SEACAR Discrete Water Quality Analysis: Field Surface 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: 600331, Number Passed Filter: 600331
## I Codes: 0 (0%)
## Q Codes: 0 (0%)
## U Codes: 0 (0%)
```

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

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

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

Managed Area Statistics

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

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

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

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

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

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

Monitoring Location Statistics

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

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

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

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

Seasonal Kendall Tau Analysis

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

The following steps are performed:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

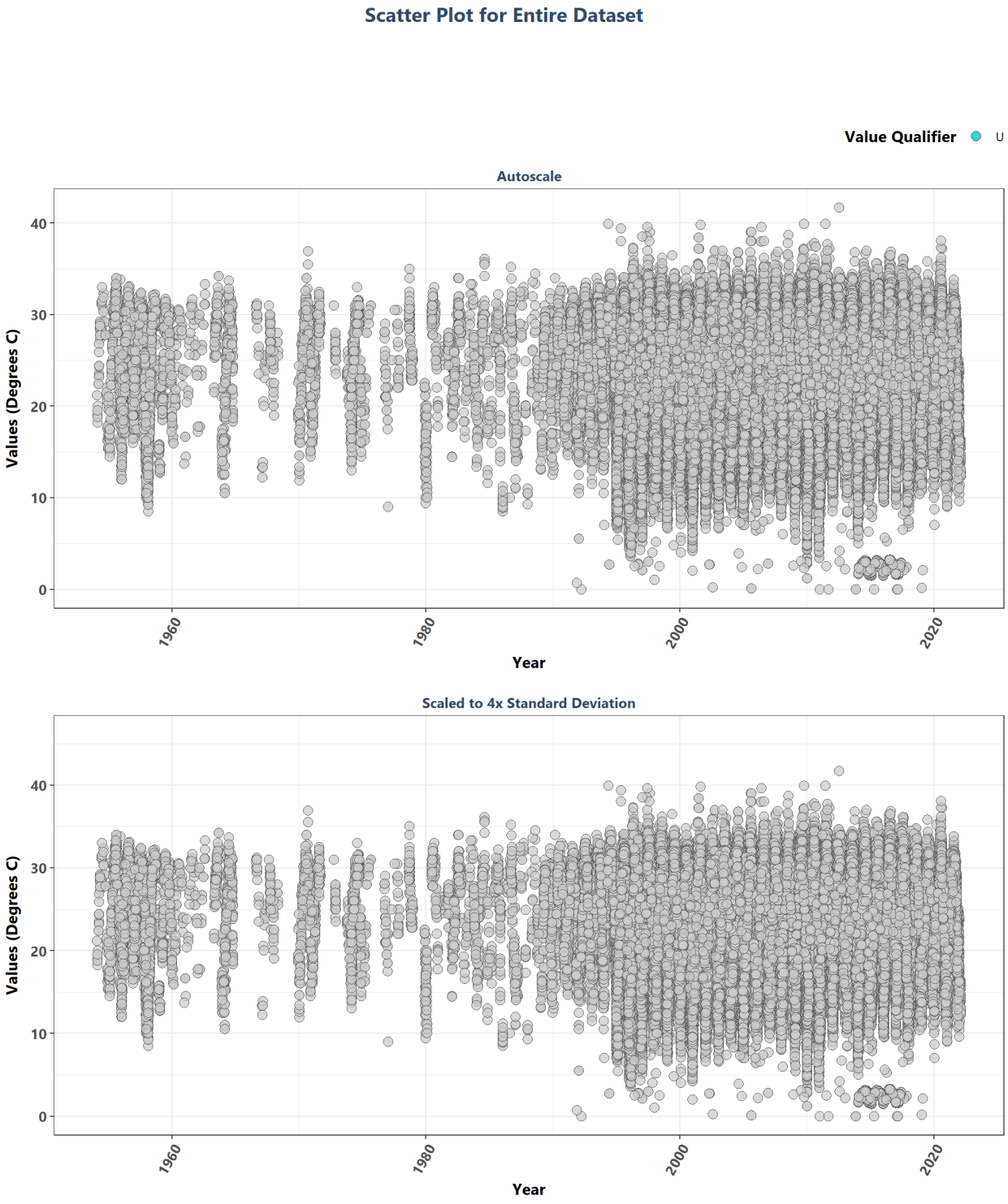
```
KT.Stats2$MonitoringID <- NULL
fwrite(KT.Stats2, paste0(out_dir,"/", param_name, "_", activity, "_", depth,
                         "_KendallTau_Stats.txt"), sep="|")
rm(KT.Stats2)
data <- data[!is.na(data$ResultValue),]
```

Appendix I: Scatter Plot of Entire Dataset

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

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

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


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

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


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

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

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

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



Appendix II: Dataset Summary Box Plots

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

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

This set of box plots are grouped by year.

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

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

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

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

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

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

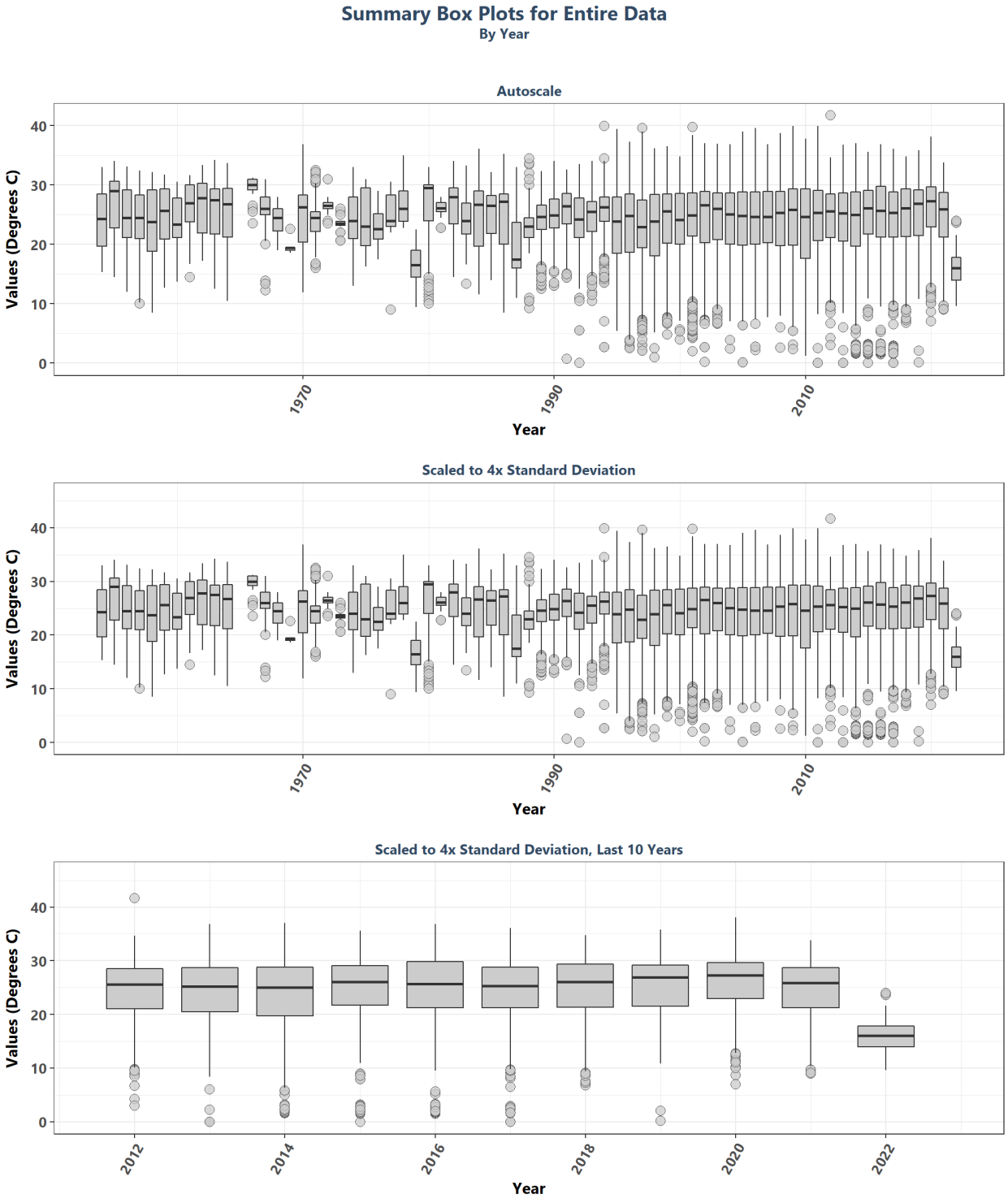
```
                      subtitle="By Year") + plot_theme +
   theme(panel.border=element_blank(), panel.grid.major=element_blank(),
         panel.grid.minor=element_blank(), axis.line=element_blank())

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

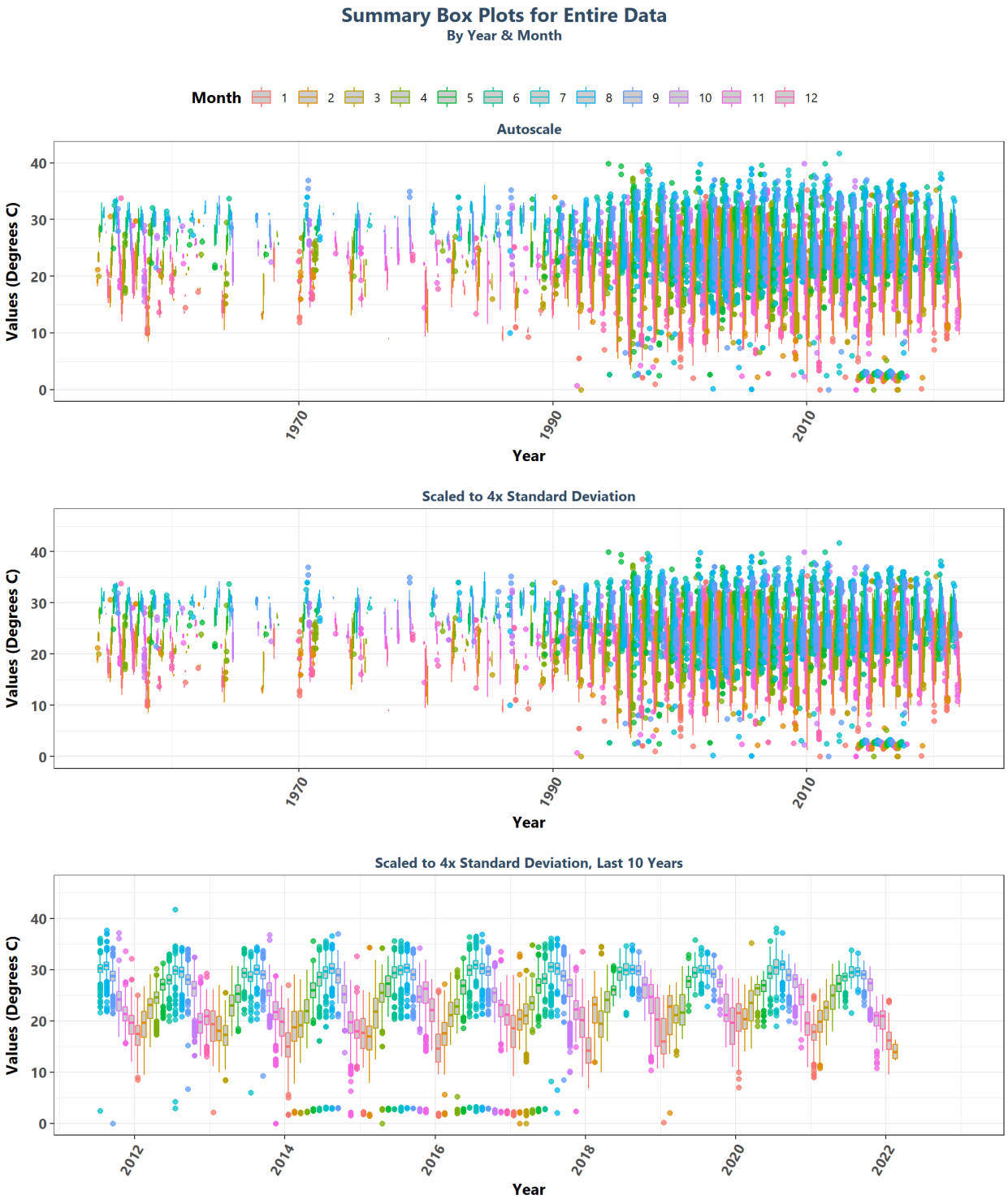
This set of box plots are grouped by year and month with the color being related to the month.

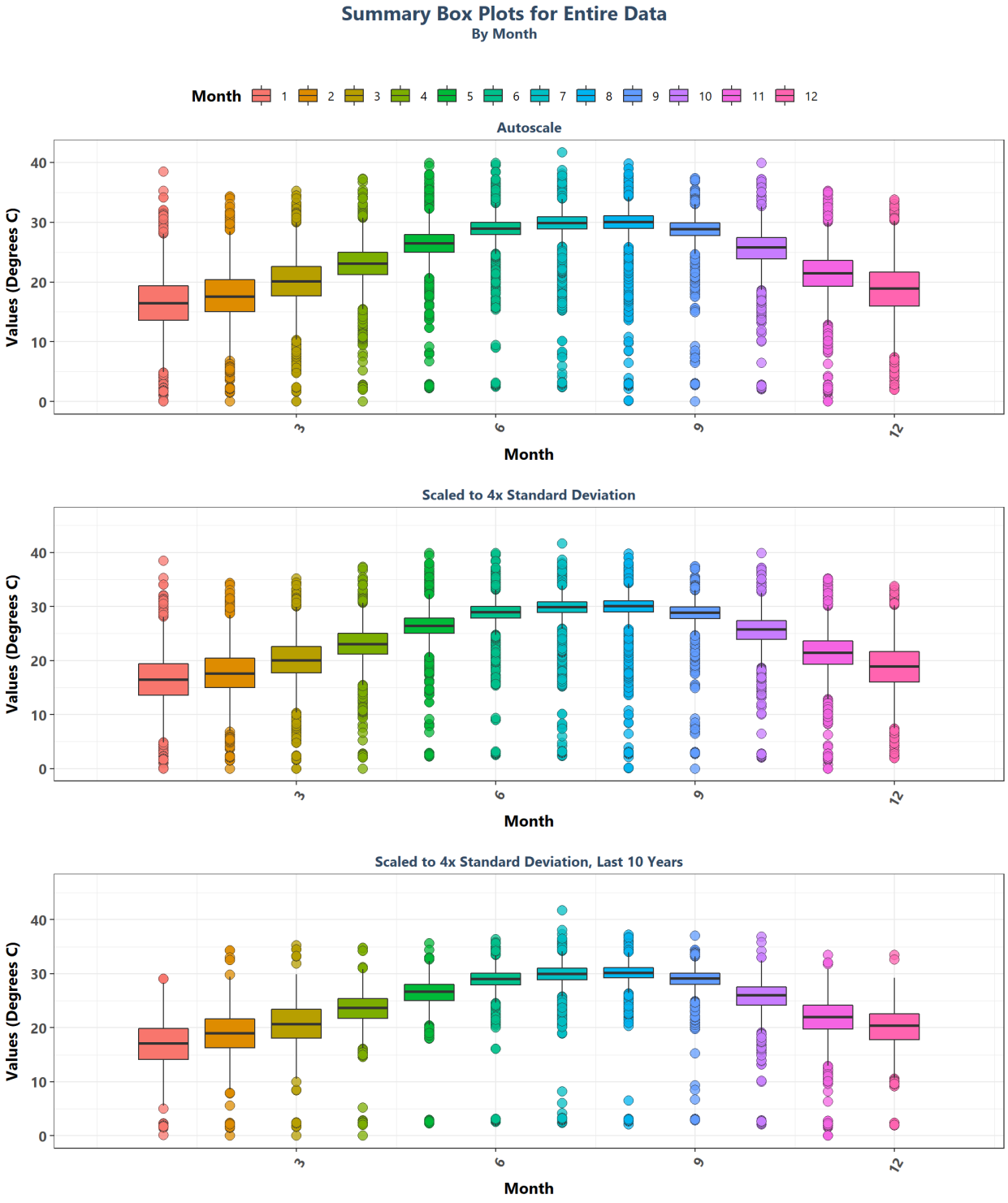
```
p1 <- ggplot(data=data[data$Include==TRUE, ],
             aes(x=YearMonthDec, y=ResultValue,
                 group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale", x="Year",
        y=paste0("Values (", unit,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

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

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

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

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

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

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

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

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

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

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


Summary Box Plots for Entire Data
By Year
Autoscale
Values (Degrees C)
0
10
20
30
40
1970
1990
2010
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
1970
1990
2010
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

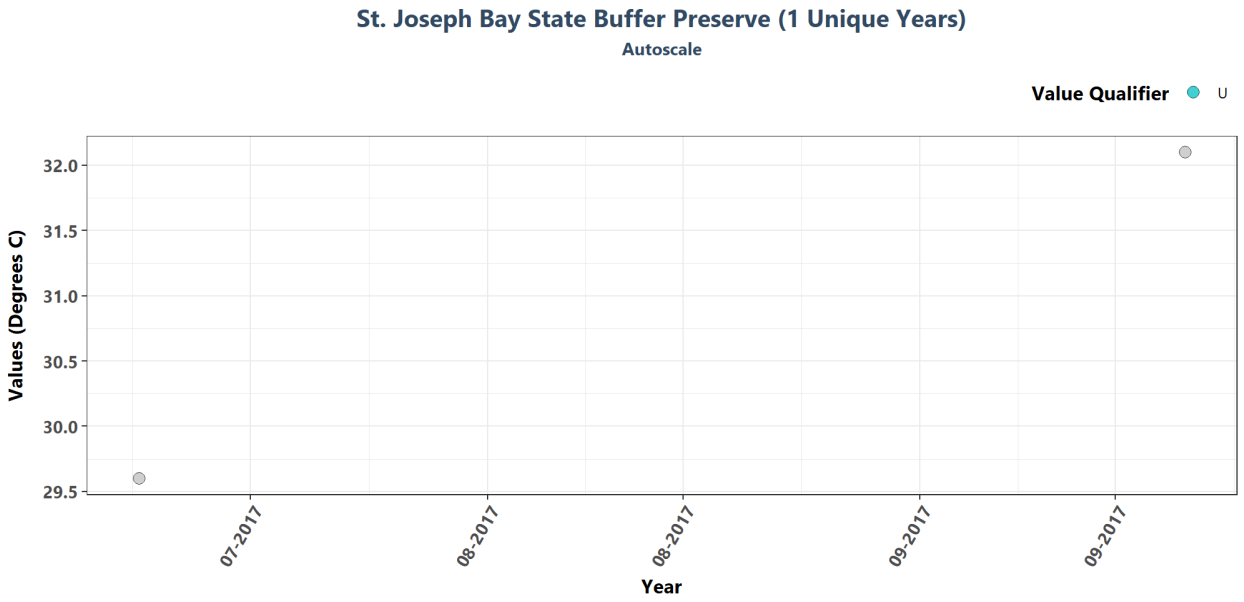

Summary Box Plots for Entire Data
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30 40
1970 1990 2010
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40
1970 1990 2010
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30 40
2012 2014 2016 2018 2020 2022
Year



Appendix III: Excluded Managed Areas

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

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



Appendix IV: Managed Area Trendlines

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

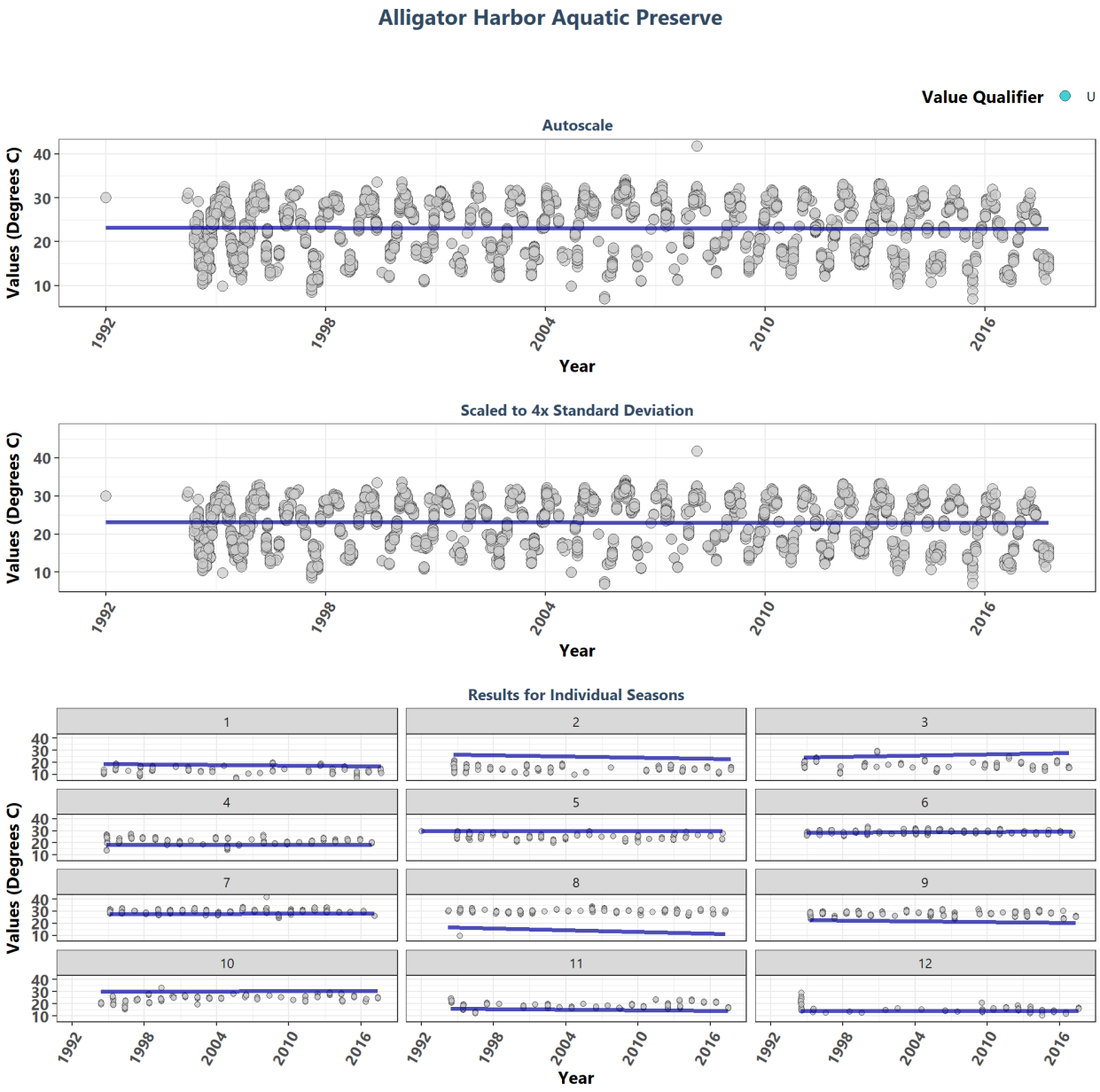
1. Use the data set that only has `SufficientData` of `TRUE` for the desired managed area
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Determine the minimum, mean, and standard deviation for the data to be used for y-axis scales
    - Excludes the top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

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

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

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

      xbrks <- seq(round_any(min(plot_data$relyear_dd), 5, floor), round_any(max(plot_data$relyear_dd),
```

```
              by = (round_any(max(plot_data$relyear_dd), 5, ceiling) - round_any(min(plot_data$rely

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

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

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

p2 <- ggplot(data=plot_data,
             aes(x=relyear_dd, y=ResultValue, fill=VQ_Plot)) +
   geom_point(shape=21, size=3, color="#333333", alpha=0.75) +
   # geom_abline(aes(slope=s_slope, intercept=s_int),
   #             color="#000099", size=1.2, alpha=0.7) +
   geom_segment(data = KT.Stats[ManagedAreaName==MA_Include[i] & Season == "All", ], aes(x = rely
                                                                                         y = relyear_
                                                                                         xend = relye
                                                                                         yend = relye
                color="#000099", size=1.2, alpha=0.7, inherit.aes = FALSE) +
   ylim(min_RV, y_scale) +
   labs(subtitle="Scaled to 4x Standard Deviation",
        x="Year", y=paste0("Values (", unit, ")")) +
```

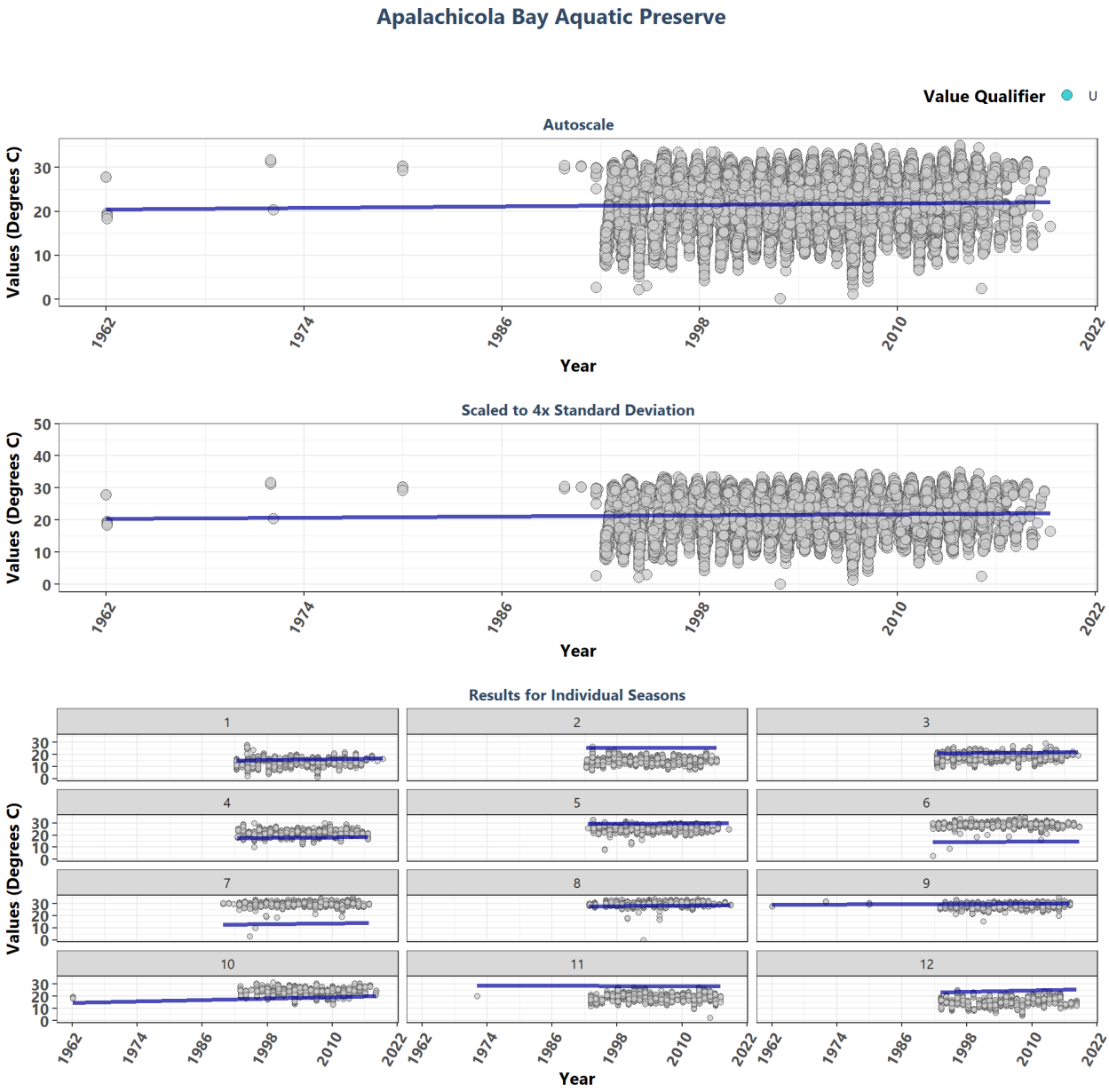
```
      plot_theme +
      theme(legend.position="none") +
      {if(inc_H==TRUE){
         scale_fill_manual(values=c("H"= "#F8766D", "U"= "#00BFC4",
                                    "HU"="#7CAE00"), na.value="#cccccc")
      } else if(param_name=="Secchi_Depth"){
         scale_fill_manual(values=c("S"=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

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

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

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

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

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

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



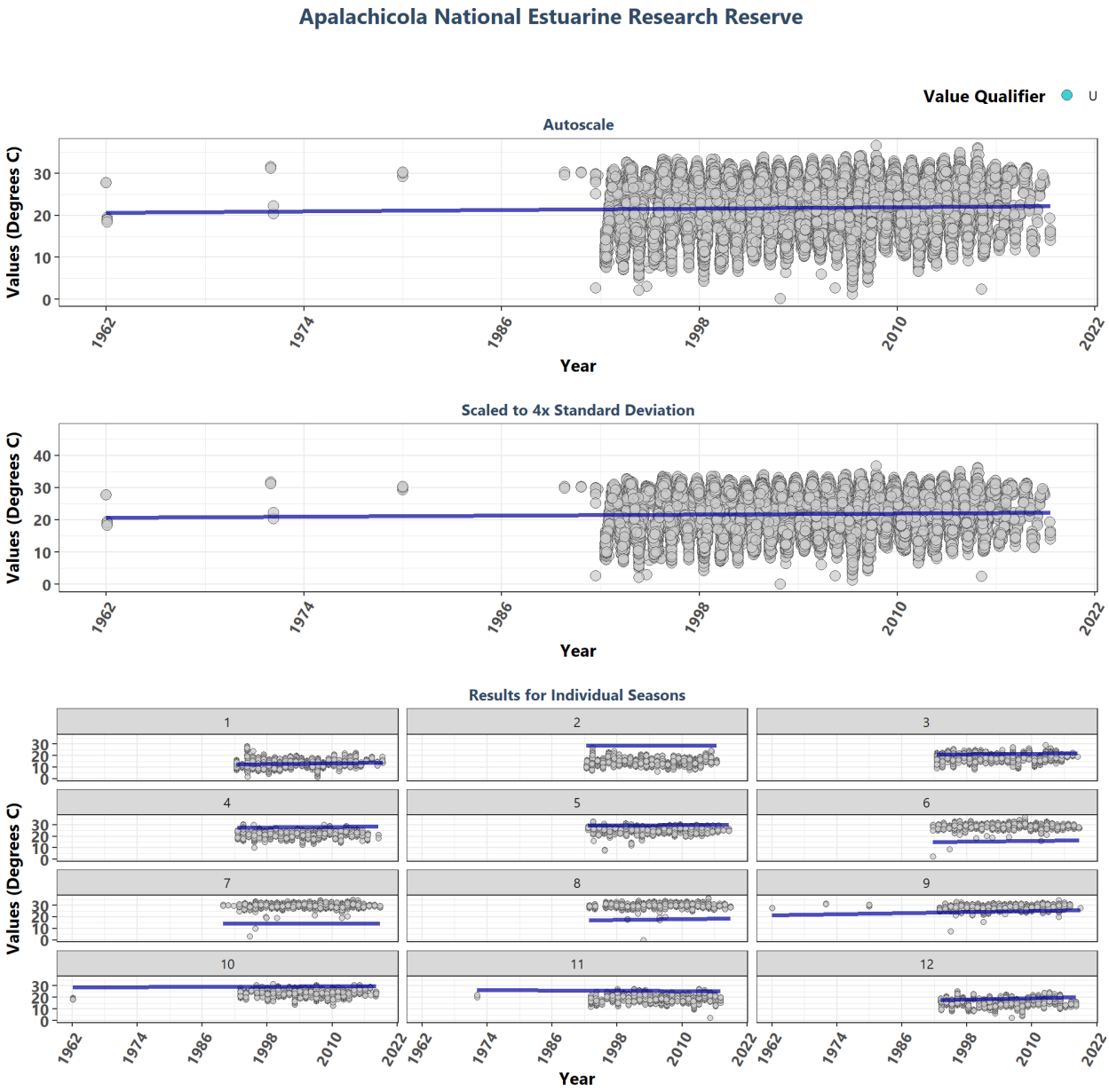
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5844	25.80	-0.0348	-0.0091	23.1859	-2.3	0.0190	331.4	0	-1
1	256	14.10	-0.1108	-0.0733	18.7233	0.3	0.7446	NA	NA	-1
2	500	15.45	-0.3140	-0.1571	26.7571	-4.7	0.0000	NA	NA	-1
3	362	18.10	0.2525	0.1833	23.4333	-3.2	0.0015	NA	NA	1
4	598	22.20	-0.0141	-0.0067	18.5600	-6.2	0.0000	NA	NA	-1
5	501	25.50	0.0231	0.0067	29.8067	-10.5	0.0000	NA	NA	1
6	722	28.90	0.1024	0.0400	28.3800	4.1	0.0000	NA	NA	1
7	740	29.90	0.0570	0.0250	27.7250	0.9	0.3466	NA	NA	1
8	469	30.20	-0.3307	-0.2529	17.4176	1.8	0.0801	NA	NA	-1
9	646	28.00	-0.1675	-0.0923	22.9385	2.2	0.0293	NA	NA	-1
10	453	24.90	0.0536	0.0176	30.0588	8.1	0.0000	NA	NA	1
11	288	18.50	-0.1404	-0.0824	16.1912	-0.4	0.7195	NA	NA	-1
12	309	15.90	0.0136	0.0091	14.0364	-9.0	0.0000	NA	NA	1

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



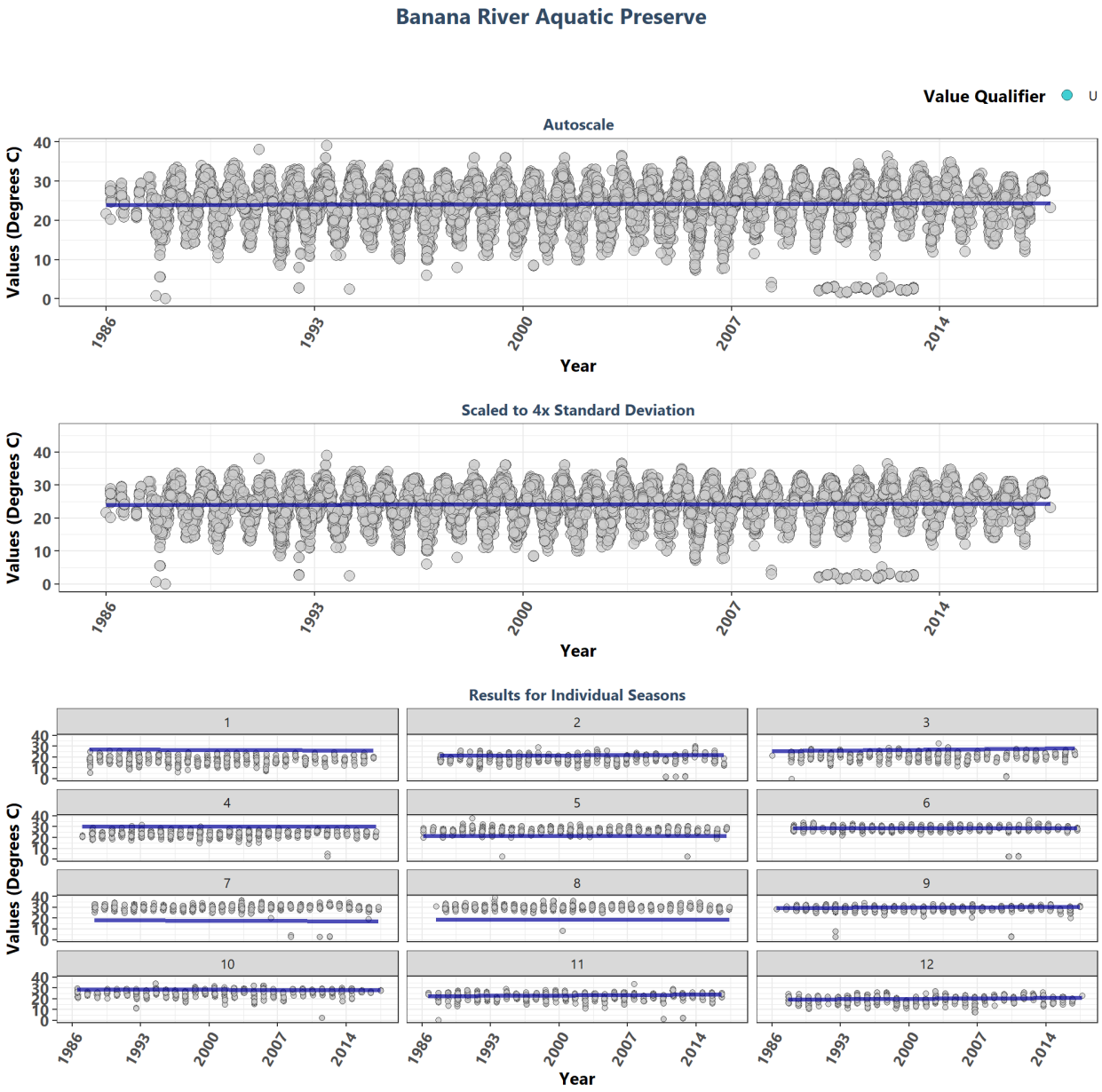
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	35896	23.8	0.0593	0.0273	20.5000	16.4	0.0000	186.6	0	1
1	2827	13.2	0.0769	0.0600	13.2200	6.5	0.0000	NA	NA	1
2	2855	14.4	-0.0055	0.0000	25.6000	2.9	0.0039	NA	NA	-1
3	3425	17.9	0.0503	0.0333	20.0667	6.2	0.0000	NA	NA	1
4	2747	21.5	0.0707	0.0500	15.8500	4.0	0.0001	NA	NA	1
5	2737	25.6	0.0321	0.0100	29.0600	-0.4	0.6647	NA	NA	1
6	3117	28.6	0.0360	0.0235	13.3882	-2.7	0.0068	NA	NA	1
7	3582	29.5	0.0817	0.0533	10.9600	2.9	0.0040	NA	NA	1
8	3529	29.7	0.0746	0.0273	27.0000	5.0	0.0000	NA	NA	1
9	3141	28.2	0.0565	0.0182	28.9182	6.3	0.0000	NA	NA	1
10	2779	24.6	0.1512	0.1000	14.7000	10.8	0.0000	NA	NA	1
11	2757	19.1	-0.0323	-0.0125	29.1500	11.9	0.0000	NA	NA	-1
12	2400	15.8	0.1368	0.0833	20.9333	5.7	0.0000	NA	NA	1

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



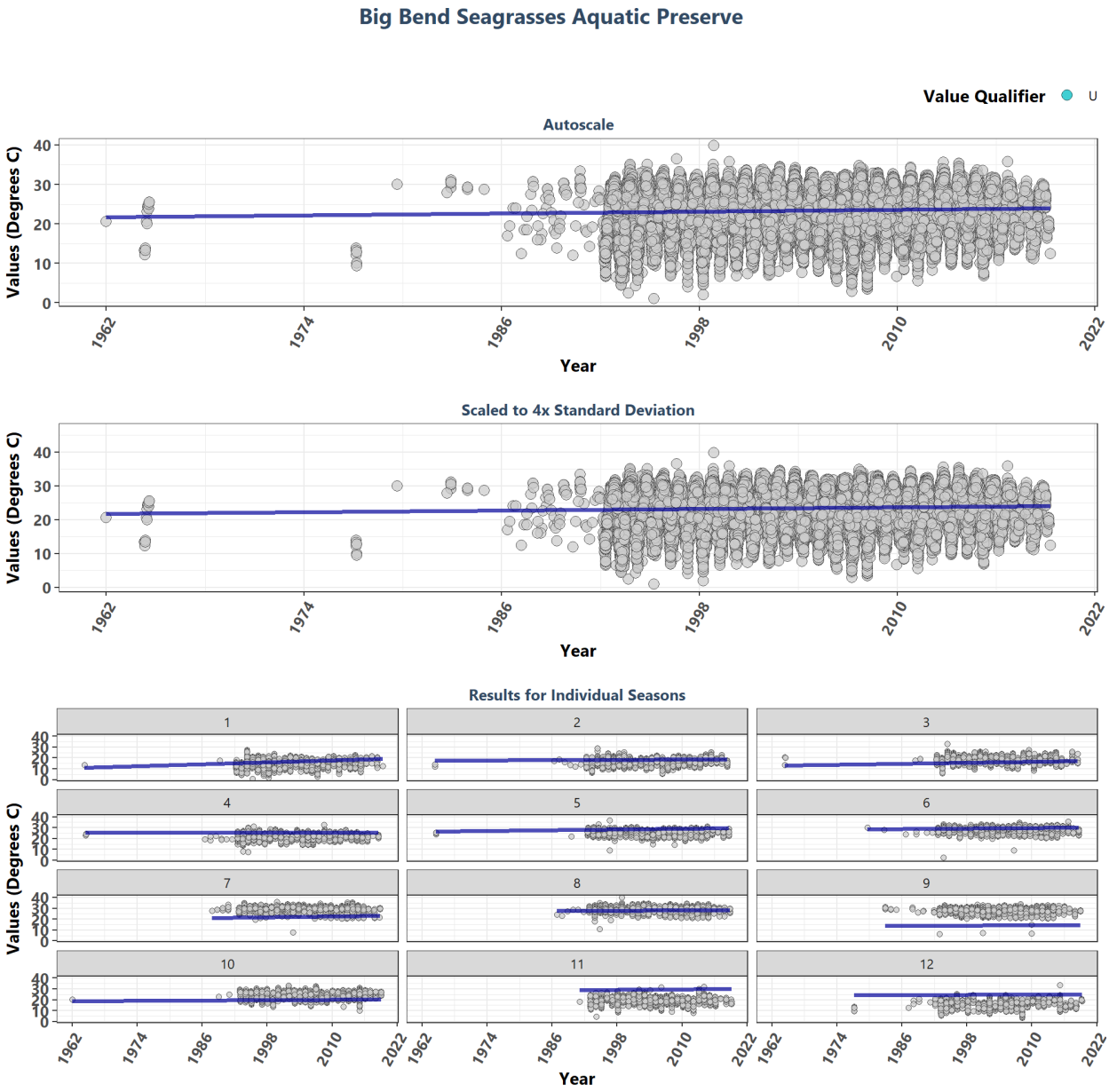
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	53965	23.2	0.0572	0.0286	20.5910	19.9	0.0000	269.2	0	1
1	4391	13.0	0.0650	0.0429	11.2000	6.5	0.0000	NA	NA	1
2	4532	14.3	-0.0051	0.0000	28.6000	1.8	0.0742	NA	NA	-1
3	5355	17.8	0.0549	0.0333	19.9667	8.0	0.0000	NA	NA	1
4	4400	21.4	0.0918	0.0333	26.6333	5.5	0.0000	NA	NA	1
5	4200	25.6	0.0327	0.0100	29.1600	-3.7	0.0002	NA	NA	1
6	4390	28.6	0.0744	0.0571	13.2429	-0.5	0.6149	NA	NA	1
7	4947	29.6	0.0177	0.0111	13.8222	3.4	0.0006	NA	NA	1
8	4870	29.7	0.0731	0.0500	15.7500	6.4	0.0000	NA	NA	1
9	4579	28.1	0.1255	0.0769	21.2154	9.3	0.0000	NA	NA	1
10	4485	24.6	0.0610	0.0200	28.8400	12.6	0.0000	NA	NA	1
11	4133	19.1	-0.0381	-0.0200	26.4800	13.0	0.0000	NA	NA	-1
12	3683	15.7	0.1346	0.1000	14.7000	6.8	0.0000	NA	NA	1

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



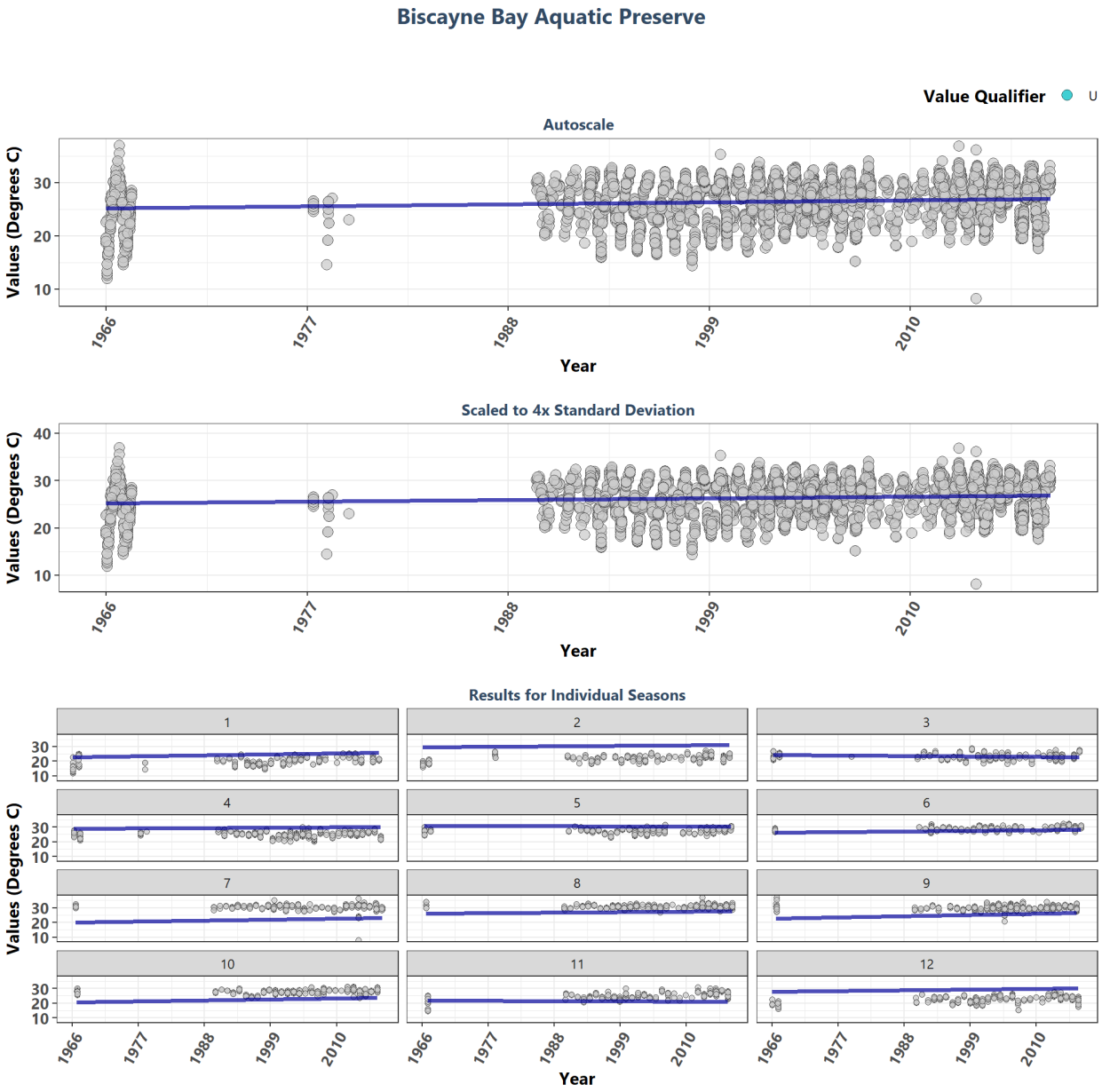
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	23394	25.40	0.0311	0.0136	23.9417	7.3	0.0000	357.5	0	1
1	1527	17.90	-0.1092	-0.0429	27.0143	-2.3	0.0198	NA	NA	-1
2	2044	18.60	0.0191	0.0091	21.3909	0.4	0.6753	NA	NA	1
3	1798	21.50	0.1808	0.0833	25.1833	-0.2	0.8407	NA	NA	1
4	2179	23.30	0.0157	0.0000	30.0000	8.0	0.0000	NA	NA	-1
5	2425	26.50	-0.0032	0.0000	21.5000	-8.1	0.0000	NA	NA	-1
6	2018	28.60	0.0094	0.0000	28.9440	-2.4	0.0160	NA	NA	-1
7	2028	30.00	-0.0397	-0.0290	18.2480	1.1	0.2882	NA	NA	-1
8	2111	29.80	0.0061	0.0000	18.6000	8.5	0.0000	NA	NA	-1
9	1734	28.94	0.1229	0.0400	29.3200	0.6	0.5556	NA	NA	1
10	2323	26.10	-0.0356	-0.0130	28.7565	13.1	0.0000	NA	NA	-1
11	1858	21.50	0.1139	0.0500	22.7000	1.2	0.2153	NA	NA	1
12	1349	20.00	0.0794	0.0486	19.4657	4.4	0.0000	NA	NA	1

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



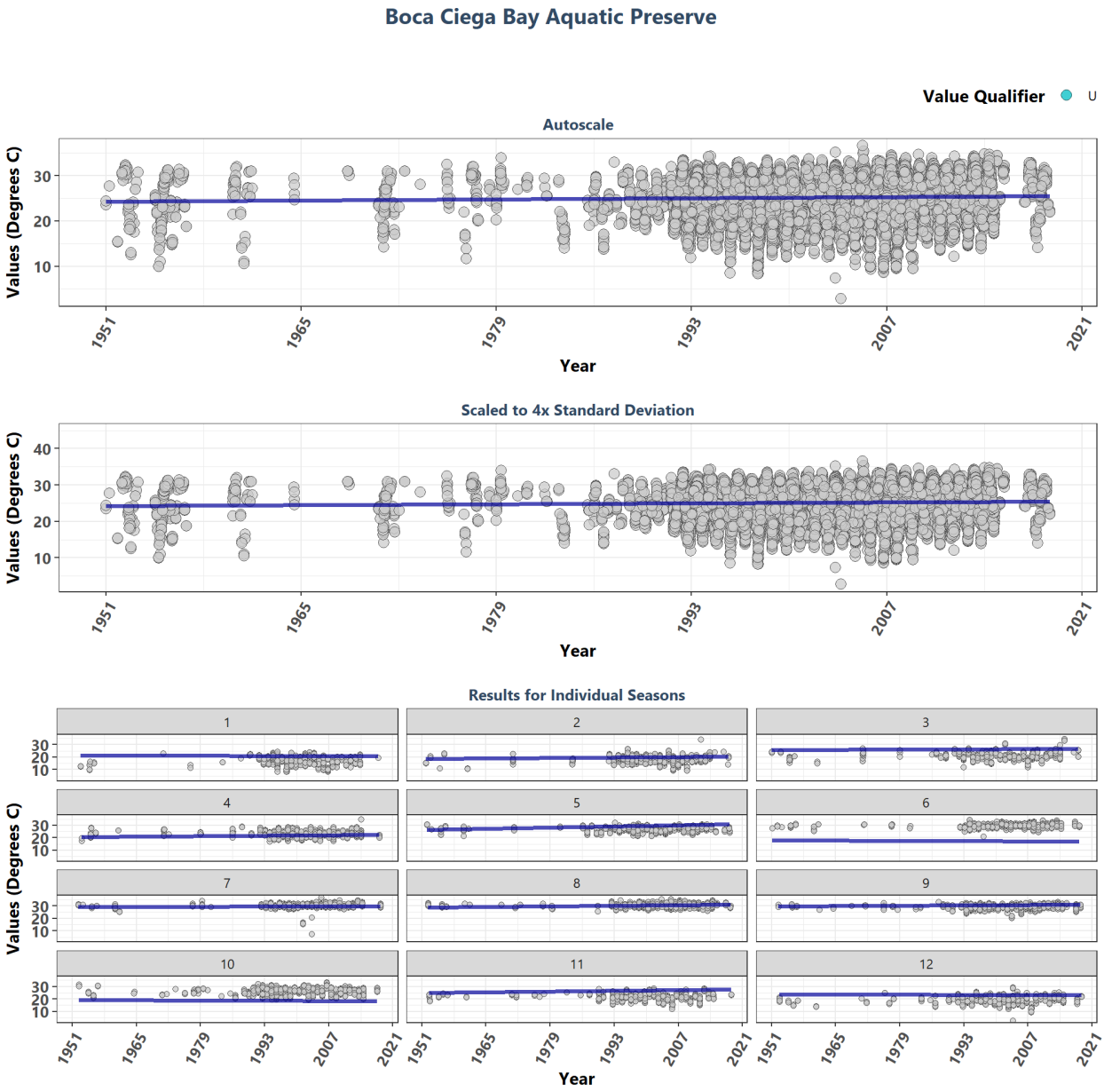
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	96944	23.8	0.0802	0.0389	21.7567	37.1	0.0000	754	0	1
1	6915	14.4	0.1781	0.1500	10.4500	2.6	0.0100	NA	NA	1
2	7111	16.0	0.0255	0.0182	17.8182	13.6	0.0000	NA	NA	1
3	8749	18.6	0.1071	0.0750	12.8500	3.6	0.0003	NA	NA	1
4	8658	22.0	-0.0021	0.0000	25.7000	17.6	0.0000	NA	NA	-1
5	8277	25.7	0.1598	0.0583	26.1917	-0.3	0.7725	NA	NA	1
6	7819	28.7	0.1164	0.0400	27.9800	21.2	0.0000	NA	NA	1
7	8220	29.6	0.1262	0.0667	19.2000	17.2	0.0000	NA	NA	1
8	8979	29.7	0.0675	0.0222	27.1444	16.6	0.0000	NA	NA	1
9	8758	28.1	0.0206	0.0167	13.7000	9.5	0.0000	NA	NA	1
10	8651	24.6	0.0315	0.0210	18.9970	1.7	0.0830	NA	NA	1
11	7502	19.9	0.1263	0.0400	27.8800	4.1	0.0000	NA	NA	1
12	7305	16.6	0.0124	0.0067	24.3133	22.9	0.0000	NA	NA	1

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



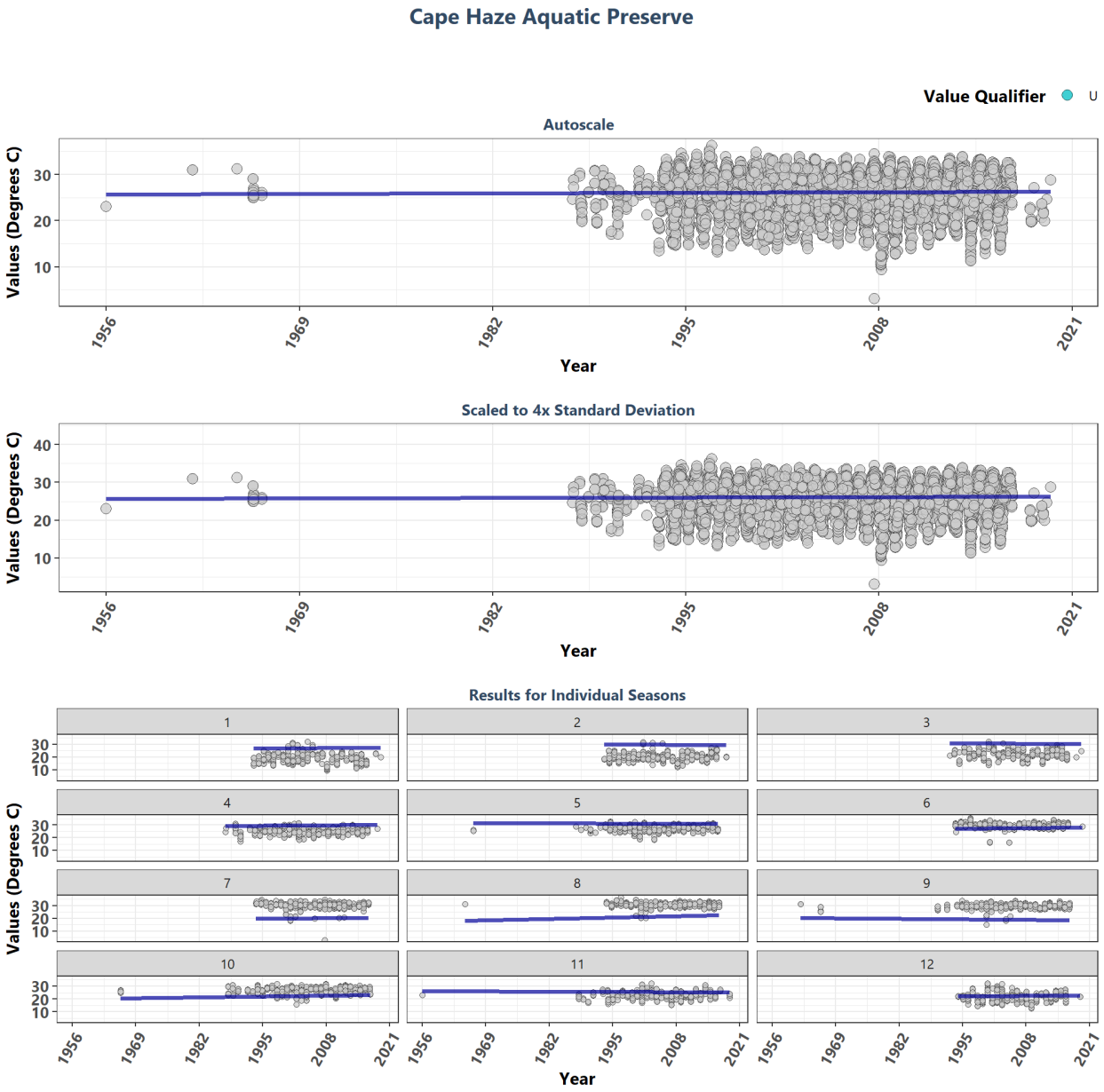
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	6948	27.11	0.1310	0.0338	25.1730	15.4	0.0000	226	0	1
1	411	21.25	0.1680	0.0605	22.8395	-1.0	0.3291	NA	NA	1
2	414	22.40	0.1506	0.0273	29.6818	8.4	0.0000	NA	NA	1
3	405	23.04	-0.1465	-0.0318	24.2158	-4.4	0.0000	NA	NA	-1
4	902	25.20	0.0991	0.0250	28.9400	7.6	0.0000	NA	NA	1
5	602	27.30	-0.0312	-0.0067	30.6667	4.5	0.0000	NA	NA	-1
6	450	29.27	0.1652	0.0387	26.1303	8.8	0.0000	NA	NA	1
7	525	30.40	0.2731	0.0638	19.9122	-1.1	0.2820	NA	NA	1
8	533	30.80	0.1226	0.0286	26.2143	5.2	0.0000	NA	NA	1
9	1088	29.91	0.3130	0.0789	22.4396	4.9	0.0000	NA	NA	1
10	612	27.56	0.1767	0.0576	20.5994	6.1	0.0000	NA	NA	1
11	506	25.59	-0.0321	-0.0111	21.6611	10.6	0.0000	NA	NA	-1
12	500	22.79	0.2770	0.0455	27.5842	5.9	0.0000	NA	NA	1

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



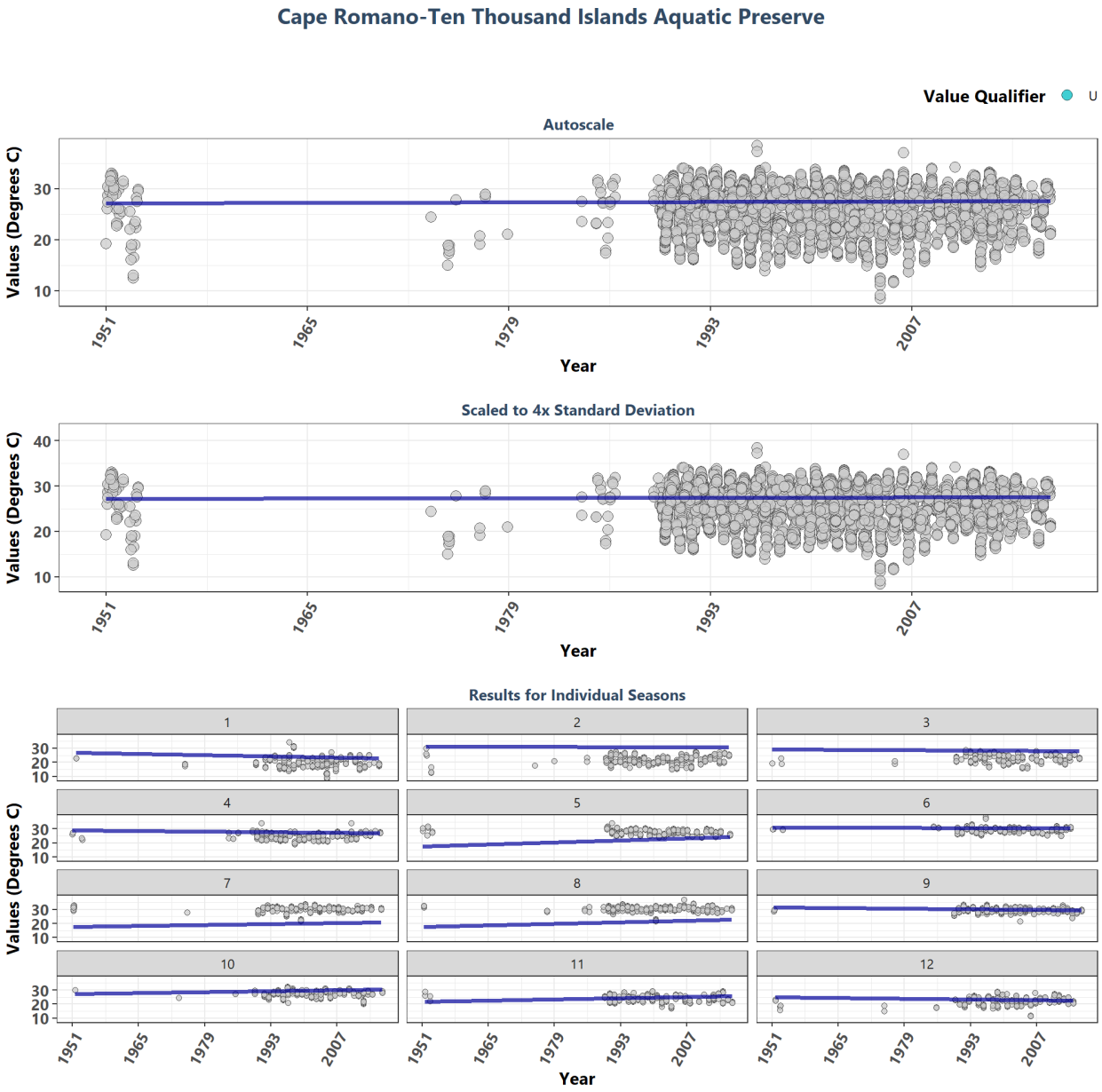
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	17567	26.26	0.0560	0.0178	24.2620	10.1	0.0000	214.6	0	1
1	1239	17.20	-0.0200	-0.0093	21.3436	-1.4	0.1546	NA	NA	-1
2	1202	18.35	0.0397	0.0222	18.8222	-1.2	0.2447	NA	NA	1
3	1195	20.87	0.0375	0.0158	25.6729	-1.0	0.3005	NA	NA	1
4	1092	23.30	0.0658	0.0250	20.5250	-0.6	0.5515	NA	NA	1
5	1604	27.12	0.2589	0.0667	26.1333	7.6	0.0000	NA	NA	1
6	1180	29.60	-0.0270	-0.0180	18.1180	13.3	0.0000	NA	NA	-1
7	1228	30.53	0.0238	0.0059	29.2085	6.7	0.0000	NA	NA	1
8	1516	30.65	0.1273	0.0308	28.8992	5.2	0.0000	NA	NA	1
9	2542	29.48	0.0887	0.0233	29.3900	1.8	0.0705	NA	NA	1
10	2075	26.40	-0.0224	-0.0127	18.9991	2.6	0.0103	NA	NA	-1
11	1376	21.80	0.1265	0.0413	24.9750	3.7	0.0002	NA	NA	1
12	1318	20.00	-0.0120	-0.0049	23.5489	2.2	0.0307	NA	NA	-1

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



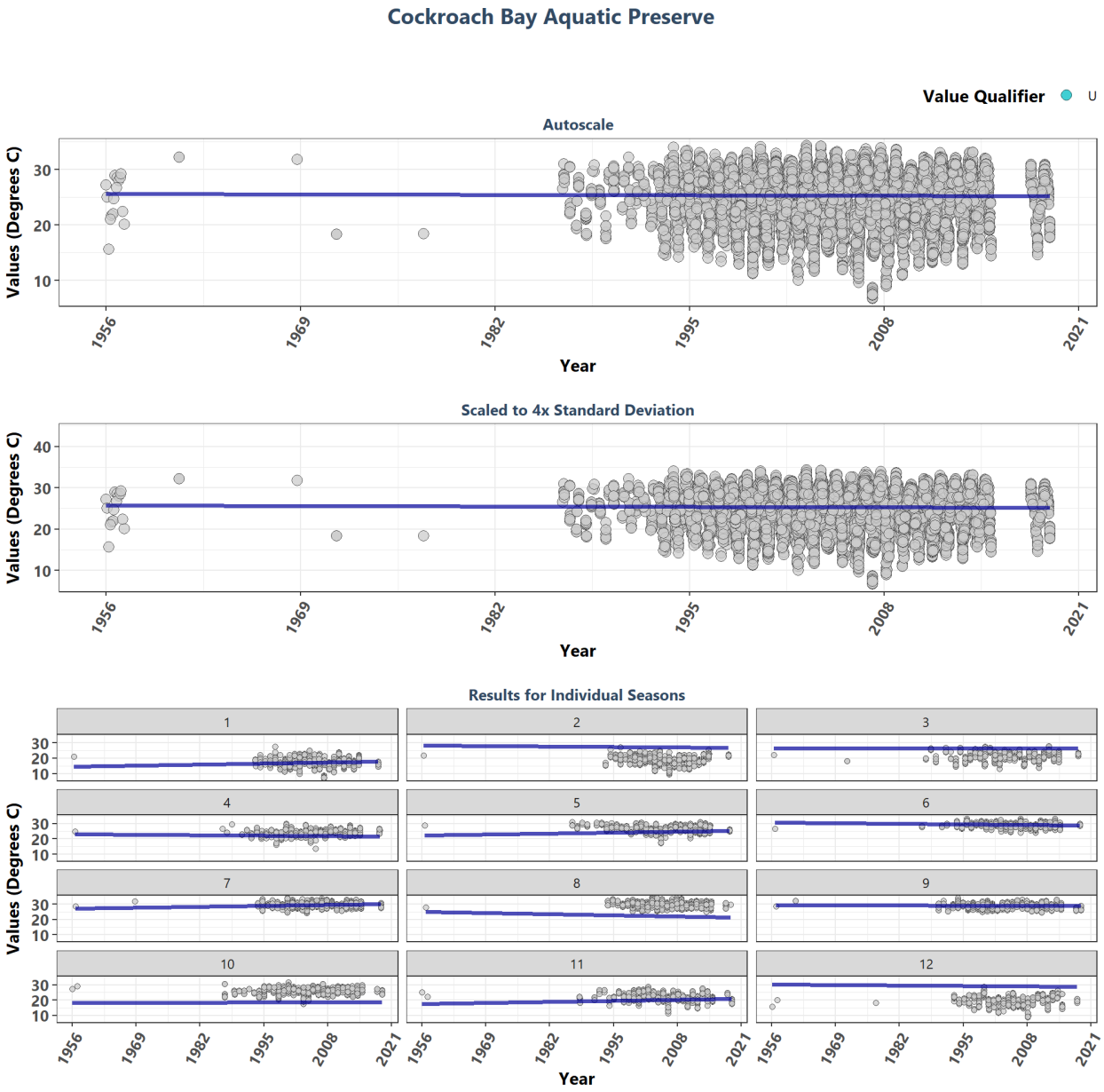
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	7414	26.80	0.0249	0.0100	25.6681	3.0	0.0026	60.3	0	1
1	478	18.80	0.0439	0.0167	26.3500	-1.3	0.1927	NA	NA	1
2	526	20.10	-0.0484	-0.0182	30.6273	0.9	0.3422	NA	NA	-1
3	528	22.50	-0.0554	-0.0200	31.5600	2.2	0.0276	NA	NA	-1
4	622	25.38	0.1077	0.0321	28.0286	-0.9	0.3656	NA	NA	1
5	674	27.40	-0.0268	-0.0088	31.3637	3.8	0.0001	NA	NA	-1
6	570	29.70	0.0989	0.0389	25.4167	-1.7	0.0839	NA	NA	1
7	809	30.50	0.0277	0.0174	19.2103	-2.4	0.0182	NA	NA	1
8	649	30.90	0.1090	0.0857	17.1429	-1.0	0.3055	NA	NA	1
9	667	29.70	-0.0398	-0.0333	20.4667	4.2	0.0000	NA	NA	-1
10	715	27.20	0.0640	0.0500	19.9000	1.8	0.0789	NA	NA	1
11	643	22.50	-0.0242	-0.0111	25.9194	1.9	0.0585	NA	NA	-1
12	533	21.60	0.0498	0.0263	21.1316	3.8	0.0002	NA	NA	1

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



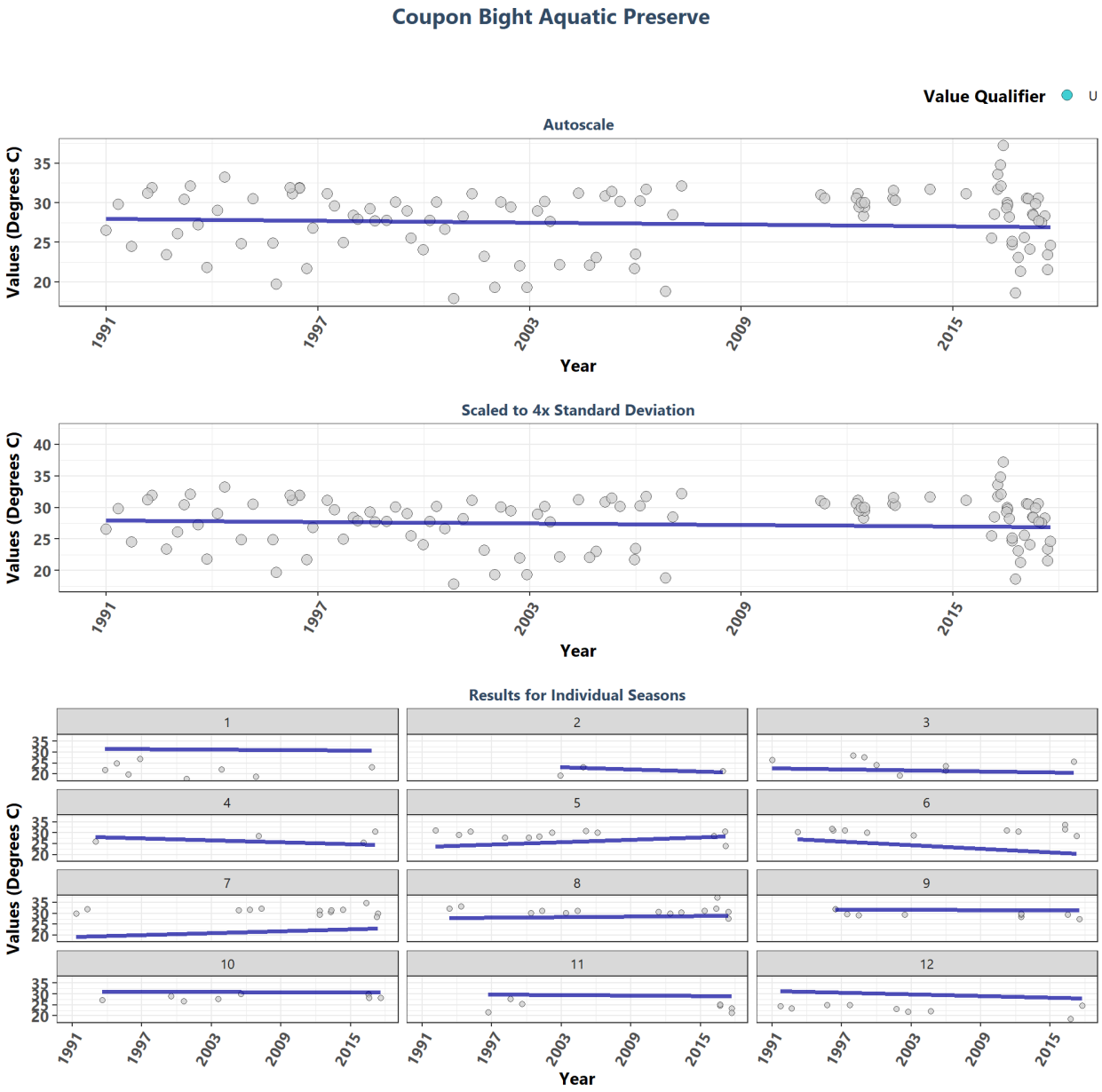
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	6835	26.50	0.0129	0.0059	27.1326	1.9	0.0520	163.1	0	0
1	558	19.80	-0.1352	-0.0630	27.0240	2.3	0.0192	NA	NA	0
2	628	21.60	-0.0303	-0.0077	31.2019	4.6	0.0000	NA	NA	0
3	473	23.10	-0.0632	-0.0223	29.2936	-2.3	0.0214	NA	NA	0
4	561	24.60	-0.0718	-0.0330	28.9500	4.3	0.0000	NA	NA	0
5	559	28.18	0.1562	0.0976	17.6429	-2.2	0.0252	NA	NA	0
6	585	29.50	-0.0151	-0.0031	30.7157	5.5	0.0000	NA	NA	0
7	533	30.56	0.0662	0.0483	17.4800	-0.5	0.6027	NA	NA	0
8	636	30.83	0.1237	0.0800	17.6400	-1.1	0.2527	NA	NA	0
9	575	29.96	-0.1143	-0.0335	31.6015	-4.1	0.0000	NA	NA	0
10	619	27.30	0.1506	0.0471	27.2412	-2.7	0.0074	NA	NA	0
11	513	24.00	0.1219	0.0579	21.8211	-4.6	0.0000	NA	NA	0
12	595	22.33	-0.0707	-0.0400	25.0200	5.7	0.0000	NA	NA	0

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



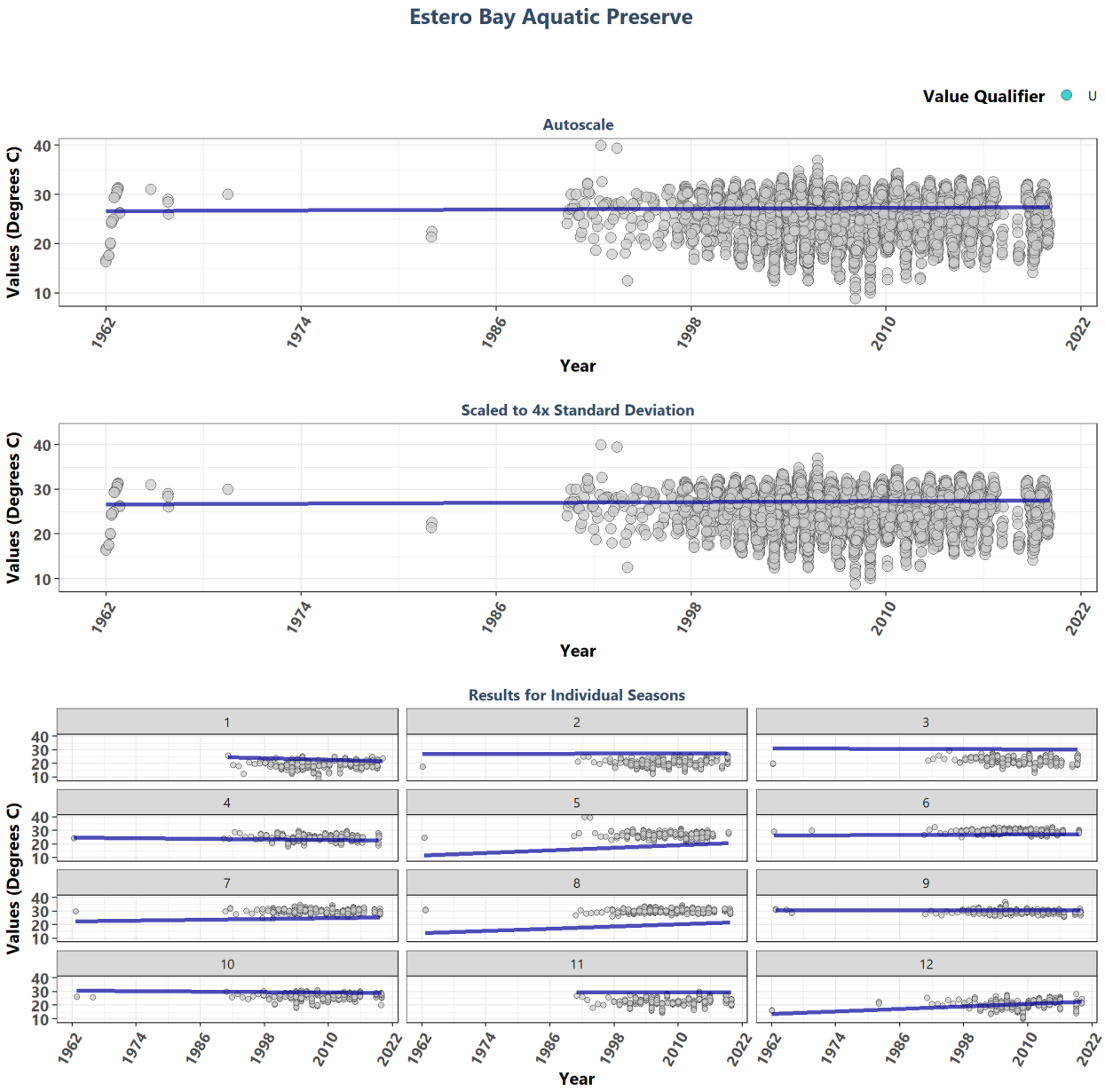
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	14863	26.30	-0.0058	-0.0073	25.6756	-2.7	0.0075	139.8	0	-1
1	977	17.22	0.0710	0.0556	14.3867	3.3	0.0009	NA	NA	1
2	980	18.68	-0.0594	-0.0250	28.2600	0.5	0.5929	NA	NA	-1
3	1059	21.80	-0.0076	0.0000	26.3950	-1.8	0.0795	NA	NA	-1
4	1027	24.60	-0.0359	-0.0250	23.0500	4.5	0.0000	NA	NA	-1
5	1207	27.01	0.0939	0.0457	22.2686	-3.1	0.0020	NA	NA	1
6	1018	29.13	-0.0350	-0.0238	30.5025	-1.1	0.2834	NA	NA	-1
7	1130	29.50	0.1152	0.0508	26.9838	5.8	0.0000	NA	NA	1
8	2074	29.41	-0.0951	-0.0562	24.9562	-2.4	0.0151	NA	NA	-1
9	1885	29.15	-0.0224	-0.0077	29.5273	-4.6	0.0000	NA	NA	-1
10	1238	26.40	0.0114	0.0075	18.2975	-0.4	0.6885	NA	NA	1
11	1202	22.20	0.0620	0.0517	17.4683	-4.9	0.0000	NA	NA	1
12	1066	20.00	-0.0693	-0.0258	30.3125	3.0	0.0024	NA	NA	-1

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



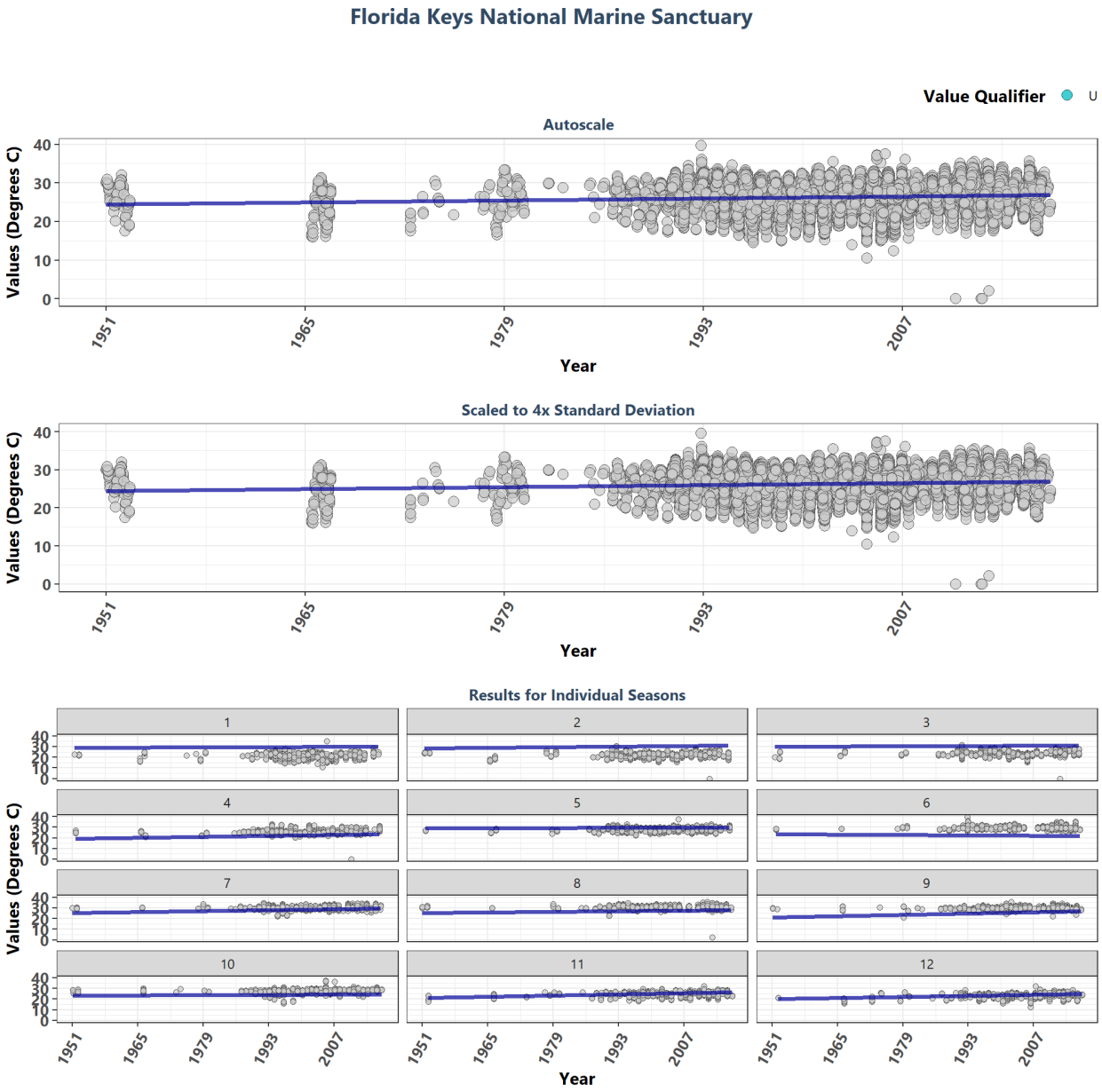
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	108	28.95	-0.1481	-0.0390	27.9680	-2.0	0.0436	11.3	0.4215	-1
1	8	21.97	-0.0879	-0.0315	31.4974	-0.1	0.9015	NA	NA	-1
2	3	21.30	-0.3333	-0.1628	25.0276	0.0	1.0000	NA	NA	-1
3	8	24.82	-0.0714	-0.0804	22.6113	-1.2	0.2125	NA	NA	-1
4	4	27.30	-0.4286	-0.1433	28.2833	0.3	0.7341	NA	NA	-1
5	12	29.54	0.3333	0.1792	23.6230	-0.5	0.5824	NA	NA	1
6	11	31.00	-0.3929	-0.2738	27.5558	-0.3	0.7540	NA	NA	-1
7	13	31.44	0.3333	0.1424	19.3070	-0.5	0.6236	NA	NA	1
8	14	30.85	0.3056	0.0462	27.6526	-0.4	0.7003	NA	NA	1
9	10	29.44	-0.1154	-0.0169	31.7927	-2.4	0.0178	NA	NA	-1
10	9	28.30	-0.0909	-0.0100	31.1200	1.1	0.2873	NA	NA	-1
11	7	24.70	-0.1364	-0.0291	29.8410	-1.2	0.2189	NA	NA	-1
12	9	23.40	-0.6000	-0.1280	31.4929	-1.1	0.2515	NA	NA	-1

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



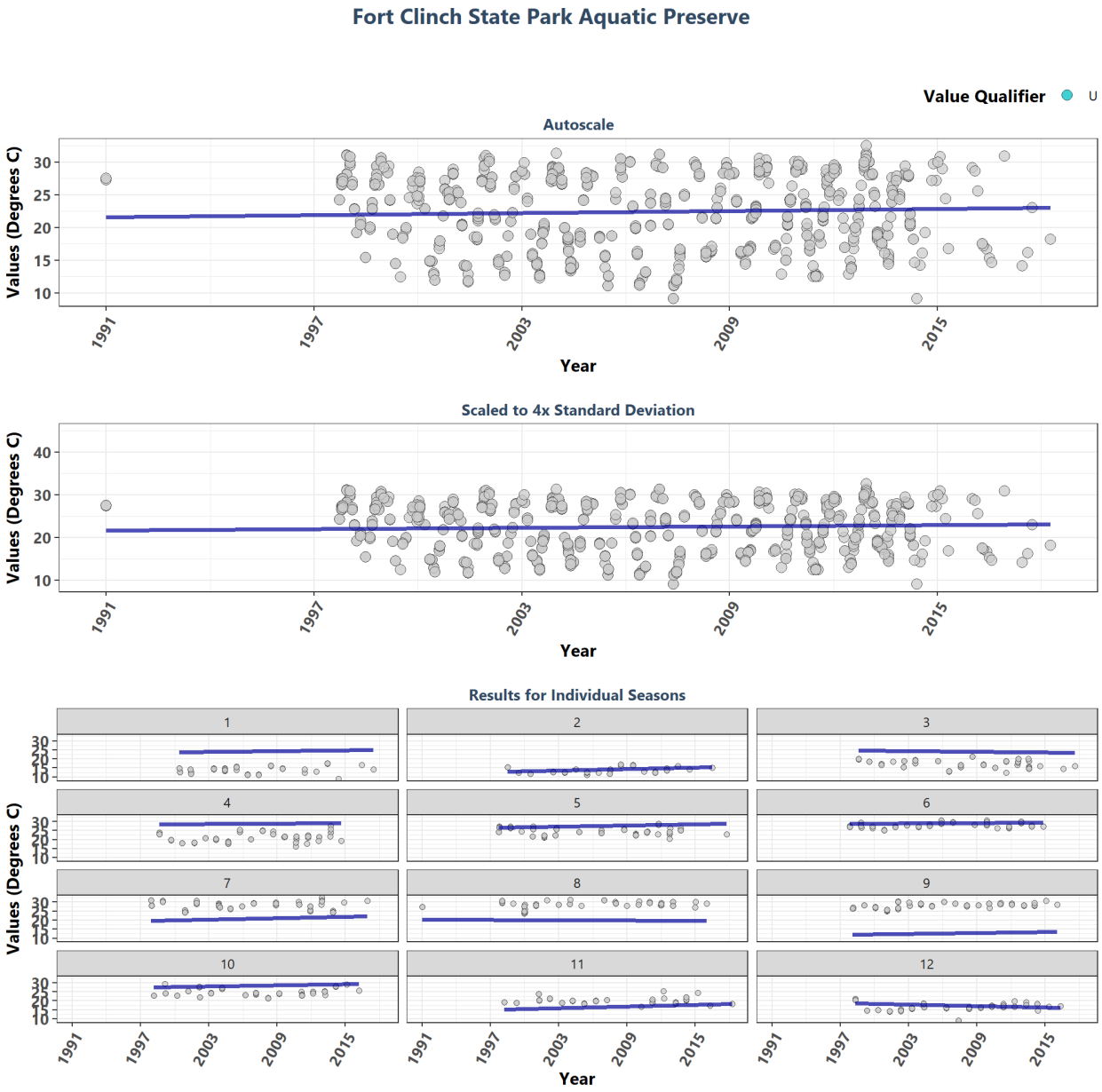
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7684	26.20	0.0298	0.0133	26.6740	3.6	0.0004	167.5	0	1
1	568	18.70	-0.1161	-0.0944	27.1500	7.4	0.0000	NA	NA	-1
2	598	19.98	0.0160	0.0089	26.9000	6.0	0.0000	NA	NA	1
3	631	22.90	-0.0352	-0.0118	30.9294	-4.4	0.0000	NA	NA	-1
4	629	24.80	-0.0586	-0.0400	24.9600	3.3	0.0009	NA	NA	-1
5	658	27.30	0.2061	0.1556	11.7000	0.6	0.5361	NA	NA	1
6	654	29.70	0.0247	0.0120	26.4480	-2.5	0.0128	NA	NA	1
7	739	30.40	0.0882	0.0500	22.5500	-1.4	0.1501	NA	NA	1
8	657	30.70	0.1644	0.1333	13.9800	-0.6	0.5783	NA	NA	1
9	638	29.59	-0.0145	0.0000	30.7000	-0.2	0.8735	NA	NA	-1
10	698	27.00	-0.0648	-0.0250	30.8250	1.0	0.3261	NA	NA	-1
11	574	23.20	-0.0042	0.0000	29.5900	-2.1	0.0350	NA	NA	-1
12	640	20.80	0.1840	0.1545	13.8455	7.0	0.0000	NA	NA	1

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



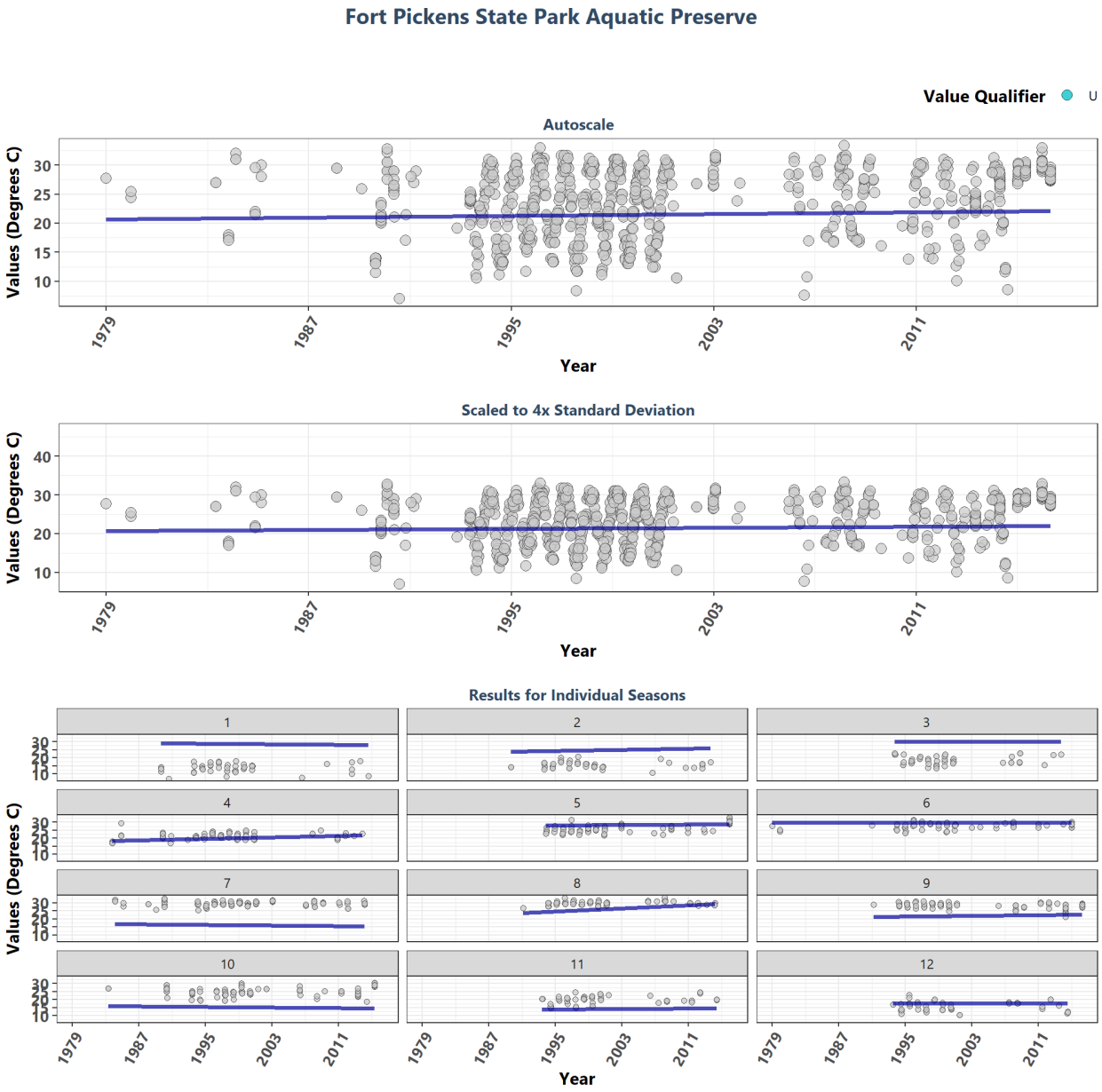
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	23997	27.05	0.1337	0.0383	24.3714	30.8	0.0000	474.4	0	1
1	1674	22.44	0.0770	0.0173	29.0740	-3.5	0.0004	NA	NA	1
2	2152	22.90	0.1470	0.0340	28.6654	11.3	0.0000	NA	NA	1
3	2050	24.30	0.0771	0.0136	30.1435	3.7	0.0002	NA	NA	1
4	1837	25.70	0.1626	0.0667	19.0333	16.2	0.0000	NA	NA	1
5	2613	27.58	0.0270	0.0061	29.0813	9.5	0.0000	NA	NA	1
6	1626	29.40	-0.0577	-0.0226	23.5971	1.6	0.1014	NA	NA	-1
7	2223	30.40	0.2771	0.0609	25.1457	10.4	0.0000	NA	NA	1
8	1730	30.85	0.1244	0.0396	25.4395	4.8	0.0000	NA	NA	1
9	2476	30.01	0.2508	0.0867	21.1933	5.8	0.0000	NA	NA	1
10	1883	28.25	0.0551	0.0170	23.4164	18.1	0.0000	NA	NA	1
11	1945	25.30	0.2420	0.0812	20.9944	16.0	0.0000	NA	NA	1
12	1788	24.00	0.2065	0.0714	20.0034	13.1	0.0000	NA	NA	1

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



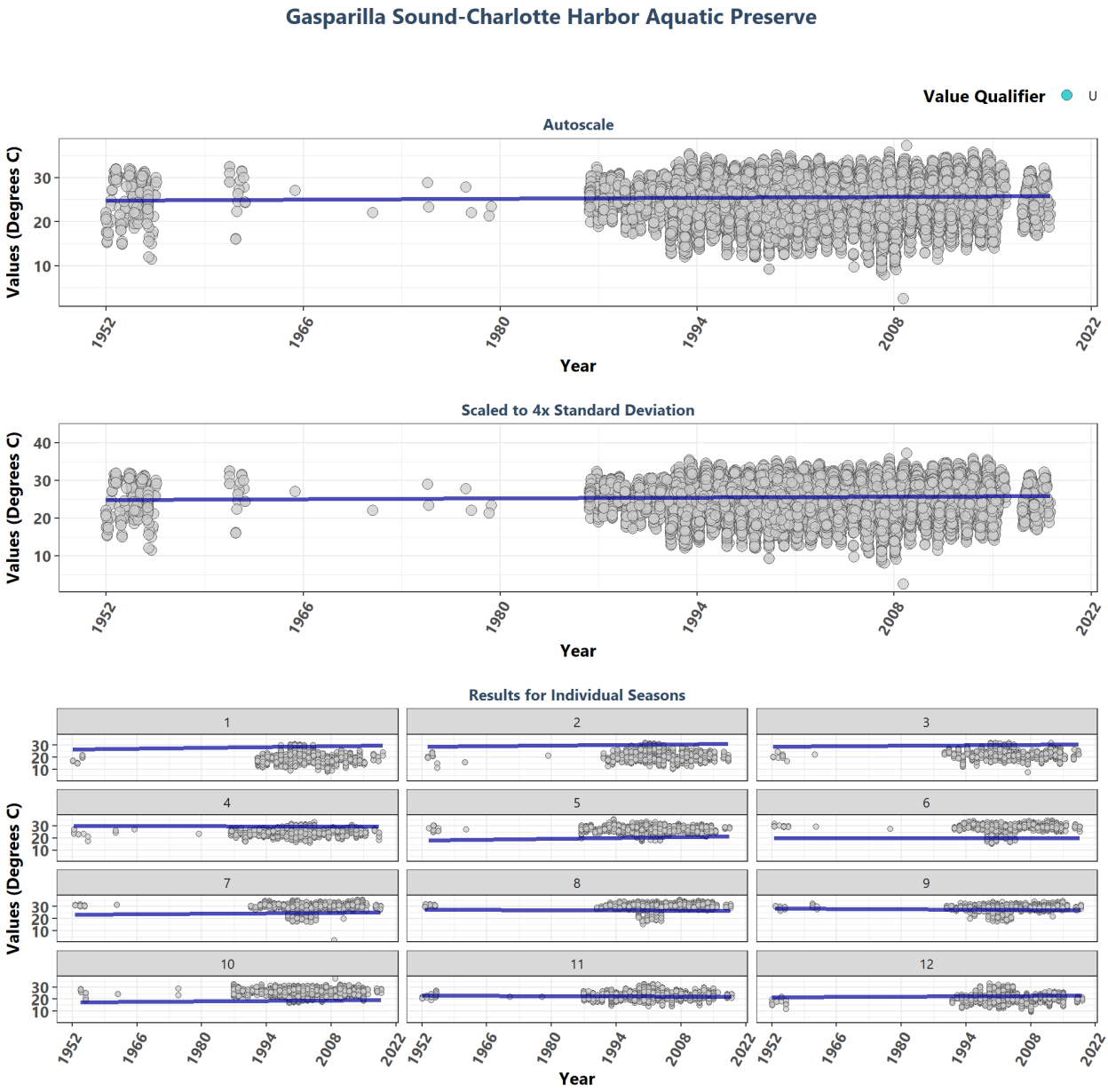
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	813	23.60	0.1011	0.0536	21.5950	4.1	0.0000	40.2	0	1
1	50	14.10	0.1560	0.0750	23.0000	1.2	0.2445	NA	NA	1
2	55	12.90	0.1135	0.1400	12.0000	2.2	0.0282	NA	NA	1
3	68	17.20	-0.1078	-0.0692	25.3077	-2.3	0.0198	NA	NA	-1
4	75	21.10	0.0510	0.0375	28.0750	2.2	0.0286	NA	NA	1
5	79	24.20	0.2469	0.1000	26.1000	-1.4	0.1589	NA	NA	1
6	74	27.45	0.1333	0.0444	28.4389	3.1	0.0018	NA	NA	1
7	79	28.60	0.1719	0.1333	18.8333	0.7	0.5063	NA	NA	1
8	60	29.15	-0.0794	-0.0200	20.1900	1.5	0.1317	NA	NA	-1
9	71	28.20	0.2027	0.1000	11.2000	3.8	0.0001	NA	NA	1
10	67	24.20	0.3111	0.1083	26.5750	1.9	0.0613	NA	NA	1
11	64	19.80	0.2298	0.1500	14.1500	-0.9	0.3521	NA	NA	1
12	71	16.40	-0.1927	-0.1286	19.5143	2.8	0.0045	NA	NA	-1

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



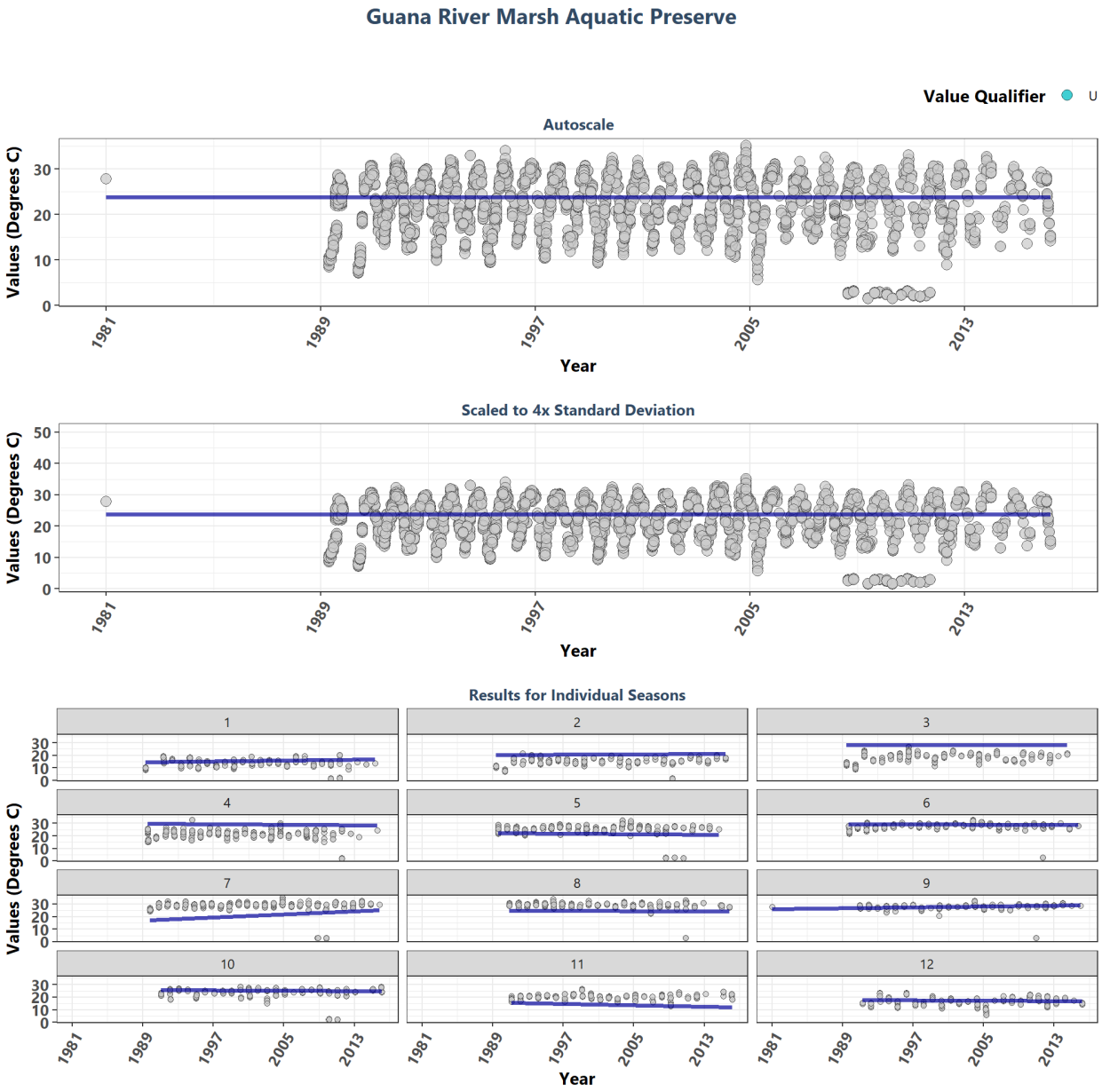
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	863	25.10	0.0807	0.0365	20.7017	3.9	0.0001	51.1	0	1
1	50	13.89	-0.1522	-0.0417	29.5570	0.5	0.6255	NA	NA	-1
2	46	15.25	0.1679	0.0907	22.8400	-0.8	0.4123	NA	NA	1
3	51	17.78	0.0148	0.0000	30.0000	0.0	0.9609	NA	NA	-1
4	73	21.50	0.1314	0.1211	17.7278	1.4	0.1512	NA	NA	1
5	85	26.53	0.1246	0.0310	27.7258	6.3	0.0000	NA	NA	1
6	87	28.50	0.0199	0.0000	29.7000	1.7	0.0843	NA	NA	-1
7	88	29.70	-0.0522	-0.0500	17.1556	0.3	0.7861	NA	NA	-1
8	61	30.00	0.4597	0.2371	20.8406	0.2	0.8707	NA	NA	1
9	96	28.12	0.1145	0.0551	20.5628	-2.2	0.0253	NA	NA	1
10	126	24.79	-0.0841	-0.0359	15.9554	2.8	0.0049	NA	NA	-1
11	50	20.15	0.0482	0.0343	13.2364	1.3	0.1786	NA	NA	1
12	50	16.16	-0.0055	0.0000	17.7778	-0.5	0.5963	NA	NA	-1

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



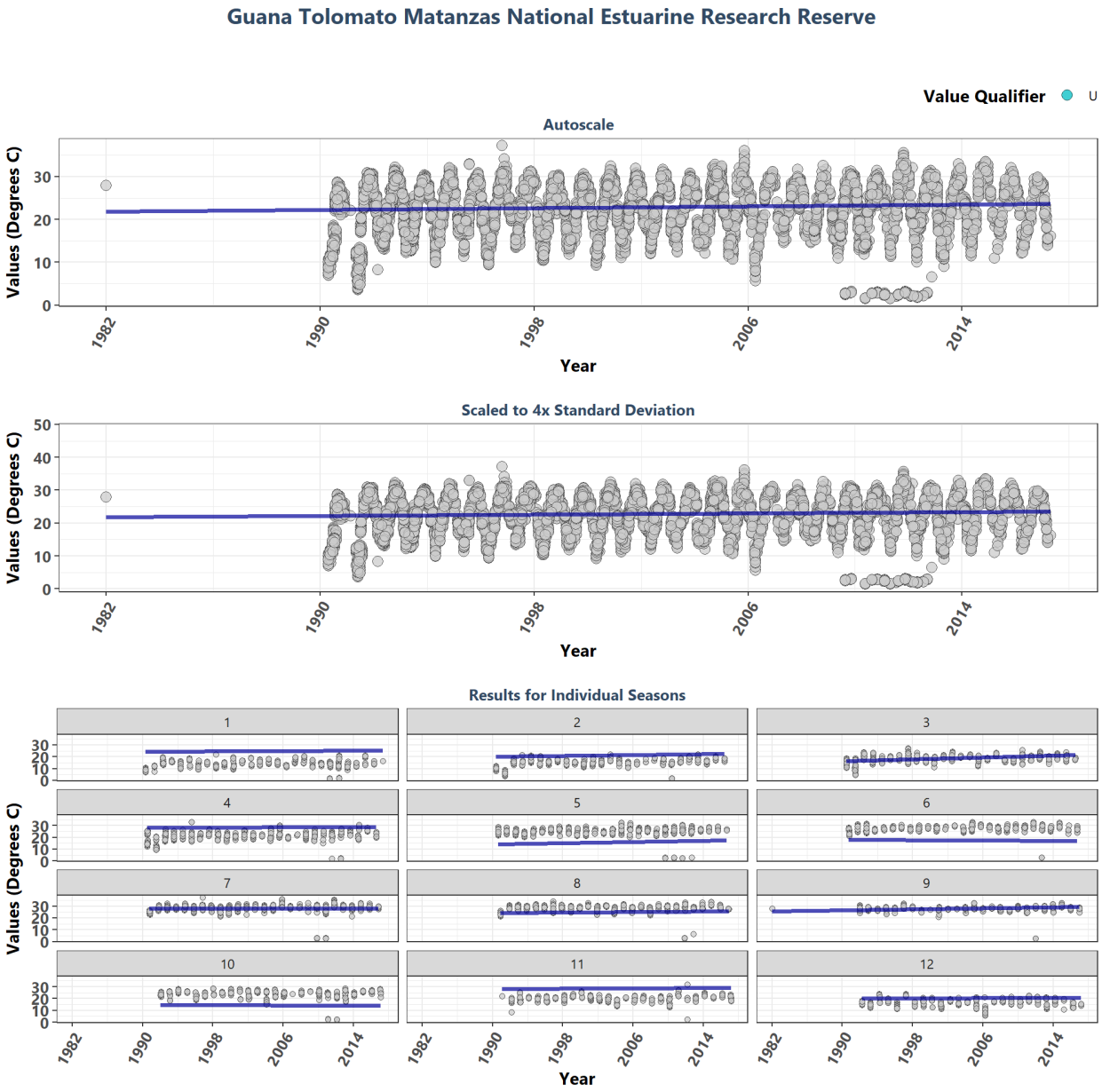
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	36215	26.00	0.0367	0.0145	24.8378	10.0	0.0000	316.1	0	1
1	2546	19.00	0.1634	0.0533	26.3200	2.9	0.0034	NA	NA	1
2	2805	19.80	0.1045	0.0313	28.7625	-0.4	0.7165	NA	NA	1
3	2630	22.30	0.0915	0.0314	28.5429	-1.5	0.1254	NA	NA	1
4	3327	24.80	-0.0266	-0.0100	29.9300	5.9	0.0000	NA	NA	-1
5	3206	27.30	0.0738	0.0500	17.9000	-3.2	0.0012	NA	NA	1
6	3006	29.40	-0.0046	0.0000	19.8000	-2.2	0.0284	NA	NA	-1
7	3016	30.24	0.0681	0.0278	23.3556	7.5	0.0000	NA	NA	1
8	2879	30.45	-0.0365	-0.0143	27.5286	8.4	0.0000	NA	NA	-1
9	3051	29.20	-0.0382	-0.0143	28.0429	13.6	0.0000	NA	NA	-1
10	3745	26.80	0.0387	0.0286	17.4571	-3.4	0.0008	NA	NA	1
11	3215	22.70	-0.0199	-0.0130	23.0043	3.4	0.0006	NA	NA	-1
12	2789	20.60	0.0401	0.0190	21.7095	5.8	0.0000	NA	NA	1

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



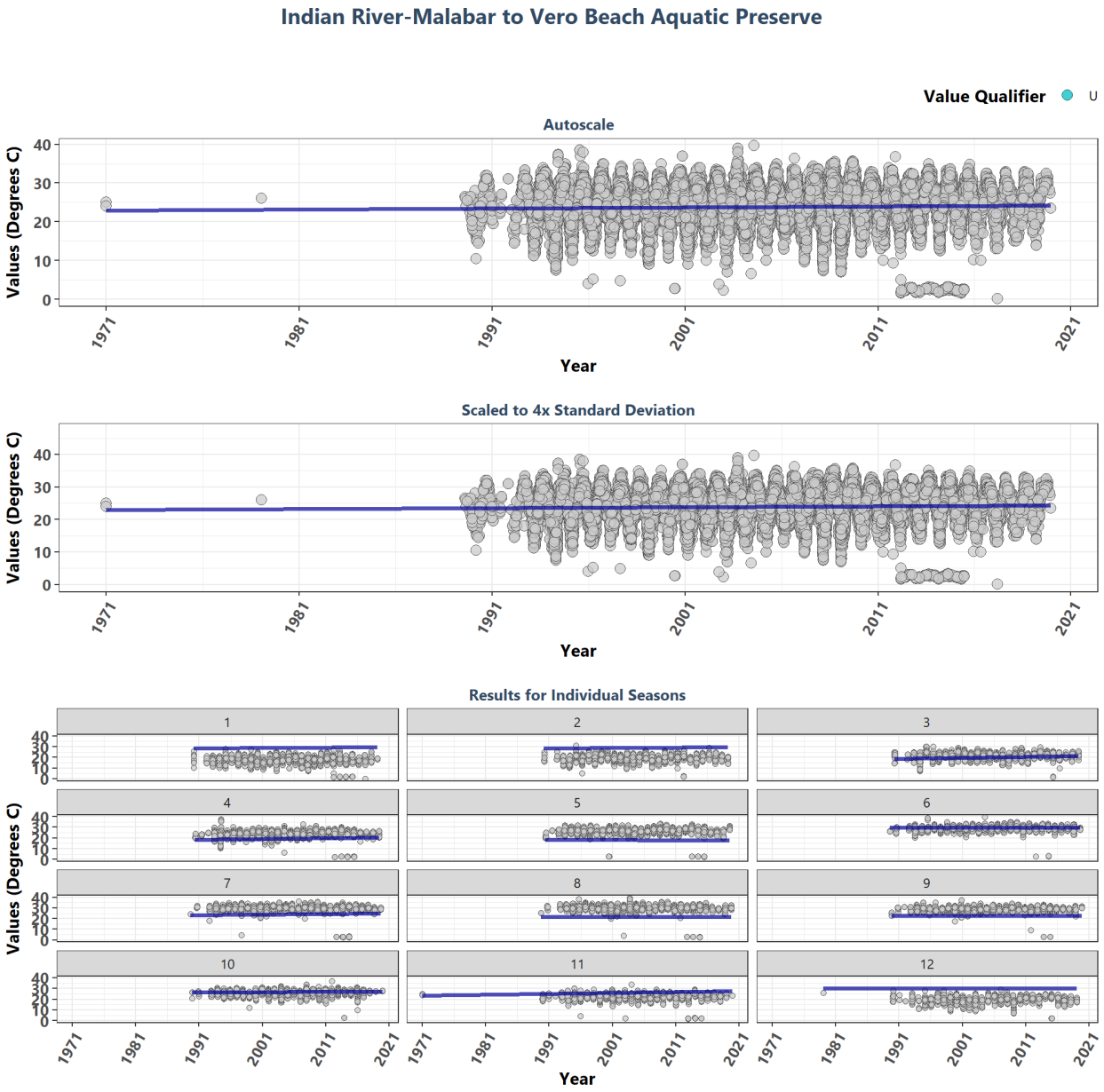
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5301	22.70	0.0032	0.0000	23.8334	-1.0	0.3051	183	0	0
1	507	14.30	0.1226	0.0800	14.0200	-4.2	0.0000	NA	NA	0
2	421	15.30	0.1224	0.0455	19.7273	3.8	0.0002	NA	NA	0
3	362	19.00	-0.0467	-0.0105	28.5684	7.8	0.0000	NA	NA	0
4	587	21.70	-0.1186	-0.0385	29.9154	-2.8	0.0045	NA	NA	0
5	588	25.60	-0.0782	-0.0538	22.6154	-2.5	0.0109	NA	NA	0
6	417	27.30	-0.0316	-0.0118	29.4000	6.6	0.0000	NA	NA	0
7	553	29.20	0.2743	0.3000	14.6500	-1.1	0.2658	NA	NA	0
8	450	29.30	-0.0478	-0.0286	25.0514	-3.8	0.0002	NA	NA	0
9	285	28.40	0.2170	0.0900	26.0400	-1.2	0.2370	NA	NA	0
10	483	24.48	-0.0700	-0.0400	26.3200	-1.6	0.1158	NA	NA	0
11	361	20.50	-0.1240	-0.1333	16.8333	3.5	0.0005	NA	NA	0
12	287	17.30	-0.0403	-0.0368	17.9632	-1.0	0.3073	NA	NA	0

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



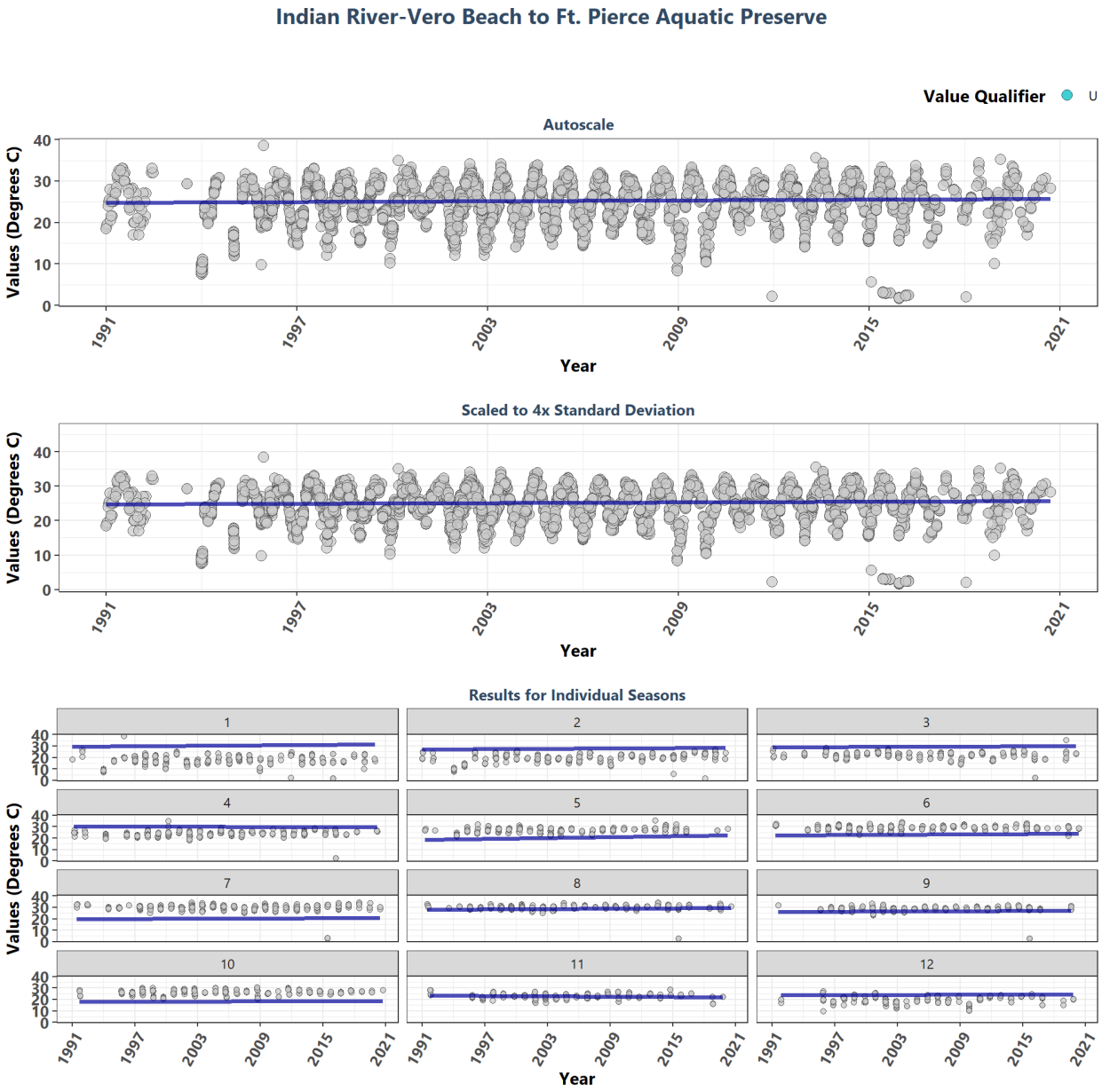
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	15446	23.0	0.1043	0.0500	21.8227	19.5	0.0000	343.5	0	1
1	1382	14.5	0.0451	0.0209	24.2826	-0.8	0.4044	NA	NA	1
2	1158	15.2	0.1312	0.0964	19.2618	12.6	0.0000	NA	NA	1
3	1356	17.8	0.2656	0.1993	14.8105	14.7	0.0000	NA	NA	1
4	1385	20.9	0.0408	0.0167	28.0333	7.3	0.0000	NA	NA	1
5	1508	24.9	0.2464	0.1333	13.0667	5.6	0.0000	NA	NA	1
6	1269	27.2	-0.0347	-0.0333	18.3000	13.6	0.0000	NA	NA	-1
7	1677	28.5	0.0035	0.0000	28.2000	6.0	0.0000	NA	NA	-1
8	1516	28.3	0.0953	0.0455	24.0364	2.4	0.0172	NA	NA	1
9	1010	28.2	0.2543	0.1022	25.6676	0.2	0.8666	NA	NA	1
10	1361	24.7	-0.0149	-0.0118	14.7235	2.5	0.0124	NA	NA	-1
11	983	20.3	0.0969	0.0400	27.8200	3.8	0.0002	NA	NA	1
12	841	17.7	0.0800	0.0364	19.6091	-1.5	0.1315	NA	NA	1

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



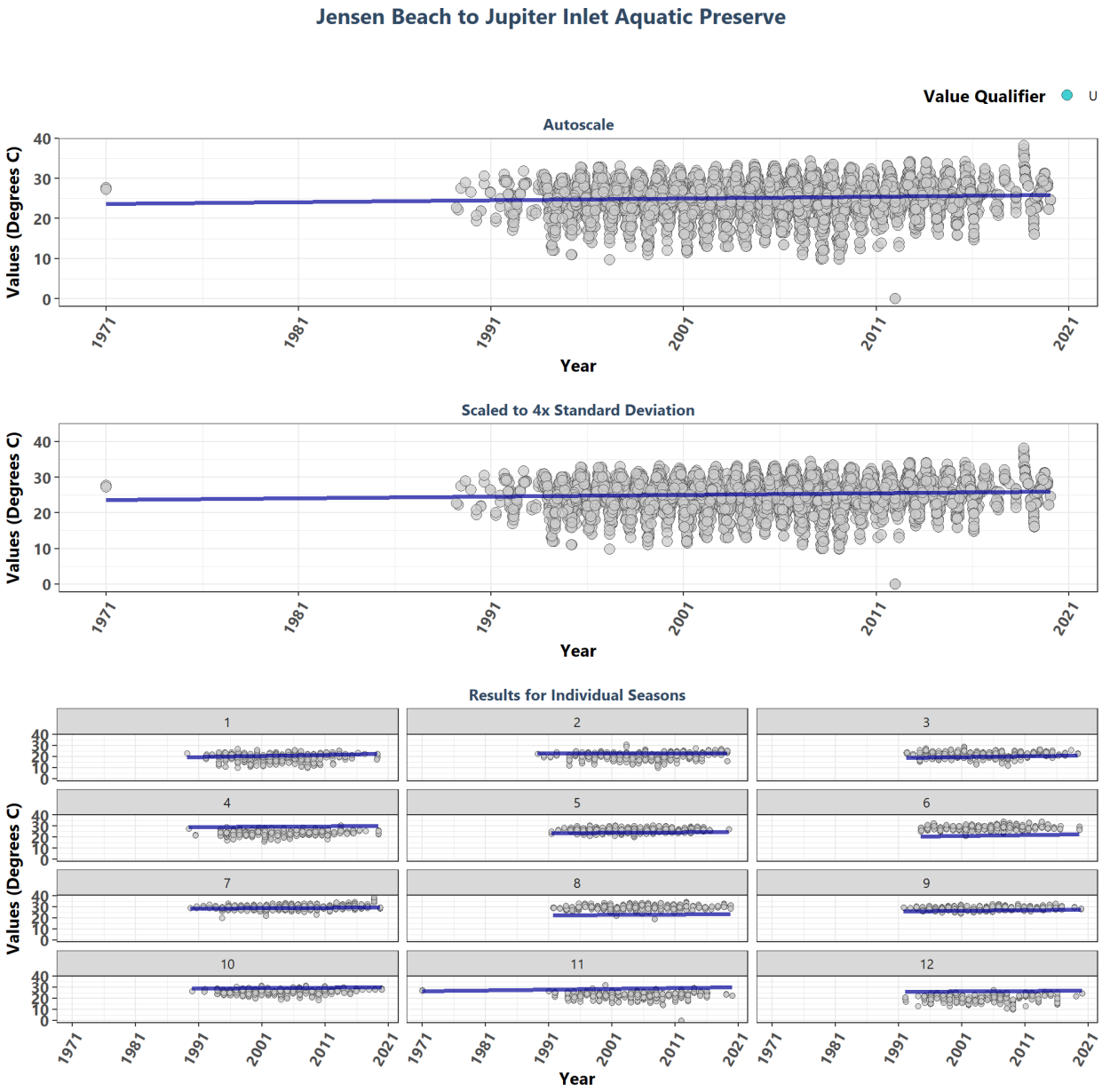
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	42288	25.20	0.0536	0.0275	22.9304	16.3	0.0000	293.1	0	1
1	3550	18.16	0.1062	0.0429	27.5857	-2.3	0.0236	NA	NA	1
2	3221	19.50	0.1139	0.0422	27.4144	8.2	0.0000	NA	NA	1
3	3135	21.90	0.1213	0.0950	16.8650	-0.4	0.7012	NA	NA	1
4	3934	24.00	0.0958	0.0714	17.1429	5.4	0.0000	NA	NA	1
5	3596	26.70	-0.0253	-0.0200	18.8550	3.8	0.0002	NA	NA	-1
6	3439	29.00	-0.0073	0.0000	29.6000	9.3	0.0000	NA	NA	-1
7	3986	29.60	0.0575	0.0379	22.6750	-0.7	0.4893	NA	NA	1
8	3703	30.00	-0.0046	0.0000	21.9000	0.2	0.8625	NA	NA	-1
9	3670	28.85	0.0129	0.0059	22.3800	10.4	0.0000	NA	NA	1
10	3851	26.10	0.0420	0.0200	26.0400	12.6	0.0000	NA	NA	1
11	3225	22.58	0.1349	0.0857	23.1857	1.1	0.2701	NA	NA	1
12	2978	20.00	0.0019	0.0000	30.0000	9.9	0.0000	NA	NA	-1

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



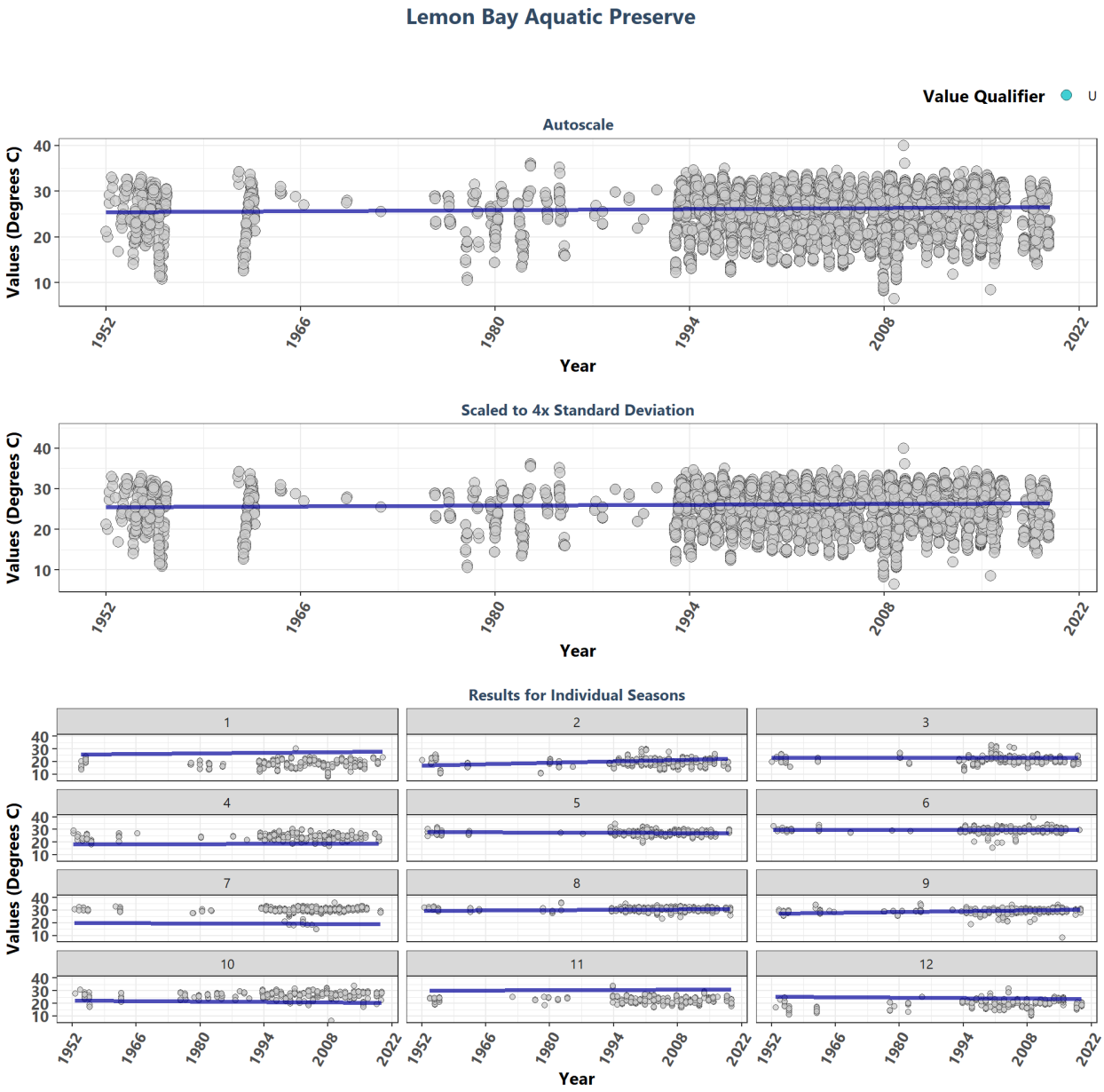
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5841	25.40	0.0782	0.0333	24.7644	8.1	0.0000	101.5	0	1
1	584	18.30	0.2040	0.0667	29.4333	0.6	0.5655	NA	NA	1
2	489	20.20	0.1545	0.0546	26.8990	7.3	0.0000	NA	NA	1
3	456	23.00	0.0866	0.0333	28.9000	-2.8	0.0047	NA	NA	1
4	622	24.00	-0.0471	-0.0182	30.3727	2.6	0.0093	NA	NA	-1
5	483	27.50	0.2200	0.1400	18.5200	5.1	0.0000	NA	NA	1
6	373	29.20	0.1019	0.0483	22.3683	2.5	0.0123	NA	NA	1
7	653	30.10	0.0435	0.0300	19.6700	-1.8	0.0711	NA	NA	1
8	442	30.10	0.1377	0.0455	28.0182	6.4	0.0000	NA	NA	1
9	422	28.70	0.1024	0.0389	25.9002	4.2	0.0000	NA	NA	1
10	608	26.41	0.0159	0.0125	18.1375	3.8	0.0002	NA	NA	1
11	335	22.90	-0.0883	-0.0571	23.6286	2.8	0.0053	NA	NA	-1
12	374	20.00	0.0696	0.0312	23.5938	1.3	0.2080	NA	NA	1

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



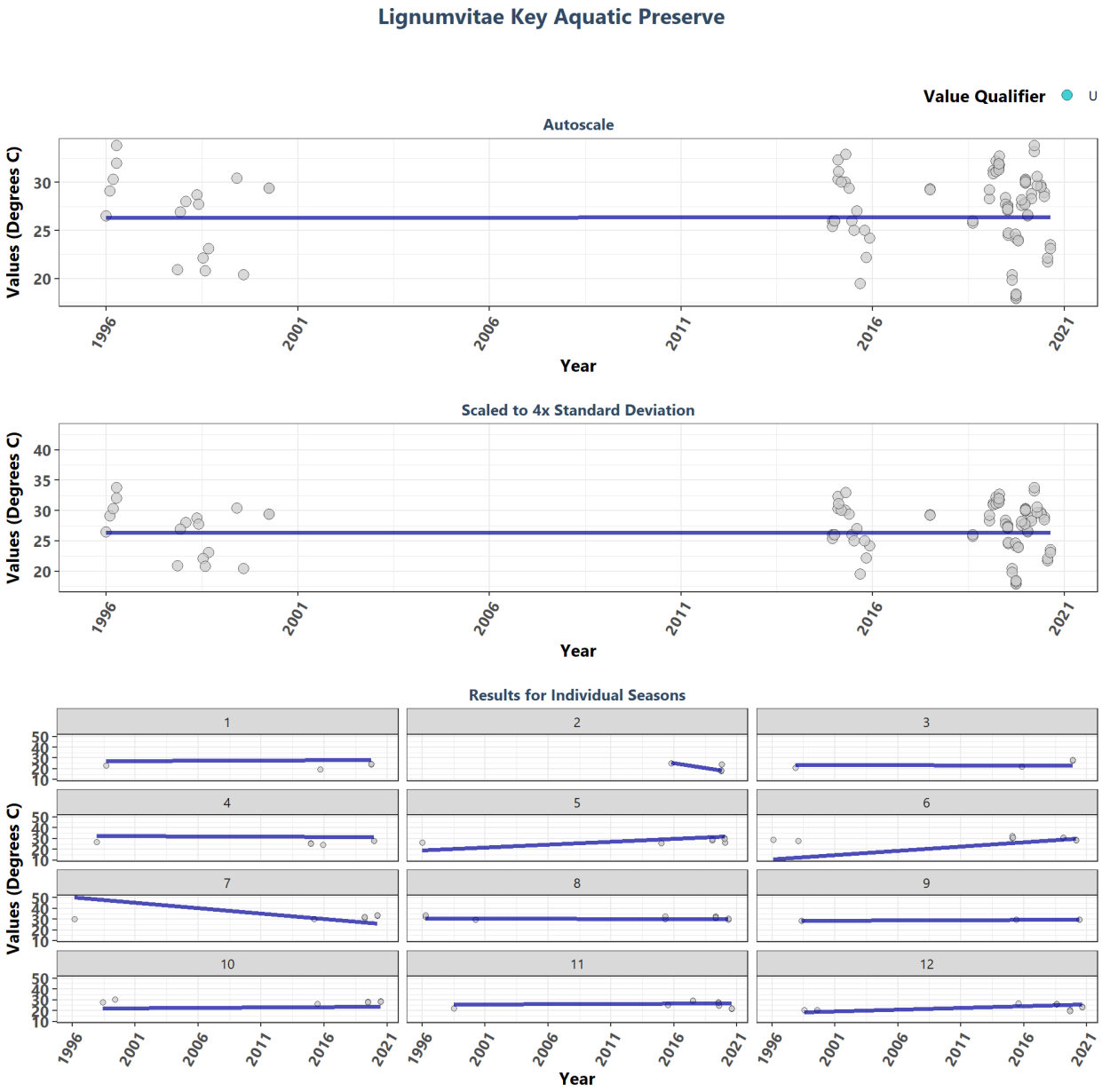
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	7080	25.50	0.0967	0.0467	23.6167	12.1	0.0000	18.5	0.0705	1
1	572	19.90	0.1430	0.0980	17.4720	3.3	0.0011	NA	NA	1
2	584	21.00	0.0138	0.0000	23.0000	5.2	0.0000	NA	NA	-1
3	563	22.60	0.0915	0.0667	17.5667	1.8	0.0688	NA	NA	1
4	553	24.00	0.1120	0.0400	28.3400	3.2	0.0016	NA	NA	1
5	603	26.40	0.0898	0.0500	22.3000	3.5	0.0004	NA	NA	1
6	565	28.80	0.1018	0.0750	18.7500	4.4	0.0000	NA	NA	1
7	688	29.68	0.1163	0.0364	27.4636	3.7	0.0002	NA	NA	1
8	647	29.70	0.0512	0.0278	21.6556	4.3	0.0000	NA	NA	1
9	590	28.70	0.1349	0.0667	24.2333	4.2	0.0000	NA	NA	1
10	585	26.50	0.0936	0.0333	28.5133	4.9	0.0000	NA	NA	1
11	631	23.00	0.1222	0.0625	26.6750	0.5	0.6043	NA	NA	1
12	499	21.30	0.0954	0.0429	24.9429	3.4	0.0007	NA	NA	1

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



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	8541	26.40	0.0495	0.0154	25.4381	7.5	0.0000	187.3	0	1
1	524	18.77	0.1038	0.0358	25.4377	0.7	0.4587	NA	NA	1
2	664	19.35	0.1891	0.0837	16.6953	-0.9	0.3679	NA	NA	1
3	576	20.90	0.0063	0.0000	22.9000	-2.5	0.0134	NA	NA	-1
4	673	23.90	0.0216	0.0095	18.2905	-3.3	0.0009	NA	NA	1
5	639	27.40	-0.0364	-0.0083	27.8333	-1.4	0.1686	NA	NA	-1
6	754	29.90	0.0254	0.0055	29.6164	1.0	0.2959	NA	NA	1
7	878	30.50	-0.0233	-0.0111	19.9167	3.5	0.0004	NA	NA	-1
8	798	30.88	0.0792	0.0186	29.5343	2.3	0.0228	NA	NA	1
9	732	29.80	0.2647	0.0500	27.0500	10.7	0.0000	NA	NA	1
10	890	27.30	-0.0688	-0.0250	22.2250	4.6	0.0000	NA	NA	-1
11	751	22.90	0.0537	0.0102	30.3250	0.3	0.7952	NA	NA	1
12	662	21.30	-0.0854	-0.0308	25.4385	7.3	0.0000	NA	NA	-1

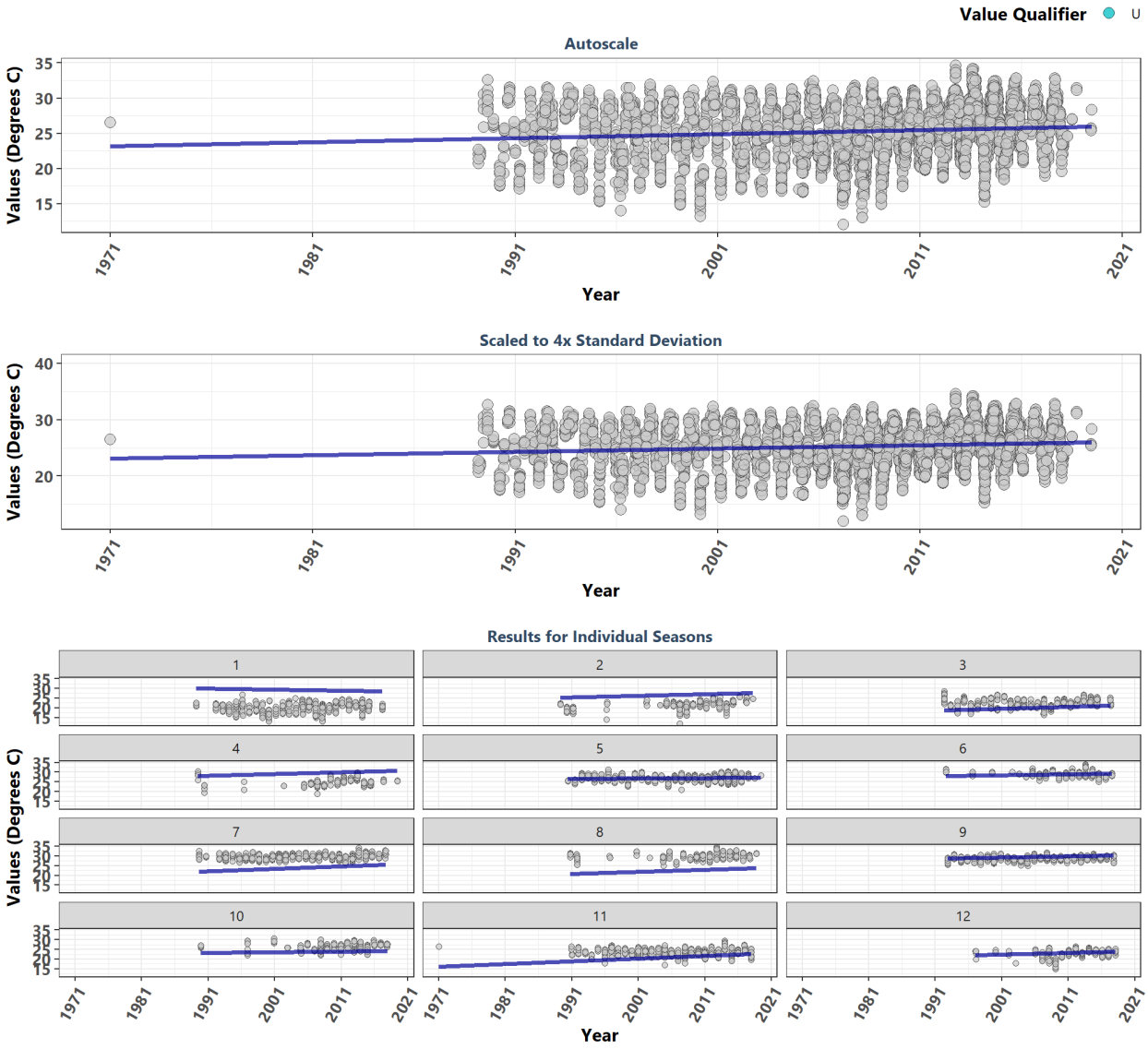
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	127	27.70	0.1074	0.0021	26.3545	0.1	0.9365	24.3	0.0117	0
1	4	23.65	0.2857	0.0545	27.1455	0.7	0.4701	NA	NA	0
2	15	NA	-0.1333	-1.6875	58.8000	NA	NA	NA	NA	NA
3	4	24.90	-0.1111	-0.0300	23.7600	1.4	0.1486	NA	NA	0
4	6	26.45	-0.1225	-0.0489	32.6250	0.4	0.6967	NA	NA	0
5	18	30.00	0.7333	0.5500	19.0000	2.9	0.0035	NA	NA	0
6	9	30.30	0.3922	0.8000	10.8000	0.0	1.0000	NA	NA	0
7	6	31.65	-0.1645	-1.0000	50.2000	1.9	0.0513	NA	NA	0
8	23	31.50	-0.0278	-0.0125	30.5375	-0.9	0.3457	NA	NA	0
9	4	29.45	0.8333	0.0412	28.5647	1.4	0.1486	NA	NA	0
10	7	28.40	0.5000	0.0714	22.0786	0.8	0.4373	NA	NA	0
11	22	27.20	0.2000	0.0455	25.5636	-1.4	0.1725	NA	NA	0
12	9	23.10	0.8333	0.3318	17.6000	-0.3	0.7492	NA	NA	0

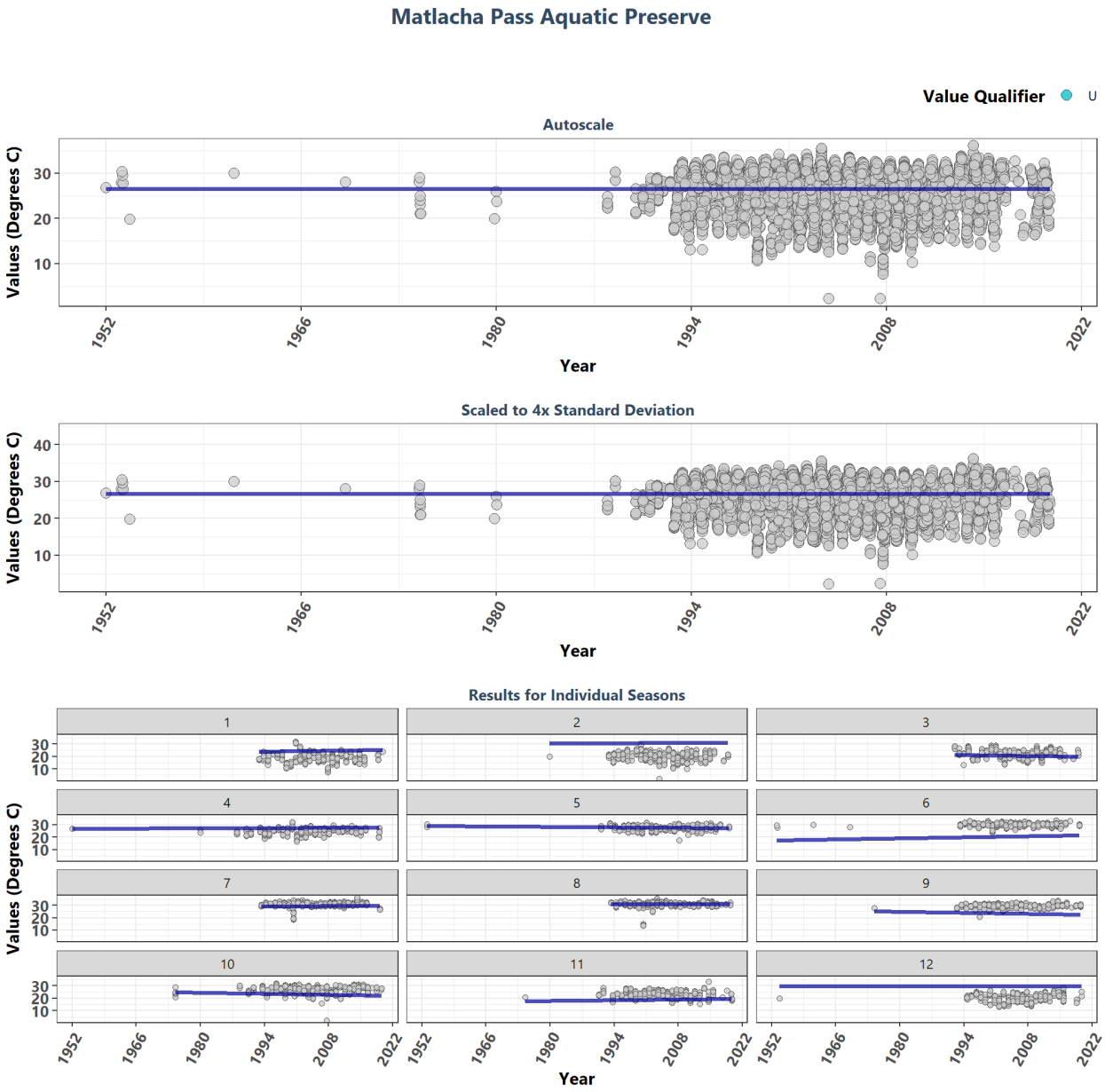
[a] $p < 0.00005$ appear as 0 due to rounding

Loxahatchee River-Lake Worth Creek Aquatic Preserve



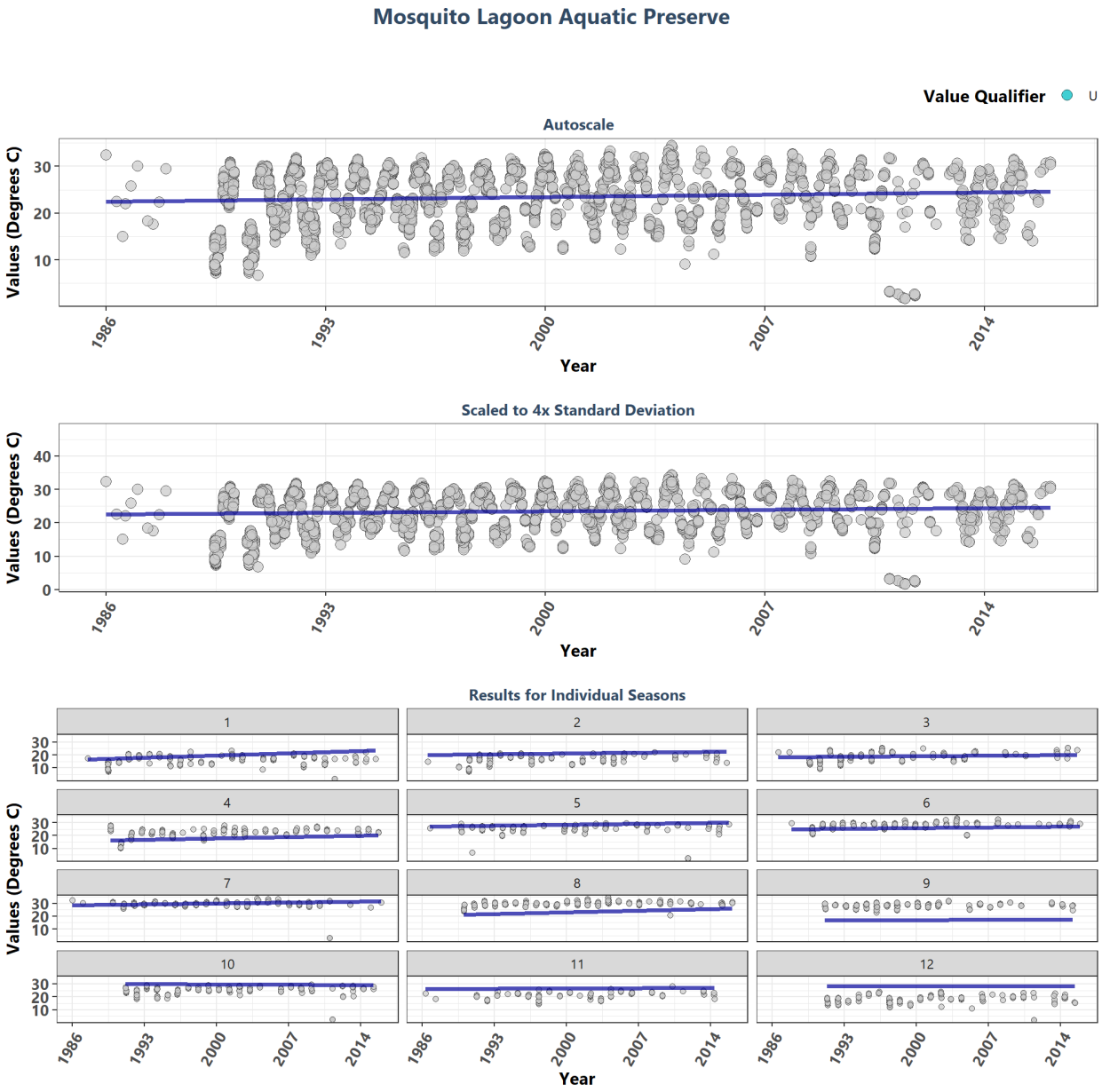
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5511	25.6	0.1571	0.0583	23.1227	16.8	0.0000	107.8	0	1
1	670	20.3	-0.1122	-0.0500	30.9000	7.3	0.0000	NA	NA	-1
2	301	21.6	0.1552	0.0818	23.6455	6.9	0.0000	NA	NA	1
3	613	22.7	0.1882	0.1000	16.6050	8.2	0.0000	NA	NA	1
4	257	24.7	0.1979	0.0818	26.6455	4.8	0.0000	NA	NA	1
5	702	27.0	0.0665	0.0235	26.1294	2.6	0.0083	NA	NA	1
6	247	28.8	0.2105	0.0526	26.9053	-2.6	0.0084	NA	NA	1
7	670	29.7	0.1984	0.1200	19.7800	8.1	0.0000	NA	NA	1
8	267	30.0	0.2201	0.1067	18.7533	4.8	0.0000	NA	NA	1
9	608	28.8	0.2091	0.0533	27.6733	7.8	0.0000	NA	NA	1
10	292	27.0	0.0903	0.0333	22.6000	4.0	0.0001	NA	NA	1
11	647	23.8	0.2661	0.1333	16.2667	3.4	0.0006	NA	NA	1
12	237	23.3	0.1275	0.0857	19.7000	2.9	0.0033	NA	NA	1

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



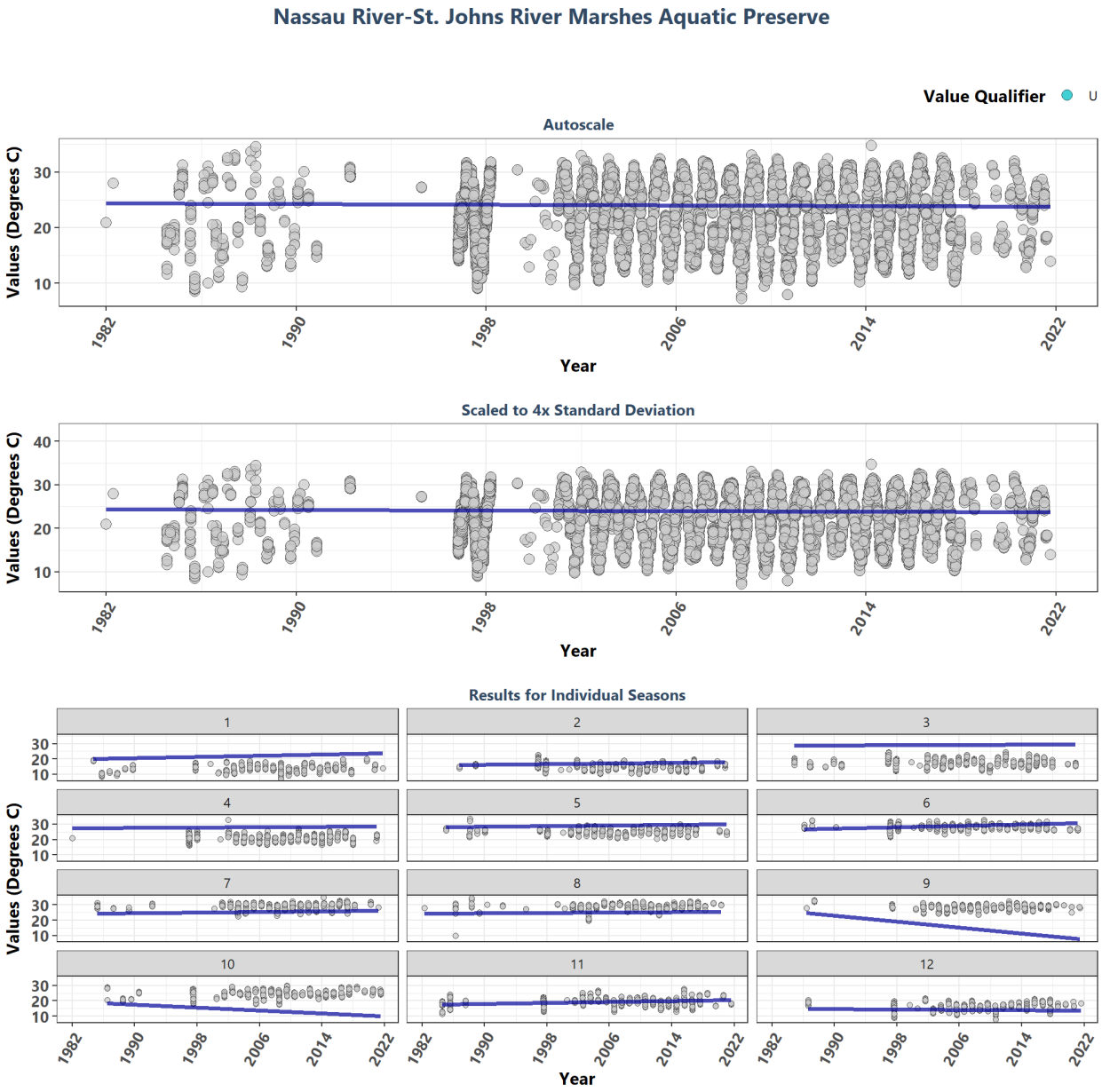
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	6686	26.10	0.0045	0.0000	26.6188	0.5	0.6450	60.8	0	0
1	464	19.05	0.1180	0.0500	21.9000	1.4	0.1473	NA	NA	0
2	547	20.40	0.0564	0.0154	29.9154	-2.6	0.0087	NA	NA	0
3	533	23.10	-0.0748	-0.0538	23.3077	-2.3	0.0242	NA	NA	0
4	592	24.50	0.0258	0.0114	26.8091	4.3	0.0000	NA	NA	0
5	602	27.50	-0.0605	-0.0250	28.8000	-2.2	0.0263	NA	NA	0
6	551	29.90	0.0730	0.0600	17.5200	0.0	0.9981	NA	NA	0
7	579	30.70	0.0360	0.0111	28.8111	2.0	0.0419	NA	NA	0
8	566	30.77	0.0046	0.0000	30.7750	0.2	0.8688	NA	NA	0
9	519	29.40	-0.0930	-0.0571	26.4286	1.2	0.2202	NA	NA	0
10	674	27.40	-0.0652	-0.0538	25.9538	1.0	0.3153	NA	NA	0
11	587	23.40	0.0449	0.0444	16.6944	-3.4	0.0007	NA	NA	0
12	472	20.70	-0.0001	0.0000	29.9000	2.4	0.0176	NA	NA	0

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



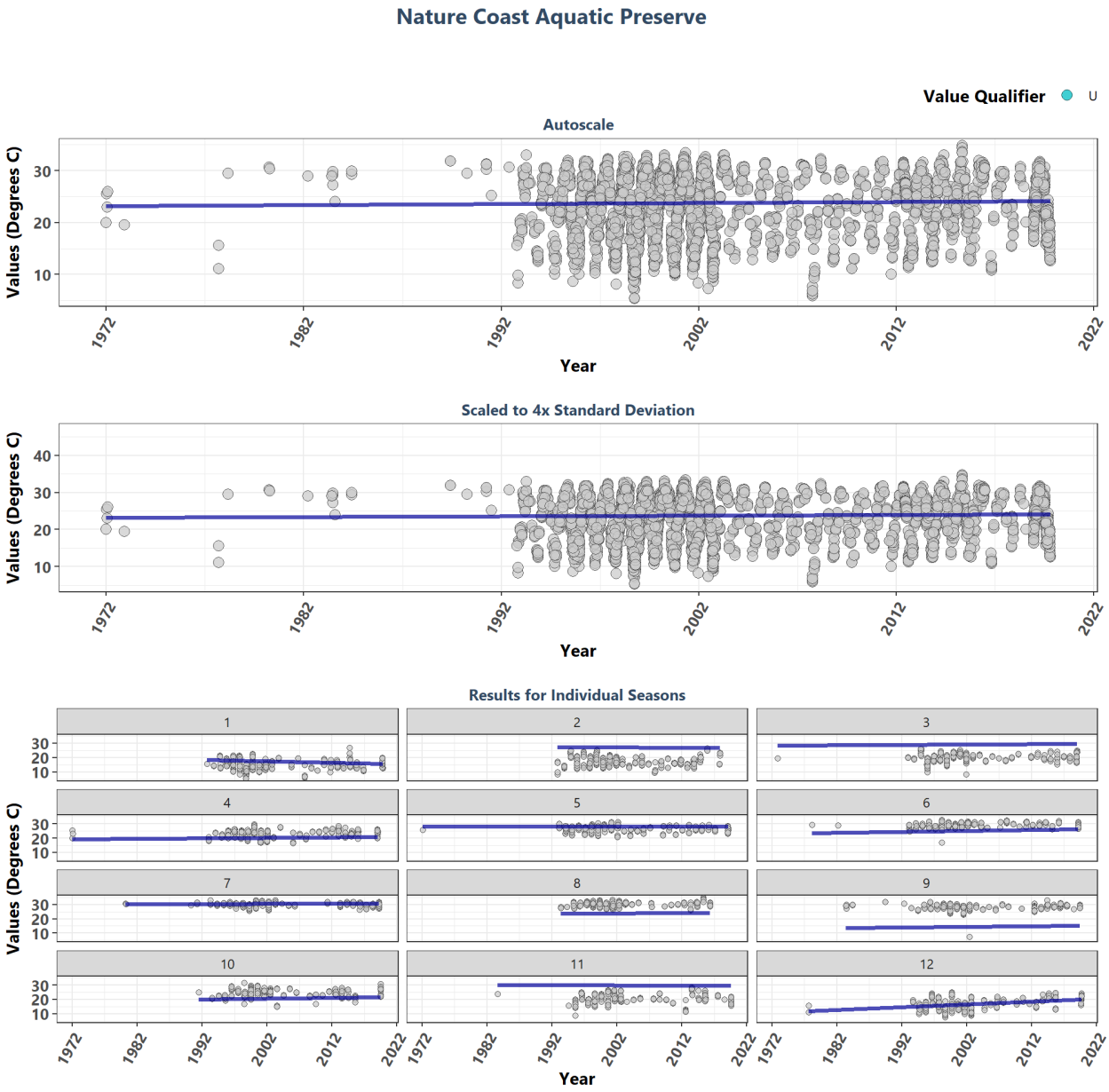
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	3922	25.10	0.1456	0.0692	22.4806	13.3	0.0000	134.7	0	1
1	348	17.10	0.2939	0.2500	16.0000	0.6	0.5290	NA	NA	1
2	243	17.30	0.1892	0.0875	20.1125	4.5	0.0000	NA	NA	1
3	290	18.50	0.1109	0.0615	18.0615	7.5	0.0000	NA	NA	1
4	319	22.90	0.1914	0.1429	15.8714	7.2	0.0000	NA	NA	1
5	273	26.30	0.2381	0.1083	27.0083	1.9	0.0641	NA	NA	1
6	395	28.20	0.1511	0.0875	24.5500	7.1	0.0000	NA	NA	1
7	385	29.60	0.3055	0.1000	28.4000	-2.6	0.0104	NA	NA	1
8	426	29.70	0.2702	0.1778	20.4111	9.4	0.0000	NA	NA	1
9	370	28.30	0.0226	0.0200	16.8600	0.5	0.6384	NA	NA	1
10	361	25.60	-0.0873	-0.0333	30.0000	4.3	0.0000	NA	NA	-1
11	216	21.25	0.0750	0.0350	25.9500	4.2	0.0000	NA	NA	1
12	296	18.80	0.0163	0.0000	28.3000	2.9	0.0043	NA	NA	-1

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



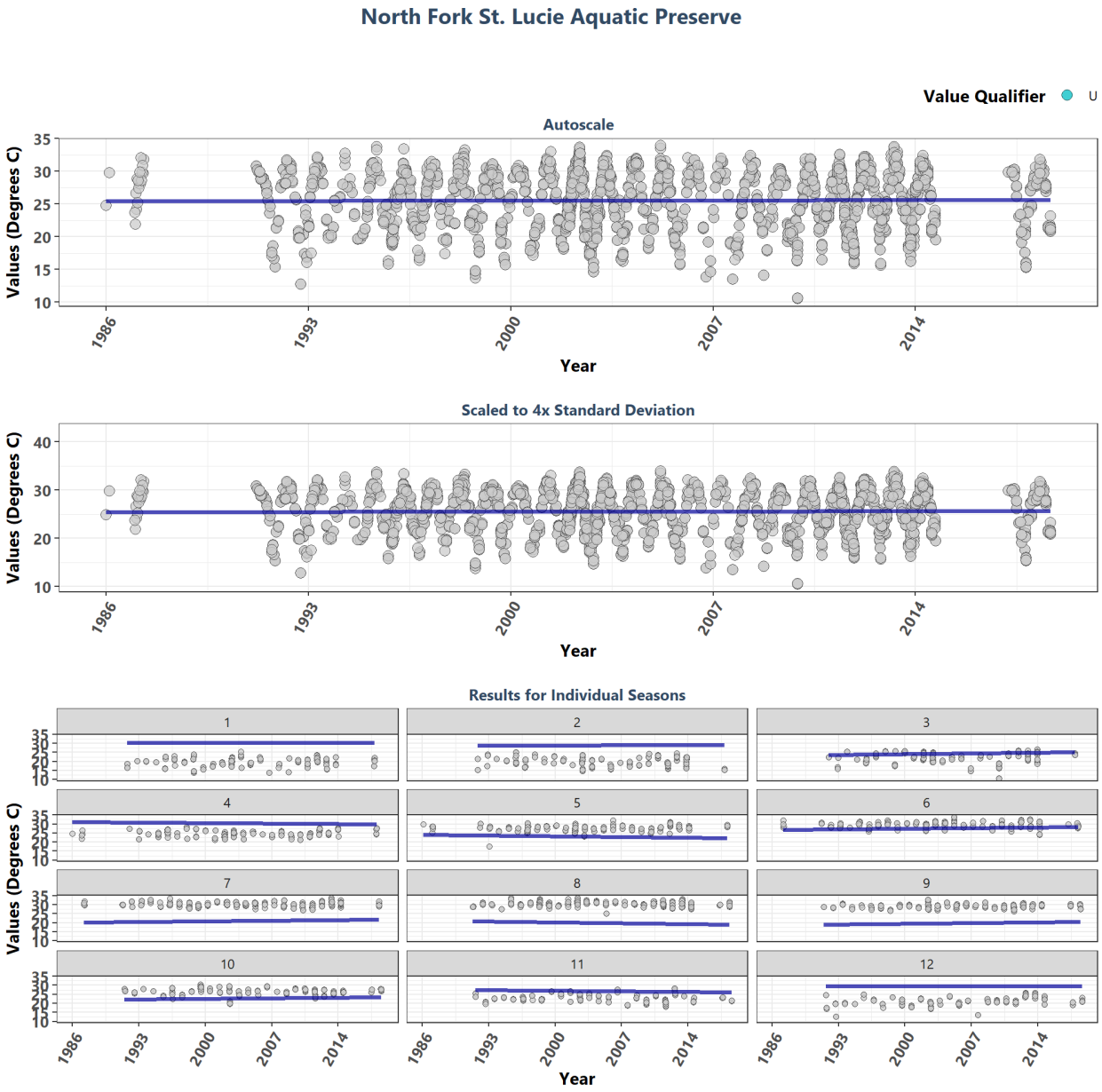
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	28419	21.29	-0.0129	-0.0162	24.3675	-5.8	0.0000	2816.5	0	-1
1	1519	14.30	0.1138	0.0890	20.0150	-4.5	0.0000	NA	NA	1
2	3296	15.74	0.1062	0.0610	15.6850	-27.4	0.0000	NA	NA	1
3	3909	19.09	0.0487	0.0182	28.8091	-36.5	0.0000	NA	NA	1
4	3864	21.35	0.0703	0.0250	27.6250	11.8	0.0000	NA	NA	1
5	1649	24.91	0.1462	0.0500	28.1000	4.0	0.0001	NA	NA	1
6	4570	28.20	0.1600	0.1140	26.3760	17.5	0.0000	NA	NA	1
7	886	29.40	0.0553	0.0450	24.2250	6.5	0.0000	NA	NA	1
8	780	29.30	0.0629	0.0250	24.5100	2.0	0.0416	NA	NA	1
9	802	28.30	-0.3610	-0.4823	26.8069	3.0	0.0028	NA	NA	-1
10	2859	24.90	-0.2980	-0.2408	19.5933	5.6	0.0000	NA	NA	-1
11	2711	18.60	0.1154	0.0842	17.3375	11.0	0.0000	NA	NA	1
12	1574	16.60	-0.0728	-0.0315	14.8040	6.8	0.0000	NA	NA	-1

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



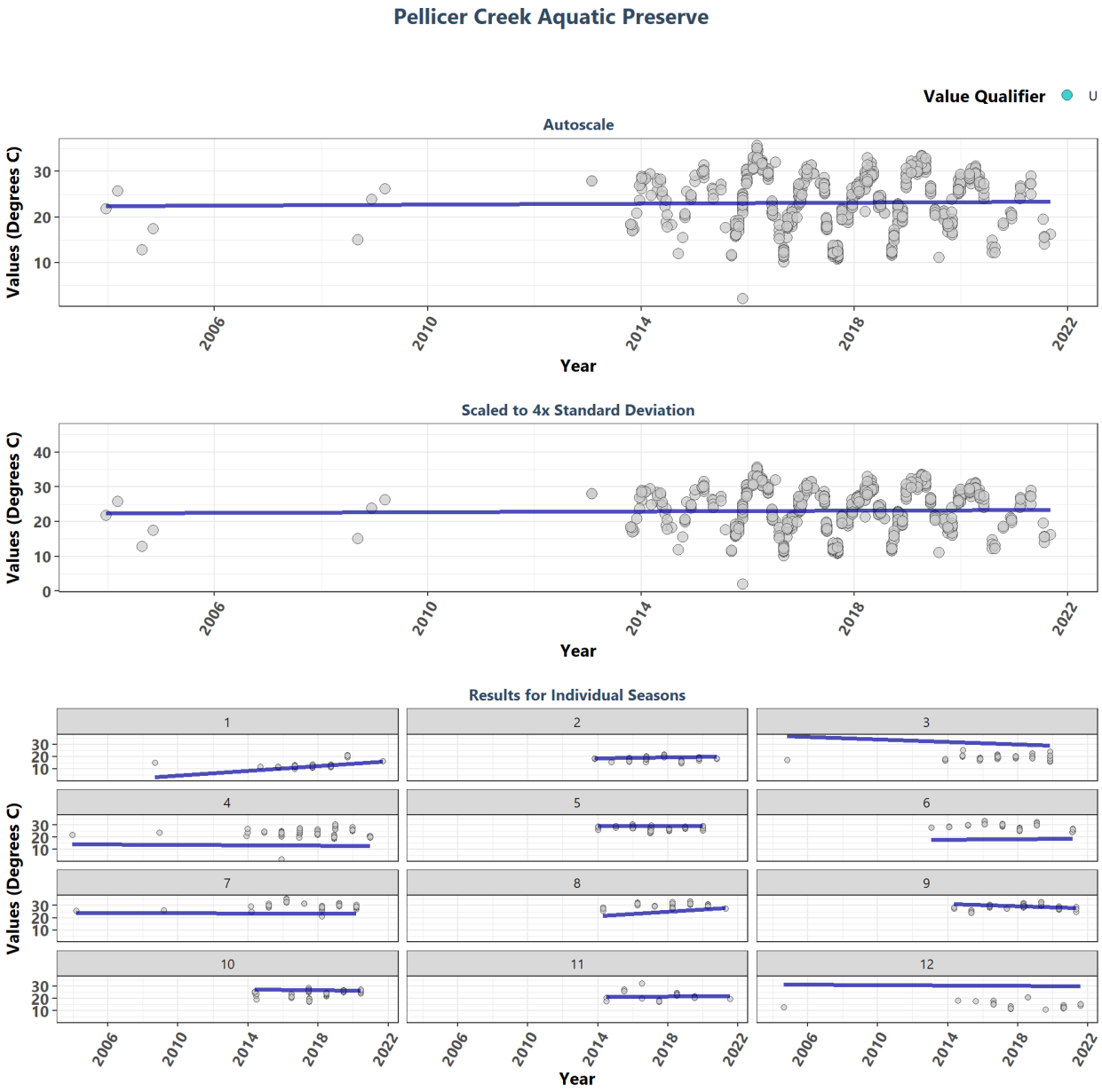
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	4981	24.44	0.0434	0.0200	23.1828	4.6	0.0000	58.4	0	1
1	441	14.40	-0.0983	-0.0900	20.1100	2.1	0.0319	NA	NA	-1
2	403	17.59	-0.0264	-0.0100	27.4900	-3.0	0.0031	NA	NA	-1
3	390	20.35	0.0786	0.0250	28.4750	1.8	0.0675	NA	NA	1
4	439	23.70	0.0617	0.0365	19.2924	1.4	0.1535	NA	NA	1
5	416	27.20	0.0167	0.0046	28.1314	-0.8	0.4198	NA	NA	1
6	435	29.20	0.1149	0.0639	23.2456	2.5	0.0139	NA	NA	1
7	438	30.00	0.0170	0.0077	30.1846	-1.1	0.2507	NA	NA	1
8	334	30.40	0.0454	0.0200	23.1200	0.5	0.6411	NA	NA	1
9	442	28.26	0.0681	0.0478	13.0130	0.5	0.5991	NA	NA	1
10	457	25.10	0.0870	0.0567	18.8567	3.7	0.0002	NA	NA	1
11	379	20.50	-0.0366	-0.0091	30.2636	2.5	0.0111	NA	NA	-1
12	407	16.38	0.1868	0.2022	10.5173	5.7	0.0000	NA	NA	1

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



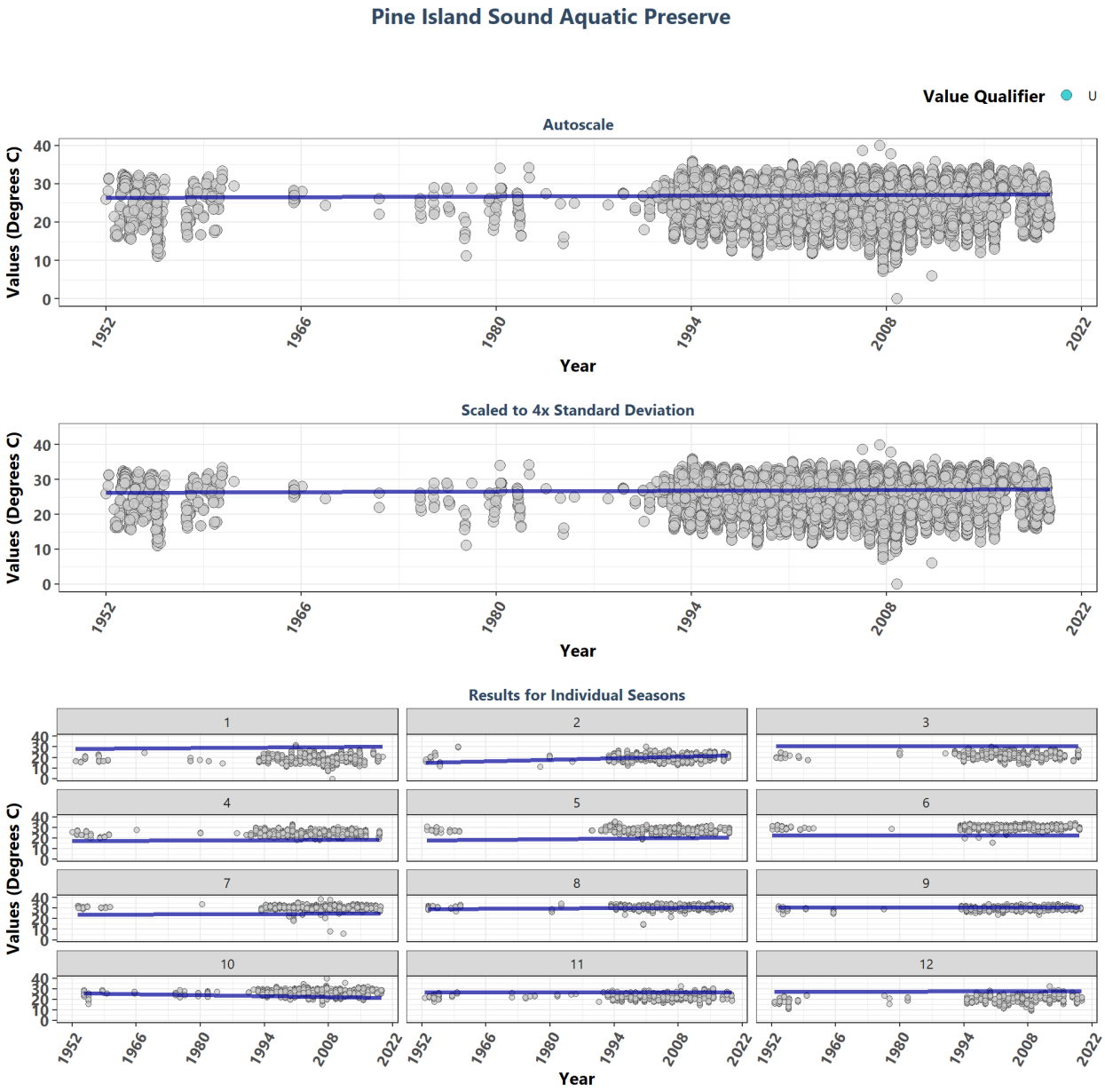
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	3193	26.10	0.0170	0.0060	25.4596	1.5	0.1396	74.2	0	0
1	243	19.90	-0.0270	-0.0067	30.5267	2.6	0.0086	NA	NA	0
2	248	19.73	0.0516	0.0125	28.7375	-2.3	0.0236	NA	NA	0
3	268	22.76	0.1415	0.0588	23.3506	1.9	0.0559	NA	NA	0
4	241	24.35	-0.1113	-0.0312	31.0050	3.3	0.0010	NA	NA	0
5	291	27.60	-0.1206	-0.0575	24.1650	3.7	0.0002	NA	NA	0
6	268	29.40	0.1446	0.0529	26.7543	0.0	0.9629	NA	NA	0
7	280	30.40	0.1358	0.0533	20.0667	-0.7	0.4997	NA	NA	0
8	289	30.38	-0.0961	-0.0656	21.0361	-2.8	0.0047	NA	NA	0
9	299	29.00	0.1126	0.0584	18.7317	1.3	0.1813	NA	NA	0
10	249	26.69	0.0778	0.0442	22.0032	-2.6	0.0100	NA	NA	0
11	252	22.90	-0.1093	-0.0400	27.5700	-2.9	0.0042	NA	NA	0
12	265	21.08	-0.0019	0.0000	29.4000	3.3	0.0010	NA	NA	0

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



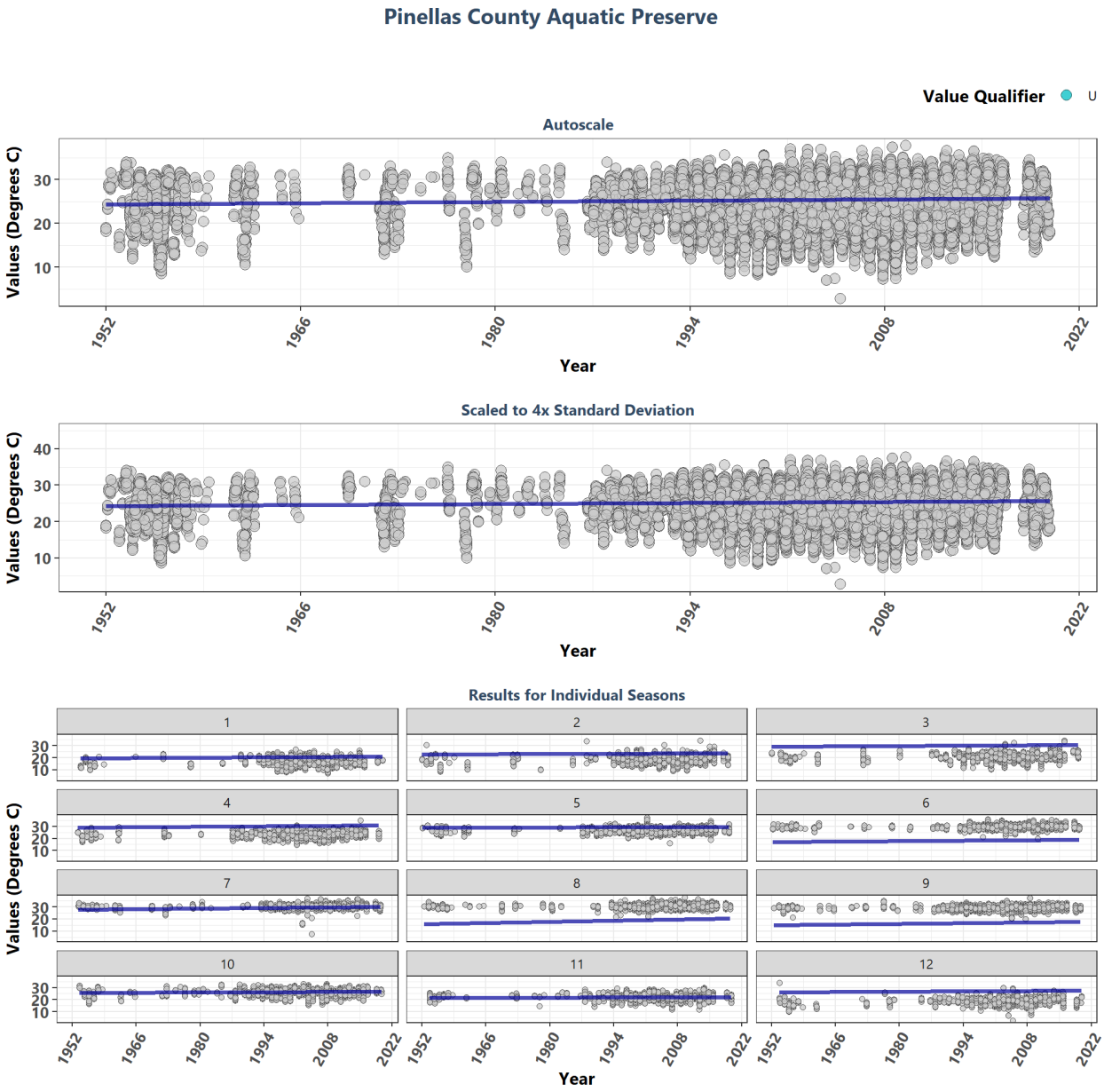
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	778	24.70	0.0265	0.0553	22.3102	1.0	0.3223	53.3	0	0
1	61	12.45	0.4038	0.9740	-3.1340	4.7	0.0000	NA	NA	0
2	73	18.22	0.1230	0.1779	16.6533	1.0	0.3225	NA	NA	0
3	79	19.50	-0.2101	-0.5002	37.8027	1.6	0.1040	NA	NA	0
4	89	23.56	-0.0379	-0.0692	14.2385	-0.1	0.9340	NA	NA	0
5	72	27.05	0.0169	0.0343	28.5520	-1.1	0.2907	NA	NA	0
6	70	29.80	0.0780	0.1200	16.3020	-2.7	0.0080	NA	NA	0
7	71	29.80	-0.0061	-0.0138	23.7813	-2.6	0.0084	NA	NA	0
8	61	30.30	0.2949	0.9000	10.5000	-0.5	0.6056	NA	NA	0
9	60	29.10	-0.2128	-0.4600	36.7000	0.2	0.8502	NA	NA	0
10	67	24.00	-0.0837	-0.1510	29.3175	3.6	0.0003	NA	NA	0
11	42	21.49	0.0046	0.0410	20.8390	0.0	0.9732	NA	NA	0
12	33	13.20	-0.0448	-0.0758	31.5133	-0.3	0.7551	NA	NA	0

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



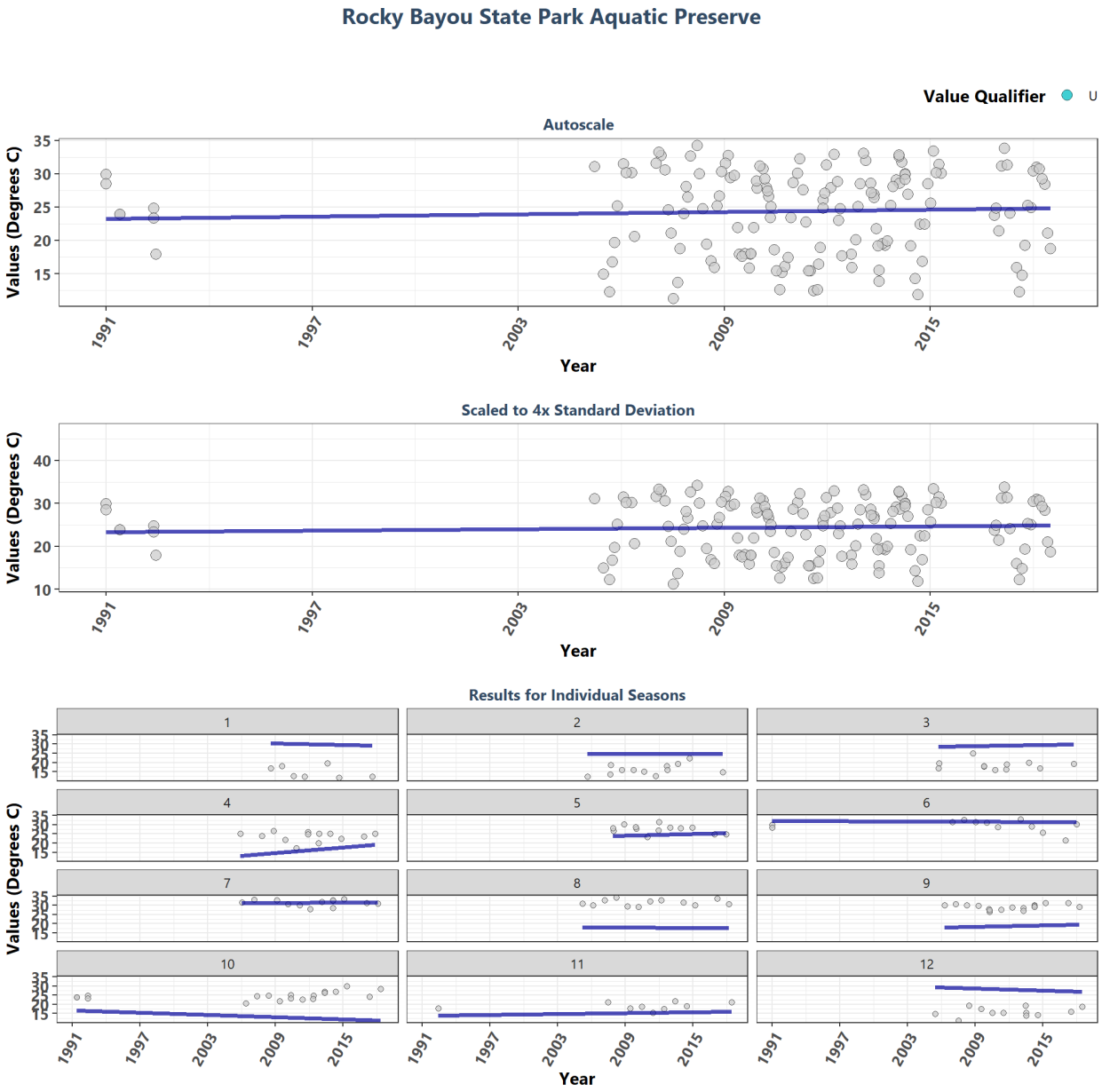
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	21313	26.2	0.0396	0.0142	26.3000	8.4	0.0000	263.3	0	1
1	1544	18.3	0.1277	0.0333	27.8333	1.9	0.0602	NA	NA	1
2	1495	19.8	0.2134	0.1097	14.6802	3.7	0.0002	NA	NA	1
3	1637	22.4	-0.0109	0.0000	30.9000	-6.9	0.0000	NA	NA	-1
4	1726	24.7	0.0319	0.0200	17.2000	3.0	0.0028	NA	NA	1
5	1762	27.6	0.0643	0.0365	17.7904	2.1	0.0343	NA	NA	1
6	1768	30.2	0.0133	0.0022	22.6826	4.9	0.0000	NA	NA	1
7	2035	30.6	0.0480	0.0200	23.6400	0.7	0.4721	NA	NA	1
8	1943	30.9	0.0774	0.0227	28.9727	-0.7	0.4696	NA	NA	1
9	1918	29.7	0.0106	0.0000	30.6000	8.4	0.0000	NA	NA	-1
10	1982	26.6	-0.1136	-0.0667	26.0000	-0.4	0.7115	NA	NA	-1
11	1882	22.8	-0.0055	0.0000	26.6000	0.9	0.3856	NA	NA	-1
12	1621	20.6	0.0336	0.0125	26.9375	12.9	0.0000	NA	NA	1

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



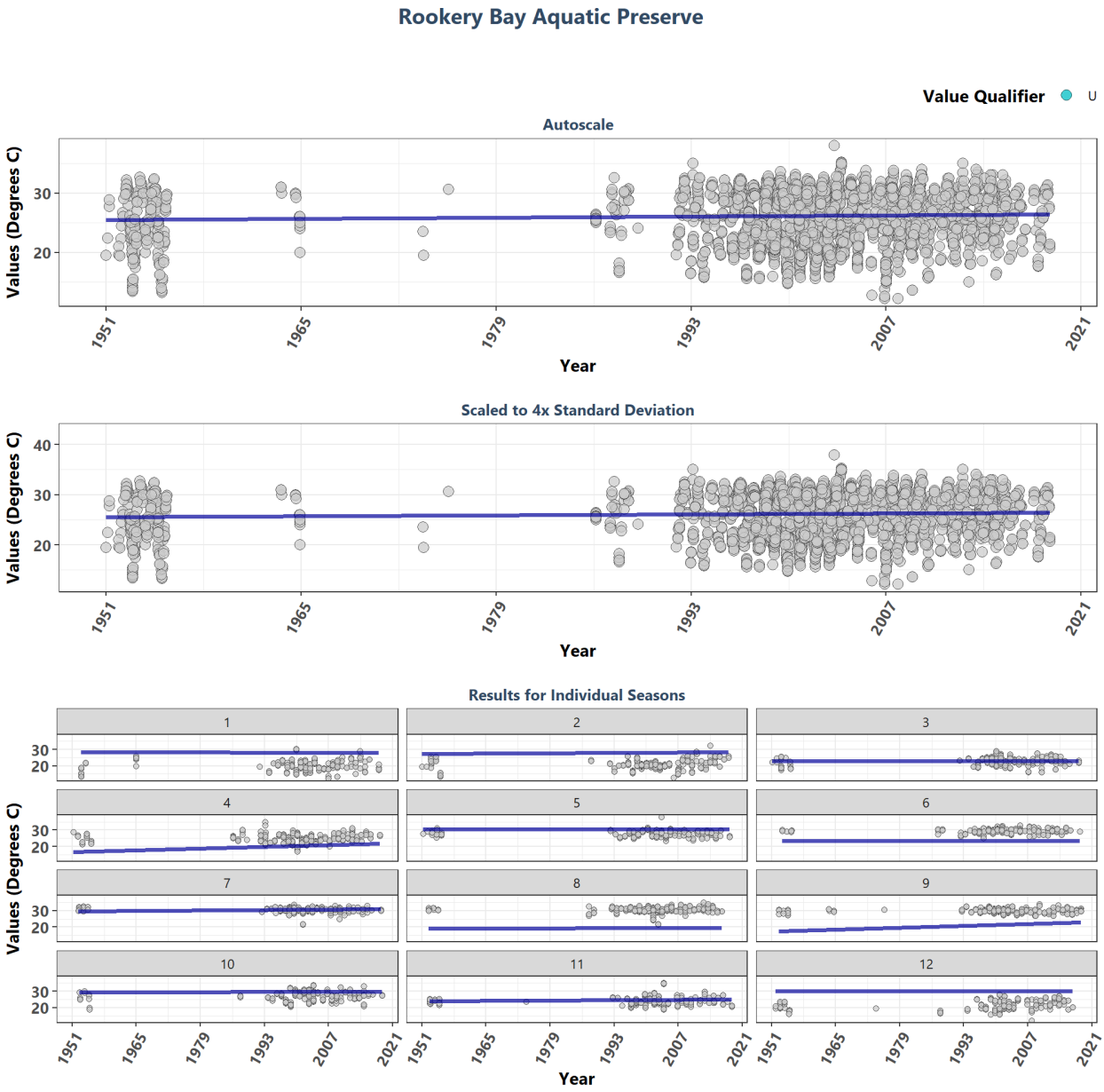
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	57274	26.13	0.0686	0.0202	24.3358	23.3	0.0000	221	0	1
1	3845	17.10	0.0434	0.0200	19.6600	5.8	0.0000	NA	NA	1
2	3944	18.38	0.0314	0.0122	23.0134	4.2	0.0000	NA	NA	1
3	3993	20.70	0.1098	0.0243	29.0286	4.1	0.0000	NA	NA	1
4	4149	23.65	0.1182	0.0270	29.0920	3.0	0.0024	NA	NA	1
5	5699	27.10	0.0303	0.0067	28.9800	7.1	0.0000	NA	NA	1
6	4680	29.70	0.0449	0.0246	17.0739	15.7	0.0000	NA	NA	1
7	4155	30.34	0.1526	0.0375	27.6750	10.6	0.0000	NA	NA	1
8	5032	30.55	0.1212	0.0667	15.8000	12.6	0.0000	NA	NA	1
9	7220	29.32	0.0619	0.0400	15.0200	3.9	0.0001	NA	NA	1
10	5800	26.34	0.0389	0.0136	25.6582	4.4	0.0000	NA	NA	1
11	4200	21.93	0.0214	0.0072	21.5534	2.1	0.0370	NA	NA	1
12	4557	19.40	0.0629	0.0200	26.0400	12.3	0.0000	NA	NA	1

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



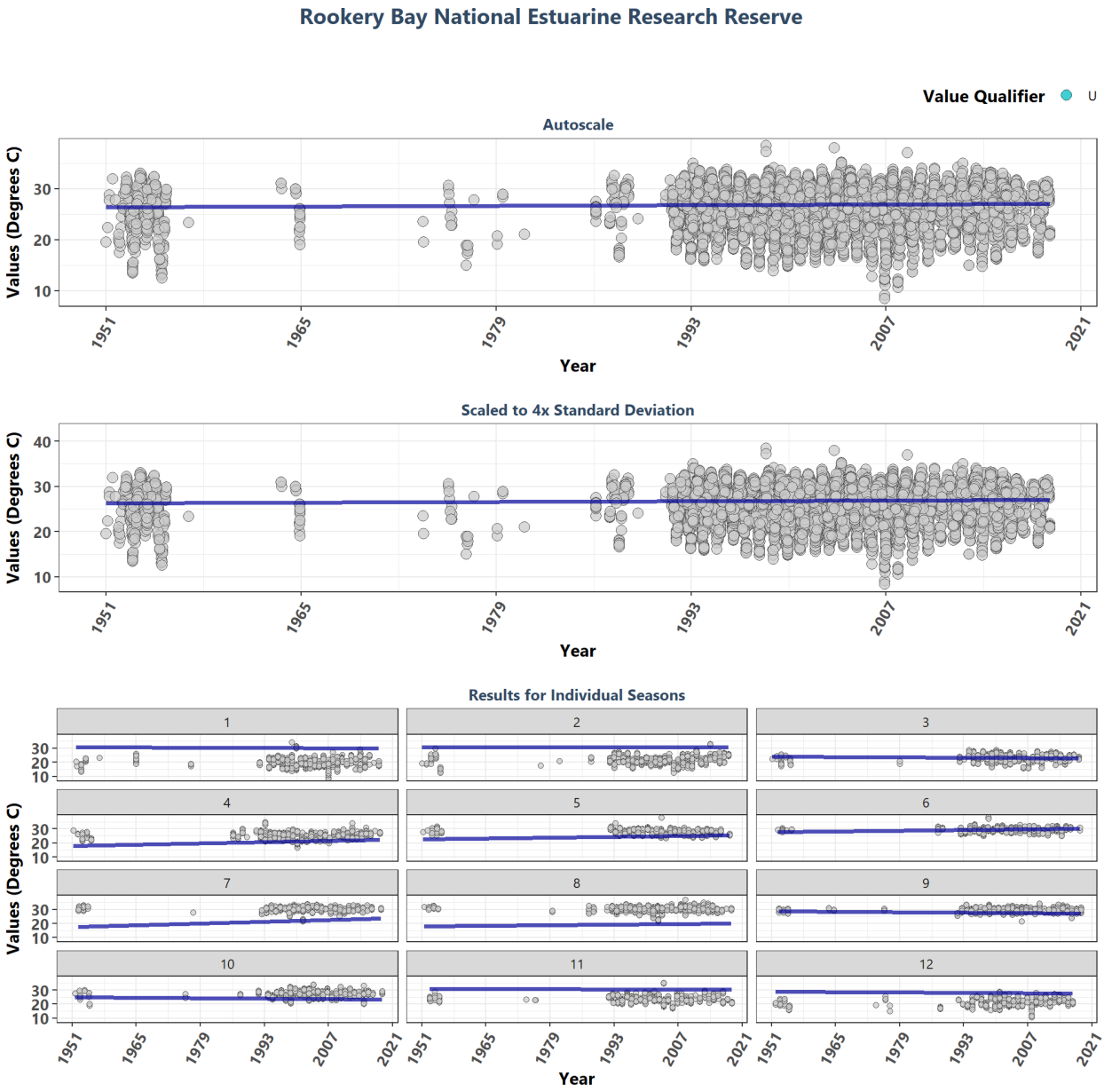
Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	151	25.06	0.0610	0.0576	23.2831	1.5	0.1257	12	0.3617	0
1	7	12.56	-0.2121	-0.1481	33.0330	-1.2	0.2296	NA	NA	0
2	12	15.87	0.0000	-0.0040	24.8833	1.3	0.1905	NA	NA	0
3	11	17.96	0.1029	0.1061	27.0506	0.0	1.0000	NA	NA	0
4	13	24.80	0.3030	0.5079	5.4540	0.0	1.0000	NA	NA	0
5	13	28.06	0.4000	0.1380	21.6829	-0.7	0.4990	NA	NA	0
6	12	30.14	-0.0385	-0.0303	32.1667	-0.9	0.3704	NA	NA	0
7	13	31.50	0.0455	0.0398	30.6402	-0.1	0.9022	NA	NA	0
8	12	31.42	0.0000	-0.0131	18.2224	0.1	0.8907	NA	NA	0
9	17	29.28	0.1667	0.1162	16.2863	0.5	0.5894	NA	NA	0
10	20	24.31	-0.4286	-0.2037	16.8333	2.4	0.0143	NA	NA	0
11	9	18.61	0.1515	0.0833	13.6944	0.5	0.6022	NA	NA	0
12	12	15.44	-0.1538	-0.1852	31.9445	0.6	0.5326	NA	NA	0

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



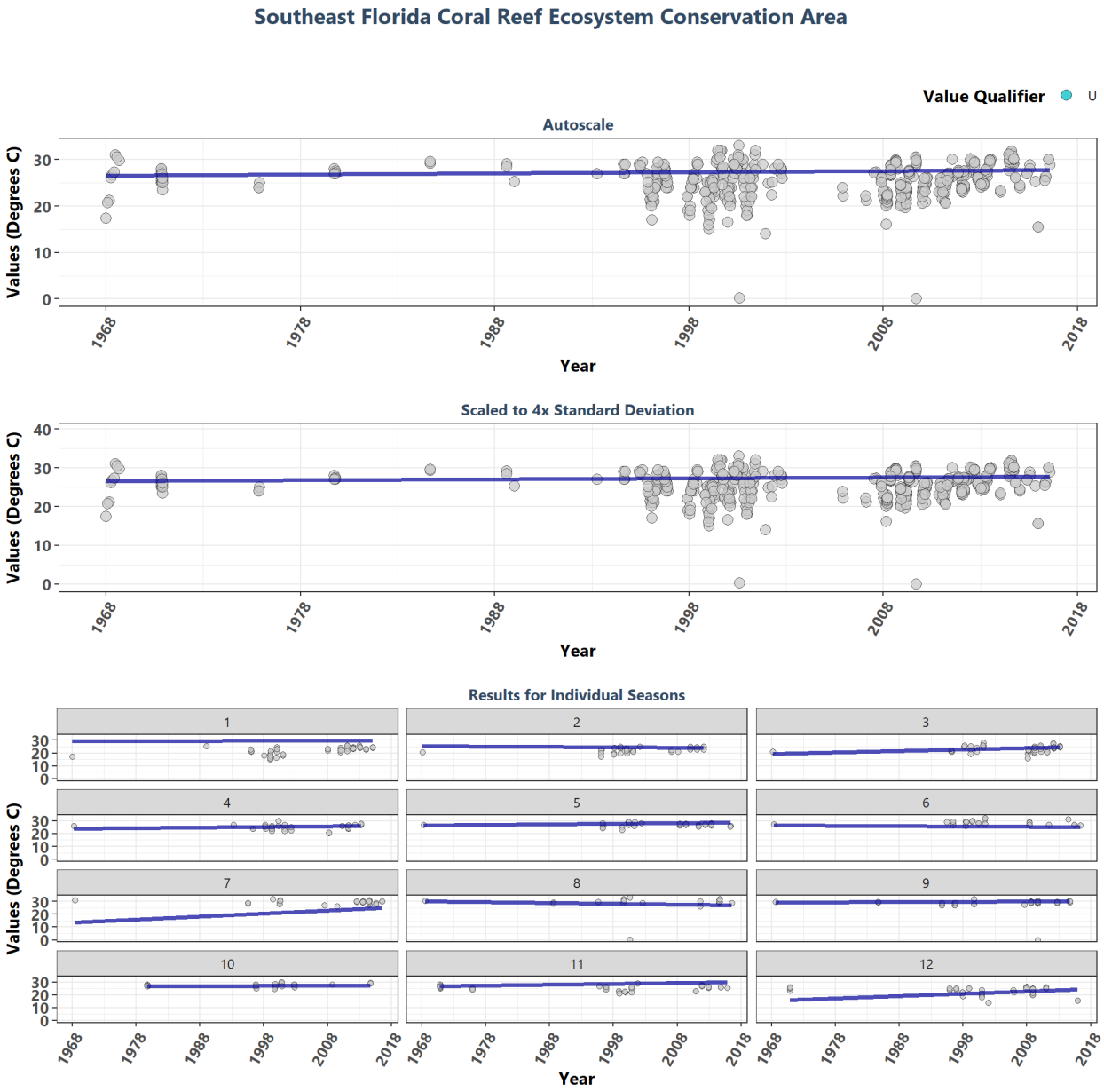
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	3433	26.59	0.0454	0.0135	25.5056	4.0	0.0001	33.7	0.0004	1
1	261	19.50	-0.0178	-0.0037	28.1926	0.3	0.7523	NA	NA	-1
2	277	20.50	0.0341	0.0154	27.1808	3.7	0.0002	NA	NA	1
3	287	23.00	-0.0023	0.0000	23.0000	-0.1	0.9547	NA	NA	-1
4	311	24.70	0.1503	0.0740	16.7260	1.2	0.2267	NA	NA	1
5	254	28.00	-0.0095	0.0000	30.6000	-0.4	0.6720	NA	NA	-1
6	295	29.51	0.0103	0.0029	23.3571	0.6	0.5610	NA	NA	1
7	278	30.60	0.0730	0.0167	29.7367	-0.2	0.8132	NA	NA	1
8	305	30.57	0.0131	0.0073	19.1268	1.9	0.0570	NA	NA	1
9	307	30.00	0.2082	0.0833	17.1833	0.1	0.9205	NA	NA	1
10	287	27.95	0.0227	0.0050	29.2600	0.9	0.3878	NA	NA	1
11	276	23.50	0.0459	0.0174	23.8304	0.3	0.7982	NA	NA	1
12	295	21.35	0.0038	0.0000	30.0000	5.3	0.0000	NA	NA	-1

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



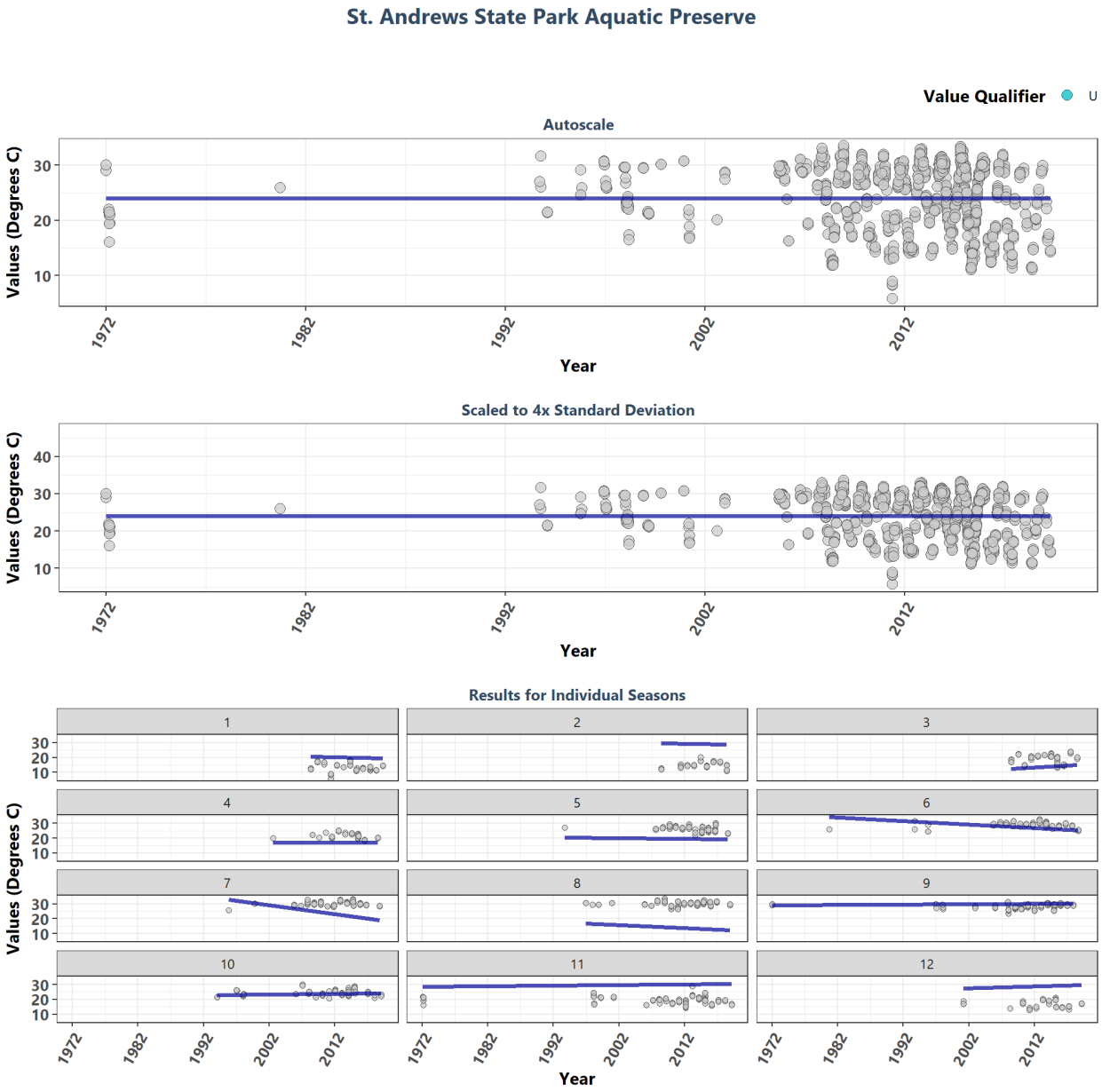
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	10721	26.50	0.0271	0.0100	26.3222	4.5	0.0000	161.7	0	1
1	865	19.70	-0.0576	-0.0133	30.6350	2.2	0.0263	NA	NA	-1
2	939	21.20	0.0075	0.0000	30.6000	5.4	0.0000	NA	NA	-1
3	803	23.10	-0.0449	-0.0220	24.2209	-1.9	0.0563	NA	NA	-1
4	922	24.70	0.1172	0.0667	17.7333	4.7	0.0000	NA	NA	1
5	853	28.02	0.1041	0.0417	22.6167	-1.9	0.0614	NA	NA	1
6	920	29.50	0.1202	0.0333	27.8333	5.5	0.0000	NA	NA	1
7	852	30.50	0.1763	0.0870	17.6522	-0.7	0.4740	NA	NA	1
8	983	30.60	0.0504	0.0308	18.1308	0.4	0.7242	NA	NA	1
9	918	29.95	-0.0537	-0.0231	28.7319	-2.6	0.0088	NA	NA	-1
10	926	27.56	-0.0610	-0.0222	24.8111	-2.5	0.0143	NA	NA	-1
11	820	23.70	-0.0164	-0.0032	30.6645	-2.6	0.0088	NA	NA	-1
12	920	22.00	-0.0427	-0.0125	28.6700	8.0	0.0000	NA	NA	-1

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



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	593	26.17	0.1158	0.0249	26.5466	3.4	0.0006	65.2	0	1
1	58	23.34	0.0206	0.0057	29.2729	5.5	0.0000	NA	NA	1
2	35	23.00	-0.0675	-0.0247	25.4004	2.9	0.0039	NA	NA	-1
3	78	23.10	0.3395	0.1100	19.4800	3.4	0.0007	NA	NA	1
4	34	25.45	0.1586	0.0500	23.7500	1.3	0.1864	NA	NA	1
5	65	27.20	0.2233	0.0435	26.6522	0.8	0.4184	NA	NA	1
6	30	28.00	-0.2083	-0.0245	26.5490	-1.5	0.1287	NA	NA	-1
7	36	29.53	0.4936	0.2258	13.7371	0.2	0.8676	NA	NA	1
8	24	30.00	-0.1954	-0.0643	30.0893	0.6	0.5682	NA	NA	-1
9	65	29.38	0.0833	0.0192	29.1747	3.4	0.0007	NA	NA	1
10	25	28.00	0.0678	0.0160	26.5441	1.6	0.1098	NA	NA	1
11	92	26.50	0.2817	0.0644	26.7408	-3.4	0.0007	NA	NA	1
12	51	24.44	0.2584	0.1855	15.4959	-0.7	0.4814	NA	NA	1

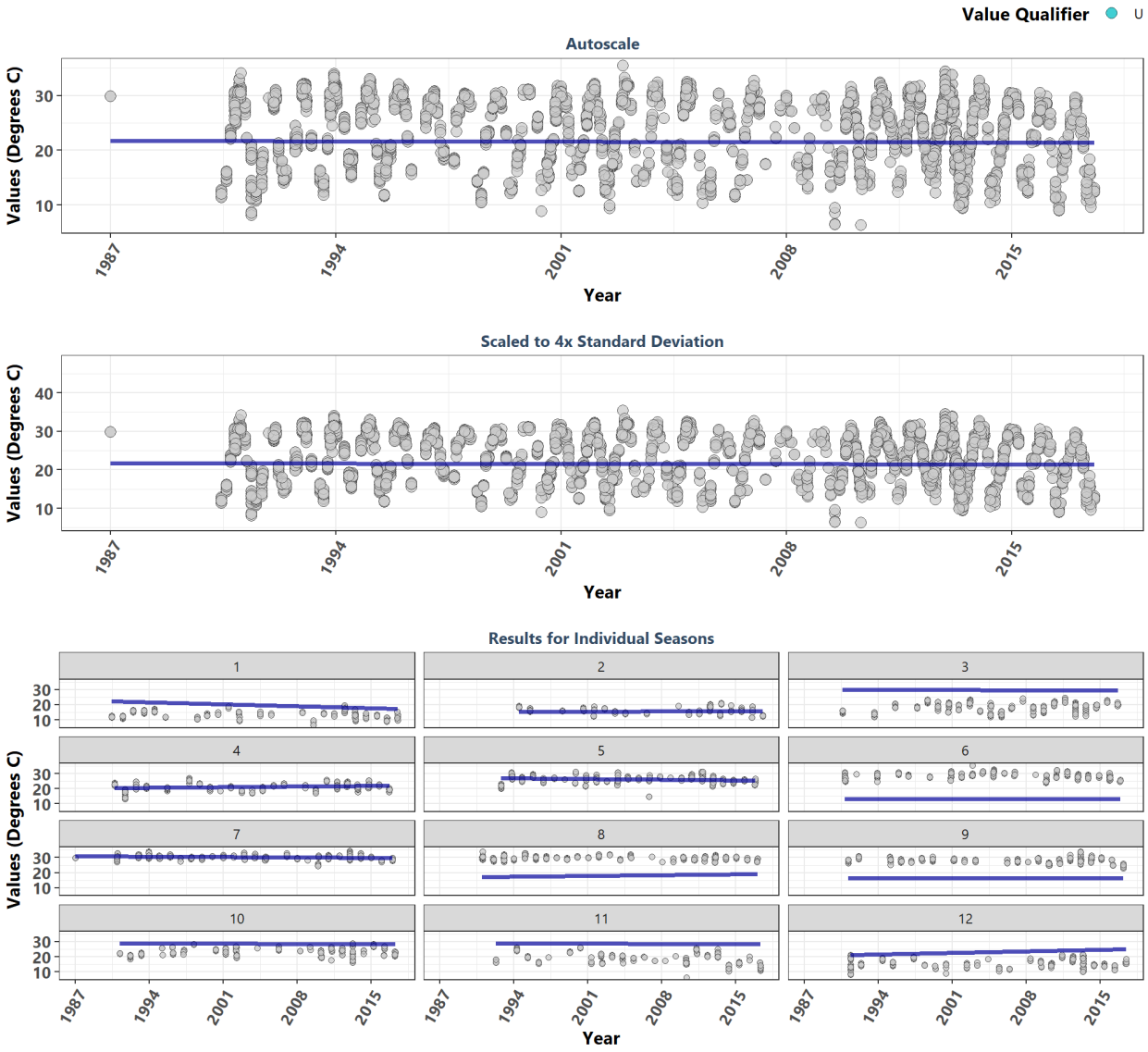
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	1015	25.90	-0.0120	0.0000	24.0000	-0.2	0.8640	62.6	0	0
1	49	13.20	-0.0399	-0.1000	24.4000	-1.4	0.1471	NA	NA	0
2	45	14.50	-0.1449	-0.0857	32.7071	2.7	0.0075	NA	NA	0
3	58	20.10	0.2727	0.2600	3.0600	-0.4	0.6597	NA	NA	0
4	41	21.40	0.0028	0.0000	16.9000	-3.5	0.0006	NA	NA	0
5	106	26.10	-0.0929	-0.0383	21.0814	-4.1	0.0000	NA	NA	0
6	104	29.15	-0.2652	-0.2375	36.3125	-2.2	0.0277	NA	NA	0
7	113	30.20	-0.3671	-0.6000	47.2000	0.8	0.3964	NA	NA	0
8	114	30.20	-0.1429	-0.2000	21.8000	1.7	0.0820	NA	NA	0
9	117	28.60	0.0536	0.0258	29.1150	3.3	0.0008	NA	NA	0
10	99	23.90	0.1159	0.0563	21.5375	1.7	0.0847	NA	NA	0
11	122	19.55	0.1098	0.0429	28.4857	-1.5	0.1279	NA	NA	0
12	47	16.90	0.2076	0.1250	23.6000	0.0	0.9851	NA	NA	0

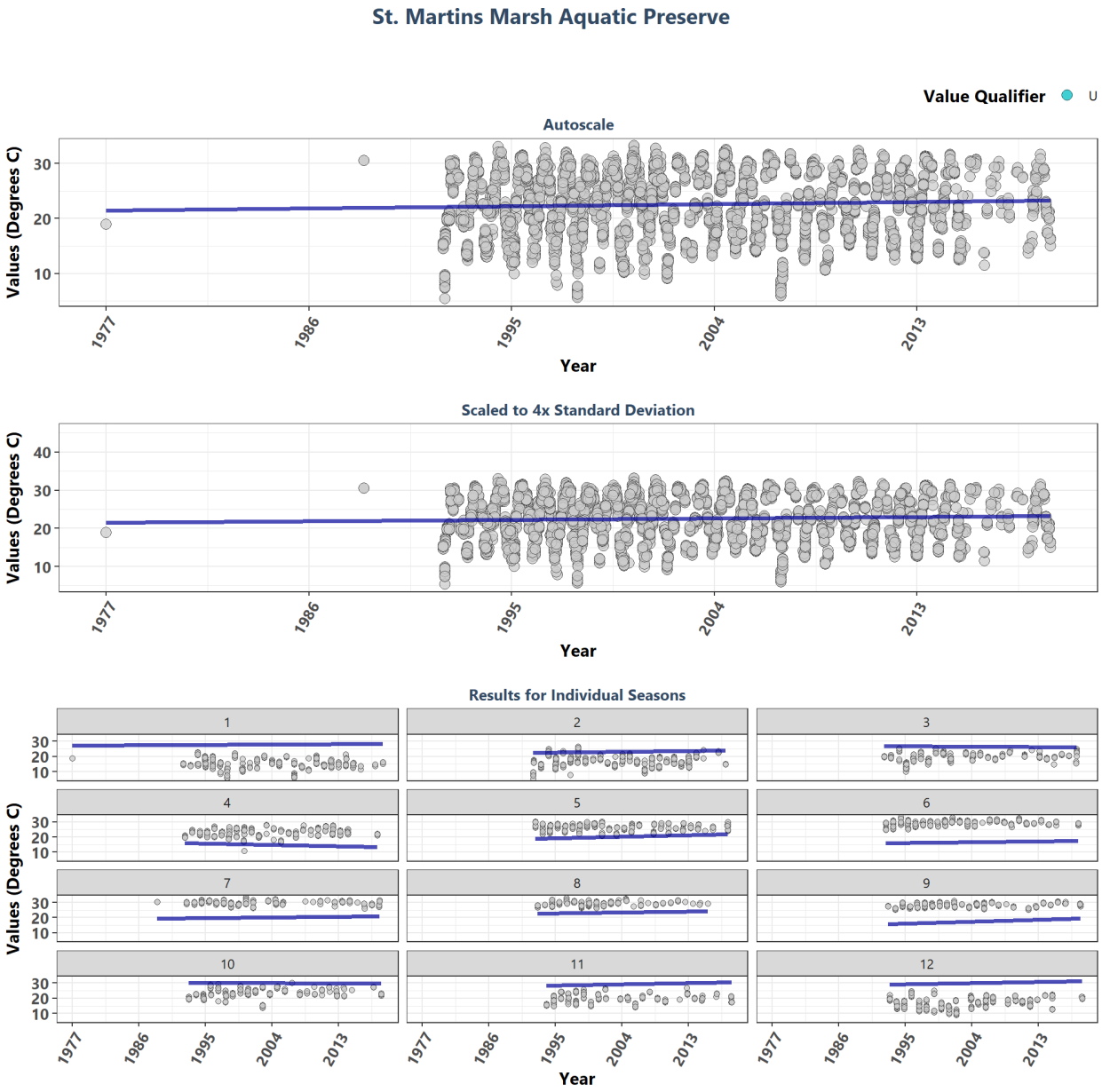
[a] $p < 0.00005$ appear as 0 due to rounding

St. Joseph Bay Aquatic Preserve



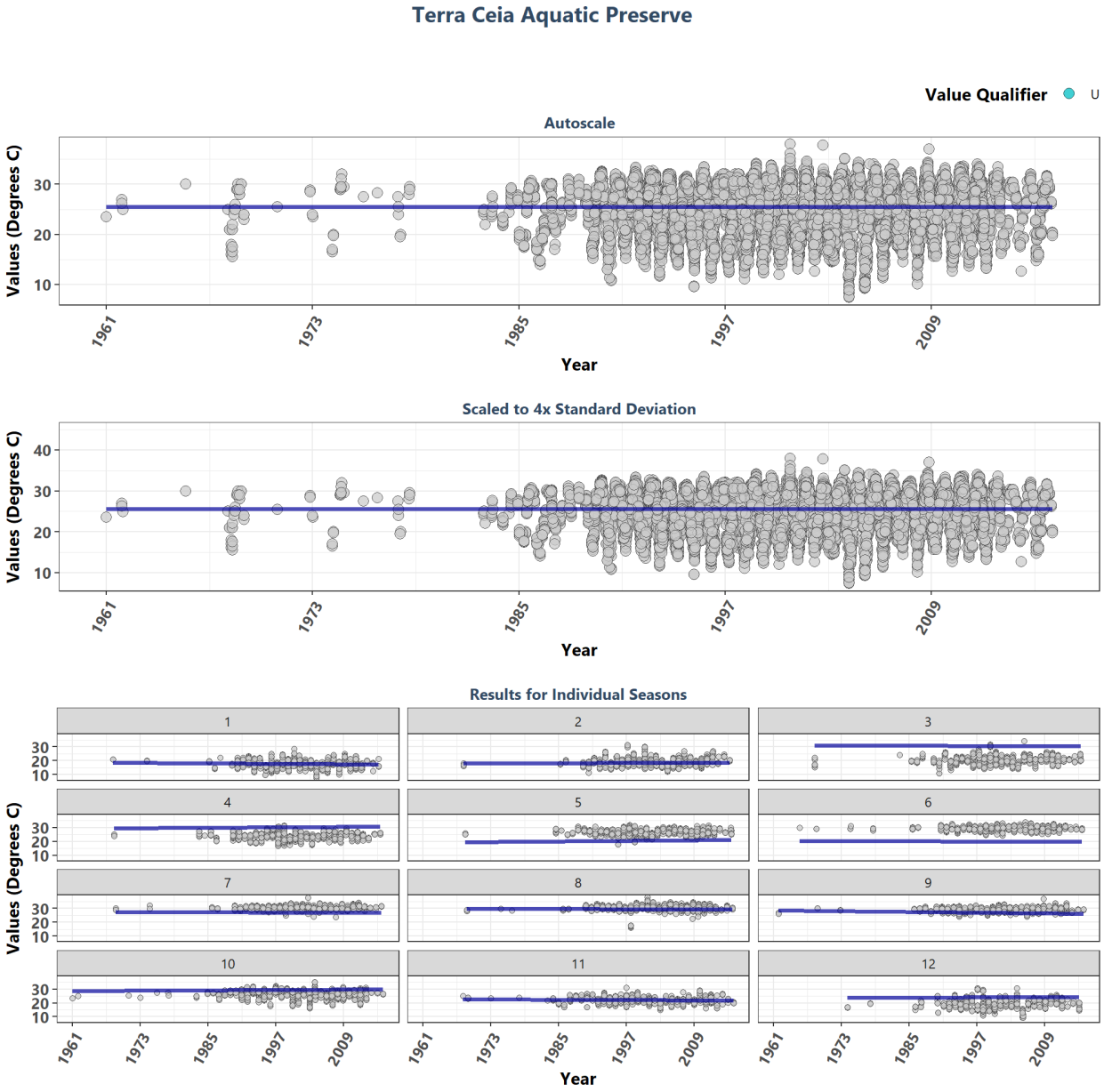
Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	4104	25.30	-0.0206	-0.0111	21.7338	-2.9	0.0033	147.8	0	-1
1	281	13.20	-0.2141	-0.1824	22.9176	0.0	0.9975	NA	NA	-1
2	212	16.60	0.0316	0.0158	15.3632	-0.2	0.8379	NA	NA	1
3	391	18.30	-0.0687	-0.0187	29.8875	3.3	0.0010	NA	NA	-1
4	327	21.10	0.1187	0.0500	20.3500	3.2	0.0013	NA	NA	1
5	391	26.30	-0.1214	-0.0556	27.2444	-3.6	0.0003	NA	NA	-1
6	452	28.65	0.0002	0.0000	13.2000	-1.0	0.3293	NA	NA	-1
7	497	30.50	-0.1758	-0.0429	31.0571	-5.9	0.0000	NA	NA	-1
8	465	29.70	0.1115	0.0800	16.8600	-2.2	0.0265	NA	NA	1
9	317	28.50	-0.0094	0.0000	16.6000	-2.0	0.0466	NA	NA	-1
10	281	24.00	-0.0306	-0.0111	28.8611	7.0	0.0000	NA	NA	-1
11	221	20.00	-0.0747	-0.0222	28.9889	-4.8	0.0000	NA	NA	-1
12	269	15.60	0.2777	0.1500	20.5500	0.8	0.4382	NA	NA	1

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



Season	N	Median	tau	Slope	Int.	z	p__z	chi__sq	p__chi__sq	Trend
All	5737	22.40	0.0845	0.0417	21.4806	8.5	0.0000	148.4	0	1
1	577	15.10	0.0695	0.0250	27.3300	-4.5	0.0000	NA	NA	1
2	741	16.60	0.0976	0.0538	21.6000	3.2	0.0014	NA	NA	1
3	428	19.70	-0.0510	-0.0250	27.1500	5.8	0.0000	NA	NA	-1
4	515	23.00	-0.1255	-0.0969	17.5219	3.3	0.0009	NA	NA	-1
5	588	26.50	0.1874	0.1167	17.0167	-1.9	0.0642	NA	NA	1
6	553	29.10	0.0785	0.0571	15.0000	7.7	0.0000	NA	NA	1
7	326	30.00	0.1003	0.0522	18.8000	-0.9	0.3915	NA	NA	1
8	270	29.54	0.1309	0.0778	21.3611	6.4	0.0000	NA	NA	1
9	362	27.93	0.1778	0.1375	13.8250	2.0	0.0480	NA	NA	1
10	476	23.15	-0.0318	-0.0091	30.2091	4.3	0.0000	NA	NA	-1
11	380	20.00	0.2180	0.0818	27.1364	2.9	0.0034	NA	NA	1
12	521	17.40	0.2606	0.0909	27.4491	6.1	0.0000	NA	NA	1

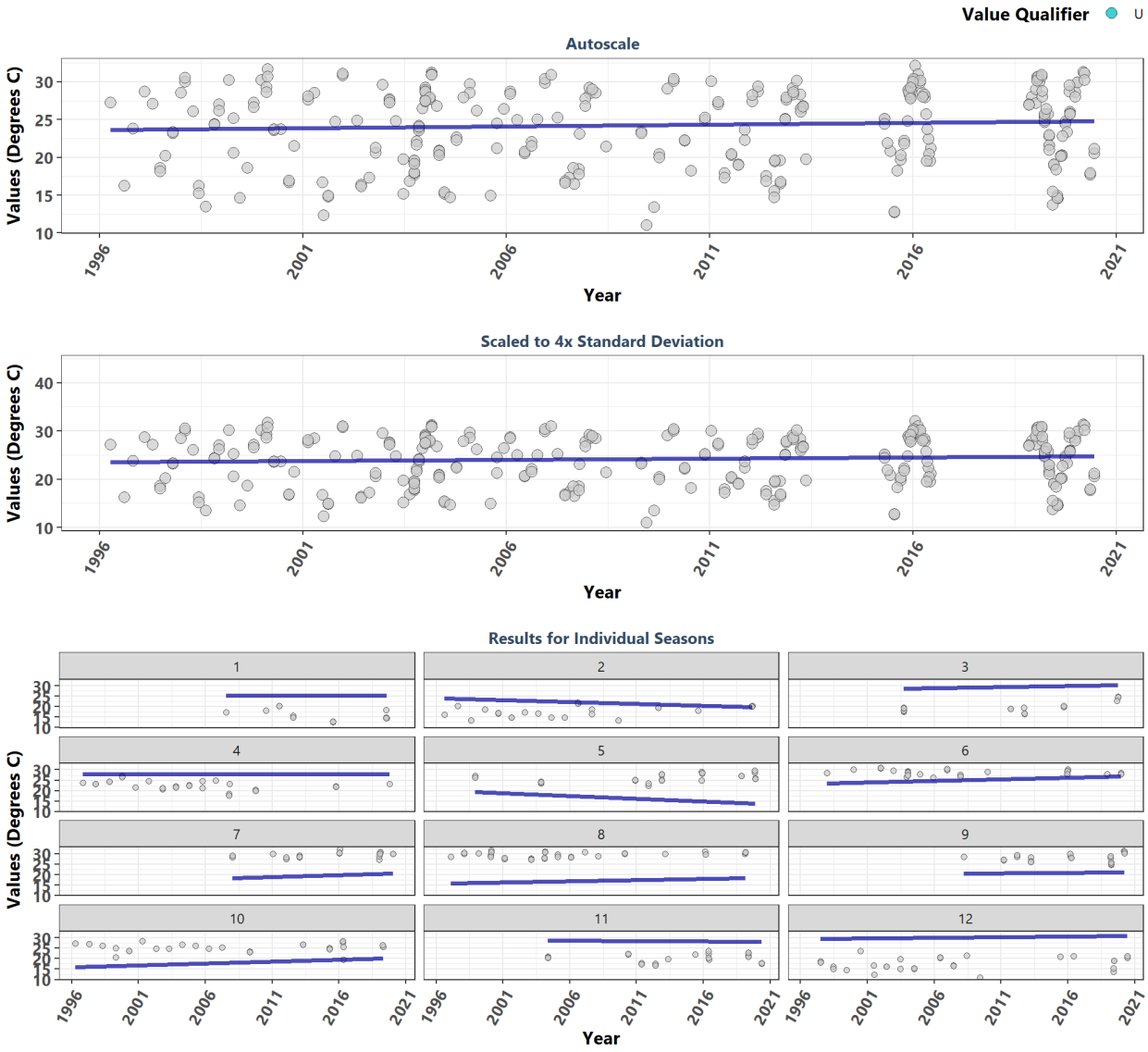
[a] $p < 0.00005$ appear as 0 due to rounding



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	17636	26.10	-0.0035	0.0000	25.5631	-1.2	0.2226	128.3	0	0
1	1301	17.70	-0.0360	-0.0250	18.7000	-1.9	0.0516	NA	NA	0
2	1230	18.60	0.0171	0.0086	18.2571	0.9	0.3683	NA	NA	0
3	1322	20.59	-0.0535	-0.0145	31.2709	2.9	0.0039	NA	NA	0
4	1333	24.05	0.1090	0.0250	29.5400	0.6	0.5351	NA	NA	0
5	1544	27.00	0.0529	0.0333	19.2567	-1.1	0.2669	NA	NA	0
6	1392	29.40	-0.0105	-0.0081	20.3250	-1.8	0.0787	NA	NA	0
7	1510	30.56	-0.0188	-0.0064	27.2571	6.4	0.0000	NA	NA	0
8	1897	30.66	-0.0314	-0.0077	29.7077	-3.5	0.0005	NA	NA	0
9	1581	29.40	-0.0966	-0.0429	28.5236	4.3	0.0000	NA	NA	0
10	1720	26.90	0.0723	0.0188	28.6500	-6.0	0.0000	NA	NA	0
11	1451	22.10	-0.0313	-0.0143	22.6714	-1.8	0.0741	NA	NA	0
12	1355	20.00	0.0113	0.0048	23.8690	-0.6	0.5615	NA	NA	0

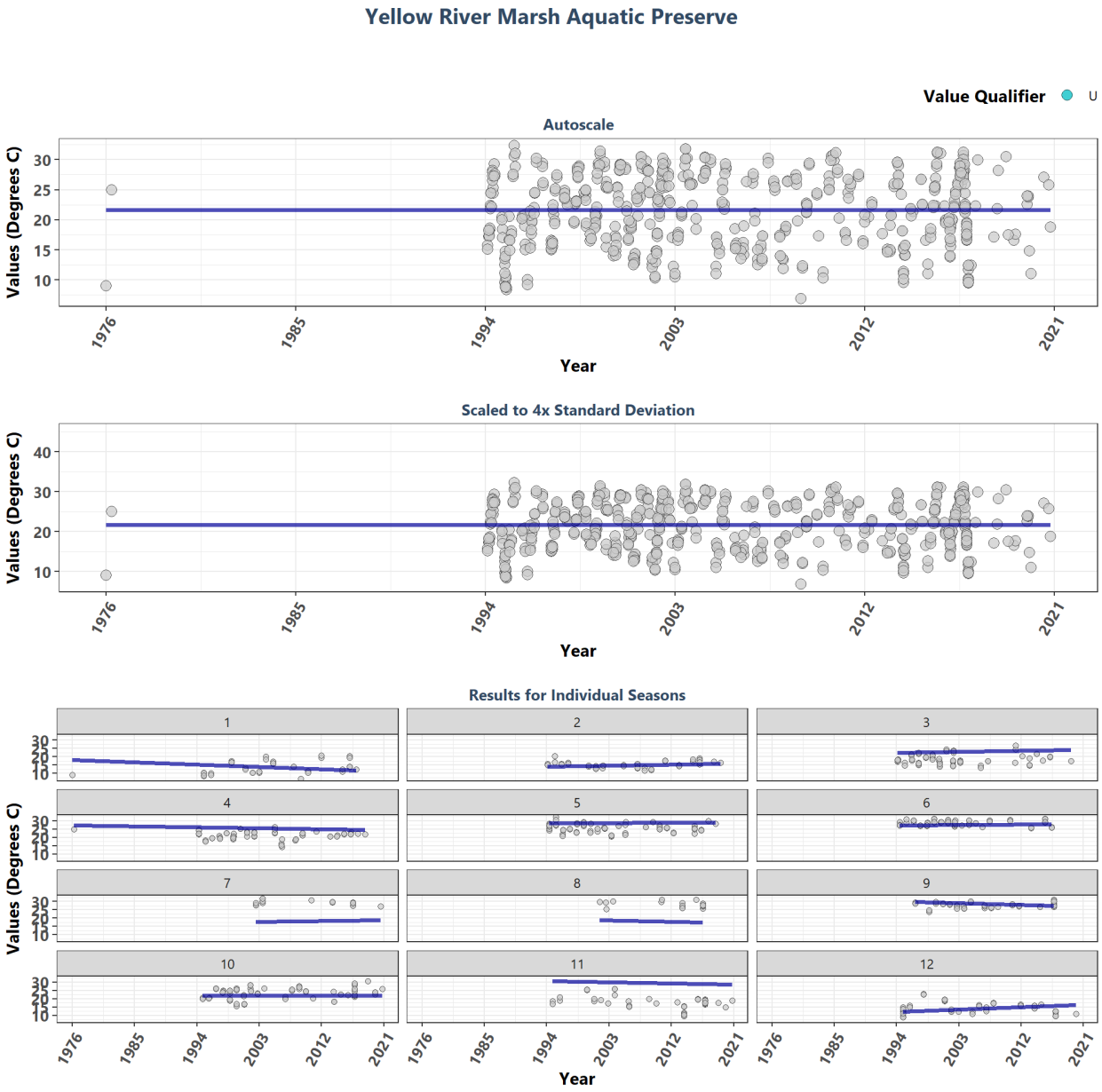
[a] $p < 0.00005$ appear as 0 due to rounding

Tomoka Marsh Aquatic Preserve



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	295	24.70	0.0978	0.0489	23.5931	2.4	0.0177	24.9	0.0093	1
1	13	14.90	0.0256	0.0145	25.0908	-1.5	0.1276	NA	NA	1
2	24	17.74	-0.2646	-0.1689	23.9006	2.1	0.0342	NA	NA	-1
3	17	19.10	0.2398	0.1114	27.6714	2.3	0.0218	NA	NA	1
4	26	22.46	0.0000	0.0004	27.9949	-1.9	0.0601	NA	NA	1
5	27	25.70	-0.3205	-0.2639	20.1775	2.8	0.0049	NA	NA	-1
6	30	28.60	0.3818	0.1420	23.2857	-0.5	0.5847	NA	NA	1
7	19	29.90	0.3971	0.1983	15.9270	1.4	0.1514	NA	NA	1
8	35	29.88	0.1667	0.1087	15.7767	1.3	0.1977	NA	NA	1
9	22	28.00	0.0826	0.0467	20.0533	0.0	1.0000	NA	NA	1
10	27	25.25	0.3116	0.1750	15.9065	0.2	0.8670	NA	NA	1
11	27	20.80	-0.0713	-0.0375	28.9563	0.6	0.5506	NA	NA	-1
12	28	16.75	0.1529	0.0615	29.3907	1.2	0.2190	NA	NA	1

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



Season	N	Median	tau	Slope	Int.	z	p_z	chi_sq	p_chi_sq	Trend
All	586	22.10	-0.0039	0.0000	21.6500	0.1	0.9180	24.6	0.0106	0
1	37	13.60	-0.2090	-0.1500	17.9500	2.5	0.0135	NA	NA	0
2	45	15.10	0.1465	0.0690	12.9595	1.4	0.1569	NA	NA	0
3	65	17.70	0.0889	0.0667	21.2000	0.9	0.3936	NA	NA	0
4	64	21.90	-0.1019	-0.0615	27.2000	0.0	0.9815	NA	NA	0
5	69	25.60	0.0189	0.0100	28.5400	-1.2	0.2166	NA	NA	0
6	55	28.80	0.0717	0.0300	26.5700	0.2	0.8440	NA	NA	0
7	19	29.20	0.0726	0.0471	16.5706	-1.7	0.0857	NA	NA	0
8	21	27.80	-0.1828	-0.1000	21.4000	-1.2	0.2381	NA	NA	0
9	45	27.50	-0.1810	-0.1114	31.9230	0.7	0.4854	NA	NA	0
10	71	23.20	0.0025	0.0000	21.9000	1.1	0.2720	NA	NA	0
11	45	17.70	-0.2807	-0.0667	31.6667	-1.8	0.0737	NA	NA	0
12	50	13.45	0.2838	0.1714	8.8000	-2.2	0.0309	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)])
```

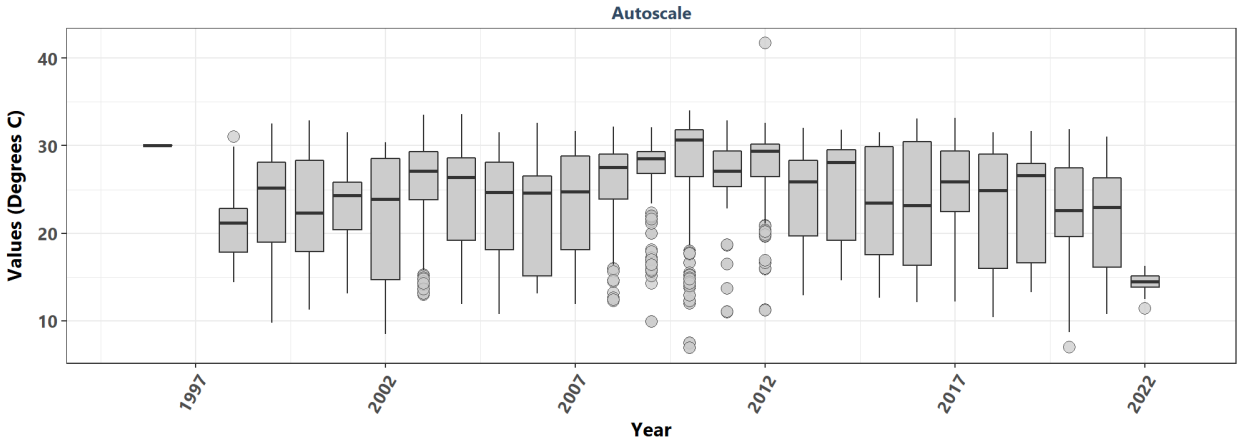
```
x_scale <- ifelse(year_upper - year_lower > 30, 10, 5)
y_scale <- mn_RV + 4 * sd_RV

##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

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

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

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

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

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

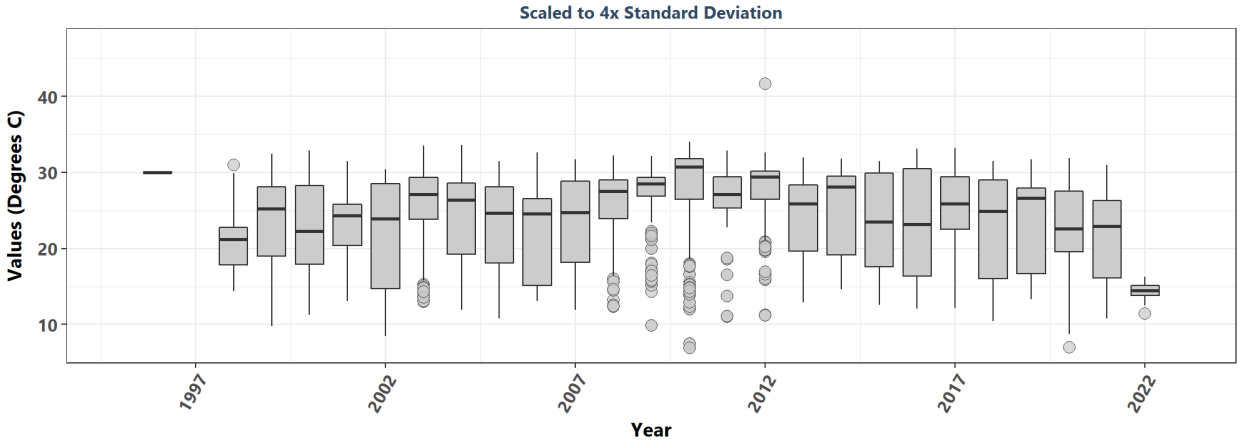
```
                  group=YearMonth, color=as.factor(Month))) +
   geom_boxplot(fill="#cccccc", outlier.size=1.5, outlier.alpha=0.75) +
   labs(subtitle="Autoscale",
        x="Year", y=paste0("Values (", unit, ")"), color="Month") +
   scale_x_continuous(limits=c(year_lower - 1, year_upper + 1),
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

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

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

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

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

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

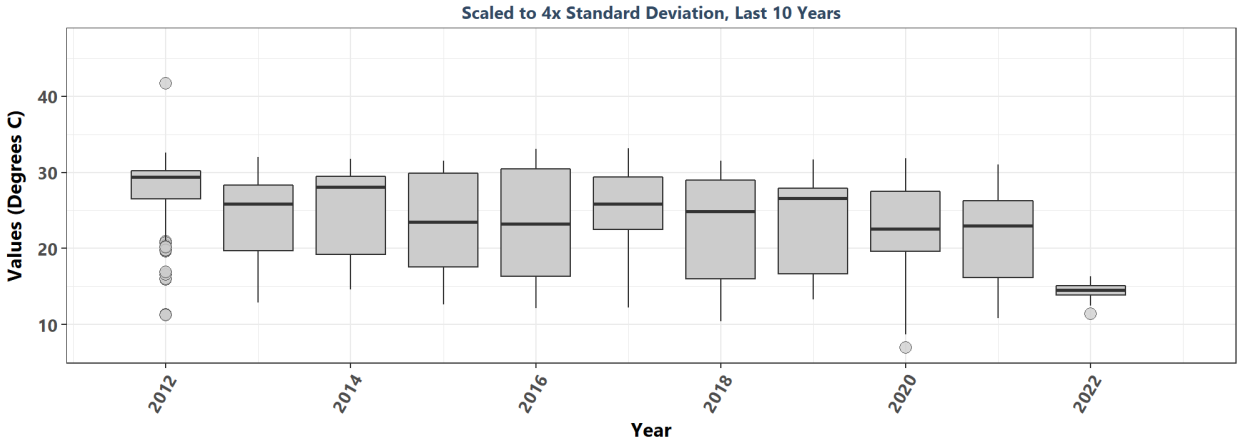
```
            x="Month", y=paste0("Values (", unit, ")"), fill="Month") +
        scale_x_continuous(limits=c(0, 13), breaks=seq(3, 12, 3)) +
        plot_theme +
        theme(legend.position="none")

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

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

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

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

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

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

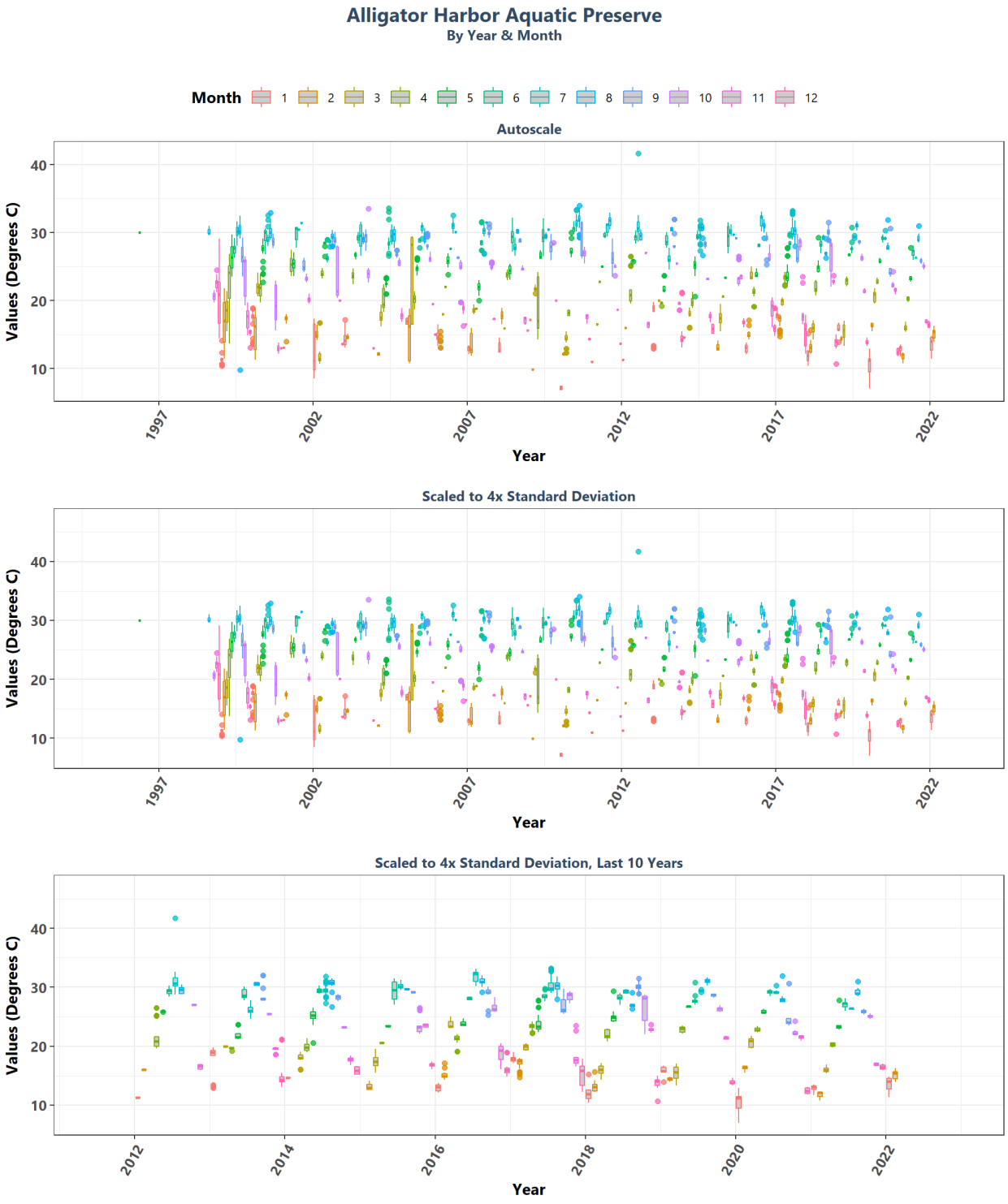
Alligator Harbor Aquatic Preserve

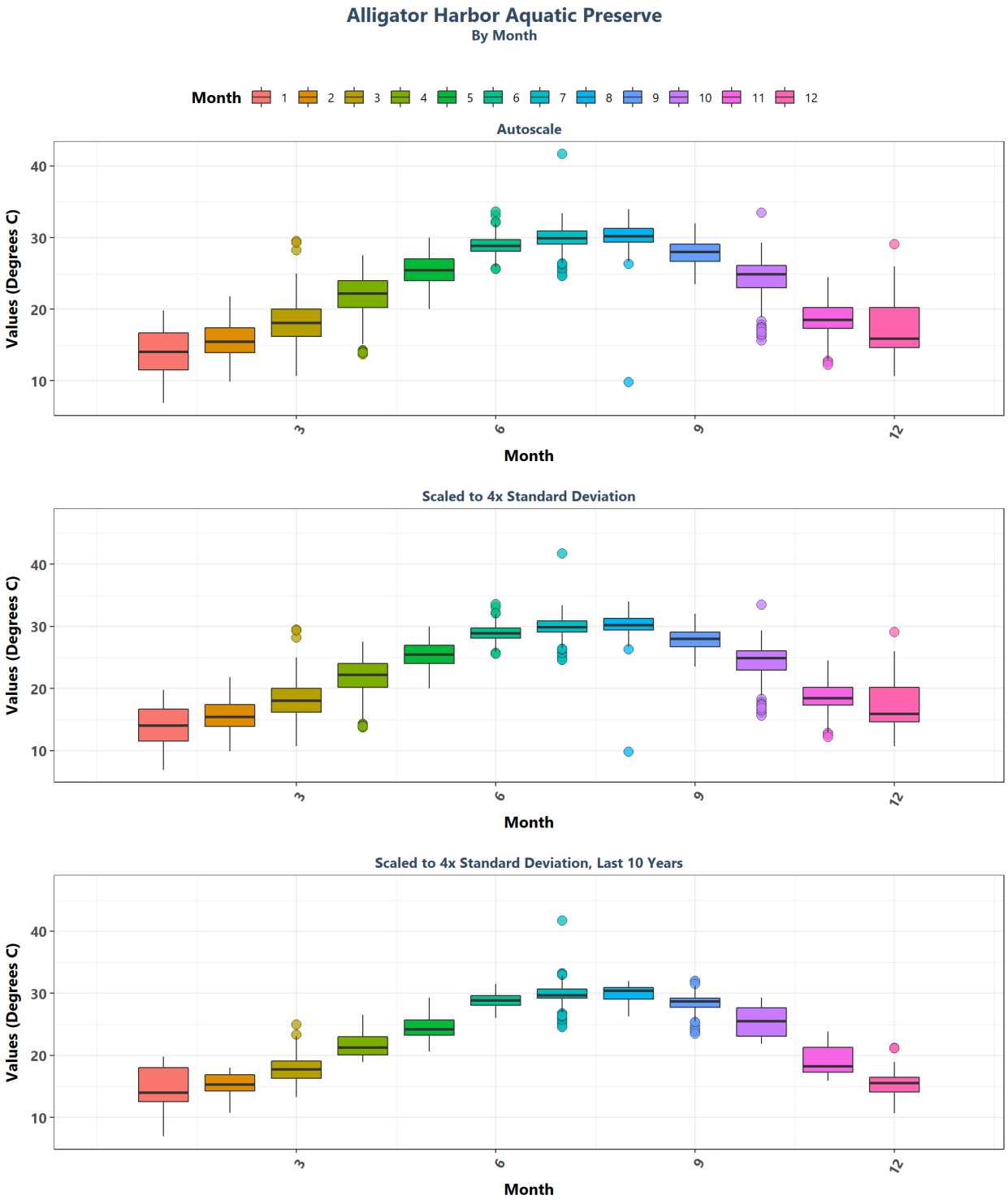
By Year





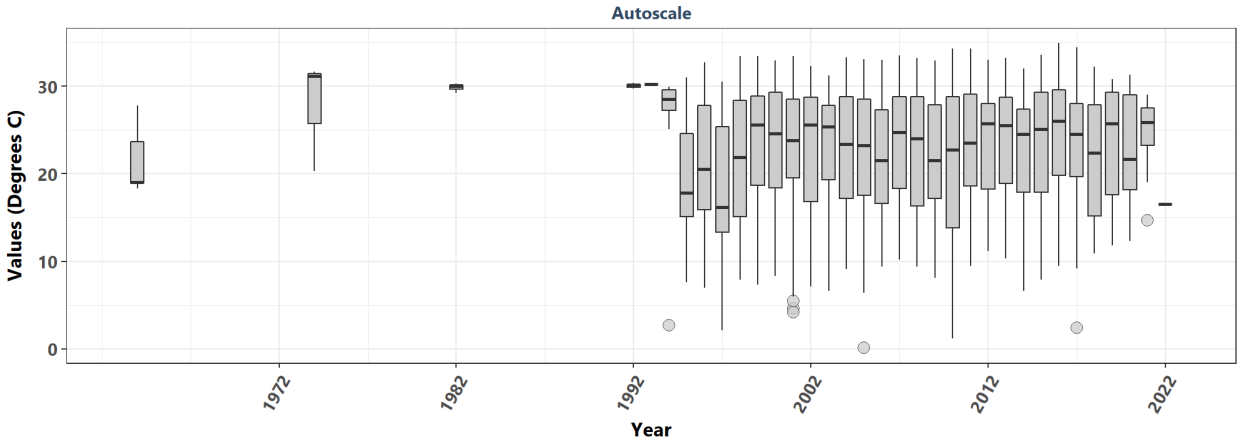


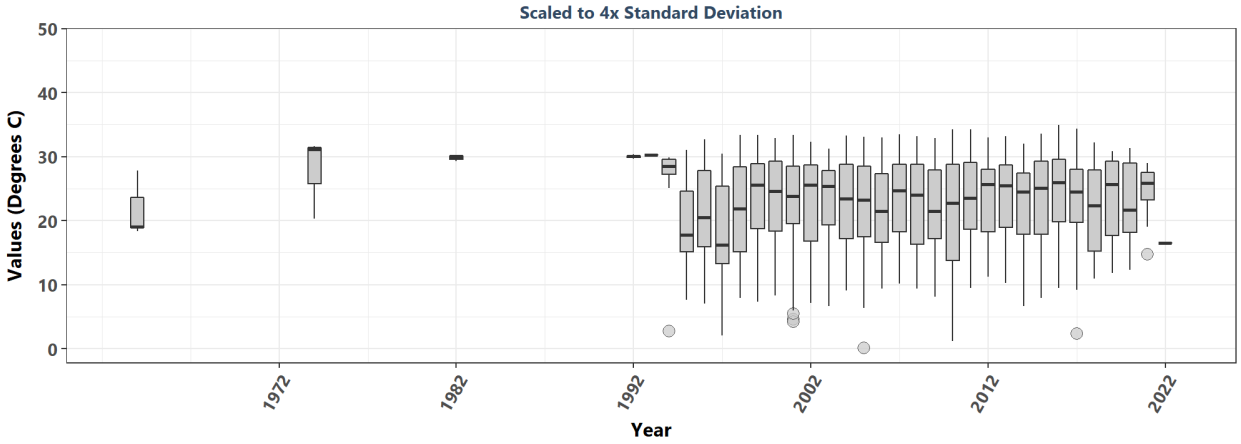

Alligator Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

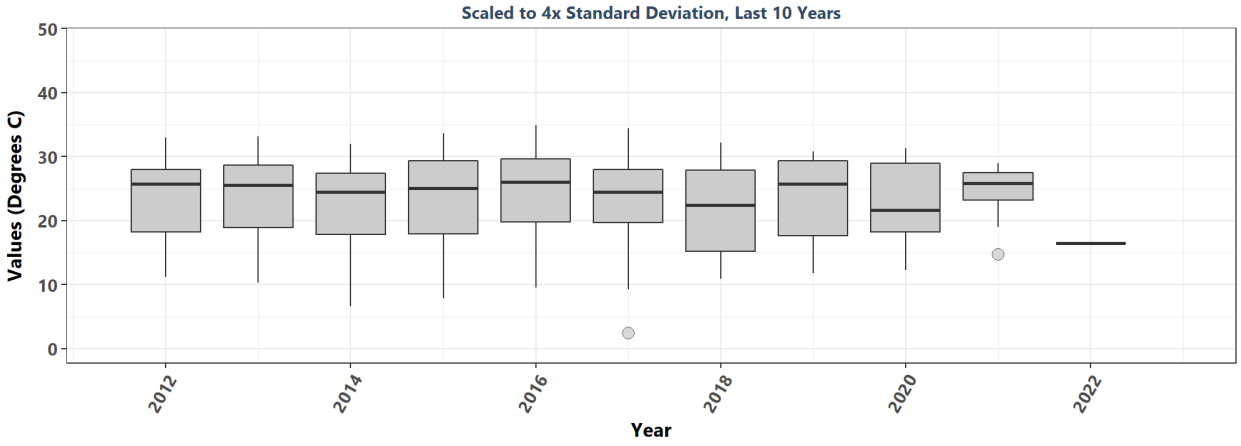

Alligator Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

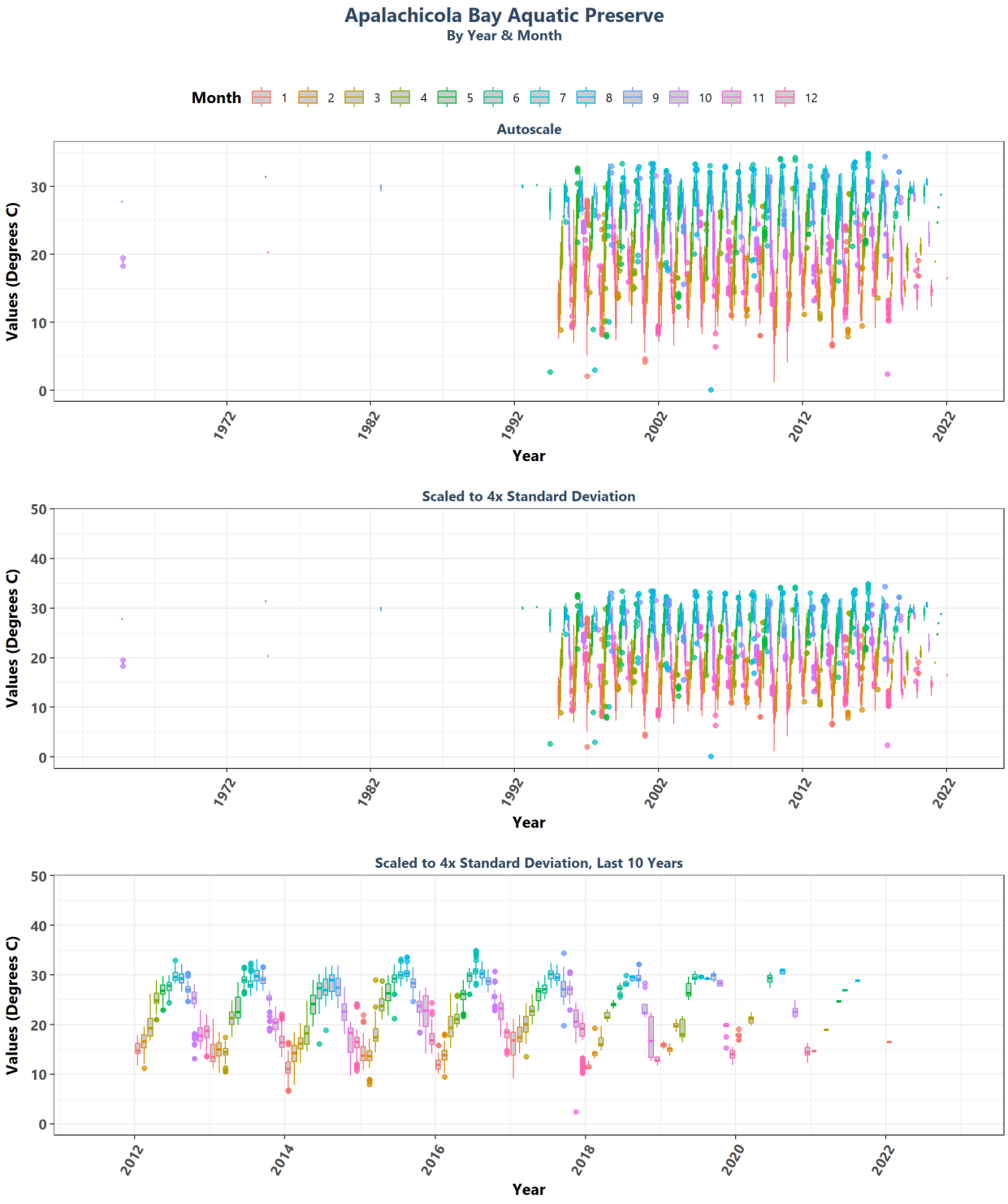
Apalachicola Bay Aquatic Preserve

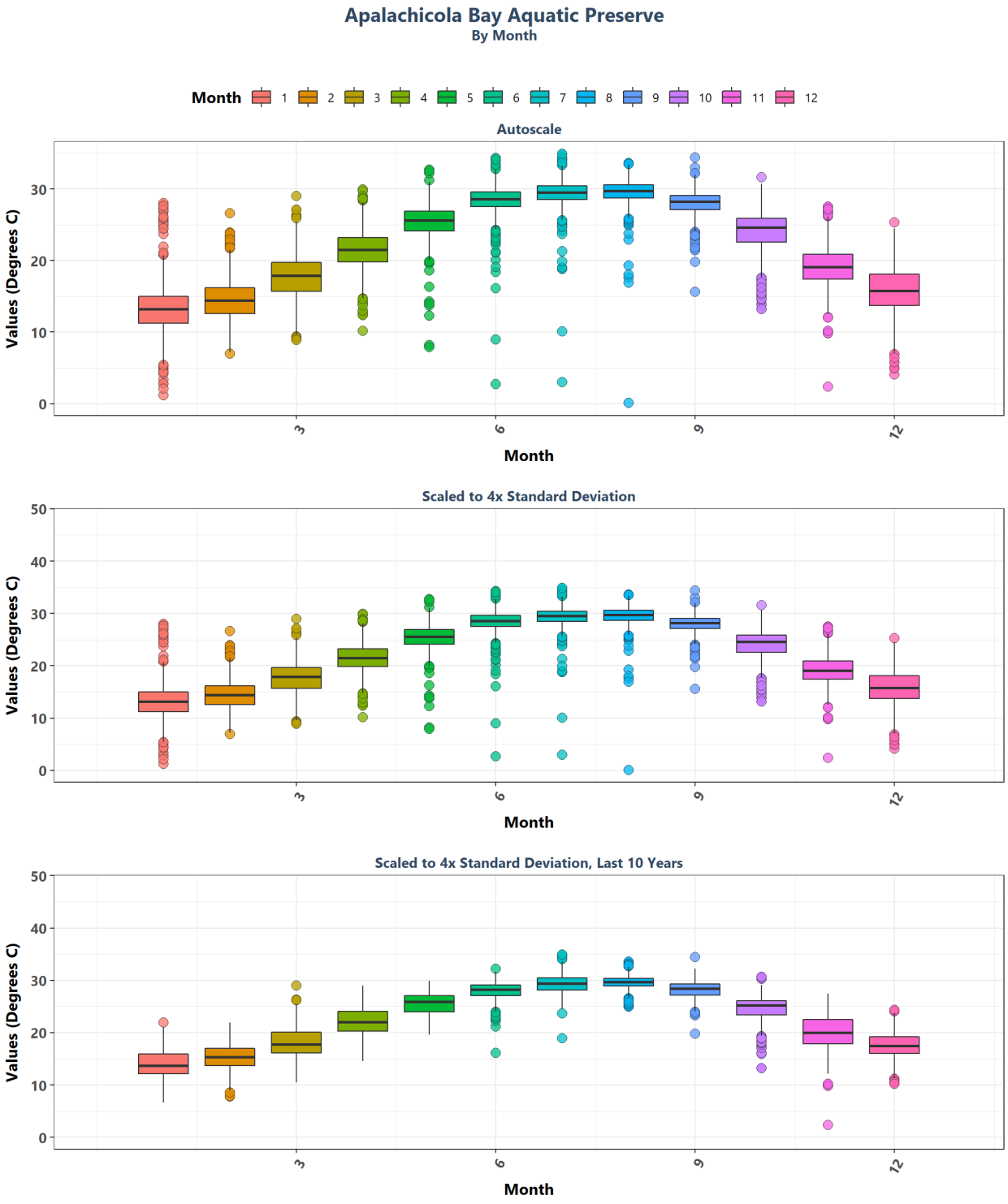
By Year





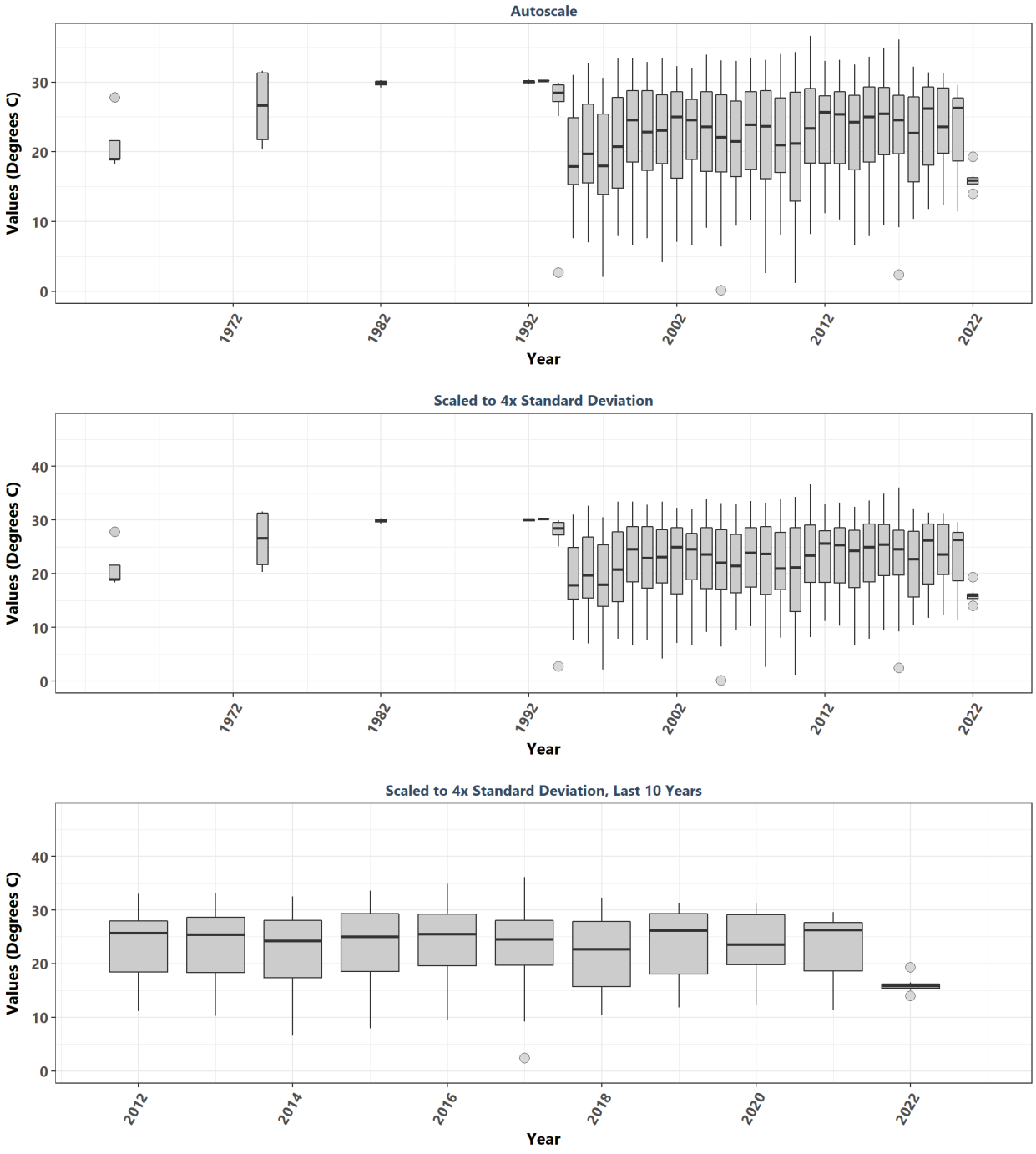


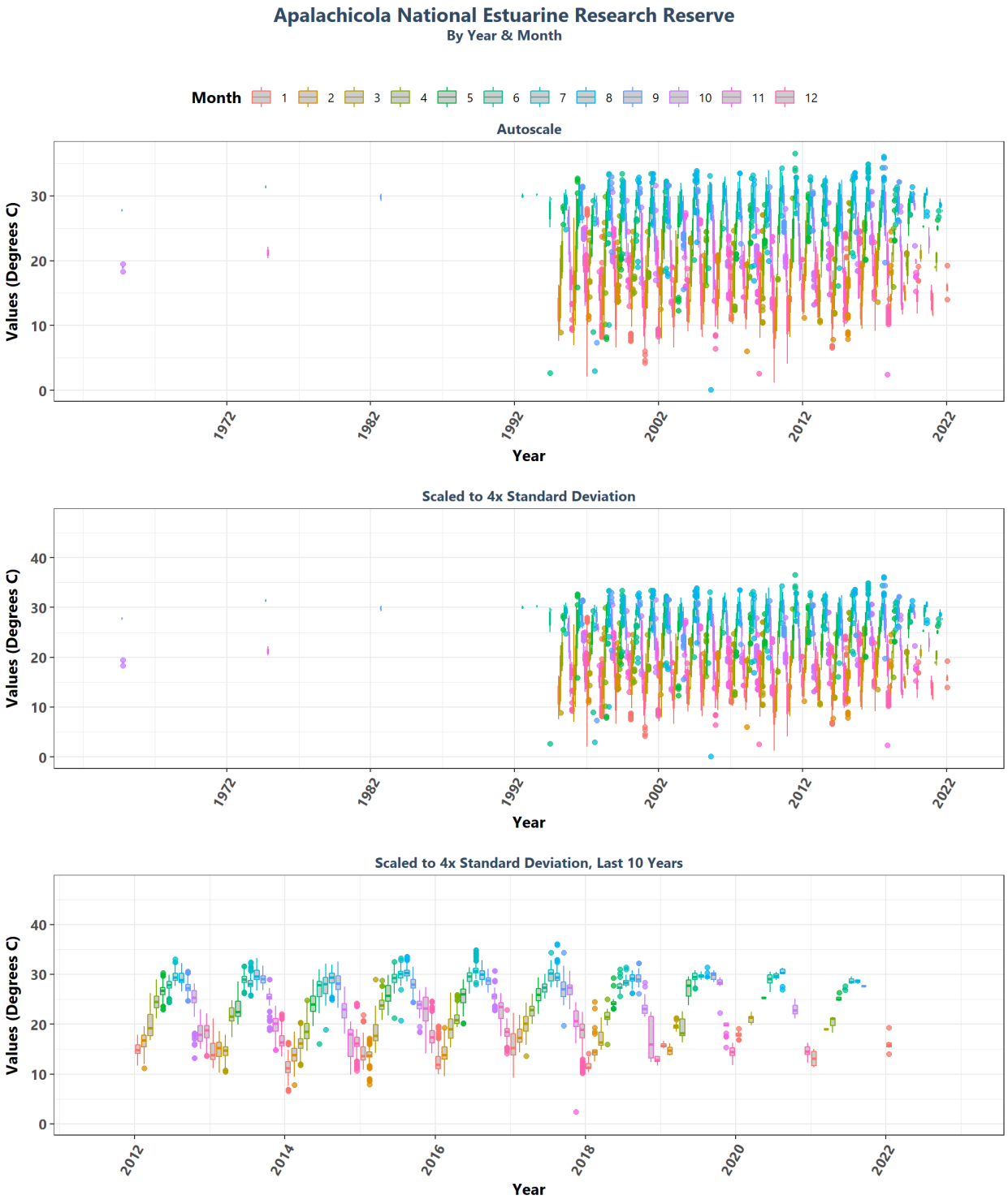

Apalachicola Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

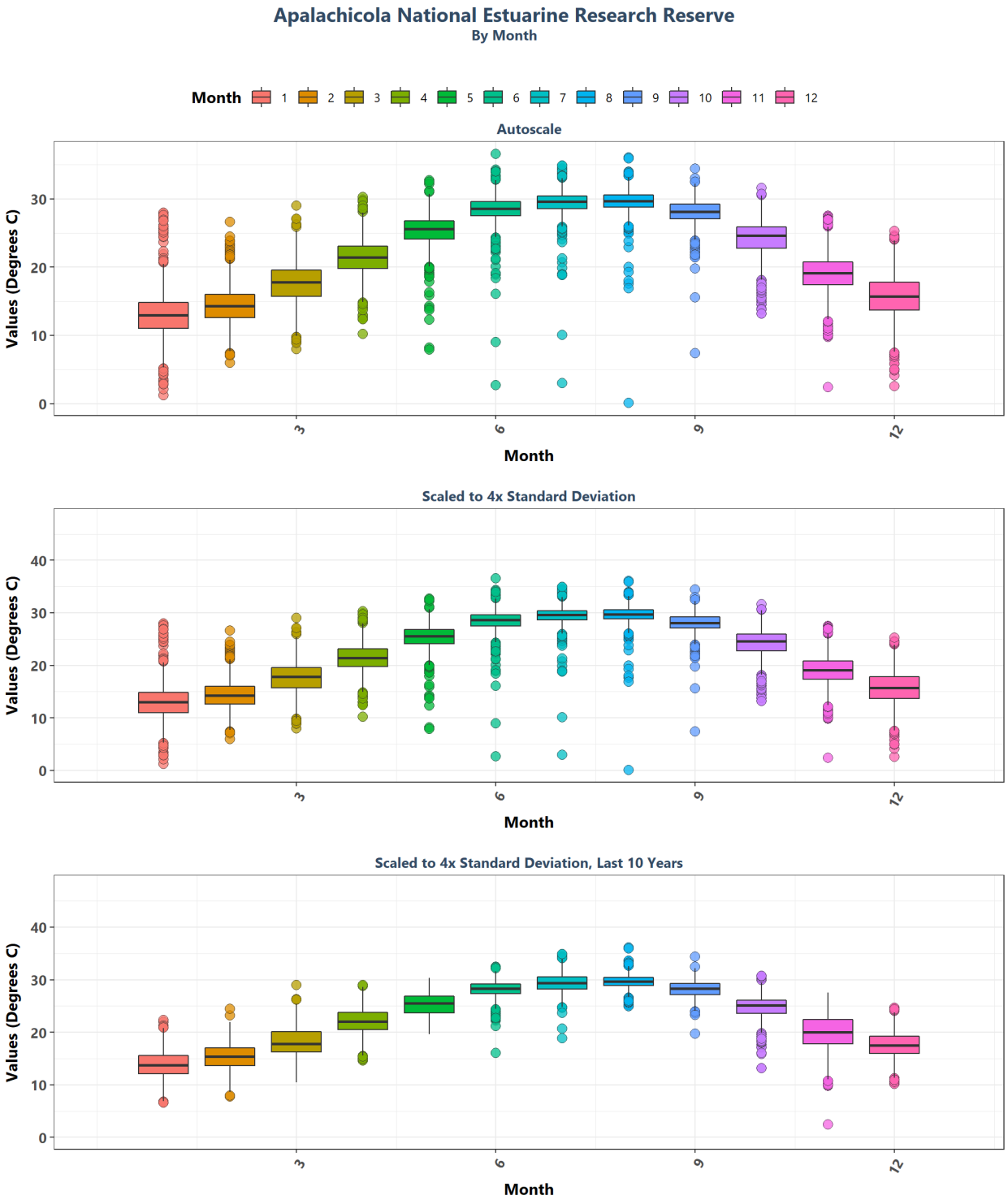

Apalachicola Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40 50
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30 40 50
3 6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

Apalachicola National Estuarine Research Reserve

By Year

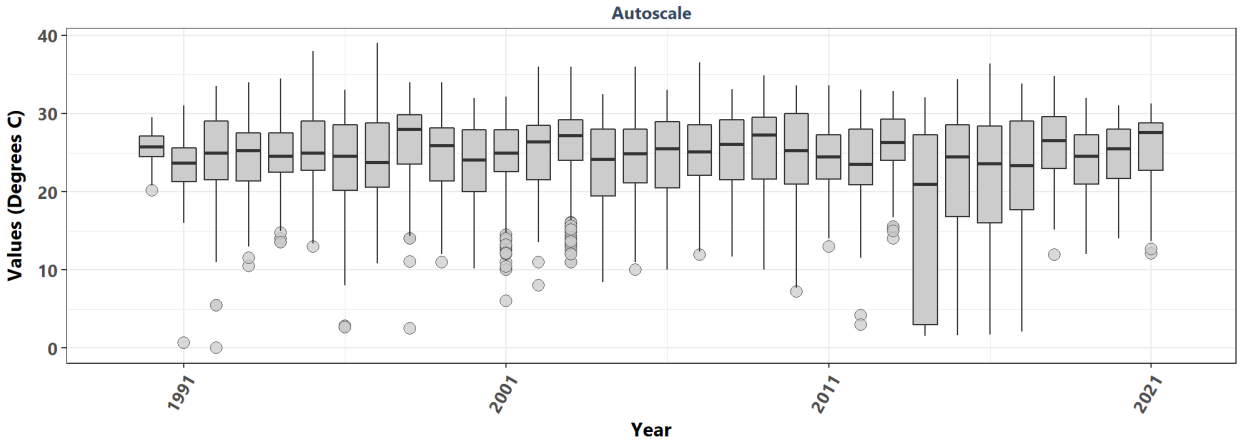


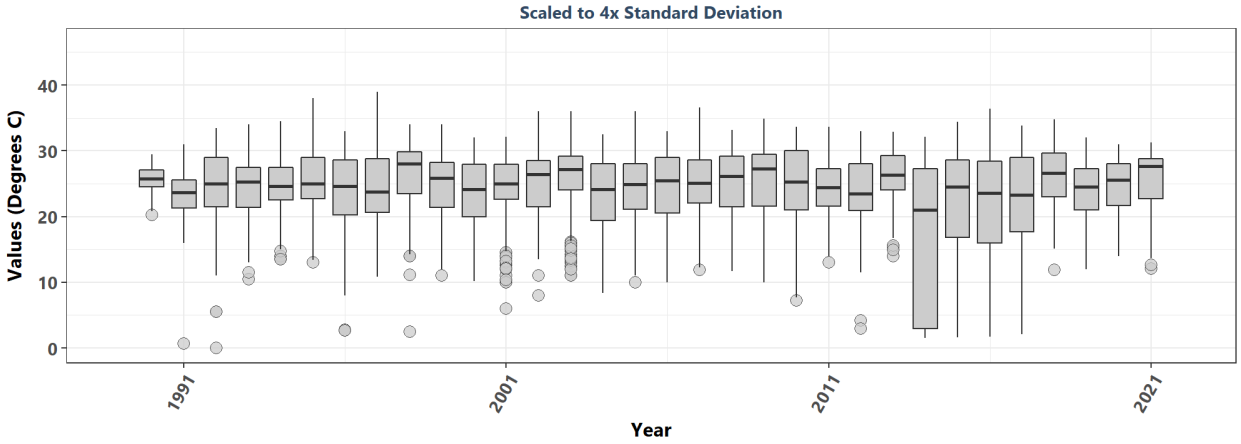

Apalachicola National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30
1972 1982 1992 2002 2012 2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40
1972 1982 1992 2002 2012 2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30 40
2012 2014 2016 2018 2020 2022
Year

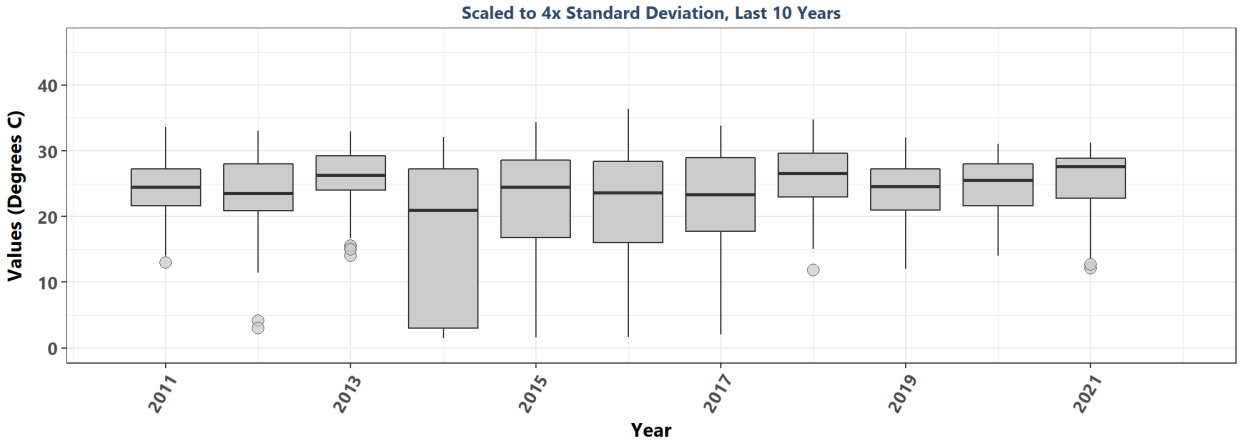

Apalachicola National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30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

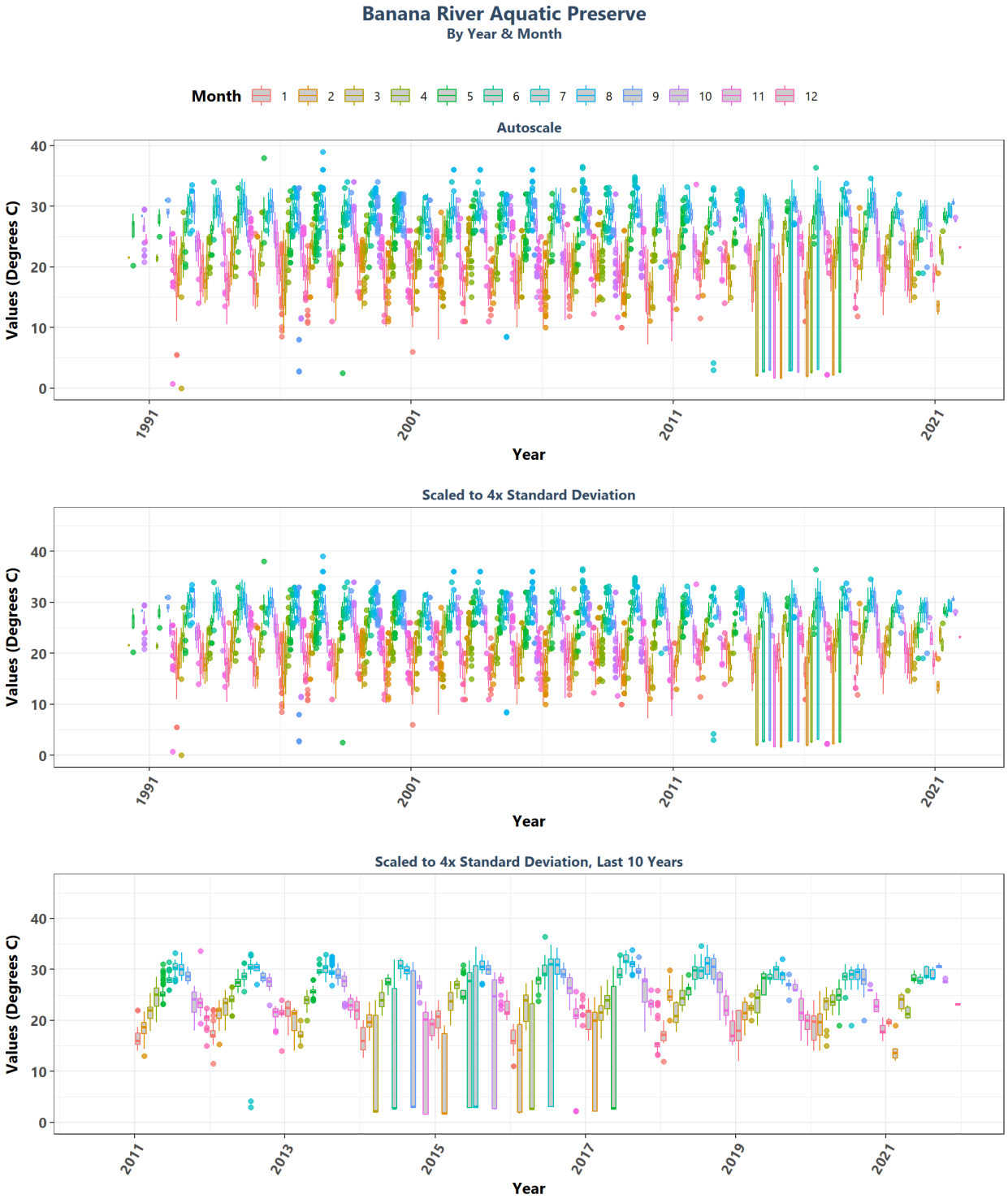
Banana River Aquatic Preserve

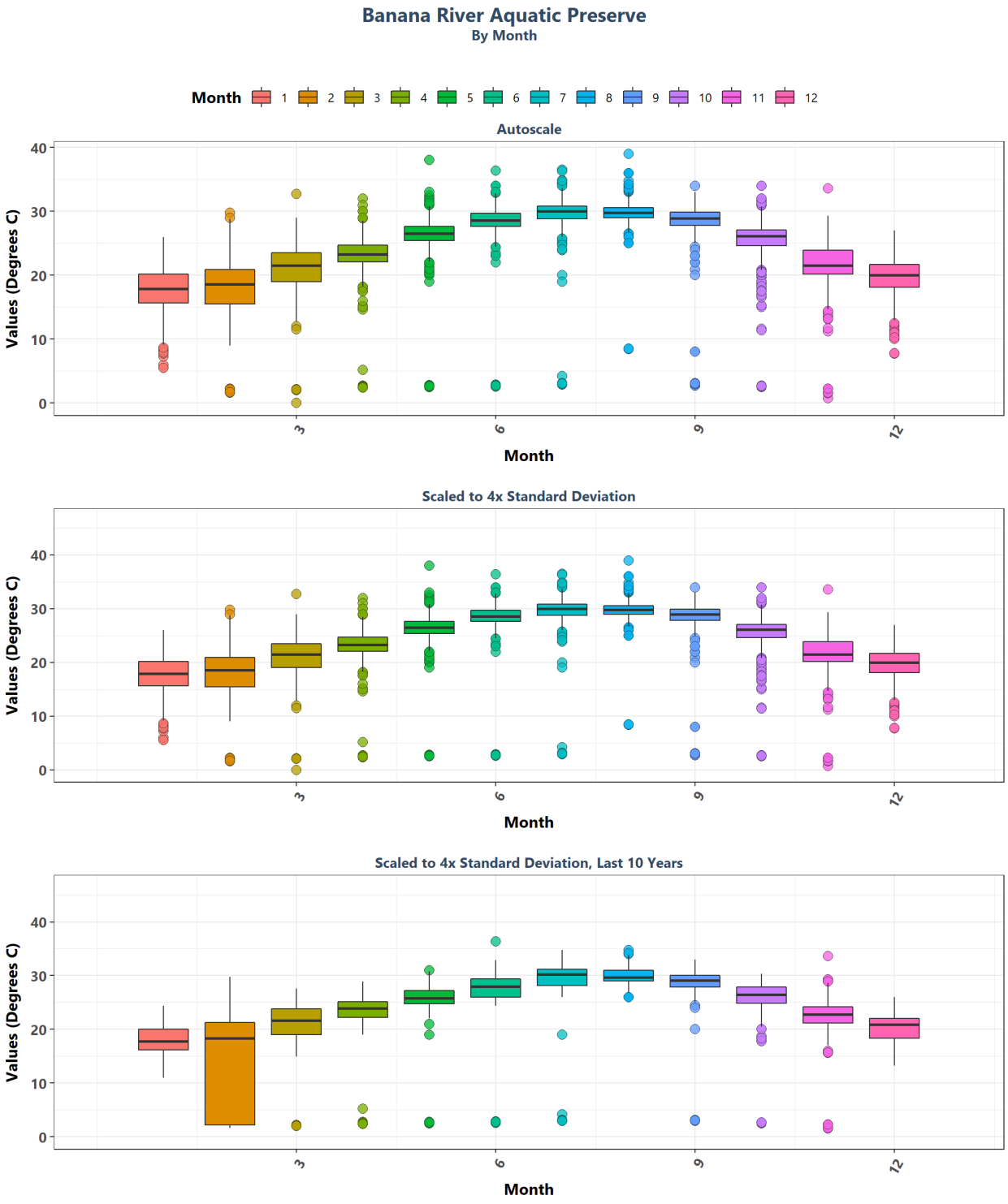
By Year

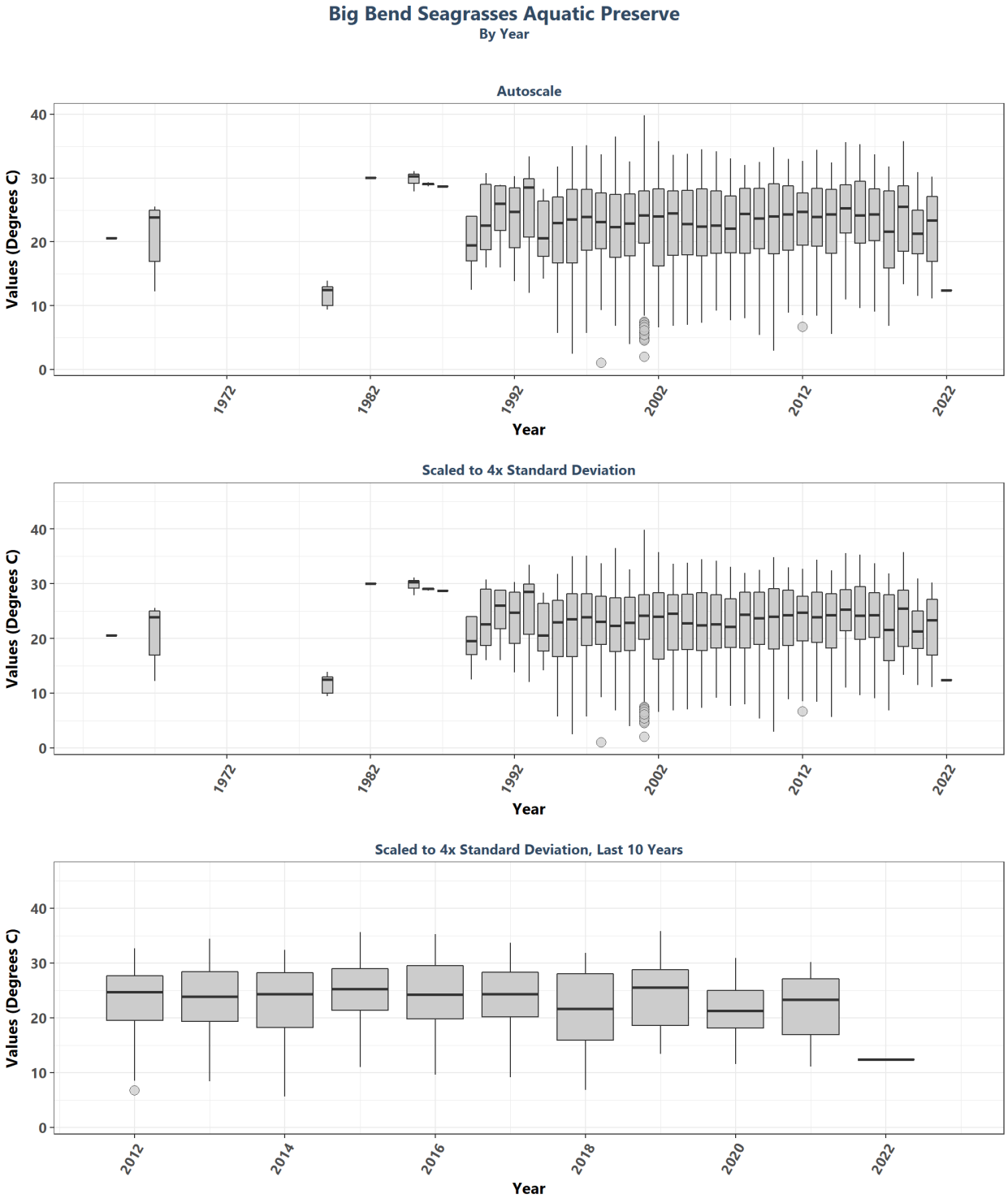


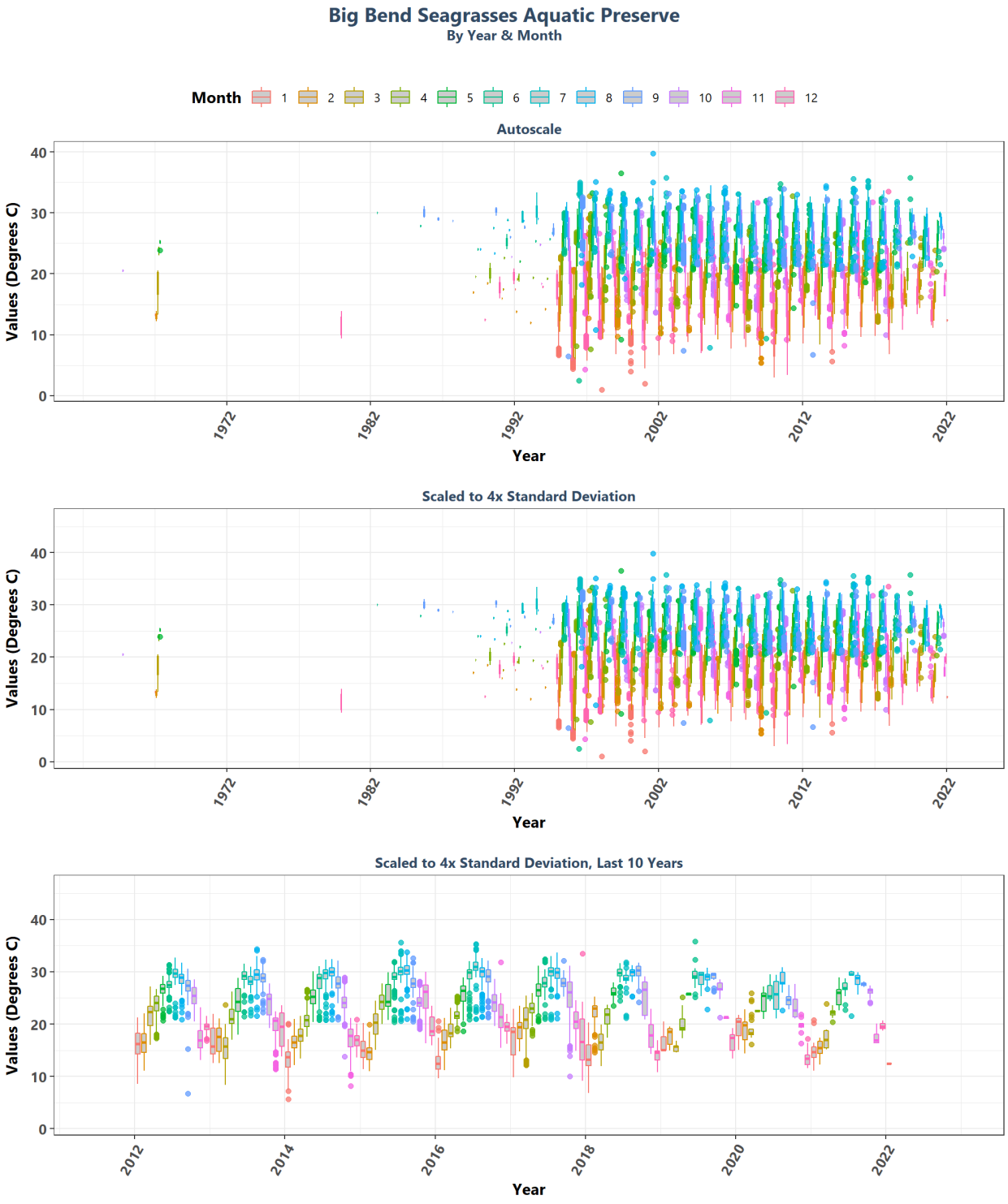


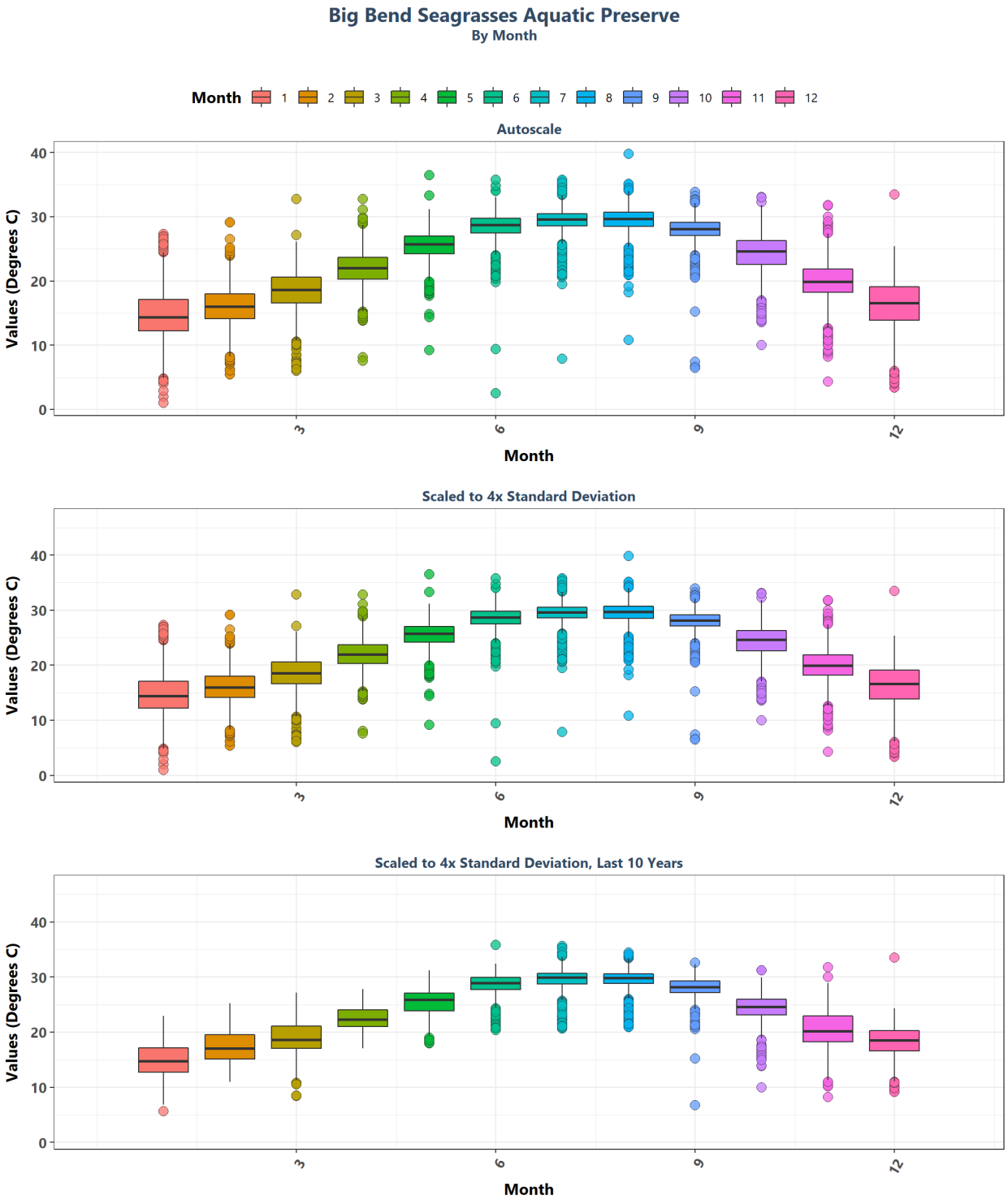



Banana River Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


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

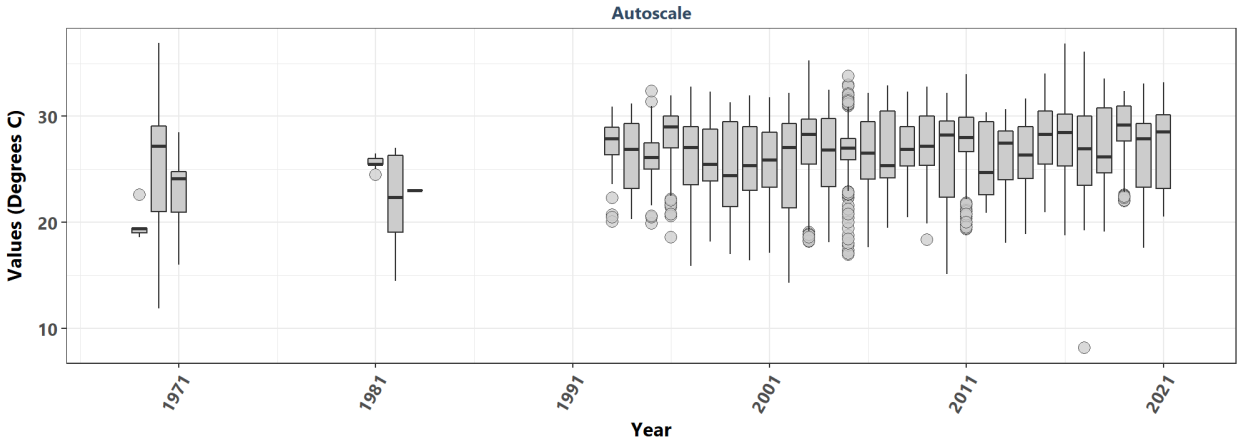

Big Bend Seagrasses Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
0
10
20
30
40
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0
10
20
30
40
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

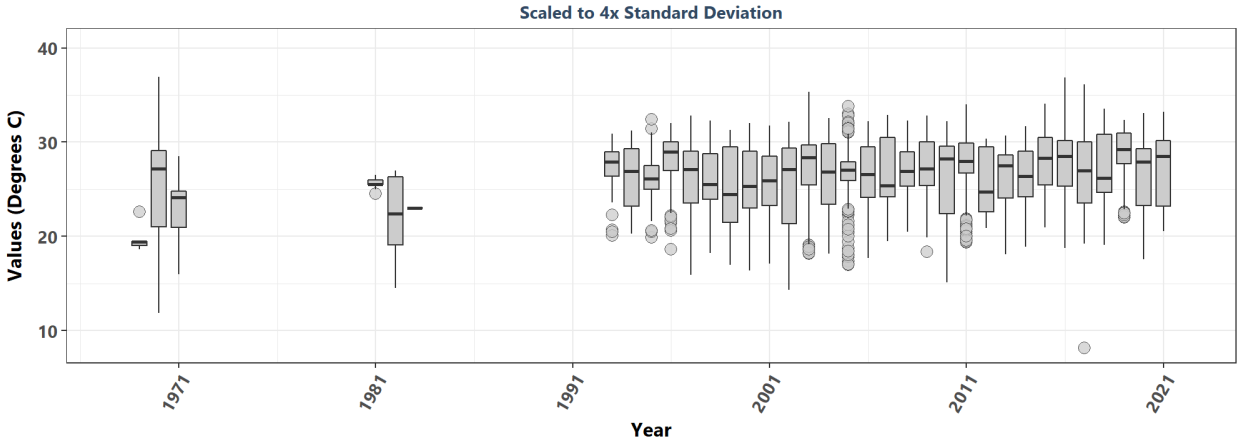

Big Bend Seagrasses Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

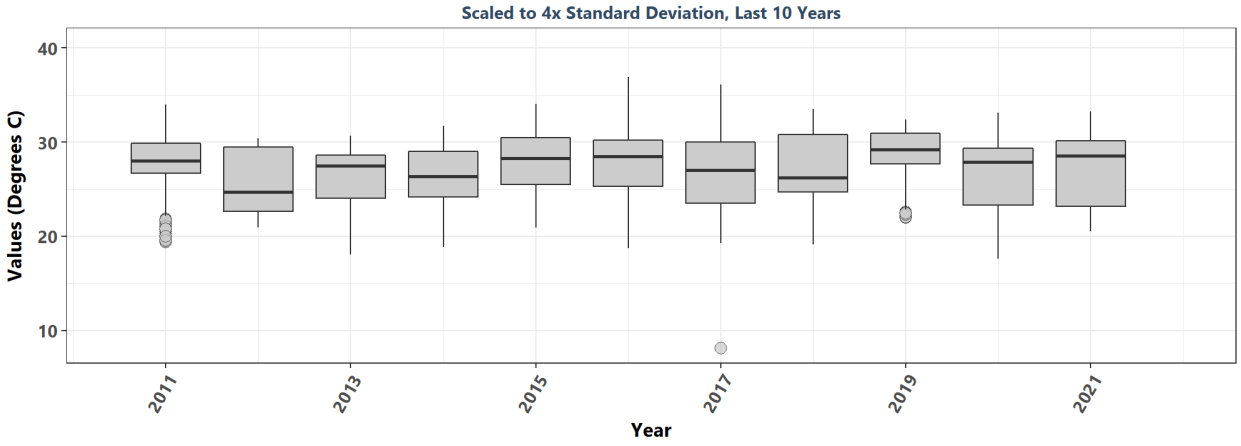

Big Bend Seagrasses Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30 40
3 6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

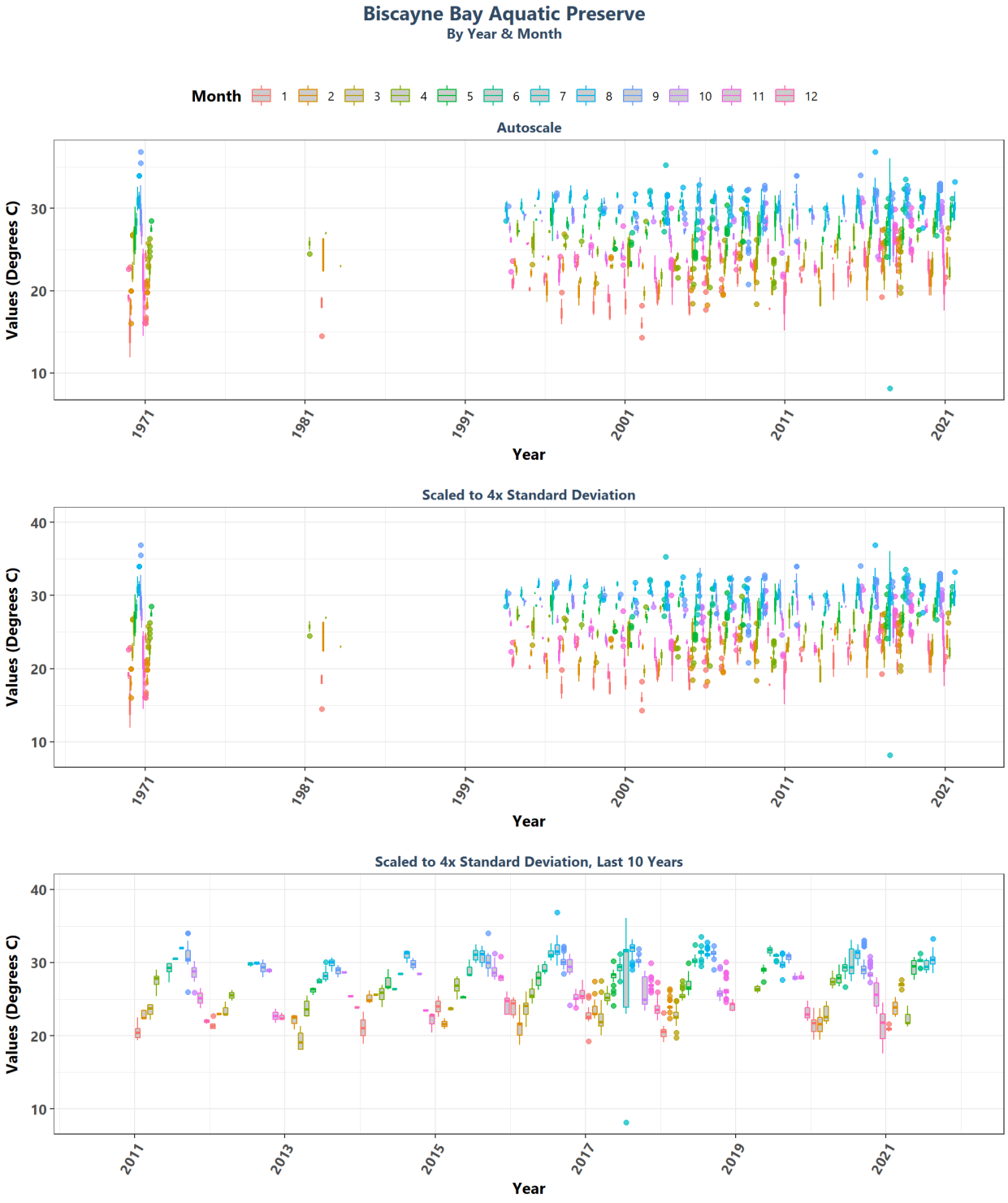
Biscayne Bay Aquatic Preserve

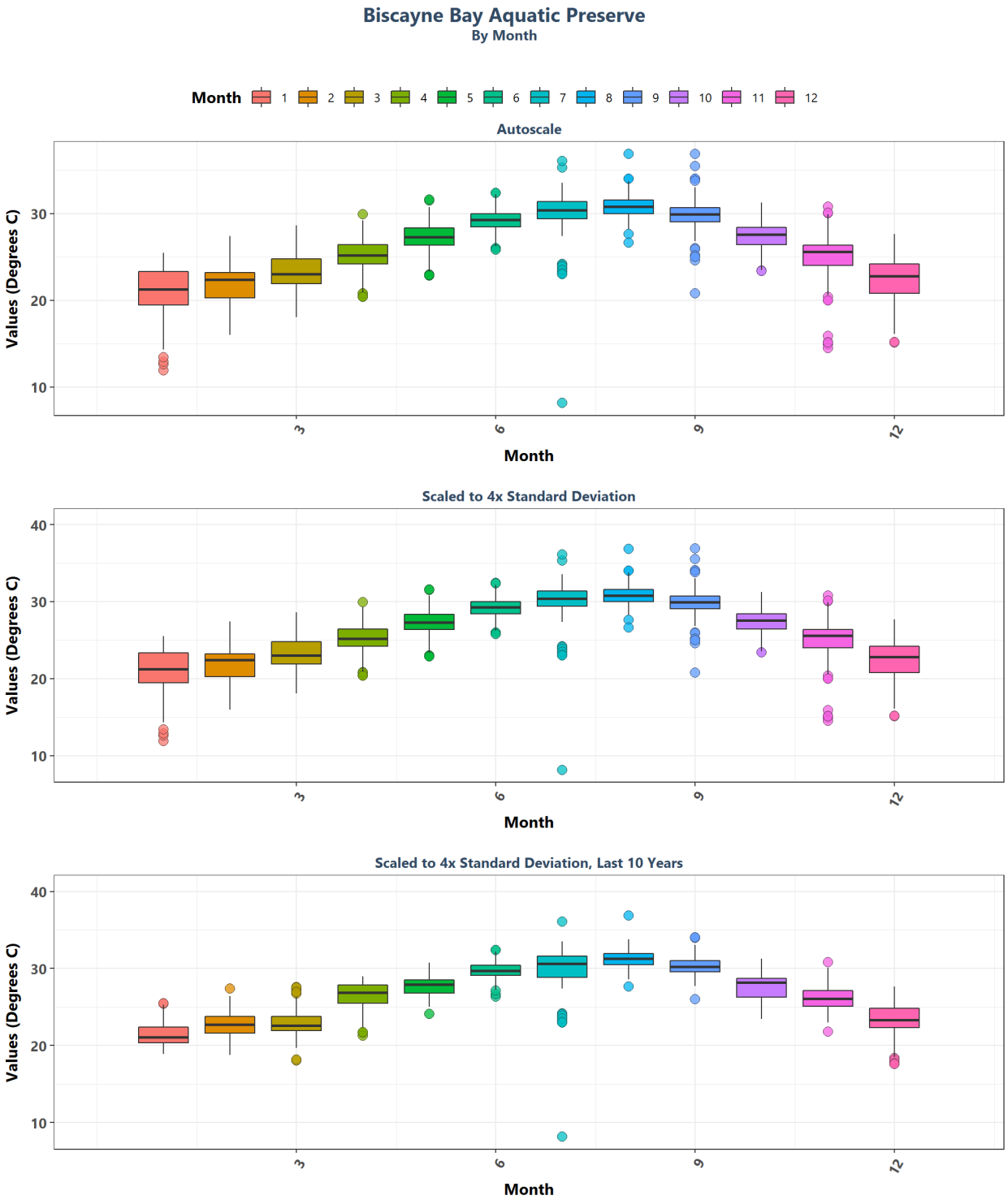
By Year





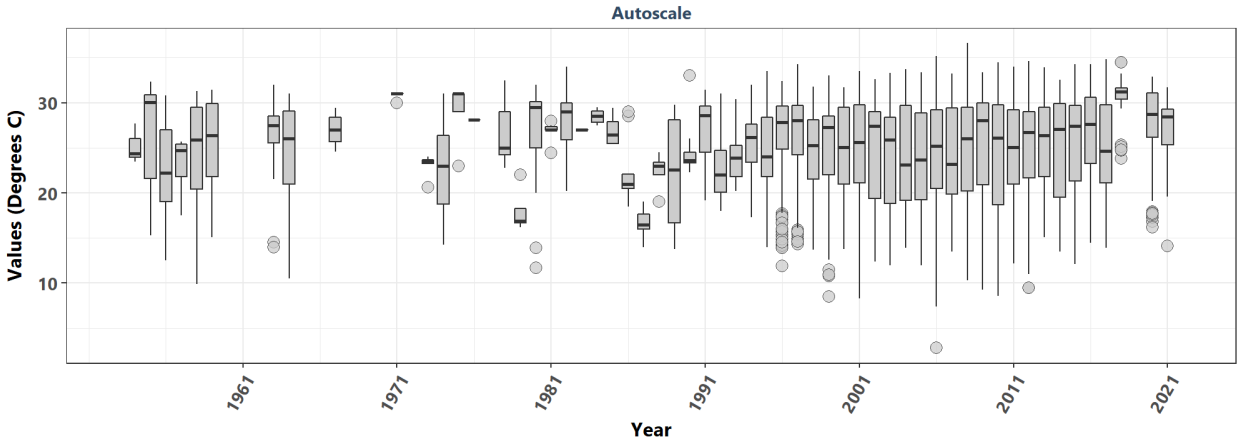


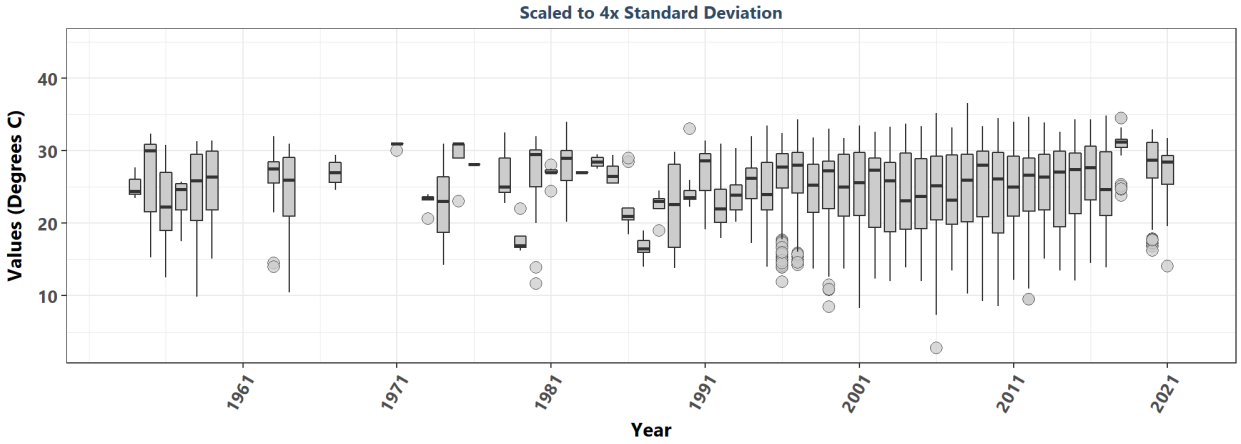

Biscayne Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2011
2013
2015
2017
2019
2021
Year

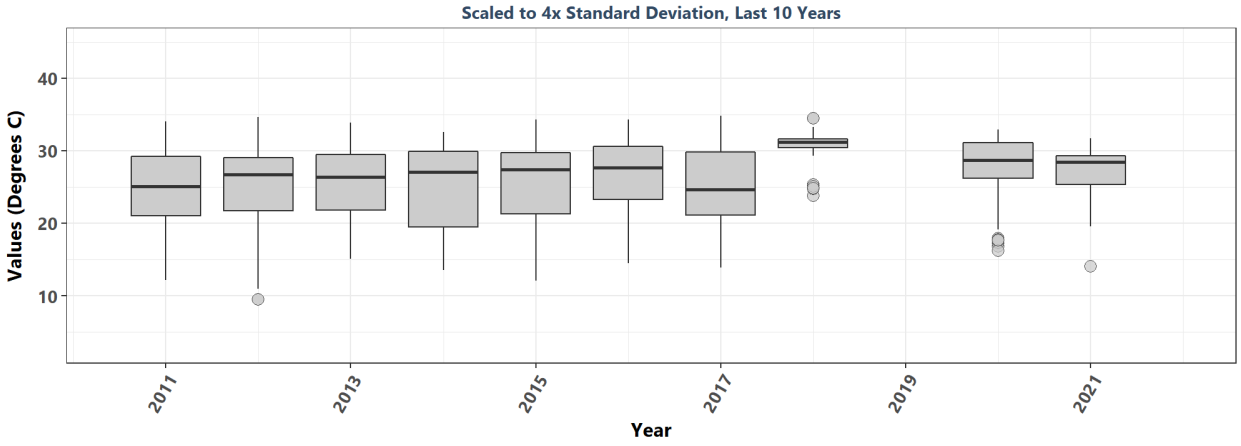

Biscayne Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

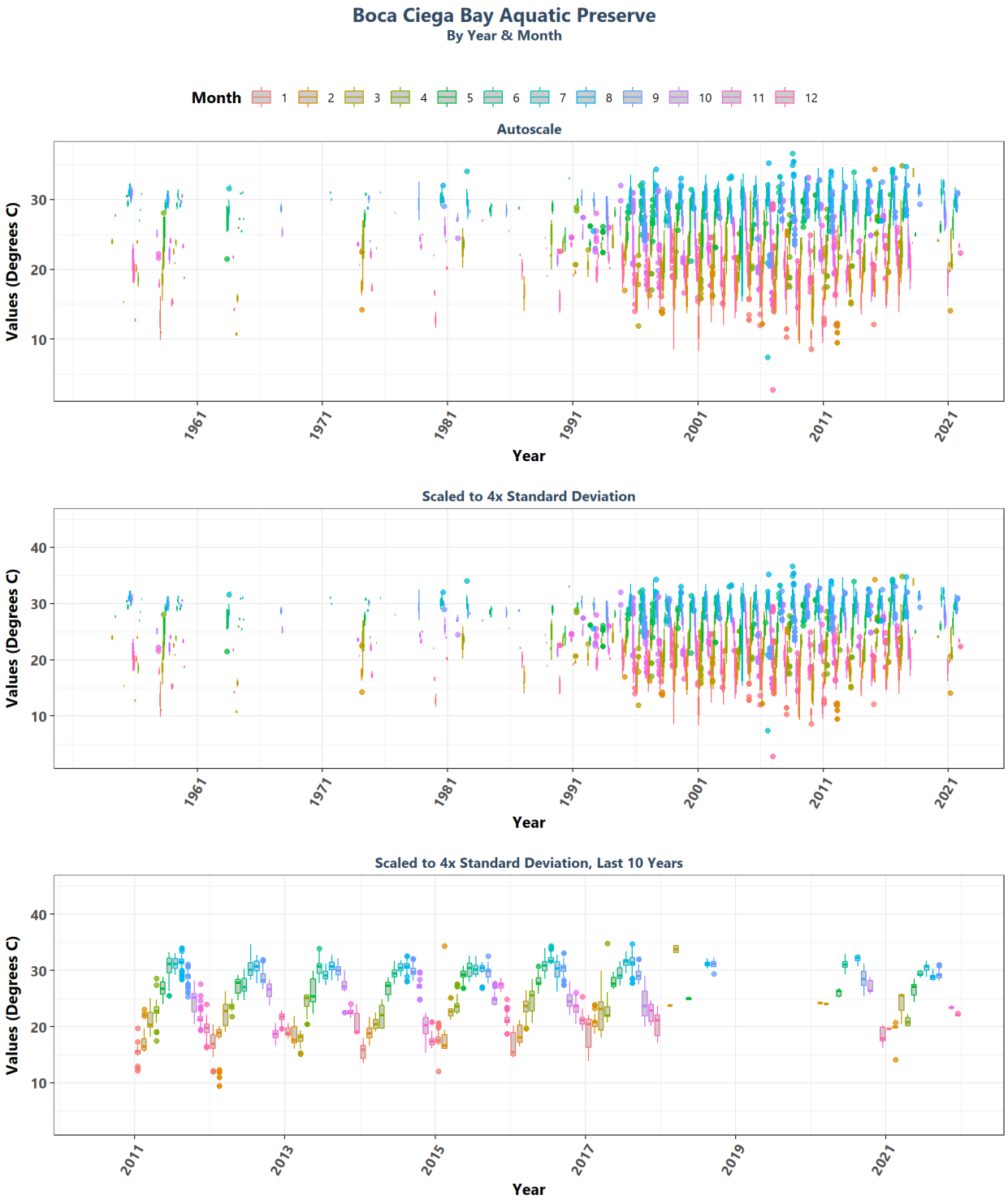
Boca Ciega Bay Aquatic Preserve

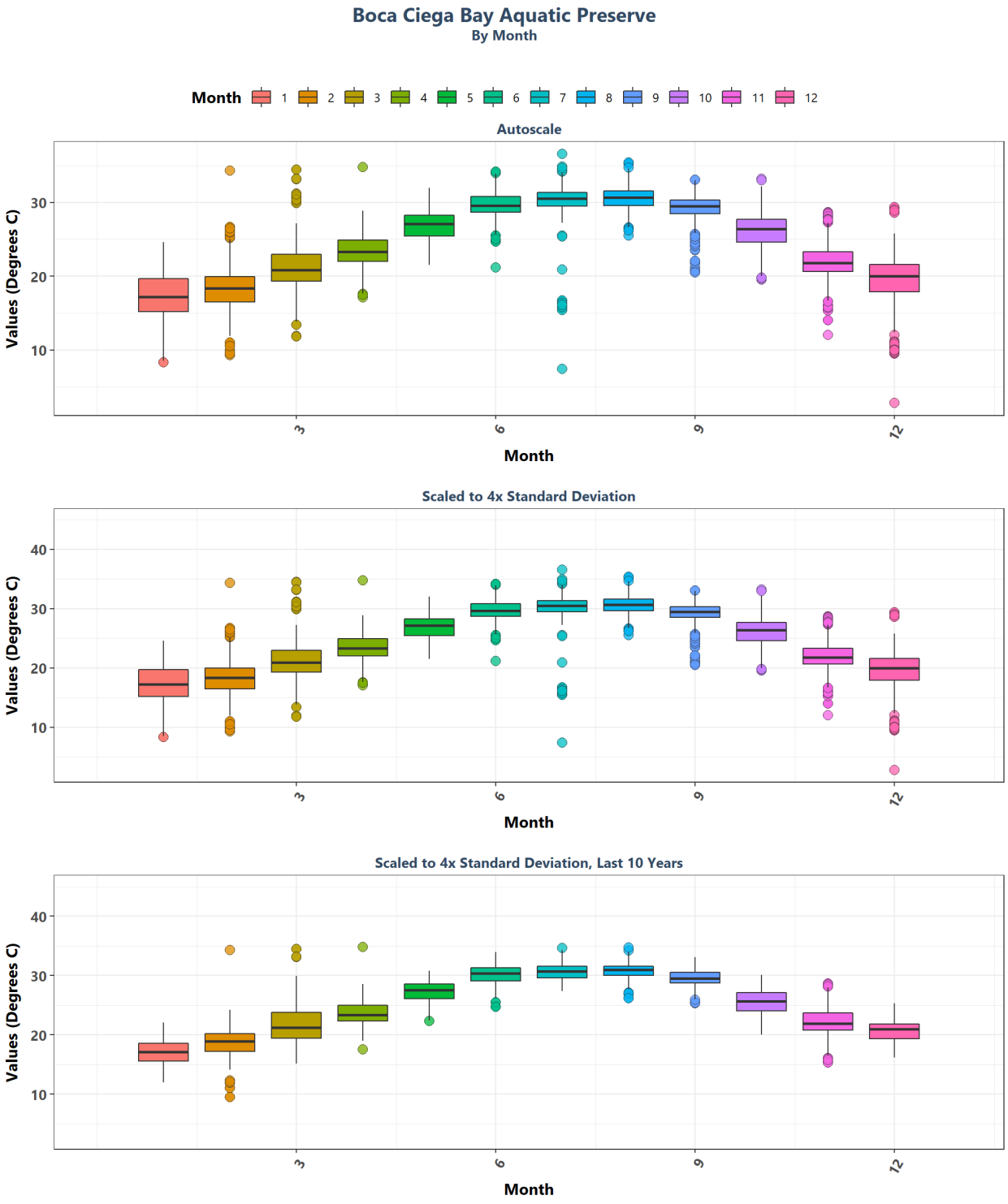
By Year





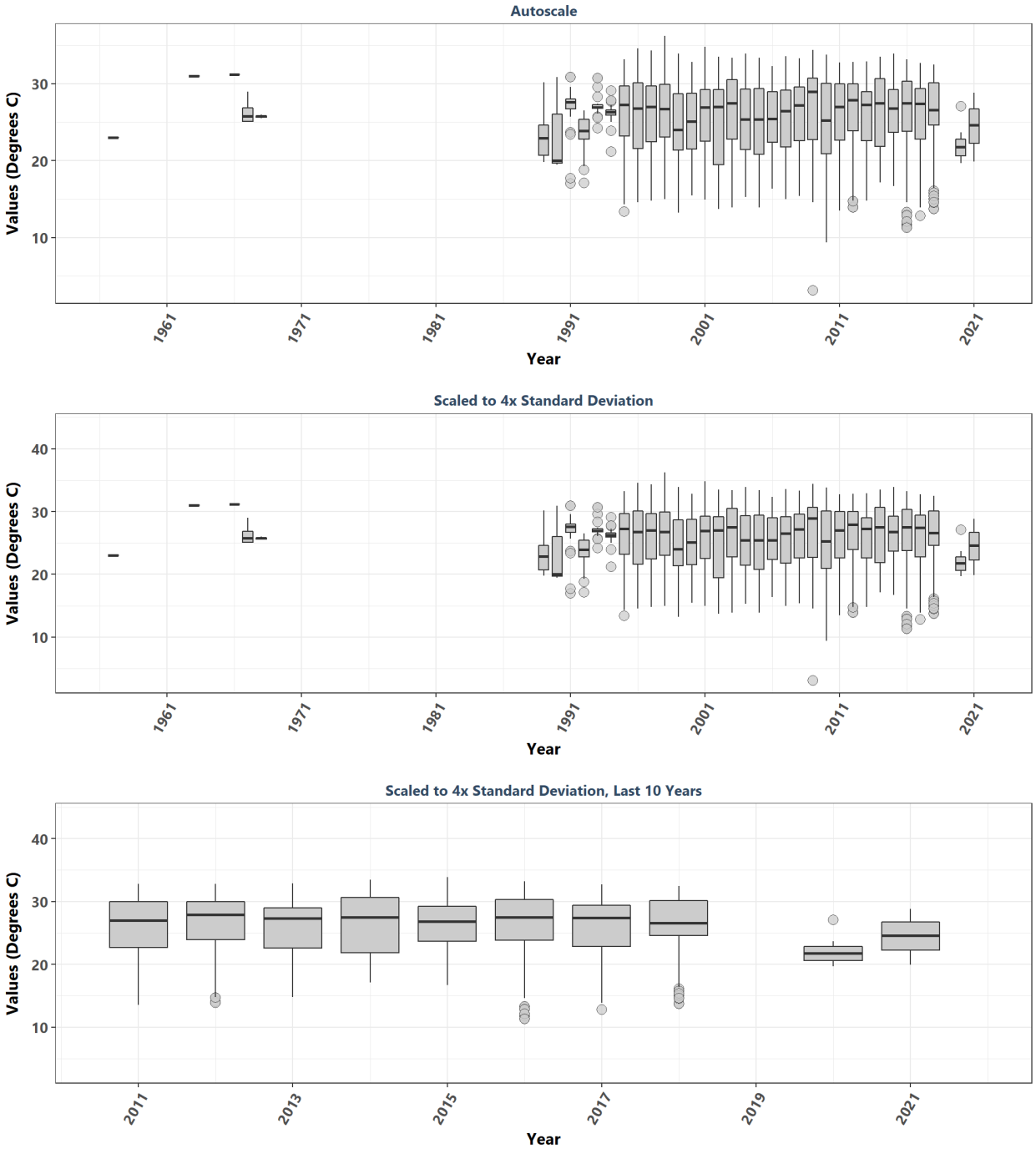


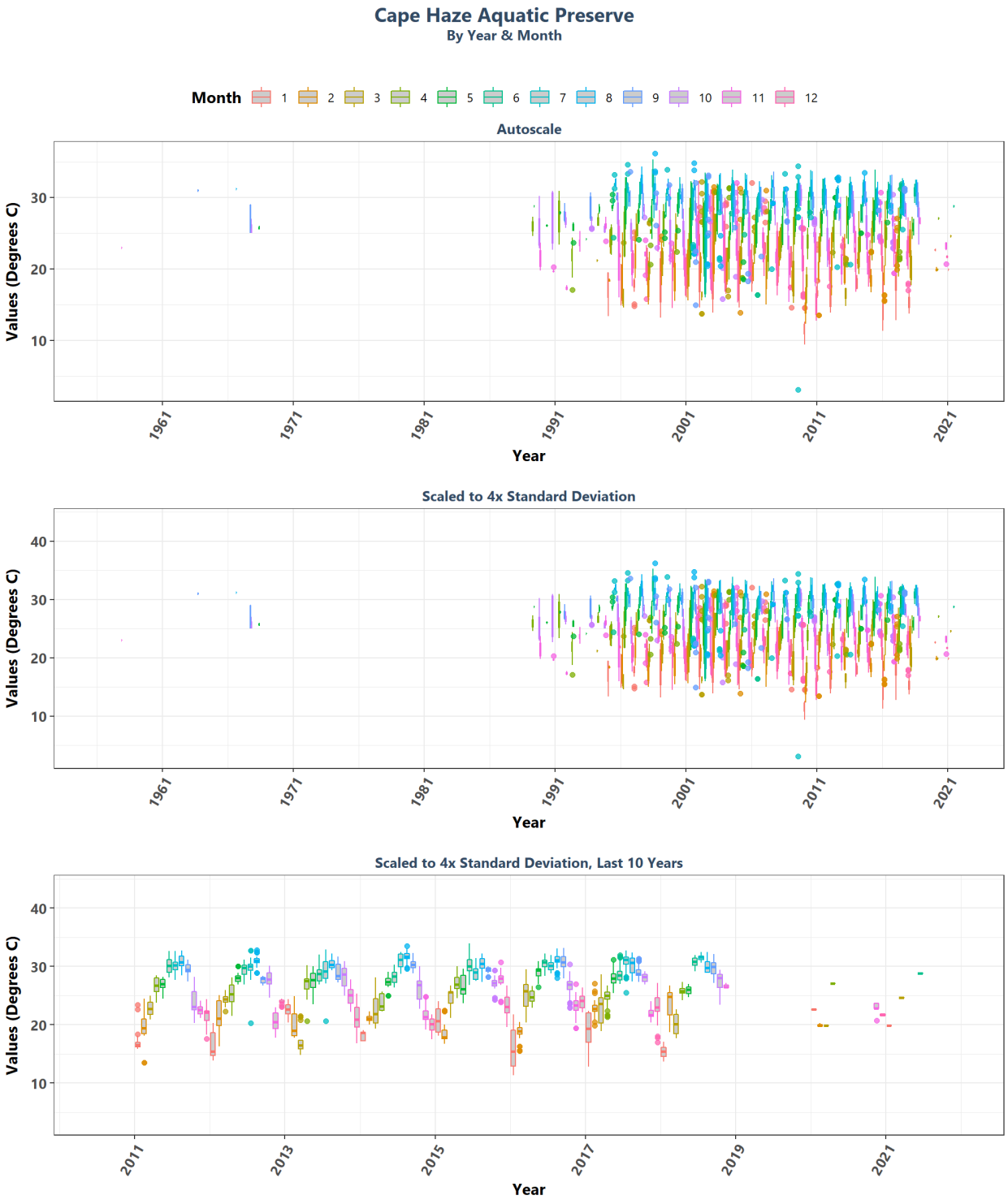

Boca Ciega Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

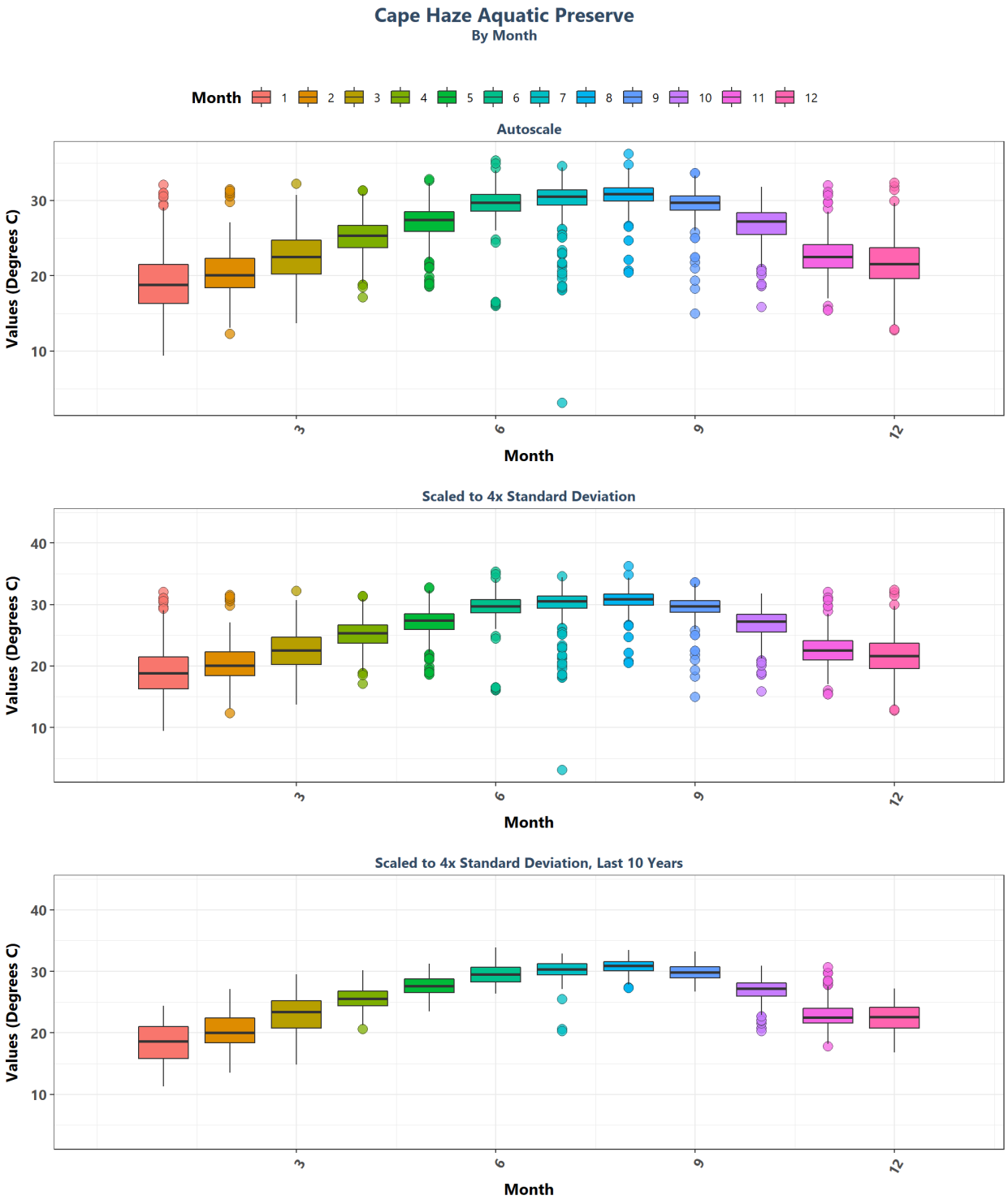

Boca Ciega Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

Cape Haze Aquatic Preserve

By Year

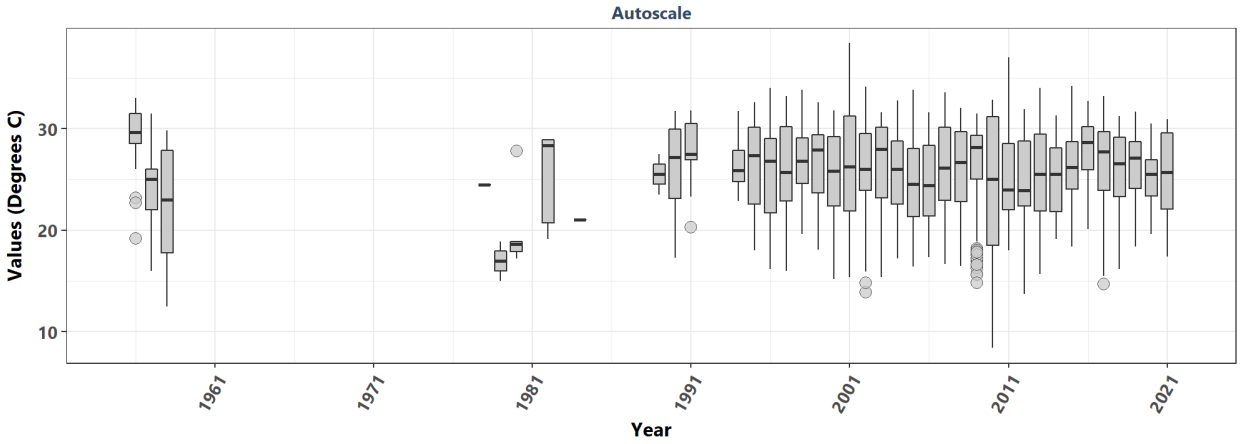


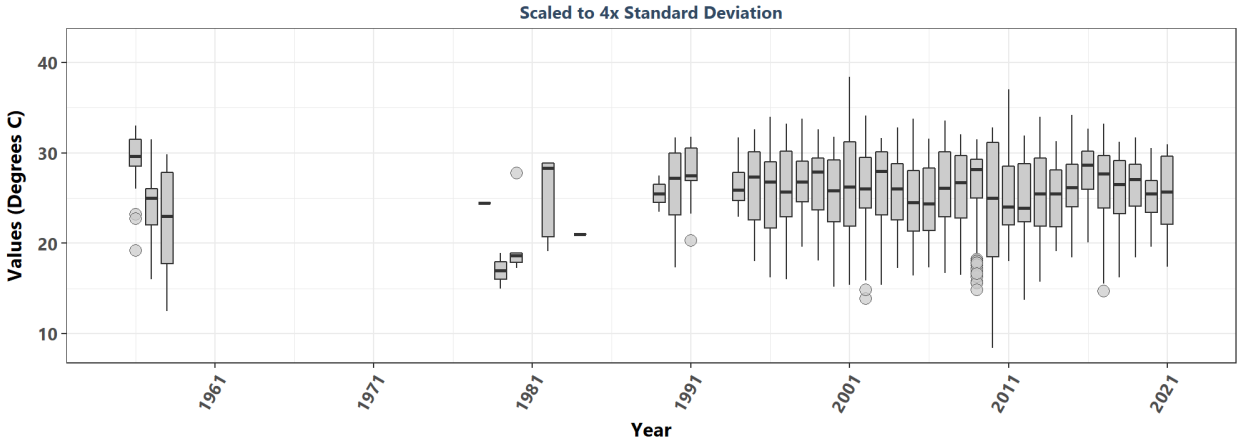

Cape Haze Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

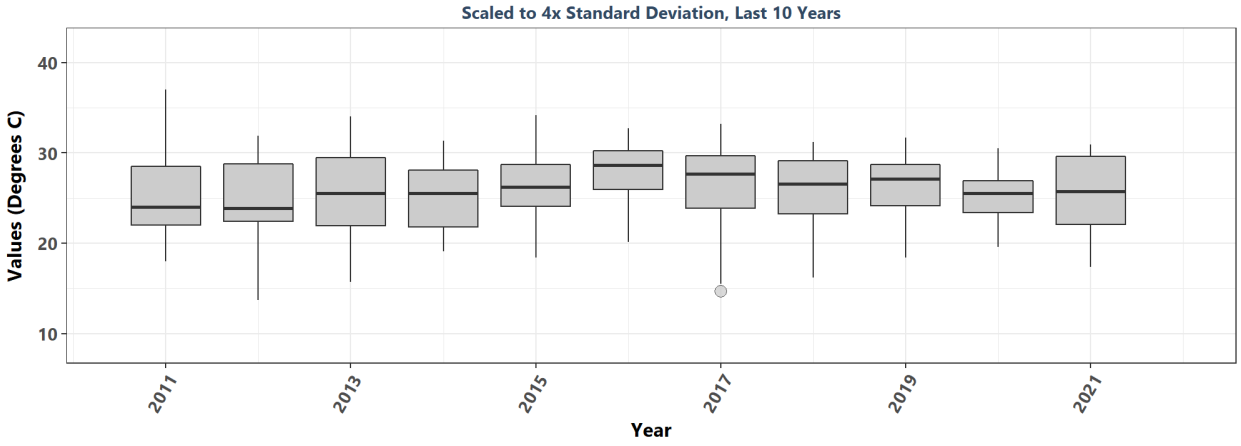

Cape Haze Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

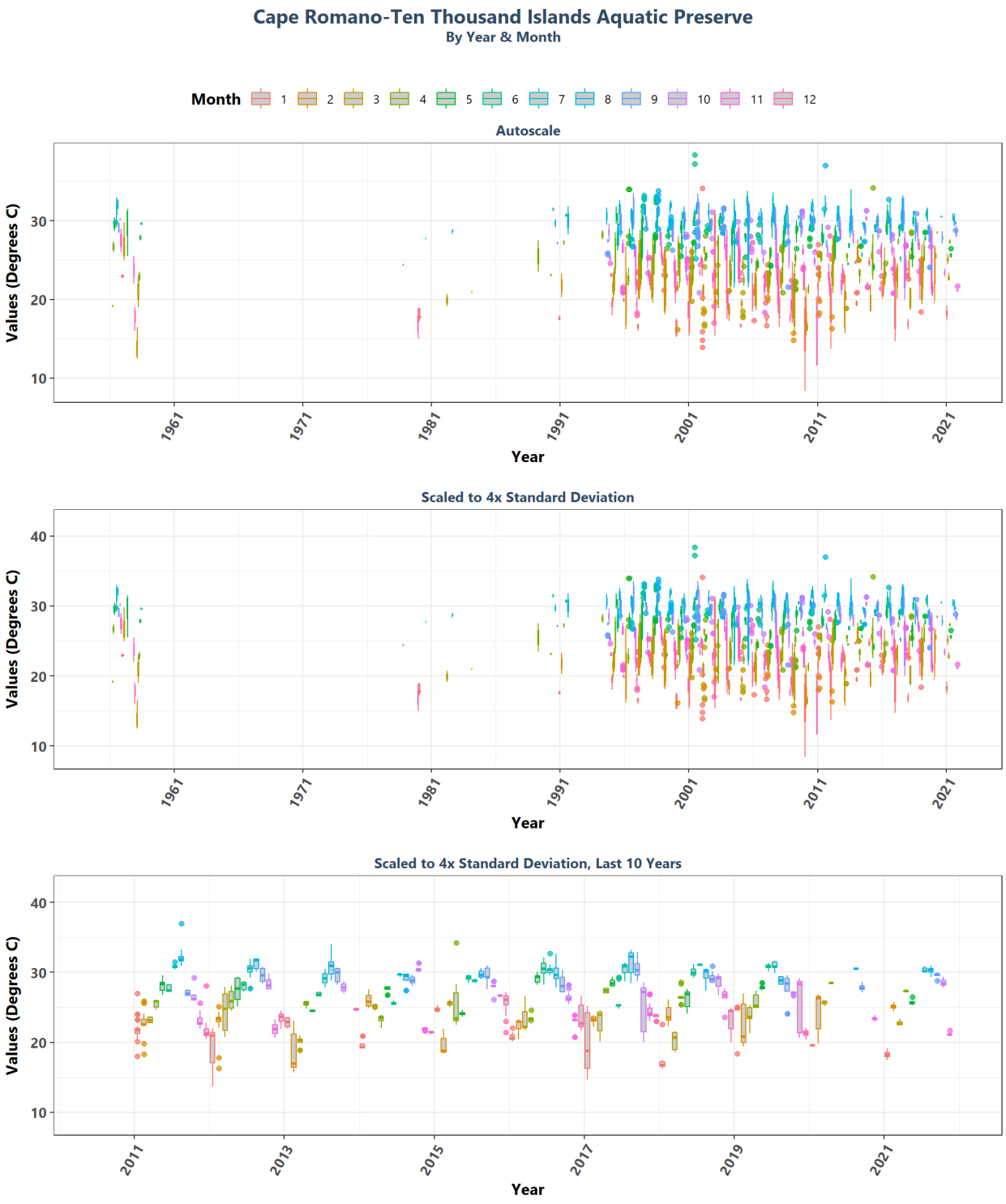
Cape Romano-Ten Thousand Islands Aquatic Preserve

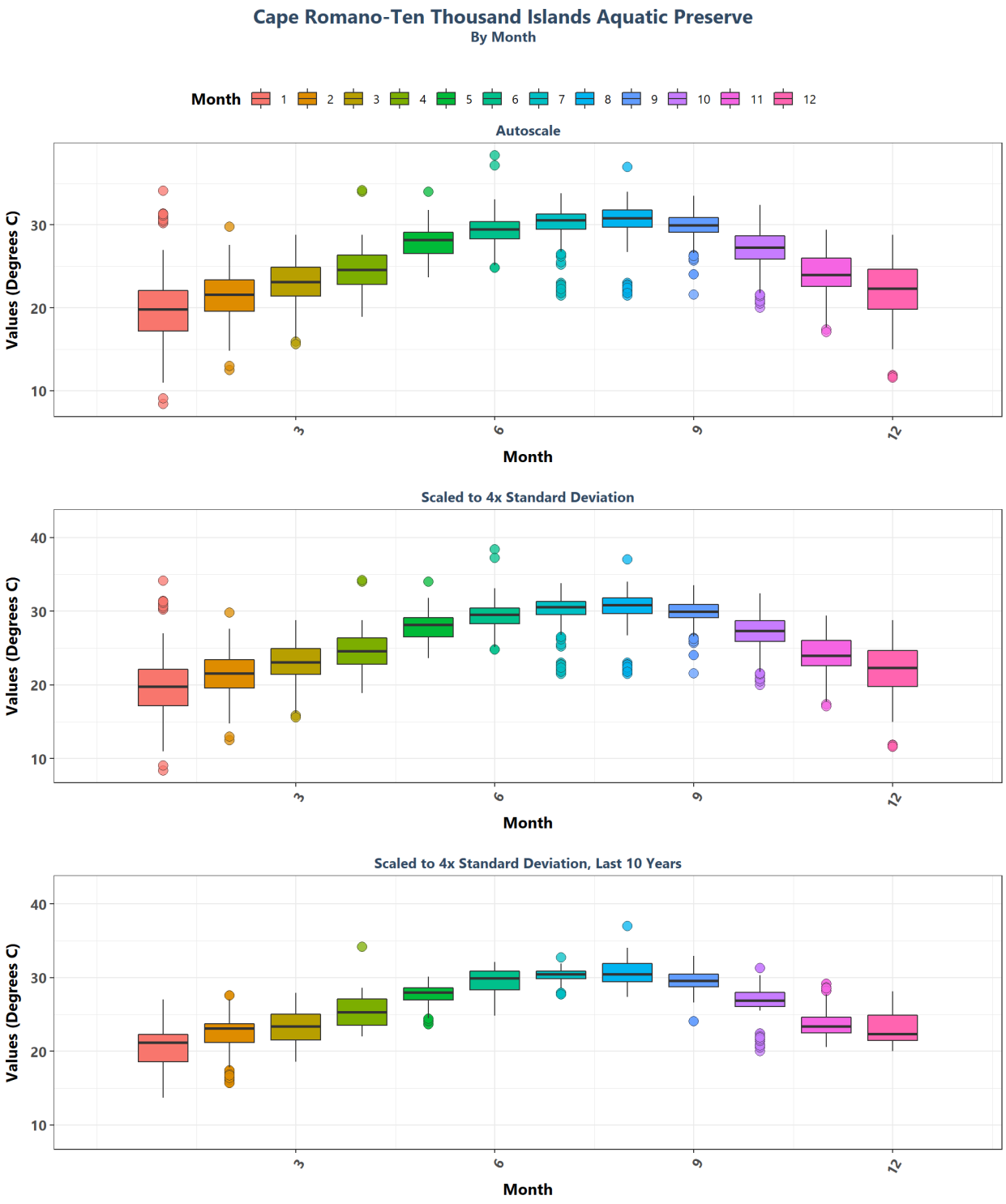
By Year





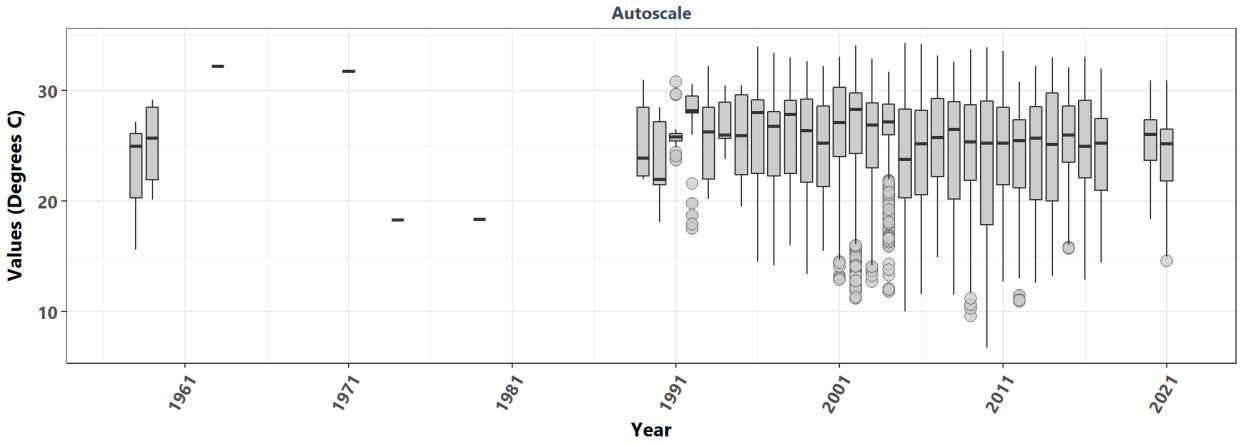


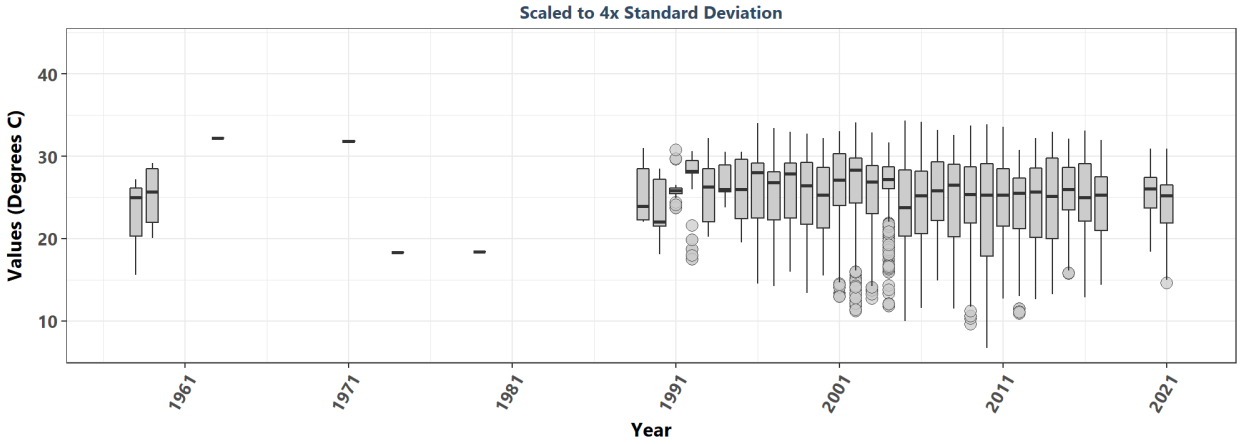

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

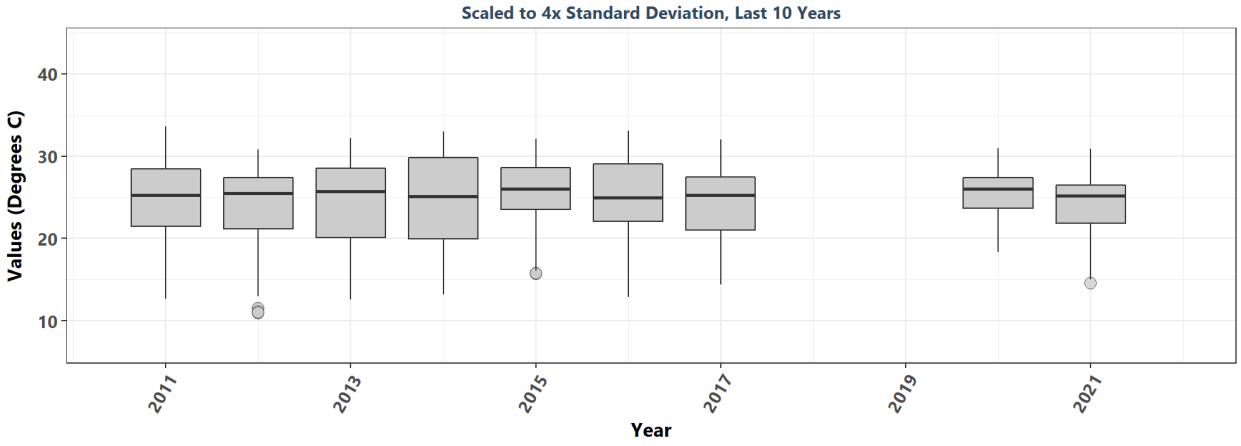

Cape Romano-Ten Thousand Islands Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40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

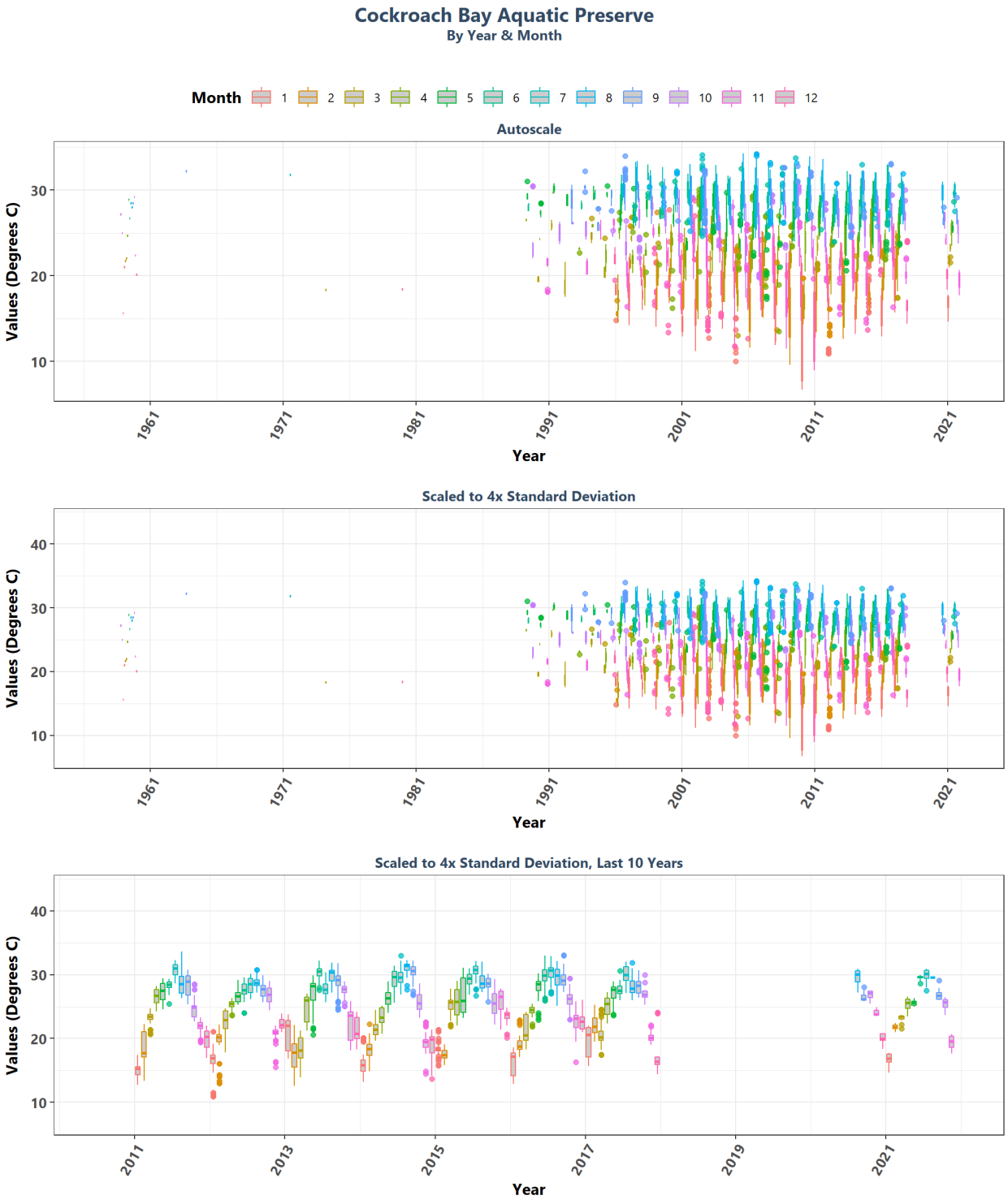
Cockroach Bay Aquatic Preserve

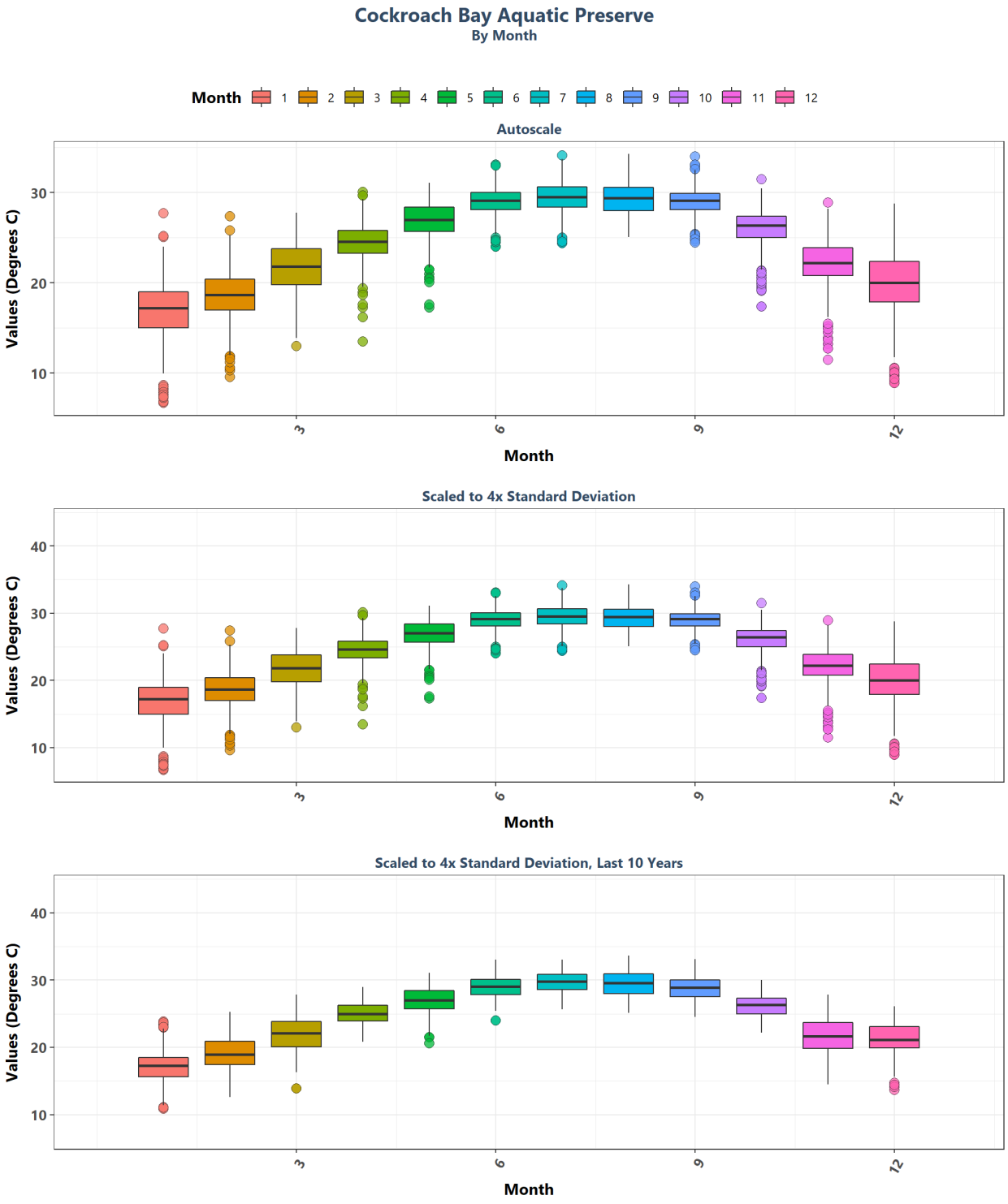
By Year

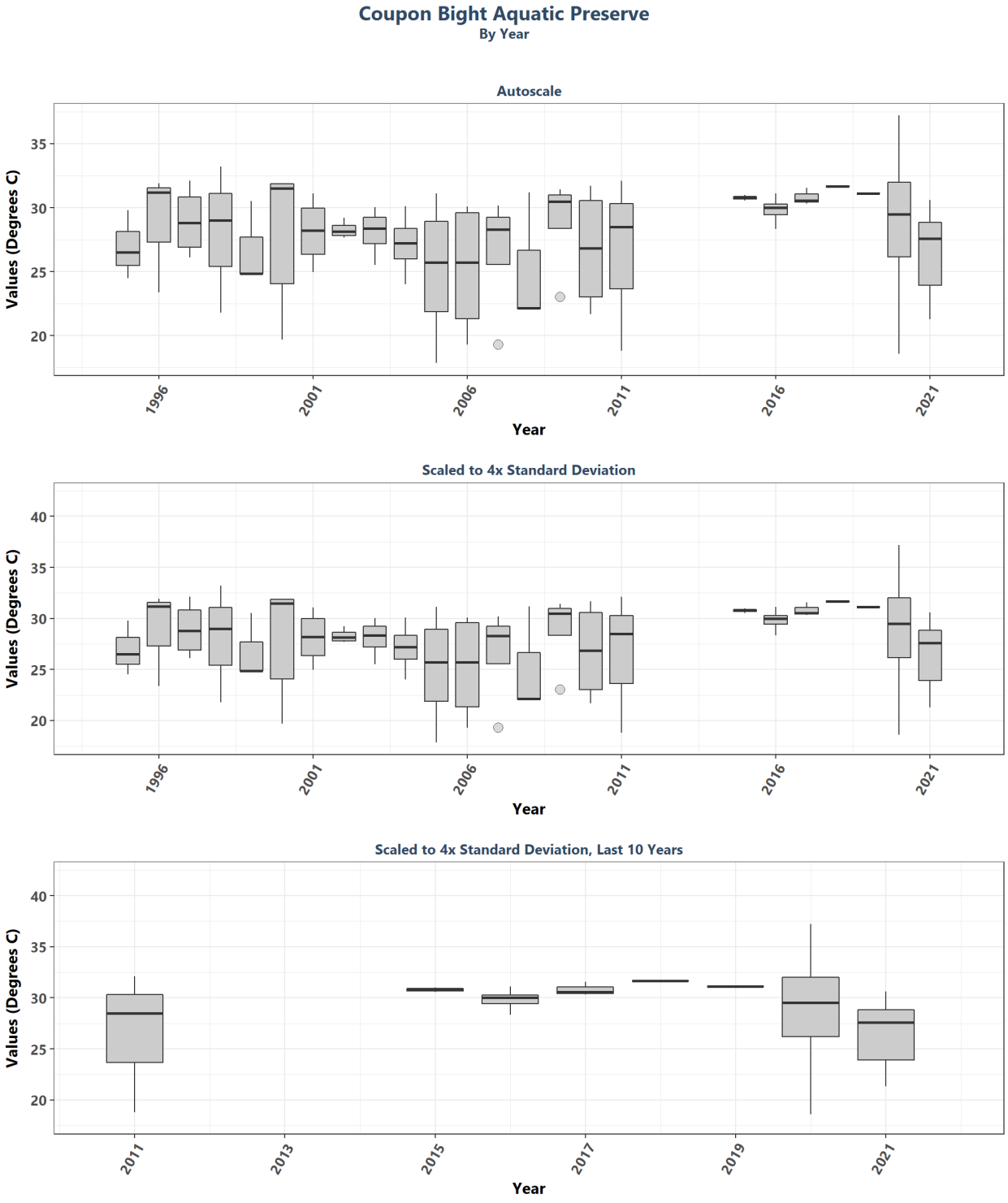


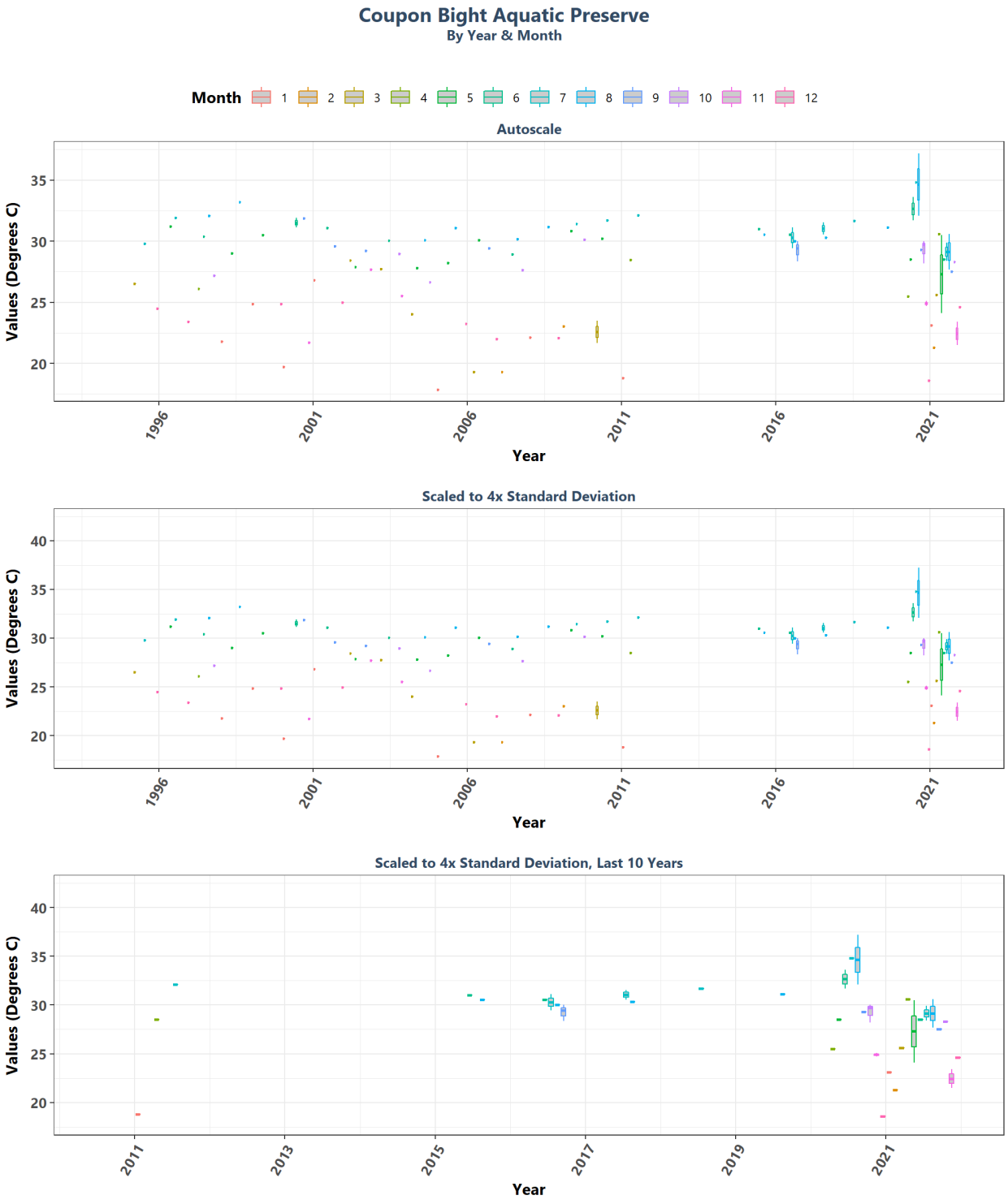


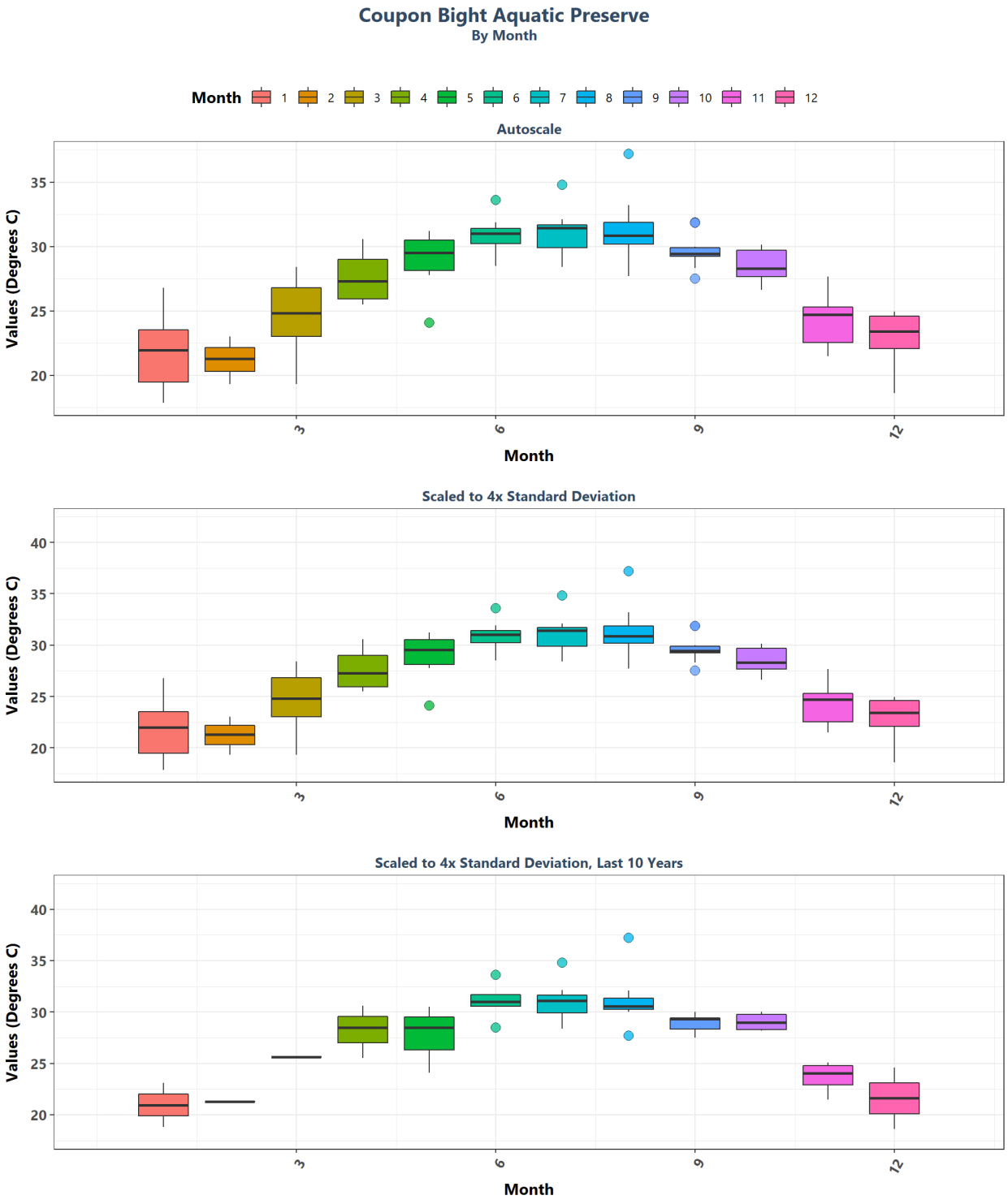


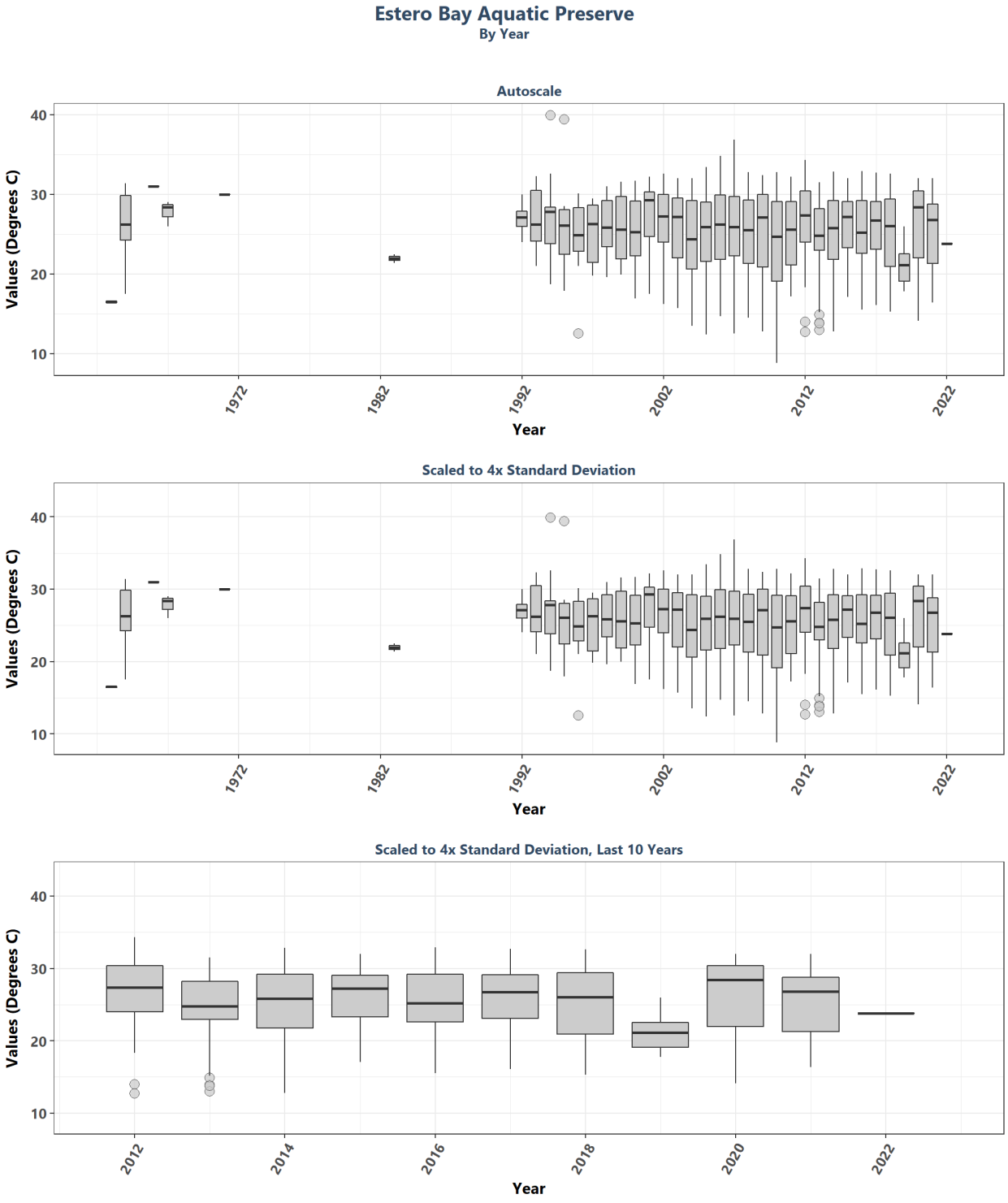

Cockroach Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

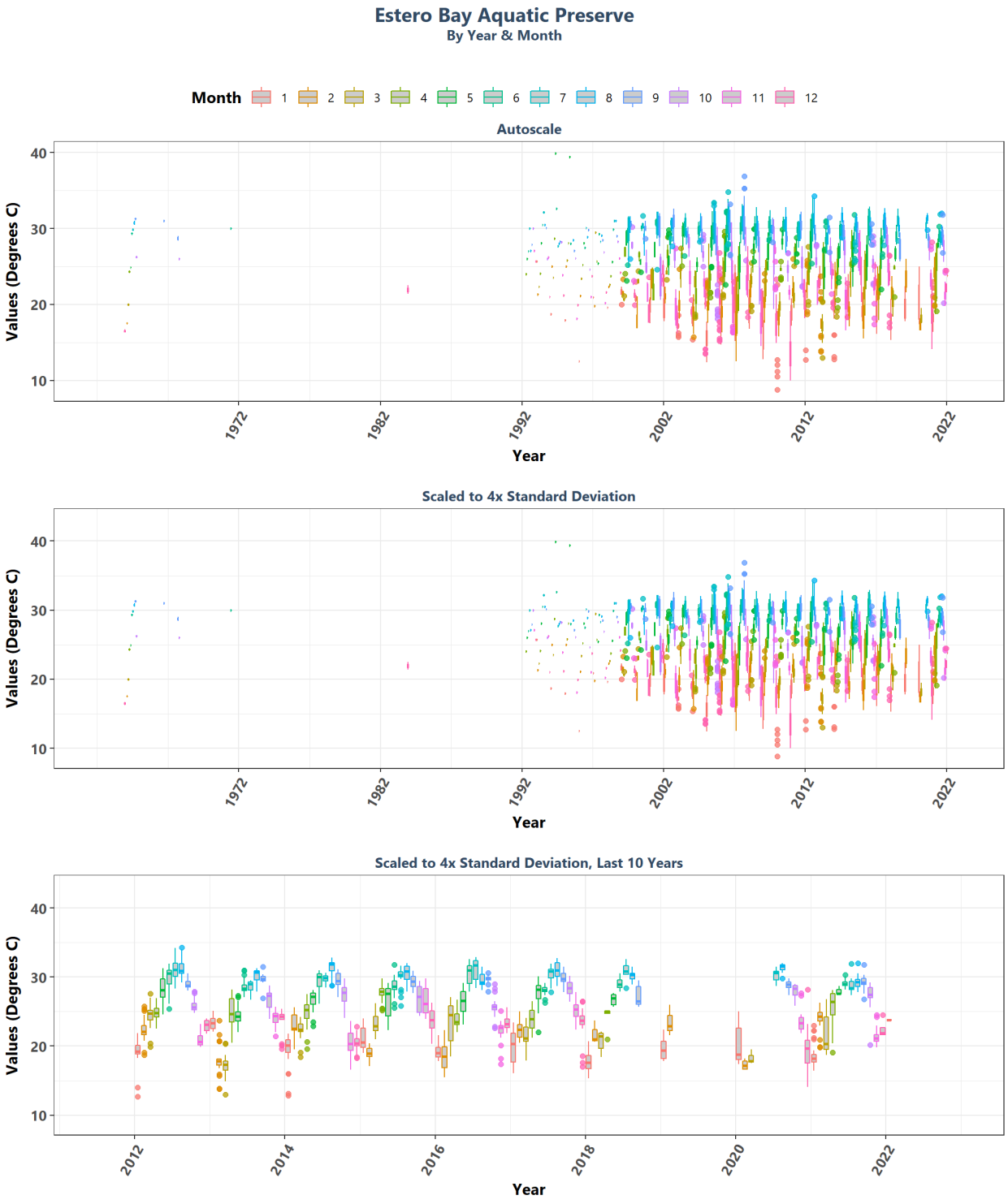

Cockroach Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

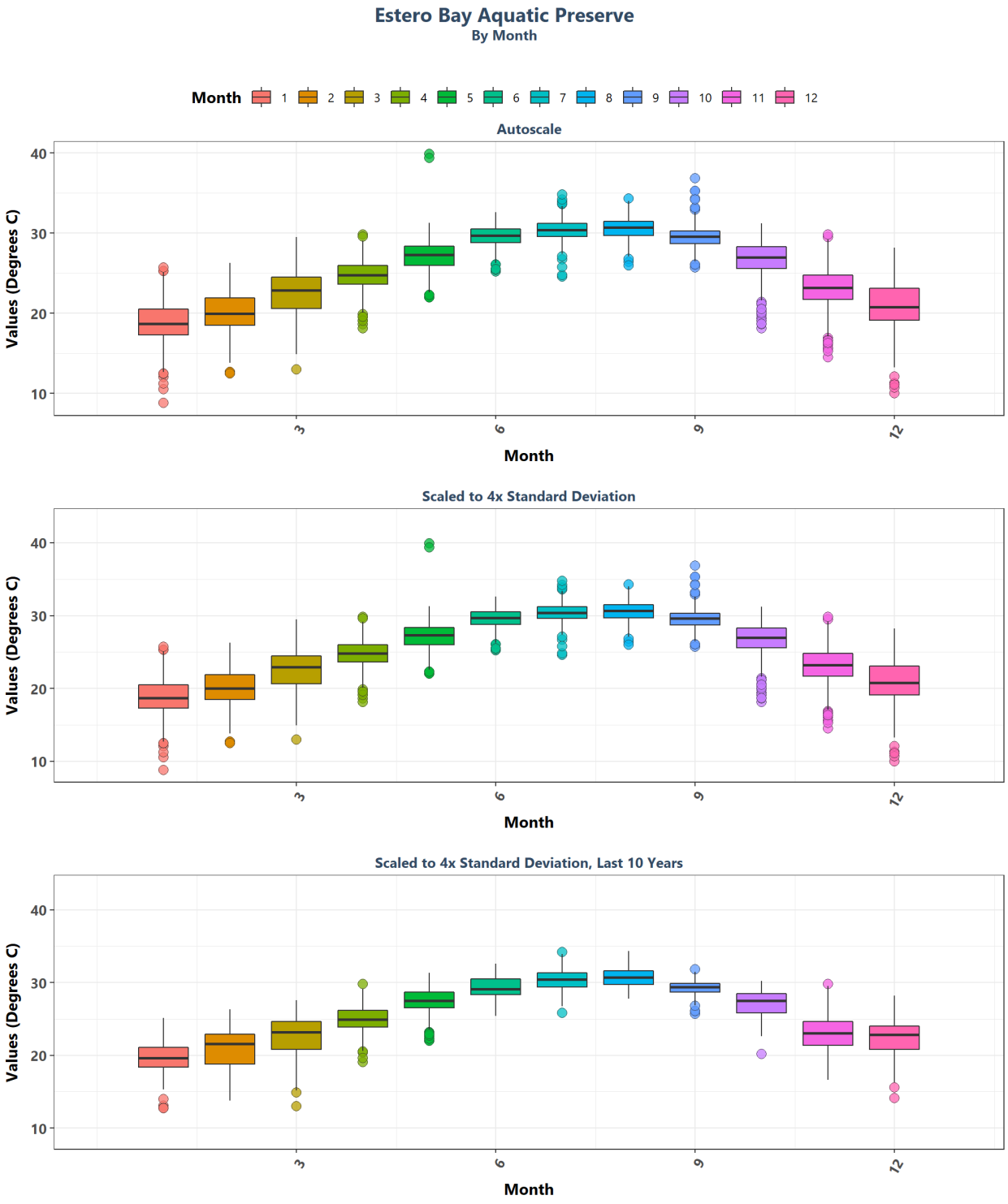

Coupon Bight Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
35
30
25
20
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
35
30
25
20
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
35
30
25
20
2011
2013
2015
2017
2019
2021
Year


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


Coupon Bight 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
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
35
30
25
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
35
30
25
20
3
6
9
12
Month

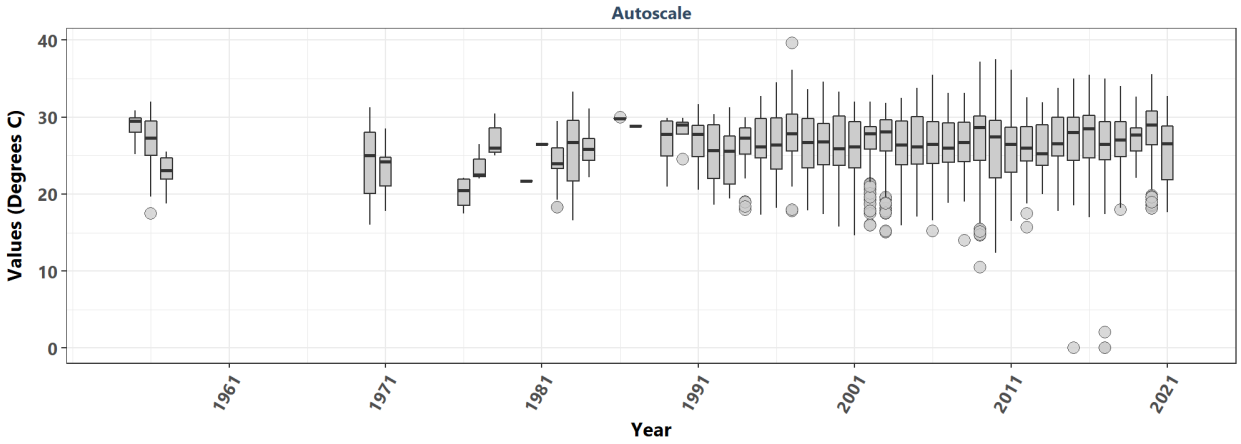

Estero Bay Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
40
30
20
10
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2012
2014
2016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

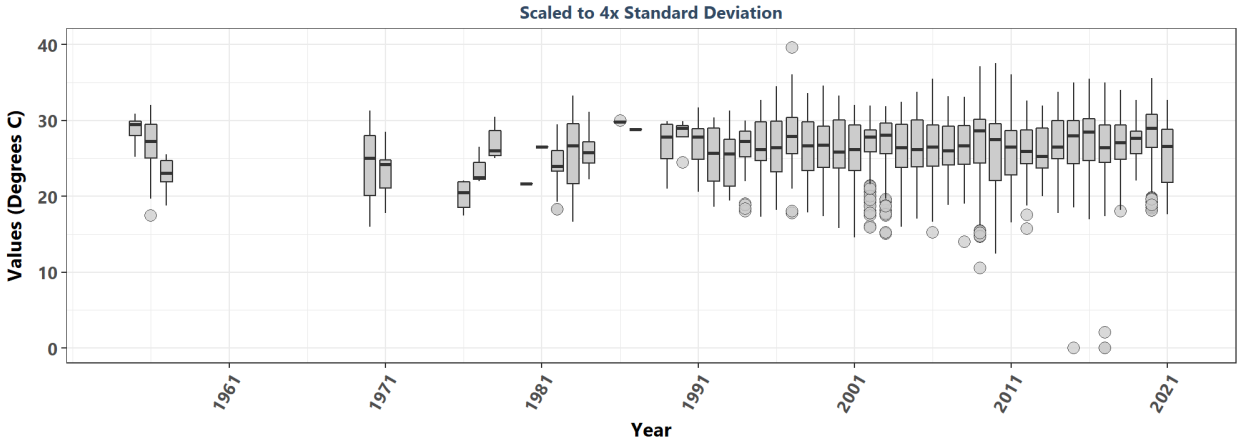

Estero Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
40
30
20
10
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2012
2014
2016
2018
2020
2022
Year

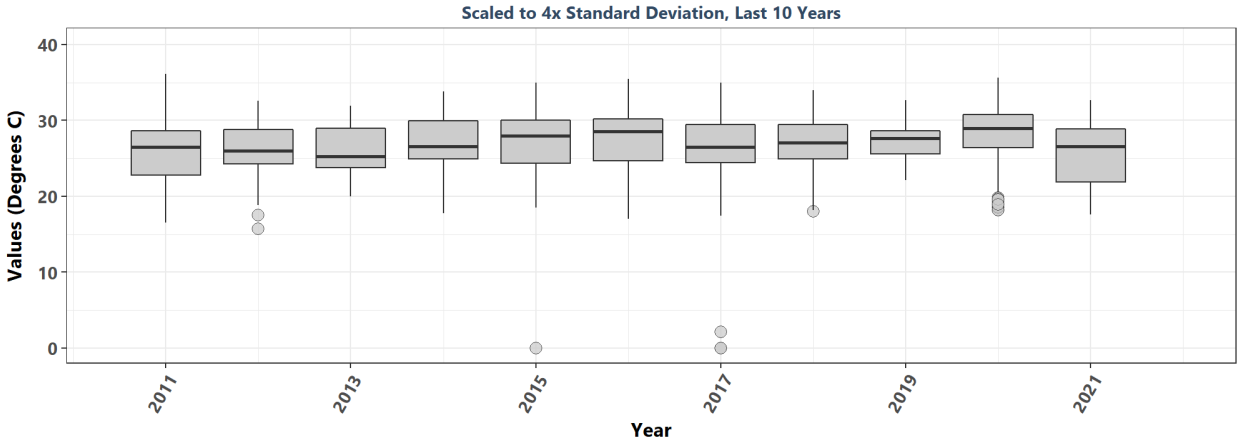

Estero Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6
9
12
Month

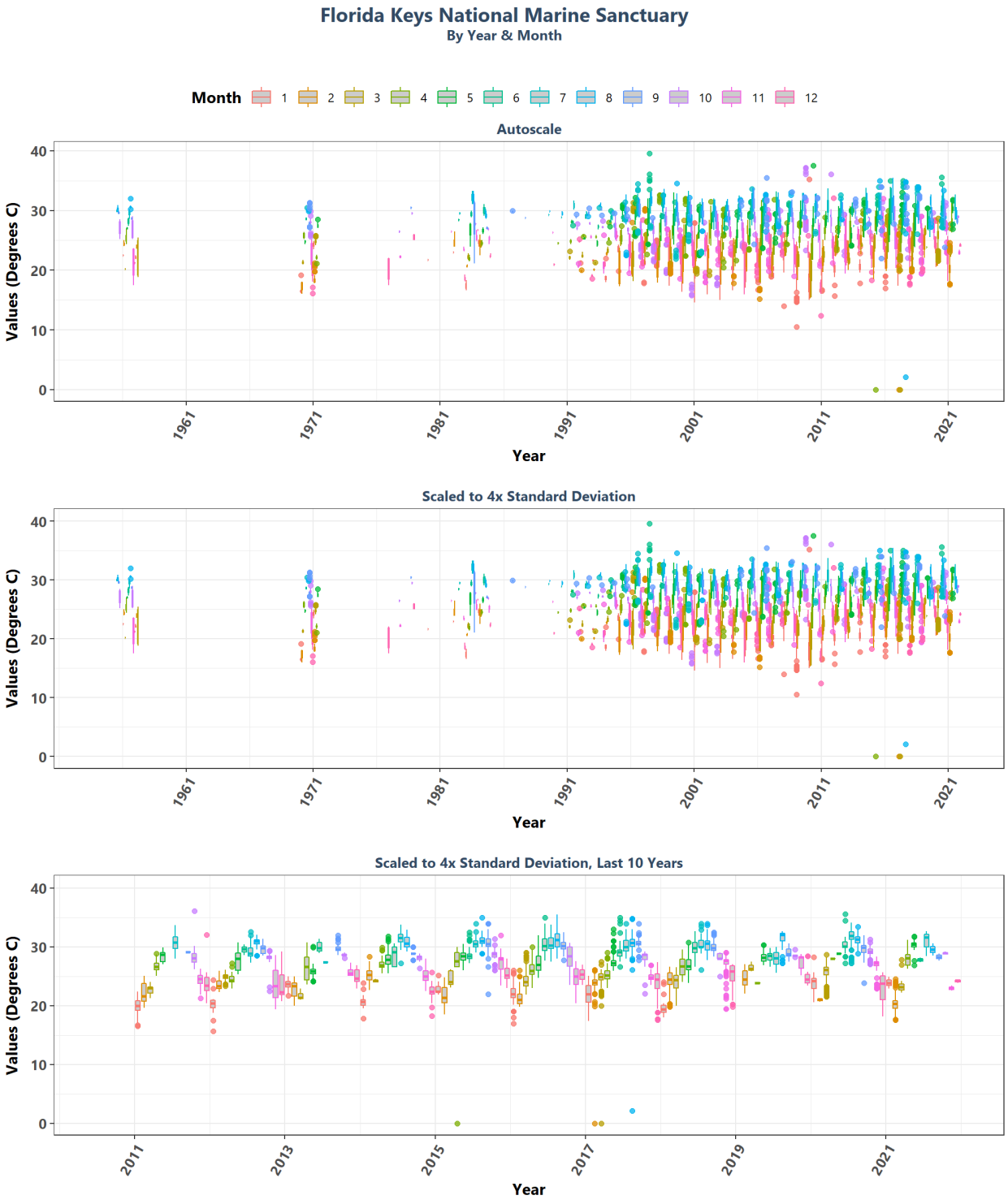
Florida Keys National Marine Sanctuary

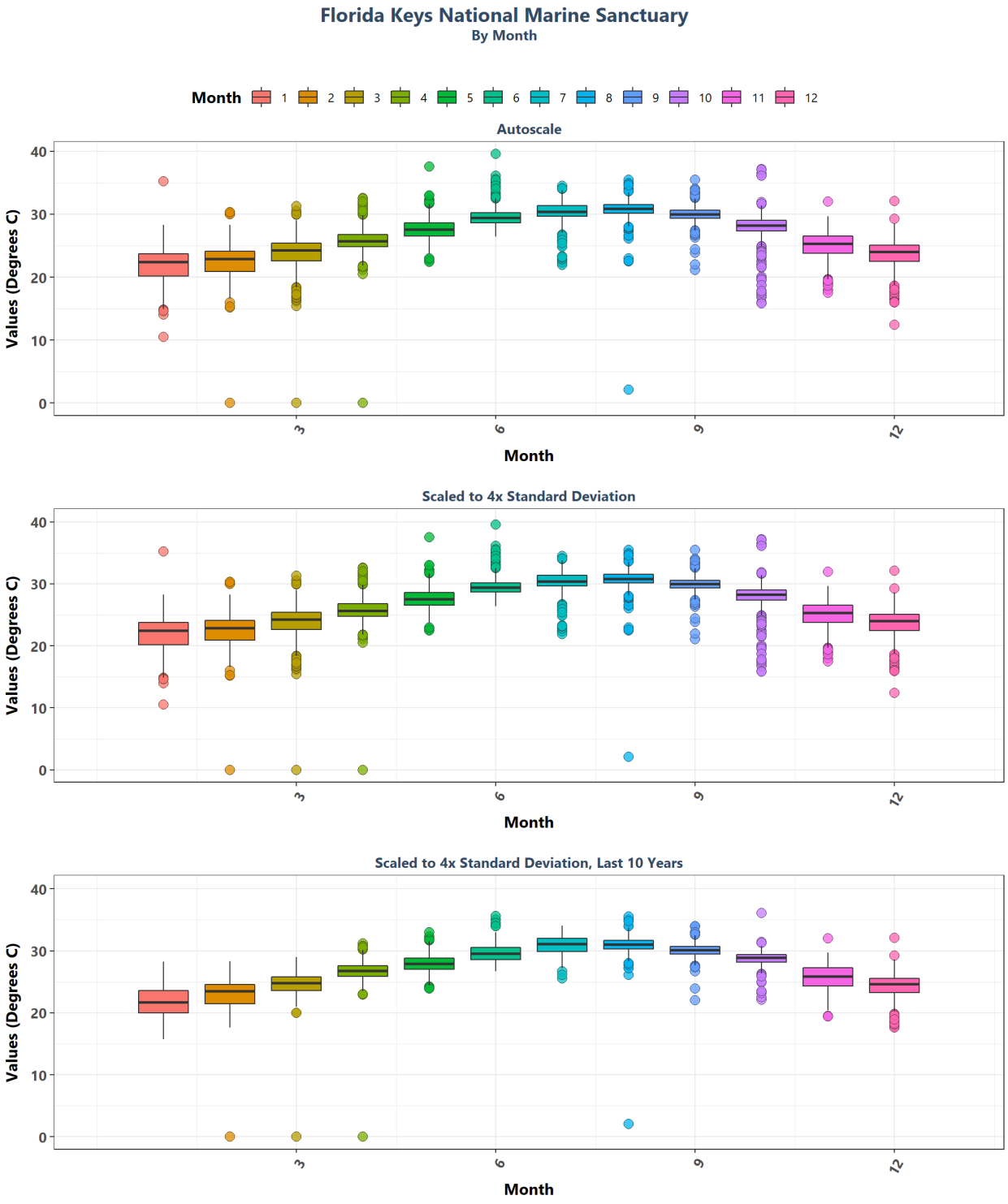
By Year





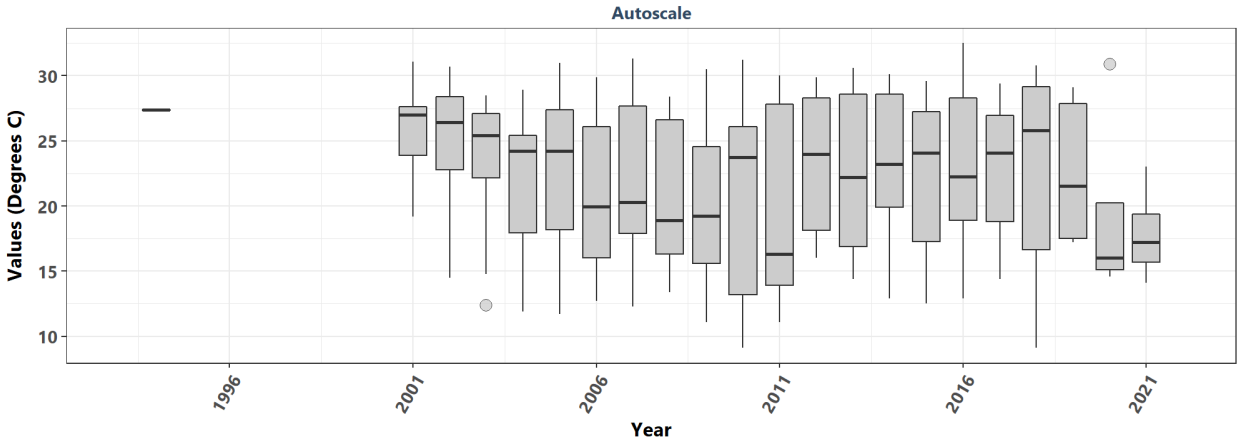


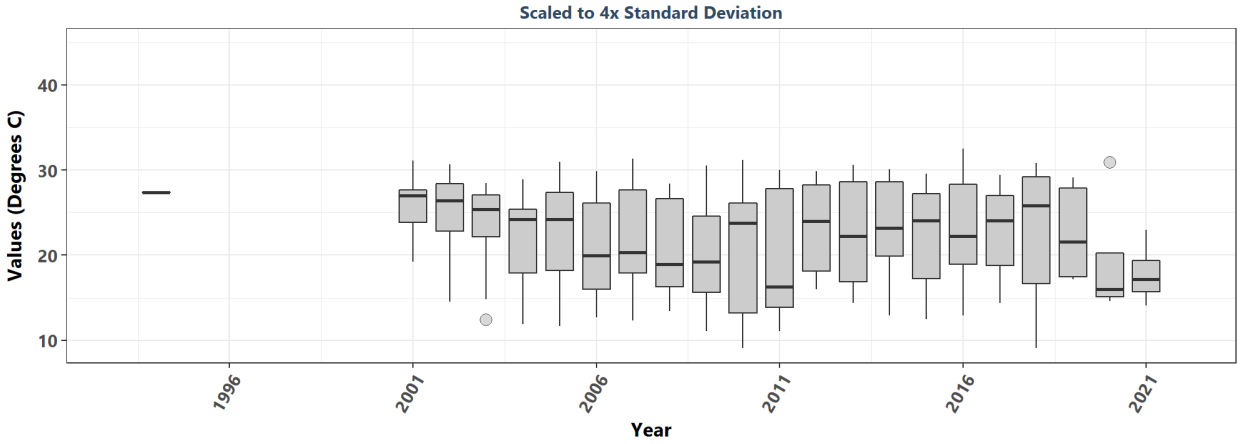

Florida Keys National Marine Sanctuary
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

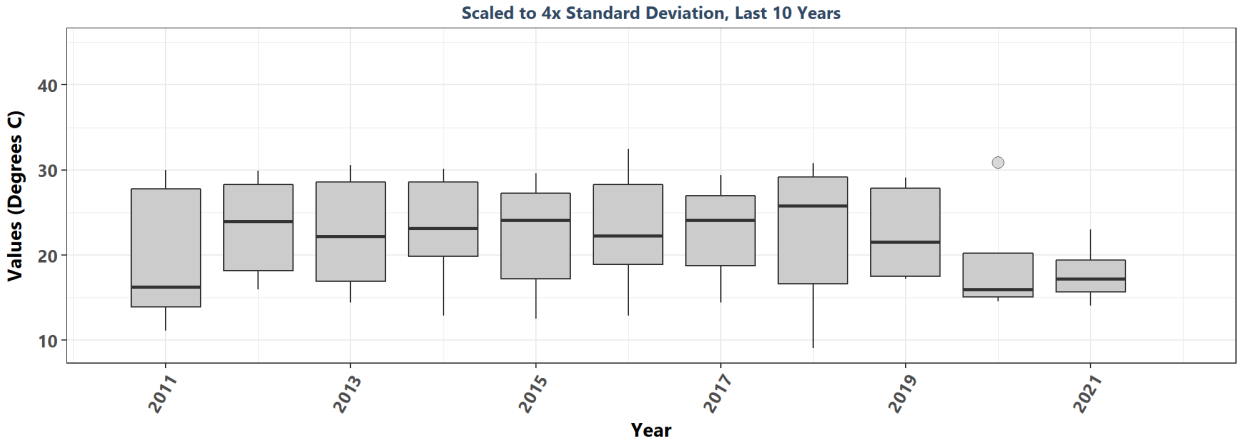

Florida Keys National Marine Sanctuary
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

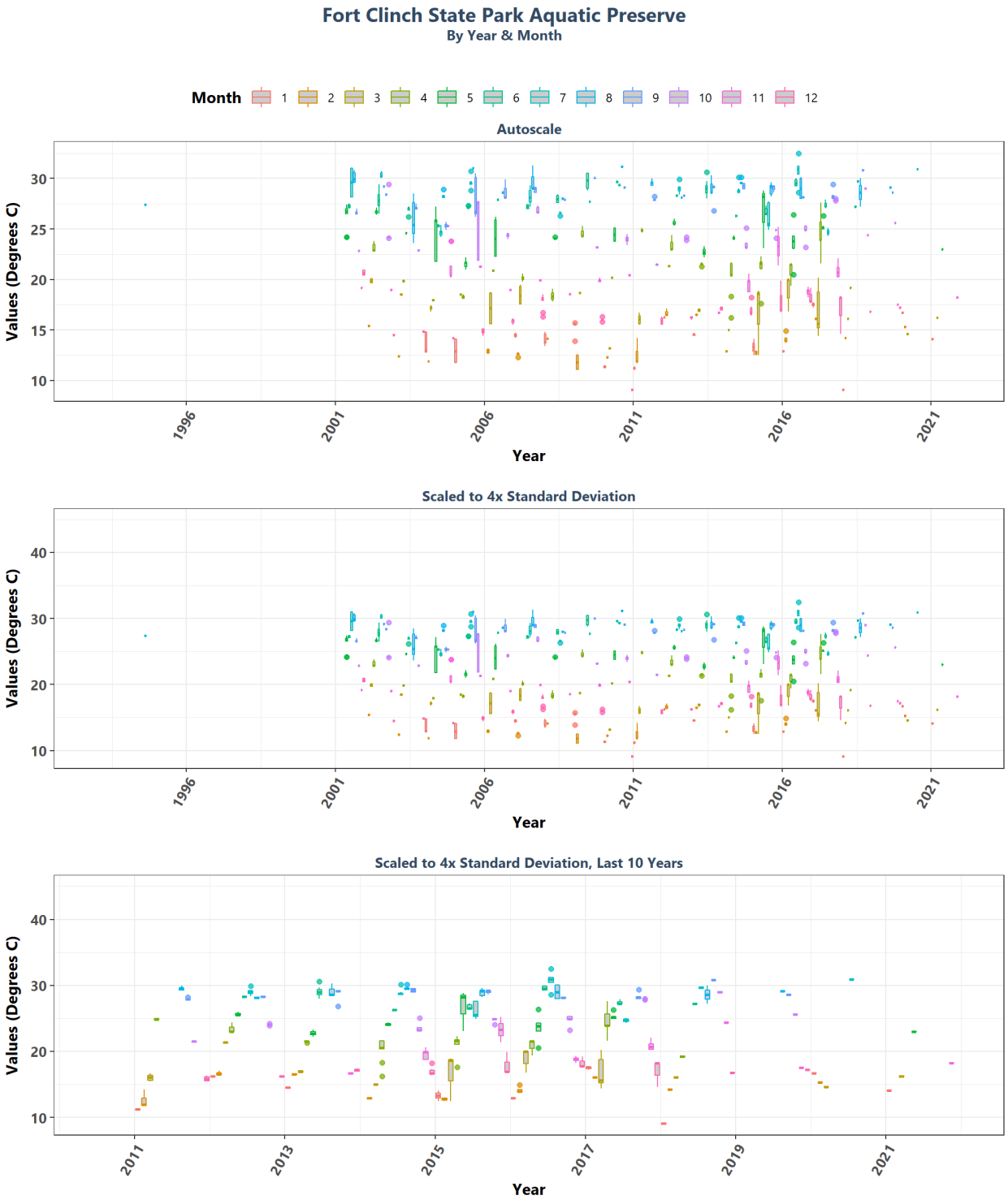
Fort Clinch State Park Aquatic Preserve

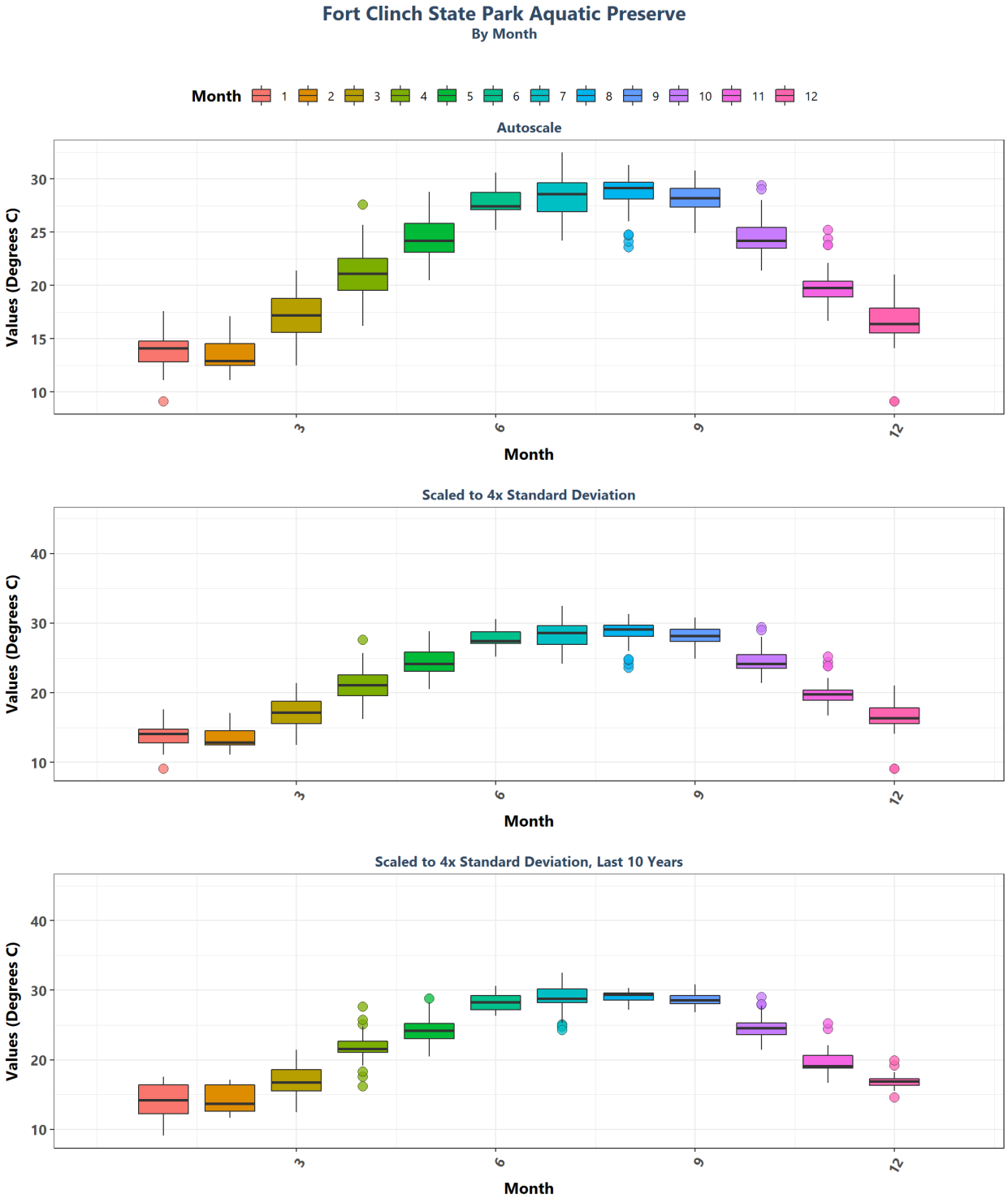
By Year

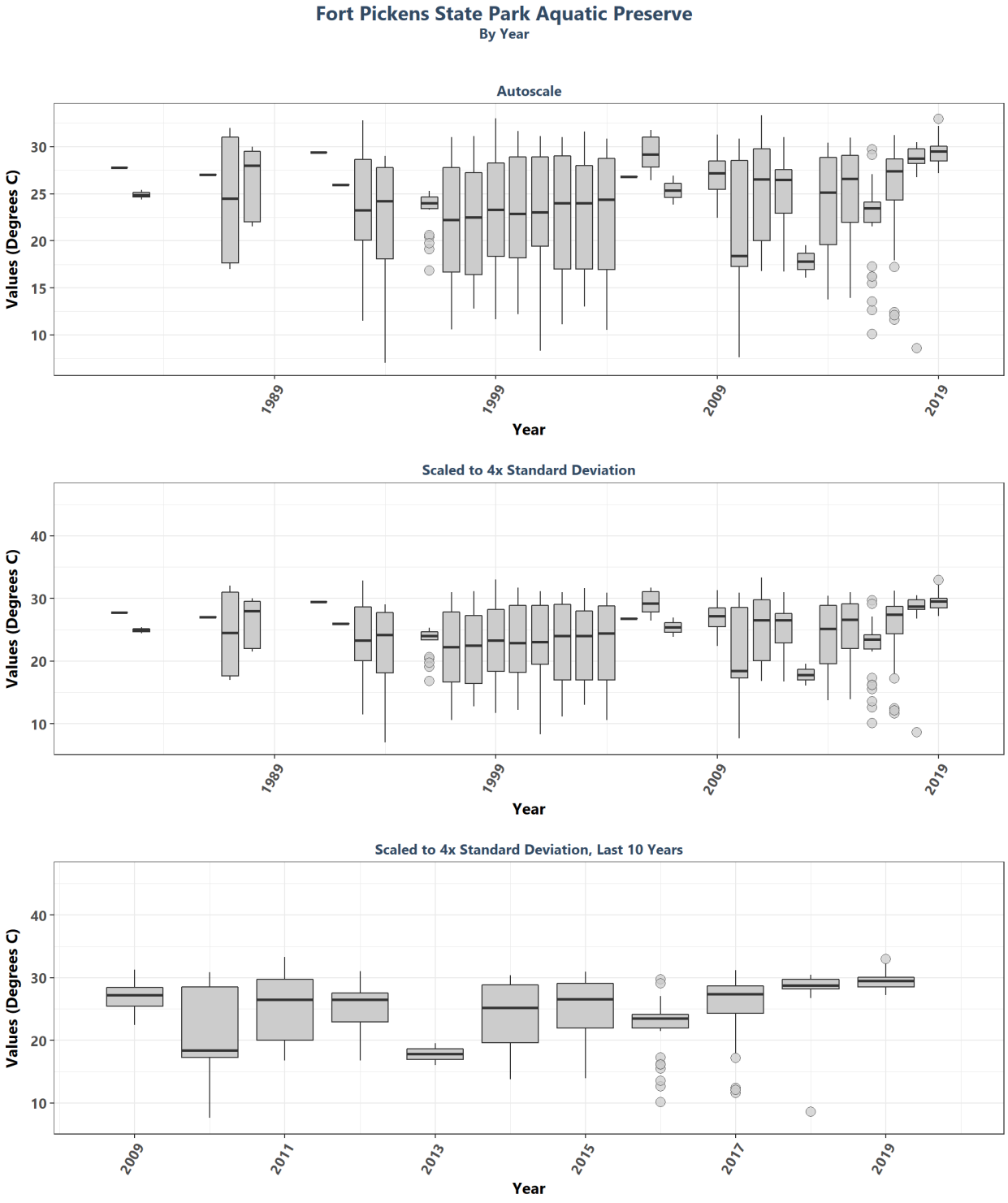


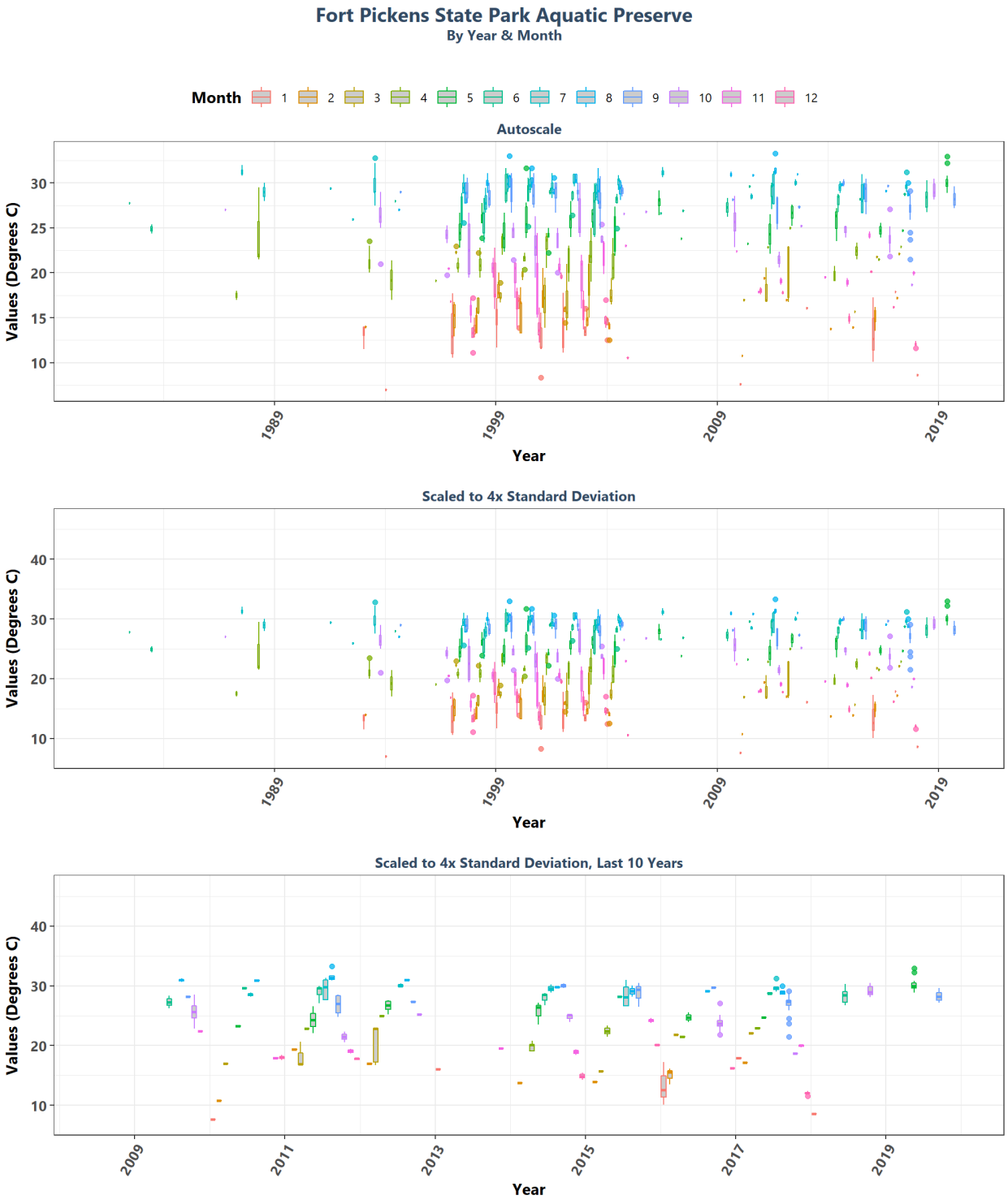


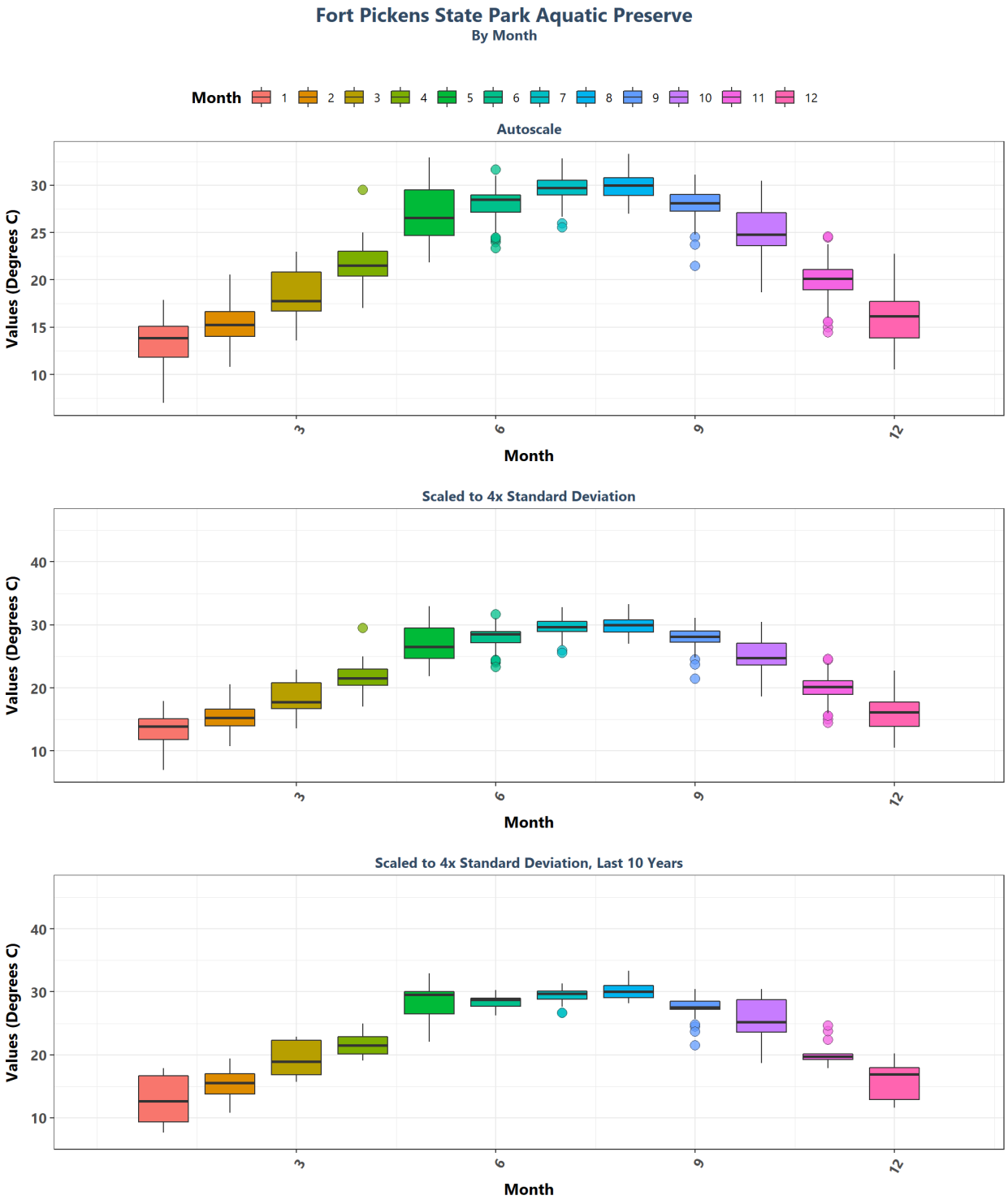



Fort Clinch State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


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

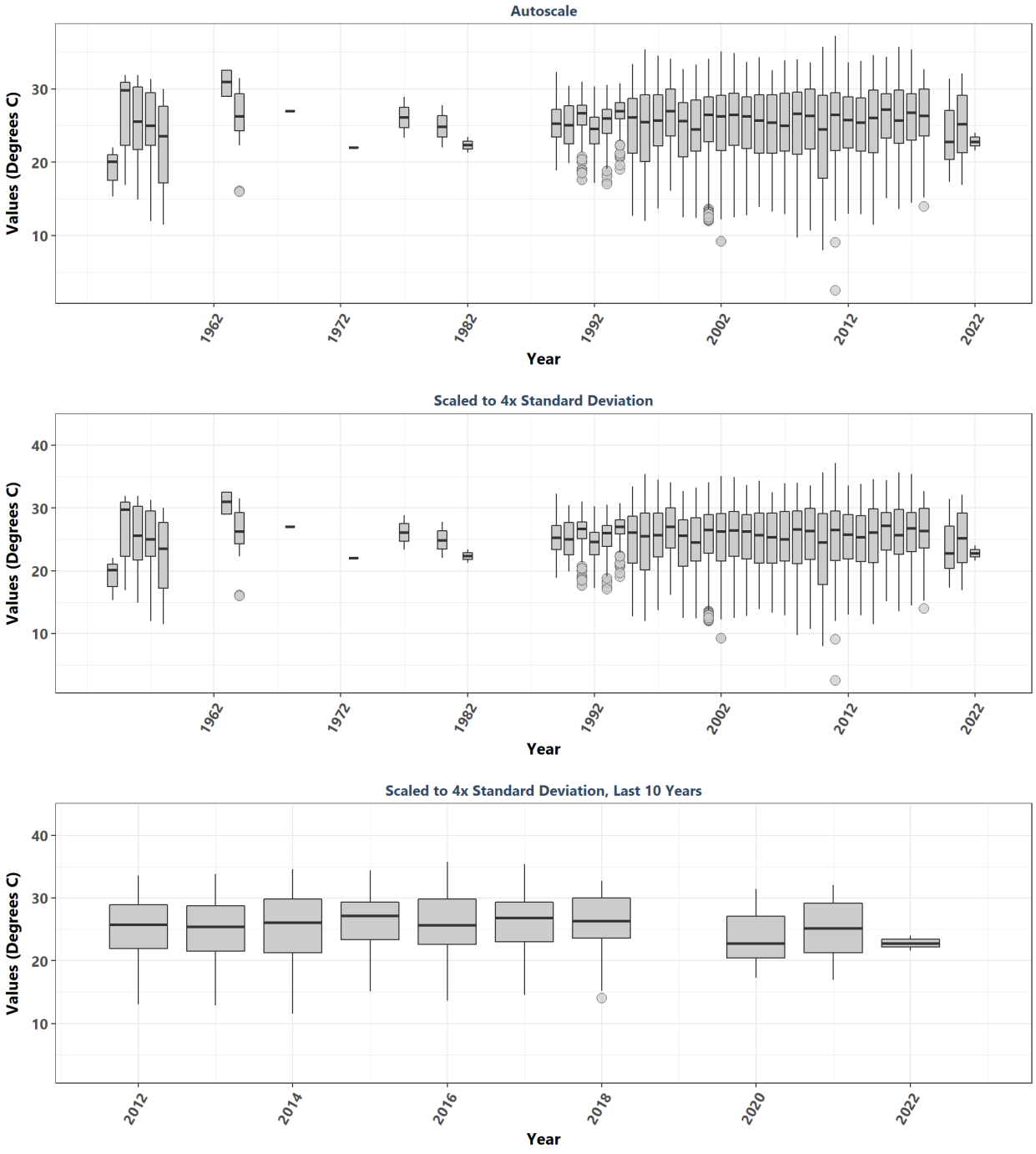

Fort Pickens State Park Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
30
25
20
15
10
1989
1999
2009
2019
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1989
1999
2009
2019
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2009
2011
2013
2015
2017
2019
Year

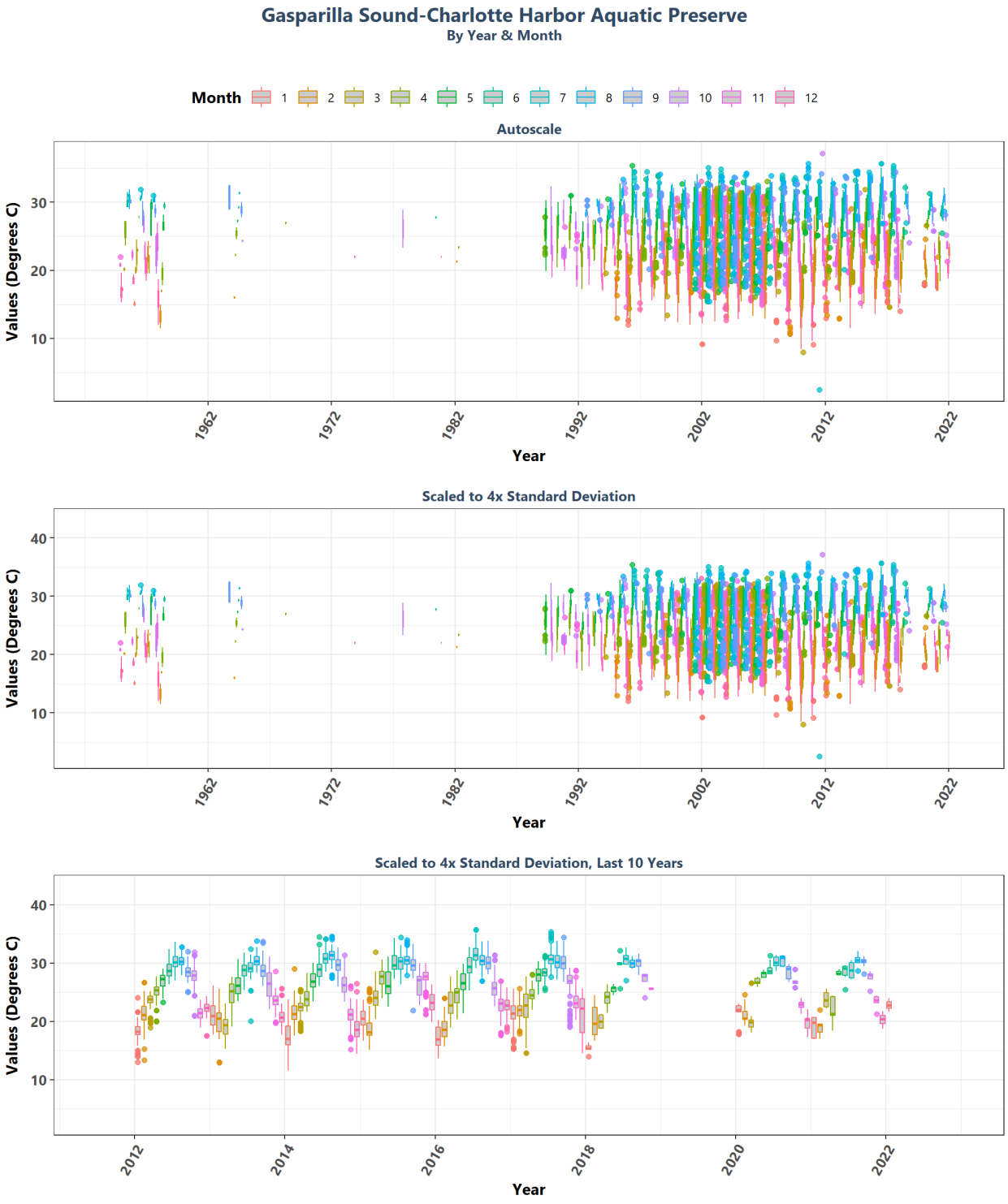

Fort Pickens State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

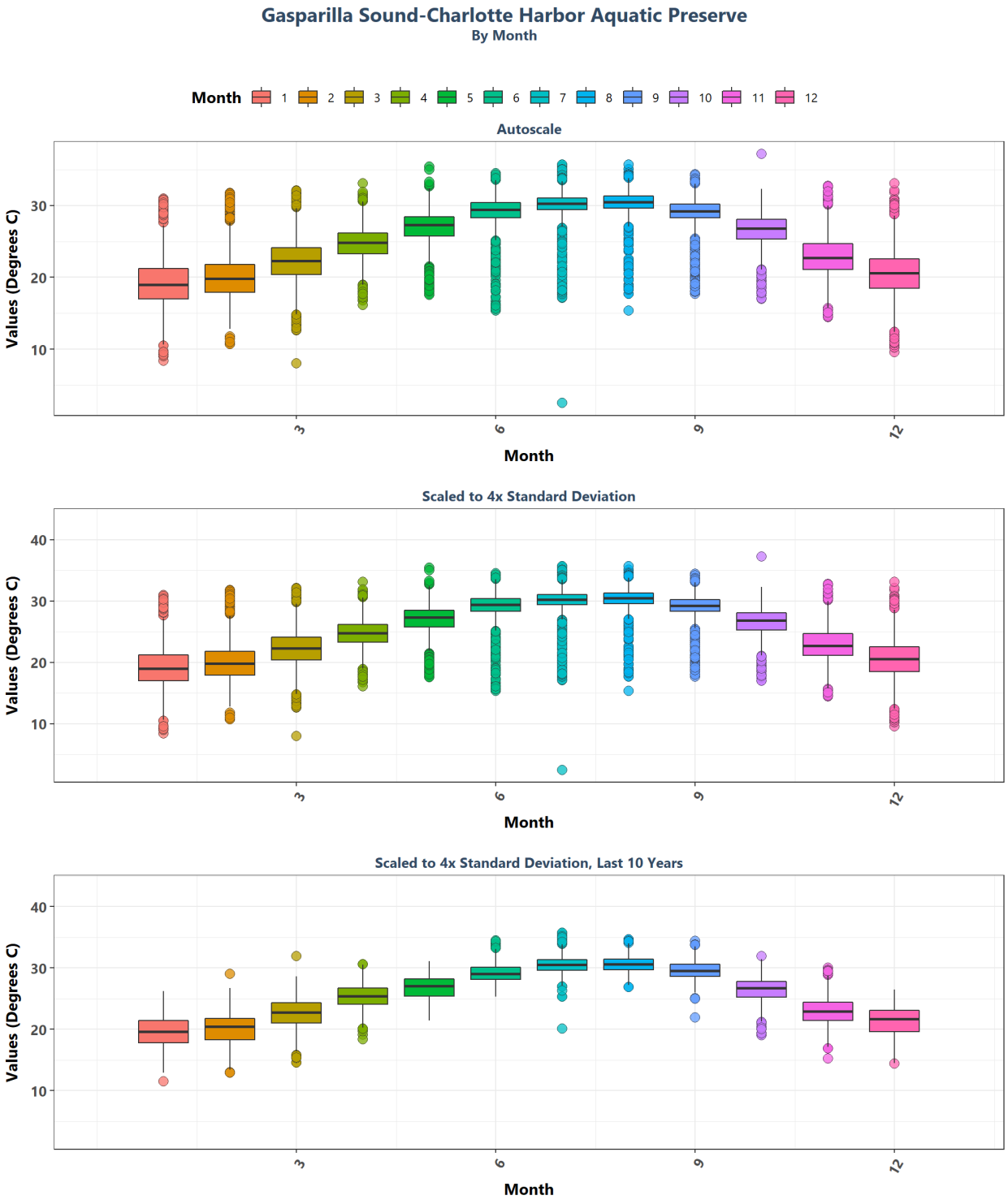

Fort Pickens State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10 15 20 25 30
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10 20 30 40
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10 20 30 40
3 6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

Gasparilla Sound-Charlotte Harbor Aquatic Preserve

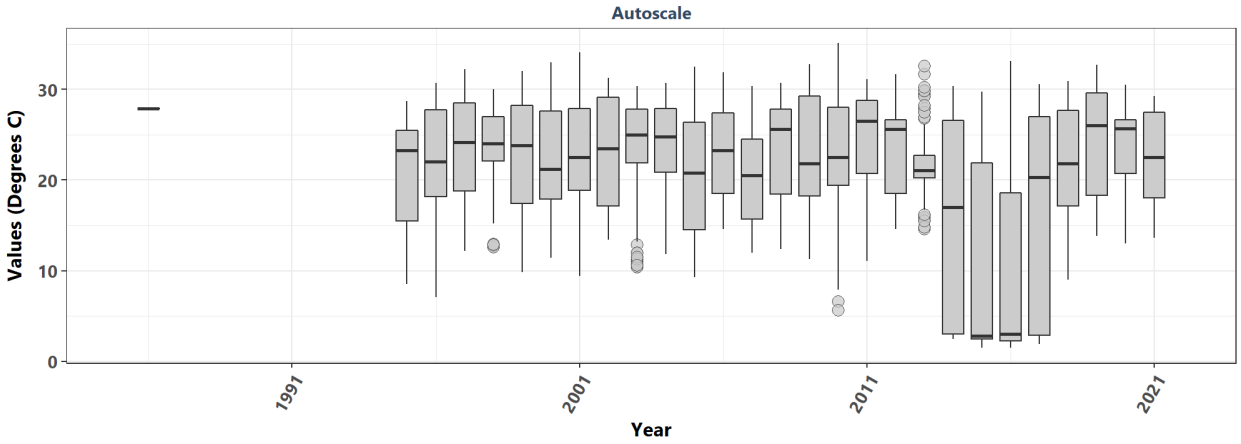
By Year

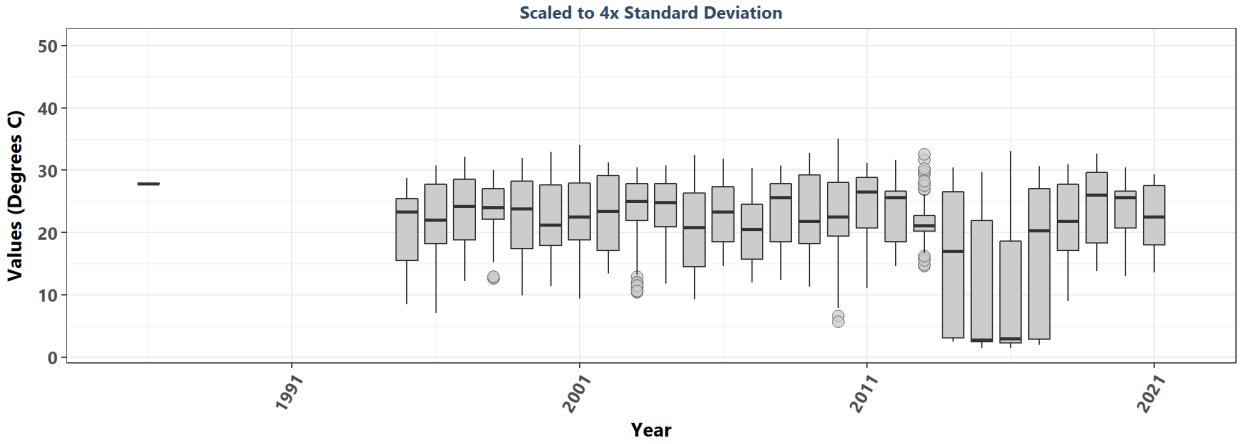


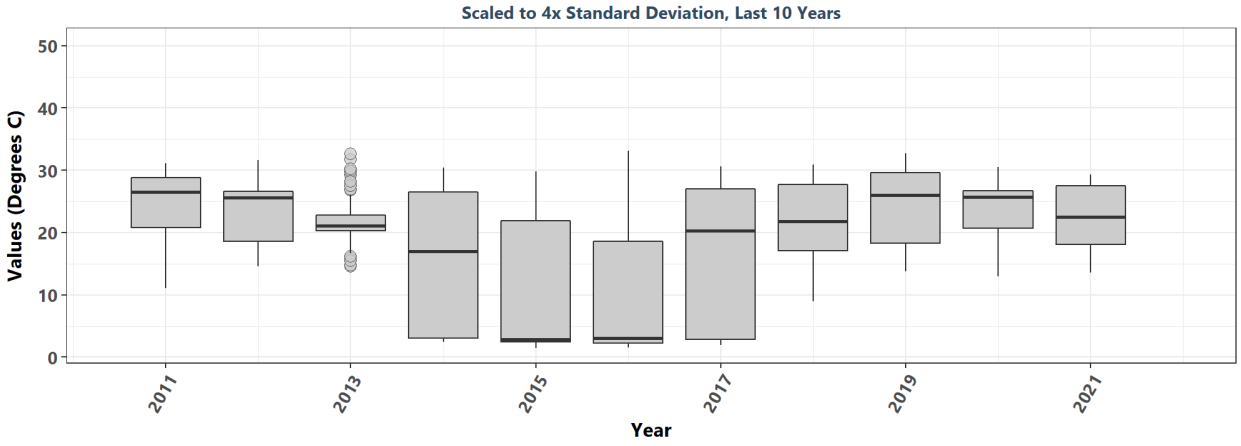

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

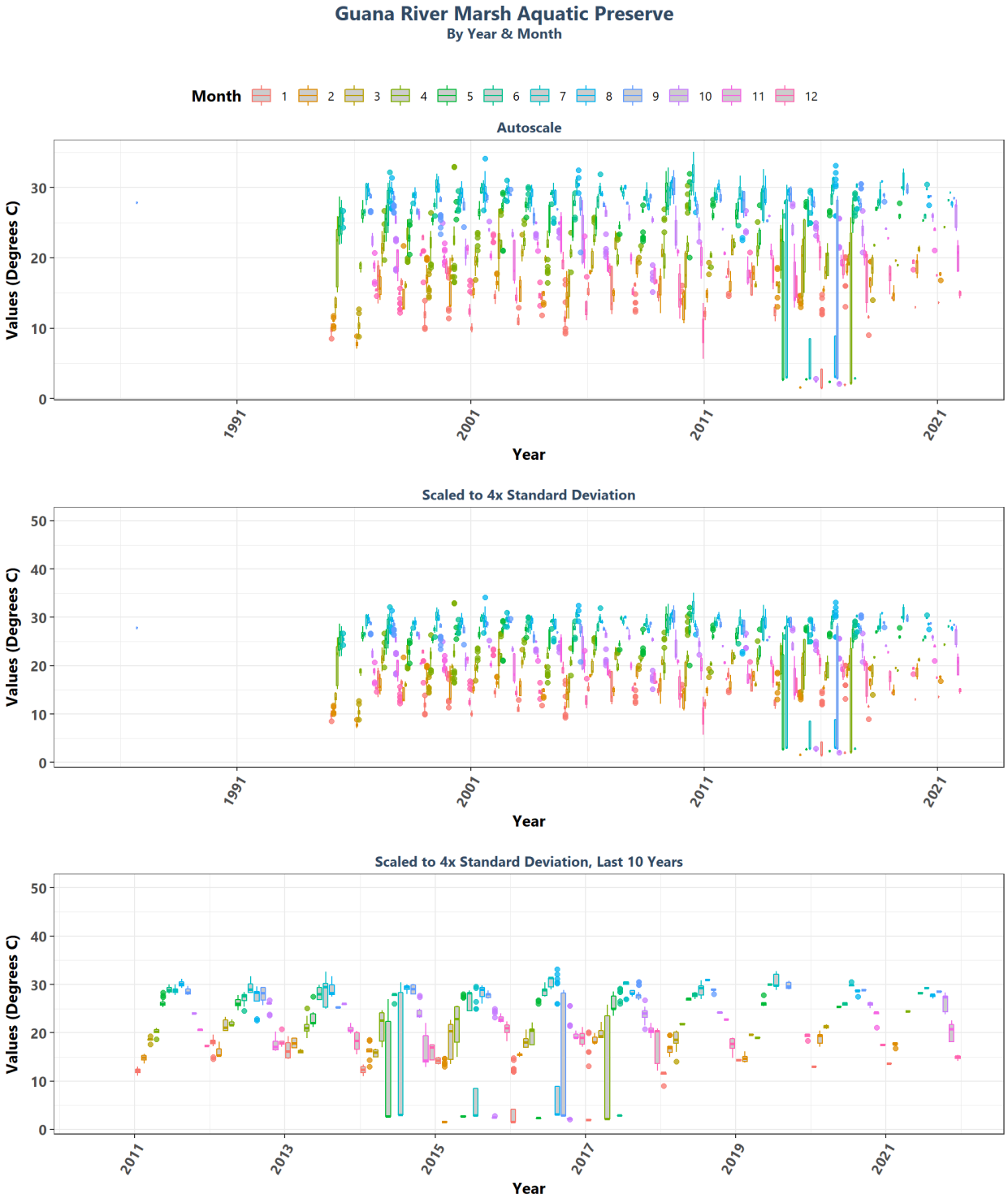

Gasparilla Sound-Charlotte Harbor Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

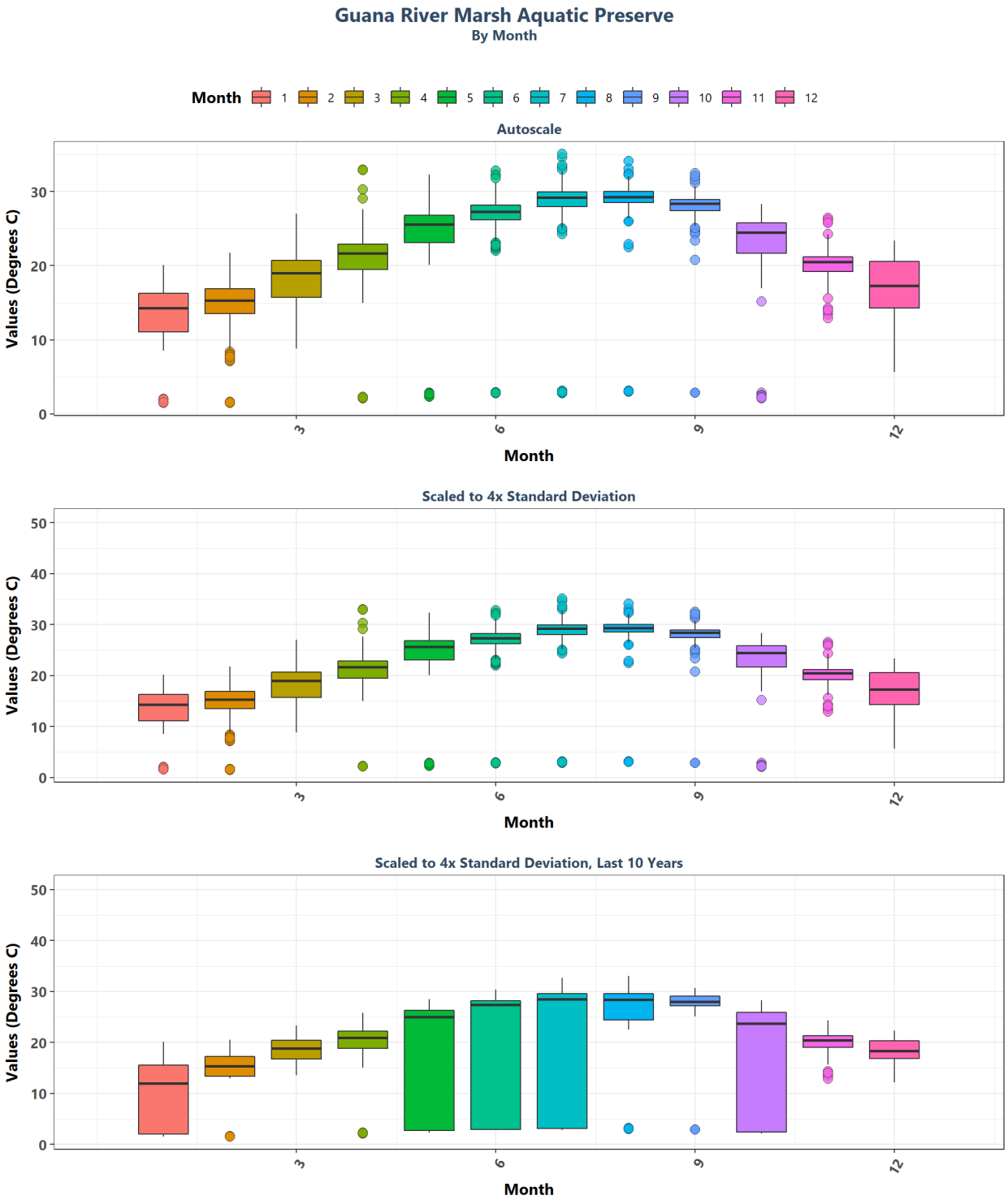
Guana River Marsh Aquatic Preserve
By Year





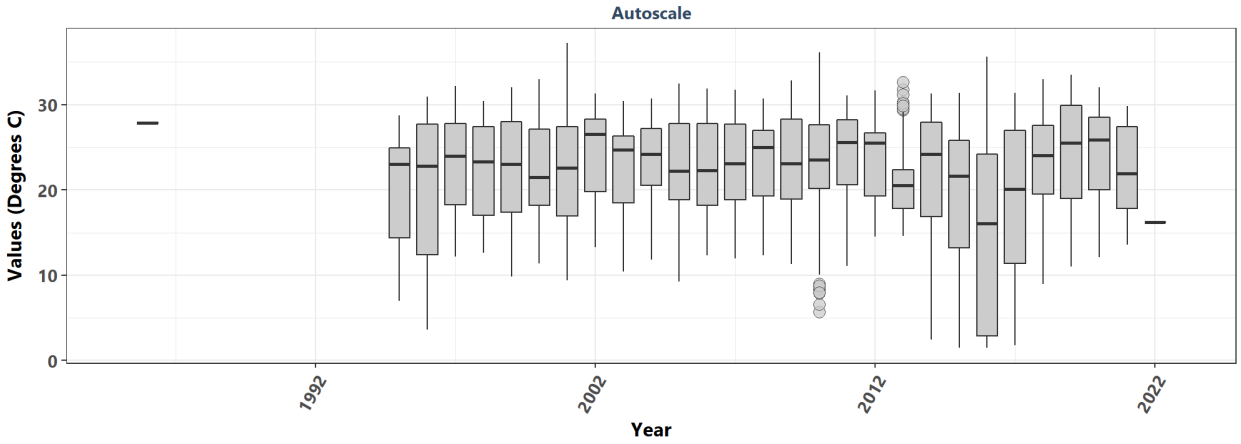


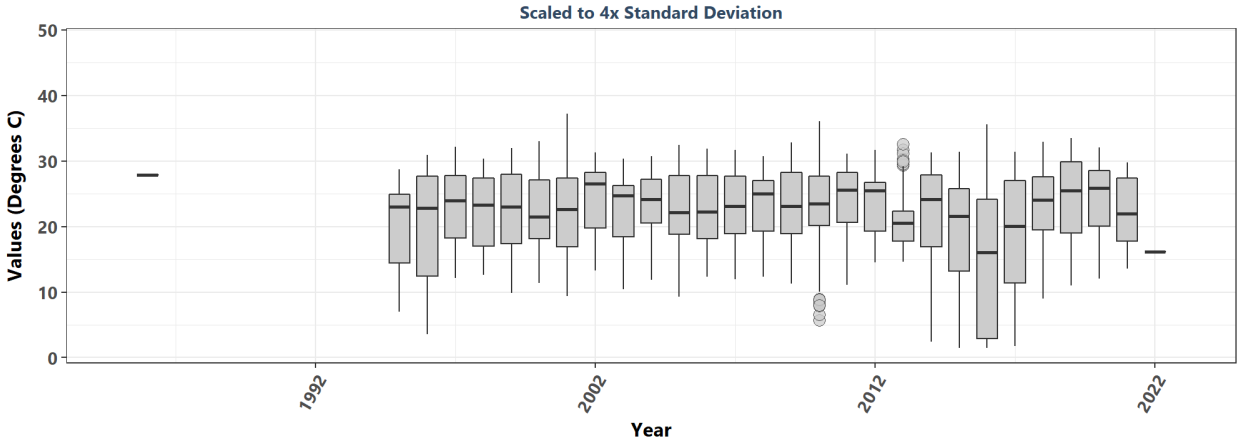

Guana River Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30
1991 2001 2011 2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40 50
1991 2001 2011 2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees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

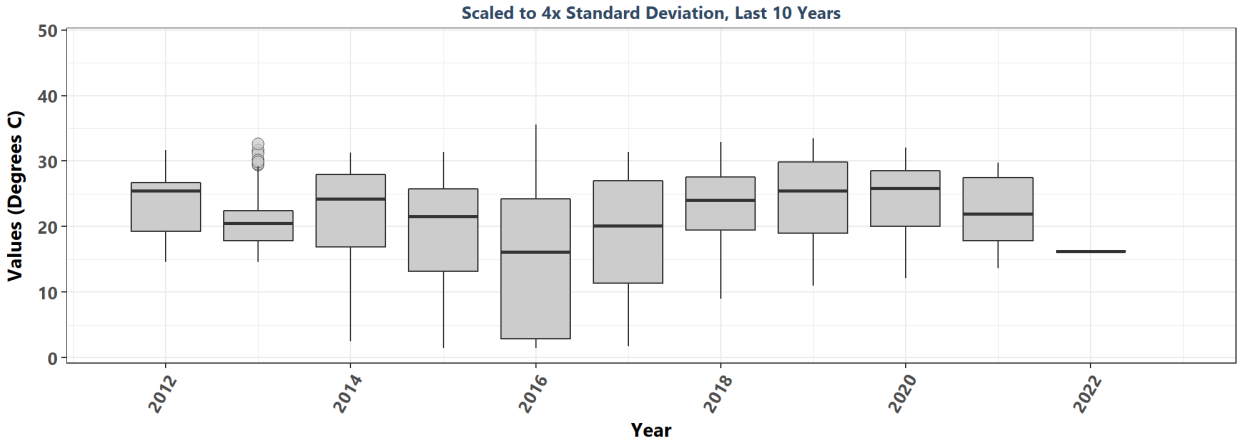

Guana River Marsh Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30
3 6 9 12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40 50
3 6 9 12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30 40 50
3 6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

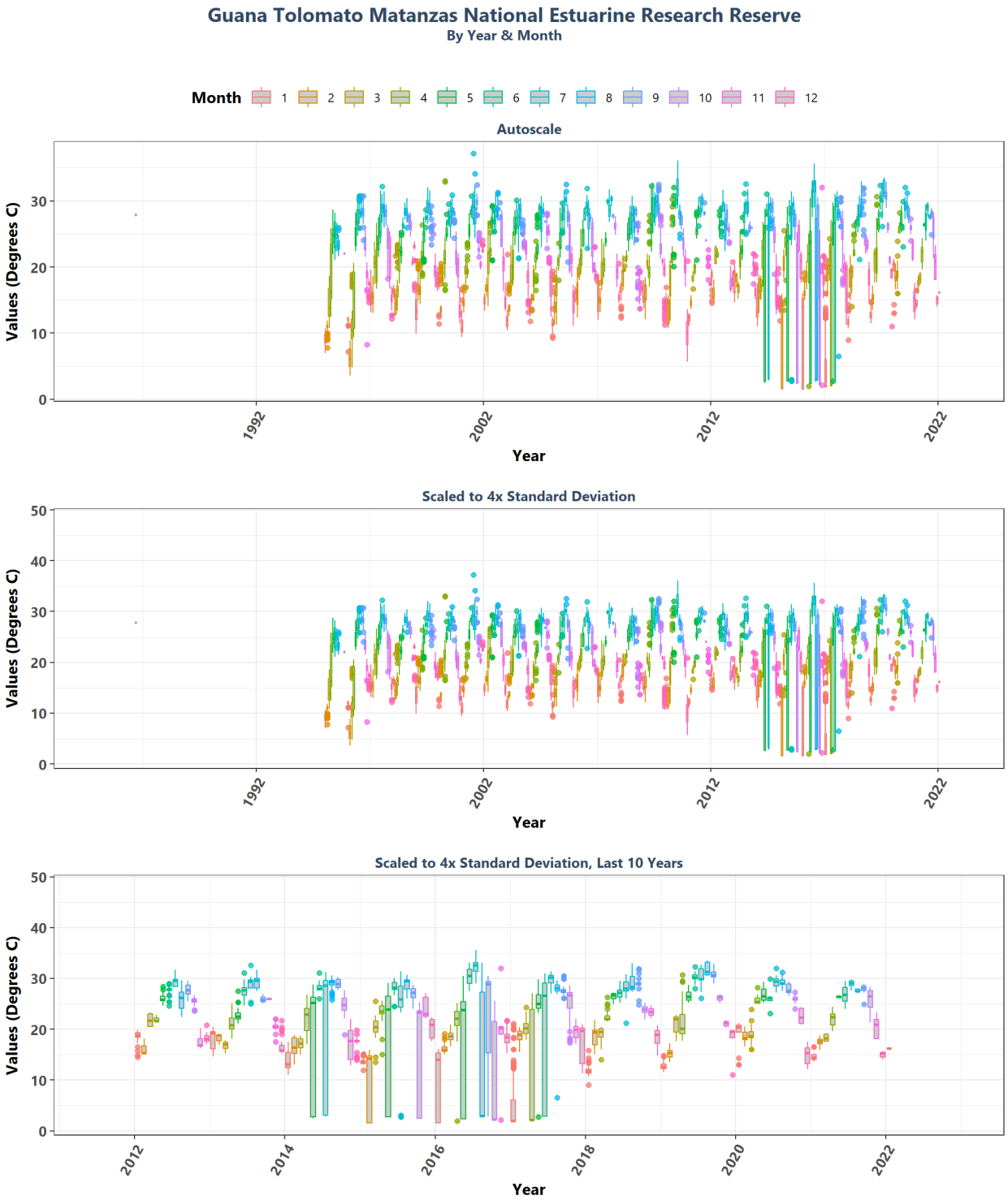
Guana Tolomato Matanzas National Estuarine Research Reserve

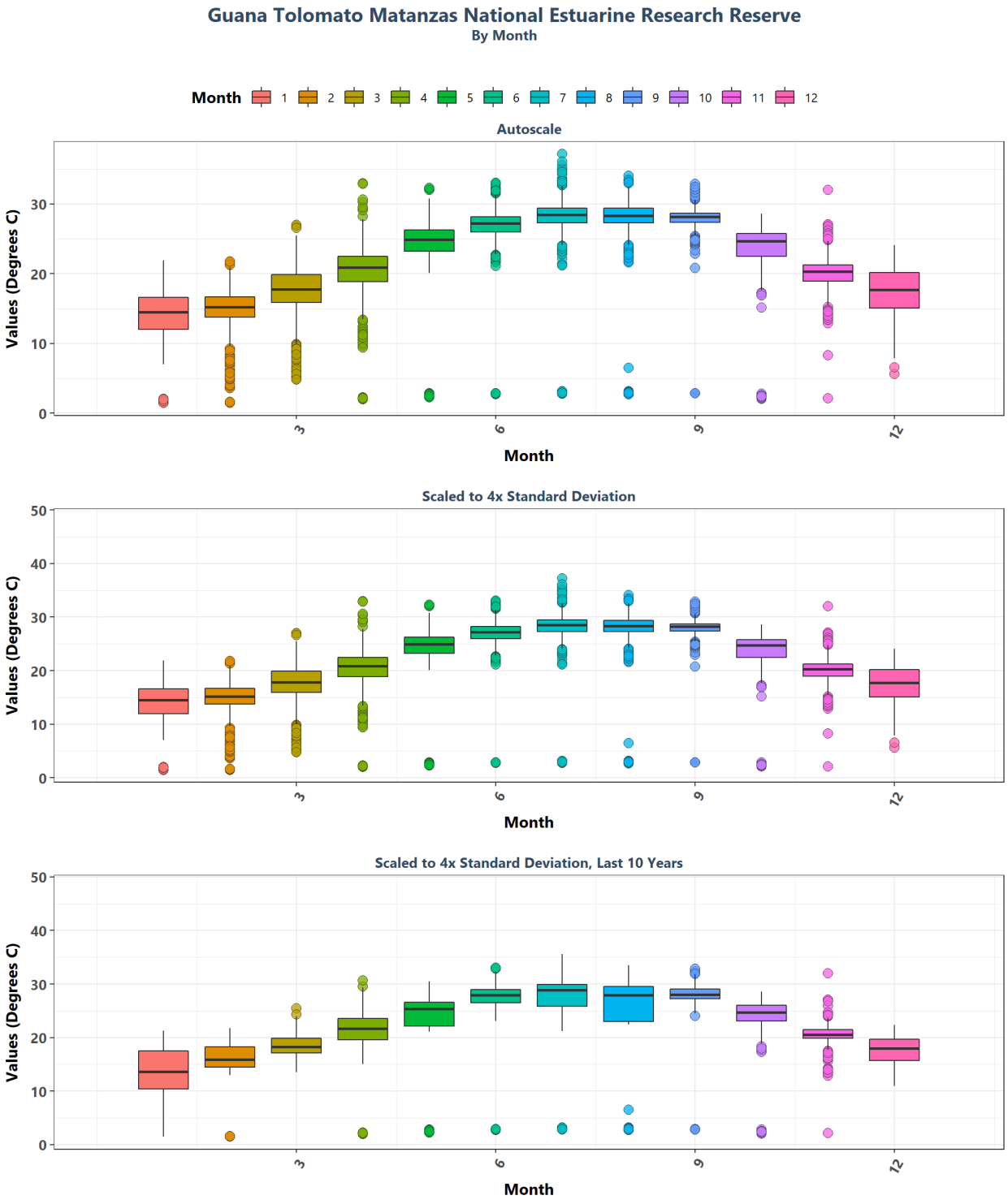
By Year

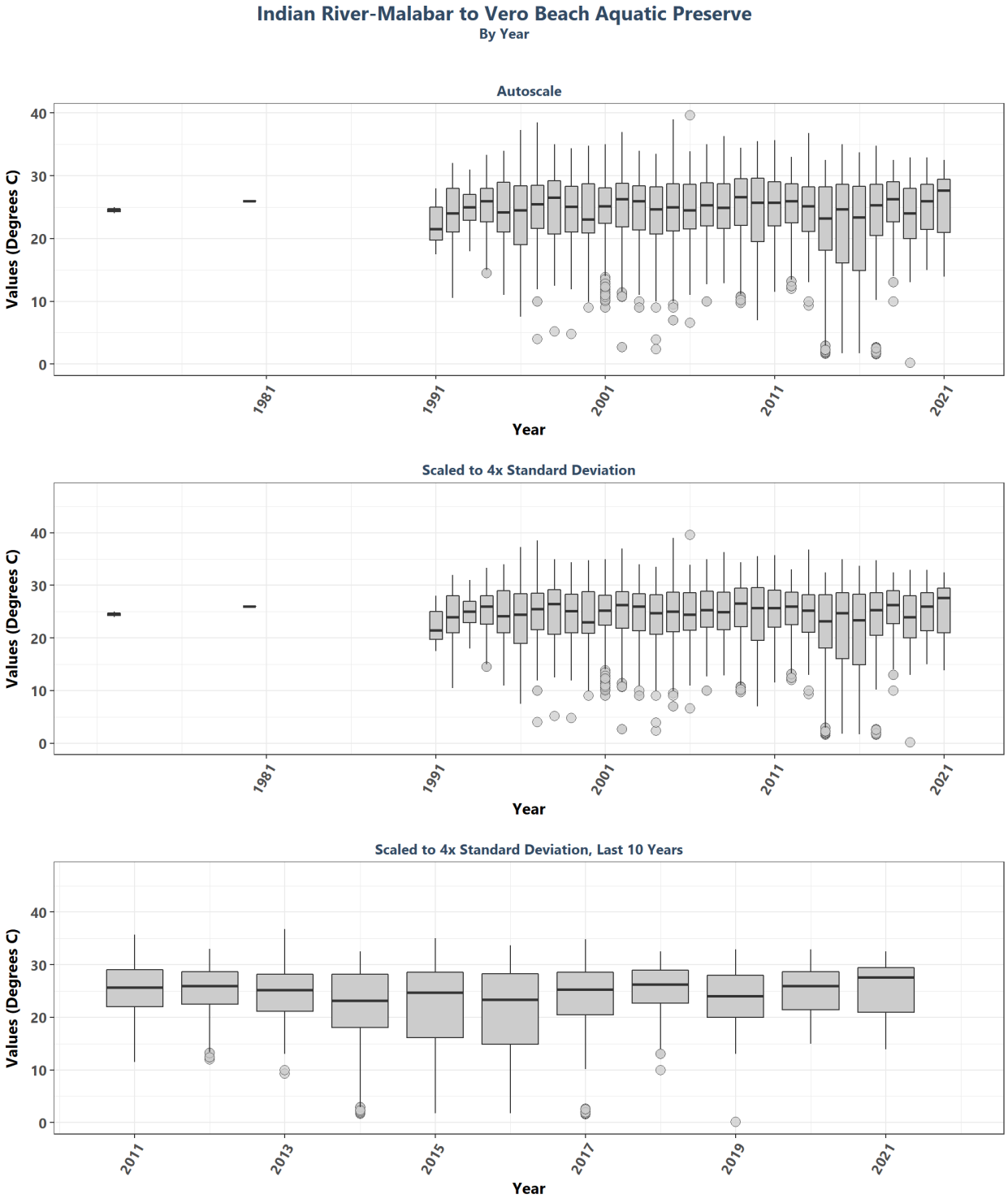


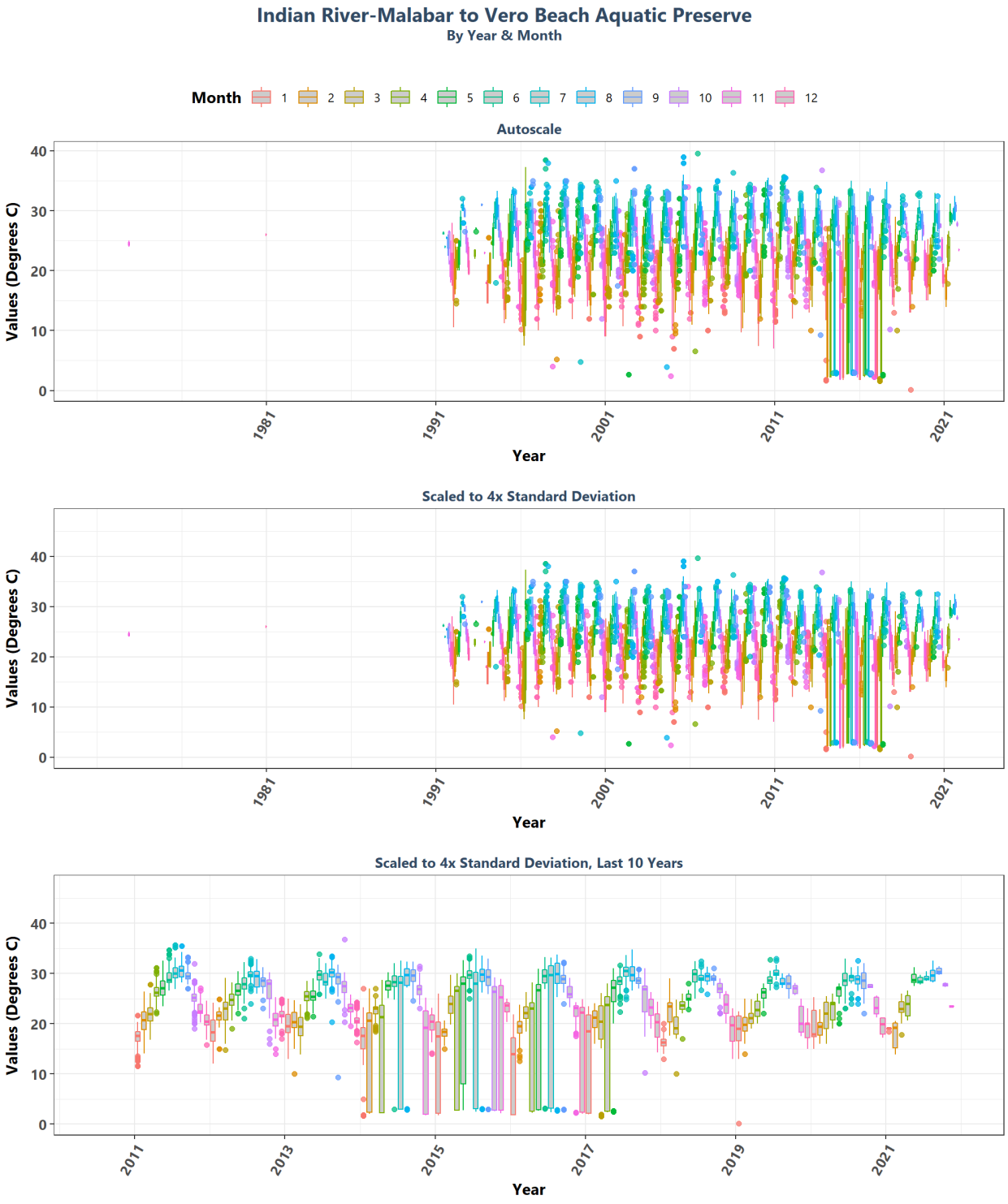


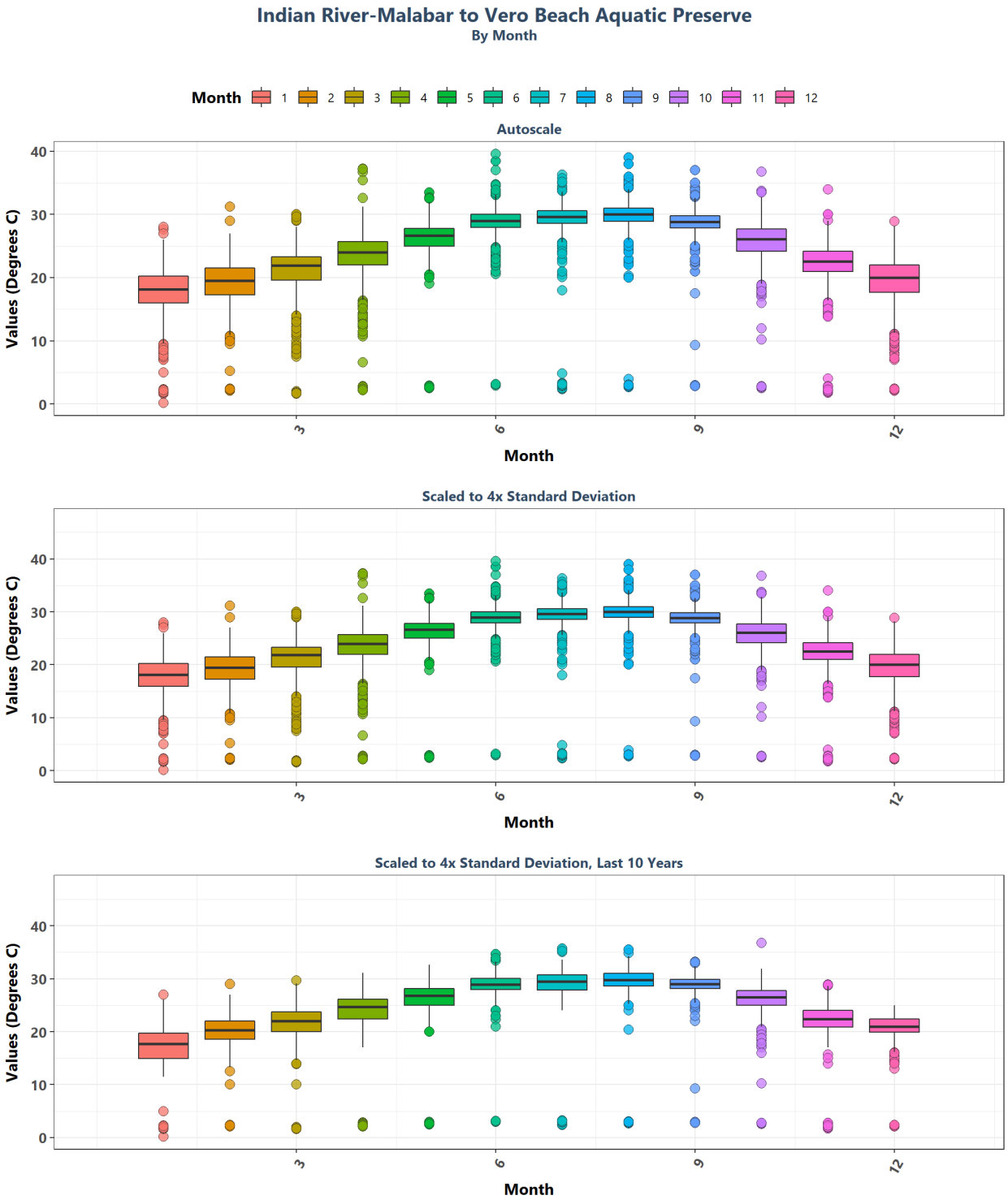



Guana Tolomato Matanzas National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


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

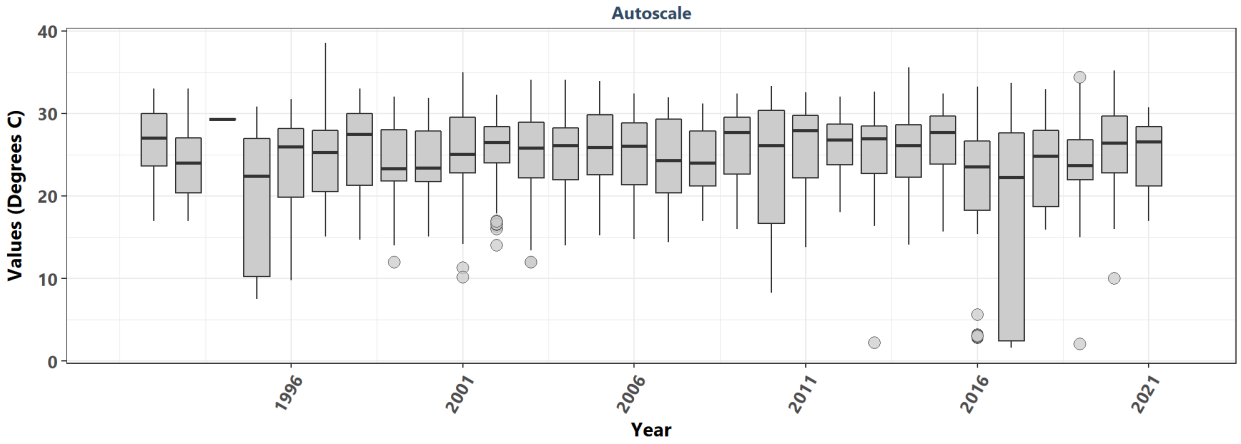

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

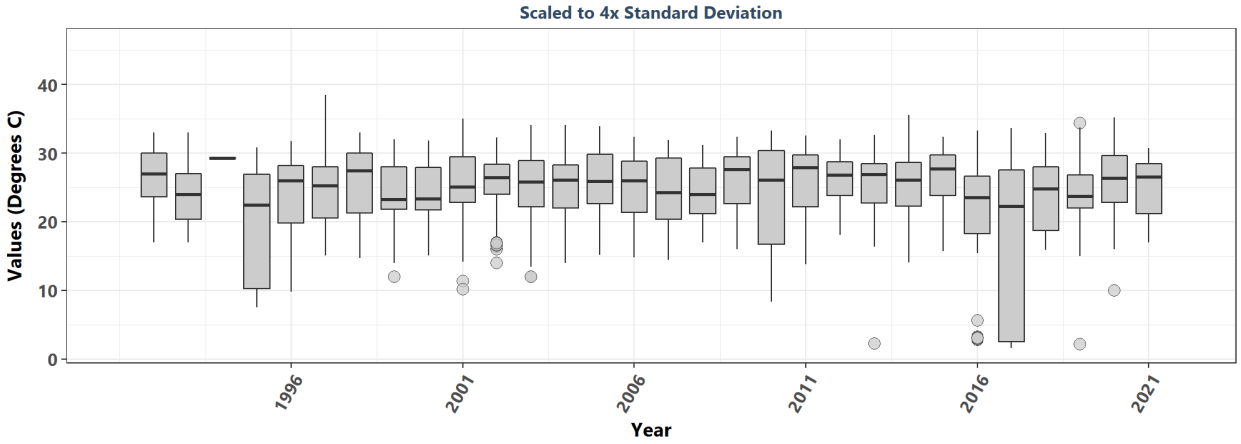

Indian River-Malabar to Vero Beach Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

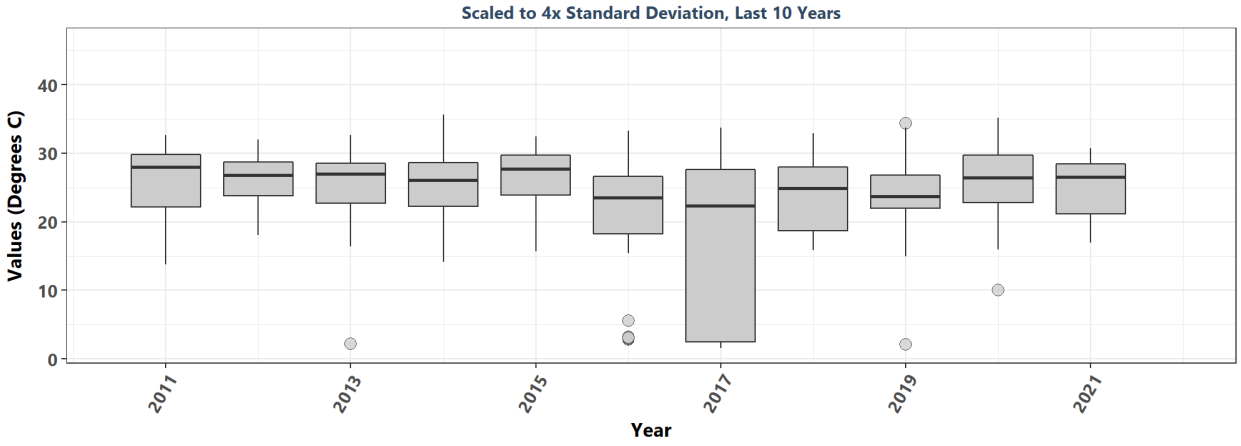

Indian River-Malabar to Vero Beach Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

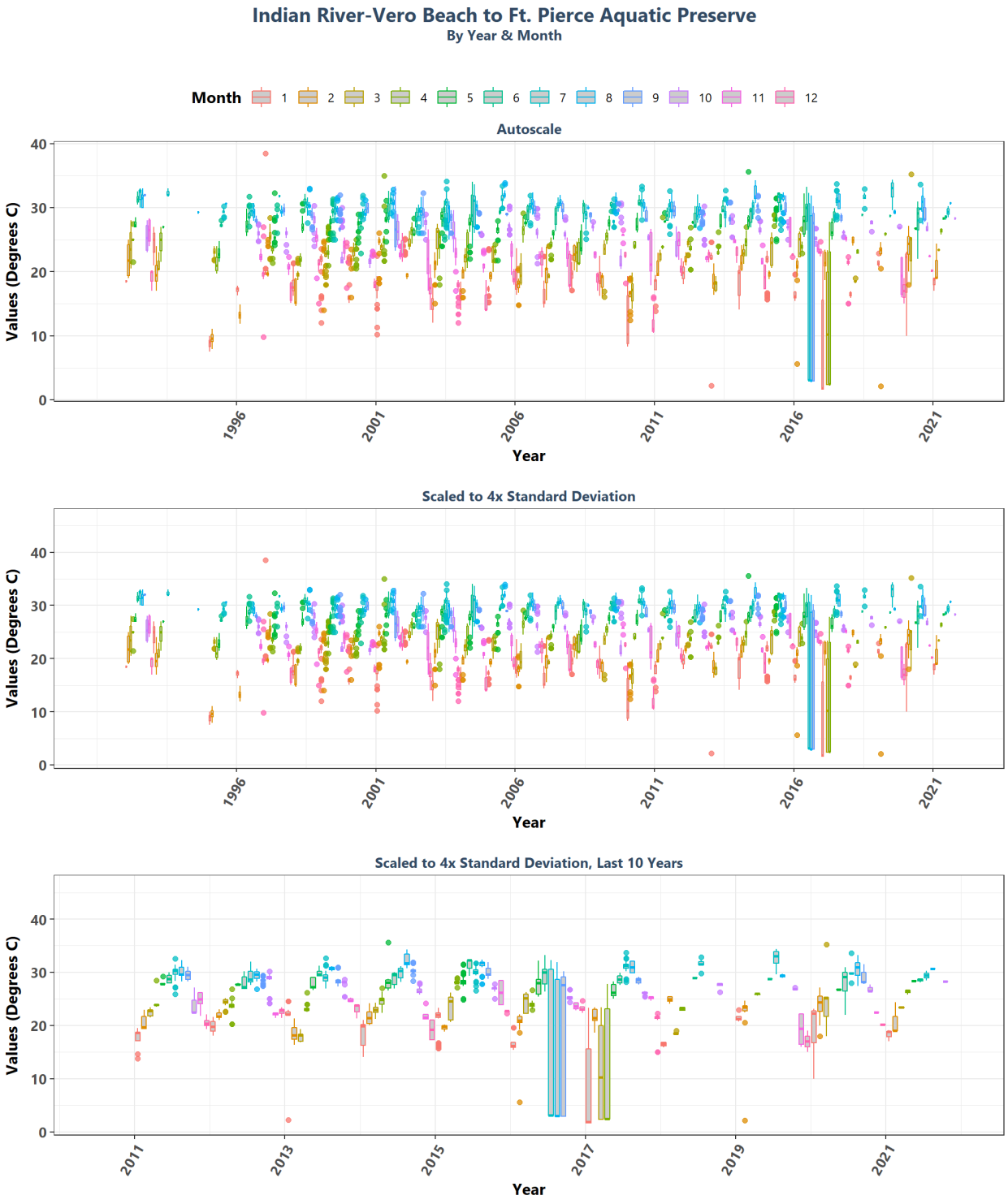
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

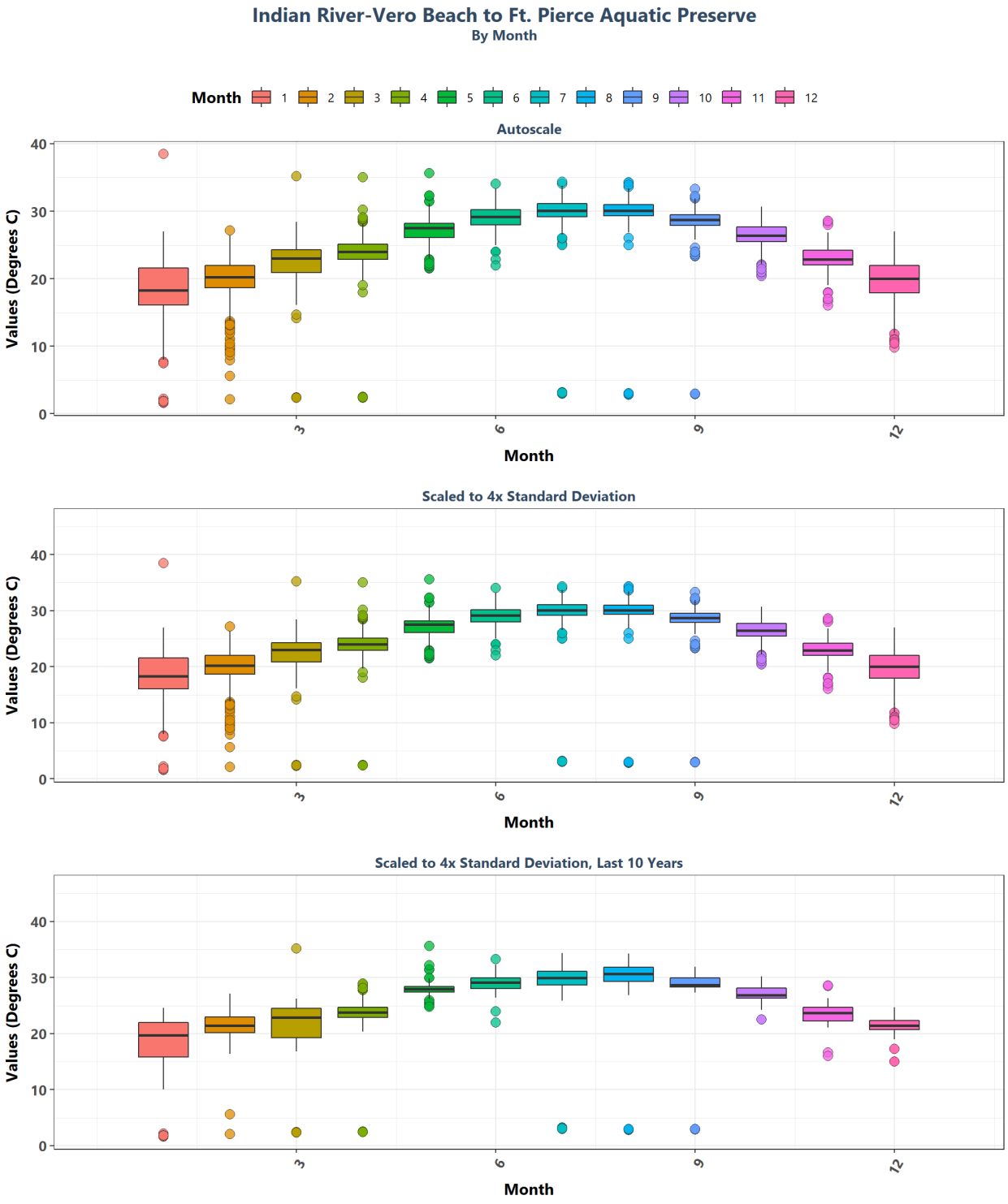
By Year





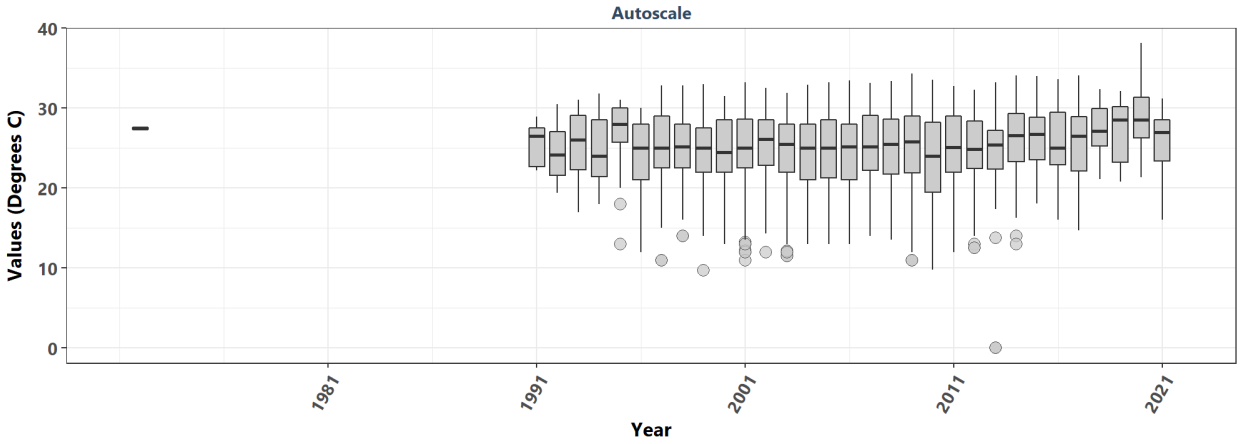


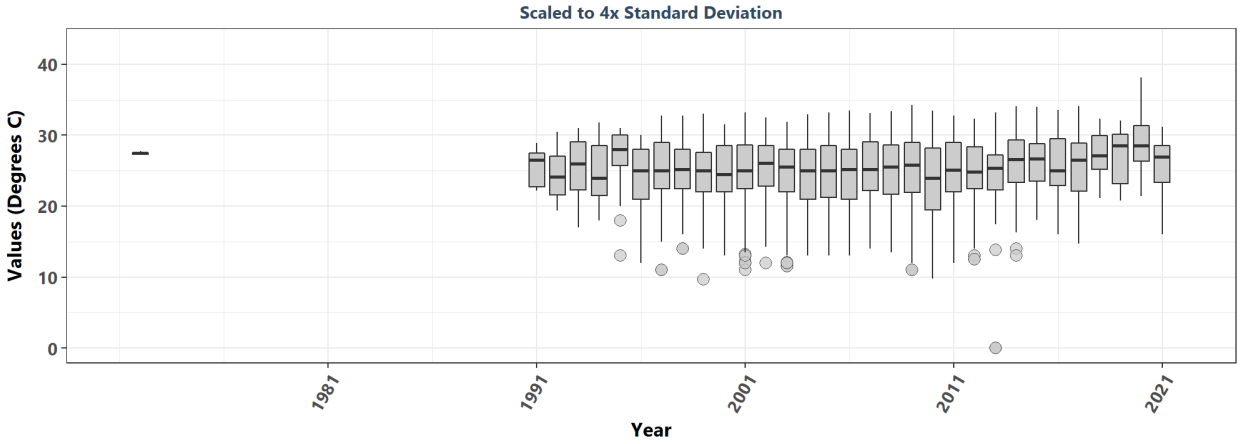

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

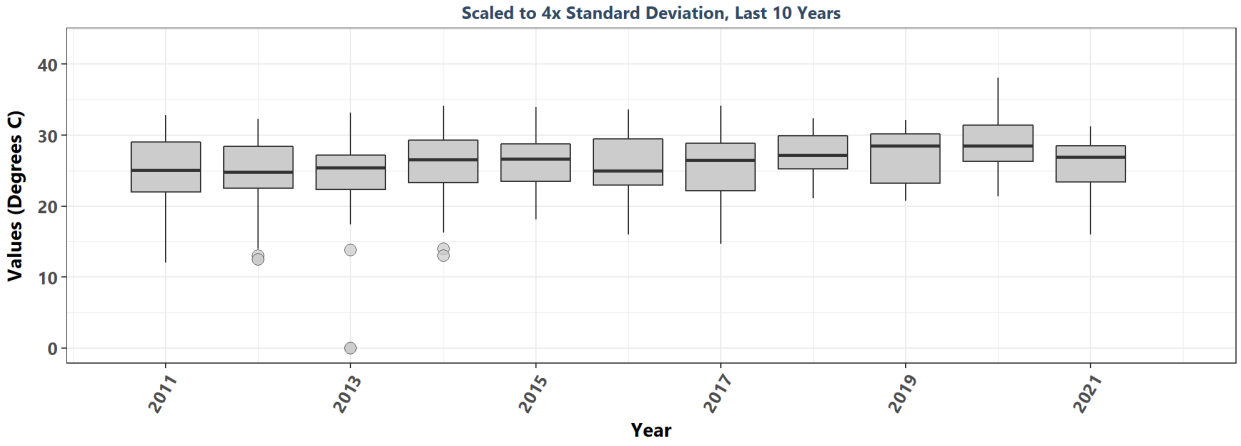

Indian River-Vero Beach to Ft. Pierce Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

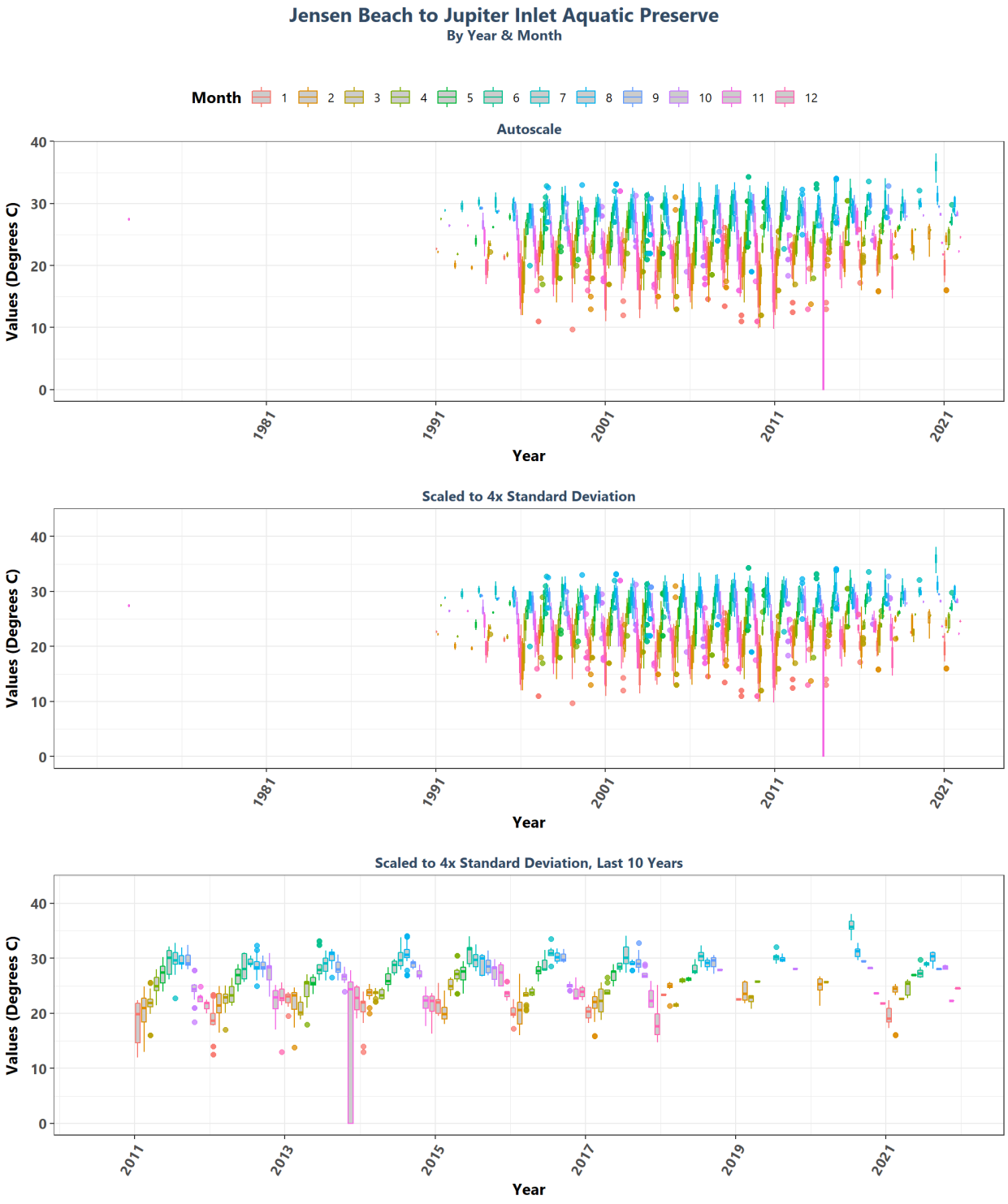
Jensen Beach to Jupiter Inlet Aquatic Preserve

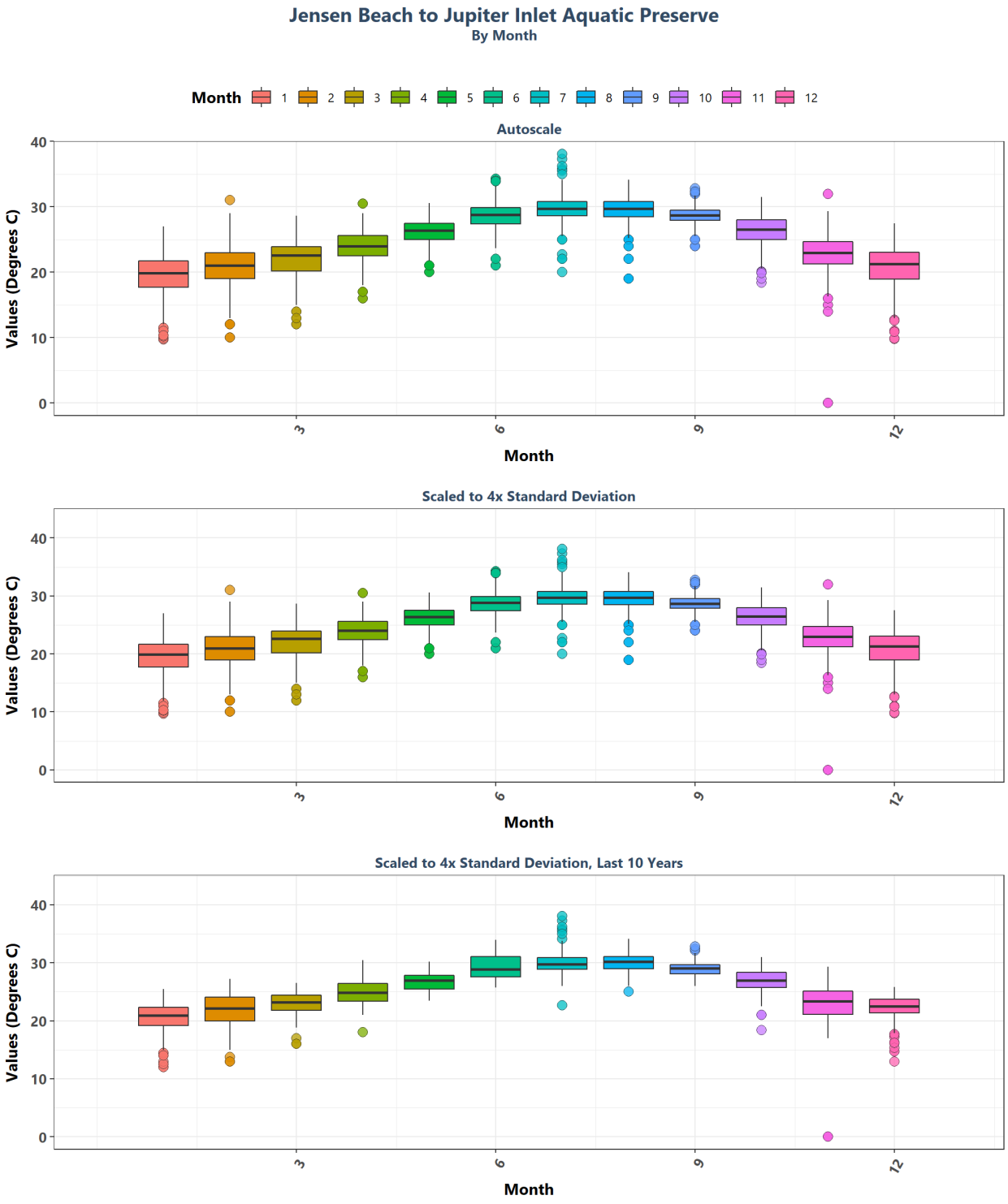
By Year

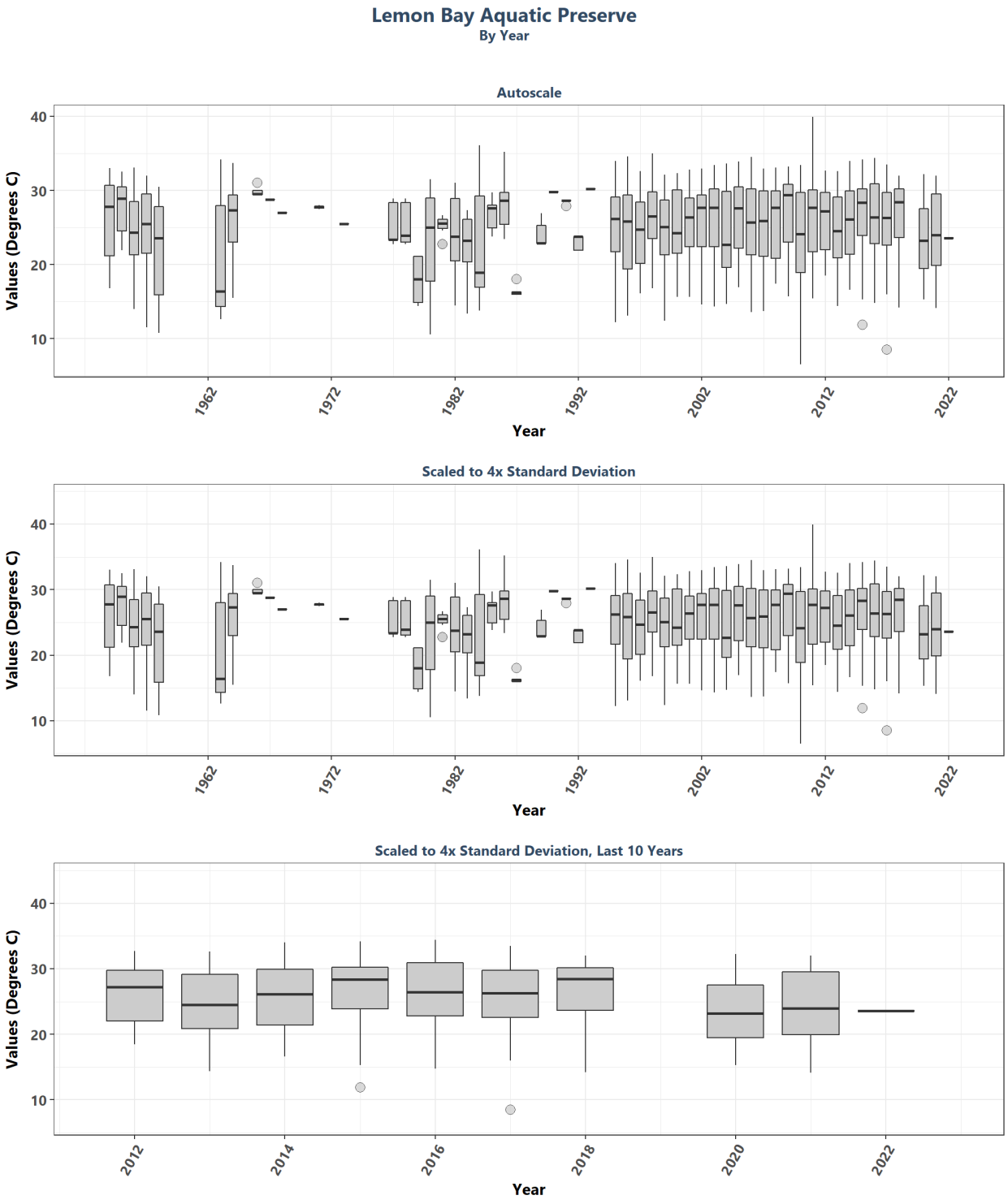


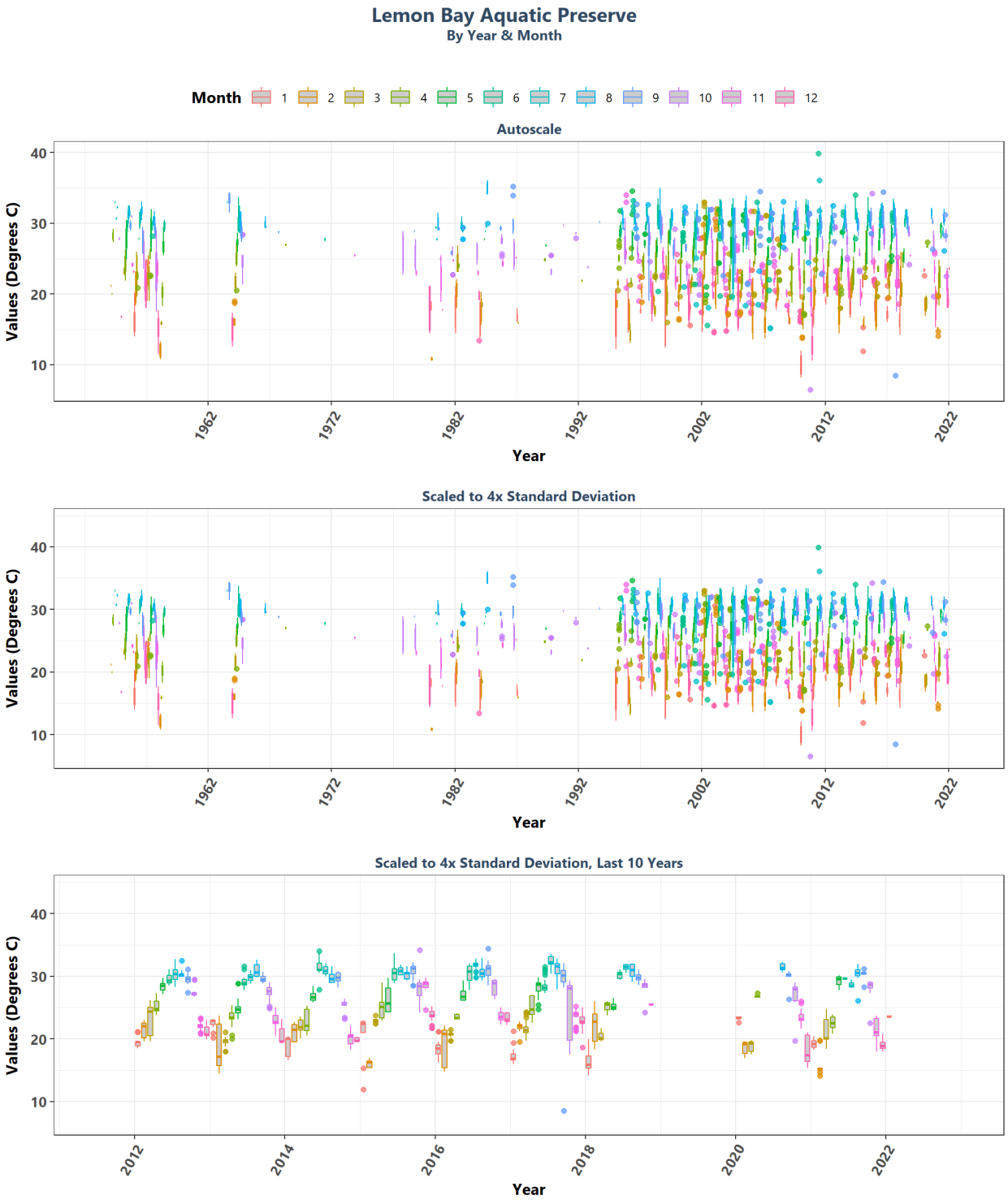


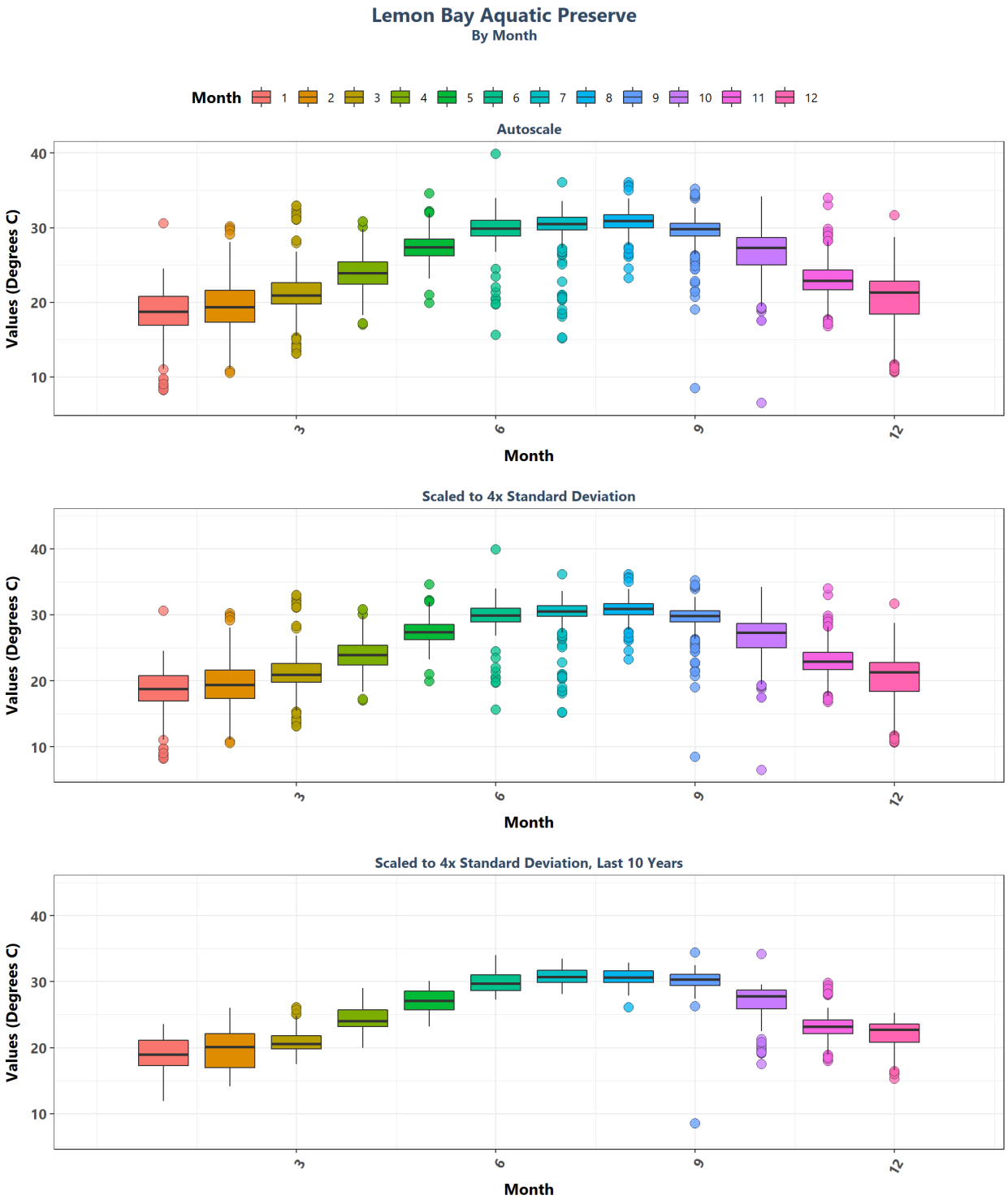


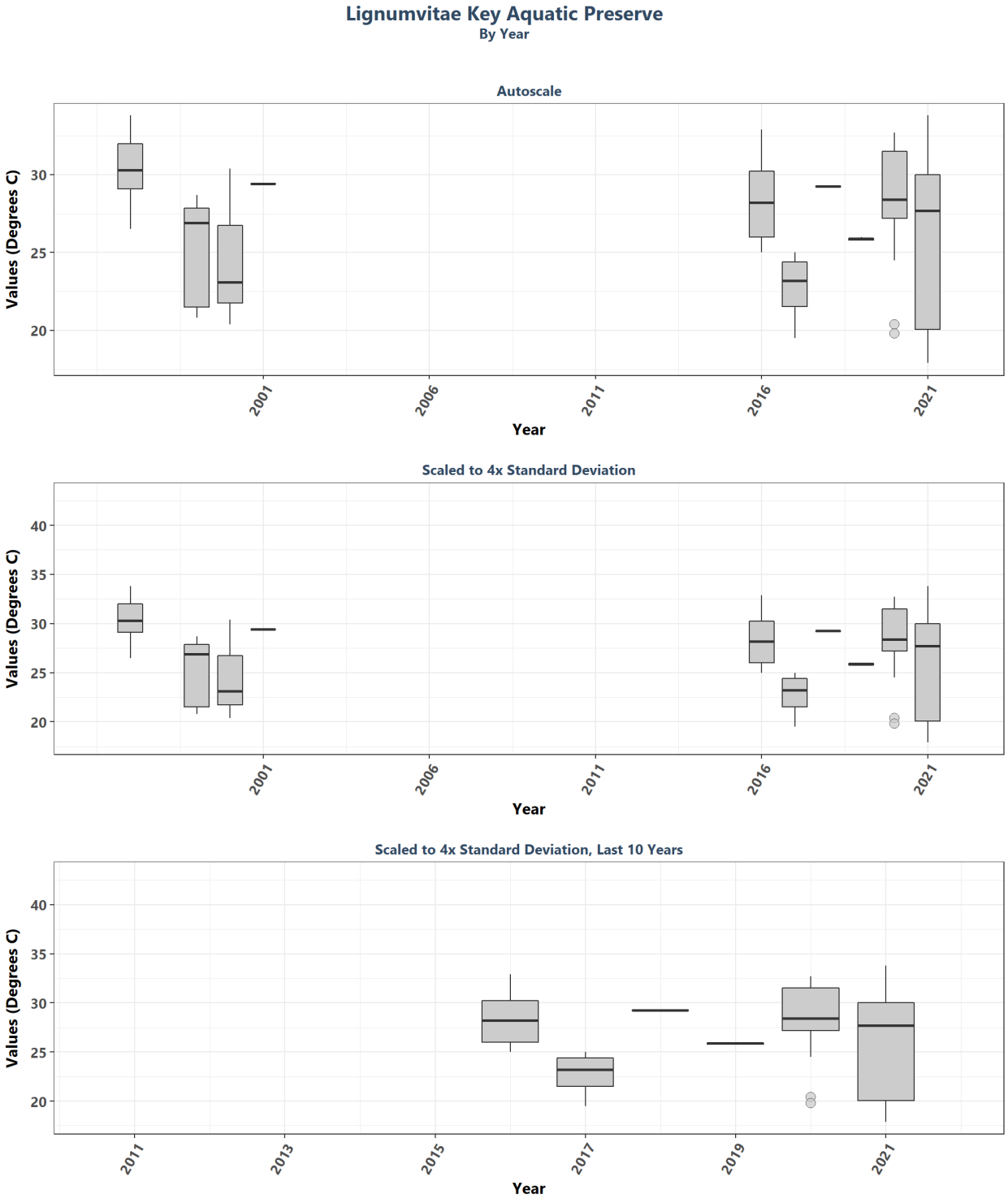

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

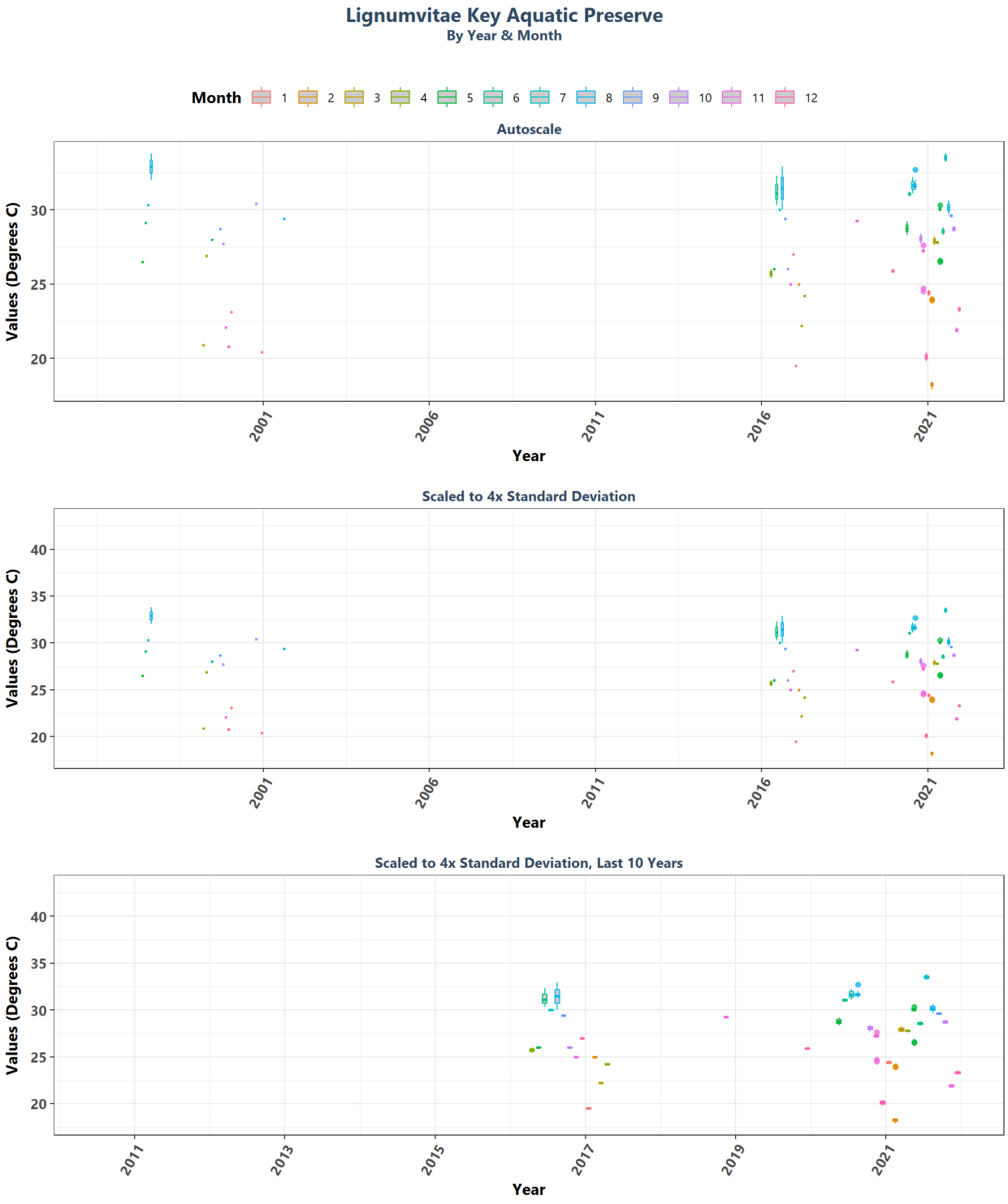

Jensen Beach to Jupiter Inlet Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

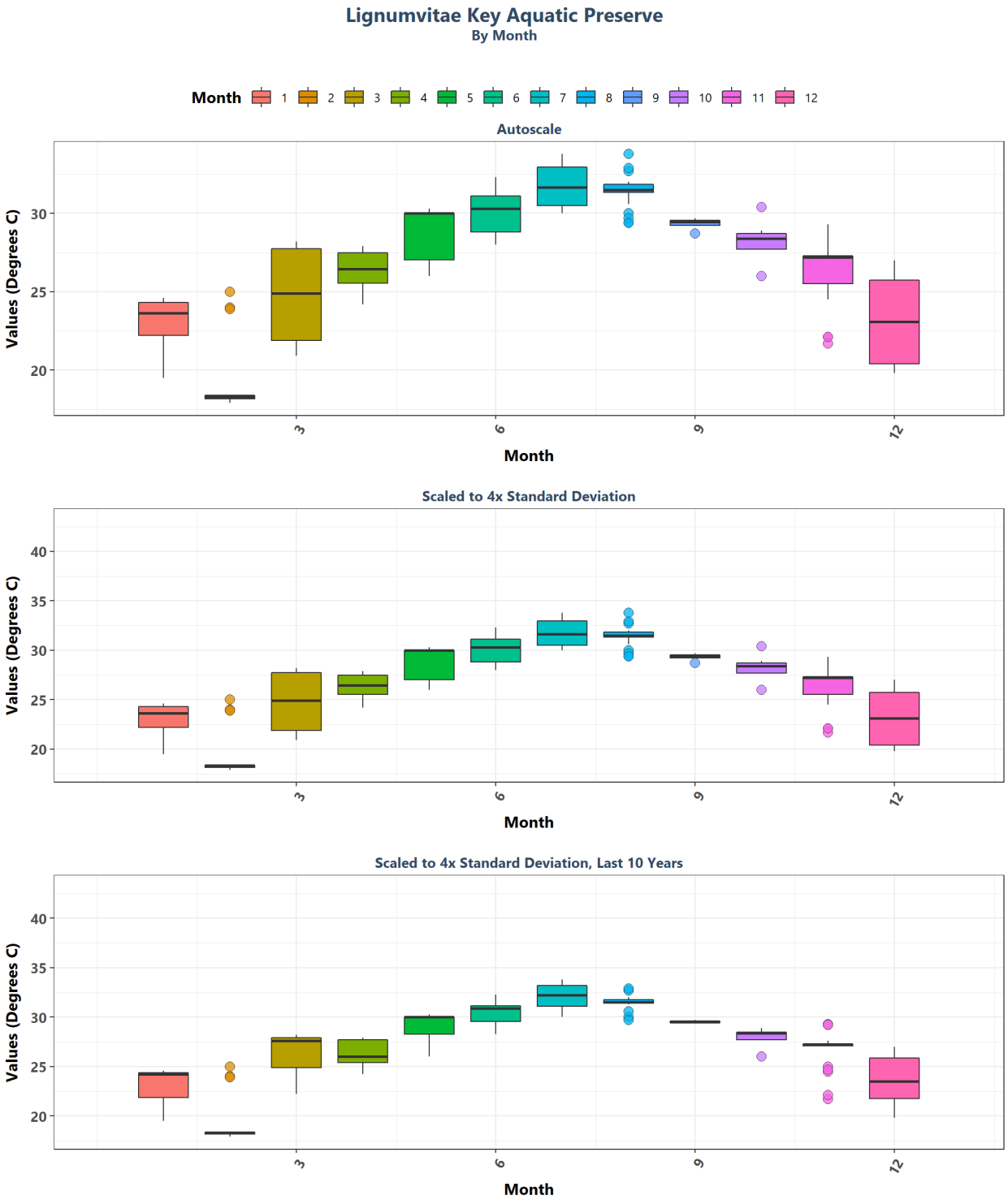

Lemon Bay Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
40
30
20
10
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2012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


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


Lemon Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
40
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

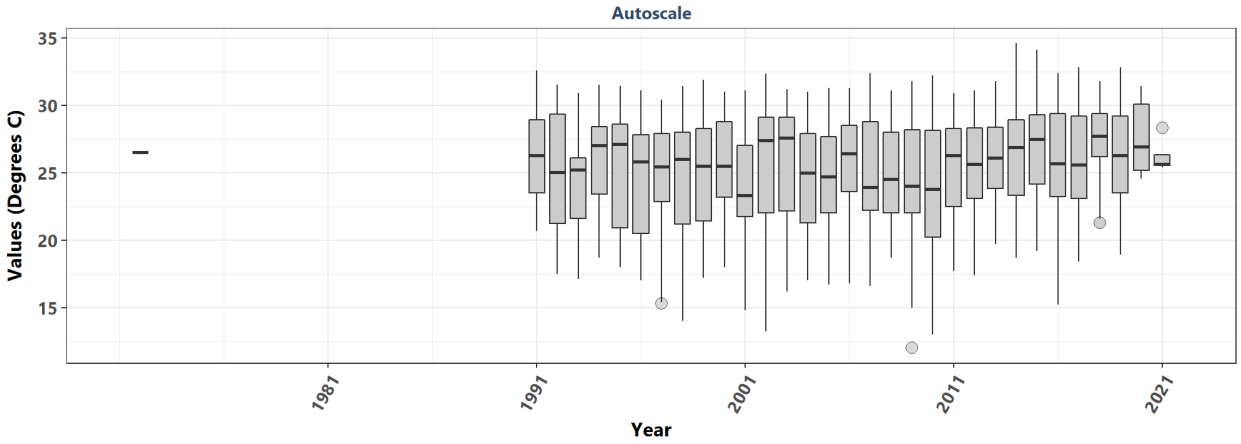

Lignumvitae Key Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
30
25
20
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
35
30
25
20
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
35
30
25
20
2011
2013
2015
2017
2019
2021
Year

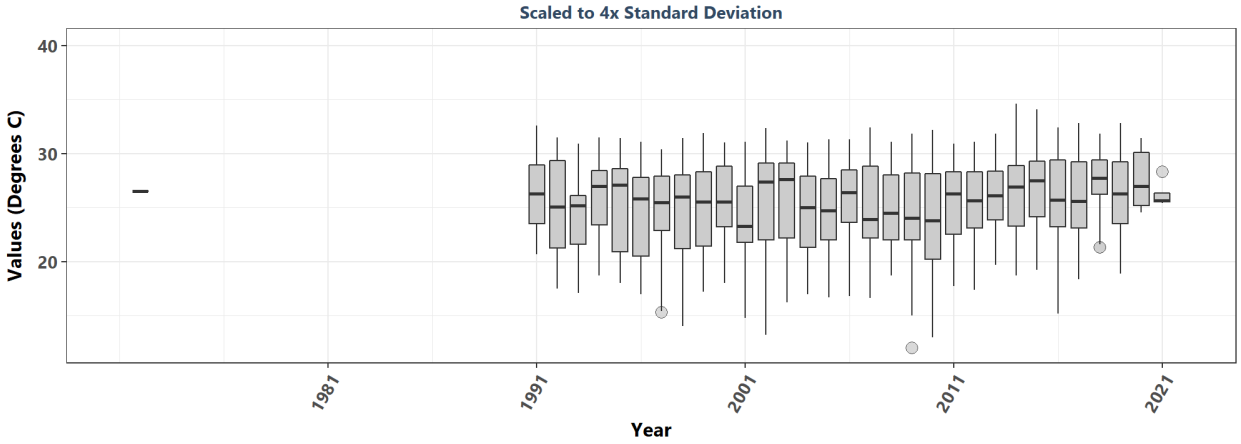

Lignumvitae Key Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
25
20
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
35
30
25
20
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees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

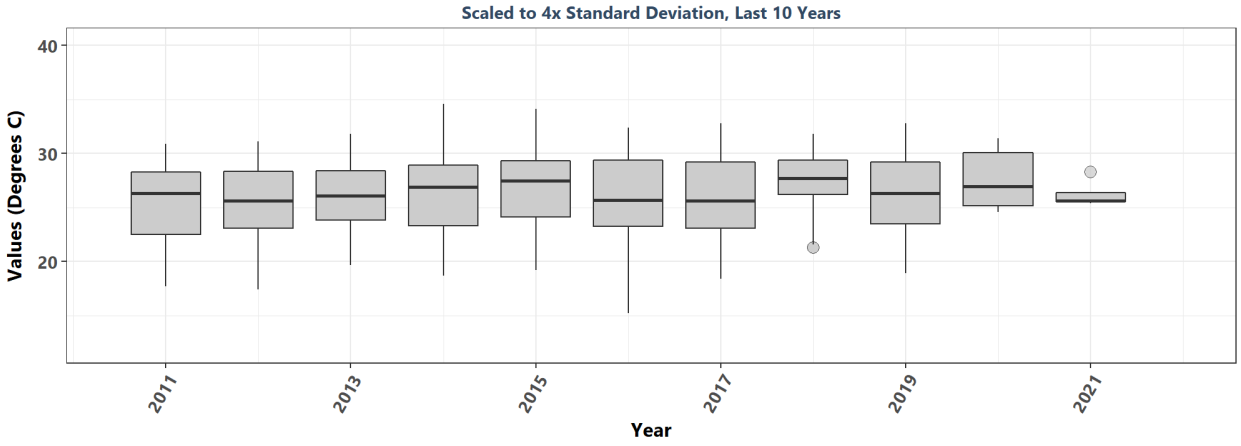

Lignumvitae Key Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
20
25
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
20
25
30
35
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
20
25
30
35
40
3
6
9
12
Month

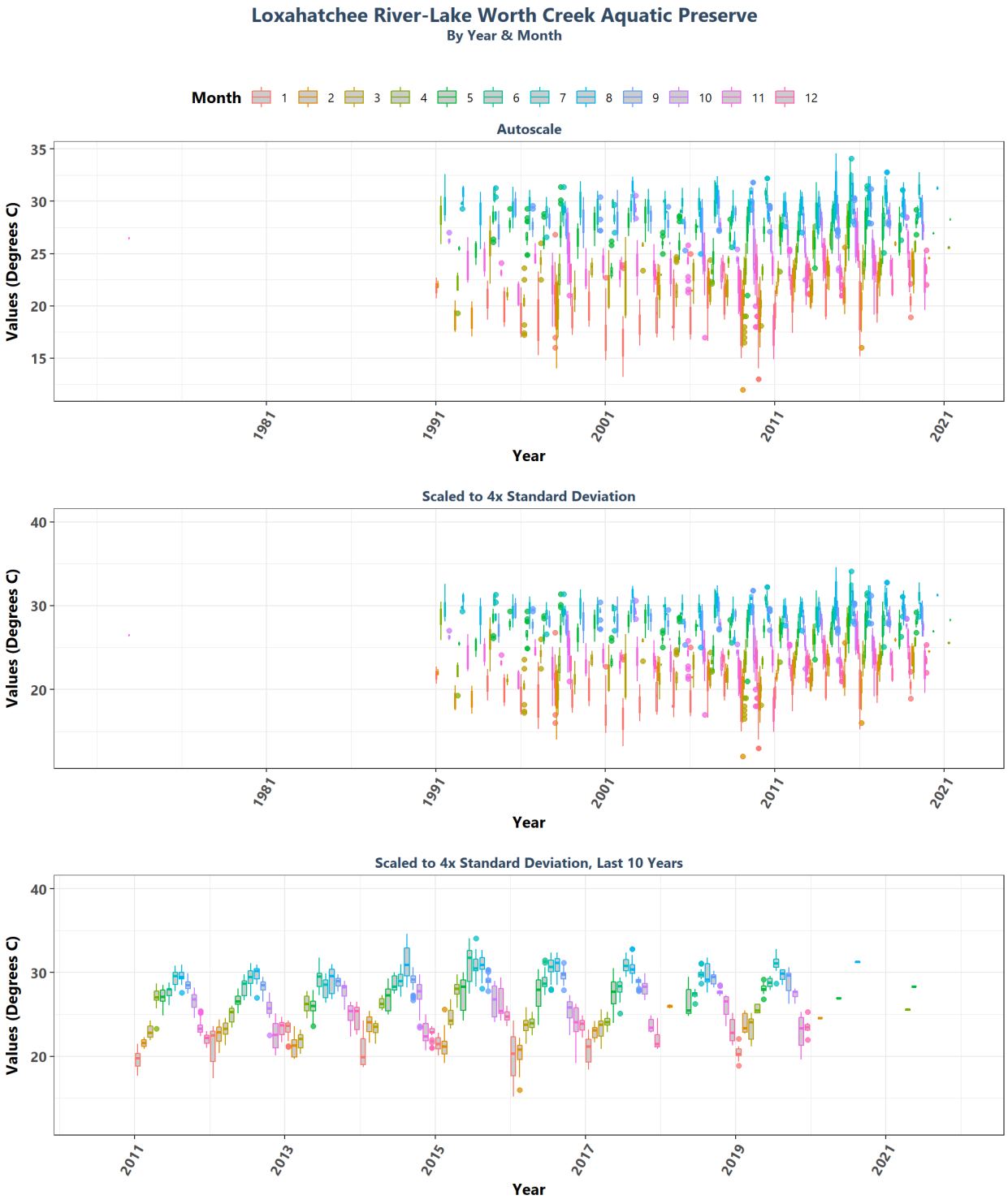
Loxahatchee River-Lake Worth Creek Aquatic Preserve

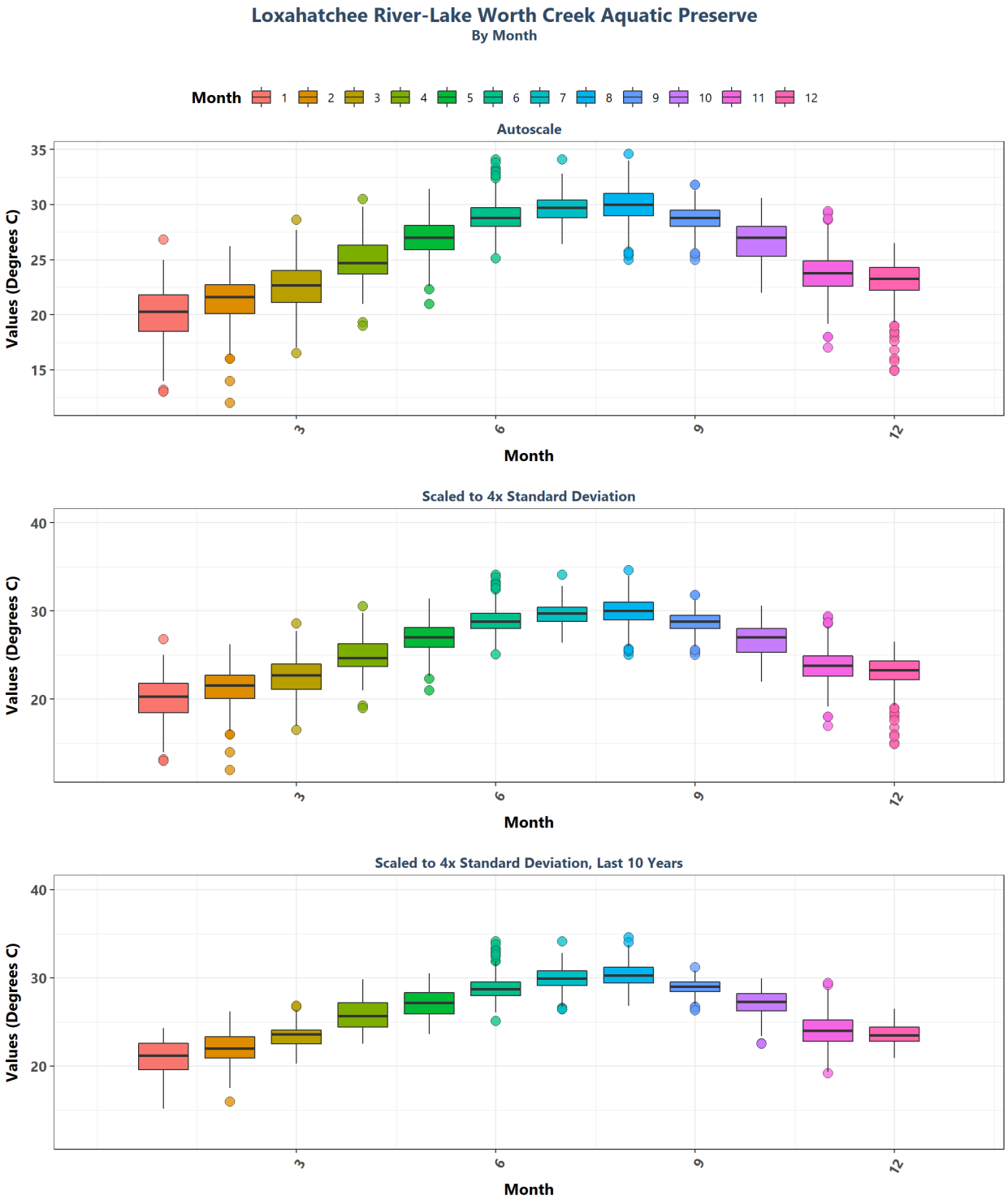
By Year





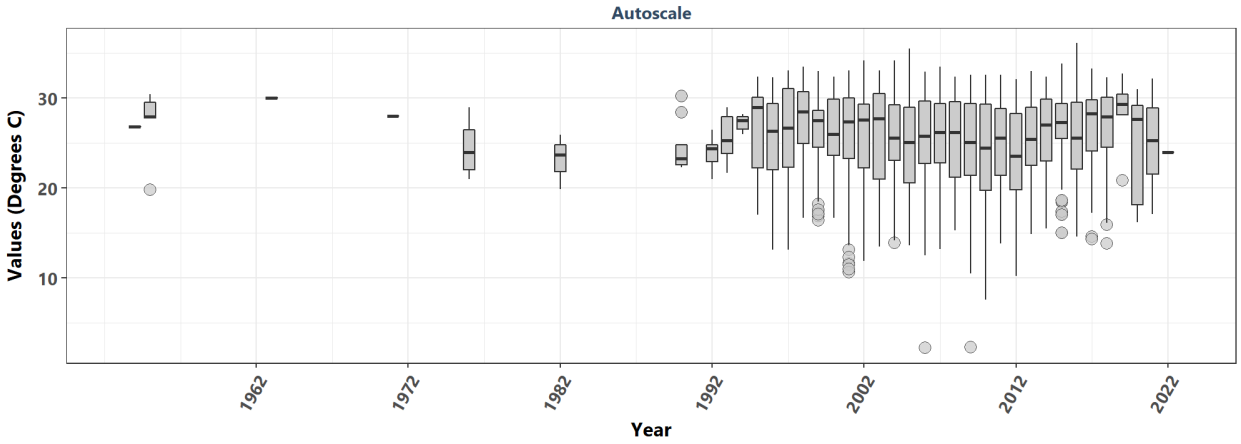


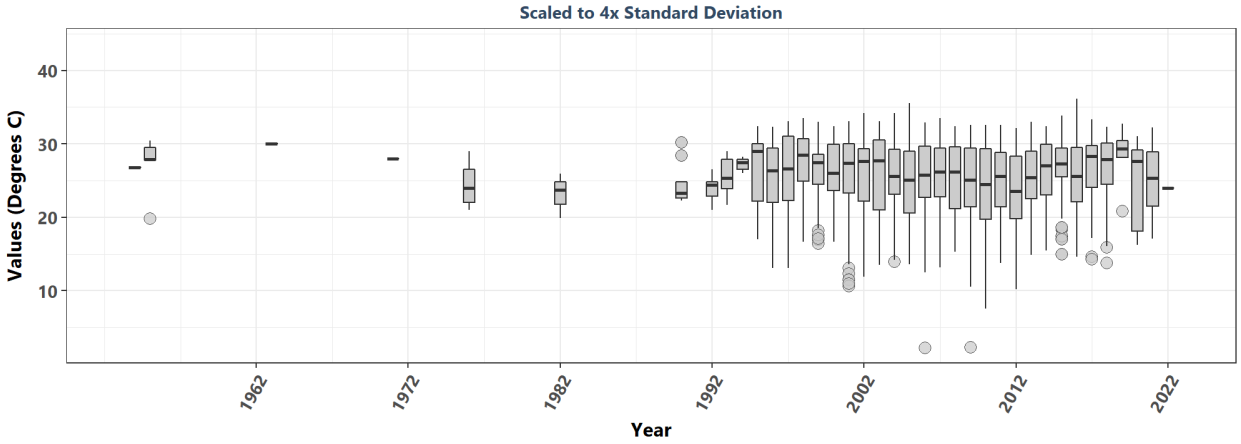

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
2011
2013
2015
2017
2019
2021
Year

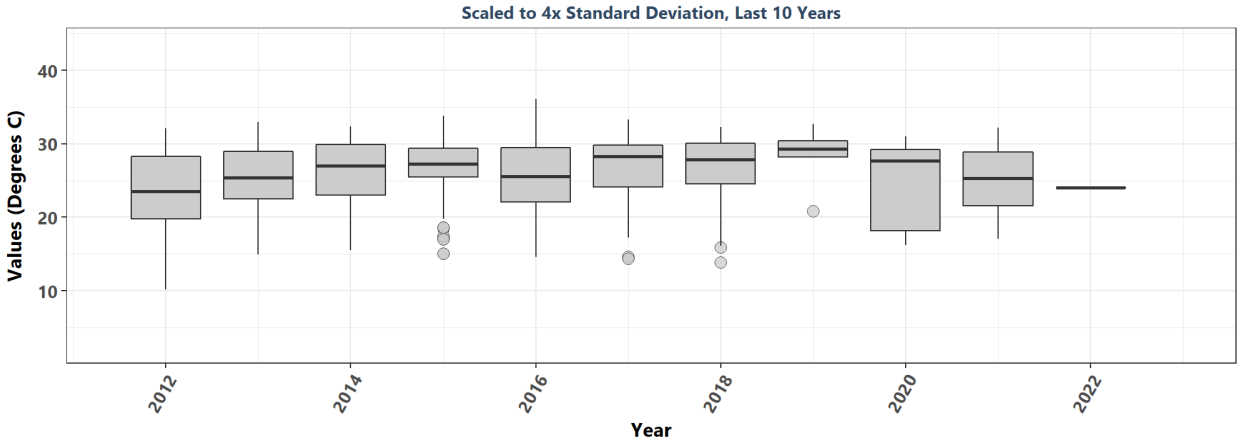

Loxahatchee River-Lake Worth Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20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

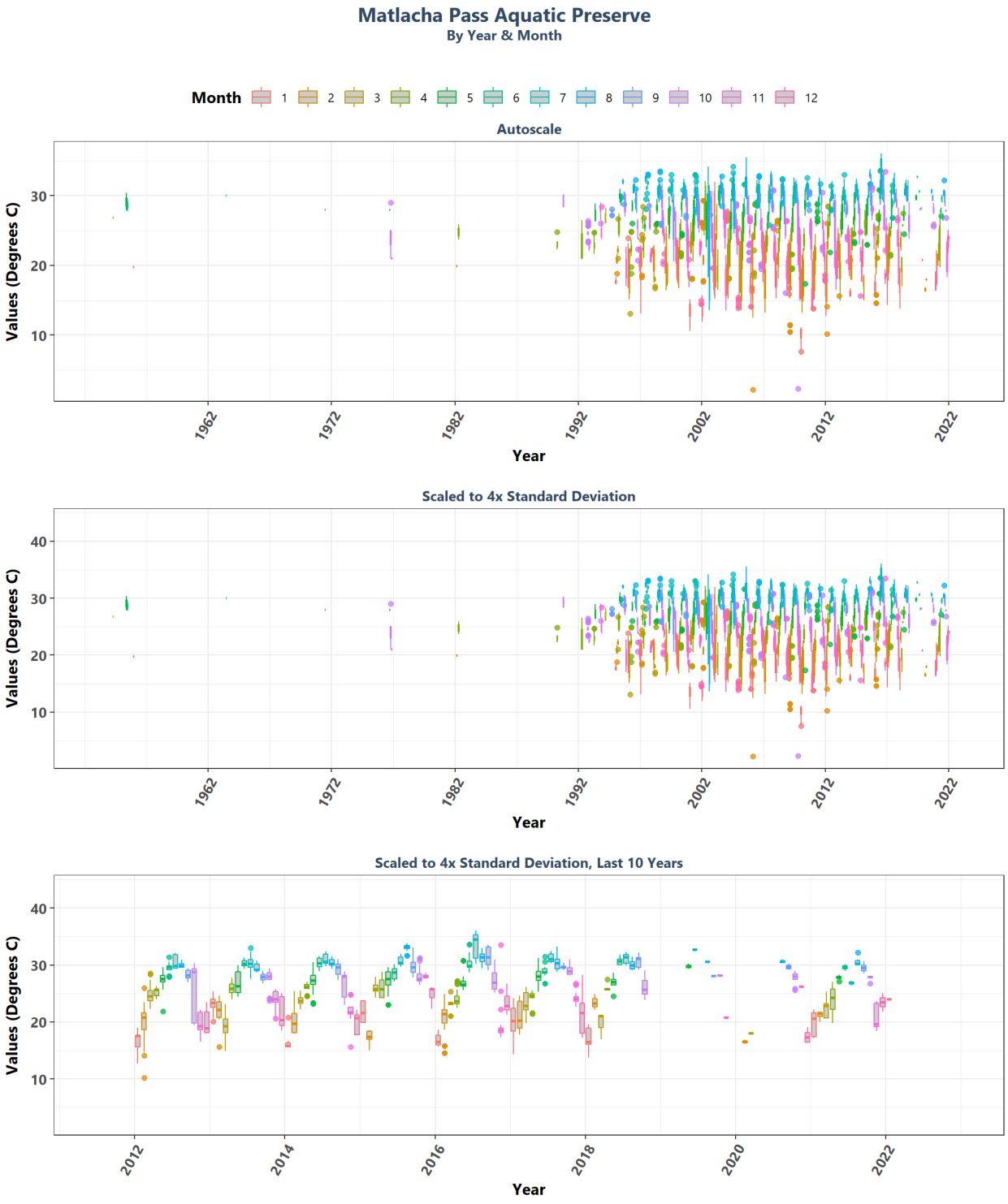
Matlacha Pass Aquatic Preserve

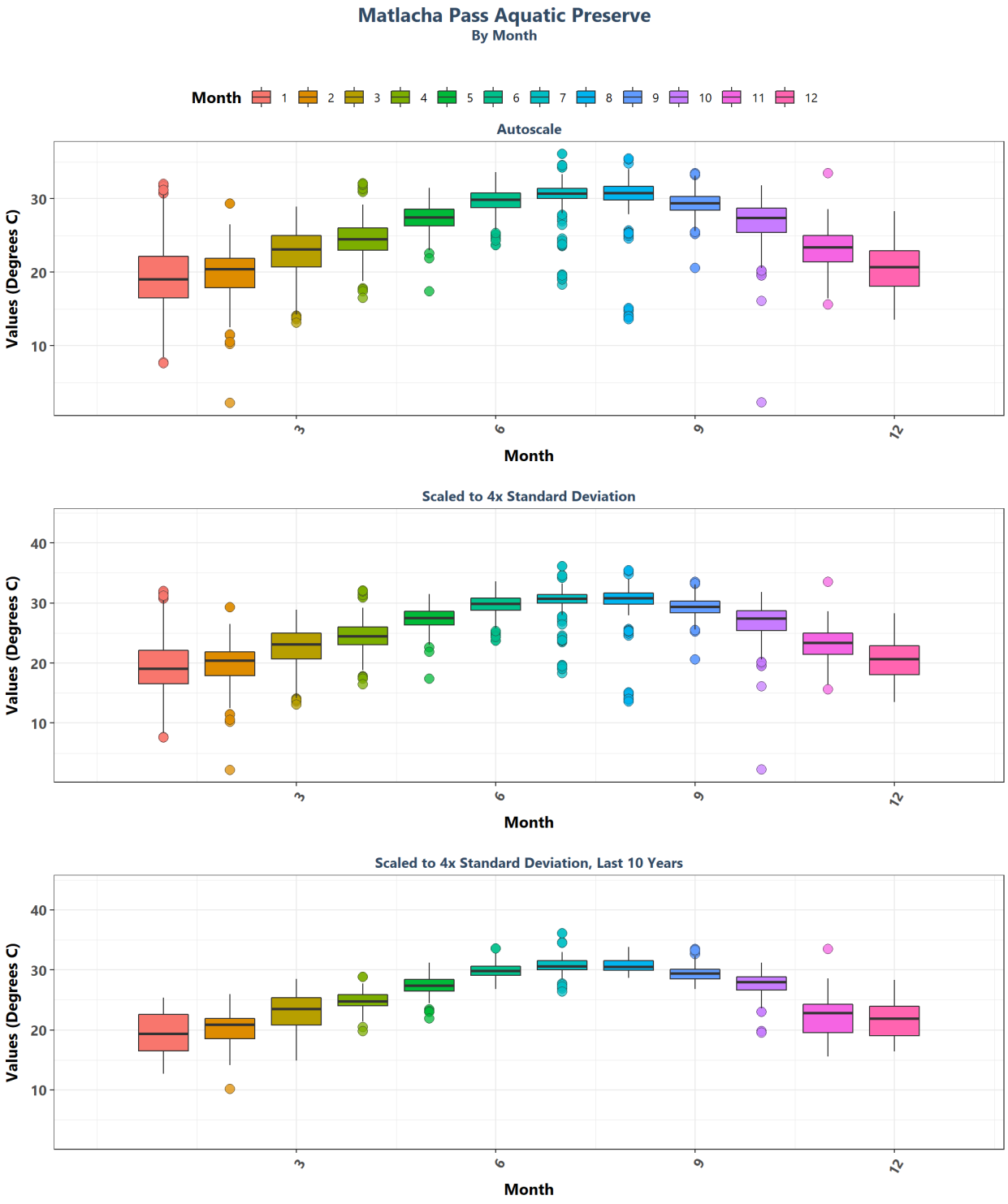
By Year

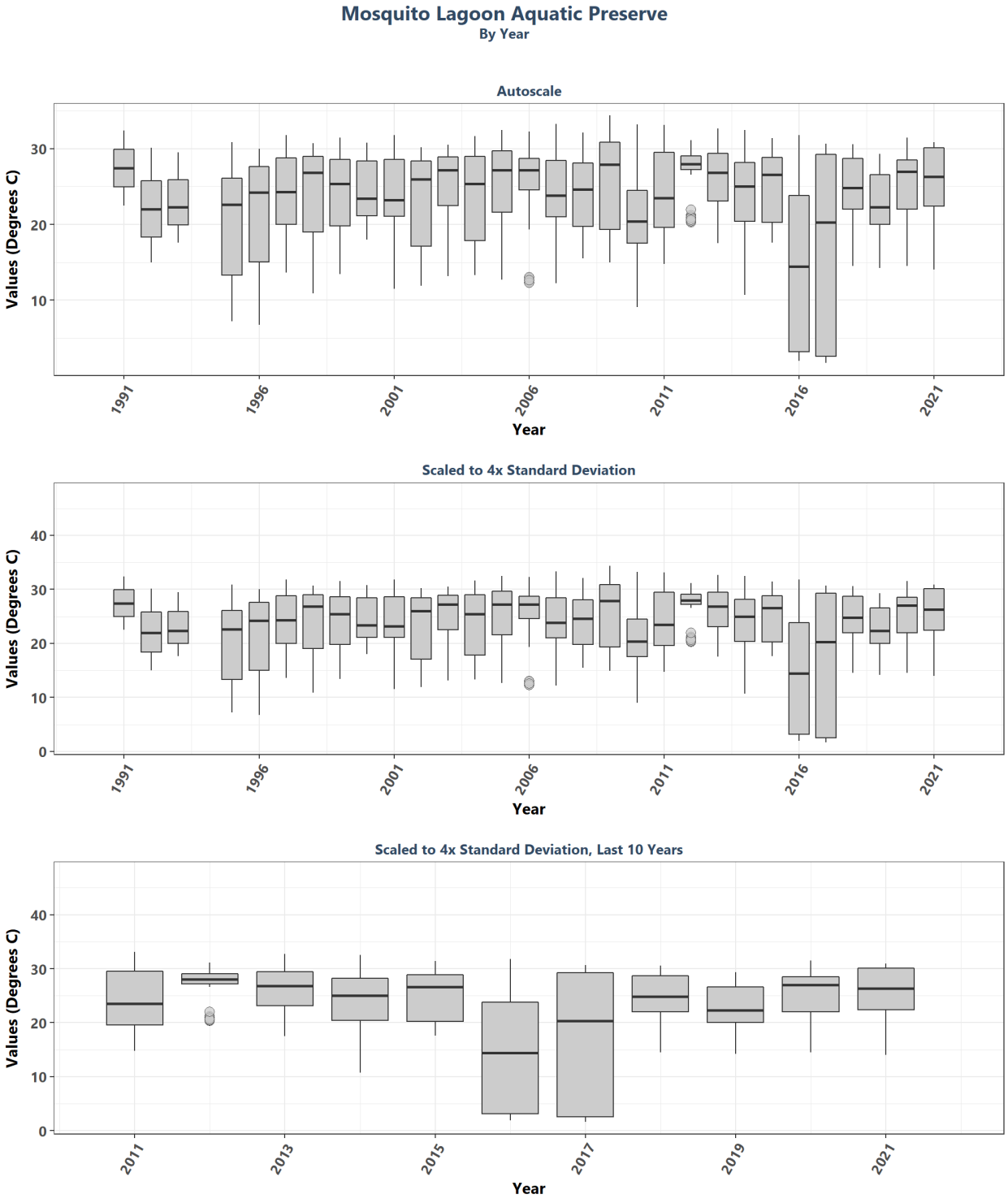


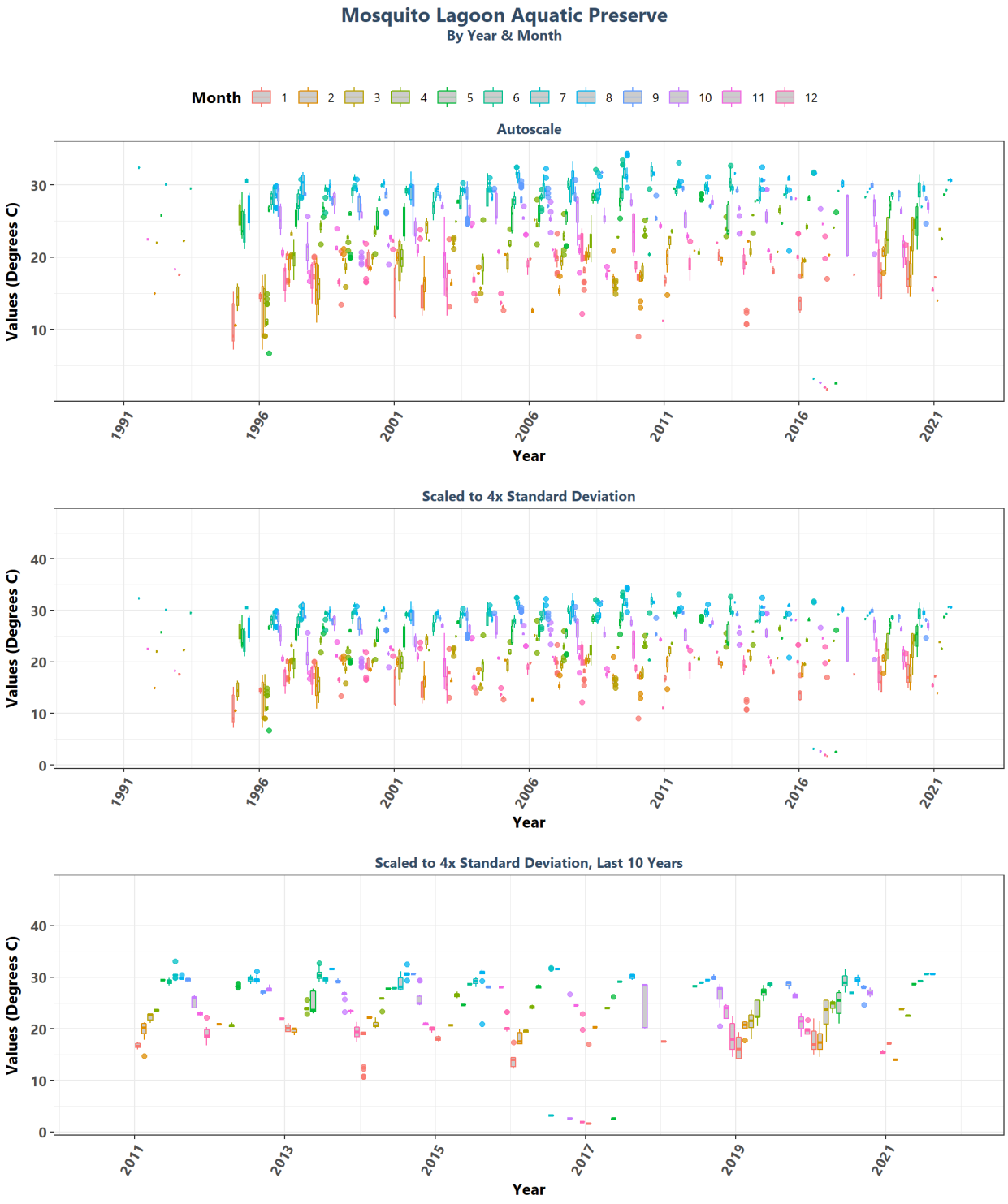


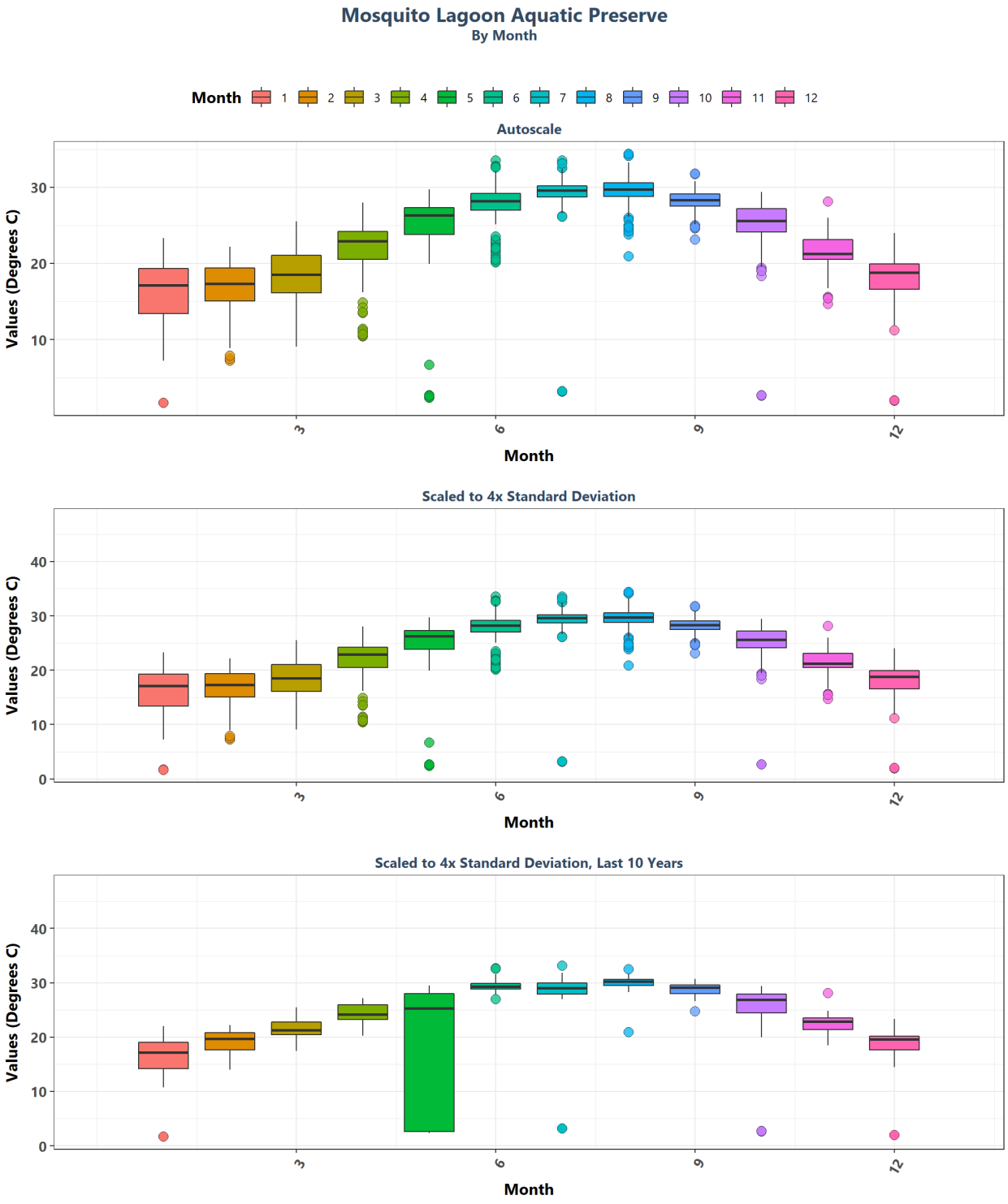


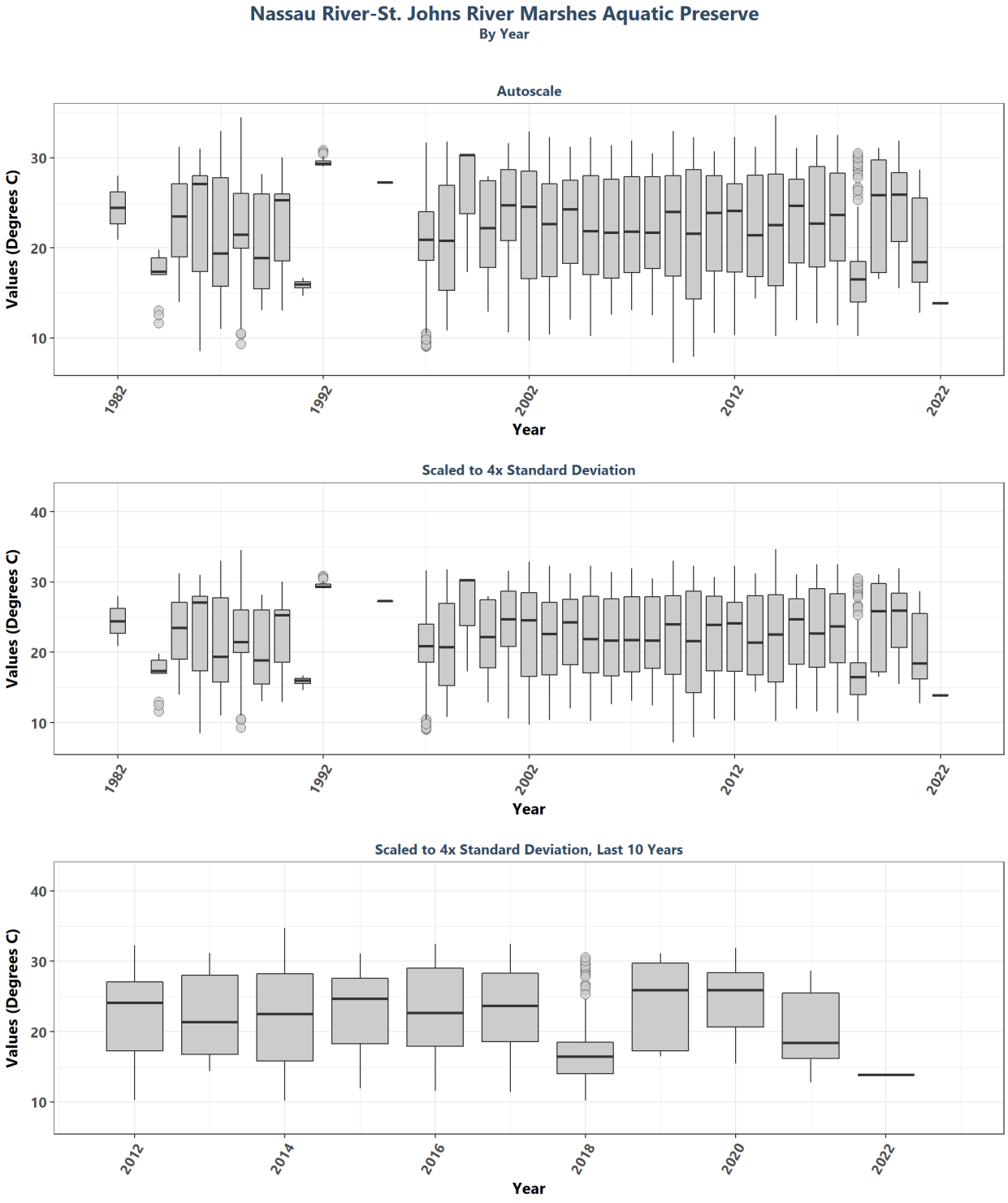

Matlacha Pass Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

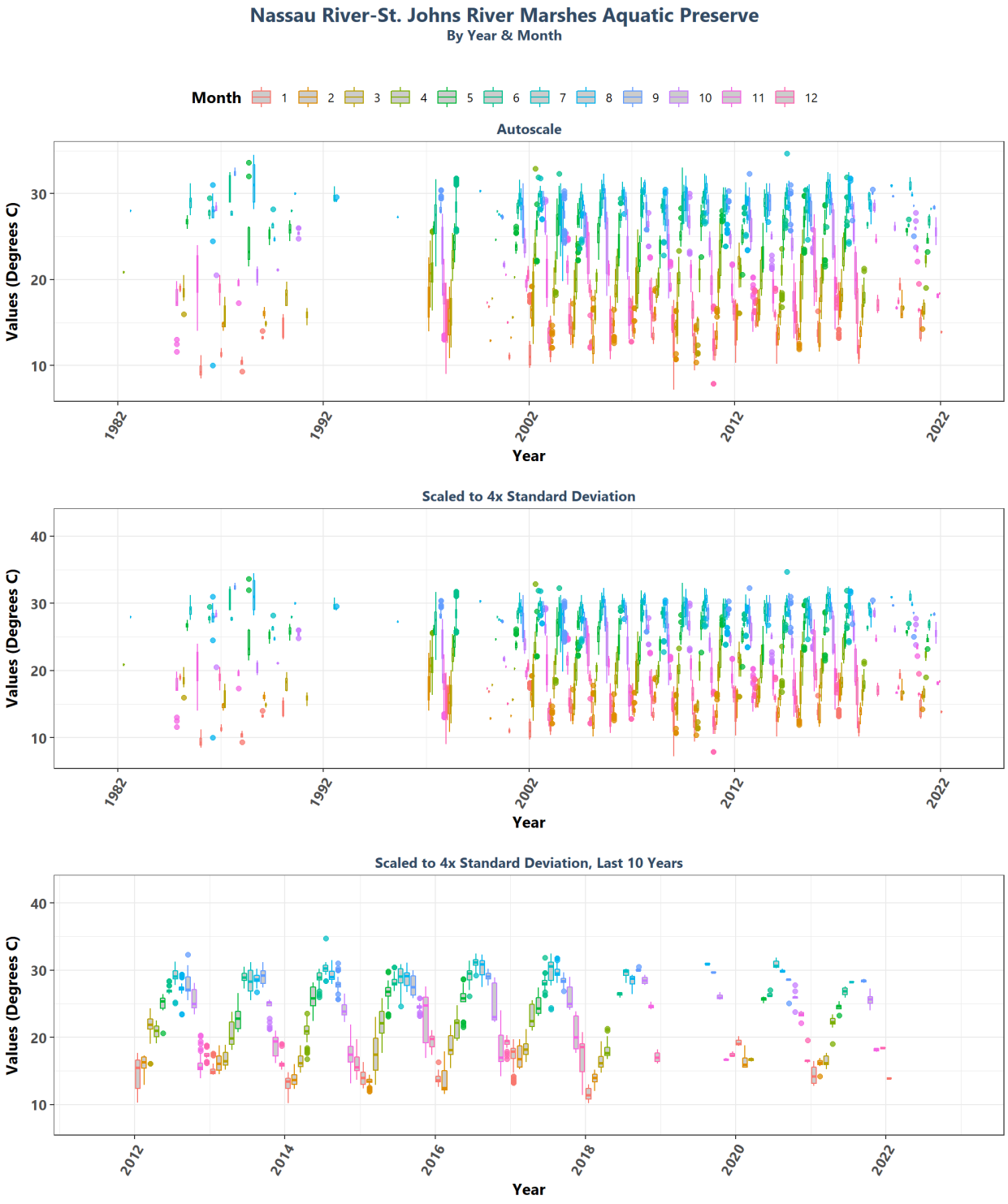

Matlacha Pass Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

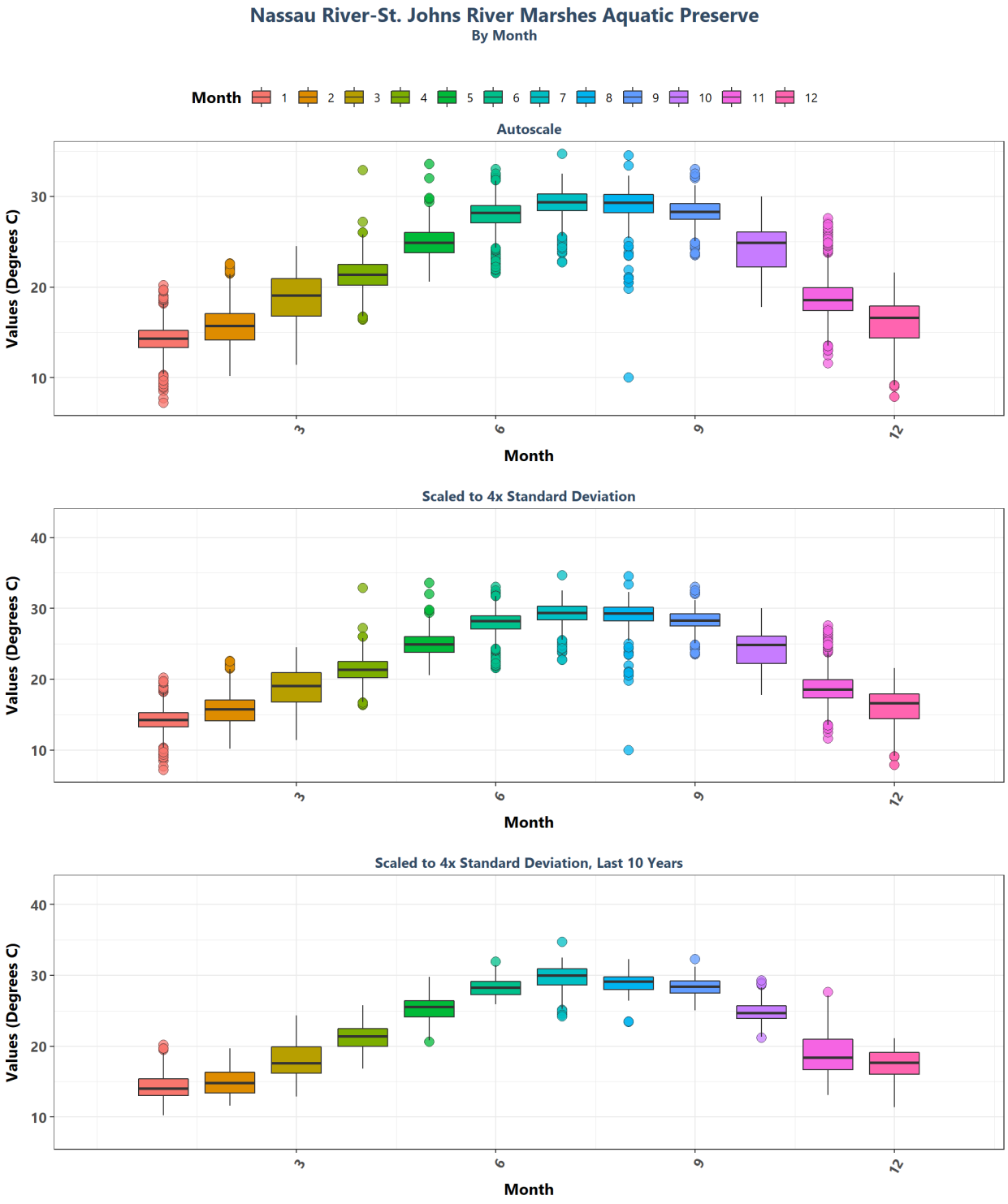

Mosquito Lagoon Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
30
20
10
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
0
1991
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
0
2011
2013
2015
2017
2019
2021
Year


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


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

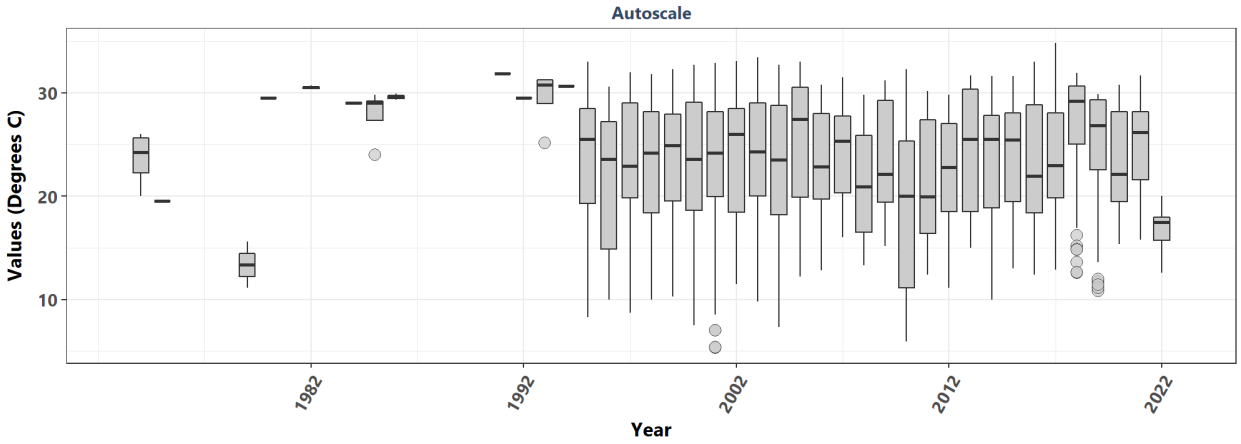

Nassau River-St. Johns River Marshes Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
10
20
30
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

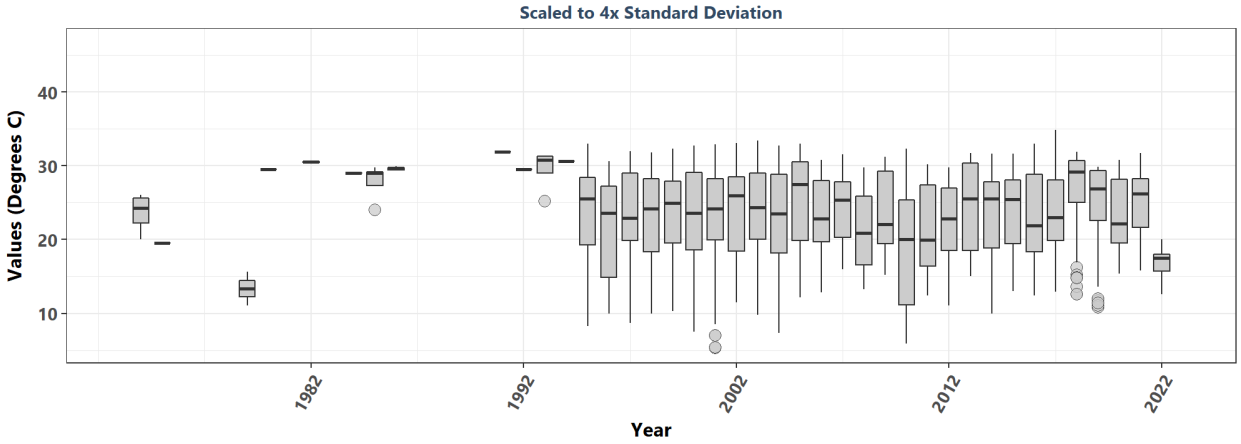

Nassau River-St. Johns River Marshes Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

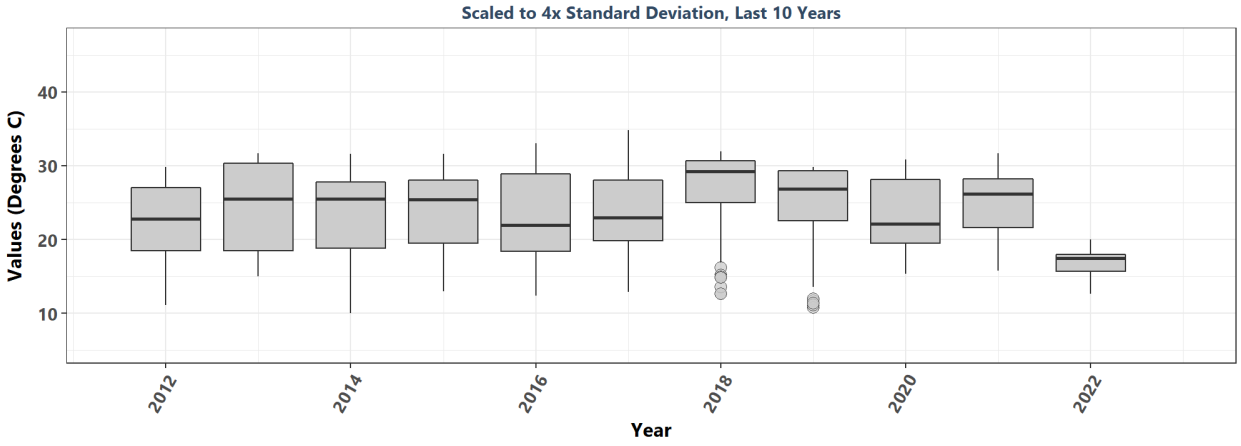

Nassau River-St. Johns River Marshes Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

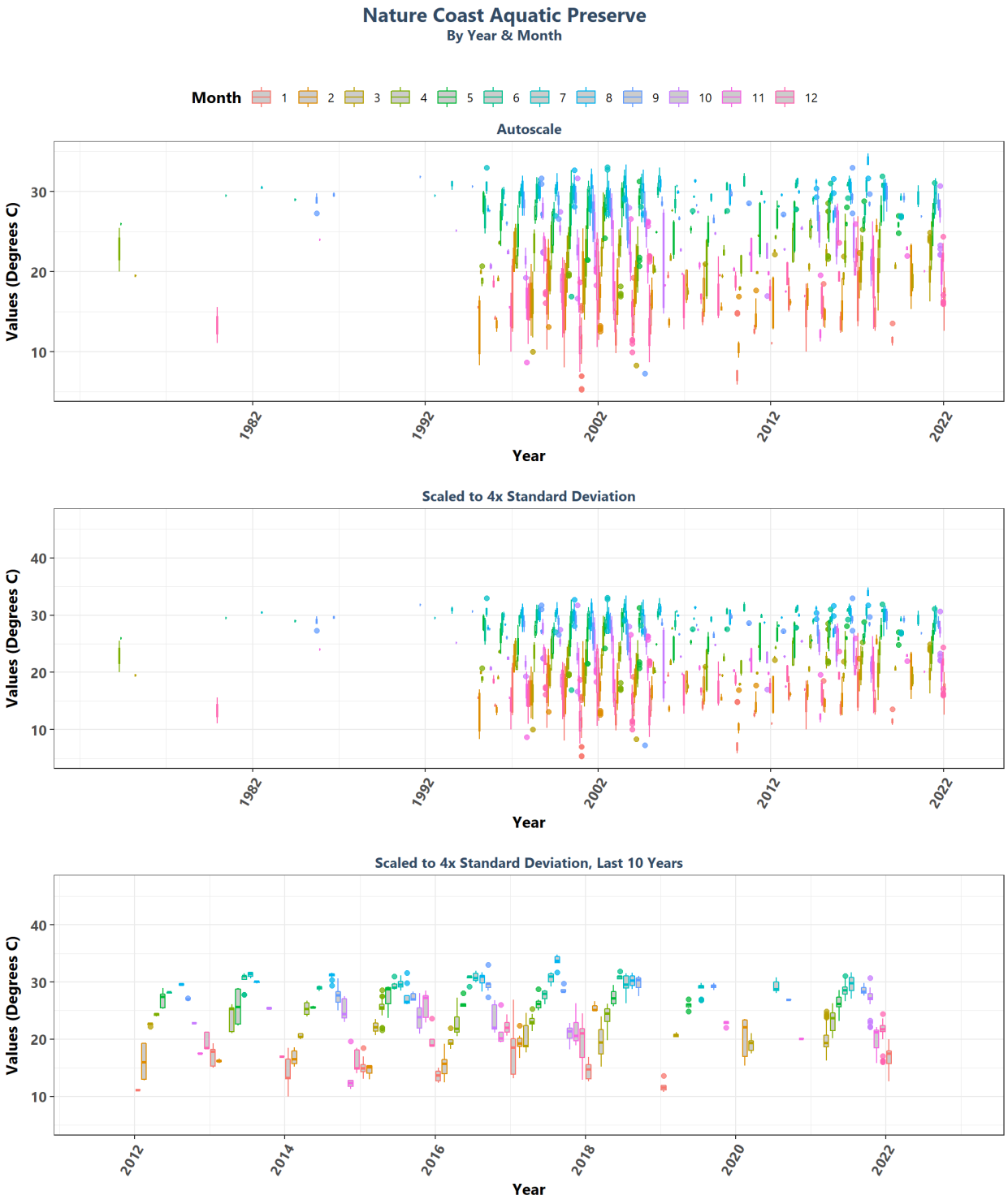
Nature Coast Aquatic Preserve

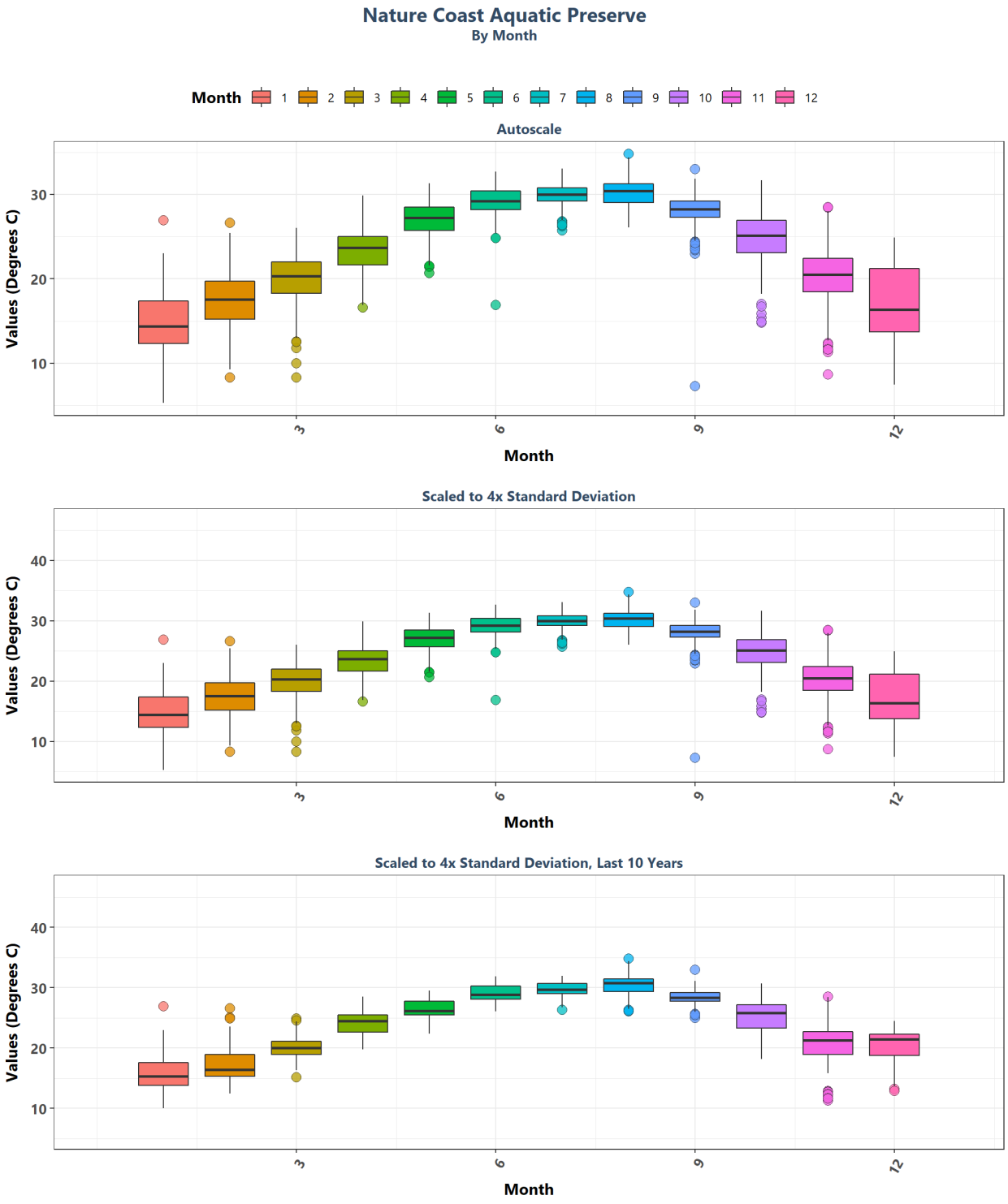
By Year





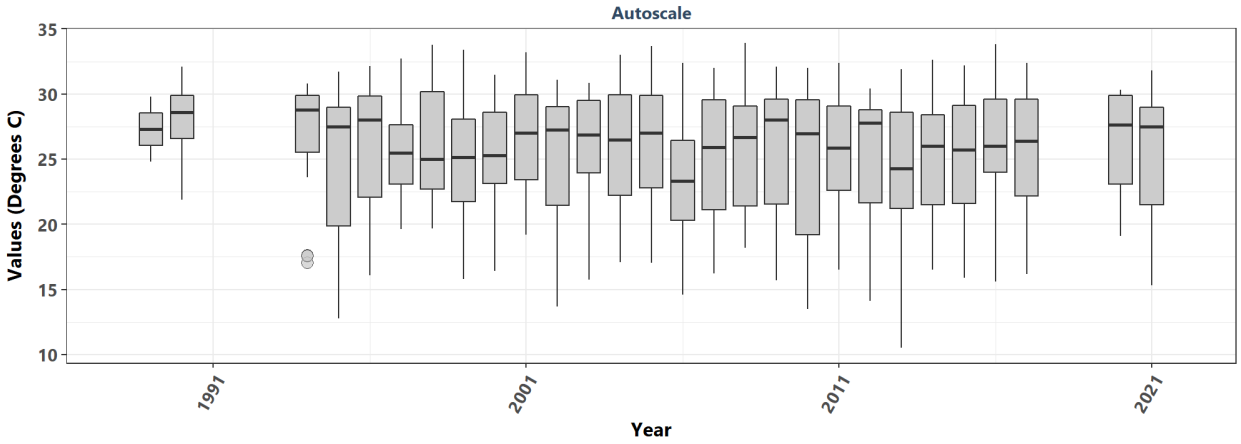


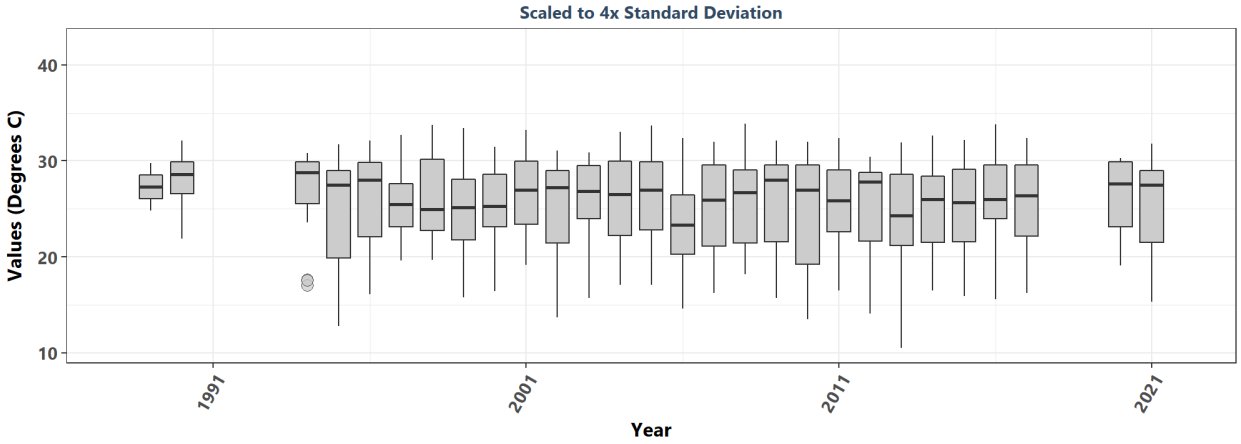

Nature Coast Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

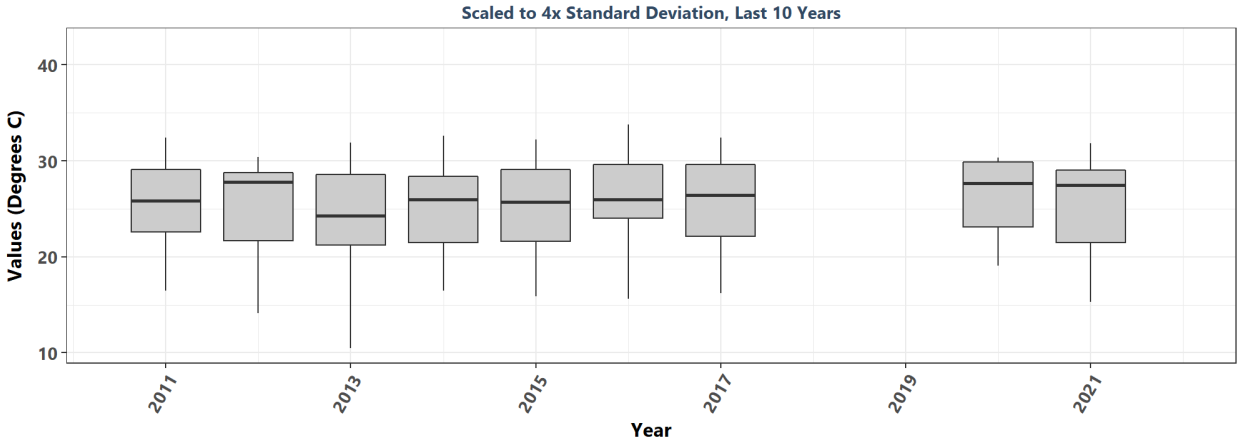

Nature Coast Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

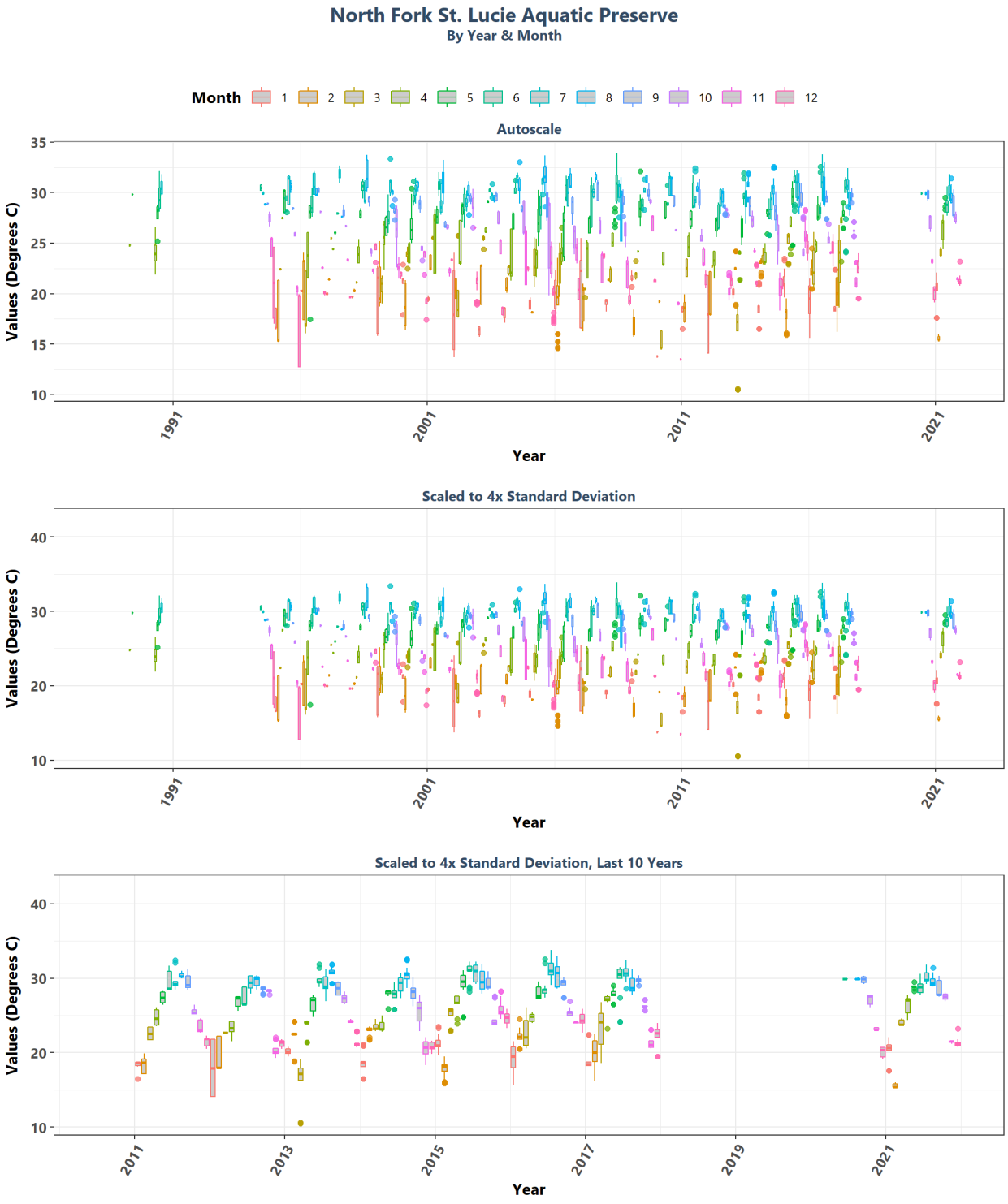
North Fork St. Lucie Aquatic Preserve

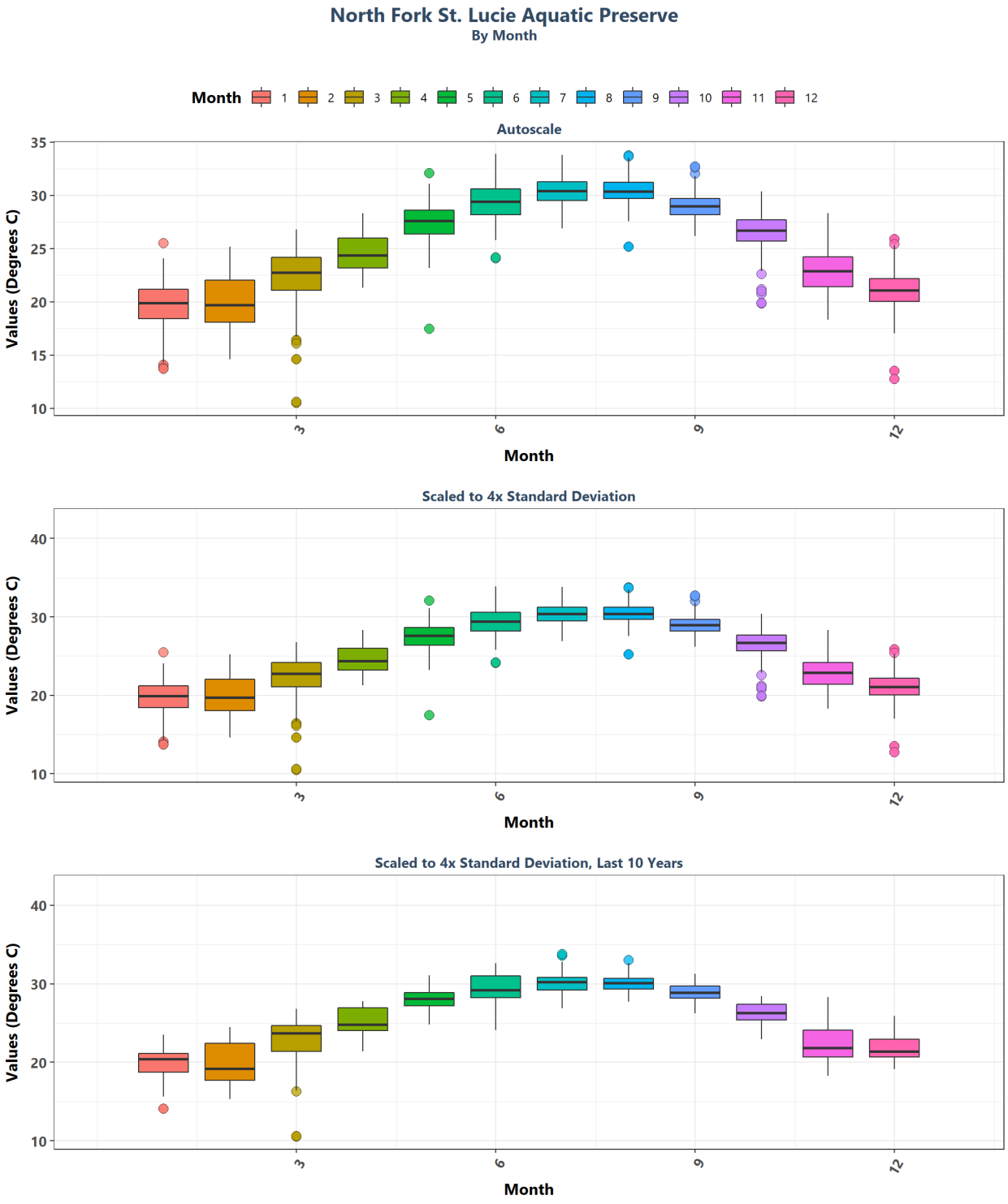
By Year





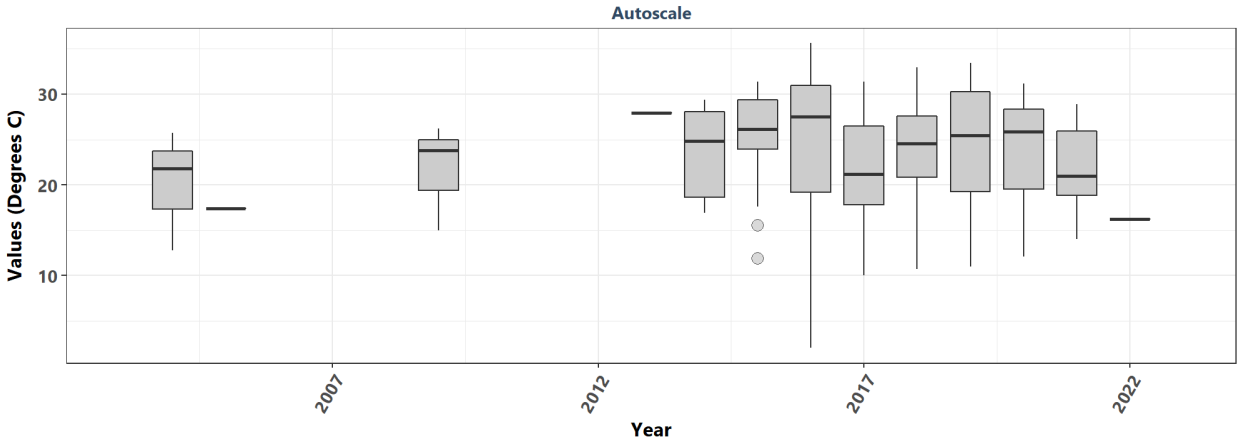


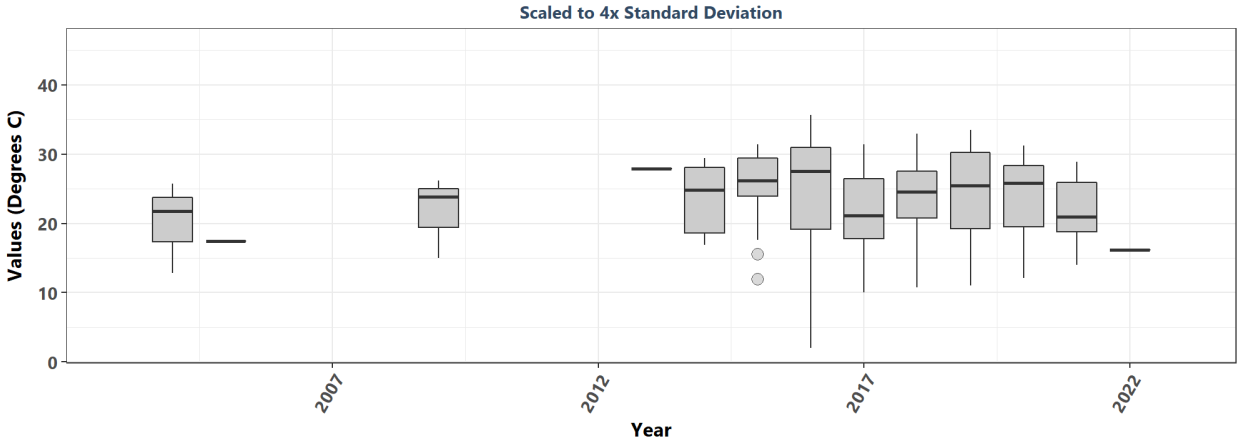

North Fork St. Lucie Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

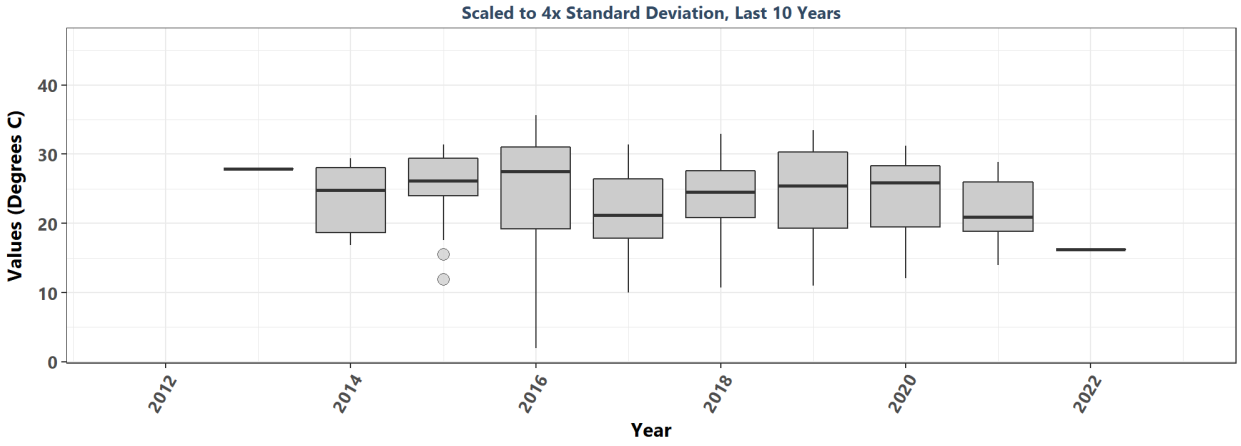

North Fork St. Lucie Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

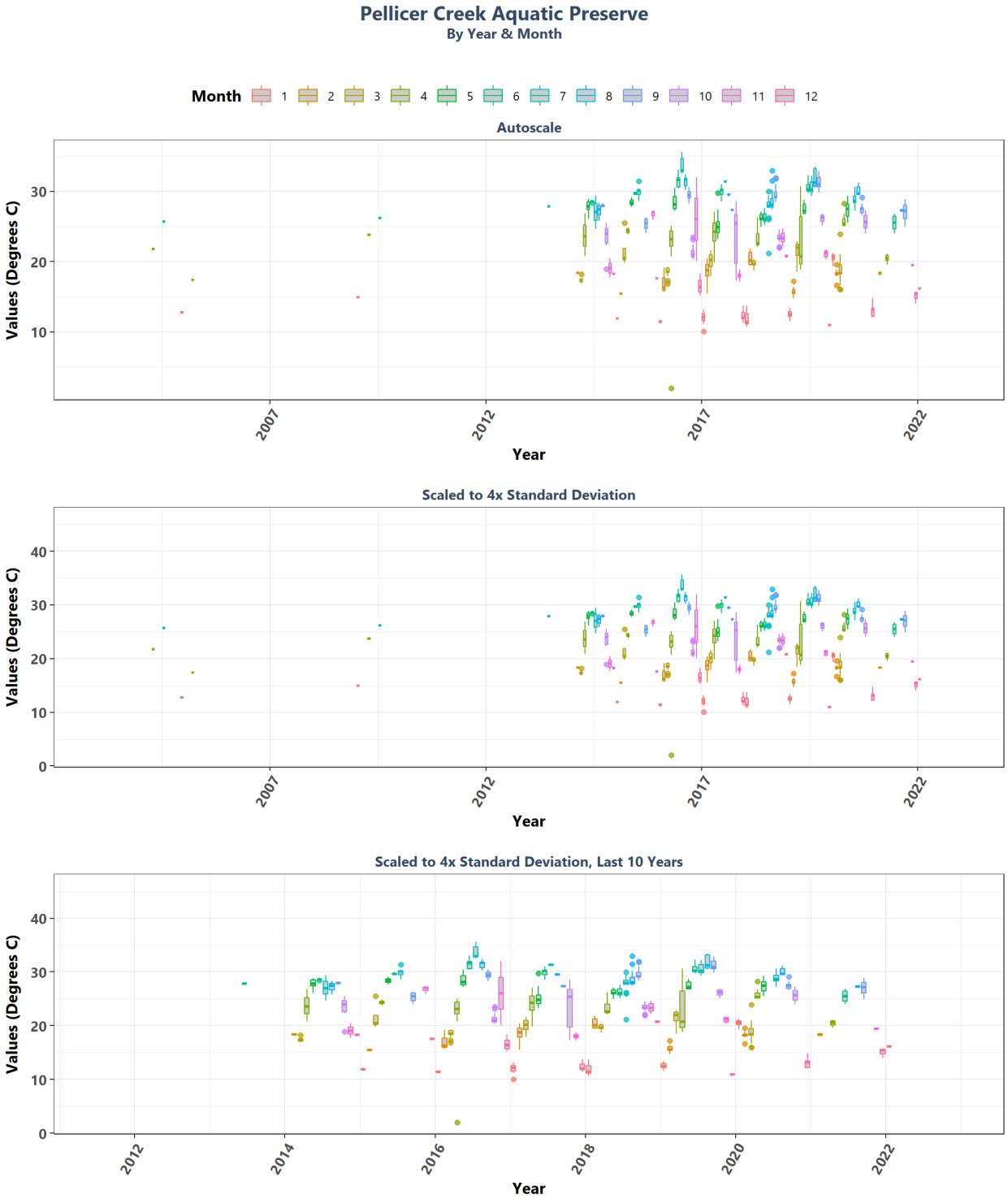
Pellicer Creek Aquatic Preserve

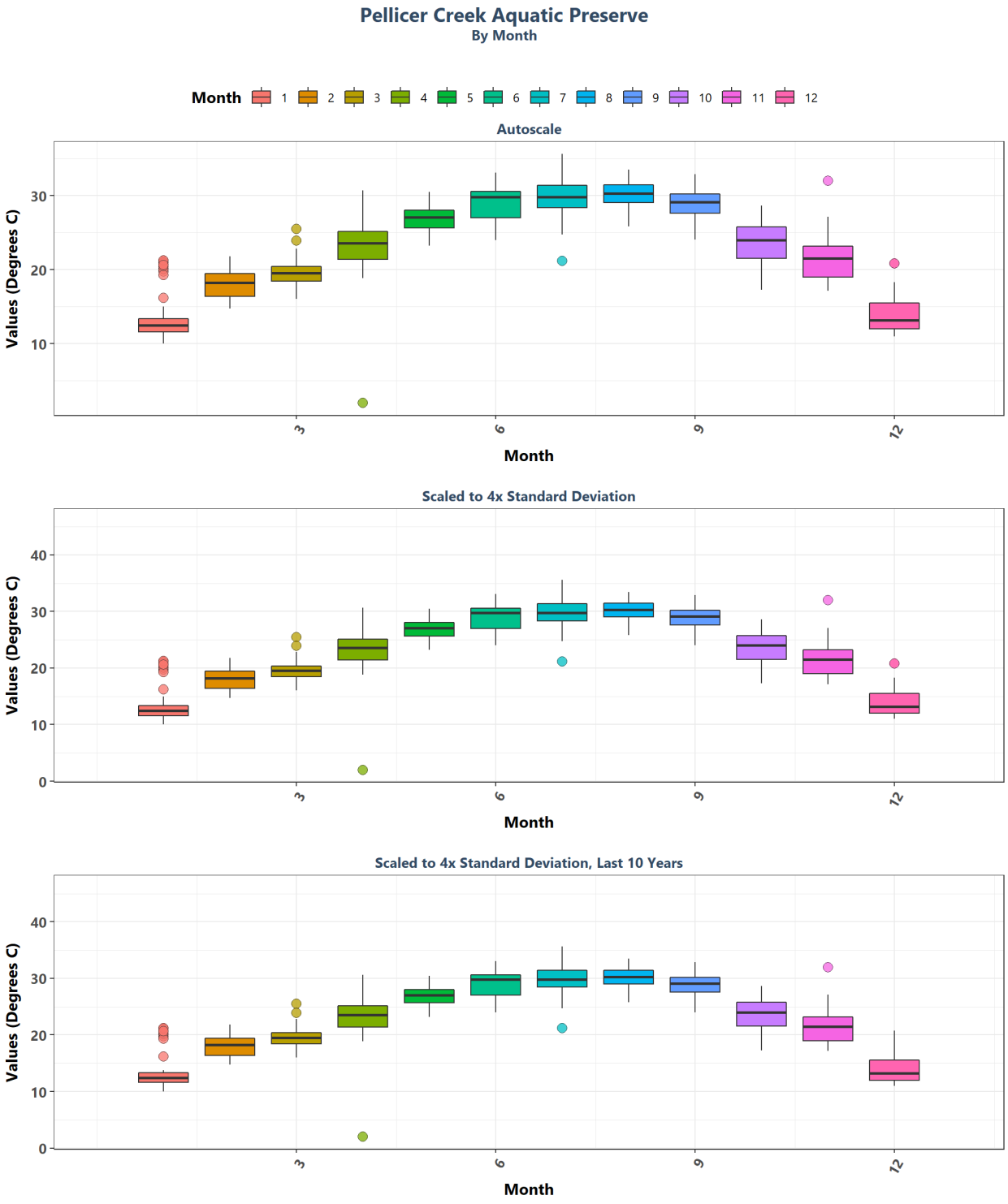
By Year

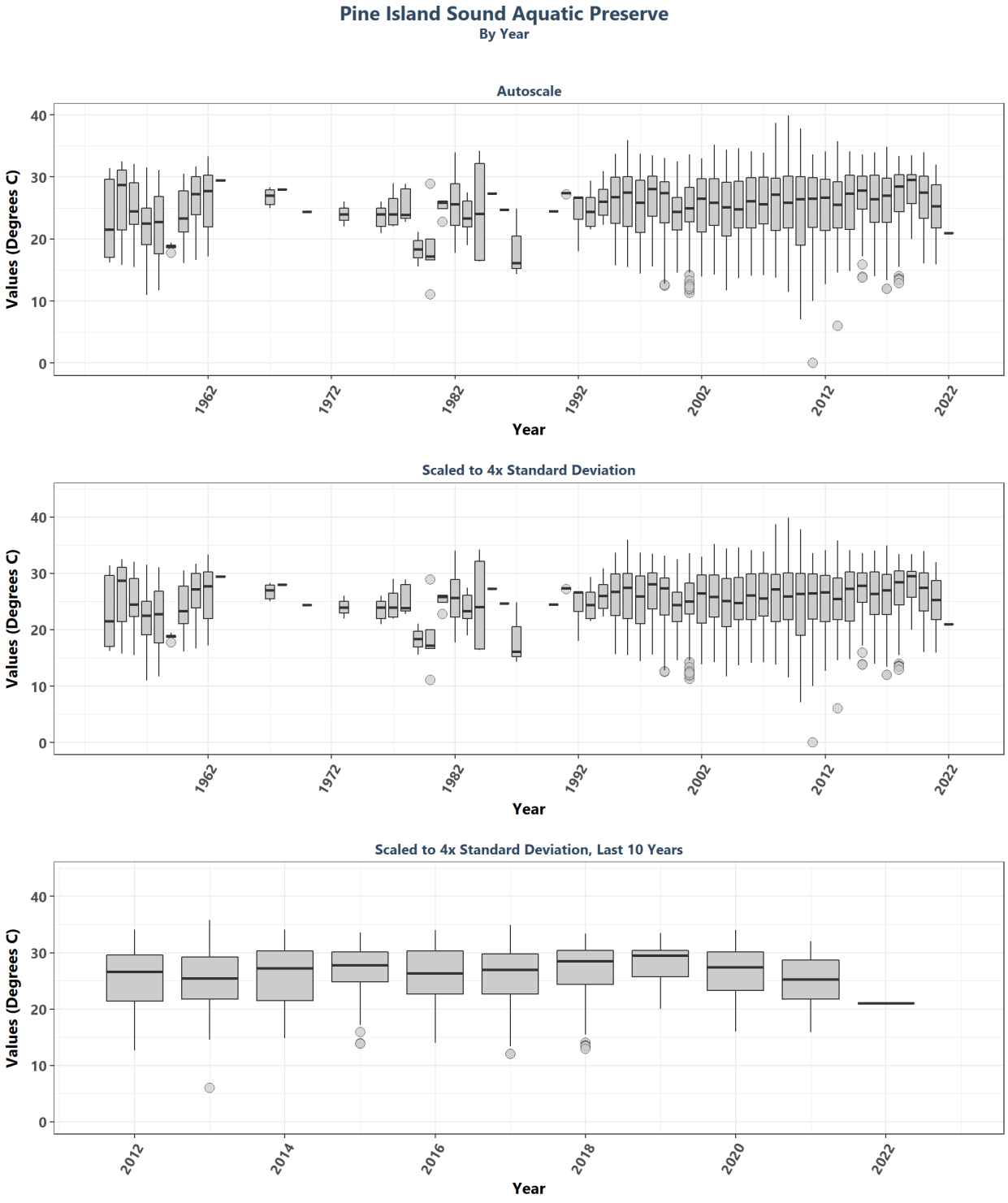


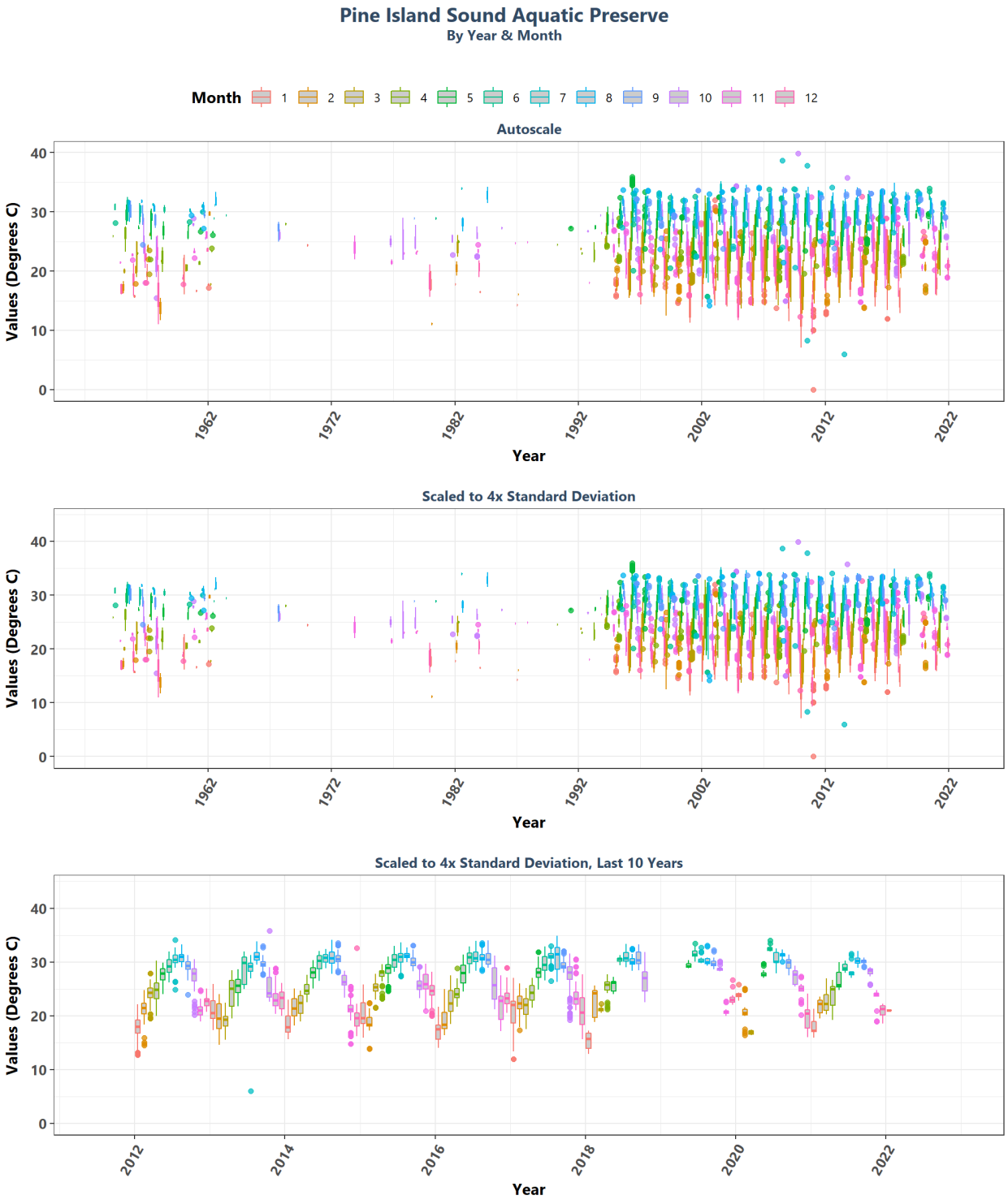


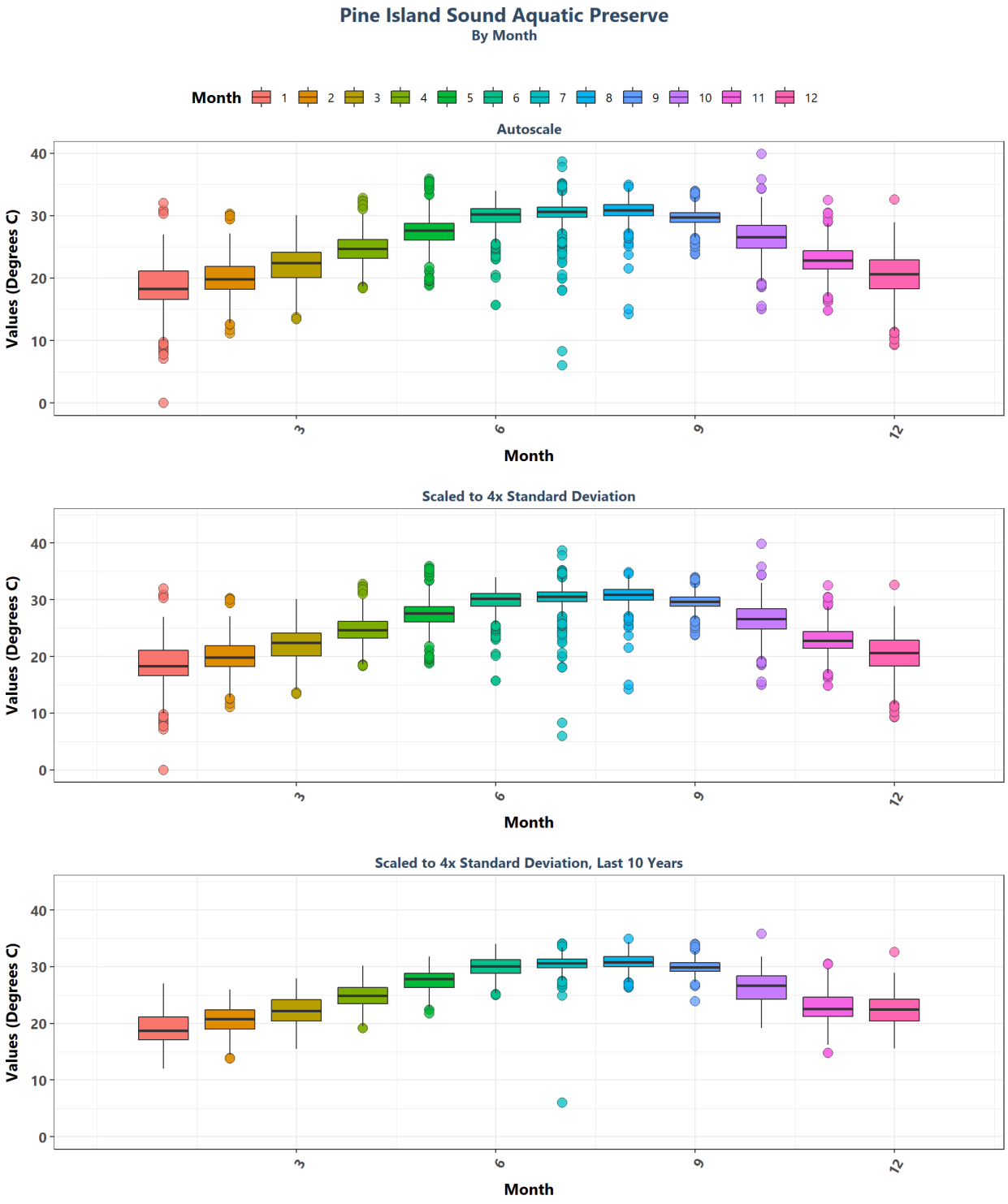


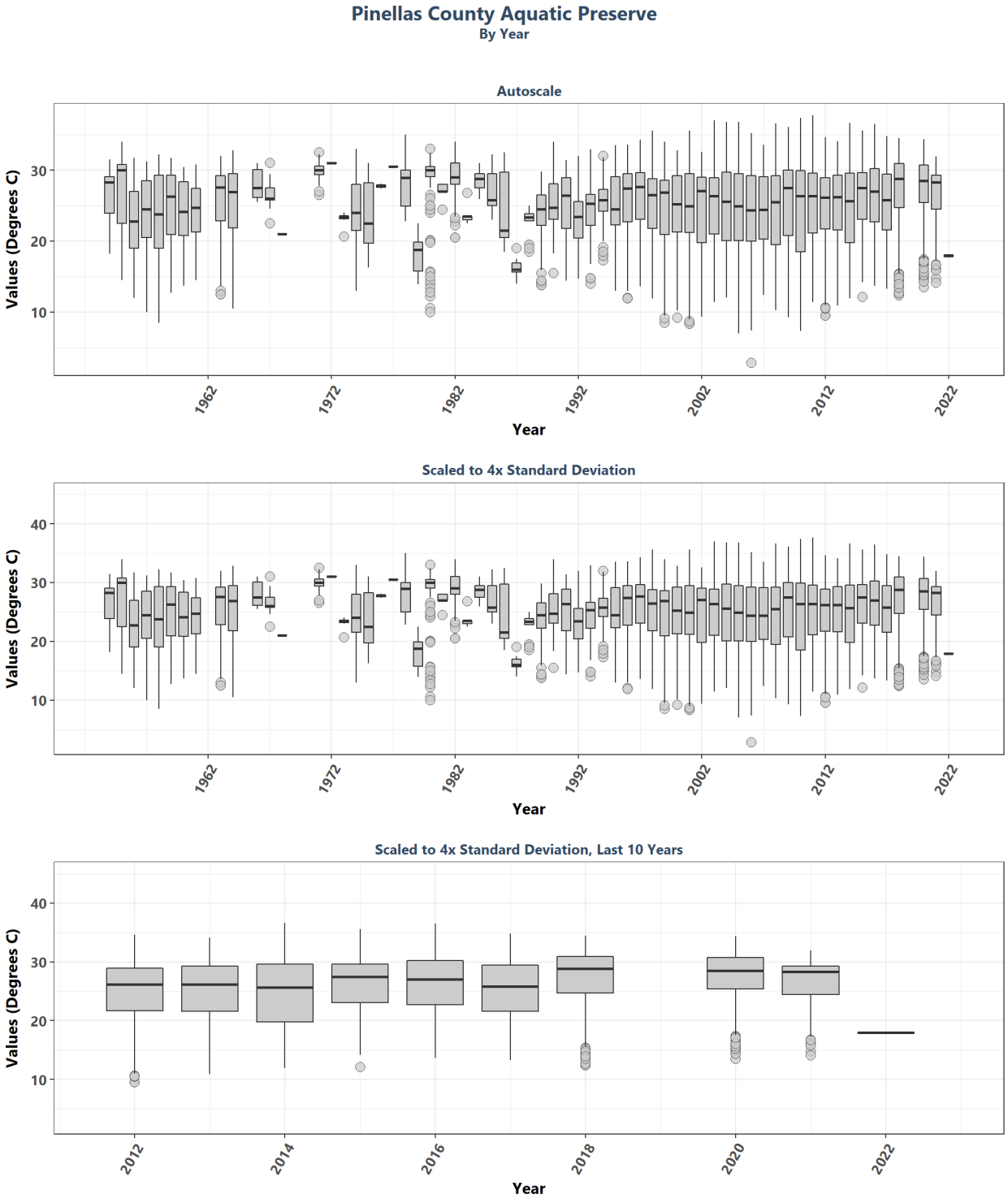

Pellicer Creek Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

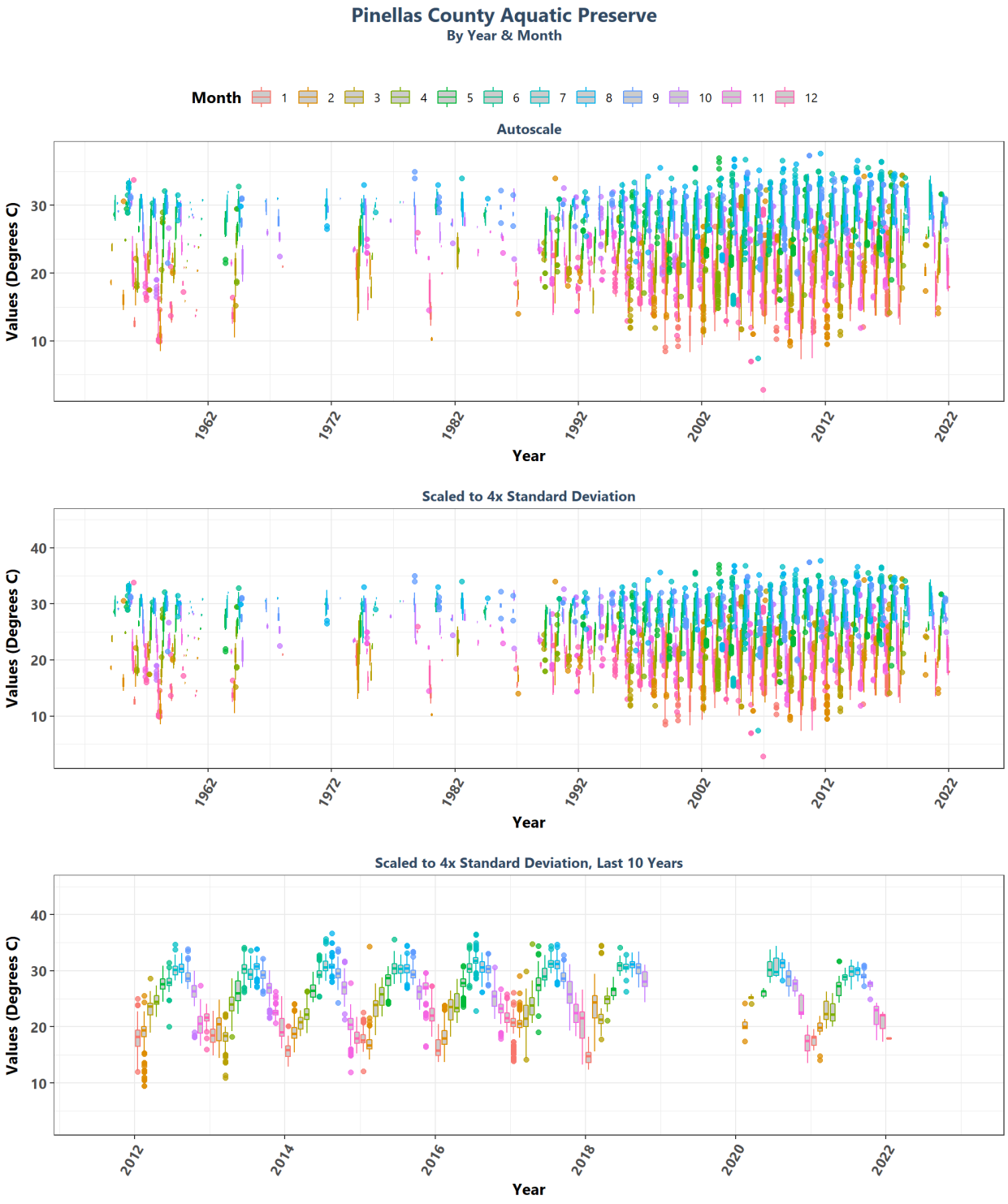

Pellicer Creek Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

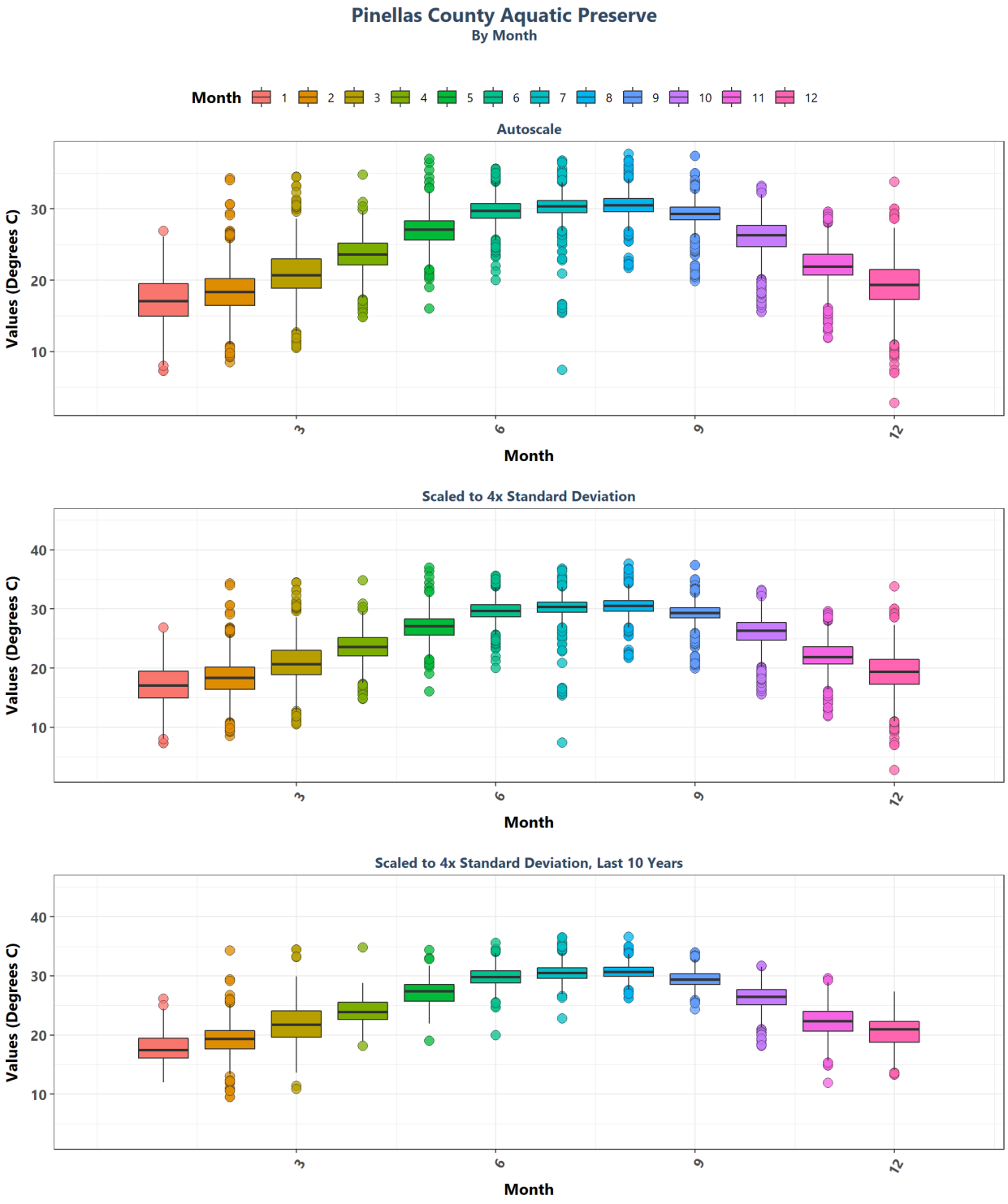

Pine Island Sound Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


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)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
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)
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

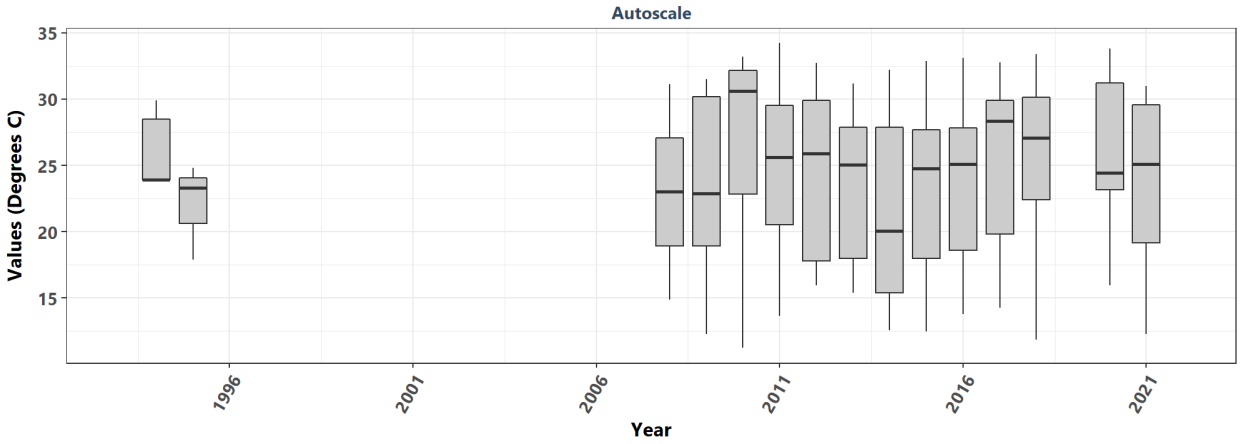

Pinellas County Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
30
20
10
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2012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

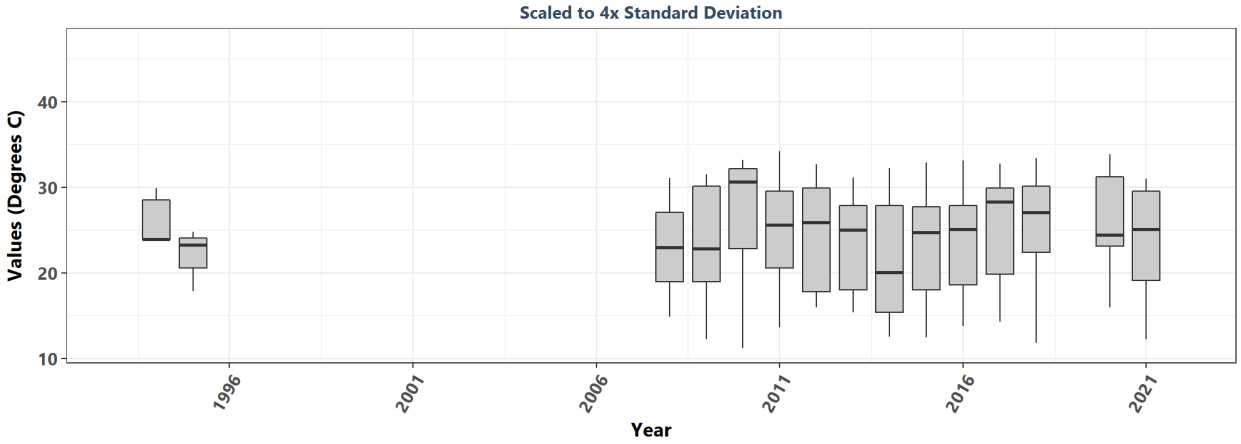

Pinellas County Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1962
1972
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

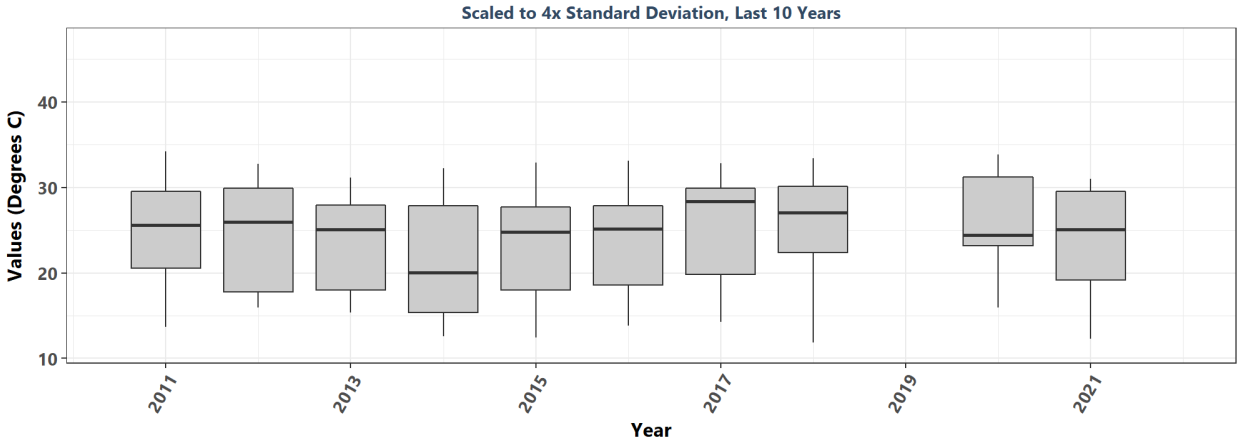

Pinellas County Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

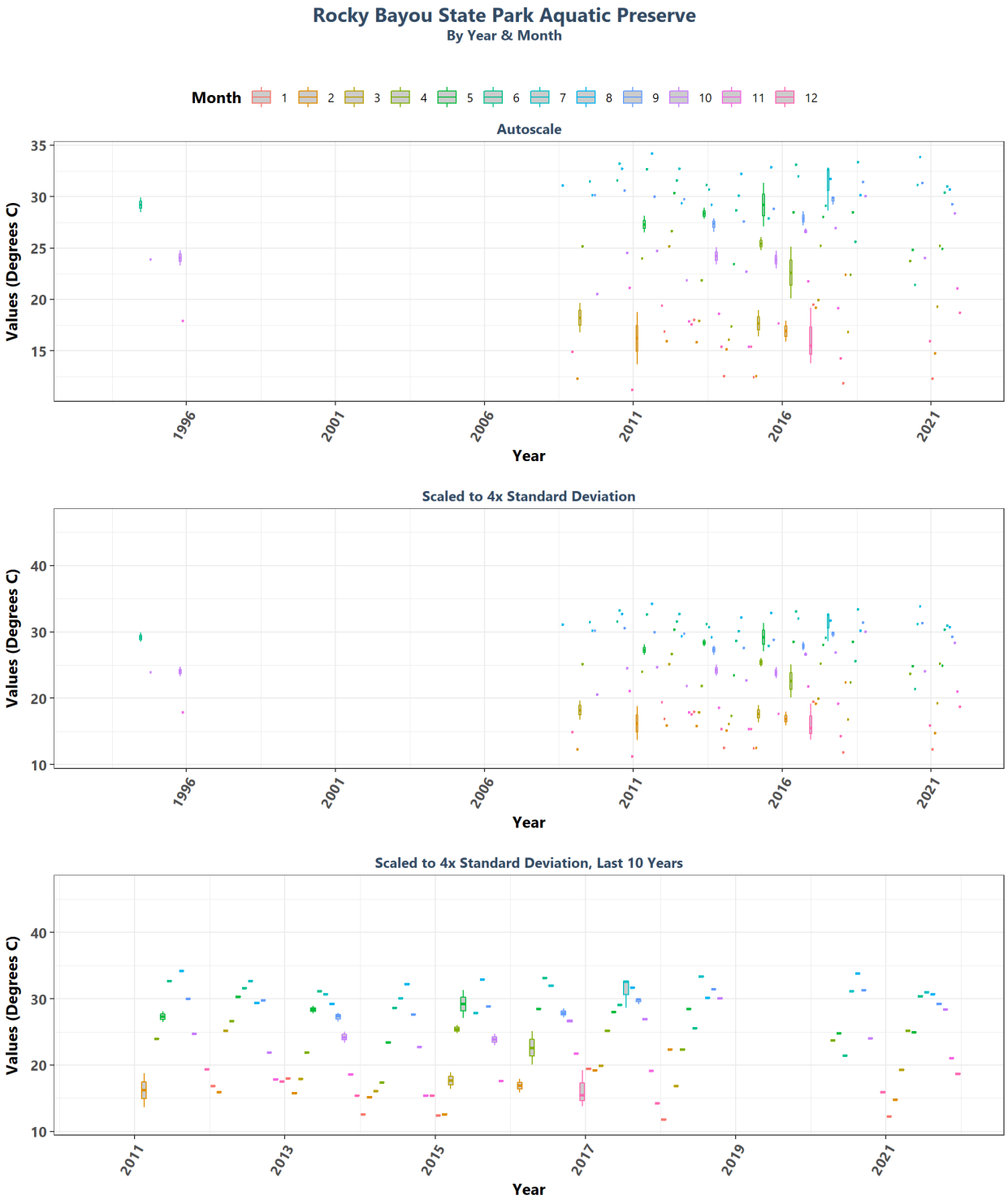
Rocky Bayou State Park Aquatic Preserve

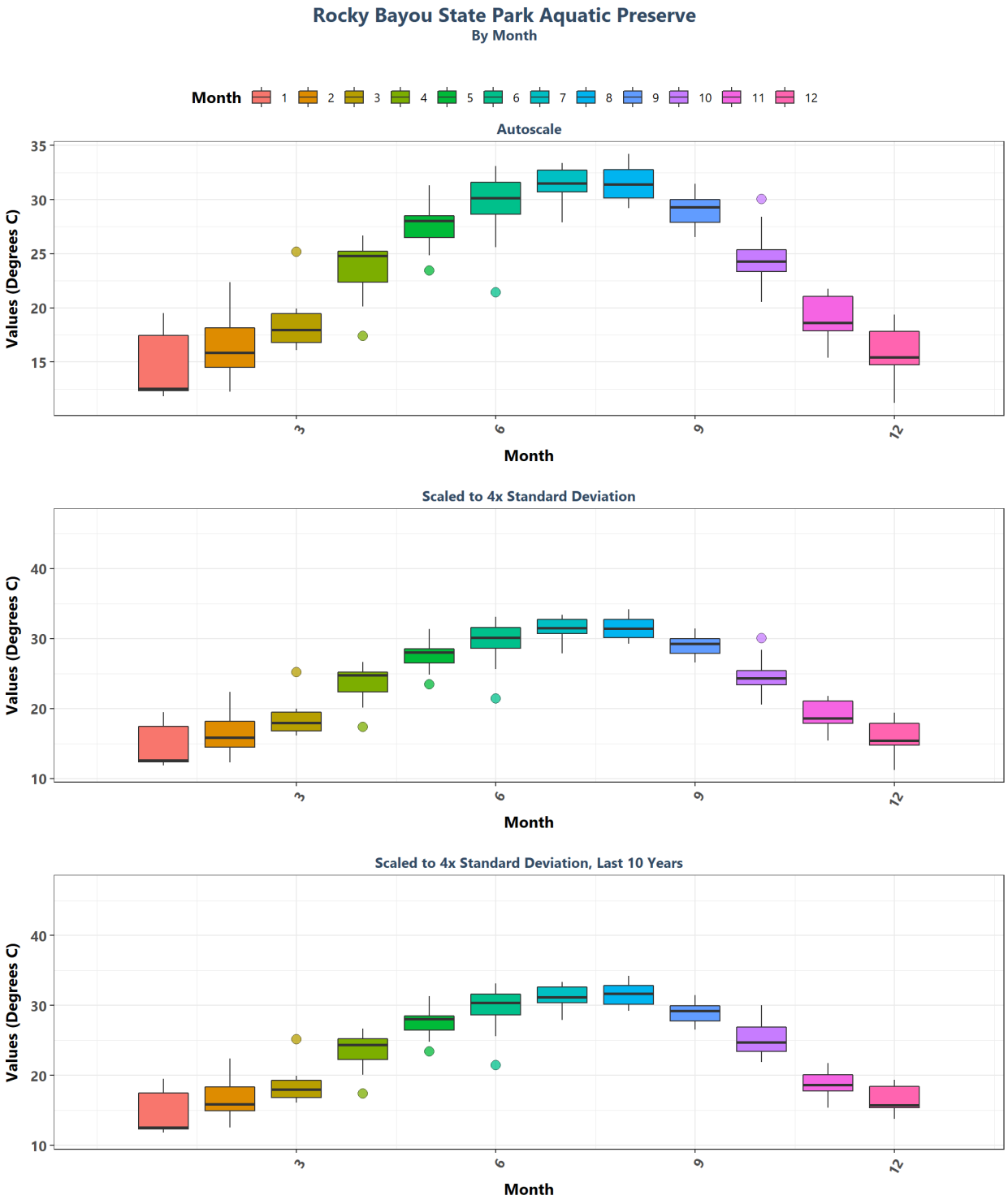
By Year





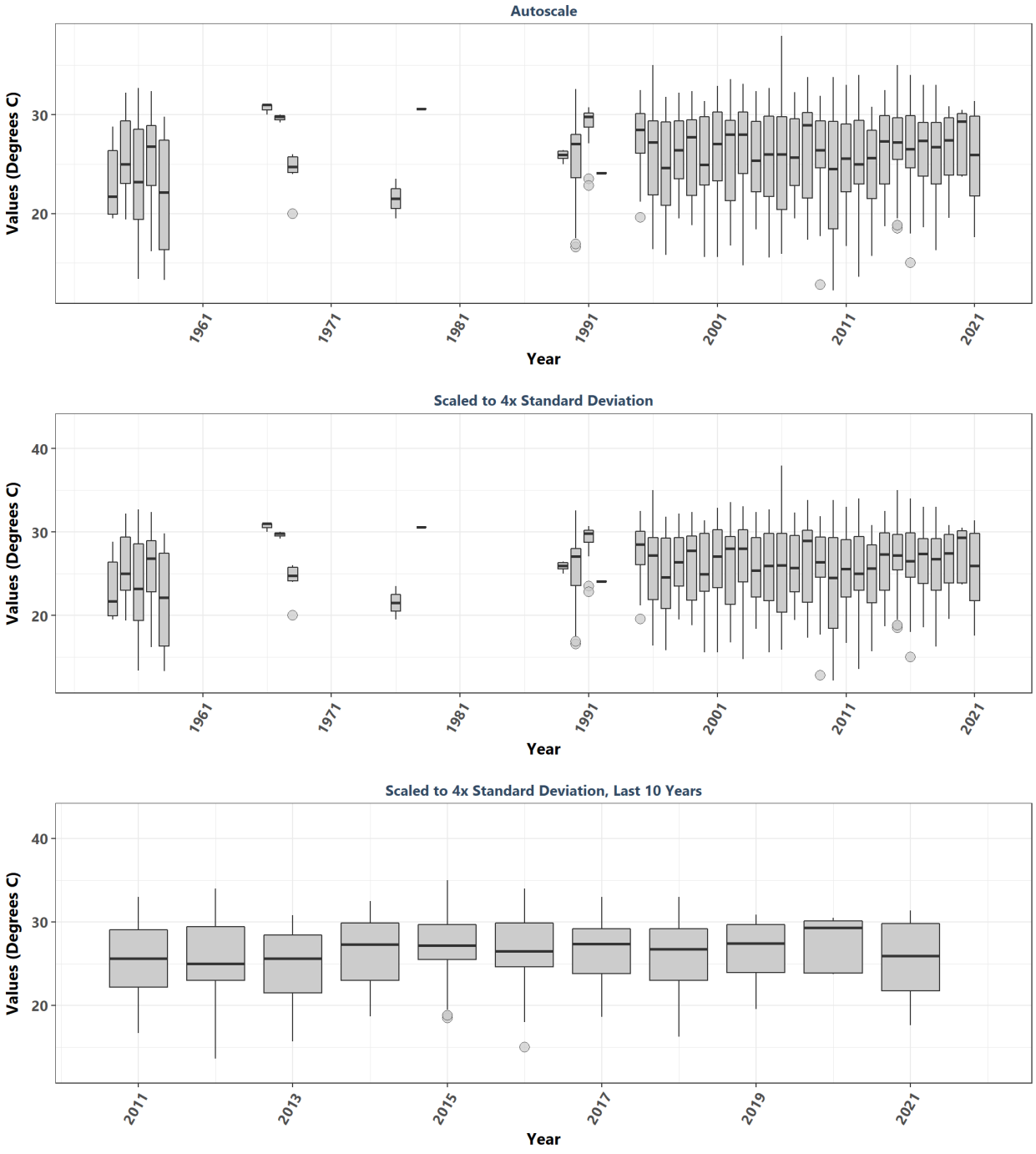


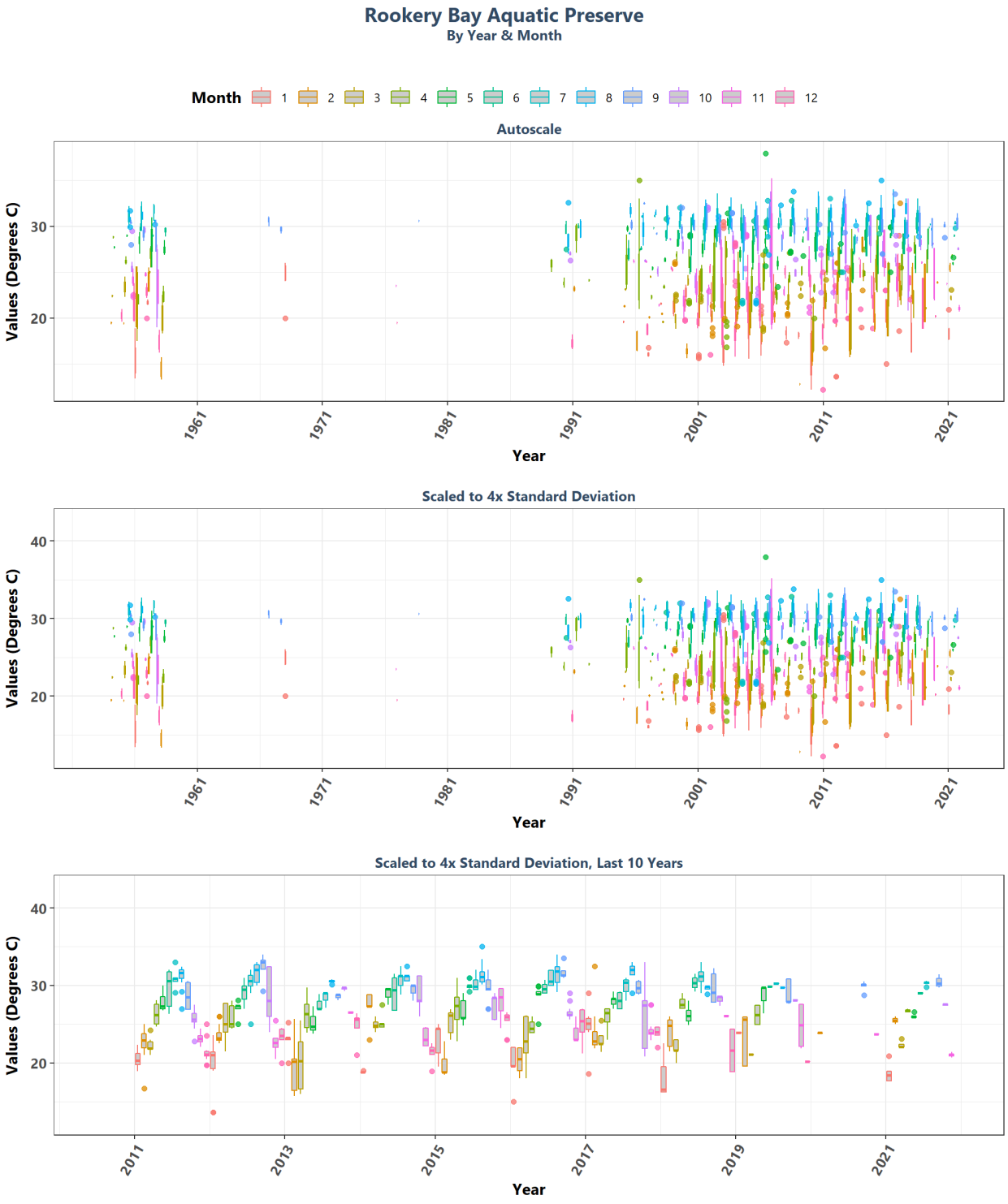

Rocky Bayou State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1996
2001
2006
2011
2016
2021
Year
Scaled to 4x Standard Deviation, Last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

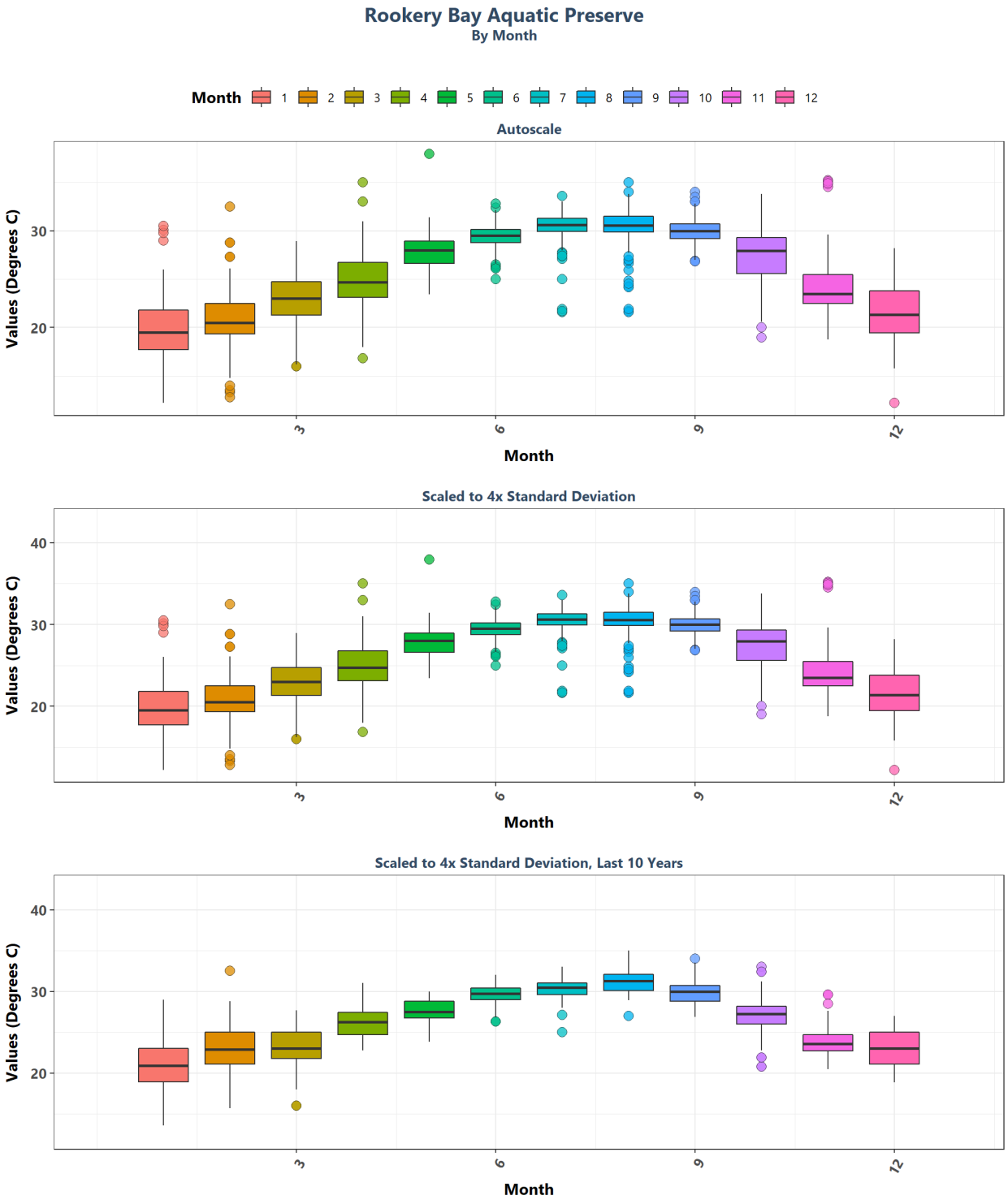

Rocky Bayou State Park Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
35
30
25
20
15
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
3
6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

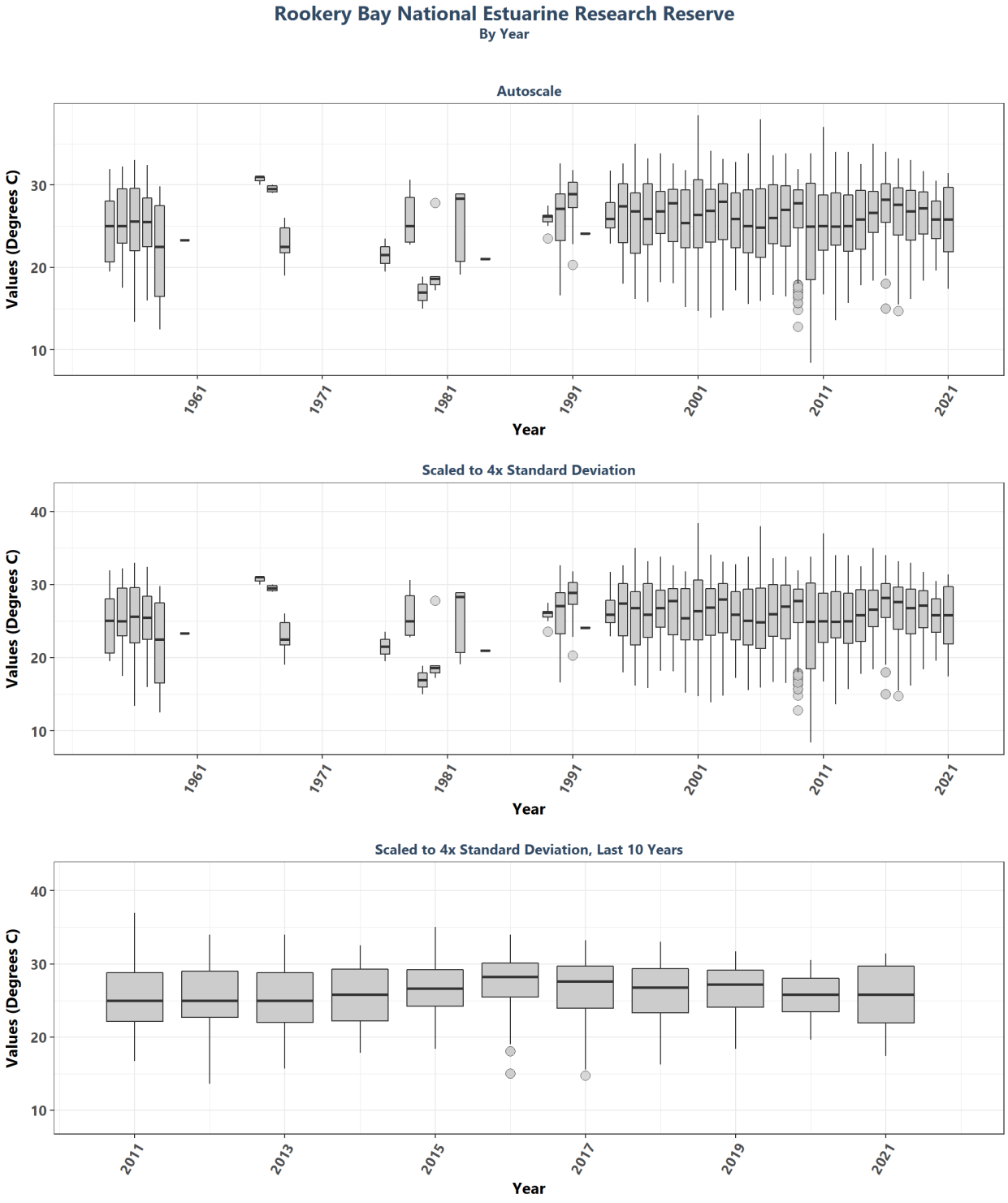
Rookery Bay Aquatic Preserve

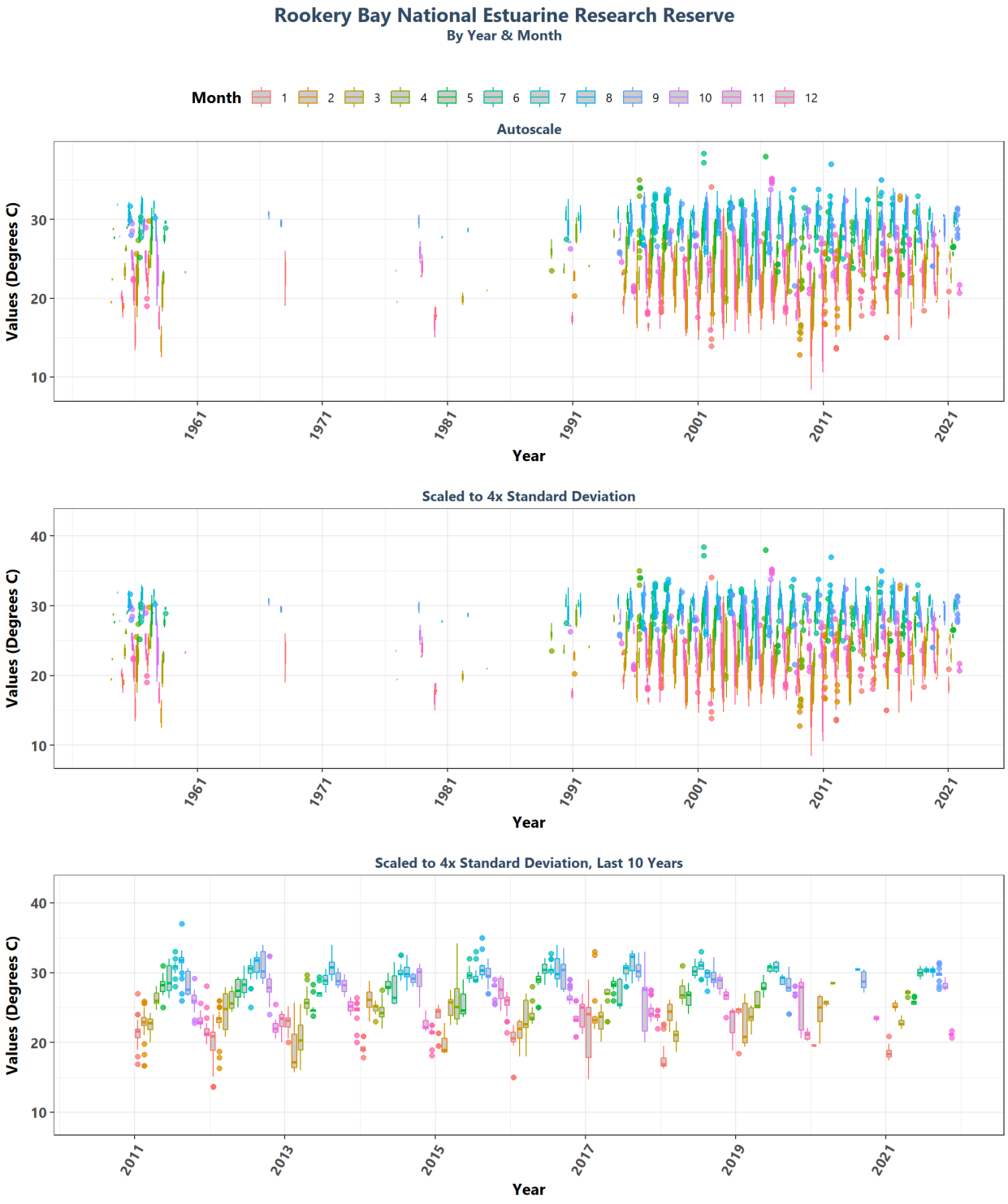
By Year

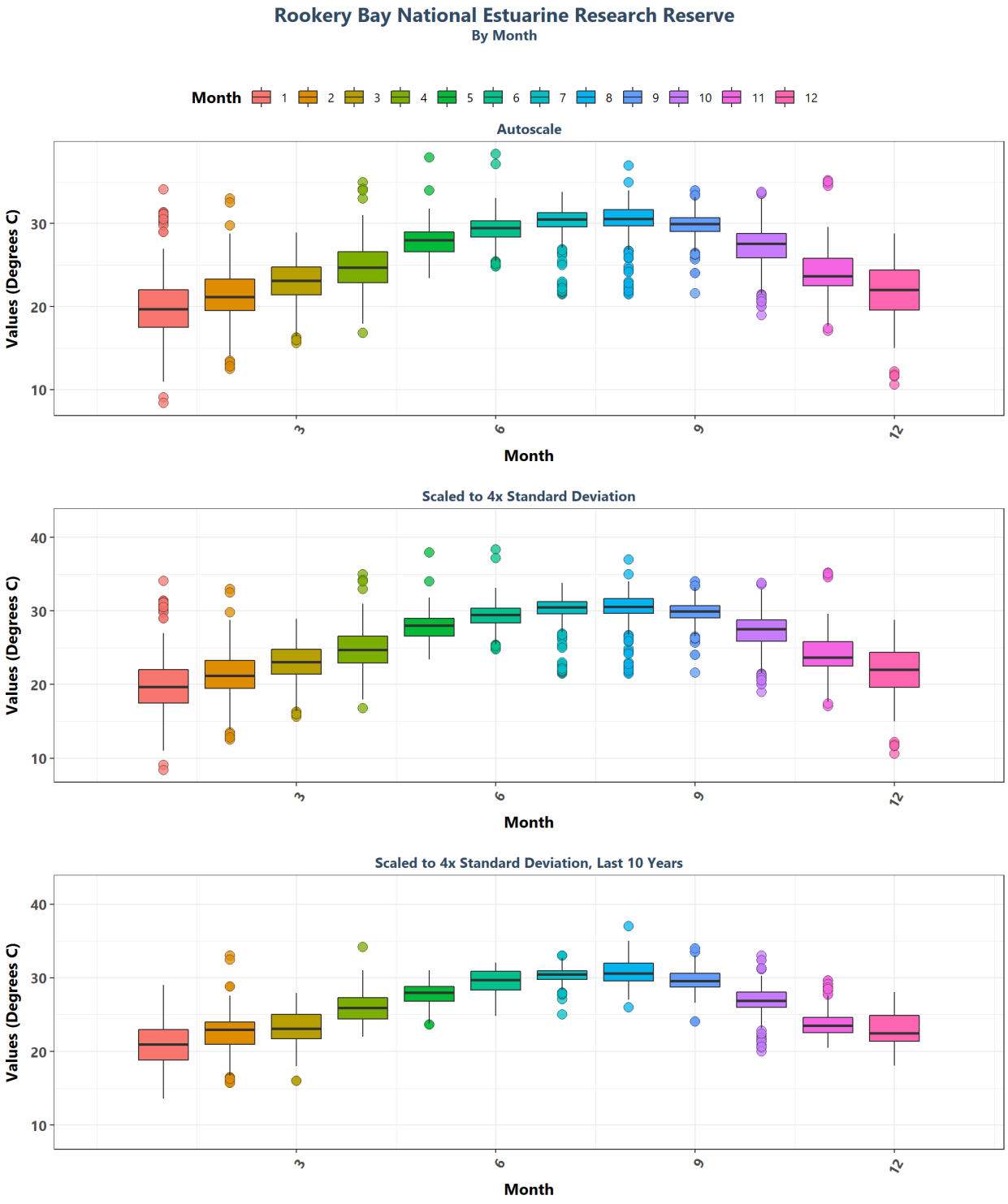



Rookery Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
2011
2013
2015
2017
2019
2021
Year


Rookery Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
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

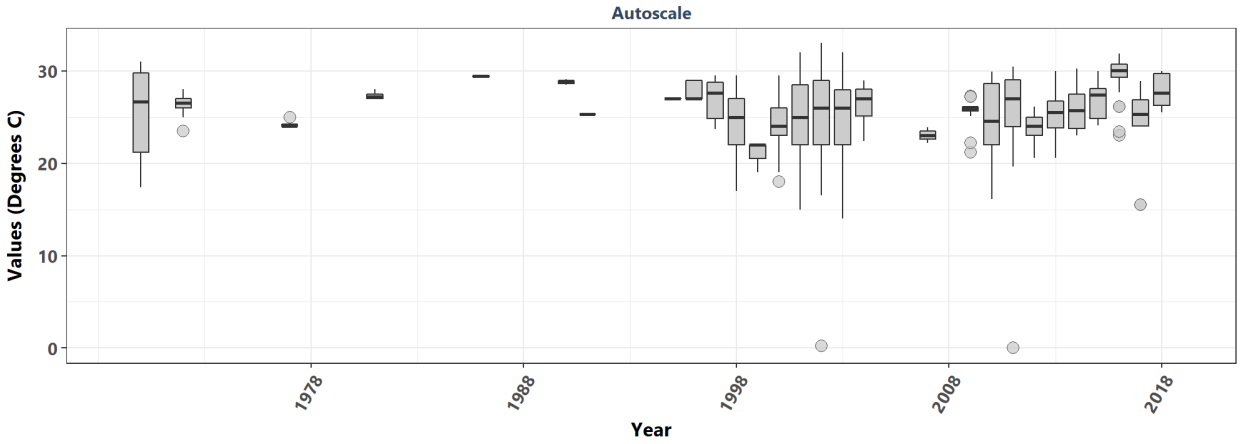

Rookery Bay National Estuarine Research Reserve
By Year
Autoscale
Values (Degrees C)
10
20
30
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1961
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)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

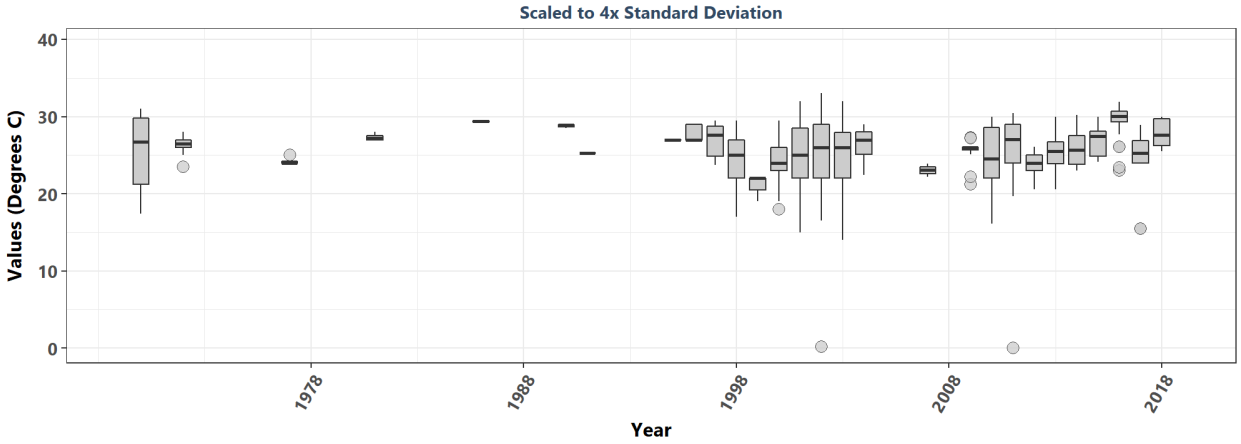

Rookery Bay National Estuarine Research Reserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

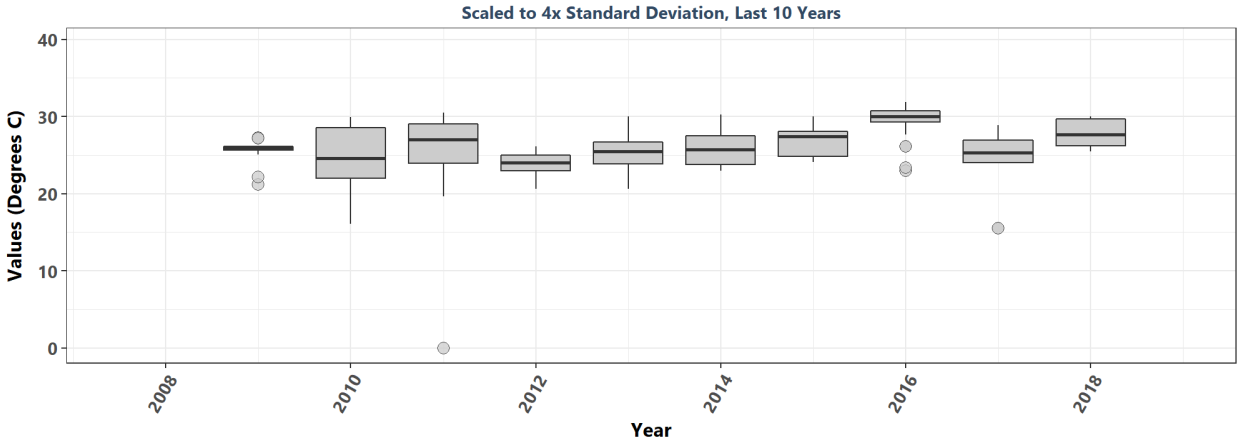

Rookery Bay National Estuarine Research Reserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3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

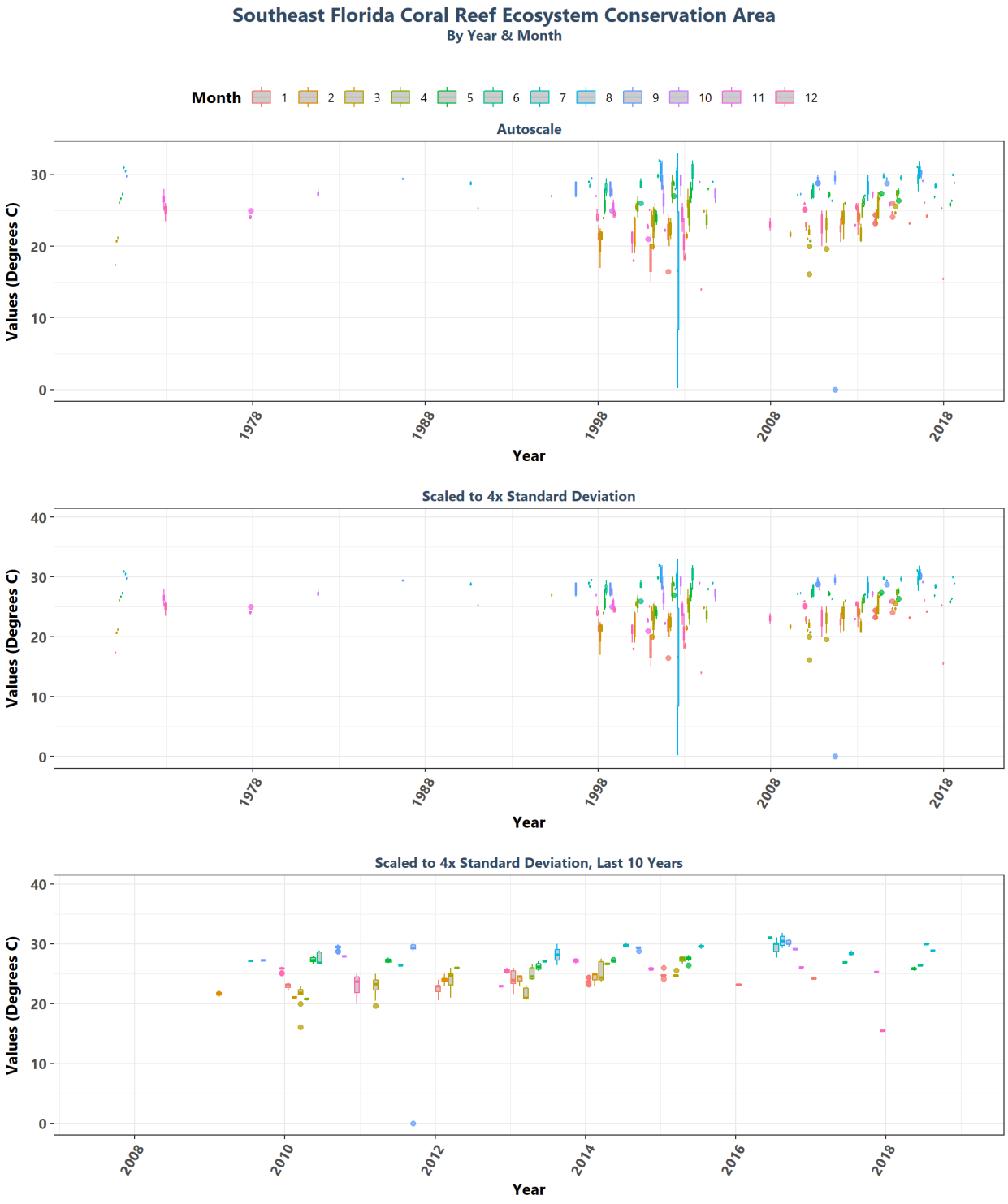
Southeast Florida Coral Reef Ecosystem Conservation Area

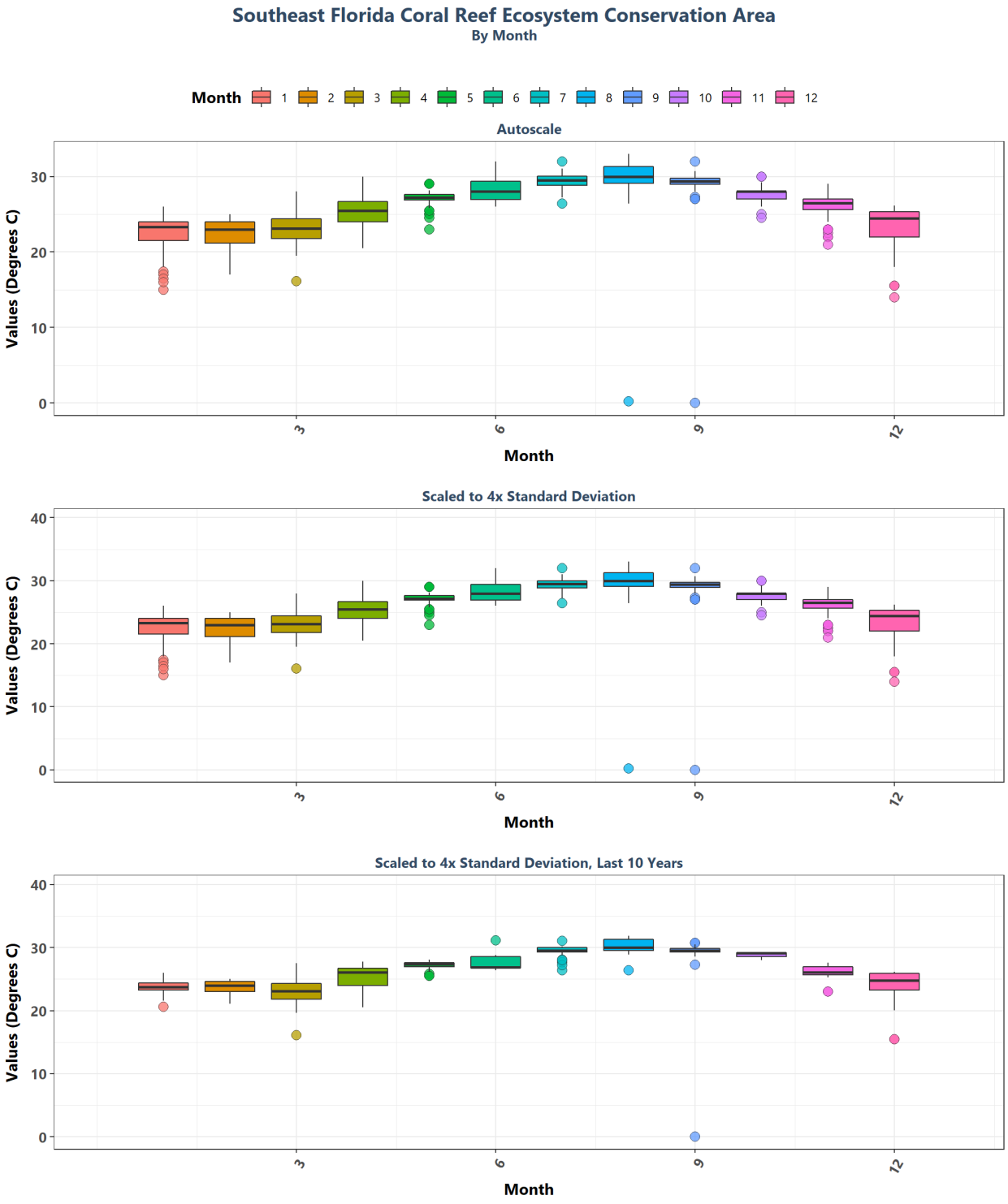
By Year



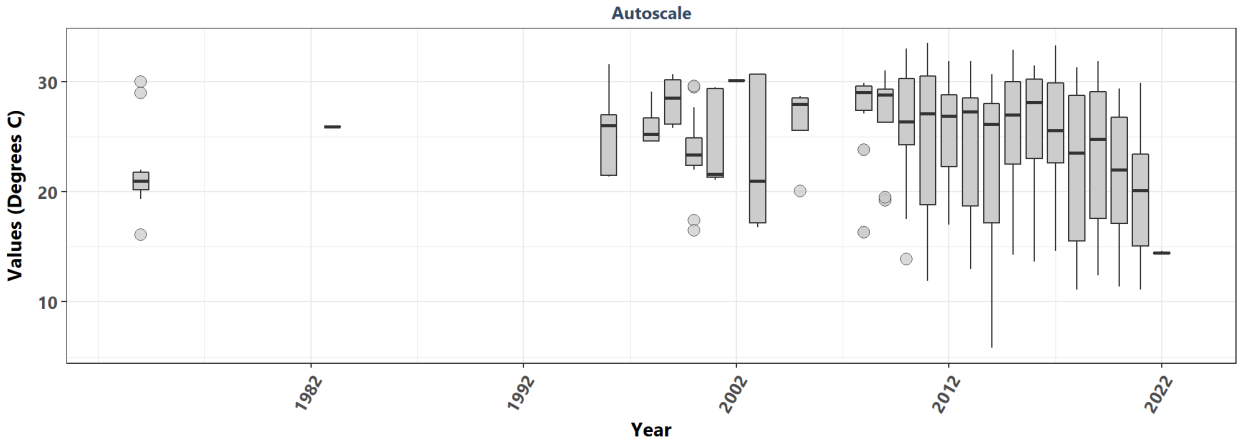


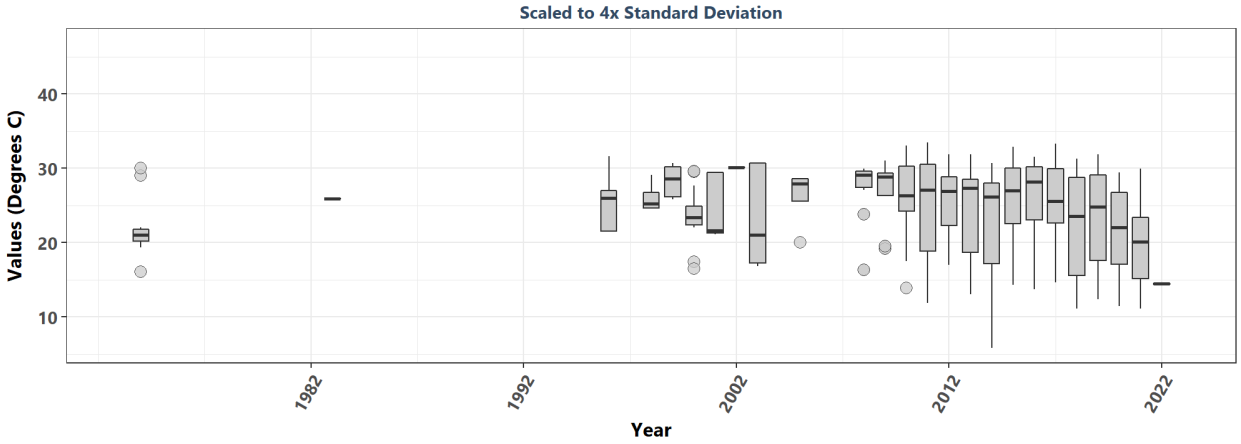


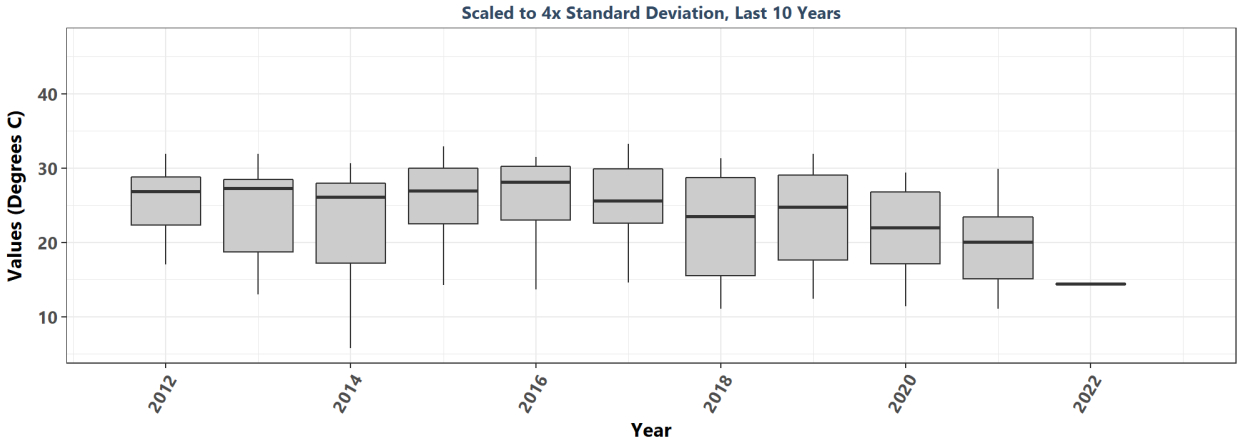

Southeast Florida Coral Reef Ecosystem Conservation Area
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
0 10 20 30
1978 1988 1998 2008 2018
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
0 10 20 30 40
1978 1988 1998 2008 2018
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
0 10 20 30 40
2008 2010 2012 2014 2016 2018
Year

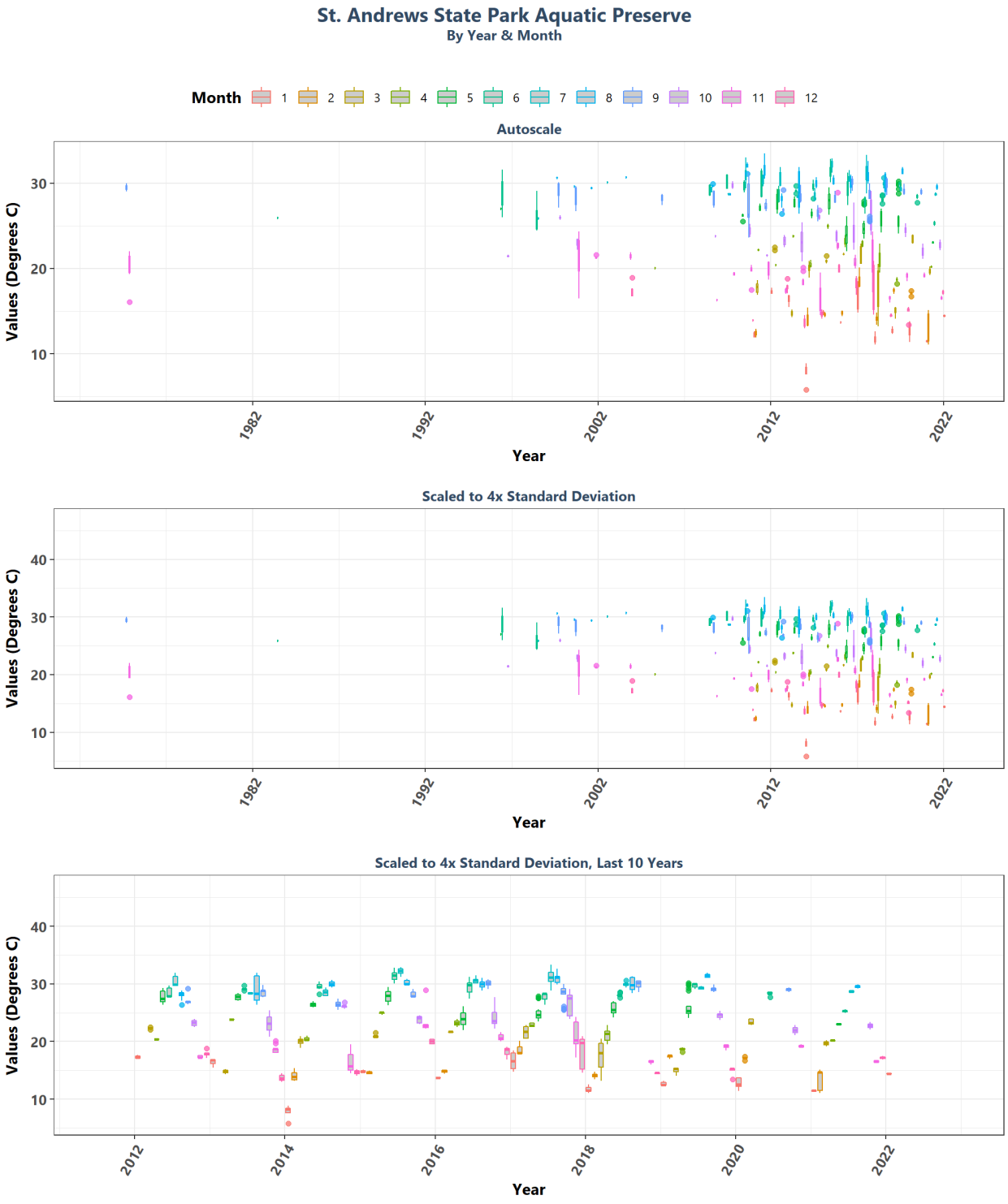

Southeast Florida Coral Reef Ecosystem Conservation Area
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Month

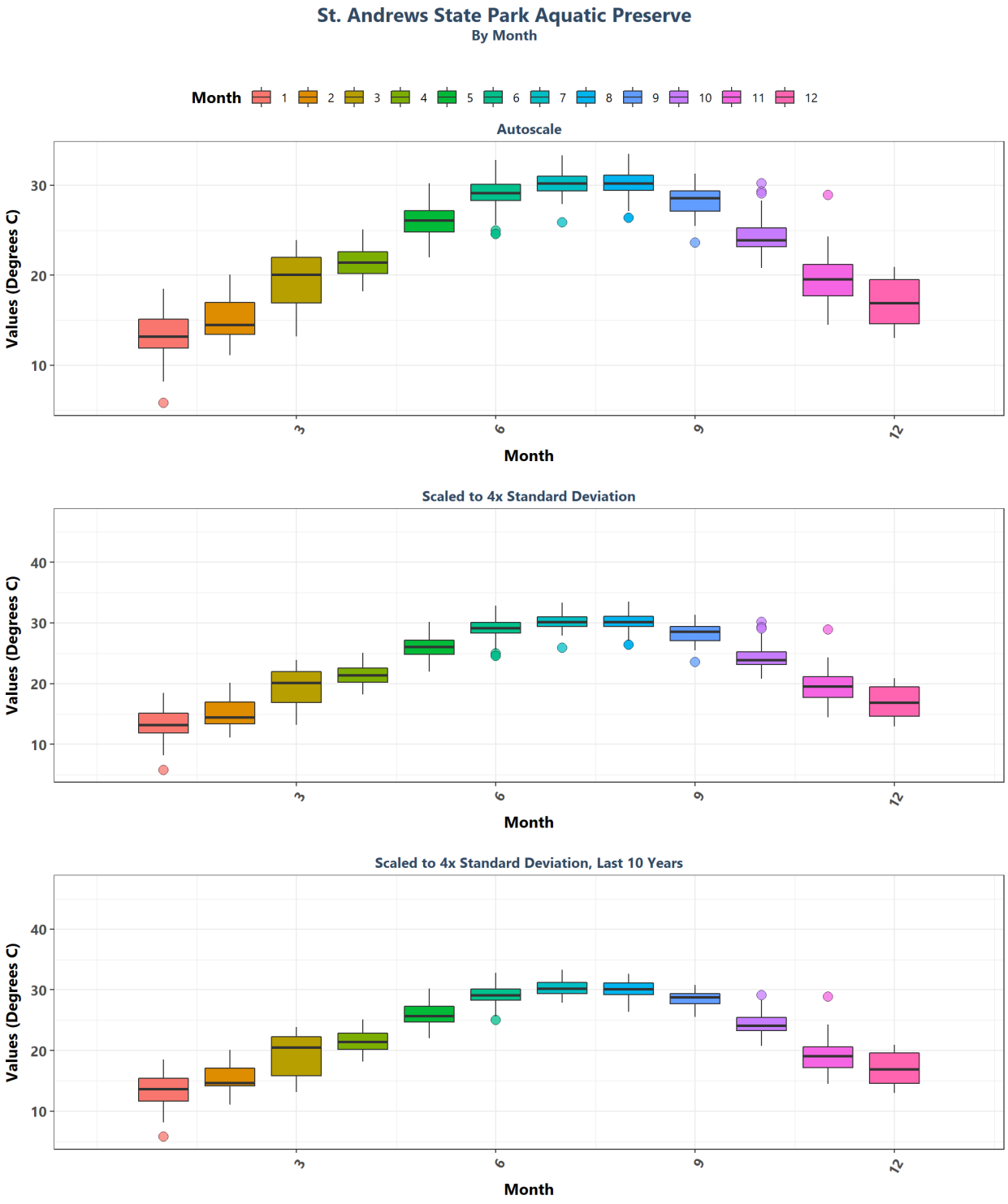
St. Andrews State Park Aquatic Preserve
By Year

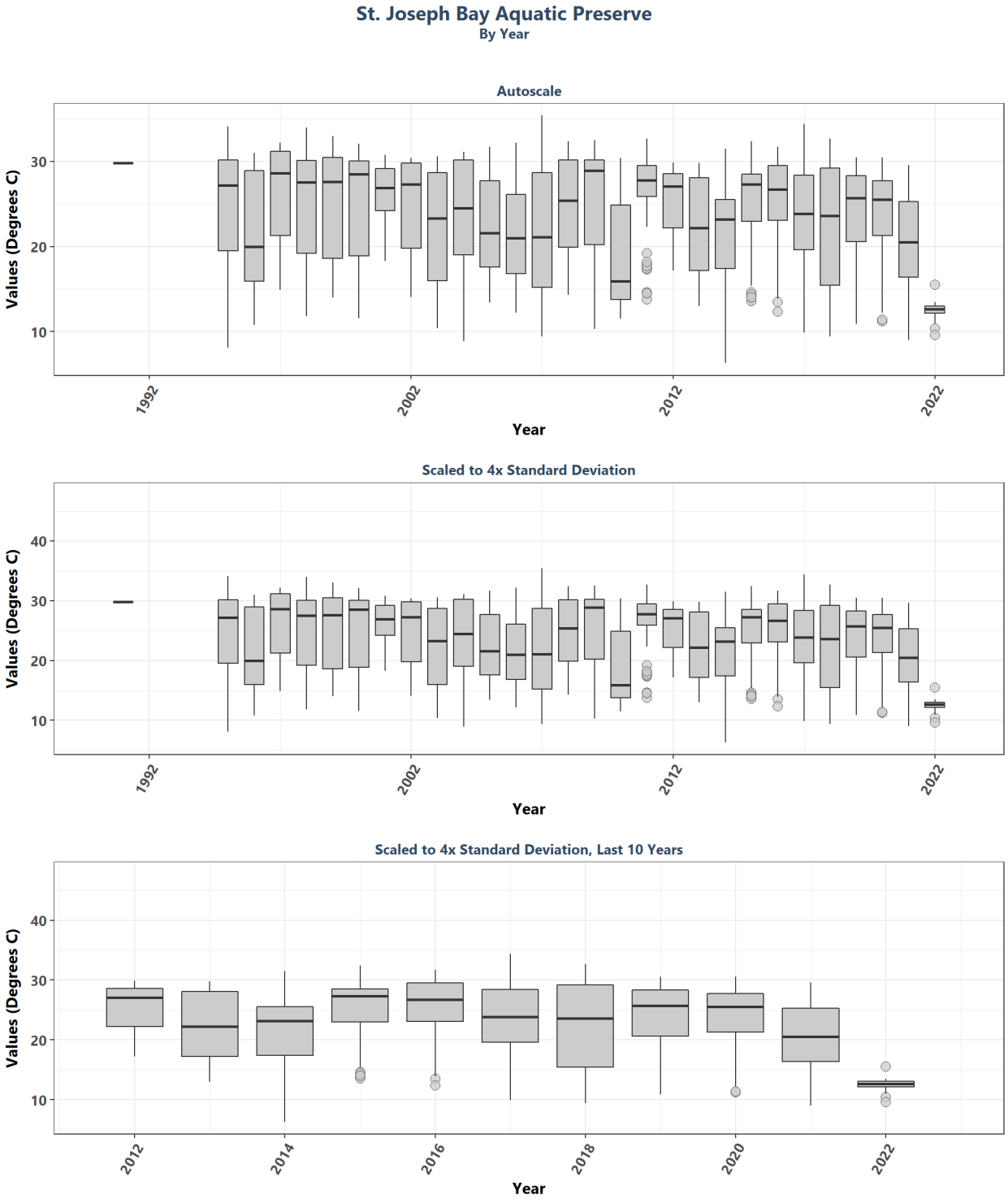


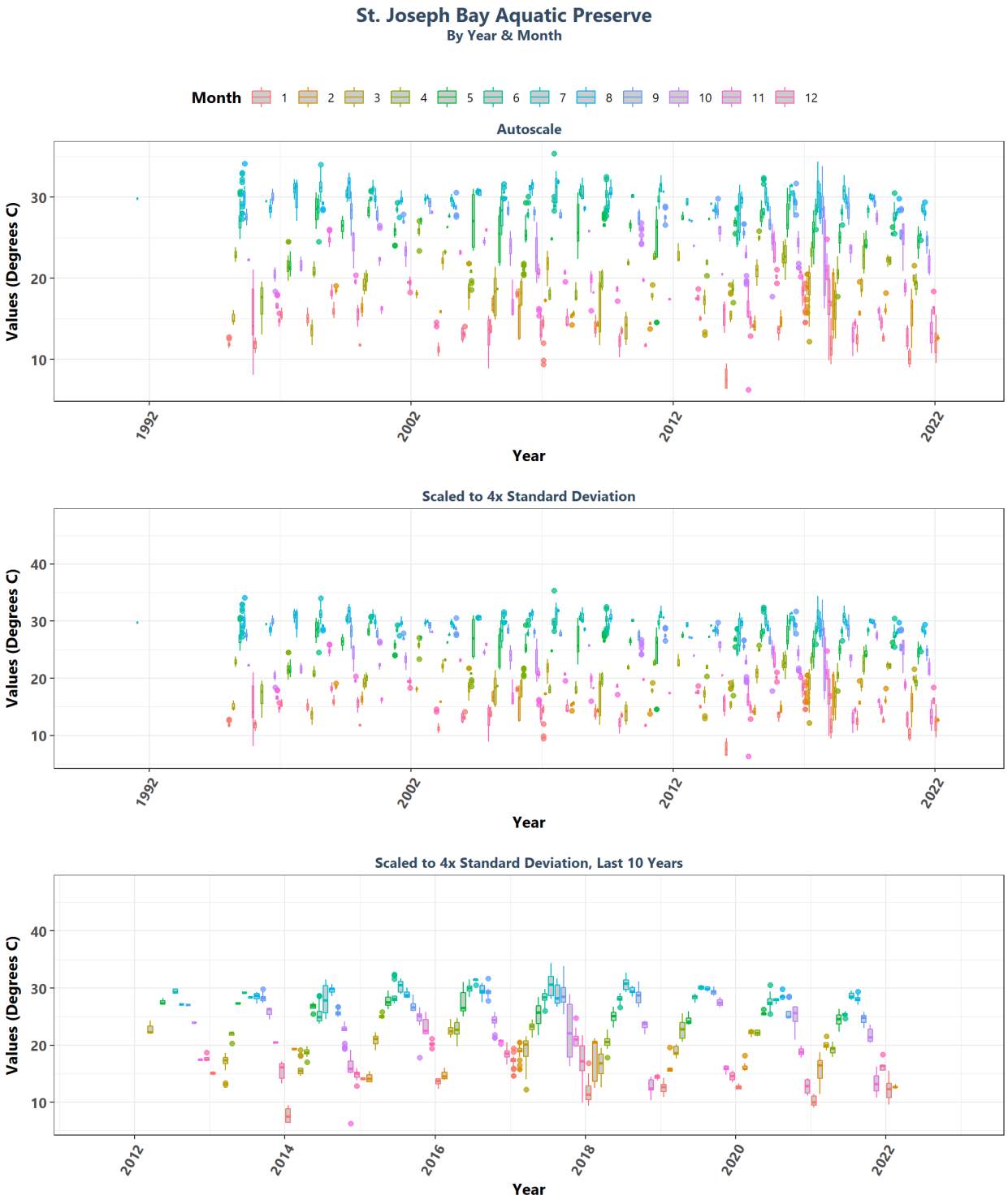


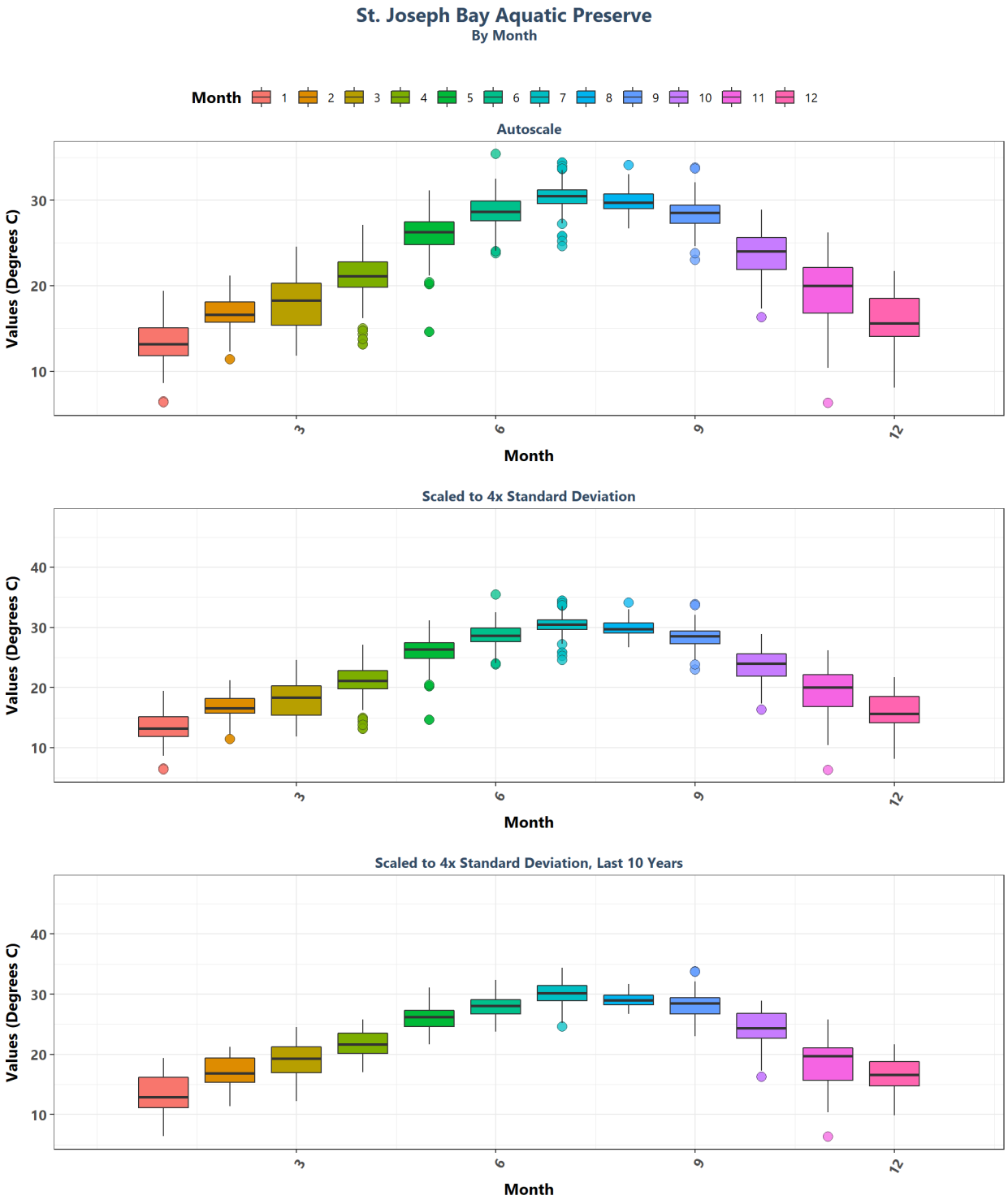



St. Andrews State Park Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
30
20
10
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1982
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
40
30
20
10
2012
2014
2016
2018
2020
2022
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)
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

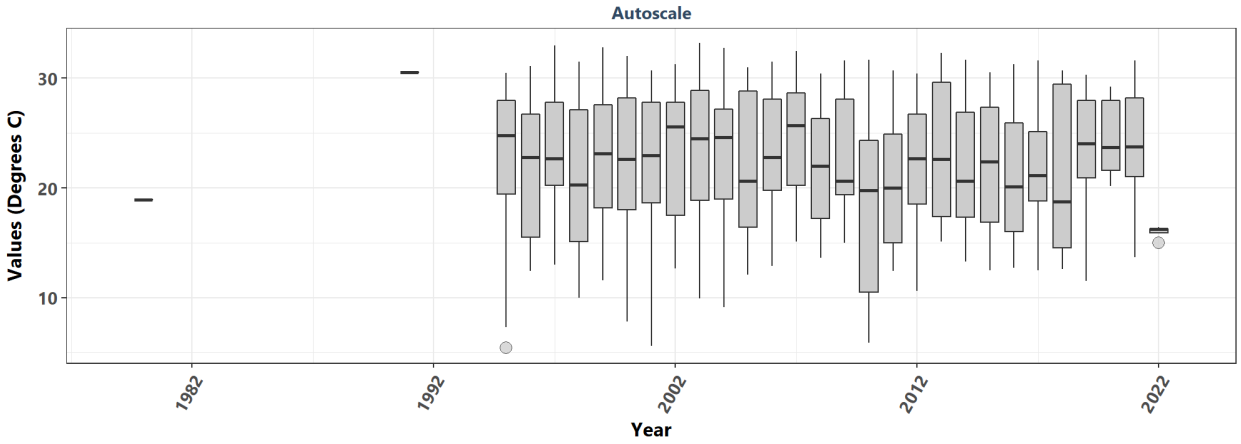

St. Joseph Bay Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
10
20
30
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1992
2002
2012
2022
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2012
2014
2016
2018
2020
2022
Year

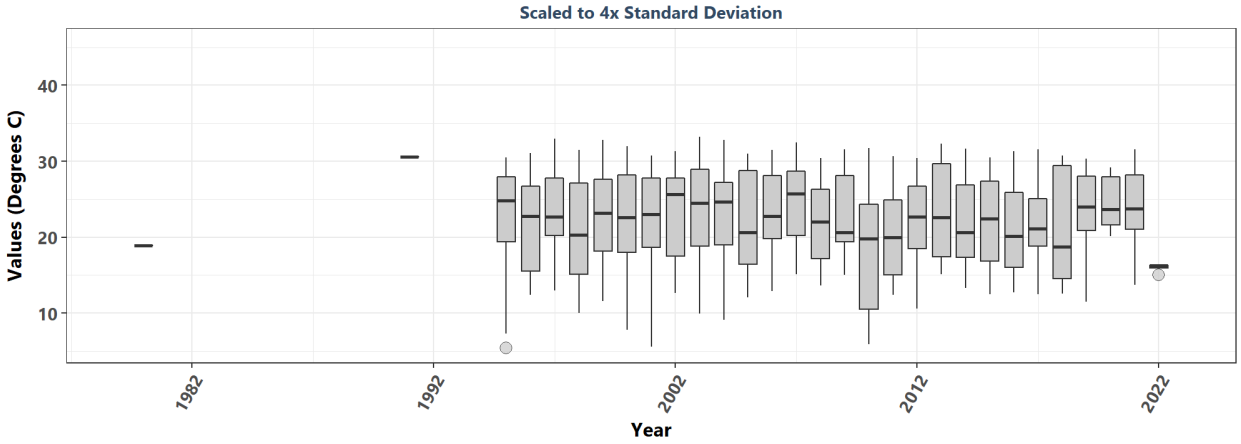

St. Joseph Bay Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

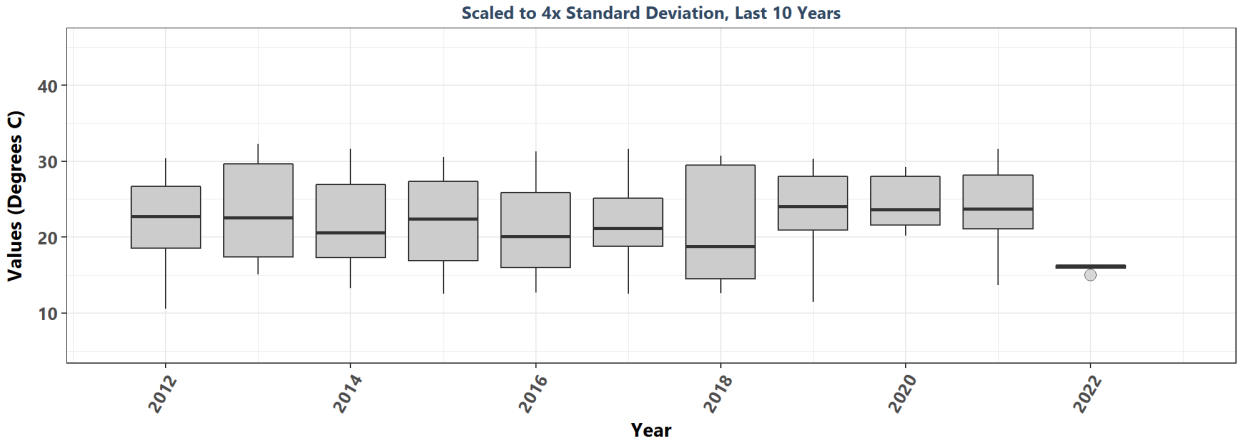

St. Joseph Bay Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

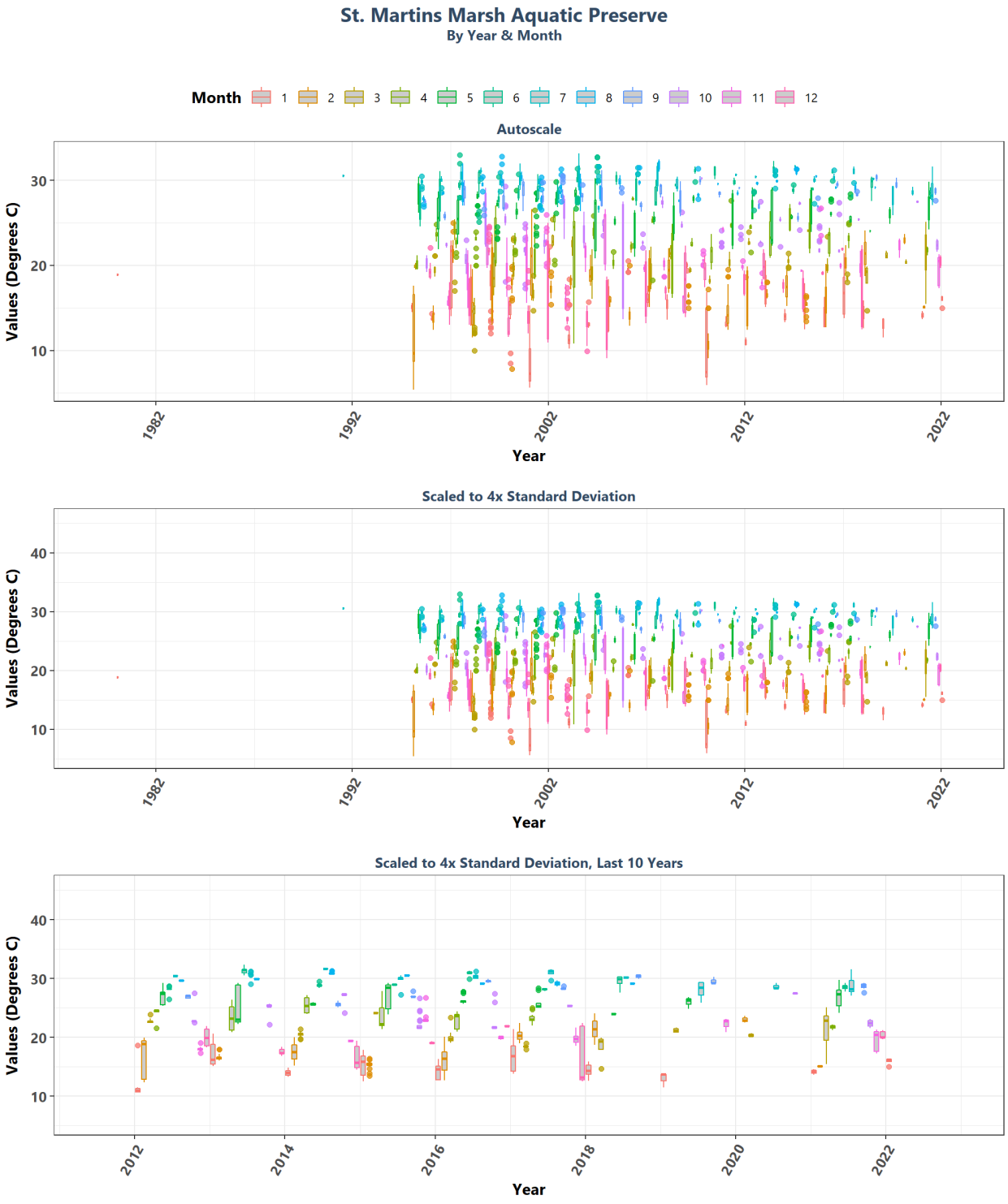
St. Martins Marsh Aquatic Preserve

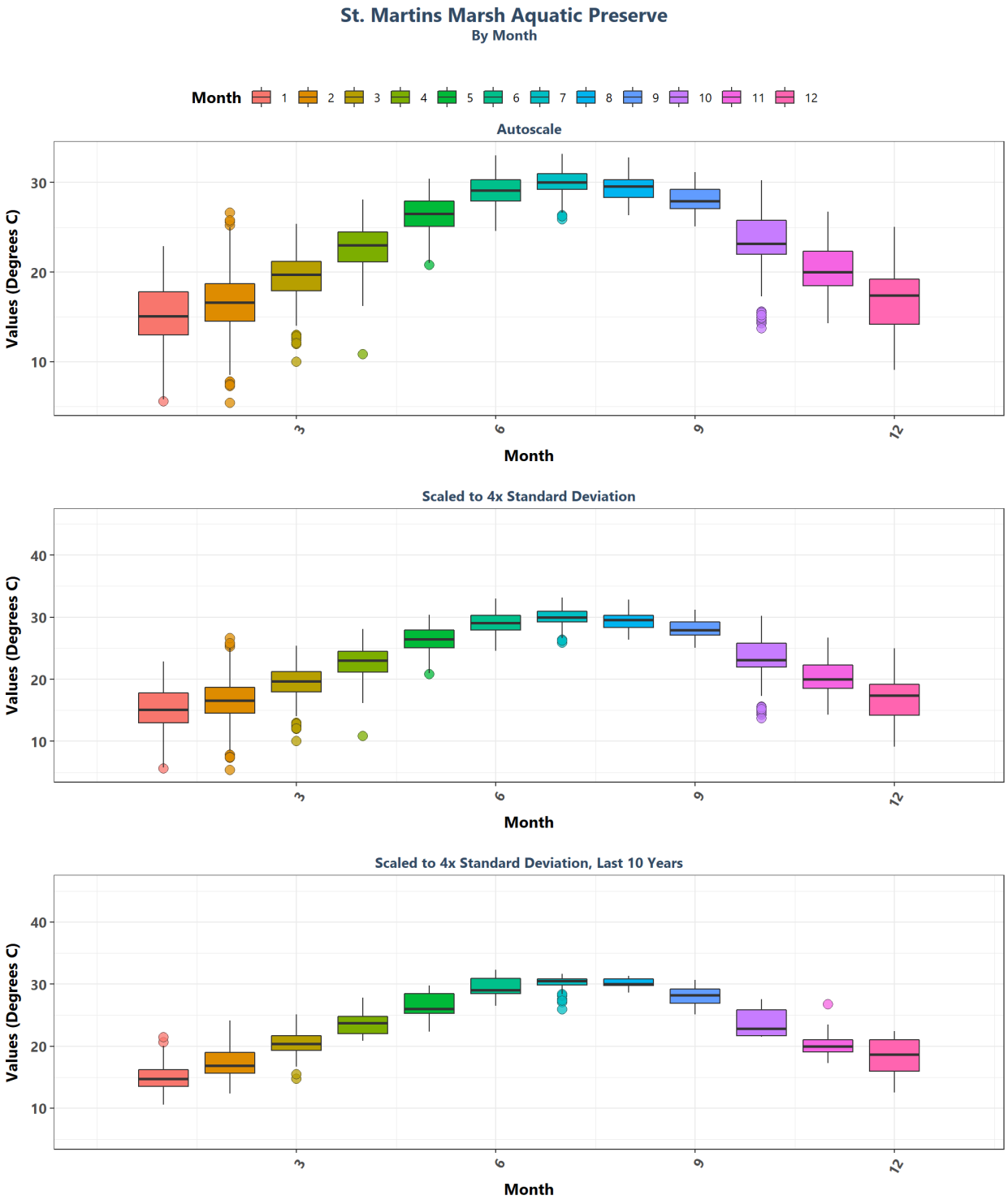
By Year

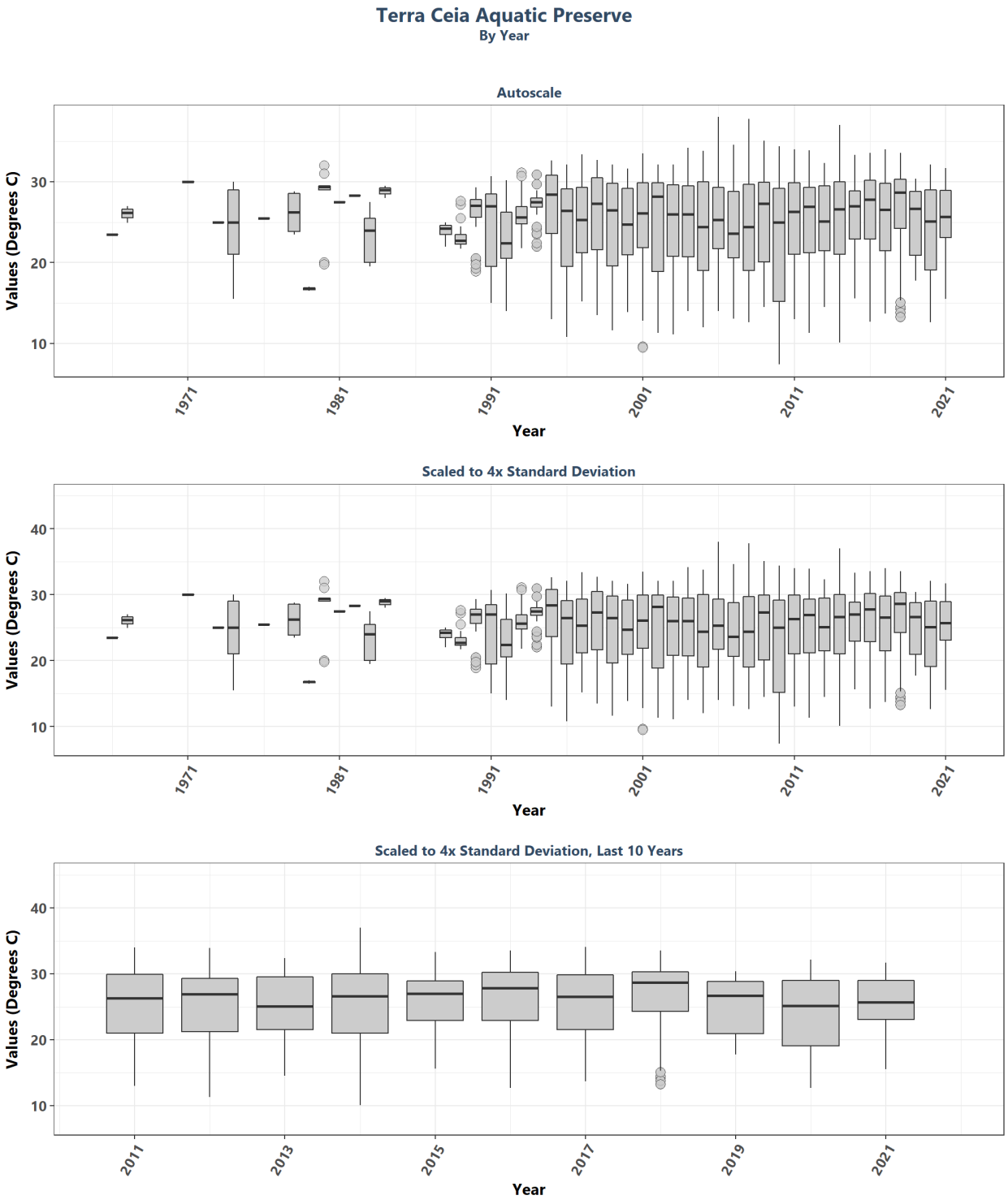


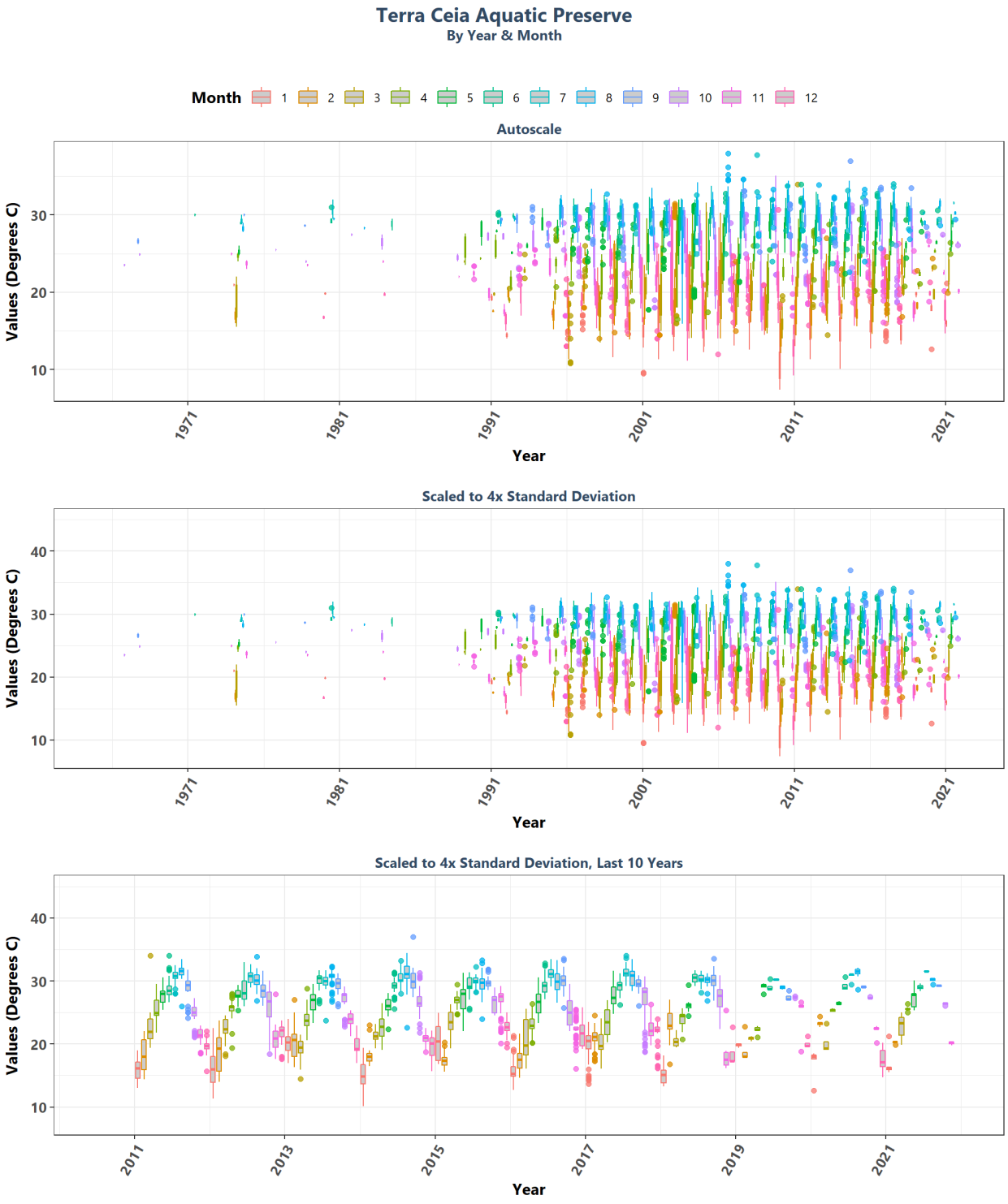


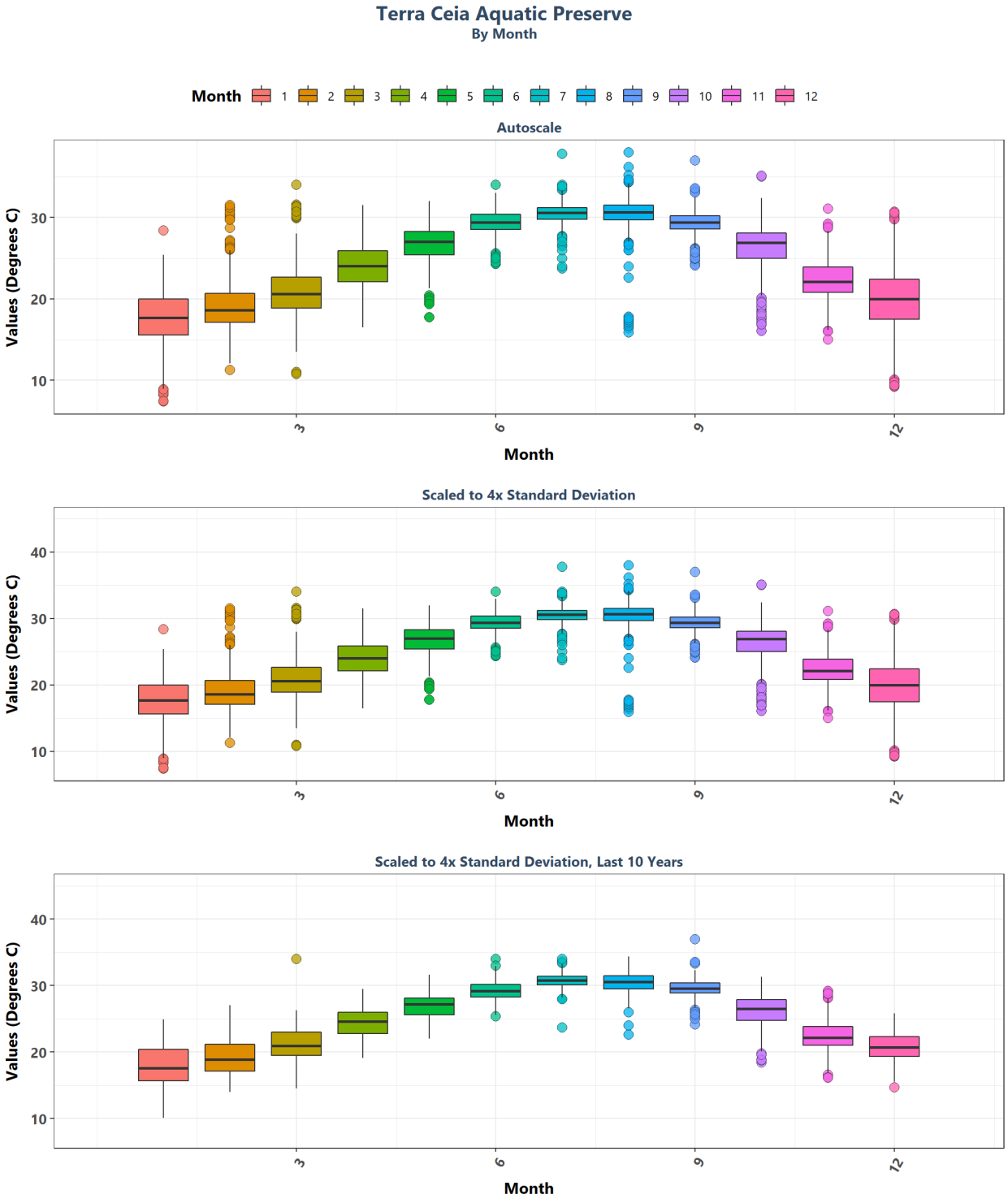



St. Martins Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


St. Martins Marsh 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

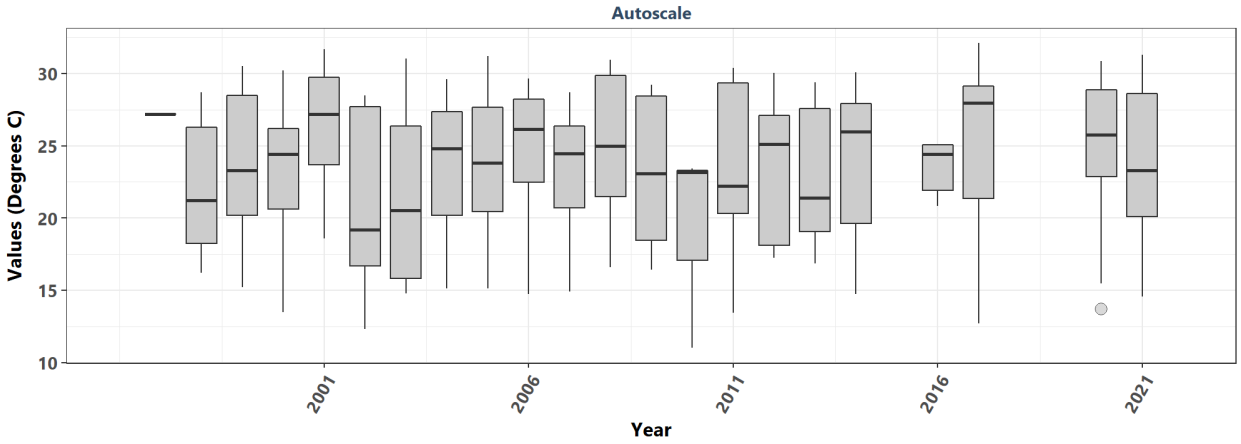

Terra Ceia Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
10
20
30
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
1971
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
2011
2013
2015
2017
2019
2021
Year

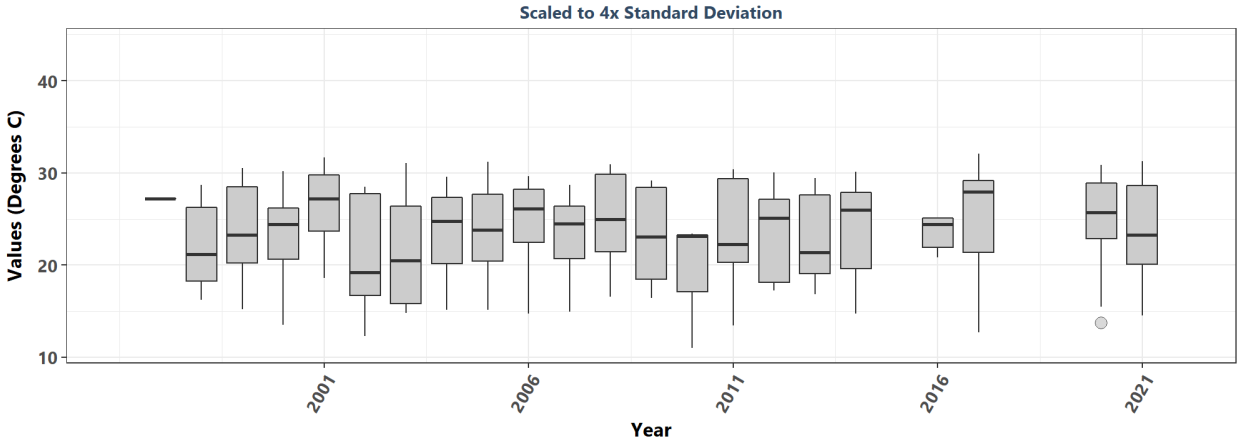

Terra Ceia Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year

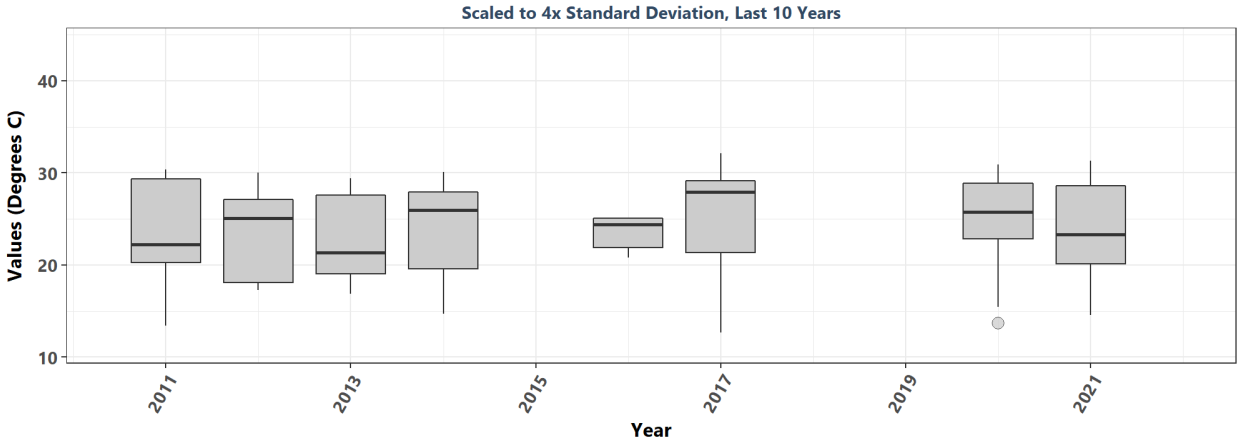

Terra Ceia Aquatic Preserve
By Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
10
20
30
3
6
9
12
Month
Scaled to 4x Standard Deviation
Values (Degrees C)
10
20
30
40
3
6
9
12
Month
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
10
20
30
40
3
6
9
12
Month

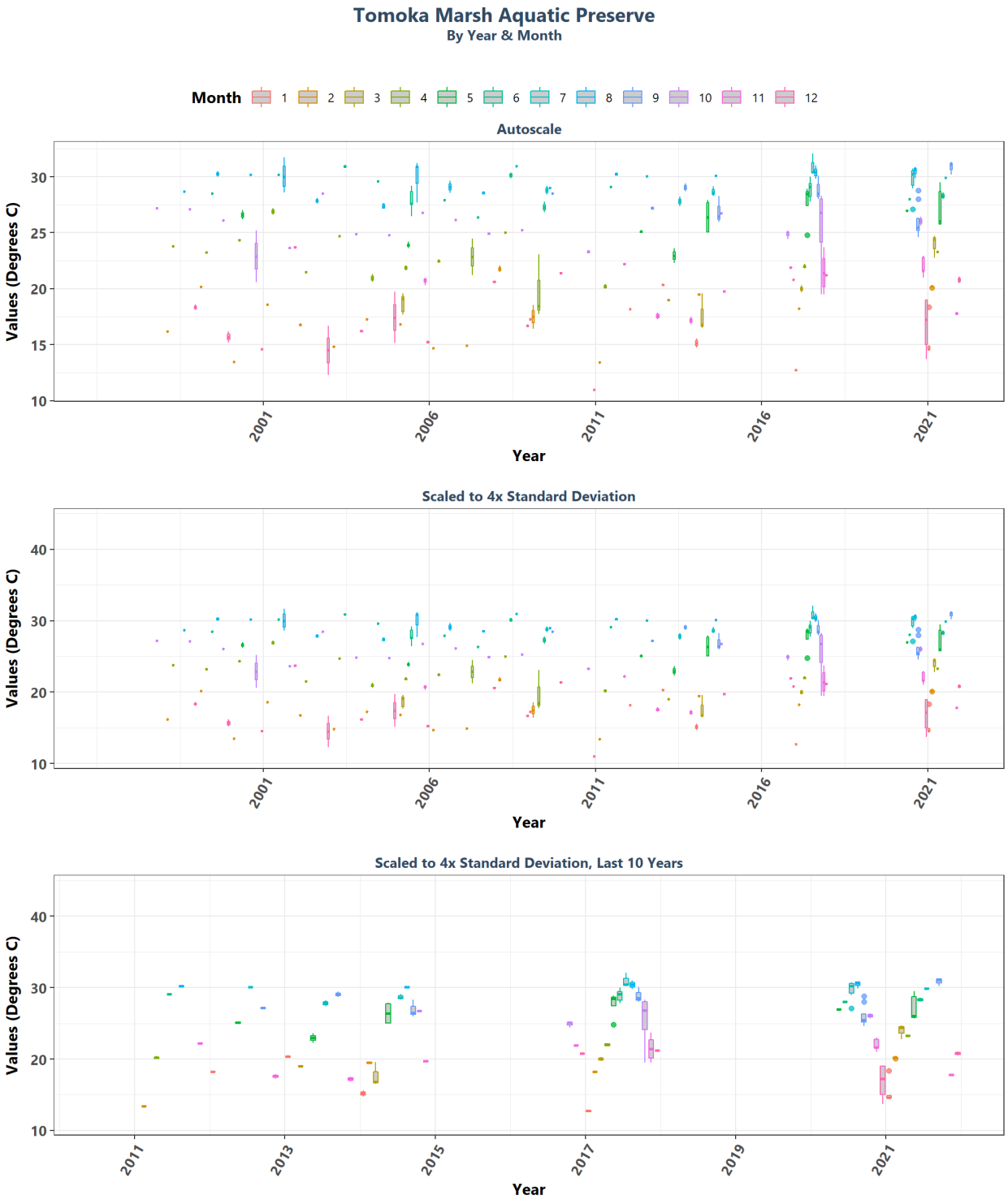
Tomoka Marsh Aquatic Preserve

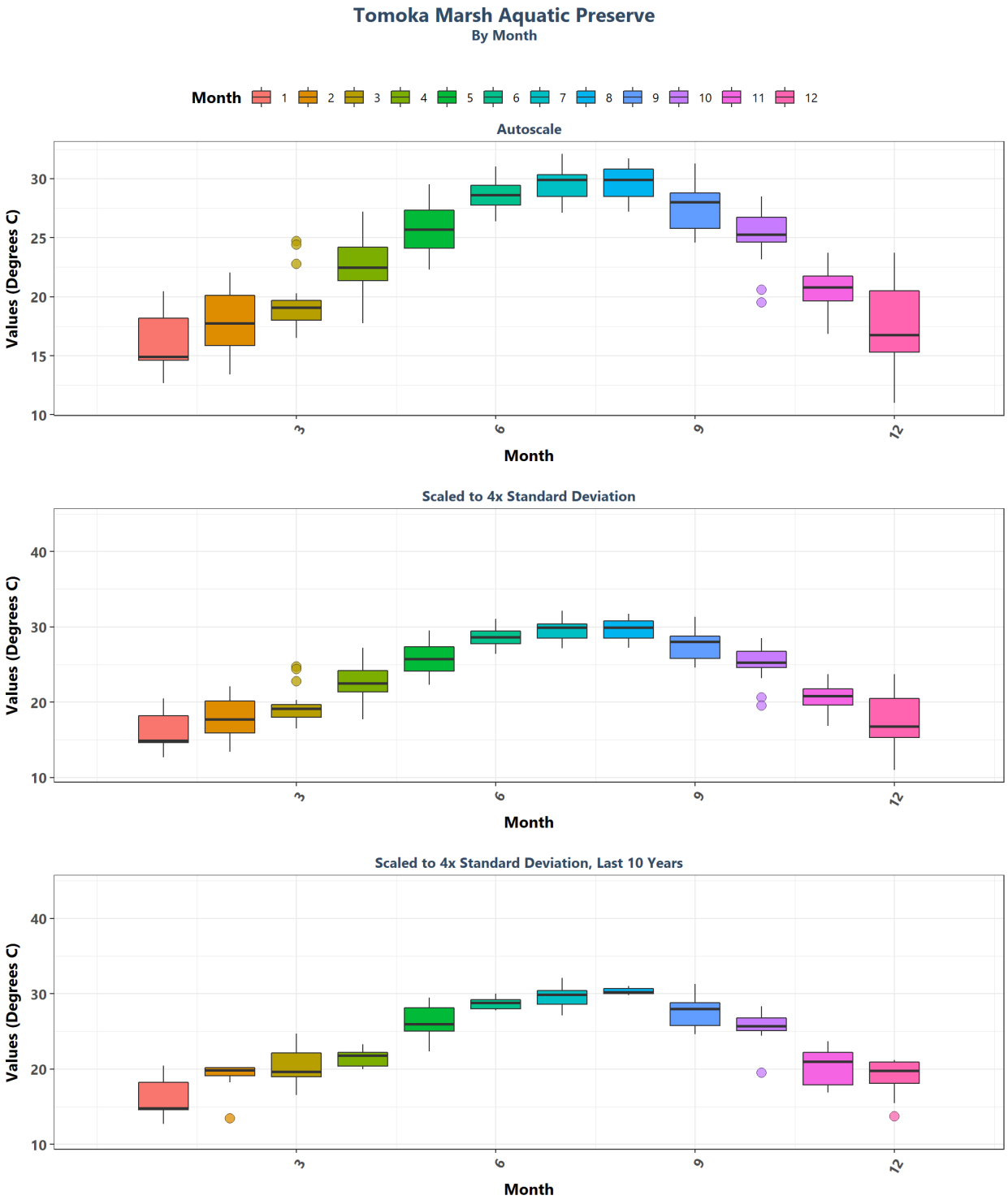
By Year

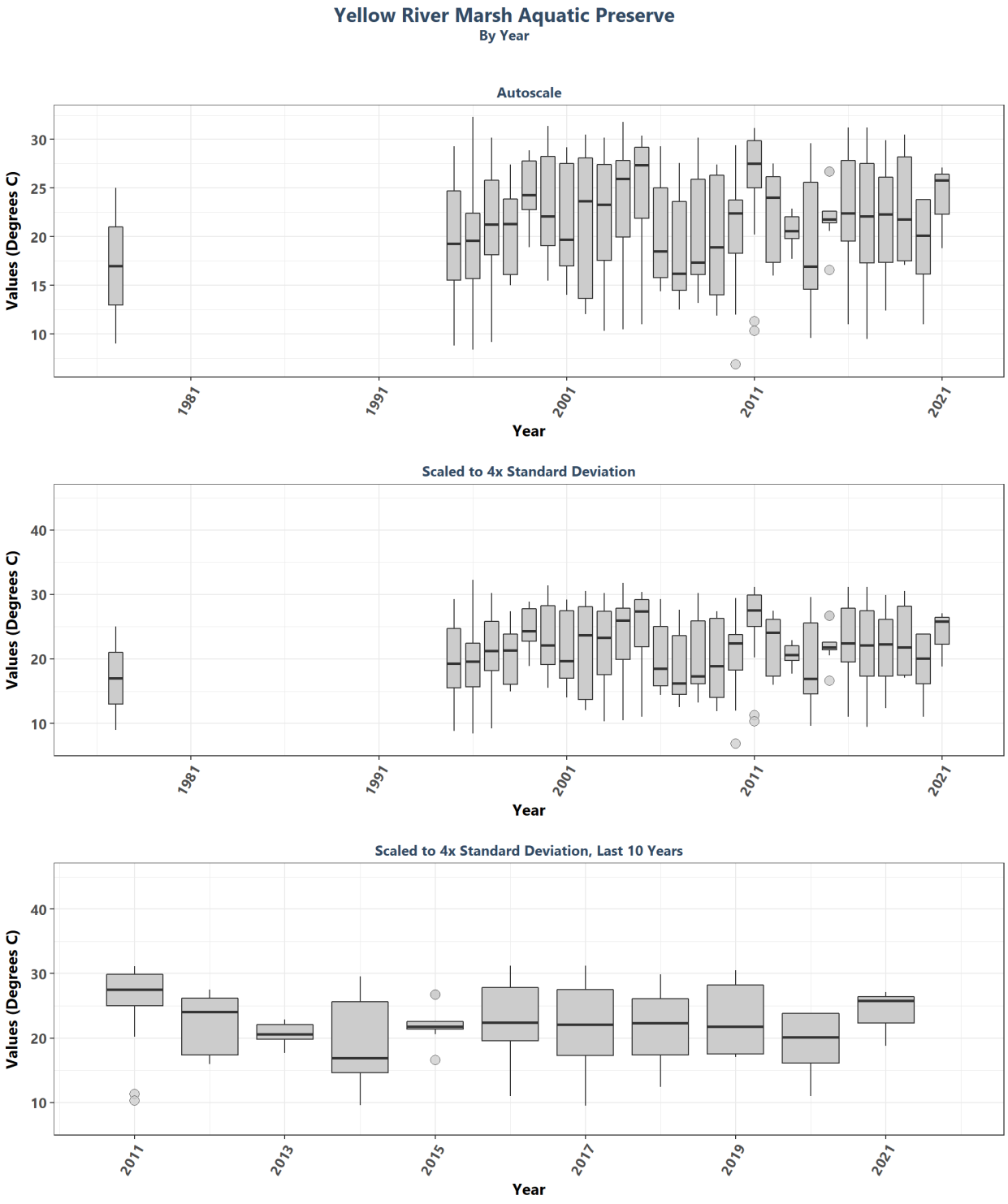


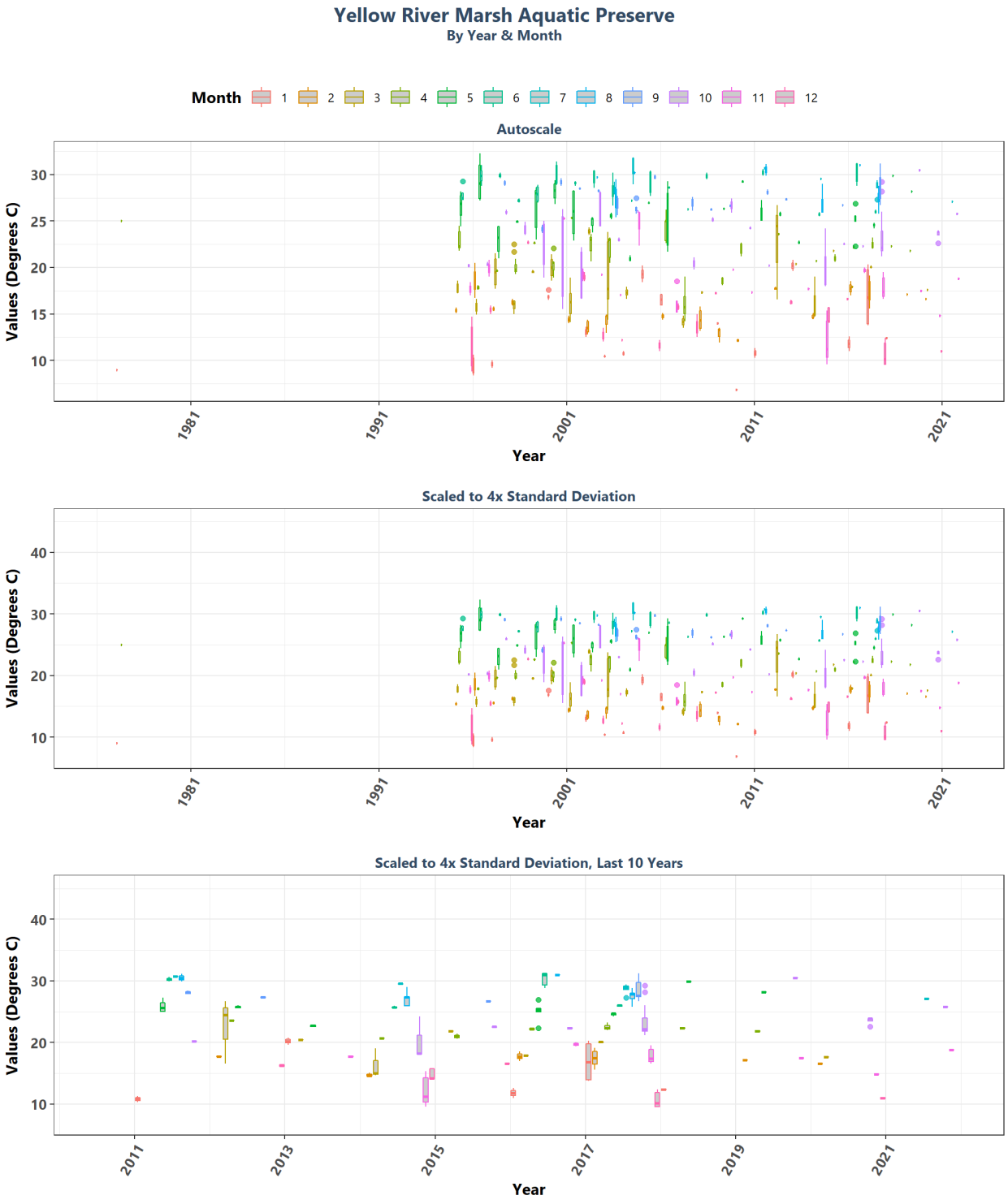


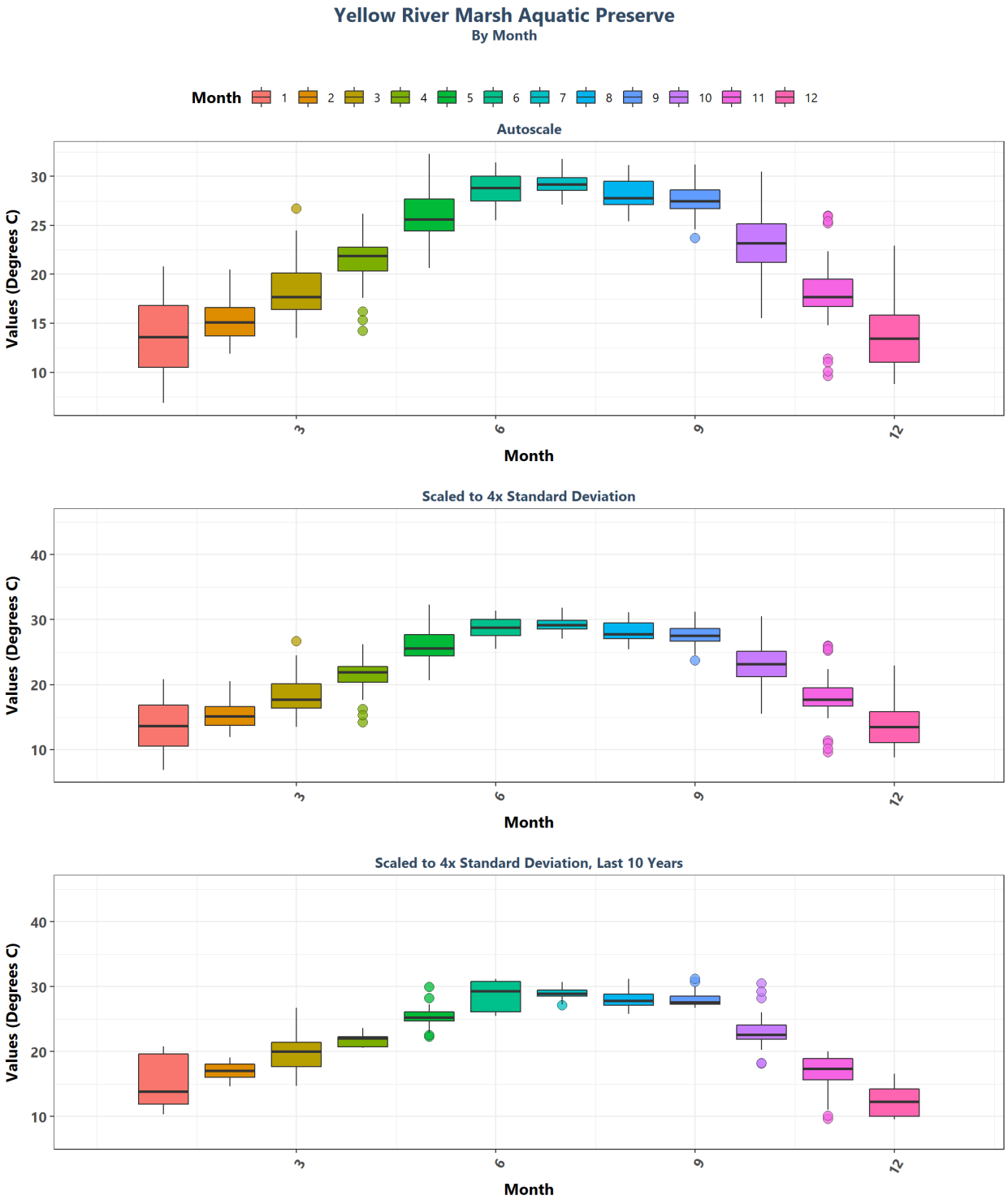



Tomoka Marsh Aquatic Preserve
By Year & Month
Month 1 2 3 4 5 6 7 8 9 10 11 12
Autoscale
Values (Degrees C)
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
Year
Scaled to 4x Standard Deviation, Last 10 Years
Values (Degrees C)
Year


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


Yellow River Marsh Aquatic Preserve
By Year
Autoscale
Values (Degrees C)
30
25
20
15
10
1981
1991
2001
2011
2021
Year
Scaled to 4x Standard Deviation
Values (Degrees C)
40
30
20
10
1981
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


Yellow River Marsh 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




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