

SEACAR Coral Analysis: Species Richness - Grazers and Reef-Dependent Species

Last compiled on 09 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

The purpose of this script is to determine species richness by species of grazers and reef-dependent species, create managed area statistics, generate plots, and create reports in pdf and Word document form for Coral data.

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_Coral_SpeciesRichness_ReportRender.R](https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Coral/SEACAR_Coral_SpeciesRichness_ReportRender.R) (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/blob/main/Coral/SEACAR_Coral_SpeciesRichness_ReportRender.R).

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)
knitr::opts_chunk$set(
  warning=FALSE,
  message=FALSE,
  dpi=200
)

```

File Import

Imports file that is determined in the SEACAR_Coral_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 Coral data is available at: <https://usf.box.com/s/8hyj2ur5arothlifg1isnq2gxjsjzbdg>

The file being used for the analysis is: **All_CORAL_Parameters-2023-Jun-06.txt**

```

data <- fread(file_in, sep="|", header=TRUE, stringsAsFactors=FALSE,
              na.strings=c("NULL", "", "NA"))

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

```

```

## The data file(s) used:
## All_CORAL_Parameters-2023-Jun-06.txt

```

Data Filtering

The processing and filtering that is done to the data is as follows:

1. Only take data rows that are Presence measurements for grazers and reef-dependent species
2. Set parameter names to **Species Richness**
3. Sets units
4. Removes rows that contains NA values in **ManagedAreaName**, **GenusName**, **SpeciesName**, **Month**, **Year**, **SpeciesGroup1**, and removes invasive species data
5. Sets **ResultValue** to be numeric values and removes rows where presence is 0
6. Removes duplicates (**MADup==1**)
7. Combines genus and species names

8. Corrects some managed area names to match what is being used with other habitats
9. Creates species richness data set
 - Grouped based on common ManagedAreaName, ProgramID, ProgramName, ProgramLocationID, and SampleDate
 - SpeciesRichness determined based on the number of unique species (`gensp`) in each group
10. Merges data with managed area data to determine correct AreaID
11. Writes to file with “_UsedData” file name to indicate what data was used for species richness.

```
# Only keep data for Presence of grazers and reef-dependent species
data <- data[data$ParameterName=="Presence - Grazers and Reef Dependent Species"]

# Create ParameterName Column
data$ParameterName <- "Species Richness"
parameter <- "Species Richness"
title_param <- "Species Richness - Grazers and Reef-Dependent Species"

# Sets units for species richness
unit <- "# of species"
data$ParameterUnits <- unit

# Remove rows with missing ManagedAreaName
data <- data[!is.na(data$ManagedAreaName),]
data <- data[data$ManagedAreaName!="NA",]
# Remove rows with missing GenusName
data <- data[!is.na(data$GenusName),]
# Remove rows with missing SpeciesName
data <- data[!is.na(data$SpeciesName),]
# Remove rows with missing Months
data <- data[!is.na(data$Month),]
# Remove rows with missing Years
data <- data[!is.na(data$Year),]
# Remove rows with missing SpeciesGroup1
data <- data[!is.na(data$SpeciesGroup1),]
# Remove rows with invasive species
data <- data[data$SpeciesGroup1!="Invasive",]
# Set ResultValue to be a number value
data$ResultValue <- as.numeric(data$ResultValue)
# Remove rows where ResultValue is 0 and missing
data <- data[data$ResultValue!=0,]
data <- data[!is.na(data$ResultValue),]
# Remove duplicate rows
data <- data[data$MADup==1,]
# Create variable that combines the genus and species name
data$gensp <- paste(data$GenusName, data$SpeciesName, sep=" ")
# Corrects Managed Area names to be consistent with official names
data$ManagedAreaName[data$ManagedAreaName=="Florida Keys NMS"] <-
  "Florida Keys National Marine Sanctuary"
data$ManagedAreaName[data$ManagedAreaName==
  "Biscayne Bay-Cape Florida to Monroe County Line"] <-
  "Biscayne Bay-Cape Florida to Monroe County Line Aquatic Preserve"
data$ManagedAreaName[data$ManagedAreaName=="Coupon Bight"] <-
  "Coupon Bight Aquatic Preserve"
data$ManagedAreaName[data$ManagedAreaName=="Coral ECA"] <-
```

```

    "Southeast Florida Coral Reef Ecosystem Conservation Area"

# Create Species Richness values for groups of unique combinations of
# ManagedAreaName, ProgramID, ProgramName, ProgramLocationID, and SampleDate.
data <- data[data$ResultValue==1] %>%
  group_by(ManagedAreaName, ProgramID, ProgramName, ProgramLocationID,
    SampleDate) %>%
  summarise(ParameterName=parameter,
    Year=unique(Year), Month=unique(Month),
    SpeciesRichness=length(unique(gensp)))

# 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, "/Coral_", 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$ManagedAreaName[!is.na(data$SpeciesRichness)])

# 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**, 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**
 - Fourth summary statistics are only grouped based on **ManagedAreaName**
 - 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 **Programs** 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
 - [Coral Species Richness Output Files in SEACAR GitHub \(https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/tree/main/Coral/output/SpeciesRichness\)](https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/tree/main/Coral/output/SpeciesRichness)

```

# Create summary statistics for each managed area based on Year and Month
# intervals.
MA_YM_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, Year, Month) %>%
  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 Month
MA_YM_Stats <- as.data.table(MA_YM_Stats[order(MA_YM_Stats$ManagedAreaName,
                                              MA_YM_Stats$Year,
                                              MA_YM_Stats$Month), ])

# Writes summary statistics to file
fwrite(MA_YM_Stats, paste0(out_dir, "/Coral_", 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
MA_Y_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, Year) %>%
  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 then Year
MA_Y_Stats <- as.data.table(MA_Y_Stats[order(MA_Y_Stats$ManagedAreaName,
                                              MA_Y_Stats$Year), ])

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

# Create summary statistics for each managed area based on Month intervals.
MA_M_Stats <- data %>%
  group_by(AreaID, ManagedAreaName, Month) %>%
  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 then Month
MA_M_Stats <- as.data.table(MA_M_Stats[order(MA_M_Stats$ManagedAreaName,
                                             MA_M_Stats$Month), ])

# Writes summary statistics to file
fwrite(MA_M_Stats, paste0(out_dir, "/Coral_", 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.
MA_Ov_Stats <- data %>%
  group_by(AreaID, ManagedAreaName) %>%
  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
MA_Ov_Stats <- as.data.table(MA_Ov_Stats[order(MA_Ov_Stats$ManagedAreaName), ])
# Creates Year_MinRichness and Year_MaxRichness columns
MA_Ov_Stats$Year_MinRichness <- NA
MA_Ov_Stats$Year_MaxRichness <- NA

# Loops through each ManagedAreaName.
# Determines what year the minimum and maximum species richness occurred
for(m in 1:nrow(MA_Ov_Stats)){
  # Stores ManagedAreaName for this row
  ma <- MA_Ov_Stats$ManagedAreaName[m]

  # Skips to next row if there are no data for this combination
  if(MA_Ov_Stats$N_Data[m]==0){
    next
  }
  # Gets subset of data from MA_Y_Stats (yearly summary stats) with this
  # ManagedAreaName
  ds <- MA_Y_Stats[MA_Y_Stats$ManagedAreaName==ma,]
  # 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, "/Coral_", 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.

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. Add the plot line
 5. Set the plot type as a point plot with the size of the points
 6. Create the title, x-axis, y-axis
 7. Set the y and x limits
 8. Apply common plot theme
 9. 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
 10. Create file name to save figure
 11. Save figure as png file
- Coral Species Richness Figures in SEACAR GitHub (https://github.com/FloridaSEACAR/SEACAR_Trend_Analyses/tree/main/Coral/output/SpeciesRichness/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", "#F17B00")

# 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$ManagedAreaName==MA_Include[i]]
    # Determines most recent year with available data for managed area
    t_max <- max(MA_Ov_Stats$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)

# Creates plot object using plot_data.
# Data is plotted as symbols with connected lines.
p1 <- ggplot(data=plot_data) +
  geom_line(aes(x=Year, y=Mean), color=color_palette[1],
            size=0.75, alpha=1) +
  geom_point(aes(x=Year, y=Mean), fill=color_palette[1],
            shape=21, size=2, color="#333333", alpha=1) +
  labs(title="Grazers and Reef-Dependent Species Richness",
        subtitle=MA_Include[i],
        x="Year", y="Richness (# of species)") +
  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)) +
  plot_theme
# Sets file name of plot created
outname <- paste0("Coral_", param_file, "_", gsub(" ", "", MA_Include[i]),
                  ".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, and parameter information because it is in plot
# labels
ResultTable <- ResultTable[,-c("AreaID", "ManagedAreaName",

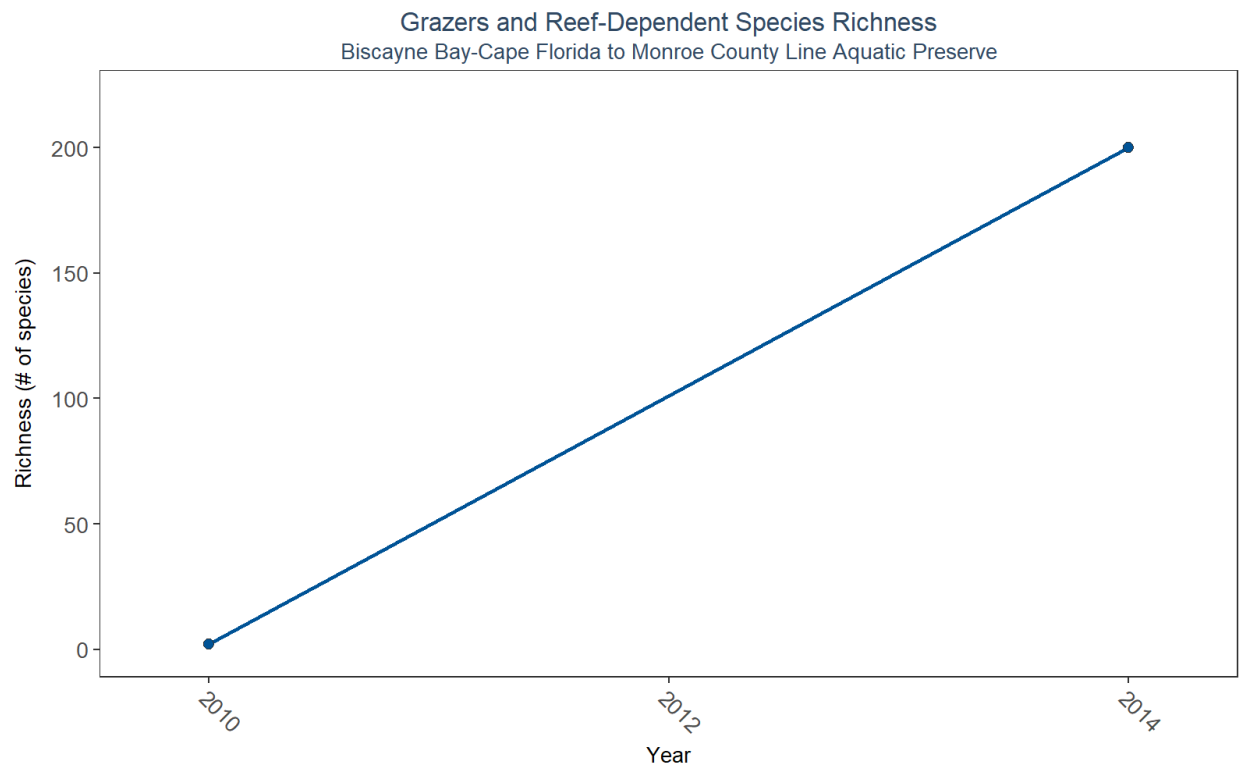
```

```

                                "ProgramIDs", "Programs", "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)))

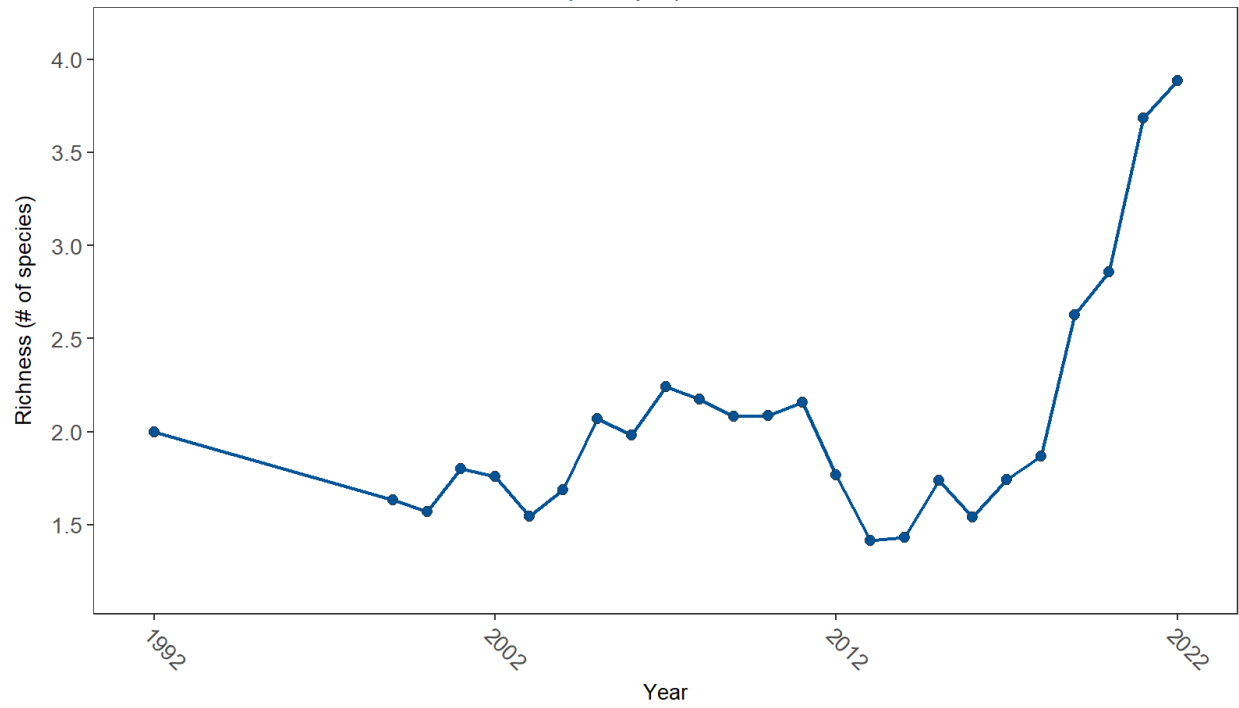
# Add extra space at the end to prevent the next figure from being too
# close. Does not add space after last plot
if(i!=n){
  cat("\n \n \n \n")
}
}
}

```

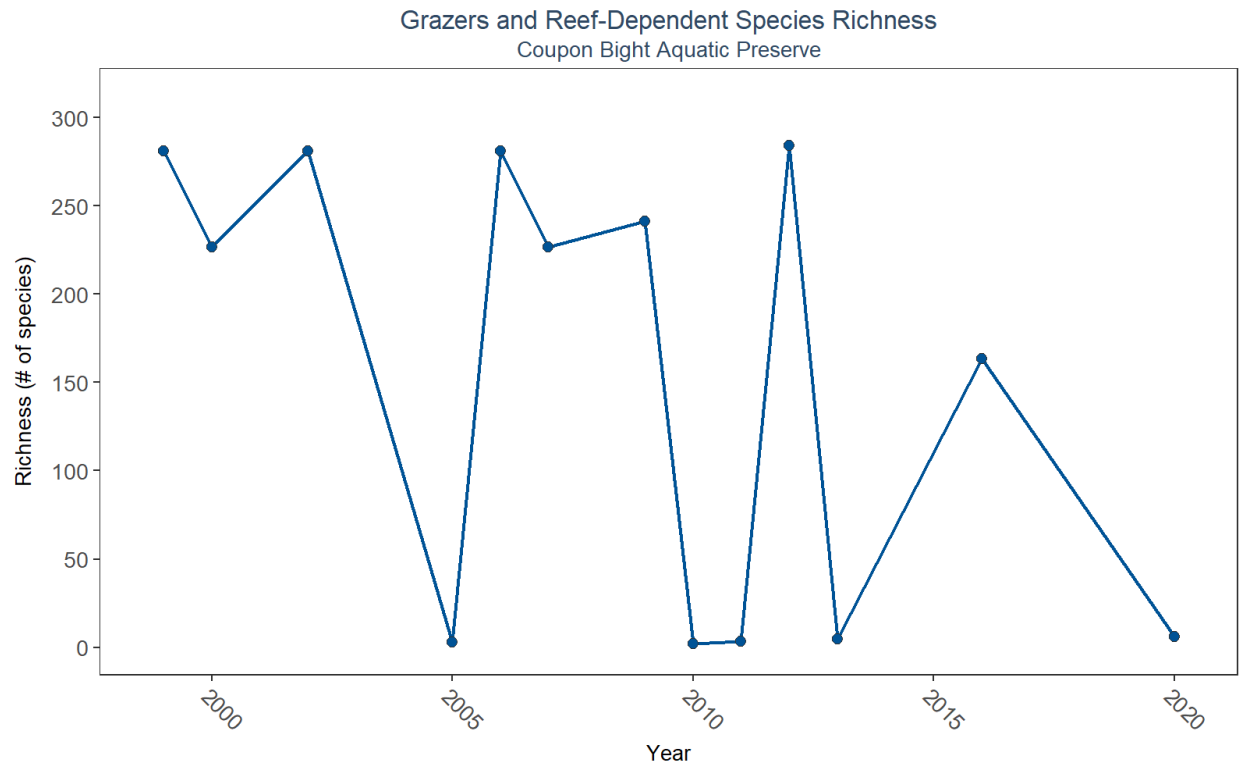


N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
2	2010	2014	3	2	200	200	134	114.32	2010	2014

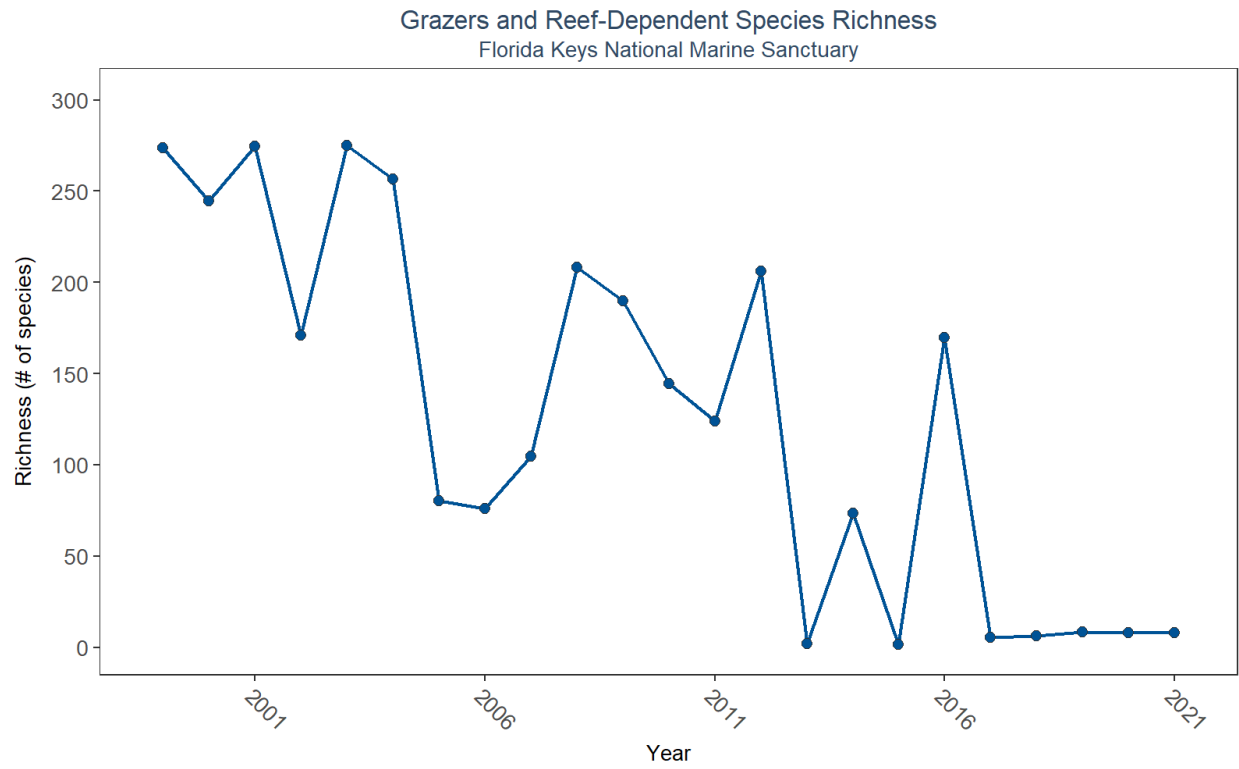
Grazers and Reef-Dependent Species Richness Biscayne Bay Aquatic Preserve



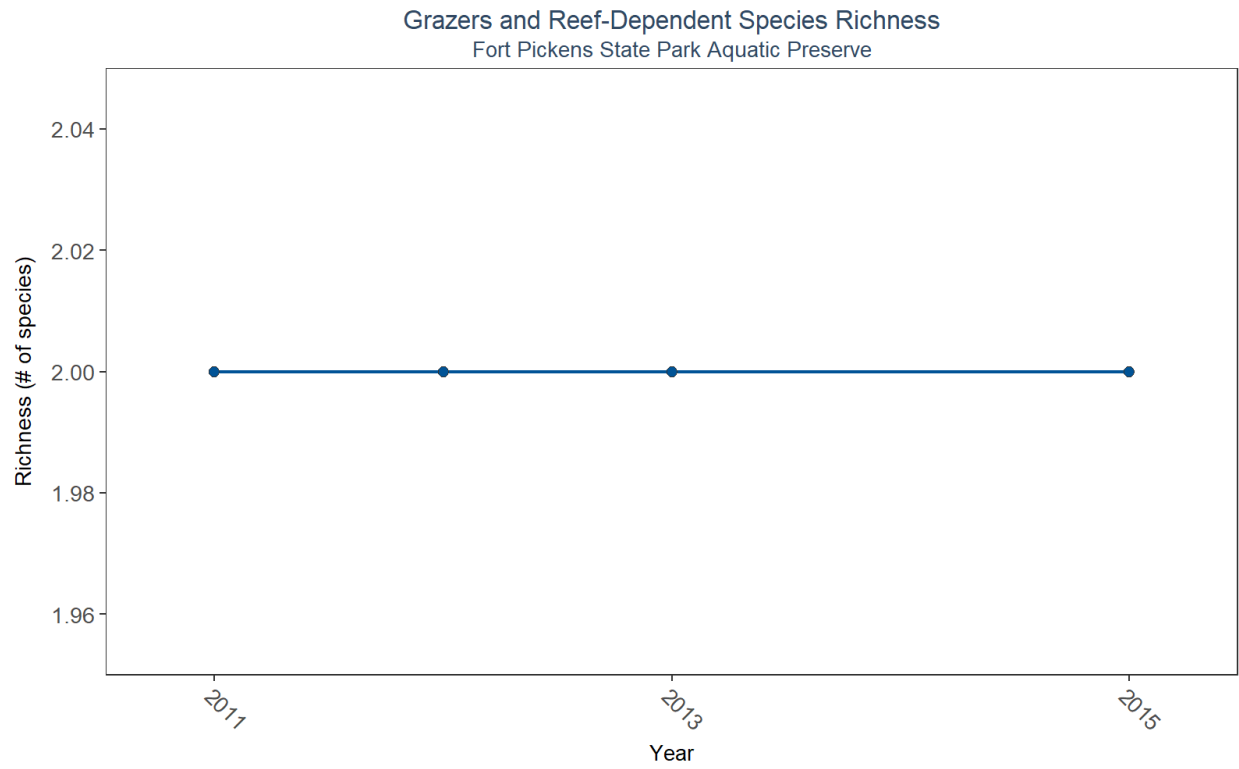
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
25	1992	2022	1963	1	8	2	2.18	1.11	2013	2022



N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
13	1999	2020	44	2	284	204	145.41	133.65	2010	2012

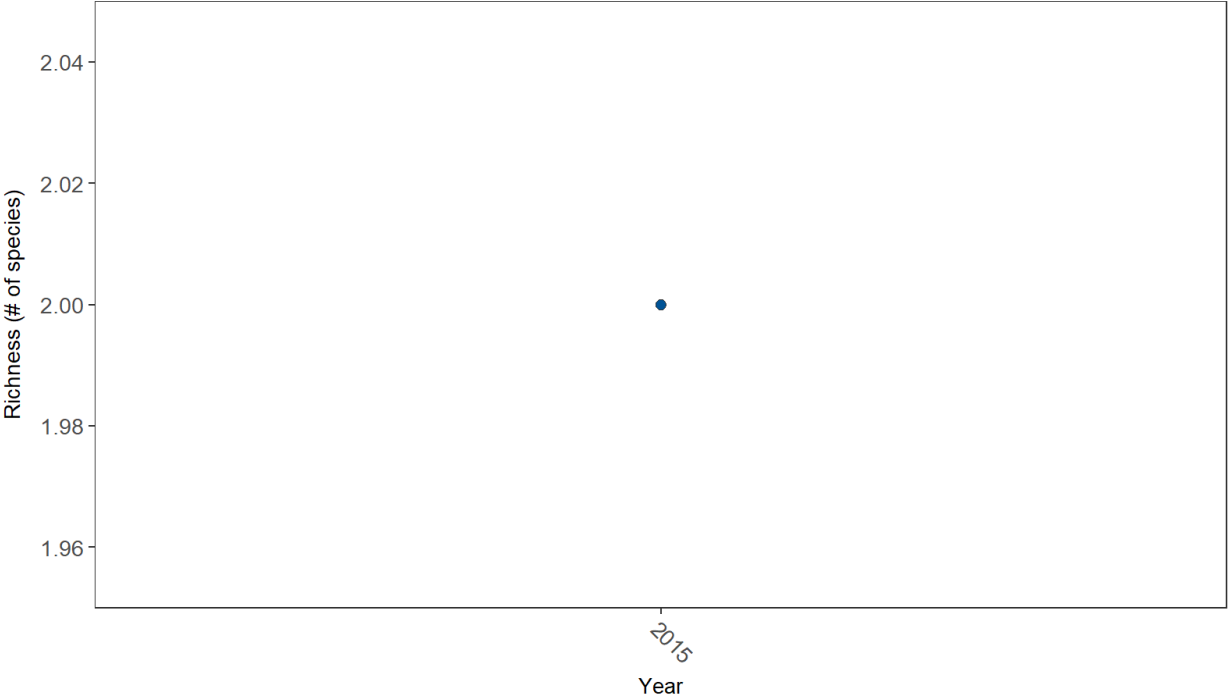


N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
23	1999	2021	20738	1	302	7	119.51	133.65	2015	2003



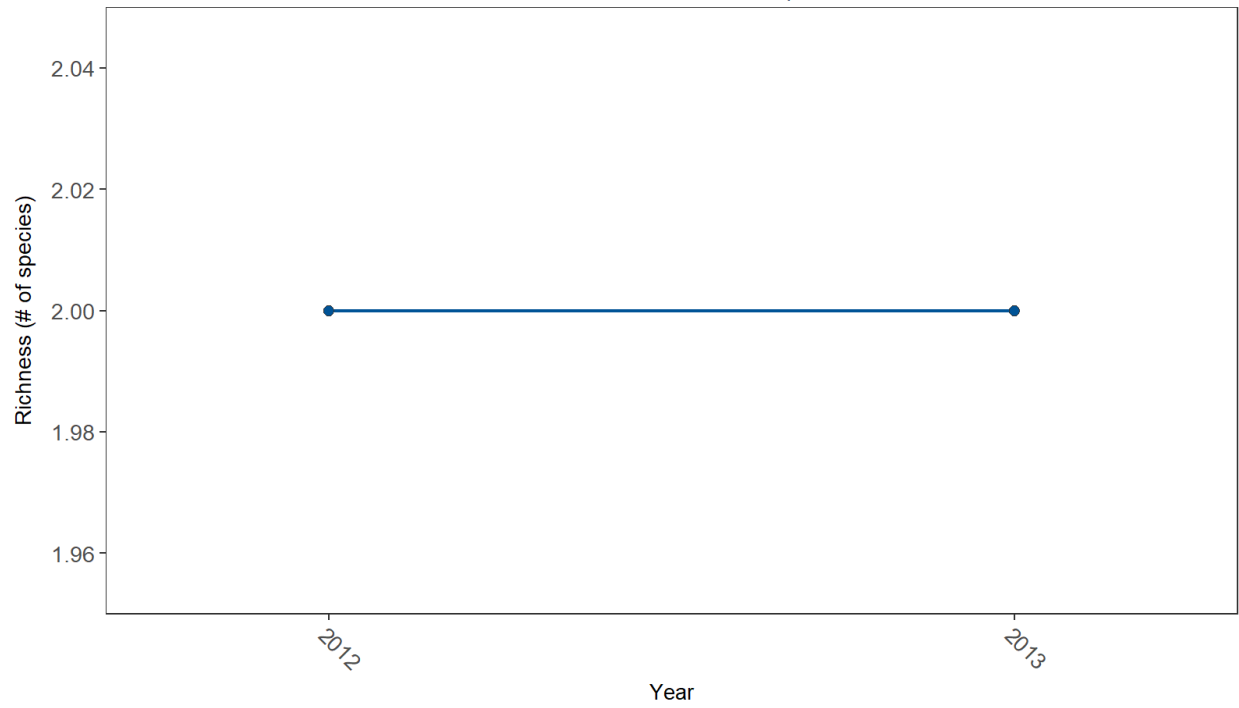
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
4	2011	2015	6	2	2	2	2	0	2011	2011

Grazers and Reef-Dependent Species Richness
Gasparilla Sound-Charlotte Harbor Aquatic Preserve



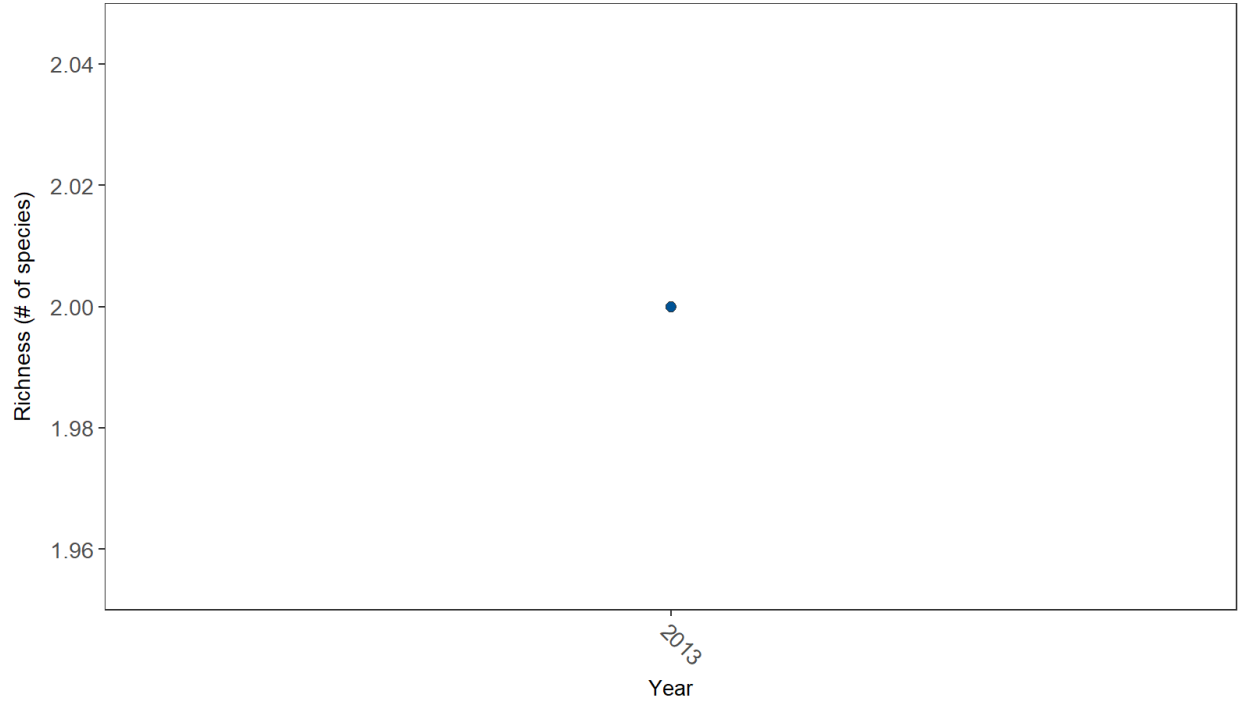
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
1	2015	2015	1	2	2	2	2	NA	2015	2015

Grazers and Reef-Dependent Species Richness Indian River-Malabar to Vero Beach Aquatic Preserve



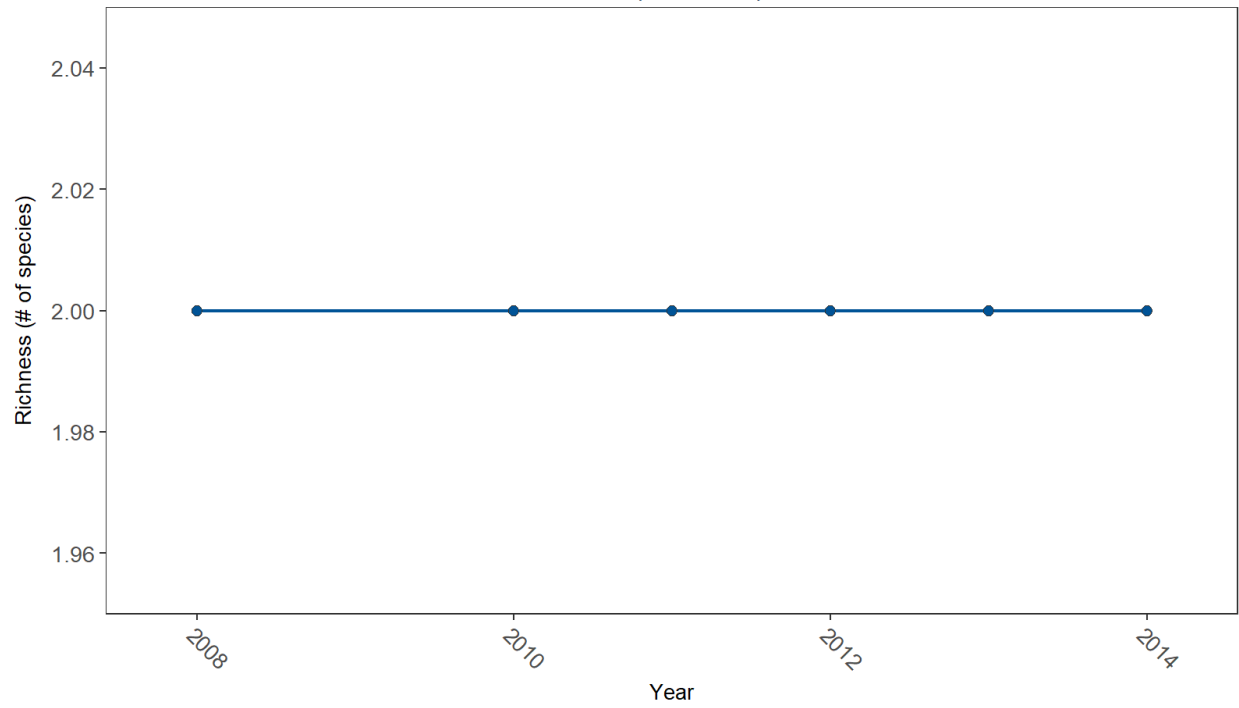
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
2	2012	2013	3	2	2	2	2	0	2012	2012

Grazers and Reef-Dependent Species Richness
Indian River-Vero Beach to Ft. Pierce Aquatic Preserve

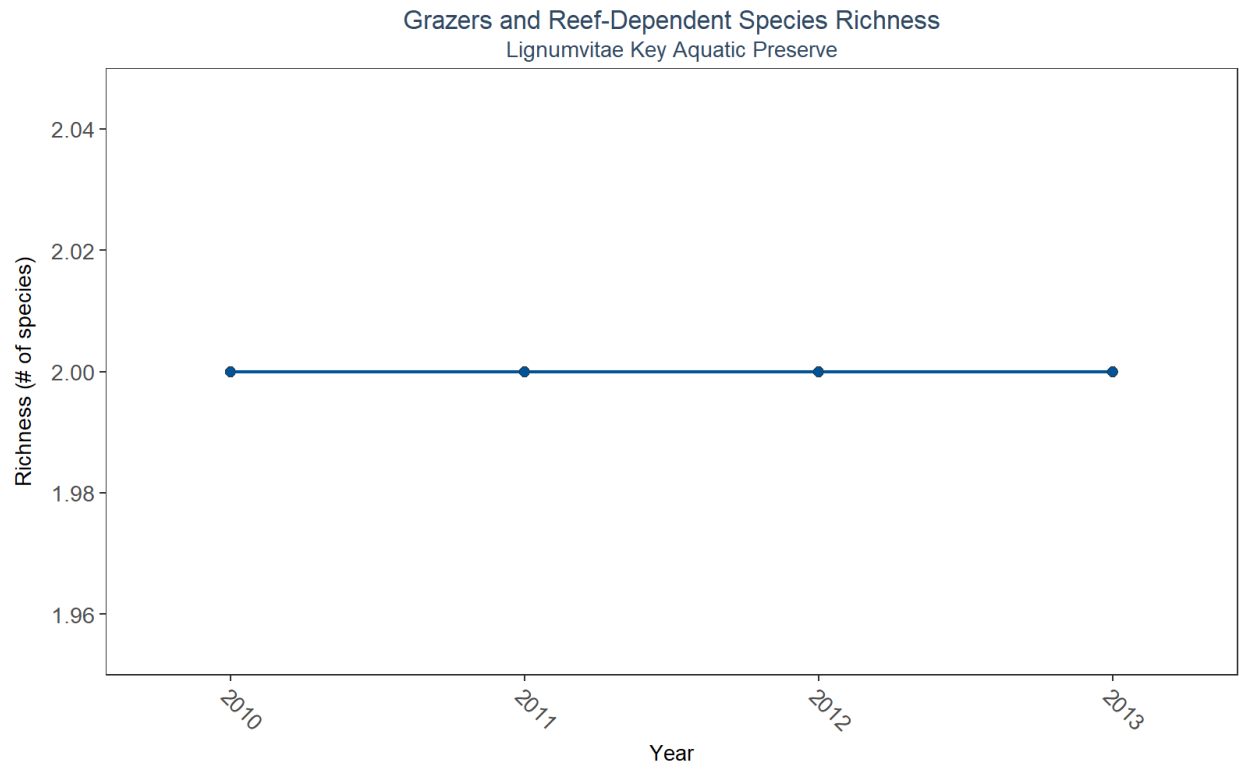


N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
1	2013	2013	5	2	2	2	2	0	2013	2013

Grazers and Reef-Dependent Species Richness Jensen Beach to Jupiter Inlet Aquatic Preserve

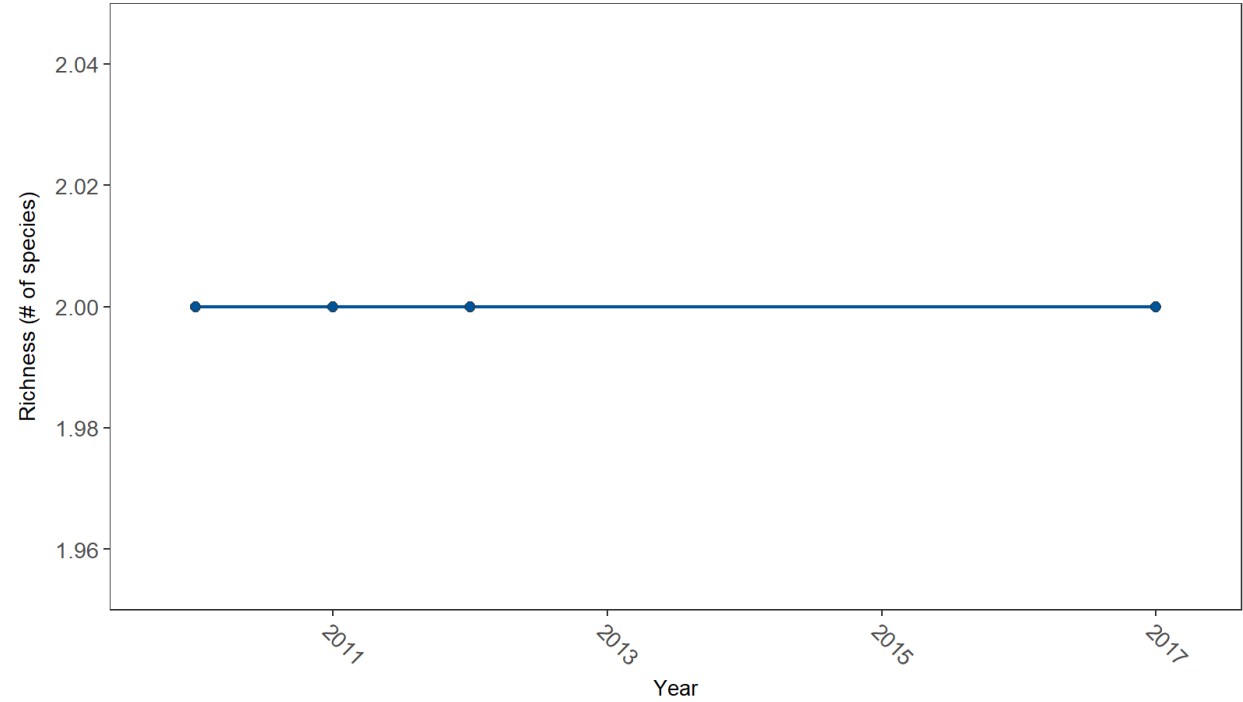


N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
6	2008	2014	24	2	2	2	2	0	2008	2008



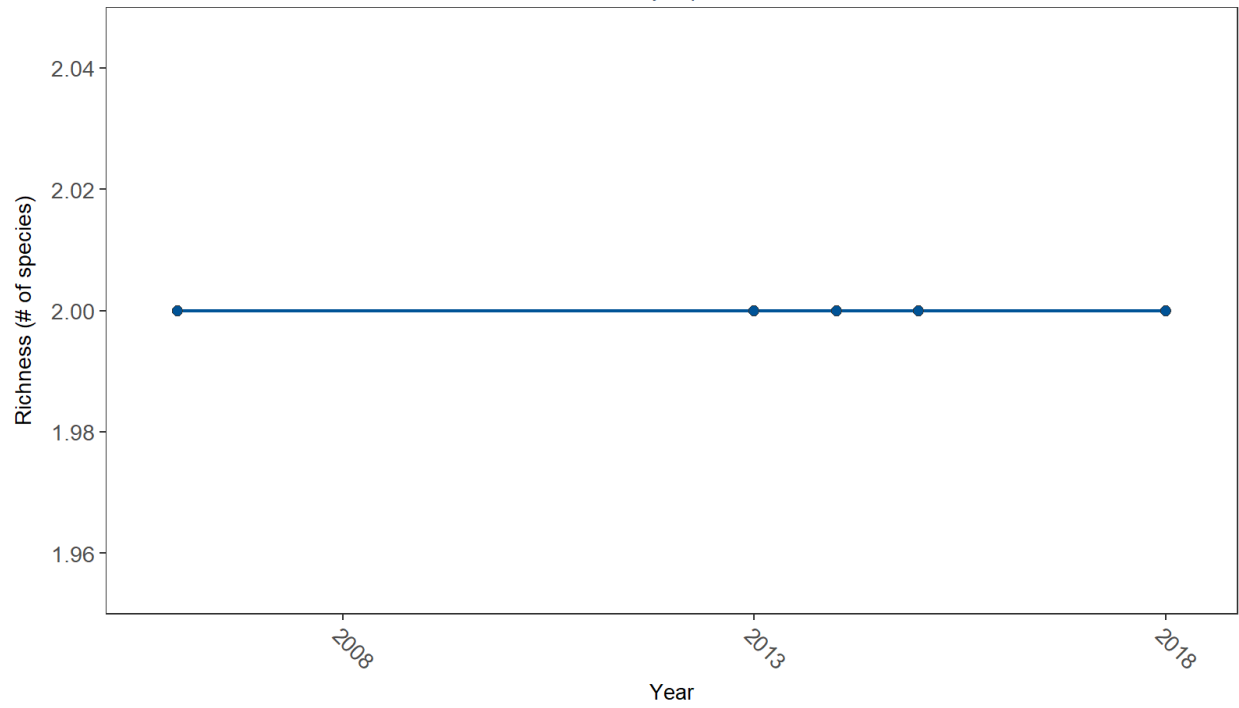
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
4	2010	2013	4	2	2	2	2	0	2010	2010

Grazers and Reef-Dependent Species Richness
Loxahatchee River-Lake Worth Creek Aquatic Preserve



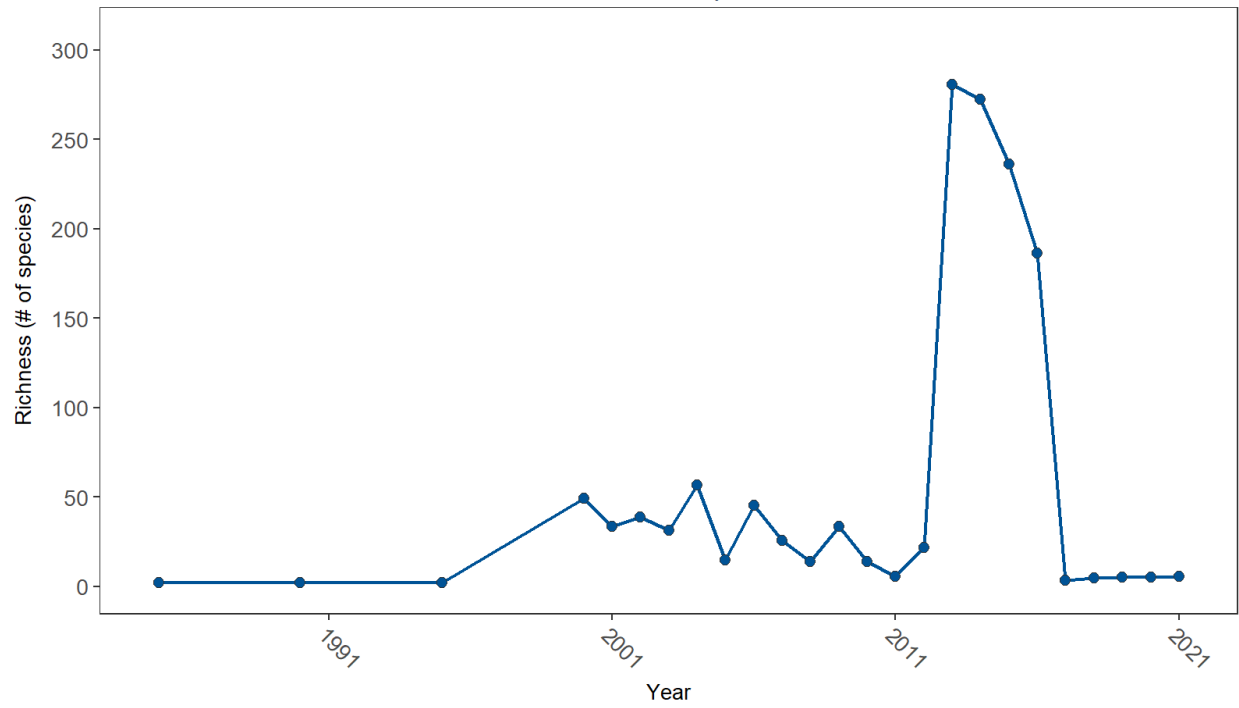
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
4	2010	2017	8	2	2	2	2	0	2010	2010

Grazers and Reef-Dependent Species Richness Pinellas County Aquatic Preserve



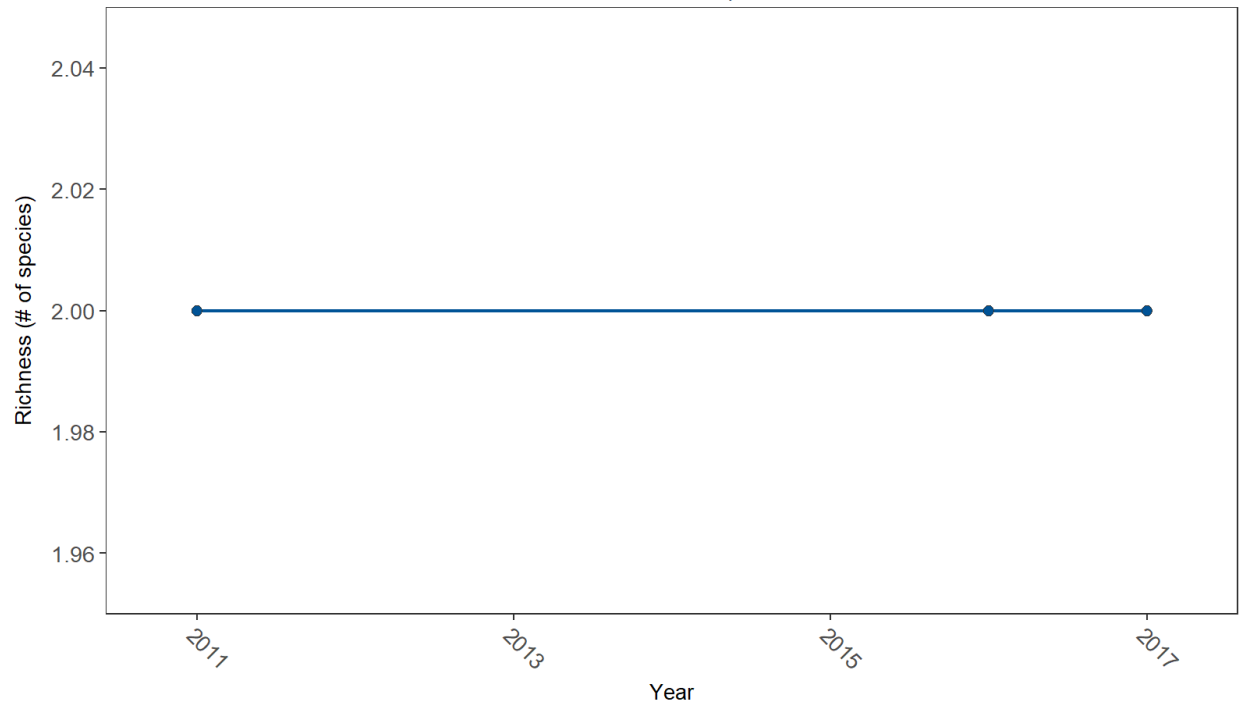
N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
5	2006	2018	9	2	2	2	2	0	2006	2006

Grazers and Reef-Dependent Species Richness
Southeast Florida Coral Reef Ecosystem Conservation Area

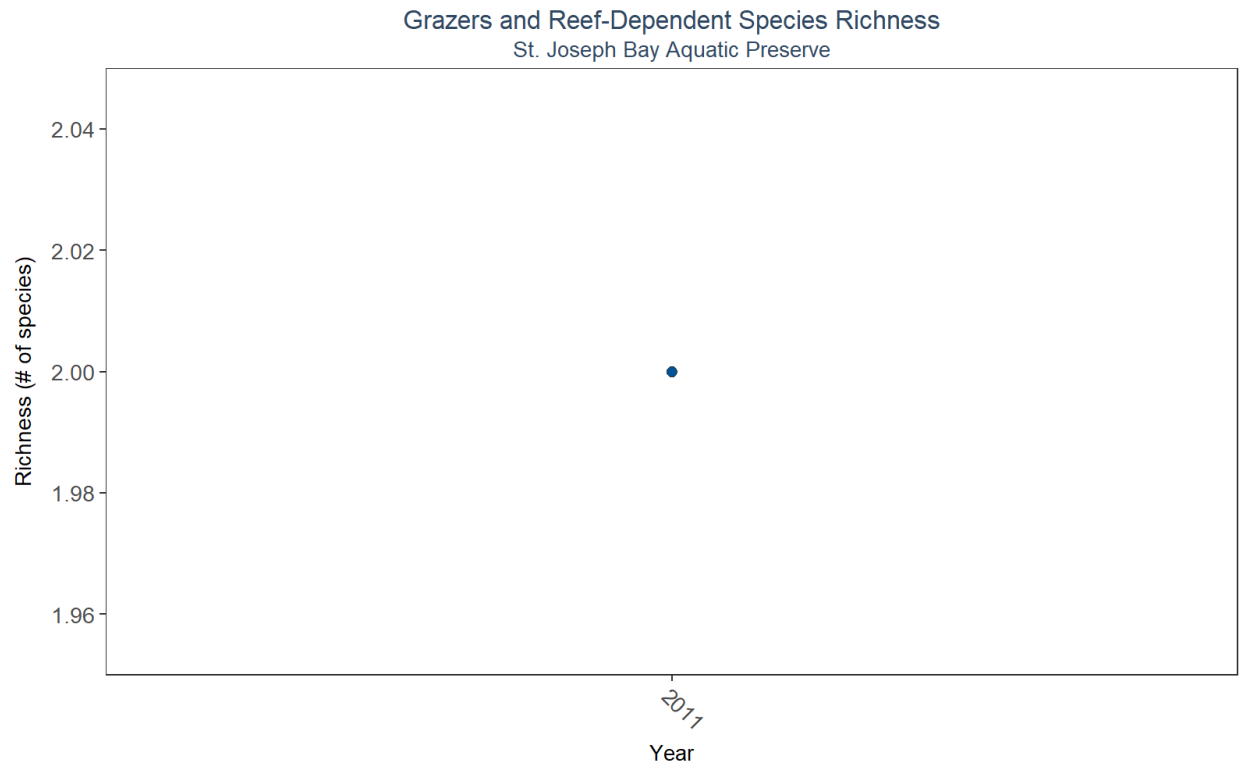


N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
25	1985	2021	4030	1	302	202	172.44	134.89	1985	2013

Grazers and Reef-Dependent Species Richness St. Andrews State Park Aquatic Preserve



N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
3	2011	2017	5	2	2	2	2	0	2011	2011



N_Years	EarliestYear	LatestYear	N_Data	Min	Max	Median	Mean	StDev	Year_MinRichness	Year_MaxRichness
1	2011	2011	3	2	2	2	2	0	2011	2011