SEACAR Discrete Water Quality Analysis: Field Surface Salinity

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: 562906, Number Passed Filter: 562906
## I Codes: 0 (0%)
## Q Codes: 0 (0%)
## U Codes: 0 (0%)
```

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

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

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

Managed Area Statistics

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

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

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

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

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

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

Monitoring Location Statistics

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

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

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

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

Seasonal Kendall Tau Analysis

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

The following steps are performed:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

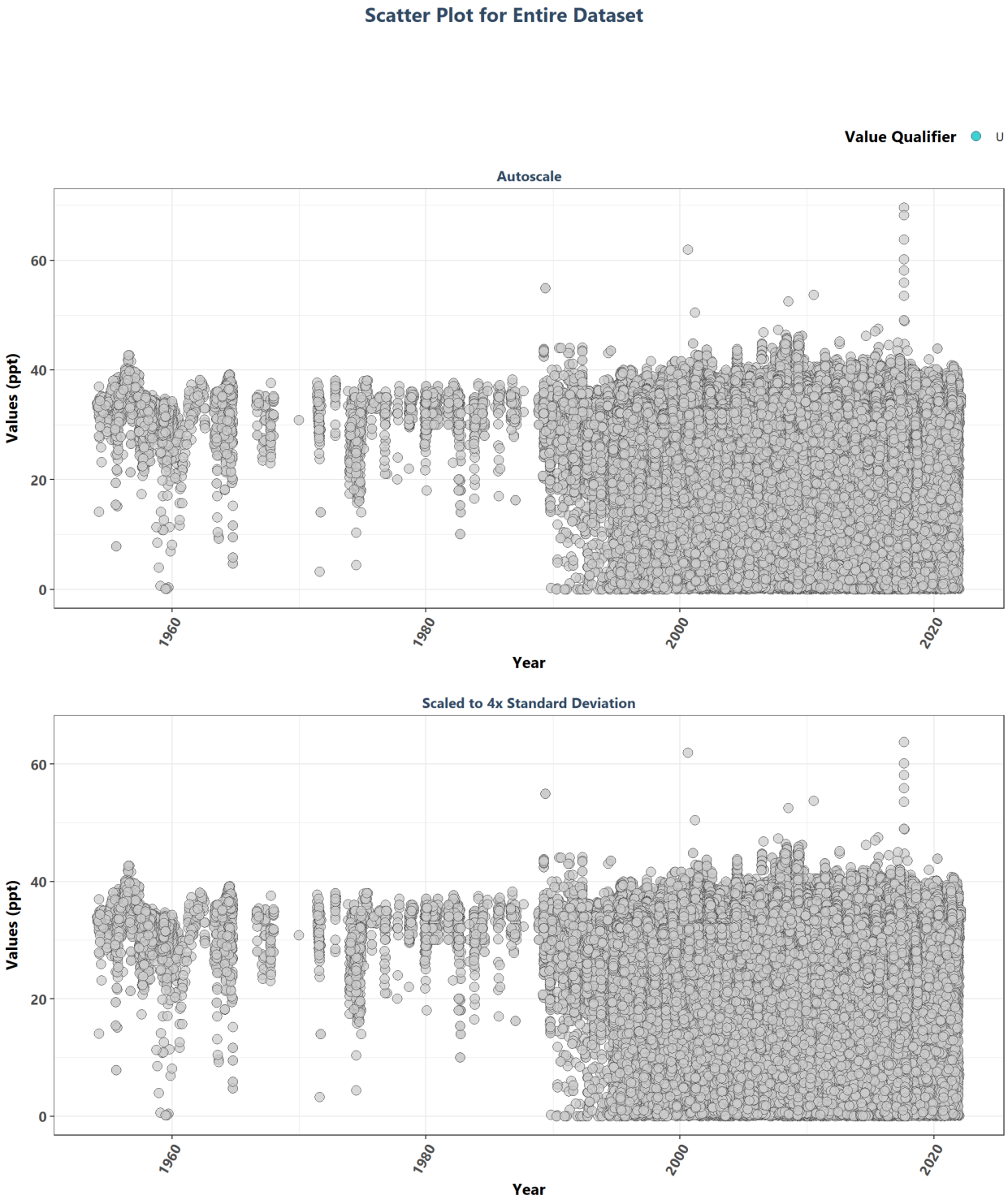
```
KT.Stats2$MonitoringID <- NULL
fwrite(KT.Stats2, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                         "_KendallTau_Stats.txt"), sep="|")
rm(KT.Stats2)
data <- data[!is.na(data$ResultValue),]
```

Appendix I: Scatter Plot of Entire Dataset

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

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

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


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

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


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

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

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

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



Appendix II: Dataset Summary Box Plots

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

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

This set of box plots are grouped by year.

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

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

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

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

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

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

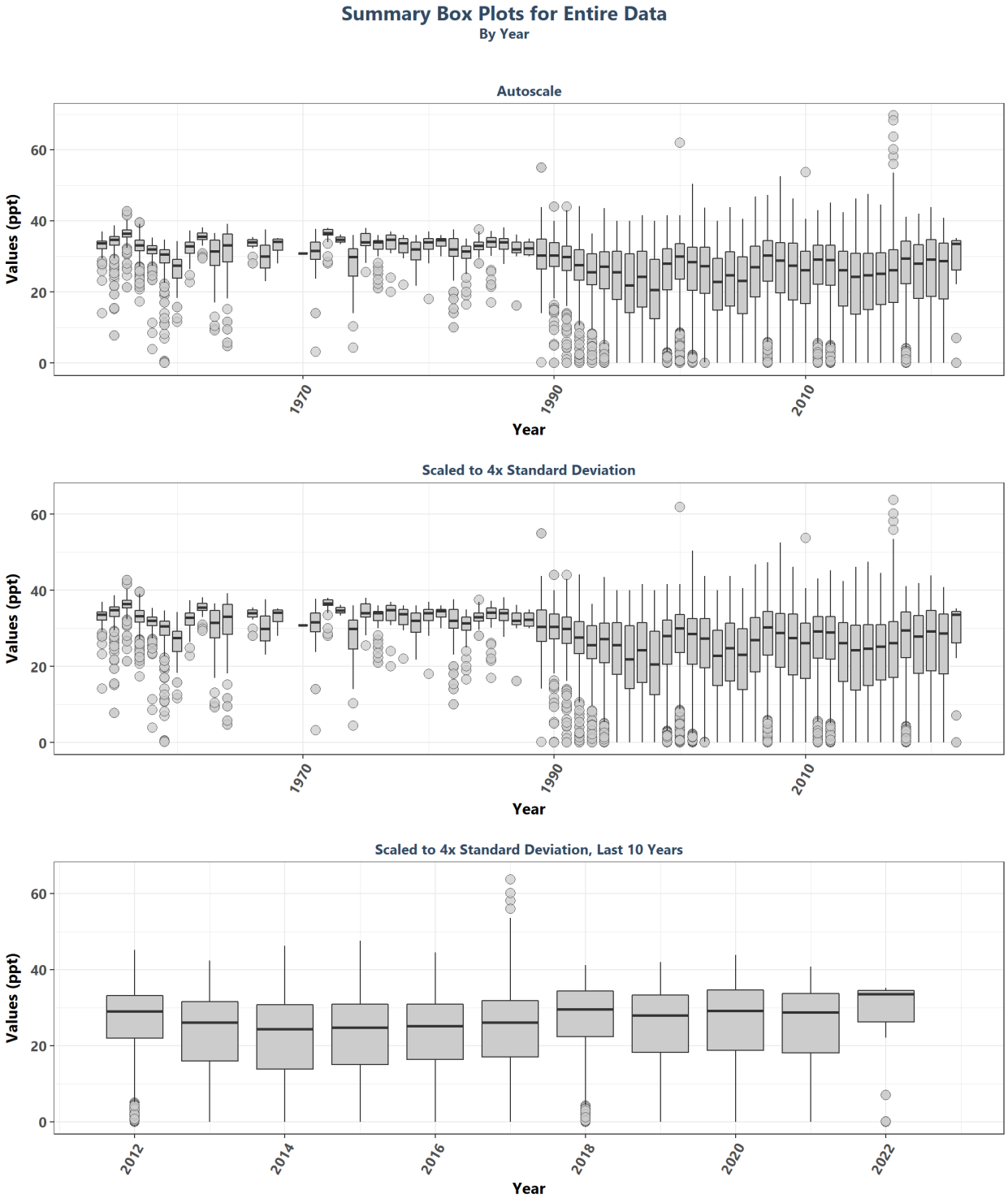
```
                      subtitle="By Year") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

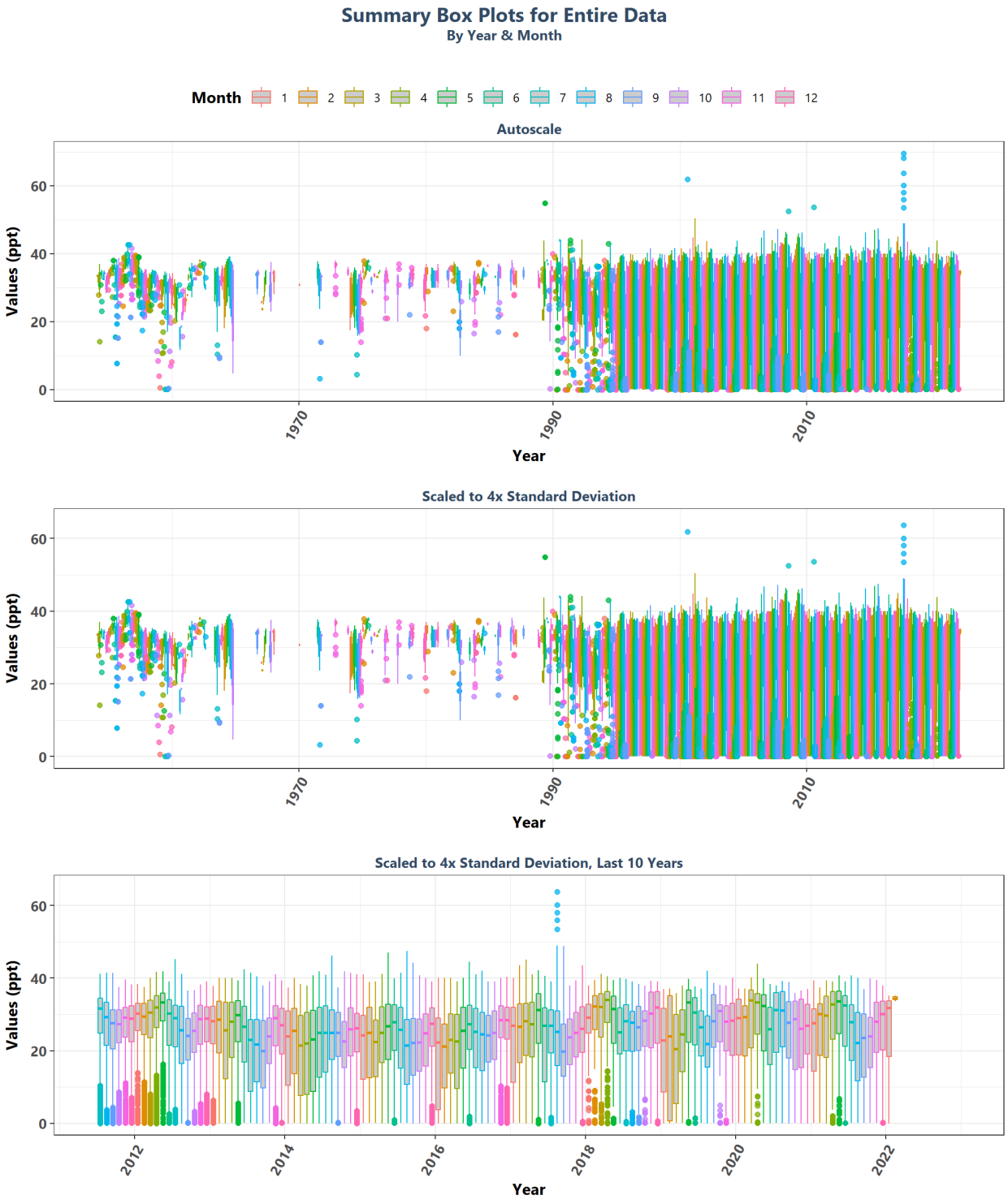
This set of box plots are grouped by year and month with the color being related to the month.

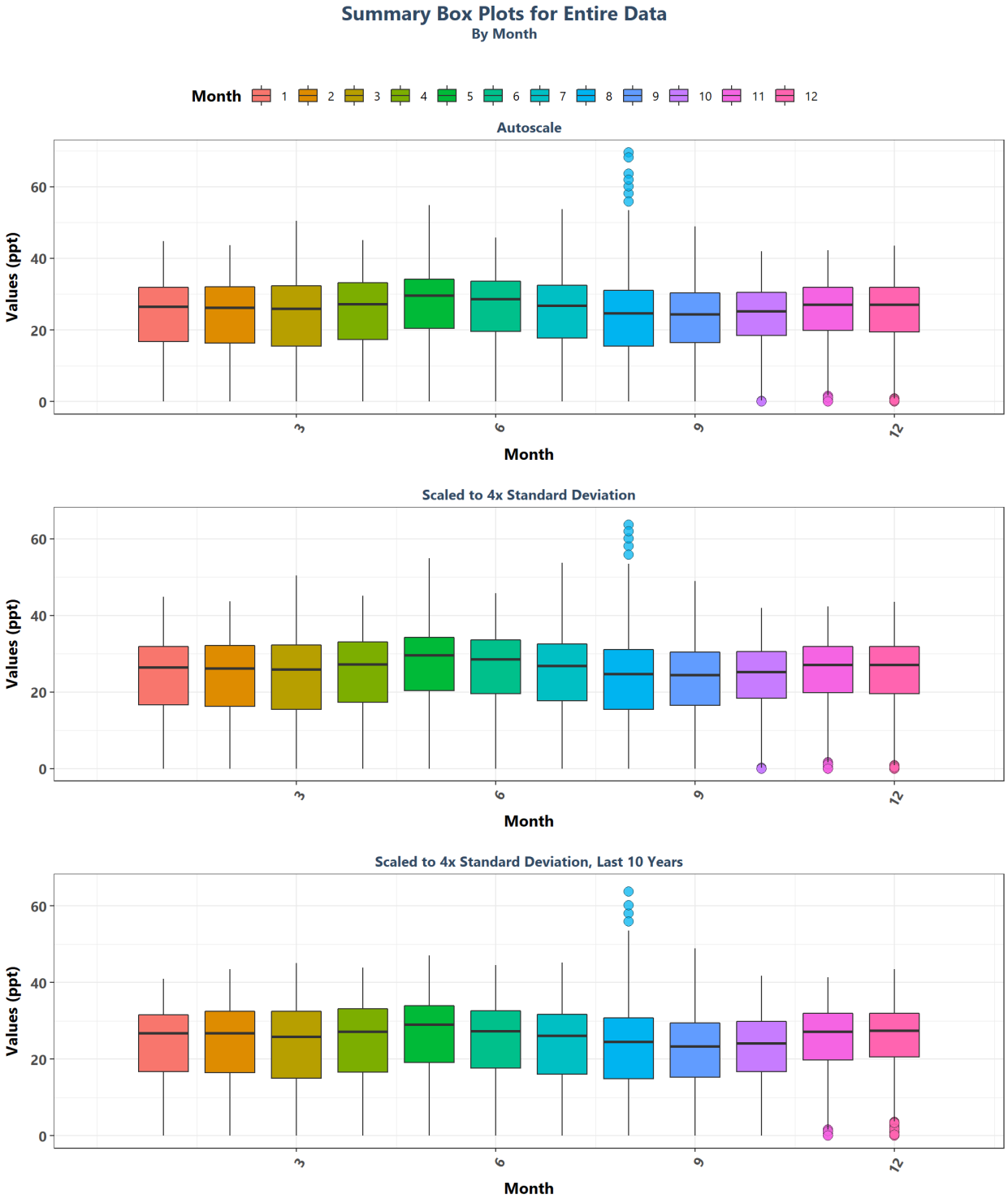
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit,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

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

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

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

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

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

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

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

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

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

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


Summary Box Plots for Entire Data
By Year
Autoscale
Values (ppt)
0
20
40
60
1970
1990
2010
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
1970
1990
2010
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
2012
2014
2016
2018
2020
2022
Year

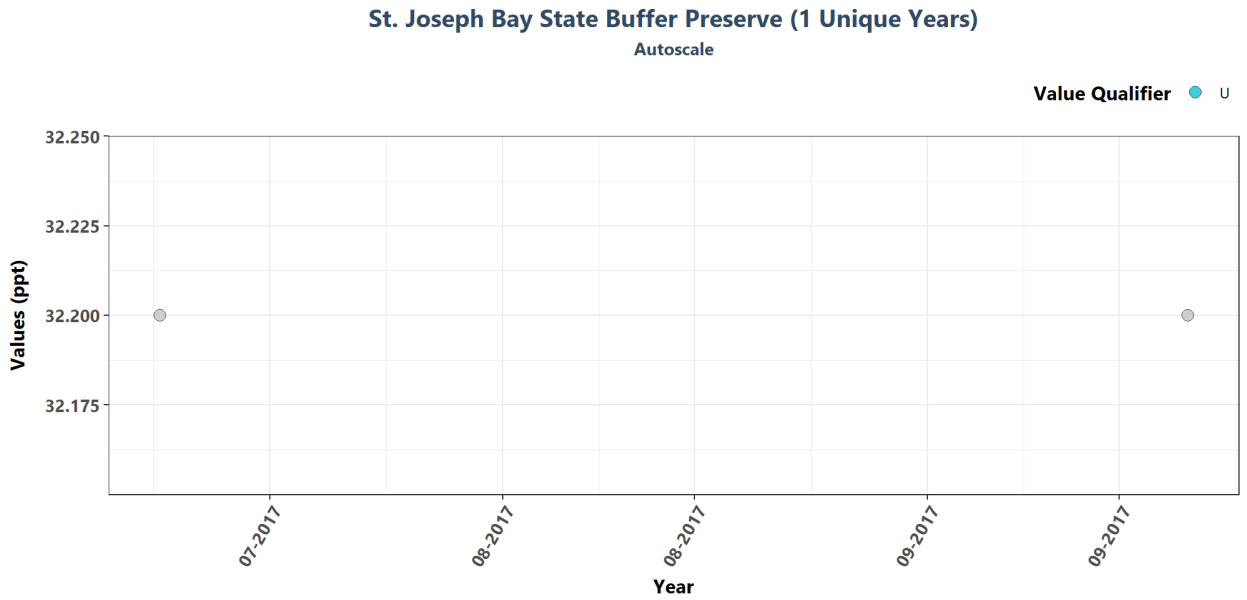

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
20
40
60
1970
1990
2010
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
1970
1990
2010
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
2012
2014
2016
2018
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)
   }
}
```



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

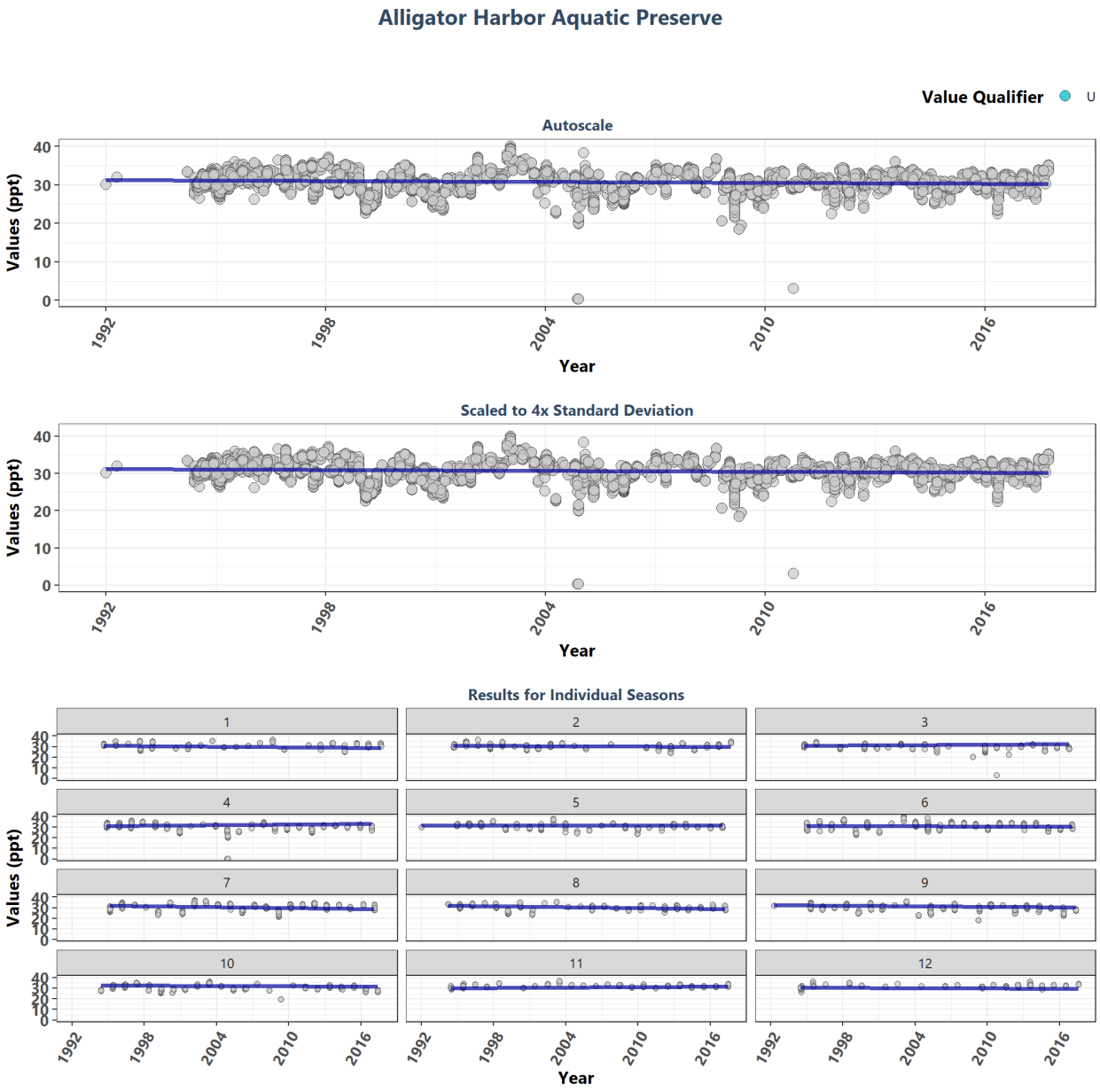
1. Use the data set that only has `SufficientData` of `TRUE` for the desired managed area
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Determine the minimum, mean, and standard deviation for the data to be used for y-axis scales
    - Excludes the top 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 = rely
                                                                                         y = relyear_
                                                                                         xend = relye
                                                                                         yend = relye
                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 = rely
                                                                                         y = relyear_
                                                                                         xend = relye
                                                                                         yend = relye
                color="#000099", size=1.2, alpha=0.7, inherit.aes = FALSE) +
   ylim(min_RV, y_scale) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
```

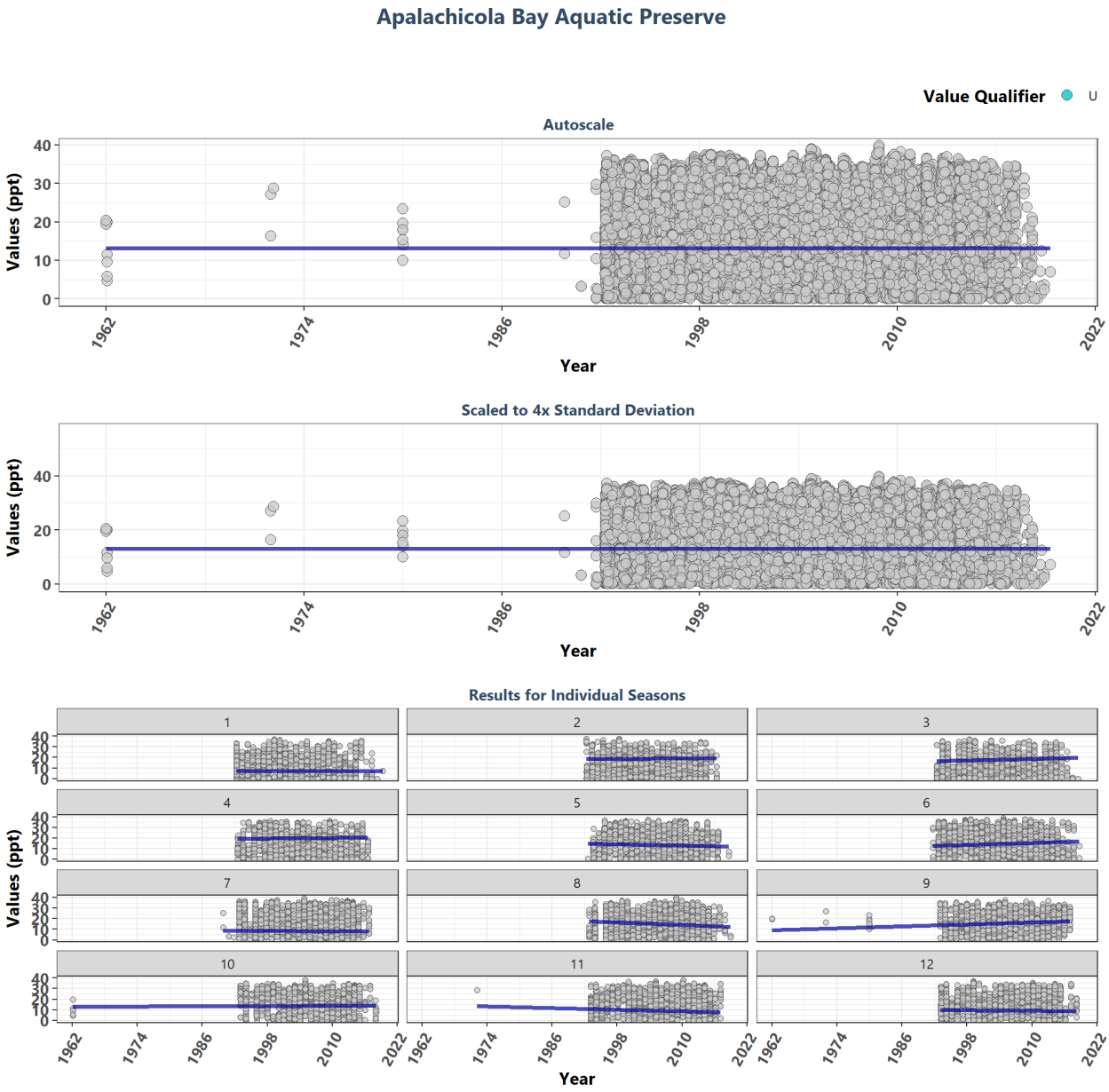
```
      plot_theme +
      theme(legend.position="none") +
      {if(inc_H==TRUE){
         scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                    "HU"="#7CAE00"), na.value="#cccccc")
      } else if(param_name=="Secchi_Depth"){
         scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                    "SU"="#7CAE00"), na.value="#cccccc")
      } else {
         scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
      }} +
      scale_x_continuous(breaks = xbrks,
                         labels = xlabs)

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

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

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

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

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

```
            add_footnote(
            "p < 0.00005 appear as 0 due to rounding"))
      cat('\n')
      rm(plot_data)
      rm(KTset, leg)
   }
}
```



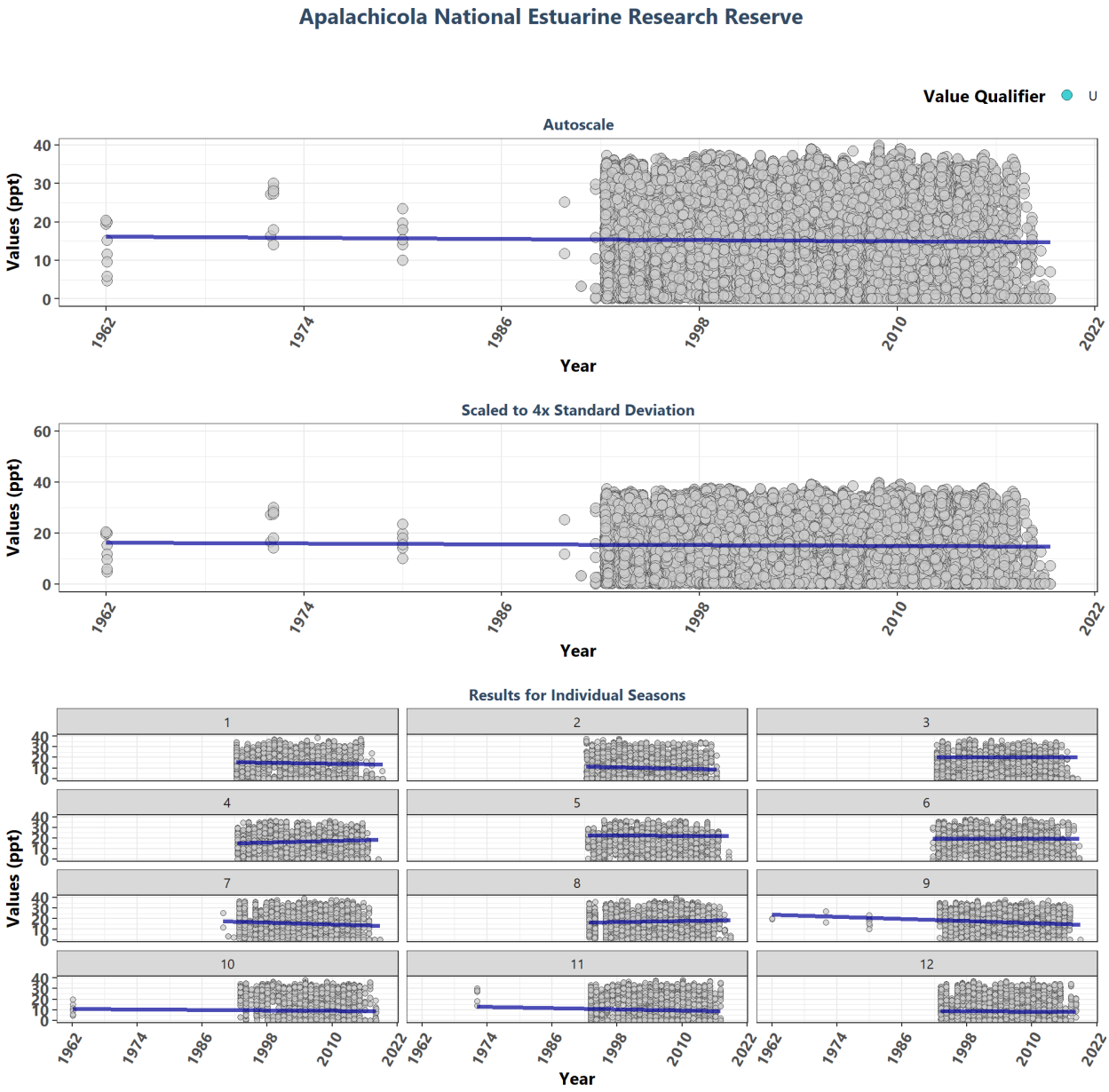
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5978	31.00	-0.0586	-0.0400	31.2588	-7.6	0.0000	411.2	0	-1
1	254	31.83	-0.1439	-0.1000	31.3000	0.0	0.9804	NA	NA	-1
2	507	30.90	-0.0675	-0.0450	31.0600	-2.8	0.0052	NA	NA	-1
3	384	30.60	0.2344	0.0600	30.7550	-8.7	0.0000	NA	NA	1
4	620	31.20	0.3985	0.0941	30.6353	-10.6	0.0000	NA	NA	1
5	508	31.70	0.0011	0.0000	31.8250	-4.8	0.0000	NA	NA	-1
6	726	32.00	-0.0826	-0.0353	31.2176	-1.7	0.0887	NA	NA	-1
7	762	30.80	-0.2956	-0.1236	31.9594	3.7	0.0002	NA	NA	-1
8	481	30.30	-0.2831	-0.1500	32.4000	-3.6	0.0003	NA	NA	-1
9	658	30.20	-0.1433	-0.0889	32.5000	-5.5	0.0000	NA	NA	-1
10	471	30.70	-0.0421	-0.0360	32.4680	-2.2	0.0281	NA	NA	-1
11	290	31.33	0.0892	0.0800	29.6800	6.0	0.0000	NA	NA	1
12	317	31.20	-0.1089	-0.0438	30.6938	11.0	0.0000	NA	NA	-1

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



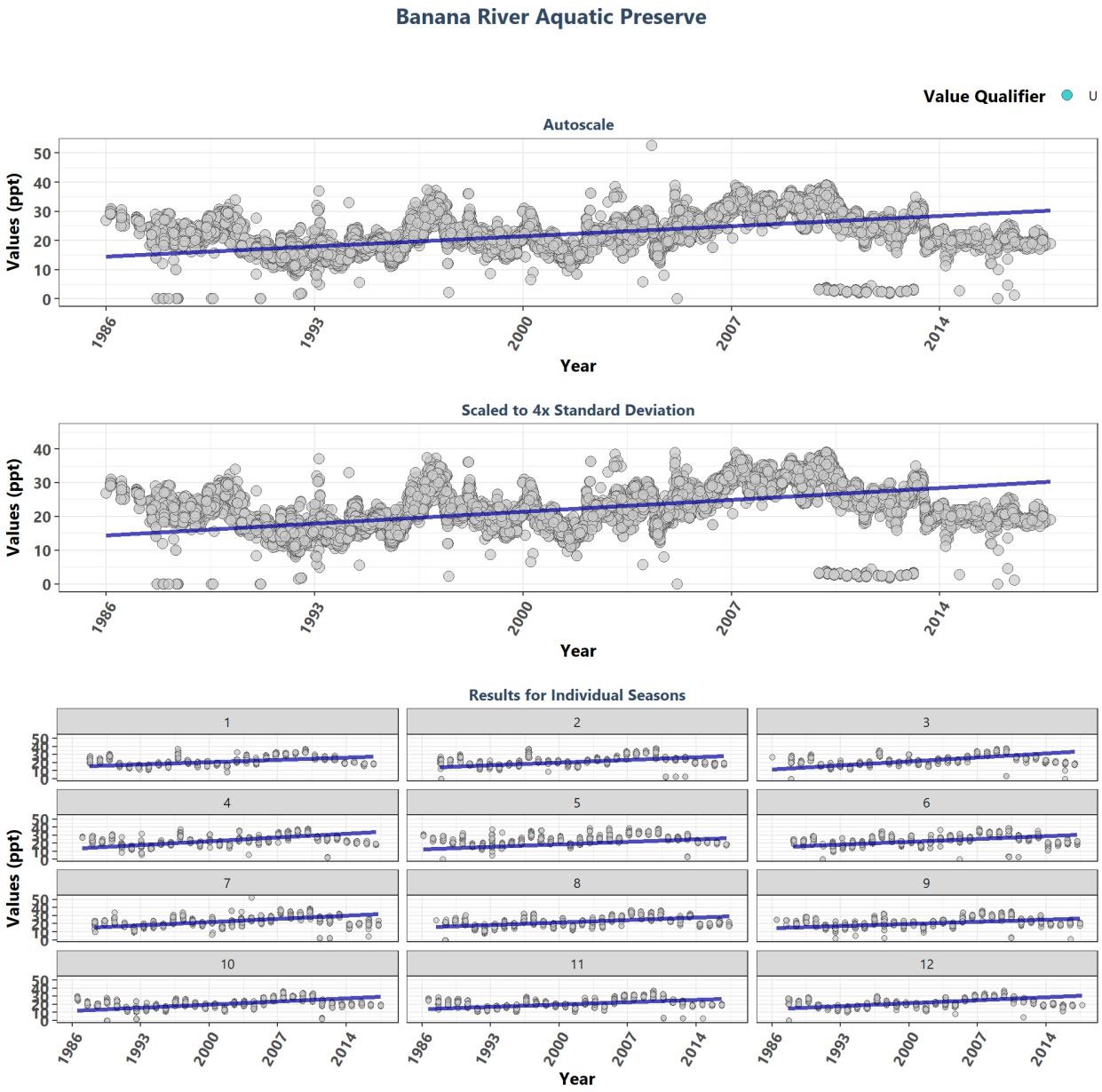
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	35550	13.4	-0.0029	0.0000	13.1125	-0.2	0.8525	207.1	0	0
1	2821	9.4	-0.0087	0.0000	7.0000	-2.9	0.0041	NA	NA	0
2	2812	9.2	0.0164	0.0333	17.5333	-7.2	0.0000	NA	NA	0
3	3314	7.0	0.0485	0.1125	13.2500	-0.8	0.4524	NA	NA	0
4	2723	8.5	0.0272	0.0538	17.7308	-1.8	0.0708	NA	NA	0
5	2723	13.1	-0.0454	-0.1000	17.6000	-3.6	0.0004	NA	NA	0
6	3080	14.9	0.0638	0.1524	8.2476	-7.0	0.0000	NA	NA	0
7	3510	15.7	-0.0230	-0.0125	9.0375	5.5	0.0000	NA	NA	0
8	3458	14.8	-0.0834	-0.2111	24.1889	5.6	0.0000	NA	NA	0
9	3139	19.0	0.0617	0.1500	9.1000	1.4	0.1685	NA	NA	0
10	2810	20.1	0.0102	0.0188	12.9750	2.2	0.0304	NA	NA	0
11	2748	18.2	-0.0903	-0.1267	14.6467	3.8	0.0001	NA	NA	0
12	2412	13.8	-0.0360	-0.0444	11.2667	0.8	0.4524	NA	NA	0

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



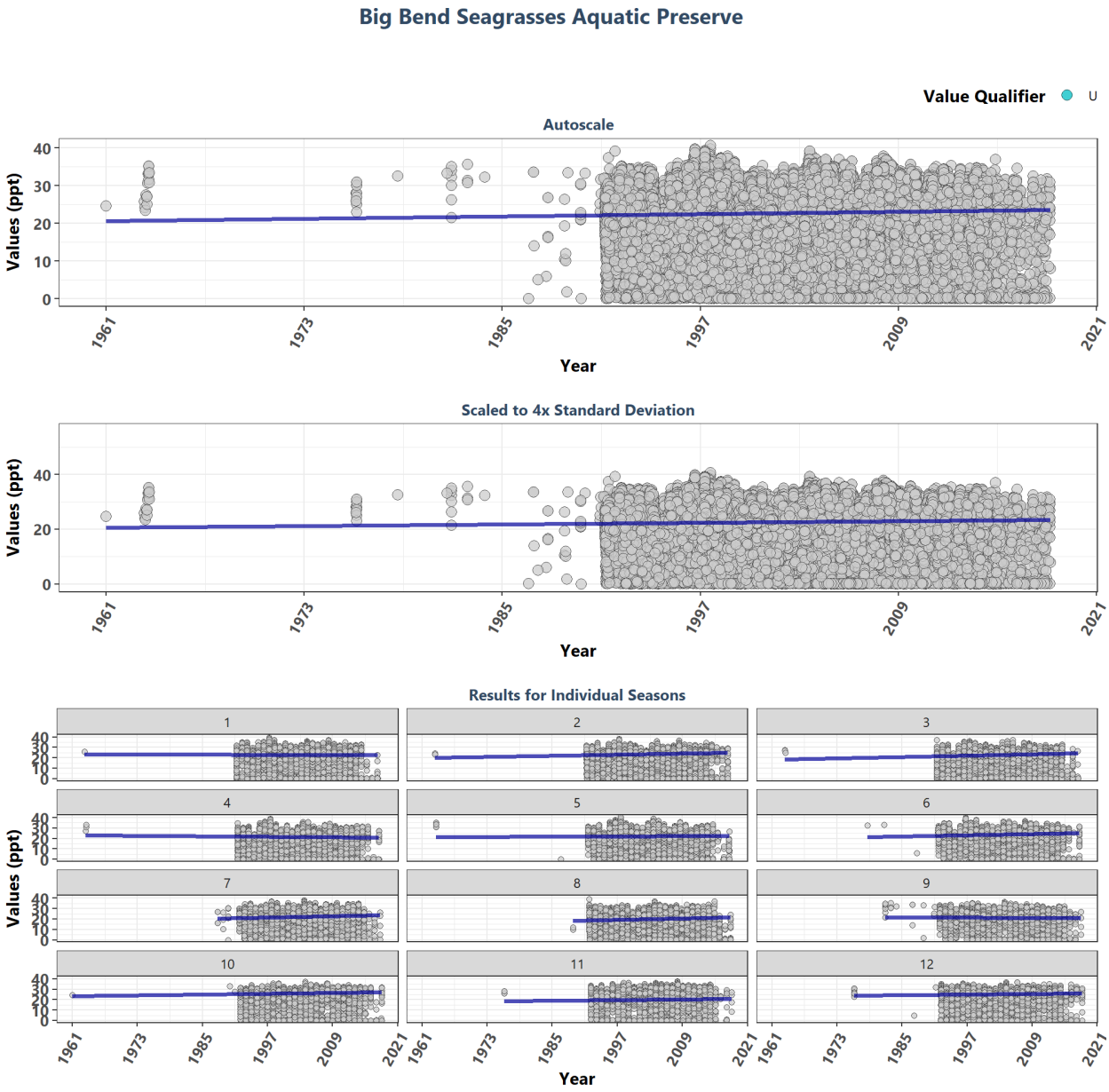
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	53373	14.7	-0.0275	-0.0250	16.2123	-9.0	0.0000	212.9	0	-1
1	4381	9.9	-0.0416	-0.0769	17.7846	-6.6	0.0000	NA	NA	-1
2	4458	9.8	-0.0862	-0.1100	14.5300	-8.7	0.0000	NA	NA	-1
3	5253	8.5	0.0062	0.0000	20.4000	-3.7	0.0002	NA	NA	-1
4	4382	9.3	0.0510	0.1143	11.6857	-4.6	0.0000	NA	NA	1
5	4162	15.3	-0.0163	-0.0222	23.2778	-6.7	0.0000	NA	NA	-1
6	4304	16.5	-0.0065	0.0000	19.6000	-7.1	0.0000	NA	NA	-1
7	4850	17.5	-0.0692	-0.1364	21.3000	3.7	0.0002	NA	NA	-1
8	4751	16.6	0.0351	0.0650	14.6400	5.3	0.0000	NA	NA	1
9	4544	20.4	-0.0724	-0.1667	23.8333	0.6	0.5313	NA	NA	-1
10	4484	22.3	-0.0465	-0.0368	10.8842	-1.6	0.1023	NA	NA	-1
11	4107	19.6	-0.0665	-0.1000	14.1000	-0.6	0.5324	NA	NA	-1
12	3697	14.4	-0.0337	-0.0200	9.3400	-3.8	0.0001	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	23220	19.90	0.3486	0.5000	14.4489	79.0	0	188.3	0	1
1	1527	18.80	0.3188	0.4059	14.6294	24.5	0	NA	NA	1
2	2036	17.60	0.3382	0.4667	13.7250	21.6	0	NA	NA	1
3	1764	19.33	0.4585	0.7200	11.5600	21.3	0	NA	NA	1
4	2176	21.00	0.4291	0.6667	13.0933	21.0	0	NA	NA	1
5	2390	21.20	0.3153	0.4444	12.7111	26.1	0	NA	NA	1
6	2000	21.00	0.3050	0.4895	15.1263	20.5	0	NA	NA	1
7	2013	21.76	0.3553	0.5588	14.4941	29.0	0	NA	NA	1
8	2128	20.20	0.2991	0.4332	15.8021	31.8	0	NA	NA	1
9	1716	18.60	0.3170	0.3778	14.4444	19.8	0	NA	NA	1
10	2317	19.50	0.4176	0.5500	12.2000	23.1	0	NA	NA	1
11	1826	19.40	0.2736	0.4083	14.5000	17.6	0	NA	NA	1
12	1327	20.10	0.3663	0.5133	14.4533	20.1	0	NA	NA	1

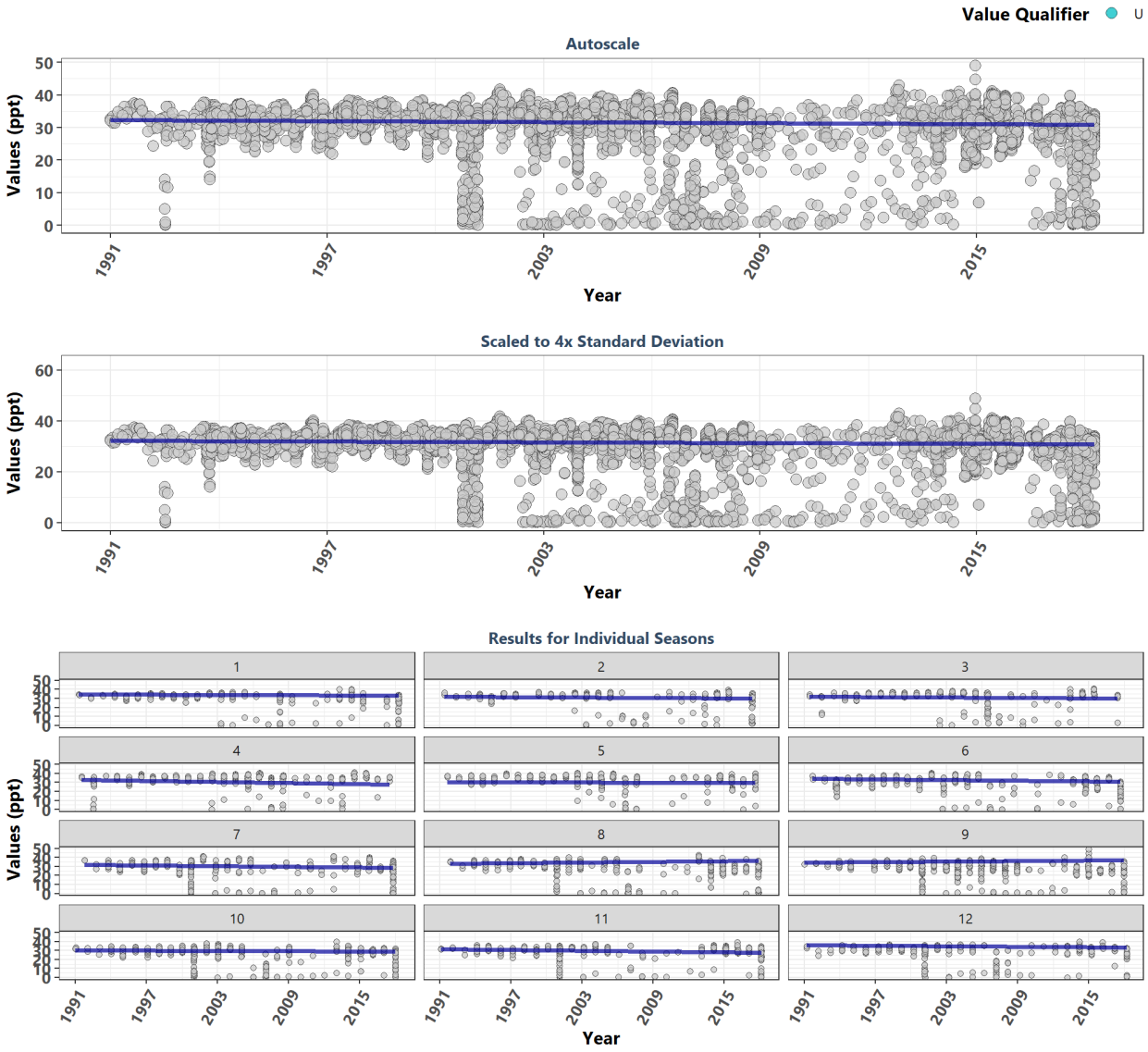
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	95882	22.6	0.0322	0.0500	20.6539	14.7	0.0000	218.7	0	1
1	6844	23.7	-0.0053	-0.0059	23.0471	7.1	0.0000	NA	NA	-1
2	7046	22.8	0.0569	0.0889	19.9667	-0.7	0.5048	NA	NA	1
3	8674	21.3	0.0627	0.1067	18.4133	-1.4	0.1552	NA	NA	1
4	8530	20.0	-0.0268	-0.0375	22.9125	8.7	0.0000	NA	NA	-1
5	8167	22.1	0.0104	0.0176	21.3412	1.4	0.1596	NA	NA	1
6	7723	23.0	0.0659	0.0875	19.9375	8.3	0.0000	NA	NA	1
7	8107	22.1	0.0599	0.1077	17.4692	8.1	0.0000	NA	NA	1
8	8900	20.1	0.0628	0.1154	15.1538	3.8	0.0002	NA	NA	1
9	8668	21.3	-0.0102	-0.0143	21.9143	-3.7	0.0002	NA	NA	-1
10	8585	23.7	0.0534	0.0667	23.2333	9.2	0.0000	NA	NA	1
11	7431	26.1	0.0266	0.0462	18.1154	6.9	0.0000	NA	NA	1
12	7207	25.0	0.0371	0.0500	22.9500	4.7	0.0000	NA	NA	1

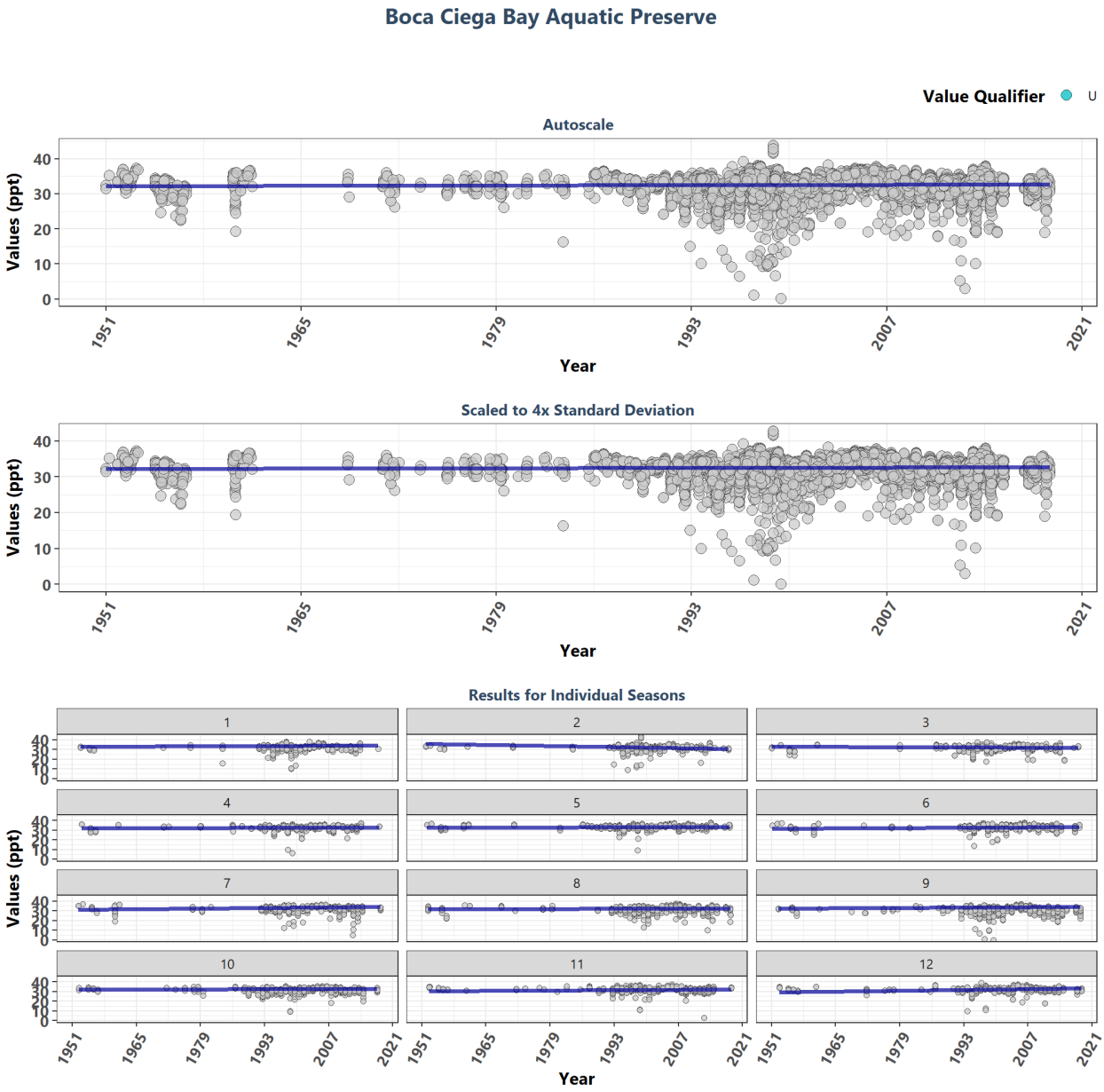
[a] $p < 0.00005$ appear as 0 due to rounding

Biscayne Bay Aquatic Preserve



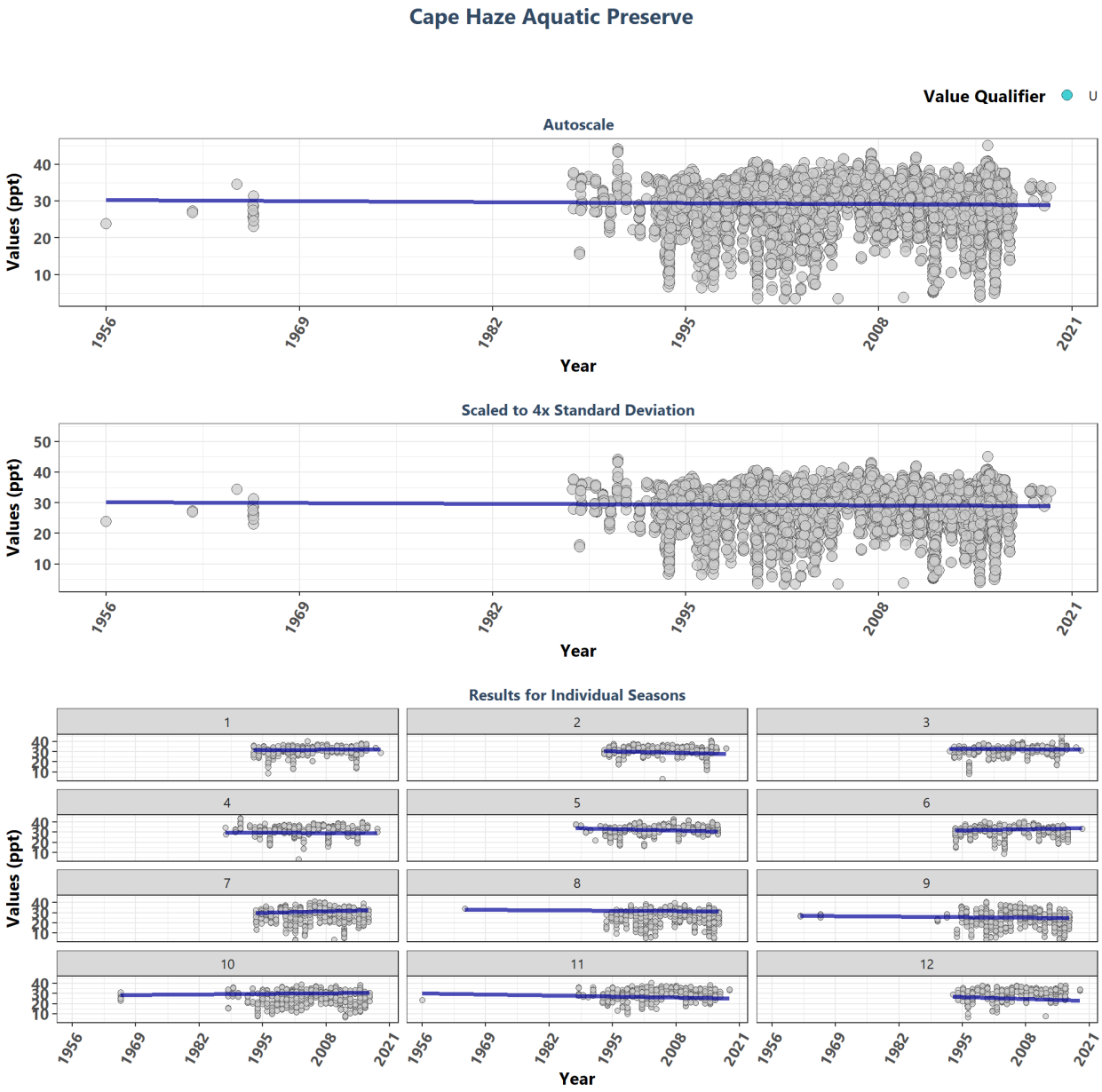
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	6270	31.50	-0.0670	-0.0545	32.3432	-6.5	0.0000	173.5	0	-1
1	378	31.76	-0.0628	-0.0467	34.5533	-6.7	0.0000	NA	NA	-1
2	329	33.48	-0.1179	-0.0850	31.9450	-1.7	0.0860	NA	NA	-1
3	323	34.70	-0.1194	-0.0880	32.0080	5.1	0.0000	NA	NA	-1
4	832	35.70	-0.2067	-0.1944	32.9756	5.4	0.0000	NA	NA	-1
5	545	34.50	-0.0276	-0.0343	30.3143	-3.7	0.0002	NA	NA	-1
6	420	29.35	-0.2304	-0.1400	34.4250	-4.2	0.0000	NA	NA	-1
7	459	29.80	-0.1037	-0.1333	31.8900	-0.9	0.3731	NA	NA	-1
8	493	29.67	0.1888	0.1348	32.6784	-6.9	0.0000	NA	NA	1
9	1008	29.89	0.1237	0.1014	34.1786	-4.9	0.0000	NA	NA	1
10	566	29.14	-0.0595	-0.0481	29.8850	-2.1	0.0334	NA	NA	-1
11	461	30.50	-0.1359	-0.1580	31.5620	-3.8	0.0001	NA	NA	-1
12	456	30.60	-0.1055	-0.0887	35.7413	-3.8	0.0001	NA	NA	-1

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



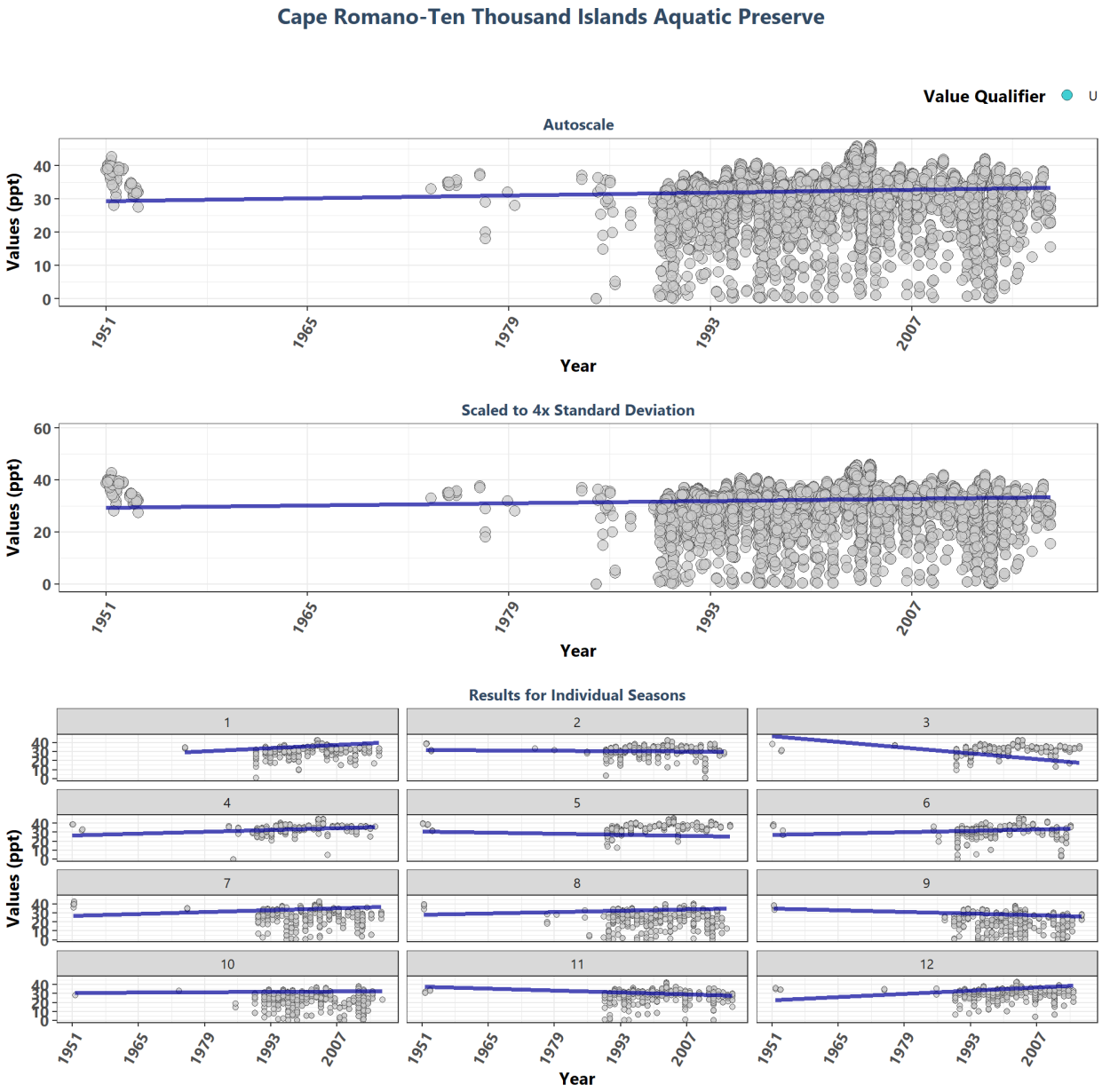
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	16366	32.70	0.0271	0.0060	32.2129	3.4	0.0006	280.4	0	1
1	1223	32.20	0.0408	0.0130	33.1940	7.4	0.0000	NA	NA	1
2	1126	32.70	-0.1782	-0.0820	36.0460	0.9	0.3483	NA	NA	-1
3	1128	33.04	-0.0565	-0.0167	33.2400	4.5	0.0000	NA	NA	-1
4	1081	33.30	0.0264	0.0071	32.0759	5.1	0.0000	NA	NA	1
5	1595	33.90	0.0294	0.0118	32.5165	5.3	0.0000	NA	NA	1
6	1178	33.87	0.0895	0.0330	31.3570	2.1	0.0357	NA	NA	1
7	1215	33.14	0.1025	0.0364	31.4447	1.5	0.1249	NA	NA	1
8	1495	32.40	0.0305	0.0125	31.7250	1.8	0.0768	NA	NA	1
9	1898	31.70	0.0892	0.0250	32.6000	-11.7	0.0000	NA	NA	1
10	1758	31.70	0.0186	0.0069	32.3500	5.3	0.0000	NA	NA	1
11	1368	32.39	0.0842	0.0333	30.0717	-3.1	0.0017	NA	NA	1
12	1301	32.45	0.1412	0.0577	29.2577	1.4	0.1532	NA	NA	1

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



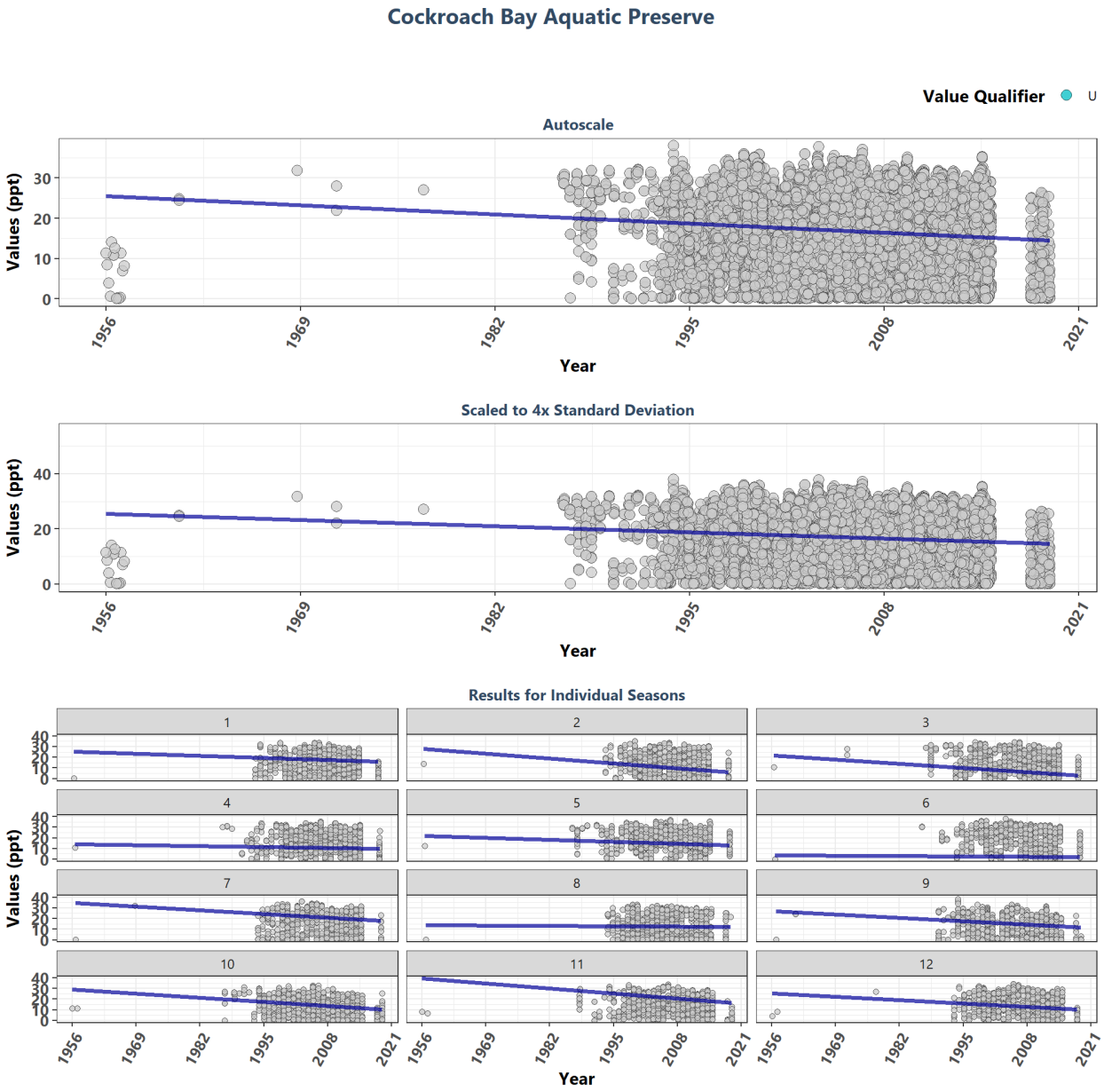
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	7419	30.00	-0.0137	-0.0200	30.2450	-2.3	0.0231	68.3	0	-1
1	481	31.20	0.0282	0.0300	30.3400	3.9	0.0001	NA	NA	1
2	532	32.00	-0.0688	-0.1091	34.8118	-1.1	0.2701	NA	NA	-1
3	529	31.90	-0.0092	-0.0077	32.8846	1.0	0.3314	NA	NA	-1
4	611	32.50	-0.0183	-0.0167	30.1500	-0.3	0.7344	NA	NA	-1
5	666	33.00	-0.0983	-0.1091	37.3636	3.5	0.0004	NA	NA	-1
6	570	31.80	0.0913	0.0800	28.9200	-3.5	0.0004	NA	NA	1
7	804	29.03	0.1194	0.1083	25.7833	-2.9	0.0035	NA	NA	1
8	648	25.88	-0.0319	-0.0250	33.2750	-1.7	0.0824	NA	NA	-1
9	666	24.91	-0.0242	-0.0333	27.1367	-3.6	0.0003	NA	NA	-1
10	723	25.47	0.0389	0.0429	27.9714	-1.0	0.3299	NA	NA	1
11	656	29.30	-0.0455	-0.0779	30.0064	-0.7	0.4820	NA	NA	-1
12	533	30.20	-0.0943	-0.1471	32.5571	1.3	0.1789	NA	NA	-1

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



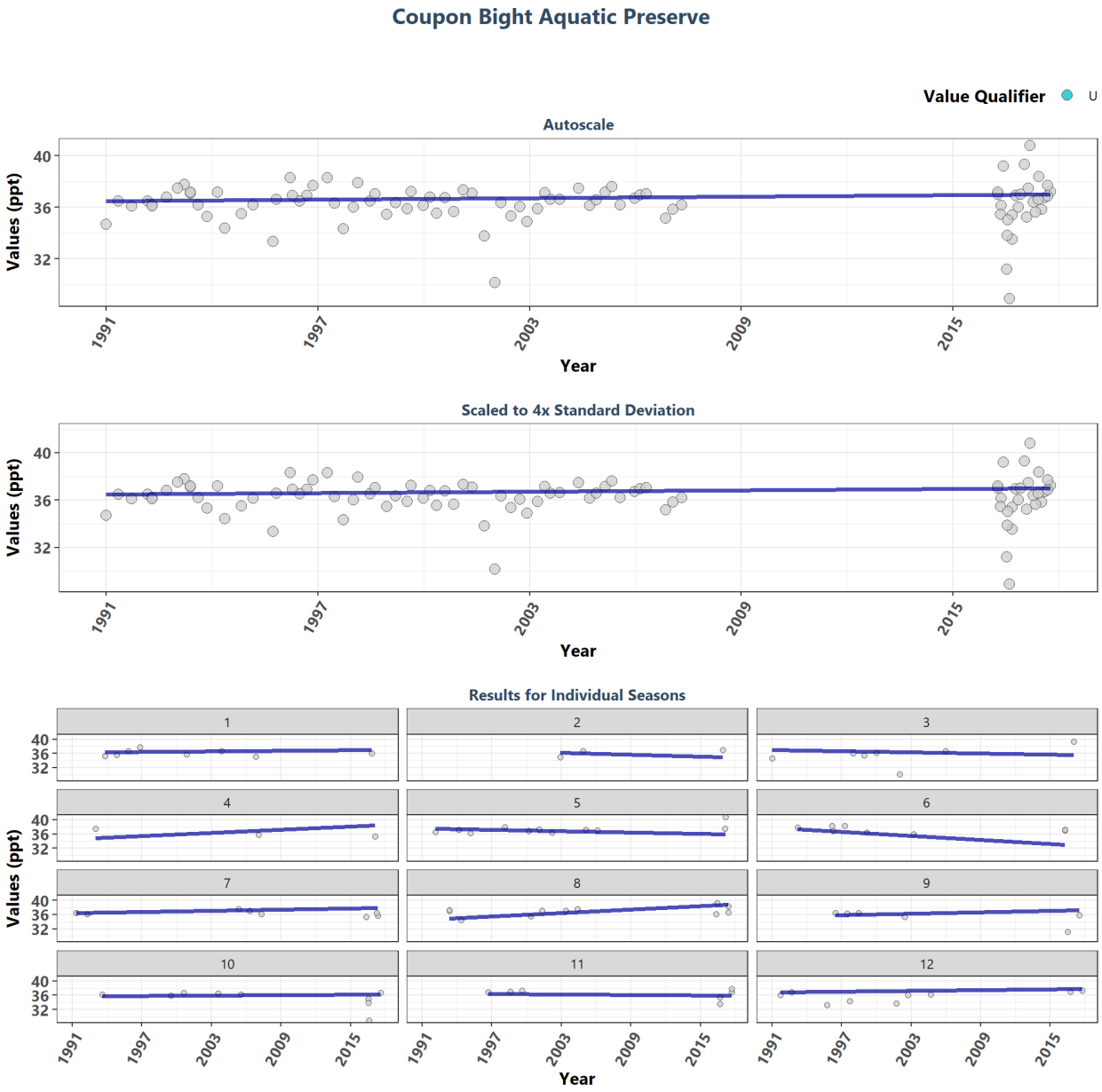
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	6731	32.00	0.0604	0.0611	29.4000	7.4	0.0000	467.3	0	1
1	556	32.40	0.3220	0.2600	23.1000	4.4	0.0000	NA	NA	1
2	626	33.00	-0.0234	-0.0250	31.7000	5.2	0.0000	NA	NA	-1
3	476	33.89	-0.2744	-0.4556	47.0222	6.8	0.0000	NA	NA	-1
4	565	35.00	0.1507	0.1400	26.6800	11.7	0.0000	NA	NA	1
5	559	36.10	-0.0619	-0.0857	31.0657	11.4	0.0000	NA	NA	-1
6	570	33.40	0.1242	0.1036	27.4255	5.4	0.0000	NA	NA	1
7	514	30.10	0.2094	0.1455	26.9132	-4.5	0.0000	NA	NA	1
8	592	28.30	0.1397	0.1000	28.1500	-3.5	0.0005	NA	NA	1
9	547	24.70	-0.0957	-0.1333	34.8333	-9.6	0.0000	NA	NA	-1
10	634	26.78	0.0390	0.0312	30.6500	-2.3	0.0195	NA	NA	1
11	490	30.50	-0.1313	-0.1462	37.5538	-0.8	0.4377	NA	NA	-1
12	602	32.15	0.3291	0.2600	22.5200	1.4	0.1512	NA	NA	1

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



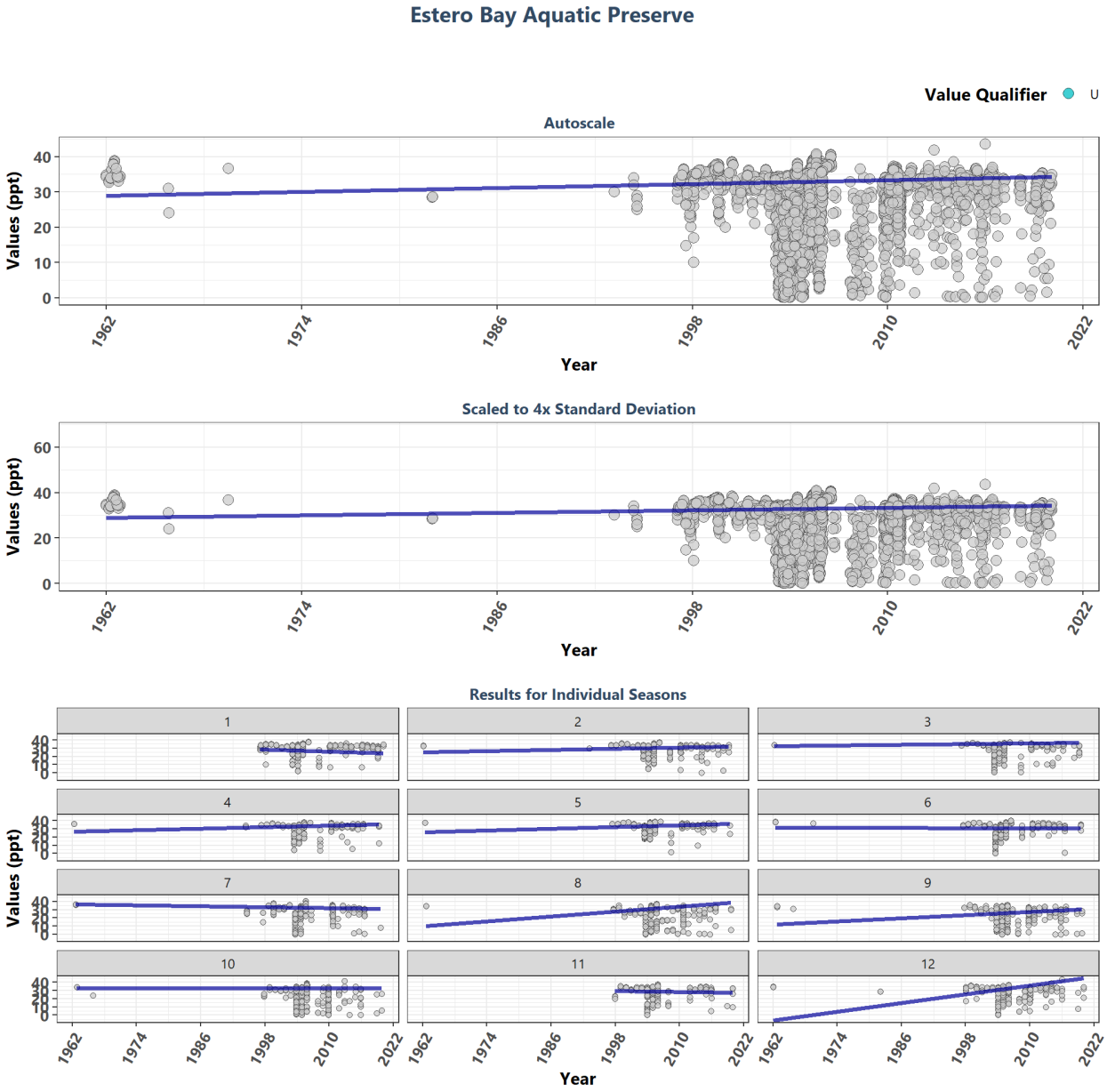
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	14670	13.55	-0.1057	-0.1737	25.4931	-18.8	0.0000	119.8	0	-1
1	978	12.60	-0.0624	-0.1538	25.3462	-0.7	0.4812	NA	NA	-1
2	981	12.89	-0.2021	-0.3500	27.8050	-5.1	0.0000	NA	NA	-1
3	1057	14.94	-0.1754	-0.3000	21.6800	-5.4	0.0000	NA	NA	-1
4	1023	17.50	-0.0579	-0.0714	14.2714	-3.0	0.0028	NA	NA	-1
5	1193	21.52	-0.0596	-0.1400	21.9600	-6.7	0.0000	NA	NA	-1
6	1016	21.20	-0.0382	-0.0200	3.6200	-7.3	0.0000	NA	NA	-1
7	1129	10.70	-0.1288	-0.2625	34.6450	-2.9	0.0035	NA	NA	-1
8	1936	2.70	-0.0150	-0.0250	13.8750	-2.6	0.0106	NA	NA	-1
9	1867	8.18	-0.1114	-0.2412	26.9988	-11.5	0.0000	NA	NA	-1
10	1228	11.00	-0.1334	-0.2933	28.8733	-10.6	0.0000	NA	NA	-1
11	1202	14.50	-0.1528	-0.3615	39.6385	-6.9	0.0000	NA	NA	-1
12	1060	15.10	-0.1096	-0.2500	25.6400	-2.9	0.0036	NA	NA	-1

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



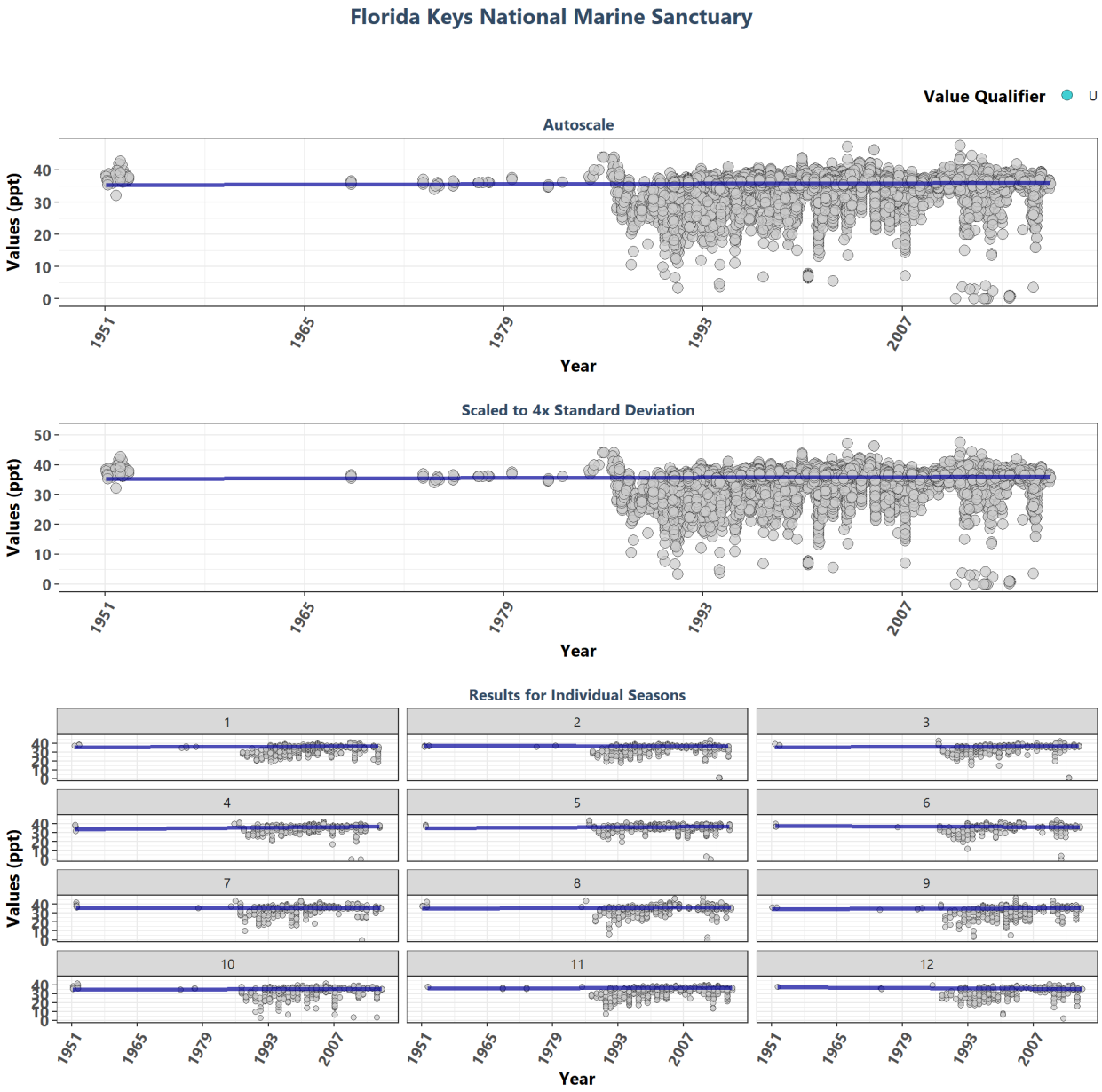
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	91	36.50	0.0615	0.0190	36.4868	1.0	0.3186	18.7	0.0671	0
1	8	35.84	0.0476	0.0267	36.2333	0.4	0.7105	NA	NA	0
2	3	36.59	-1.0000	-0.0942	37.3480	1.0	0.2963	NA	NA	0
3	7	36.03	-0.2500	-0.0511	36.9121	1.5	0.1331	NA	NA	0
4	3	35.84	1.0000	0.1511	34.4714	-1.0	0.2963	NA	NA	0
5	11	37.16	-0.2500	-0.0575	37.5023	1.3	0.1844	NA	NA	0
6	8	37.10	-0.5333	-0.1916	37.7894	-0.8	0.4475	NA	NA	0
7	9	36.20	0.2545	0.0581	36.4024	-0.9	0.3428	NA	NA	0
8	11	37.10	0.5238	0.1596	34.5939	1.0	0.3071	NA	NA	0
9	6	36.06	0.4444	0.0705	35.3948	-1.3	0.1806	NA	NA	0
10	9	36.20	0.1429	0.0238	35.6530	-0.9	0.3947	NA	NA	0
11	7	36.90	-0.2778	-0.0247	36.5711	0.0	1.0000	NA	NA	0
12	9	36.10	0.3273	0.0393	36.7639	1.6	0.1179	NA	NA	0

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



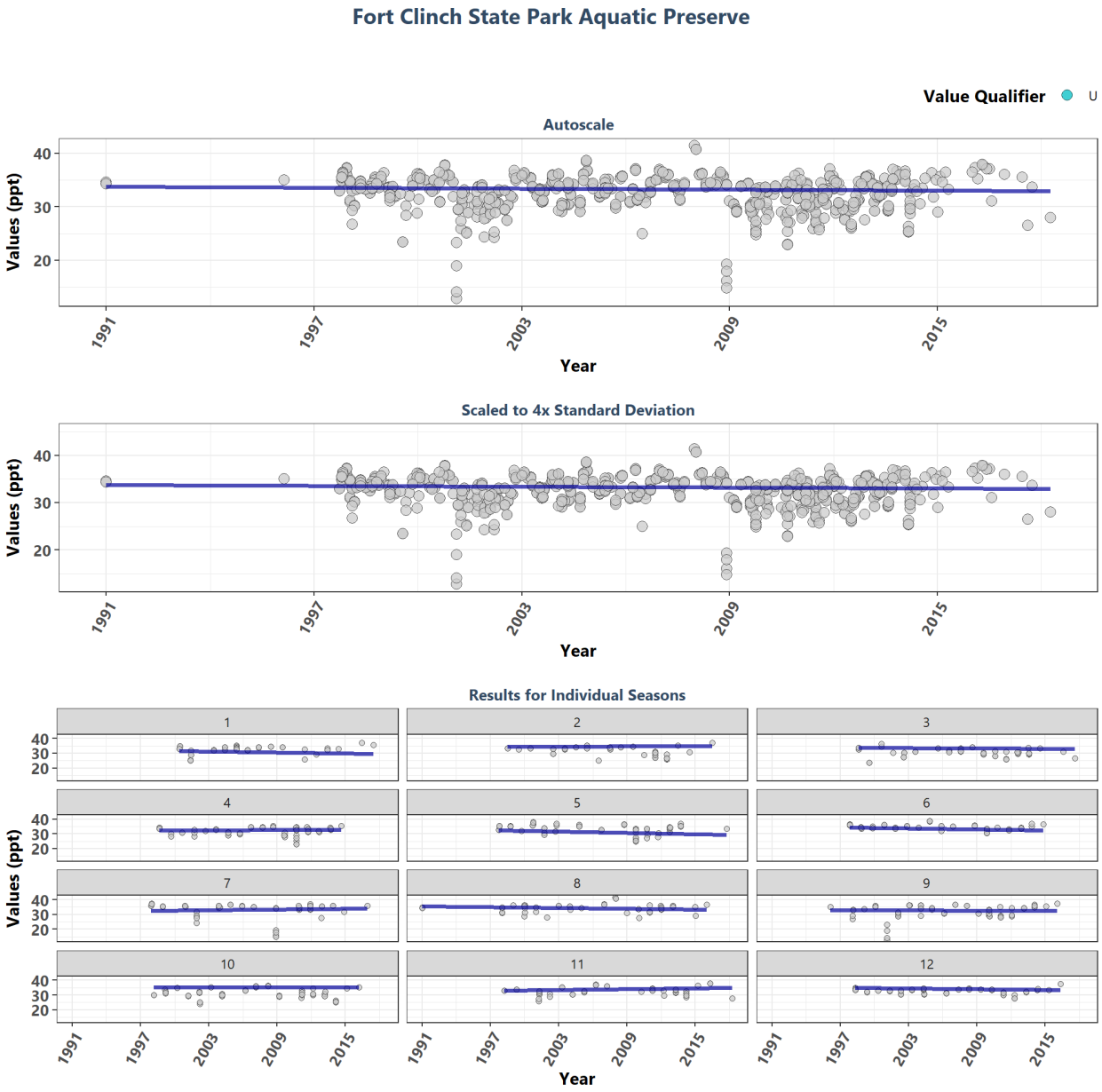
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	3538	31.80	0.0560	0.0925	28.9521	5.0	0.0000	85.4	0	1
1	271	32.50	-0.0806	-0.2000	35.3000	0.0	0.9935	NA	NA	-1
2	260	32.50	0.0383	0.1273	24.9273	-2.3	0.0238	NA	NA	1
3	297	33.20	0.0669	0.0707	32.6886	3.0	0.0031	NA	NA	1
4	280	34.10	0.1133	0.1545	26.4042	3.7	0.0002	NA	NA	1
5	341	35.80	0.1469	0.1783	26.2533	1.9	0.0599	NA	NA	1
6	300	32.65	-0.0094	-0.0125	31.5000	7.6	0.0000	NA	NA	-1
7	381	30.40	-0.0928	-0.0932	36.5993	1.1	0.2569	NA	NA	-1
8	279	26.10	0.1407	0.5000	10.0000	2.4	0.0166	NA	NA	1
9	288	26.50	0.0949	0.3286	11.6429	-2.1	0.0388	NA	NA	1
10	306	28.35	-0.0004	0.0000	32.5000	-0.7	0.4537	NA	NA	-1
11	223	31.50	-0.0282	-0.0833	32.0167	3.2	0.0013	NA	NA	-1
12	312	30.95	0.2896	0.8913	-6.5687	-0.2	0.8032	NA	NA	1

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



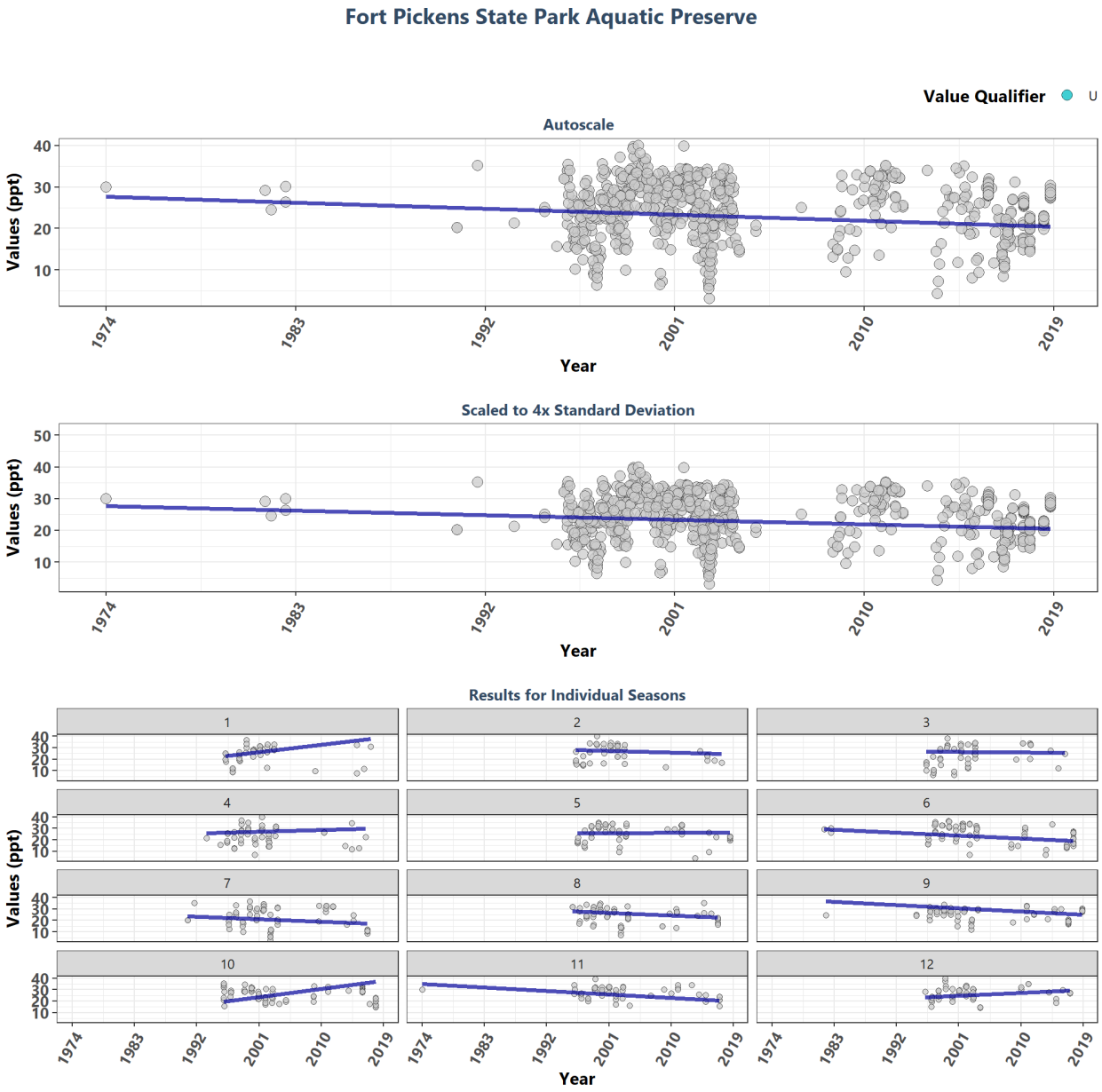
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	22740	36.16	0.0496	0.0099	35.3152	10.7	0.0000	663.8	0	1
1	1607	36.13	0.0938	0.0251	35.0939	6.5	0.0000	NA	NA	1
2	2051	36.10	-0.0729	-0.0122	37.1353	-7.1	0.0000	NA	NA	-1
3	1876	36.18	0.1005	0.0153	35.5365	18.2	0.0000	NA	NA	1
4	1774	36.40	0.2803	0.0443	33.8762	5.9	0.0000	NA	NA	1
5	2522	36.51	0.2001	0.0315	34.6960	3.0	0.0031	NA	NA	1
6	1466	36.50	-0.1047	-0.0222	37.3901	-4.2	0.0000	NA	NA	-1
7	2072	36.30	0.0040	0.0005	35.8727	6.9	0.0000	NA	NA	1
8	1544	36.30	0.1076	0.0231	34.9564	11.8	0.0000	NA	NA	1
9	2340	35.49	0.0721	0.0175	34.8346	3.2	0.0013	NA	NA	1
10	1819	35.76	0.0442	0.0161	34.6382	-9.5	0.0000	NA	NA	1
11	1938	35.90	0.0393	0.0072	36.1252	0.3	0.7905	NA	NA	1
12	1731	35.82	-0.1478	-0.0357	37.5798	4.5	0.0000	NA	NA	-1

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



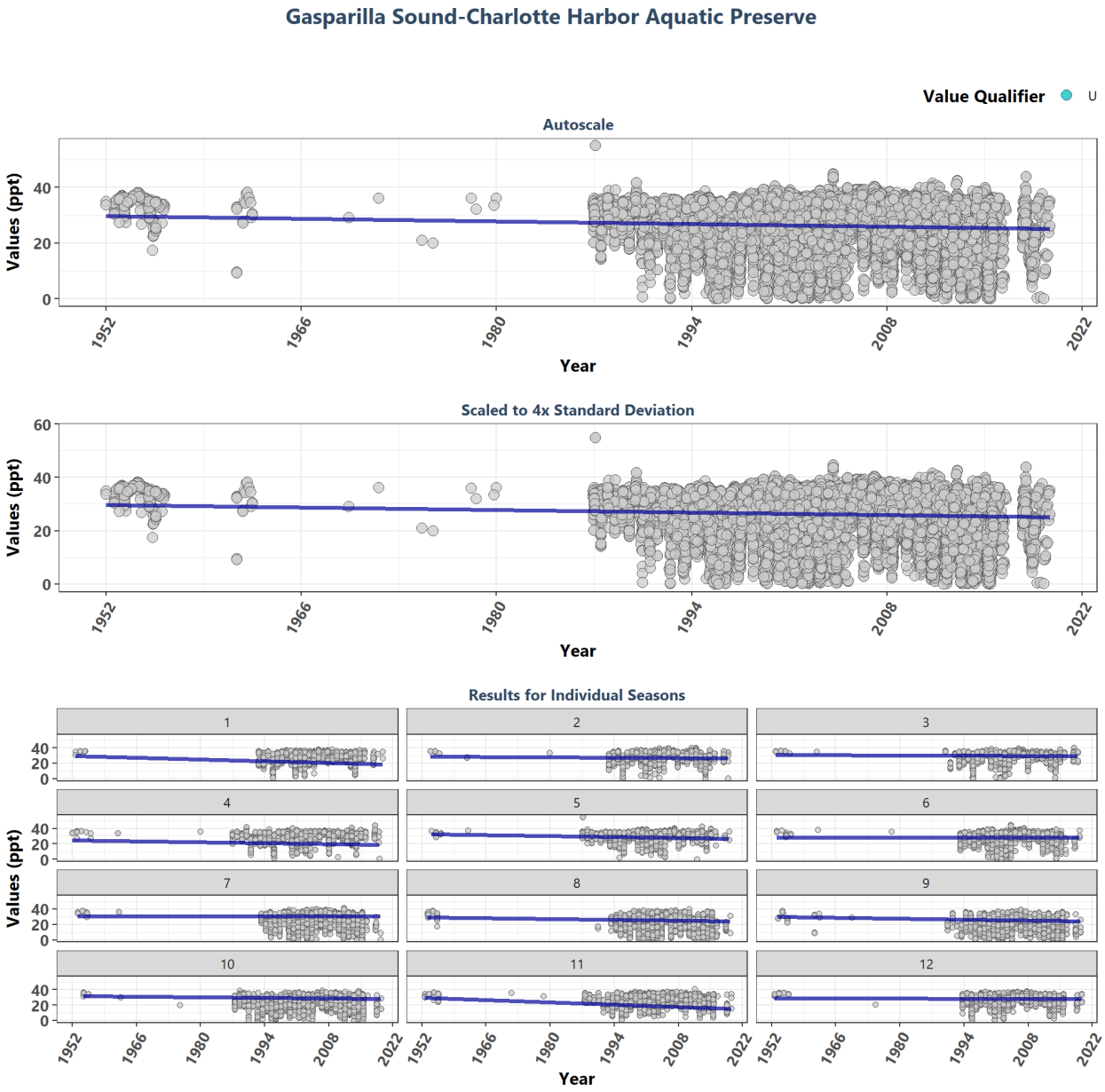
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	811	33.40	-0.0503	-0.0300	33.7714	-2.3	0.0217	23.6	0.0145	-1
1	50	33.08	-0.1135	-0.1095	32.6527	1.5	0.1216	NA	NA	-1
2	55	33.20	0.0742	0.0300	34.2200	-0.8	0.4515	NA	NA	1
3	68	30.80	-0.1103	-0.0429	34.0429	-2.9	0.0038	NA	NA	-1
4	75	32.50	0.0317	0.0158	32.2308	0.4	0.6893	NA	NA	1
5	79	34.25	-0.2392	-0.1500	33.5000	-1.4	0.1540	NA	NA	-1
6	74	34.20	-0.0700	-0.1000	34.9000	-2.6	0.0105	NA	NA	-1
7	77	35.33	0.1502	0.0800	31.8800	-0.6	0.5320	NA	NA	1
8	59	34.70	-0.2025	-0.0894	35.4075	0.8	0.4076	NA	NA	-1
9	72	33.75	-0.0193	-0.0111	32.9667	1.6	0.1162	NA	NA	-1
10	67	30.90	-0.0485	-0.0133	35.5167	-1.4	0.1741	NA	NA	-1
11	64	32.75	0.1264	0.0869	32.4469	-0.2	0.8243	NA	NA	1
12	71	33.40	-0.1091	-0.0714	35.3929	-1.4	0.1734	NA	NA	-1

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



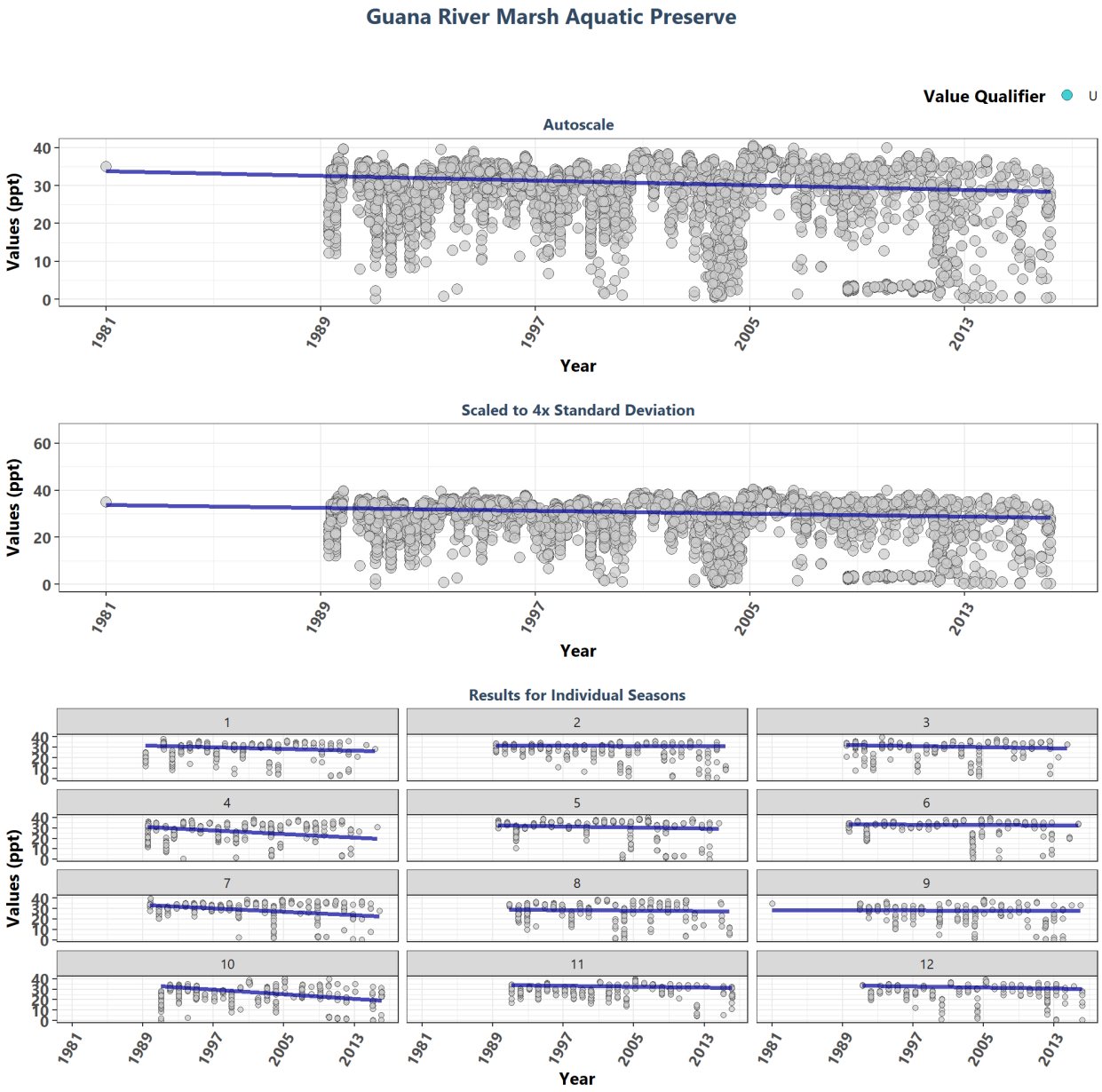
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	771	25.80	-0.0902	-0.1605	27.6375	-5.1	0.0000	58.1	0	-1
1	41	26.09	0.2671	0.7125	6.8534	2.5	0.0137	NA	NA	1
2	41	25.98	-0.1430	-0.1581	31.6678	0.0	0.9730	NA	NA	-1
3	48	24.25	-0.0732	-0.0535	27.6950	2.9	0.0042	NA	NA	-1
4	52	24.74	0.0718	0.1537	22.9968	0.8	0.4132	NA	NA	1
5	76	23.00	0.0049	0.0071	25.7803	-2.8	0.0059	NA	NA	1
6	84	26.02	-0.2119	-0.2771	31.3143	-2.7	0.0060	NA	NA	-1
7	63	21.00	-0.1111	-0.2350	27.5800	-1.3	0.1982	NA	NA	-1
8	60	25.46	-0.2017	-0.2573	33.7346	-2.5	0.0122	NA	NA	-1
9	99	25.50	-0.3179	-0.3045	38.8470	-1.1	0.2760	NA	NA	-1
10	106	27.88	0.2846	0.8000	1.8500	-4.9	0.0000	NA	NA	1
11	53	27.40	-0.2215	-0.3400	34.9850	-1.5	0.1310	NA	NA	-1
12	48	27.30	0.0784	0.2672	17.5215	0.7	0.4748	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	35125	26.40	-0.0456	-0.0667	29.6425	-13.5	0.0000	116.4	0	-1
1	2480	28.00	-0.1166	-0.1667	29.6000	0.0	0.9862	NA	NA	-1
2	2667	28.20	-0.0242	-0.0329	28.9214	-4.5	0.0000	NA	NA	-1
3	2559	28.10	-0.0257	-0.0256	30.7919	-0.6	0.5237	NA	NA	-1
4	3226	29.48	-0.0406	-0.0920	24.8680	-2.2	0.0284	NA	NA	-1
5	3130	30.40	-0.0583	-0.0875	32.9250	-0.3	0.7980	NA	NA	-1
6	2890	29.10	-0.0002	0.0000	28.0000	-2.9	0.0039	NA	NA	-1
7	2910	24.90	-0.0030	0.0000	30.4000	-2.8	0.0055	NA	NA	-1
8	2767	19.90	-0.0343	-0.0800	29.2200	-3.2	0.0014	NA	NA	-1
9	2946	17.77	-0.0708	-0.0800	29.7200	-8.3	0.0000	NA	NA	-1
10	3668	21.10	-0.0357	-0.0575	32.0900	-10.6	0.0000	NA	NA	-1
11	3167	25.56	-0.1016	-0.2206	29.6850	-6.0	0.0000	NA	NA	-1
12	2715	27.18	-0.0084	-0.0110	28.6830	-1.9	0.0585	NA	NA	-1

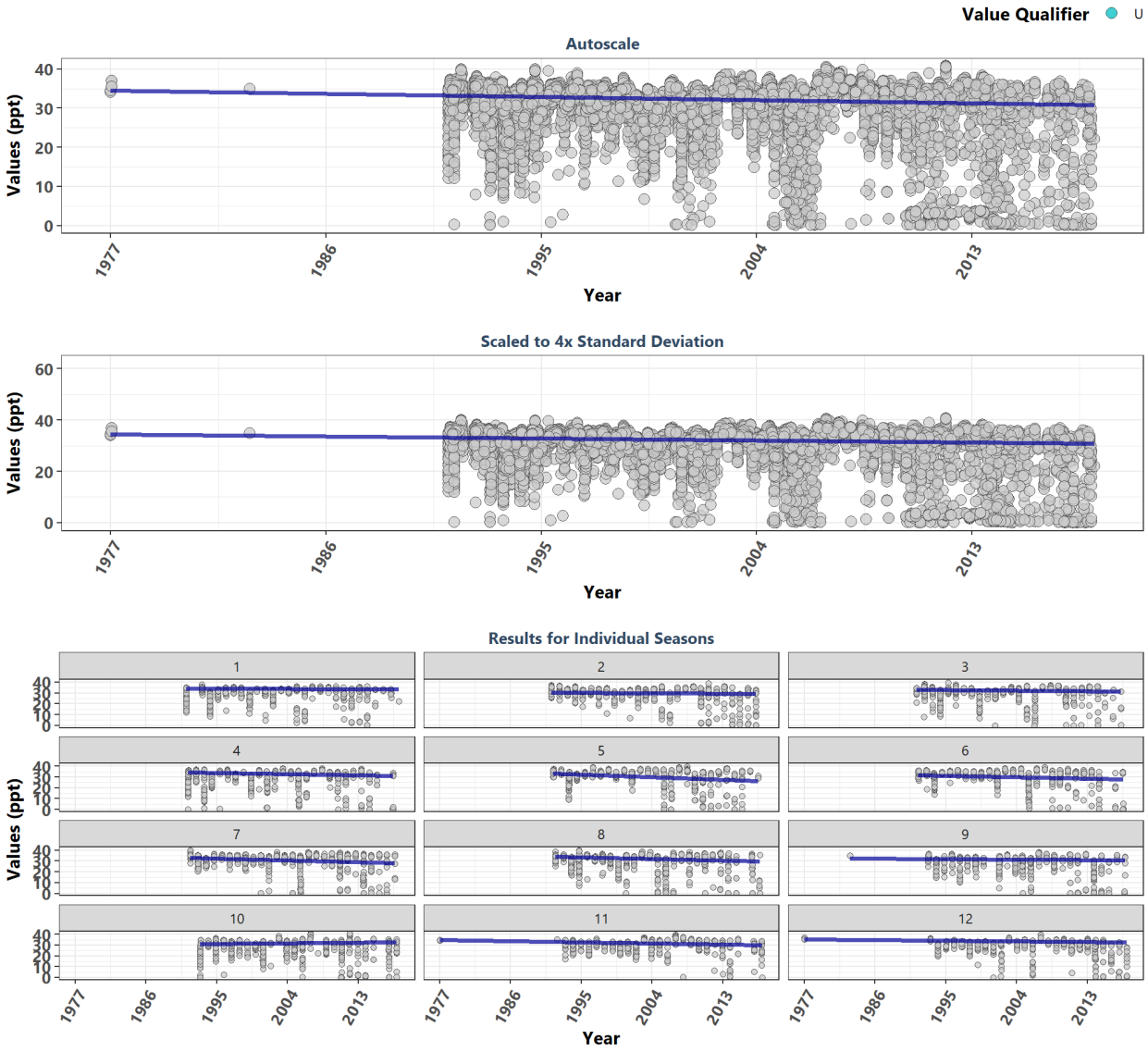
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	5321	30.70	-0.1154	-0.1500	33.7500	-11.6	0.0000	94.7	0	-1
1	514	30.80	-0.1015	-0.2000	33.5000	-3.3	0.0009	NA	NA	-1
2	425	30.30	-0.0195	-0.0200	31.7400	-9.9	0.0000	NA	NA	-1
3	354	31.72	-0.0980	-0.1273	33.2182	-3.1	0.0021	NA	NA	-1
4	570	31.40	-0.2038	-0.4143	34.2857	-0.7	0.4858	NA	NA	-1
5	600	33.30	-0.1092	-0.1142	33.3133	-2.8	0.0046	NA	NA	-1
6	397	32.90	-0.0469	-0.0500	34.0000	-3.1	0.0016	NA	NA	-1
7	564	33.10	-0.3209	-0.4000	36.7000	-1.7	0.0956	NA	NA	-1
8	457	30.30	-0.0637	-0.0750	29.9750	-3.3	0.0011	NA	NA	-1
9	290	29.73	-0.0102	-0.0109	28.3957	-7.6	0.0000	NA	NA	-1
10	511	26.00	-0.2991	-0.5608	38.6974	-6.9	0.0000	NA	NA	-1
11	357	28.70	-0.0773	-0.1187	35.4375	-1.8	0.0717	NA	NA	-1
12	282	28.20	-0.1054	-0.1333	34.7667	-0.3	0.7984	NA	NA	-1

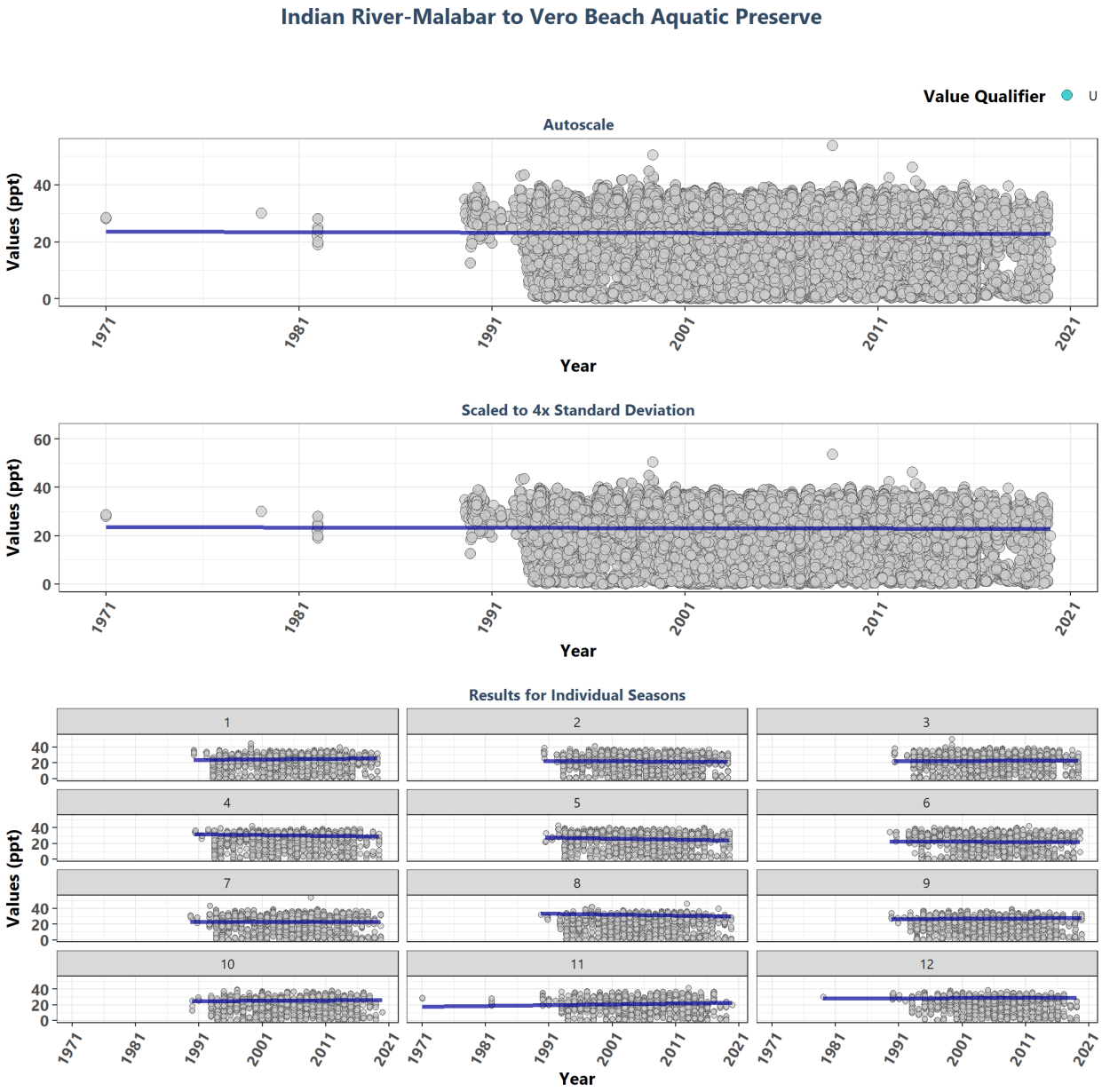
[a] $p < 0.00005$ appear as 0 due to rounding

Guana Tolomato Matanzas National Estuarine Research Reserve



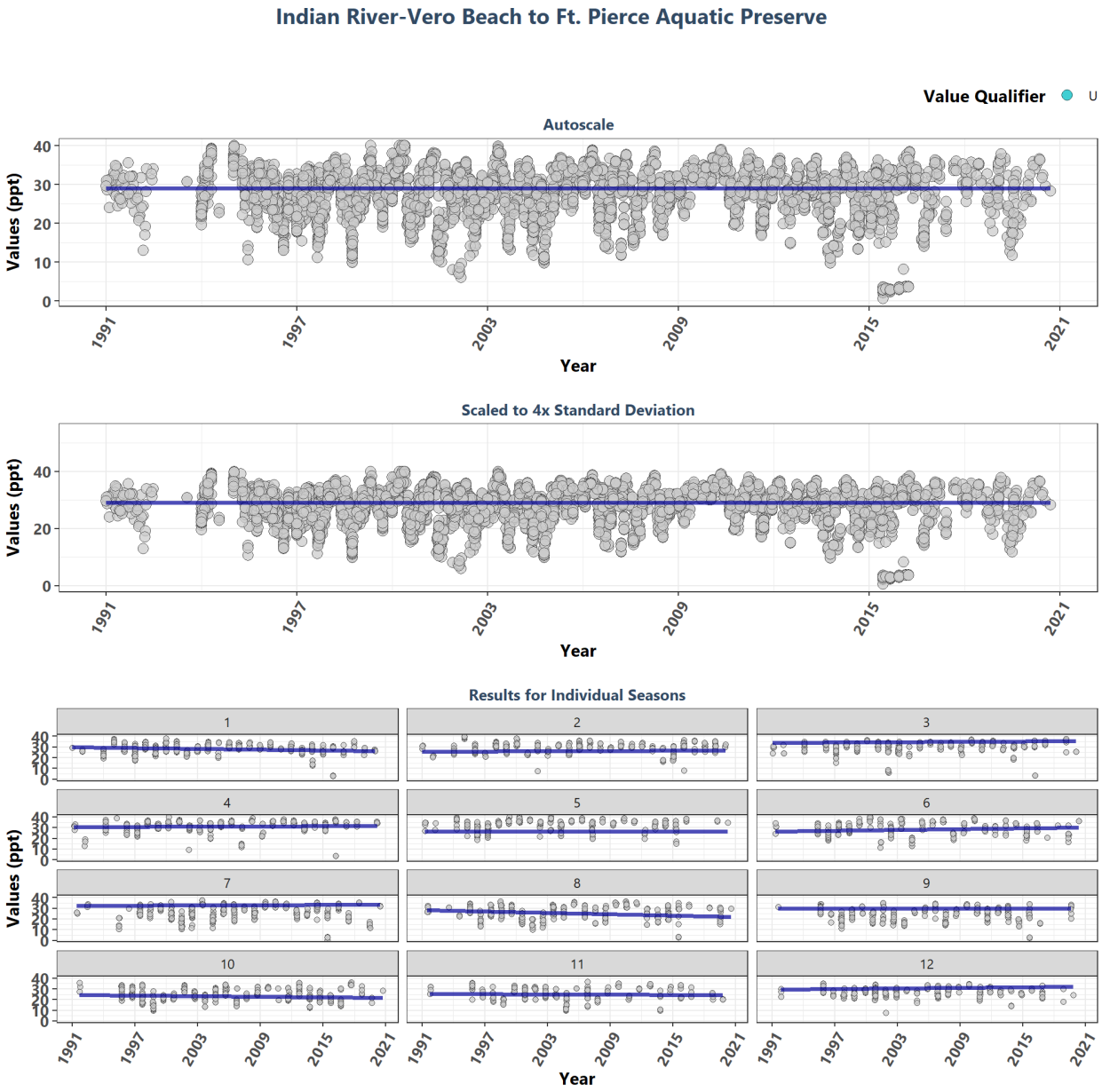
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	14831	32.1	-0.1016	-0.0900	34.4846	-17.9	0.0000	198.8	0	-1
1	1315	31.8	-0.0179	-0.0154	34.1692	-8.5	0.0000	NA	NA	-1
2	1085	31.7	-0.0504	-0.0455	31.2682	-11.7	0.0000	NA	NA	-1
3	1285	31.9	-0.0658	-0.0586	33.9486	-2.7	0.0067	NA	NA	-1
4	1276	32.6	-0.1202	-0.1200	35.9400	-3.5	0.0004	NA	NA	-1
5	1473	33.8	-0.2444	-0.2556	36.7222	-1.0	0.3016	NA	NA	-1
6	1188	33.0	-0.1011	-0.1429	33.5714	-8.2	0.0000	NA	NA	-1
7	1648	33.8	-0.2369	-0.2000	36.1000	-5.3	0.0000	NA	NA	-1
8	1477	33.3	-0.1592	-0.1556	36.1111	-6.9	0.0000	NA	NA	-1
9	964	31.1	-0.0503	-0.0500	32.9000	-11.4	0.0000	NA	NA	-1
10	1333	30.0	0.0536	0.0545	30.3364	-5.5	0.0000	NA	NA	1
11	963	31.7	-0.1565	-0.1200	34.8000	2.5	0.0126	NA	NA	-1
12	824	30.2	-0.0862	-0.0571	35.1143	-2.2	0.0301	NA	NA	-1

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



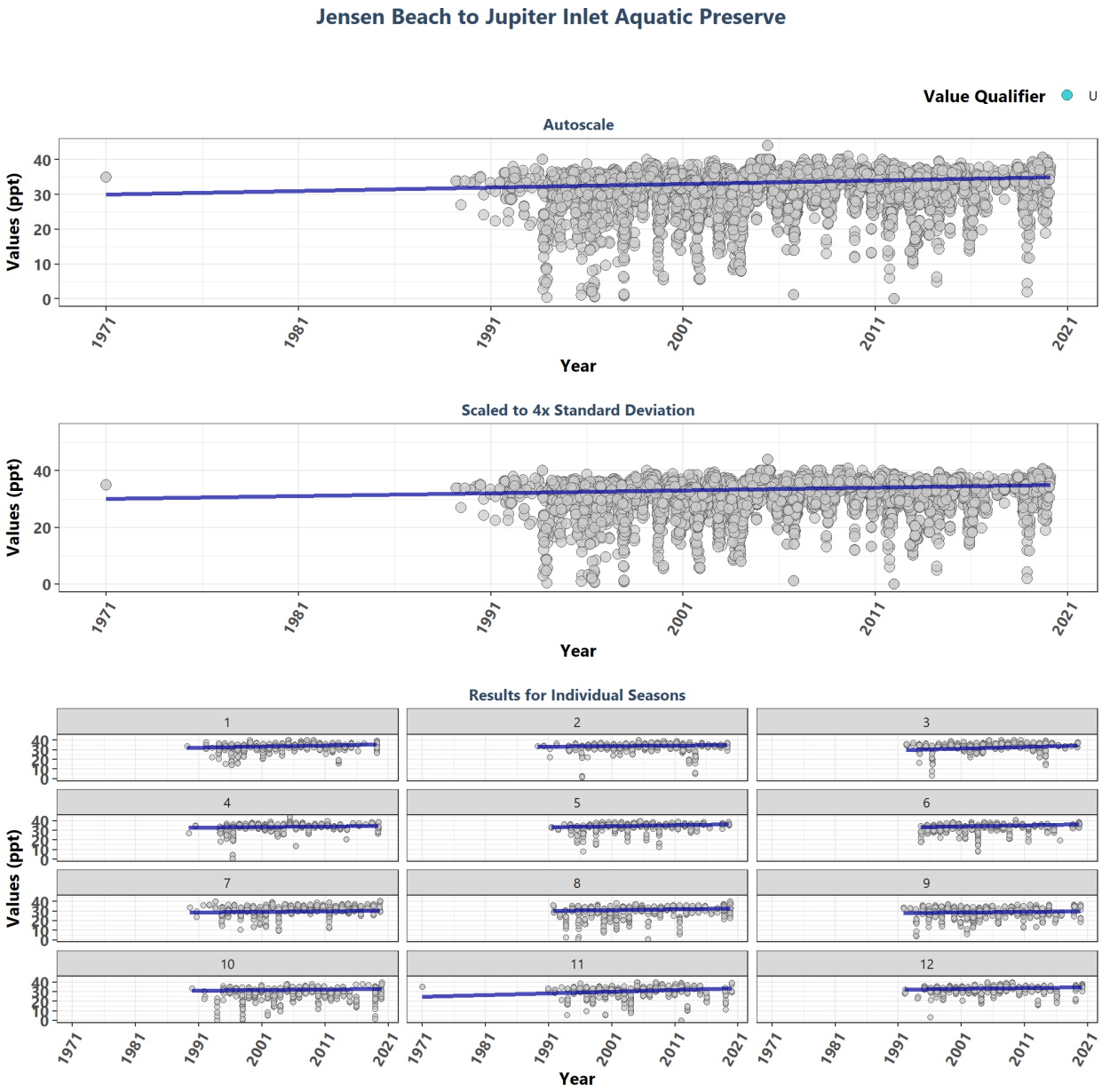
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	42387	25.20	-0.0079	-0.0160	23.5988	-2.6	0.0085	152.8	0	-1
1	3619	22.90	0.0356	0.0706	22.3706	-0.8	0.4311	NA	NA	1
2	3217	24.70	-0.0109	-0.0250	22.7500	3.0	0.0025	NA	NA	-1
3	3148	27.00	0.0072	0.0143	22.1143	2.3	0.0192	NA	NA	1
4	3942	28.80	-0.0617	-0.1000	34.0000	1.2	0.2403	NA	NA	-1
5	3598	31.62	-0.0604	-0.1250	29.8750	-7.5	0.0000	NA	NA	-1
6	3406	30.70	-0.0115	-0.0238	23.0095	-5.4	0.0000	NA	NA	-1
7	4021	25.50	-0.0087	-0.0176	23.5176	-5.7	0.0000	NA	NA	-1
8	3648	25.38	-0.0831	-0.1200	35.7000	2.2	0.0285	NA	NA	-1
9	3684	22.20	0.0278	0.0500	25.3000	-1.0	0.2965	NA	NA	1
10	3851	21.30	0.0242	0.0500	23.6800	4.0	0.0001	NA	NA	1
11	3270	21.90	0.0432	0.1000	17.9000	-0.9	0.3475	NA	NA	1
12	2983	22.60	0.0125	0.0214	28.0450	0.6	0.5548	NA	NA	1

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



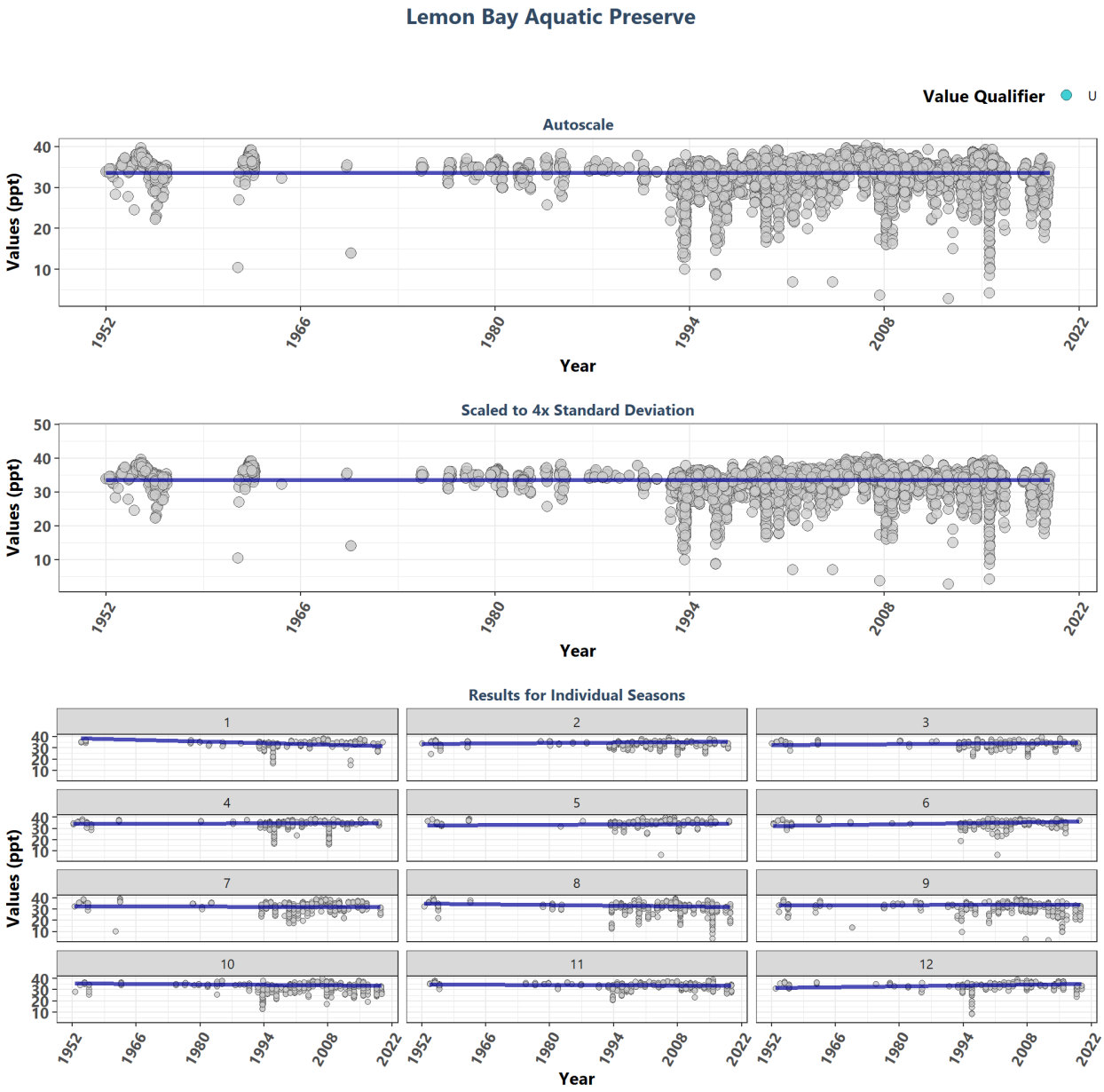
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5788	28.90	0.0002	0.0000	29.0186	0.2	0.8378	62.9	0	0
1	585	28.30	-0.1169	-0.1187	29.8438	-4.2	0.0000	NA	NA	0
2	466	29.70	0.0289	0.0444	25.5111	0.3	0.7985	NA	NA	0
3	459	30.80	0.0647	0.0533	34.1133	1.3	0.1814	NA	NA	0
4	625	32.80	0.0417	0.0362	30.4013	1.8	0.0654	NA	NA	0
5	487	34.70	-0.0149	-0.0143	26.6771	2.1	0.0324	NA	NA	0
6	352	30.36	0.0827	0.1182	26.6273	1.7	0.0929	NA	NA	0
7	645	28.40	0.0492	0.0400	32.2800	3.1	0.0016	NA	NA	0
8	429	26.00	-0.1371	-0.2100	28.4300	0.9	0.3707	NA	NA	0
9	424	24.90	0.0079	0.0059	29.6294	-0.5	0.6138	NA	NA	0
10	606	23.40	-0.0721	-0.0917	24.5917	-2.7	0.0078	NA	NA	0
11	334	26.12	-0.0164	-0.0250	25.2750	-3.7	0.0002	NA	NA	0
12	376	26.52	0.0598	0.0842	29.6071	-0.4	0.6655	NA	NA	0

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



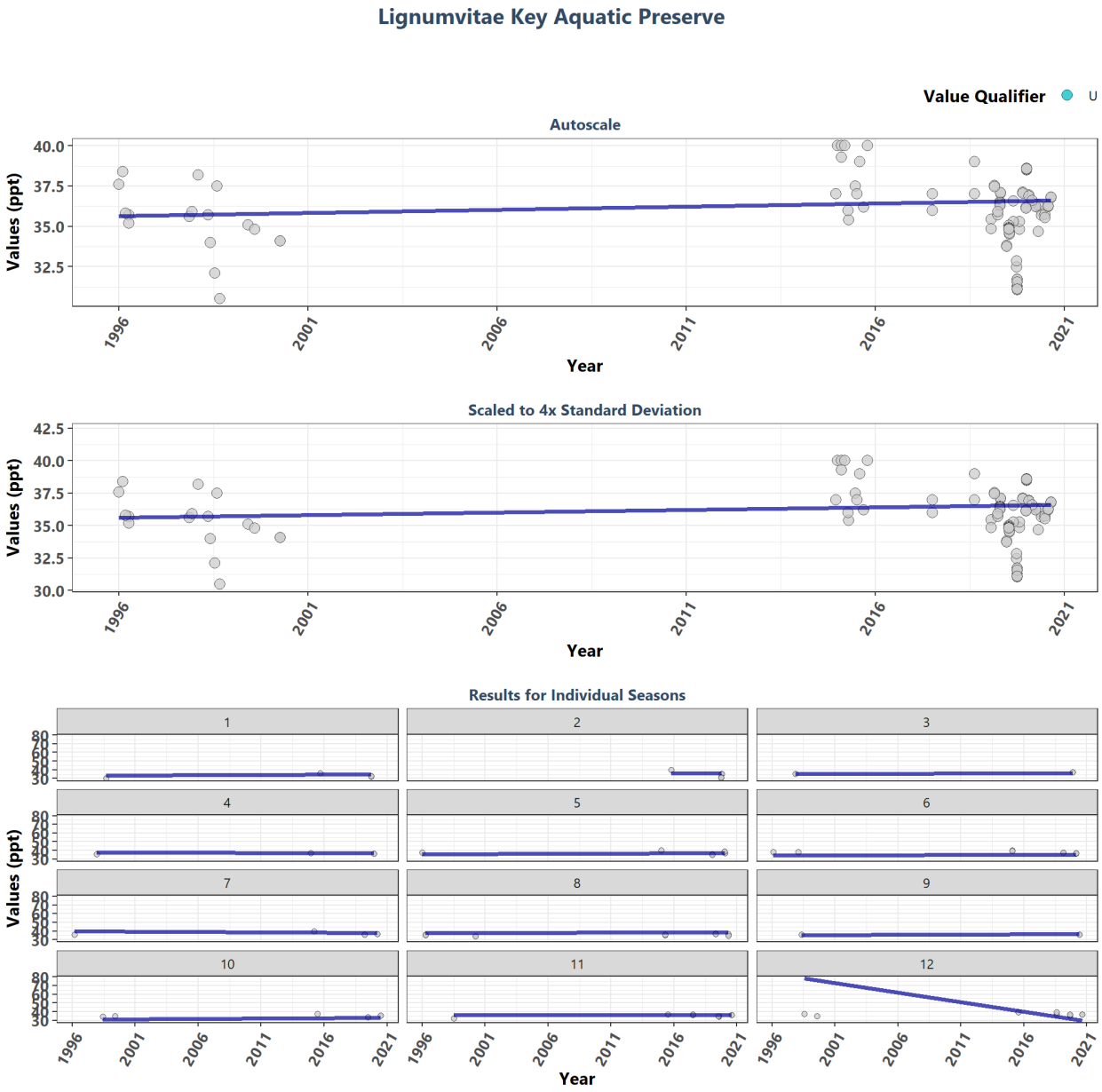
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7278	32.80	0.1123	0.1000	30.0141	14.5	0.0000	38.1	0.0001	1
1	585	33.20	0.1588	0.1154	30.1615	3.9	0.0001	NA	NA	1
2	611	34.00	0.0909	0.0667	31.9333	3.3	0.0011	NA	NA	1
3	559	34.20	0.1815	0.1589	26.6400	3.2	0.0013	NA	NA	1
4	587	34.70	0.0878	0.0550	32.0200	5.2	0.0000	NA	NA	1
5	601	34.80	0.1421	0.1000	31.3000	5.7	0.0000	NA	NA	1
6	592	34.20	0.1541	0.1000	31.4000	5.8	0.0000	NA	NA	1
7	699	32.20	0.0421	0.0571	27.2571	7.2	0.0000	NA	NA	1
8	662	30.85	0.0673	0.0800	28.6000	6.4	0.0000	NA	NA	1
9	588	29.20	0.0559	0.0750	26.1750	1.5	0.1266	NA	NA	1
10	604	28.80	0.0752	0.0667	29.8667	2.1	0.0396	NA	NA	1
11	649	31.40	0.1664	0.1775	24.6375	2.6	0.0100	NA	NA	1
12	541	32.20	0.1070	0.0833	30.2833	2.6	0.0088	NA	NA	1

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



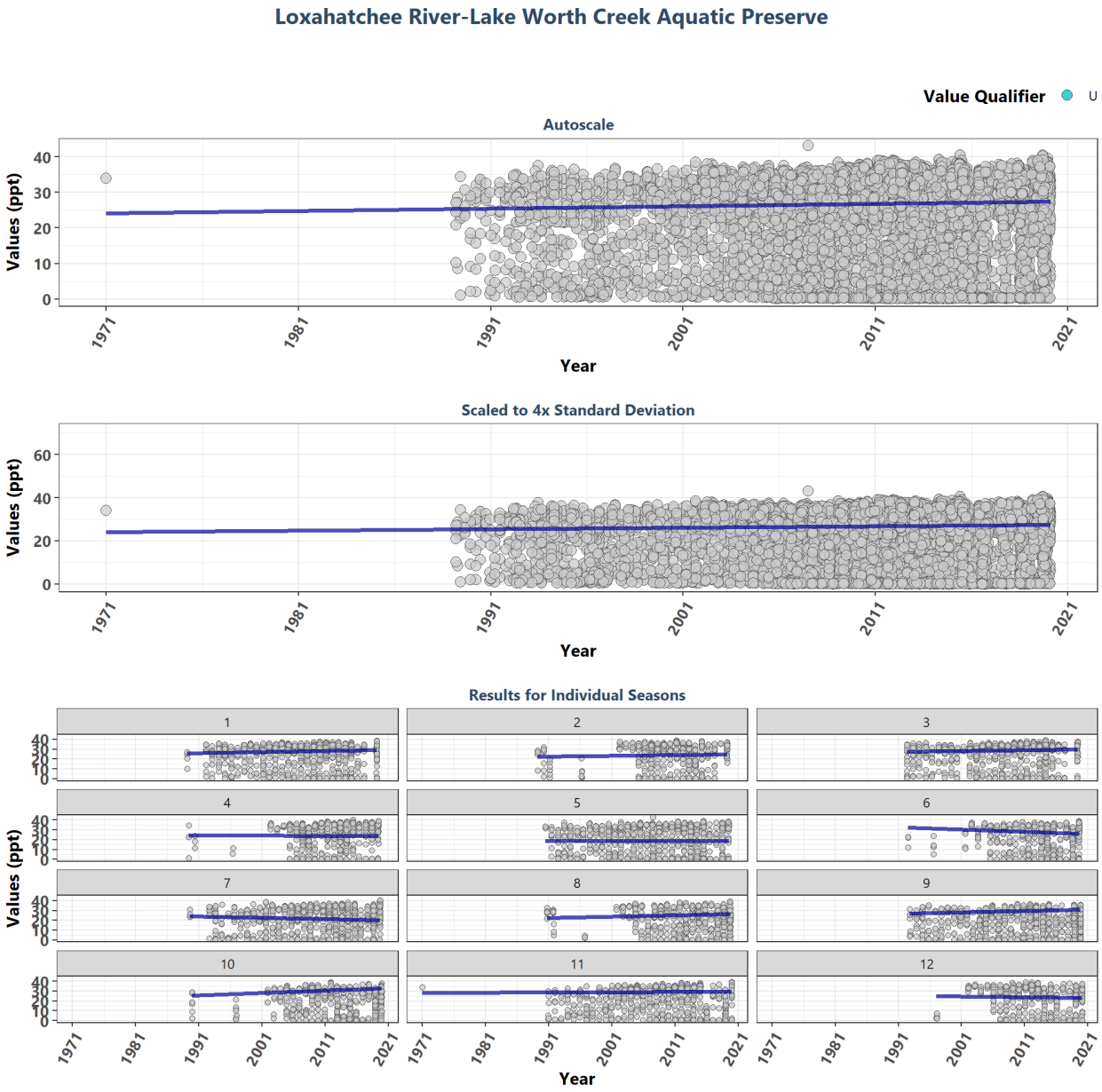
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5907	34.00	0.0083	0.0000	33.5425	0.3	0.7761	161.8	0	0
1	314	34.20	-0.2200	-0.1053	38.6895	4.2	0.0000	NA	NA	0
2	445	33.80	0.1127	0.0300	33.5600	-2.2	0.0308	NA	NA	0
3	409	34.60	0.1163	0.0329	32.3929	0.4	0.6606	NA	NA	0
4	526	34.00	0.0145	0.0038	34.3962	-2.8	0.0050	NA	NA	0
5	375	35.00	0.0460	0.0259	32.5525	3.3	0.0011	NA	NA	0
6	515	35.20	0.1830	0.0556	32.4222	6.2	0.0000	NA	NA	0
7	590	33.90	-0.0149	-0.0068	32.6461	1.7	0.0938	NA	NA	0
8	549	32.49	-0.0810	-0.0486	35.1614	-2.8	0.0044	NA	NA	0
9	522	32.90	0.0274	0.0083	33.5250	-7.5	0.0000	NA	NA	0
10	648	32.30	-0.0814	-0.0311	35.5568	-0.6	0.5691	NA	NA	0
11	520	34.00	-0.0684	-0.0180	34.7000	0.9	0.3488	NA	NA	0
12	494	34.20	0.1587	0.0533	31.5333	3.9	0.0001	NA	NA	0

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



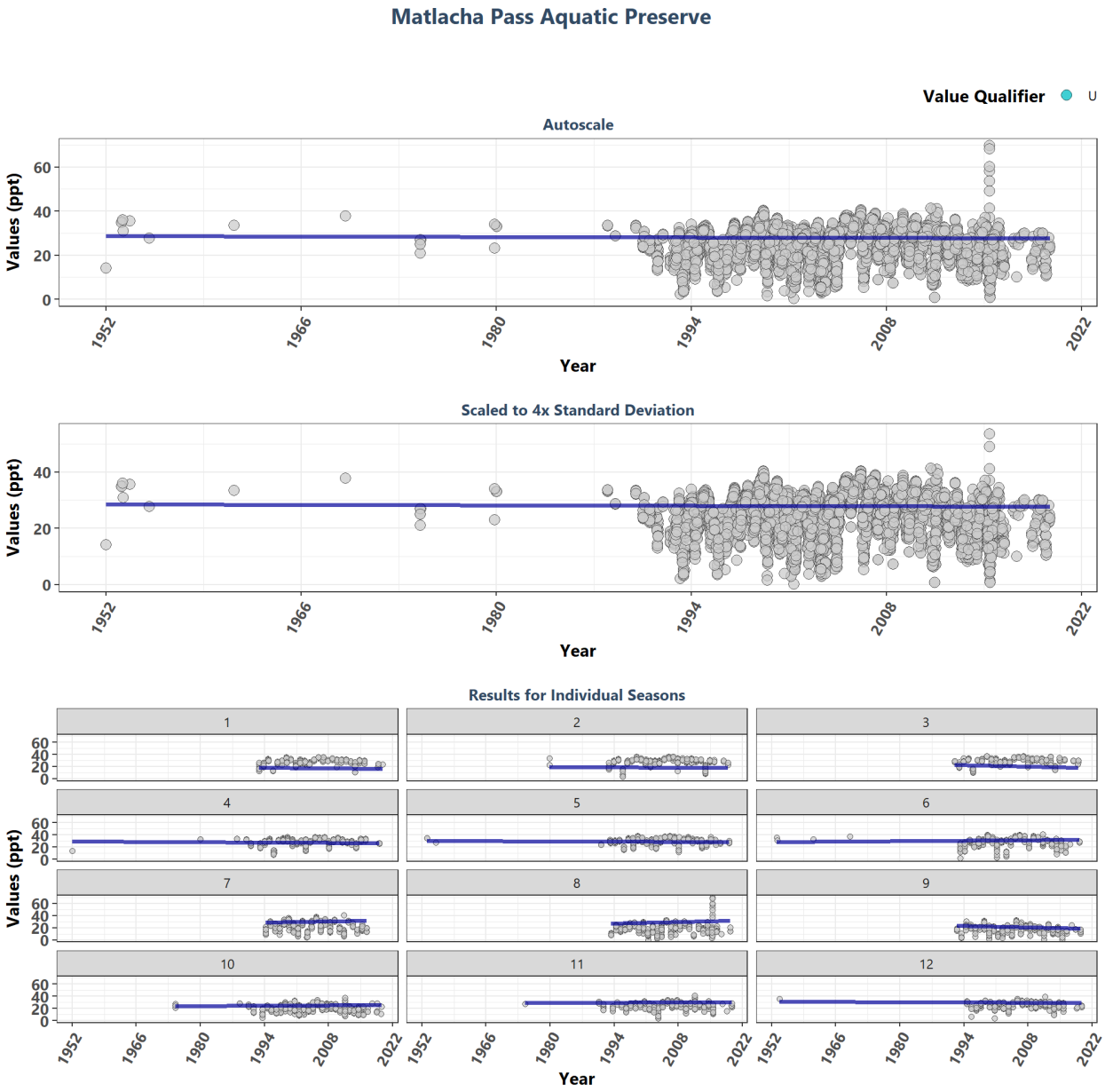
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	121	36.12	0.0538	0.0396	35.6271	1.3	0.1788	16.2	0.1325	0
1	4	32.66	0.0087	0.0700	33.2500	0.0	1.0000	NA	NA	0
2	15	NA	0.2000	0.0179	35.6579	NA	NA	NA	NA	NA
3	3	NA	0.0000	0.0043	35.5964	NA	NA	NA	NA	NA
4	4	36.16	-0.1667	-0.0336	37.5500	0.0	1.0000	NA	NA	0
5	18	38.55	0.6667	0.0677	35.4245	1.1	0.2877	NA	NA	0
6	8	37.86	0.1429	0.0310	34.3881	-2.0	0.0427	NA	NA	0
7	6	36.07	-0.6071	-0.1000	39.9600	0.4	0.6919	NA	NA	0
8	22	36.31	0.1373	0.0413	37.5600	2.3	0.0228	NA	NA	0
9	3	NA	0.3030	0.0580	34.9800	NA	NA	NA	NA	NA
10	7	35.10	0.1667	0.0933	30.6067	0.3	0.7585	NA	NA	0
11	22	34.86	0.1667	0.0100	35.9400	0.0	0.9709	NA	NA	0
12	9	36.81	-0.1333	-2.2075	84.1800	-0.5	0.5942	NA	NA	0

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



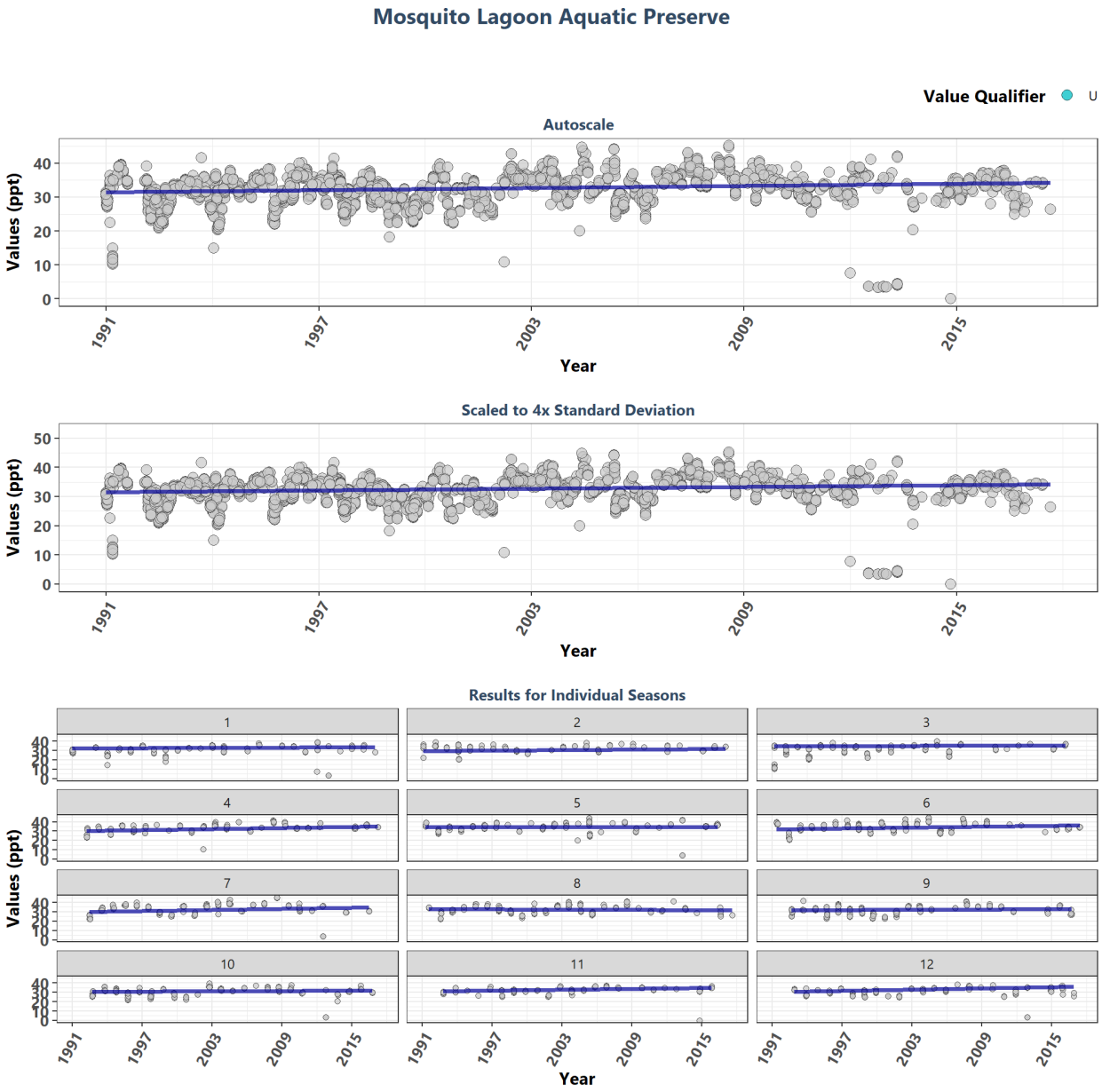
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	7066	26.20	0.0306	0.0667	24.1703	4.6	0.0000	60.7	0	1
1	724	29.15	0.0775	0.1231	23.3731	2.0	0.0446	NA	NA	1
2	441	29.10	0.0386	0.0667	21.5333	0.4	0.6732	NA	NA	1
3	673	29.30	0.0498	0.0800	26.1100	3.9	0.0001	NA	NA	1
4	505	31.20	-0.0140	-0.0225	24.9675	4.4	0.0000	NA	NA	-1
5	760	28.05	0.0084	0.0100	18.3000	3.2	0.0014	NA	NA	1
6	473	24.00	-0.1033	-0.2273	36.9955	-0.5	0.6475	NA	NA	-1
7	708	24.20	-0.0523	-0.1345	26.6511	1.5	0.1239	NA	NA	-1
8	498	23.65	0.0619	0.1333	19.8833	-1.1	0.2865	NA	NA	1
9	673	18.70	0.1009	0.1600	23.2200	0.3	0.7437	NA	NA	1
10	483	21.00	0.1293	0.2500	20.4500	-1.7	0.0848	NA	NA	1
11	686	24.95	0.0134	0.0273	28.0091	2.4	0.0151	NA	NA	1
12	442	27.45	-0.0319	-0.0667	26.4833	-3.3	0.0011	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	6042	26.20	-0.0065	-0.0125	28.5901	-1.2	0.2176	78.4	0	0
1	416	28.70	-0.0533	-0.0818	21.7414	4.9	0.0000	NA	NA	0
2	504	29.56	-0.0220	-0.0333	20.1333	-1.0	0.3179	NA	NA	0
3	462	30.01	-0.1440	-0.1696	29.2174	0.6	0.5212	NA	NA	0
4	549	29.30	-0.0437	-0.0500	29.6250	-0.8	0.4167	NA	NA	0
5	539	31.60	-0.0232	-0.0250	30.6000	2.0	0.0431	NA	NA	0
6	503	29.90	0.0582	0.0647	28.2353	2.9	0.0044	NA	NA	0
7	508	22.00	0.0849	0.1600	21.7400	-2.3	0.0201	NA	NA	0
8	508	18.40	0.1593	0.1800	19.3400	-0.7	0.4582	NA	NA	0
9	470	17.41	-0.0689	-0.1714	30.7429	-1.7	0.0837	NA	NA	0
10	615	20.40	0.0329	0.0400	23.1100	-5.3	0.0000	NA	NA	0
11	536	25.23	0.0200	0.0200	28.9450	1.1	0.2542	NA	NA	0
12	432	26.95	-0.0297	-0.0333	31.3600	-1.4	0.1738	NA	NA	0

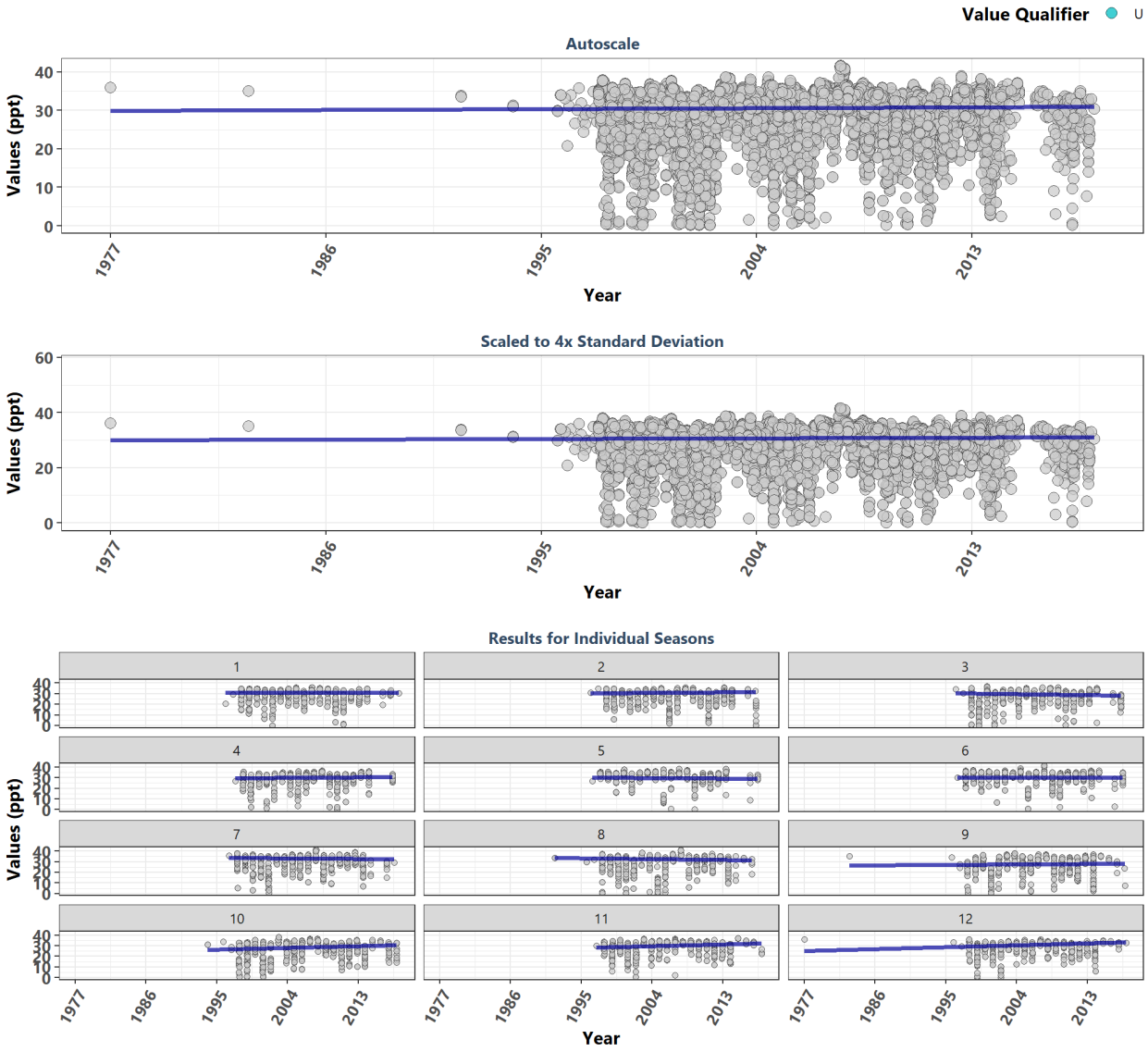
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	3808	32.30	0.1005	0.1000	31.5247	8.8	0.0000	74.4	0	1
1	342	31.40	0.0230	0.0375	32.4625	6.1	0.0000	NA	NA	1
2	233	32.60	0.1130	0.1000	29.4000	-1.2	0.2385	NA	NA	1
3	288	32.11	0.0241	0.0294	34.5941	3.4	0.0007	NA	NA	1
4	289	34.00	0.1615	0.2062	29.8500	4.8	0.0000	NA	NA	1
5	275	34.50	-0.0046	0.0000	34.5000	-0.1	0.9094	NA	NA	-1
6	372	34.80	0.1890	0.1612	32.2271	0.7	0.4862	NA	NA	1
7	379	32.80	0.2215	0.1848	29.7367	0.7	0.5031	NA	NA	1
8	391	32.20	-0.0518	-0.0333	32.8333	1.2	0.2396	NA	NA	-1
9	374	31.50	0.0398	0.0456	31.7444	4.7	0.0000	NA	NA	1
10	363	30.20	0.0796	0.0688	30.4500	3.2	0.0013	NA	NA	1
11	210	32.45	0.1335	0.1333	31.3050	7.1	0.0000	NA	NA	1
12	292	31.00	0.3299	0.2250	30.2000	2.0	0.0419	NA	NA	1

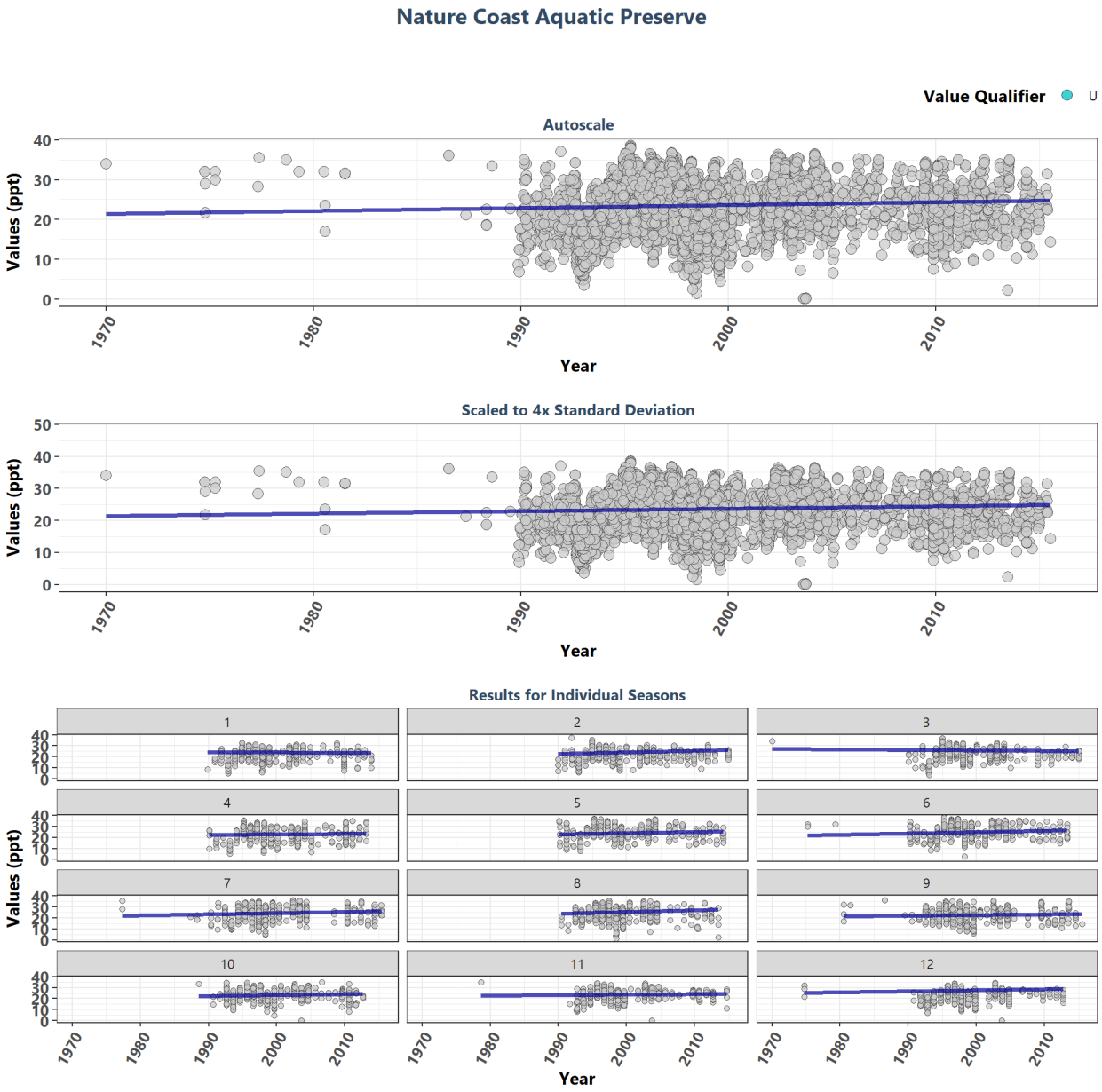
[a] $p < 0.00005$ appear as 0 due to rounding

Nassau River-St. Johns River Marshes Aquatic Preserve



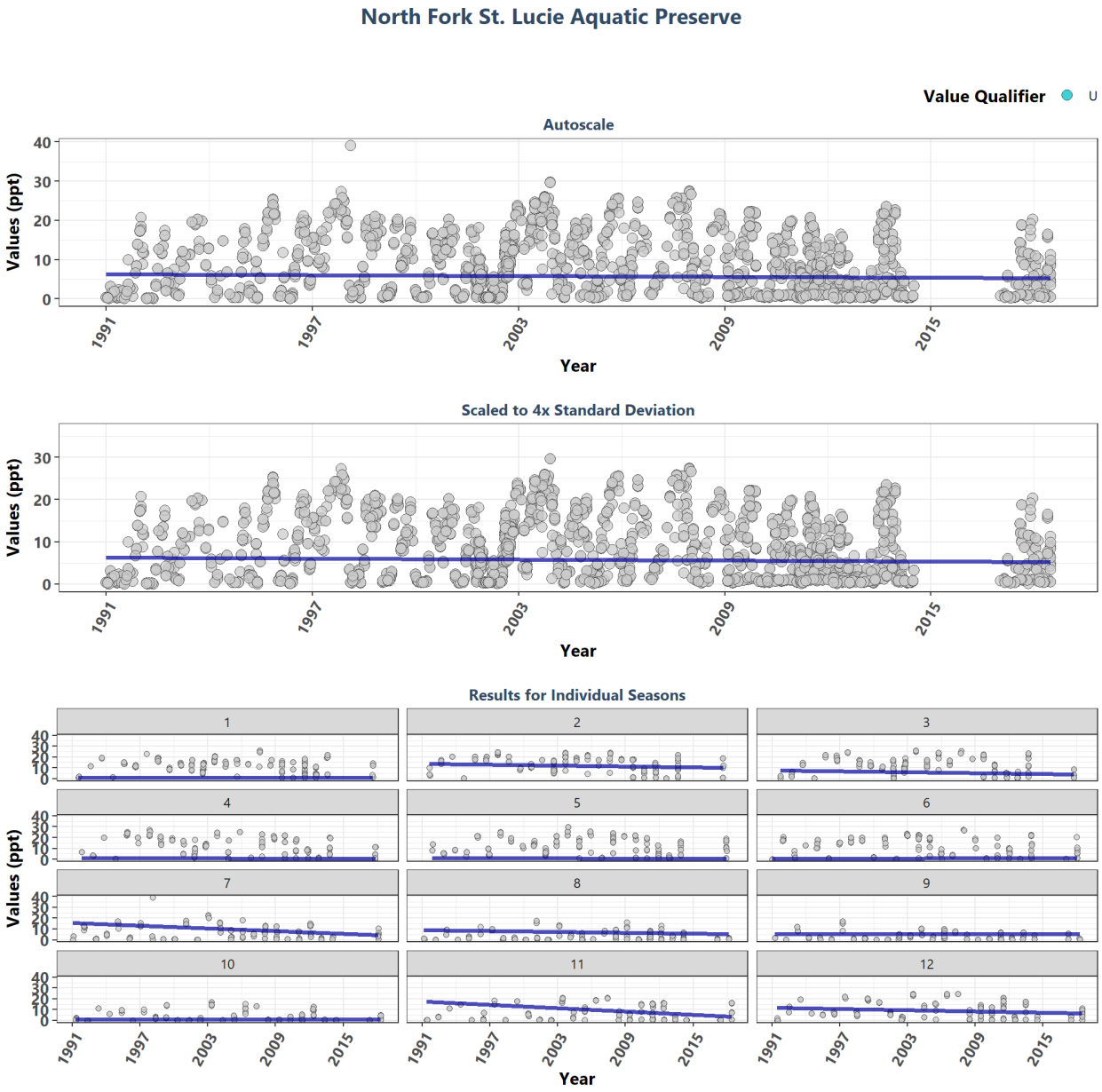
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	9178	30.40	0.0152	0.0250	29.9708	2.2	0.0259	88.4	0	1
1	754	29.60	-0.0015	0.0000	30.8000	-1.8	0.0709	NA	NA	-1
2	738	29.20	0.0297	0.0500	29.6000	-2.4	0.0144	NA	NA	1
3	774	28.65	-0.0601	-0.1000	32.2000	4.8	0.0000	NA	NA	-1
4	744	30.20	0.0410	0.0800	27.6800	4.9	0.0000	NA	NA	1
5	779	33.00	-0.0439	-0.0615	31.4462	-2.8	0.0052	NA	NA	-1
6	730	32.10	-0.0067	-0.0083	30.3417	-2.8	0.0060	NA	NA	-1
7	827	31.00	-0.0667	-0.0688	35.0633	1.3	0.2008	NA	NA	-1
8	747	30.80	-0.0679	-0.0826	34.5775	3.9	0.0001	NA	NA	-1
9	766	30.00	0.0237	0.0500	26.3500	1.7	0.0889	NA	NA	1
10	780	27.80	0.1155	0.2000	22.6500	1.0	0.3206	NA	NA	1
11	789	30.80	0.1205	0.1667	25.2000	-0.1	0.9494	NA	NA	1
12	750	30.10	0.0958	0.2000	25.0000	-0.3	0.7823	NA	NA	1

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



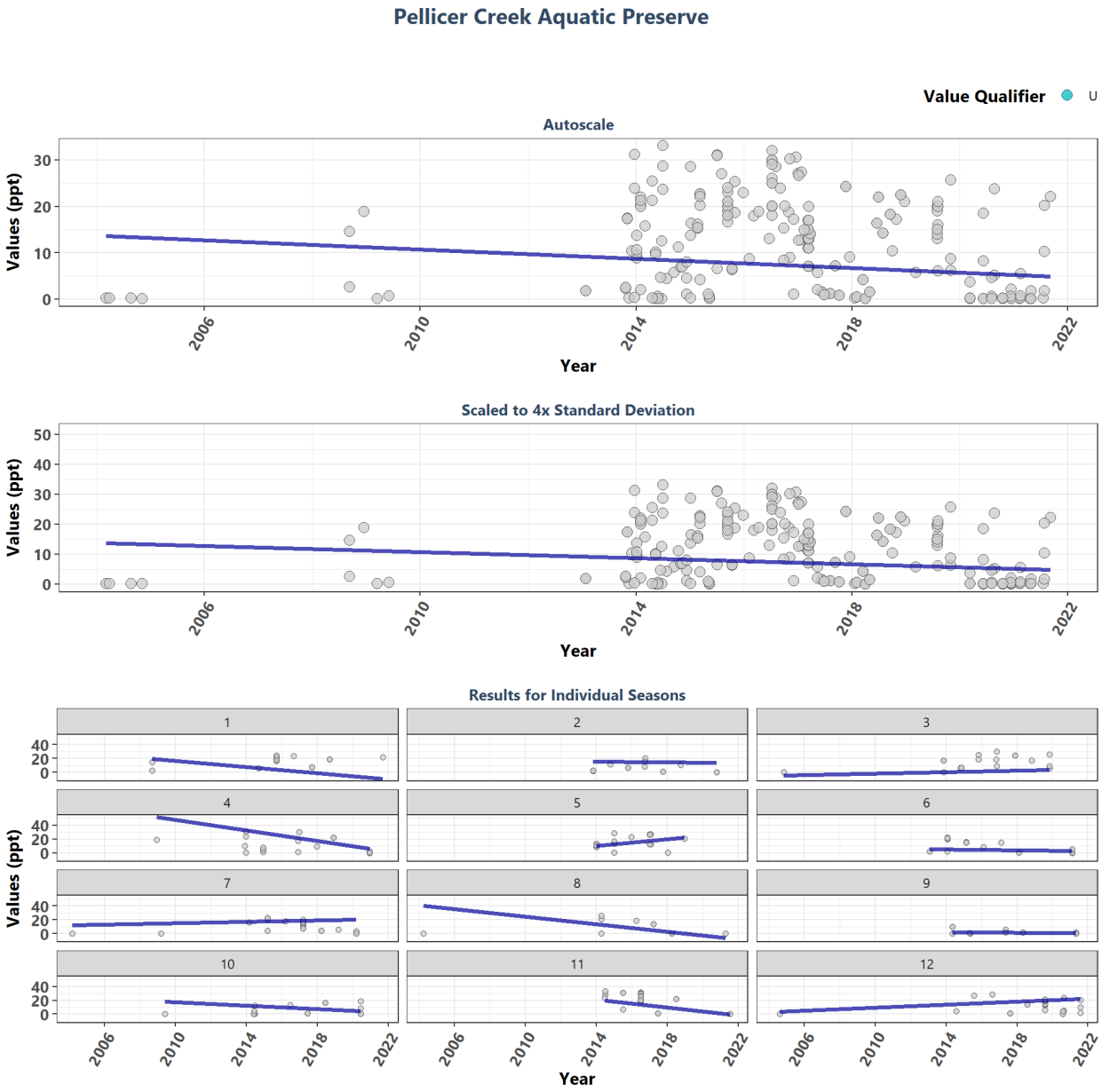
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5710	24.18	0.0432	0.0727	21.4605	4.8	0.0000	13.4	0.2685	1
1	489	22.73	-0.0086	-0.0140	24.2920	1.2	0.2211	NA	NA	-1
2	502	22.44	0.0924	0.1525	19.5000	1.2	0.2260	NA	NA	1
3	452	23.00	-0.0233	-0.0450	27.3050	1.7	0.0939	NA	NA	-1
4	478	23.59	0.0369	0.0569	21.0792	0.8	0.4242	NA	NA	1
5	468	26.00	0.0644	0.1071	20.7329	-0.8	0.4503	NA	NA	1
6	487	27.32	0.0755	0.1188	21.4650	1.5	0.1242	NA	NA	1
7	490	25.40	0.0512	0.1033	21.4767	2.6	0.0095	NA	NA	1
8	440	24.37	0.0782	0.1360	21.4560	1.6	0.1078	NA	NA	1
9	492	23.84	0.0361	0.0550	20.8500	2.1	0.0324	NA	NA	1
10	519	23.90	0.0526	0.0900	20.4800	-0.3	0.7698	NA	NA	1
11	440	24.79	0.0244	0.0450	22.2850	2.4	0.0176	NA	NA	1
12	453	23.77	0.0465	0.0888	24.8350	2.9	0.0032	NA	NA	1

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



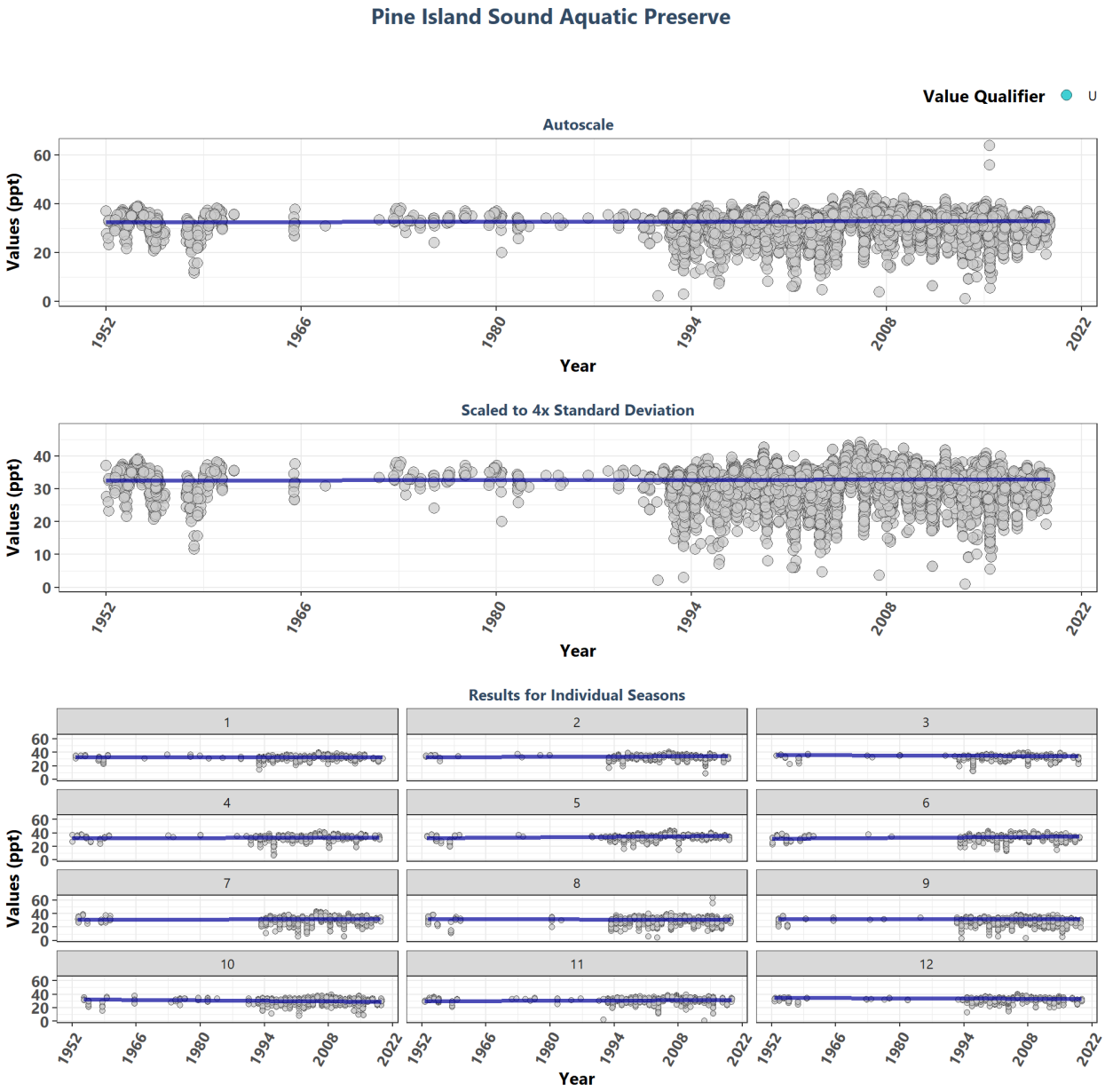
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2887	4.40	-0.0737	-0.0385	6.3005	-5.8	0.0000	54.2	0	-1
1	234	11.62	0.0439	0.0056	0.7056	-2.0	0.0496	NA	NA	1
2	234	9.98	-0.0857	-0.1364	13.7386	-5.6	0.0000	NA	NA	-1
3	243	9.44	-0.1185	-0.1250	7.4000	-2.3	0.0226	NA	NA	-1
4	214	10.18	-0.1212	-0.0260	1.2680	-5.0	0.0000	NA	NA	-1
5	249	7.22	-0.0491	-0.0089	0.8600	-2.0	0.0440	NA	NA	-1
6	228	5.26	0.0619	0.0120	0.7840	0.2	0.8050	NA	NA	1
7	242	0.80	-0.2269	-0.4286	15.7514	0.2	0.8218	NA	NA	-2
8	255	0.80	-0.0850	-0.1340	8.9620	1.0	0.2944	NA	NA	-2
9	276	0.80	0.0110	0.0045	5.2009	-3.0	0.0026	NA	NA	1
10	233	0.70	0.0097	0.0016	0.7762	-1.1	0.2632	NA	NA	1
11	240	1.00	-0.2440	-0.4988	17.4620	1.4	0.1515	NA	NA	-2
12	239	5.40	-0.0974	-0.1838	11.6450	-2.7	0.0062	NA	NA	-1

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



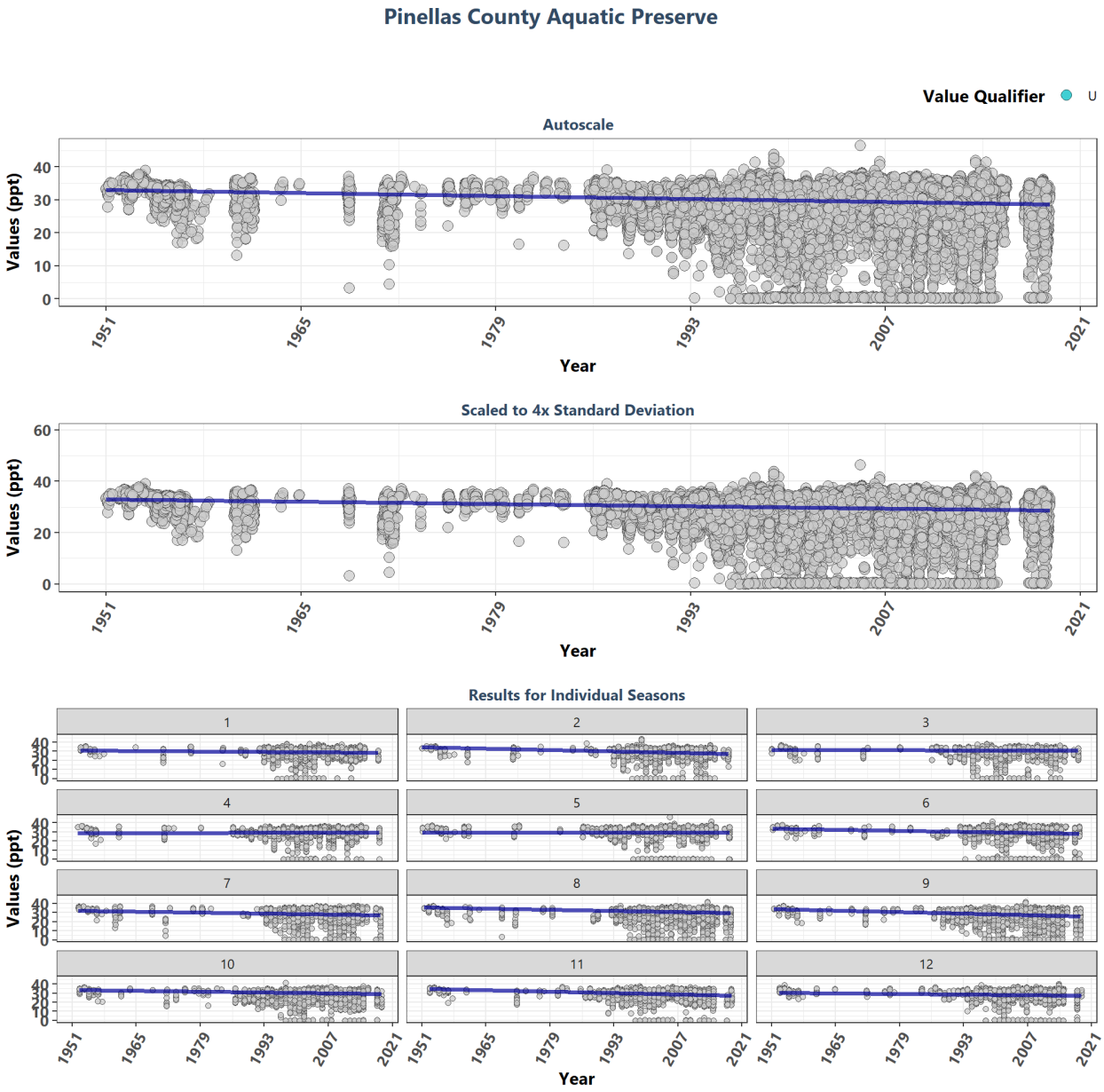
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	222	12.75	-0.1133	-0.5000	14.6983	-2.6	0.0097	31.9	0.0008	-1
1	21	18.29	-0.4561	-2.1964	34.0336	1.3	0.1986	NA	NA	-2
2	14	4.38	-0.0277	-0.2400	18.2000	-0.7	0.4685	NA	NA	-1
3	19	17.30	0.2417	0.4967	-5.6817	1.5	0.1269	NA	NA	1
4	18	8.48	-0.3953	-3.8350	78.6900	-1.7	0.0968	NA	NA	-2
5	16	13.72	0.2000	2.3620	-18.1670	1.1	0.2832	NA	NA	2
6	19	5.48	-0.1538	-0.3164	9.1264	-2.8	0.0058	NA	NA	-1
7	27	12.81	0.1857	0.5067	11.1967	-2.6	0.0082	NA	NA	1
8	10	7.10	-0.4222	-2.7310	46.6988	-1.7	0.0969	NA	NA	-2
9	13	1.46	-0.1282	-0.1517	3.7350	-0.6	0.5722	NA	NA	-2
10	16	1.52	-0.2876	-1.2733	27.5850	1.3	0.1882	NA	NA	-2
11	23	25.00	-0.3447	-2.8900	56.1600	-2.8	0.0051	NA	NA	-2
12	26	14.12	0.2573	1.1282	0.3768	-0.2	0.8547	NA	NA	1

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



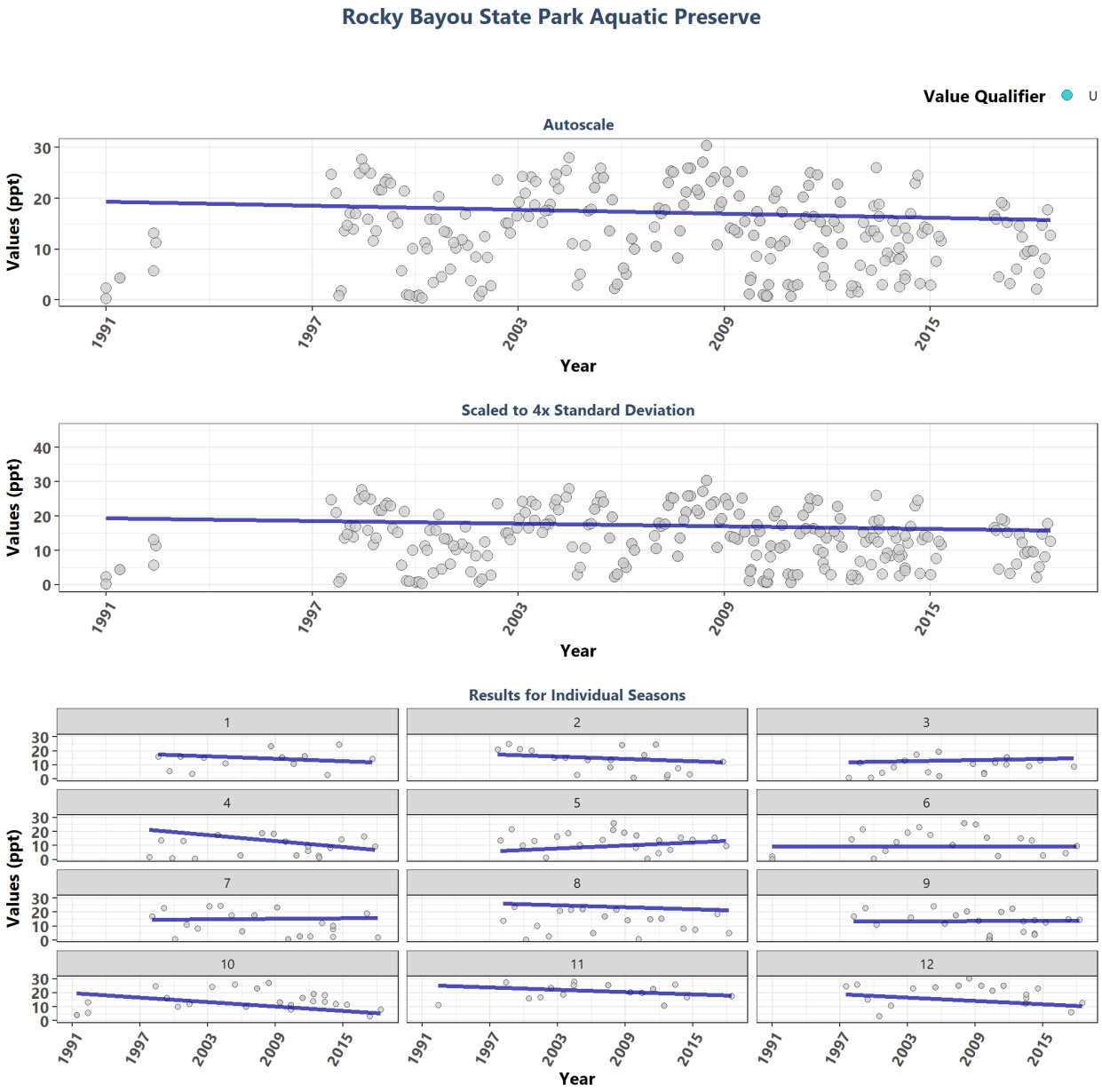
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	19710	32.8	0.0117	0.0056	32.4768	2.5	0.0141	112.5	0	1
1	1430	33.0	-0.0276	-0.0098	33.2171	1.0	0.3404	NA	NA	-1
2	1328	33.6	0.0439	0.0211	33.0792	-2.0	0.0464	NA	NA	1
3	1514	34.2	-0.0393	-0.0250	36.0500	2.6	0.0104	NA	NA	-1
4	1630	34.3	0.0168	0.0071	32.6071	6.1	0.0000	NA	NA	1
5	1624	34.9	0.1021	0.0482	32.3464	6.2	0.0000	NA	NA	1
6	1610	34.7	0.1012	0.0492	31.7408	-2.4	0.0179	NA	NA	1
7	1830	32.1	0.0325	0.0125	31.3250	-0.1	0.8849	NA	NA	1
8	1799	30.9	-0.0185	-0.0143	31.7000	-1.2	0.2384	NA	NA	-1
9	1756	29.6	-0.0023	0.0000	32.1000	-3.8	0.0001	NA	NA	-1
10	1930	30.8	-0.0609	-0.0569	32.7308	1.7	0.0904	NA	NA	-1
11	1746	32.0	0.0257	0.0167	29.8833	2.0	0.0417	NA	NA	1
12	1513	32.7	-0.0364	-0.0167	34.5000	-1.6	0.1076	NA	NA	-1

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



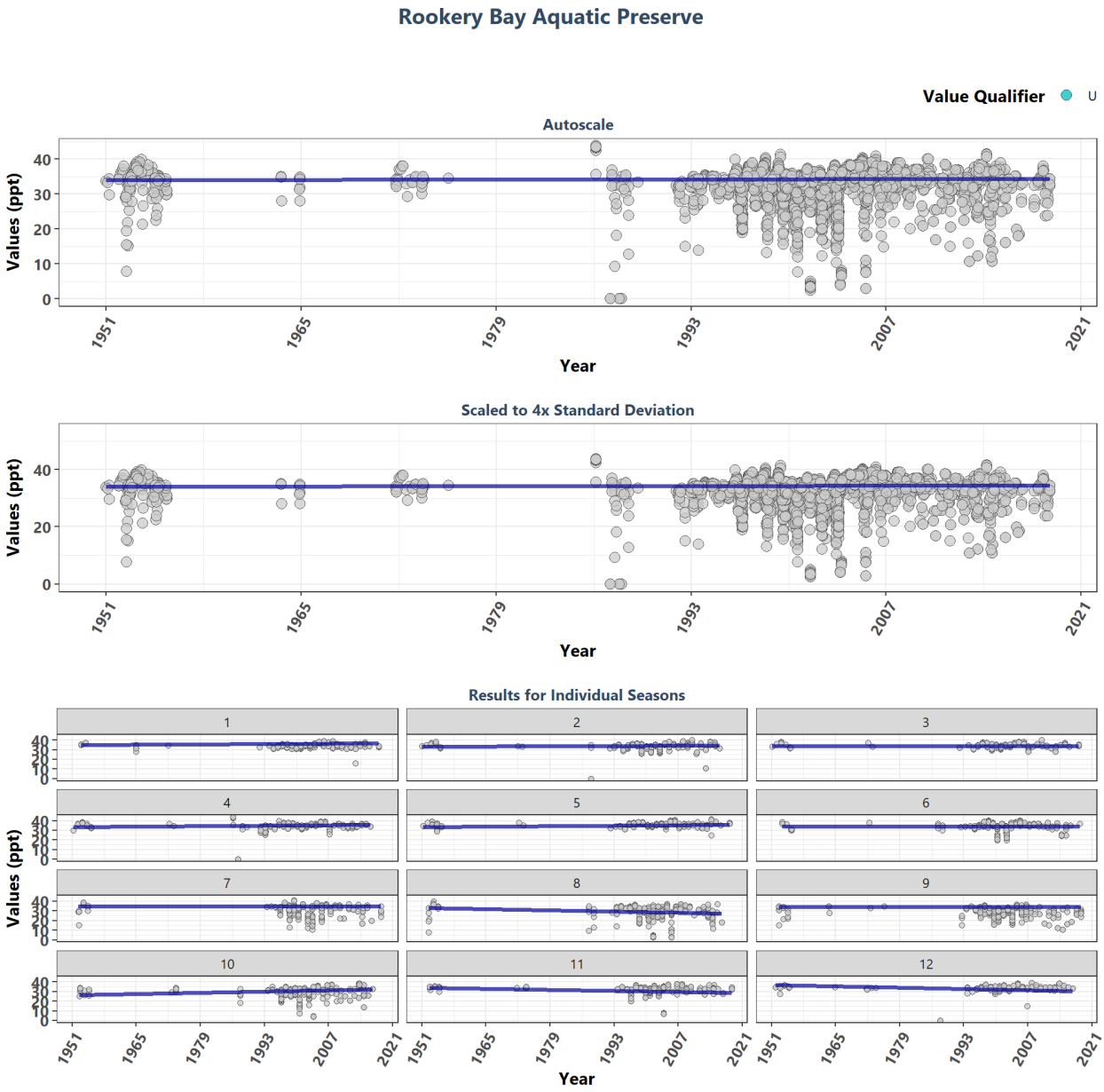
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	54257	29.20	-0.0743	-0.0625	32.9468	-26.2	0.0000	329.7	0	-1
1	3692	29.40	-0.0429	-0.0330	30.9210	0.7	0.5089	NA	NA	-1
2	3648	29.18	-0.1469	-0.1163	34.7474	-9.5	0.0000	NA	NA	-1
3	3722	29.20	-0.0179	-0.0117	31.7800	-3.9	0.0001	NA	NA	-1
4	3962	29.34	0.0139	0.0104	28.7992	1.3	0.1891	NA	NA	1
5	5545	31.15	0.0072	0.0058	29.0967	-2.0	0.0449	NA	NA	1
6	4565	30.80	-0.1043	-0.0846	33.6646	-11.0	0.0000	NA	NA	-1
7	4043	30.00	-0.1003	-0.0790	32.2290	-8.1	0.0000	NA	NA	-1
8	4909	28.10	-0.1083	-0.0860	35.4440	-4.8	0.0000	NA	NA	-1
9	6357	27.60	-0.1124	-0.1194	33.8111	-13.5	0.0000	NA	NA	-1
10	5328	28.20	-0.0850	-0.0689	33.7200	-11.0	0.0000	NA	NA	-1
11	4090	28.70	-0.1281	-0.1100	34.6700	-14.1	0.0000	NA	NA	-1
12	4396	28.73	-0.0453	-0.0443	30.4914	-12.7	0.0000	NA	NA	-1

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



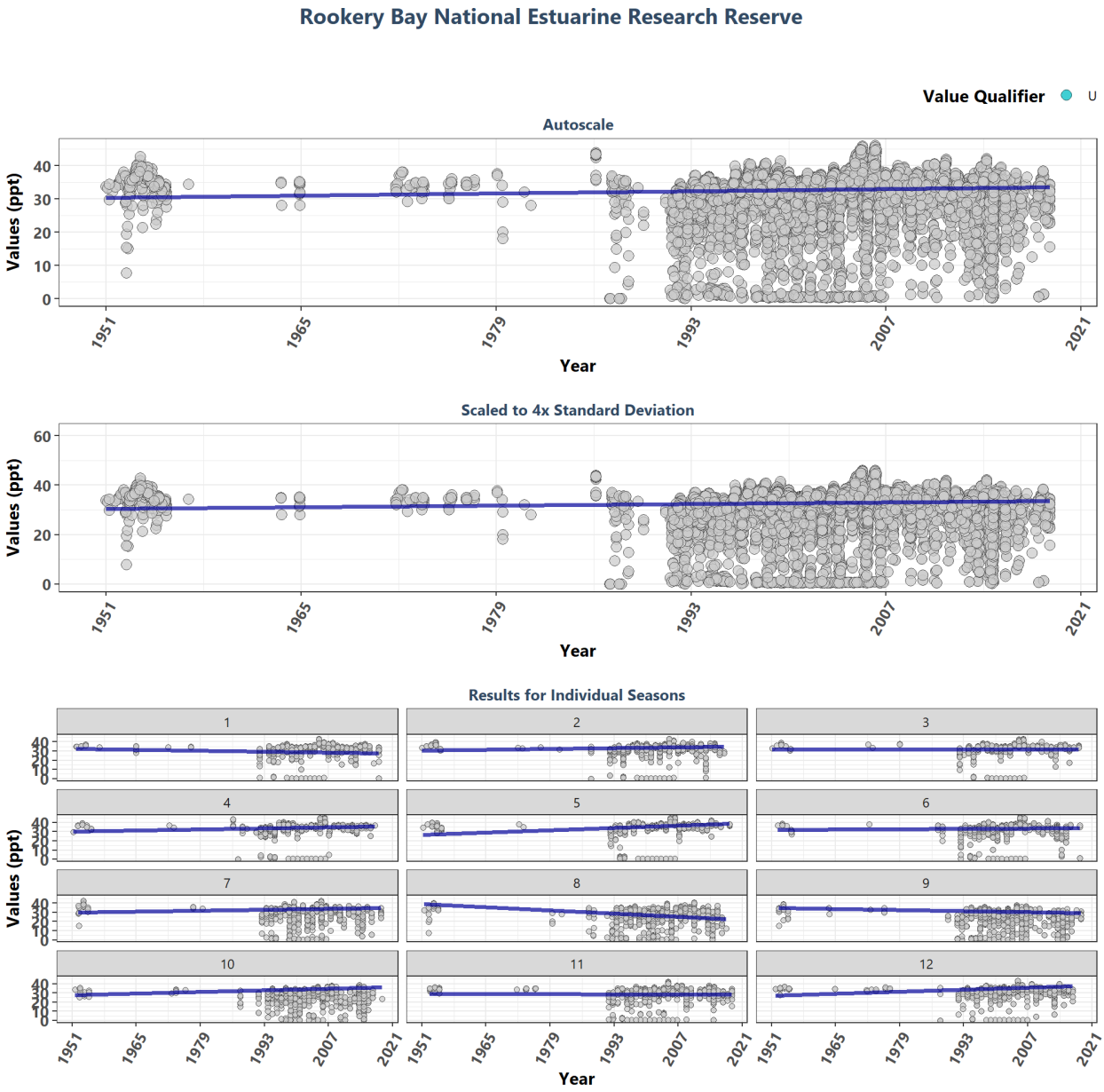
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	305	14.08	-0.0693	-0.1295	19.3584	-1.7	0.0923	9.5	0.5717	0
1	17	15.40	-0.0985	-0.2885	19.6198	0.3	0.7716	NA	NA	0
2	26	13.58	-0.1264	-0.2727	19.1568	-1.9	0.0627	NA	NA	0
3	25	10.20	0.0568	0.1393	10.8826	1.3	0.1960	NA	NA	0
4	25	9.42	-0.2615	-0.7171	26.1300	0.0	1.0000	NA	NA	0
5	28	14.10	0.1867	0.3550	3.8100	-1.2	0.2416	NA	NA	0
6	25	13.61	0.0033	0.0100	9.2300	0.1	0.9440	NA	NA	0
7	25	10.20	0.0588	0.0516	14.4192	-1.4	0.1666	NA	NA	0
8	26	14.86	-0.1703	-0.2383	27.7958	-0.7	0.4930	NA	NA	0
9	30	13.98	0.0133	0.0220	13.2360	-1.0	0.3326	NA	NA	0
10	33	13.25	-0.2000	-0.5200	19.5600	0.5	0.6521	NA	NA	0
11	21	20.49	-0.1143	-0.2715	25.3775	-0.7	0.4852	NA	NA	0
12	24	23.62	-0.1587	-0.3991	21.2836	-1.1	0.2516	NA	NA	0

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



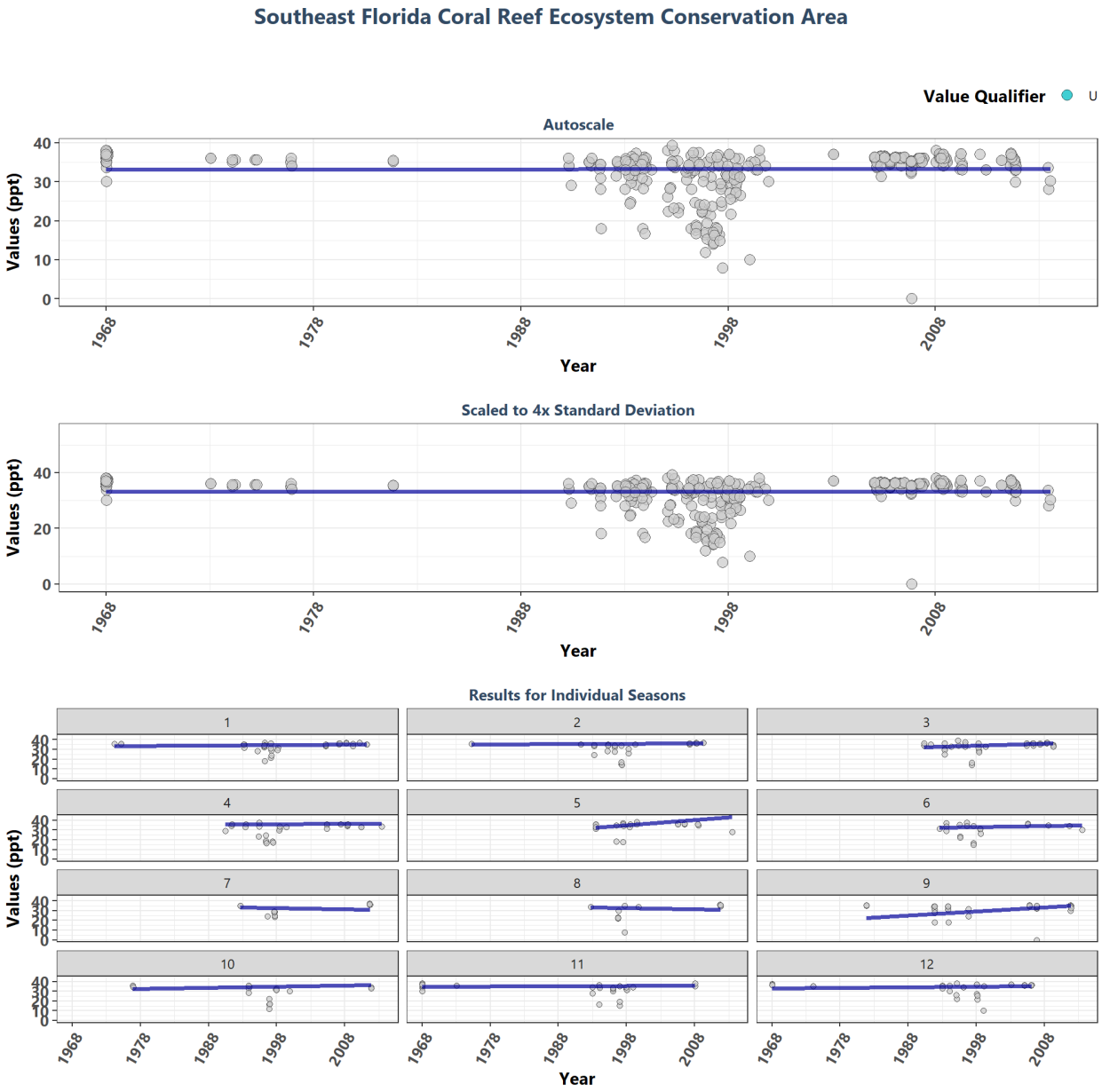
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	3560	34.00	0.0162	0.0055	33.9833	1.5	0.1451	42.5	0	0
1	268	34.16	0.1024	0.0300	34.5800	1.3	0.1913	NA	NA	0
2	303	34.10	0.0535	0.0187	33.2088	0.2	0.8757	NA	NA	0
3	310	34.67	0.0043	0.0000	34.0000	0.2	0.8572	NA	NA	0
4	326	35.11	0.1000	0.0400	33.1100	2.7	0.0070	NA	NA	0
5	254	36.11	0.0511	0.0347	33.5647	2.4	0.0149	NA	NA	0
6	299	35.30	0.0060	0.0000	34.1000	1.3	0.1869	NA	NA	0
7	277	32.00	0.0068	0.0000	34.6700	-2.4	0.0161	NA	NA	0
8	310	29.96	-0.0999	-0.0840	32.9000	-2.3	0.0242	NA	NA	0
9	297	28.70	0.0356	0.0067	33.9667	-2.6	0.0101	NA	NA	0
10	310	30.66	0.1165	0.0870	26.3095	3.1	0.0022	NA	NA	0
11	296	34.00	-0.0856	-0.0829	34.1093	0.1	0.9127	NA	NA	0
12	310	34.30	-0.0968	-0.0927	36.7302	0.9	0.3494	NA	NA	0

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



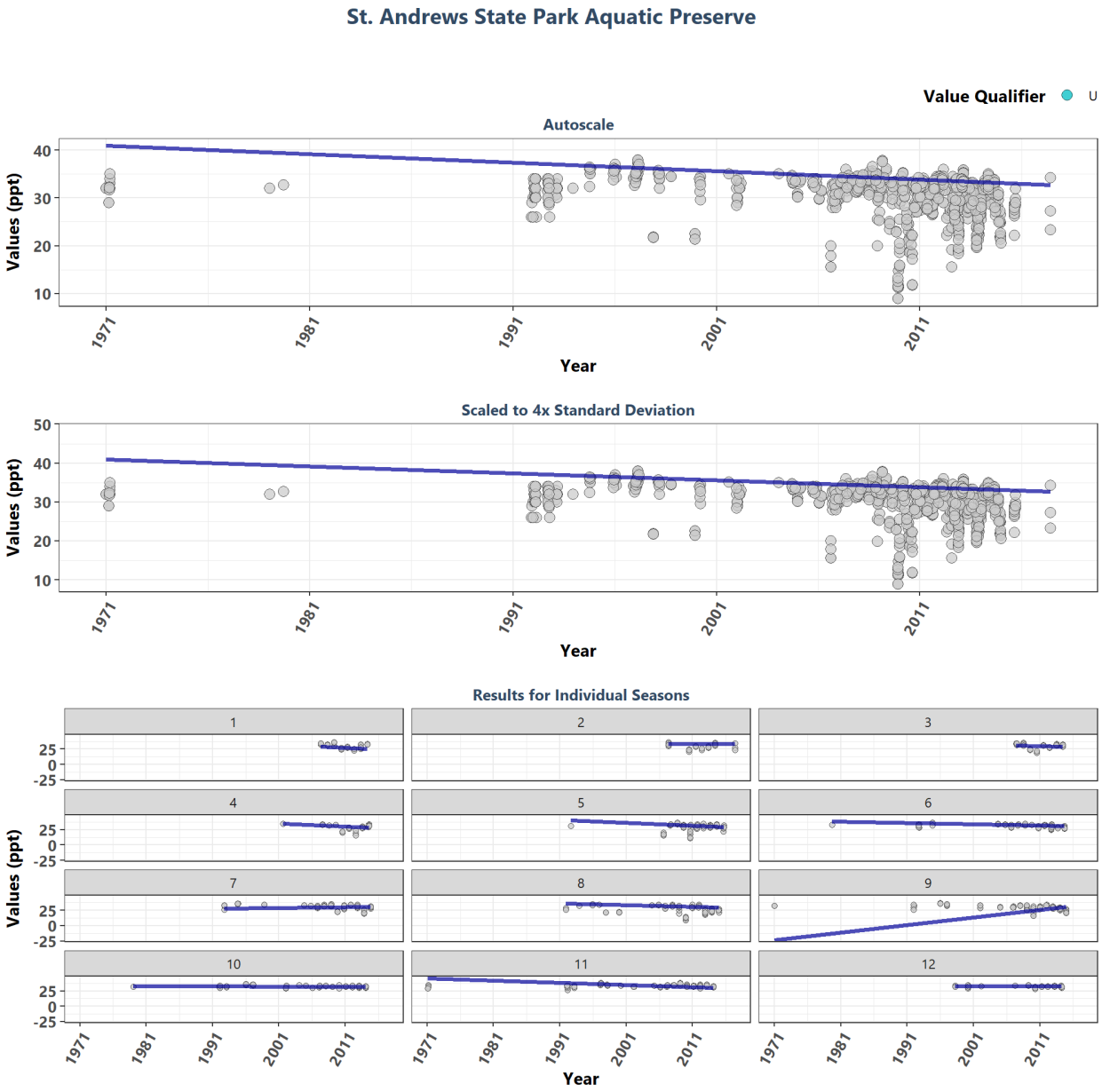
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	10872	32.70	0.0596	0.0464	30.3595	9.4	0.0000	395.7	0	1
1	879	33.40	-0.0577	-0.0691	32.4236	4.8	0.0000	NA	NA	-1
2	984	33.64	0.1047	0.0638	30.4491	4.9	0.0000	NA	NA	1
3	844	34.20	0.0002	0.0000	31.8550	5.9	0.0000	NA	NA	-1
4	951	35.00	0.1353	0.0785	30.2700	11.5	0.0000	NA	NA	1
5	861	36.00	0.2598	0.1778	26.7556	11.4	0.0000	NA	NA	1
6	917	34.00	0.0456	0.0250	31.9500	6.2	0.0000	NA	NA	1
7	833	30.70	0.1084	0.0700	29.8300	-3.1	0.0021	NA	NA	1
8	949	28.90	-0.1957	-0.2542	39.3083	-2.7	0.0077	NA	NA	-1
9	881	26.60	-0.0713	-0.0800	34.7800	-8.7	0.0000	NA	NA	-1
10	973	28.56	0.1362	0.1225	27.8750	-0.4	0.6967	NA	NA	1
11	840	31.86	-0.0083	-0.0083	28.9850	0.0	0.9930	NA	NA	-1
12	960	33.20	0.2497	0.1667	26.6667	2.1	0.0339	NA	NA	1

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



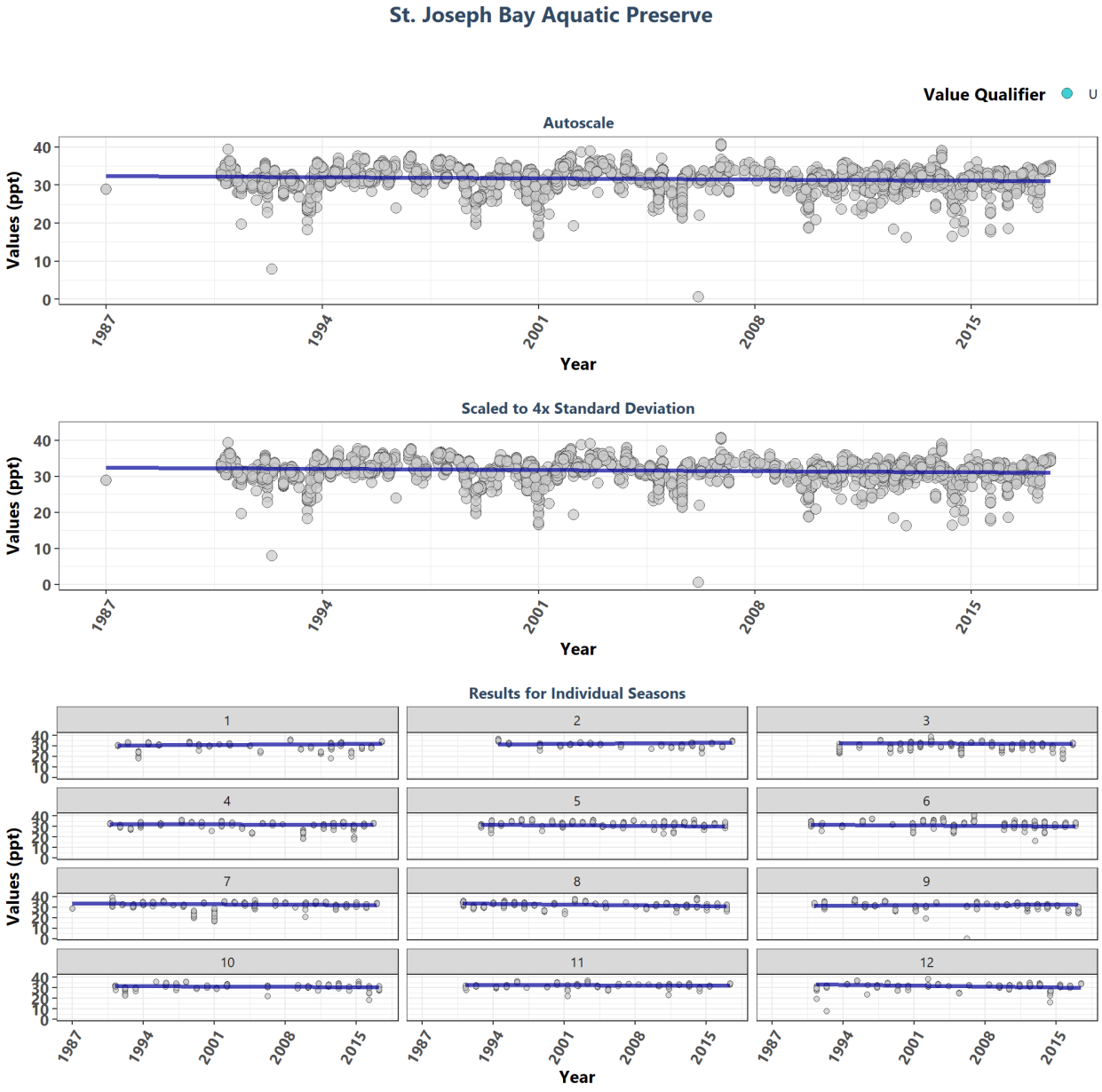
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	481	35.27	0.0723	0.0031	33.1056	0.6	0.5654	63	0	0
1	39	34.88	0.1571	0.0514	32.9979	1.4	0.1673	NA	NA	0
2	32	34.80	0.1314	0.0322	34.7850	2.1	0.0341	NA	NA	0
3	68	36.01	0.1538	0.2007	27.8800	0.9	0.3666	NA	NA	0
4	34	33.20	0.1584	0.0341	34.8758	1.4	0.1630	NA	NA	0
5	46	36.04	0.4476	0.5343	19.0714	1.3	0.1820	NA	NA	0
6	26	34.00	0.1676	0.1200	29.4200	1.1	0.2723	NA	NA	0
7	15	35.10	-0.2810	-0.1238	36.5905	2.5	0.0131	NA	NA	0
8	18	35.16	-0.3717	-0.1286	36.5000	3.3	0.0010	NA	NA	0
9	56	34.95	0.5163	0.4220	16.5970	1.8	0.0770	NA	NA	0
10	21	33.00	0.2621	0.1143	31.3714	-1.8	0.0723	NA	NA	0
11	74	36.50	0.0737	0.0357	34.6300	-5.7	0.0000	NA	NA	0
12	52	36.10	0.1538	0.0556	33.2133	1.7	0.0926	NA	NA	0

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



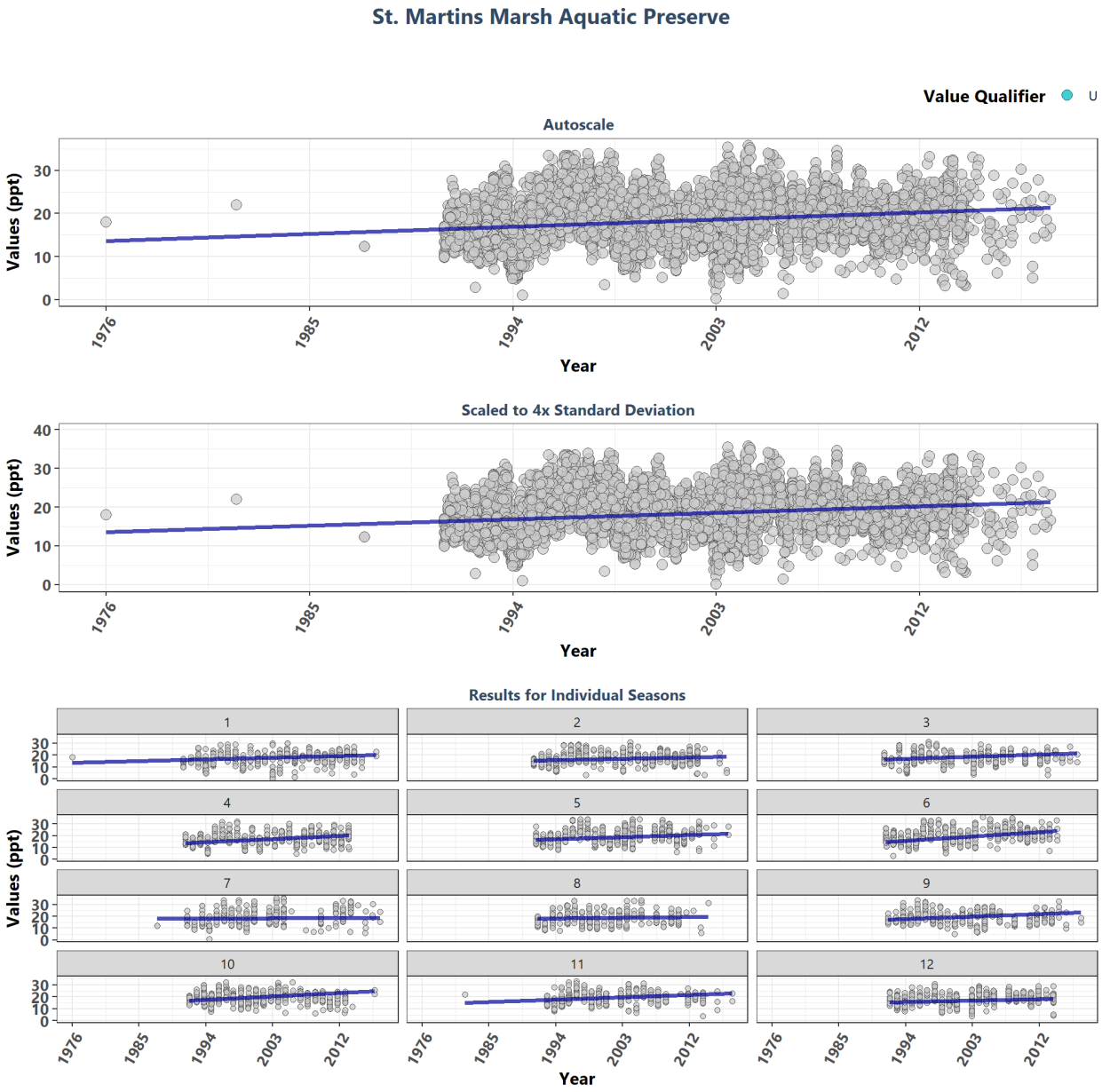
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	902	31.60	-0.2120	-0.1773	40.9014	-9.9	0.0000	84.9	0	-1
1	31	30.00	-0.2555	-0.5500	49.1800	-1.3	0.1896	NA	NA	-1
2	31	30.90	-0.0556	-0.0189	34.1178	0.6	0.5775	NA	NA	-1
3	41	31.50	-0.3911	-0.3231	42.8000	-0.9	0.3797	NA	NA	-1
4	31	29.70	-0.1656	-0.4400	48.0400	1.5	0.1294	NA	NA	-1
5	96	30.11	-0.4686	-0.5000	51.2700	-2.6	0.0088	NA	NA	-1
6	97	30.96	-0.0951	-0.2000	39.7000	-6.1	0.0000	NA	NA	-1
7	99	31.27	0.0710	0.1430	24.7510	-6.9	0.0000	NA	NA	1
8	103	27.73	-0.4186	-0.2786	42.1029	-3.8	0.0001	NA	NA	-1
9	109	30.20	0.1892	1.2300	-23.1900	-6.1	0.0000	NA	NA	1
10	111	32.25	-0.0698	-0.0260	33.2654	-1.1	0.2718	NA	NA	-1
11	118	33.40	-0.1794	-0.3633	45.7333	-0.9	0.3708	NA	NA	-1
12	35	32.50	-0.0790	-0.0364	33.9545	-0.7	0.5075	NA	NA	-1

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



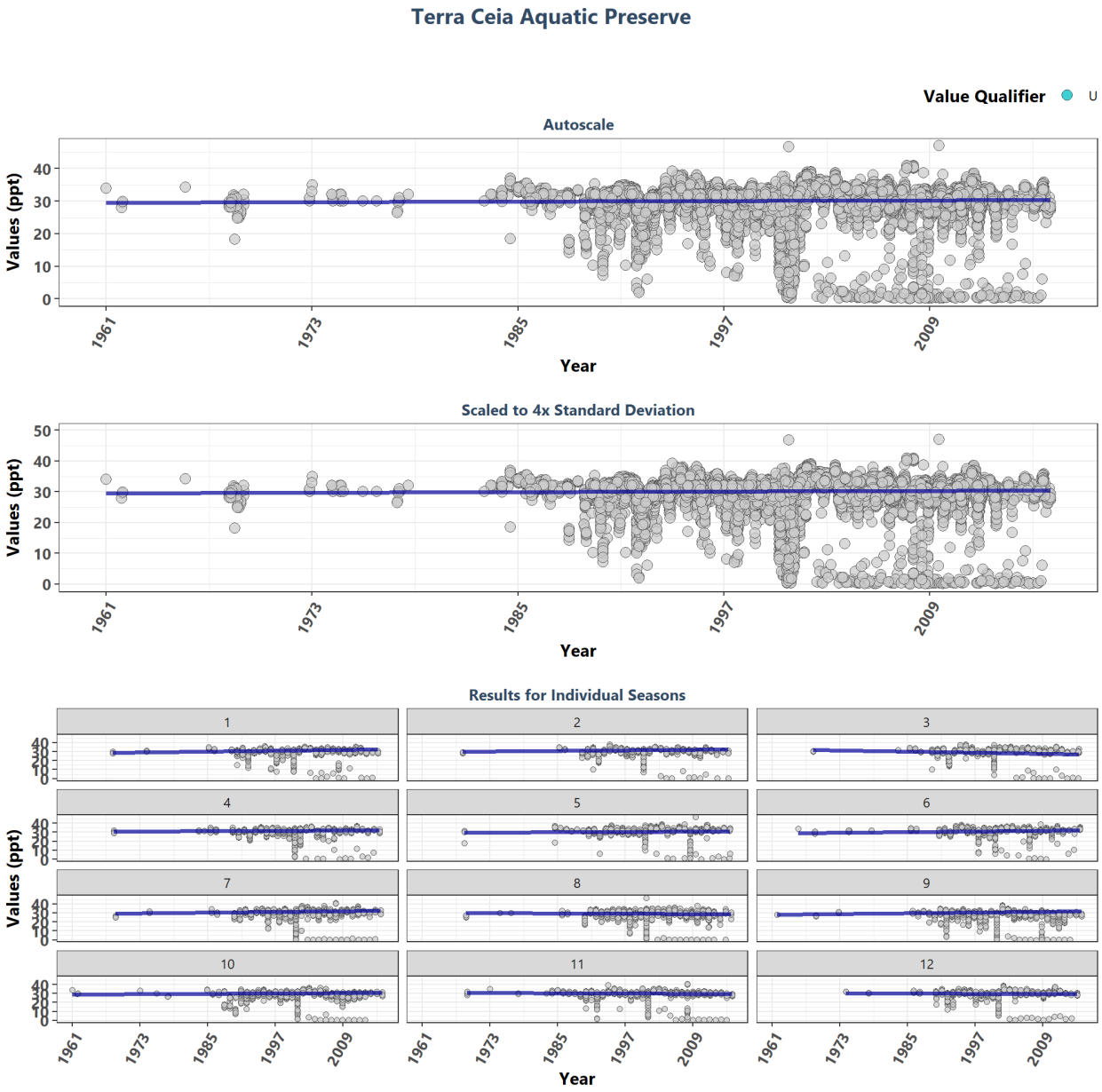
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	3987	32.00	-0.0665	-0.0414	32.3667	-7.6	0.0000	145.3	0	-1
1	256	31.25	0.1838	0.0700	30.1950	-1.2	0.2425	NA	NA	1
2	207	31.20	0.1368	0.0524	31.4143	-3.4	0.0008	NA	NA	1
3	396	30.80	-0.0737	-0.0375	32.9750	-2.1	0.0340	NA	NA	-1
4	304	32.00	-0.0669	-0.0333	32.5333	-1.7	0.0815	NA	NA	-1
5	387	32.90	-0.1561	-0.0667	32.2000	-3.3	0.0009	NA	NA	-1
6	438	32.30	-0.0711	-0.0579	31.8421	-2.3	0.0207	NA	NA	-1
7	488	32.80	-0.1124	-0.0536	33.8118	-8.0	0.0000	NA	NA	-1
8	461	32.70	-0.2423	-0.1000	34.0000	-1.7	0.0902	NA	NA	-1
9	304	31.30	0.1279	0.0389	31.5167	-5.7	0.0000	NA	NA	1
10	260	31.80	-0.0489	-0.0281	31.6995	4.4	0.0000	NA	NA	-1
11	217	32.10	-0.0527	-0.0235	32.9353	2.8	0.0049	NA	NA	-1
12	269	32.20	-0.2168	-0.1105	33.5105	3.4	0.0008	NA	NA	-1

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



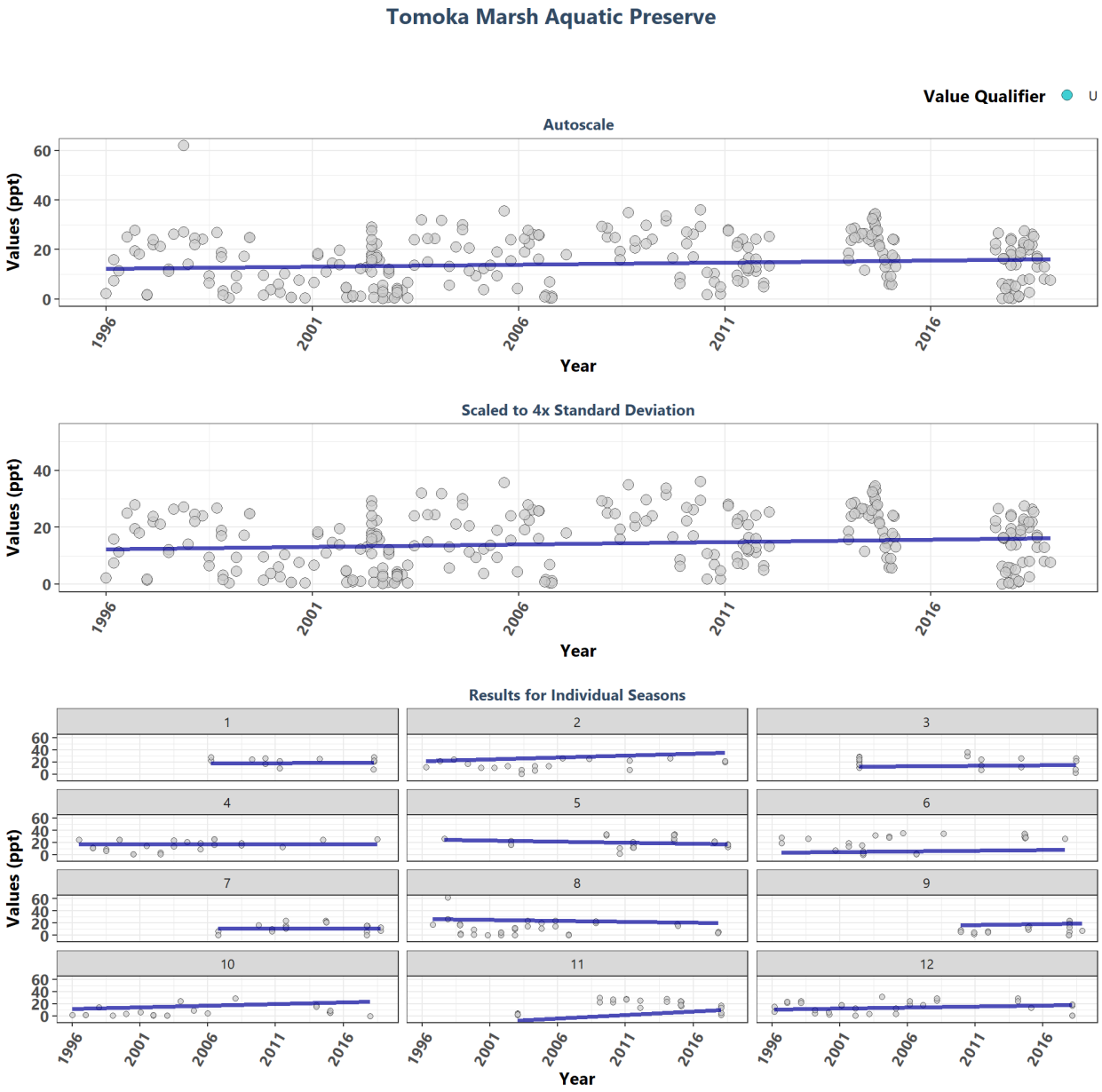
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5799	17.90	0.1561	0.1857	13.5991	17.5	0.0000	78.7	0	1
1	586	16.78	0.1197	0.1562	13.5063	9.3	0.0000	NA	NA	1
2	749	17.10	0.1037	0.1231	13.8000	4.9	0.0000	NA	NA	1
3	433	17.10	0.1851	0.1909	13.6364	3.7	0.0002	NA	NA	1
4	514	17.00	0.2573	0.3053	9.1484	3.5	0.0004	NA	NA	1
5	583	19.40	0.1392	0.2000	13.4000	6.3	0.0000	NA	NA	1
6	546	19.35	0.2575	0.4000	8.5500	8.5	0.0000	NA	NA	1
7	334	17.75	0.0103	0.0111	17.9222	7.0	0.0000	NA	NA	1
8	284	18.00	0.0500	0.0625	17.0625	3.5	0.0005	NA	NA	1
9	370	18.20	0.1908	0.2300	13.6100	0.3	0.7670	NA	NA	1
10	481	18.50	0.2430	0.3167	11.7500	1.6	0.1002	NA	NA	1
11	395	18.90	0.1754	0.2235	13.5882	5.7	0.0000	NA	NA	1
12	524	18.60	0.1184	0.1167	13.8333	6.3	0.0000	NA	NA	1

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



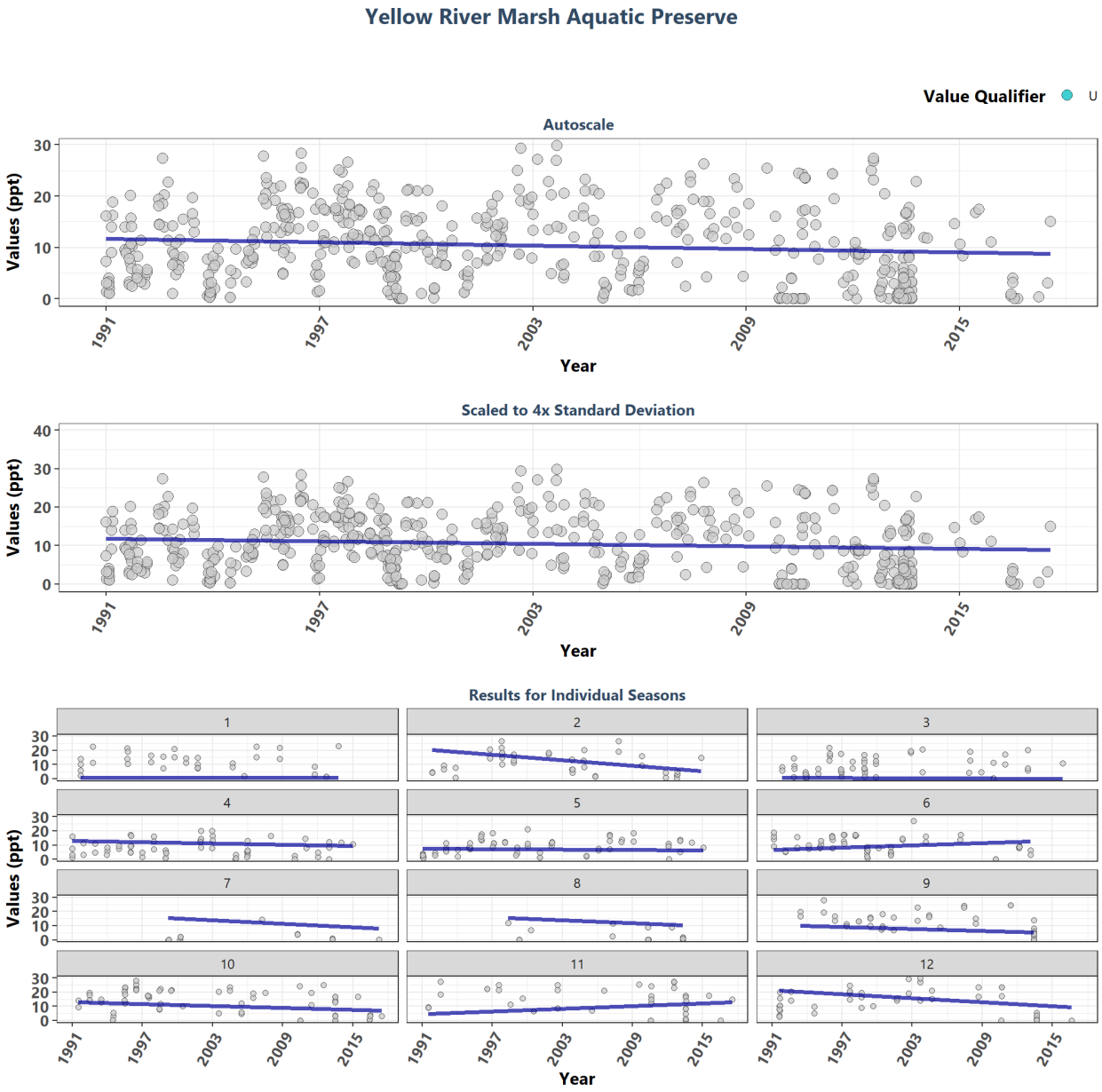
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	17108	30.56	0.0353	0.0162	29.5400	5.6	0.0000	242.2	0	1
1	1276	30.60	0.1365	0.0800	27.9600	6.7	0.0000	NA	NA	1
2	1203	30.50	0.0977	0.0550	29.8000	3.4	0.0007	NA	NA	1
3	1278	31.20	-0.1309	-0.1000	32.6000	7.3	0.0000	NA	NA	-1
4	1242	31.30	0.0544	0.0250	31.0000	6.5	0.0000	NA	NA	1
5	1484	32.00	0.0525	0.0280	29.2800	5.6	0.0000	NA	NA	1
6	1379	32.00	0.0814	0.0500	29.1000	3.0	0.0024	NA	NA	1
7	1476	31.20	0.1235	0.0667	28.7000	4.7	0.0000	NA	NA	1
8	1787	29.90	-0.0485	-0.0257	30.0771	-0.8	0.3973	NA	NA	-1
9	1541	28.50	0.1241	0.0688	27.8500	-7.7	0.0000	NA	NA	1
10	1697	29.10	0.0655	0.0389	28.9421	-3.0	0.0027	NA	NA	1
11	1427	30.00	-0.0444	-0.0182	30.7273	-2.5	0.0119	NA	NA	-1
12	1318	30.40	-0.0133	-0.0077	30.2231	2.9	0.0043	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	274	15.99	0.0776	0.1687	12.2788	1.8	0.0724	8.4	0.681	0
1	13	22.30	0.0414	0.0622	16.9533	-0.9	0.3750	NA	NA	0
2	19	13.62	0.2154	0.5900	21.5650	1.1	0.2617	NA	NA	0
3	20	19.68	0.0789	0.1450	11.7000	-0.8	0.4086	NA	NA	0
4	25	15.83	-0.0233	-0.0125	17.7200	1.5	0.1397	NA	NA	0
5	25	17.52	-0.1316	-0.3513	24.9488	-0.1	0.8869	NA	NA	0
6	26	26.58	0.1255	0.2256	3.2933	1.5	0.1226	NA	NA	0
7	20	14.46	0.0000	-0.0031	11.2019	0.5	0.6419	NA	NA	0
8	32	11.18	-0.1923	-0.2814	26.8029	0.0	1.0000	NA	NA	0
9	18	8.06	0.1014	0.3050	12.8575	1.4	0.1499	NA	NA	0
10	22	4.98	0.2133	0.5578	11.3678	0.8	0.4275	NA	NA	0
11	24	17.59	0.2418	1.1467	-15.4417	0.7	0.4962	NA	NA	0
12	30	17.42	0.1930	0.3112	10.8194	0.3	0.7609	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	561	8.90	-0.0852	-0.1091	11.7308	-2.5	0.0139	46.6	0	-1
1	34	11.55	0.0468	0.0086	0.5371	-0.8	0.4129	NA	NA	1
2	39	10.00	-0.3901	-0.6333	20.6333	-1.3	0.2106	NA	NA	-1
3	58	7.00	-0.0905	-0.0253	0.6816	2.3	0.0219	NA	NA	-1
4	60	7.10	-0.0998	-0.1538	13.0885	-0.4	0.7011	NA	NA	-1
5	66	8.80	-0.0345	-0.0487	7.6355	3.1	0.0018	NA	NA	-1
6	55	8.80	0.2625	0.2744	6.6050	-1.7	0.0869	NA	NA	1
7	19	0.70	-0.2600	-0.4317	19.1809	0.3	0.7979	NA	NA	-2
8	21	0.20	-0.1879	-0.3389	18.0665	-0.6	0.5640	NA	NA	-2
9	44	12.40	-0.1589	-0.2333	10.6667	-3.8	0.0001	NA	NA	-1
10	71	14.00	-0.1404	-0.2150	12.7950	-3.2	0.0013	NA	NA	-1
11	45	12.60	0.2063	0.3111	4.6667	-2.5	0.0116	NA	NA	1
12	49	14.00	-0.2576	-0.4636	21.4091	-1.9	0.0549	NA	NA	-1

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

Appendix V: Managed Area Summary Box Plots

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

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

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

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

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

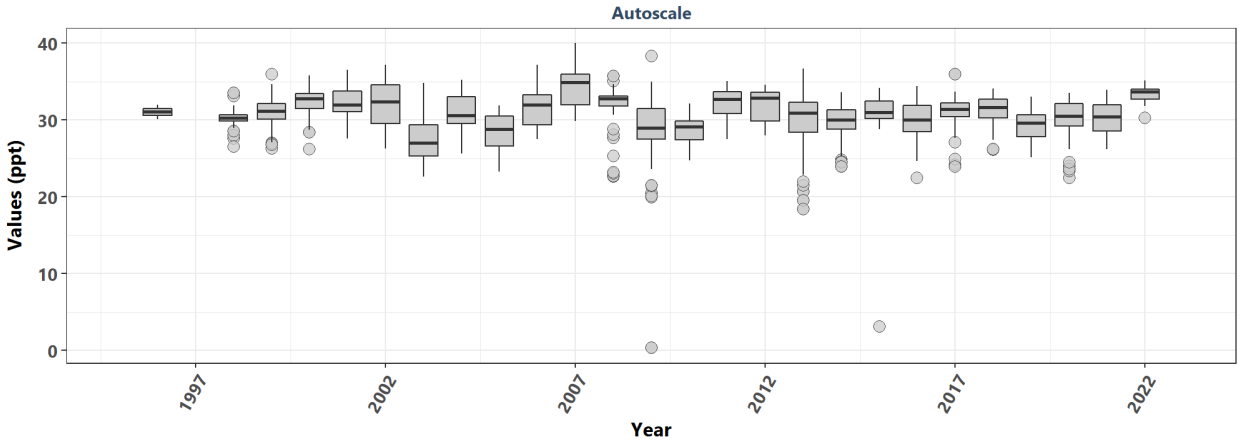
```
x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
y_scale <- mn_RV + 4 * sd_RV

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

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

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

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

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

## Year & Month Plots
p4 <- ggplot(data=plot_data,
             aes(x=YearMonthDec, y=ResultValue,
```

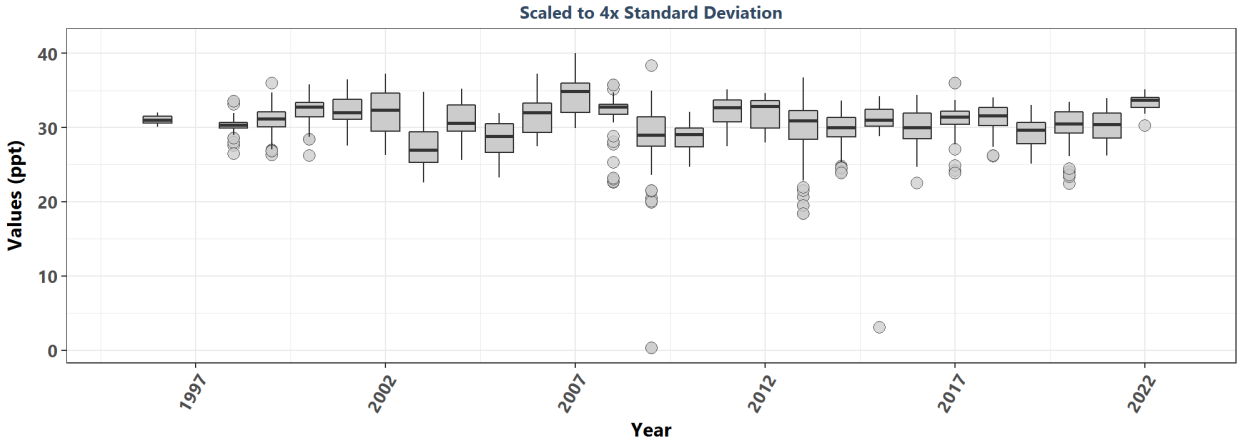
```
                  group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")"), color="Month") +
   scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
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

p5 <- ggplot(data=plot_data,
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")"), color="Month") +
   ylim(min_RV, y_scale) +
   scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
                      breaks=rev(seq(year_upper,
                                     year_lower, -x_scale))) +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal") +
   guides(color=guide_legend(nrow=1))

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

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

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

## Month Plots
p7 <- ggplot(data=plot_data,
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Autoscale",
```

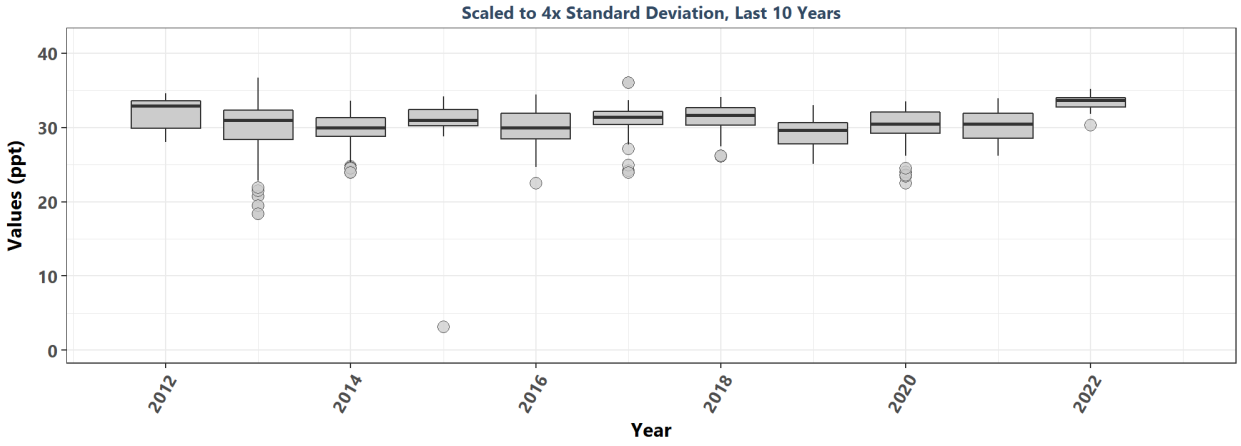
```
            x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
        scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
        plot_theme +
        theme(legend.position="none")

    p8 <- ggplot(data=plot_data,
                 aes(x=Month, y=ResultValue,
                     group=Month, fill=as.factor(Month))) +
        geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                     outlier.color="#333333", outlier.alpha=0.75) +
        labs(subtitle="Scaled to 4x Standard Deviation",
             x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
        ylim(min_RV, y_scale) +
        scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
        plot_theme +
        theme(legend.position="top", legend.box="horizontal") +
        guides(fill=guide_legend(nrow=1))

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

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

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

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

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

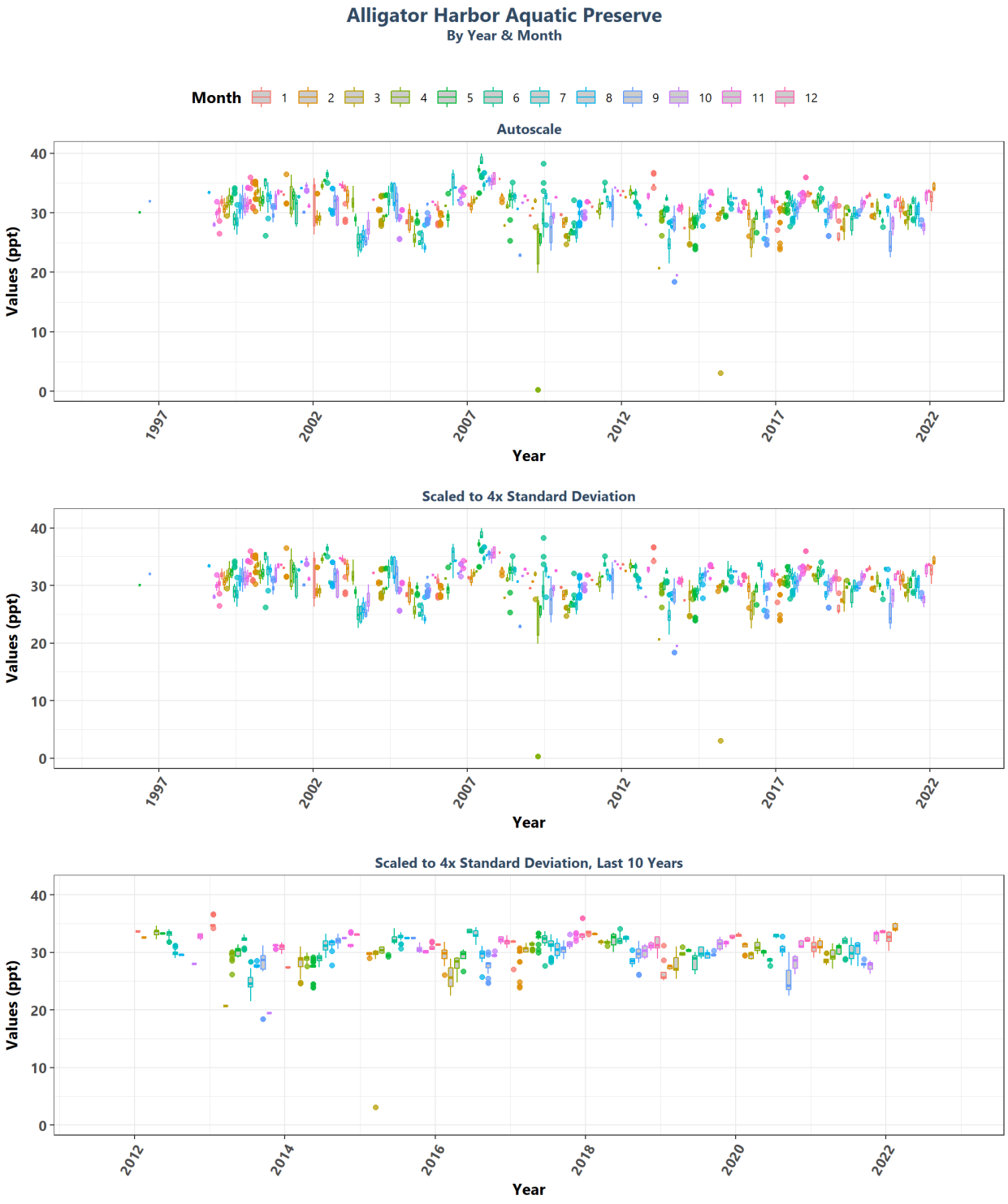
Alligator Harbor Aquatic Preserve

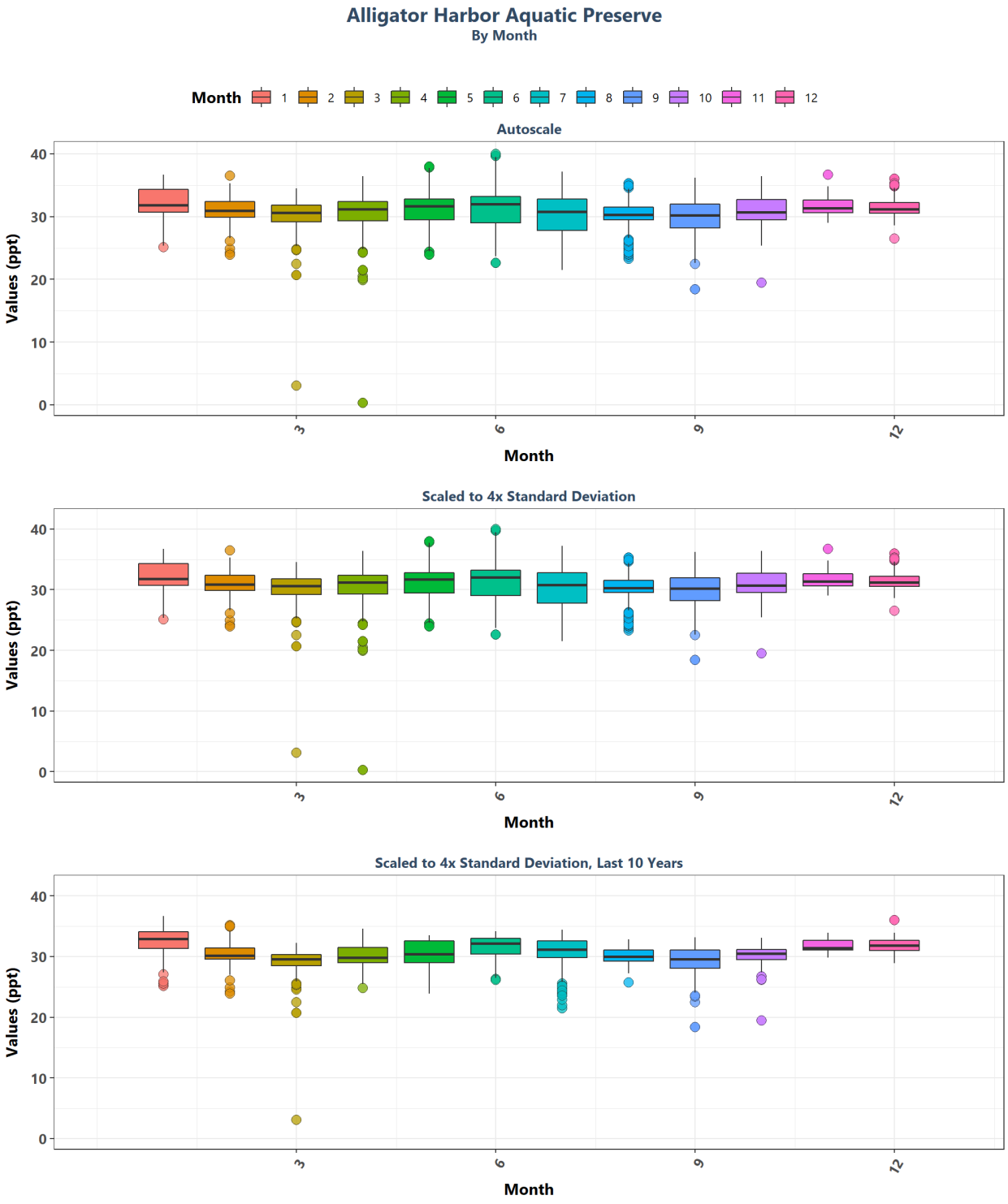
By Year





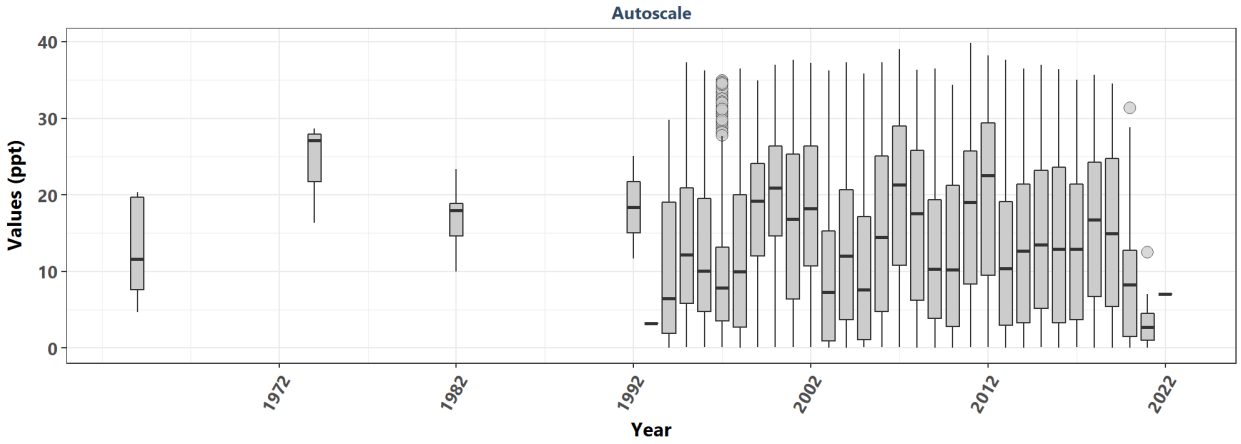


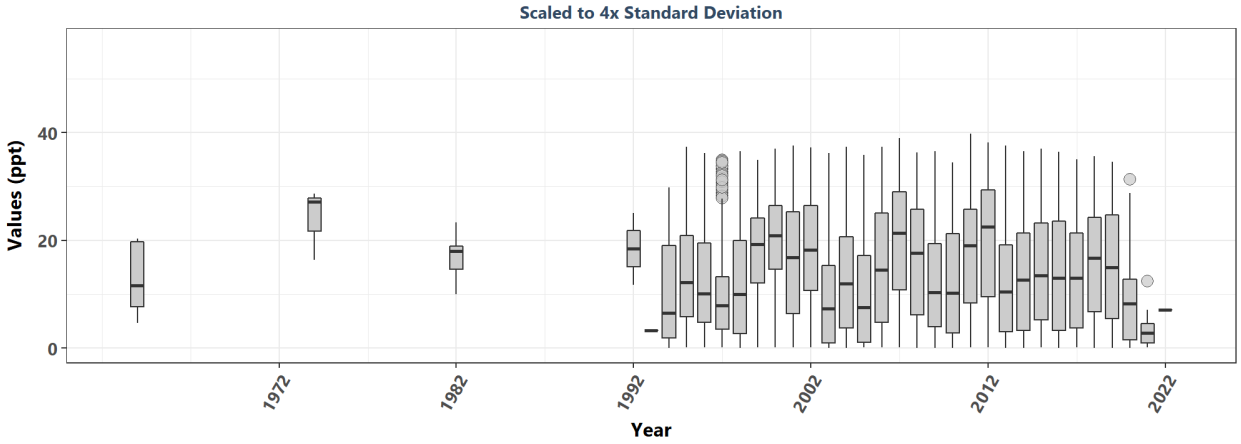

Alligator Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

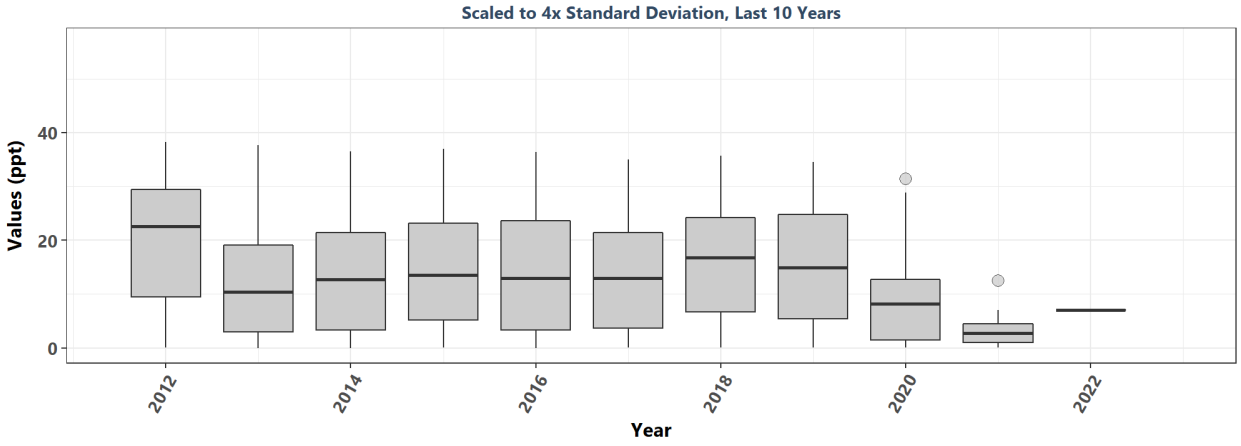

Alligator Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

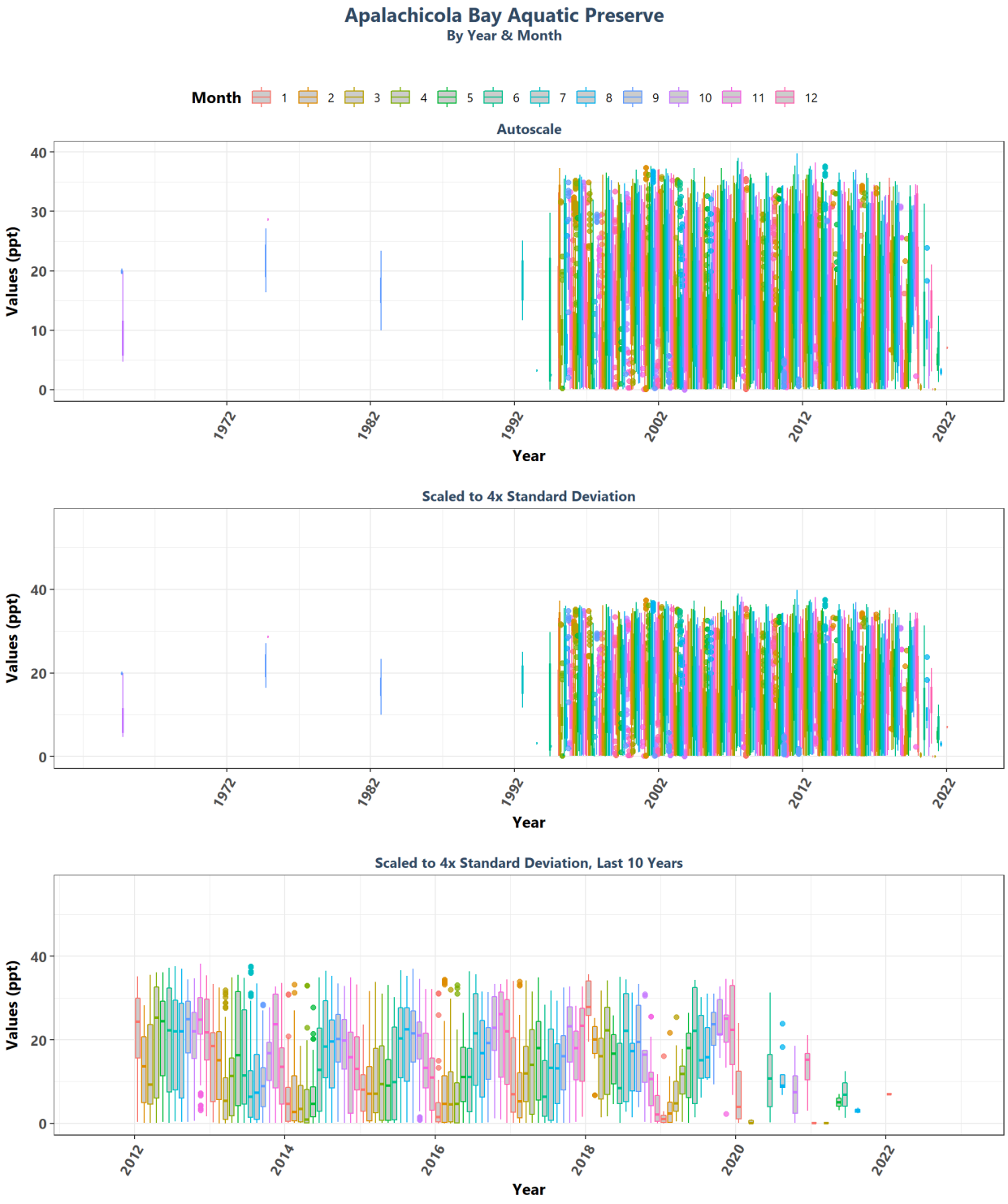
Apalachicola Bay Aquatic Preserve

By Year

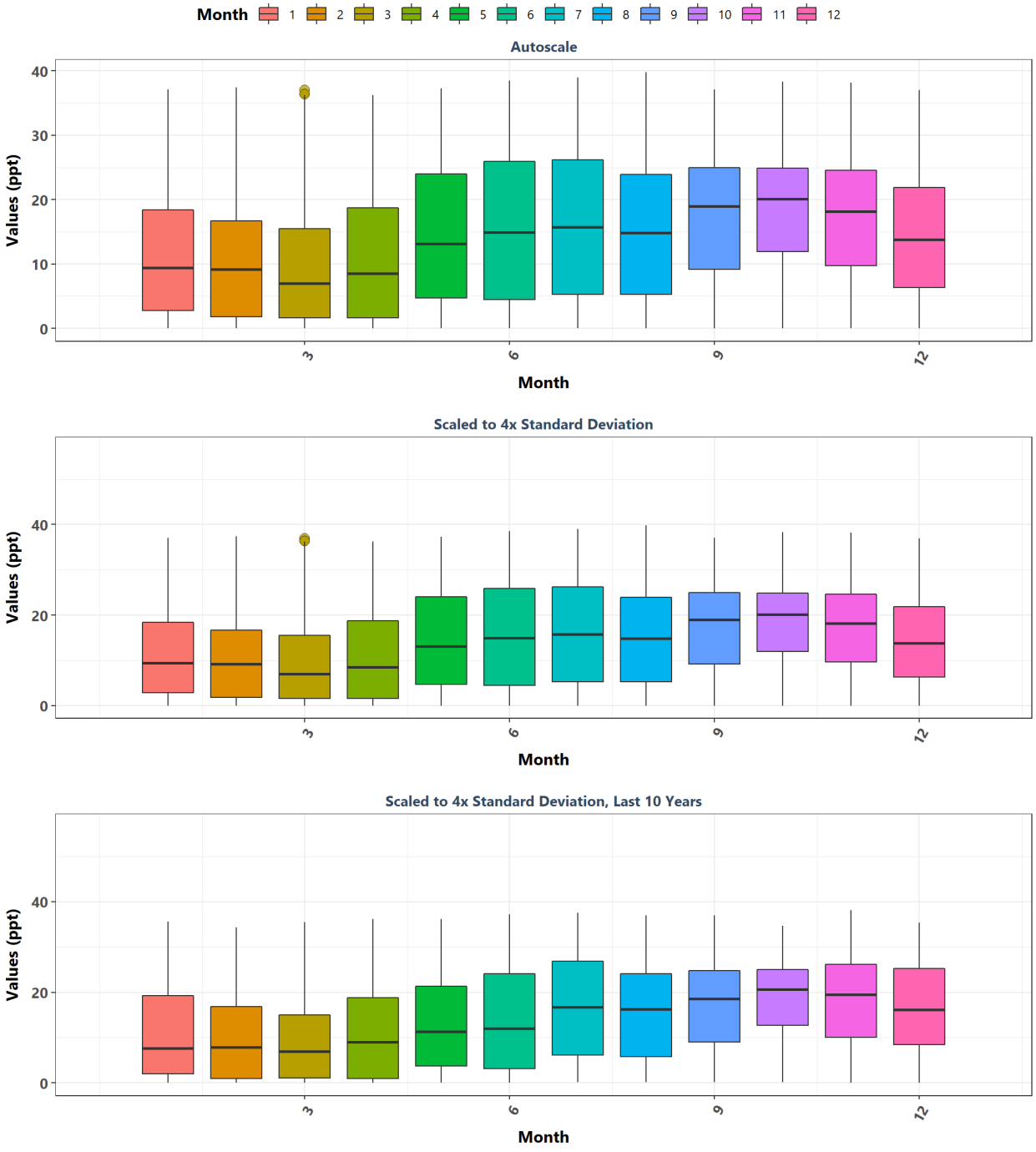






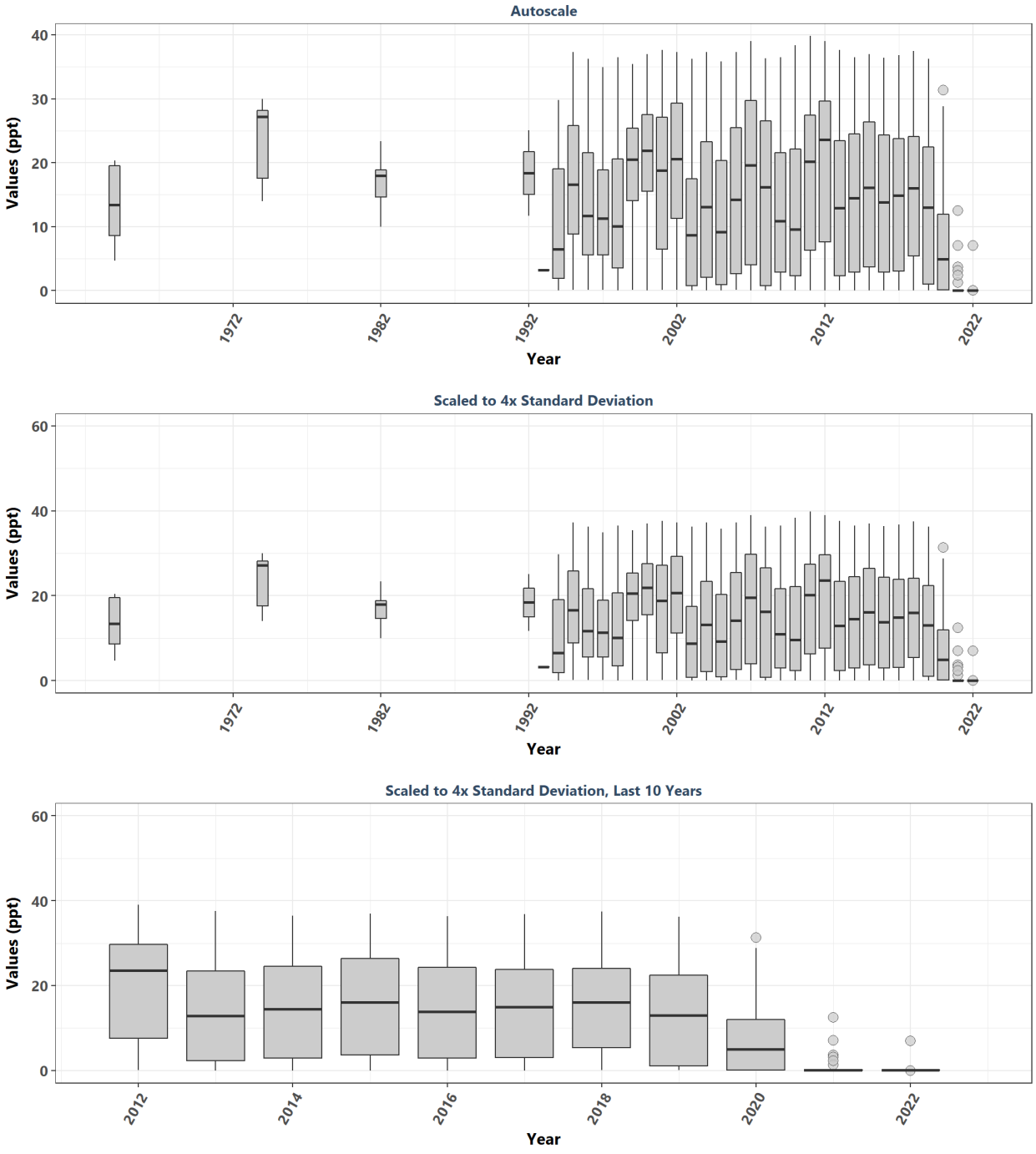

Apalachicola Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

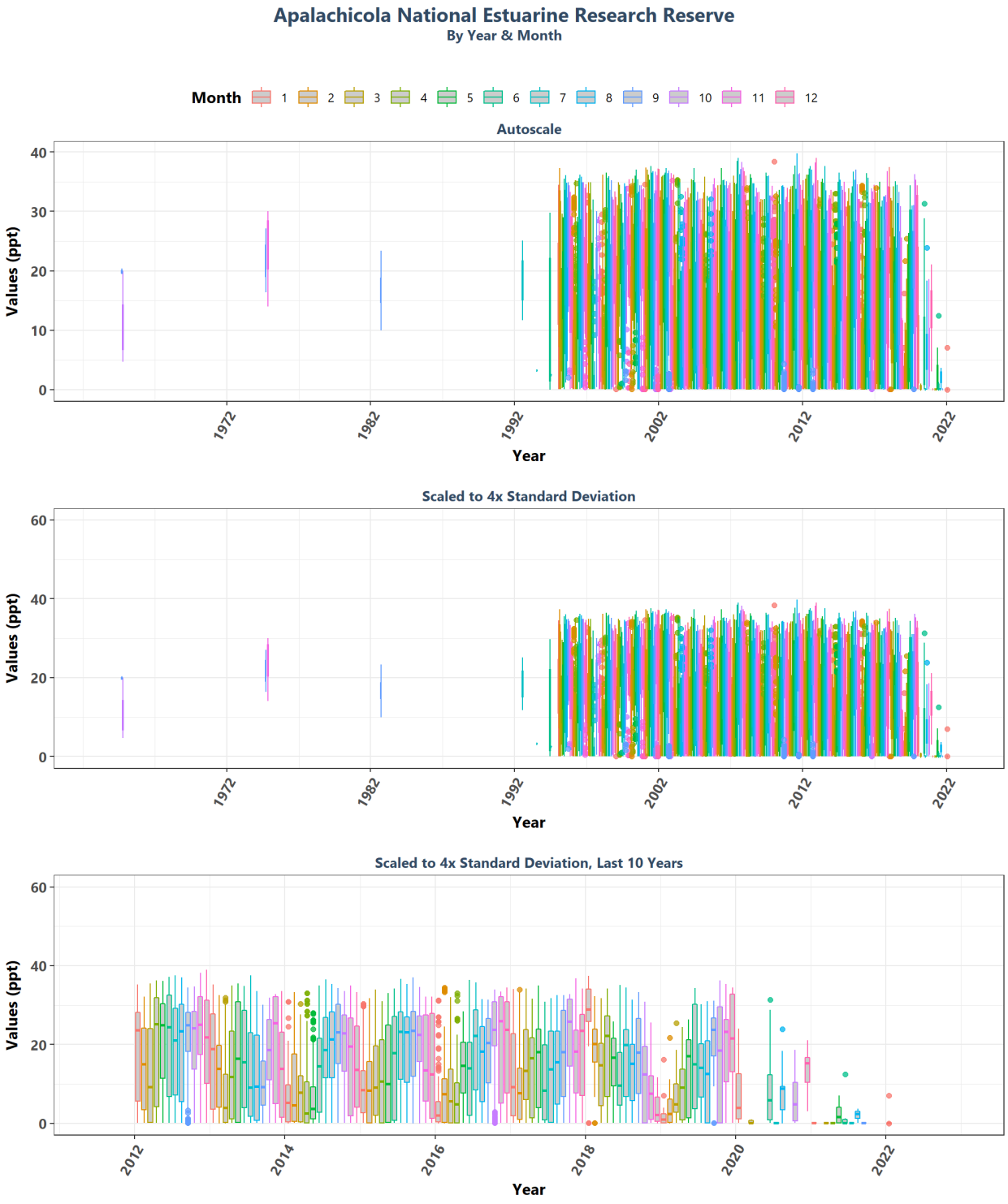
Apalachicola Bay Aquatic Preserve
By Month

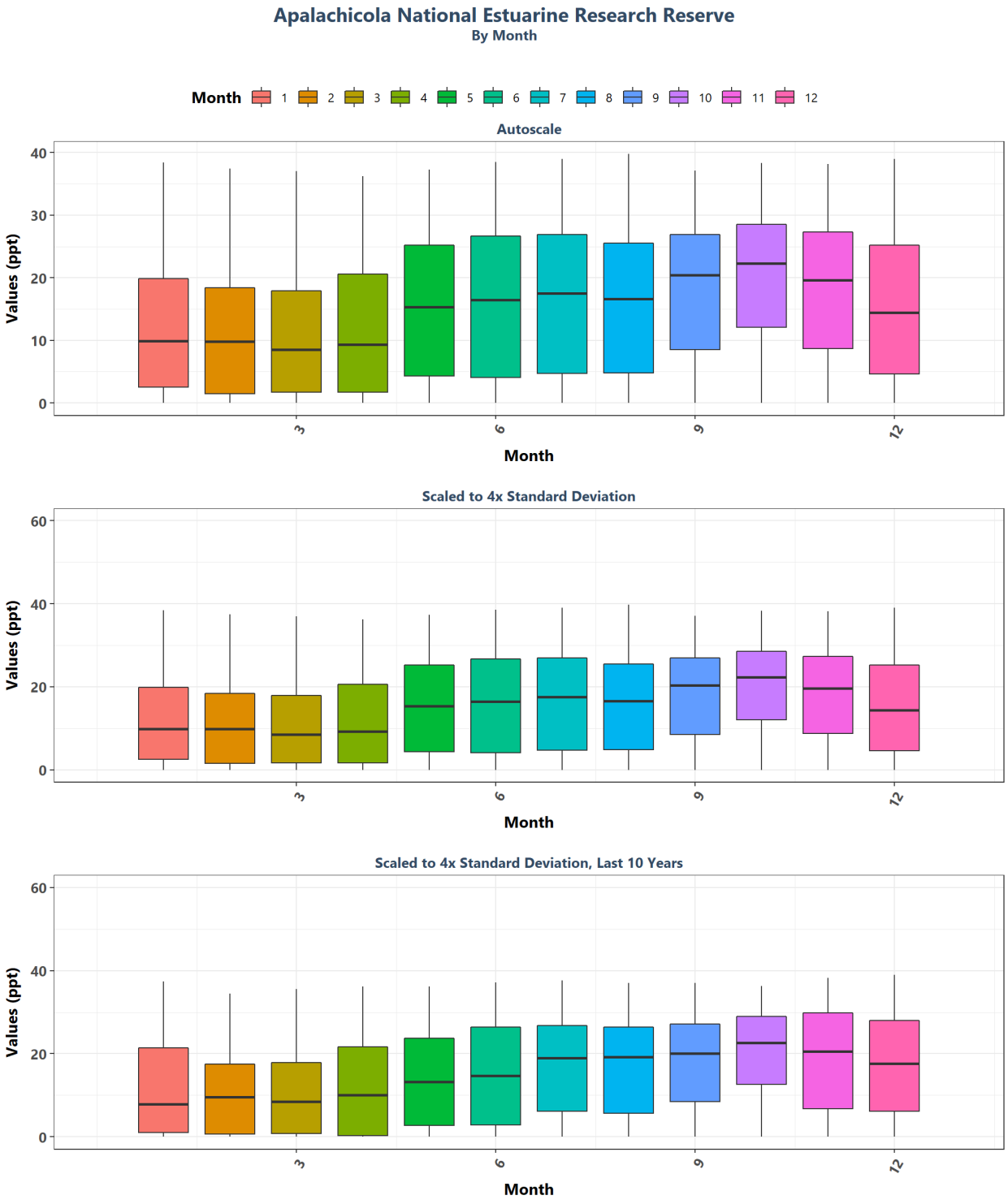


Apalachicola National Estuarine Research Reserve

By Year

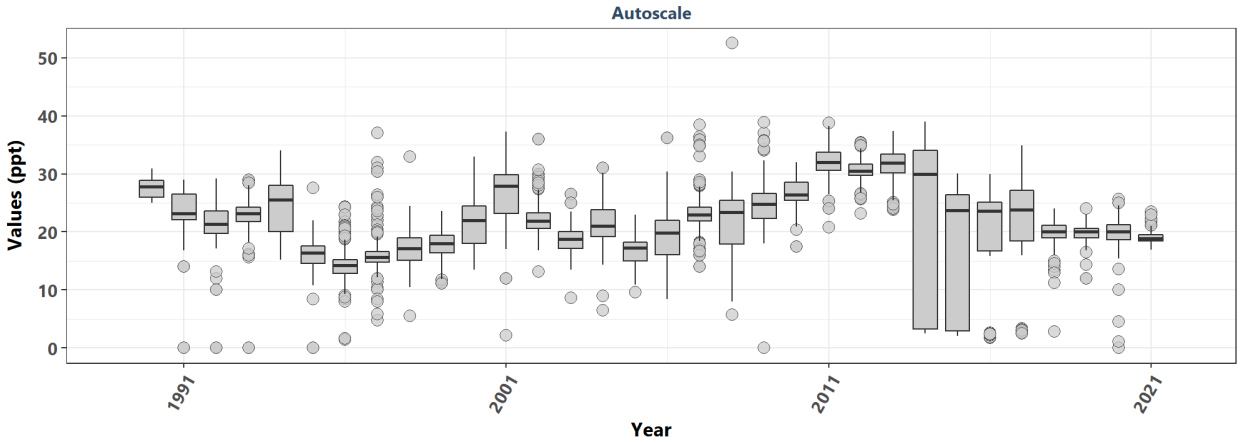


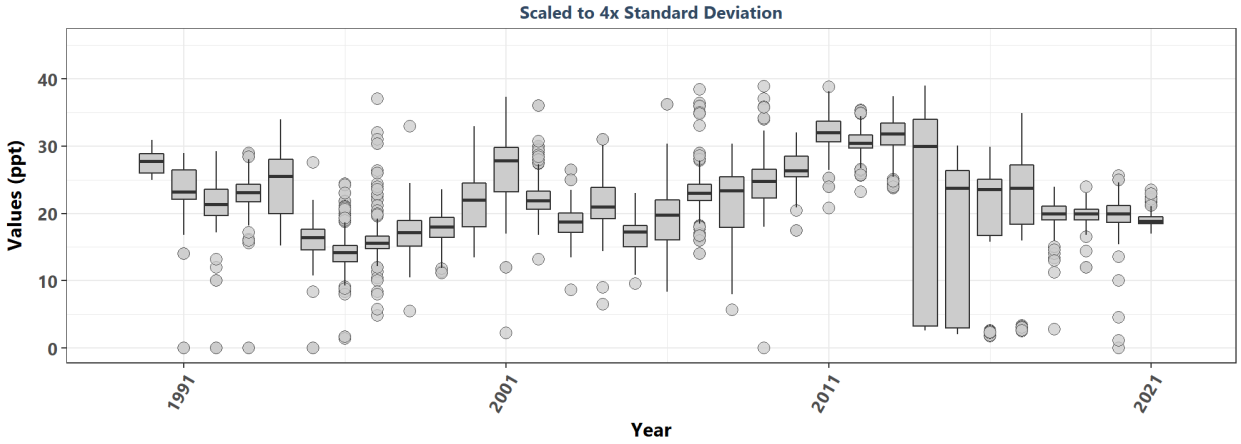

Apalachicola National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

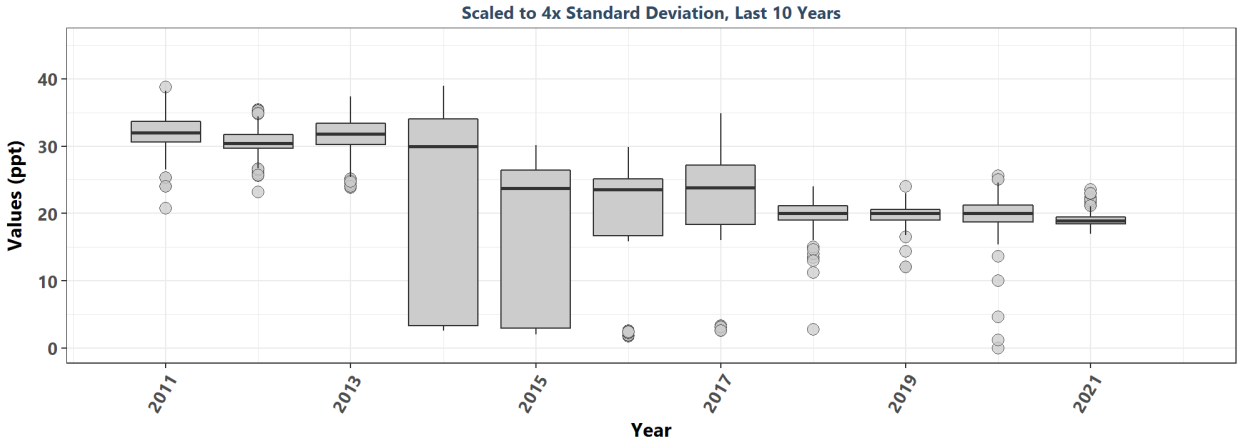

Apalachicola National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

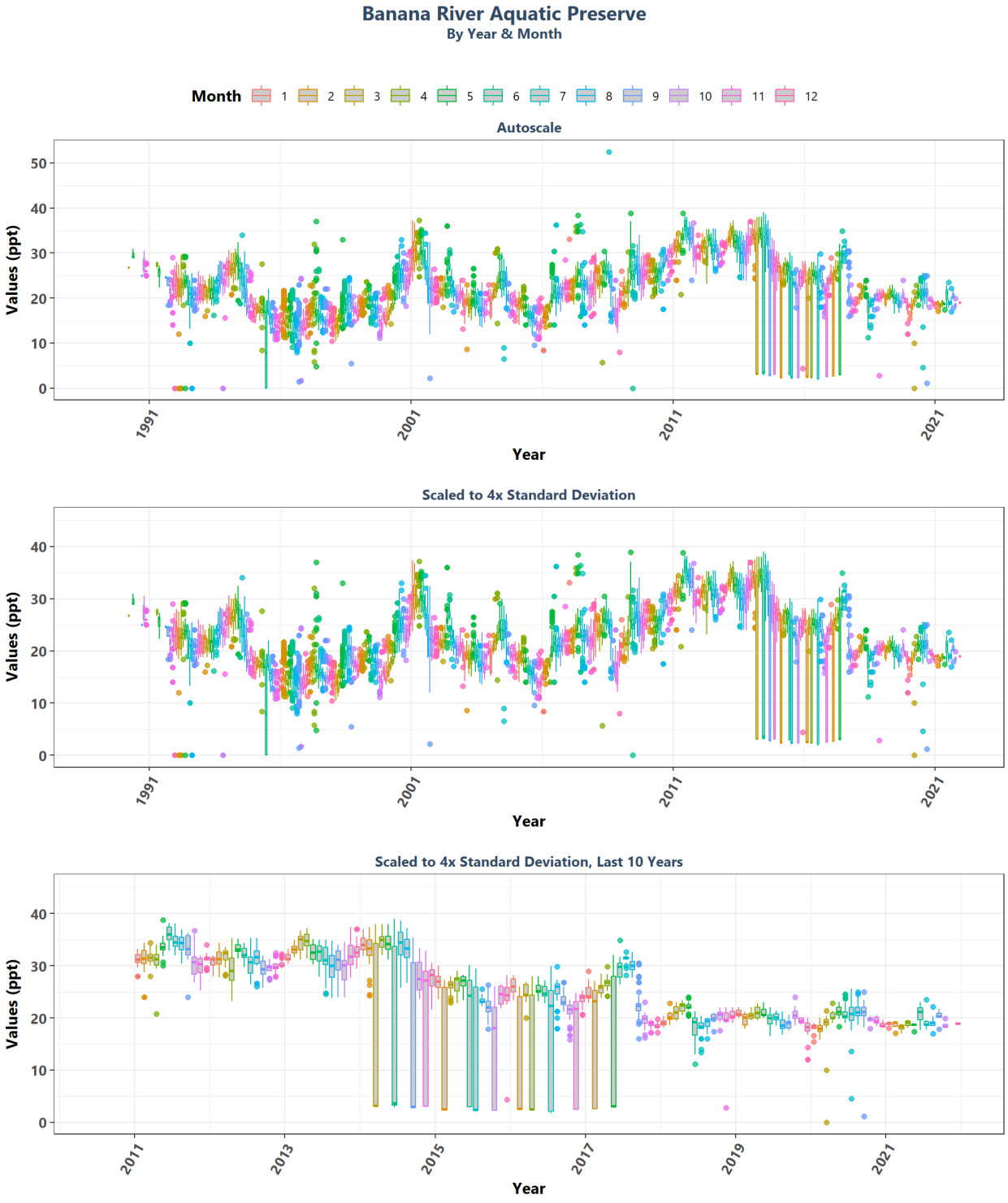
Banana River Aquatic Preserve

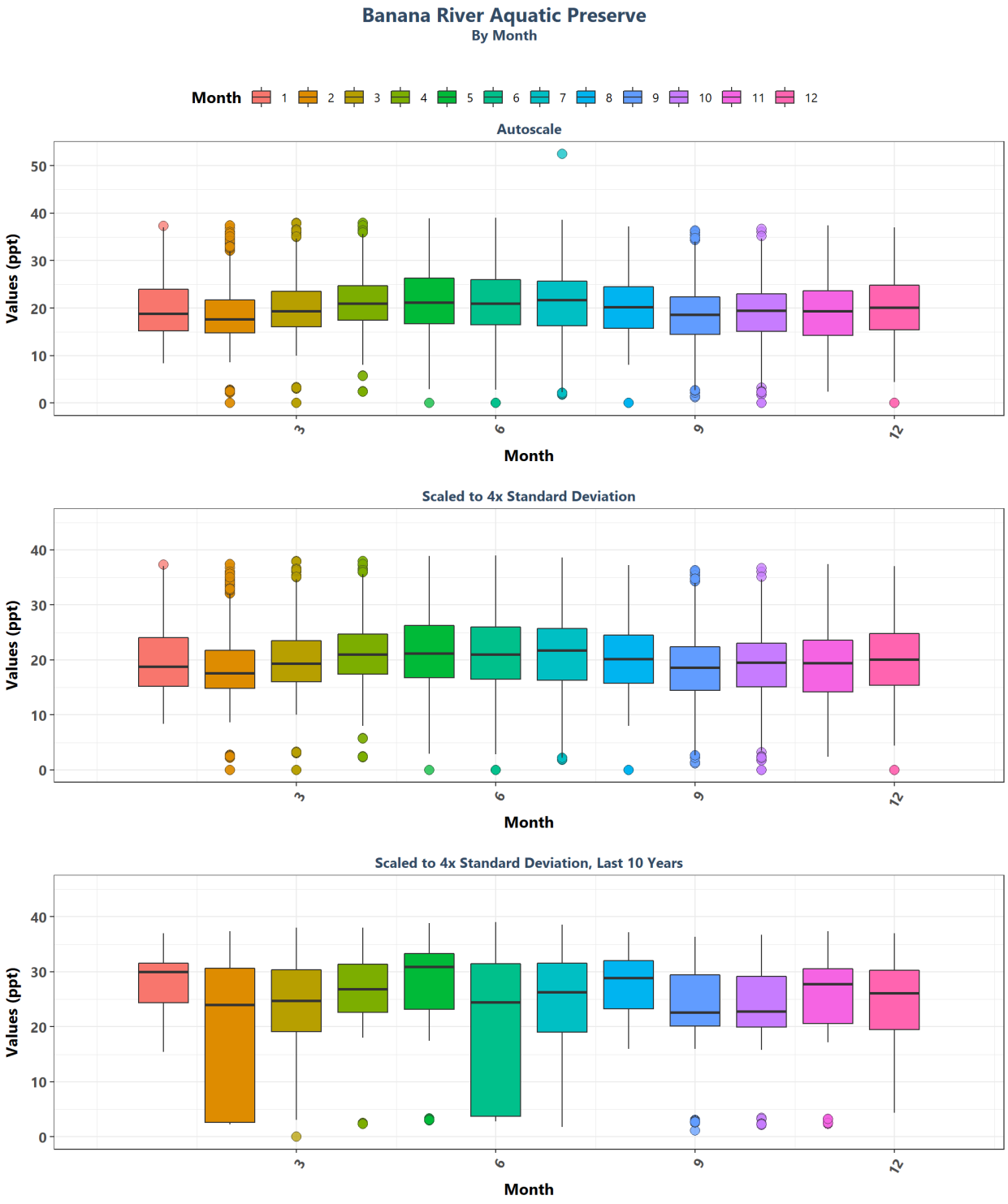
By Year





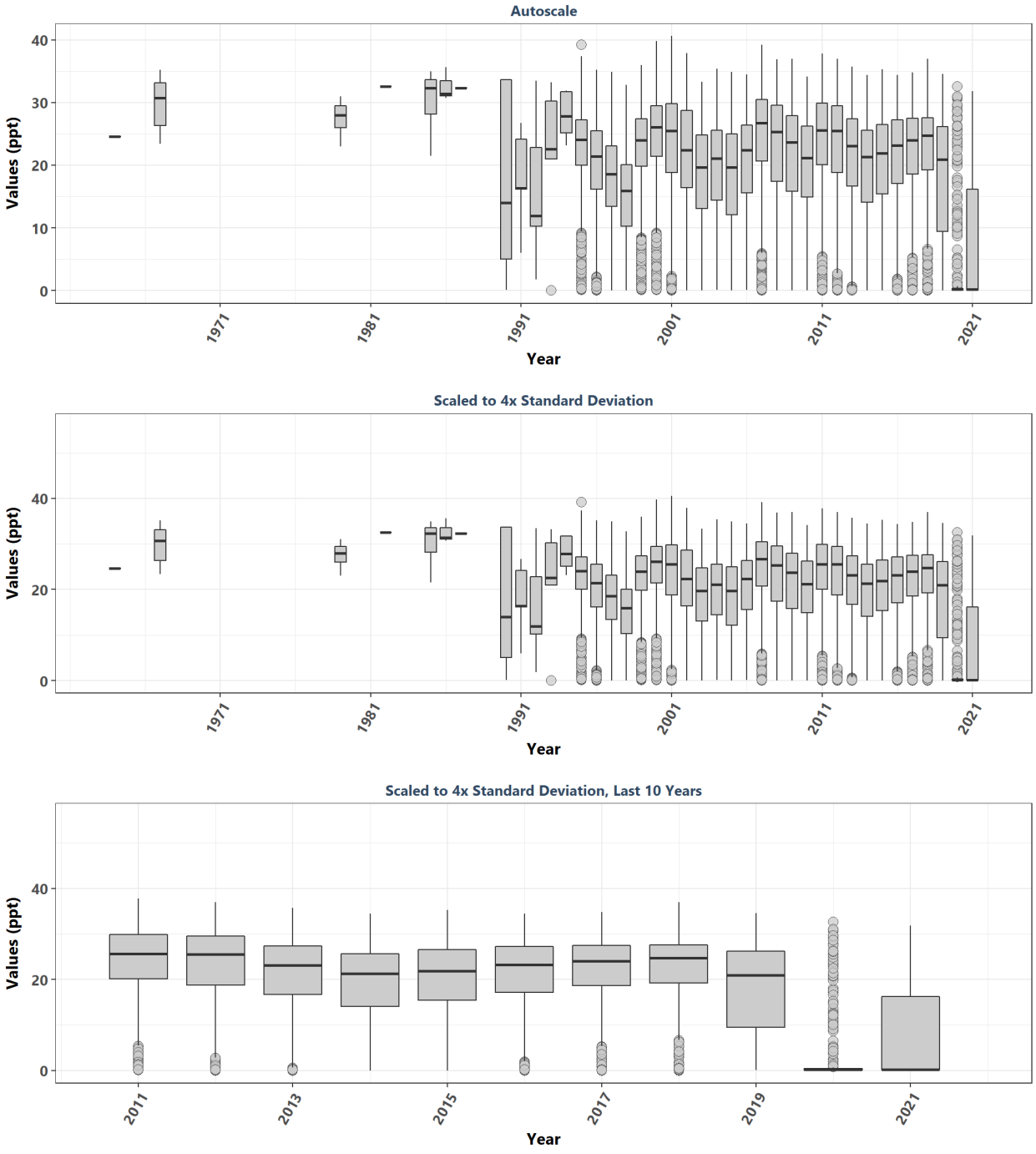


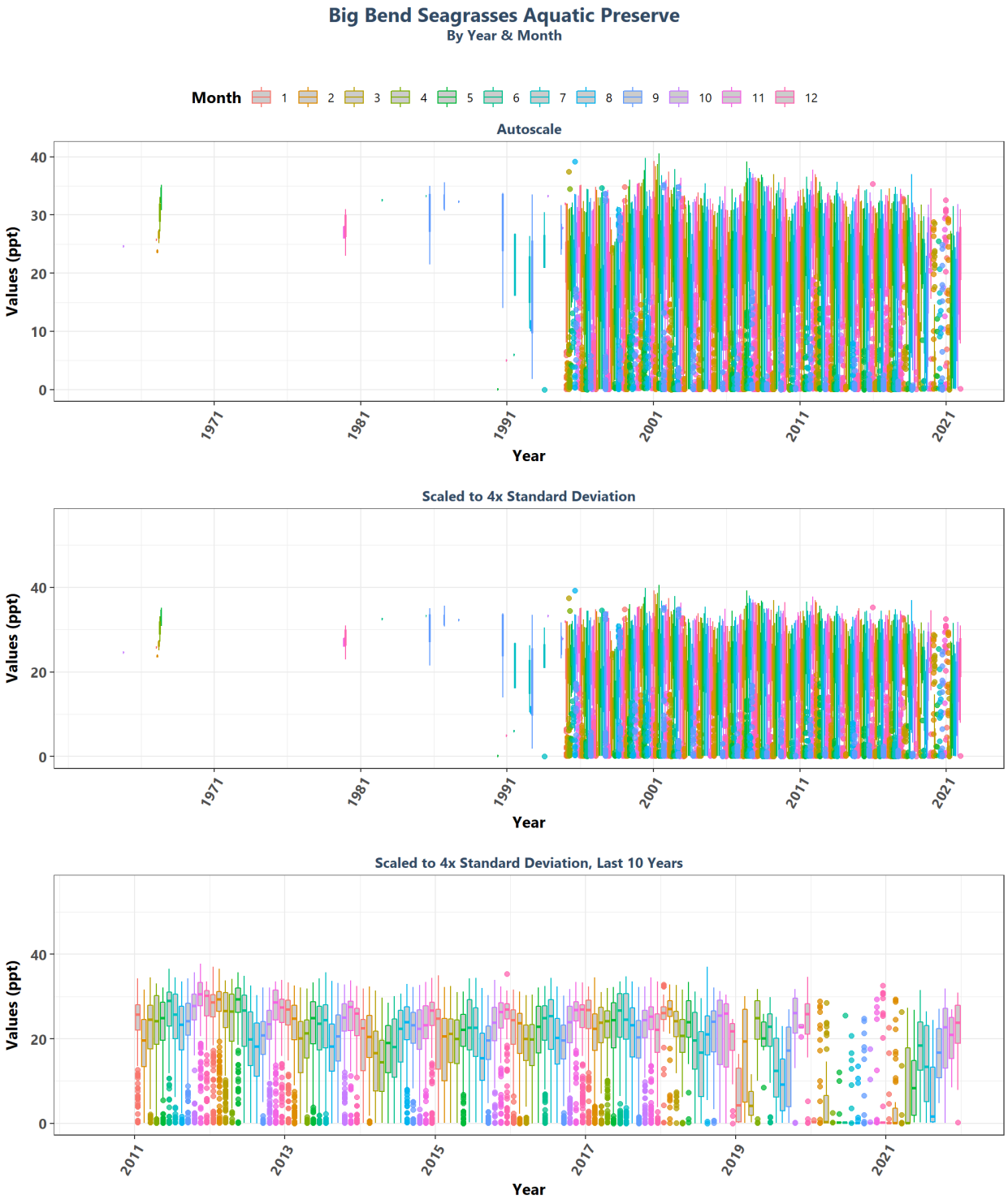

Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

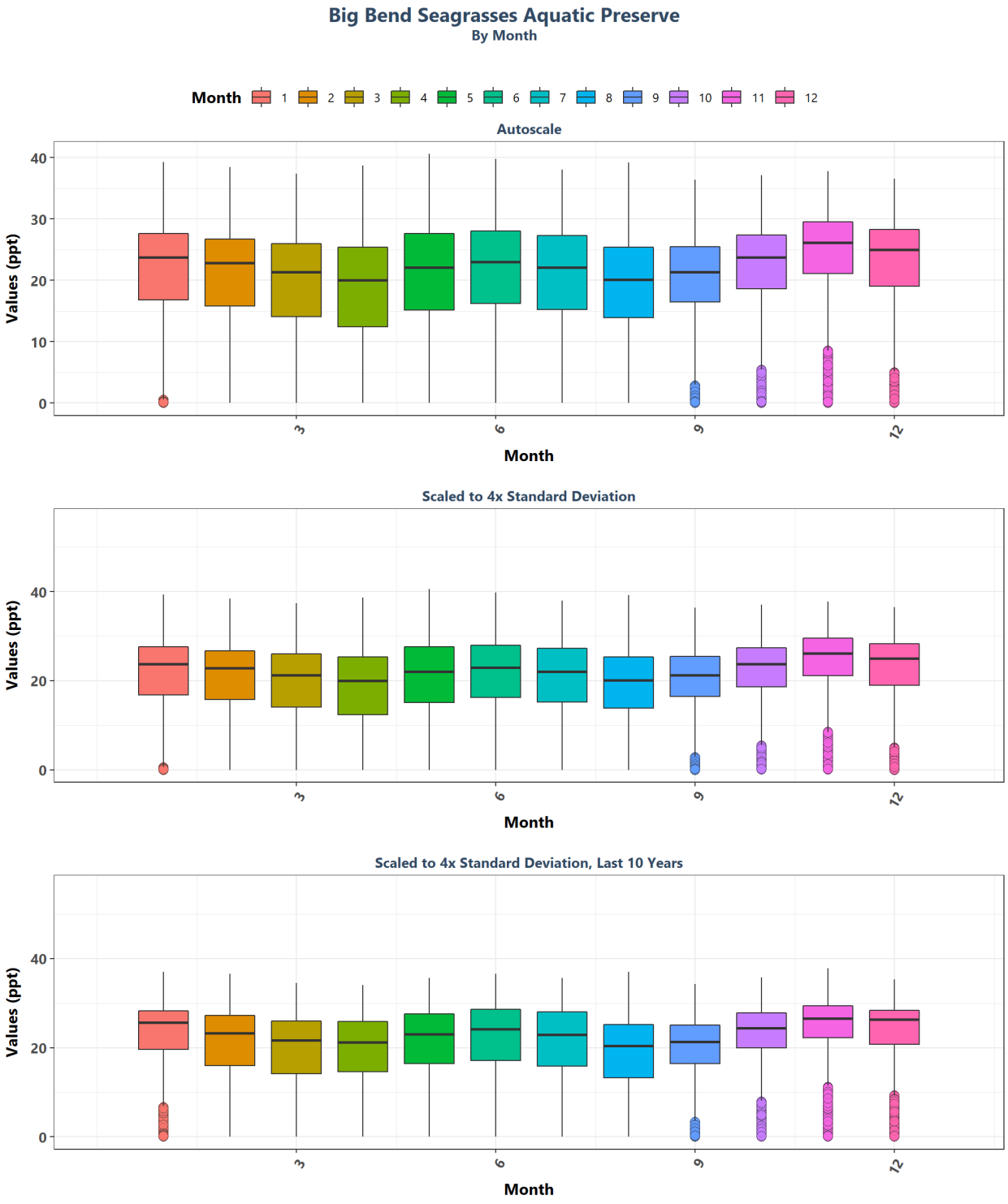

Banana River Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

Big Bend Seagrasses Aquatic Preserve

By Year

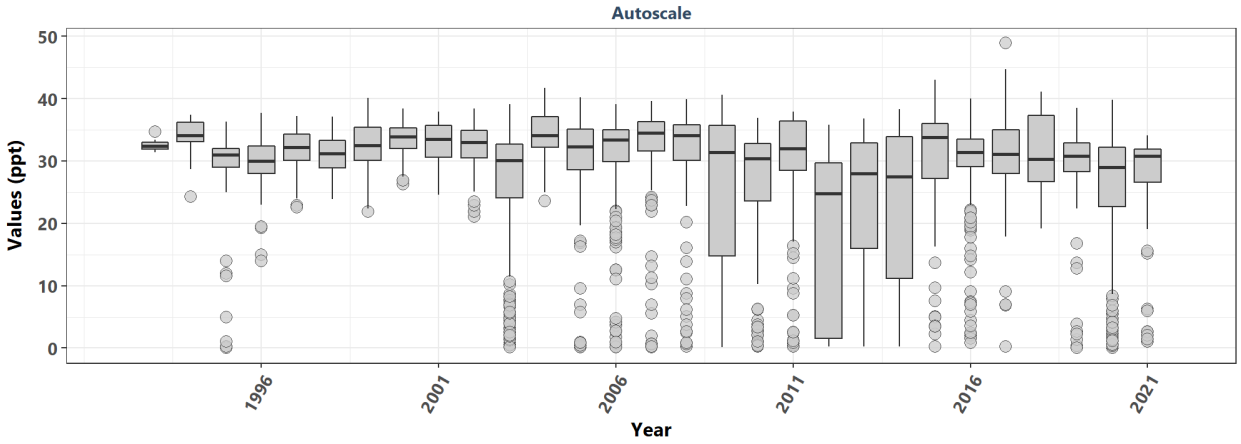


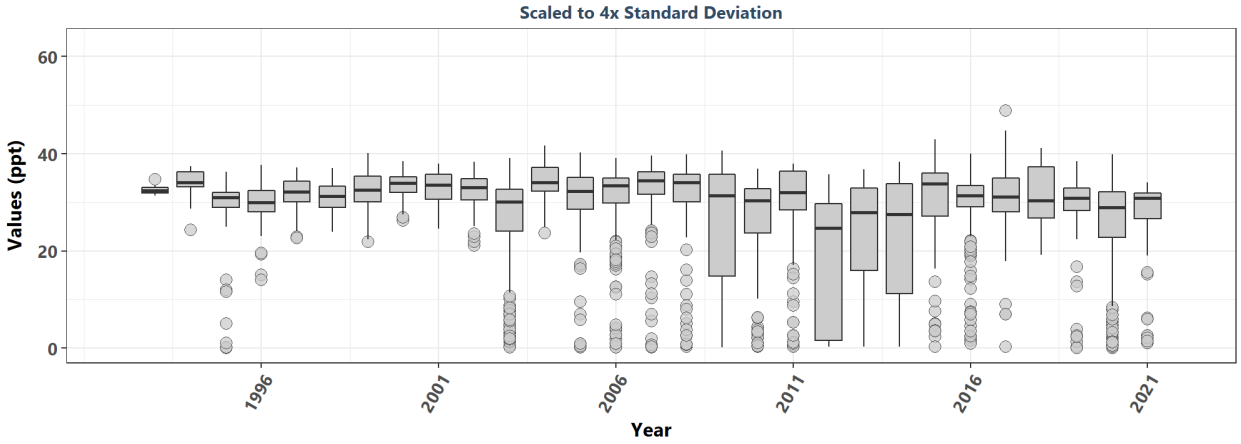

Big Bend Seagrasses Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

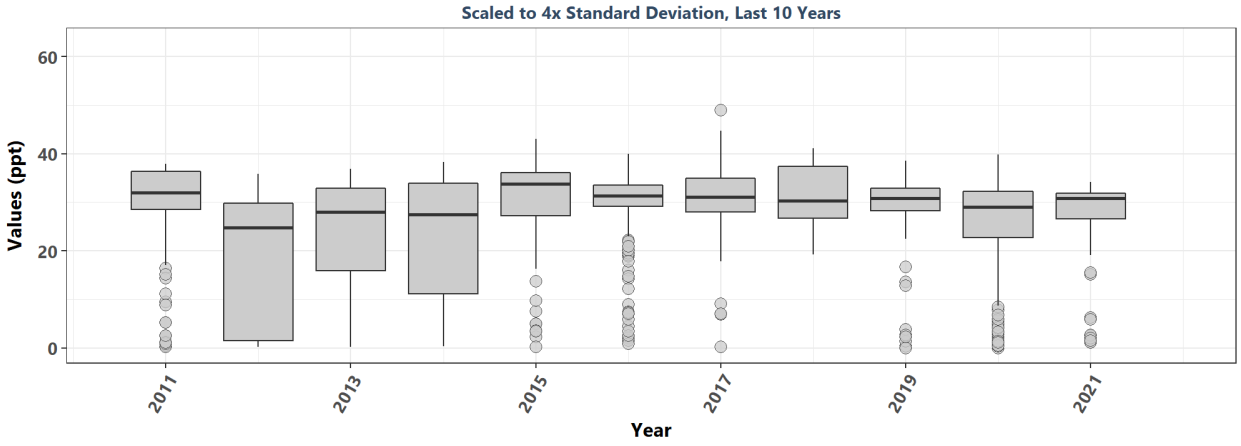

Big Bend Seagrasses Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

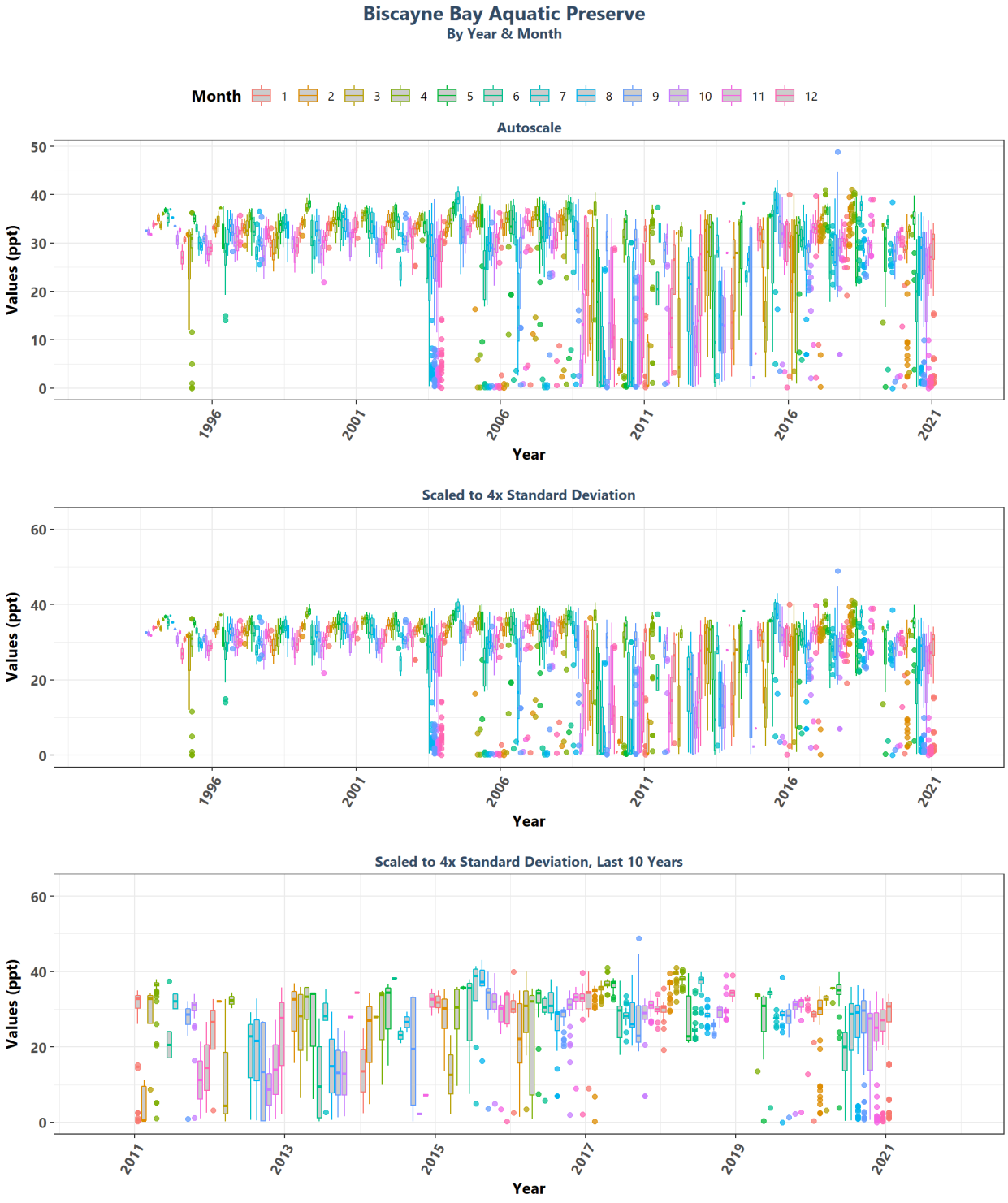
Biscayne Bay Aquatic Preserve

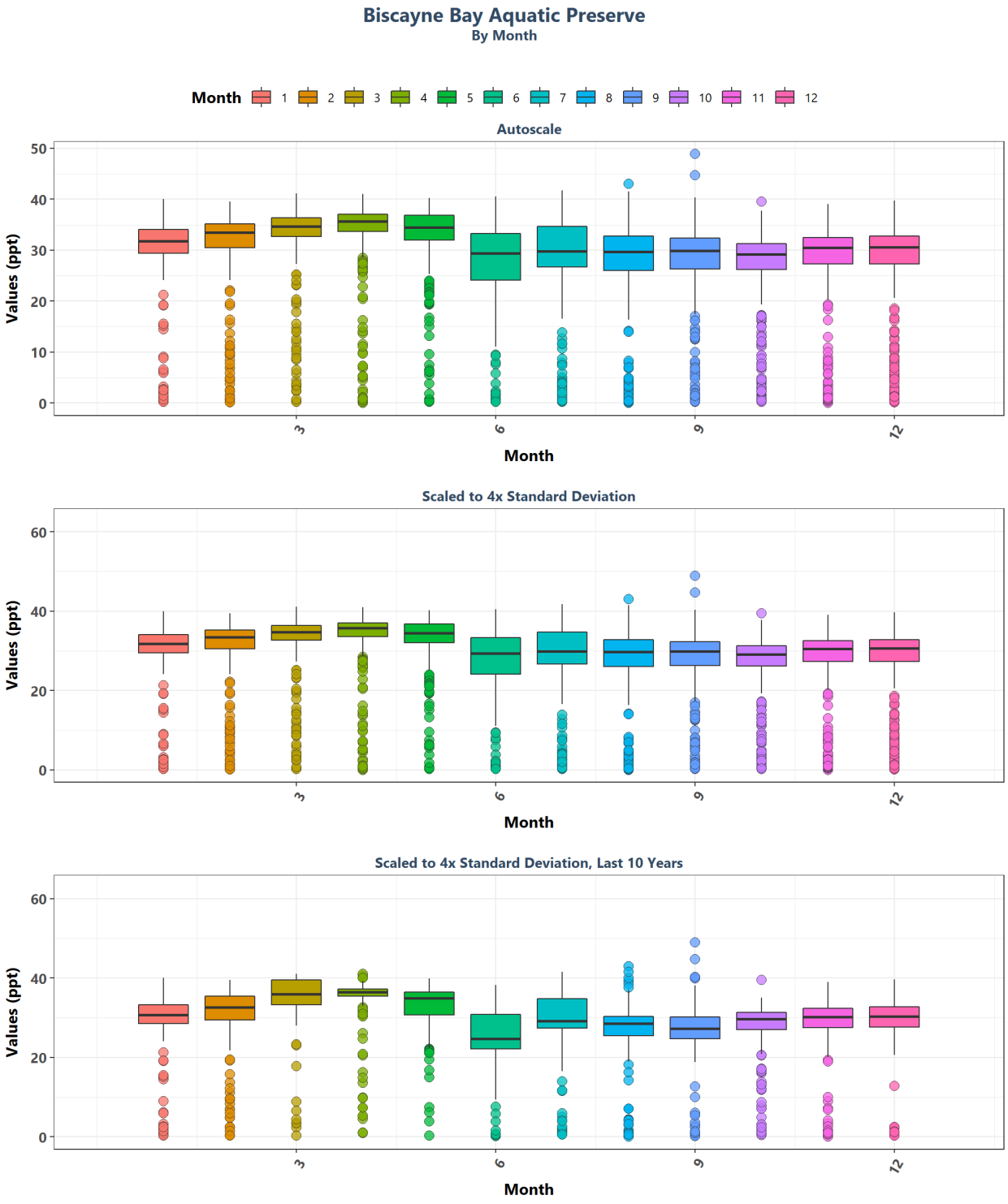
By Year

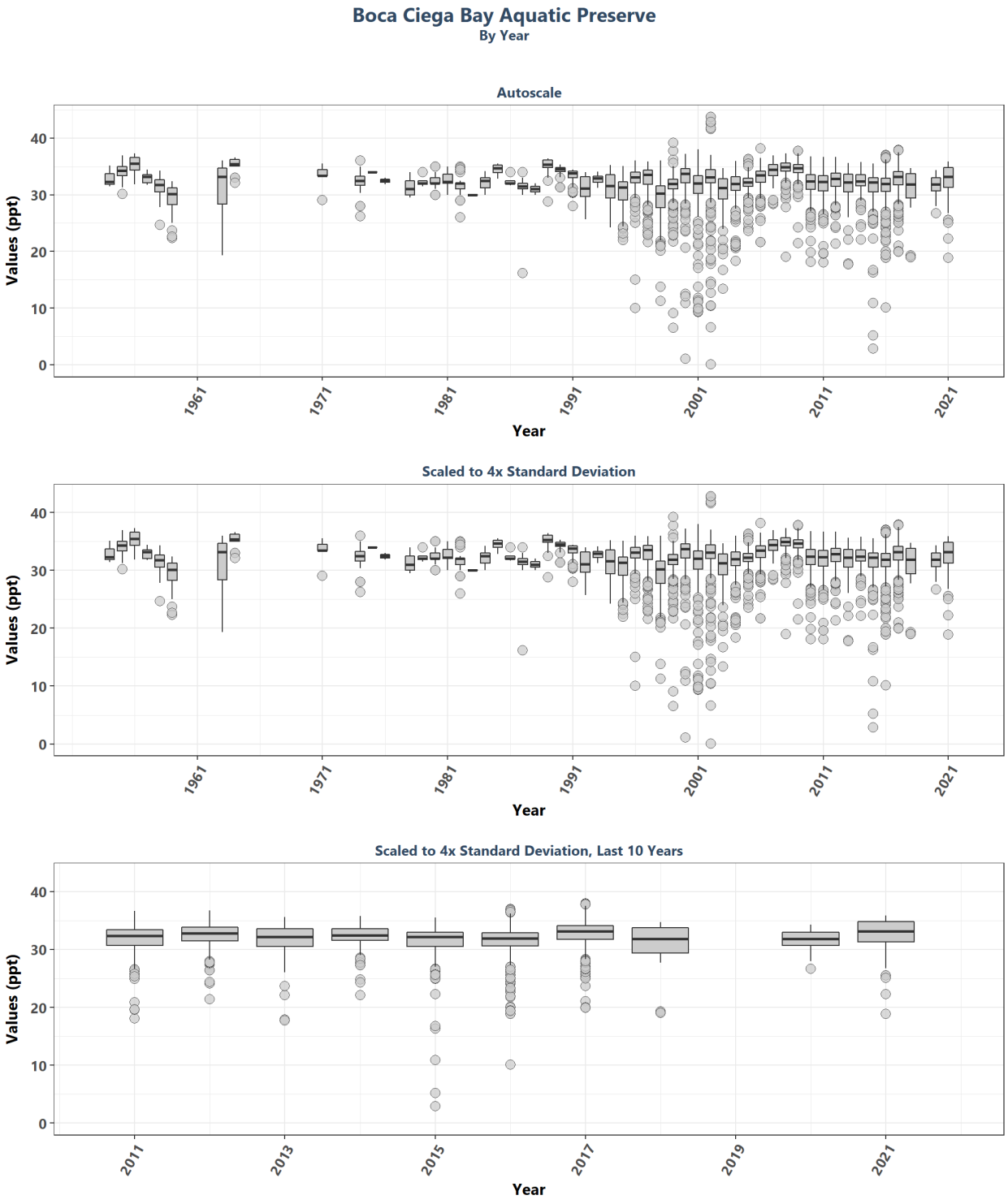


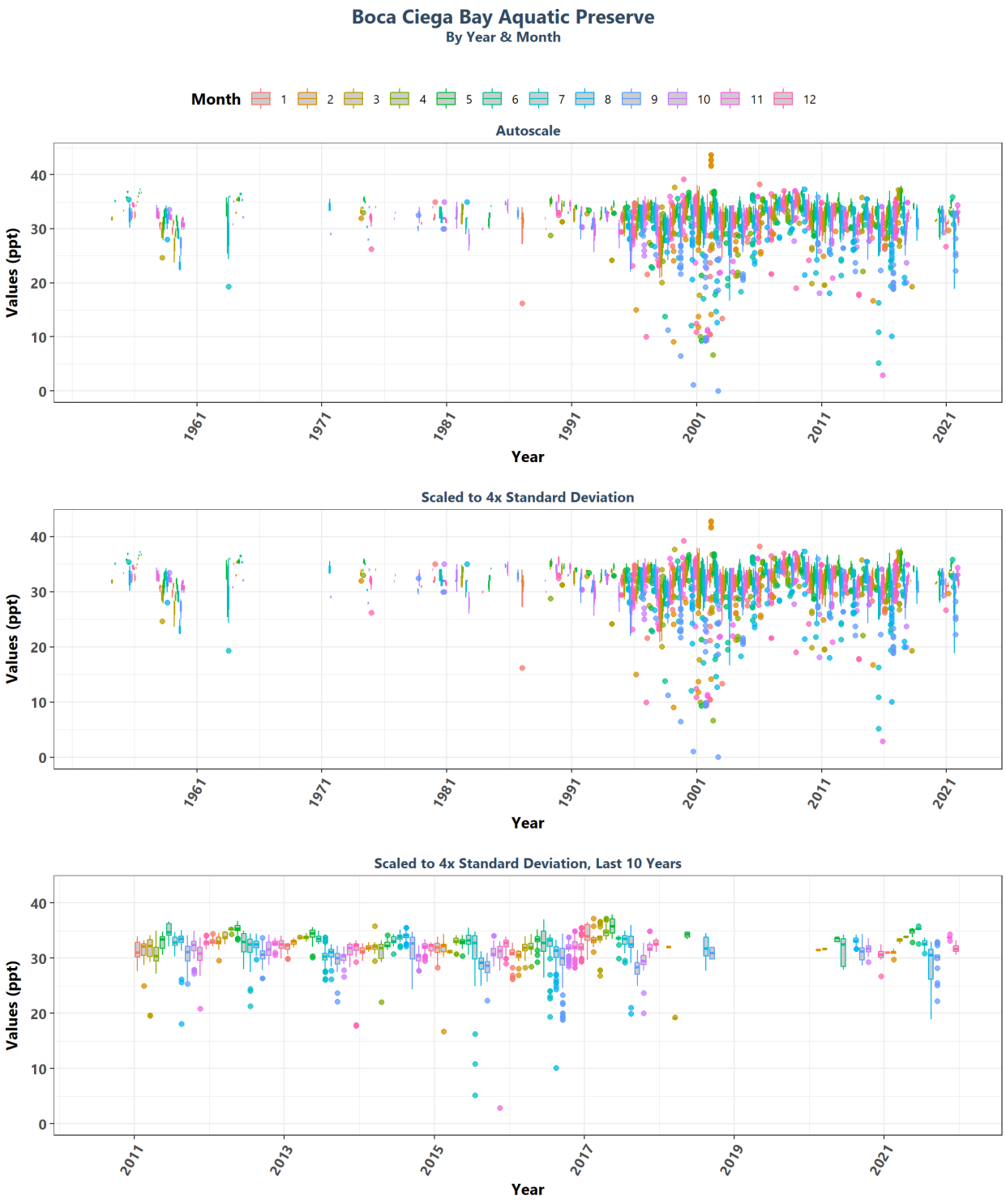


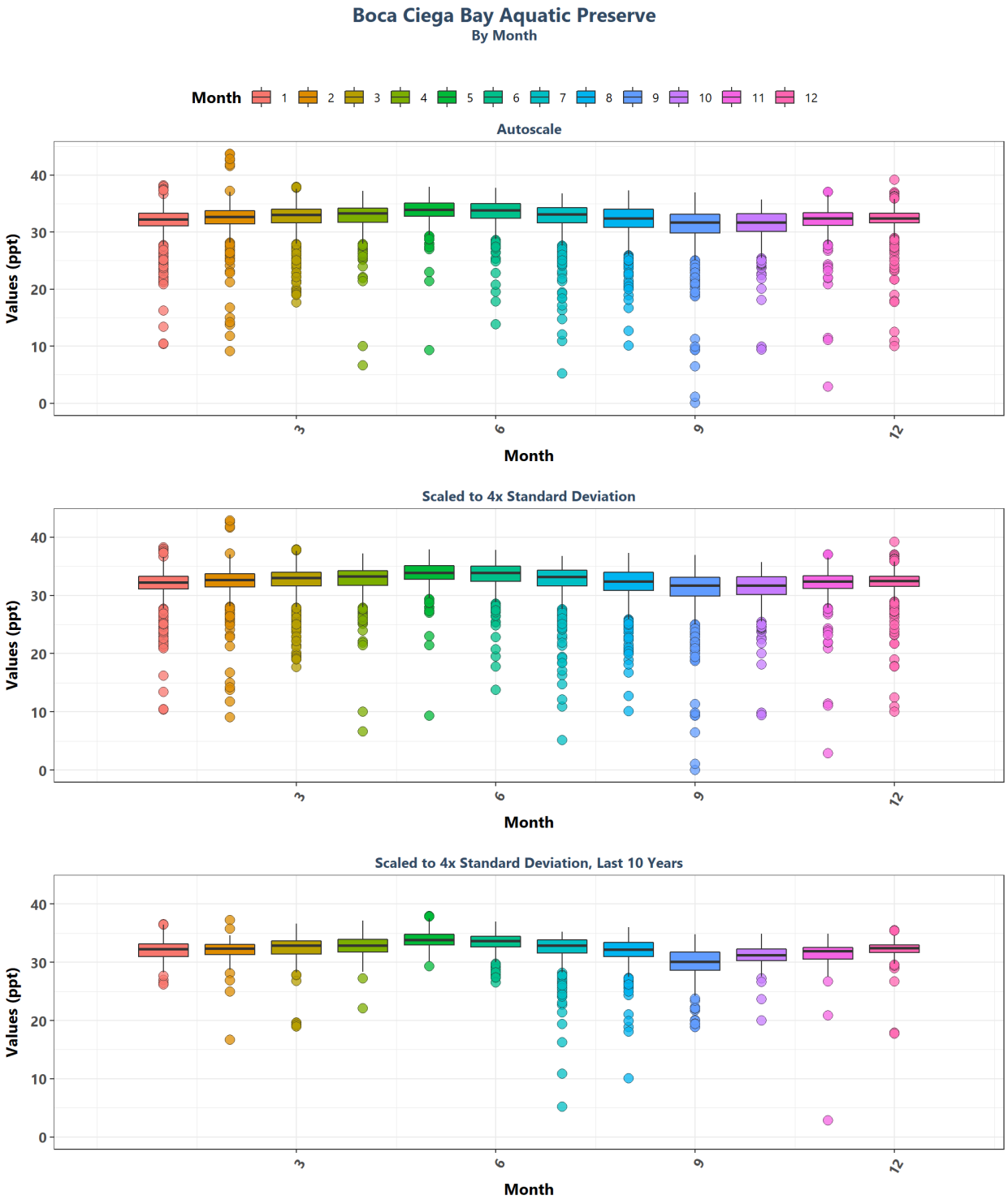



Biscayne Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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

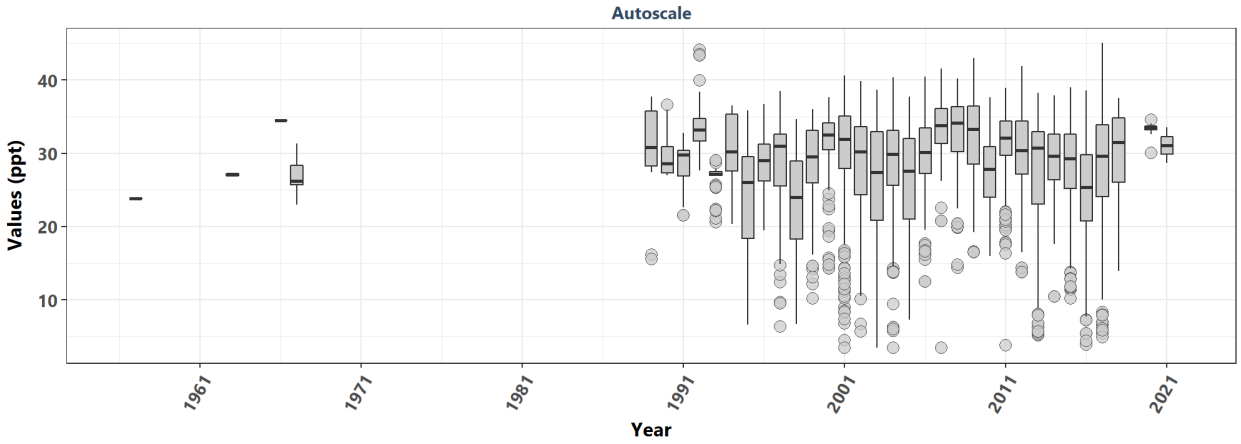

Boca Ciega Bay Aquatic Preserve
By Year
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

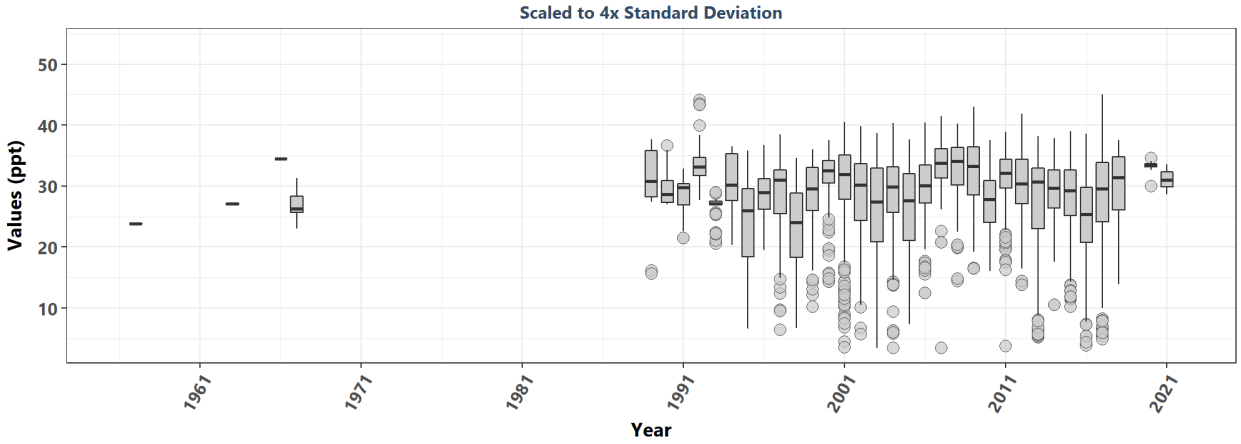

Boca Ciega Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

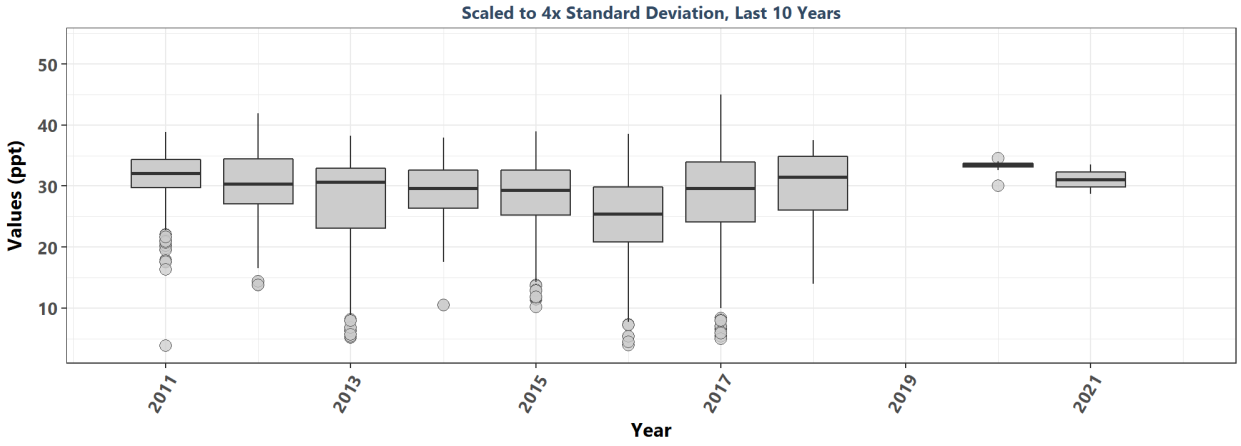

Boca Ciega Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

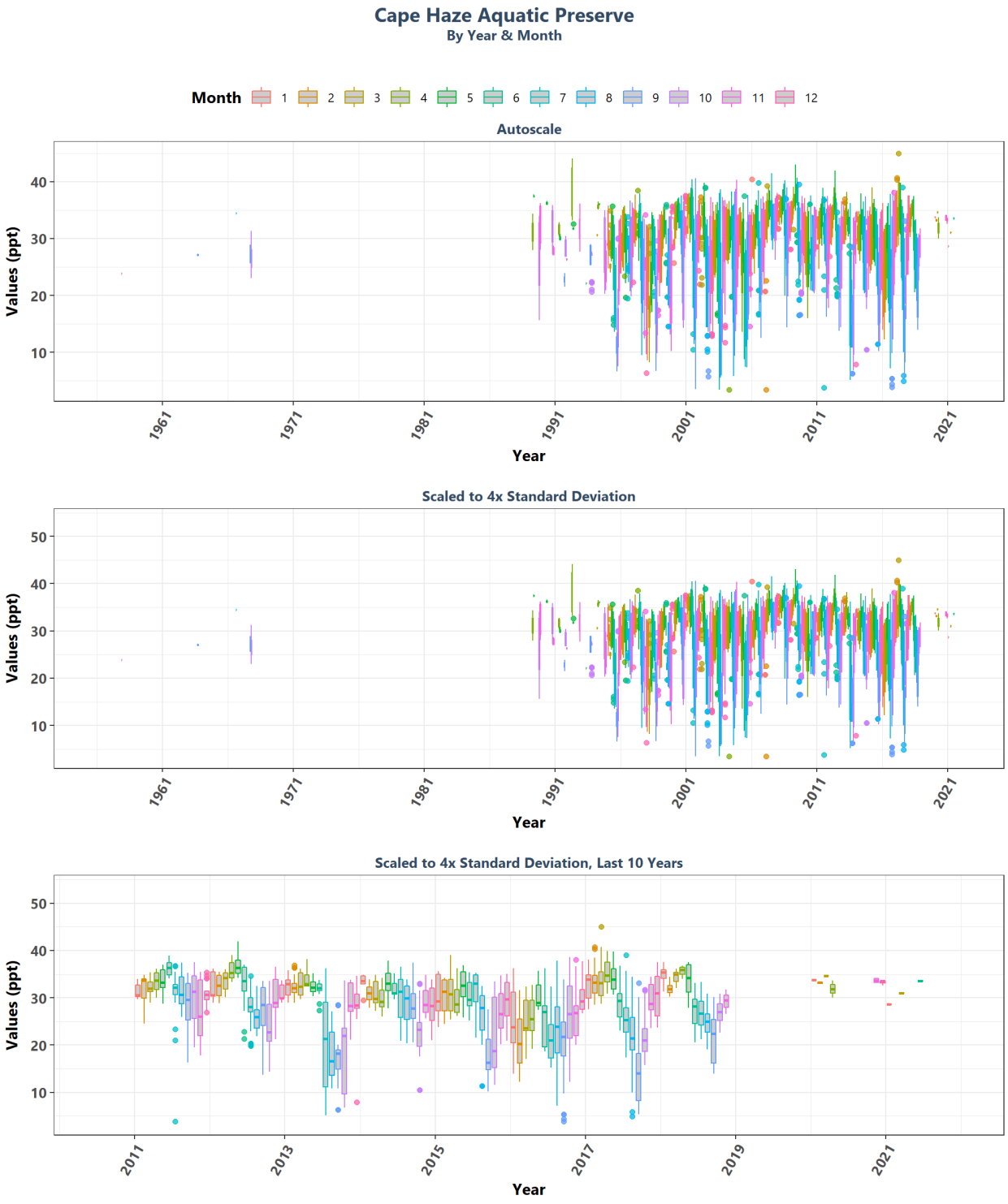
Cape Haze Aquatic Preserve

By Year

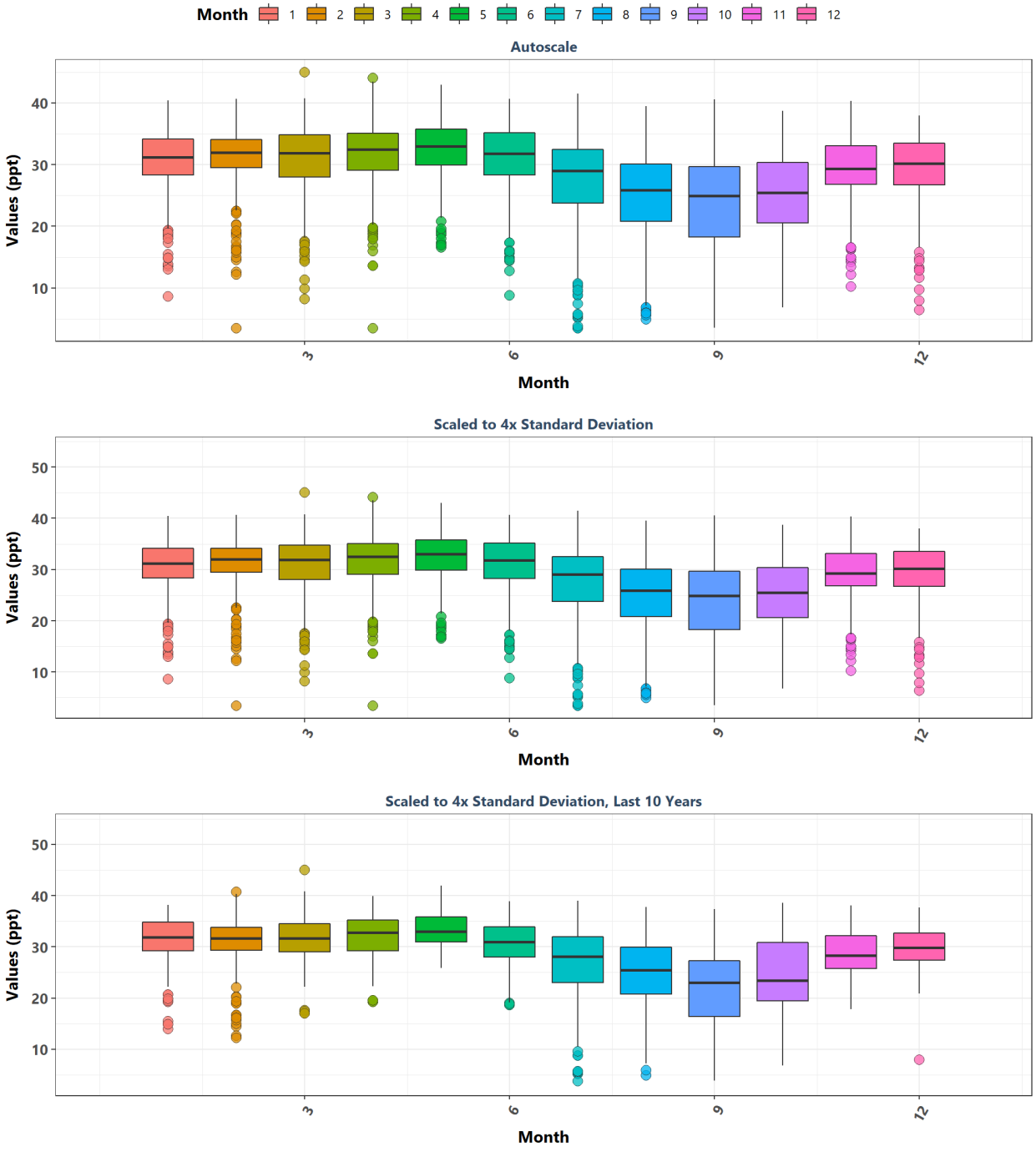






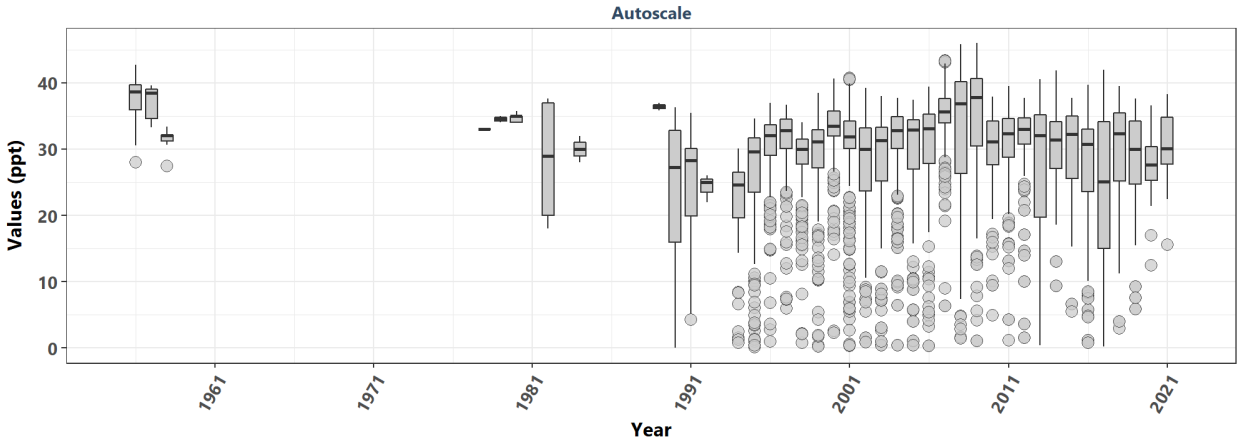

Cape Haze Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

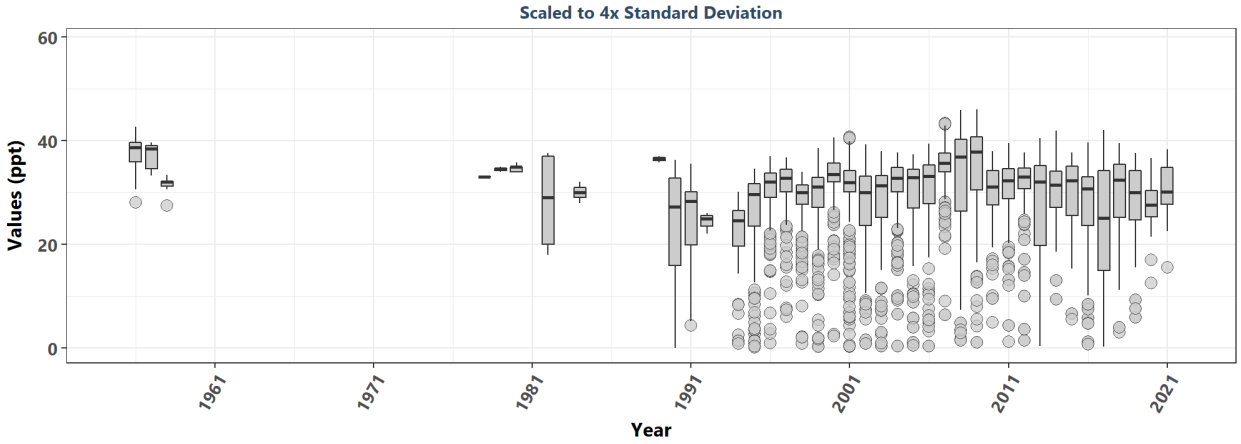
Cape Haze Aquatic Preserve
By Month

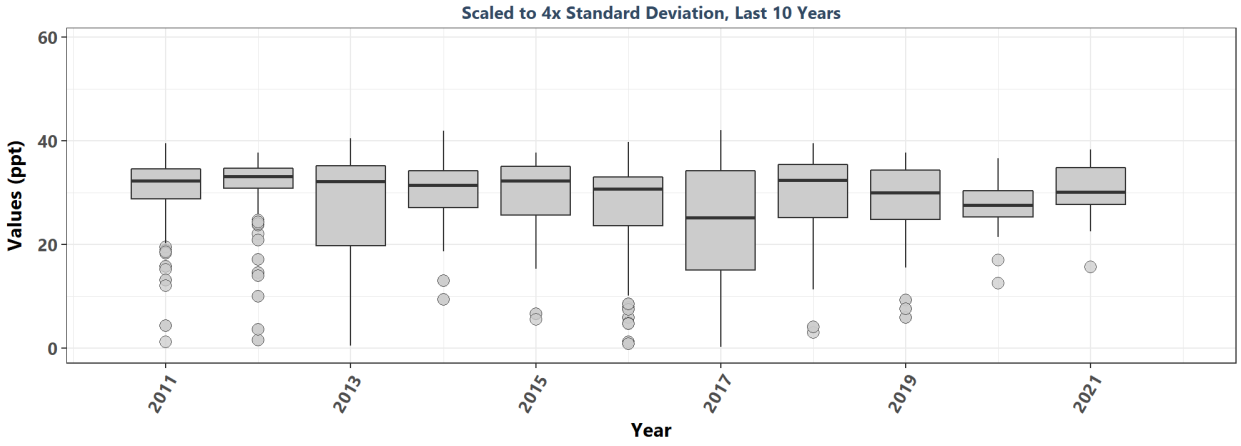


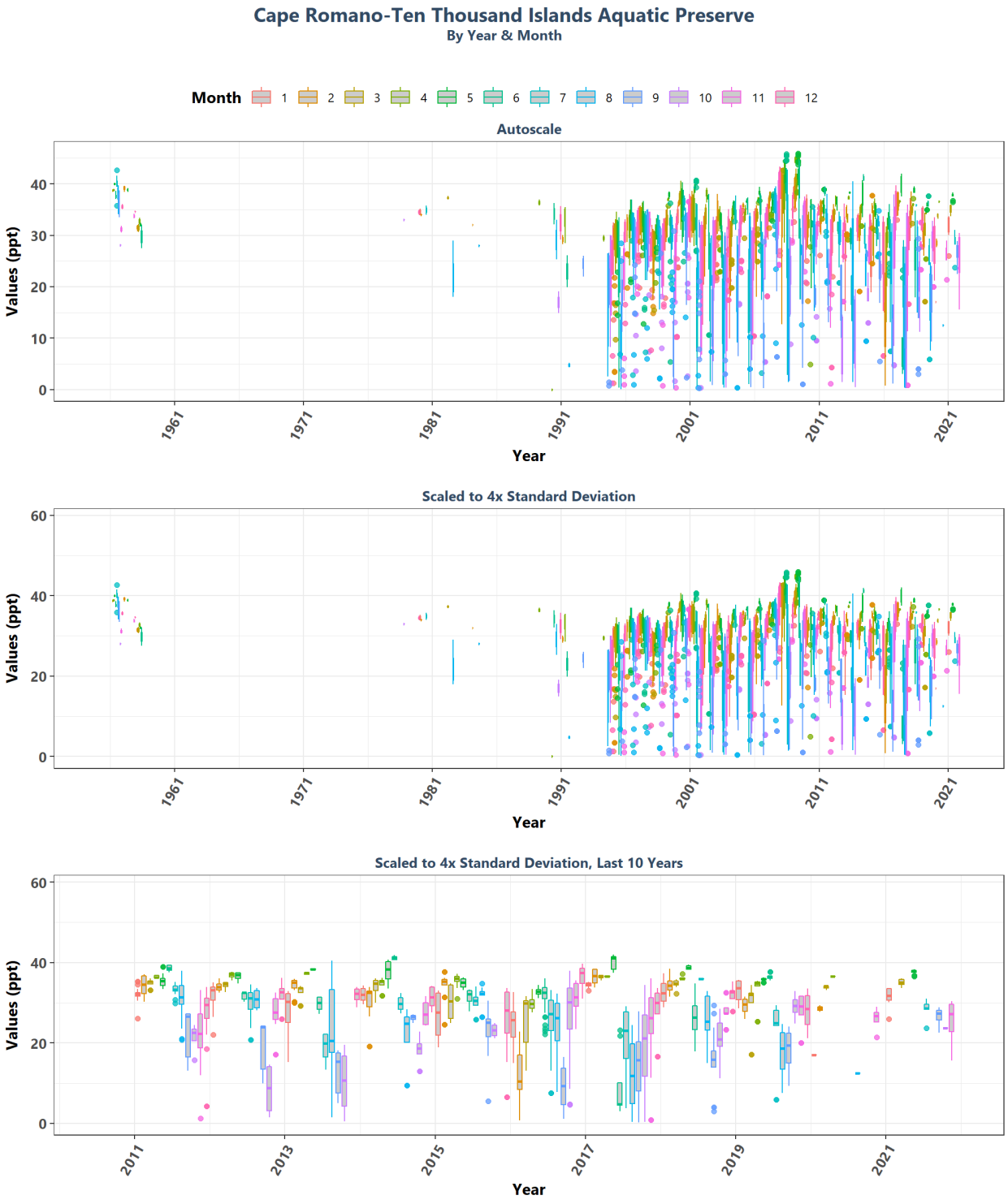
Cape Romano-Ten Thousand Islands Aquatic Preserve

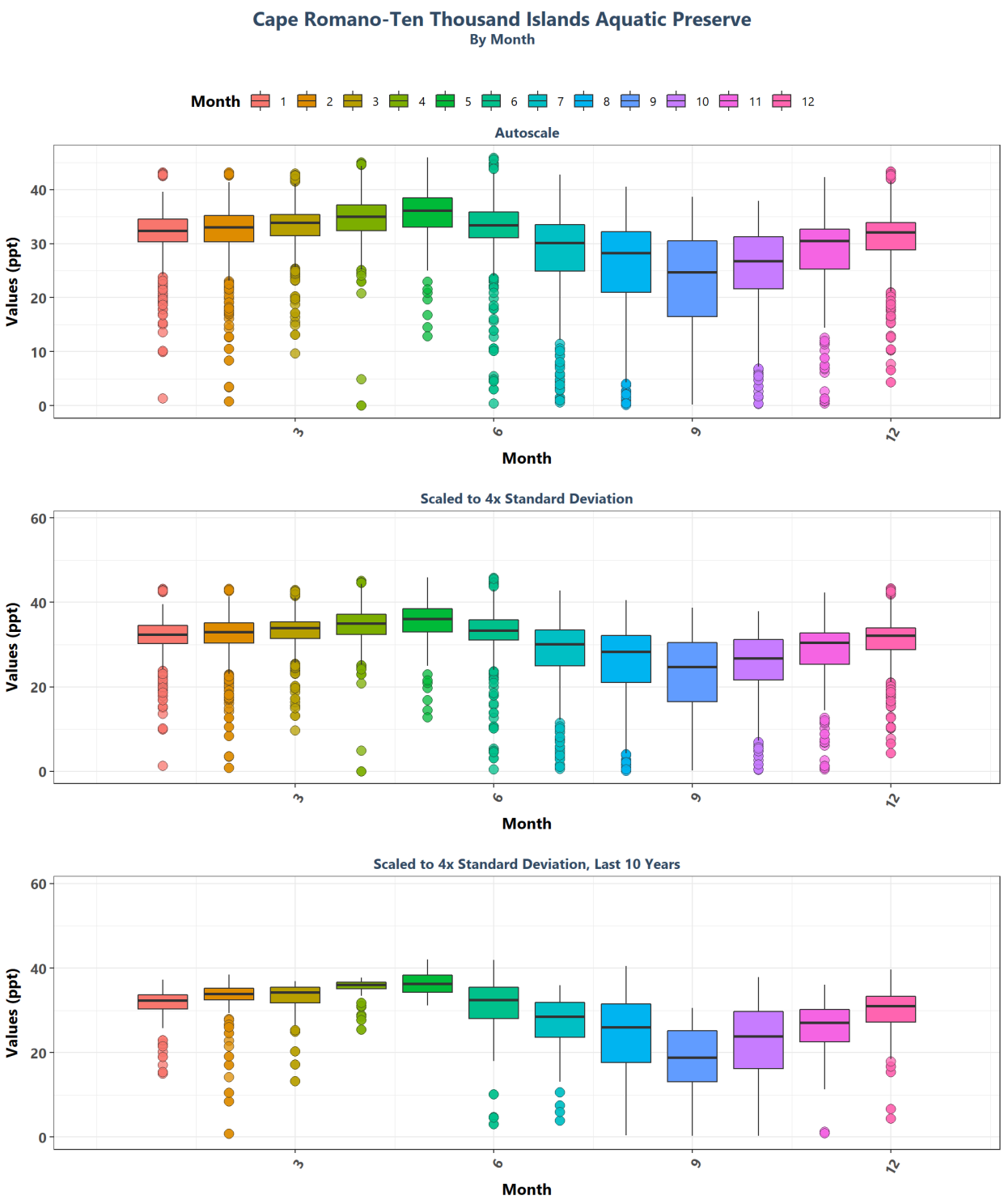
By Year





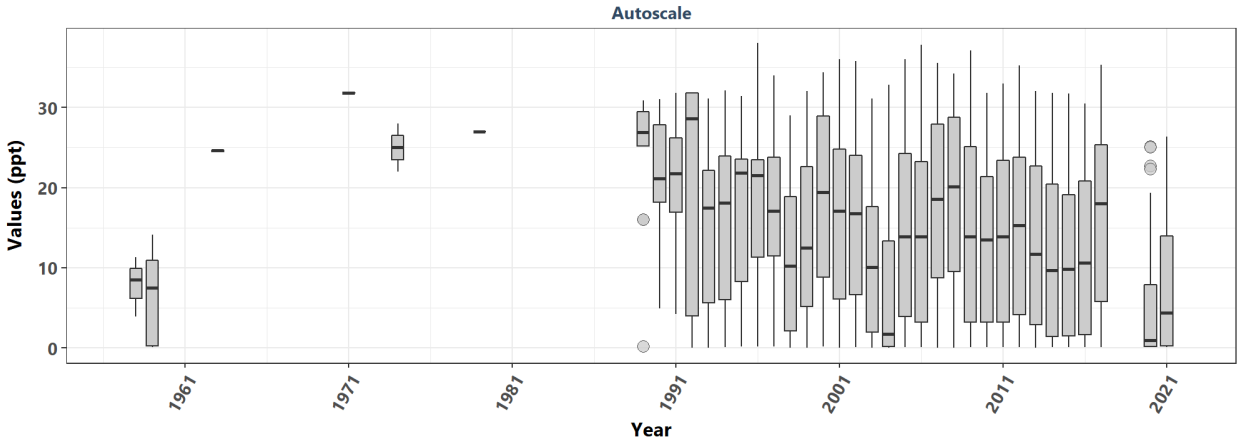


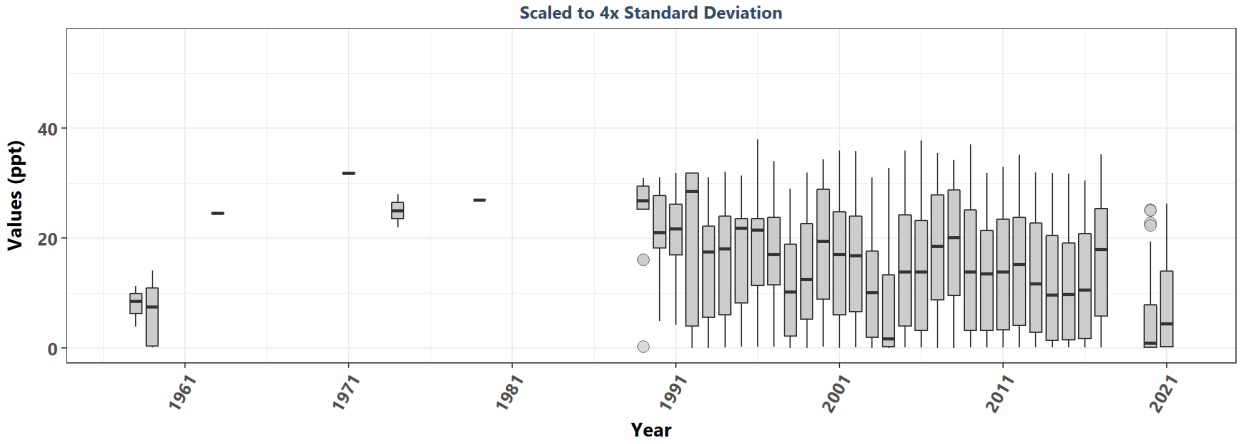

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

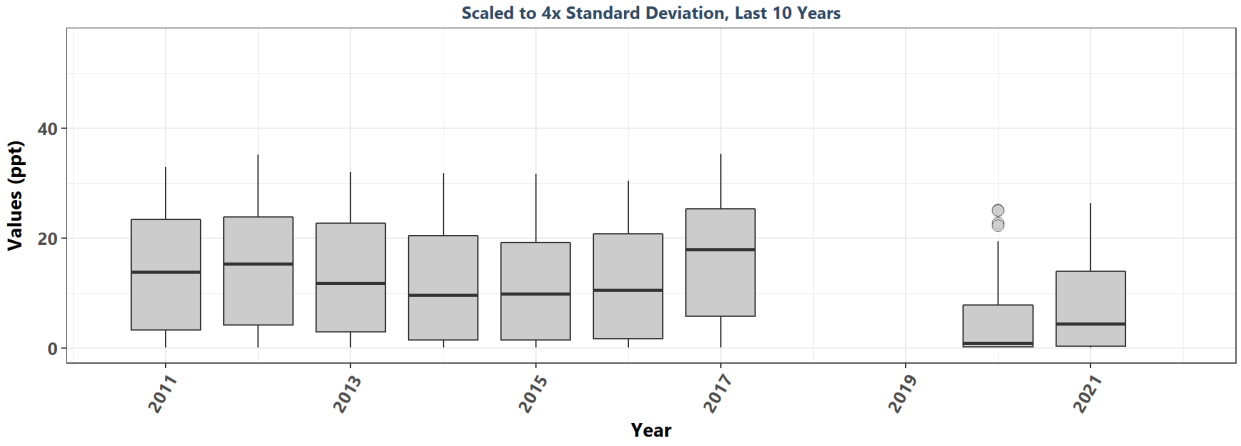

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0 20 40 60
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0 20 40 60
3 6 9 12
Month

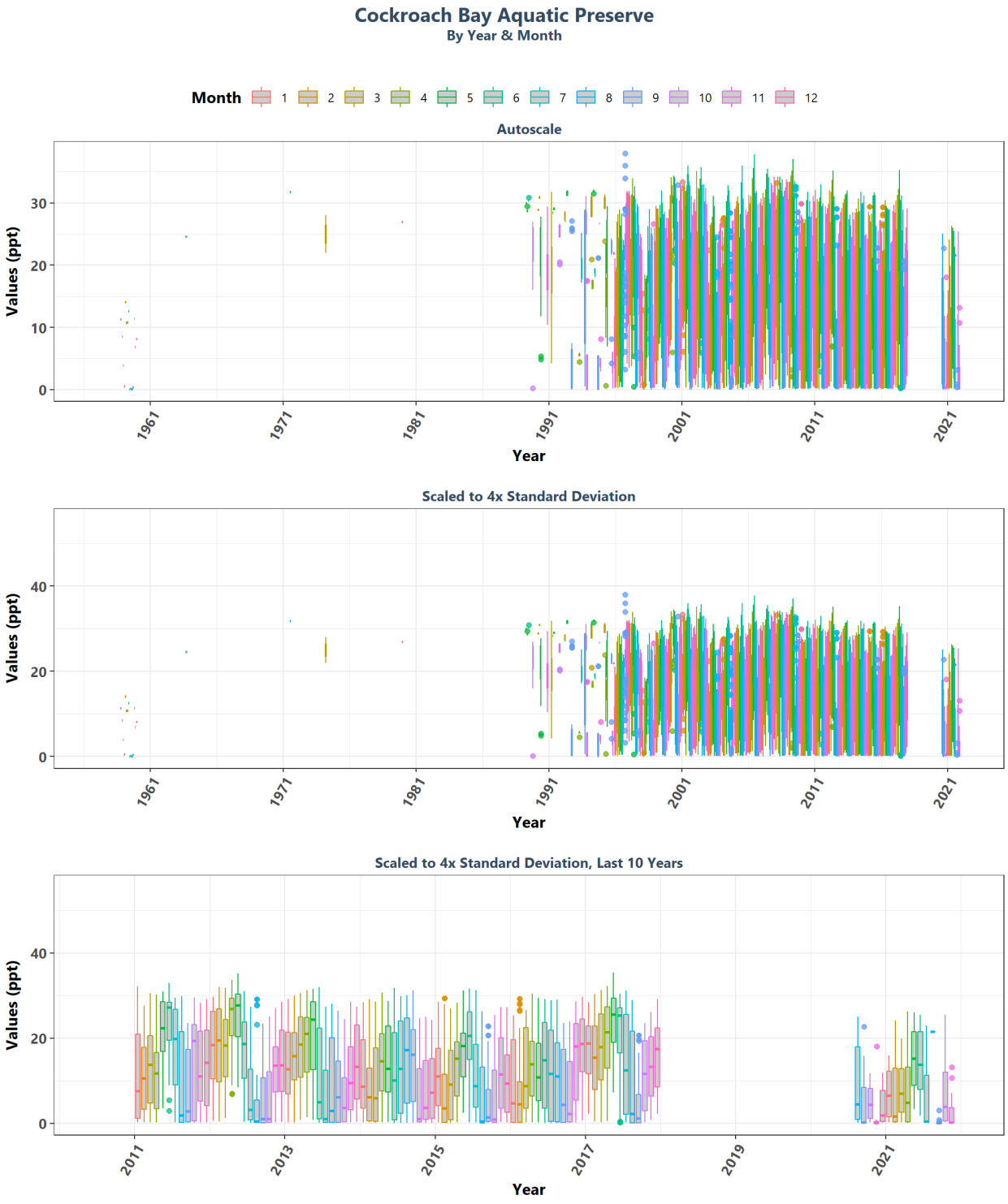
Cockroach Bay Aquatic Preserve

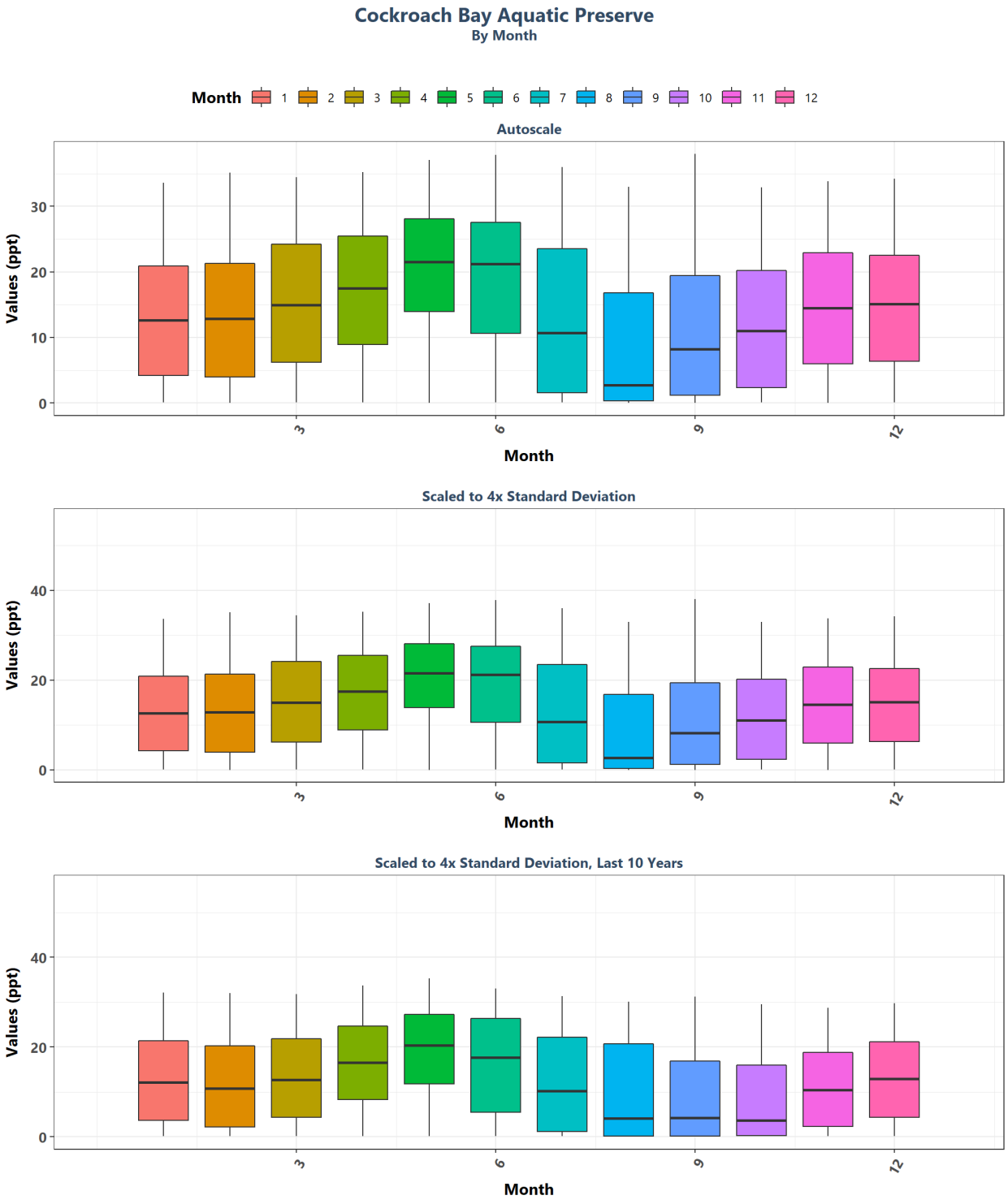
By Year

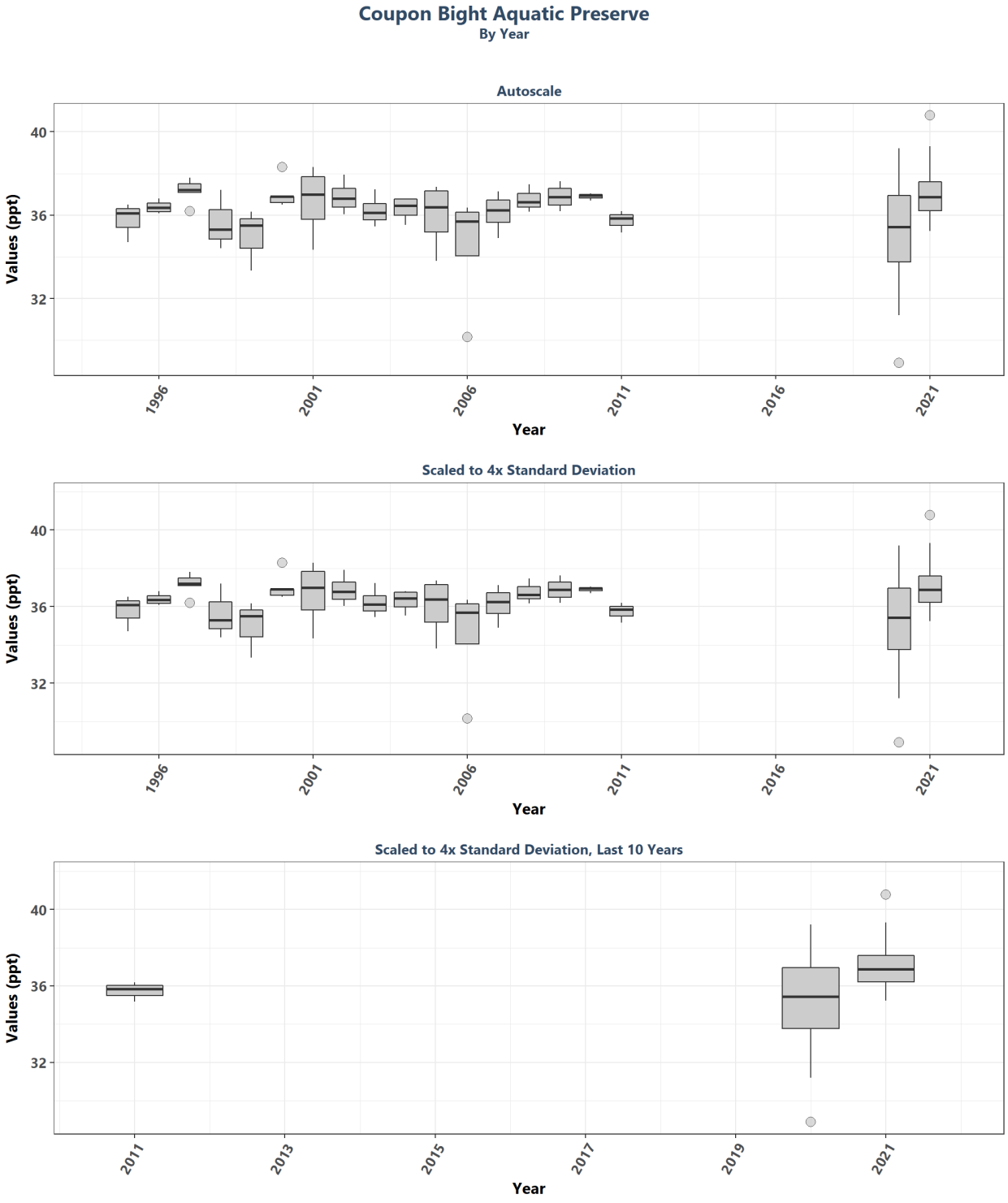


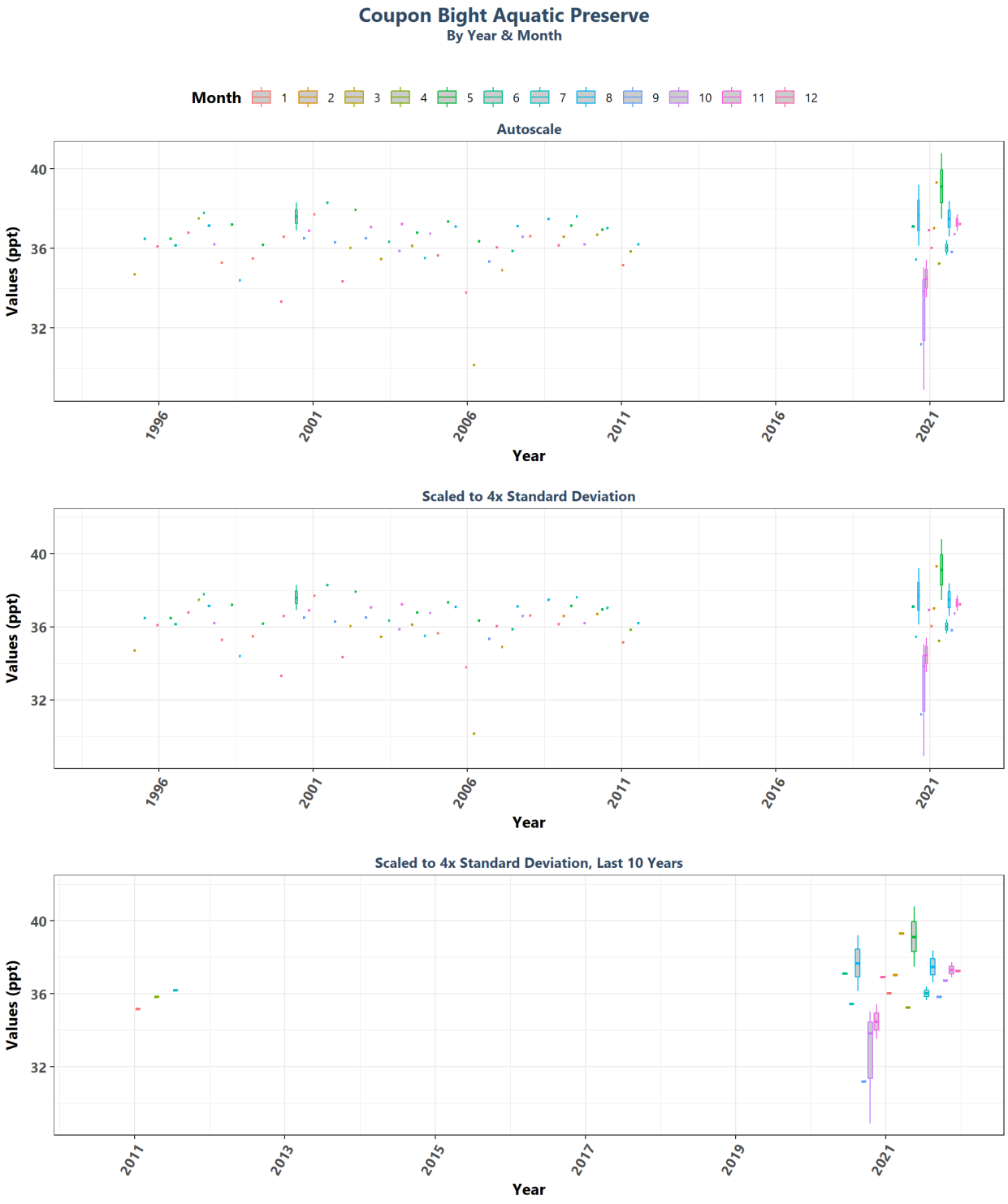


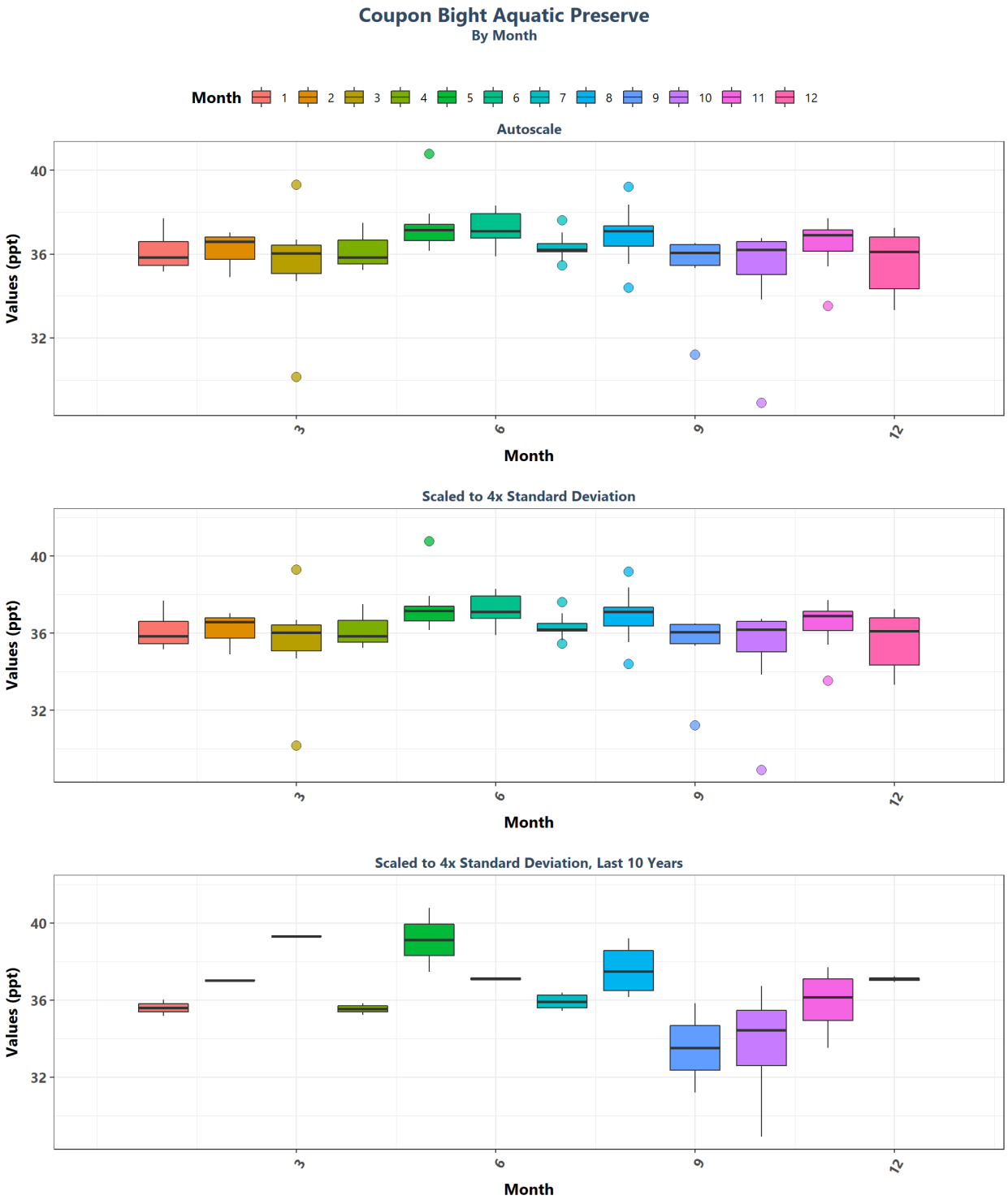



Cockroach Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


Cockroach Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
3
6
9
12
Month

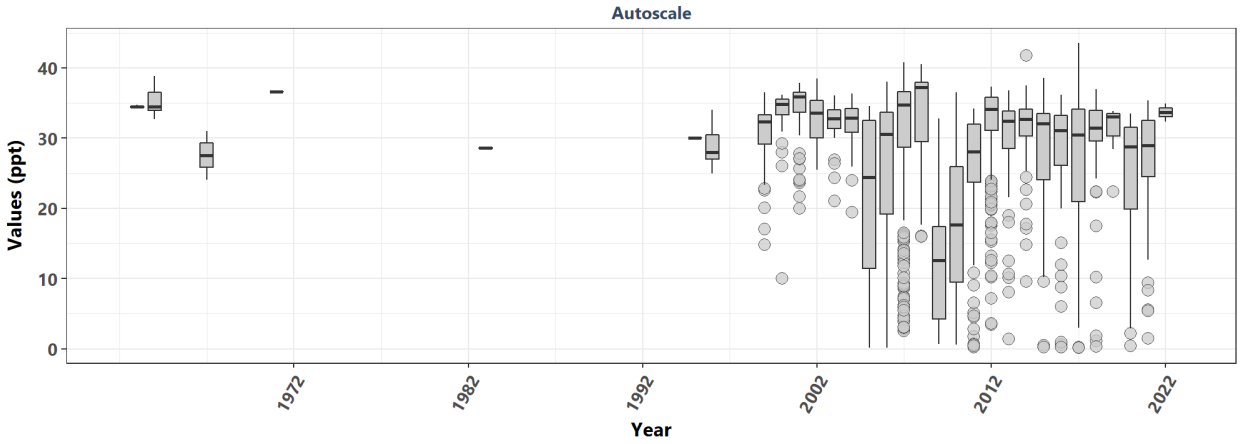

Coupon Bight Aquatic Preserve
By Year
Autoscale
Values (ppt)
40
36
32
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
40
36
32
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
40
36
32
2011
2013
2015
2017
2019
2021
Year

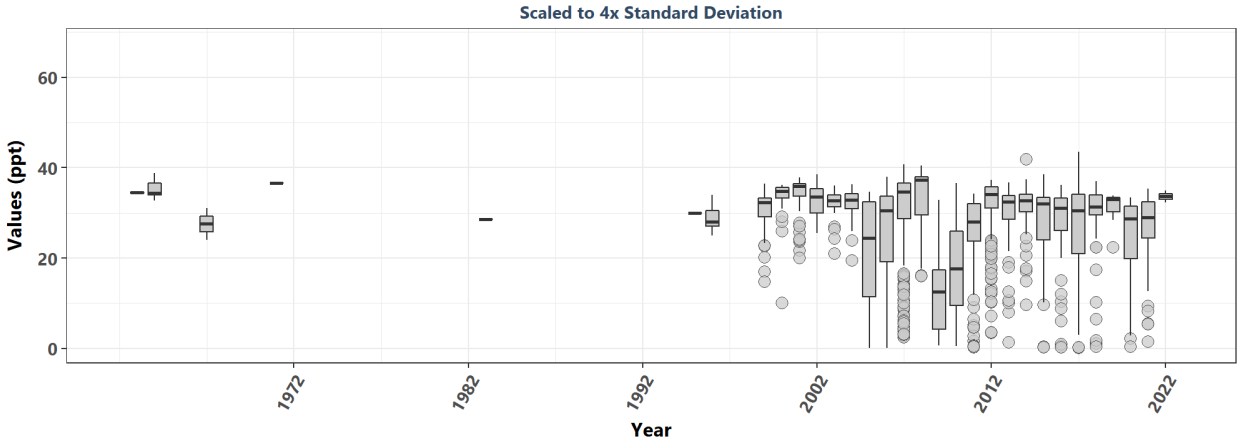

Coupon Bight Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
36
32
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
40
36
32
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
40
36
32
2011
2013
2015
2017
2019
2021
Year

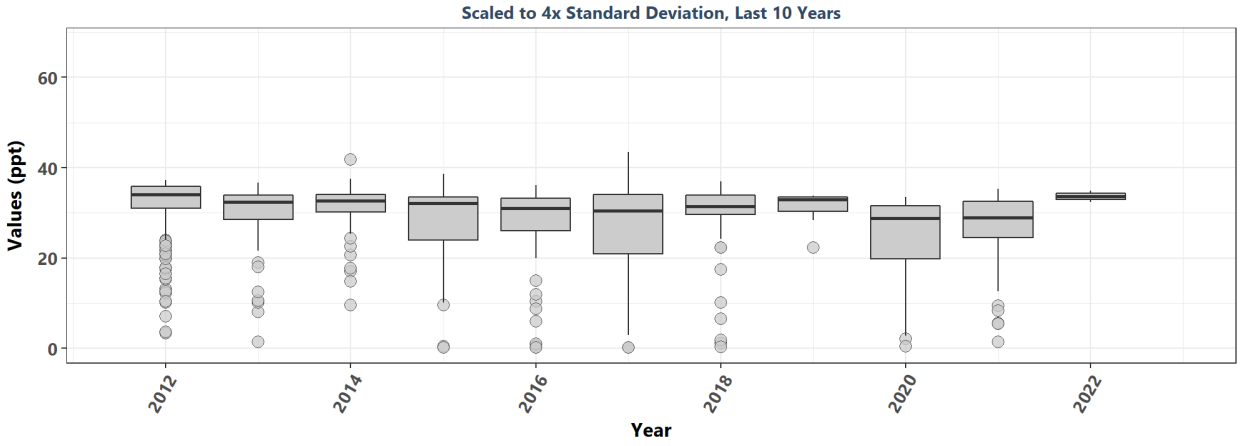

Coupon Bight Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
36
32
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
40
36
32
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
40
36
32
3
6
9
12
Month

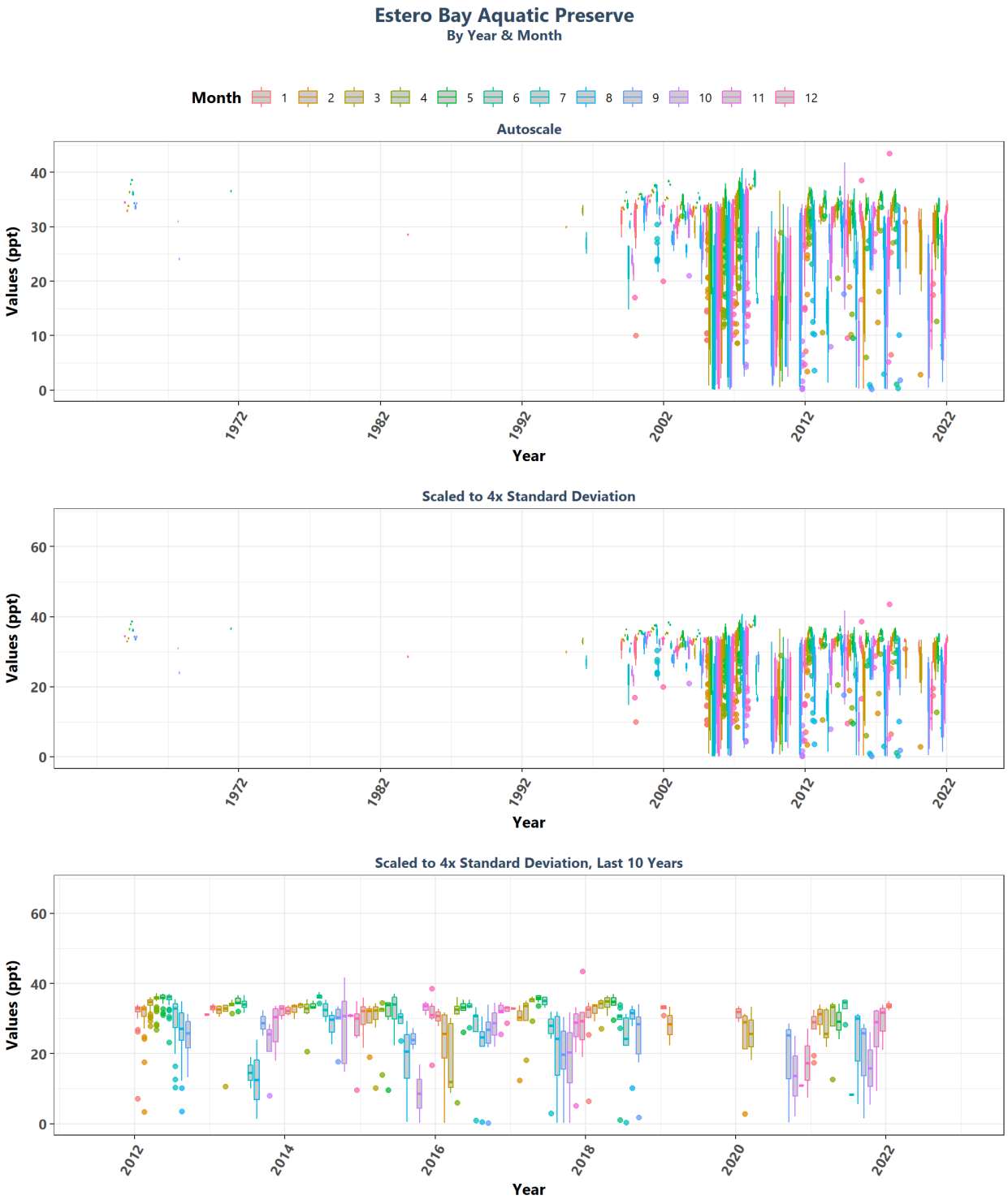
Estero Bay Aquatic Preserve

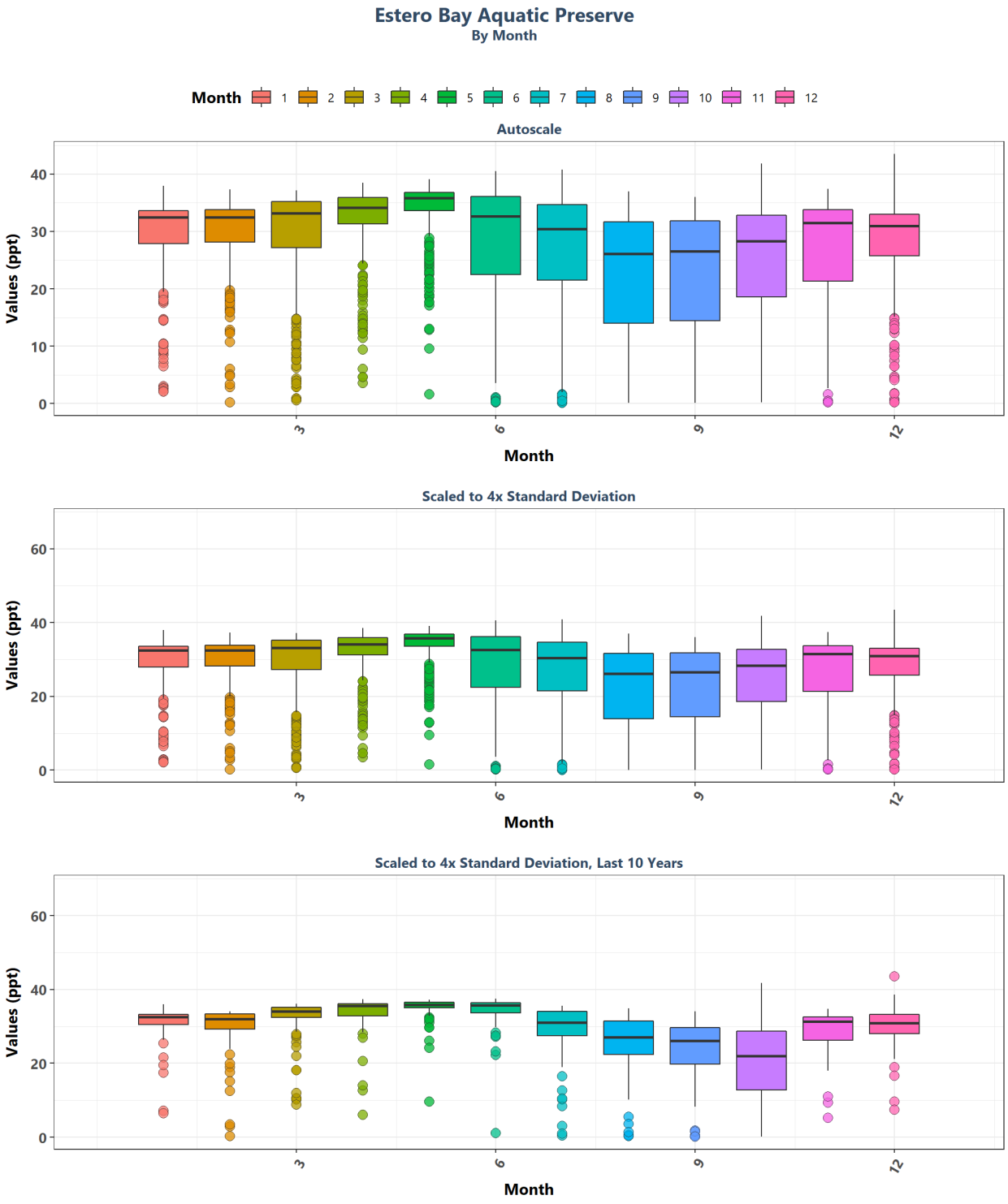
By Year





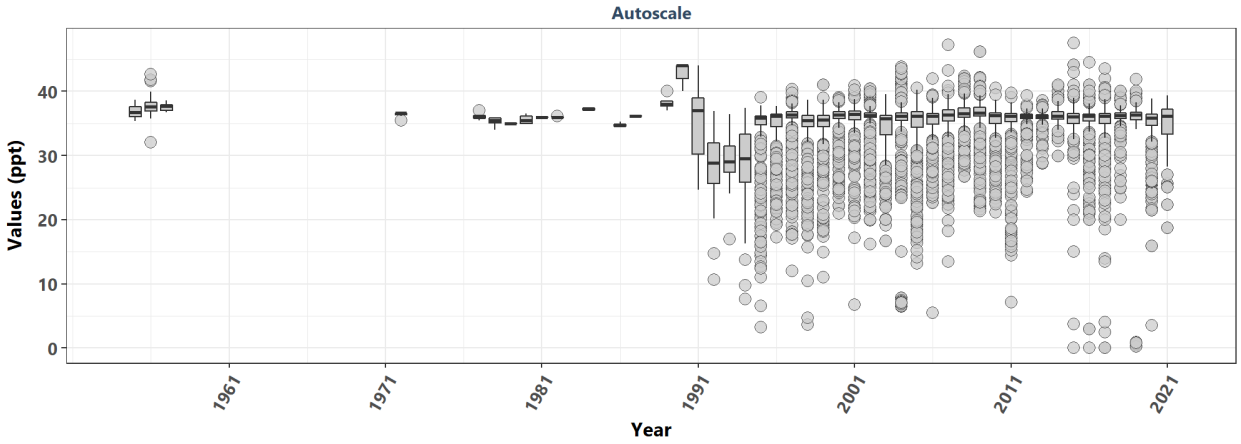


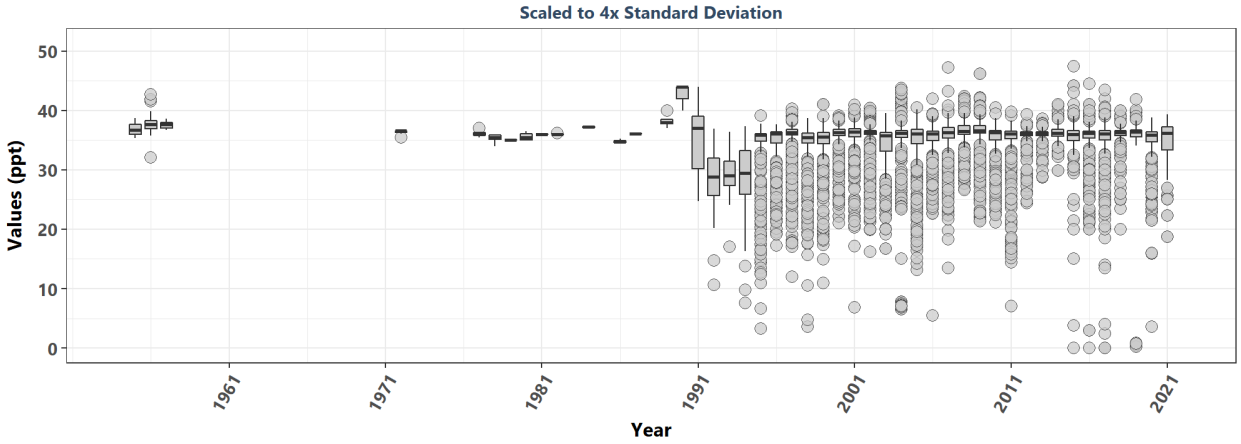

Estero Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

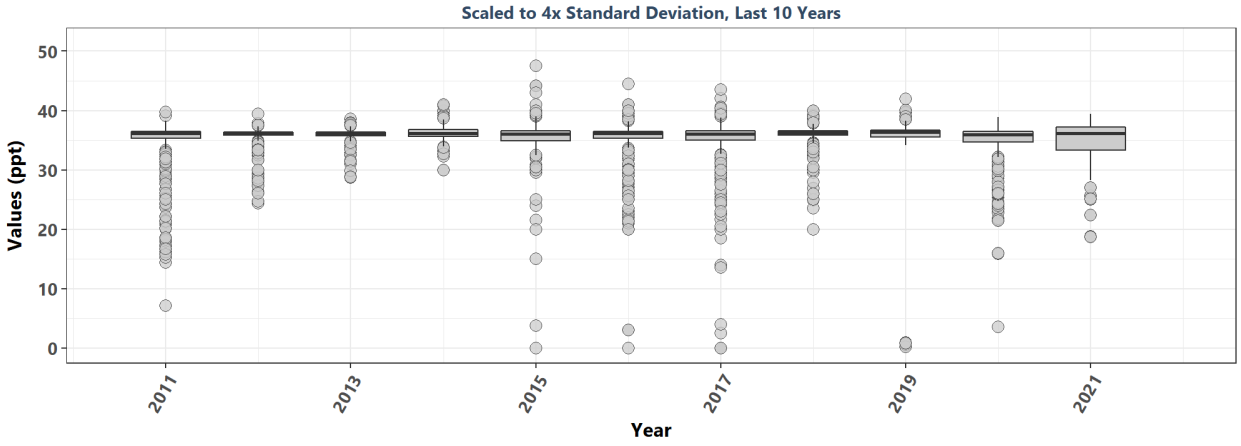

Estero Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0 20 40 60
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0 20 40 60
3 6 9 12
Month

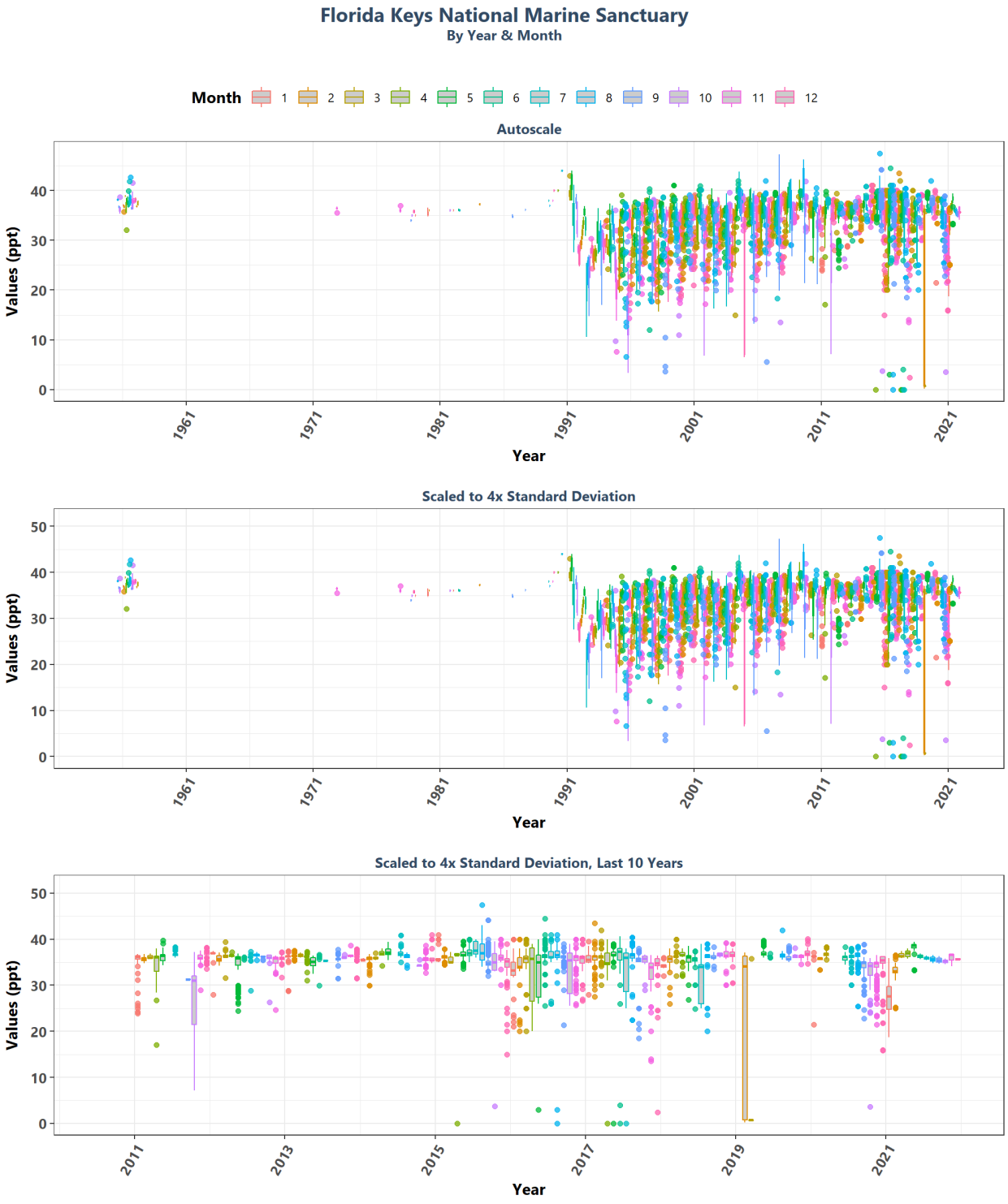
Florida Keys National Marine Sanctuary

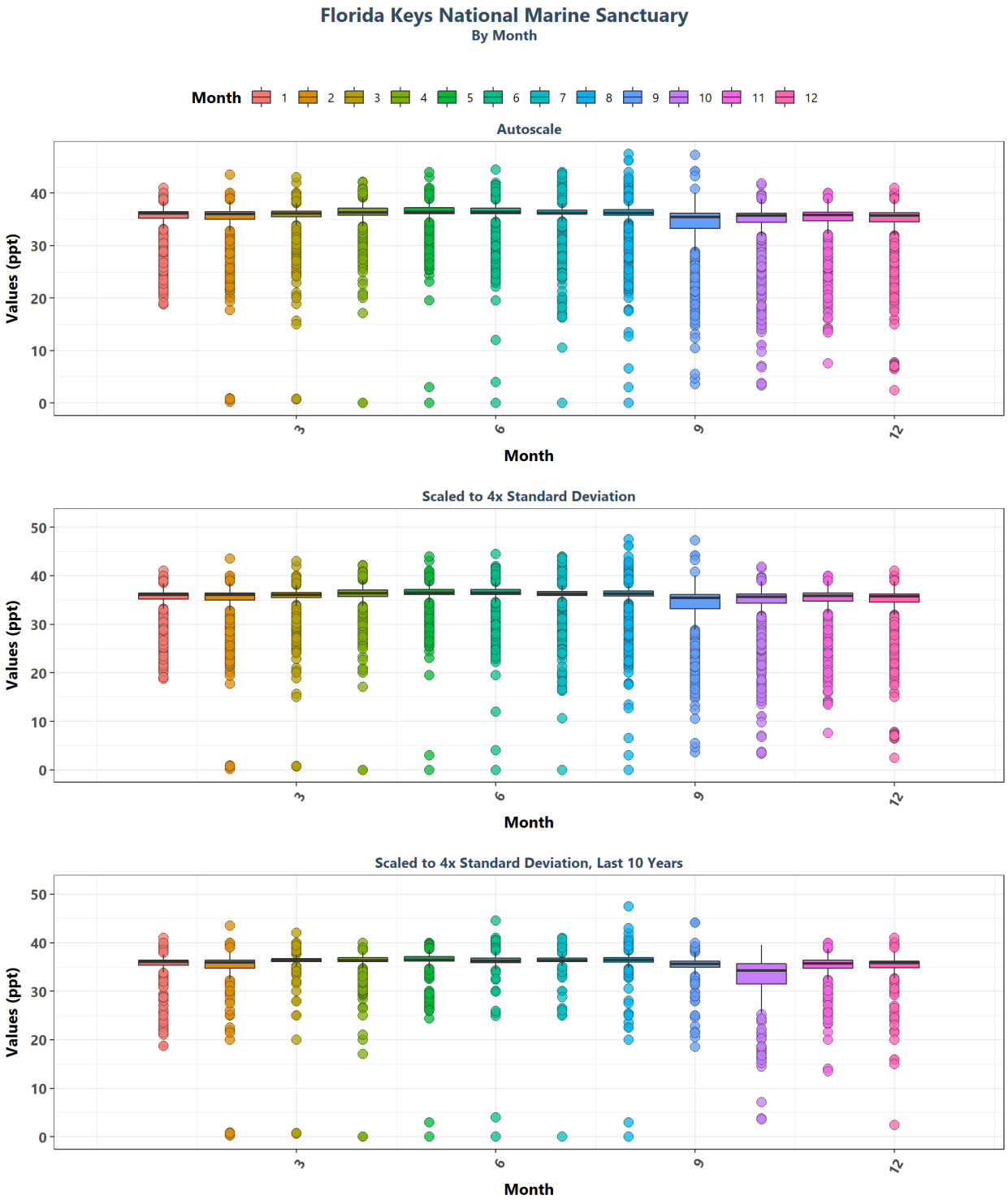
By Year





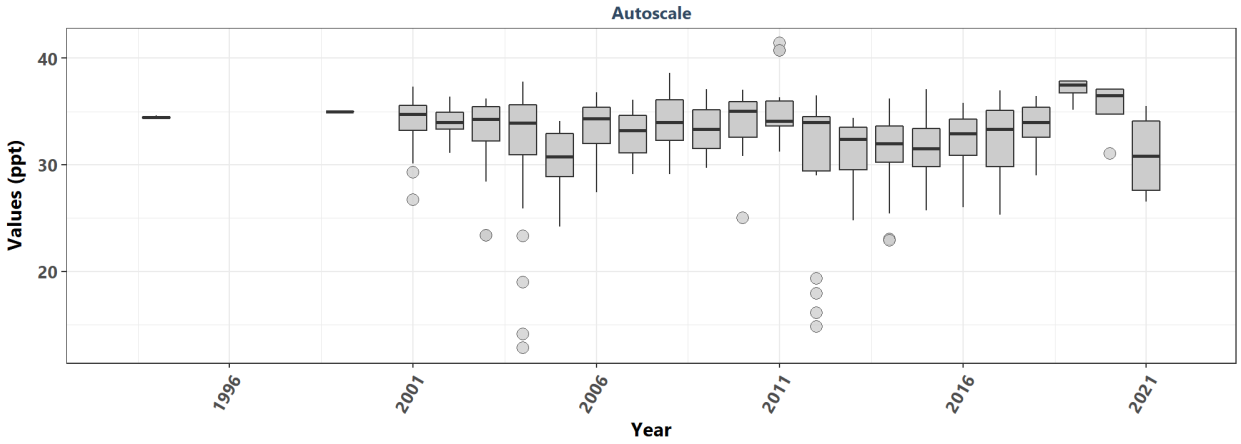


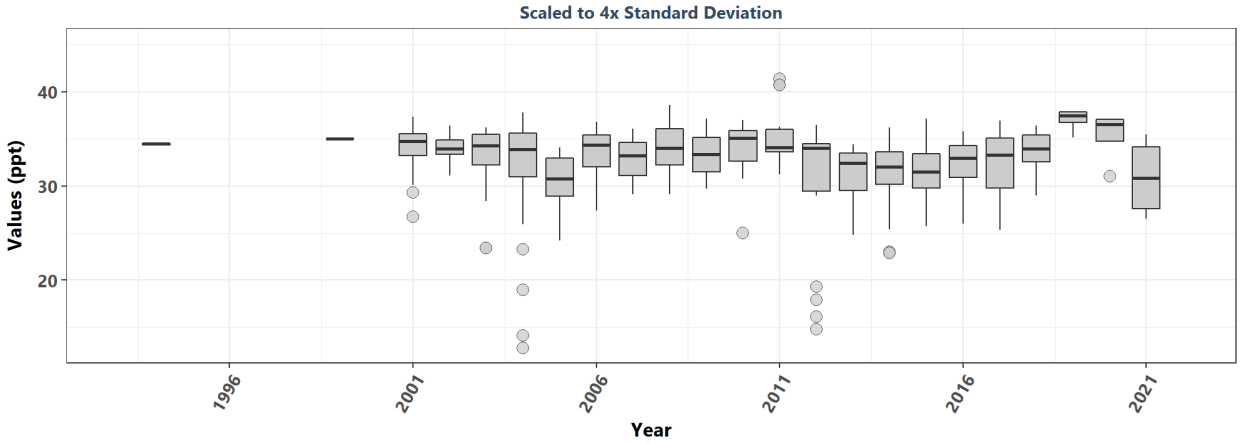

Florida Keys National Marine Sanctuary
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

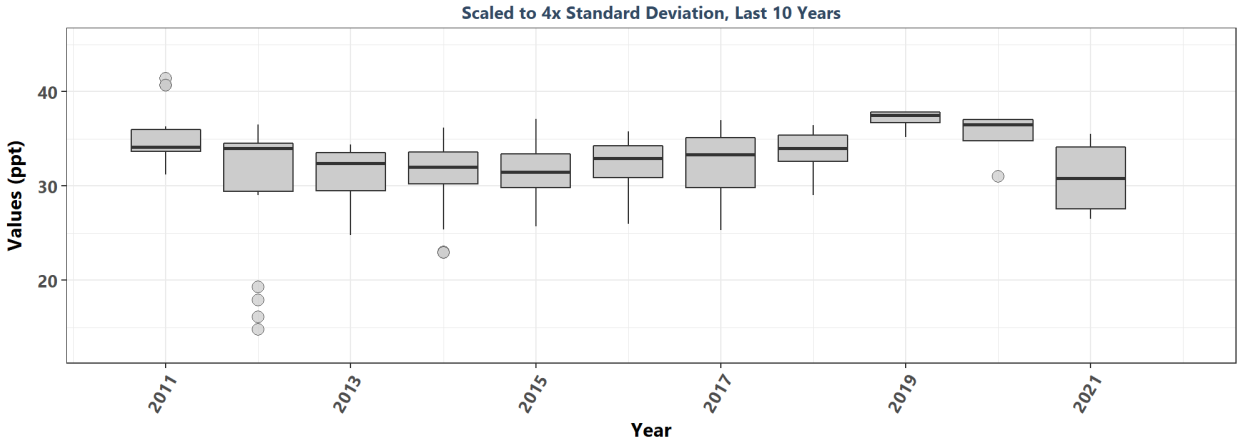

Florida Keys National Marine Sanctuary
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

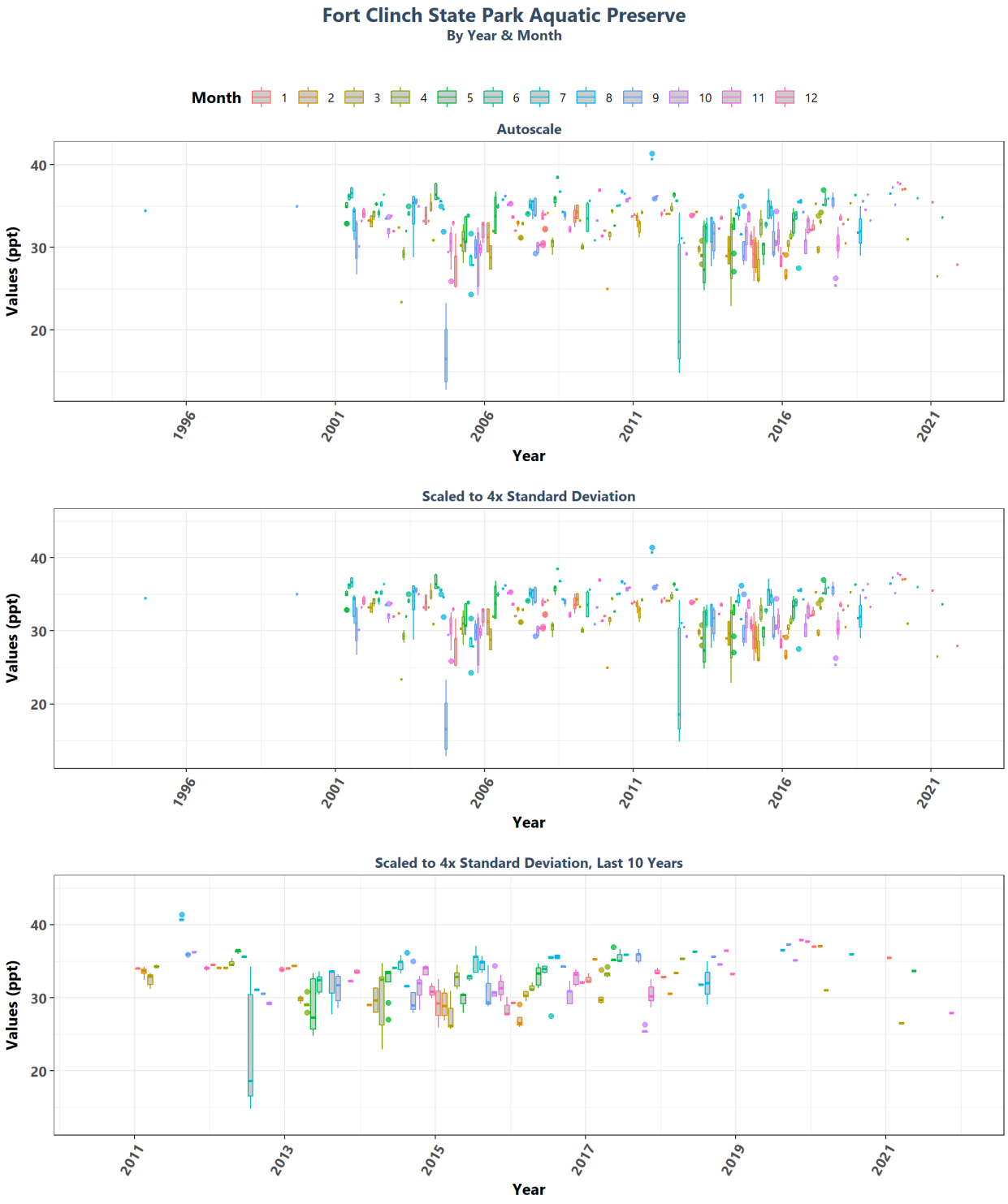
Fort Clinch State Park Aquatic Preserve

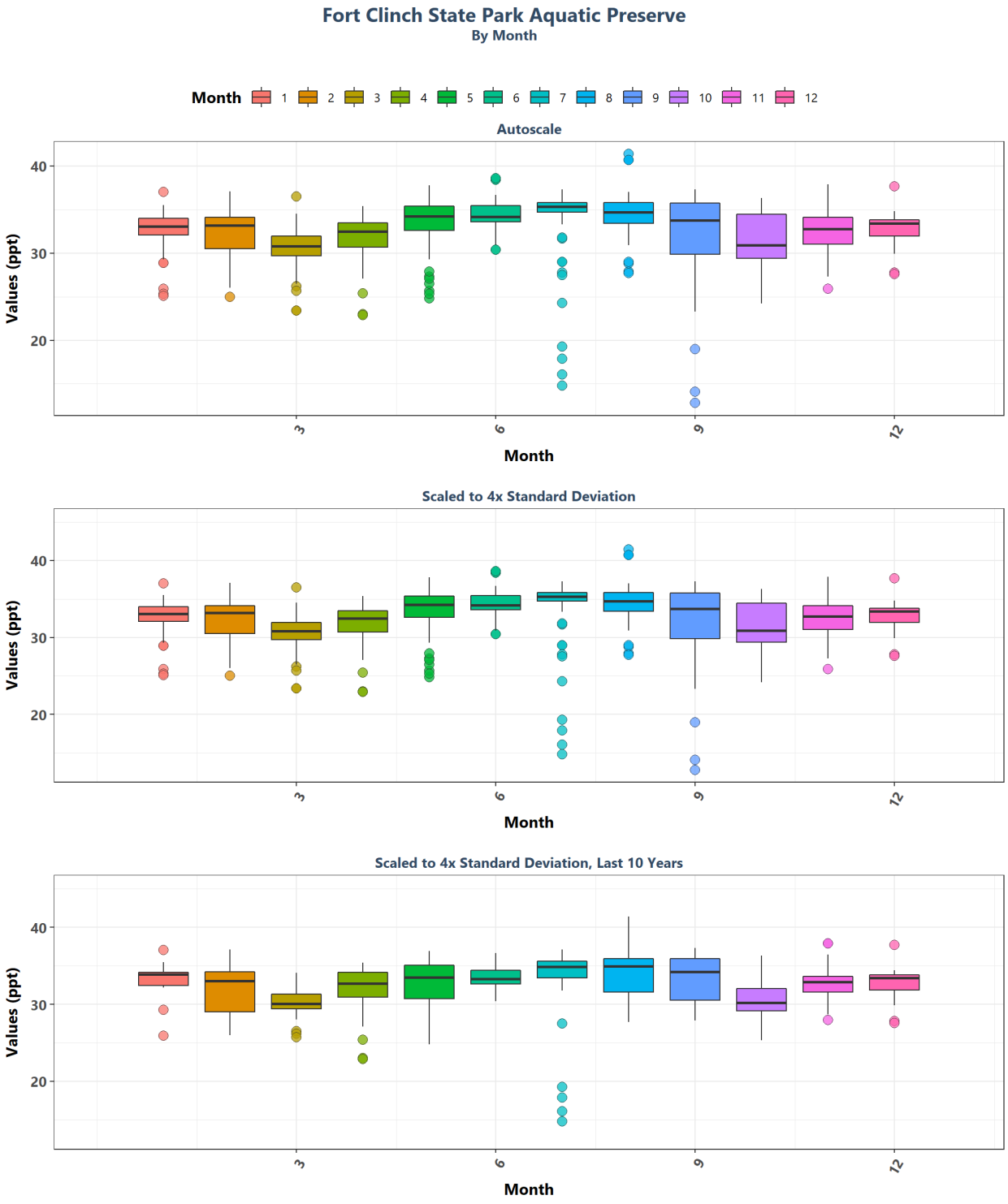
By Year





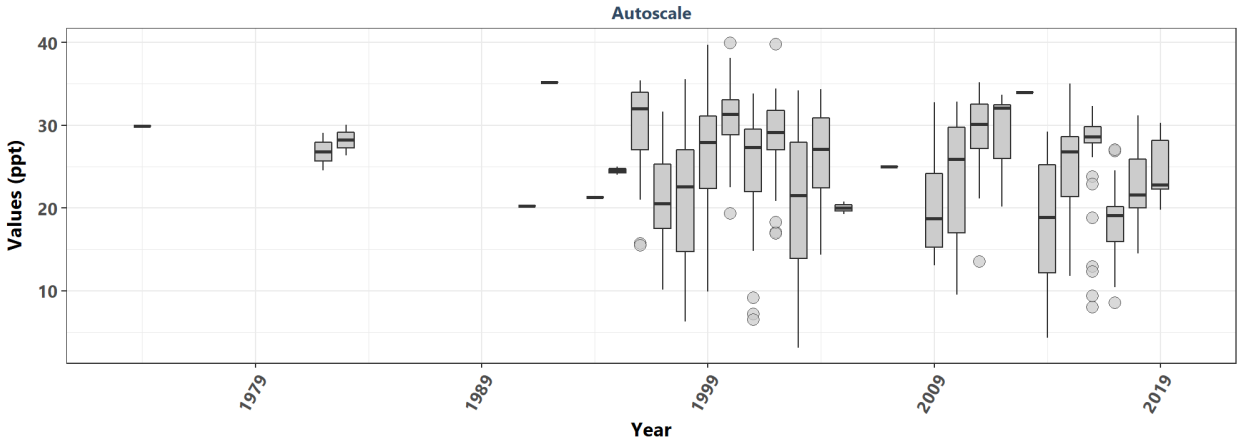


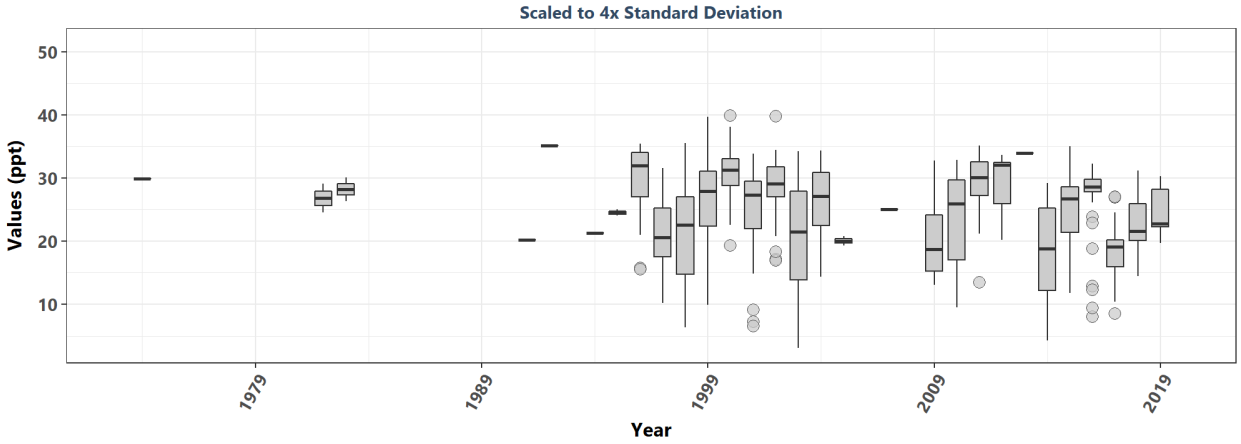

Fort Clinch State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
30
20
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
40
30
20
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
40
30
20
2011
2013
2015
2017
2019
2021
Year

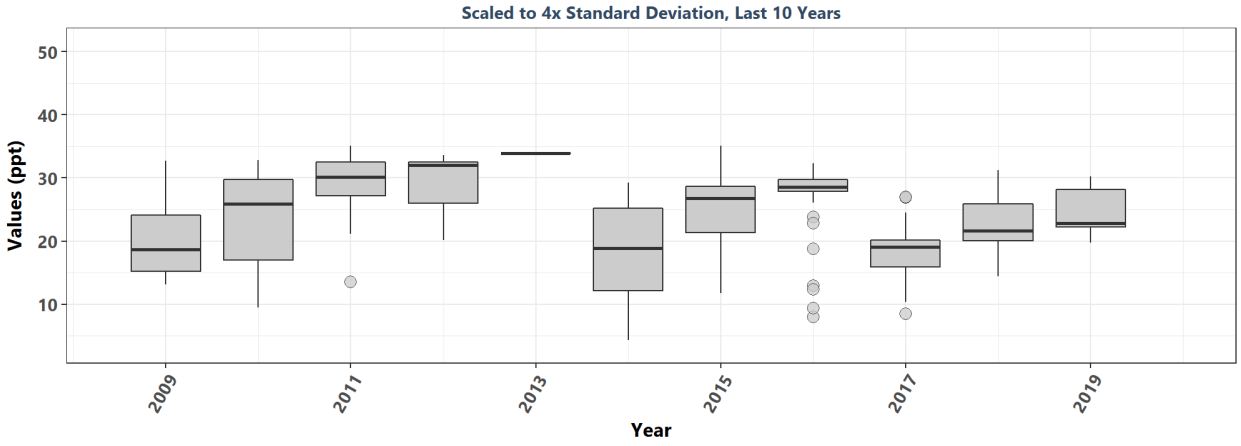

Fort Clinch State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
40
30
20
3
6
9
12
Month

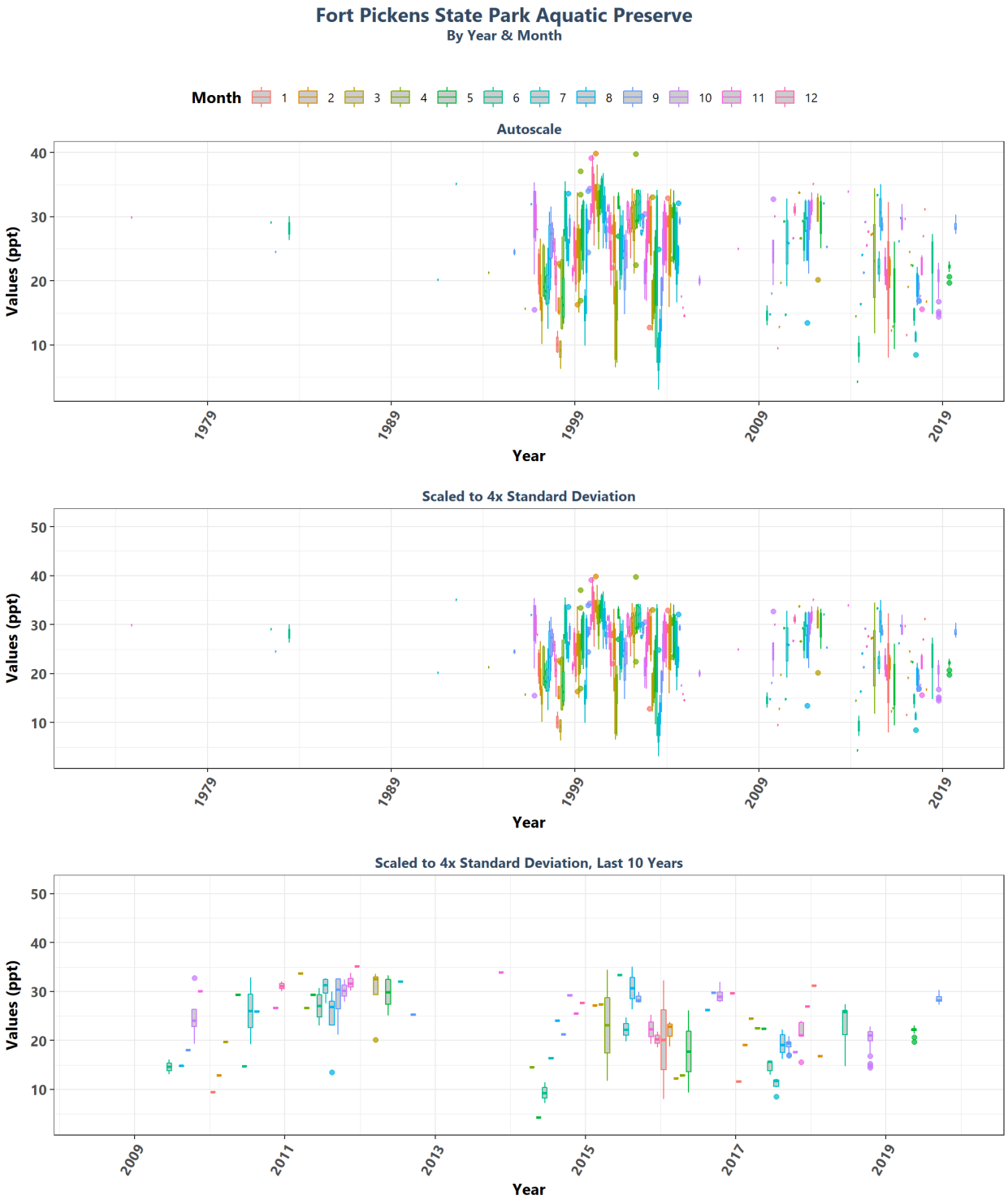
Fort Pickens State Park Aquatic Preserve

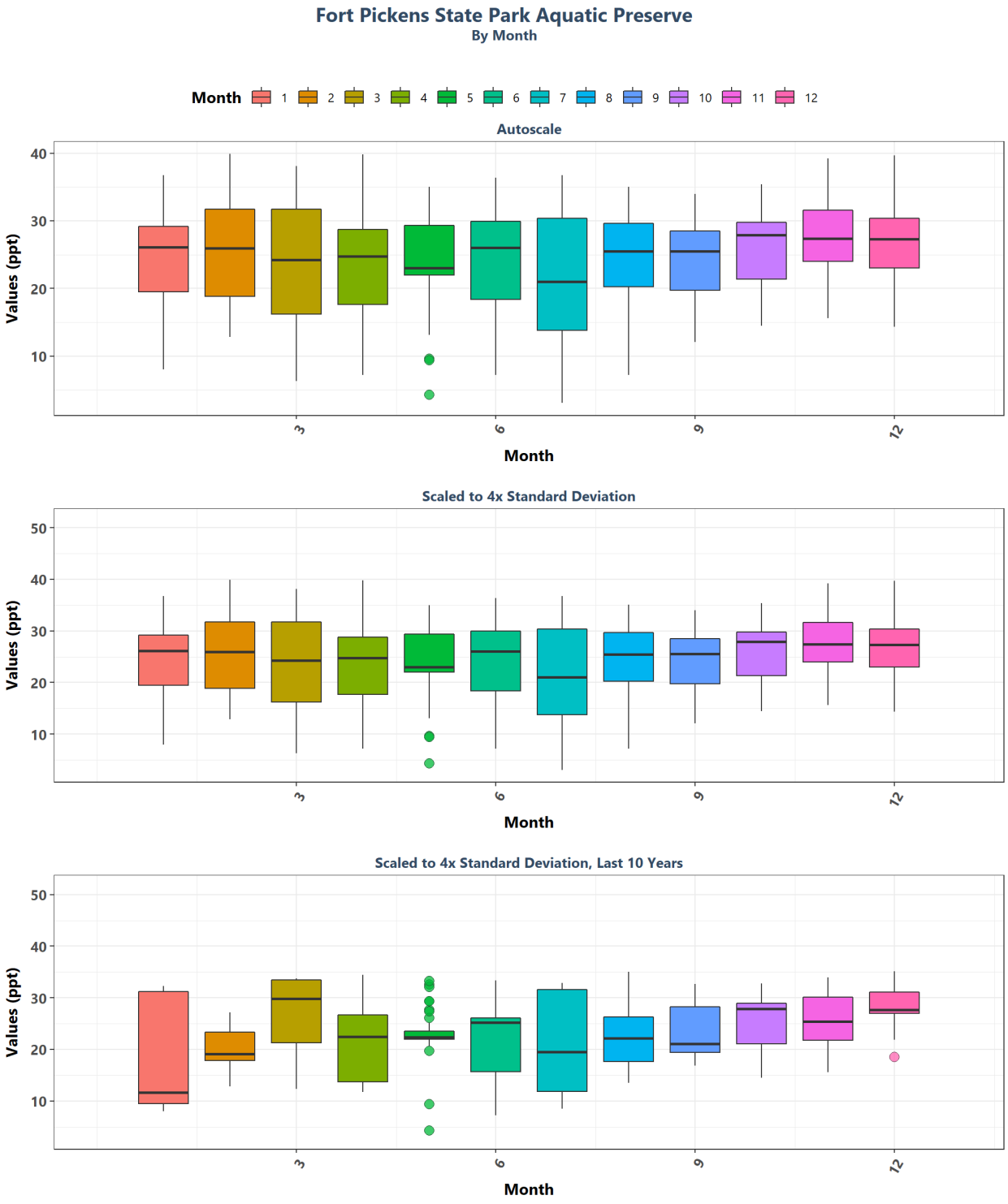
By Year





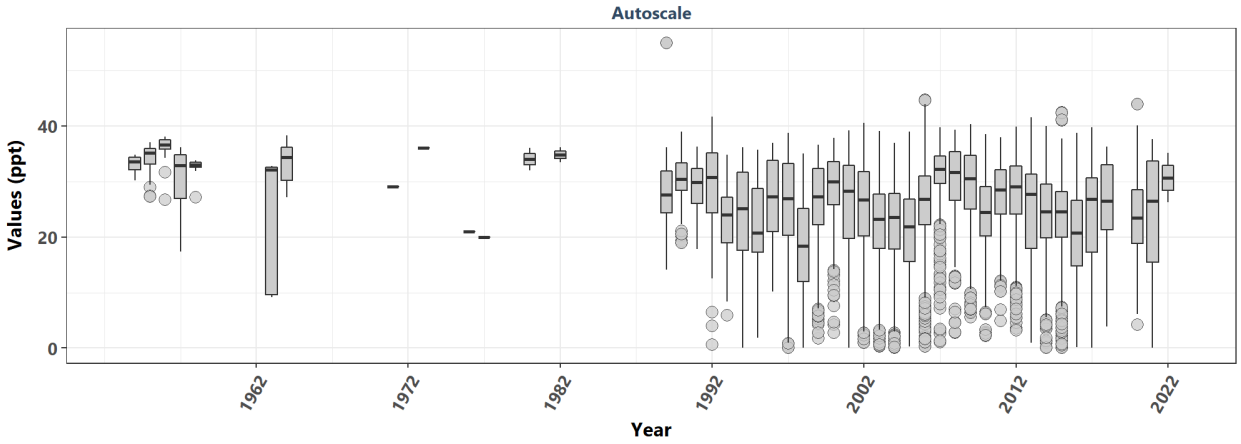


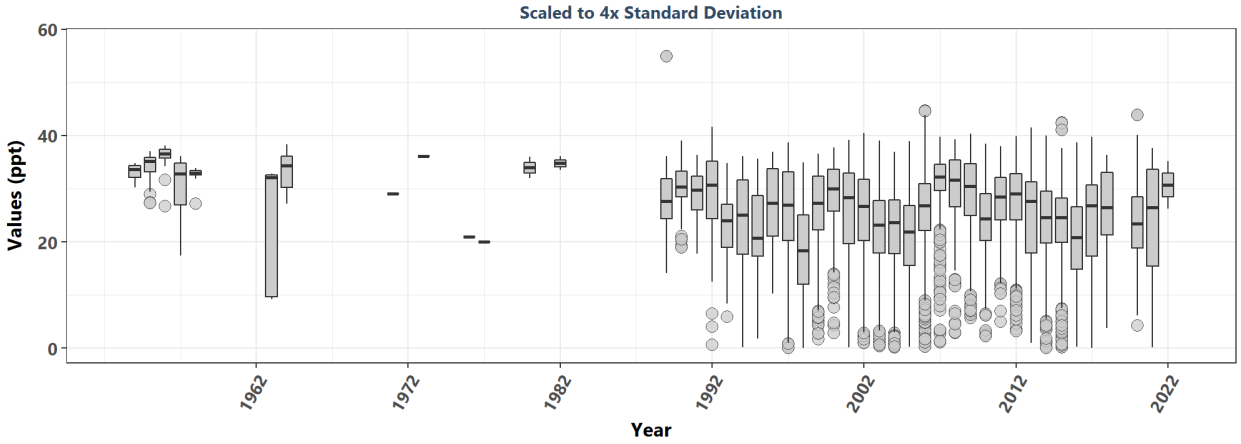

Fort Pickens State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

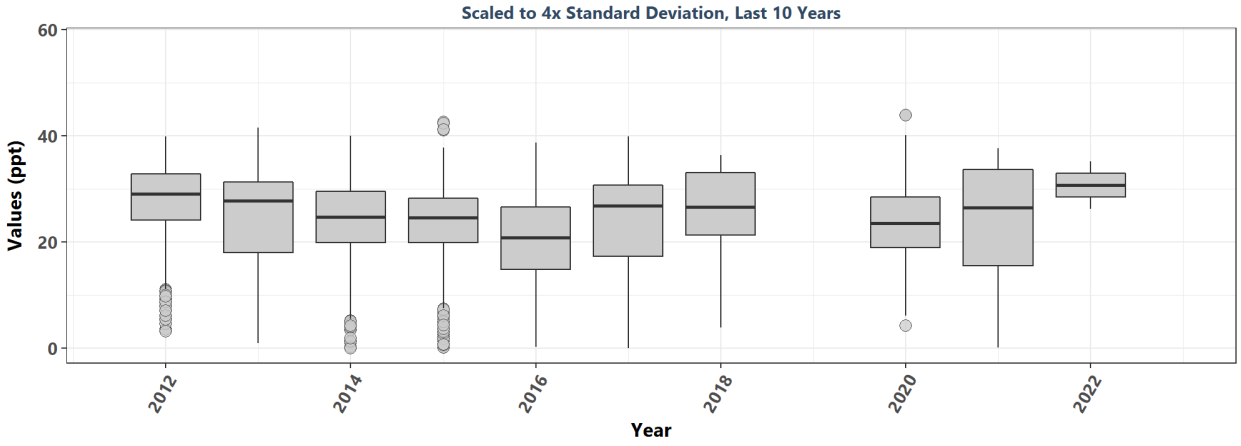

Fort Pickens State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
50
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
50
40
30
20
10
3
6
9
12
Month

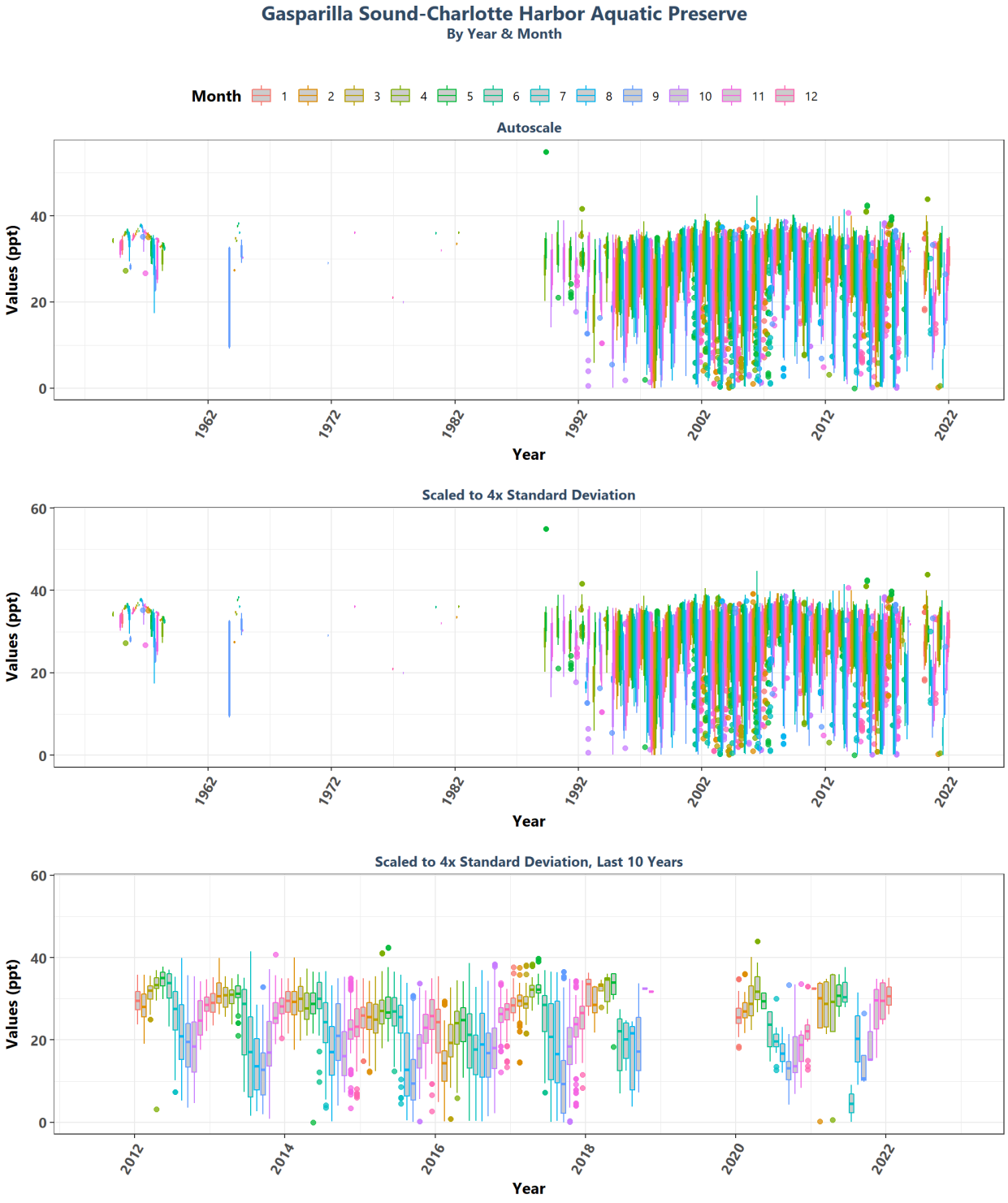
Gasparilla Sound-Charlotte Harbor Aquatic Preserve

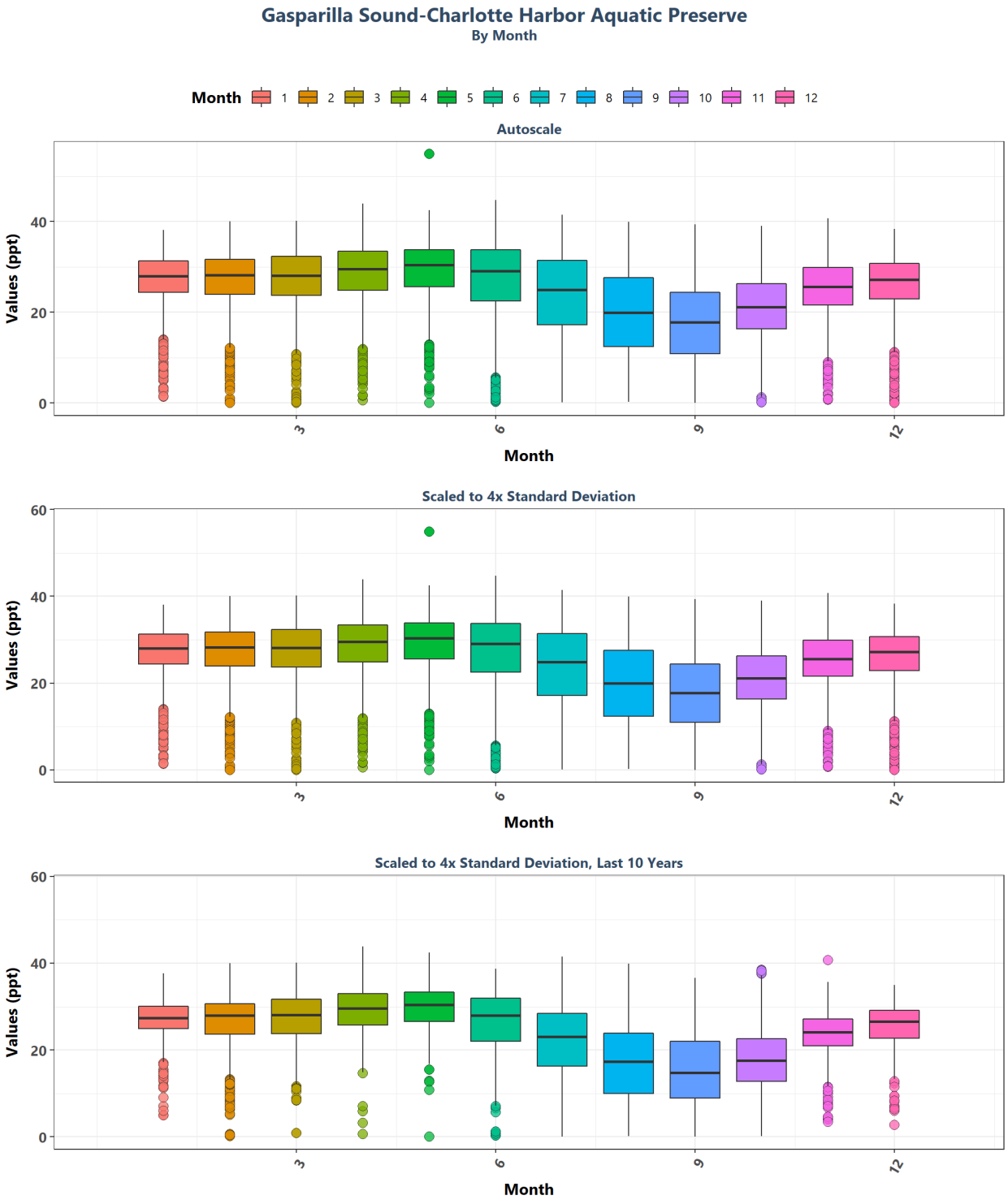
By Year





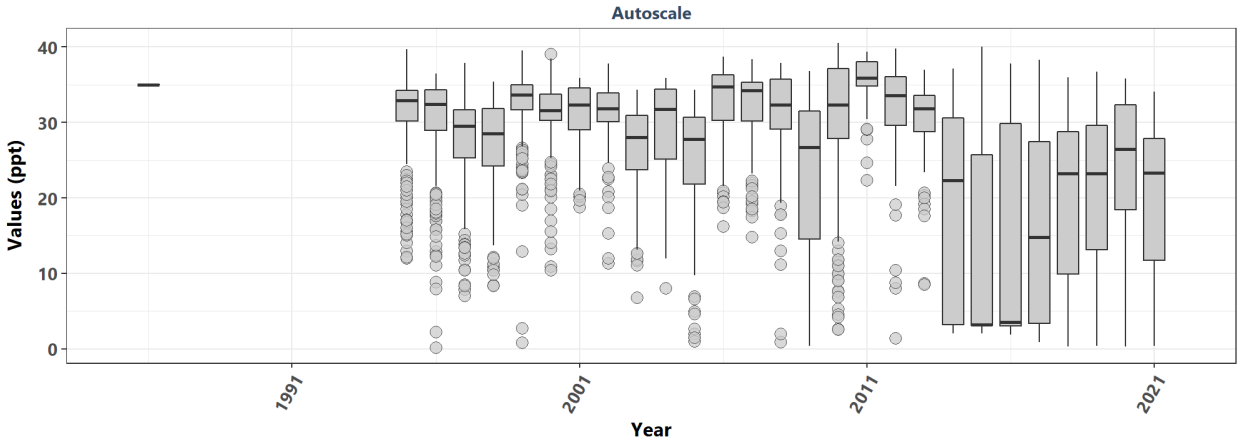


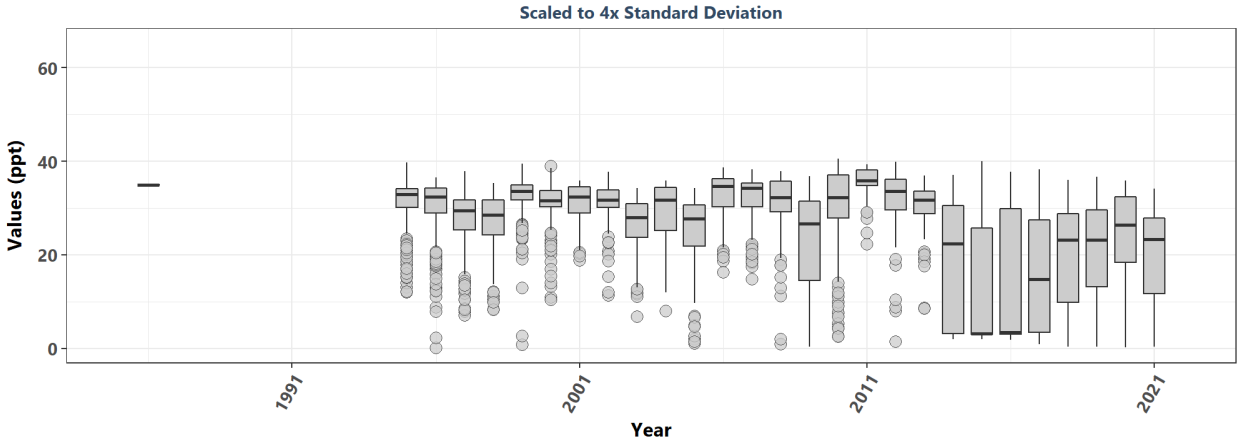

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

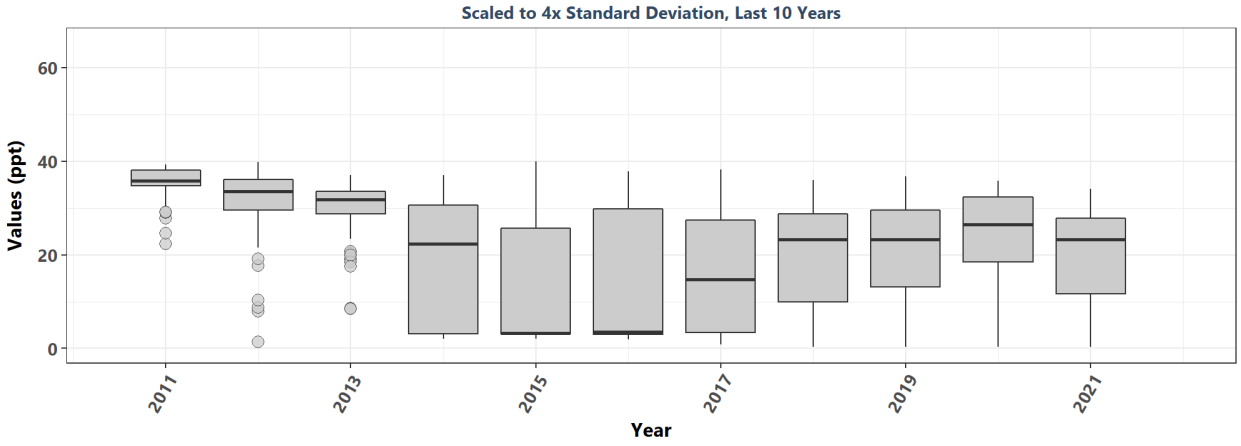

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
20
40
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
3
6
9
12
Month

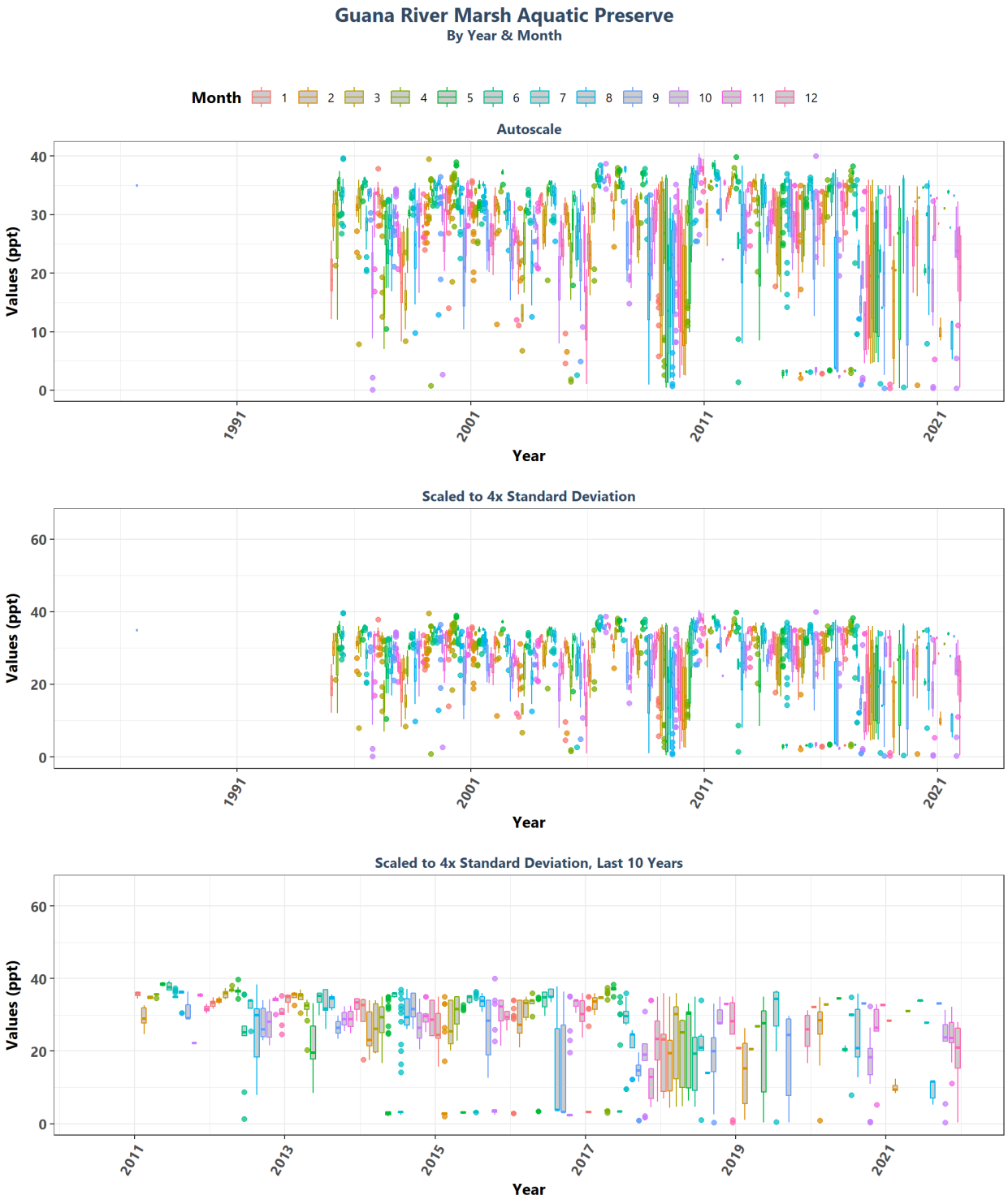
Guana River Marsh Aquatic Preserve

By Year

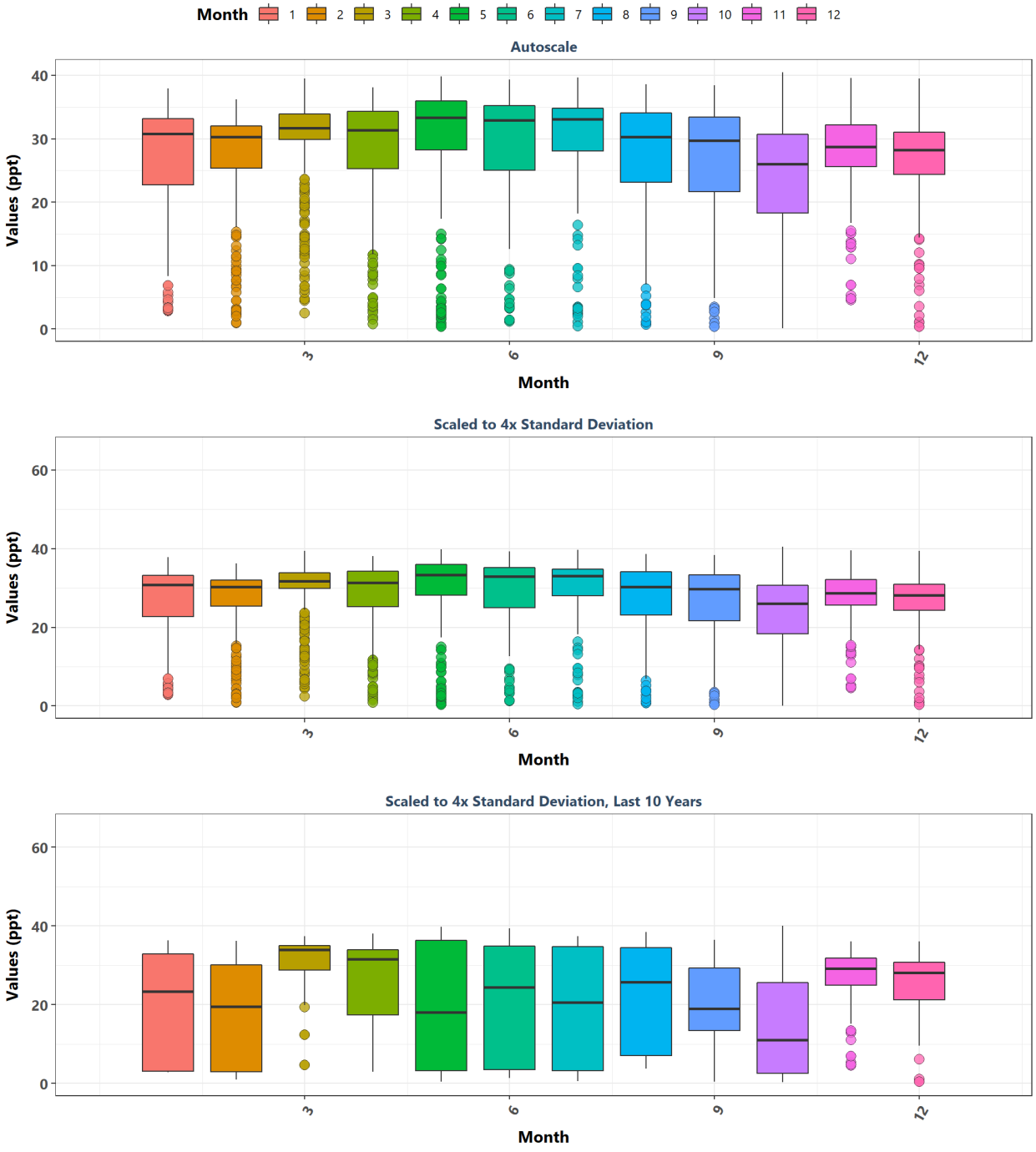






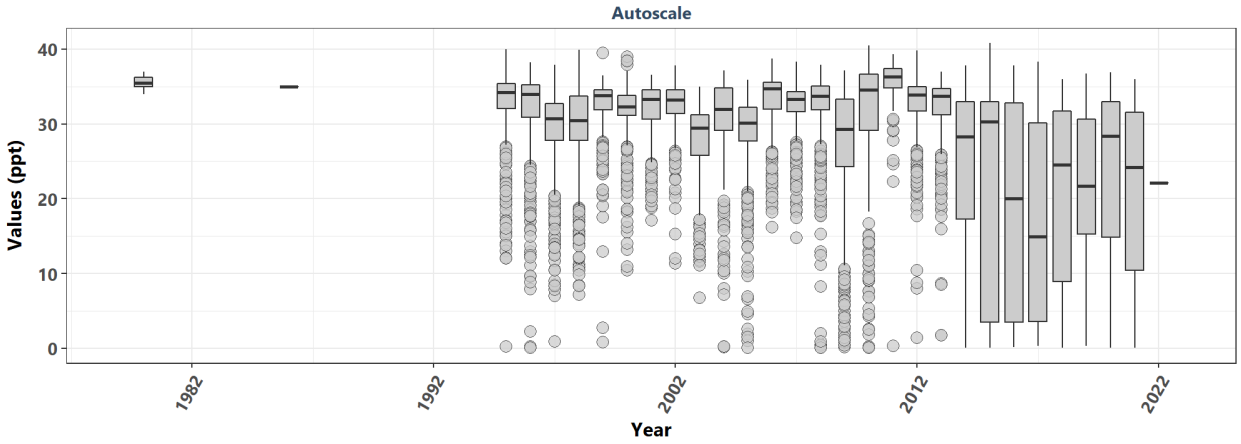

Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

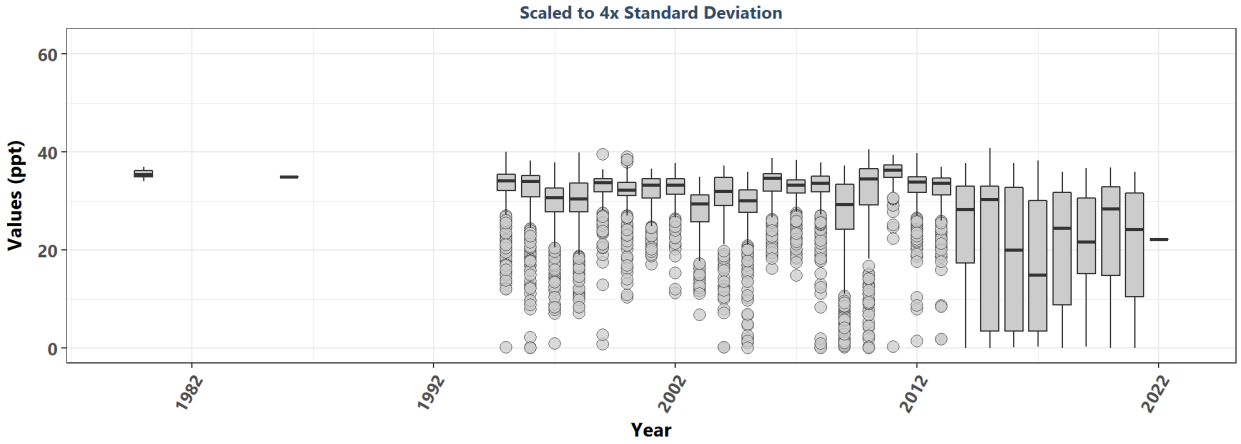
Guana River Marsh Aquatic Preserve
By Month

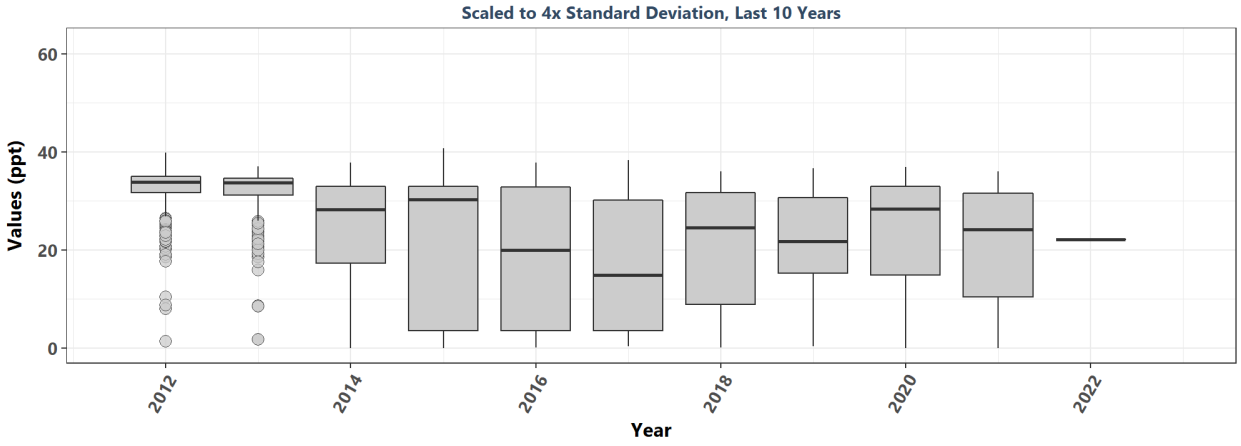


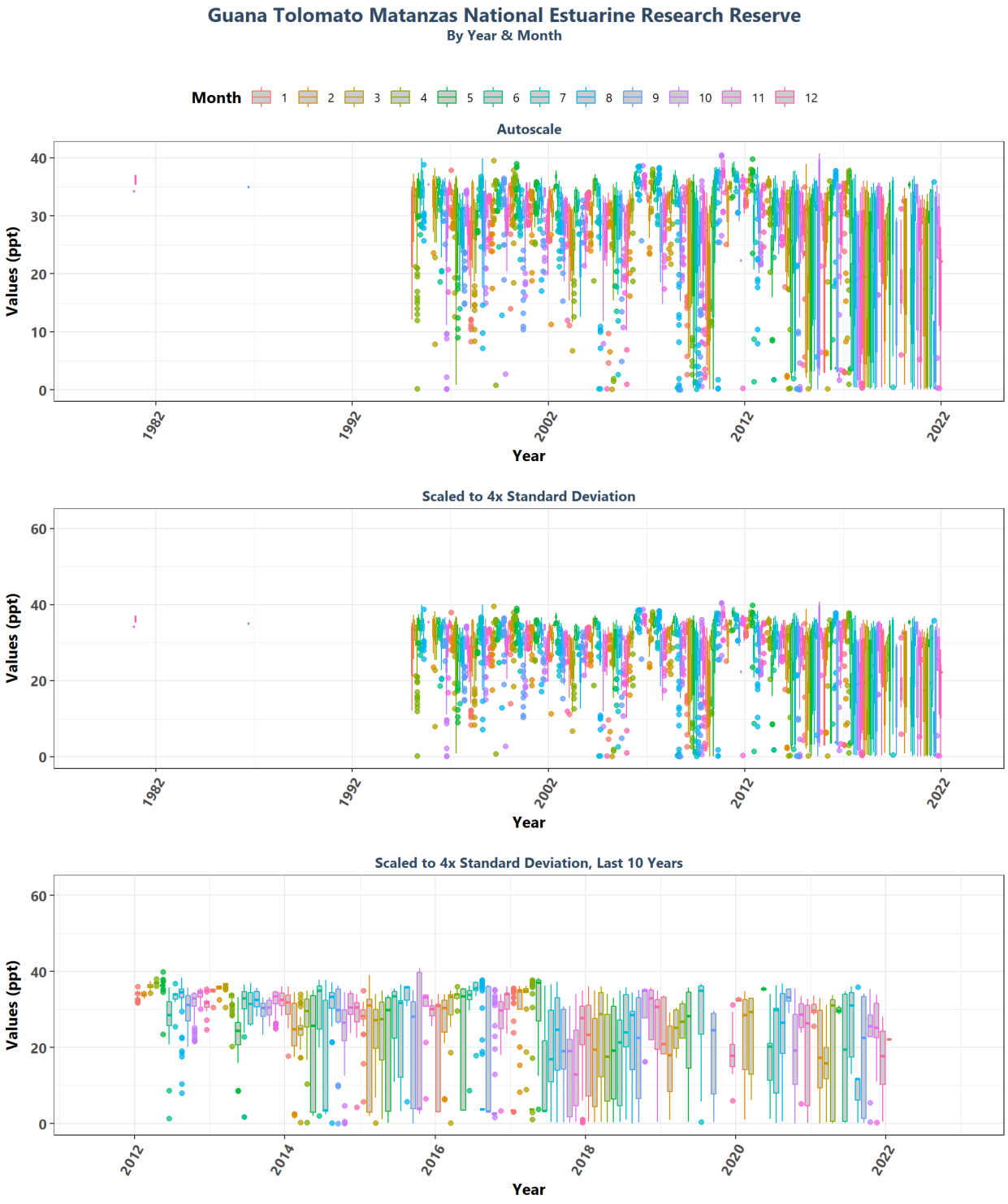
Guana Tolomato Matanzas National Estuarine Research Reserve

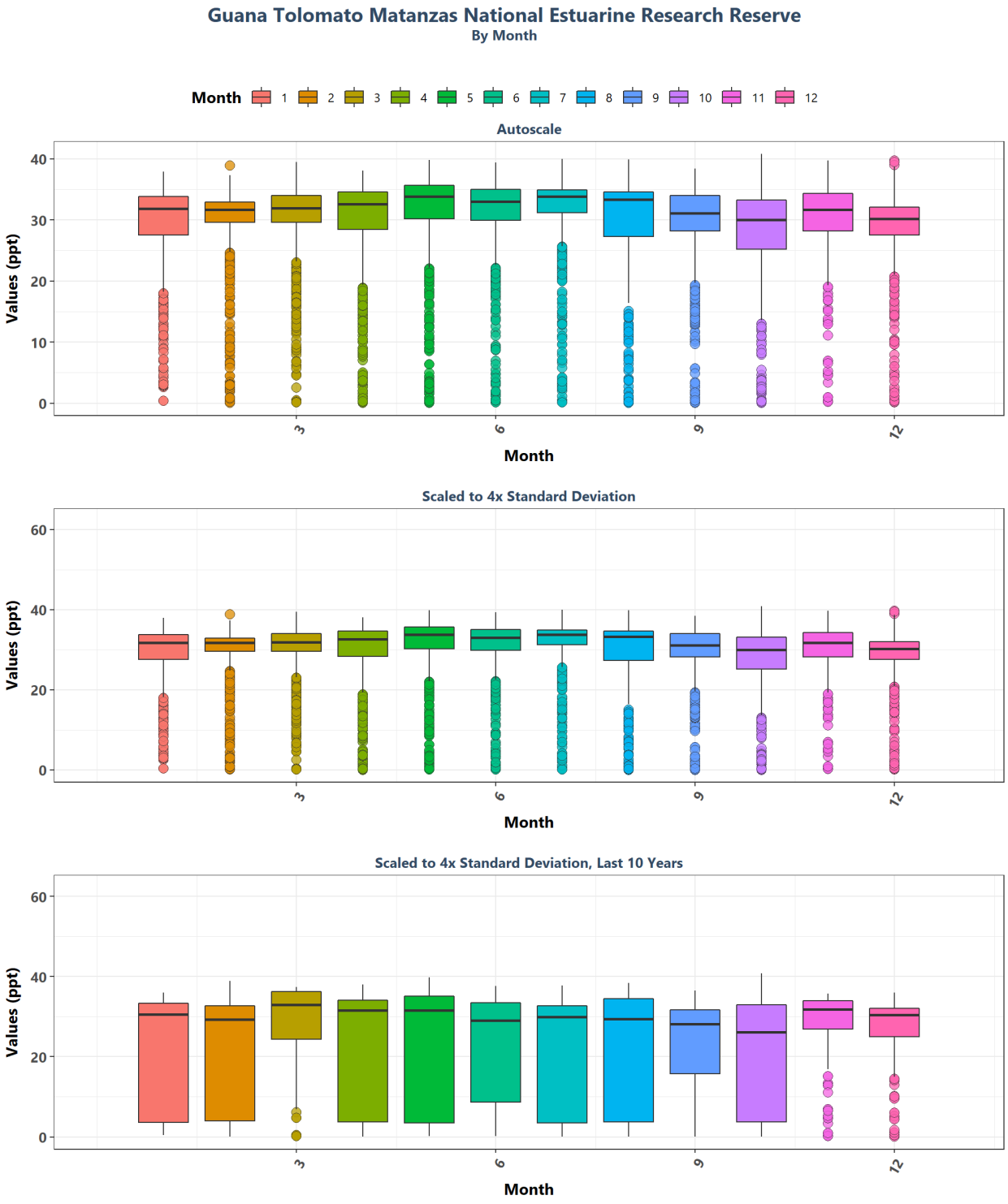
By Year





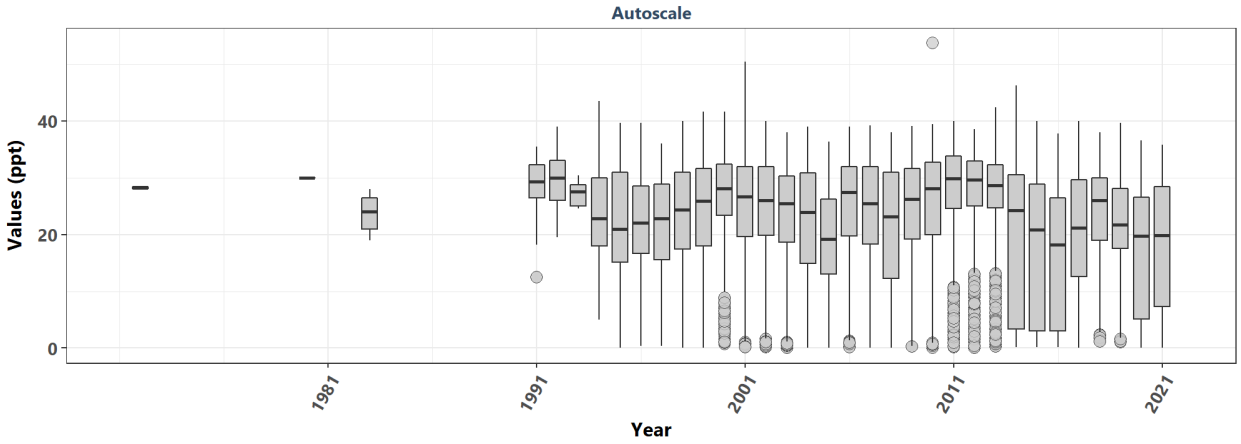


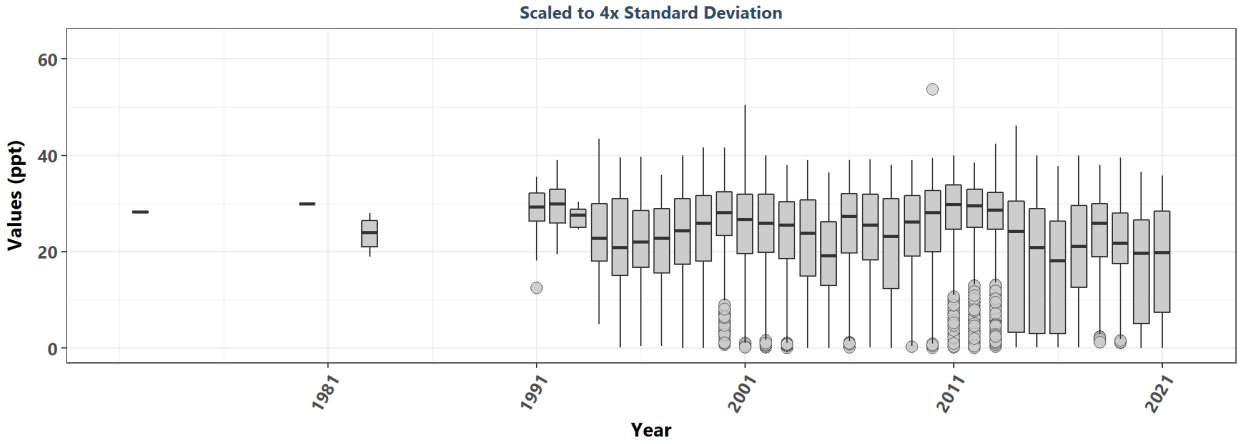

Guana Tolomato Matanzas National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

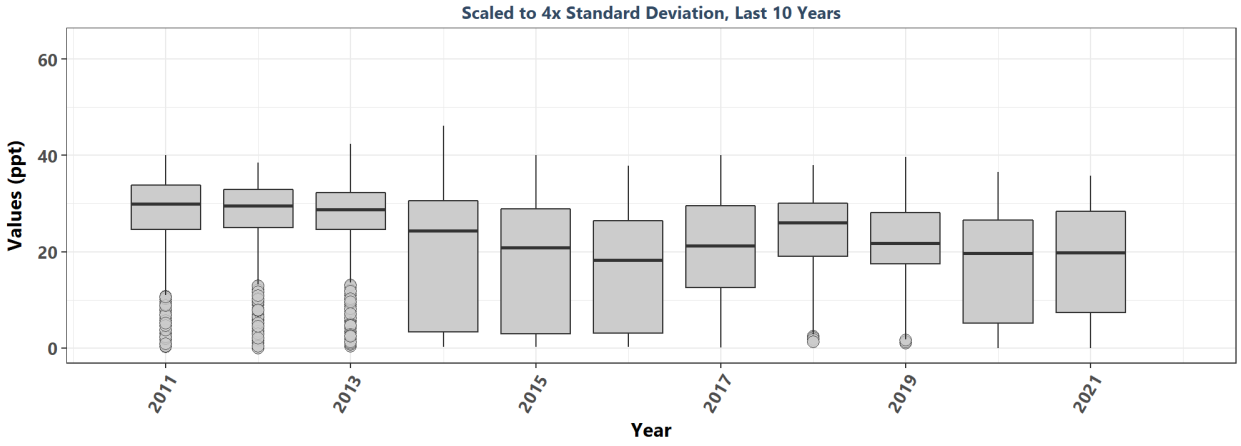

Guana Tolomato Matanzas National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

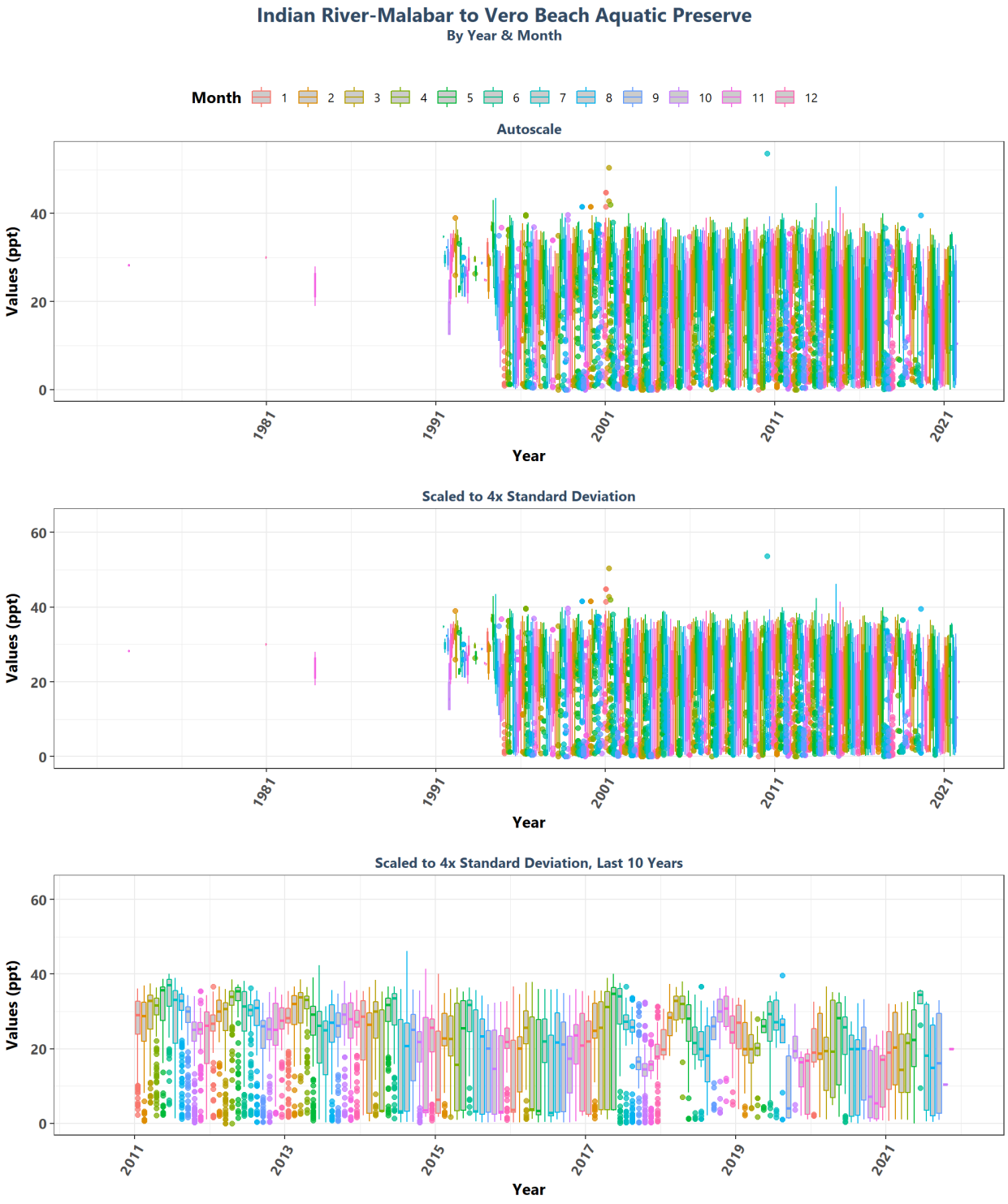
Indian River-Malabar to Vero Beach Aquatic Preserve

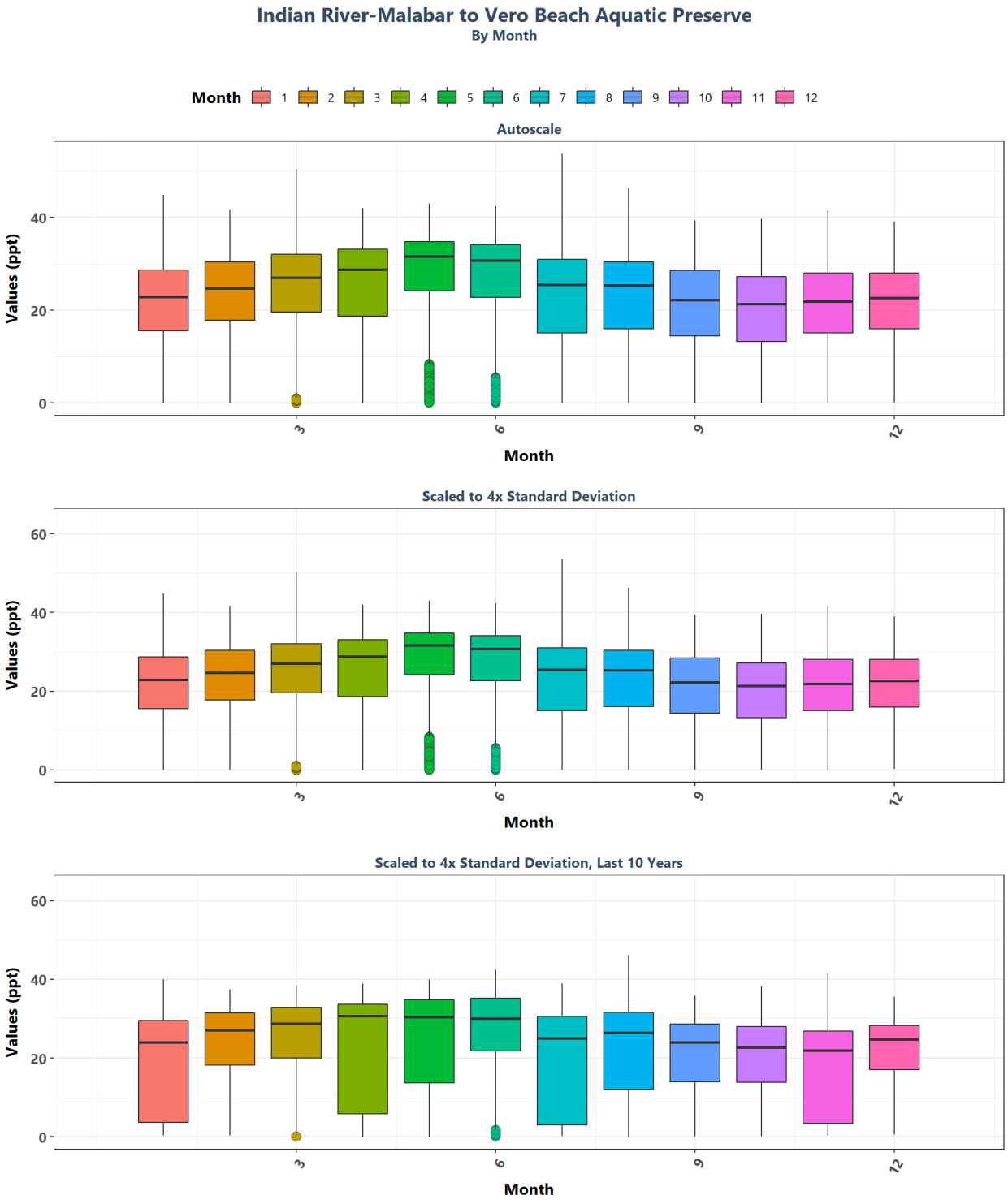
By Year





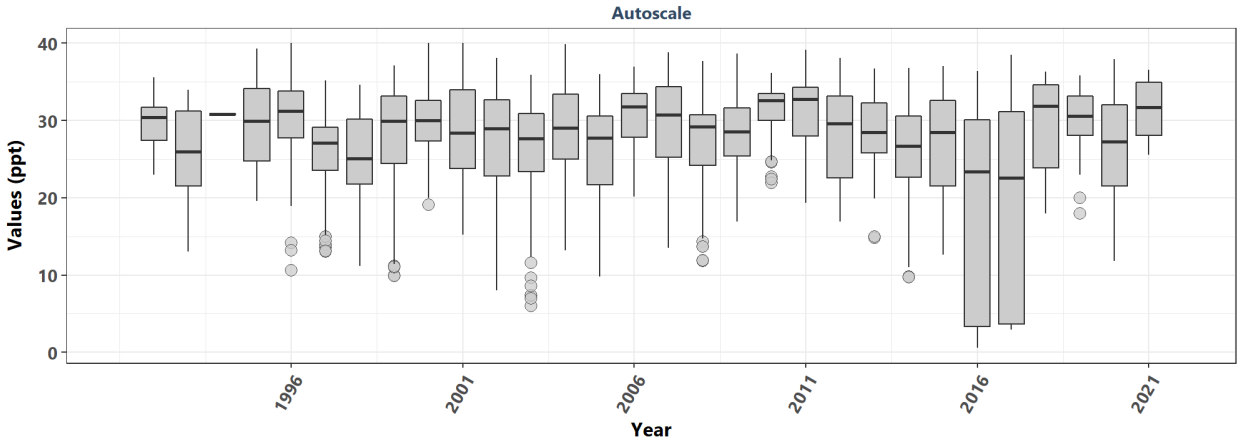


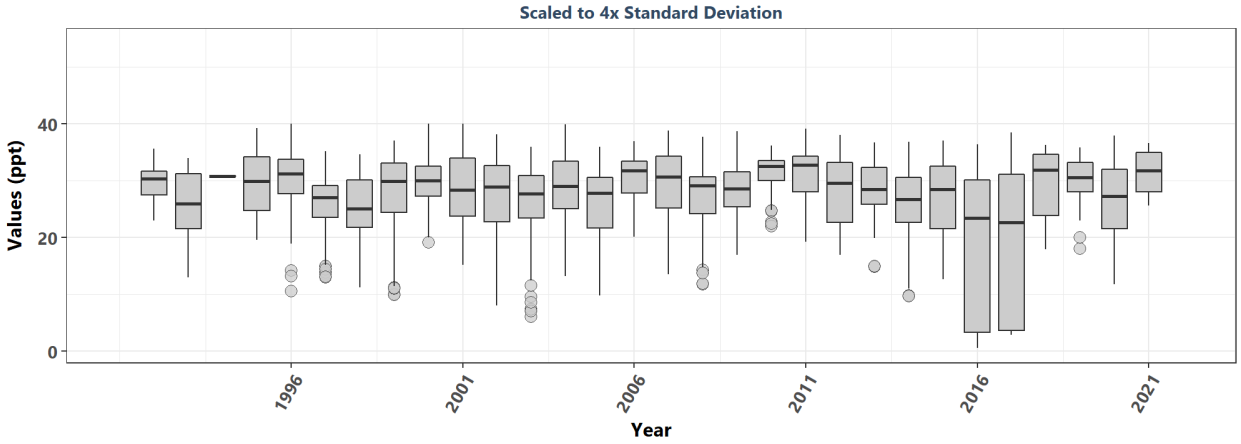

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
20
40
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
2011
2013
2015
2017
2019
2021
Year

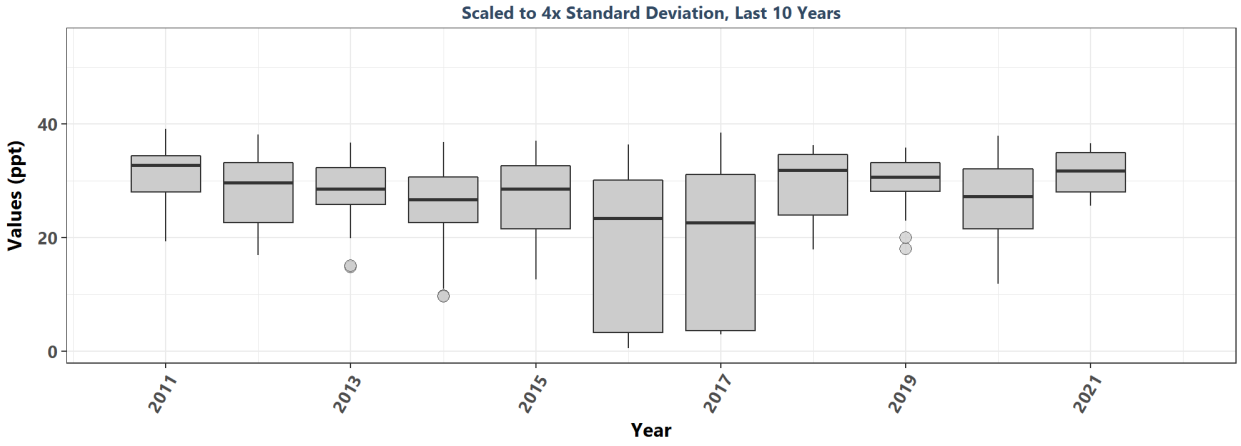

Indian River-Malabar to Vero Beach Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
20
40
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
3
6
9
12
Month

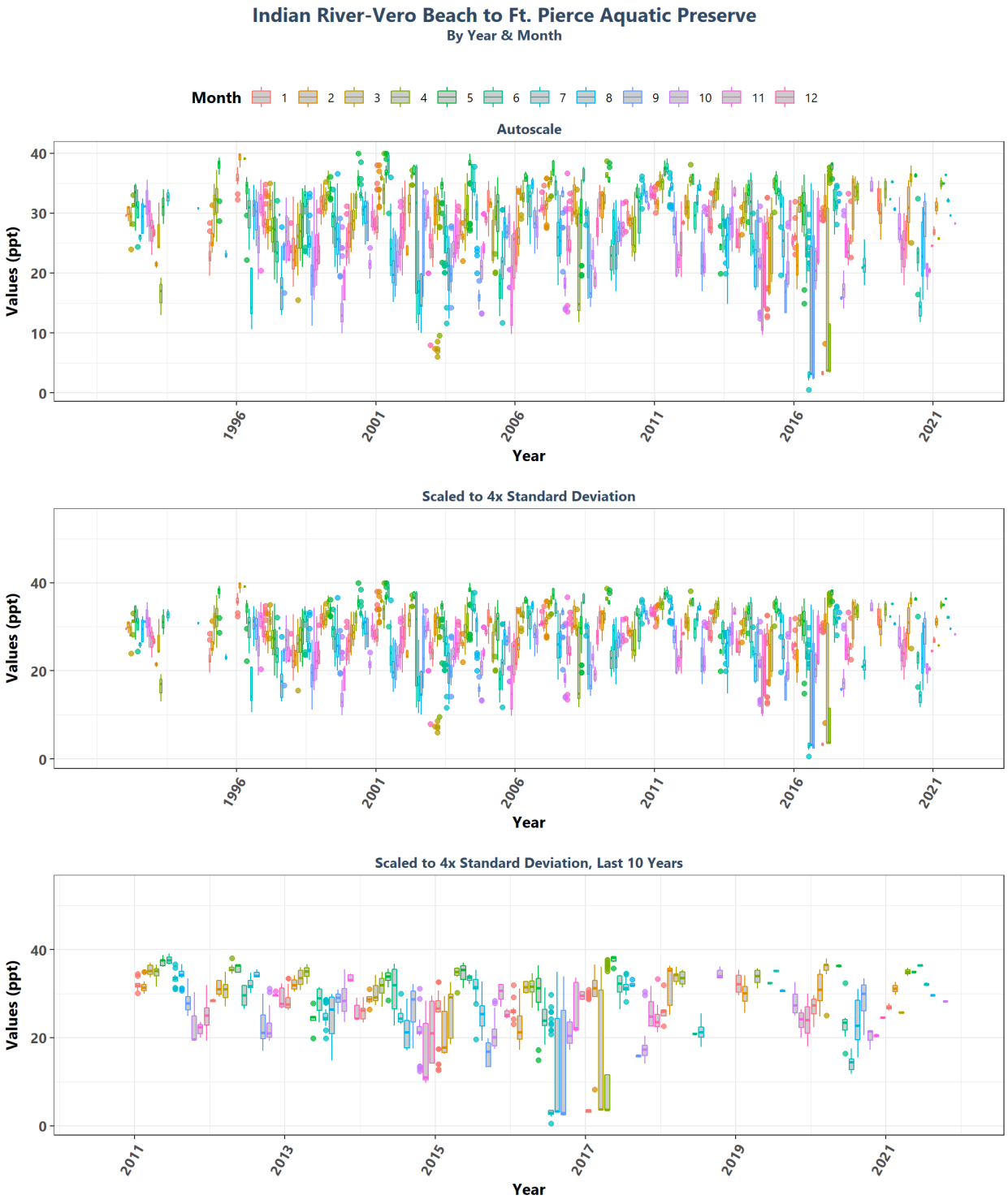
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

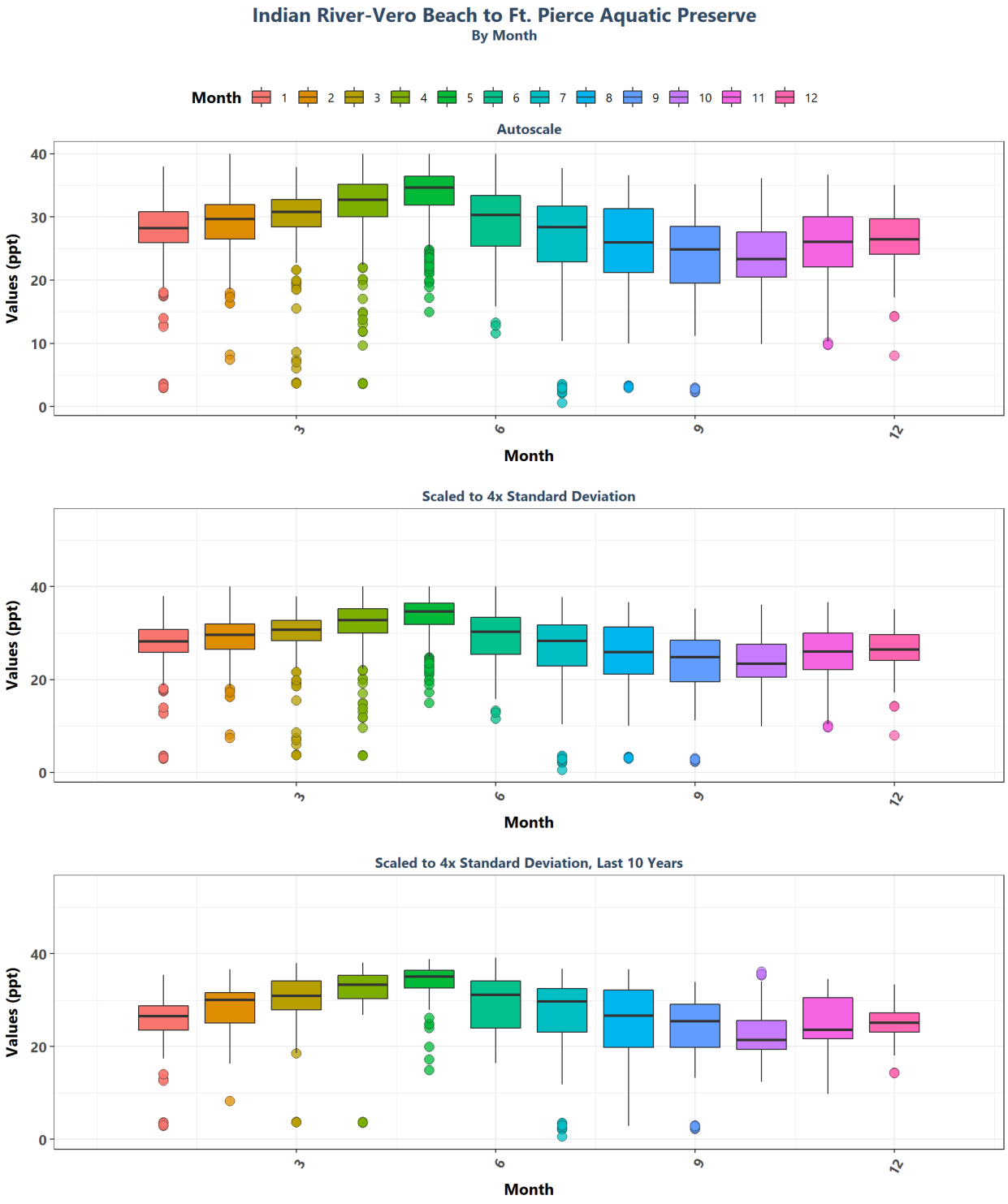
By Year





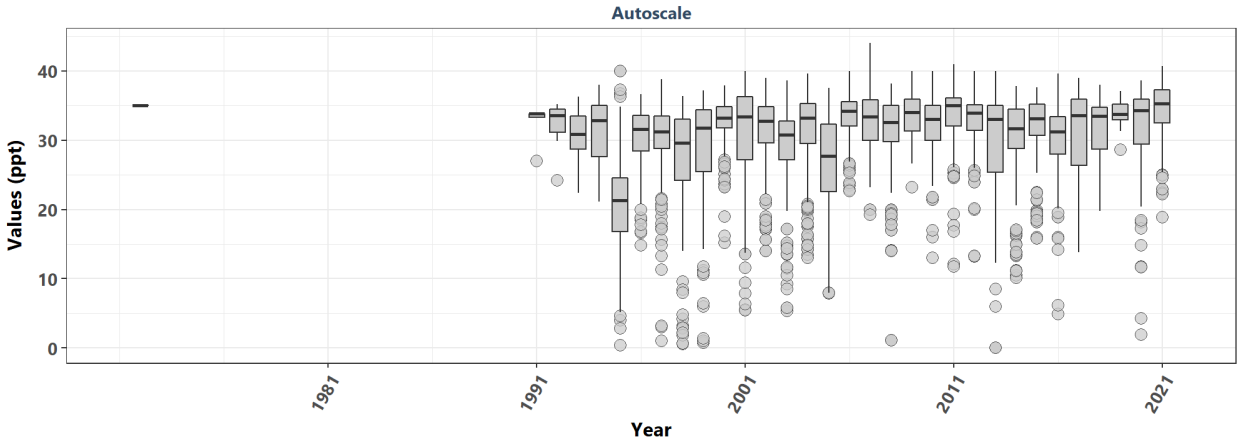


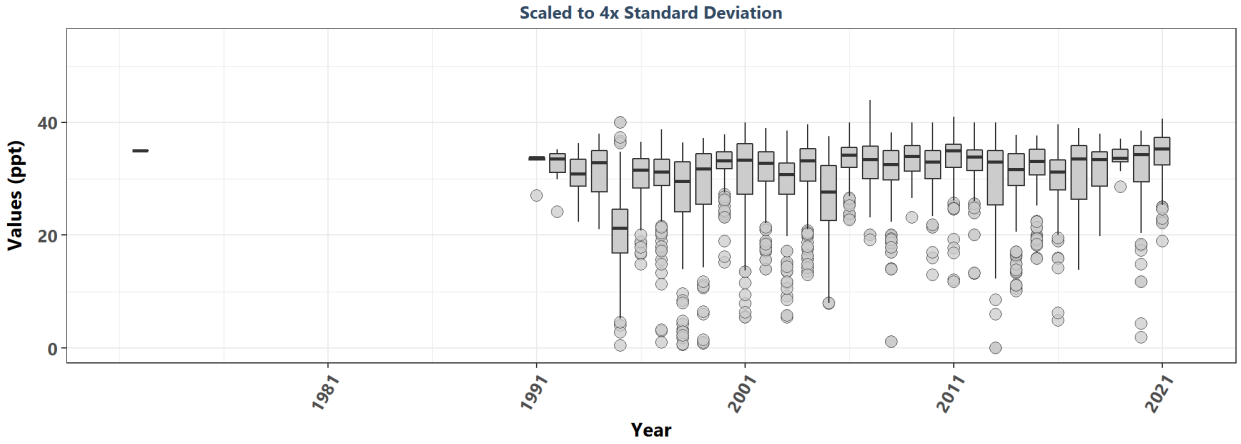

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

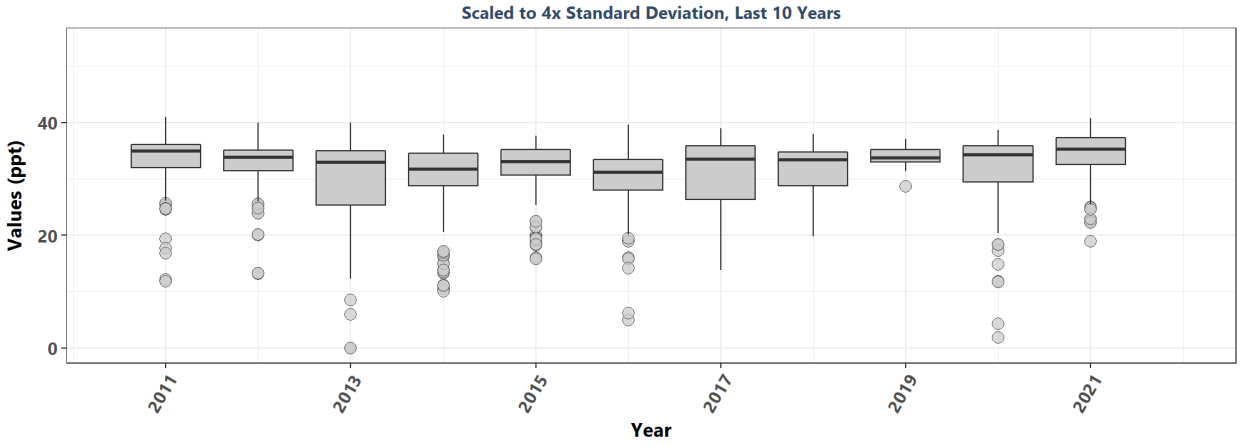

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

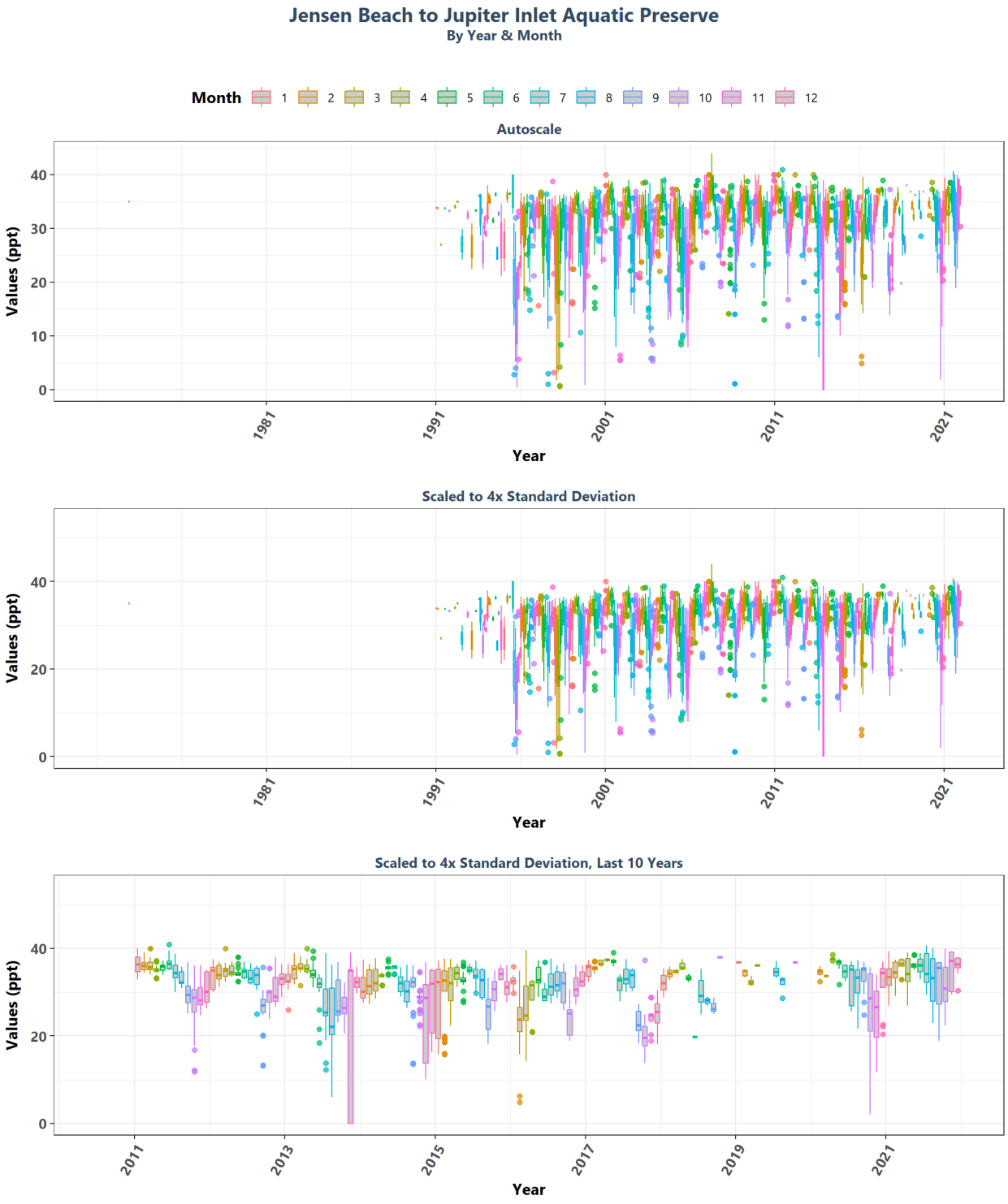
Jensen Beach to Jupiter Inlet Aquatic Preserve

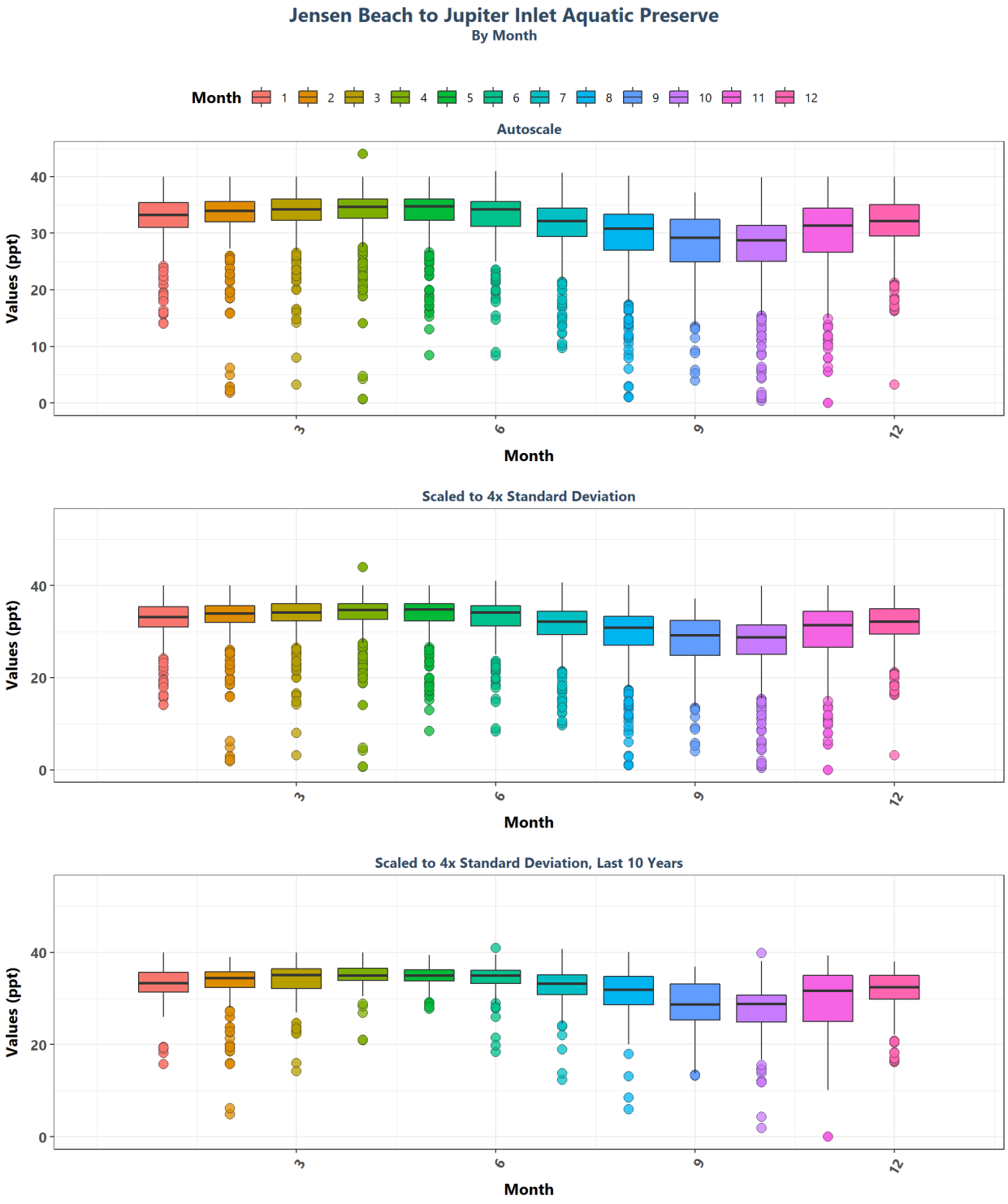
By Year

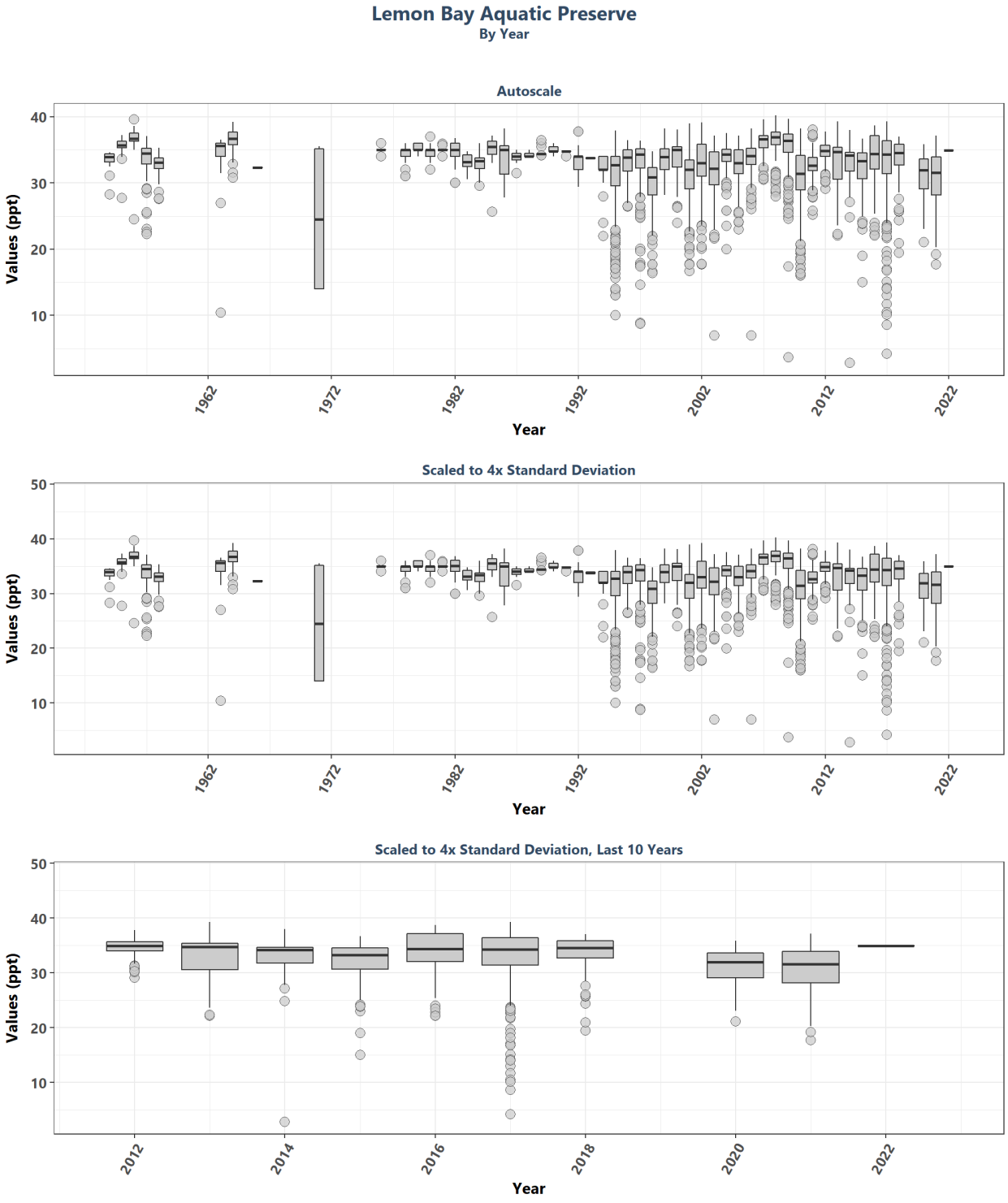


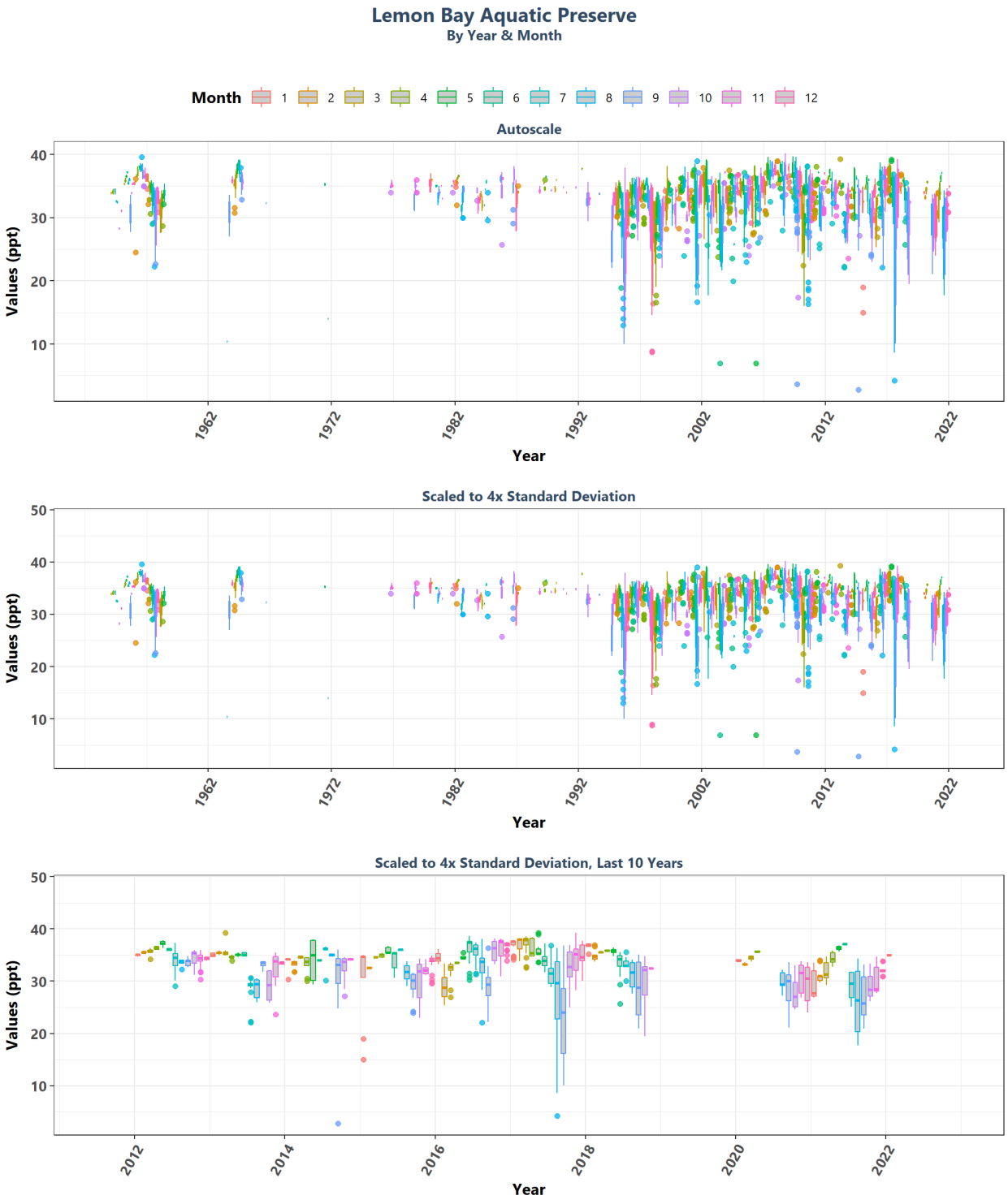


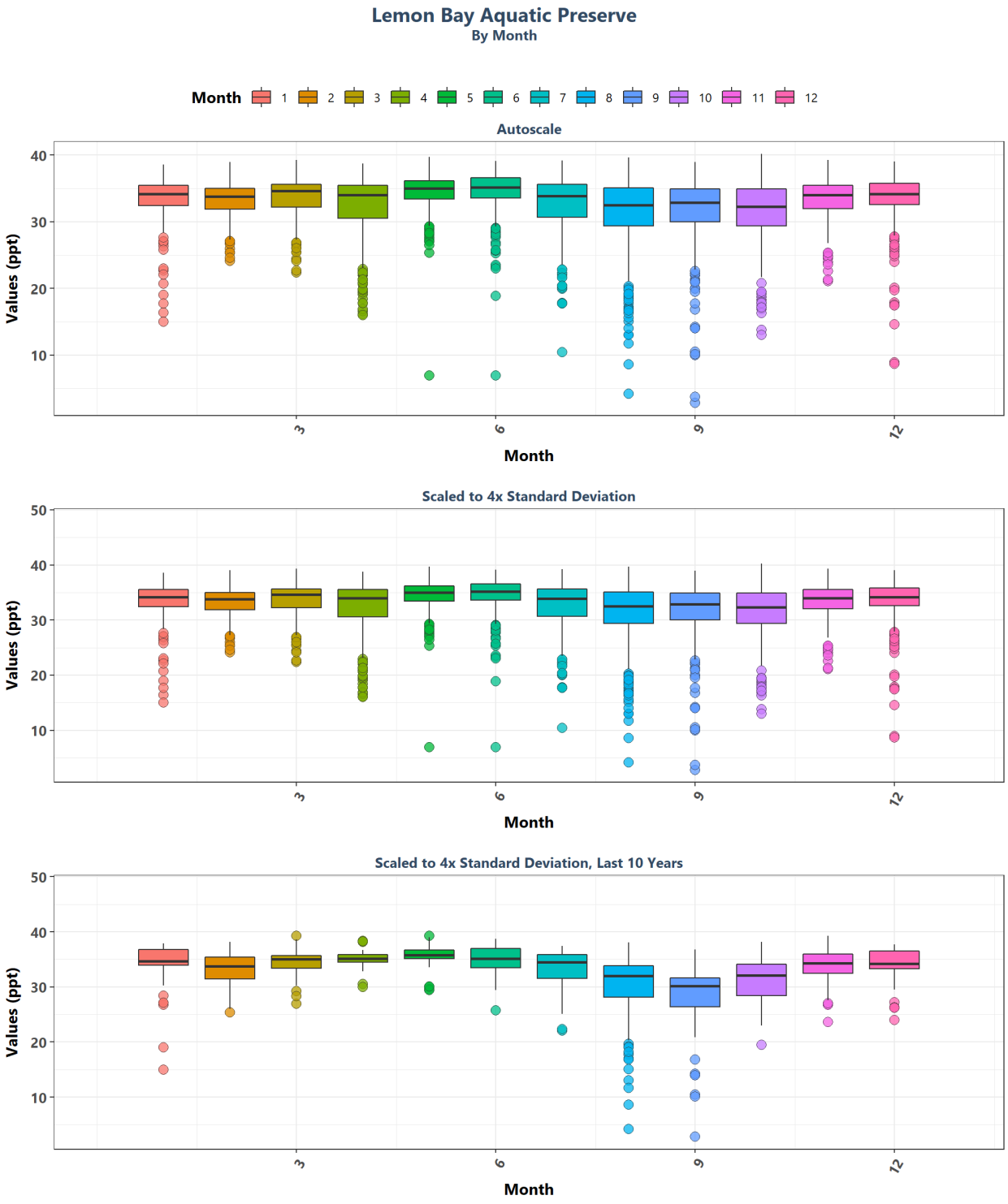


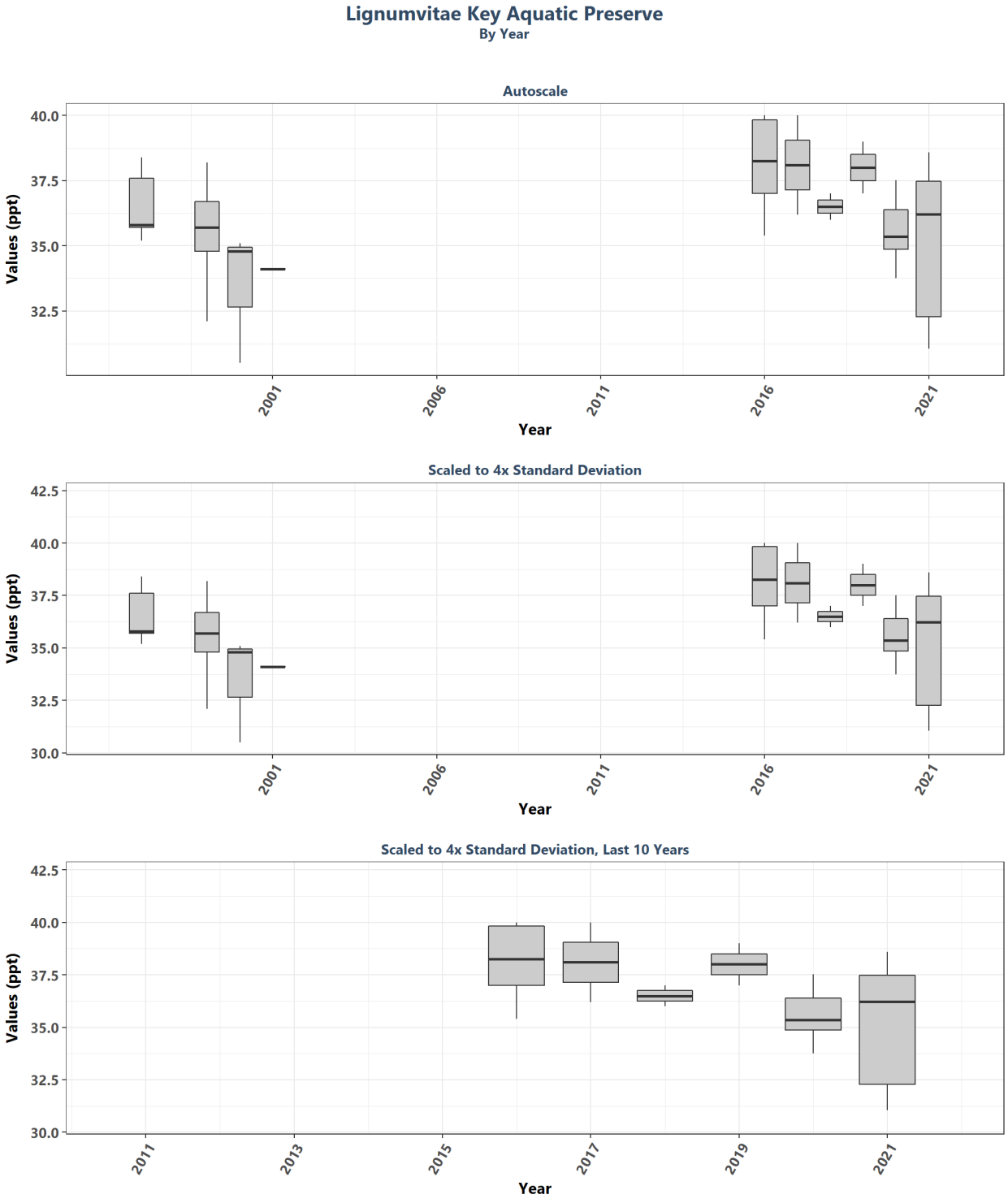

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

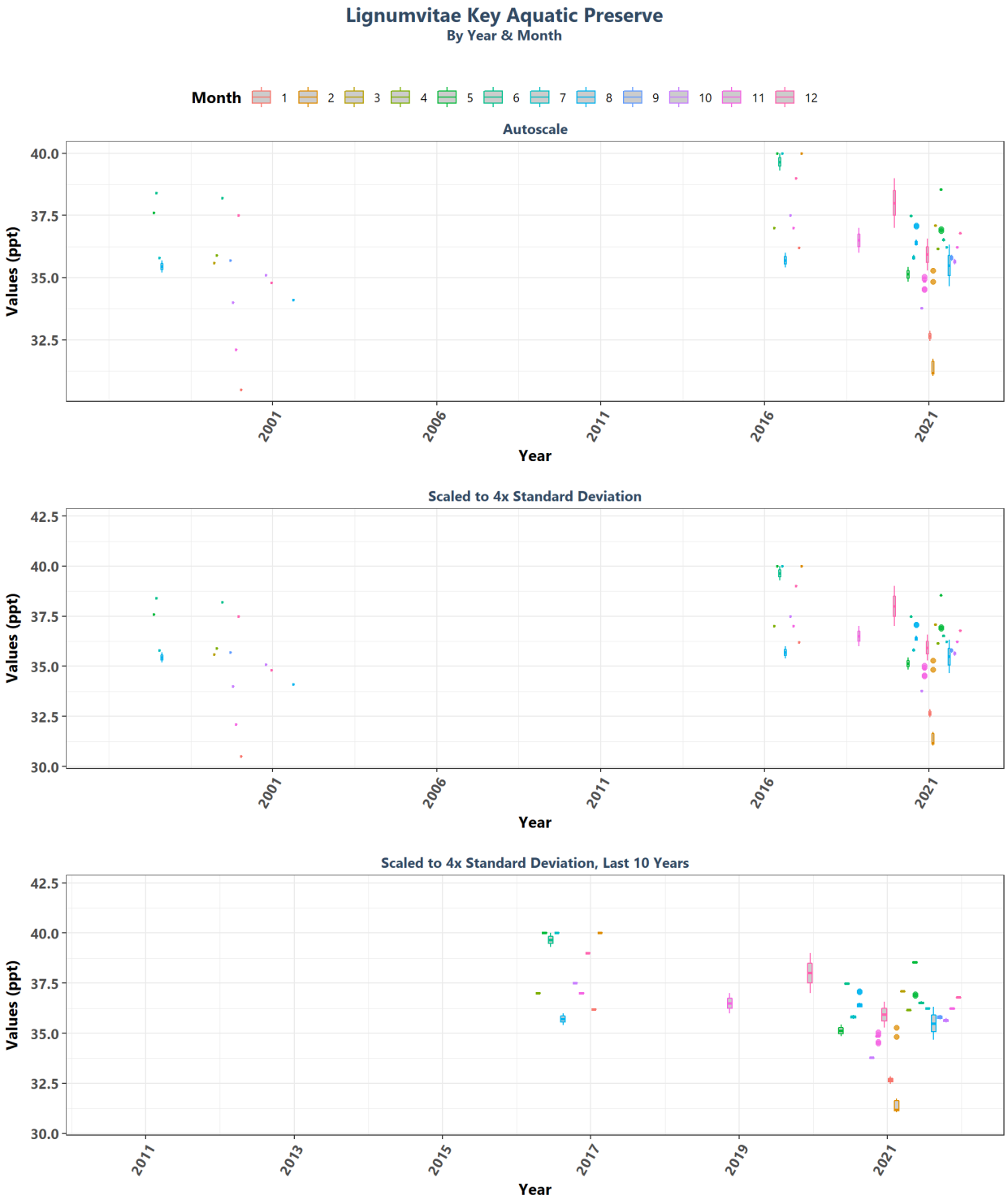

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

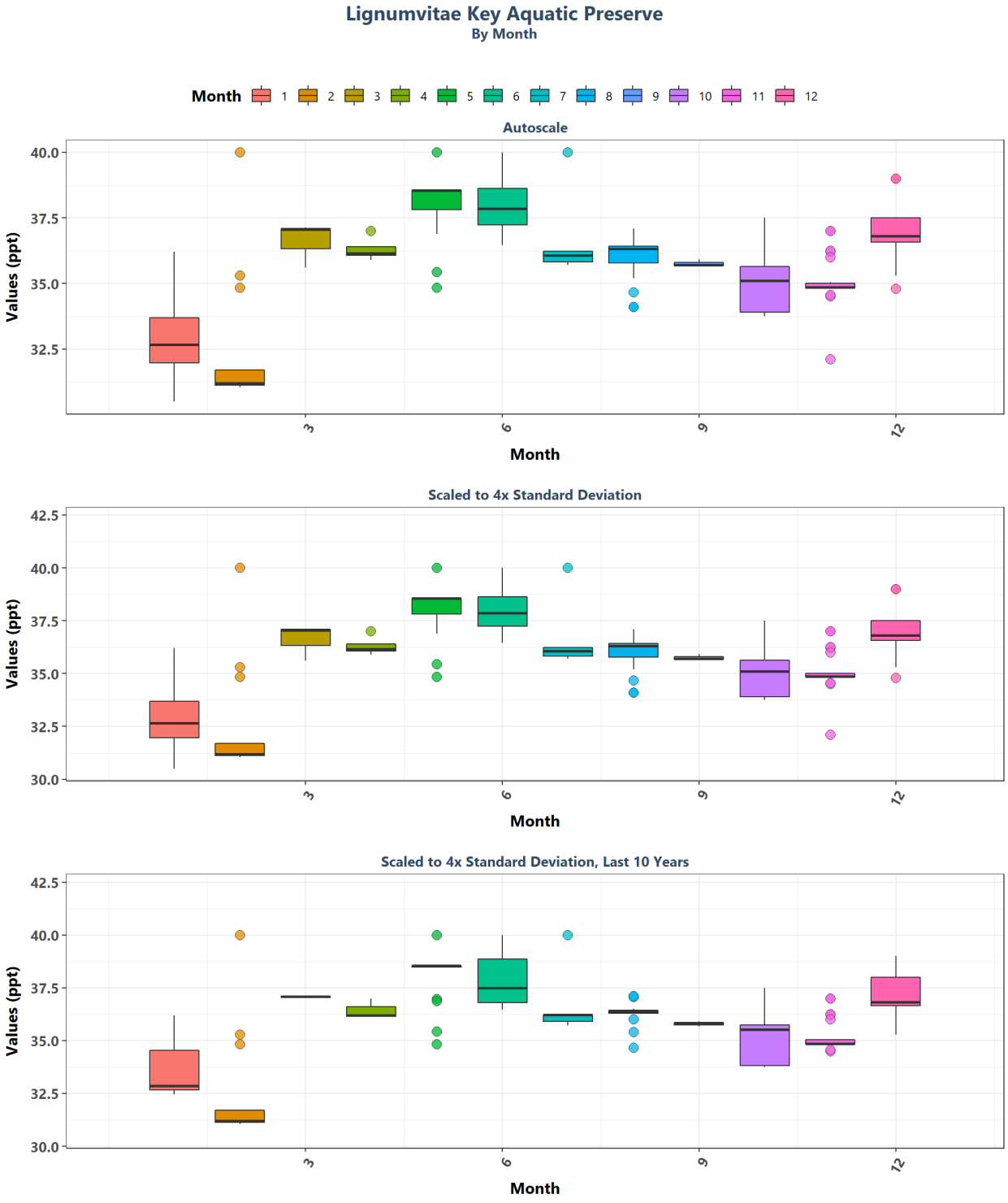

Lemon Bay Aquatic Preserve
By Year
Autoscale
Values (ppt)
40
30
20
10
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (ppt)
50
40
30
20
10
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
50
40
30
20
10
2012
2014
2016
2018
2020
2022
Year


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


Lemon Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
50
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
50
40
30
20
10
3
6
9
12
Month

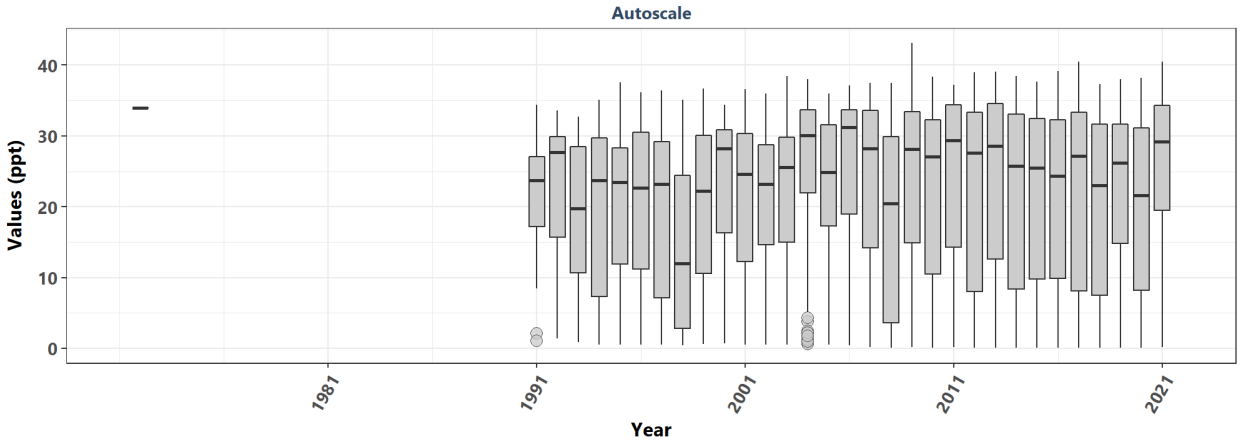

Lignumvitae Key Aquatic Preserve
By Year
Autoscale
Values (ppt)
40.0
37.5
35.0
32.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
42.5
40.0
37.5
35.0
32.5
30.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
42.5
40.0
37.5
35.0
32.5
30.0
2011
2013
2015
2017
2019
2021
Year

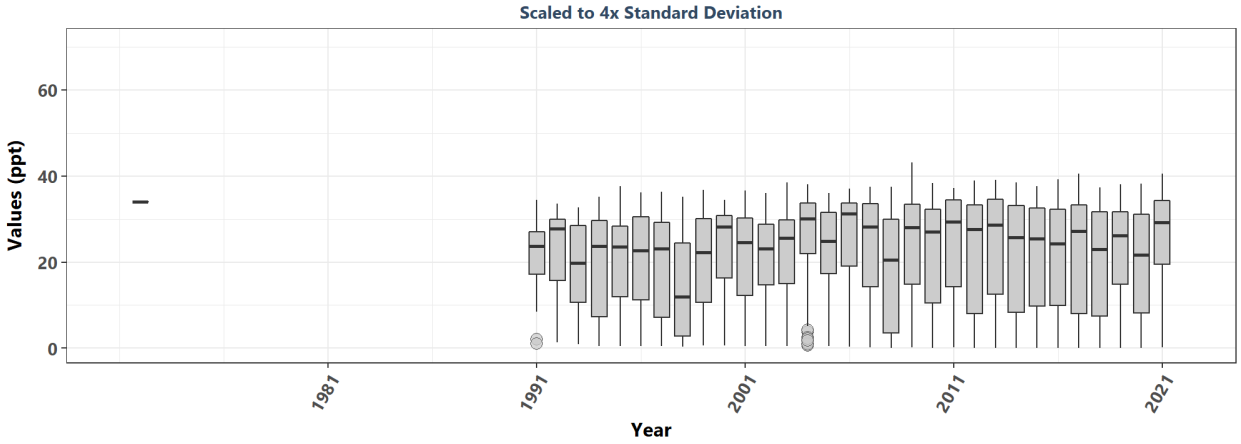

Lignumvitae Key Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40.0
37.5
35.0
32.5
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
42.5
40.0
37.5
35.0
32.5
30.0
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
42.5
40.0
37.5
35.0
32.5
30.0
2011
2013
2015
2017
2019
2021
Year

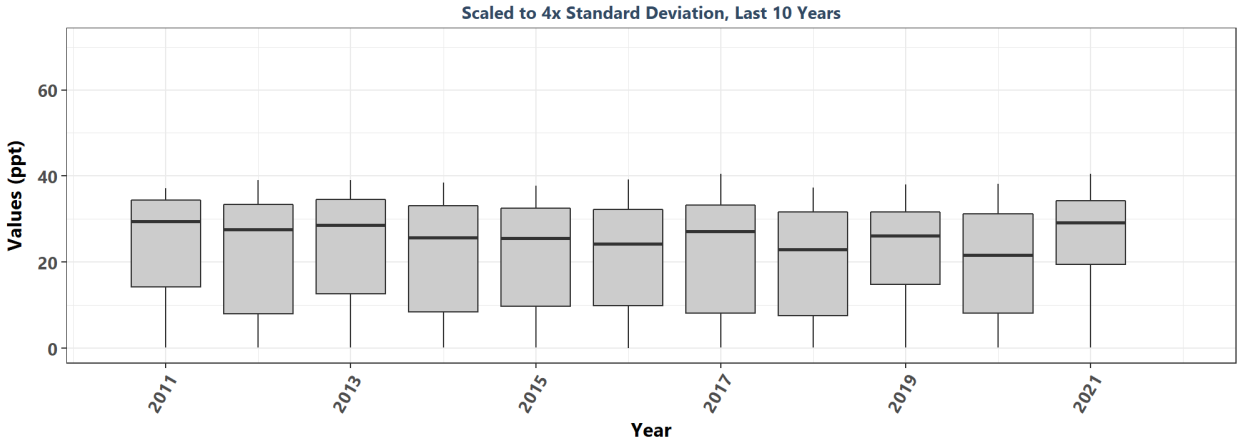

Lignumvitae Key Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40.0
37.5
35.0
32.5
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
42.5
40.0
37.5
35.0
32.5
30.0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
42.5
40.0
37.5
35.0
32.5
30.0
3
6
9
12
Month

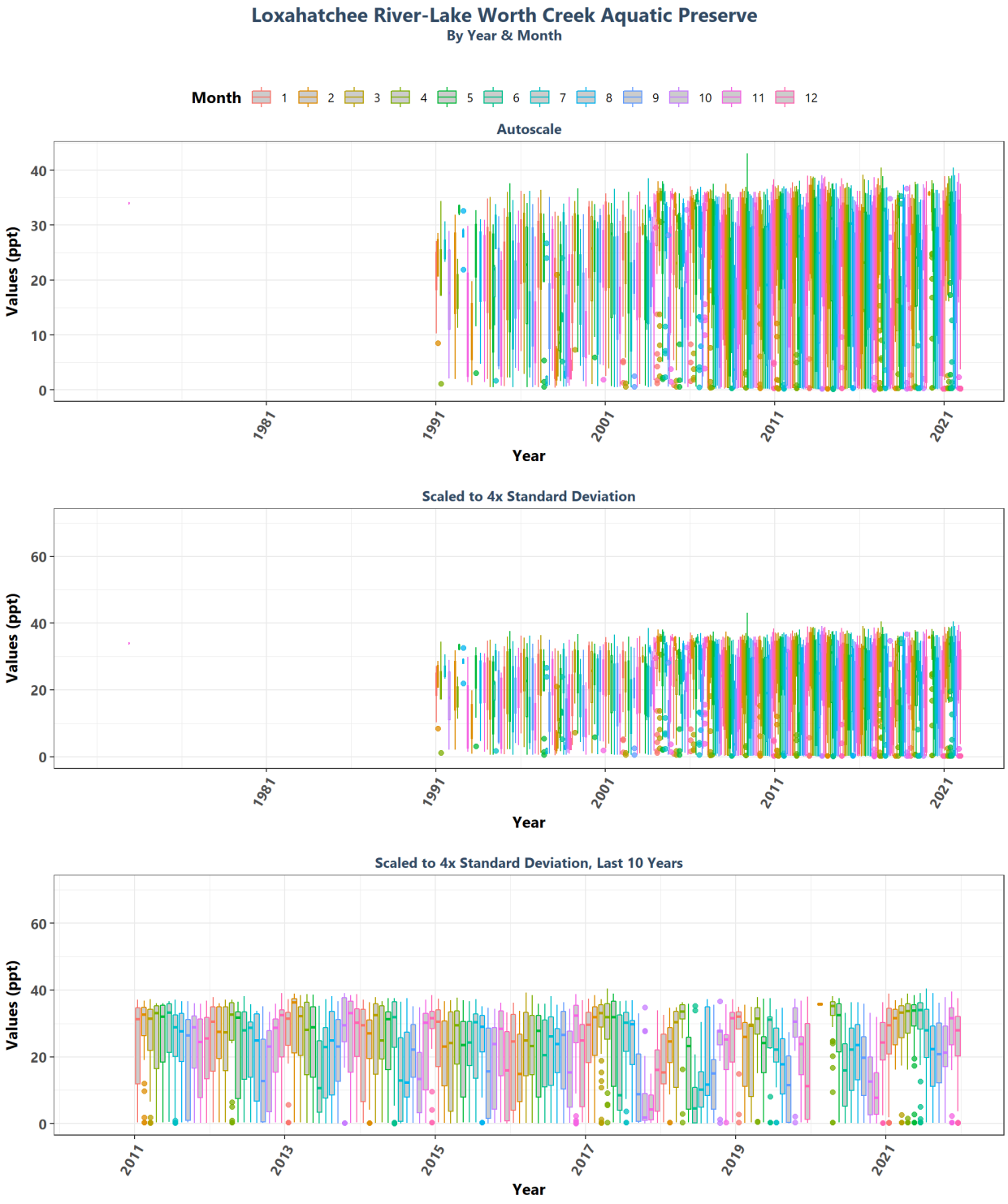
Loxahatchee River-Lake Worth Creek Aquatic Preserve

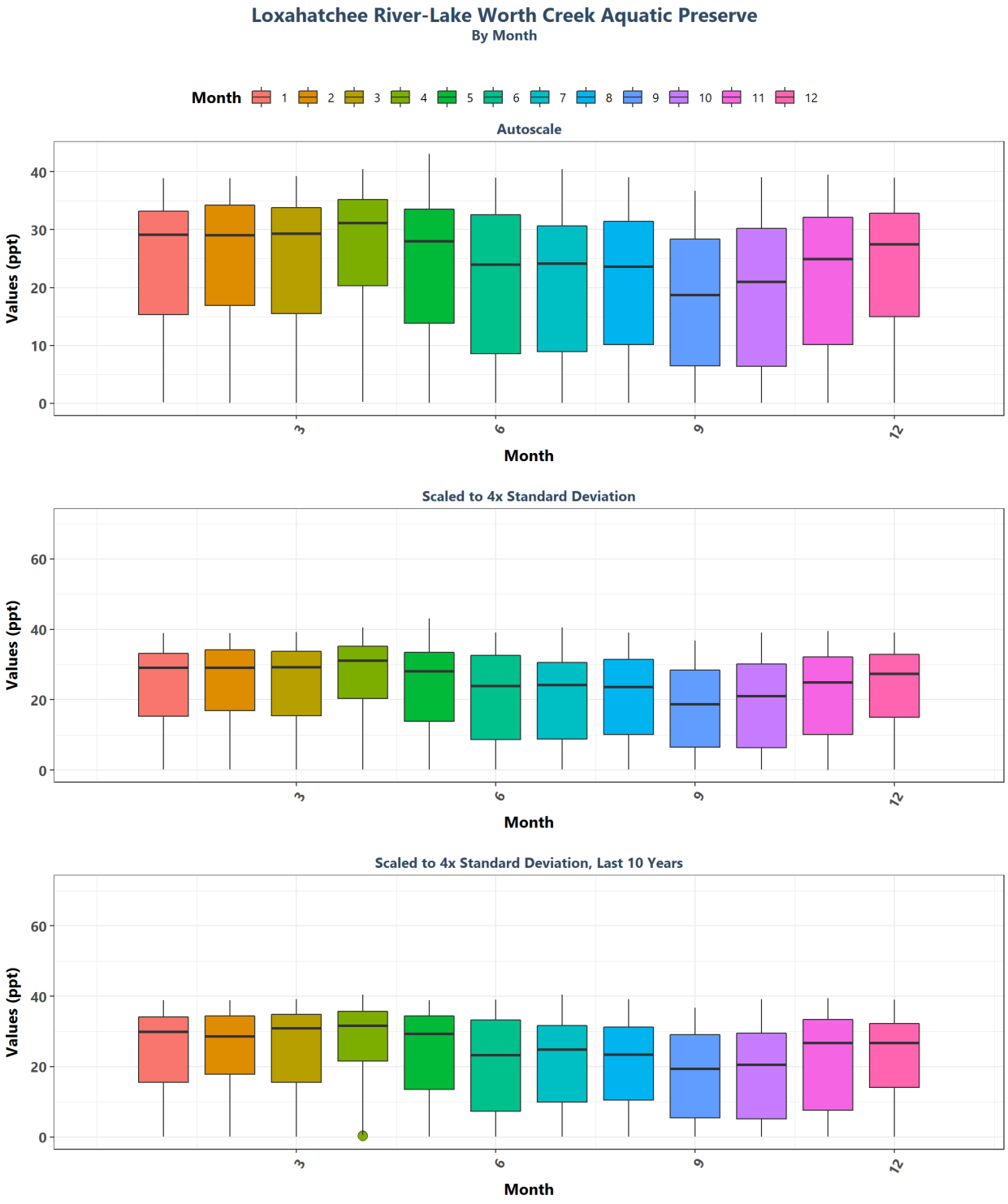
By Year





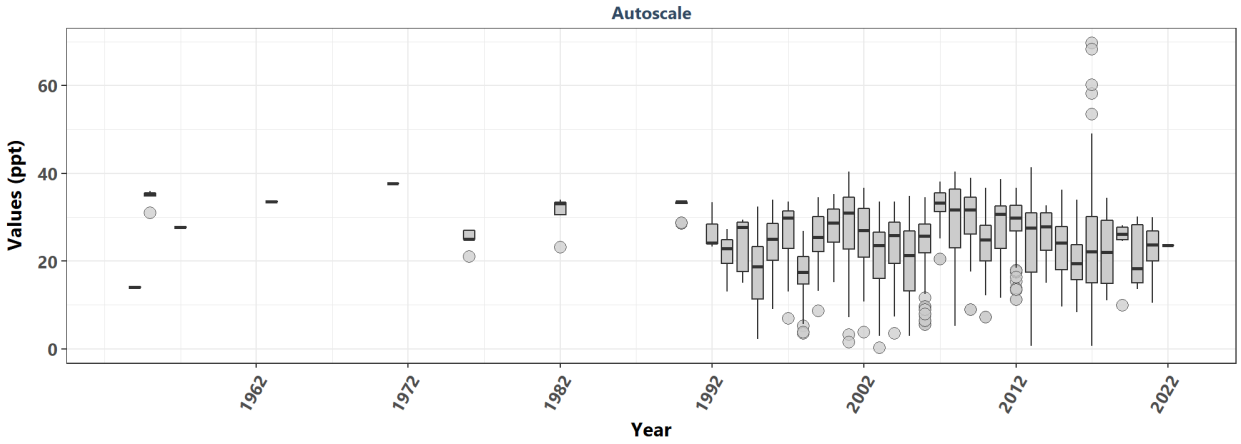


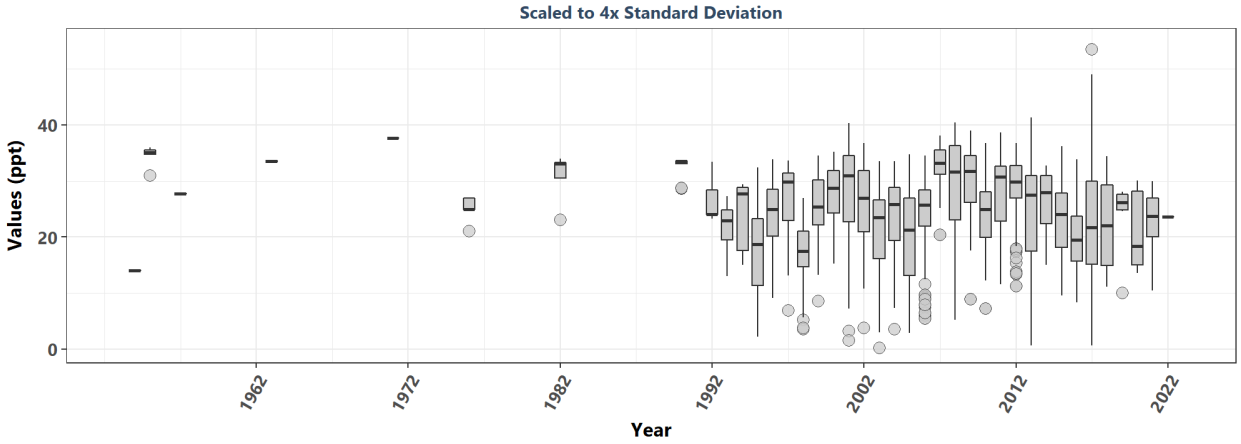

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

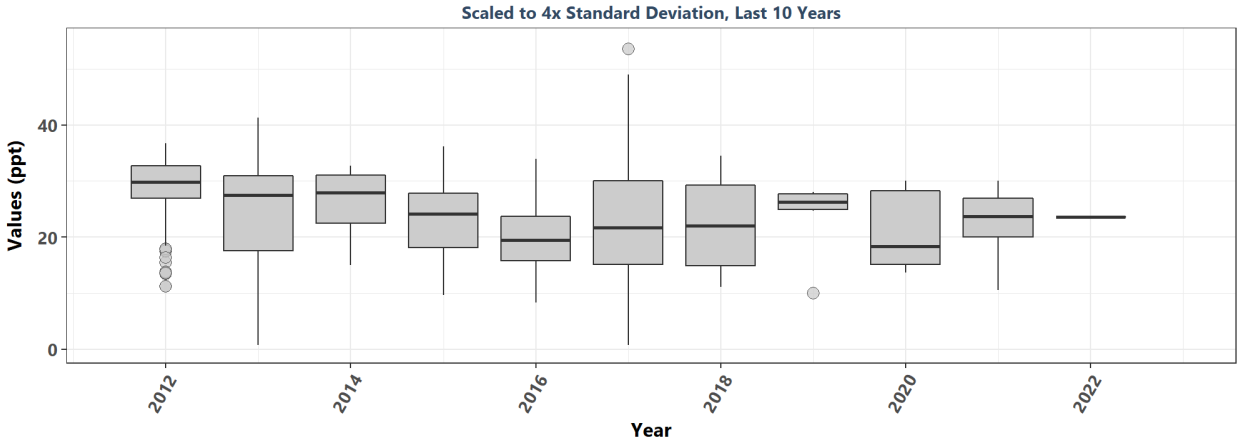

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

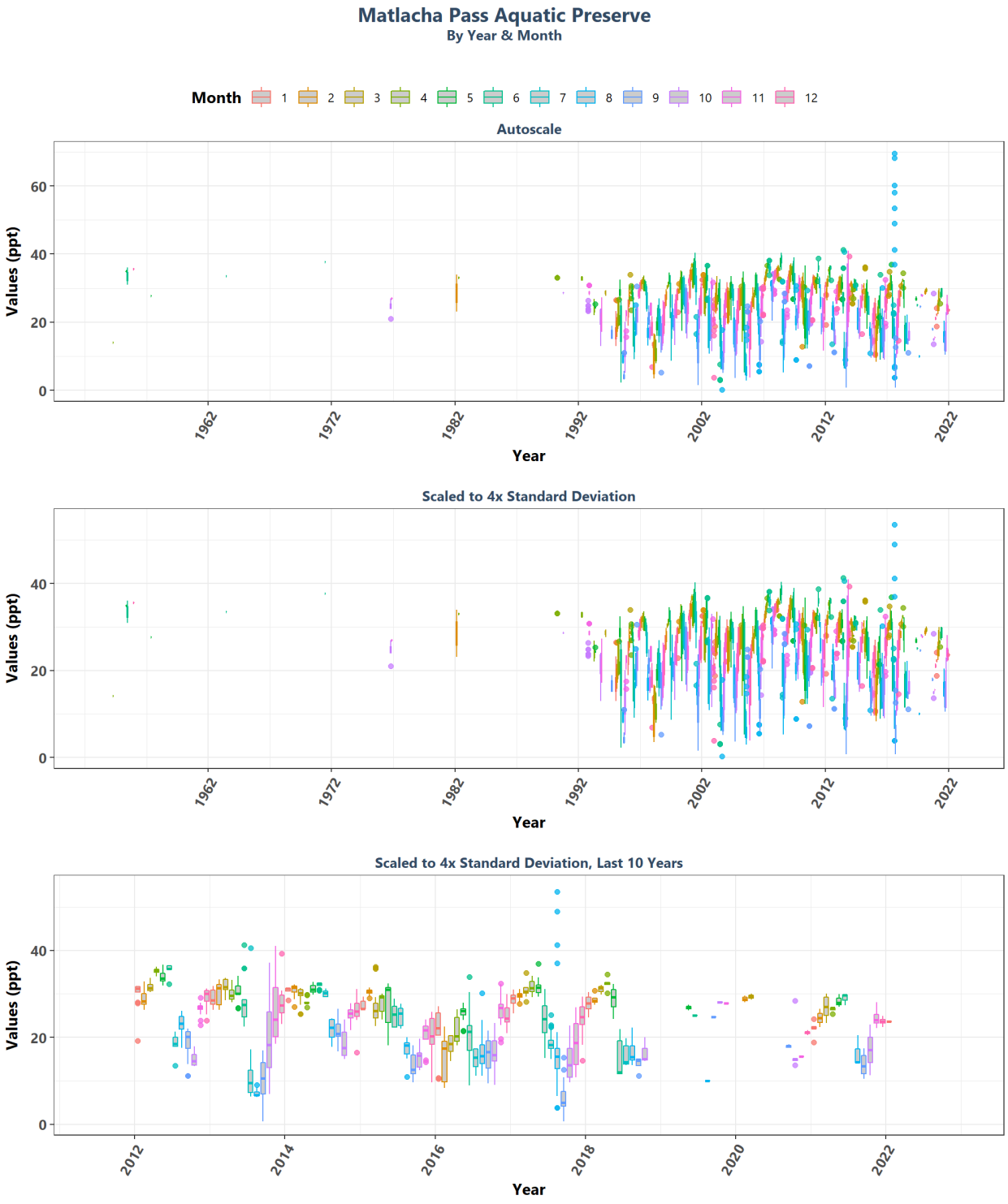
Matlacha Pass Aquatic Preserve

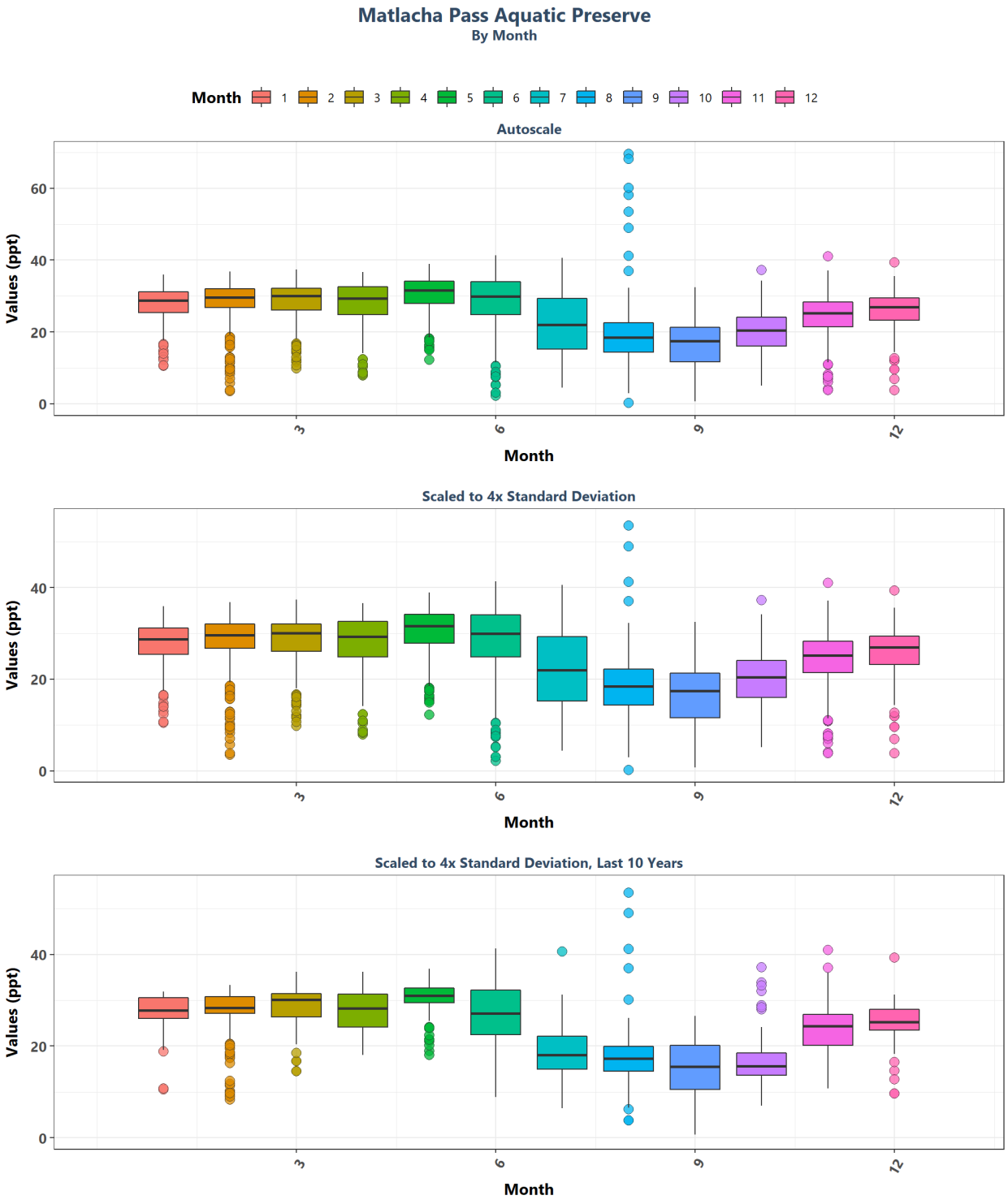
By Year



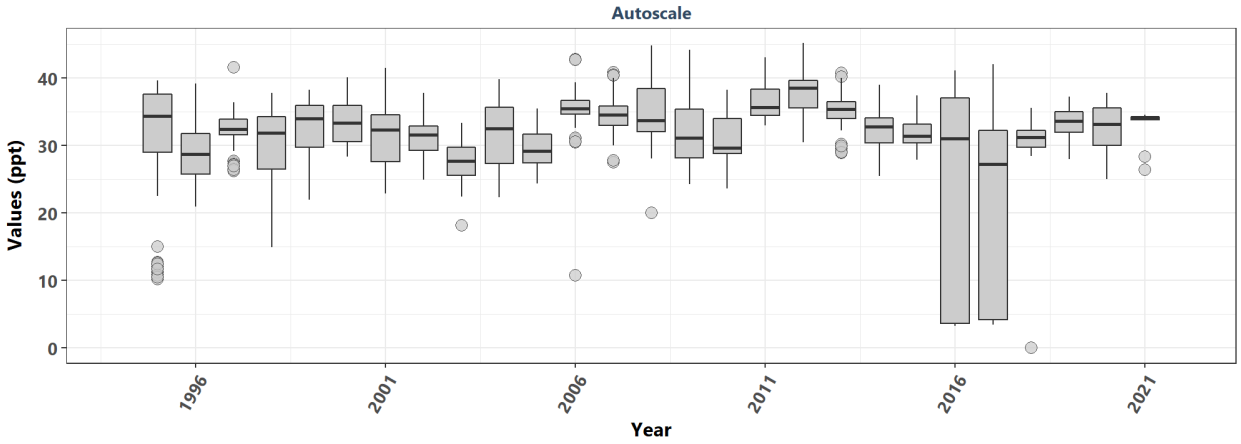


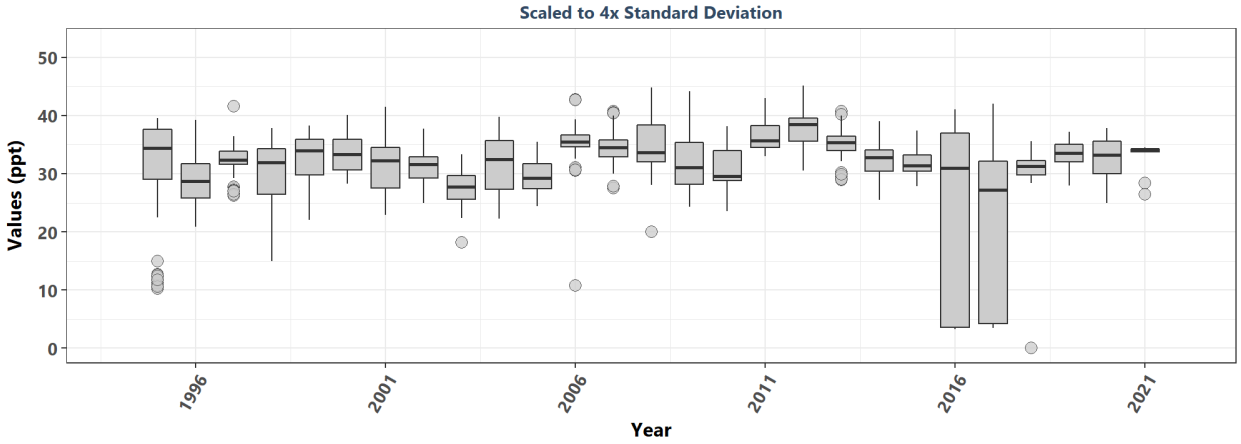


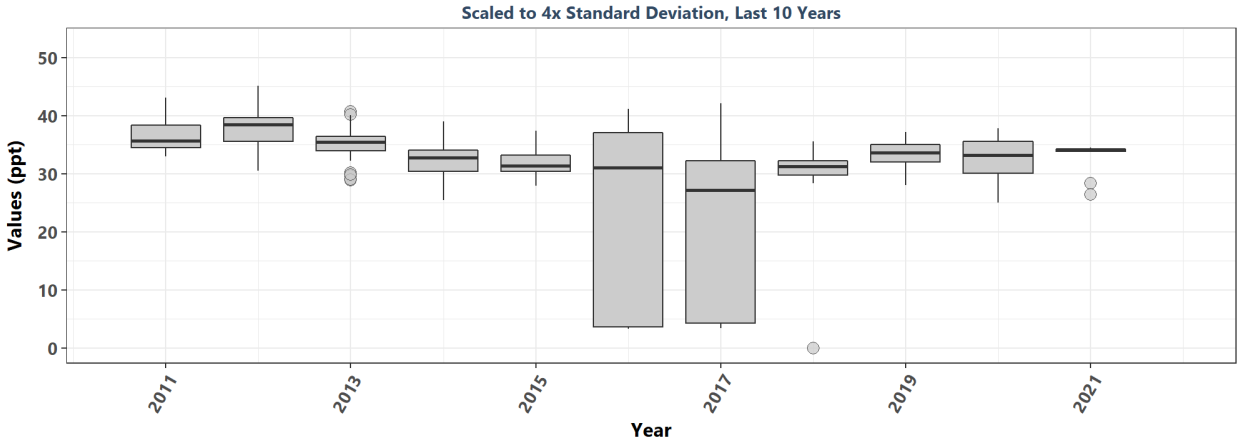

Matlacha Pass Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
20
40
60
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
2012
2014
2016
2018
2020
2022
Year

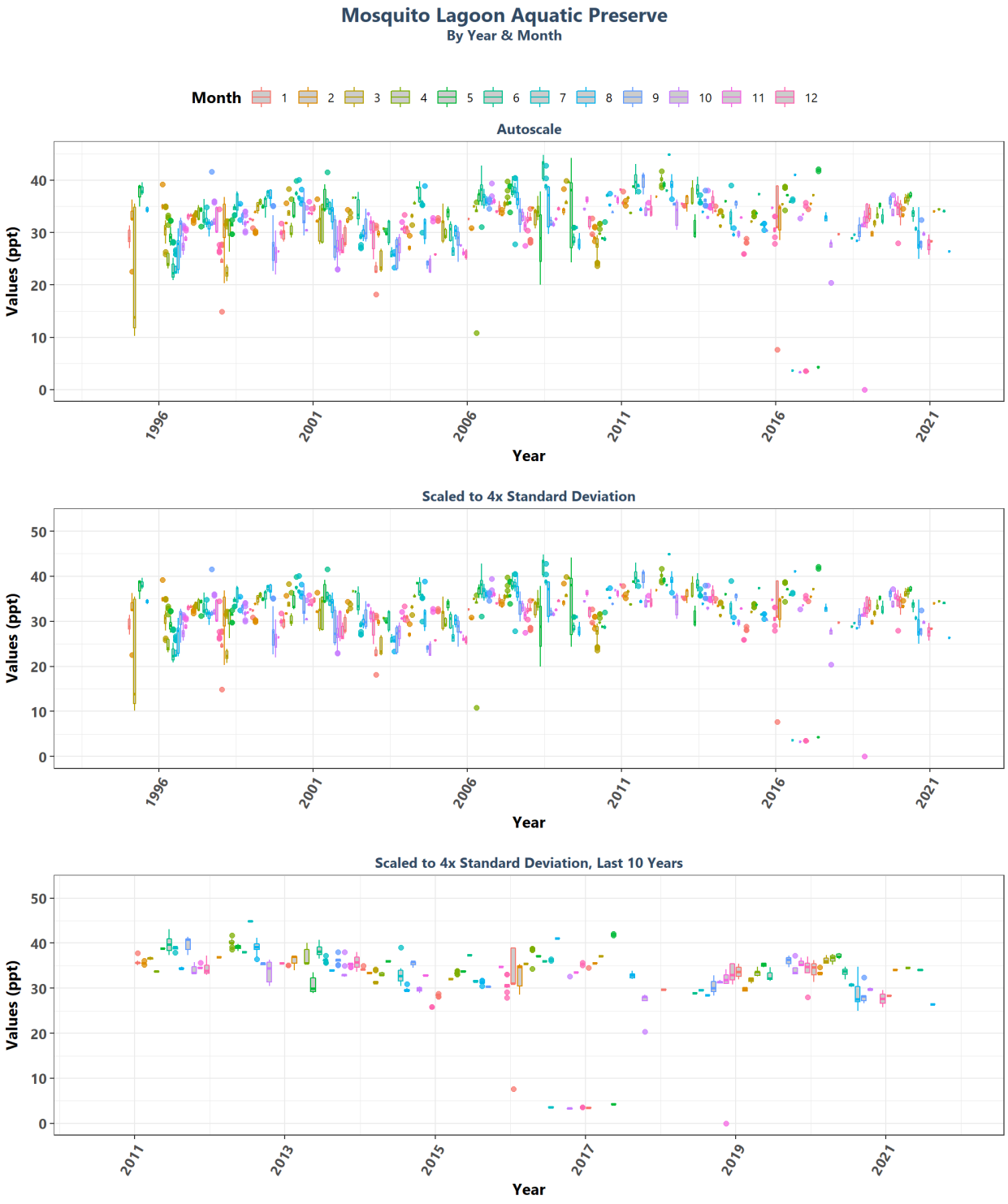

Matlacha Pass Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
20
40
60
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
3
6
9
12
Month

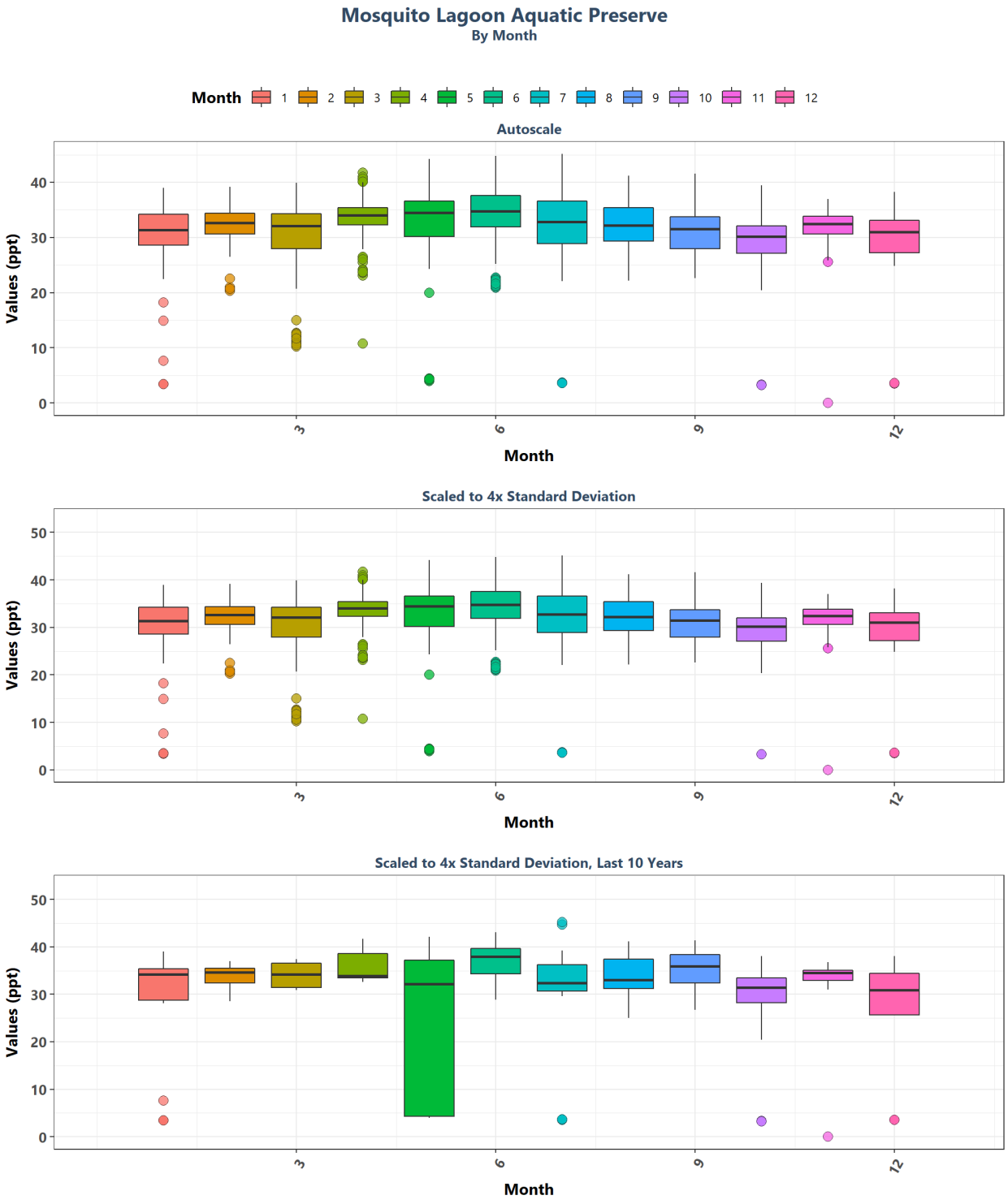
Mosquito Lagoon Aquatic Preserve
By Year





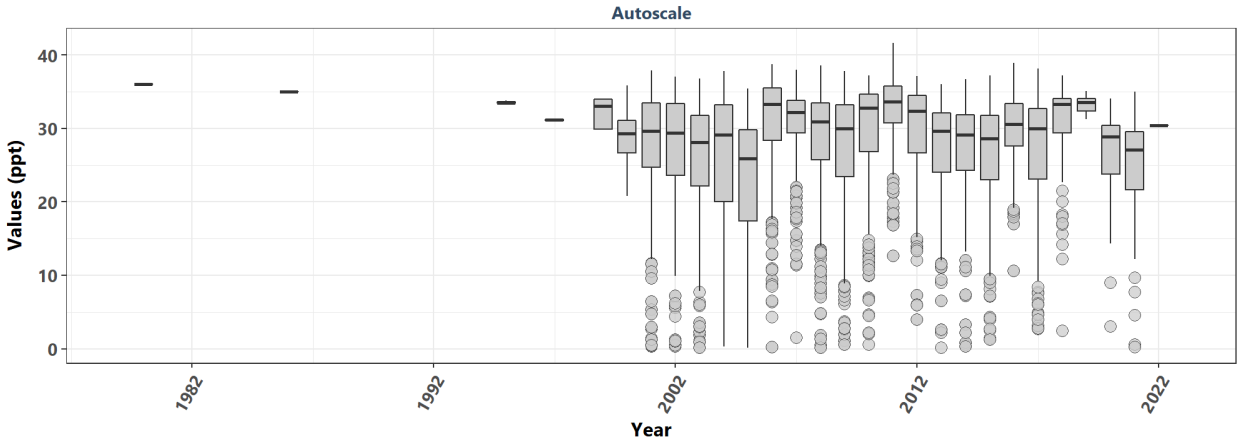


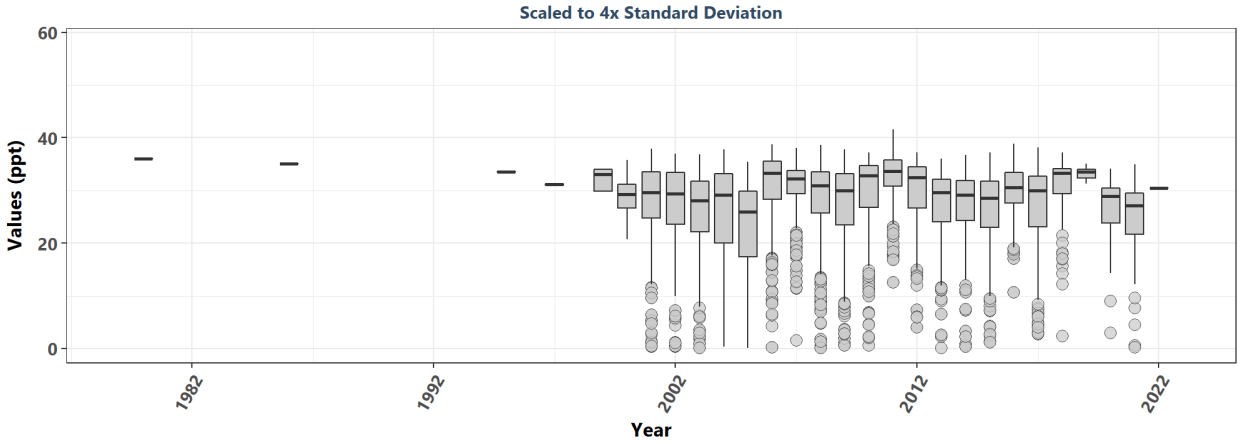

Mosquito Lagoon Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

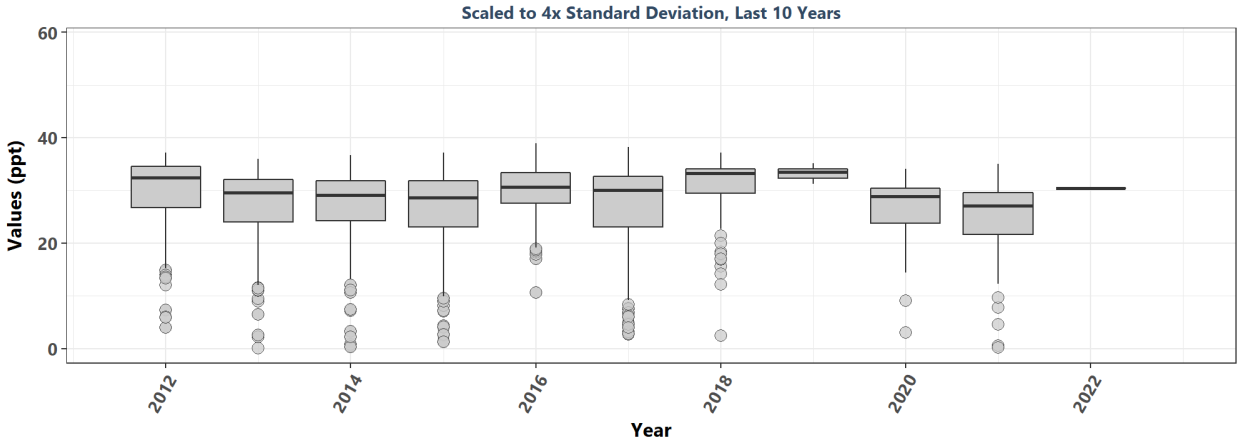

Mosquito Lagoon Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

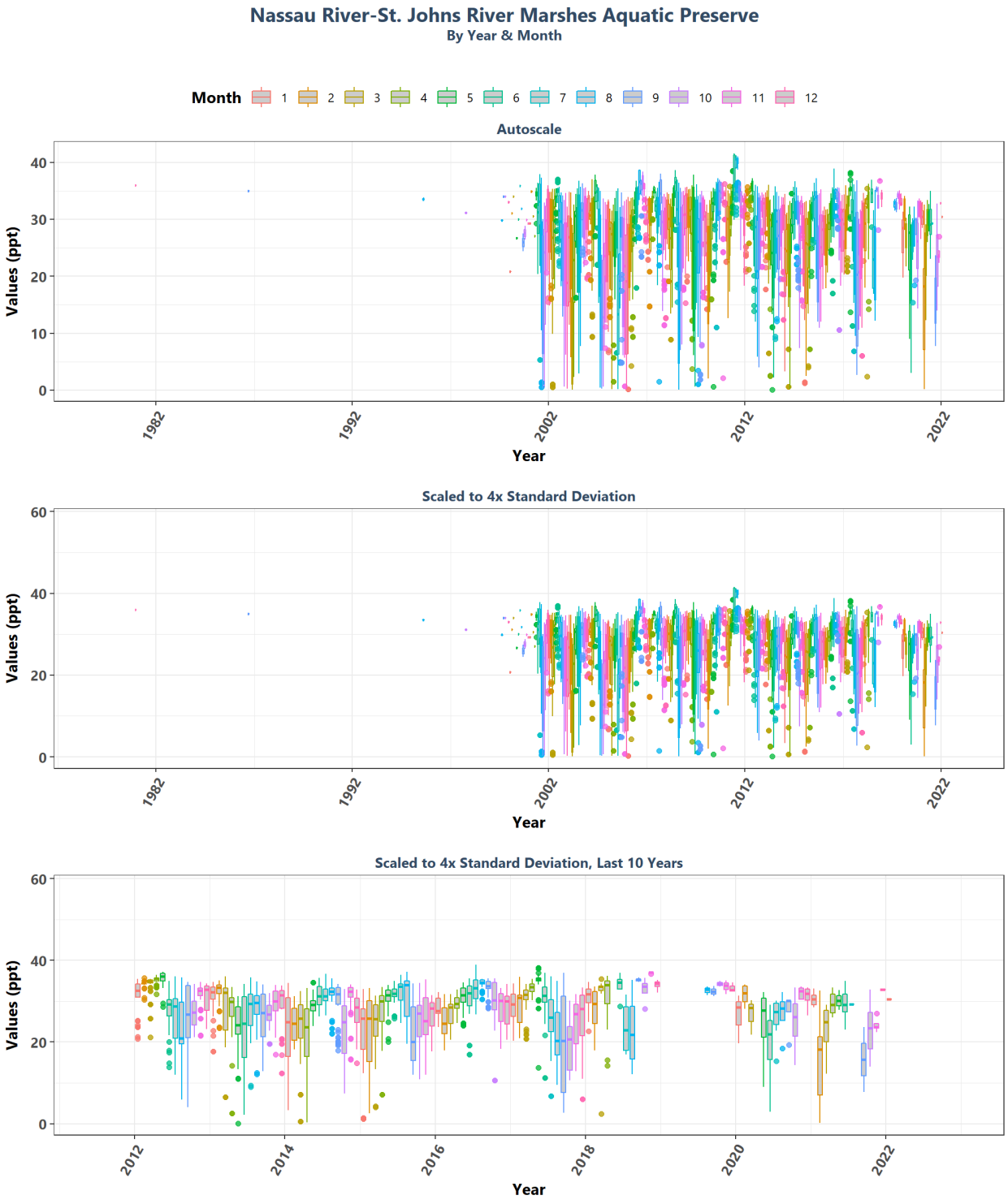
Nassau River-St. Johns River Marshes Aquatic Preserve

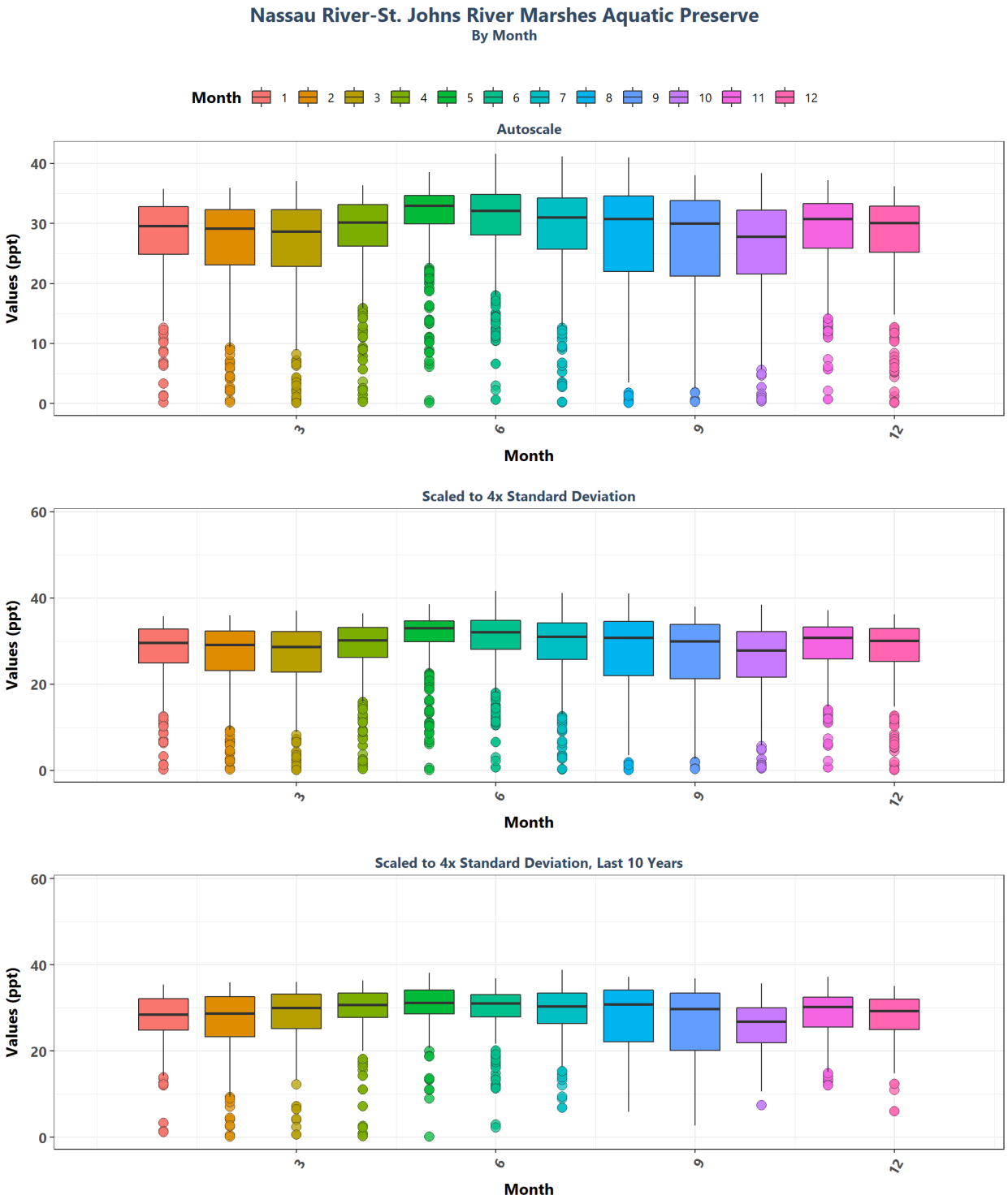
By Year

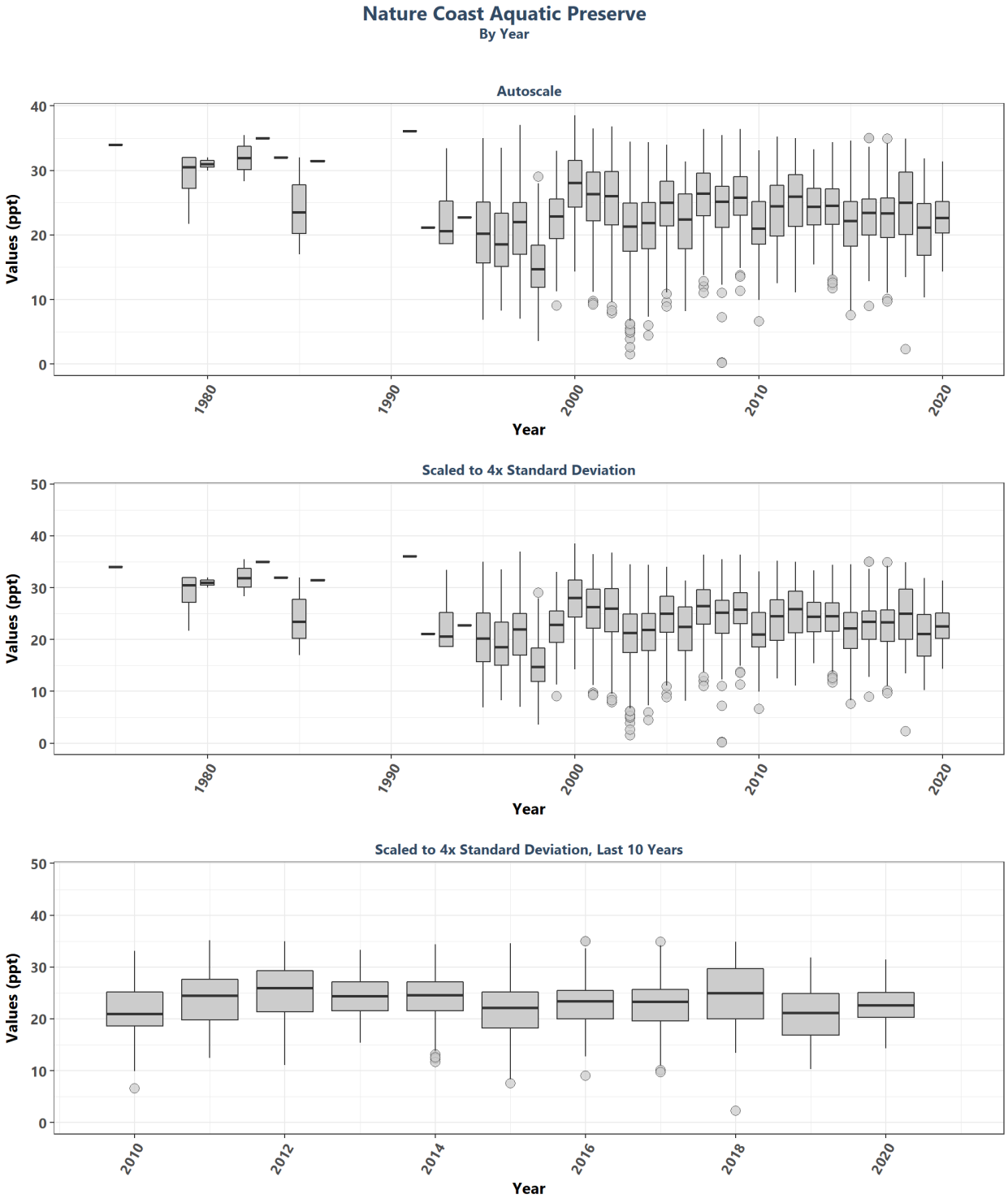


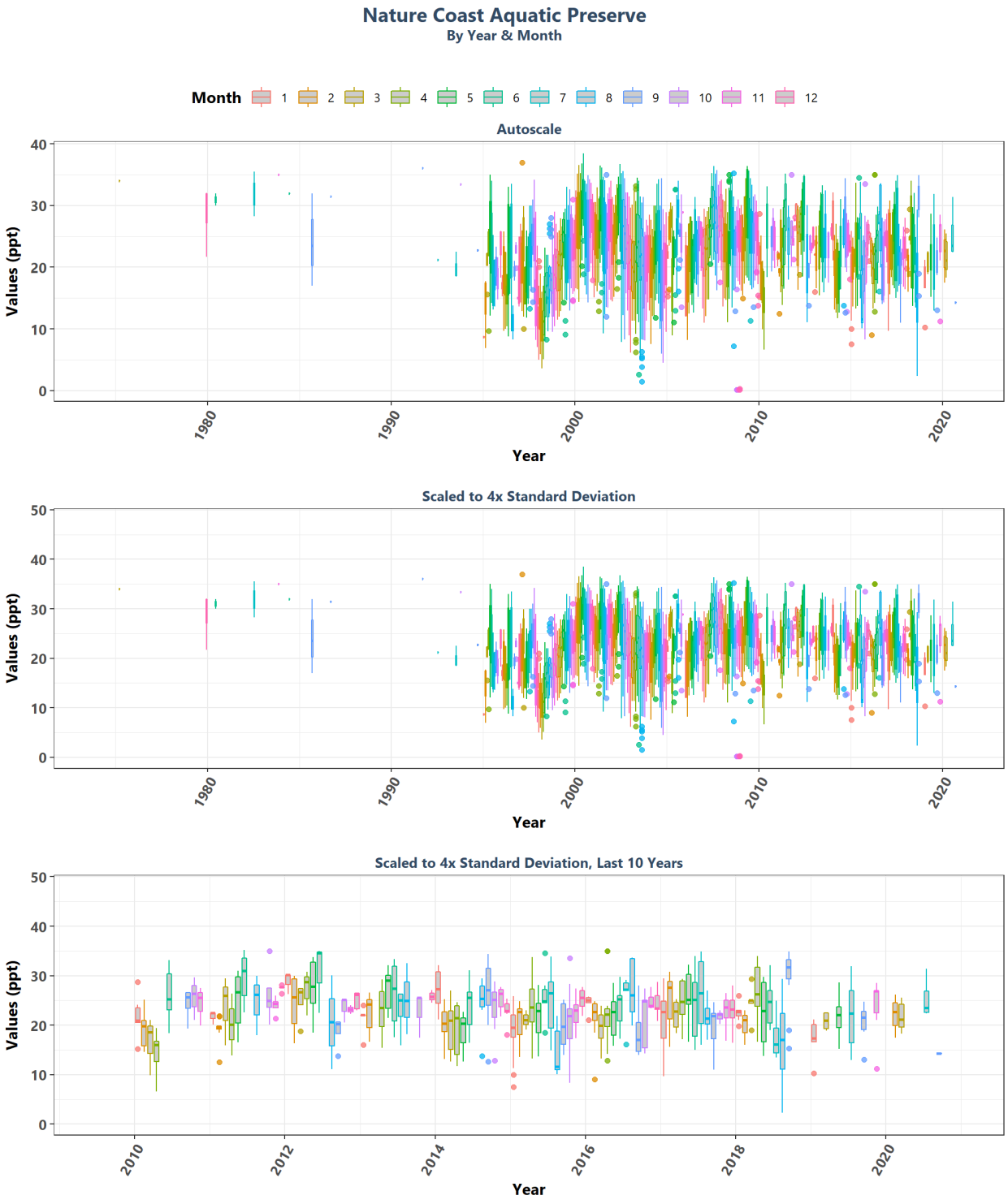


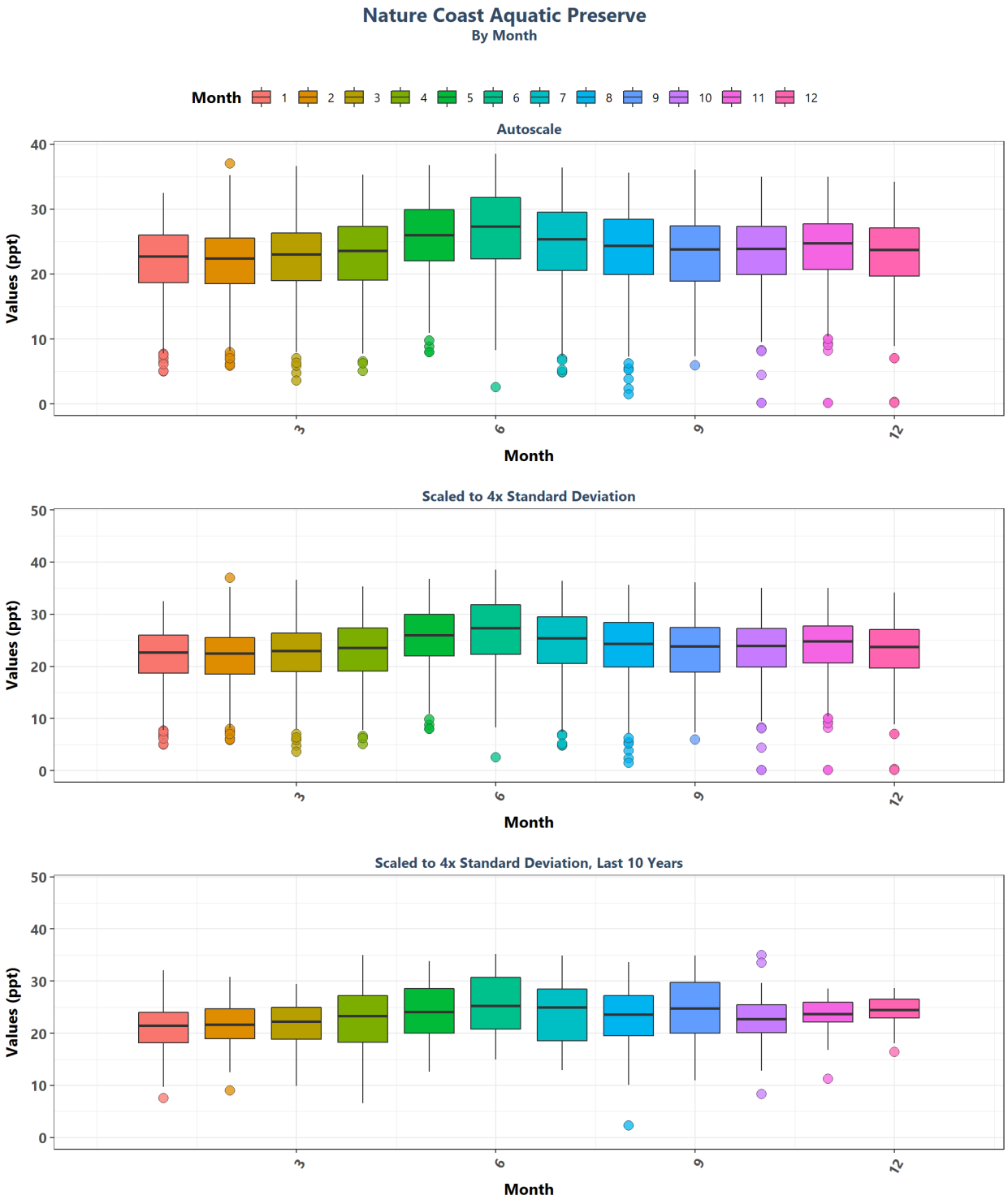



Nassau River-St. Johns River Marshes Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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 (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

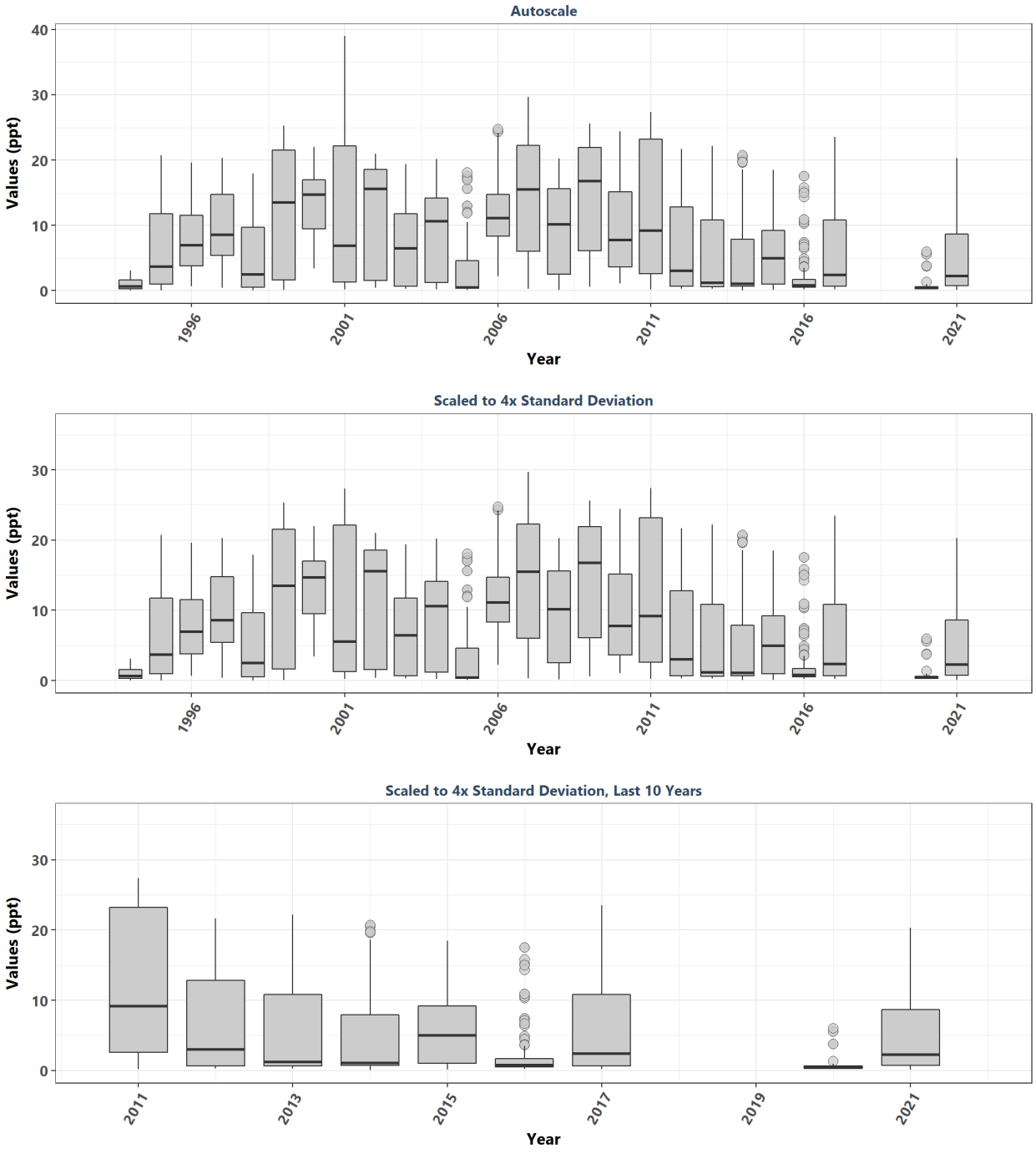

Nature Coast Aquatic Preserve
By Year
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

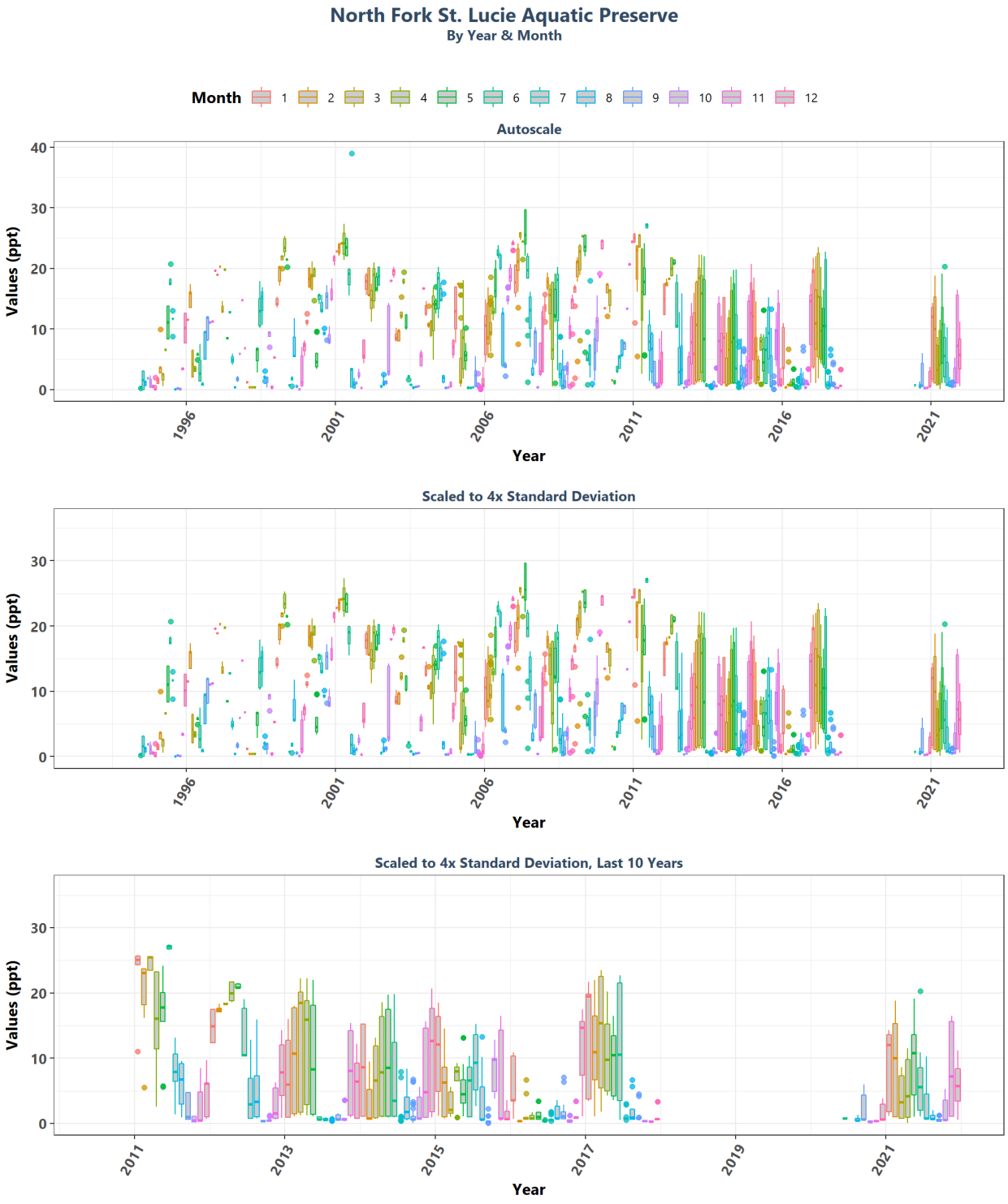

Nature Coast Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

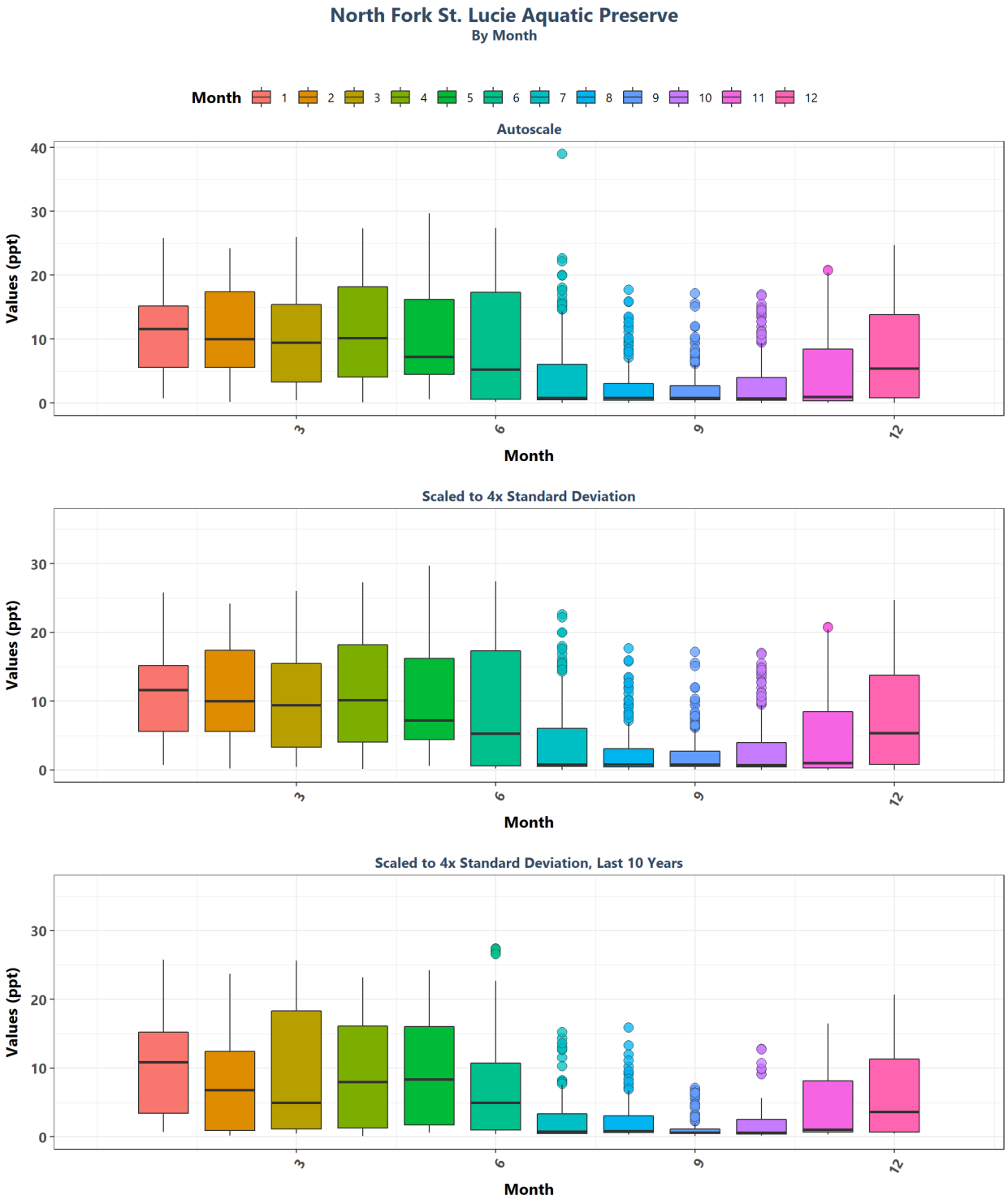

Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

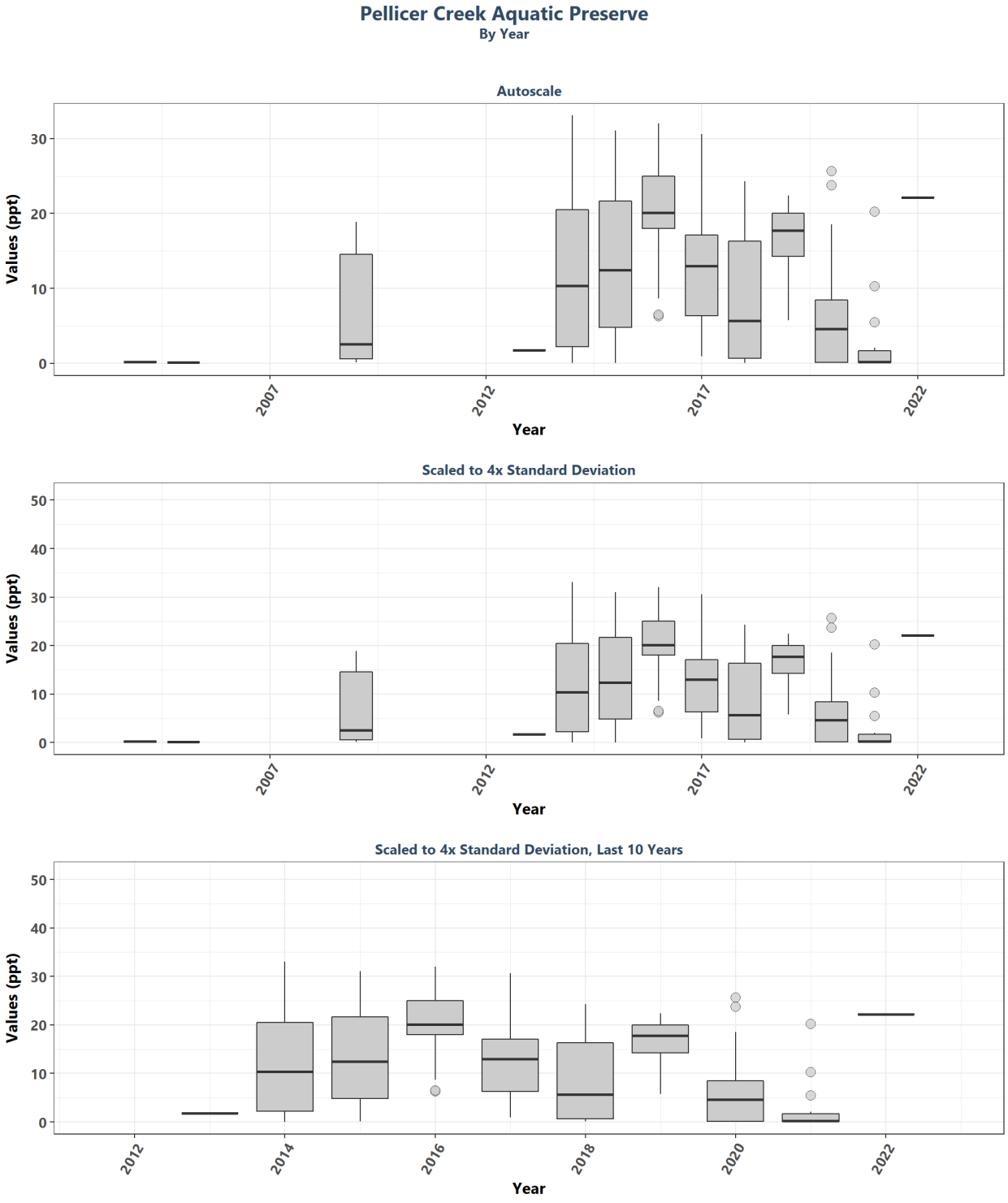
North Fork St. Lucie Aquatic Preserve

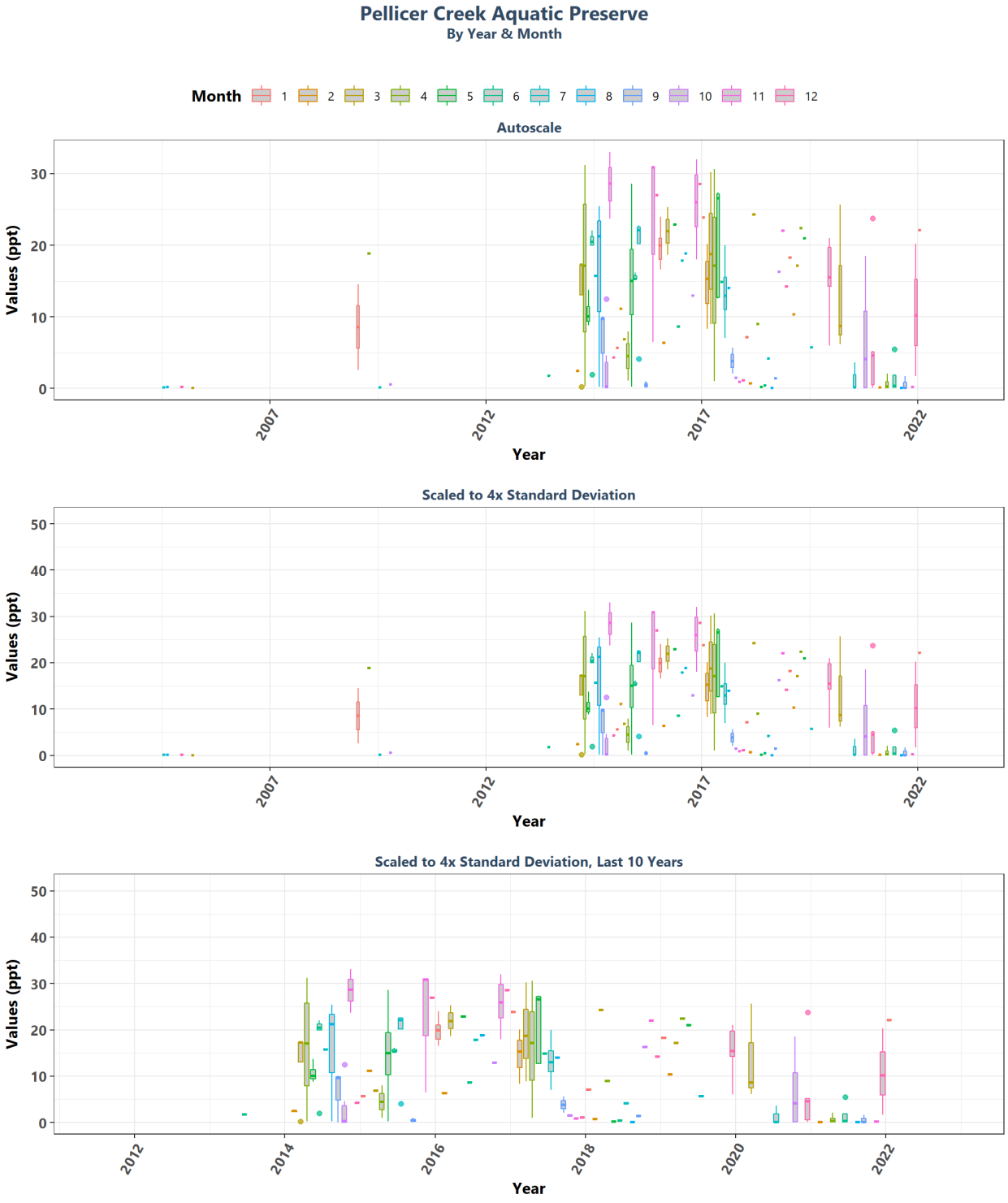
By Year

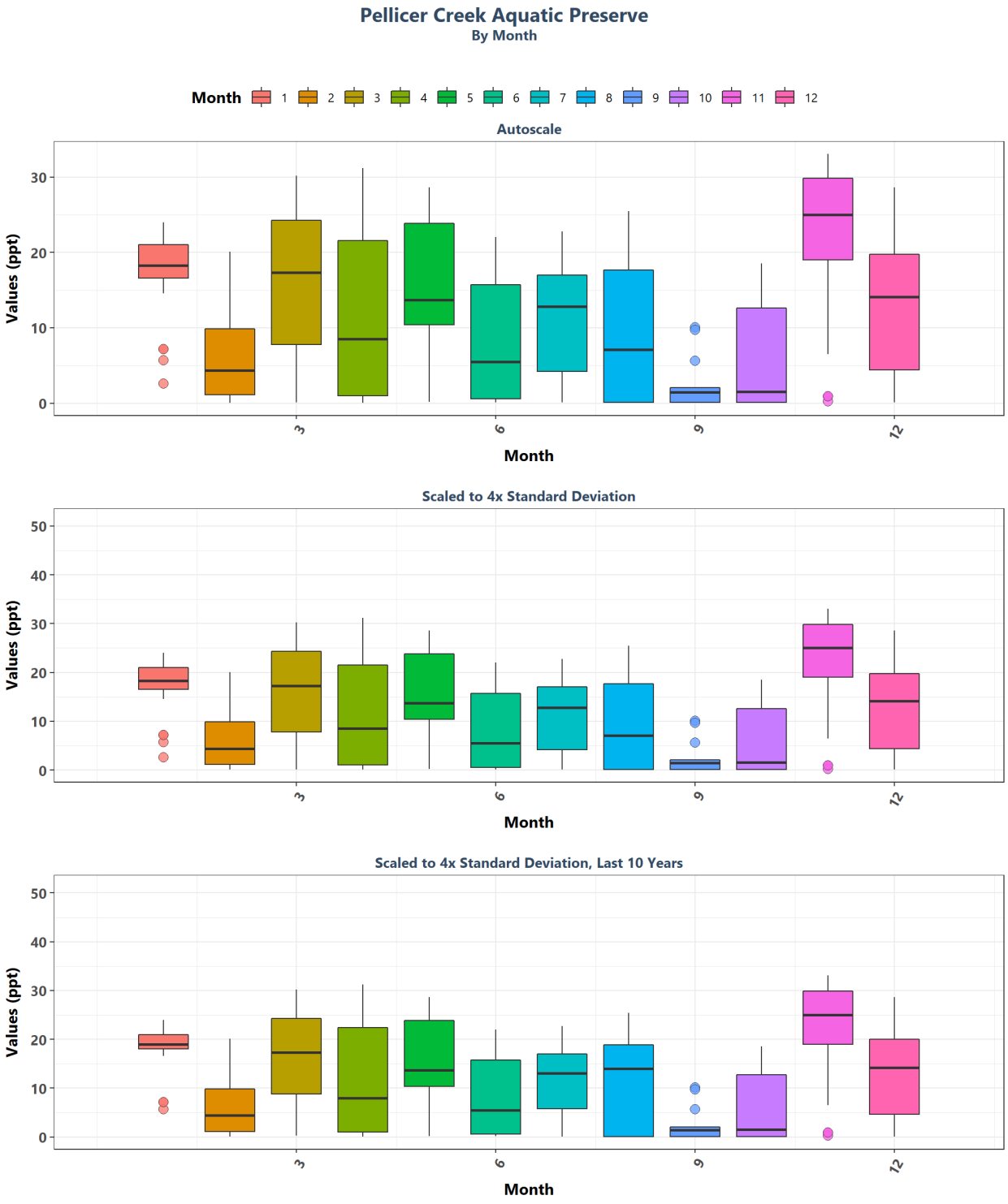



North Fork St. Lucie Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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

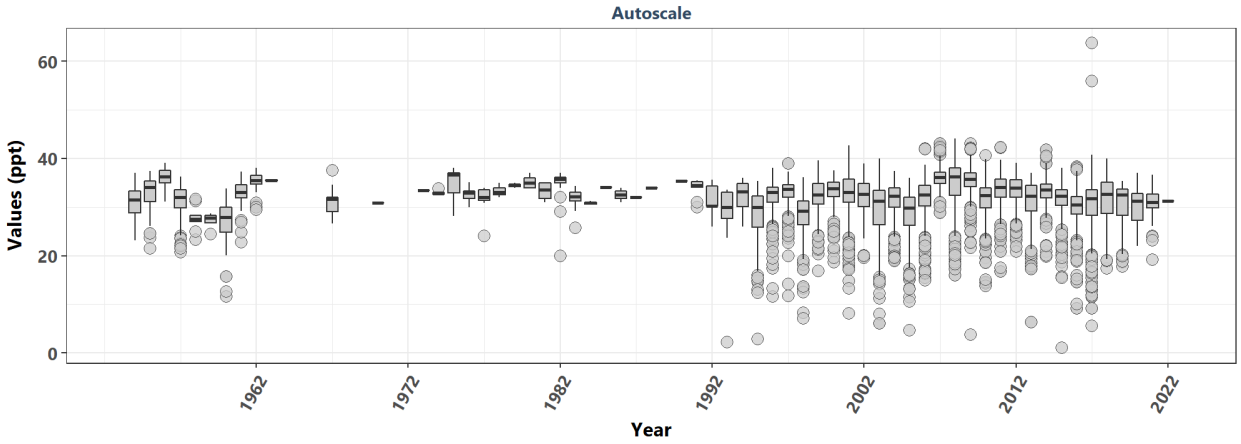

Pellicer Creek Aquatic Preserve
By Year
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

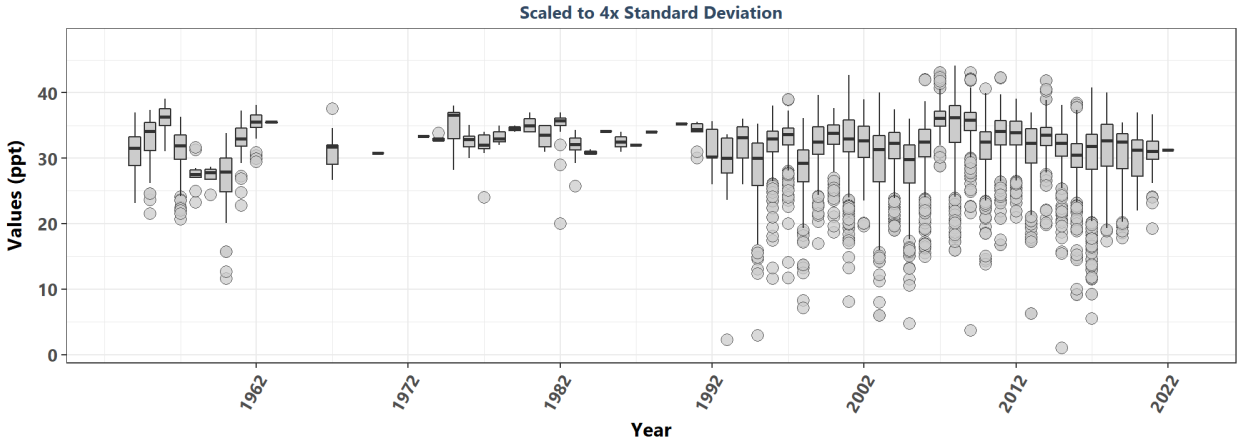

Pellicer Creek Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

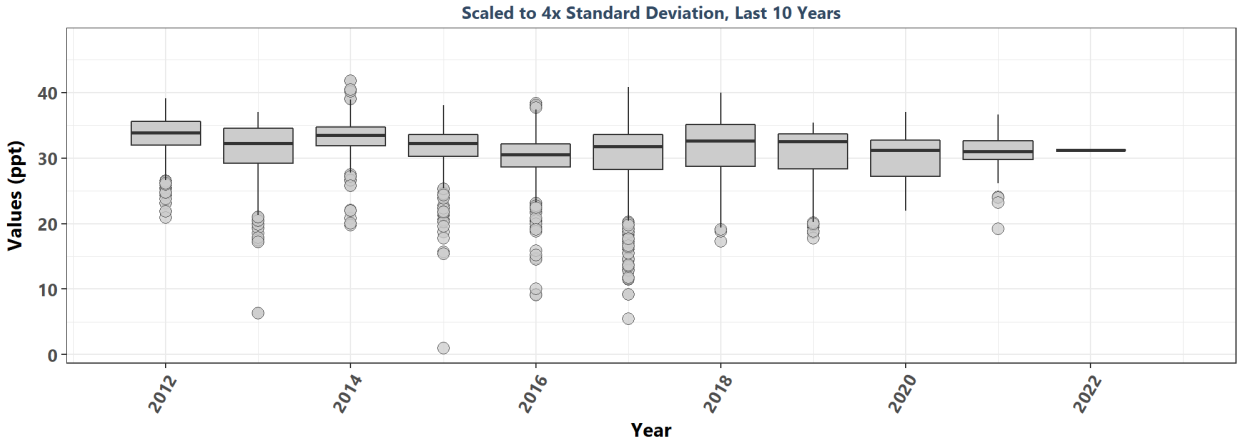

Pellicer Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

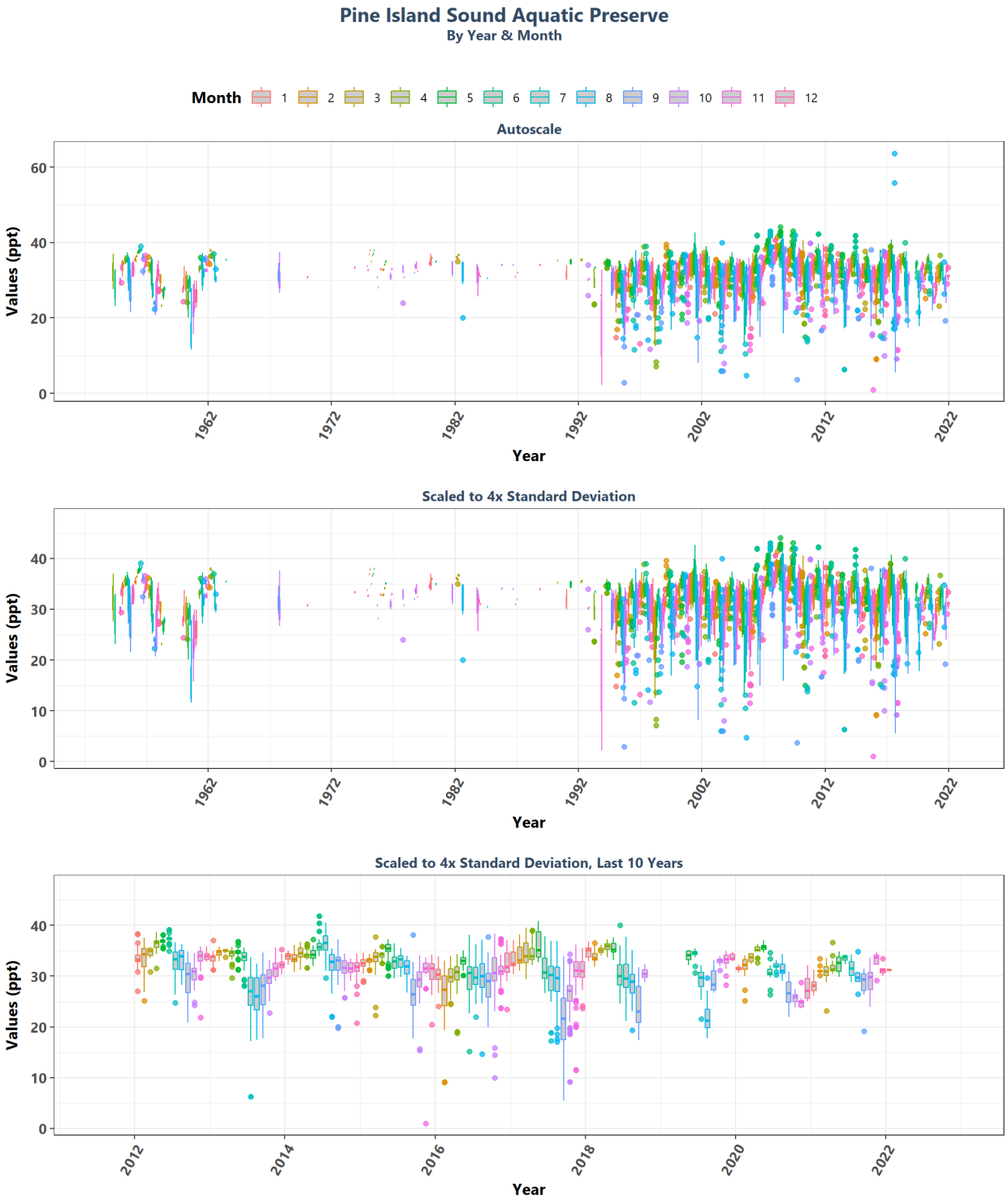
Pine Island Sound Aquatic Preserve

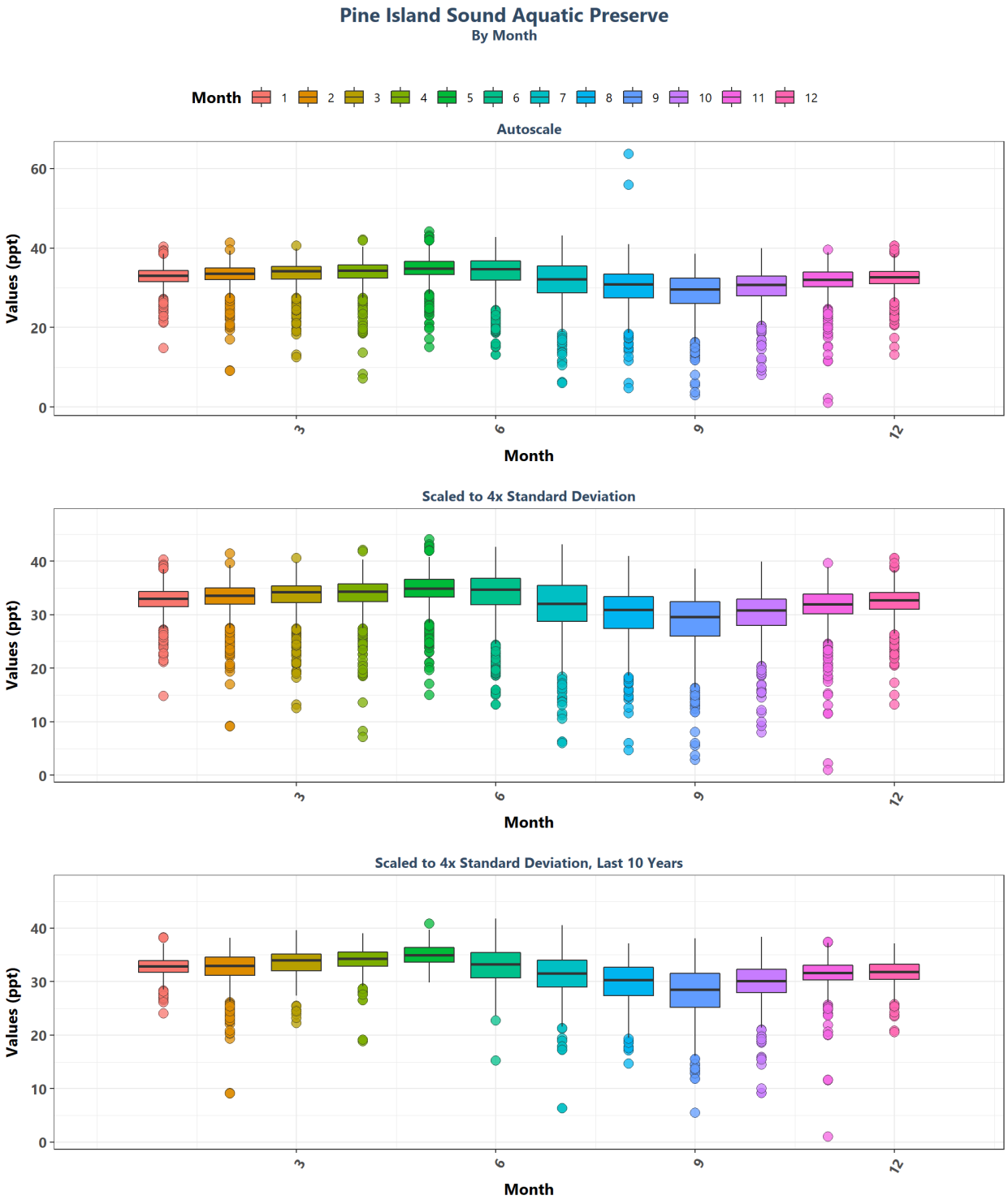
By Year

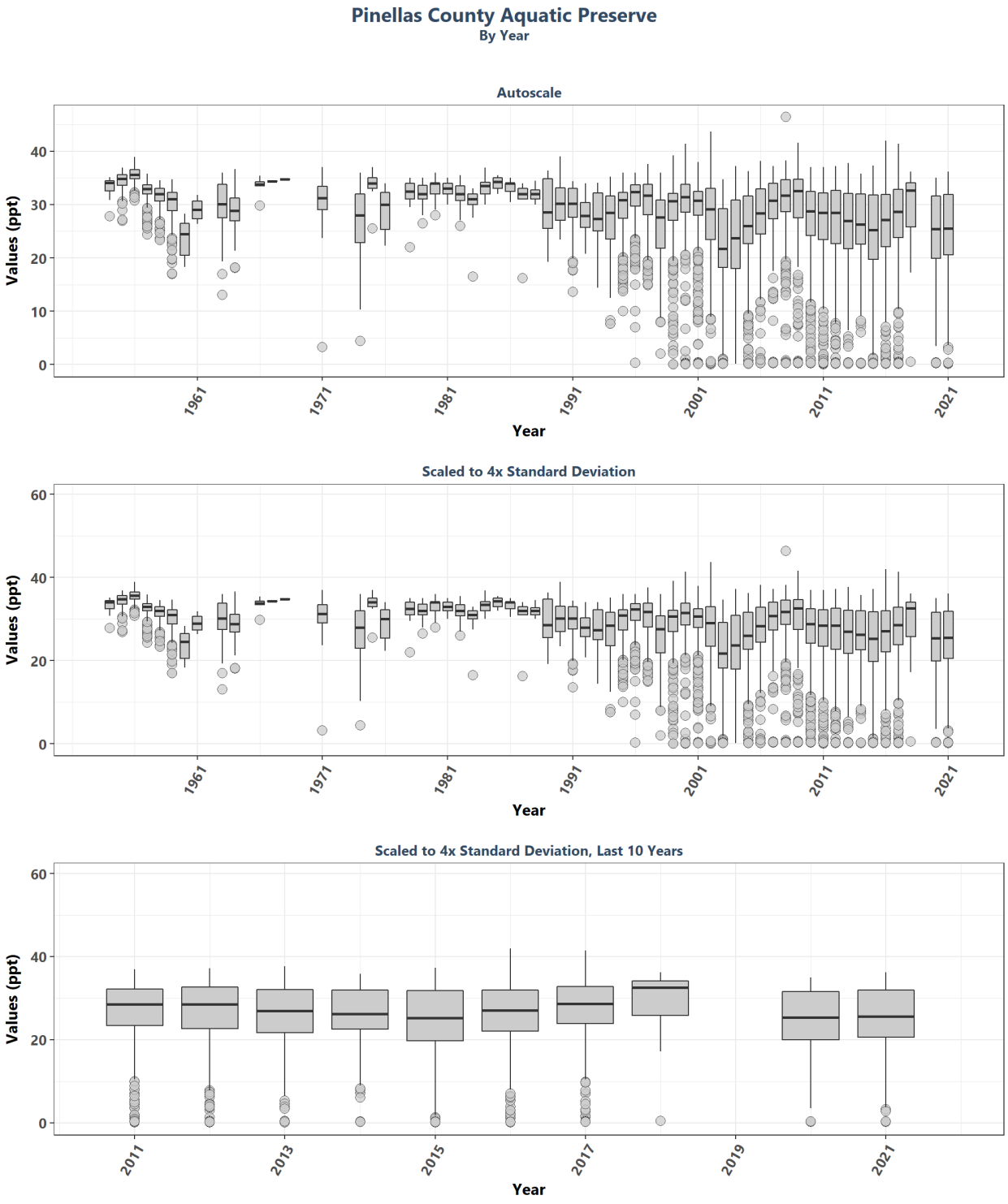


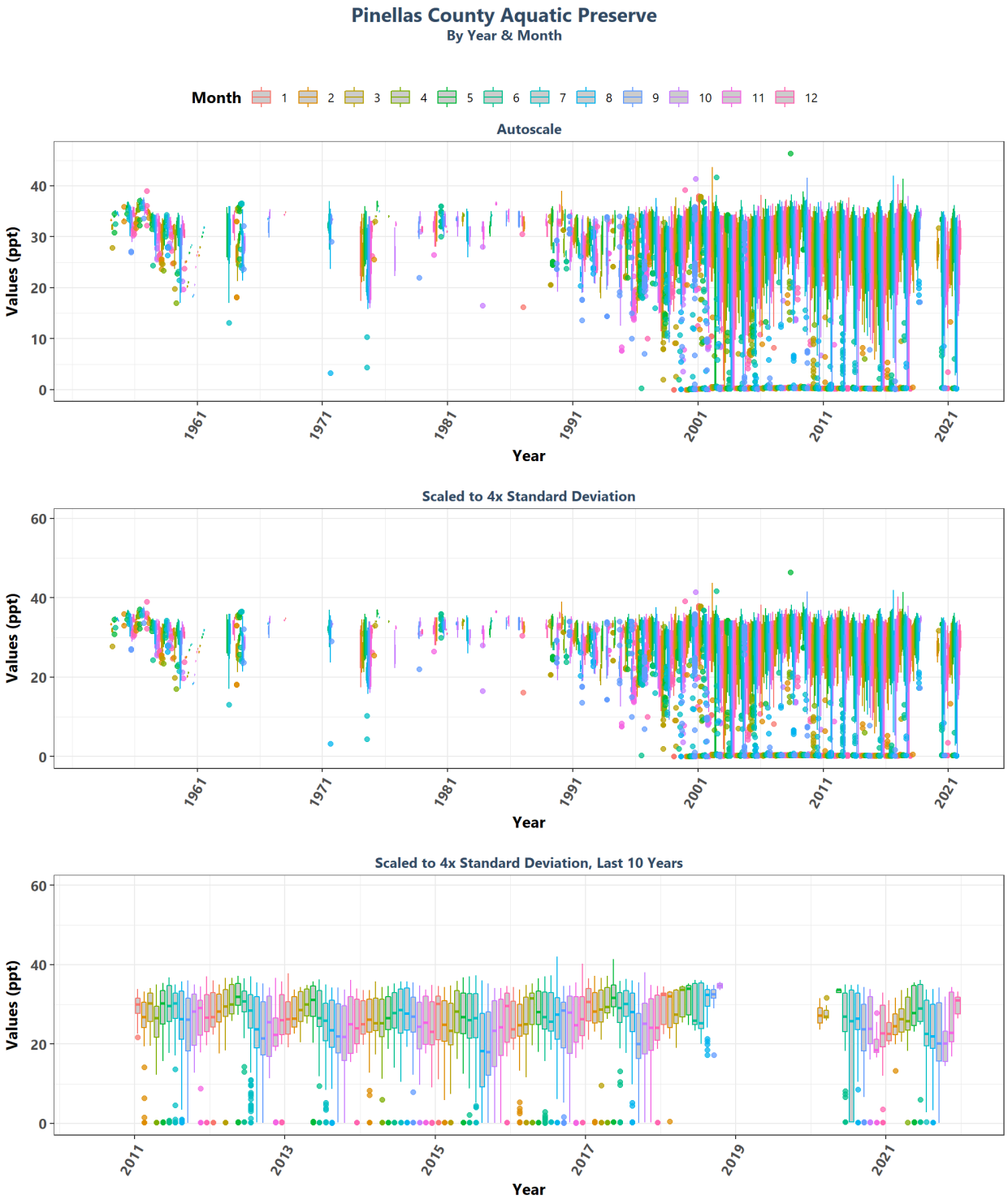


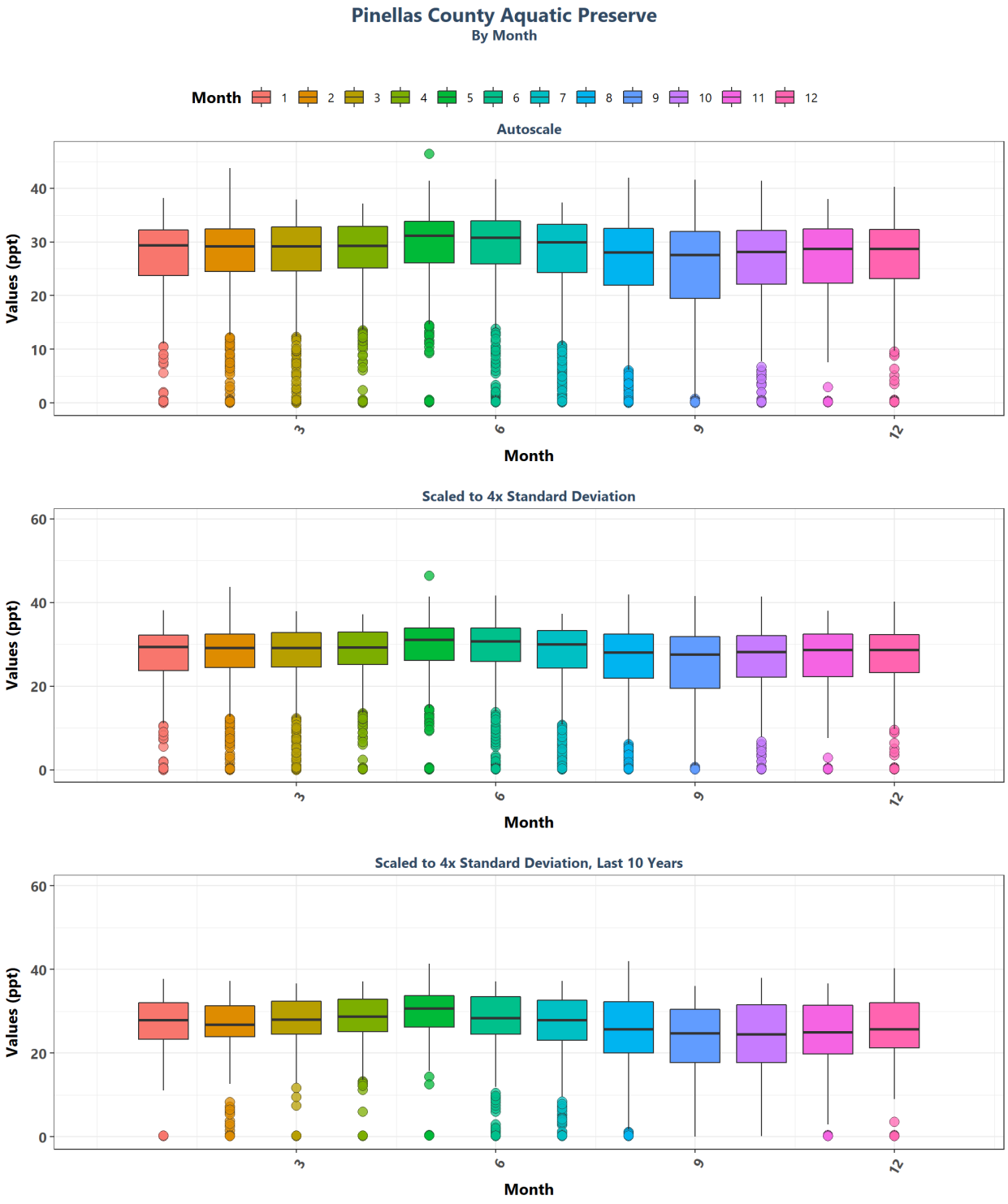



Pine Island Sound Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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

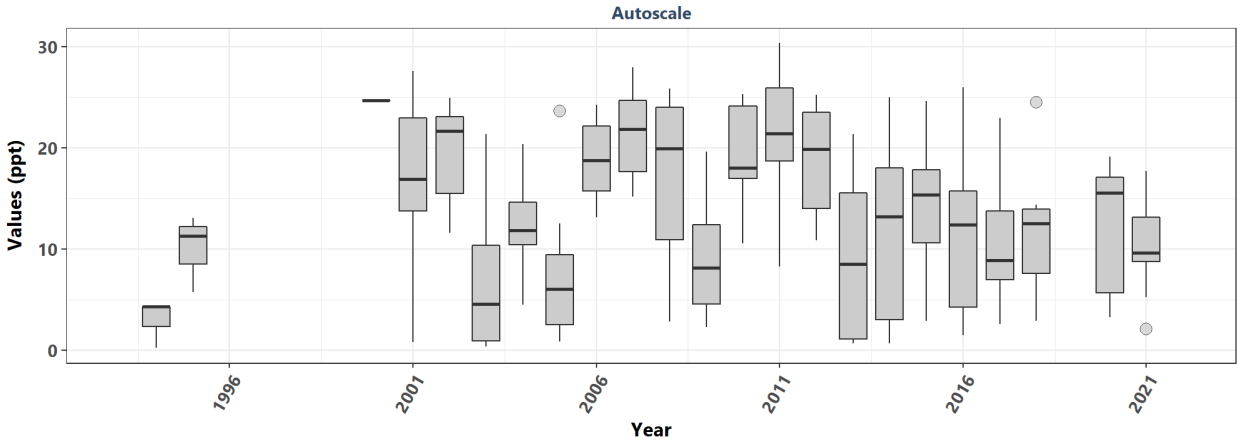

Pinellas County Aquatic Preserve
By Year
Autoscale
Values (ppt)
0
10
20
30
40
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
2011
2013
2015
2017
2019
2021
Year

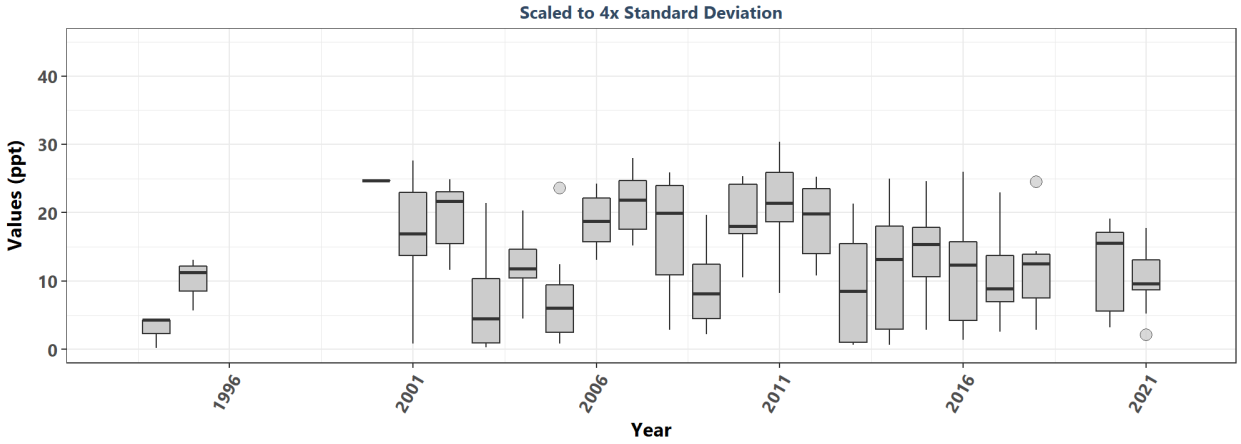

Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

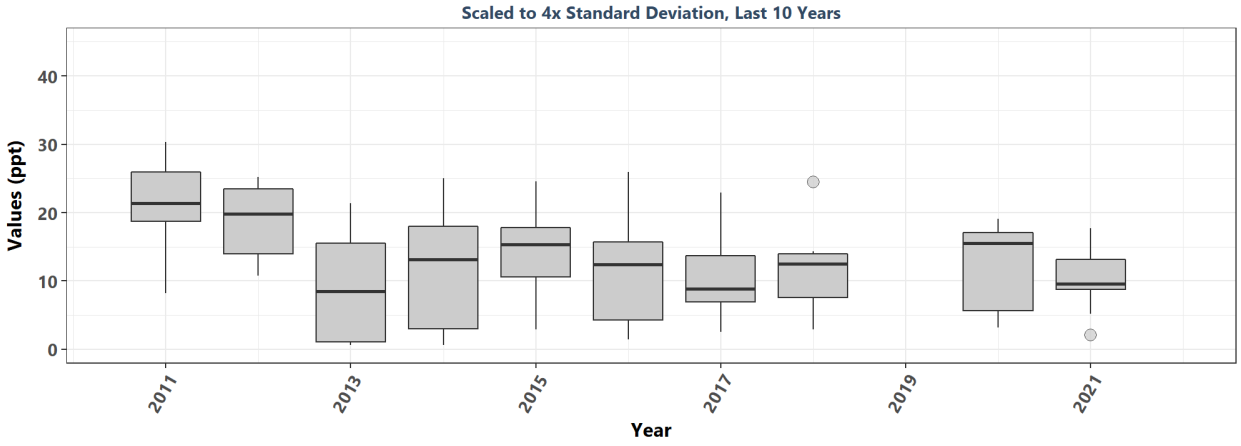

Pinellas County Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0 20 40 60
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0 20 40 60
3 6 9 12
Month

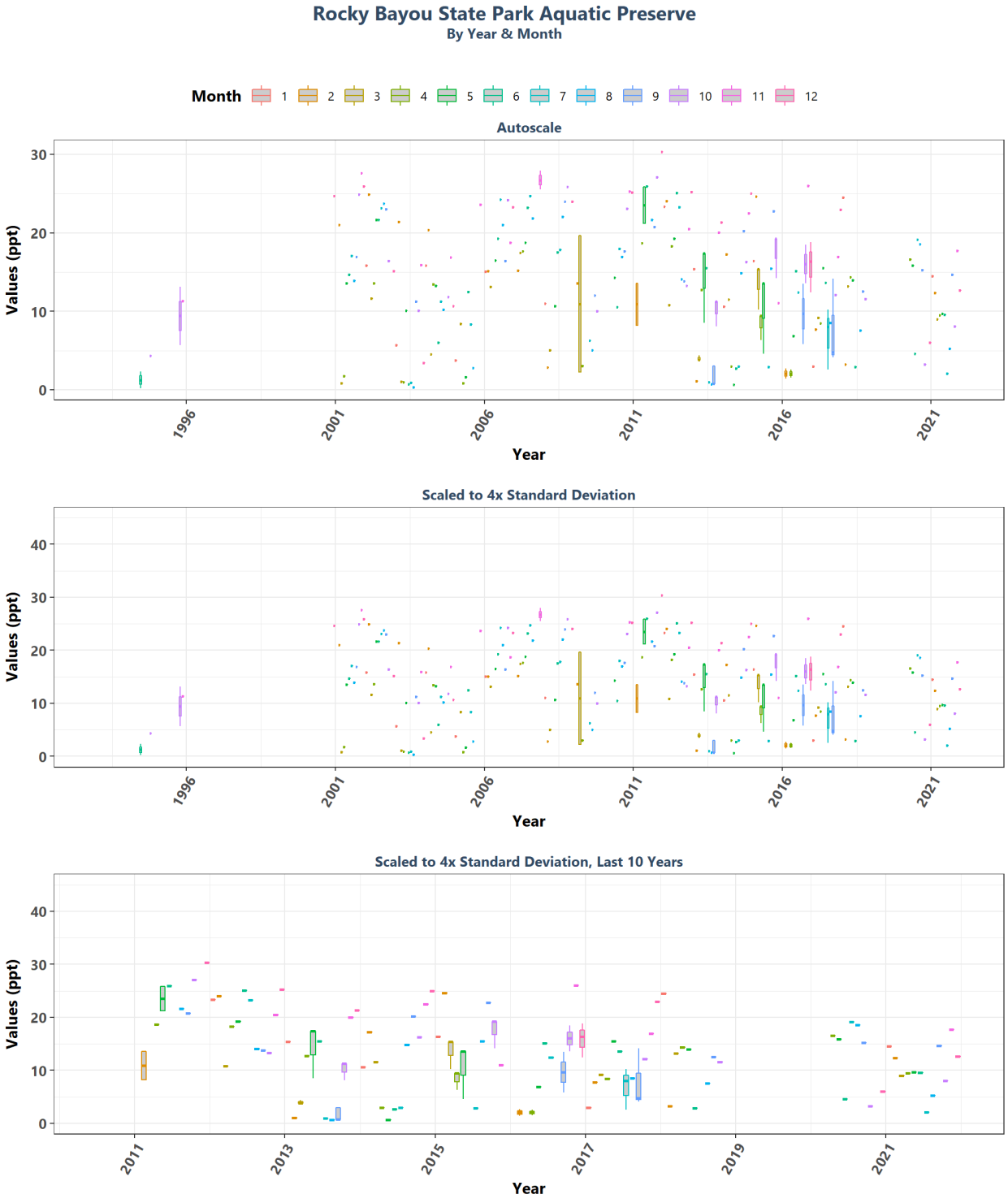
Rocky Bayou State Park Aquatic Preserve

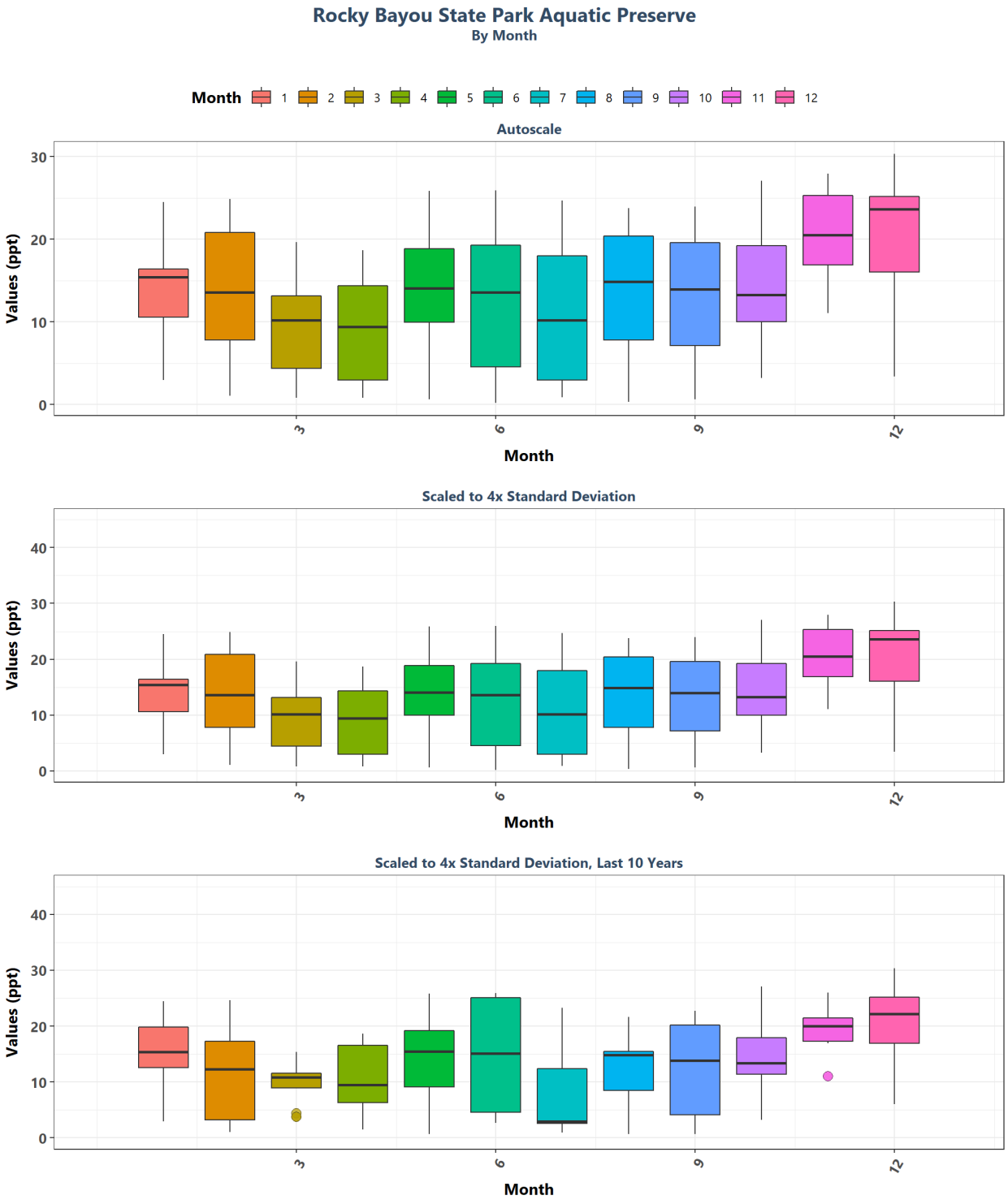
By Year





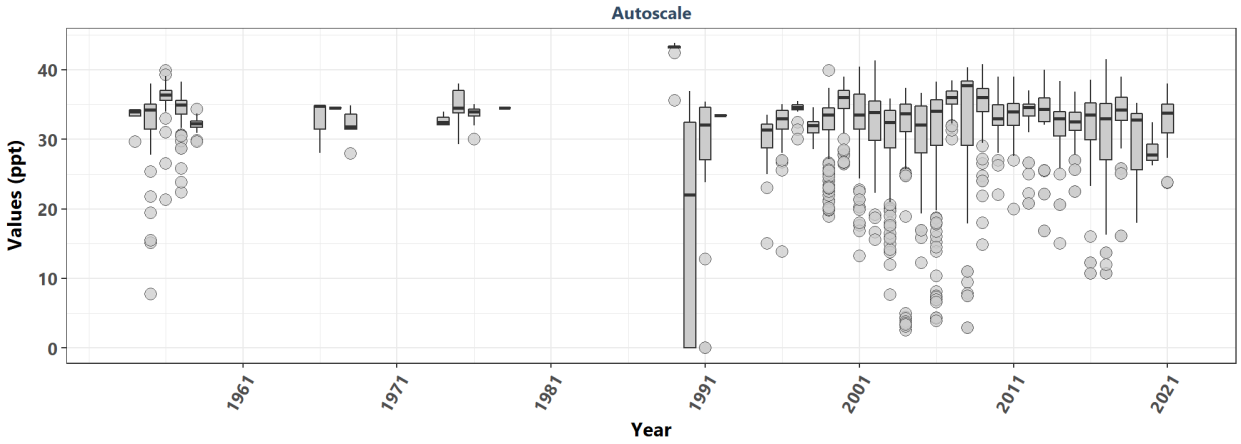


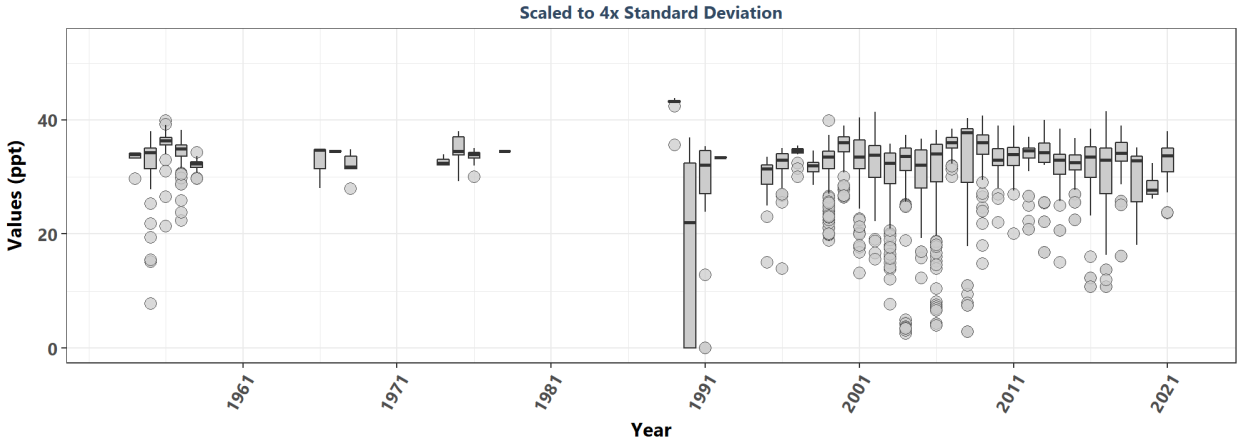

Rocky Bayou State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

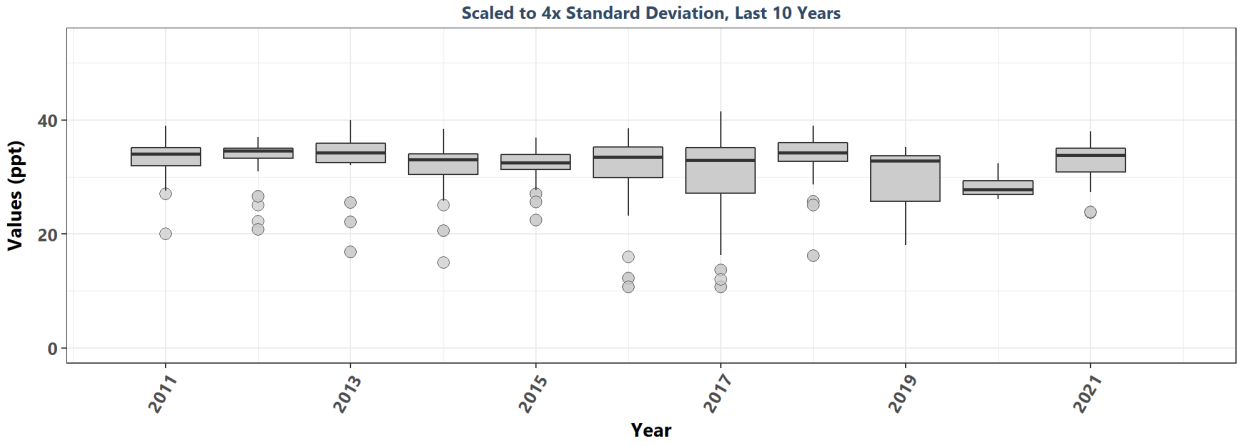

Rocky Bayou State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

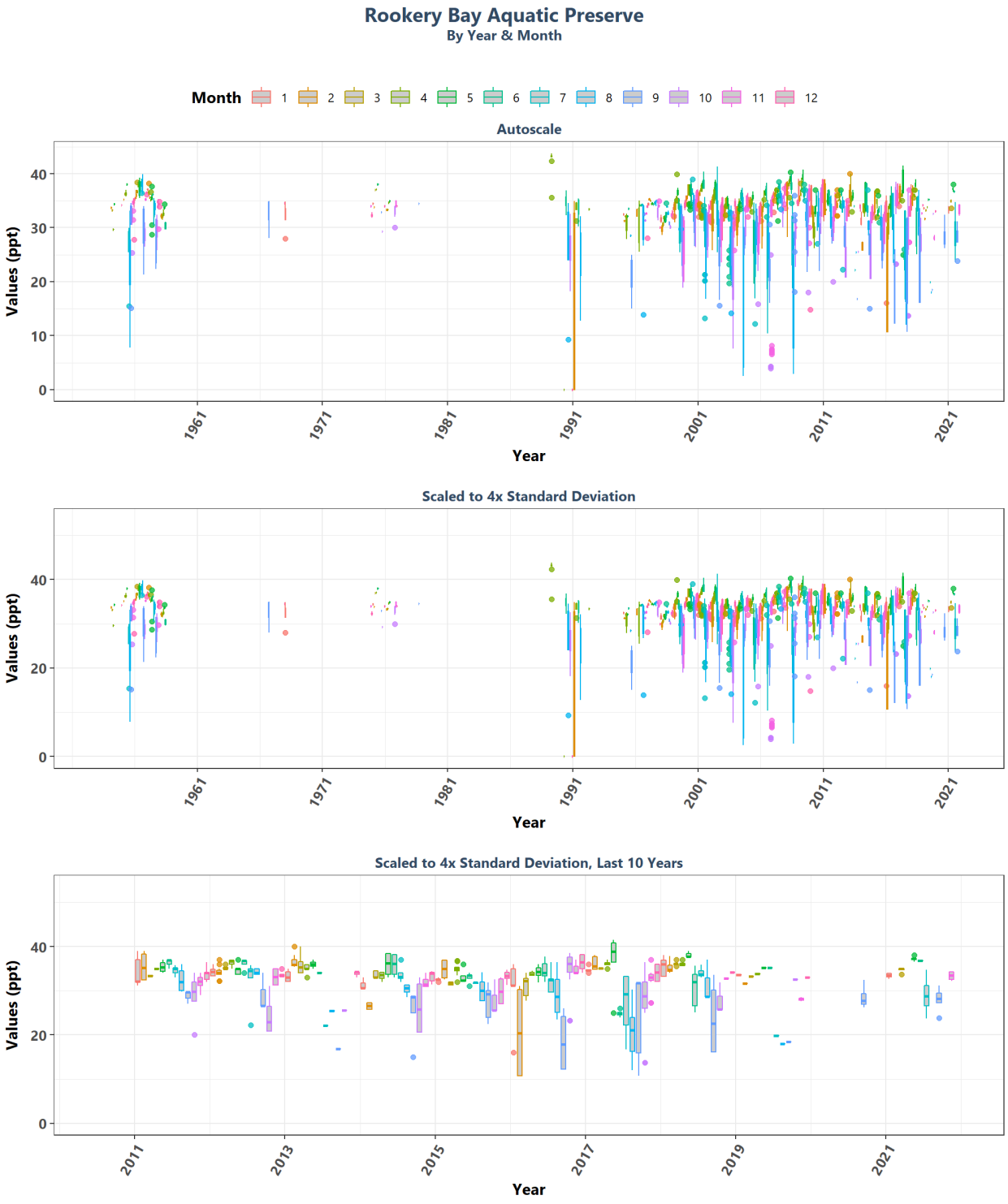
Rookery Bay Aquatic Preserve

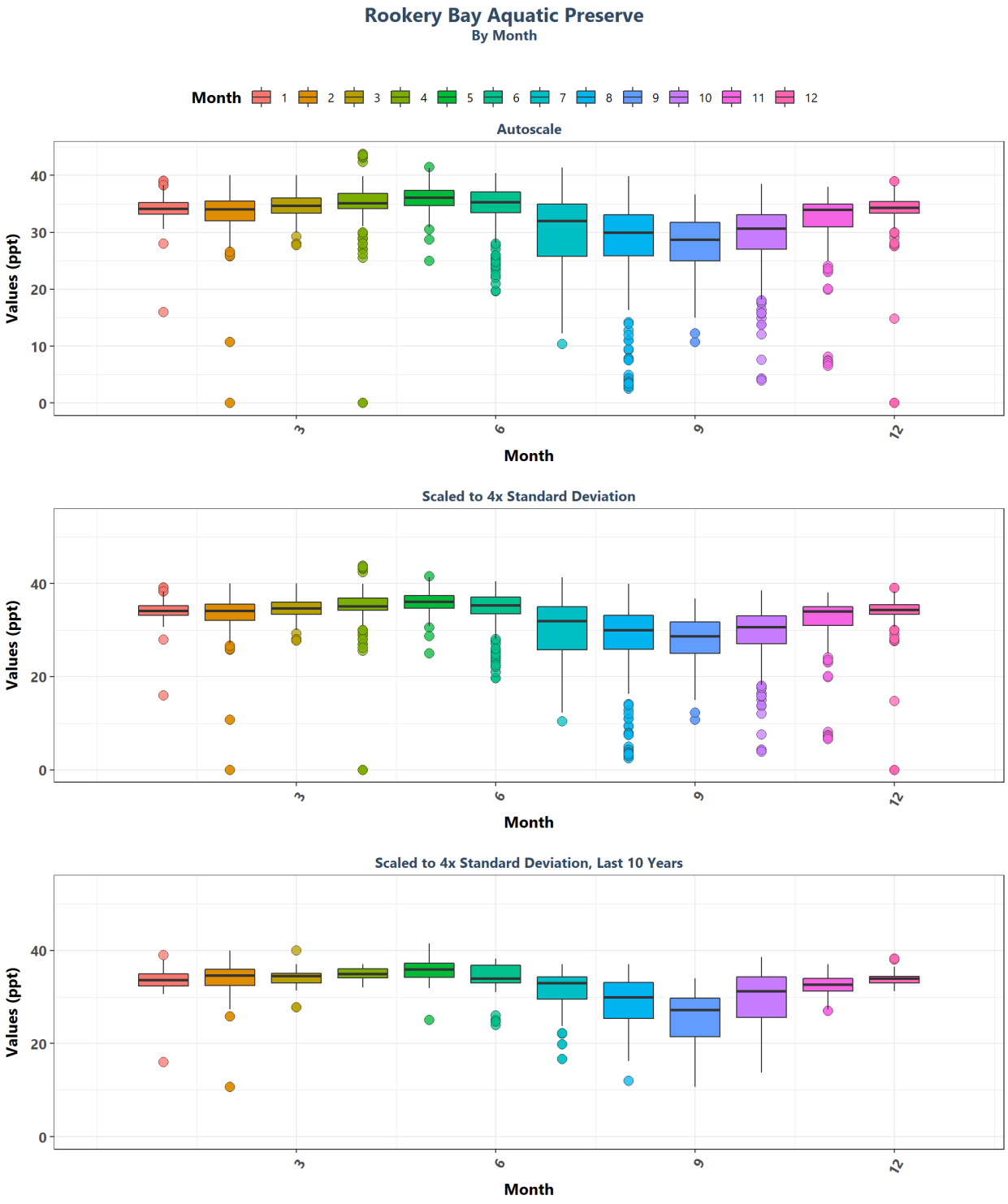
By Year

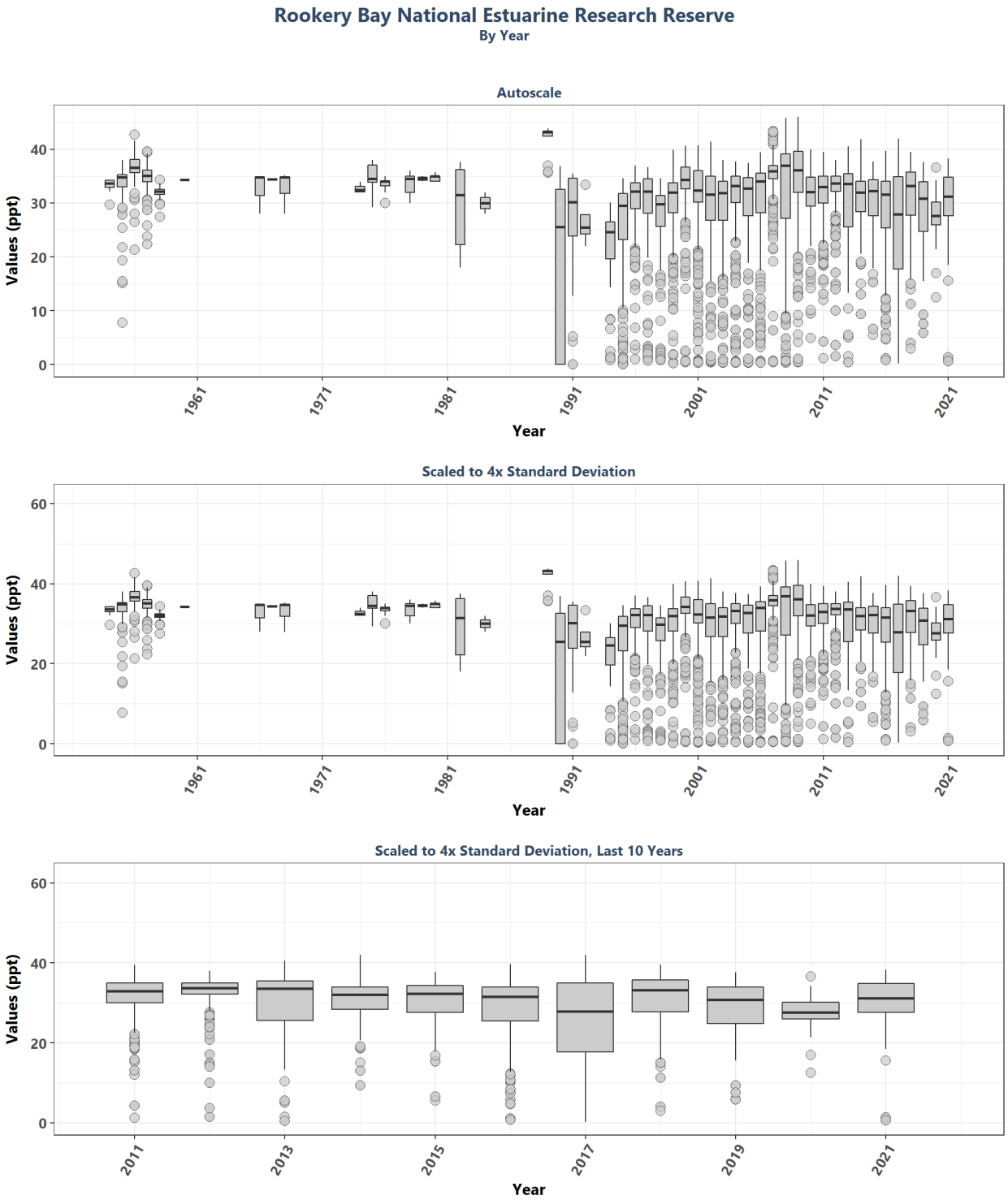


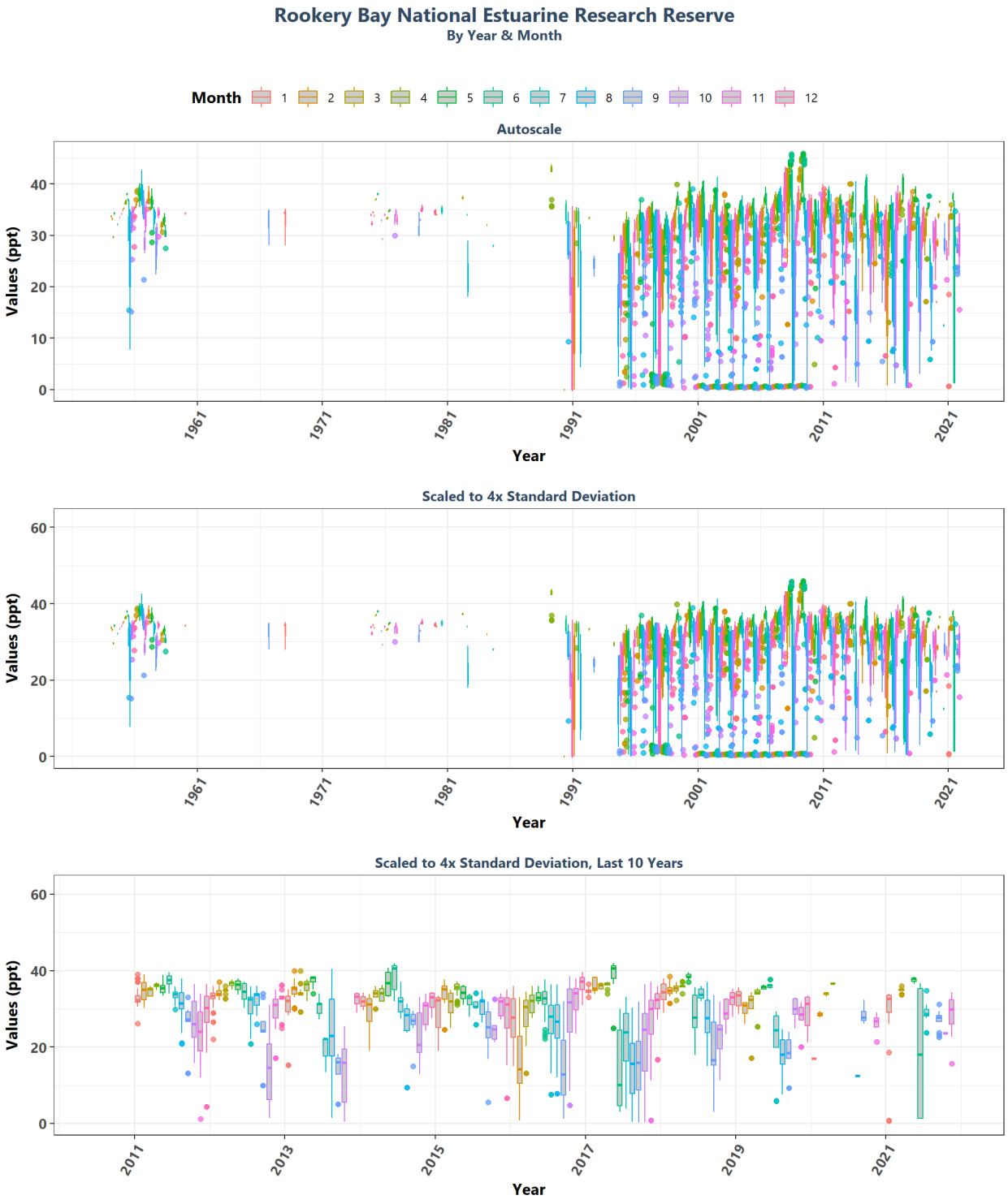


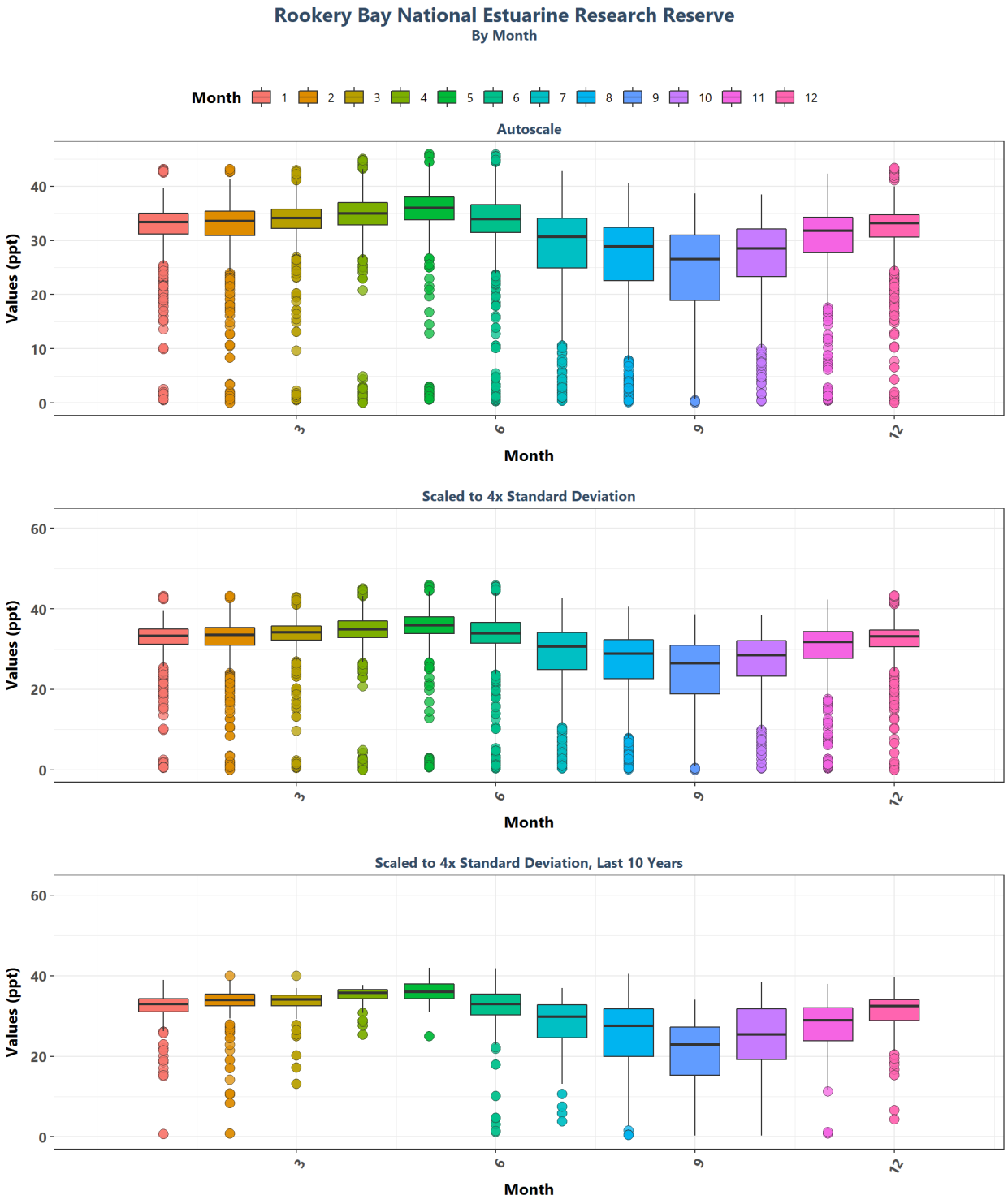



Rookery Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


Rookery Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
3
6
9
12
Month

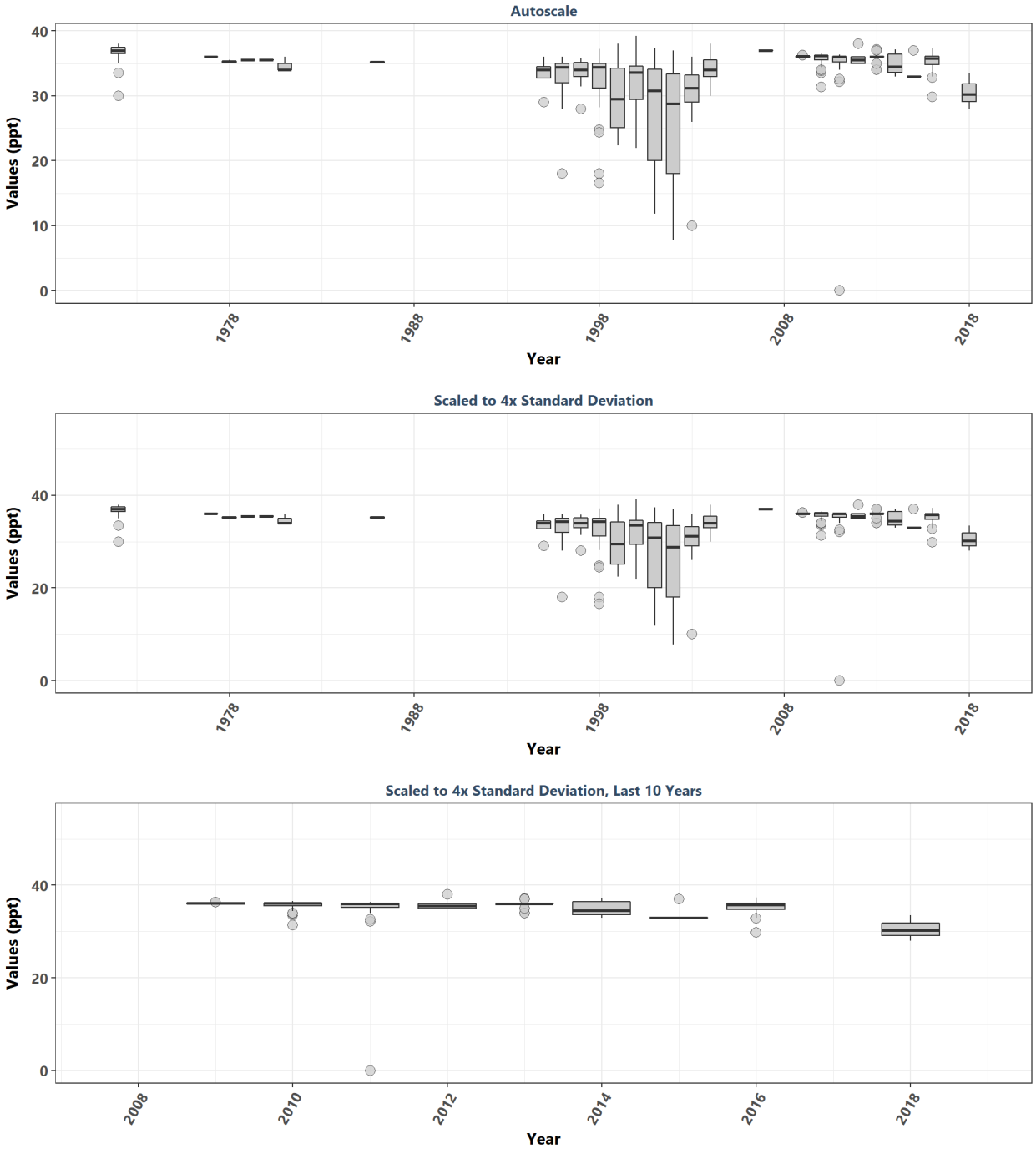

Rookery Bay National Estuarine Research Reserve
By Year
Autoscale
Values (ppt)
0
10
20
30
40
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
2011
2013
2015
2017
2019
2021
Year

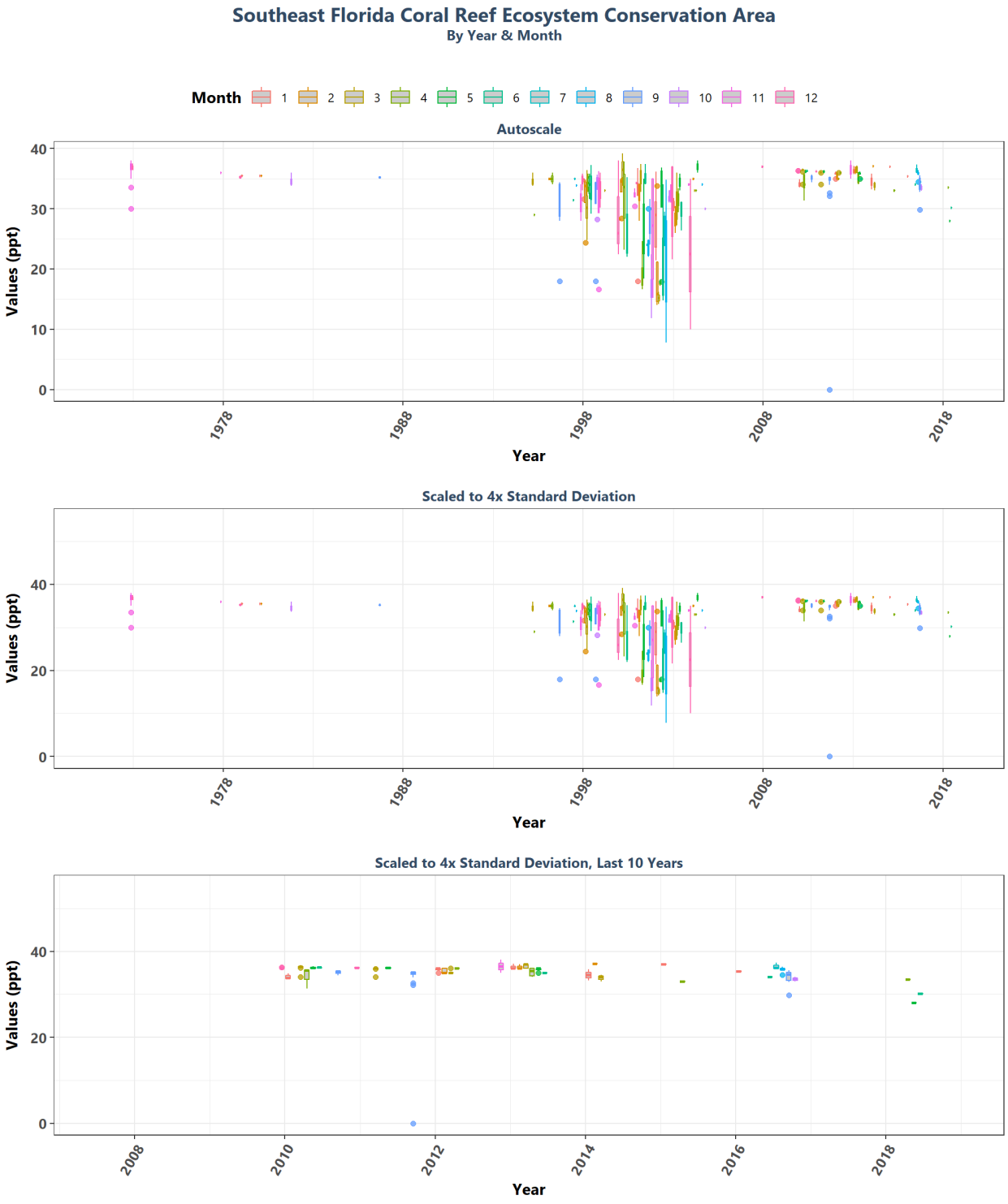

Rookery Bay National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

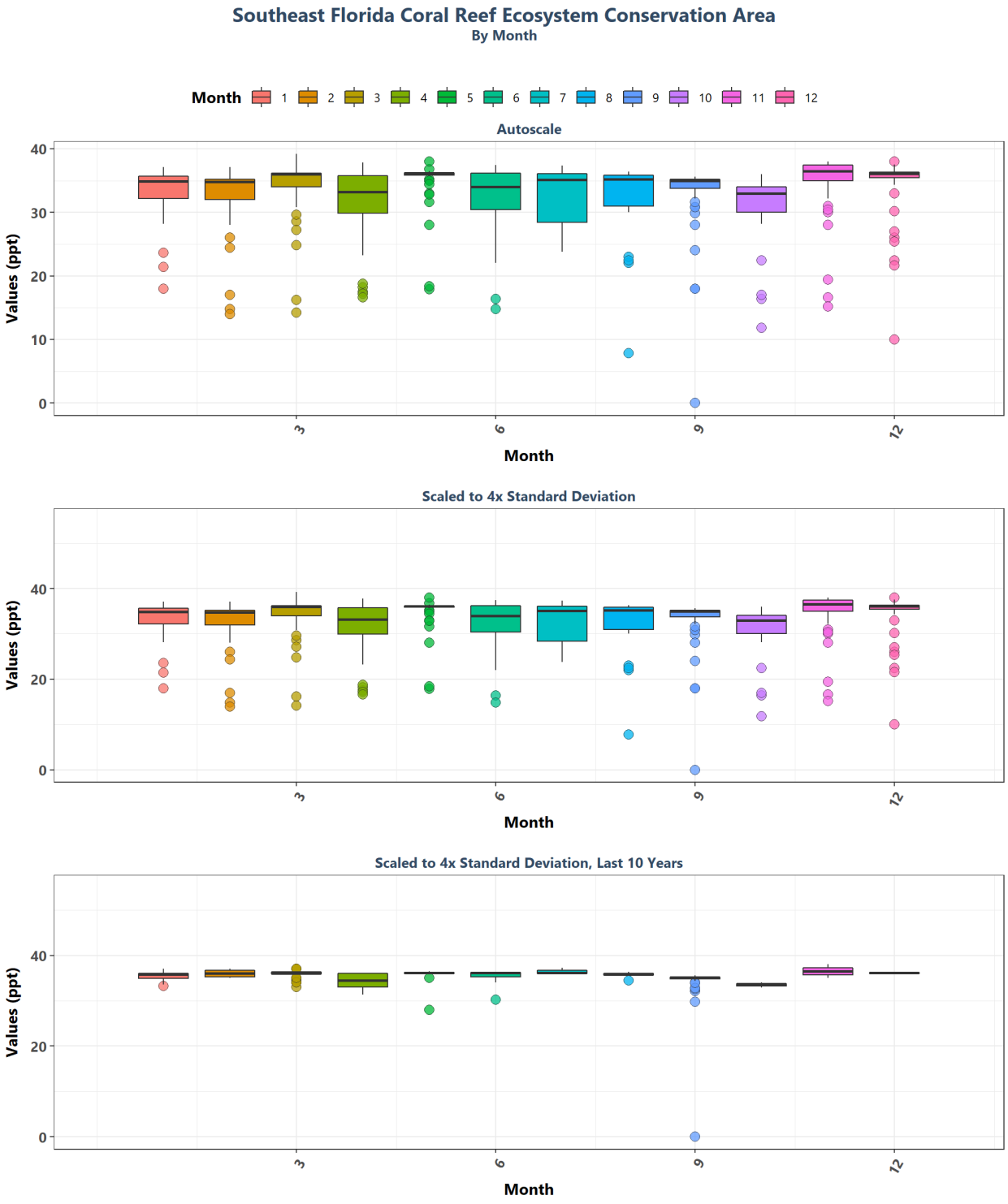

Rookery Bay National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0 20 40 60
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0 20 40 60
3 6 9 12
Month

Southeast Florida Coral Reef Ecosystem Conservation Area

By Year

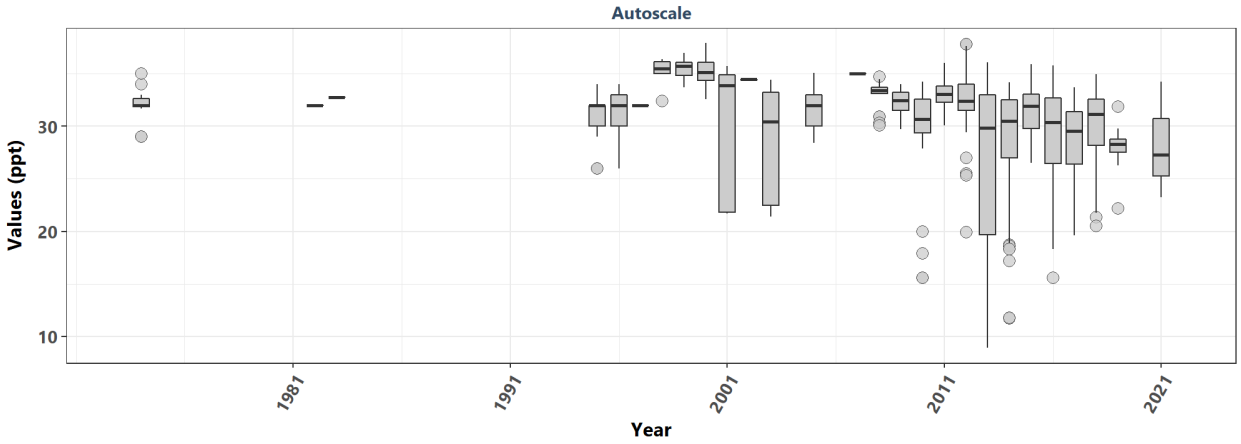


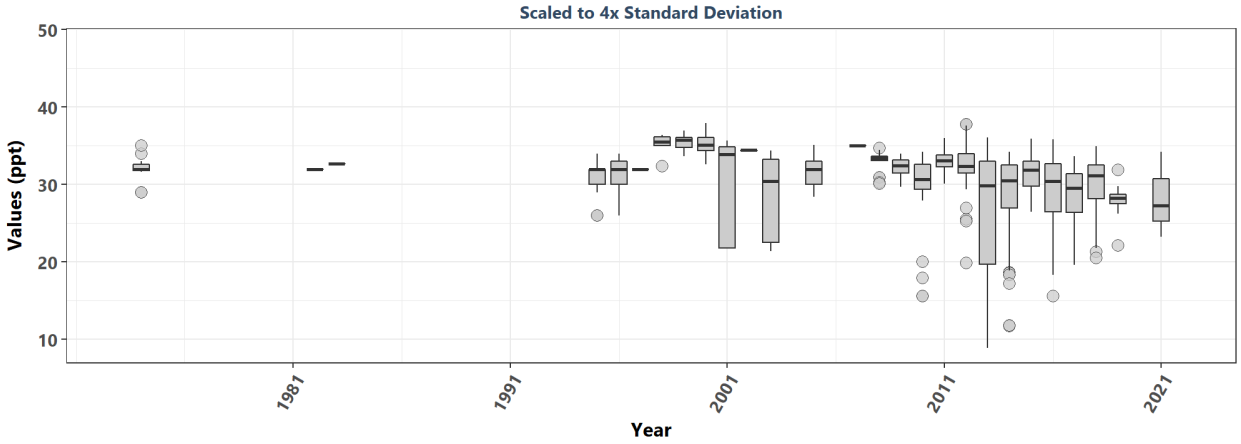

Southeast Florida Coral Reef Ecosystem Conservation Area
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

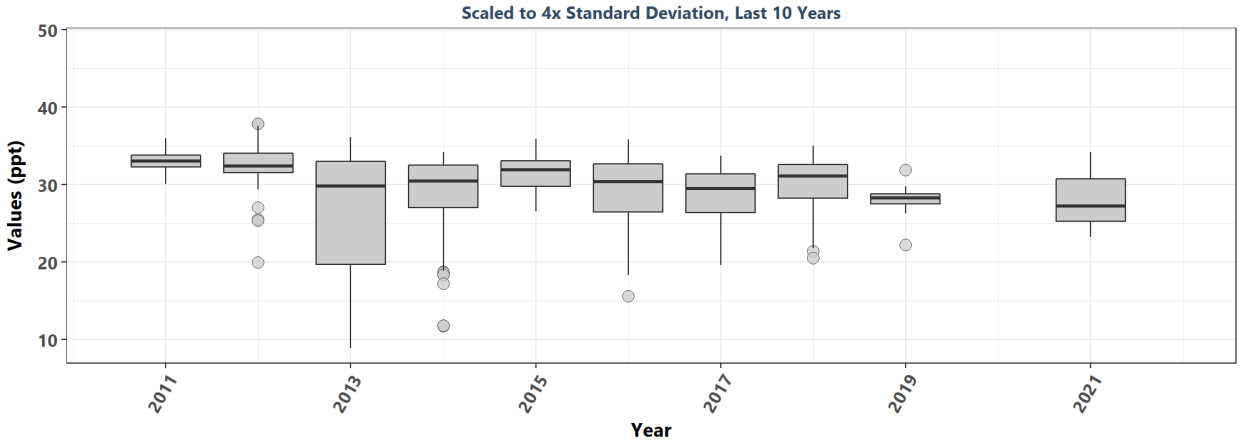

Southeast Florida Coral Reef Ecosystem Conservation Area
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

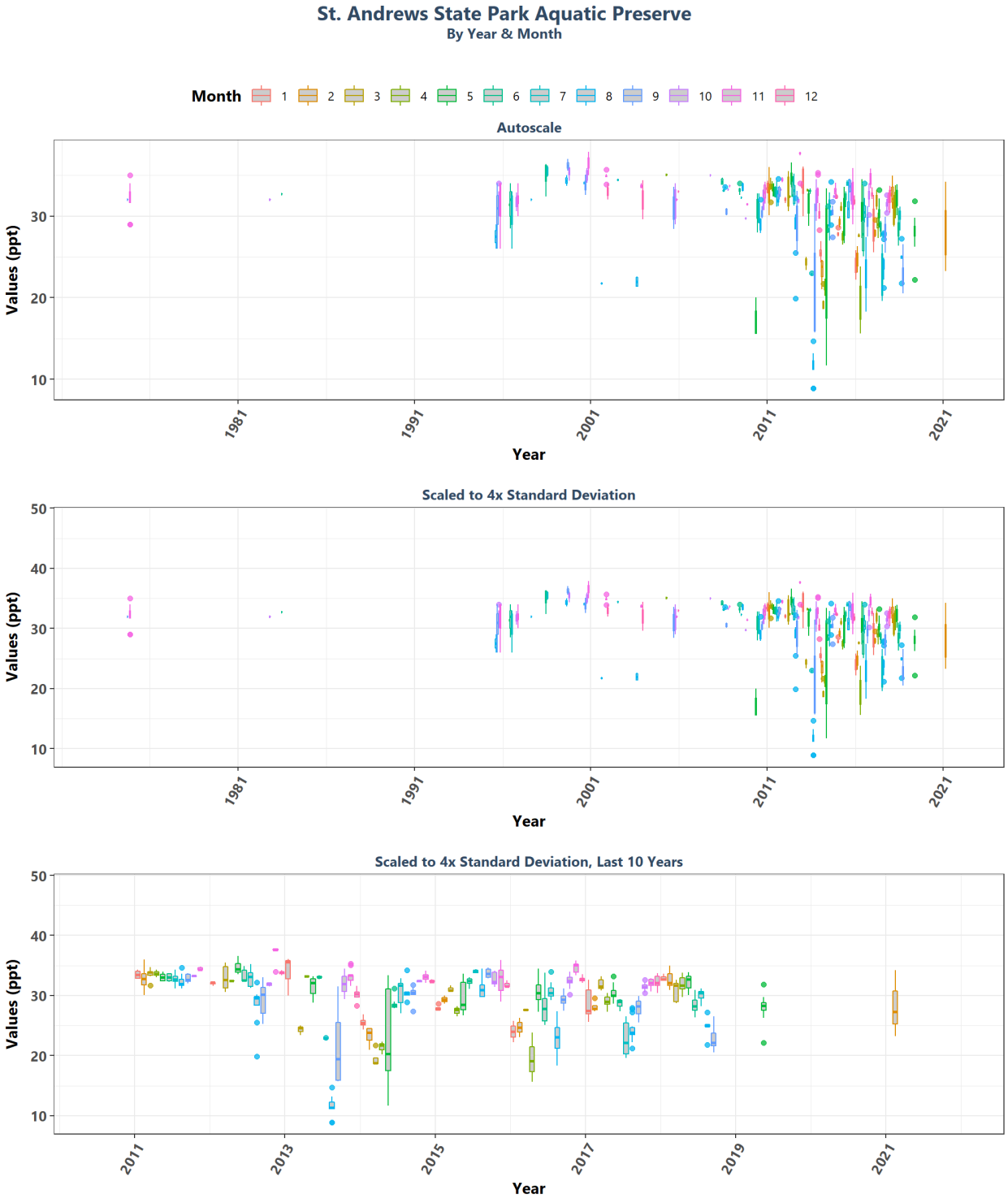
St. Andrews State Park Aquatic Preserve

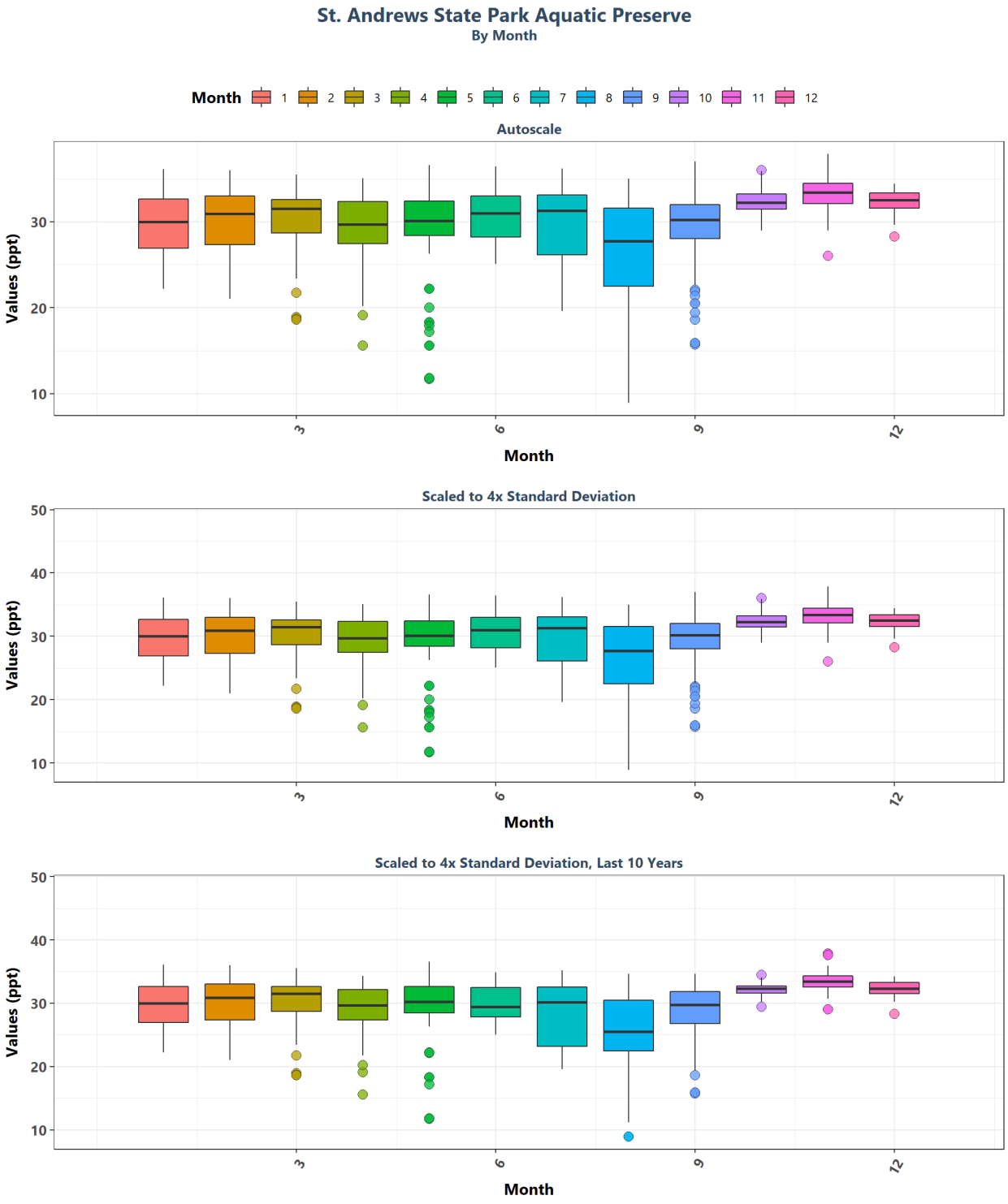
By Year





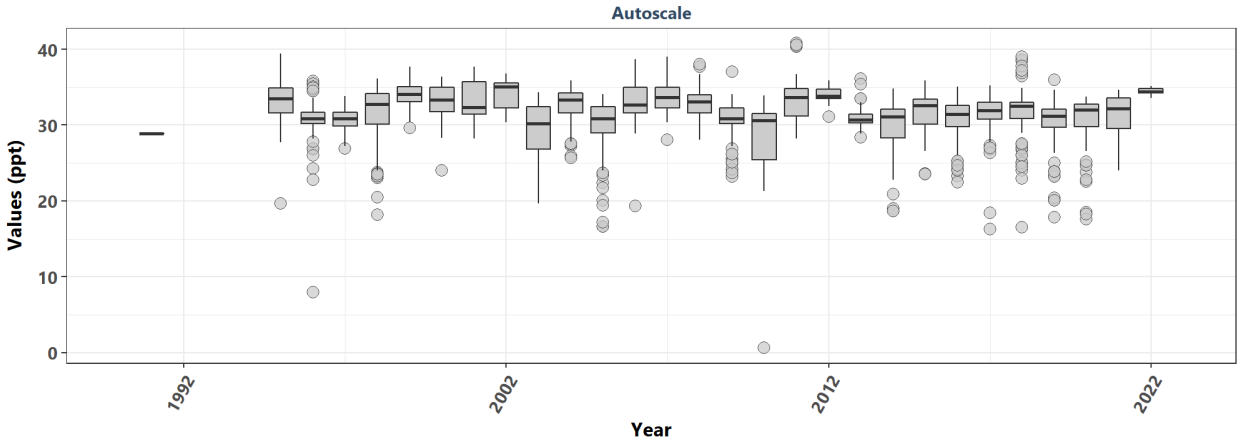


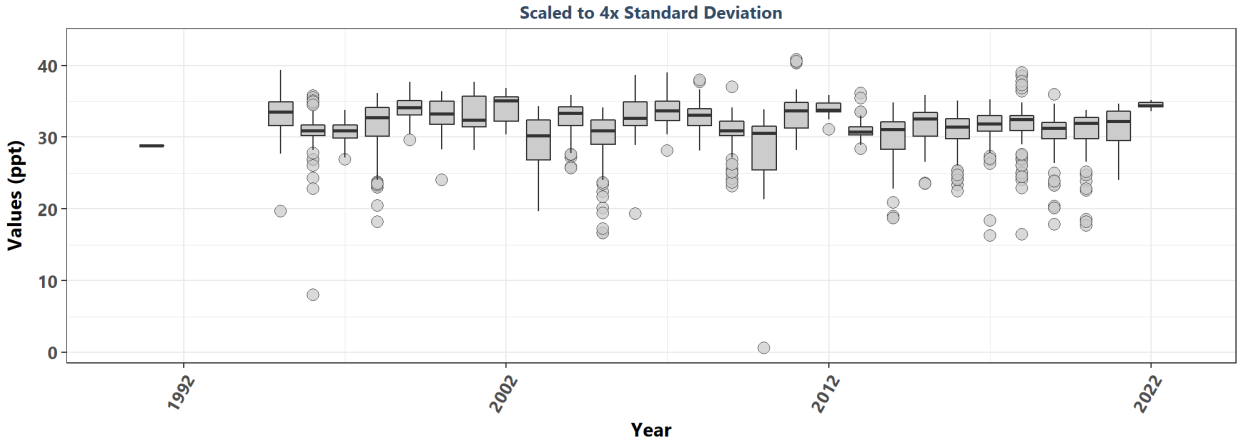

St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
10
20
30
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
10
20
30
40
50
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
10
20
30
40
50
2011
2013
2015
2017
2019
2021
Year

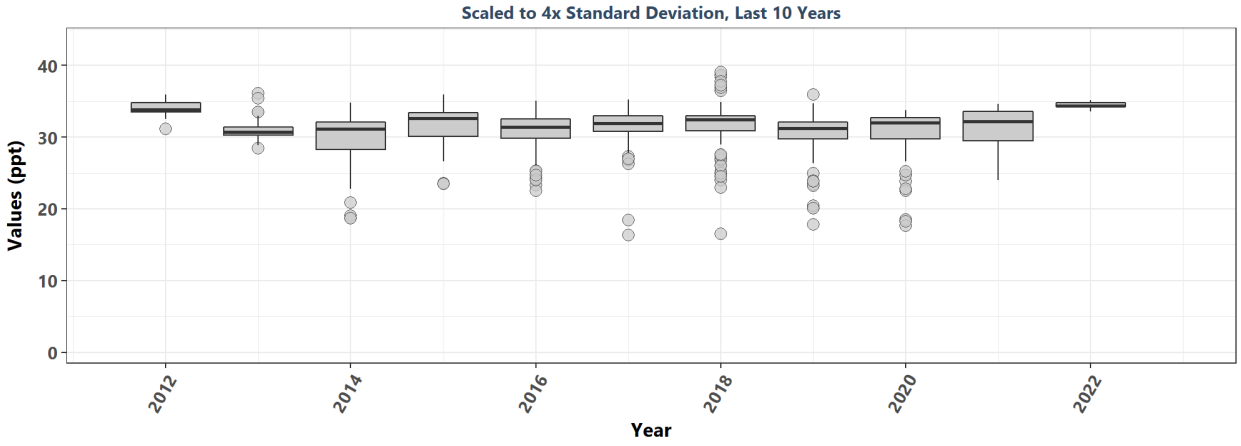

St. Andrews State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Month
Scaled to 4x Standard Deviation
Values (ppt)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Month

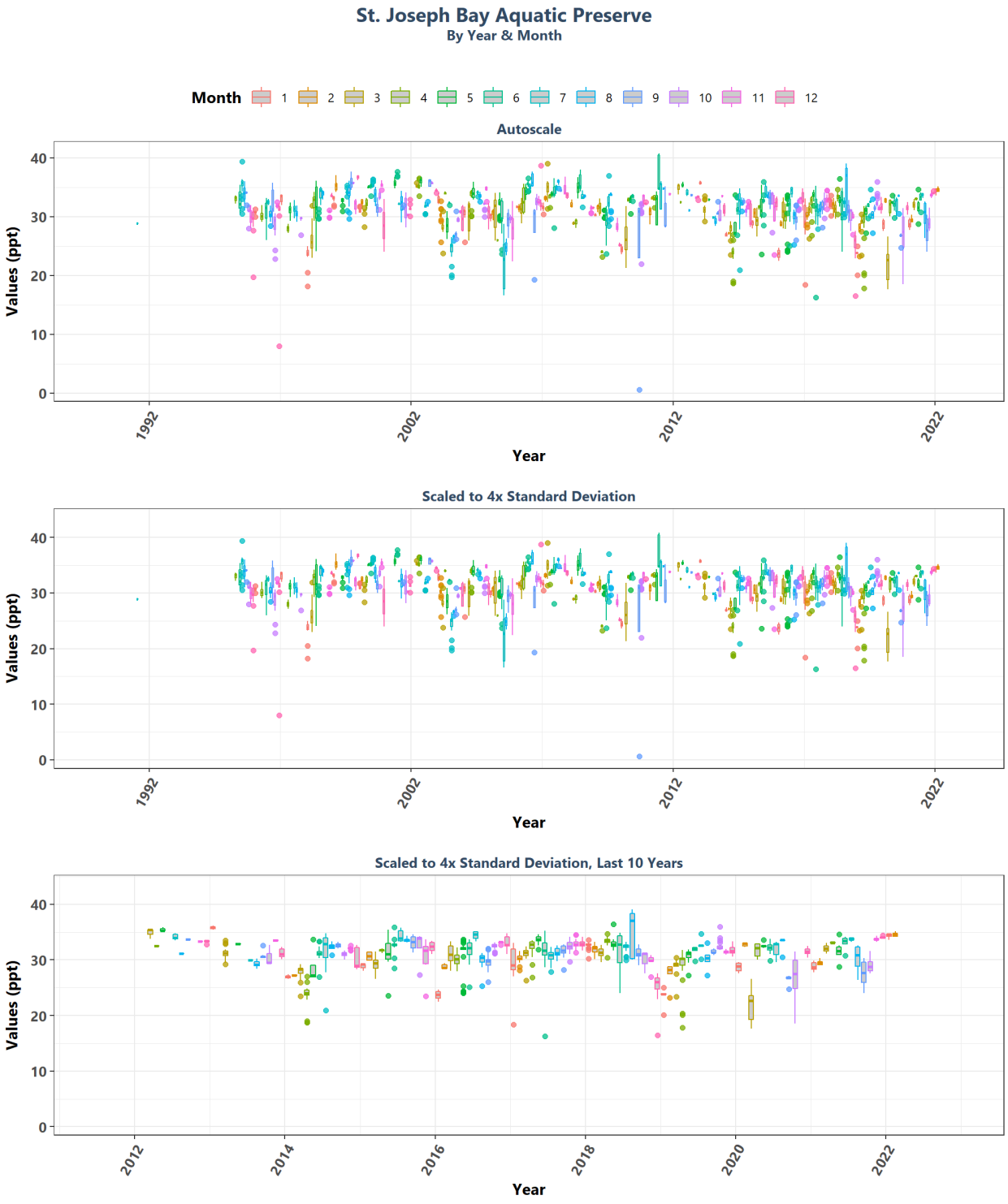
St. Joseph Bay Aquatic Preserve

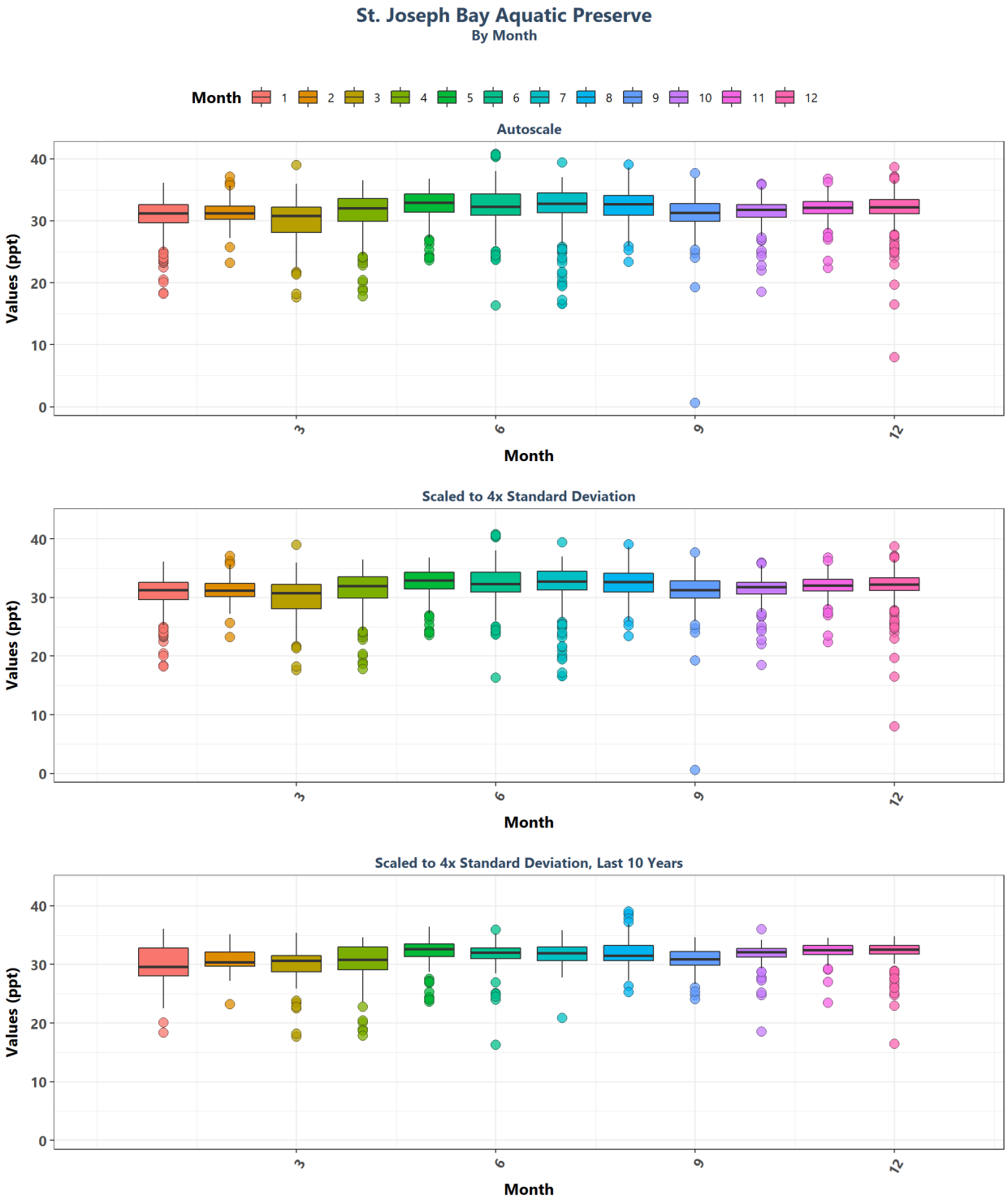
By Year

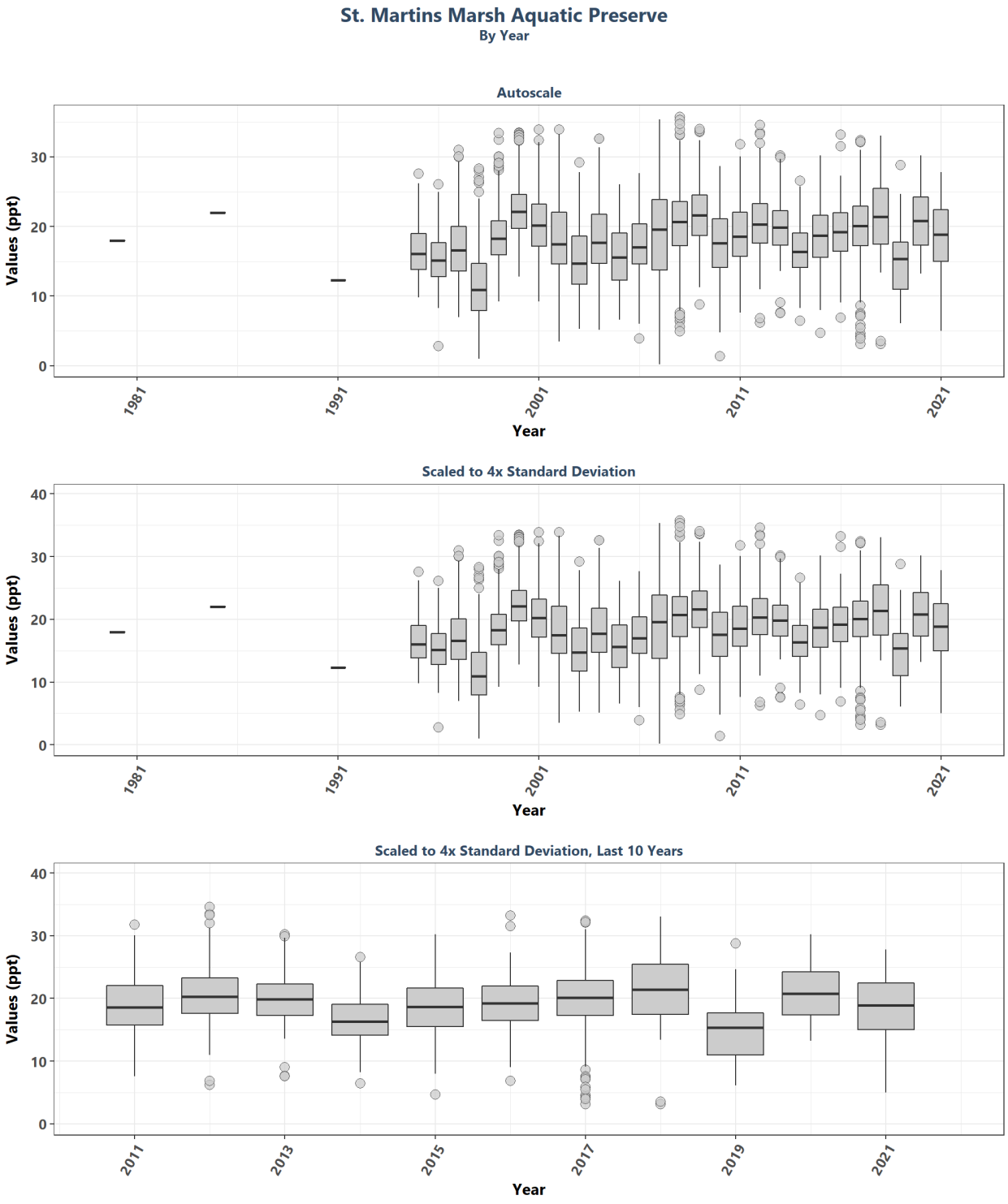


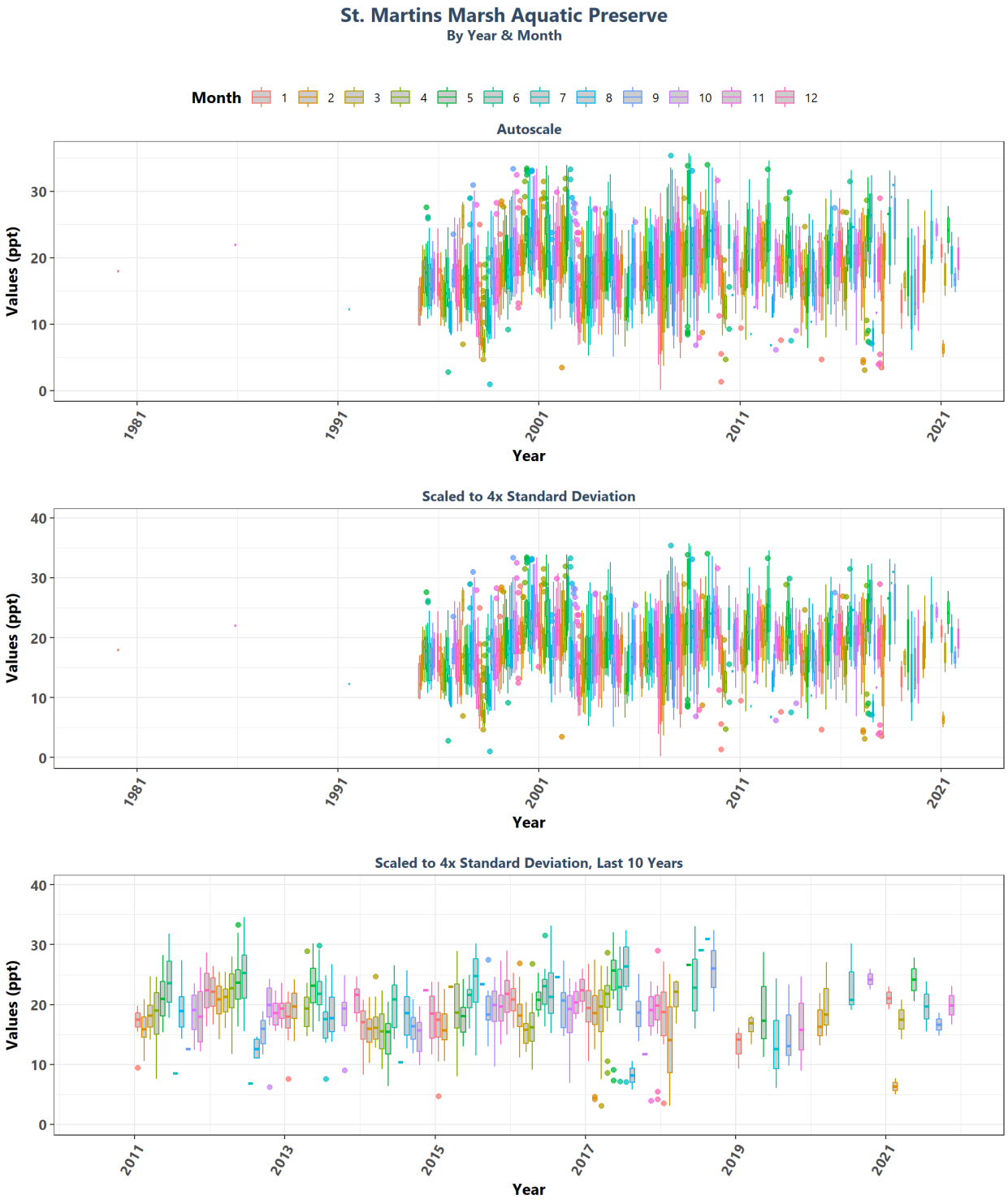


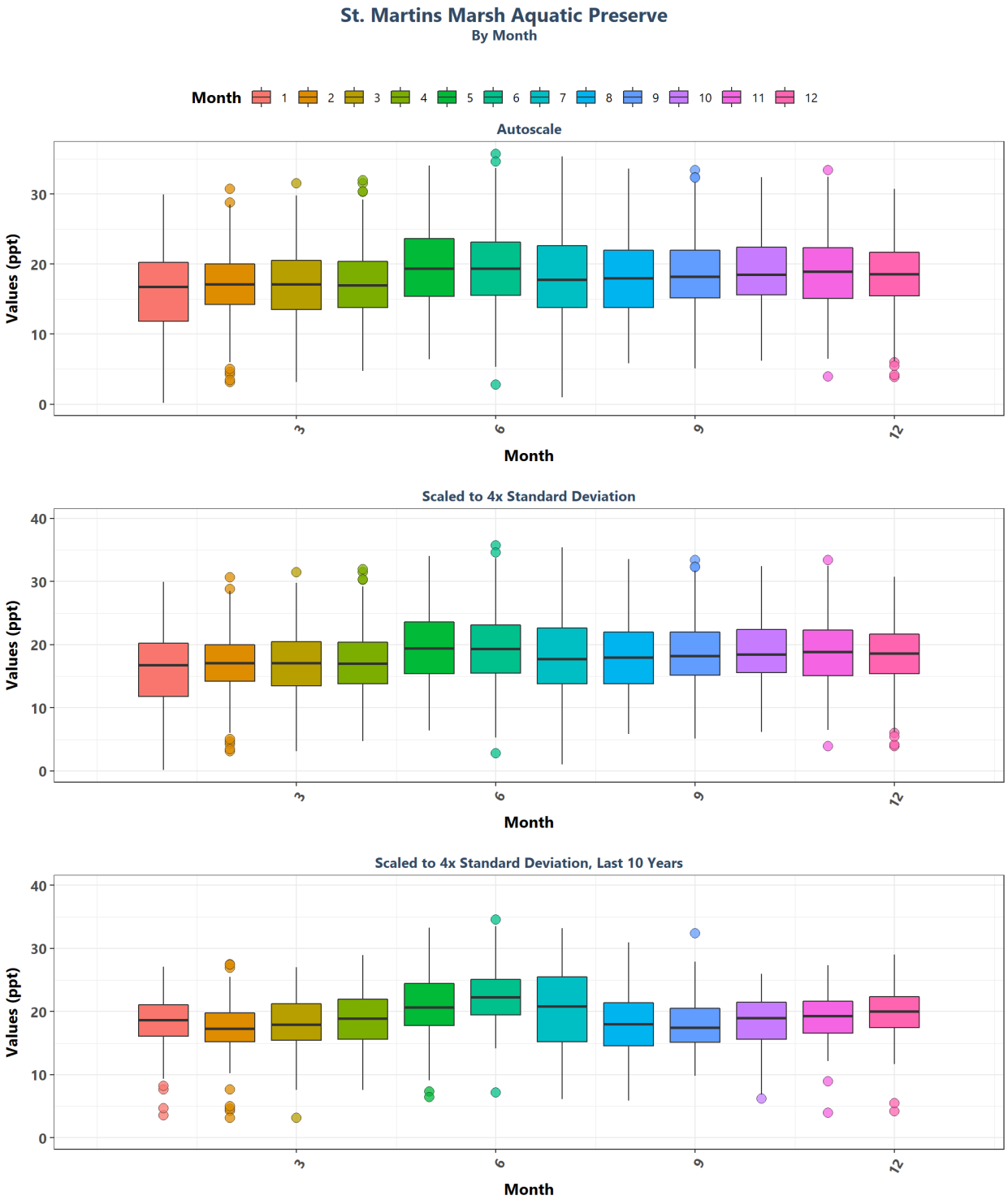



St. Joseph Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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

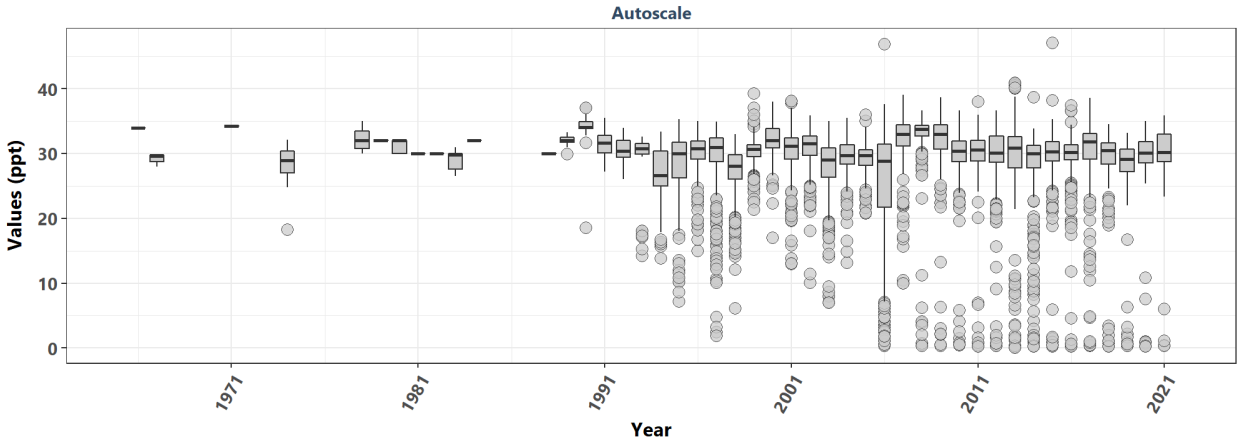

St. Martins Marsh Aquatic Preserve
By Year
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

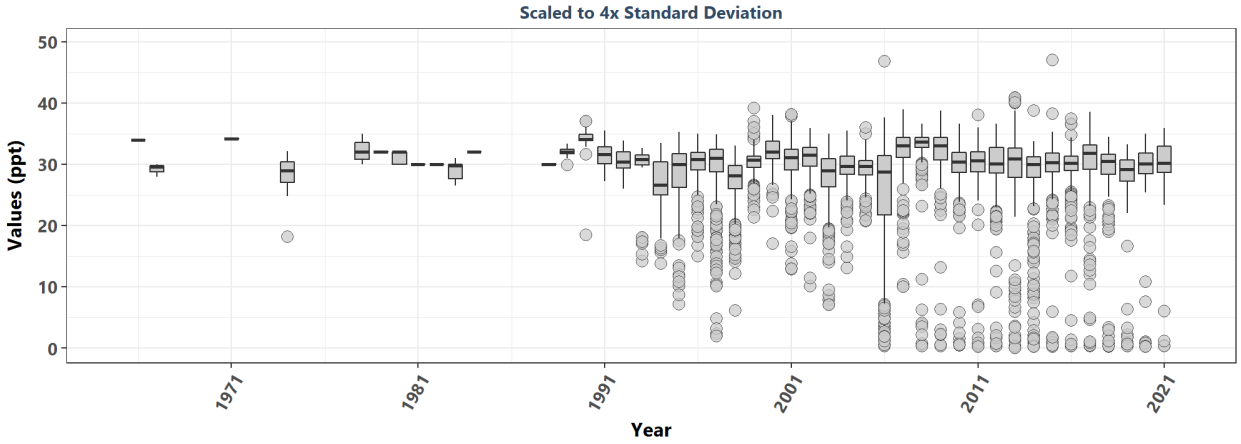

St. Martins Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

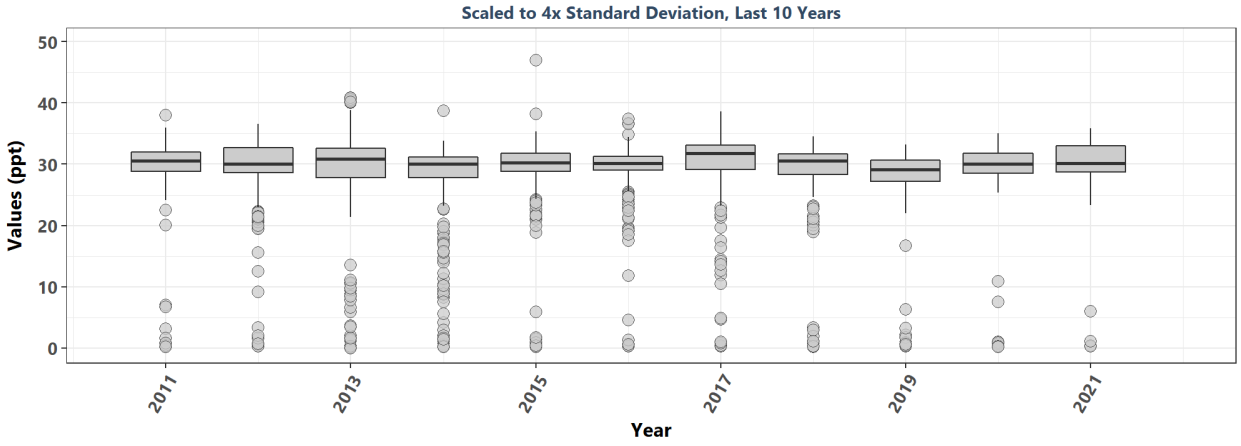

St. Martins Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
0
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (ppt)
0
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
10
20
30
40
3
6
9
12
Month

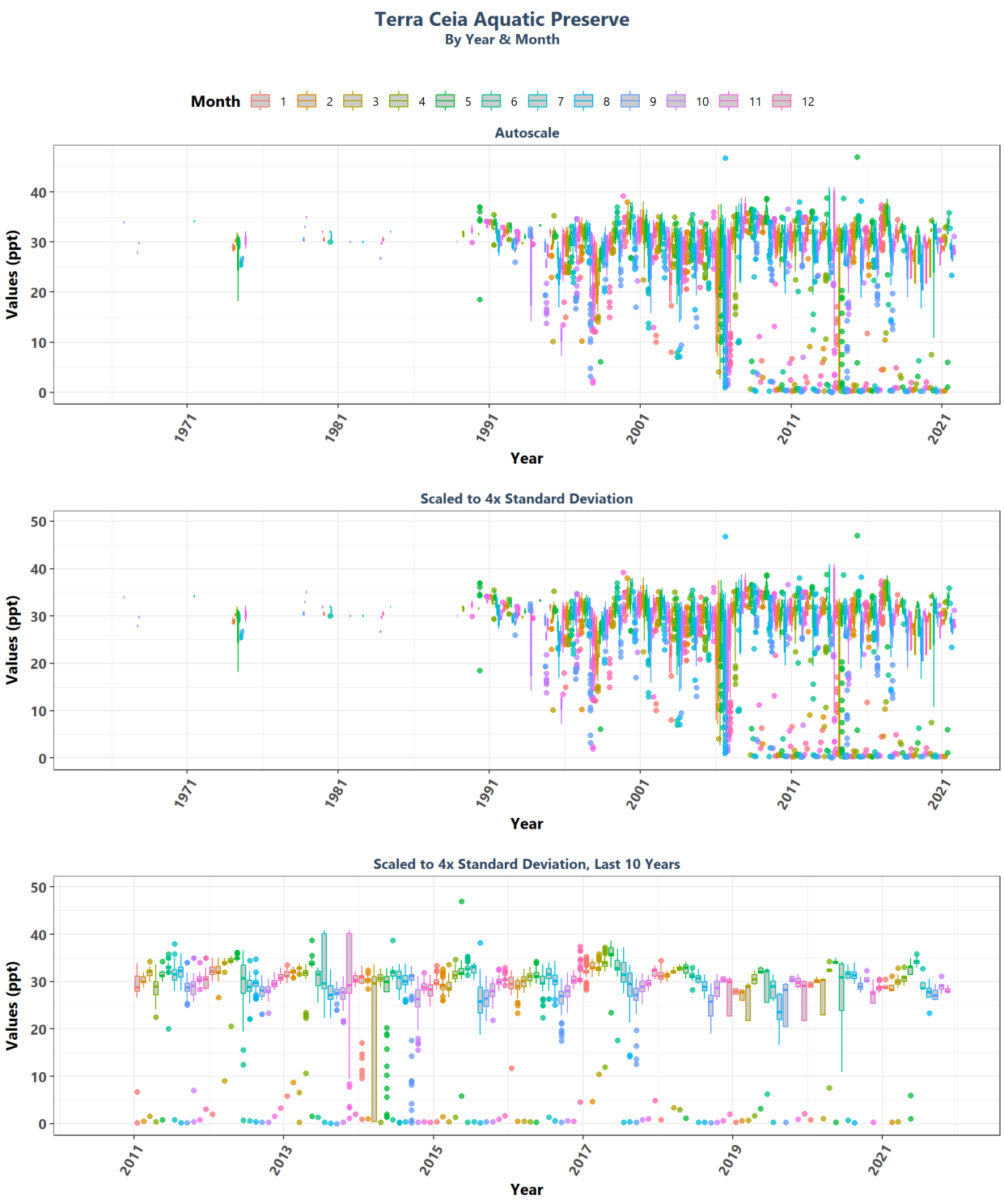
Terra Ceia Aquatic Preserve

By Year

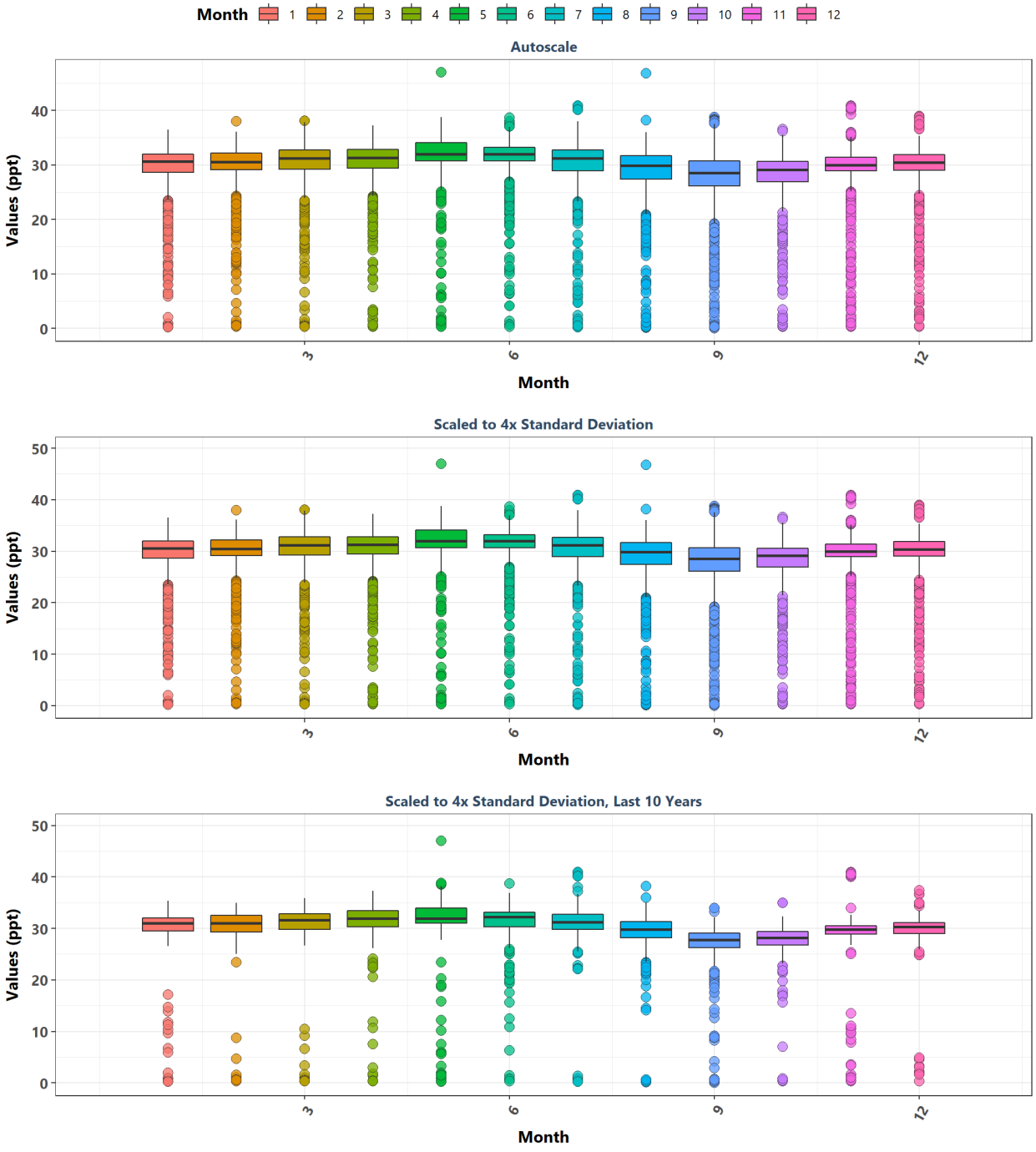






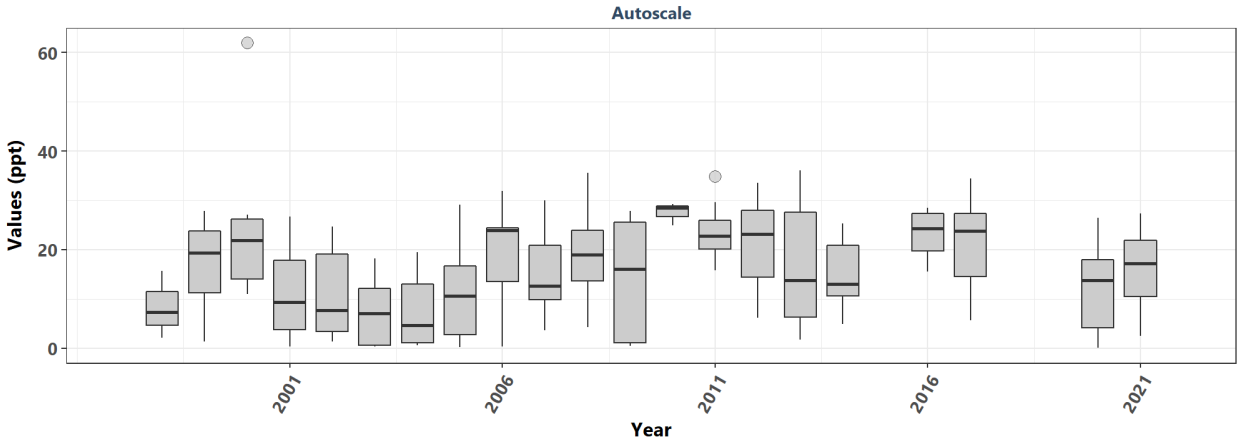

Terra Ceia Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

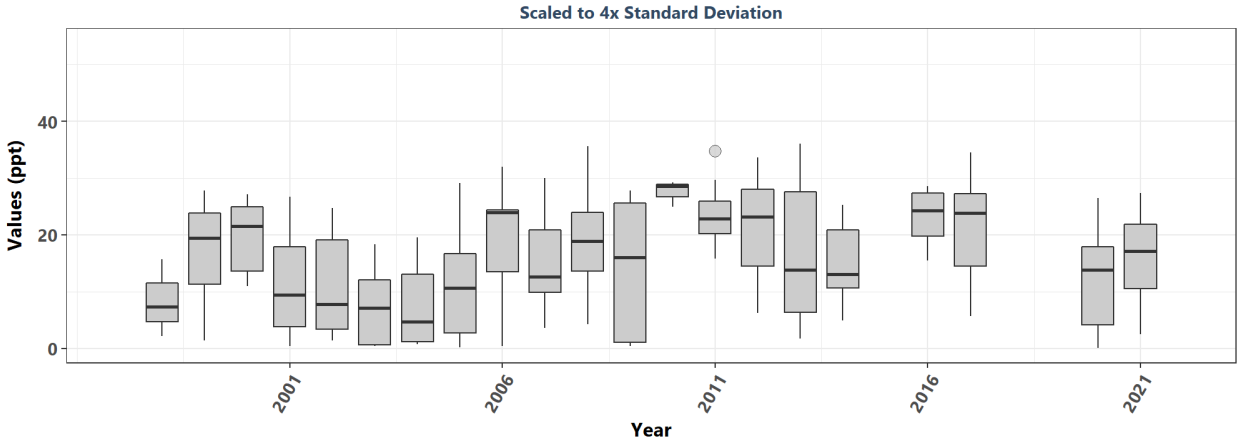
Terra Ceia Aquatic Preserve
By Month

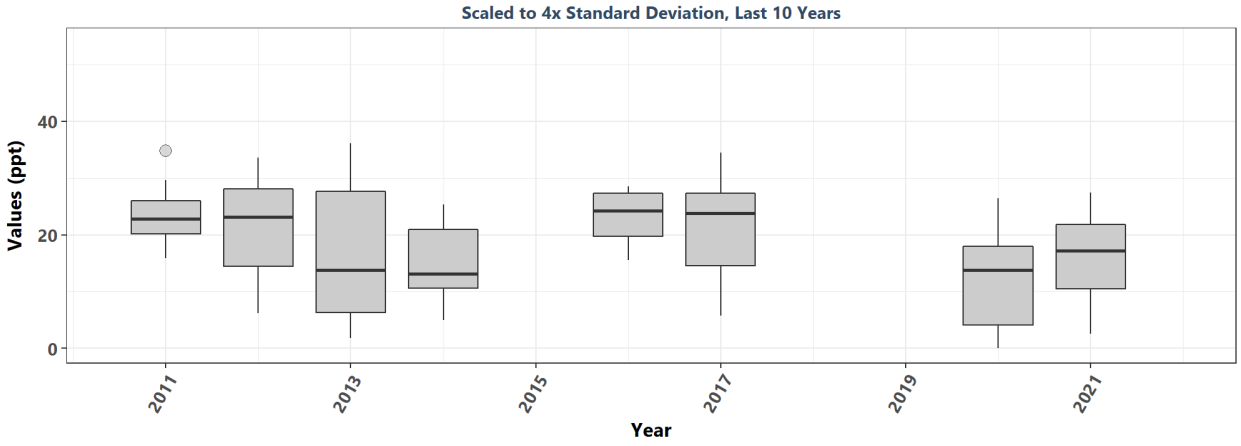


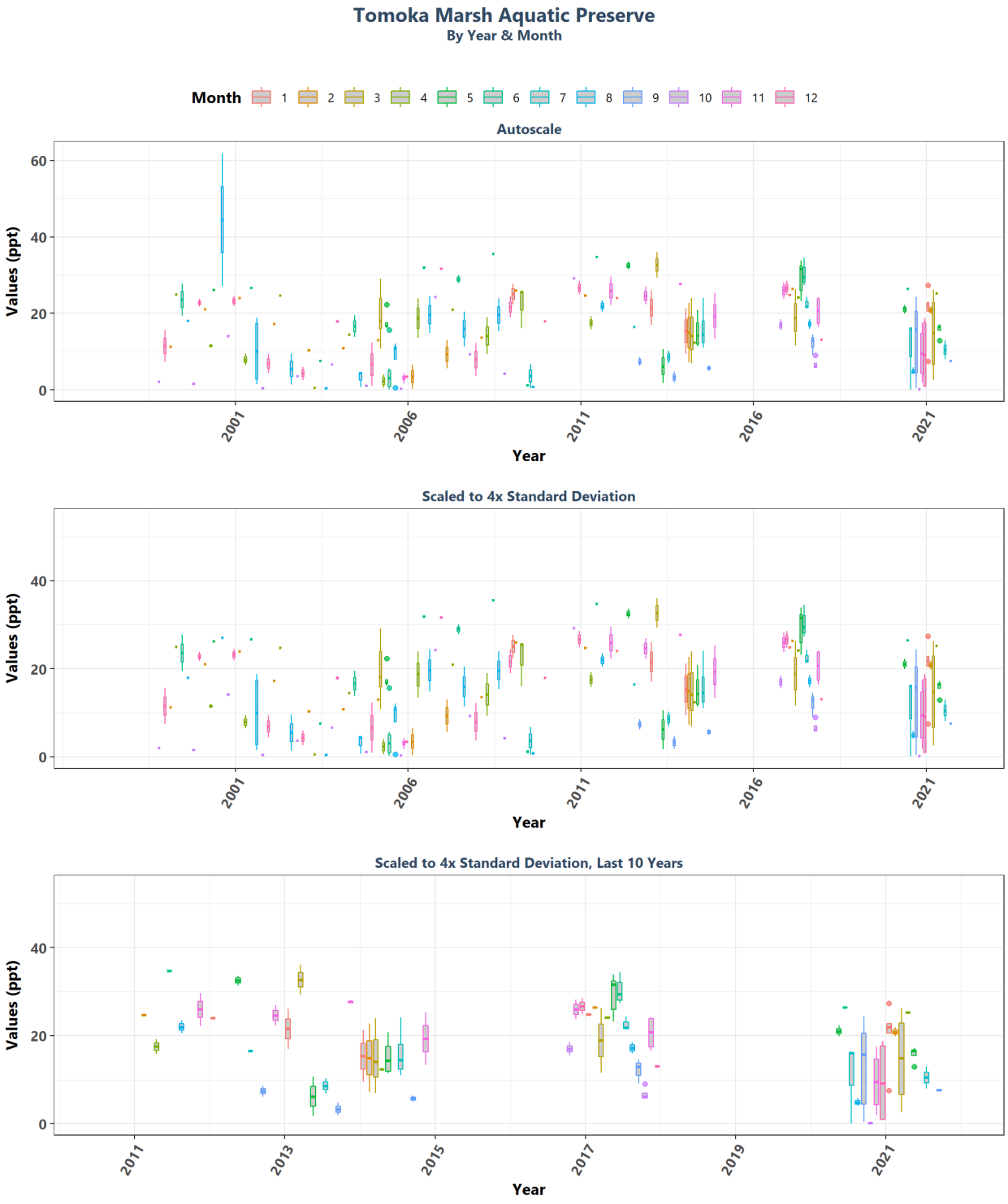
Tomoka Marsh Aquatic Preserve

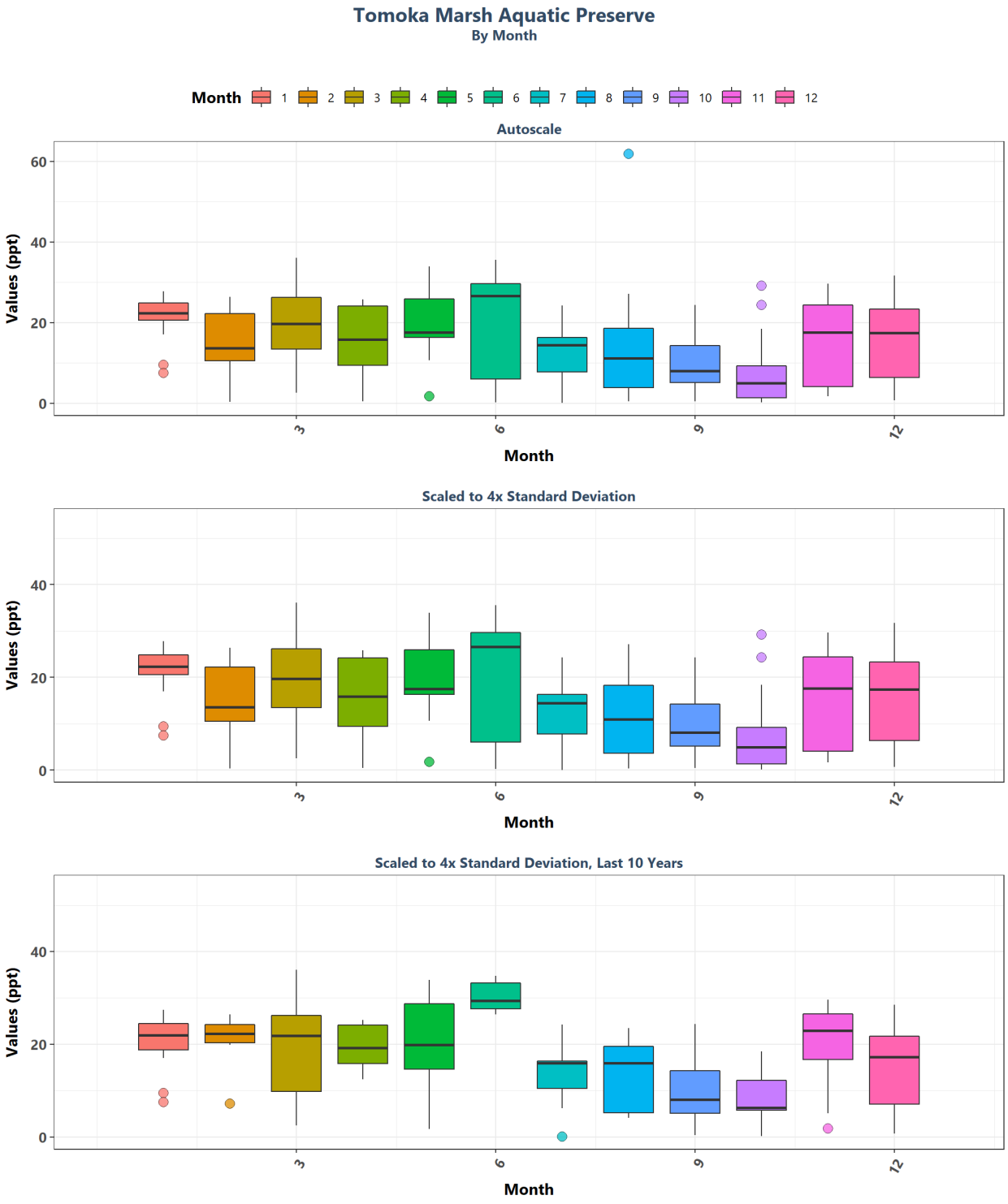
By Year

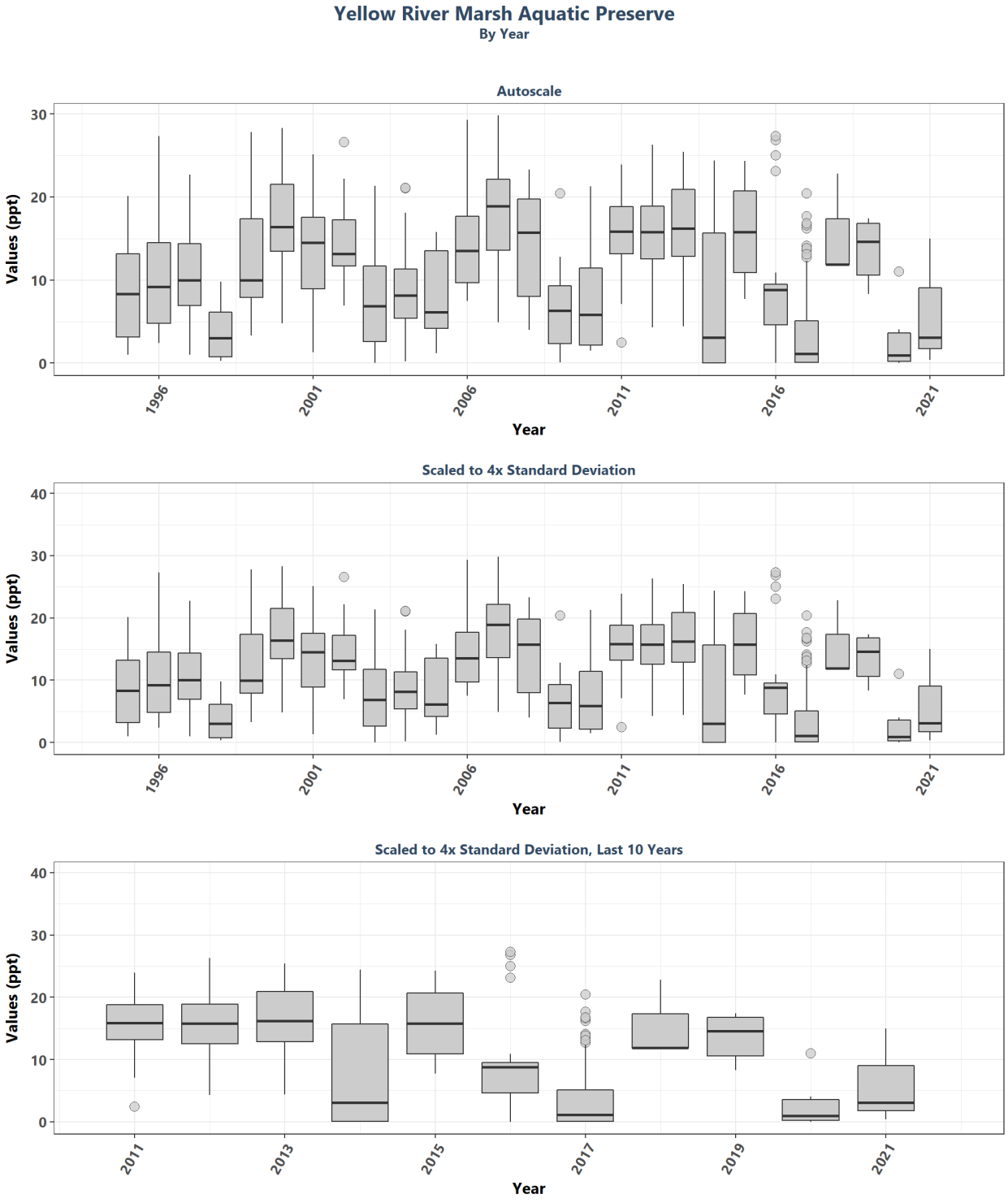


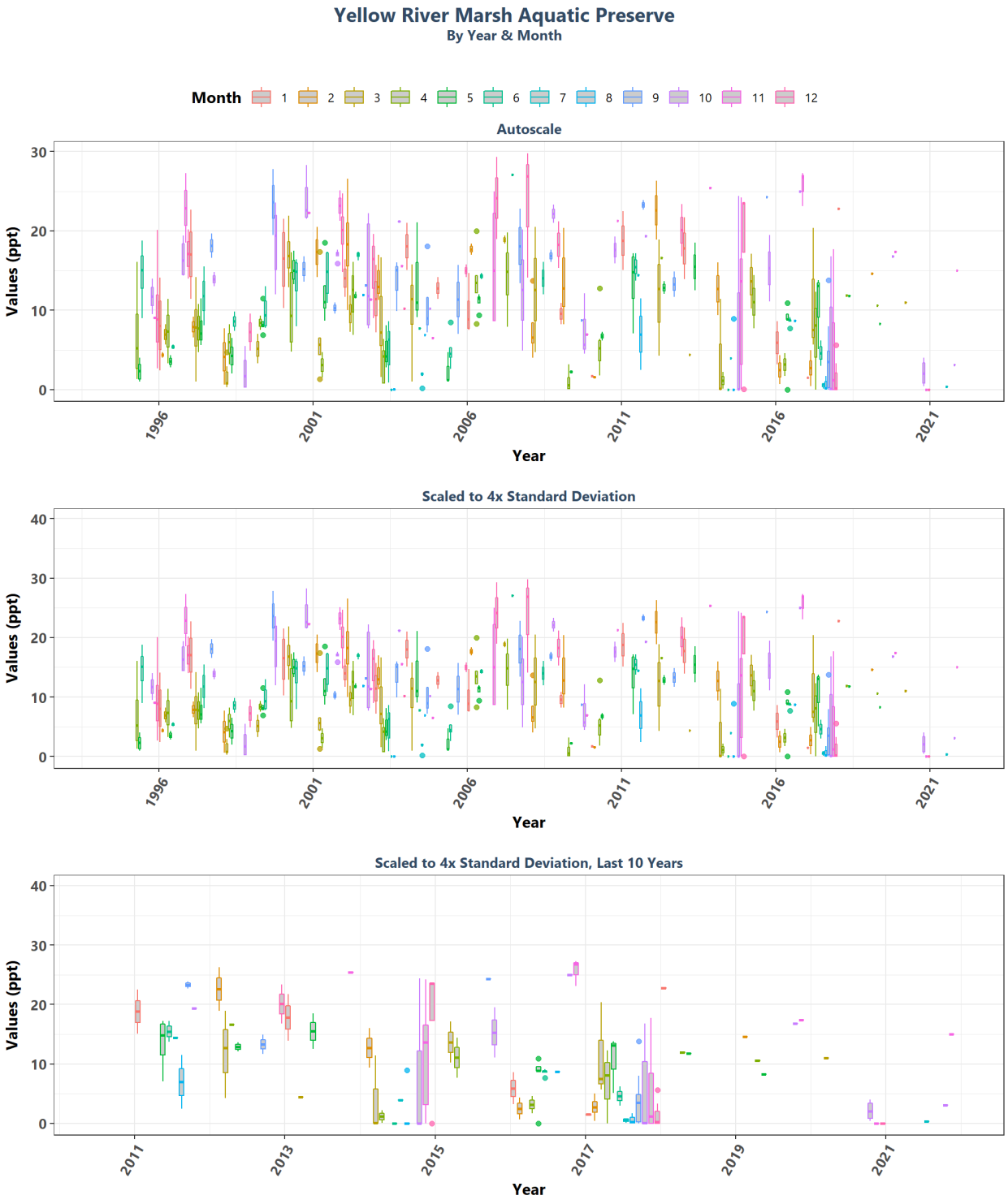


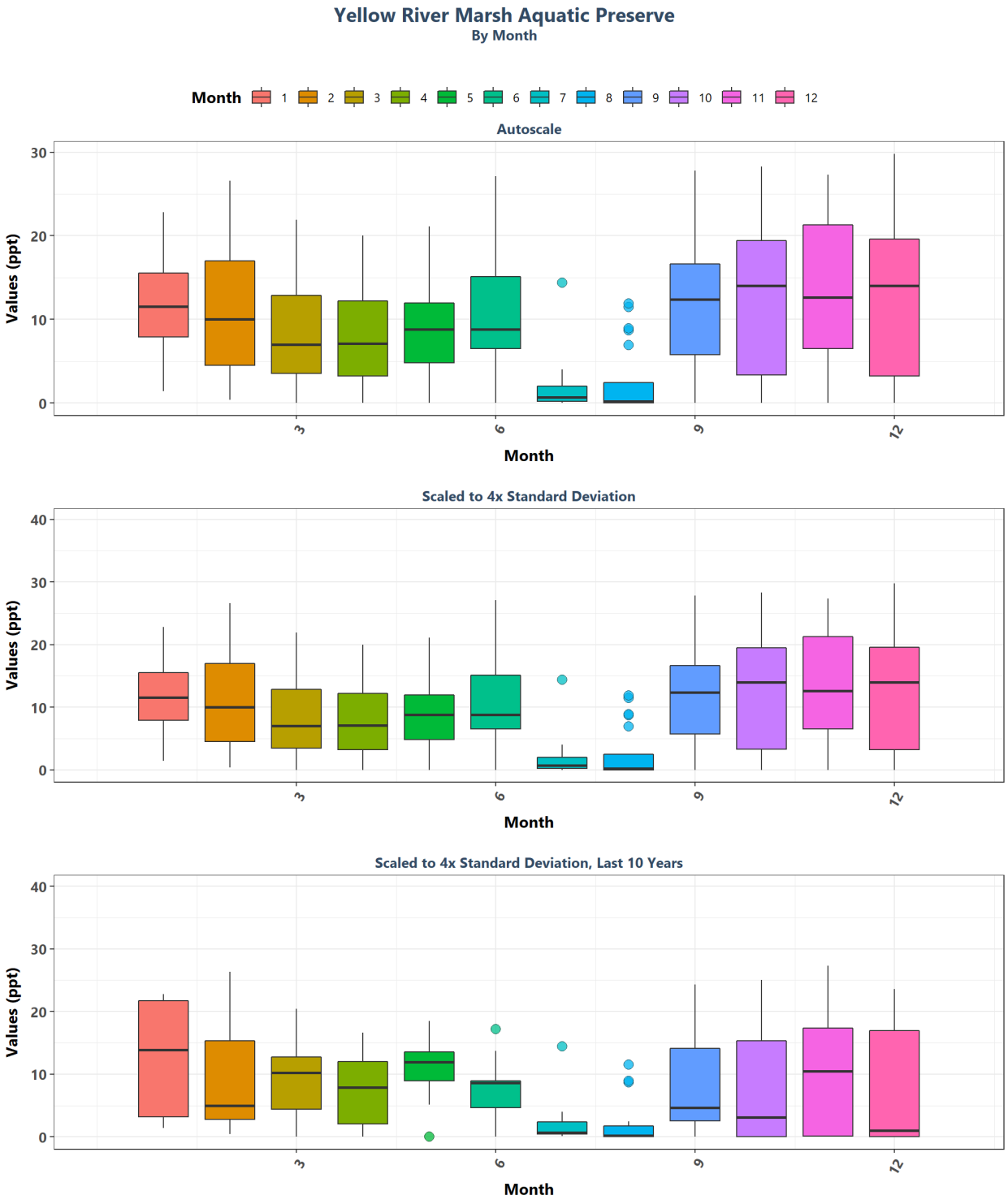



Tomoka Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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


Yellow River Marsh Aquatic Preserve
By Year
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year


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




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