SEACAR Continuous Water Quality Analysis: SW Region for Salinity

Last compiled on 19 May, 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(dplyr)
library(lubridate)
library(ggplot2)
library(ggpubr)
library(scales)
library(EnvStats)
library(tidyr)
options(scipen = 999)
```

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$ResultValue>=-5,]
} else{
   data <- data[data$ResultValue>=0,]
}



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

```
## `summarise()` has grouped output by 'ManagedAreaName', 'ProgramID', 'ProgramName', 'ProgramLocationI
## `.groups` argument.
```

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

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 <- data %>%
    group_by(ManagedAreaName, ProgramID, ProgramName, ProgramLocationID) %>%
    mutate(MonitoringID = cur_group_id())
# data <- data %>%
#    mutate(MonitoringID = group_indices(., ManagedAreaName, ProgramID,
#                                        ProgramName, ProgramLocationID))
Mon_Years <- data[data$Include == TRUE, ] %>%
   group_by(MonitoringID) %>%
   summarize(AreaID = unique(AreaID),
             ManagedAreaName = unique(ManagedAreaName),
             ProgramID = unique(ProgramID),
             ProgramName = unique(ProgramName),
             ProgramLocationID = unique(ProgramLocationID),
             ParameterName = parameter,
             RelativeDepth = unique(RelativeDepth),
             Y = length(unique(Year)))
```

```
Mon_Years <- as.data.table(Mon_Years[
   order(Mon_Years$MonitoringID), ])
Mon_Years$Enough_Time <- ifelse(Mon_Years$Y < 5, FALSE, TRUE)
data <- merge.data.frame(data, Mon_Years[,c("MonitoringID", "Enough_Time")],
                         by = "MonitoringID")
# data$Exclude_MonitoringID <- is.element(data$MonitoringID,
#                                         Mon_Years$MonitoringID[
#                                            Mon_Years$Enough_Time == FALSE])
data$Use_In_Analysis <- ifelse(data$Include == TRUE &
                                  data$Enough_Time == TRUE,
                               TRUE, FALSE)
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$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            Year, Month) %>%
   summarize(ParameterName = parameter,
             RelativeDepth = unique(RelativeDepth),
             EarliestSampleDate = min(SampleDate),
             LastSampleDate = max(SampleDate), N = length(ResultValue),
             Min = min(ResultValue), Max = max(ResultValue),
             Median = median(ResultValue), Mean = mean(ResultValue),
             StandardDeviation = sd(ResultValue))
Mon_YM_Stats <- as.data.table(Mon_YM_Stats[order(Mon_YM_Stats$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$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            Year) %>%
   summarize(ParameterName = parameter,
             RelativeDepth = unique(RelativeDepth),
             EarliestSampleDate = min(SampleDate),
             LastSampleDate = max(SampleDate), N = length(ResultValue),
             Min = min(ResultValue), Max = max(ResultValue),
             Median = median(ResultValue), Mean = mean(ResultValue),
             StandardDeviation = sd(ResultValue))
Mon_Y_Stats <- as.data.table(Mon_Y_Stats[order(Mon_Y_Stats$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$Use_In_Analysis == TRUE, ] %>%
   group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
            Month) %>%
   summarize(ParameterName = parameter,
             RelativeDepth = unique(RelativeDepth),
             EarliestSampleDate = min(SampleDate),
             LastSampleDate = max(SampleDate), N = length(ResultValue),
             Min = min(ResultValue), Max = max(ResultValue),
             Median = median(ResultValue), Mean = mean(ResultValue),
             StandardDeviation = sd(ResultValue))
Mon_M_Stats <- as.data.table(Mon_M_Stats[order(Mon_M_Stats$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) {
   tau <- NULL
   tryCatch({
      ken <-
         kendallSeasonalTrendTest(
            y = data$ResultValue,
            season = data$Month,
            year = data$Year,
            independent.obs = independent
         )
      tau <- ken$estimate[1]
      p <- ken$p.value[2]
      slope <- ken$estimate[2]
      intercept <- ken$estimate[3]
      trend <- trend_calculator(slope, stats.median, p)
   }, warning = function(w) {
      print(w)
   }, error = function(e) {
      print(e)
   }, finally = {
      if (!exists("tau")) {
         tau <- NULL
      }
      if (!exists("p")) {
         p <- NULL
      }
      if (!exists("slope")) {
         slope <- NULL
      }
      if (!exists("intercept")) {
         intercept <- NULL
```

```
      }
      if (!exists("trend")) {
         trend <- NULL
      }
   })
   KT <-c(unique(data$MonitoringID),
          independent,
          stats.median,
          nrow(data),
          stats.minYear,
          stats.maxYear,
          tau,
          p,
          slope,
          intercept,
          trend)
   return(KT)
}
runStats <- function(data) {
   data$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$Year, na.rm = TRUE)
   stats.maxYear <- max(data$Year, na.rm = TRUE)
   # Calculate Kendall Tau and Slope stats, then update appropriate columns and table
   KT <- tauSeasonal(data, TRUE, stats.median,
                     stats.minYear, stats.maxYear)
   if (is.null(KT[11])) {
      KT <- tauSeasonal(data, FALSE, stats.median,
                        stats.minYear, stats.maxYear)
   }
   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$ManagedAreaName
c_names <- c("MonitoringID", "Independent", "Median", "N", "EarliestYear",
             "LatestYear", "tau", "p", "SennSlope", "SennIntercept", "Trend")
if(n==0){
   c_names <- c("AreaID", "ManagedAreaName", "ProgramID", "ProgramName",
                "ProgramLocationID", "ParameterName", "RelativeDepth",
                "Independent", "Median", "N", "EarliestYear", "LatestYear",
                "tau", "p", "SennSlope", "SennIntercept", "Trend")
   KT.Stats <- data.frame(matrix(ncol=17, nrow=0))
   colnames(KT.Stats) <- c_names
   fwrite(KT.Stats, paste0(out_dir,"/", param_name, "_", region,
                           "_KendallTau_Stats.txt"), sep = "|")
} else{
   for (i in 1:n) {
      values <- data[data$Use_In_Analysis == TRUE &
                        data$MonitoringID == Mon_IDs[i], ]
      if (nrow(values) > 0) {
         KT.Stats <- runStats(values)
      }
   }
   KT.Stats <- as.data.frame(KT.Stats)
   if(dim(KT.Stats)[2]==1){
      KT.Stats <- as.data.frame(t(KT.Stats))
   }

   c_names <- c("MonitoringID", "Independent", "Median", "N", "EarliestYear",
                "LatestYear", "tau", "p", "SennSlope", "SennIntercept", "Trend")
   colnames(KT.Stats) <- c_names
   rownames(KT.Stats) <- seq(1:nrow(KT.Stats))
   KT.Stats$Independent <- as.logical(KT.Stats$Independent)
   KT.Stats$Median <- as.numeric(KT.Stats$Median)
   KT.Stats$N <- as.integer(KT.Stats$N)
   KT.Stats$EarliestYear <- as.integer(KT.Stats$EarliestYear)
   KT.Stats$LatestYear <- as.integer(KT.Stats$LatestYear)
   KT.Stats$tau <- round(as.numeric(KT.Stats$tau), digits=4)
   KT.Stats$p <- round(as.numeric(KT.Stats$p), 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_Years[,-c("Y", "Enough_Time")],
                                KT.Stats, by = "MonitoringID")
   KT.Stats$MonitoringID <- NULL
   fwrite(KT.Stats, paste0(out_dir,"/", param_name, "_", region,
```

```
                              "_KendallTau_Stats.txt"), sep = "|")
   KT.Stats$MonitoringID <- Mon_IDs
}
```

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.

```
   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(outlier.size = 0.5) +
      labs(subtitle = "Autoscale", x = "Year",
           y = paste0("Values (", unit, ")")) +
      theme_bw() +
      theme(axis.text.x = element_text(face = "bold"),
            axis.text.y = element_text(face = "bold"))

   p2 <- ggplot(data = data[data$Include == TRUE, ],
                aes(x = Year, y = ResultValue, group = Year)) +
      geom_boxplot(outlier.size = 0.5) +
      labs(subtitle = "Scaled to 4x Standard Deviation", x = "Year",
           y = paste0("Values (", unit, ")")) +
      ylim(0, y_scale) +
      theme_bw() + theme(axis.text.x = element_text(face = "bold"),
                         axis.text.y = element_text(face = "bold"))

   p3 <- ggplot(data = data[data$Include == TRUE, ],
                aes(x = as.integer(Year), y = ResultValue, group = Year)) +
      geom_boxplot(outlier.size = 0.5) +
      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)) +
    theme_bw() +
    theme(axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

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

p0 <- ggplot() + labs(title = "Summary Box Plots for Entire Data",
                      subtitle = "By Year") + theme_bw() +
    theme(plot.title = element_text(face="bold", hjust=0.5),
          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(outlier.size = 0.5) +
    labs(subtitle = "Autoscale", x = "Year",
         y = paste0("Values (", unit, ")"), color="Month") +
    theme_bw() +
    theme(legend.position = "top", legend.box = "horizontal",
          axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold")) +
    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(outlier.size = 0.5) +
    labs(subtitle = "Scaled to 5x Standard Deviation",
         x = "Year", y = paste0("Values (", unit, ")")) +
    ylim(0, y_scale) +
    theme_bw() +
    theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
          axis.text.y = element_text(face = "bold"))

p3 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = YearMonthDec, y = ResultValue,
                 group = YearMonth, color = as.factor(Month))) +
    geom_boxplot(outlier.size = 0.5) +
    labs(subtitle = "Scaled to 5x Standard Deviation, Last 10 Years",
         x = "Year", y = paste0("Values (", unit, ")")) +
    ylim(0, y_scale) +
    scale_x_continuous(limits = c(max(data$Year) - 10.5, max(data$Year)+0.5),
                       breaks = seq(max(data$Year) - 10, max(data$Year), 2)) +
    theme_bw() +
```

```
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))
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") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust=0.5),
         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.

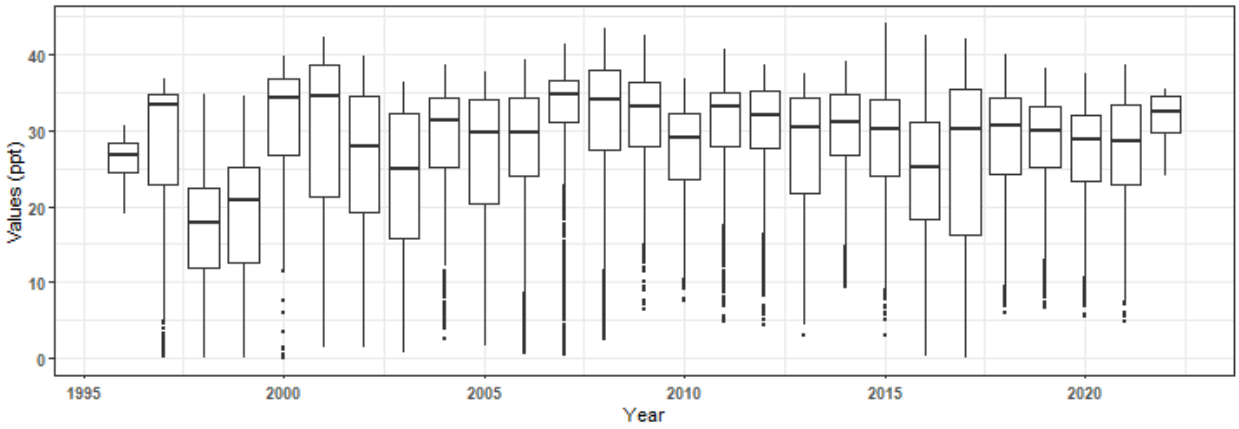
```
p1 <- ggplot(data = data[data$Include == TRUE, ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   labs(subtitle = "Autoscale", x = "Month",
        y = paste0("Values (", unit, ")"), fill="Month") +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "top", legend.box = "horizontal",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold")) +
   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(outlier.size = 0.5) +
   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)) +
   theme_bw() +
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

p3 <- ggplot(data = data[data$Include == TRUE &
                           data$Year >= max(data$Year) - 10, ],
             aes(x = Month, y = ResultValue,
                 group = Month, fill = as.factor(Month))) +
   geom_boxplot(outlier.size = 0.5) +
   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)) +
   theme_bw() +
   theme(legend.position = "none", axis.text.x = element_text(face = "bold"),
```

```
          axis.text.y = element_text(face = "bold"))
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") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust=0.5),
         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))
```

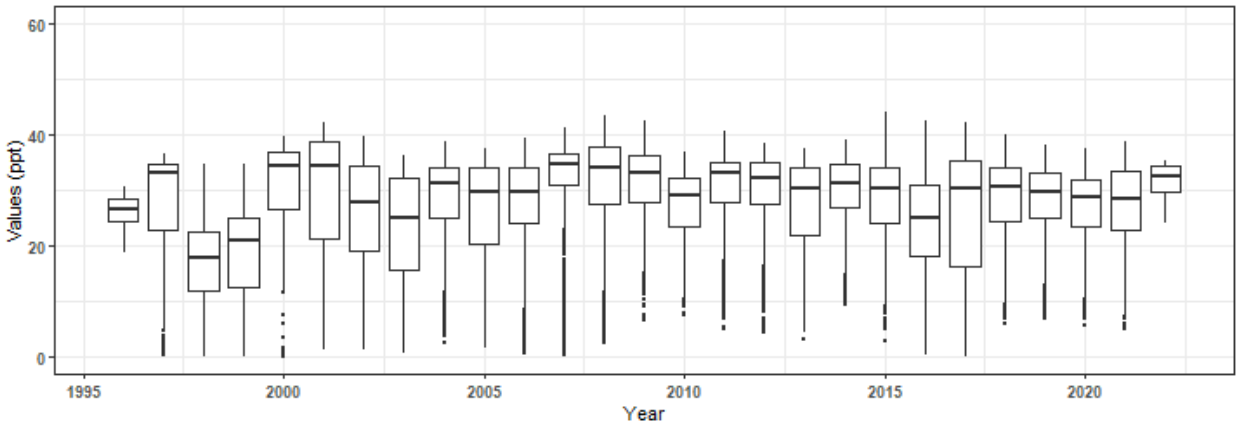
Summary Box Plots for Entire Data

By Year

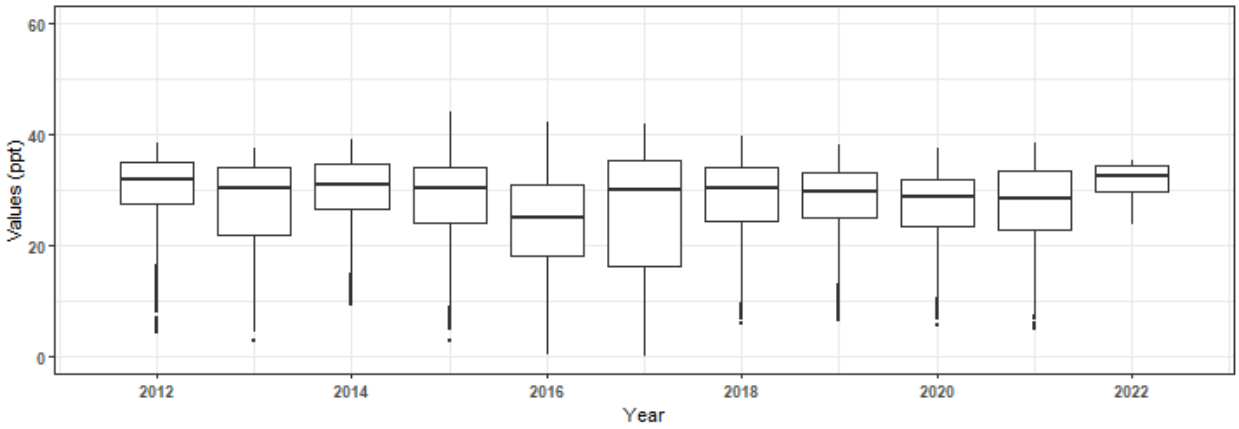
Autoscale



Scaled to 4x Standard Deviation

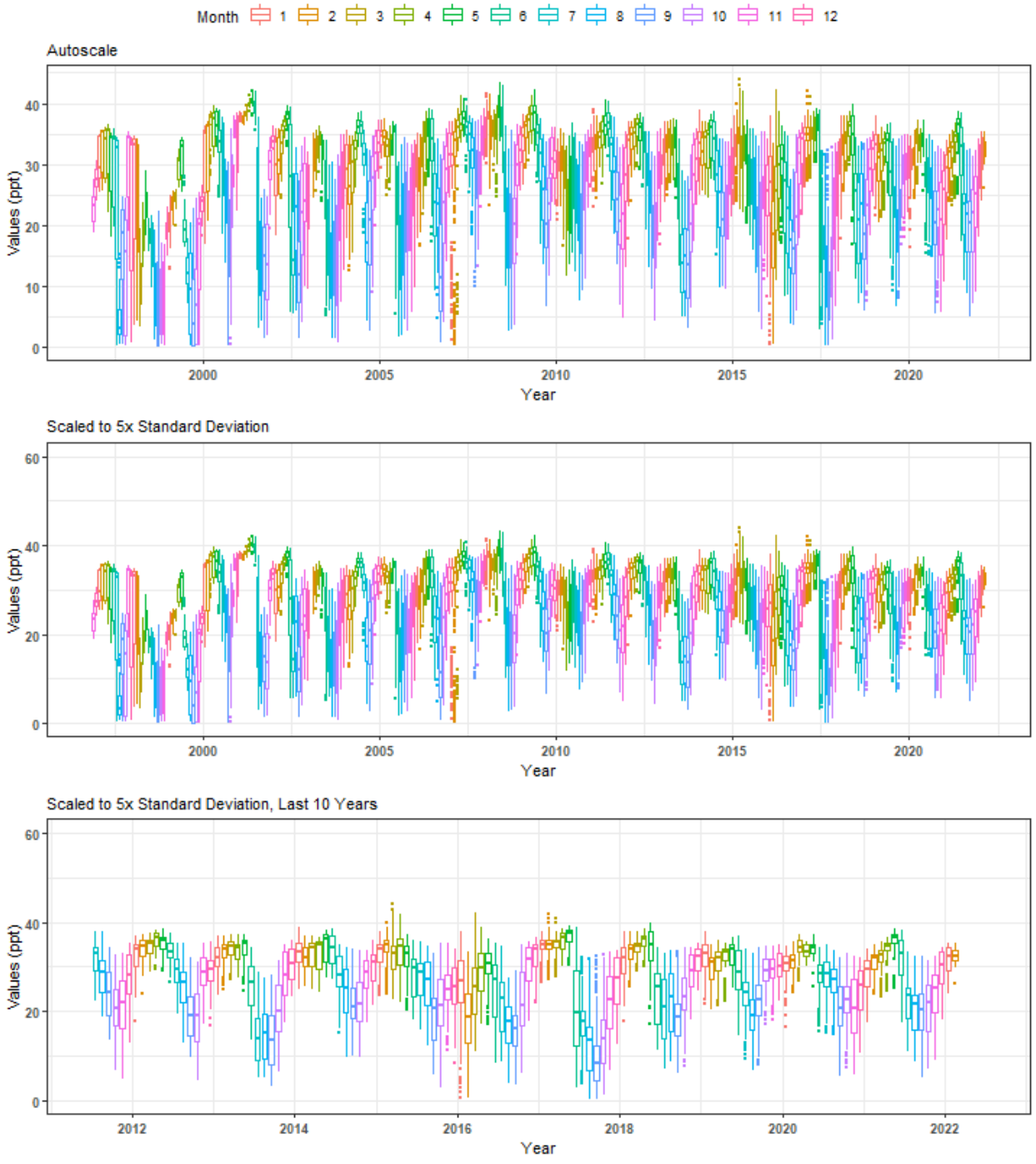


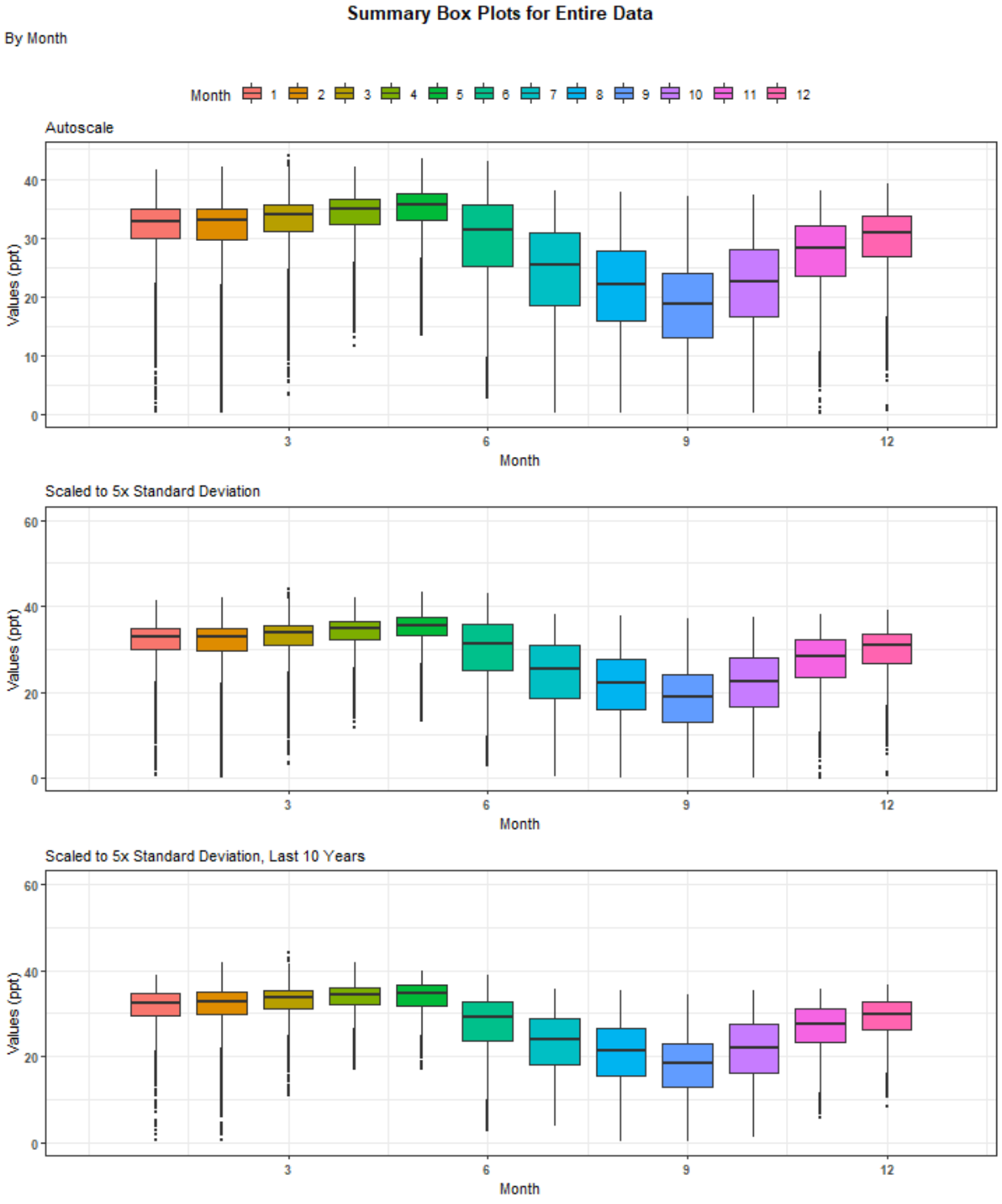
Scaled to 4x Standard Deviation, Last 10 Years



Summary Box Plots for Entire Data

By Year & Month



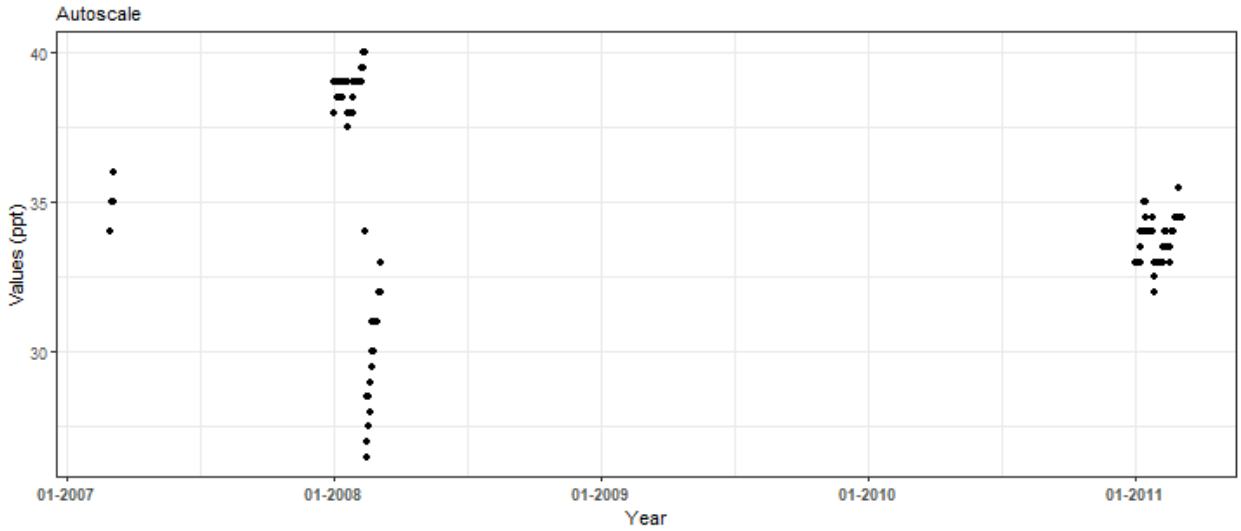


Appendix II: Excluded Monitoring Locations

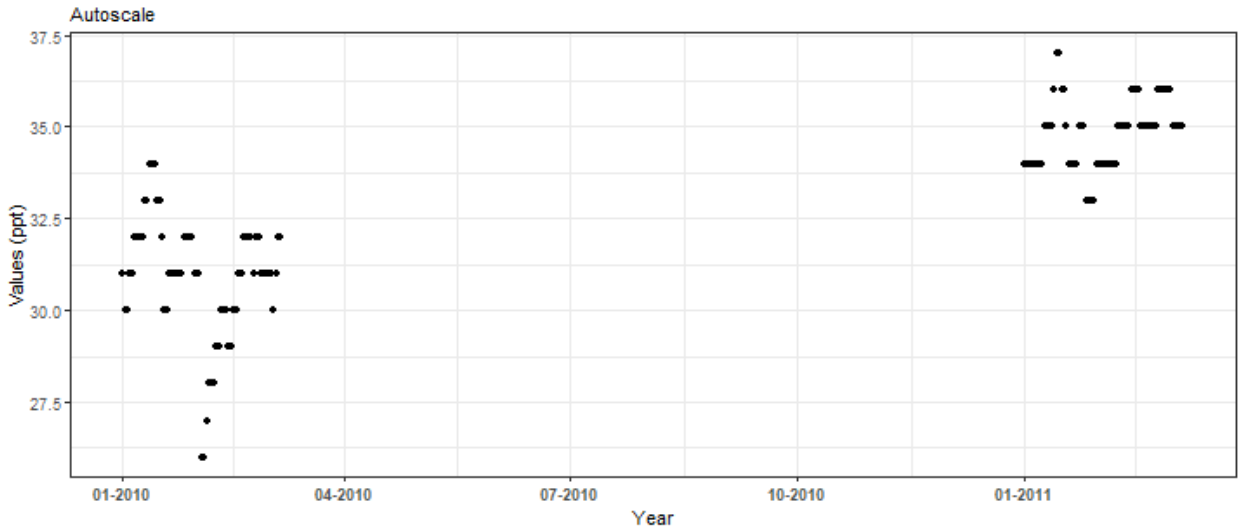
Scatter plots of data values are created for monitoring locationss that have fewer than 5 separate years of data entries.

```
Mon_Exclude <- Mon_Years[Mon_Years$Enough_Time==FALSE,]
Mon_Exclude <- Mon_Exclude[order(Mon_Exclude$MonitoringID),]
z=length(Mon_Exclude$MonitoringID)

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$MonitoringID[i]])
      Mon_name <- paste(unique(data$ProgramID[
         data$MonitoringID==Mon_Exclude$MonitoringID[i]]),
         unique(data$ProgramName[
            data$MonitoringID==Mon_Exclude$MonitoringID[i]]),
         unique(data$ProgramLocationID[
            data$MonitoringID==Mon_Exclude$MonitoringID[i]]),
         sep = " | ")

      p1<-ggplot(data=data[data$MonitoringID==Mon_Exclude$MonitoringID[i]&
                              data$Include == TRUE, ],
                 aes(x = SampleDate, y = ResultValue)) +
         geom_point() +
         labs(title=
                 paste0("Scatter Plot of Excluded Monitoring Location ",
                        MA_name, "\n", Mon_name, "\n(", Mon_Exclude$Y[i],
                        " Unique Years)"),
              subtitle="Autoscale", x = "Year",
              y = paste0("Values (", unit, ")")) +
         theme_bw() +
         theme(plot.title = element_text(face="bold", hjust=0.5),
               axis.text.x = element_text(face = "bold")) +
         scale_x_date(labels = date_format("%m-%Y"))
      print(p1)
   }
}
```

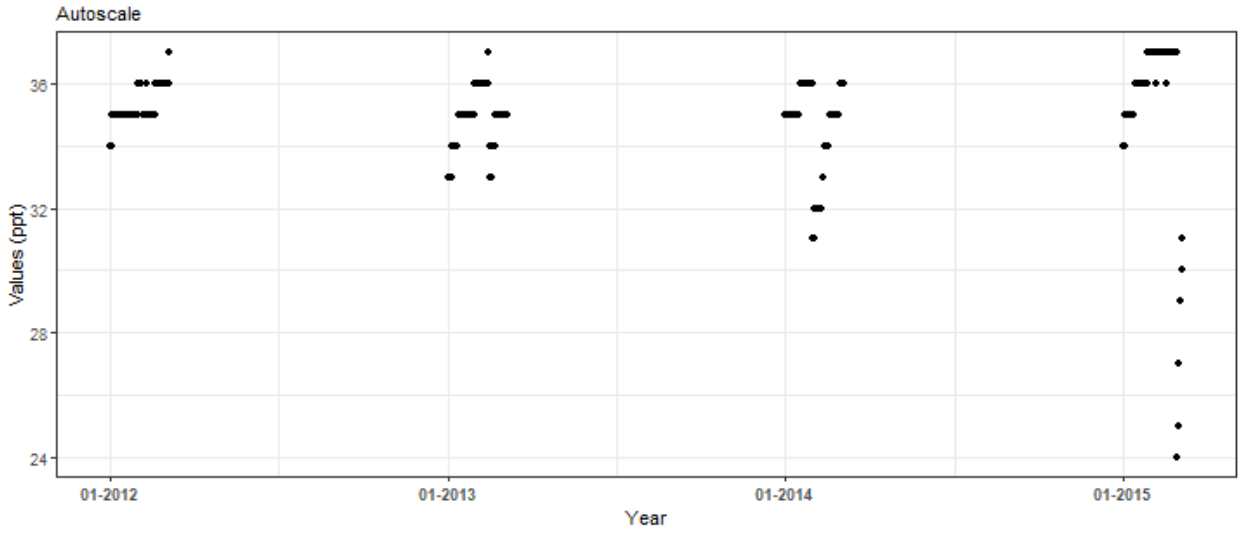
Scatter Plot of Excluded Monitoring Location Cape Romano-Ten Thousand Islands Aquatic Preserve 7 | National Water Information System | 255443081314700 (3 Unique Years)



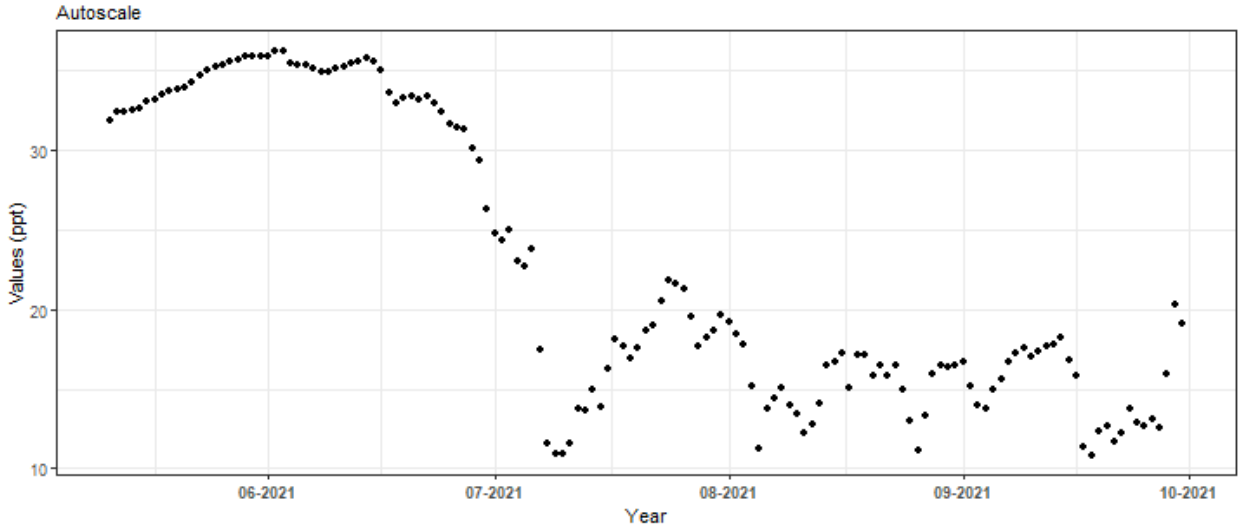
Scatter Plot of Excluded Monitoring Location Cape Romano-Ten Thousand Islands Aquatic Preserve 7 | National Water Information System | 255532081314300 (2 Unique Years)



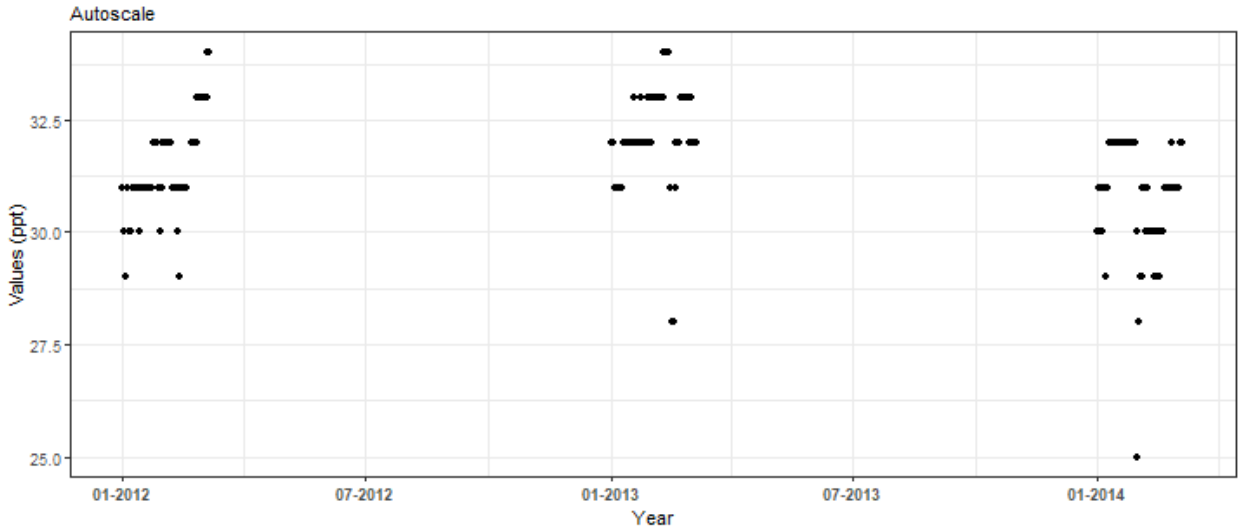
Scatter Plot of Excluded Monitoring Location Cape Romano-Ten Thousand Islands Aquatic Preserve 7 | National Water Information System | 255732081363700 (4 Unique Years)



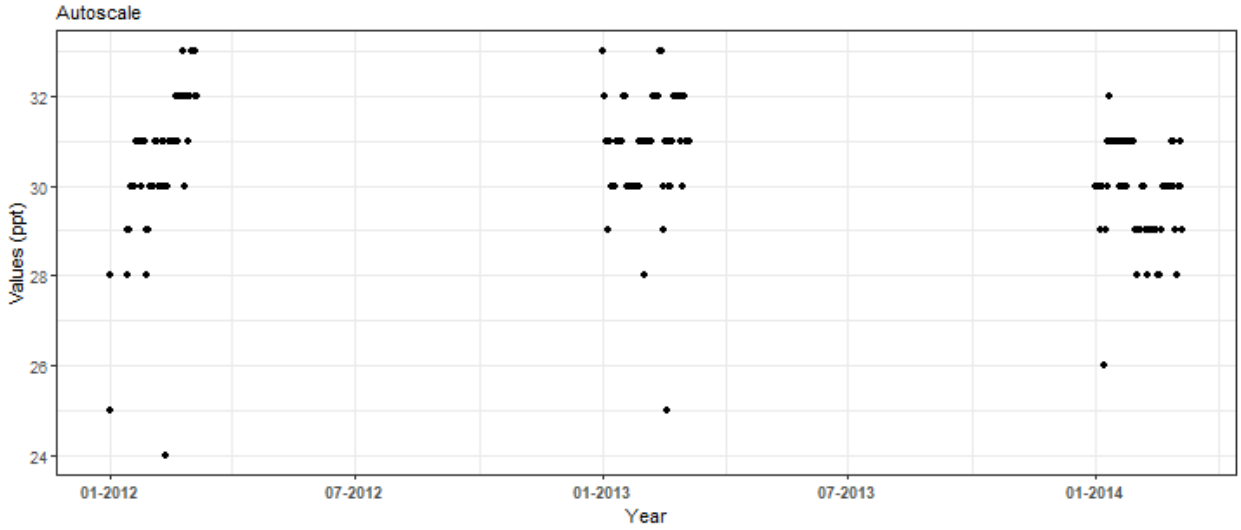
Scatter Plot of Excluded Monitoring Location Estero Bay Aquatic Preserve 474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB04 (1 Unique Years)



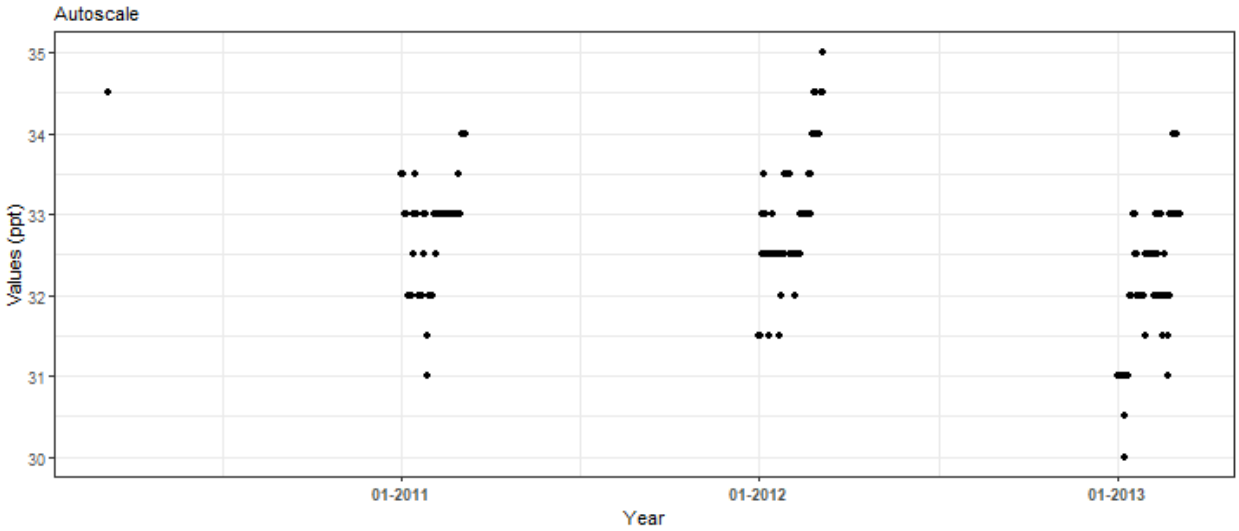
Scatter Plot of Excluded Monitoring Location Gasparilla Sound-Charlotte Harbor Aquatic Preserve 7 | National Water Information System | 02293252 (3 Unique Years)



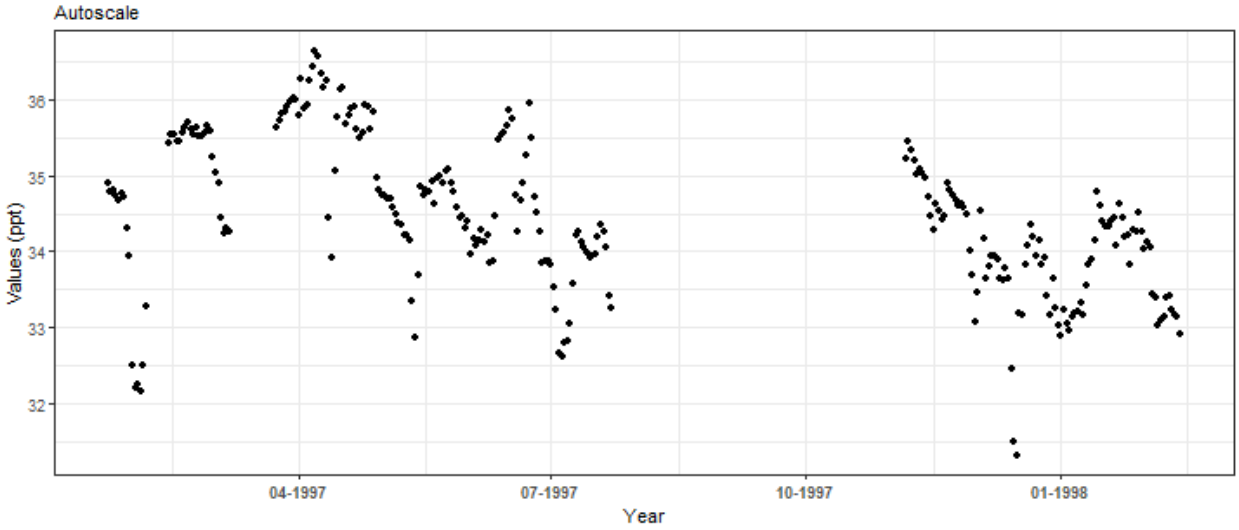
Scatter Plot of Excluded Monitoring Location Gasparilla Sound-Charlotte Harbor Aquatic Preserve 7 | National Water Information System | 02293254 (3 Unique Years)



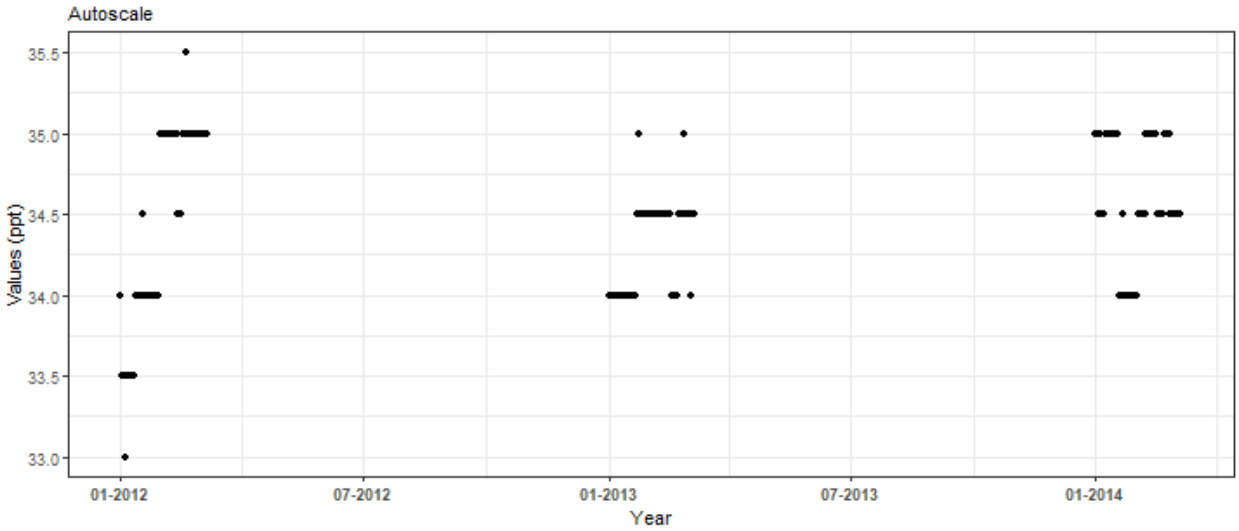
Scatter Plot of Excluded Monitoring Location Pine Island Sound Aquatic Preserve
7 | National Water Information System | 02293249
(4 Unique Years)



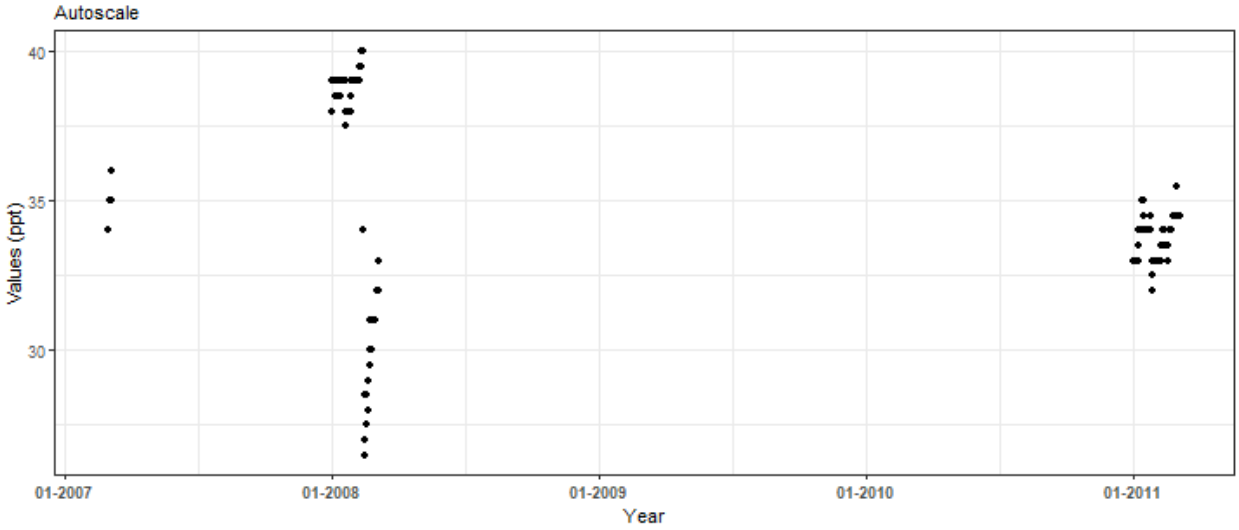
Scatter Plot of Excluded Monitoring Location Rookery Bay Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbbmwq
(2 Unique Years)



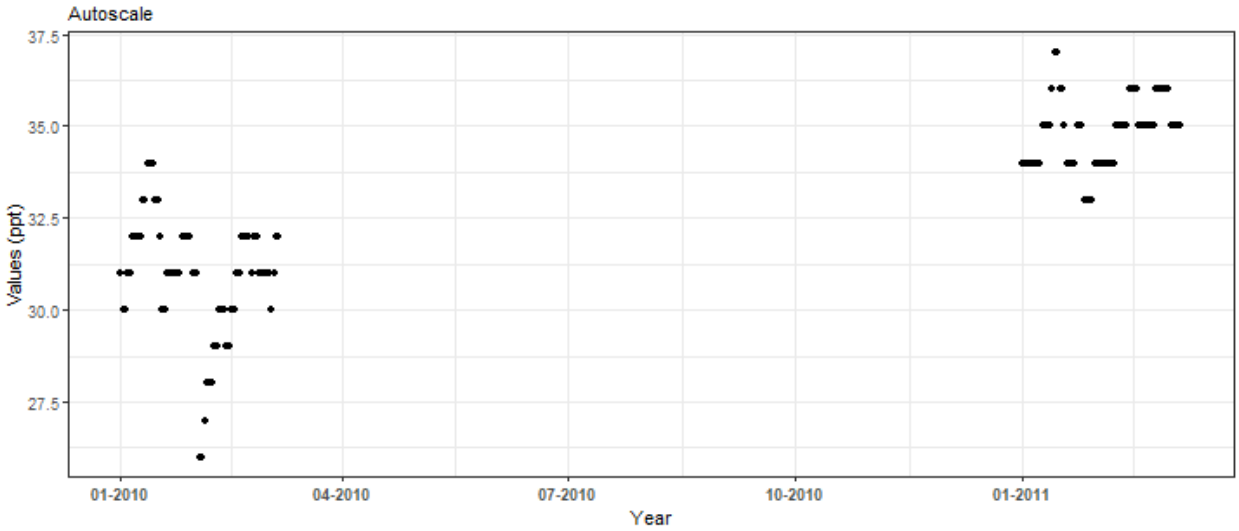
Scatter Plot of Excluded Monitoring Location Rookery Bay National Estuarine Research Reserve 7 | National Water Information System | 02291330 (3 Unique Years)



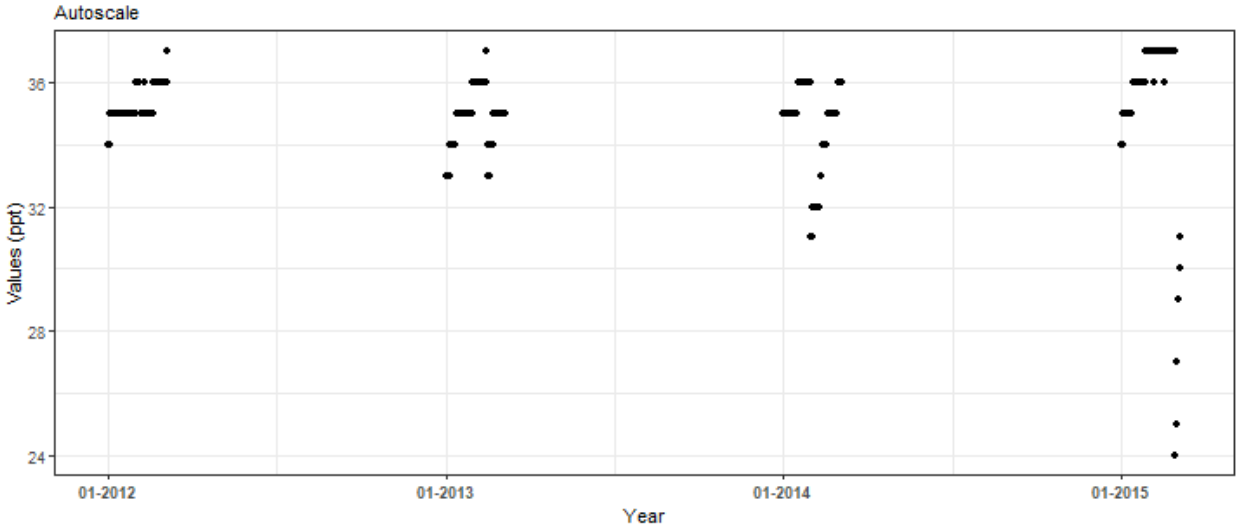
Scatter Plot of Excluded Monitoring Location Rookery Bay National Estuarine Research Reserve 7 | National Water Information System | 255443081314700 (3 Unique Years)



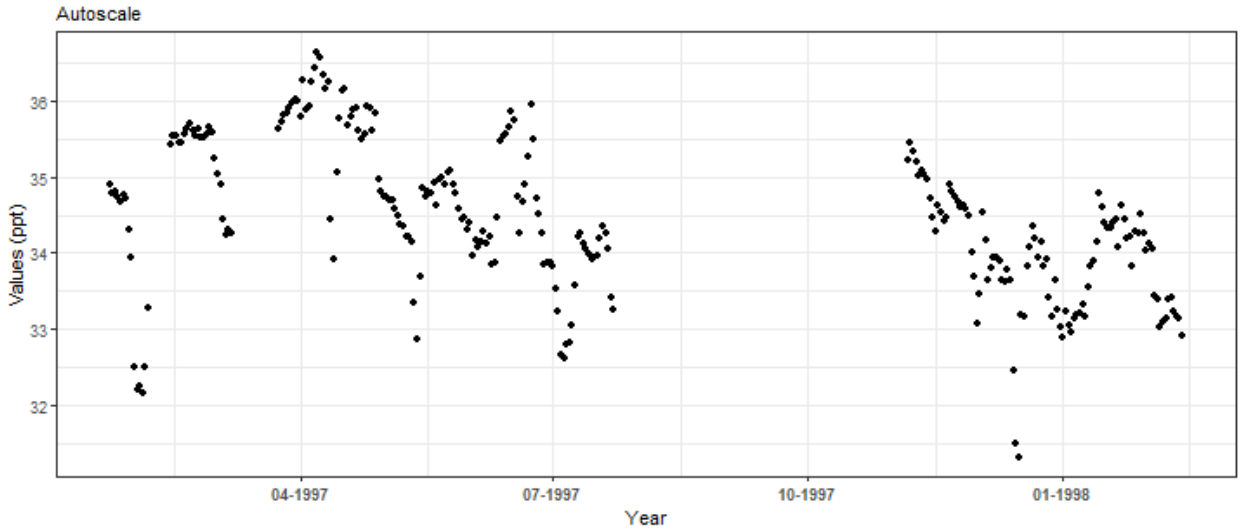
Scatter Plot of Excluded Monitoring Location Rookery Bay National Estuarine Research Reserve 7 | National Water Information System | 255532081314300 (2 Unique Years)



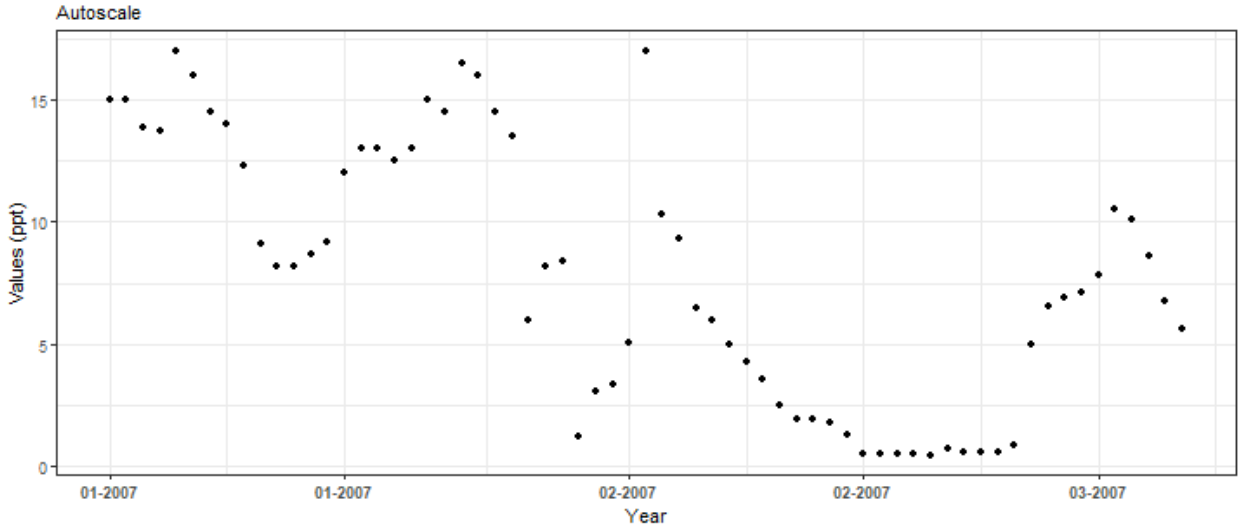
Scatter Plot of Excluded Monitoring Location Rookery Bay National Estuarine Research Reserve 7 | National Water Information System | 255732081363700 (4 Unique Years)

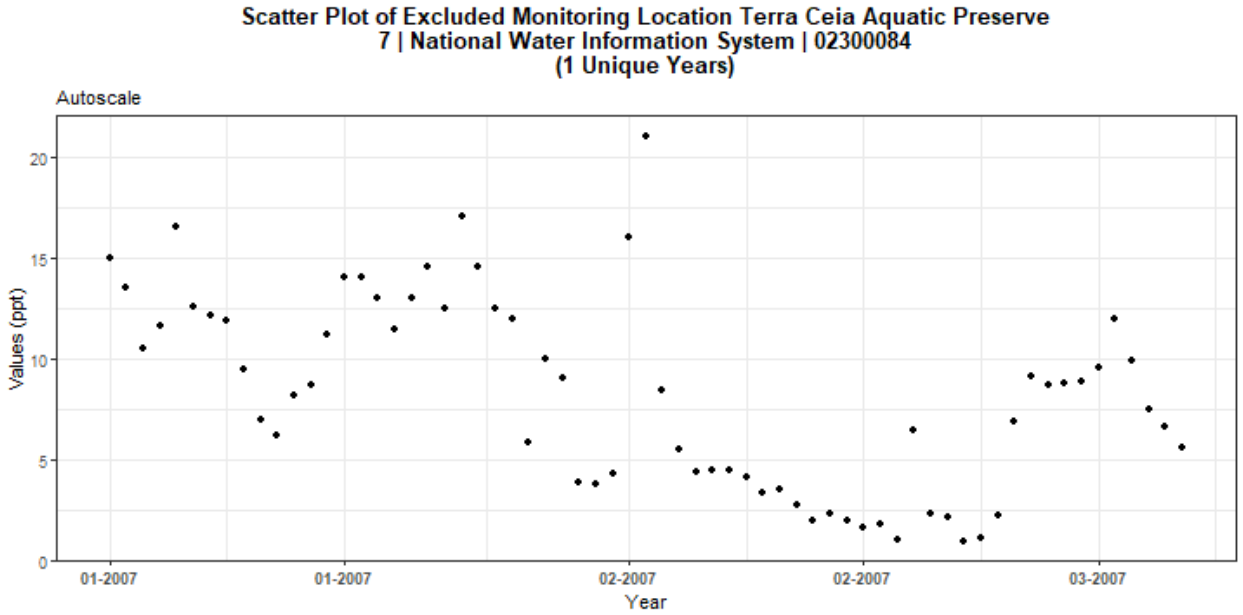


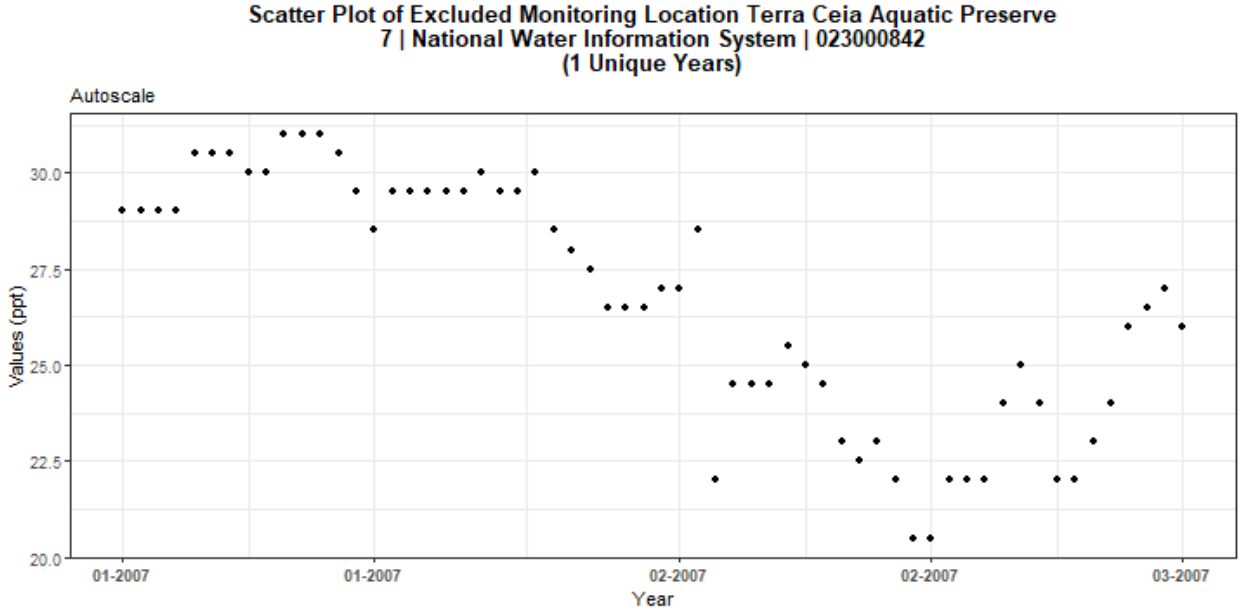
Scatter Plot of Excluded Monitoring Location Rookery Bay National Estuarine Research Reserve 354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbbmwq (2 Unique Years)

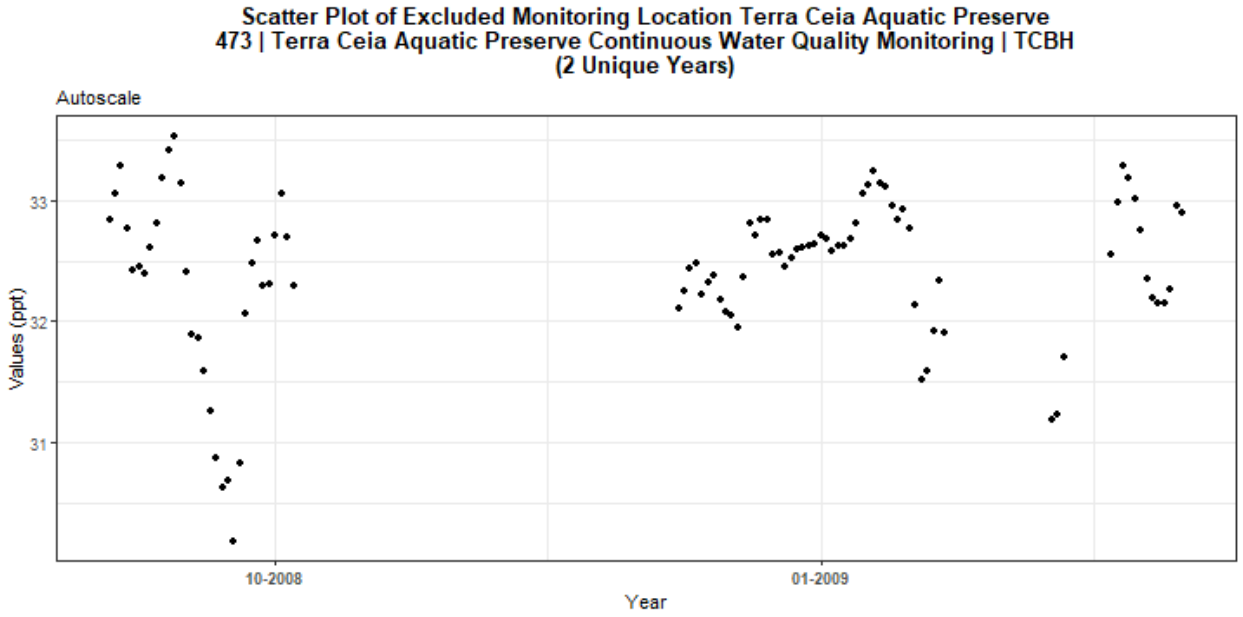


Scatter Plot of Excluded Monitoring Location Terra Ceia Aquatic Preserve 7 | National Water Information System | 023000825 (1 Unique Years)




Scatter Plot of Excluded Monitoring Location Terra Ceia Aquatic Preserve
7 | National Water Information System | 02300084
(1 Unique Years)
Autoscale
Values (ppt)
0
5
10
15
20
01-2007
01-2007
02-2007
02-2007
03-2007
Year


Scatter Plot of Excluded Monitoring Location Terra Ceia Aquatic Preserve
7 | National Water Information System | 023000842
(1 Unique Years)
Autoscale
Values (ppt)
20.0
22.5
25.0
27.5
30.0
01-2007
01-2007
02-2007
02-2007
03-2007
Year



Appendix III: Monitoring Location Trendlines

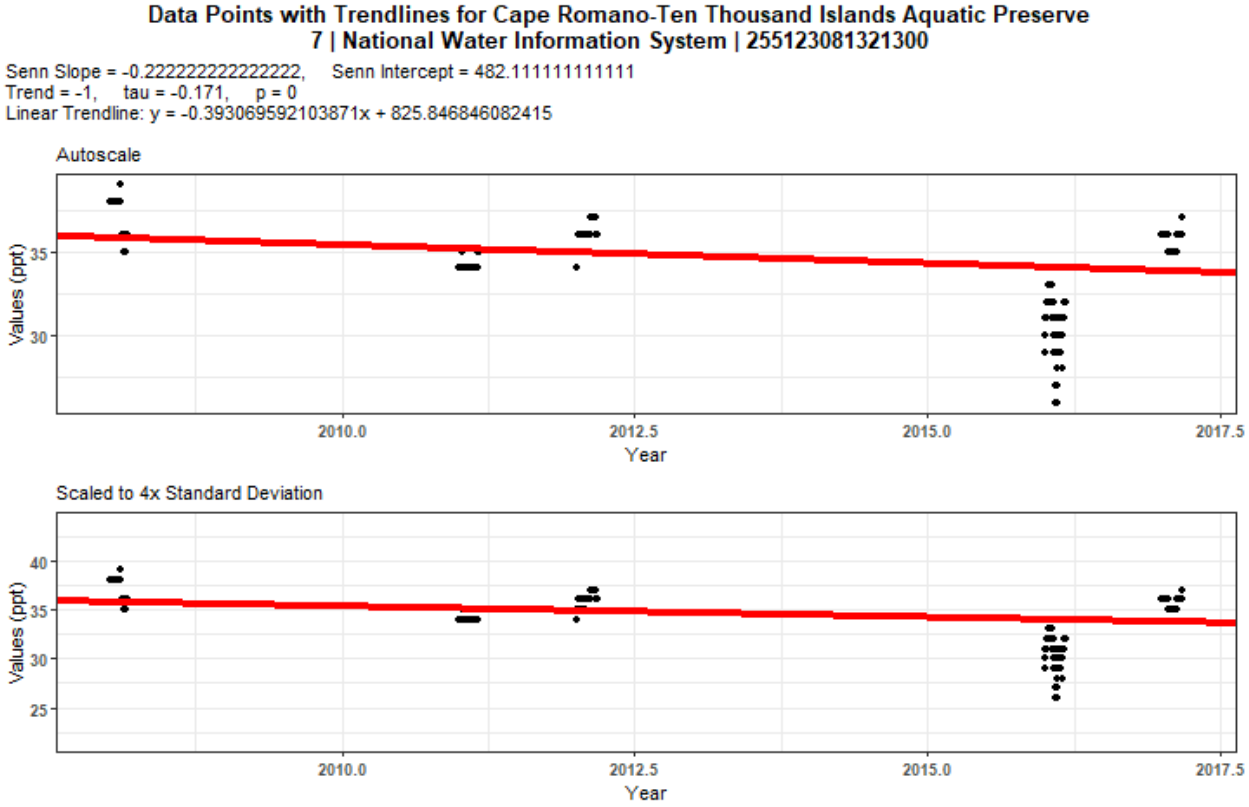
The plots created in this section are designed to show the general trend of the data. Data is taken and grouped by `MonitoringID`. The trendlines on the plots are created using the Senn slope and intercept from the seasonal Kendall Tau analysis. The scripts that create plots follow this format

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],]
      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$ResultValue <
                                       quantile(plot_data$ResultValue, 0.98)])
```

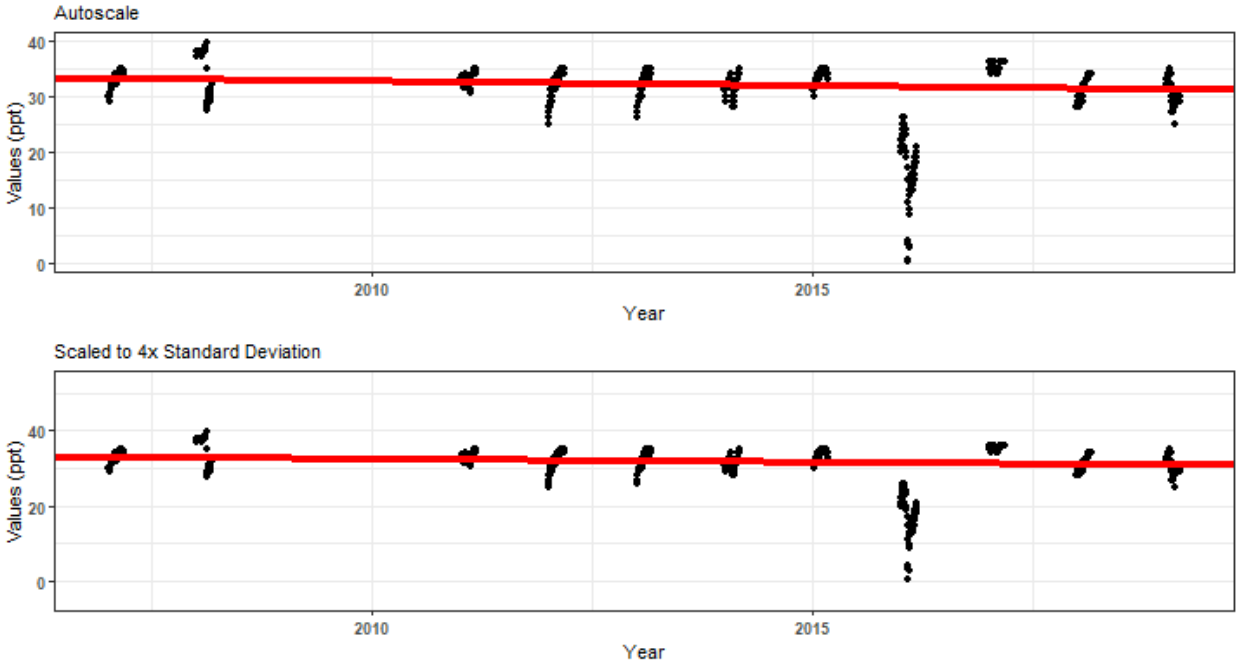
```
sd_RV <- sd(plot_data$ResultValue[plot_data$ResultValue <
                                  quantile(plot_data$ResultValue, 0.98)])
x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
y_scale <- mn_RV + 4 * sd_RV

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]]
p <- KT.Stats$p[KT.Stats$MonitoringID == Mon_IDs[i]]

model <- lm(ResultValue ~ DecDate,
            data = plot_data)
m_int <- coef(model)[[1]]
m_slope <- coef(model)[[2]]
MA_name <- KT.Stats$ManagedAreaName[KT.Stats$MonitoringID == Mon_IDs[i]]
Mon_name <- paste(KT.Stats$ProgramID[KT.Stats$MonitoringID == Mon_IDs[i]],
   KT.Stats$ProgramName[KT.Stats$MonitoringID == Mon_IDs[i]],
   KT.Stats$ProgramLocationID[KT.Stats$MonitoringID == Mon_IDs[i]],
   sep = " | ")

p1 <- ggplot(data = plot_data,
             aes(x = DecDate, y = ResultValue)) +
   geom_point(size = 1.5) +
   geom_abline(aes(slope=s_slope, intercept=s_int),
               color="red", size=1.5) +
   labs(subtitle = "Autoscale",
        x = "Year", y = paste0("Values (", unit, ")")) +
   theme_bw() +
   theme(axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face="bold"))

p2 <- ggplot(data = plot_data,
             aes(x = DecDate, y = ResultValue)) +
   geom_point(size = 1.5) +
   geom_abline(aes(slope=s_slope, intercept=s_int),
               color="red", size=1.5) +
   ylim(min_RV-0.1*y_scale, y_scale) +
   labs(subtitle = "Scaled to 4x Standard Deviation",
        x = "Year", y = paste0("Values (", unit, ")")) +
   theme_bw() +
   theme(axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face="bold"))
KTset <- ggarrange(p1, p2, ncol = 1, heights = c(1, 1))

p0 <- ggplot() + labs(title = paste0("Data Points with Trendlines for ",
                                     MA_name, "\n", Mon_name),
                      subtitle =paste0("Senn Slope = ", s_slope,
                                       ",     Senn Intercept = ", s_int,
                                       "\nTrend = ", trend,
                                       ",     tau = ", tau,
                                       ",     p = ", p,
                                       "\nLinear Trendline: ",
```

```
                                          "y = ", m_slope,"x + ",m_int)) +
        theme_bw() + theme(plot.title = element_text(face="bold", hjust=0.5),
                           panel.border = element_blank(),
                           panel.grid.major = element_blank(),
                           panel.grid.minor = element_blank(),
                           axis.line = element_blank())

      print(ggarrange(p0, KTset, ncol = 1, heights = c(0.20, 1)))
   }
}
```



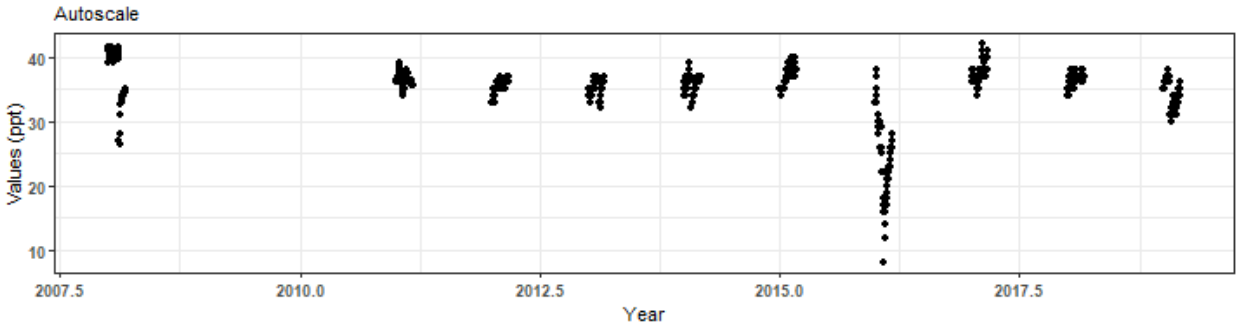
Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve 7 | National Water Information System | 255432081303900

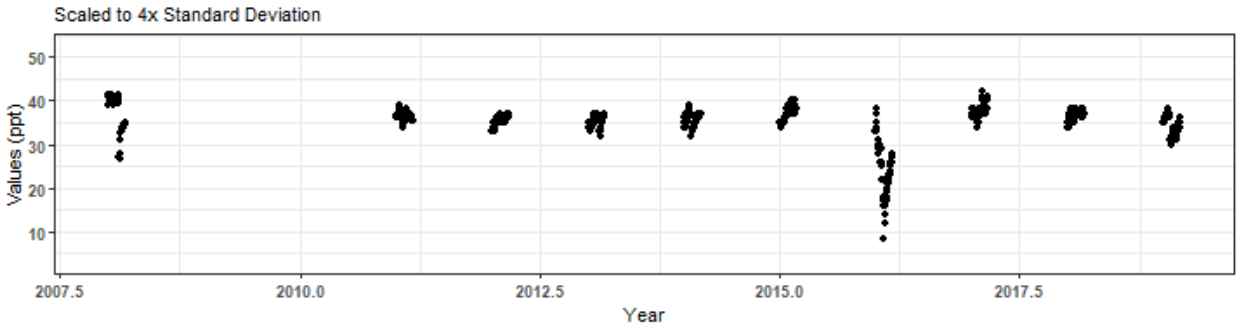
Senn Slope = -0.166666666666667, Senn Intercept = 367.666666666667
Trend = -1, tau = -0.1307, p = 0
Linear Trendline: y = -0.381647692934698x + 799.890251318757



Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve 7 | National Water Information System | 255534081324000

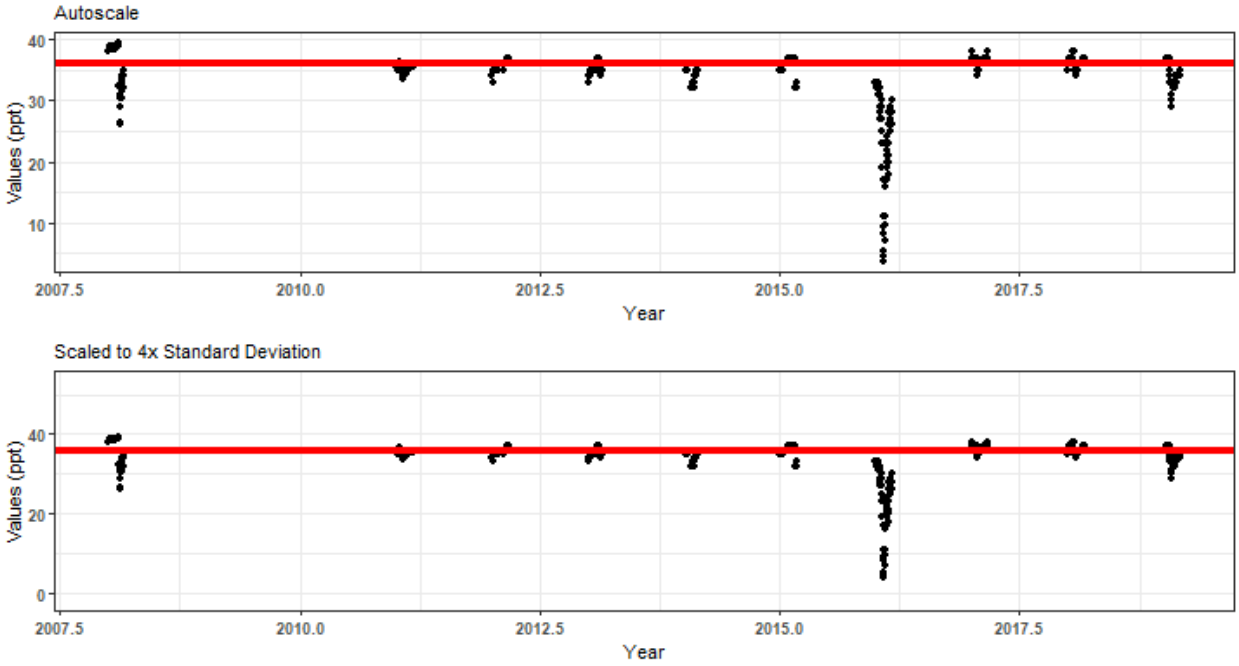
Senn Slope = -0.0454545454545455, Senn Intercept = 36
Trend = -1, tau = -0.0636, p = 0.0013
Linear Trendline: y = -0.29313486190503x + 625.49286308167





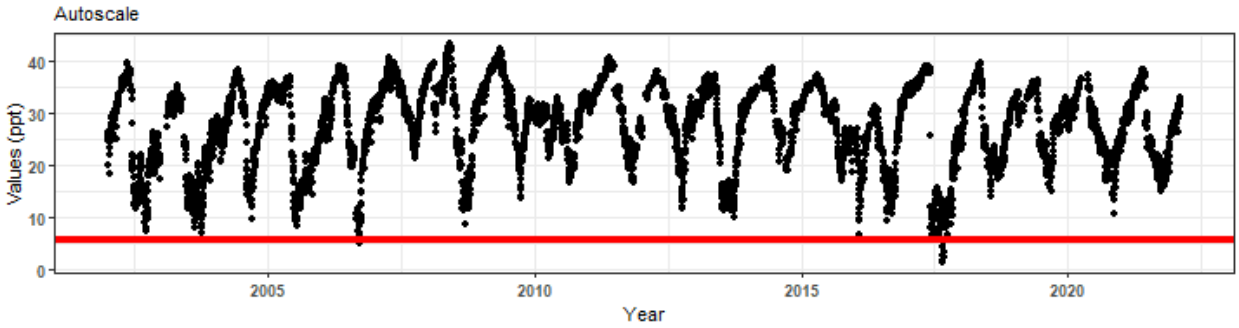
Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255654081350200

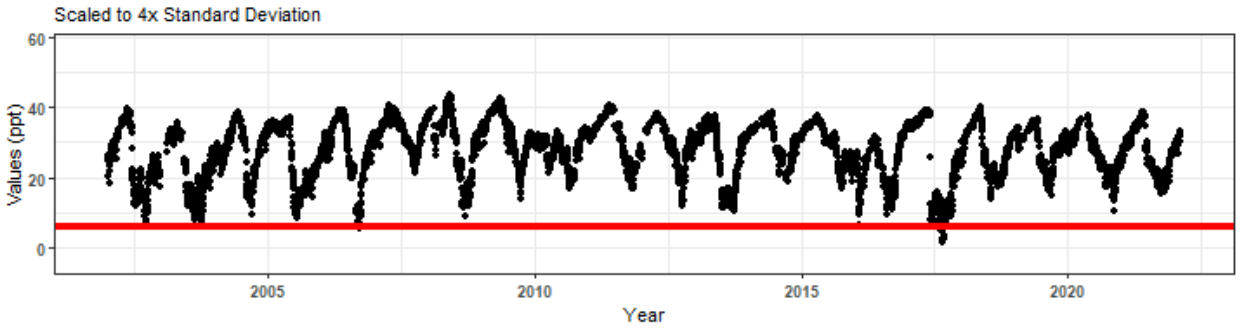
Senn Slope = 0, Senn Intercept = 36
Trend = 0, tau = -0.0501, p = 0.0718
Linear Trendline: y = -0.263119128841335x + 564.334466575947



Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

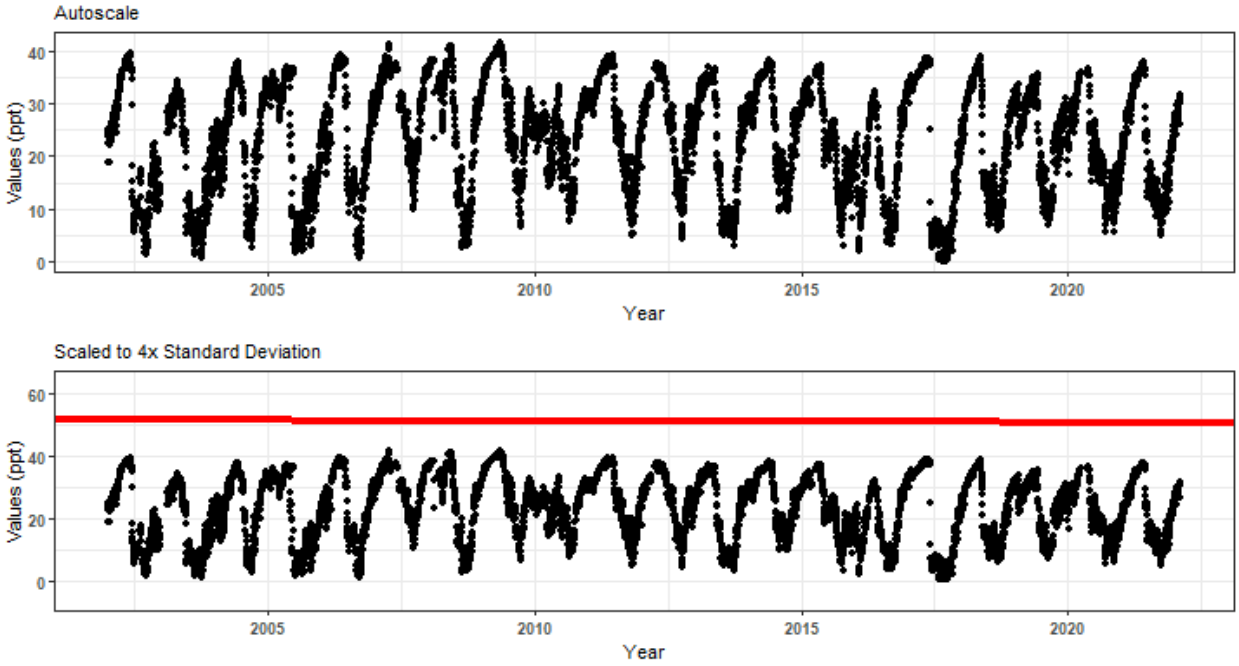
Senn Slope = 0.0044189288720536, Senn Intercept = -2.96293711314273
Trend = 0, tau = 0.0051, p = 0.6492
Linear Trendline: y = -0.00561621590655441x + 39.2087798790061





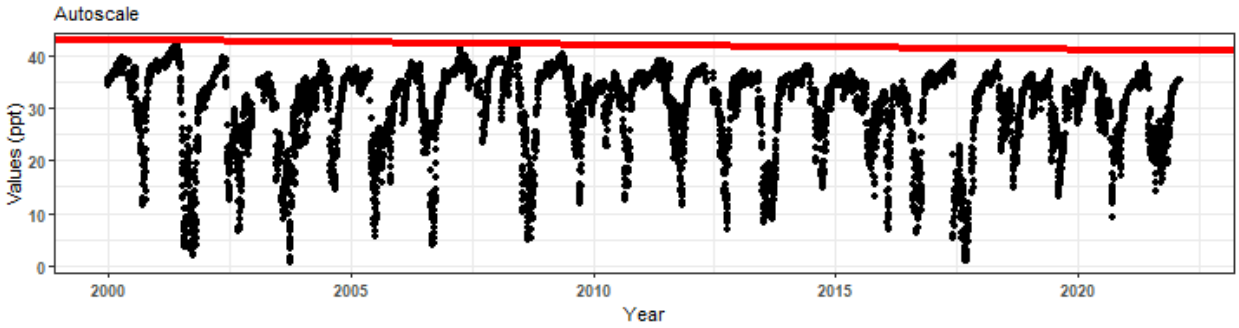
Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

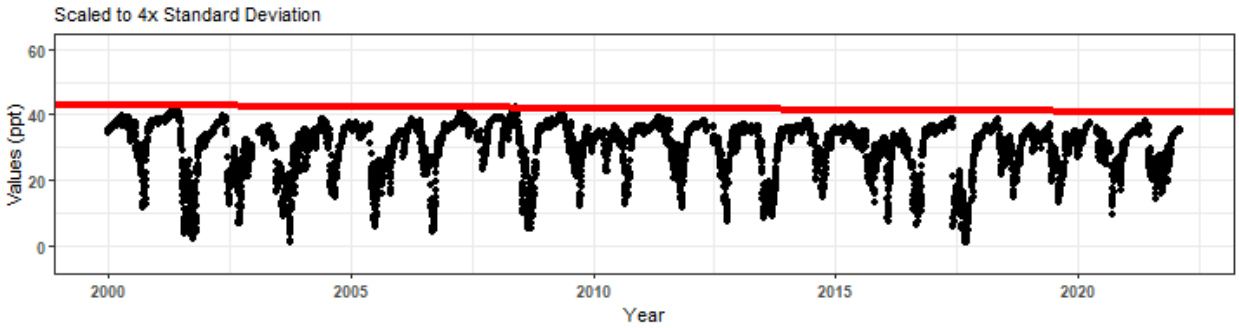
Senn Slope = -0.0420138888888898, Senn Intercept = 135.460677083333
Trend = -1, tau = -0.0295, p = 0.0003
Linear Trendline: y = -0.0906384094641716x + 205.645596764183



Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

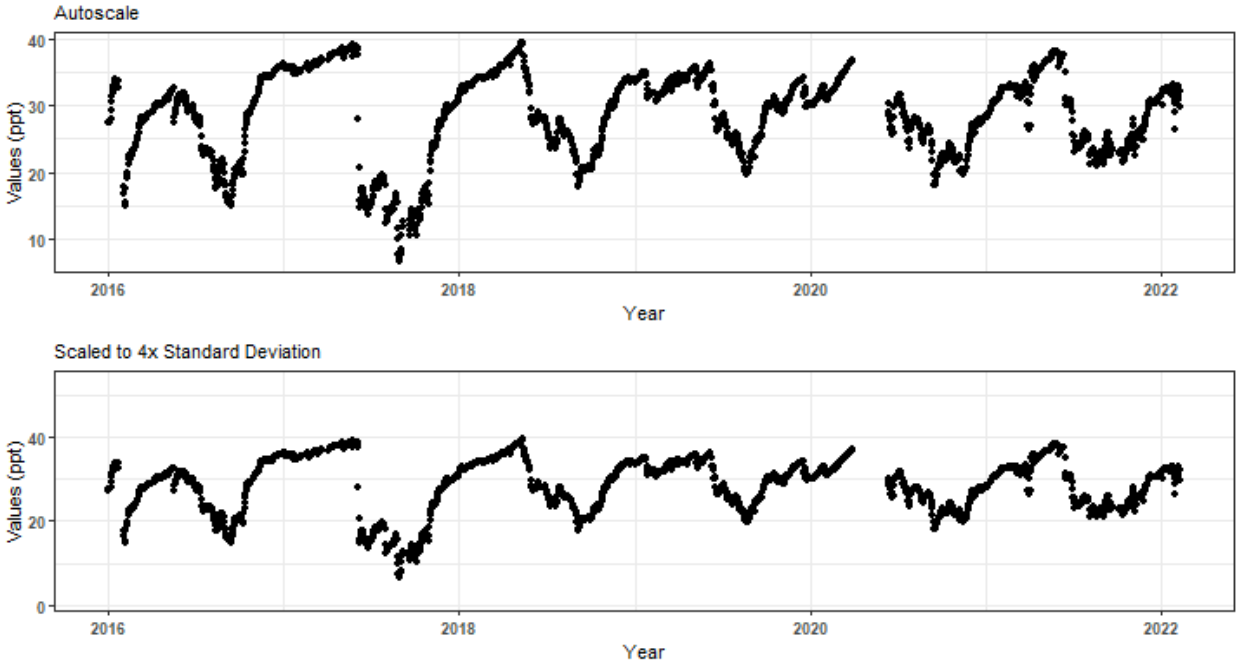
Senn Slope = -0.0915972222222223, Senn Intercept = 226.209967496867
Trend = -1, tau = -0.1148, p = 0
Linear Trendline: y = -0.0679282121012376x + 166.934549668287





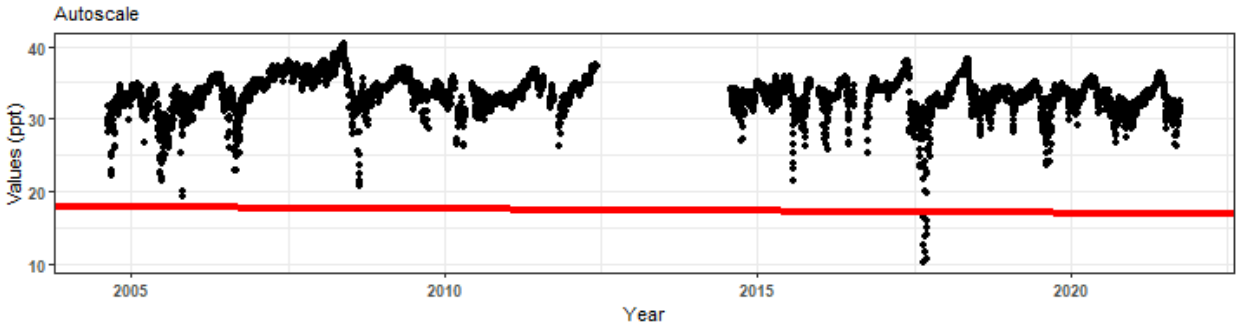
Data Points with Trendlines for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

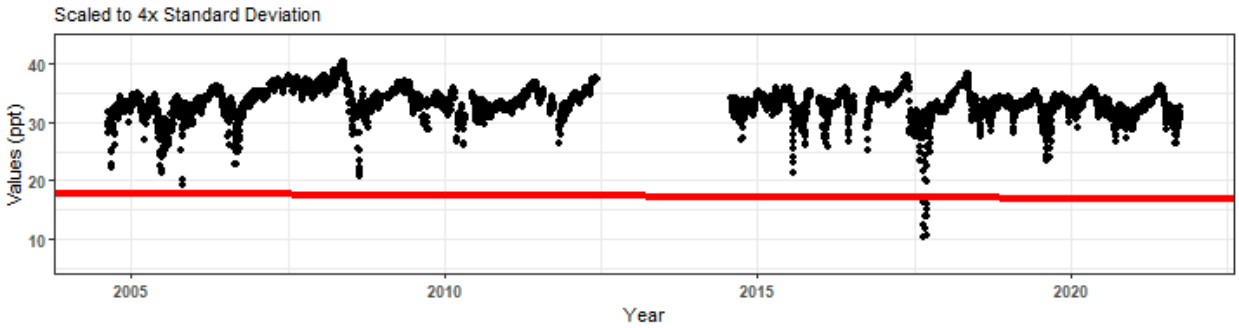
Senn Slope = 0.0367960164835163, Senn Intercept = -431.273041376749
Trend = 0, tau = 0.0203, p = 0.5144
Linear Trendline: y = 0.222462119535974x + -420.501614231042



Data Points with Trendlines for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB02

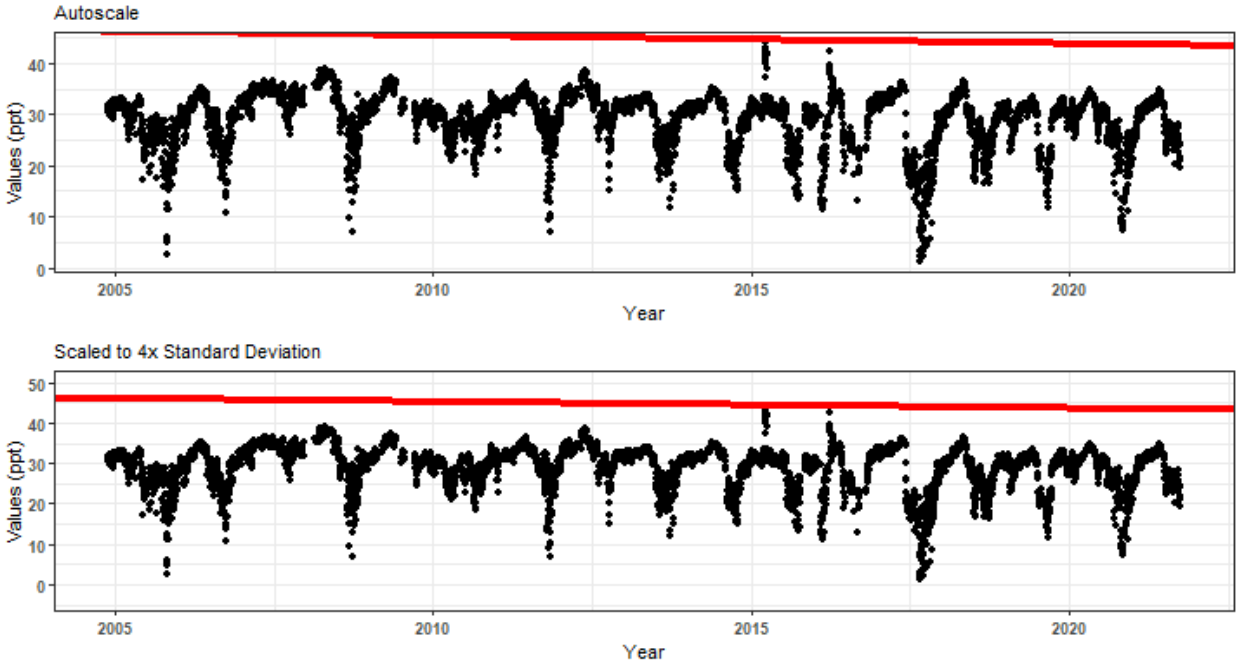
Senn Slope = -0.0555989583333334, Senn Intercept = 129.388533442982
Trend = -1, tau = -0.1043, p = 0
Linear Trendline: y = -0.0552807147191861x + 144.528877101757





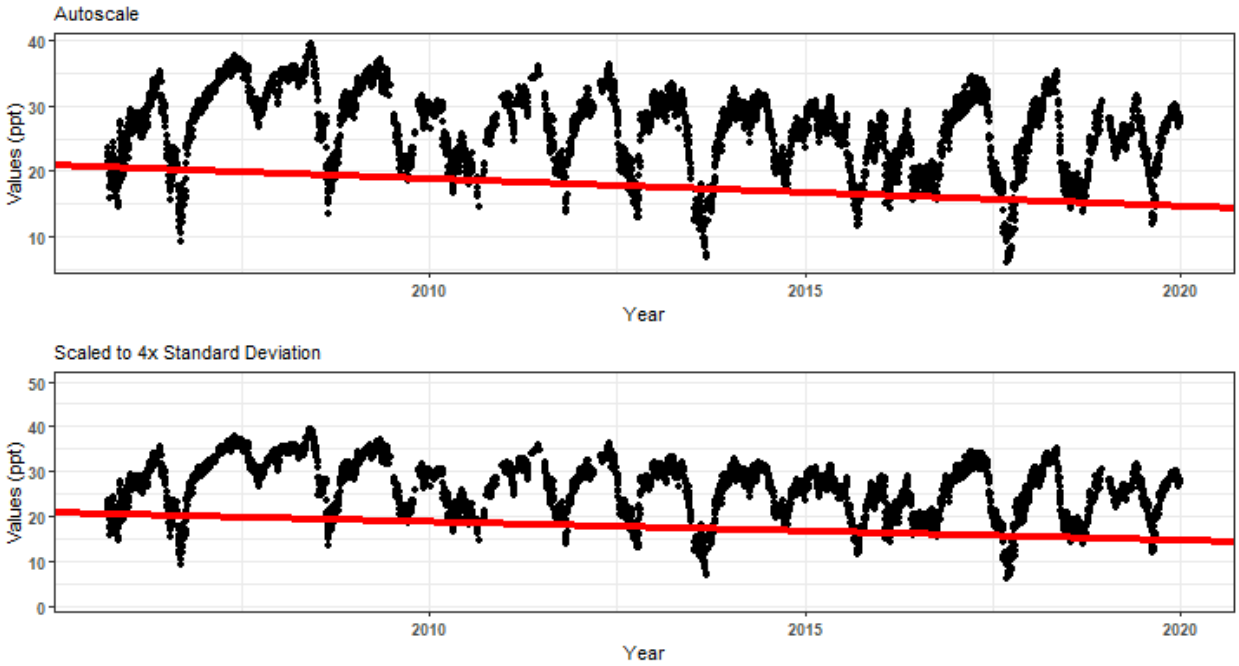
Data Points with Trendlines for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB03

Senn Slope = -0.159027777777778, Senn Intercept = 365.098406597694
Trend = -1, tau = -0.1706, p = 0
Linear Trendline: y = -0.197158730514183x + 426.127137437363



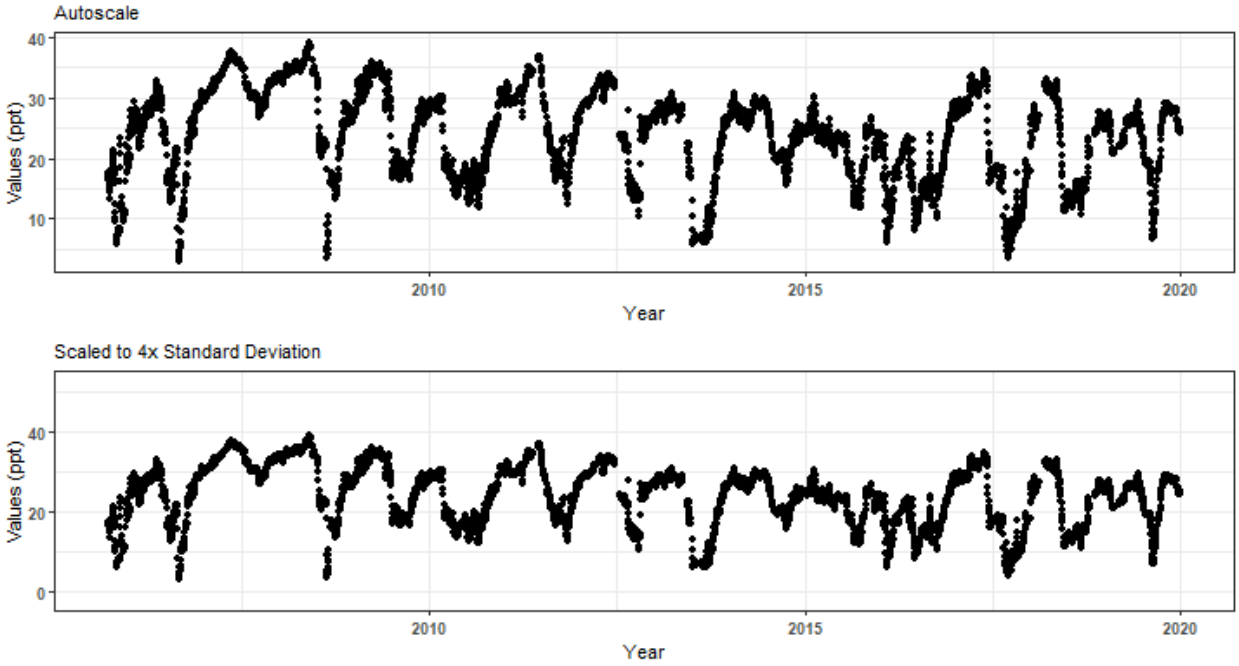
Data Points with Trendlines for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP1A

Senn Slope = -0.41484375, Senn Intercept = 852.71640625
Trend = -1, tau = -0.2793, p = 0
Linear Trendline: y = -0.461313405632791x + 954.940002354675



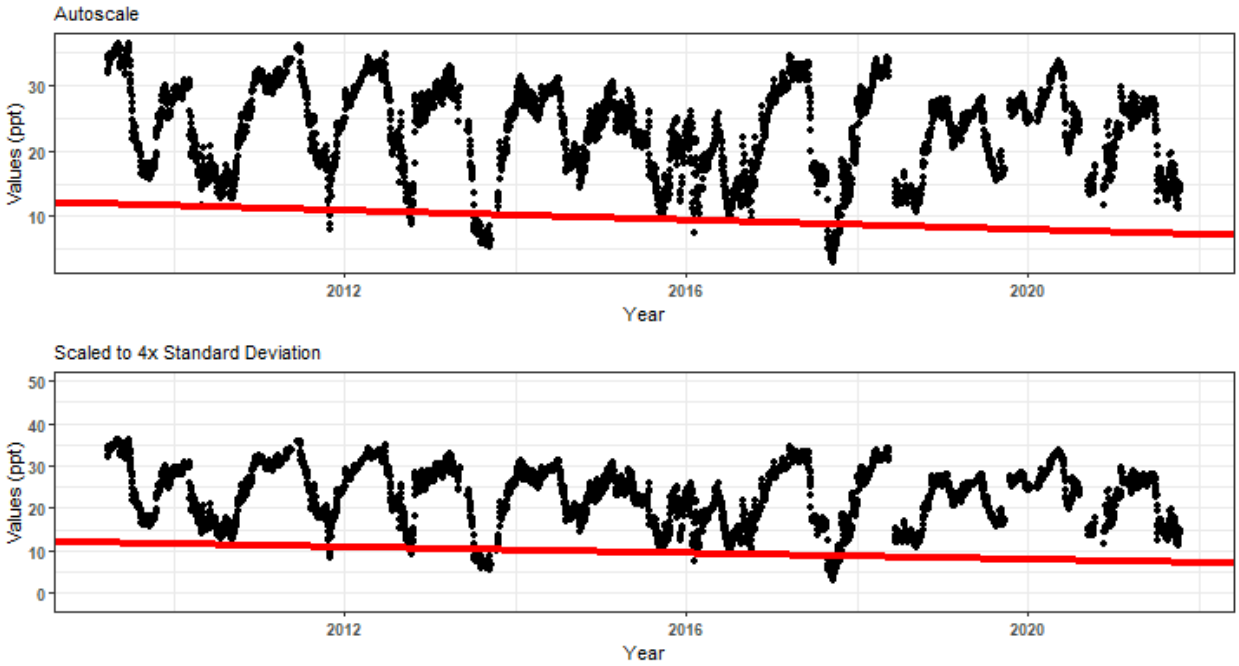
Data Points with Trendlines for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP2B

Senn Slope = -0.484895833333333, Senn Intercept = 1086.97016361532
Trend = -1, tau = -0.2792, p = 0
Linear Trendline: y = -0.490412762225093x + 1010.93850283644



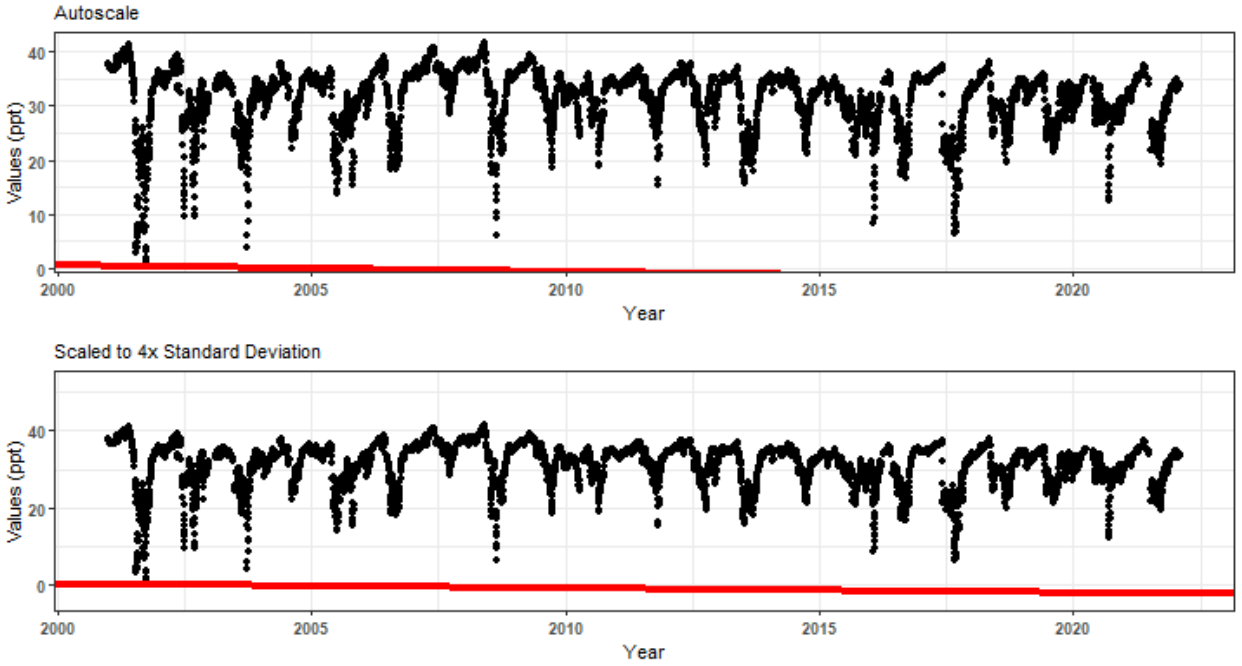
Data Points with Trendlines for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP3C

Senn Slope = -0.363932291666667, Senn Intercept = 743.291290509258
Trend = -1, tau = -0.2067, p = 0
Linear Trendline: y = -0.282445728389998x + 592.167490574561



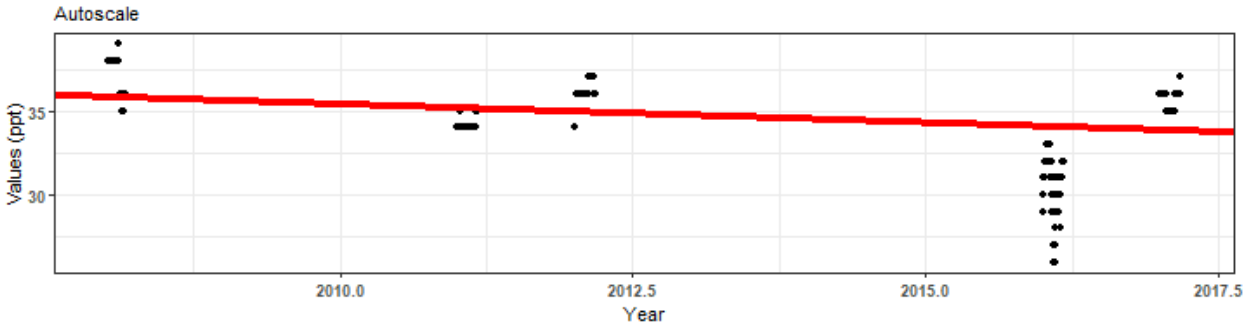
Data Points with Trendlines for Rookery Bay Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

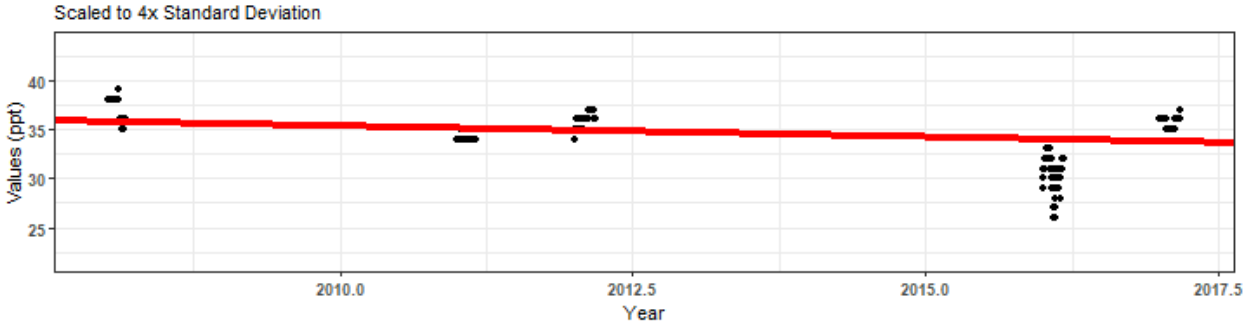
Senn Slope = -0.116927083333334, Senn Intercept = 234.546345263533
Trend = -1, tau = -0.1667, p = 0
Linear Trendline: y = -0.0946827311205902x + 221.905012897758



Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 25512308132130O

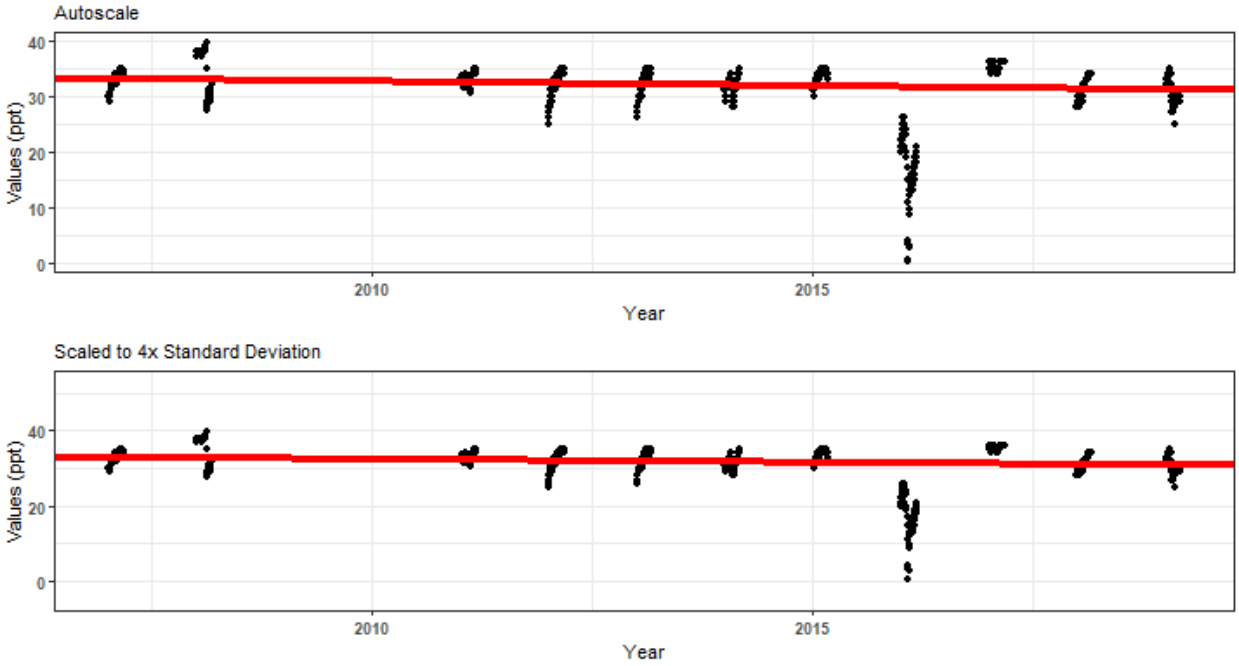
Senn Slope = -0.222222222222222, Senn Intercept = 482.111111111111
Trend = -1, tau = -0.171, p = 0
Linear Trendline: y = -0.393069592103871x + 825.846846082415





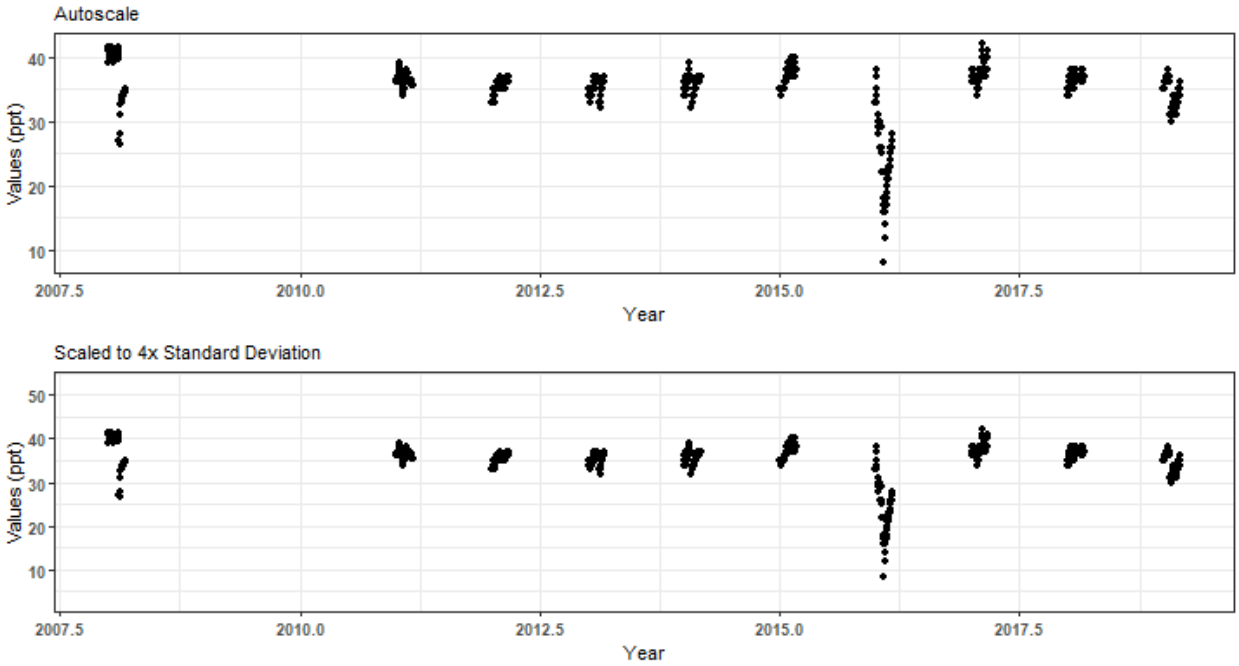
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve 7 | National Water Information System | 255432081303900

Senn Slope = -0.166666666666667, Senn Intercept = 367.666666666667
Trend = -1, tau = -0.1307, p = 0
Linear Trendline: y = -0.381647692934698x + 799.890251318757



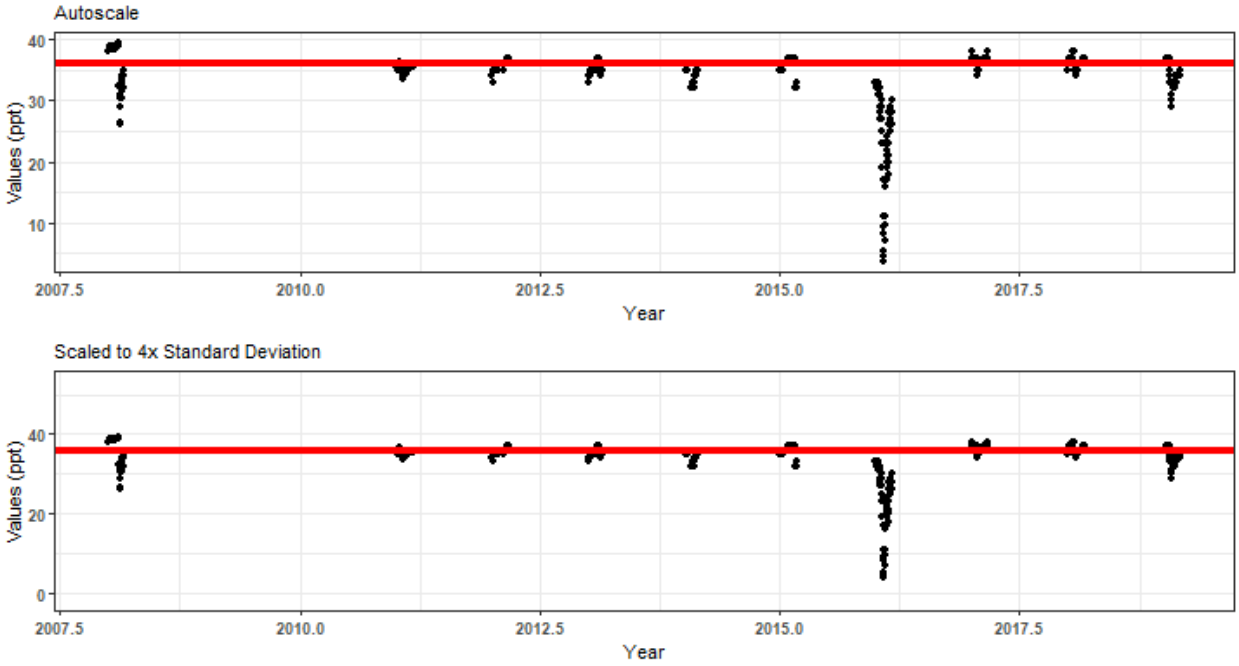
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve 7 | National Water Information System | 255534081324000

Senn Slope = -0.0454545454545455, Senn Intercept = 36
Trend = -1, tau = -0.0636, p = 0.0013
Linear Trendline: y = -0.29313486190503x + 625.49286308167



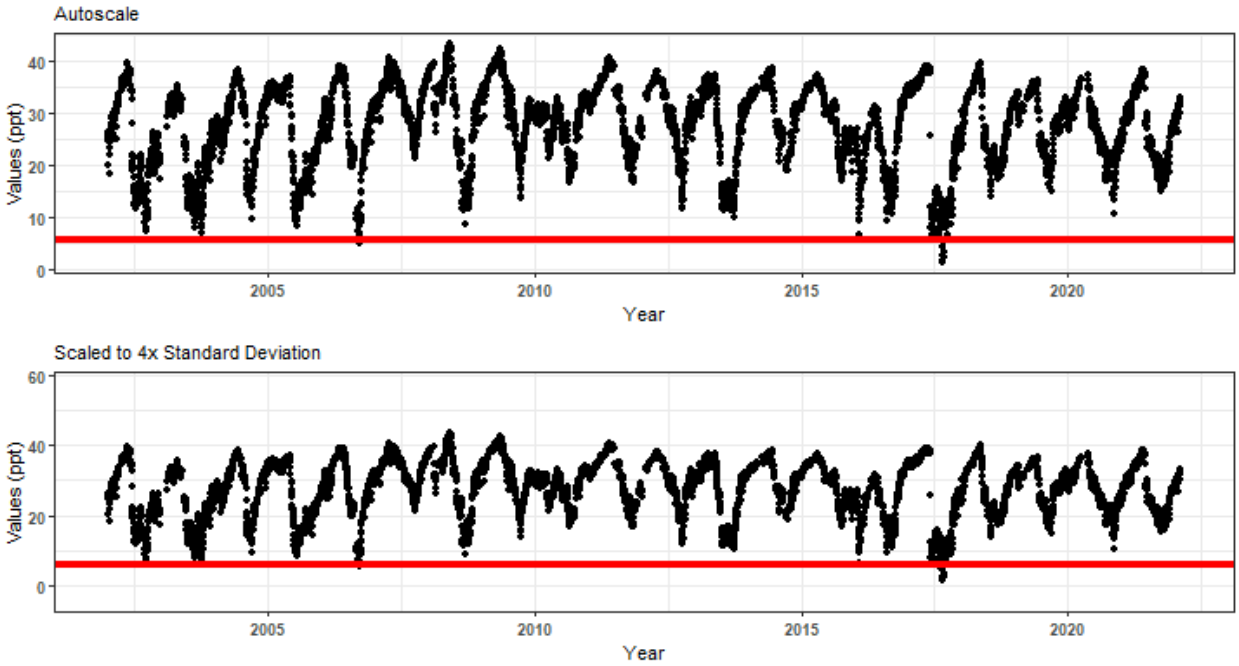
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255654081350200

Senn Slope = 0, Senn Intercept = 36
Trend = 0, tau = -0.0501, p = 0.0718
Linear Trendline: y = -0.263119128841335x + 564.334466575947



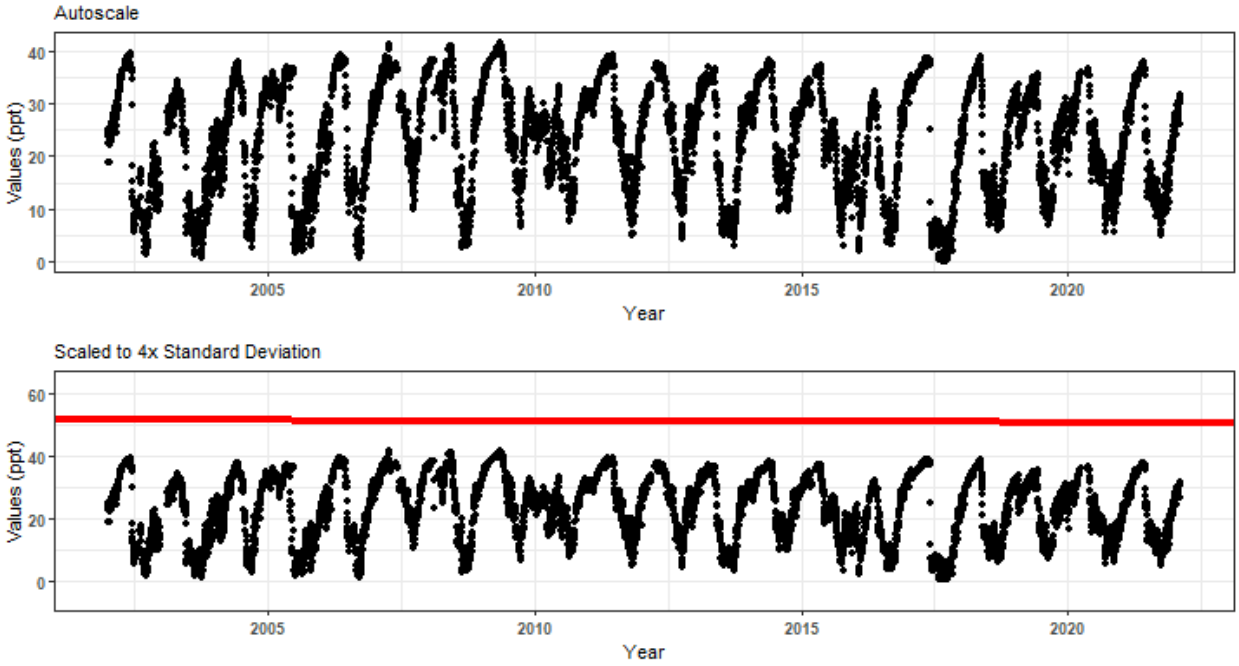
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

Senn Slope = 0.0044189288720536, Senn Intercept = -2.96293711314273
Trend = 0, tau = 0.0051, p = 0.6492
Linear Trendline: y = -0.00561621590655441x + 39.2087798790061



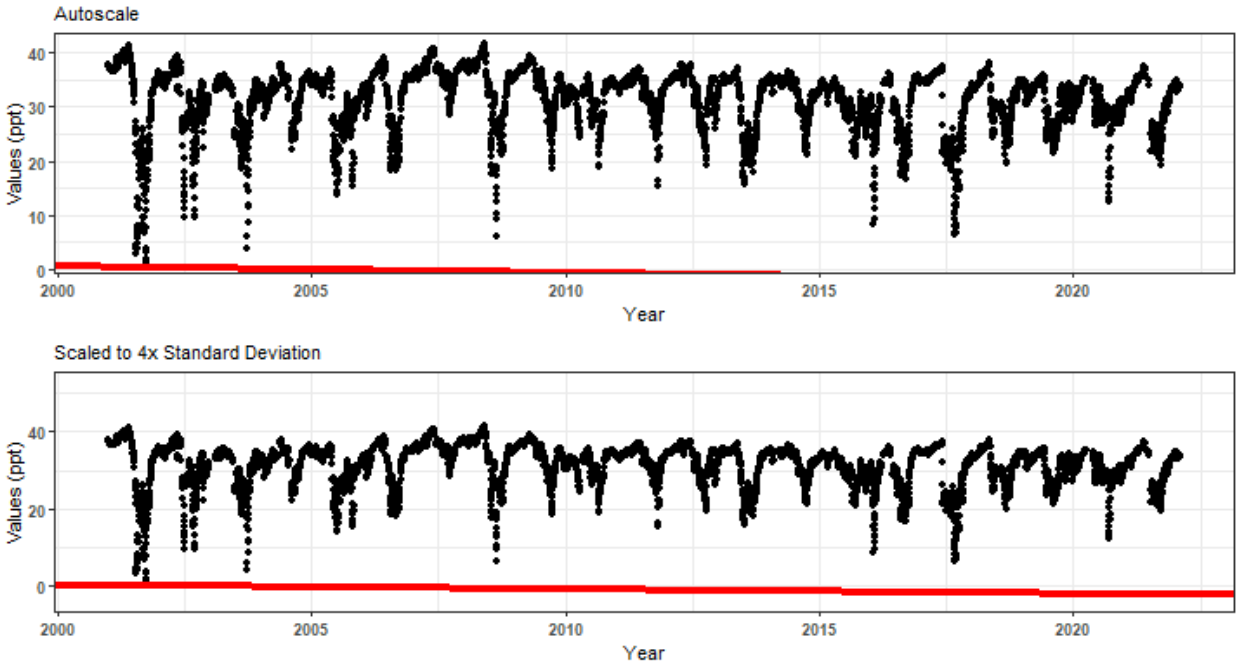
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

Senn Slope = -0.0420138888888898, Senn Intercept = 135.460677083333
Trend = -1, tau = -0.0295, p = 0.0003
Linear Trendline: y = -0.0906384094641716x + 205.645596764183



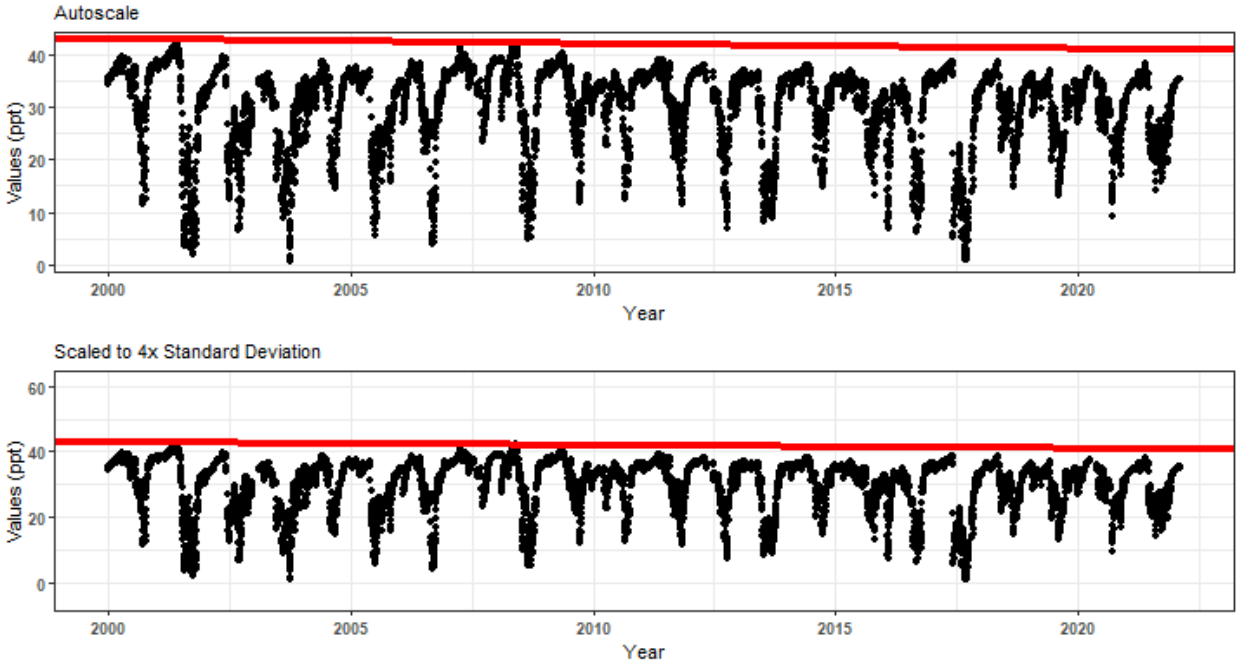
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

Senn Slope = -0.116927083333334, Senn Intercept = 234.546345263533
Trend = -1, tau = -0.1667, p = 0
Linear Trendline: y = -0.0946827311205902x + 221.905012897758



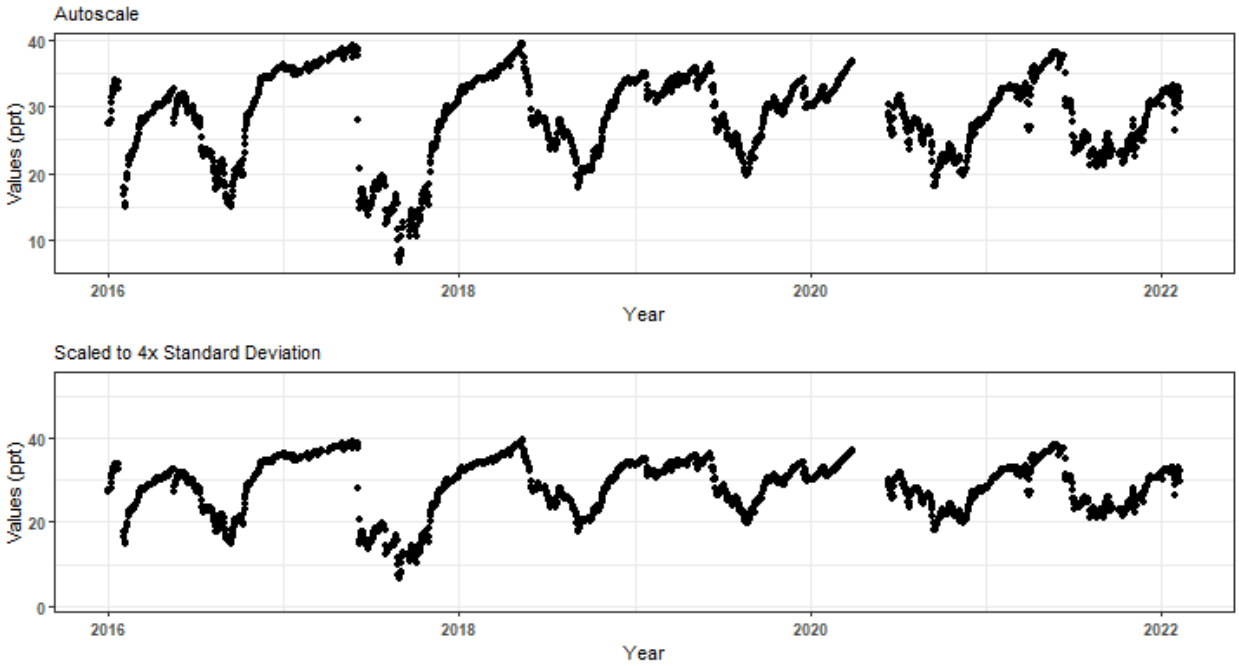
Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

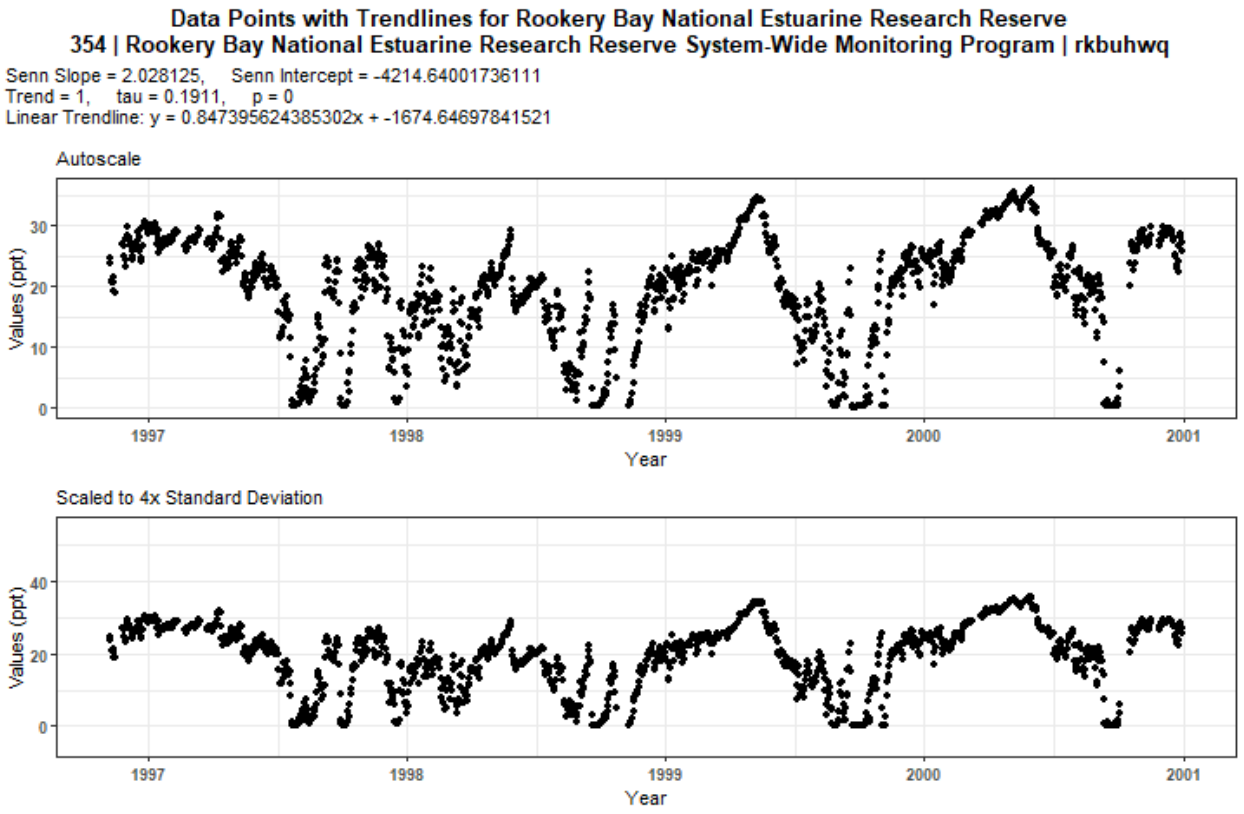
Senn Slope = -0.0915972222222223, Senn Intercept = 226.209967496867
Trend = -1, tau = -0.1148, p = 0
Linear Trendline: y = -0.0679282121012376x + 166.934549668287



Data Points with Trendlines for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

Senn Slope = 0.0367960164835163, Senn Intercept = -431.273041376749
Trend = 0, tau = 0.0203, p = 0.5144
Linear Trendline: y = 0.222462119535974x + -420.501614231042





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$MonitoringID == Mon_IDs[i]]
      Mon_name <- paste(KT.Stats$ProgramID[KT.Stats$MonitoringID == Mon_IDs[i]],
         KT.Stats$ProgramName[KT.Stats$MonitoringID == Mon_IDs[i]],
         KT.Stats$ProgramLocationID[KT.Stats$MonitoringID == Mon_IDs[i]],
         sep = " | ")

      ##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(outlier.size = 0.5) +
         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))) +
         theme_bw() +
         theme(axis.text.x = element_text(face = "bold"),
               axis.text.y = element_text(face = "bold"))

      p2 <- ggplot(data = data[data$Use_In_Analysis == TRUE &
                               data$MonitoringID == Mon_IDs[i], ],
                   aes(x = Year, y = ResultValue, group = Year)) +
         geom_boxplot(outlier.size = 0.5) +
         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))) +
         theme_bw() +
         theme(axis.text.x = element_text(face = "bold"),
               axis.text.y = element_text(face = "bold"))
```

```
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(outlier.size = 0.5) +
   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))) +
   theme_bw() +
   theme(axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

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

p0 <- ggplot() + labs(title = paste0("Summary Box Plots for ",
                                     MA_name, "\n", Mon_name),
                      subtitle = "By Year") +
   theme_bw() + theme(plot.title = element_text(face="bold", hjust=0.5),
         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(outlier.size = 0.5) +
   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))) +
   theme_bw() +
   theme(legend.position = "none",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

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(outlier.size = 0.5) +
   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))) +
   theme_bw() +
```

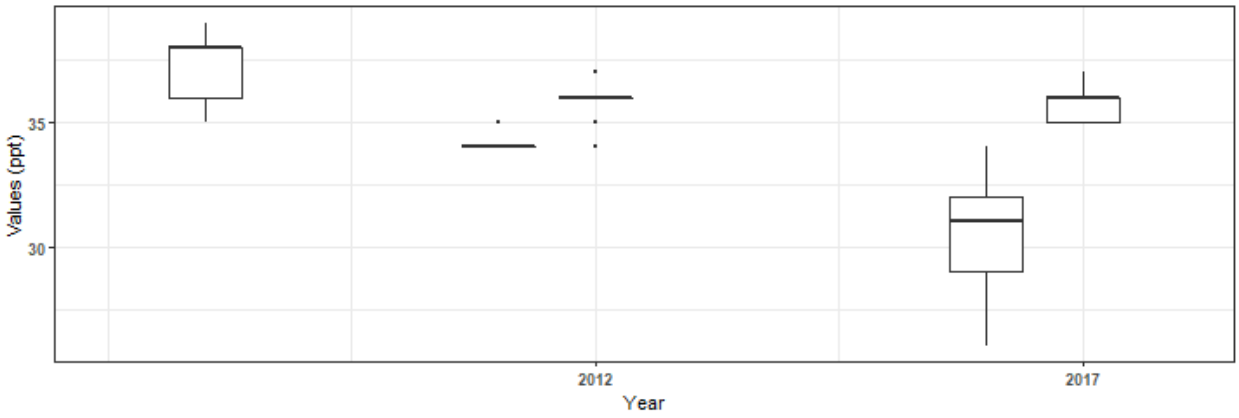
```
   theme(legend.position = "top", legend.box = "horizontal",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold")) +
   guides(color = guide_legend(nrow = 1))

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(outlier.size = 0.5) +
   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))) +
   theme_bw() +
   theme(legend.position = "none",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

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("Summary Box Plots for ",
                                      MA_name, "\n", Mon_name),
                       subtitle = "By Year & Month") + theme_bw() +
   theme(plot.title = element_text(face="bold", hjust=0.5),
         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(outlier.size = 0.5) +
   labs(subtitle = "Autoscale",
        x = "Month", y = paste0("Values (", unit, ")"), fill = "Month") +
   scale_x_continuous(limits = c(0, 13), breaks = seq(3, 12, 3)) +
   theme_bw() +
   theme(legend.position = "none",
         axis.text.x = element_text(face = "bold"),
         axis.text.y = element_text(face = "bold"))

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(outlier.size = 0.5) +
   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)) +
       theme_bw() +
       theme(legend.position = "top", legend.box = "horizontal",
             axis.text.x = element_text(face = "bold"),
             axis.text.y = element_text(face = "bold")) +
       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(outlier.size = 0.5) +
       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)) +
       theme_bw() +
       theme(legend.position = "none",
             axis.text.x = element_text(face = "bold"),
             axis.text.y = element_text(face = "bold"))

    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("Summary Box Plots for ",
                                           MA_name, "\n", Mon_name),
                            subtitle = "By Month") + theme_bw() +
       theme(plot.title = element_text(face="bold", hjust=0.5),
             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)))
  }
}
```

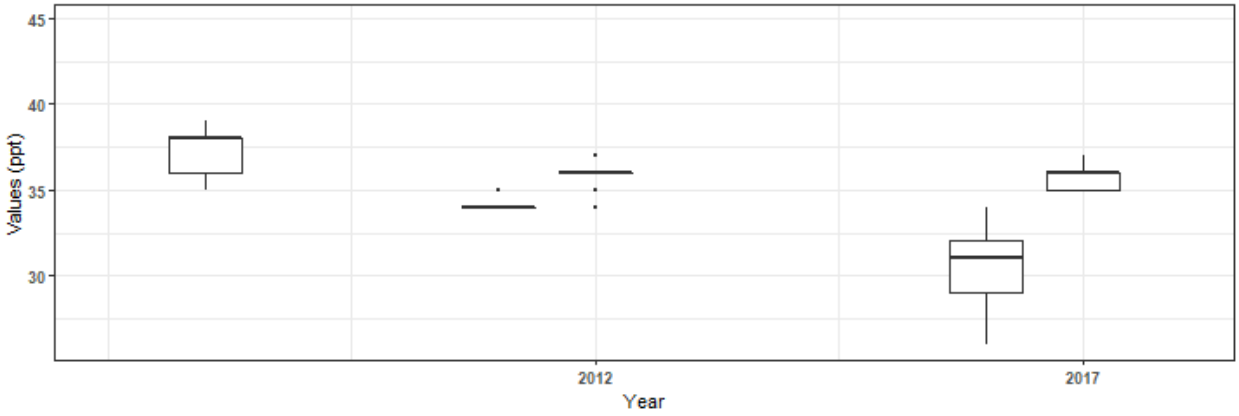
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255123081321300

By Year

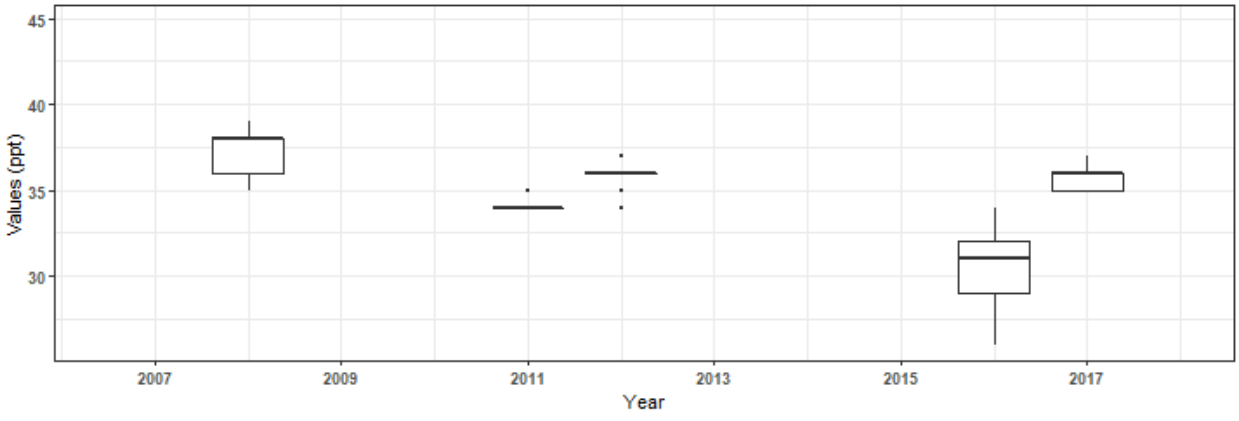
Autoscale



Scaled to 4x Standard Deviation

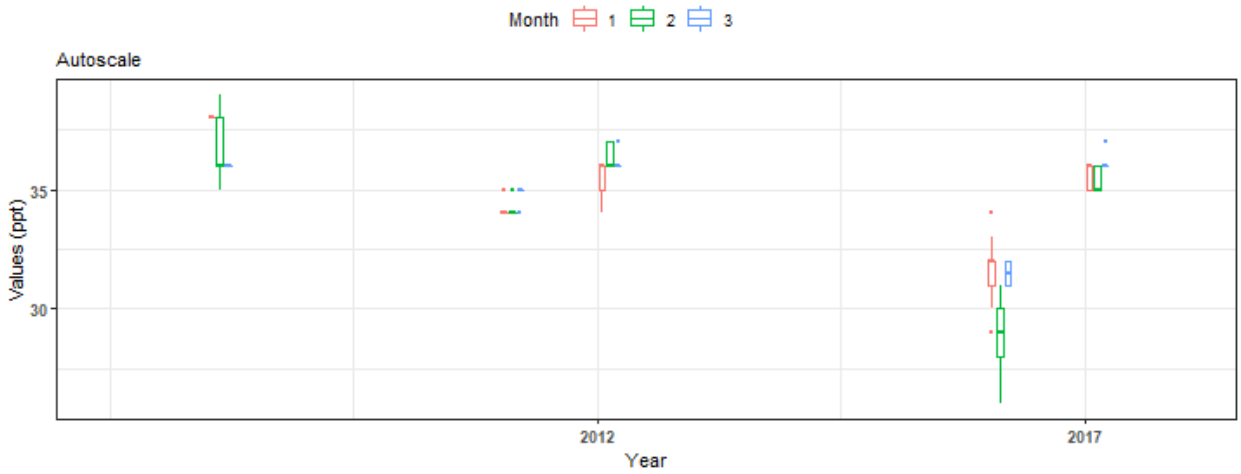


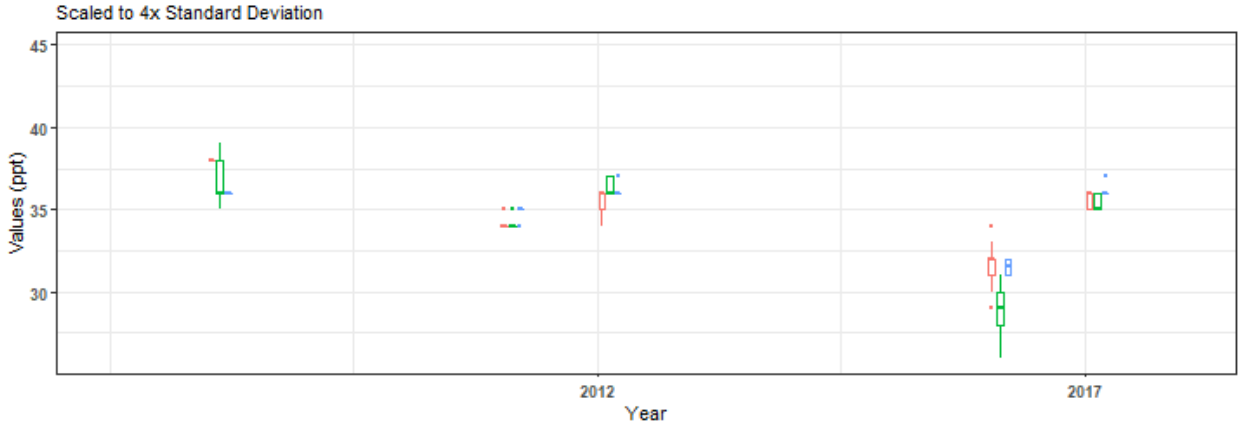
Scaled to 4x Standard Deviation, Last 10 Years

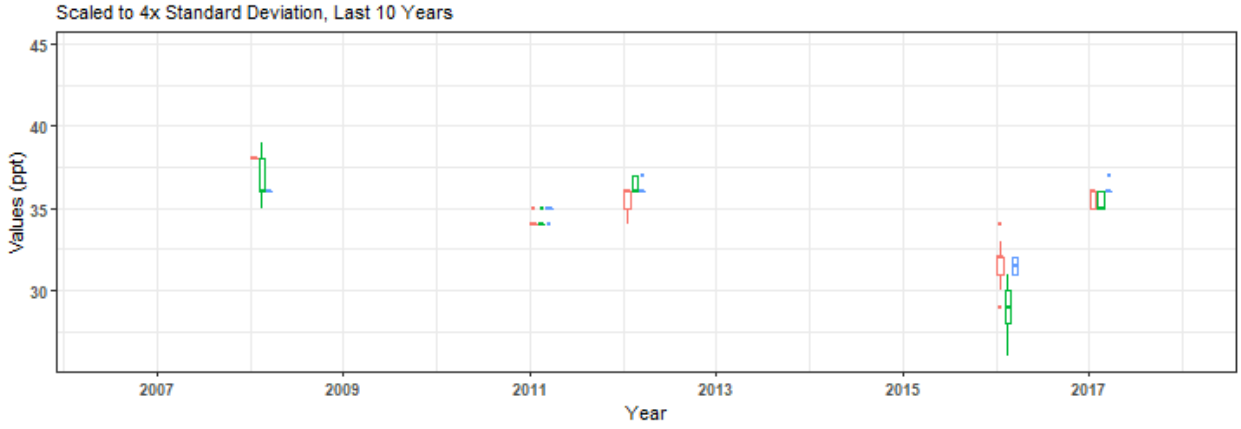


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255123081321300

By Year & Month

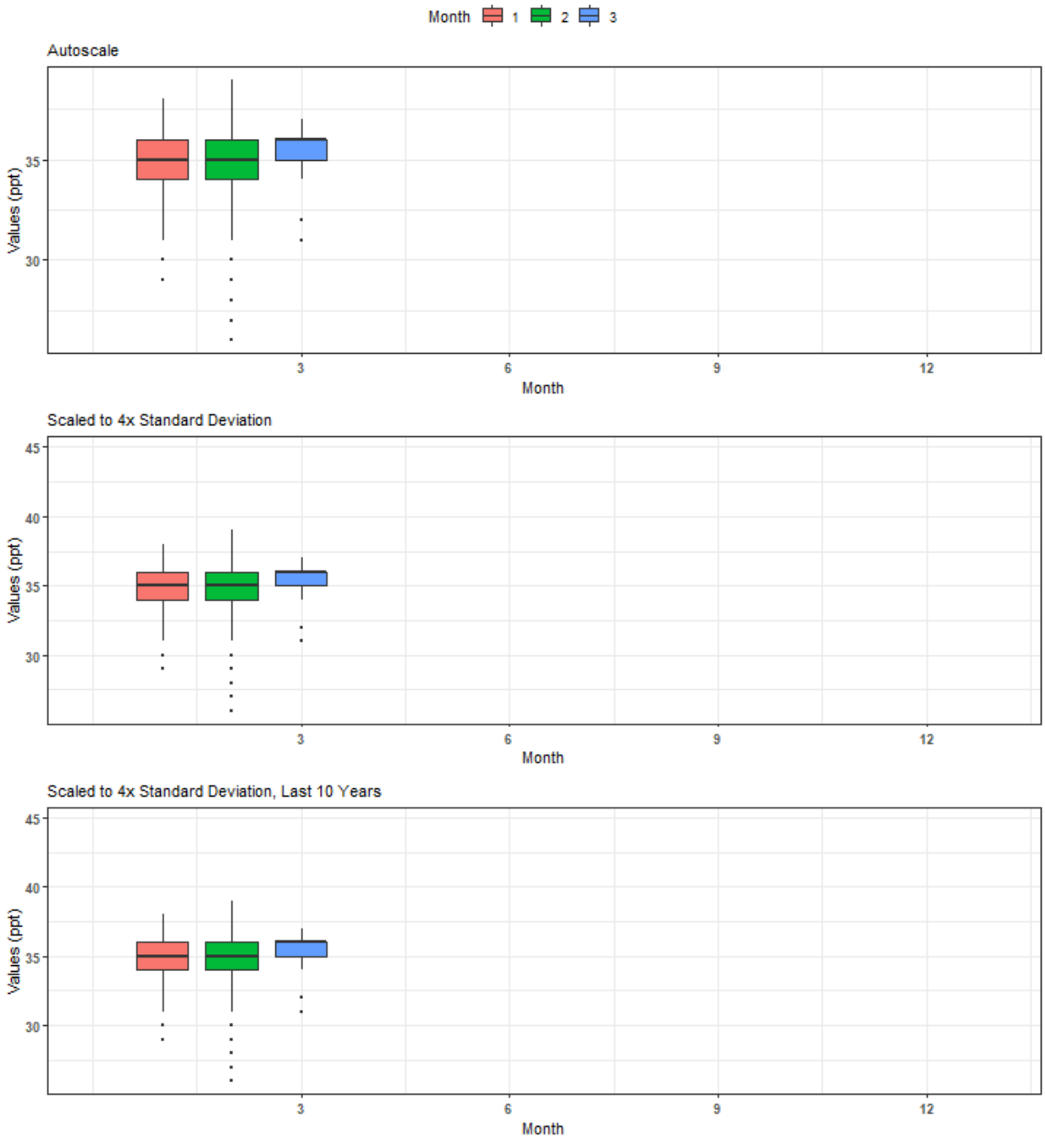






Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255123081321300

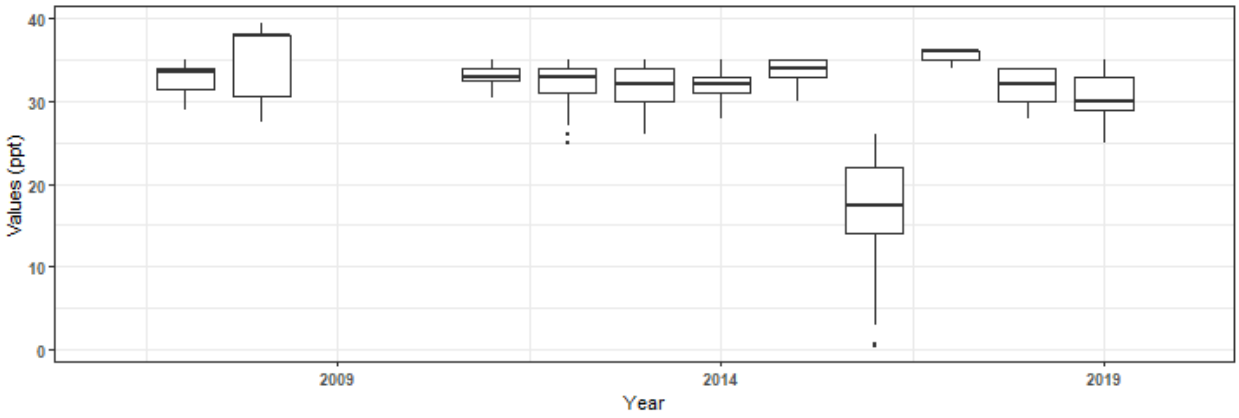
By Month



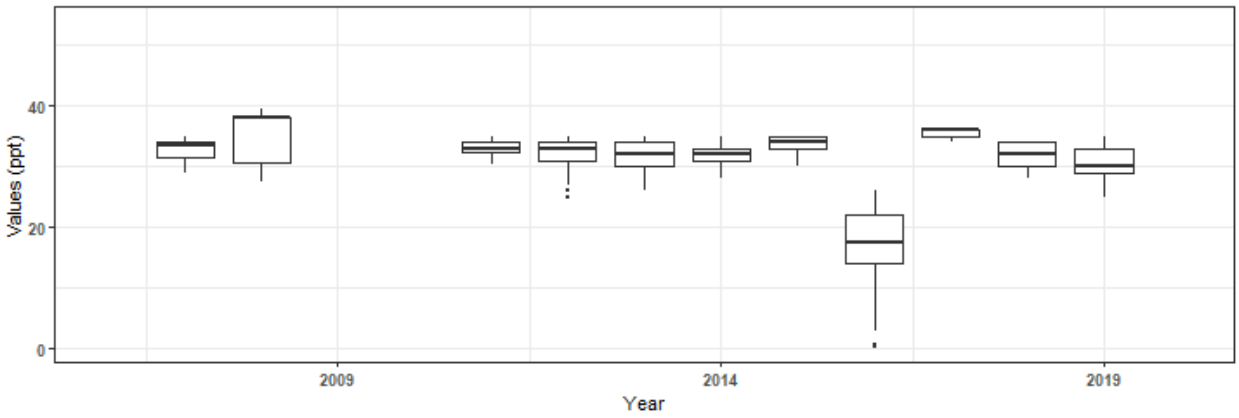
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255432081303900

By Year

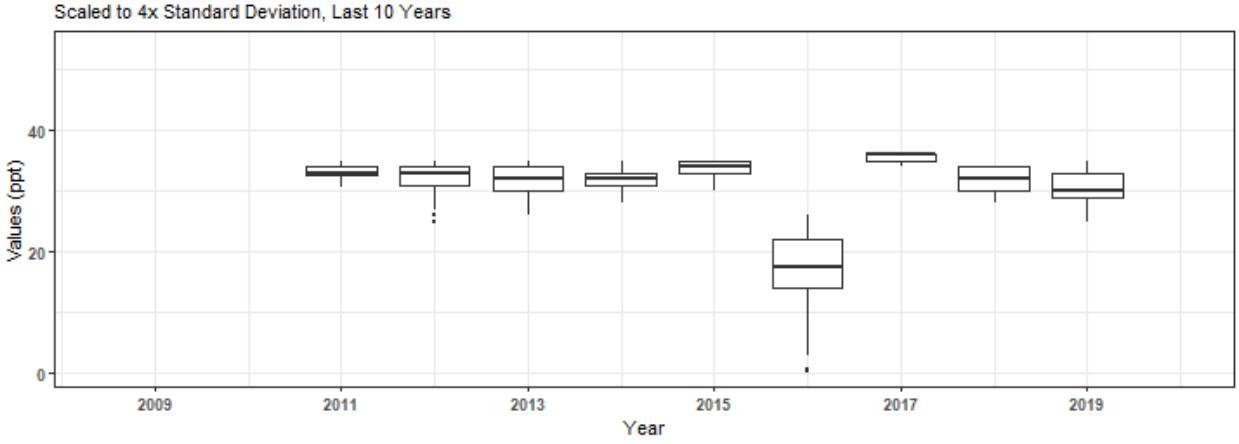
Autoscale



Scaled to 4x Standard Deviation

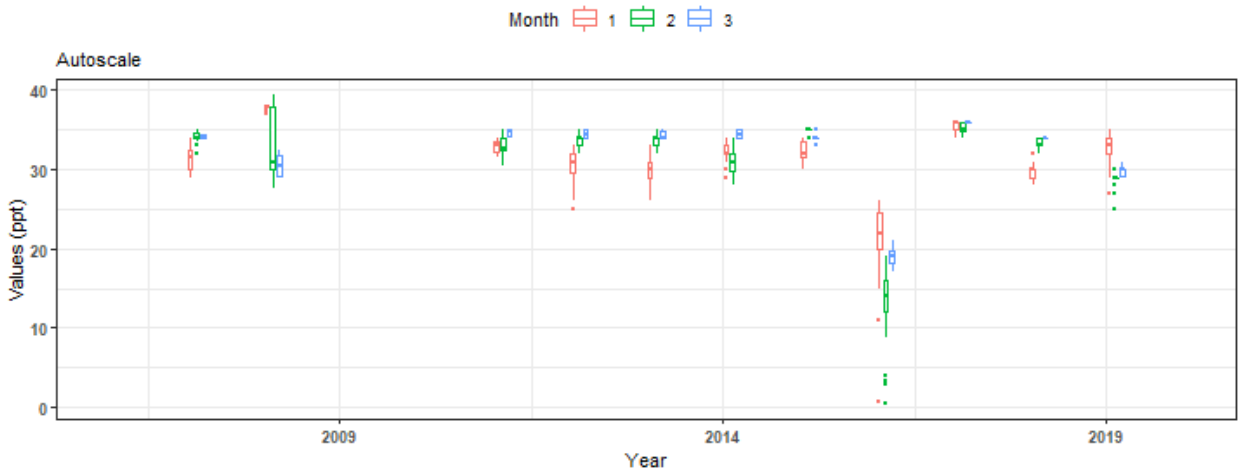


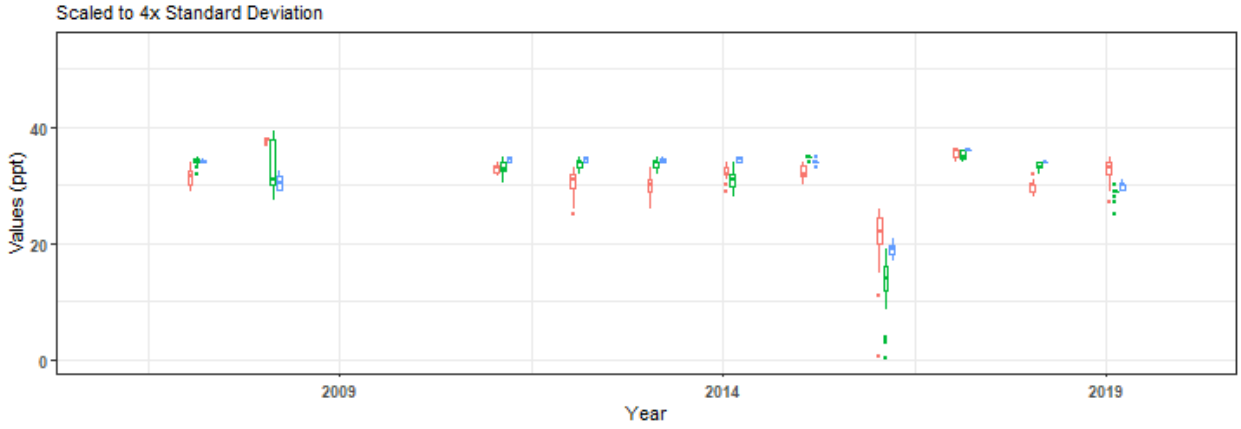
Scaled to 4x Standard Deviation, Last 10 Years

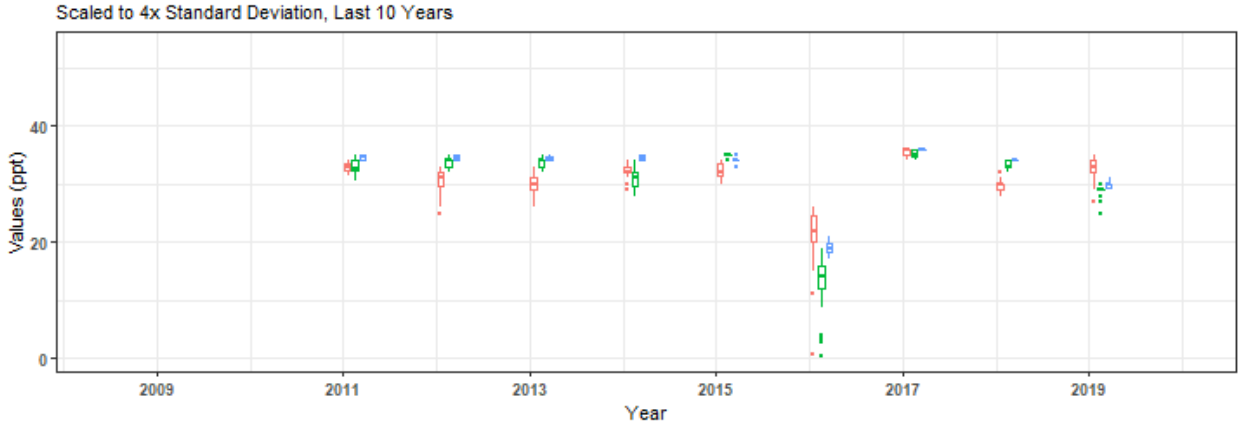


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255432081303900

By Year & Month

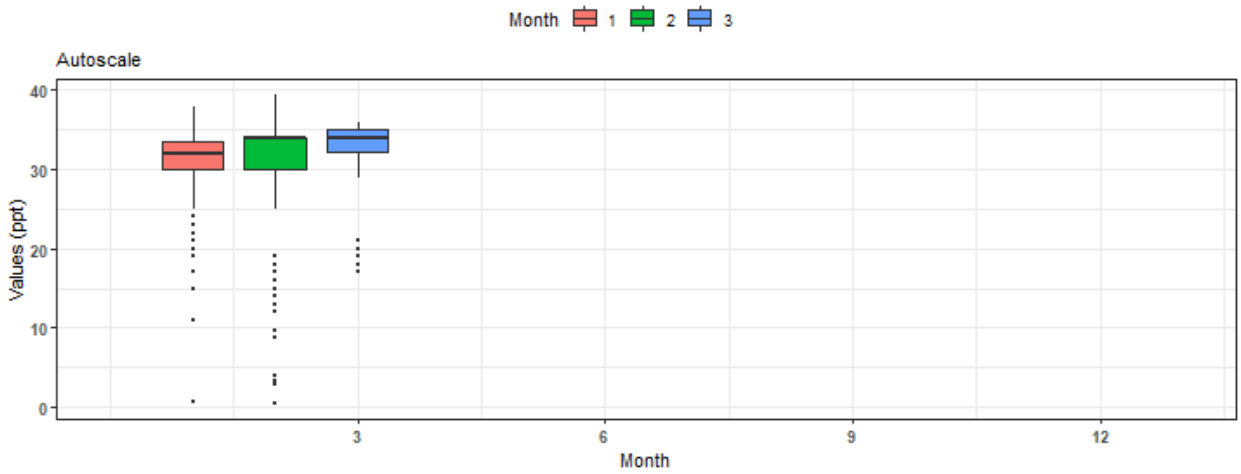


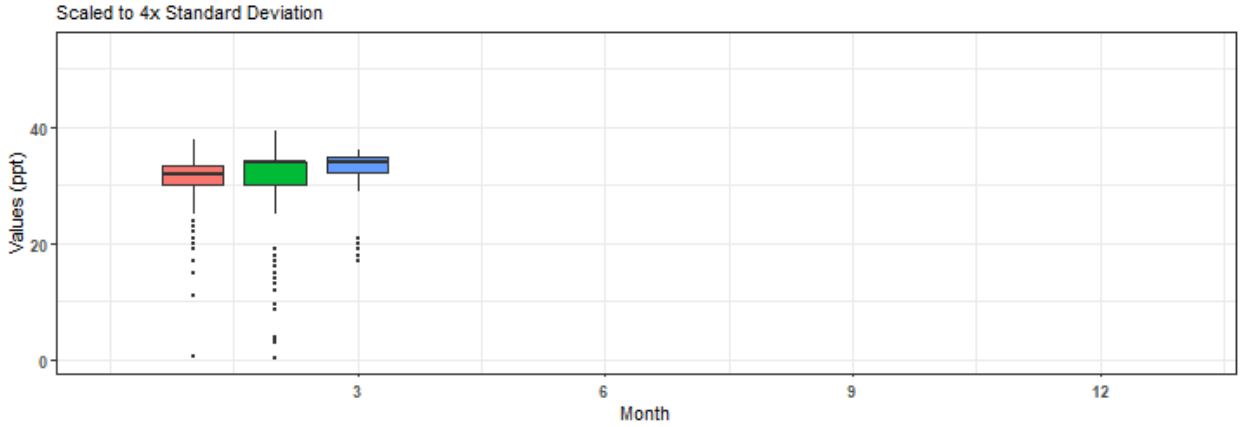


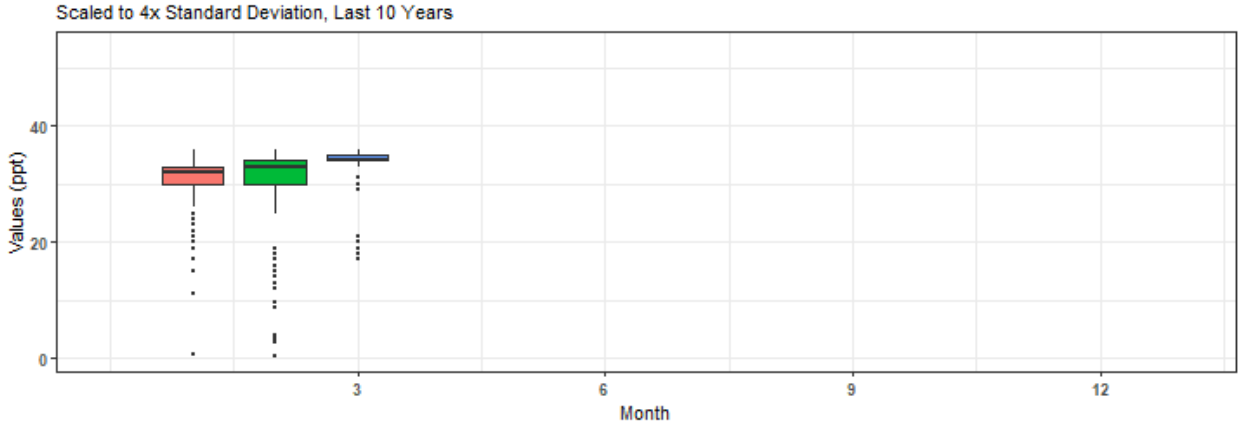


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255432081303900

By Month



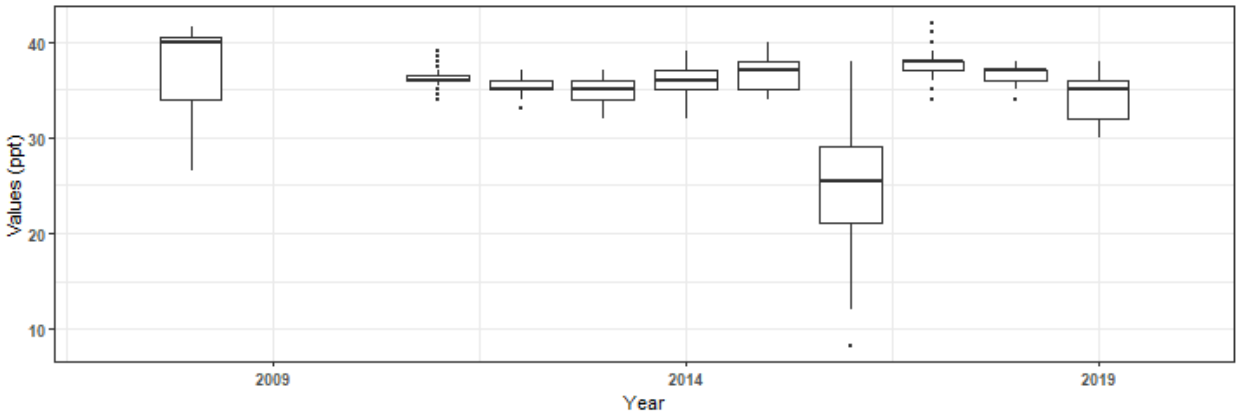




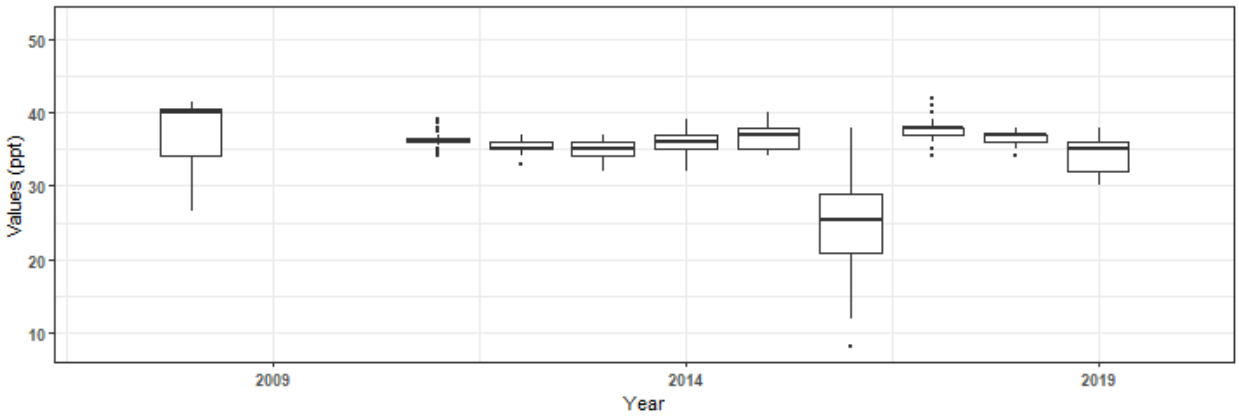
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255534081324000

By Year

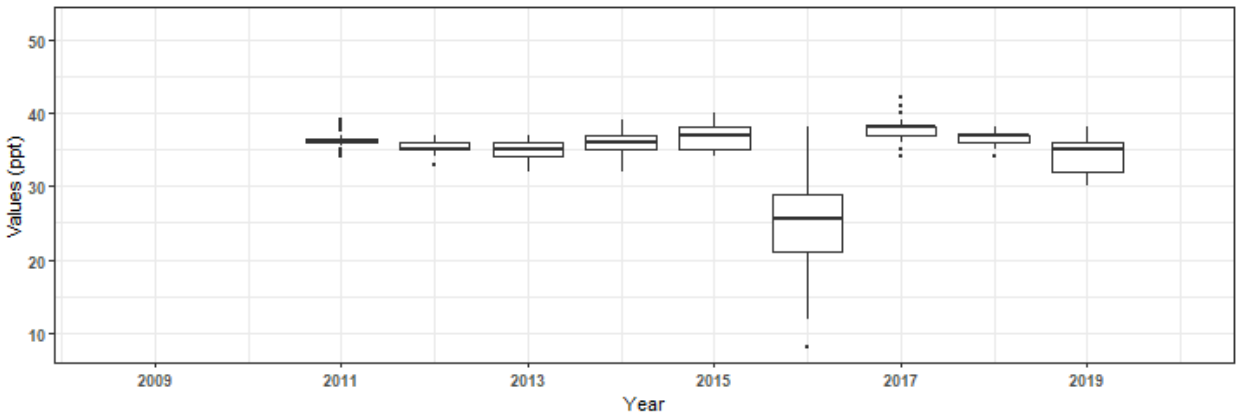
Autoscale



Scaled to 4x Standard Deviation

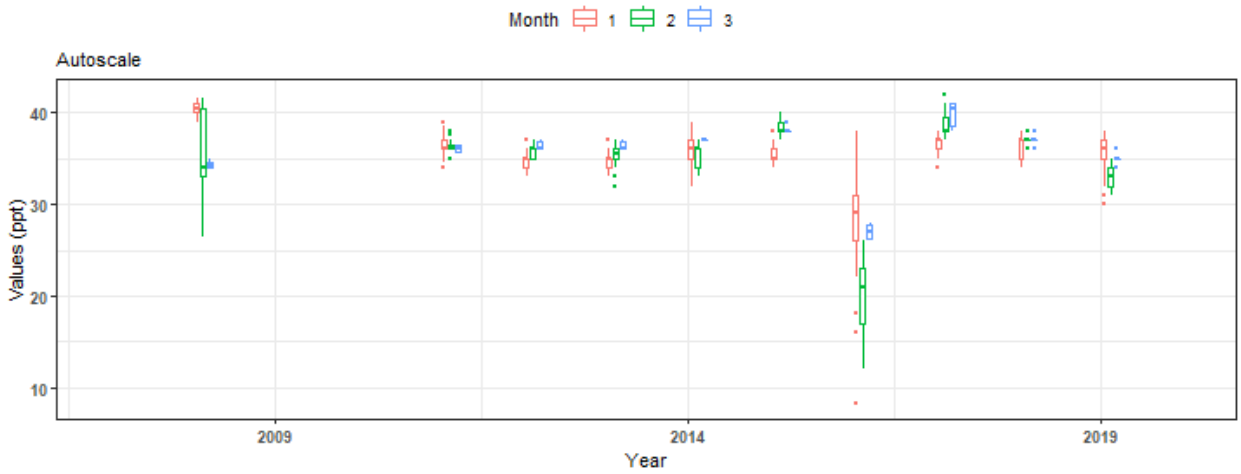


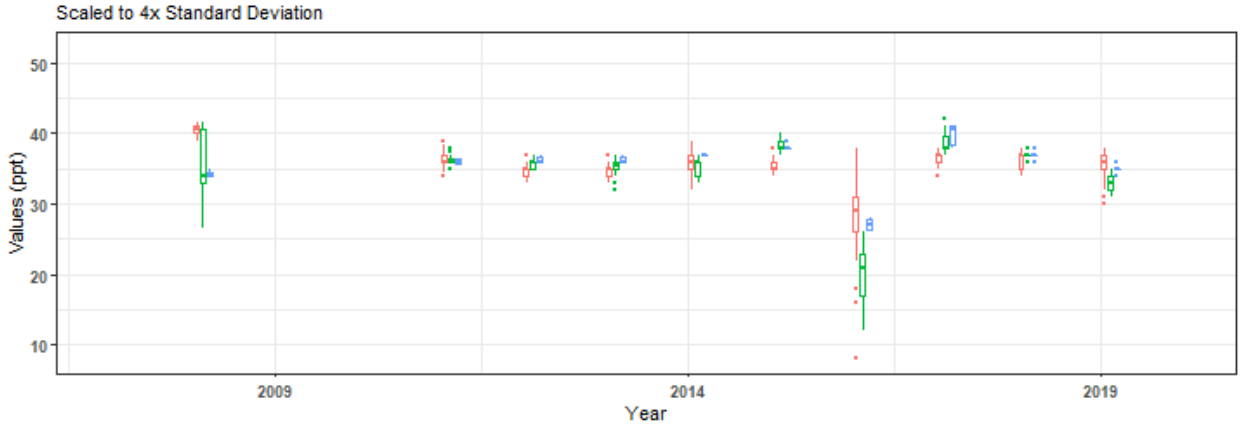
Scaled to 4x Standard Deviation, Last 10 Years

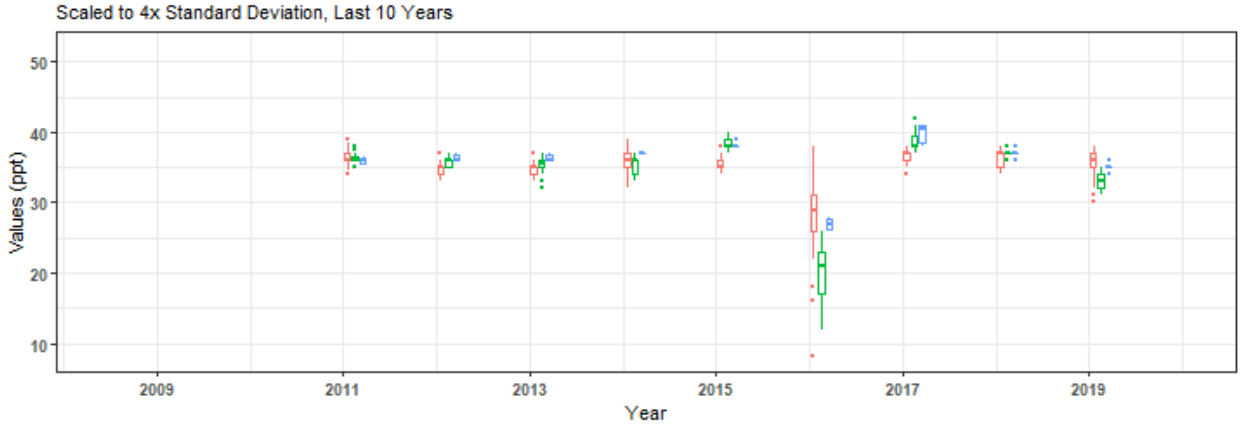


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255534081324000

By Year & Month

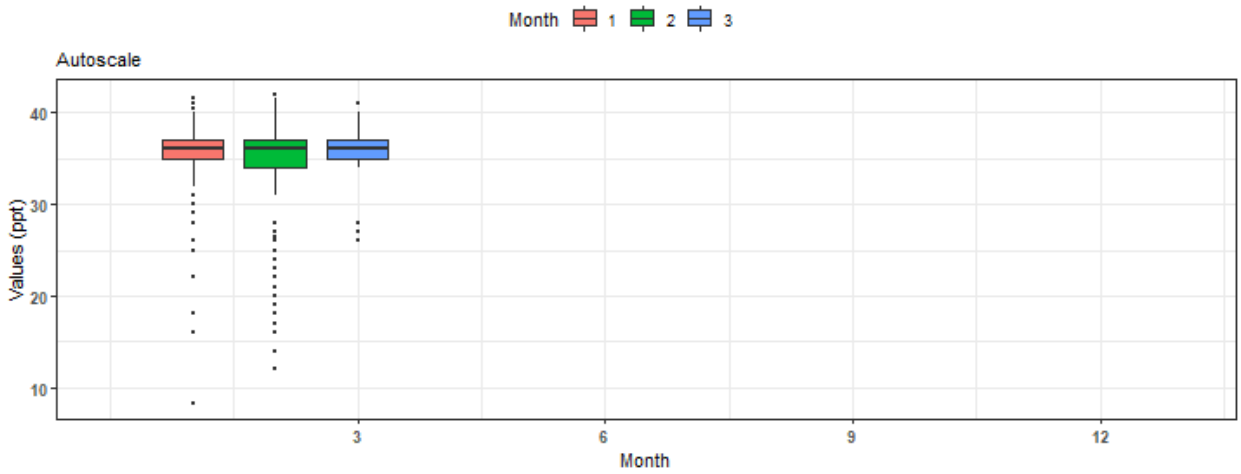


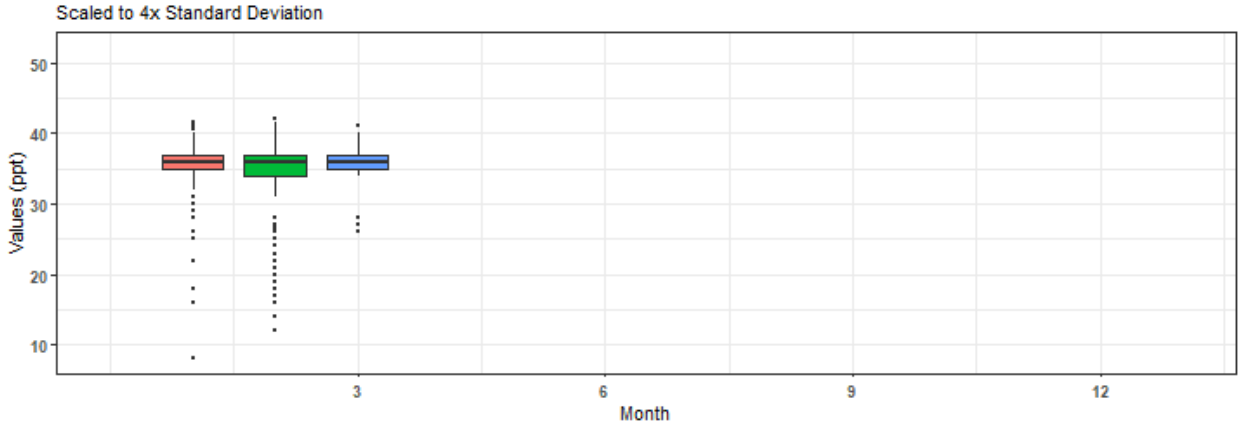


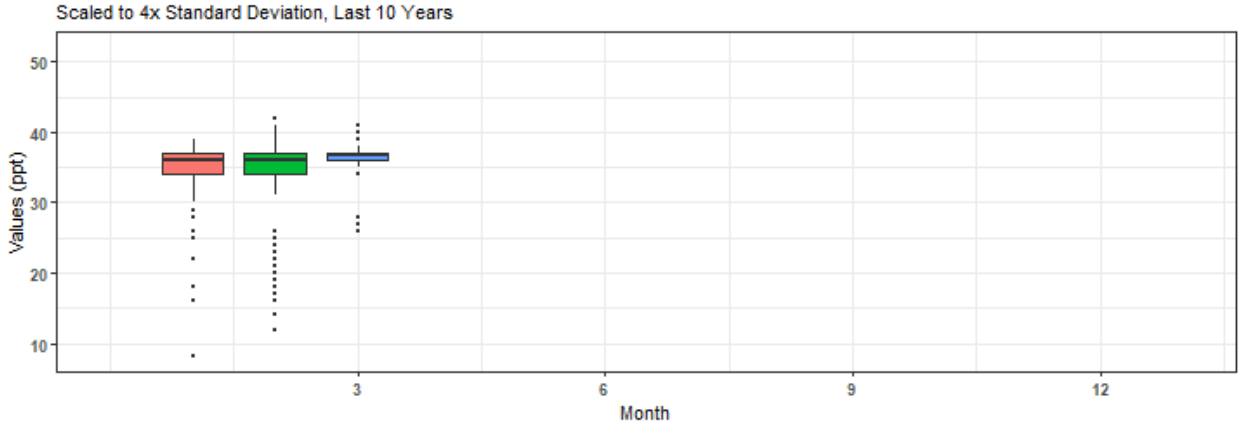


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255534081324000

By Month



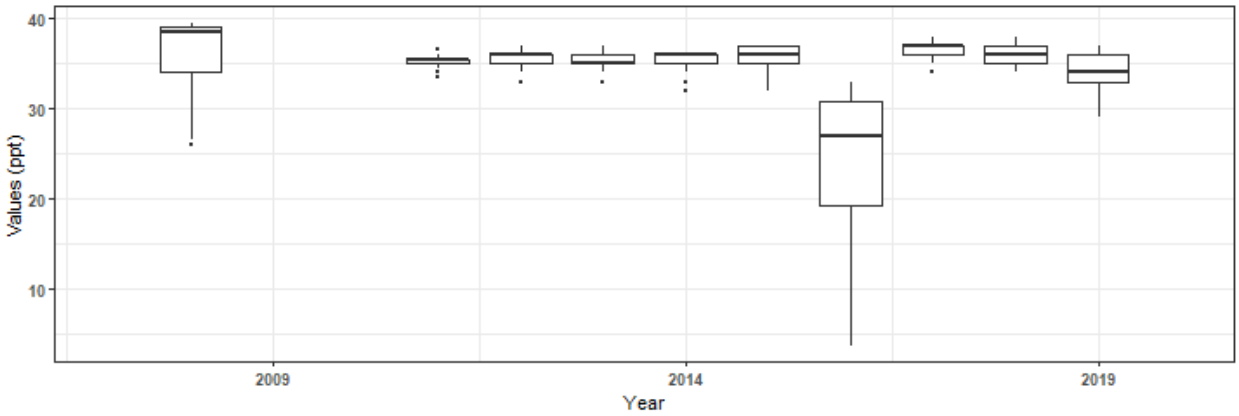




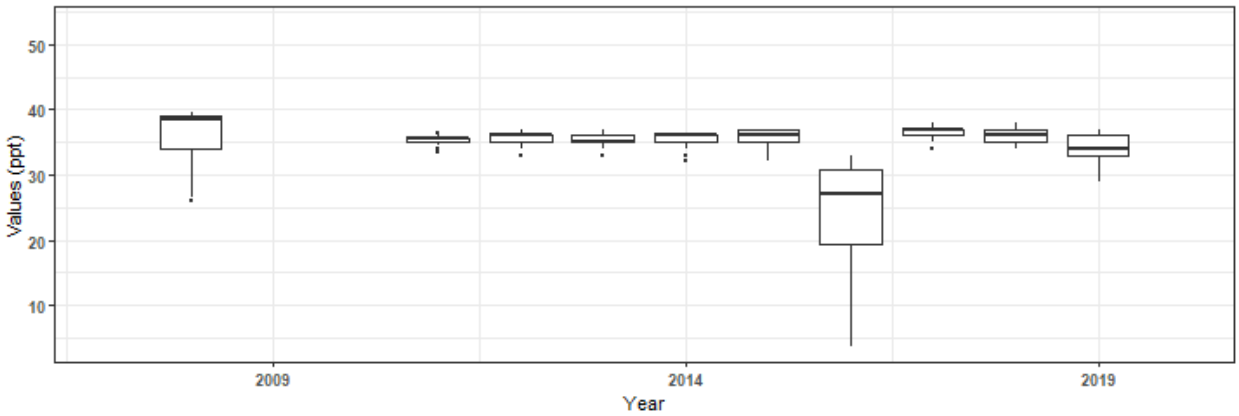
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255654081350200

By Year

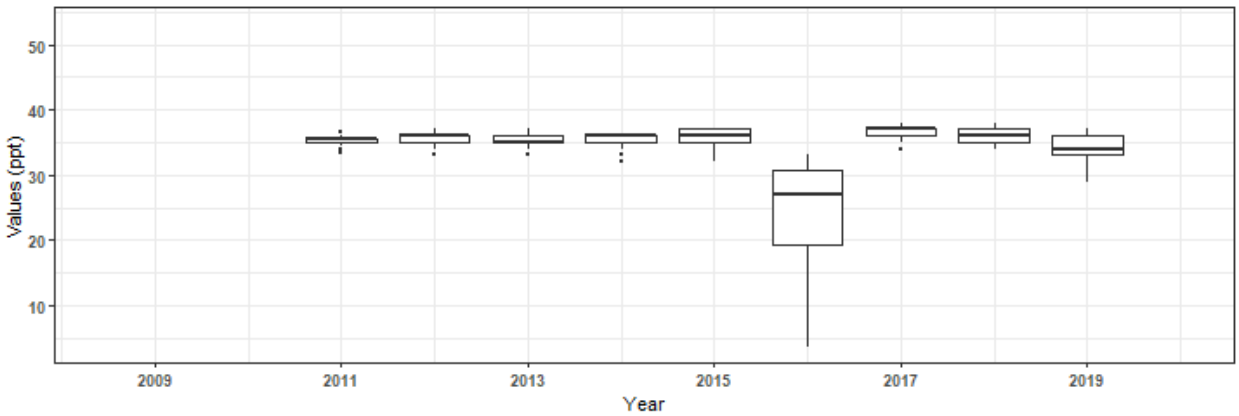
Autoscale



Scaled to 4x Standard Deviation

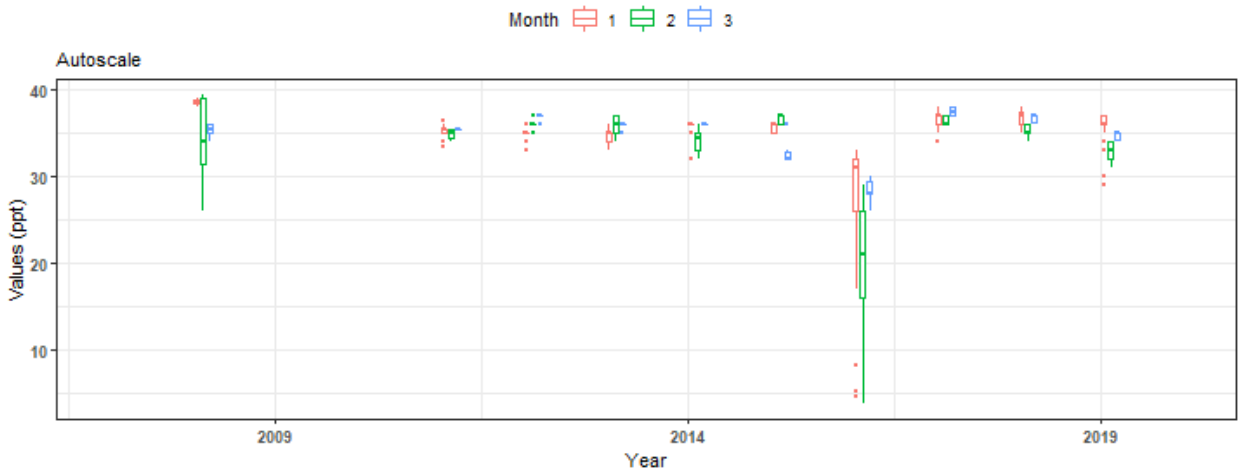


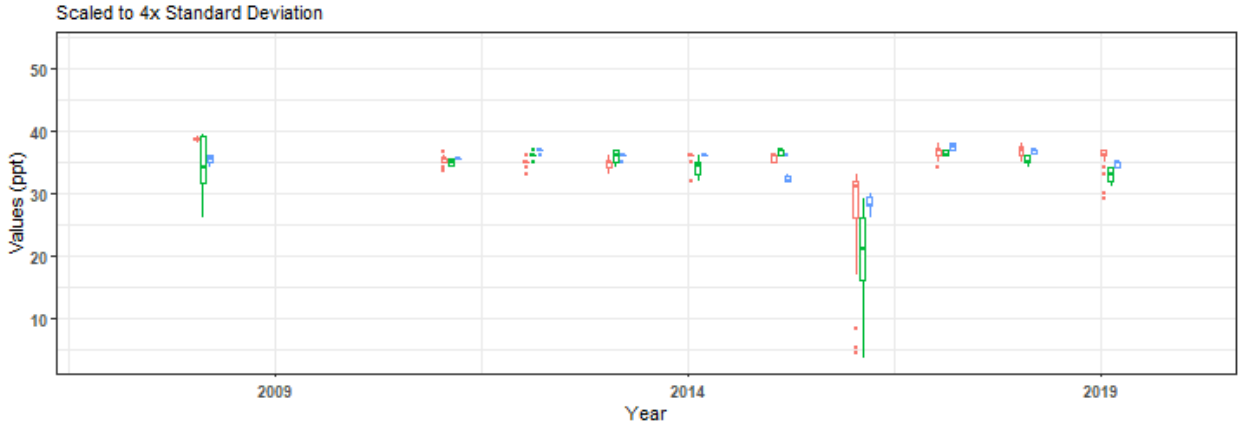
Scaled to 4x Standard Deviation, Last 10 Years

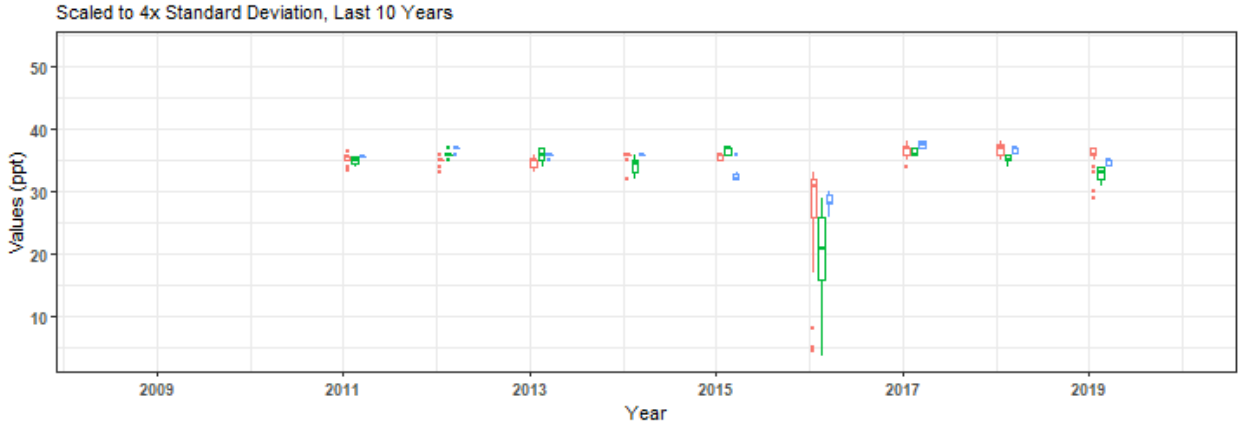


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255654081350200

By Year & Month

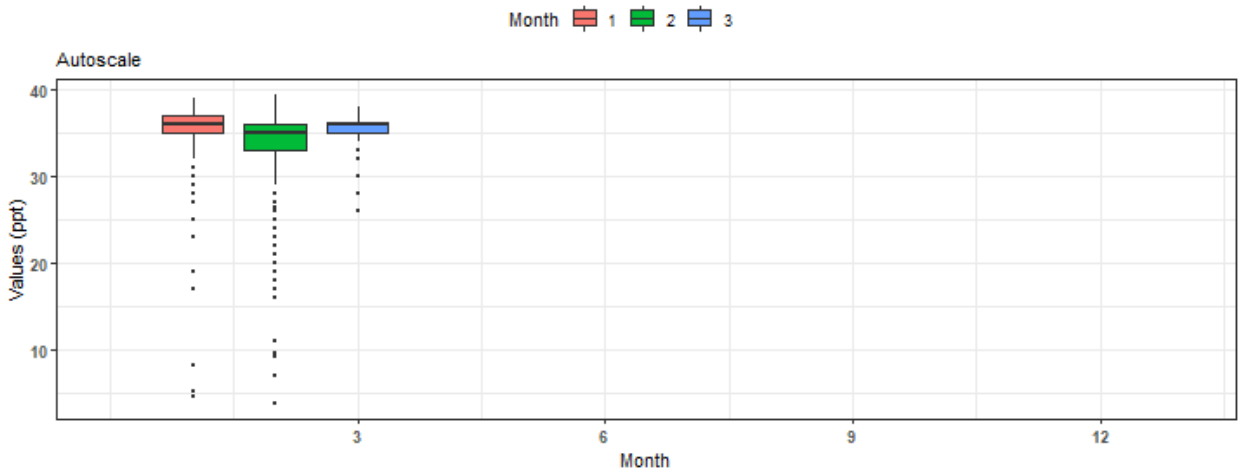


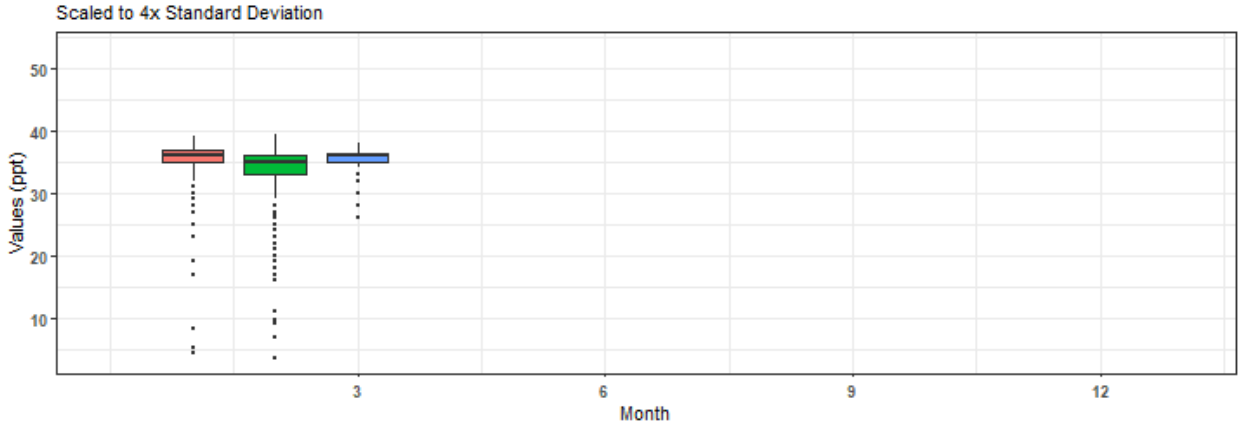


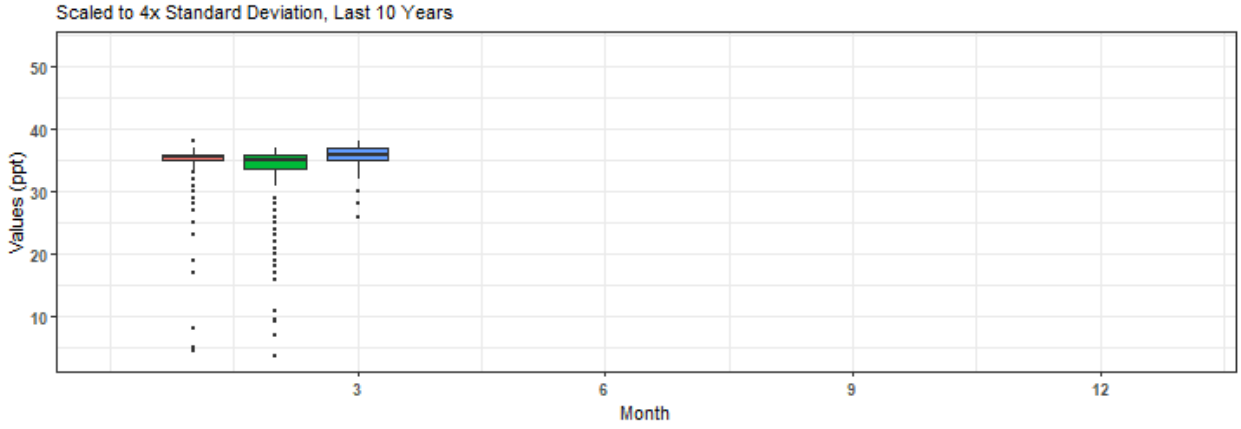


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
7 | National Water Information System | 255654081350200

By Month



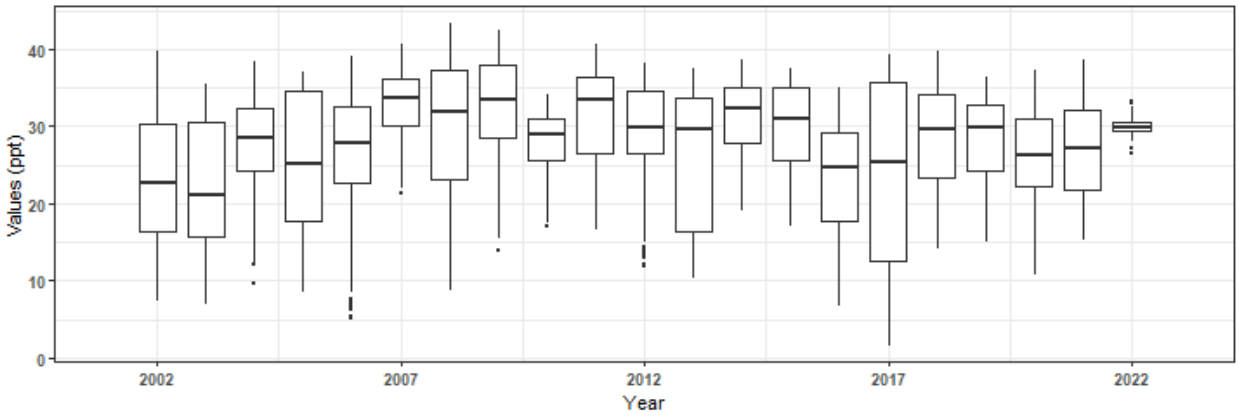




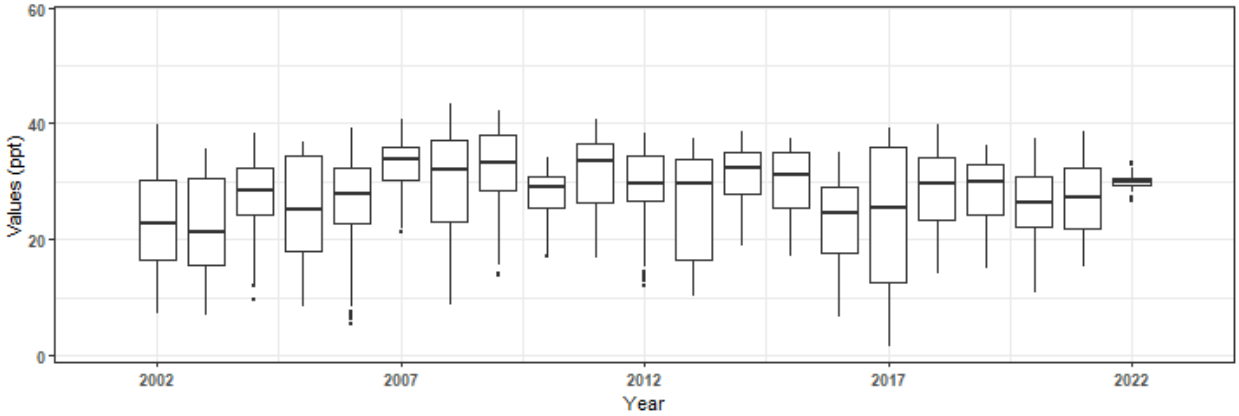
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

By Year

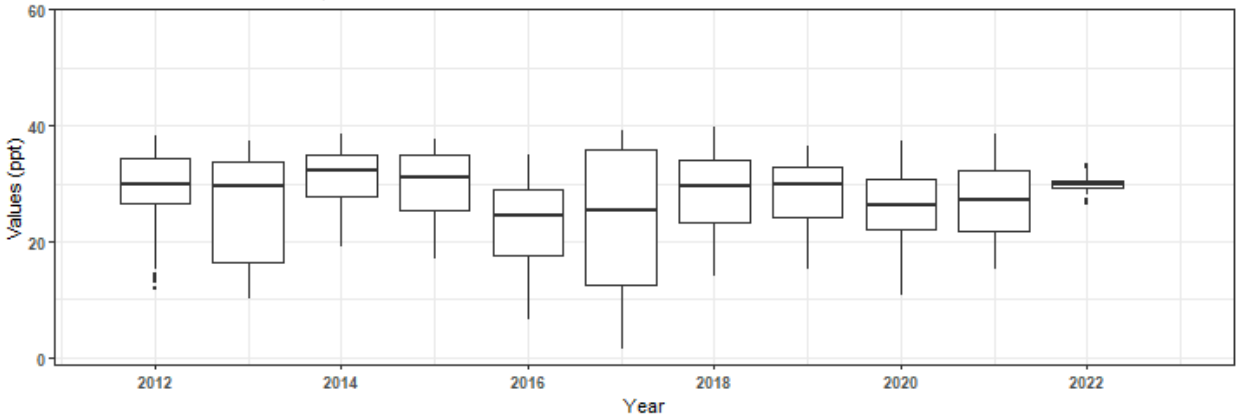
Autoscale



Scaled to 4x Standard Deviation

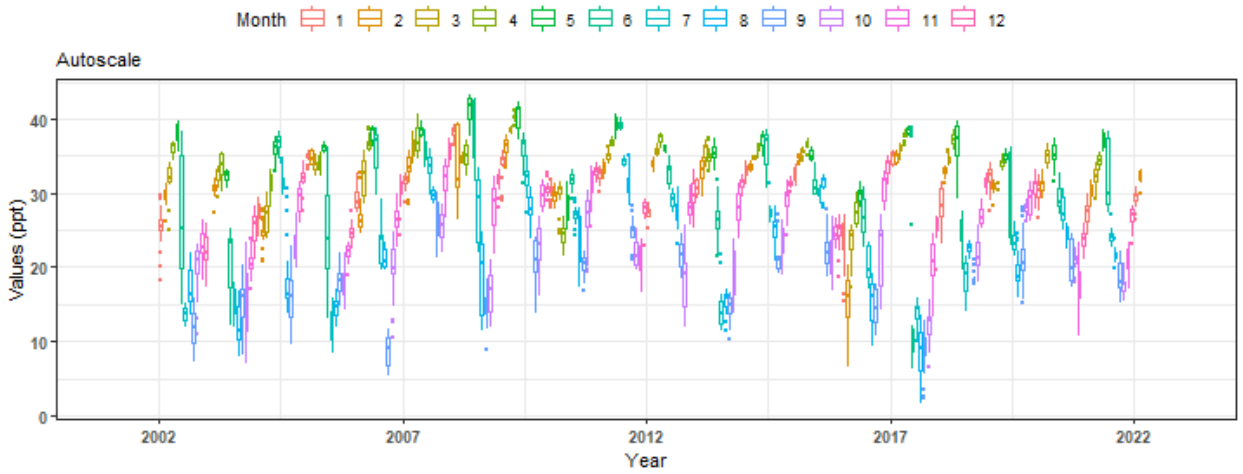


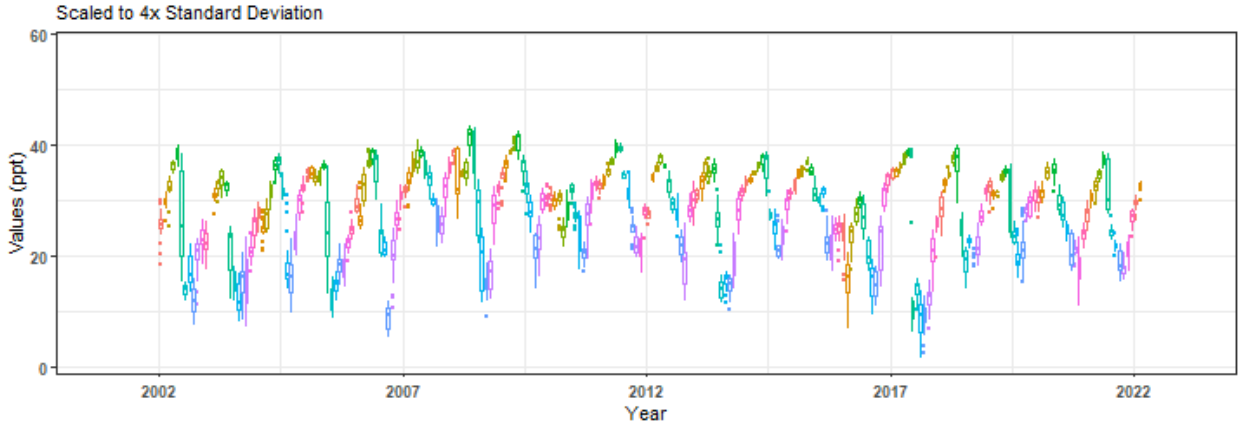
Scaled to 4x Standard Deviation, Last 10 Years

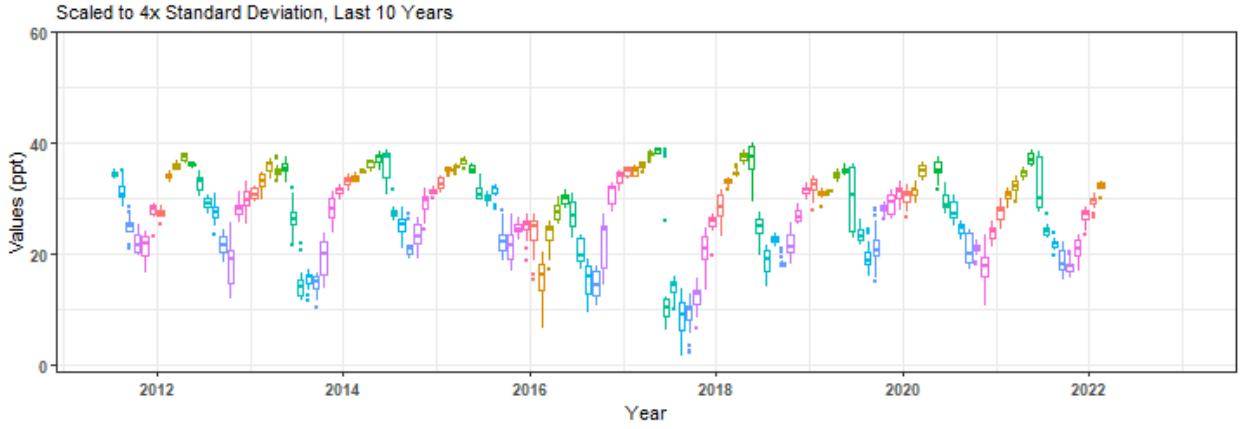


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

By Year & Month

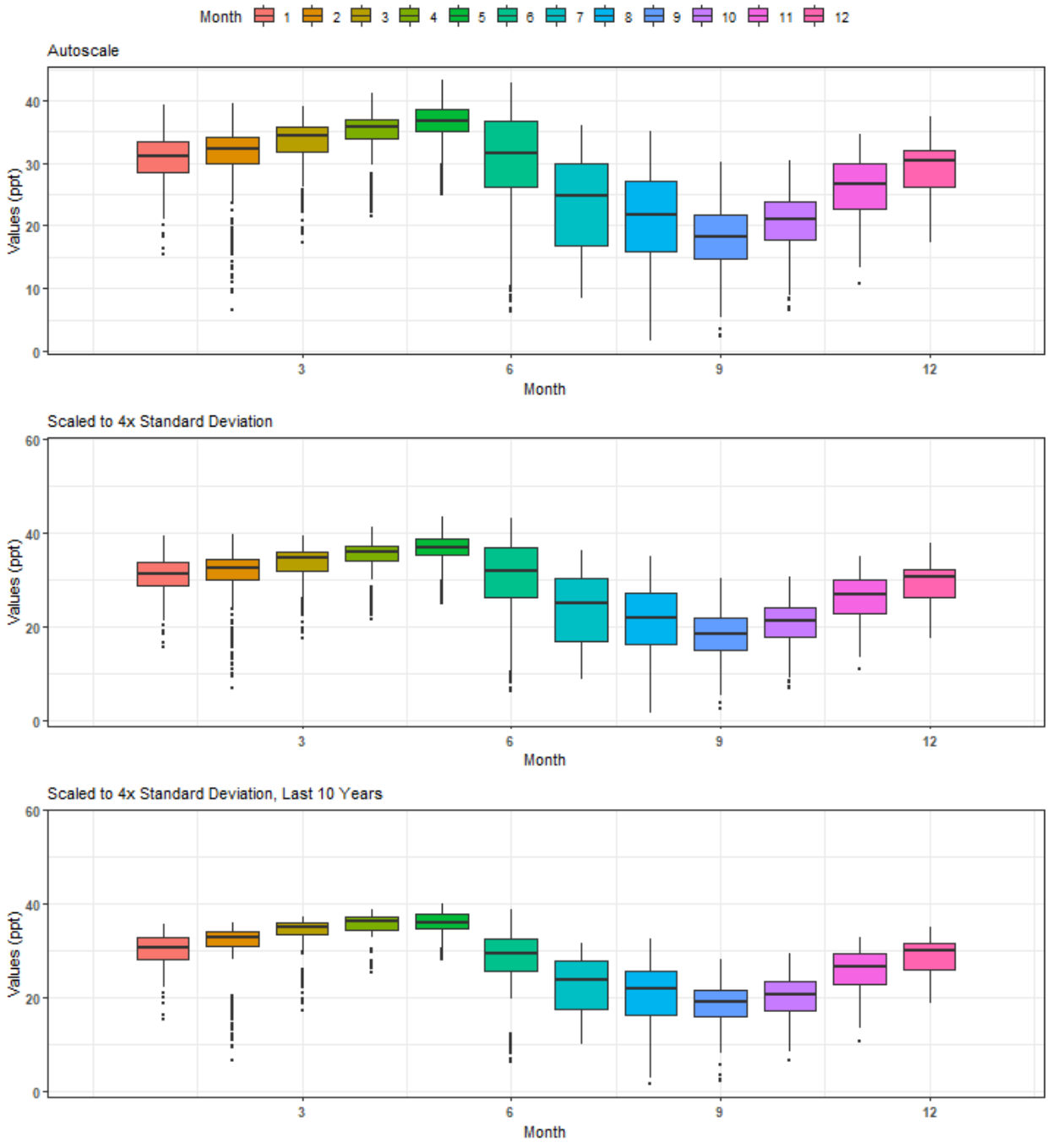






Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

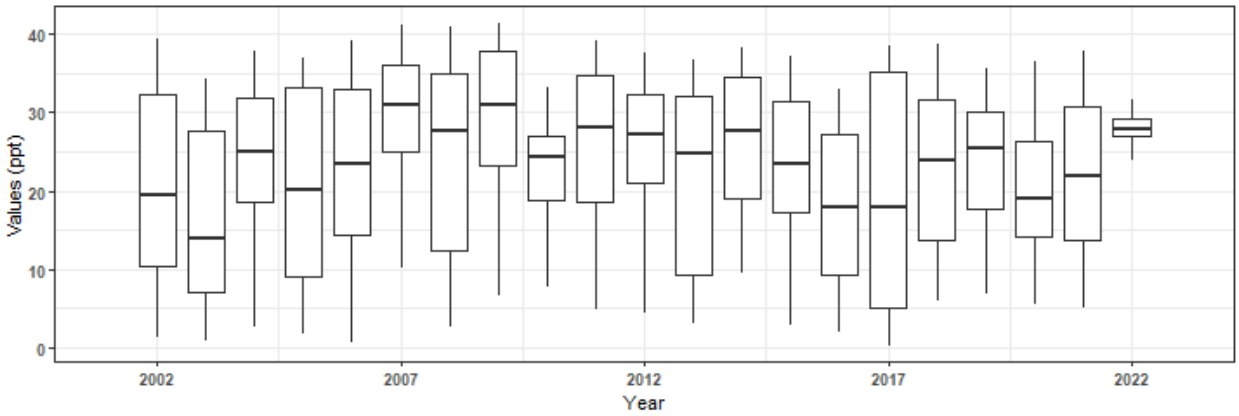
By Month



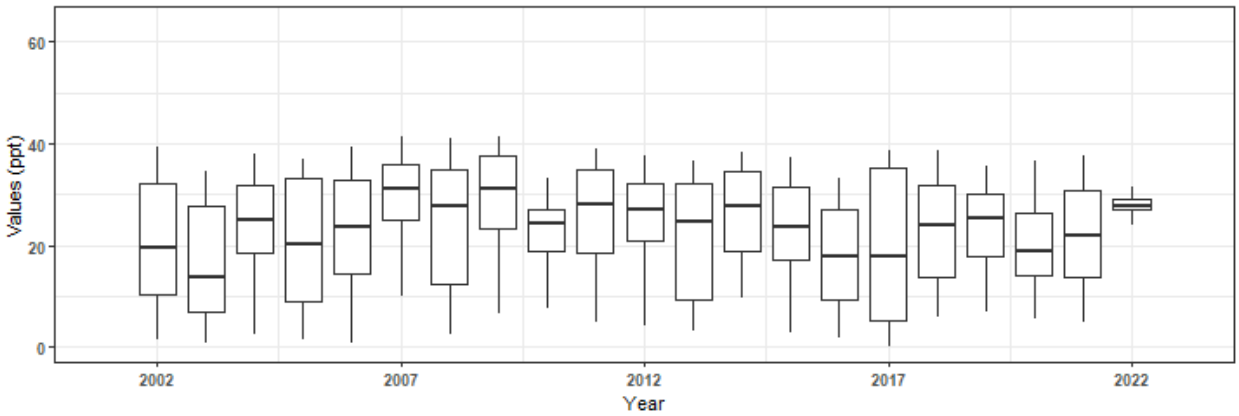
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

By Year

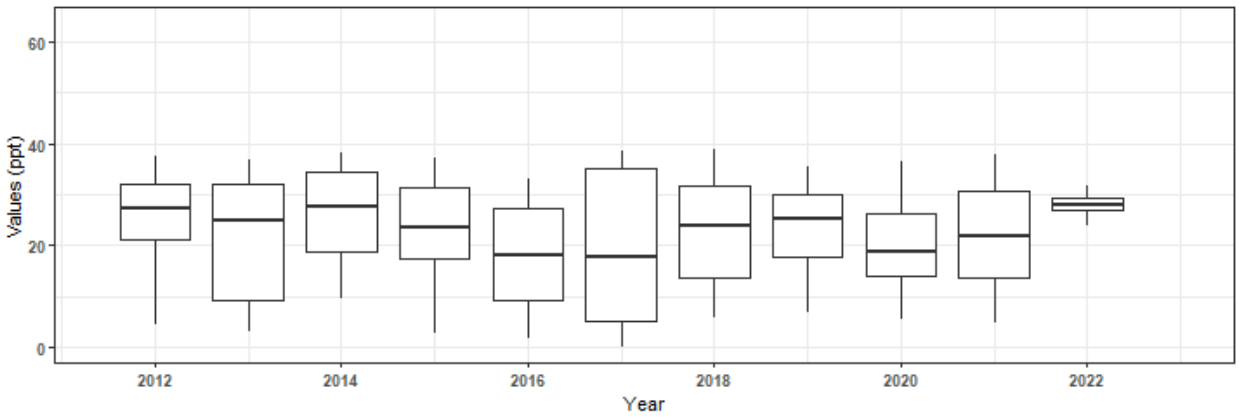
Autoscale



Scaled to 4x Standard Deviation

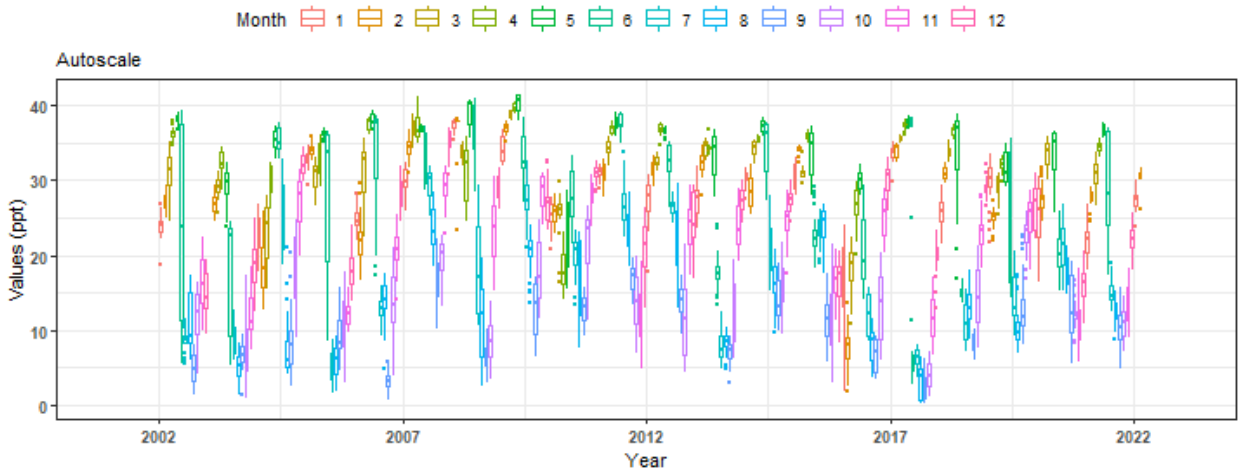


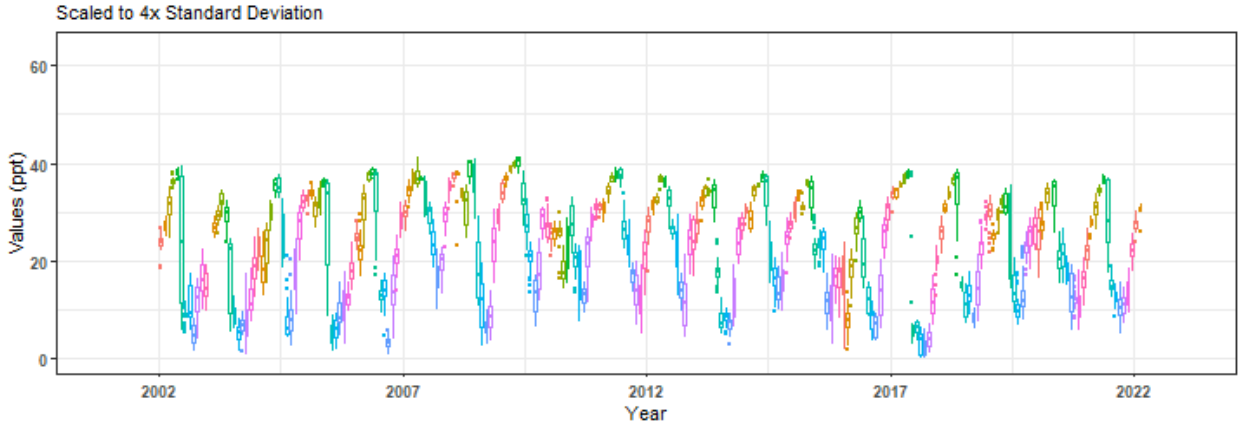
Scaled to 4x Standard Deviation, Last 10 Years

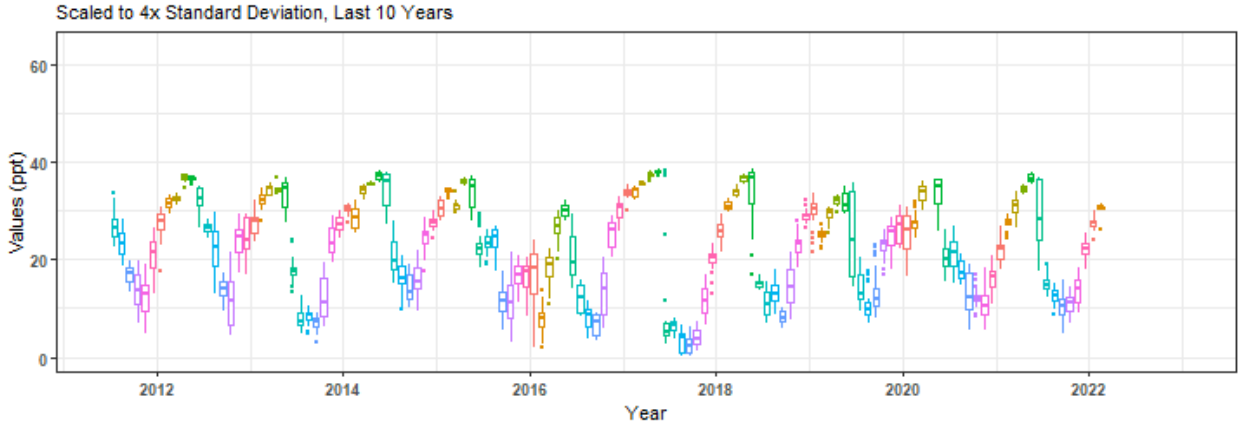


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

By Year & Month

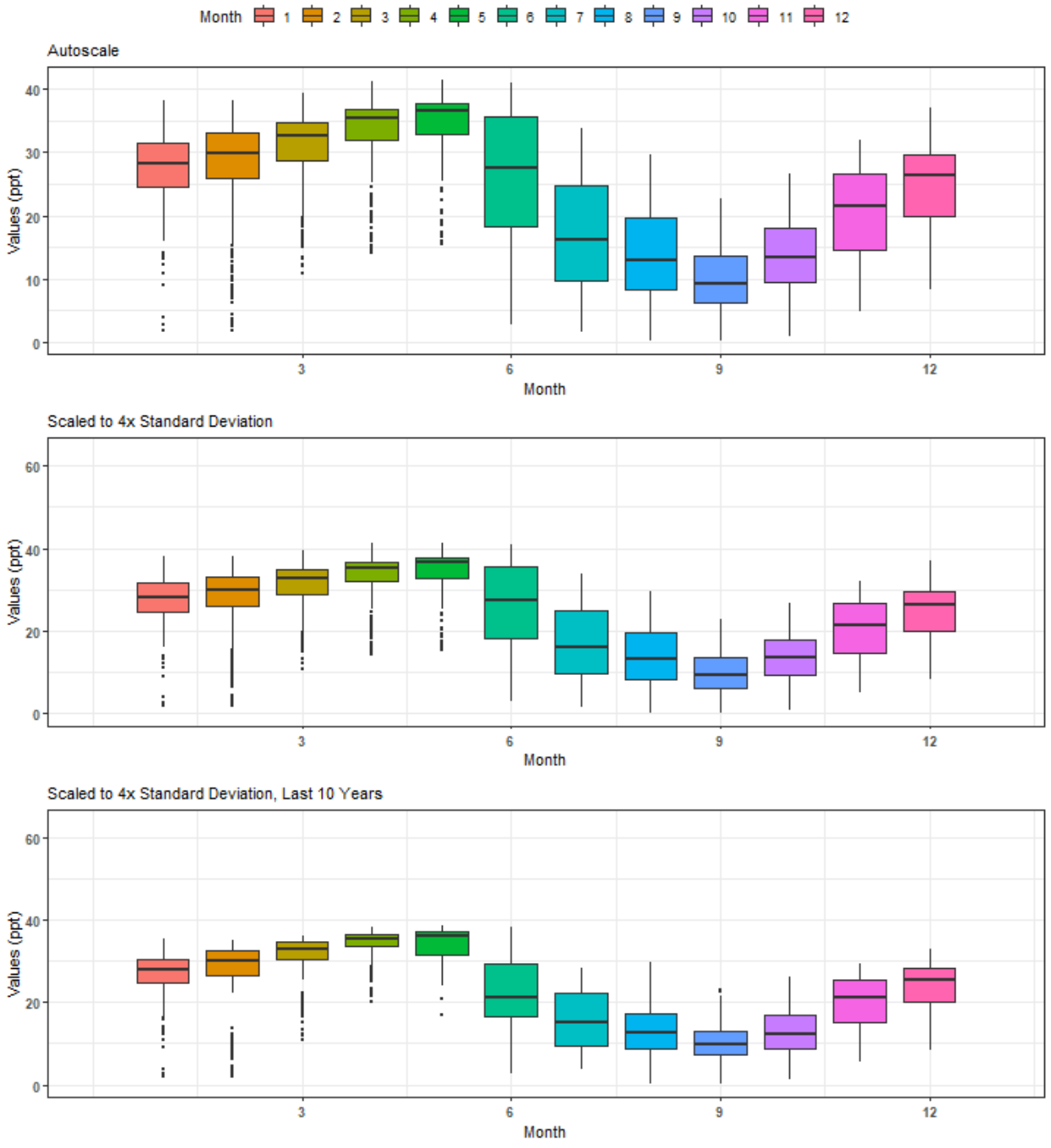






Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

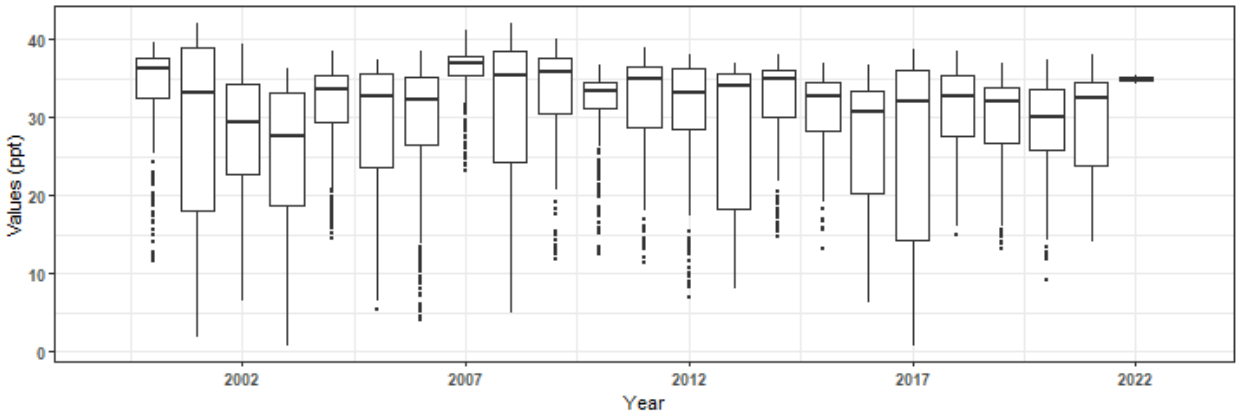
By Month



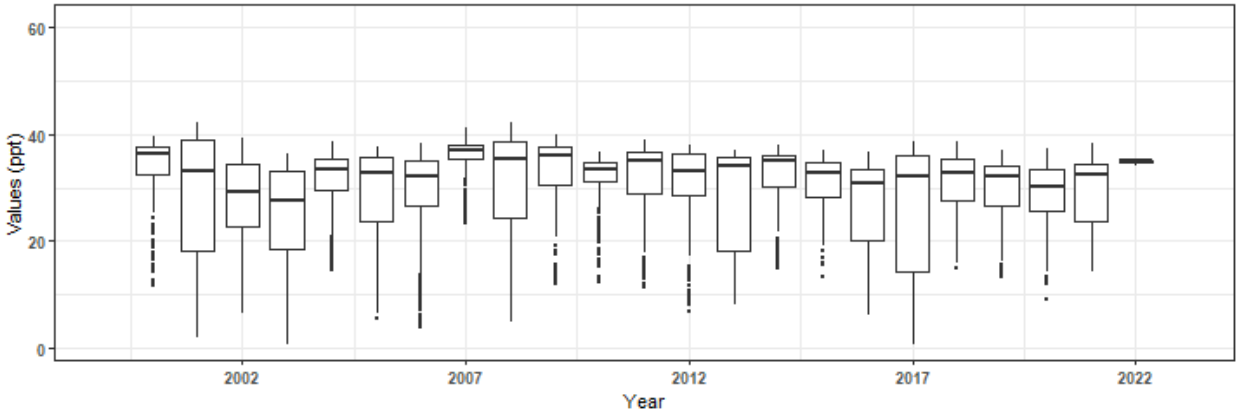
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

By Year

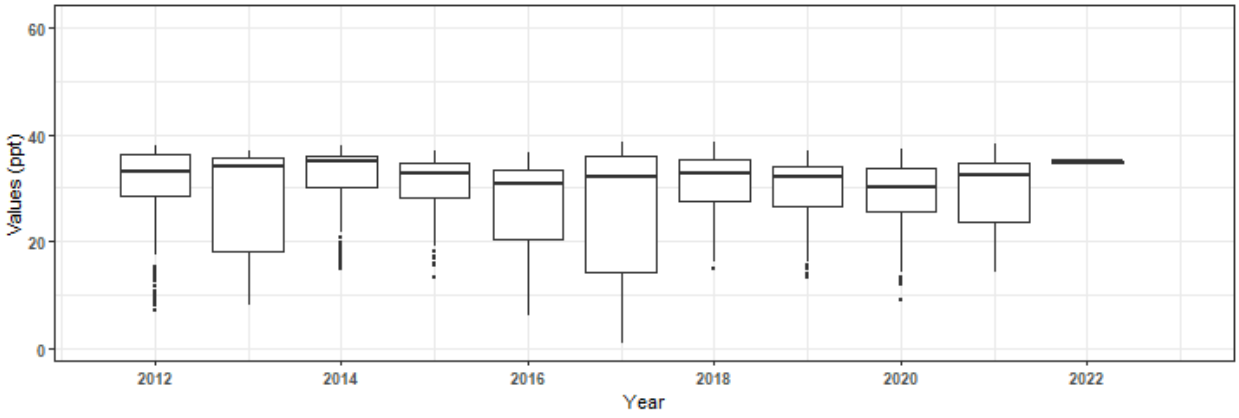
Autoscale



Scaled to 4x Standard Deviation

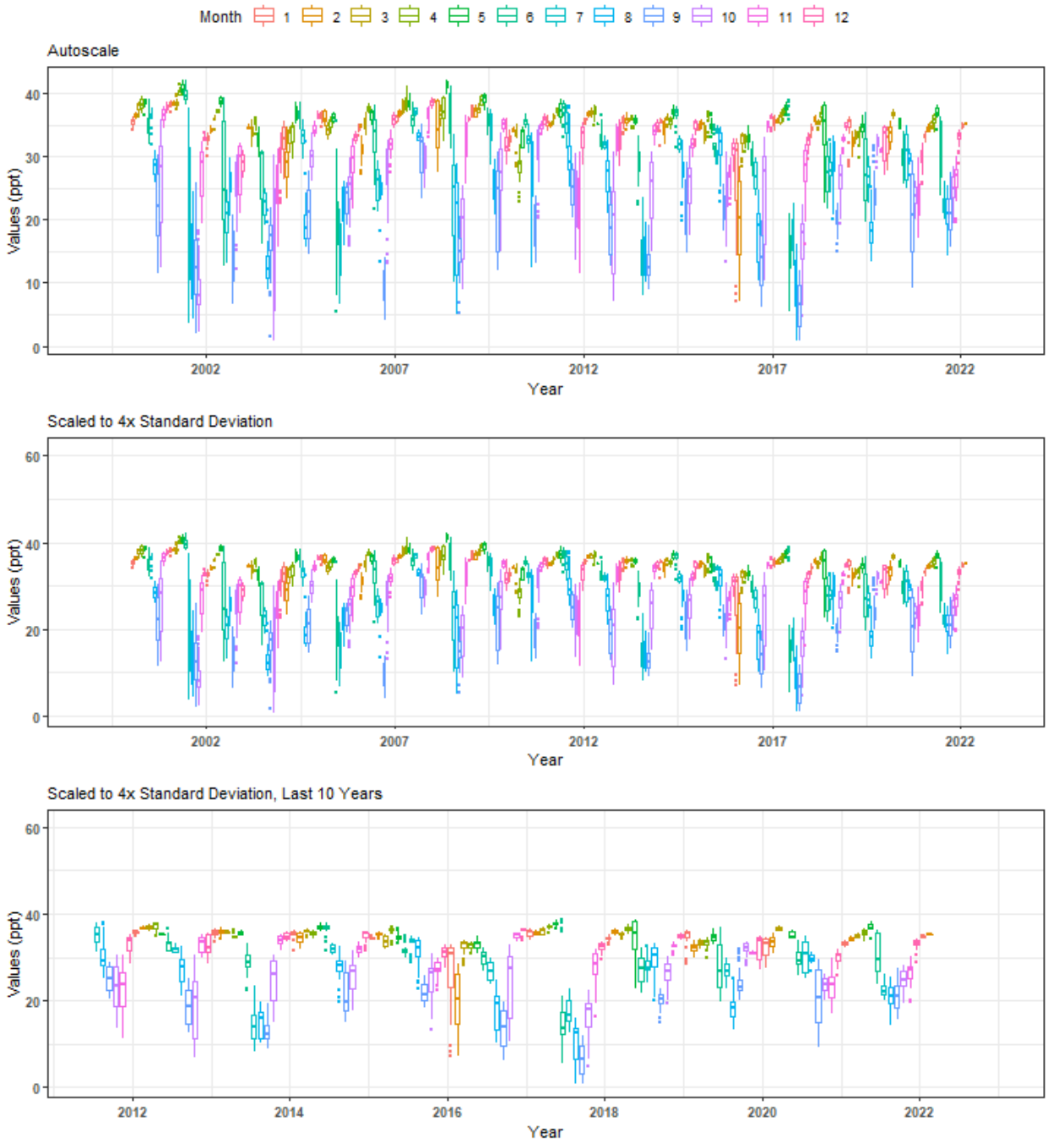


Scaled to 4x Standard Deviation, Last 10 Years



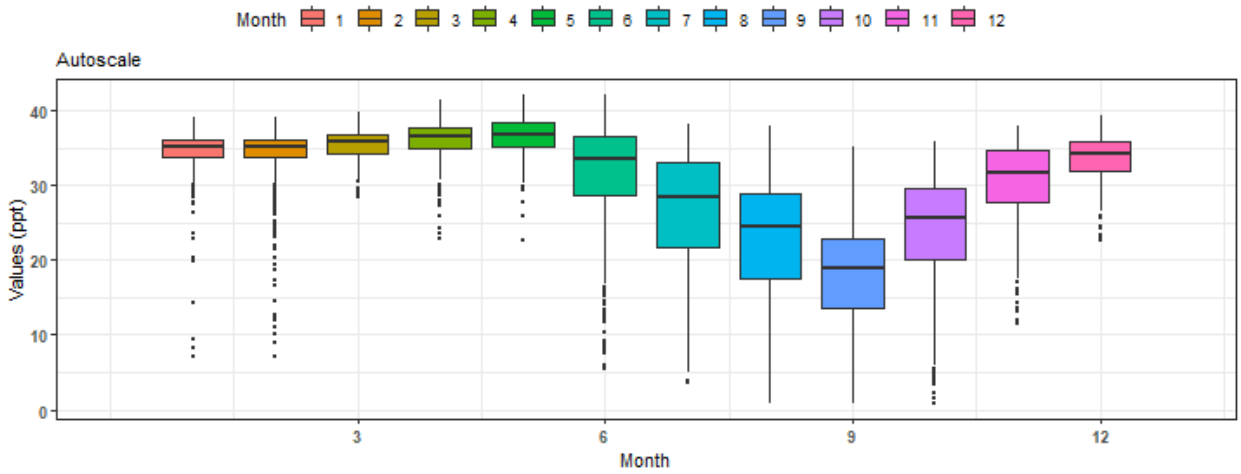
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

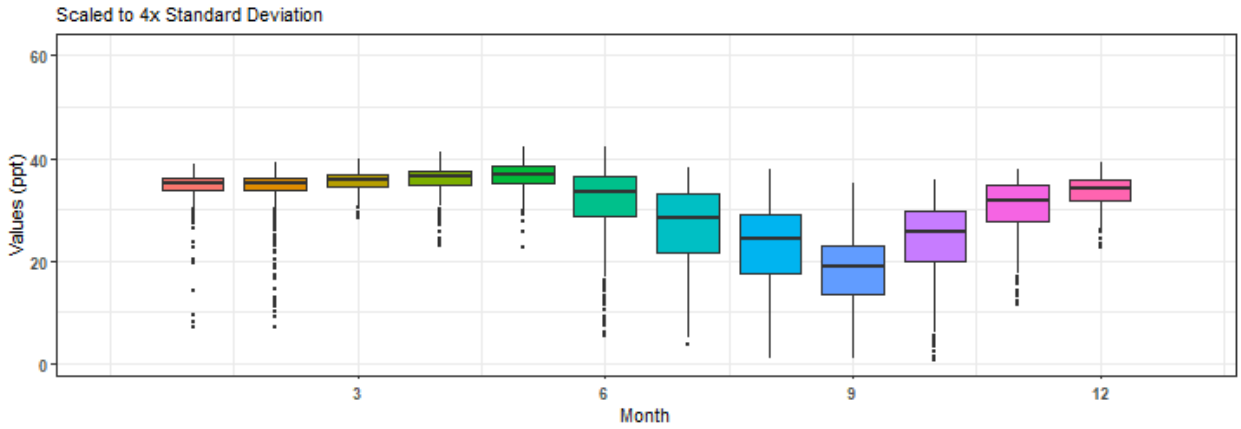
By Year & Month

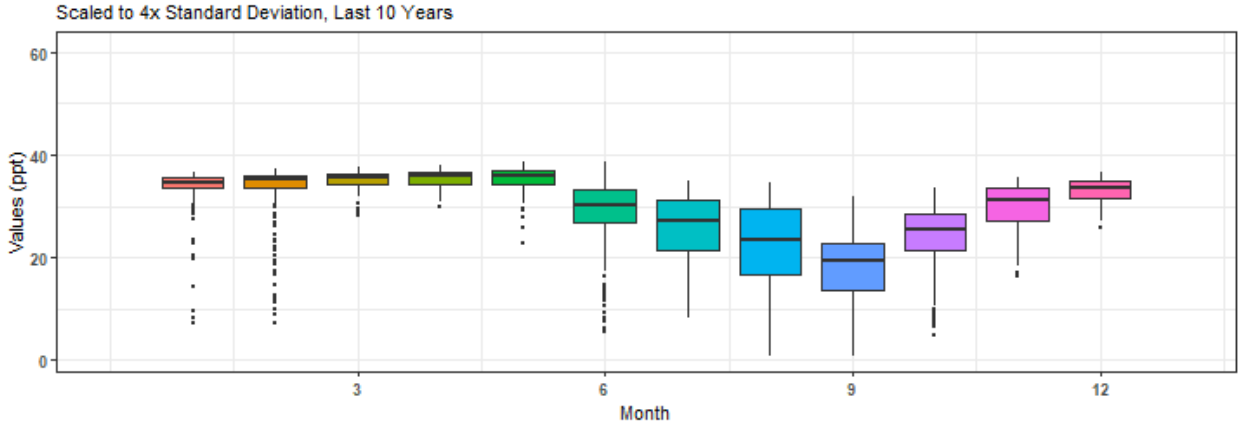


Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

By Month



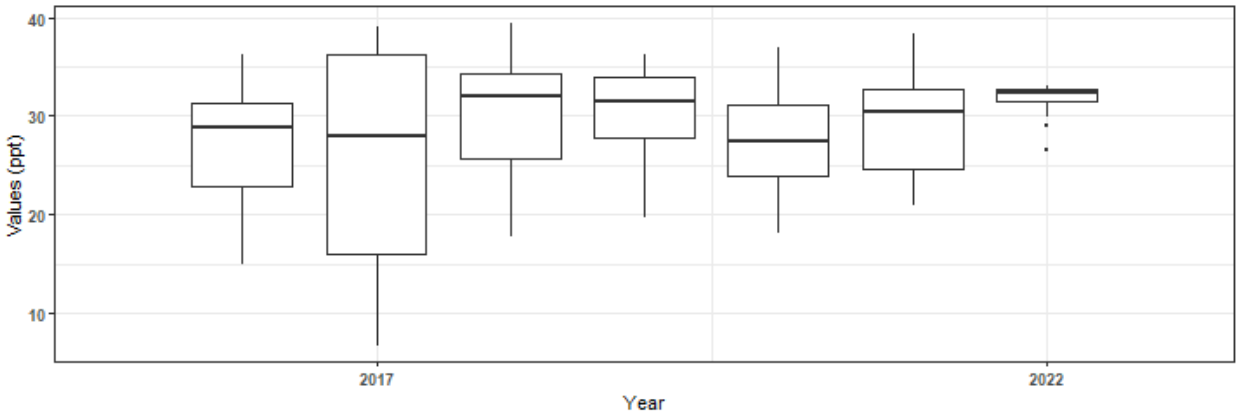




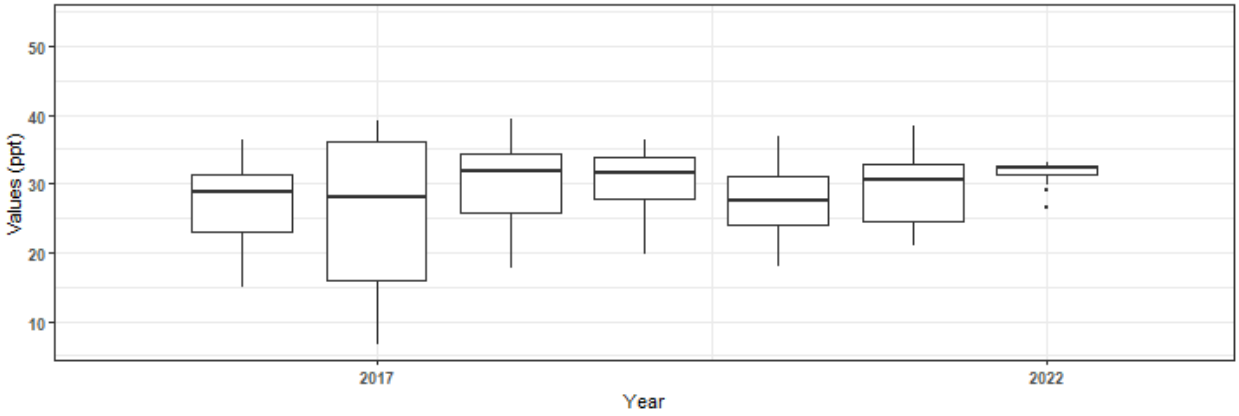
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

By Year

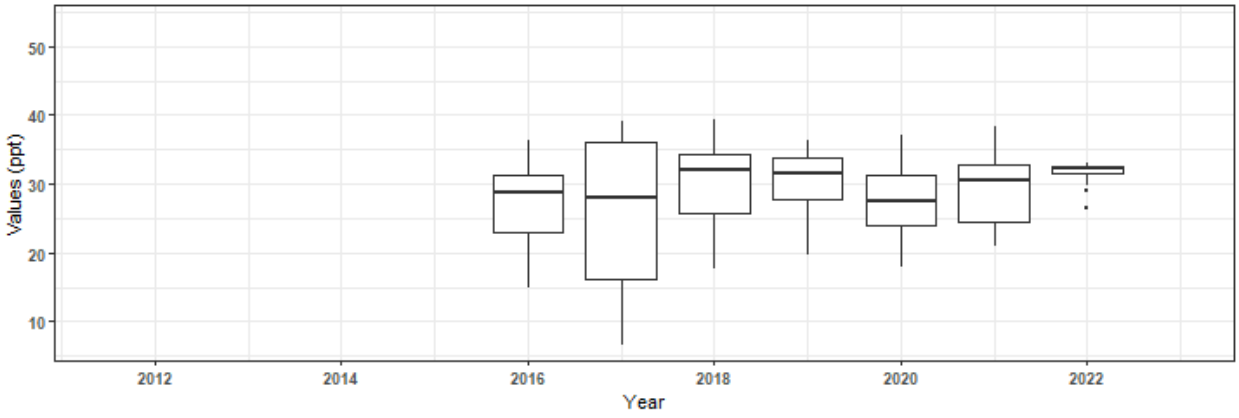
Autoscale



Scaled to 4x Standard Deviation

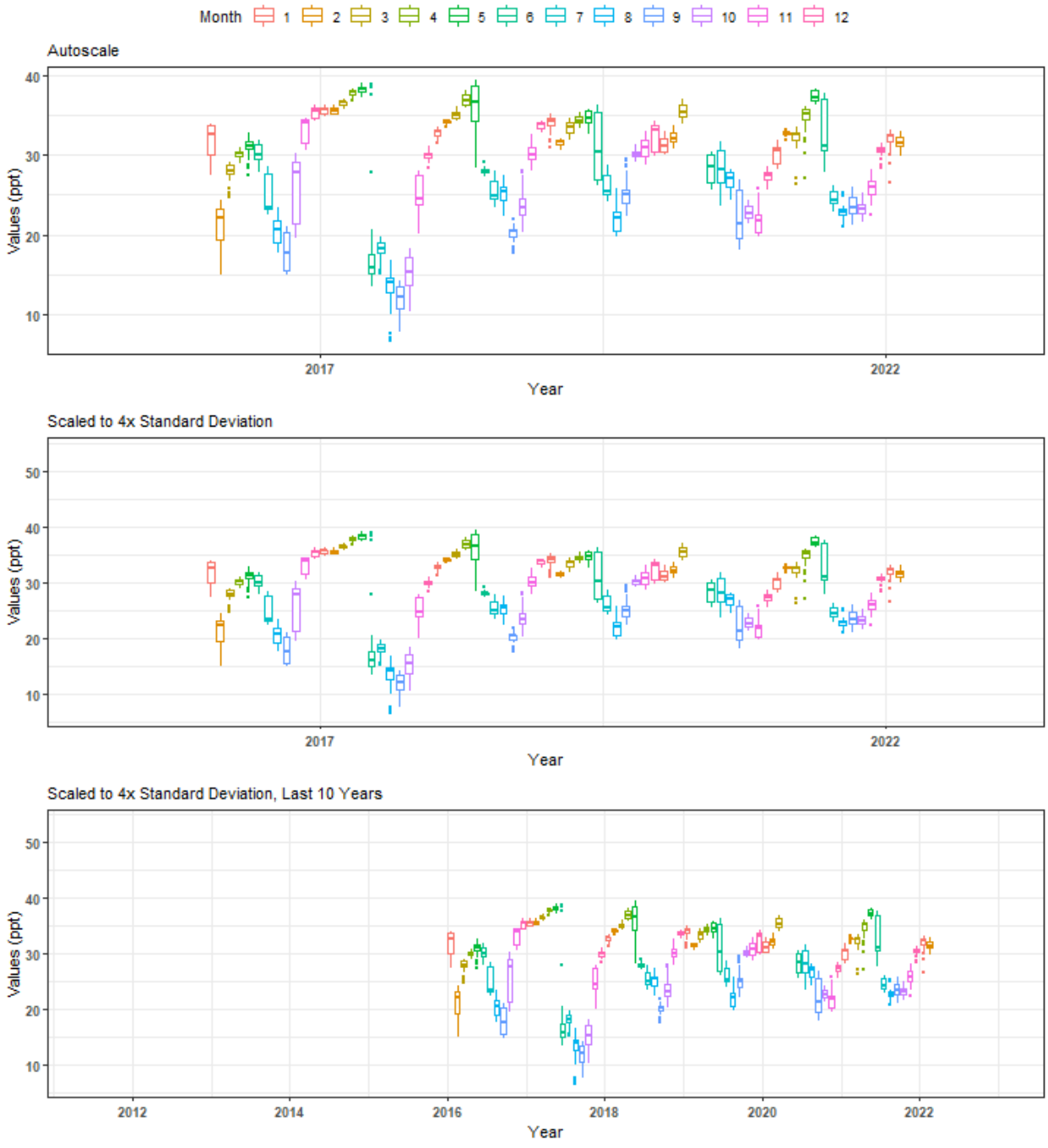


Scaled to 4x Standard Deviation, Last 10 Years



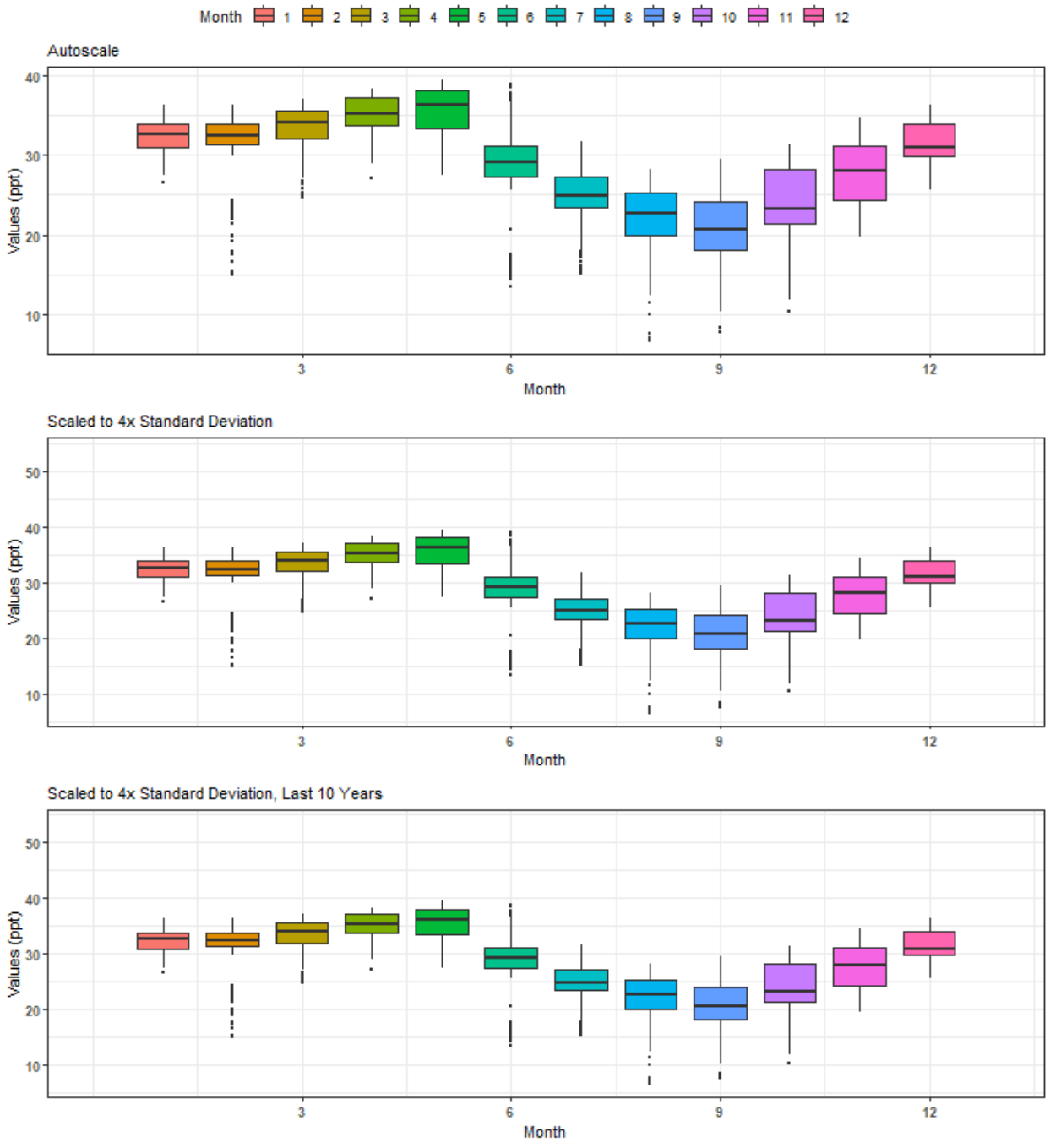
Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

By Year & Month



Summary Box Plots for Cape Romano-Ten Thousand Islands Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

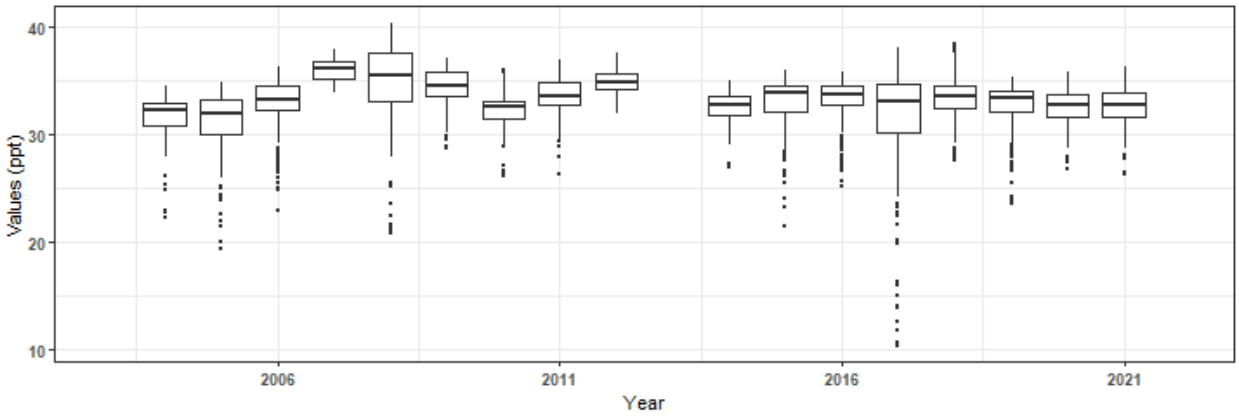
By Month



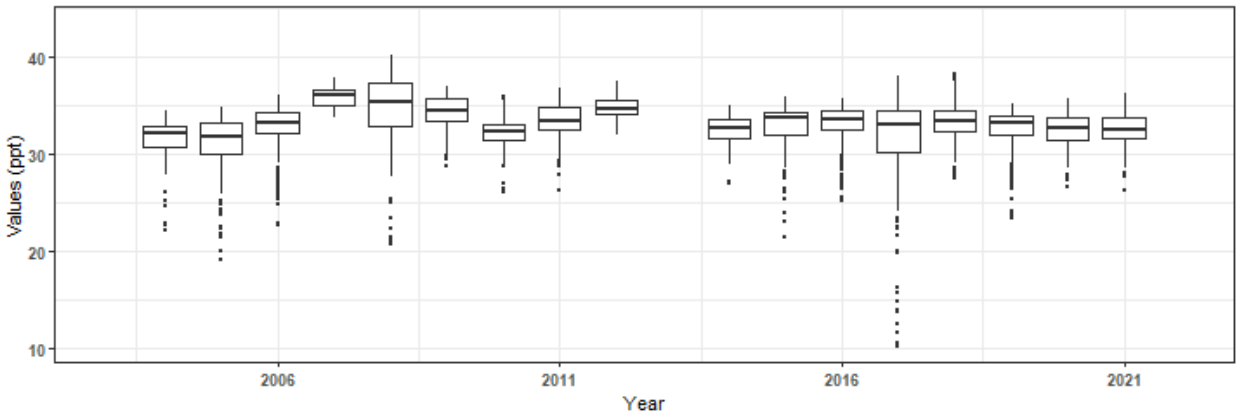
Summary Box Plots for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB02

By Year

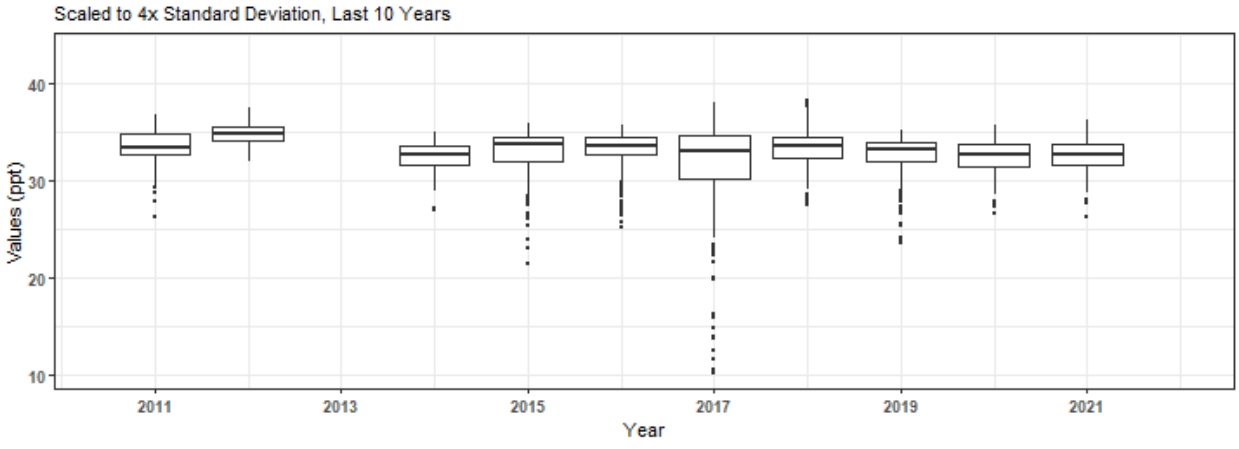
Autoscale



Scaled to 4x Standard Deviation

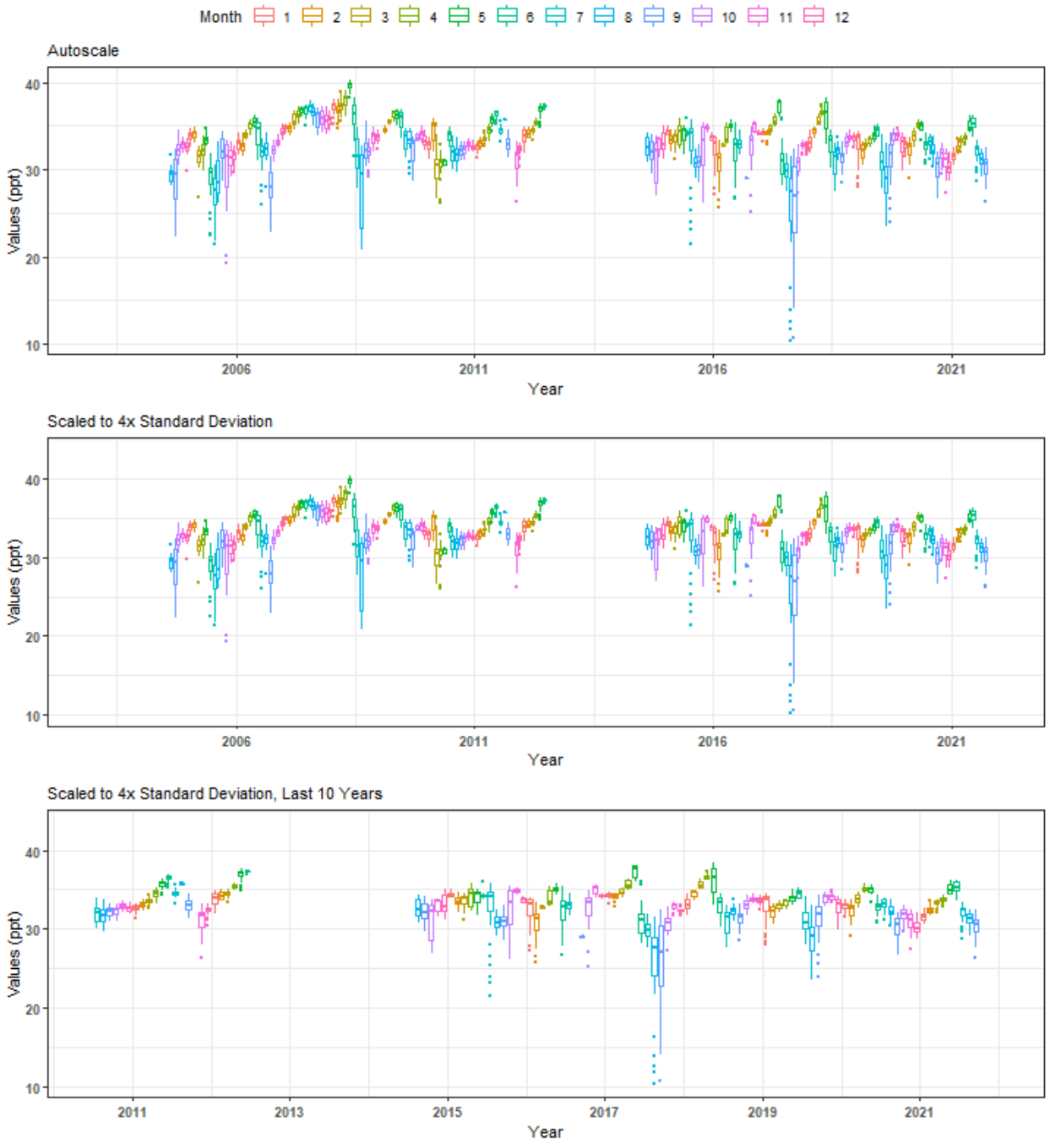


Scaled to 4x Standard Deviation, Last 10 Years



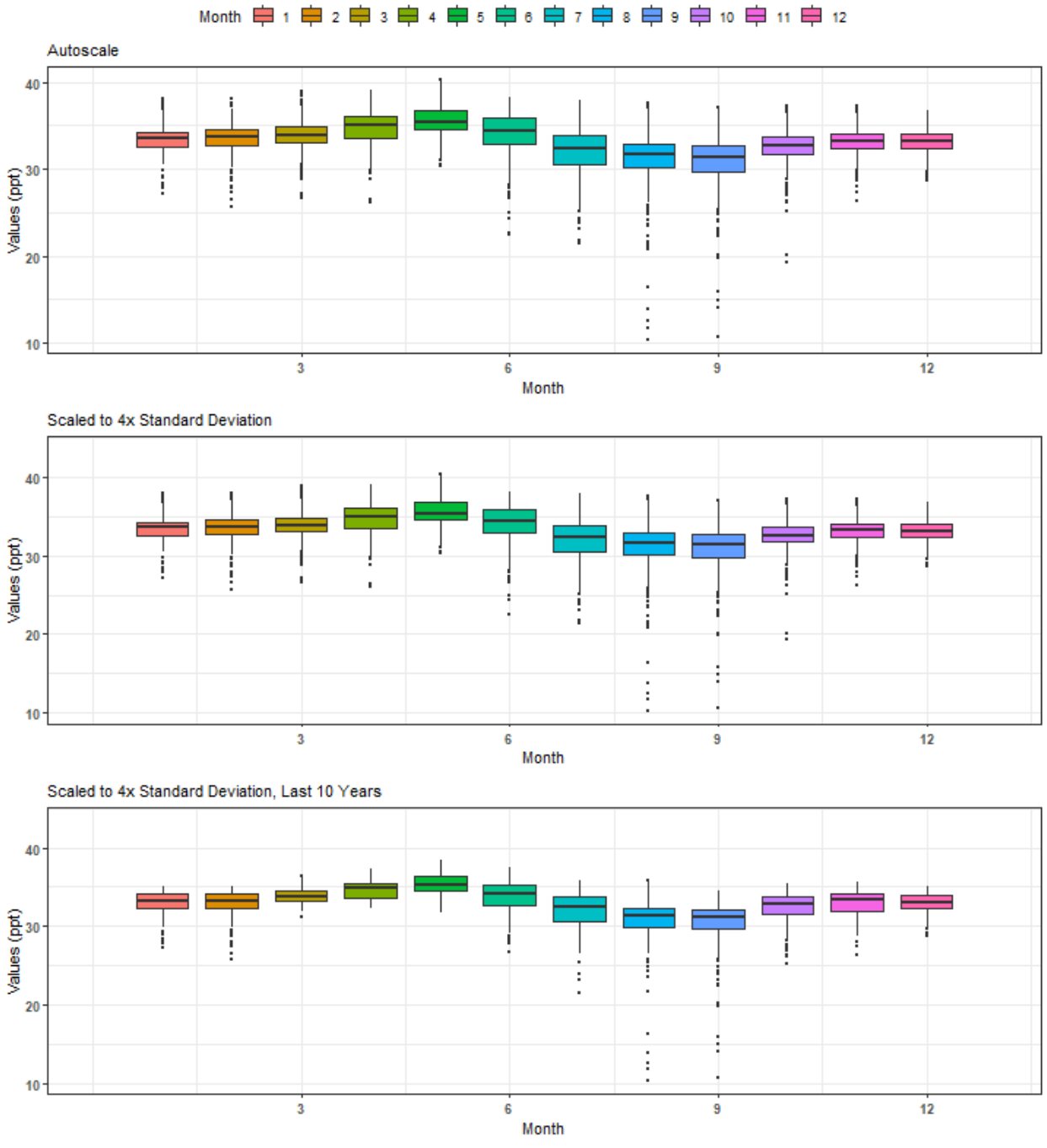
Summary Box Plots for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB02

By Year & Month



Summary Box Plots for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB02

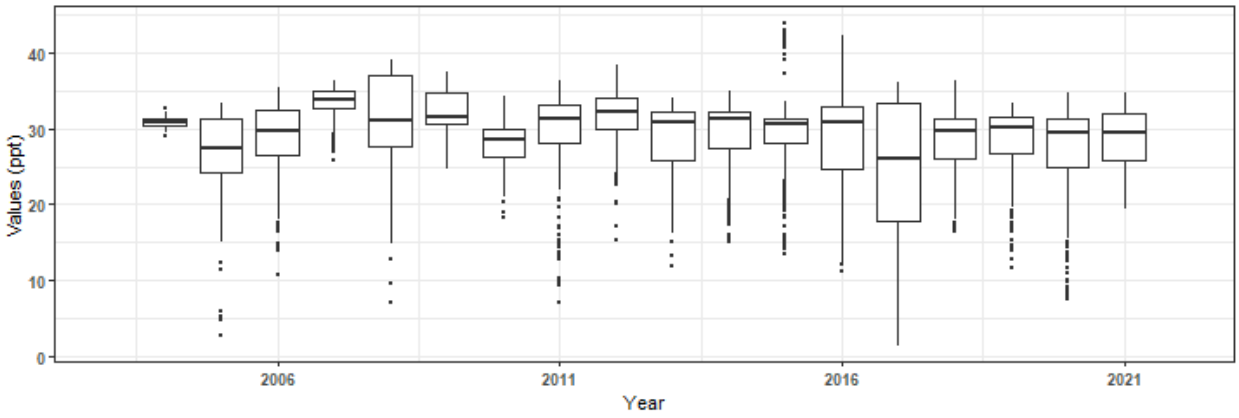
By Month



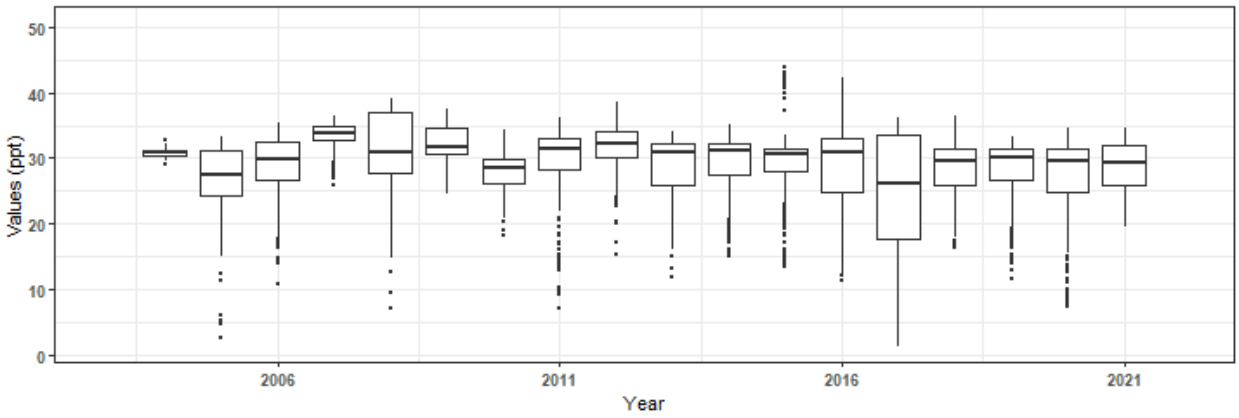
Summary Box Plots for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB03

By Year

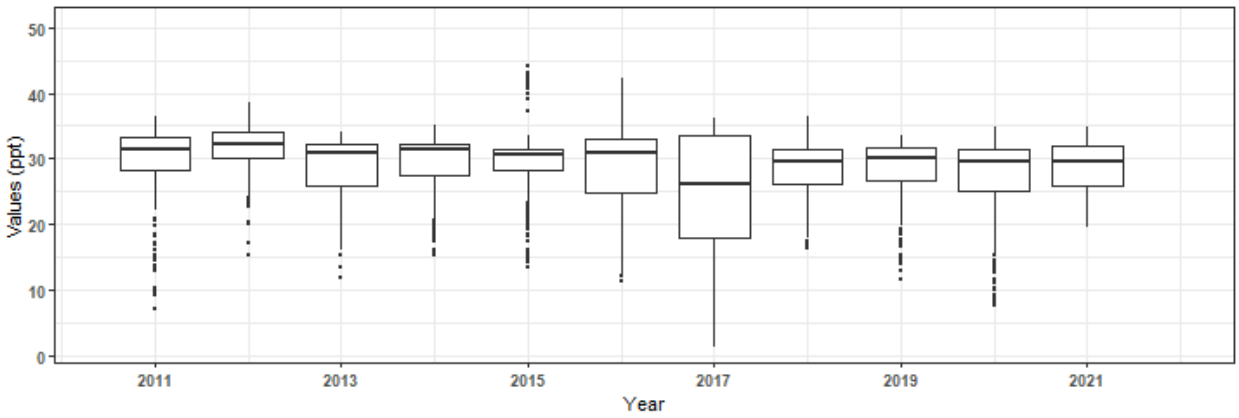
Autoscale



Scaled to 4x Standard Deviation

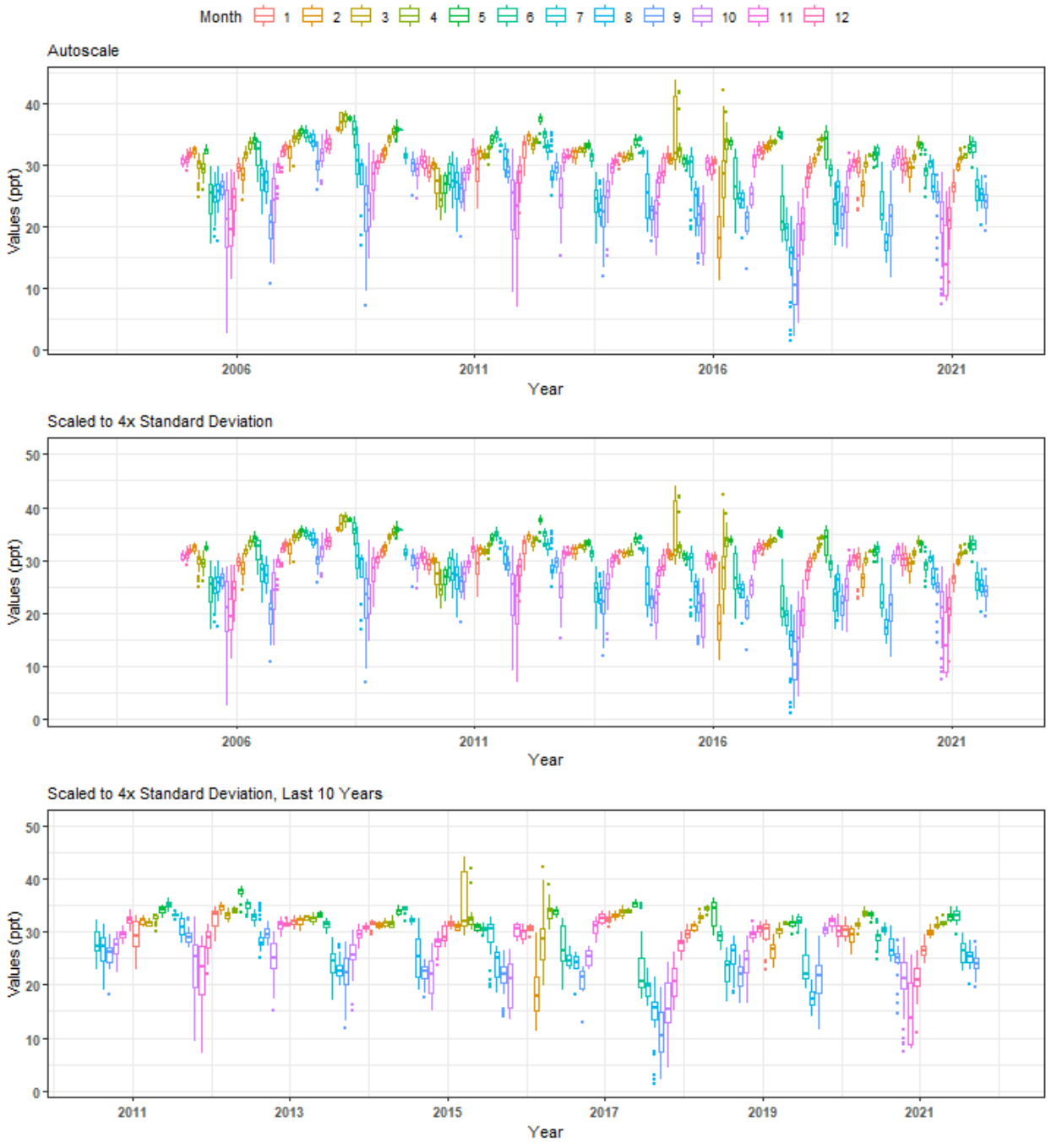


Scaled to 4x Standard Deviation, Last 10 Years



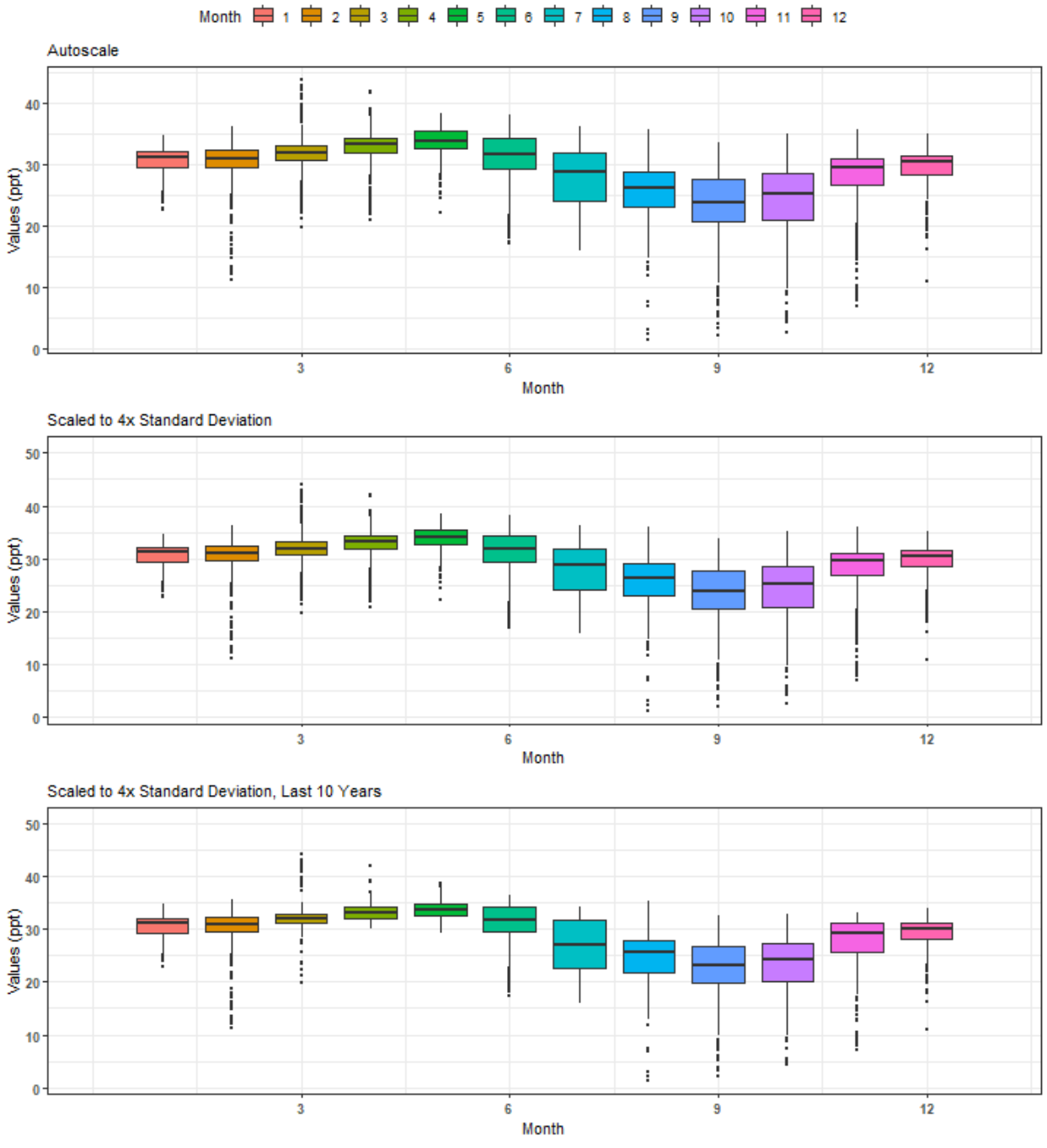
Summary Box Plots for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB03

By Year & Month



Summary Box Plots for Estero Bay Aquatic Preserve
474 | Estero Bay Aquatic Preserve Continuous Water Quality Monitoring | EB03

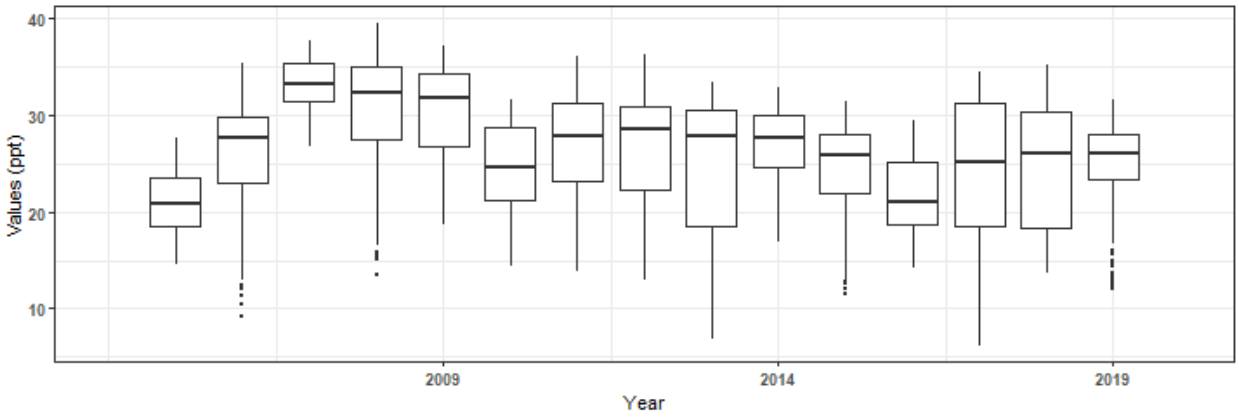
By Month



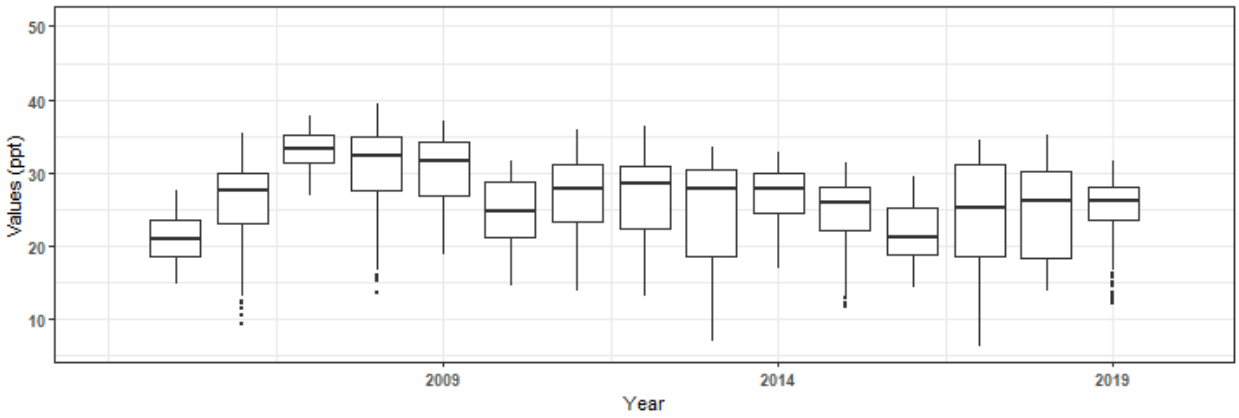
Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP1A

By Year

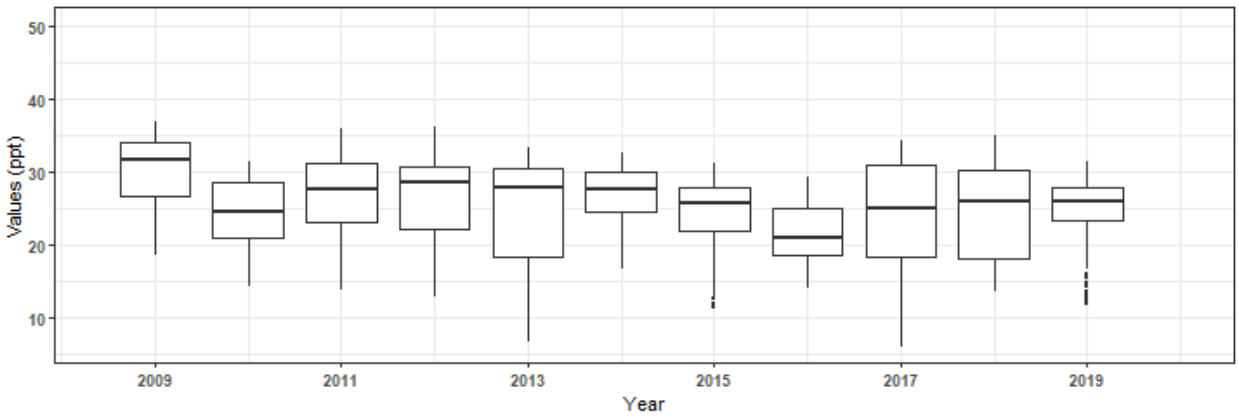
Autoscale



Scaled to 4x Standard Deviation

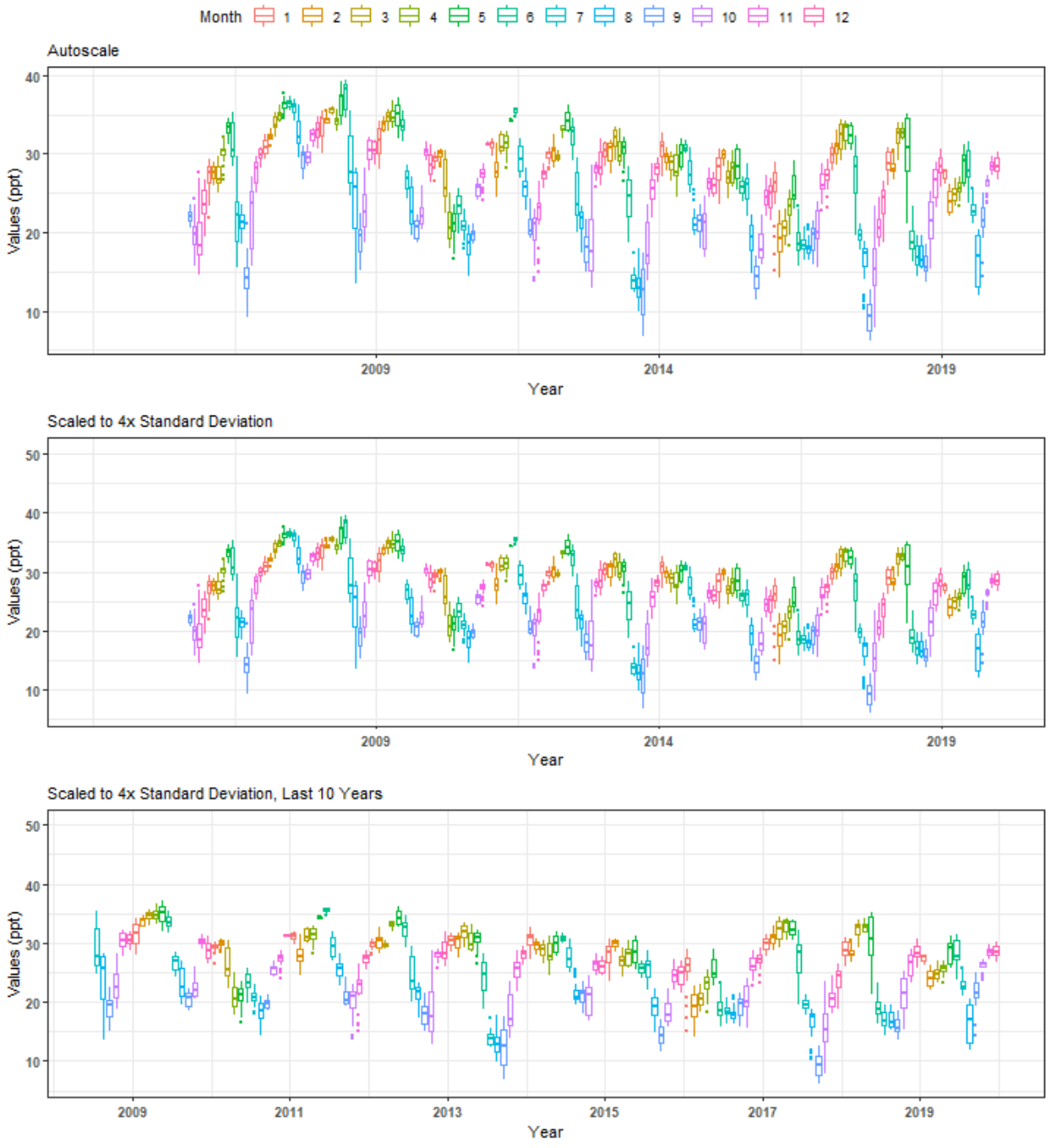


Scaled to 4x Standard Deviation, Last 10 Years



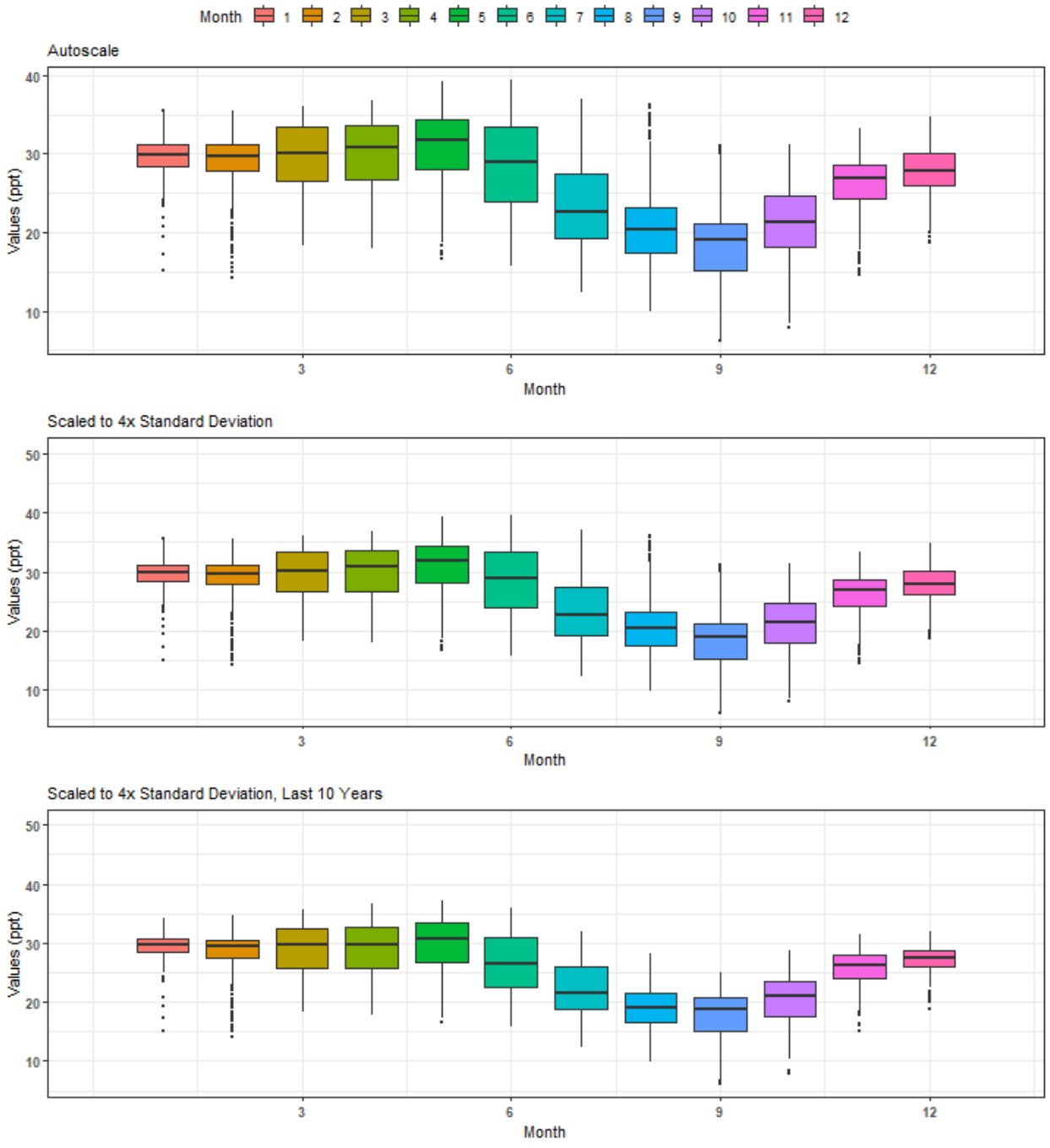
Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP1A

By Year & Month



Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP1A

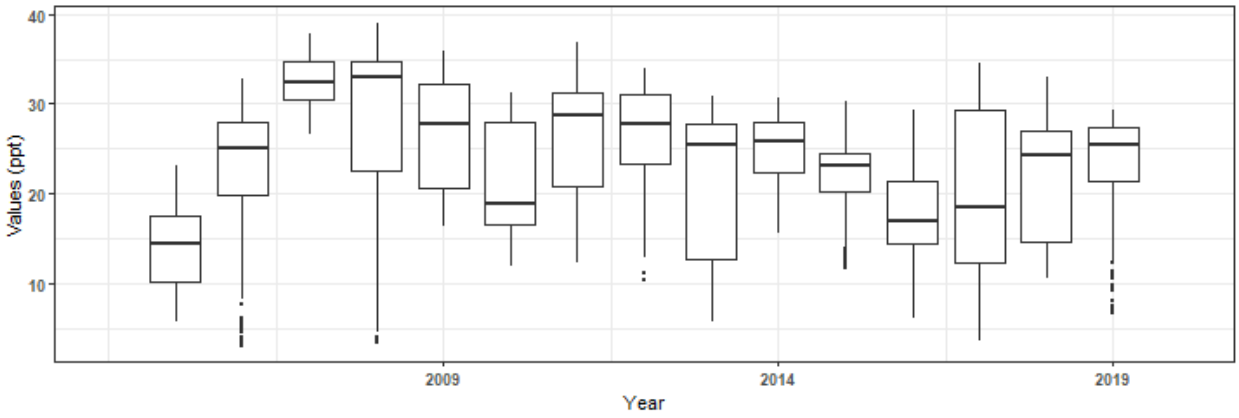
By Month



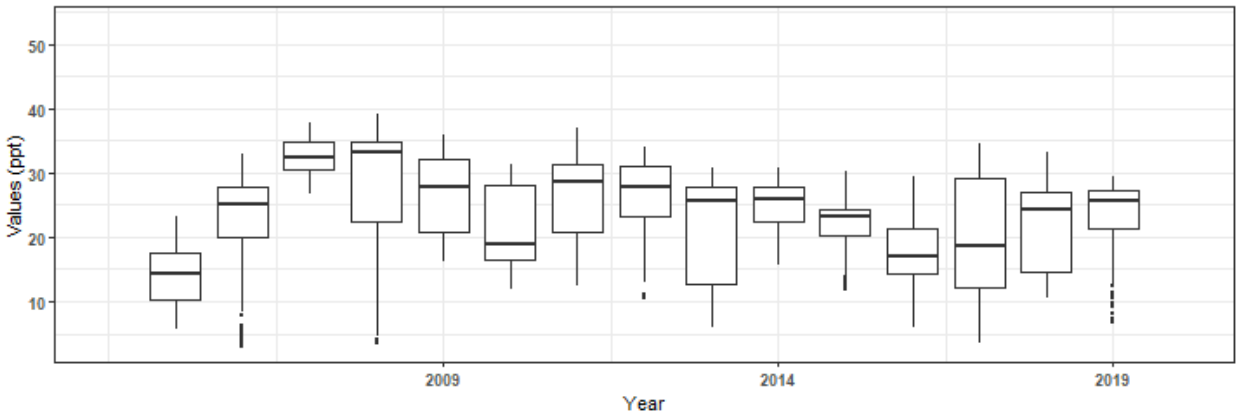
Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP2B

By Year

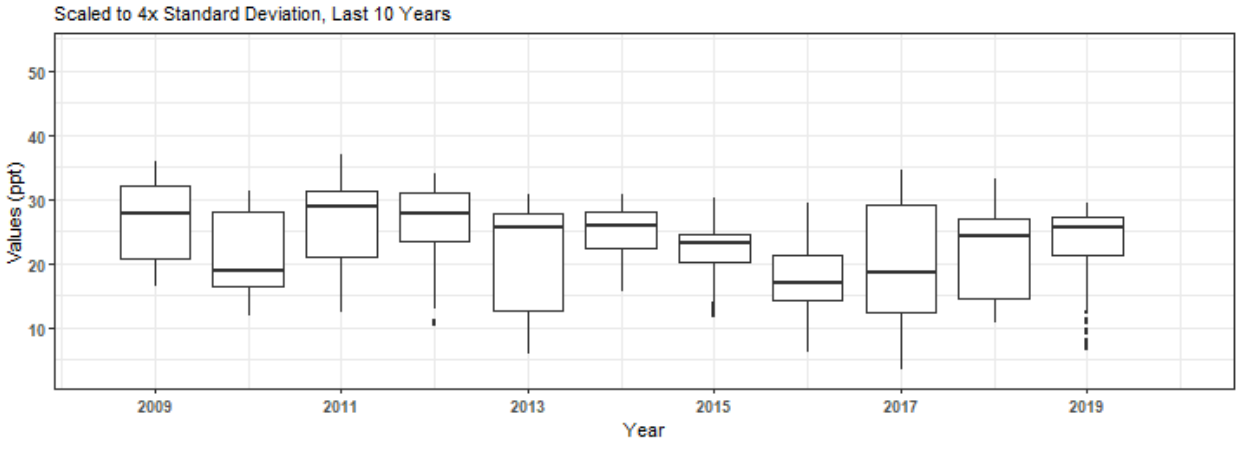
Autoscale



Scaled to 4x Standard Deviation

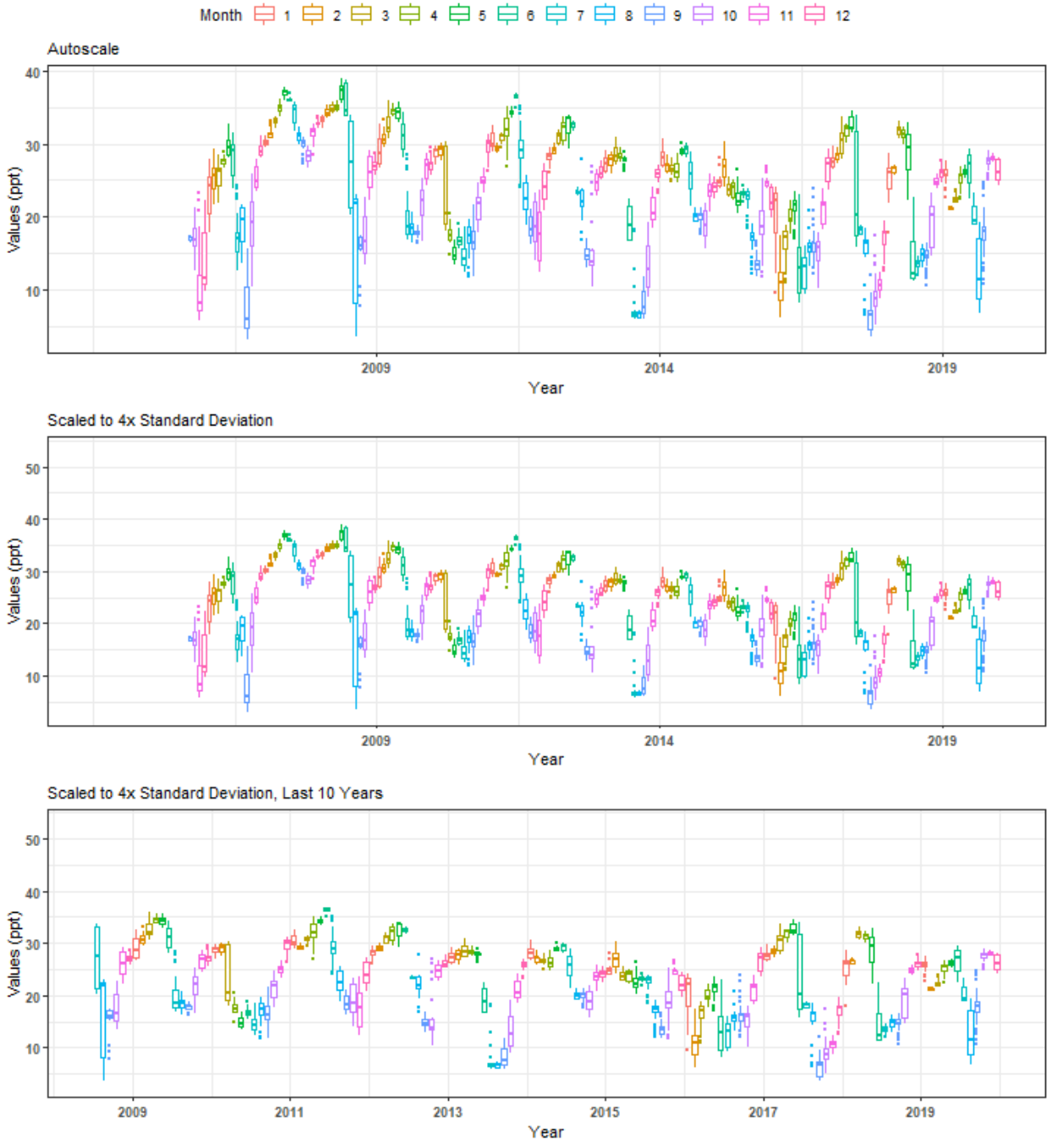


Scaled to 4x Standard Deviation, Last 10 Years



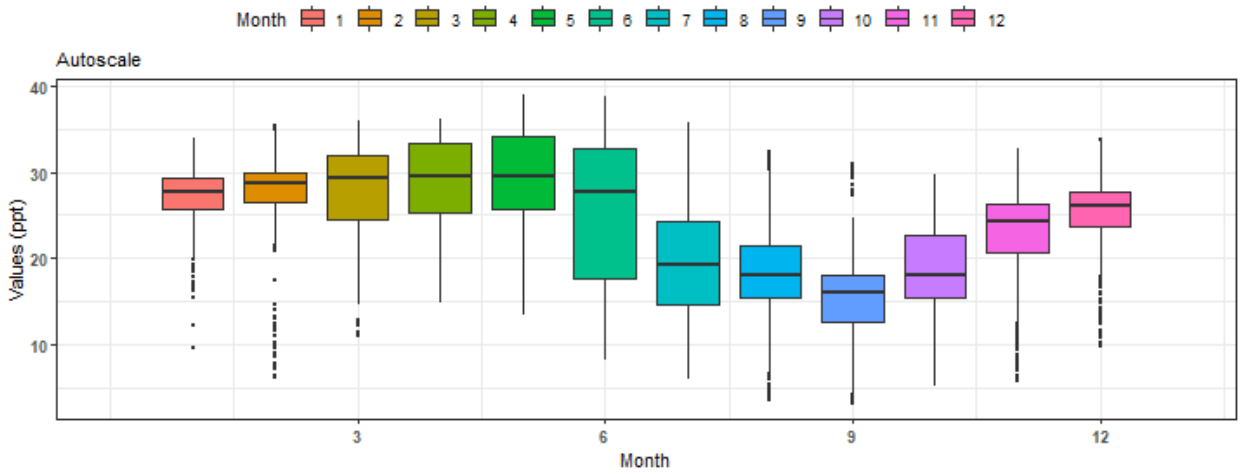
Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP2B

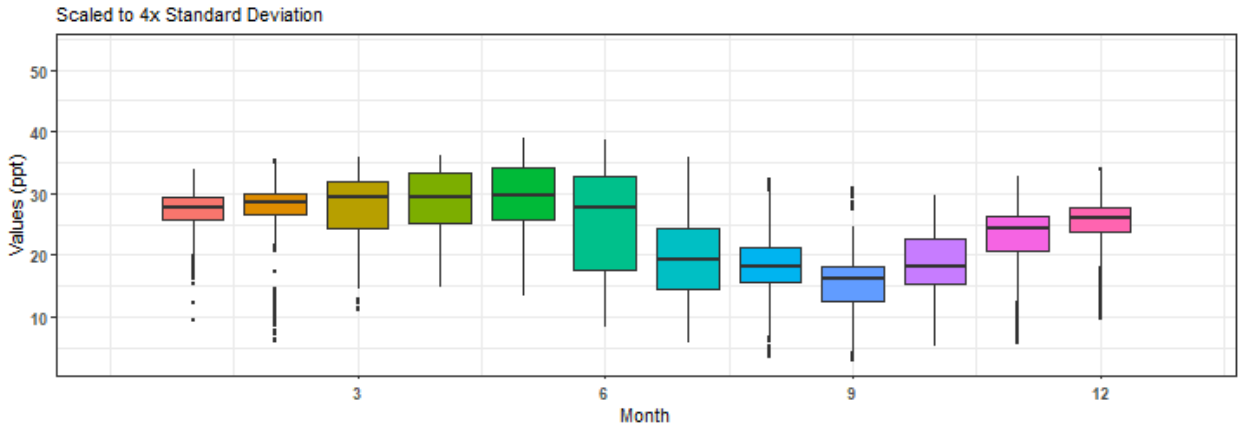
By Year & Month

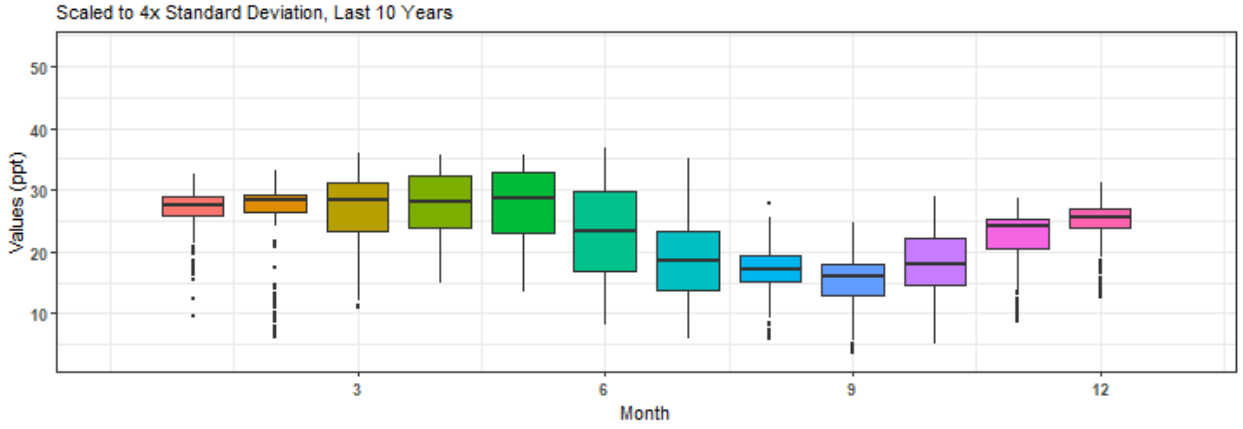


Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP2B

By Month



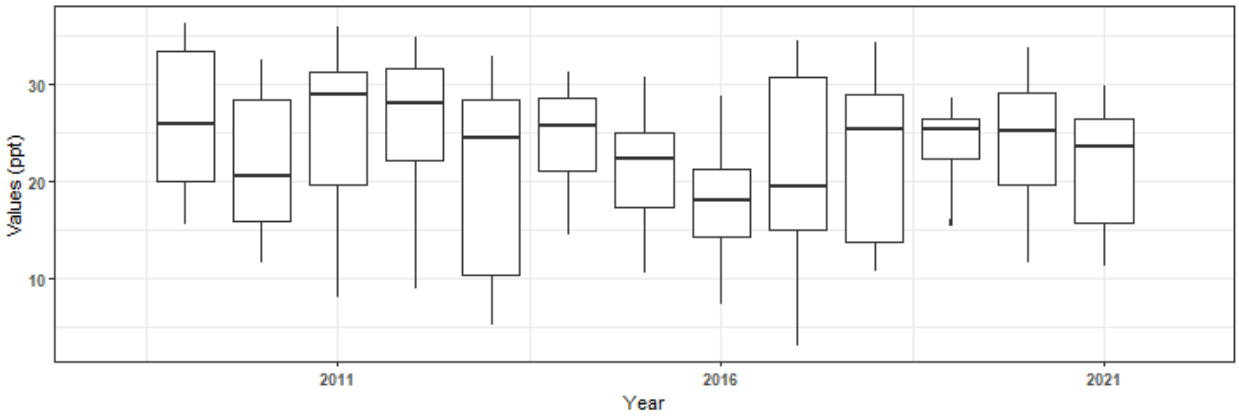




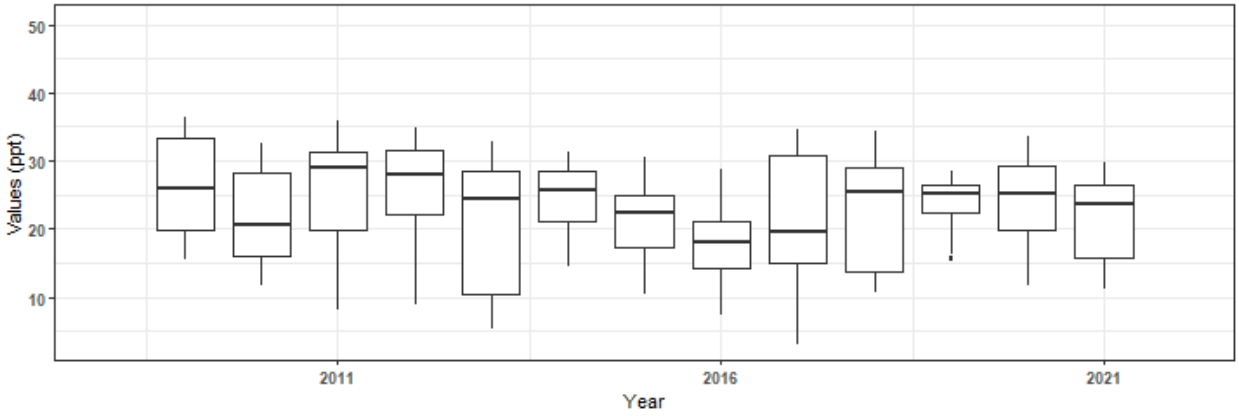
Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP3C

By Year

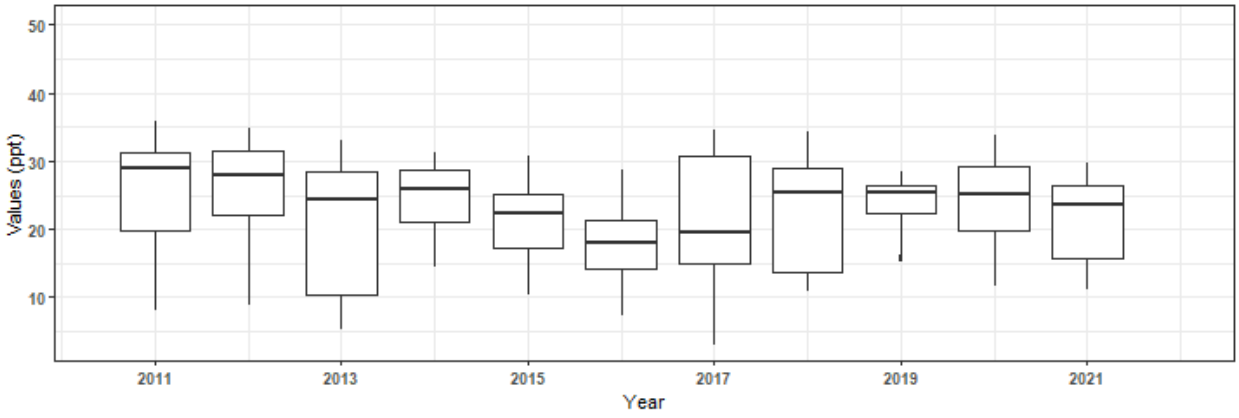
Autoscale



Scaled to 4x Standard Deviation

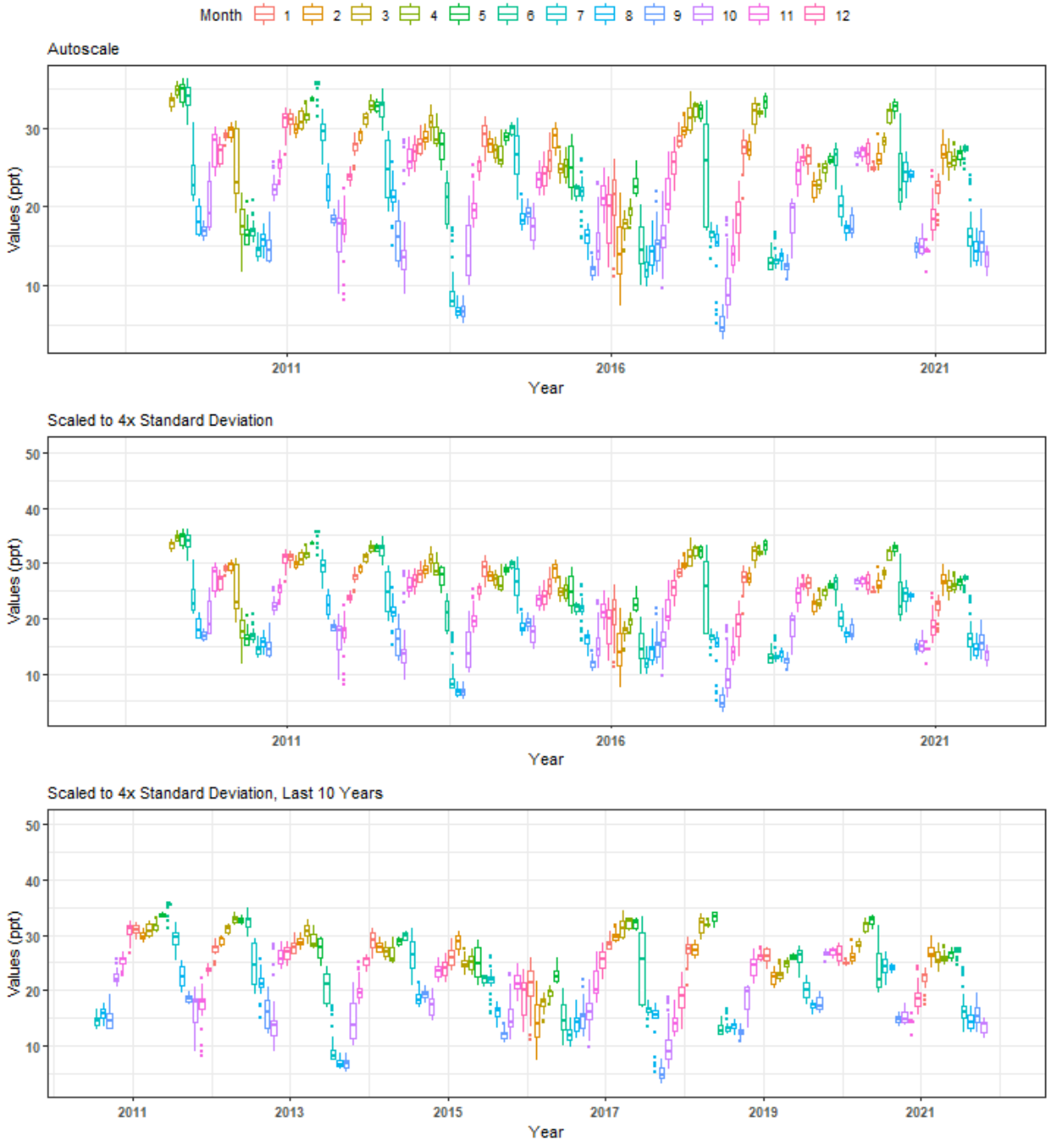


Scaled to 4x Standard Deviation, Last 10 Years



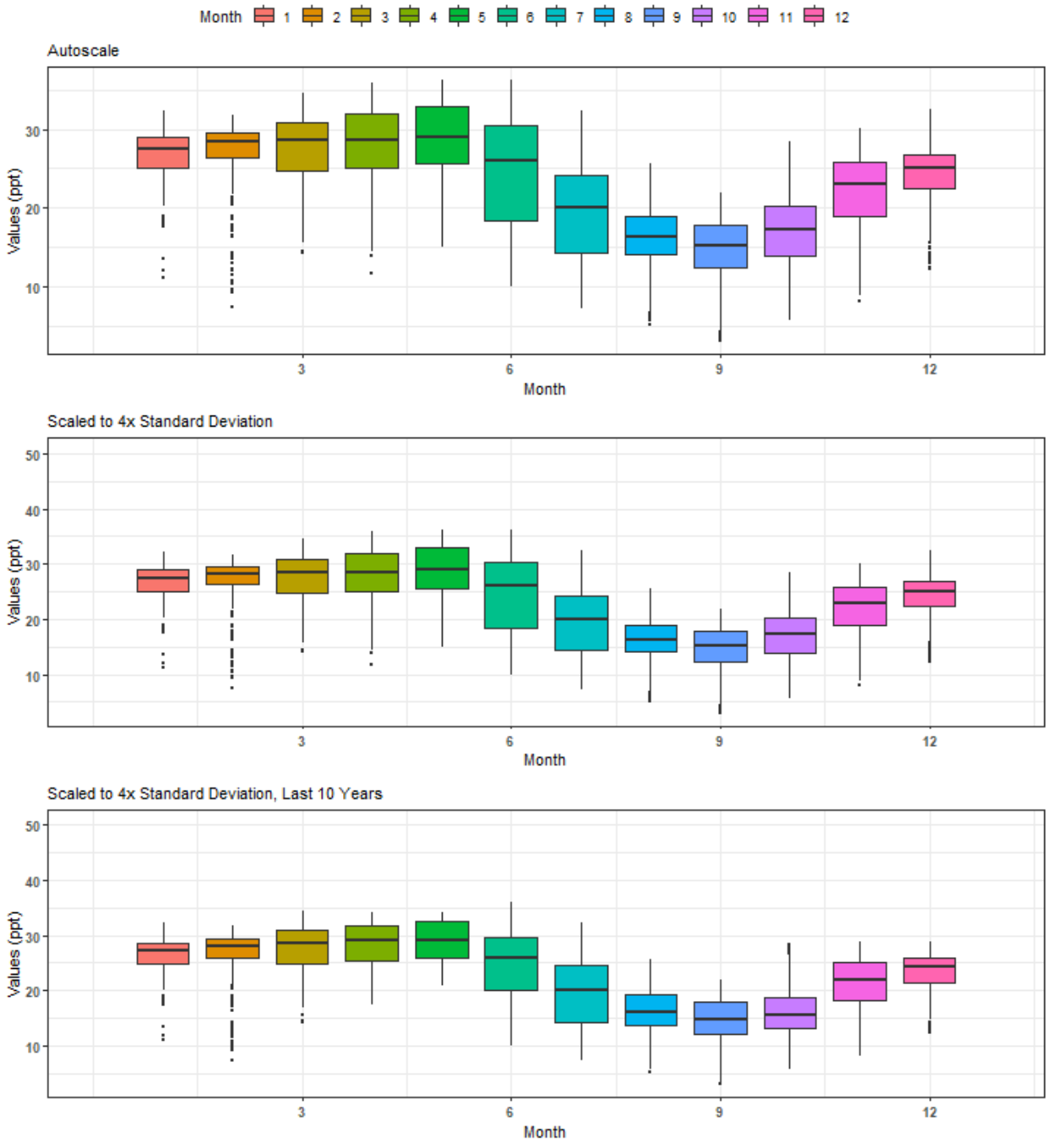
Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP3C

By Year & Month



Summary Box Plots for Matlacha Pass Aquatic Preserve
512 | Matlacha Pass Aquatic Preserve Continuous Water Quality Monitoring Program | MP3C

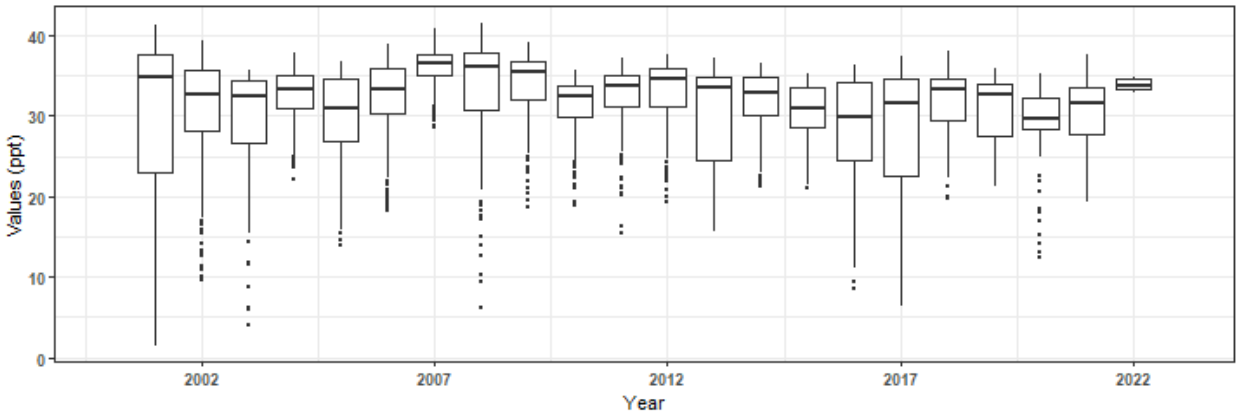
By Month



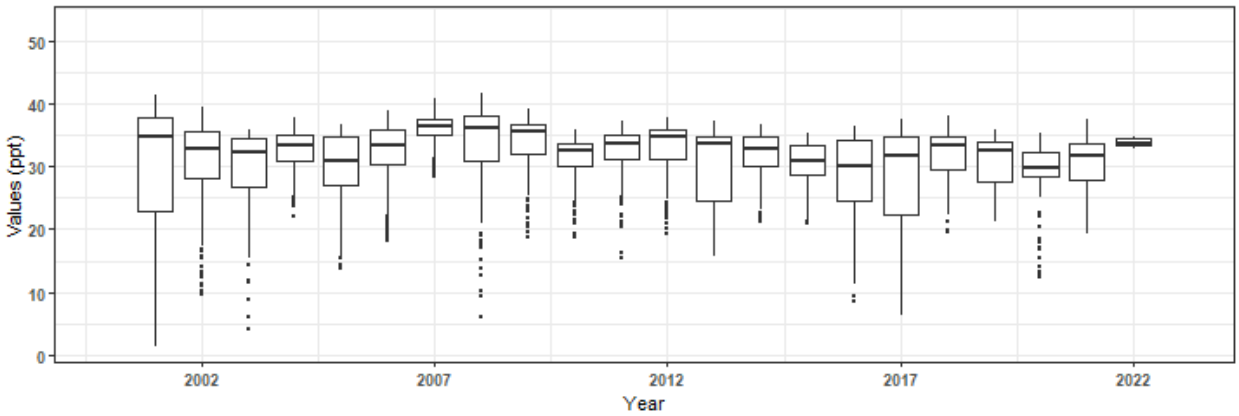
Summary Box Plots for Rookery Bay Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

By Year

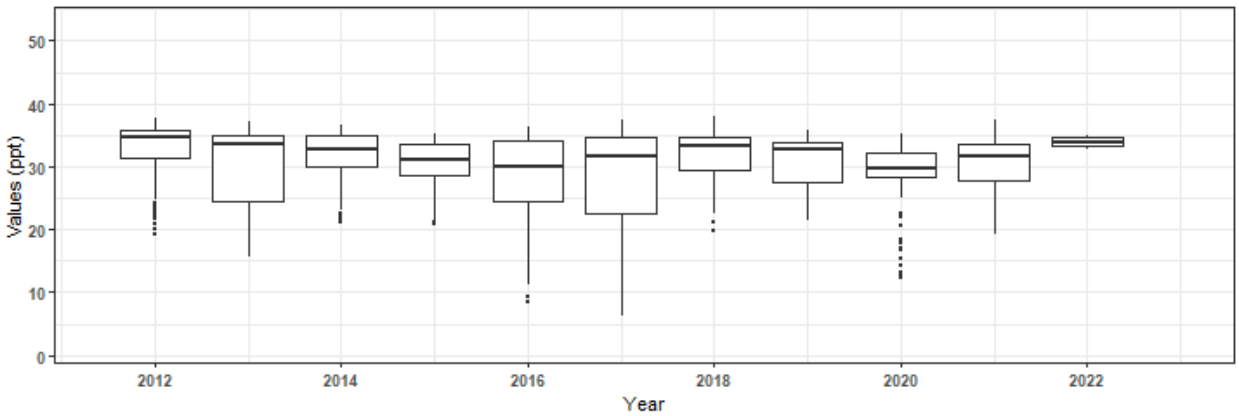
Autoscale



Scaled to 4x Standard Deviation

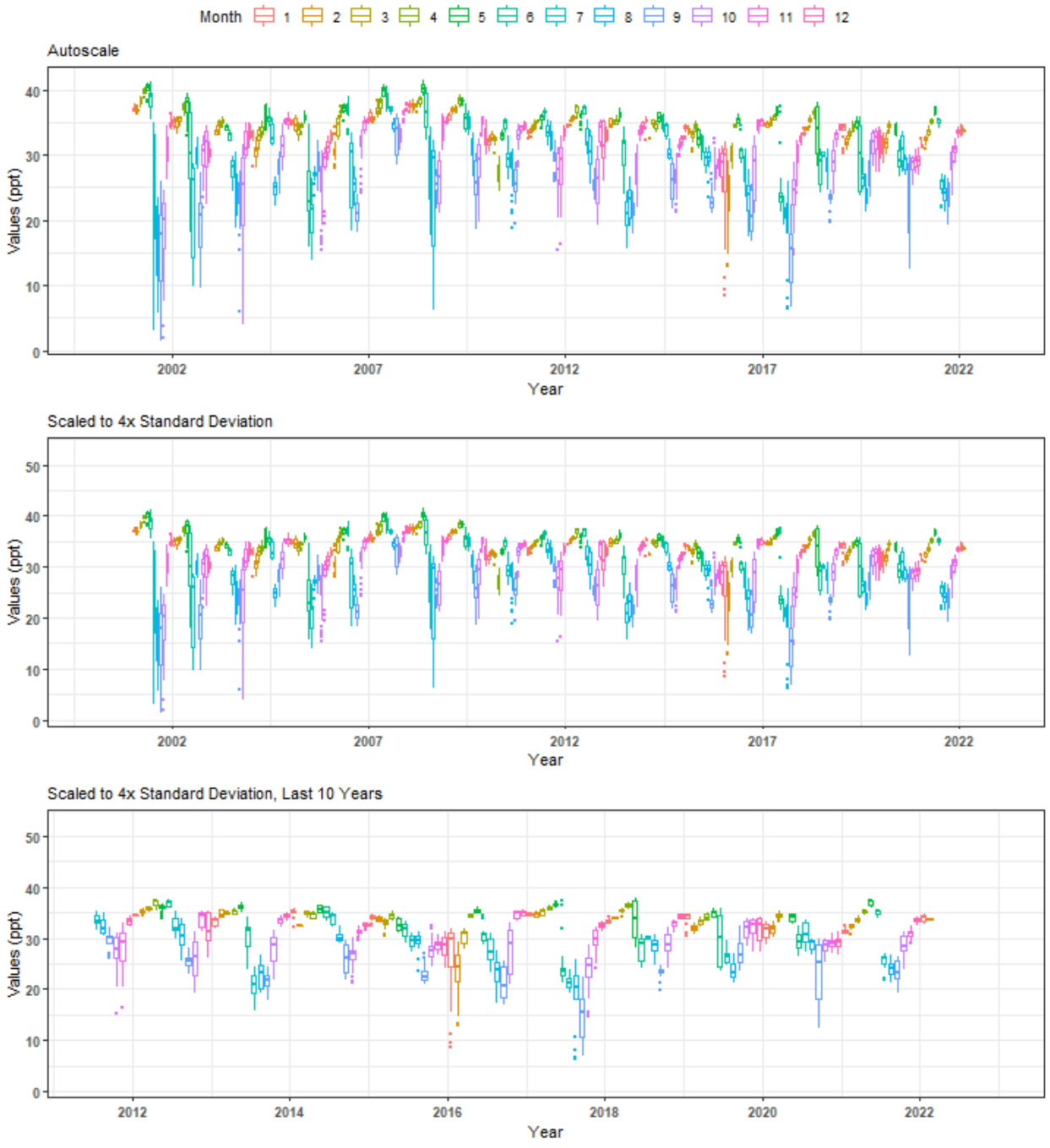


Scaled to 4x Standard Deviation, Last 10 Years



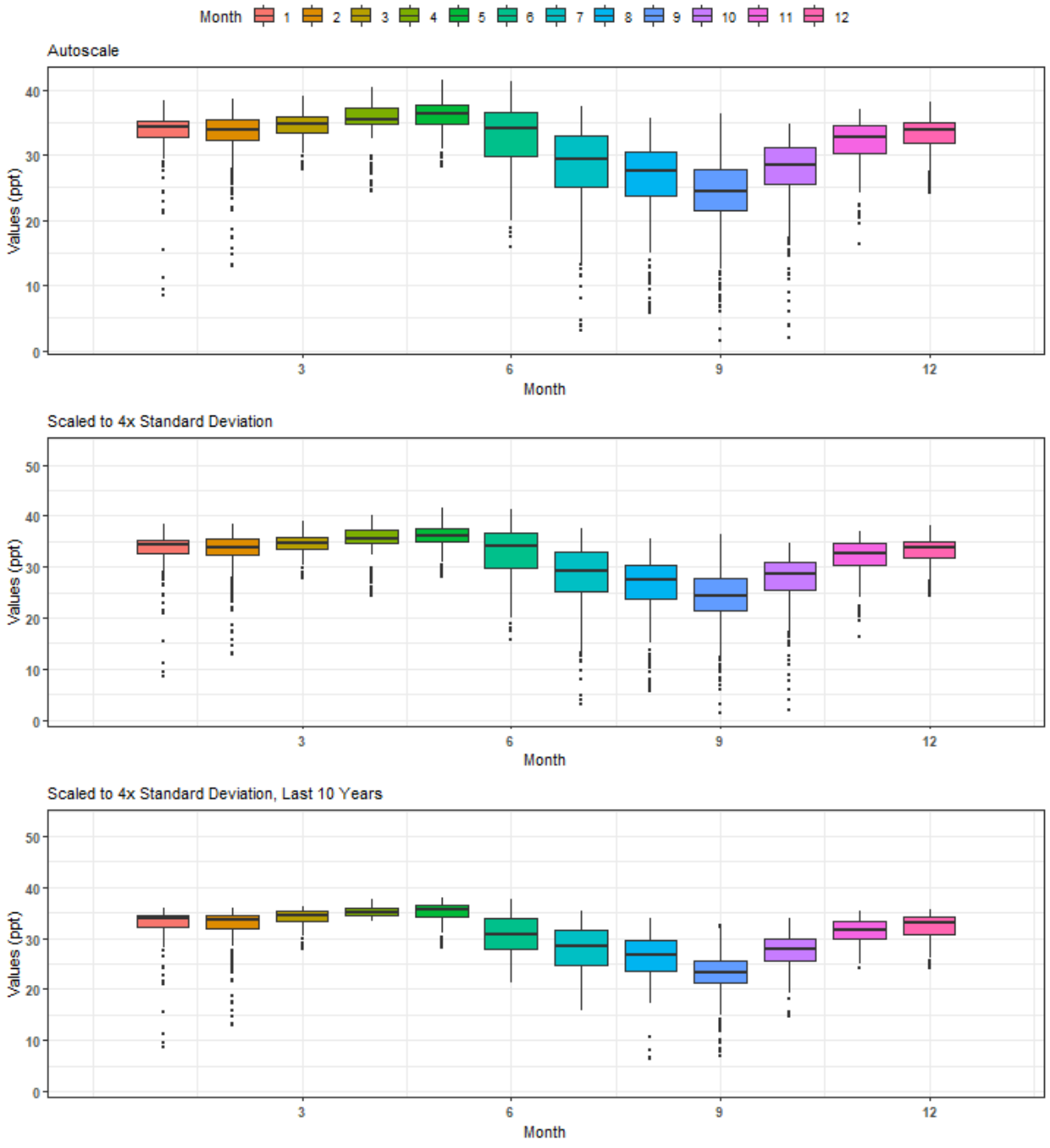
Summary Box Plots for Rookery Bay Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

By Year & Month



Summary Box Plots for Rookery Bay Aquatic Preserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

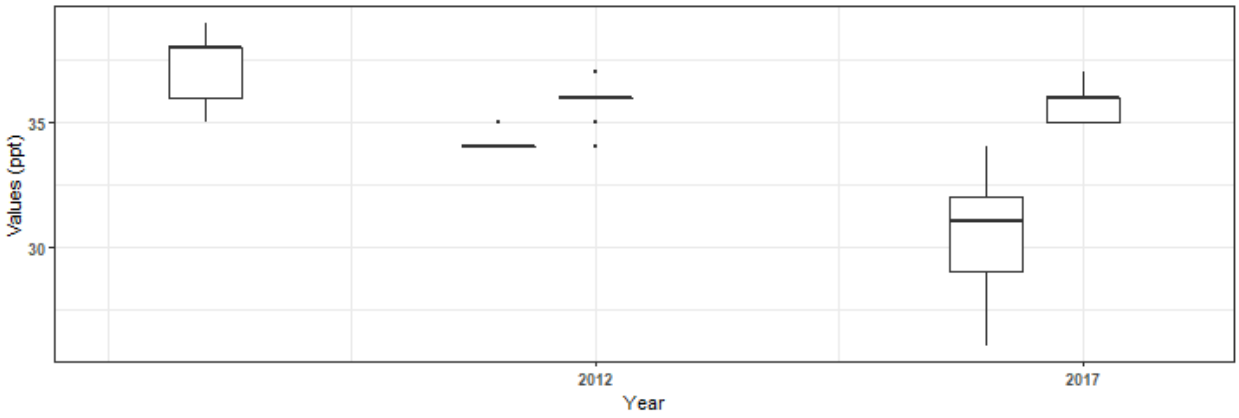
By Month



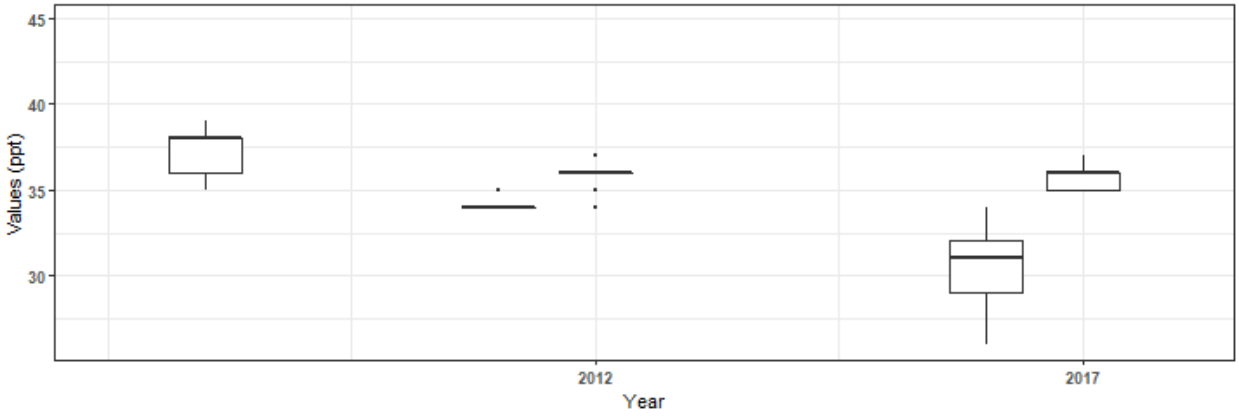
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255123081321300

By Year

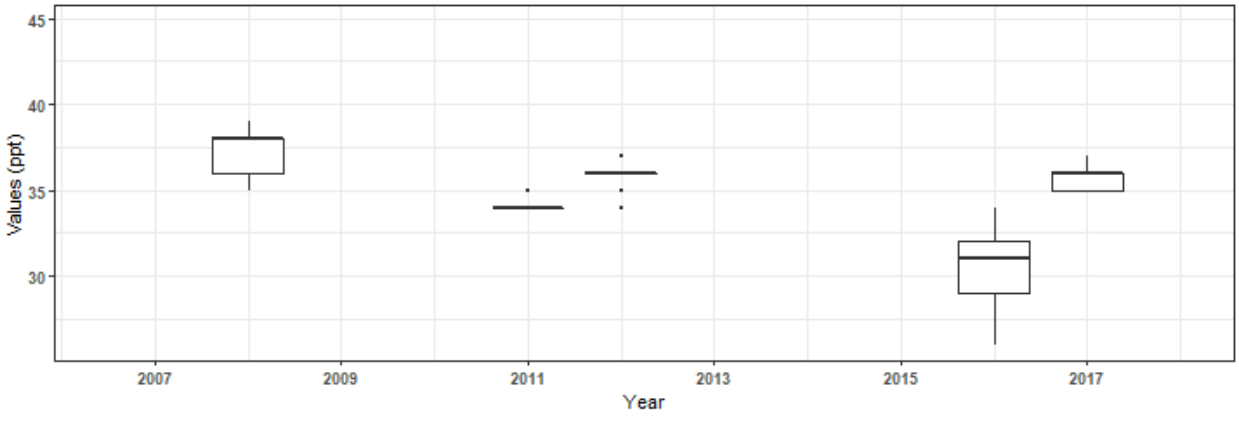
Autoscale



Scaled to 4x Standard Deviation

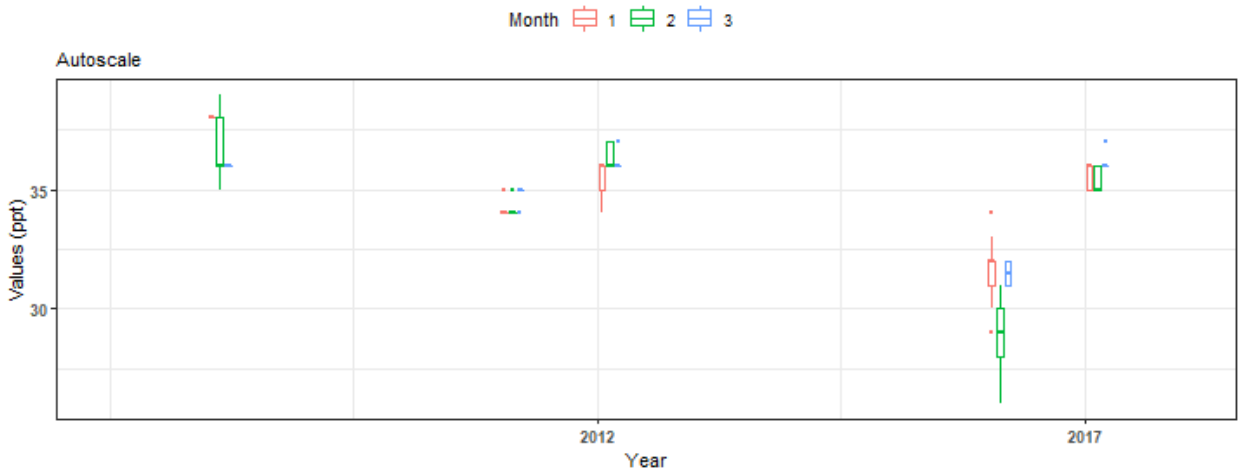


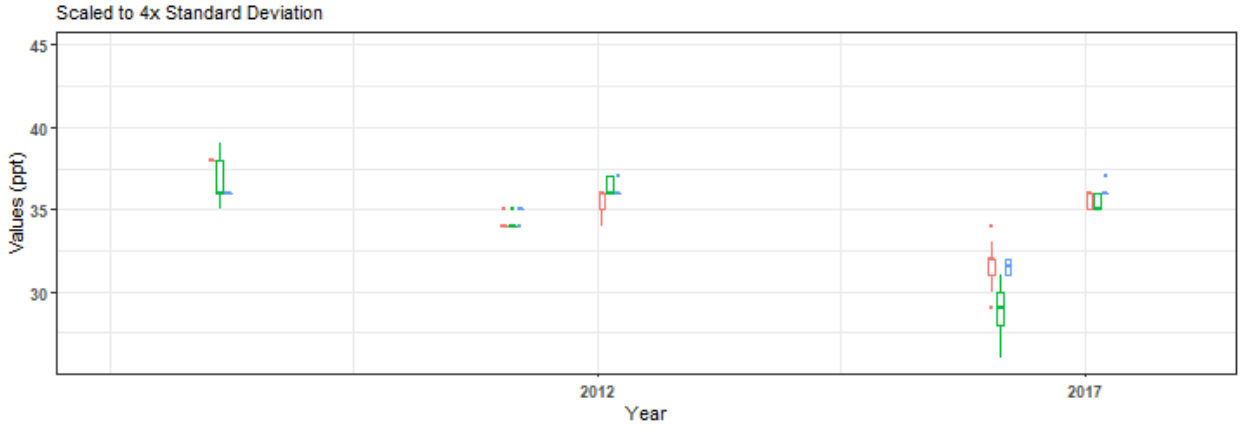
Scaled to 4x Standard Deviation, Last 10 Years

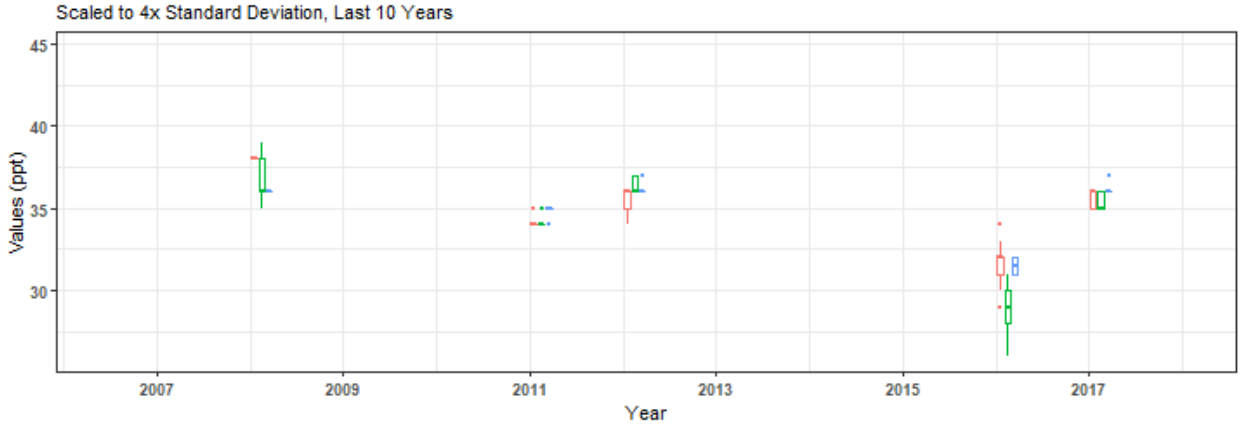


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 25512308132130O

By Year & Month

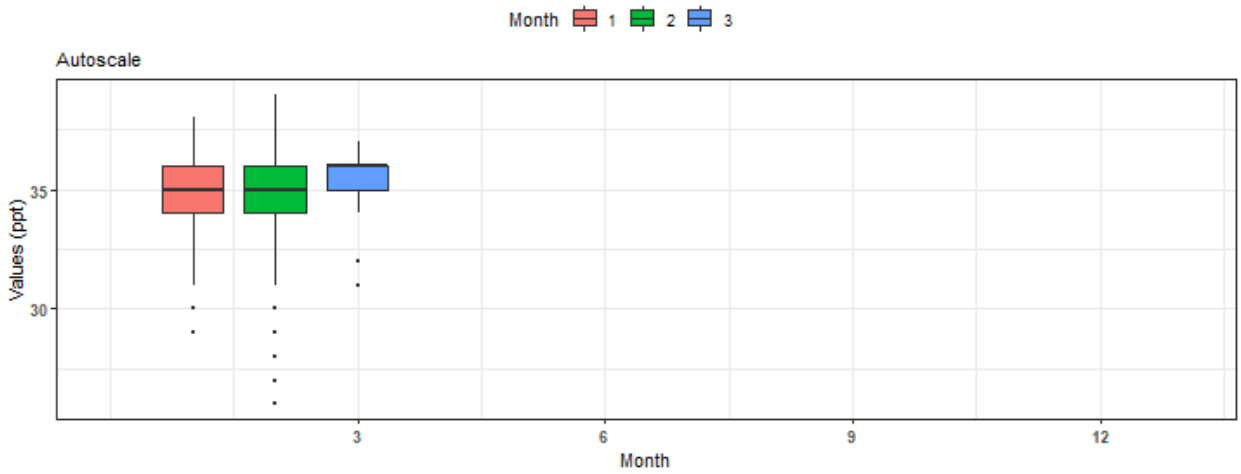


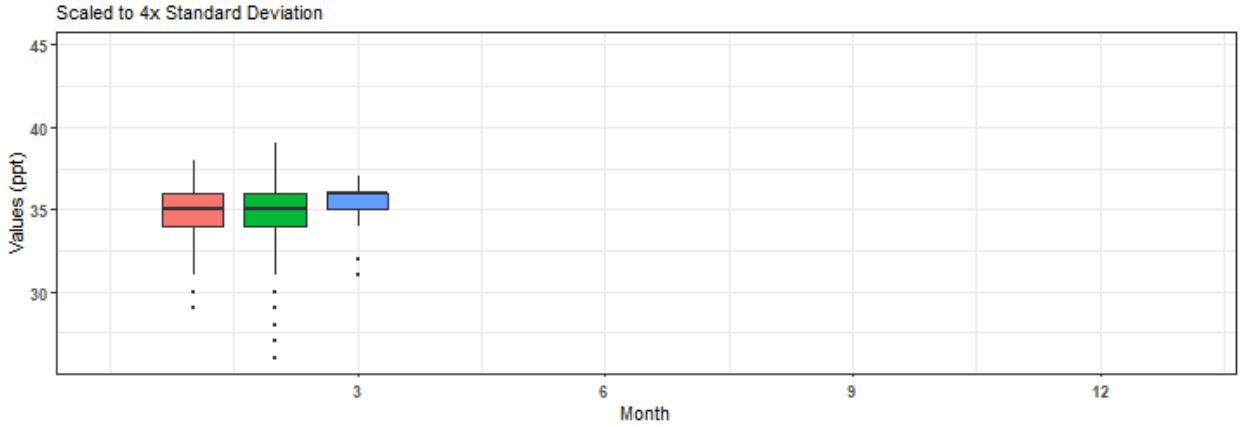


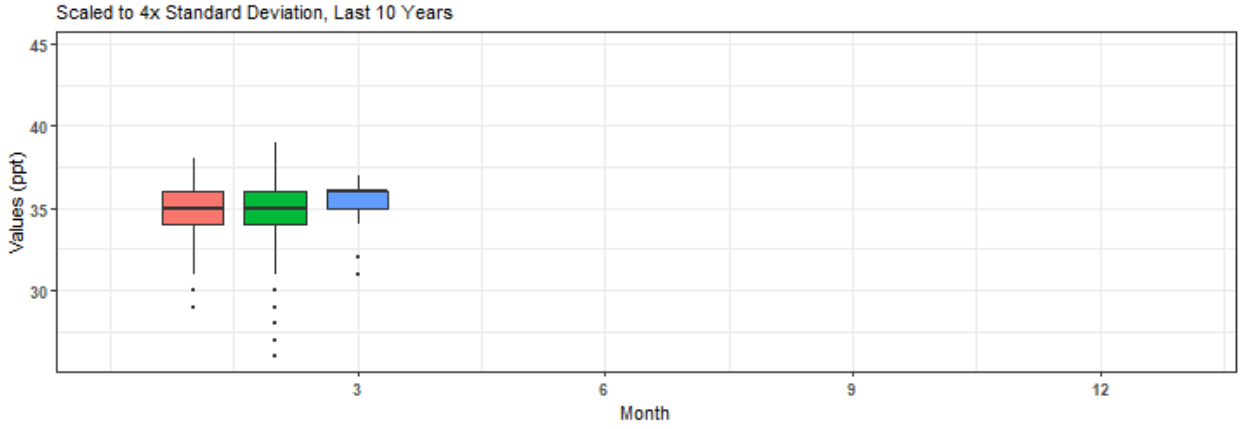


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255123081321300

By Month



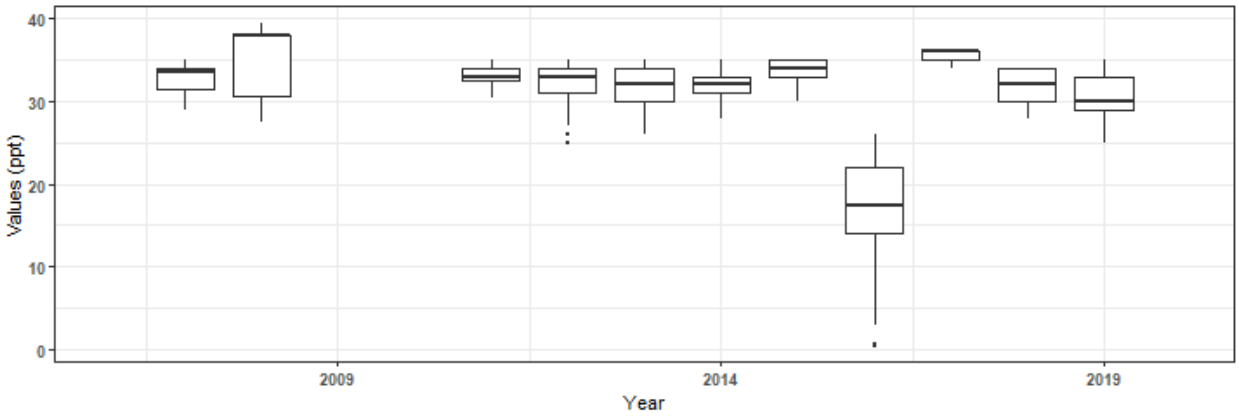




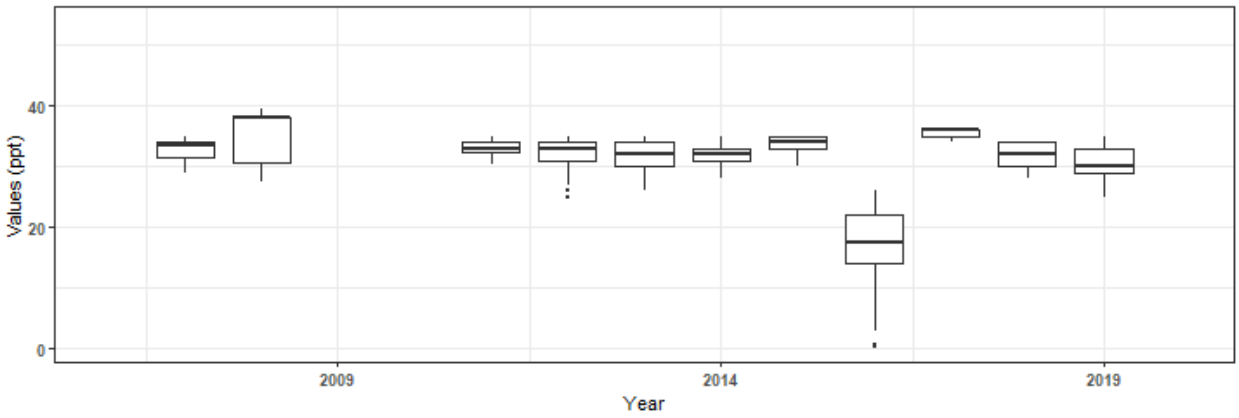
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255432081303900

By Year

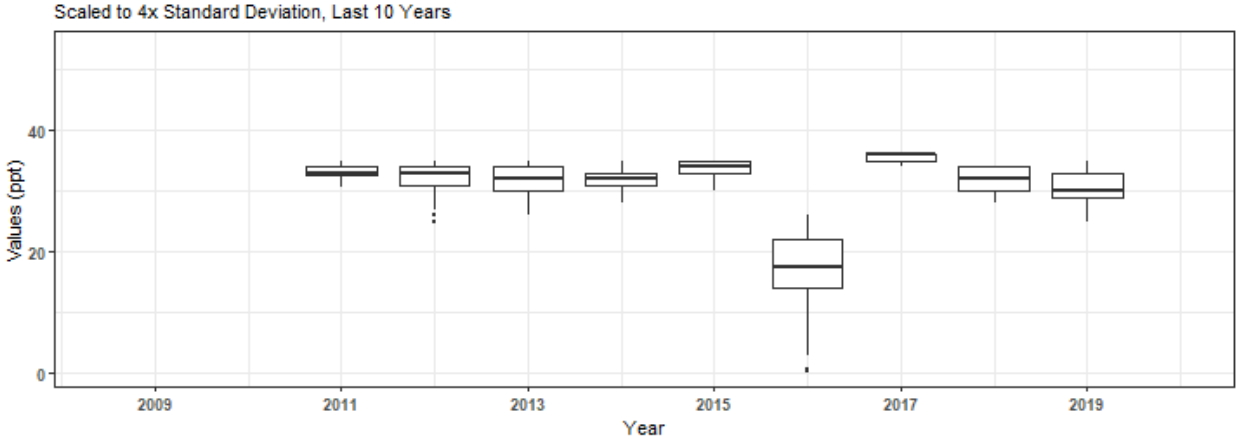
Autoscale



Scaled to 4x Standard Deviation

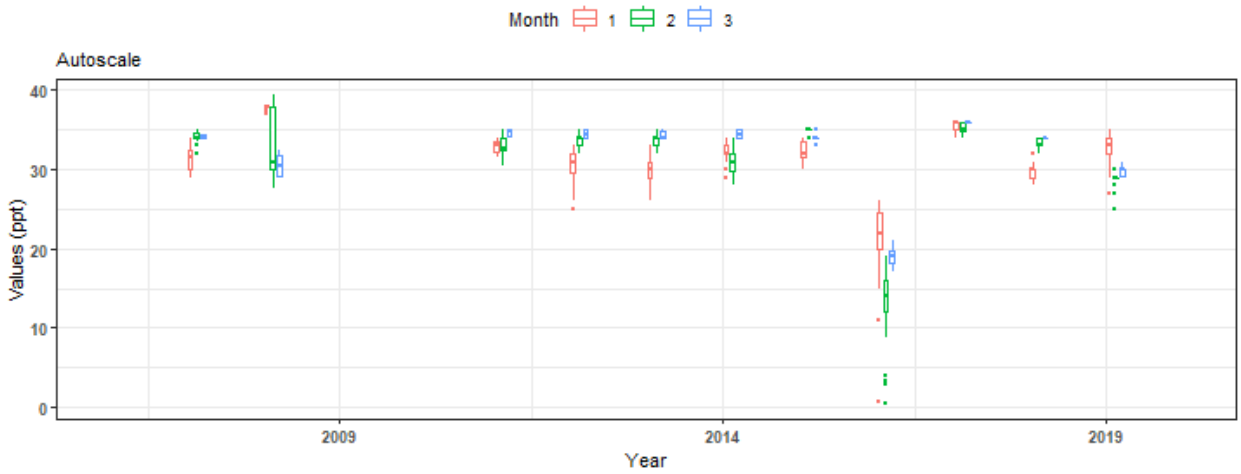


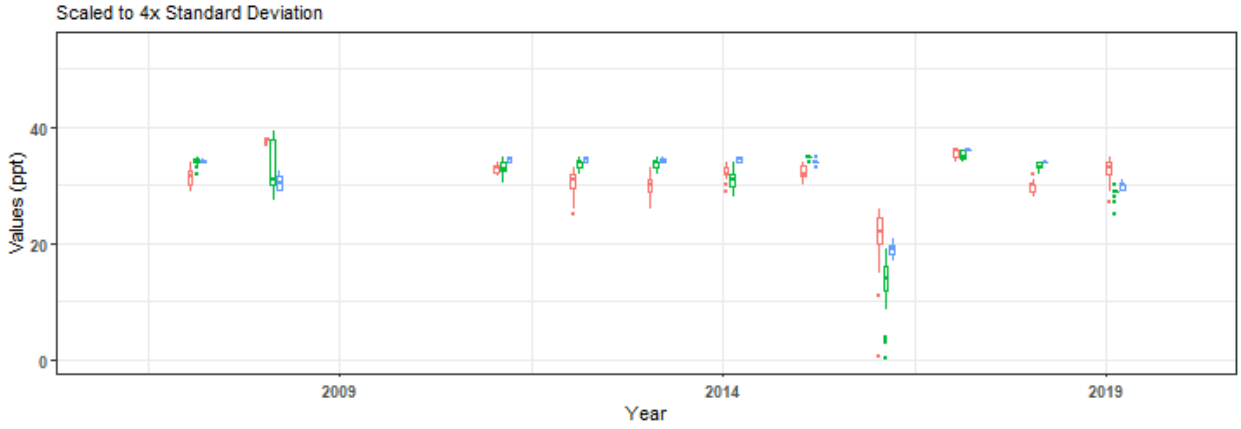
Scaled to 4x Standard Deviation, Last 10 Years

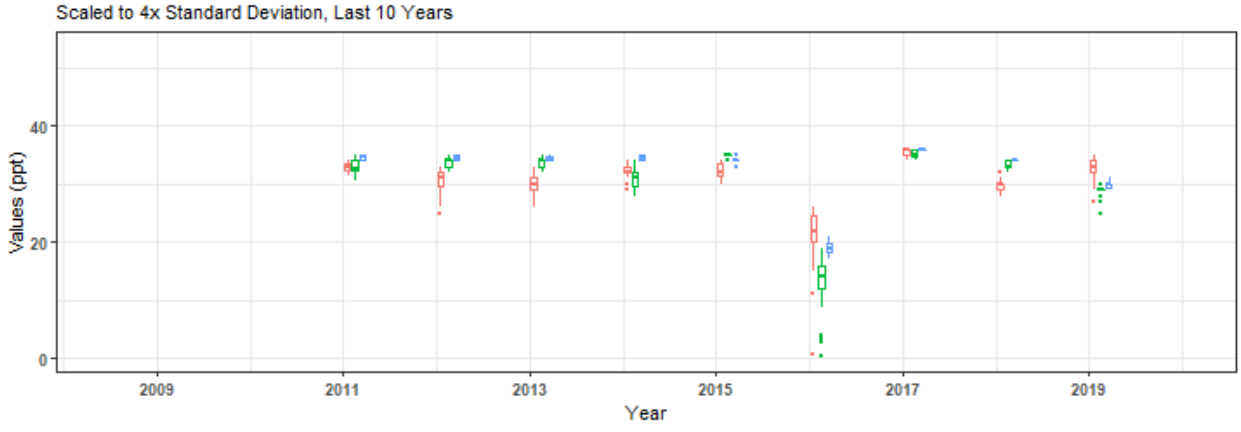


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255432081303900

By Year & Month

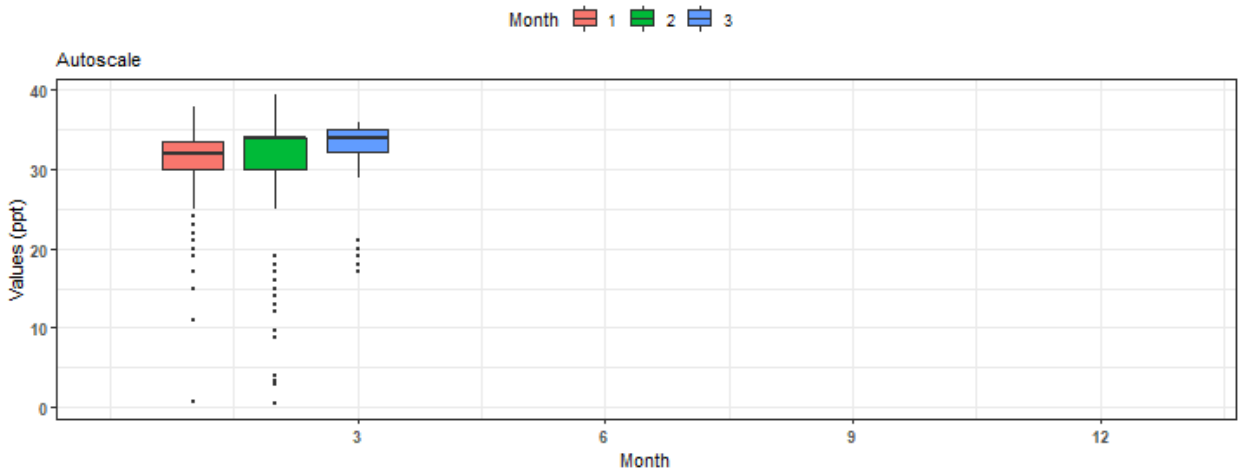


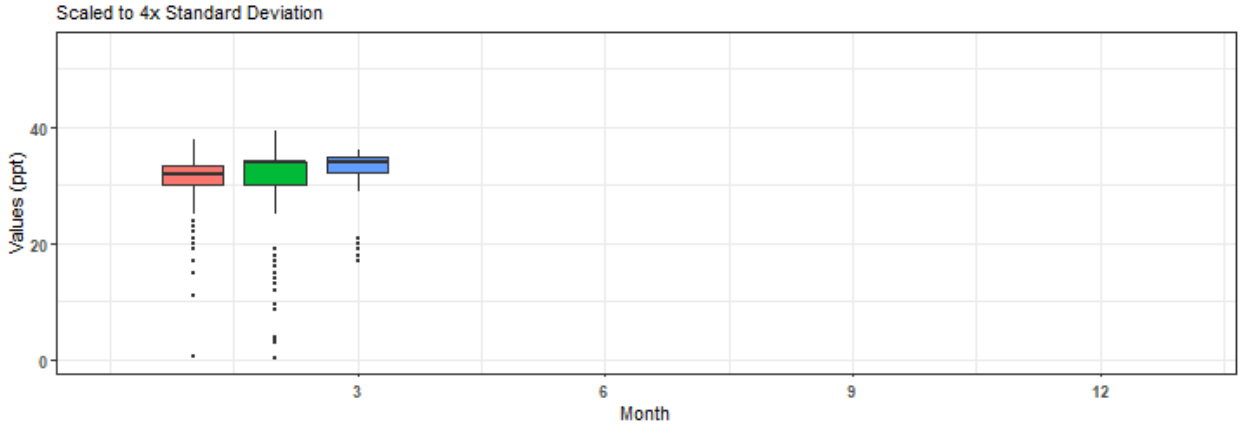


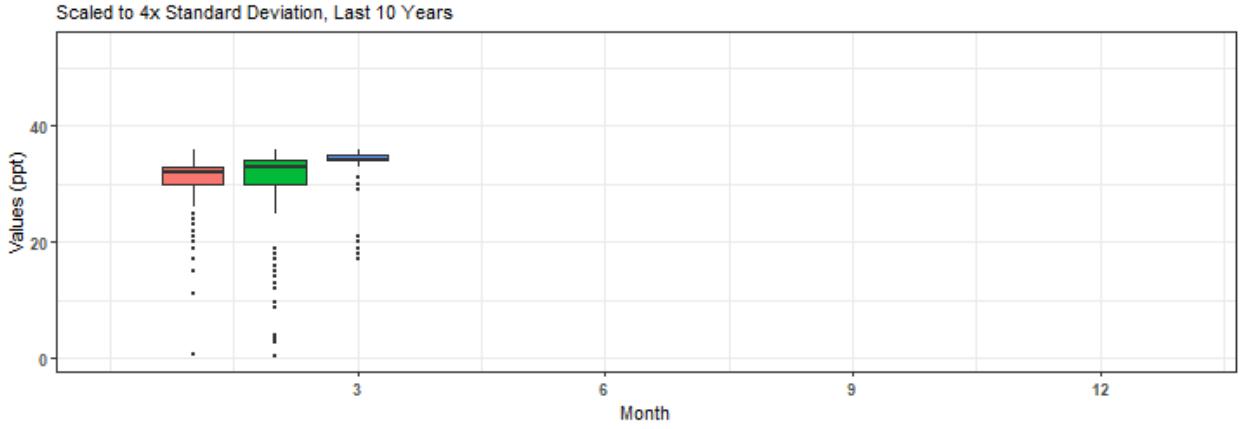


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255432081303900

By Month



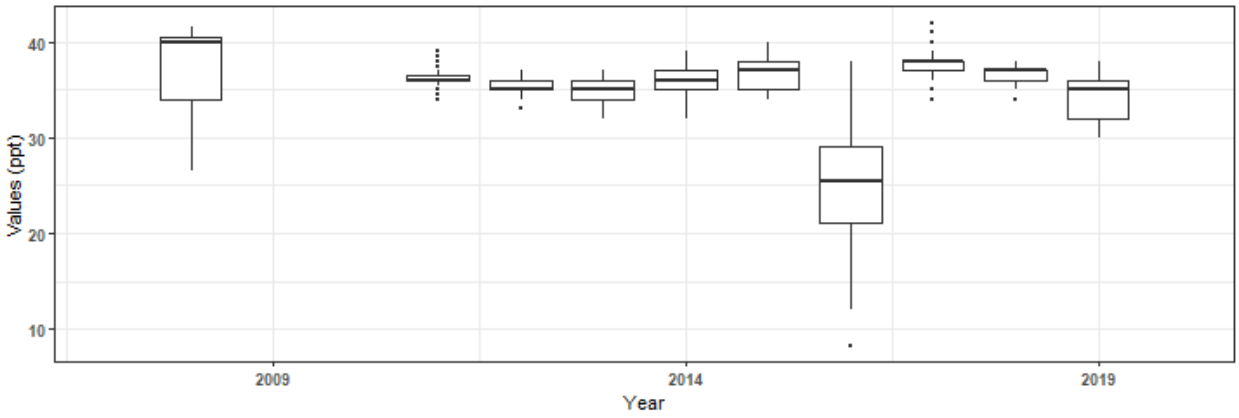




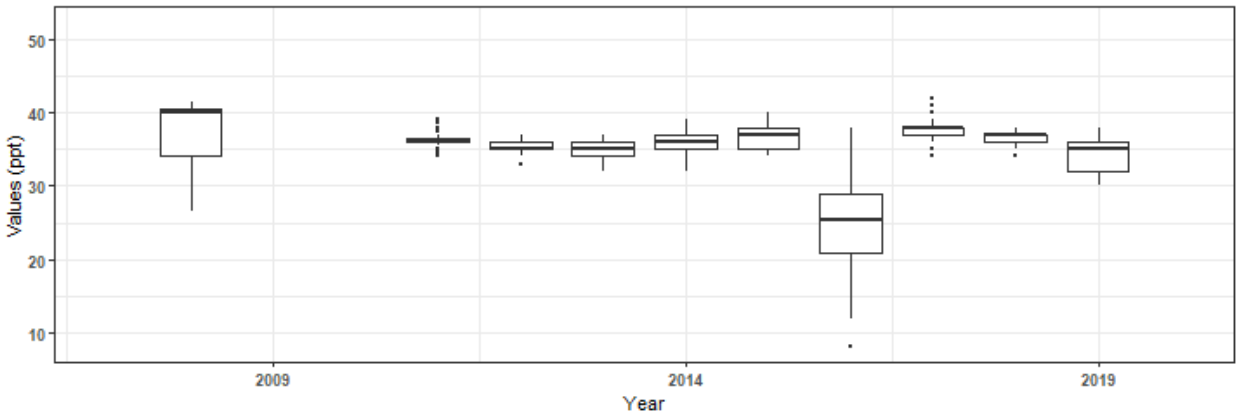
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255534081324000

By Year

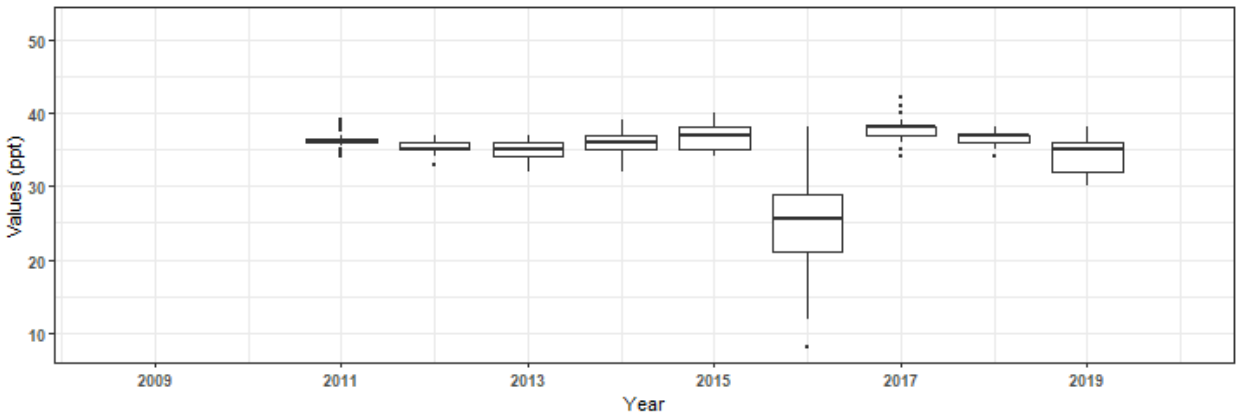
Autoscale



Scaled to 4x Standard Deviation

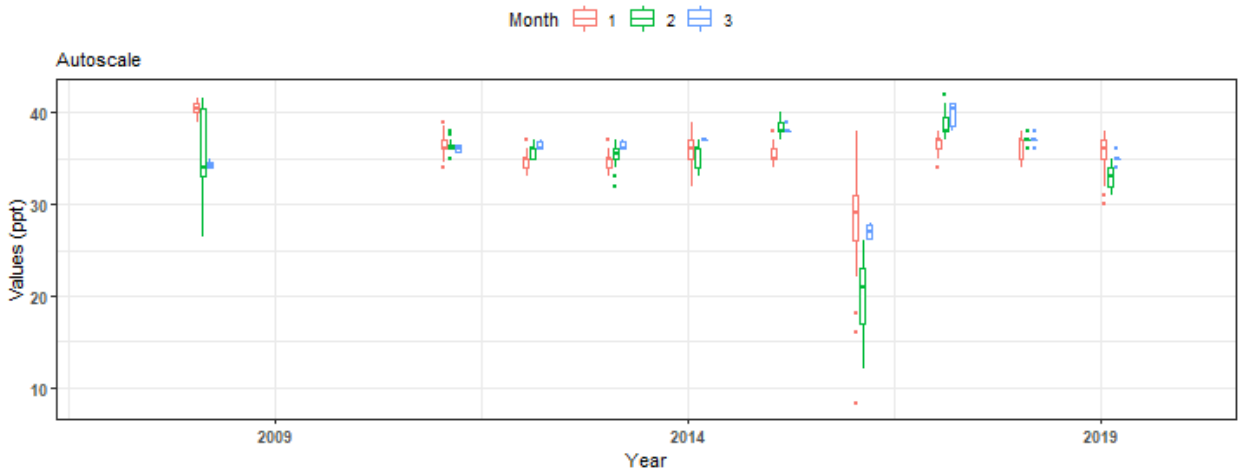


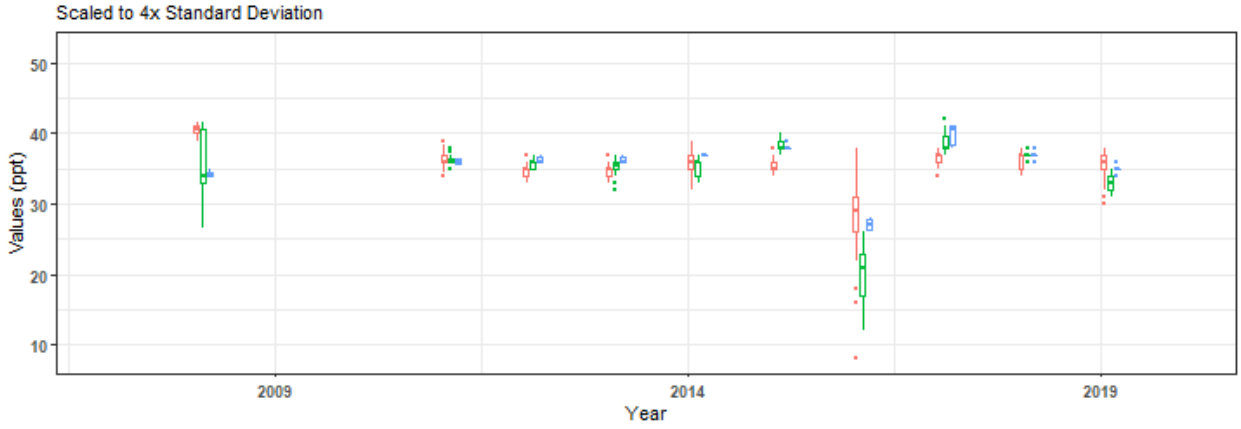
Scaled to 4x Standard Deviation, Last 10 Years

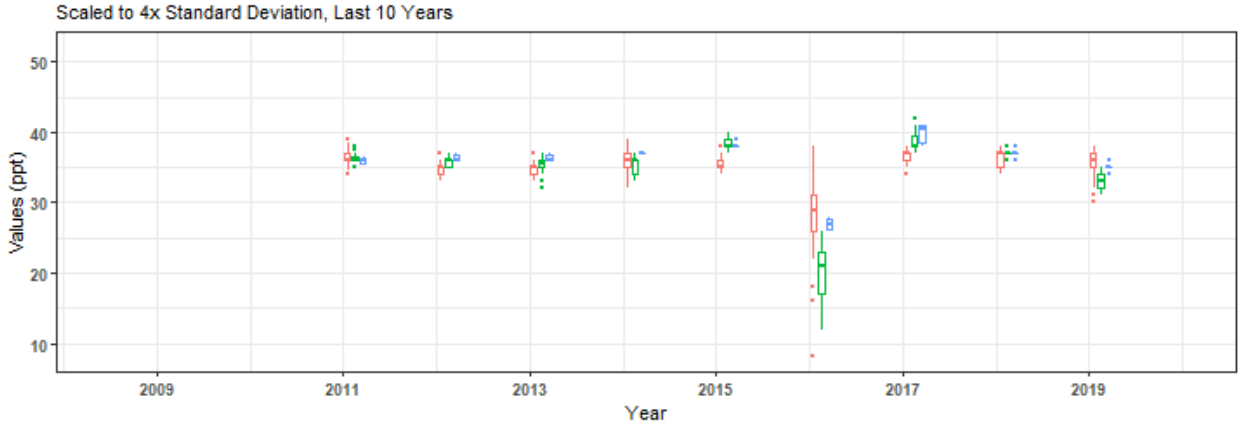


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255534081324000

By Year & Month

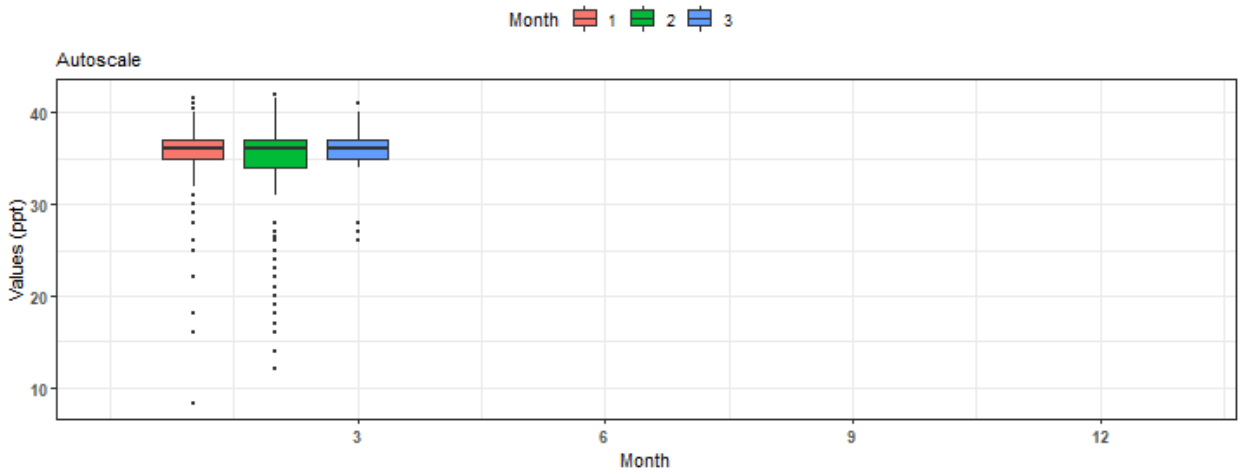


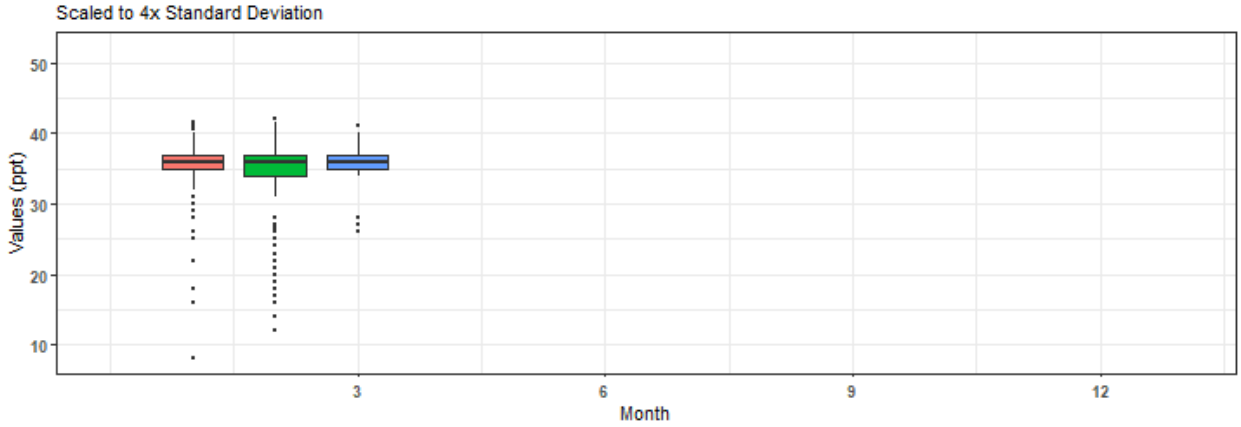


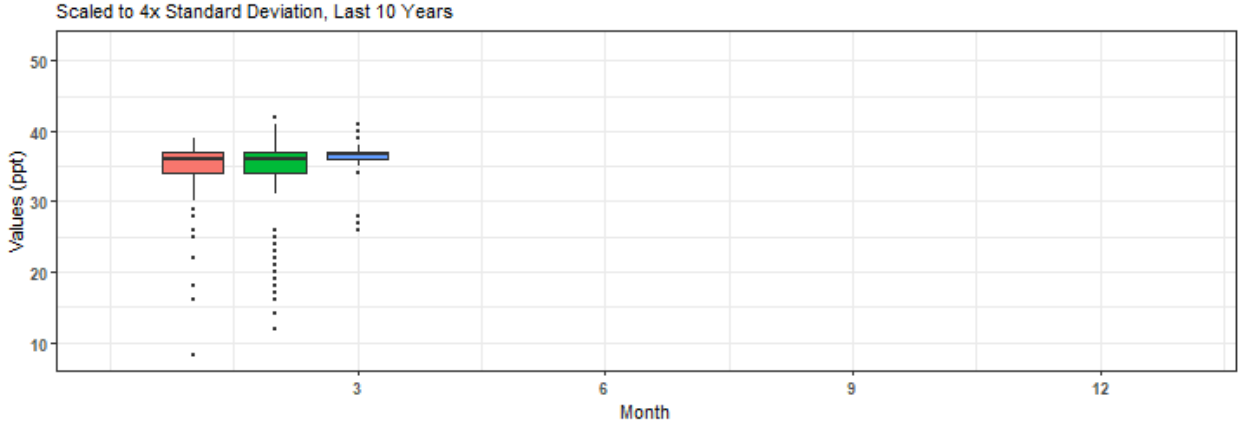


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255534081324000

By Month



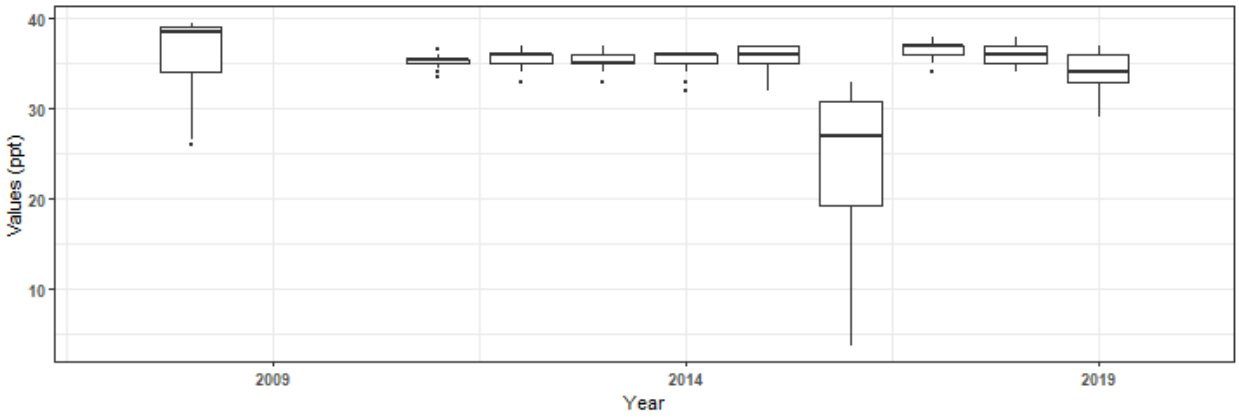




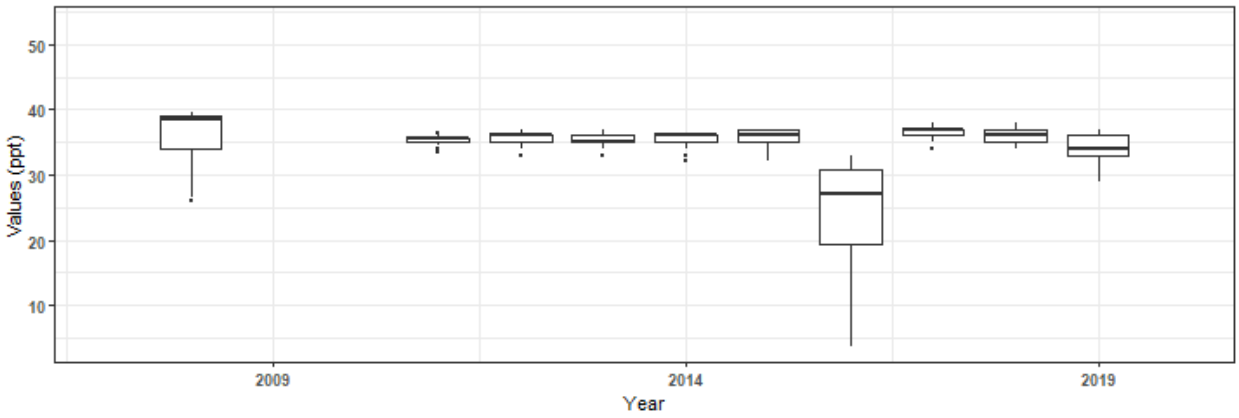
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255654081350200

By Year

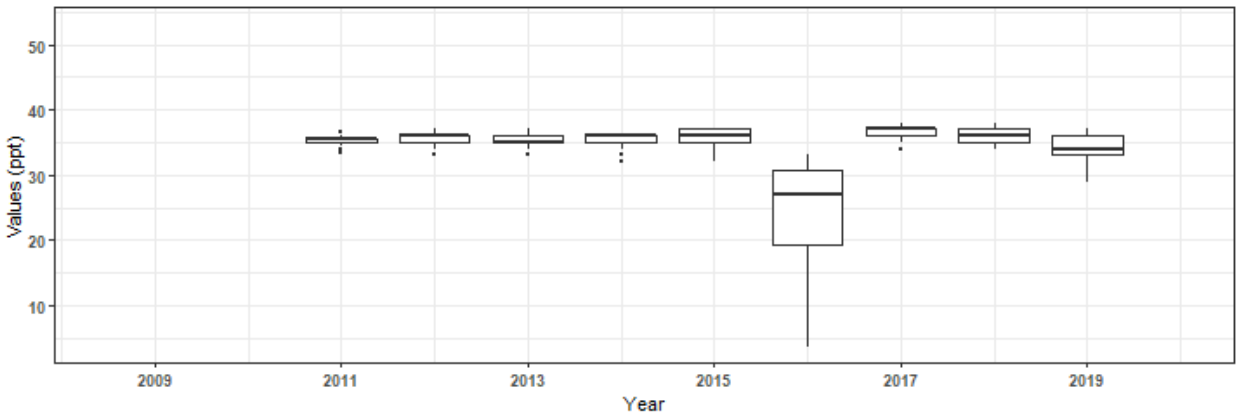
Autoscale



Scaled to 4x Standard Deviation

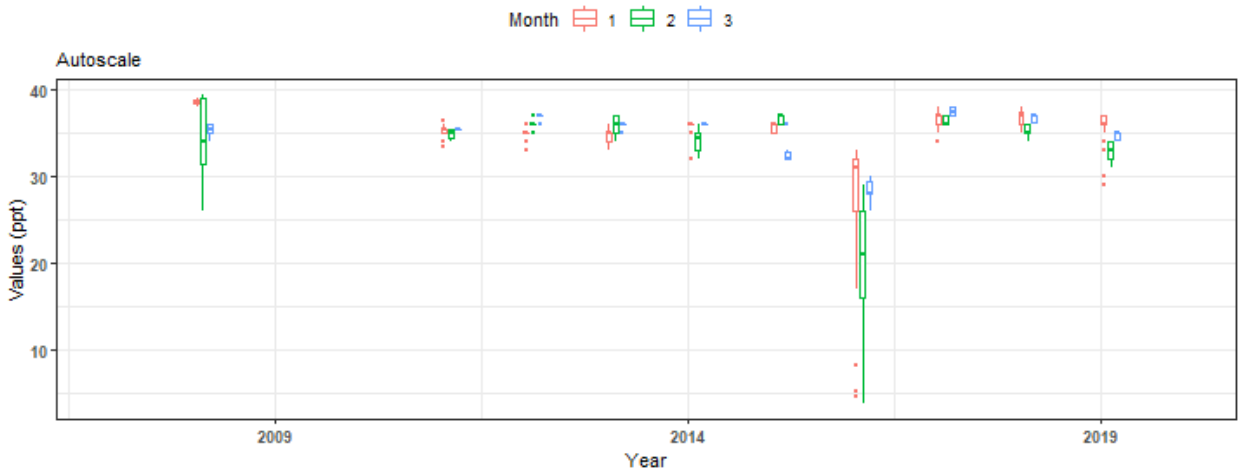


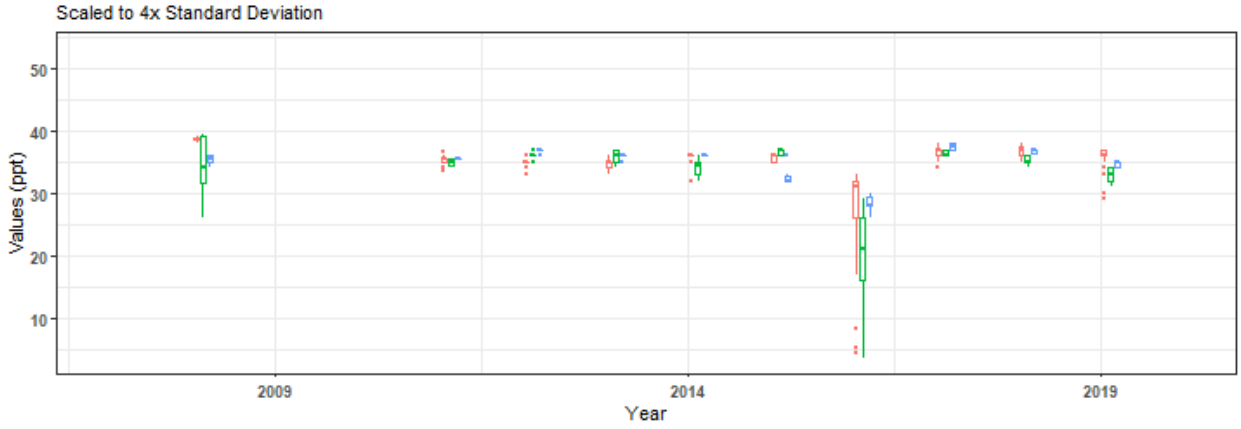
Scaled to 4x Standard Deviation, Last 10 Years

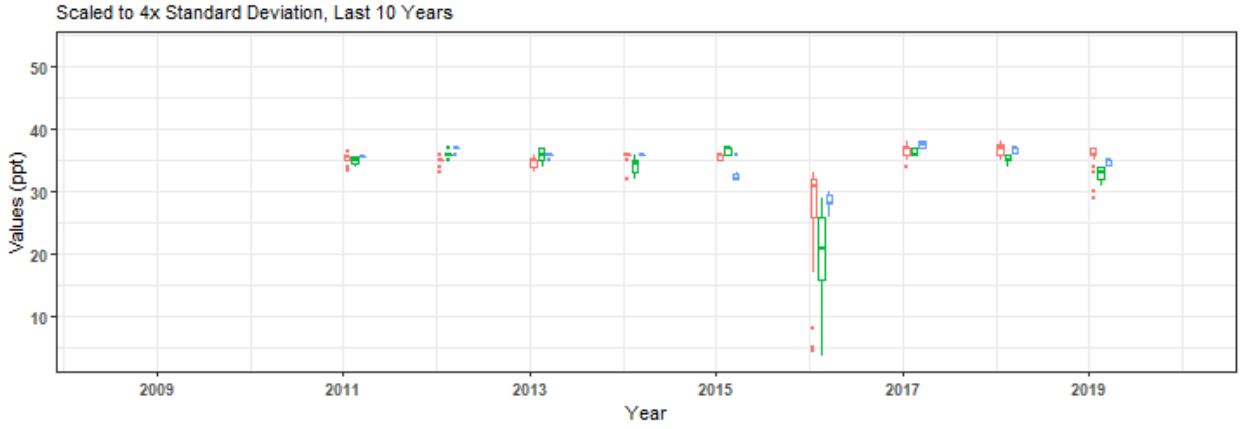


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255654081350200

By Year & Month

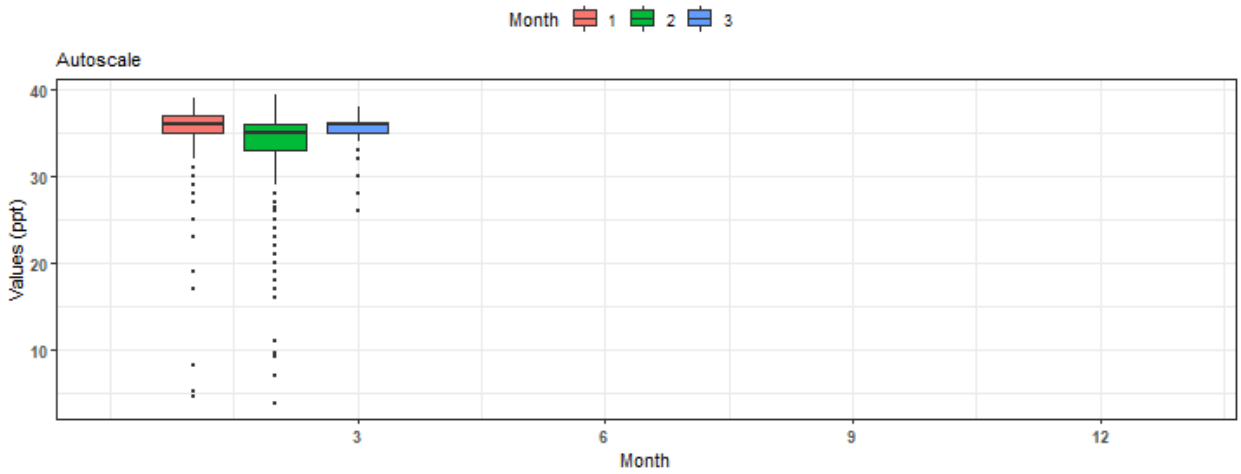


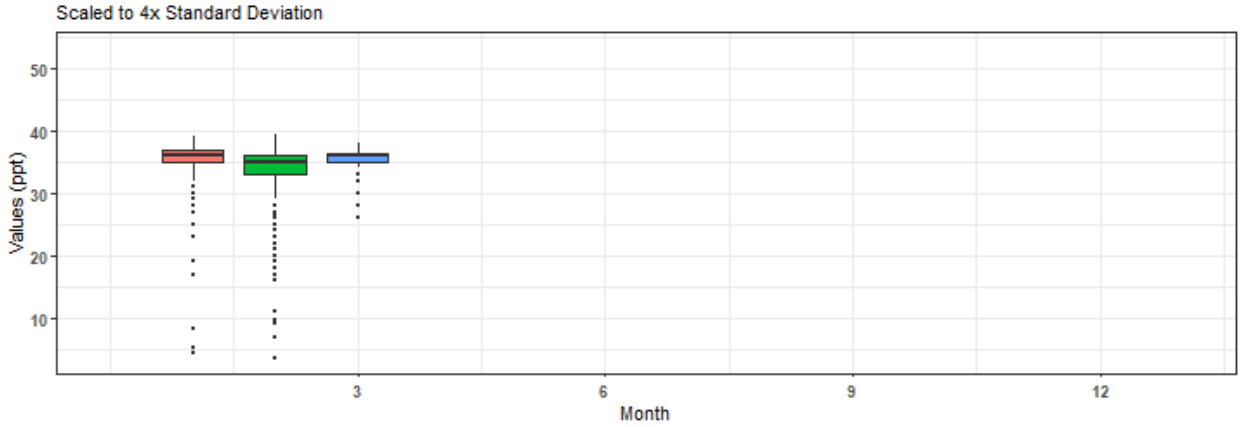


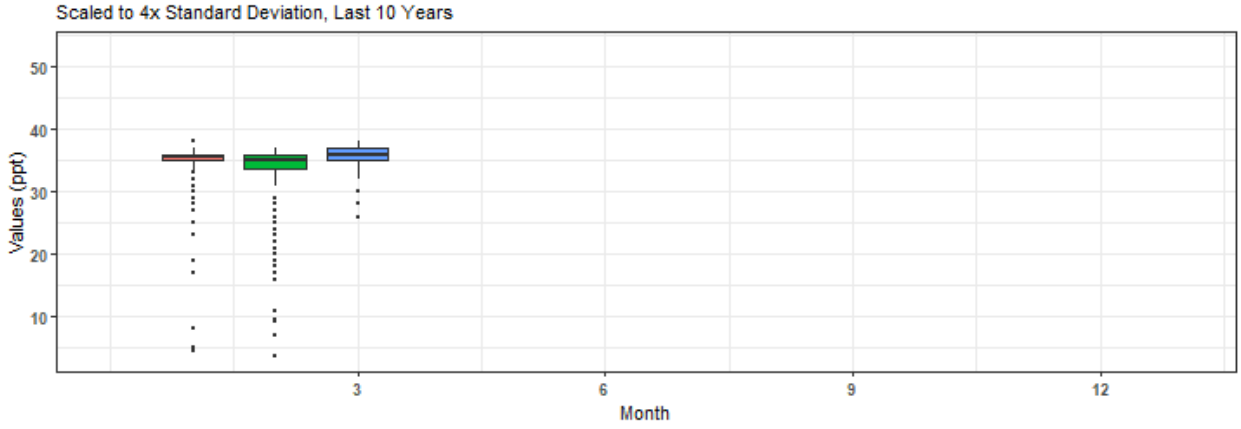


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
7 | National Water Information System | 255654081350200

By Month



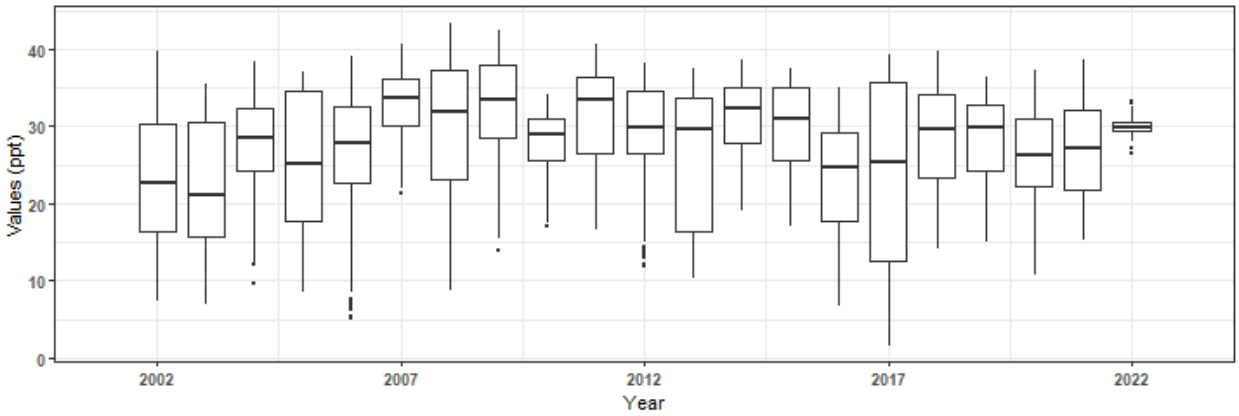




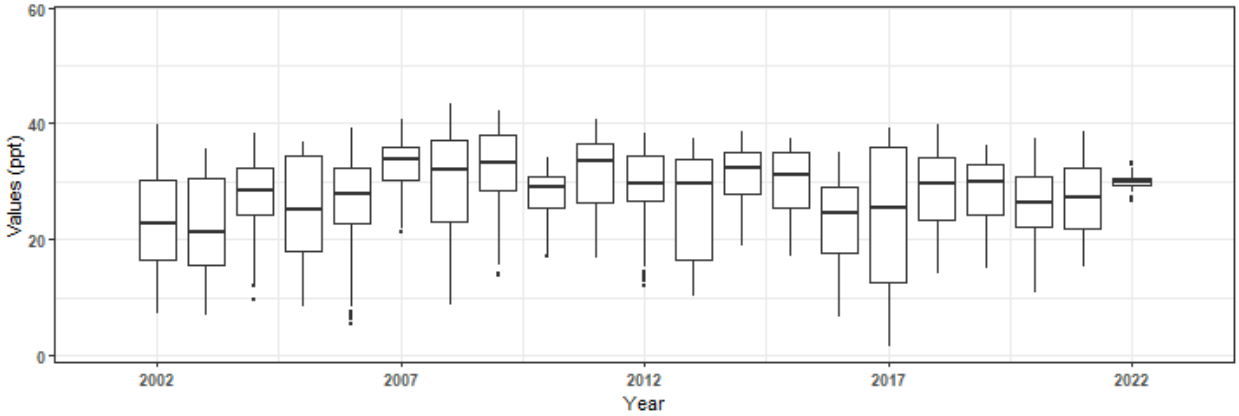
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

By Year

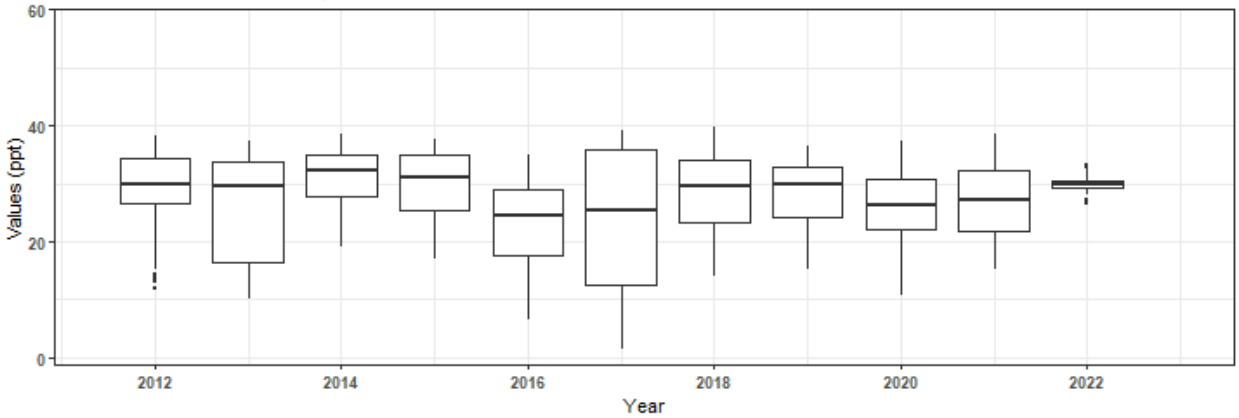
Autoscale



Scaled to 4x Standard Deviation

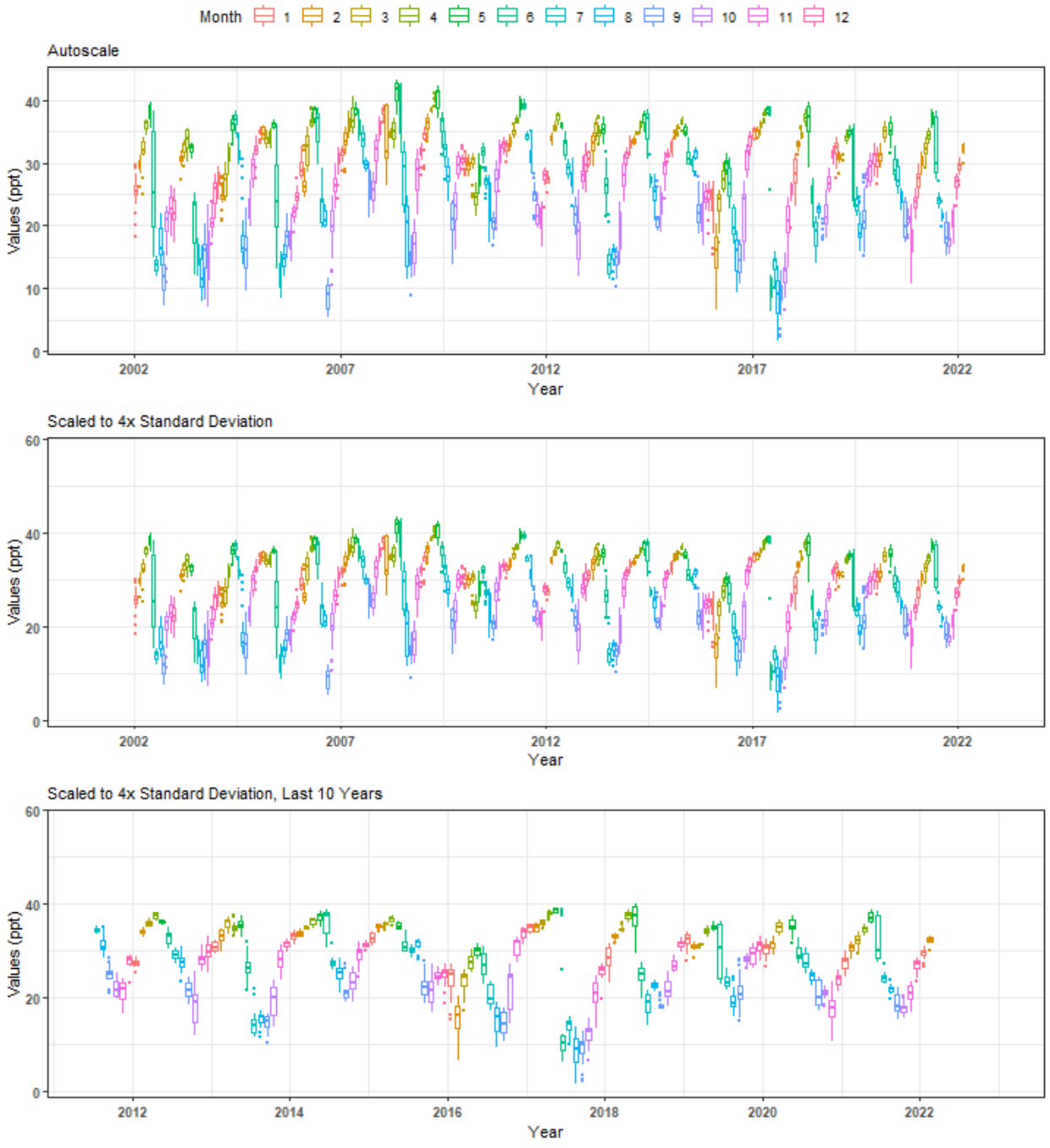


Scaled to 4x Standard Deviation, Last 10 Years



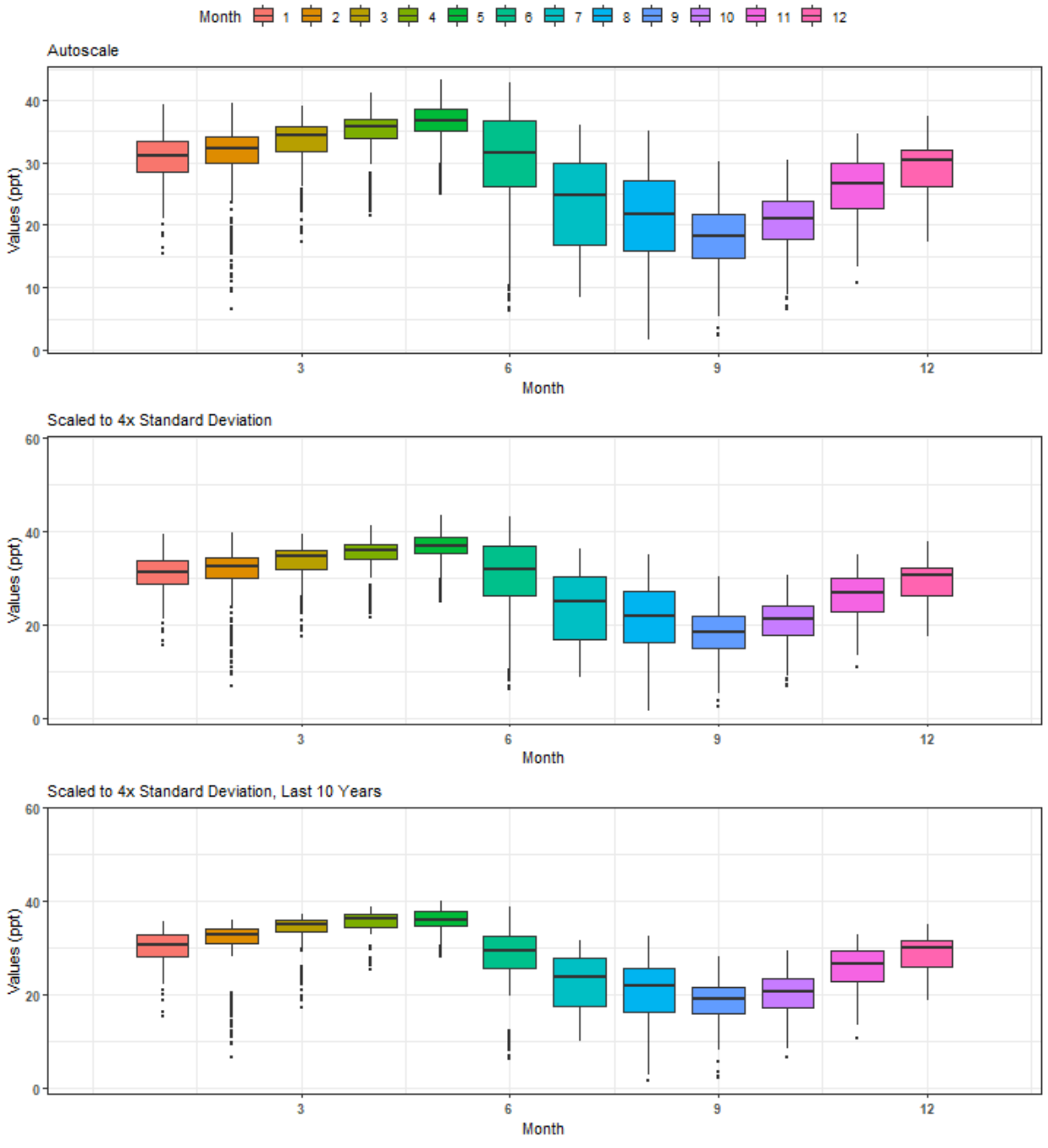
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

By Year & Month



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfbwq

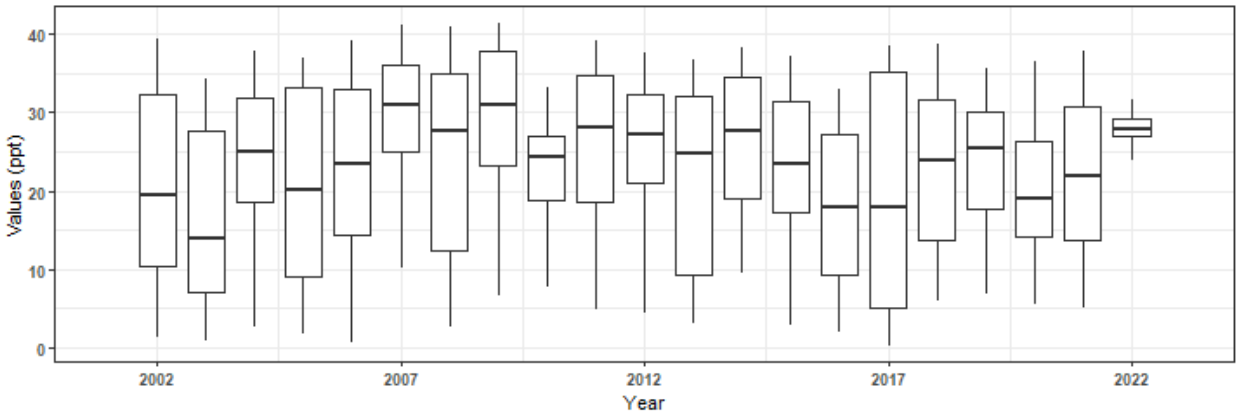
By Month



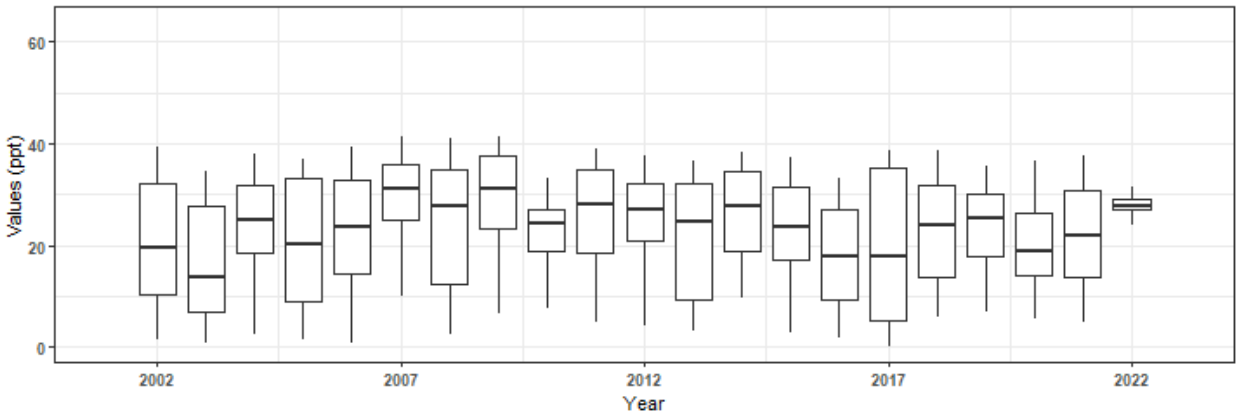
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

By Year

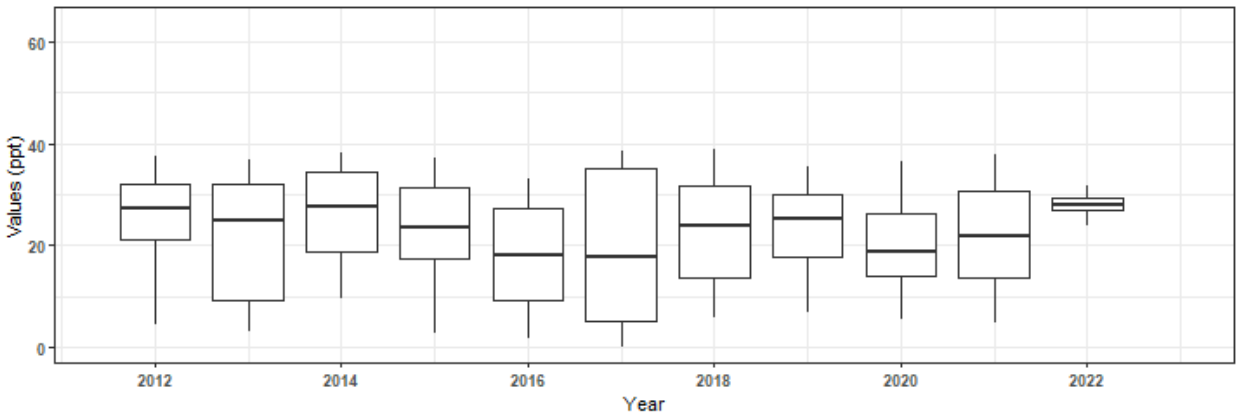
Autoscale



Scaled to 4x Standard Deviation

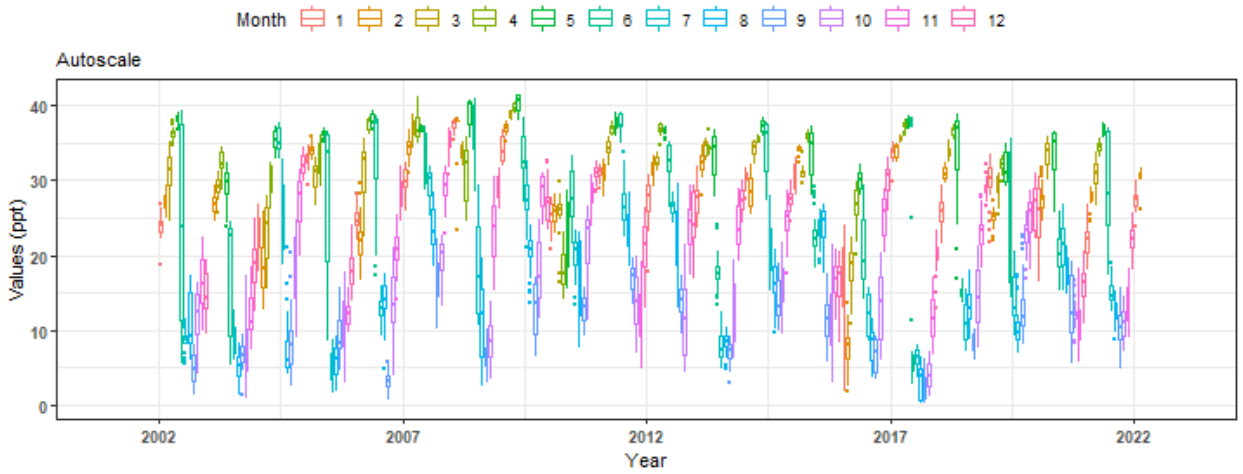


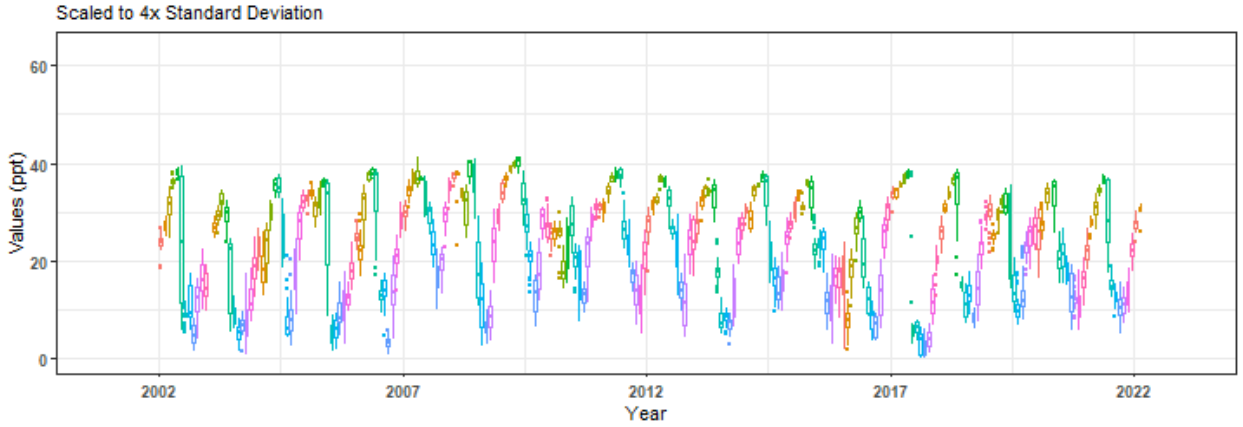
Scaled to 4x Standard Deviation, Last 10 Years

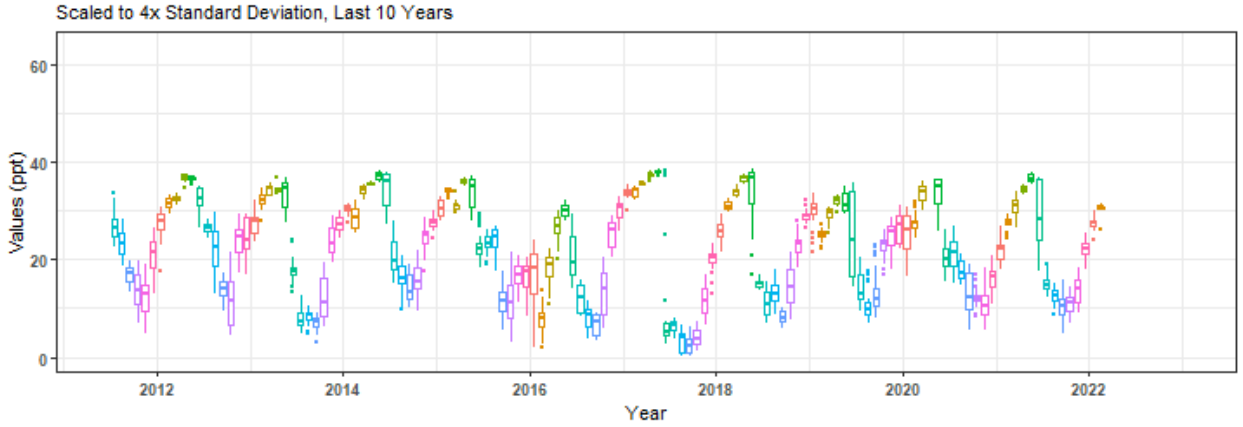


Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

By Year & Month

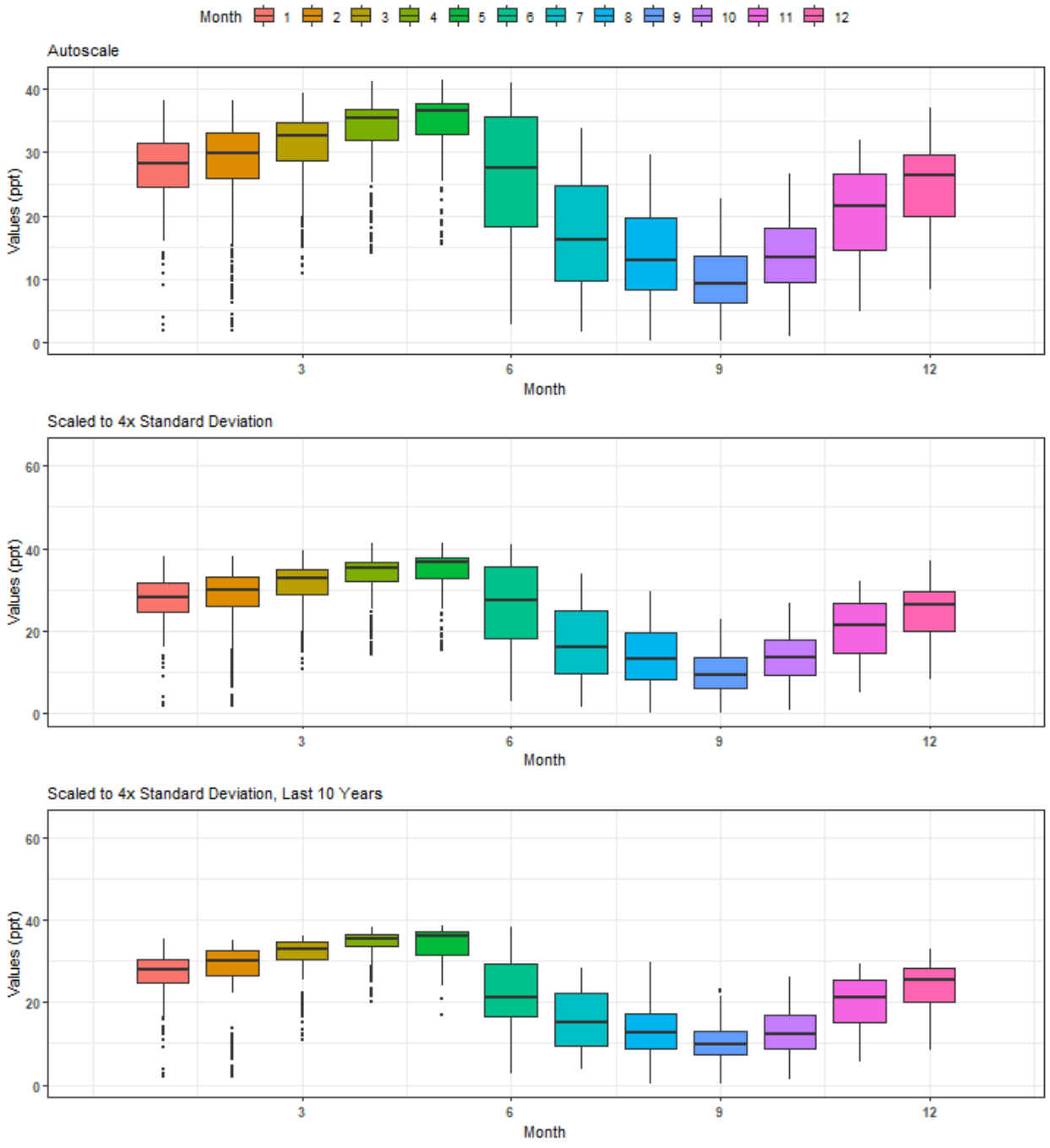






Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbfuwq

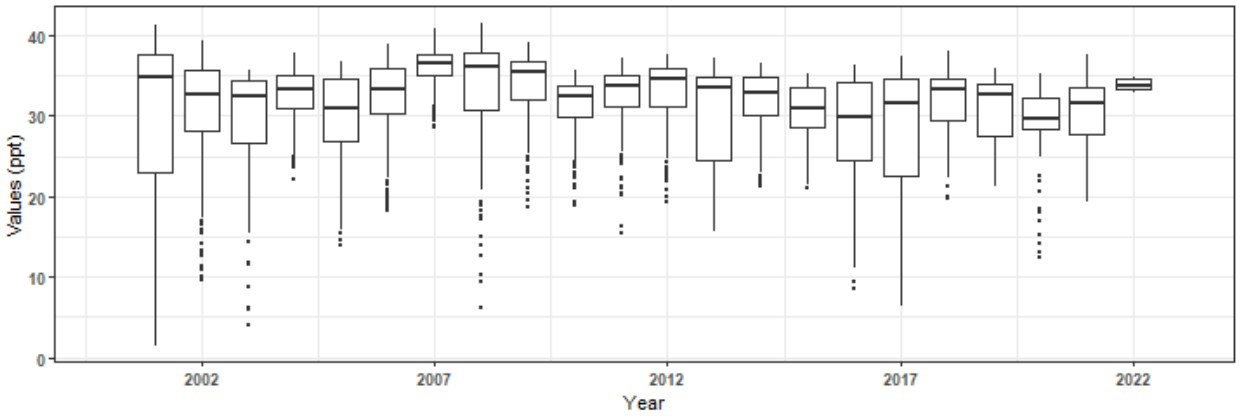
By Month



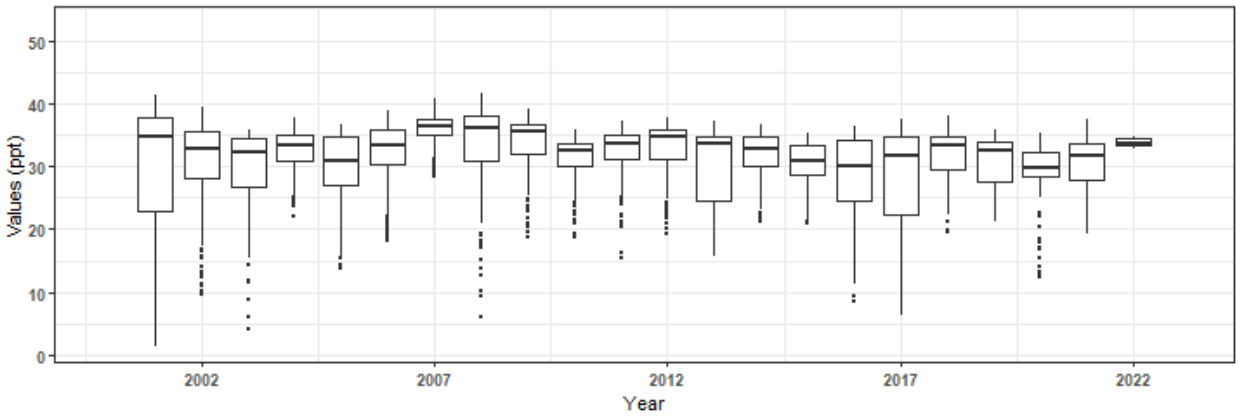
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

By Year

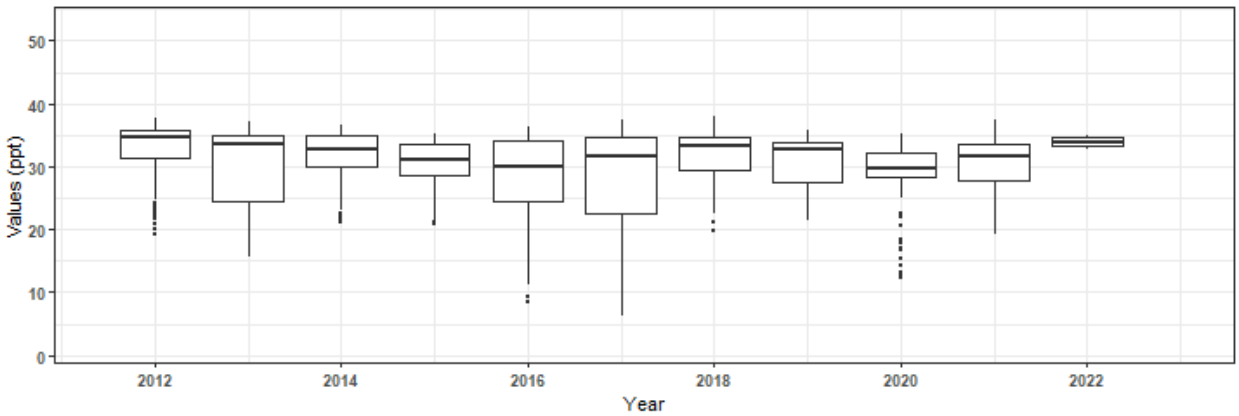
Autoscale



Scaled to 4x Standard Deviation

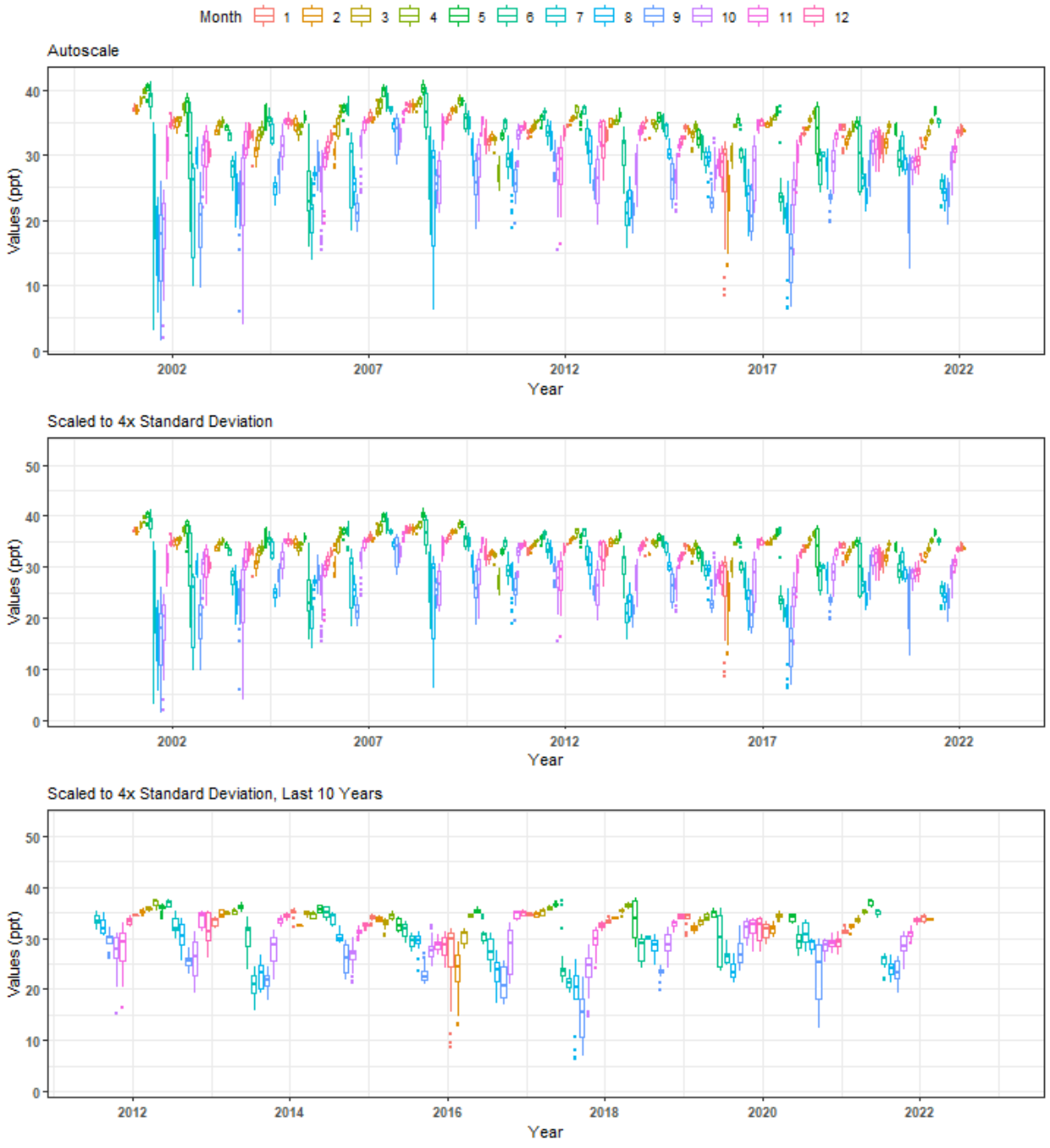


Scaled to 4x Standard Deviation, Last 10 Years



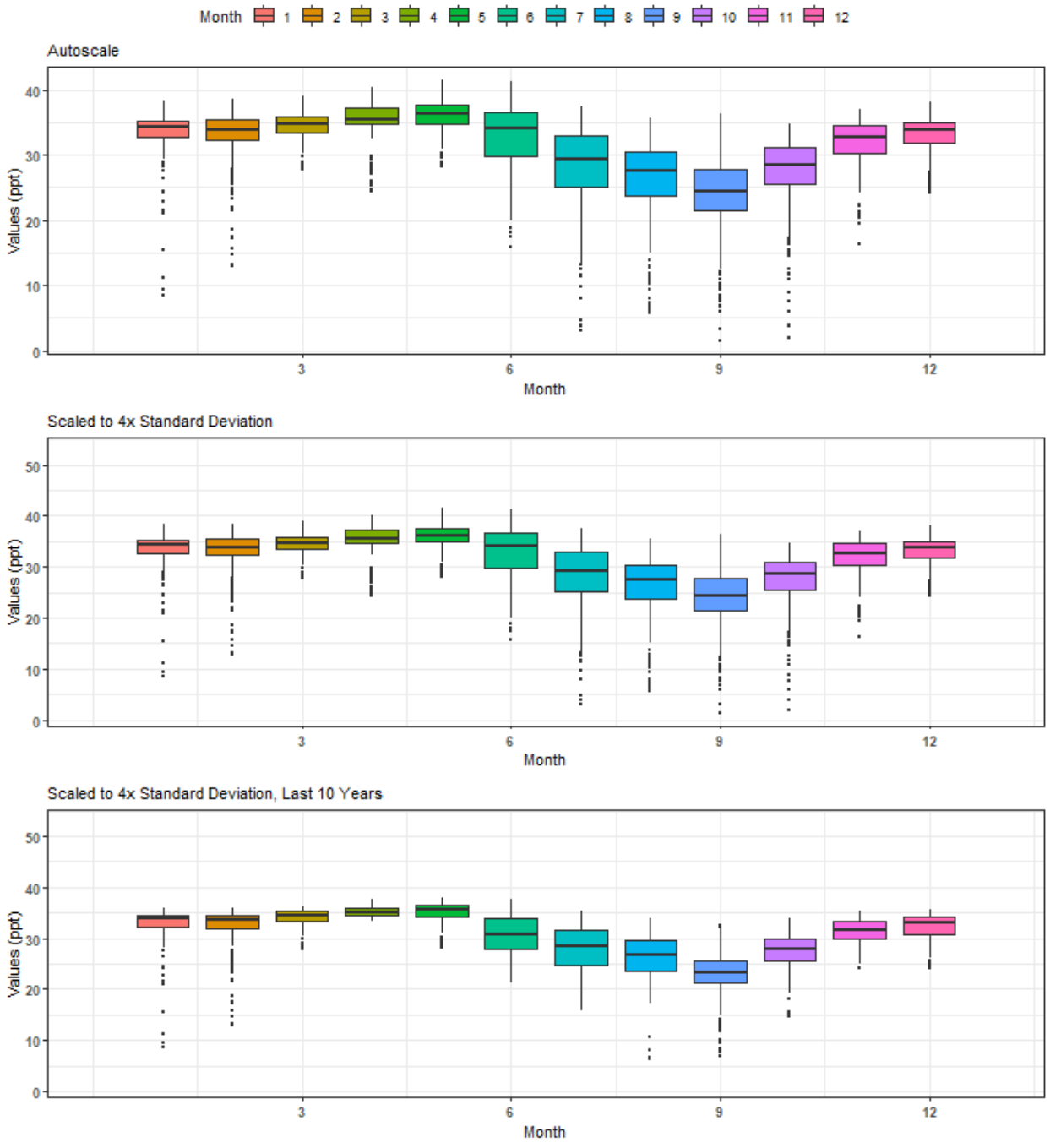
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

By Year & Month



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkblhwq

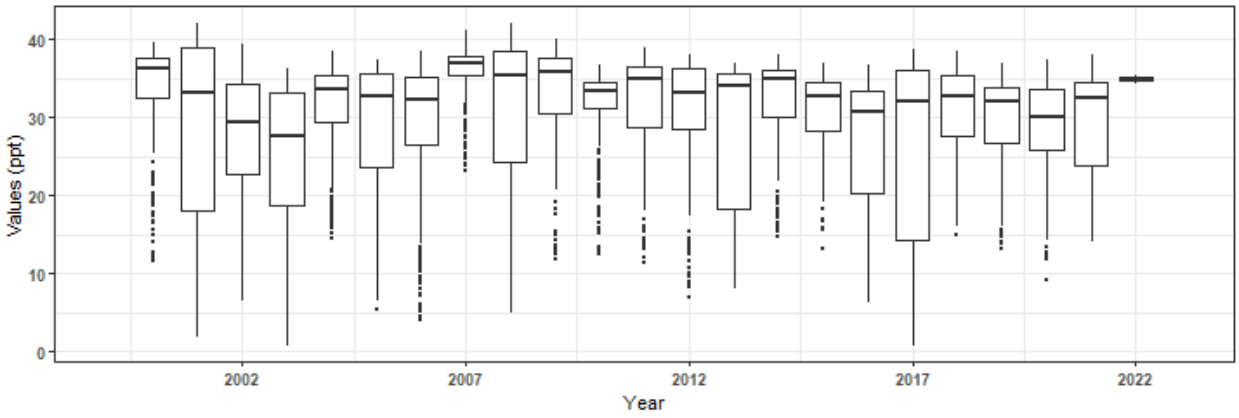
By Month



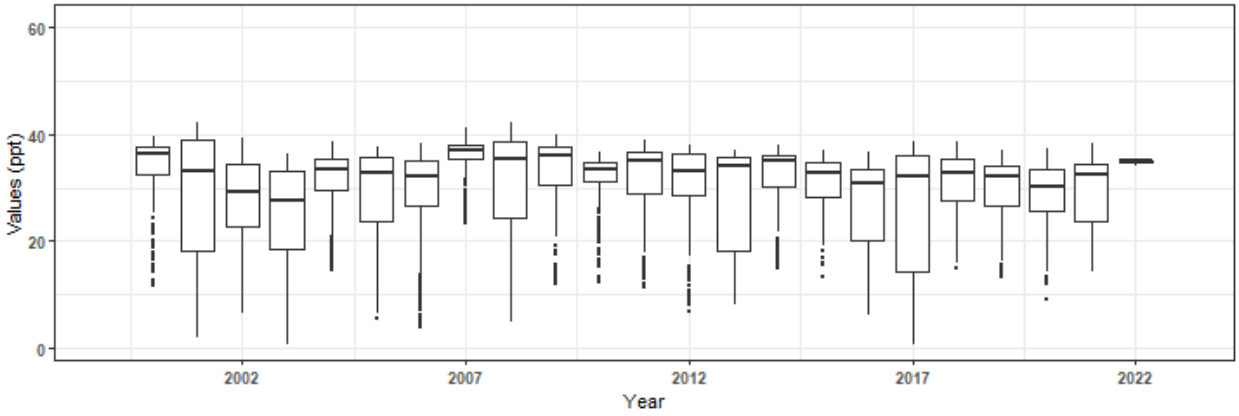
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

By Year

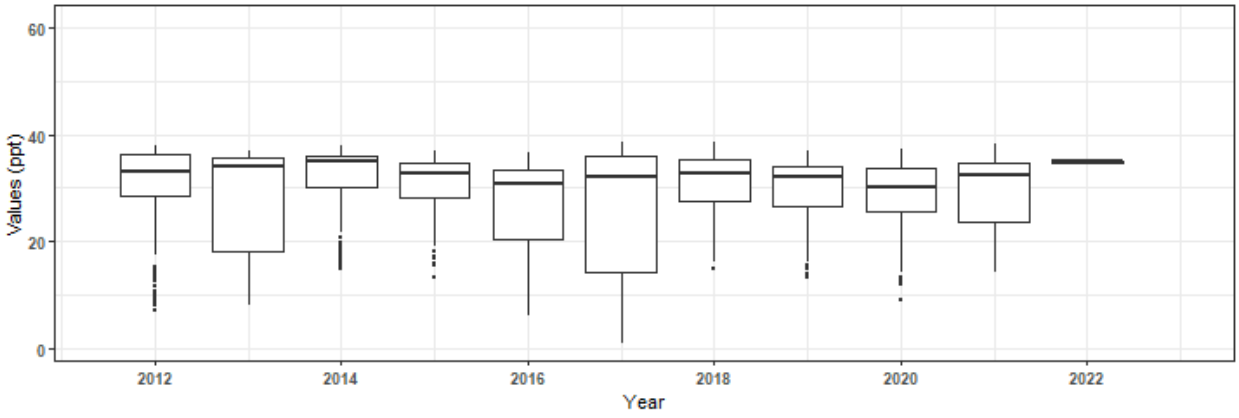
Autoscale



Scaled to 4x Standard Deviation

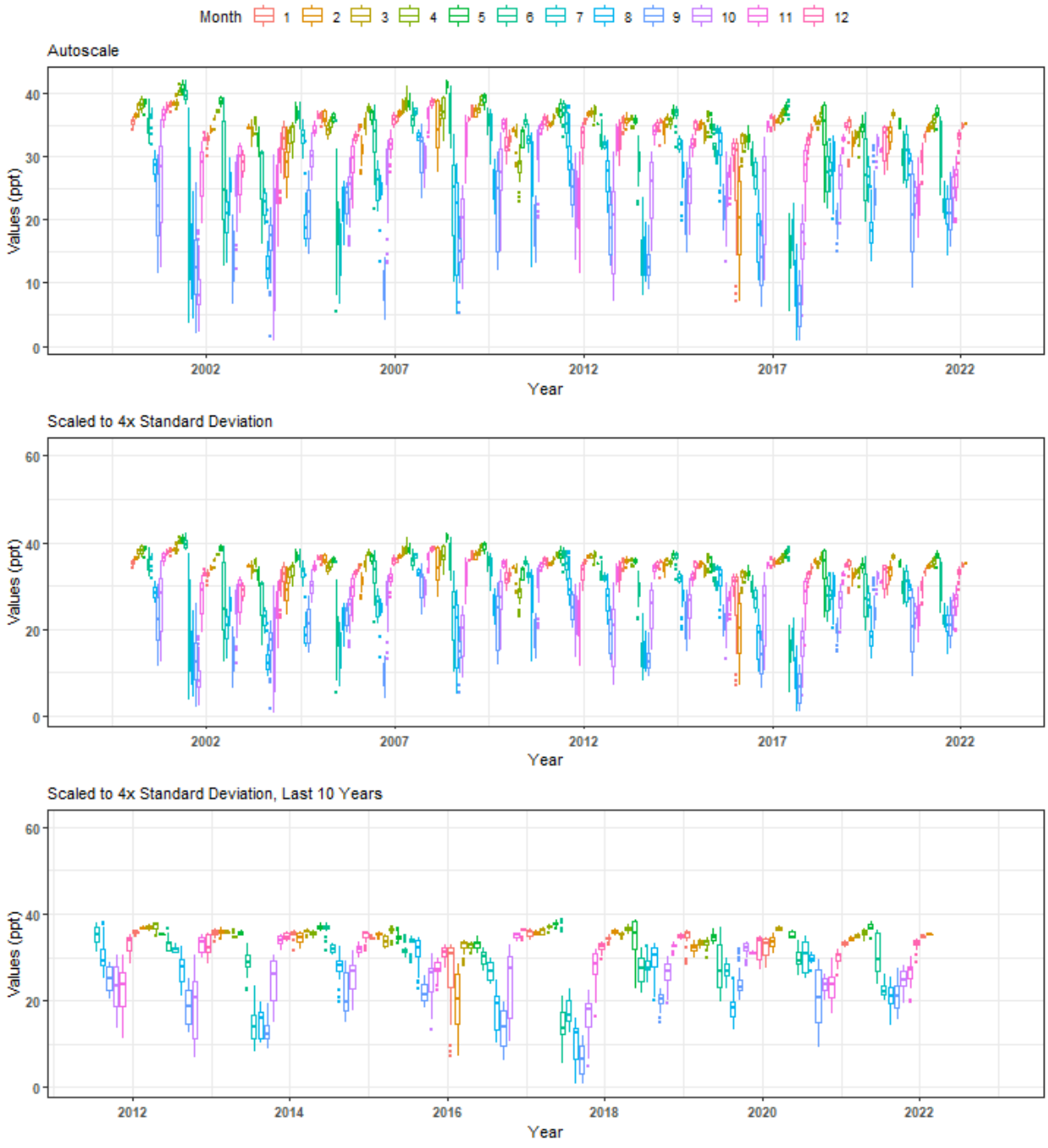


Scaled to 4x Standard Deviation, Last 10 Years



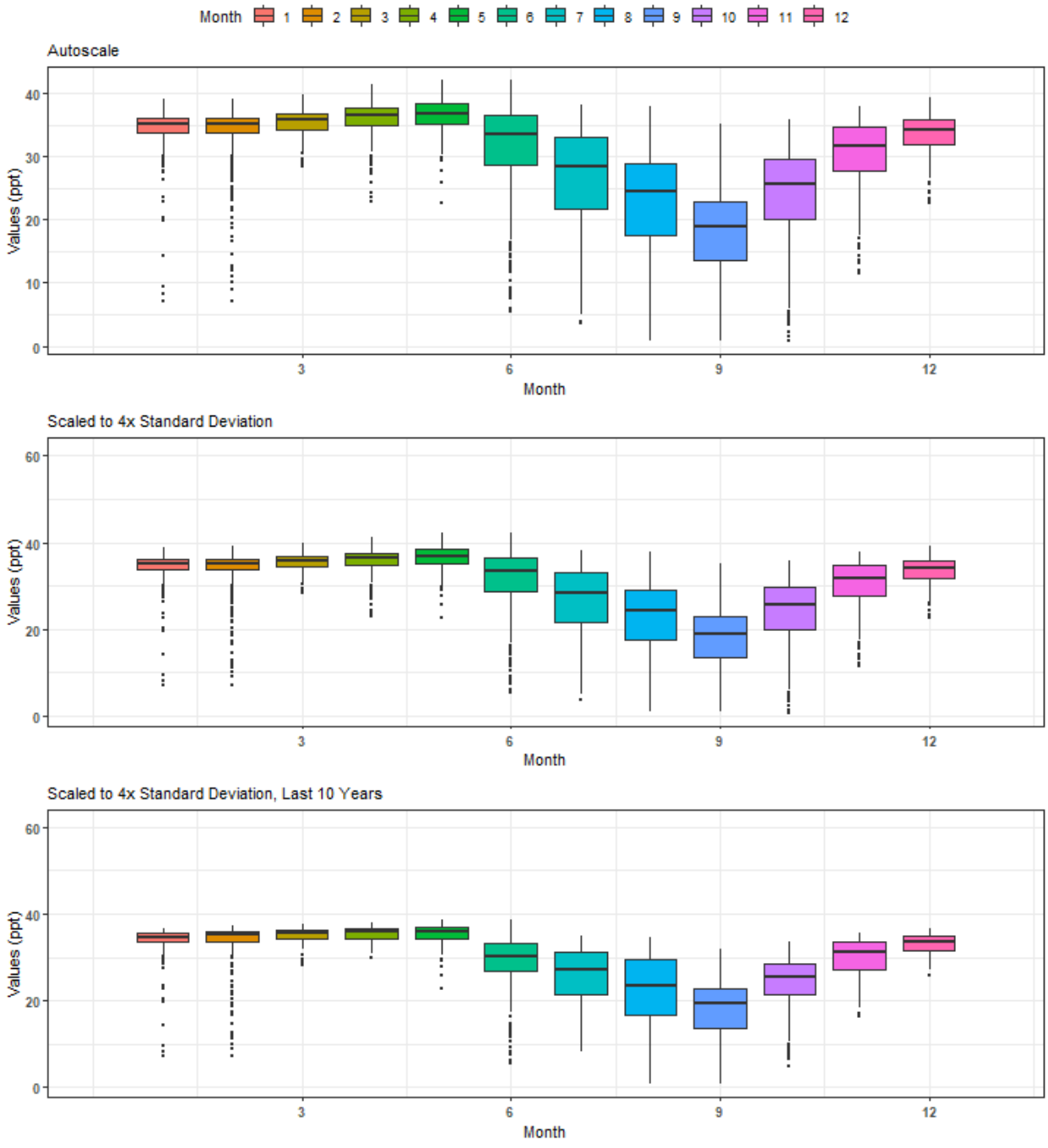
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

By Year & Month



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbmbwq

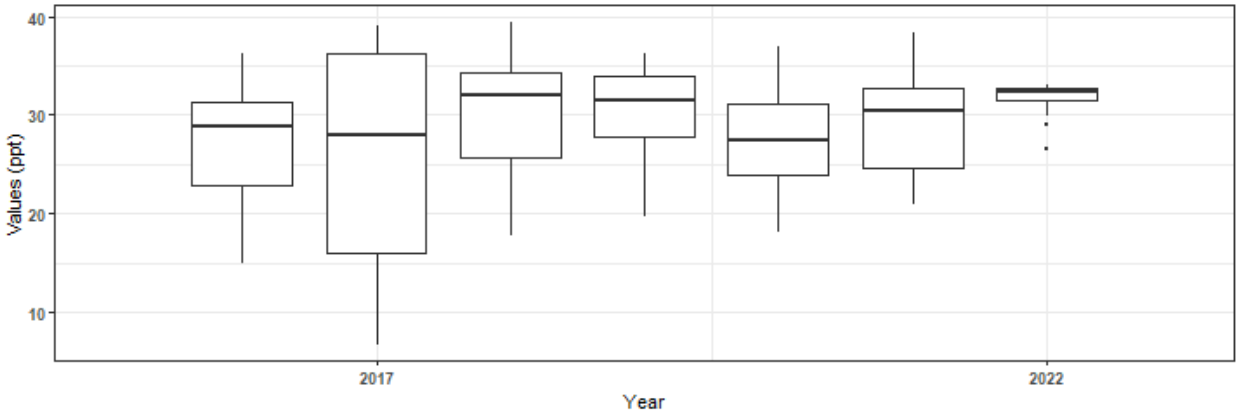
By Month



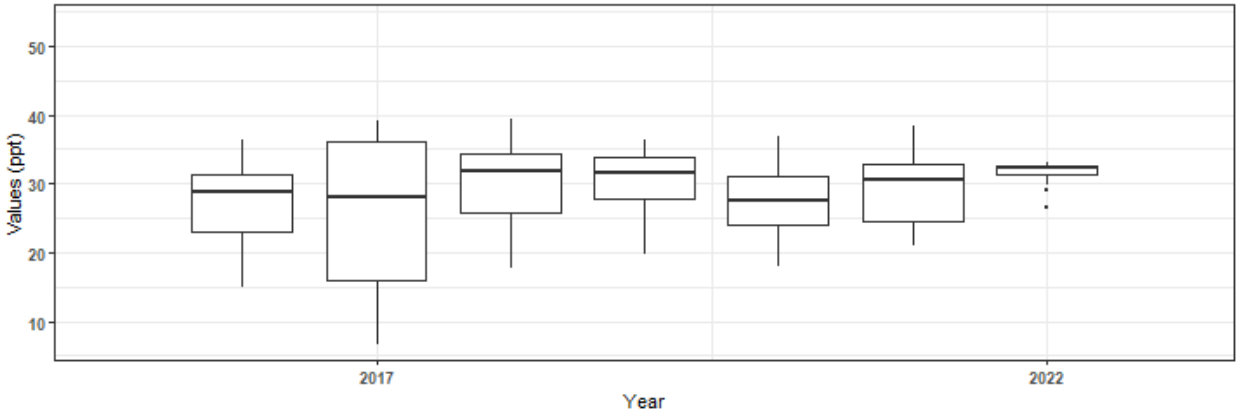
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

By Year

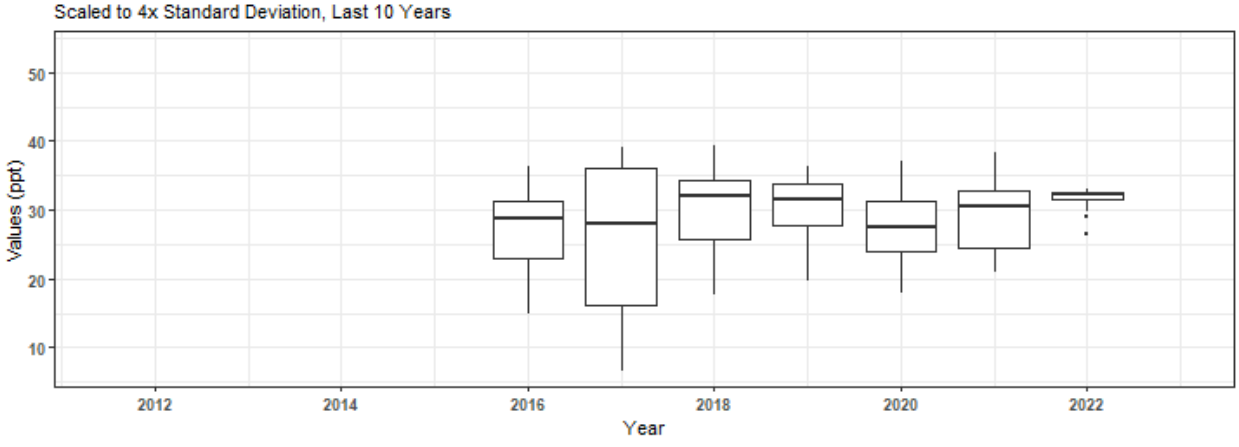
Autoscale



Scaled to 4x Standard Deviation

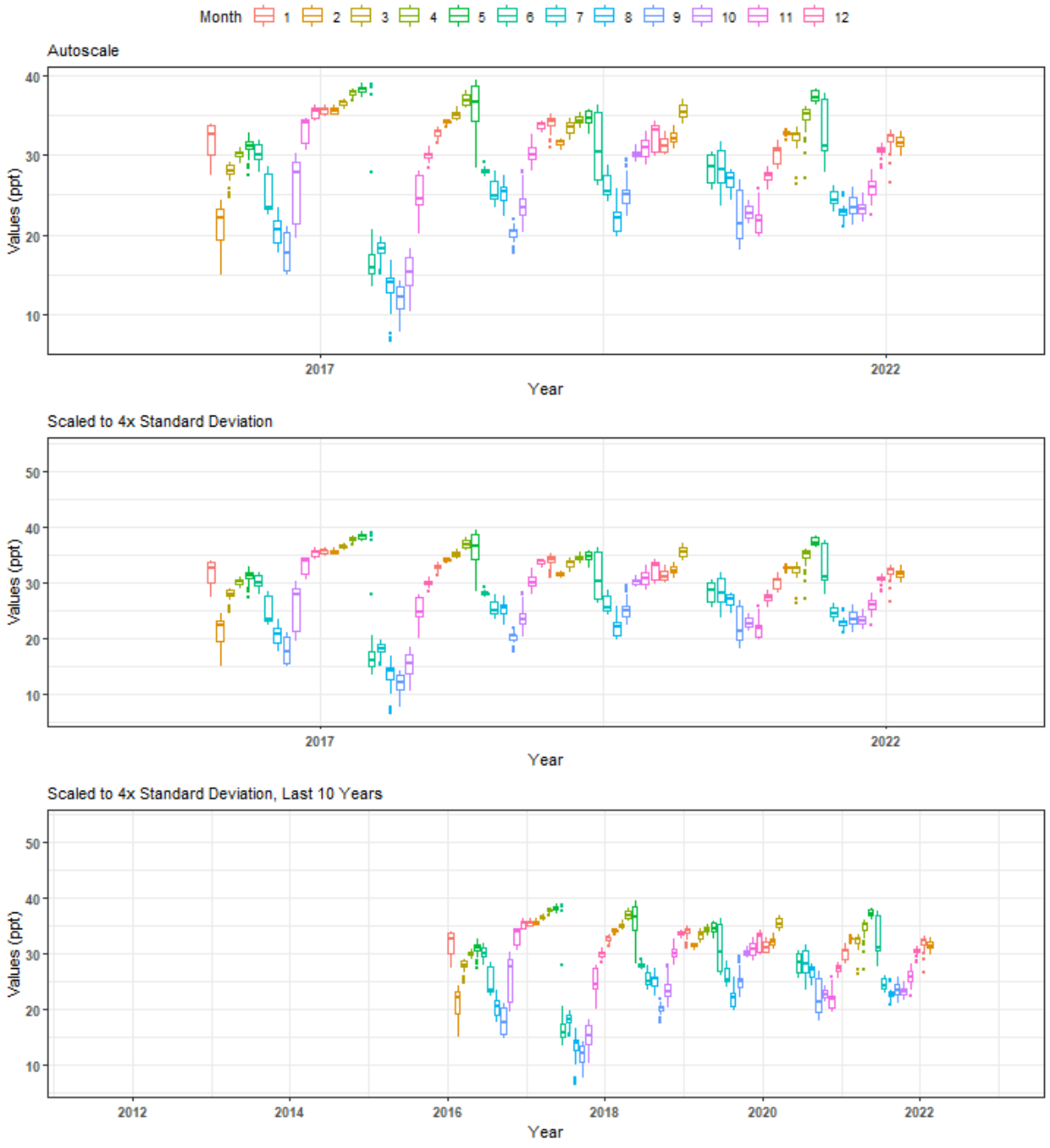


Scaled to 4x Standard Deviation, Last 10 Years



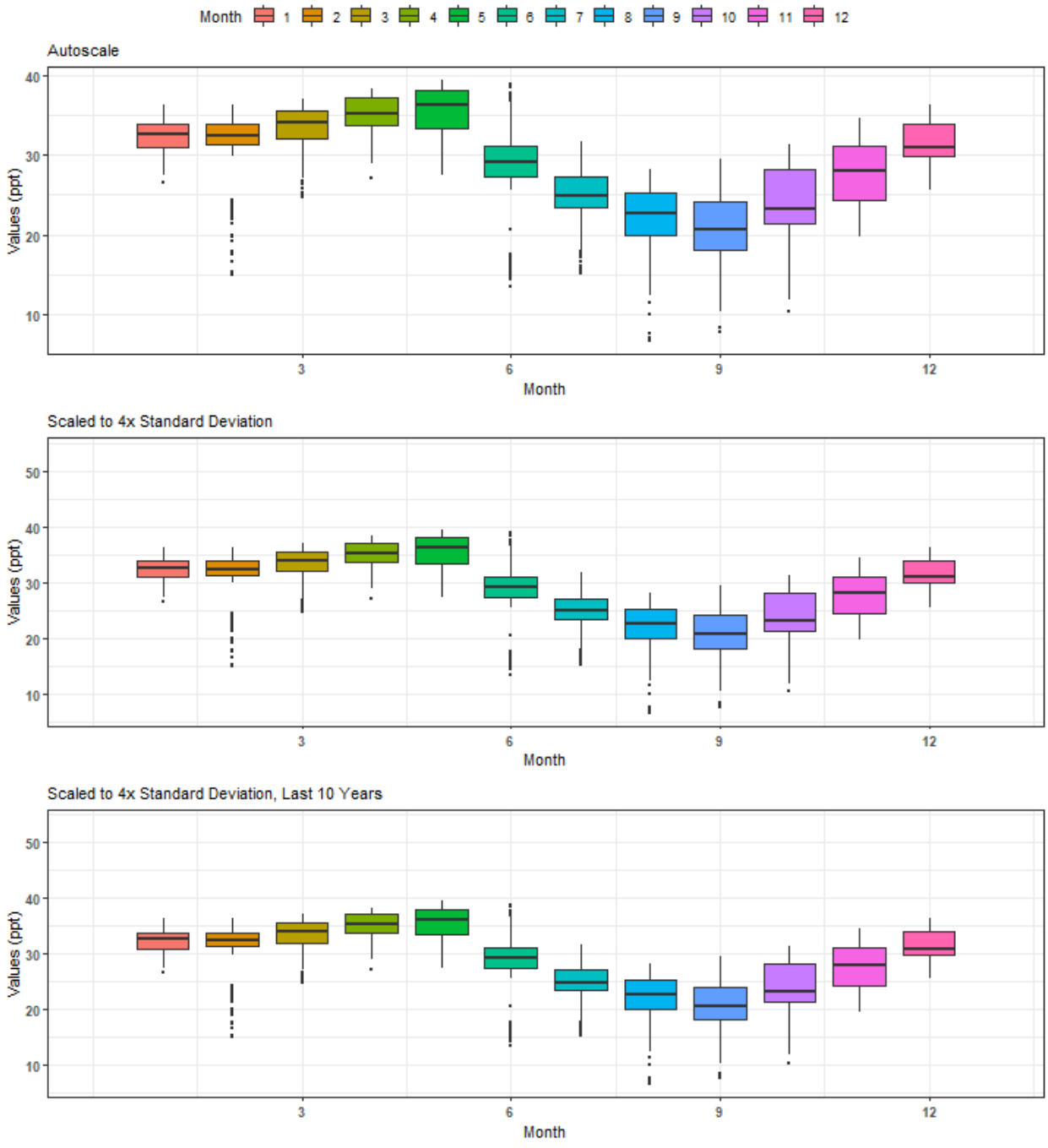
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

By Year & Month



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbpbwq

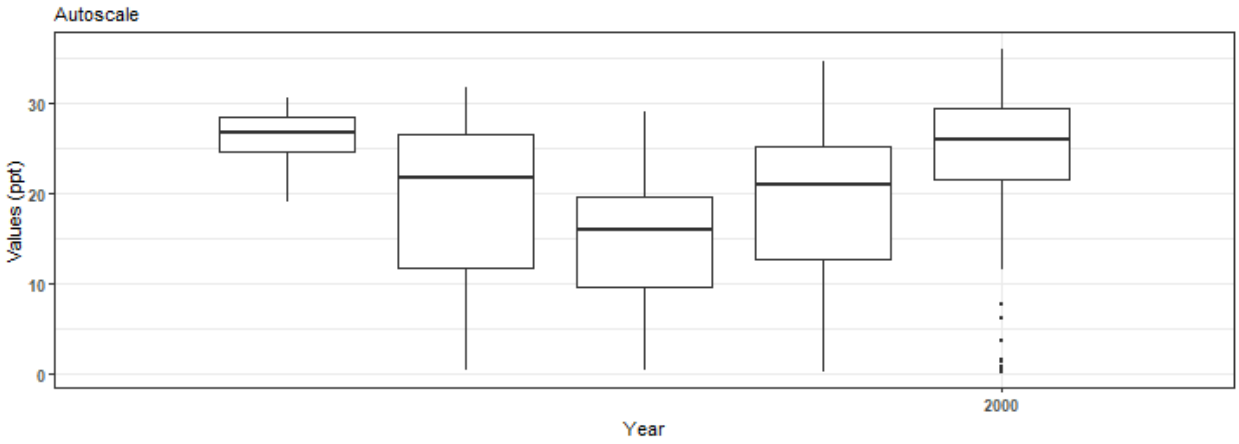
By Month



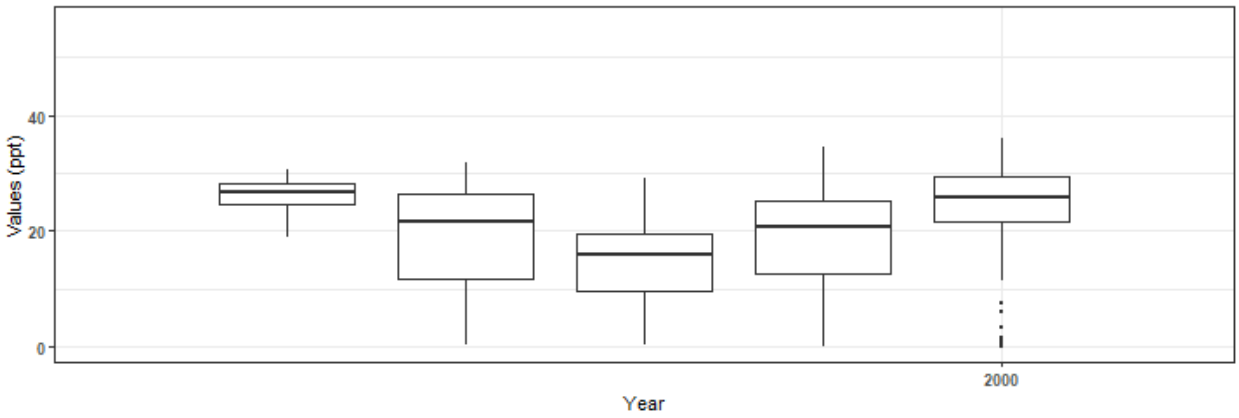
Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbuhwq

By Year

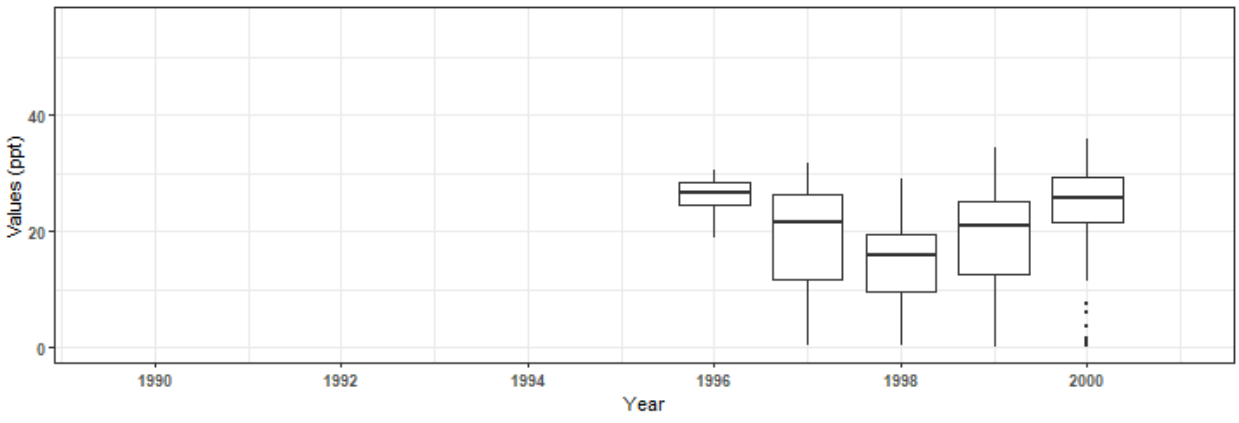
Autoscale



Scaled to 4x Standard Deviation

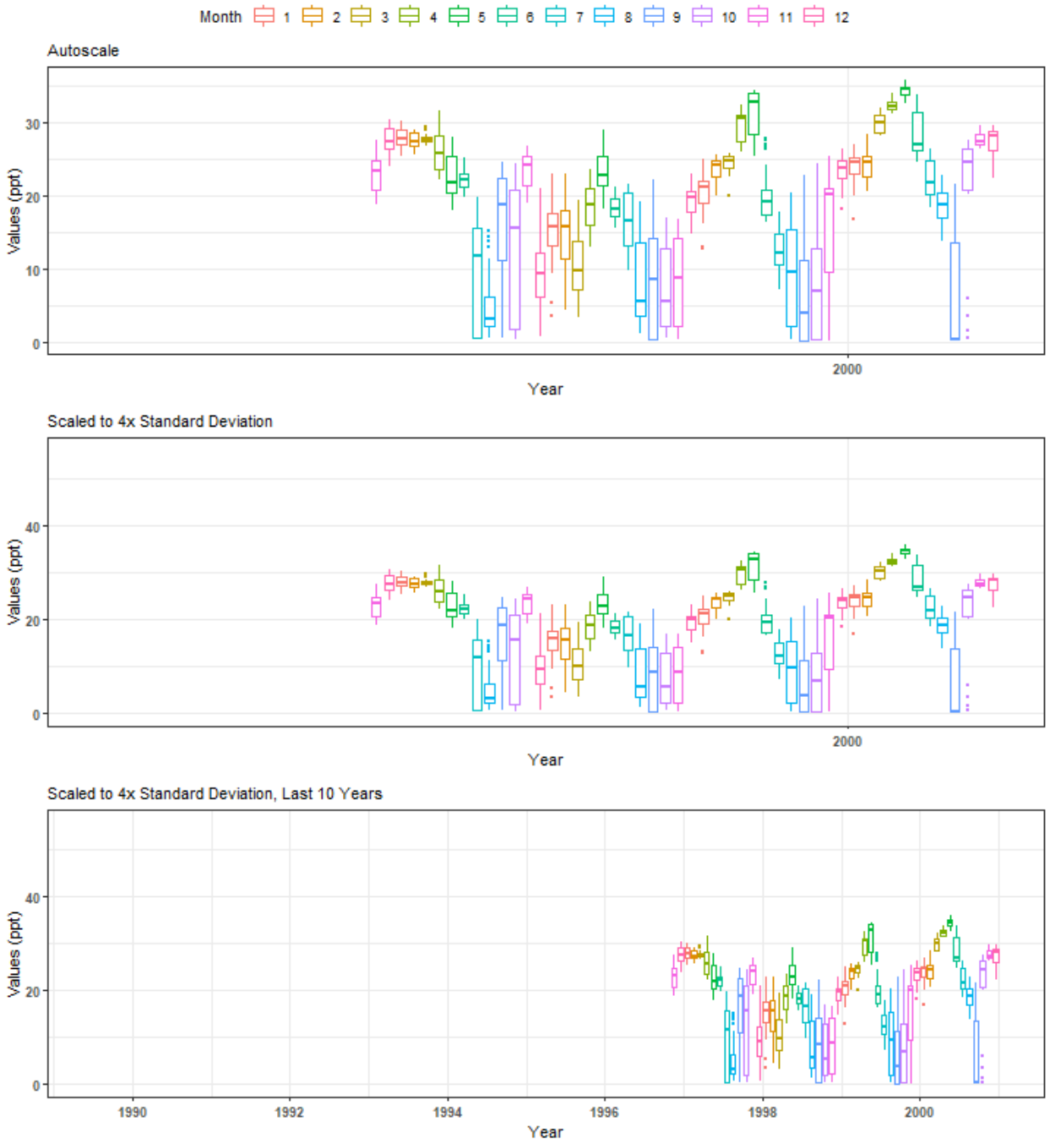


Scaled to 4x Standard Deviation, Last 10 Years



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbuhwq

By Year & Month



Summary Box Plots for Rookery Bay National Estuarine Research Reserve
354 | Rookery Bay National Estuarine Research Reserve System-Wide Monitoring Program | rkbuhwq

By Month