0.70	-0.0763	0.0000	1.1000	-1.2	0.2264	10.5	0.4852	0
1	6	3.60	0.2667	0.6000	-4.5000	0.6	0.5661	NA	NA	0
2	11	0.80	-0.0182	0.0000	1.0000	-0.4	0.6921	NA	NA	0
3	11	0.80	-0.4727	-0.4000	6.8000	-1.3	0.1781	NA	NA	0
4	11	1.20	0.0952	0.0000	0.0000	-2.0	0.0437	NA	NA	0
5	11	1.00	-0.1786	-0.2625	4.6312	0.0	1.0000	NA	NA	0
6	8	1.35	-0.3556	-0.6500	8.8750	-0.5	0.6180	NA	NA	0
7	8	1.20	-0.3273	-0.2000	3.4000	0.0	1.0000	NA	NA	0
8	10	1.15	0.0000	0.0000	1.2000	1.0	0.3232	NA	NA	0
9	13	0.00	0.0128	0.0000	0.0000	0.0	1.0000	NA	NA	0
10	13	0.10	0.0769	0.0000	0.1000	0.3	0.7458	NA	NA	0
11	7	0.00	0.2667	0.1111	-0.1278	0.2	0.8026	NA	NA	0
12	10	0.75	-0.1091	-0.0500	1.4500	-1.4	0.1584	NA	NA	0

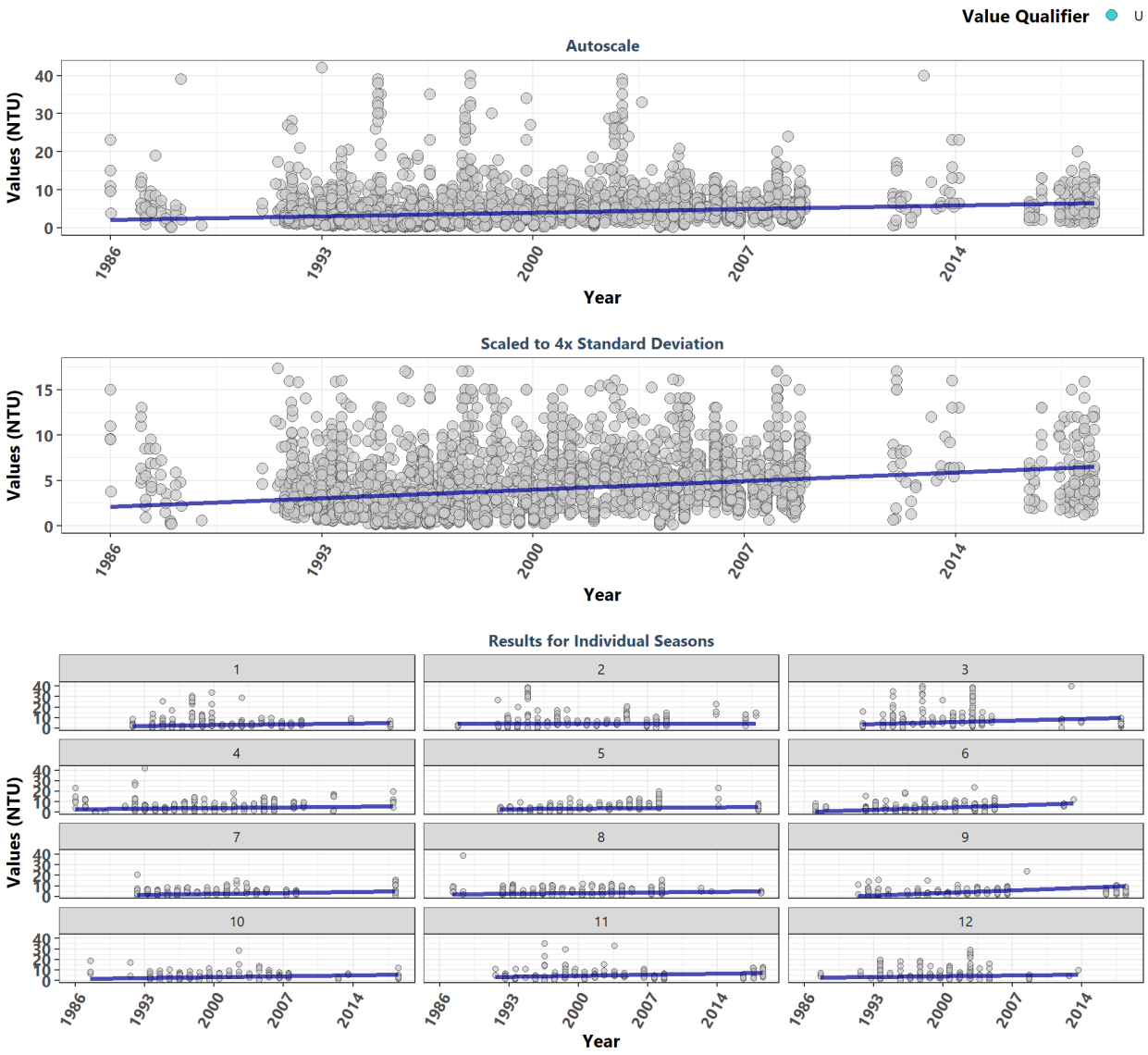
[a] $p < 0.00005$ appear as 0 due to rounding

Rookery Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	711	4.40	0.1802	0.1333	3.0512	6.8	0.0000	23.3	0.0161	1
1	58	3.65	0.2994	0.2408	0.4692	1.2	0.2179	NA	NA	1
2	49	6.76	0.0876	0.0858	3.9267	2.5	0.0134	NA	NA	1
3	49	5.30	0.2440	0.2375	3.1925	0.9	0.3773	NA	NA	1
4	81	5.60	0.1113	0.0700	2.6675	0.2	0.8572	NA	NA	1
5	62	3.93	0.0766	0.0550	3.4300	5.7	0.0000	NA	NA	1
6	67	3.60	0.4913	0.4157	-1.4765	3.6	0.0003	NA	NA	2
7	42	4.83	0.1797	0.0826	2.1309	1.4	0.1607	NA	NA	1
8	73	5.30	0.1428	0.0960	2.8445	2.1	0.0389	NA	NA	1
9	60	3.37	0.1510	0.1050	3.2575	2.0	0.0424	NA	NA	1
10	58	4.09	0.2156	0.1100	2.9100	1.6	0.1133	NA	NA	1
11	56	4.45	0.1648	0.1500	3.6500	2.3	0.0189	NA	NA	1
12	56	4.20	0.0139	0.0118	5.4588	0.8	0.4062	NA	NA	1

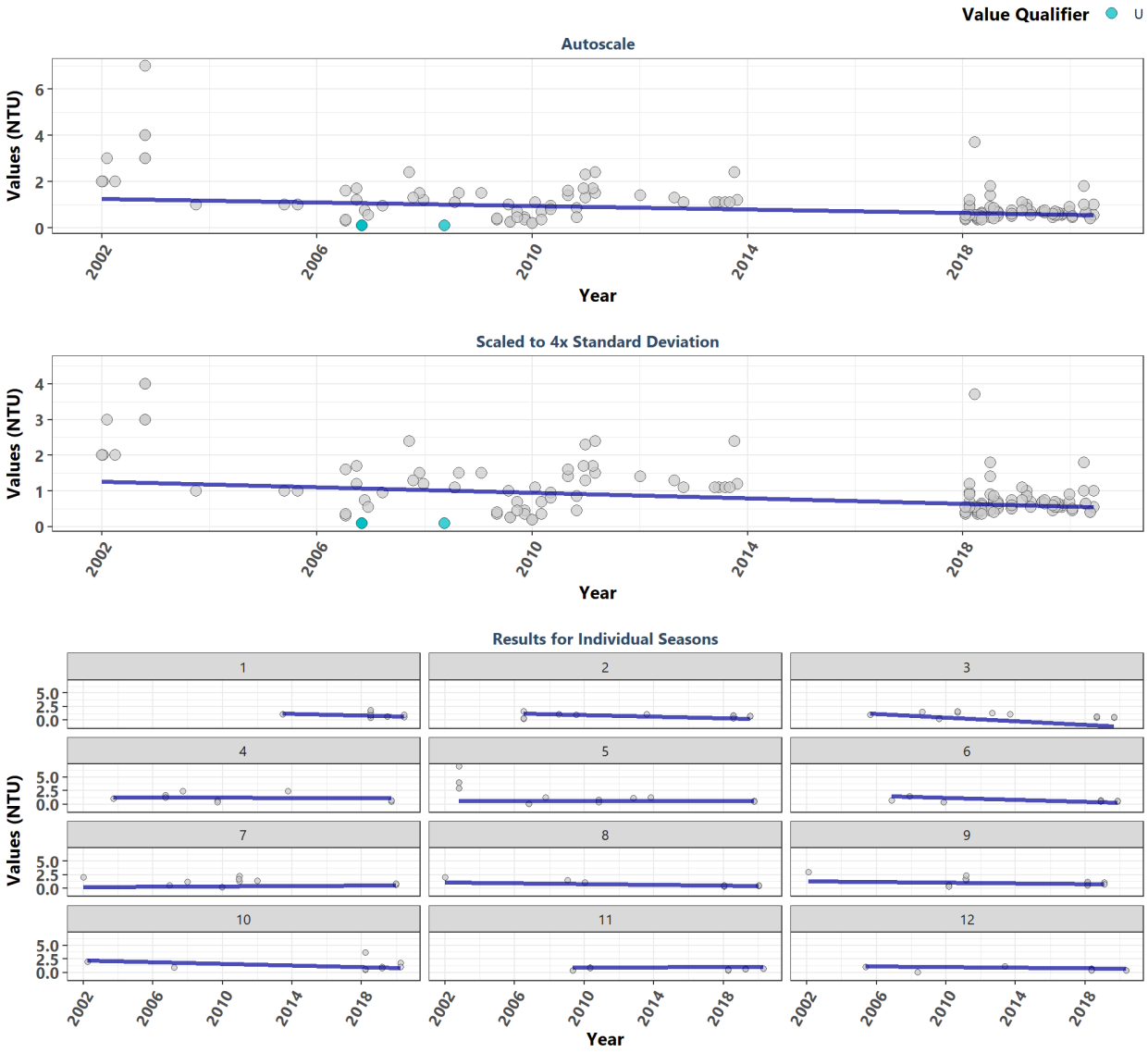
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3982	3.40	0.1908	0.1357	2.1050	17.3	0.0000	123.5	0	1
1	349	3.10	0.0942	0.0900	1.8100	3.7	0.0002	NA	NA	1
2	288	3.62	0.0116	0.0100	4.0900	9.2	0.0000	NA	NA	1
3	223	5.00	0.2291	0.2417	2.1000	5.1	0.0000	NA	NA	1
4	404	4.00	0.0925	0.0900	2.9200	2.8	0.0053	NA	NA	1
5	384	2.60	0.1209	0.0900	2.1100	12.8	0.0000	NA	NA	1
6	392	2.80	0.3616	0.3000	0.3150	2.8	0.0051	NA	NA	2
7	255	4.00	0.2577	0.1241	0.9653	5.5	0.0000	NA	NA	1
8	387	3.70	0.1333	0.0738	2.1400	3.8	0.0002	NA	NA	1
9	295	2.46	0.4365	0.3423	-1.5077	6.6	0.0000	NA	NA	2
10	335	3.10	0.2681	0.1333	1.4000	3.3	0.0009	NA	NA	1
11	326	3.00	0.2288	0.1385	2.4769	7.3	0.0000	NA	NA	1
12	344	4.23	0.1284	0.0917	2.7833	0.3	0.7477	NA	NA	1

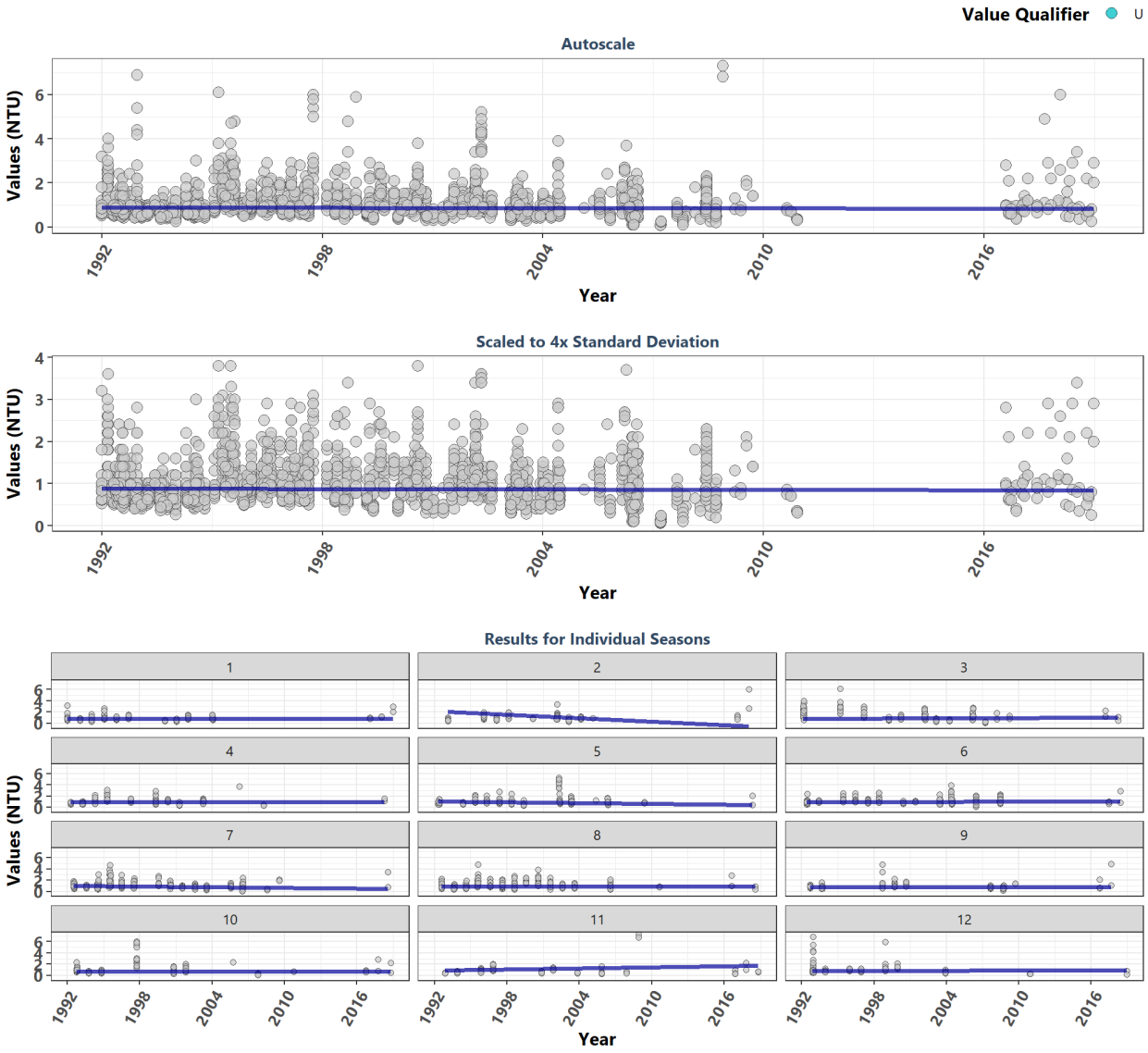
[a] $p < 0.00005$ appear as 0 due to rounding

St. Andrews State Park Aquatic Preserve



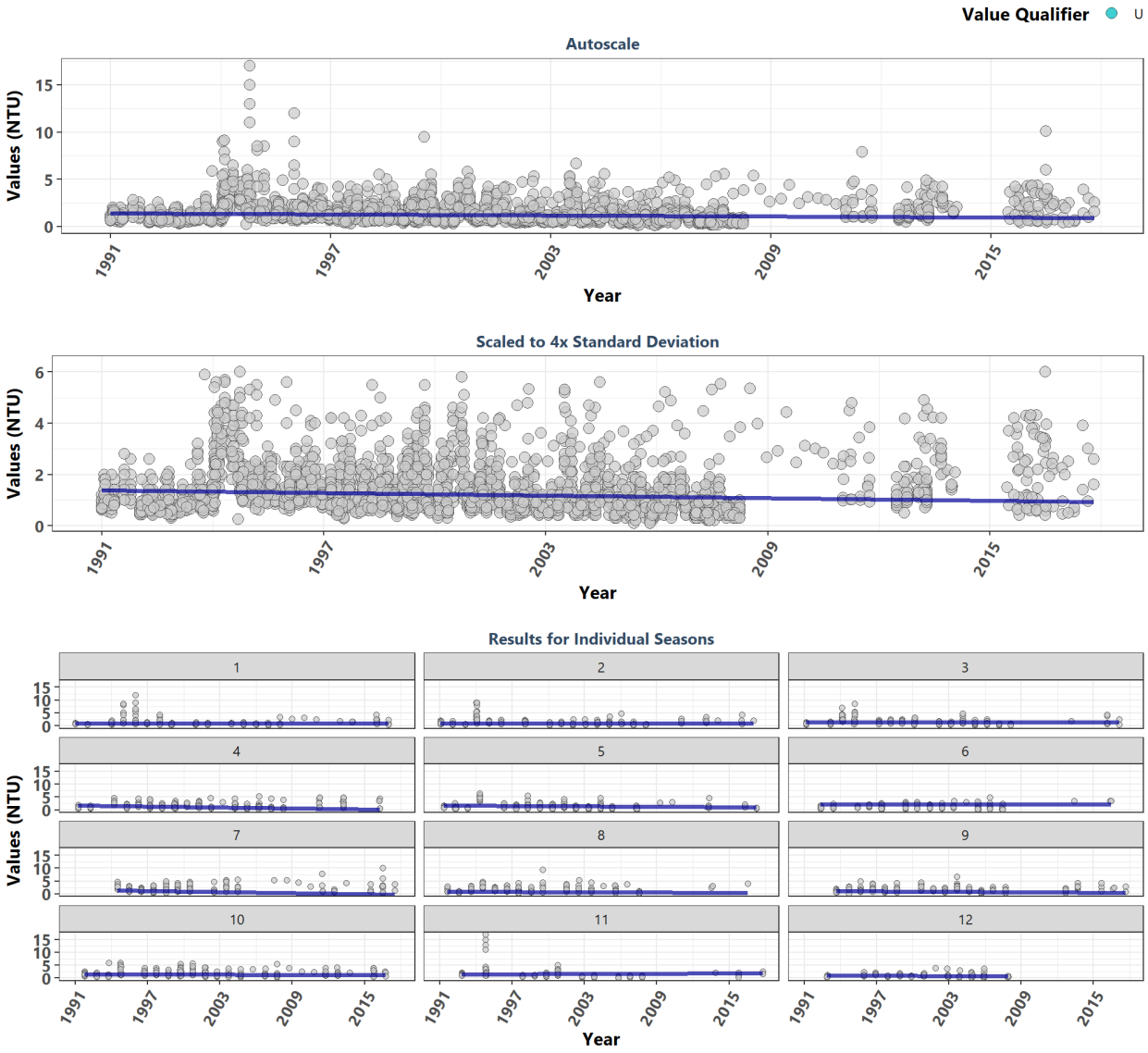
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	141	0.70	-0.1358	-0.0385	1.2586	-2.2	0.0246	8.4	0.6744	-1
1	10	0.80	-0.1556	-0.0667	1.9333	-0.6	0.5593	NA	NA	-1
2	13	0.75	-0.3048	-0.0809	1.6209	-0.6	0.5658	NA	NA	-2
3	15	0.65	-0.2667	-0.1636	1.7932	-1.6	0.1150	NA	NA	-2
4	9	1.00	0.0000	-0.0067	1.2606	-0.6	0.5204	NA	NA	-1
5	16	0.98	-0.0364	0.0000	0.6000	-1.5	0.1468	NA	NA	-1
6	11	0.60	-0.4000	-0.0875	1.8750	-0.1	0.9331	NA	NA	-2
7	11	1.20	0.2436	0.0211	0.2122	0.0	1.0000	NA	NA	1
8	10	0.48	-0.2667	-0.0400	1.1150	-1.7	0.0962	NA	NA	-1
9	13	0.95	-0.1282	-0.0298	1.2566	-0.8	0.4067	NA	NA	-1
10	10	0.98	-0.1795	-0.0786	2.2071	0.3	0.7775	NA	NA	-1
11	13	0.55	0.0889	0.0042	0.9083	1.2	0.2491	NA	NA	1
12	10	0.48	-0.1944	-0.0281	1.2250	-1.1	0.2866	NA	NA	-1

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



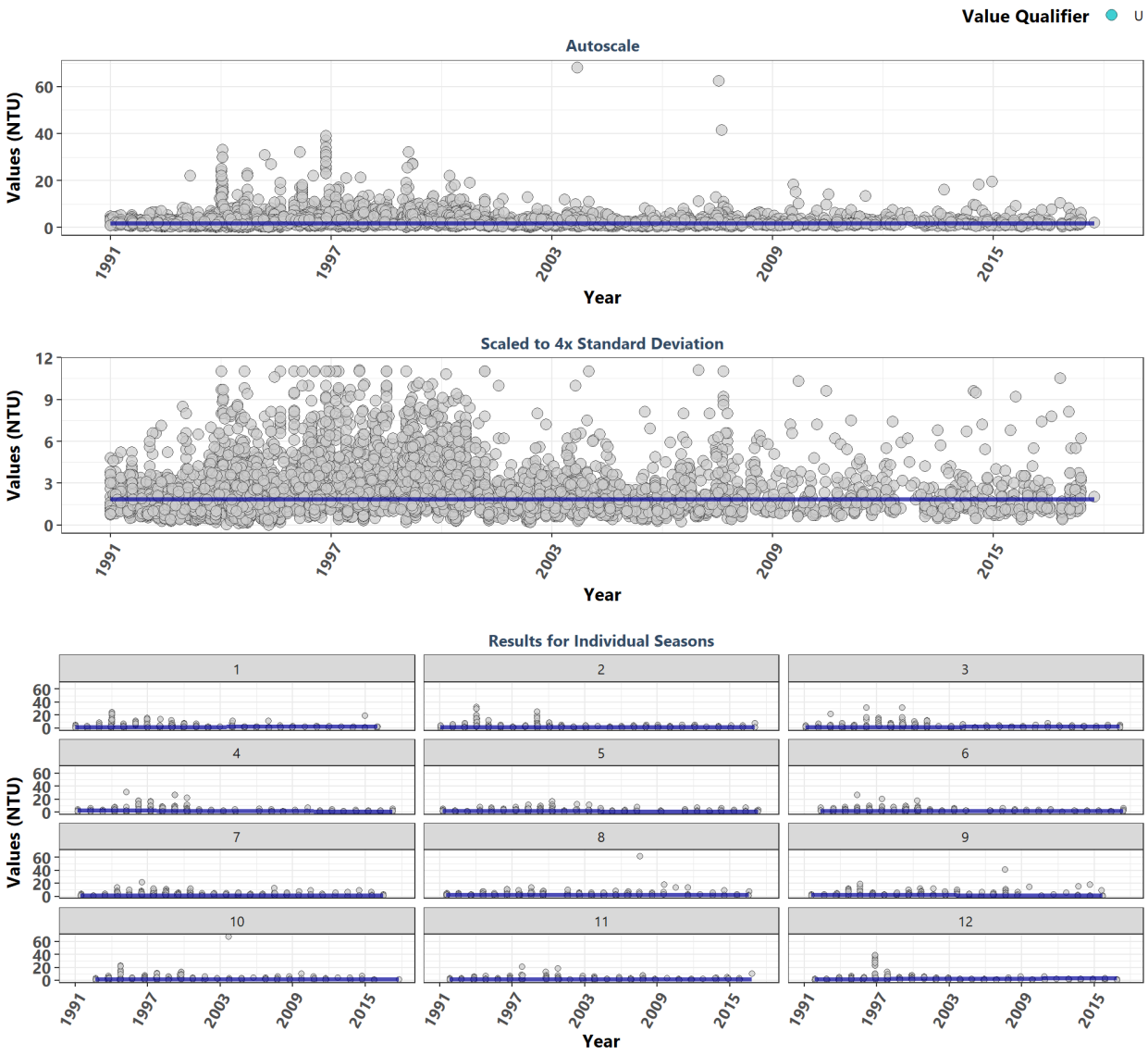
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2537	0.90	-0.0307	-0.0017	0.8833	-2.3	0.0230	132.4	0	-1
1	160	0.80	0.0191	0.0000	0.8200	2.0	0.0416	NA	NA	-1
2	146	0.80	-0.3868	-0.1000	2.1000	-0.8	0.4118	NA	NA	-2
3	233	1.10	0.1079	0.0111	0.7444	-8.9	0.0000	NA	NA	1
4	173	0.94	0.0266	0.0000	0.9200	0.7	0.4539	NA	NA	-1
5	228	0.92	-0.1500	-0.0200	1.0000	0.6	0.5466	NA	NA	-1
6	359	0.90	0.0379	0.0050	0.9050	-1.1	0.2864	NA	NA	1
7	368	0.90	-0.1366	-0.0200	0.9500	-4.3	0.0000	NA	NA	-1
8	343	1.00	-0.0373	0.0000	0.9000	5.2	0.0000	NA	NA	-1
9	121	0.82	-0.0453	-0.0017	0.8167	0.3	0.7541	NA	NA	-1
10	122	0.85	0.0164	0.0000	0.6800	-2.3	0.0240	NA	NA	-1
11	139	0.68	0.1872	0.0333	0.8667	0.3	0.7739	NA	NA	1
12	145	0.80	0.0374	0.0054	0.7838	0.7	0.4915	NA	NA	1

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



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2943	1.10	-0.1074	-0.0167	1.3857	-8.9	0.0000	120.3	0	-1
1	262	0.80	-0.0116	0.0000	1.0000	-2.5	0.0140	NA	NA	-1
2	387	0.86	-0.0372	-0.0020	0.9780	-8.5	0.0000	NA	NA	-1
3	225	1.10	0.0074	0.0000	1.4000	-7.2	0.0000	NA	NA	-1
4	297	1.30	-0.3200	-0.0667	1.7000	-1.1	0.2560	NA	NA	-1
5	302	0.96	-0.0928	-0.0273	1.7909	-1.0	0.3337	NA	NA	-1
6	263	1.00	-0.0065	0.0000	2.2000	-0.3	0.7783	NA	NA	-1
7	155	2.20	-0.3793	-0.0643	1.5786	-0.1	0.9061	NA	NA	-1
8	173	1.60	-0.1080	-0.0107	0.8173	-1.8	0.0693	NA	NA	-1
9	202	1.60	-0.2875	-0.0333	1.2600	2.0	0.0498	NA	NA	-1
10	293	1.40	-0.0441	-0.0071	1.3714	0.2	0.8505	NA	NA	-1
11	152	1.00	0.0926	0.0171	1.4286	-7.0	0.0000	NA	NA	1
12	232	0.70	-0.1013	-0.0100	0.9000	-2.5	0.0134	NA	NA	-1

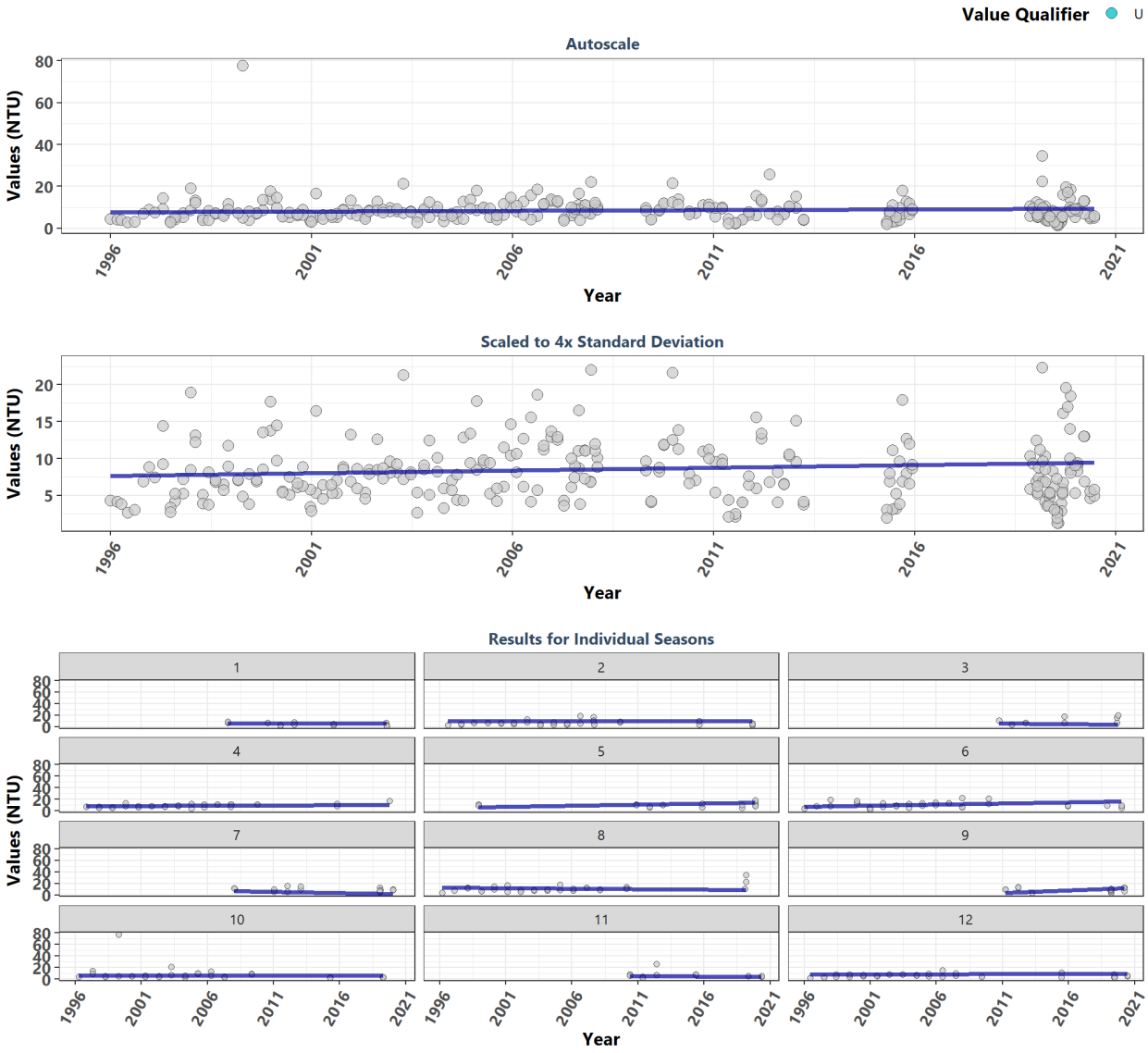
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6905	1.8	0.0051	0.0000	1.8538	0.9	0.3915	116.9	0	0
1	536	1.9	0.0499	0.0137	1.9175	-0.6	0.5187	NA	NA	0
2	494	2.0	0.0027	0.0000	1.9000	-6.7	0.0000	NA	NA	0
3	605	1.8	0.1047	0.0344	1.3594	-3.1	0.0022	NA	NA	0
4	601	1.9	-0.2012	-0.0571	2.4286	-1.6	0.1017	NA	NA	0
5	630	1.6	-0.0830	-0.0208	1.9667	1.3	0.1787	NA	NA	0
6	588	1.6	0.0527	0.0154	1.8077	3.8	0.0001	NA	NA	0
7	694	1.8	0.0357	0.0086	1.5314	2.2	0.0274	NA	NA	0
8	723	1.9	-0.0186	0.0000	1.9000	2.1	0.0332	NA	NA	0
9	485	2.1	-0.0444	-0.0125	1.9875	5.1	0.0000	NA	NA	0
10	535	1.9	0.0558	0.0154	1.6923	0.1	0.9246	NA	NA	0
11	423	2.0	-0.0630	-0.0167	1.7167	1.5	0.1238	NA	NA	0
12	591	1.6	0.1551	0.0667	1.7000	-2.3	0.0216	NA	NA	0

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

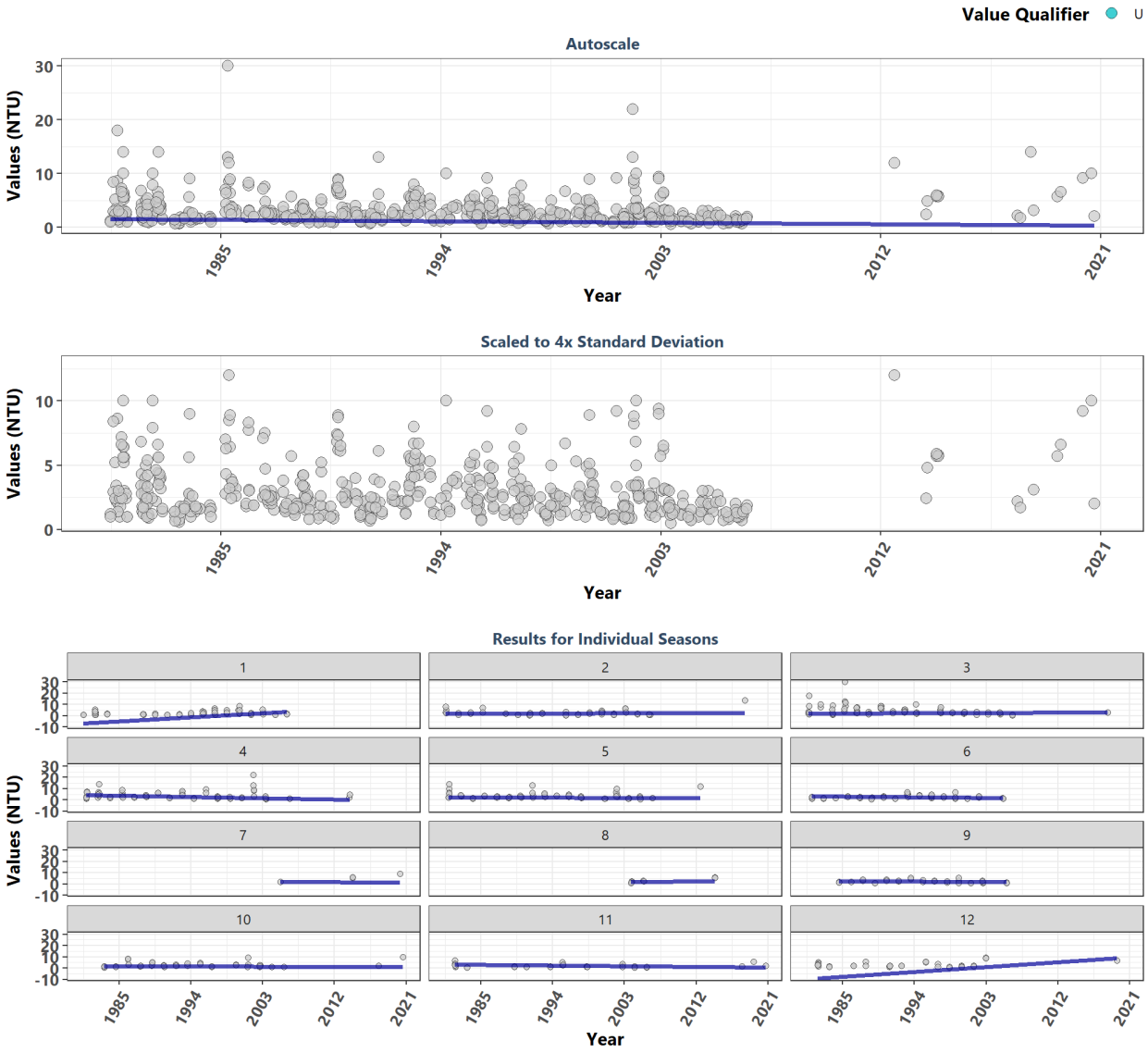
Tomoka Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	275	7.50	0.0645	0.0741	7.6758	1.8	0.0660	29.4	0.0019	0
1	15	3.21	0.0814	0.0406	5.4753	-2.2	0.0266	NA	NA	0
2	34	6.00	0.0294	0.0219	9.5840	0.1	0.9525	NA	NA	0
3	10	6.78	-0.1905	-0.2103	9.4006	1.8	0.0660	NA	NA	0
4	29	8.70	0.0667	0.0769	8.5987	3.2	0.0016	NA	NA	0
5	17	10.02	0.4163	0.3833	5.6333	0.1	0.8997	NA	NA	0
6	31	9.29	0.3670	0.4000	6.9000	0.5	0.6091	NA	NA	0
7	18	8.91	-0.4190	-0.3972	11.1560	-0.9	0.3864	NA	NA	0
8	29	10.10	-0.1503	-0.1609	12.6103	2.8	0.0054	NA	NA	0
9	15	9.31	0.4667	0.9540	-10.8721	0.4	0.7168	NA	NA	0
10	29	5.41	0.0089	0.0056	5.9472	-1.2	0.2436	NA	NA	0
11	15	5.40	-0.1552	-0.0763	5.9439	-1.0	0.3339	NA	NA	0
12	33	5.80	0.0762	0.0375	8.4516	0.7	0.5140	NA	NA	0

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

Yellow River Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	563	2.40	-0.0777	-0.0455	2.3258	-3.8	0.0002	45.5	0	-1
1	62	1.80	0.6429	0.5933	-17.4817	1.5	0.1307	NA	NA	2
2	54	1.90	0.0678	0.0250	1.4250	-0.6	0.5338	NA	NA	1
3	76	3.30	0.1299	0.0478	1.1569	-3.3	0.0011	NA	NA	1
4	72	2.65	-0.2544	-0.1800	7.6200	-2.8	0.0049	NA	NA	-1
5	80	2.65	-0.0587	-0.0233	2.5767	-3.4	0.0006	NA	NA	-1
6	56	2.40	-0.2620	-0.1000	5.2000	1.4	0.1572	NA	NA	-1
7	4	5.80	-0.1472	-0.0500	3.6750	1.4	0.1486	NA	NA	-1
8	8	2.40	0.1311	0.0333	0.8667	2.3	0.0215	NA	NA	1
9	32	2.30	-0.0802	-0.0333	3.0000	-1.2	0.2399	NA	NA	-1
10	50	2.05	-0.1092	-0.0250	2.0750	0.7	0.4908	NA	NA	-1
11	35	1.40	-0.2261	-0.1000	5.2500	-0.9	0.3576	NA	NA	-1
12	34	2.20	0.8333	0.7090	-22.5600	-0.7	0.5095	NA	NA	2

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

Appendix V: Managed Area Summary Box Plots

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

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

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

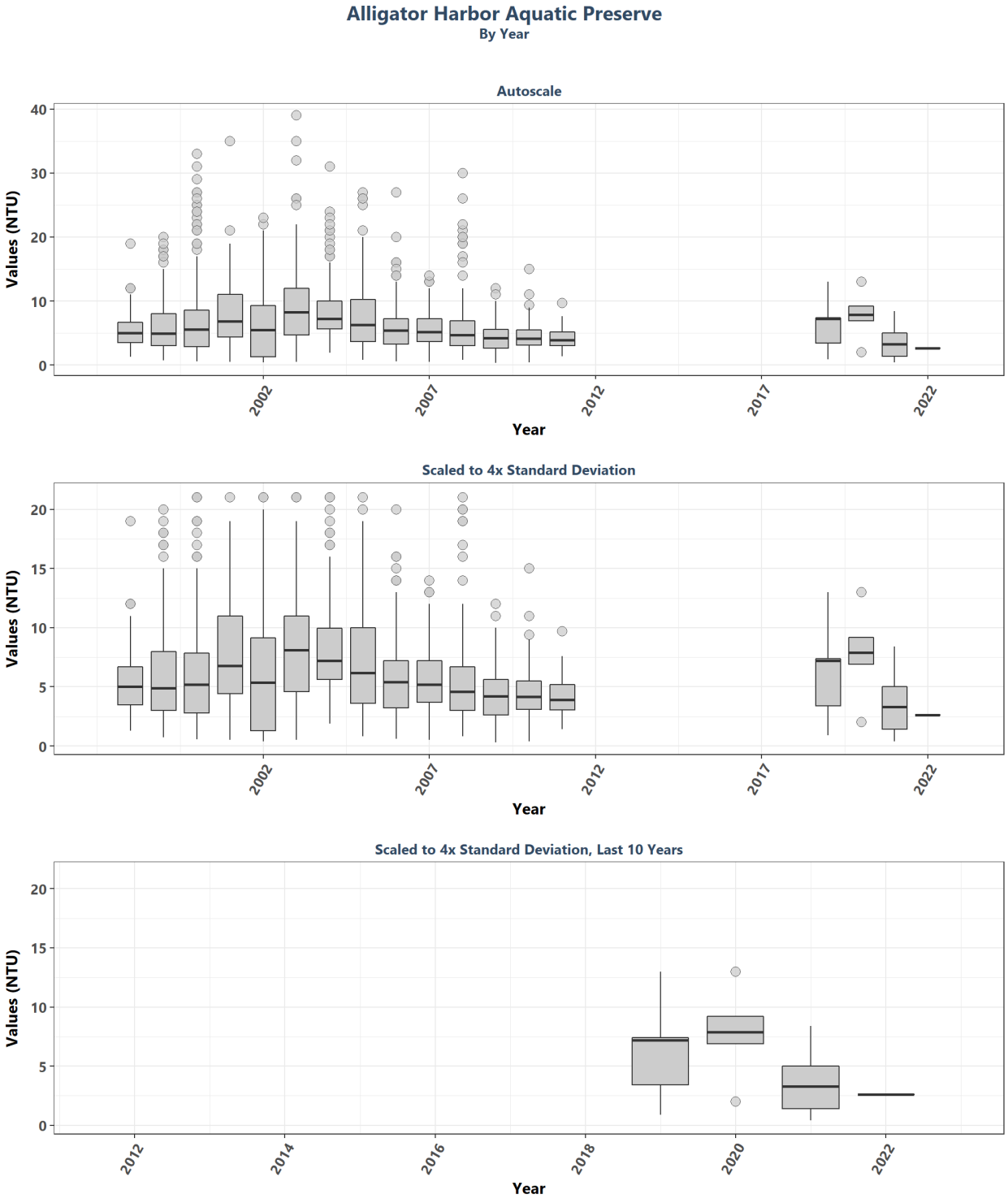
1. Y-axis autoscaled
2. Y-axis set to be mean + 4 times the standard deviation
3. Y-axis set to be mean + 4 times the standard deviation for most recent 10 years of data

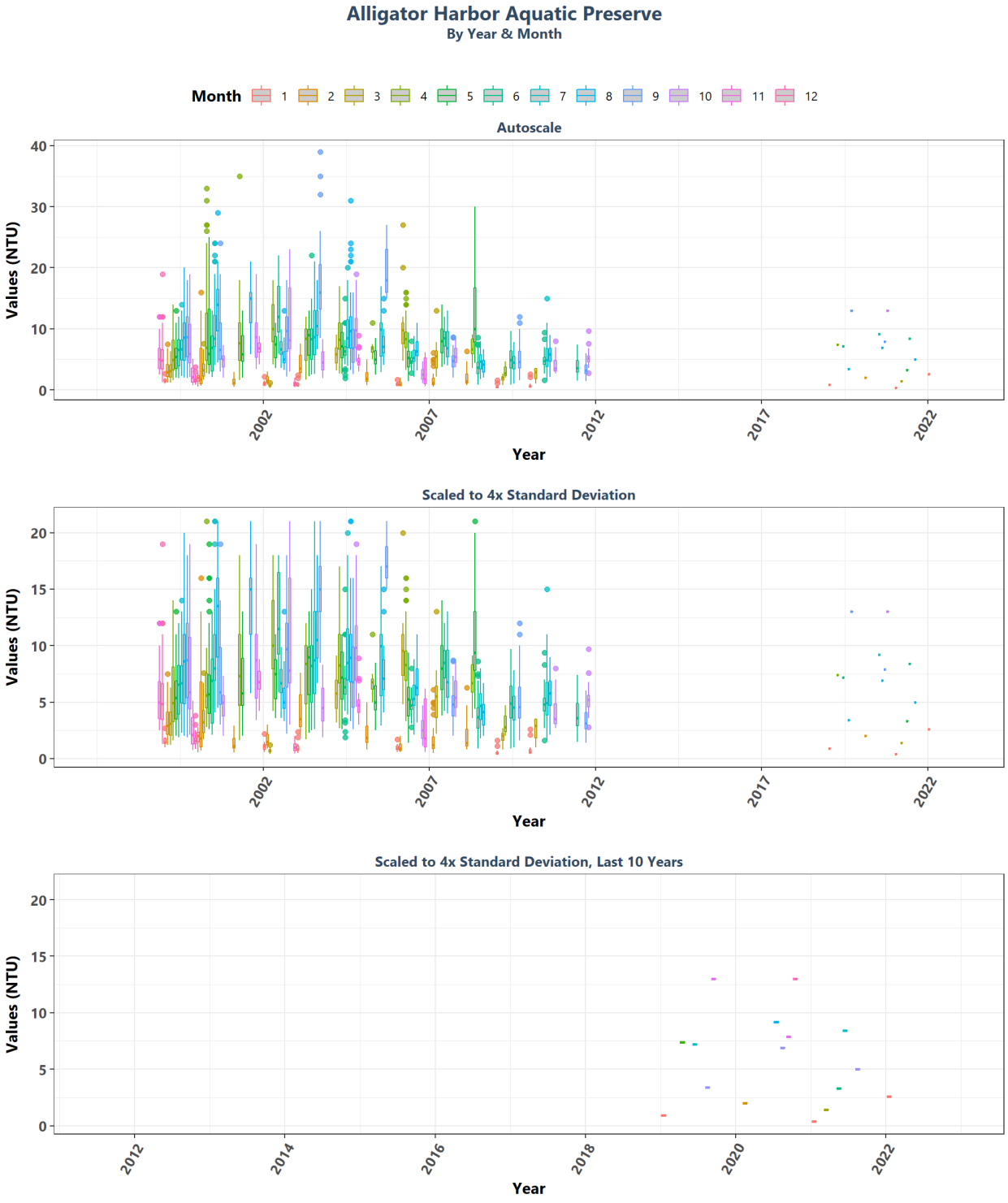
```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$SufficientData==TRUE &
                           data$ManagedAreaName==MA_Include[i],]
      year_lower <- min(plot_data$Year)
      year_upper <- max(plot_data$Year)
      min_RV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

      ##Year plots
      p1 <- ggplot(data=plot_data,
                   aes(x=Year, y=ResultValue, group=Year)) +
         geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                      outlier.size=3, outlier.color="#333333",
                      outlier.fill="#cccccc", outlier.alpha=0.75) +
         labs(subtitle="Autoscale",
              x="Year", y=paste0("Values (", unit, ")")) +
         scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
                            breaks=rev(seq(year_upper,
                                           year_lower, -x_scale))) +
         plot_theme
```

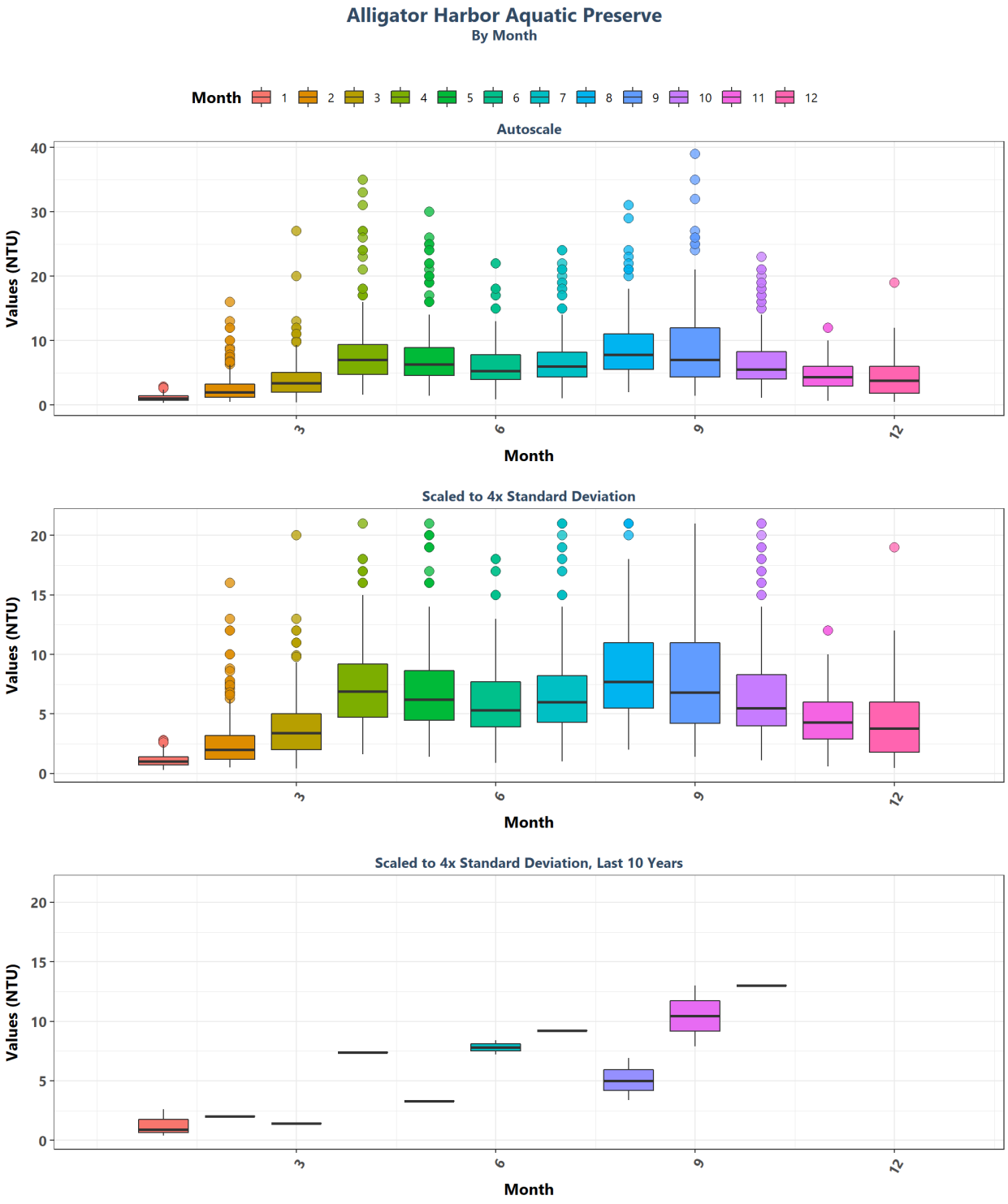
```
p2 <- ggplot(data=plot_data,
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
                      breaks=rev(seq(year_upper,
                                     year_lower, -x_scale))) +
   plot_theme

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

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

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

## Year & Month Plots
p4 <- ggplot(data=plot_data,
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")"), color="Month") +
   scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
                      breaks=rev(seq(year_upper,
                                     year_lower, -x_scale))) +
   plot_theme +
   theme(legend.position="none")

p5 <- ggplot(data=plot_data,
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")"), color="Month") +
```

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

p6 <- ggplot(data=plot_data[plot_data$Year >= year_upper - 10, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
        x="Year", y=paste0("Values (", unit, ")"), color="Month") +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(year_upper - 10.5, year_upper + 1),
                      breaks=rev(seq(year_upper, year_upper - 10,-2))) +
   plot_theme +
   theme(legend.position="none")

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

p00 <- ggplot() + labs(title=paste0(MA_Include[i]),
                       subtitle="By Year & Month") + plot_theme +
   theme(panel.border=element_blank(),
         panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

p8 <- ggplot(data=plot_data,
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
```

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

    p9 <- ggplot(data=plot_data[plot_data$Year >= year_upper - 10, ],
                 aes(x=Month, y=ResultValue,
                     group=Month, fill=as.factor(Month))) +
        geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                     outlier.color="#333333", outlier.alpha=0.75) +
        labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
             x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
        ylim(min_RV, y_scale) +
        scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
        plot_theme +
        theme(legend.position="none")

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

    p000 <- ggplot() + labs(title=paste0(MA_Include[i]),
                            subtitle="By Month") + plot_theme +
        theme(panel.border=element_blank(),
              panel.grid.major=element_blank(),
              panel.grid.minor=element_blank(), axis.line=element_blank())

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

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

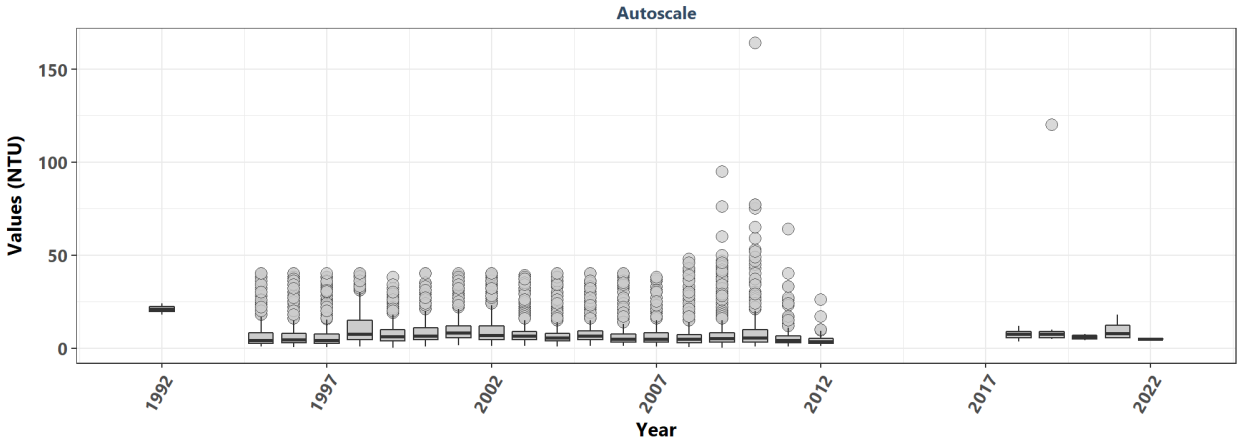

Alligator Harbor Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

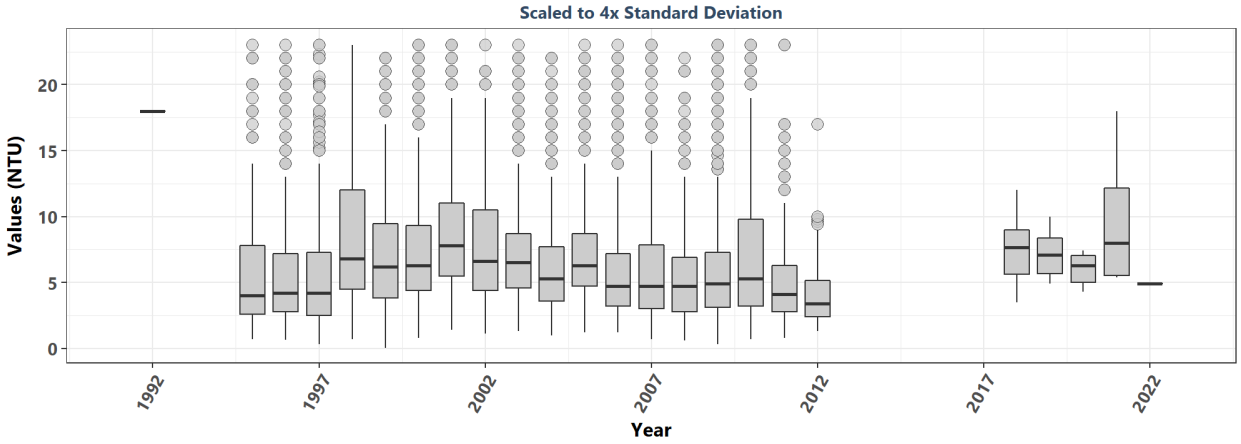

Alligator Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

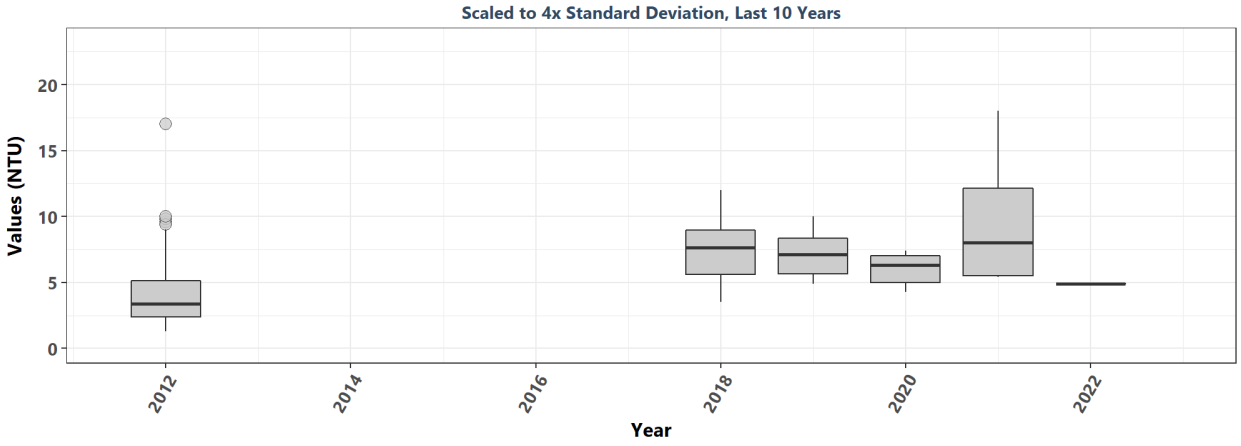

Alligator Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

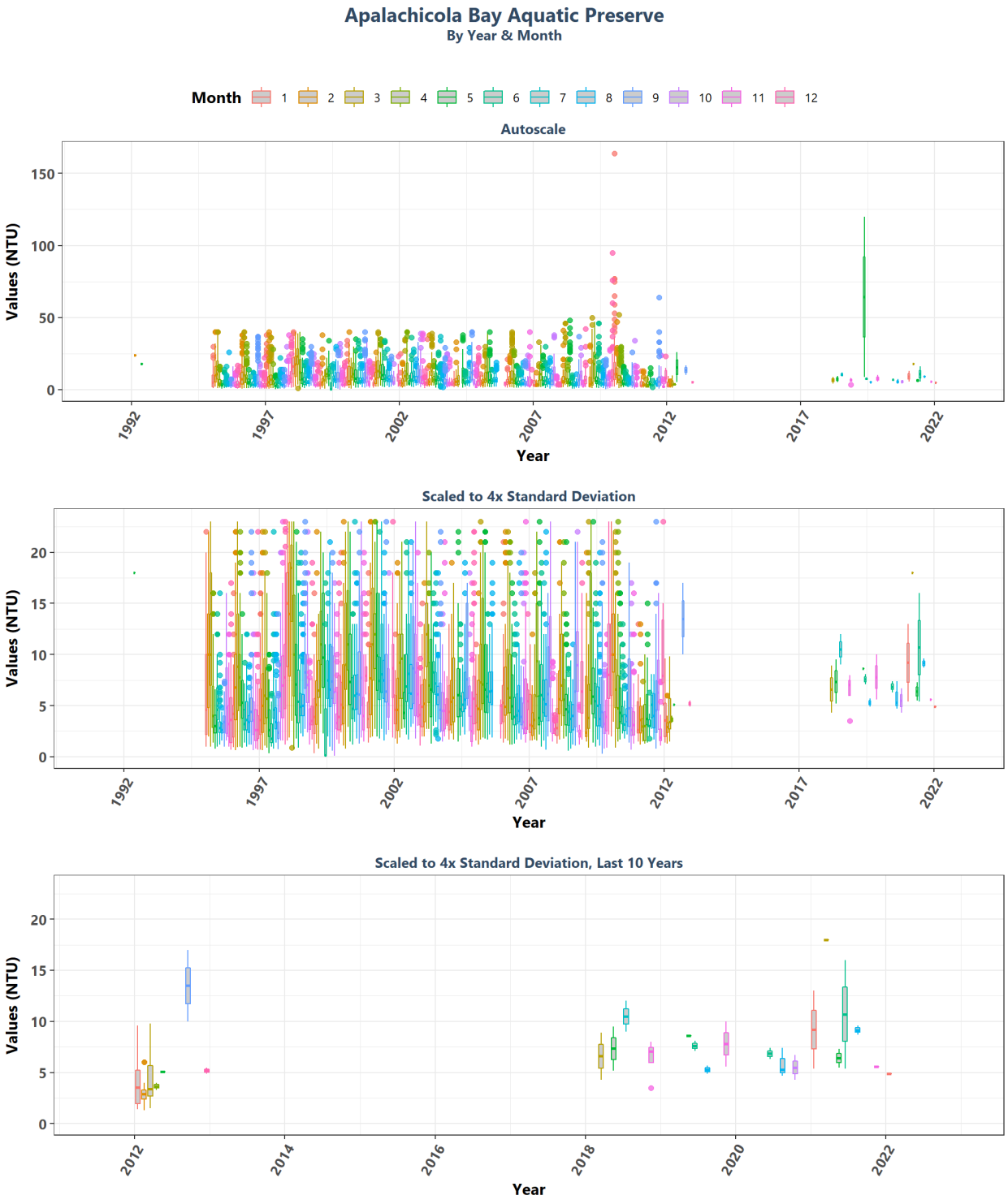
Apalachicola Bay Aquatic Preserve

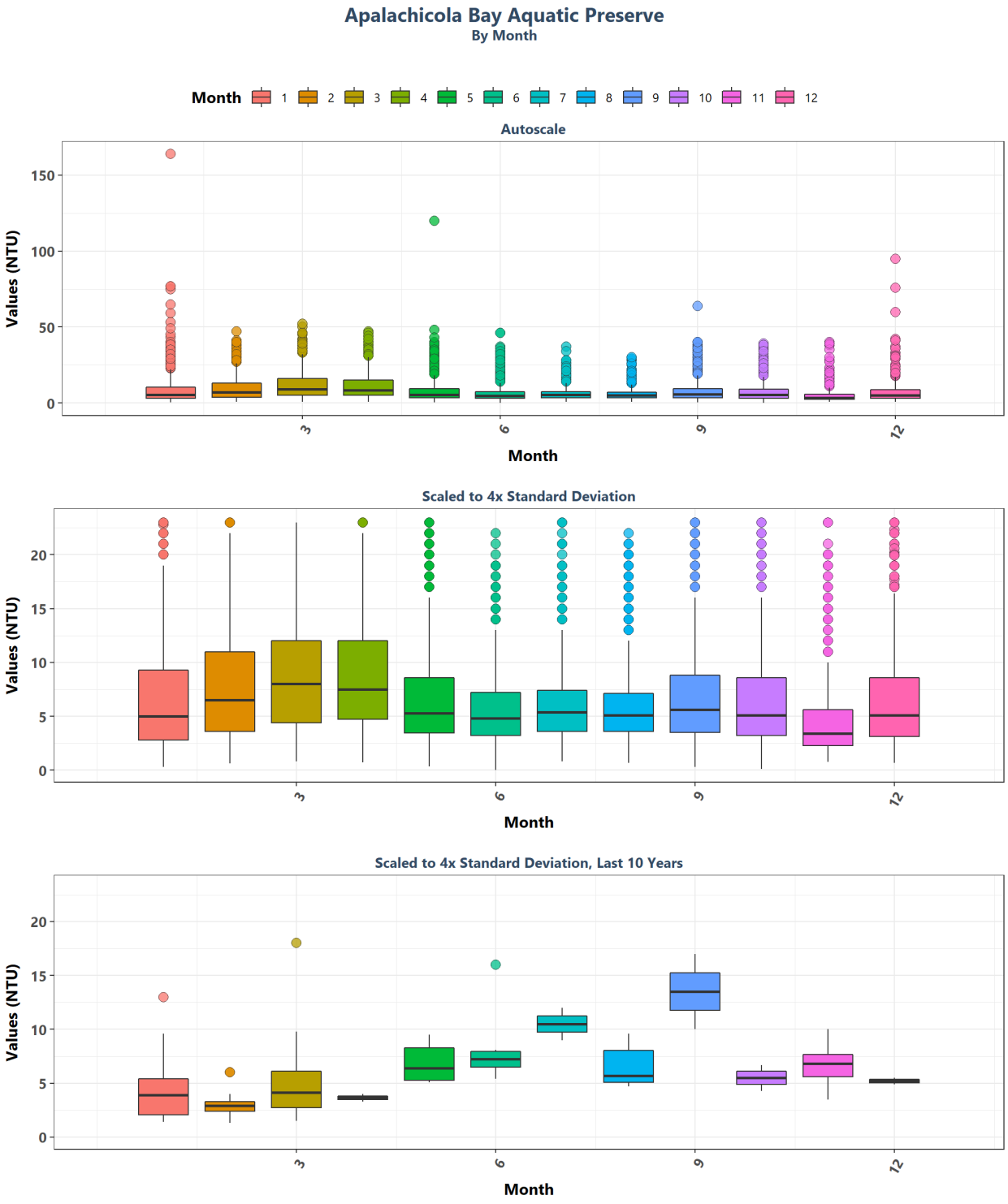
By Year





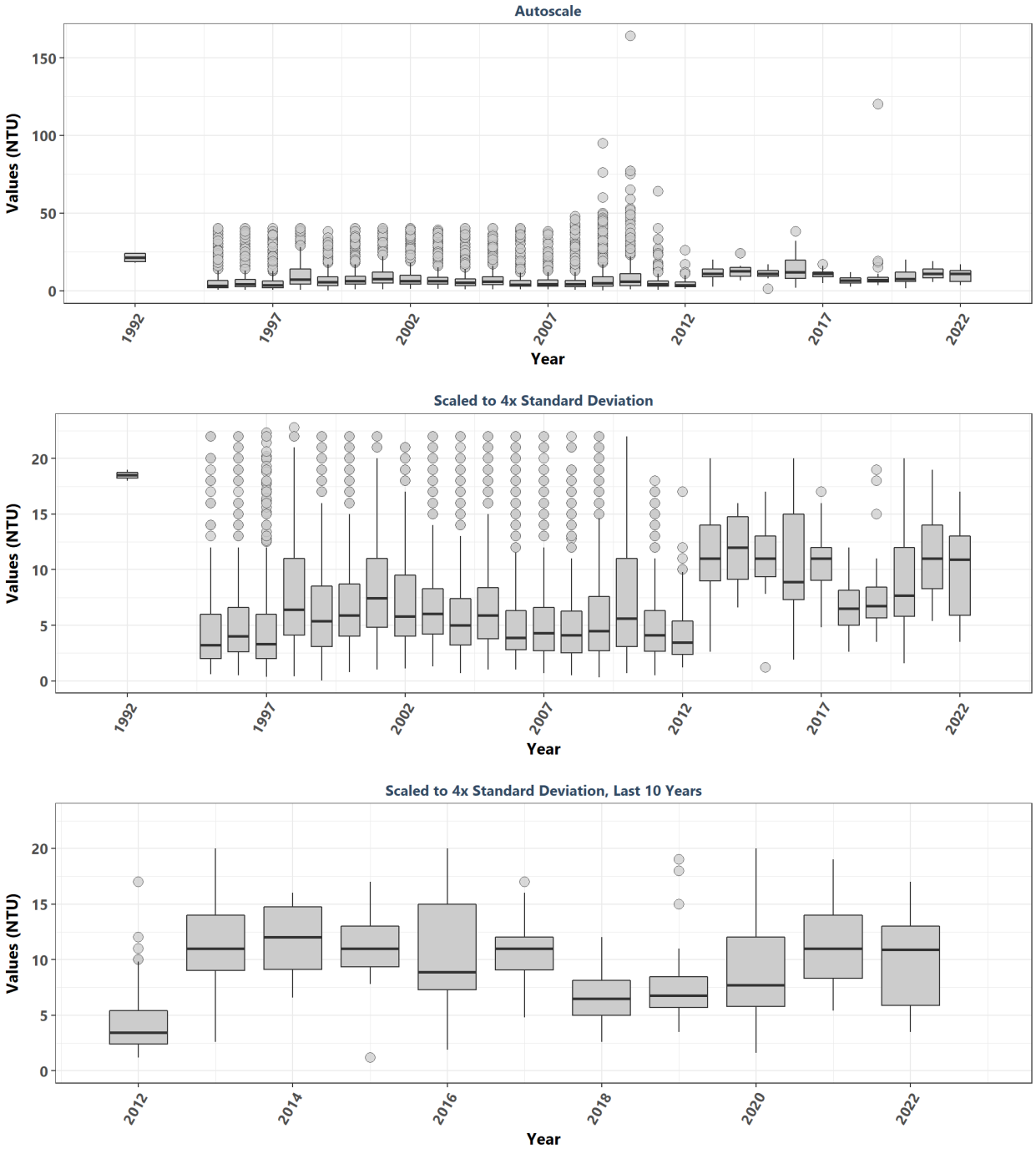


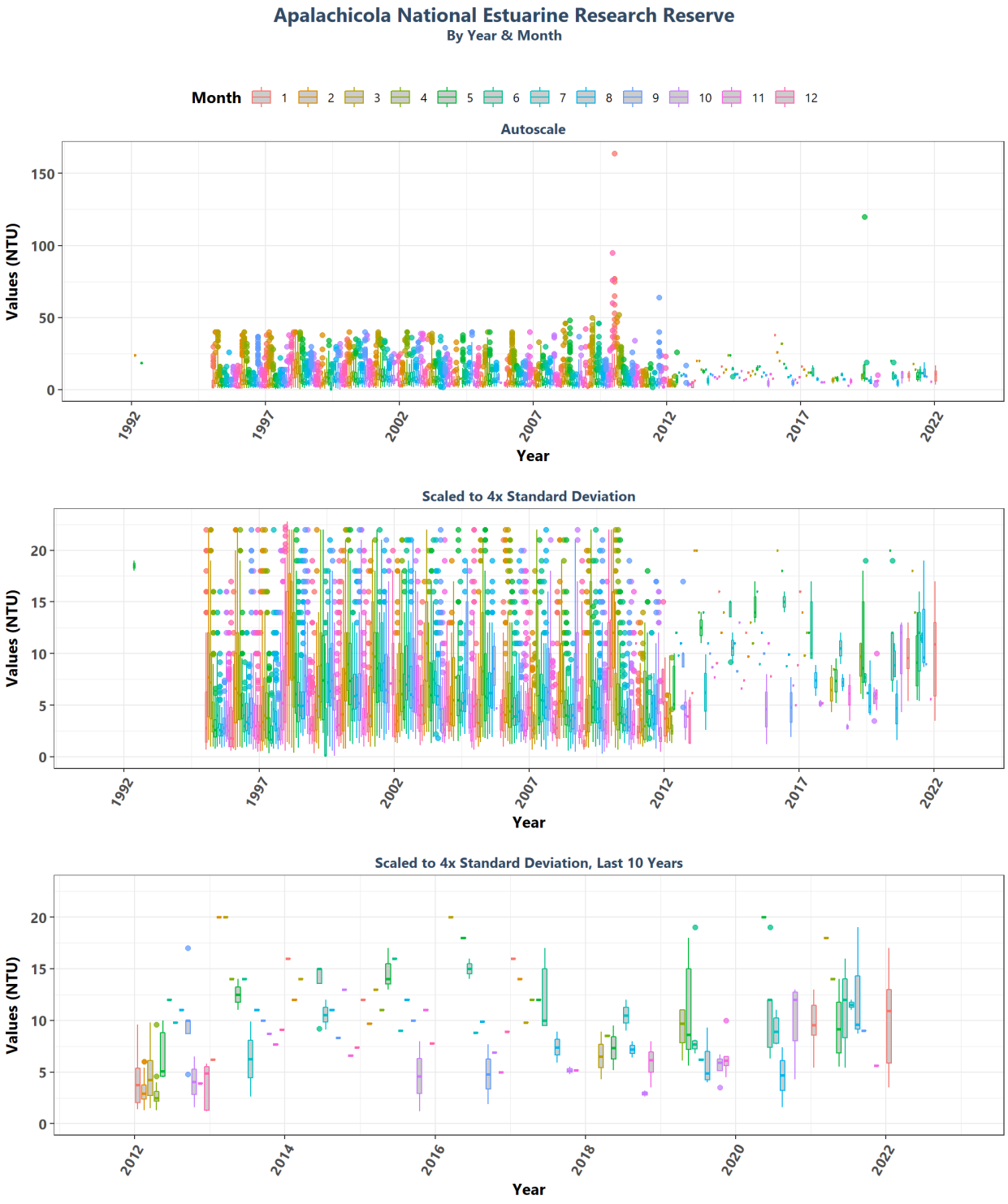

Apalachicola Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

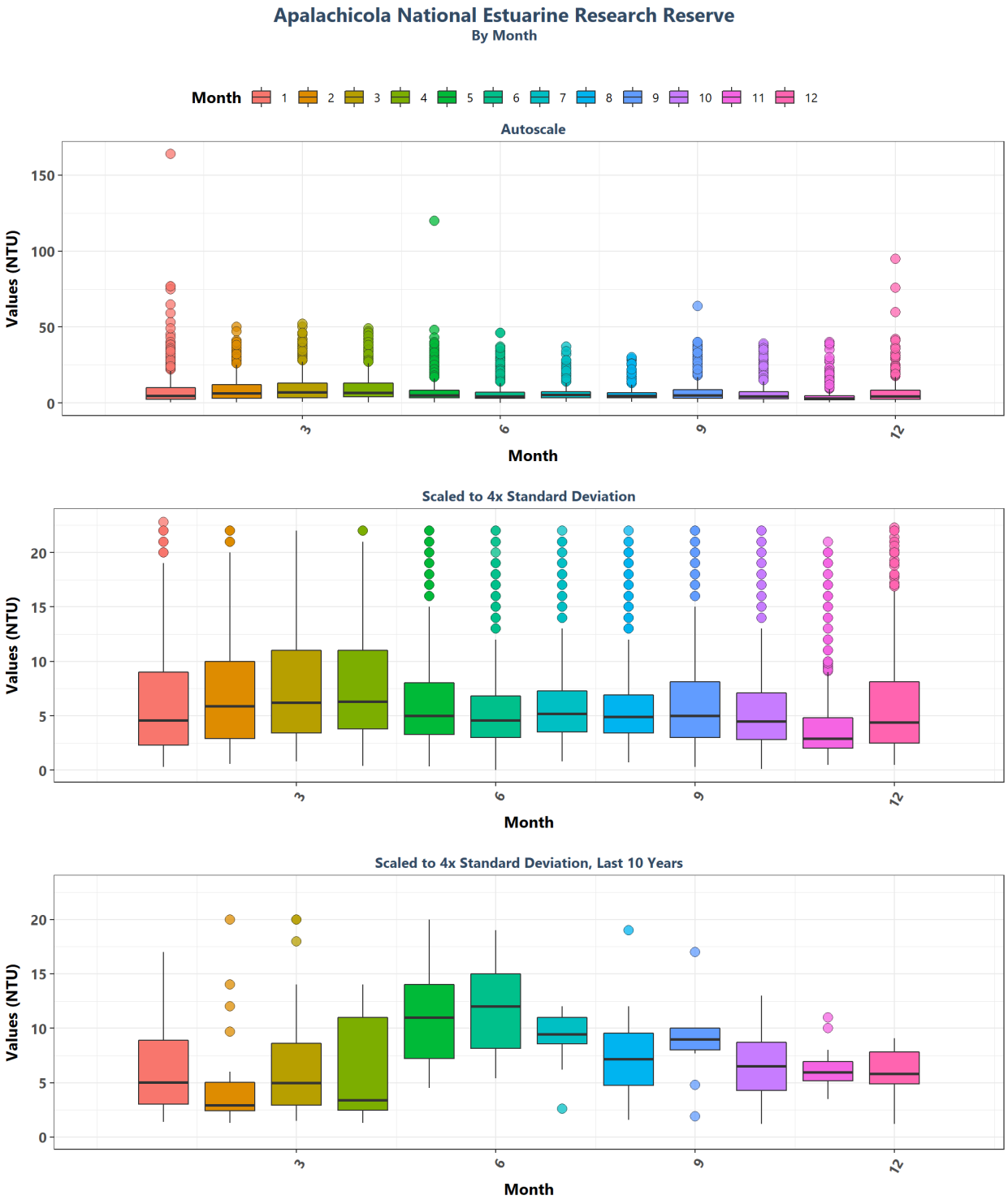

Apalachicola Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
150
100
50
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
20
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
20
15
10
5
0
3
6
9
12
Month

Apalachicola National Estuarine Research Reserve

By Year

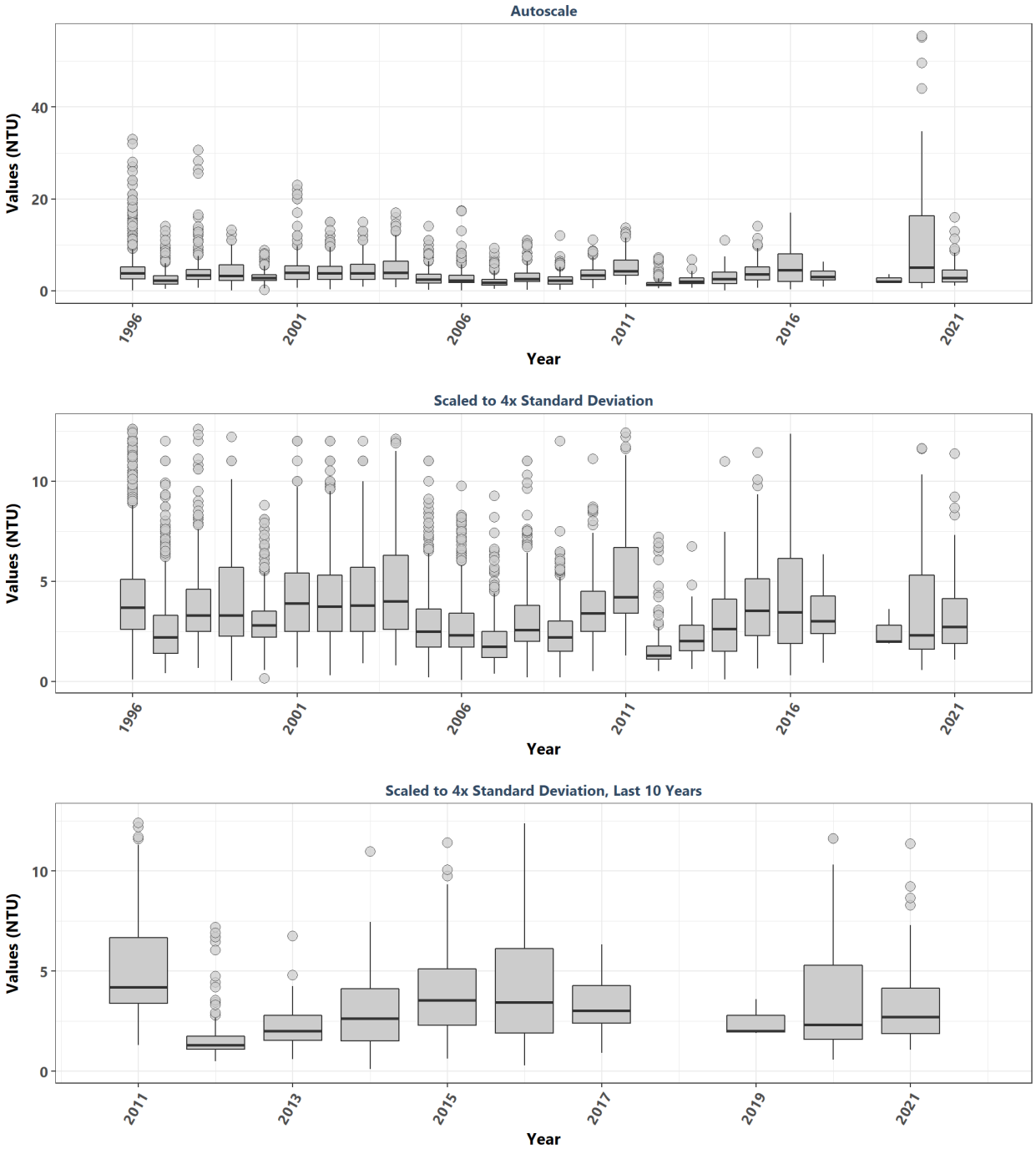


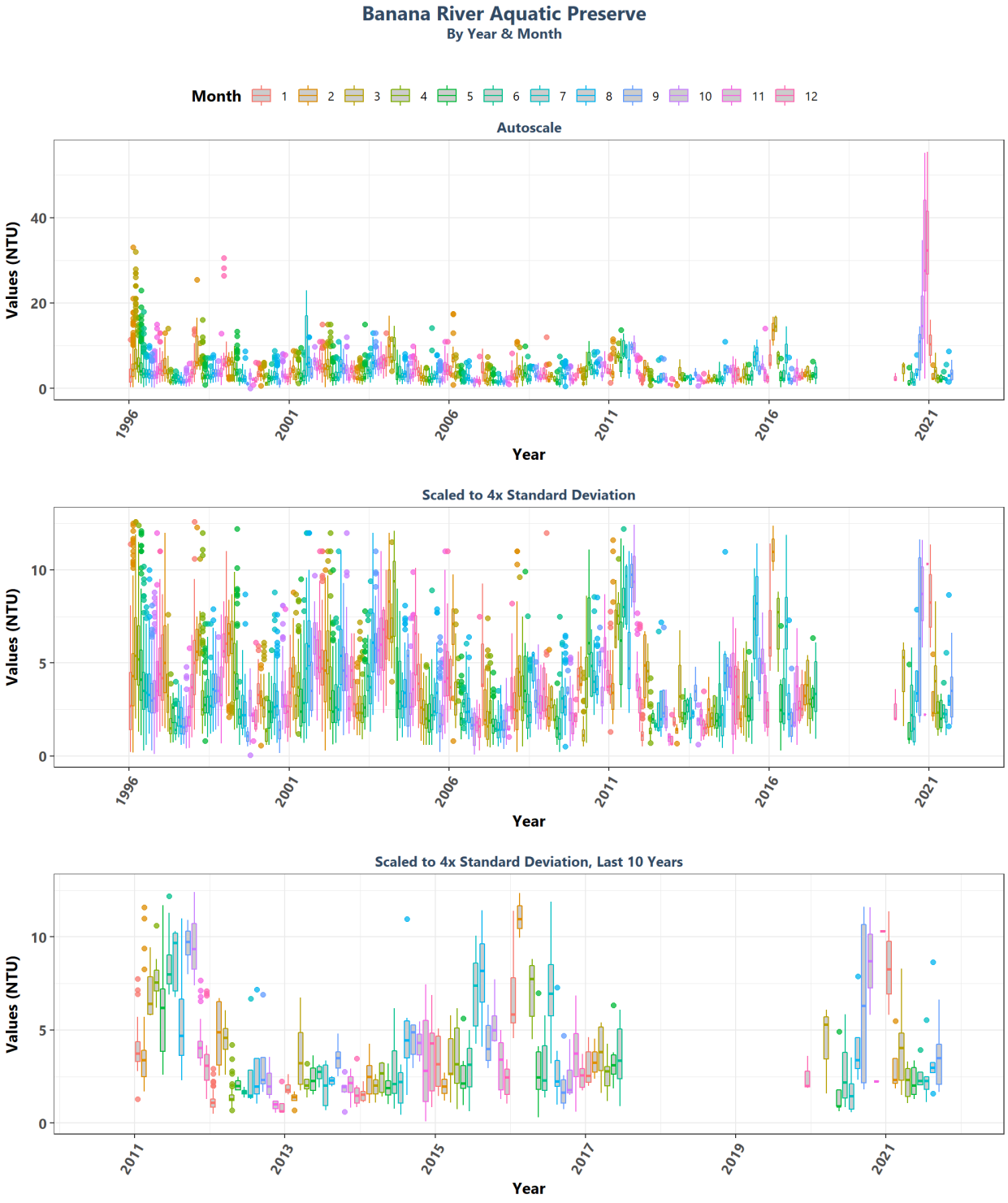

Apalachicola National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

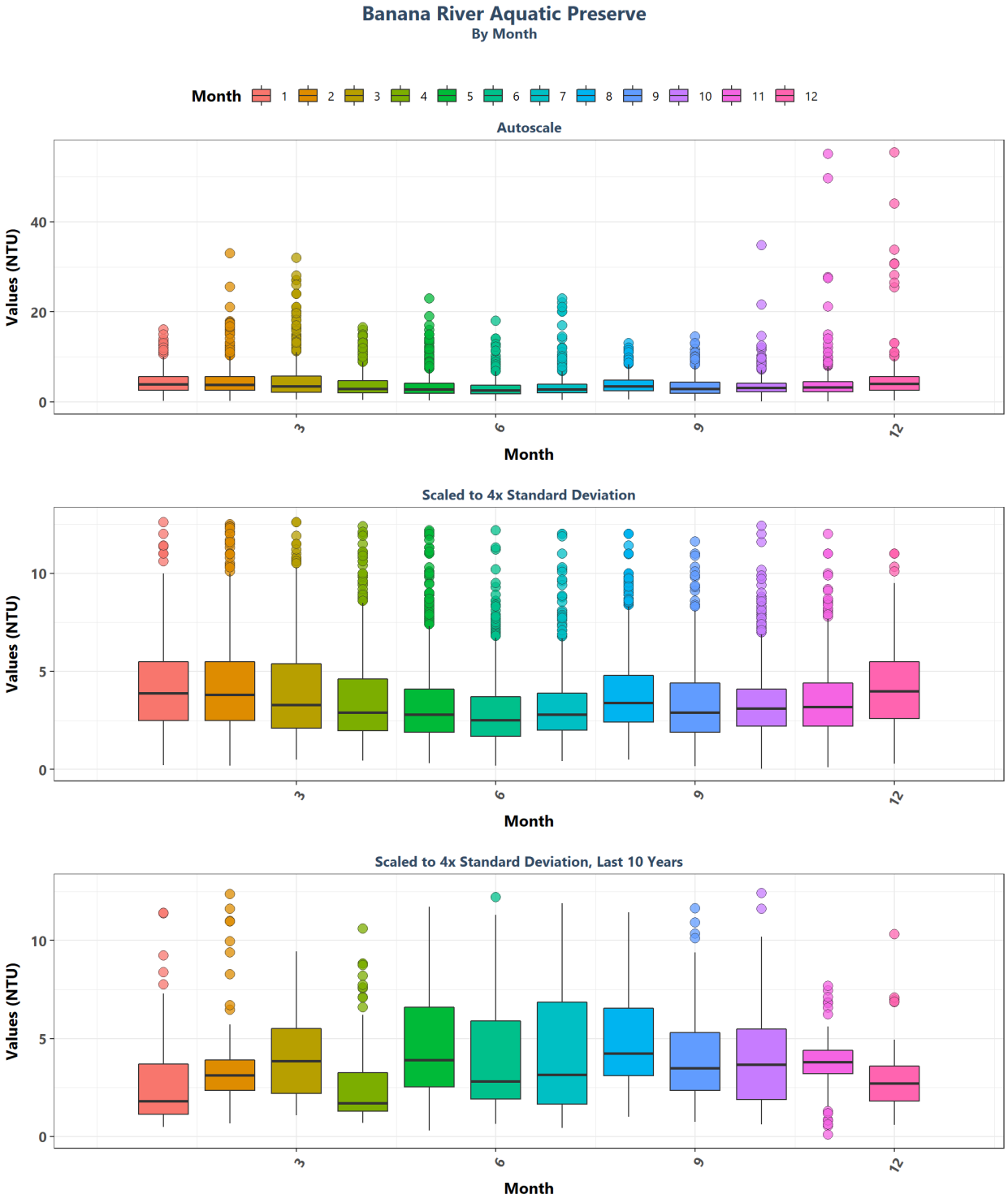

Apalachicola National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
150
100
50
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
20
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
20
15
10
5
0
3
6
9
12
Month

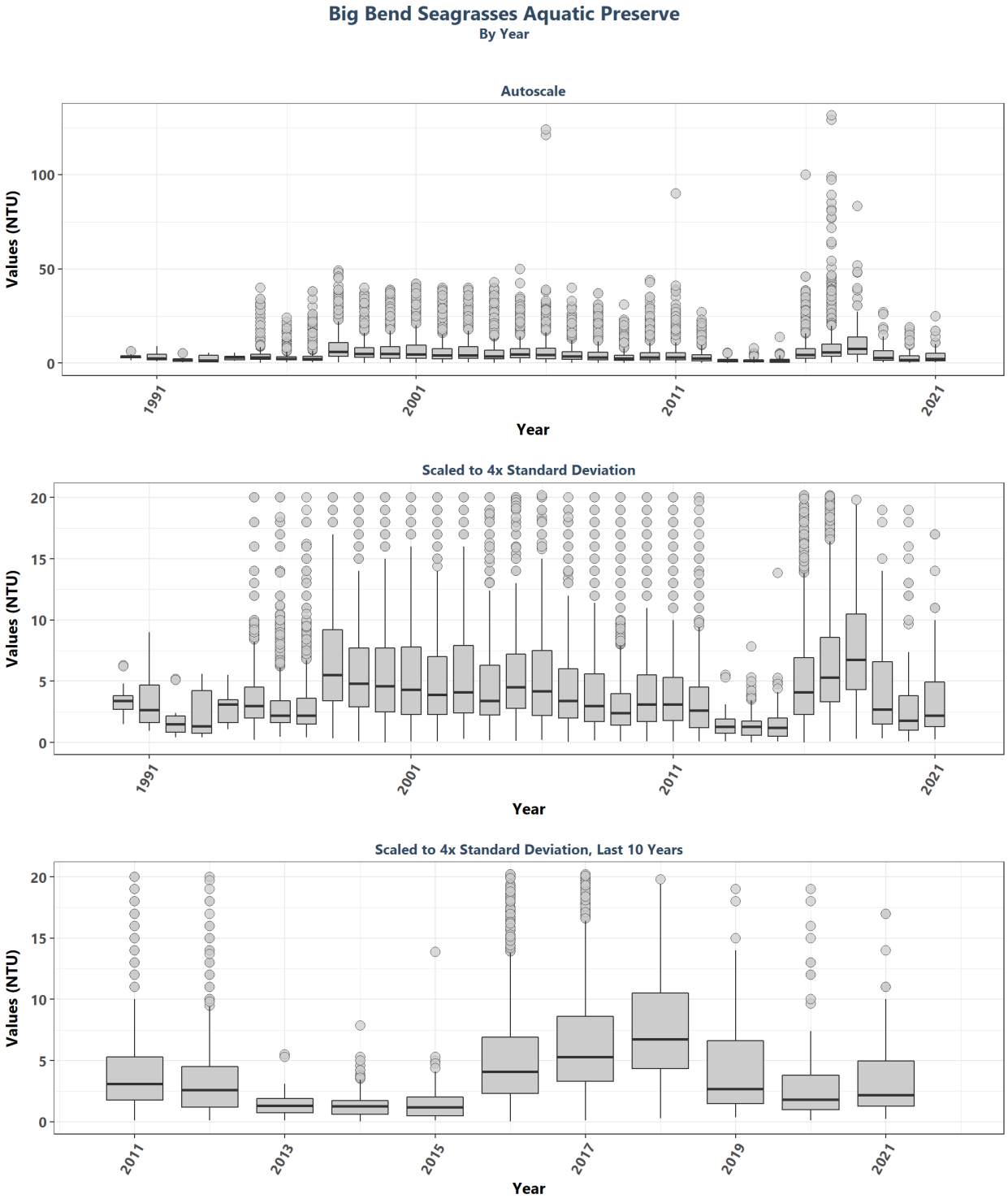
Banana River Aquatic Preserve

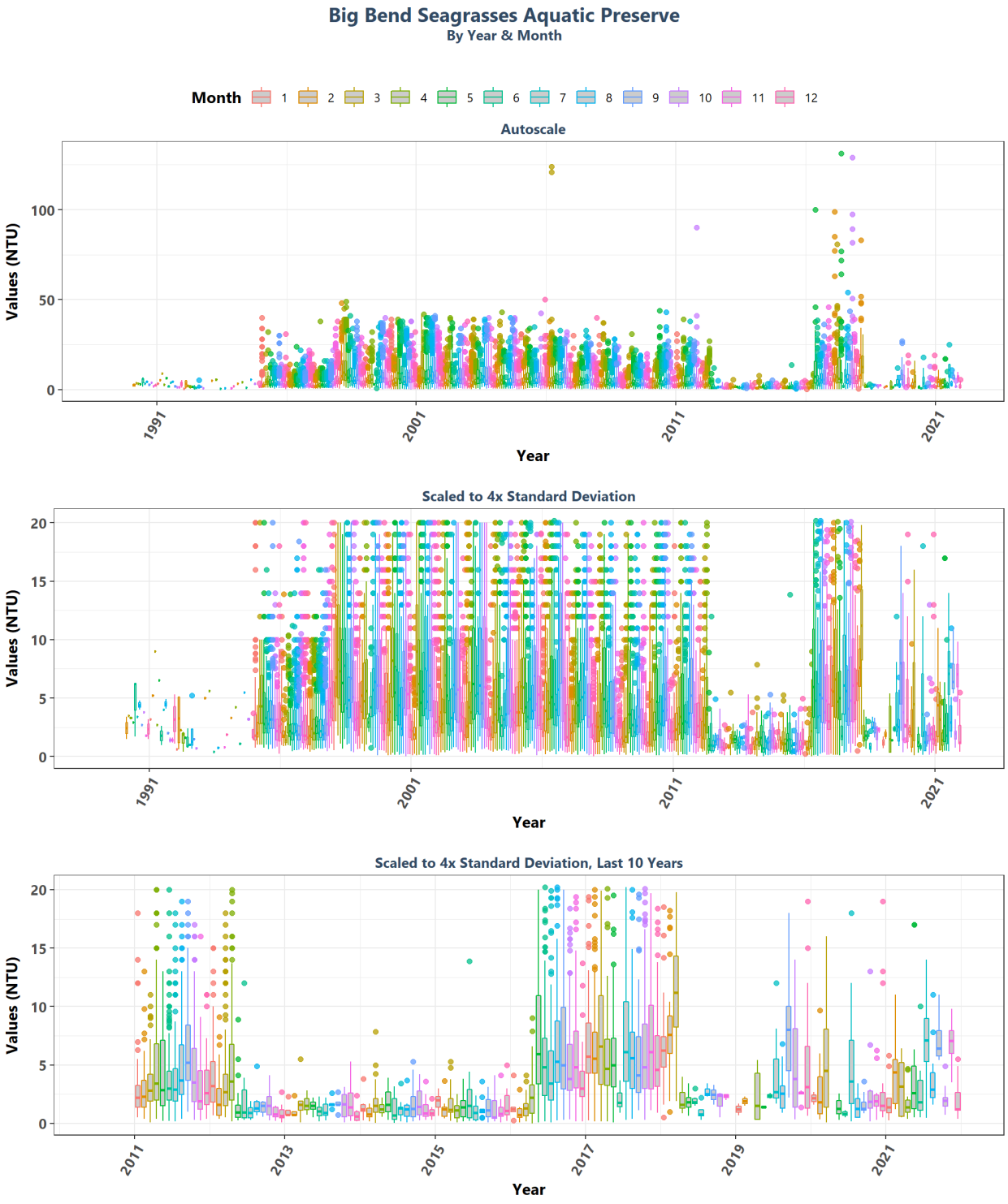
By Year

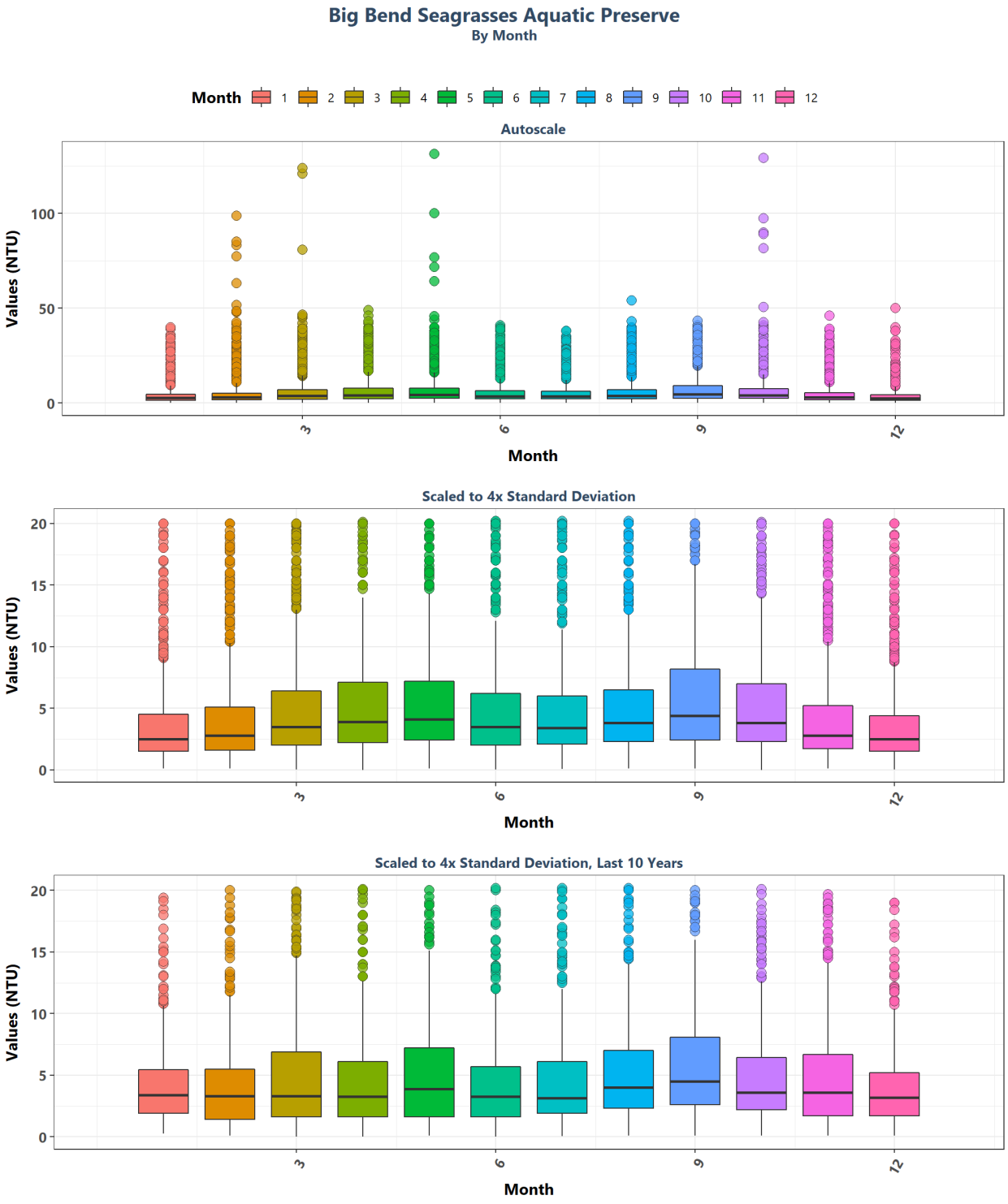


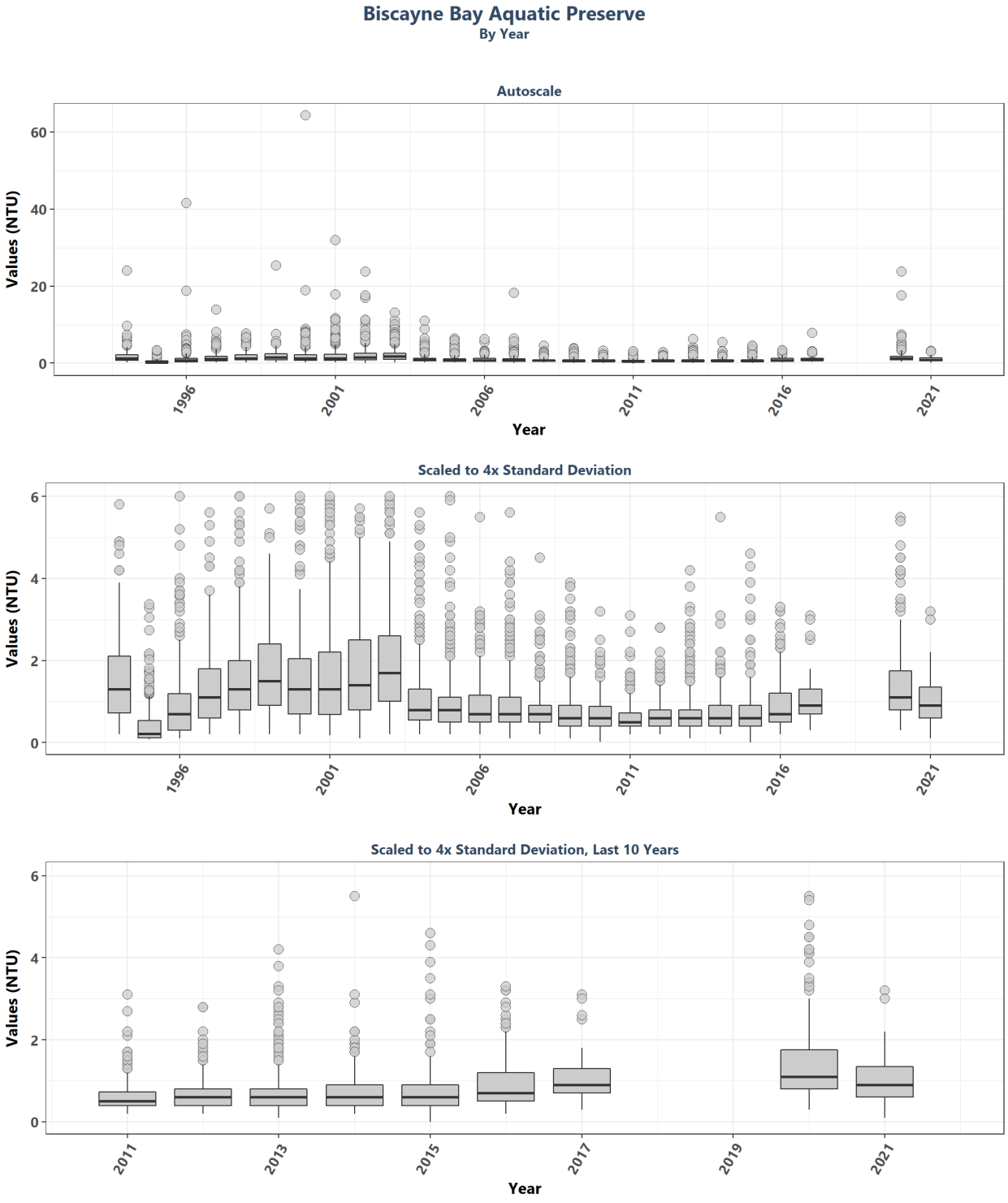

Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
20
40
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
2011
2013
2015
2017
2019
2021
Year

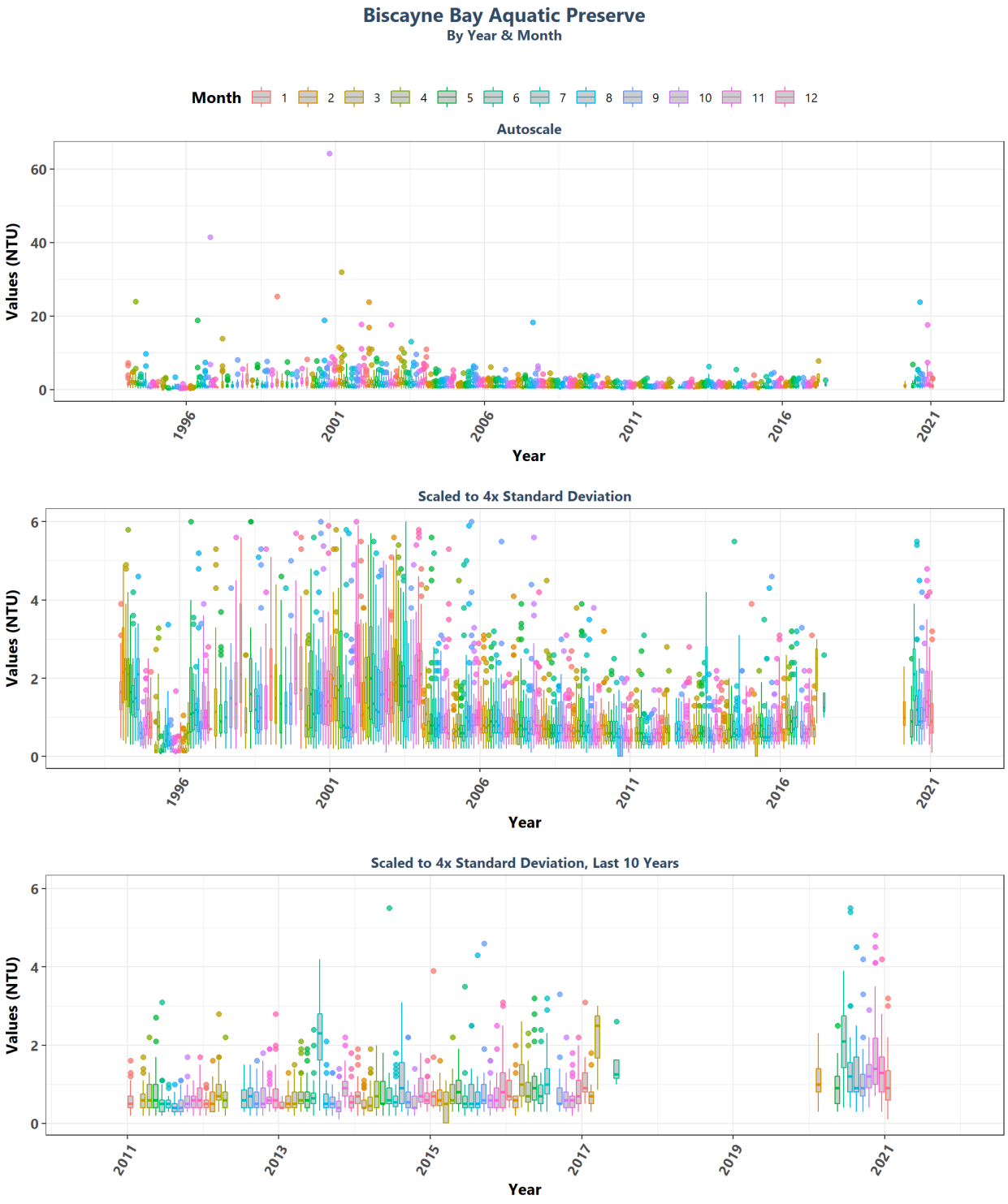

Banana River Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
20
40
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
3
6
9
12
Month

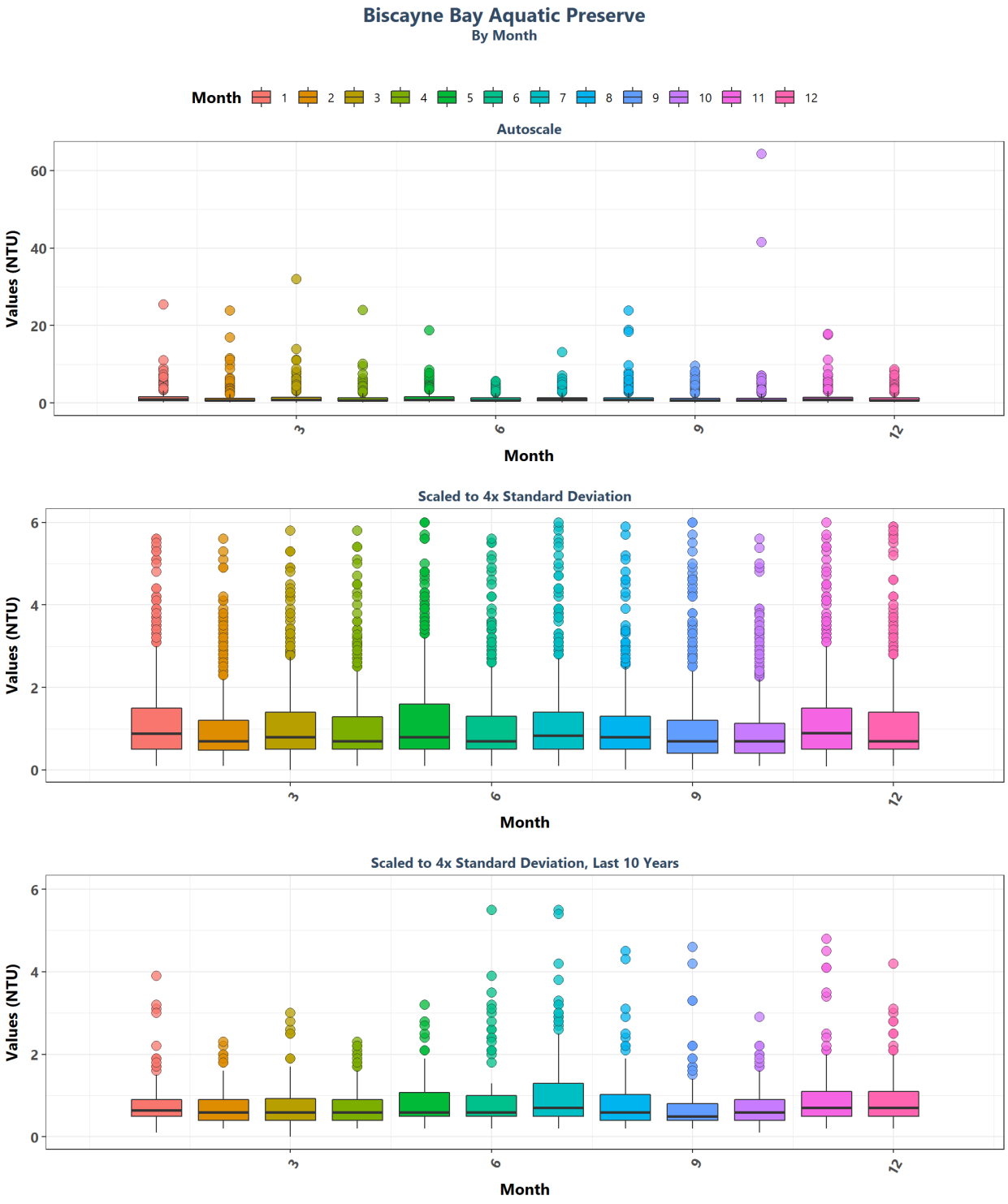

Big Bend Seagrasses Aquatic Preserve
By Year
Autoscale
Values (NTU)
100
50
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
20
15
10
5
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
20
15
10
5
0
2011
2013
2015
2017
2019
2021
Year


Big Bend Seagrasses Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
100
50
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
20
15
10
5
0
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
20
15
10
5
0
2011
2013
2015
2017
2019
2021
Year


Big Bend Seagrasses Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
100
50
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
20
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
20
15
10
5
0
3
6
9
12
Month

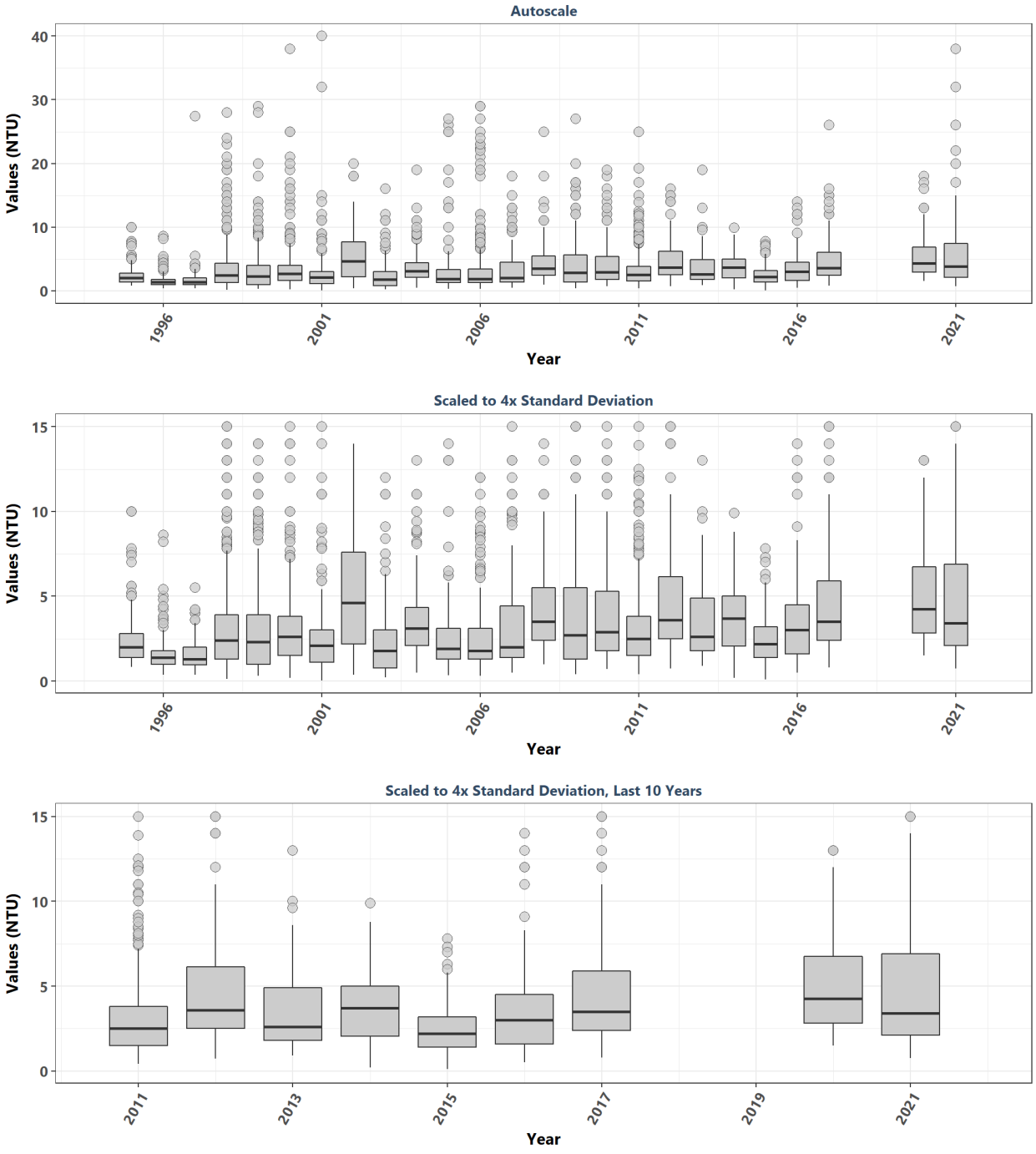

Biscayne Bay Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

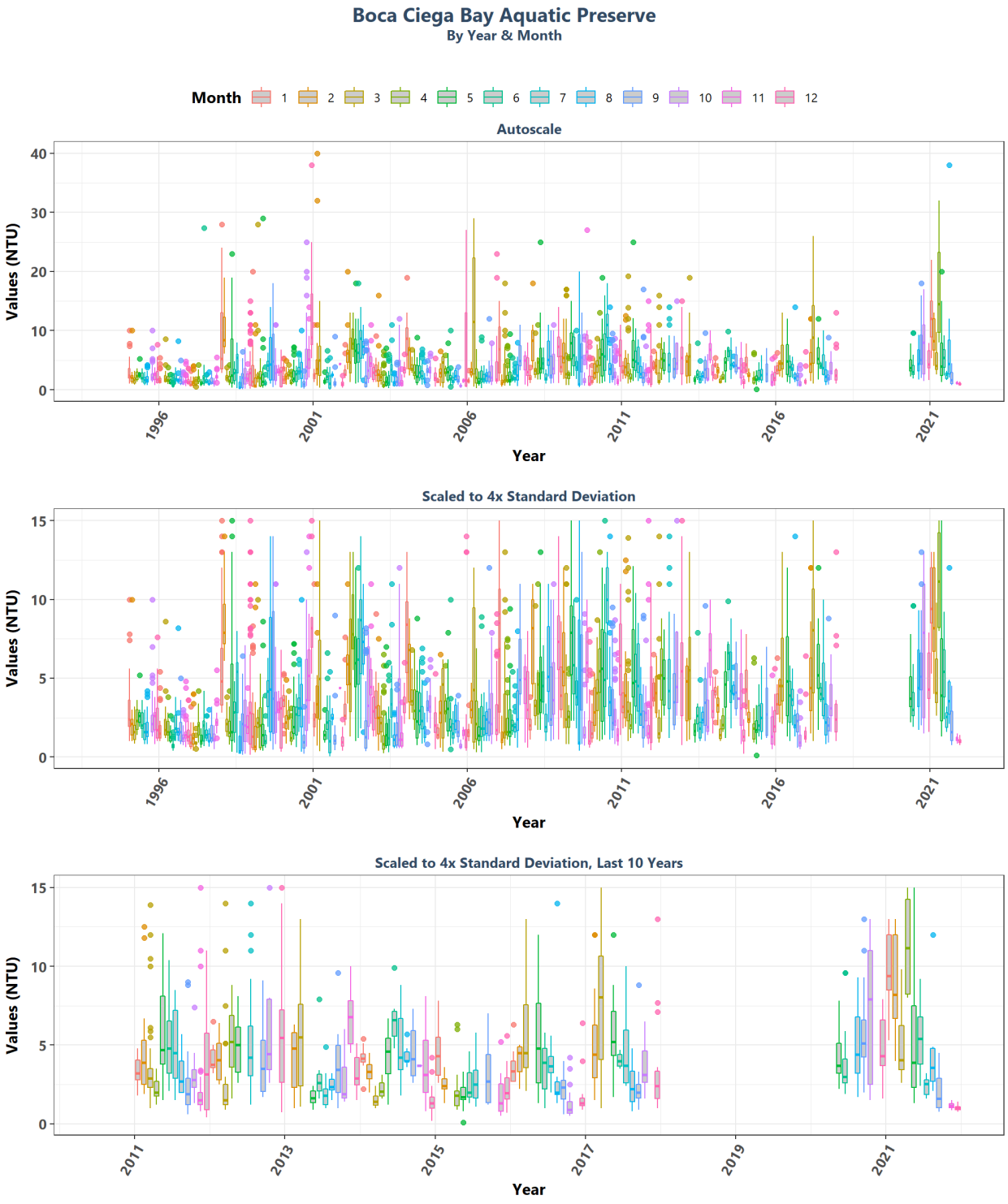

Biscayne Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

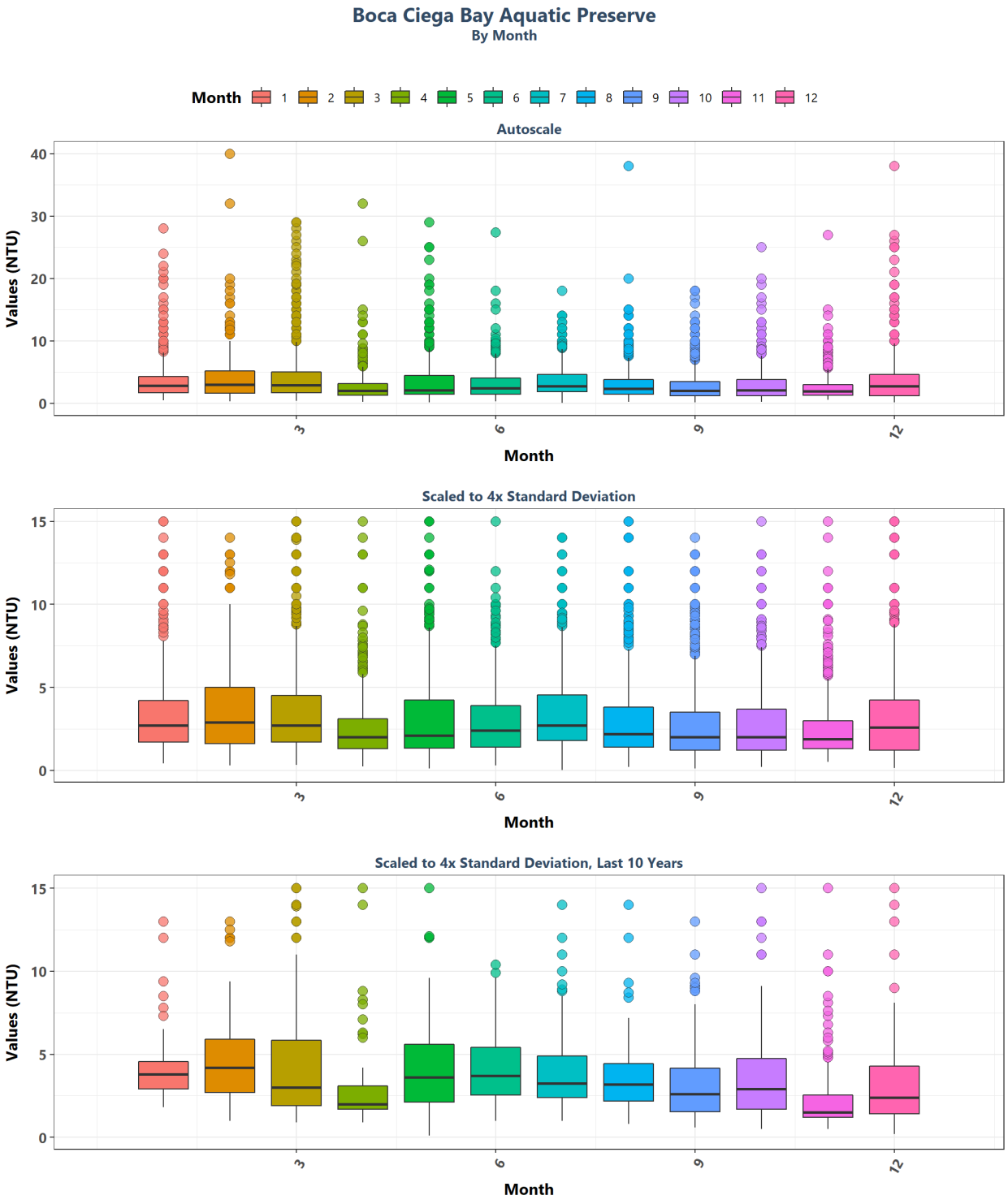

Biscayne Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
20
40
60
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
0
2
4
6
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
2
4
6
3
6
9
12
Month

Boca Ciega Bay Aquatic Preserve

By Year

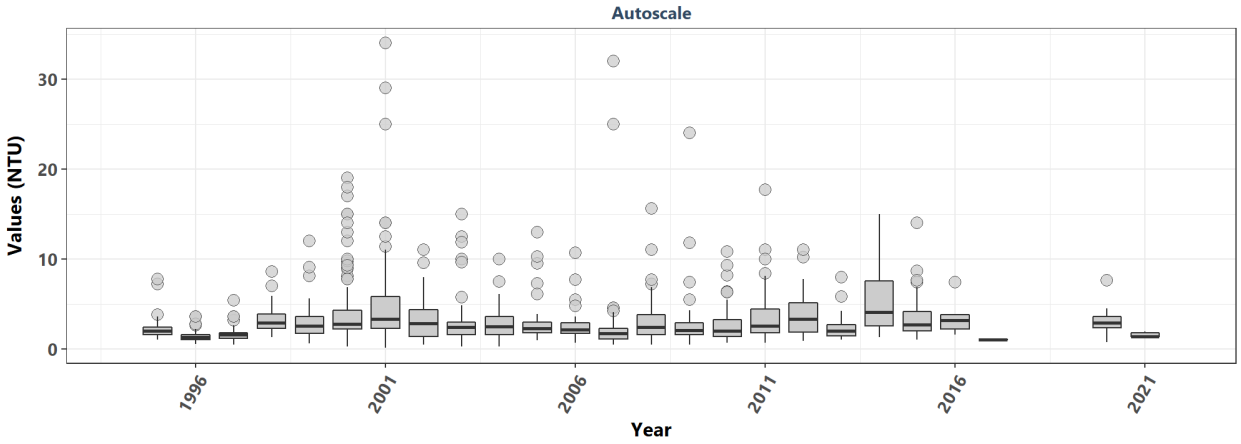


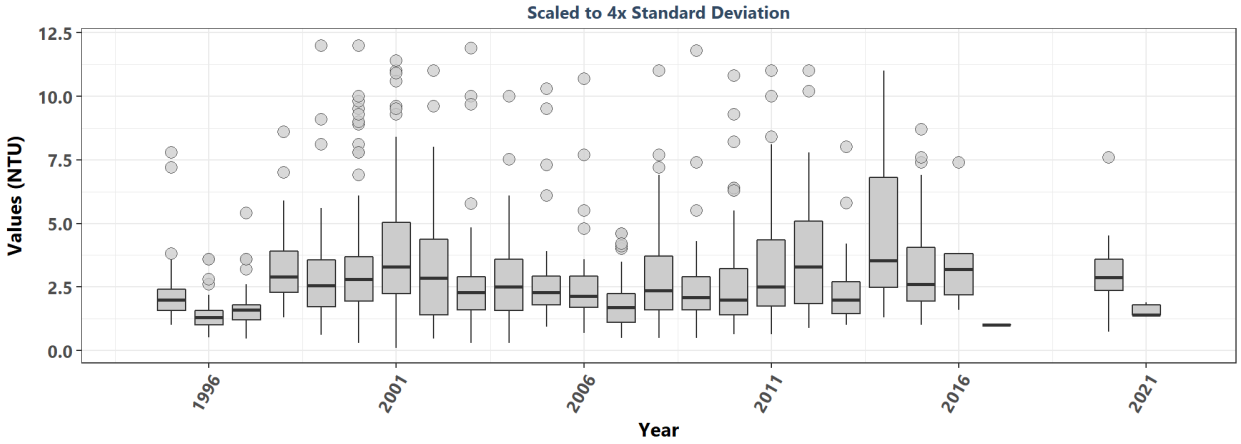

Boca Ciega Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

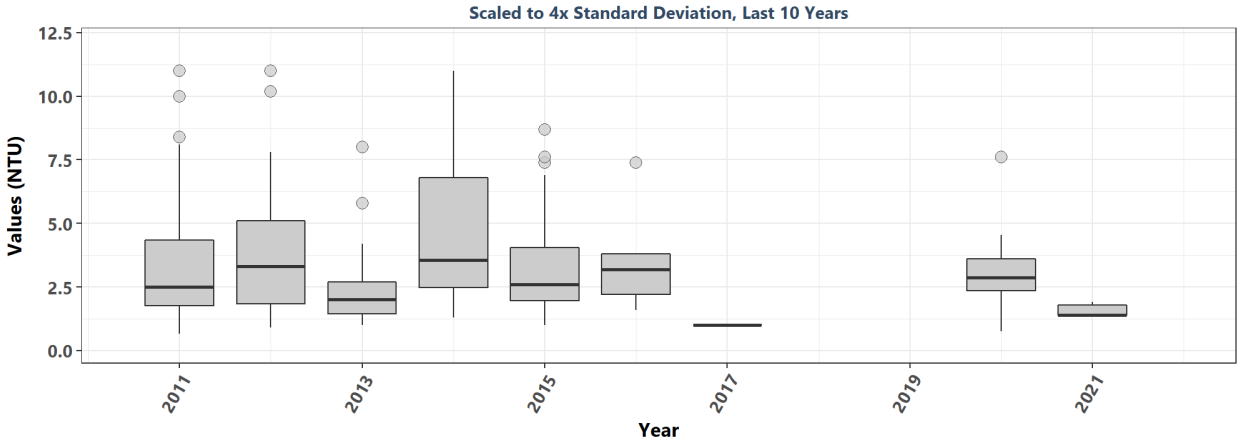

Boca Ciega Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

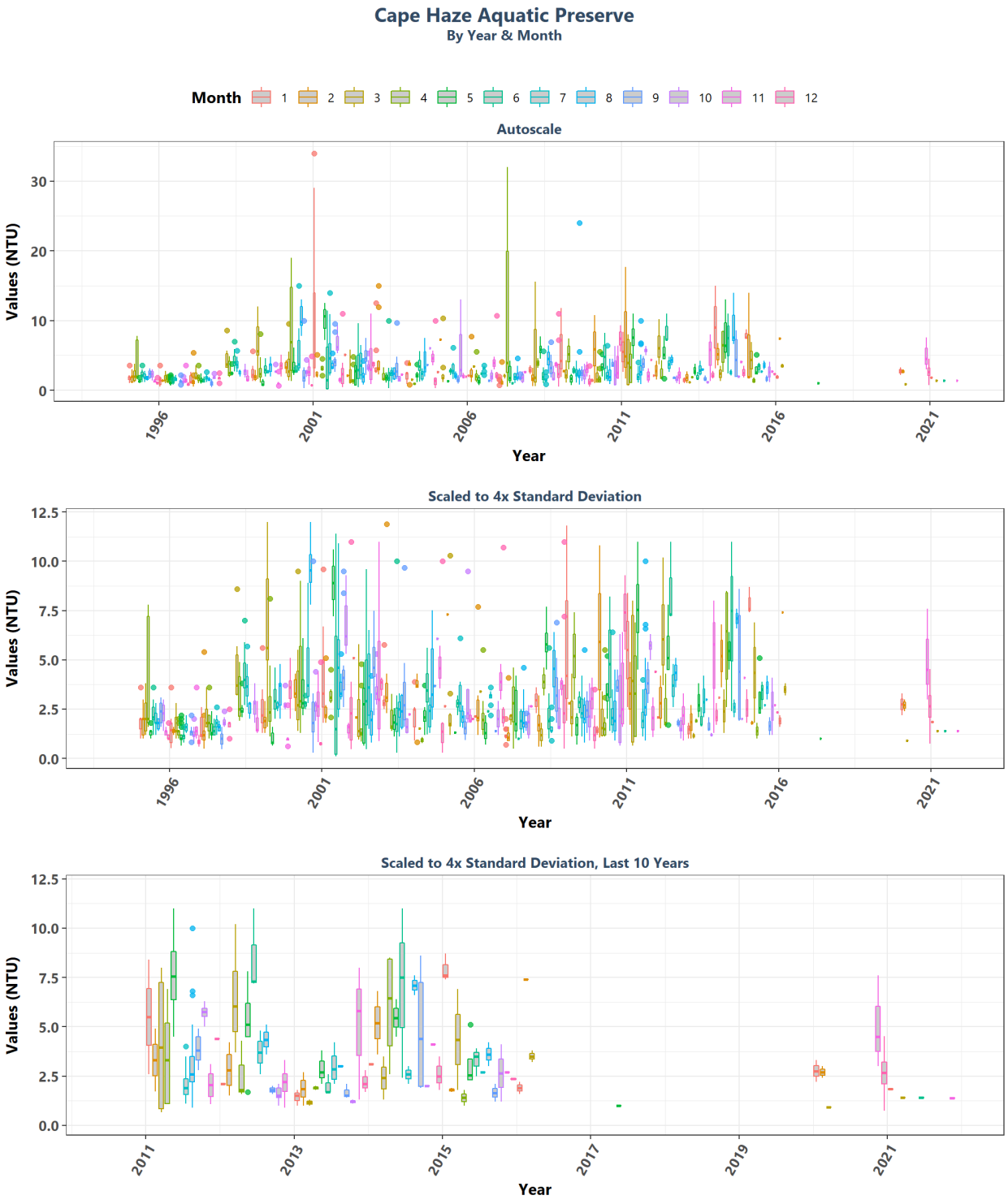
Cape Haze Aquatic Preserve

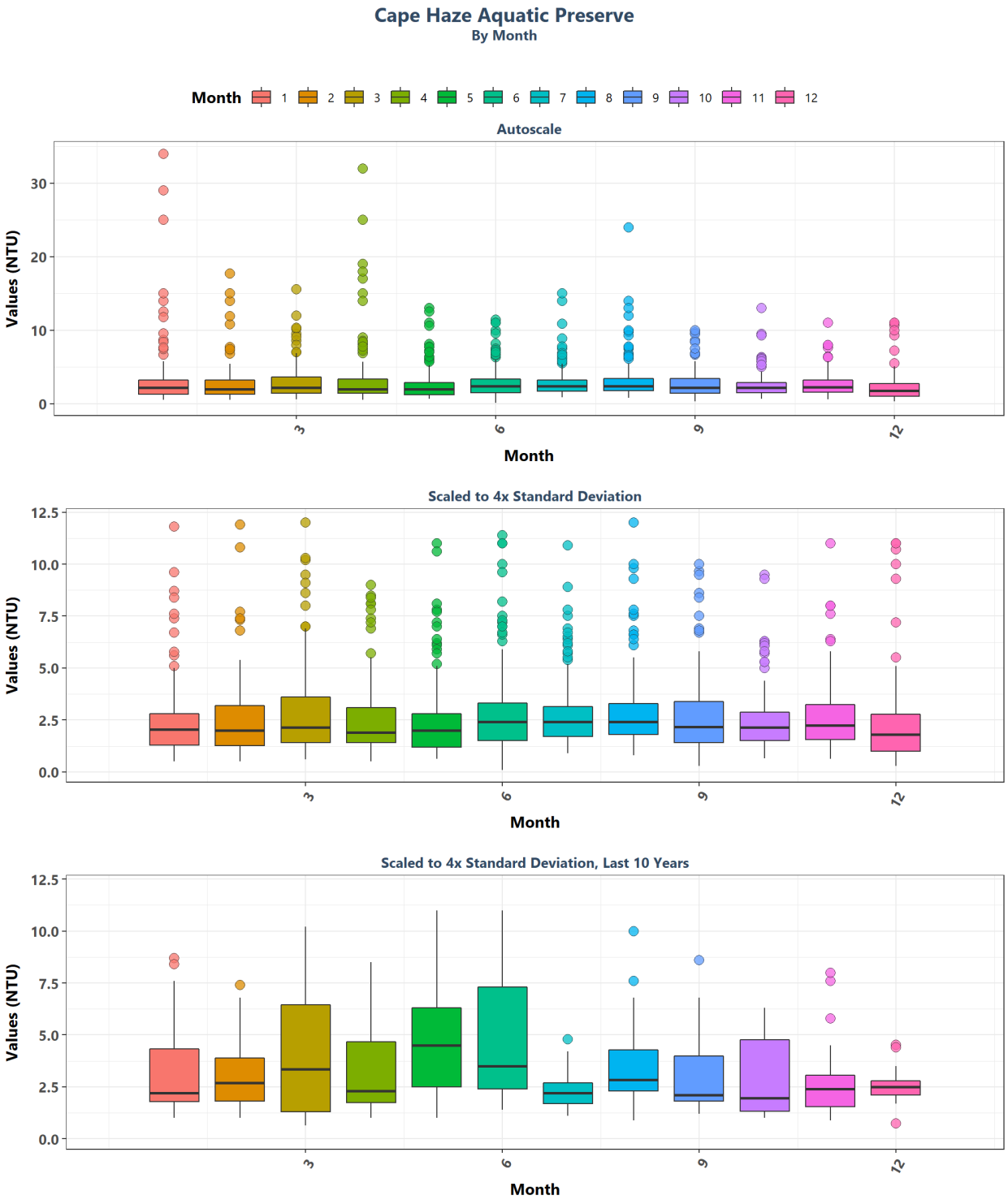
By Year





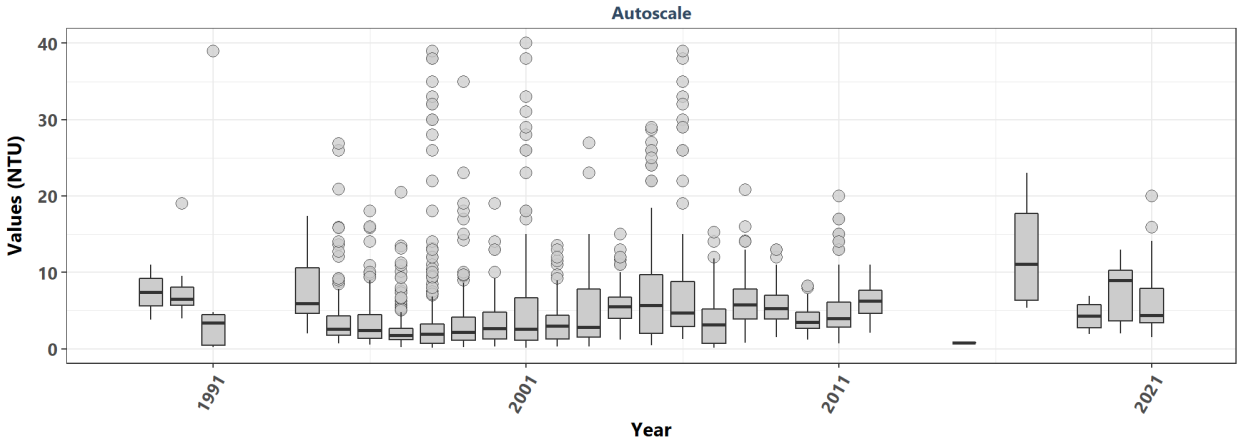


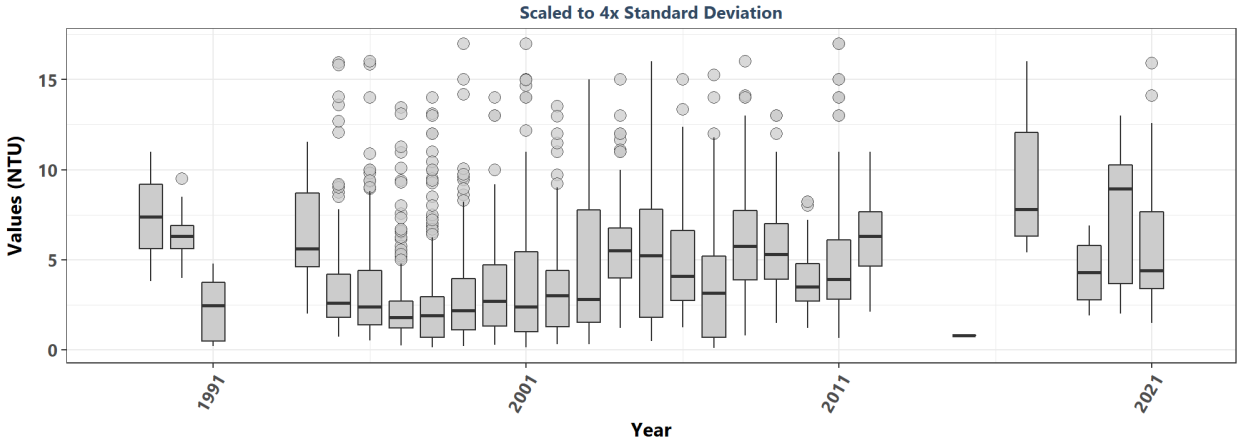

Cape Haze Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

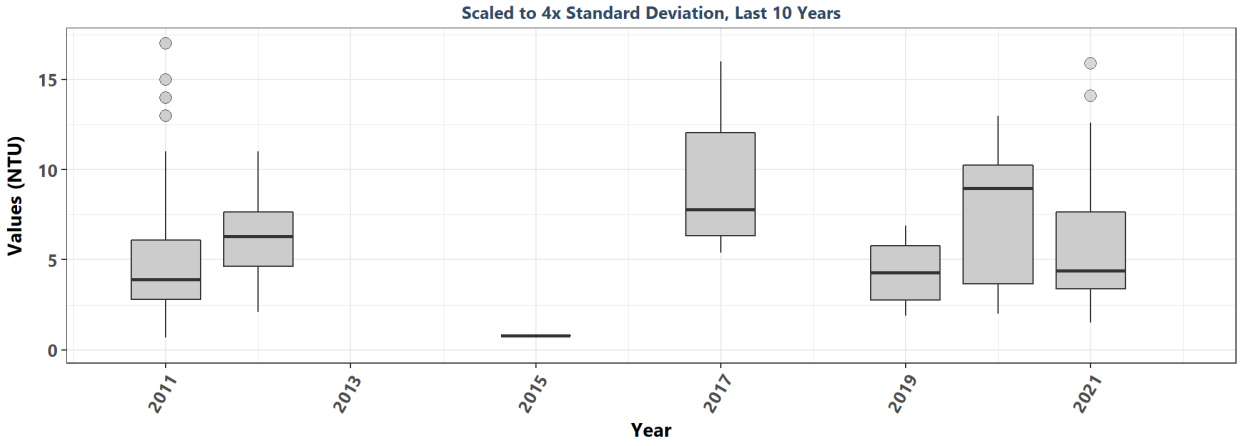

Cape Haze Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

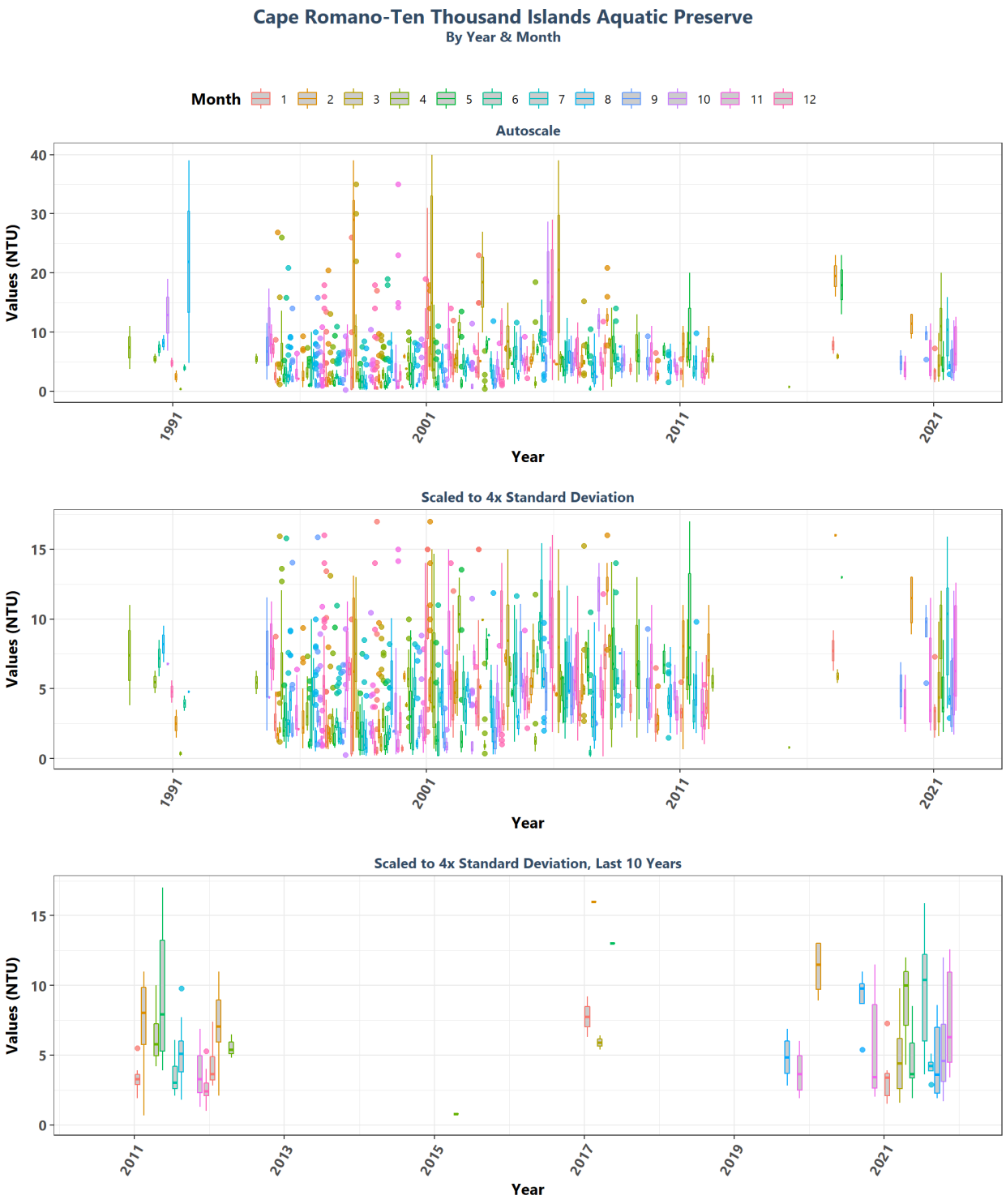
Cape Romano-Ten Thousand Islands Aquatic Preserve

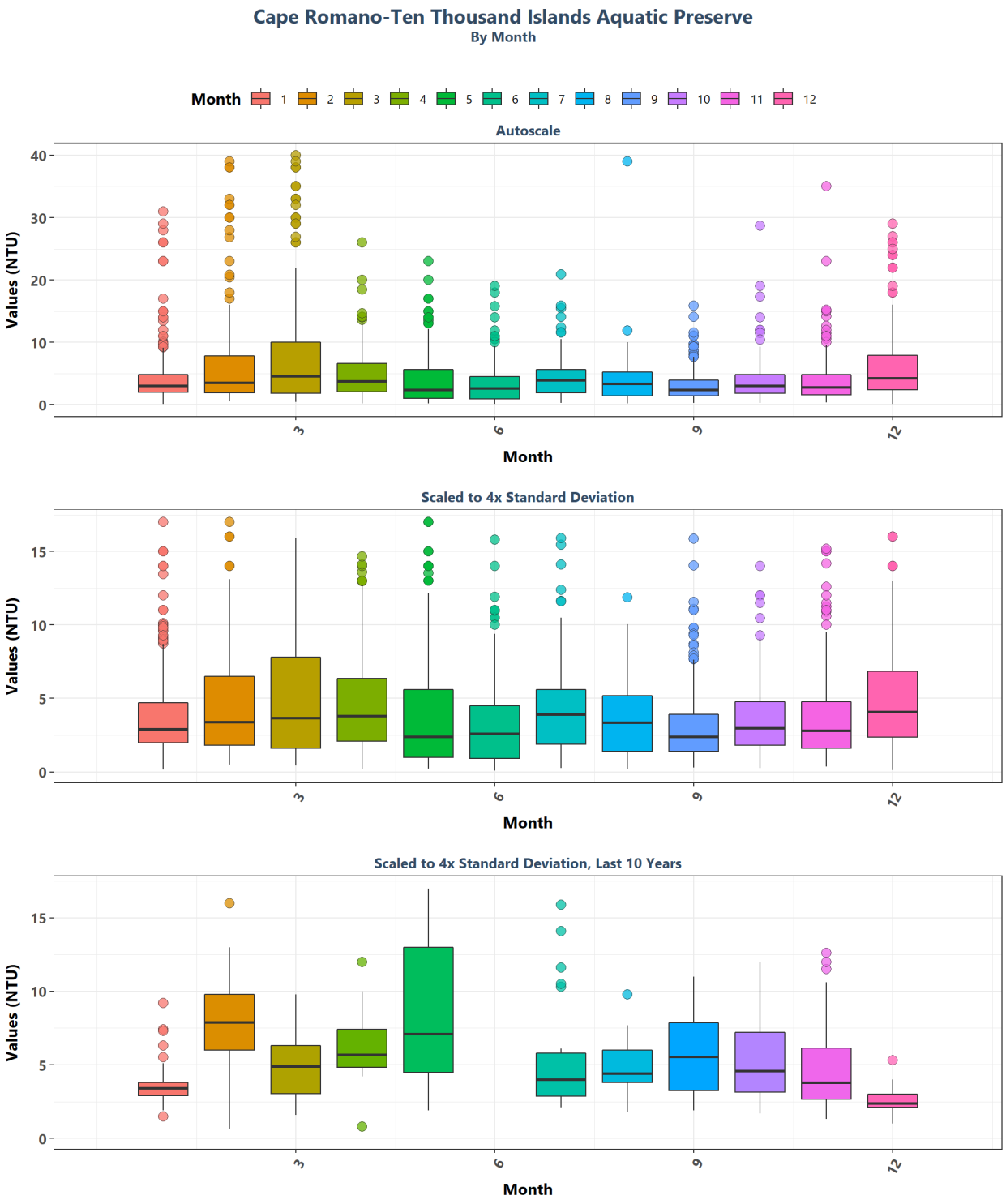
By Year





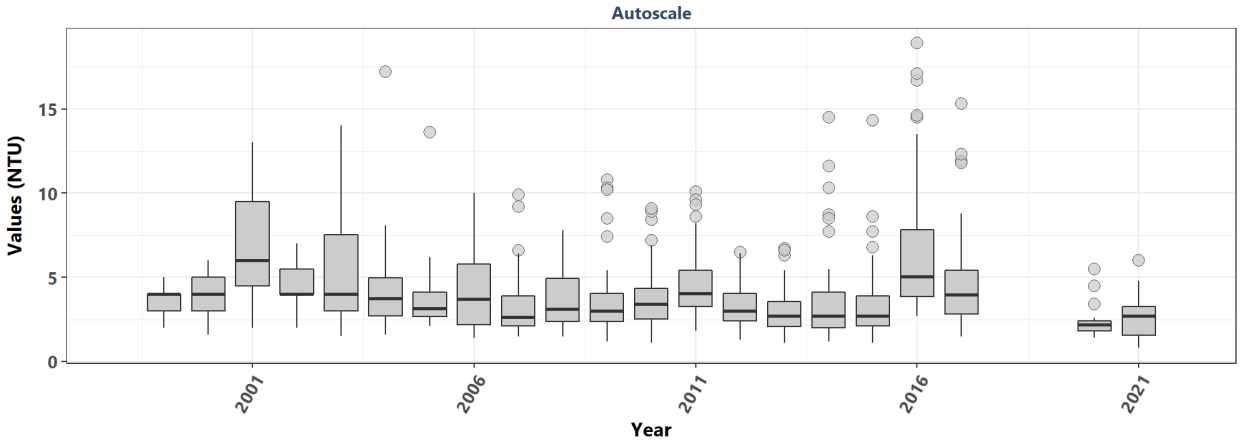


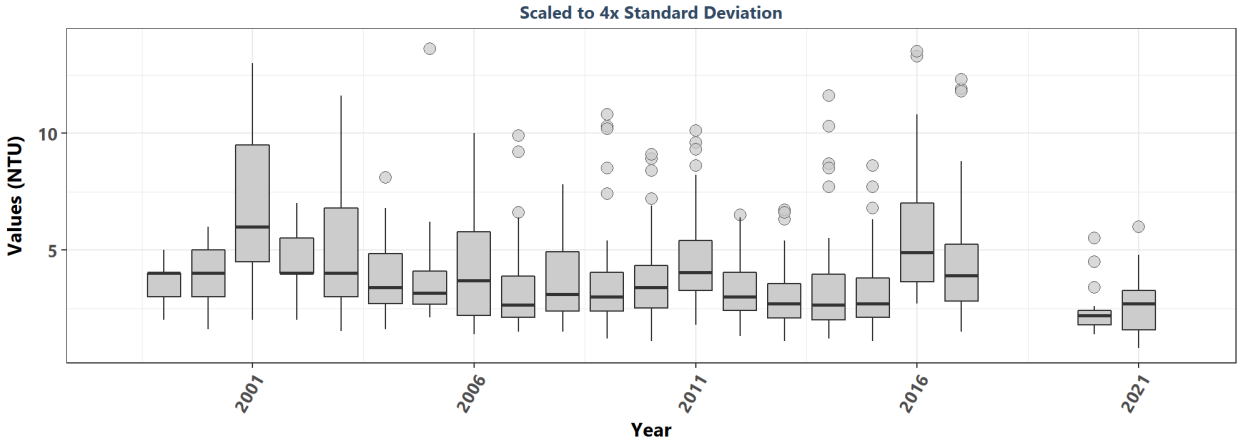

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

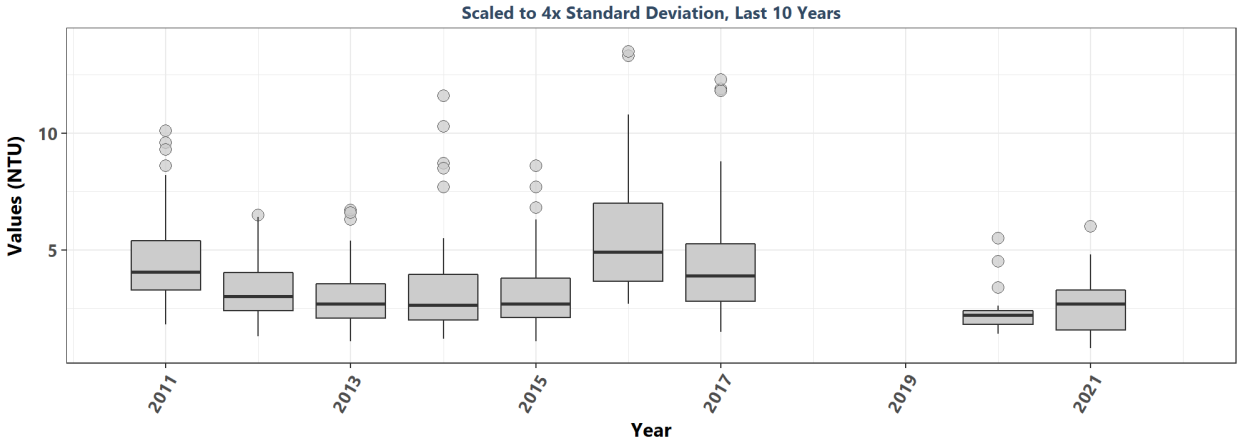

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

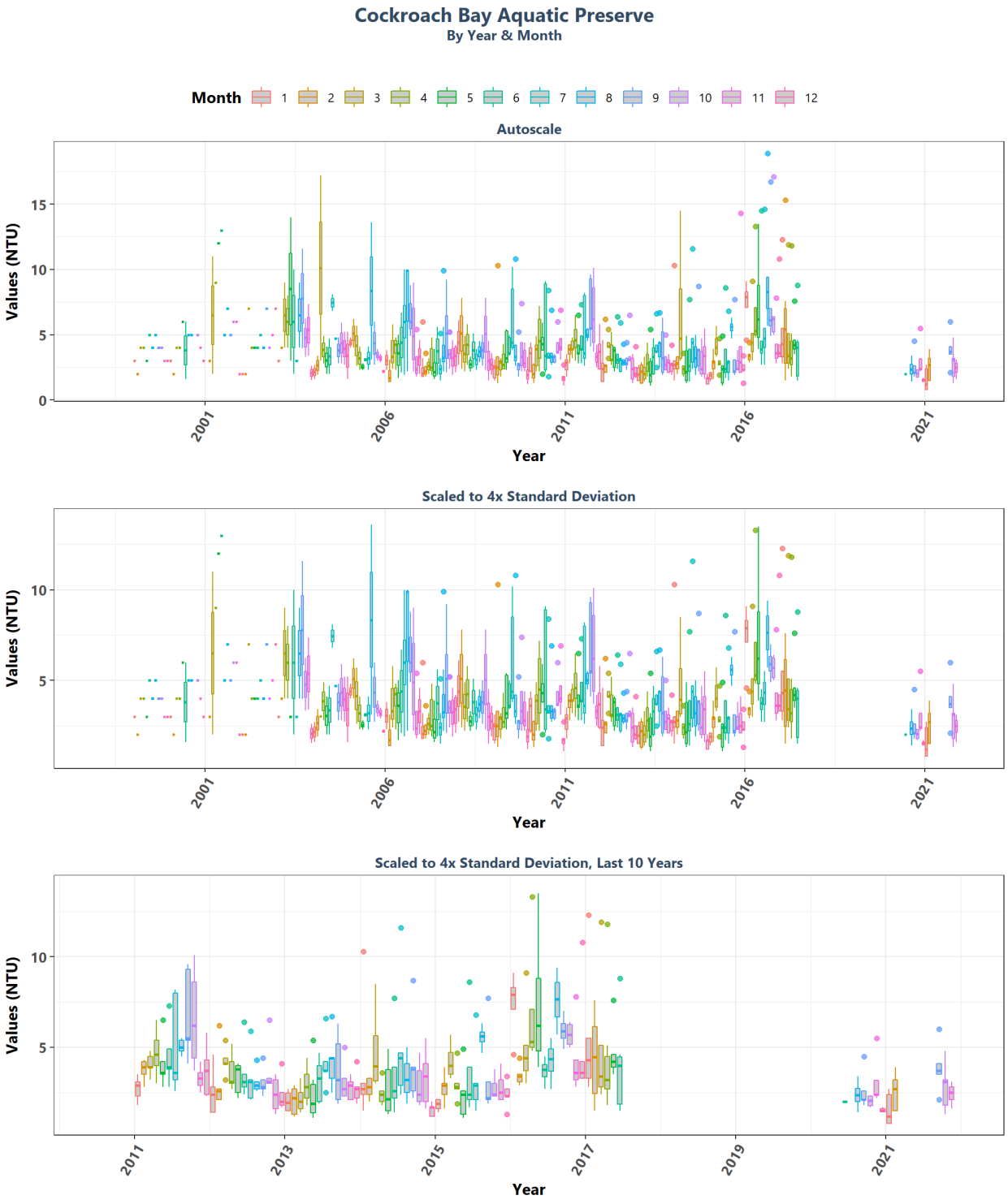
Cockroach Bay Aquatic Preserve

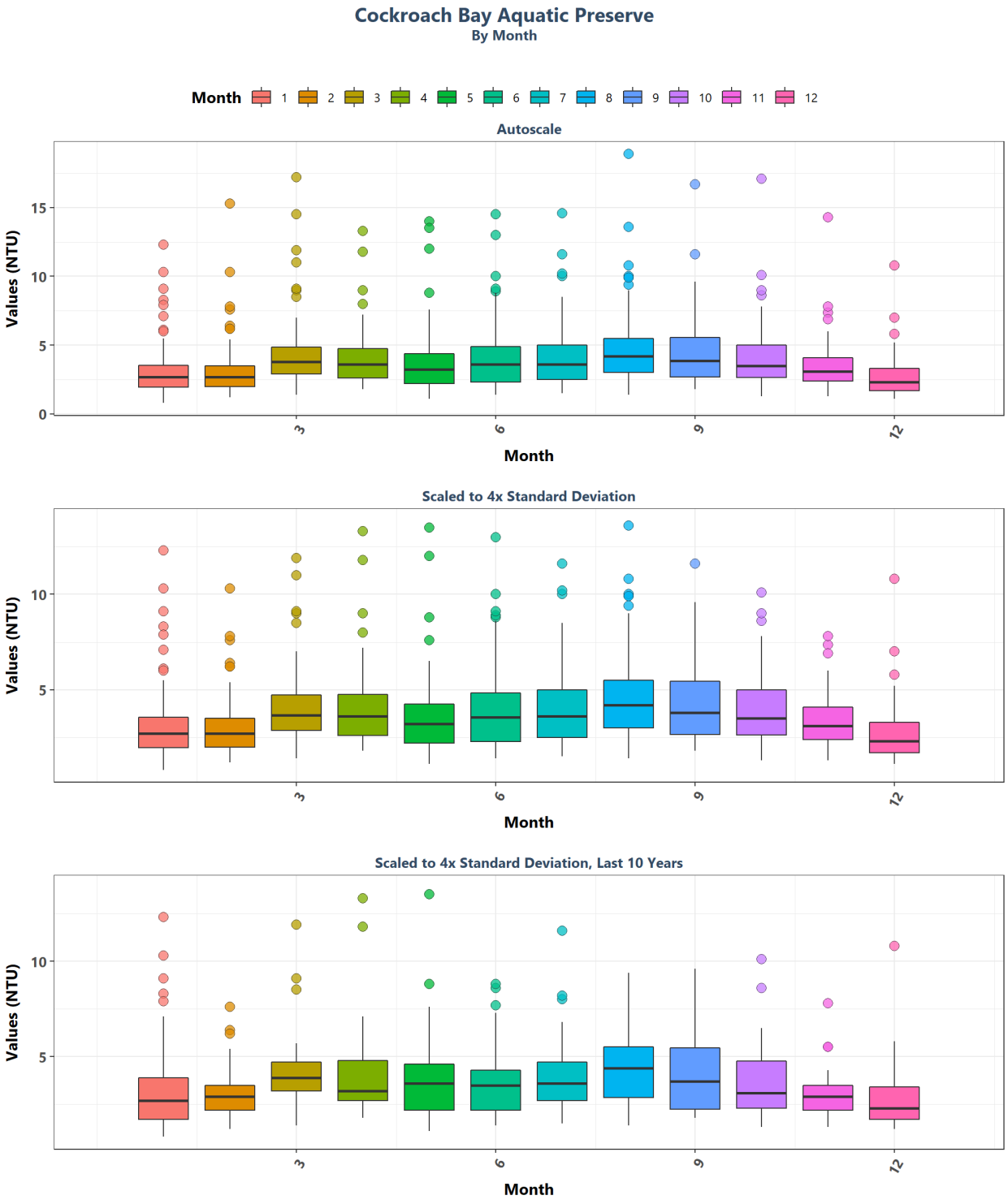
By Year



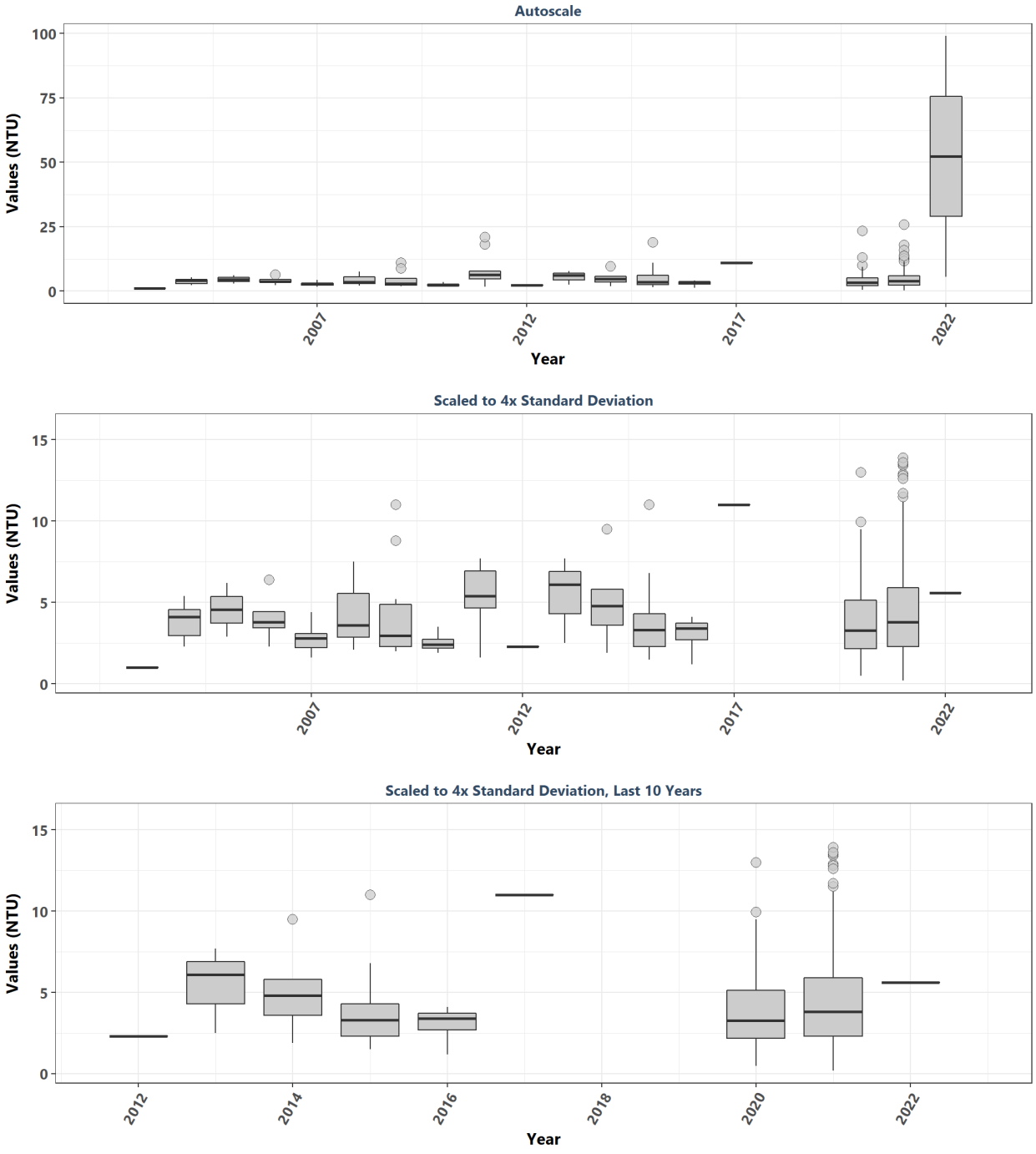


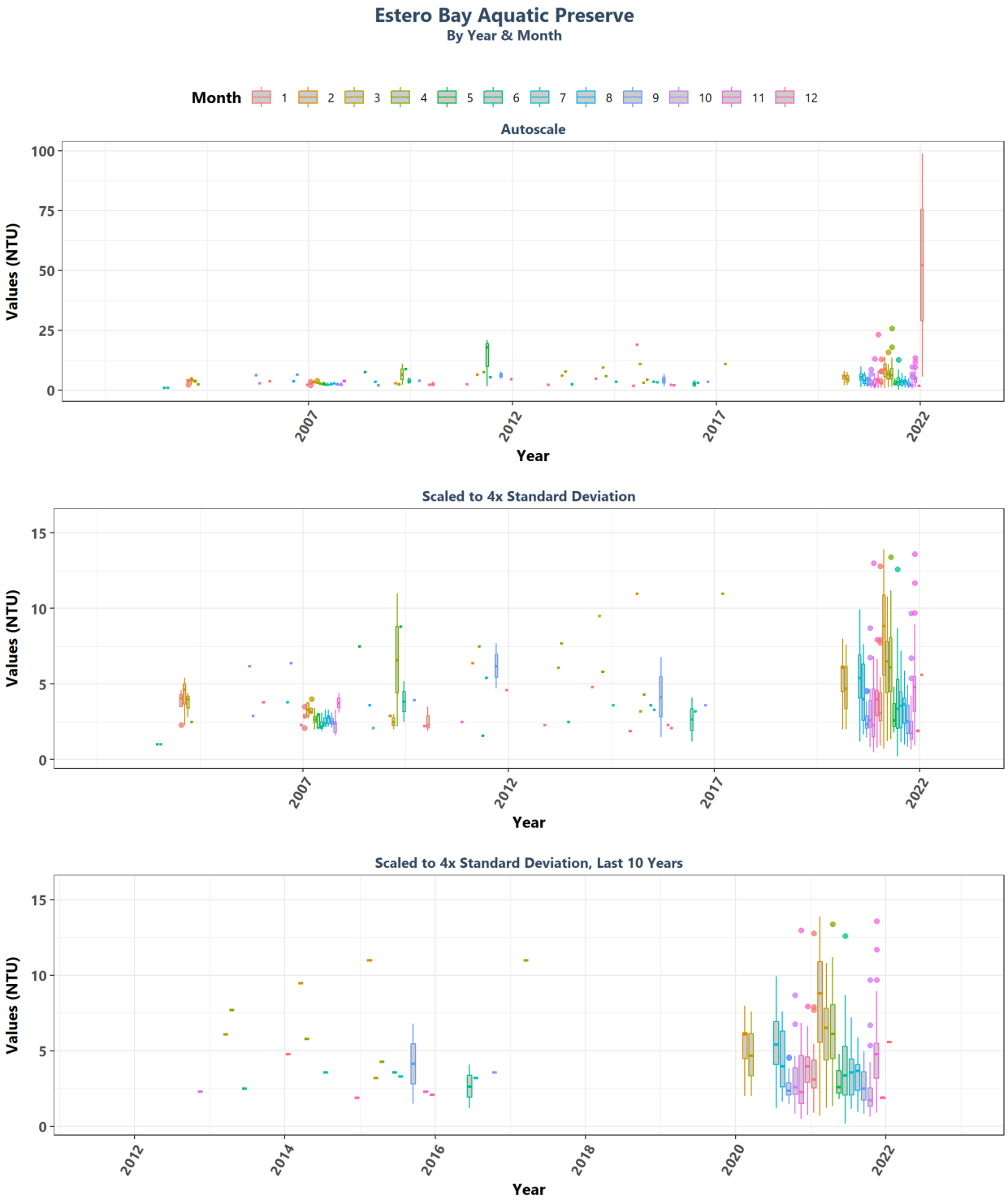


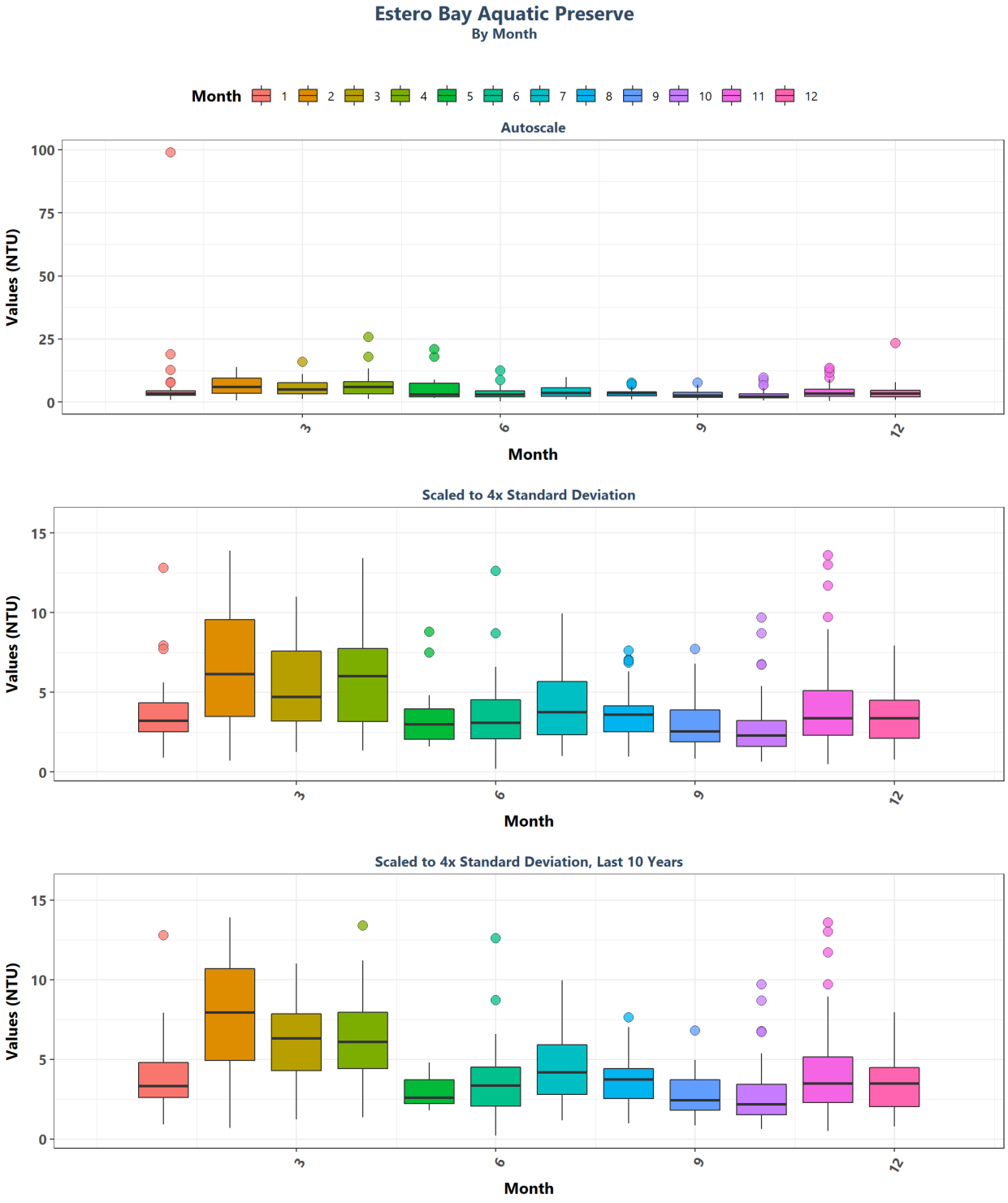

Cockroach Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
5
10
15
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
5
10
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
5
10
2011
2013
2015
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 (NTU)
0
5
10
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
5
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
5
10
3
6
9
12
Month

Estero Bay Aquatic Preserve
By Year

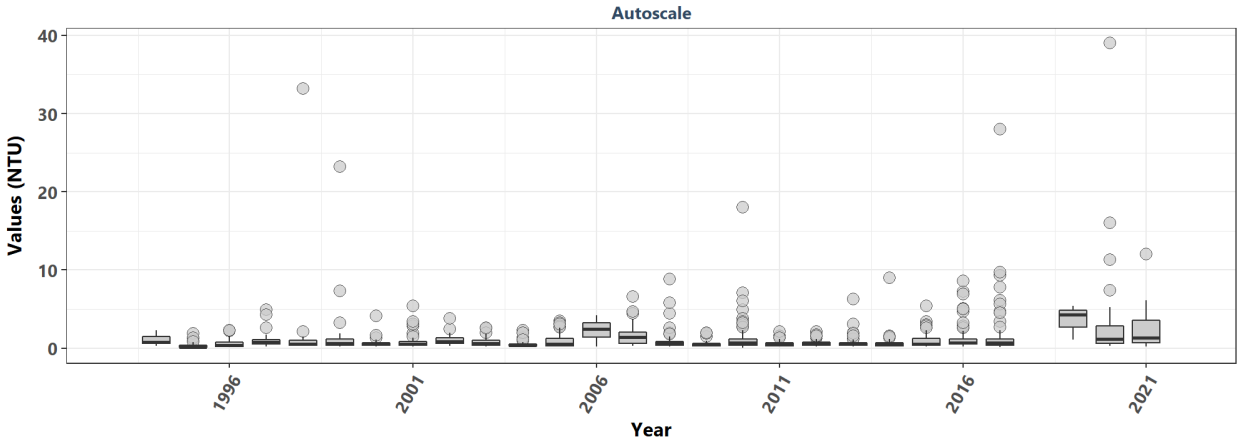


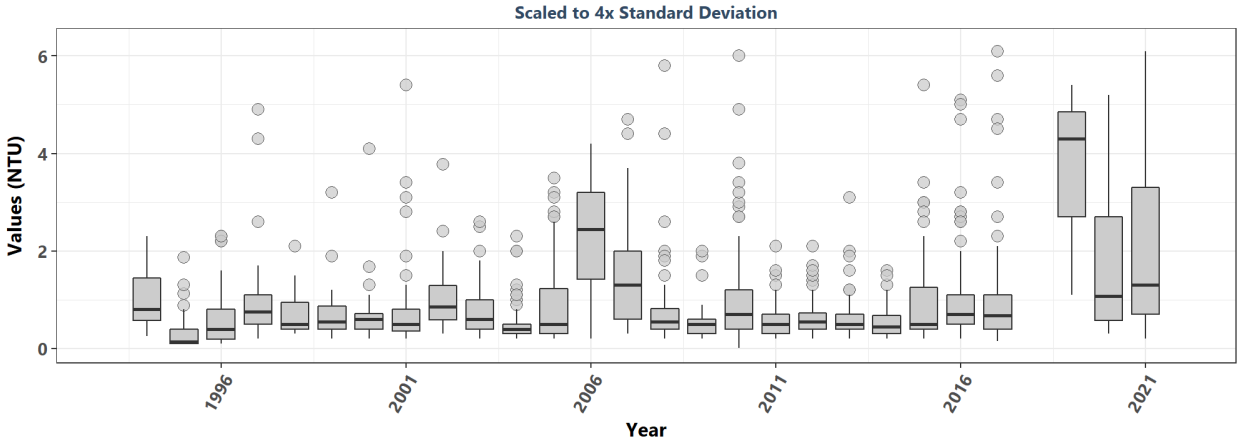

Estero Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
100
75
50
25
0
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (NTU)
15
10
5
0
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
15
10
5
0
2012
2014
2016
2018
2020
2022
Year

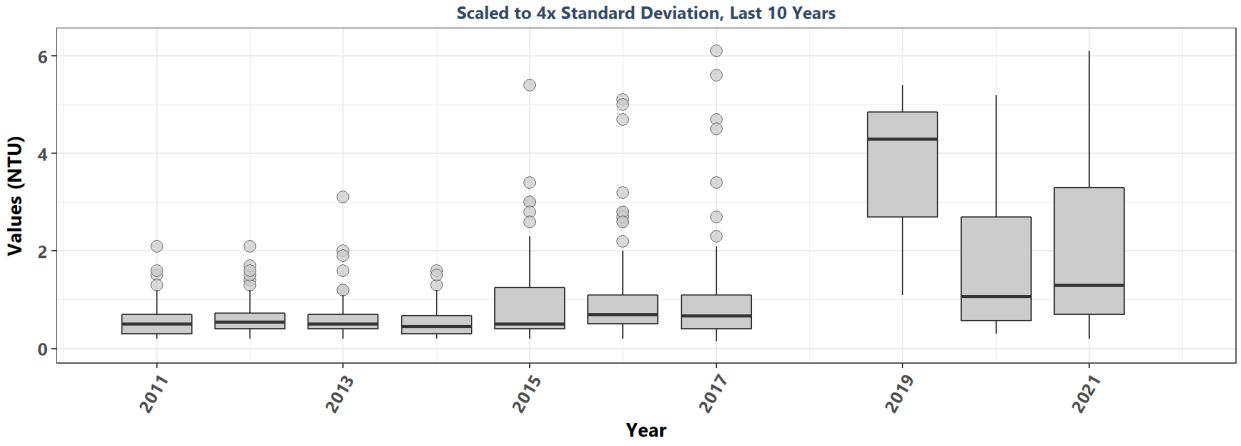

Estero Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
100
75
50
25
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
15
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
15
10
5
0
3
6
9
12
Month

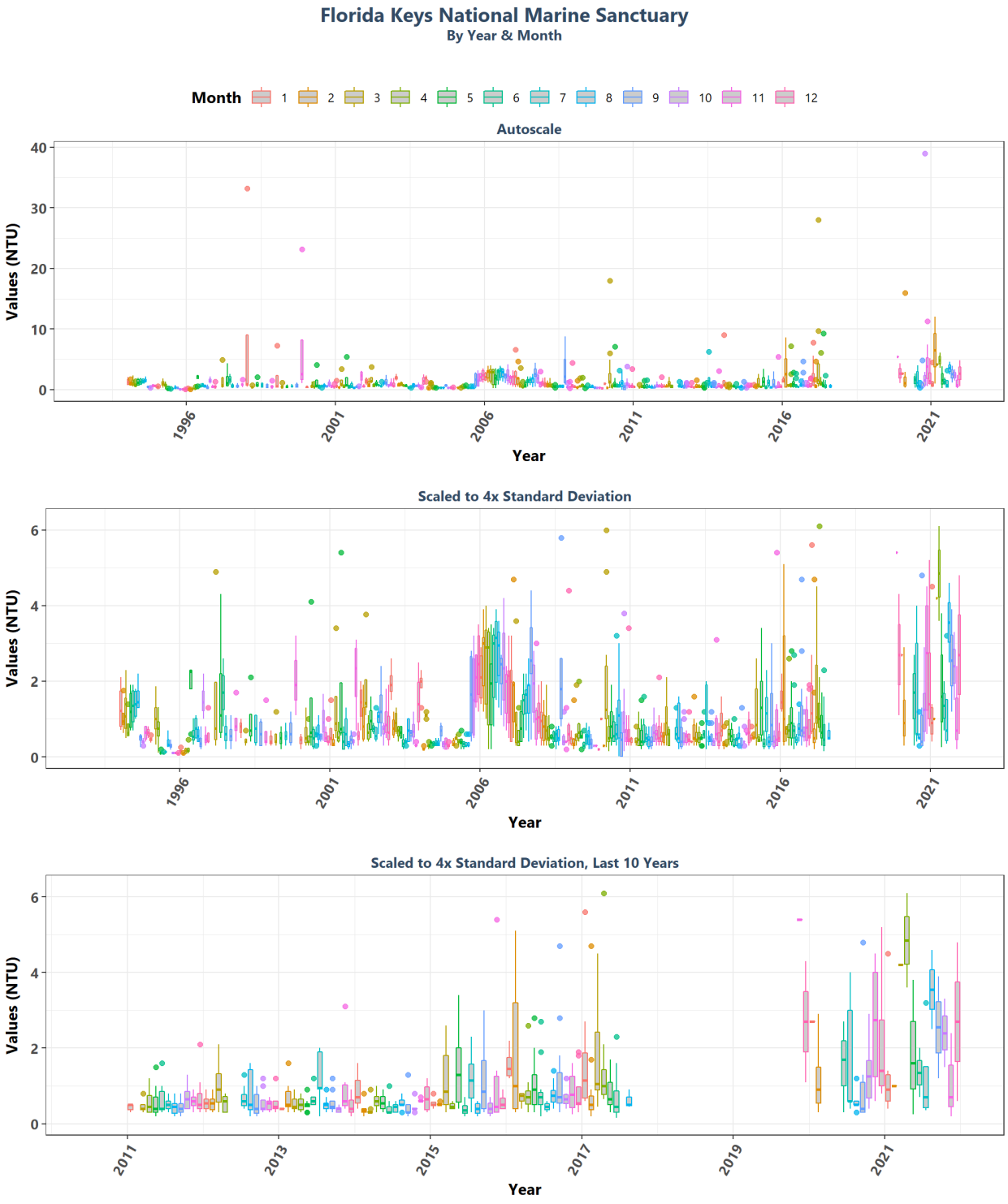
Florida Keys National Marine Sanctuary

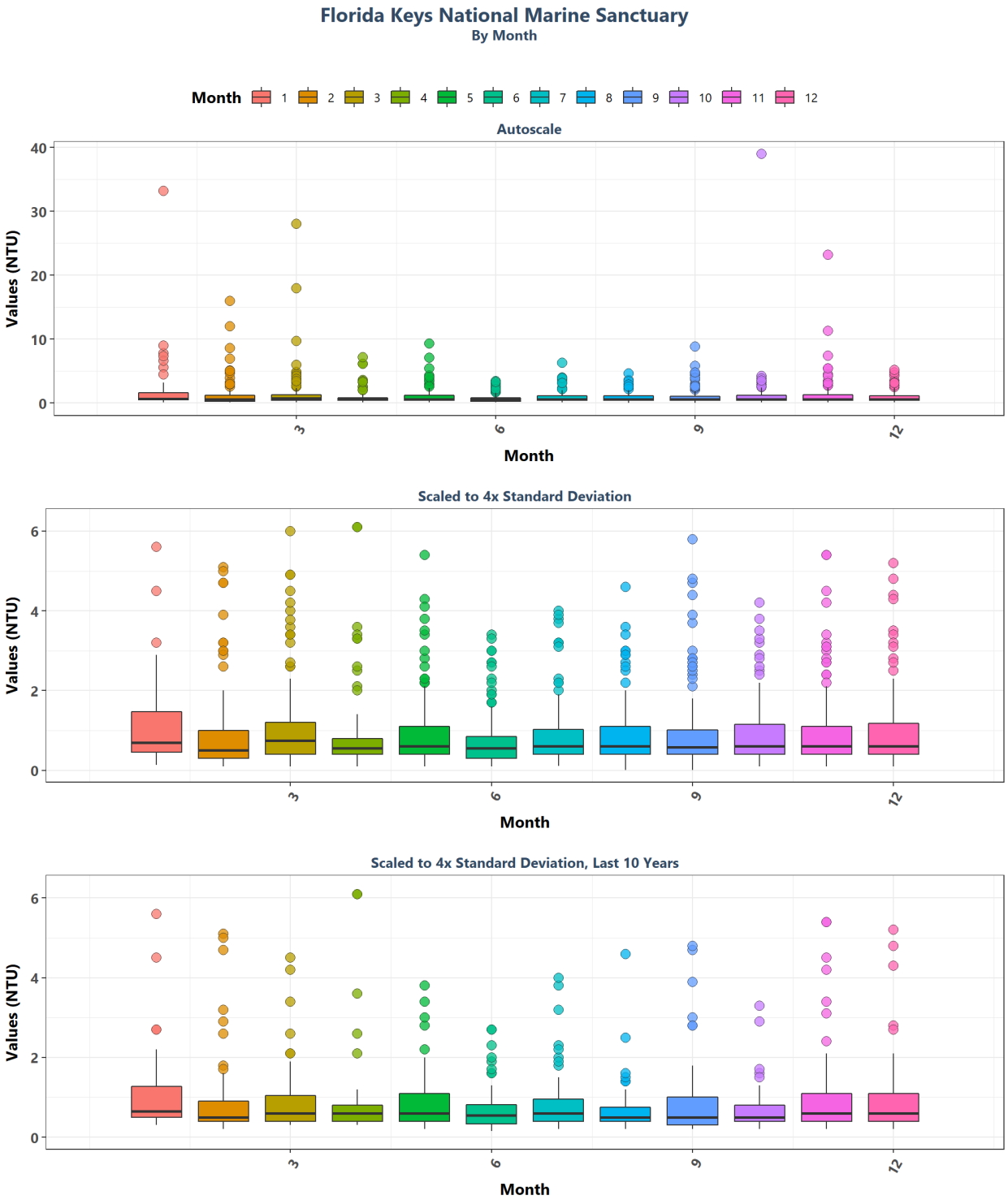
By Year





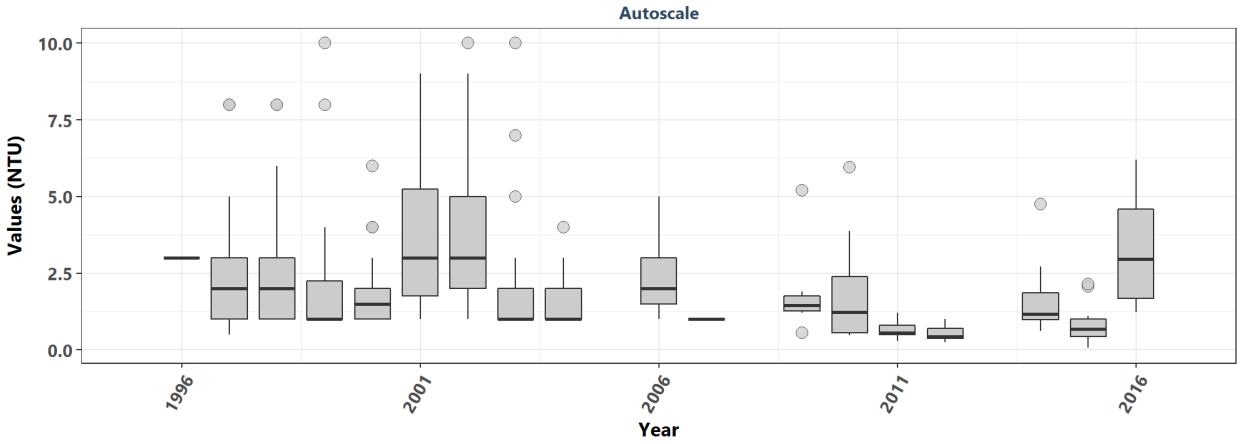


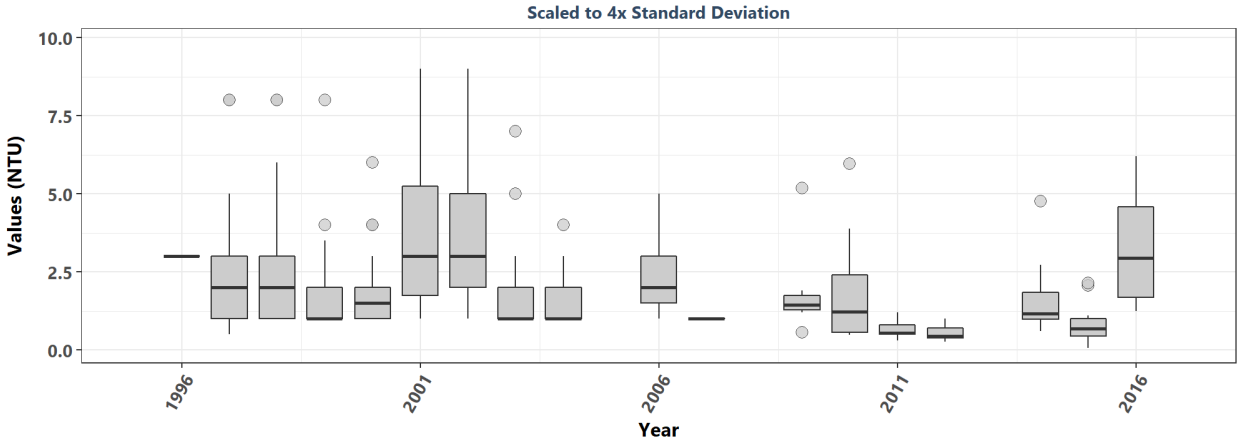

Florida Keys National Marine Sanctuary
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

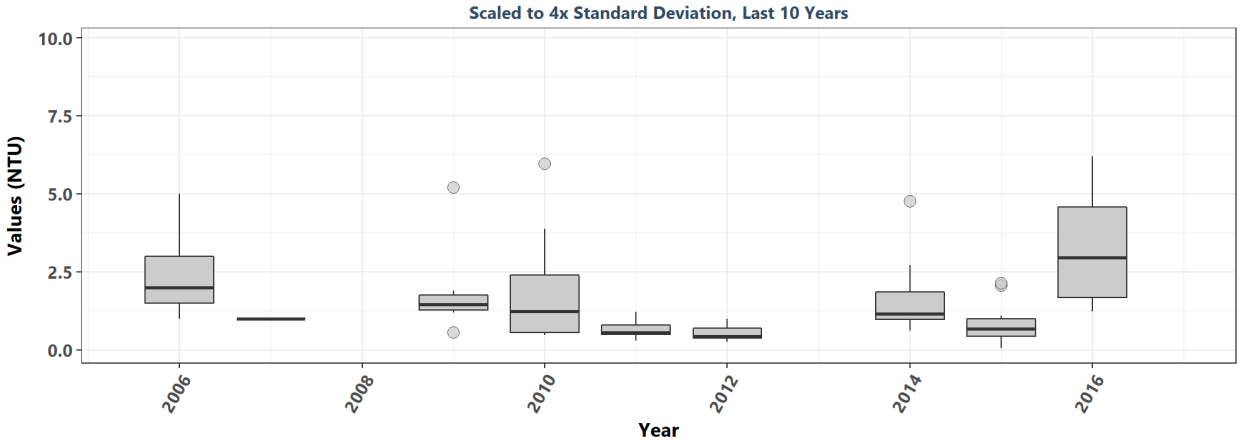

Florida Keys National Marine Sanctuary
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

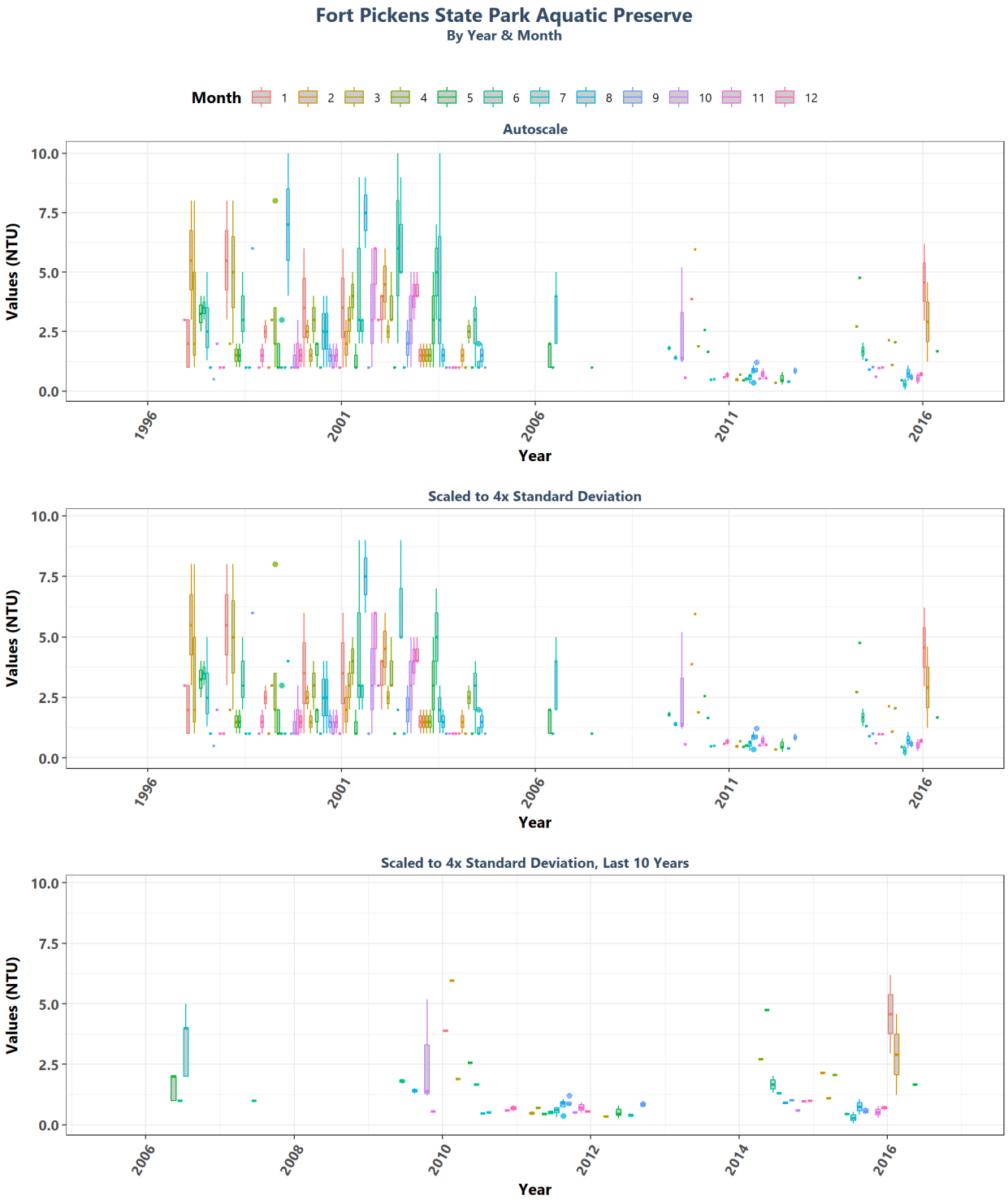
Fort Pickens State Park Aquatic Preserve

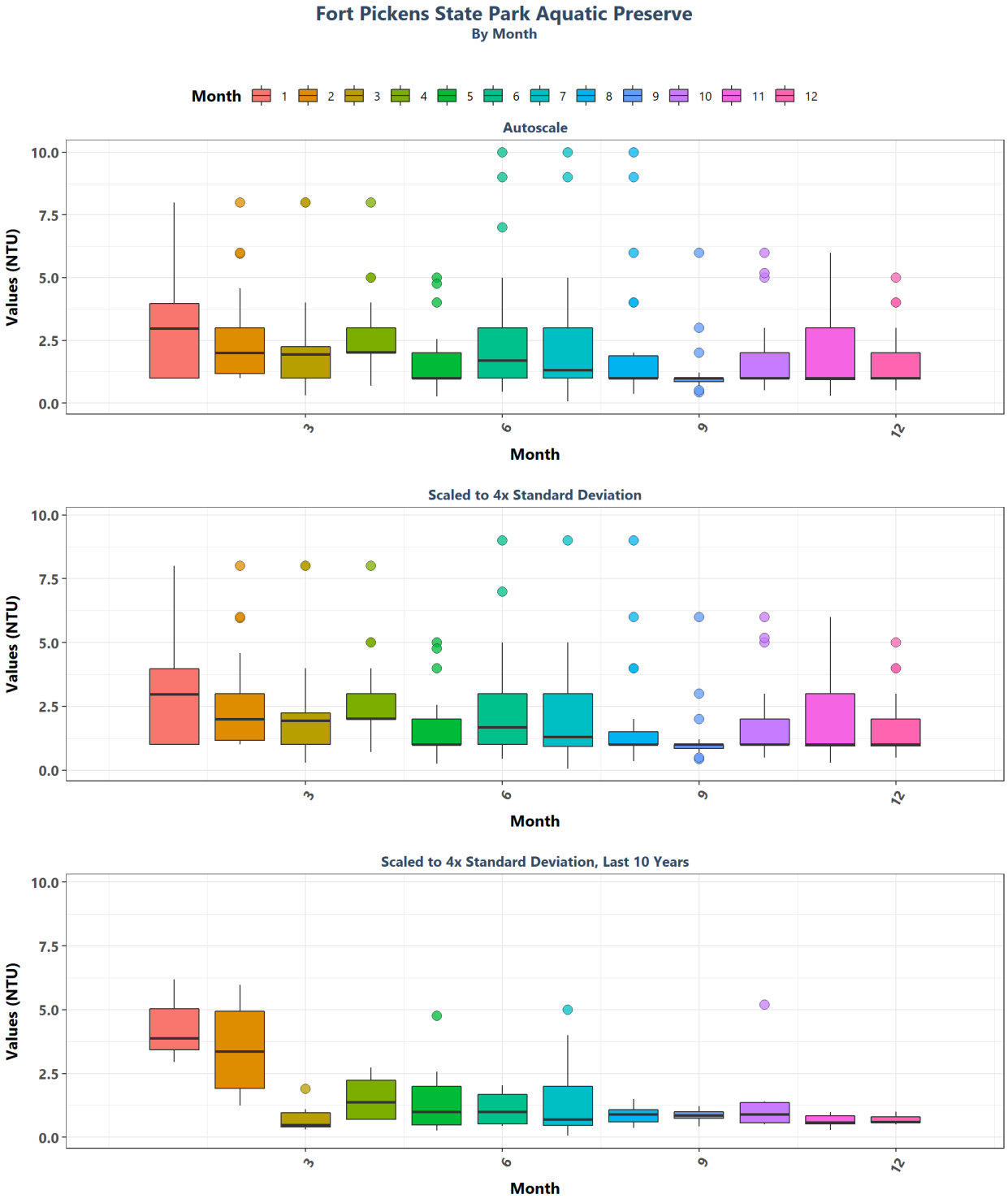
By Year





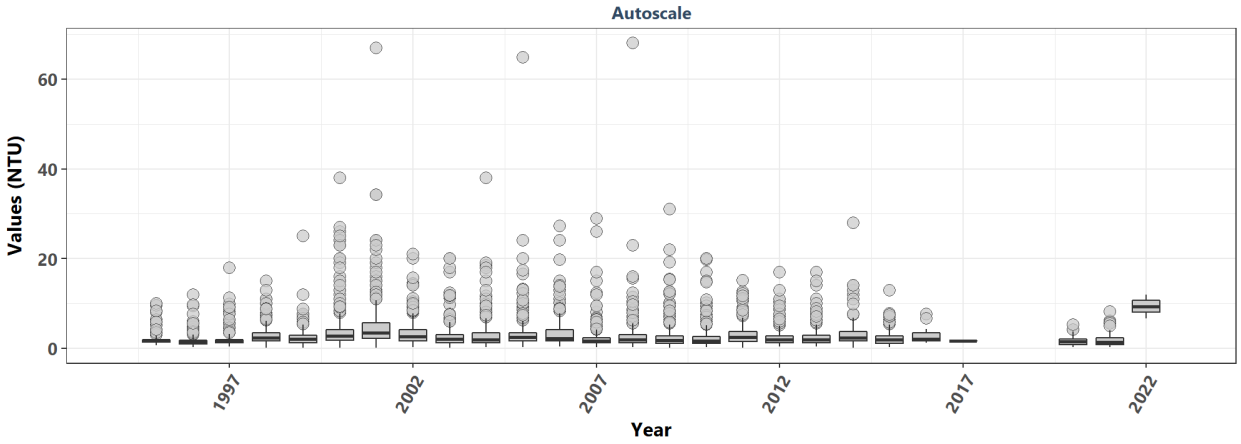


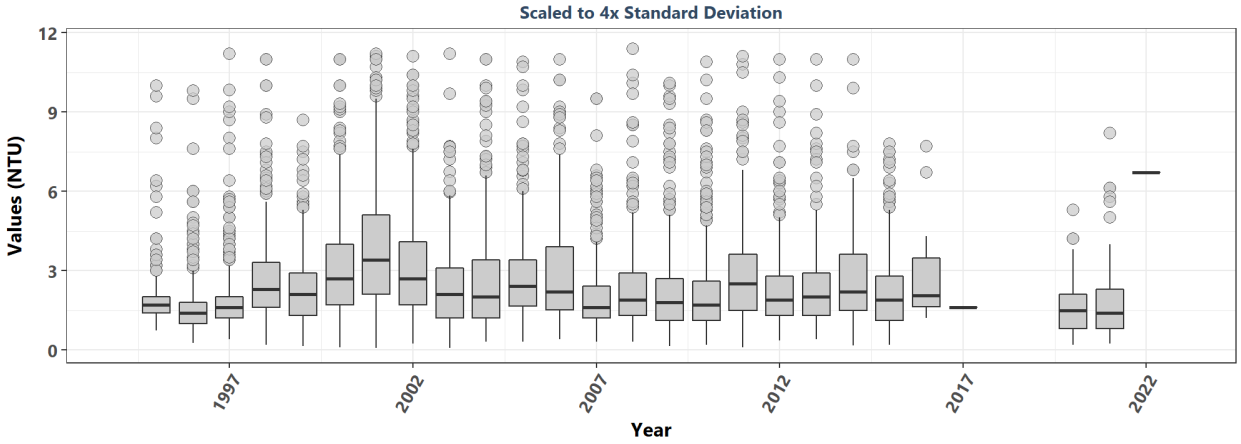

Fort Pickens State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
10.0
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
Year
Scaled to 4x Standard Deviation
Values (NTU)
10.0
7.5
5.0
2.5
0.0
1996
2001
2006
2011
2016
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
10.0
7.5
5.0
2.5
0.0
2006
2008
2010
2012
2014
2016
Year

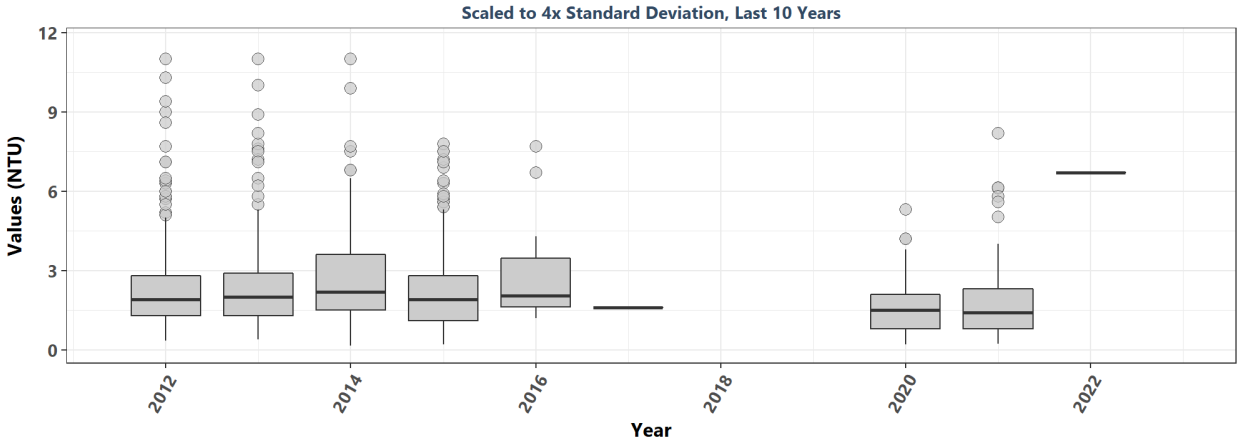

Fort Pickens State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month

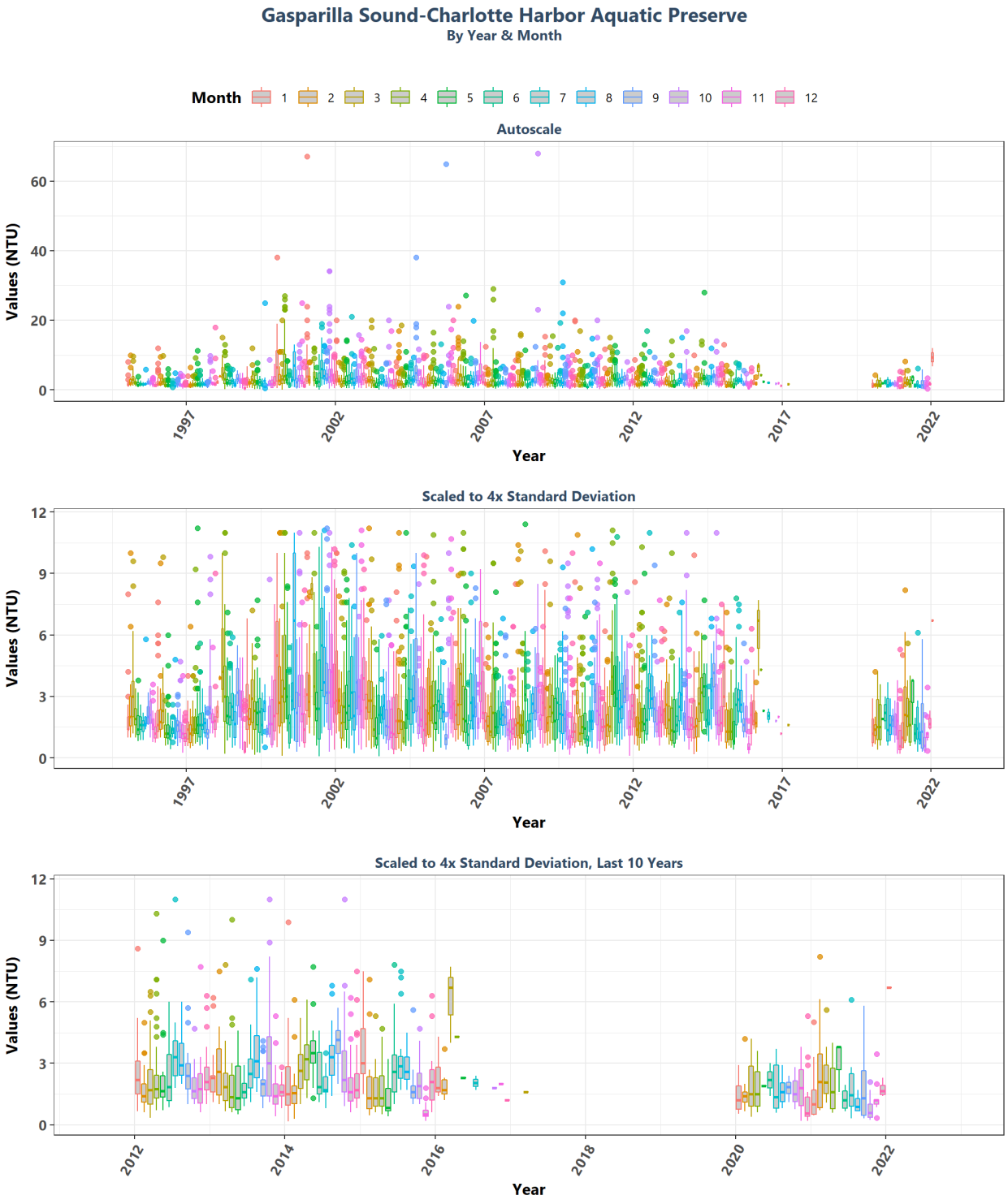
Gasparilla Sound-Charlotte Harbor Aquatic Preserve

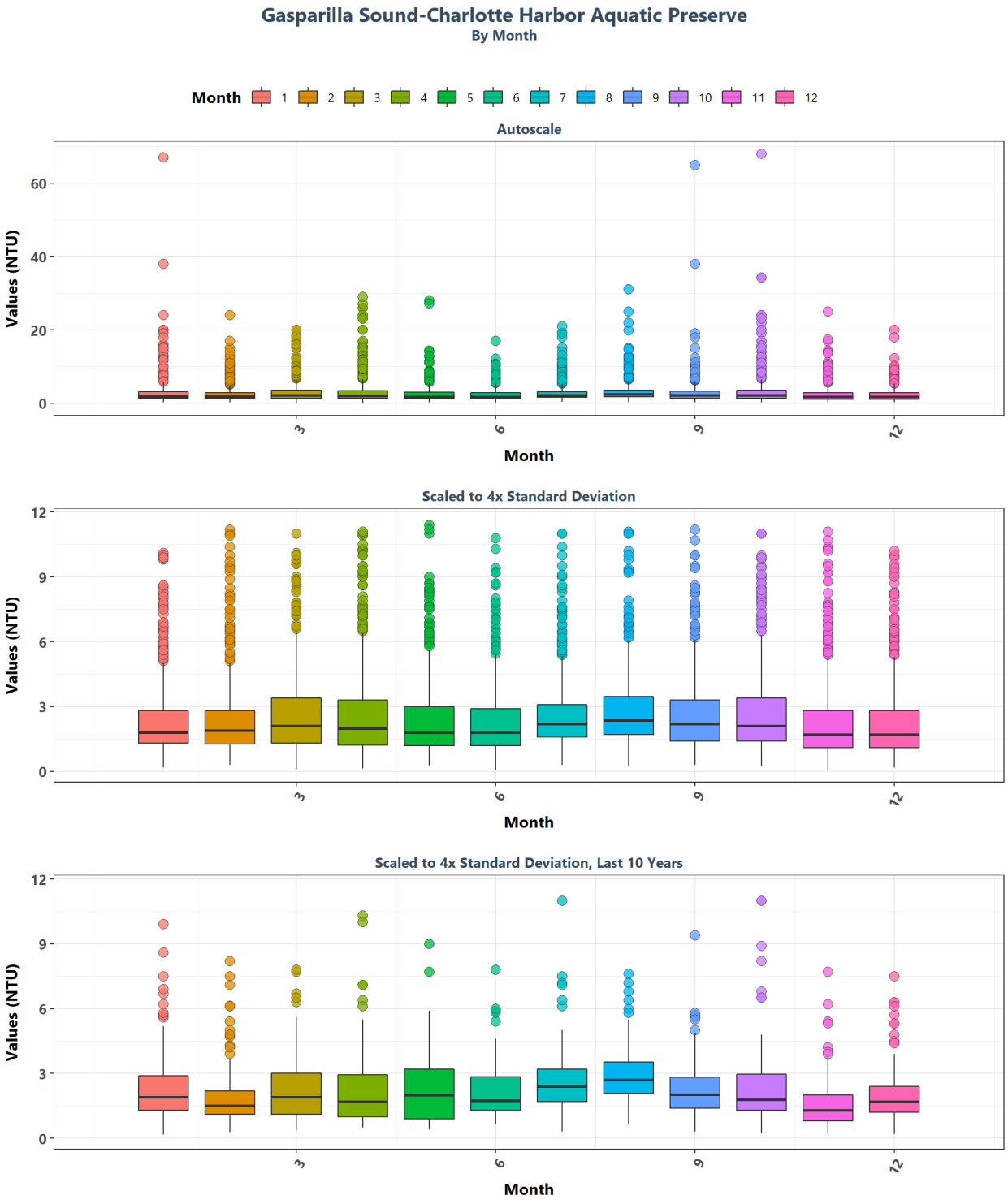
By Year



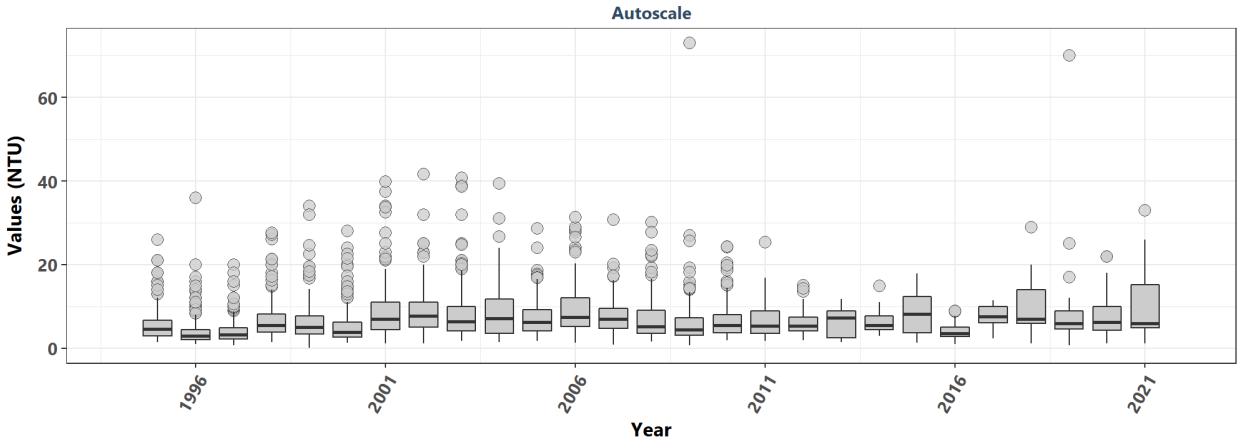


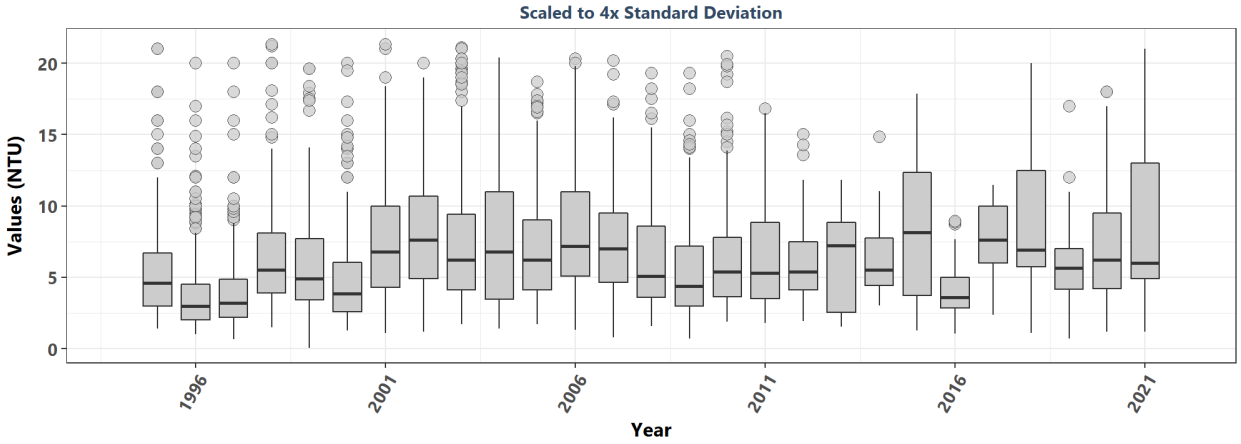


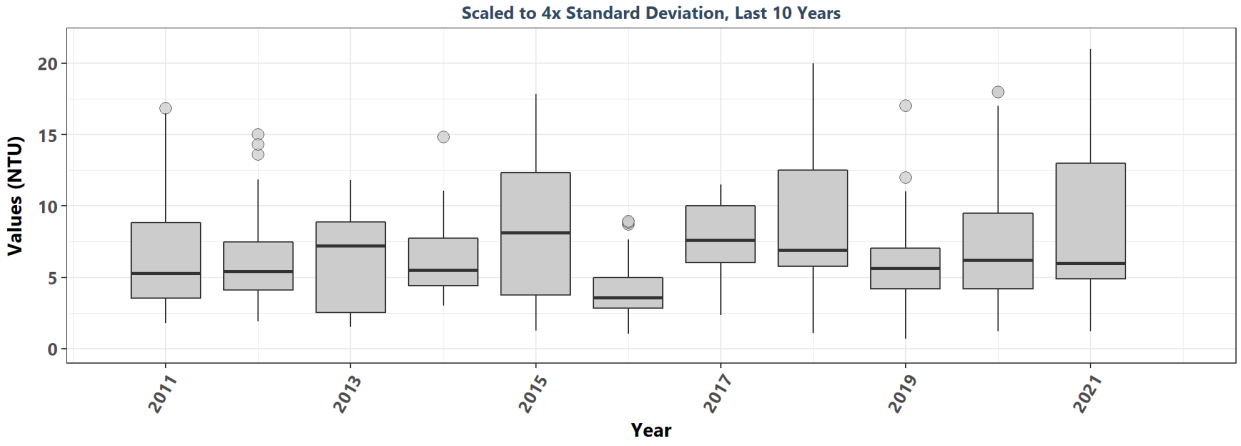

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0 20 40 60
1997 2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation
Values (NTU)
0 3 6 9 12
1997 2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0 3 6 9 12
2012 2014 2016 2018 2020 2022
Year

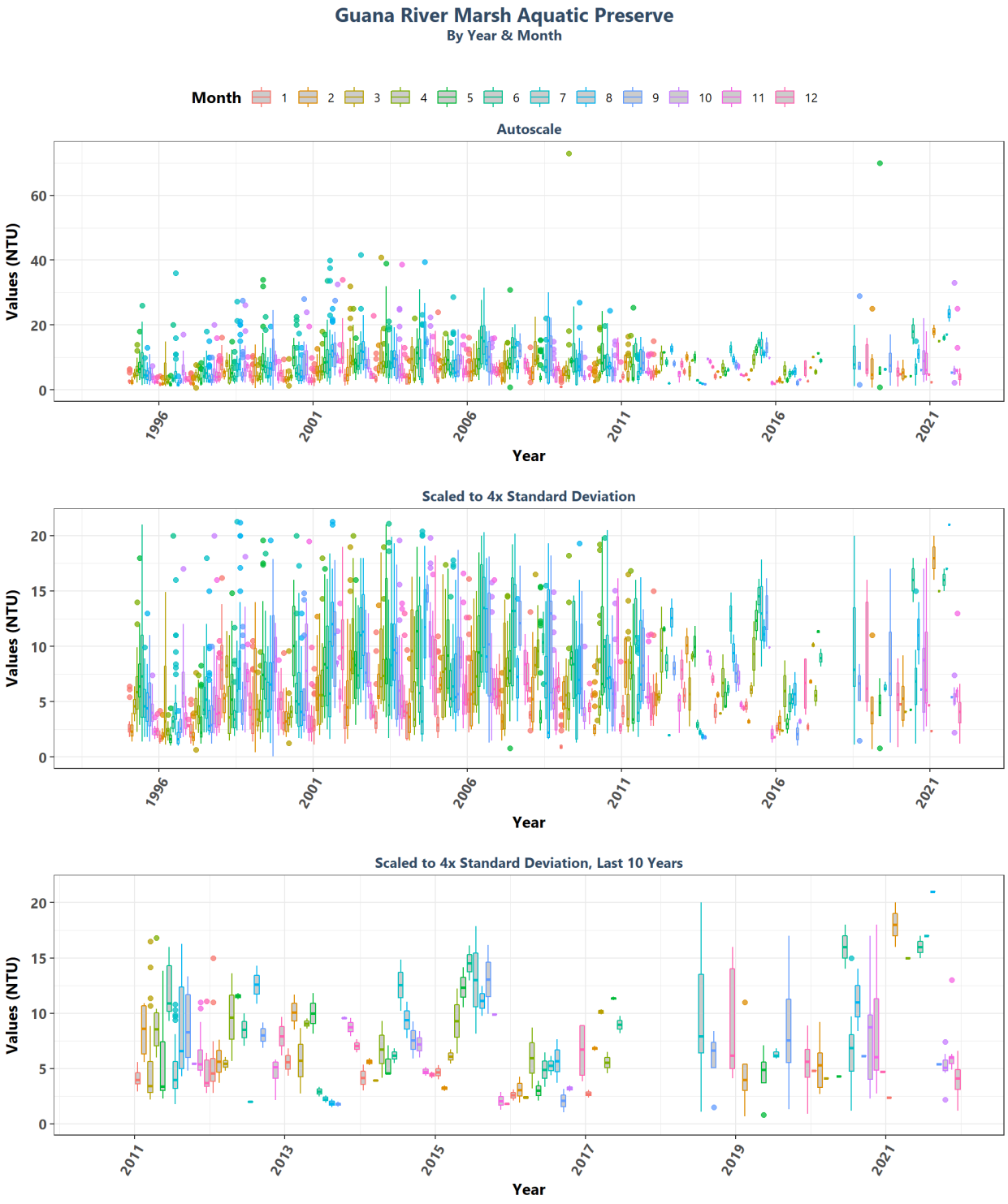

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
20
40
60
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
0
3
6
9
12
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
3
6
9
12
3
6
9
12
Month

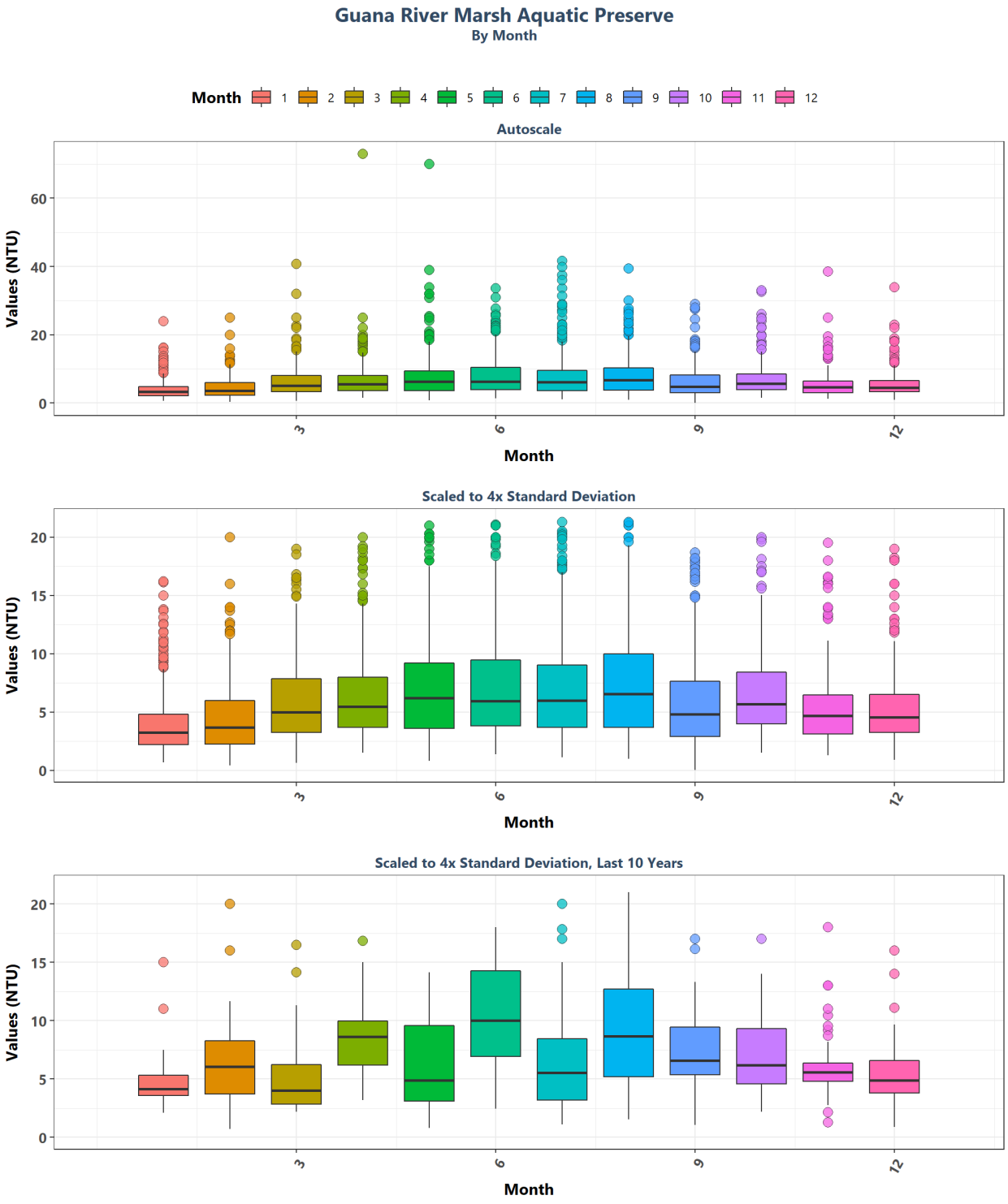
Guana River Marsh Aquatic Preserve
By Year





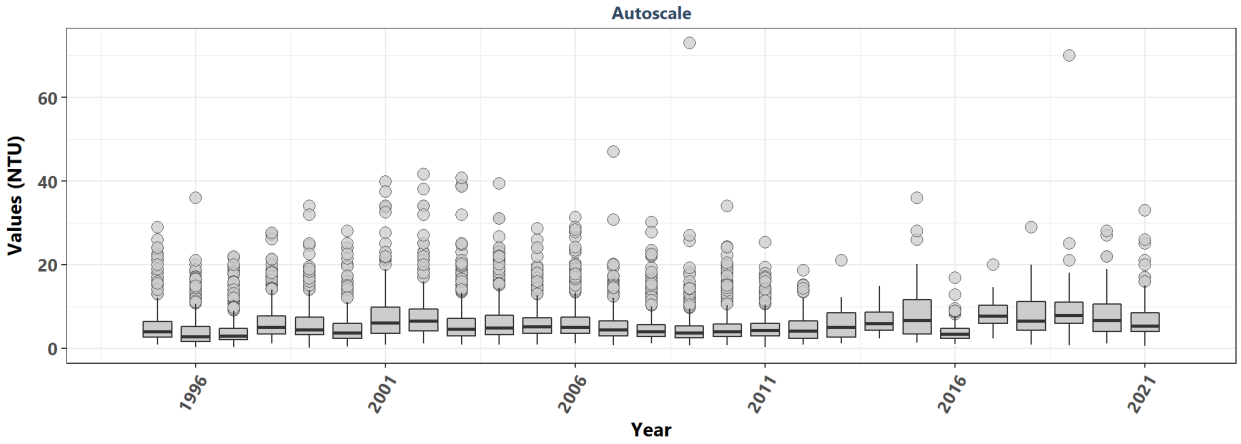


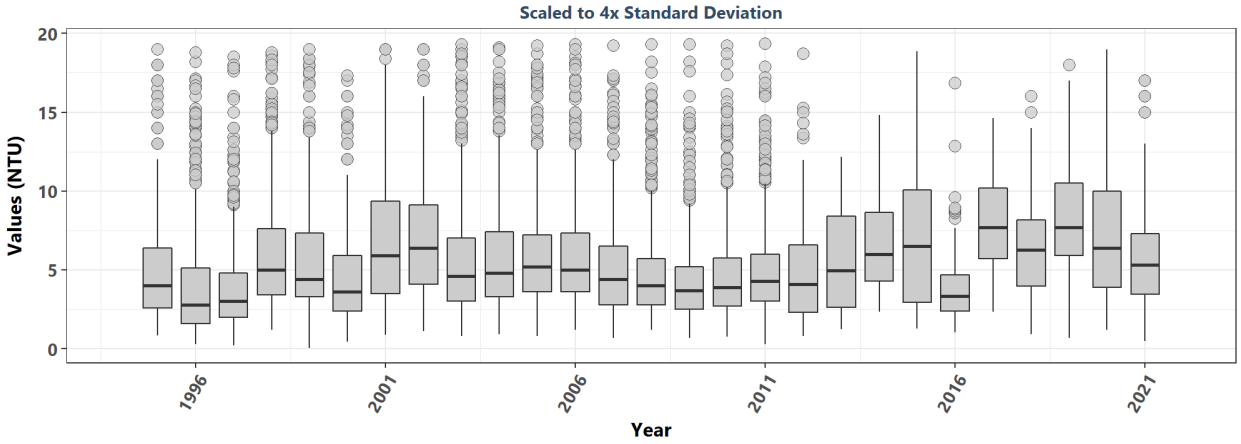

Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0 20 40 60
1996 2001 2006 2011 2016 2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
0 5 10 15 20
1996 2001 2006 2011 2016 2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0 5 10 15 20
2011 2013 2015 2017 2019 2021
Year

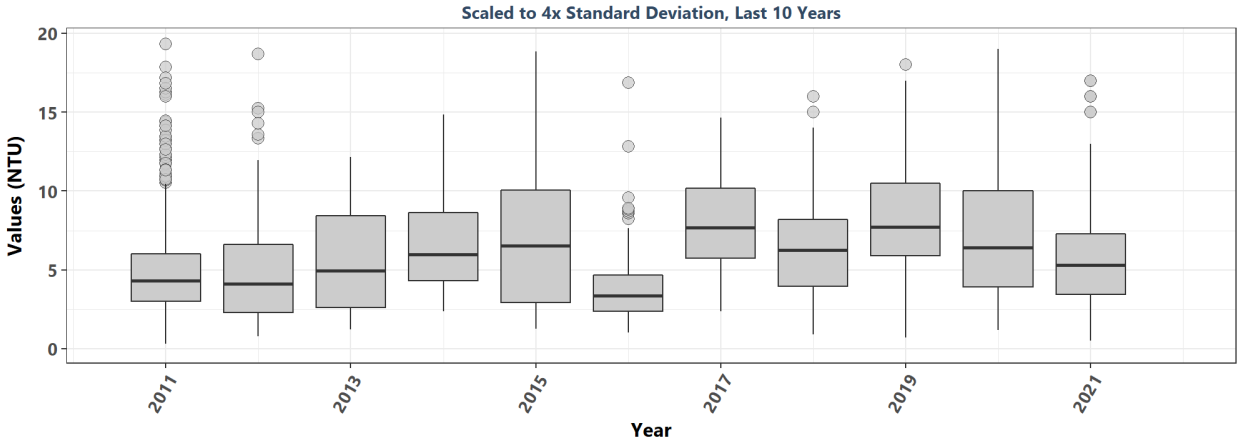

Guana River Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

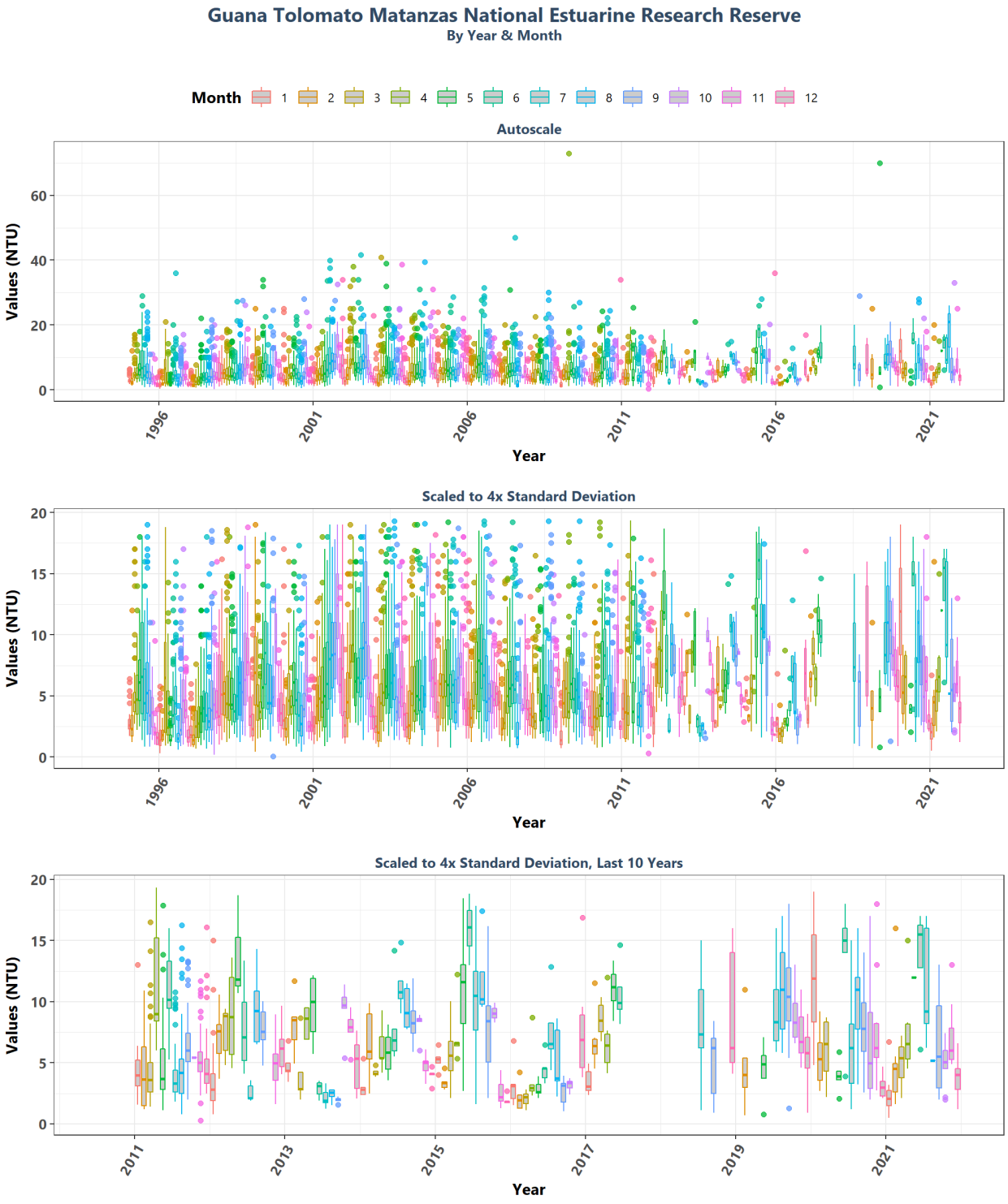
Guana Tolomato Matanzas National Estuarine Research Reserve

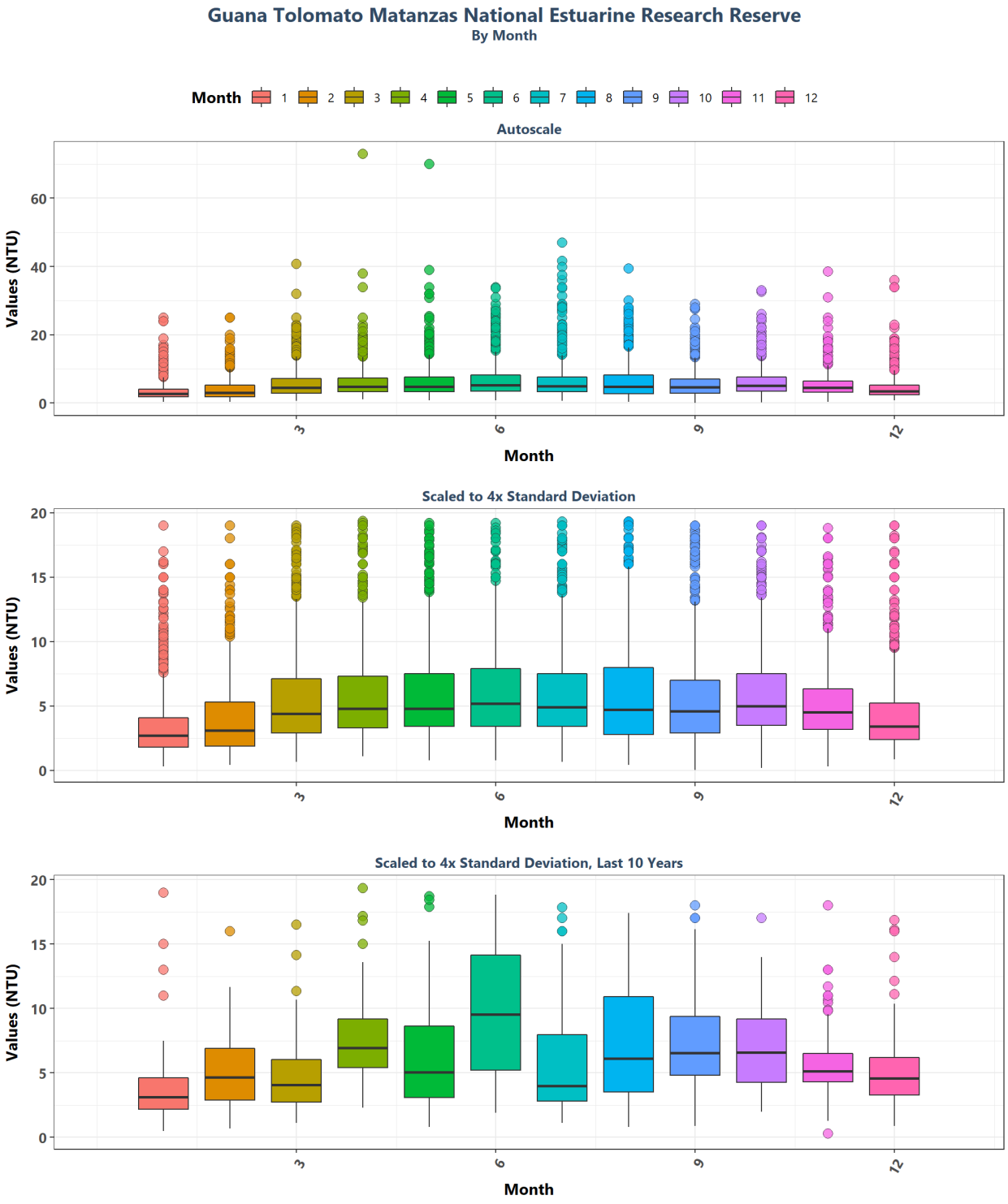
By Year

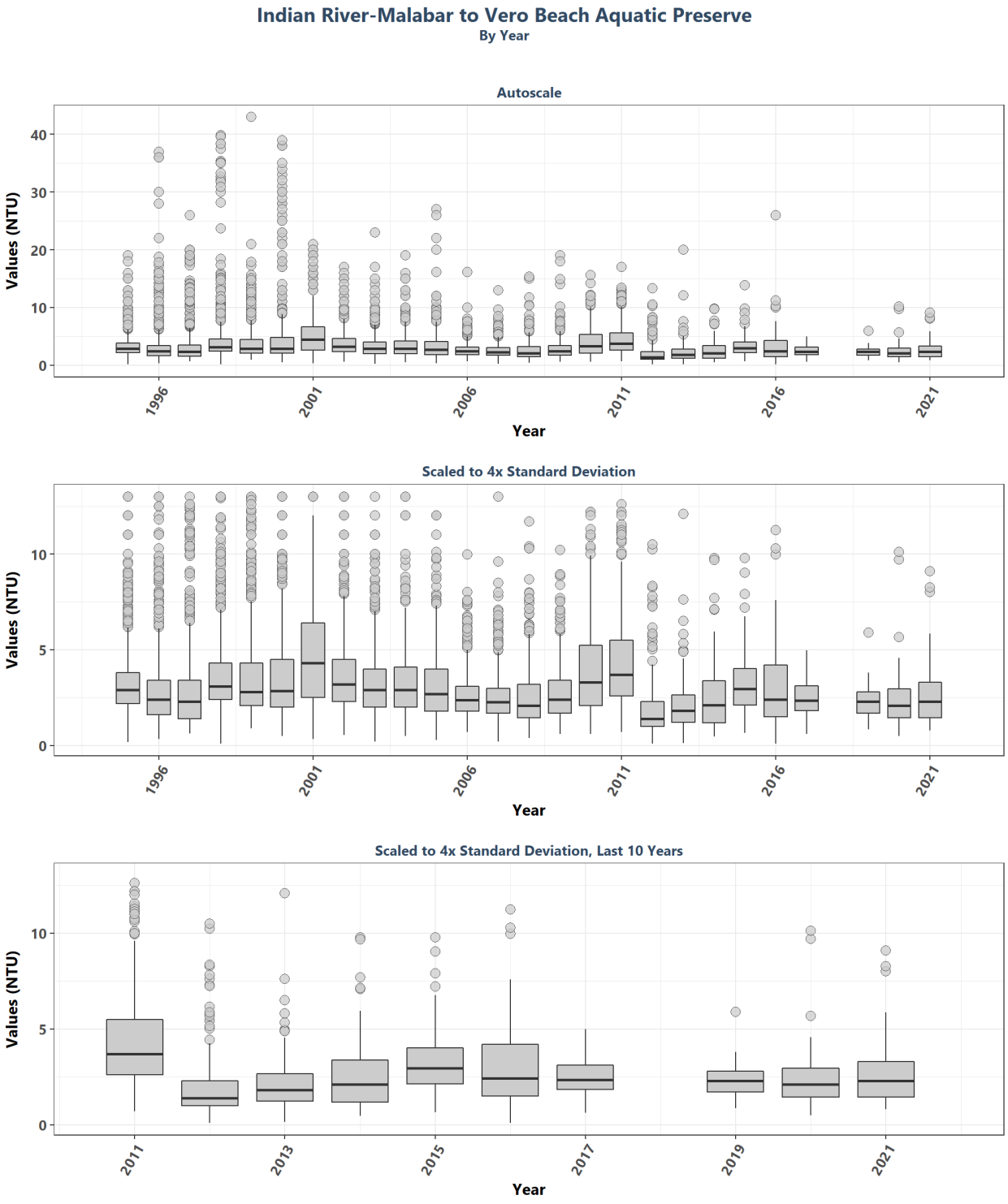


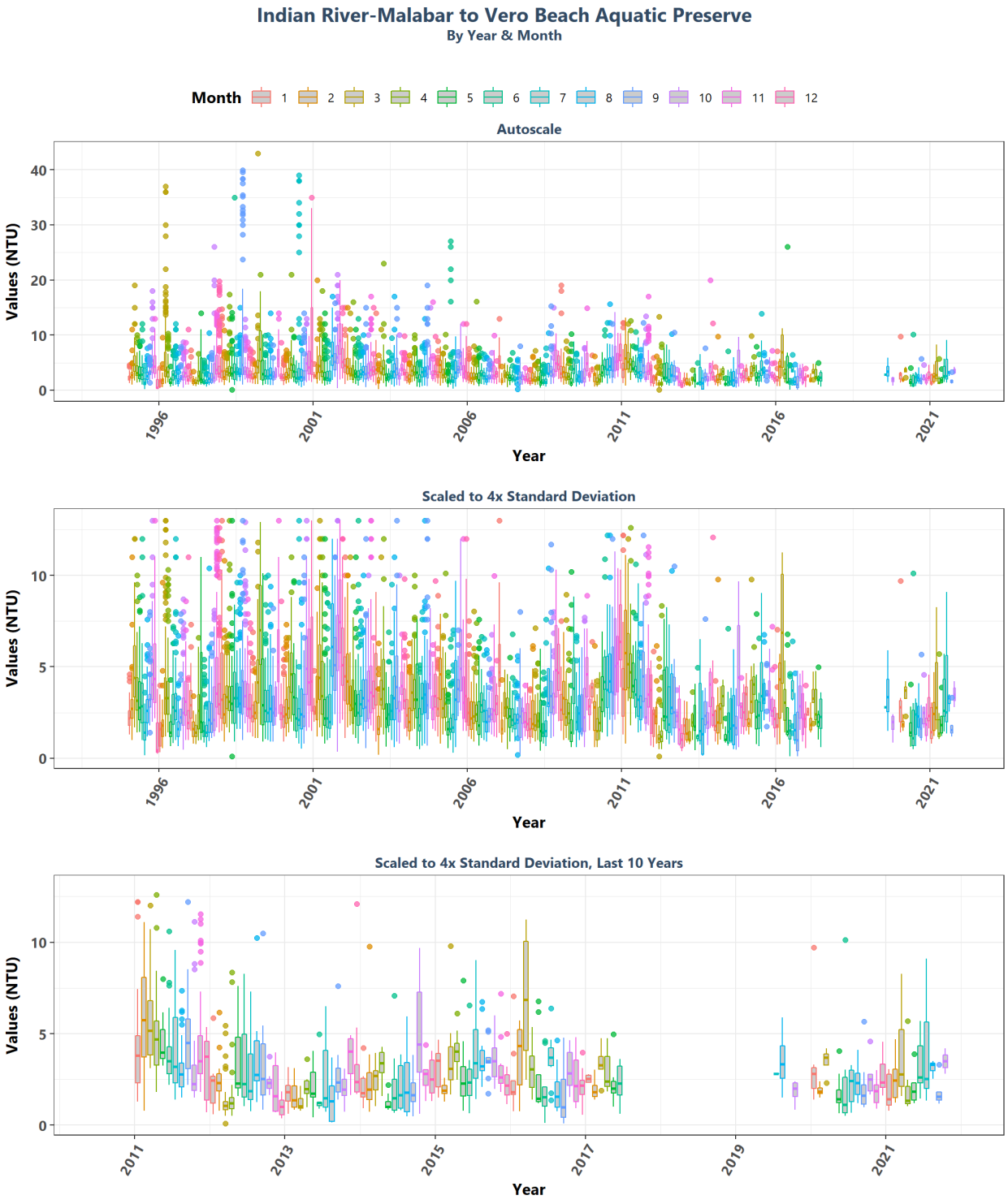


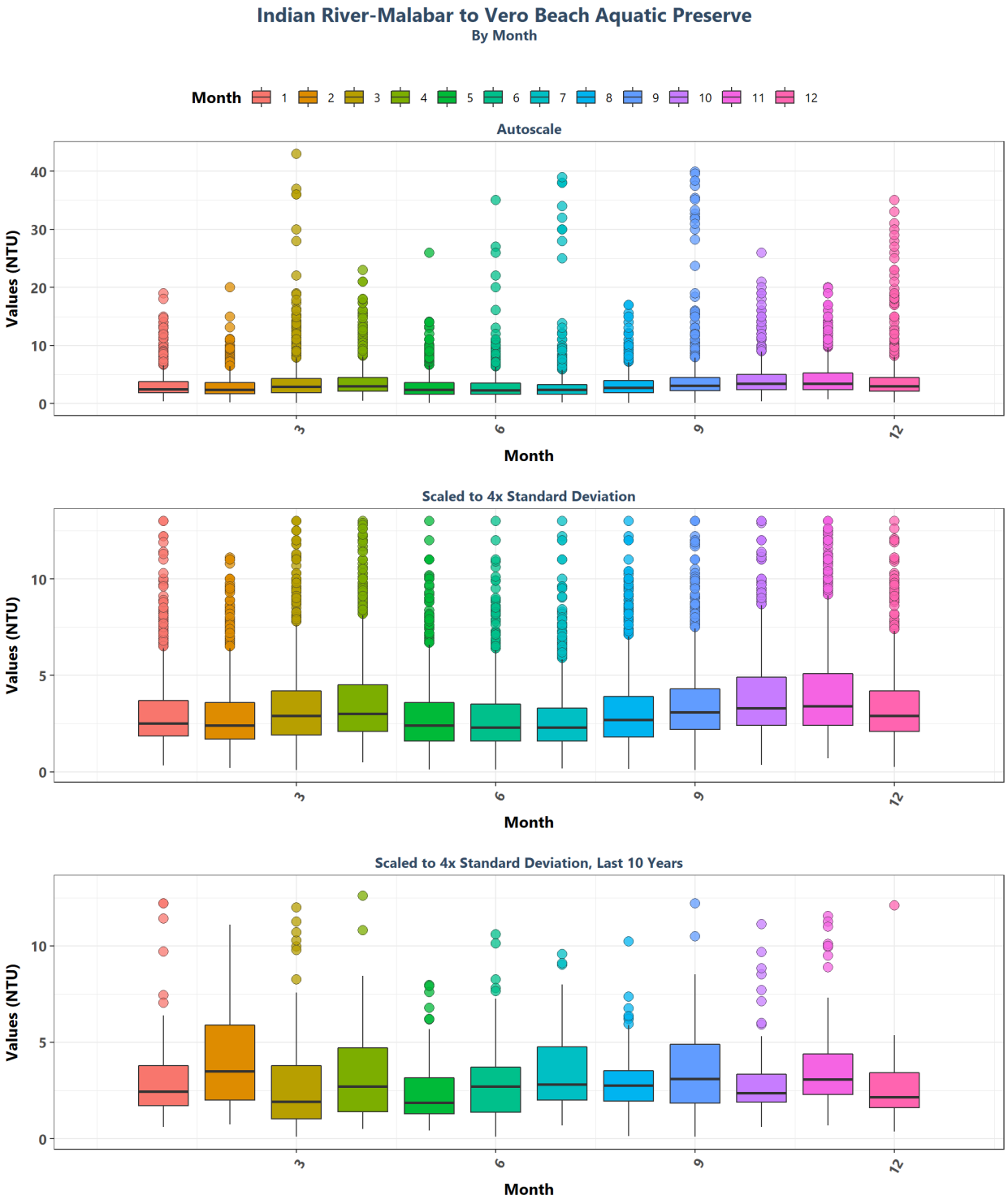


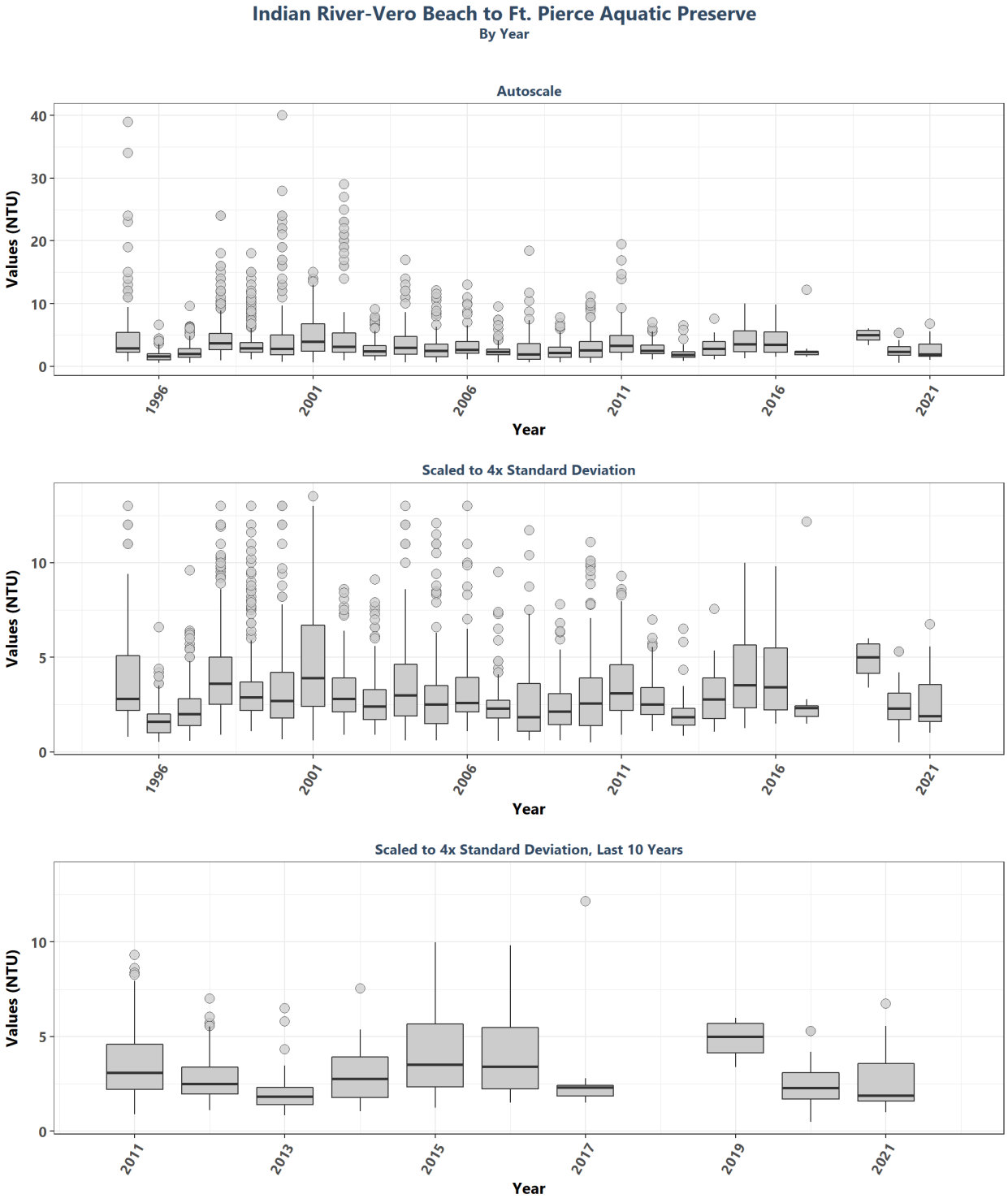

Guana Tolomato Matanzas National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

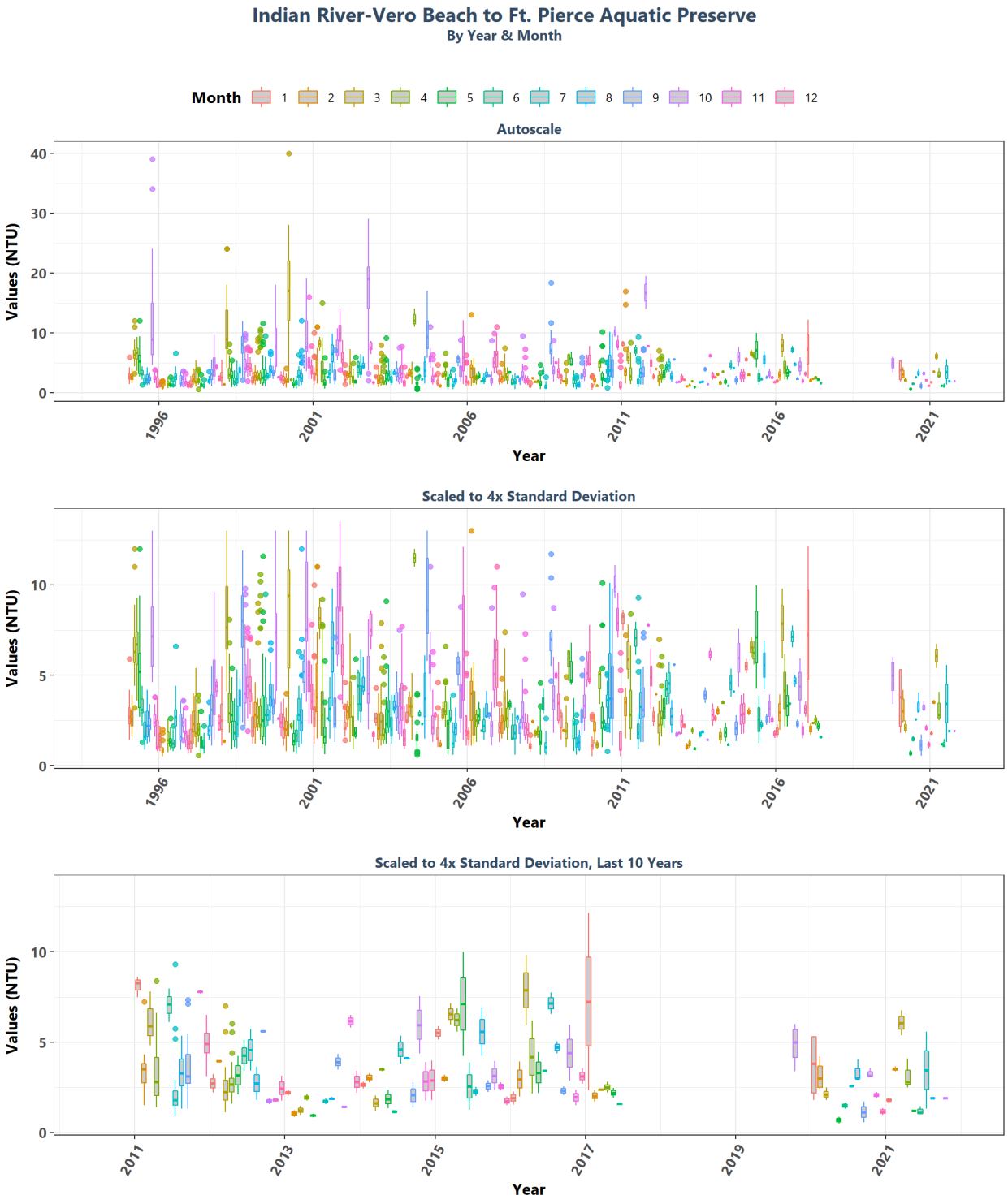

Guana Tolomato Matanzas National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

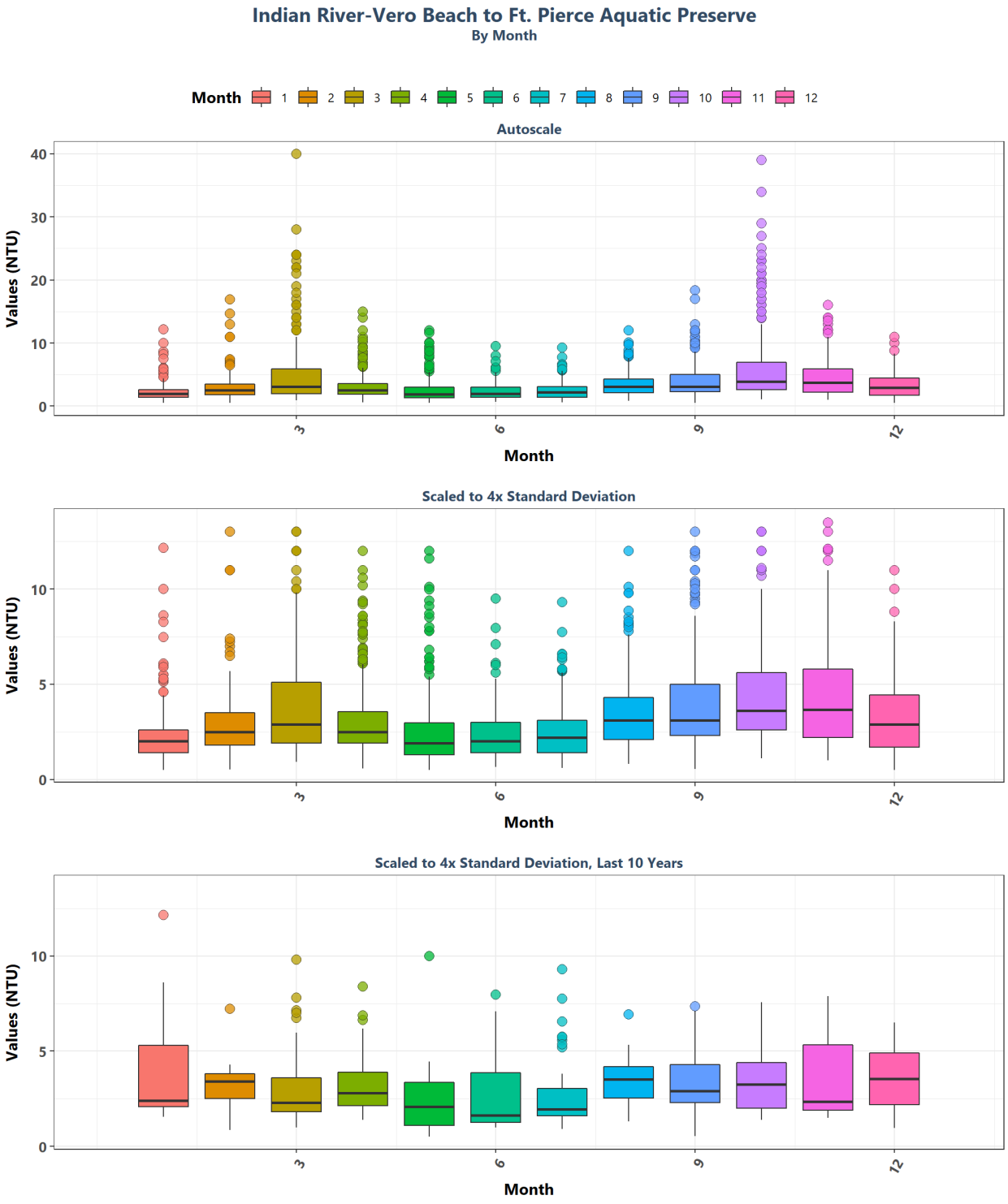

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

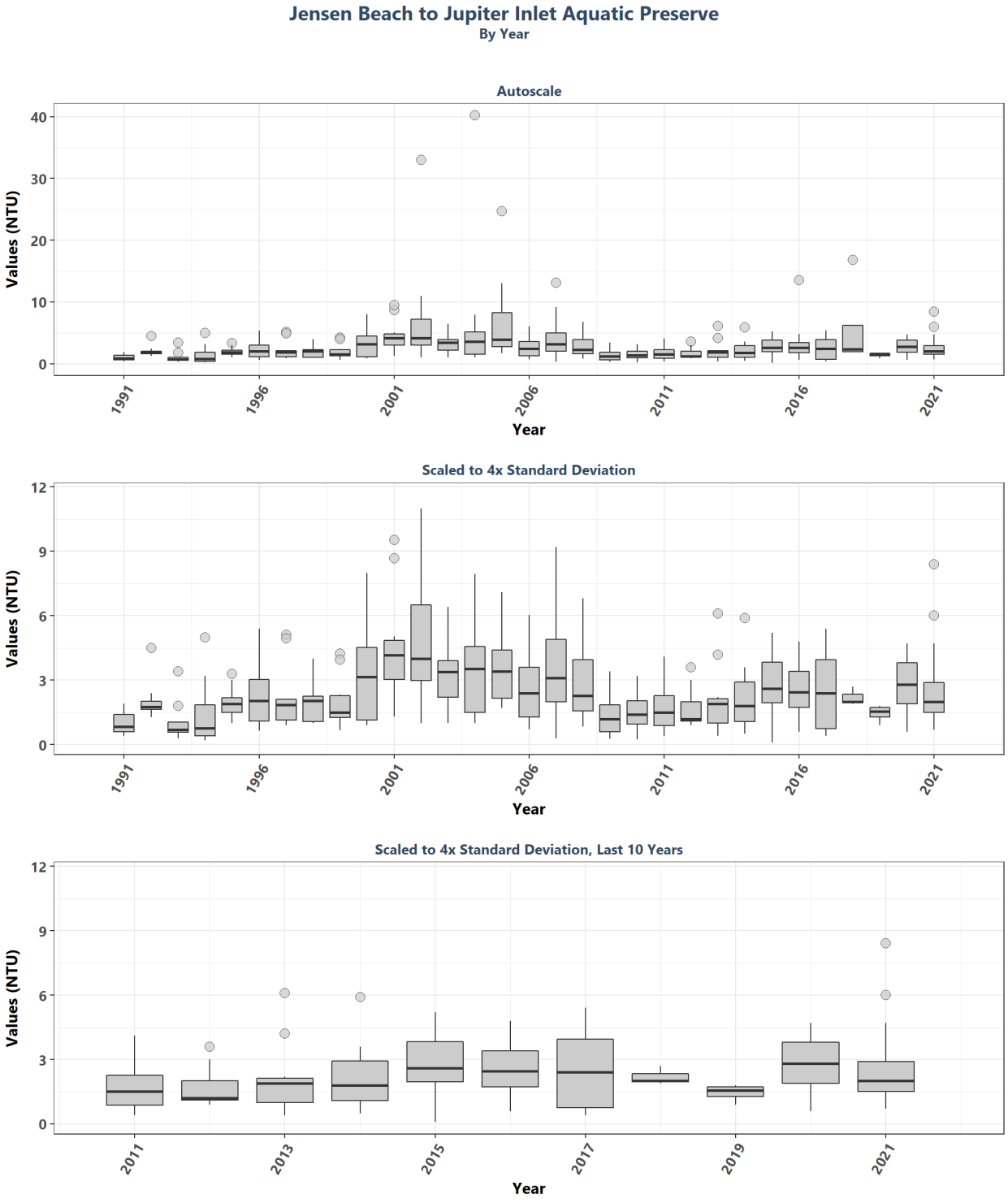

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

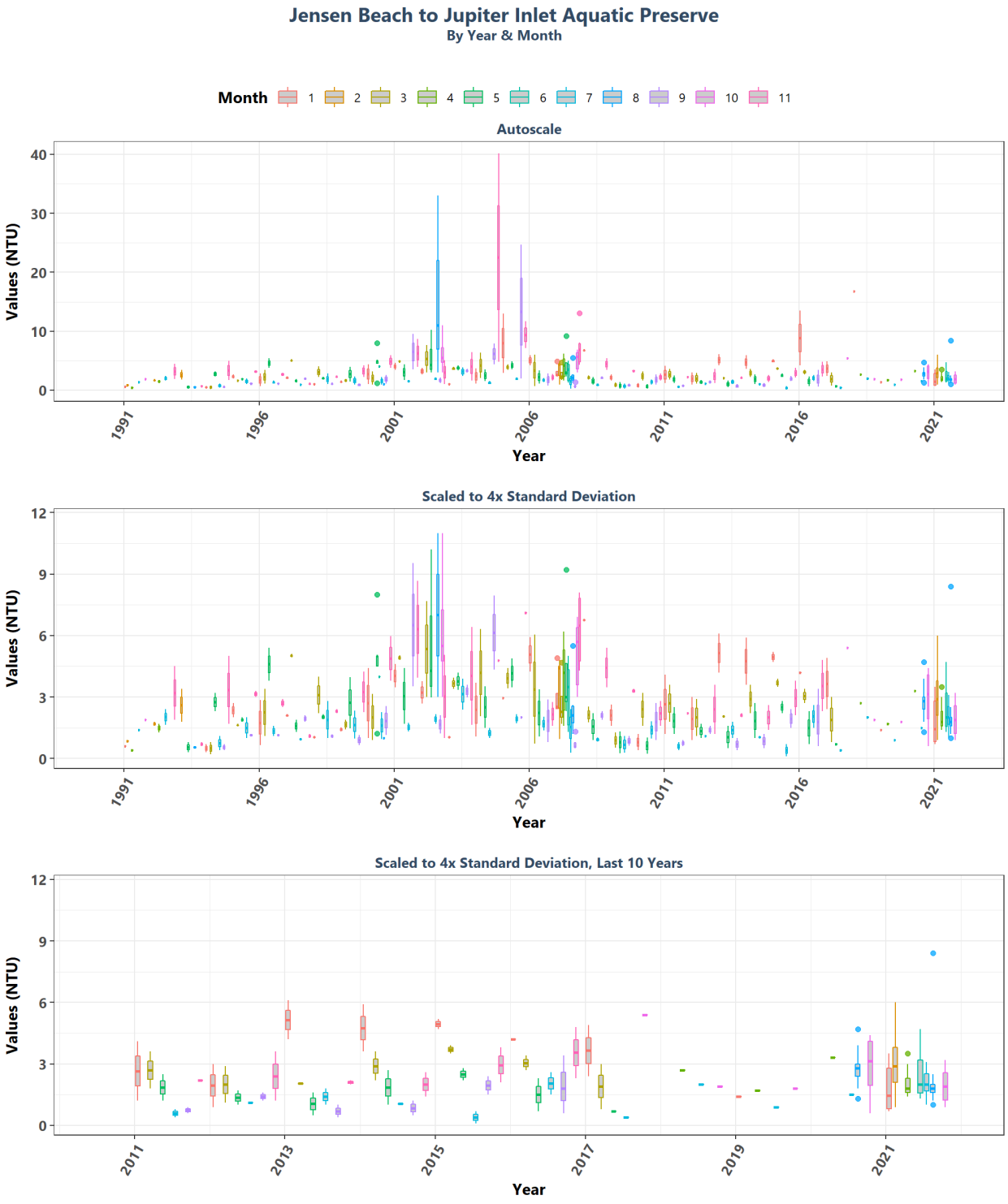

Indian River-Malabar to Vero Beach Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

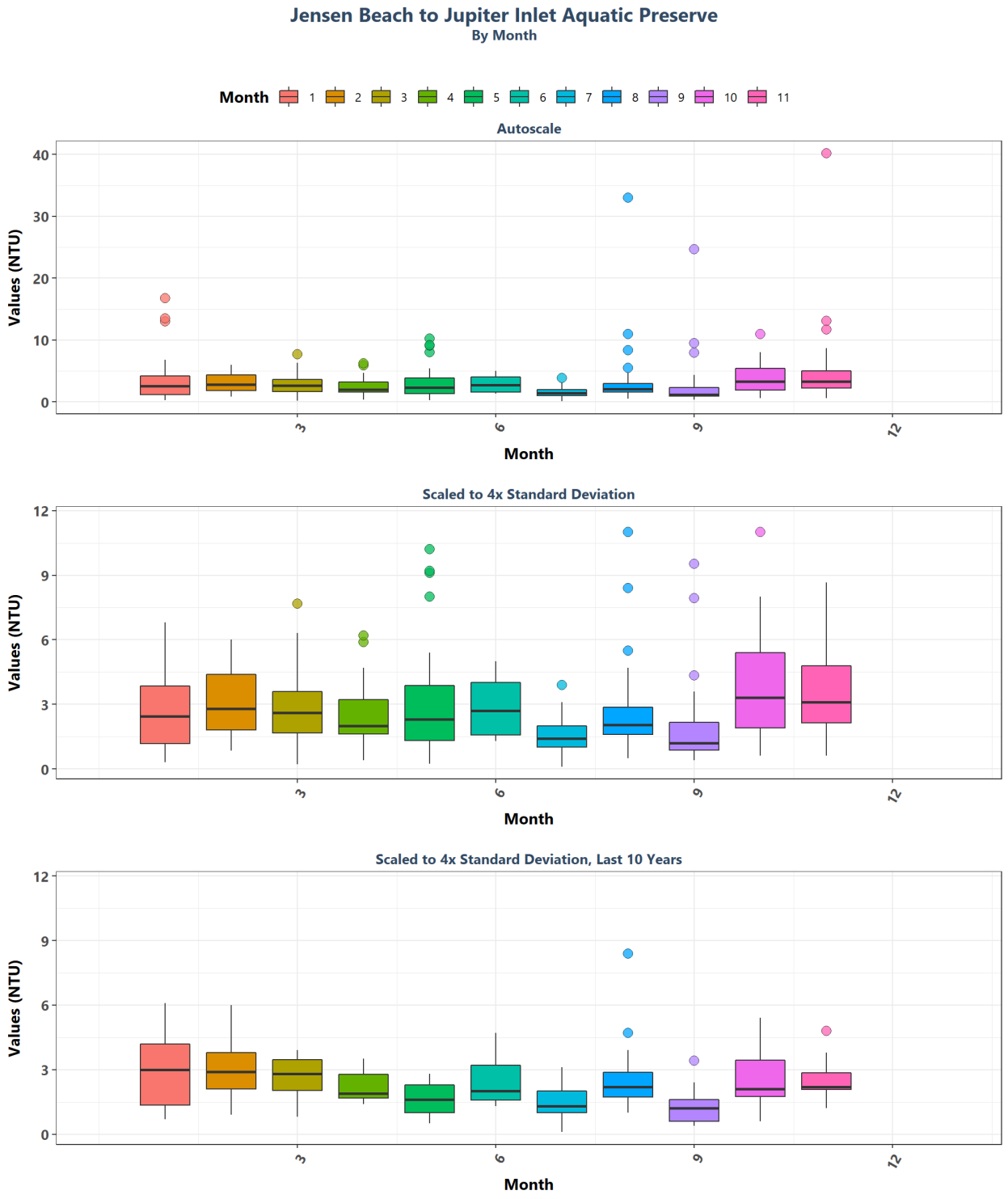

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

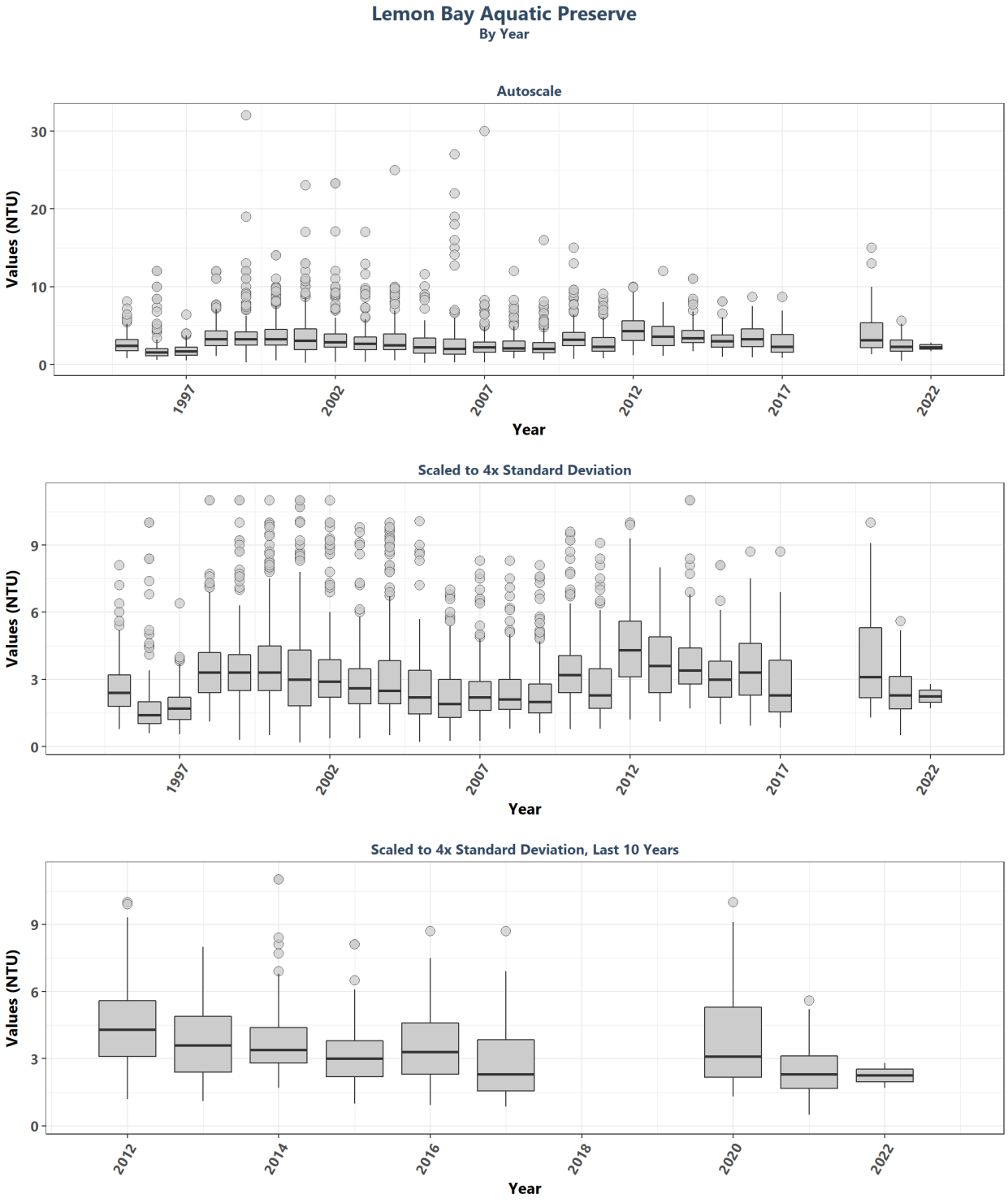

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

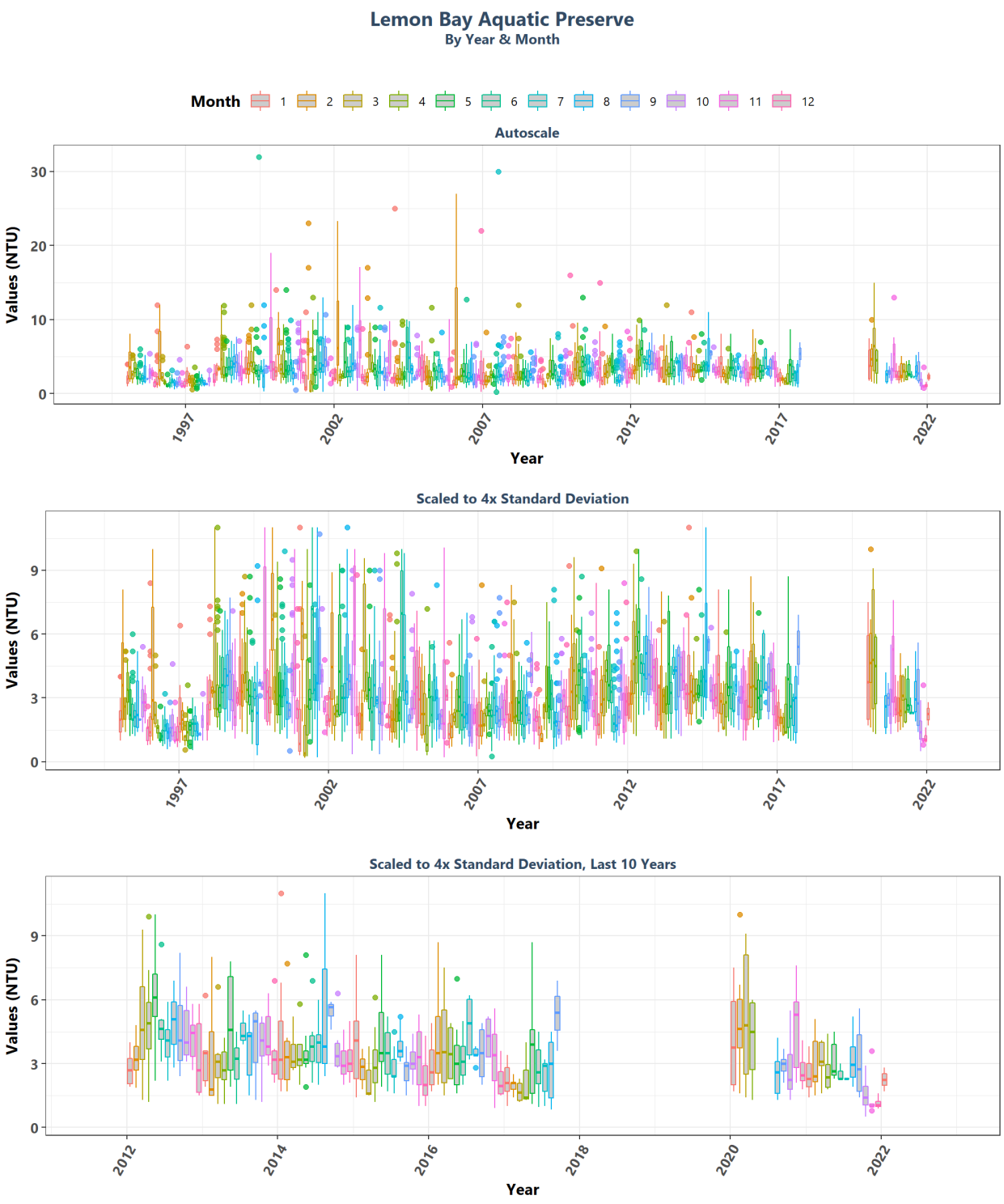

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

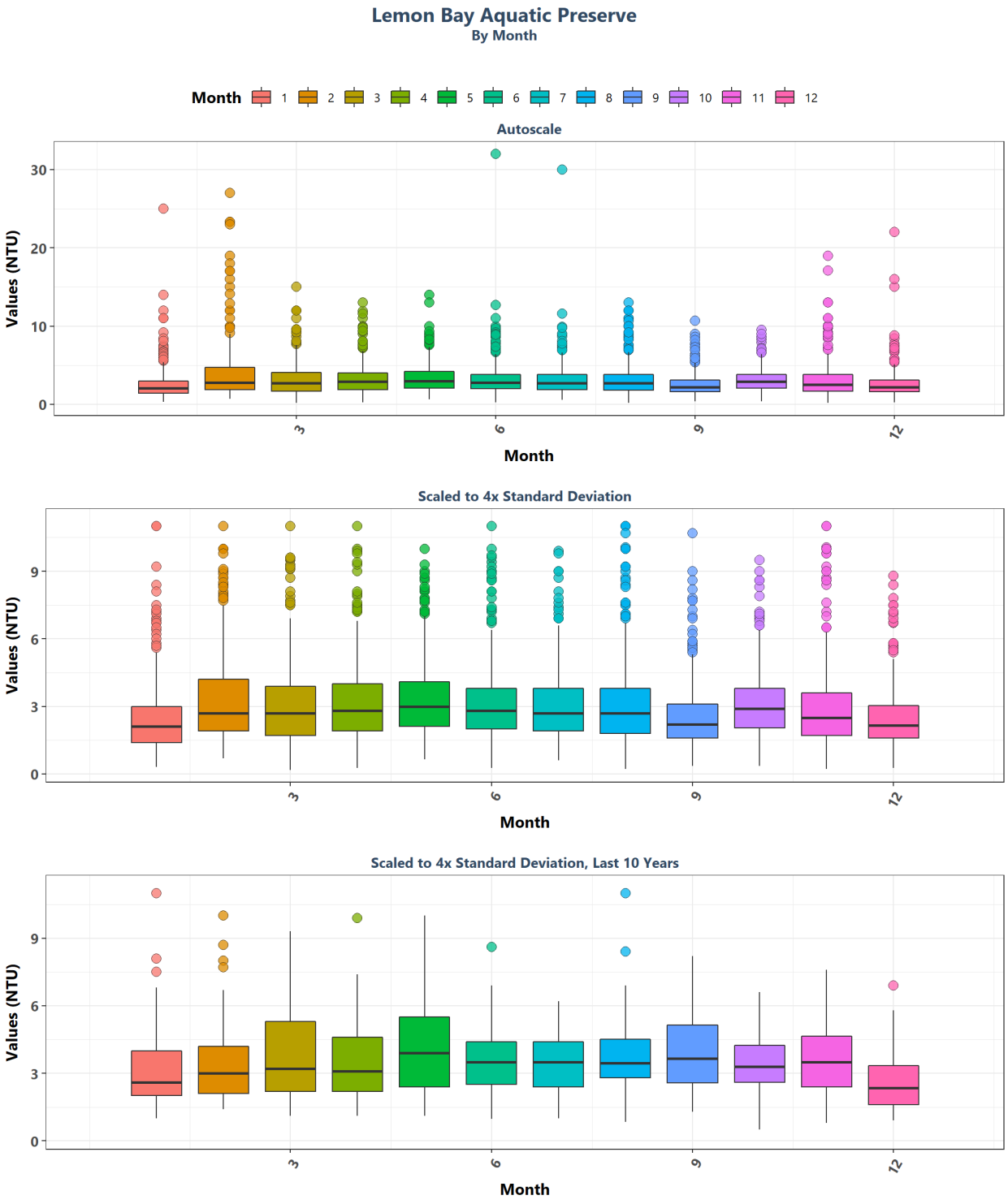

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

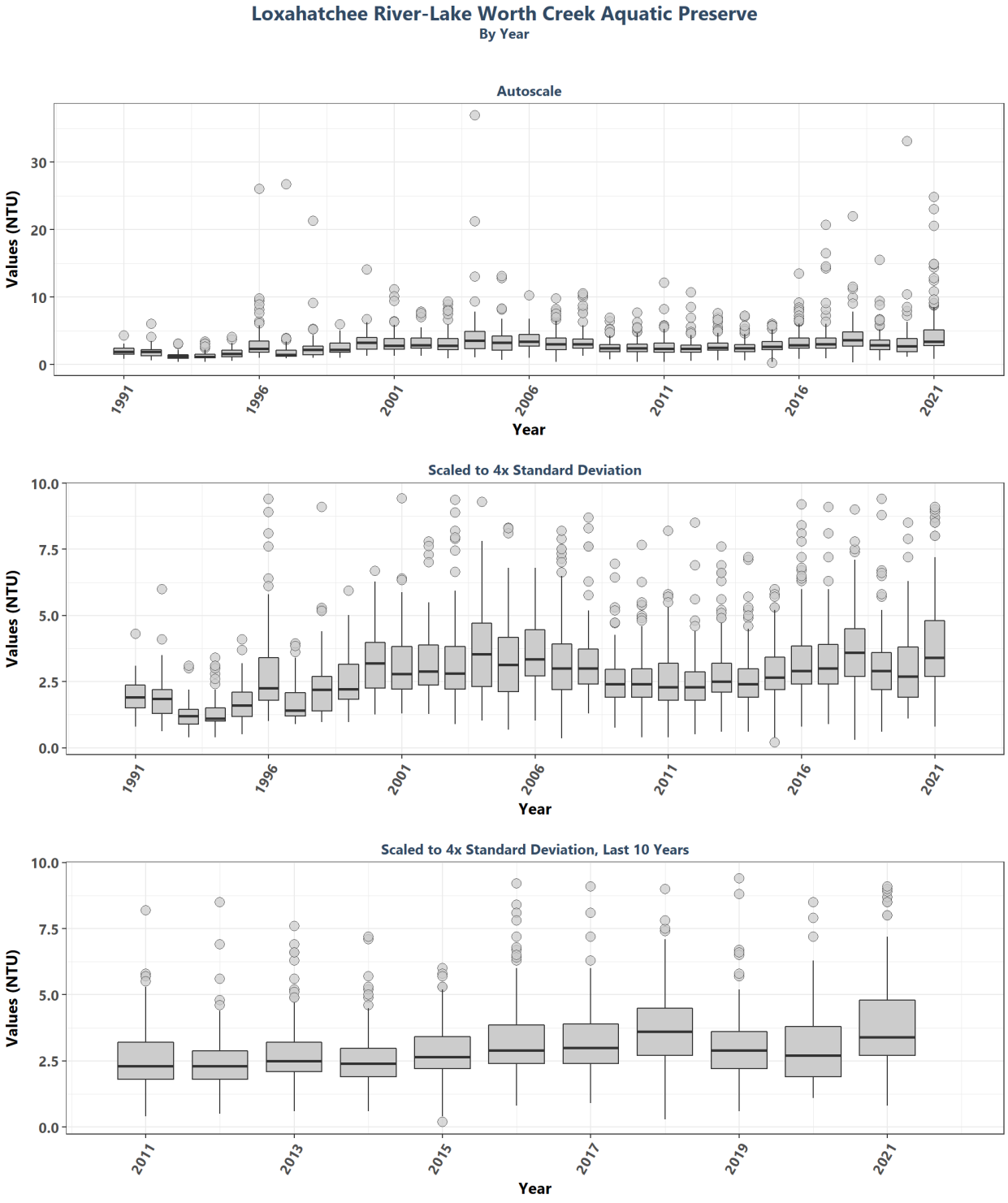

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

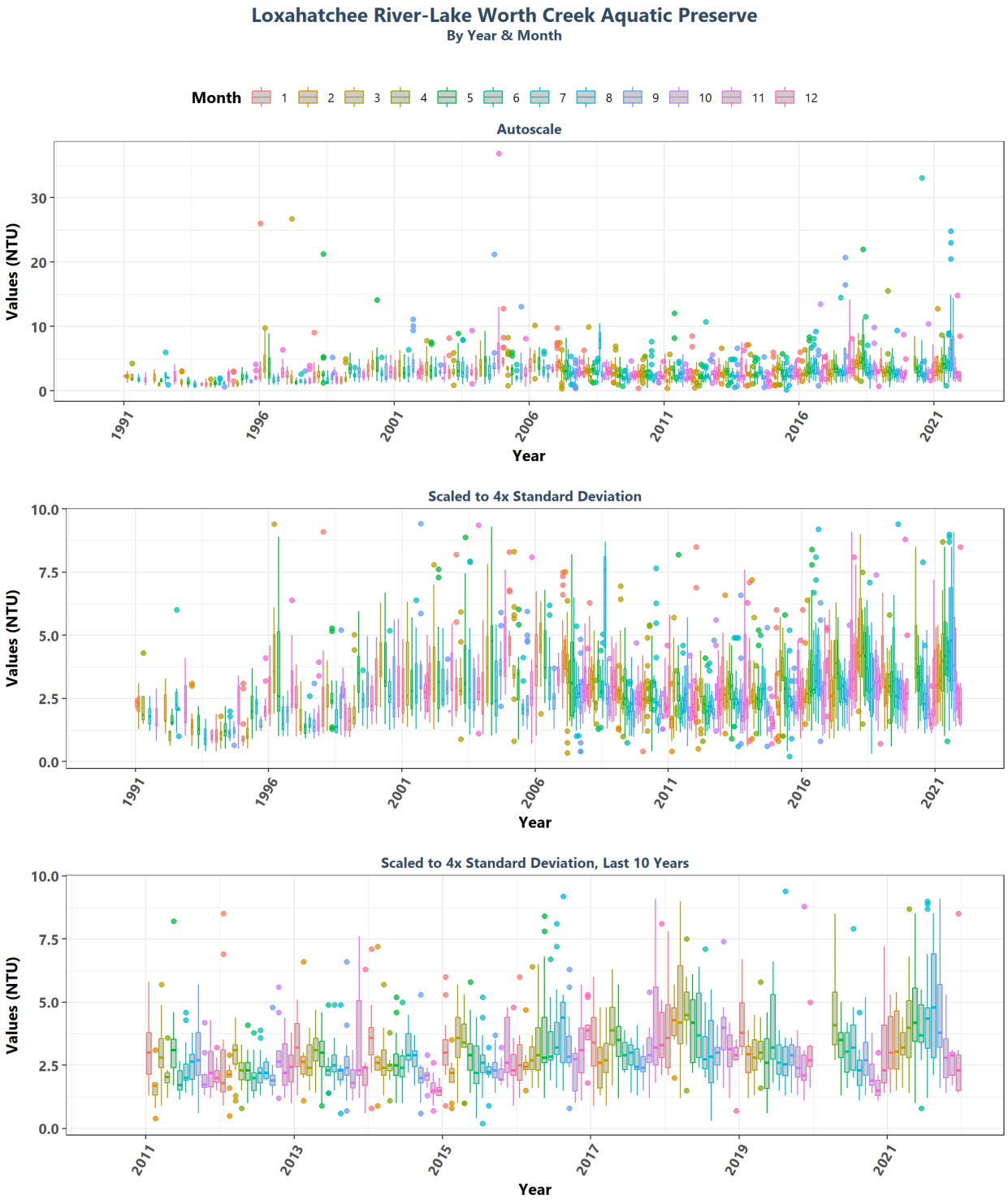

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

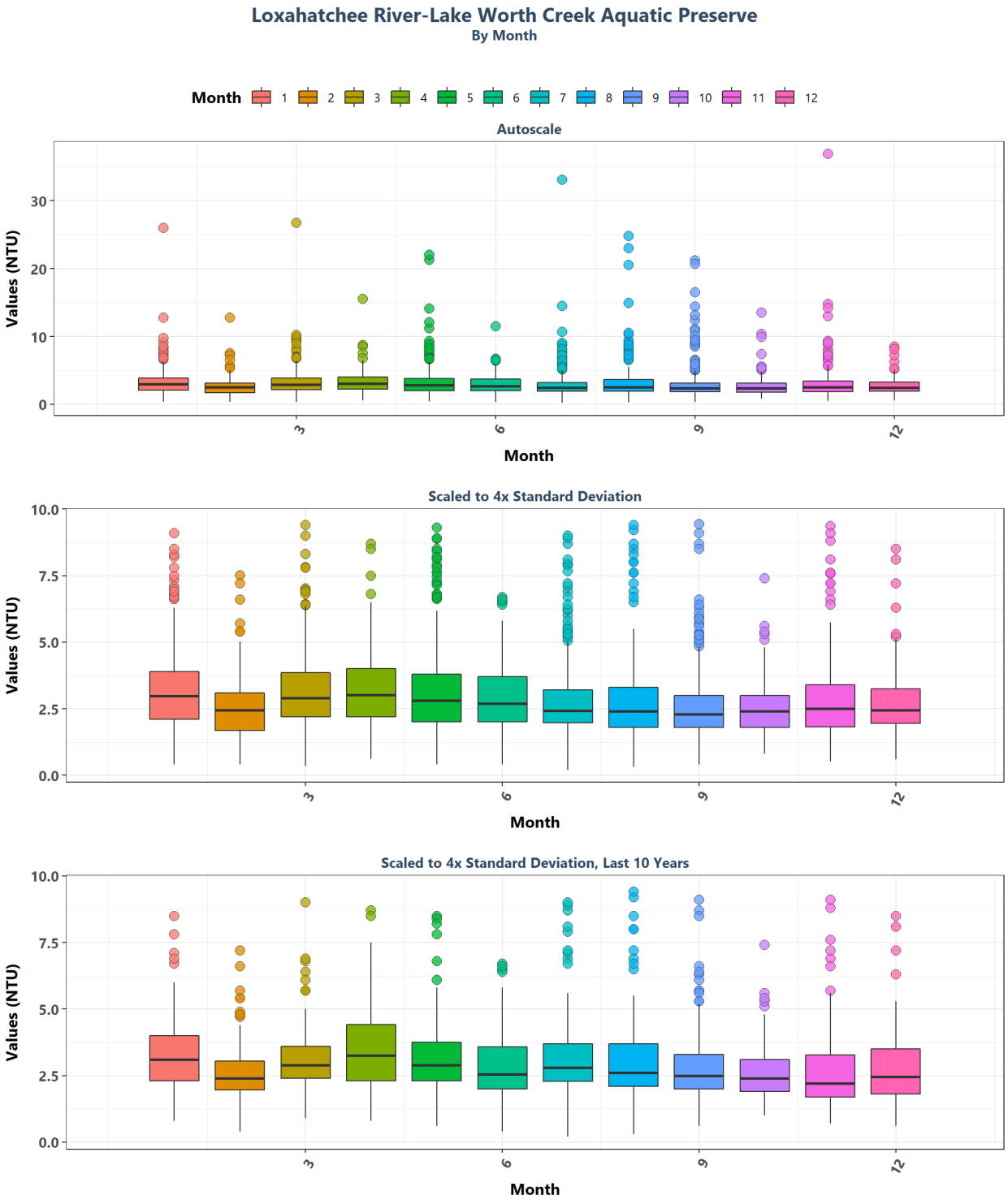

Lemon Bay Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year


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


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

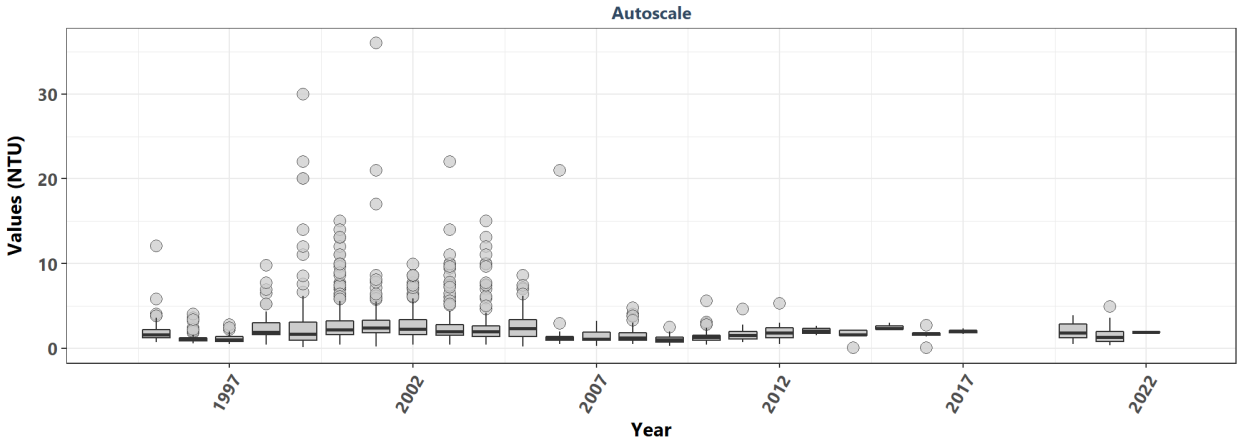

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

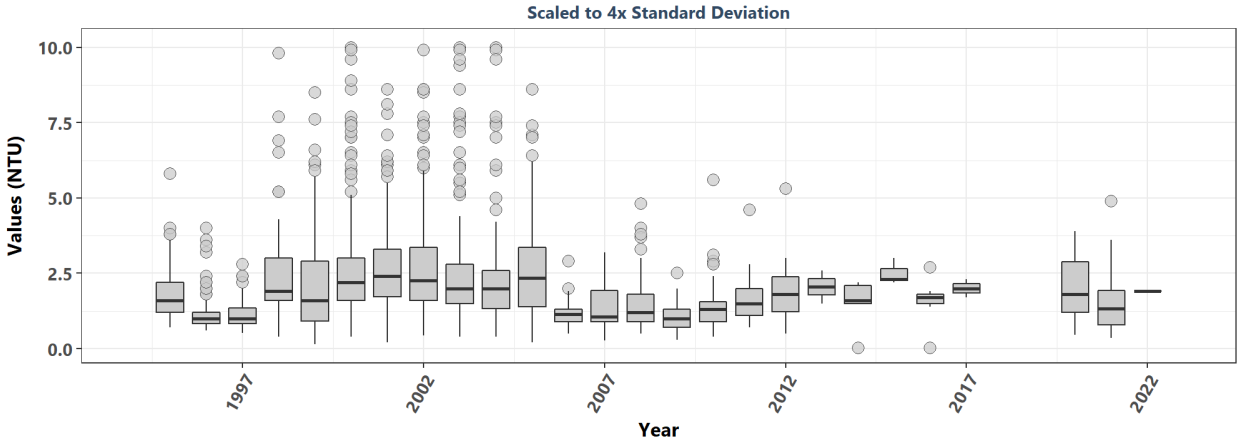

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

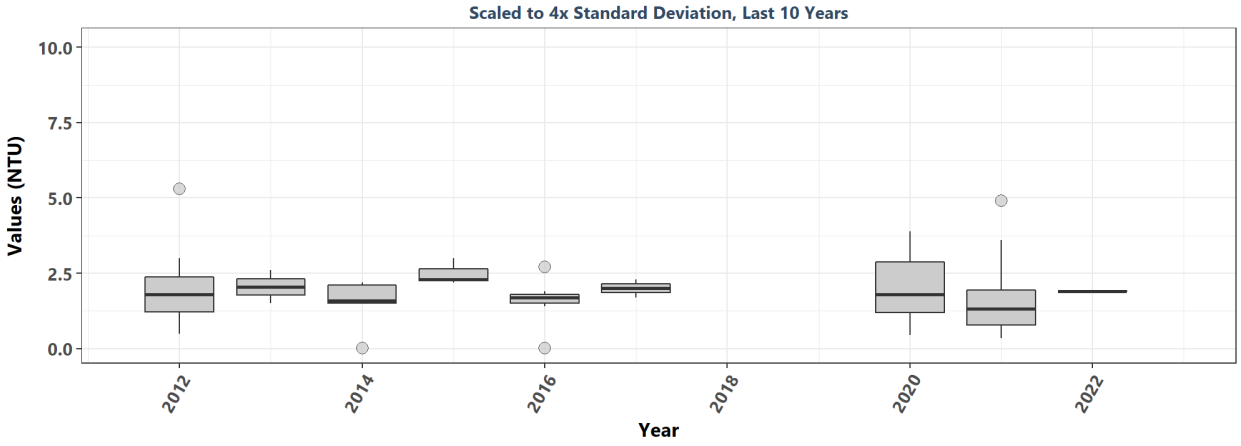

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

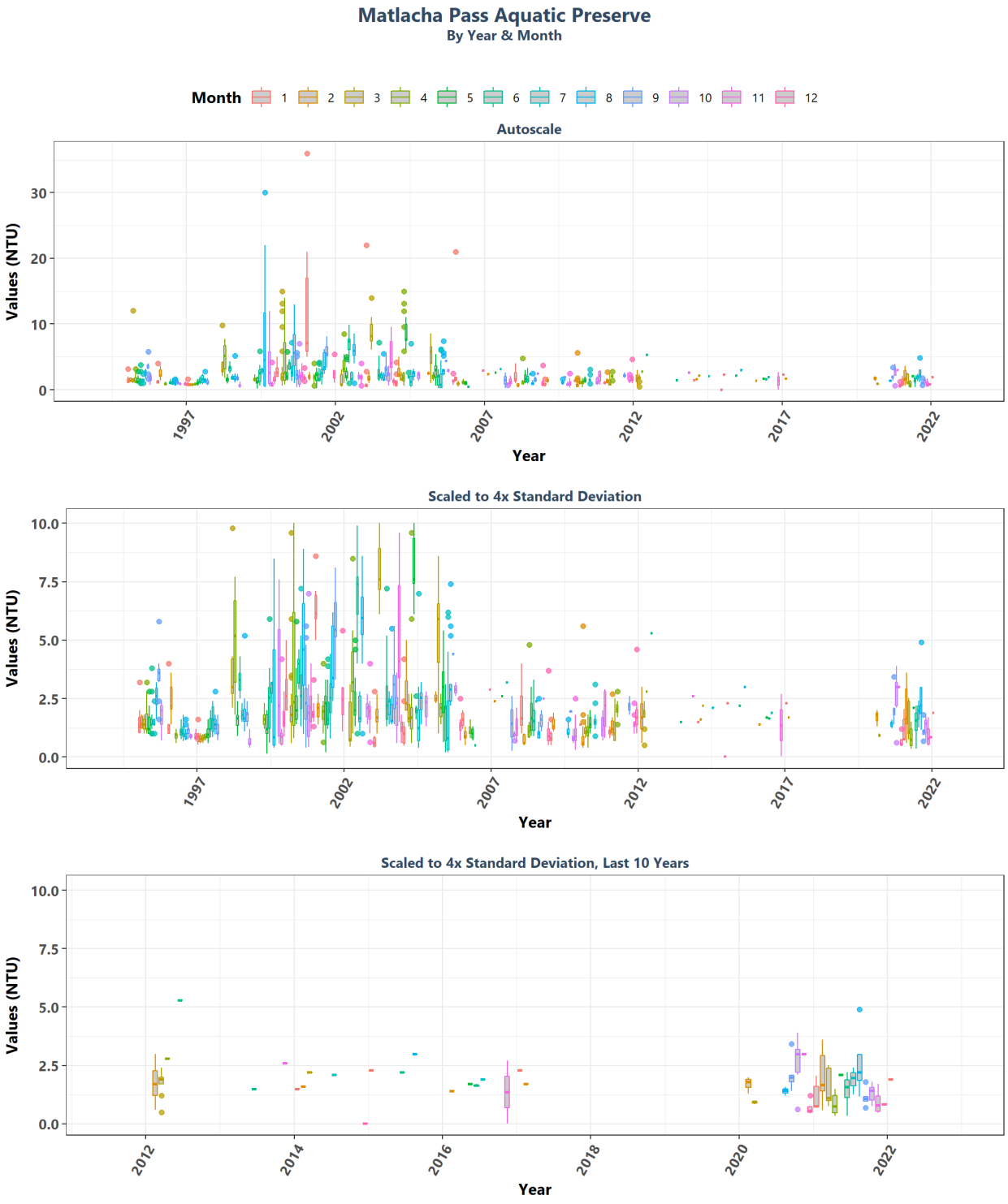
Matlacha Pass Aquatic Preserve

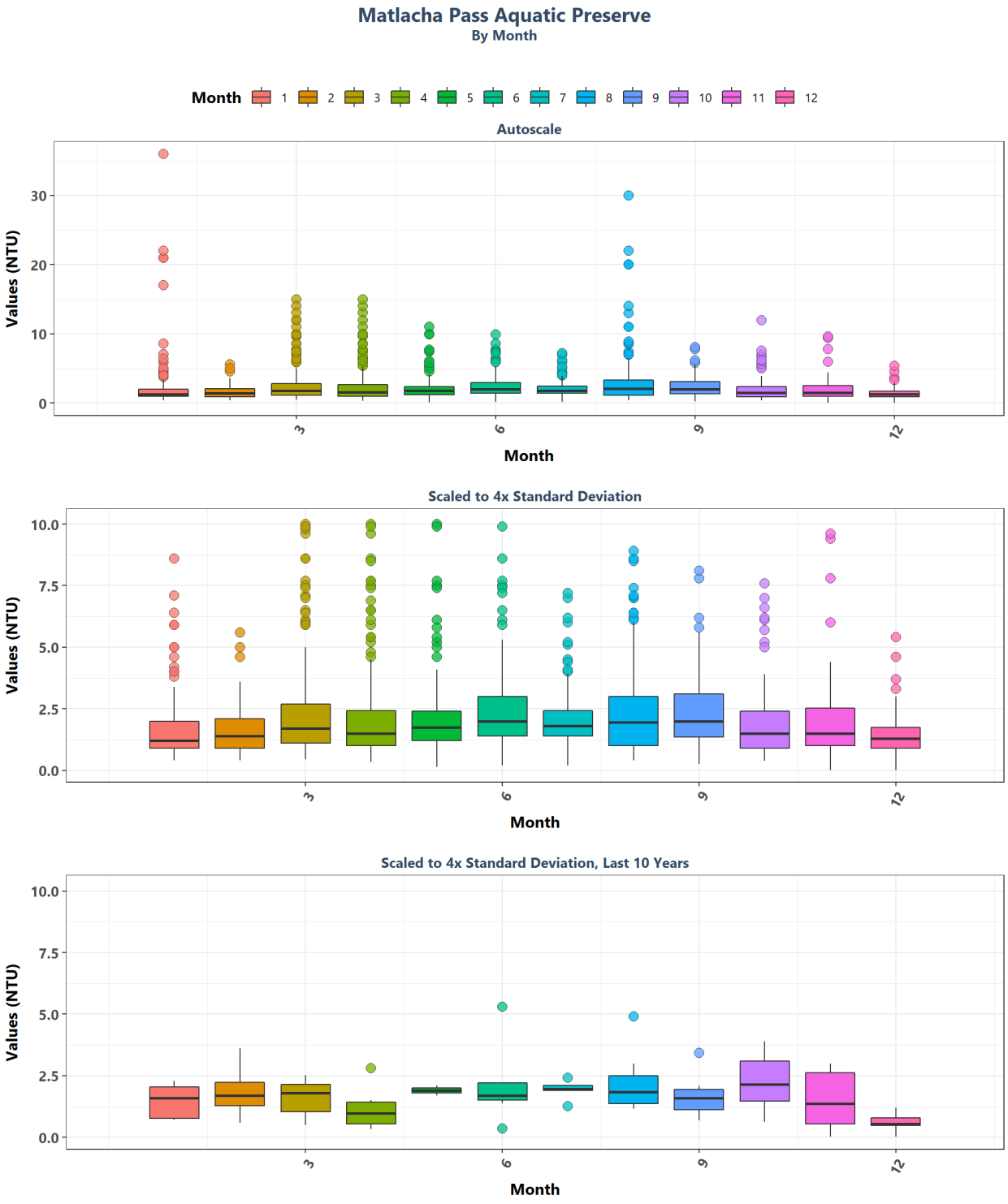
By Year

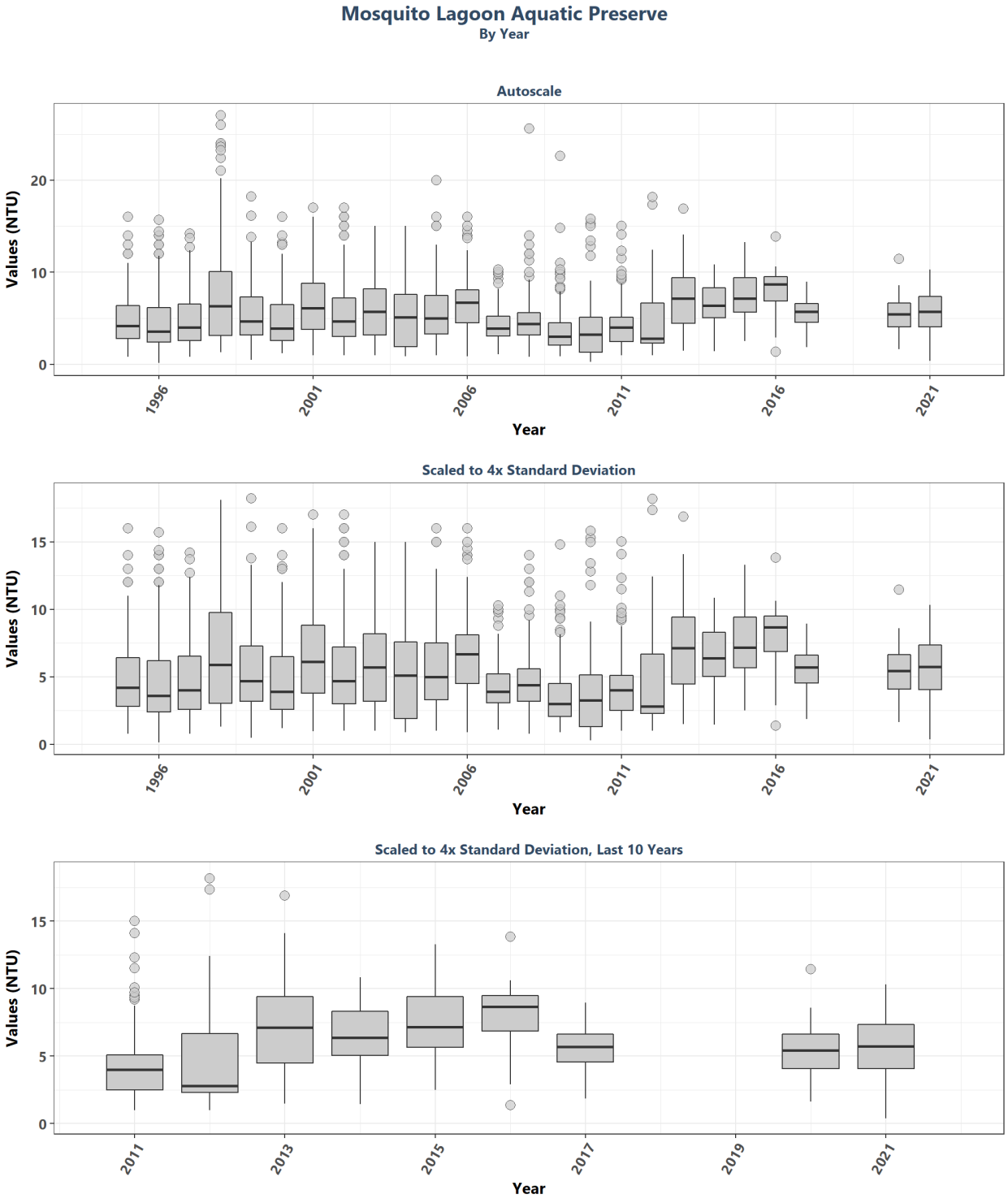


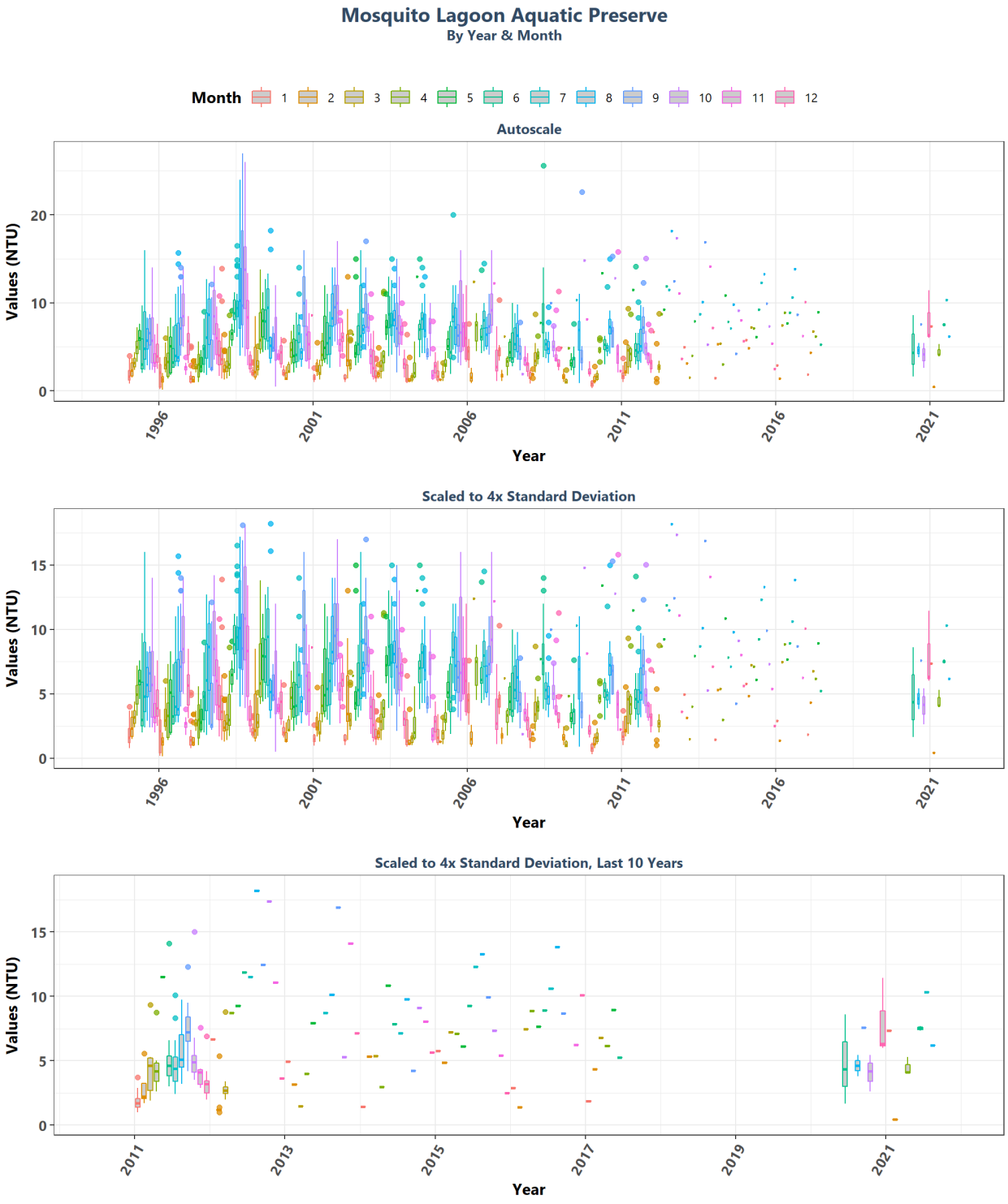


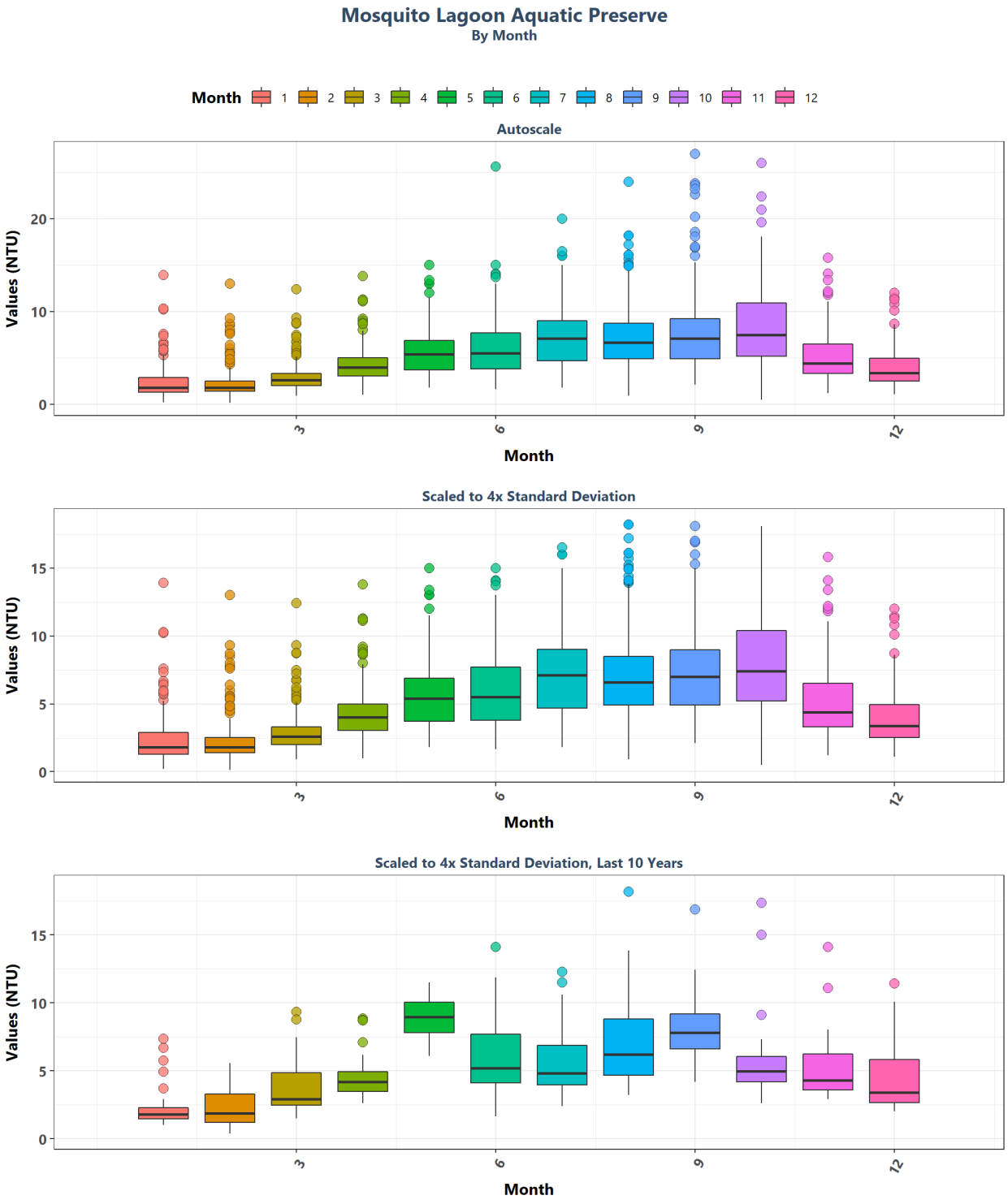


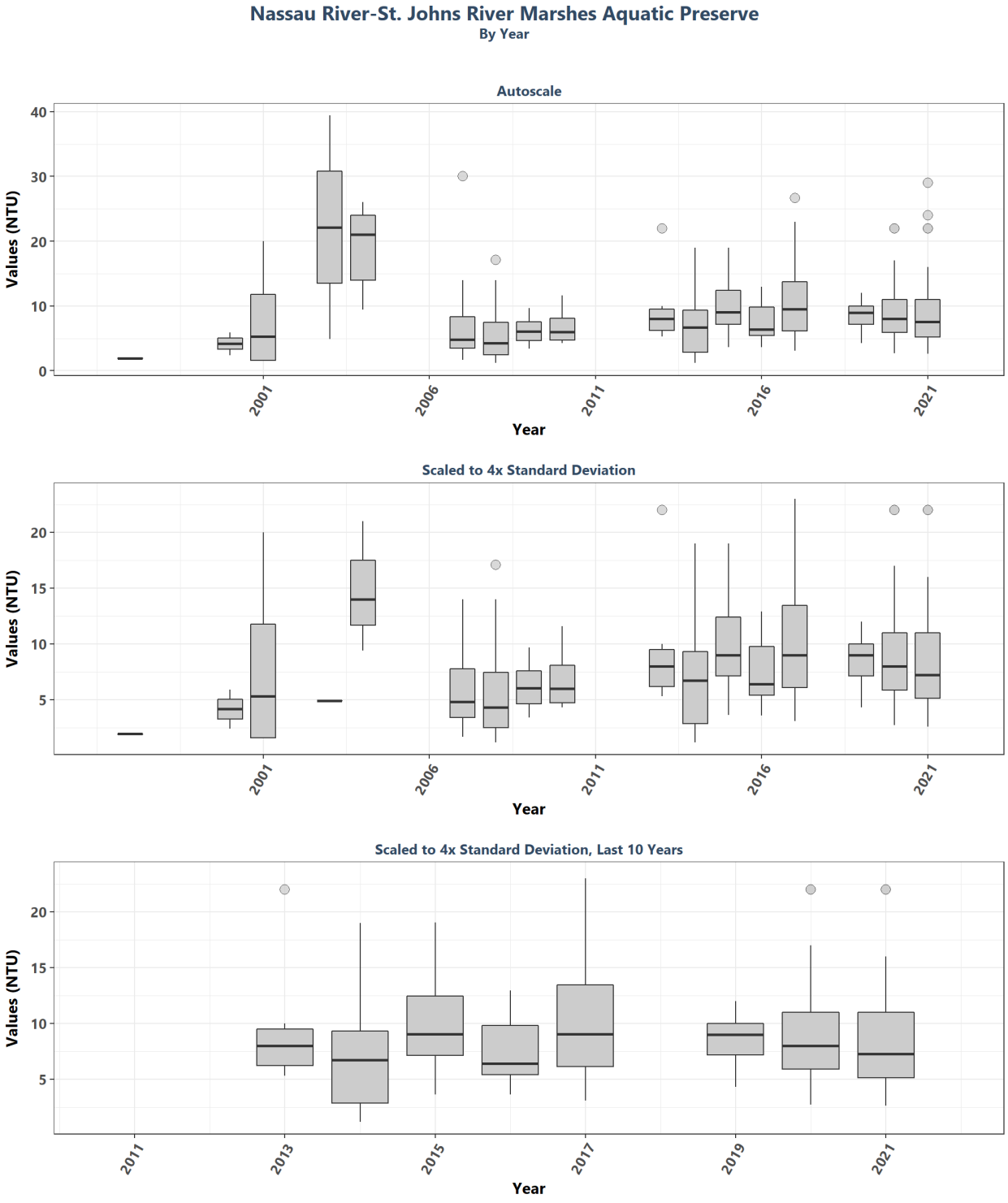

Matlacha Pass Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

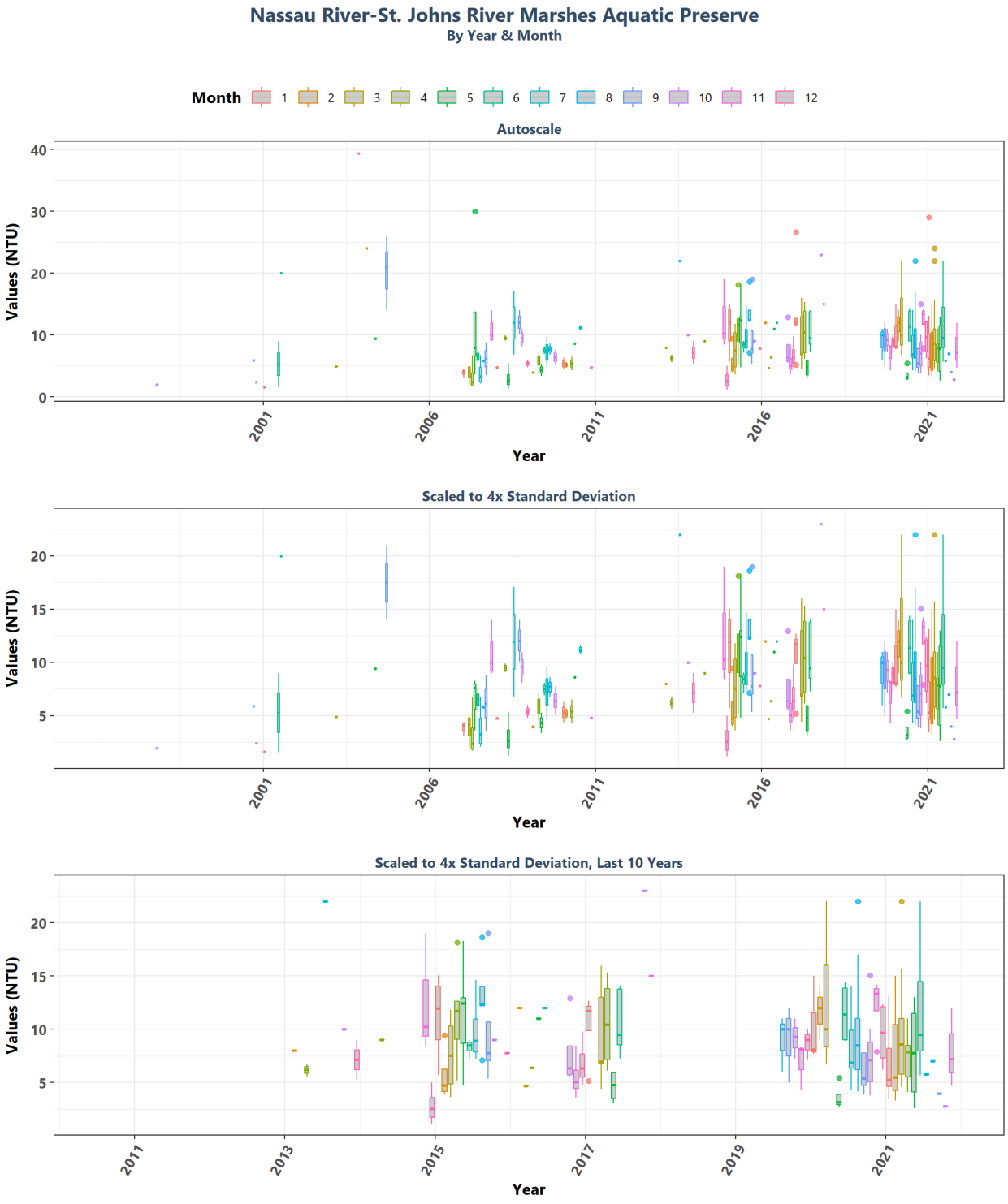

Matlacha Pass Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
30
20
10
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
10.0
7.5
5.0
2.5
0.0
3
6
9
12
Month

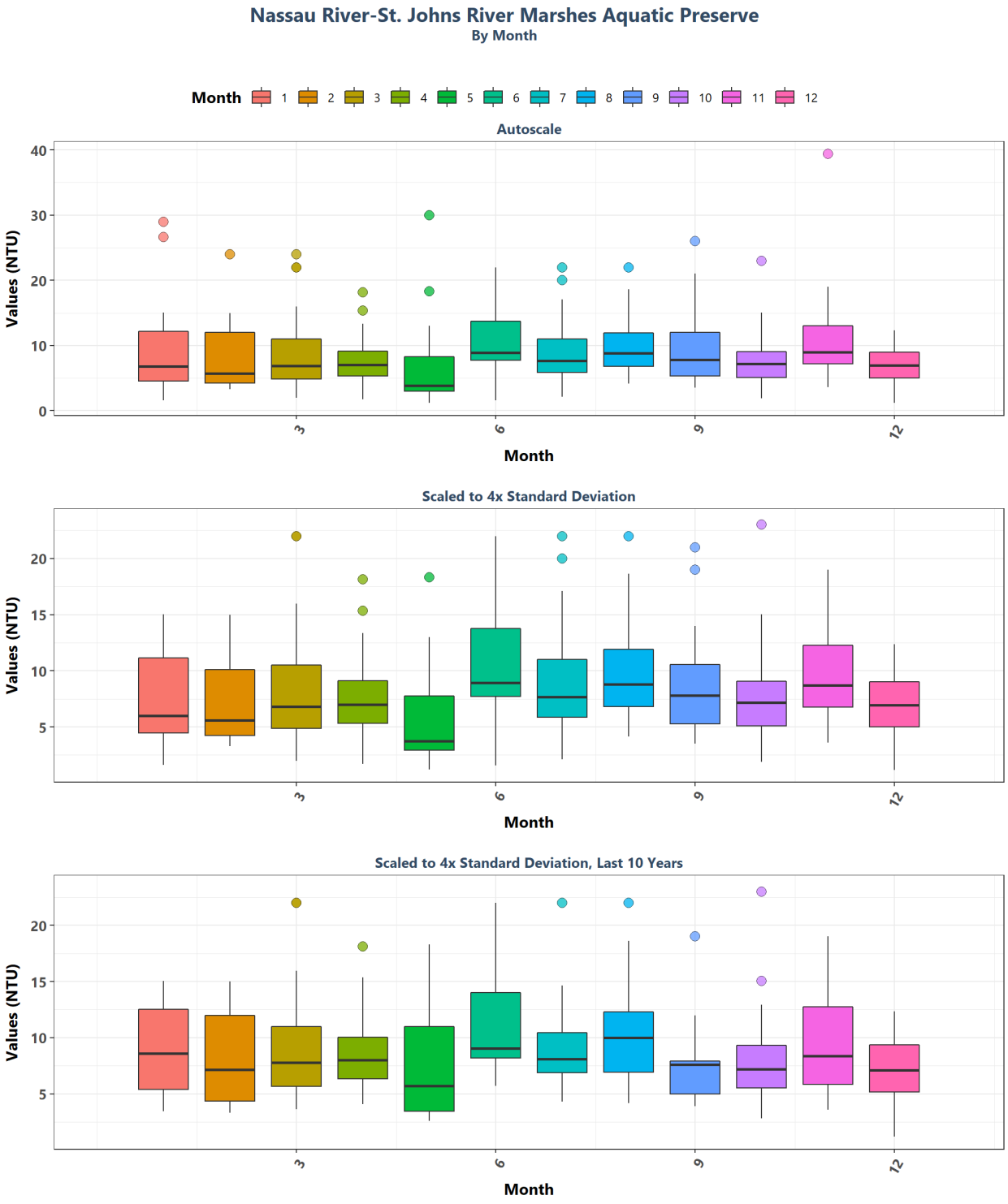

Mosquito Lagoon Aquatic Preserve
By Year
Autoscale
Values (NTU)
0
10
20
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
15
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
15
2011
2013
2015
2017
2019
2021
Year


Mosquito Lagoon Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year


Mosquito Lagoon Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
10
20
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
15
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
15
3
6
9
12
Month

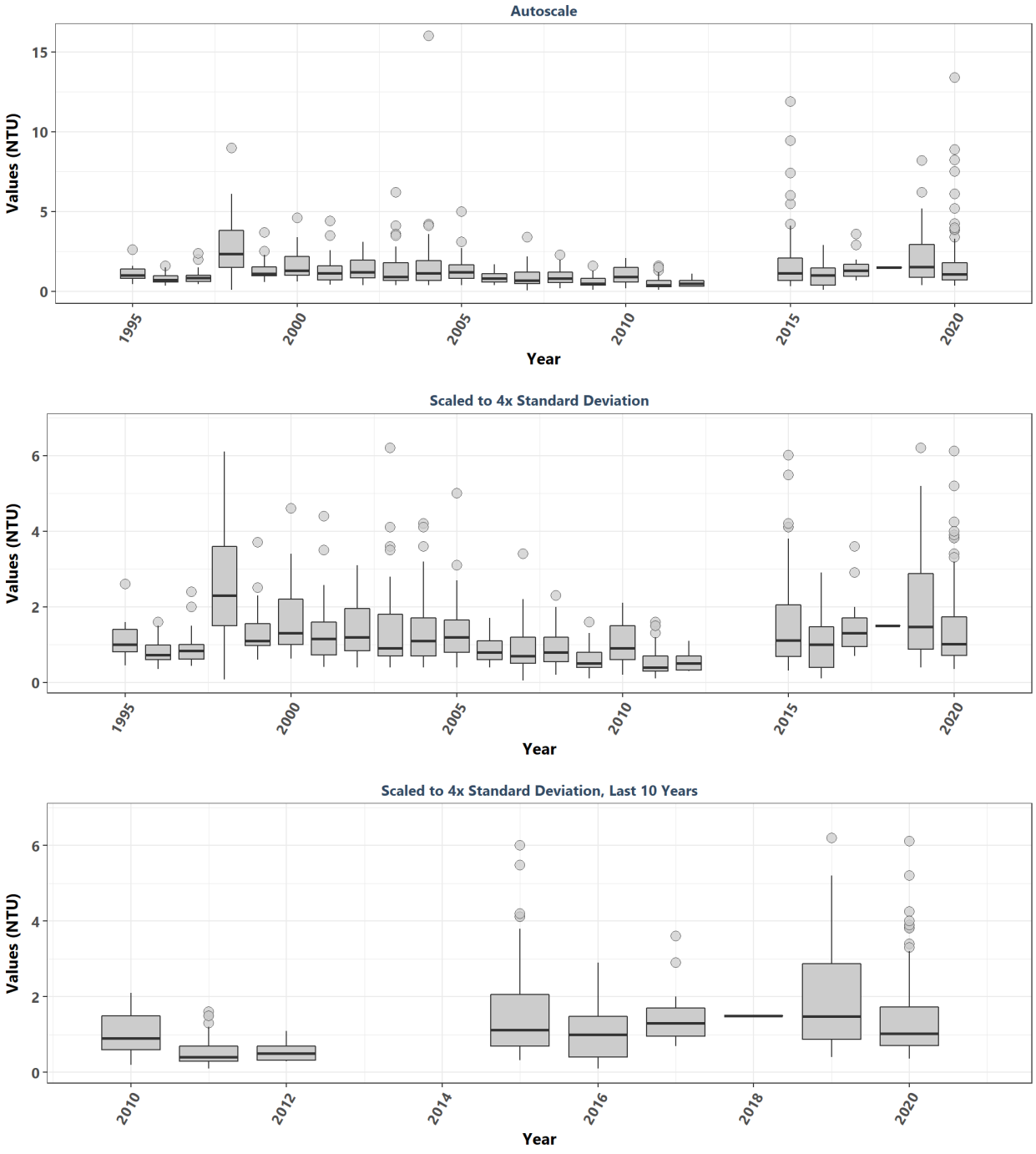

Nassau River-St. Johns River Marshes Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

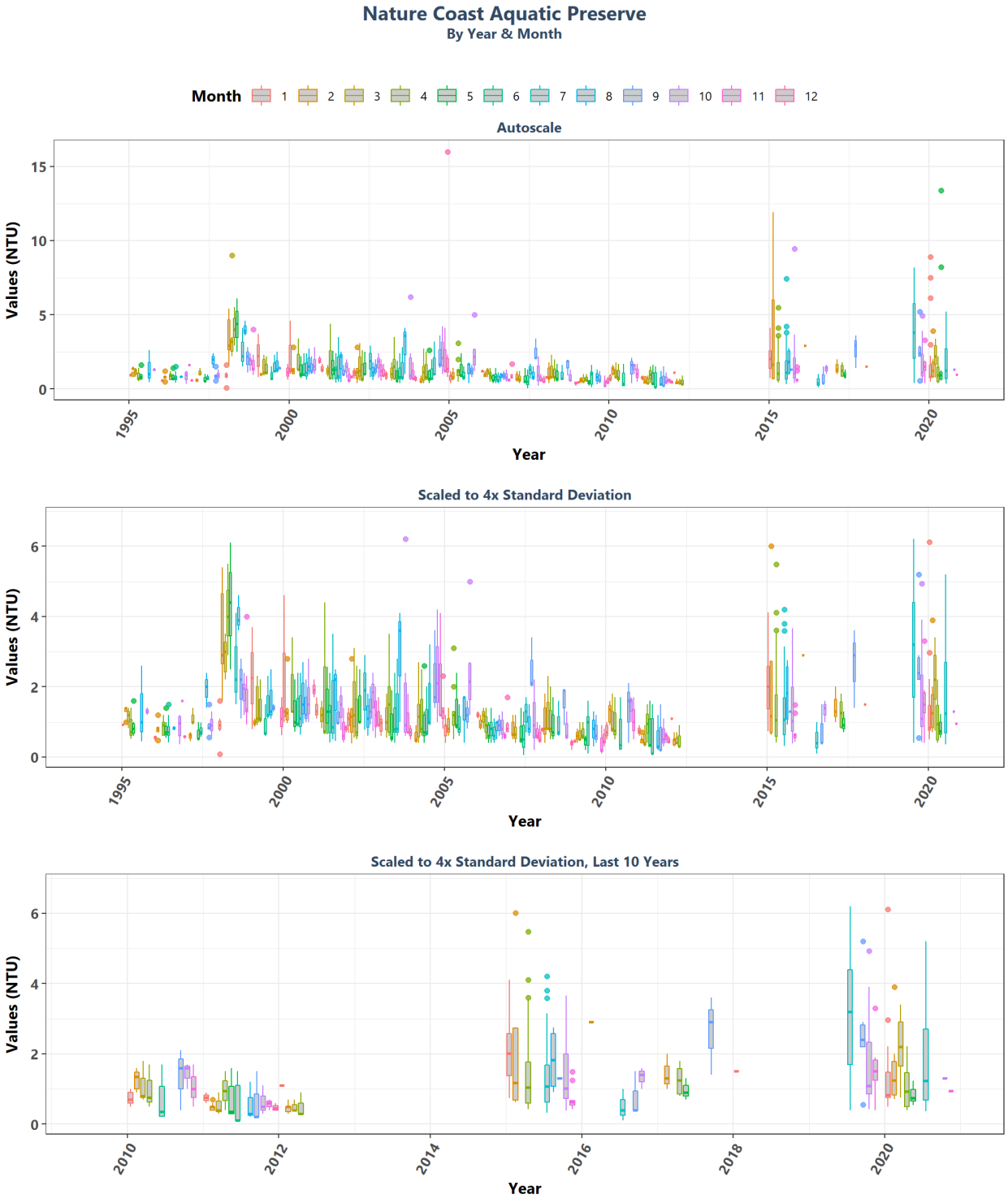

Nassau River-St. Johns River Marshes Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

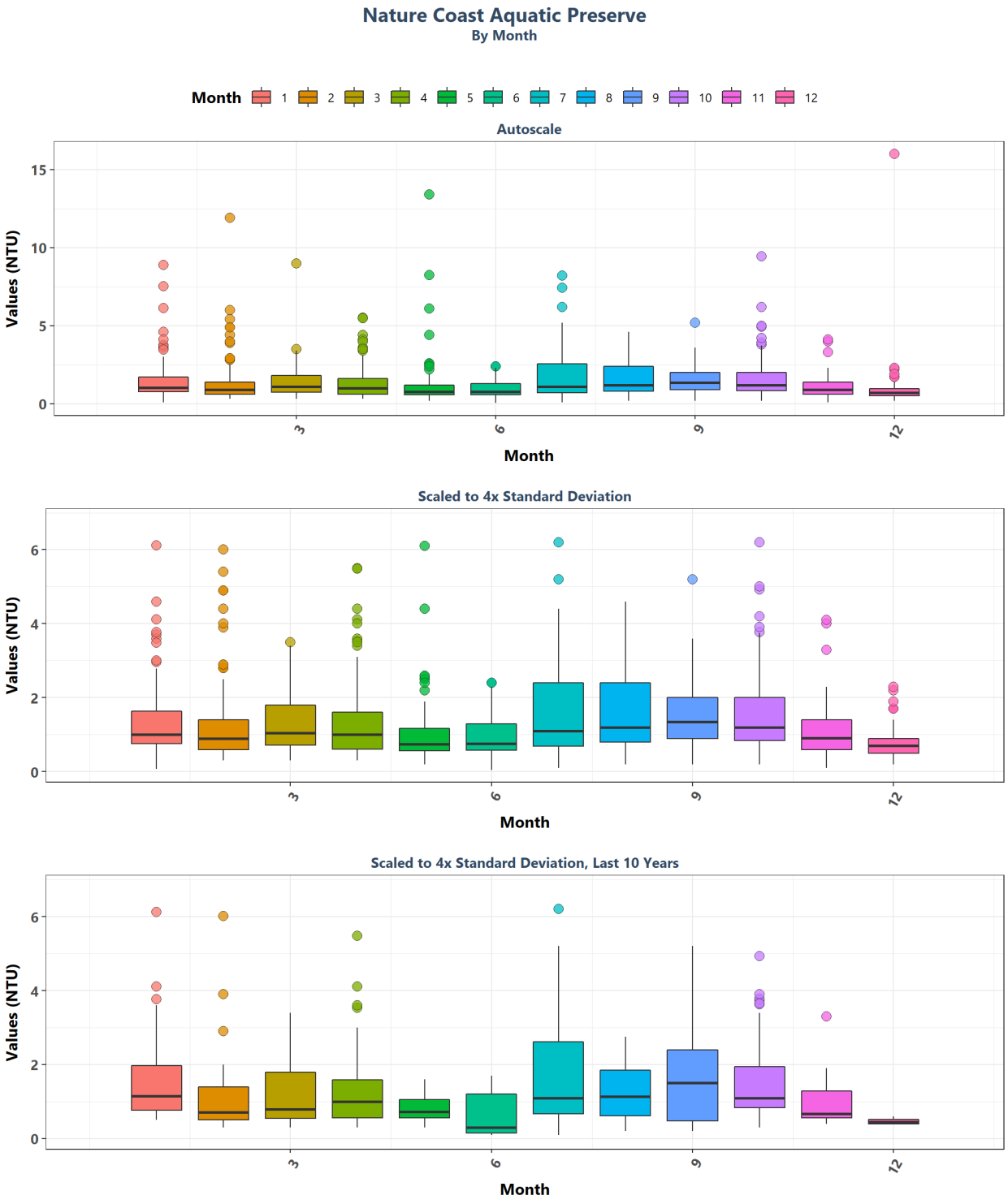

Nassau River-St. Johns River Marshes Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

Nature Coast Aquatic Preserve

By Year

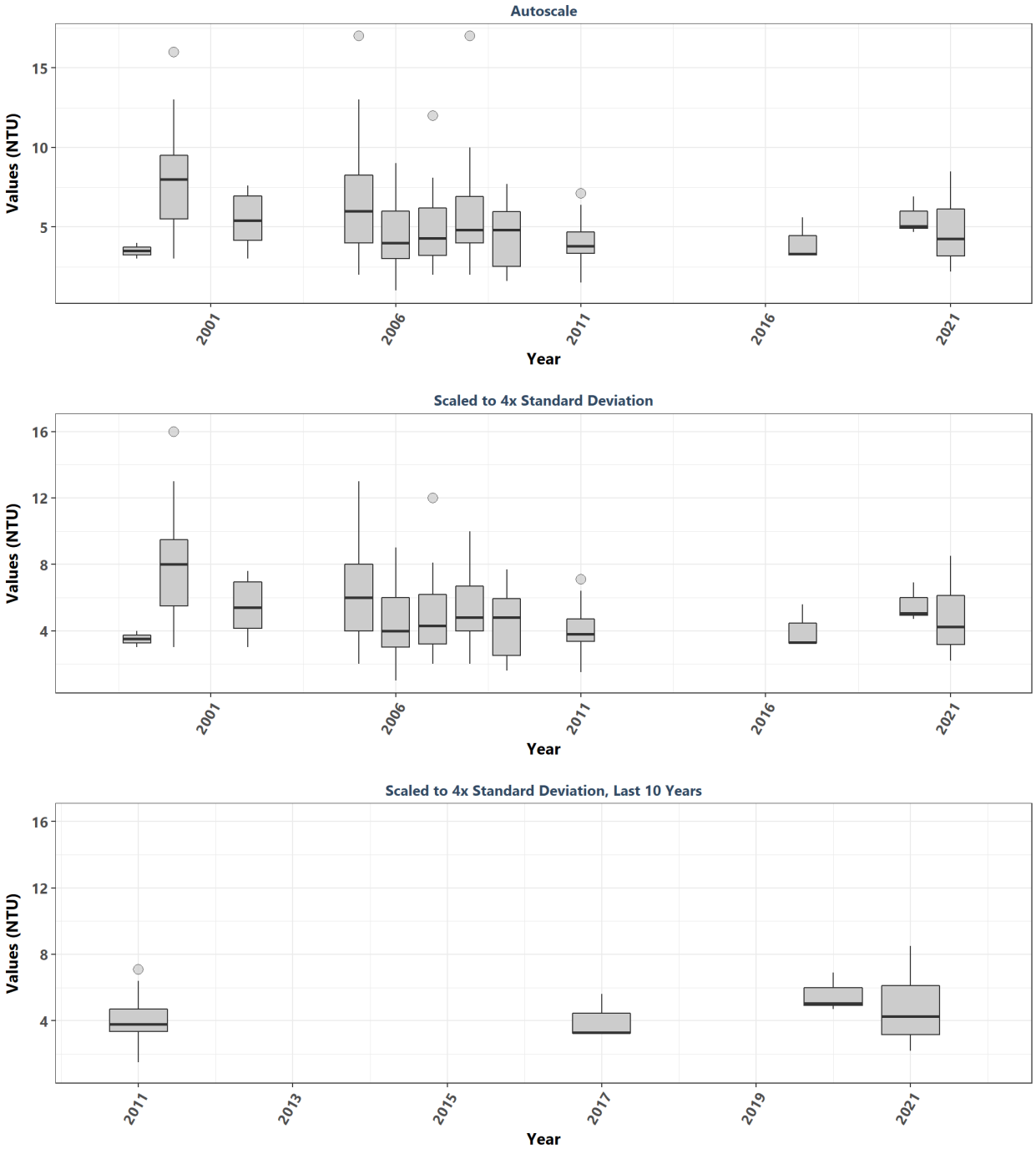


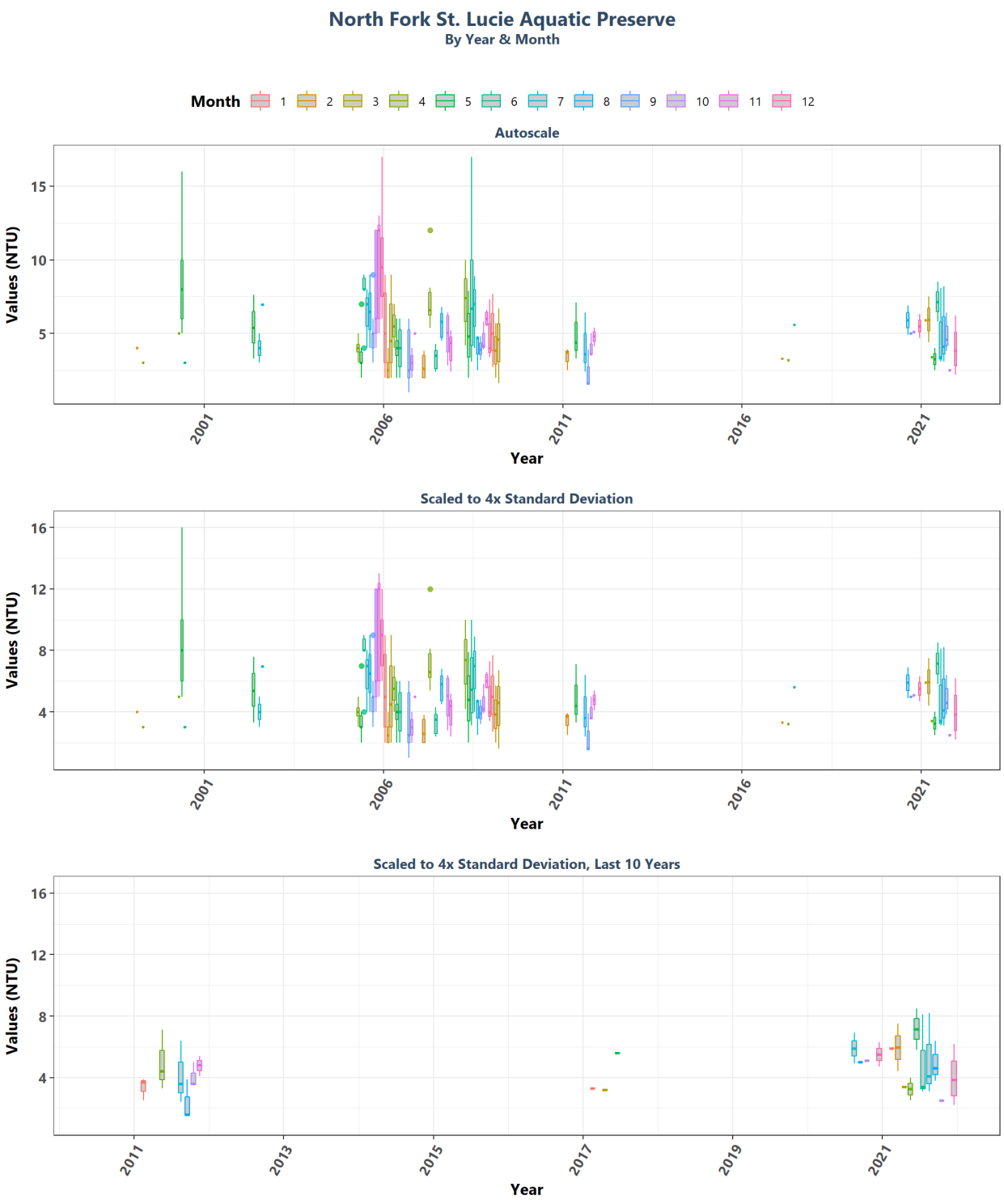

Nature Coast Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

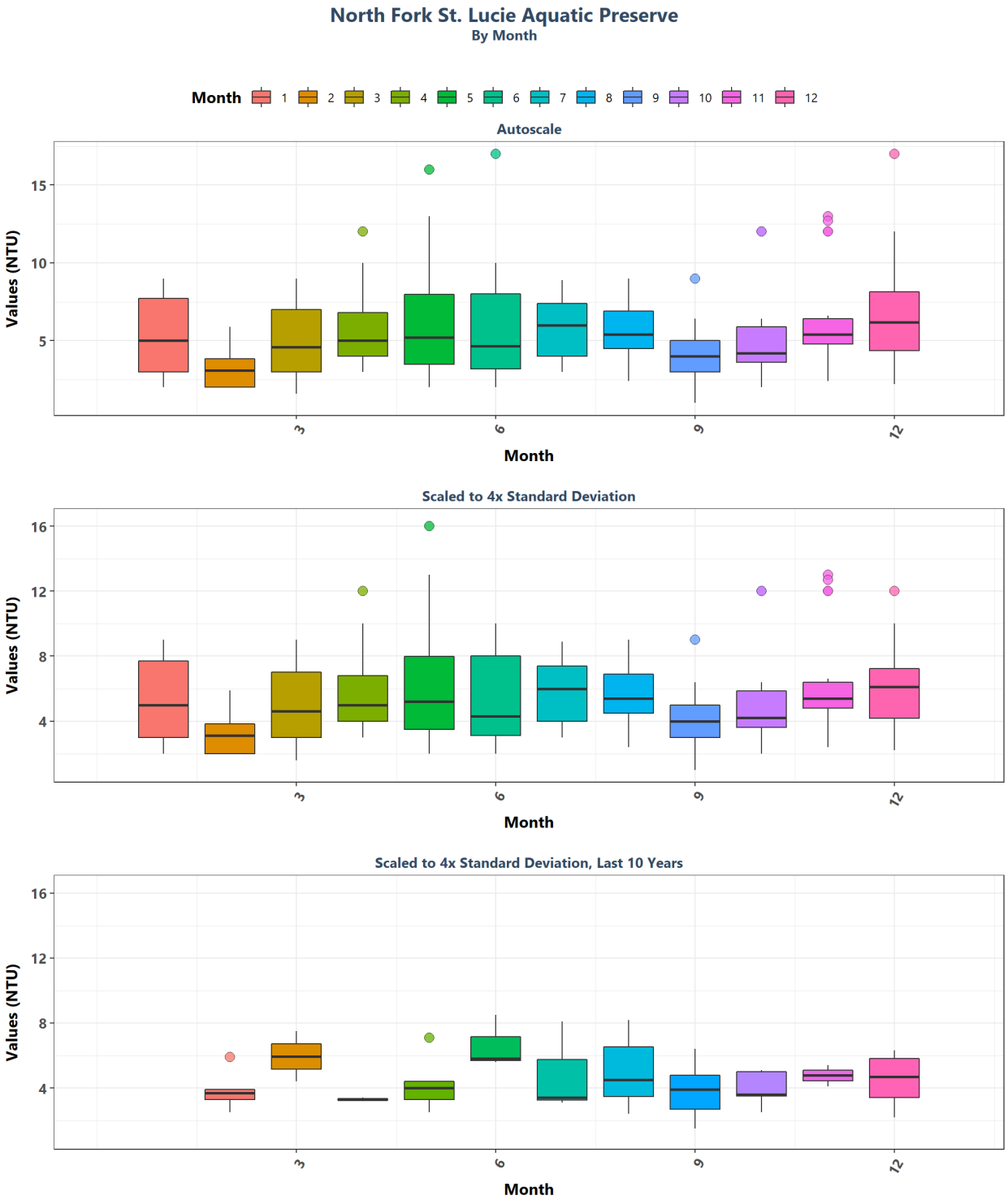

Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

North Fork St. Lucie Aquatic Preserve

By Year

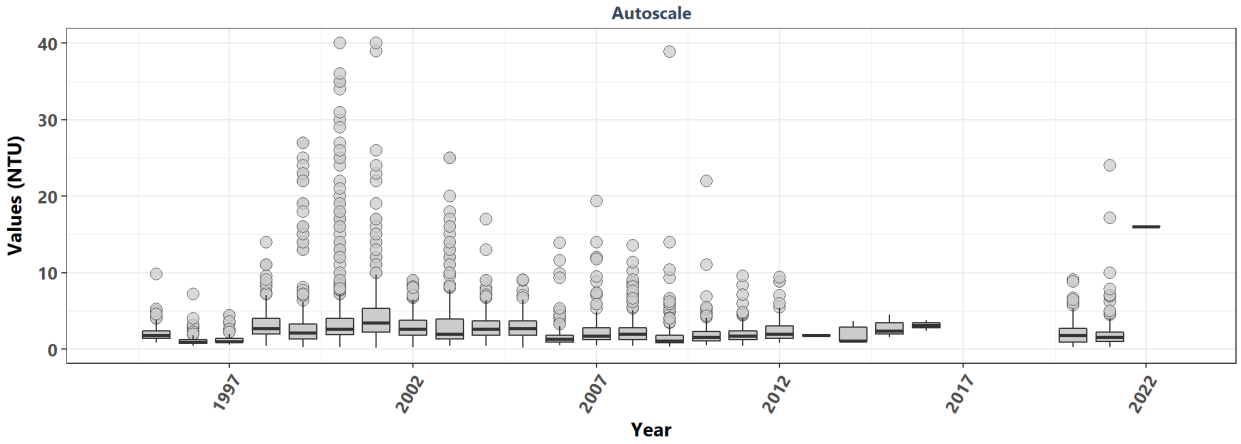


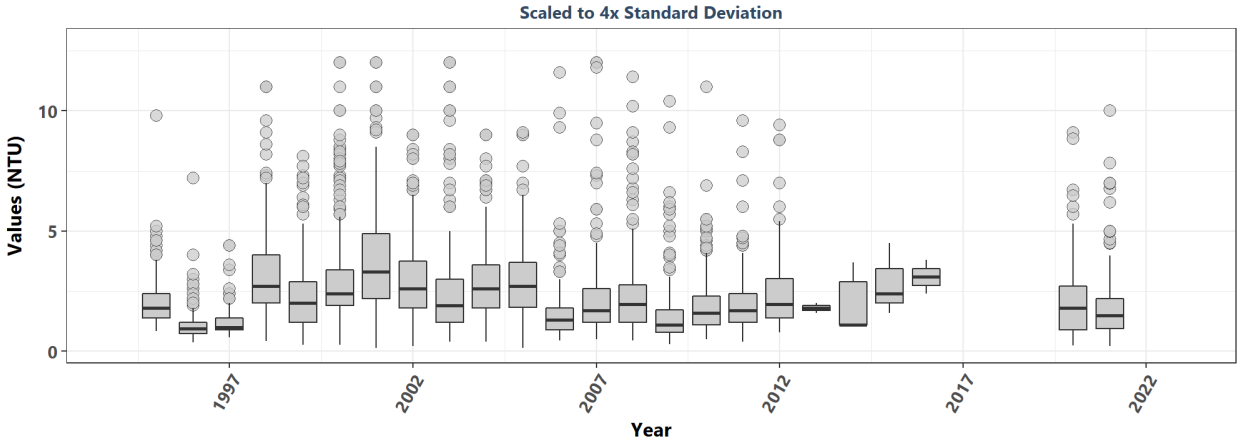

North Fork St. Lucie Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
15
10
5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
16
12
8
4
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
16
12
8
4
2011
2013
2015
2017
2019
2021
Year

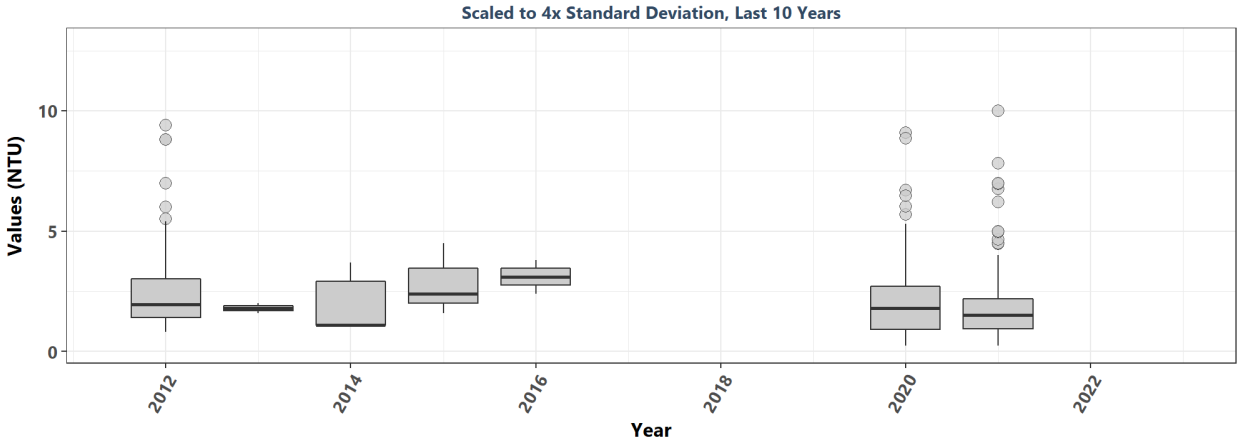

North Fork St. Lucie Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
15
10
5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
16
12
8
4
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
16
12
8
4
3
6
9
12
Month

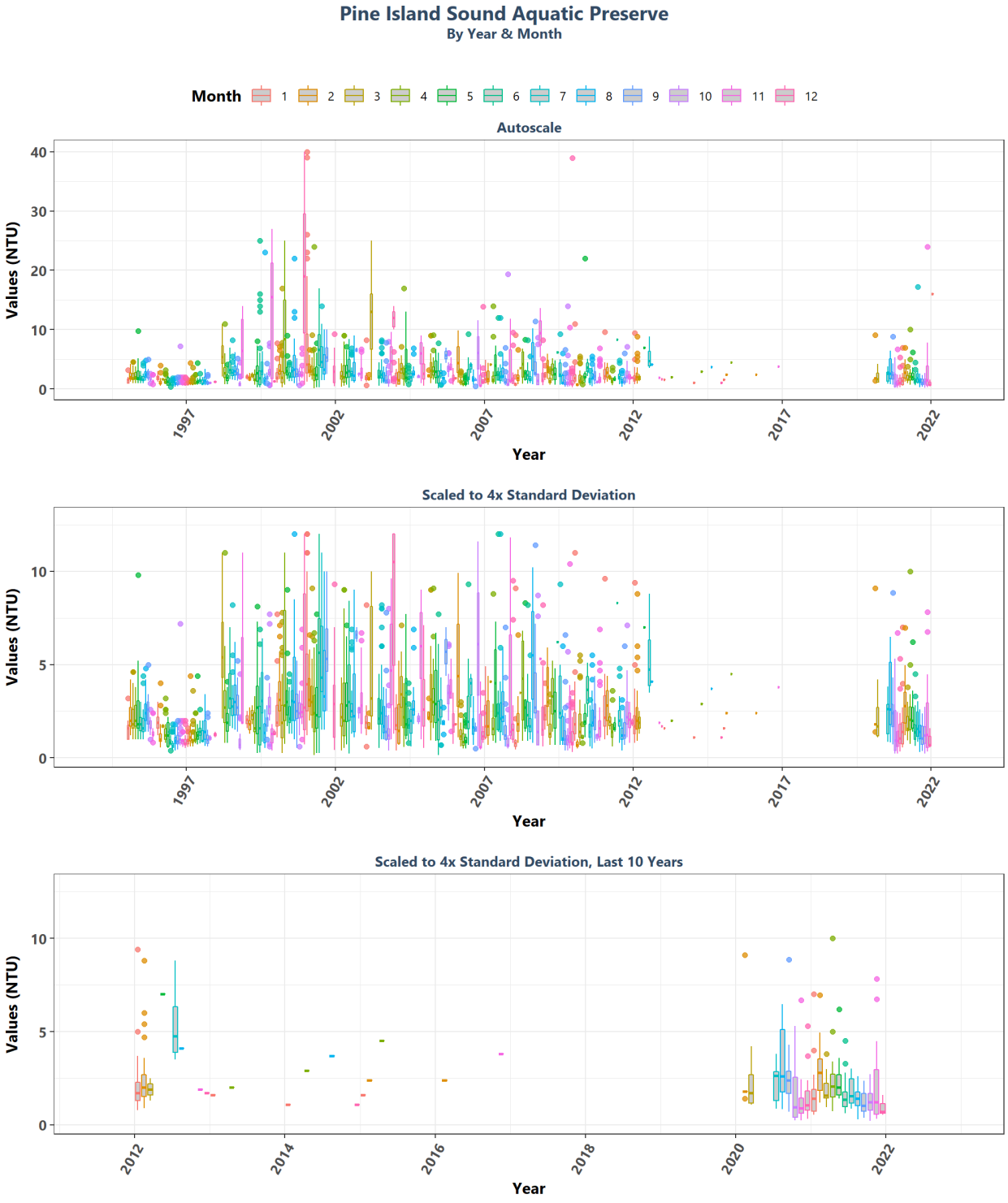
Pine Island Sound Aquatic Preserve

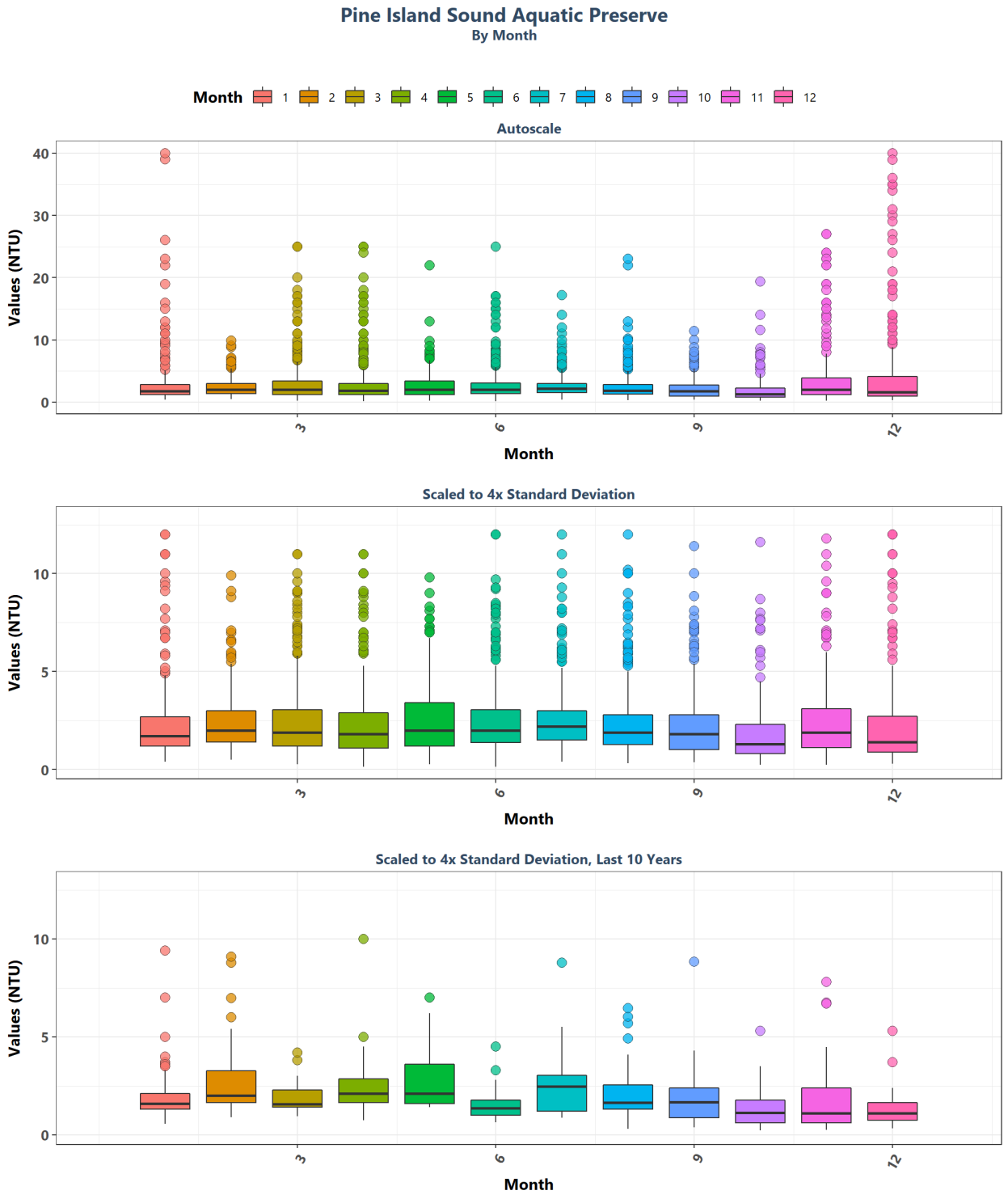
By Year

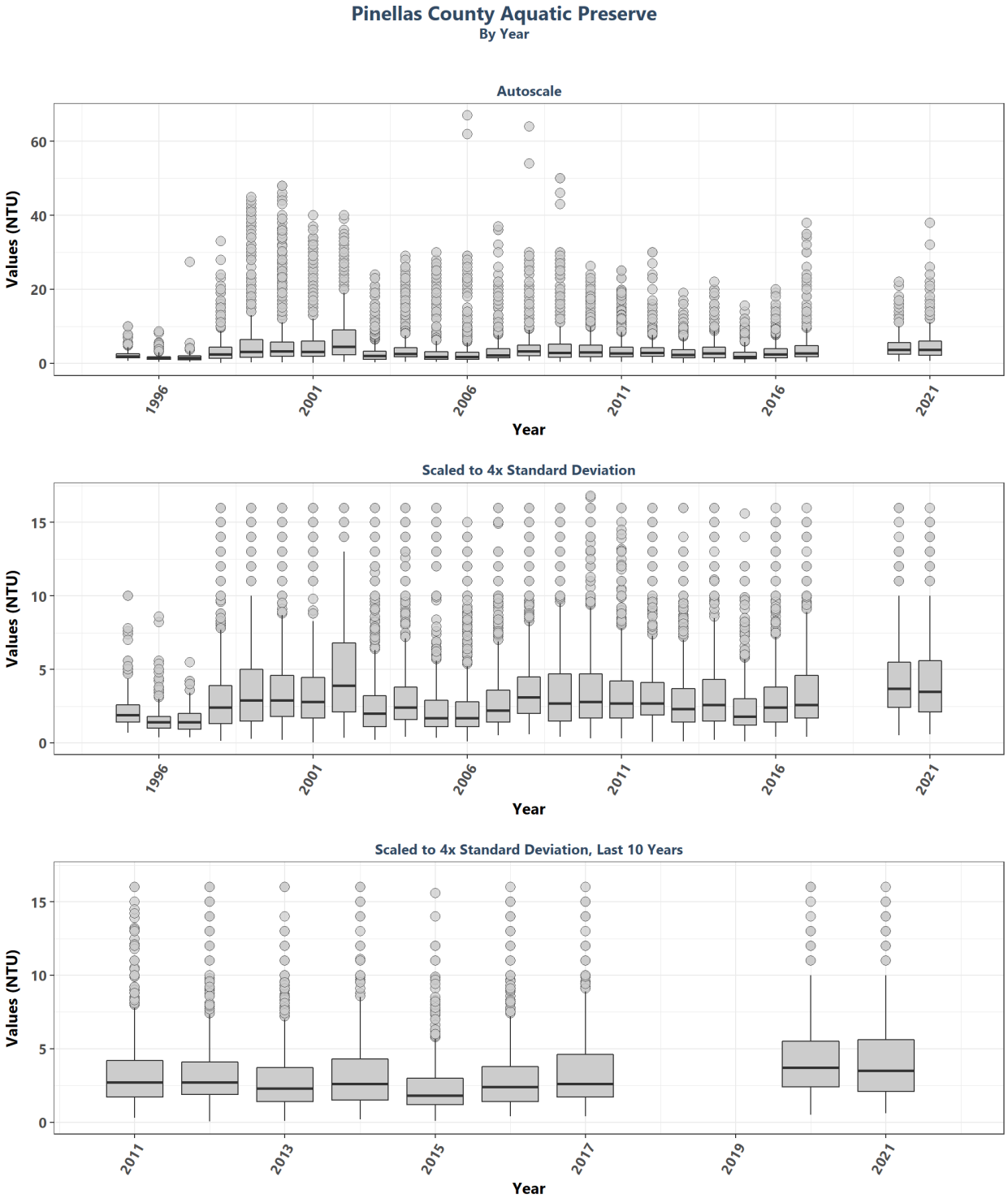


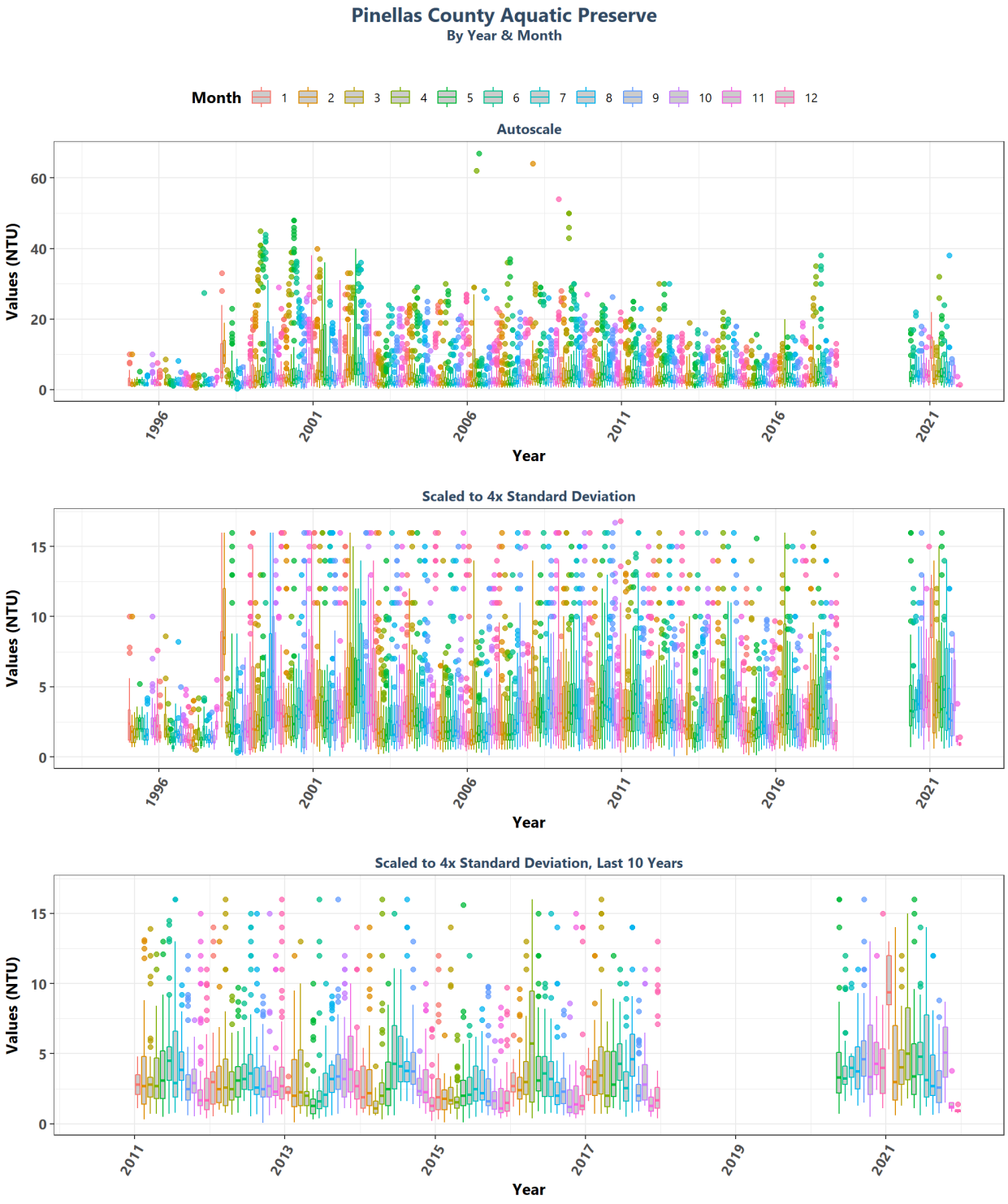


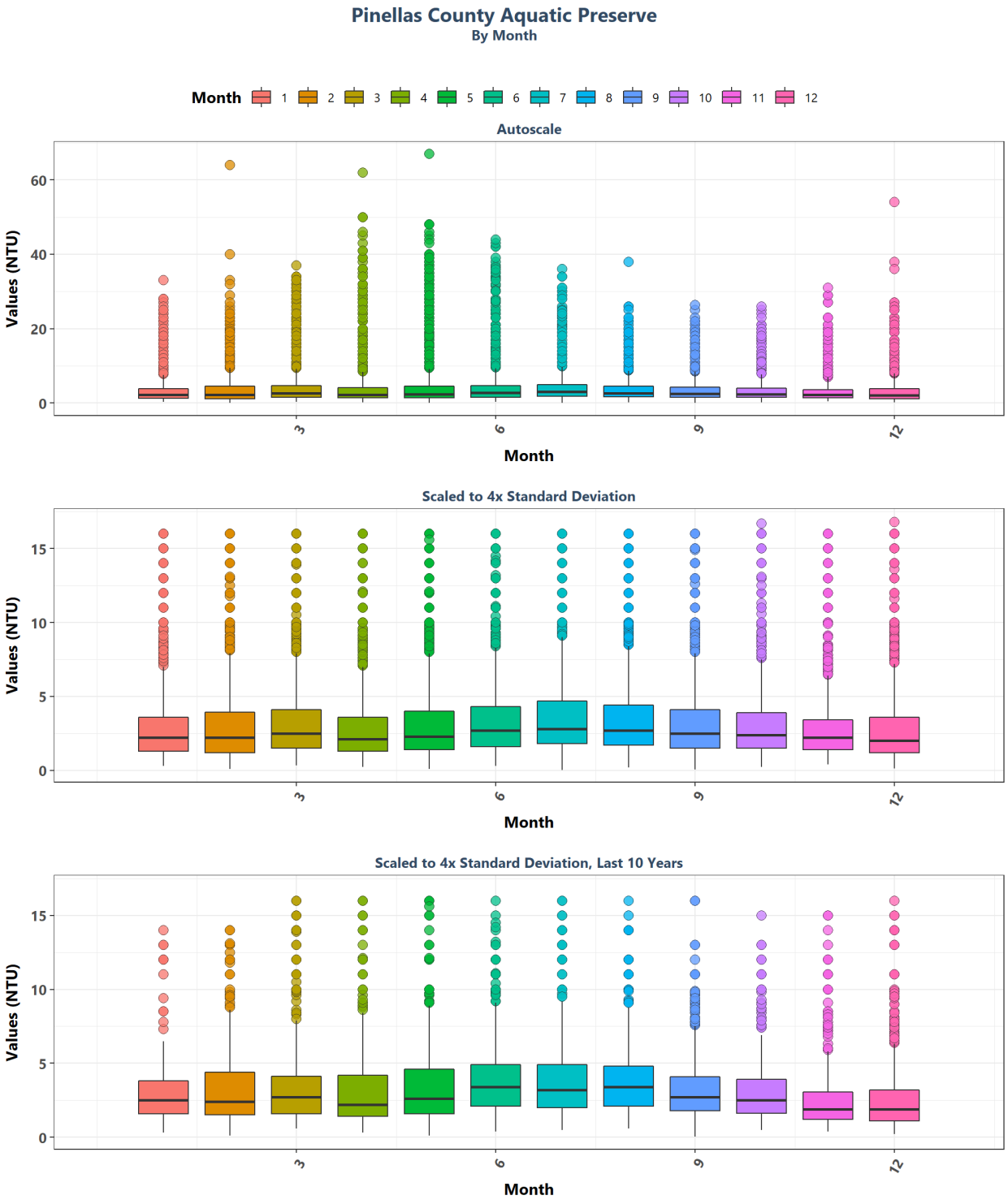


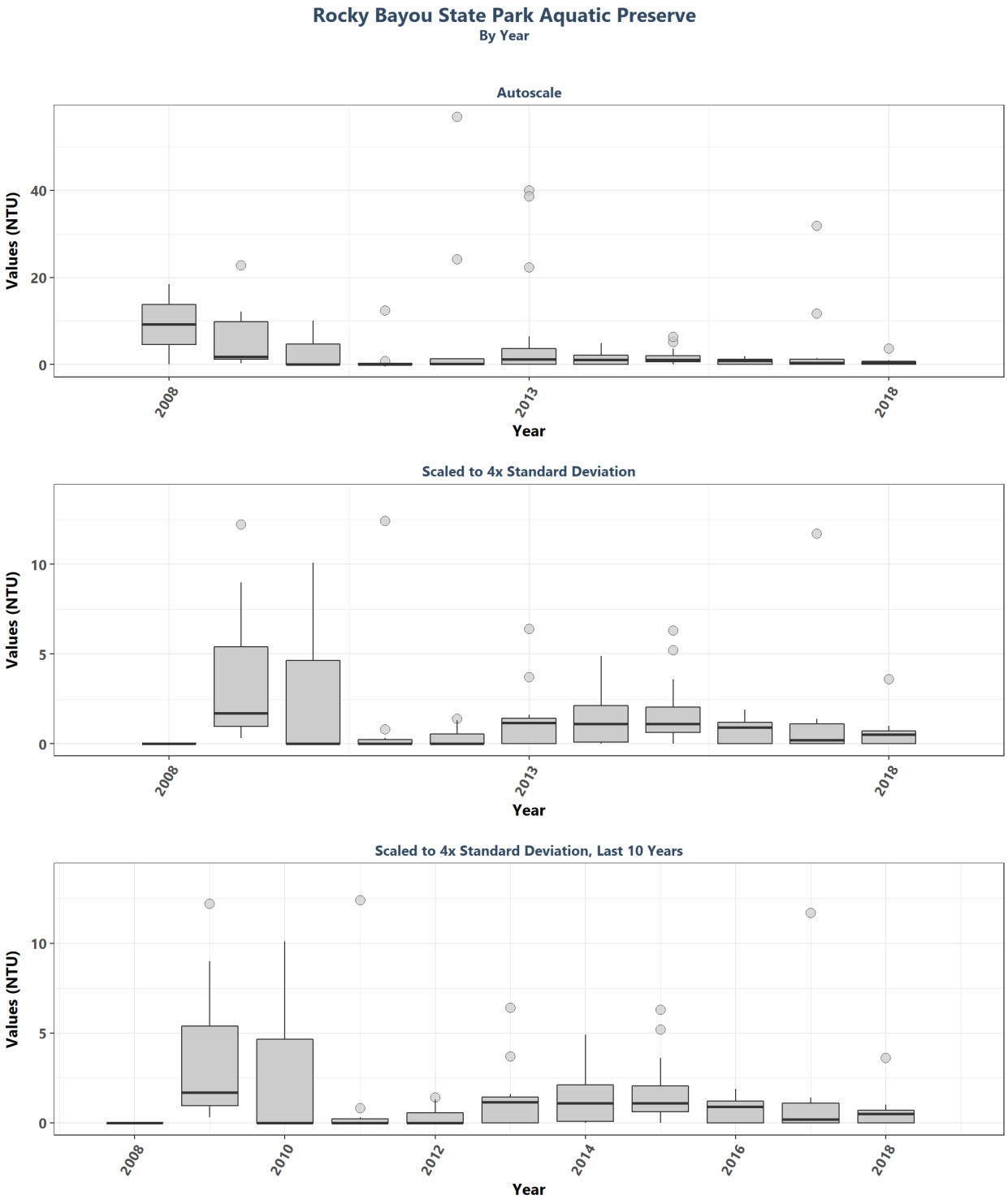

Pine Island Sound Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

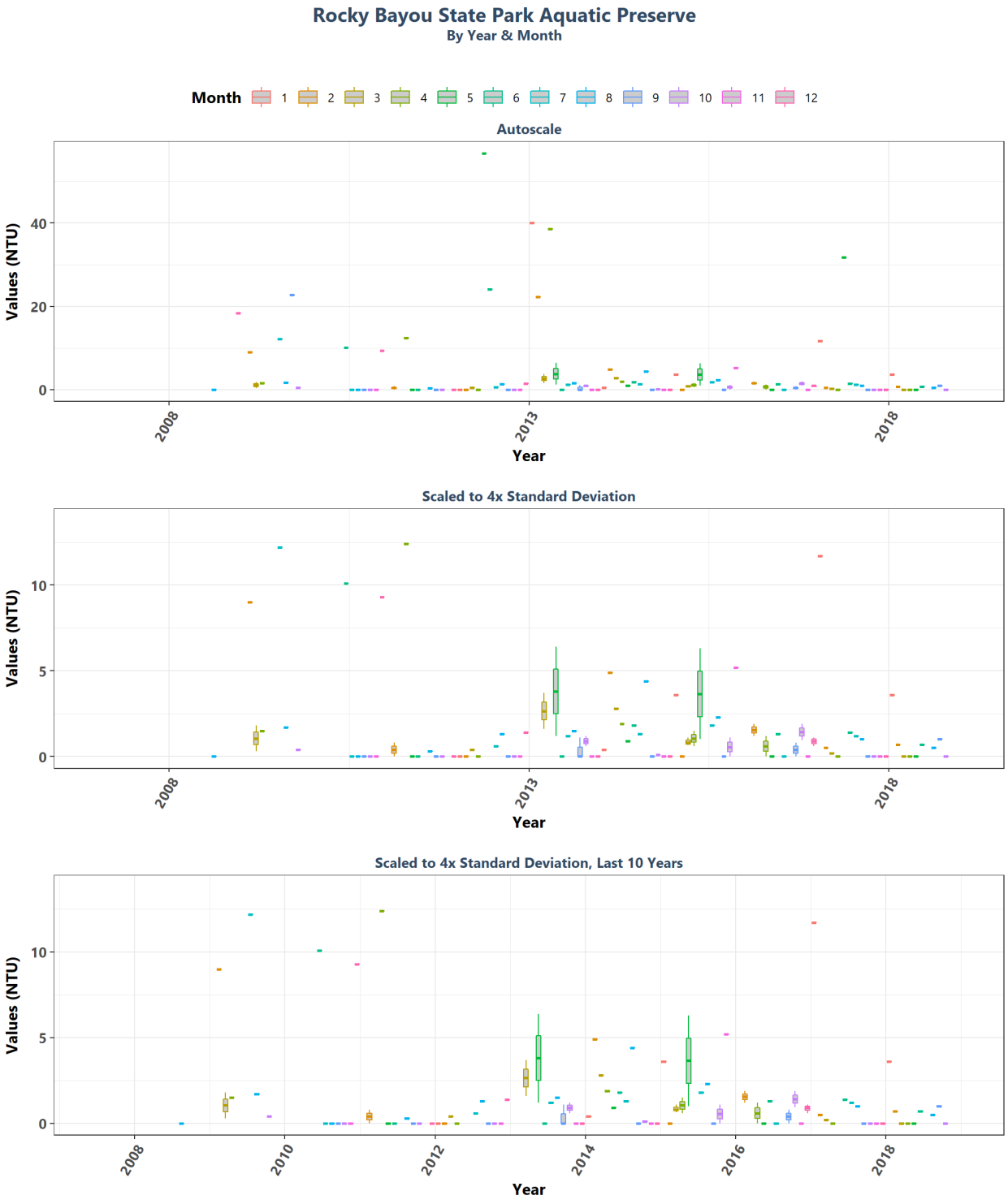

Pine Island Sound Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

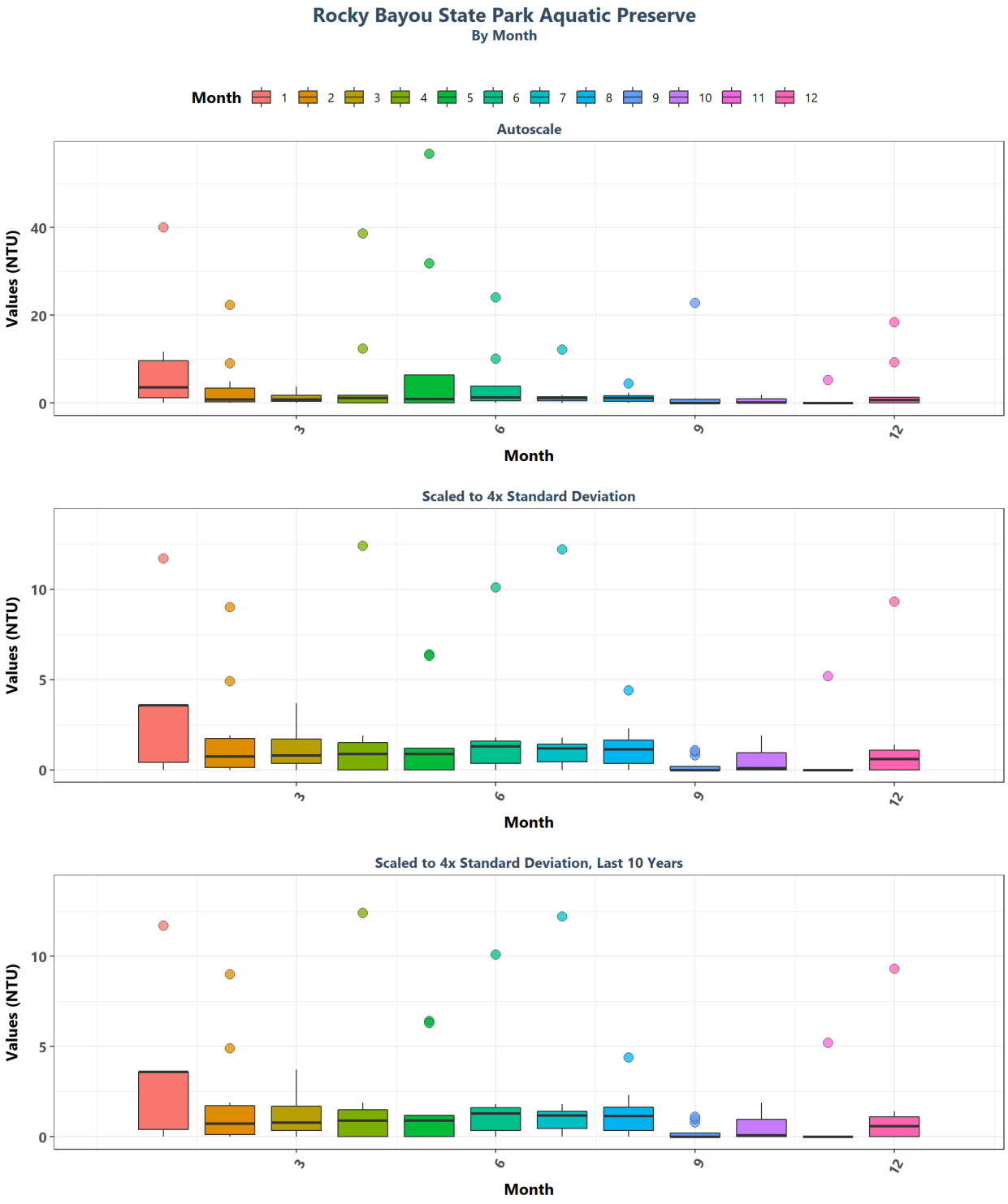

Pinellas County Aquatic Preserve
By Year
Autoscale
Values (NTU)
0
20
40
60
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
15
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
15
2011
2013
2015
2017
2019
2021
Year

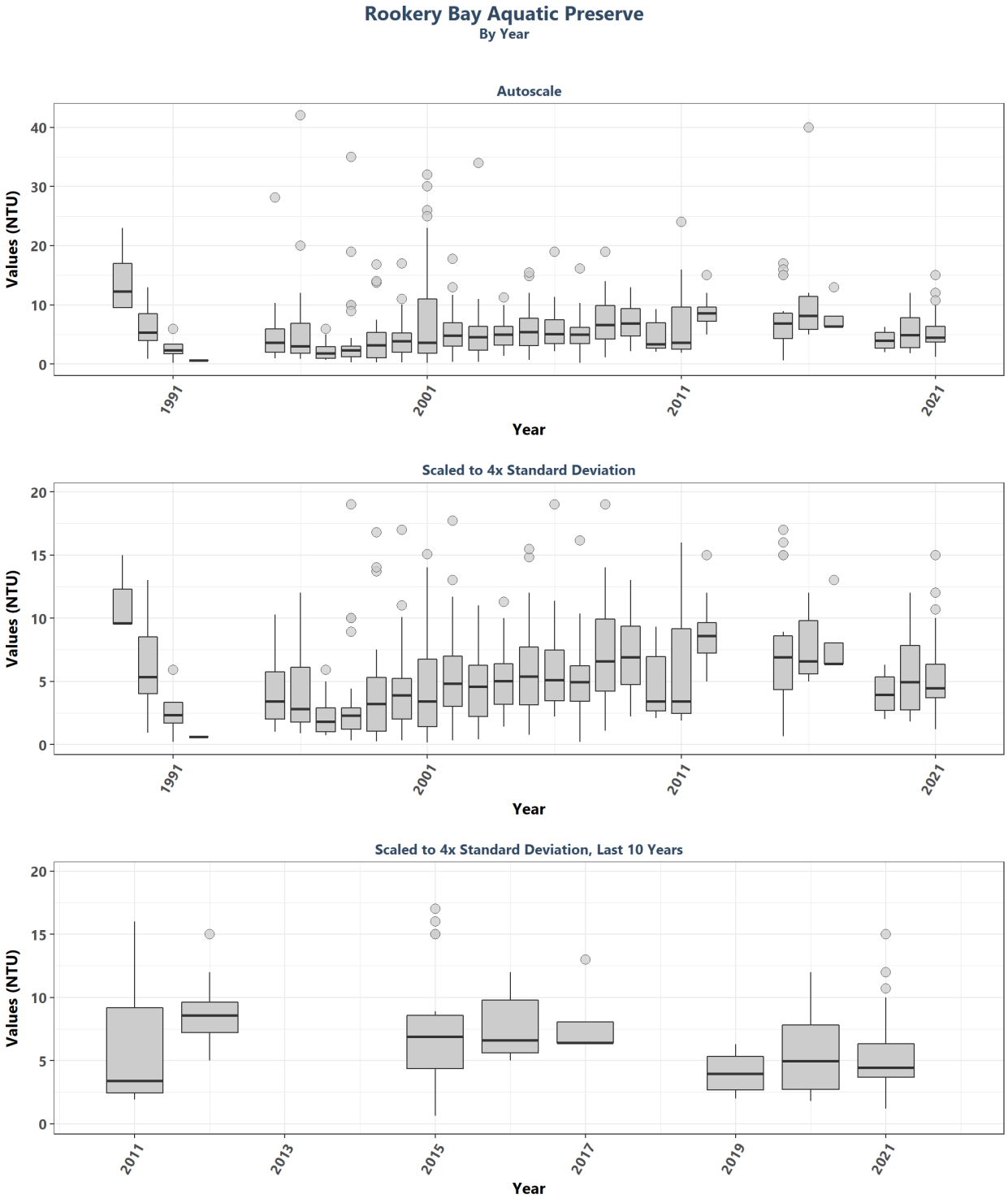

Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

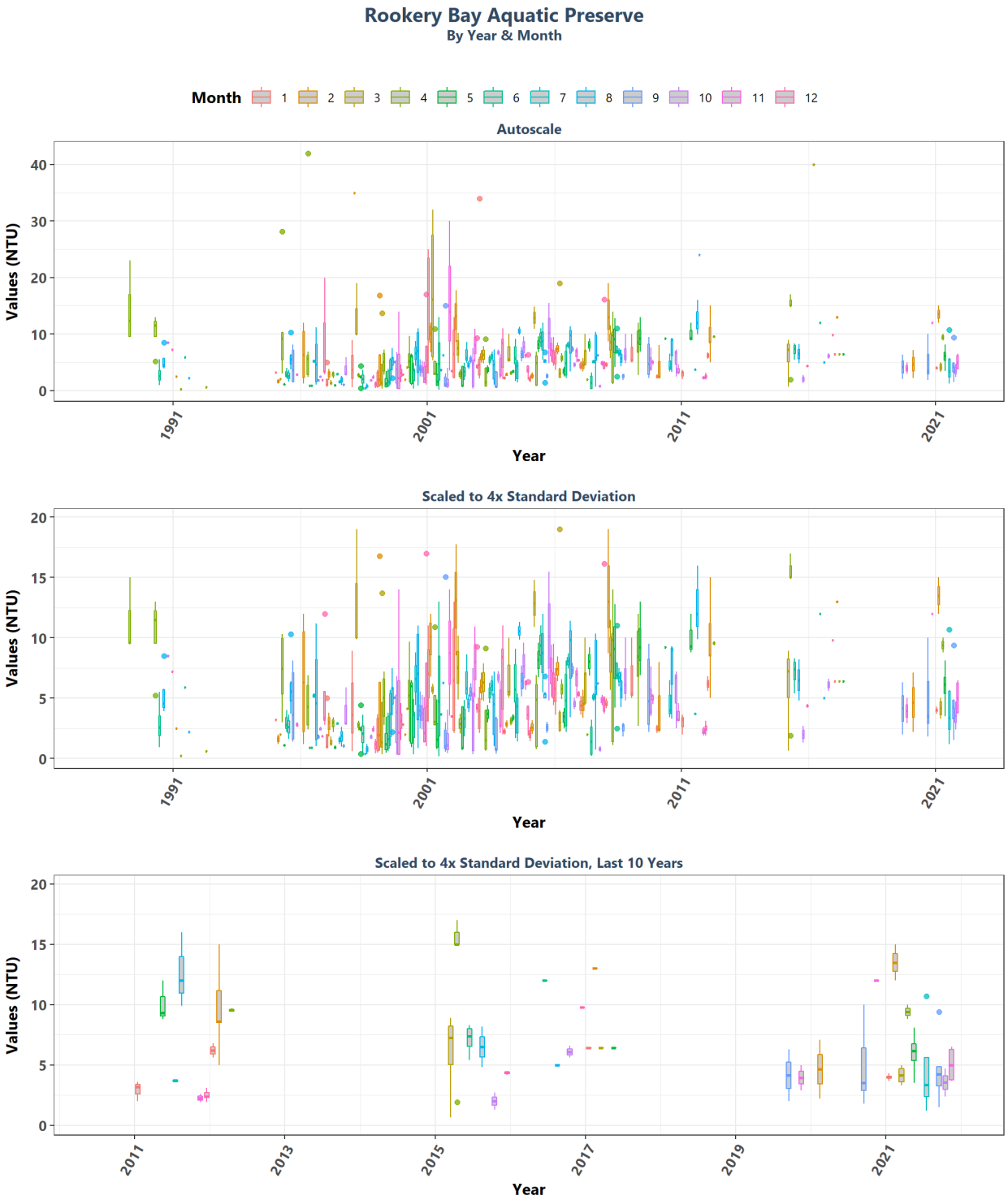

Pinellas County Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
20
40
60
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
15
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
15
3
6
9
12
Month

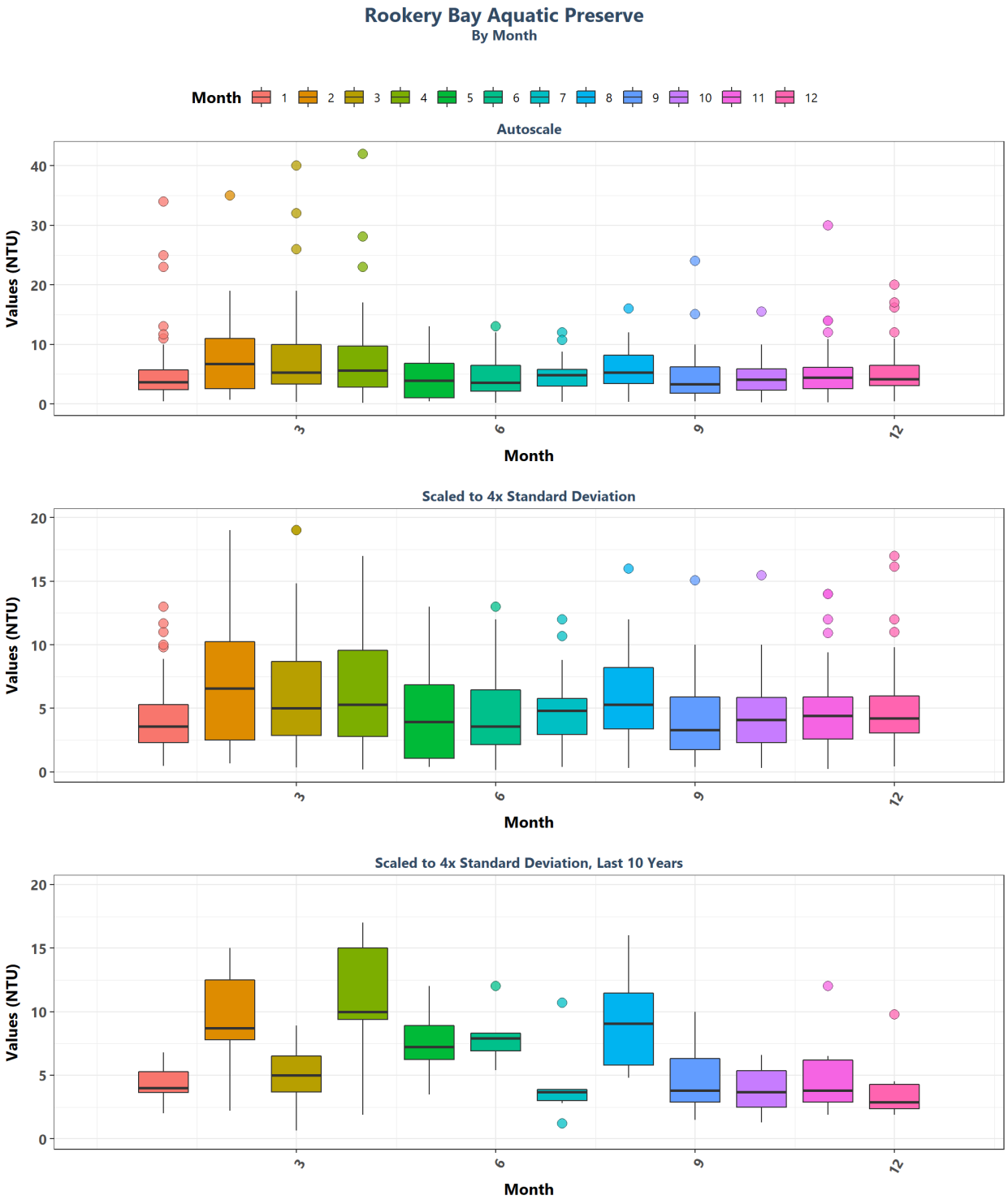

Rocky Bayou State Park Aquatic Preserve
By Year
Autoscale
Values (NTU)
0
20
40
2008
2013
2018
Year
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
2008
2013
2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
2008
2010
2012
2014
2016
2018
Year

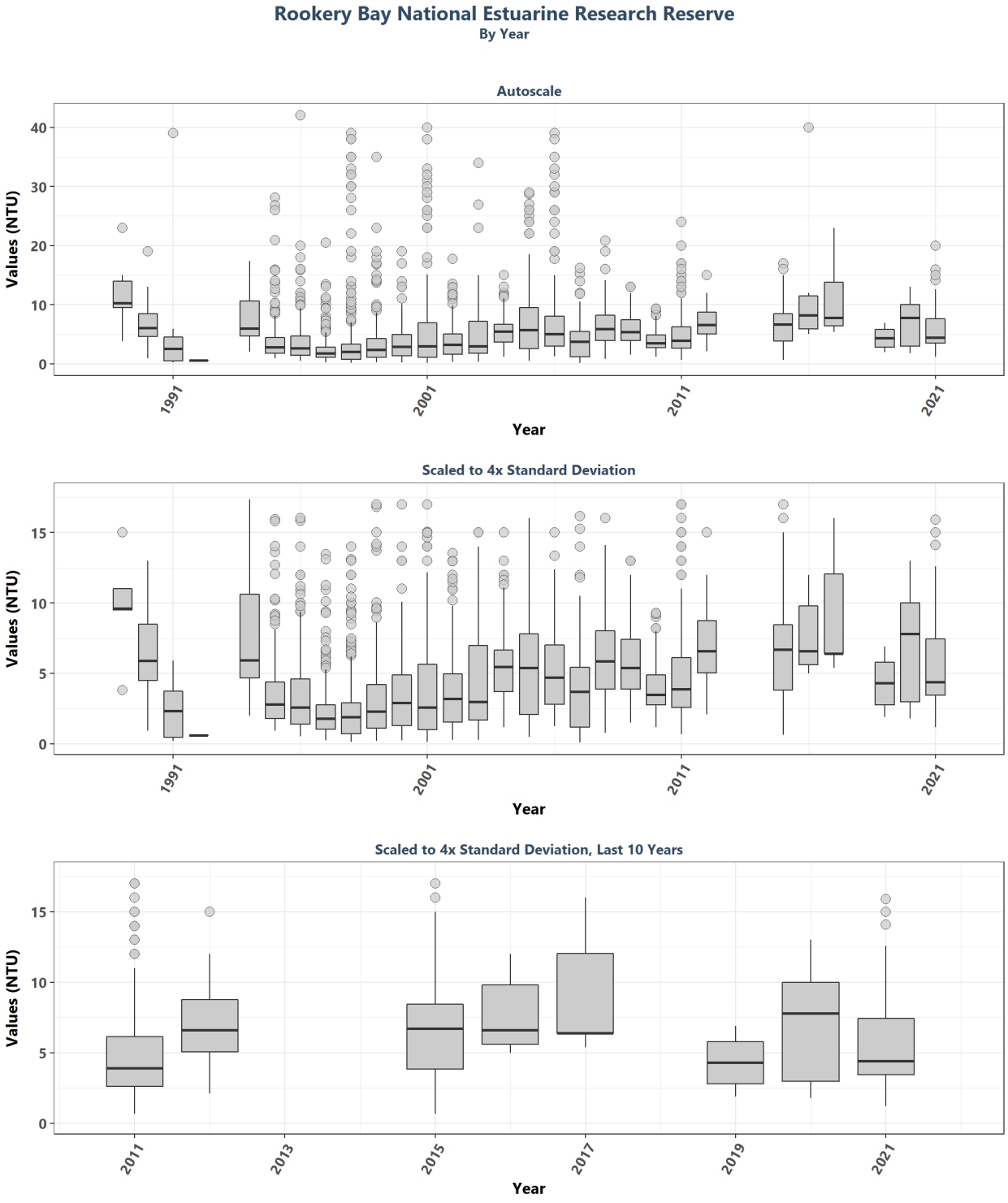

Rocky Bayou State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0
20
40
2008
2013
2018
Year
Scaled to 4x Standard Deviation
Values (NTU)
0
5
10
2008
2013
2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0
5
10
2008
2010
2012
2014
2016
2018
Year

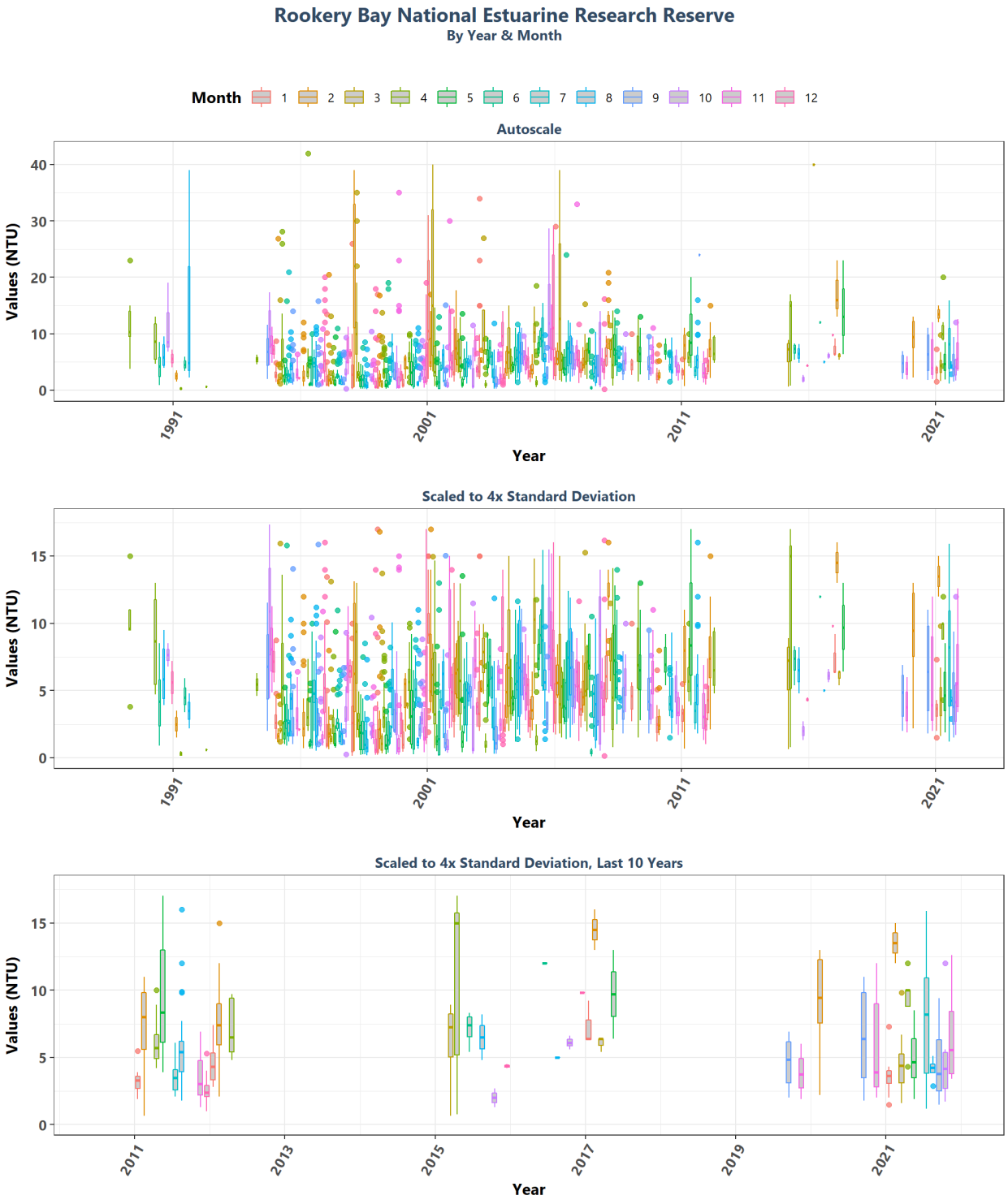

Rocky Bayou State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
40
20
0
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (NTU)
10
5
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
10
5
0
3
6
9
12
Month

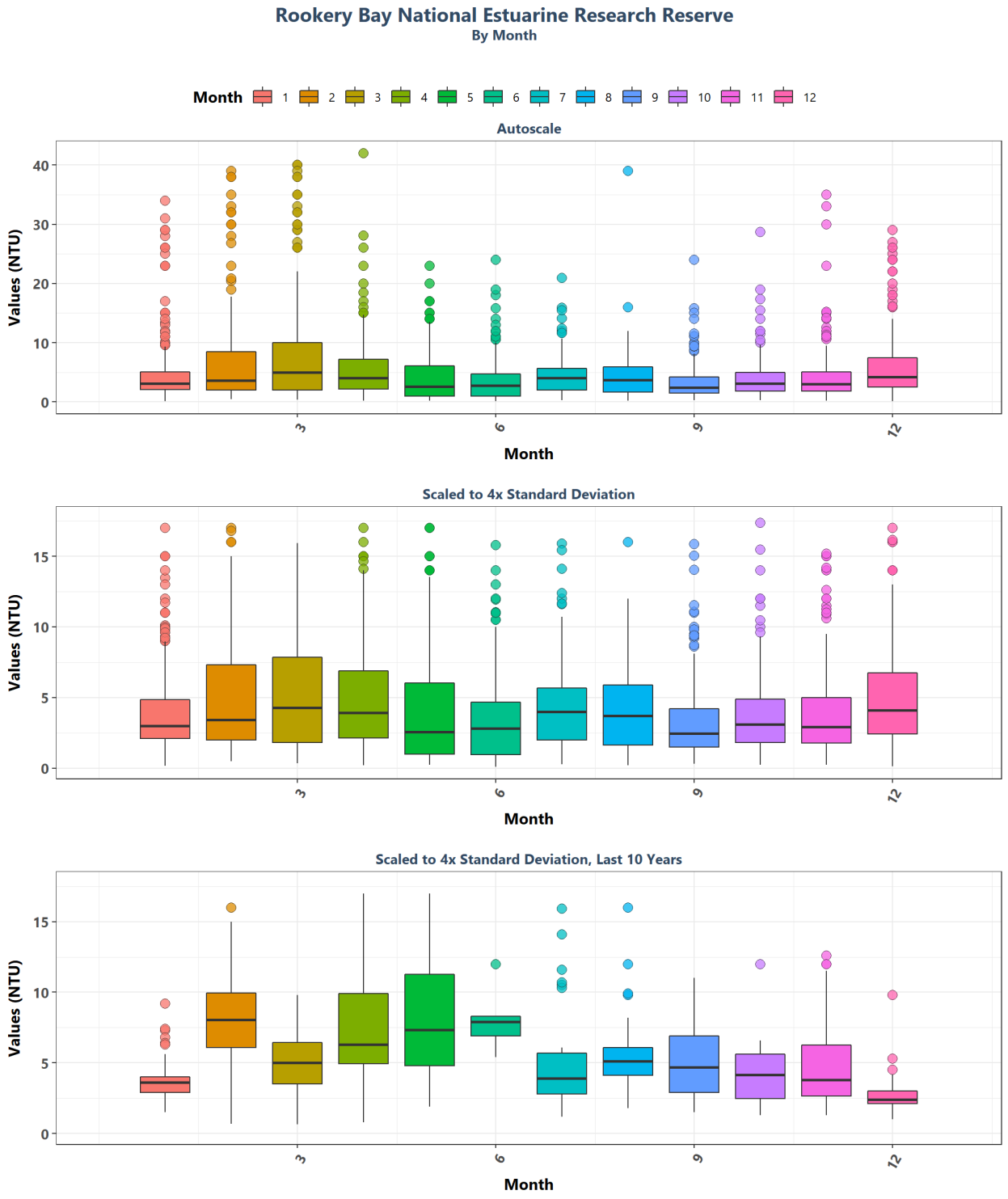

Rookery Bay Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year


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

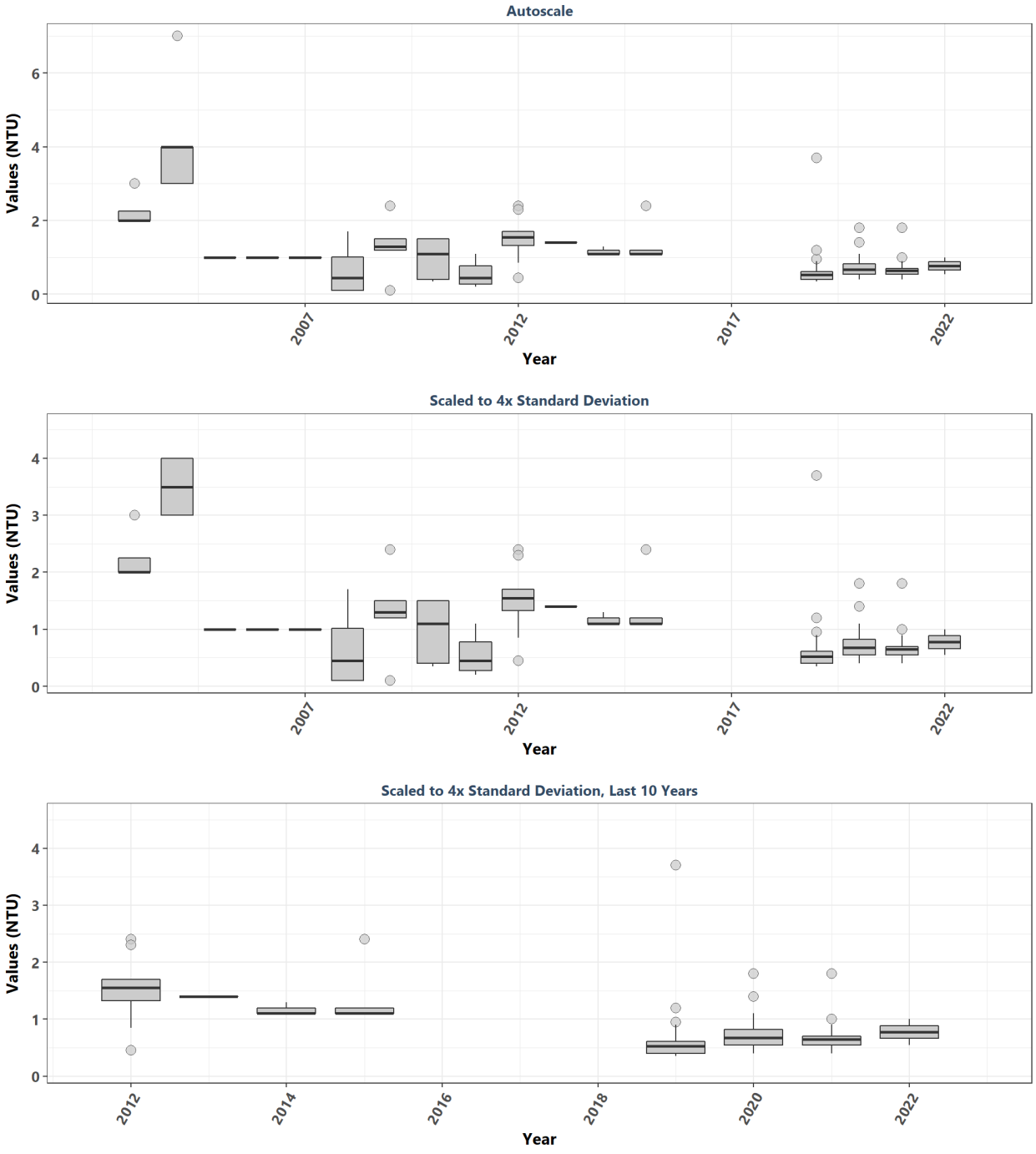

Rookery Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

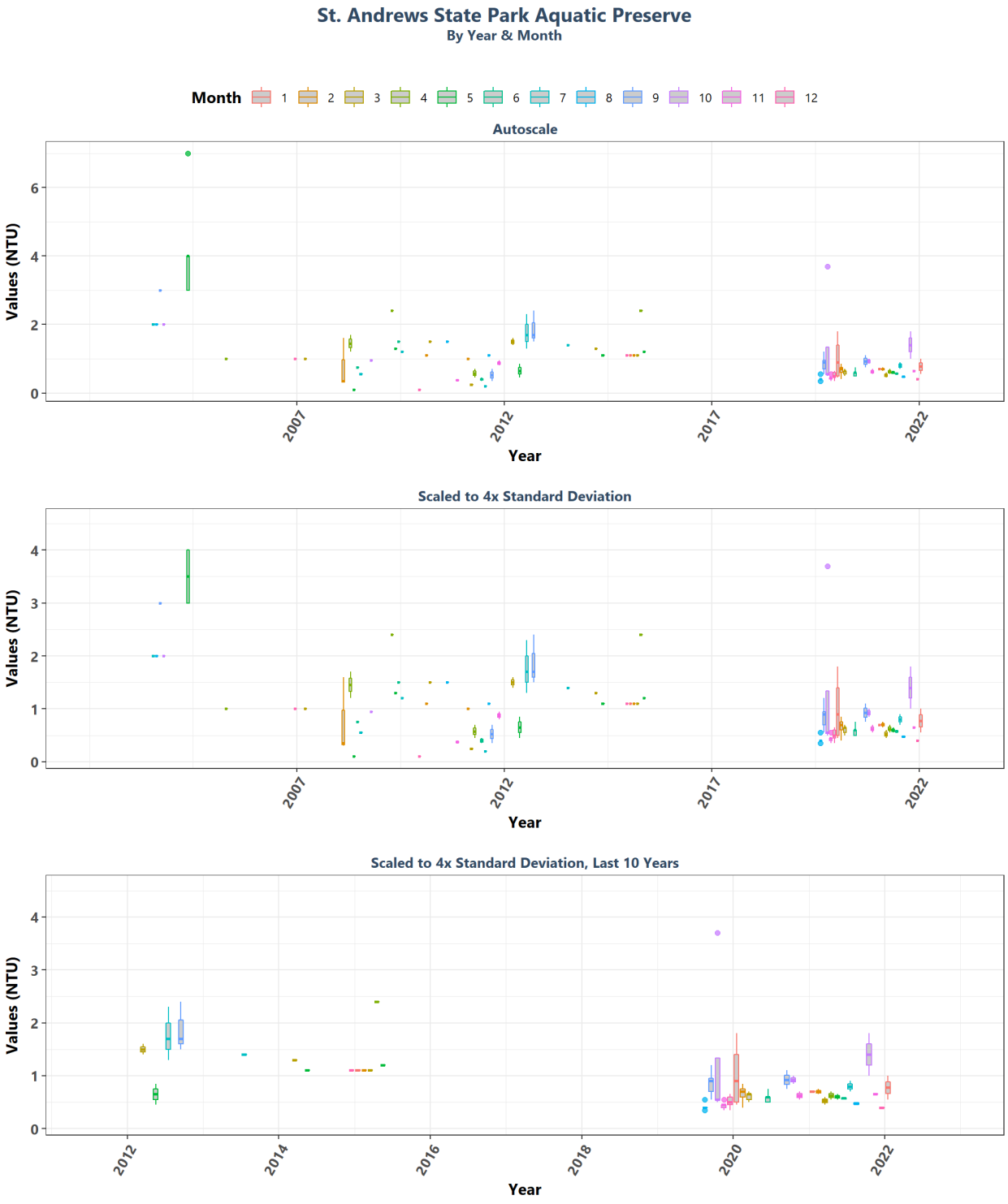

Rookery Bay National Estuarine Research Reserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

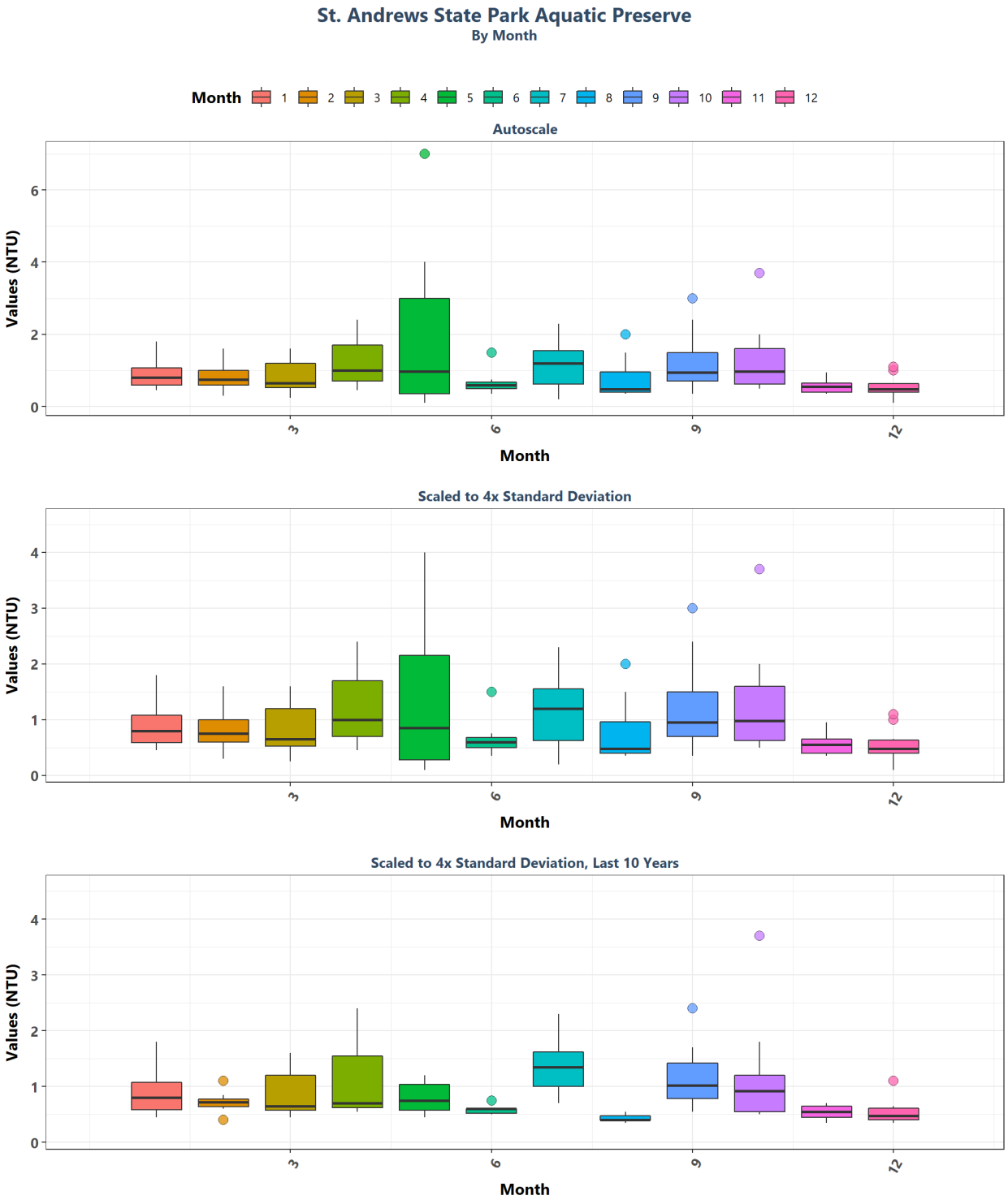

Rookery Bay National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year


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

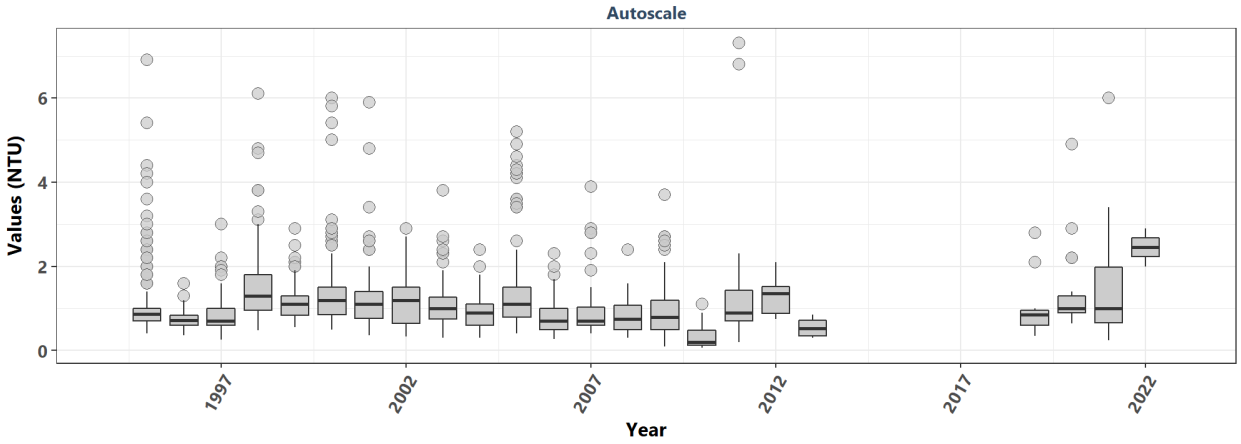
St. Andrews State Park Aquatic Preserve
By Year

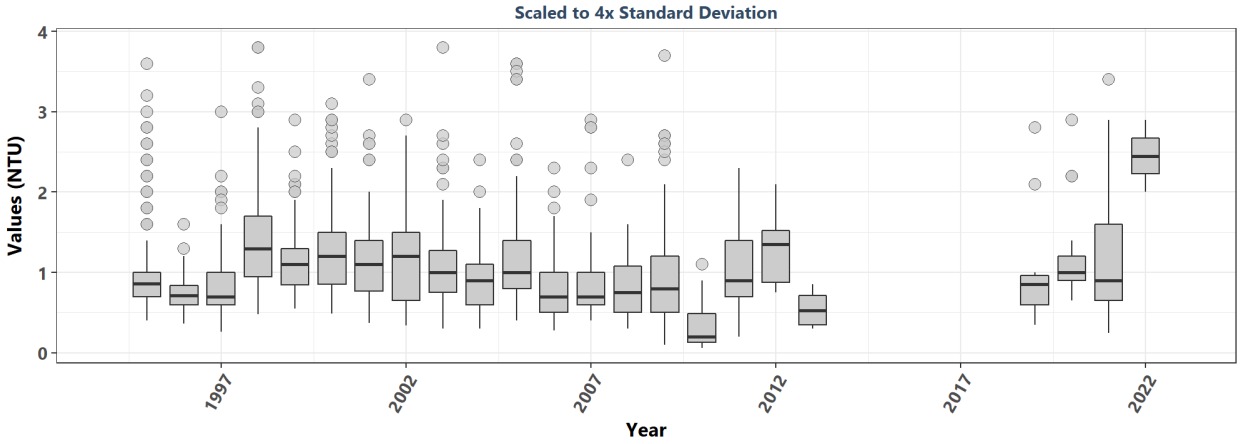


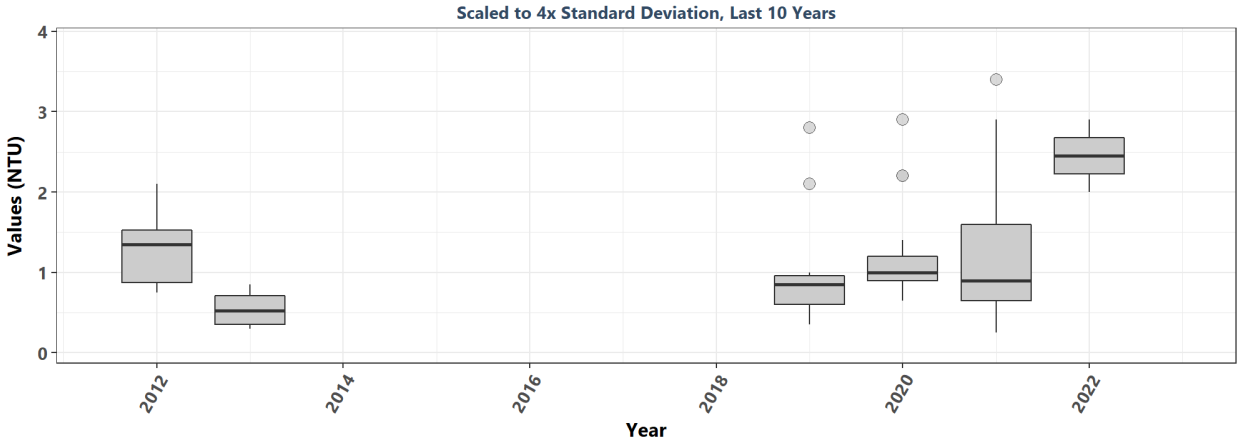

St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

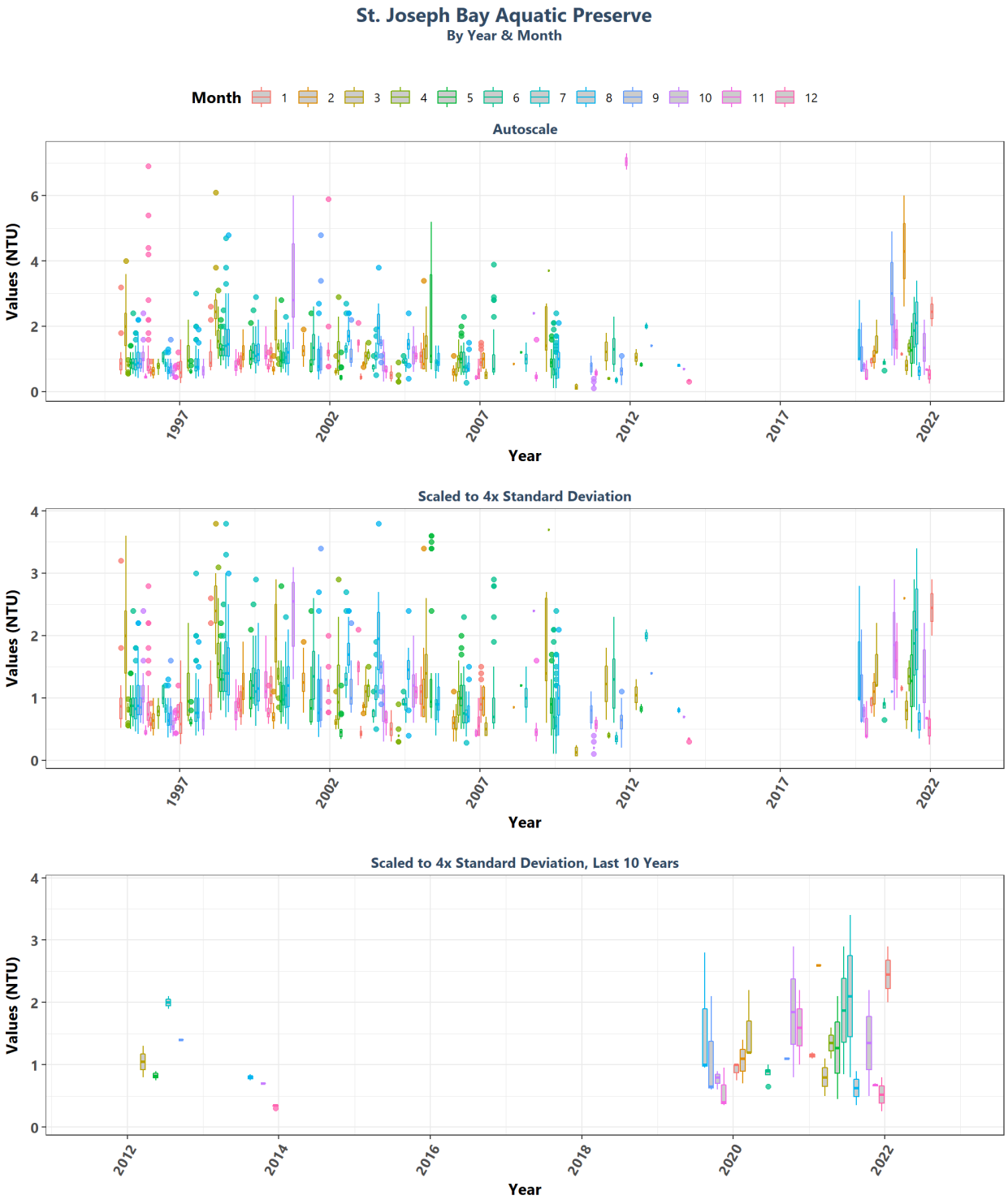

St. Andrews State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

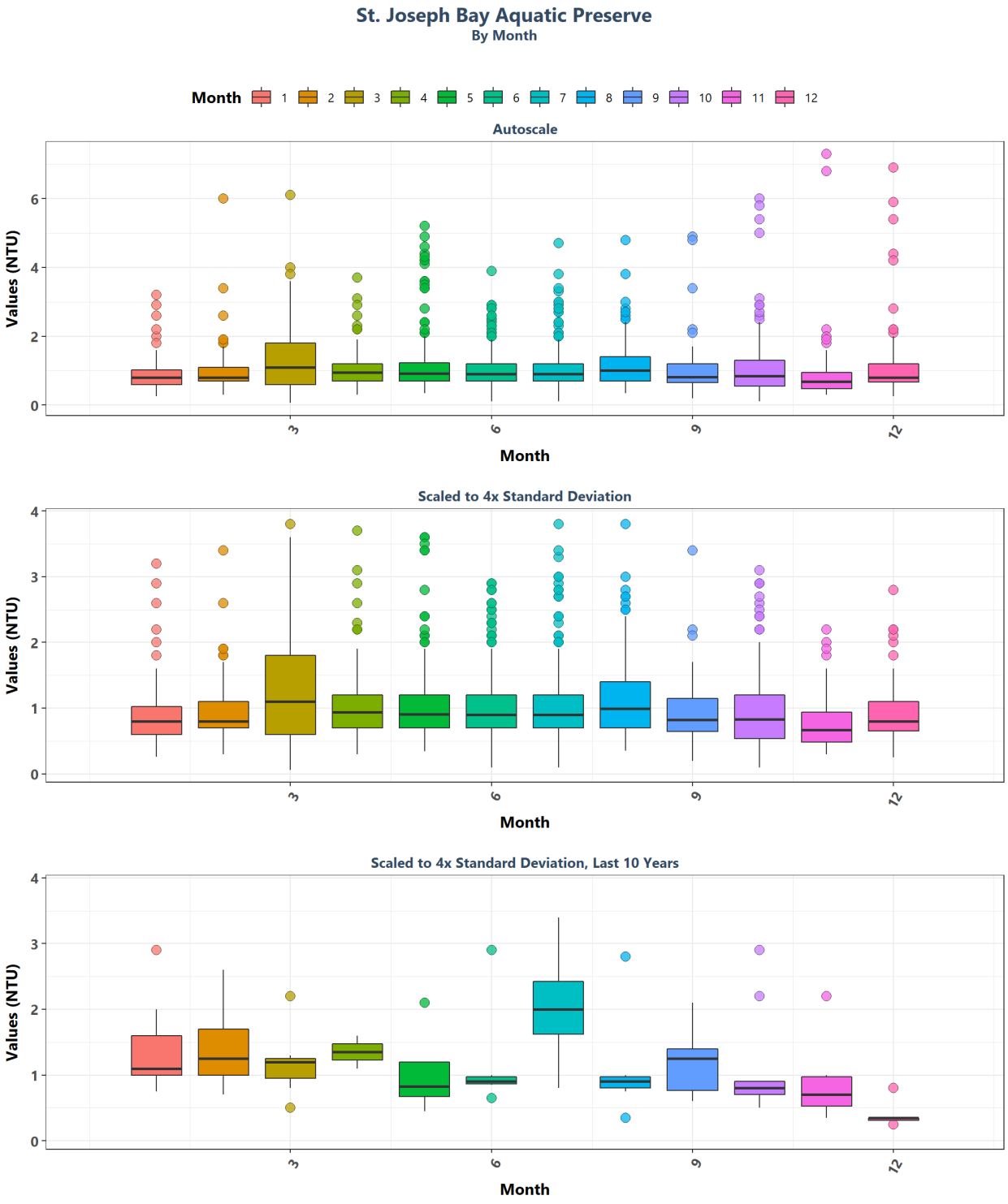
St. Joseph Bay Aquatic Preserve
By Year



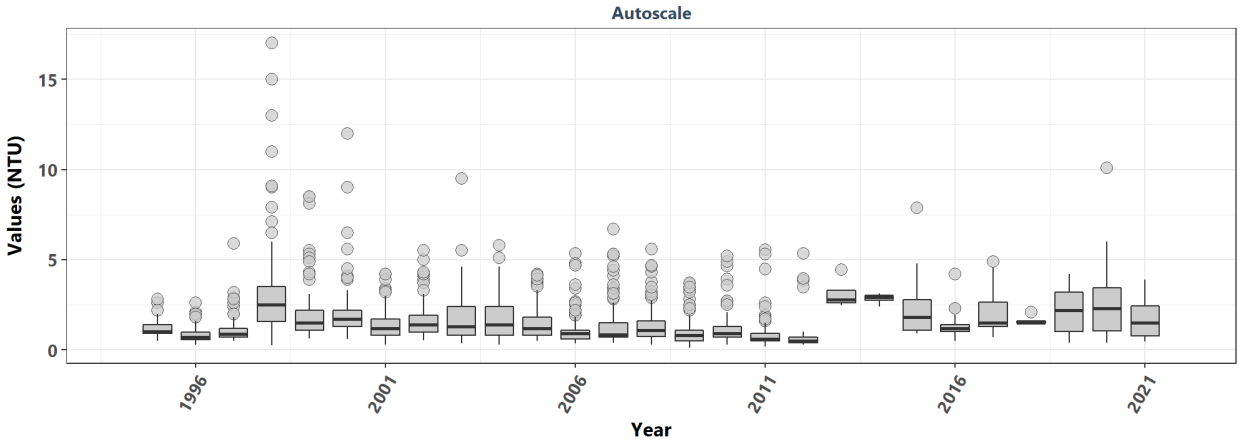


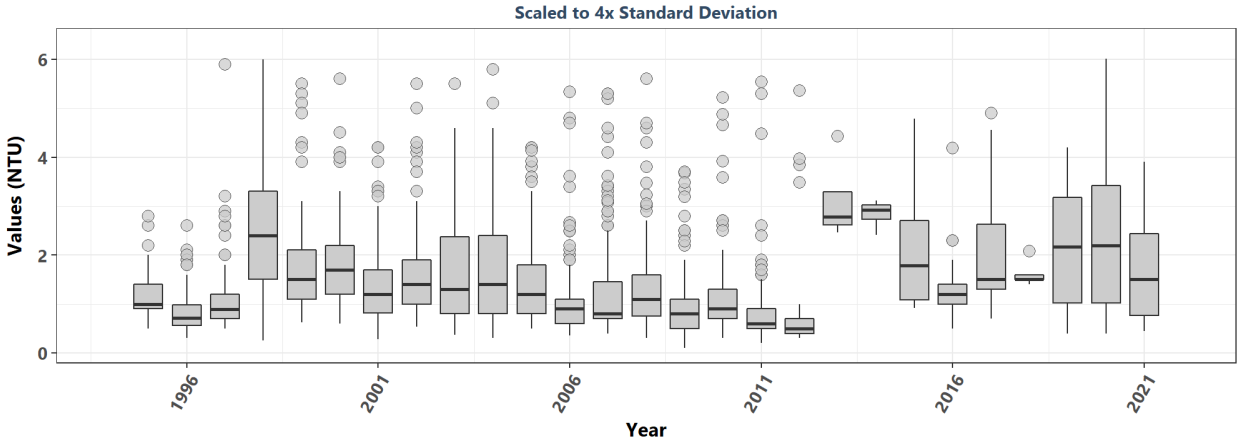


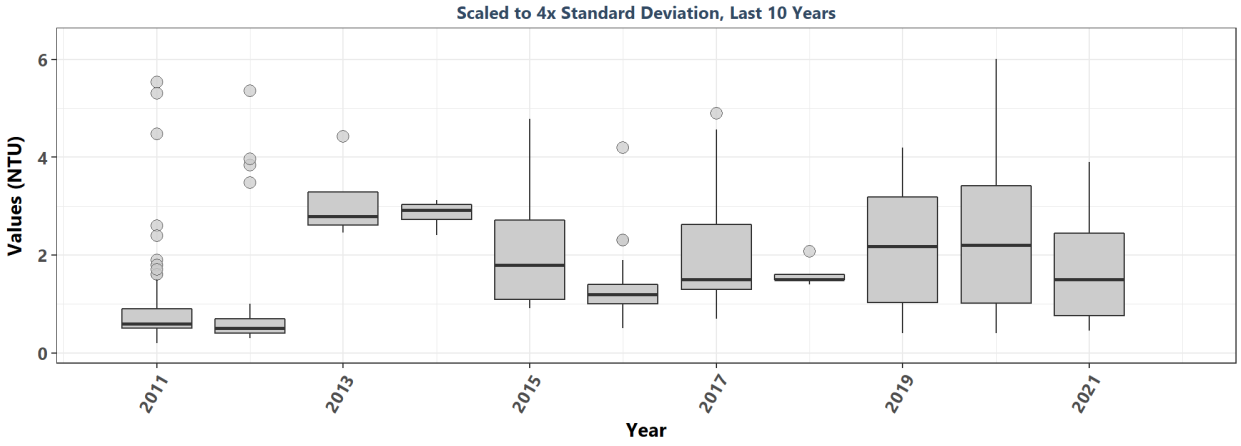

St. Joseph Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

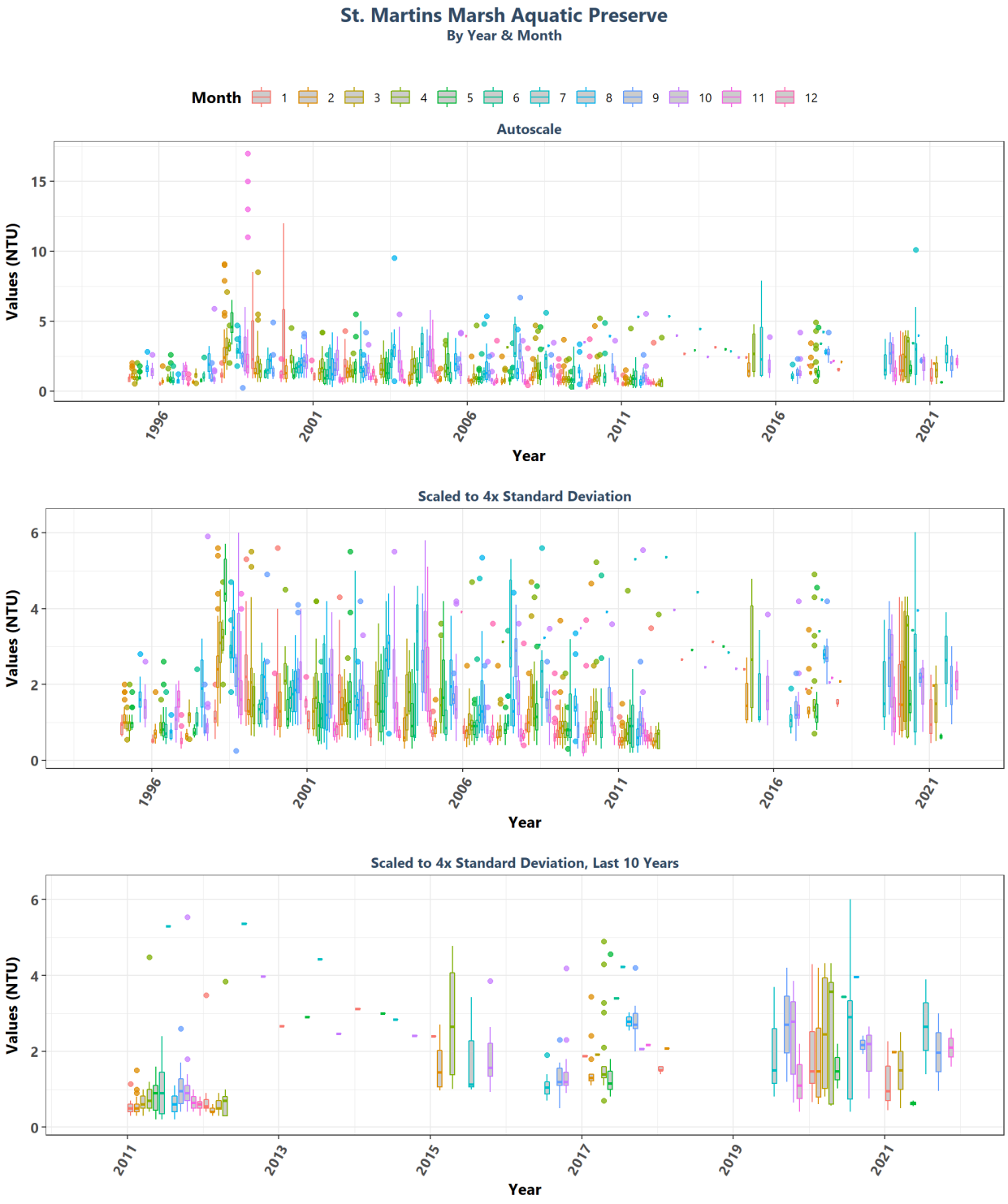

St. Joseph Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
0 2 4 6
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (NTU)
0 1 2 3 4
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
0 1 2 3 4
3 6 9 12
Month

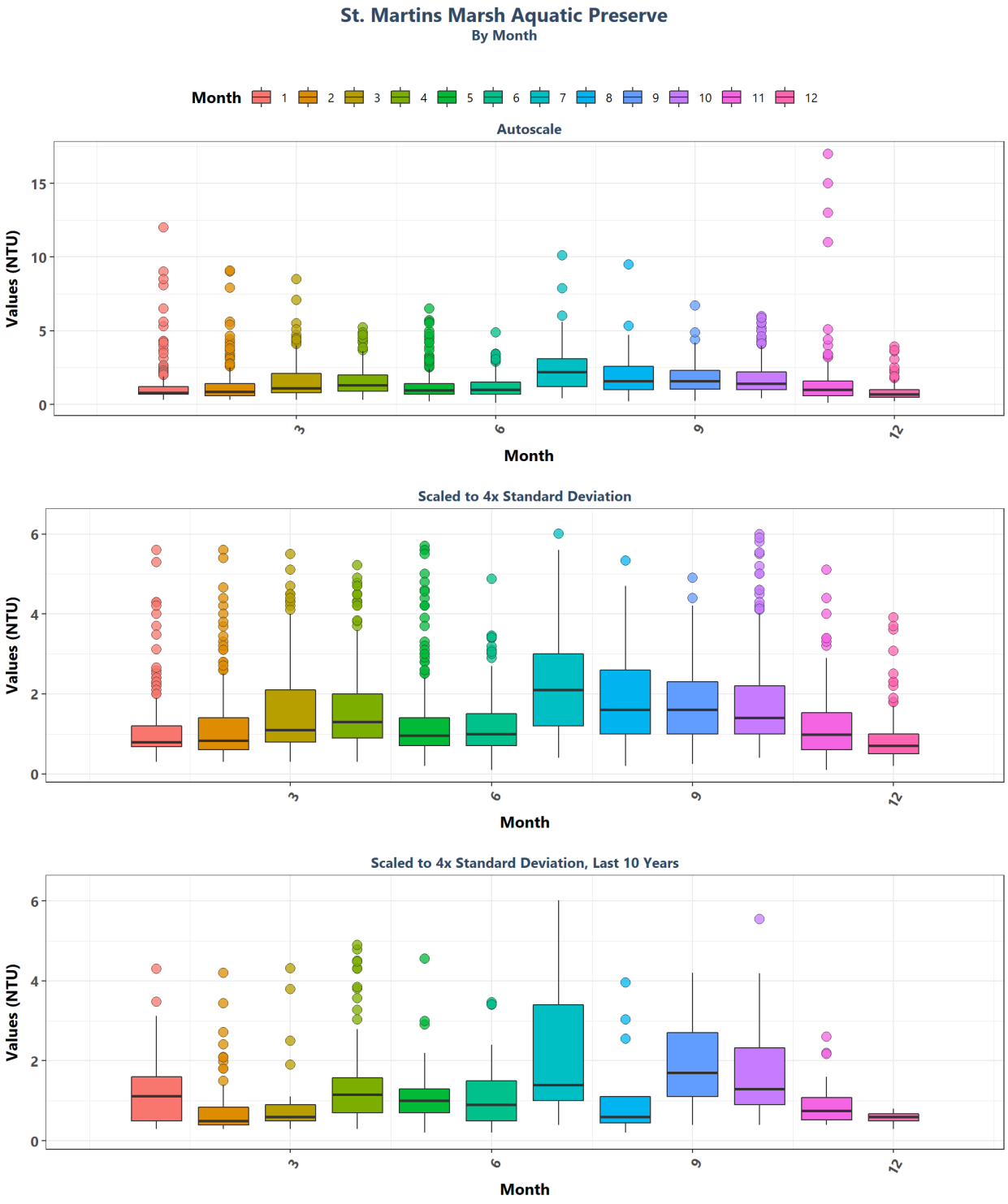
St. Martins Marsh Aquatic Preserve
By Year





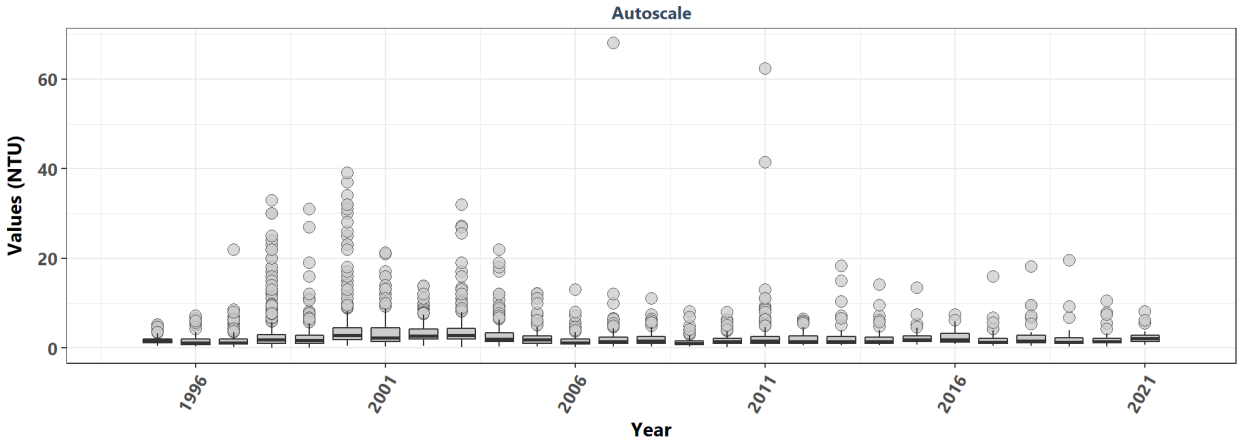


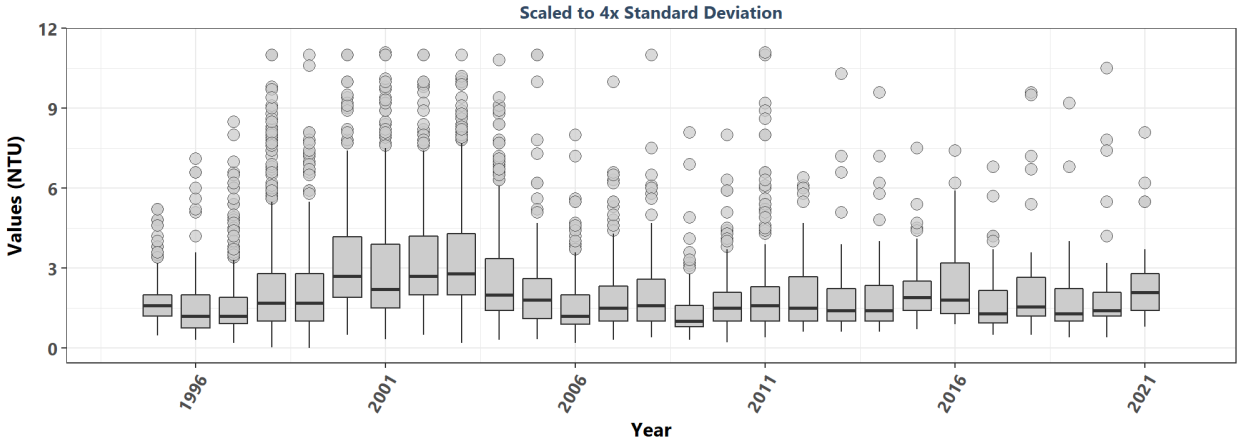

St. Martins Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

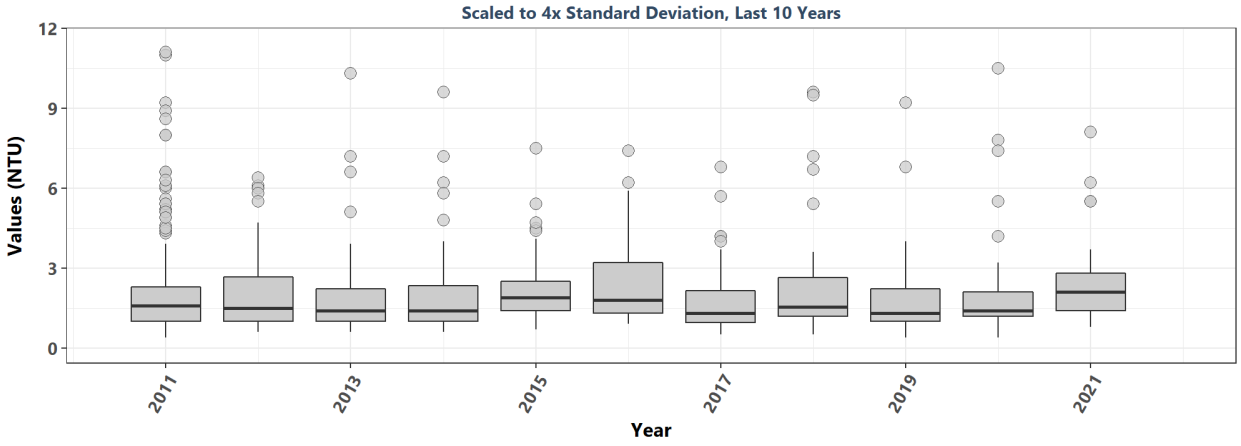

St. Martins Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

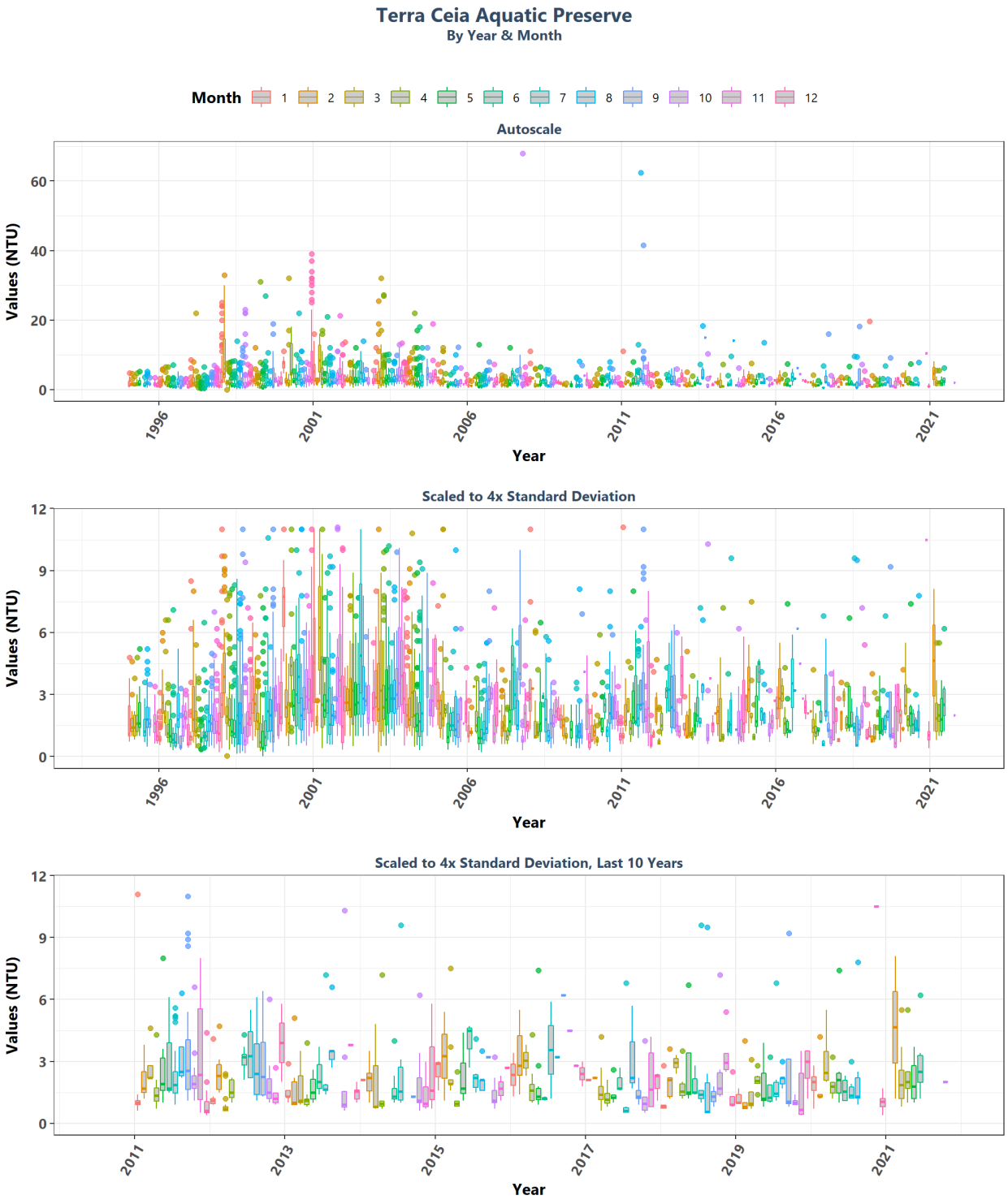
Terra Ceia Aquatic Preserve

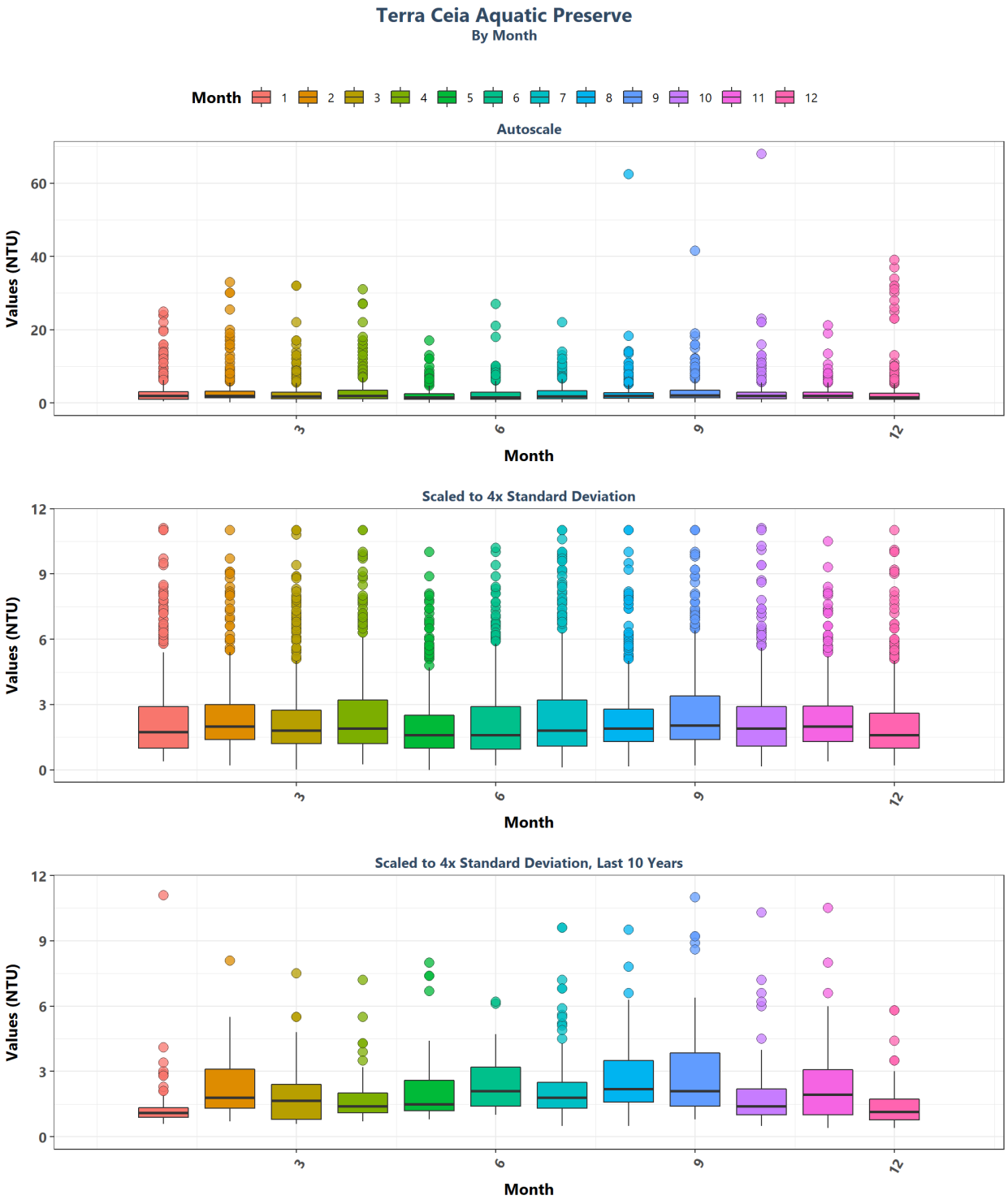
By Year

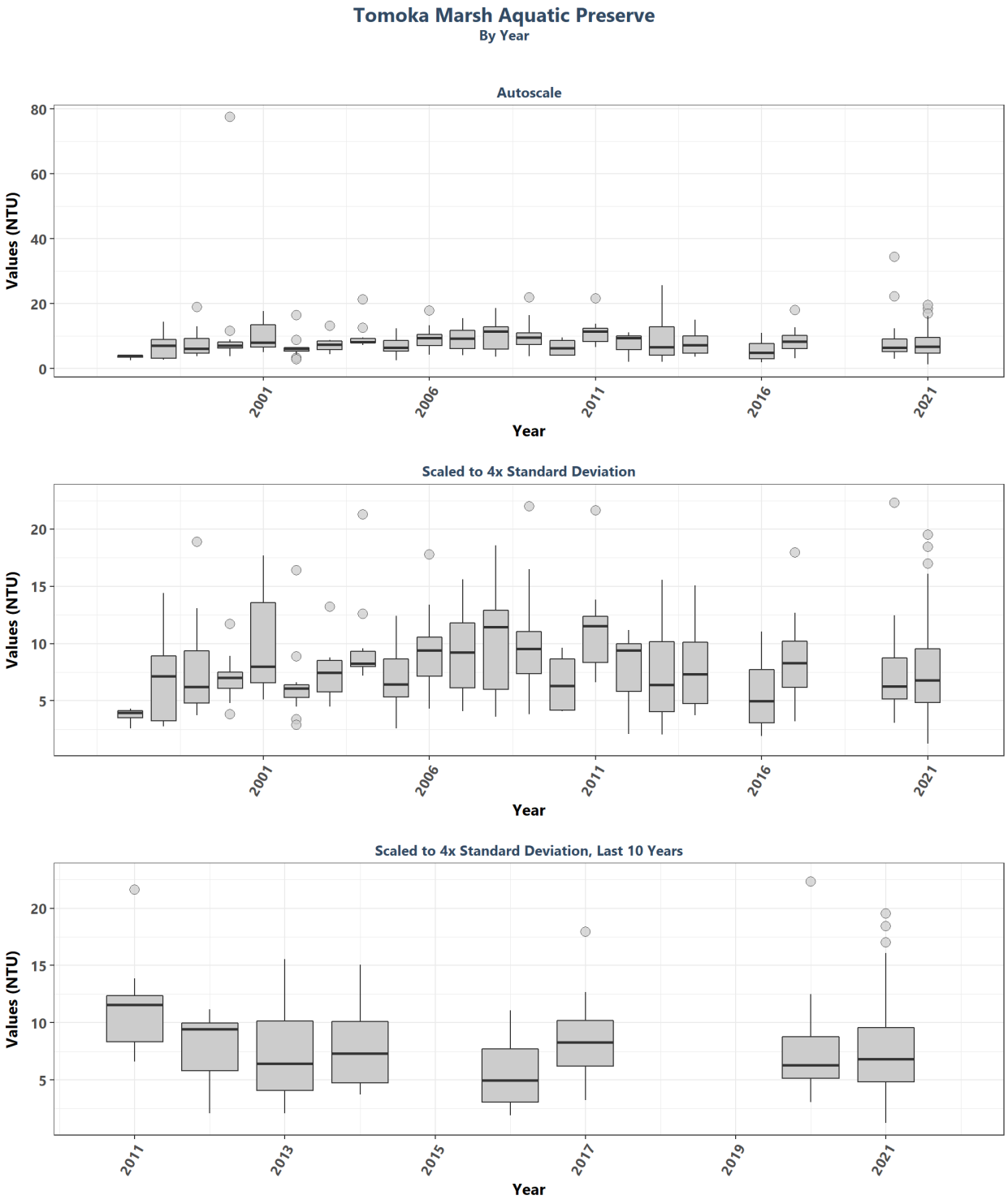


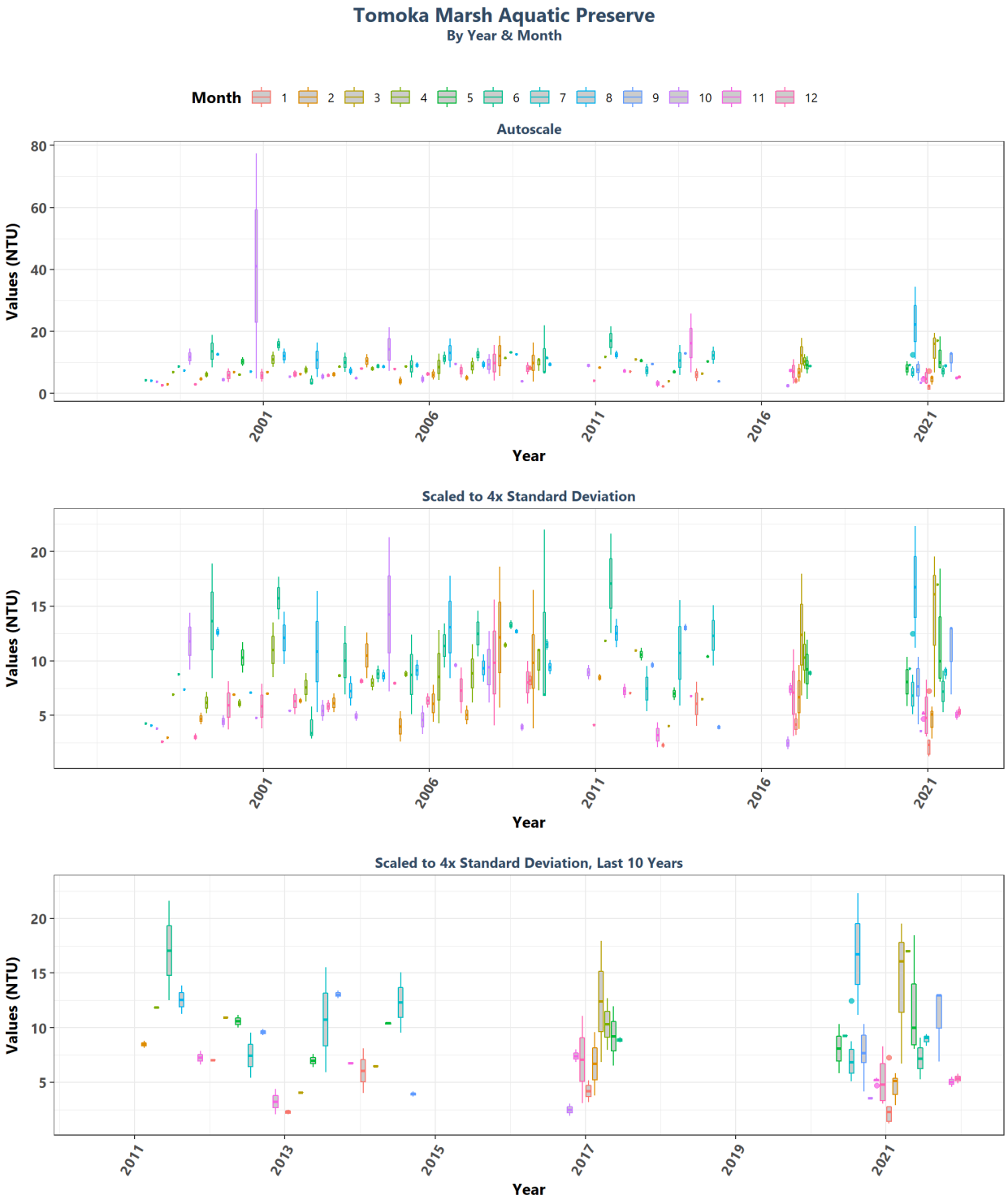


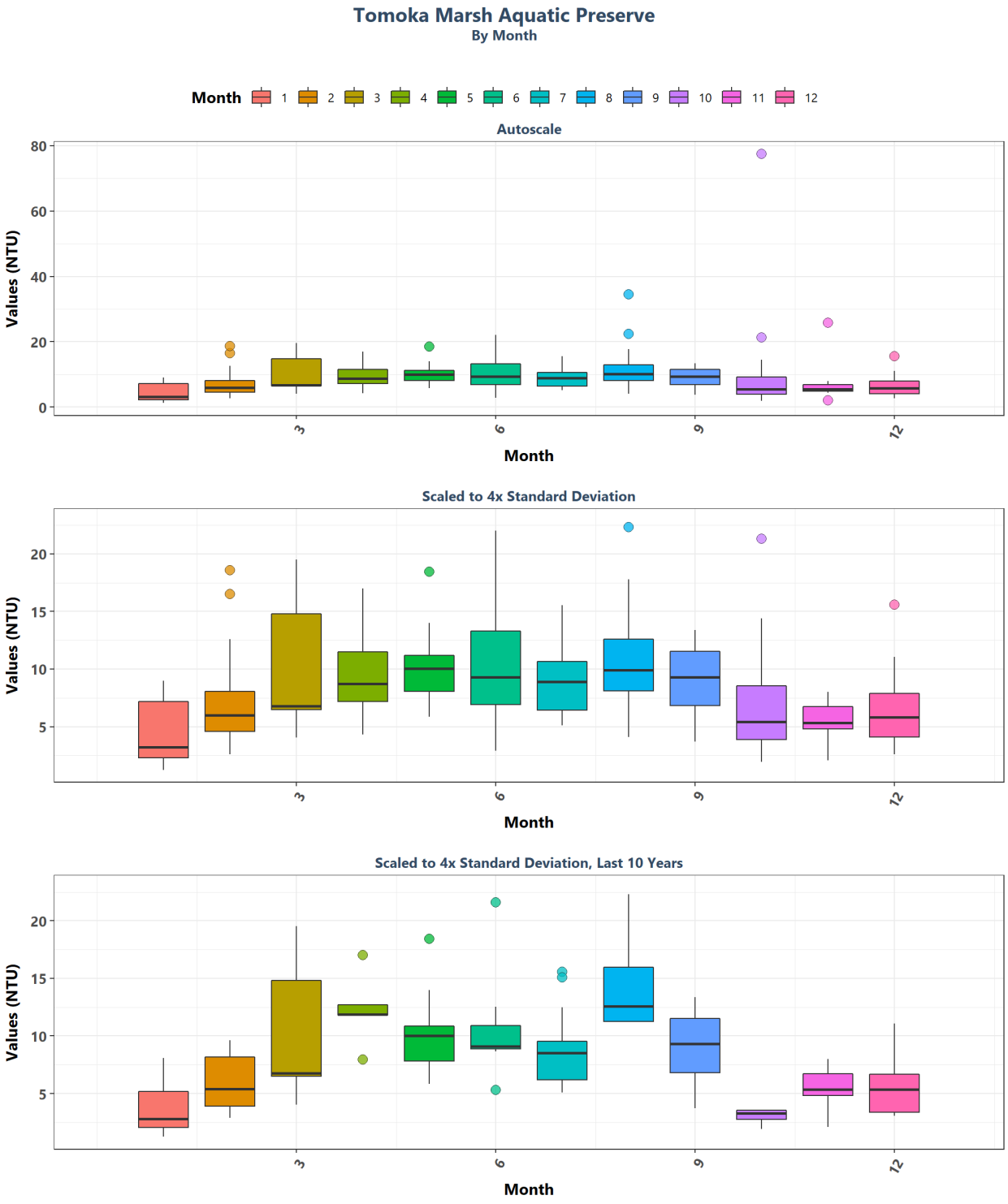



Terra Ceia Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year


Terra Ceia Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

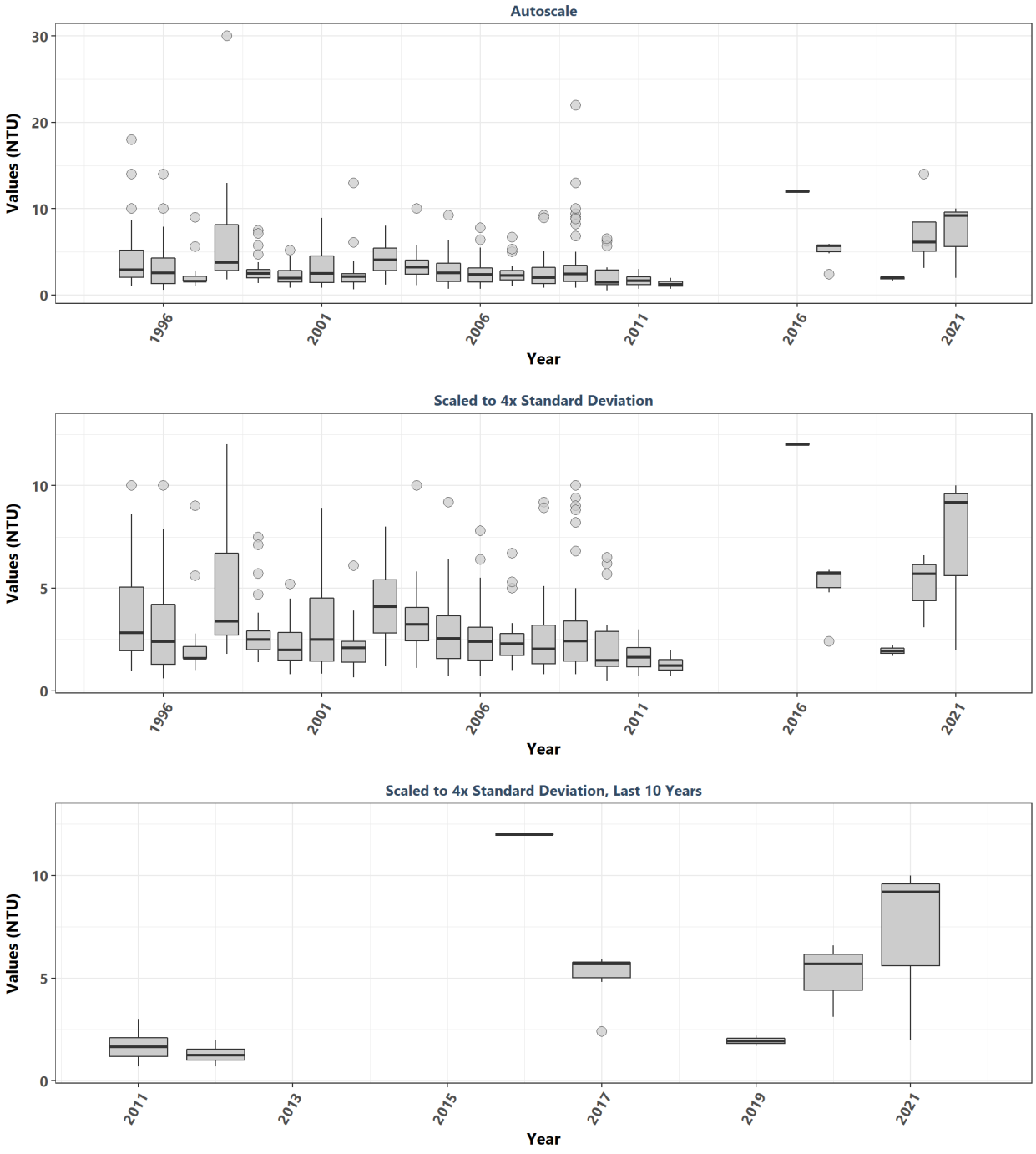

Tomoka Marsh Aquatic Preserve
By Year
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

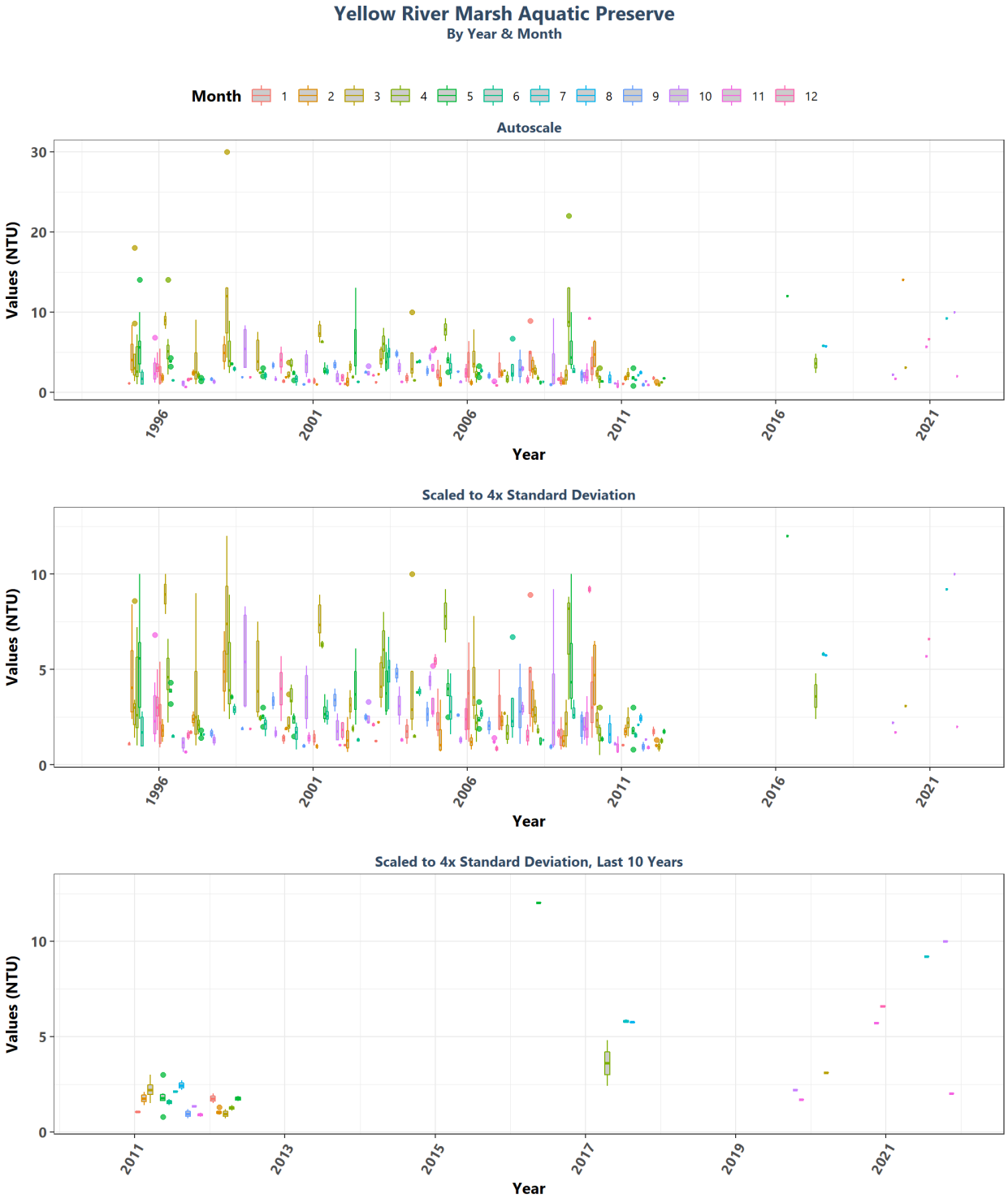

Tomoka Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year

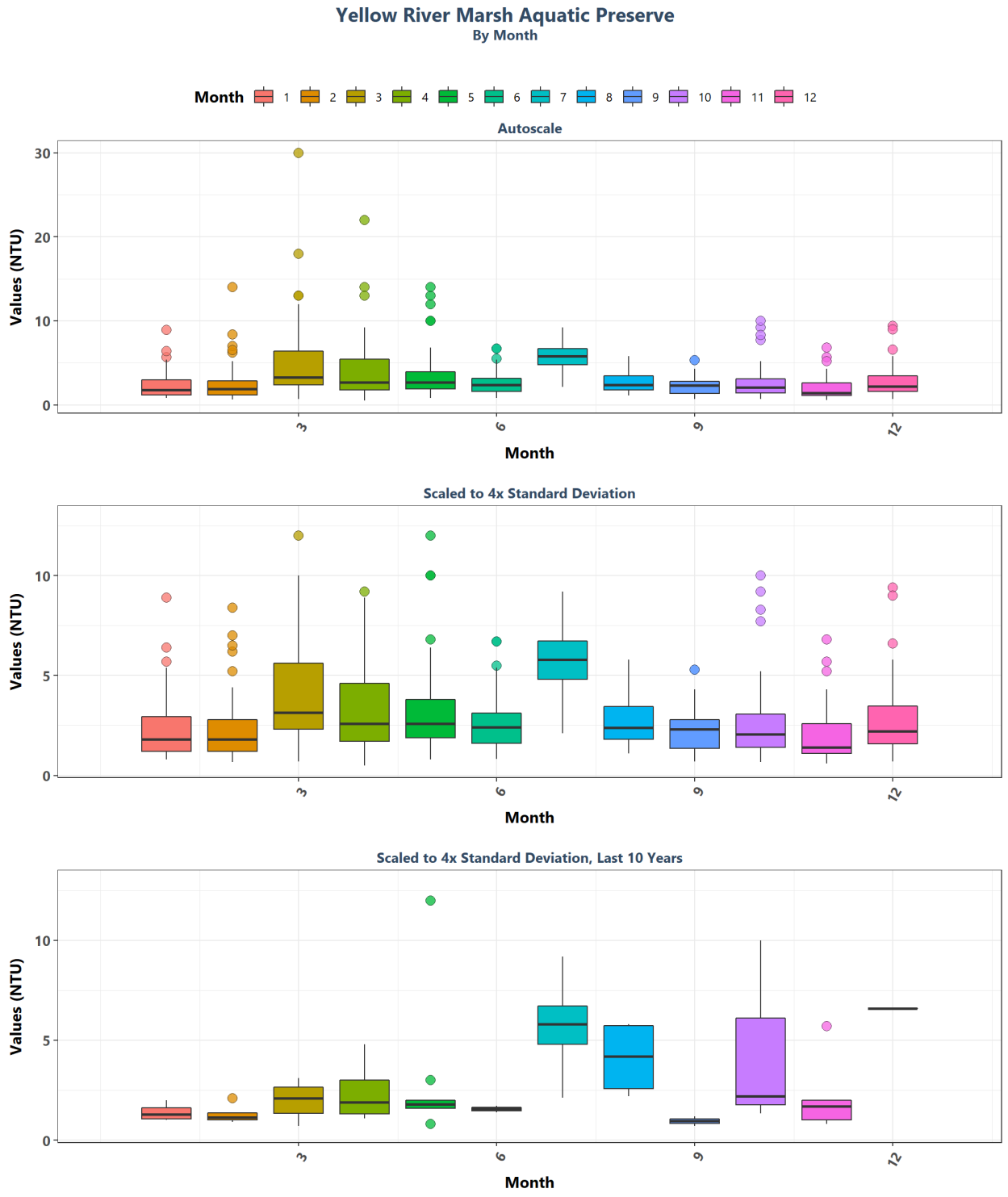

Tomoka Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Month
Scaled to 4x Standard Deviation
Values (NTU)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Month

Yellow River Marsh Aquatic Preserve

By Year




Yellow River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (NTU)
Year
Scaled to 4x Standard Deviation
Values (NTU)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (NTU)
Year




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