SEACAR Discrete Water Quality Analysis: Field Bottom Water Temperature

Last compiled on 24 June, 2022

Contents

Important Notes

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

https://github.com/FloridaSEACAR/SEACAR_Panzik

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

Libraries

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

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

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

File Import

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

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

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

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

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

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

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

Data Filtering and Data Impacted by Specific Value Qualifiers

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

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

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

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

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

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

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

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

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

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

```
}

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


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

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

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

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

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


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

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

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

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

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

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

```
data$VQ_Plot <- data$ValueQualifier

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

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


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

```
## Number of Measurements: 4471835, Number Passed Filter: 4471835
## I Codes: 0 (0%)
## Q Codes: 0 (0%)
## U Codes: 0 (0%)
```

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

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

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

Managed Area Statistics

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

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

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

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

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

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

Monitoring Location Statistics

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

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

    - Click this text to open Git directory with output files

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

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

Seasonal Kendall Tau Analysis

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

The following steps are performed:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

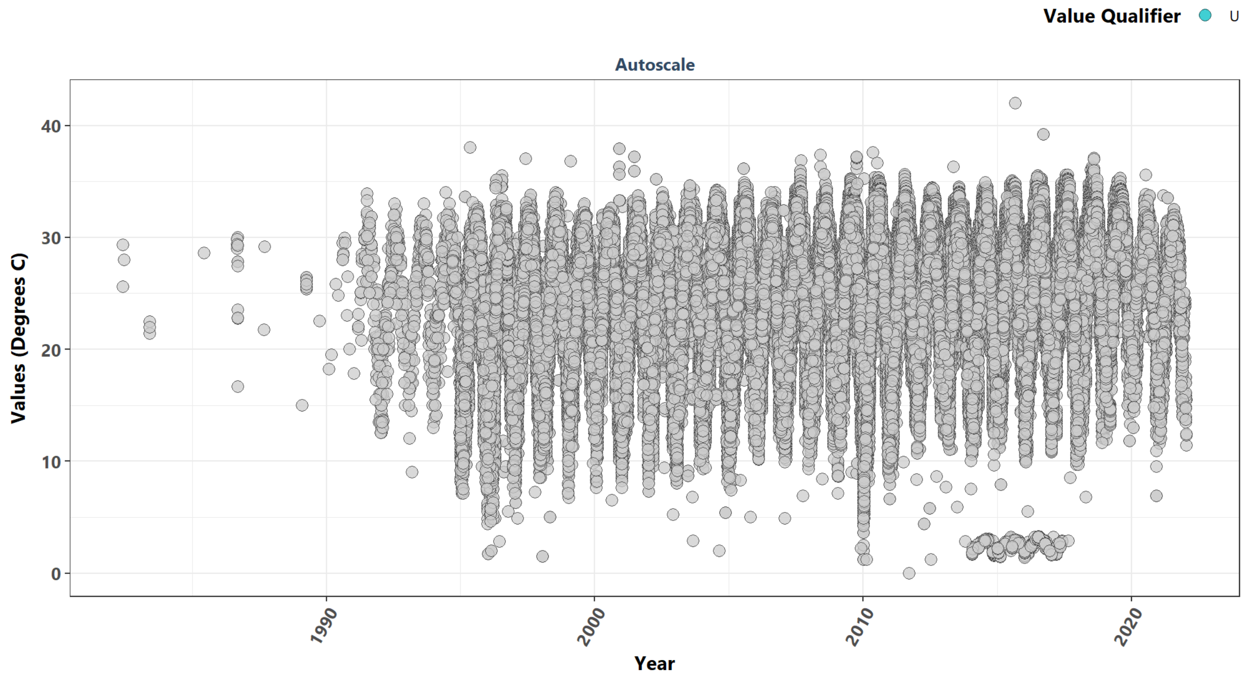
Appendix I: Scatter Plot of Entire Dataset

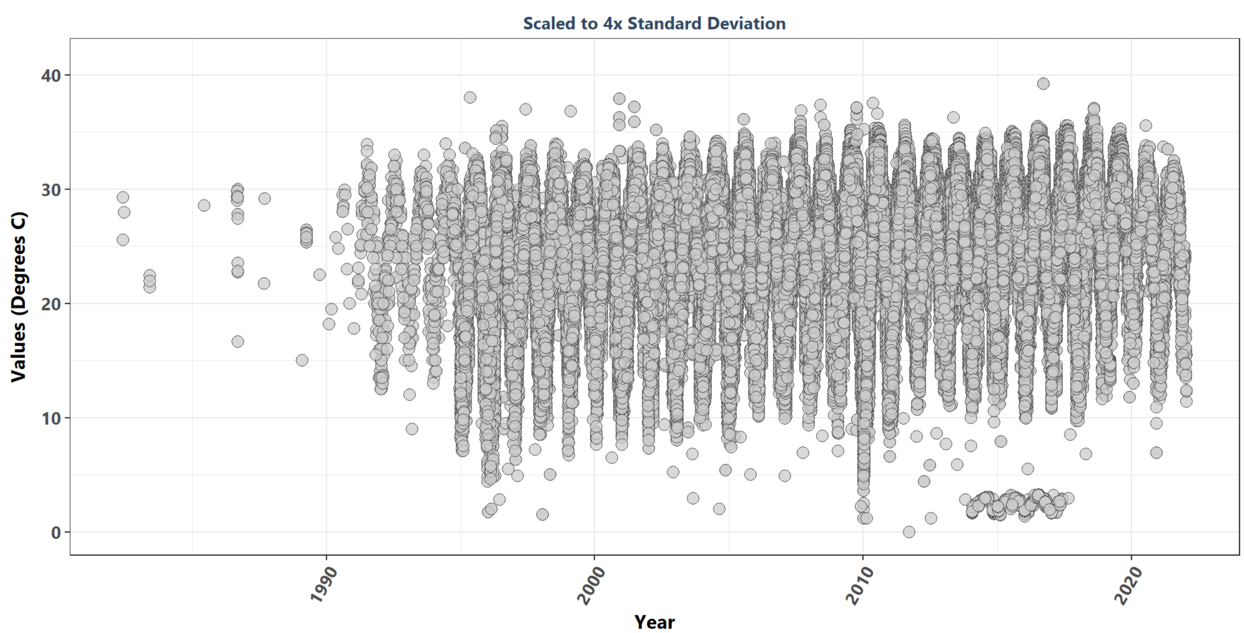
This part will create a scatter plot of the all data that passed initial filtering criteria with points colored based on specific value qualifiers. The values determined at the beginning (`year_lower`, `year_upper`, `min_RV`, `mn_RV`, `x_scale`, and `y_scale`) are solely for use by the plotting functions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

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

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


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

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


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

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

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

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

Scatter Plot for Entire Dataset





Appendix II: Dataset Summary Box Plots

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

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

This set of box plots are grouped by year.

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

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

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

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

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

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

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

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

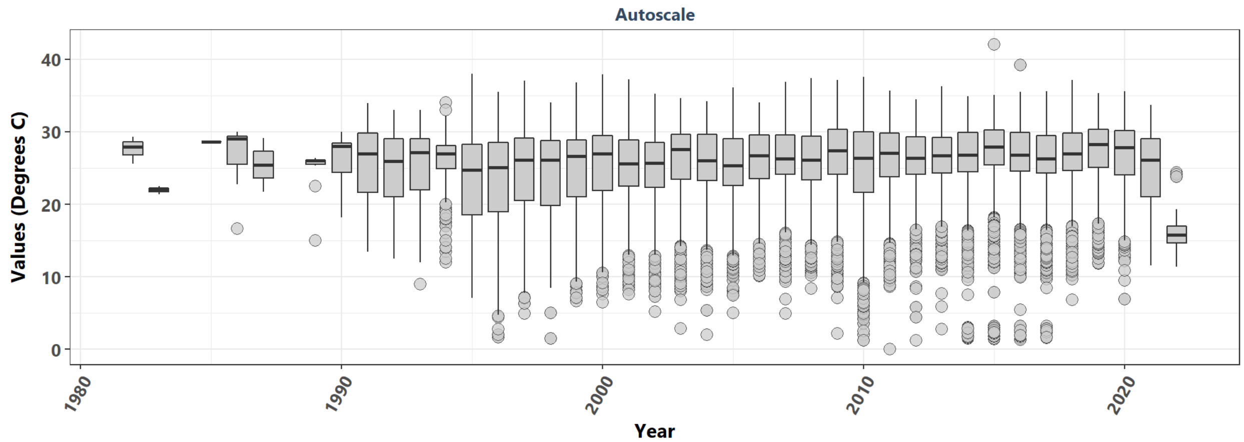
This set of box plots are grouped by year and month with the color being related to the month.

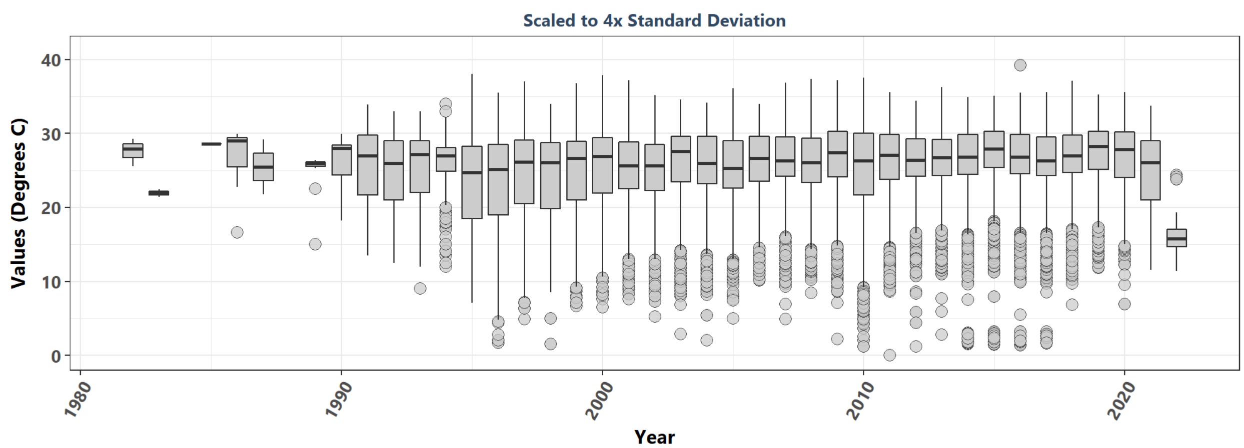
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit,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

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

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

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

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

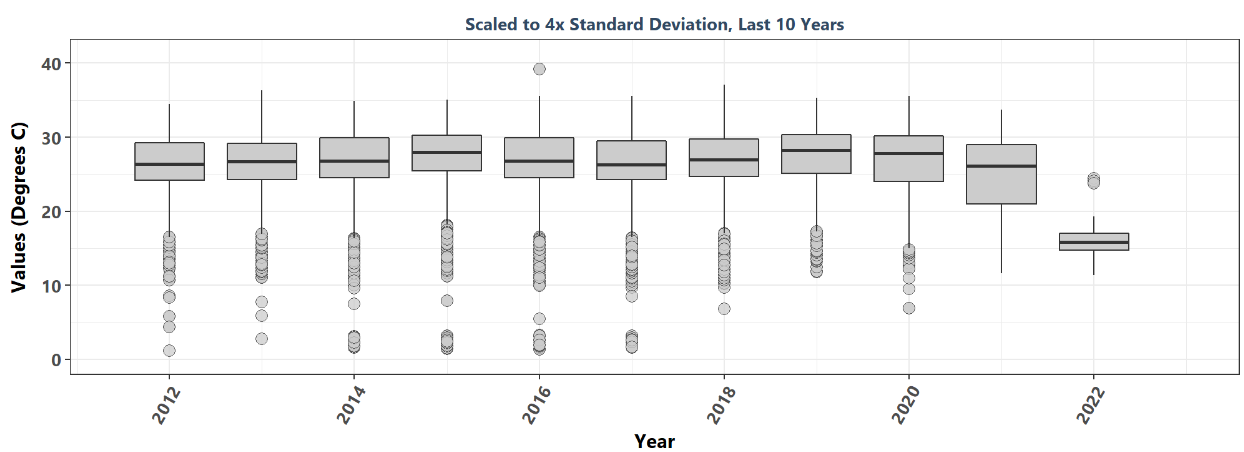
The following box plots are grouped by month with fill color being related to the month. This is designed to view potential seasonal trends.

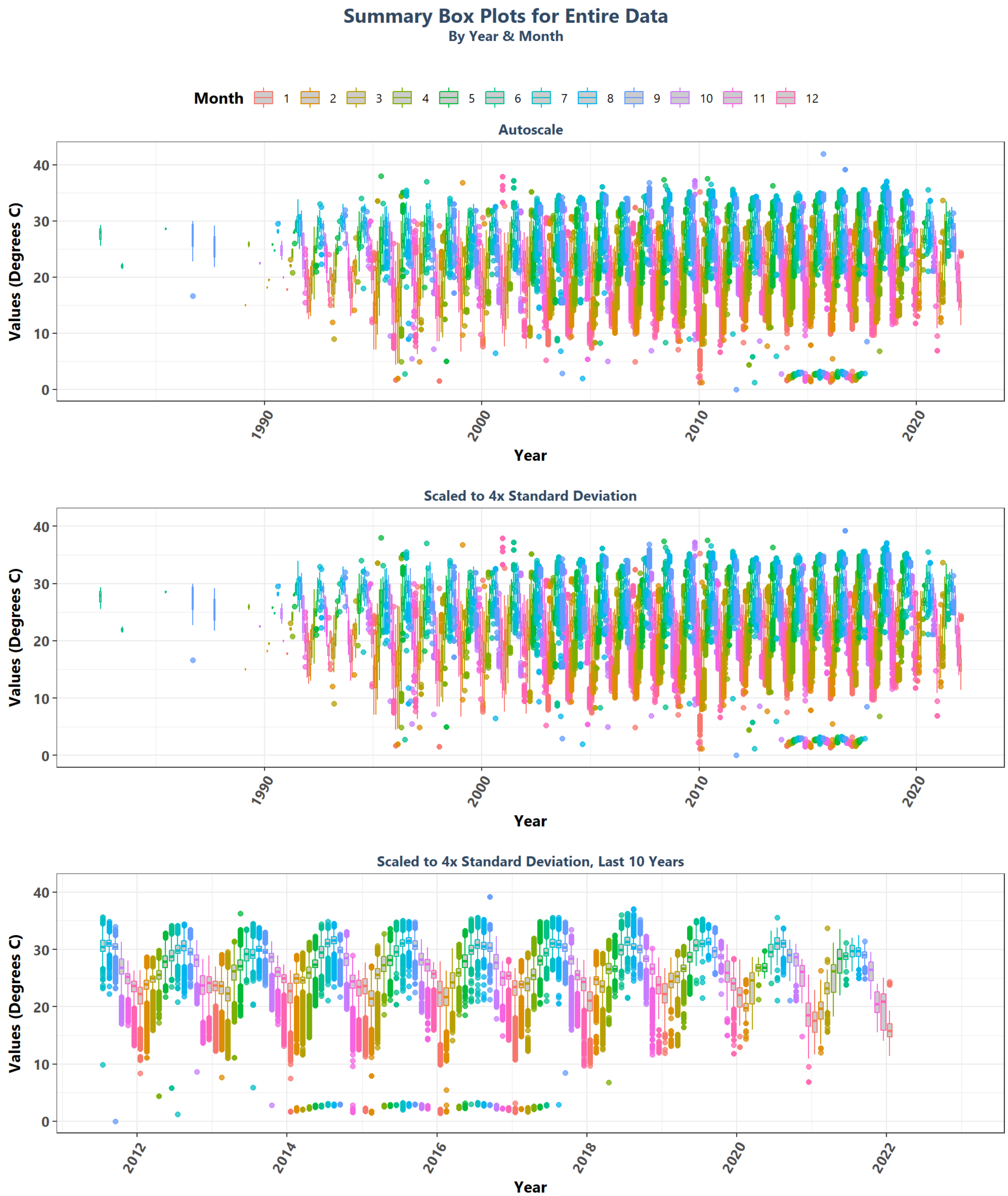
```r
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=Month, y=ResultValue,
                 group=Month, fill=as.factor(Month))) +
   geom_boxplot(color="#333333", outlier.shape=21, outlier.size=3,
                outlier.color="#333333", outlier.alpha=0.75) +
   labs(subtitle="Autoscale",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

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

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

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

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

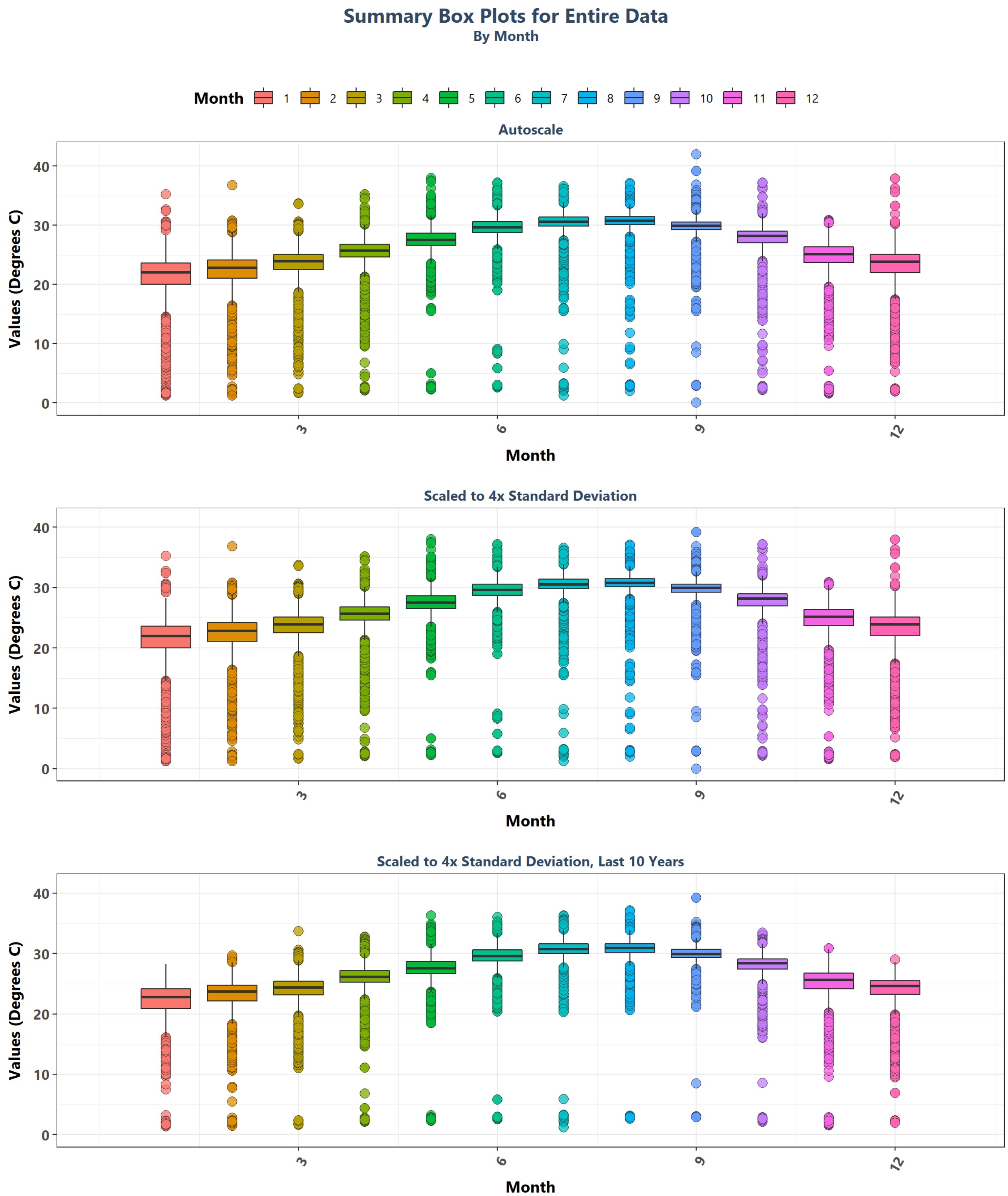
Summary Box Plots for Entire Data
By Year







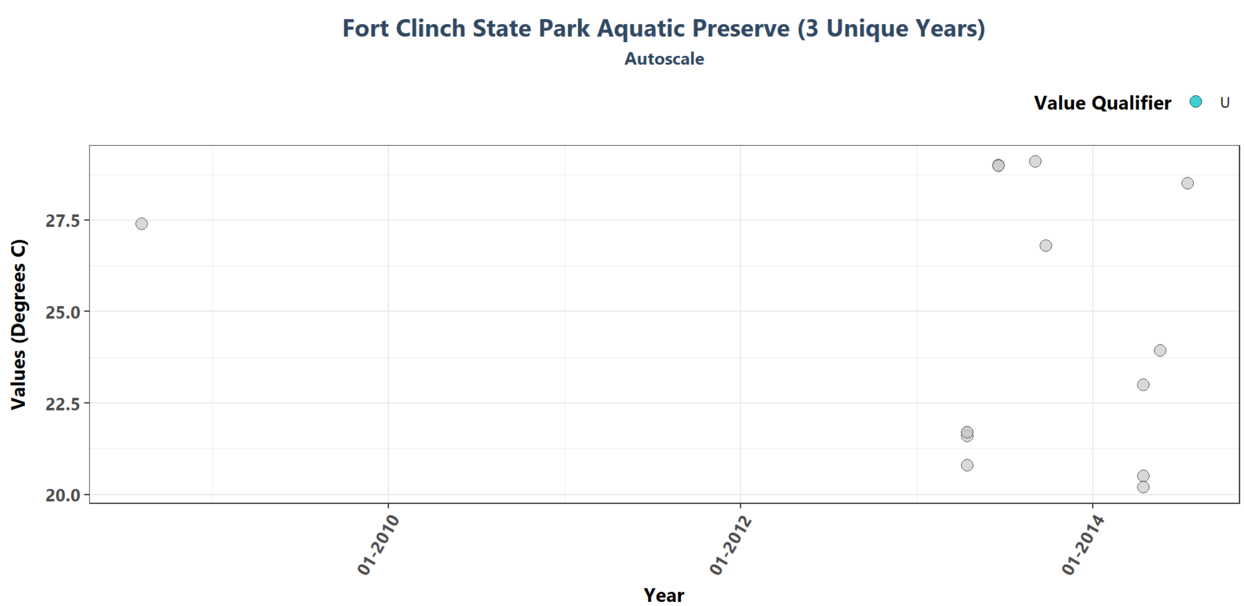

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
40
30
20
10
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
0
1990
2000
2010
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
0
2012
2014
2016
2018
2020
2022
Year



Appendix III: Excluded Managed Areas

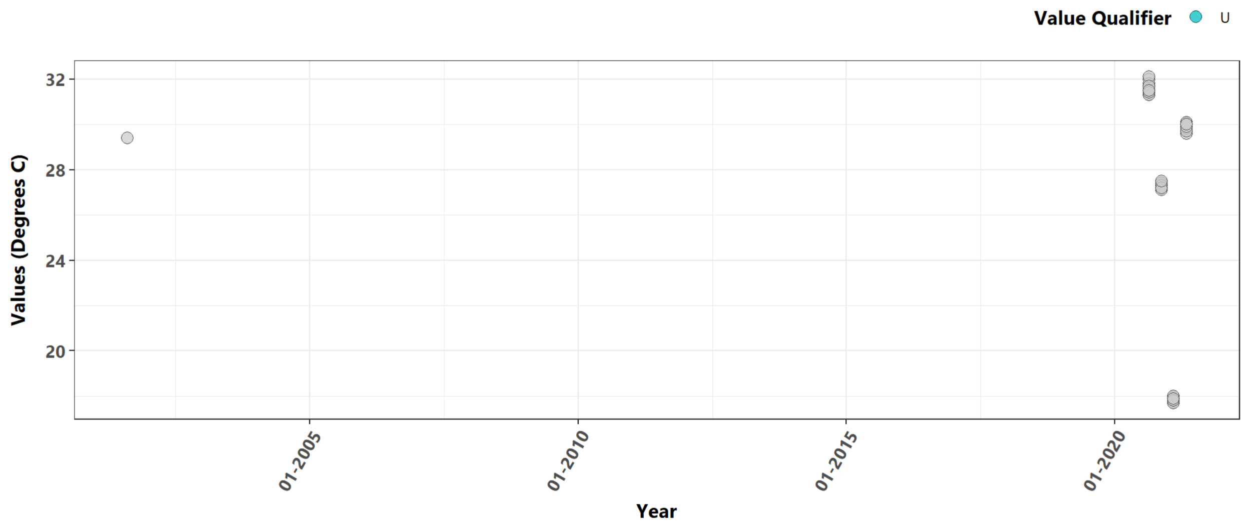
Scatter plots of data values are created for managed areas that have fewer than 10 separate years of data entries. Data points are colored based on specific value qualifiers of interest.

```r
if(z==0){
  print("There are no managed areas that qualify.")
} else {
  for(i in 1:z){
    p1<-ggplot(data=data[data$ManagedAreaName==MA_Exclude$ManagedAreaName[i]&
                           data$Include==TRUE, ],
               aes(x=SampleDate, y=ResultValue, fill=VQ_Plot)) +
      geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
      labs(title=paste0(MA_Exclude$ManagedAreaName[i], " (",
                        MA_Exclude$N_Years[i], " Unique Years)"),
           subtitle="Autoscale", x="Year",
           y=paste0("Values (", unit, ")"), fill="Value Qualifier") +
      plot_theme +
      theme(legend.position="top", legend.box="horizontal",
            legend.justification="right") +
      scale_x_date(labels=date_format("%m-%Y")) +
      {if(inc_H==TRUE){
        scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                   "HU"="#7CAE00"), na.value="#cccccc")
      } else if(param_name=="Secchi_Depth"){
        scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                   "SU"="#7CAE00"), na.value="#cccccc")
      } else {
        scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
      }}
    print(p1)
  }
}
```



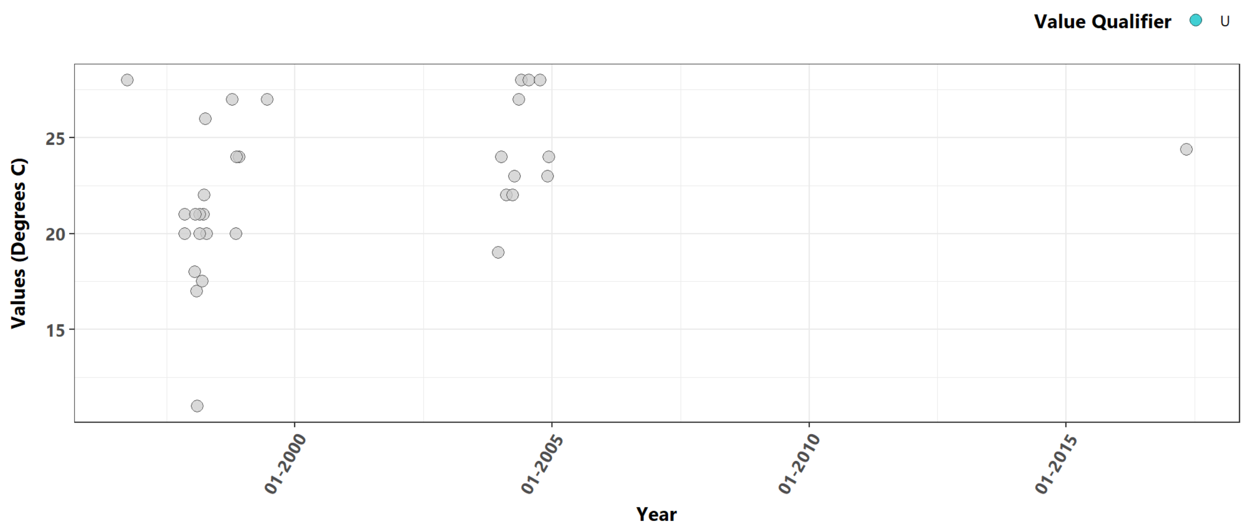
Lignumvitae Key Aquatic Preserve (3 Unique Years)

Autoscale



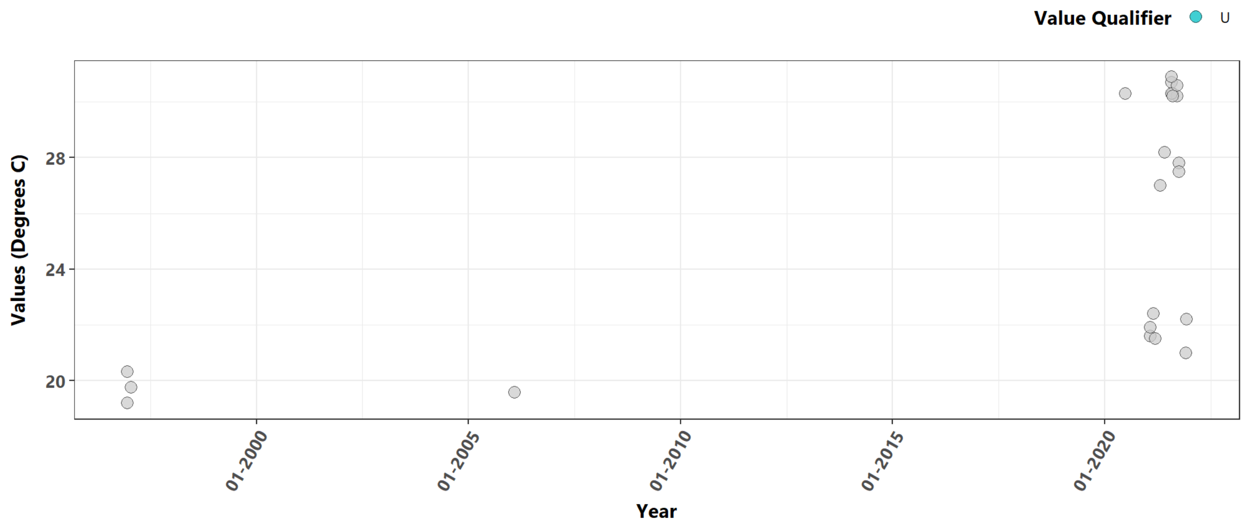
Loxahatchee River-Lake Worth Creek Aquatic Preserve (7 Unique Years)

Autoscale



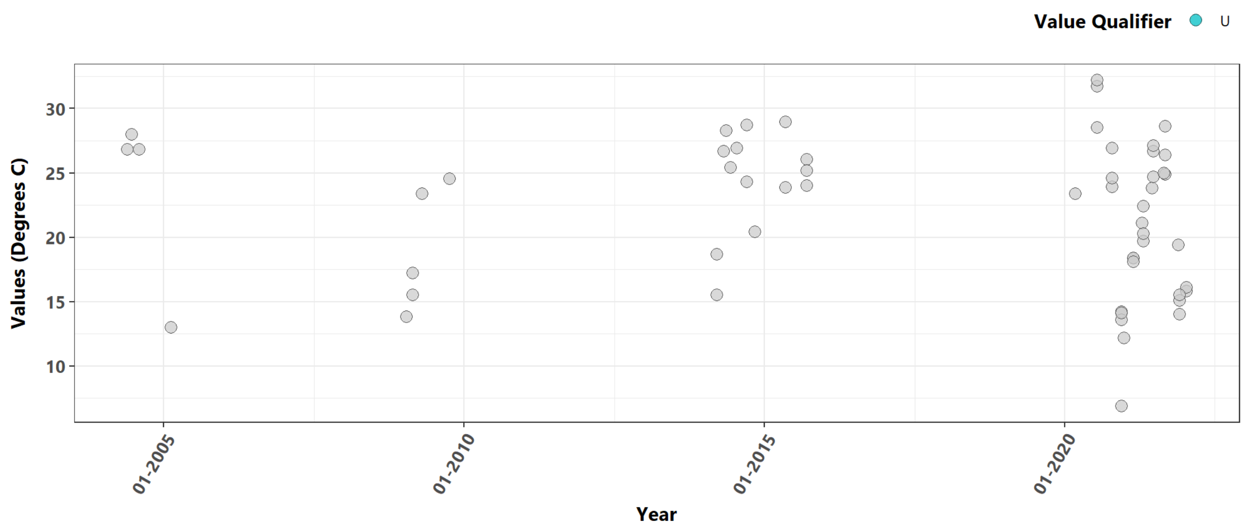
North Fork St. Lucie Aquatic Preserve (5 Unique Years)

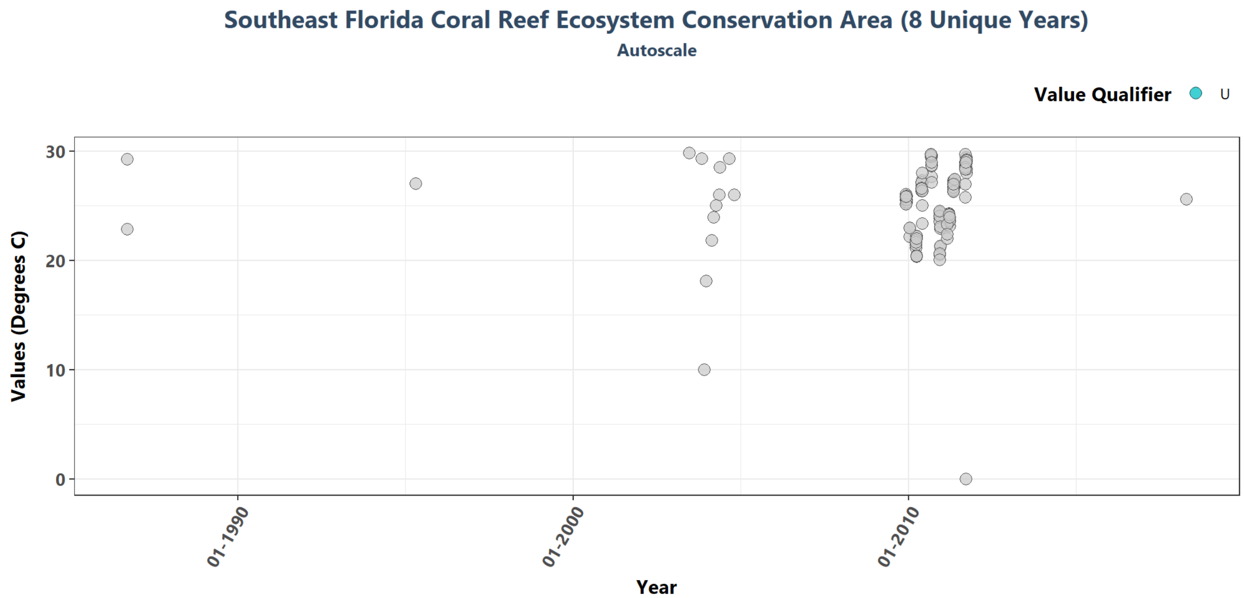
Autoscale



Pellicer Creek Aquatic Preserve (8 Unique Years)

Autoscale





Appendix IV: Managed Area Trendlines

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

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

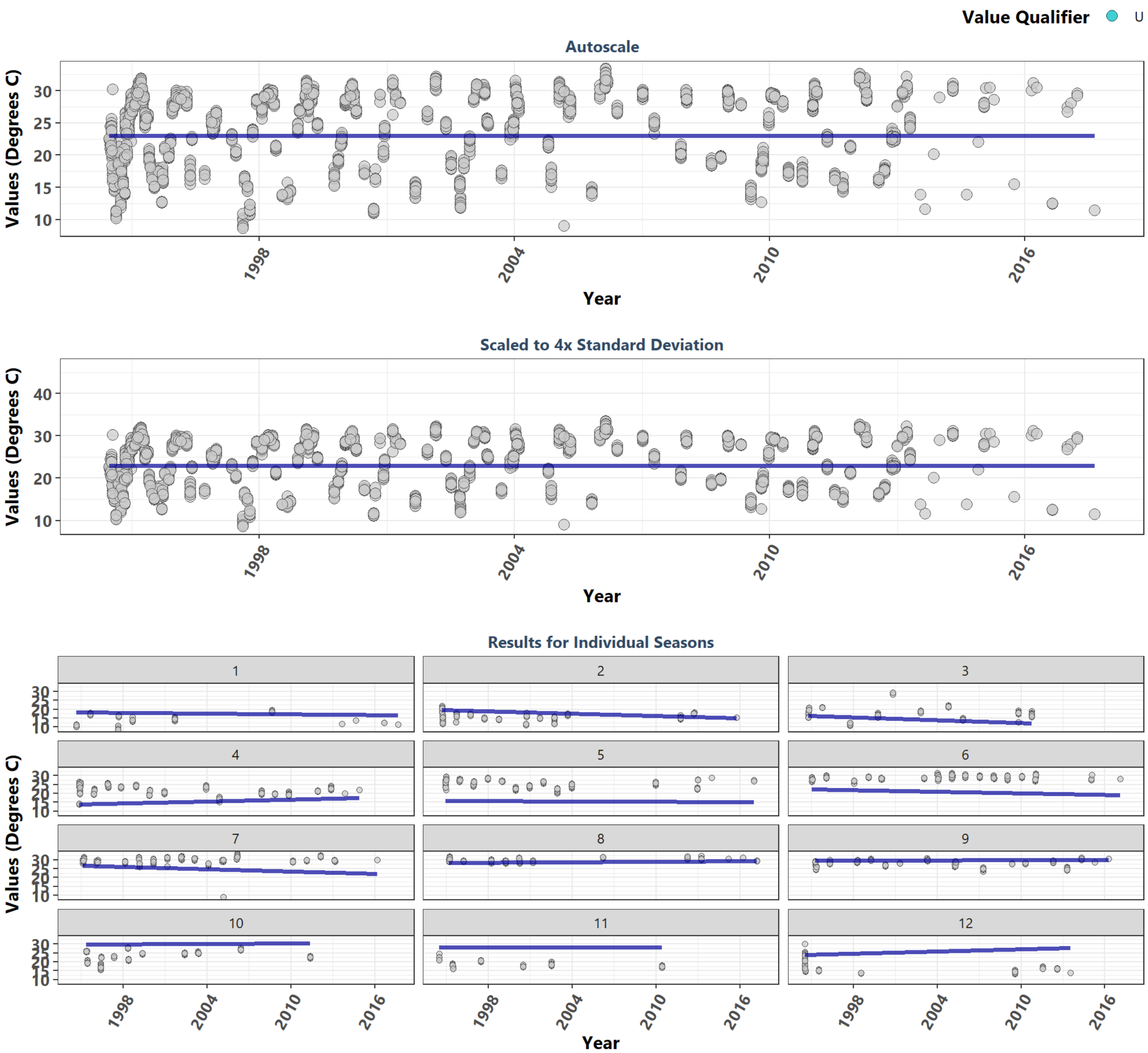
```
if(n==0){
   print("There are no managed areas that qualify.")
} else {
   for (i in 1:n) {
      plot_data <- data[data$SufficientData==TRUE &
                           data$ManagedAreaName==MA_Include[i],]
      plot_data$Season <- factor(plot_data$Month, levels = c("All", seq(1, 12)), ordered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

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

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

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

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

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

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

# plot_data[is.na(VQ_Plot), VQ_Plot := "None"]
p1 <- ggplot(data=plot_data,
             aes(x=relyear_dd, y=ResultValue, fill = VQ_Plot)) +
   geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
   # geom_abline(aes(slope=s_slope, intercept=s_int),
   #             color="#000099", size=1.2, alpha=0.7) +
   geom_segment(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", ], aes(x = rely
                                                                                          y = relyear_
                                                                                          xend = relyea
                                                                                          yend = relyea
                color="#000099", size=1.2, alpha=0.7, inherit.aes = FALSE) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")"),
        fill="Value Qualifier") +
   plot_theme +
   theme(legend.position="top", legend.box="horizontal",
         legend.justification="right") +
   {if(inc_H==TRUE){
      scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                 "HU"="#7CAE00"), na.value="#cccccc")
```

```
    } else if(param_name=="Secchi_Depth"){
      scale_fill_manual(values=c("S"= "#F8766D", "U"= "#00BFC4",
                                 "SU"="#7CAE00"), na.value="#cccccc")
    } else {
      scale_fill_manual(values=c("U"= "#00BFC4"), na.value="#cccccc")
    }} +
    scale_x_continuous(breaks = xbrks,
                       labels = xlabs)

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

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

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

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

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

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

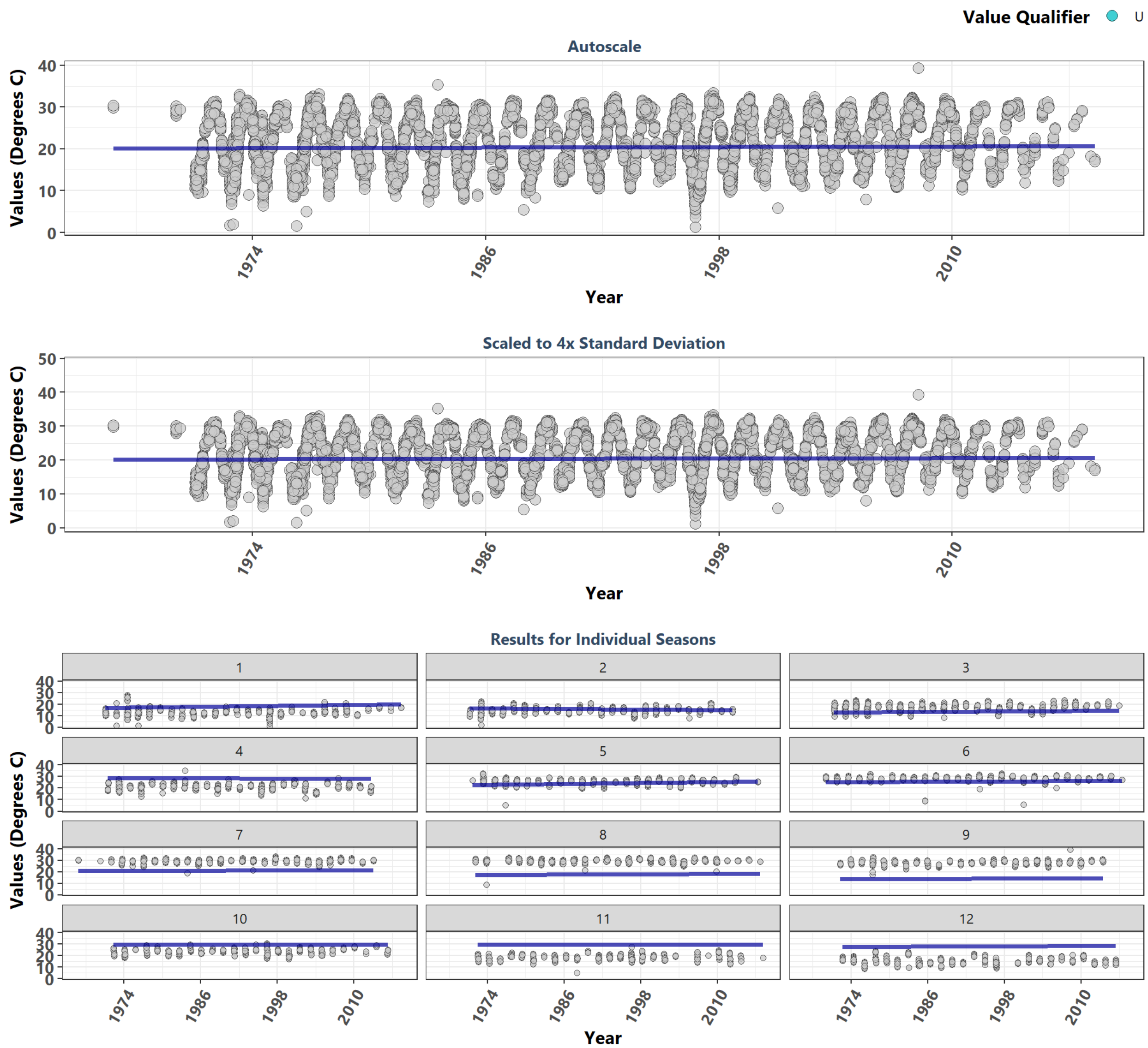
Alligator Harbor Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2822	26.10	-0.0349	0.0000	22.9900	-0.7	0.5124	212.9	0	0
1	87	14.50	-0.0856	-0.0818	18.5864	1.6	0.1000	NA	NA	0
2	201	15.50	-0.3814	-0.2125	20.1812	-0.9	0.3825	NA	NA	0
3	170	17.85	-0.3039	-0.2706	17.0118	-1.7	0.0904	NA	NA	0
4	312	21.60	0.1182	0.1717	13.4701	-6.2	0.0000	NA	NA	0
5	289	25.80	-0.0411	-0.0250	15.7250	-8.3	0.0000	NA	NA	0
6	449	28.70	-0.2318	-0.1500	22.8000	5.4	0.0000	NA	NA	0
7	355	29.70	-0.3264	-0.2091	27.2636	1.8	0.0660	NA	NA	0
8	190	29.90	0.1703	0.0571	27.9571	2.0	0.0417	NA	NA	0
9	309	28.10	0.0651	0.0200	29.5000	0.4	0.6690	NA	NA	0
10	223	24.50	0.0978	0.0333	29.6667	3.7	0.0002	NA	NA	0
11	88	18.80	0.0161	0.0067	28.0267	-5.4	0.0000	NA	NA	0
12	149	16.20	0.1633	0.2200	23.1800	-5.9	0.0000	NA	NA	0

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

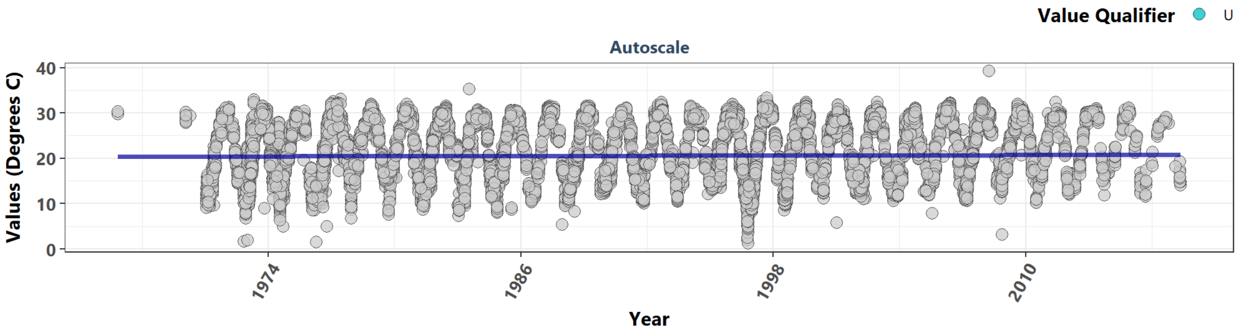
Apalachicola Bay Aquatic Preserve

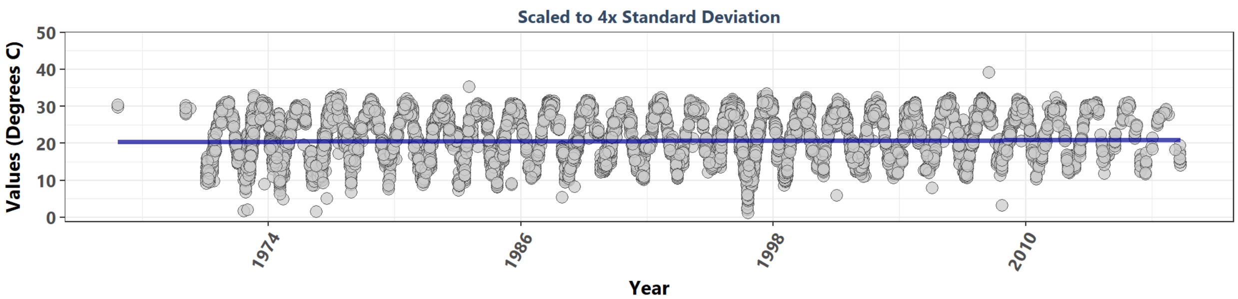


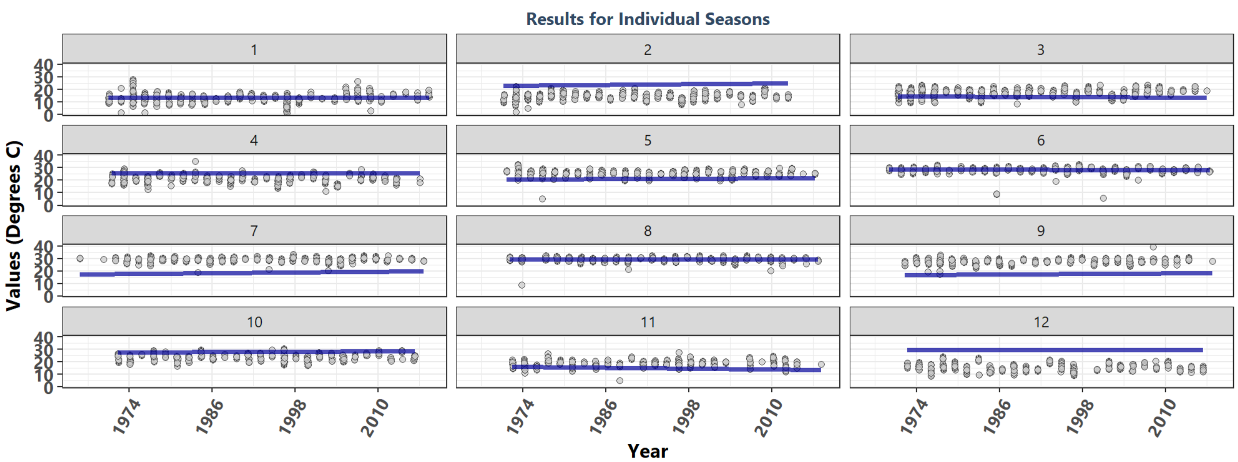
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	14879	24.5	0.0486	0.0167	19.6804	7.3	0.0000	227.6	0	1
1	1065	13.6	0.2135	0.1077	13.9692	3.9	0.0001	NA	NA	1
2	1132	14.3	-0.0920	-0.0571	18.0286	1.5	0.1432	NA	NA	-1
3	1497	17.6	0.0803	0.0462	11.8000	4.8	0.0000	NA	NA	1
4	1210	21.3	-0.0567	-0.0188	29.3250	2.0	0.0418	NA	NA	-1
5	1125	25.8	0.1790	0.1143	19.0857	3.5	0.0004	NA	NA	1
6	1349	28.5	0.0698	0.0286	24.5143	-3.1	0.0018	NA	NA	1
7	1751	29.4	0.0390	0.0250	20.2750	-0.2	0.8281	NA	NA	1
8	1746	29.7	0.0835	0.0500	15.6500	-1.9	0.0624	NA	NA	1
9	1253	28.2	0.0289	0.0190	13.5571	7.8	0.0000	NA	NA	1
10	965	24.0	-0.0035	0.0000	29.4000	8.3	0.0000	NA	NA	-1
11	966	18.6	-0.0297	-0.0077	30.0154	10.0	0.0000	NA	NA	-1
12	820	15.8	0.1474	0.0429	26.3571	-4.0	0.0001	NA	NA	1

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

Apalachicola National Estuarine Research Reserve



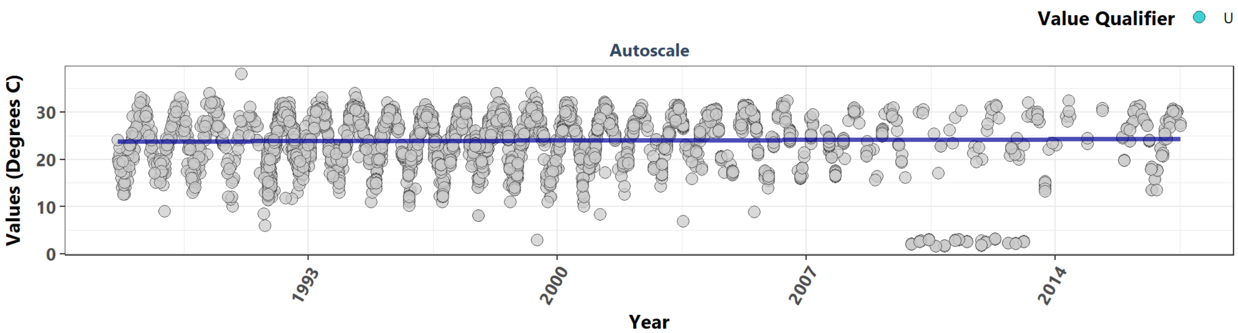


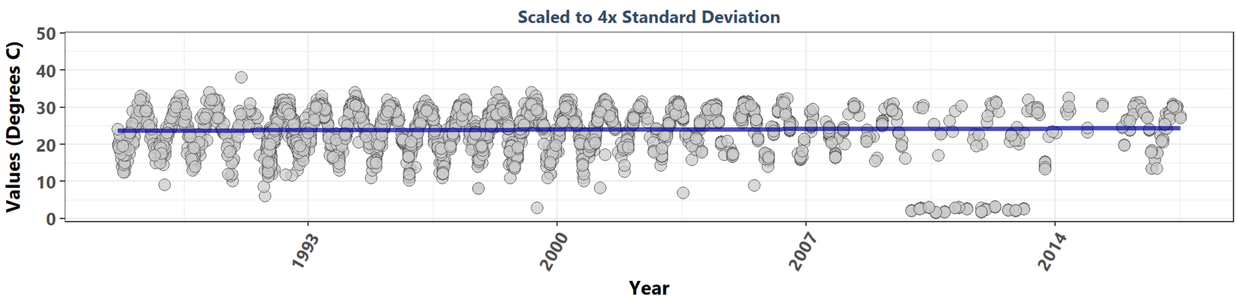


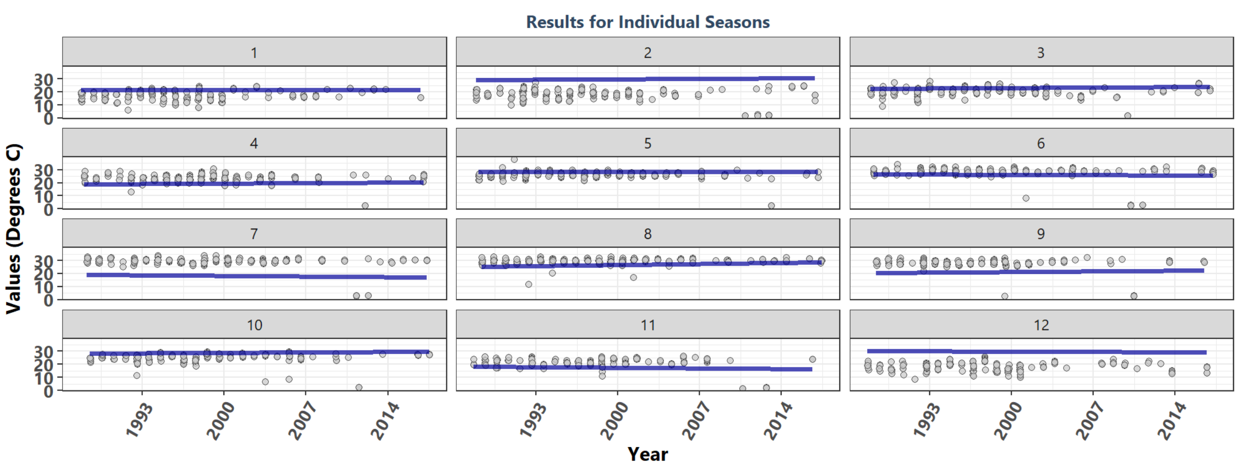
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	21233	22.9	0.0344	0.0154	19.9500	7.7	0.0000	347.4	0	1
1	1753	13.4	0.0240	0.0143	12.8429	1.5	0.1304	NA	NA	1
2	1773	14.2	0.1661	0.1000	19.8000	-2.7	0.0067	NA	NA	1
3	2387	17.5	-0.0428	-0.0286	15.3143	7.2	0.0000	NA	NA	-1
4	1951	21.1	0.0101	0.0000	25.8000	2.8	0.0050	NA	NA	-1
5	1636	25.8	0.0423	0.0250	20.1000	0.6	0.5388	NA	NA	1
6	1579	28.5	-0.0480	-0.0163	29.1988	-2.9	0.0042	NA	NA	-1
7	2131	29.4	0.1658	0.1000	14.4000	-0.3	0.7353	NA	NA	1
8	2151	29.7	-0.0049	0.0000	29.4000	-1.6	0.0993	NA	NA	-1
9	1676	28.0	0.0979	0.0600	15.2200	9.0	0.0000	NA	NA	1
10	1600	24.1	0.1457	0.0444	26.0000	10.0	0.0000	NA	NA	1
11	1385	18.6	-0.1411	-0.0923	18.9077	9.3	0.0000	NA	NA	-1
12	1211	15.4	-0.0237	-0.0056	29.9278	-7.4	0.0000	NA	NA	-1

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

Banana River Aquatic Preserve



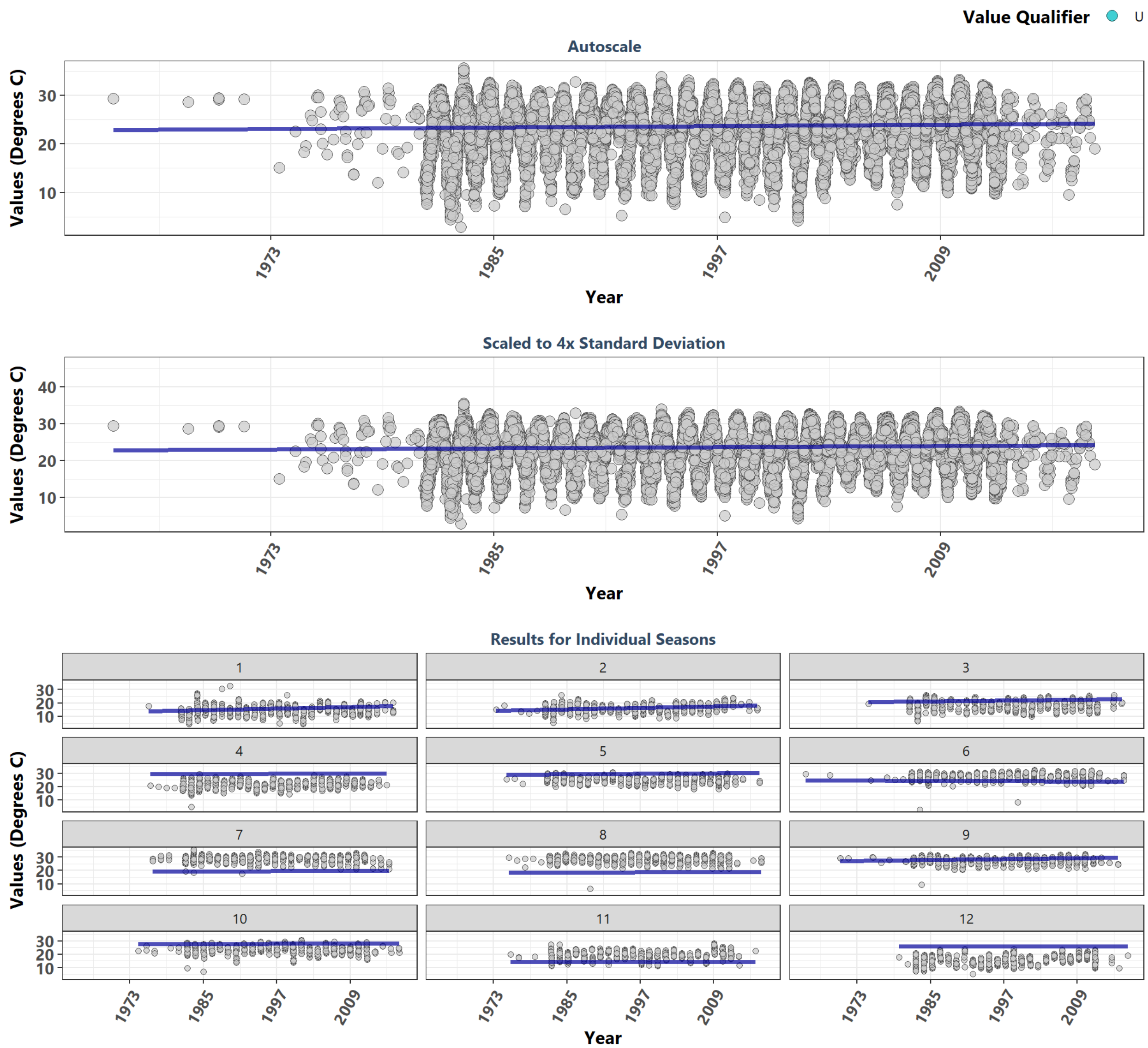




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6260	25.50	0.0367	0.0175	23.7444	4.4	0.0000	123.2	0	1
1	400	17.60	0.0031	0.0000	21.3850	-2.1	0.0391	NA	NA	-1
2	566	18.46	0.1276	0.0400	29.2600	-1.8	0.0761	NA	NA	1
3	448	21.38	0.1045	0.0500	22.5000	0.1	0.9221	NA	NA	1
4	604	23.10	0.0724	0.0571	18.6286	3.9	0.0001	NA	NA	1
5	658	26.40	-0.0343	-0.0125	28.6375	-3.0	0.0031	NA	NA	-1
6	563	28.50	-0.0767	-0.0333	26.7667	-1.2	0.2210	NA	NA	-1
7	565	29.80	-0.0489	-0.0692	19.1227	-2.0	0.0469	NA	NA	-1
8	665	29.70	0.1971	0.1111	24.9889	4.9	0.0000	NA	NA	1
9	423	28.70	0.1229	0.0680	20.2200	2.9	0.0032	NA	NA	1
10	613	26.10	0.0950	0.0397	28.3031	7.3	0.0000	NA	NA	1
11	417	20.90	-0.0686	-0.0667	18.2667	3.8	0.0002	NA	NA	-1
12	338	19.20	-0.0556	-0.0175	30.0100	2.0	0.0447	NA	NA	-1

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

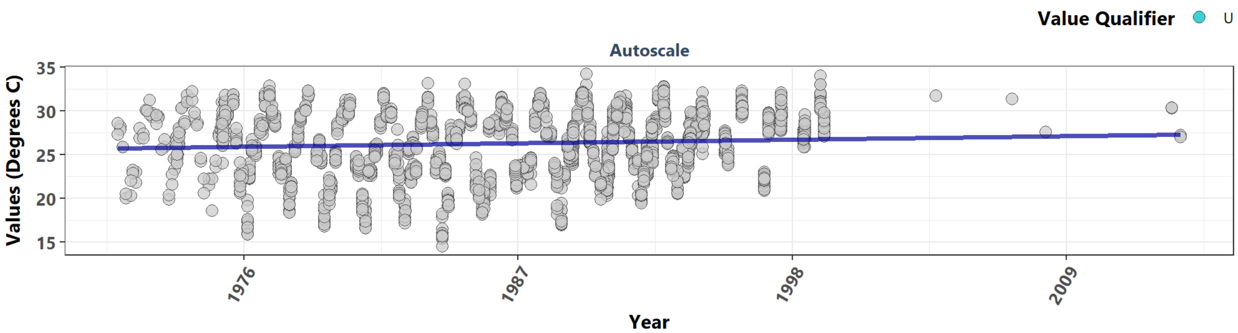
Big Bend Seagrasses Aquatic Preserve

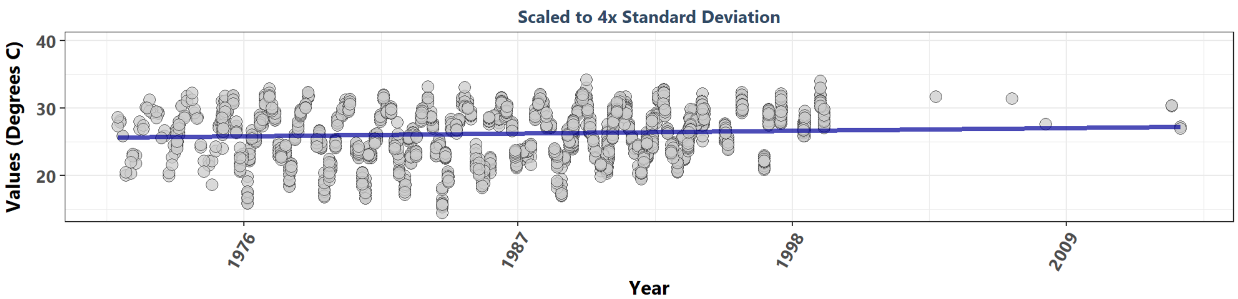


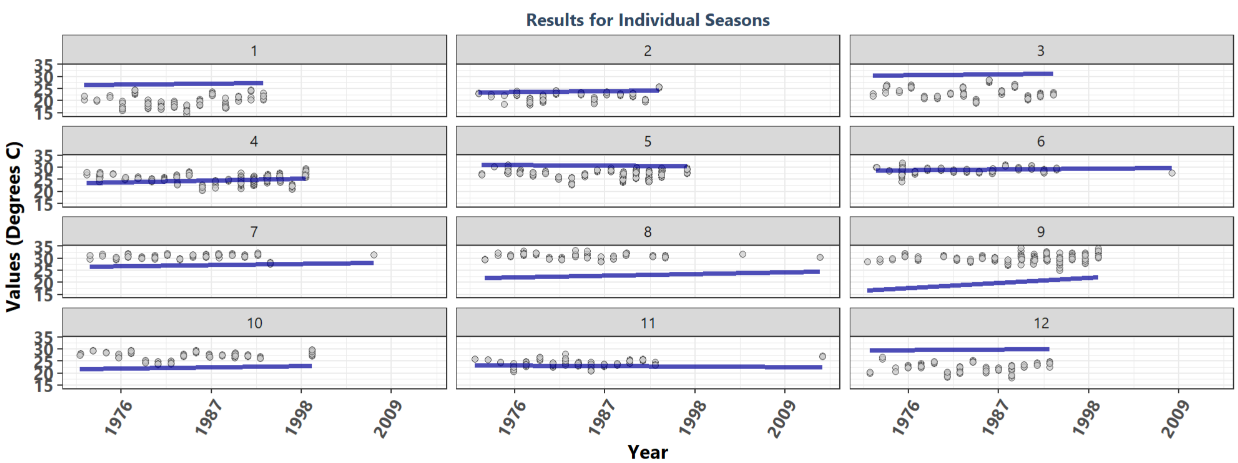
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	33426	24.4	0.0823	0.0333	22.2229	21.9	0.0000	497.4	0	1
1	1962	14.3	0.1887	0.1240	10.6920	0.3	0.7370	NA	NA	1
2	2134	15.9	0.1701	0.1267	11.1800	13.1	0.0000	NA	NA	1
3	2968	18.7	0.1431	0.0727	18.9455	2.2	0.0248	NA	NA	1
4	3318	22.0	0.1017	0.0278	28.5333	12.4	0.0000	NA	NA	1
5	3069	25.9	0.1118	0.0333	28.3667	-0.5	0.6082	NA	NA	1
6	2730	28.6	-0.0401	-0.0231	25.2923	18.5	0.0000	NA	NA	-1
7	3088	29.7	0.0200	0.0118	19.2059	8.5	0.0000	NA	NA	1
8	3433	29.8	0.0275	0.0200	17.8400	9.8	0.0000	NA	NA	1
9	3129	28.0	0.2355	0.0800	25.2400	4.5	0.0000	NA	NA	1
10	3079	24.3	0.0531	0.0167	27.3000	-3.3	0.0008	NA	NA	1
11	2343	19.7	0.0051	0.0000	14.3000	1.5	0.1465	NA	NA	-1
12	2173	16.5	-0.0062	0.0000	25.9000	11.9	0.0000	NA	NA	-1

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

Biscayne Bay Aquatic Preserve



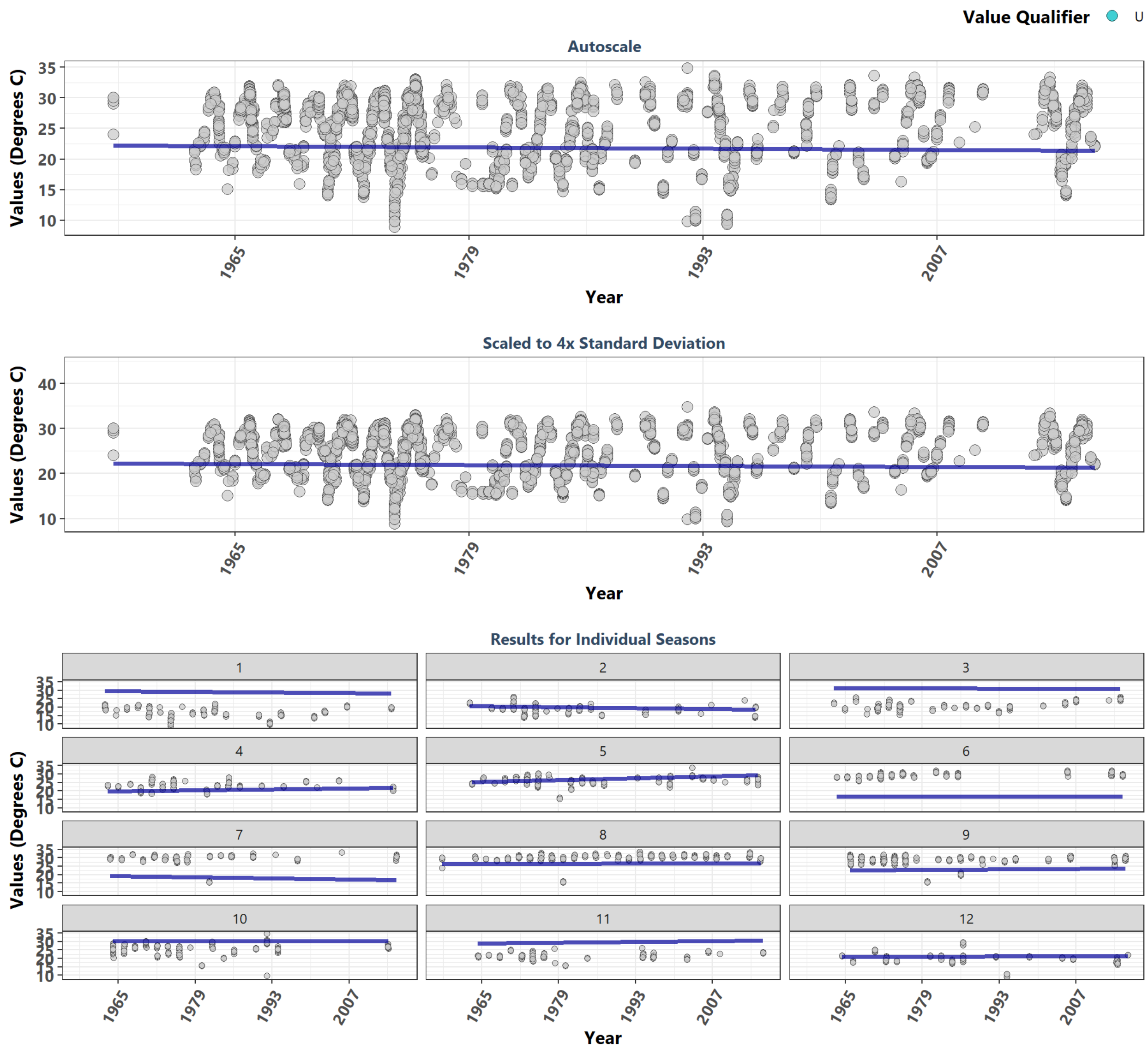




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3251	27.20	0.0874	0.0575	24.3468	6.5	0.0000	31.9	0.0008	1
1	136	19.46	0.0679	0.0600	25.0800	3.8	0.0001	NA	NA	1
2	125	22.51	0.0997	0.0567	21.9867	1.7	0.0930	NA	NA	1
3	152	22.92	0.1364	0.0540	29.2180	-0.6	0.5486	NA	NA	1
4	685	25.00	0.1265	0.1000	21.2000	5.0	0.0000	NA	NA	1
5	458	27.24	-0.0748	-0.0271	31.6686	2.2	0.0257	NA	NA	-1
6	228	29.00	0.1015	0.0400	27.8400	2.4	0.0184	NA	NA	1
7	147	31.00	0.1348	0.0667	24.9000	2.5	0.0140	NA	NA	1
8	137	30.80	0.1279	0.1033	19.3033	-1.3	0.1936	NA	NA	1
9	679	30.00	0.2222	0.3000	9.5600	1.9	0.0537	NA	NA	1
10	231	27.30	0.1013	0.0667	20.2433	3.1	0.0018	NA	NA	1
11	138	23.80	-0.0328	-0.0271	23.7936	1.7	0.0822	NA	NA	-1
12	135	22.61	0.0491	0.0317	28.7650	2.2	0.0273	NA	NA	1

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

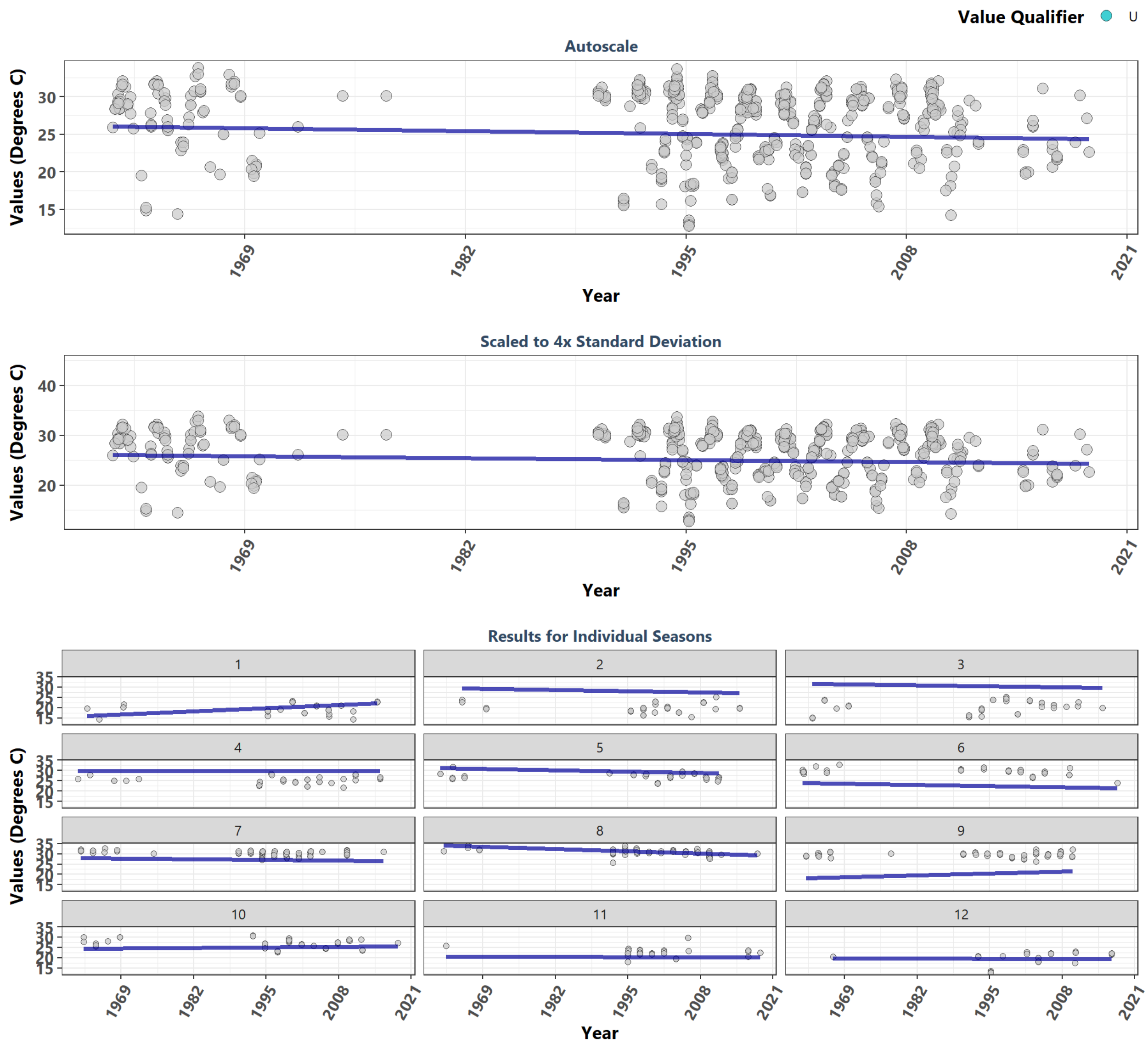
Boca Ciega Bay Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	5680	28.51	0.0002	-0.0284	23.2736	-4.1	0.0000	157	0	-1
1	216	16.80	-0.0633	-0.0422	31.0878	-0.1	0.8955	NA	NA	-1
2	181	18.66	-0.1759	-0.0800	24.0600	-2.6	0.0087	NA	NA	-1
3	182	20.51	-0.0450	-0.0140	31.9280	2.9	0.0039	NA	NA	-1
4	183	22.90	0.1434	0.0832	16.2689	1.3	0.1790	NA	NA	1
5	228	26.50	0.2024	0.1550	18.9350	0.5	0.6046	NA	NA	1
6	188	29.30	-0.0060	-0.0033	16.9733	6.3	0.0000	NA	NA	-1
7	158	30.26	-0.1304	-0.0814	22.4871	0.7	0.5115	NA	NA	-1
8	567	31.27	0.0230	0.0073	26.1327	-1.7	0.0910	NA	NA	1
9	2737	29.19	0.0666	0.0411	20.8444	-5.1	0.0000	NA	NA	1
10	657	25.60	0.0351	0.0075	29.9050	7.9	0.0000	NA	NA	1
11	214	21.30	0.3053	0.0667	26.1000	1.6	0.1206	NA	NA	1
12	169	19.90	0.0705	0.0208	20.1283	-3.4	0.0006	NA	NA	1

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

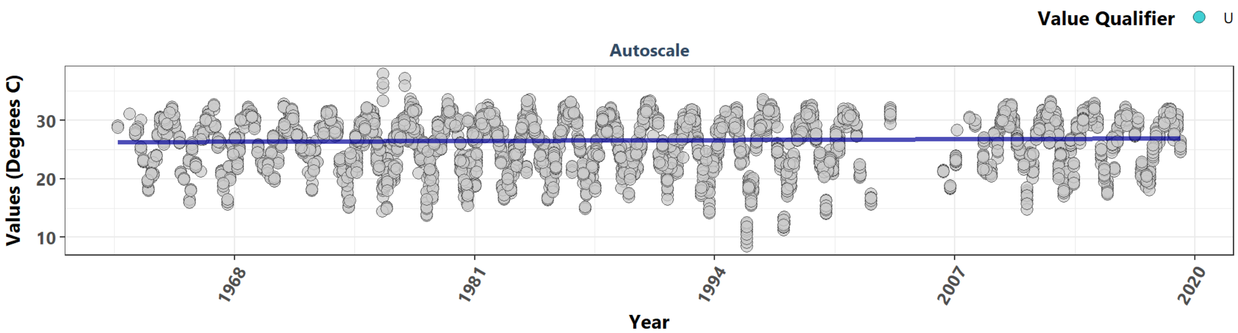
Cape Haze Aquatic Preserve

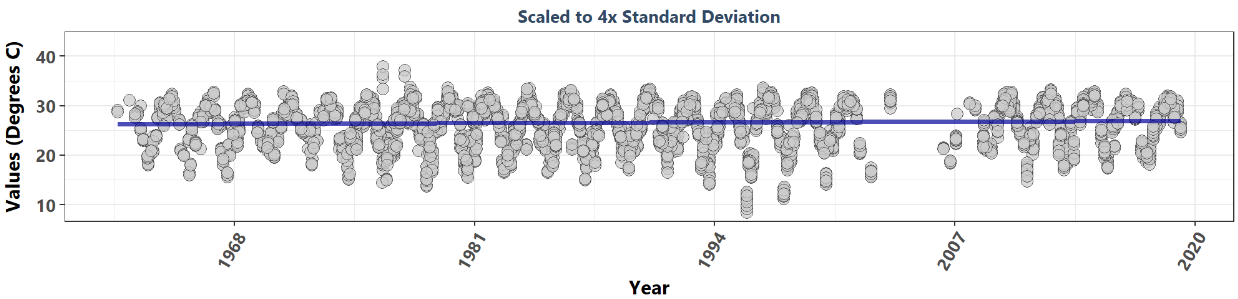


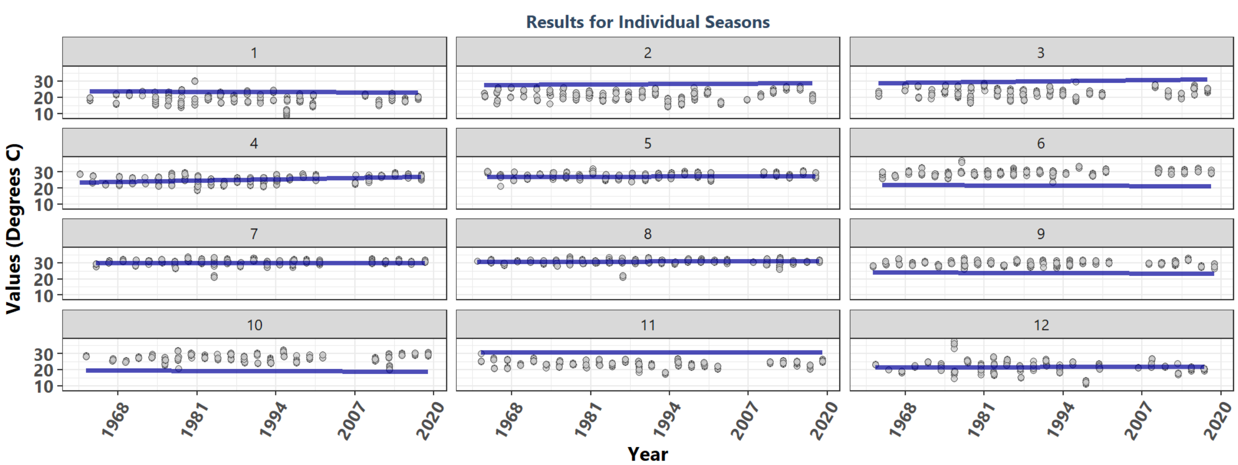
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	623	28.10	-0.0957	-0.0643	28.4568	-4.7	0.0000	48.3	0	-1
1	31	20.30	0.2849	0.2612	5.9988	-0.1	0.8907	NA	NA	1
2	34	19.55	-0.1478	-0.0968	33.2197	-0.2	0.8107	NA	NA	-1
3	45	20.70	-0.1867	-0.0727	34.3000	1.7	0.0946	NA	NA	-1
4	41	25.20	-0.0119	0.0000	29.7500	1.2	0.2389	NA	NA	-1
5	46	27.84	-0.2495	-0.1072	35.1967	-1.4	0.1472	NA	NA	-1
6	44	29.30	-0.0731	-0.0933	27.3067	-2.4	0.0165	NA	NA	-1
7	116	30.30	-0.0587	-0.0476	29.6069	-3.0	0.0028	NA	NA	-1
8	82	31.00	-0.3981	-0.1775	40.6738	-5.3	0.0000	NA	NA	-1
9	68	29.75	0.1727	0.1518	12.3529	-0.1	0.8898	NA	NA	1
10	49	26.99	0.1280	0.0459	22.6273	-0.6	0.5557	NA	NA	1
11	40	22.22	-0.0194	-0.0100	20.8800	-0.7	0.5054	NA	NA	-1
12	27	20.89	-0.0303	-0.0146	20.3685	2.1	0.0365	NA	NA	-1

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

Cape Romano-Ten Thousand Islands Aquatic Preserve



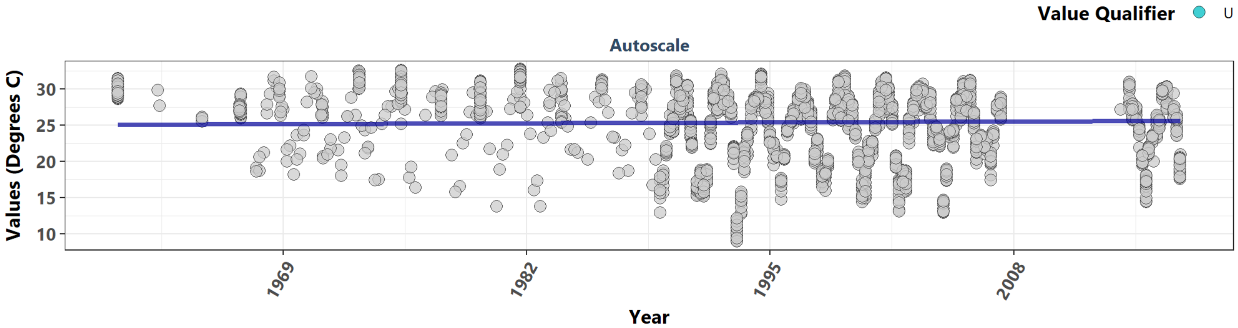


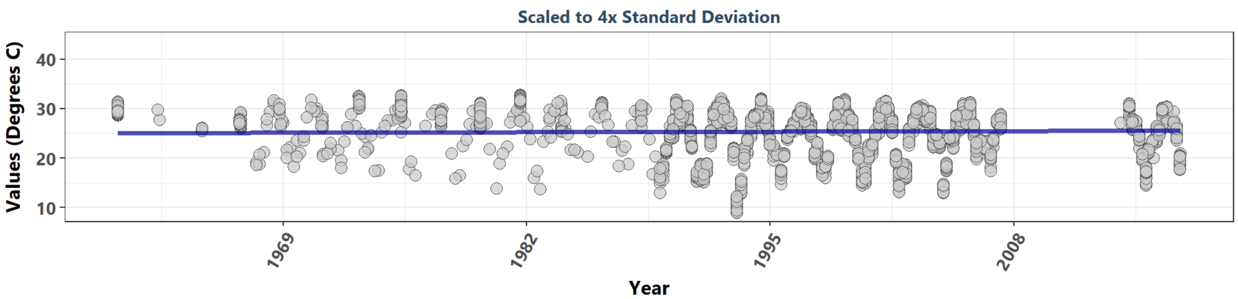


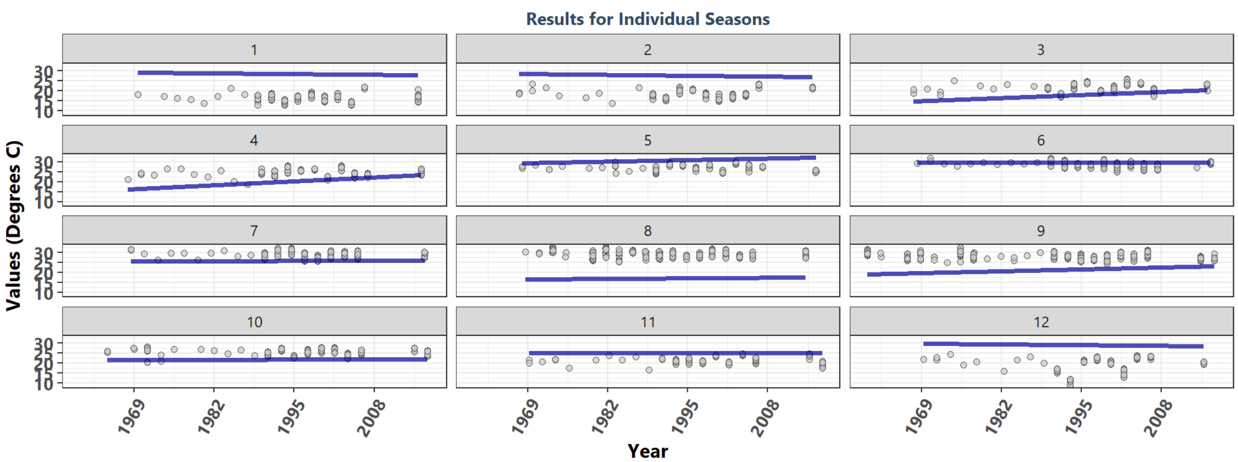
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4893	26.90	0.0566	0.0250	25.2870	6.2	0.0000	135	0	1
1	377	19.30	-0.0431	-0.0300	24.8700	-1.3	0.1812	NA	NA	-1
2	413	21.72	0.0935	0.0333	26.5000	0.3	0.7937	NA	NA	1
3	333	23.40	0.3097	0.0889	25.3356	-1.2	0.2403	NA	NA	1
4	416	25.25	0.2337	0.1333	18.5833	7.1	0.0000	NA	NA	1
5	437	28.20	0.0552	0.0286	25.7293	2.9	0.0034	NA	NA	1
6	397	29.78	-0.0397	-0.0316	23.2789	9.2	0.0000	NA	NA	-1
7	426	30.74	0.0091	0.0000	29.8000	0.8	0.4142	NA	NA	-1
8	465	30.80	0.1008	0.0250	29.5250	3.3	0.0011	NA	NA	1
9	419	29.80	-0.0713	-0.0308	25.2385	0.3	0.7815	NA	NA	-1
10	446	27.22	-0.0461	-0.0316	20.8789	1.7	0.0812	NA	NA	-1
11	383	23.70	0.0264	0.0059	30.4350	-2.1	0.0370	NA	NA	1
12	381	21.70	0.0086	0.0048	21.4771	-1.2	0.2468	NA	NA	1

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

Cockroach Bay Aquatic Preserve



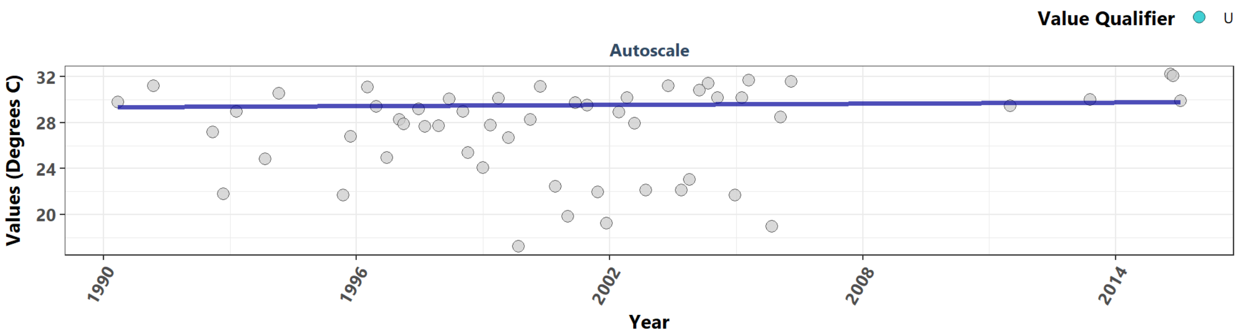


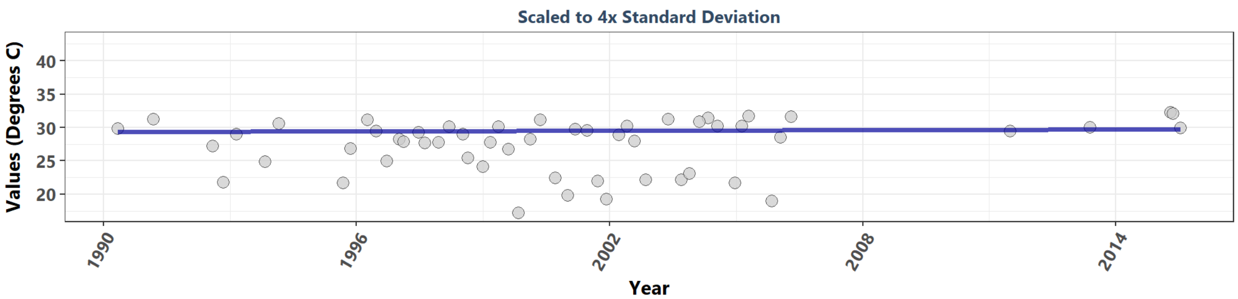


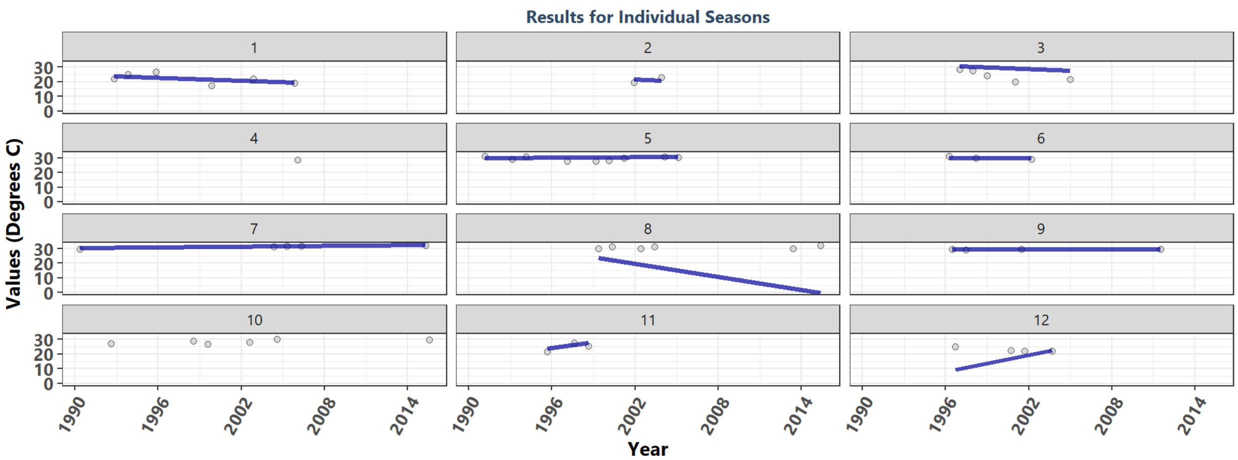
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3835	27.75	0.0230	0.0171	24.4719	1.7	0.0882	132.6	0	0
1	151	17.15	-0.1356	-0.0685	32.0923	1.3	0.1991	NA	NA	0
2	152	18.68	-0.0941	-0.0591	30.7571	4.9	0.0000	NA	NA	0
3	157	21.98	0.2685	0.2571	4.2750	3.1	0.0020	NA	NA	0
4	161	25.22	0.1993	0.3235	2.9583	0.1	0.9543	NA	NA	0
5	170	27.51	0.1110	0.1275	24.0950	-1.8	0.0675	NA	NA	0
6	164	28.90	-0.0156	-0.0100	30.0750	-2.3	0.0217	NA	NA	0
7	148	29.52	0.0417	0.0178	24.8487	-0.3	0.7781	NA	NA	0
8	1262	29.96	0.0702	0.0467	14.5367	6.2	0.0000	NA	NA	0
9	977	29.08	0.1653	0.1558	13.4092	-6.6	0.0000	NA	NA	0
10	175	25.83	0.0285	0.0233	20.6633	0.8	0.4114	NA	NA	0
11	175	21.97	0.0031	0.0023	25.0954	0.6	0.5744	NA	NA	0
12	143	20.75	-0.1203	-0.0600	32.2000	3.5	0.0004	NA	NA	0

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

Coupon Bight Aquatic Preserve



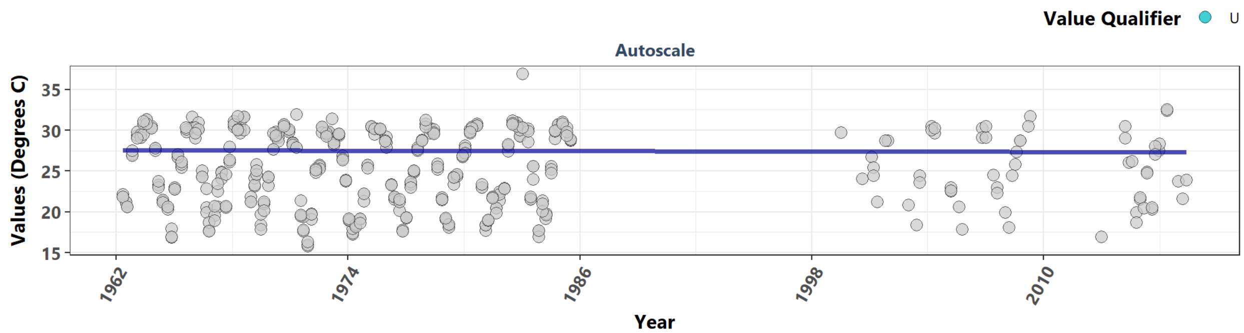


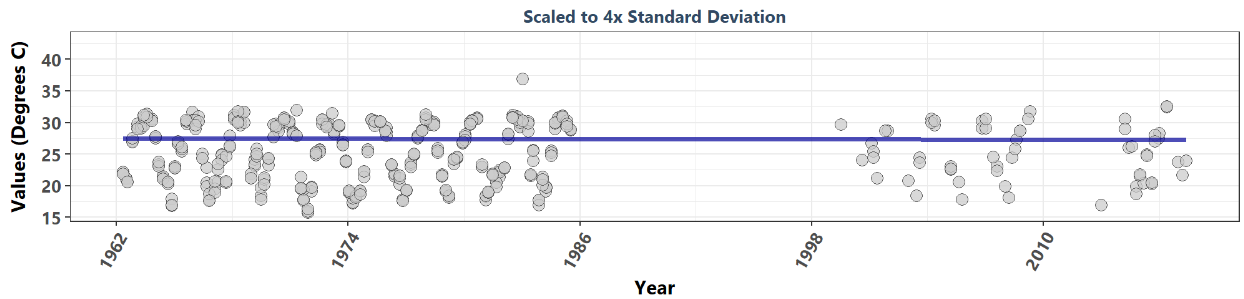


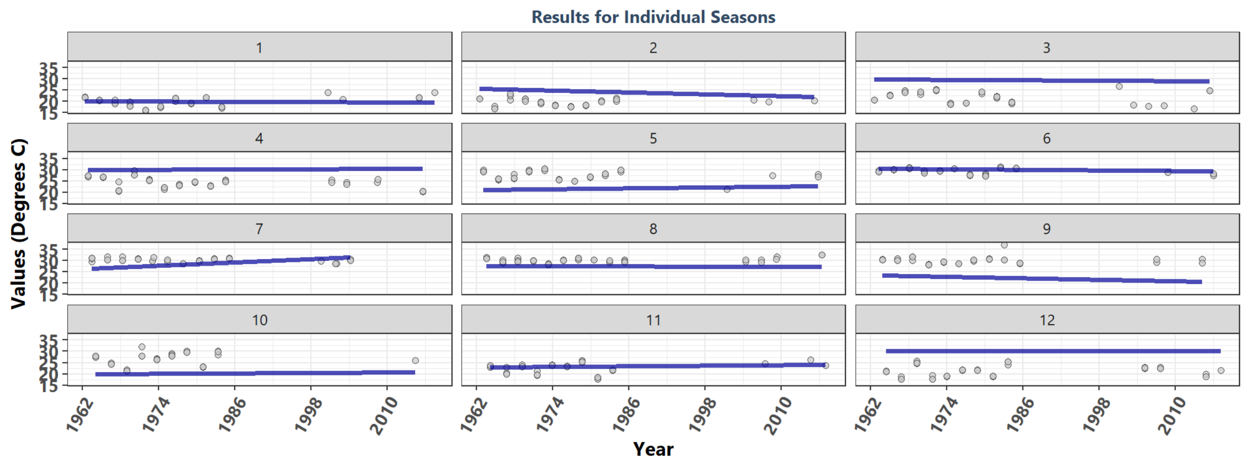
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	54	28.69	0.0428	0.0168	29.3368	0.3	0.7448	16.6	0.0847	0
1	6	21.97	-0.2000	-0.3005	24.3747	-0.4	0.7071	NA	NA	0
2	2	NA	-0.6667	-0.4474	26.9642	NA	NA	NA	NA	NA
3	5	24.10	-1.0000	-0.3652	32.9712	-1.7	0.0864	NA	NA	0
4	1	NA	0.4667	0.1512	26.8700	NA	NA	NA	NA	NA
5	9	29.76	0.3333	0.0648	29.8341	0.0	1.0000	NA	NA	0
6	3	30.05	0.0000	-0.0010	29.7677	-1.0	0.2963	NA	NA	0
7	5	31.57	0.8000	0.0848	30.2964	1.7	0.0864	NA	NA	0
8	6	30.64	-0.8000	-1.4763	37.3847	0.8	0.4524	NA	NA	0
9	4	29.42	0.3333	0.0095	29.3368	0.3	0.7341	NA	NA	0
10	6	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	3	25.41	0.3333	1.2367	16.7533	0.0	1.0000	NA	NA	0
12	4	22.27	1.0000	1.9050	-3.6371	-1.0	0.3082	NA	NA	0

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

Estero Bay Aquatic Preserve



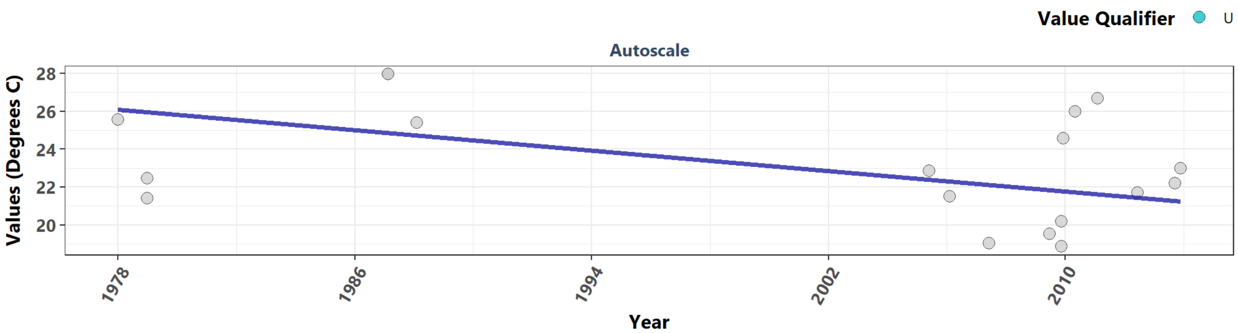


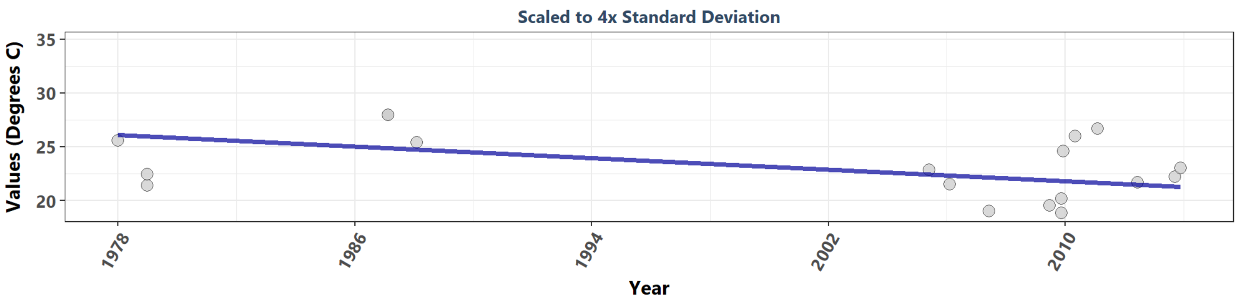


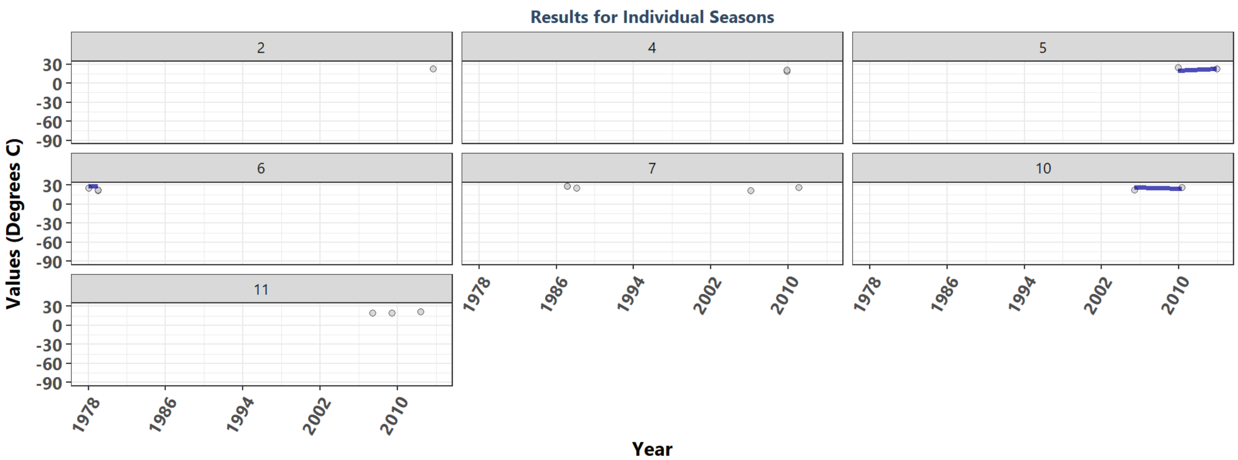
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	412	25.56	-0.0132	-0.0092	27.8478	-0.6	0.5626	17.3	0.1004	0
1	35	20.14	-0.0398	-0.0217	20.7883	0.4	0.7002	NA	NA	0
2	33	19.90	-0.2731	-0.1660	31.4420	-0.3	0.7555	NA	NA	0
3	37	22.44	-0.0749	-0.0364	31.0409	-1.4	0.1681	NA	NA	0
4	38	24.47	0.0768	0.0210	29.2730	-2.4	0.0160	NA	NA	0
5	34	27.34	0.2067	0.0824	18.0435	-0.1	0.9170	NA	NA	0
6	34	29.55	-0.1571	-0.0500	32.2400	-0.6	0.5414	NA	NA	0
7	36	30.19	0.2249	0.2937	15.9000	-1.3	0.1799	NA	NA	0
8	38	30.16	-0.0143	-0.0116	27.8202	0.7	0.5036	NA	NA	0
9	34	30.10	-0.1592	-0.1294	27.8753	-0.1	0.9525	NA	NA	0
10	28	27.65	0.0471	0.0429	18.3829	1.7	0.0947	NA	NA	0
11	30	23.28	0.1011	0.0500	21.2550	0.8	0.4401	NA	NA	0
12	35	21.42	-0.0089	-0.0014	30.1586	1.7	0.0819	NA	NA	0

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

Fort Pickens State Park Aquatic Preserve



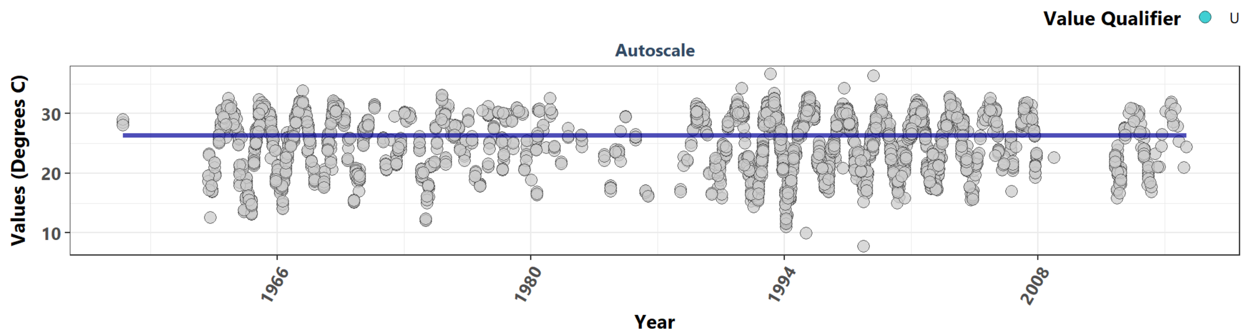


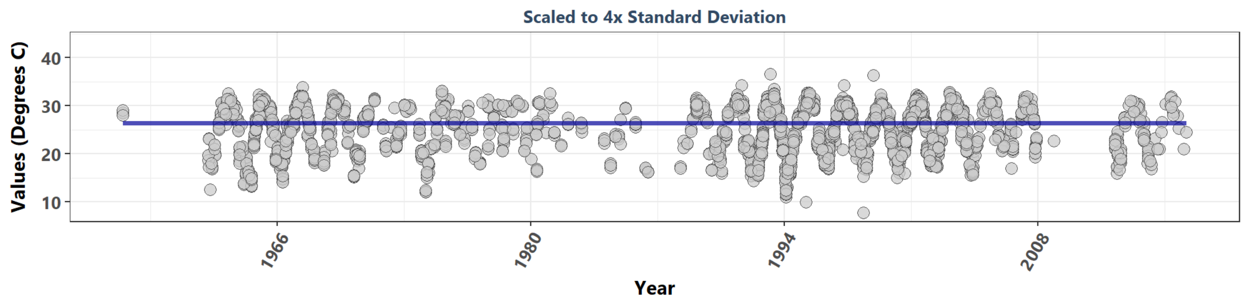


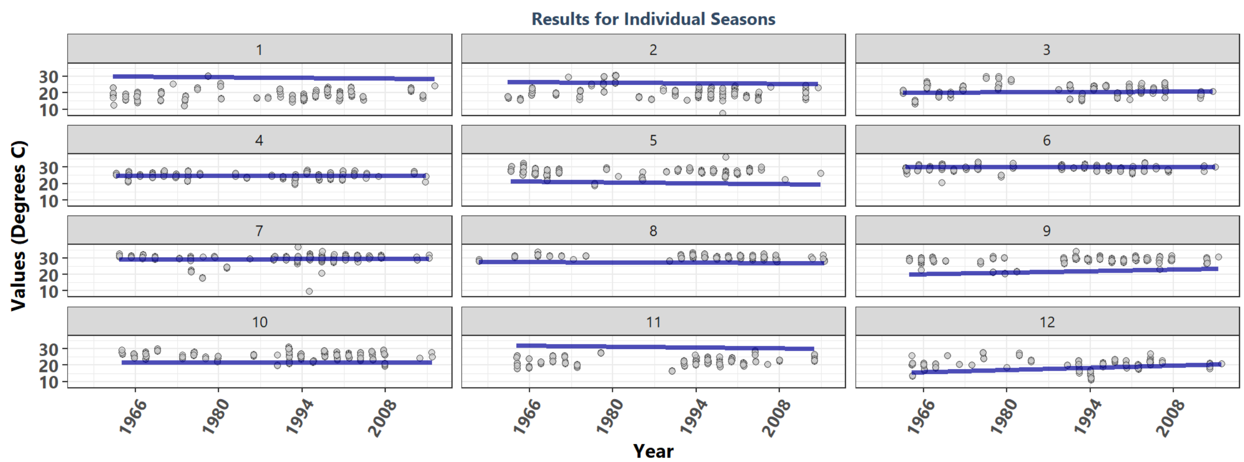
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	18	22.66	-0.1000	-0.1344	26.0850	-0.6	0.5325	7.5	0.1133	0
2	1	NA	1.0000	0.6280	5.9040	NA	NA	NA	NA	NA
4	2	NA	-0.6667	-3.6250	26.0850	NA	NA	NA	NA	NA
5	2	NA	1.0000	0.5360	2.9240	NA	NA	NA	NA	NA
6	3	NA	-0.5000	-0.2156	28.8256	NA	NA	NA	NA	NA
7	5	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	2	NA	-1.0000	-0.3950	37.2200	NA	NA	NA	NA	NA
11	3	NA	NA	NA	NA	NA	NA	NA	NA	NA

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

Gasparilla Sound-Charlotte Harbor Aquatic Preserve



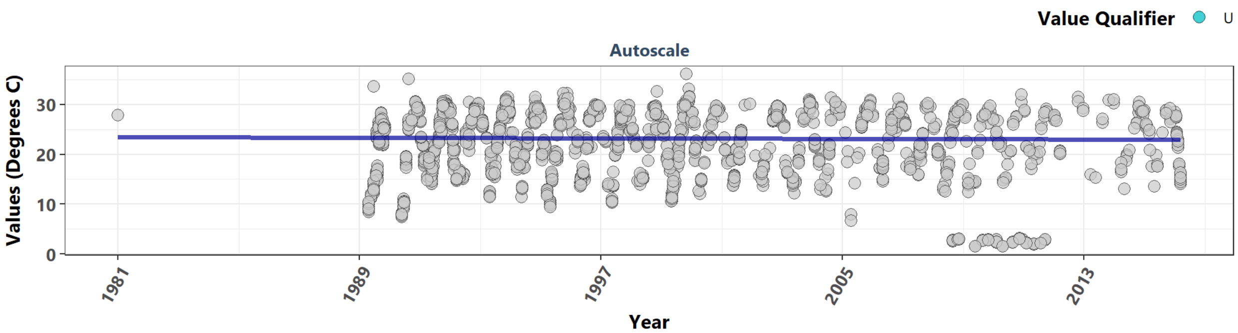


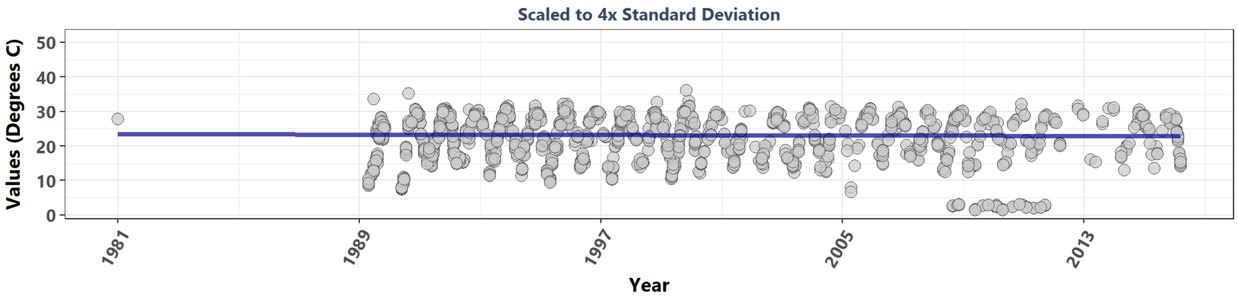


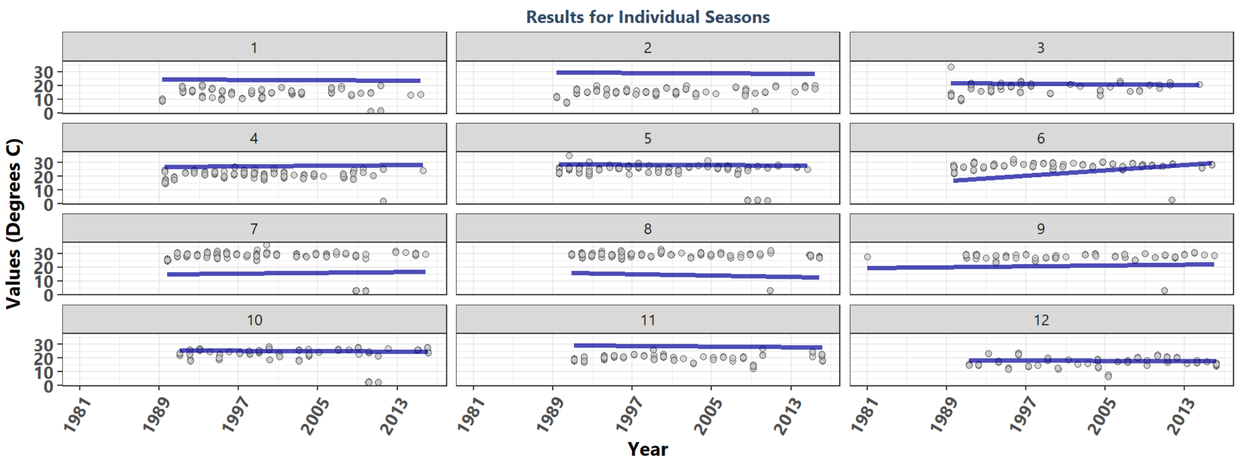
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	4144	25.80	-0.0043	0.0000	26.3902	-0.6	0.5470	185.3	0	0
1	325	18.91	-0.1940	-0.0636	32.5536	7.6	0.0000	NA	NA	0
2	331	20.20	-0.0816	-0.0437	28.4937	-2.7	0.0061	NA	NA	0
3	343	21.60	0.0491	0.0333	18.6667	0.1	0.9074	NA	NA	0
4	334	24.88	0.0211	0.0080	24.4270	0.6	0.5640	NA	NA	0
5	301	27.15	-0.1006	-0.0667	24.0667	-1.2	0.2451	NA	NA	0
6	356	28.99	0.0082	0.0000	30.2000	-5.5	0.0000	NA	NA	0
7	439	30.20	0.0619	0.0167	28.3533	0.3	0.7971	NA	NA	0
8	323	30.70	-0.0448	-0.0220	28.4040	-6.7	0.0000	NA	NA	0
9	380	29.32	0.2213	0.1307	14.8393	1.8	0.0705	NA	NA	0
10	382	26.00	0.0042	0.0034	21.4018	-2.4	0.0169	NA	NA	0
11	302	22.29	-0.2484	-0.0750	35.0500	5.8	0.0000	NA	NA	0
12	328	20.60	0.2798	0.1846	8.2008	1.3	0.1825	NA	NA	0

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

Guana River Marsh Aquatic Preserve



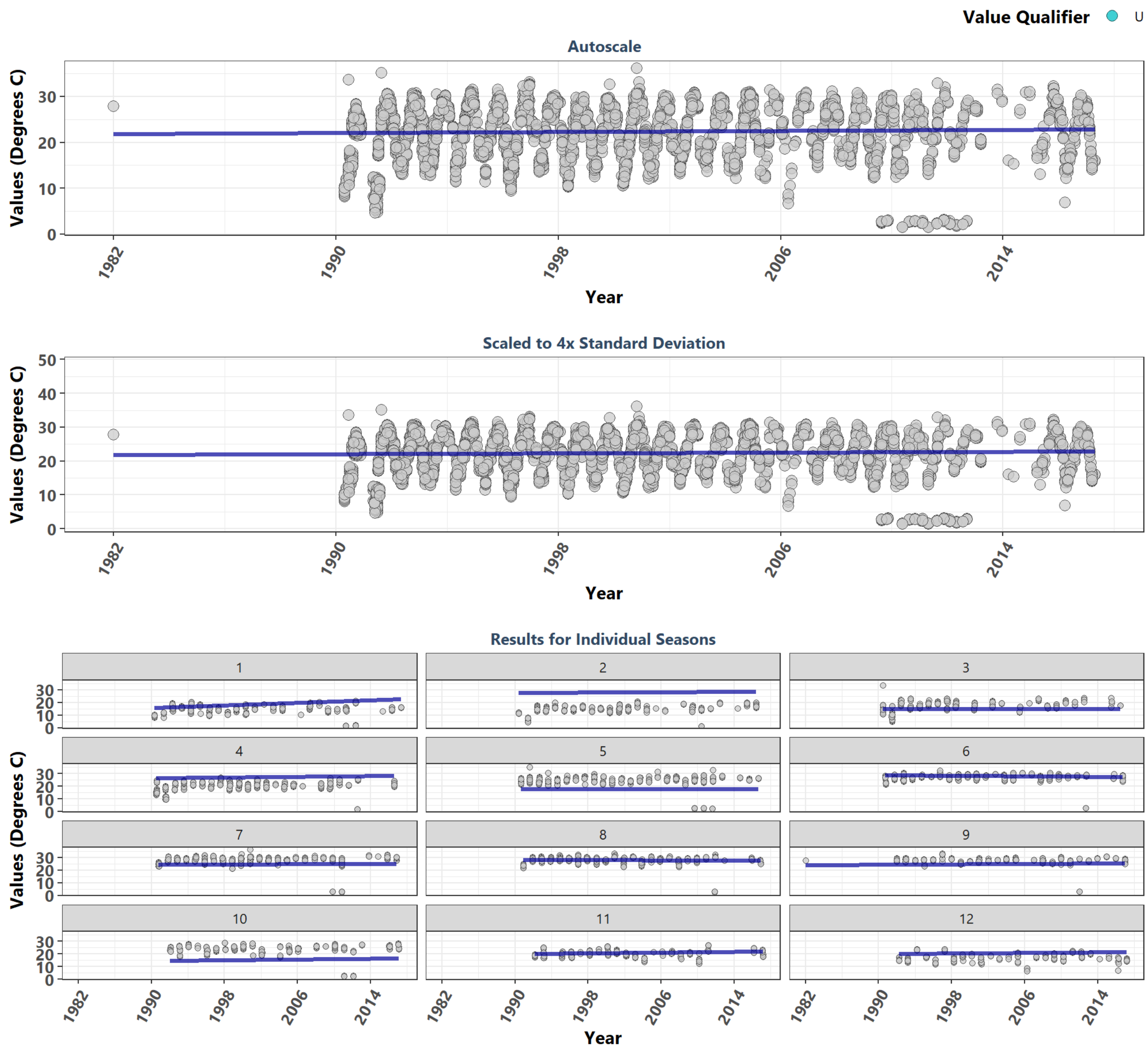




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2524	22.40	-0.0103	-0.0160	23.4823	-2.1	0.0339	93.4	0	-1
1	251	14.70	-0.0555	-0.0341	24.9146	-2.9	0.0042	NA	NA	-1
2	201	15.40	-0.1085	-0.0333	29.6333	1.7	0.0803	NA	NA	-1
3	150	19.00	-0.0606	-0.0500	22.0500	6.2	0.0000	NA	NA	-1
4	271	21.20	0.1311	0.0522	26.5714	-1.5	0.1362	NA	NA	1
5	286	25.35	-0.0855	-0.0250	28.8750	-1.7	0.0823	NA	NA	-1
6	190	27.25	0.3395	0.4889	12.6444	2.7	0.0069	NA	NA	1
7	281	28.80	0.0827	0.0667	14.3333	-2.9	0.0032	NA	NA	1
8	239	29.10	-0.1211	-0.1182	16.8273	-2.5	0.0122	NA	NA	-1
9	129	28.50	0.1781	0.0786	19.3000	-1.4	0.1478	NA	NA	1
10	231	24.30	-0.0688	-0.0444	26.1056	-1.3	0.2082	NA	NA	-1
11	175	20.40	-0.1176	-0.0538	29.7154	3.5	0.0004	NA	NA	-1
12	120	18.04	-0.0025	-0.0017	18.0700	0.0	0.9691	NA	NA	-1

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

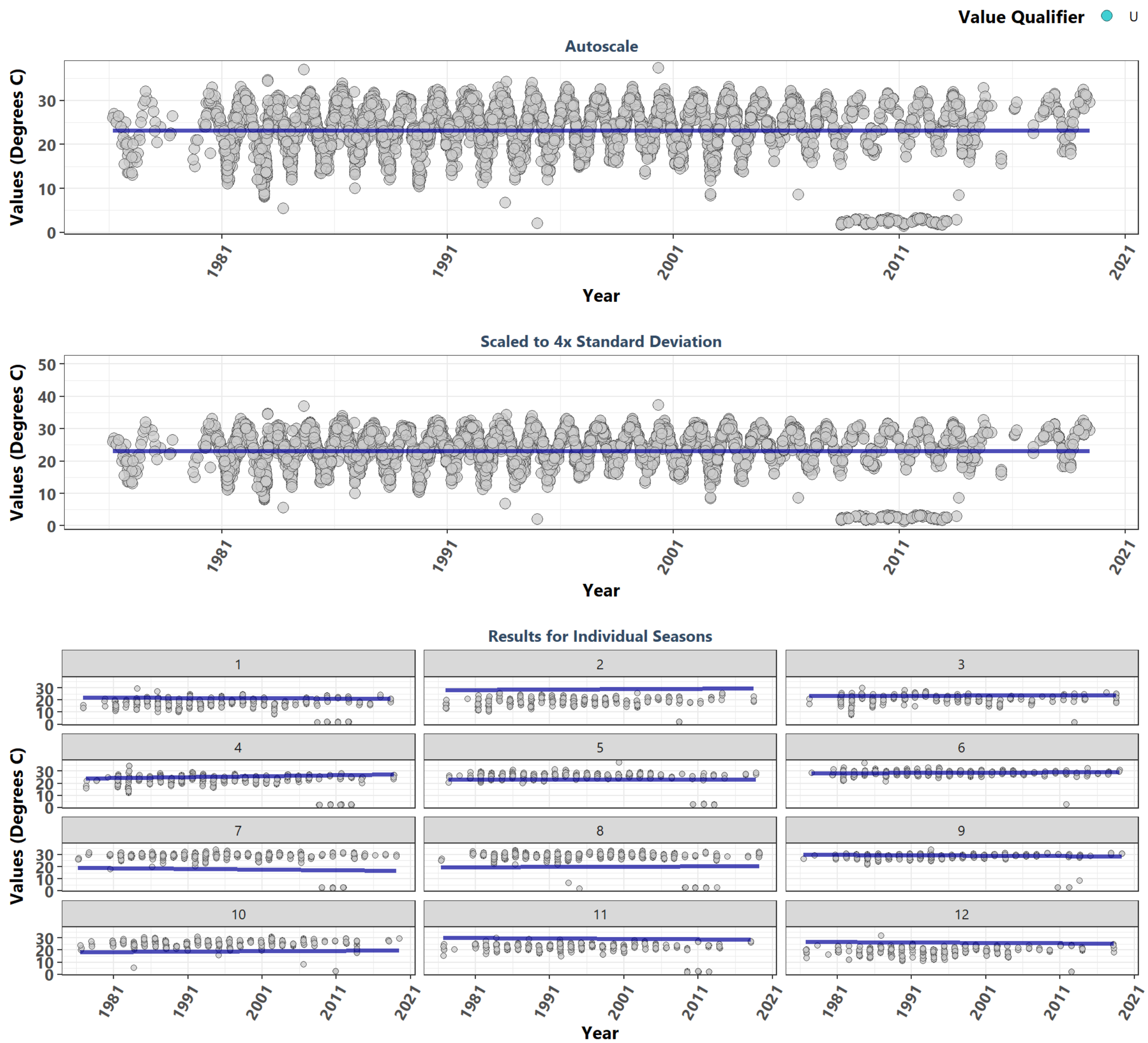
Guana Tolomato Matanzas National Estuarine Research Reserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6556	22.7	0.0590	0.0278	21.8192	6.4	0.0000	123.7	0	1
1	615	15.0	0.2698	0.2500	13.8000	0.1	0.9184	NA	NA	1
2	512	15.1	0.0547	0.0250	27.6750	4.5	0.0000	NA	NA	1
3	492	17.3	0.0028	0.0000	15.0000	9.0	0.0000	NA	NA	-1
4	600	20.5	0.1666	0.0680	25.8120	3.4	0.0006	NA	NA	1
5	644	24.6	0.0050	0.0029	17.6457	0.9	0.3840	NA	NA	1
6	478	26.9	-0.0863	-0.0400	28.9400	5.5	0.0000	NA	NA	-1
7	708	28.1	0.0229	0.0125	24.3750	2.2	0.0291	NA	NA	1
8	692	28.3	-0.0647	-0.0167	28.3833	-3.4	0.0007	NA	NA	-1
9	432	28.1	0.0641	0.0284	24.1884	-2.0	0.0439	NA	NA	1
10	602	24.7	0.1332	0.0714	13.9571	2.4	0.0183	NA	NA	1
11	444	20.3	0.0935	0.0714	19.2857	3.0	0.0028	NA	NA	1
12	337	17.7	0.0946	0.0500	19.4500	0.1	0.8905	NA	NA	1

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

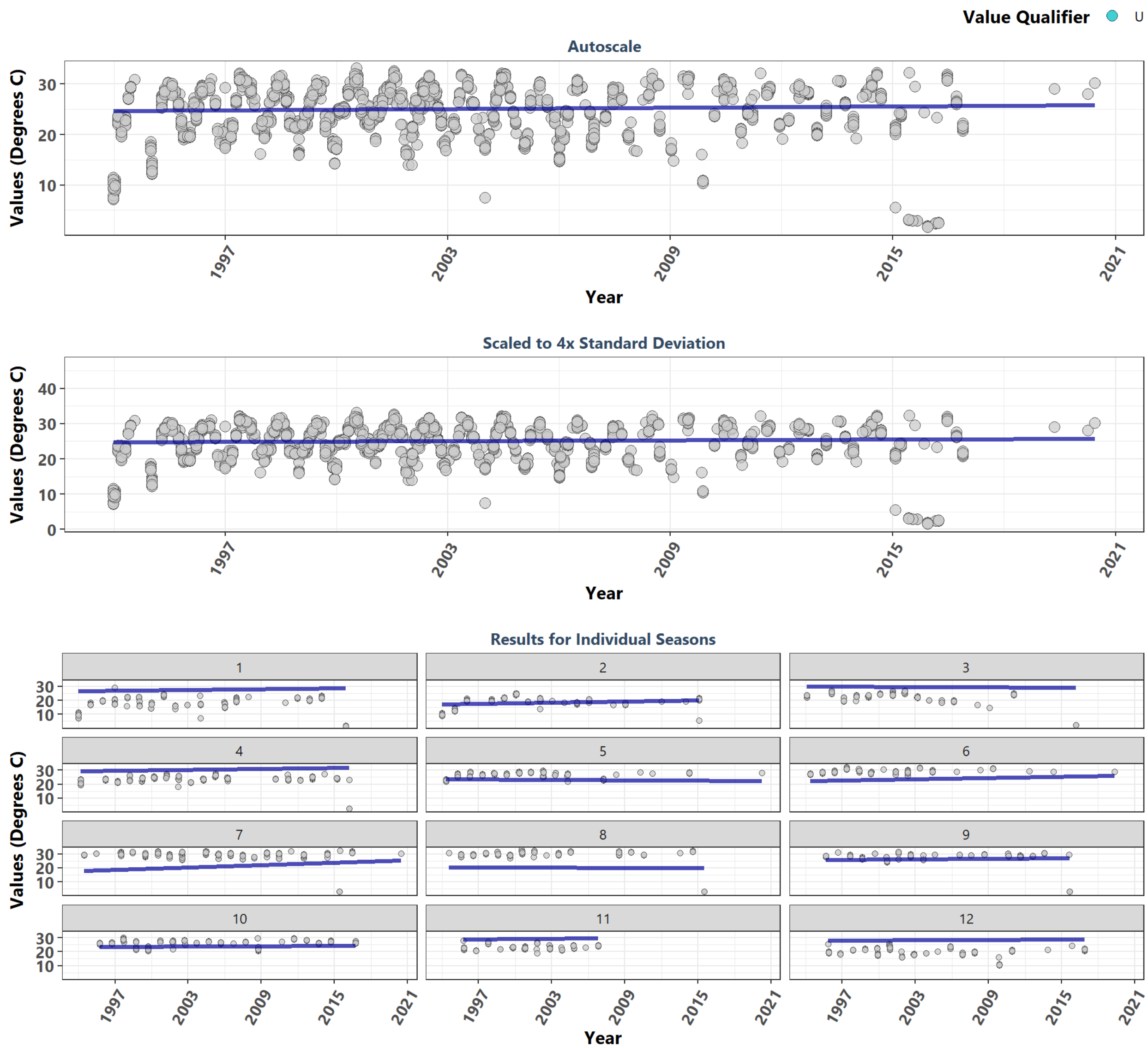
Indian River-Malabar to Vero Beach Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	10467	25.00	0.0029	0.0000	23.1333	0.1	0.9358	173.4	0	0
1	895	18.19	-0.0286	-0.0200	22.2200	-3.6	0.0003	NA	NA	0
2	722	19.07	0.1313	0.0500	27.0000	2.3	0.0207	NA	NA	0
3	686	21.60	0.0235	0.0167	23.2667	-1.1	0.2609	NA	NA	0
4	1100	23.80	0.1474	0.1062	22.1000	1.2	0.2417	NA	NA	0
5	871	26.44	-0.0004	0.0000	23.0000	-4.1	0.0000	NA	NA	0
6	780	28.70	0.0562	0.0217	28.0500	2.4	0.0186	NA	NA	0
7	1102	29.30	-0.0812	-0.0804	20.7616	-4.8	0.0000	NA	NA	0
8	958	29.50	0.0453	0.0333	19.0000	-4.6	0.0000	NA	NA	0
9	874	28.60	-0.0970	-0.0417	30.6800	5.8	0.0000	NA	NA	0
10	999	25.50	0.0574	0.0510	17.4890	7.0	0.0000	NA	NA	0
11	780	23.00	-0.0991	-0.0433	30.8867	0.0	0.9878	NA	NA	0
12	700	20.00	-0.0919	-0.0500	27.9900	1.8	0.0722	NA	NA	0

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

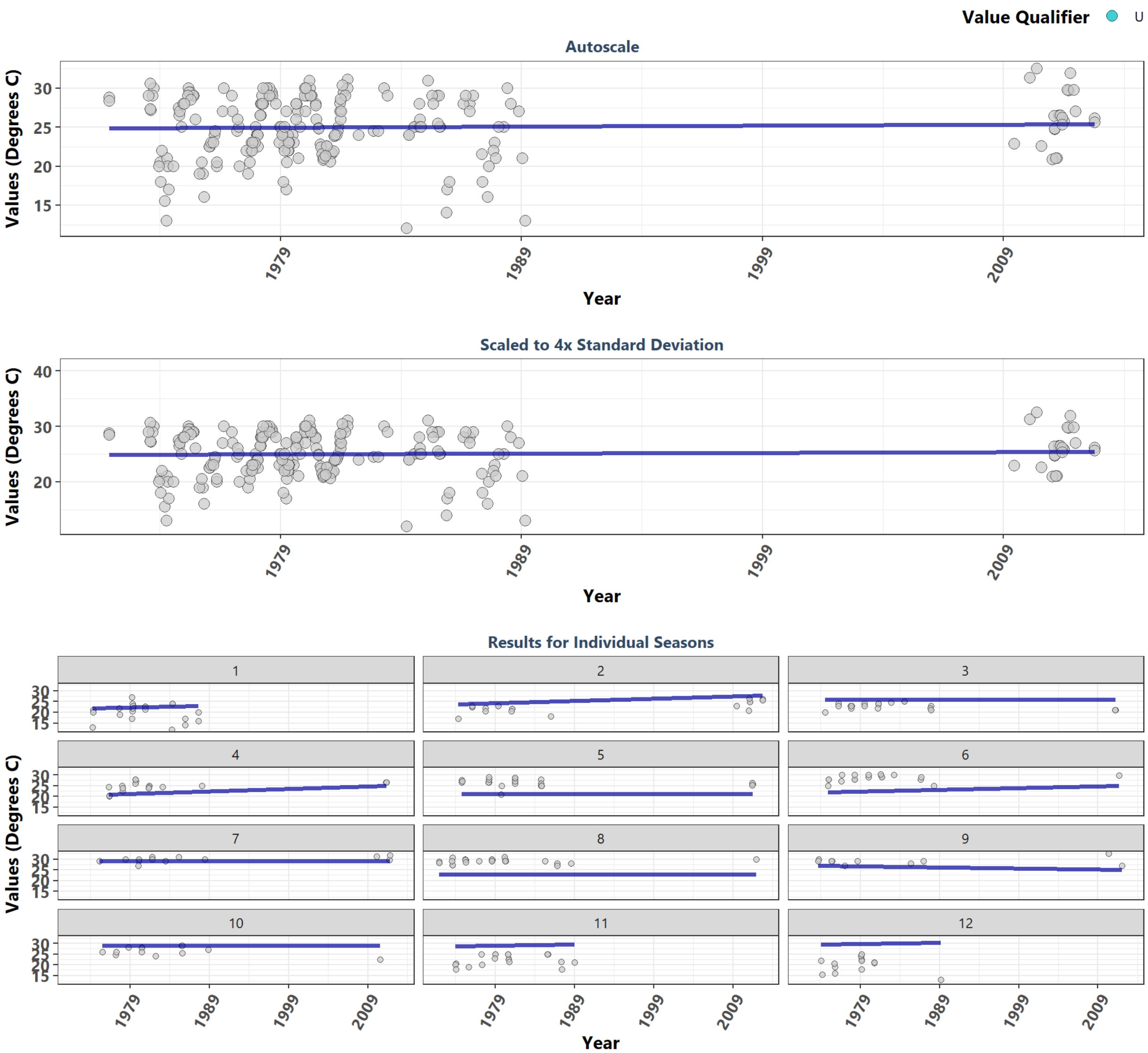
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2273	24.90	0.0993	0.0420	24.5693	5.9	0.0000	66.3	0	1
1	248	18.40	0.2166	0.1000	26.3100	3.1	0.0021	NA	NA	1
2	165	19.61	0.1311	0.1418	16.8400	4.4	0.0000	NA	NA	1
3	167	23.10	-0.1332	-0.0500	30.3000	-0.9	0.3660	NA	NA	-1
4	268	23.70	0.2384	0.1000	29.1000	2.0	0.0418	NA	NA	1
5	184	27.21	-0.0470	-0.0467	23.5667	4.4	0.0000	NA	NA	-1
6	116	29.07	0.2493	0.1500	21.6500	2.8	0.0044	NA	NA	1
7	267	29.60	0.2295	0.2900	17.0000	-3.3	0.0012	NA	NA	1
8	147	29.90	-0.0209	-0.0125	20.2625	4.3	0.0000	NA	NA	-1
9	167	28.40	0.1427	0.0560	25.5720	1.9	0.0527	NA	NA	1
10	271	26.30	0.0833	0.0353	23.2765	3.5	0.0004	NA	NA	1
11	115	23.00	0.1775	0.0850	28.4730	4.0	0.0001	NA	NA	1
12	158	20.15	0.1006	0.0391	27.9700	-0.4	0.6964	NA	NA	1

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

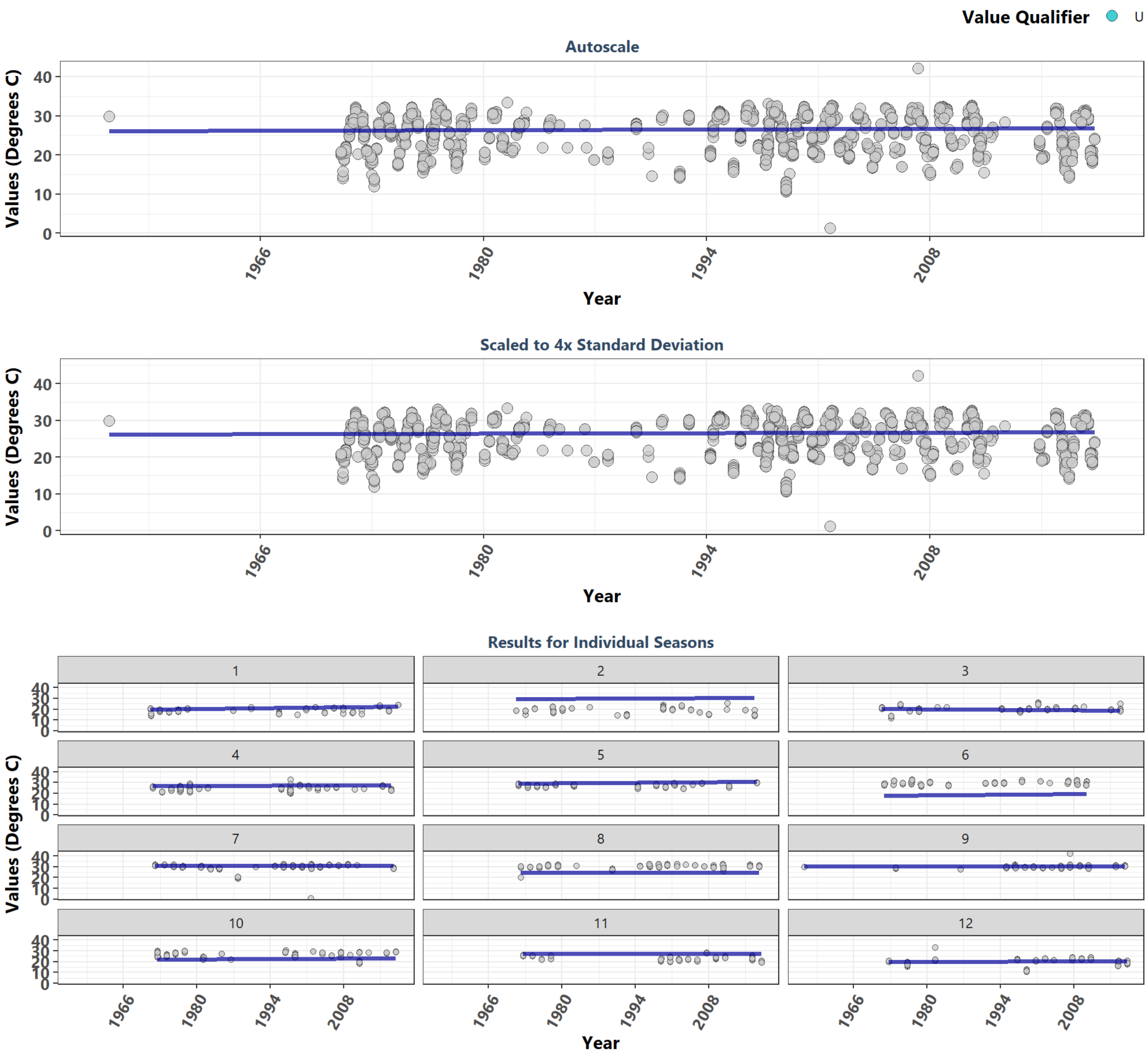
Jensen Beach to Jupiter Inlet Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	218	25.00	0.0858	0.0188	24.5000	1.6	0.1149	16.3	0.1313	0
1	22	21.15	0.1345	0.1181	19.2125	-0.1	0.9307	NA	NA	0
2	19	22.50	0.4035	0.1667	20.0000	2.0	0.0430	NA	NA	0
3	20	23.00	0.0128	0.0000	26.0000	-0.2	0.8672	NA	NA	0
4	19	24.50	0.0588	0.2000	15.9000	2.5	0.0141	NA	NA	0
5	22	26.50	-0.0173	0.0000	21.1500	-1.6	0.1180	NA	NA	0
6	18	29.00	0.3392	0.1435	18.4827	1.1	0.2738	NA	NA	0
7	13	30.00	-0.2000	0.0000	29.0000	1.8	0.0673	NA	NA	0
8	26	29.00	-0.0316	0.0000	23.0000	0.2	0.8548	NA	NA	0
9	10	29.00	-0.2381	-0.0803	28.7485	-0.8	0.4368	NA	NA	0
10	13	26.00	0.0277	0.0000	29.0000	0.0	1.0000	NA	NA	0
11	19	22.40	0.1895	0.0818	26.7909	0.8	0.4284	NA	NA	0
12	17	21.10	0.3846	0.0950	27.3400	0.3	0.7561	NA	NA	0

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

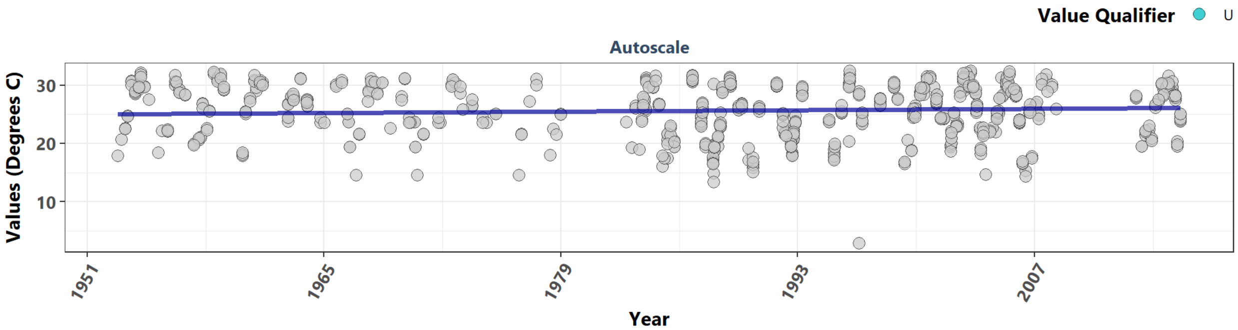
Lemon Bay Aquatic Preserve

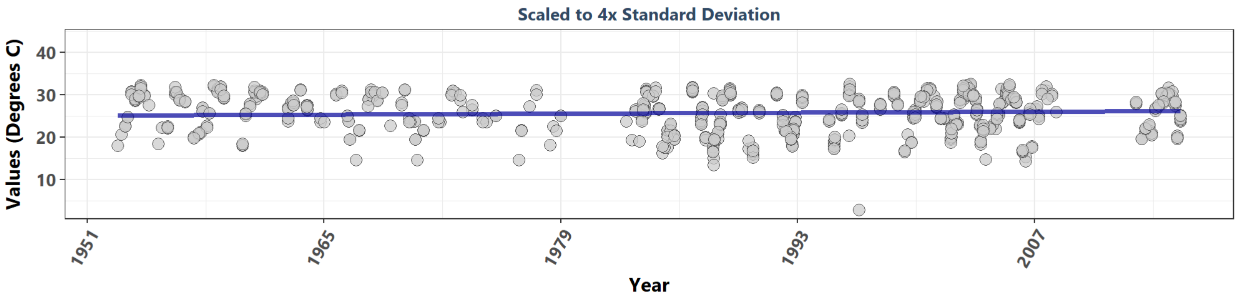


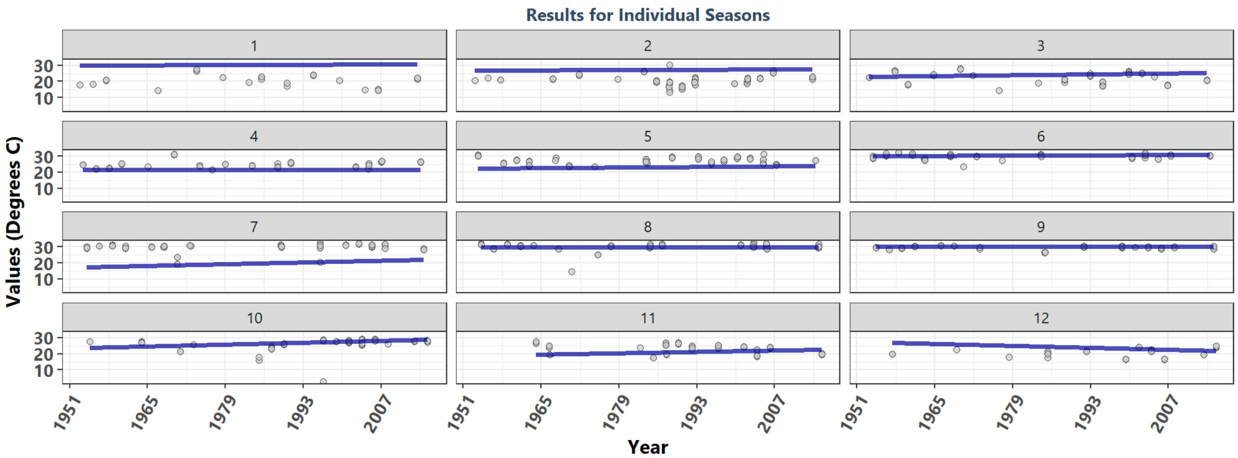
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1954	26.80	0.0594	0.0214	25.4000	3.6	0.0003	28.3	0.0029	1
1	91	18.70	0.1855	0.1045	15.4227	2.1	0.0332	NA	NA	1
2	96	19.40	0.1192	0.0426	27.4918	-1.1	0.2736	NA	NA	1
3	149	20.22	-0.0757	-0.0438	21.8938	0.8	0.4345	NA	NA	-1
4	185	24.60	0.0369	0.0111	26.7000	0.0	0.9761	NA	NA	1
5	111	27.30	0.2032	0.0667	26.2000	0.6	0.5660	NA	NA	1
6	152	29.77	0.1512	0.0714	14.7714	2.2	0.0286	NA	NA	1
7	238	30.40	0.0186	0.0000	30.8000	0.0	0.9889	NA	NA	-1
8	185	30.80	-0.0015	0.0000	24.6000	0.4	0.7062	NA	NA	-1
9	199	30.00	-0.0006	0.0000	30.4000	4.3	0.0000	NA	NA	-1
10	208	27.30	0.0683	0.0500	19.8000	-0.3	0.7303	NA	NA	1
11	189	22.70	-0.0160	-0.0053	27.5947	1.4	0.1598	NA	NA	-1
12	151	21.80	0.0430	0.0125	19.5075	3.4	0.0007	NA	NA	1

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

Matlacha Pass Aquatic Preserve



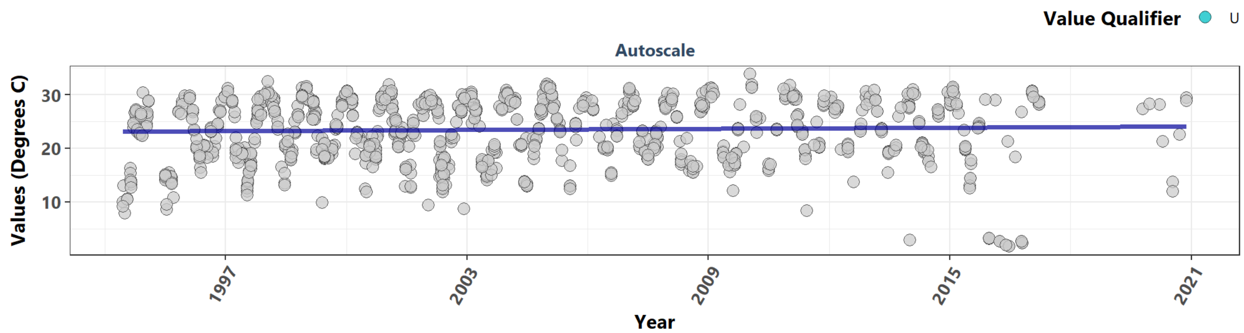


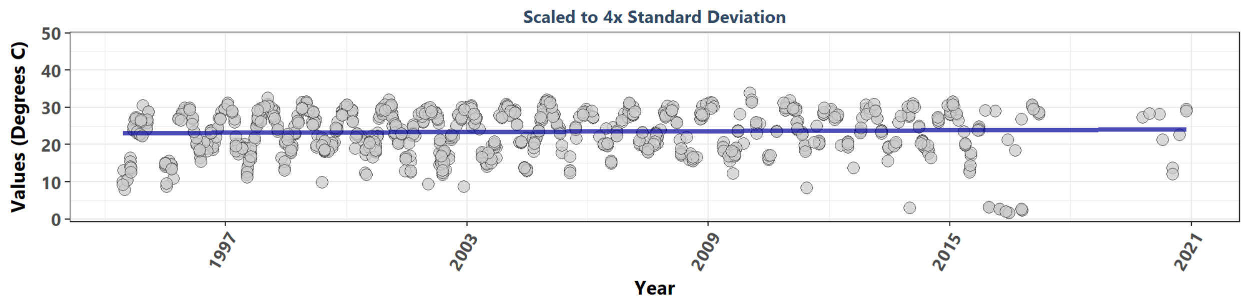


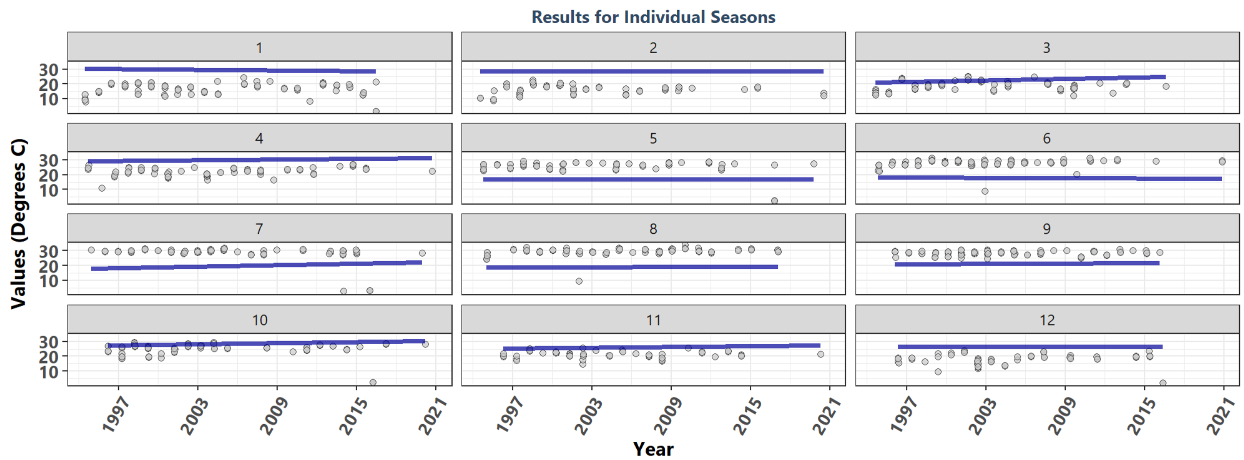
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	765	26.50	0.1128	0.0400	23.4087	4.7	0.0000	42.8	0	1
1	29	21.40	0.1307	0.0294	28.6029	0.0	0.9849	NA	NA	1
2	90	20.25	0.0491	0.0263	25.7737	2.5	0.0141	NA	NA	1
3	61	23.40	0.2498	0.0760	19.9440	0.7	0.4681	NA	NA	1
4	55	24.20	0.0049	0.0062	21.0437	2.7	0.0069	NA	NA	1
5	73	27.30	0.0634	0.0742	19.0242	0.6	0.5403	NA	NA	1
6	58	30.00	0.1176	0.0231	29.2077	1.5	0.1462	NA	NA	1
7	90	30.50	0.1728	0.1750	10.1000	1.7	0.0961	NA	NA	1
8	75	30.30	0.0094	0.0000	29.8000	0.2	0.8646	NA	NA	-1
9	51	29.80	0.0137	0.0040	30.0800	0.1	0.9282	NA	NA	1
10	72	27.43	0.3697	0.1840	16.2060	4.6	0.0000	NA	NA	1
11	59	24.10	0.3484	0.1500	12.4750	-2.9	0.0042	NA	NA	1
12	52	21.70	-0.2542	-0.1944	34.9889	3.7	0.0002	NA	NA	-1

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

Mosquito Lagoon Aquatic Preserve



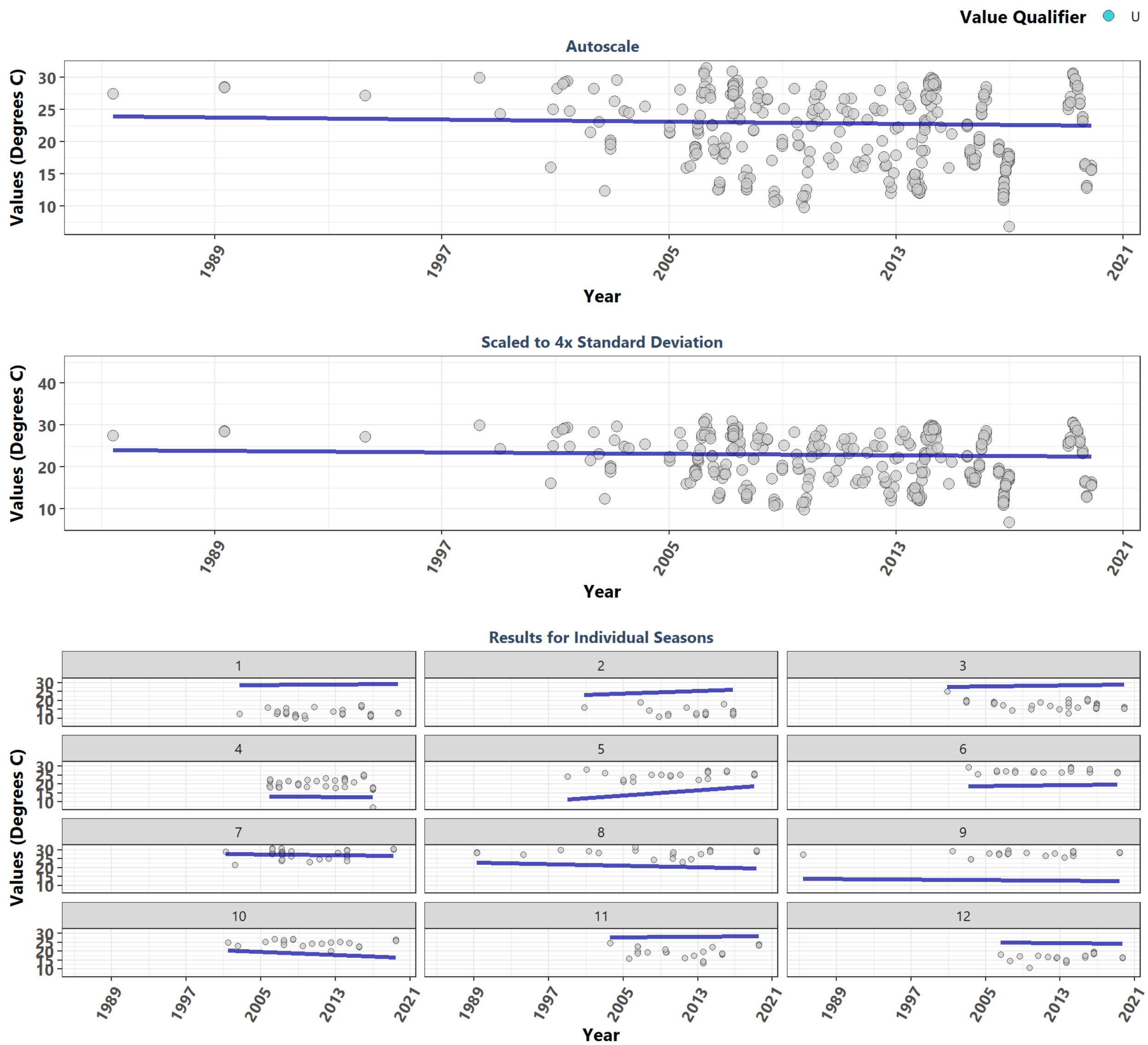




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1237	24.10	0.0689	0.0357	23.0014	3.4	0.0007	53.1	0	1
1	130	17.90	-0.2261	-0.0833	30.6333	-1.0	0.3230	NA	NA	-1
2	73	16.92	-0.0026	0.0000	28.4000	0.1	0.8859	NA	NA	-1
3	94	19.00	0.2737	0.1706	20.4412	2.5	0.0139	NA	NA	1
4	100	23.00	0.2326	0.0800	28.8100	4.0	0.0001	NA	NA	1
5	88	26.45	0.0118	0.0095	16.8245	0.0	0.9942	NA	NA	1
6	105	28.10	-0.0586	-0.0429	18.4357	3.7	0.0002	NA	NA	-1
7	104	29.55	0.1720	0.1677	17.1552	-3.4	0.0006	NA	NA	1
8	108	29.85	0.0166	0.0167	18.6000	3.6	0.0003	NA	NA	1
9	127	28.40	0.0622	0.0300	20.9100	0.0	0.9666	NA	NA	1
10	116	25.95	0.2421	0.1125	26.8625	2.0	0.0413	NA	NA	1
11	84	21.30	0.1279	0.0714	25.0929	0.8	0.4021	NA	NA	1
12	108	18.80	-0.0008	0.0000	26.4500	0.3	0.7996	NA	NA	-1

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

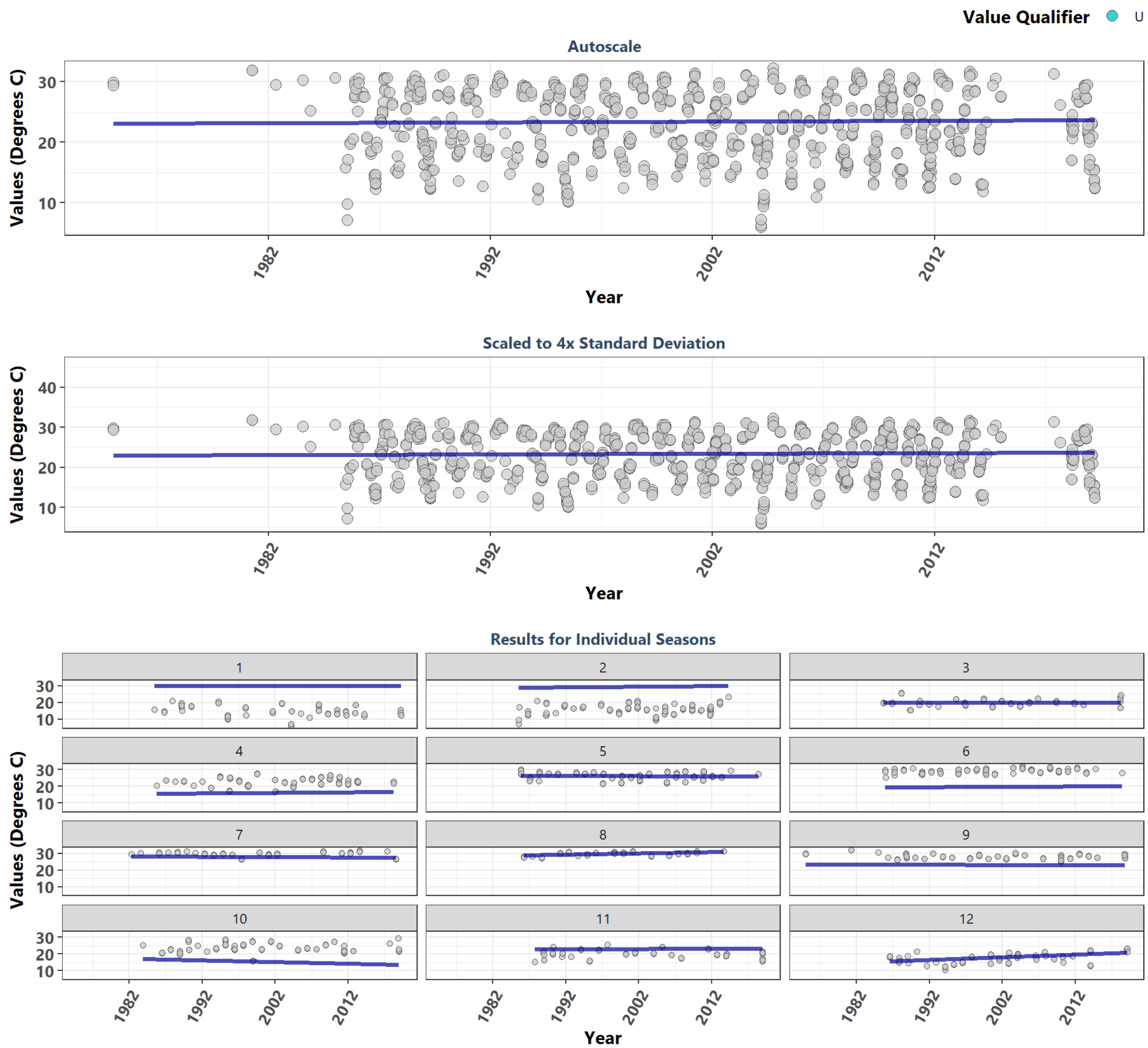
Nassau River-St. Johns River Marshes Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	381	21.05	-0.0161	-0.0413	24.1775	-1.5	0.1226	36.7	0.0001	0
1	43	12.60	0.2000	0.0387	27.7500	-0.8	0.4215	NA	NA	0
2	23	12.70	0.2808	0.1853	19.5878	-0.4	0.6822	NA	NA	0
3	53	17.11	0.0321	0.0728	26.2568	-4.2	0.0000	NA	NA	0
4	41	20.32	-0.0632	-0.0467	14.2400	-1.1	0.2810	NA	NA	0
5	29	25.70	0.3011	0.3700	4.7100	2.1	0.0319	NA	NA	0
6	23	27.08	0.0433	0.0595	17.5448	-1.0	0.2981	NA	NA	0
7	40	28.15	-0.1581	-0.0475	28.6483	0.3	0.7662	NA	NA	0
8	25	28.91	-0.1171	-0.1063	23.7200	1.4	0.1632	NA	NA	0
9	23	28.11	-0.0853	-0.0389	13.8833	1.4	0.1731	NA	NA	0
10	23	24.80	-0.3861	-0.2150	24.6350	-0.6	0.5236	NA	NA	0
11	25	19.45	0.2055	0.0441	26.8748	0.3	0.7770	NA	NA	0
12	33	17.29	-0.0988	-0.0500	26.3000	2.5	0.0114	NA	NA	0

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

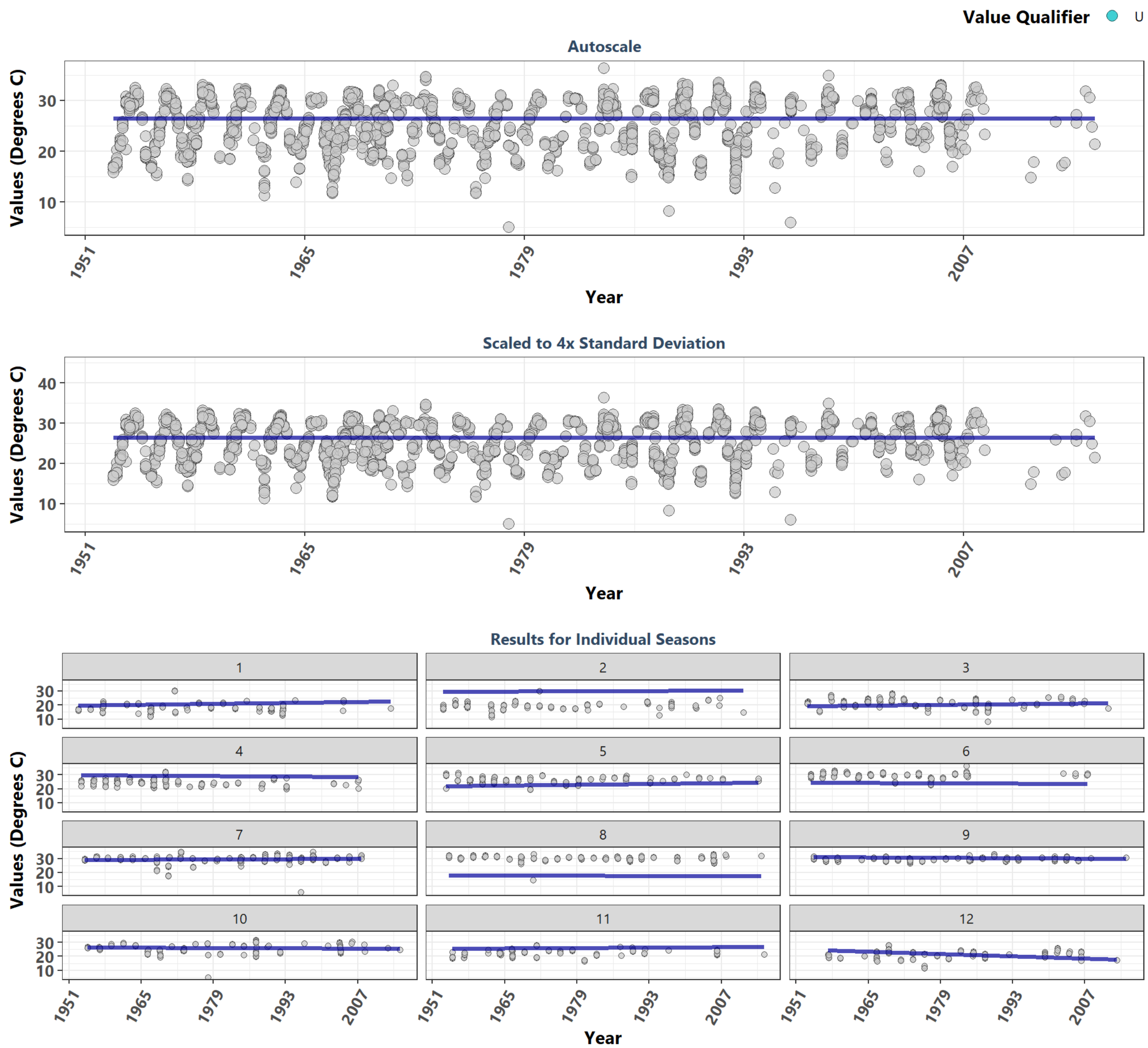
Nature Coast Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	840	23.10	0.0419	0.0167	22.8562	1.5	0.1289	32.3	0.0007	0
1	67	14.90	0.0307	0.0044	29.7687	-2.2	0.0264	NA	NA	0
2	97	16.20	0.1625	0.0500	27.9000	0.9	0.3885	NA	NA	0
3	58	19.70	-0.0148	0.0000	20.0000	0.7	0.4839	NA	NA	0
4	86	22.95	0.0595	0.0487	14.4961	0.4	0.6781	NA	NA	0
5	92	26.25	-0.0172	-0.0100	26.5900	-0.2	0.8102	NA	NA	0
6	90	29.50	0.0635	0.0304	18.6669	2.3	0.0231	NA	NA	0
7	44	29.90	-0.1091	-0.0286	28.7357	0.3	0.7762	NA	NA	0
8	41	29.90	0.3183	0.1000	26.7000	2.9	0.0034	NA	NA	0
9	76	27.75	-0.0356	-0.0167	23.6250	-1.4	0.1619	NA	NA	0
10	72	23.10	-0.1854	-0.1250	19.4000	-0.4	0.6610	NA	NA	0
11	55	20.00	0.0306	0.0250	22.0875	-0.2	0.8783	NA	NA	0
12	62	17.95	0.3184	0.2000	11.1500	3.7	0.0003	NA	NA	0

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

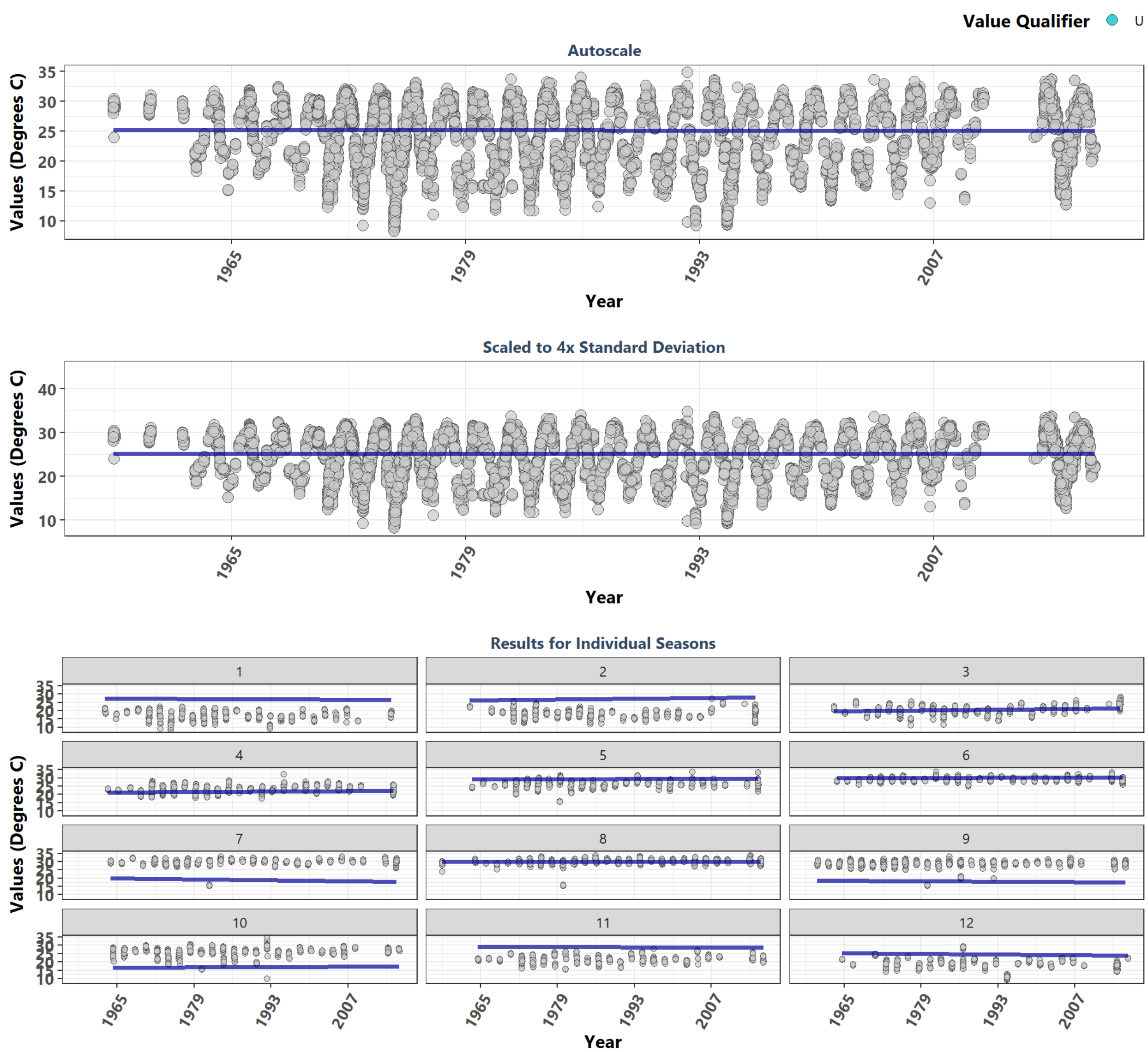
Pine Island Sound Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2444	26.40	0.0015	0.0000	26.4562	-0.4	0.7149	131.9	0	0
1	175	17.70	0.1868	0.1133	14.8800	-0.5	0.6111	NA	NA	0
2	137	19.70	0.0646	0.0250	28.6000	2.5	0.0113	NA	NA	0
3	254	21.91	0.1457	0.0750	16.1750	-6.7	0.0000	NA	NA	0
4	192	24.05	-0.0857	-0.0500	31.7000	-1.1	0.2526	NA	NA	0
5	166	26.20	0.2202	0.0956	18.0637	-0.9	0.3690	NA	NA	0
6	168	29.35	-0.0554	-0.0375	25.8125	-1.7	0.0974	NA	NA	0
7	359	30.00	0.1836	0.0429	27.1000	1.8	0.0669	NA	NA	0
8	257	30.30	-0.0257	-0.0133	18.3933	-4.8	0.0000	NA	NA	0
9	268	29.50	-0.2011	-0.0556	33.4111	4.5	0.0000	NA	NA	0
10	174	26.10	-0.0469	-0.0250	27.3750	1.3	0.1860	NA	NA	0
11	142	22.84	0.0669	0.0500	23.3500	3.9	0.0001	NA	NA	0
12	152	21.00	-0.2795	-0.2733	35.8450	3.4	0.0006	NA	NA	0

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

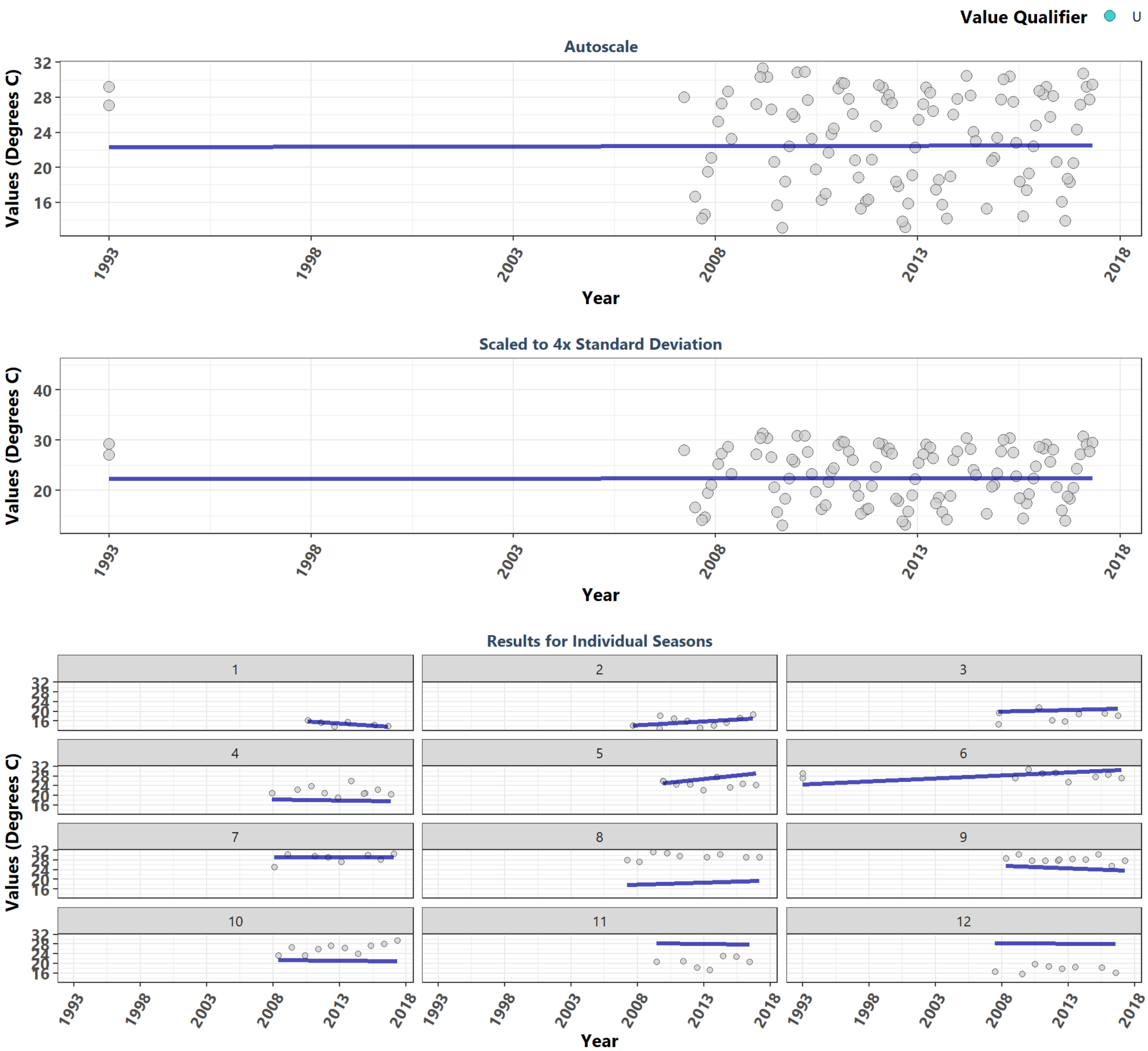
Pinellas County Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	15438	28.07	0.0099	-0.0036	25.3499	-1.3	0.2111	180.4	0	0
1	700	16.79	-0.0400	-0.0190	28.0333	1.9	0.0628	NA	NA	0
2	678	18.23	0.1421	0.0730	23.2320	-2.2	0.0263	NA	NA	0
3	634	20.28	0.1016	0.0614	17.1471	3.8	0.0001	NA	NA	0
4	964	24.76	0.1070	0.0436	19.2745	-4.2	0.0000	NA	NA	0
5	1002	27.10	0.0638	0.0148	28.4471	-1.9	0.0566	NA	NA	0
6	758	29.20	0.0433	0.0143	29.4200	2.6	0.0084	NA	NA	0
7	660	30.00	-0.1366	-0.0756	23.0244	1.3	0.1804	NA	NA	0
8	1851	30.12	0.0347	0.0091	29.5364	2.8	0.0049	NA	NA	0
9	5320	29.06	-0.0568	-0.0367	20.0950	-3.2	0.0014	NA	NA	0
10	1600	26.59	0.0469	0.0350	15.0050	8.6	0.0000	NA	NA	0
11	684	21.50	-0.0289	-0.0129	29.6424	4.2	0.0000	NA	NA	0
12	587	19.02	-0.0878	-0.0553	27.4679	-5.0	0.0000	NA	NA	0

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

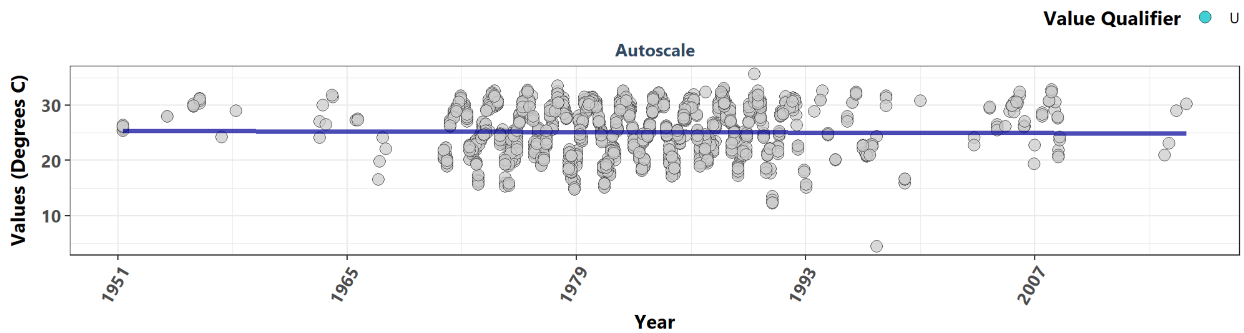
Rocky Bayou State Park Aquatic Preserve

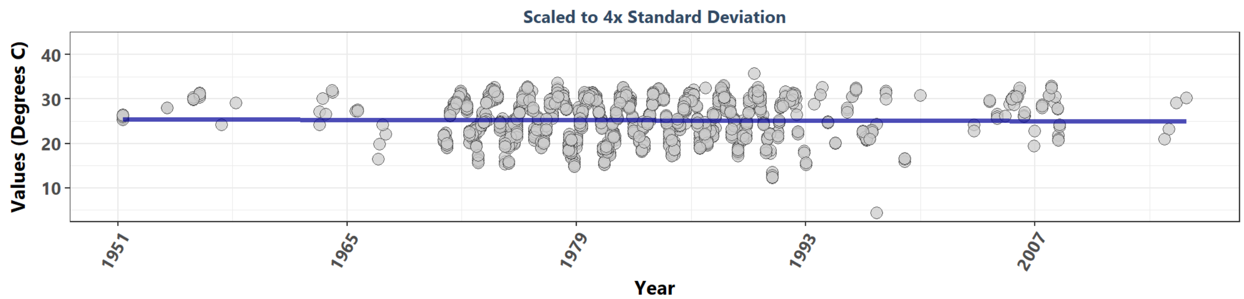


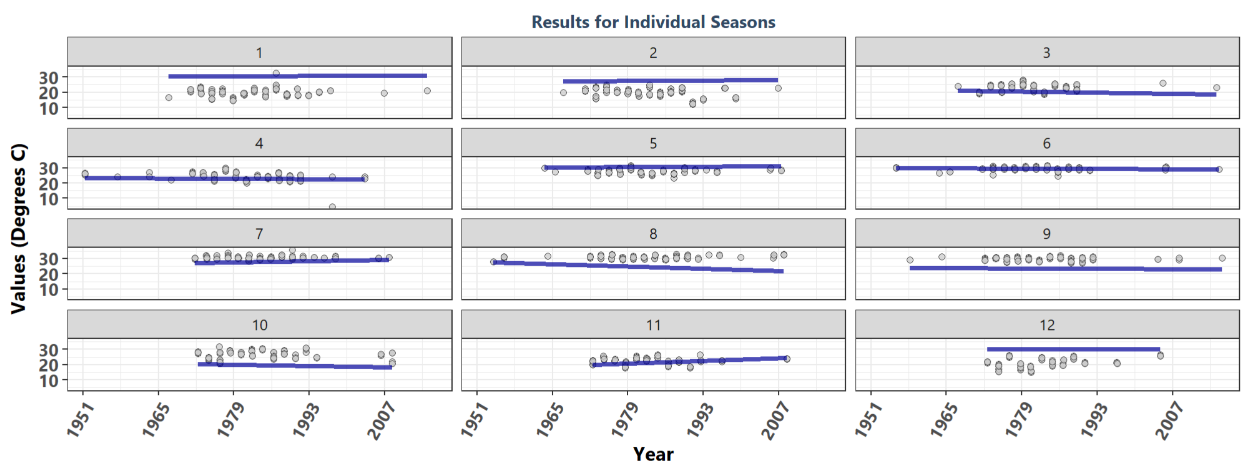
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	106	24.19	0.0141	0.0079	22.3039	0.3	0.7867	12.1	0.3594	0
1	6	14.83	-0.4667	-0.3667	22.3499	-1.1	0.2597	NA	NA	0
2	10	15.69	0.3111	0.3032	9.7812	1.2	0.2430	NA	NA	0
3	8	18.61	0.0952	0.1111	18.3891	0.0	1.0000	NA	NA	0
4	10	21.06	-0.0714	-0.0725	19.4530	-0.6	0.5263	NA	NA	0
5	9	24.67	0.6000	0.5648	15.4864	-0.8	0.4017	NA	NA	0
6	10	28.22	0.2143	0.2454	24.6041	-0.4	0.6534	NA	NA	0
7	8	29.39	0.0278	0.0069	29.0416	0.6	0.5362	NA	NA	0
8	9	29.17	0.0357	0.1667	15.3605	0.0	1.0000	NA	NA	0
9	11	28.22	-0.2500	-0.2222	29.1110	-0.6	0.5309	NA	NA	0
10	10	26.50	-0.1778	-0.0586	22.2579	2.3	0.0200	NA	NA	0
11	7	20.61	-0.1636	-0.0648	29.4542	0.2	0.8793	NA	NA	0
12	8	18.11	-0.1333	-0.0382	28.9283	-0.1	0.9015	NA	NA	0

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

Rookery Bay Aquatic Preserve



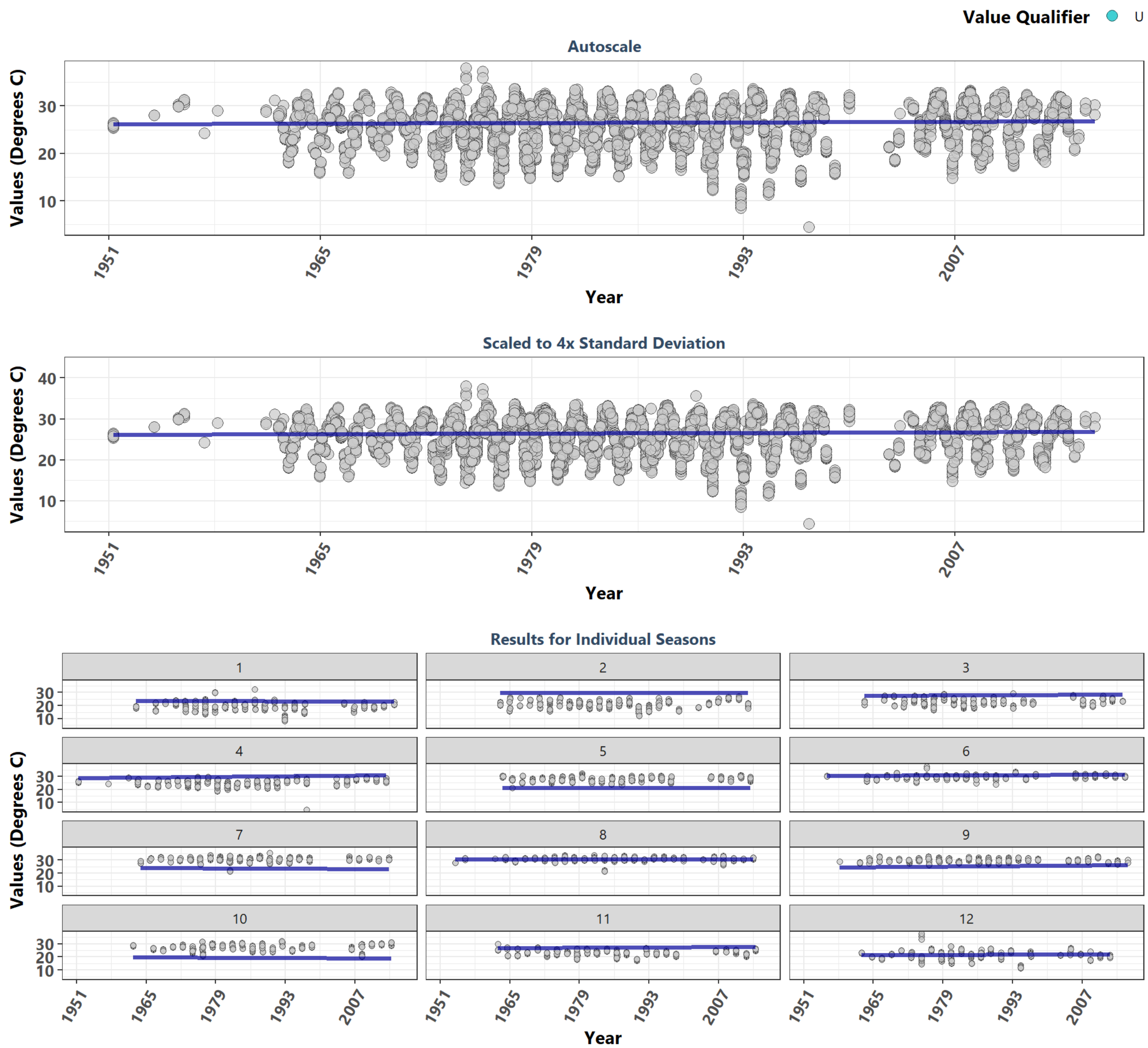




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	1665	26.20	-0.0171	-0.0125	25.7878	-1.3	0.2078	57.5	0	0
1	129	19.80	0.0762	0.0250	29.2700	-1.7	0.0940	NA	NA	0
2	142	20.40	0.0482	0.0333	25.7833	-1.6	0.1048	NA	NA	0
3	131	23.60	-0.0916	-0.1078	25.7923	-0.4	0.6994	NA	NA	0
4	156	24.60	-0.0459	-0.0400	24.8000	-5.5	0.0000	NA	NA	0
5	148	27.45	0.1109	0.0425	28.4750	0.9	0.3840	NA	NA	0
6	145	29.56	-0.1077	-0.0364	31.3782	-1.9	0.0538	NA	NA	0
7	144	30.52	0.1013	0.1143	21.7857	1.4	0.1737	NA	NA	0
8	147	30.60	-0.2973	-0.2133	35.1600	2.0	0.0456	NA	NA	0
9	140	29.98	-0.0228	-0.0240	24.8000	-0.1	0.8849	NA	NA	0
10	132	27.50	-0.0993	-0.1000	24.8000	1.7	0.0839	NA	NA	0
11	127	22.80	0.1737	0.2500	8.5000	-0.8	0.4432	NA	NA	0
12	124	21.00	-0.0083	0.0000	29.9850	2.9	0.0040	NA	NA	0

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

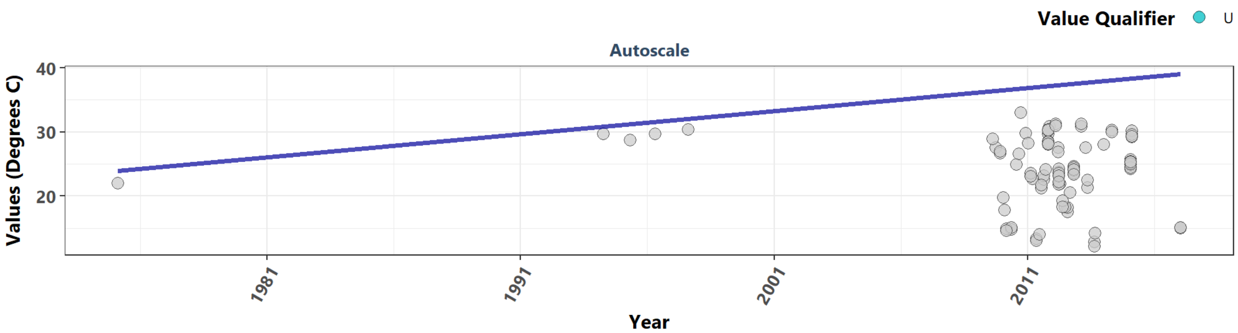
Rookery Bay National Estuarine Research Reserve

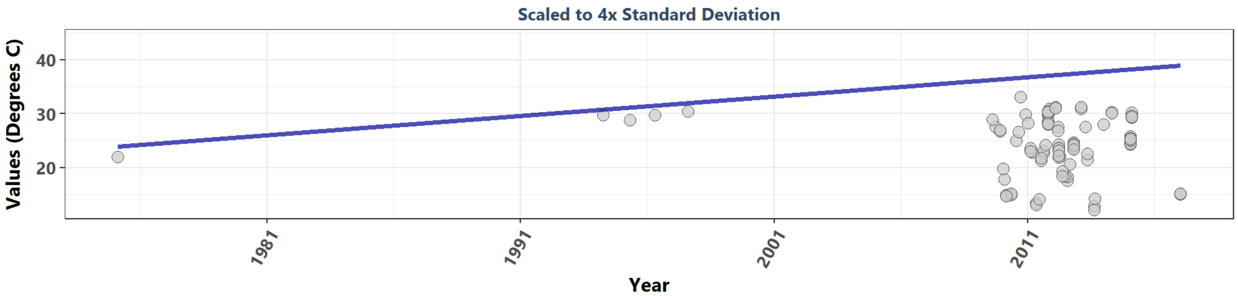


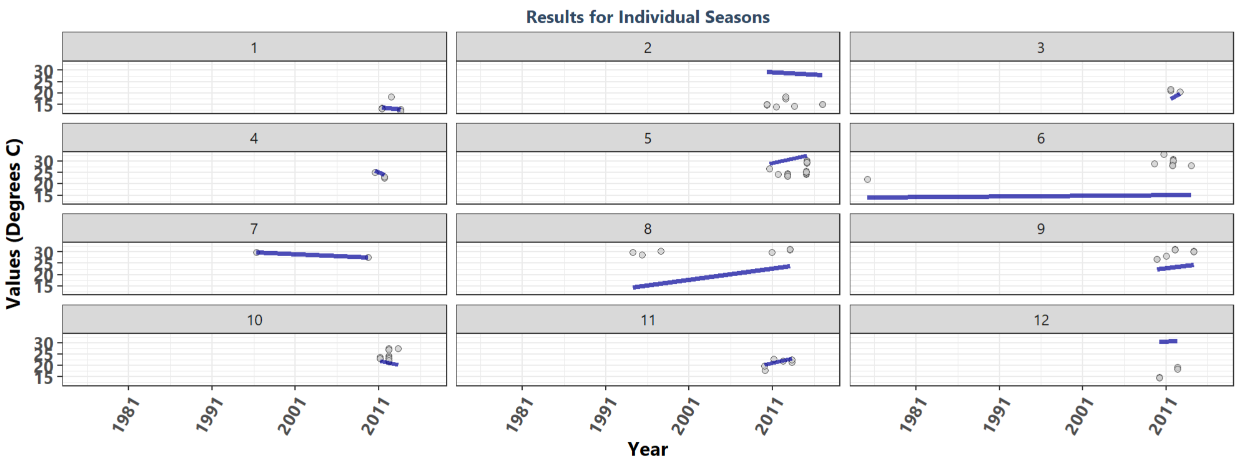
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6438	26.72	0.0434	0.0200	25.4500	5.6	0.0000	94.3	0	1
1	498	19.40	-0.0473	-0.0239	24.7201	-2.1	0.0323	NA	NA	-1
2	544	21.17	0.0177	0.0043	29.5771	-0.3	0.7431	NA	NA	1
3	459	23.50	0.0916	0.0375	26.0500	-1.4	0.1528	NA	NA	1
4	564	25.10	0.2274	0.0667	26.3000	4.0	0.0001	NA	NA	1
5	577	28.00	-0.0094	-0.0059	21.4759	3.3	0.0010	NA	NA	-1
6	531	29.70	0.1137	0.0300	29.1400	7.8	0.0000	NA	NA	1
7	559	30.70	-0.0446	-0.0333	25.2000	0.6	0.5359	NA	NA	-1
8	597	30.70	0.0175	0.0025	30.5700	4.2	0.0000	NA	NA	1
9	548	29.80	0.1114	0.0667	21.7000	0.6	0.5352	NA	NA	1
10	564	27.30	-0.0641	-0.0444	21.7111	1.9	0.0520	NA	NA	-1
11	503	23.50	0.0546	0.0308	25.7000	-1.6	0.1120	NA	NA	1
12	494	21.60	0.0125	0.0111	21.0333	0.4	0.6768	NA	NA	1

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

St. Andrews State Park Aquatic Preserve



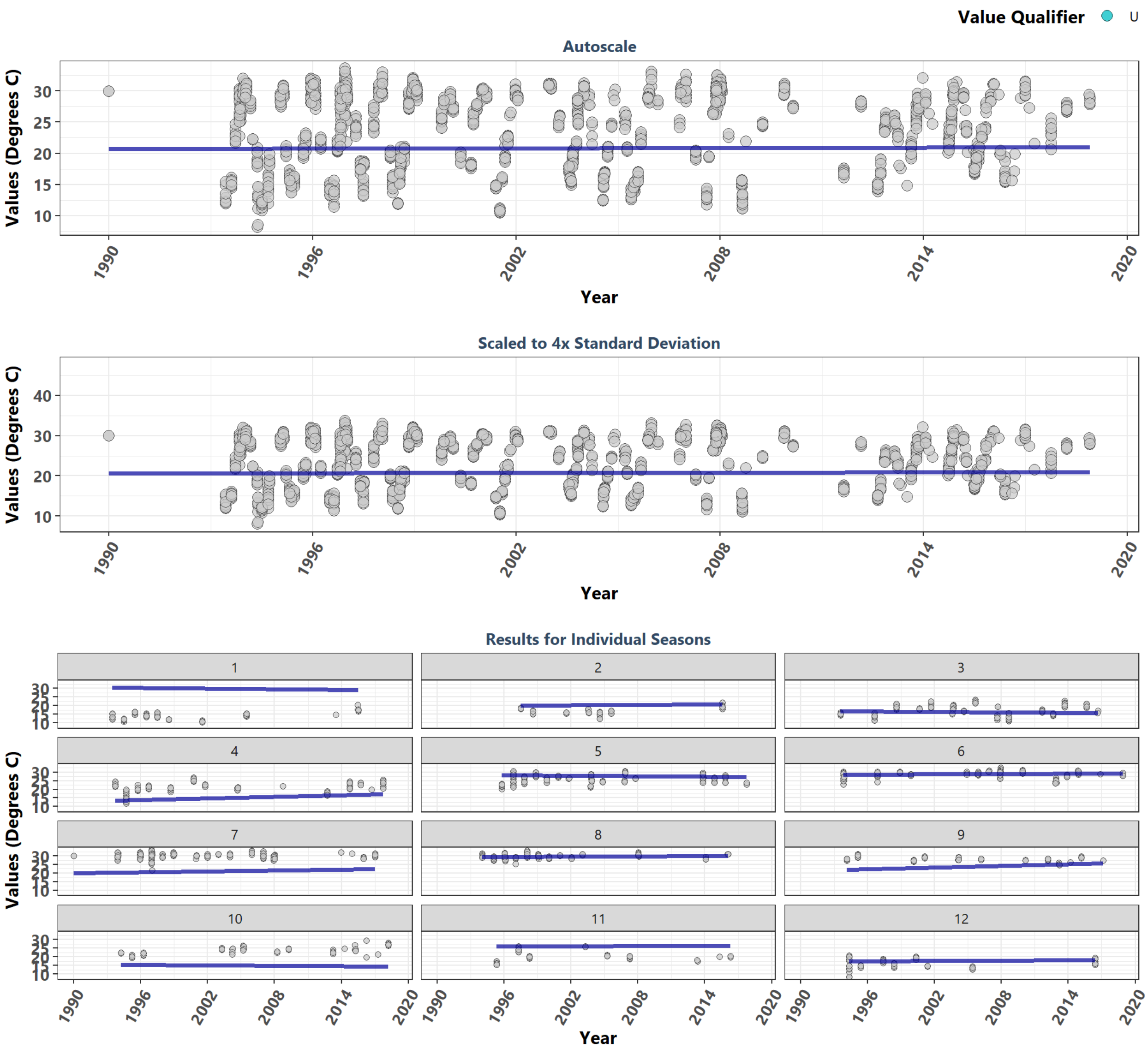




Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	104	24.15	0.1554	0.4000	20.4238	3.3	0.0010	23	0.0177	1
1	5	13.02	-0.4000	-0.3600	28.5000	-0.8	0.4334	NA	NA	-1
2	9	15.10	-0.0667	-0.2000	37.3000	0.2	0.8263	NA	NA	-1
3	3	NA	0.6667	2.0000	-65.4000	NA	NA	NA	NA	NA
4	3	NA	-0.6667	-2.0500	109.2000	NA	NA	NA	NA	NA
5	25	24.90	0.4762	0.8750	-6.7500	2.9	0.0034	NA	NA	1
6	15	28.90	0.0833	0.0333	13.6667	-0.4	0.6997	NA	NA	1
7	2	NA	-1.0000	-0.1833	34.8333	NA	NA	NA	NA	NA
8	6	30.10	0.3633	0.5500	0.1500	1.7	0.0852	NA	NA	1
9	7	30.00	0.1754	0.5000	2.3000	1.4	0.1615	NA	NA	1
10	19	23.30	-0.6667	-0.9100	59.4000	1.5	0.1461	NA	NA	-1
11	6	21.60	0.3333	0.9333	-17.1333	0.8	0.4357	NA	NA	1
12	4	NA	0.6667	0.0834	27.1809	NA	NA	NA	NA	NA

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

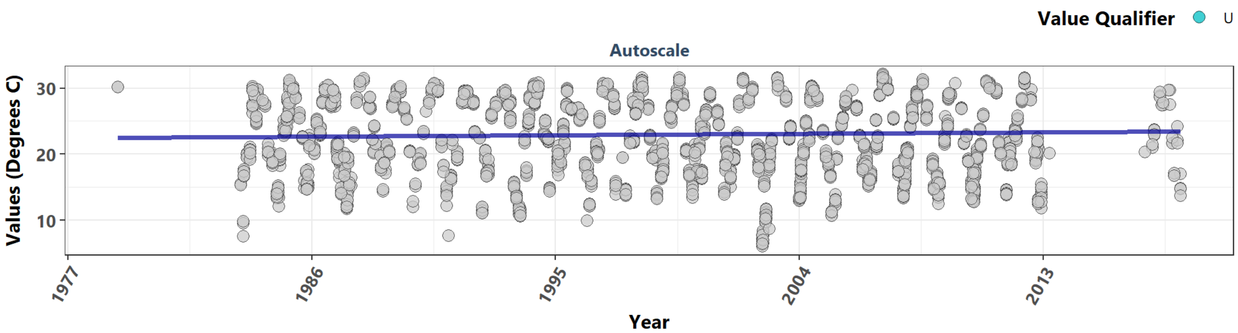
St. Joseph Bay Aquatic Preserve

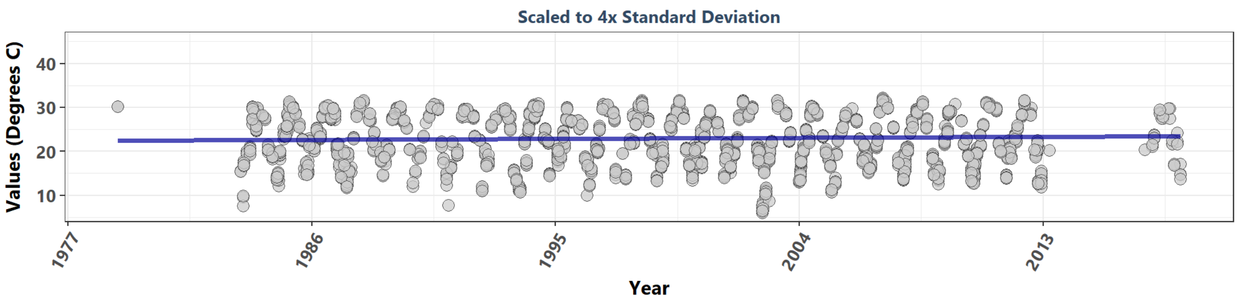


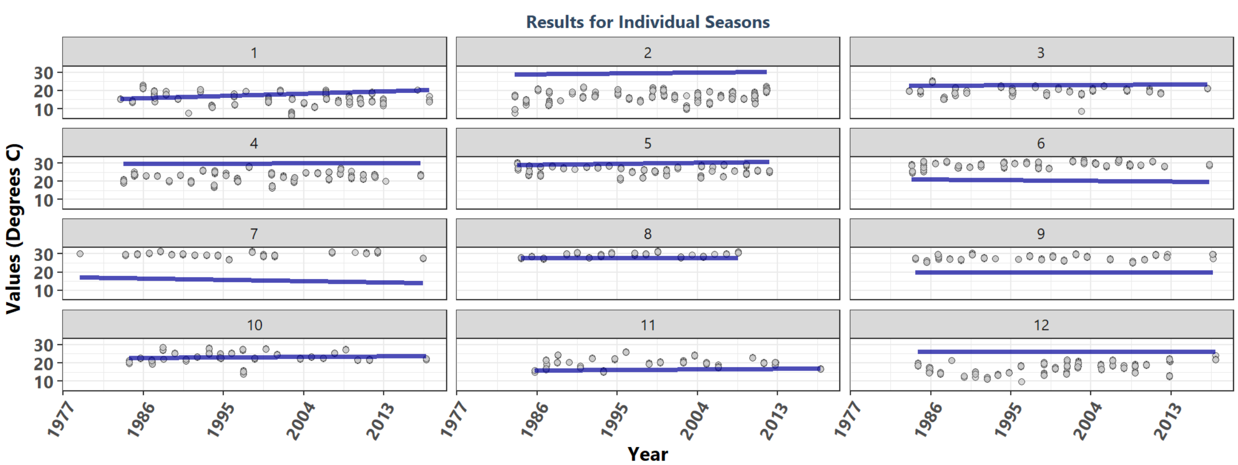
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2078	25.50	0.0424	0.0100	20.6663	1.4	0.1517	95.8	0	0
1	106	14.25	-0.1641	-0.0524	30.5714	4.2	0.0000	NA	NA	0
2	118	16.25	0.0663	0.0444	19.6222	-0.7	0.4980	NA	NA	0
3	238	17.80	-0.0416	-0.0500	16.9500	0.7	0.5089	NA	NA	0
4	174	21.00	0.2712	0.1500	12.9000	3.7	0.0002	NA	NA	0
5	214	26.20	-0.1491	-0.0450	28.4300	0.8	0.4445	NA	NA	0
6	242	28.90	0.0755	0.0350	28.2700	1.8	0.0775	NA	NA	0
7	302	30.10	0.1866	0.0875	19.9937	-4.3	0.0000	NA	NA	0
8	252	29.60	0.0805	0.0316	29.3158	1.9	0.0552	NA	NA	0
9	119	27.80	0.3953	0.1641	21.3388	-2.4	0.0154	NA	NA	0
10	122	23.80	-0.0707	-0.0474	15.7789	6.5	0.0000	NA	NA	0
11	84	20.20	0.0349	0.0188	25.9375	0.9	0.3676	NA	NA	0
12	107	15.40	0.0286	0.0222	17.4444	-1.1	0.2761	NA	NA	0

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

St. Martins Marsh Aquatic Preserve



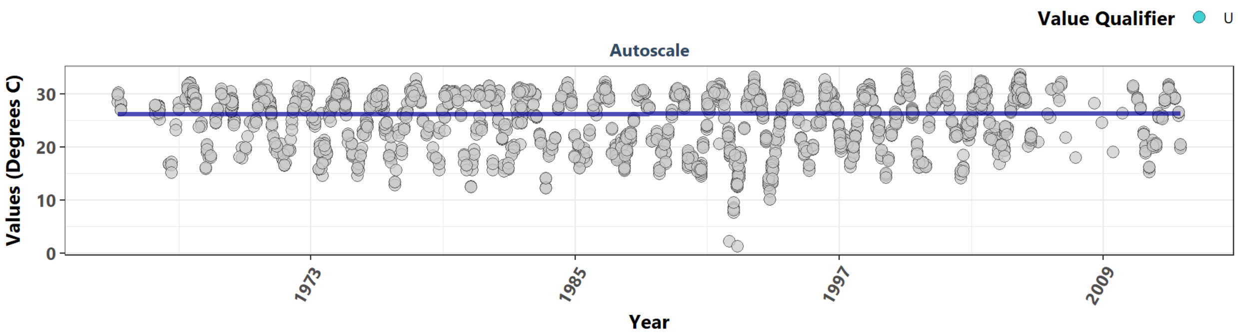


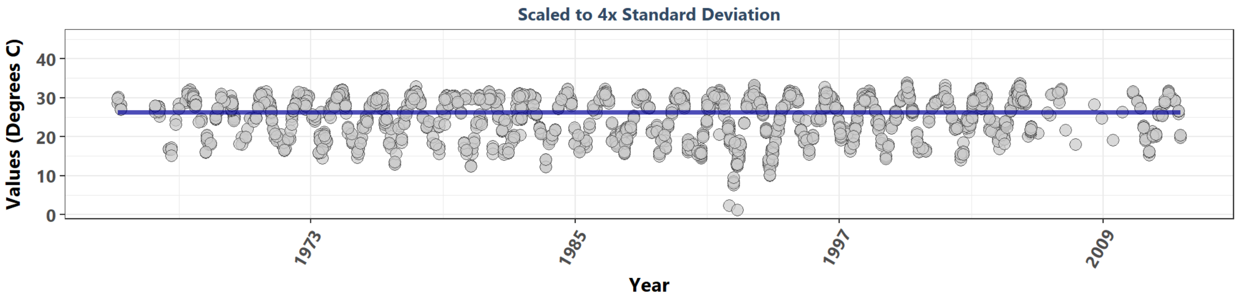


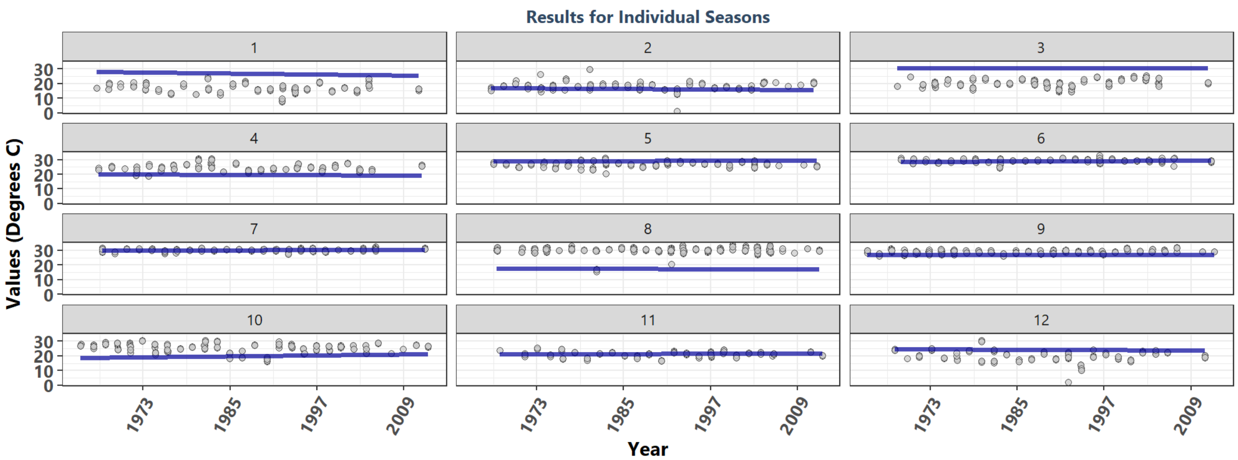
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	2800	22.80	0.0566	0.0308	22.1625	4.7	0.0000	79.4	0	1
1	238	15.15	0.2456	0.1833	12.5833	-2.7	0.0066	NA	NA	1
2	332	16.60	0.1655	0.0583	27.9833	1.6	0.1058	NA	NA	1
3	175	19.90	0.0592	0.0250	22.2750	-0.3	0.7948	NA	NA	1
4	284	23.10	0.0470	0.0143	29.3571	1.5	0.1362	NA	NA	1
5	301	26.20	0.2654	0.0765	27.7882	-0.5	0.6449	NA	NA	1
6	316	29.50	-0.1315	-0.0571	21.9571	4.4	0.0000	NA	NA	-1
7	159	29.70	-0.1180	-0.1000	18.2000	0.9	0.3787	NA	NA	-1
8	148	29.70	0.0180	0.0050	27.6700	4.8	0.0000	NA	NA	1
9	191	27.80	-0.0133	0.0000	19.9000	0.4	0.7123	NA	NA	-1
10	244	22.90	0.0592	0.0375	22.0500	1.4	0.1680	NA	NA	1
11	160	20.30	0.0593	0.0400	15.3600	-2.5	0.0133	NA	NA	1
12	252	17.90	-0.0178	-0.0087	26.4439	5.8	0.0000	NA	NA	-1

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

Terra Ceia Aquatic Preserve



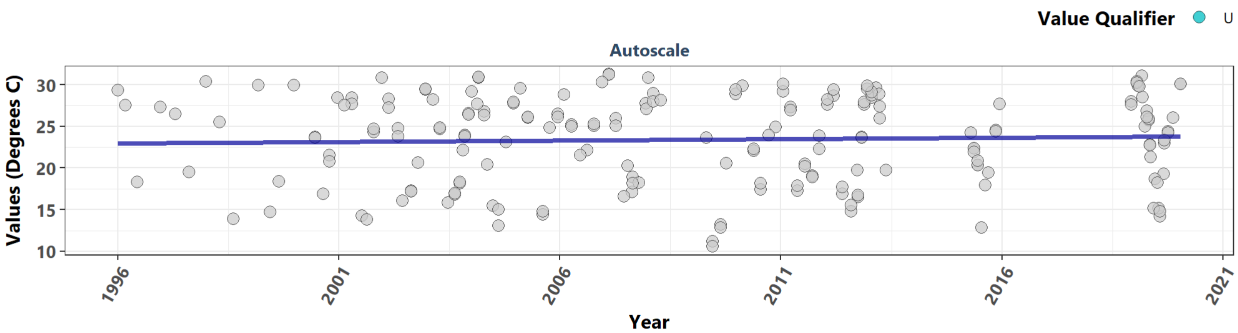


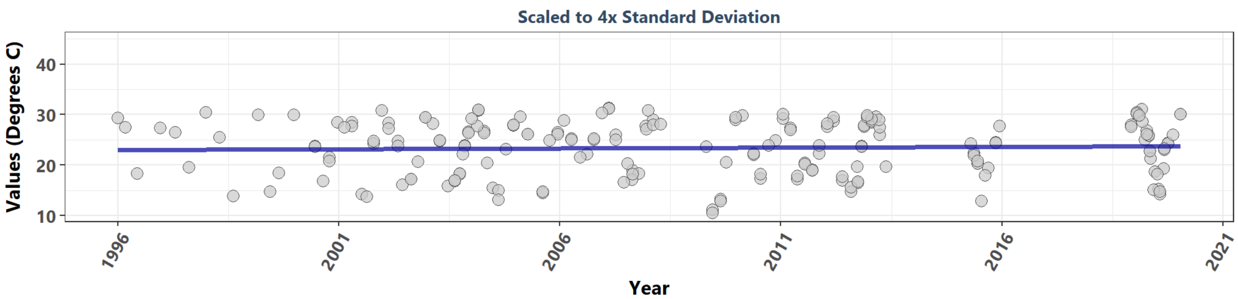


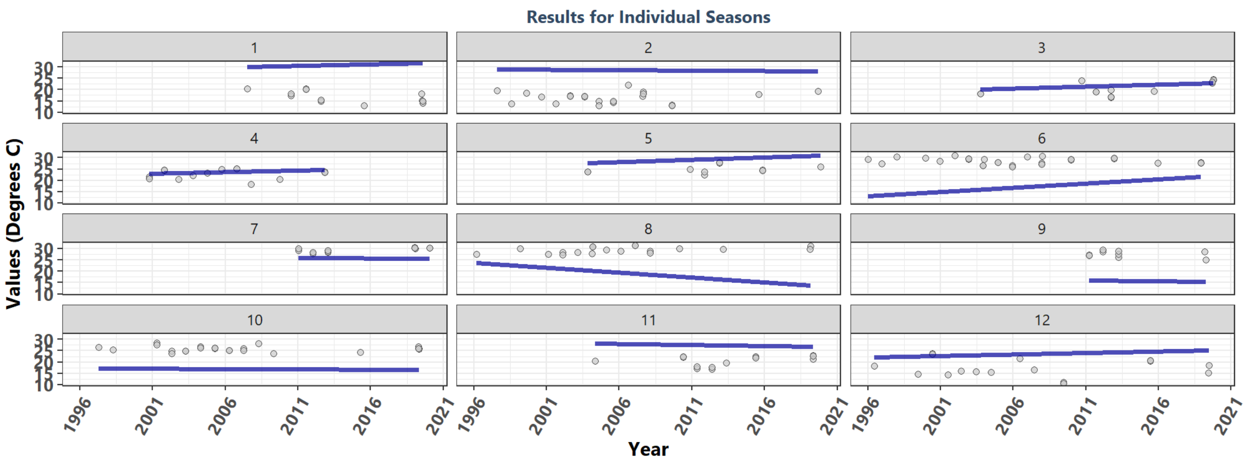
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3287	28.16	0.0022	0.0024	26.1408	0.6	0.5561	145.4	0	0
1	127	16.09	-0.2409	-0.0909	30.3173	-1.4	0.1574	NA	NA	0
2	137	17.43	-0.0845	-0.0590	18.6860	-0.6	0.5337	NA	NA	0
3	169	20.20	-0.0165	-0.0042	30.8167	2.6	0.0090	NA	NA	0
4	145	24.28	-0.0222	-0.0259	20.7119	-0.7	0.4656	NA	NA	0
5	150	27.10	0.0621	0.0189	28.4292	-0.3	0.7906	NA	NA	0
6	148	29.23	0.1655	0.0467	27.0450	1.1	0.2614	NA	NA	0
7	240	30.40	0.1031	0.0200	29.5200	2.4	0.0170	NA	NA	0
8	532	30.65	-0.0359	-0.0240	18.4380	-0.6	0.5587	NA	NA	0
9	750	28.59	-0.0147	-0.0053	27.3263	6.9	0.0000	NA	NA	0
10	596	27.59	0.1350	0.0893	16.4480	-8.9	0.0000	NA	NA	0
11	140	21.50	0.0422	0.0200	20.6200	0.7	0.4584	NA	NA	0
12	153	19.60	-0.0409	-0.0233	25.2367	-0.4	0.6841	NA	NA	0

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

Tomoka Marsh Aquatic Preserve



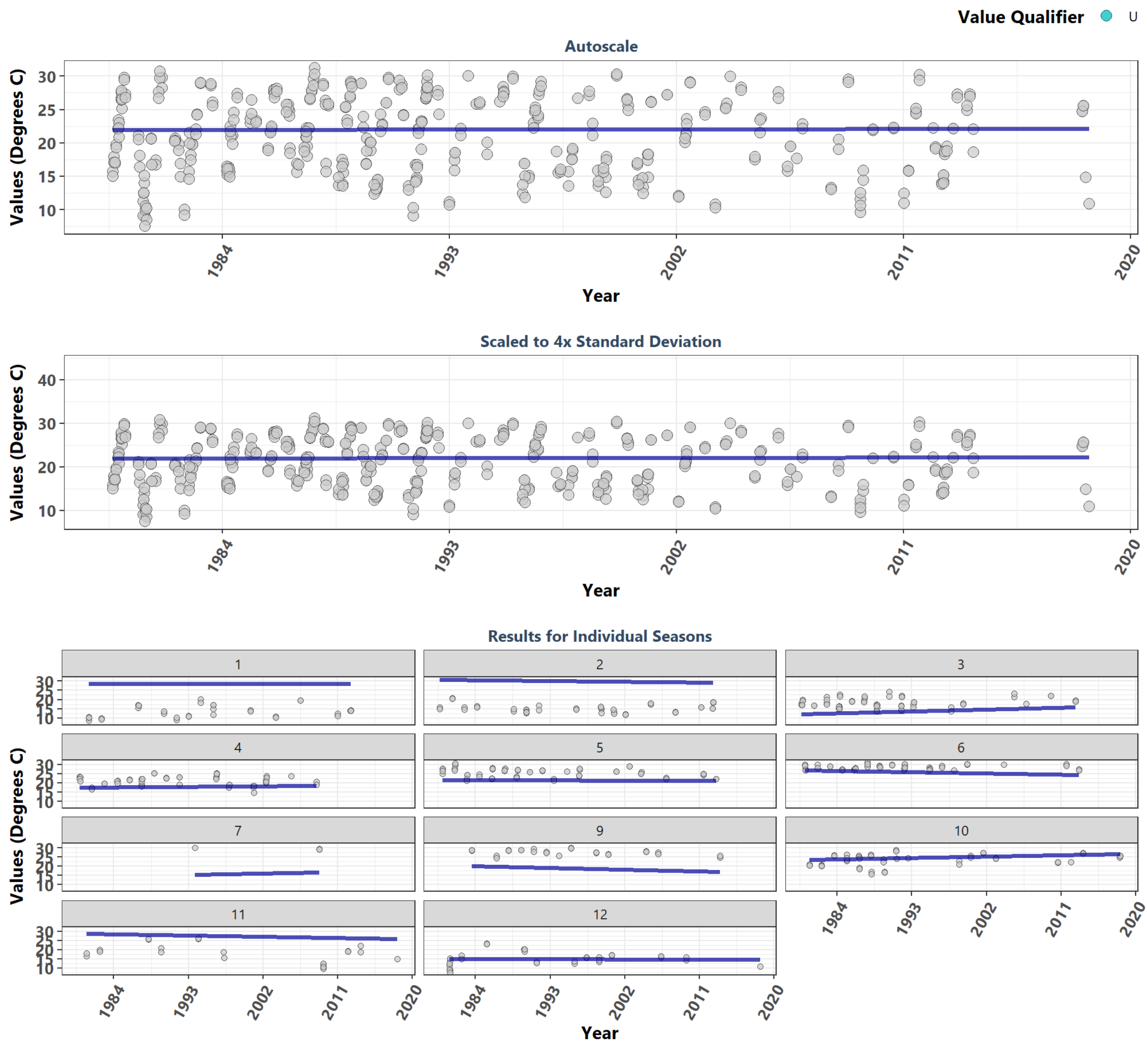




Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	193	24.50	0.0791	0.0327	22.9808	1.0	0.3177	20.3	0.0408	0
1	12	16.46	0.3095	0.1230	28.6159	-2.0	0.0483	NA	NA	0
2	21	16.96	-0.1200	-0.0373	29.0097	-0.1	0.9275	NA	NA	0
3	13	19.40	0.3516	0.1680	18.8336	2.0	0.0448	NA	NA	0
4	14	23.36	0.3455	0.1313	22.3000	0.6	0.5818	NA	NA	0
5	11	24.40	0.4091	0.1913	26.2413	1.4	0.1514	NA	NA	0
6	26	28.66	0.4231	0.3687	13.1312	-0.8	0.4007	NA	NA	0
7	12	29.49	-0.0553	-0.0197	26.0803	1.8	0.0670	NA	NA	0
8	21	29.60	-0.4394	-0.4233	23.6617	2.0	0.0506	NA	NA	0
9	9	27.41	-0.0588	-0.0989	17.5487	-0.3	0.7454	NA	NA	0
10	23	25.90	-0.0190	-0.0237	17.1730	-0.3	0.7299	NA	NA	0
11	14	21.60	-0.1111	-0.0957	29.0375	1.7	0.0847	NA	NA	0
12	17	16.56	0.1209	0.1300	22.1200	-0.3	0.7718	NA	NA	0

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

Yellow River Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	411	21.50	0.0087	0.0074	21.8333	0.4	0.6921	22.8	0.0116	0
1	34	13.70	0.0313	0.0077	28.3077	2.1	0.0382	NA	NA	0
2	39	14.80	-0.6667	-0.0700	32.0900	-0.3	0.7798	NA	NA	0
3	57	17.70	0.2496	0.1608	9.1985	0.8	0.4178	NA	NA	0
4	51	21.40	0.0739	0.0588	16.2882	-0.2	0.8383	NA	NA	0
5	56	26.05	-0.0204	-0.0167	21.8333	-2.3	0.0215	NA	NA	0
6	45	28.50	-0.2110	-0.1100	28.6900	0.3	0.7679	NA	NA	0
7	3	NA	0.1532	0.1200	12.0600	NA	NA	NA	NA	NA
9	25	27.80	-0.1667	-0.1514	23.1911	-1.8	0.0786	NA	NA	0
10	48	24.20	0.2190	0.1222	21.2667	2.2	0.0282	NA	NA	0
11	21	18.80	-0.2533	-0.1182	30.9909	-1.0	0.3007	NA	NA	0
12	32	15.00	-0.0324	-0.0207	15.4007	1.2	0.2183	NA	NA	0

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

Appendix V: Managed Area Summary Box Plots

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

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

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

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

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

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

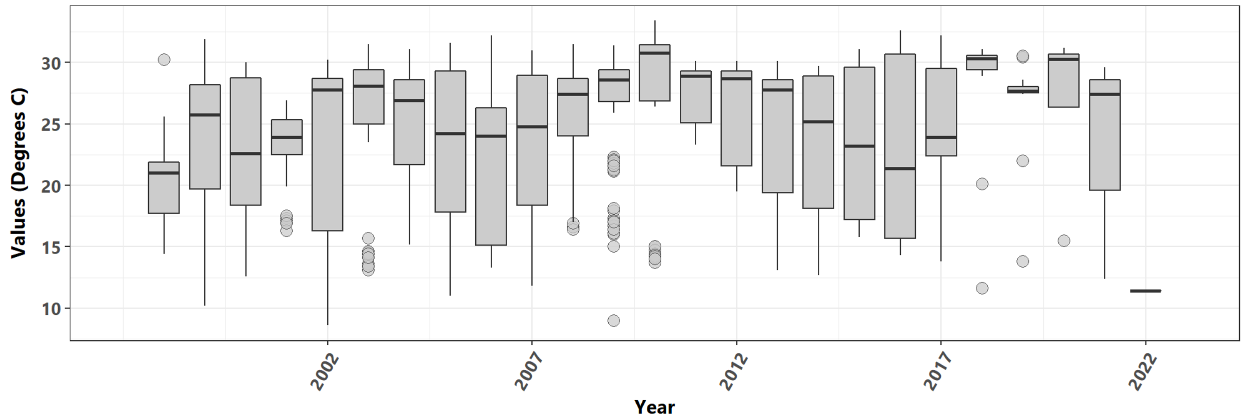
```r
p2 <- ggplot(data=plot_data,
             aes(x=Year, y=ResultValue, group=Year)) +
   geom_boxplot(color="#333333", fill="#cccccc", outlier.shape=21,
                outlier.size=3, outlier.color="#333333",
                outlier.fill="#cccccc", outlier.alpha=0.75) +
   labs(subtitle="Scaled to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

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

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

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

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

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

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

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

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

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

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

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

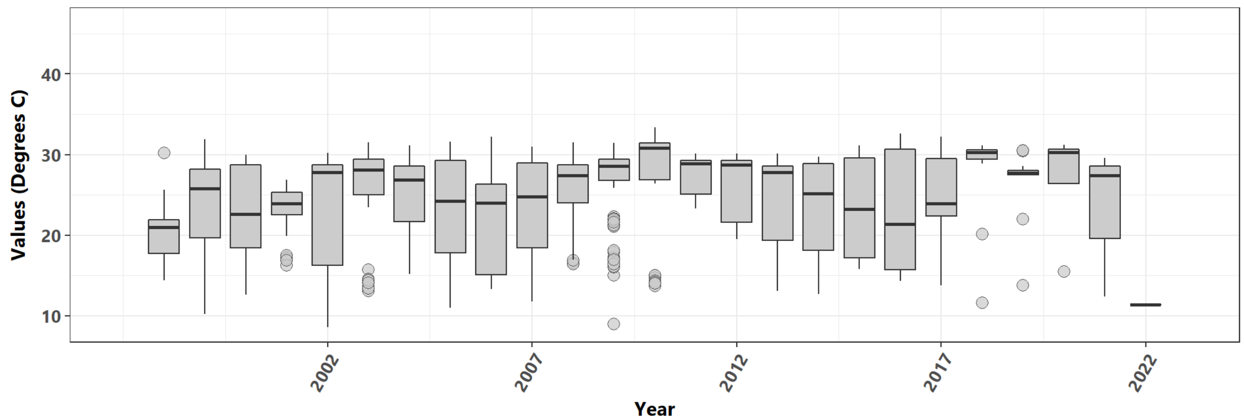
```
      guides(fill=guide_legend(nrow=1))

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

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

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

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

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

Alligator Harbor Aquatic Preserve
By Year

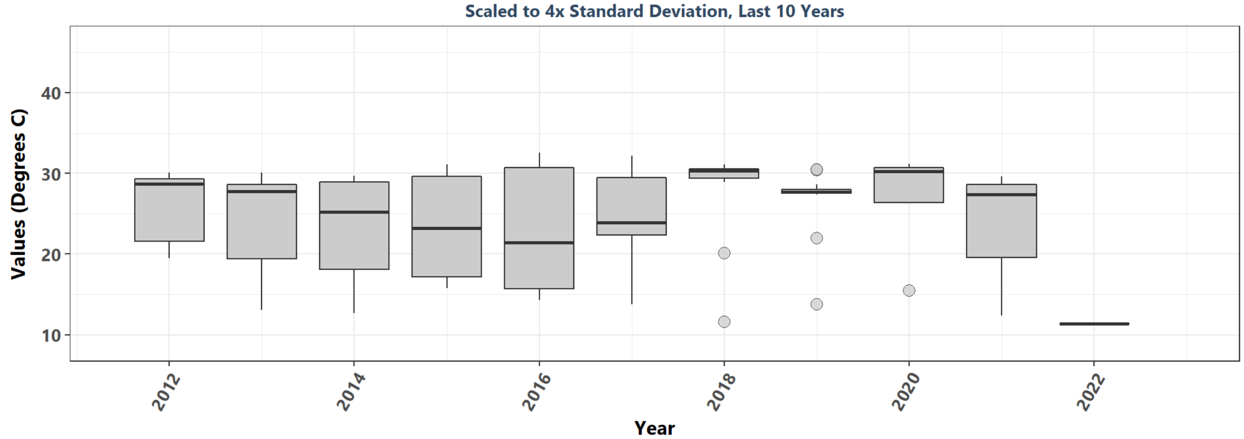
Autoscale



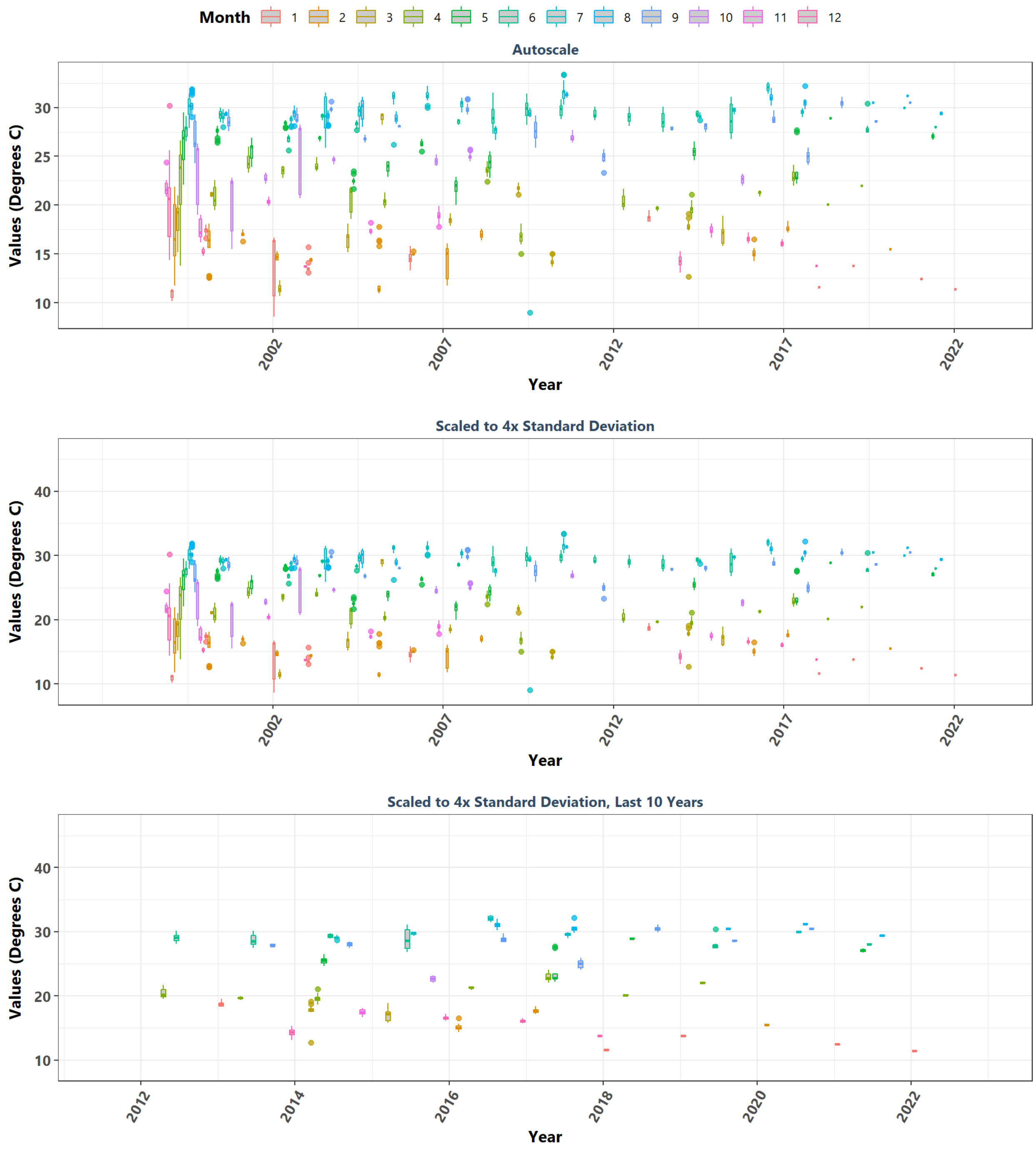
Scaled to 4x Standard Deviation



Scaled to 4x Standard Deviation, Last 10 Years

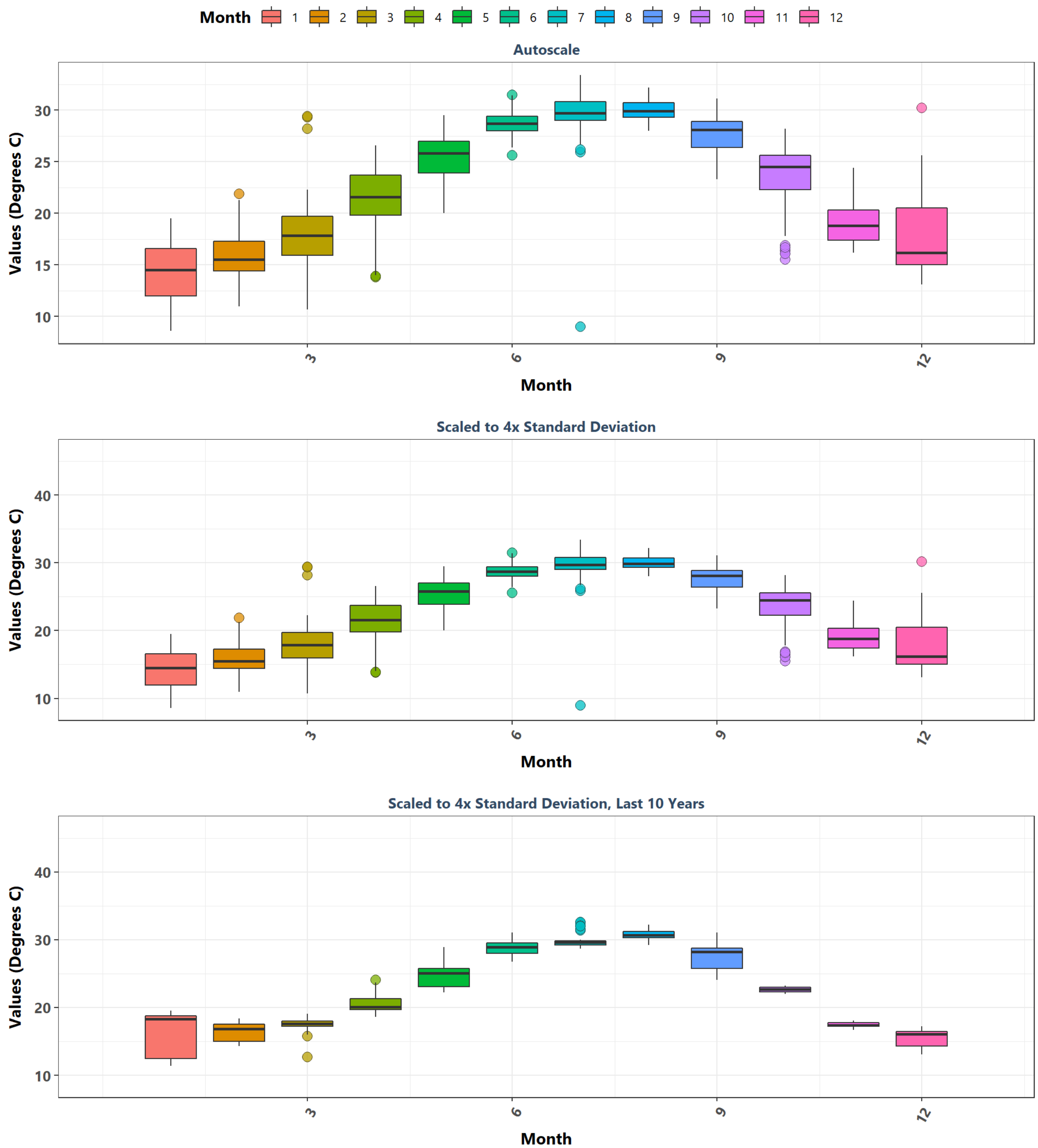


Alligator Harbor Aquatic Preserve
By Year & Month



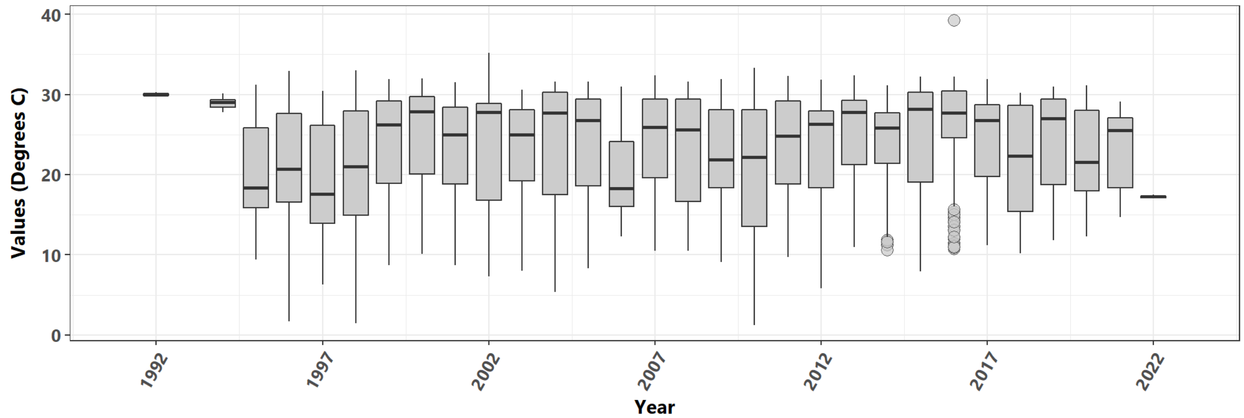
Alligator Harbor Aquatic Preserve

By Month

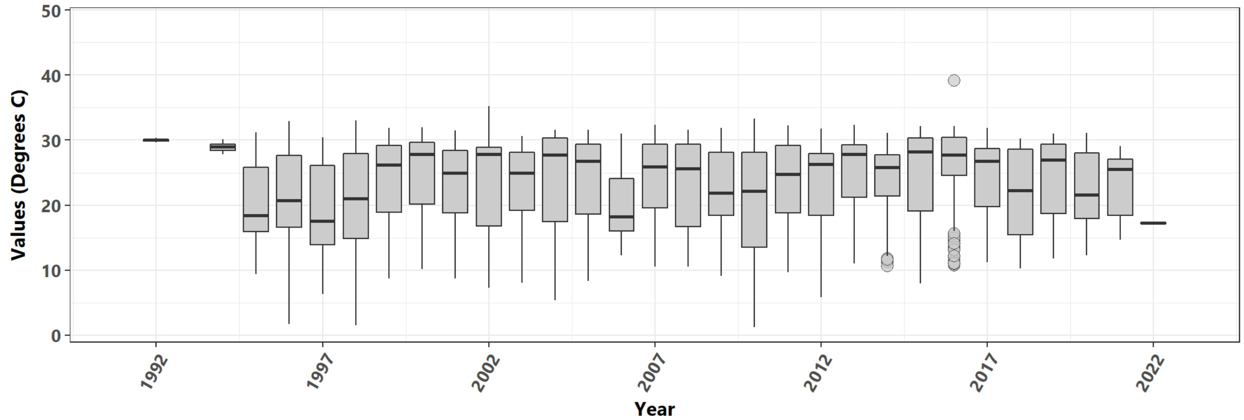


Apalachicola Bay Aquatic Preserve
By Year

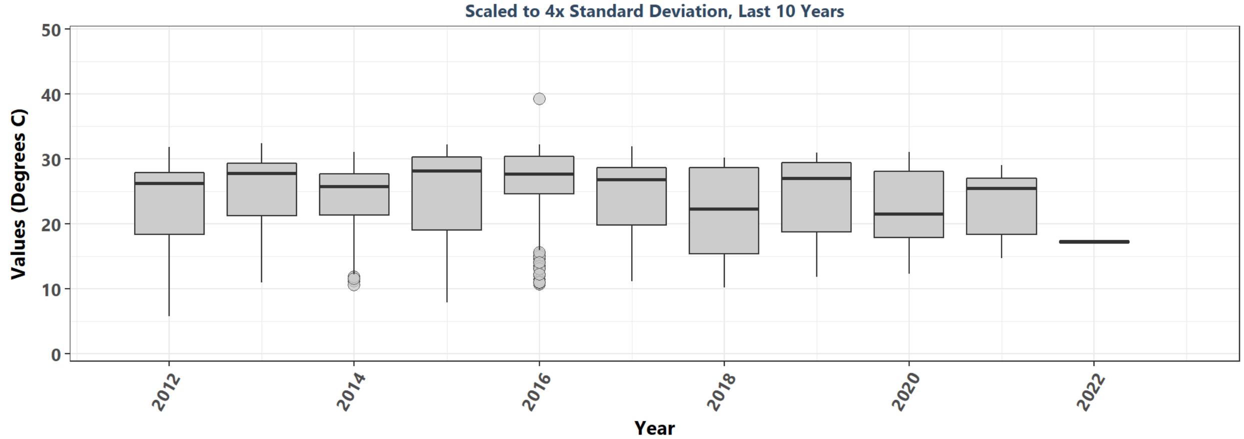
Autoscale



Scaled to 4x Standard Deviation

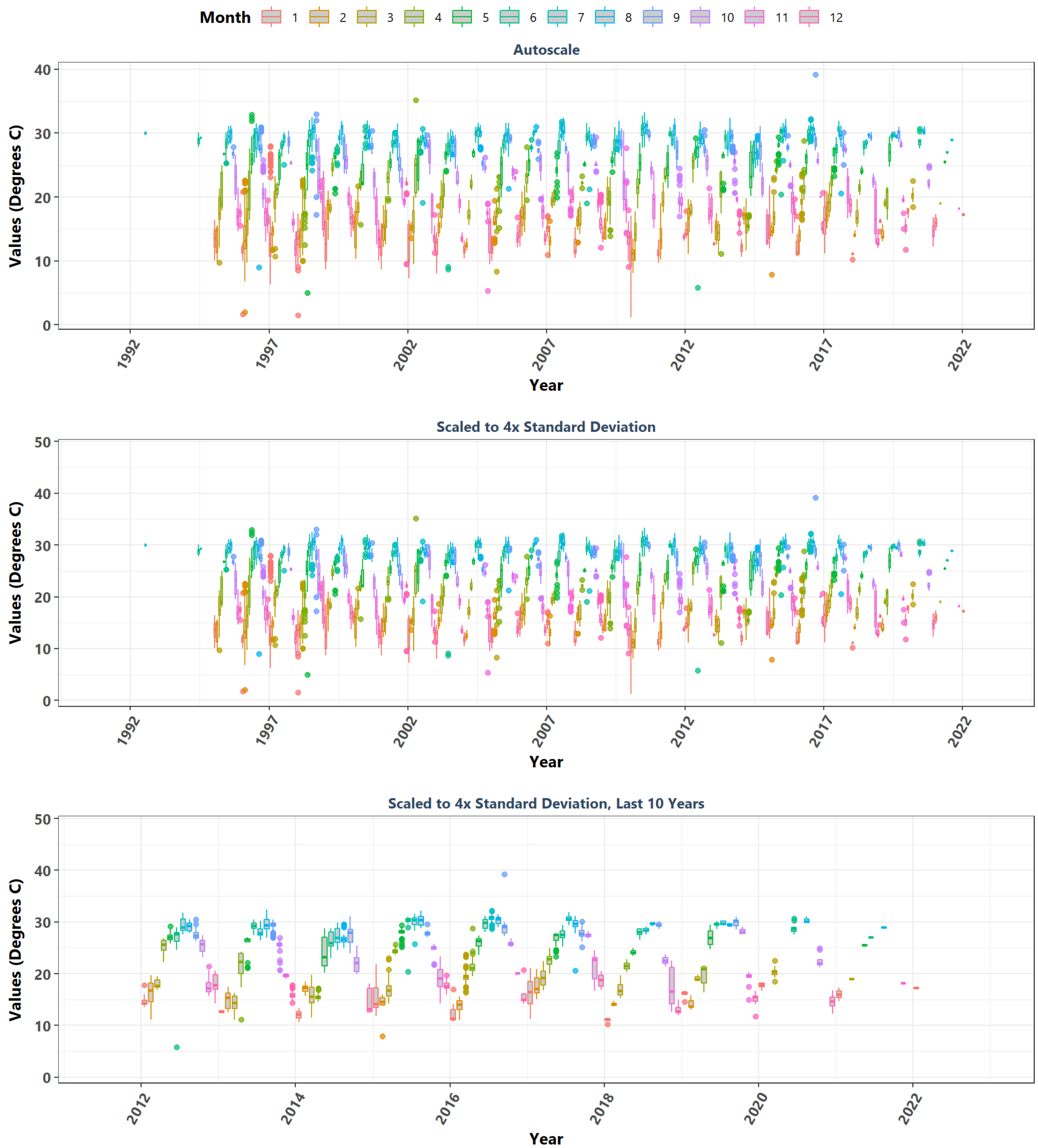


Scaled to 4x Standard Deviation, Last 10 Years

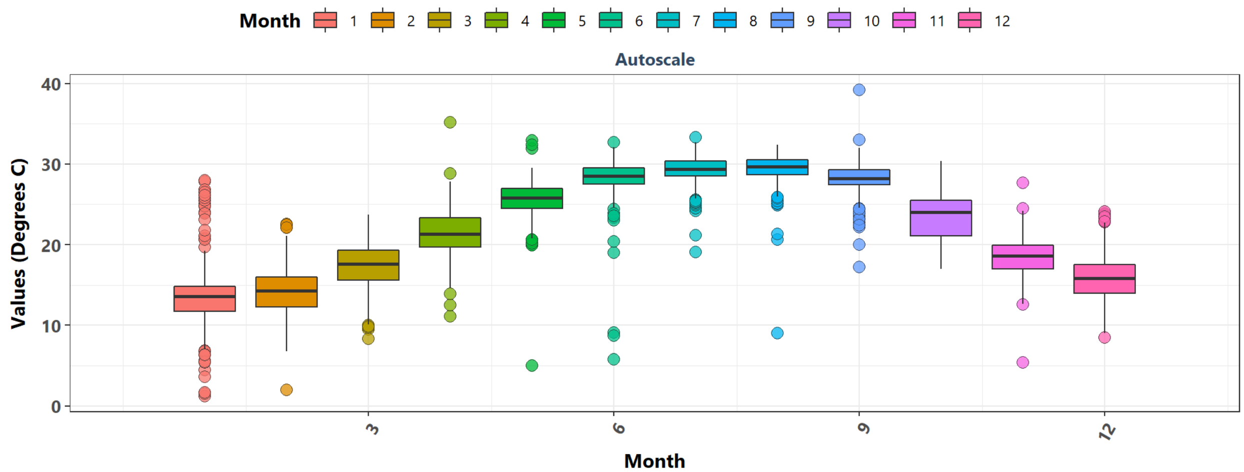


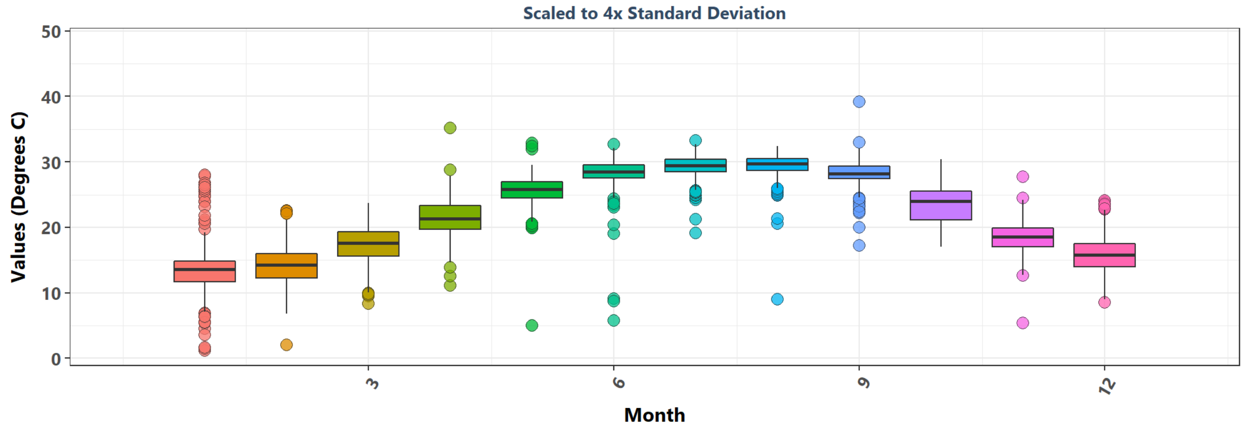
Apalachicola Bay Aquatic Preserve

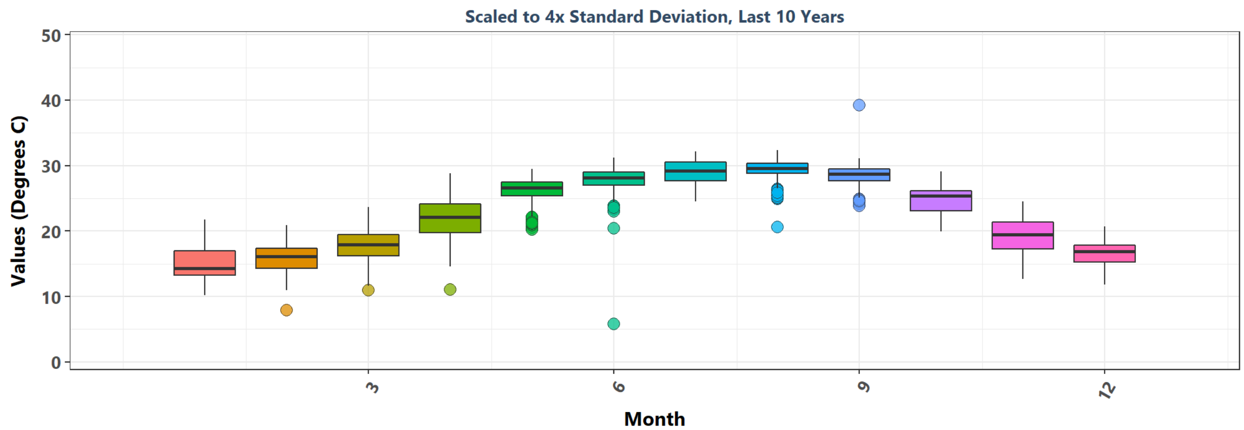
By Year & Month



Apalachicola Bay Aquatic Preserve
By Month

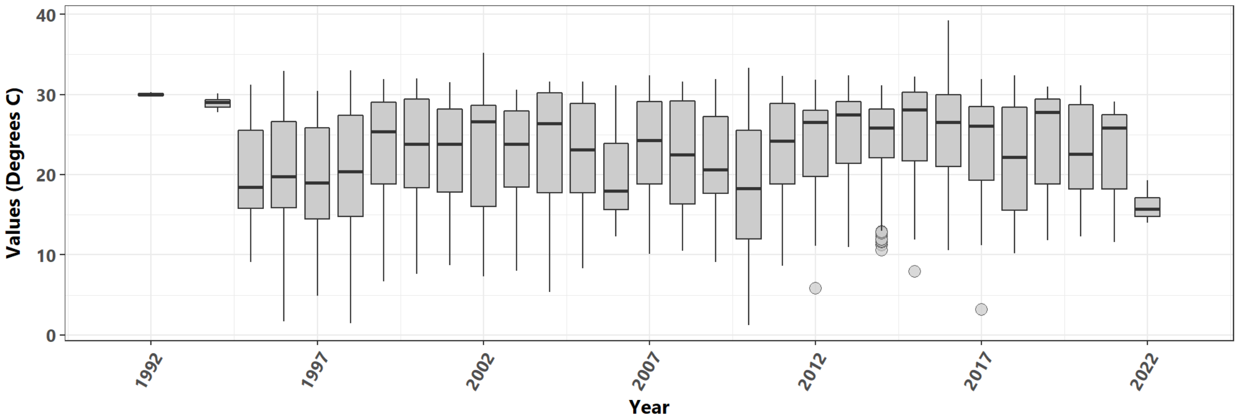




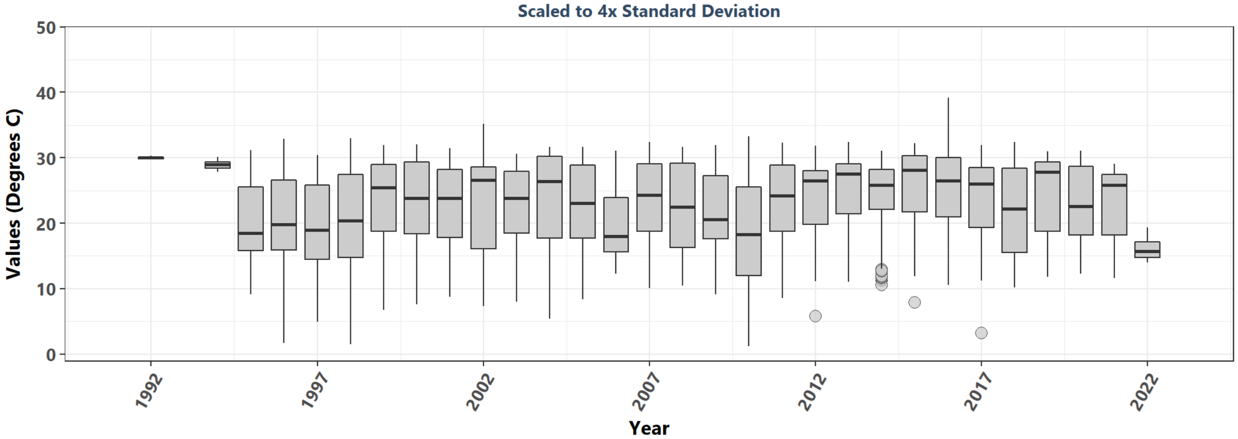


Apalachicola National Estuarine Research Reserve
By Year

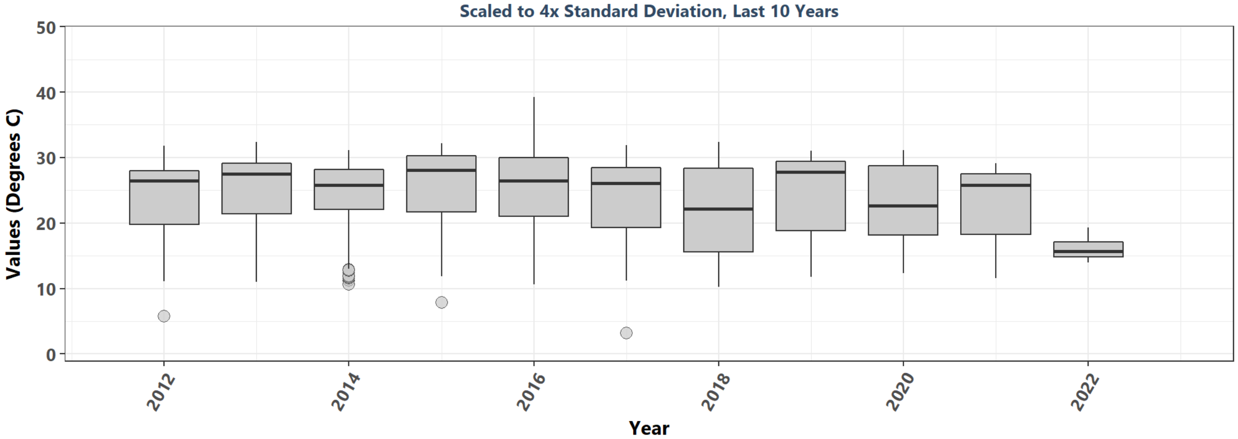
Autoscale



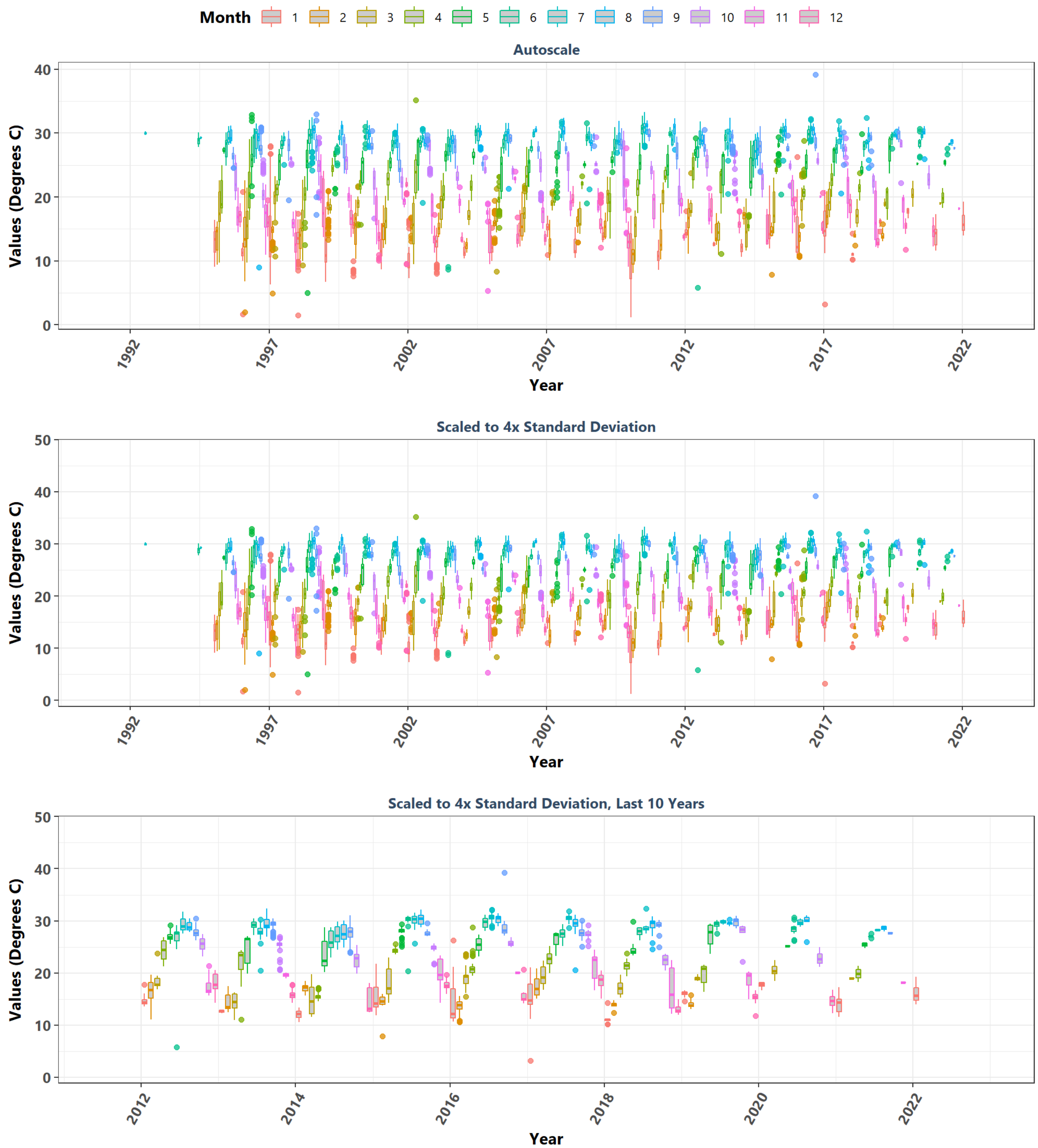
Scaled to 4x Standard Deviation

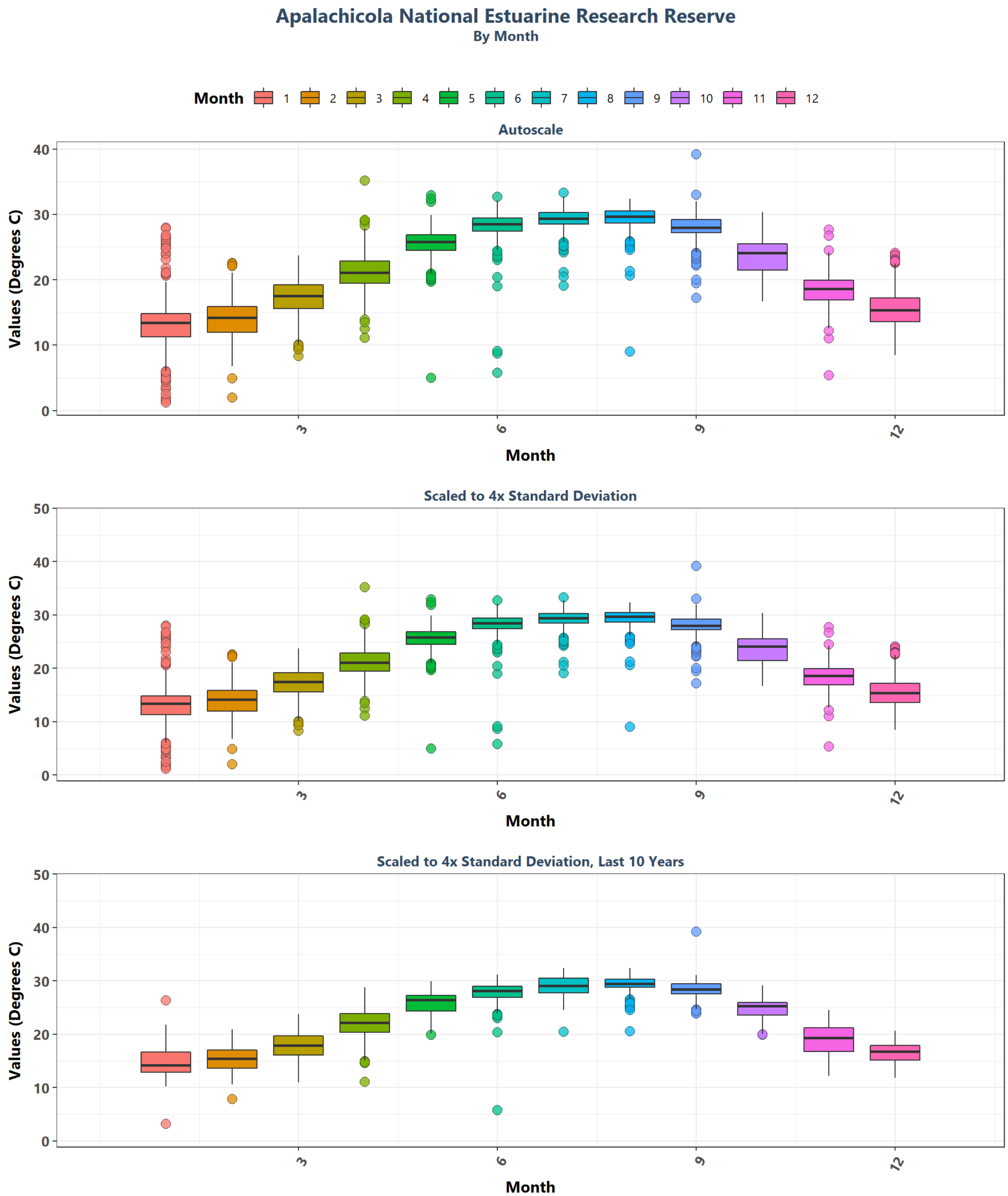


Scaled to 4x Standard Deviation, Last 10 Years

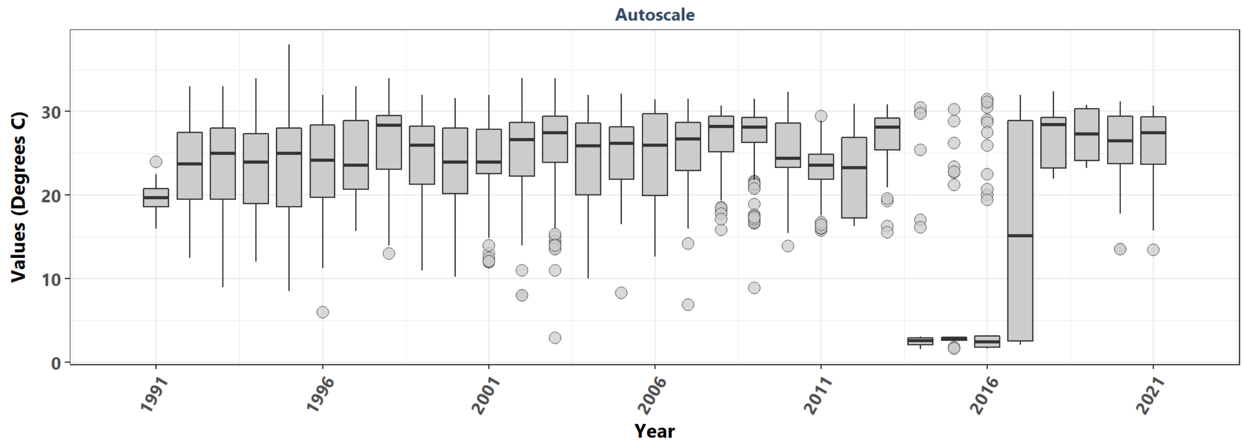


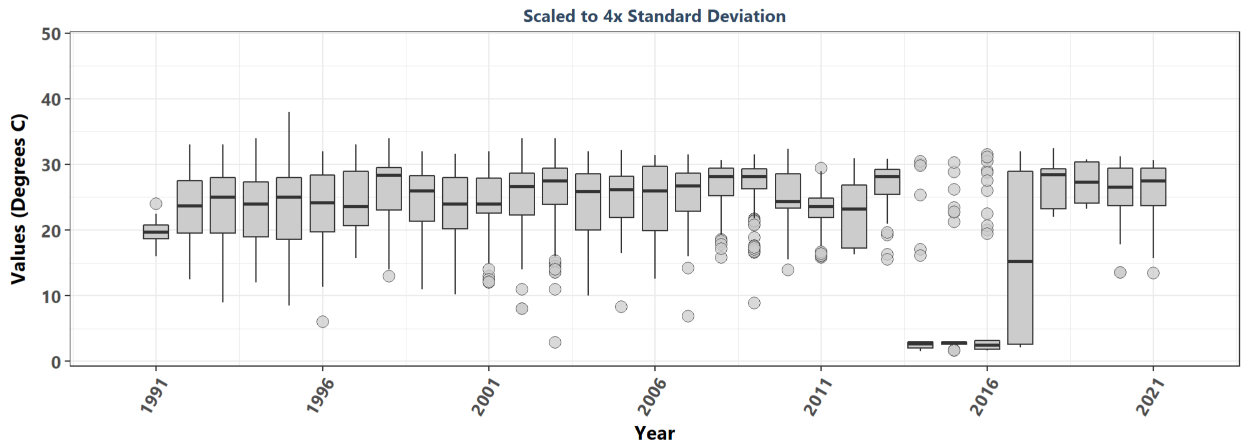
Apalachicola National Estuarine Research Reserve
By Year & Month

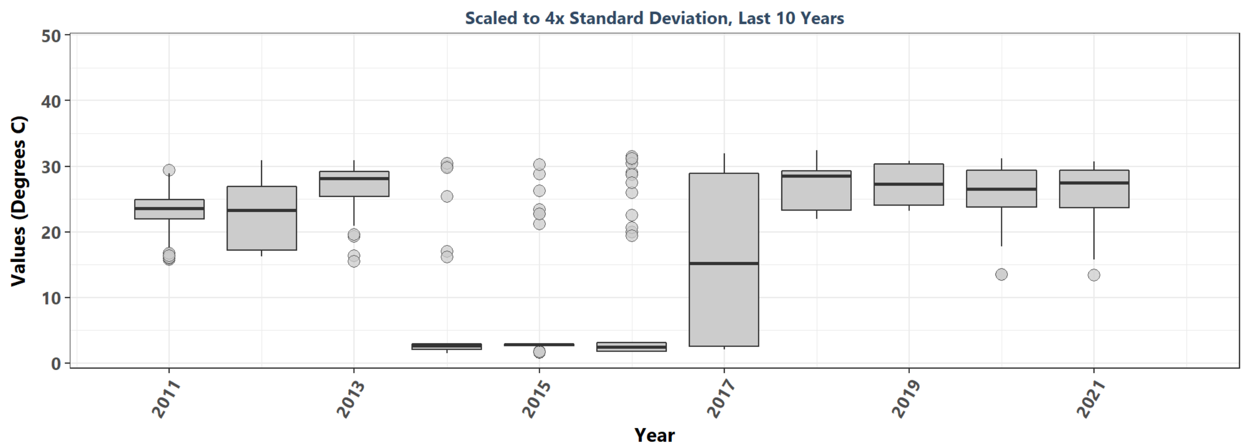


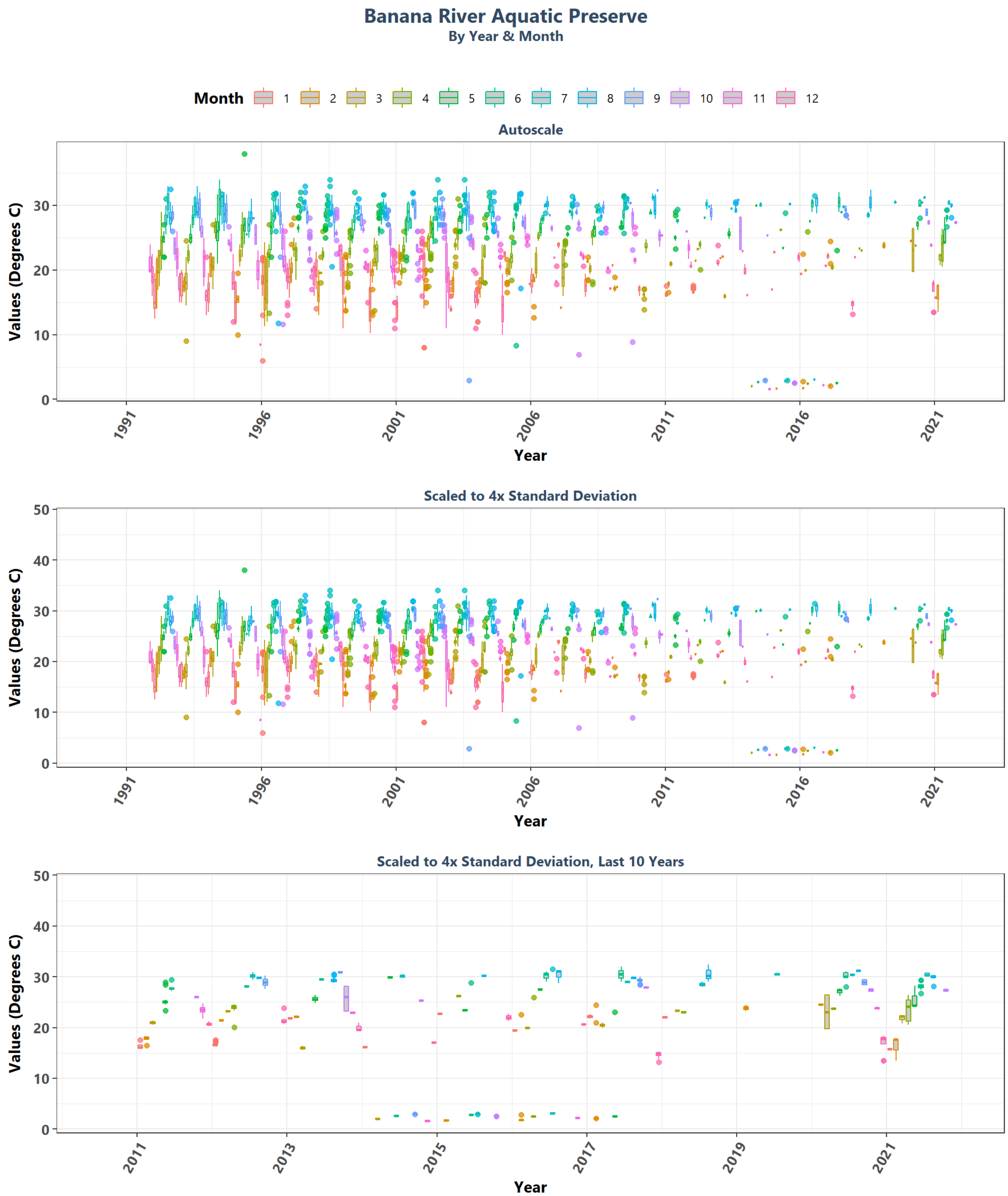

Apalachicola National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month

Banana River Aquatic Preserve
By Year



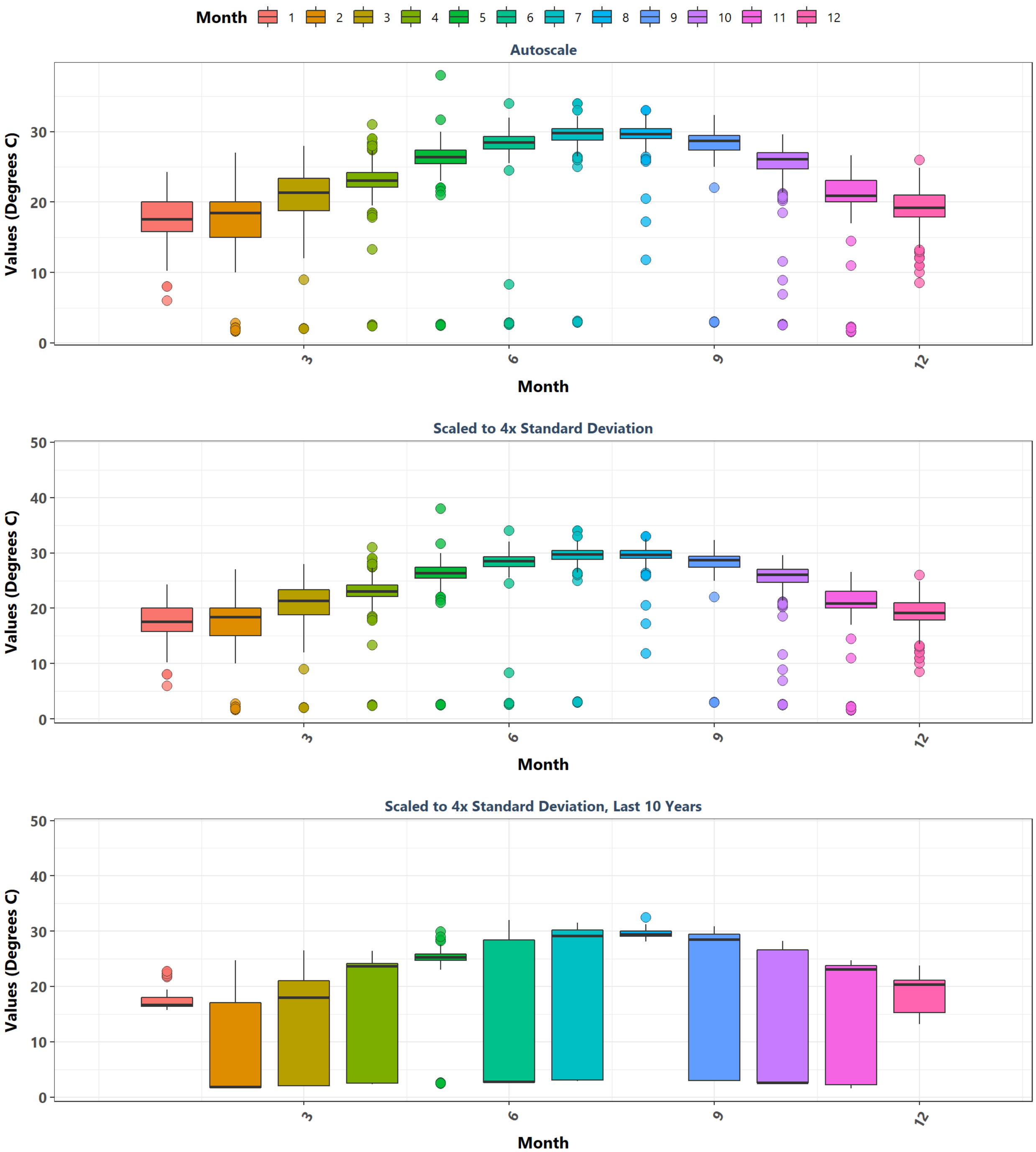






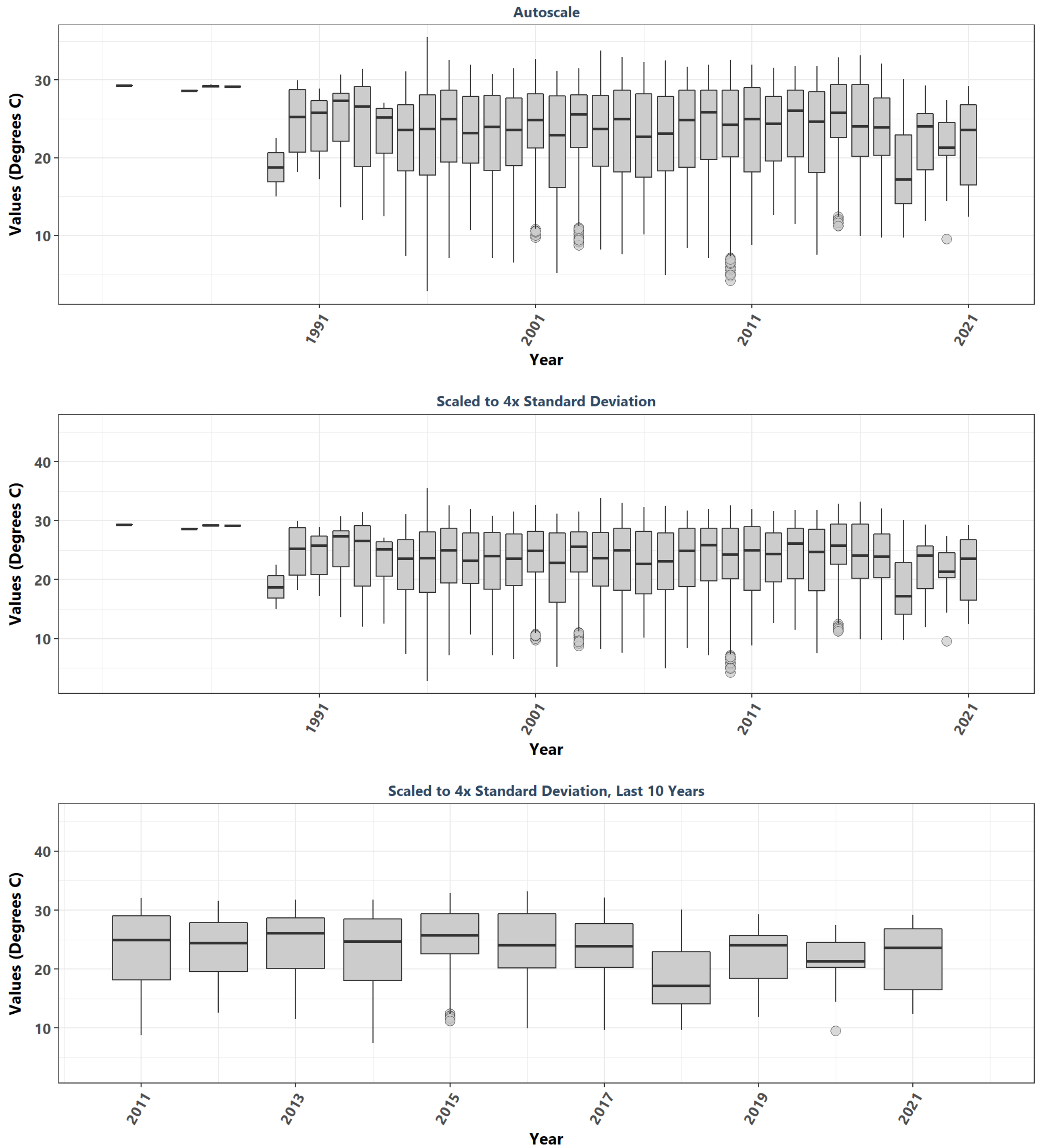
Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
1991 1996 2001 2006 2011 2016 2021
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
1991 1996 2001 2006 2011 2016 2021
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year
2011 2013 2015 2017 2019 2021

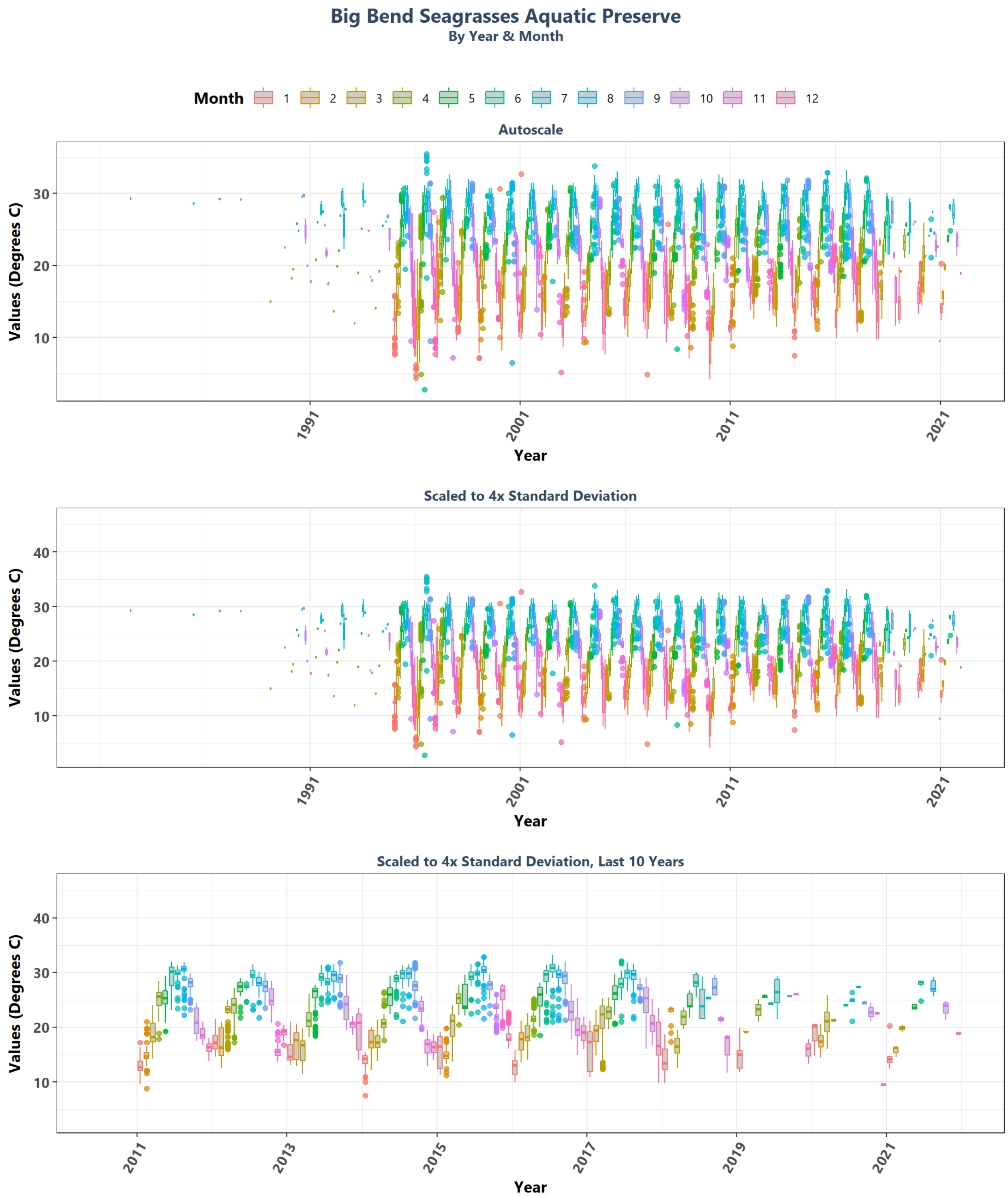
Banana River Aquatic Preserve
By Month



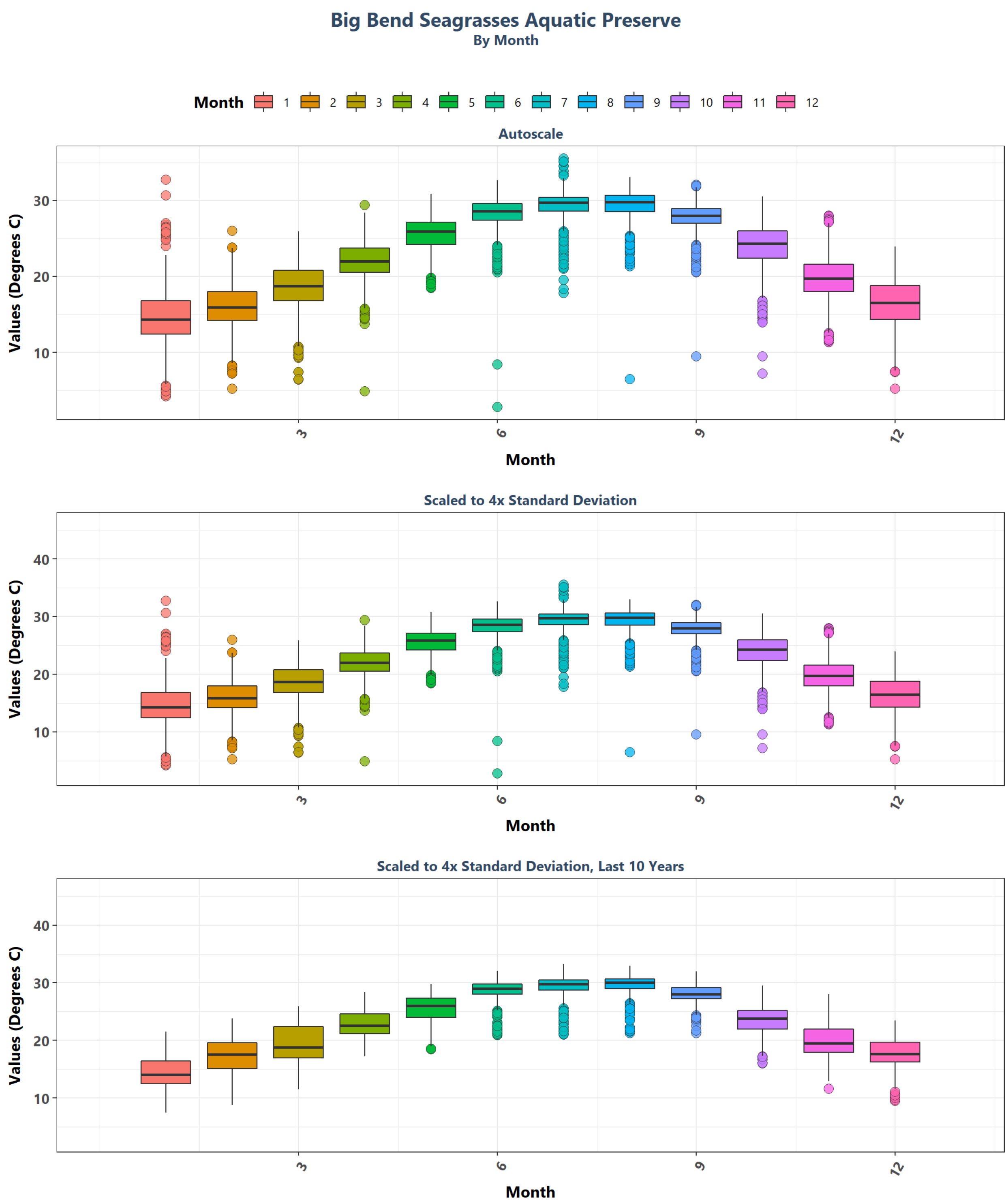
Big Bend Seagrasses Aquatic Preserve

By Year

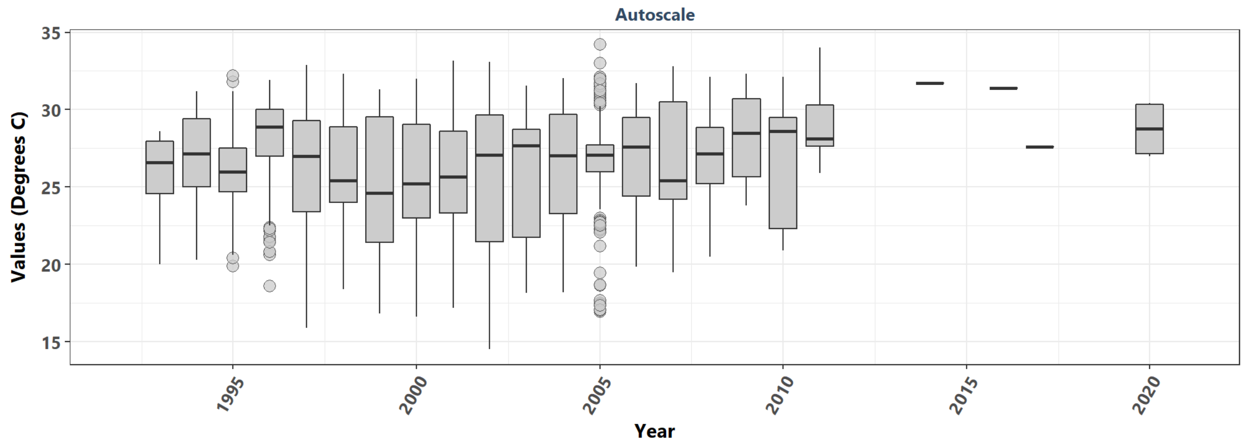


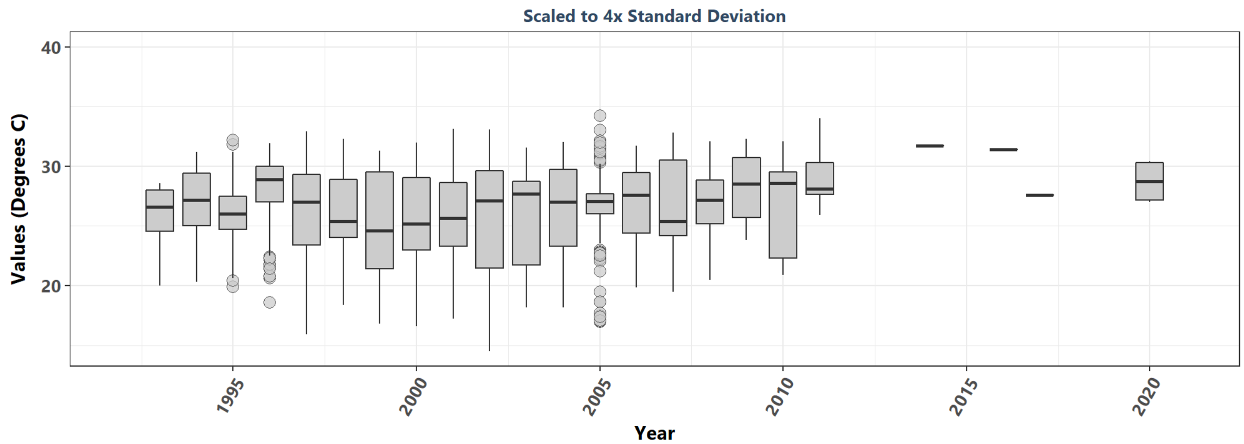


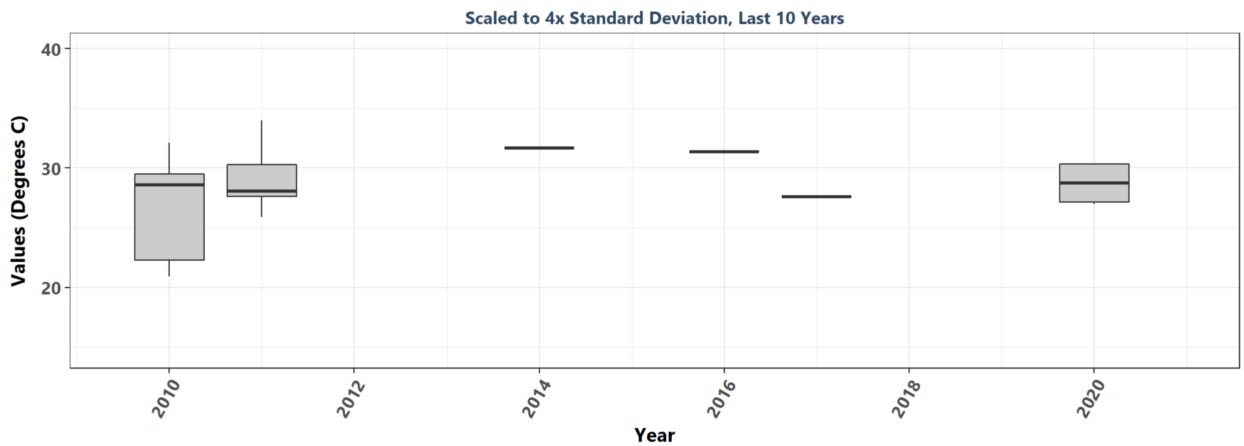
Big Bend Seagrasses Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2011
2013
2015
2017
2019
2021
Year

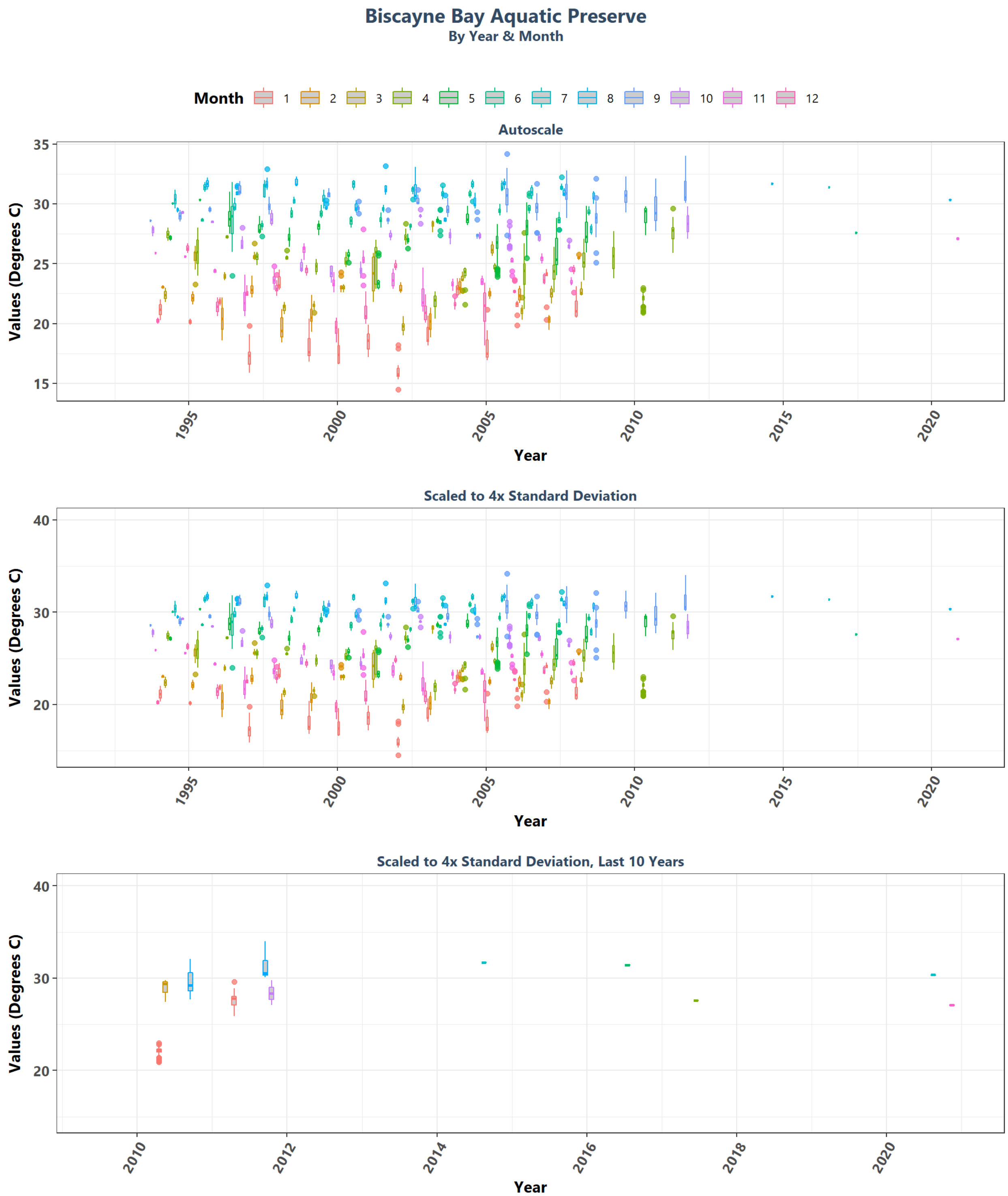

Big Bend Seagrasses Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

Biscayne Bay Aquatic Preserve
By Year





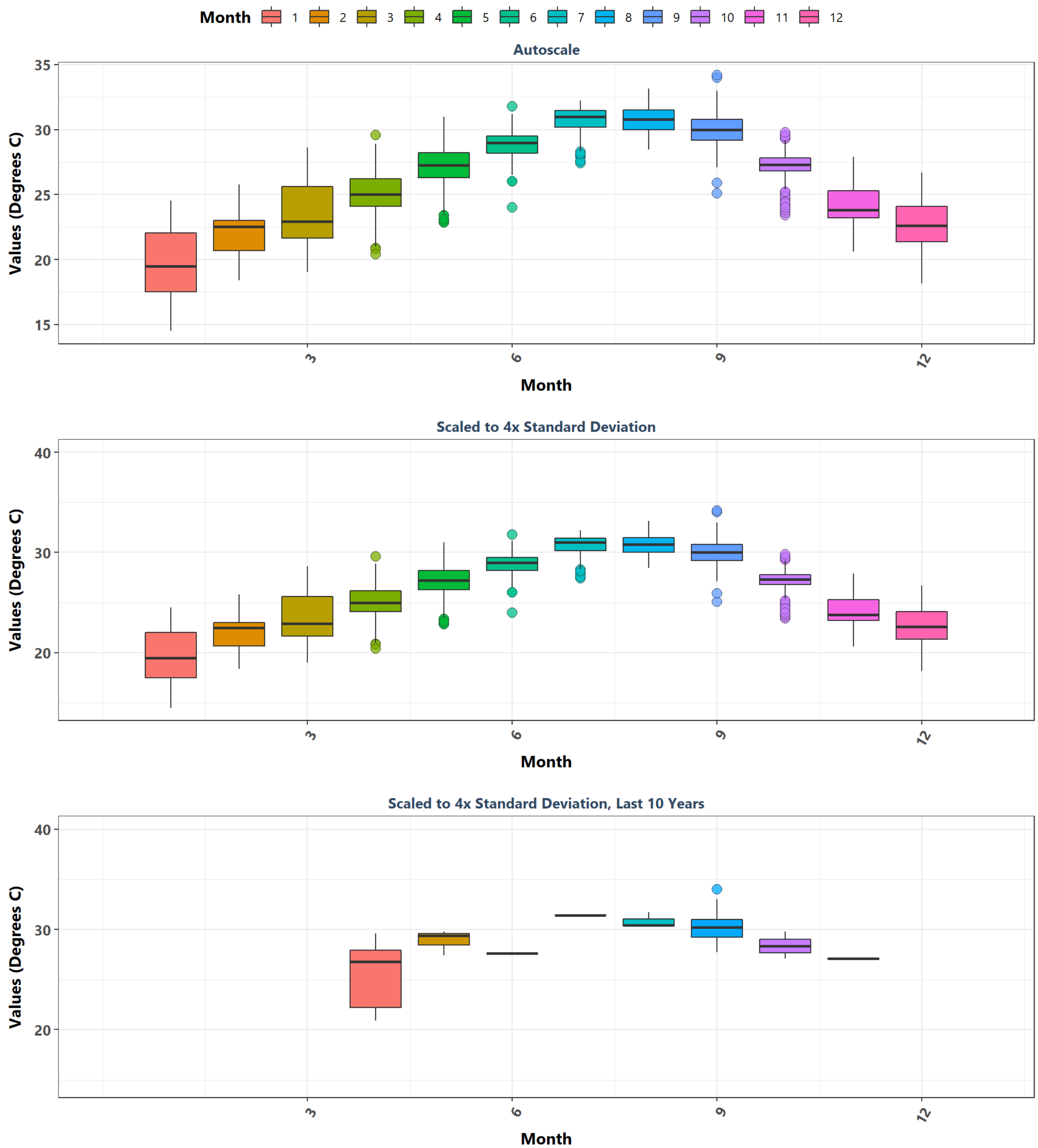




Biscayne Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
2010
2012
2014
2016
2018
2020
Year

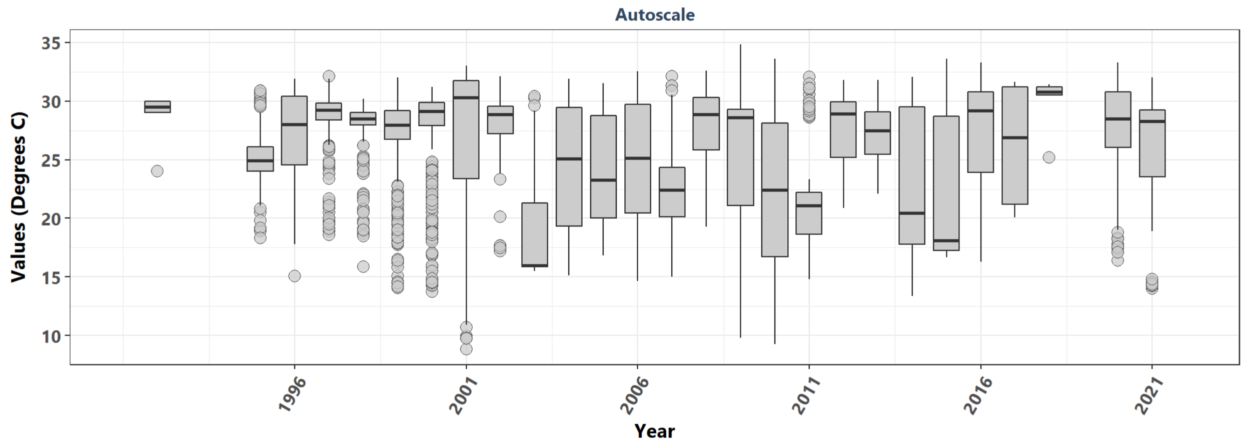
Biscayne Bay Aquatic Preserve

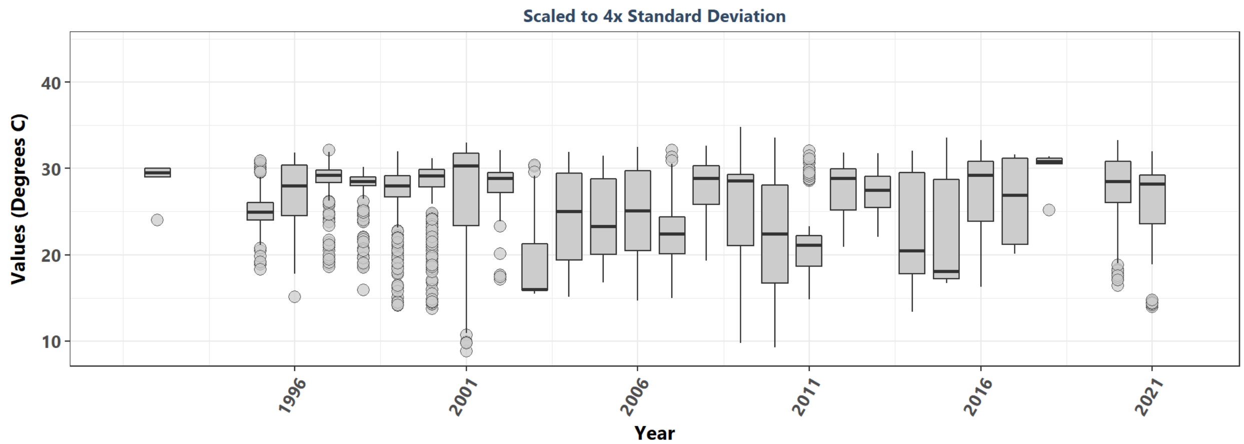
By Month

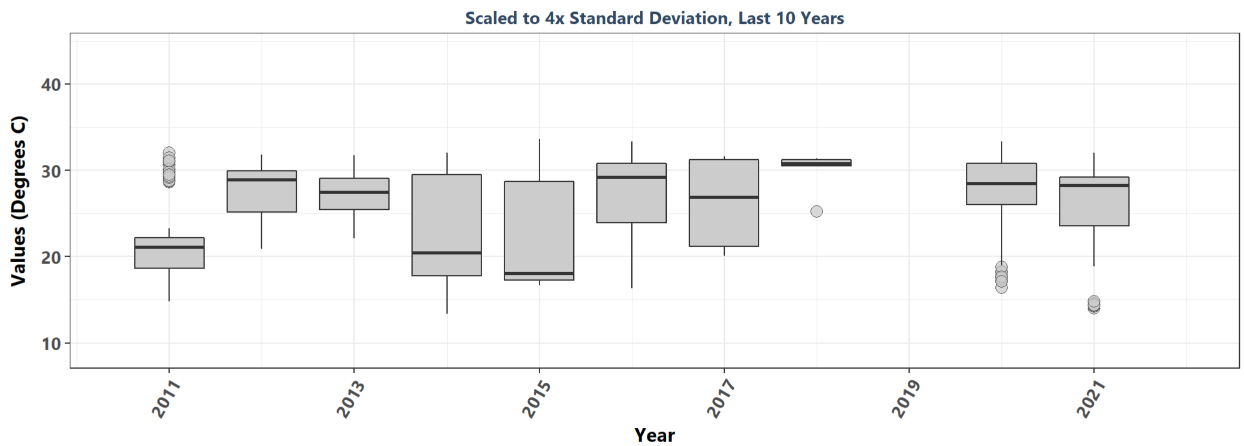


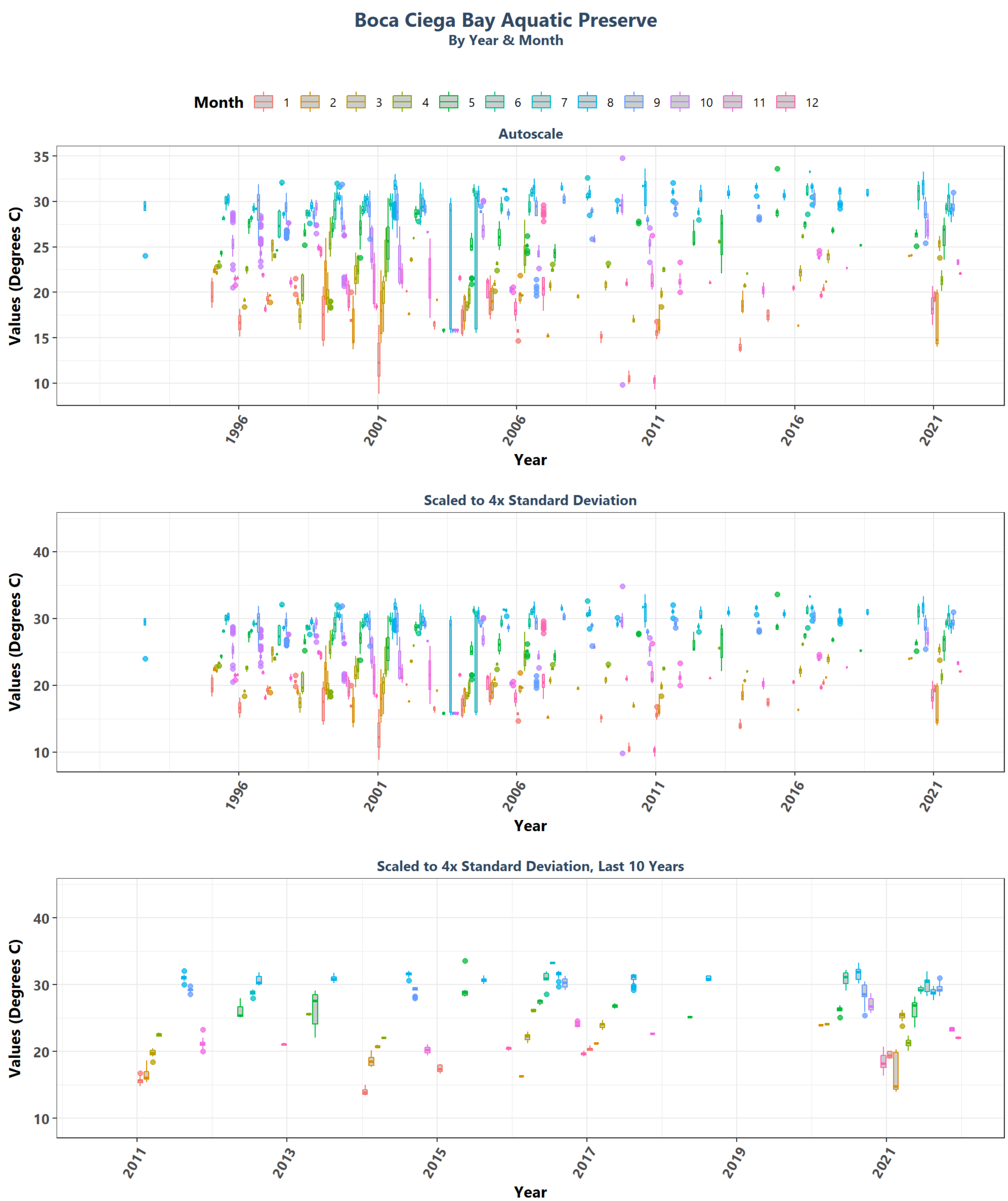
Boca Ciega Bay Aquatic Preserve

By Year

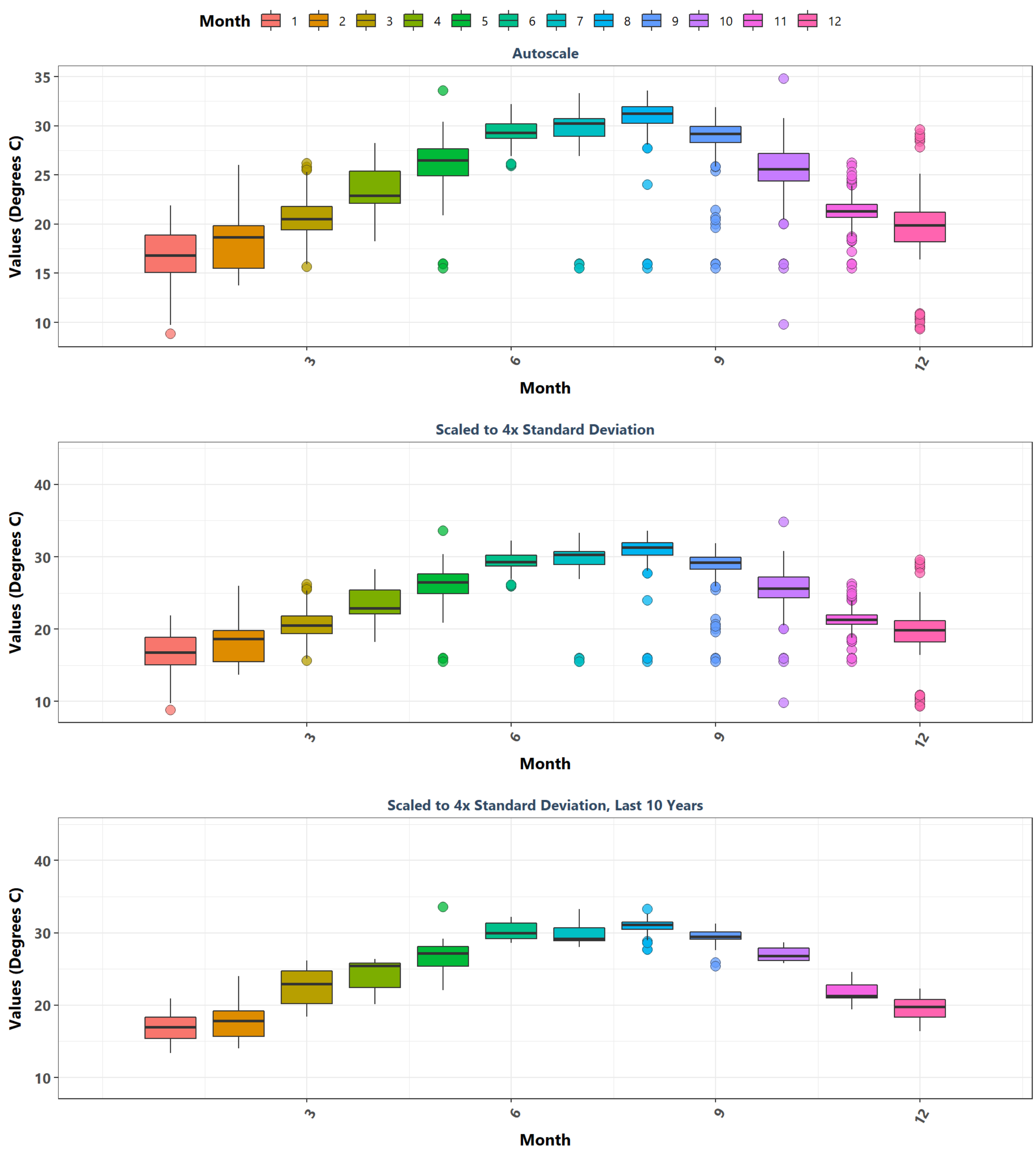




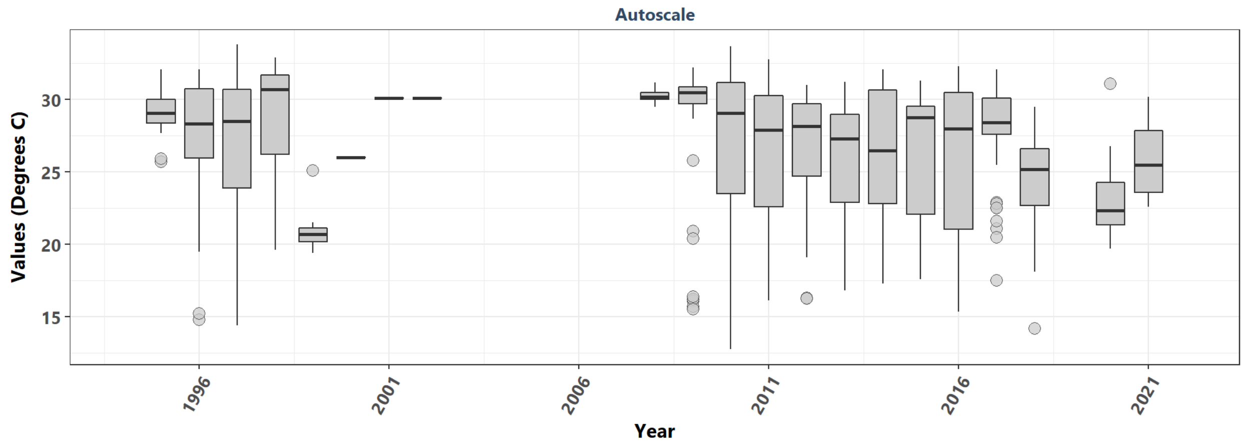


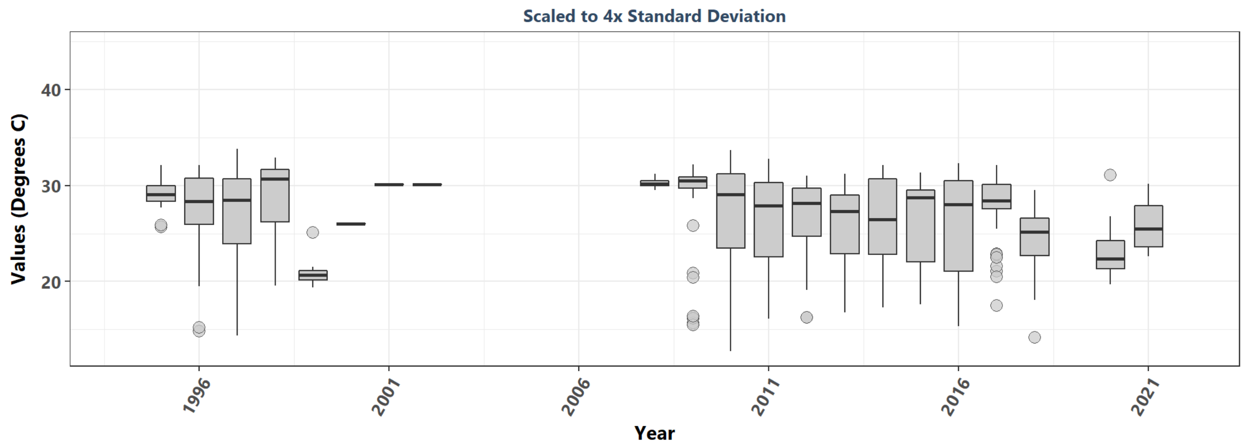

Boca Ciega Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2011
2013
2015
2017
2019
2021
Year

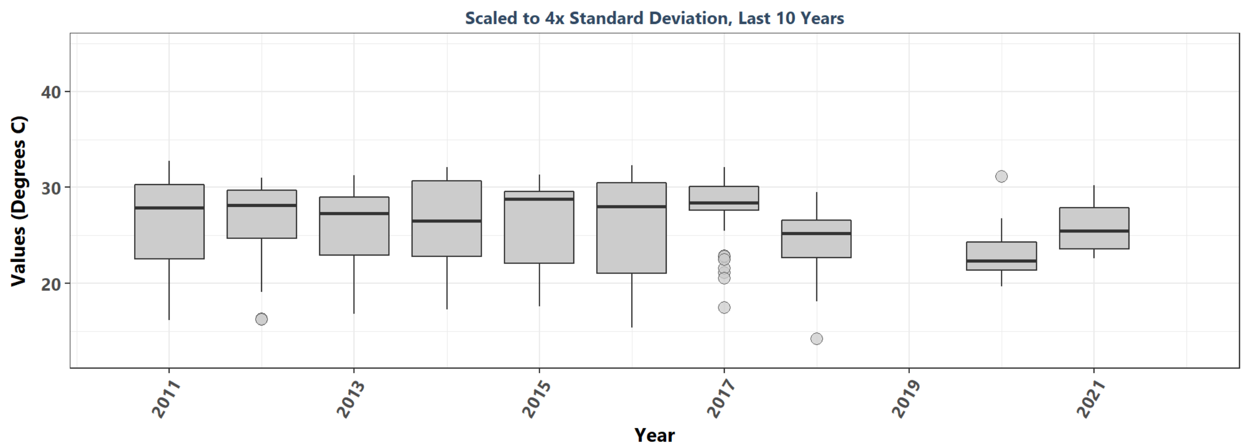
Boca Ciega Bay Aquatic Preserve
By Month

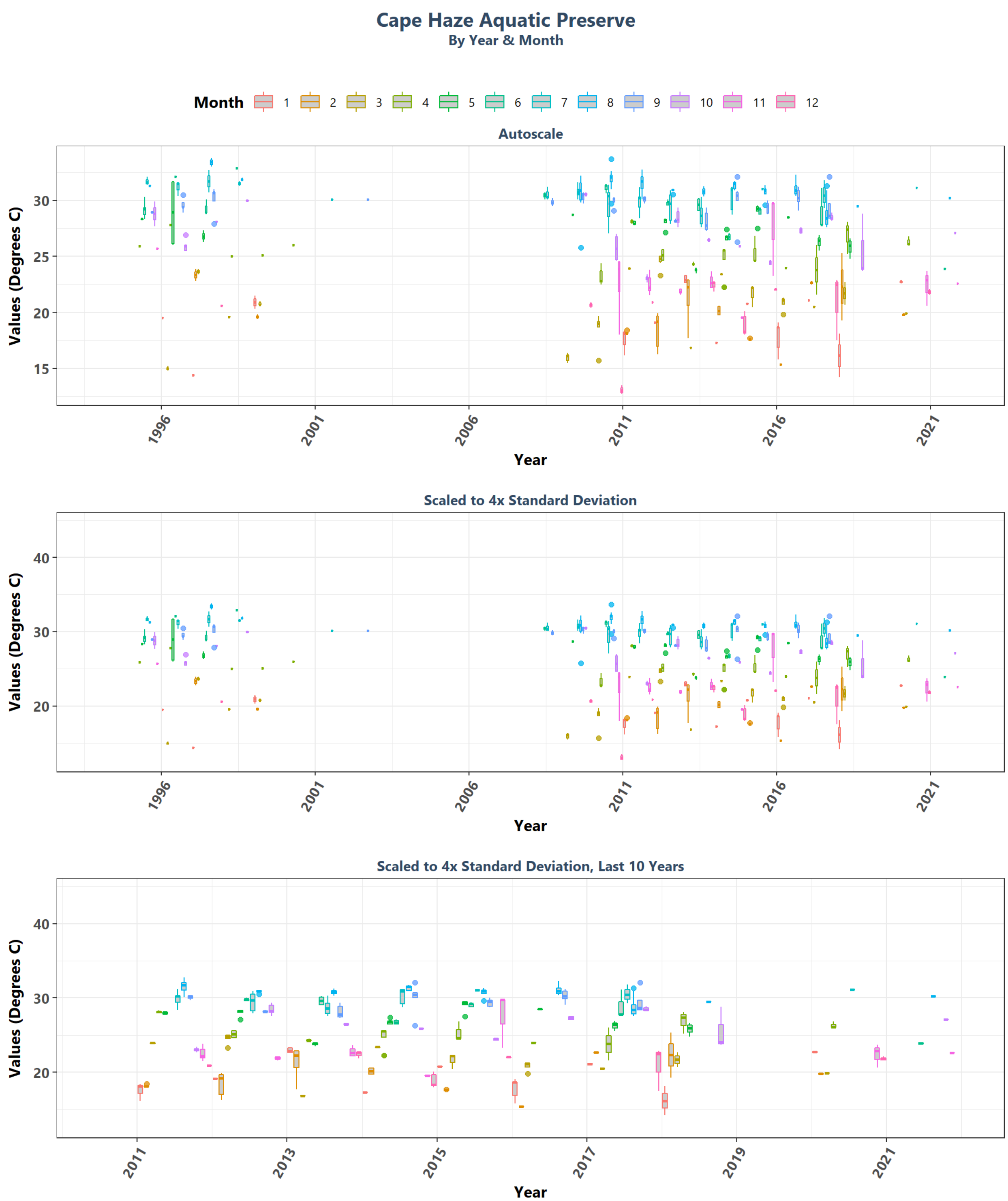


Cape Haze Aquatic Preserve
By Year

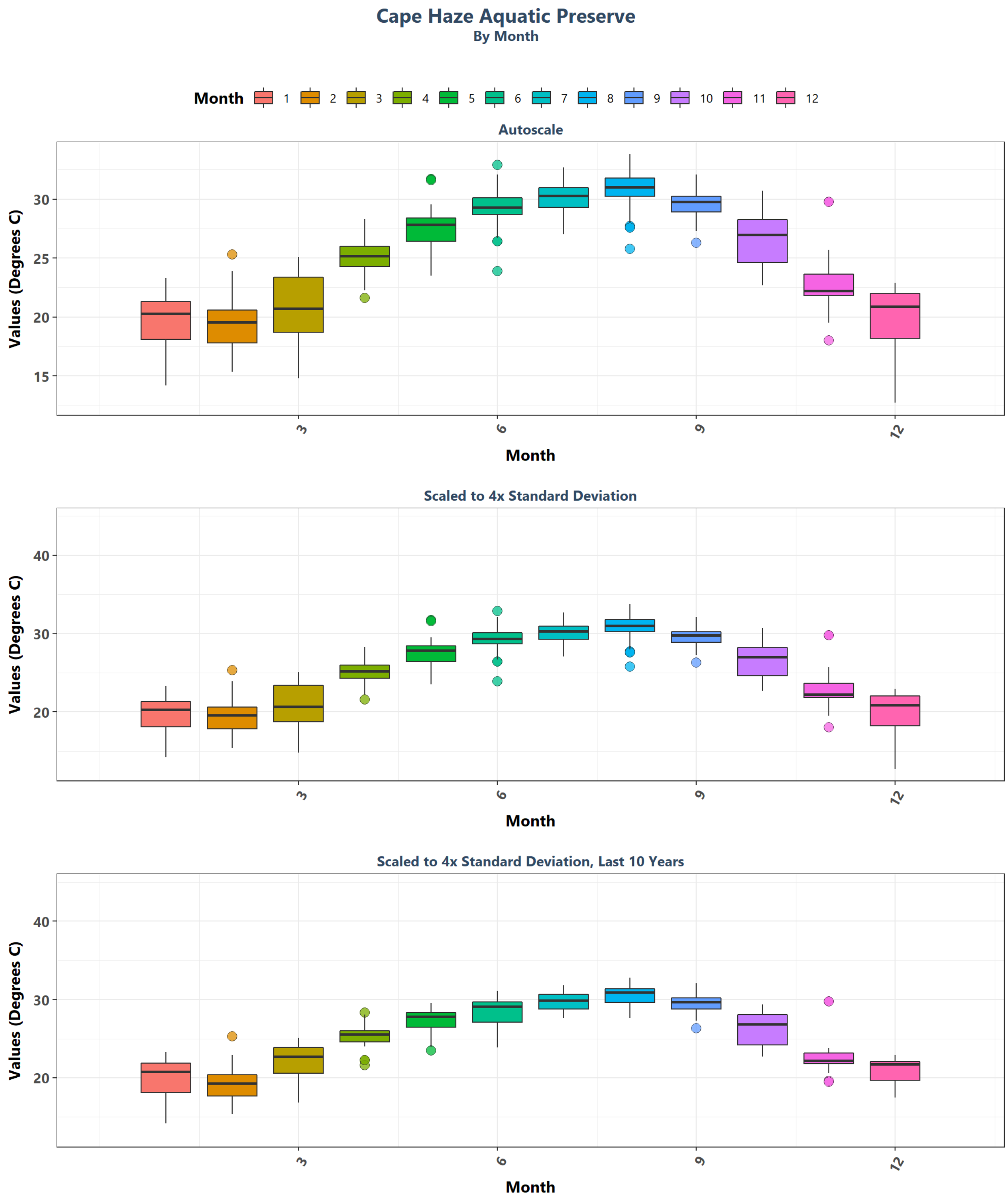


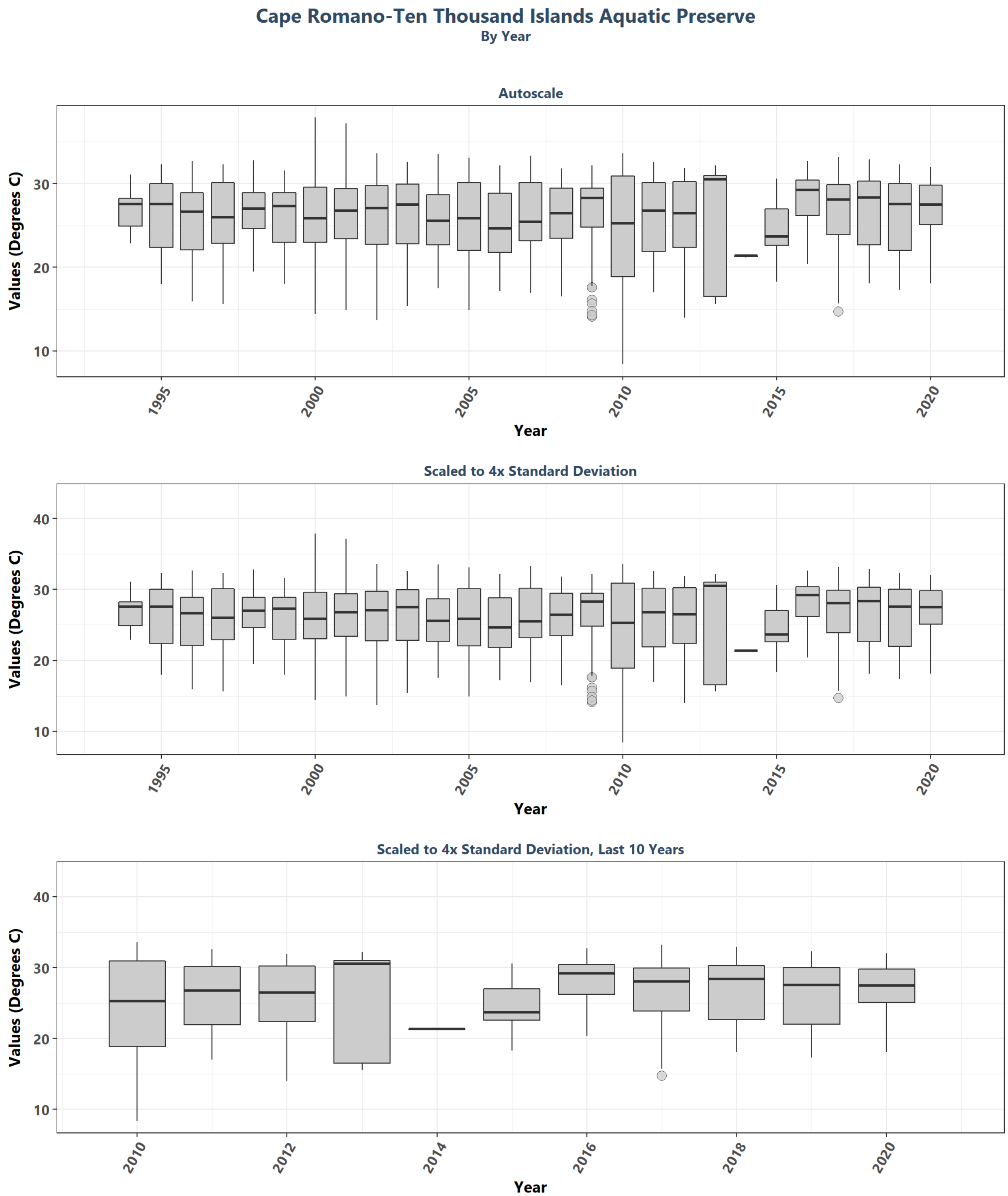


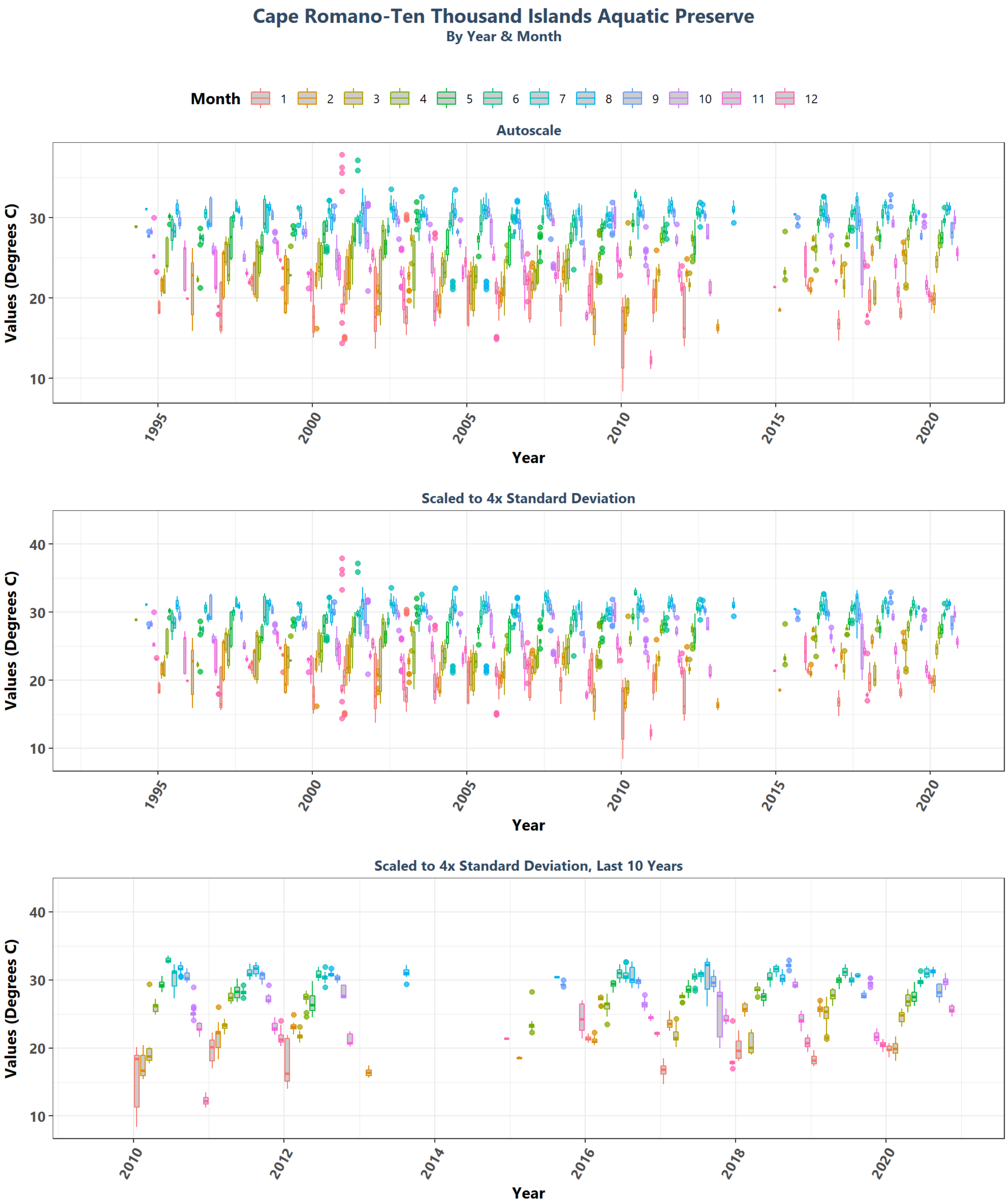




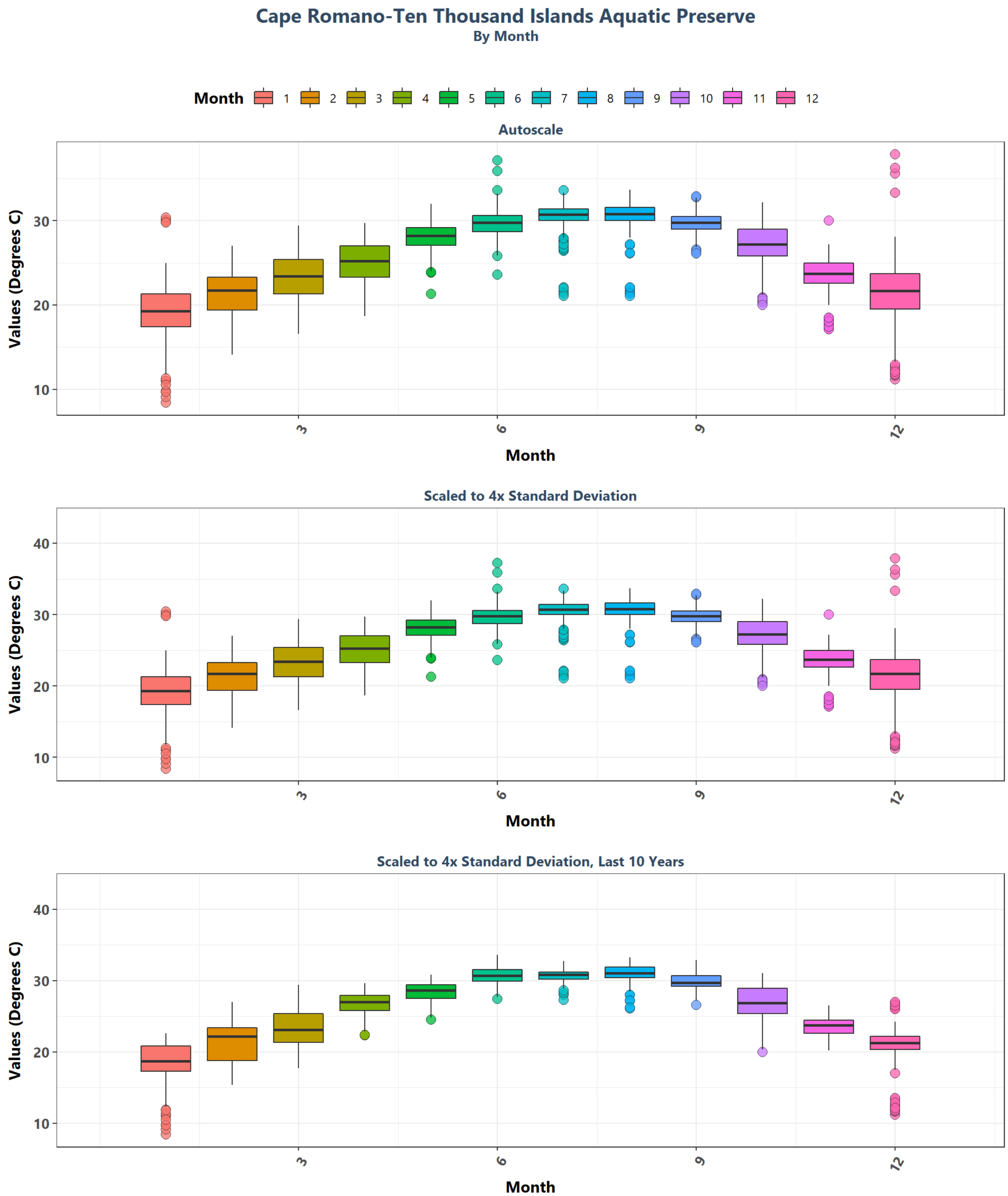
Cape Haze Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


Cape Haze Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
3
6
9
12
Month


Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
10
20
30
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2010
2012
2014
2016
2018
2020
Year

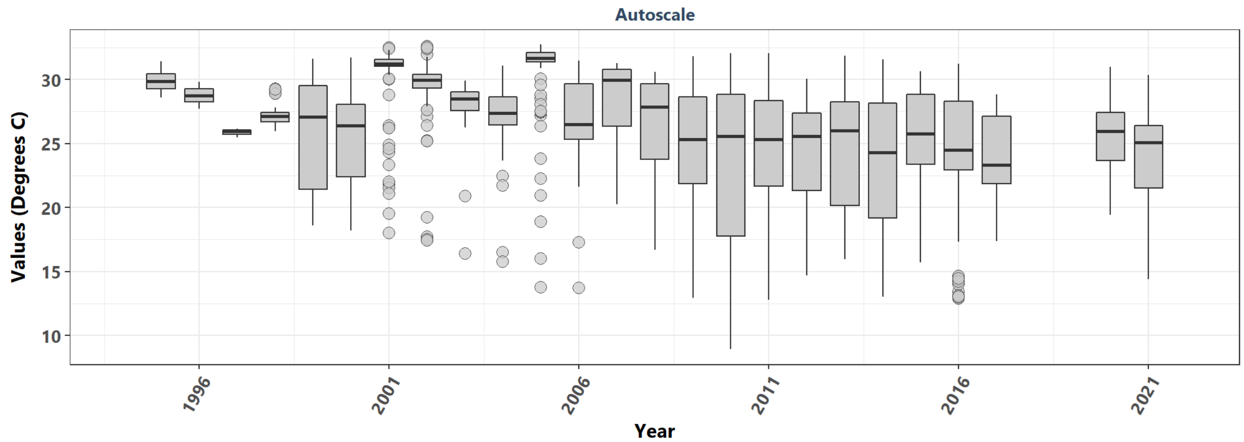


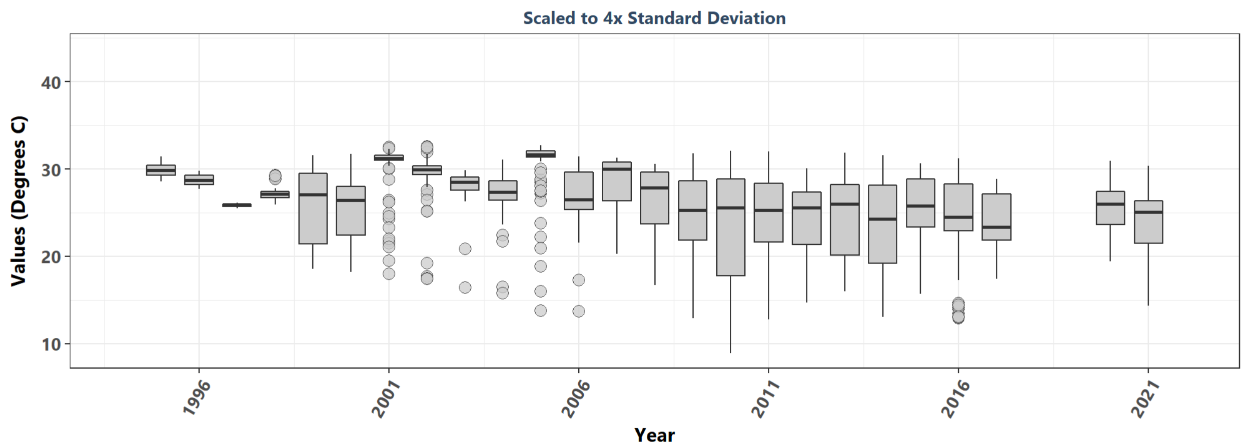
Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2010
2012
2014
2016
2018
2020
Year

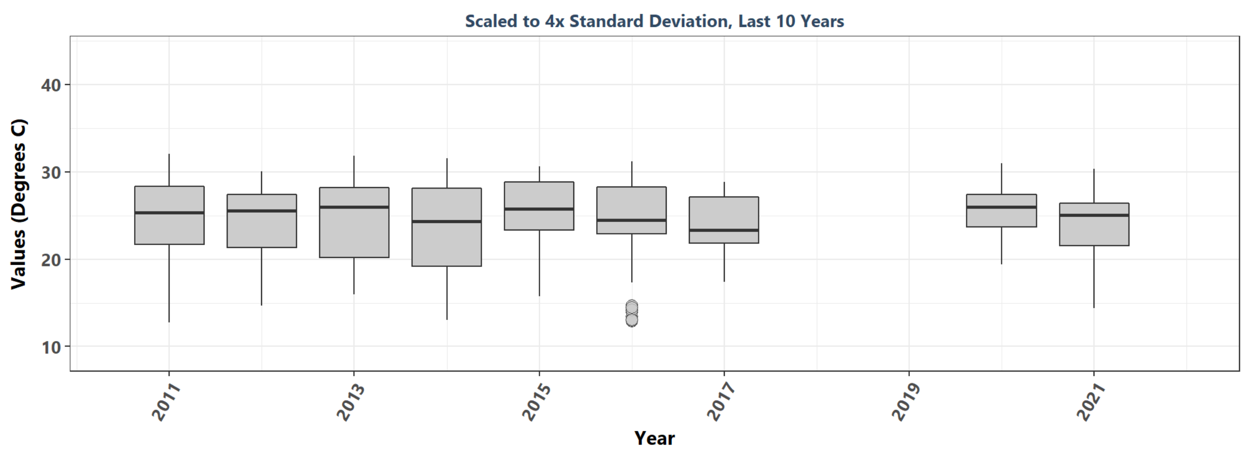


Cape Romano-Ten Thousand Islands Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40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

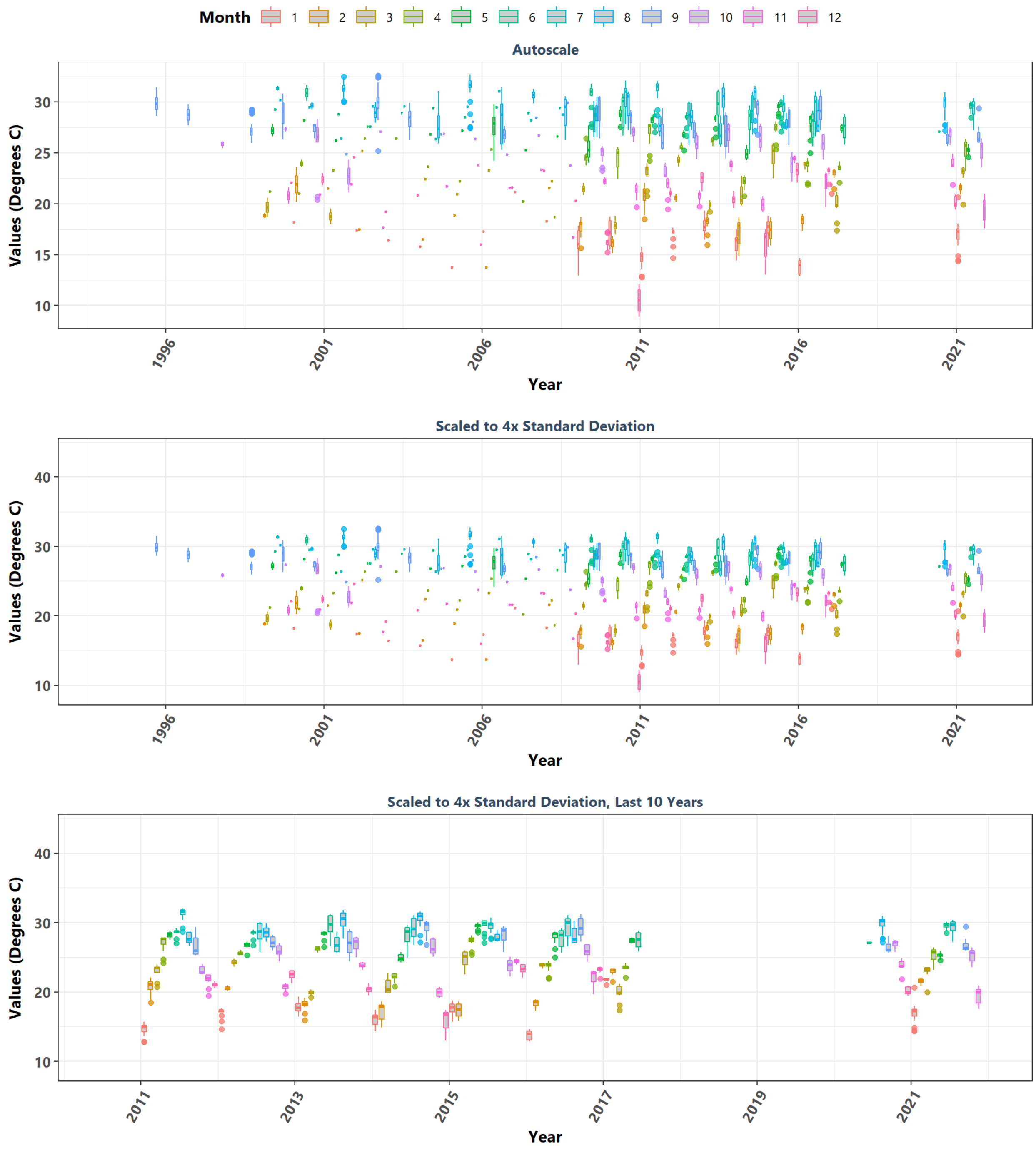
Cockroach Bay Aquatic Preserve
By Year

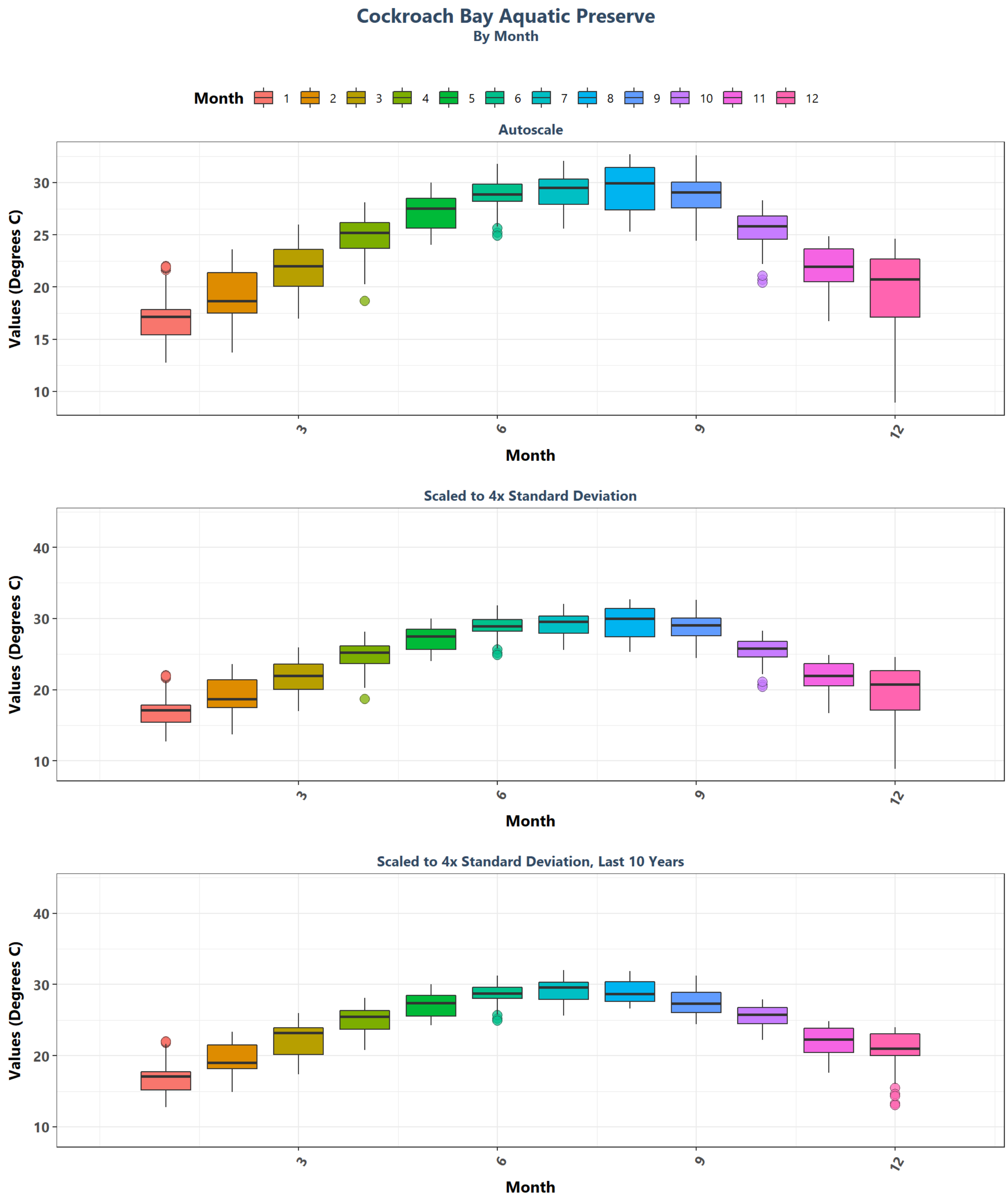




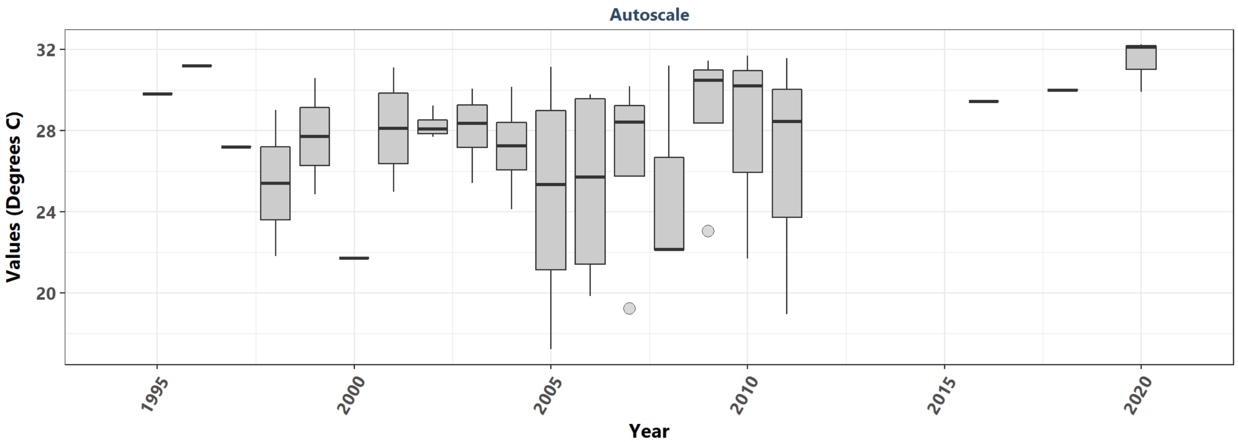


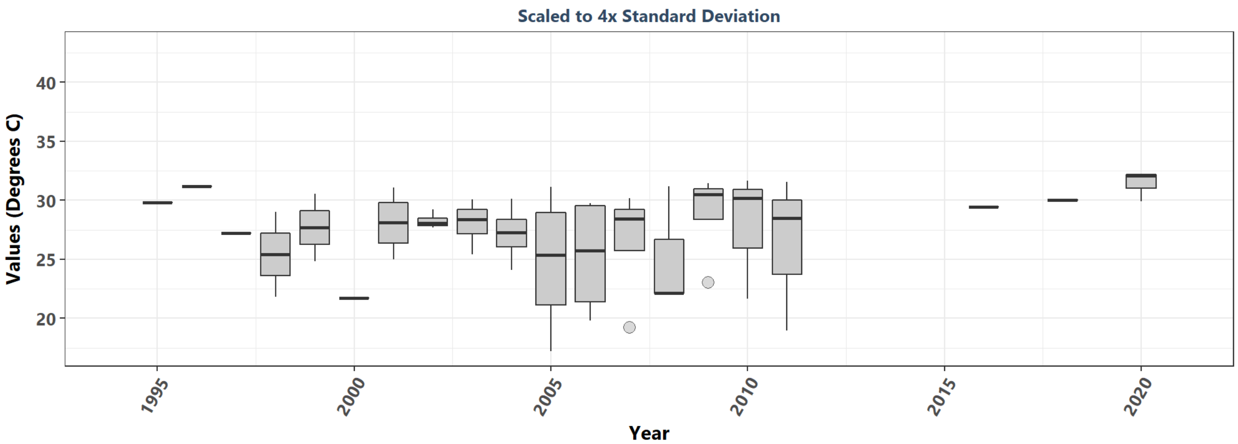
Cockroach Bay Aquatic Preserve
By Year & Month

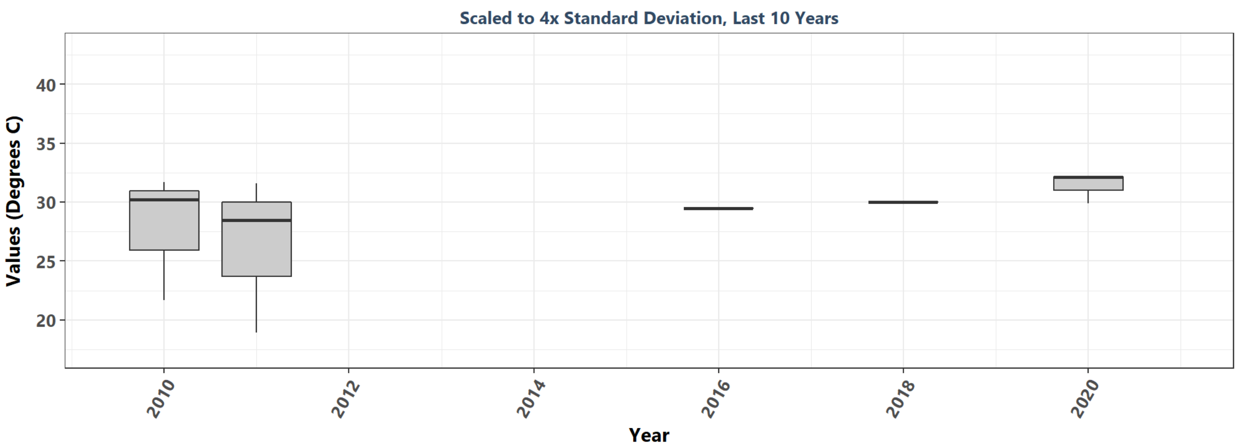


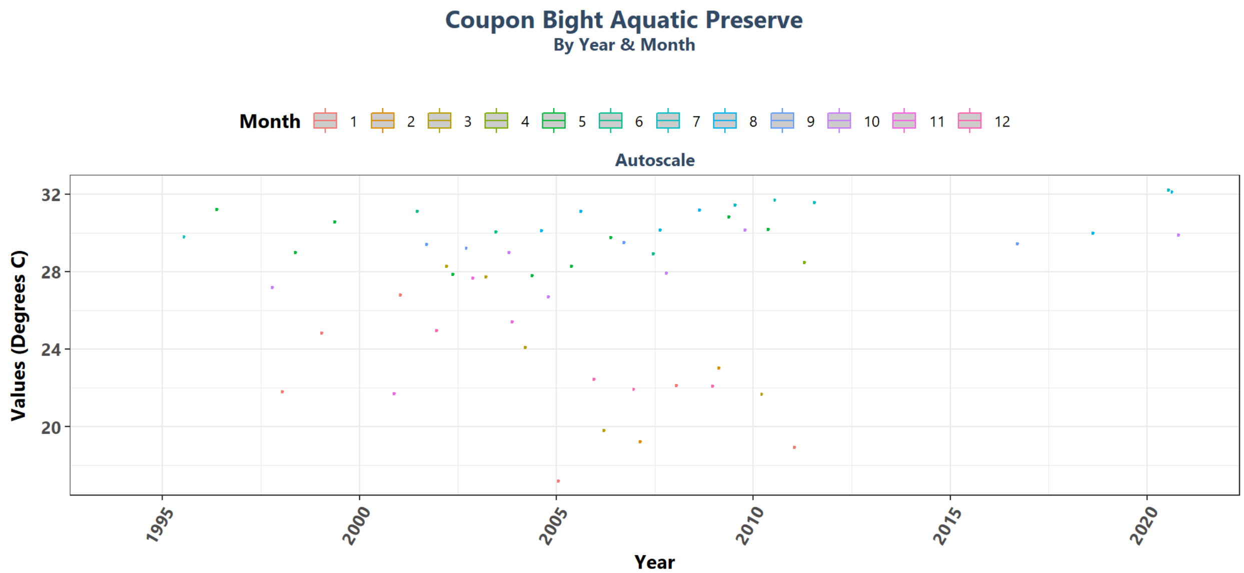

Cockroach Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

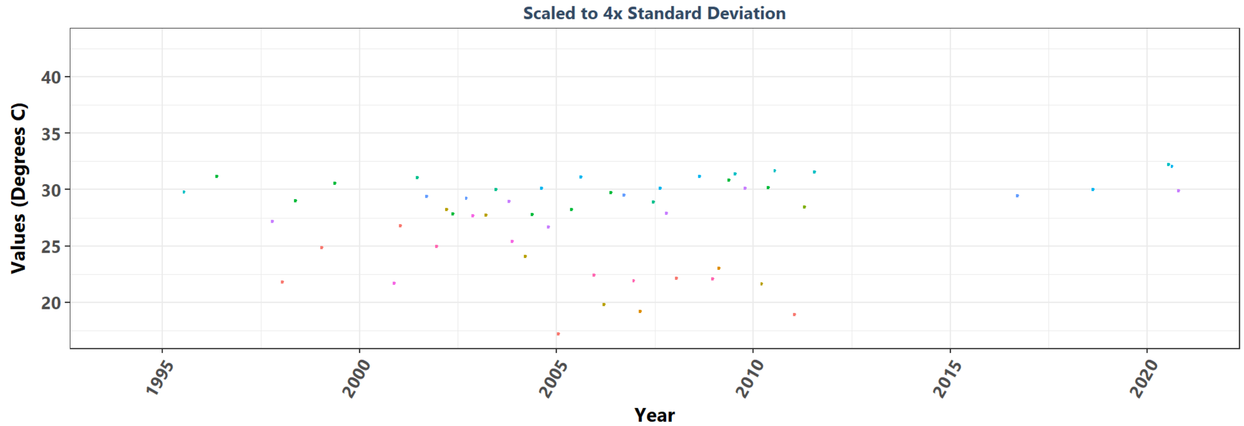
Coupon Bight Aquatic Preserve
By Year

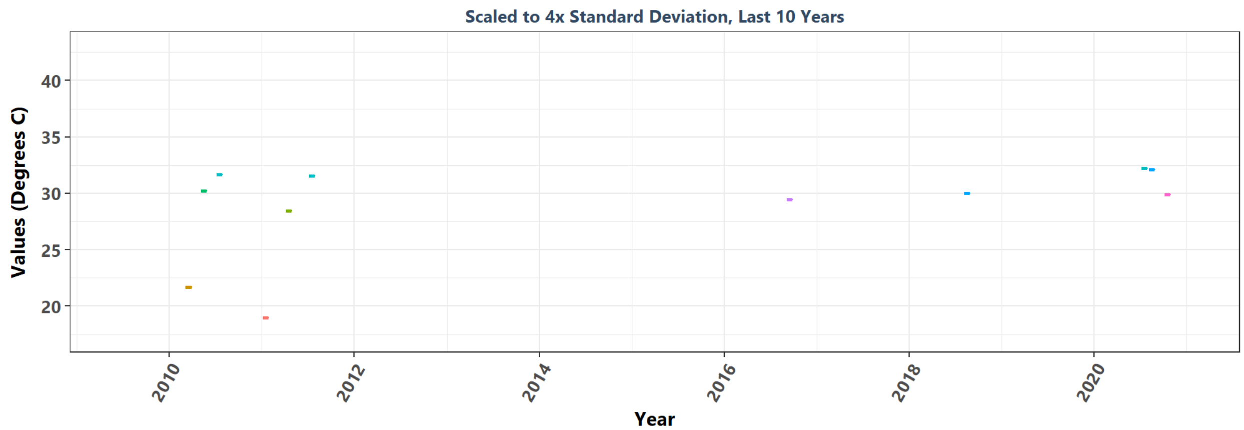




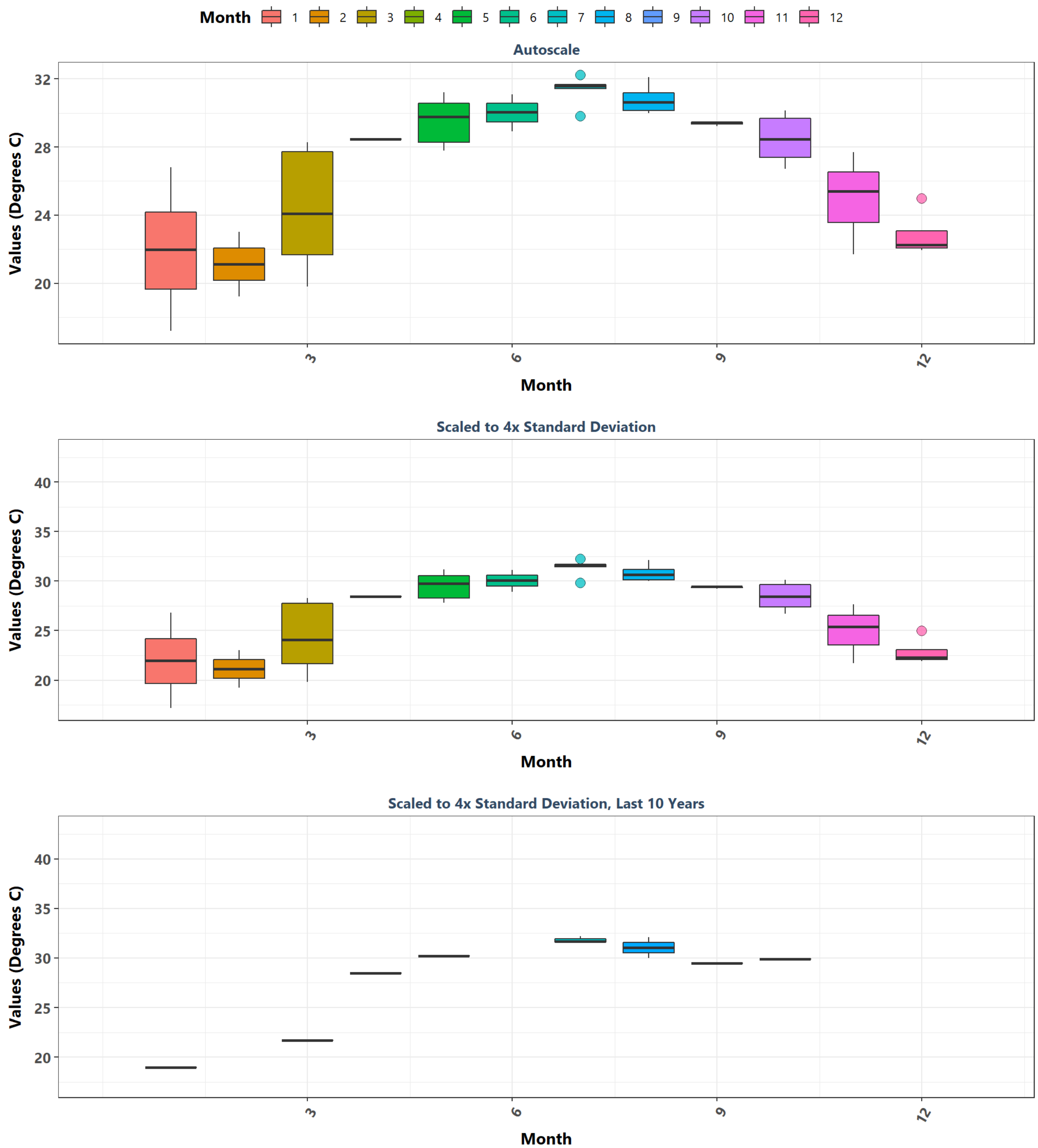



Coupon Bight Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
32
28
24
20
1995
2000
2005
2010
2015
2020
Year


Scaled to 4x Standard Deviation
Values (Degrees C)
40
35
30
25
20
1995
2000
2005
2010
2015
2020
Year

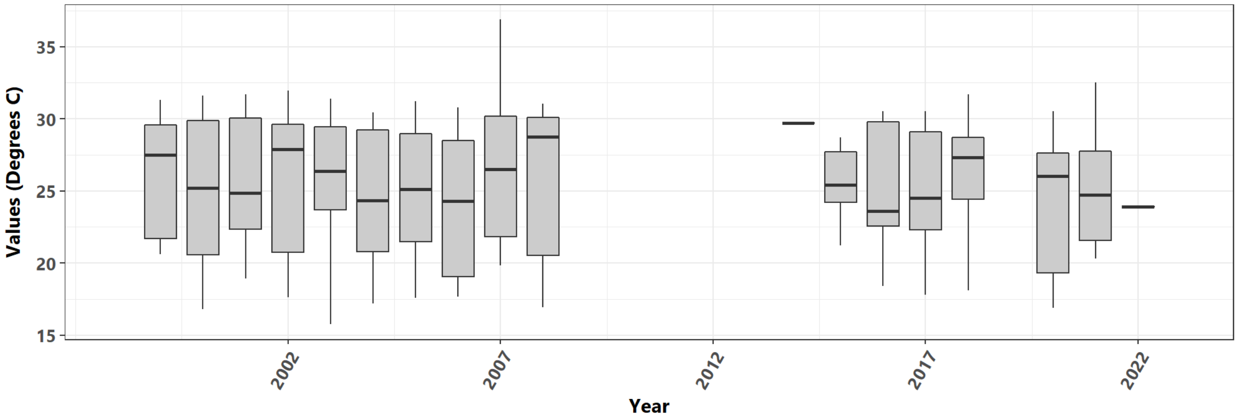

Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
35
30
25
20
2010
2012
2014
2016
2018
2020
Year

Coupon Bight Aquatic Preserve
By Month

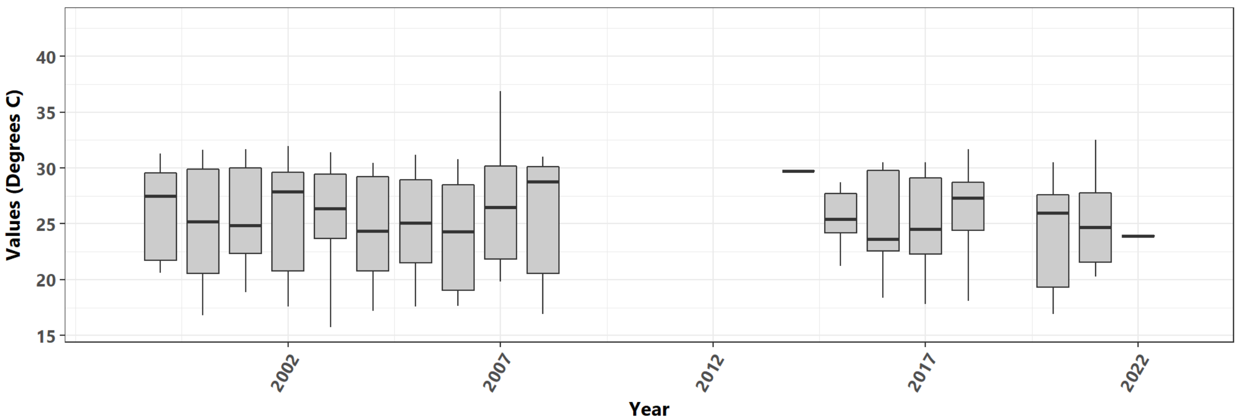


Estero Bay Aquatic Preserve
By Year

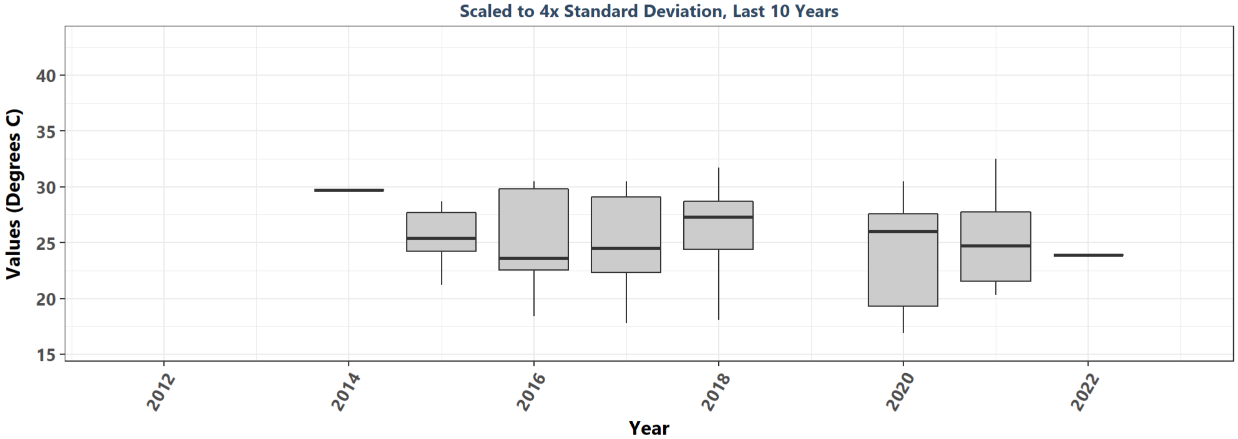
Autoscale

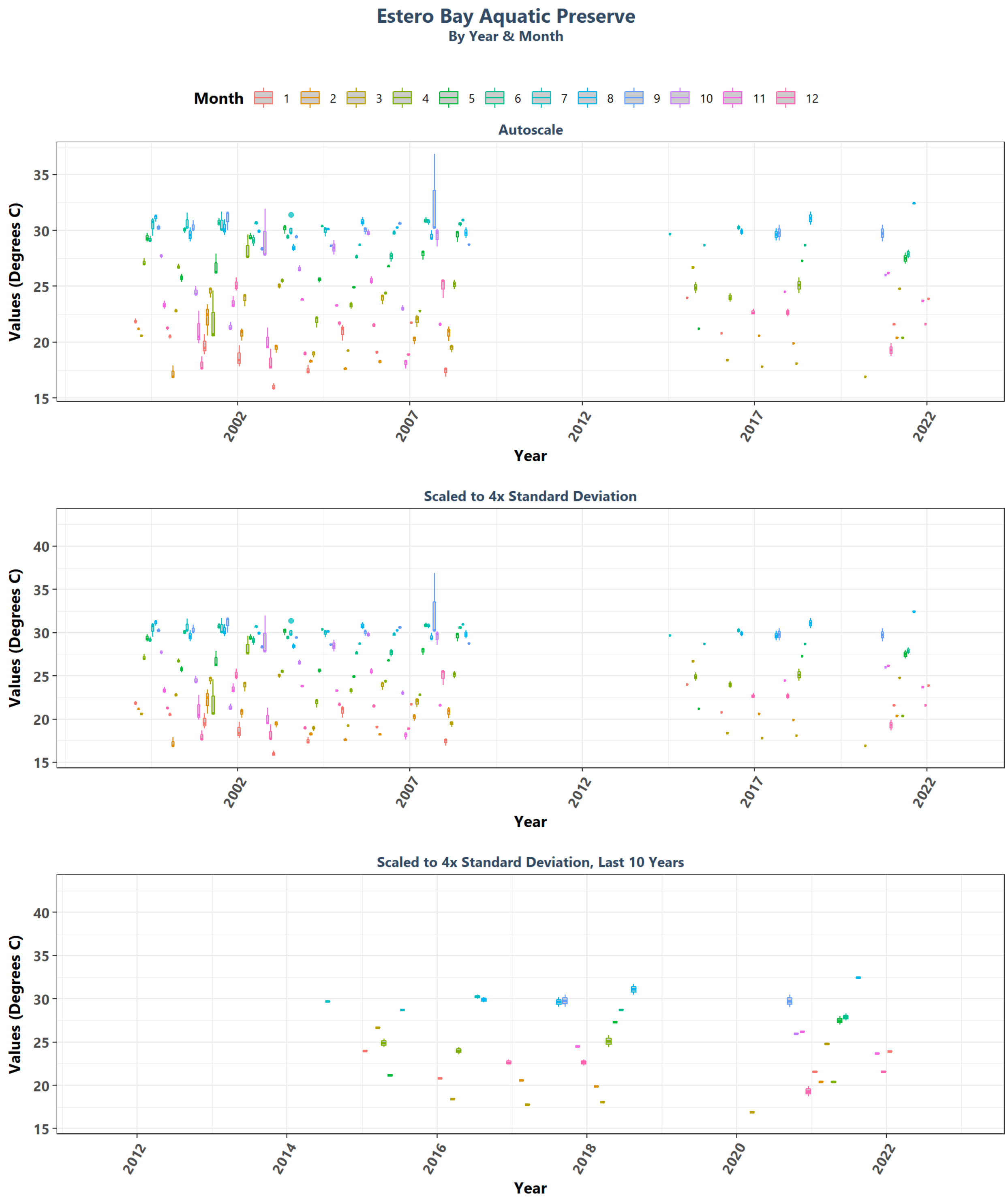


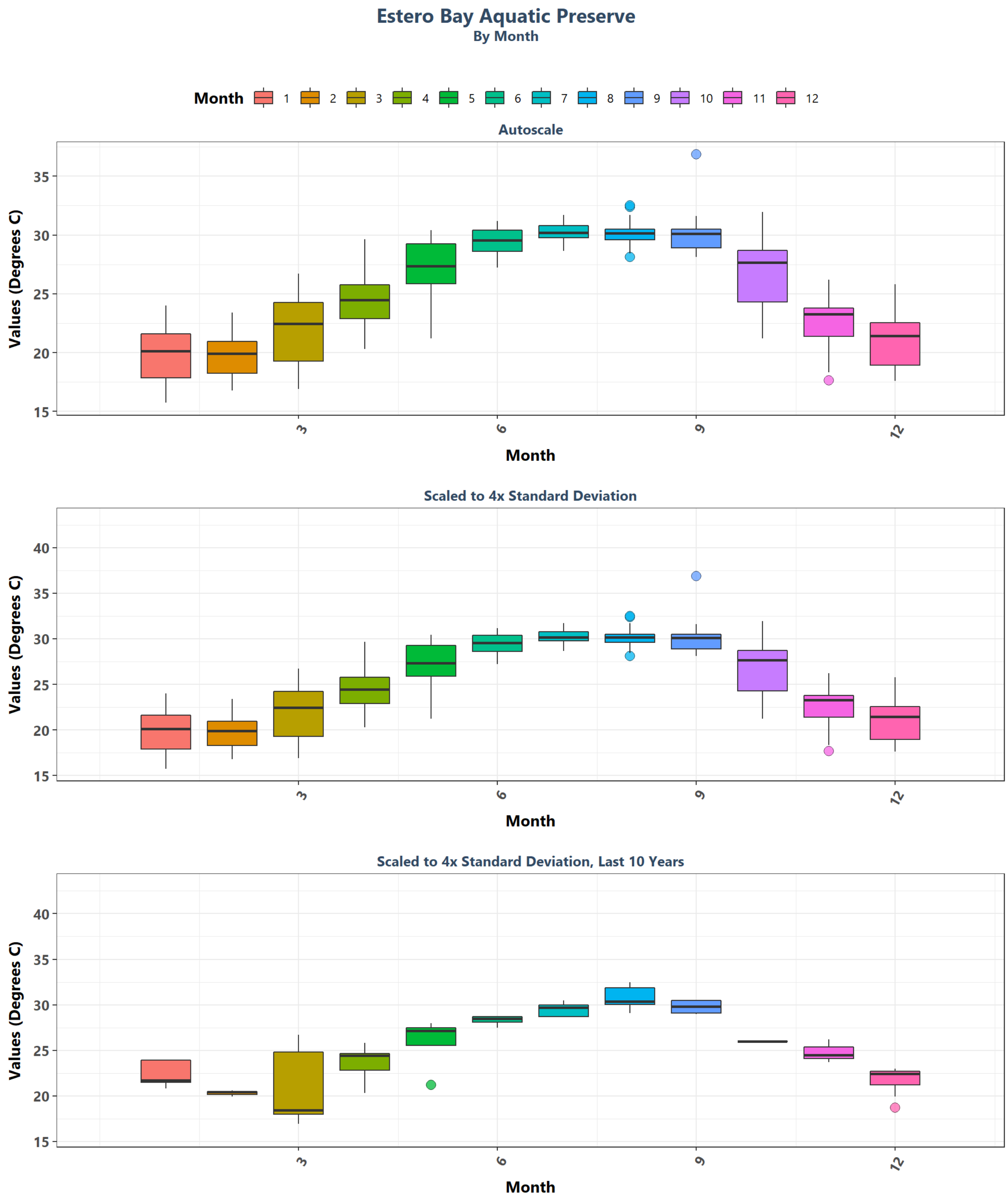
Scaled to 4x Standard Deviation



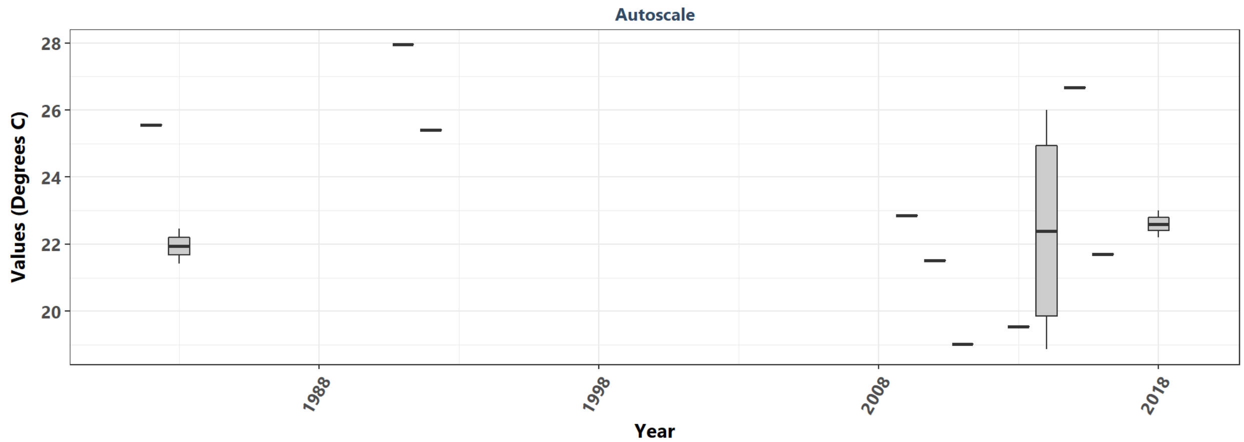
Scaled to 4x Standard Deviation, Last 10 Years

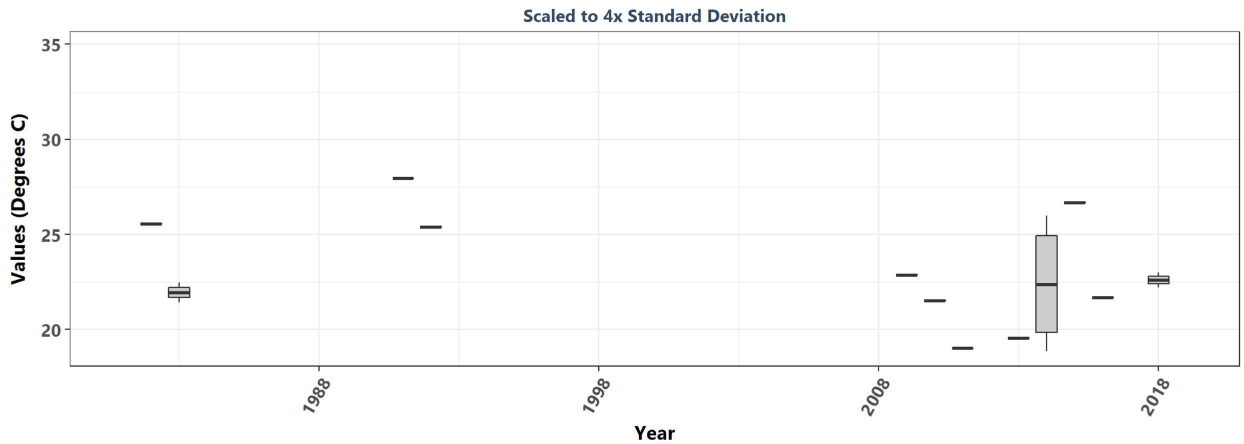


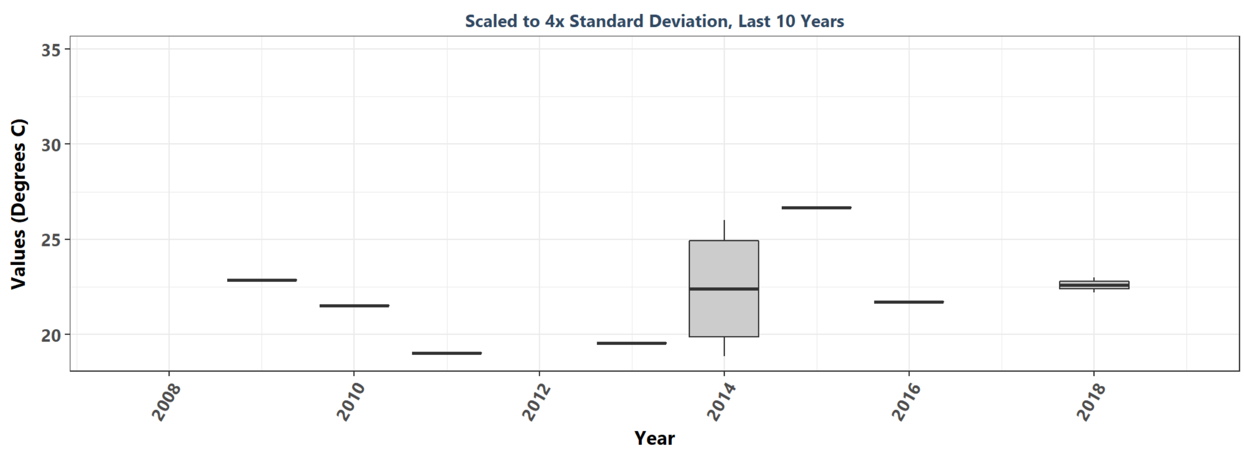

Estero Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
15 20 25 30 35
2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
15 20 25 30 35 40
2002 2007 2012 2017 2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
15 20 25 30 35 40
2012 2014 2016 2018 2020 2022
Year

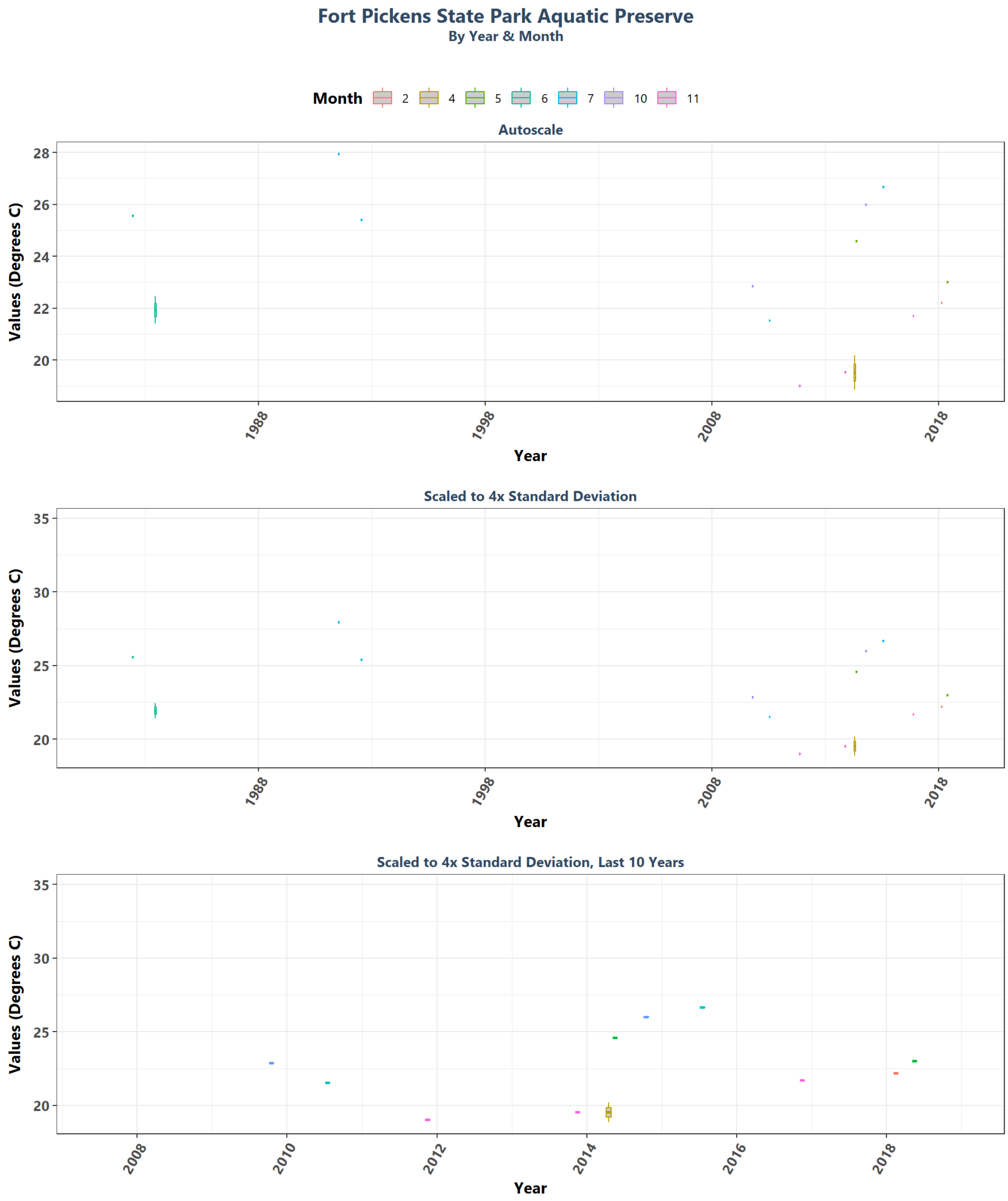

Estero Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
35
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
35
30
25
20
15
3
6
9
12
Month

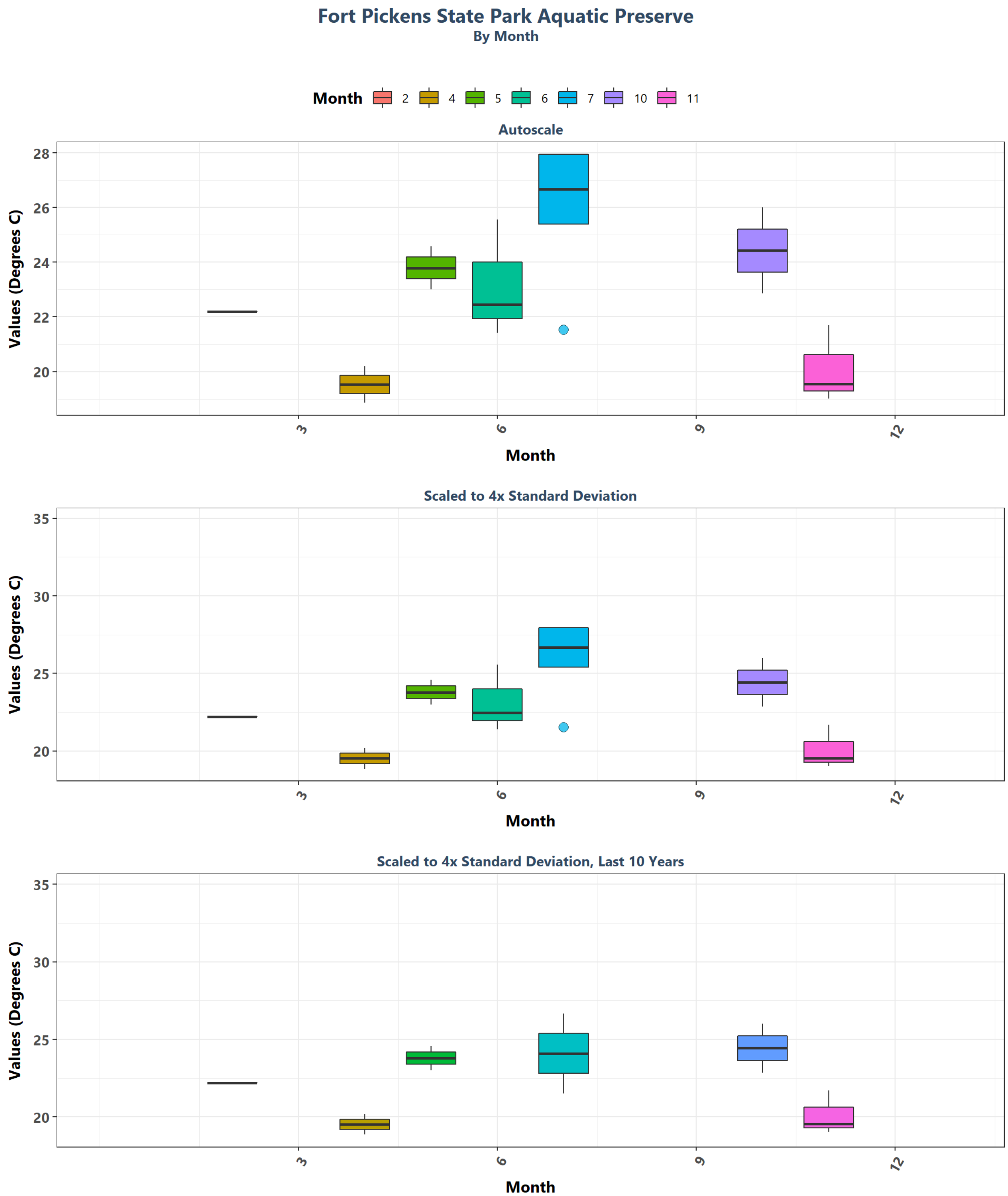
Fort Pickens State Park Aquatic Preserve
By Year





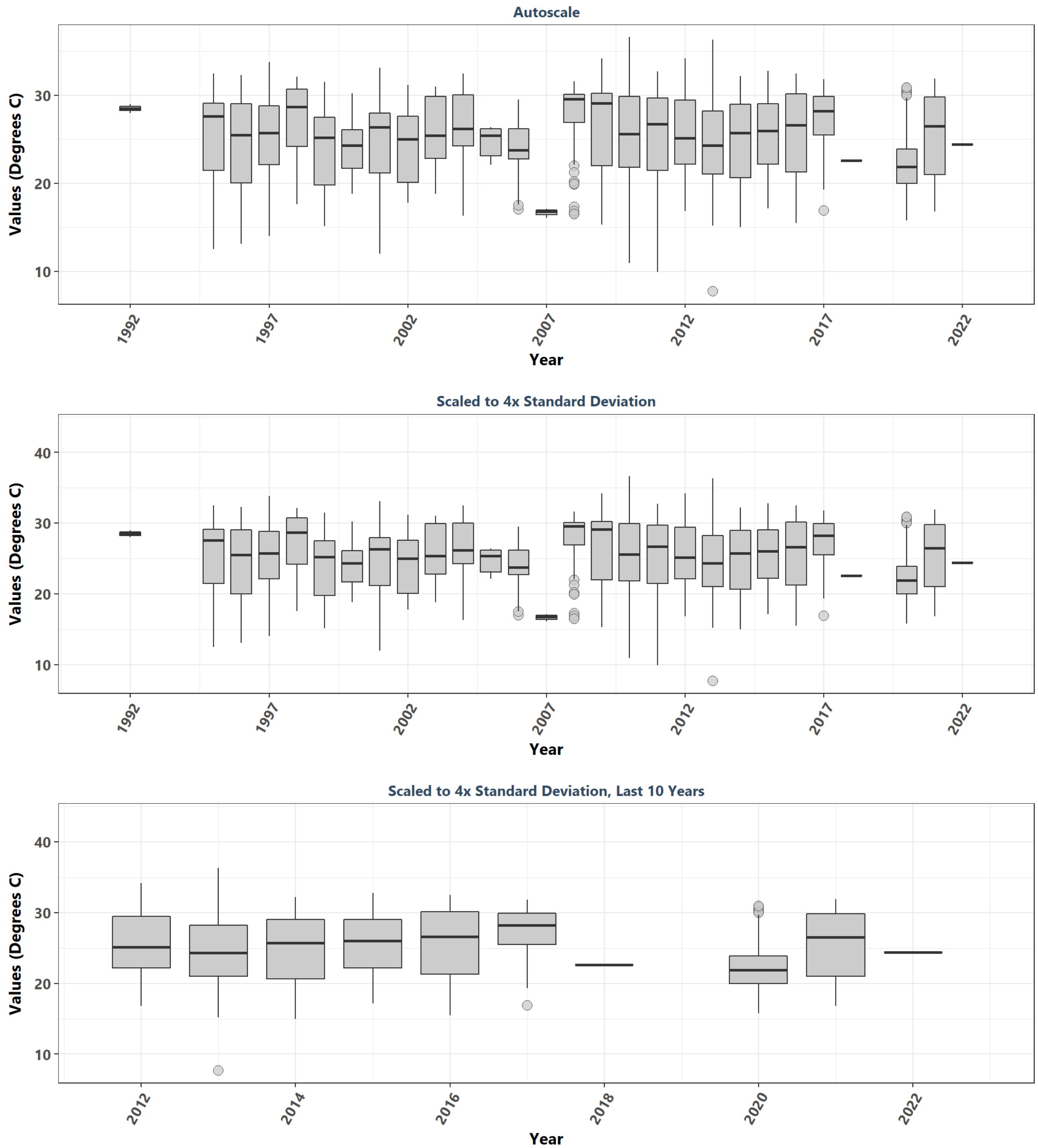


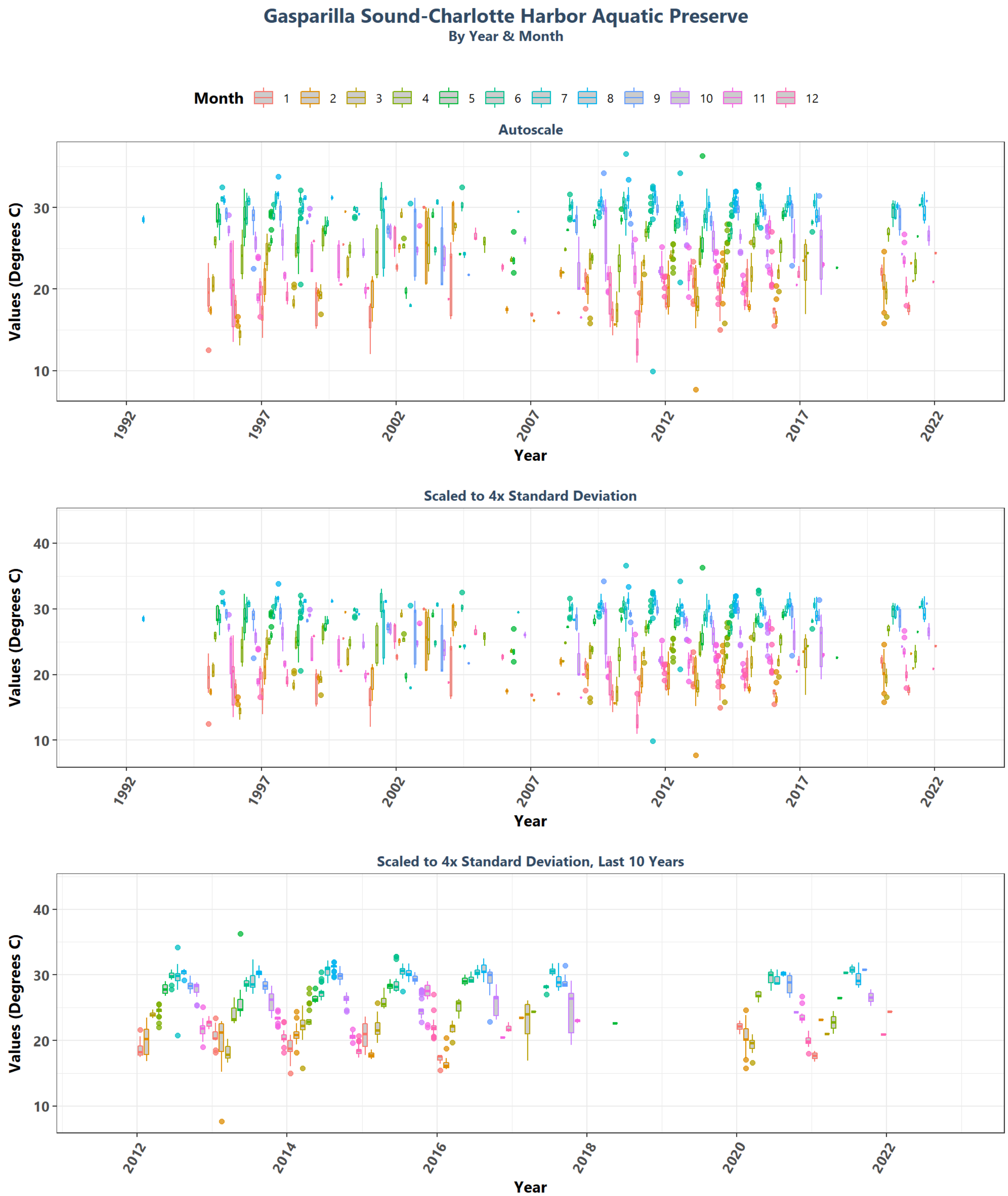

Fort Pickens State Park Aquatic Preserve
By Year & Month
Month 2 4 5 6 7 10 11
Autoscale
Values (Degrees C)
28
26
24
22
20
1988
1998
2008
2018
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
35
30
25
20
1988
1998
2008
2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
35
30
25
20
2008
2010
2012
2014
2016
2018
Year


Fort Pickens State Park Aquatic Preserve
By Month
Month 2 4 5 6 7 10 11
Autoscale
Values (Degrees C)
28
26
24
22
20
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
35
30
25
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
35
30
25
20
3
6
9
12
Month

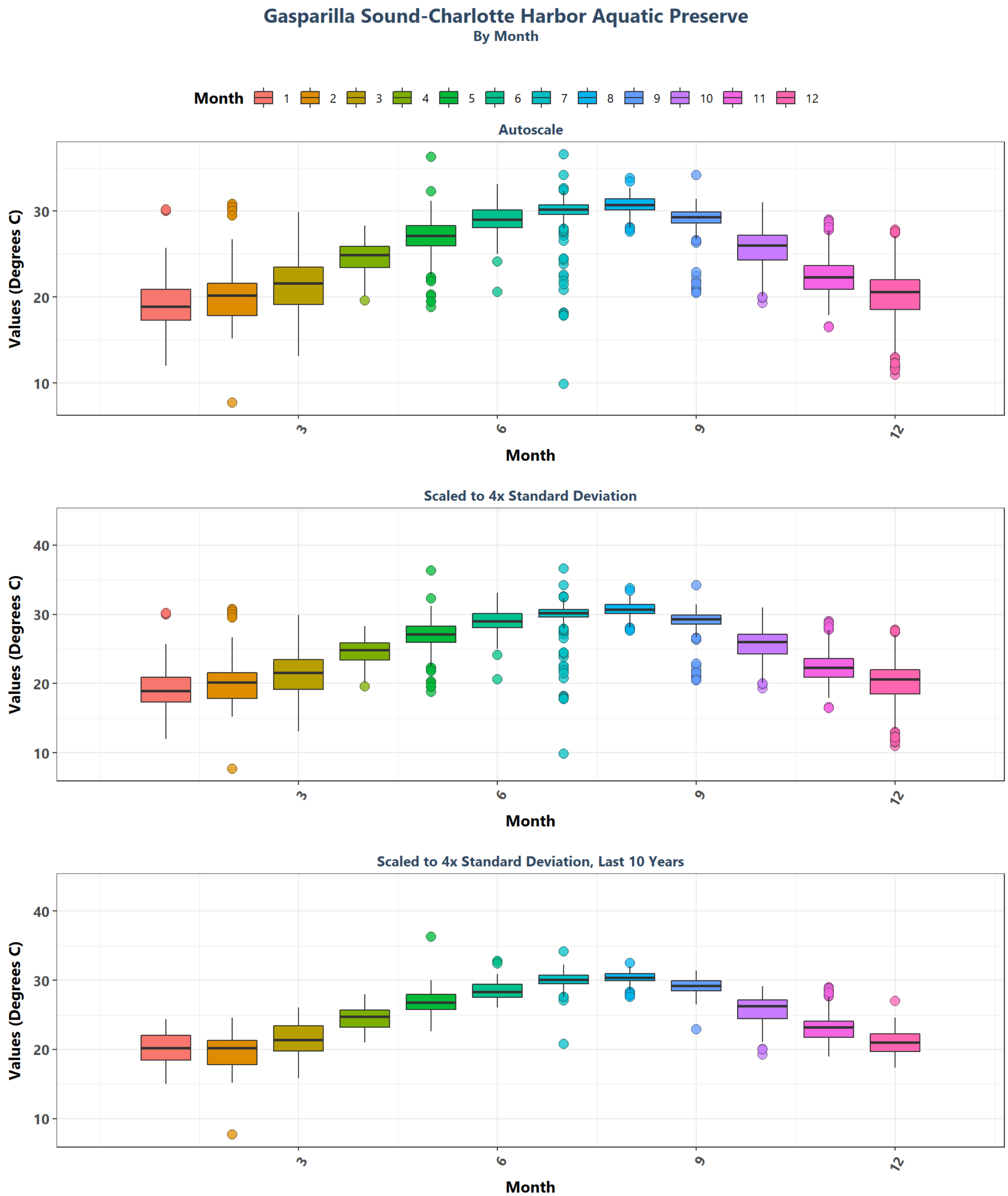
Gasparilla Sound-Charlotte Harbor Aquatic Preserve

By Year



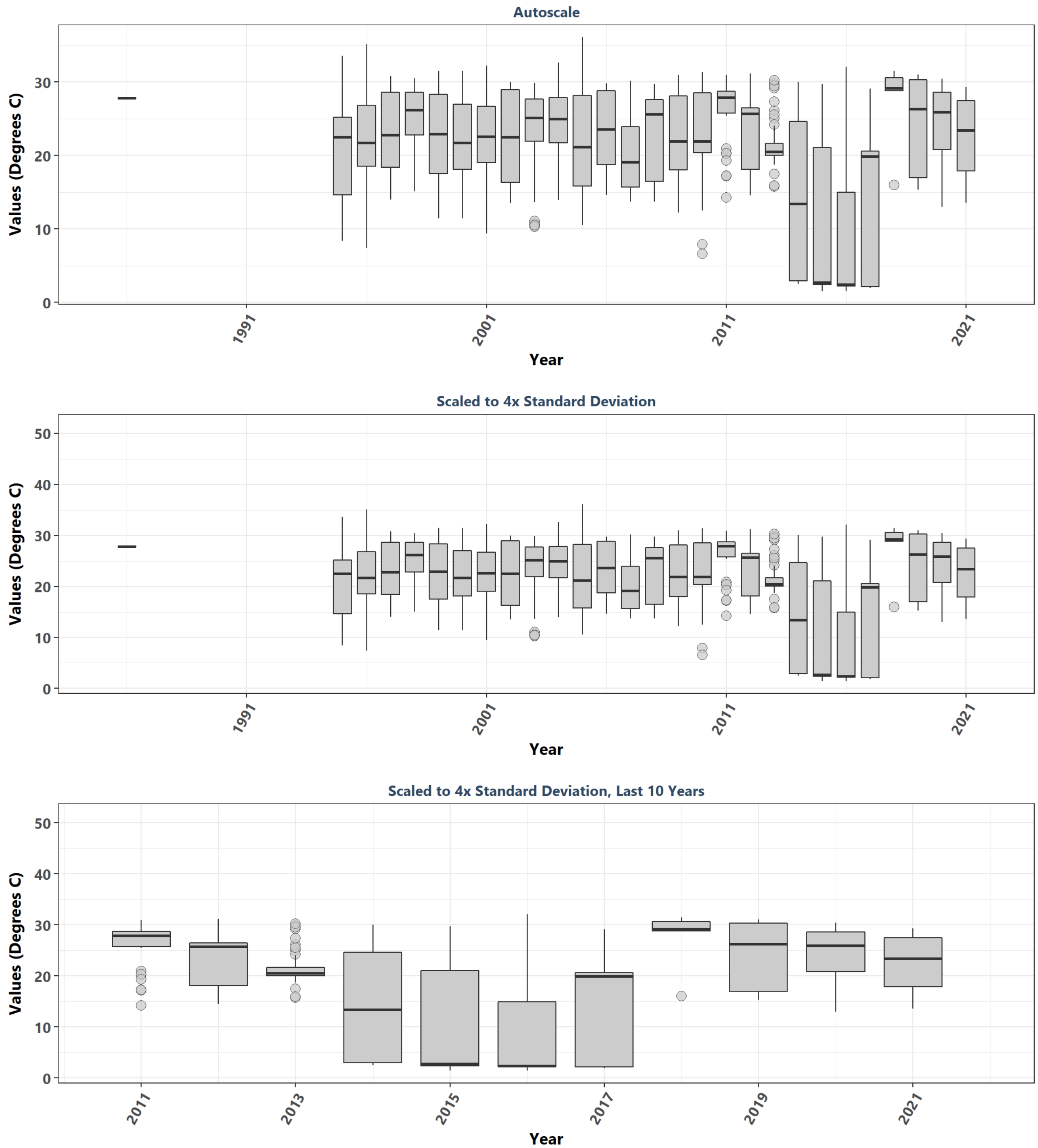


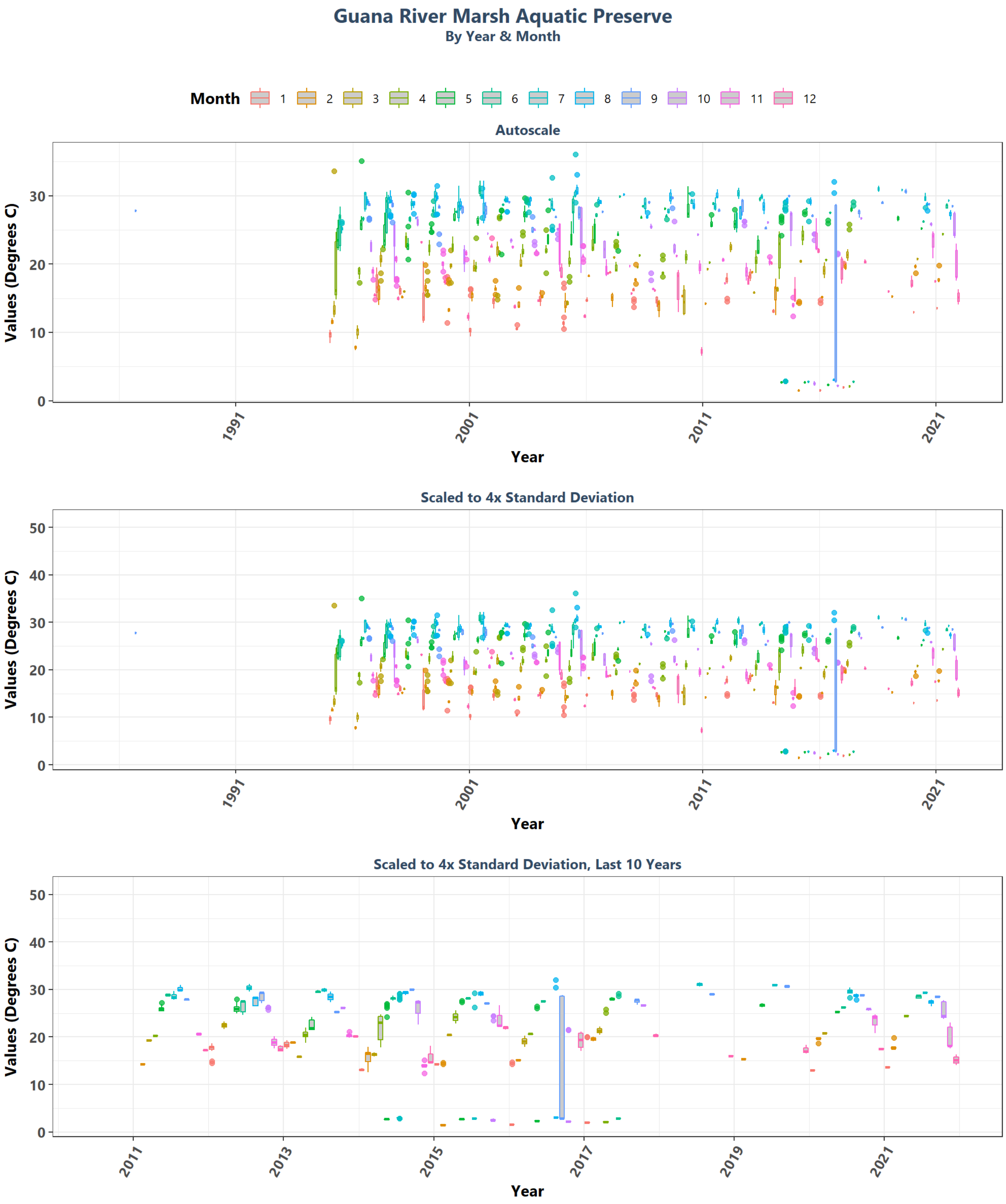
Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1992
1997
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1992
1997
2002
2007
2012
2017
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2012
2014
2016
2018
2020
2022
Year



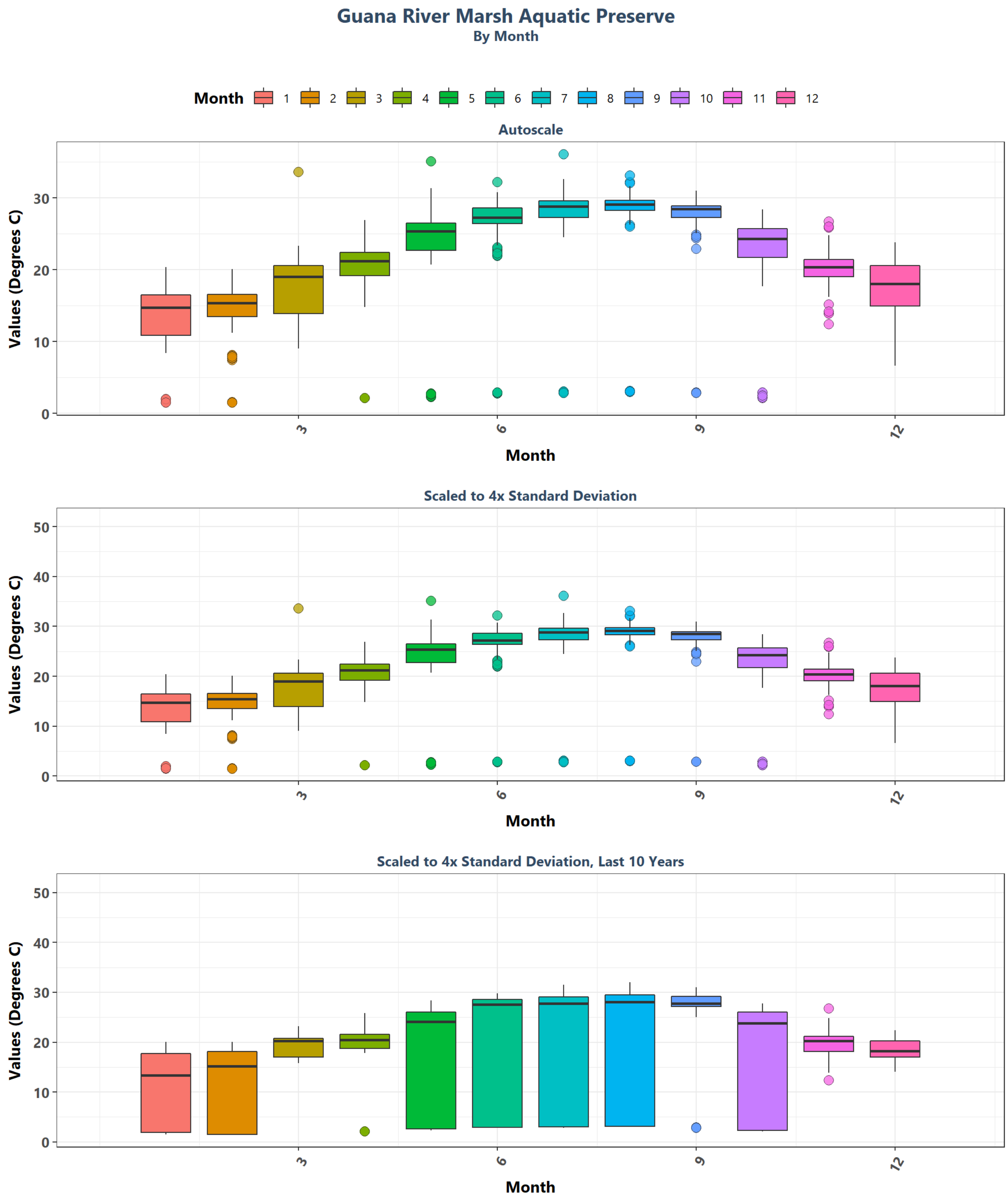
Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

Guana River Marsh Aquatic Preserve
By Year





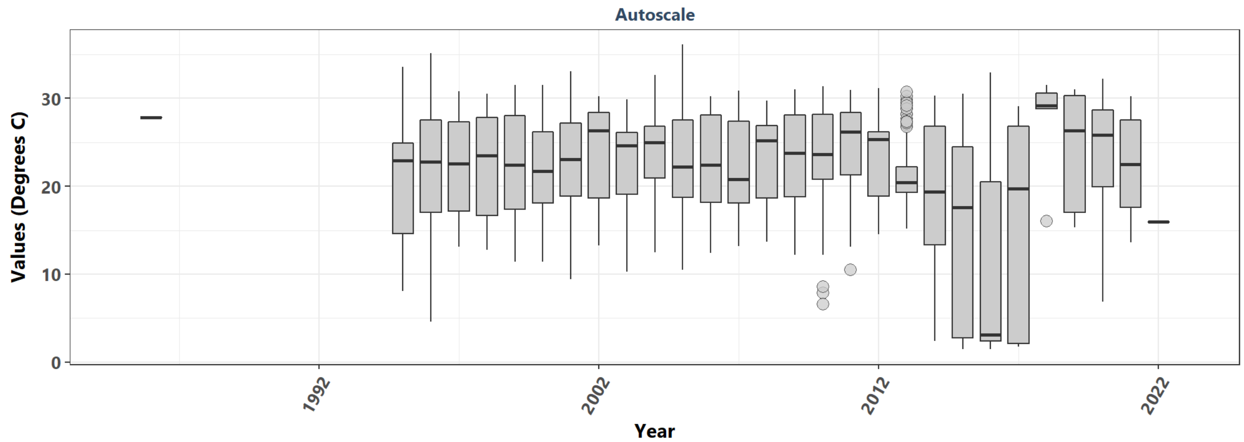
Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
50
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
50
2011
2013
2015
2017
2019
2021
Year

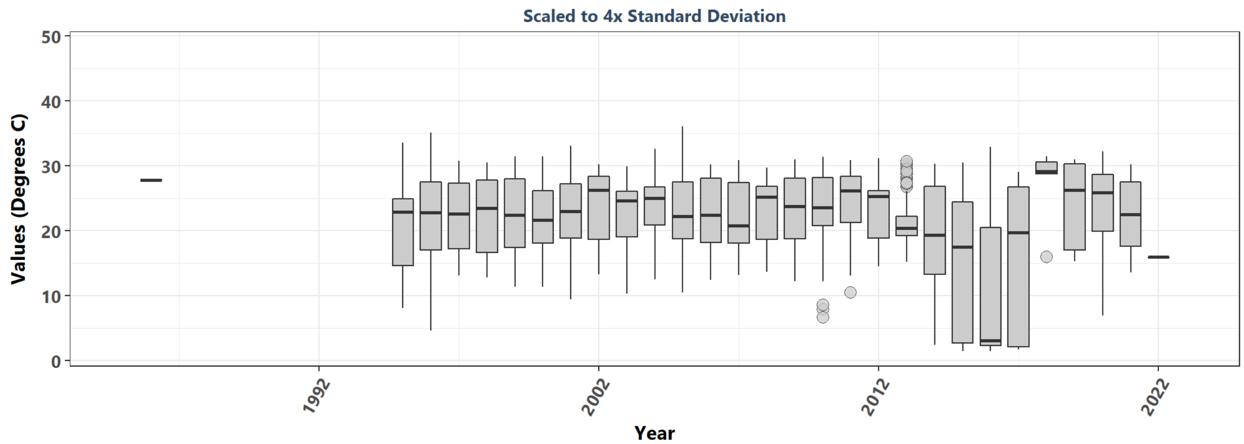


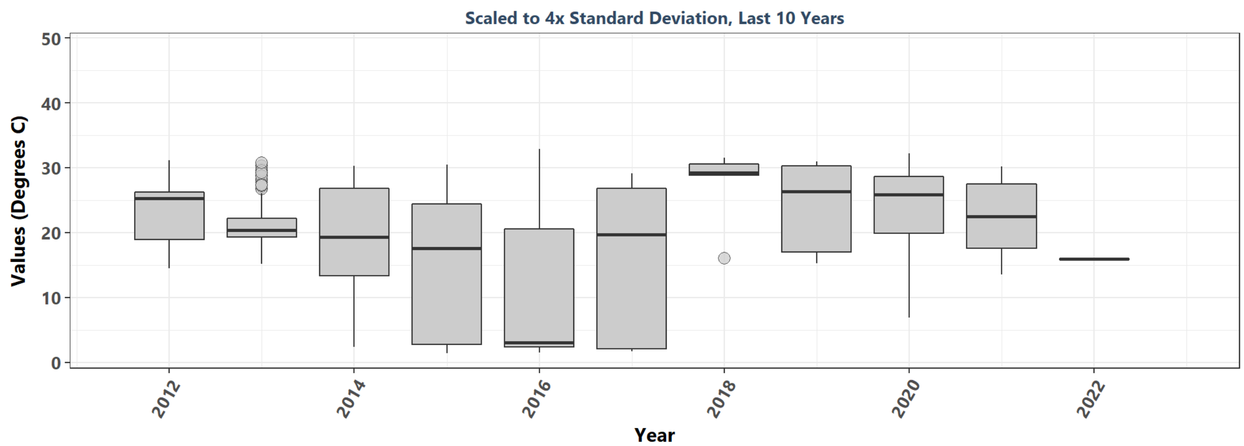
Guana River Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month

Guana Tolomato Matanzas National Estuarine Research Reserve

By Year

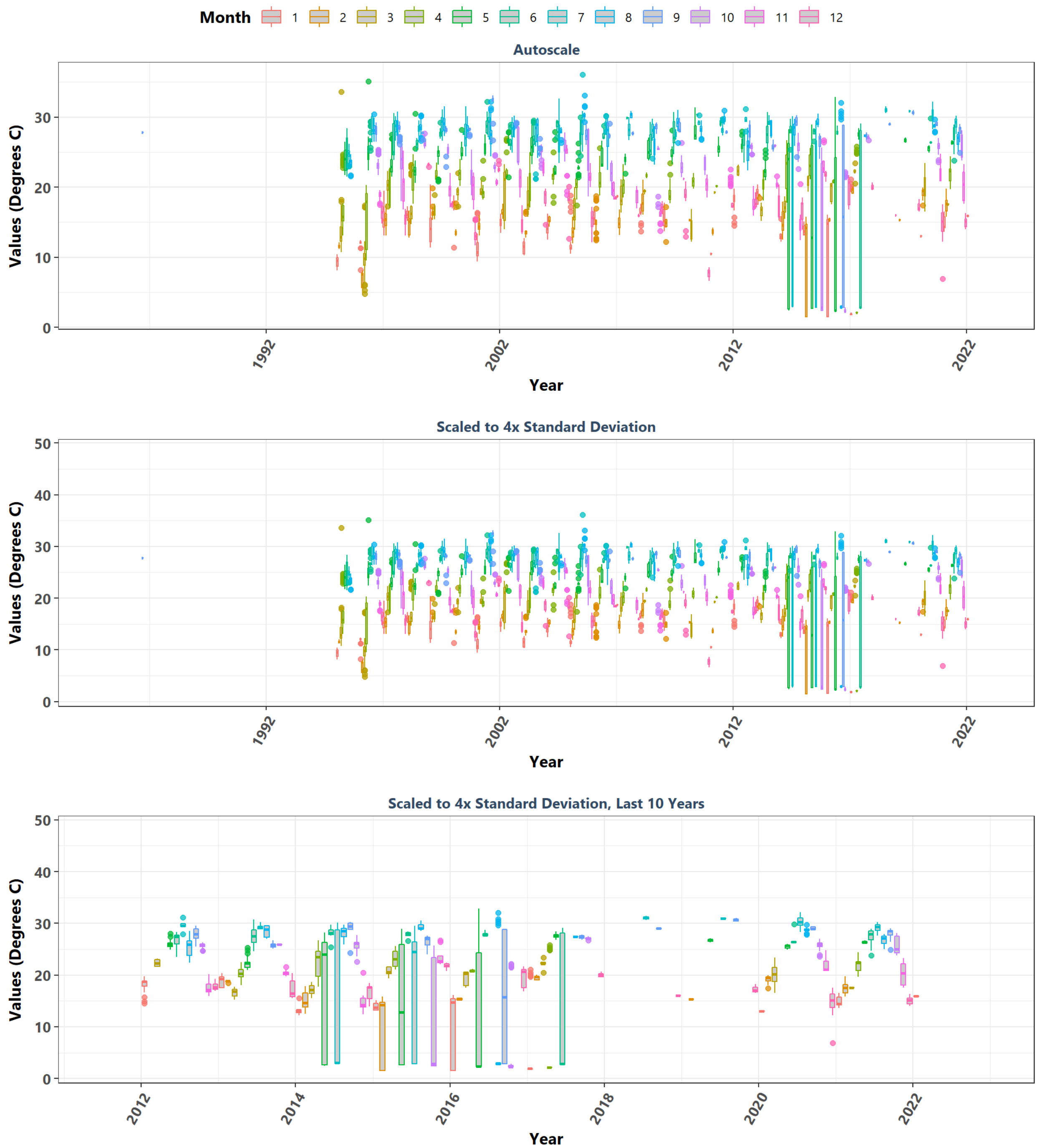


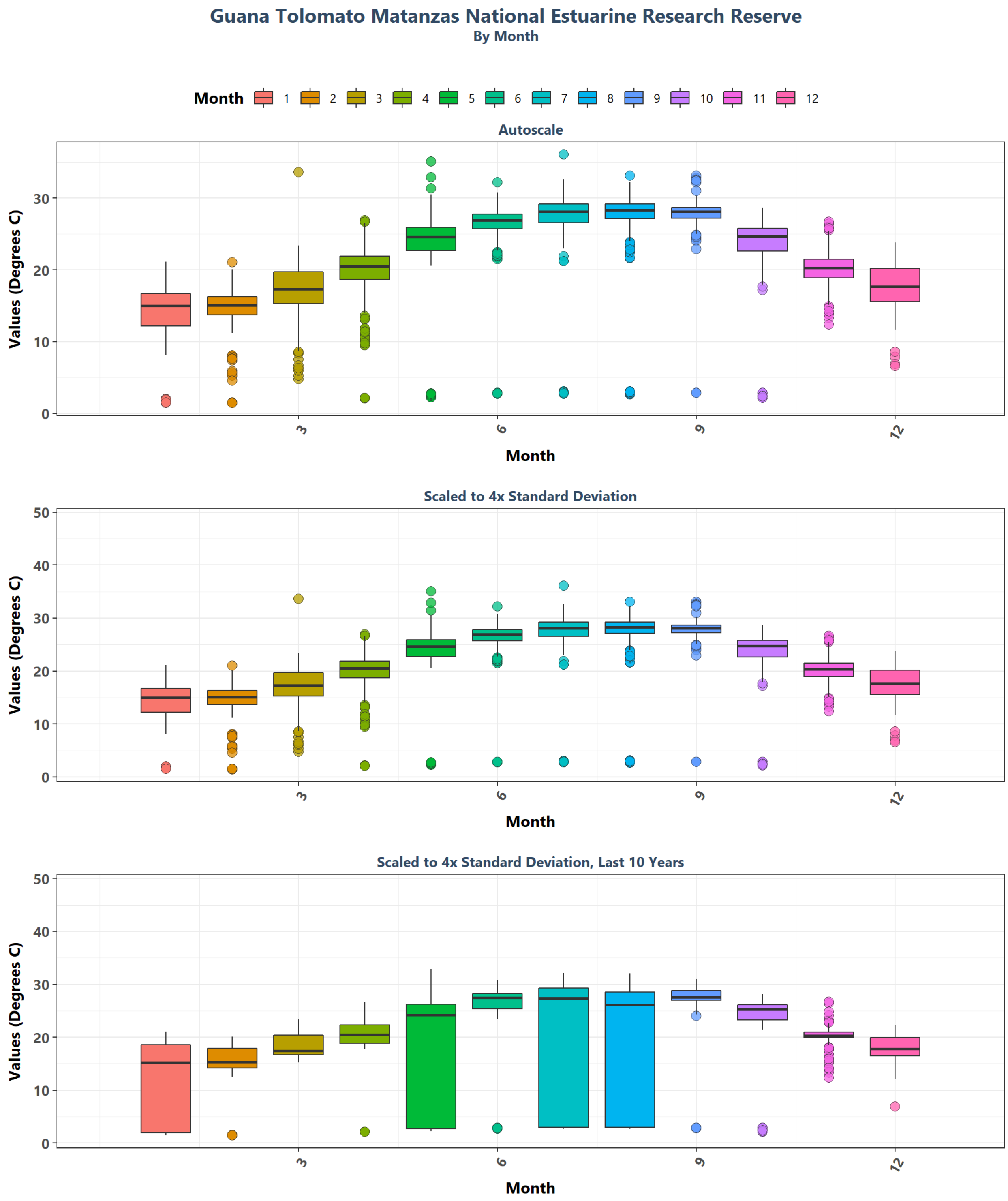




Guana Tolomato Matanzas National Estuarine Research Reserve

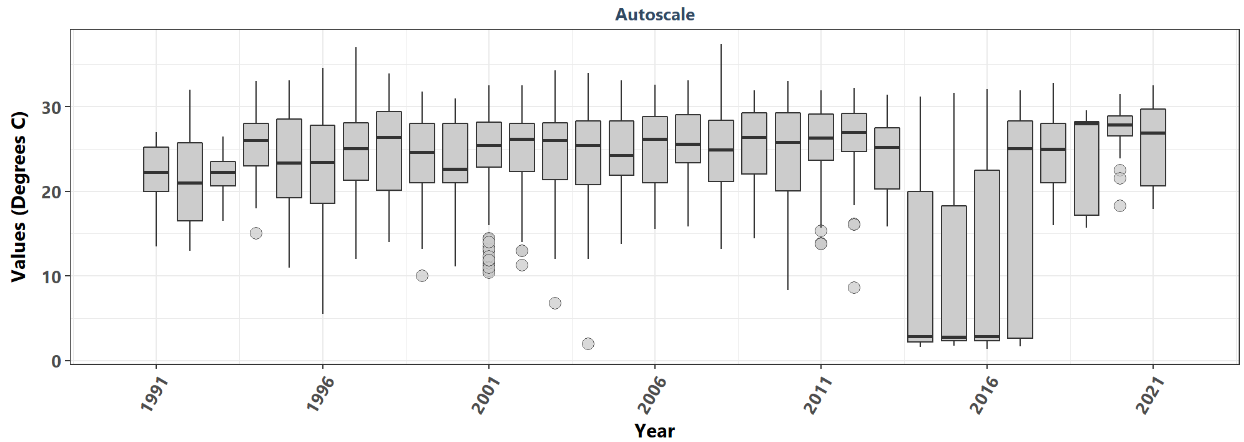
By Year & Month

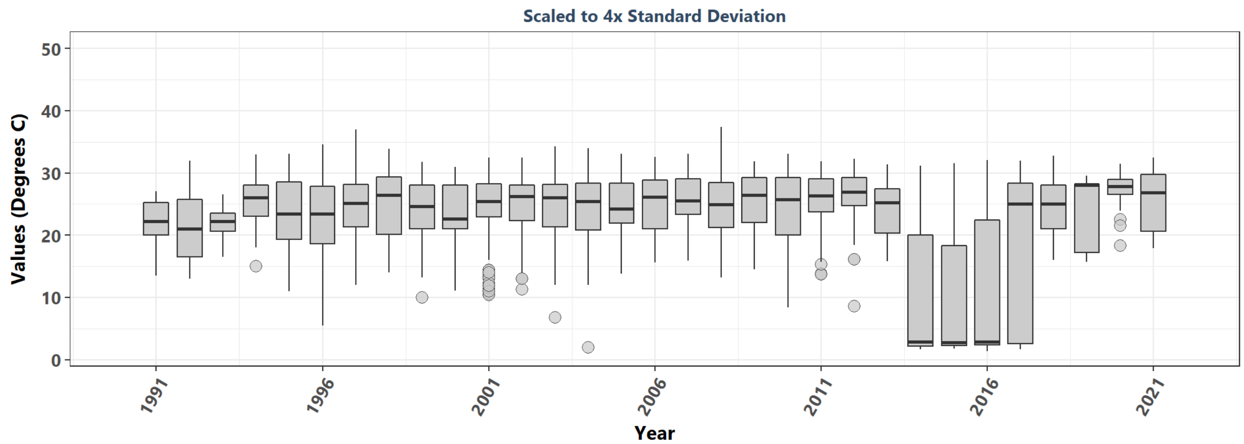


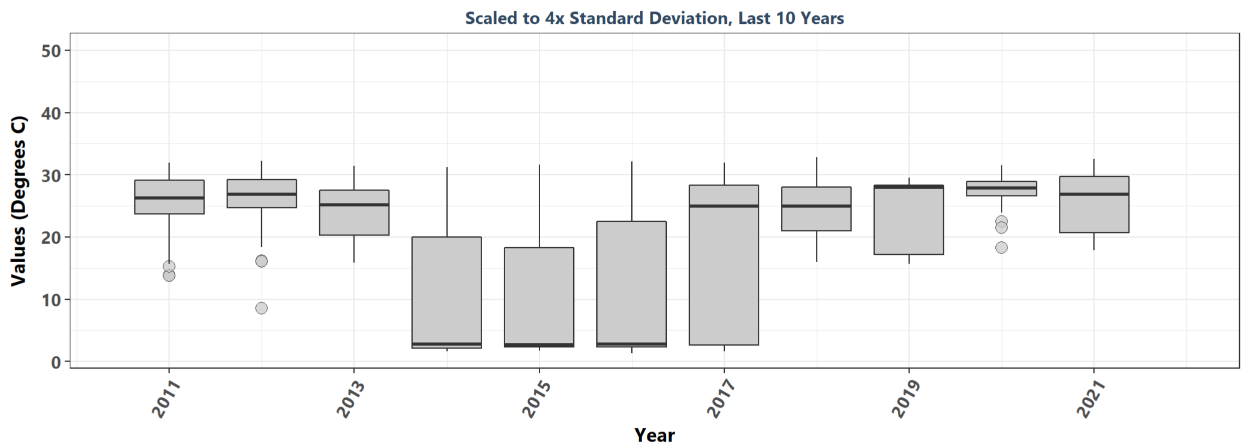

Guana Tolomato Matanzas National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month

Indian River-Malabar to Vero Beach Aquatic Preserve

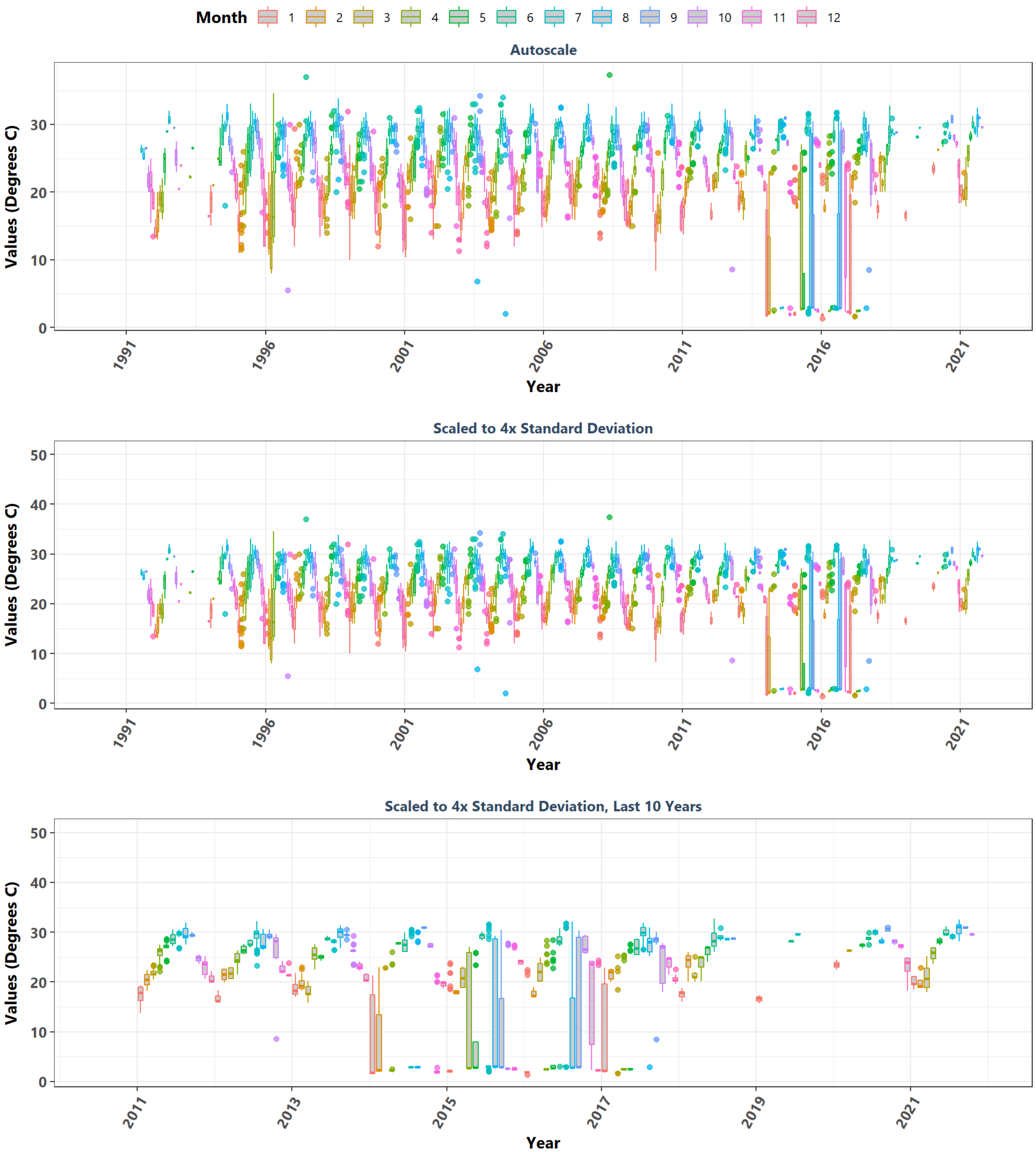
By Year

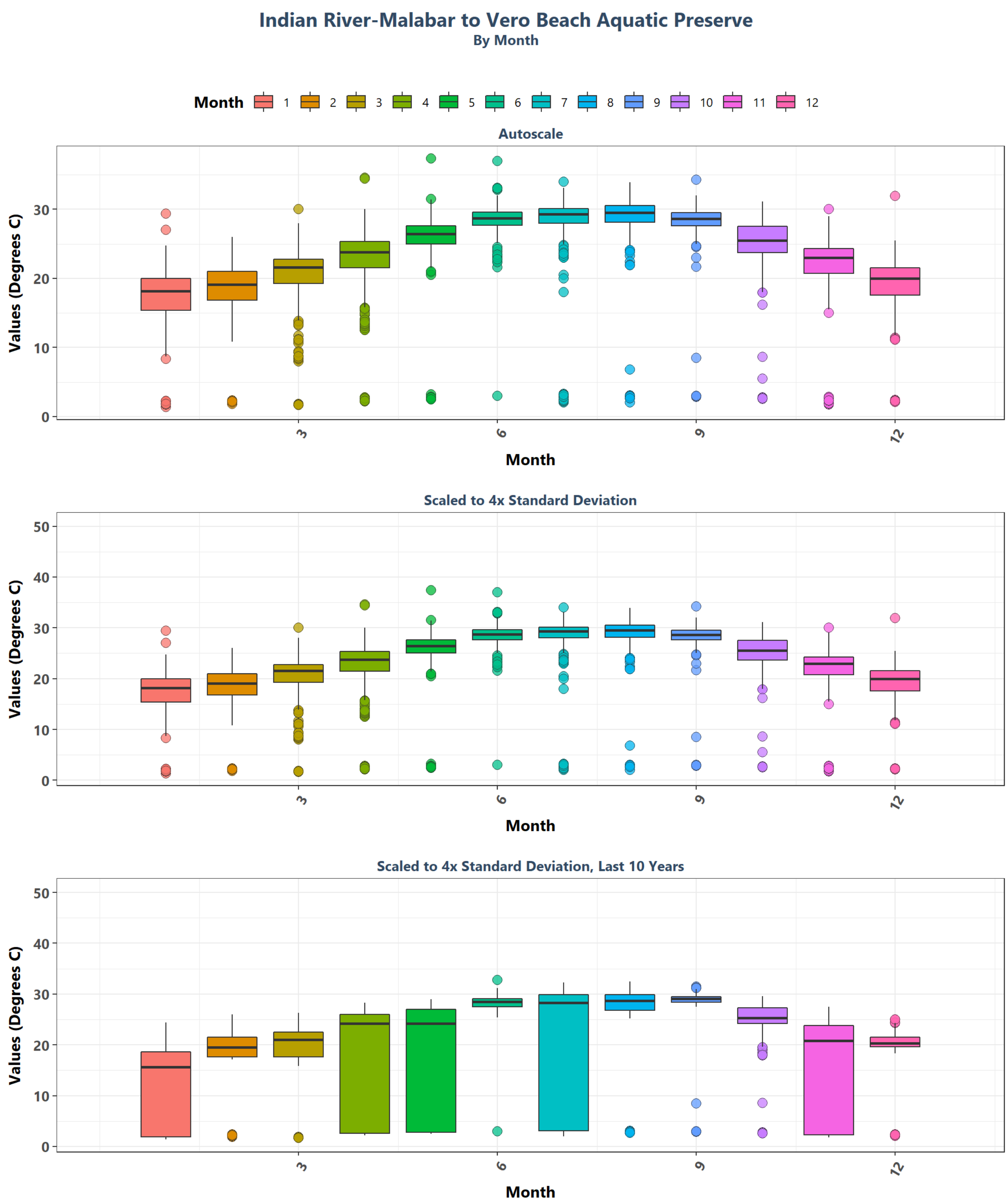






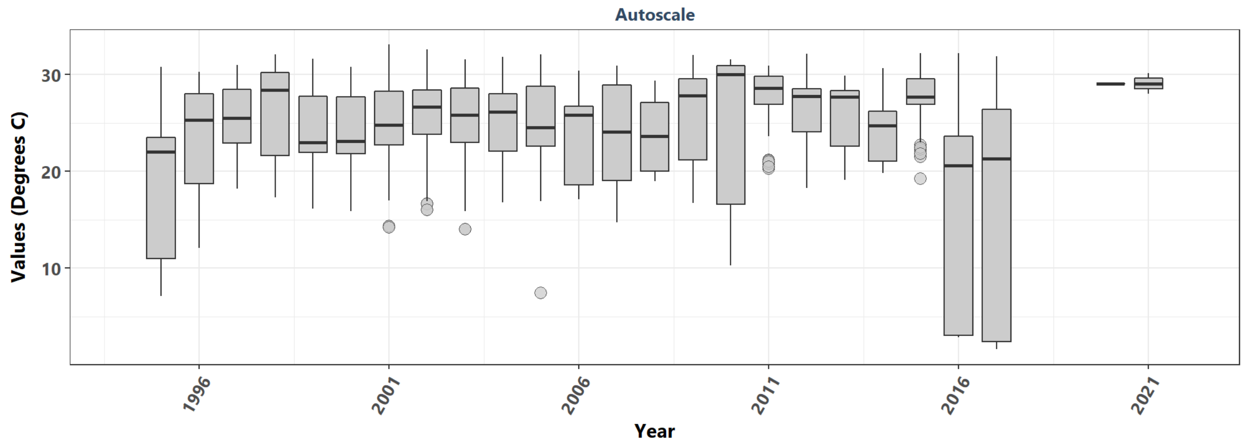
Indian River-Malabar to Vero Beach Aquatic Preserve
By Year & Month

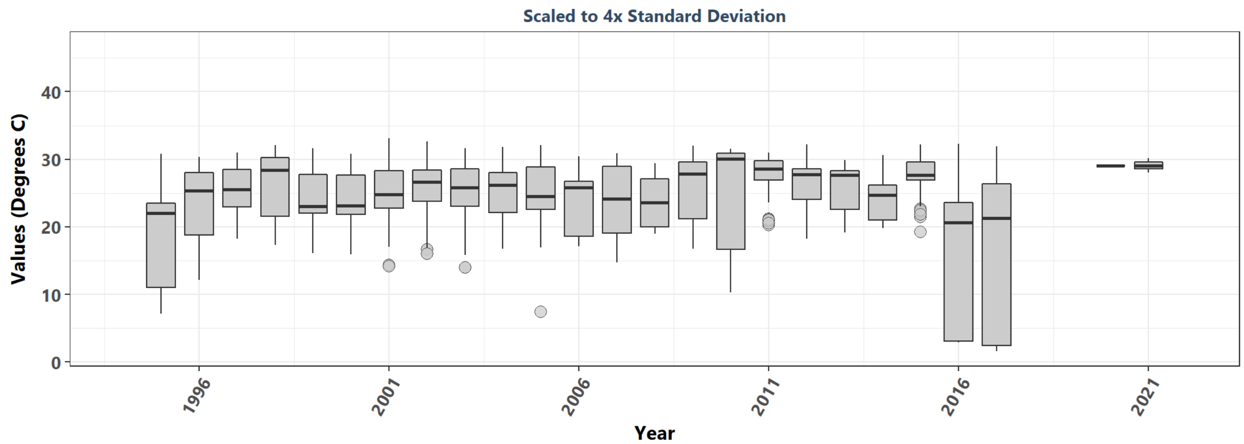


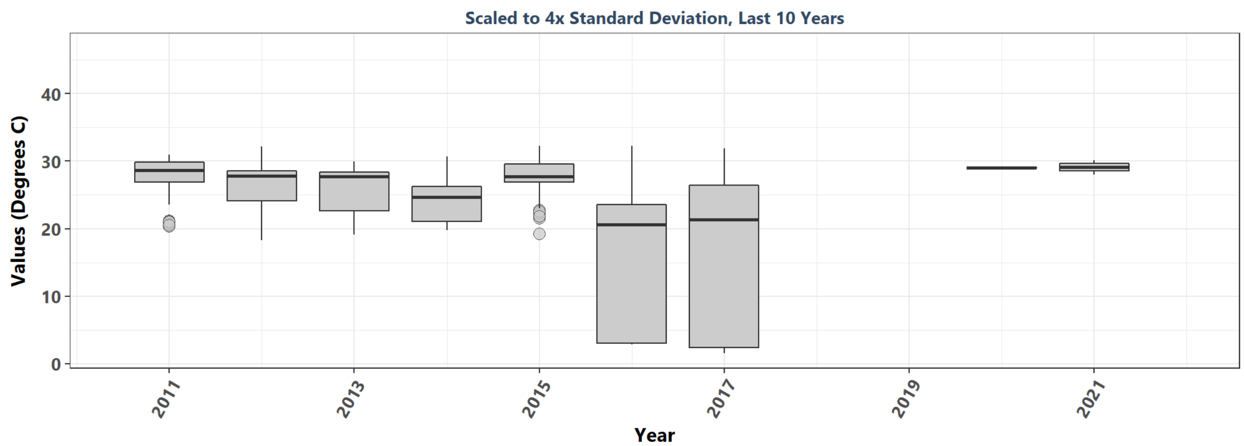

Indian River-Malabar to Vero Beach Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
50
3
6
9
12
Month

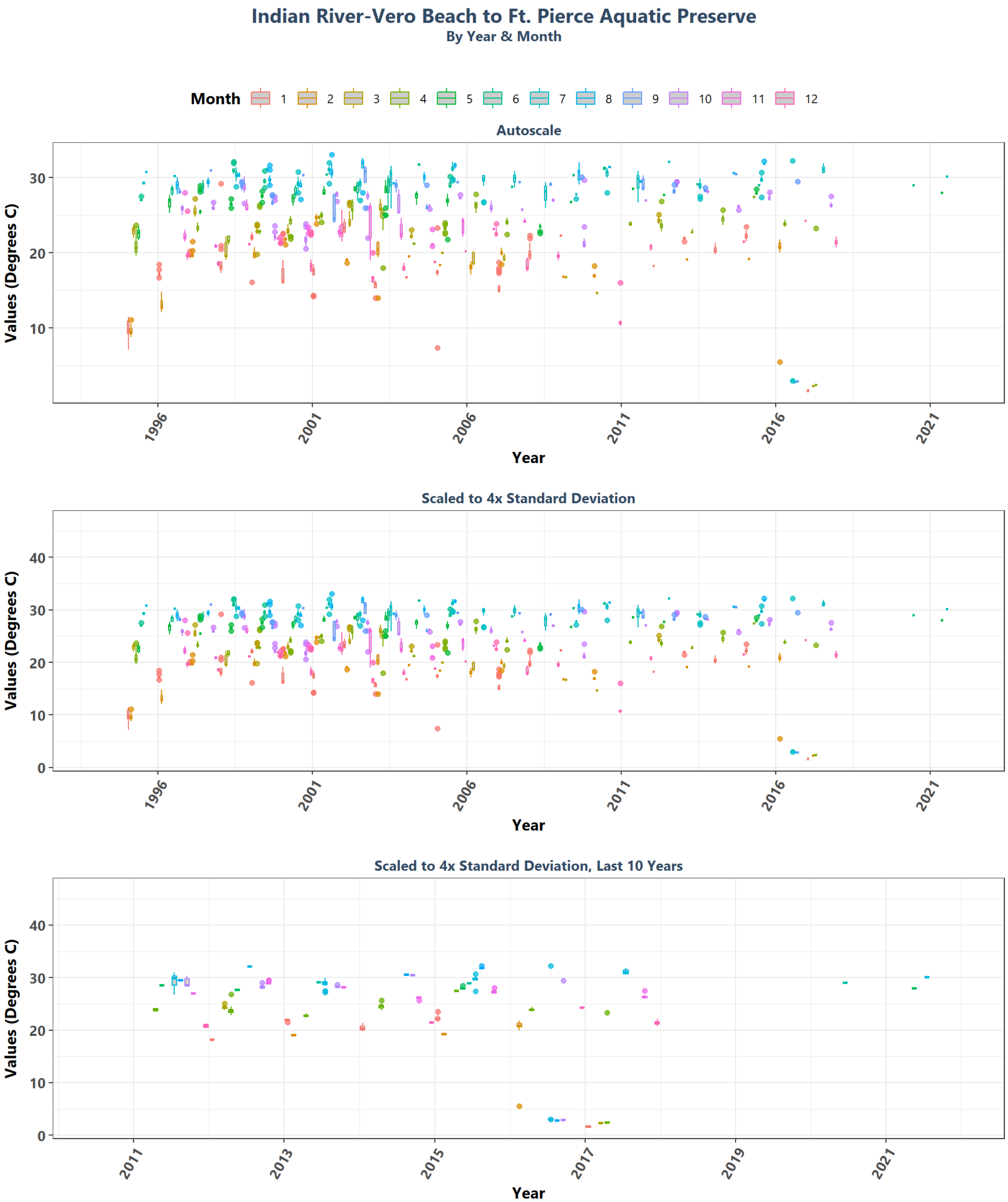
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

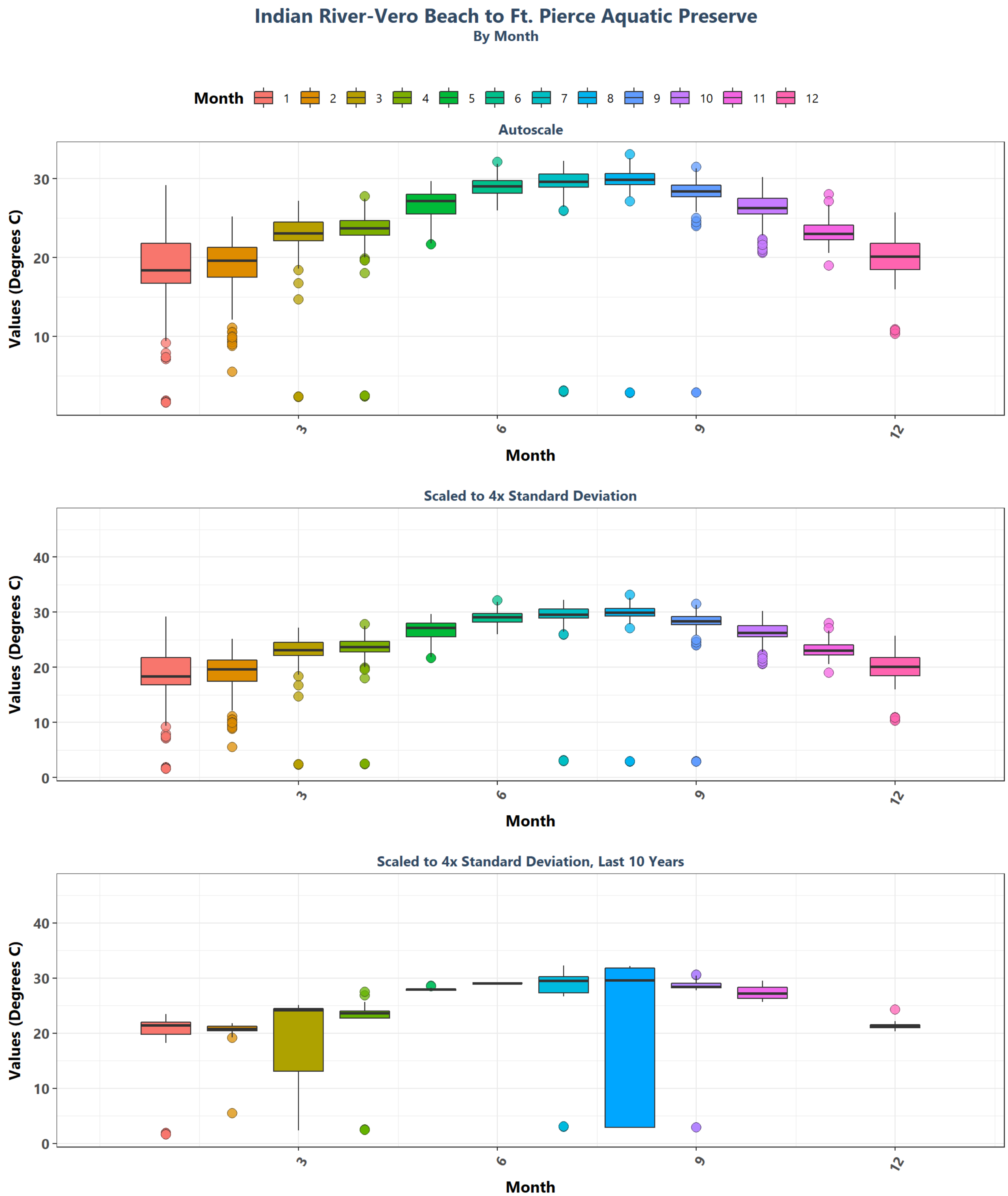
By Year





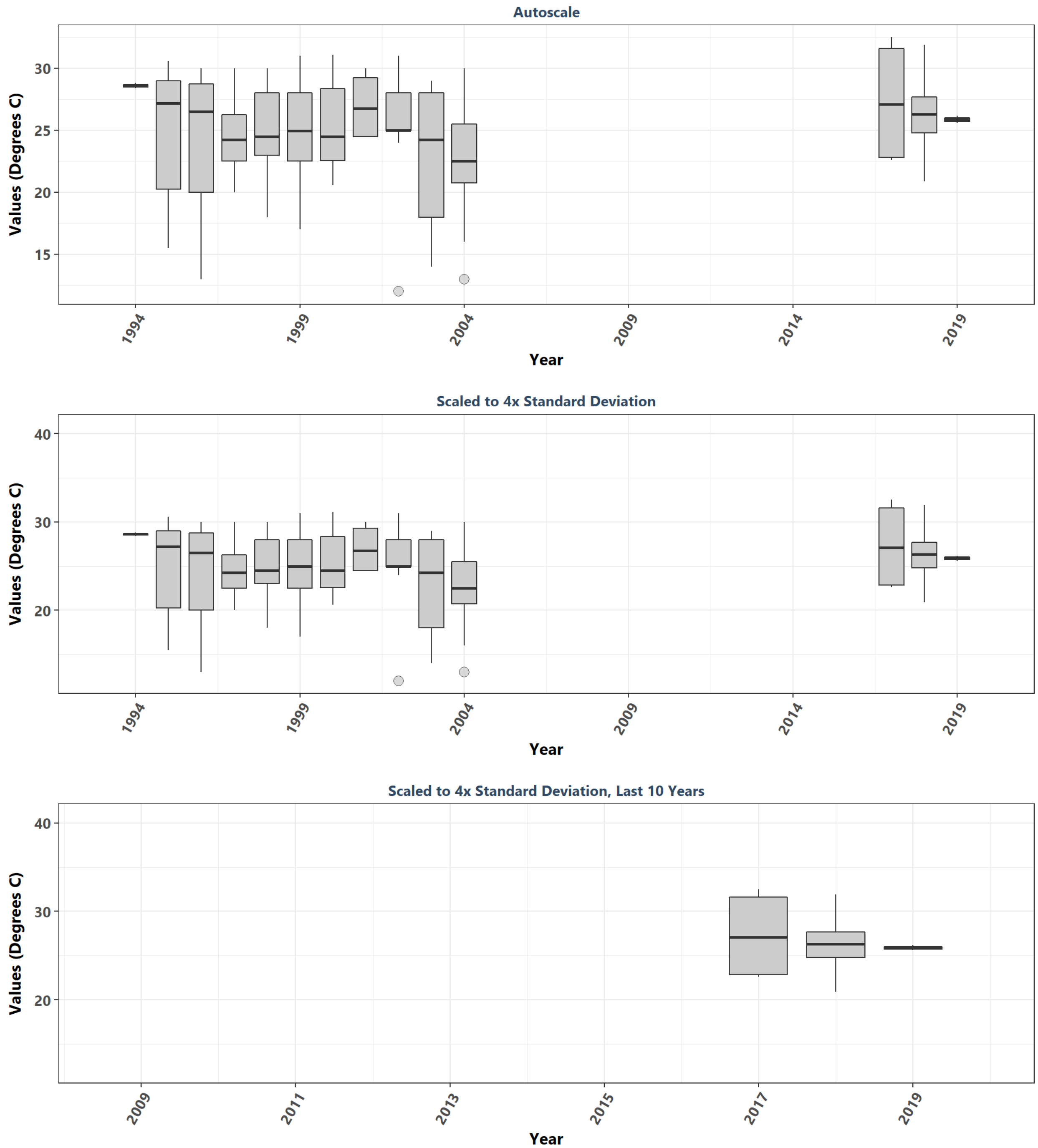



Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
2011
2013
2015
2017
2019
2021
Year

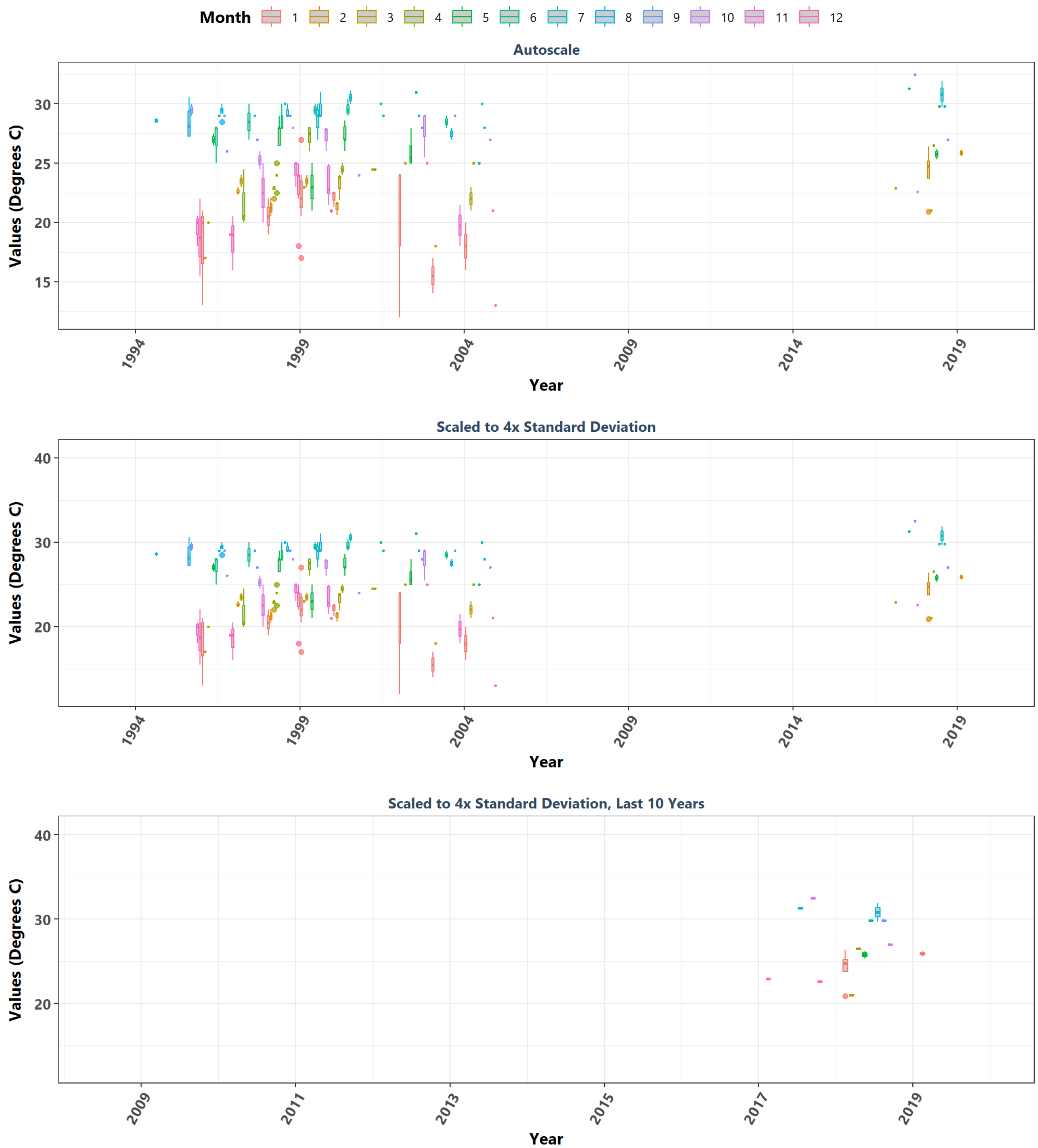

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
0
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
0
3
6
9
12
Month

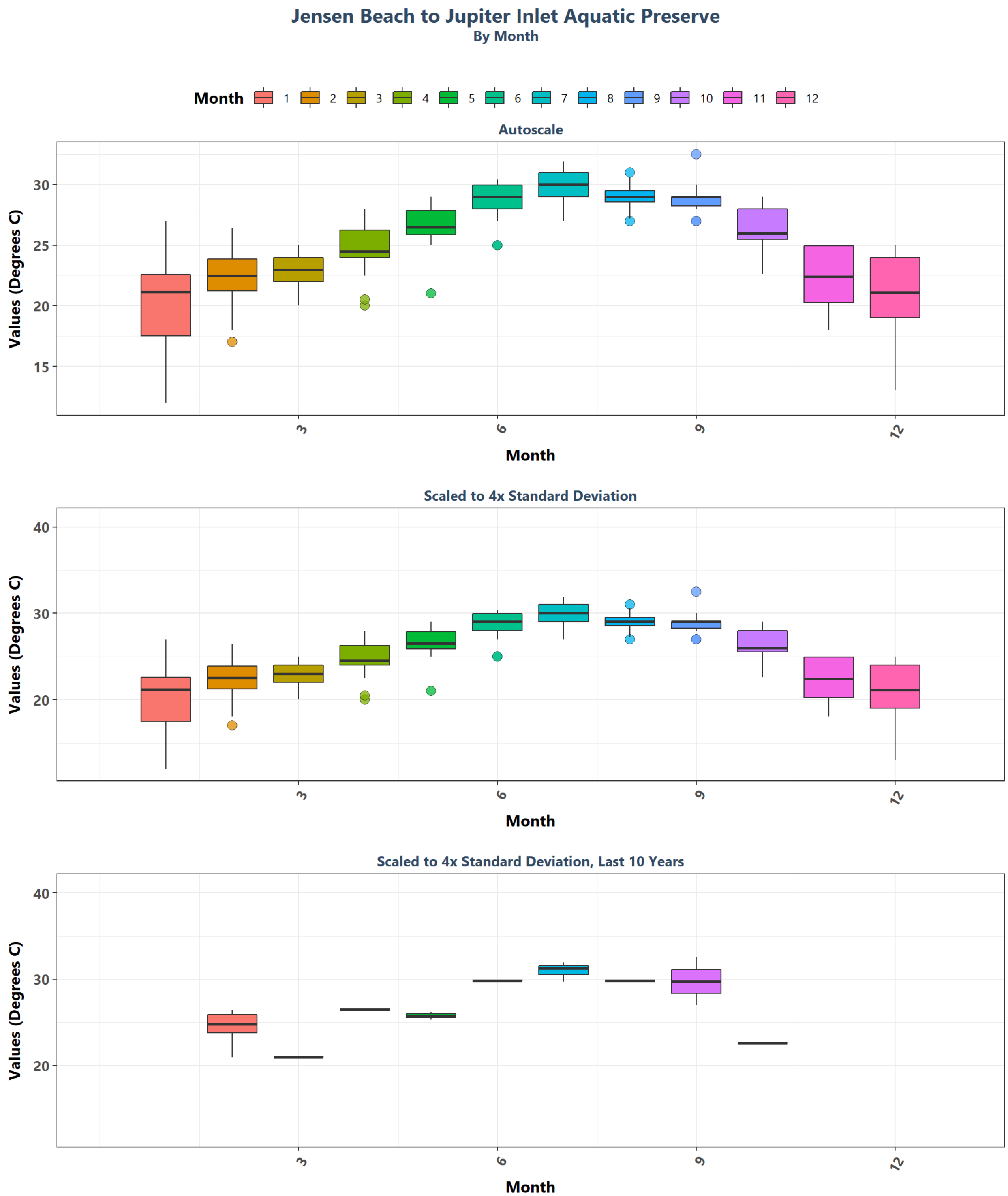
Jensen Beach to Jupiter Inlet Aquatic Preserve

By Year



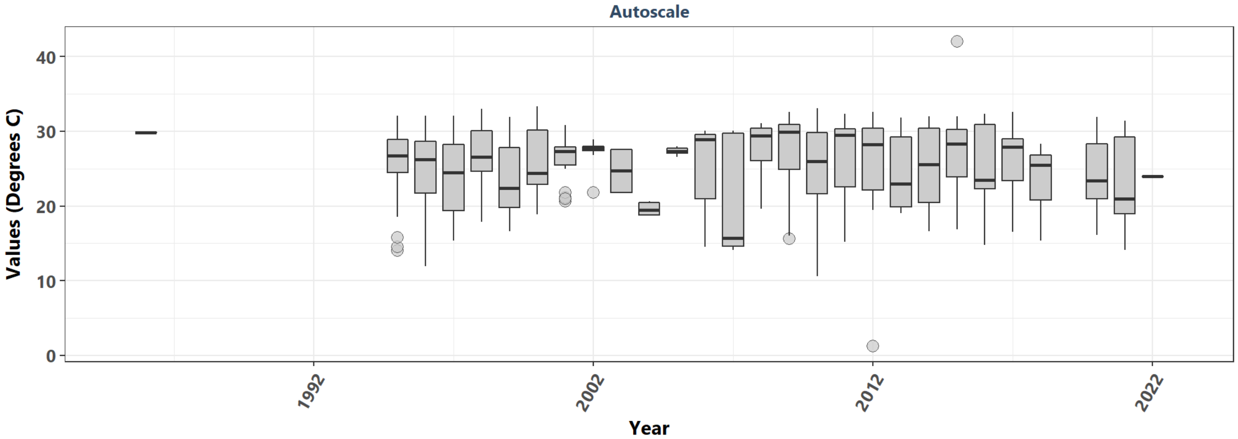
Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month

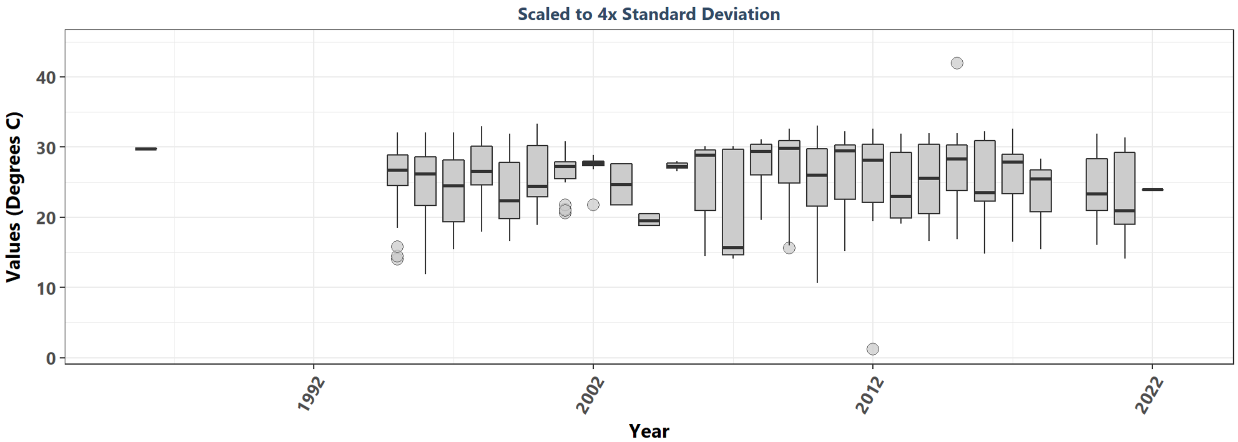


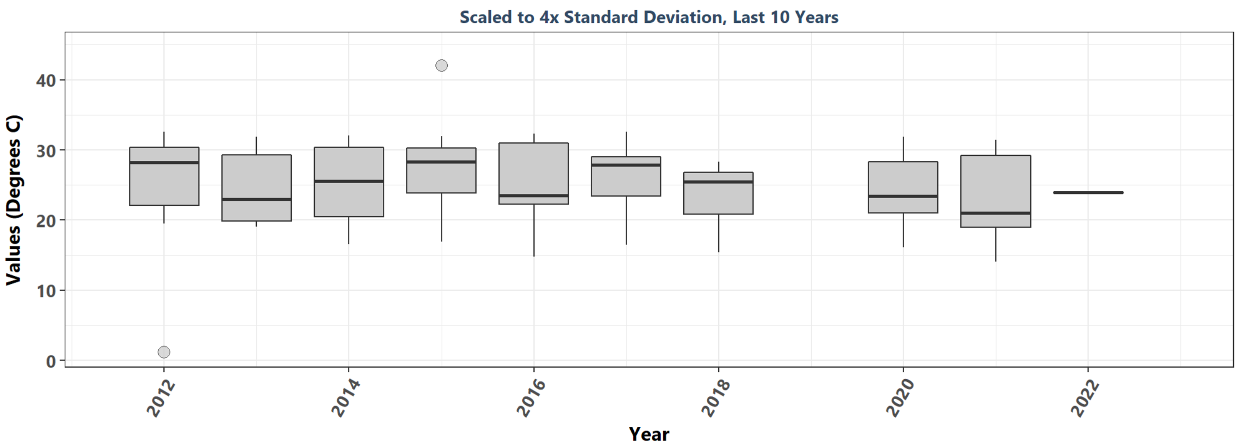


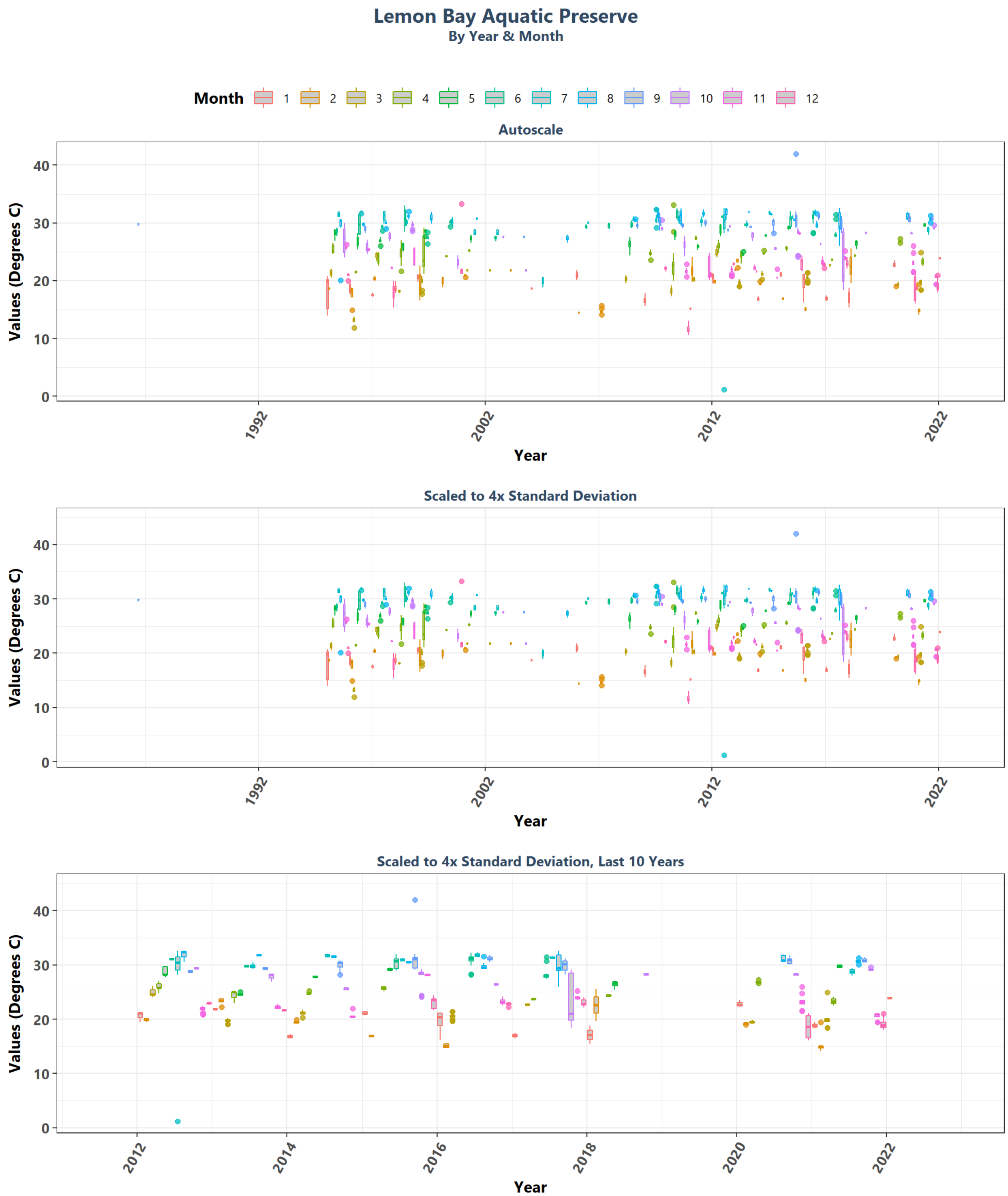
Jensen Beach to Jupiter Inlet Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
3
6
9
12
Month

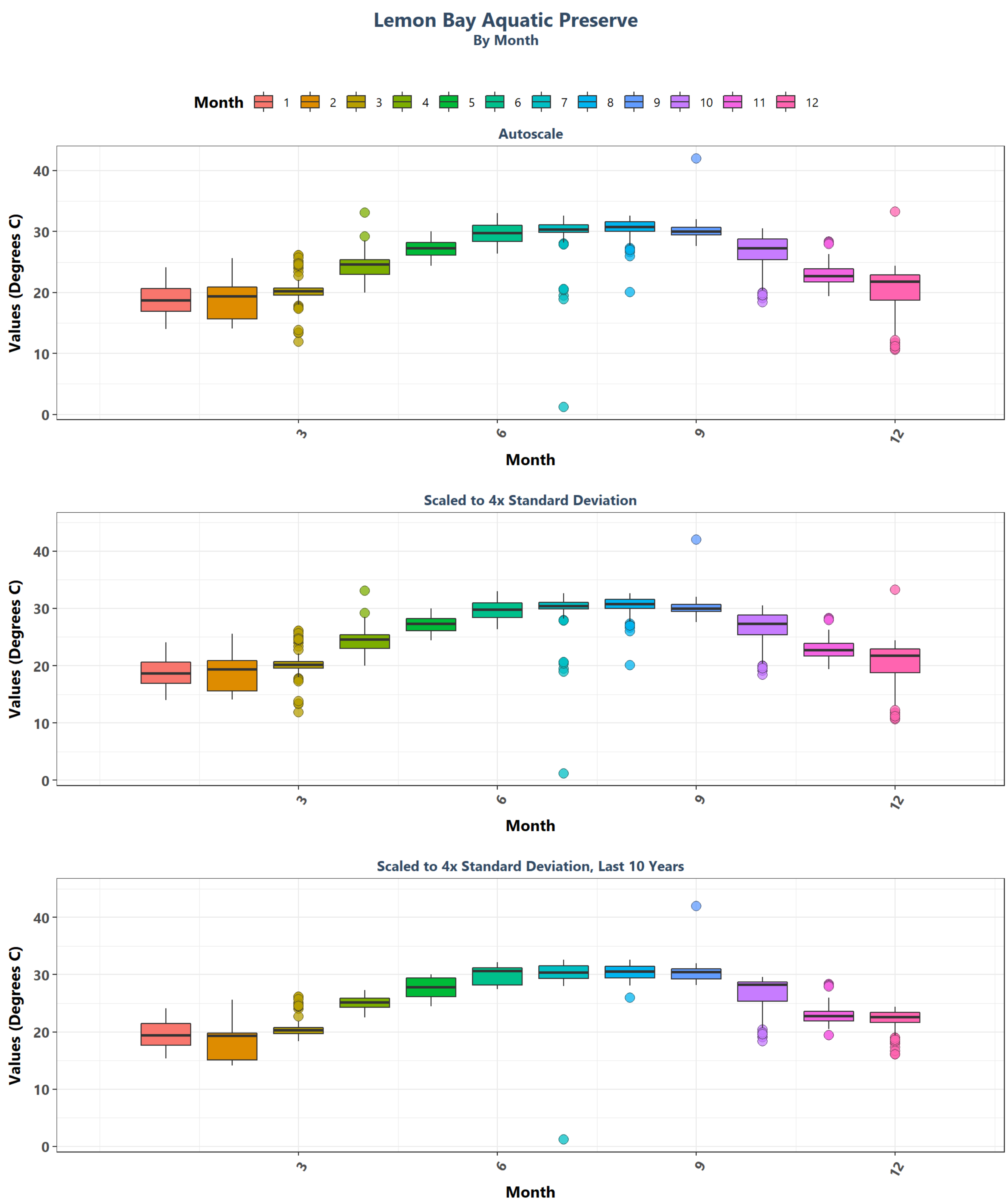
Lemon Bay Aquatic Preserve
By Year



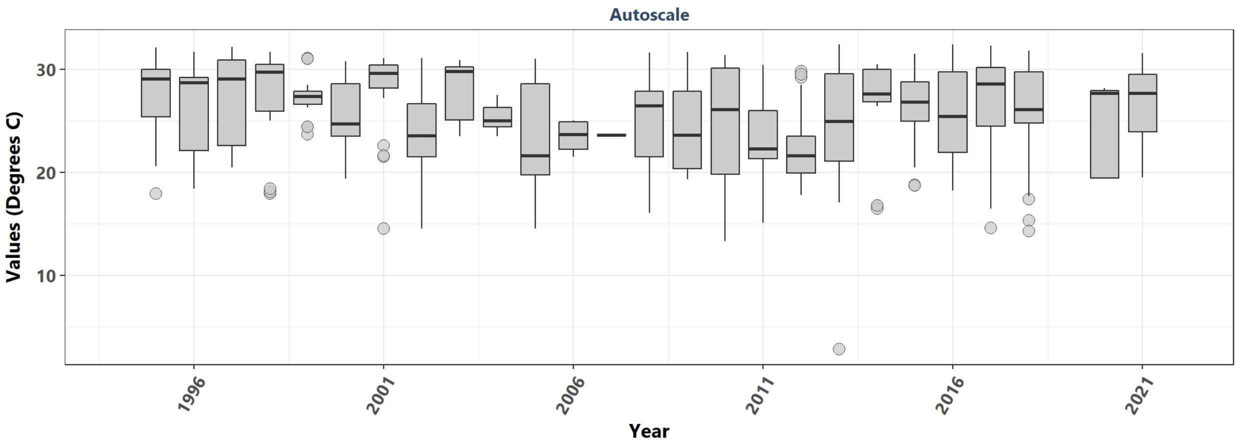


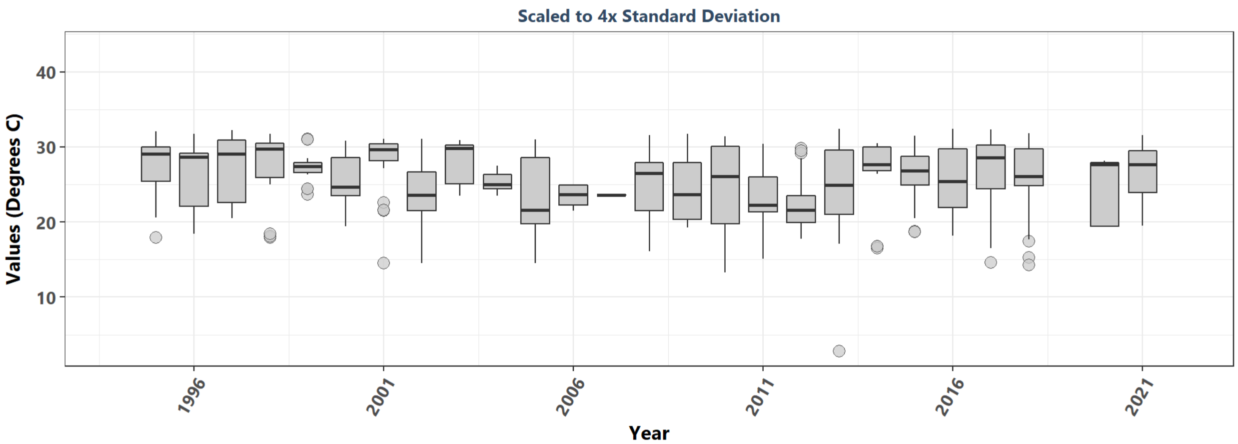


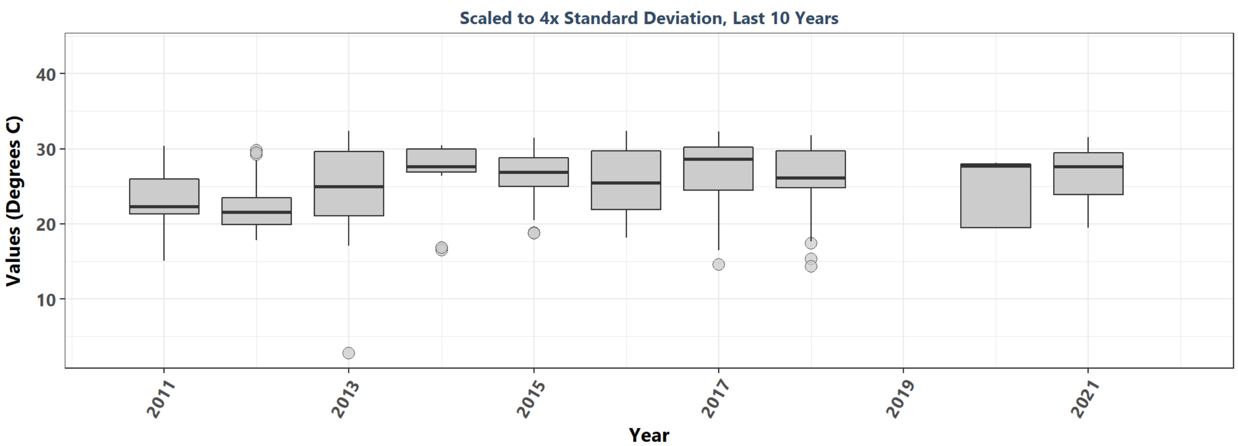

Lemon Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
40
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

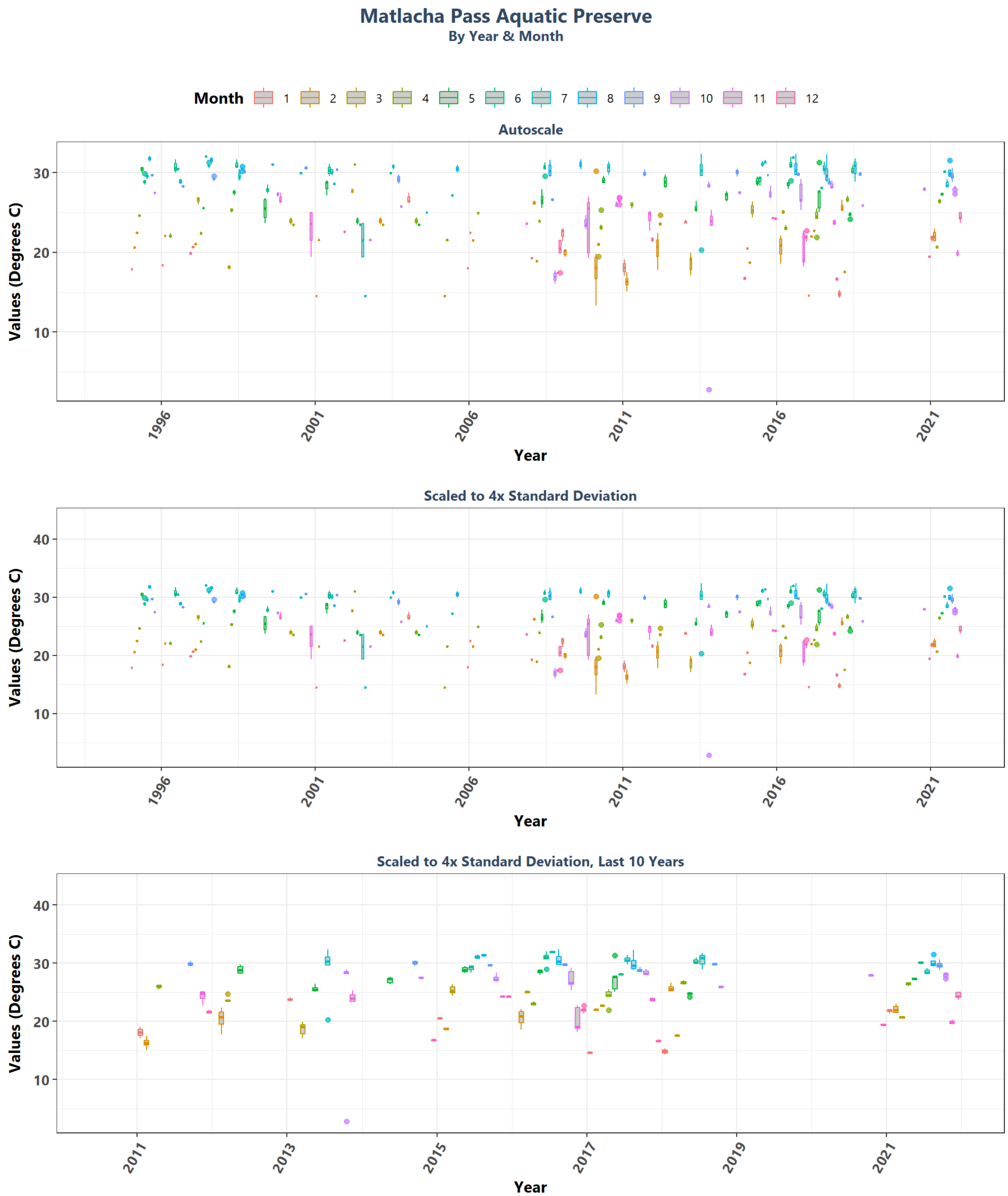

Lemon Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
3
6
9
12
Month

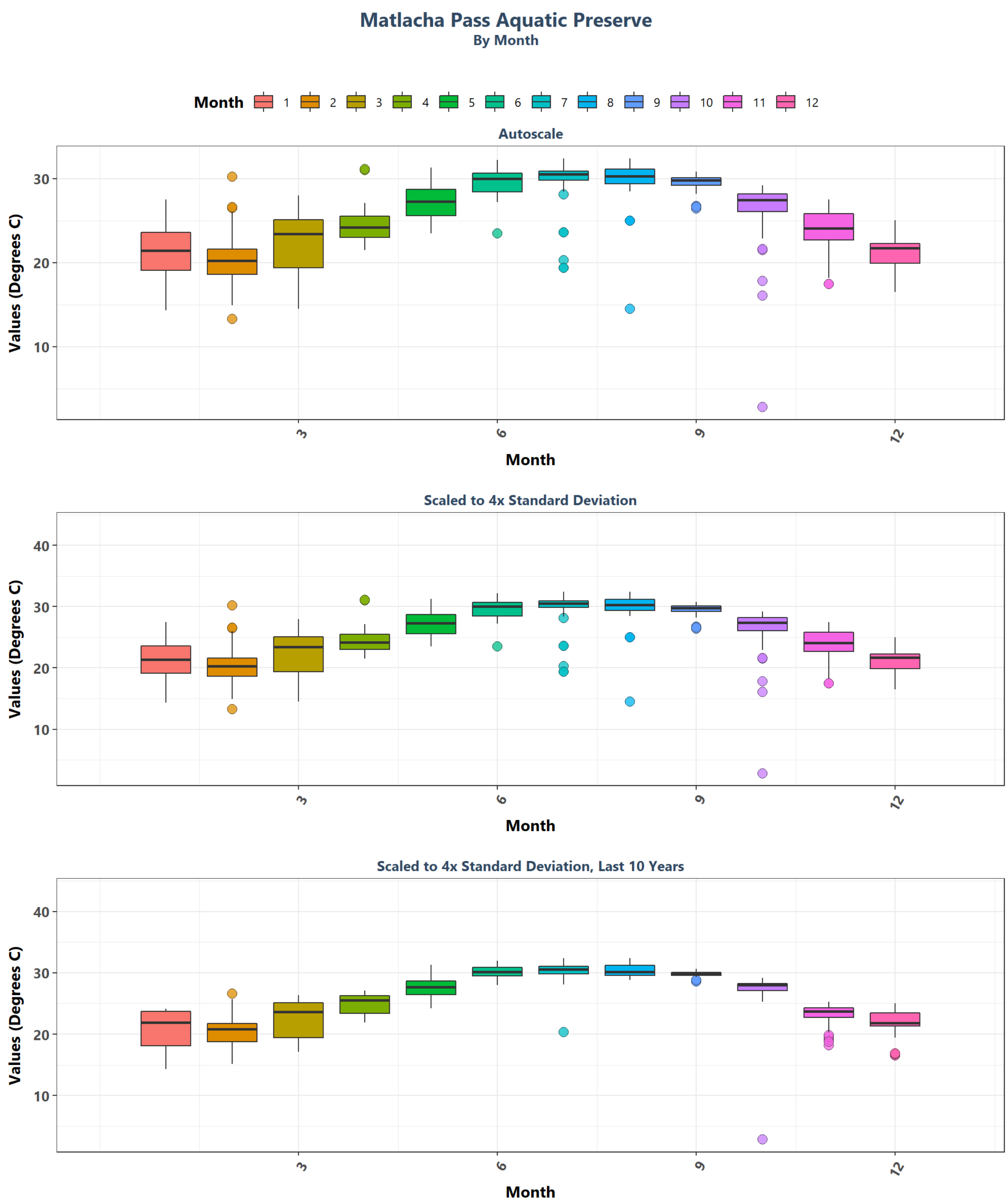
Matlacha Pass Aquatic Preserve
By Year





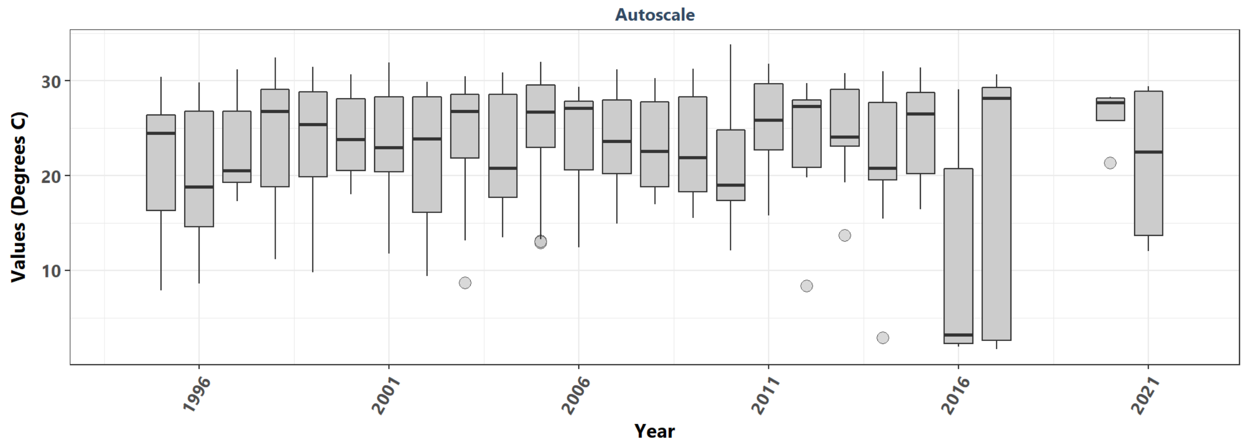


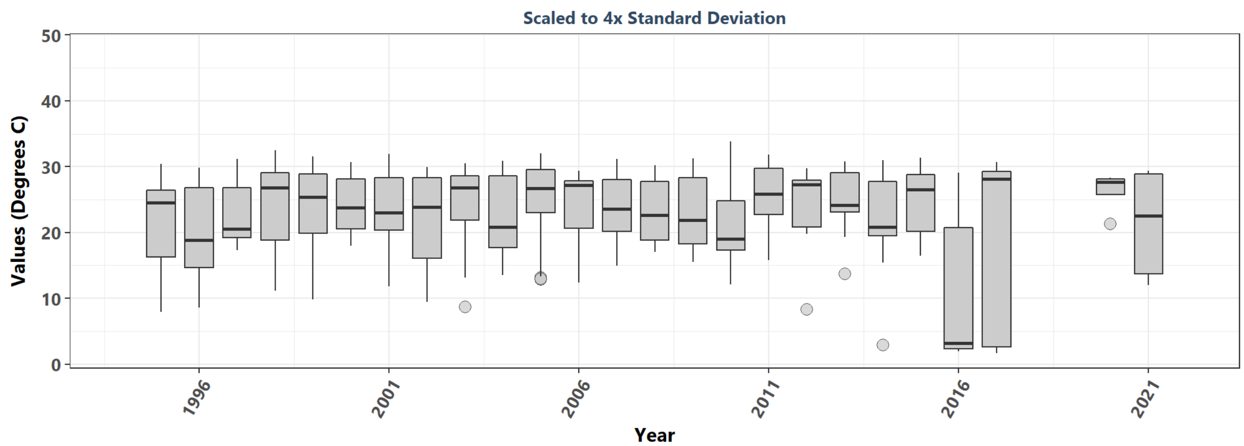

Matlacha Pass Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values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

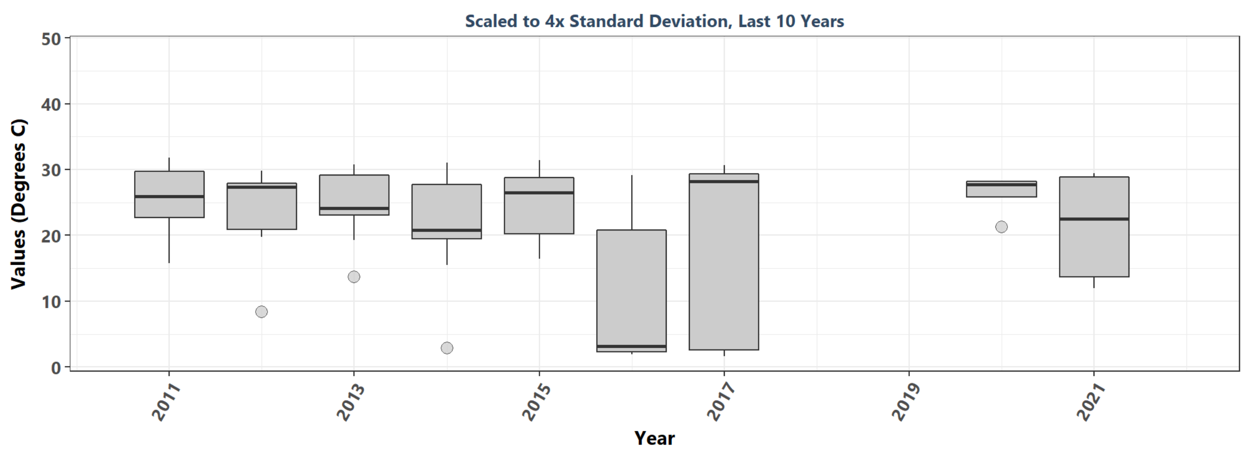

Matlacha Pass Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month
3
6
9
12
10
20
30
40

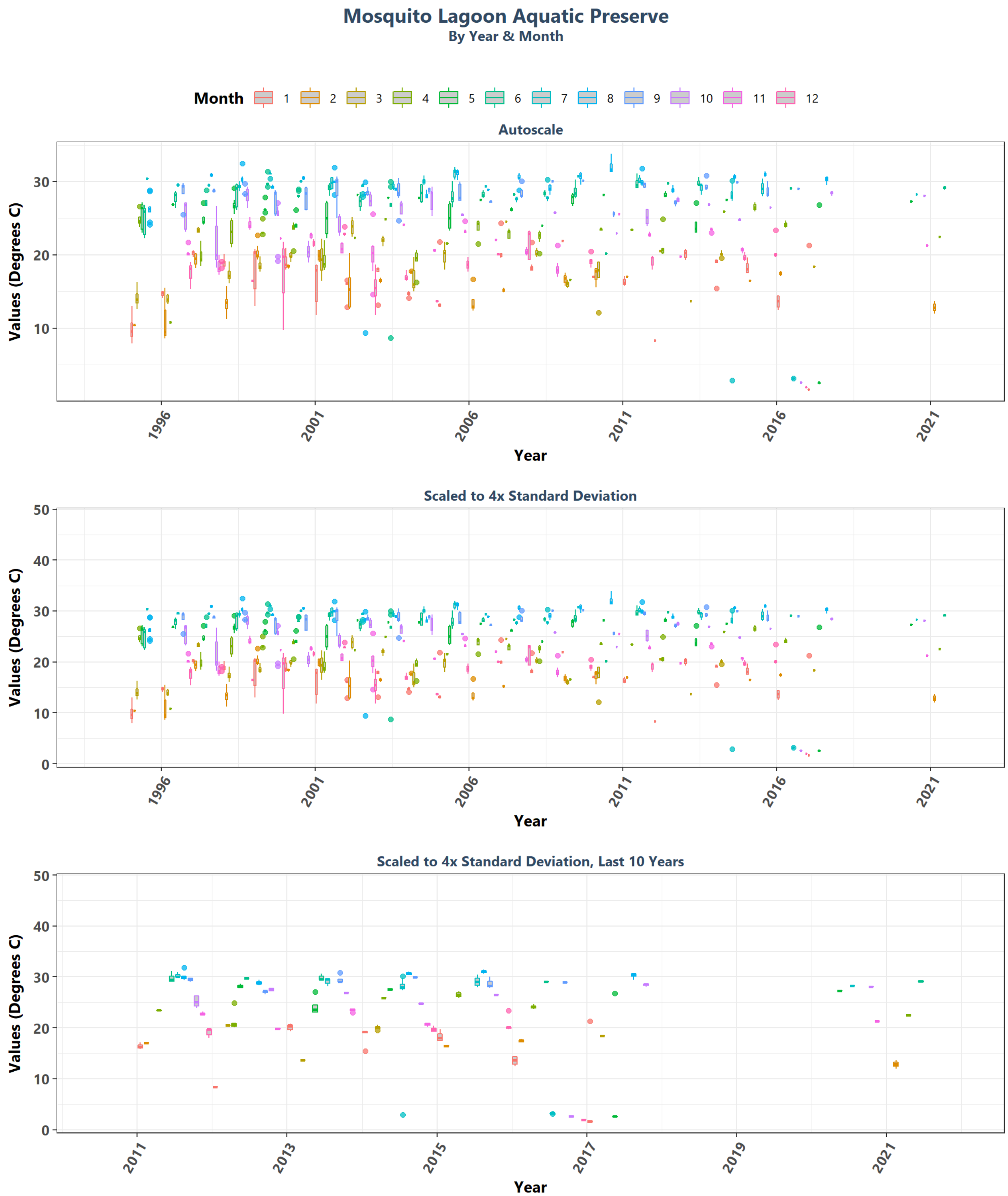
Mosquito Lagoon Aquatic Preserve

By Year





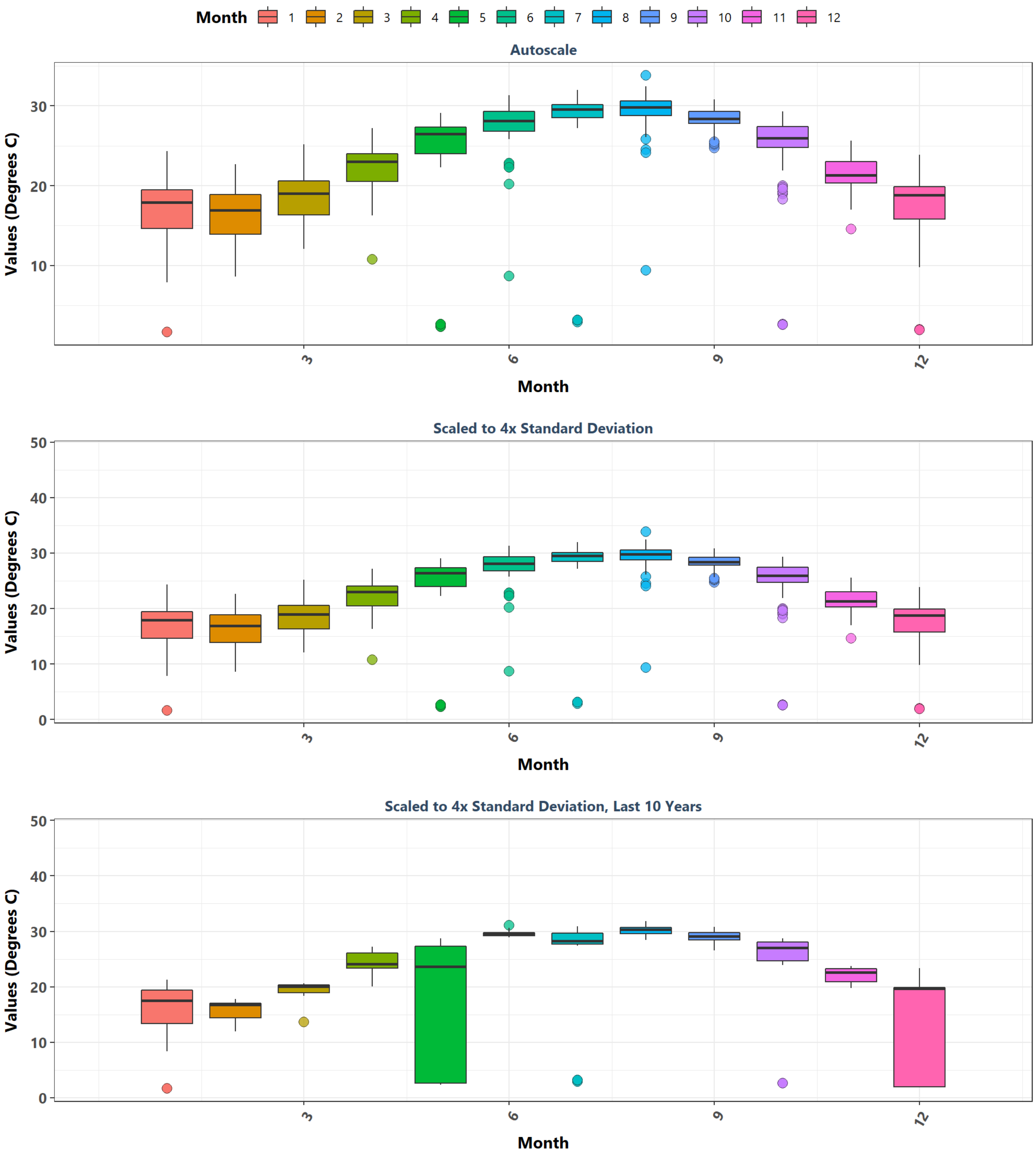




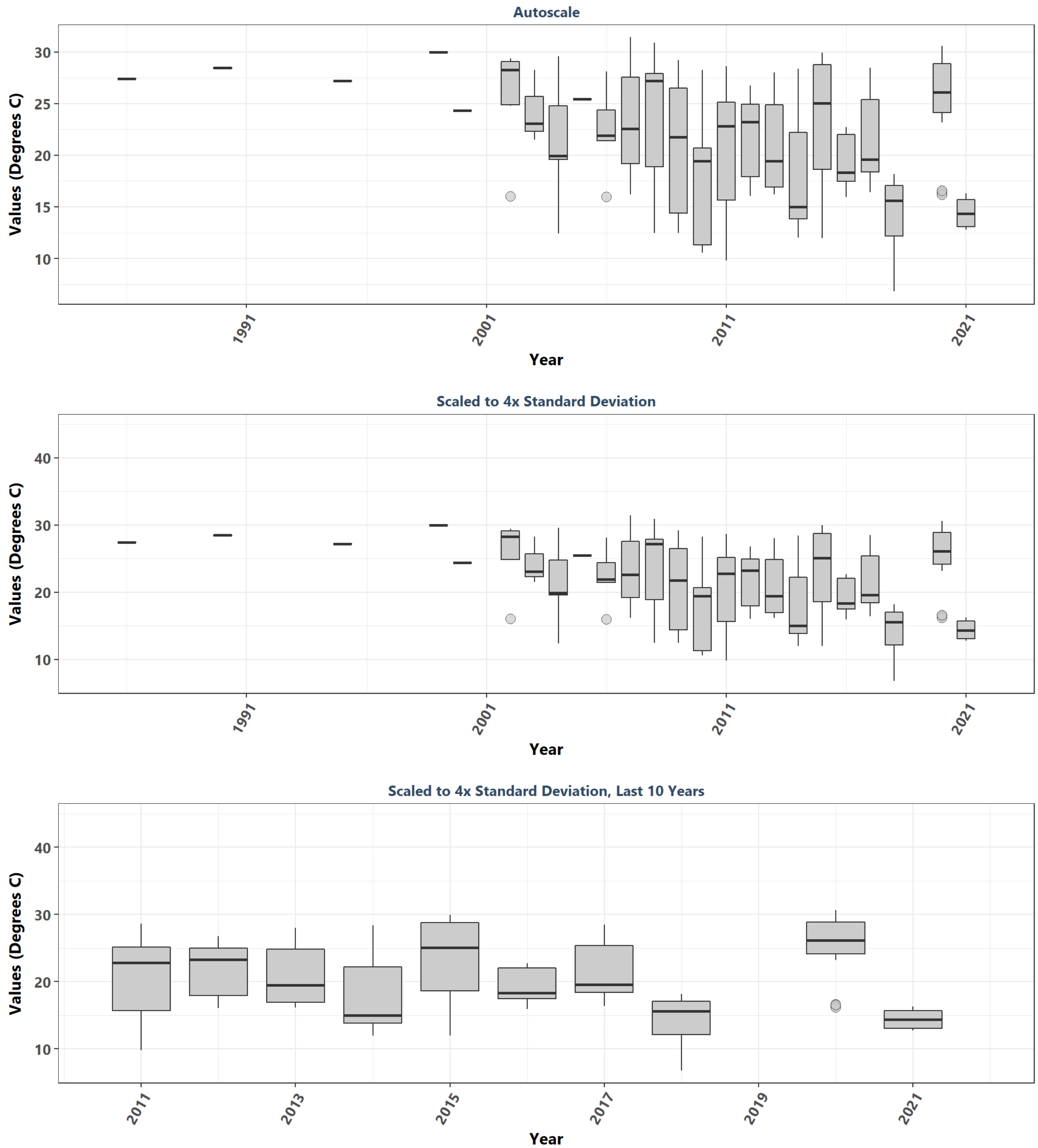
Mosquito Lagoon Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
50
40
30
20
10
0
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
50
40
30
20
10
0
2011
2013
2015
2017
2019
2021
Year

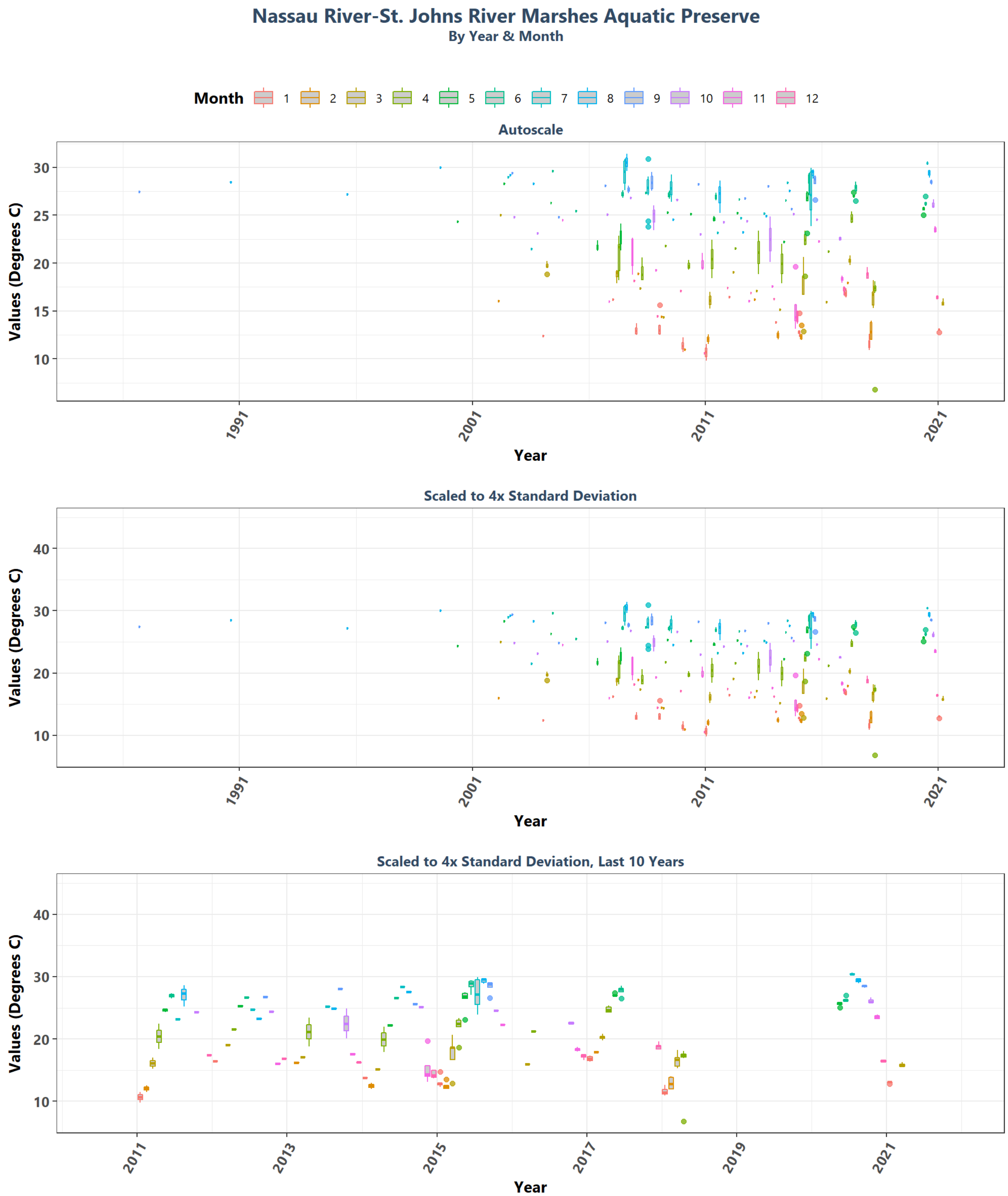
Mosquito Lagoon Aquatic Preserve

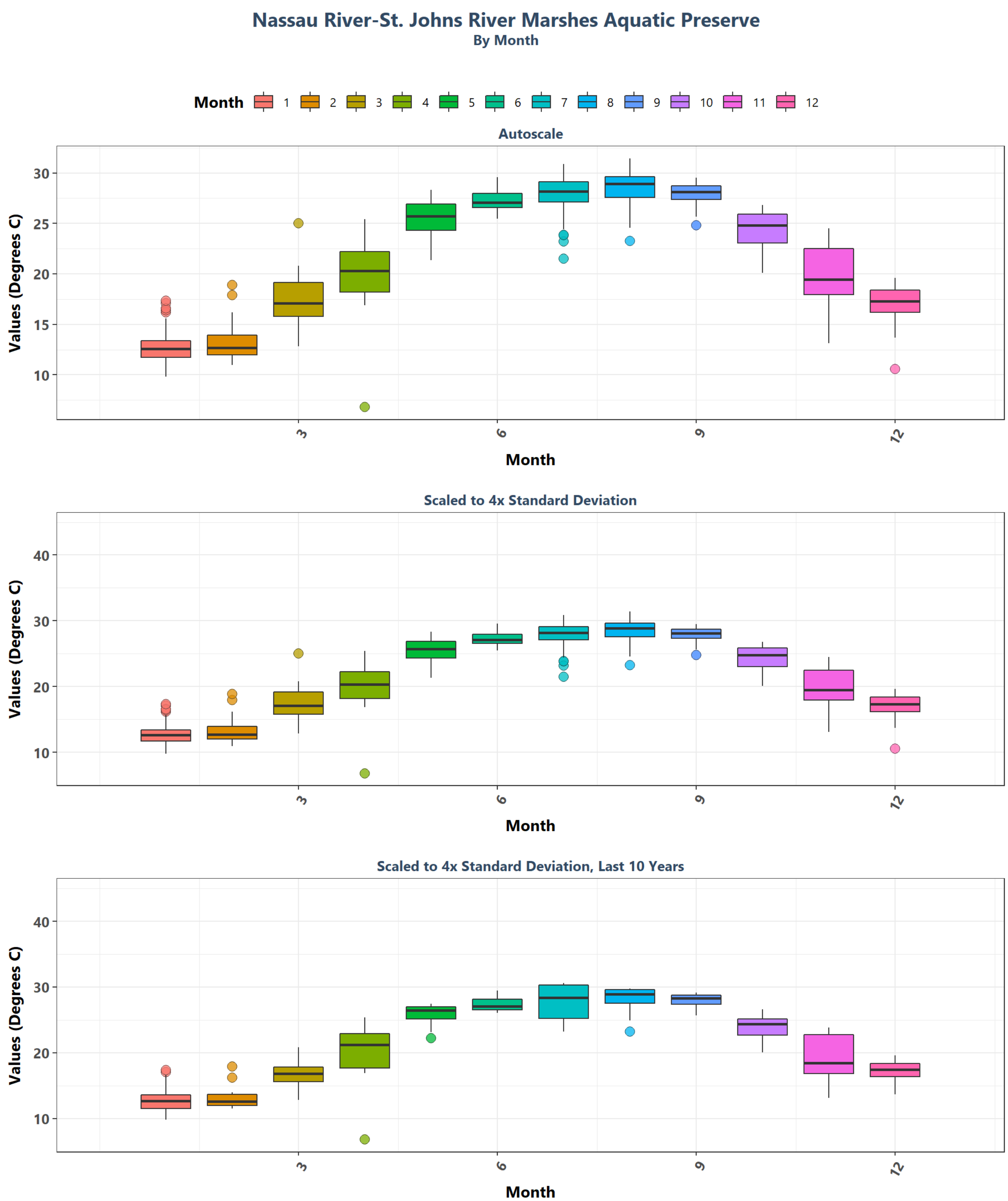
By Month



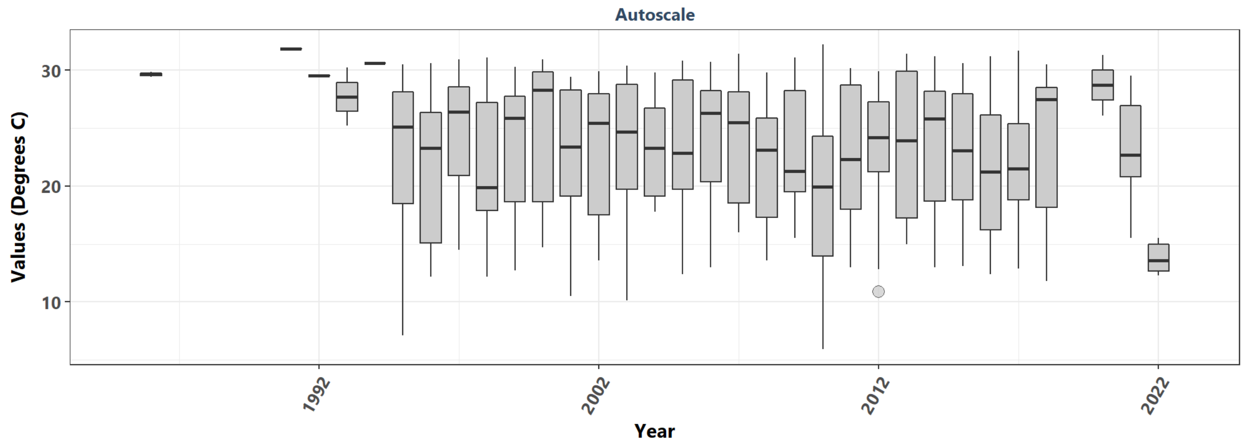
Nassau River-St. Johns River Marshes Aquatic Preserve
By Year

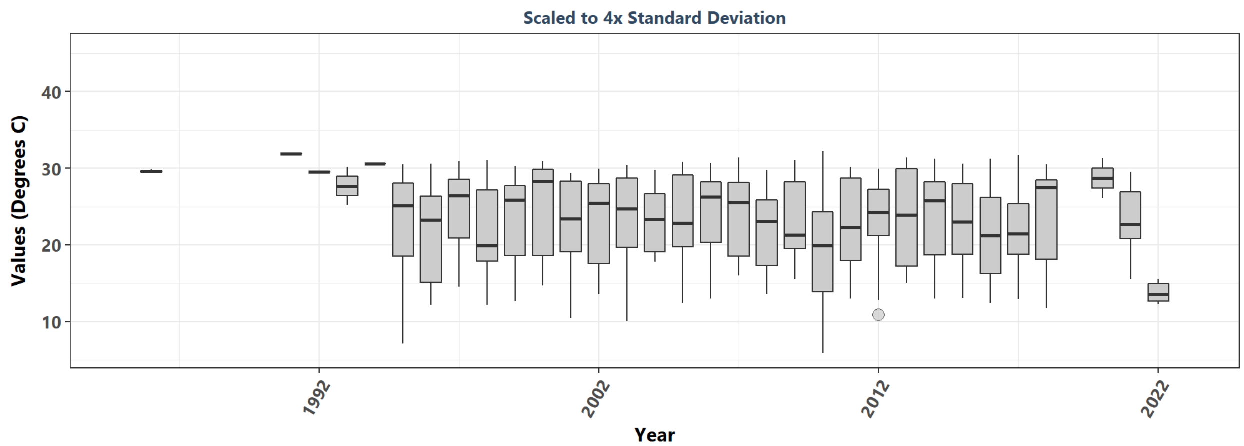


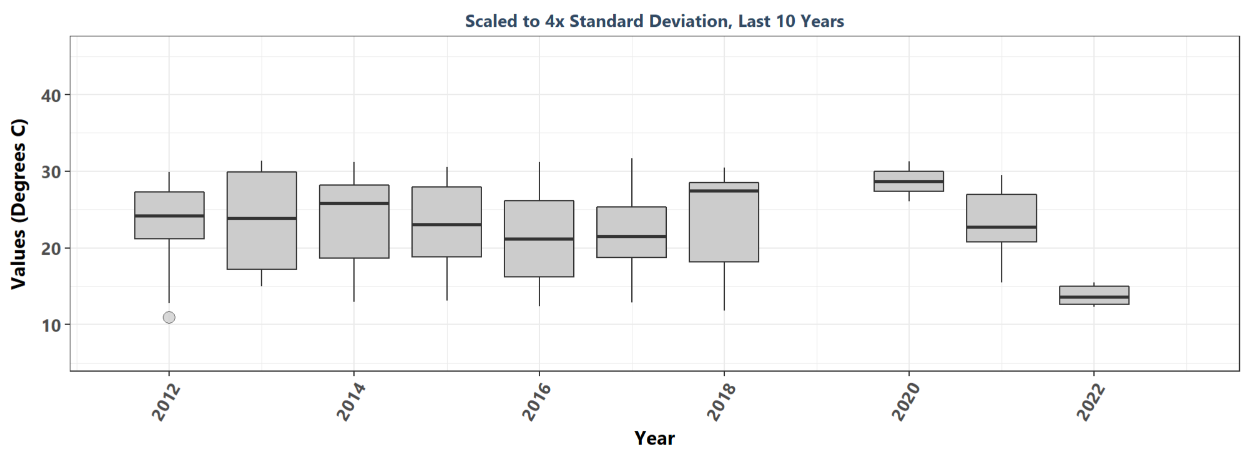

Nassau River-St. Johns River Marshes Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2011
2013
2015
2017
2019
2021
Year

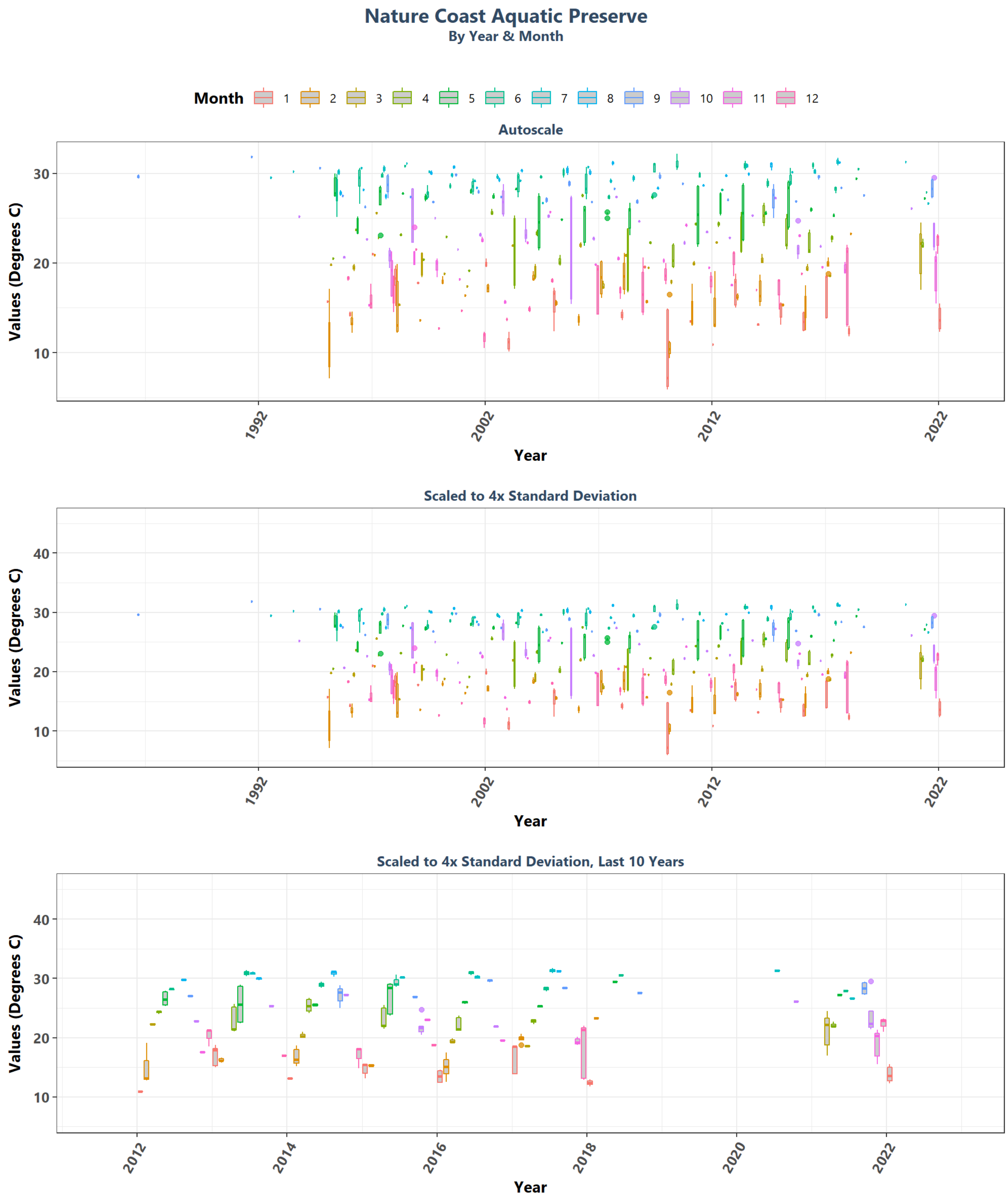

Nassau River-St. Johns River Marshes Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

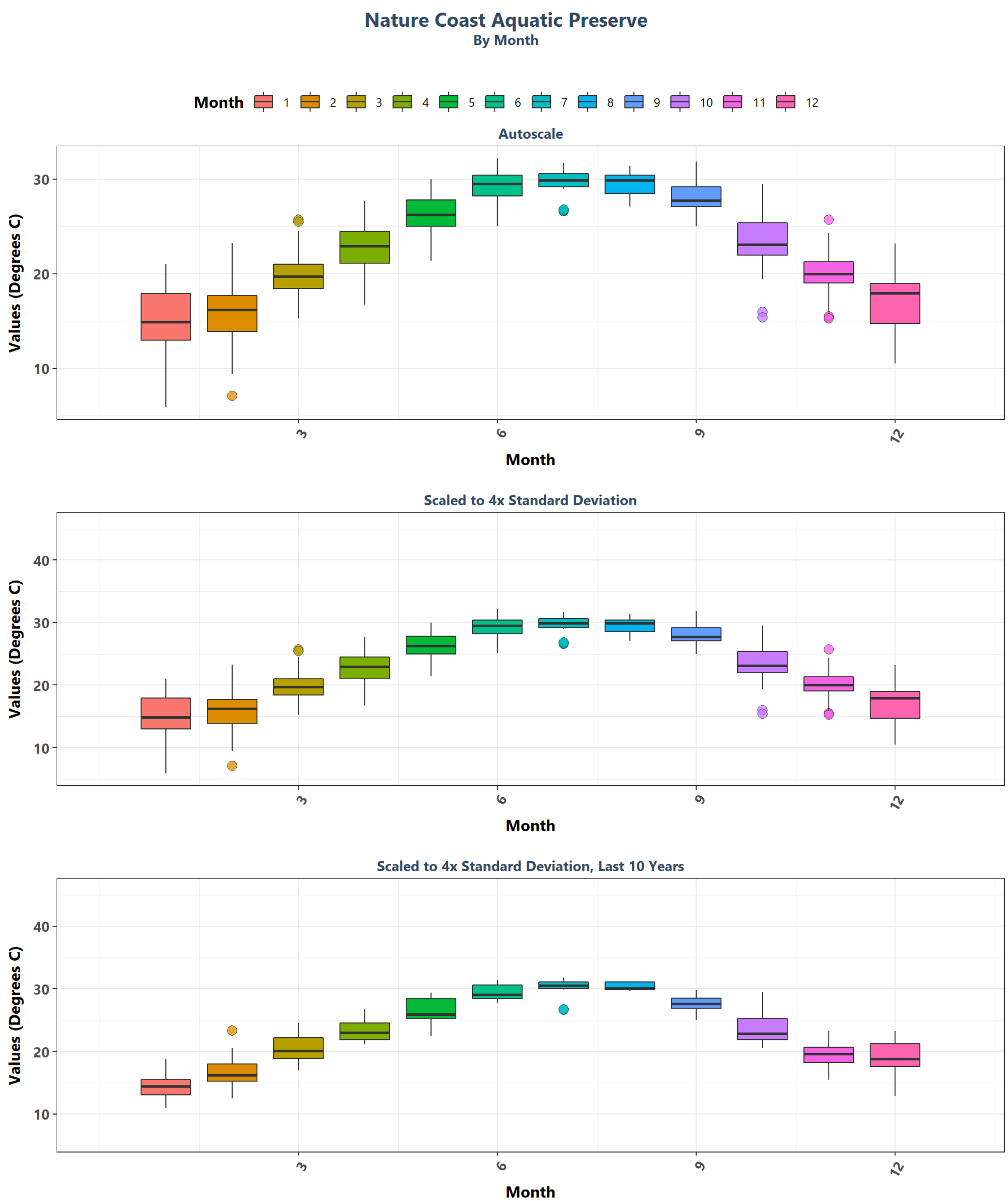
Nature Coast Aquatic Preserve
By Year





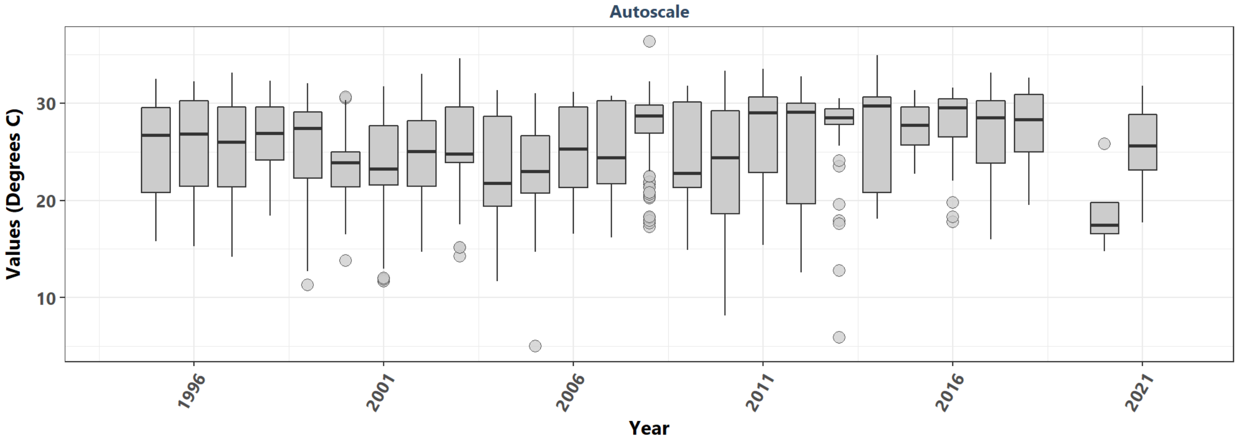


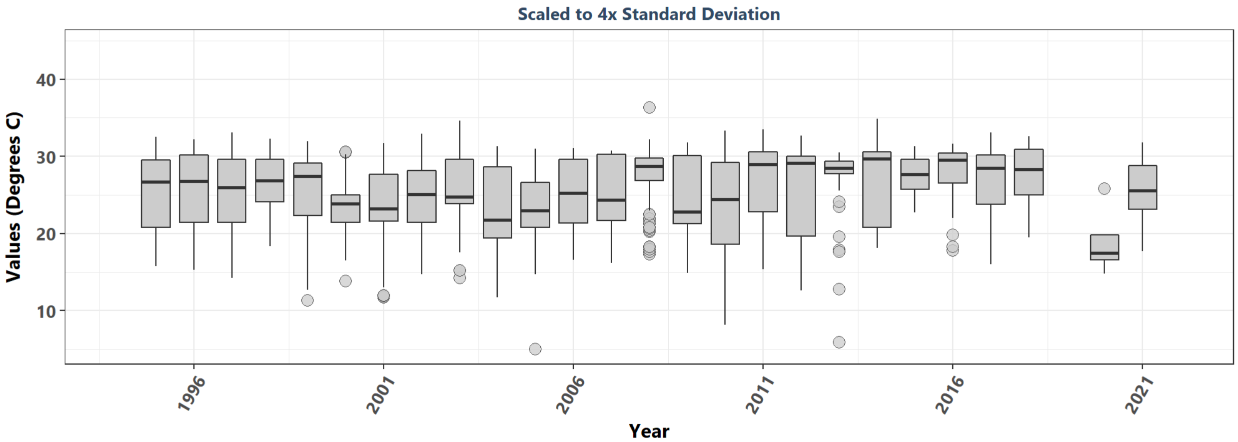

Nature Coast Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

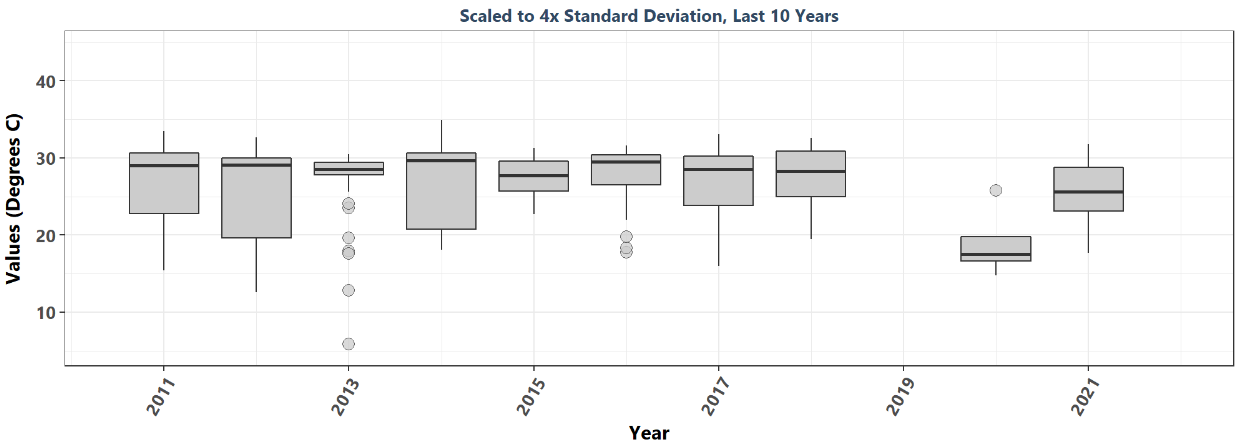


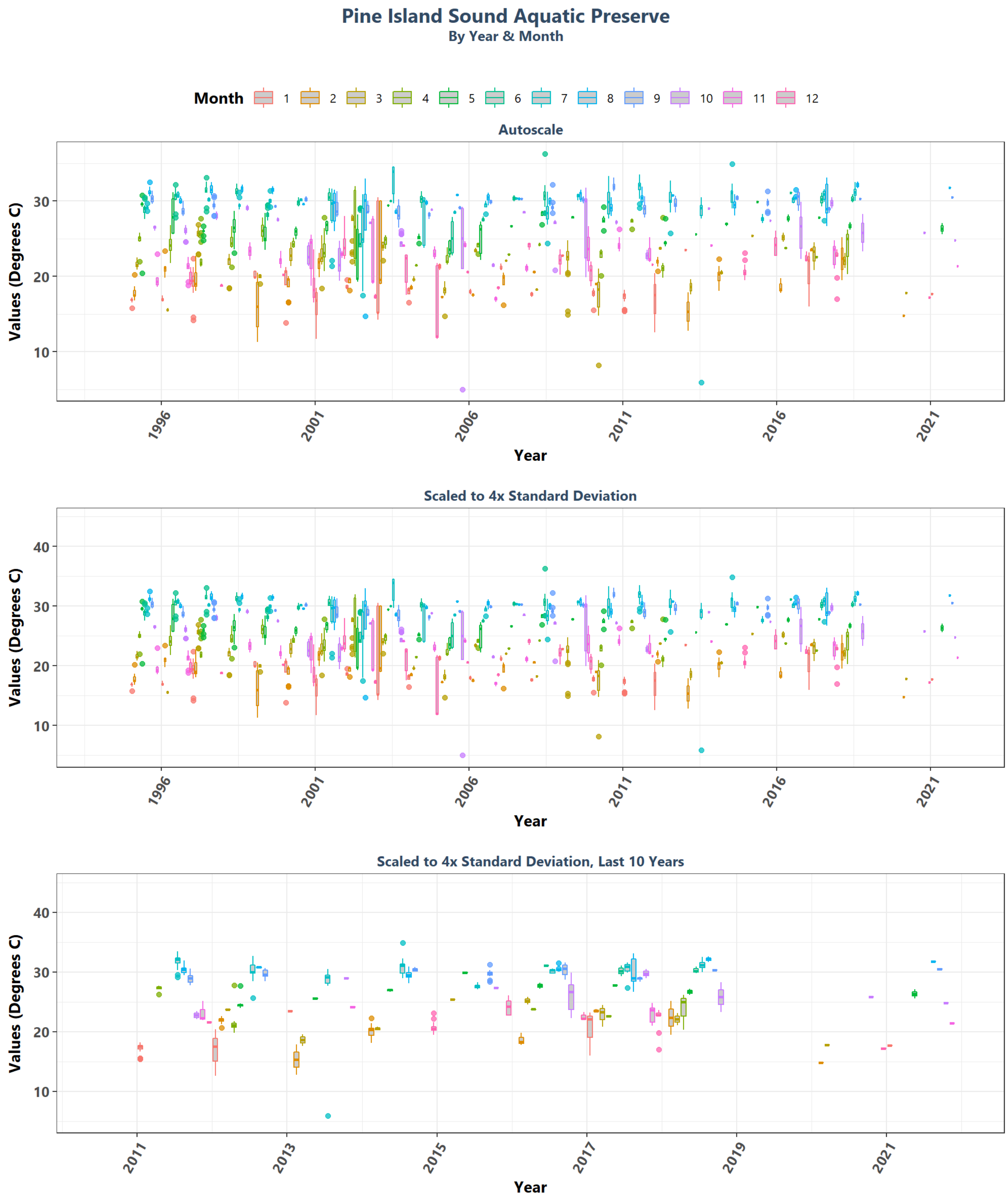
Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

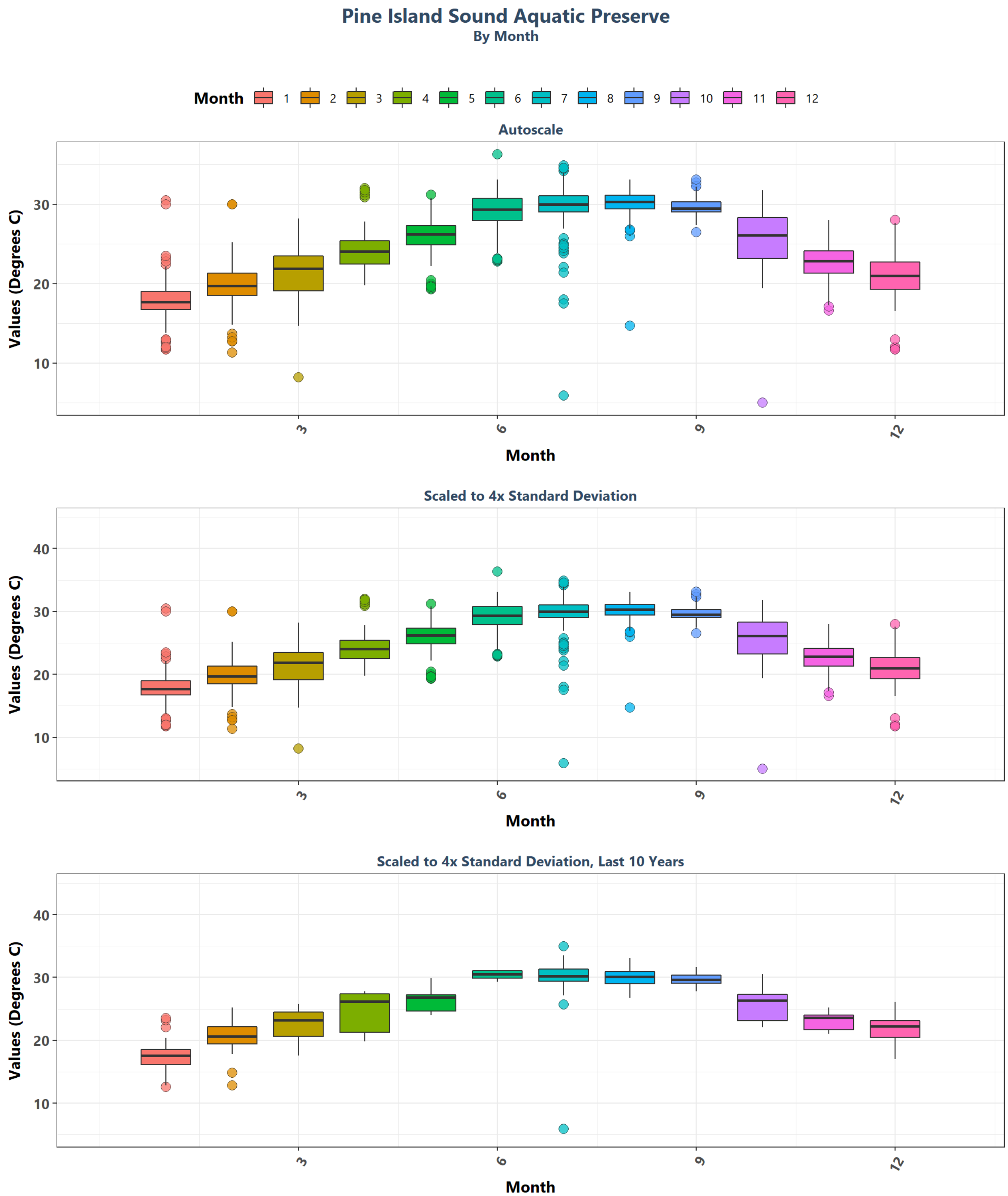
Pine Island Sound Aquatic Preserve
By Year





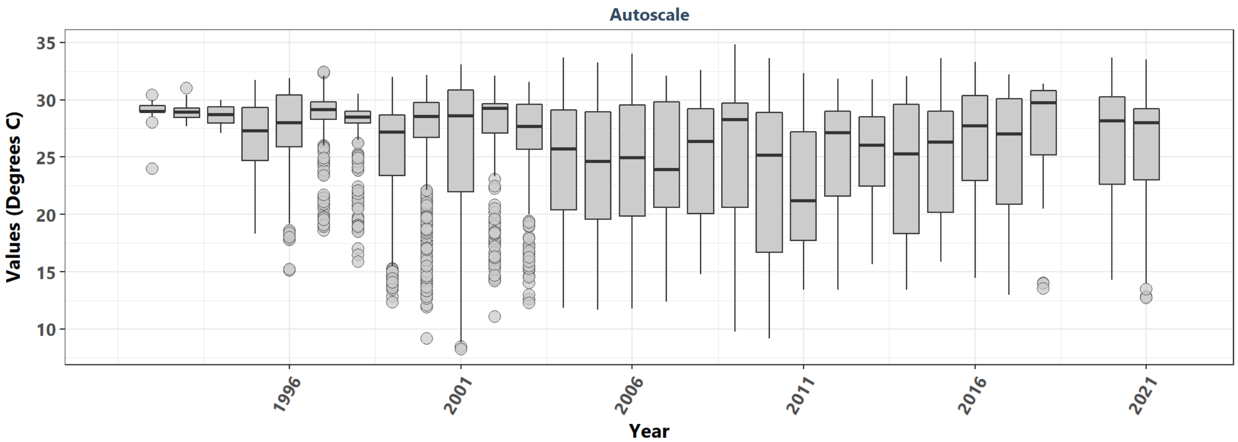


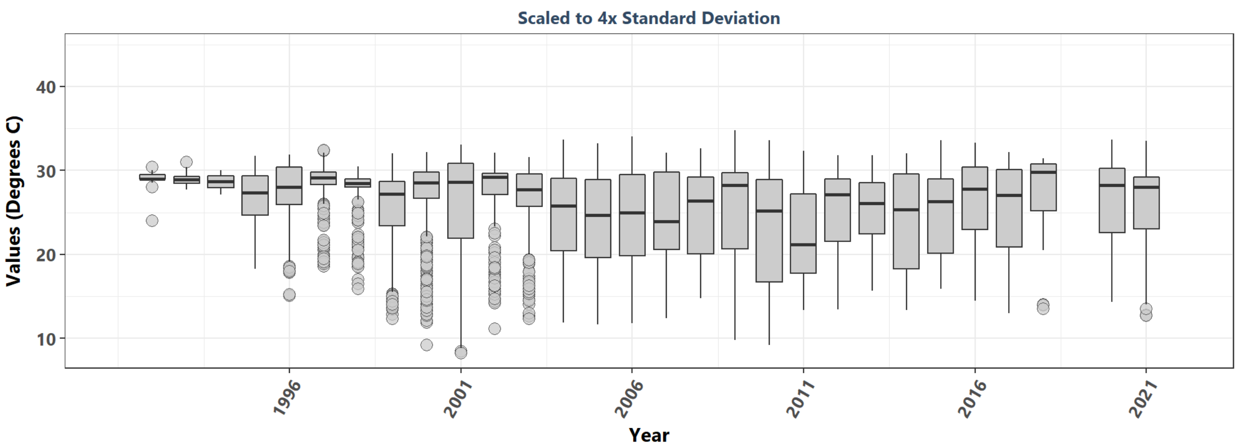

Pine Island Sound Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years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

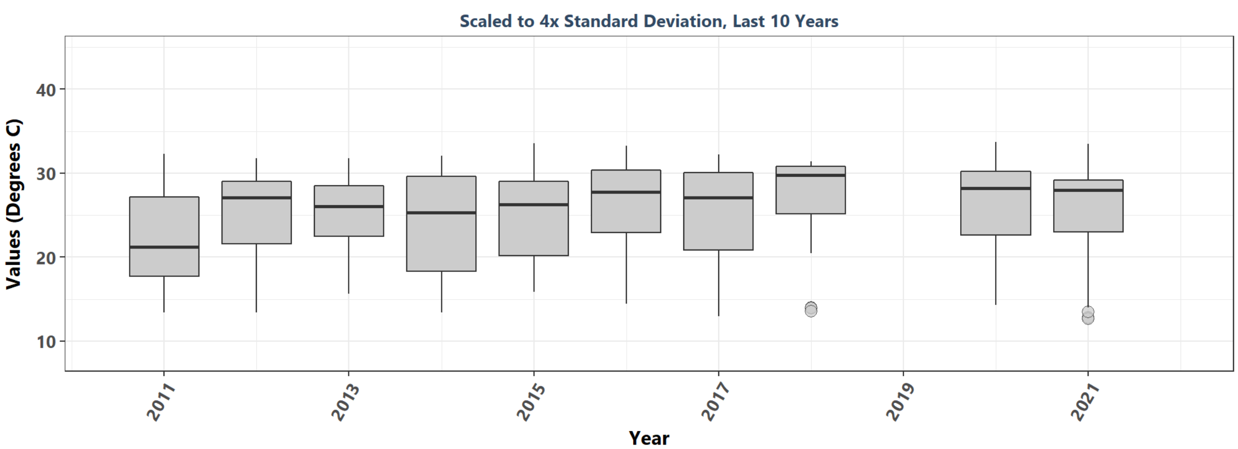


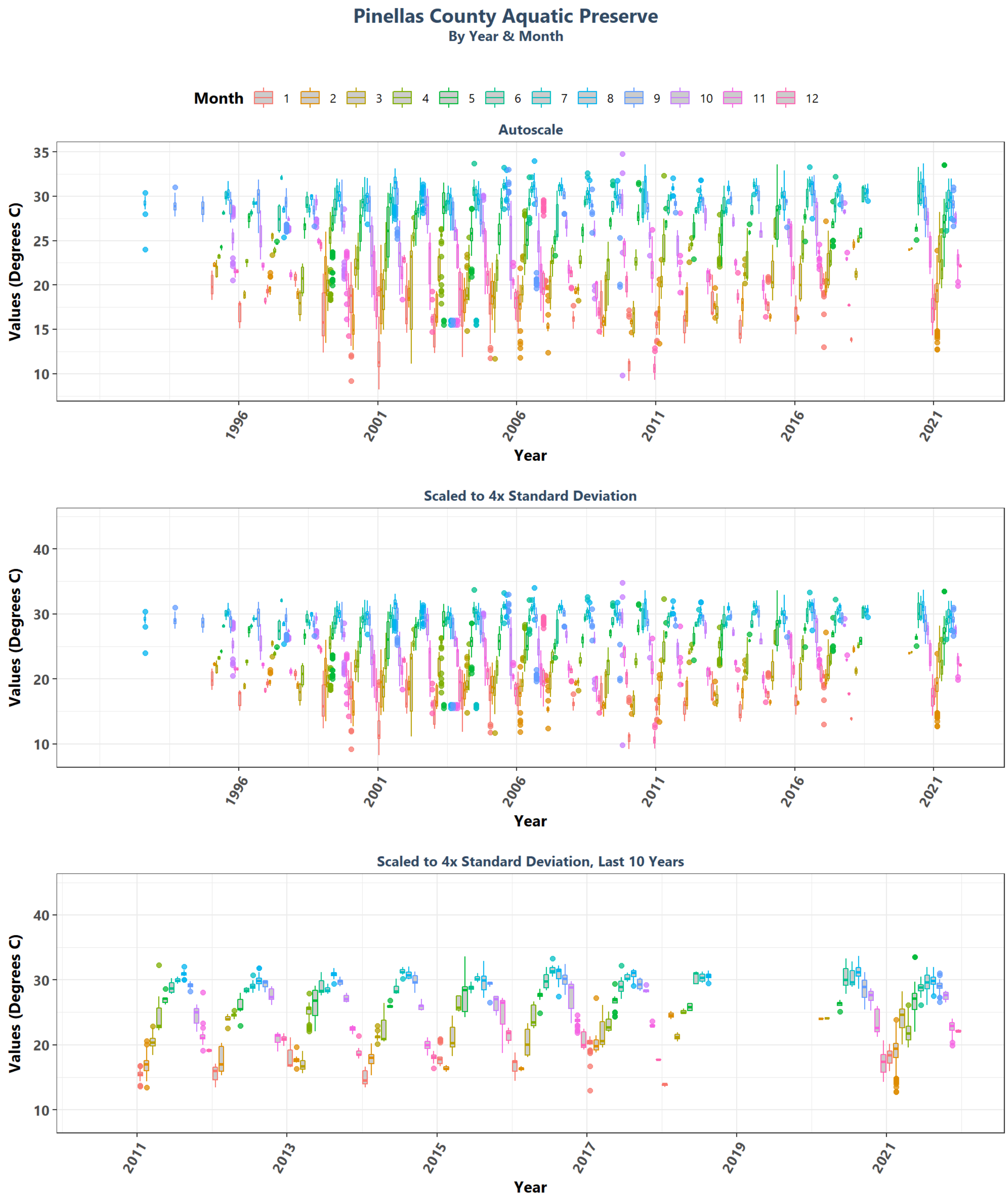
Pine Island Sound Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

Pinellas County Aquatic Preserve
By Year

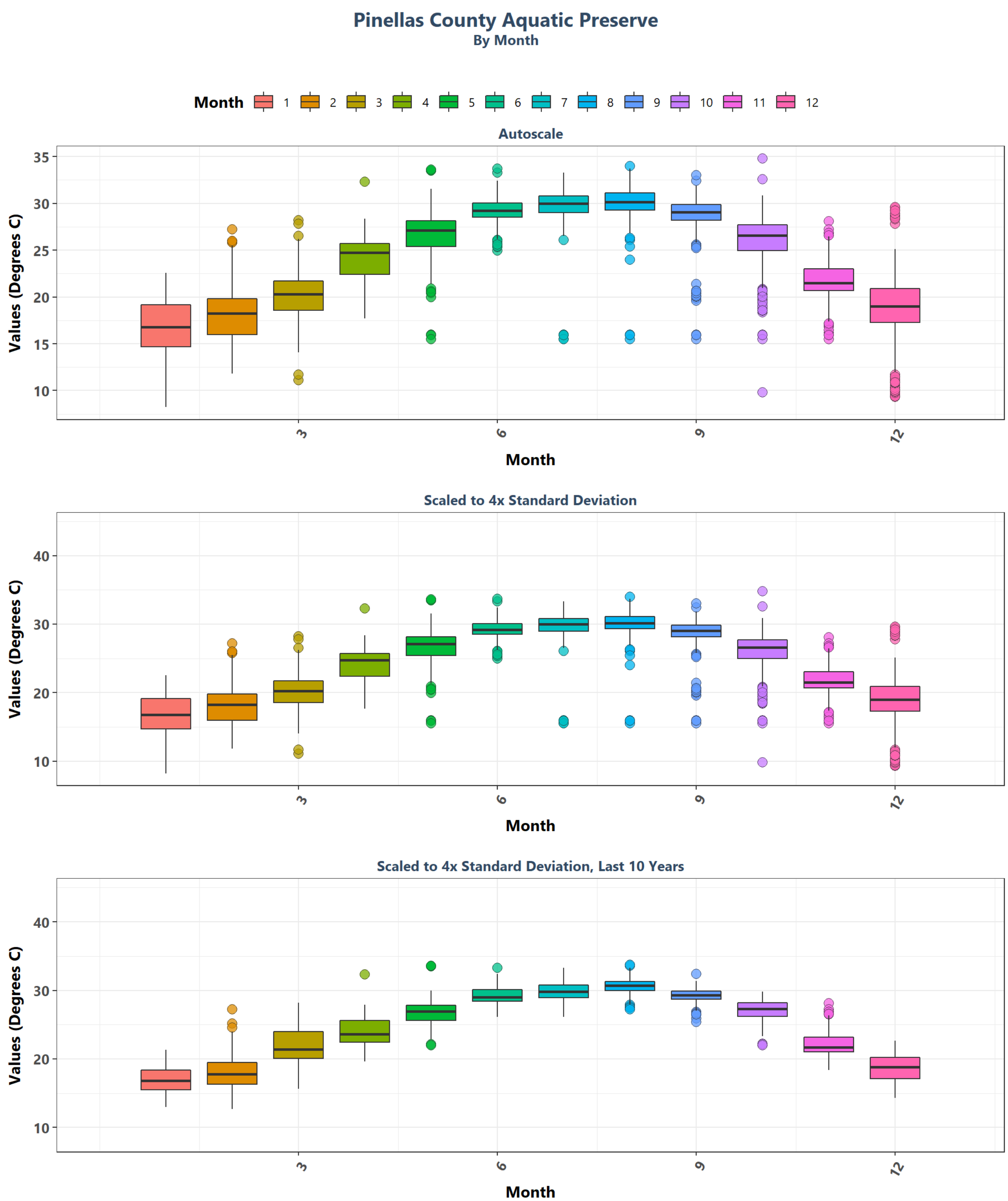






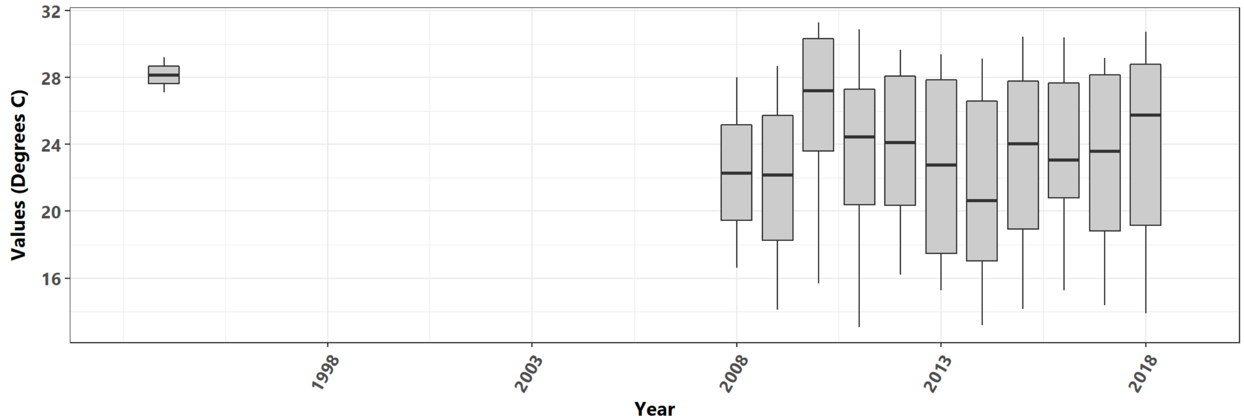


Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2011
2013
2015
2017
2019
2021
Year

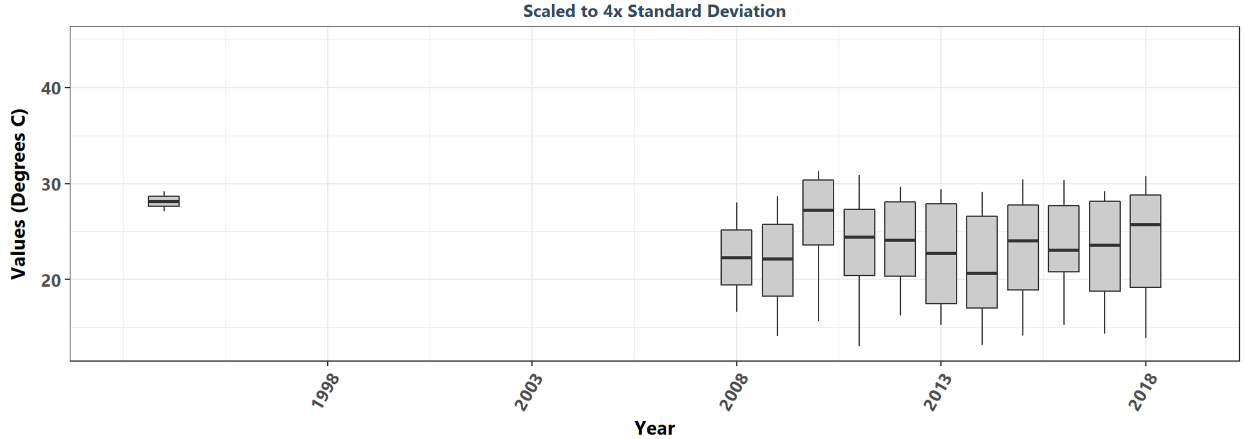

Pinellas County Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

Rocky Bayou State Park Aquatic Preserve
By Year

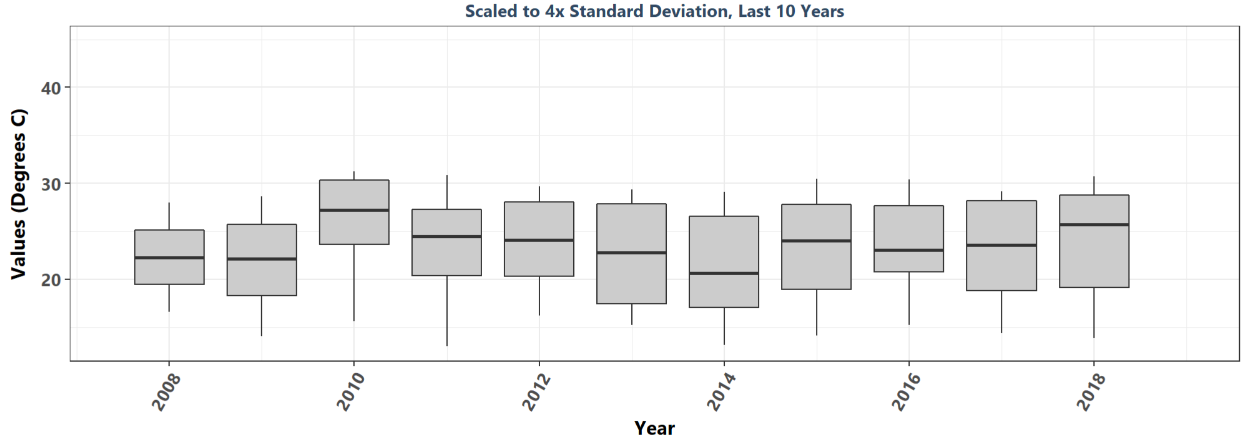
Autoscale



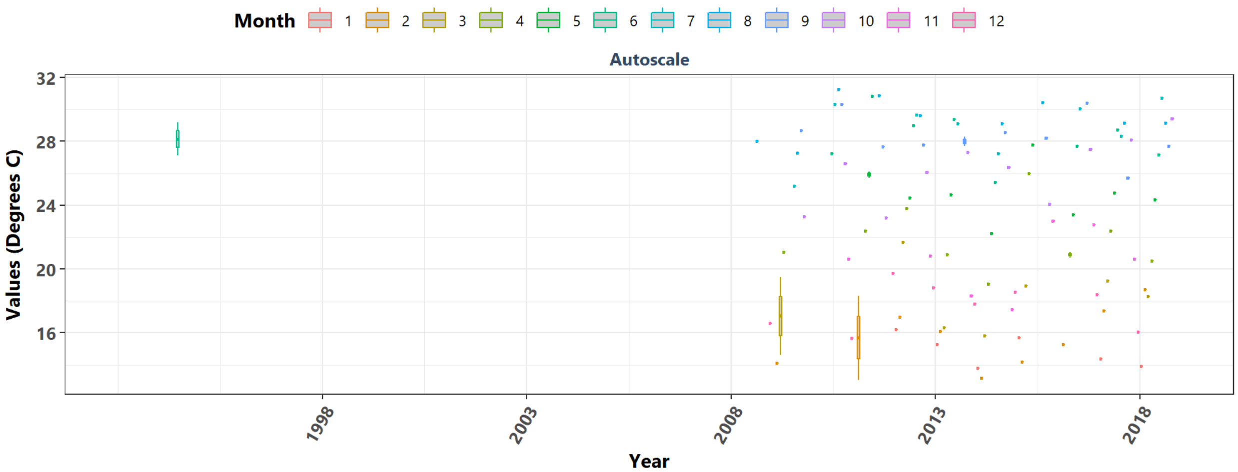
Scaled to 4x Standard Deviation

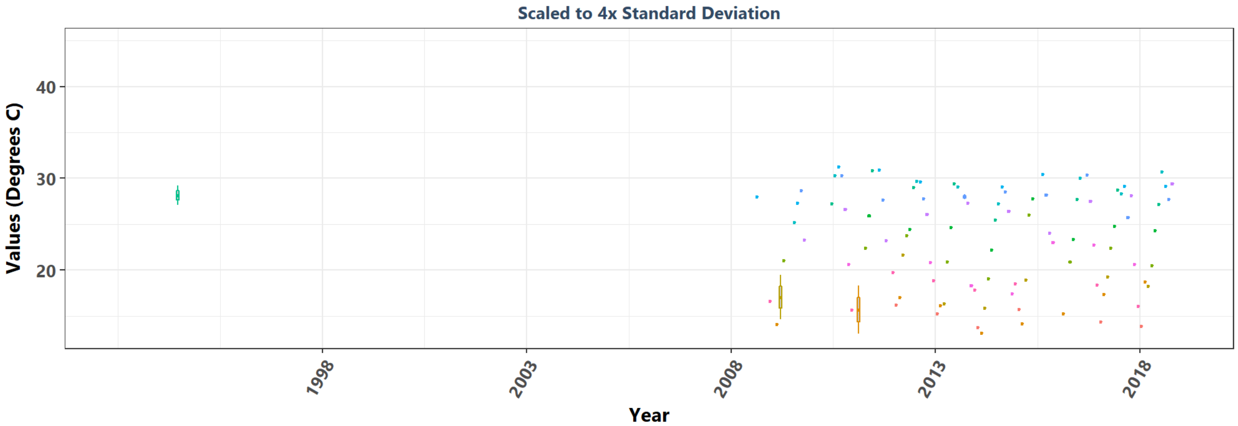


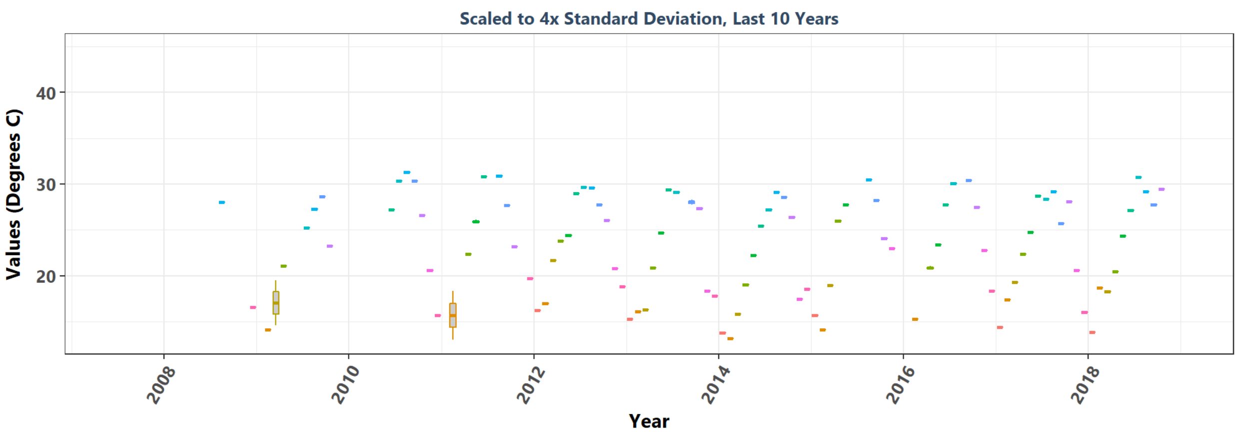
Scaled to 4x Standard Deviation, Last 10 Years

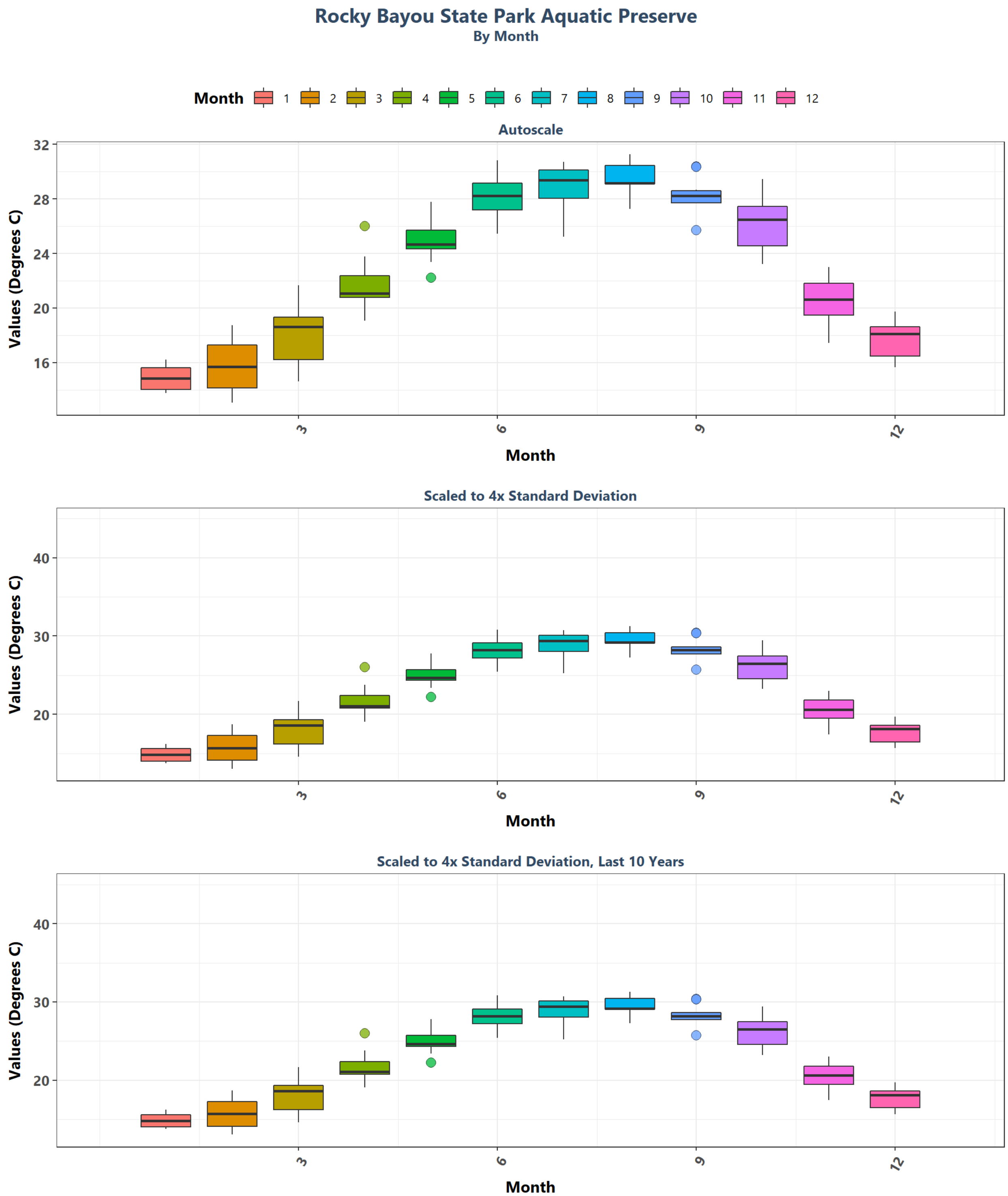


Rocky Bayou State Park Aquatic Preserve
By Year & Month





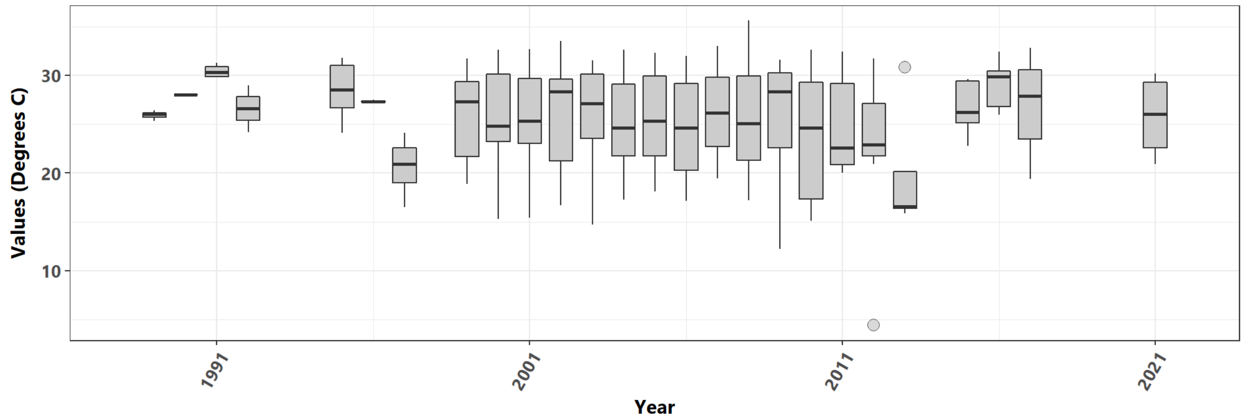




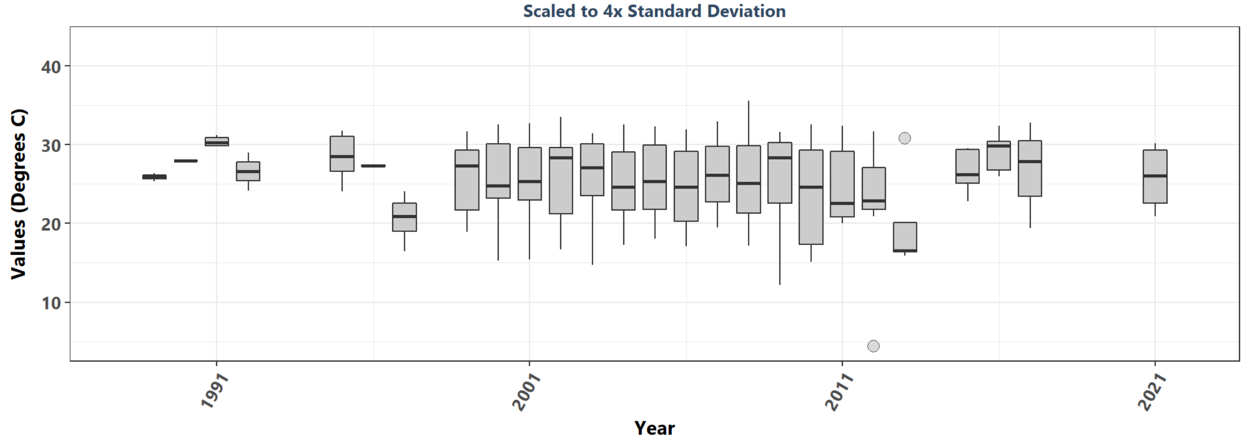
Rocky Bayou State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
32
28
24
20
16
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
3
6
9
12
Month

Rookery Bay Aquatic Preserve
By Year

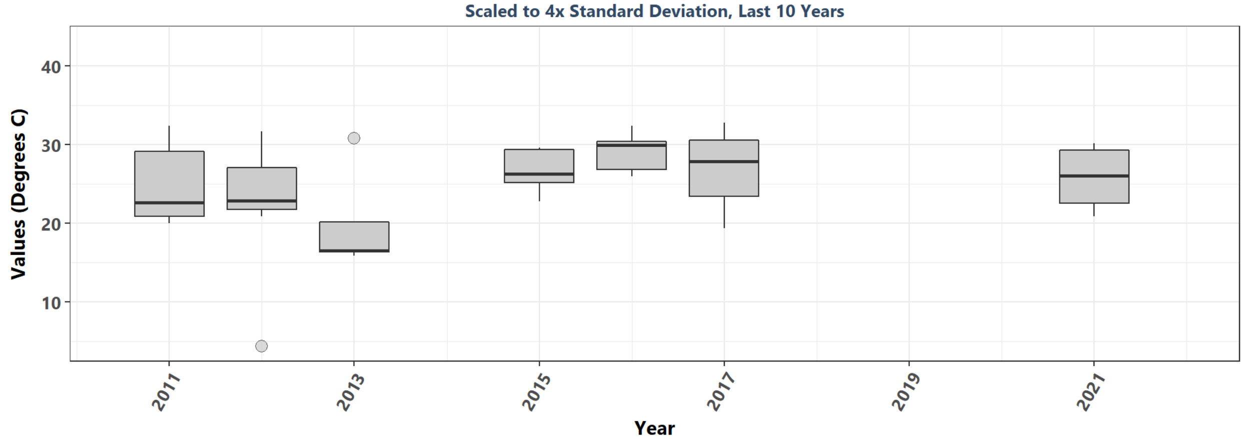
Autoscale

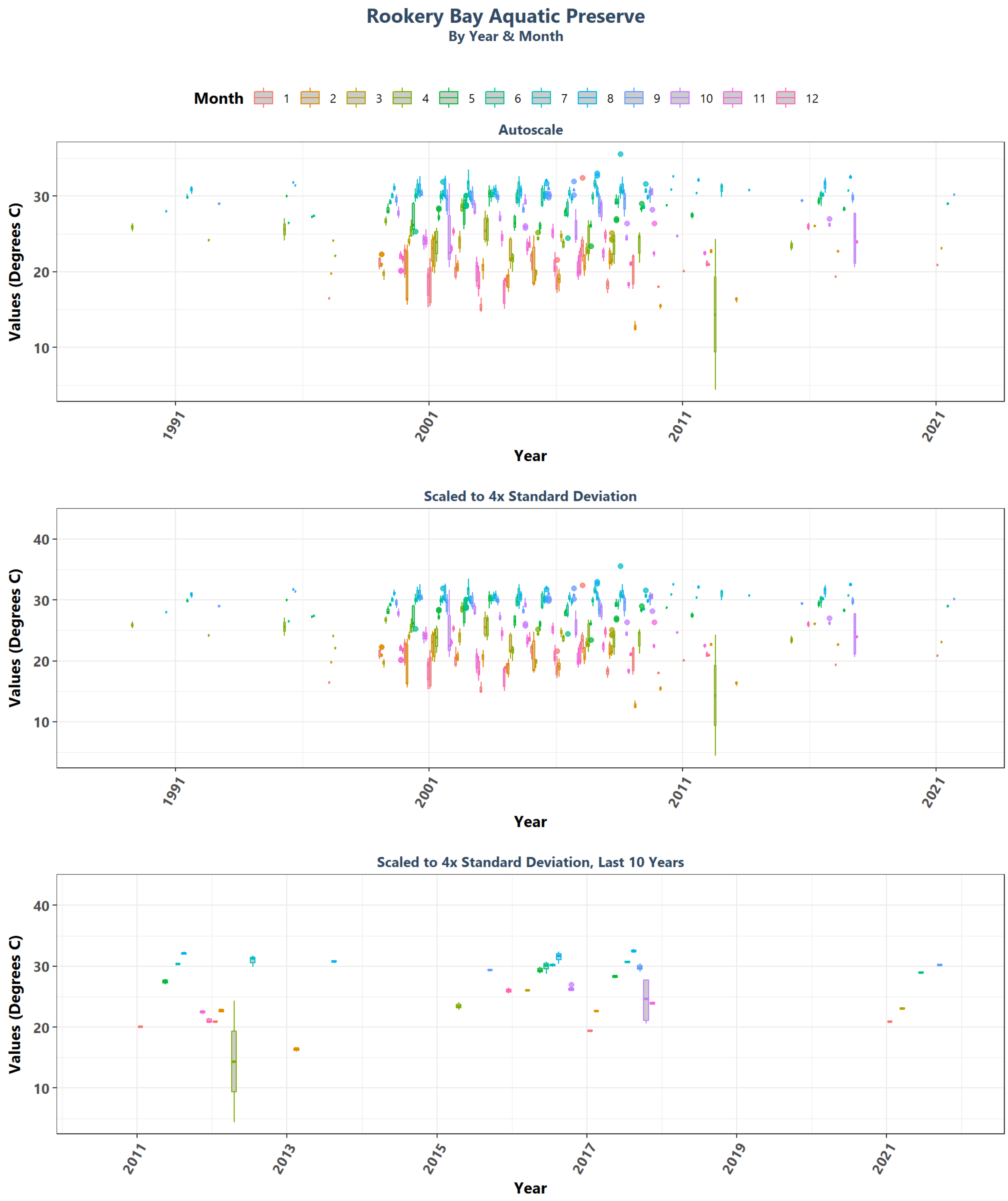


Scaled to 4x Standard Deviation

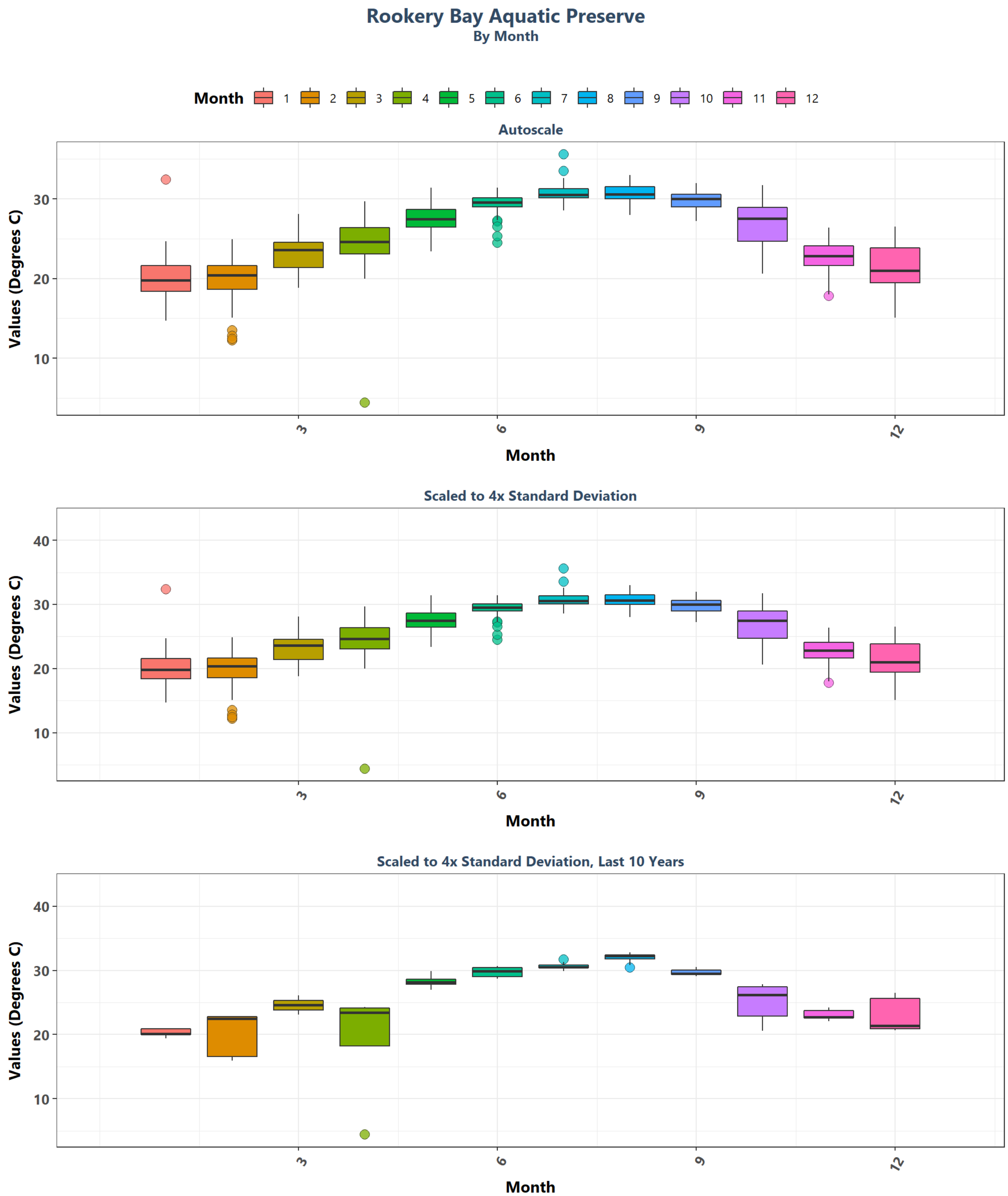


Scaled to 4x Standard Deviation, Last 10 Years

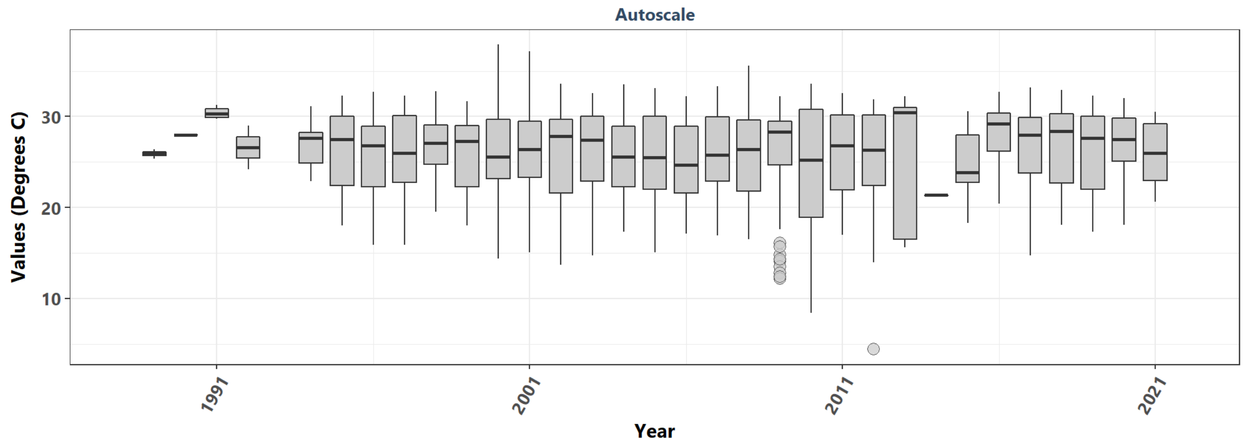


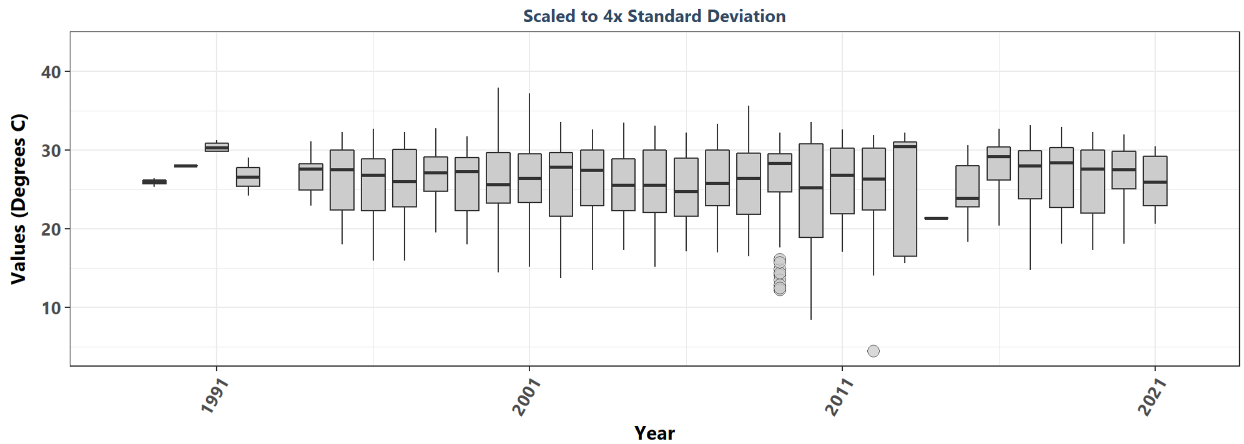


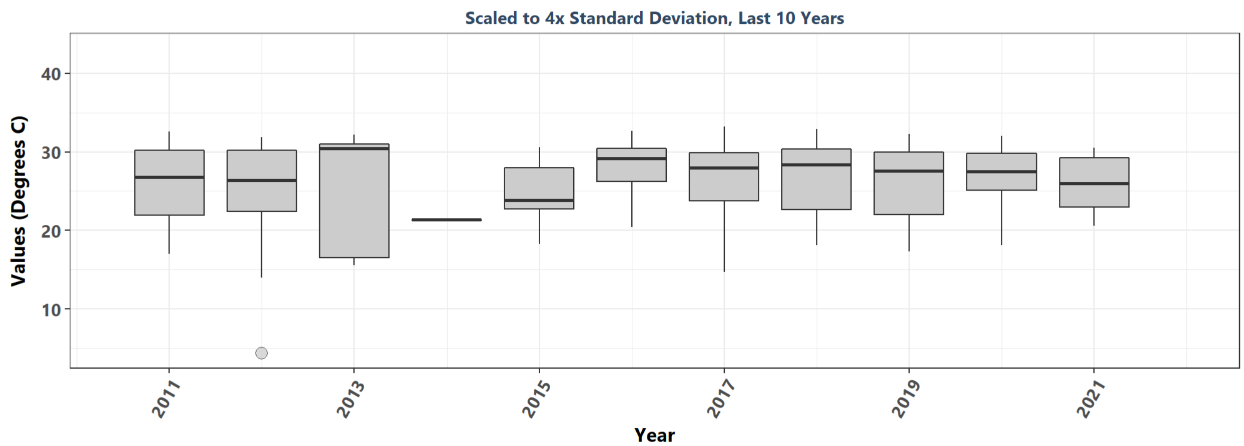
Rookery Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


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

Rookery Bay National Estuarine Research Reserve
By Year

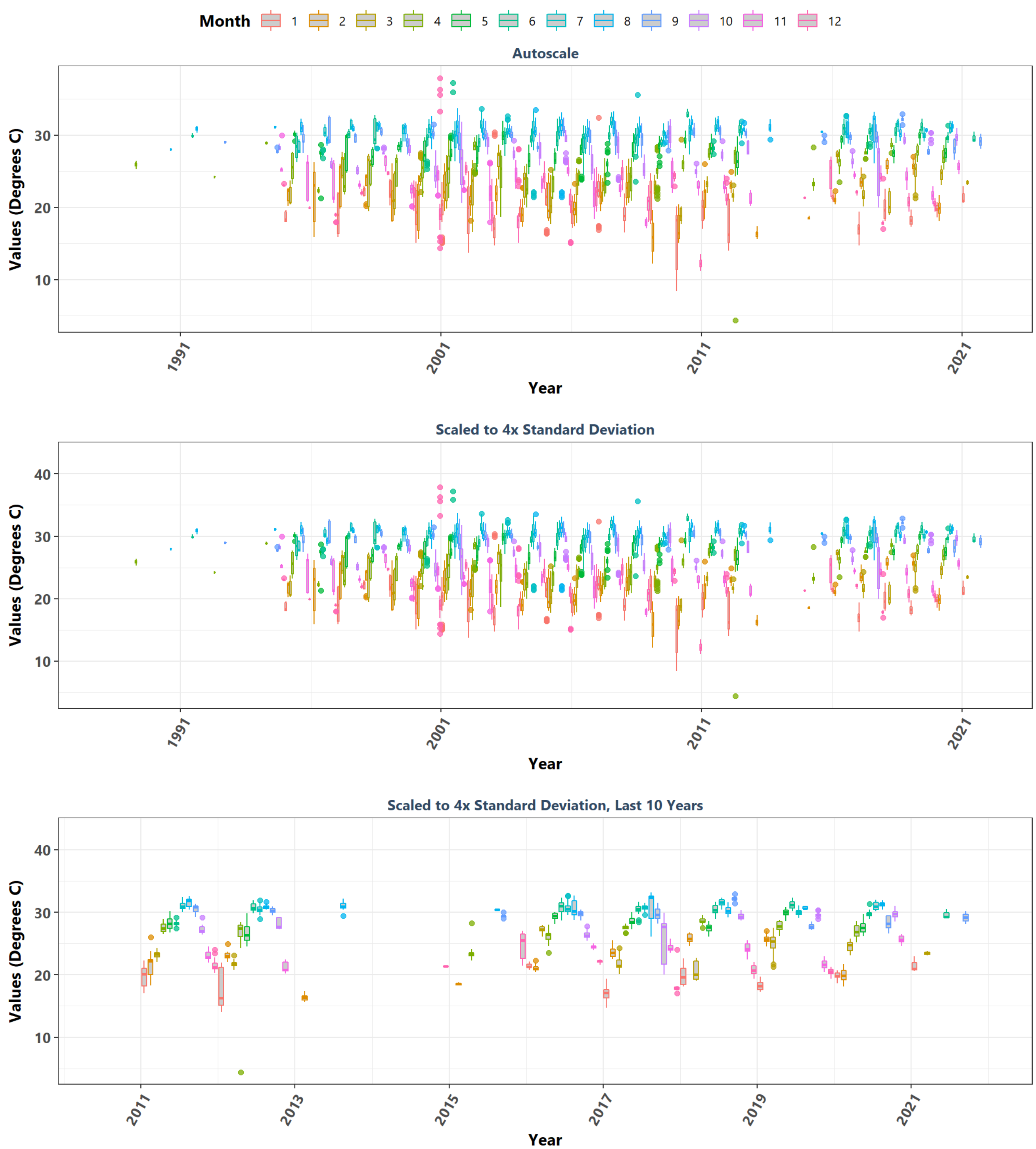


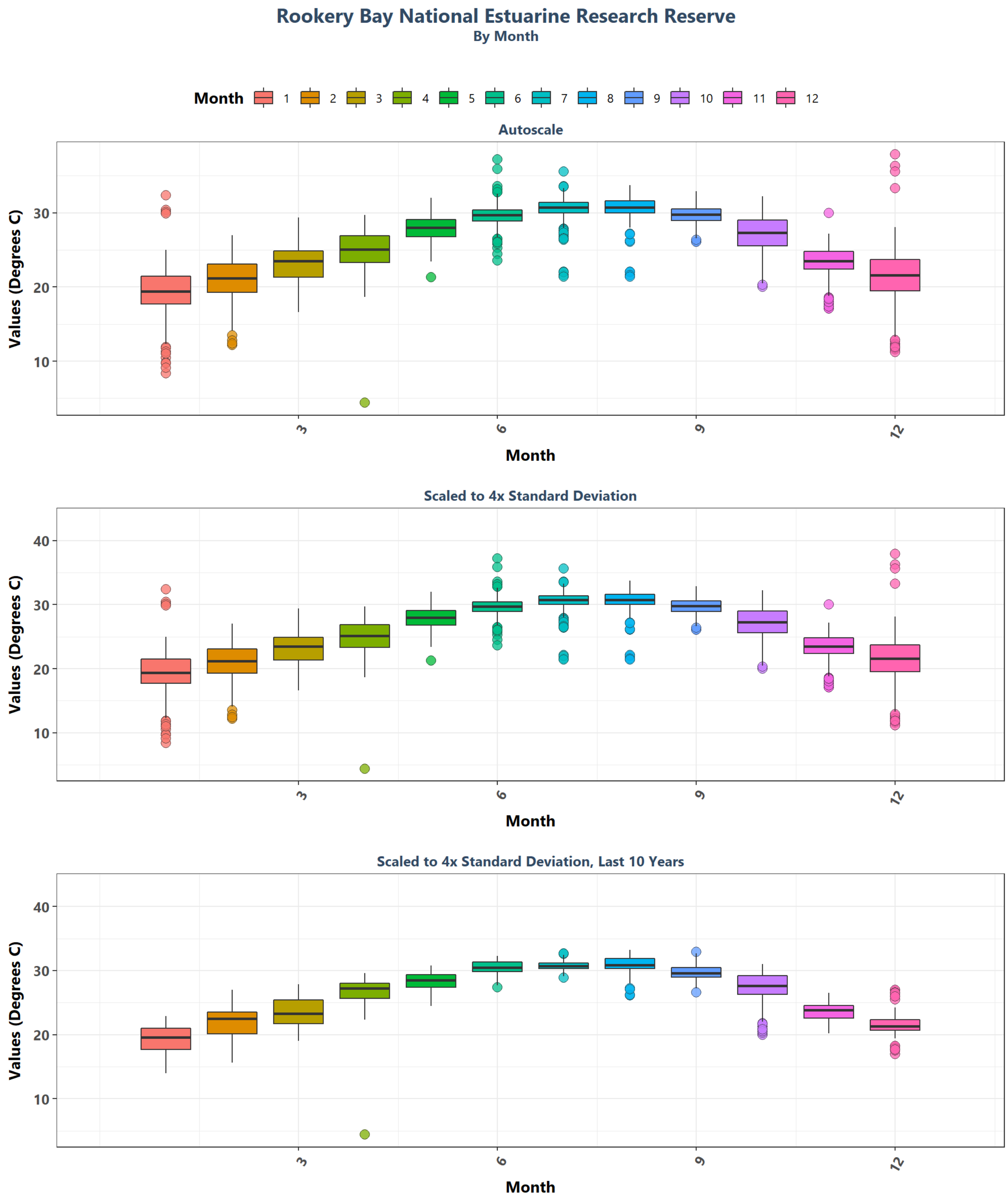




Rookery Bay National Estuarine Research Reserve

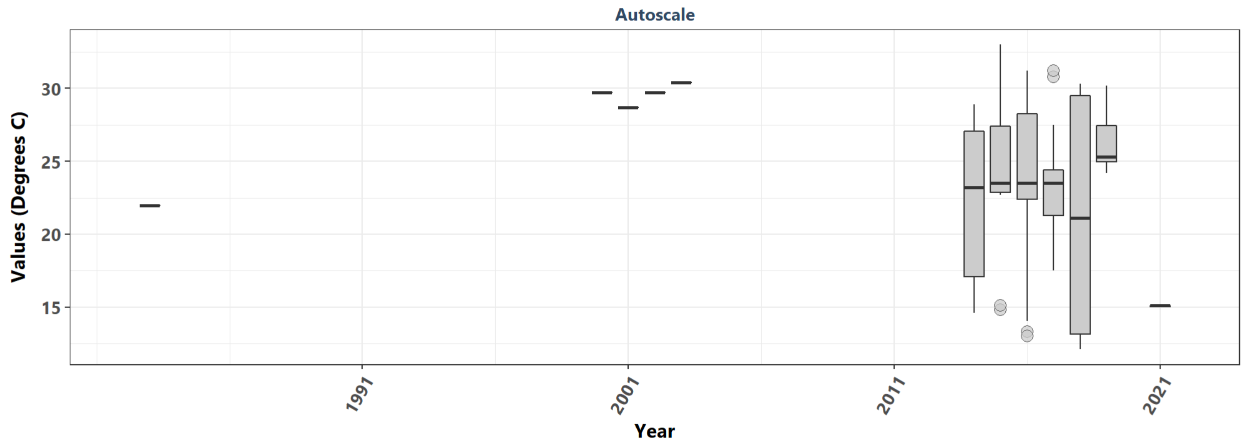
By Year & Month

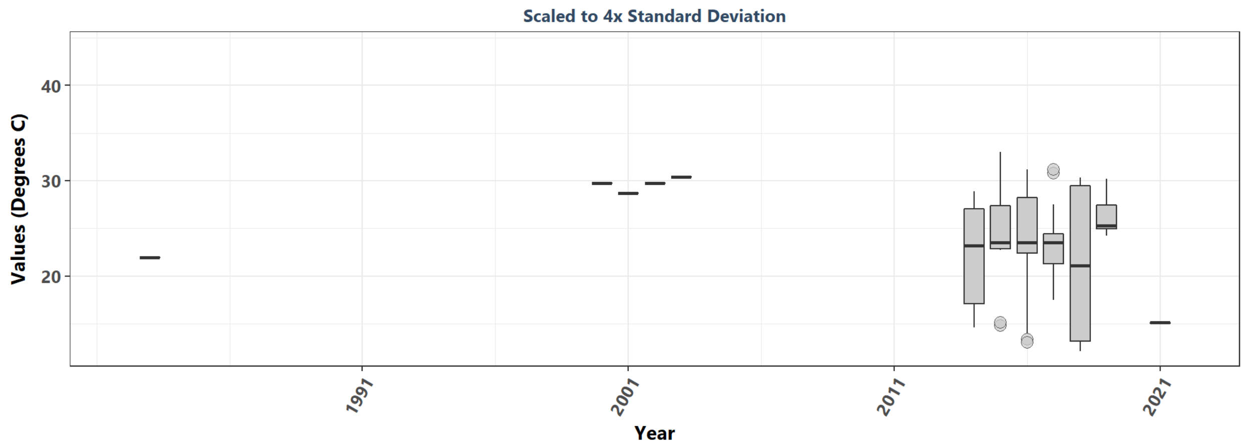


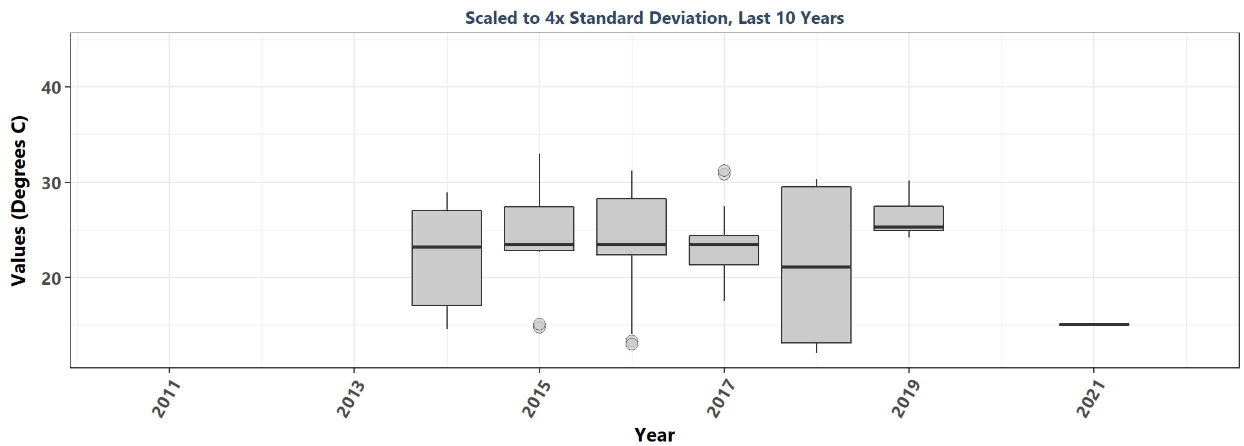

Rookery Bay National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Scaled to 4x Standard Deviation
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month
10
20
30
40
3
6
9
12

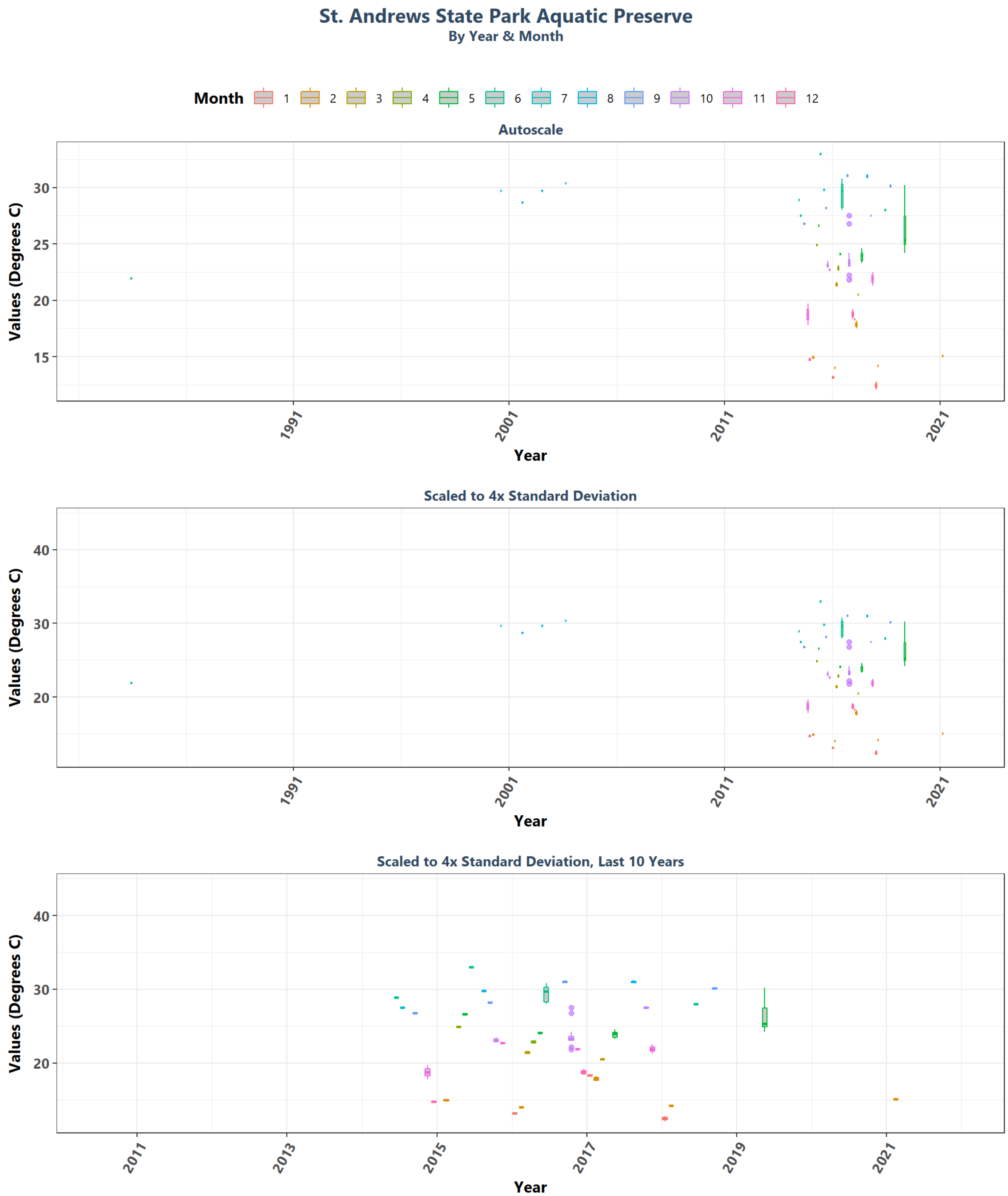
St. Andrews State Park Aquatic Preserve

By Year

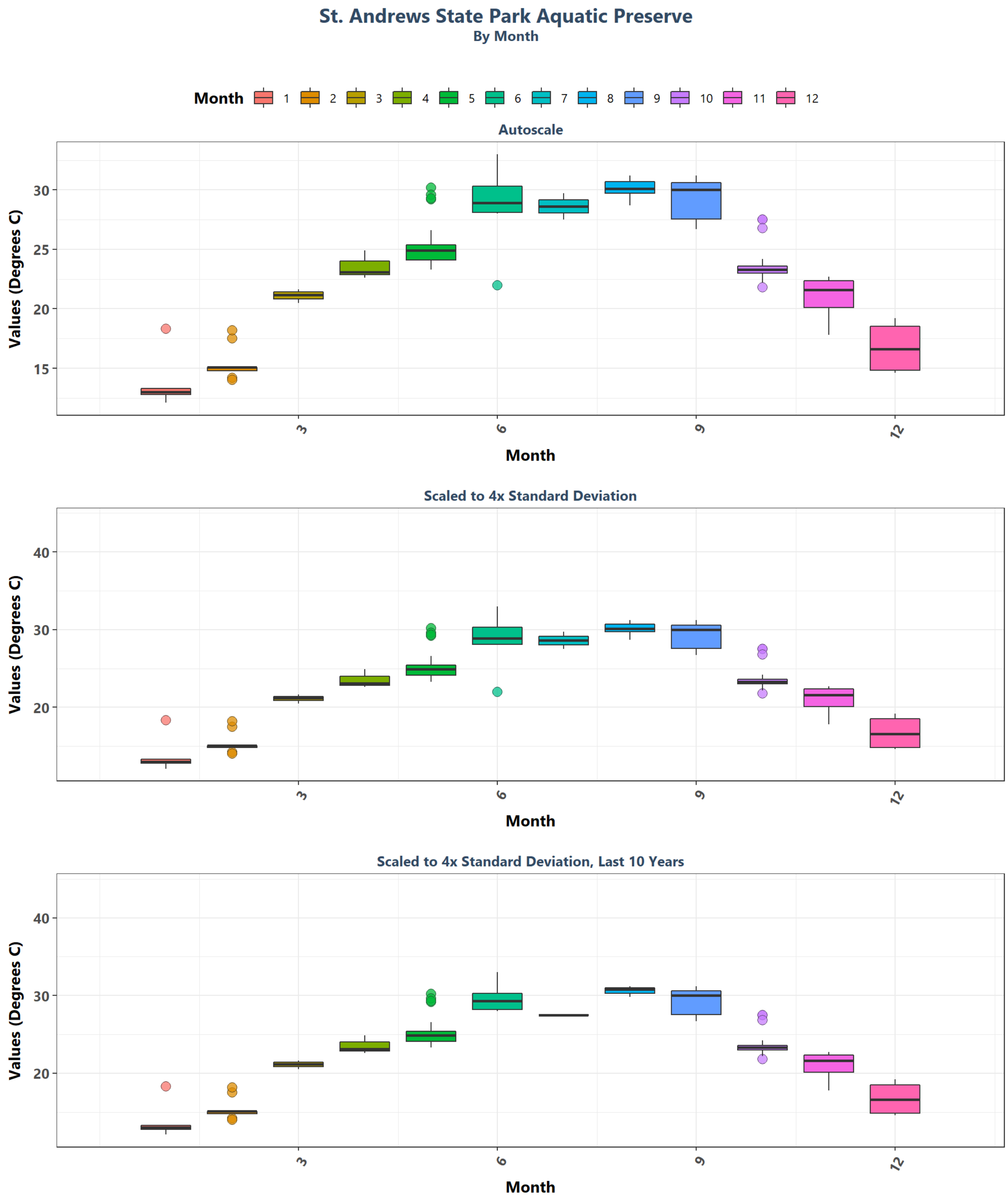




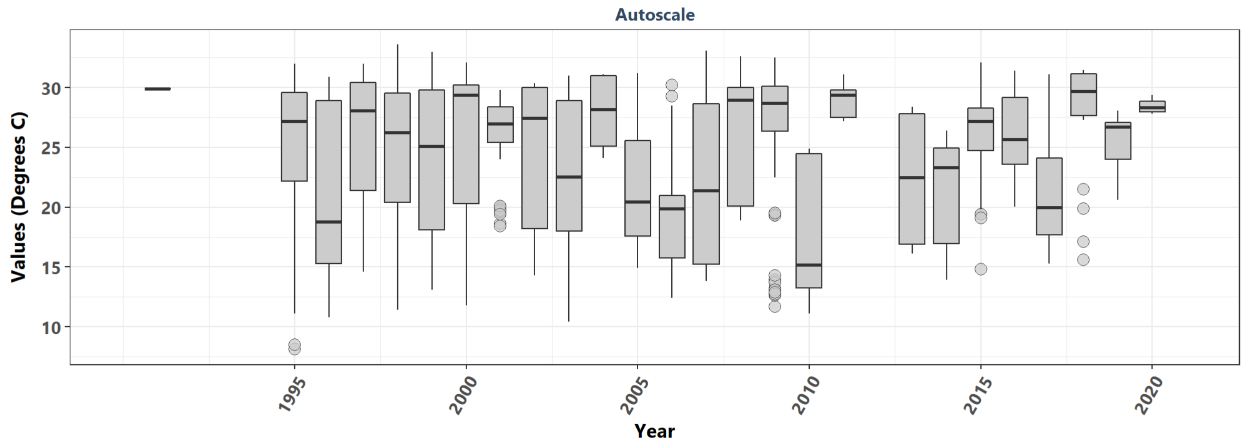


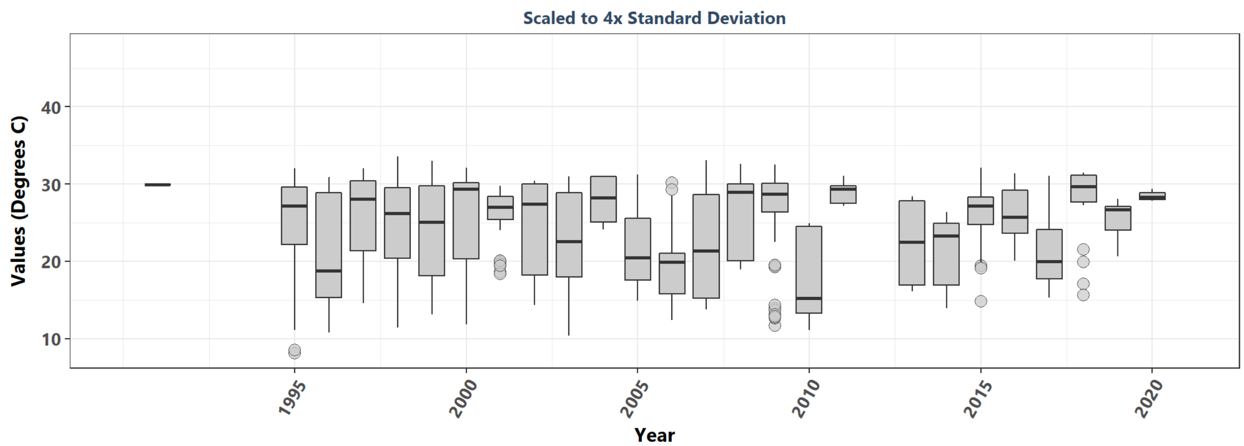


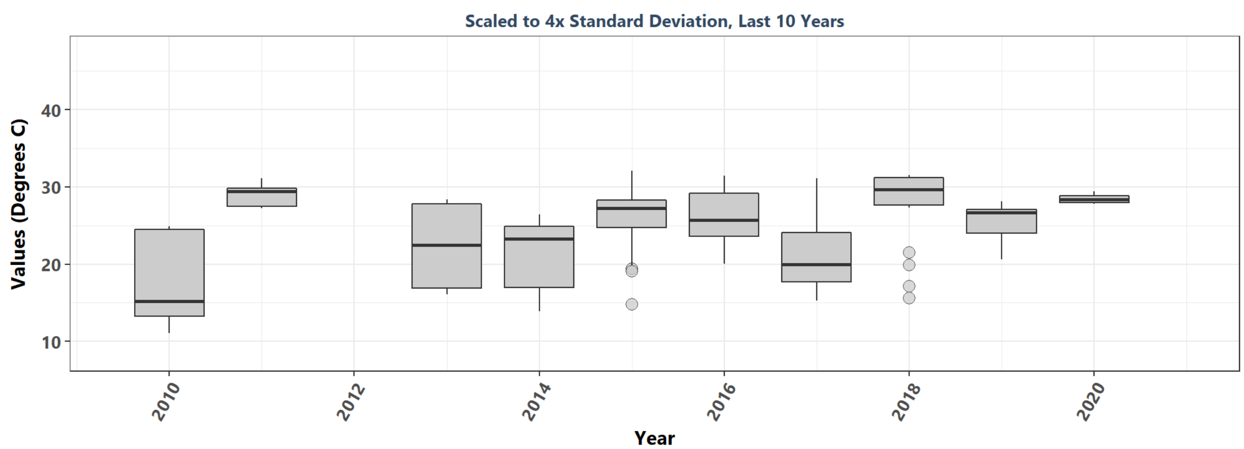
St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
2011
2013
2015
2017
2019
2021
Year

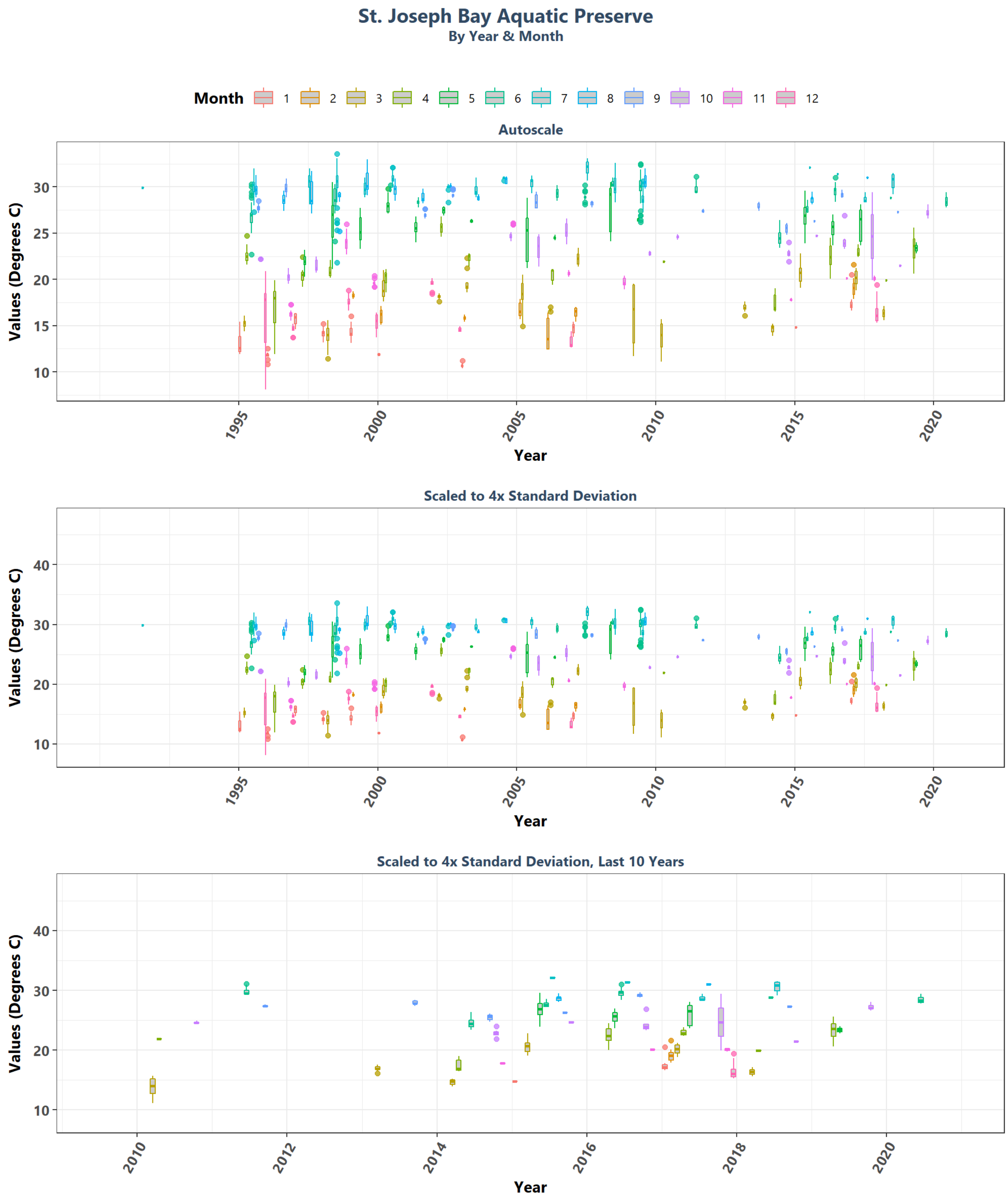

St. Andrews State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
3
6
9
12
Month

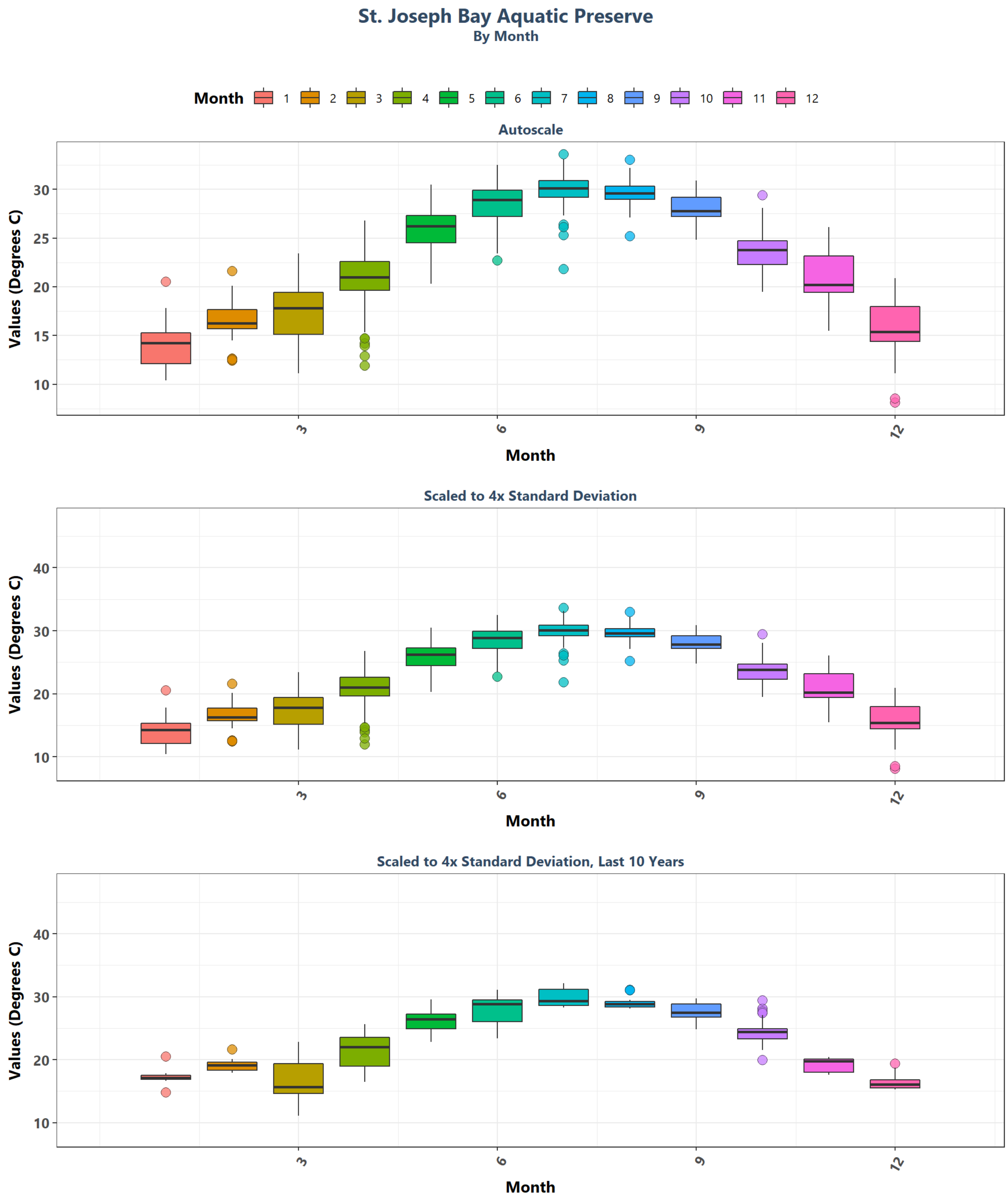
St. Joseph Bay Aquatic Preserve
By Year



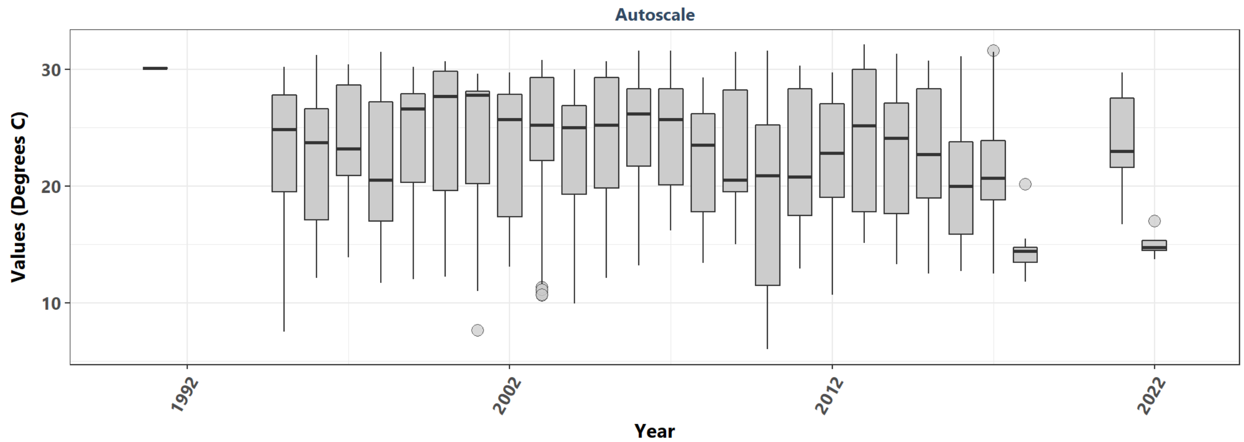


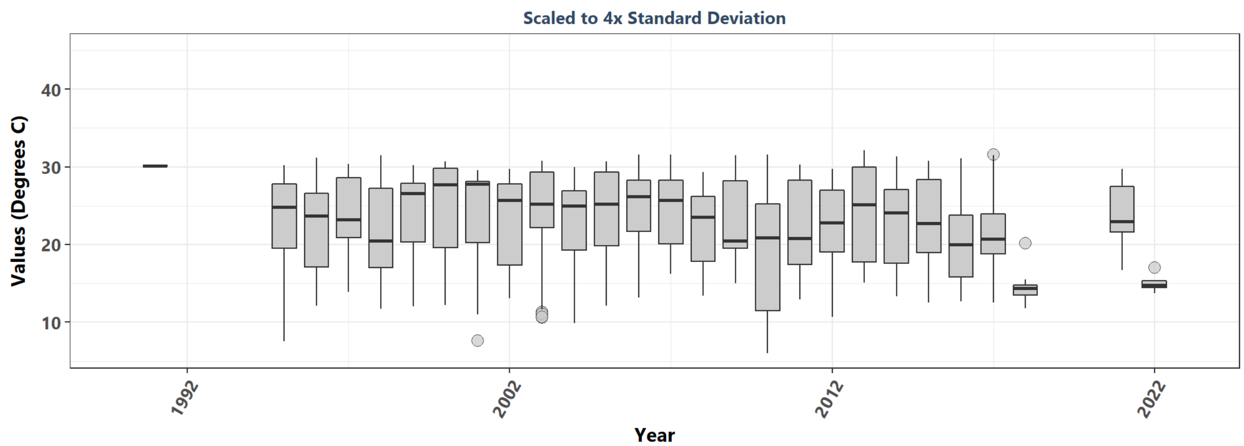


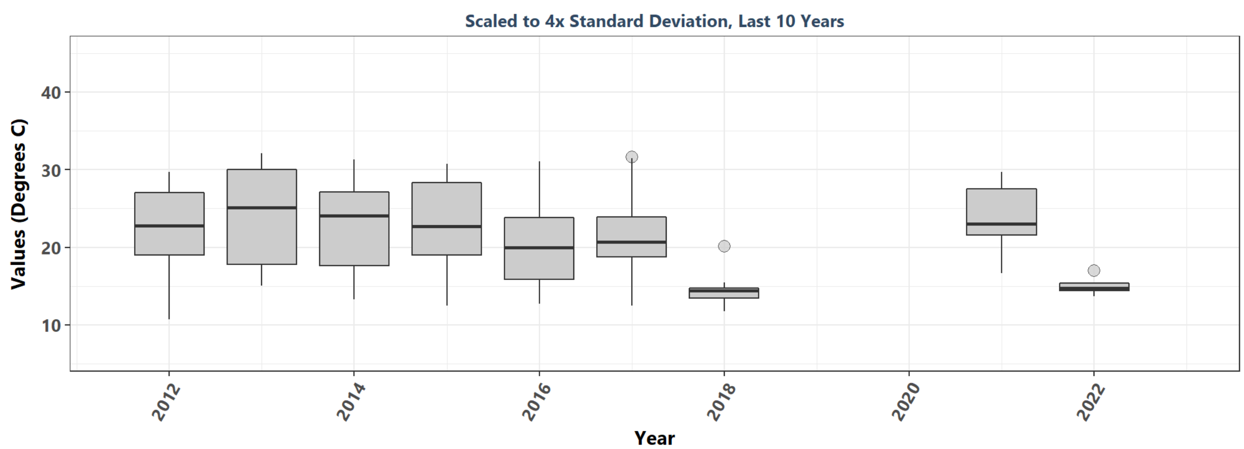

St. Joseph Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10 15 20 25 30
1995 2000 2005 2010 2015 2020
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10 20 30 40
1995 2000 2005 2010 2015 2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10 20 30 40
2010 2012 2014 2016 2018 2020
Year


St. Joseph Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
15
20
25
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

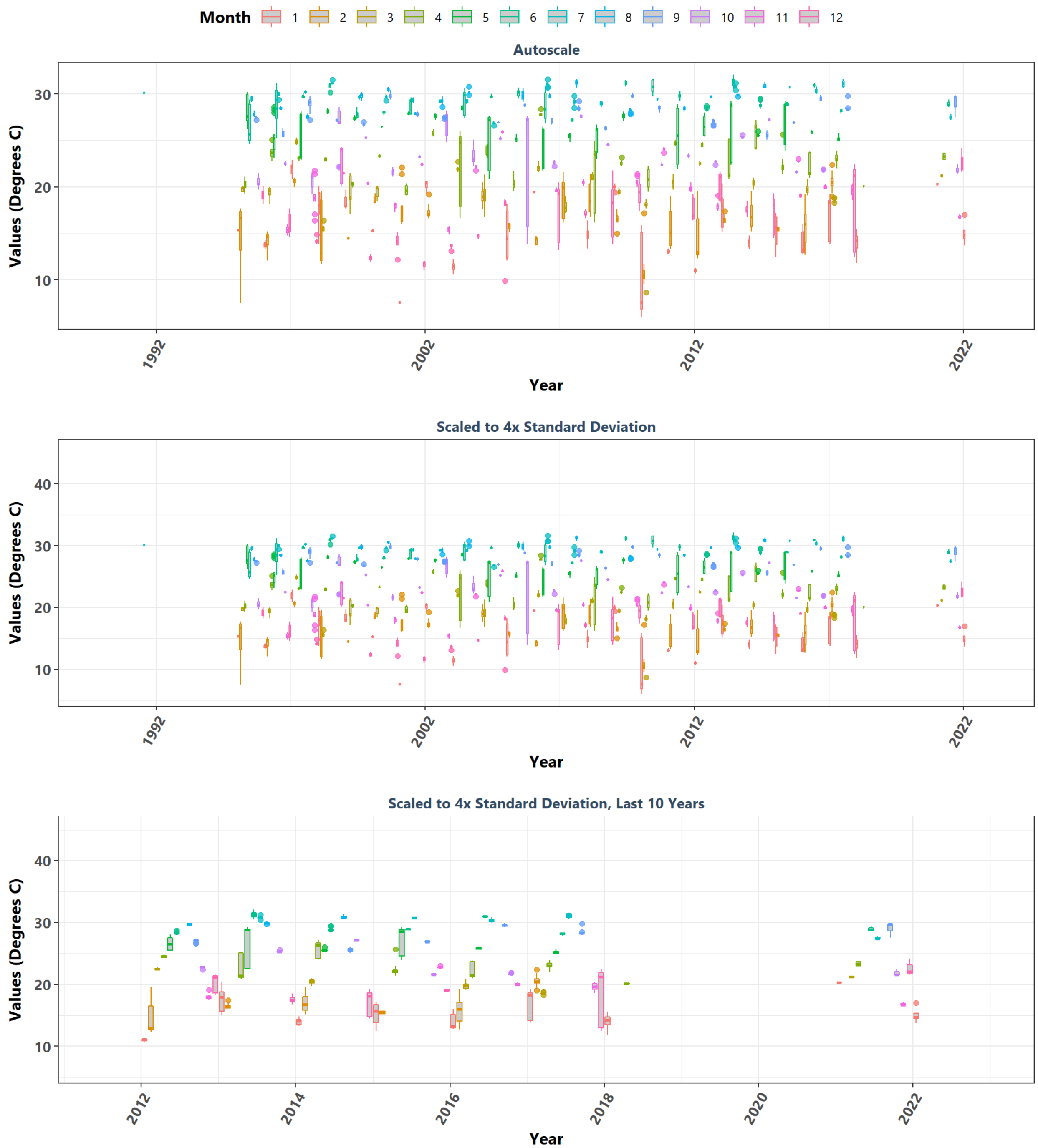
St. Martins Marsh Aquatic Preserve
By Year

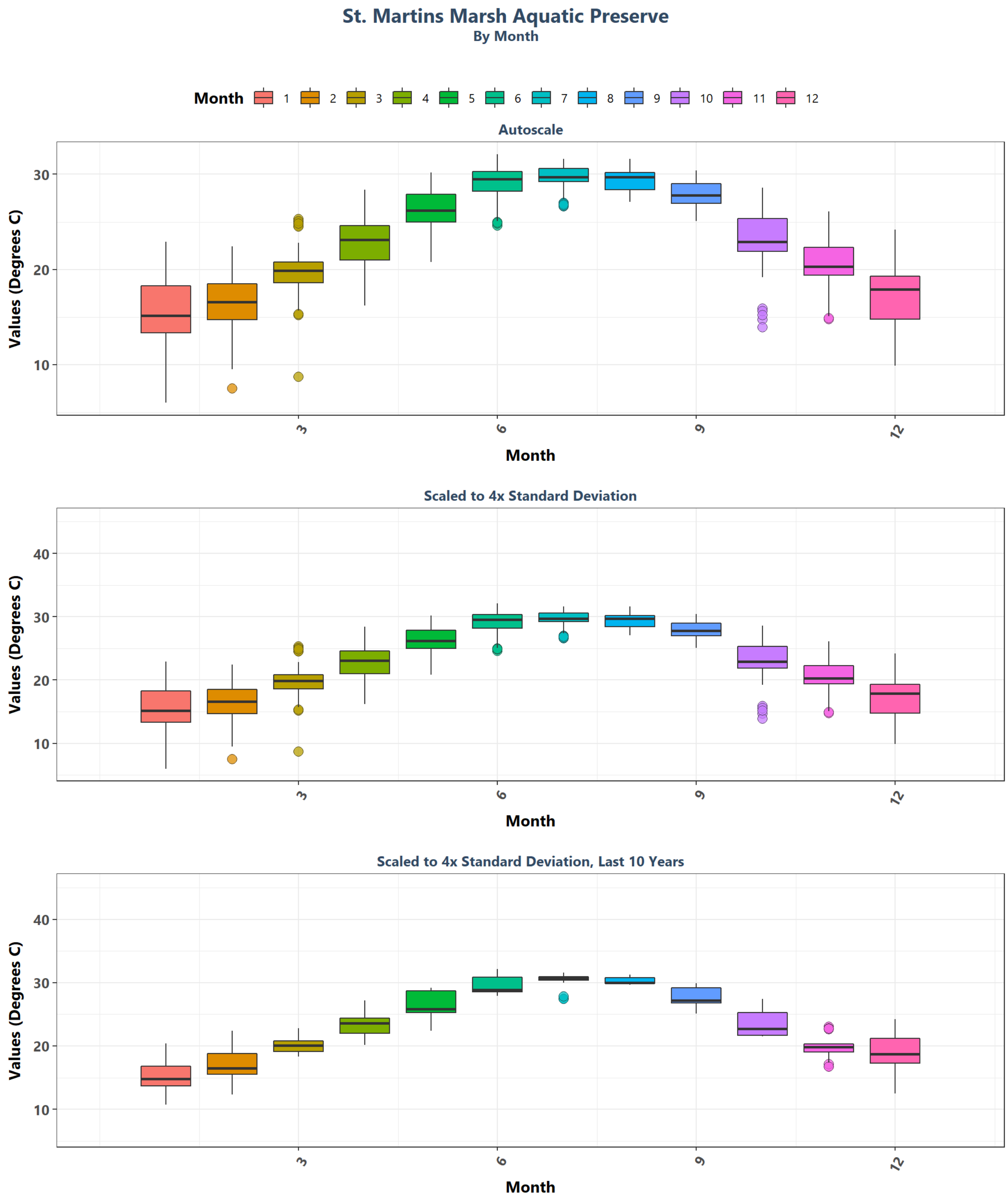




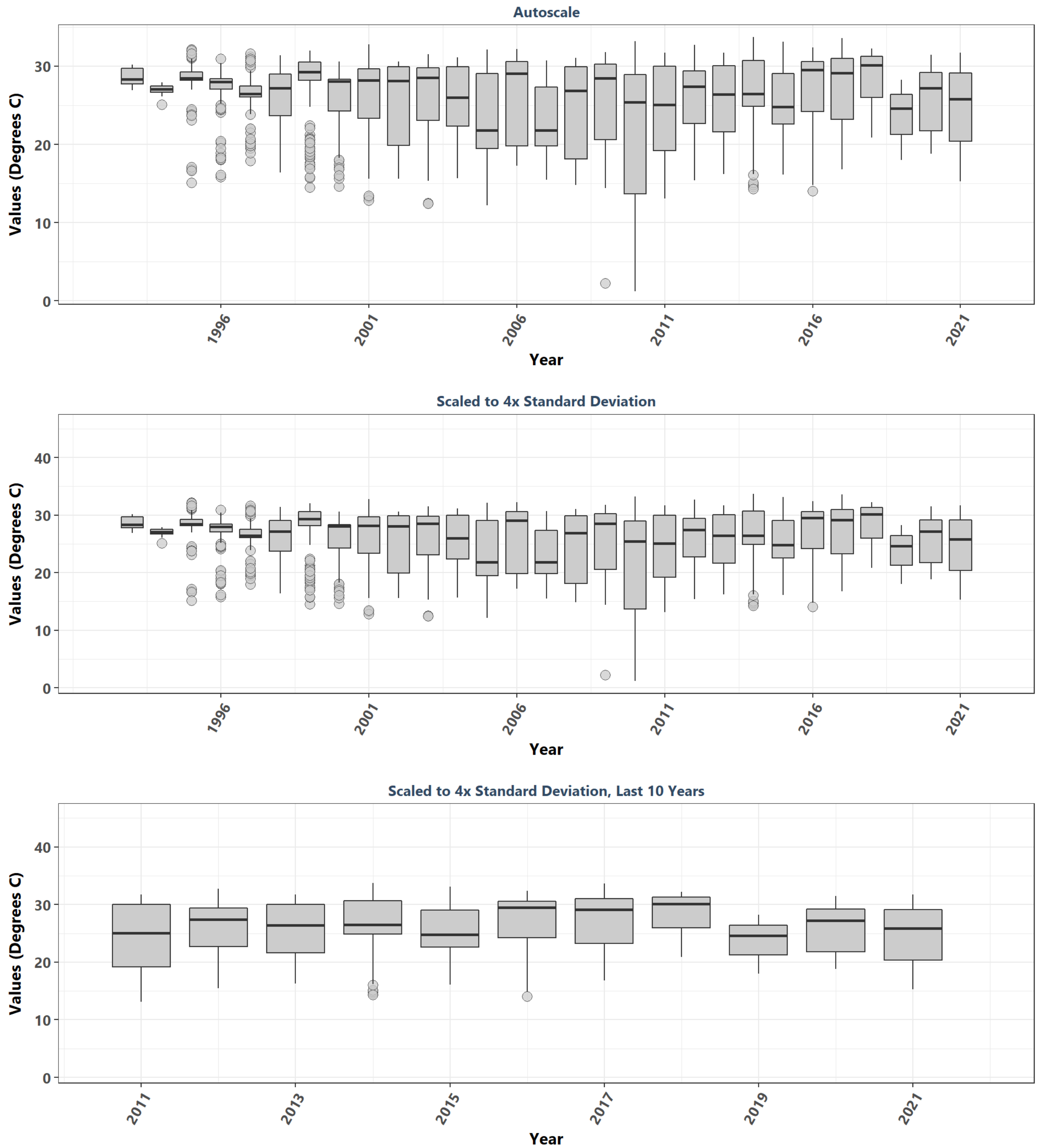


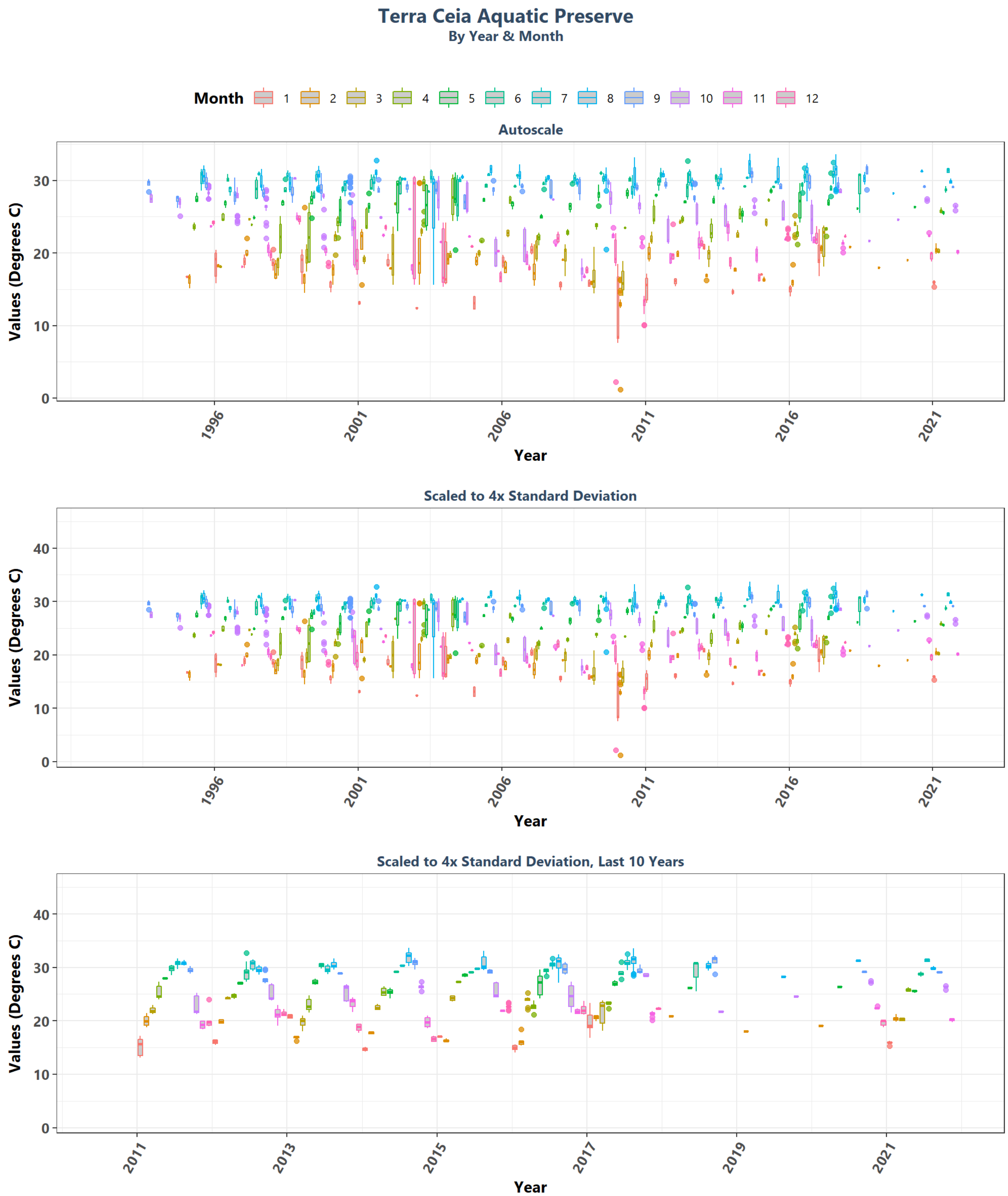
St. Martins Marsh Aquatic Preserve
By Year & Month



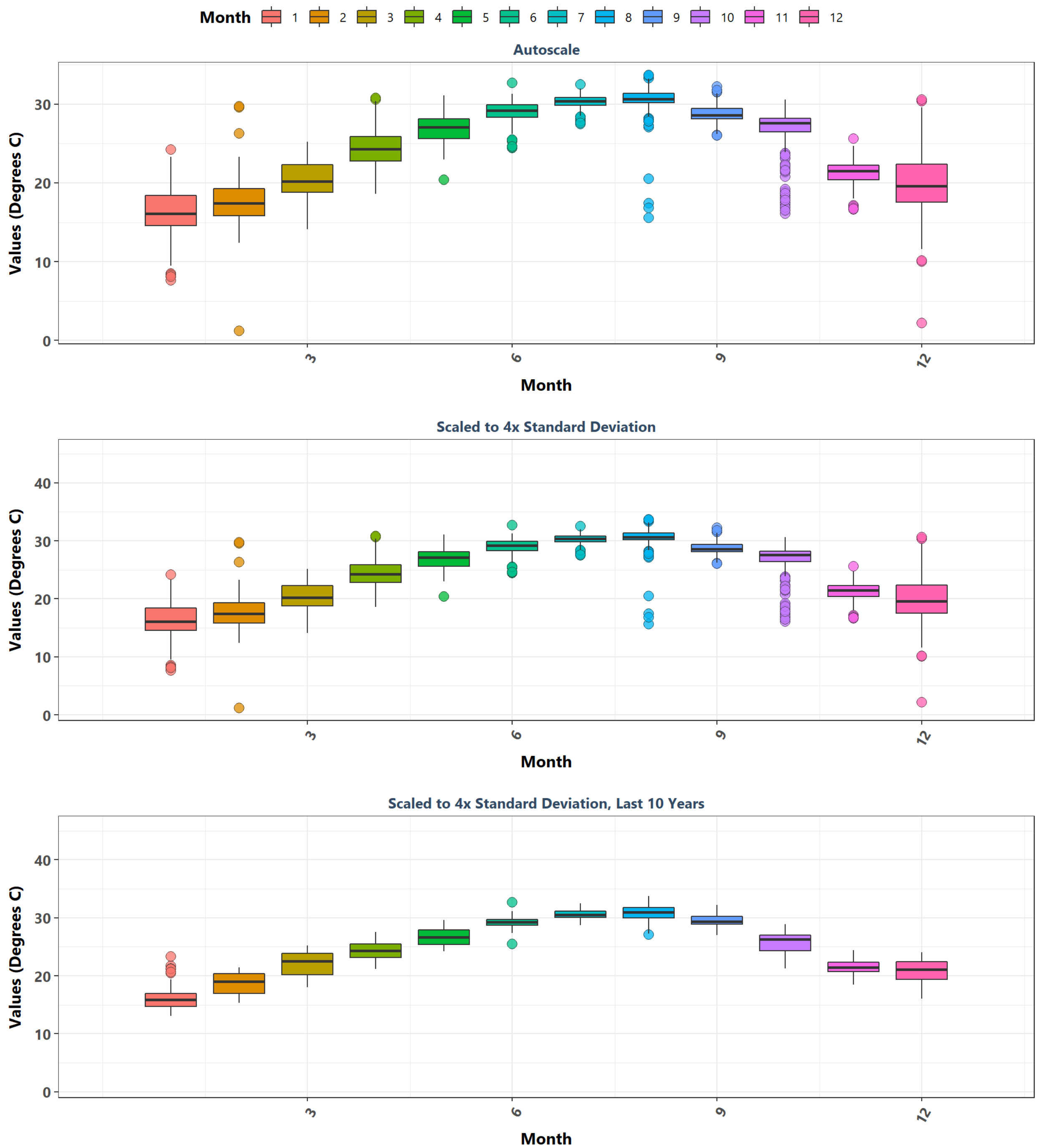

St. Martins Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

Terra Ceia Aquatic Preserve
By Year

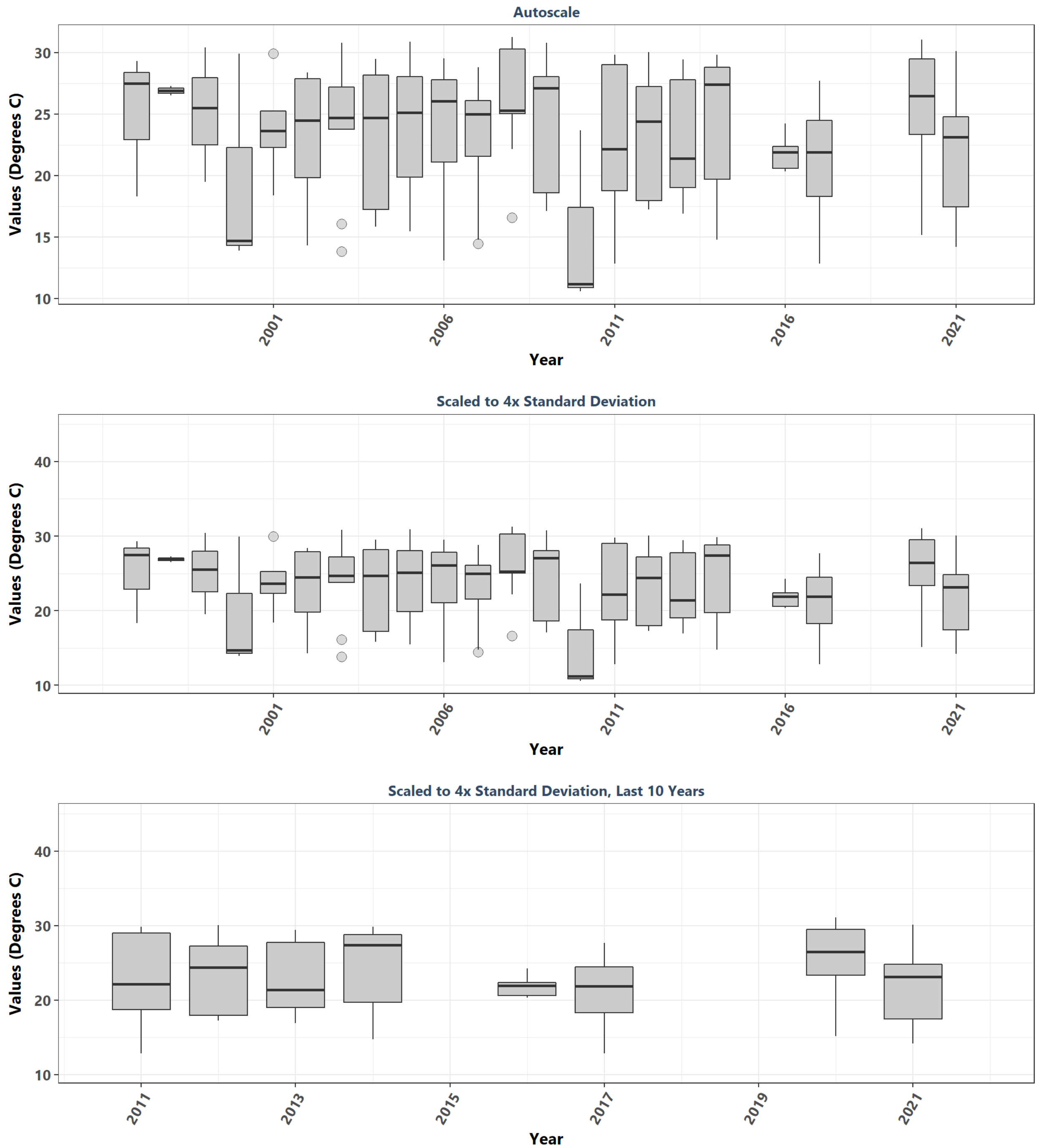


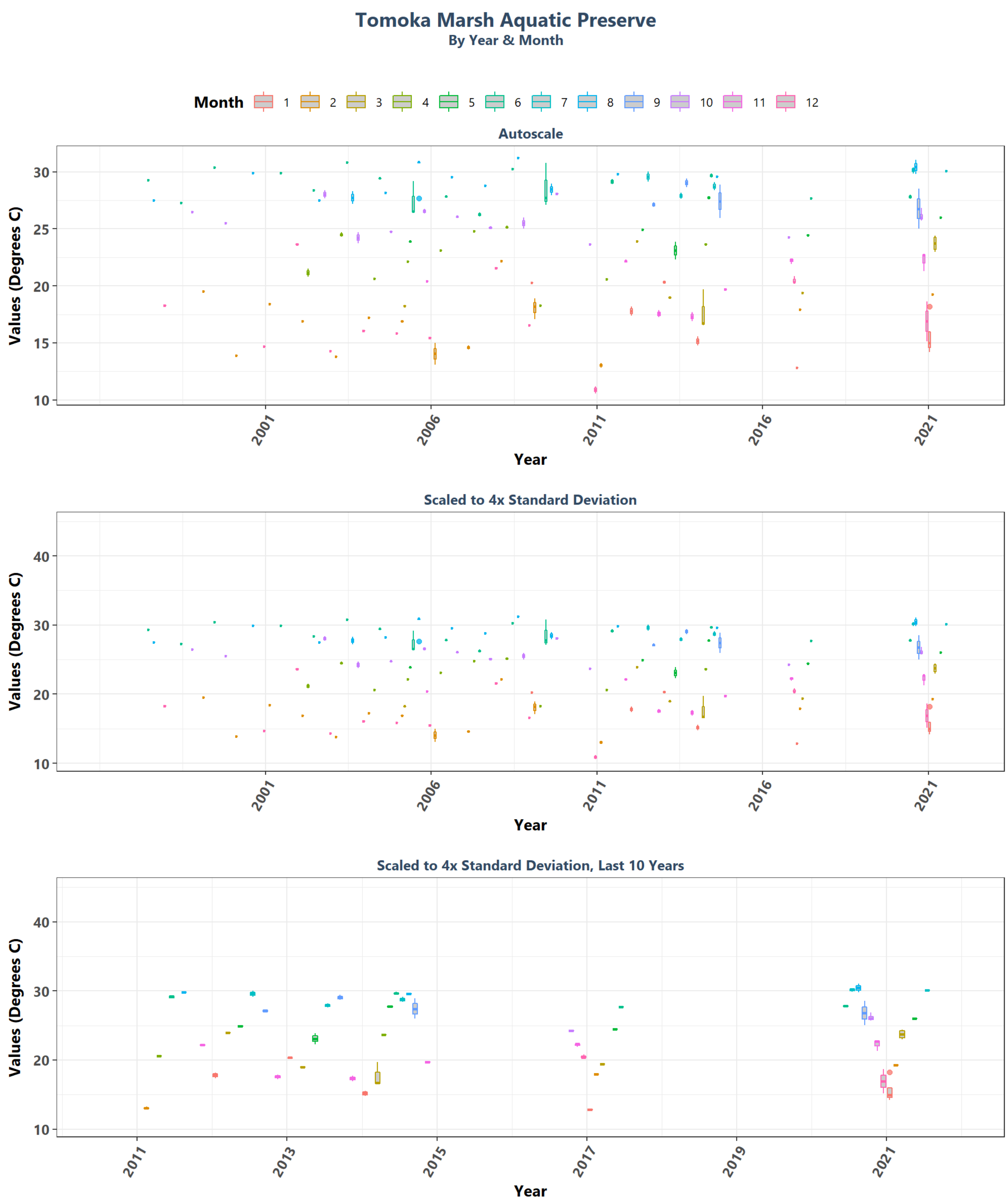

Terra Ceia Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0
10
20
30
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
2011
2013
2015
2017
2019
2021
Year

Terra Ceia Aquatic Preserve
By Month

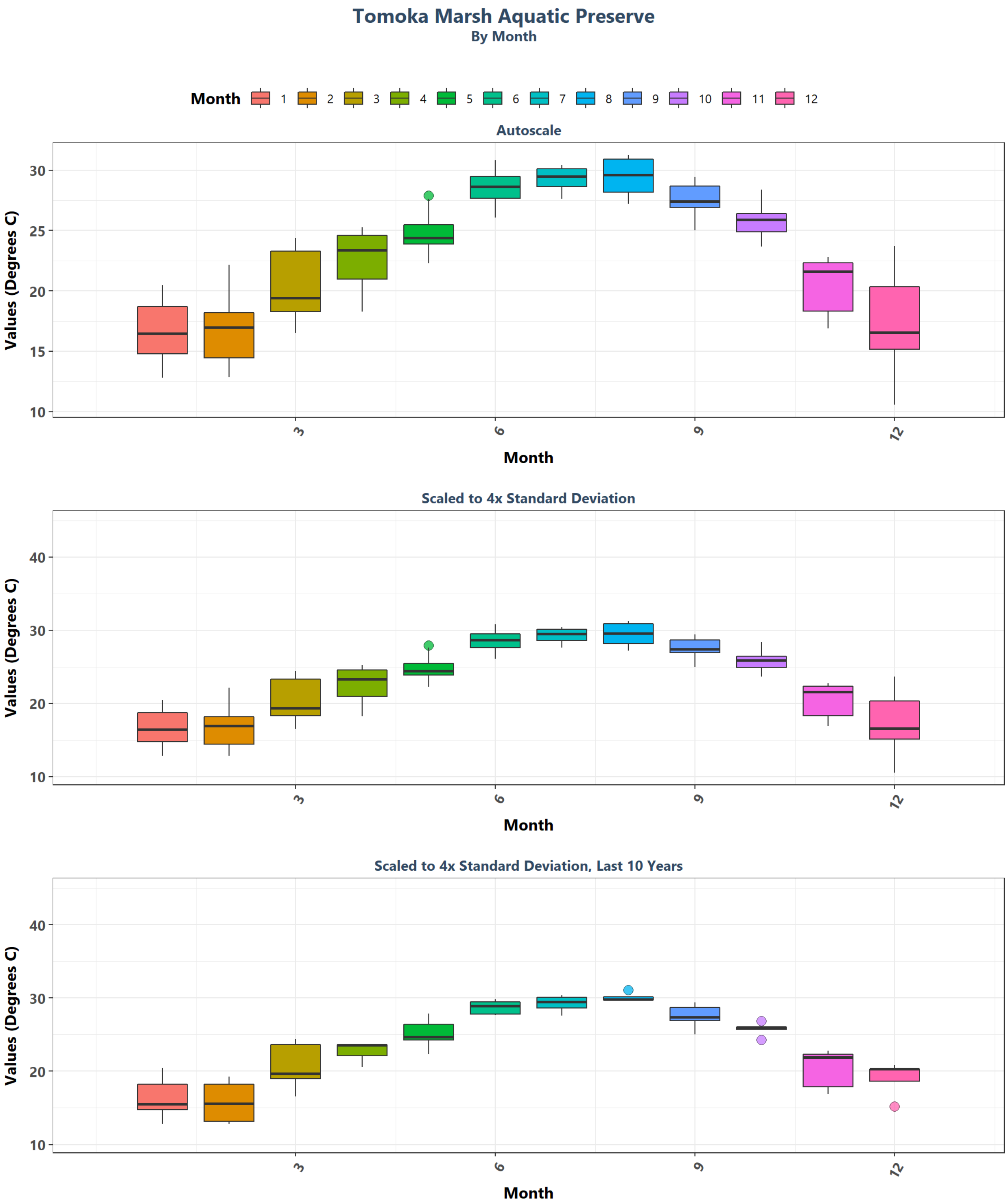


Tomoka Marsh Aquatic Preserve
By Year

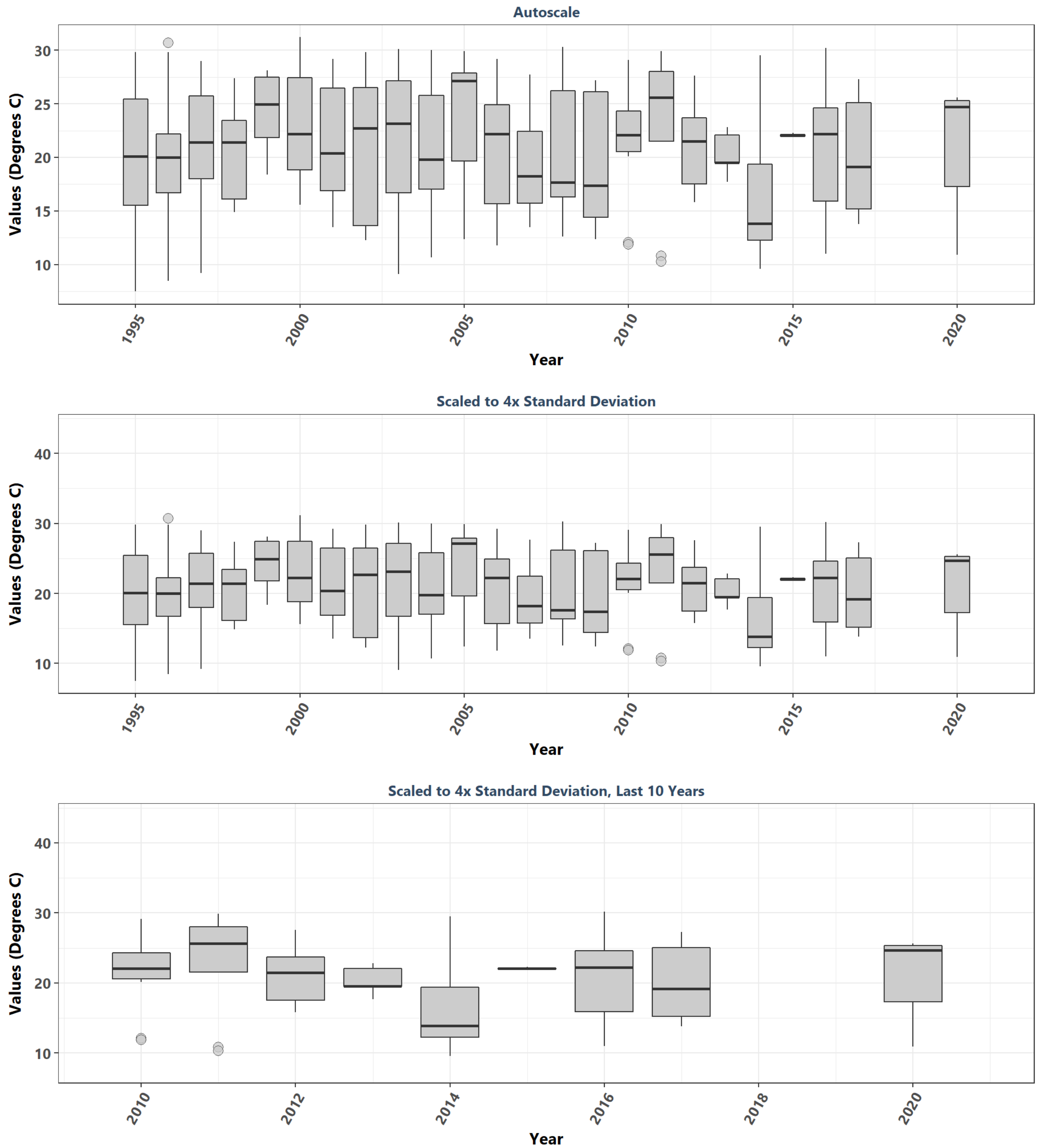


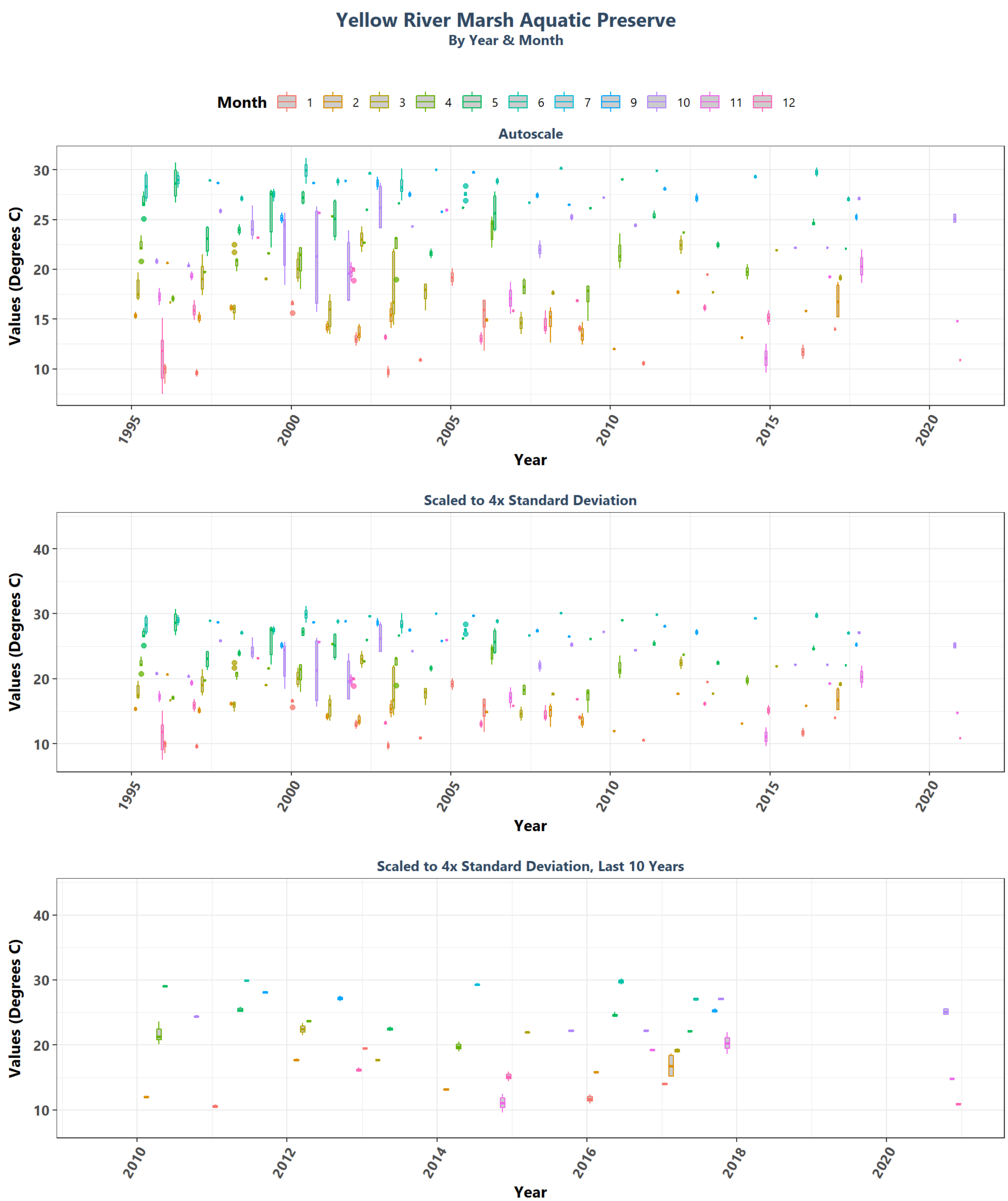


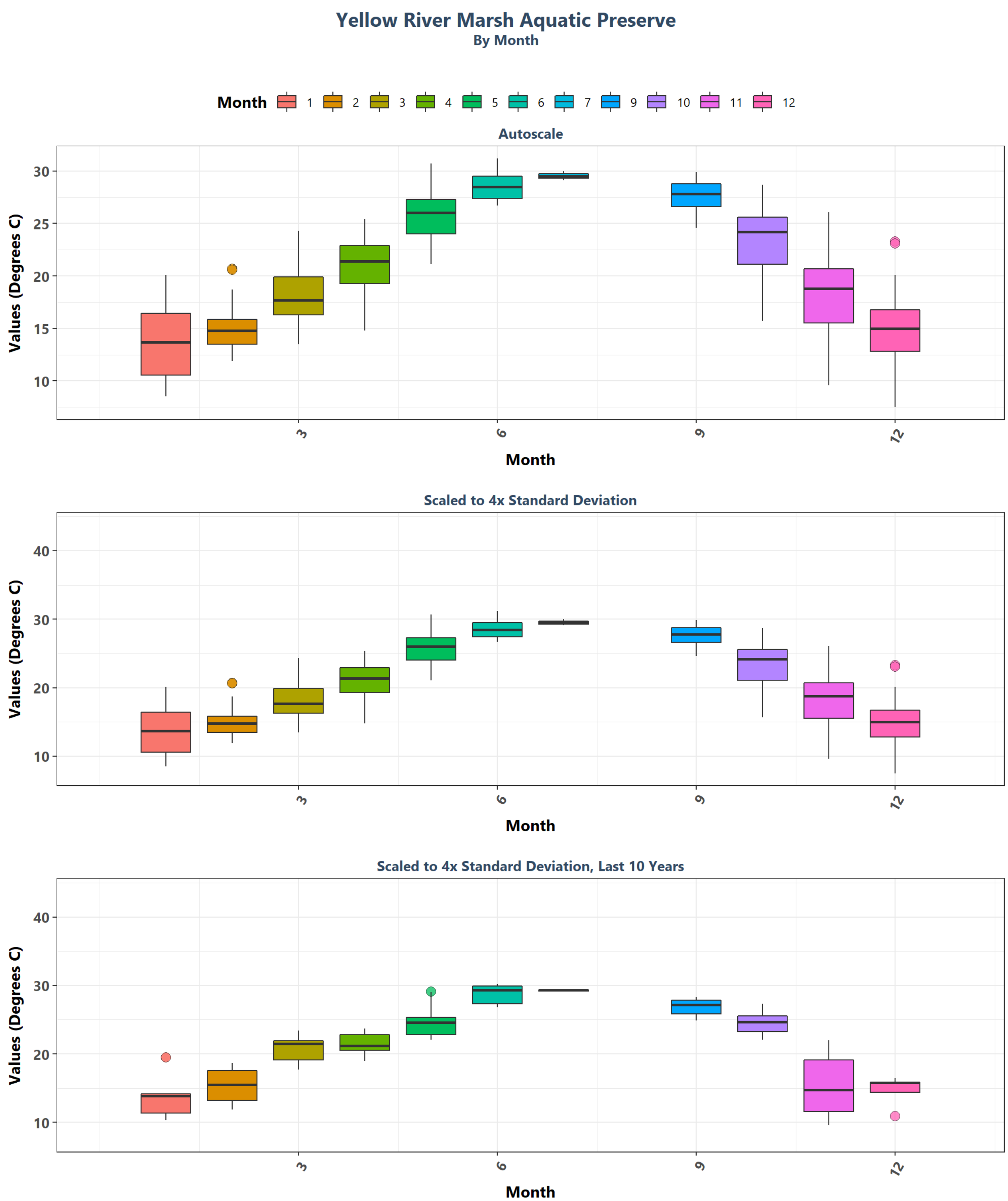
Tomoka Marsh Aquatic Preserve
By Year & Month
Month
1
2
3
4
5
6
7
8
9
10
11
12
Autoscale
Values (Degrees C)
30
25
20
15
10
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2011
2013
2015
2017
2019
2021
Year


Tomoka Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

Yellow River Marsh Aquatic Preserve
By Year




Yellow River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
15
10
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1995
2000
2005
2010
2015
2020
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2010
2012
2014
2016
2018
2020
Year




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