SEACAR Continuous Water Quality Analysis: NE Region for 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 and Settings

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(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_Continuous_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 and units of the parameter.

```
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="")
parameter <- unique(data$ParameterName)
unit <- unique(data$ParameterUnits)
```

Data Filtering

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 `RelativeDepth`, and removes any activity type that has "Blank" in the description. Data passes the filtering the process if it is has an `Include` value of 1.

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

Because the continuous data is extensive and most measurements are taken every 15 minutes, a daily average is determined and used based on grouping `ManagedAreaName`, `ProgramID`, `ProgramName`, `ProgramLocationID`, and `SampleDate`. The new `ResultValue` is the mean of all values on that date from

that specific monitoring location. Sets the `SampleDate` as a date object, and creates various scales of the date to be used by plotting functions.

Creates a variable for each `MonitoringID` which is defined as a unique combination of `ManagedAreaName`, `ProgramID`, `ProgramAreaName`, and `ProgramLocationID`.

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 5 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.

```
data$Include <- as.logical(data$Include)
data <- data[data$Include==TRUE,]
data <- data[!is.na(data$ResultValue),]
data <- data[!is.na(data$RelativeDepth),]
data <- data[!grep("Blank", data$ActivityType),]

if(param_name=="Water_Temperature"){
   data <- data[data$ResultValue>=-5,]

   #temporarily removing FKNMS Temp. data because I think it might be causing R to run out of memory.
   # data <- data[data$ManagedAreaName != "Florida Keys National Marine Sanctuary"]
} else{
   data <- data[data$ResultValue>=0,]
}



data <- data %>%
   group_by(ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            SampleDate) %>%
   dplyr::summarise(Year=unique(Year), Month=unique(Month),
             RelativeDepth=unique(RelativeDepth),
             ResultValue=mean(ResultValue), Include=unique(Include))

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

data$SampleDate <- as.Date(data$SampleDate)
data$YearMonth <- format(data$SampleDate, format = "%m-%Y")
data$YearMonthDec <- data$Year + ((data$Month-0.5) / 12)
data$DecDate <- decimal_date(data$SampleDate)

data <- data %>%
    group_by(ManagedAreaName, ProgramID, ProgramName, ProgramLocationID) %>%
    mutate(MonitoringID=cur_group_id())

Mon_Summ <- data %>%
   group_by(MonitoringID, AreaID, ManagedAreaName, ProgramID, ProgramName,
            ProgramLocationID) %>%
   dplyr::summarize(ParameterName=parameter,
             RelativeDepth=unique(RelativeDepth),
             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))

Mon_Summ <- as.data.table(Mon_Summ[order(Mon_Summ$MonitoringID), ])

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

data$Use_In_Analysis <- ifelse(data$Include==TRUE &
                                  data$SufficientData==TRUE, TRUE, FALSE)
setDT(data)
data[, `:=` (relyear = Year - min(Year), relyear_dd = DecDate - min(DecDate)), by = "ManagedAreaName"]

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

Monitoring Location Statistics

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

1. Take the `data` variable and only include rows that have a `Use_In_Analysis` value of TRUE
2. Group data that have the same `ManagedAreaName`, `ProgramID`, `ProgramName`, `ProgramLocationID`, `Year`, and `Month`.
    - Second summary statistics consider the monitoring location grouping and `Year`.
    - Third summary statistics consider the monitoring location grouping and `Month`.
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, 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`, `ProgramID`, `ProgramName`, `ProgramLocationID`, `Year`, and `Month` in that order.
5. Write summary stats to a pipe-delimited .txt file in the output directory

```
Mon_YM_Stats <- data[data$Use_In_Analysis==TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            Year, Month) %>%
   dplyr::summarize(ParameterName=parameter,
             RelativeDepth=unique(RelativeDepth),
             EarliestSampleDate=min(SampleDate),
             LastSampleDate=max(SampleDate), N=length(ResultValue),
             Min=min(ResultValue), Max=max(ResultValue),
             Median=median(ResultValue), Mean=mean(ResultValue),
             StandardDeviation=sd(ResultValue))
Mon_YM_Stats <- as.data.table(Mon_YM_Stats[order(Mon_YM_Stats$ManagedAreaName,
                                                 Mon_YM_Stats$ProgramID,
                                                 Mon_YM_Stats$ProgramName,
                                                 Mon_YM_Stats$ProgramLocationID,
                                                 Mon_YM_Stats$Year,
```

```
                                                    Mon_YM_Stats$Month), ])
fwrite(Mon_YM_Stats, paste0(out_dir,"/", param_name, "_", region,
                            "_MonitoringLoc_YearMonth_Stats.txt"), sep="|")

Mon_Y_Stats <- data[data$Use_In_Analysis==TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            Year) %>%
   dplyr::summarize(ParameterName=parameter,
             RelativeDepth=unique(RelativeDepth),
             EarliestSampleDate=min(SampleDate),
             LastSampleDate=max(SampleDate), N=length(ResultValue),
             Min=min(ResultValue), Max=max(ResultValue),
             Median=median(ResultValue), Mean=mean(ResultValue),
             StandardDeviation=sd(ResultValue))
Mon_Y_Stats <- as.data.table(Mon_Y_Stats[order(Mon_Y_Stats$ManagedAreaName,
                                               Mon_Y_Stats$ProgramID,
                                               Mon_Y_Stats$ProgramName,
                                               Mon_Y_Stats$ProgramLocationID,
                                               Mon_Y_Stats$Year), ])
fwrite(Mon_Y_Stats, paste0(out_dir,"/", param_name, "_", region,
                           "_MonitoringLoc_Year_Stats.txt"), sep="|")

Mon_M_Stats <- data[data$Use_In_Analysis==TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            Month) %>%
   dplyr::summarize(ParameterName=parameter,
             RelativeDepth=unique(RelativeDepth),
             EarliestSampleDate=min(SampleDate),
             LastSampleDate=max(SampleDate), N=length(ResultValue),
             Min=min(ResultValue), Max=max(ResultValue),
             Median=median(ResultValue), Mean=mean(ResultValue),
             StandardDeviation=sd(ResultValue))
Mon_M_Stats <- as.data.table(Mon_M_Stats[order(Mon_M_Stats$ManagedAreaName,
                                               Mon_M_Stats$ProgramID,
                                               Mon_M_Stats$ProgramName,
                                               Mon_M_Stats$ProgramLocationID,
                                               Mon_M_Stats$Month), ])

fwrite(Mon_M_Stats, paste0(out_dir,"/", param_name, "_", region,
                           "_MonitoringLoc_Month_Stats.txt"), sep="|")
```

Seasonal Kendall Tau Analysis

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

The following steps are performed:

1. Define the trend function.
2. Take the `data` variable and only include rows that have a `Use_In_Analysis` value of `TRUE`

3. Group data that have the same `ManagedAreaName`, `ProgramID`, `ProgramName`, and `ProgramLocationID`.

4. 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,

5. For each group, a temporary variable is created to run 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.
    - tau, Senn Slope (SennSlope), Senn Intercept (SennIntercept), and p are extracted from the model results.

6. The two stats tables are merged based on similar groups, and then Trend is determined from the user-defined function.

7. Write summary stats to a pipe-delimited .txt file in the output directory

    - Click this text to open Git directory with output files

8. Add the Monitoring IDS to `KT.Stats` for easier use while plotting.

```
tauSeasonal <- function(data, independent, stats.median, stats.minYear,
                        stats.maxYear, seasondata = Mon_M_Stats[Mon_M_Stats$ProgramLocationID==unique(da
   setDT(data)
   tau <- NULL
   tryCatch({ken <- kendallSeasonalTrendTest(
      y=data$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(MonitoringID = unique(data$MonitoringID),
                 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[, `:=` (MonitoringID = unique(data$MonitoringID),
                      season = sort(unique(data$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$season))){
  seasondat_s <- data[Month == s, ]
  if(!is.na(unique(seasondat_s$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, Mon_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$relyear, na.rm=TRUE)
  # Calculate Kendall Tau and Slope stats,
  # then update appropriate columns and table
  seasondata <- Mon_M_Stats[Mon_M_Stats$ProgramLocationID==unique(data$ProgramLocationID[data$Monitori
  KT <- tauSeasonal(data, TRUE, stats.median,
                    stats.minYear, stats.maxYear, seasondata)
  #if (is.null(KT[8])) {
  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$ManagedAreaName
c_names <- c("MonitoringID", "Season", "Median", "Independent",
             "tau", "z", "p_z", "chi_sq", "p_chi_sq", "SennSlope", "SennIntercept", "Trend")
if(n==0){
   KT.Stats <- data.frame(matrix(ncol=length(c_names),
                                 nrow=nrow(Mon_Summ)))
   colnames(KT.Stats) <- c_names
   #KT.Stats[, c("MonitoringID")] <- Mon_Summ[, c("MonitoringID")]
} else{
   for (i in 1:n) {
      x <- nrow(data[data$Use_In_Analysis==TRUE &
                        data$MonitoringID==Mon_IDs[i], ])
      if (x>0) {
         KT.Stats <- runStats(data[data$Use_In_Analysis==TRUE &
                                      data$MonitoringID==Mon_IDs[i], ], Mon_M_Stats)
      }
   }
   KT.Stats <- as.data.frame(KT.Stats)

   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(Mon_Summ, KT.Stats,
                              by=c("MonitoringID"), all=TRUE)

KT.Stats <- as.data.table(KT.Stats[order(KT.Stats$MonitoringID), ])
KT.Stats2 <- copy(KT.Stats)
KT.Stats[, `:=` (Region = region, Units = unit)]
KT.Stats_all <- rbind(KT.Stats_all, KT.Stats)

KT.Stats2$MonitoringID <- NULL
fwrite(KT.Stats2, paste0(out_dir,"/", param_name, "_", region,
                         "_KendallTau_Stats.txt"), sep="|")
rm(KT.Stats2)
```

```
#KT.Stats$MonitoringID <- Mon_Summ$MonitoringID
data <- data[!is.na(data$ResultValue),]
```

Appendix I: Dataset Summary Box Plots

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

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

This set of box plots are grouped by year.

```
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.x=element_text(face="bold"),
         axis.text.y=element_text(face="bold"))

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=SampleDate, 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=SampleDate, 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(0, y_scale) +
   plot_theme

p3 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to 4x Standard Deviation, Last 10 Years",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(0, y_scale) +
   scale_x_continuous(limits=c(max(data$Year) - 10.5, max(data$Year)+0.5),
                      breaks=seq(max(data$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.

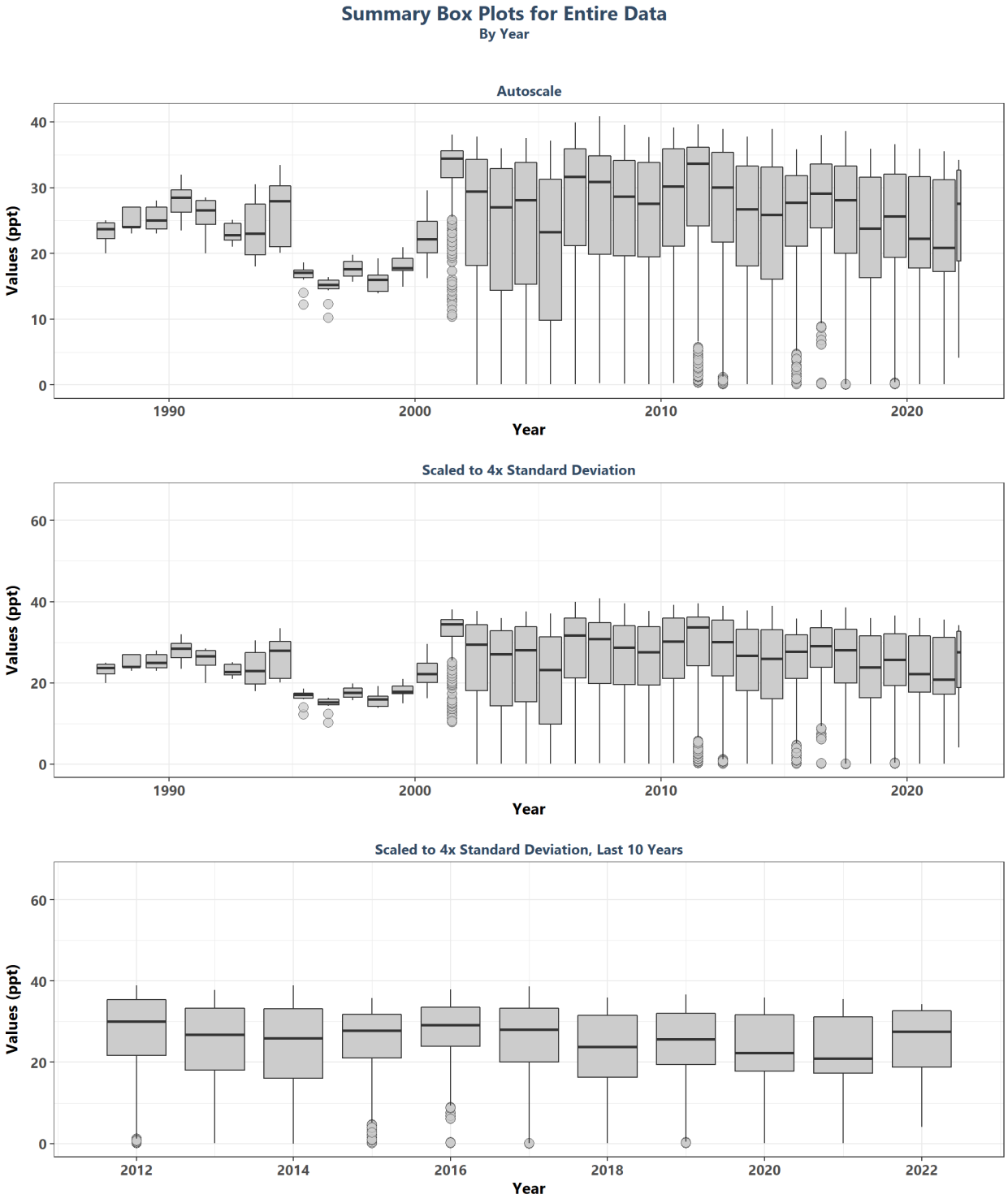
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit, ")"), 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 5x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(0, y_scale) +
   plot_theme +
   theme(legend.position="none")
```

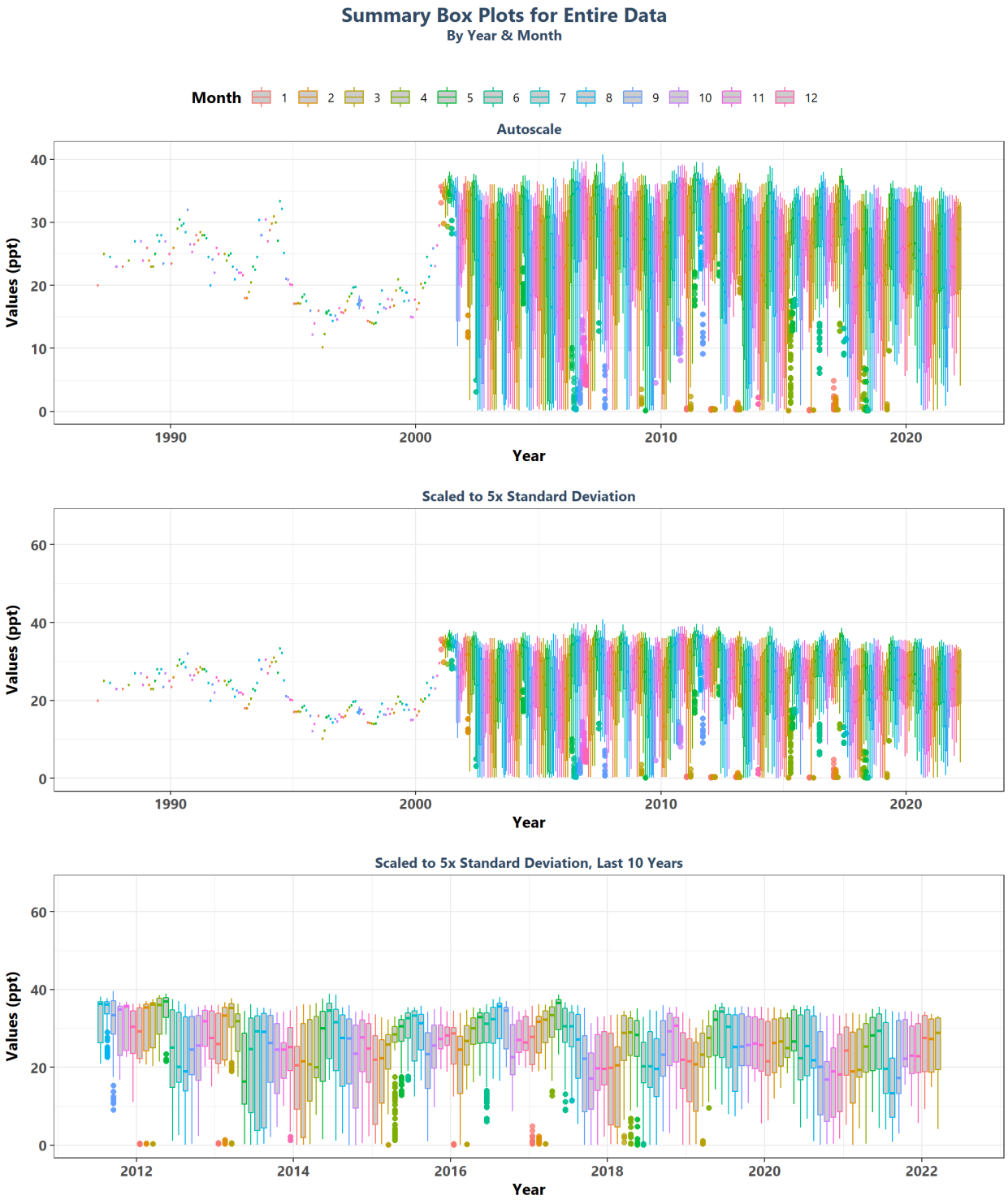
```
p3 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Scaled to 5x Standard Deviation, Last 10 Years",
        x="Year", y=paste0("Values (", unit, ")")) +
   ylim(0, y_scale) +
   scale_x_continuous(limits=c(max(data$Year) - 10.5, max(data$Year)+0.5),
                      breaks=seq(max(data$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))
```

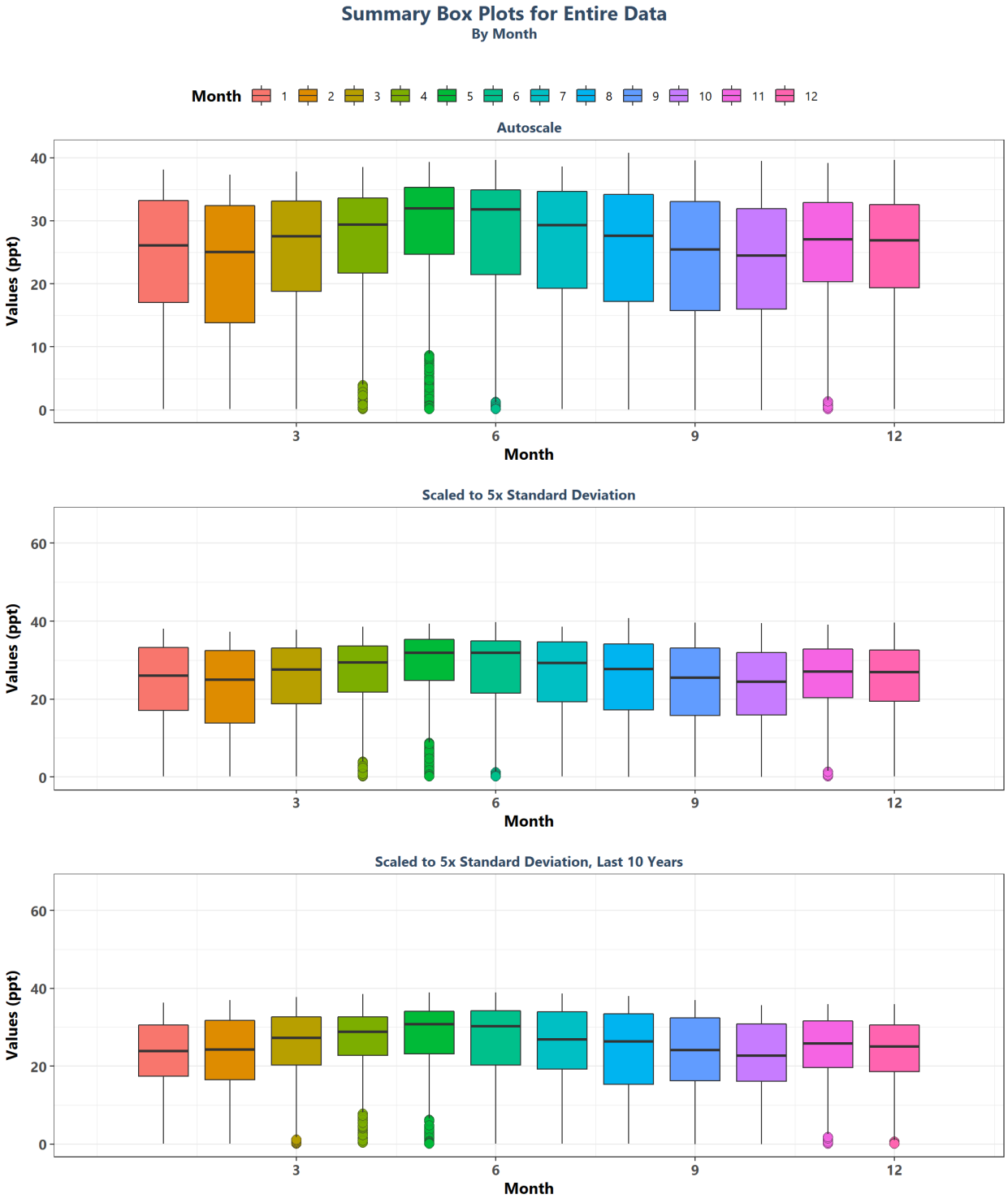
The following box plots are grouped by month with fill color being related to the month. This is designed to view potential seasonal trends.

```
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 5x Standard Deviation",
        x="Month", y=paste0("Values (", unit, ")")) +
   ylim(0, 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 5x Standard Deviation, Last 10 Years",
        x="Month", y=paste0("Values (", unit, ")")) +
   ylim(0, 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
10
20
30
40
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (ppt)
0
20
40
60
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
0
20
40
60
2012
2014
2016
2018
2020
2022
Year

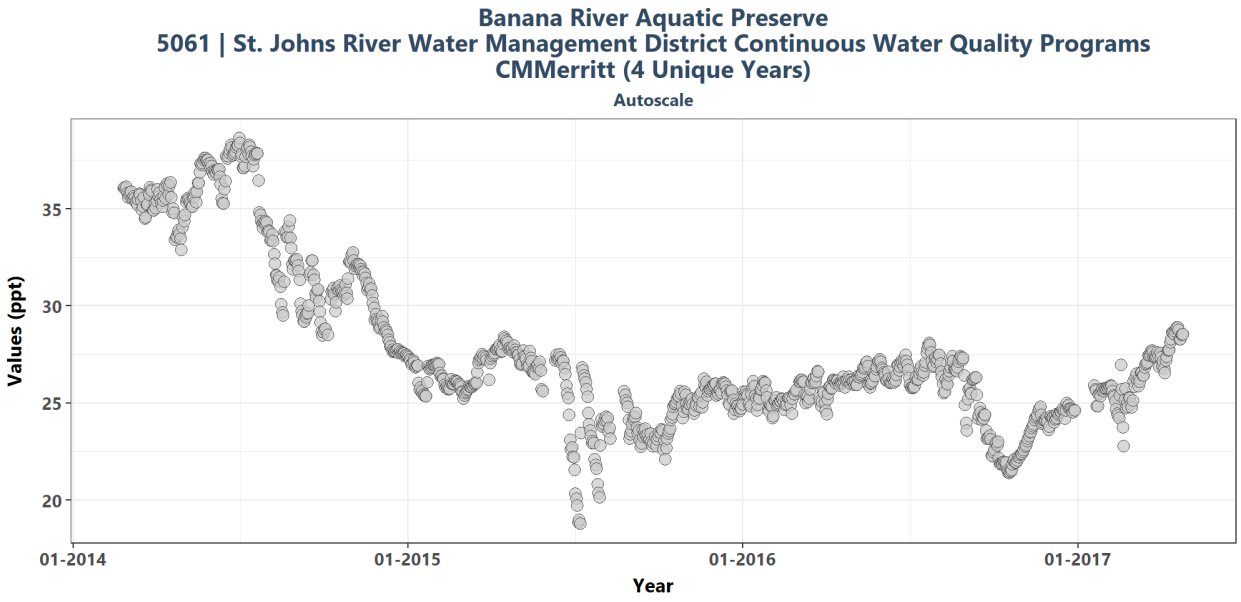

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 5x Standard Deviation
Values (ppt)
Year
Scaled to 5x Standard Deviation, Last 10 Years
Values (ppt)
Year

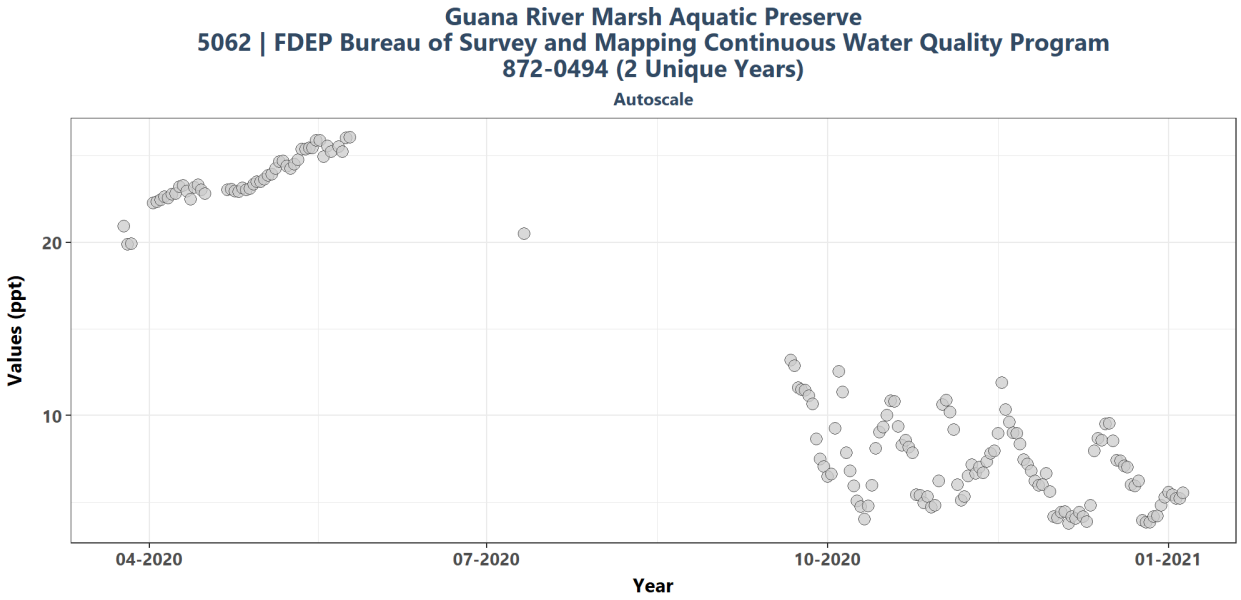


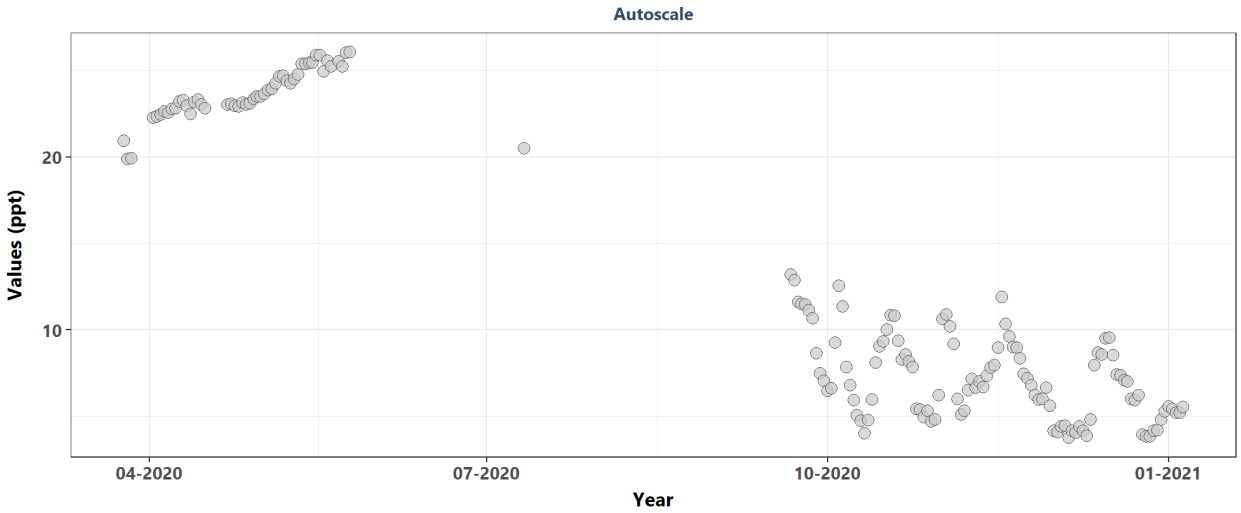
Appendix II: Excluded Monitoring Locations

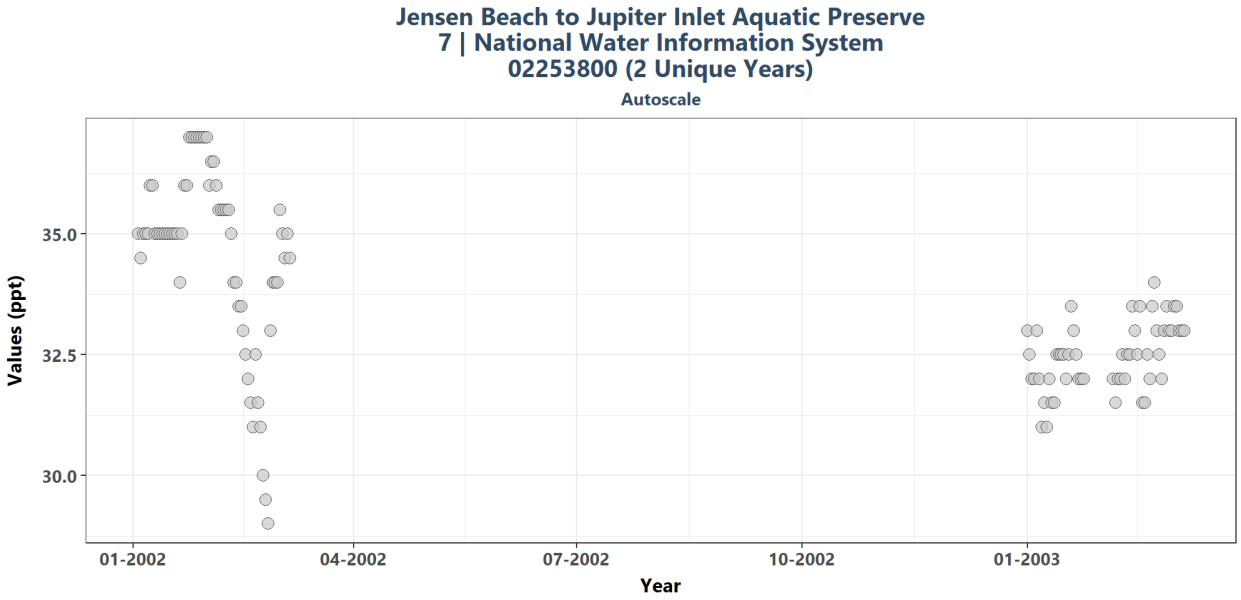
Scatter plots of data values are created for monitoring locations that have fewer than 5 separate years of data entries.

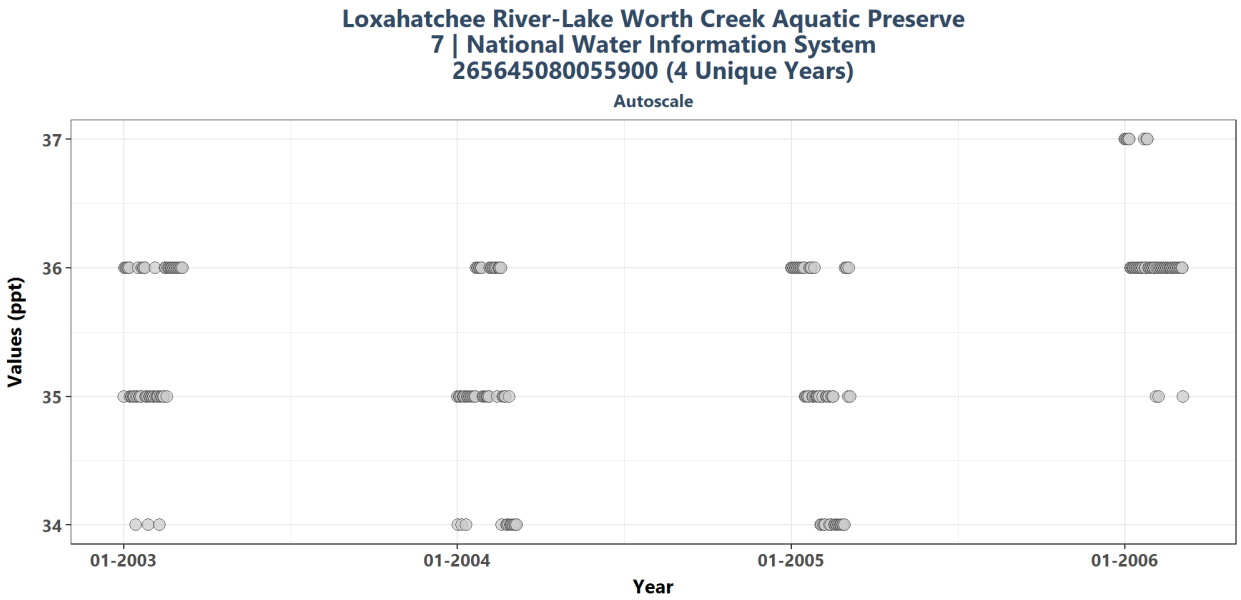
```
Mon_Exclude <- Mon_Summ[Mon_Summ$N_Years<5 & Mon_Summ$N_Years>0,]
Mon_Exclude <- Mon_Exclude[order(Mon_Exclude$MonitoringID),]
z=nrow(Mon_Exclude)

if(z==0){
   print("There are no monitoring locations that qualify.")
} else {
   for(i in 1:z){
      MA_name <- unique(data$ManagedAreaName[
         data$MonitoringID==Mon_Exclude$MonitoringID[i]])
      Mon_name <- paste0(unique(data$ProgramID[
         data$MonitoringID==Mon_Exclude$MonitoringID[i]]), " | ",
         unique(data$ProgramName[
            data$MonitoringID==Mon_Exclude$MonitoringID[i]]), "\n",
         unique(data$ProgramLocationID[
            data$MonitoringID==Mon_Exclude$MonitoringID[i]]))

      p1<-ggplot(data=data[data$MonitoringID==Mon_Exclude$MonitoringID[i]&
                           data$Include==TRUE, ],
                 aes(x=SampleDate, y=ResultValue)) +
         geom_point(shape=21, size=3, color="#333333", fill="#cccccc",
                    alpha=0.75) +
         labs(title=paste0(MA_name, "\n",
                           Mon_name, " (", Mon_Exclude$N_Years[i],
                       " Unique Years)"),
              subtitle="Autoscale", x="Year",
              y=paste0("Values (", unit, ")")) +
         plot_theme +
         scale_x_date(labels=date_format("%m-%Y"))
      print(p1)
   }
}
```

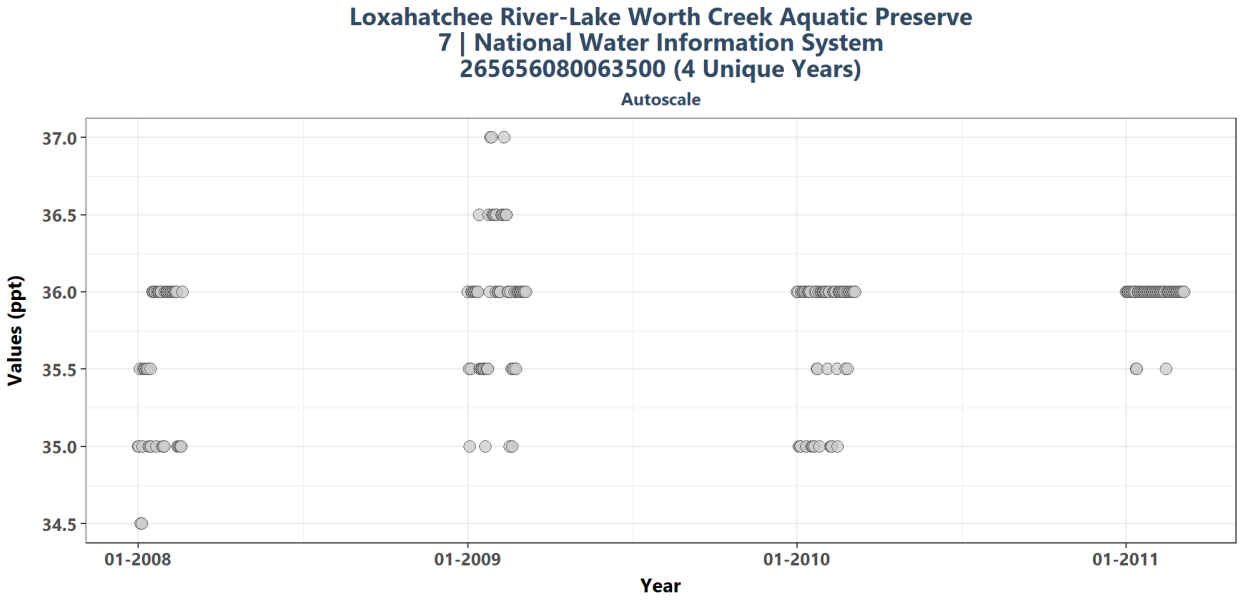


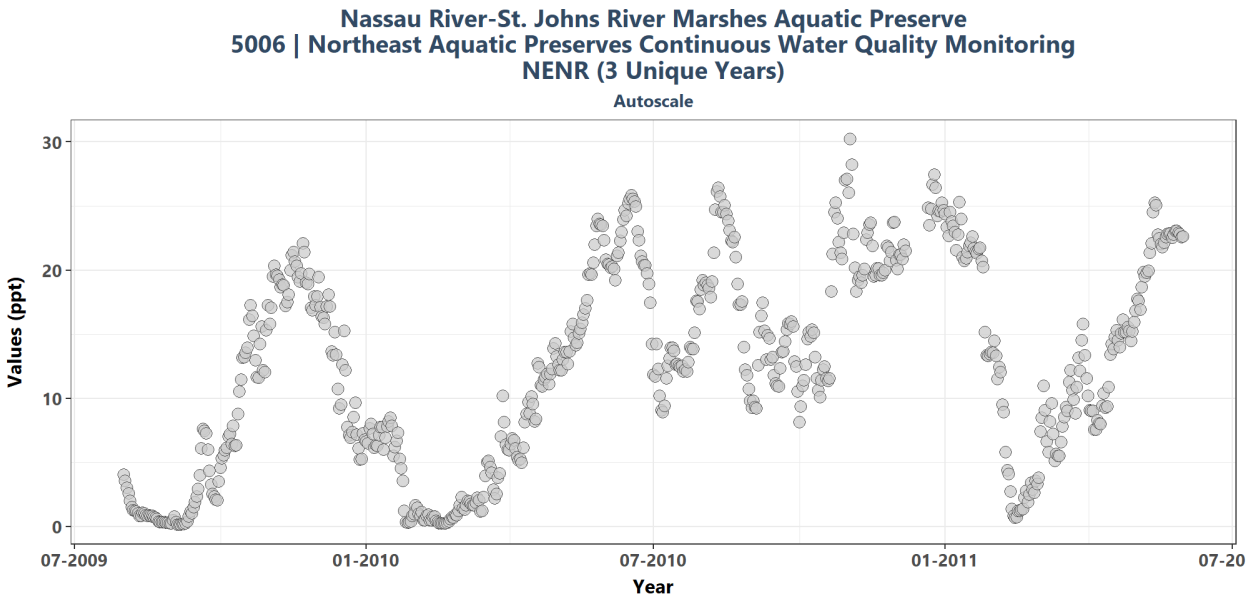

Guana River Marsh Aquatic Preserve
5062 | FDEP Bureau of Survey and Mapping Continuous Water Quality Program
872-0494 (2 Unique Years)
Autoscale
Values (ppt)
20
10
04-2020
07-2020
10-2020
01-2021
Year

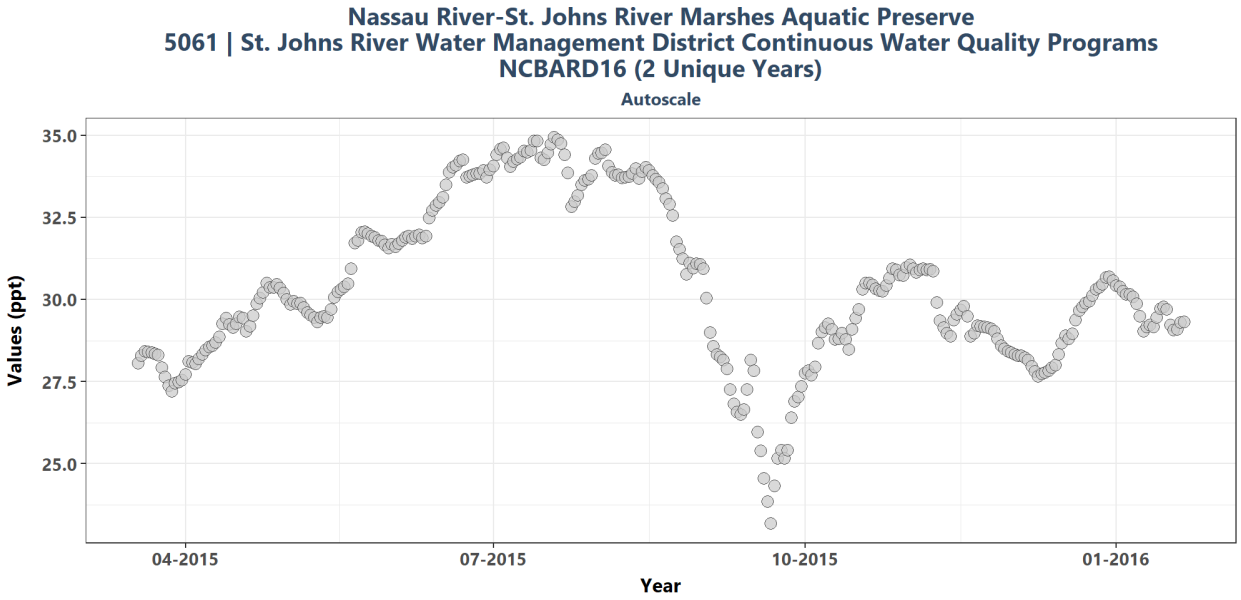

Guana Tolomato Matanzas National Estuarine Research Reserve
5062 | FDEP Bureau of Survey and Mapping Continuous Water Quality Program
872-0494 (2 Unique Years)
Autoscale
Values (ppt)
20
10
04-2020
07-2020
10-2020
01-2021
Year

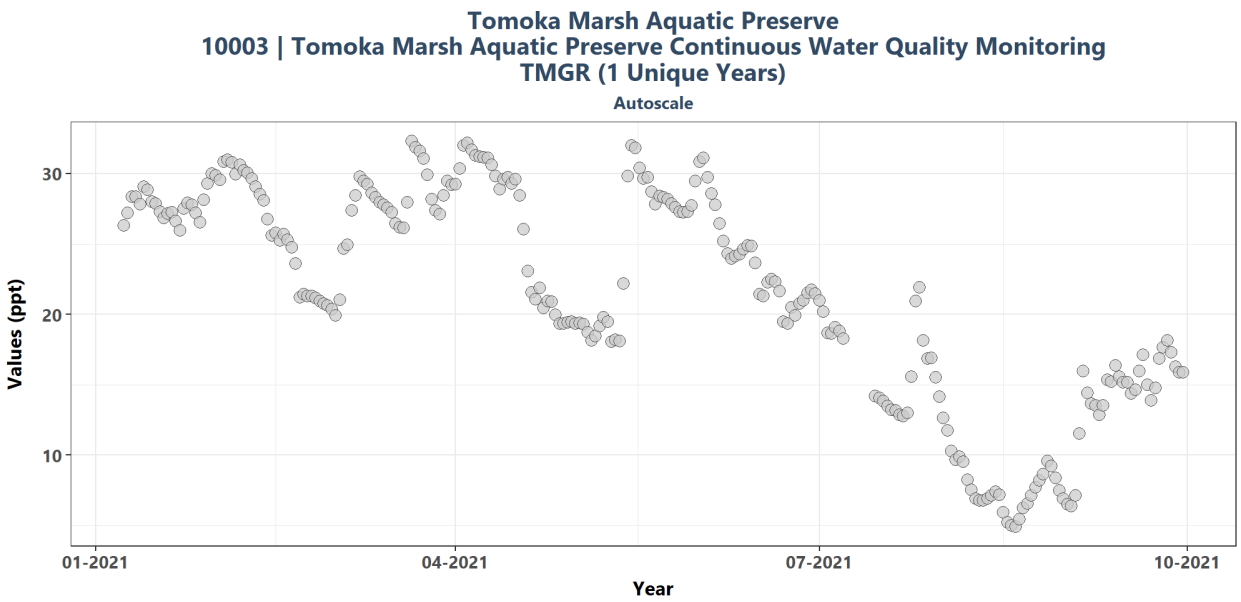

Jensen Beach to Jupiter Inlet Aquatic Preserve
7 | National Water Information System
02253800 (2 Unique Years)
Autoscale
Values (ppt)
35.0
32.5
30.0
01-2002
04-2002
07-2002
10-2002
01-2003
Year


Loxahatchee River-Lake Worth Creek Aquatic Preserve
7 | National Water Information System
265645080055900 (4 Unique Years)
Autoscale
Values (ppt)
37
36
35
34
01-2003
01-2004
01-2005
01-2006
Year


Loxahatchee River-Lake Worth Creek Aquatic Preserve
7 | National Water Information System
265656080063500 (4 Unique Years)
Autoscale
37.0
36.5
36.0
35.5
35.0
34.5
Values (ppt)
01-2008
01-2009
01-2010
01-2011
Year


Nassau River-St. Johns River Marshes Aquatic Preserve
5006 | Northeast Aquatic Preserves Continuous Water Quality Monitoring
NENR (3 Unique Years)
Autoscale
30
20
10
0
Values (ppt)
07-2009
01-2010
07-2010
01-2011
07-20
Year





Appendix III: Monitoring Location Trendlines

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

1. Use the data set that only has `Use_In_Analysis` of `TRUE` for the desired monitoring location
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

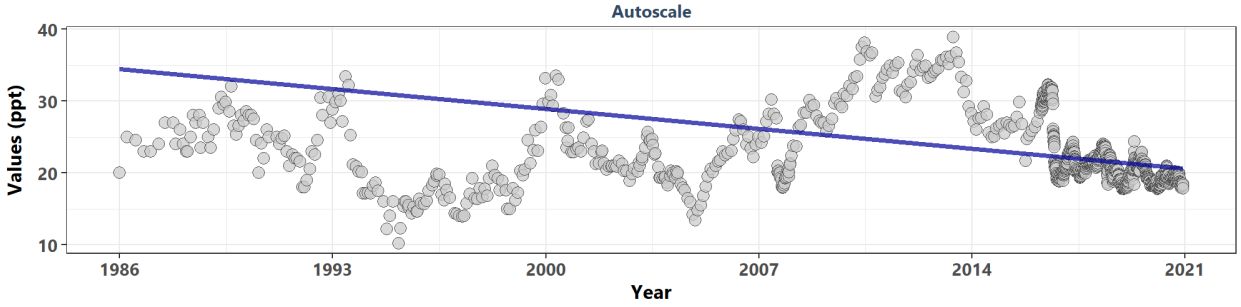
5. Set the plot type as a point plot with the size of the points
6. Add the linear trend
7. Create the title, x-axis, y-axis, and color fill labels
8. Set the y and x limits
9. Make the axis labels bold
10. Plot the arrangement as a set of panels

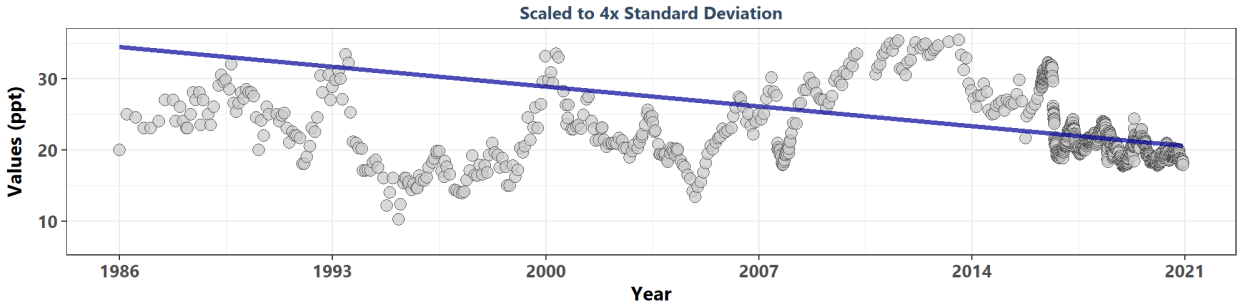
```
if(n==0){
   print("There are no monitoring locations that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$Use_In_Analysis==TRUE &
                                    data$MonitoringID==Mon_IDs[i],]
      plot_data$Season <- factor(plot_data$Month, levels = c("All", seq(1, 12)),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

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


      MA_name <- KT.Stats$ManagedAreaName[KT.Stats$MonitoringID==Mon_IDs[i]]
      Mon_name <- paste0(KT.Stats$ProgramID[KT.Stats$MonitoringID==Mon_IDs[i]],
         " | ", KT.Stats$ProgramName[KT.Stats$MonitoringID==Mon_IDs[i]], "\n",
         KT.Stats$ProgramLocationID[KT.Stats$MonitoringID==Mon_IDs[i]])

      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

      # x1 <- min(unique(data$relyear_dd[data$Year == KT.Stats[ManagedAreaName==MA_Include[i] & Season
      # y1 <- min(unique(data$relyear_dd[data$Year == KT.Stats[ManagedAreaName==MA_Include[i] & Season
      # x_end1 <- max(unique(data$relyear_dd[data$Year == KT.Stats[ManagedAreaName==MA_Include[i] & Sea
      # y_end1 <- max(unique(data$relyear_dd[data$Year == KT.Stats[ManagedAreaName==MA_Include[i] & Sea
      # x1 <- relyear_dd_lower
      # y1 <- relyear_dd_lower * KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", SennSlope]
```

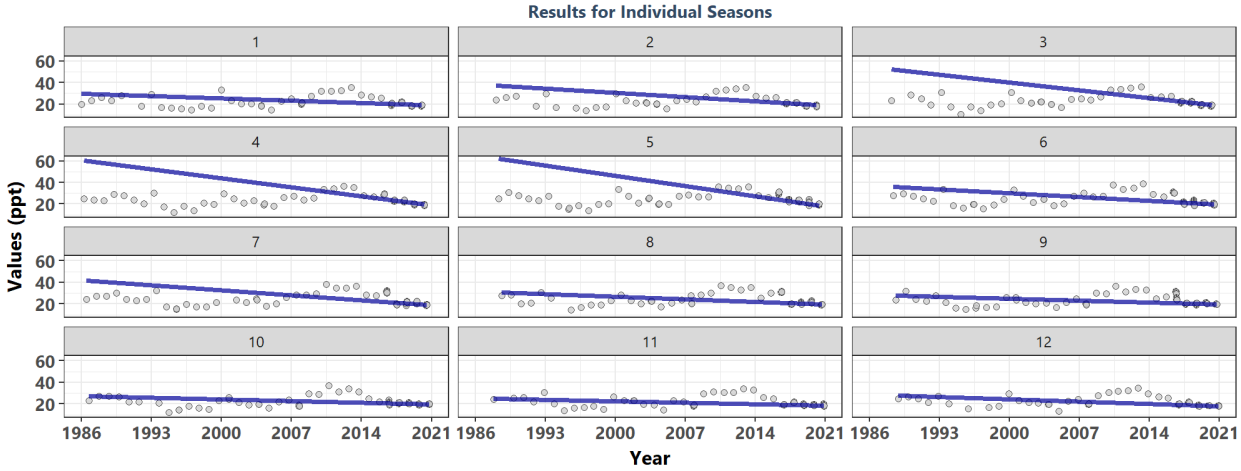
```
# x_end1 <- relyear_dd_upper
# y_end1 <- relyear_dd_upper * KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", SennSlo

KT.Stats[, season := Season]
KT.Stats[MonitoringID == Mon_IDs[i] & season != "All", `:=` (N_Data = nrow(plot_data[Season == sea
KT.Stats[MonitoringID == Mon_IDs[i] & season == "All", `:=` (relyear_dd_lower = min(plot_data[Moni
KT.Stats[, season := NULL]


p1 <- ggplot(data=plot_data,
             aes(x=relyear_dd, y=ResultValue)) +
   geom_point(shape=21, size=3, color="#333333", fill="#cccccc",
              alpha=0.75) +
   #geom_abline(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", ], aes(slope=Se
   #            color="#000099", size=1.2, alpha=0.7) +
   geom_segment(data = KT.Stats[MonitoringID == Mon_IDs[i] & Season == "All", ], aes(x = relyear_
                                                                                     y = relyear_
                                                                                     xend = relyea
                                                                                     yend = relyea


                color="#000099", size=1.2, alpha=0.7) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")")) +
   plot_theme +
   scale_x_continuous(breaks = xbrks,
                      labels = xlabs)


p2 <- ggplot(data=plot_data,
             aes(x=relyear_dd, y=ResultValue)) +
   geom_point(shape=21, size=3, color="#333333", fill="#cccccc",
              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[MonitoringID == Mon_IDs[i] & Season == "All", ], aes(x = relyear_
                                                                                     y = relyear_
                                                                                     xend = relyea
                                                                                     yend = relyea


                color="#000099", size=1.2, alpha=0.7) +
   ylim(min_RV-0.1*y_scale, y_scale) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
   plot_theme +
   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[MonitoringID == Mon_IDs[i] & Season != "All", ], aes(x = relyea
                                                                                        y = relyea
                                                                                        xend = rel
                                                                                        yend = rel


                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 Seas
            facet_wrap(~Season, ncol = 3) +
            plot_theme

      KTset <- ggarrange(p1, p2, splot, ncol=1, heights=c(1, 1, 1.5))

      p0 <- ggplot() + labs(title=paste0(MA_name, "\n", Mon_name)) +
         plot_theme + theme(panel.border=element_blank(),
                            panel.grid.major=element_blank(),
                            panel.grid.minor=element_blank(),
                            axis.line=element_blank())

      KT.Stats[MonitoringID==Mon_IDs[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$MonitoringID==Mon_IDs[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)
   }
}
```

Banana River Aquatic Preserve
5061 | St. Johns River Water Management District Continuous Water Quality Programs
IRLB04



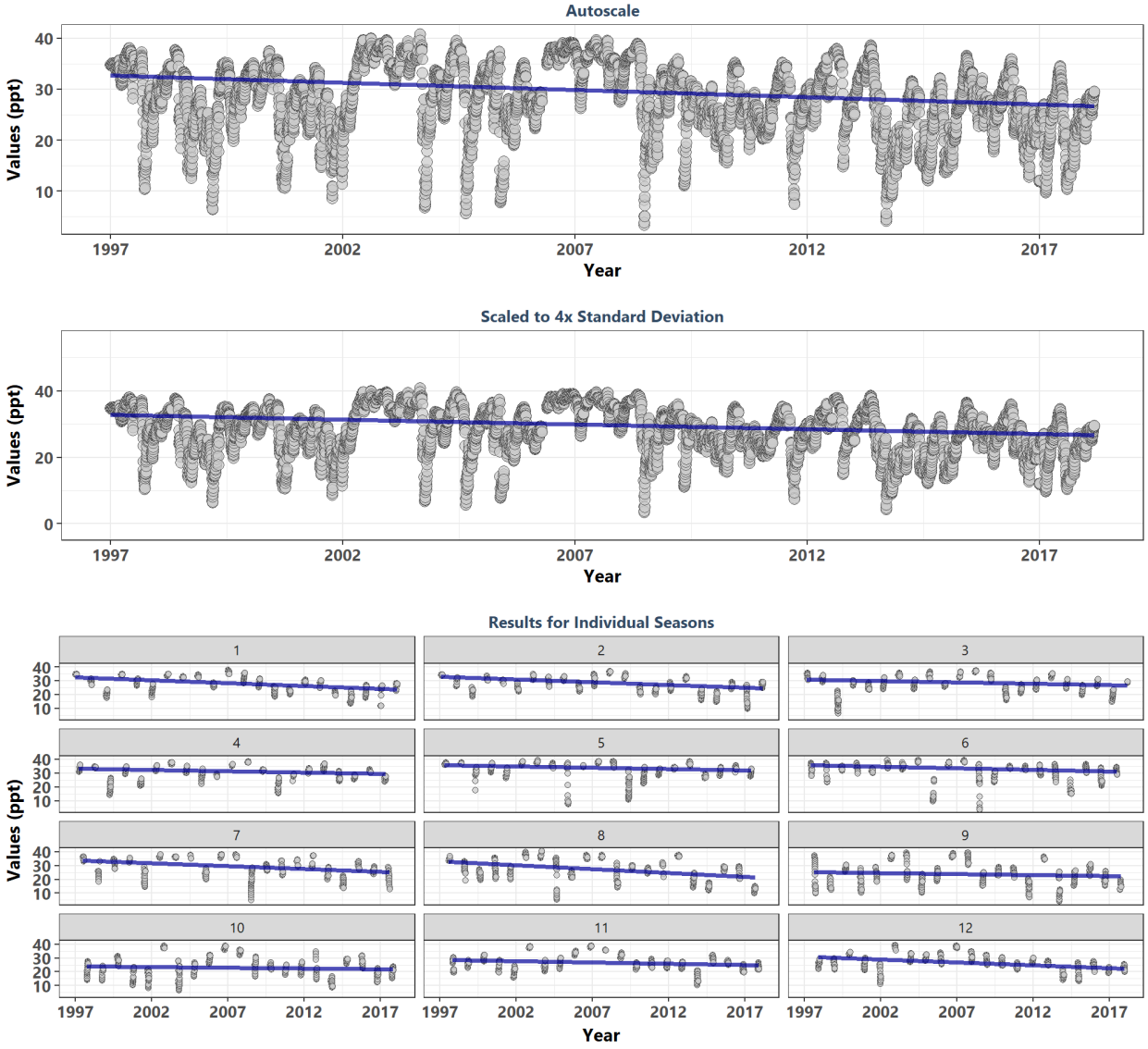




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2080	20.50	-0.3842	-0.3967	34.4631	-26.1	0	39	0.0001	-1
1	158	19.66	-0.2562	-0.3036	29.3701	-4.9	0	NA	NA	-1
2	142	20.62	-0.3876	-0.5520	38.2831	-7.0	0	NA	NA	-1
3	155	21.33	-0.5240	-1.0409	54.6379	-9.8	0	NA	NA	-1
4	157	22.59	-0.5740	-1.1940	60.7983	-10.8	0	NA	NA	-1
5	185	23.00	-0.5204	-1.3631	65.2553	-10.6	0	NA	NA	-1
6	180	21.21	-0.3672	-0.5161	37.4637	-7.4	0	NA	NA	-1
7	186	21.31	-0.3579	-0.6515	41.8311	-7.3	0	NA	NA	-1
8	185	20.57	-0.3329	-0.3403	31.4626	-6.8	0	NA	NA	-1
9	184	20.66	-0.2768	-0.2549	28.5606	-5.6	0	NA	NA	-1
10	190	20.22	-0.3275	-0.2276	27.2784	-6.8	0	NA	NA	-1
11	182	19.17	-0.3240	-0.1985	25.3223	-6.6	0	NA	NA	-1
12	176	19.25	-0.3960	-0.3034	28.6524	-7.9	0	NA	NA	-1

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

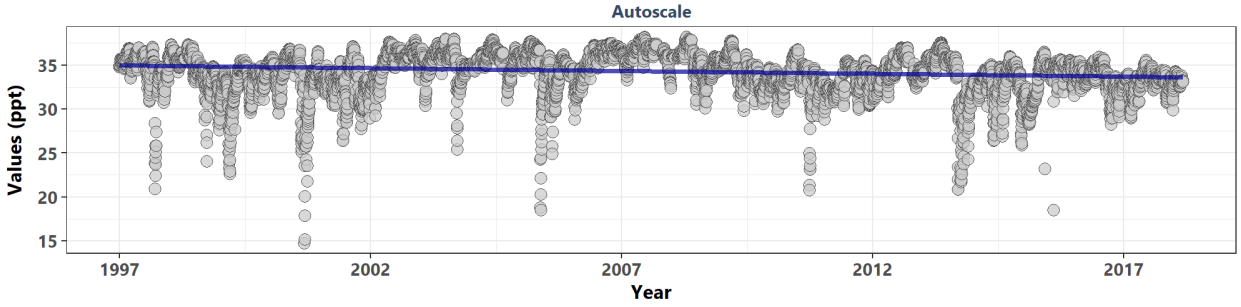
Guana River Marsh Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq

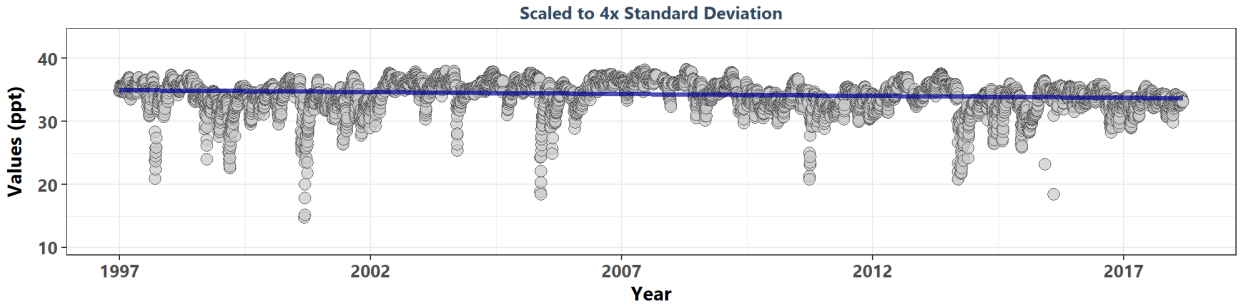


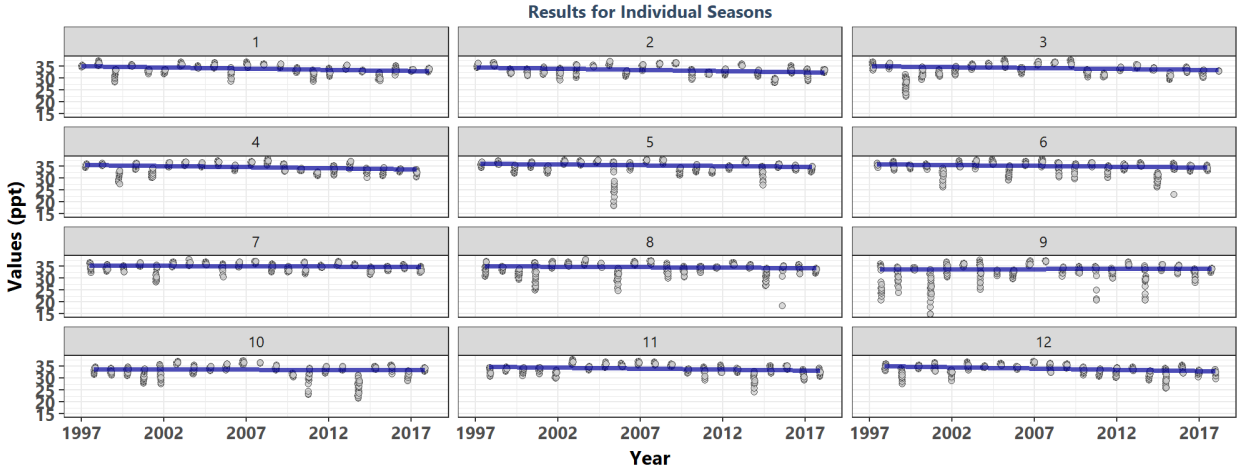
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7023	28.40	-0.1969	-0.2859	32.7457	-24.8	0.0000	122.5	0	-1
1	648	28.46	-0.3130	-0.4183	32.6453	-11.9	0.0000	NA	NA	-1
2	605	28.47	-0.2817	-0.3982	32.8461	-10.4	0.0000	NA	NA	-1
3	626	28.95	-0.1328	-0.2046	30.9952	-5.0	0.0000	NA	NA	-1
4	540	31.47	-0.1607	-0.1874	33.2466	-5.6	0.0000	NA	NA	-1
5	562	33.91	-0.1647	-0.1833	35.9292	-5.8	0.0000	NA	NA	-1
6	579	33.41	-0.2199	-0.2299	35.9384	-7.9	0.0000	NA	NA	-1
7	516	29.46	-0.2399	-0.4163	33.8288	-8.2	0.0000	NA	NA	-1
8	568	27.53	-0.2464	-0.5580	33.1105	-8.8	0.0000	NA	NA	-1
9	557	24.00	-0.0695	-0.1556	25.3994	-2.5	0.0140	NA	NA	-1
10	611	22.90	-0.0510	-0.0960	23.8618	-1.9	0.0592	NA	NA	-1
11	598	26.75	-0.1473	-0.1813	28.5667	-5.4	0.0000	NA	NA	-1
12	613	26.86	-0.3230	-0.4159	31.0164	-12.0	0.0000	NA	NA	-1

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

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmfmwq



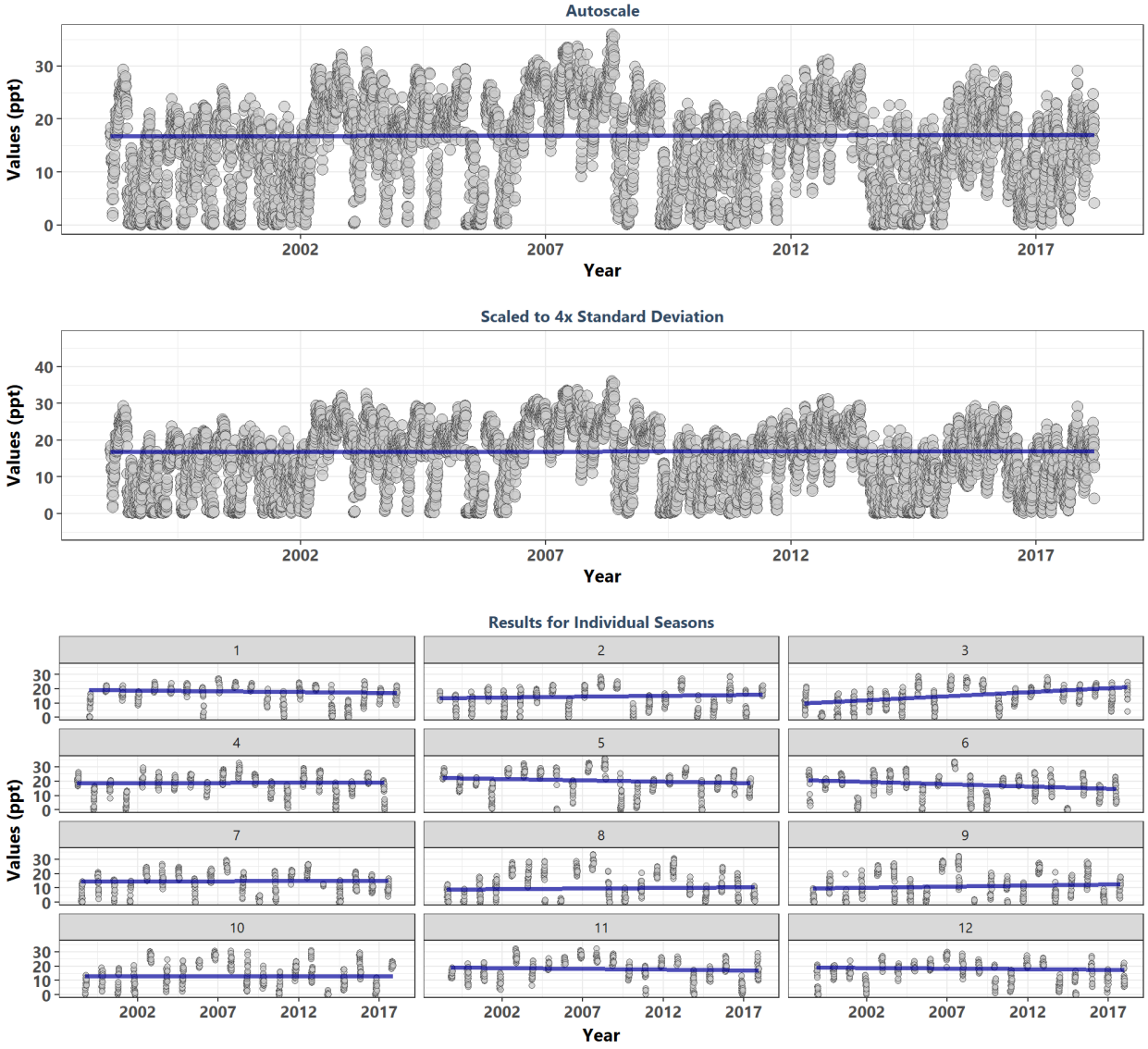




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7156	34.41	-0.1294	-0.0617	34.9622	-16.5	0.0000	93.5	0	-1
1	592	34.04	-0.2260	-0.0910	34.9498	-8.2	0.0000	NA	NA	-1
2	577	33.51	-0.1948	-0.0960	34.5622	-7.0	0.0000	NA	NA	-1
3	603	34.33	-0.1344	-0.0782	35.1070	-4.9	0.0000	NA	NA	-1
4	608	34.74	-0.1718	-0.0928	35.6666	-6.3	0.0000	NA	NA	-1
5	584	35.39	-0.1277	-0.0610	36.0035	-4.6	0.0000	NA	NA	-1
6	581	35.26	-0.1564	-0.0620	35.8812	-5.6	0.0000	NA	NA	-1
7	592	35.14	-0.0722	-0.0242	35.3584	-2.6	0.0085	NA	NA	-1
8	619	34.62	-0.0612	-0.0307	34.9302	-2.3	0.0226	NA	NA	-1
9	570	33.85	0.0227	0.0127	33.7184	0.8	0.4166	NA	NA	1
10	592	33.64	-0.0259	-0.0131	33.7572	-0.9	0.3447	NA	NA	-1
11	614	34.00	-0.1674	-0.0814	34.8171	-6.2	0.0000	NA	NA	-1
12	624	33.96	-0.2298	-0.1017	34.9745	-8.6	0.0000	NA	NA	-1

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

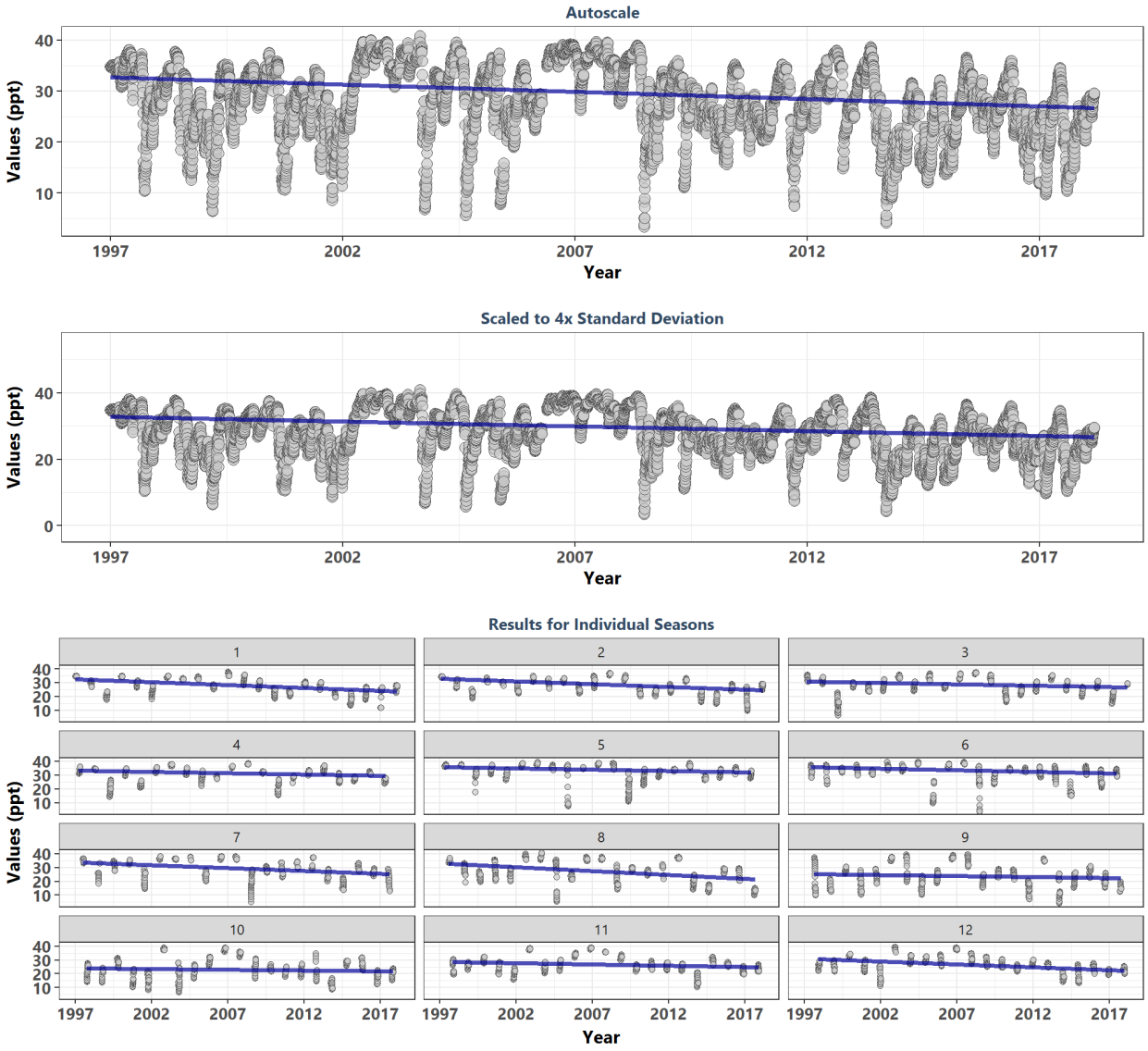
Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7059	16.64	0.0044	0.0082	16.8160	0.6	0.5331	170.2	0	0
1	587	18.02	-0.0675	-0.0935	19.1465	-2.4	0.0143	NA	NA	0
2	567	14.71	0.0573	0.1107	13.4904	2.0	0.0411	NA	NA	0
3	612	15.10	0.2673	0.5524	9.3010	9.9	0.0000	NA	NA	0
4	588	18.92	0.0024	0.0043	18.8717	0.1	0.9321	NA	NA	0
5	608	20.48	-0.0836	-0.1783	22.4393	-3.1	0.0020	NA	NA	0
6	567	17.58	-0.1657	-0.3259	21.1670	-5.9	0.0000	NA	NA	0
7	578	14.84	0.0054	0.0083	14.7604	0.2	0.8471	NA	NA	0
8	575	9.77	0.0545	0.0887	8.8843	2.0	0.0502	NA	NA	0
9	570	11.00	0.0742	0.1526	9.4754	2.7	0.0080	NA	NA	0
10	605	13.01	0.0029	0.0066	12.9368	0.1	0.9138	NA	NA	0
11	586	18.26	-0.0507	-0.1083	19.3414	-1.8	0.0663	NA	NA	0
12	616	18.25	-0.0499	-0.0931	19.1759	-1.9	0.0638	NA	NA	0

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

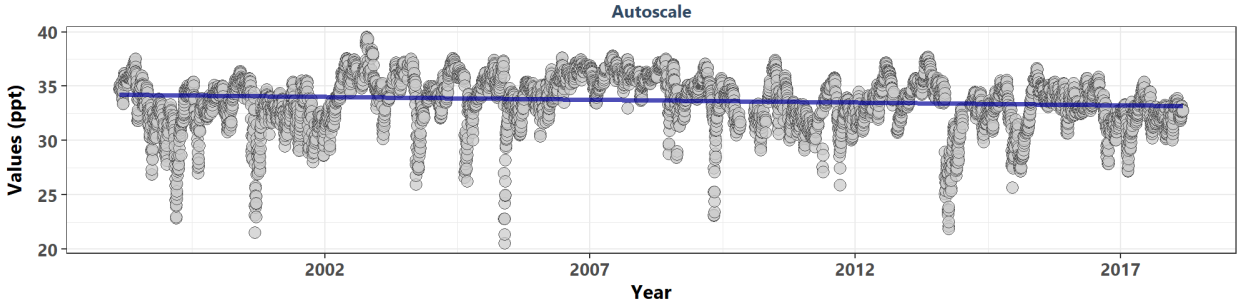
Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq

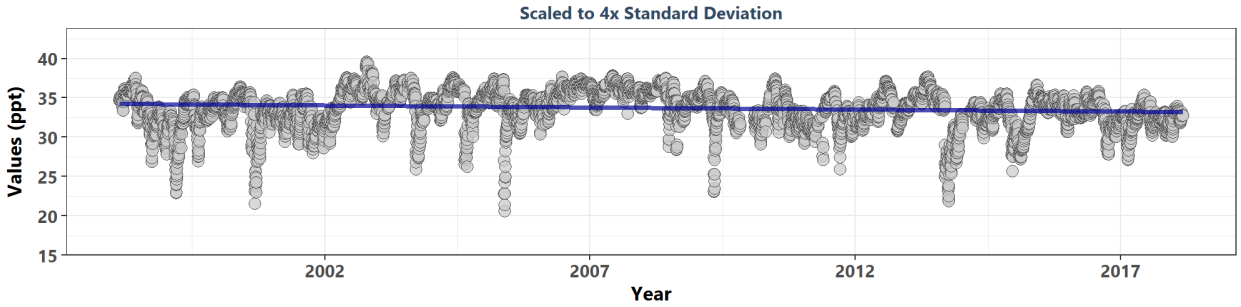


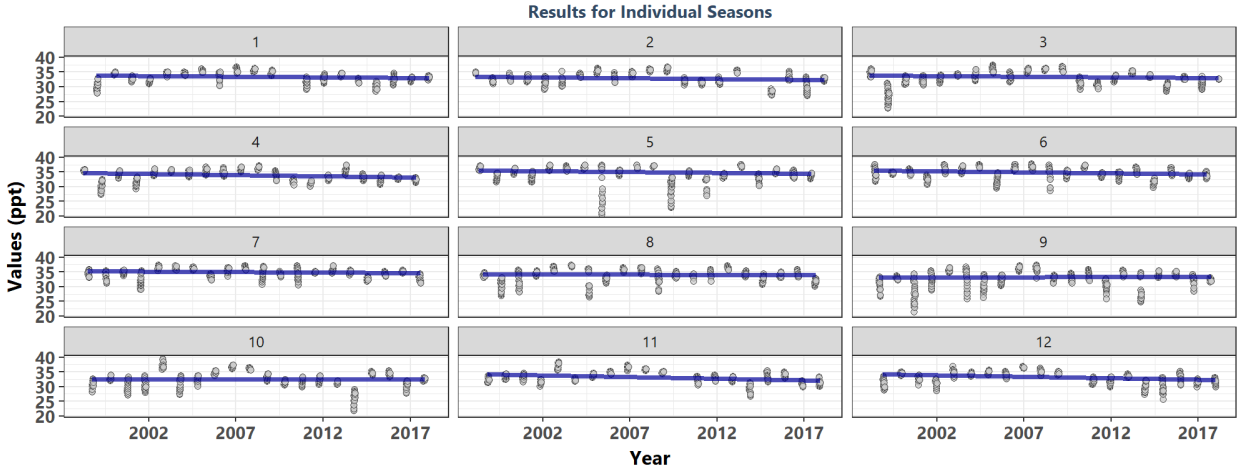
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7023	28.40	-0.1969	-0.2859	32.7457	-24.8	0.0000	122.5	0	-1
1	648	28.46	-0.3130	-0.4183	32.6453	-11.9	0.0000	NA	NA	-1
2	605	28.47	-0.2817	-0.3982	32.8461	-10.4	0.0000	NA	NA	-1
3	626	28.95	-0.1328	-0.2046	30.9952	-5.0	0.0000	NA	NA	-1
4	540	31.47	-0.1607	-0.1874	33.2466	-5.6	0.0000	NA	NA	-1
5	562	33.91	-0.1647	-0.1833	35.9292	-5.8	0.0000	NA	NA	-1
6	579	33.41	-0.2199	-0.2299	35.9384	-7.9	0.0000	NA	NA	-1
7	516	29.46	-0.2399	-0.4163	33.8288	-8.2	0.0000	NA	NA	-1
8	568	27.53	-0.2464	-0.5580	33.1105	-8.8	0.0000	NA	NA	-1
9	557	24.00	-0.0695	-0.1556	25.3994	-2.5	0.0140	NA	NA	-1
10	611	22.90	-0.0510	-0.0960	23.8618	-1.9	0.0592	NA	NA	-1
11	598	26.75	-0.1473	-0.1813	28.5667	-5.4	0.0000	NA	NA	-1
12	613	26.86	-0.3230	-0.4159	31.0164	-12.0	0.0000	NA	NA	-1

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

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmsswq

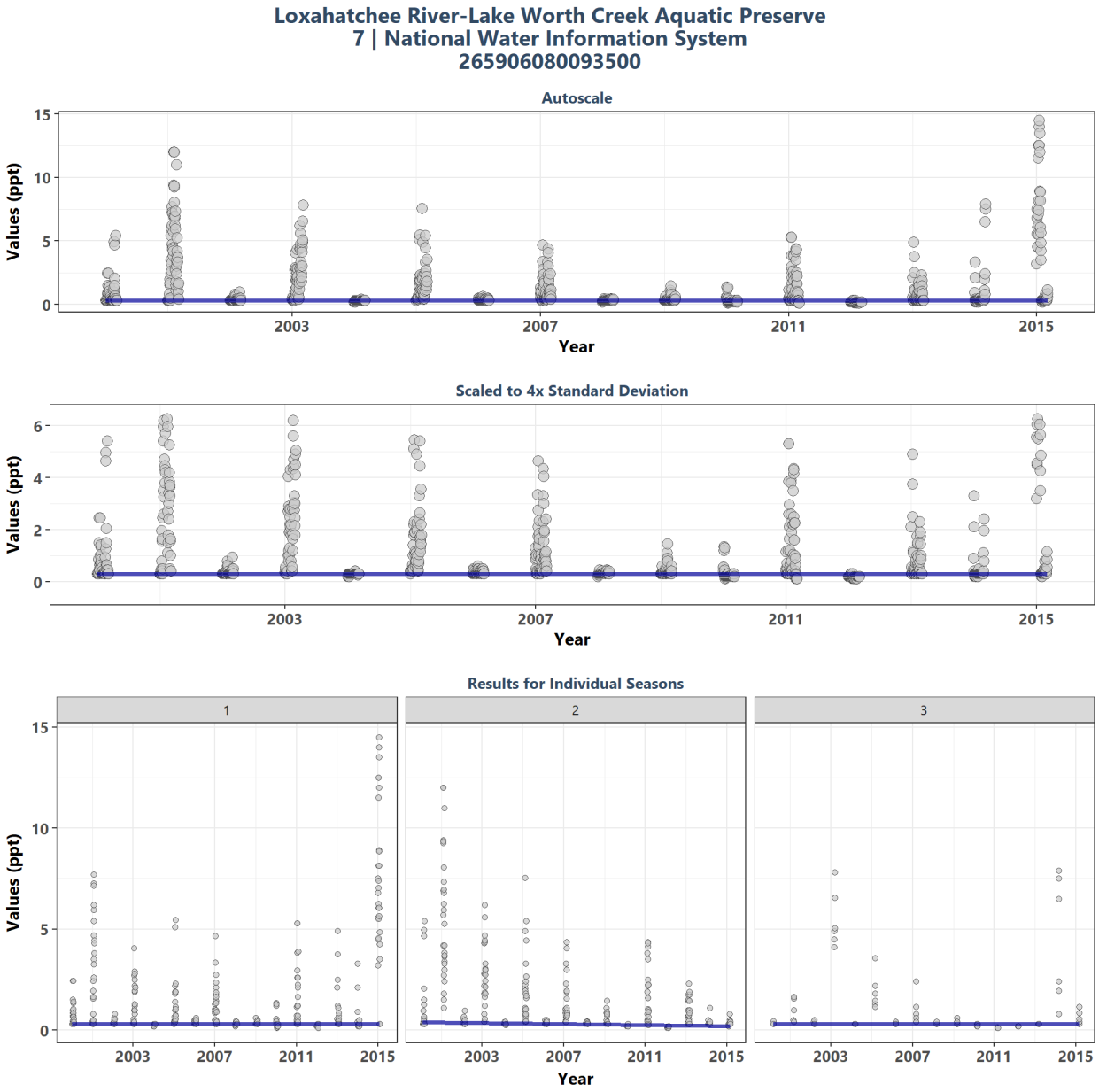






Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6687	33.82	-0.0964	-0.0505	34.2642	-11.6	0.0000	68.9	0	-1
1	552	33.47	-0.0961	-0.0493	33.9661	-3.4	0.0007	NA	NA	-1
2	513	32.86	-0.0872	-0.0444	33.3466	-3.0	0.0031	NA	NA	-1
3	592	33.37	-0.0665	-0.0396	33.8010	-2.4	0.0154	NA	NA	-1
4	524	33.97	-0.1452	-0.0804	34.7740	-5.0	0.0000	NA	NA	-1
5	529	35.07	-0.1178	-0.0587	35.6610	-4.1	0.0000	NA	NA	-1
6	576	34.89	-0.1351	-0.0624	35.5113	-4.9	0.0000	NA	NA	-1
7	572	34.98	-0.0999	-0.0385	35.3990	-3.6	0.0003	NA	NA	-1
8	547	34.07	-0.0314	-0.0158	34.2416	-1.1	0.2717	NA	NA	-1
9	581	33.22	0.0268	0.0165	33.0509	1.0	0.3335	NA	NA	1
10	590	32.47	-0.0099	-0.0076	32.5488	-0.4	0.7192	NA	NA	-1
11	527	33.24	-0.2143	-0.1046	34.2867	-7.4	0.0000	NA	NA	-1
12	584	33.36	-0.1950	-0.1031	34.3953	-7.1	0.0000	NA	NA	-1

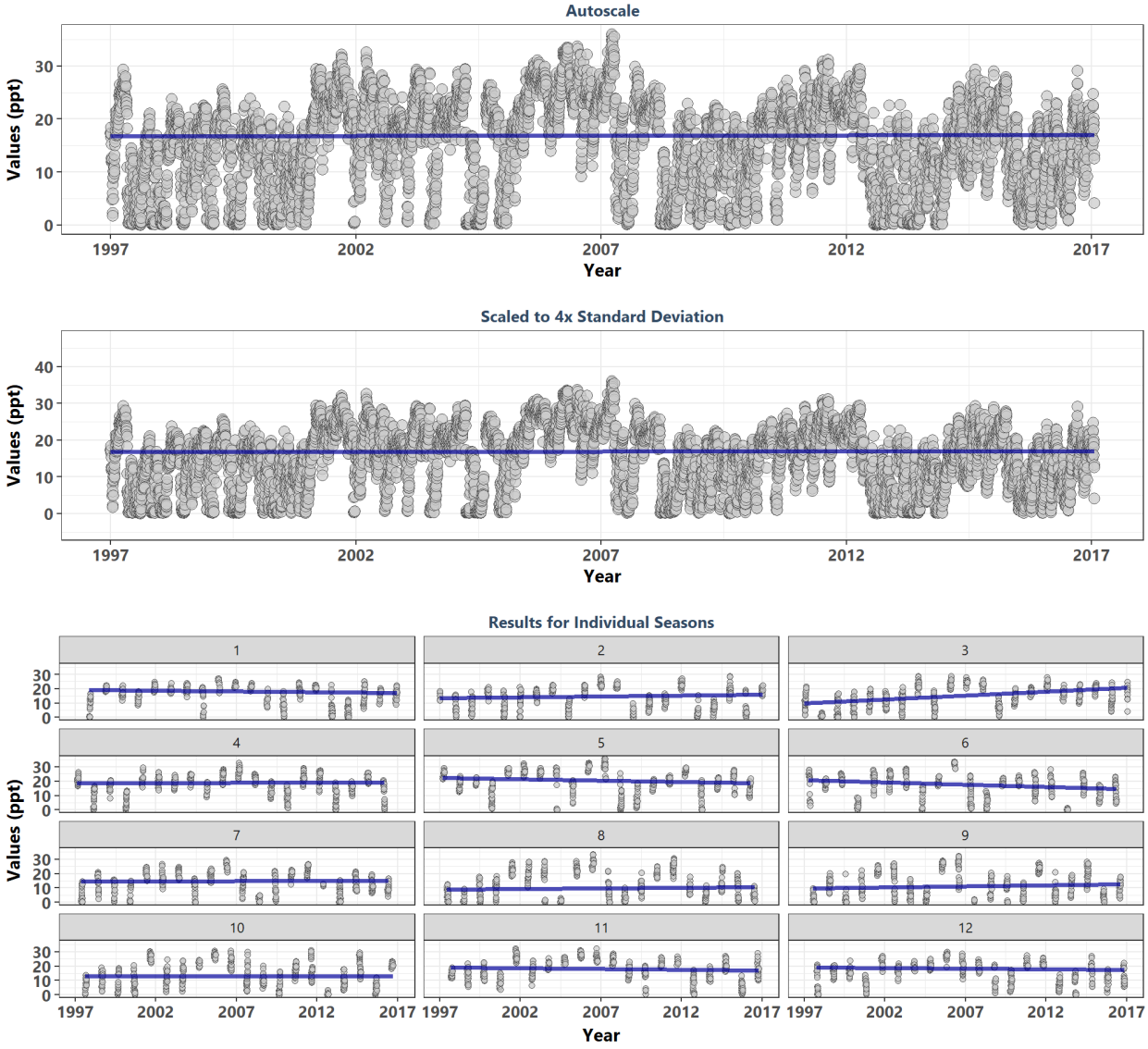
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1044	0.3	-0.1571	0.0000	0.3000	-7.5	0.0000	19.2	0.0001	-1
1	496	0.3	-0.0838	0.0000	0.3000	-2.9	0.0037	NA	NA	-1
2	452	0.3	-0.2395	-0.0125	0.4062	-7.9	0.0000	NA	NA	-1
3	96	0.3	-0.1478	0.0000	0.3000	-2.2	0.0275	NA	NA	-1

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

Pellicer Creek Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7059	16.64	0.0044	0.0082	16.8224	0.6	0.5331	170.2	0	0
1	587	18.02	-0.0675	-0.0935	19.0529	-2.4	0.0143	NA	NA	0
2	567	14.71	0.0573	0.1107	13.6011	2.0	0.0411	NA	NA	0
3	612	15.10	0.2673	0.5524	9.8535	9.9	0.0000	NA	NA	0
4	588	18.92	0.0024	0.0043	18.8760	0.1	0.9321	NA	NA	0
5	608	20.48	-0.0836	-0.1783	22.2610	-3.1	0.0020	NA	NA	0
6	567	17.58	-0.1657	-0.3259	20.8411	-5.9	0.0000	NA	NA	0
7	578	14.84	0.0054	0.0083	14.7688	0.2	0.8471	NA	NA	0
8	575	9.77	0.0545	0.0887	8.9729	2.0	0.0502	NA	NA	0
9	570	11.00	0.0742	0.1526	9.6279	2.7	0.0080	NA	NA	0
10	605	13.01	0.0029	0.0066	12.9434	0.1	0.9138	NA	NA	0
11	586	18.26	-0.0507	-0.1083	19.2331	-1.8	0.0663	NA	NA	0
12	616	18.25	-0.0499	-0.0931	19.0828	-1.9	0.0638	NA	NA	0

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

Appendix IV: Monitoring Location Summary Box Plots

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

1. Use the data set that only has `Use_In_Analysis` of `TRUE` for the desired monitoring location
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 `MonitoringID` with data grouped by `Year`, then `Year` and `Month`, then finally `Month` only. Each program 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

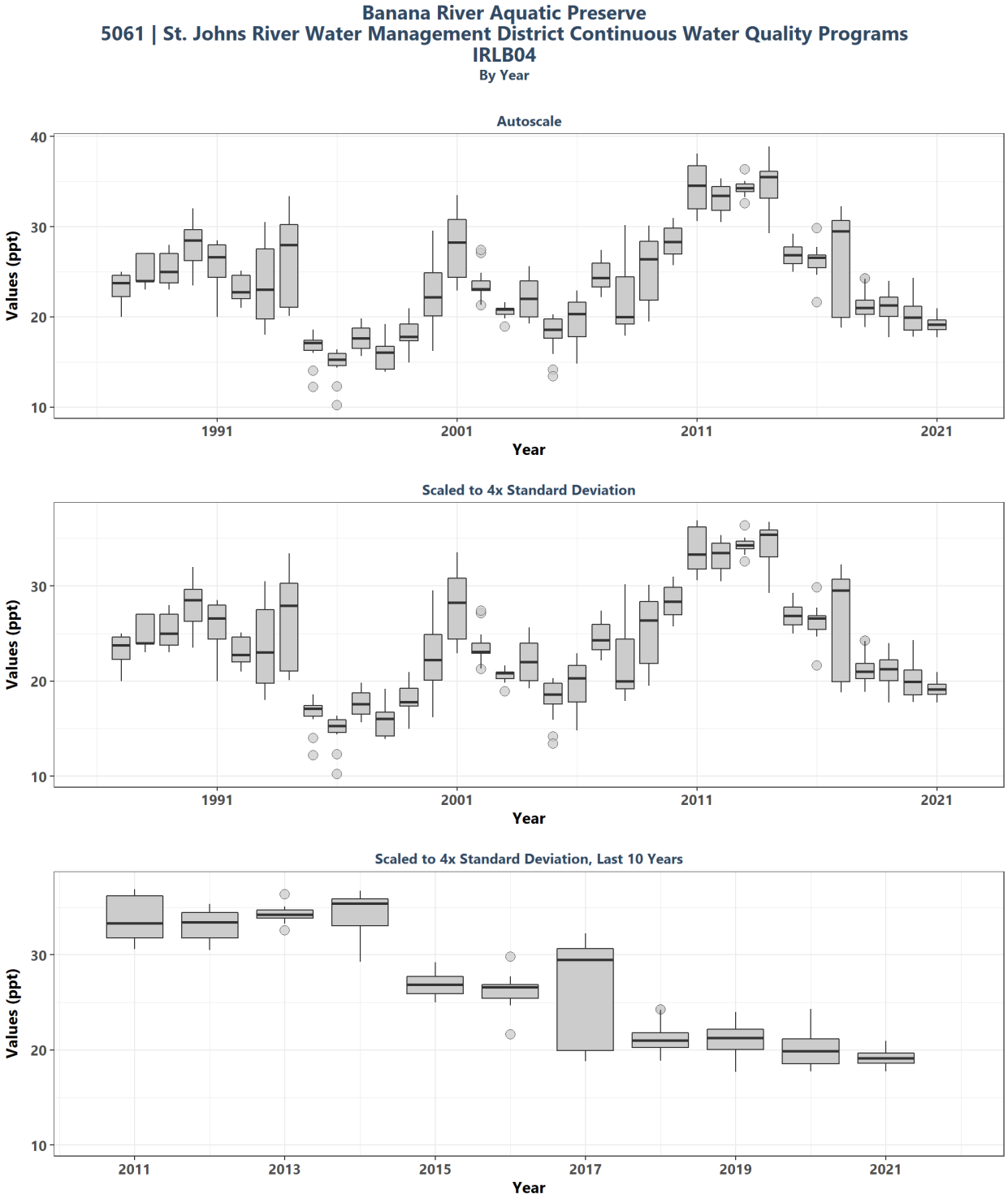
```
if(n==0){
   print("There are no monitoring locations that qualify.")
} else {
   for (i in 1:n) {
      year_lower <- min(data$Year[data$Use_In_Analysis==TRUE &
                                  data$MonitoringID==Mon_IDs[i]])
      year_upper <- max(data$Year[data$Use_In_Analysis==TRUE &
                                  data$MonitoringID==Mon_IDs[i]])
      min_RV <- min(data$ResultValue[data$Use_In_Analysis==TRUE &
                                     data$MonitoringID==Mon_IDs[i]])
      mn_RV <- mean(data$ResultValue[data$Use_In_Analysis==TRUE &
                                     data$MonitoringID==Mon_IDs[i] &
                                     data$ResultValue <
                                     quantile(data$ResultValue, 0.98)])
      sd_RV <- sd(data$ResultValue[data$Use_In_Analysis==TRUE &
                                   data$MonitoringID==Mon_IDs[i] &
                                   data$ResultValue <
                                   quantile(data$ResultValue, 0.98)])
      x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
      y_scale <- mn_RV + 4 * sd_RV
      MA_name <- KT.Stats$ManagedAreaName[KT.Stats$MonitoringID==Mon_IDs[i]]
      Mon_name <- paste0(KT.Stats$ProgramID[KT.Stats$MonitoringID==Mon_IDs[i]],
         " | ", KT.Stats$ProgramName[KT.Stats$MonitoringID==Mon_IDs[i]], "\n",
         KT.Stats$ProgramLocationID[KT.Stats$MonitoringID==Mon_IDs[i]])

      ##Year plots
      p1 <- ggplot(data=data[data$Use_In_Analysis==TRUE &
                             data$MonitoringID==Mon_IDs[i], ],
```

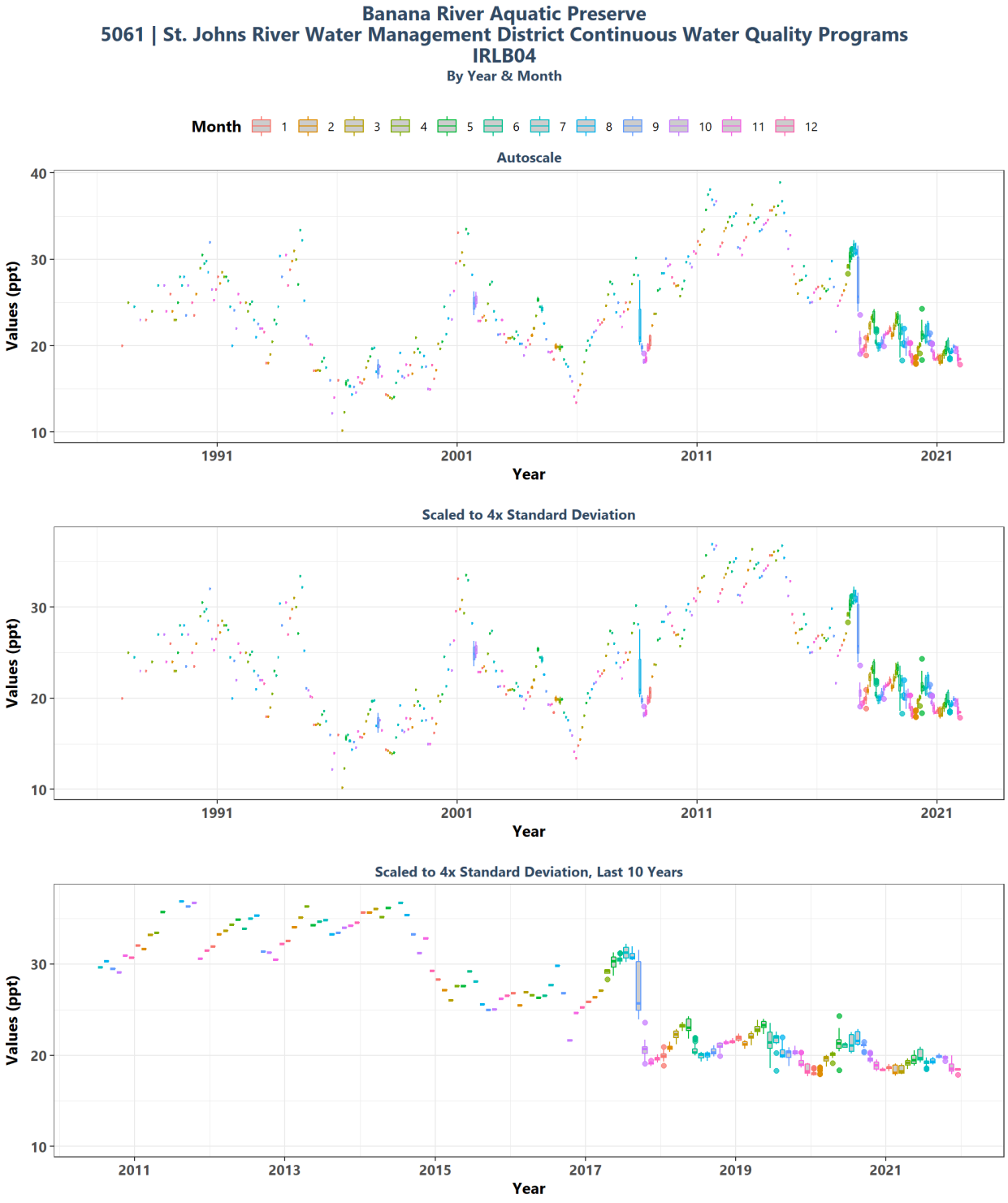
```
                aes(x=Year, y=ResultValue, group=Year)) +
    geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                 outlier.size=3, outlier.color="#333333",
                 outlier.fill="#cccccc", outlier.alpha=0.75) +
    labs(subtitle="Autoscale",
         x="Year", 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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i], ],
                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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i] &
                           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_name, "\n", Mon_name),
                        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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i], ],
                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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i], ],
             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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i], ],
             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_name, "\n", Mon_name),
                       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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i], ],
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
```

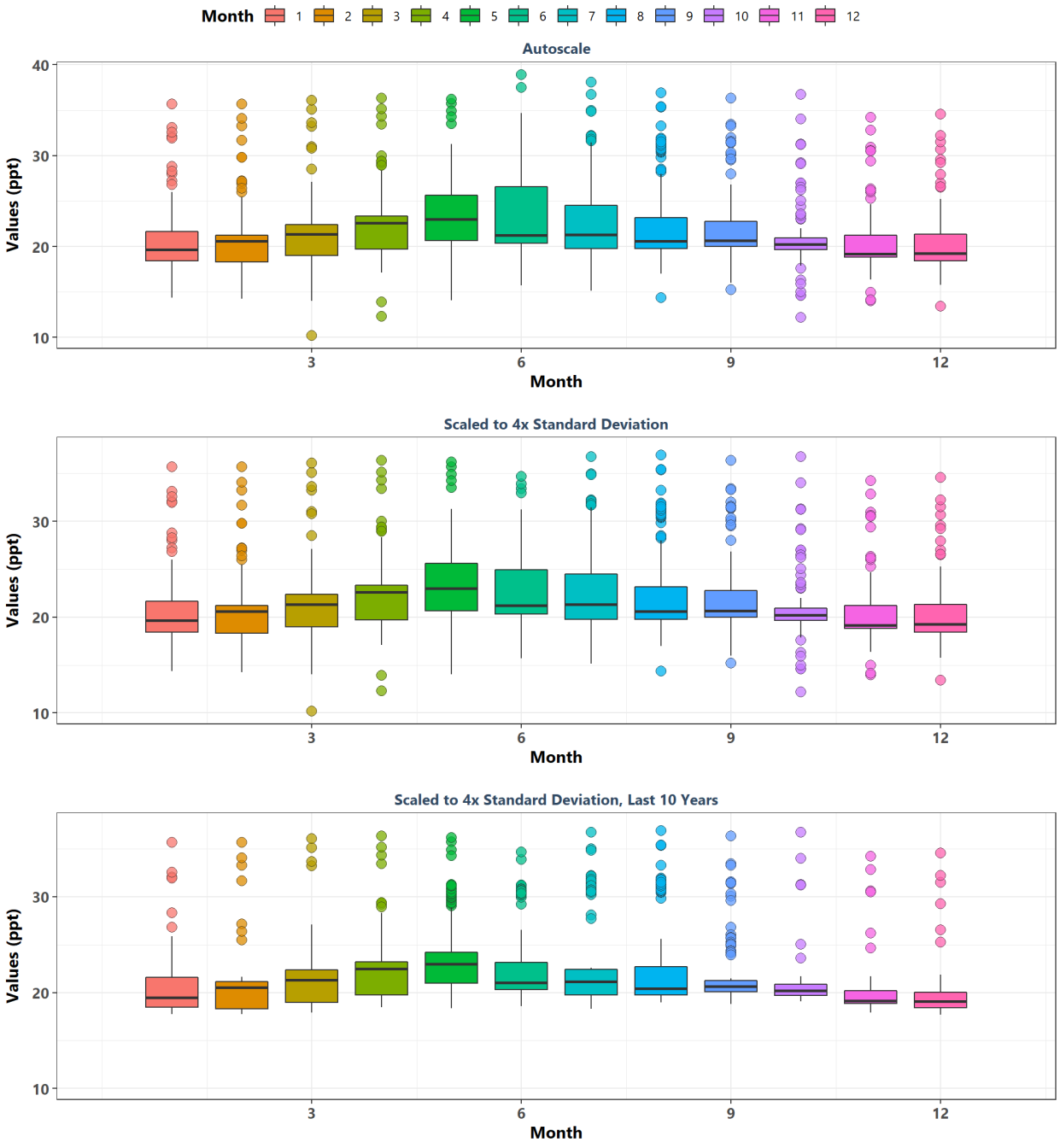
```
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
   scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
   plot_theme +
   theme(legend.position="none")

p8 <- ggplot(data=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i], ],
             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=data[data$Use_In_Analysis==TRUE &
                           data$MonitoringID==Mon_IDs[i] &
                           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_name, "\n", Mon_name),
                        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(0.1, 1)))
print(ggarrange(p00, YMset, ncol=1, heights=c(0.1, 1)))
print(ggarrange(p000, Mset, ncol=1, heights=c(0.1, 1)))

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

```
    }
}
```

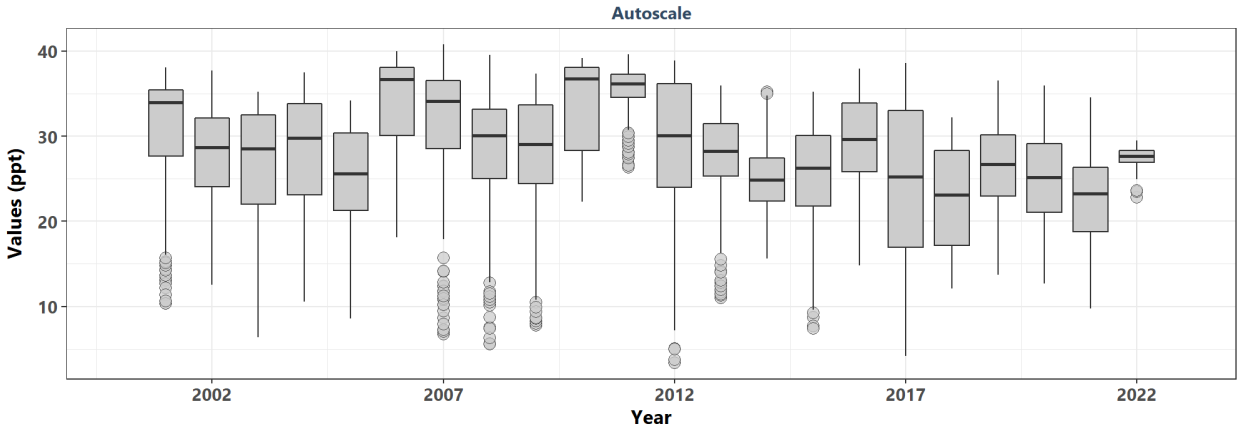

Banana River Aquatic Preserve
5061 | St. Johns River Water Management District Continuous Water Quality Programs
IRLB04
By Year
Autoscale
Values (ppt)
40
30
20
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
30
20
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
30
20
10
2011
2013
2015
2017
2019
2021
Year

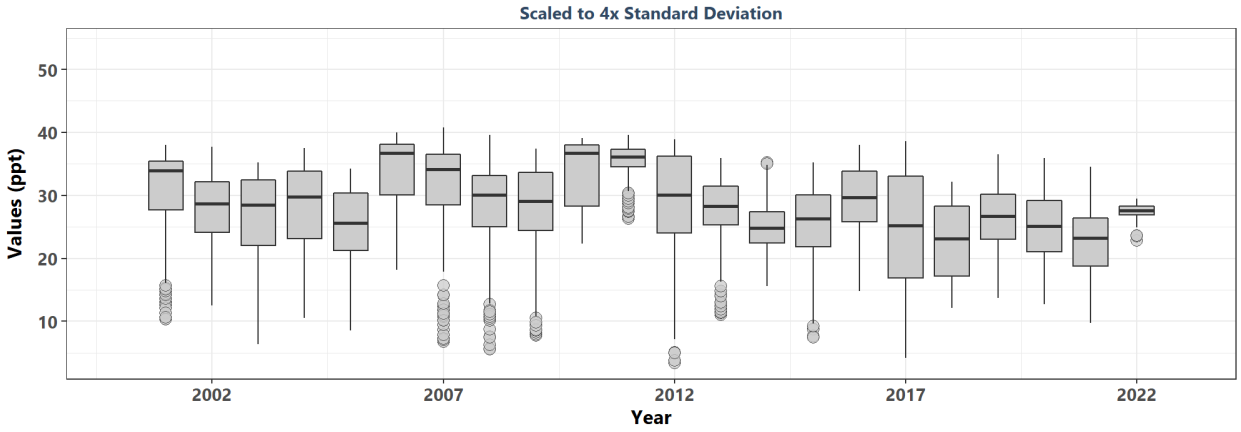

Banana River Aquatic Preserve
5061 | St. Johns River Water Management District Continuous Water Quality Programs
IRLB04
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
40
30
20
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (ppt)
30
20
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
30
20
10
2011
2013
2015
2017
2019
2021
Year

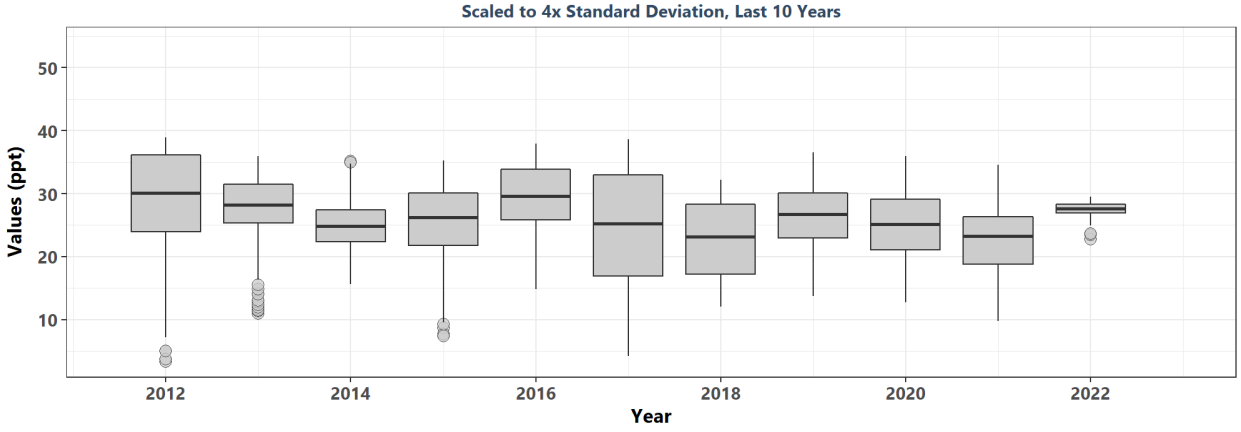
Banana River Aquatic Preserve
5061 | St. Johns River Water Management District Continuous Water Quality Programs
IRLB04
By Month



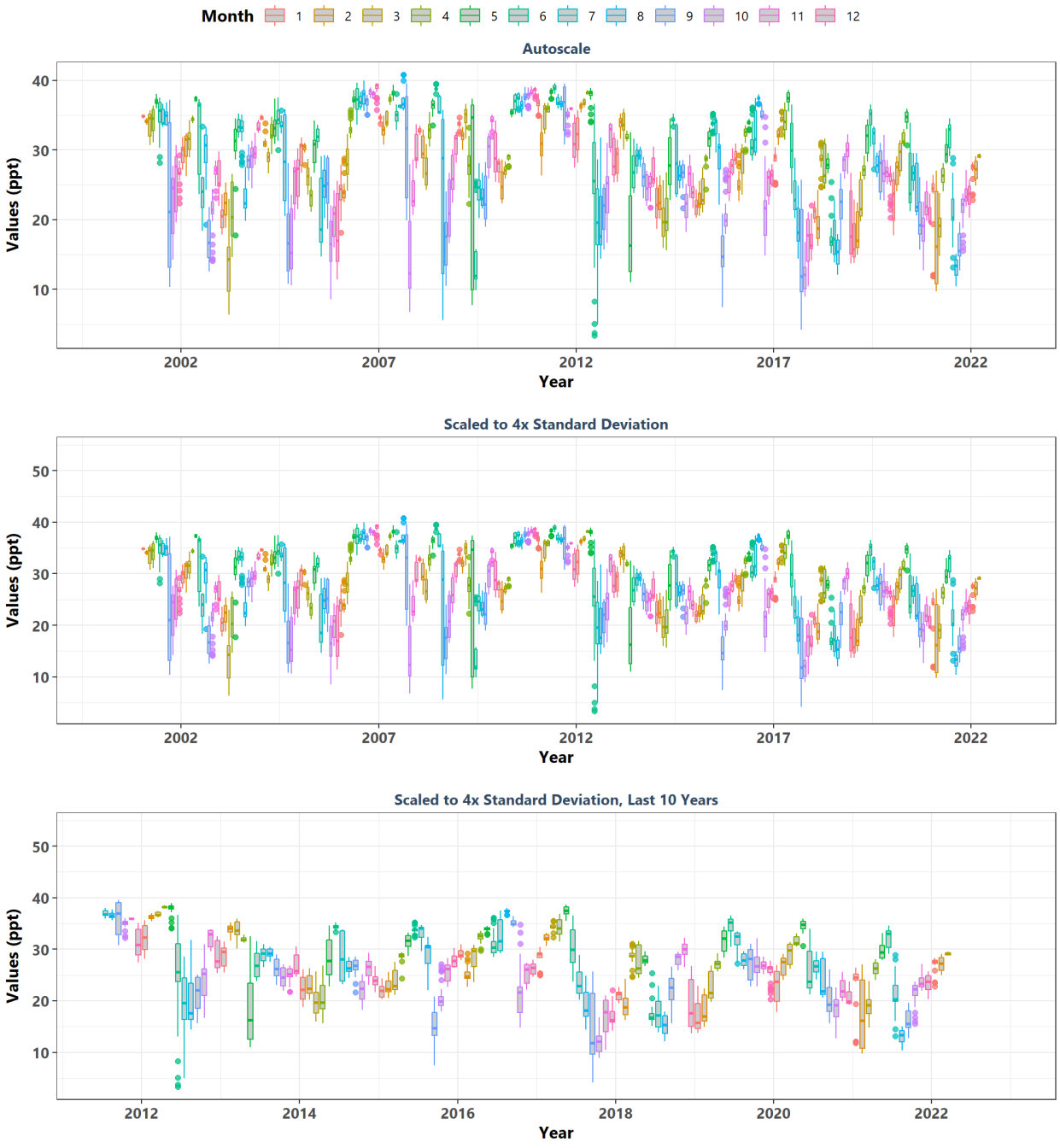
Guana River Marsh Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq
By Year



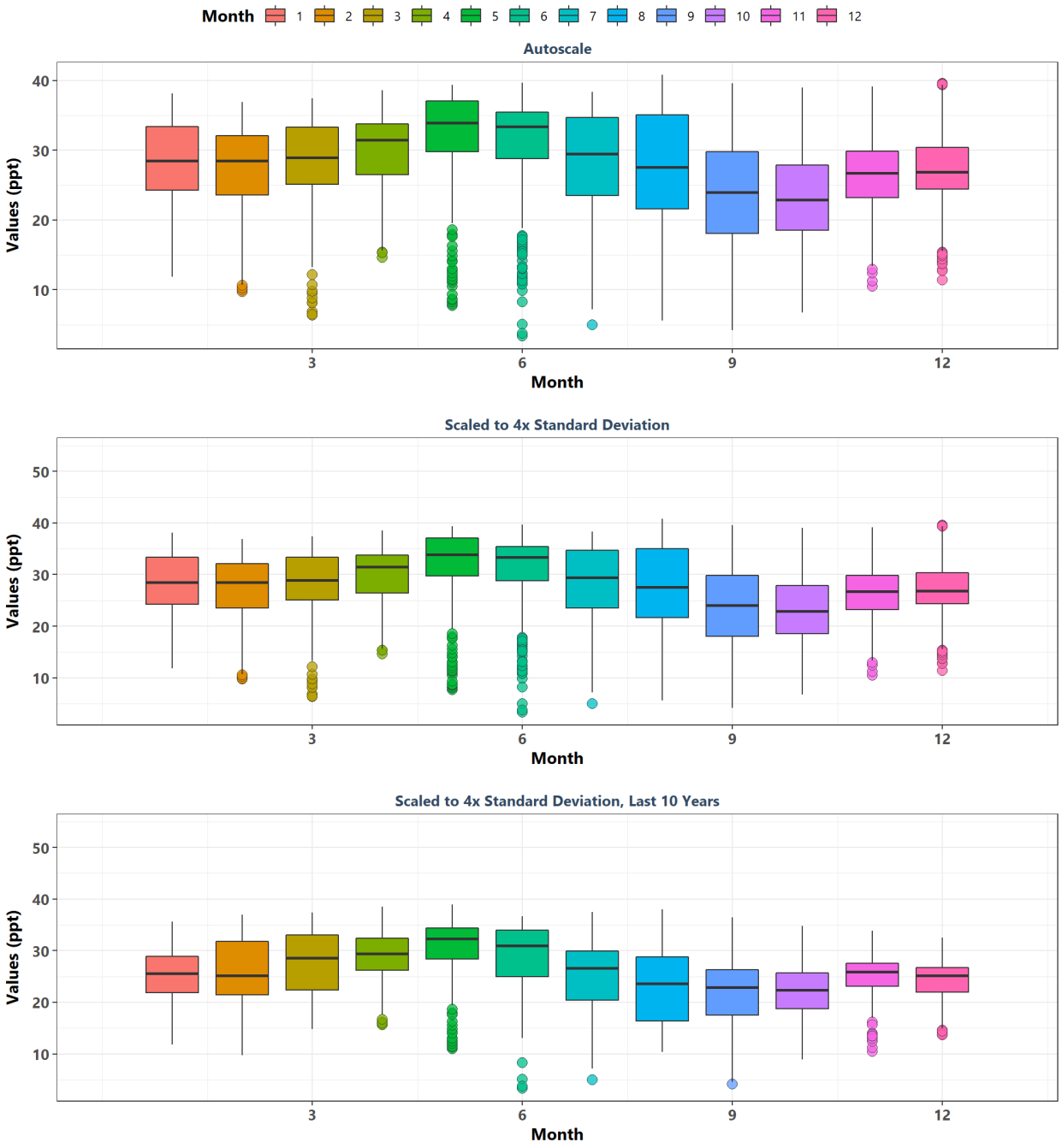




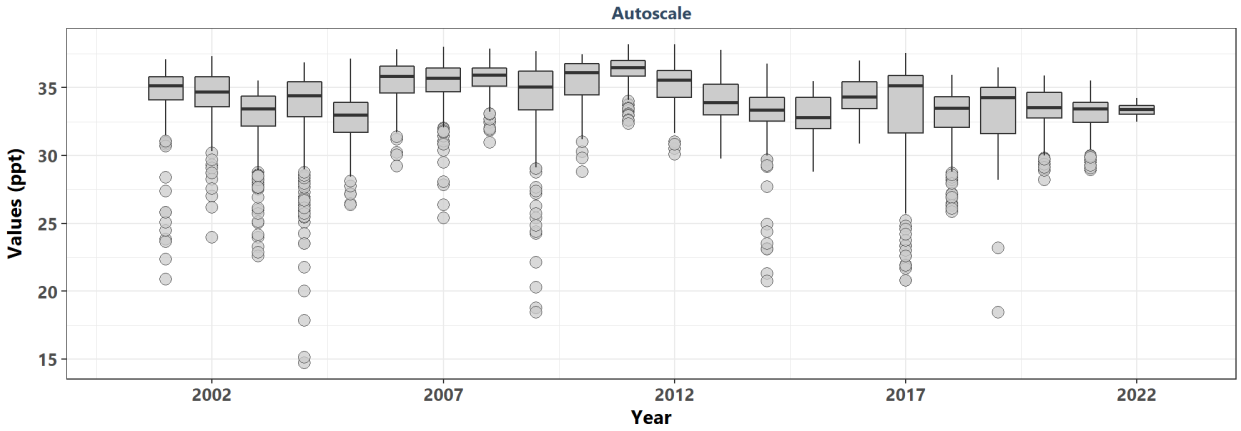
Guana River Marsh Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq
By Year & Month

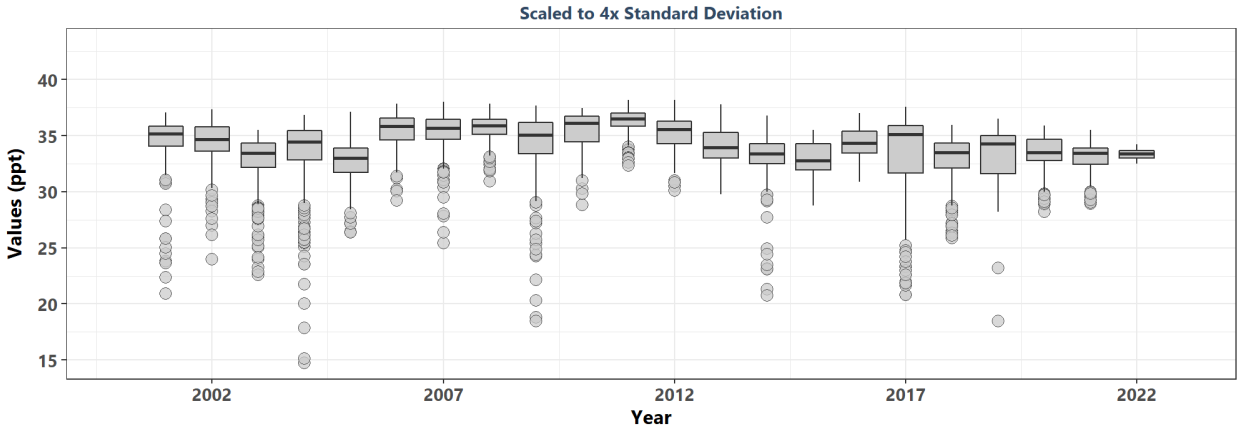


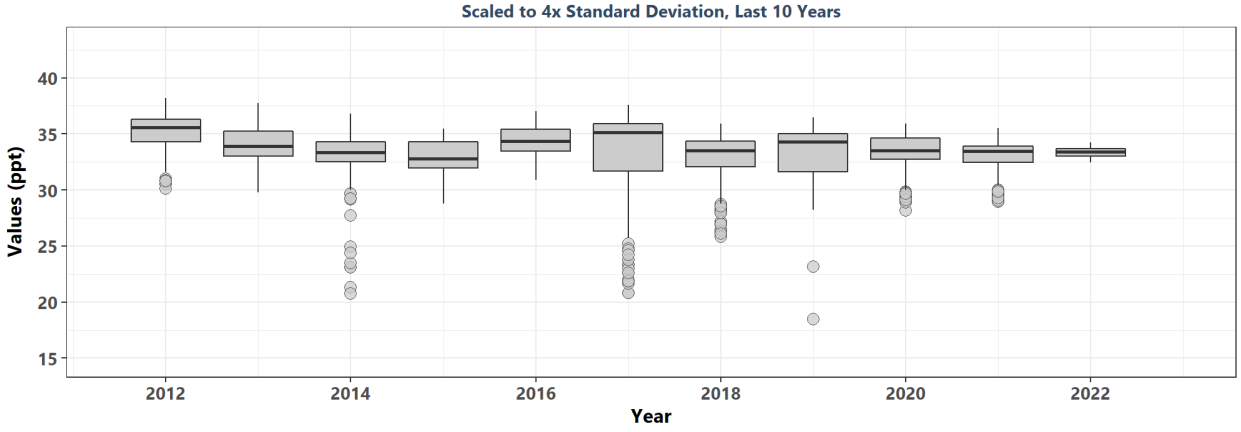
Guana River Marsh Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq
By Month

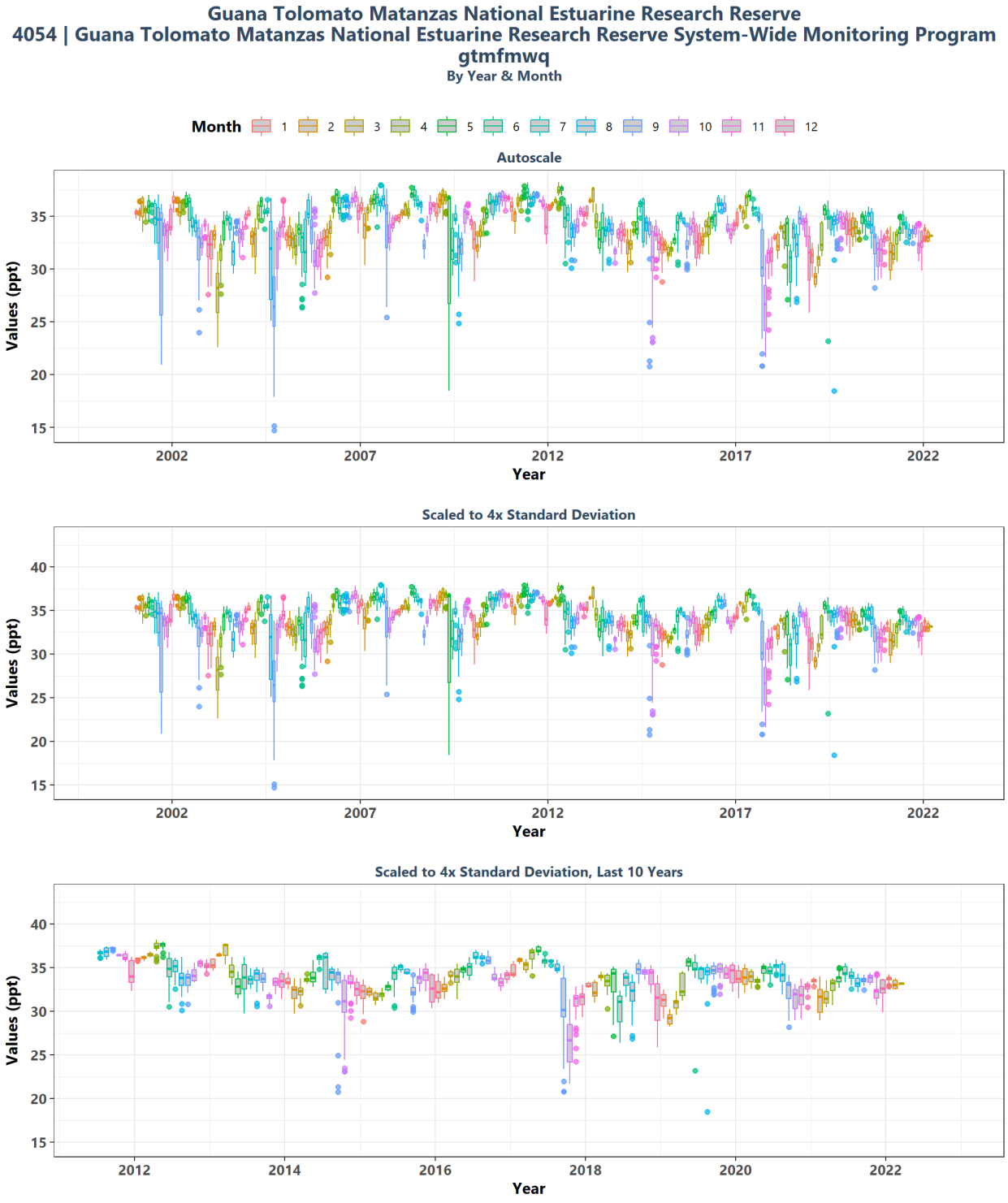


Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmfmwq
By Year

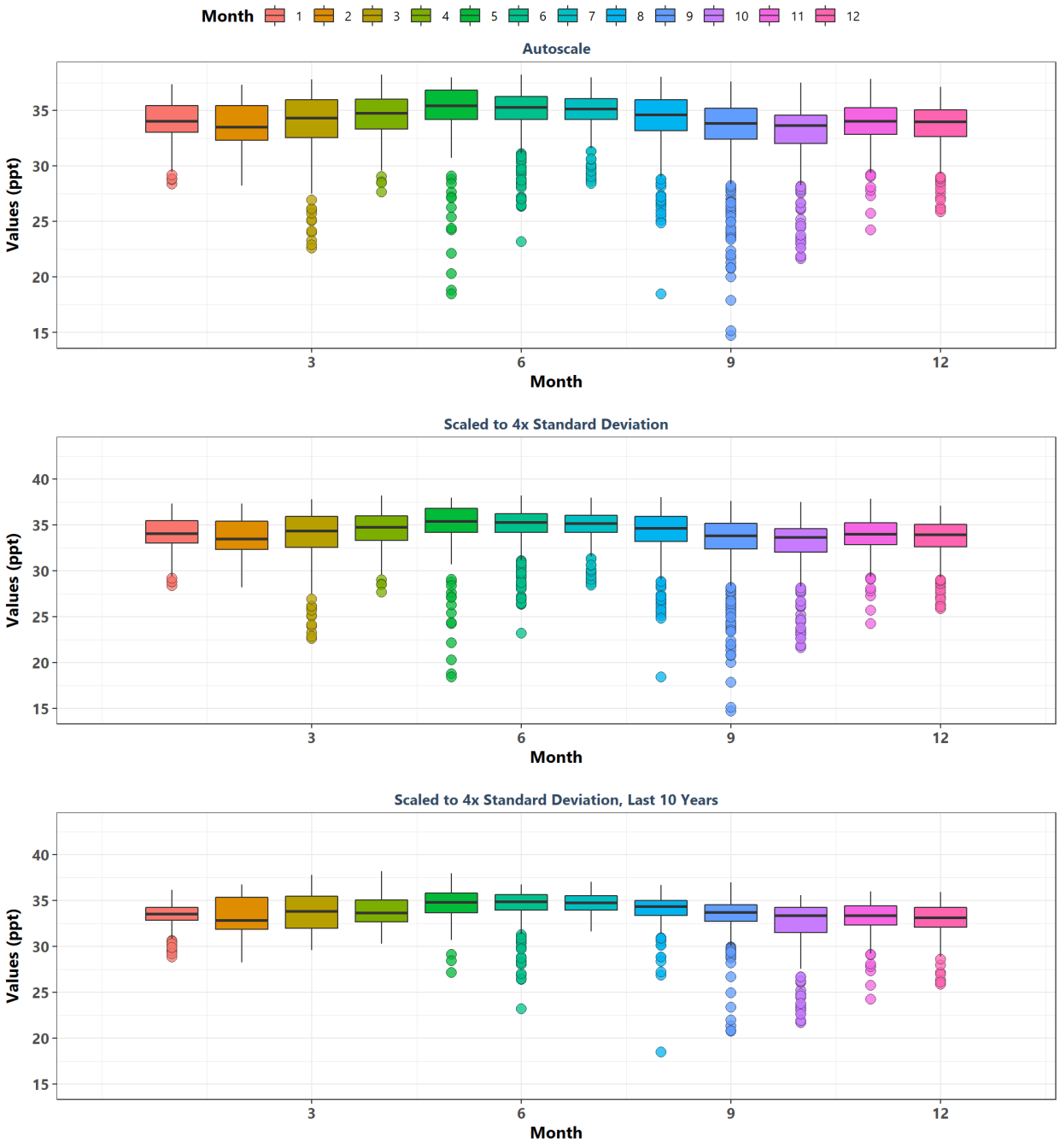




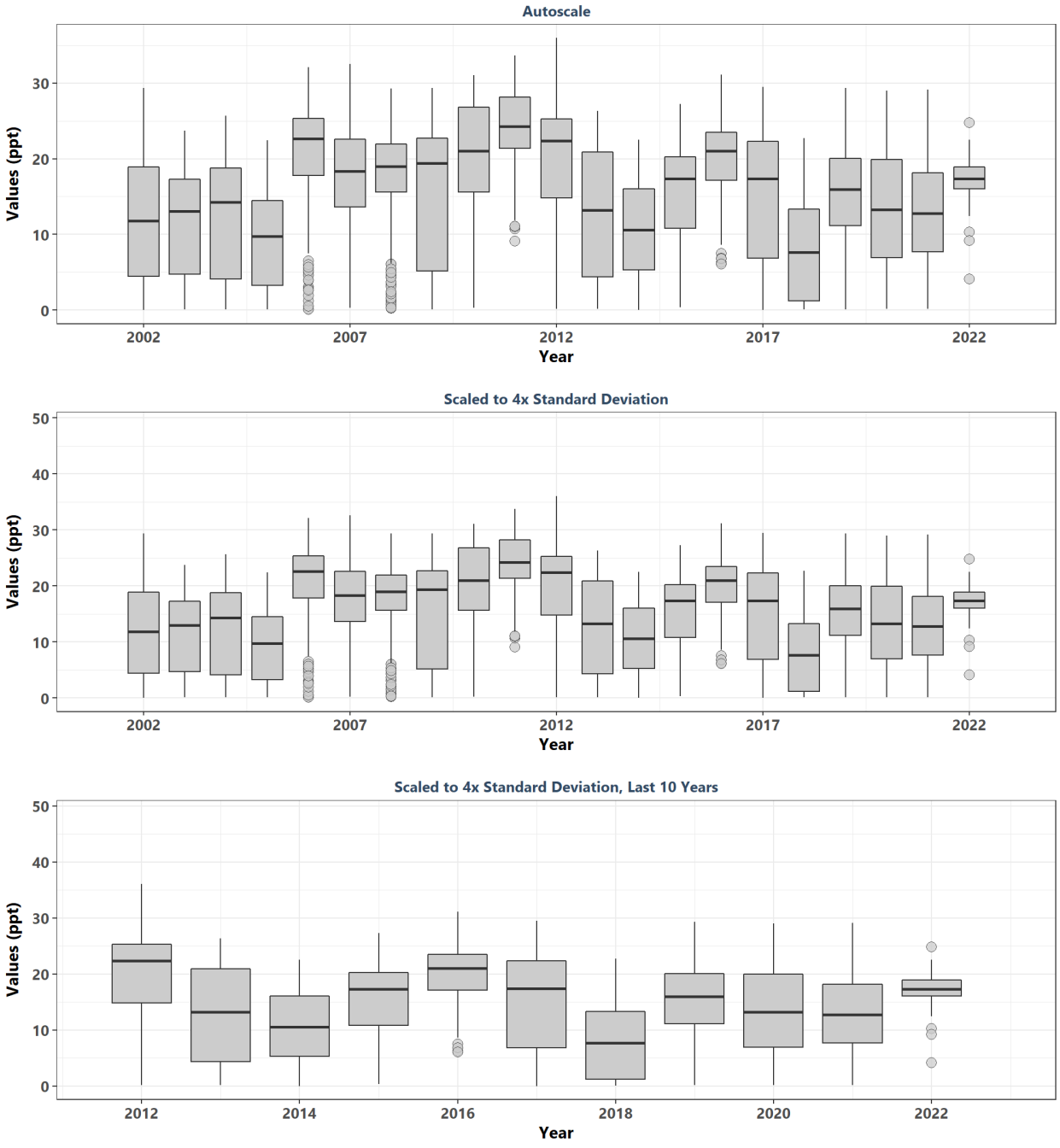


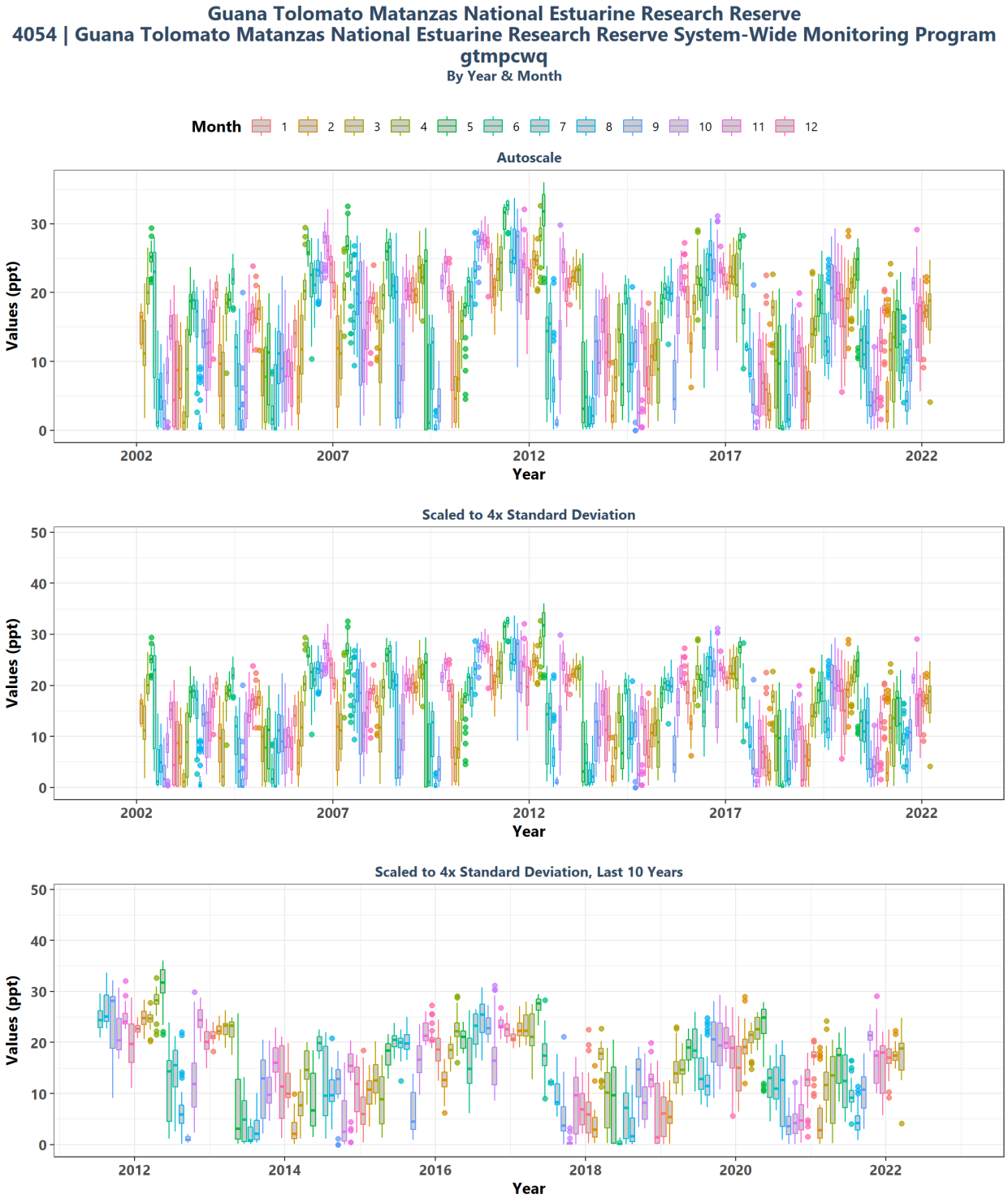

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmfmwq
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmfmwq
By Month

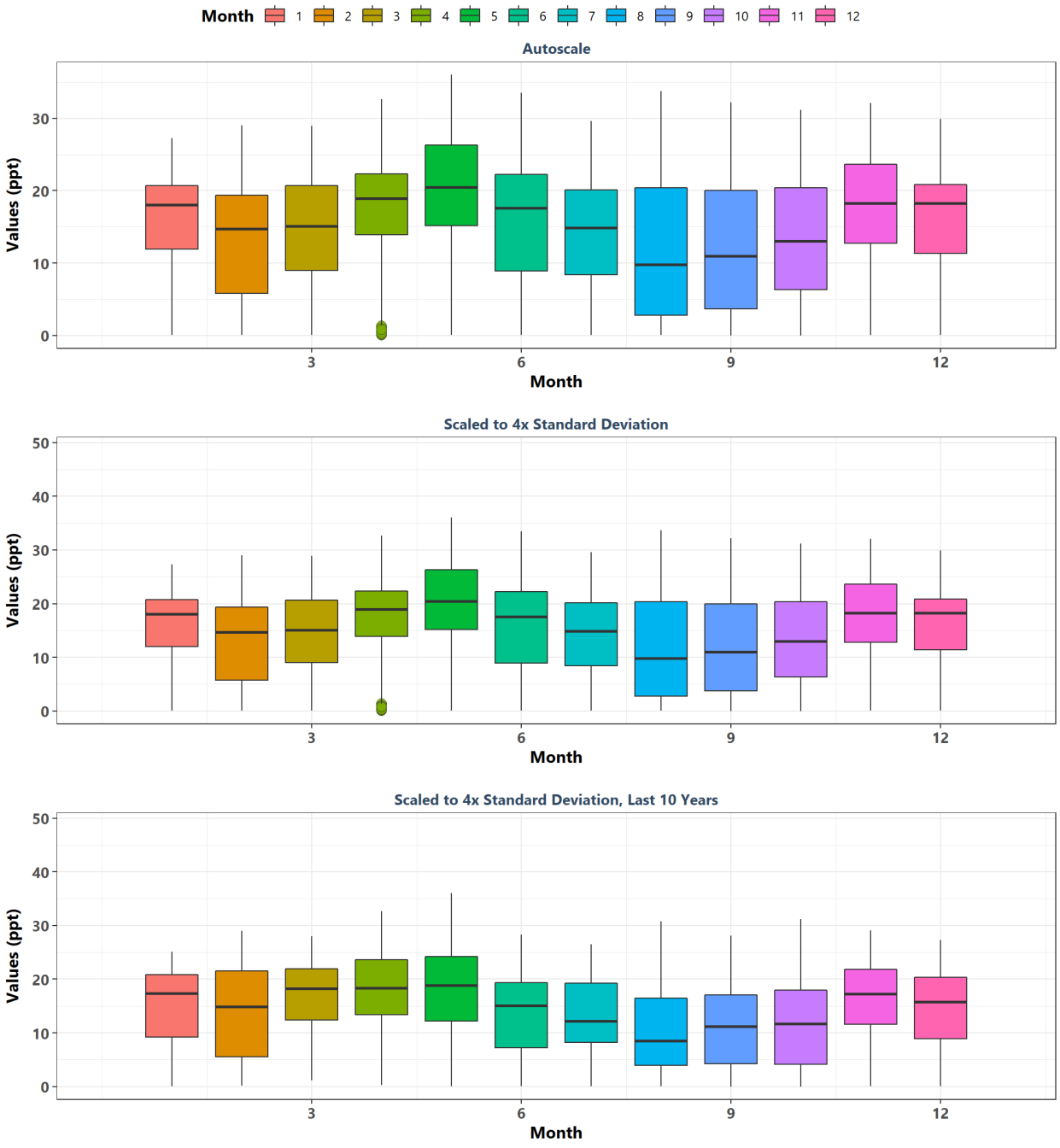


Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq
By Year

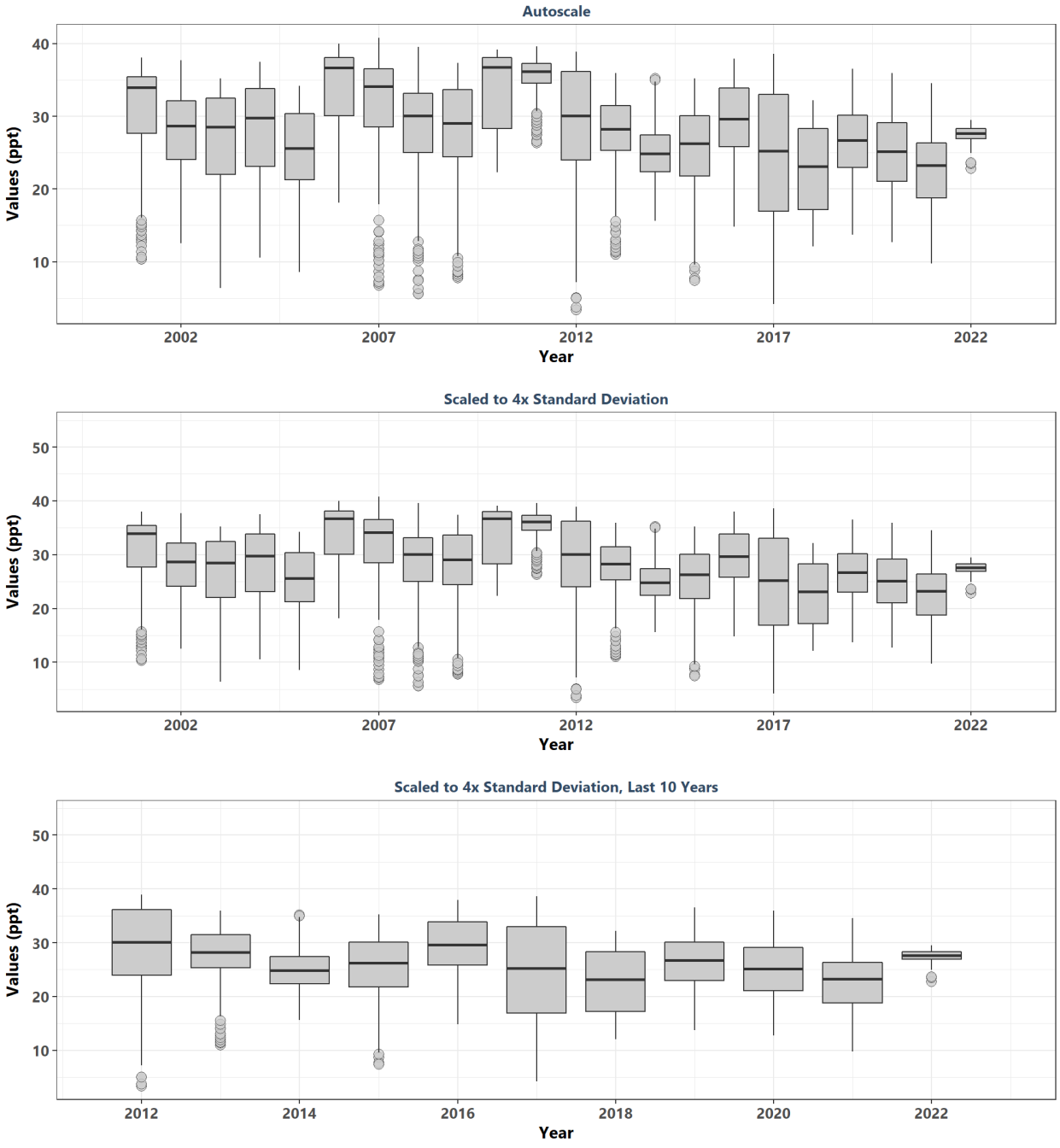


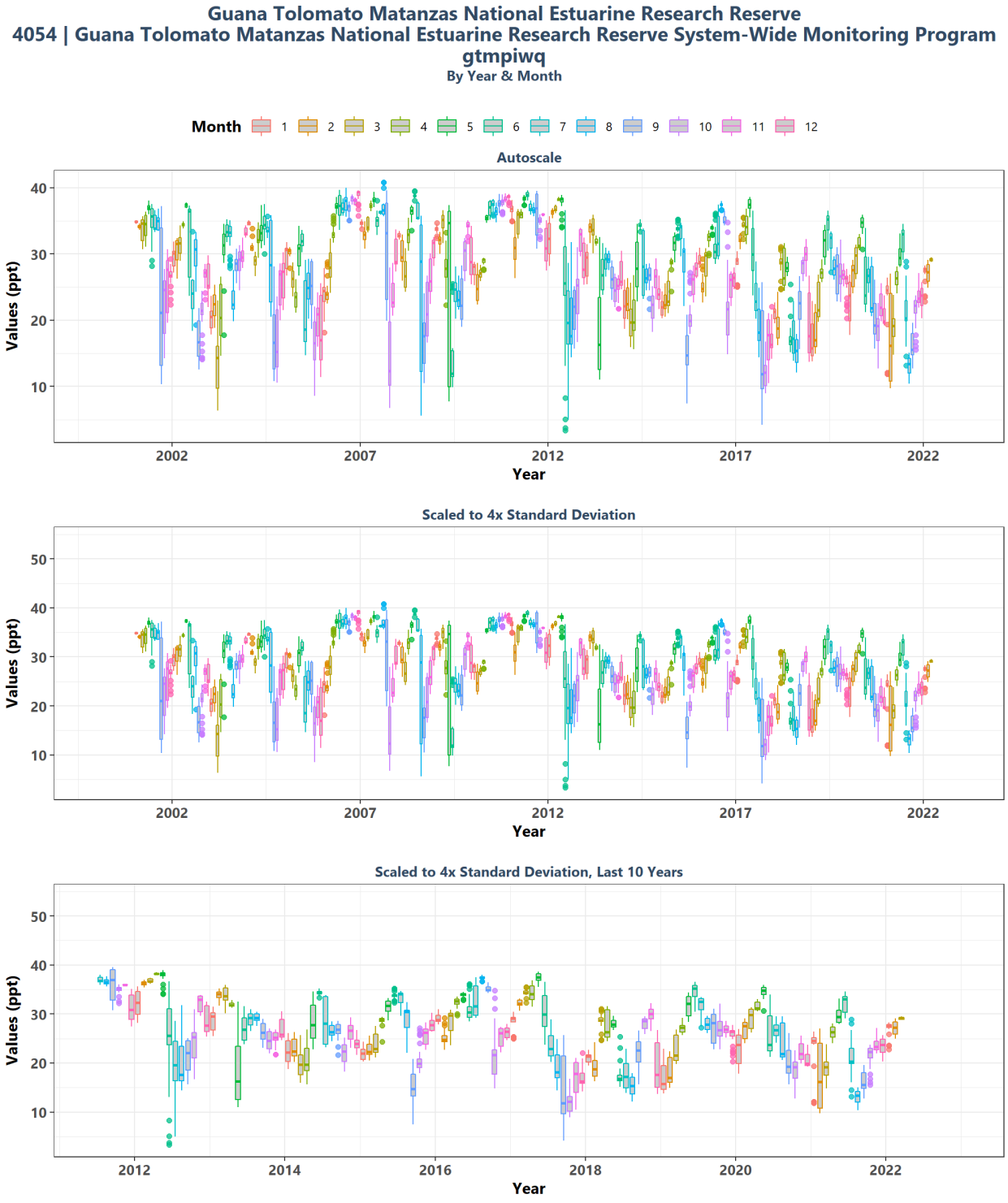

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq
By Month

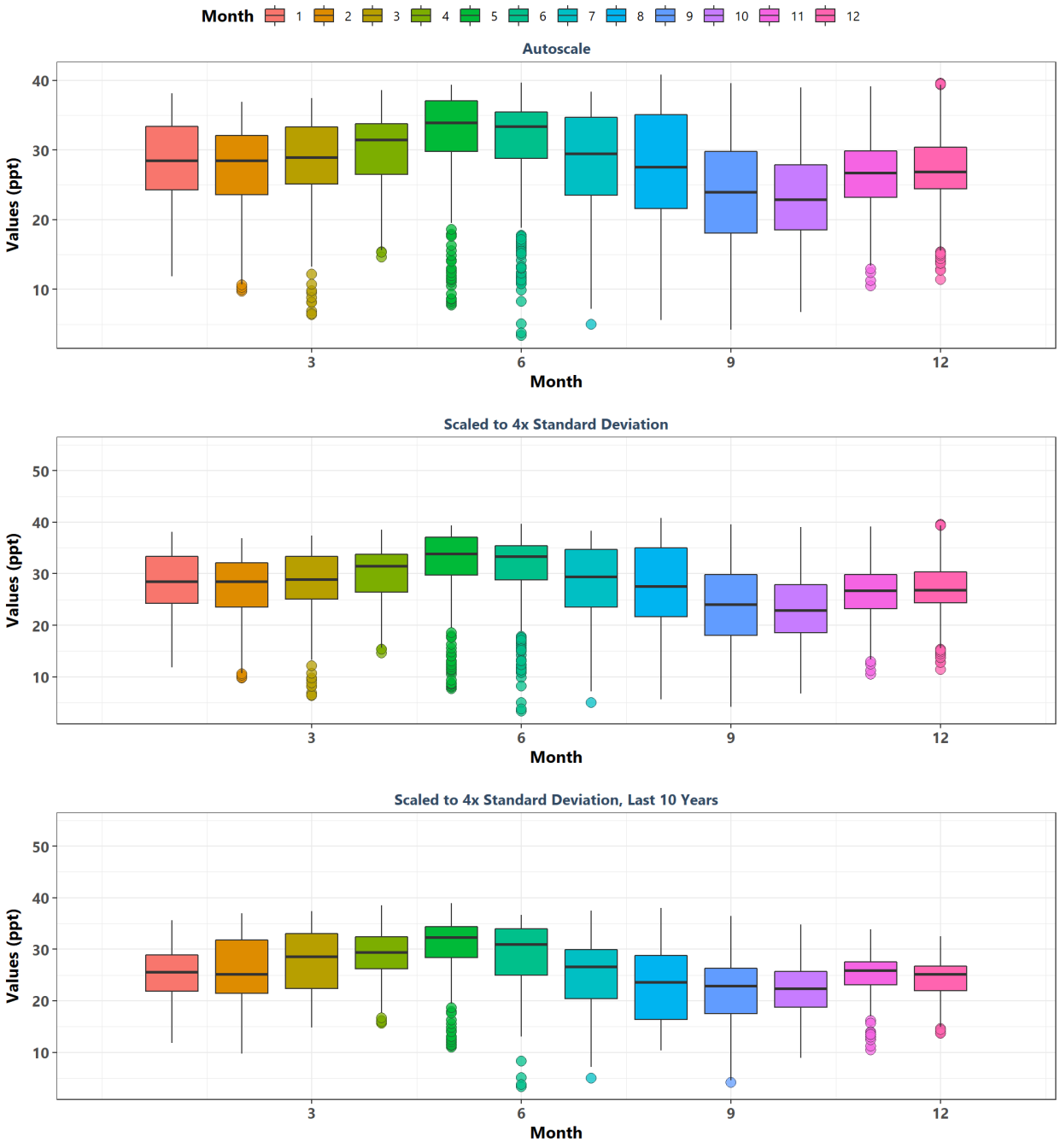


Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq
By Year

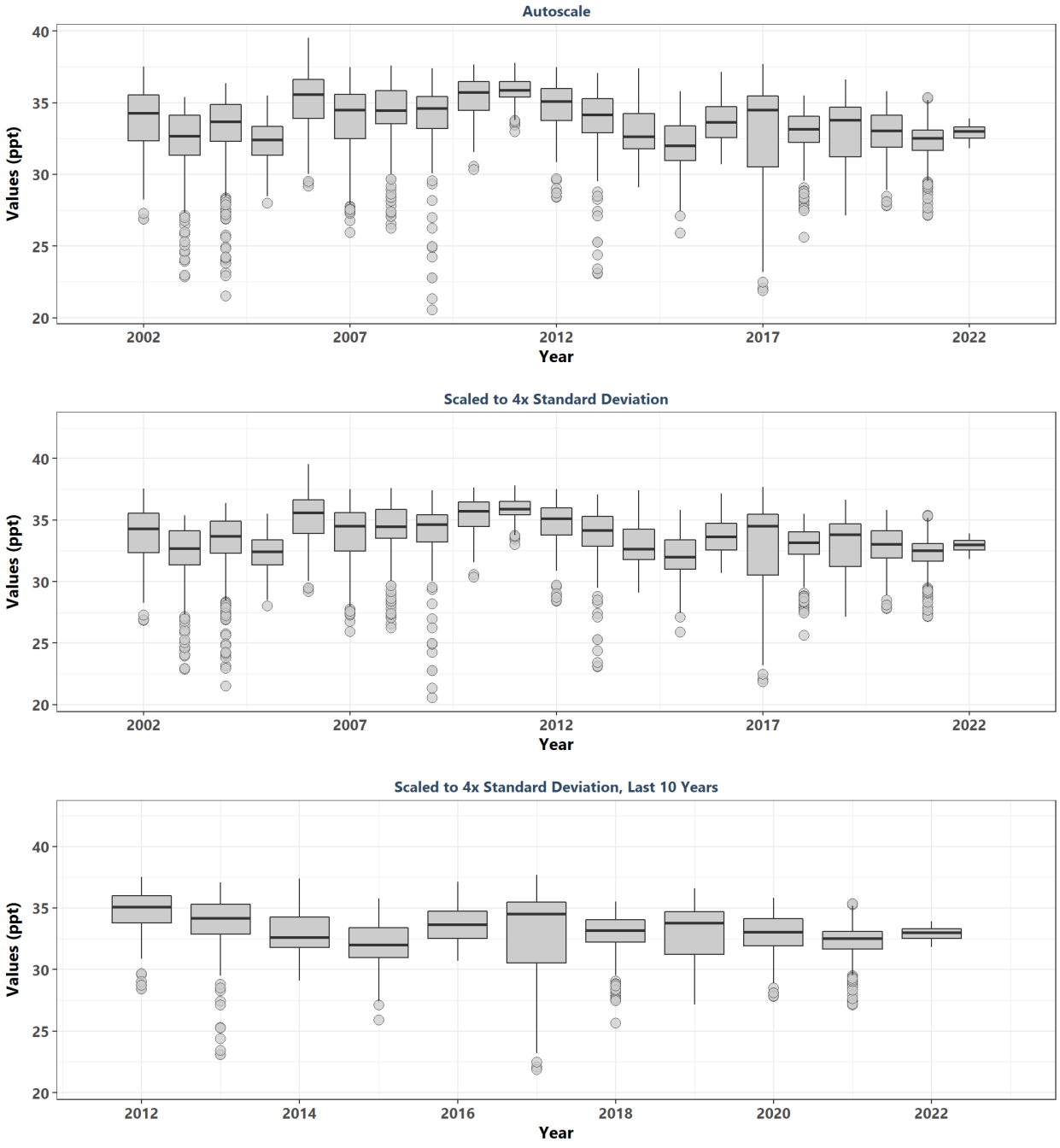


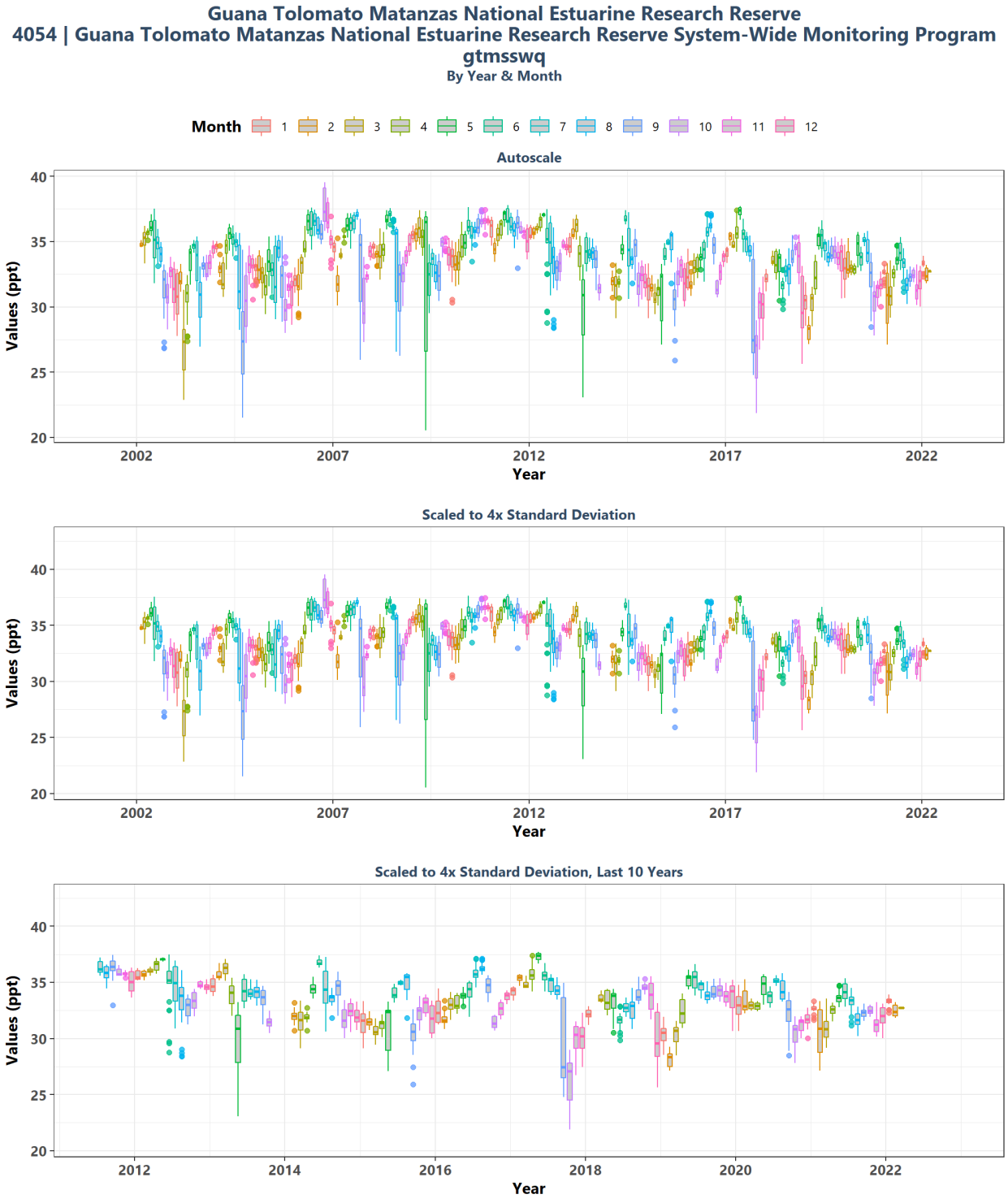

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpiwq
By Month

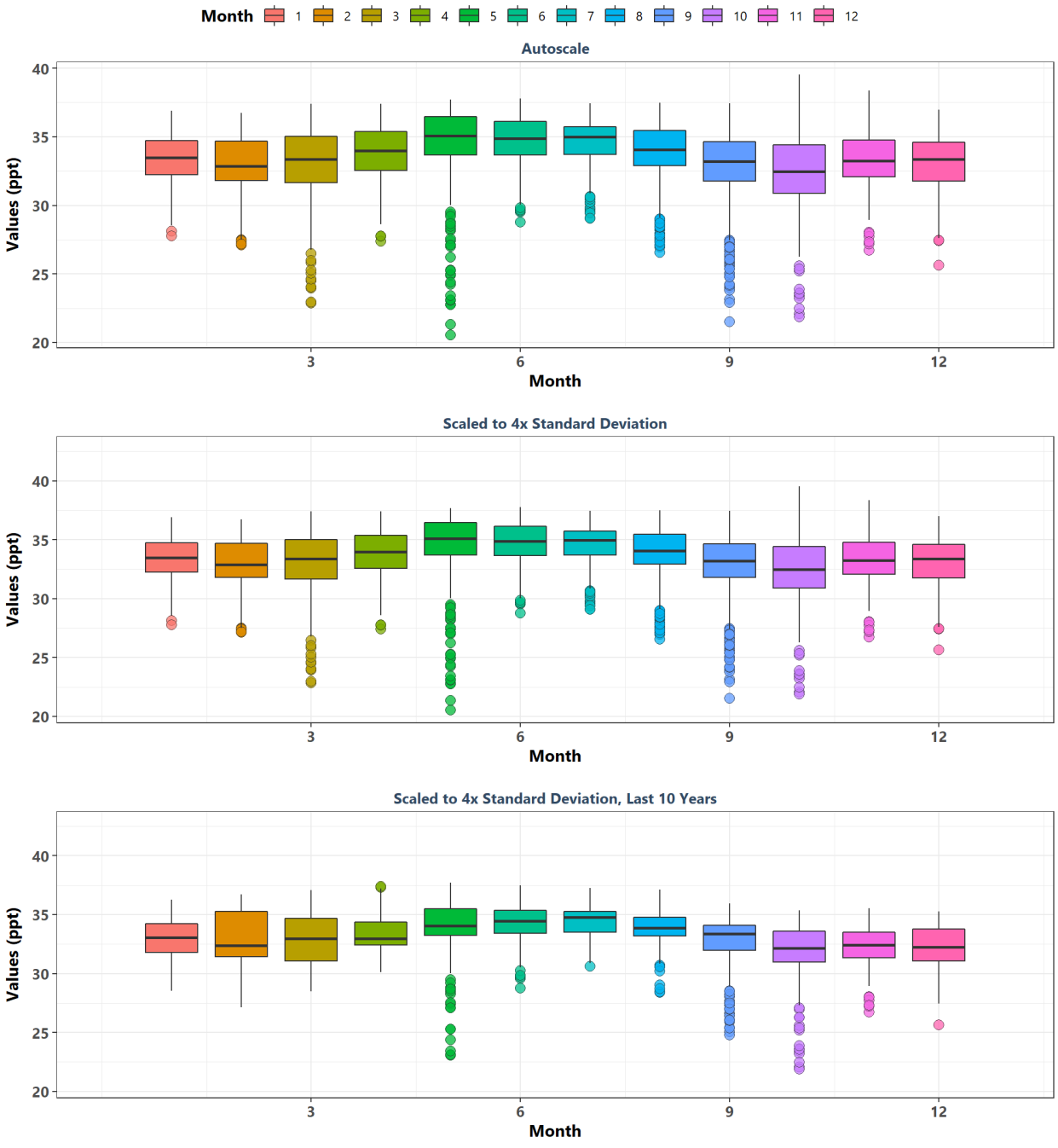


Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmsswq
By Year

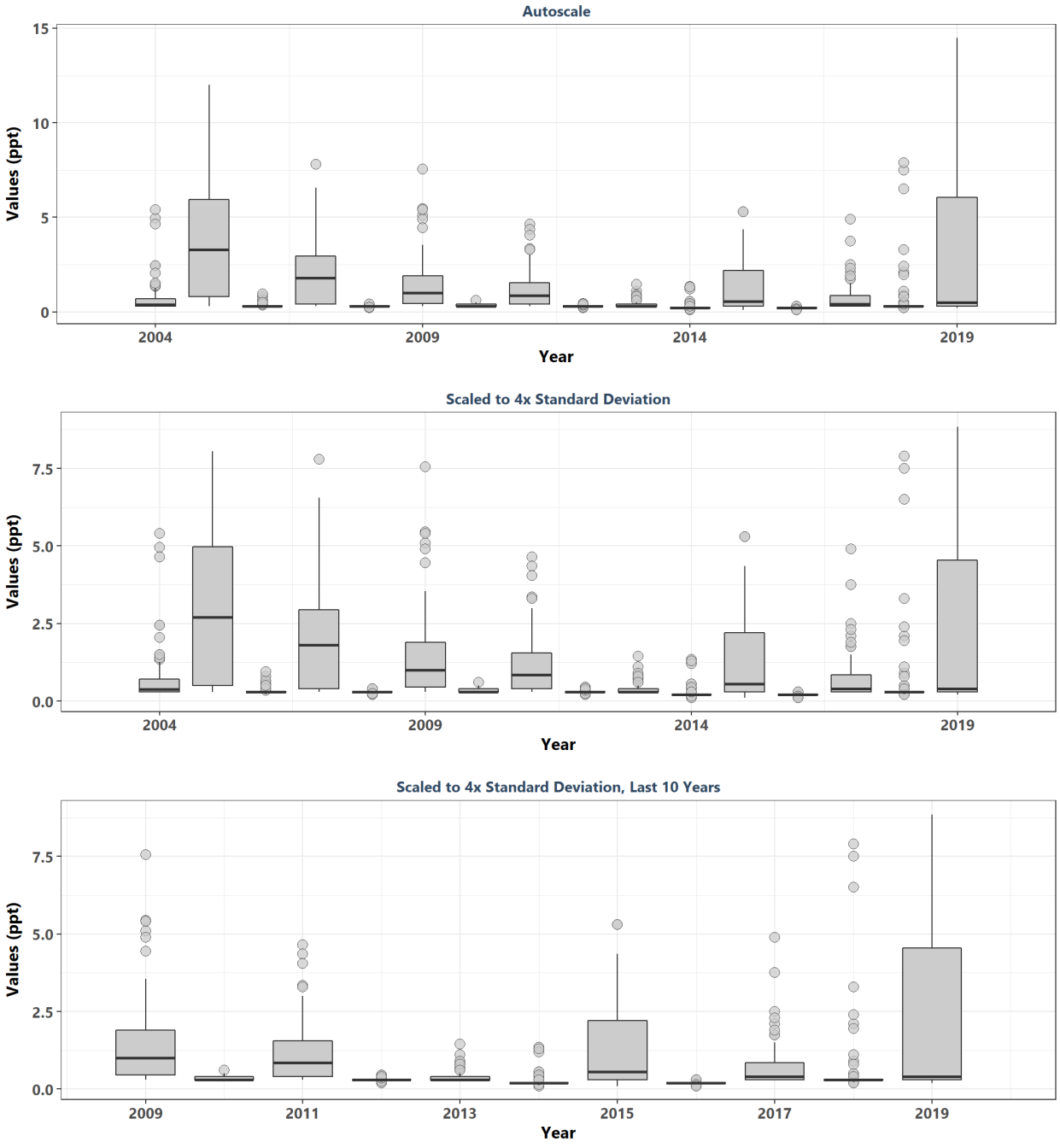


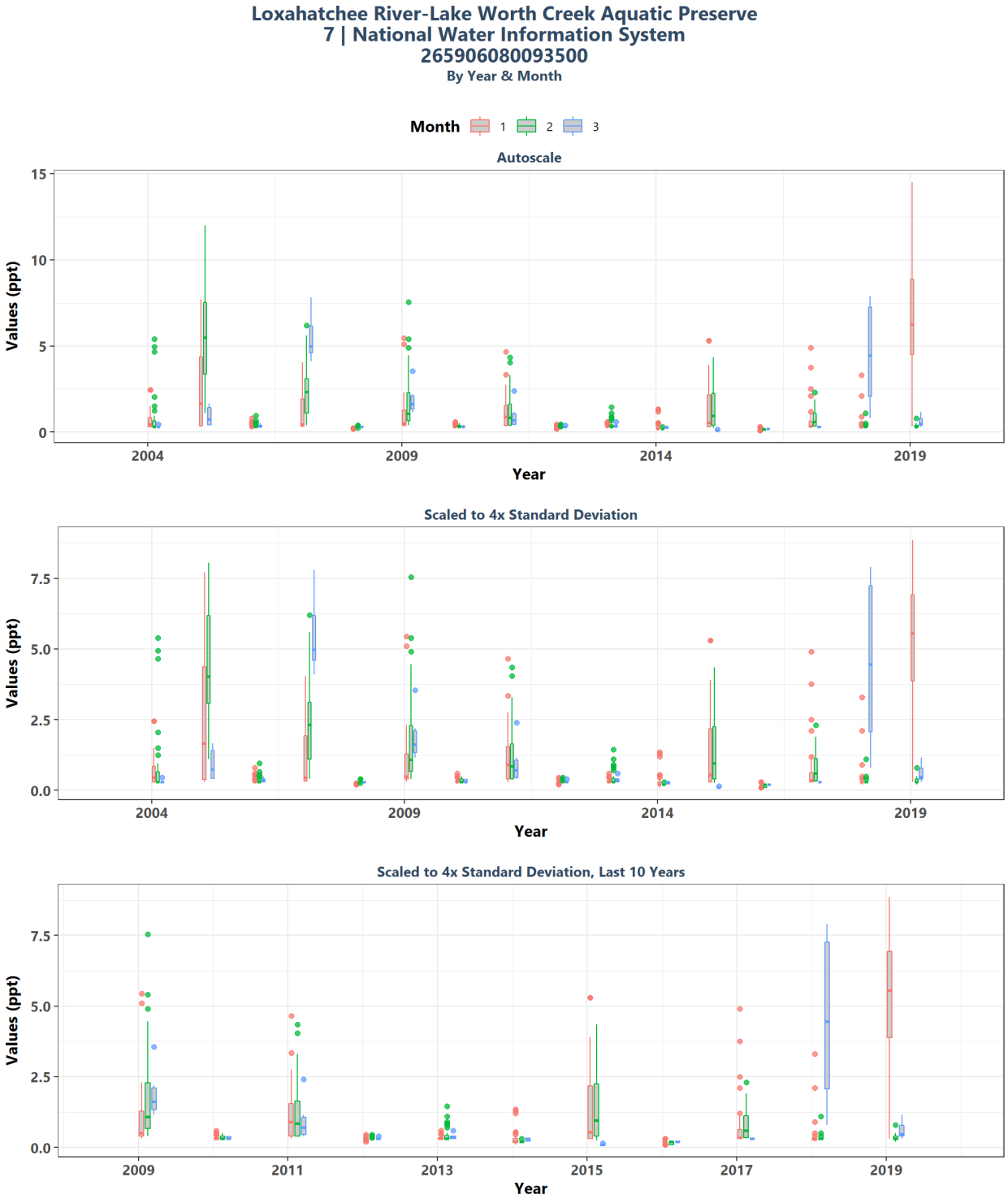

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmsswq
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

Guana Tolomato Matanzas National Estuarine Research Reserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmsswq
By Month

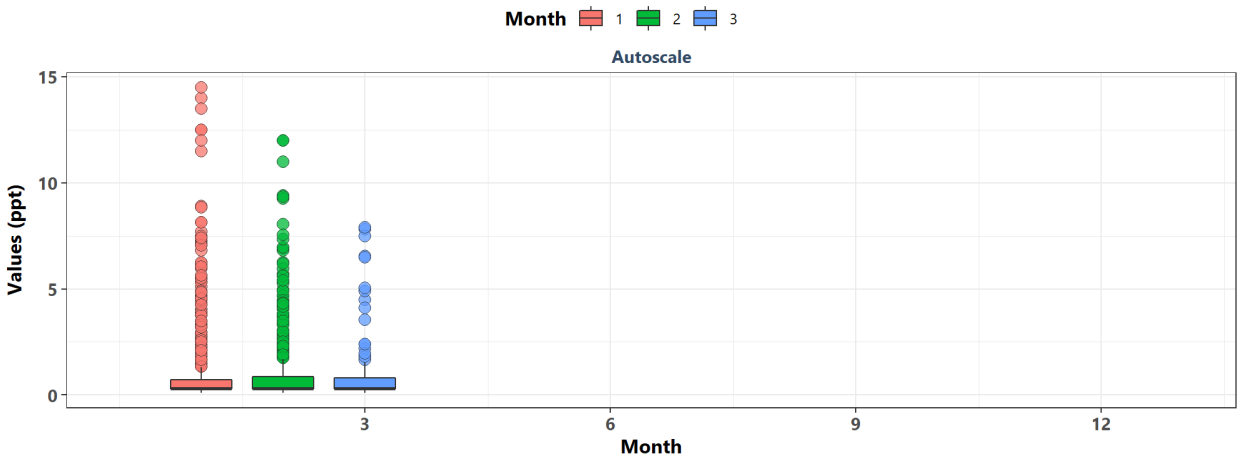


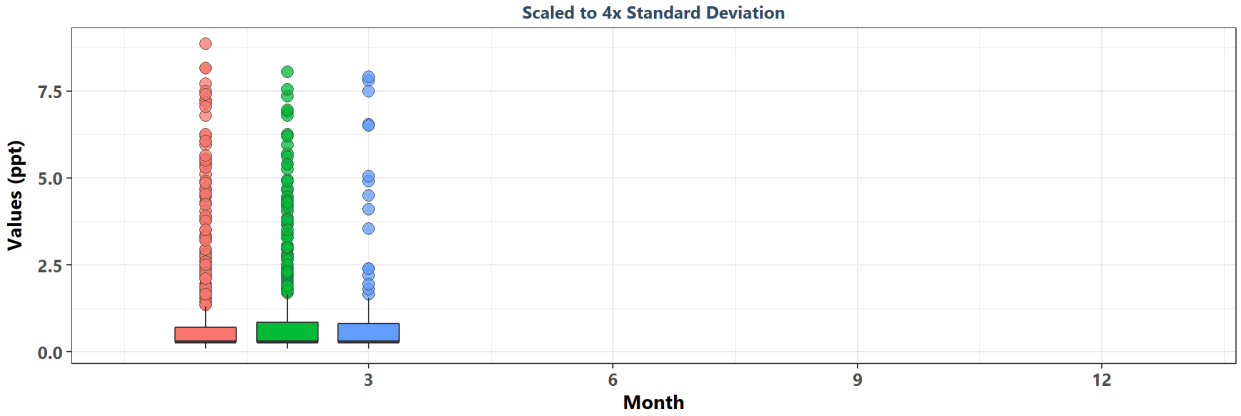
Loxahatchee River-Lake Worth Creek Aquatic Preserve
7 | National Water Information System
265906080093500
By Year

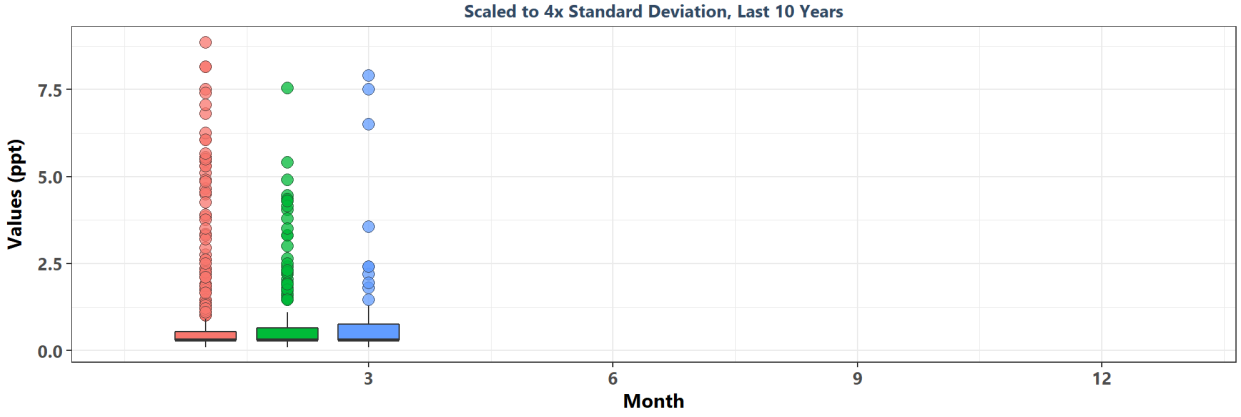



Loxahatchee River-Lake Worth Creek Aquatic Preserve
7 | National Water Information System
265906080093500
By Year & Month
Month 1 2 3
Autoscale
Values (ppt)
Year
Scaled to 4x Standard Deviation
Values (ppt)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (ppt)
Year

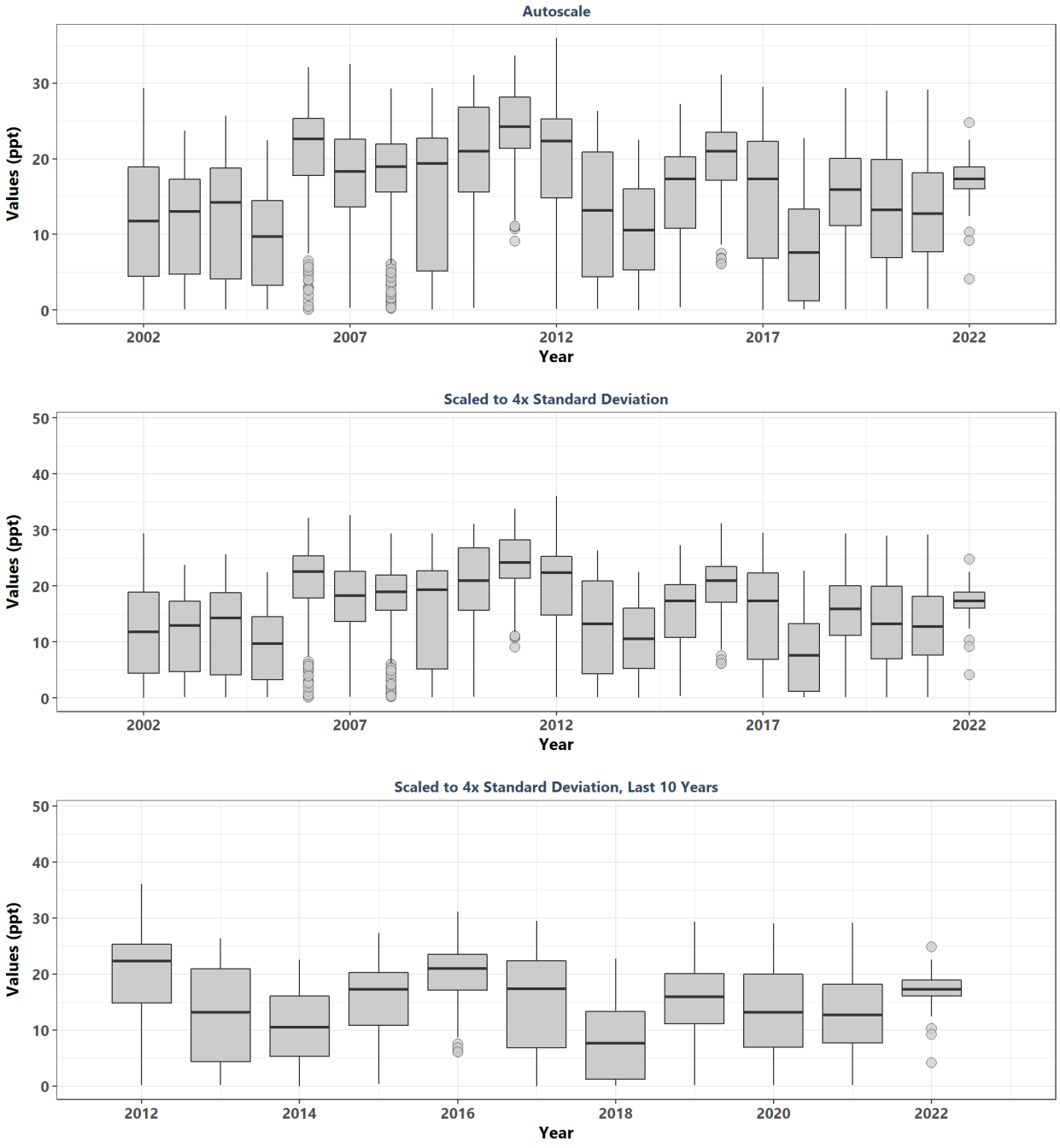
Loxahatchee River-Lake Worth Creek Aquatic Preserve
7 | National Water Information System
265906080093500
By Month



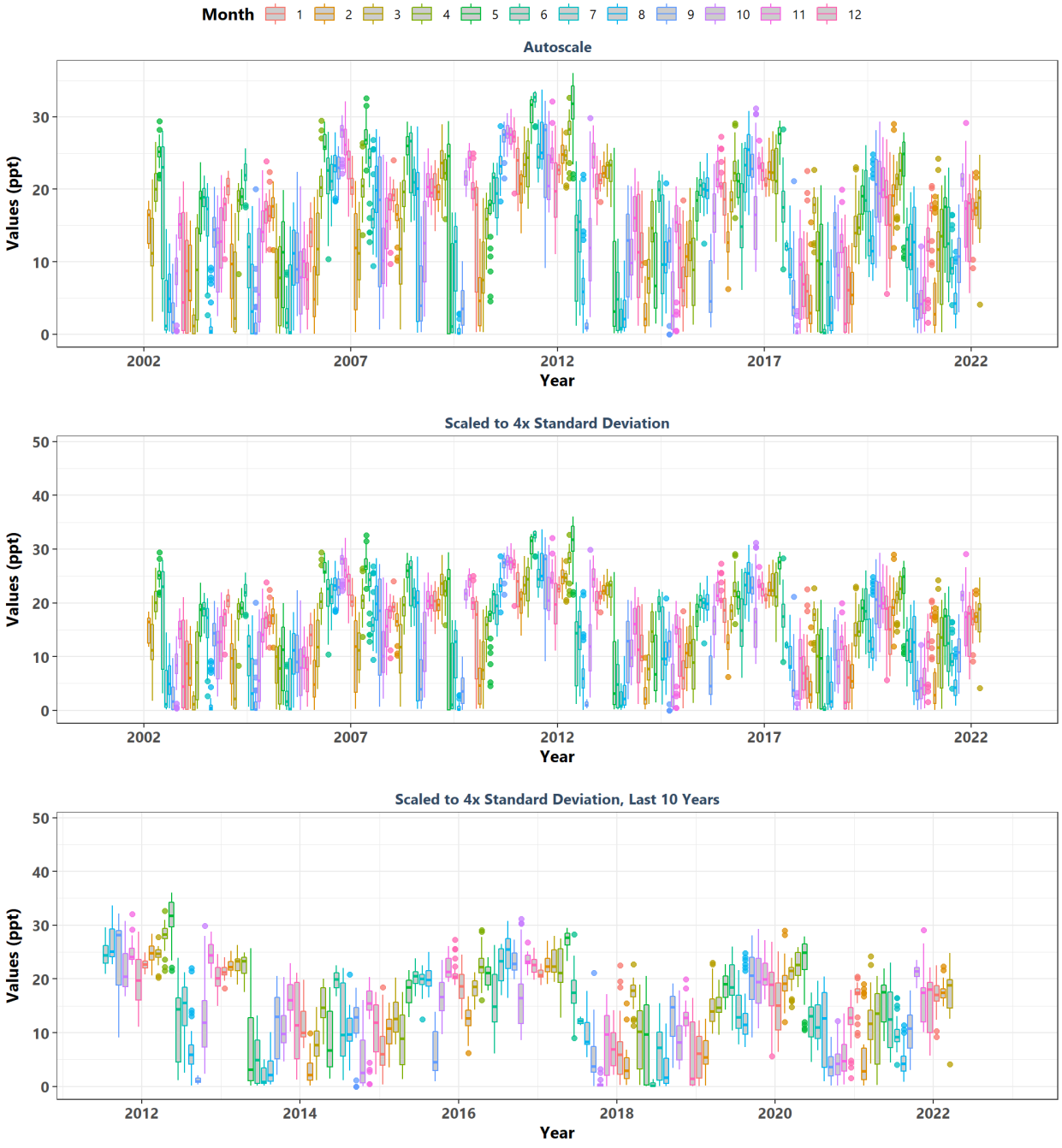


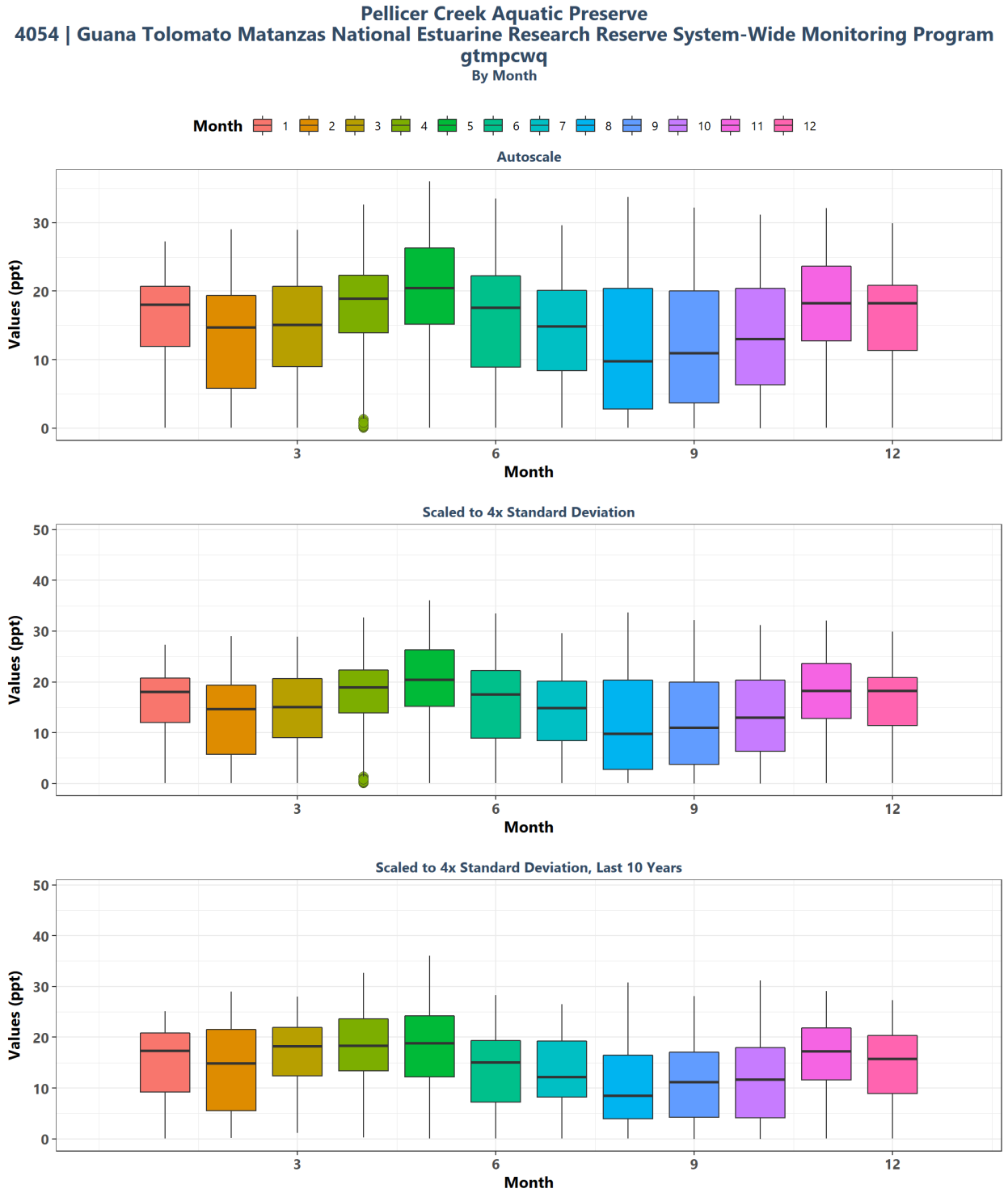


Pellicer Creek Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq
By Year



Pellicer Creek Aquatic Preserve
4054 | Guana Tolomato Matanzas National Estuarine Research Reserve System-Wide Monitoring Program
gtmpcwq
By Year & Month






```
rm(list = setdiff(ls(), c("param_name", "all_regions", "file_list", "KT.Stats_all", "MA_All", "APP_Plots
```