ALY6000 Introduction to Analytics

Northeastern University

Module 3 Project – Executive Summary Report 3

Date: 05/02/2023

Submitted To: Richard He

Submitted By: Shree Vipulbhai Tejani

Analysis

This dataset ‘inchBio’ stores information regarding 8 types of fish species. This data has 7 variables and 676 records. It is a clean & organized data that is easy to interpret by all individuals. Therefore, it becomes easy for people to analyse and perform various statistical functions. The data stores numeric, character and Boolean datatype values only.

The **key findings** for the given set of instructions are as follows:

We have to import various libraries like FSA, FSAdata, dplyr, tidyr and more to perform the required tasks and support the analysis through data visualization.

```
Console   Terminal ×   Background Jobs ×
R 4.3.0 · ~/
> print("Shree Tejani")  # print author name
[1] "Shree Tejani"
> # use installed libraries
> library(FSA)
```

```
> library(FSAdata)
## FSAdata v0.4.0. See ?FSAdata to find data for specific fisheries analyses.
> library(magrittr)
> library(dplyr)
```

Importing the ‘inchBio.csv’ dataset using read.csv() function and have named this imported table as ‘tBio’. Creating a variable object for such functions helps in minimizing the size of code and also is easy to understand.

```
Console   Terminal ×   Background Jobs ×
R 4.3.0 · ~/
> #importing dataset from FSA library
> tBio <- read.csv("D:\\MPS_Quater 1\\ALY6000_Intro to Analytics\\inchBio.csv",
header=TRUE, stringsAsFactors=FALSE)
> tBio
    netID fishID  species  tl     w  tag scale
1      12     16 Bluegill  61   2.9      FALSE
2      12     23 Bluegill  66   4.5      FALSE
3      12     30 Bluegill  70   5.2      FALSE
4      12     44 Bluegill  38   0.5      FALSE
5      12     50 Bluegill  42   1.0      FALSE
6      12     65 Bluegill  54   2.1      FALSE
7      12     66 Bluegill  27    NA      FALSE
8      13     68 Bluegill  36   0.5      FALSE
9      13     69 Bluegill  59   2.0      FALSE
10     13     70 Bluegill  39   0.5      FALSE
```

Below I have displayed first 5 and last 5 records of ‘tBio’ dataset by using head() and tail() functions. The str() function is used to identify the datatypes of all the variables present in the data. It also specifies the number of records and variables. On the other hand, the summary() function is used to provide the statistical information like mean, median, mode, max, min, etc for all the columns.

```
Console  Terminal  Background Jobs
R 4.3.0 · ~/
> str(tBio)   #structure of tBio
'data.frame':   676 obs. of  7 variables:
 $ netID  : int  12 12 12 12 12 12 12 13 13 13 ...
 $ fishID : int  16 23 30 44 50 65 66 68 69 70 ...
 $ species: chr  "Bluegill" "Bluegill" "Bluegill" "Bluegill" ...
 $ tl     : int  61 66 70 38 42 54 27 36 59 39 ...
 $ w      : num  2.9 4.5 5.2 0.5 1 2.1 NA 0.5 2 0.5 ...
 $ tag    : chr  "" "" "" "" ...
 $ scale  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
> summary(tBio)   #summary of tBio
```

```
> summary(tBio)   #summary of tBio
     netID           fishID          species               tl              w               tag                scale
 Min.   :  1.00   Min.   :  7.0   Length:676         Min.   : 27.0   Min.   :   0.2   Length:676         Mode :logical
 1st Qu.: 13.00   1st Qu.:175.8   Class :character   1st Qu.: 66.0   1st Qu.:   2.0   Class :character   FALSE:213
 Median : 37.00   Median :345.5   Mode  :character   Median :189.5   Median :  54.5   Mode  :character   TRUE :463
 Mean   : 67.65   Mean   :434.2                      Mean   :186.5   Mean   : 126.8
 3rd Qu.:109.00   3rd Qu.:695.5                      3rd Qu.:295.0   3rd Qu.: 190.5
 Max.   :206.00   Max.   :915.0                      Max.   :429.0   Max.   :1070.0
                                                                     NA's   :165
```

```
Console  Terminal  Background Jobs
R 4.3.0 · ~/
> head(tBio,5) # head values for tBio
  netID fishID  species tl   w tag scale
1    12     16 Bluegill 61 2.9     FALSE
2    12     23 Bluegill 66 4.5     FALSE
3    12     30 Bluegill 70 5.2     FALSE
4    12     44 Bluegill 38 0.5     FALSE
5    12     50 Bluegill 42 1.0     FALSE
> tail(tBio,5) # tail values for tBio
    netID fishID       species  tl   w  tag scale
672   121    809 Black Crappie 282 352 1700  TRUE
673   121    812 Black Crappie 142  37       TRUE
674   110    863 Black Crappie 307 415 1783  TRUE
675   129    870 Black Crappie 279 344 1789  TRUE
676   129    879 Black Crappie 302 397 1792  TRUE
>
```

Creating an object ‘cts’ that counts total value of a particular species of the ‘tBio’ table.

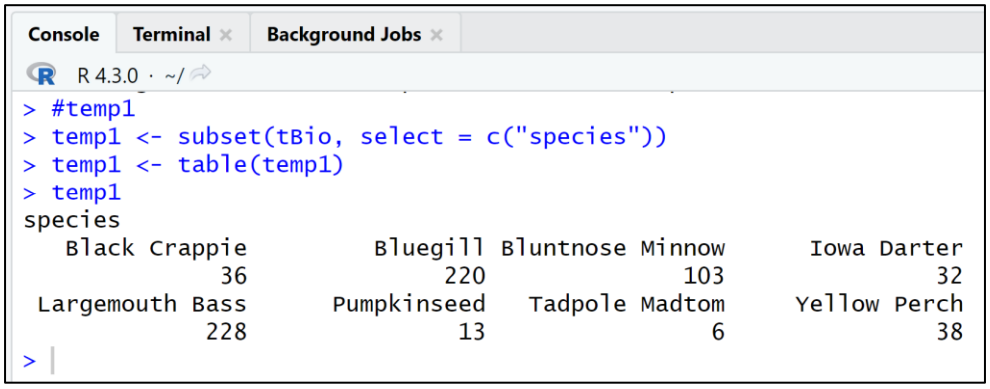
```
R 4.3.0 · ~/
> cts <- table(tBio$species) #obj 'CTS' to displays count
> cts

   Black Crappie         Bluegill Bluntnose Minnow      Iowa Darter
              36              220              103               32
 Largemouth Bass      Pumpkinseed   Tadpole Madtom     Yellow Perch
             228               13                6               38
> class(cts)
[1] "table"
>
```

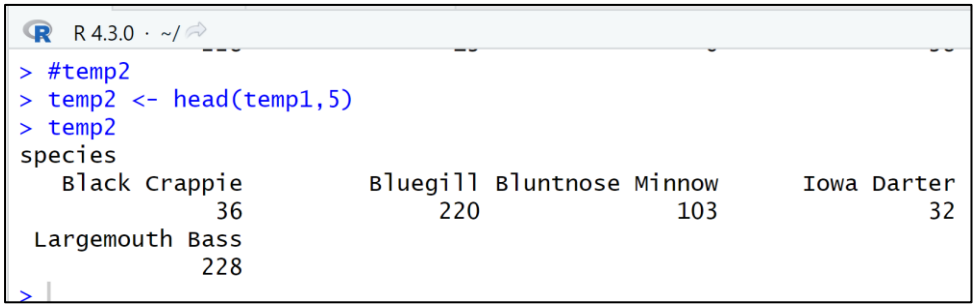
Names() function is used to display all the variable names of the ‘tBio’ table.

```
> names(cts)  # names of species
[1] "Black Crappie"    "Bluegill"         "Bluntnose Minnow" "Iowa Darter"
[5] "Largemouth Bass"  "Pumpkinseed"      "Tadpole Madtom"   "Yellow Perch"
>
```

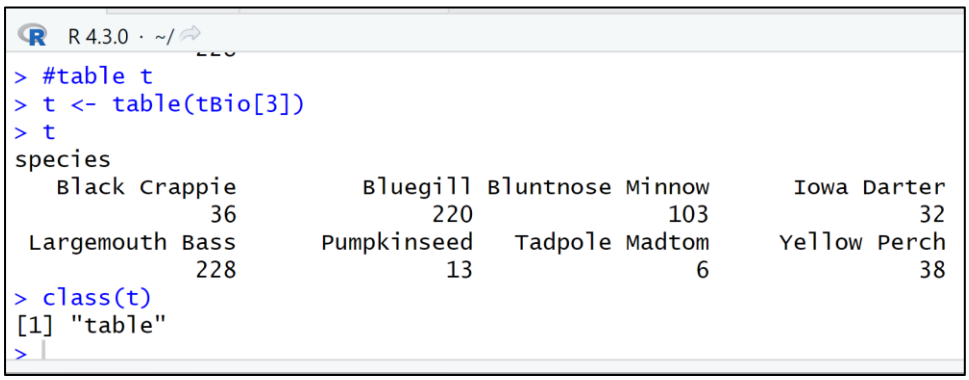
Created a ‘temp1’ object that lists all the species present and display number of records of each species in the dataset.



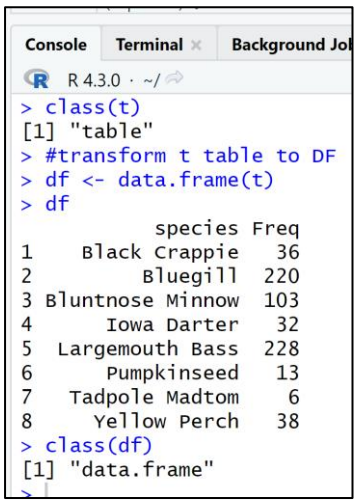
Creating a subset ‘temp2’ of the above temp1 object to display just first 5 records.



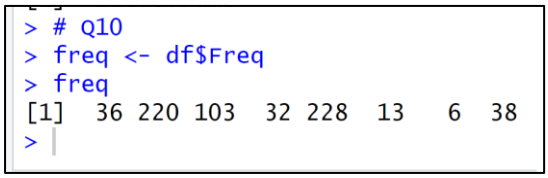
Create a ‘t’ table for just the species variable and display its class.



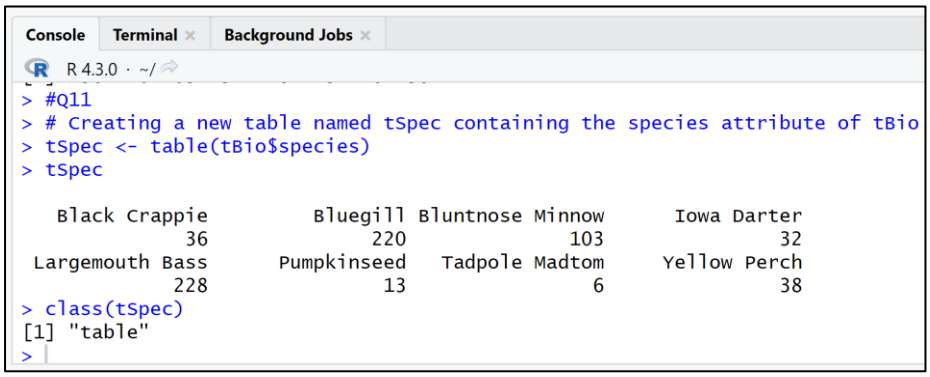
Converting the table ‘t’ to a dataframe ‘df’ by using data.frame() function.



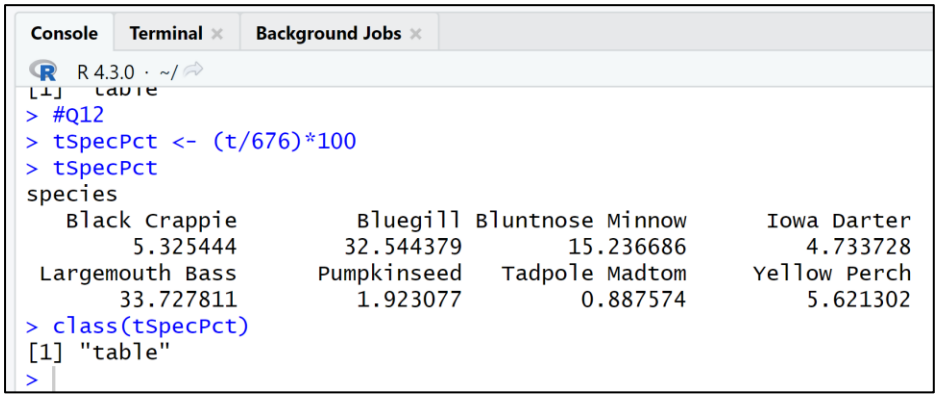
Displaying frequency values of ‘df’ dataframe as a variable ‘freq’.



Creating a new table ‘tSpec’ from the original dataset’s species variable.



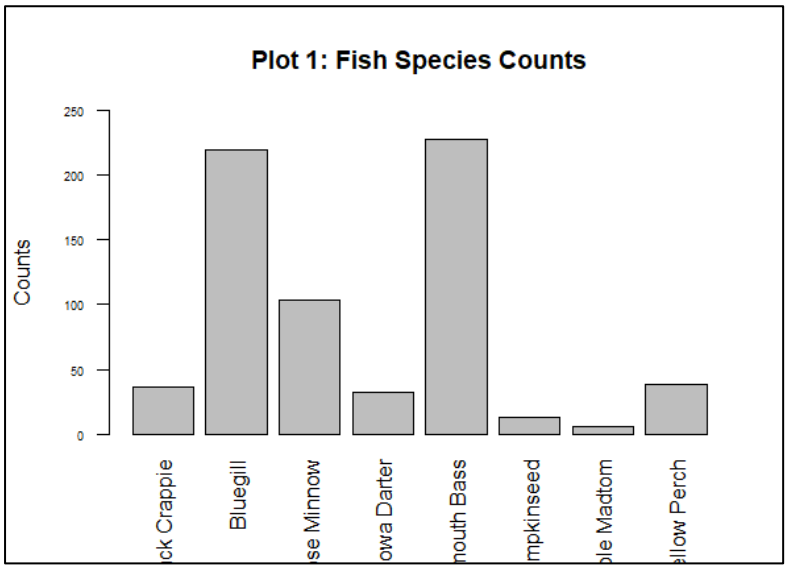
Creating a new table ‘tSpecPct’ that shows species and its percentage of records.



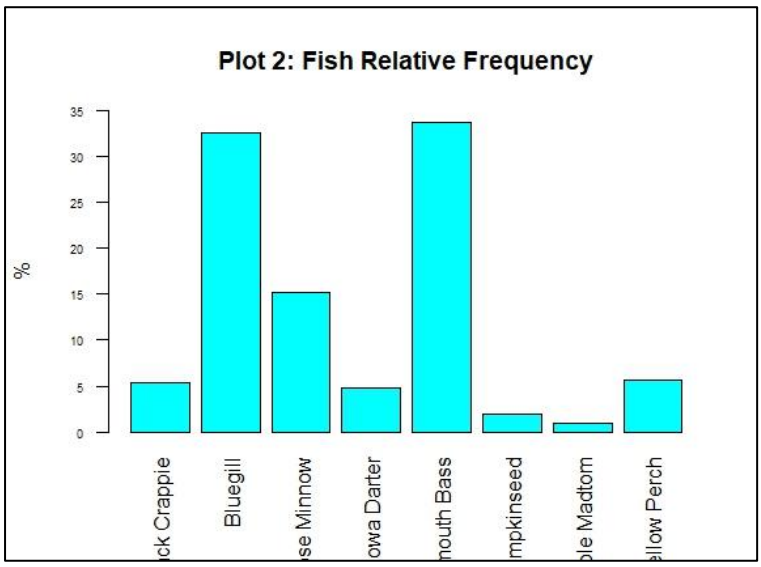
Converted table ‘tSpecPct’ to a ‘dfSP’ dataframe.

```
Console   Terminal   Background Jobs
R 4.3.0 · ~/
> dfSP <- as.data.frame(tSpecPct)
> dfSP
            species      Freq
1    Black Crappie  5.325444
2         Bluegill 32.544379
3 Bluntnose Minnow 15.236686
4      Iowa Darter  4.733728
5  Largemouth Bass 33.727811
6       Pumpkinseed  1.923077
7    Tadpole Madtom  0.887574
8     Yellow Perch  5.621302
> class(tSpecPct)
```

Creating a barplot for ‘tSpec’ table to identify the species and number of counts for that data.



Created a bar graph for ‘tSpecPct’ table for species and its percentage ratio.



Rearranging the ‘dfSP’ dataframe in descending order of relative frequency.

```
> #Q16
> data <- dfSP[order(-dfSP$Freq),]
> data
           species      Freq
5  Largemouth Bass 33.727811
2         Bluegill 32.544379
3 Bluntnose Minnow 15.236686
8     Yellow Perch  5.621302
1    Black Crappie  5.325444
4      Iowa Darter  4.733728
6      Pumpkinseed  1.923077
7   Tadpole Madtom  0.887574
>
```

Renamed the columns of data object to “Species” & “RelFreq”.

```
> #Q17
> colnames(data) <- c("Species", "RelFreq")
> data
           Species   RelFreq
5  Largemouth Bass 33.727811
2         Bluegill 32.544379
3 Bluntnose Minnow 15.236686
8     Yellow Perch  5.621302
1    Black Crappie  5.325444
4      Iowa Darter  4.733728
6      Pumpkinseed  1.923077
7   Tadpole Madtom  0.887574
>
```

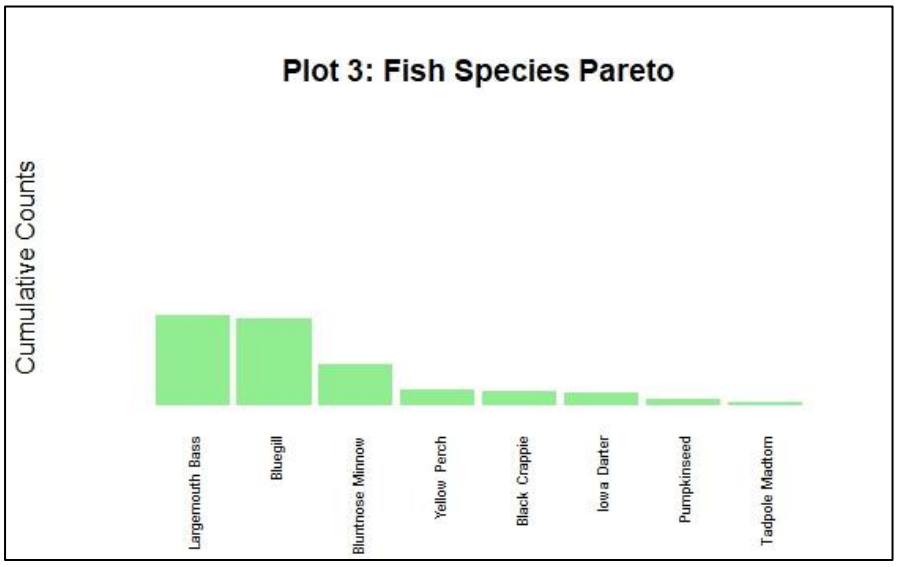
Created Additional variables in the ‘data’ object for cumFreq, cts and cumCts.

```
Console   Terminal   Background Jobs
R 4.3.0 · ~/
> #Q18
> # Calculate the cumulative relative frequency
> data$cumFreq <- cumsum(data$RelFreq)
> # Add a column to data with the count of each species
> data$cts <- data$RelFreq*676
> data$cumCts <- cumsum(cts)
> data
           Species   RelFreq   cumFreq   cts cumCts
5  Largemouth Bass 33.727811  33.72781 22800     36
2         Bluegill 32.544379  66.27219 22000    256
3 Bluntnose Minnow 15.236686  81.50888 10300    359
8     Yellow Perch  5.621302  87.13018  3800    391
1    Black Crappie  5.325444  92.45562  3600    619
4      Iowa Darter  4.733728  97.18935  3200    632
6      Pumpkinseed  1.923077  99.11243  1300    638
7   Tadpole Madtom  0.887574 100.00000   600    676
>
```

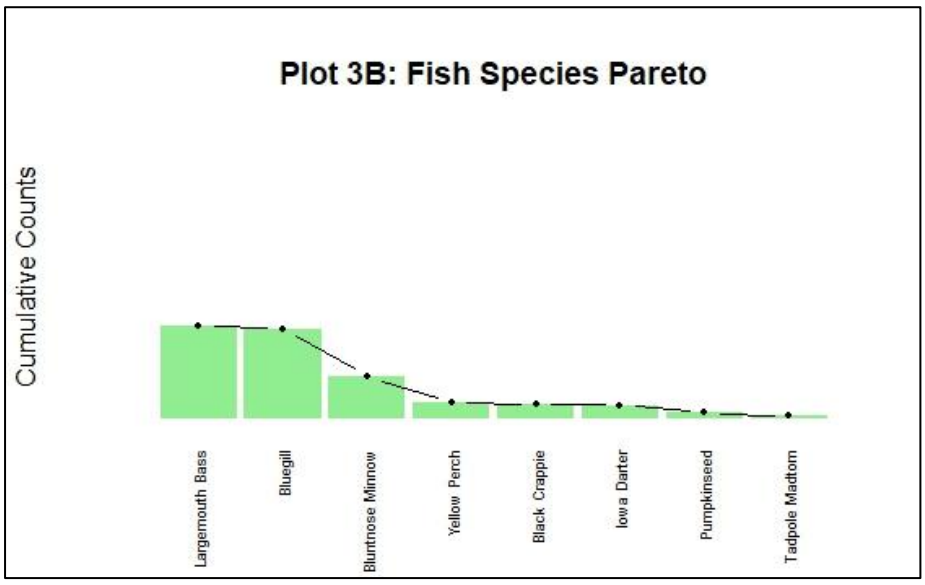
Creating a parameter variable ‘varPar’ for storing graphical parameters to plot graphs.

```
> #Q19
> varPar <- colnames(data)
> varPar
[1] "Species" "RelFreq" "cumFreq" "cts"     "cumCts"
>
```

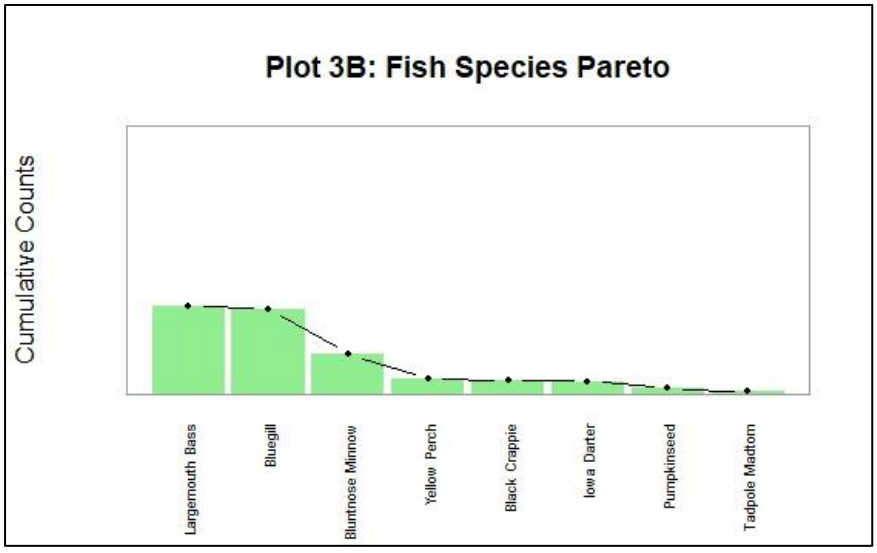
Plotted a bar graph for ‘data$cts’ by creating a ‘pc’ object for it.



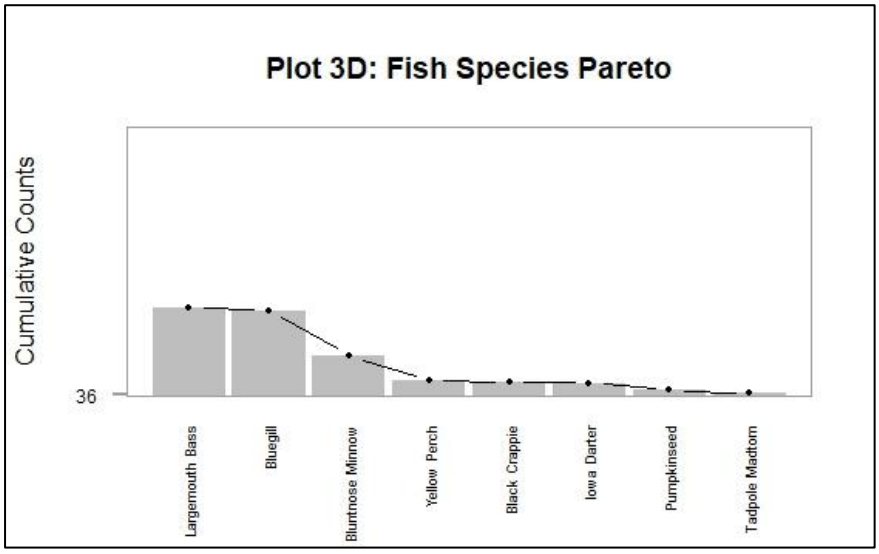
Additionally created a cumulative count line to the ‘pc’ plot.



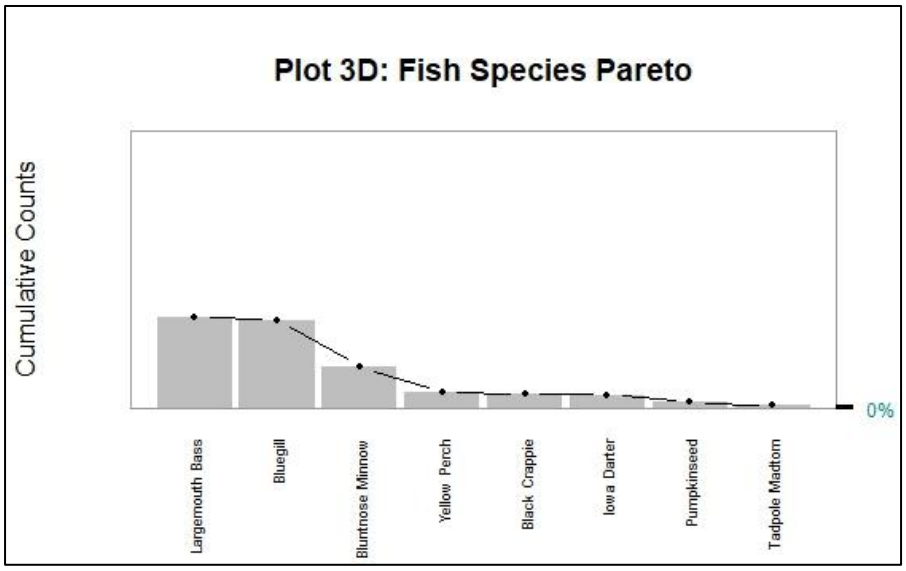
Have added a grey color box to the pareto plot.



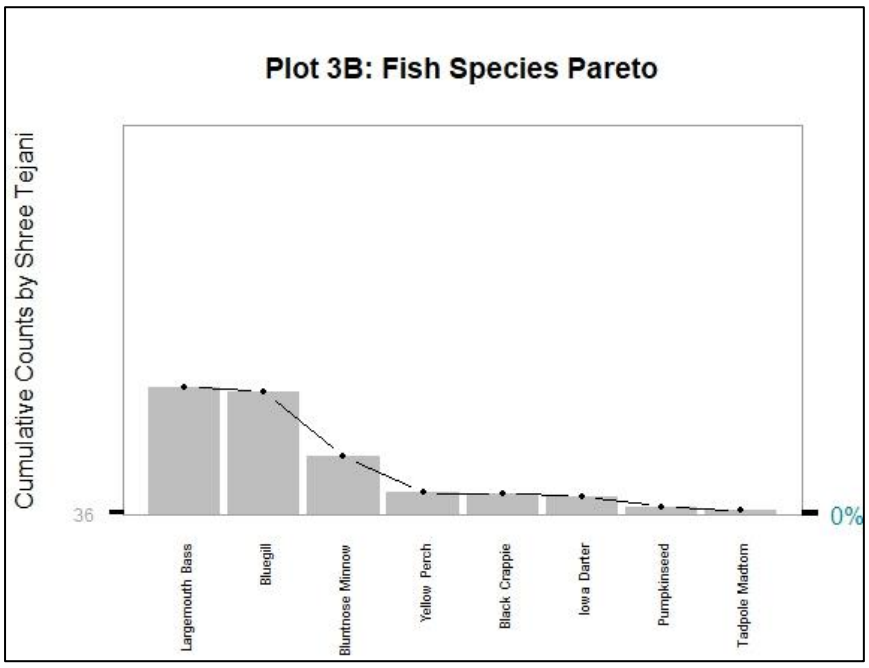
Displaying left side axis, by marking horizontal values at cumCts on side 2.



Displaying right side axis, by marking horizontal values at cumCts with % on side 4.



Displaying the whole Pareto plot



Summary

The above 'inchBio.csv' dataset helps us in understanding the basics of analytics by importing an excel file, creating a table and various objects from that dataset. It helps us understand the structure of the data and summarizing the information in just few lines. I have also created new variables for a table object that is further utilized in graphs.

We converted various objects and tables into dataframe for plotting various graphs like bar graph and Pareto plot. Creating data visualization is important for analytics as it helps viewers summarize a large dataset into graphical representation for easy interpretation for future uses and for prediction purposes.

Bibliograpghy

Kabacoff, R. I. (2015). *R in action: Data analysis and graphics with R*. Manning.

robk@statmethods.net, R. K.-. (n.d.). *Graphical parameters*. Quick-R: Graphical Parameters. Retrieved May 2, 2023, from https://www.statmethods.net/advgraphs/parameters.html

robk@statmethods.net, R. K.-. (n.d.). *Axes and text*. Quick-R: Axes and Text. Retrieved May 2, 2023, from https://www.statmethods.net/advgraphs/axes.html

Appendix

```
print("Shree Tejani")  # print author name

#importing the libraries

install.packages("FSA")

install.packages("FSAdata")

install.packages("magrittr")

install.packages("plotrix")

install.packages("ggplot2")

install.packages("moments")

install.packages("plyr")

install.packages("tidyverse")

# use installed libraries

library(FSA)

library(FSAdata)

library(magrittr)

library(dplyr)

library(tidyr)

library(plyr)

library(tidyverse)

library(plotrix)

library(ggplot2)

library(moments)

library(Hmisc)
```

```
#importing dataset from FSA library

tBio <- read.csv("D:\\MPS_Quater 1\\ALY6000_Intro to Analytics\\inchBio.csv",
header=TRUE, stringsAsFactors=FALSE)

tBio

str(tBio)   #structure of tBio

summary(tBio)   #summary of tBio


head(tBio,5) # head values for tBio

tail(tBio,5) # tail values for tBio


cts <- table(tBio$species) #obj 'CTS' to displays count

cts

class(cts)


names(cts)  # names of species


#temp1

temp1 <- subset(tBio, select = c("species"))

temp1 <- table(temp1)

temp1


#temp2

temp2 <- head(temp1,5)

temp2


#table t
```

```
t <- table(tBio[3])
t
class(t)

#transform t table to DF
df <- data.frame(t)
df
class(df)

# Q10
freq <- df$Freq
freq

#Q11
# Creating a new table named tSpec containing the species attribute of tBio
tSpec <- table(tBio$species)
tSpec
class(tSpec)

#Q12
tSpecPct <- (t/676)*100
tSpecPct
class(tSpecPct)

#Q13
```

```
dfSP <- as.data.frame(tSpecPct)
dfSP
class(tSpecPct)



#Q14 bar plot for tSpec
polt1 <- barplot(tSpec,main="Plot 1: Fish Species Counts",ylim = c(0,250),
      ylab = "Counts",col="grey",las=2,cex.axis=0.58)

#Q15
polt2 <- barplot(tSpecPct,main="Plot 2: Fish Relative Frequency",ylim = c(0,35),
            ylab = "%",col="cyan",las=2,cex.axis=0.58)

#Q16
data <- dfSP[order(-dfSP$Freq),]
data

#Q17
colnames(data) <- c("Species", "RelFreq")
data

#Q18
# Calculate the cumulative relative frequency
data$cumFreq <- cumsum(data$RelFreq)
# Add a column to data with the count of each species
```

```
data$cts <- data$RelFreq*676
data$cumCts <- cumsum(cts)
data



#Q19
varPar <- colnames(data)
varPar


#Q20
pc <- barplot(data$cts, width = 1, space = .1, border = NA,
        axes = FALSE, ylim = c(0, 3.05*max(data$cts, na.rm=TRUE)),
        ylab = "Cumulative Counts", names.arg = data$Species,col = "lightgreen",
        main = "Plot 3: Fish Species Pareto", las = 2, cex.names = 0.58)


#Q21
pc <- barplot(data$cts, width = 1, space = .1, border = NA,
        axes = FALSE, ylim = c(0, 3.05*max(data$cts, na.rm=T)),
        ylab = "Cumulative Counts", names.arg = data$Species,col = "lightgreen",
        main = "Plot 3B: Fish Species Pareto", las = 2, cex.names = 0.58)
lines(pc,data$cts,type = "b", cex = 0.75, pch = 20, col = "black")
box(col = "grey62") #Q22


#Q23
pc <- barplot(data$cts, width = 1, space = .1, border = NA,
```

```
        axes = FALSE, ylim = c(0, 3.05*max(data$cts, na.rm=TRUE)),
        ylab = "Cumulative Counts", names.arg = data$Species,
        main = "Plot 3D: Fish Species Pareto", las = 2, cex.names = 0.58, cex.axis = 0.75)
lines(pc,data$cts,type = "b", cex = 0.75, pch = 20, col = "black")
box(col = "grey62") #Q22
axis(side = 2, at = data$cumCts, col.ticks = "grey62",
     col = "grey62", cex.axis = 0.75,las = 2)

#Q24
pc <- barplot(data$cts, width = 1, space = .1, border = NA,
        axes = FALSE, ylim = c(0, 3.05*max(data$cts, na.rm=T)),
        ylab = "Cumulative Counts", names.arg = data$Species,
        main = "Plot 3D: Fish Species Pareto", las = 2, cex.names = 0.58)
lines(pc,data$cts,type = "b", cex = 0.75, pch = 20, col = "black")
box(col = "grey62") #Q22
axis(side = 4, at = c(0,data$cumCts), labels = paste(c(0,round(data$cumFreq * 100)),"%",sep
=""),
    col.lab = "cyan4", col.axis = "cyan4",
    cex.axis = 0.75, las = 2)

#Q25
pc <- barplot(data$cts, width = 1, space = .1, border = NA,
        axes = FALSE, ylim = c(0, 3.05*max(data$cts, na.rm=T)),
        ylab = "Cumulative Counts by Shree Tejani", names.arg = data$Species,
        main = "Plot 3B: Fish Species Pareto", las = 2, cex.names = 0.58)
lines(pc,data$cts,type = "b", cex = 0.75, pch = 20, col = "black")
```

```
box(col = "grey62") #Q22

axis(side = 2, at = data$cumCts, labels = data$cumCts,

    col.axis = "grey62", col.lab = "grey62", cex.axis = 0.75,las = 2)

axis(side = 4, at = c(0,data$cumCts), labels = paste(c(0,round(data$cumFreq * 100)),"%",sep
="",

    col.lab = "cyan4", col.axis = "cyan4",

    cex = 0.75, las = 2)
```