Executive Summary Report 3

12th October 2021

HARSHIT GAUR

NUID: 001093079

To: Prof. Richard He

KEY FINDINGS

1. Import of Libraries & Load of 'inchBio.csv' data set

```
#STEP 1A: Printing my name.
print("HARSHIT GAUR")

#STEP 1B: Importing the packages.
library("FSA")
library("FSAdata")
library("magrittr")
library("dplyr")
library("tidyr")

install.packages("plyr")
install.packages("tidyverse")
library("plyr")
library("tidyverse")
```

Figure 1.1

	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
11	13	71	Bluegill	34	0.5		FALSE
12	13	73	Bluegill	40	1.0		FALSE

Figure 1.3

Data	
bio	676 obs. of 7 variables

Figure 1.2

```
#STEP 2: Import 'inchBio.csv' data set
#Note: Change the working directory as per the file's location.
setwd("/Users/HarshitGaur/Documents/Northeastern University/MPS
bio <- read.csv("inchBio.csv", header = TRUE)
View(bio)
```

Figure 1.4

a. The libraries - FSA, FSAdata, magrittr, dplyr, tidyr, plyr, and tidyverse were installed and imported successfully in the project.
b. ***inchBio*** CSV data set was loaded into the memory of the project and used further with the name '*bio*'. A screenshot has been attached of a snippet of the data present in it.

2. Operations on the data set

```
#STEP 3A: Print the head of 'inchBio.csv' data set
View(head(bio))
```

Figure 2.1

```
#STEP 3B: Print the tail of 'inchBio.csv' data set
View(tail(bio))
```

Figure 2.2

	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	44	Bluegill	42	1.0		FALSE
6	12	65	Bluegill	54	2.1		FALSE

Figure 2.3

	netID	fishID	species	tl	w	tag	scale
671	121	808	Black Crappie	323	509	1050	TRUE
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

Figure 2.4

a. Basic R operations on the data set were performed.
    i. *head(x)* function allows to select first *6 (default)* records from *x* data set.
    ii. *tail(x)* function allows to select last *6 (default)* records from *x* data set.

3. Listing the '*species*' variable from the data set

```
#STEP 4A: List the species of 'inchBio.csv' data set
#speciesList <- bio[,3]
speciesList <- list(bio$species)
speciesList
```

Figure 3.1

```
[213] "Bluegill"          "Bluegill"          "Bluegill"          "Bluegill"
[217] "Bluegill"          "Bluegill"          "Bluegill"          "Bluegill"
[221] "Bluntnose Minnow" "Bluntnose Minnow" "Bluntnose Minnow" "Bluntnose Minnow"
[225] "Bluntnose Minnow" "Bluntnose Minnow" "Bluntnose Minnow" "Bluntnose Minnow"
```

Figure 3.2

a. List functions can be used to list out the variables or the whole data set
    i. *list(x, sorted)* function allows to list either the variable from data set with second argument as the ordering parameter of the new list.

4. Structure of the inchBio data set

```
> #STEP 3C: Print the structure of 'inchBio.csv' data set
> str(bio)
'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 ...
```

Figure 4.1

a. The inchBio data set contains 676 observations with 7 variables in it about the different species of fish available.
b. The variables *'netID'* , *'fishID', & 'tl'* contain observations in integer data type.
c. The variables *'species'* & *'tag'* contain observations in character data type. The elements in *'species'* will be converted implicitly & internally into *'Factors'* by the R-script for efficiency.

5. Frequency of the '*species*' variable from the data set

```
> #STEP 4B: Count the records of species of 'inchBio.csv'
> #counts <- length(bio$species)
> #counts <- count(bio$species)
> counts <- table(bio$species)
> View(counts)
> is.object(counts)
[1] TRUE
```

Figure 5.1

	x	freq
1	Black Crappie	36
2	Bluegill	220
3	Bluntnose Minnow	103
4	Iowa Darter	32
5	Largemouth Bass	228
6	Pumpkinseed	13
7	Tadpole Madtom	6
8	Yellow Perch	38

Figure 5.2

```
> #STEP 5: Display the 8 levels of species of 'inchBio.csv' data set
> speciesLevel <- unique(bio$species)
> speciesLevel
[1] "Bluegill"         "Bluntnose Minnow" "Iowa Darter"      "Largemouth Bass"
[5] "Pumpkinseed"      "Tadpole Madtom"   "Yellow Perch"     "Black Crappie"
```

Figure 5.3

a. A list of elements of 'species' has been generated. We also have created a frequencies table of all the *8 levels of species* from the data set.
    i. *count(x, wt = NULL, sort = FALSE), table(x)* and *unique(x, incomparables = FALSE)* functions allow to get the frequency table of the variable of a data set and the list of unique elements in the variable of a data set respectively.

6. Selecting & Sub-setting the first 5 levels of '*species*' variable from the data set

```
#STEP 7: Create variable to Display the subset of first 5 levels of species
tmp2 <- subset(head(bio, 5), select = "species")
View(tmp2)
```

Figure 6.1

	species
1	Bluegill
2	Bluegill
3	Bluegill
4	Bluegill
5	Bluegill

Figure 6.2

a. Subset functions can be used to select a subset from the data set with a parameter to even choose which variable needs to be used for the subset.
    i. *subset(x, select=)* function allows to make subset of *'select'* variable from *x* data set. We have modified the x data set to contain first 5 records only.

7. Data set conversions - Tables, Data Frames

```
> #STEP 8A: Create a table containing species
> w <- table(bio$species)
> View(w)
> #STEP 8B: Display the class of above table
> class(w)
[1] "table"
```

Figure 7.1

	Var1	Freq
1	Black Crappie	36
2	Bluegill	220
3	Bluntnose Minnow	103
4	Iowa Darter	32
5	Largemouth Bass	228
6	Pumpkinseed	13
7	Tadpole Madtom	6
8	Yellow Perch	38

Figure 7.2

```
> #STEP 9A: Convert the above table to data frame
> t <- data.frame(w)
> #STEP 9B: Class of the data frame 't'
> class(t)
[1] "data.frame"
> #STEP 9C: Structure of the data frame 't'
> str(t)
```

Figure 7.3

```
> str(t)
'data.frame':   8 obs. of  2 variables:
 $ Var1: Factor w/ 8 levels "Black Crappie",..: 1 2 3 4 5 6 7 8
 $ Freq: int  36 220 103 32 228 13 6 38
> #STEP 9D: Summary of the data frame 't'
> summary(t)
                Var1        Freq
 Black Crappie   :1   Min.   :  6.00
 Bluegill        :1   1st Qu.: 27.25
 Bluntnose Minnow:1   Median : 37.00
 Iowa Darter     :1   Mean   : 84.50
 Largemouth Bass :1   3rd Qu.:132.25
 Pumpkinseed     :1   Max.   :228.00
 (Other)         :2
```

Figure 7.4

a. Few operations of data type conversions were performed on the data set. A table and a data frame were created from the data set whose class and structure were confirmed.
b. The frequency values are also present in *Figure 6.4* for all the 8 levels of species. We have also found out the statistics of this data frame of species.

8. Prop Table (with Percentage)

```
> #STEP 11: Create a table 'cSpec' from species and confirm with
 class and View
> cSpec <- table(bio$species)
> class(cSpec)
[1] "table"
> View(cSpec)
> #STEP 12: Create a table 'cSpecPct' displaying the species and
 its percentages (not frequencies)
> cSpecPct <- prop.table(cSpec) * 100
> class(cSpecPct)
[1] "table"
> View(cSpecPct)
```

Figure 8.1

	Var1	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

Figure 8.2

a. A Prop table has been generated to contain the bio-species attribute from the data set and the percentage of records for each species.
b. The table has been multiplied with *100 (hundred)* to make the frequency table as a percentage table.

9. Bar Plot for *Fish Count*

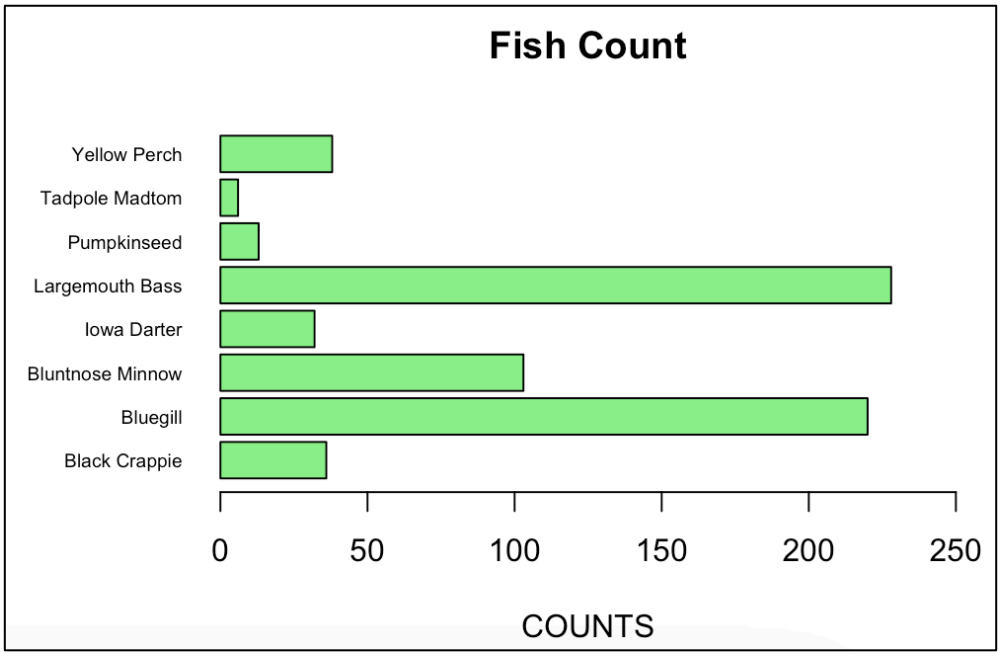

Figure 9.1 - Plot 1: Fish Count

a. The bar plot above shows the fish count of each fish species available. It depicts that the largest frequency of any kind of species available from the data set is around **230**.
b. We can observe that the *Largemouth Bass* and *Bluegill* species of fish are the most abundant fish species found and *Tadpole Madtom* and *Pumpkinseed* are the least abundant fish species found.

10. Bar Plot for *Fish Relative Frequency (Percentage, NOT Fraction)*

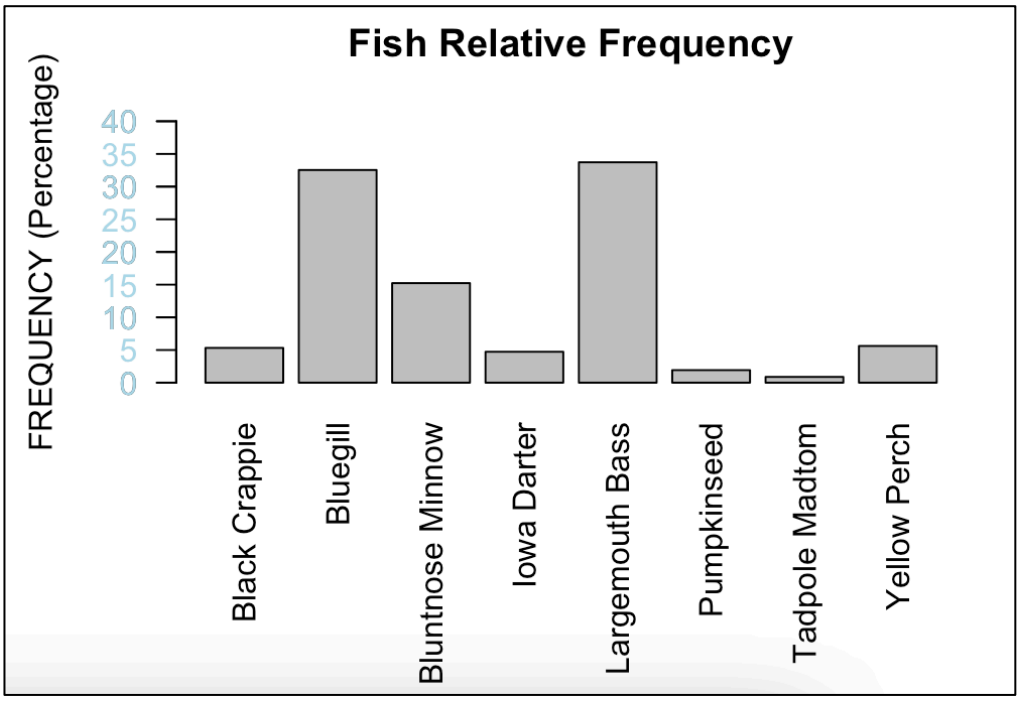

Figure 10.1 - Plot 2: Fish Relative Frequency

a. The bar plot depicts that the frequencies of *Largemouth Bass* and *Bluegill* species of fish are the most amongst the species of fish available at around *33% (0.33 - fraction/decimal).*
b. It also shows that the least frequencies are of the species - *Tadpole Madtom* and *Pumpkinseed*.
c. X-axis belongs to species to fish and Y-axis belongs to Frequency (in percentage) of fishes.

11. Ordering of Frequency, Column-names in the data frame

```
#STEP 16: Rearrange the 'u' data frame
#with decreasing order of frequency
d <- u[order(u$Freq, decreasing = TRUE), ]
View(d)
```

Figure 11.1

```
#STEP 17: Rename the columns of 'd' data frame
colnames(d) <- c("Species", "RelFreq")
View(d)
```

Figure 11.2

	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

Figure 11.3

12. Adding variables *'cumfreq'*, *'counts'*, and *'cumcounts'* to data frame

```
#STEP 18: Add 'cumfreq', 'counts', 'cumcounts' to the 'd' data frame
d <- mutate(d, cumfreq = cumsum(d$RelFreq))
d <- merge(d, counts, by.x = "Species", by.y = "x", sort = FALSE)
d <- rename(d, replace = c("freq" = "counts"))
d <- mutate(d, cumcounts = cumsum(d$counts))
View(d)
```

Figure 12.1

	Species	RelFreq	cumfreq	counts	cumcounts
1	Largemouth Bass	33.727811	33.72781	228	228
2	Bluegill	32.544379	66.27219	220	448
3	Bluntnose Minnow	15.236686	81.50888	103	551
4	Yellow Perch	5.621302	87.13018	38	589
5	Black Crappie	5.325444	92.45562	36	625
6	Iowa Darter	4.733728	97.18935	32	657
7	Pumpkinseed	1.923077	99.11243	13	670
8	Tadpole Madtom	0.887574	100.00000	6	676

Figure 12.2

a. The new variable *'cumfreq'* denotes the cumulative frequency of the 8 fish species cumulating from the highest frequency holding record to the least.
b. *'counts'* variable contains the total number of records each species has in the data set.
c. The variable *'cumcounts'* signifies the cumulative count of the 8 fish species cumulating the same way as the *'cumfreq'* does.

13. Summary of the *inchBio* data set

```
> #STEP 3D: Print the summary of 'inchBio.csv' data set
> summary(bio)
     netID            fishID          species                tl               w         
 Min.   :  1.00   Min.   :  7.0   Length:676         Min.   : 27.0   Min.   :   0.2  
 1st Qu.: 13.00   1st Qu.:175.8   Class :character   1st Qu.: 66.0   1st Qu.:   2.0  
 Median : 37.00   Median :345.5   Mode  :character   Median :189.5   Median :  54.5  
 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     

    tag              scale          
 Length:676         Mode :logical  
 Class :character   FALSE:213      
 Mode  :character   TRUE :463      
```

Figure 13.1

a. netID & fishID :
   i. These are the identification numbers with relationships to other tables/data sets.
   ii. The statistics (median, mean, etc.) values have been found out but without proper knowledge of these fields, we cannot infer anything on them.

b. species :
    i. There are **676 number of fish records** in the data set each belonging to a type of species.
    ii. The **number of species** in the data set is **8.**
    iii. The data type of species in *character*

c. w, tl, tag, scale :
    i. We don't have any prior knowledge of what these variable signify to.
    ii. We can only figure out the data types and statistics of these variables from the data set.
    iii. There are some *empty values and NAs present in tl and w* respectively.

14. Pareto Plot for *Species Pareto*

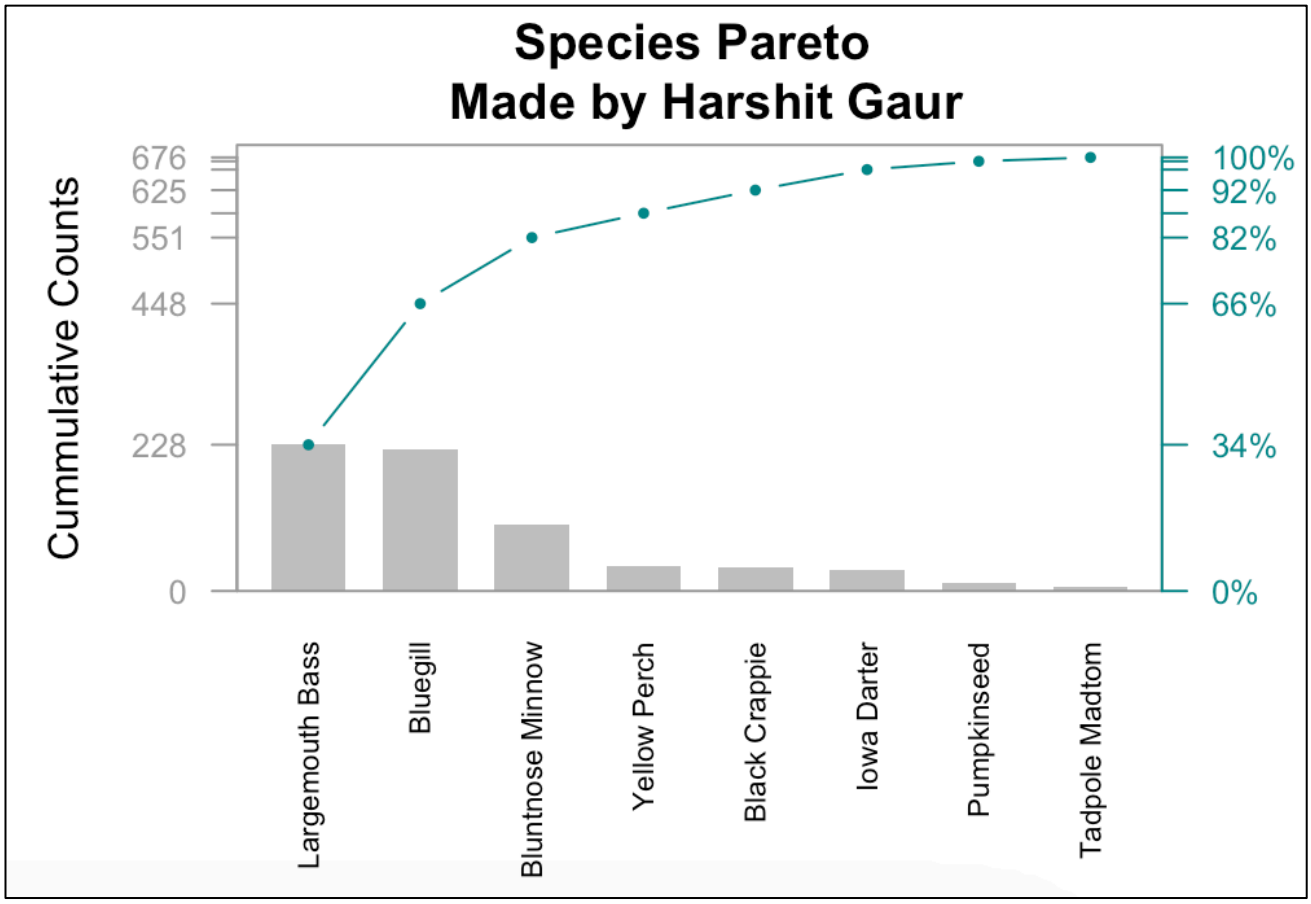

Figure 14.1 Plot 3: Species Pareto

a. In this pareto plot, the height (Y-axis) of the graph depicts the *'Cumulative Count'* on the left side and *'Cumulative Frequency'* on the right side. The X-axis belongs to the fish species available in the data set.
b. We analysed that :
    i. Around **70% of the fish population** belong to **25% of the species** available in the data set, i.e., **70% of the fish population (*228 + 220*)** belong to the species **(*Largemouth Bass* and *Bluegill*)** which constitute to around **25% of the species**.
    ii. The species (***Pumpkinseed and Tadpole Madtom***) constitute a very small percentage of the fish population.

SUMMARY

1. We were able to find out the statistics of the inchBio data set and analysed the following :
    a. The variable ***'species'*** constituted of 8 levels of fish species spread amongst 676 observations of fishes.
    b. The variables ***'netID, fishID, tl, w, tag, scope'*** belong to the identification numbers, weight, or unknown parameters whose information could not be retrieved directly from the data set. Some of the variables hold foreign relationships with other tables/ data set.
    c. Statistics (mean, median, quartiles, etc.) were figured out from the data set but couldn't signify that much due to non-information of the variables itself.

2. The graphs plotted helped us infer that the fishes belong adversely to the species. ***Largemouth Bass and Bluegill*** species hold the most abundant population of fishes amongst themselves.

3. ***Pumpkinseed and Tadpole Madtom*** constitute a very small amount of population of fishes amongst themselves.

4. The Pareto Chart (Species Pareto) also helped us figure out a pattern :
    a. Almost 70% - 80% of the fish population belong to 20% - 25% of the fish species available in the data set.

BIBLIOGRAPHY

1. *Make table show percentages instead of frequencies in R*. (2014, August 27). Stack Overflow. https://stackoverflow.com/questions/25517013/make-table-show-percentages-instead-of-frequencies-in-r

2. *R is plotting labels off the page*. (2010, May 10). Stack Overflow. https://stackoverflow.com/questions/2807060/r-is-plotting-labels-off-the-page

3. *How to rename a single column in a data.frame?* (2011, September 23). Stack Overflow. https://stackoverflow.com/questions/7531868/how-to-rename-a-single-column-in-a-data-frame

4. S., K., Munyengwa, N., Lotfi, S., K., Kumar, A., K., T., K., T., M., Chan, F. K., K., Chan, F. K., & Sarr, A. (2018, October 19). *Rename Data Frame Columns in R*. Datanovia. https://www.datanovia.com/en/lessons/rename-data-frame-columns-in-r/

5. Coder, R. (2020, December 22). *Plot in R*. R CODER. https://r-coder.com/plot-r/

6. *Pareto Chart With Base R Plotting System*. (2021). Pareto Plot | AWS. https://rstudio-pubs-static.s3.amazonaws.com/72023_670962b57f444c04999fd1a0a393e113.html

7. *Pareto Analysis: Choosing the Solution With the Most Impact*. (2021). Mind Tools. https://www.mindtools.com/pages/article/newTED_01.htm

APPENDIX

```
#---------------------- GAUR_M3_PROJECT3 ----------------------#

#STEP 1A: Printing my name.
print("HARSHIT GAUR")

#STEP 1B: Importing the packages.
library("FSA")
library("FSAdata")
library("magrittr")
library("dplyr")
library("tidyr")

install.packages("plyr")
install.packages("tidyverse")
library("plyr")
library("tidyverse")

#STEP 2: Import 'inchBio.csv' data set
#Note: Change the working directory as per the file's location.
setwd("/Users/HarshitGaur/Documents/Northeastern University/MPS Analytics/ALY 6000/Class 3/Assignment")
bio <- read.csv("inchBio.csv", header = TRUE)
View(bio)

#STEP 3A: Print the head of 'inchBio.csv' data set
View(head(bio))
#STEP 3B: Print the tail of 'inchBio.csv' data set
View(tail(bio))
#STEP 3C: Print the structure of 'inchBio.csv' data set
str(bio)
#STEP 3D: Print the summary of 'inchBio.csv' data set
summary(bio)

#STEP 4A: List the species of 'inchBio.csv' data set
#speciesList <- bio[,3]
speciesList <- list(bio$species)
speciesList

#STEP 4B: Count the records of species of 'inchBio.csv' data set
#counts <- length(bio$species)
#counts <- count(bio$species)
counts <- table(bio$species)
View(counts)
is.object(counts)

#STEP 5: Display the 8 levels of species of 'inchBio.csv' data set
speciesLevel <- unique(bio$species)
speciesLevel

#STEP 6: Create variable to Display the levels of species & their frequencies
tmp <- count(bio$species)
View(tmp)

#STEP 7: Create variable to Display the subset of first 5 levels of species
tmp2 <- subset(head(bio, 5), select = "species")
View(tmp2)

#STEP 8A: Create a table containing species
w <- table(bio$species)
View(w)
#STEP 8B: Display the class of above table
class(w)

#STEP 9A: Convert the above table to data frame
t <- data.frame(w)
#STEP 9B: Class of the data frame 't'
class(t)
#STEP 9C: Structure of the data frame 't'
str(t)
#STEP 9D: Summary of the data frame 't'
summary(t)
```

```
#STEP 10: Display the 'frequency' values from the data frame 't'
t$Freq

#STEP 11: Create a table 'cSpec' from species and confirm with class and View
cSpec <- table(bio$species)
class(cSpec)
View(cSpec)

#STEP 12: Create a table 'cSpecPct' displaying the species and its percentages (not frequencies)
cSpecPct <- prop.table(cSpec) * 100
class(cSpecPct)
View(cSpecPct)

#STEP 13A: Convert the table 'cSpecPct' to data frame
u <- data.frame(cSpecPct)
#STEP 13B: Class of the data frame 'u'
class(u)

#STEP 14: Plot a Barplot of 'cSpec'
# ---------- Plot 1: Fish Count ---------- #
par(mar = c(5, 6, 4, 2) + 0.1)
barplot(cSpec, main = "Fish Count", xlab = "COUNTS", las = 1, horiz = TRUE,
      cex.names = 0.6, xlim = c(0,250), col = 'LIGHTGREEN')
#barplot(cSpec, main = "Fish Count", ylab = "COUNTS", las = 2, horiz = FALSE, cex.names = 0.6, ylim = c(0,250), col =
'LIGHTGREEN')


#STEP 15: Plot a Barplot of 'cSpecPct'
# ---------- Plot 2: Fish Relative Frequency ---------- #
par(mar = c(8, 5, 4, 2) + 0.1)
barplot(cSpecPct, main = "Fish Relative Frequency", ylab = "FREQUENCY (Percentage)", las = 2, horiz = FALSE, ylim = c(0,40))
axis(side = 2, at = 5*(0:8), label = 5*(0:8), col.axis = "LIGHTBLUE", cex.axis = 1, las = 2)
#Both axes as LightBlue
#barplot(cSpecPct, main = "Fish Relative Frequency", ylab = "FREQUENCY", las = 2, horiz = FALSE, cex.names = 1, ylim = c(0,40),
col.axis = 'LIGHTBLUE')

#STEP 16: Rearrange the 'u' data frame
#with decreasing order of frequency
d <- u[order(u$Freq, decreasing = TRUE), ]
View(d)

#STEP 17: Rename the columns of 'd' data frame
colnames(d) <- c("Species", "RelFreq")

#STEP 18: Add 'cumfreq', 'counts', 'cumcounts' to the 'd' data frame
d <- mutate(d, cumfreq = cumsum(d$RelFreq))
d <- merge(d, counts, by.x = "Species", by.y = "Var1", sort = FALSE)
d <- rename(d, replace = c("Freq" = "counts"))
d <- mutate(d, cumcounts = cumsum(d$counts))
View(d)

#STEP 19: Define variables for parameter variables
def_par <- 3.05 * max(d$counts)

#STEP 20: Plot a Barplot of 'pc'
# ---------- Plot 3: Species Pareto ---------- #
par(mar = c(7, 6, 3, 3))
pc <- barplot(d$counts,  main = "Species Pareto \n Made by Harshit Gaur", width = 1, space = 0.5, border = NA, axes = F, ylim = c(0,
def_par),
          na.rm = TRUE, ylab = "Cummulative Counts", cex.names = 0.7, names.arg = d$Species, las = 2)

#STEP 21: Add cummulative counts line to the plot
lines(pc, d$cumcounts, type = "b", cex = 0.7, pch = 20, col = "CYAN4")

#STEP 22: Place a grey box around the pareto plot
box(col = "GREY62")

#STEP 23: Add a left-side axis
axis(side = 2, at = c(0, d$cumcounts), col.axis = "GREY62", col = "GREY62", cex.axis = 0.8, las = 1)

#STEP 24: Add a right-side axis
axis(side = 4, at = c(0, d$cumcounts), labels = paste( c(0, round(d$cumfreq)), "%", sep = "" ), col.axis = "CYAN4", col = "CYAN4",
cex.axis = 0.8, las = 1)
```