

SEACAR Nekton Analysis: Species Richness

Last compiled on 05 June, 2023

Contents

Important Notes	1
Libraries and Settings	1
File Import	2
Data Filtering	2
Managed Area Statistics	4
Appendix I: Managed Area Species Richness	7

Important Notes

These scripts were created by [J.E. Panzik \(jepanzik@usf.edu\)](mailto:jepanzik@usf.edu) for SEACAR.

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 [SEACAR_Nekton_SpeciesRichness_ReportRender.R](https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Nekton/SEACAR_Nekton_SpeciesRichness_ReportRender.R) (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Nekton/SEACAR_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).

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

File Import

Imports file that is determined in the SEACAR_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/35sn0n0lrrxi9dtkik030nozbnj9dyj>

The file being used for the analysis is: **All_NEKTON_Parameters-2023-Jun-01.txt**

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

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

```
## The data file used is:
## All_NEKTON_Parameters-2023-Jun-01.txt
```

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 **ManagedAreaName** values from the data are actually shortened versions, and are merged with the full versions. 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[data$ParameterName==param_name,]

if (param_name=="Presence"){
  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(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))

# Adds AreaID for each managed area by combining the MA_All datatable to the
# data based on ManagedAreaName
data <- merge.data.frame(MA_All[,c("AreaID", "ManagedAreaName")],
  data, by="ManagedAreaName", all=TRUE)

# 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
MA_Include <- unique(data$ManagedAreaName[!is.na(data$N_Species)])

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

# Determines the number of managed areas used
n <- length(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) (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=', '))

# 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=', '))
# 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=', '))
# 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=', '))
# 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$Programs=="", NA)
# Write overall statistics to file
fwrite(MA_Ov_Stats, paste0(out_dir, "/Nekton-", param_file,
                            "_MA_Overall_Stats.txt"), sep="|")
# Removes entries from the overall statistics that do not have data.
# Based on presence or absence of EarliestYear
MA_Ov_Stats <- MA_Ov_Stats[!is.na(MA_Ov_Stats$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) (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.align = 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")

# Defines and sets variable with standardized gear colors for plots
```

```

gear_colors <- c("Trawl (4.8 m)"=color_palette[1],
               "Trawl (6.1 m)"=color_palette[2],
               "Seine (183 m)"=color_palette[3])

# Defines and sets variable with standardized gear shapes for plots
gear_shapes <- c("Trawl (4.8 m)"=21,
               "Trawl (6.1 m)"=22,
               "Seine (183 m)"=24)

# 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) {
    # Gets data for target managed area
    plot_data <- MA_Y_Stats[MA_Y_Stats$ManagedAreaName==MA_Include[i]]
    # Gets the gear type(s) present for the managed area.
    # Combine type and size into one label for plots
    plot_data$GearType_Plot <- paste0(plot_data$GearType, " (",
                                     plot_data$GearSize_m, " m)")
    # Determines most recent year with available data for managed area
    t_max <- max(MA_Ov_Stats$LatestYear[MA_Ov_Stats$ManagedAreaName==
                                     MA_Include[i]])
    # Determines earliest recent year with available data for managed area
    t_min <- min(MA_Ov_Stats$EarliestYear[MA_Ov_Stats$ManagedAreaName==
                                     MA_Include[i]])
    # 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){
      # Set breaks to every year if less than 4 years of data
      brk <- -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)

# Determines what combination of gear are present for managed area
# and subsets color and shape scheme to be used by plots.
# Used so only gear combinations present for managed area appear in legend.
gear_colors_plot <- gear_colors[unique(plot_data$GearType_Plot)]
gear_shapes_plot <- gear_shapes[unique(plot_data$GearType_Plot)]

# 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, group=as.factor(GearType_Plot)) +
  geom_line(aes(x=Year, y=Mean, color=as.factor(GearType_Plot)),
            size=0.75, alpha=1) +
  geom_point(aes(x=Year, y=Mean, fill=as.factor(GearType_Plot),
               shape=as.factor(GearType_Plot)), size=2,
            color="#333333", alpha=1) +
  labs(title="Nekton Species Richness",
       subtitle=MA_Include[i],
       x="Year", y=bquote('Richness (species/100'*m^{2}*')'),
       fill="Gear type", color="Gear type", shape="Gear type") +
  scale_x_continuous(limits=c(t_min-0.25, t_max+0.25),
                    breaks=seq(t_max, t_min, brk)) +
  scale_y_continuous(limits=c(y_min, y_max),
                    breaks=pretty_breaks(n=5)) +
  scale_fill_manual(values=gear_colors_plot) +
  scale_color_manual(values=gear_colors_plot) +
  scale_shape_manual(values=gear_shapes_plot) +
  plot_theme

# Sets file name of plot created
outname <- paste0("Nekton_", gsub(" ", "", MA_Include[i]), "_",
                 param_file, ".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[MA_Ov_Stats$ManagedAreaName==MA_Include[i],]
# Removes location, gear, and parameter information because it is in plot
# labels

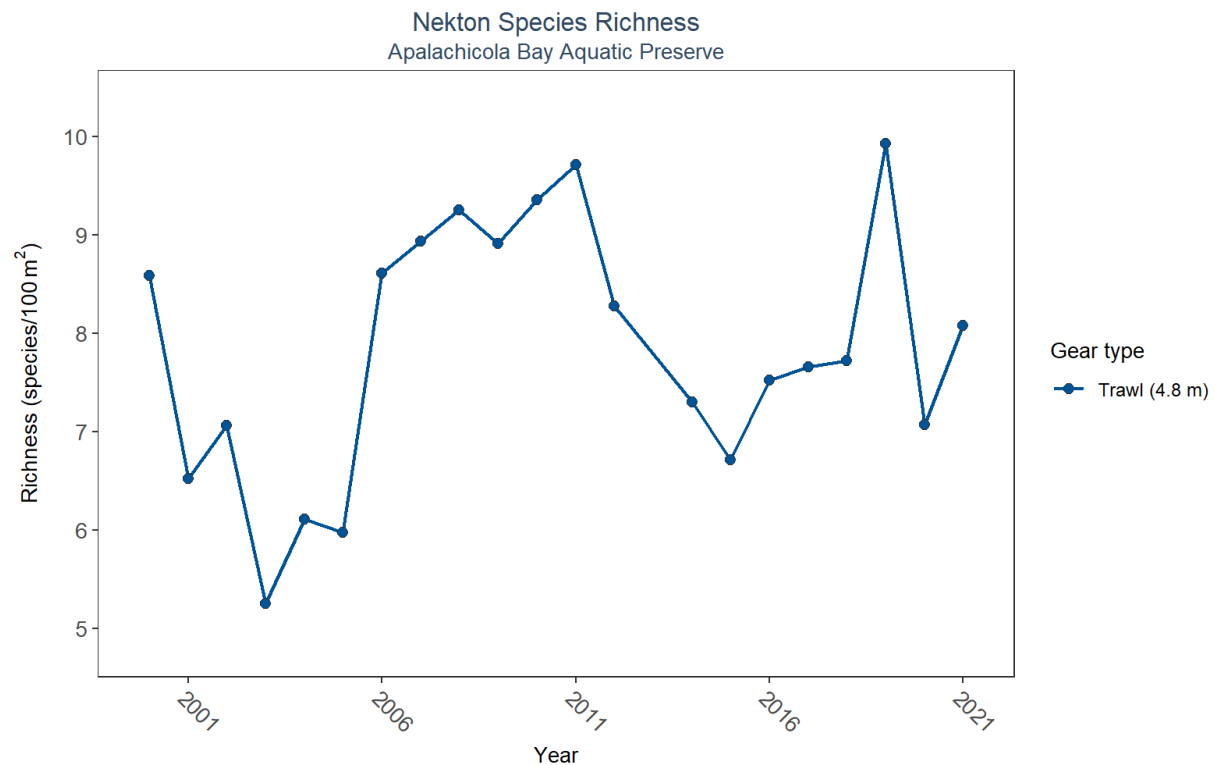
```

```

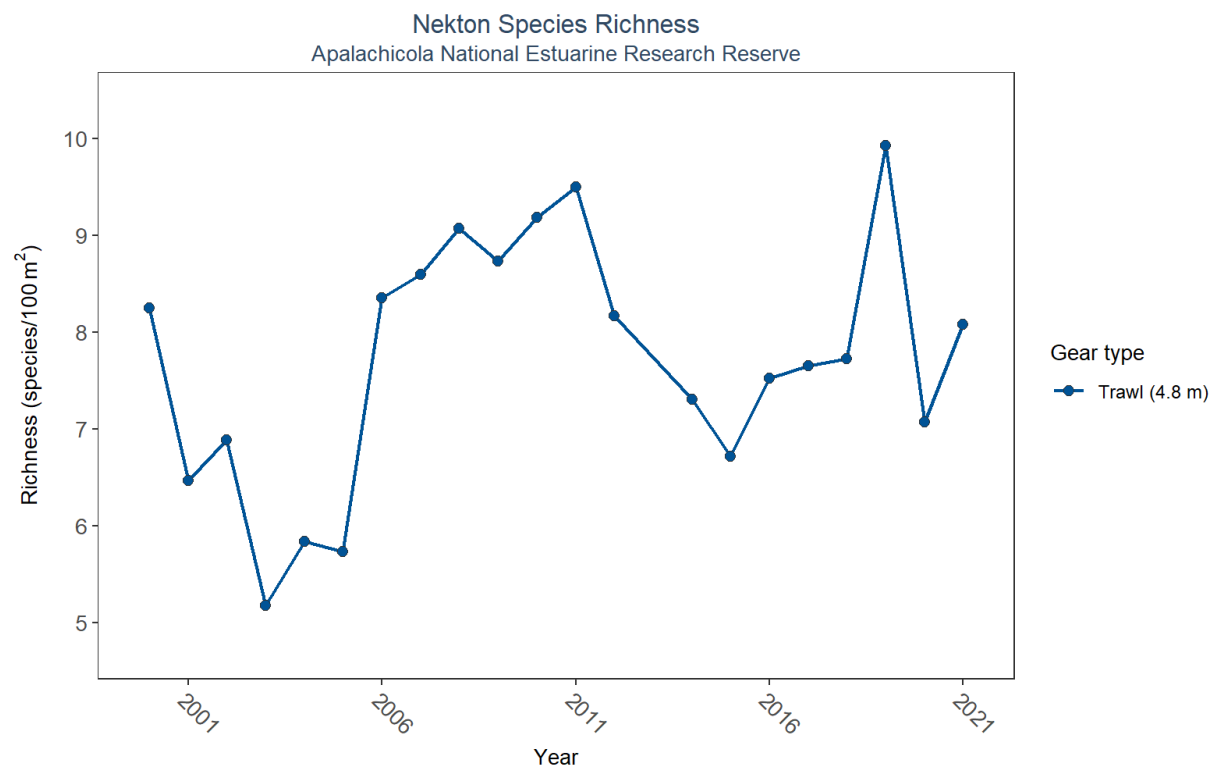
ResultTable <- ResultTable[,-c("AreaID", "ManagedAreaName",
                               "ProgramIDs", "GearType_Plot",
                               "ParameterName")]
# Renames StandardDeviation to StDev to save horizontal space
ResultTable <- ResultTable %>%
  rename("StDev"="StandardDeviation")
# 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
cat("\n \n \n \n \n")
}
}

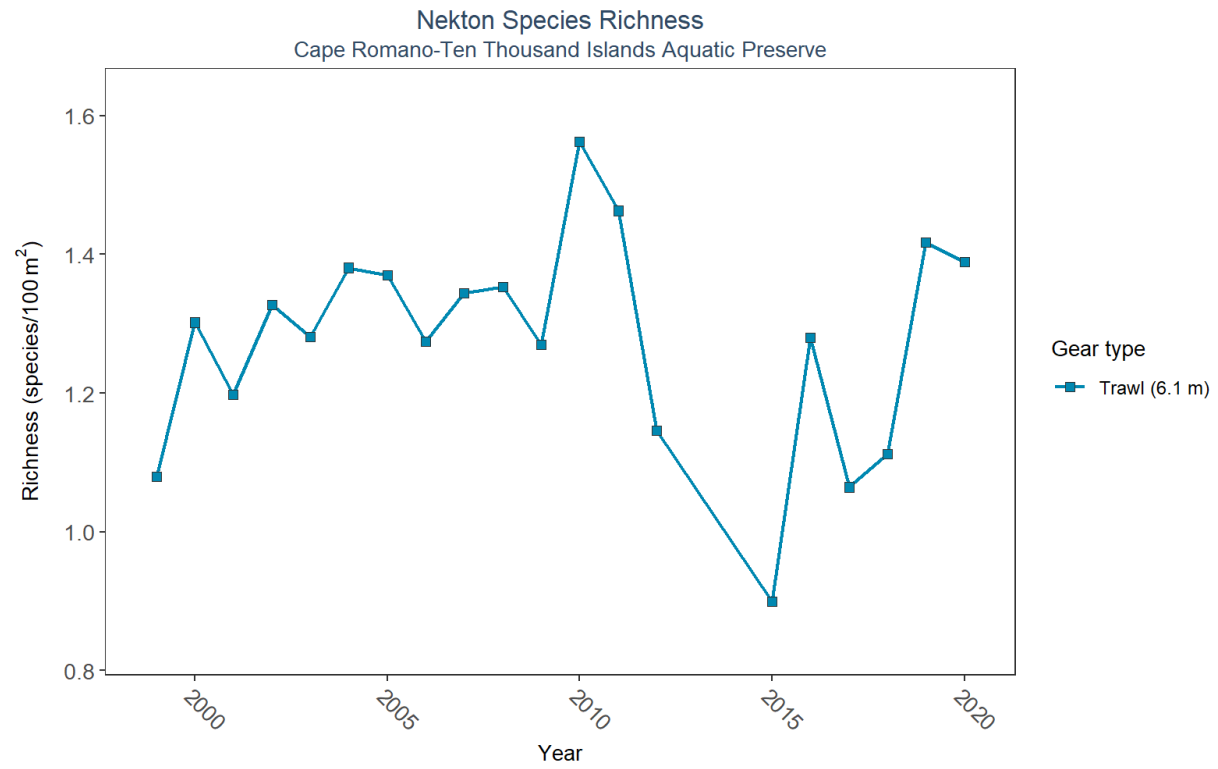
```



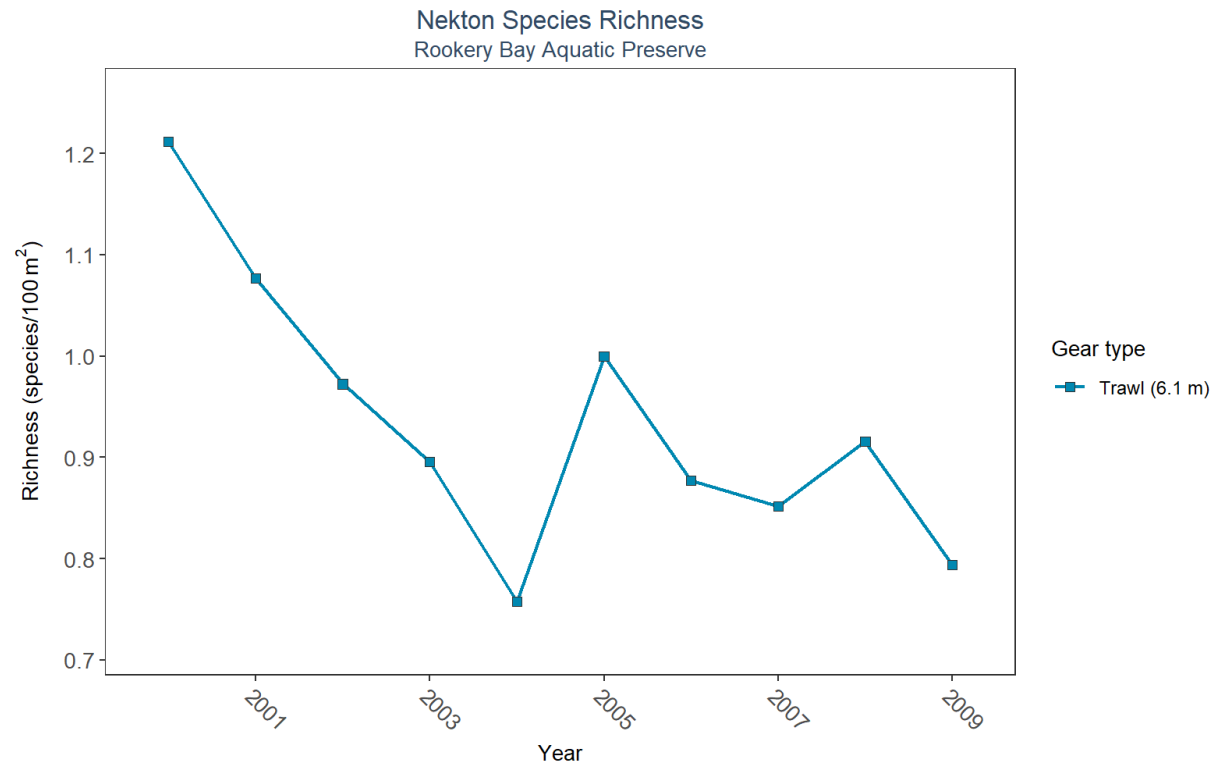
tYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Programs
10	2021	1664	0.19	24.81	7.78	7.79	3.62	Apalachicola National Estuarine Research Reserve Juvenile Fish and Benthic Macroinvertebrate Monitor



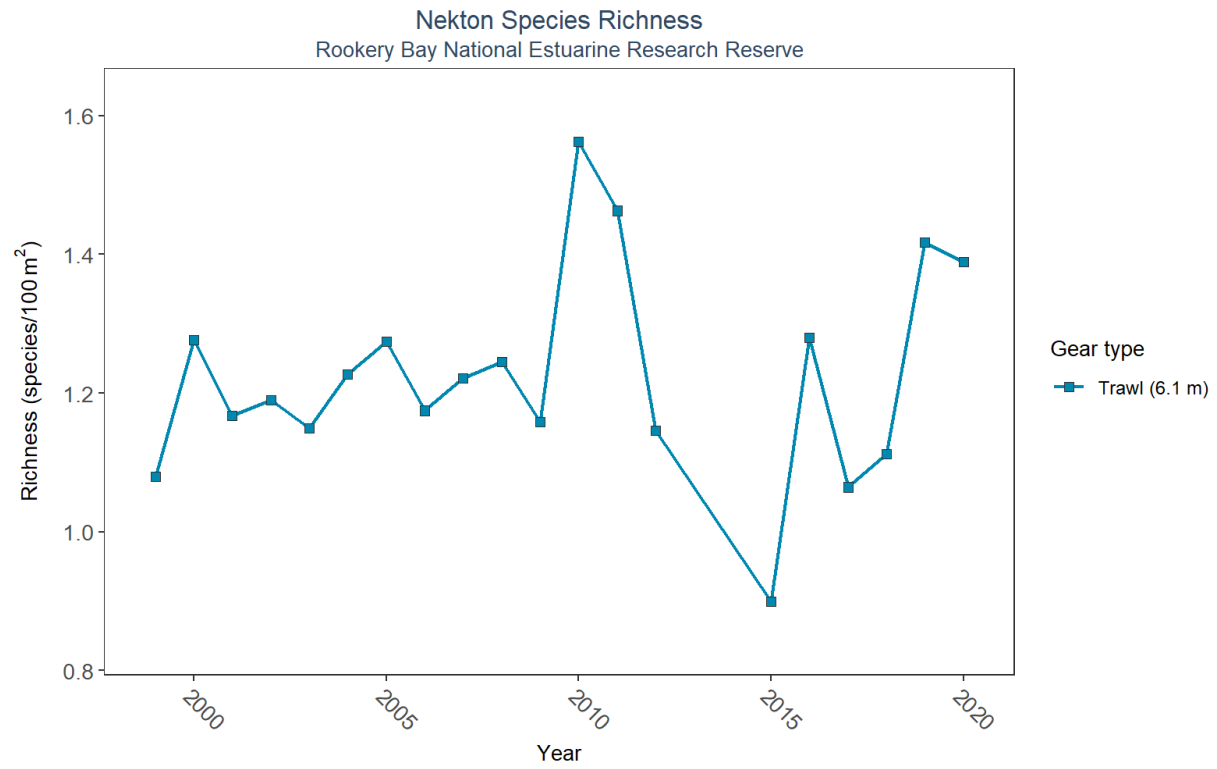
tYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Programs
10	2021	1931	0.19	24.81	7.41	7.62	3.72	Apalachicola National Estuarine Research Reserve Juvenile Fish and Benthic Macroinvertebrate Monitor



GearSize_m	N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Programs	Year_MinRichness	Year_MaxRichness
6.1	20	1999	2020	2555	0	3.37	1.35	1.31	0.53	RBNERR Fish Assessment	2015	2010



GearSize_m	N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Programs	Year_MinRichness	Year_MaxRichness
6.1	10	2000	2009	535	0	2.56	0.94	0.94	0.42	RBNERR Fish Assessment	2004	2000



GearSize_m	N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Programs	Year_MinRichness	Year_MaxRichness
6.1	20	1999	2020	3098	0	3.37	1.21	1.24	0.53	RBNERR Fish Assessment	2015	2010