

SEACAR Nekton Analysis: Species Richness

Last compiled on 16 June, 2025

Contents

Important Notes	1
Nekton Species Richness	2
Apalachicola Bay Aquatic Preserve	2
Apalachicola National Estuarine Research Reserve	3
Boca Ciega Bay Aquatic Preserve	5
Cape Romano-Ten Thousand Islands Aquatic Preserve	6
Cockroach Bay Aquatic Preserve	8
Lemon Bay Aquatic Preserve	9
Nature Coast Aquatic Preserve	11
Pinellas County Aquatic Preserve	12
Rookery Bay Aquatic Preserve	14
Rookery Bay National Estuarine Research Reserve	15
St. Martins Marsh Aquatic Preserve	17
Terra Ceia Aquatic Preserve	18
Libraries and Settings	19
File Import	20
Data Filtering	20
Managed Area Statistics	21
Appendix I: Managed Area Species Richness	24

Important Notes

The purpose of this script is to determine species richness by gear type and size, create managed area statistics, generate plots, and create reports in pdf and Word document form for Nekton data.

These scripts were created by [J.E. Panzik \(jepanzik@usf.edu\)](mailto:jepanzik@usf.edu) for SEACAR. Updated by [T.G. Hill \(Tyler.Hill@FloridaDEP.gov\)](mailto:Tyler.Hill@FloridaDEP.gov).

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

https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses

This markdown file is designed to be compiled by `Nekton_SpeciesRichness_ReportRender.R` (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Nekton/Nekton_SpeciesRichness_ReportRender.R).

Details on the determination of catch per unit effort can be found in the document [SEACAR Nekton catch per unit effort.pdf](https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Nekton/SEACAR%20Nekton%20catch%20per%20unit%20effort.pdf) (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Nekton/SEACAR%20Nekton%20catch%20per%20unit%20effort.pdf).

Nekton Species Richness

Apalachicola Bay Aquatic Preserve

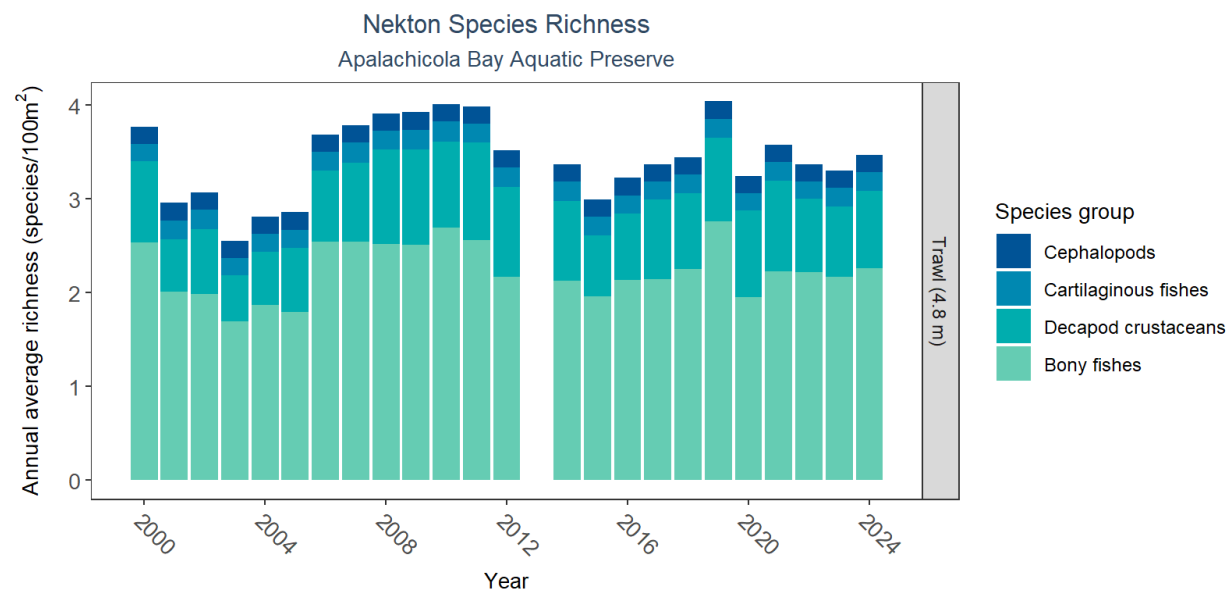
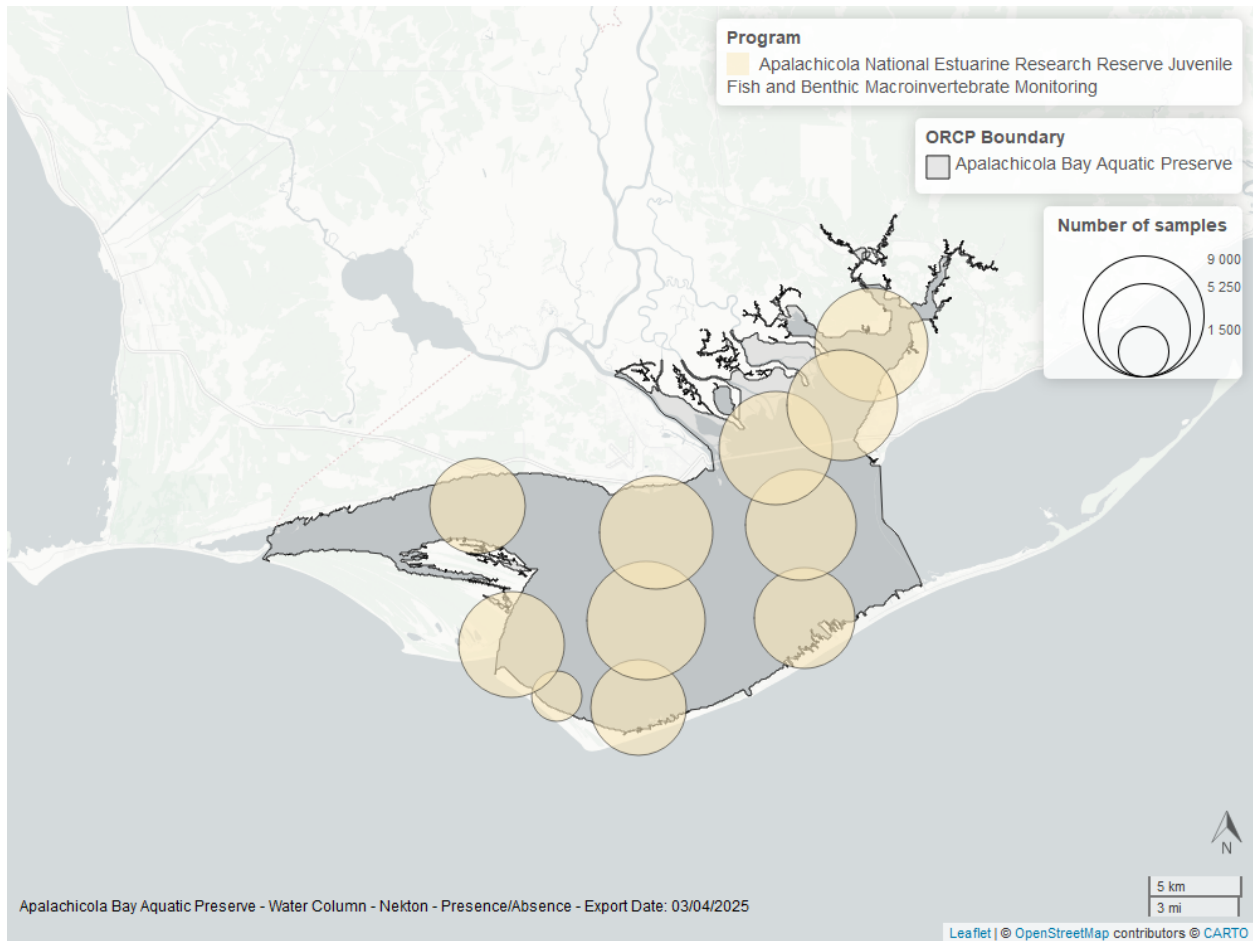


Table 1: Nekton Species Richness - Apalachicola Bay Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	4.8	24	2000	2024	4967	0.19	5.56	0.74	1.13	1.05	2015	2010



Apalachicola National Estuarine Research Reserve

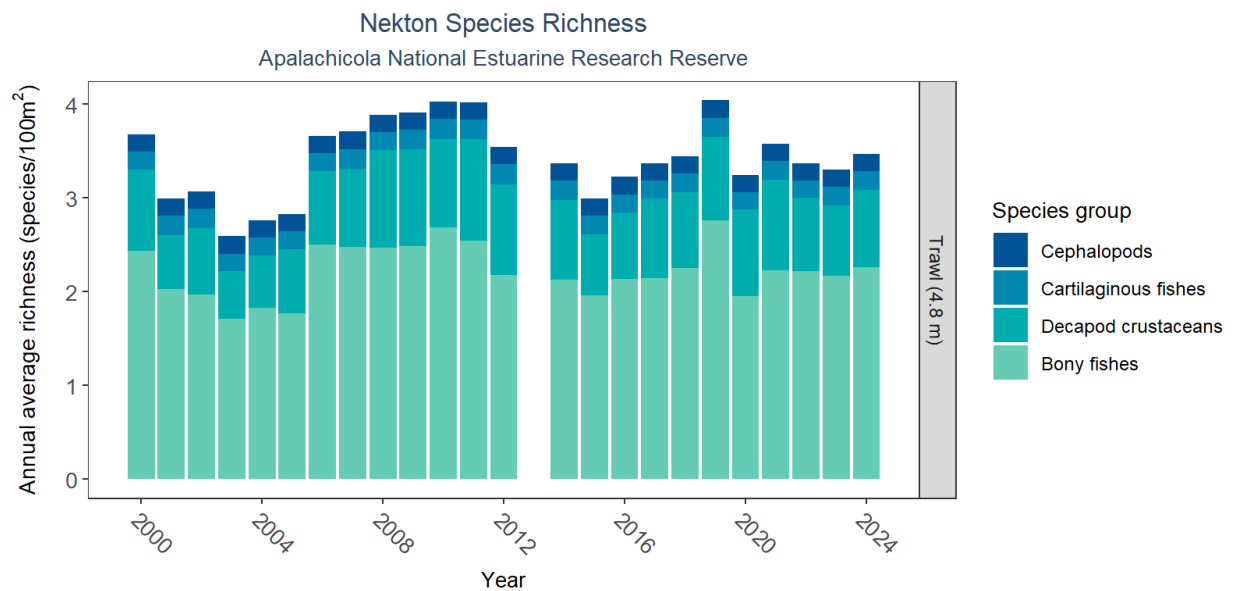
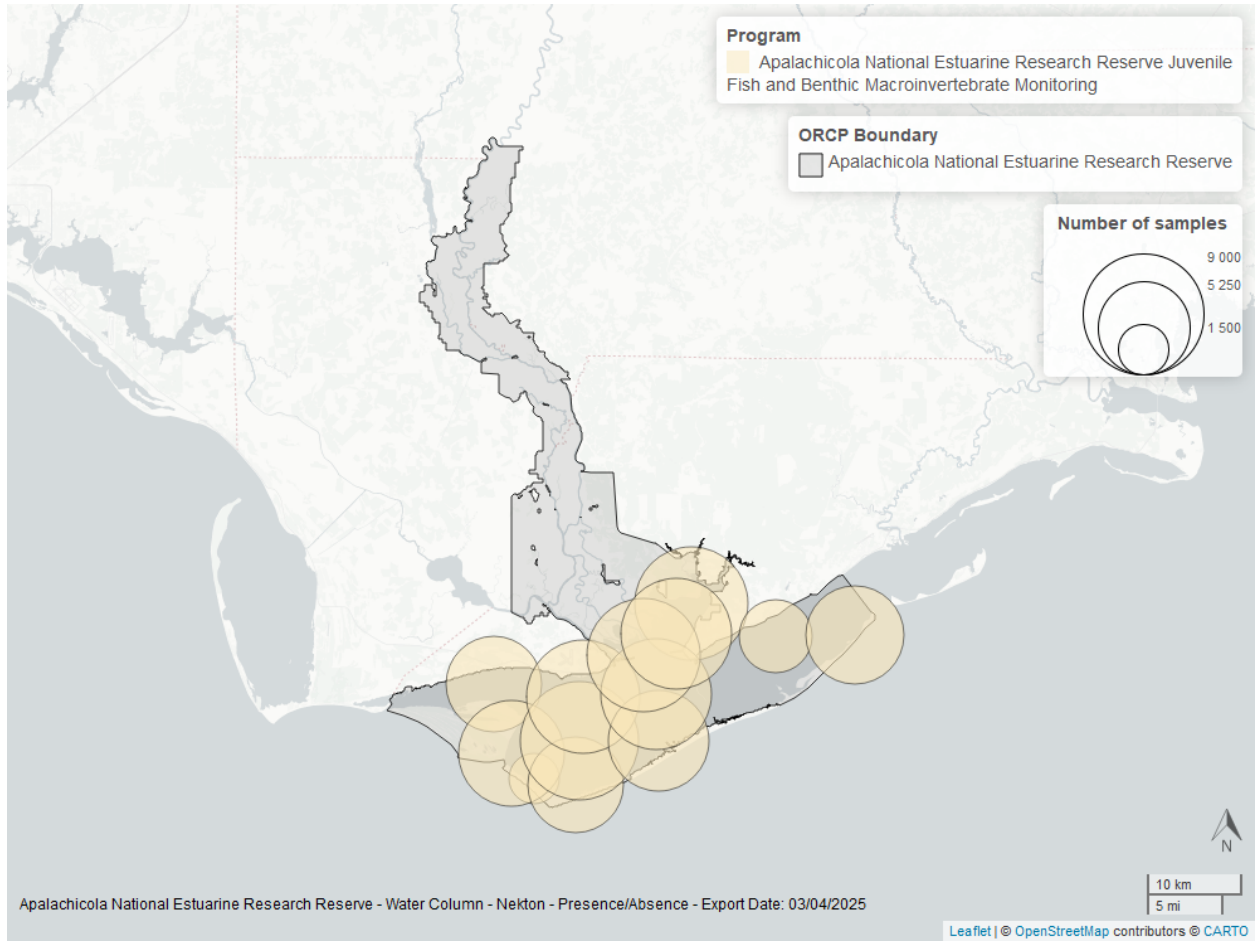


Table 2: Nekton Species Richness - Apalachicola National Estuarine Research Reserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	4.8	24	2000	2024	5685	0.19	5.93	0.74	1.13	1.06	2015	2010



Boca Ciega Bay Aquatic Preserve

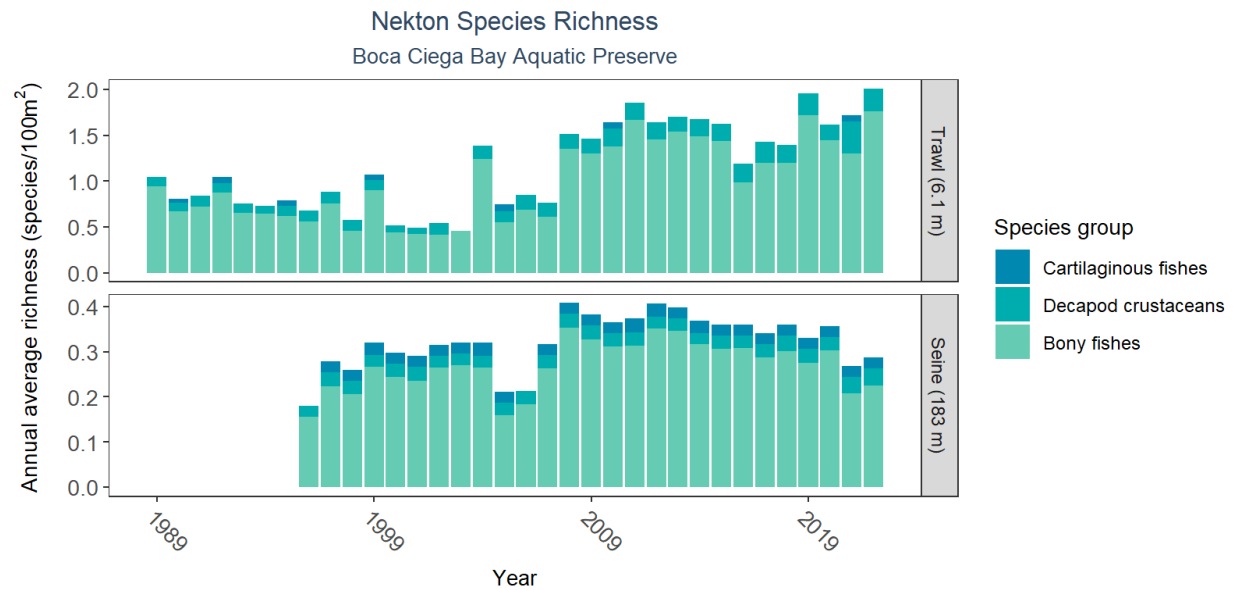
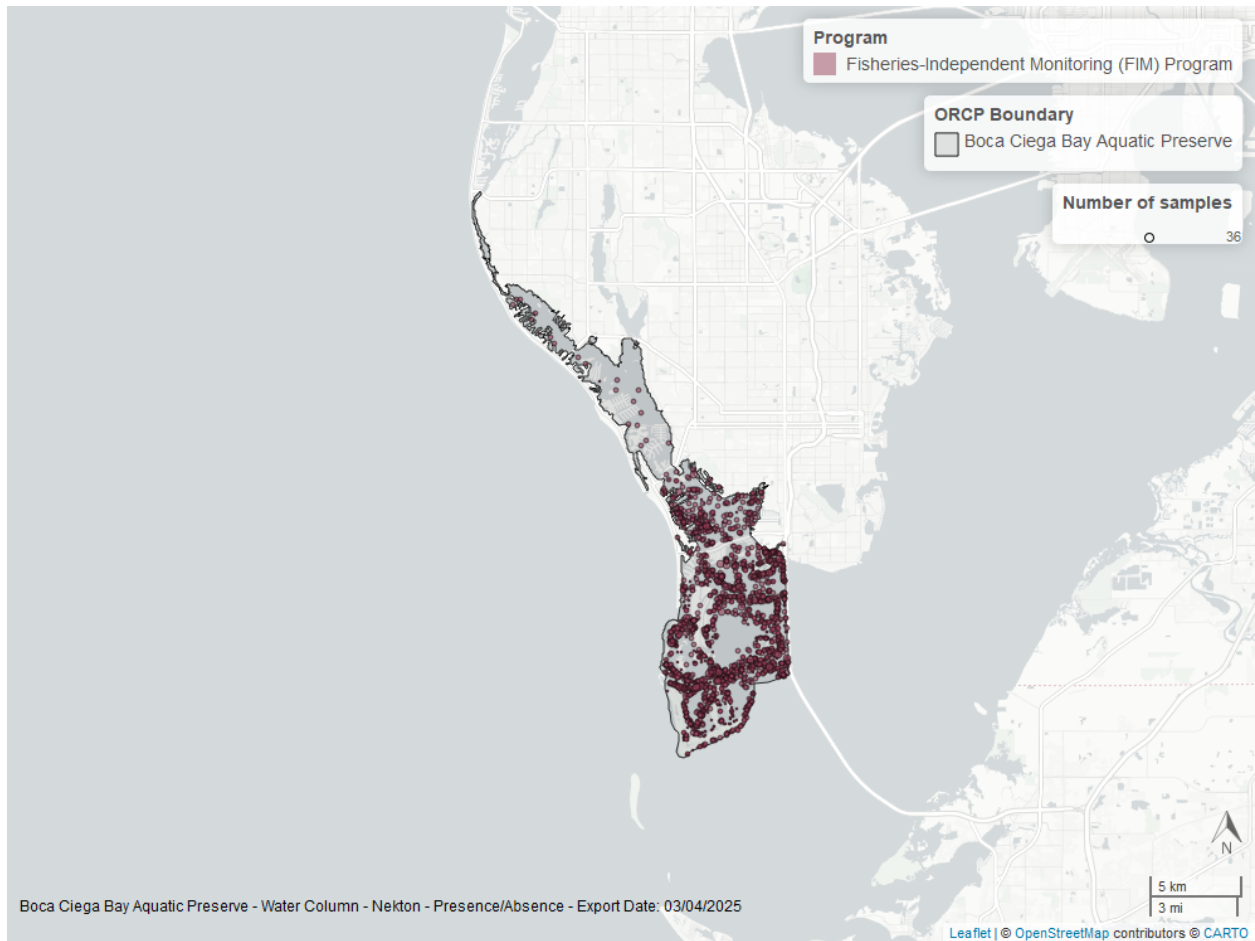


Table 3: Nekton Species Richness - Boca Ciega Bay Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	34	1989	2022	1680	0.04	3.78	0.43	0.74	0.79	2000	2022
Seine	183.0	27	1996	2022	1642	0.02	0.85	0.16	0.19	0.17	2005	2008



Cape Romano-Ten Thousand Islands Aquatic Preserve

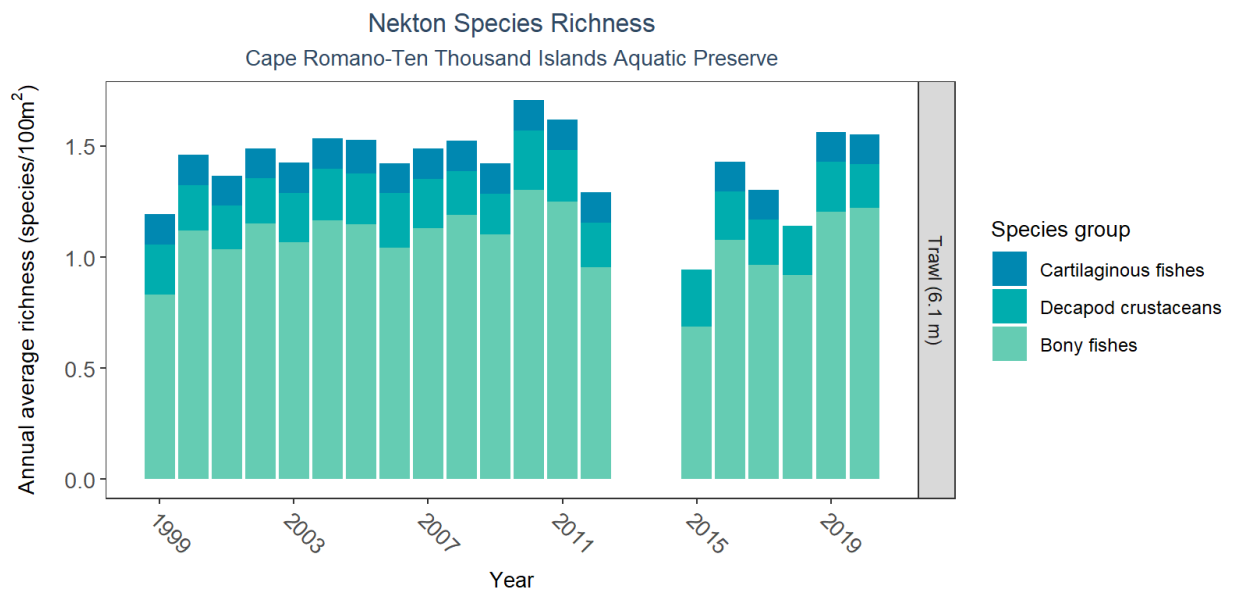
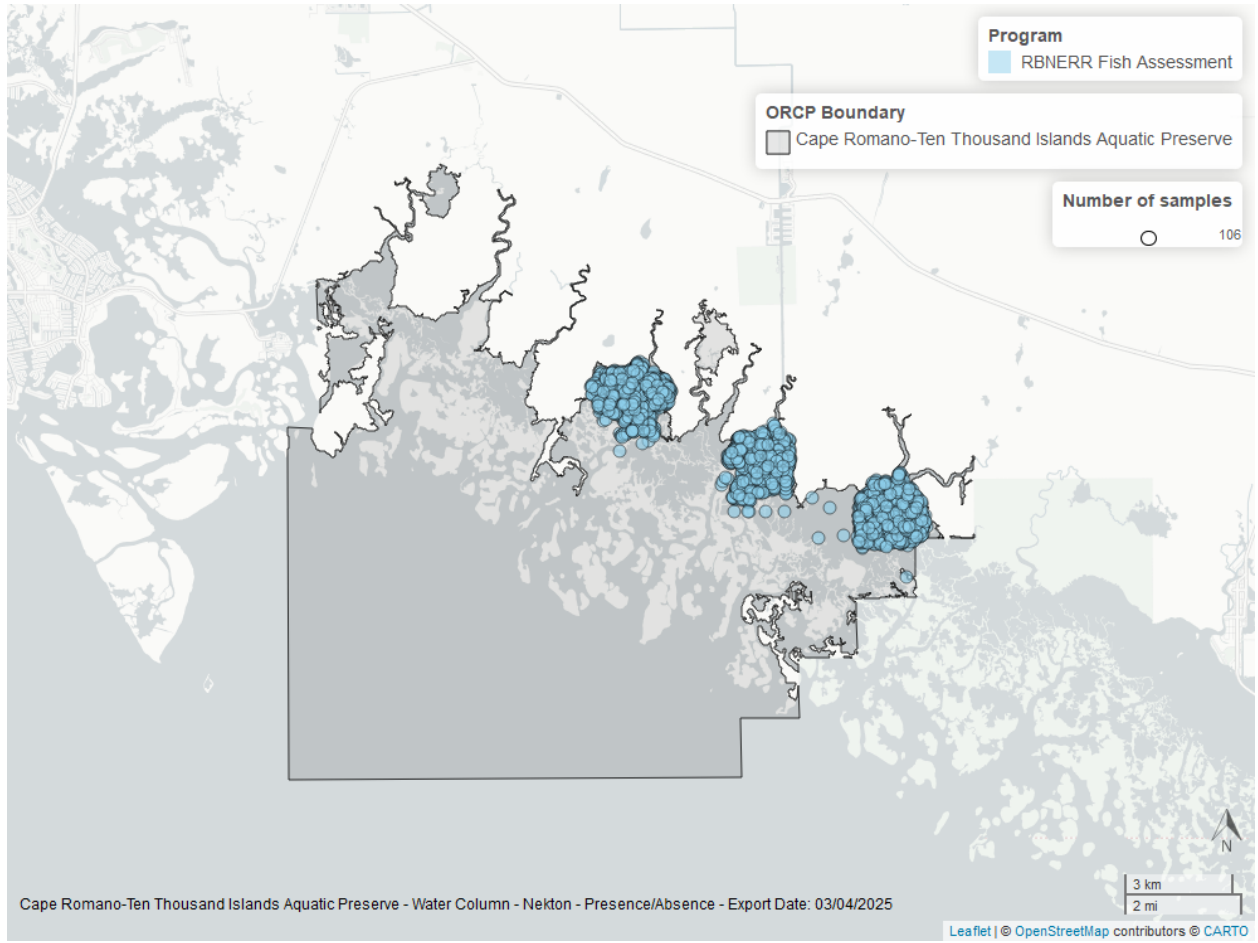


Table 4: Nekton Species Richness - Cape Romano-Ten Thousand Islands Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	20	1999	2020	5002	0.13	3.24	0.4	0.67	0.57	2015	2010



Cockroach Bay Aquatic Preserve

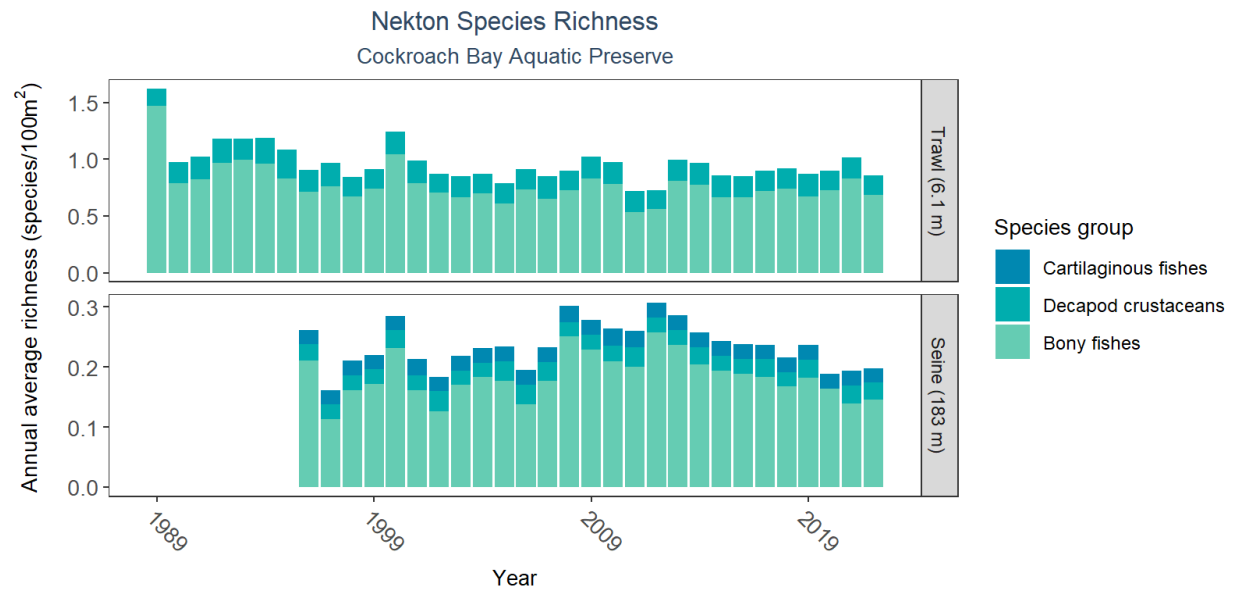
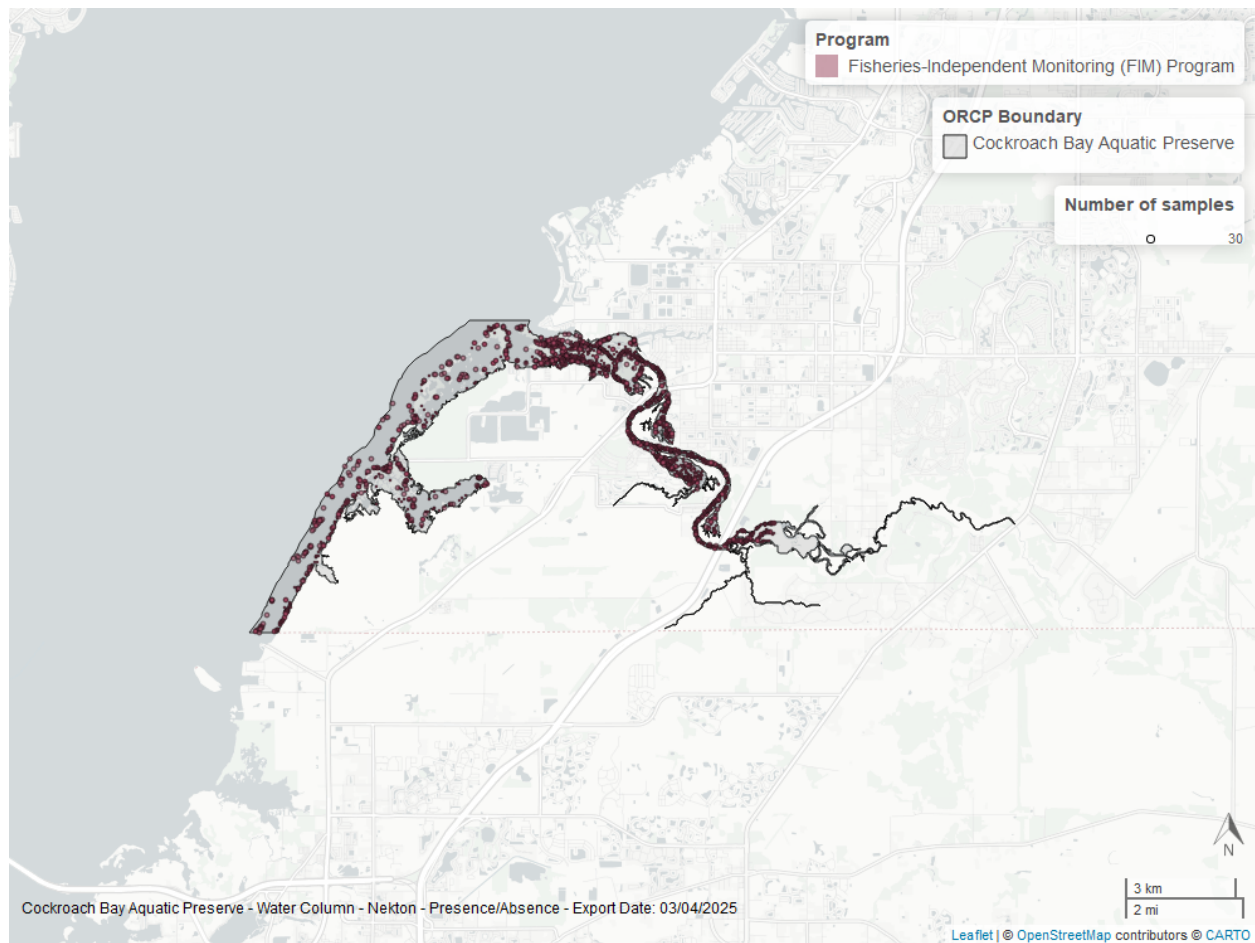


Table 5: Nekton Species Richness - Cockroach Bay Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	34	1989	2022	2832	0.07	3.30	0.3	0.50	0.45	2011	1989
Seine	183.0	27	1996	2022	789	0.02	0.63	0.1	0.13	0.11	1997	2012



Lemon Bay Aquatic Preserve

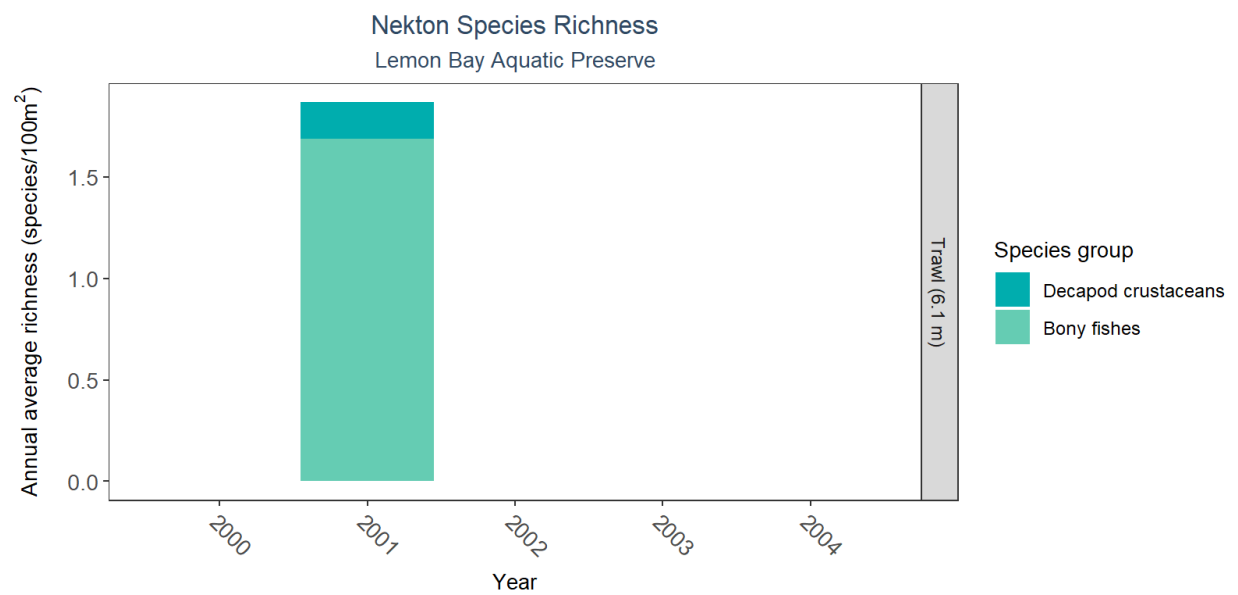
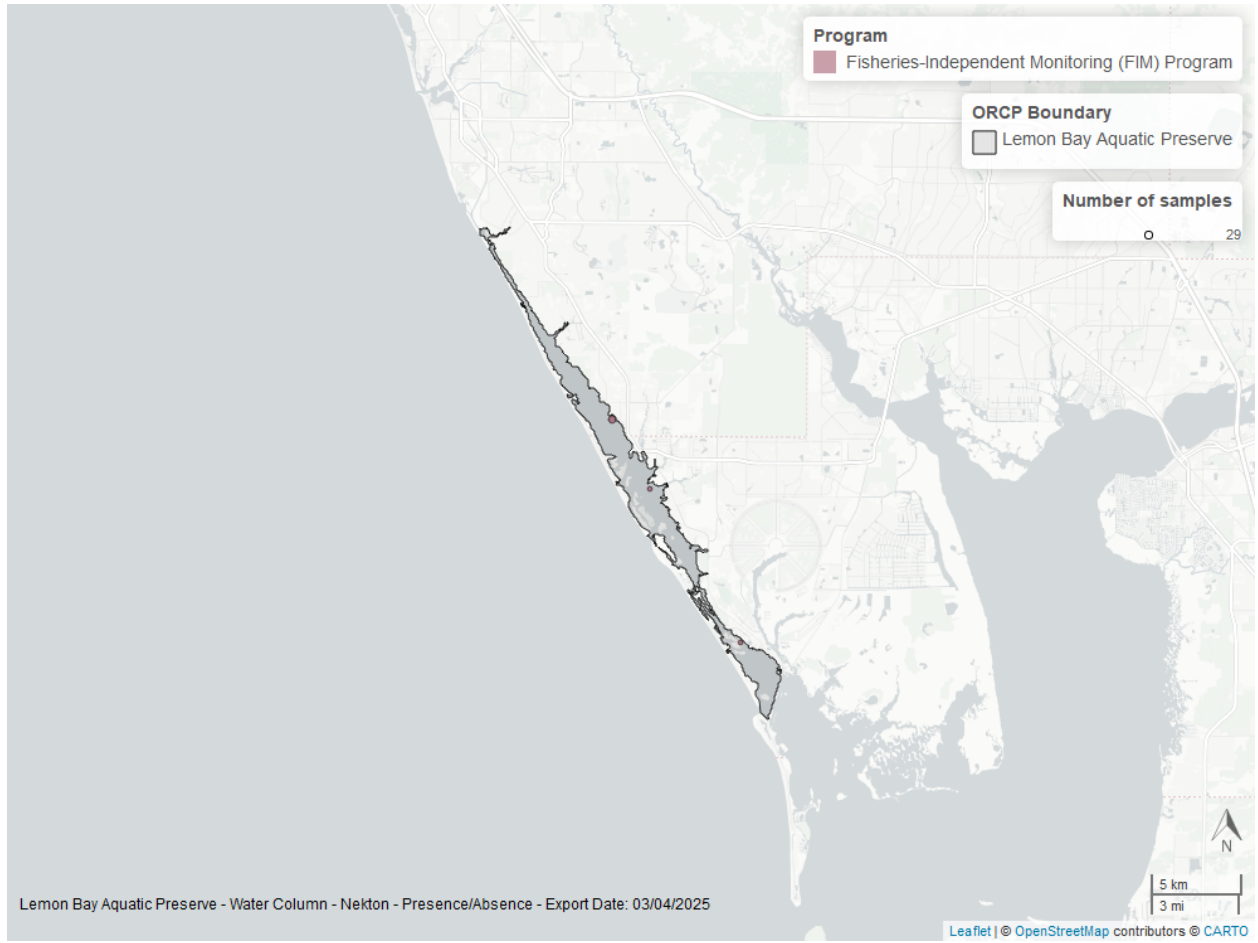


Table 6: Nekton Species Richness - Lemon Bay Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	1	2001	2001	6	0.13	2.02	0.74	0.93	0.87	2001	2001



Nature Coast Aquatic Preserve

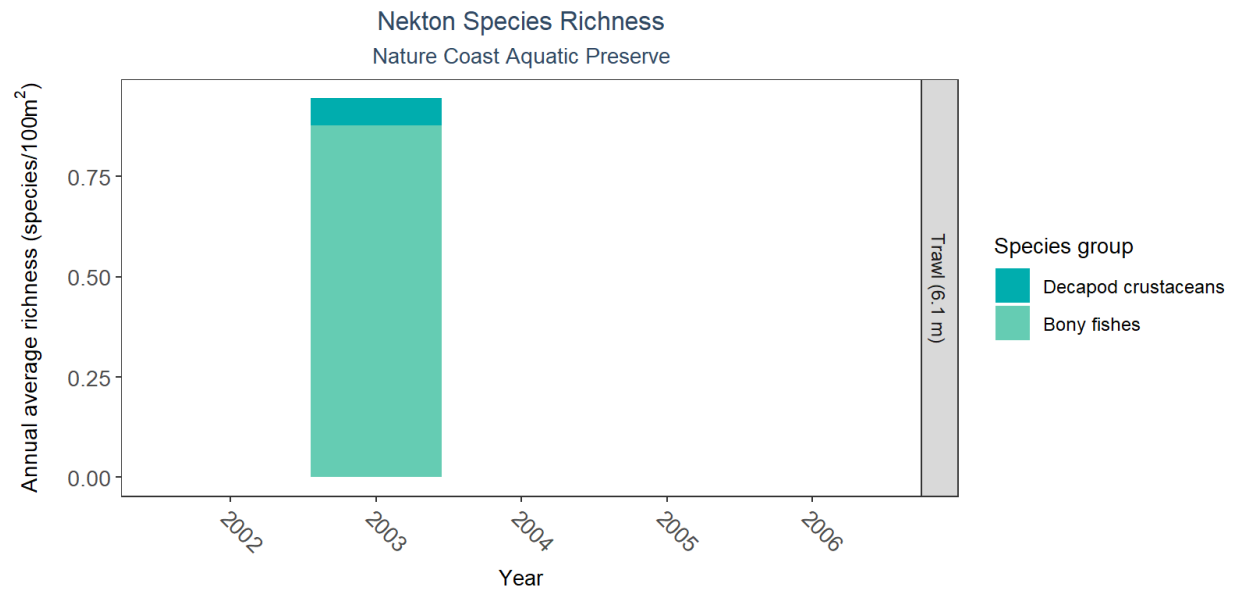
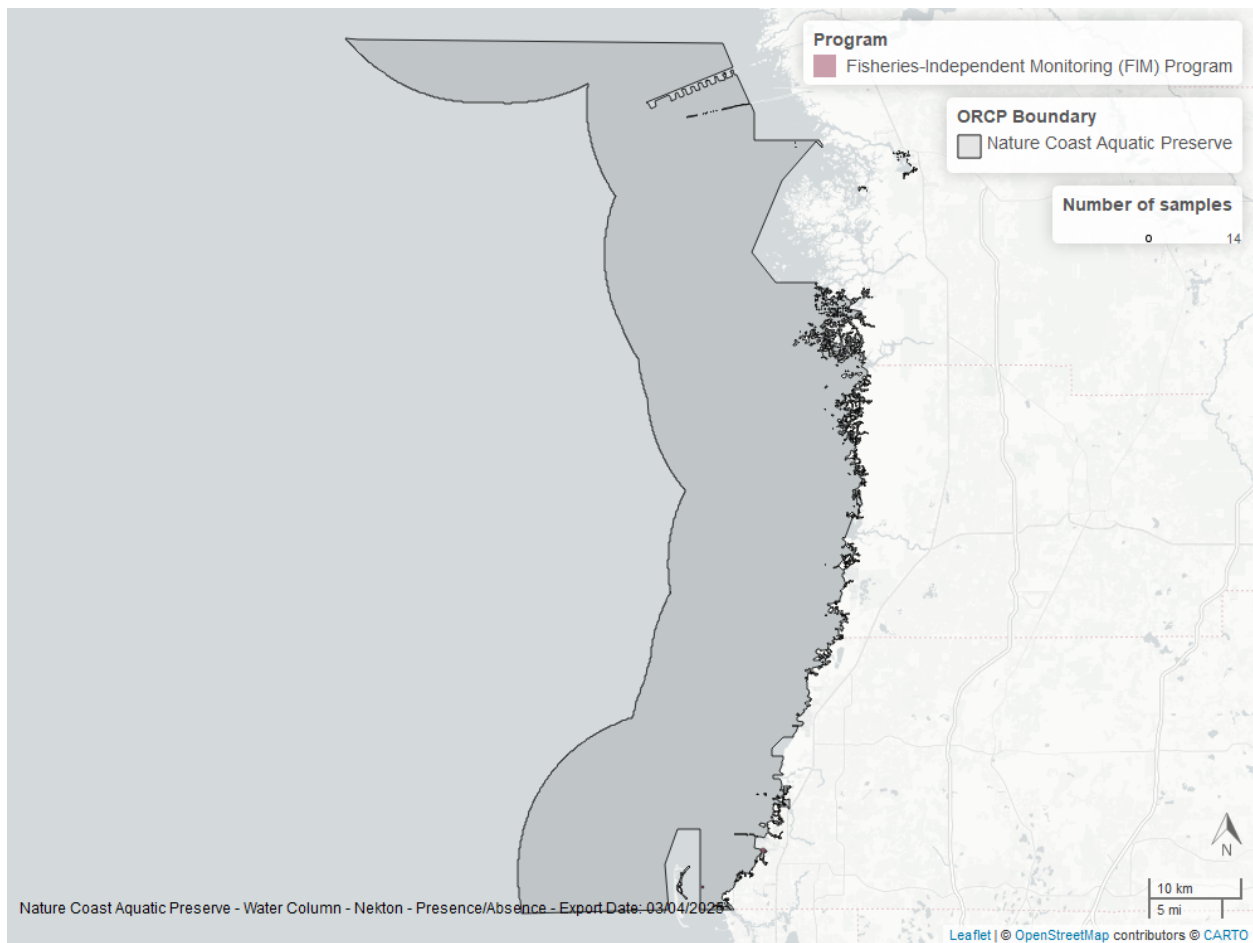


Table 7: Nekton Species Richness - Nature Coast Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	1	2003	2003	2	0.07	0.88	0.47	0.47	0.57	2003	2003



Pinellas County Aquatic Preserve

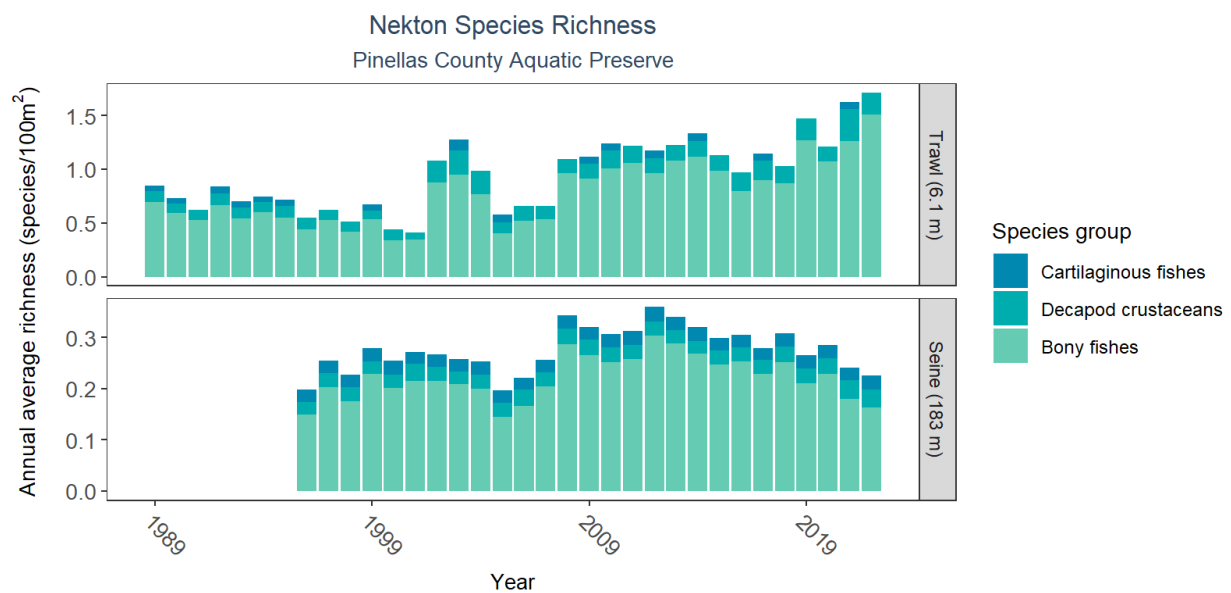
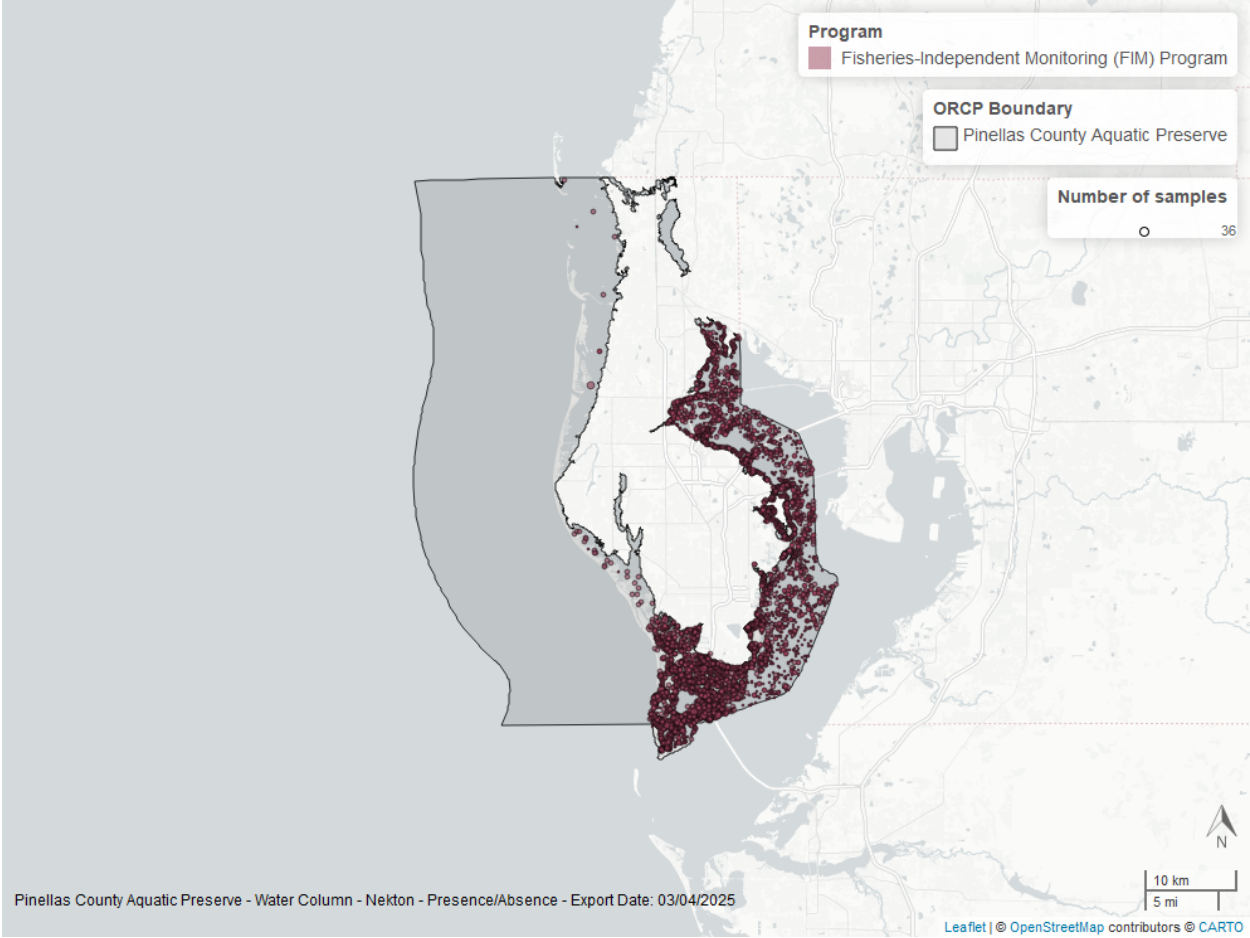


Table 8: Nekton Species Richness - Pinellas County Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	34	1989	2022	5254	0.03	4.86	0.3	0.54	0.63	2001	2022
Seine	183.0	27	1996	2022	4407	0.02	0.85	0.1	0.15	0.15	2005	2013



Rookery Bay Aquatic Preserve

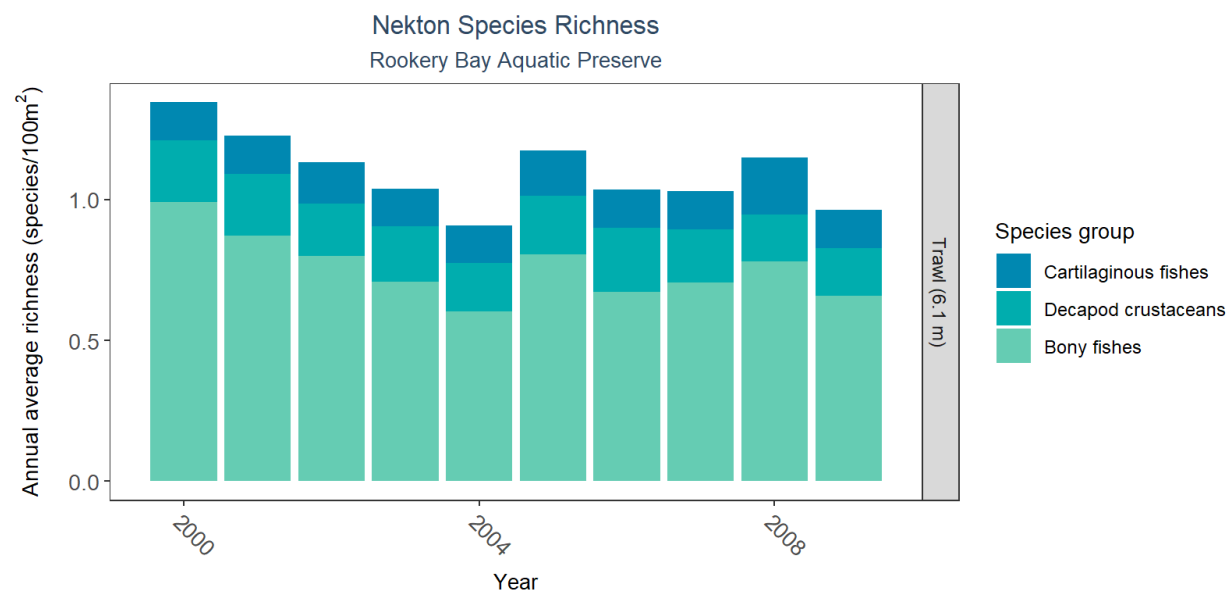
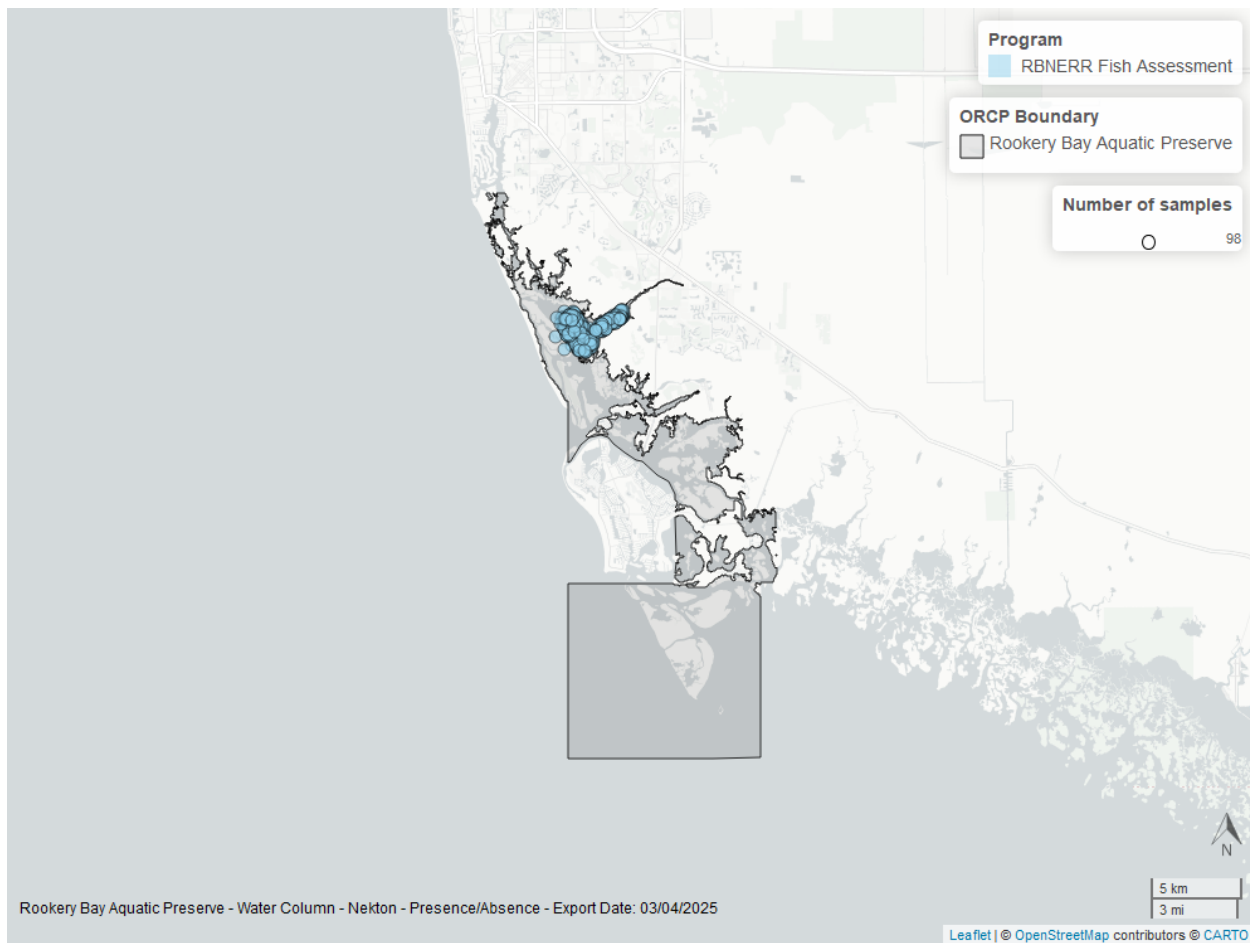


Table 9: Nekton Species Richness - Rookery Bay Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	10	2000	2009	1052	0.13	2.16	0.27	0.48	0.39	2004	2000



Rookery Bay National Estuarine Research Reserve

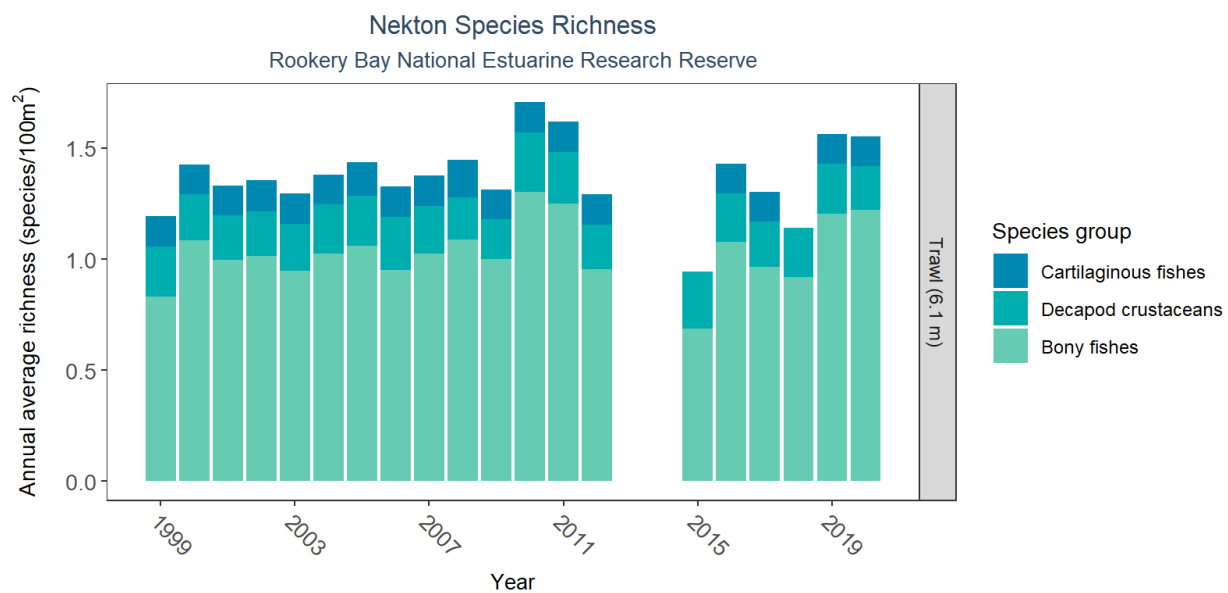
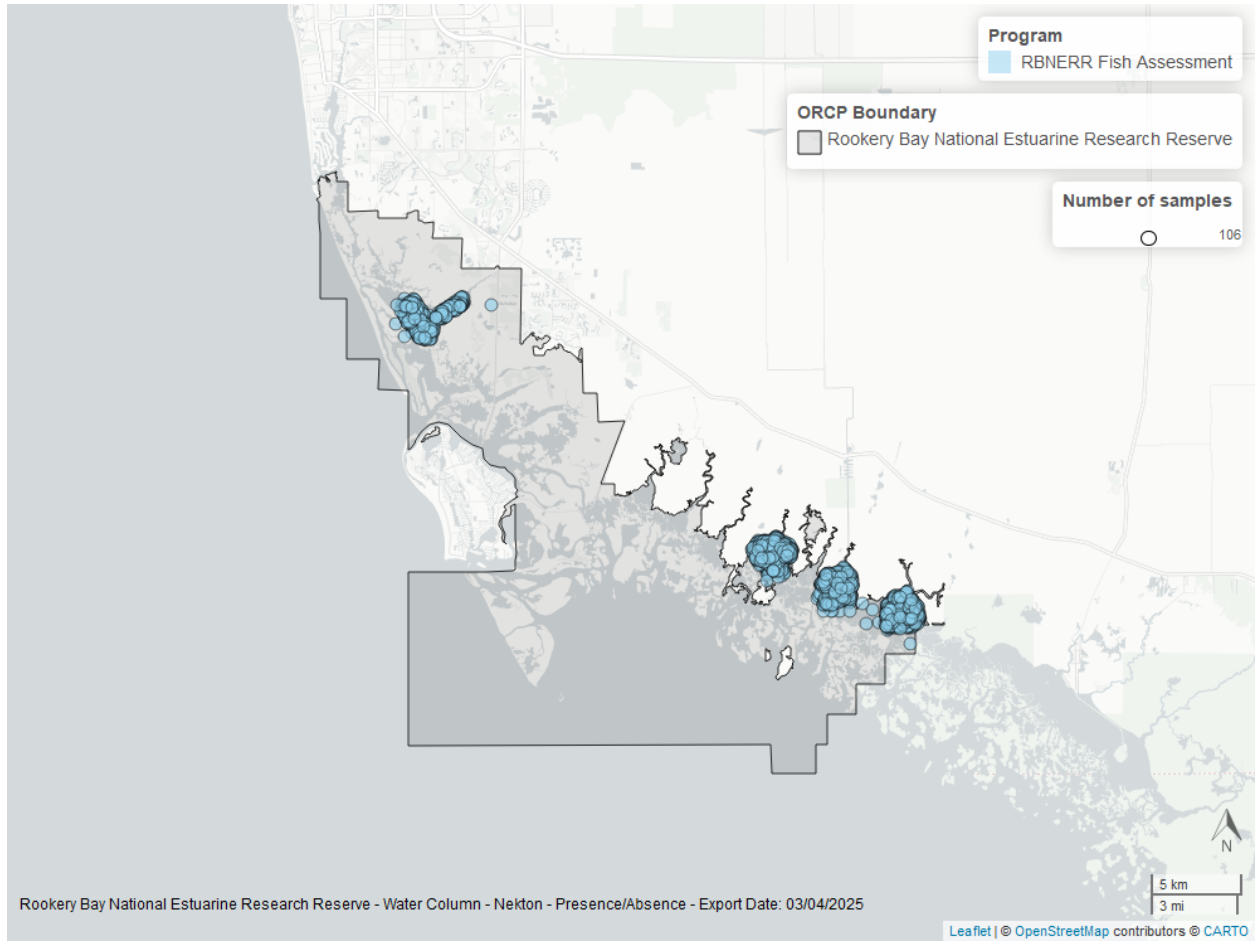


Table 10: Nekton Species Richness - Rookery Bay National Estuarine Research Reserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	20	1999	2020	6071	0.13	3.24	0.27	0.63	0.55	2015	2010



St. Martins Marsh Aquatic Preserve

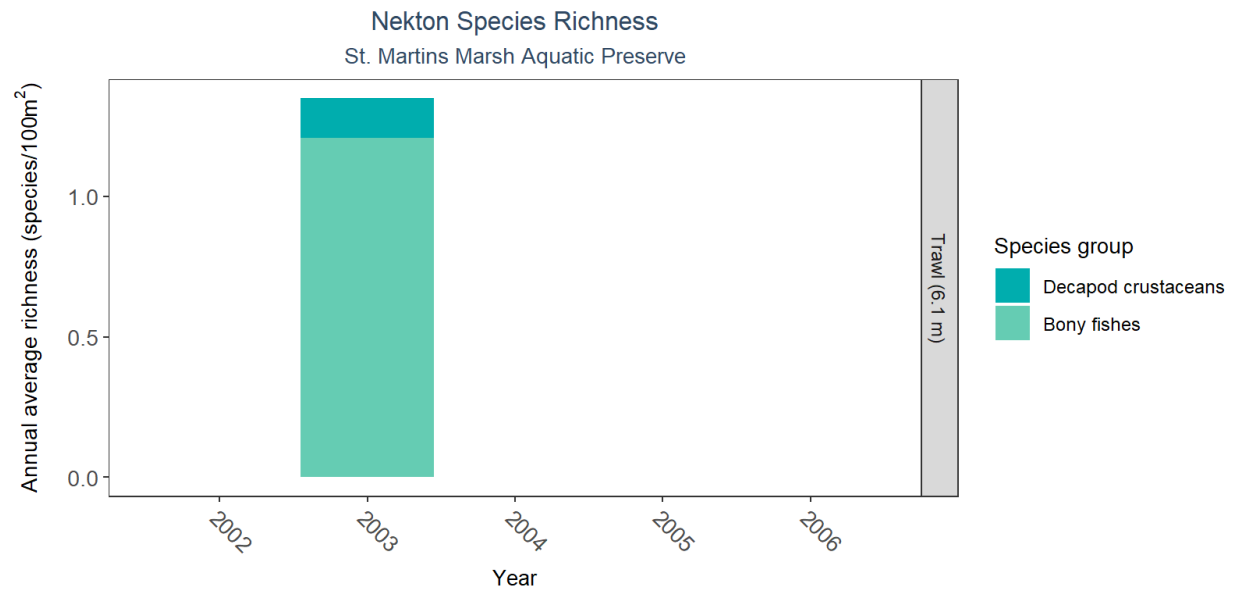
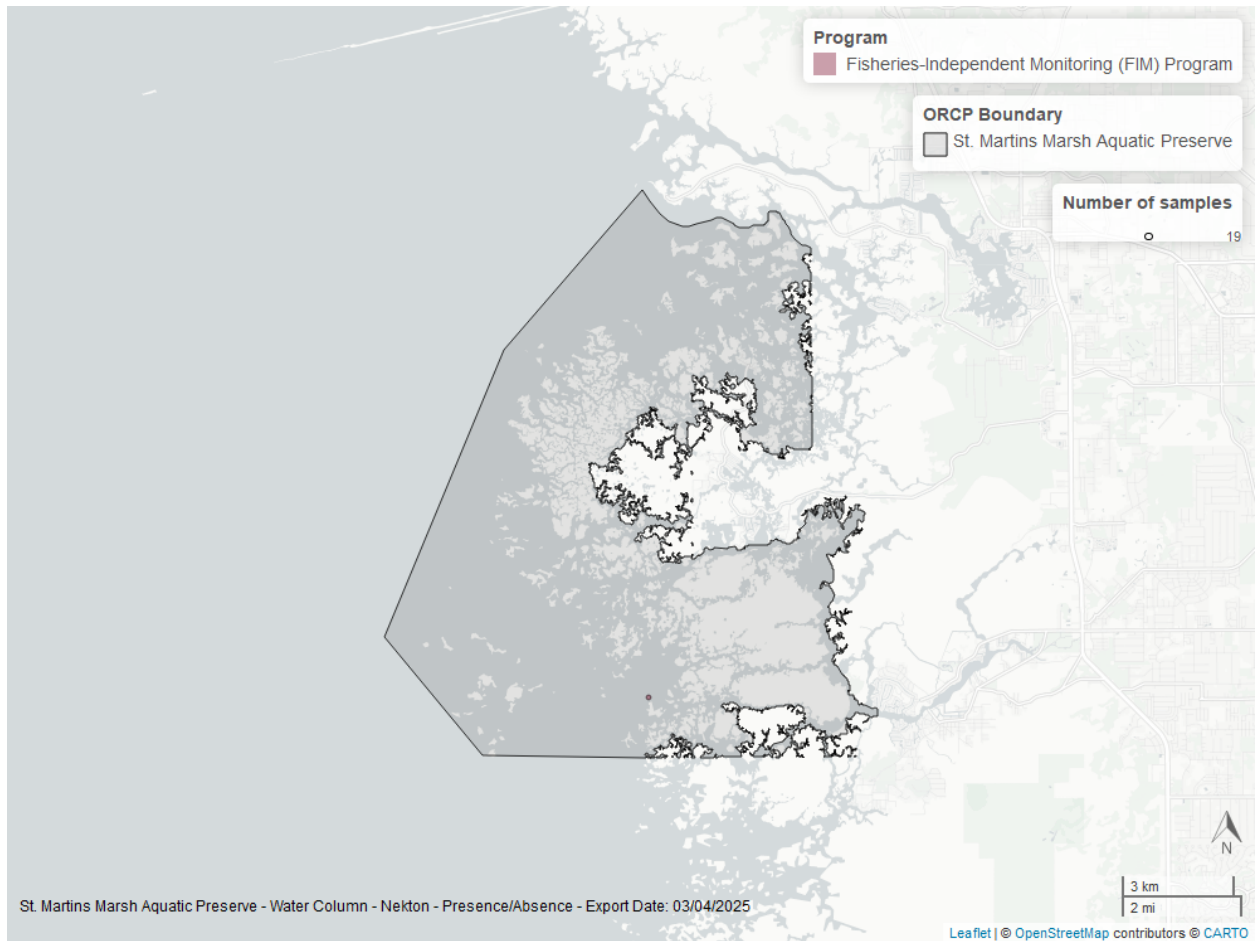


Table 11: Nekton Species Richness - St. Martins Marsh Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	1	2003	2003	2	0.14	1.21	0.67	0.67	0.75	2003	2003



Terra Ceia Aquatic Preserve

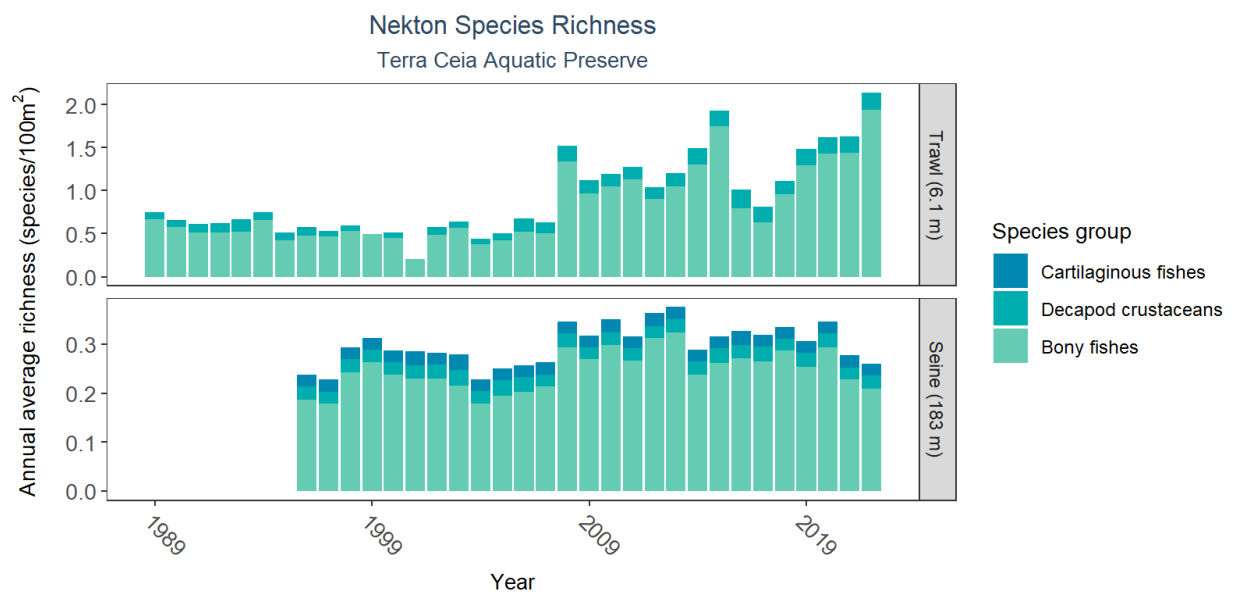
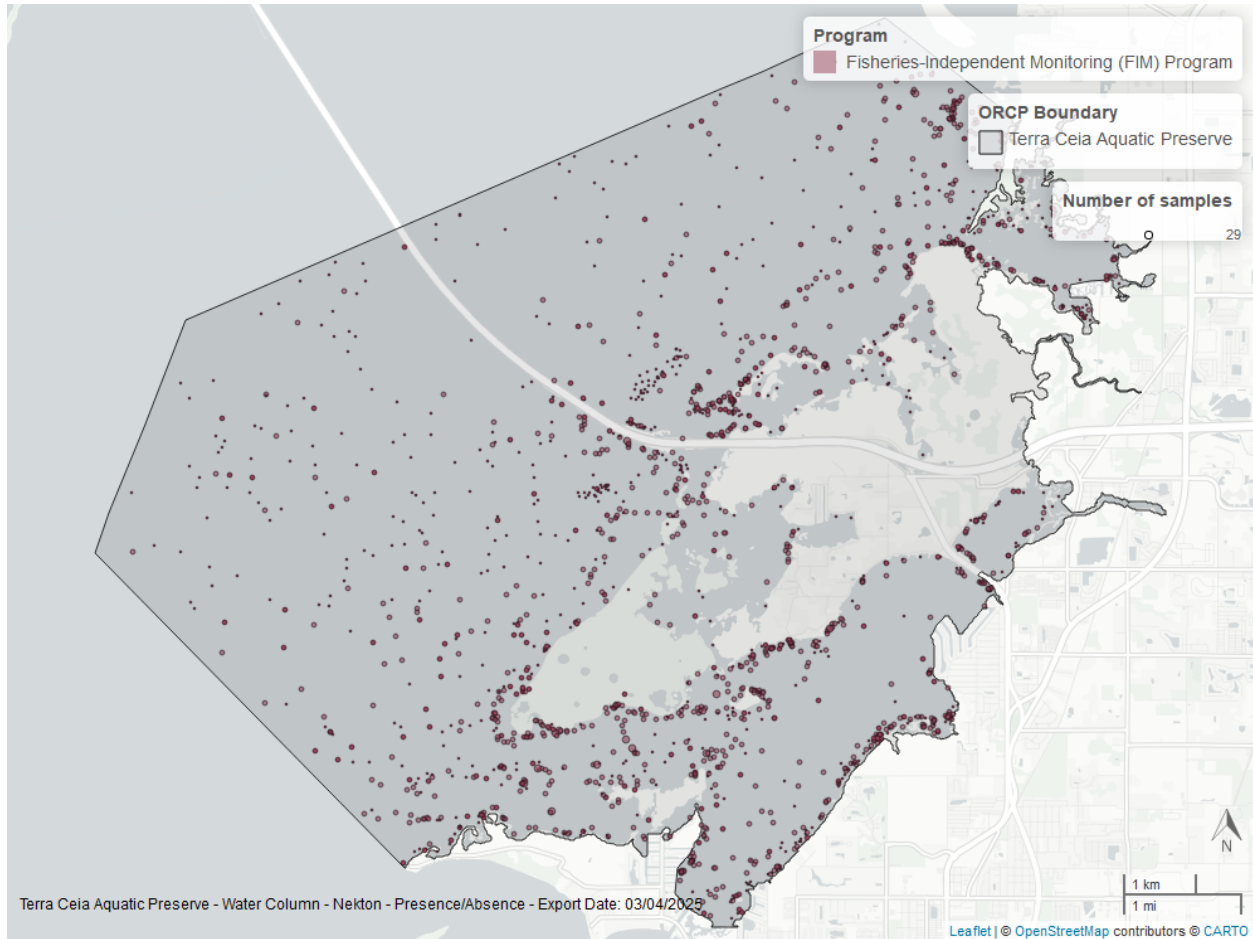


Table 12: Nekton Species Richness - Terra Ceia Aquatic Preserve

GearType	GearSize-m	N-Years	EarliestYear	LatestYear	N-Data	Min	Max	Median	Mean	StDev	Year-MinRichness	Year-MaxRichness
Trawl	6.1	34	1989	2022	1280	0.04	3.88	0.30	0.62	0.73	2001	2022
Seine	183.0	27	1996	2022	1449	0.02	0.66	0.17	0.17	0.14	2004	2012



Libraries and Settings

Loads libraries used in the script. The inclusion of `scipen` option limits how frequently R defaults to scientific notation. Sets default settings for displaying warning and messages in created document, and sets figure dpi.

```
library(knitr)
library(data.table)
library(dplyr)
library(lubridate)
library(ggplot2)
library(scales)
library(tidyr)
library(gridExtra)
#library(tidyverse)
library(ggpubr)
library(scales)
library(glue)
```

```
options(scipen=999)
knitr::opts_chunk$set(
  warning=FALSE,
  message=FALSE,
  dpi=200
)
```

File Import

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

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

The script then gets the name of the parameter as it appears in the data file and units of the parameter.

The latest version of Nekton data is available at: <https://usf.box.com/s/35sn0n0lrrxi9dtkik030nozbnvj9dyj>

The file being used for the analysis is: **All_CW_Parameters-2025-Mar-06.txt**

```
#Import data from nekton file
data <- fread(file_in, sep="|", header=TRUE, stringsAsFactors=FALSE,
              na.strings=c("NULL"))

cat(paste("The data file used is:", file_short, sep="\n"))
```

Data Filtering

Documentation on database filtering is provided here: [SEACAR Documentation- Analysis Filters and Calculations.pdf](https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/SEACAR%20Documentation%20-%20Analysis%20Filters%20and%20Calculations.pdf) (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/SEACAR%20Documentation%20-%20Analysis%20Filters%20and%20Calculations.pdf).

Imported data is initially filtered to only contain the parameter of interest.

The other filtering performed by the script at this point removes rows that are missing values for `ResultValue` and `EffortCorrection_100m2`, and removes any `EffortCorrection_100m2` that is 0 because it will cause an infinite number when determining Species Richness.

A group of unique `ManagedAreaName`, `ProgramID`, `ProgramName`, `ProgramLocationID`, `SampleDate`, and `GearSize_m` are being considered a “reference” for measurement. For each “reference”, the number of observed species is summed and then divided by the `EffortCorrection_100m2` to determine the Species Richness per 100 square meters.

The species richness data is then written to a file. And the list of Managed Areas with observations is stored.

```
# Filter data for the desired parameter
data <- data[ParameterName==param_name, ]

if(param_name=="Presence/Absence"){
  parameter <- "Species Richness"
}

# Makes sure EffortCorrection is numeric value
data$EffortCorrection_100m2 <- as.numeric(data$EffortCorrection_100m2)

# Remove any data with missing EffortCorrection values
data <- data[!is.na(data$EffortCorrection_100m2),]
```

```

# Only keep data that has non-zero EffortCorrection values
data <- data[data$EffortCorrection_100m2!=0,]

# Remove any data with missing ResultValue entries
data <- data[!is.na(data$ResultValue),]

# Create Species Richness values for groups of unique combinations of
# ManagedAreaName, ProgramID, ProgramName, ProgramLocationID, SampleDate,
# GearType, and GearSize_m.
data <- data %>%
  group_by(AreaID, ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
    SampleDate, GearType, GearSize_m) %>%
  summarise(ParameterName=parameter,
    Year=unique(Year), Month=unique(Month),
    N_Species=sum(ResultValue),
    EffortCorrection_100m2=as.numeric(unique(EffortCorrection_100m2)),
    SpeciesRichness=N_Species/unique(EffortCorrection_100m2),
    .groups = "keep")

# Writes this data that is used by the rest of the script to a text file
fwrite(data, paste0(out_dir, "/Nekton_", param_file, "_UsedData.txt"), sep="|")

# Makes sure SampleDate is being stored as a Date object
data$SampleDate <- as.Date(data$SampleDate)

# Creates a variable with the names of all the managed areas that contain
# species observations
nekton_MA_Include <- unique(data$ManagedAreaName[!is.na(data$N_Species)])

# Puts the managed areas in alphabetical order
nekton_MA_Include <- nekton_MA_Include[order(nekton_MA_Include)]

# Determines the number of managed areas used
n <- length(nekton_MA_Include)

```

Managed Area Statistics

Gets summary statistics for each managed area. Uses piping from dplyr package to feed into subsequent steps. The following steps are performed:

1. Group data that have the same ManagedAreaName, Year, Month, GearType, and GearSize_m.
 - Second summary statistics do not use the Month grouping and are only for ManagedAreaName, Year, GearType, and GearSize_m.
 - Third summary statistics do not use Year grouping and are only for ManagedAreaName, Month, GearType, and GearSize_m
 - Fourth summary statistics are only grouped based on ManagedAreaName, GearType, and GearSize_m
 - Determines the years that the minimum and maximum species richness occurred
2. For each group, provide the following information: Parameter Name (ParameterName), Number of Entries (N_Data), Lowest Value (Min), Largest Value (Max), Median, Mean, Standard Deviation, and a list of all Program IDs included in these measurements.
3. Sort the data in ascending (A to Z and 0 to 9) order based on ManagedAreaName then Year then Month
4. Write summary stats to a pipe-delimited .txt file in the output directory

- Nekton Output Files in SEACAR GitHub (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/tree/main/Nekton/output)

```
# Create summary statistics for each managed area based on Year and Month
# intervals, and each gear type and size.
MA_YM_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, Year, Month, GearType, GearSize_m) %>%
  summarize(ParameterName=parameter,
             N_Data=length(na.omit(SpeciesRichness)),
             Min=min(SpeciesRichness),
             Max=max(SpeciesRichness),
             Median=median(SpeciesRichness),
             Mean=mean(SpeciesRichness),
             StandardDeviation=sd(SpeciesRichness),
             Programs=paste(sort(unique(ProgramName), decreasing=FALSE),
                             collapse=', '),
             ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),
                               collapse=', '),
             .groups = "keep")
# Puts the data in order based on ManagedAreaName, Year, Month, then GearSize
MA_YM_Stats <- as.data.table(MA_YM_Stats[order(MA_YM_Stats$ManagedAreaName,
                                              MA_YM_Stats$Year,
                                              MA_YM_Stats$Month,
                                              MA_YM_Stats$GearSize_m), ])

# Writes summary statistics to file
fwrite(MA_YM_Stats, paste0(out_dir, "/Nekton_", param_file,
                           "_MA_MMY_Stats.txt"), sep="|")
# Removes variable storing data to improve computer memory
rm(MA_YM_Stats)

# Create summary statistics for each managed area based on Year intervals,
# and each gear type and size.
MA_Y_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, Year, GearType, GearSize_m) %>%
  summarize(ParameterName=parameter,
             N_Data=length(na.omit(SpeciesRichness)),
             Min=min(SpeciesRichness),
             Max=max(SpeciesRichness),
             Median=median(SpeciesRichness),
             Mean=mean(SpeciesRichness),
             StandardDeviation=sd(SpeciesRichness),
             Programs=paste(sort(unique(ProgramName), decreasing=FALSE),
                             collapse=', '),
             ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),
                               collapse=', '),
             .groups = "keep")
# Puts the data in order based on ManagedAreaName, Year, then GearSize
MA_Y_Stats <- as.data.table(MA_Y_Stats[order(MA_Y_Stats$ManagedAreaName,
                                              MA_Y_Stats$Year,
                                              MA_Y_Stats$GearSize_m), ])

# Writes summary statistics to file
fwrite(MA_Y_Stats, paste0(out_dir, "/Nekton_", param_file,
                           "_MA_Yr_Stats.txt"), sep="|")
```

```

# Create summary statistics for each managed area based on Month intervals,
# and each gear type and size.
MA_M_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, Month, GearType, GearSize_m) %>%
  summarize(ParameterName=parameter,
    N_Data=length(na.omit(SpeciesRichness)),
    Min=min(SpeciesRichness),
    Max=max(SpeciesRichness),
    Median=median(SpeciesRichness),
    Mean=mean(SpeciesRichness),
    StandardDeviation=sd(SpeciesRichness),
    Programs=paste(sort(unique(ProgramName), decreasing=FALSE),
      collapse=', '),
    ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),
      collapse=', '),
    .groups = "keep")
# Puts the data in order based on ManagedAreaName, Month, then GearSize
MA_M_Stats <- as.data.table(MA_M_Stats[order(MA_M_Stats$ManagedAreaName,
  MA_M_Stats$Month,
  MA_M_Stats$GearSize_m), ])
# Writes summary statistics to file
fwrite(MA_M_Stats, paste0(out_dir, "/Nekton_", param_file,
  "_MA_Mo_Stats.txt"), sep="|")
# Removes variable storing data to improve computer memory
rm(MA_M_Stats)

# Create summary overall statistics for each managed area based each gear type
# and size.
MA_Ov_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, GearType, GearSize_m) %>%
  summarize(ParameterName=parameter,
    N_Years=length(unique(na.omit(Year))),
    EarliestYear=min(Year),
    LatestYear=max(Year),
    N_Data=length(na.omit(SpeciesRichness)),
    Min=min(SpeciesRichness),
    Max=max(SpeciesRichness),
    Median=median(SpeciesRichness),
    Mean=mean(SpeciesRichness),
    StandardDeviation=sd(SpeciesRichness),
    Programs=paste(sort(unique(ProgramName), decreasing=FALSE),
      collapse=', '),
    ProgramIDs=paste(sort(unique(ProgramID), decreasing=FALSE),
      collapse=', '),
    .groups = "keep")
# Puts the data in order based on ManagedAreaName then GearSize
MA_Ov_Stats <- as.data.table(MA_Ov_Stats[order(MA_Ov_Stats$ManagedAreaName,
  MA_Ov_Stats$GearSize_m), ])
# Creates Year_MinRichness and Year_MaxRichness columns
MA_Ov_Stats$Year_MinRichness <- NA
MA_Ov_Stats$Year_MaxRichness <- NA

# Loops through each ManagedAreaName, GearType, and GearSize_m.

```

```

# determines what year the minimum and maximum species richness occurred
for(m in 1:nrow(MA_Ov_Stats)){
  # Stores ManagedAreaName, GearType, and GearSize_m for this row
  ma <- MA_Ov_Stats$ManagedAreaName[m]
  gear <- MA_Ov_Stats$GearType[m]
  size <- MA_Ov_Stats$GearSize_m[m]
  # Skips to next row if there are no data for this combination
  if(MA_Ov_Stats$N_Data[m]==0){
    next
  }
  # Gets subset of data from MA_Y_Stats (yearly summary stats) with this
  # combination of ManagedAreaName, GearType, and GearSize_m
  ds <- MA_Y_Stats[MA_Y_Stats$ManagedAreaName==ma &
                    MA_Y_Stats$GearType==gear &
                    MA_Y_Stats$GearSize_m==size,]
  # Gets the minimum and maximum Mean (yearly averages)
  min <- min(ds$Mean)
  max <- max(ds$Mean)
  #Determines what years those minimum and maximum values occurred
  year_min <- ds$Year[ds$Mean==min]
  year_max <- ds$Year[ds$Mean==max]
  # Stores the occurrence years of the minimum and maximum into the overall
  # stats for this row
  MA_Ov_Stats$Year_MinRichness[m] <- year_min
  MA_Ov_Stats$Year_MaxRichness[m] <- year_max
}
# Replaces blank ProgramIDs with NA (missing values)
MA_Ov_Stats$ProgramIDs <- replace(MA_Ov_Stats$ProgramIDs,
                                  MA_Ov_Stats$ProgramIDs=="", NA)
MA_Ov_Stats$Programs <- replace(MA_Ov_Stats$Programs,
                                MA_Ov_Stats$Programs=="", NA)
# Write overall statistics to file
fwrite(MA_Ov_Stats, paste0(out_dir,"/Nekton_", param_file,
                            "_MA_Overall_Stats.txt"), sep="|")
# Removes entries from the overall statistics that do not have data.
# Based on presence or absence of EarliestYear
MA_Ov_Stats <- MA_Ov_Stats[!is.na(EarliestYear), ]

```

Appendix I: Managed Area Species Richness

The plots shown here are the species richness for each managed area with a yearly average, separated by gear size.

1. Set common plot theme.
2. Determine the earliest and latest year of the data to create x-axis scale and intervals
3. Determine the upper and lower limit of the plot for better y-axis labels
4. Determines what gear types are present and adjusts legend entries
5. Add the plot line
6. Set the plot type as a point plot with the size of the points
7. Create the title, x-axis, y-axis, and color fill labels
8. Set the y and x limits
9. Apply common plot theme
10. Add table with summary statistics below each figure

- Numerical non-integer values are rounded to 2 decimal places
- StandardDeviation is renamed StDev for space reasons

11. Create file name to save figure

12. Save figure as png file

- Nekton Figures in SEACAR GitHub (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/tree/main/Nekton/output/Figures)

```
# Defines standard plot theme: black and white, no major or minor grid lines,
# Arial font. Title is centered, size 12, and blue (hex coded). Subtitle is
# centered, size 10, and blue (hex coded). Legend title is size 10 and the
# legend is left-justified. X-axis title is size 10 and the margins are padded
# at the top and bottom to give more space for angled axis labels. Y-axis title
# is size 10 and margins are padded on the right side to give more space for
# axis labels. Axis labels are size 10 and the x-axis labels are rotated -45
# degrees with a horizontal justification that aligns them with the tick mark
plot_theme <- theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        text=element_text(family="Arial"),
        plot.title=element_text(hjust=0.5, size=12, color="#314963"),
        plot.subtitle=element_text(hjust=0.5, size=10, color="#314963"),
        legend.title=element_text(size=10),
        legend.text = element_text(hjust=0),
        axis.title.x = element_text(size=10, margin = margin(t = 5, r = 0,
                                                              b = 10, l = 0)),
        axis.title.y = element_text(size=10, margin = margin(t = 0, r = 10,
                                                              b = 0, l = 0)),
        axis.text=element_text(size=10),
        axis.text.x=element_text(angle = -45, hjust = 0))

# Color palette for SEACAR
color_palette <- c("#005396", "#0088B1", "#00ADAE", "#65CCB3", "#AEE4C1", "#FDEBA8", "#F8CD6D", "#F5A800")

# Determine geartype palette and shapes dynamically
# Combine type and size into one label for plots
MA_Y_Stats$GearType_Plot <- paste0(MA_Y_Stats$GearType, " (",
                                   MA_Y_Stats$GearSize_m, " m)")

# Determine unique gear types to create palettes
# gear_types <- unique(MA_Y_Stats$GearType_Plot)
gear_types <- c("Trawl (4.8 m)", "Trawl (6.1 m)", "Seine (183 m)")
# Trawl = triangle, seine = square
gear_shapes <- c(24, 24, 22)
# Trawl = #005396, Seine = #00ADAE
gear_colors <- c("#005396", "#005396", "#00ADAE")
names(gear_colors) <- gear_types
names(gear_shapes) <- gear_types

# Loop that cycles through each managed area with data
if(n==0){
  # Prints a statement if there are no managed areas with appropriate data
  print("There are no monitoring locations that qualify.")
} else {
  for (i in 1:n) {
```

```

ma_i <- nekton_MA_Include[i]
ma_abrev <- MA_All[ManagedAreaName==ma_i, Abbreviation]
# Gets data for target managed area
plot_data <- MA_Y_Stats[ManagedAreaName==ma_i, ]
# Find values <5% occurrence
remove_groups <- plot_data %>%
  group_by(SpeciesGroup2) %>%
  reframe(pct = (sum(N_Data) / sum(plot_data$N_Data))*100) %>%
  filter(pct<5) %>% pull(unique(SpeciesGroup2))
# Filter values <5% occurrence
plot_data <- plot_data %>% filter(!SpeciesGroup2 %in% remove_groups)

# Determines most recent year with available data for managed area
t_max <- max(MA_Ov_Stats[ManagedAreaName==ma_i, LatestYear])
# Determines earliest recent year with available data for managed area
t_min <- min(MA_Ov_Stats[ManagedAreaName==ma_i, EarliestYear])
# Determines how many years of data are present
t <- t_max-t_min

# Creates break intervals for plots based on number of years of data
if(t>=30){
  # Set breaks to every 10 years if more than 30 years of data
  brk <- -10
}else if(t<30 & t>=10){
  # Set breaks to every 5 years if between 30 and 10 years of data
  brk <- -5
}else if(t<10 & t>=4){
  # Set breaks to every 2 years if between 10 and 4 years of data
  brk <- -2
}else if(t<4 & t>=2){
  # Set breaks to every year if between 4 and 2 years of data
  brk <- -1
}else if(t<2){
  # Set breaks to every year if less than 2 years of data
  brk <- -1
  # Sets t_max to be 1 year greater and t_min to be 1 year lower
  # Forces graph to have at least 3 tick marks
  t_max <- t_max+1
  t_min <- t_min-1
}

# Determine range of data values for the managed area
y_range <- max(plot_data$Mean) - min(plot_data$Mean)

# Determines lower bound of y-axis based on data range. Set based on
# relation of data range to minimum value. Designed to set lower boundary
# to be 10% of the data range below the minimum value
y_min <- if(min(plot_data$Mean)-(0.1*y_range)<0){
  # If 10% of the data range below the minimum value is less than 0,
  # set as 0
  y_min <- 0
} else {
  # Otherwise set minimum bound as 10% data range below minimum value
  y_min <- min(plot_data$Mean)-(0.1*y_range)
}

```

```

}

# Sets upper bound of y-axis to be 10% of the data range above the
# maximum value.
y_max <- max(plot_data$Mean)+(0.1*y_range)

## Legend labels - grab list of unique SG2 for this MA
sp_list <- unique(plot_data$SpeciesGroup2)
sp_list <- sp_list[order(match(sp_list, names(sg2_palette)))]
# Create common name labels for legend display
sp_labels <- sapply(sp_list, function(x){sg_common[[x]]})

# Creates plot object using plot_data and grouping by the plot gear types.
# Data is plotted as symbols with connected lines.
p1 <- ggplot(data=plot_data,
             aes(fill = SpeciesGroup2, y=Mean, x=Year)) +
  geom_bar(position="stack", stat="identity") +
  facet_wrap(~GearType_Plot,
             nrow=2, ncol=1,
             strip.position = "right",
             scales = "free_y") +
  labs(title="Nekton Species Richness",
       subtitle=ma_i,
       x="Year", y=bquote('Richness (species/100'*~m^{2}*')')) +
  scale_fill_manual(name = "Species group",
                   values = subset(sg2_palette, names(sg2_palette) %in%
                                   unique(plot_data$SpeciesGroup2)),
                   labels = sp_labels) +
  scale_x_continuous(limits = c(t_min-1, t_max+1),
                    breaks = seq(t_max, t_min, brk)) +
  plot_theme
# Sets file name of plot created
outname <- paste0("Nekton_", param_file, "_", ma_abrev, ".png")
# Saves plot as a png image
png(paste0(out_dir, "/Figures/", outname),
    width = 8,
    height = 4,
    units = "in",
    res = 200)
print(p1)
dev.off()

# Creates a data table object to be shown underneath plots in report
ResultTable <- MA_Ov_Stats[ManagedAreaName==ma_i, ]
# Removes location, gear, and parameter information because it is in plot
# labels
ResultTable <- ResultTable[, -c("AreaID", "ManagedAreaName",
                                "ProgramIDs", "Programs", "GearType_Plot",
                                "ParameterName")]
# Renames StandardDeviation to StDev to save horizontal space
ResultTable <- ResultTable %>%
  rename("StDev"="StandardDeviation")
# Converts all non-integer values to 2 decimal places for space

```

```

ResultTable$Min <- round(ResultTable$Min, digits=2)
ResultTable$Max <- round(ResultTable$Max, digits=2)
ResultTable$Median <- round(ResultTable$Median, digits=2)
ResultTable$Mean <- round(ResultTable$Mean, digits=2)
ResultTable$StDev <- round(ResultTable$StDev, digits=2)
# Stores as plot table object
t1 <- ggtexttable(ResultTable, rows = NULL,
                  theme=ttheme(base_size=7))
# Combines plot and table into one figure
print(ggarrange(p1, t1, ncol=1, heights=c(0.85, 0.15)))

# Add extra space at the end to prevent the next figure from being too
# close. Does not add space after last plot
if(i!=n){
  cat("\n \n \n \n")
}
}
}

```