



EXECUTIVE SUMMARY REPORT 3

Module 3 assignment

Abstract

In this assignment, I will create a summary of some data and learn how to use R

Mohammad Hossein Movahedi

Movahedi.m@northeastern.edu

Introduction

In this summary, I will use the data from the inchBio.csv dataset and use the APA system for the bibliography. Also, the R code that I used to process the data is available on my GitHub account, which address is mentioned in the bibliography part. I also put my code in the appendix part.

The objective is to firstly, Follow an introduction, provide an analysis of descriptive characteristics of the data set provided by your instructor. This includes pertinent statistics including counts, cumulative counts, frequency, percentages, etc. Include R console screen snippets to support your observations and conclusions. Secondly, Provide the executive with visualizations (at least 3) that help them see the key characteristics you want to highlight. Lastly, provide a clear two to three sentences paragraph summary of the key points.

methodology

1. first of all I add my name and imported libraries

```
#printing my name
print("mohammad Hossein Movahedi")
#importing and loading required libraries
install.packages("FSA")
install.packages("FSAdata")
install.packages("magrittr")
install.packages("dplyr")
install.packages("tidyr ")
install.packages("plyr")
install.packages("tidyverse")
#loading installed libraries
library(FSA)
library(FSAdata)
library(magrittr)
library(dplyr)
library(tidyr)
library(plyr)
library(tidyverse)
```

2. then I imported the CSV file as a table named Bio using what I learnt from (Willems, 2018)

```
#Import the inchBio.csv and name the table bio
bio <- read_csv("/cloud/project/inchBio.csv")
```

3. then I displayed head and tail and structure of bio as it can be seen in results at first glance that data has seven columns and some of the data are missing.

```
#displaying head and tail and structure of bio
headtail(bio,n = 5)
str(bio)
```

| | netID | fishID | species | tl | w | tag | scale |
|-----|-------|--------|---------------|-----|-------|------|-------|
| 1 | 12 | 16 | Bluegill | 61 | 2.9 | <NA> | FALSE |
| 2 | 12 | 23 | Bluegill | 66 | 4.5 | <NA> | FALSE |
| 3 | 12 | 30 | Bluegill | 70 | 5.2 | <NA> | FALSE |
| 4 | 12 | 44 | Bluegill | 38 | 0.5 | <NA> | FALSE |
| 5 | 12 | 50 | Bluegill | 42 | 1.0 | <NA> | FALSE |
| 672 | 121 | 809 | Black Crappie | 282 | 352.0 | 1700 | TRUE |
| 673 | 121 | 812 | Black Crappie | 142 | 37.0 | <NA> | TRUE |
| 674 | 110 | 863 | Black Crappie | 307 | 415.0 | 1783 | TRUE |
| 675 | 129 | 870 | Black Crappie | 279 | 344.0 | 1789 | TRUE |
| 676 | 129 | 879 | Black Crappie | 302 | 397.0 | 1792 | TRUE |

```
$ netID : num [1:676] 12 12 12 12 12 12 12 12 13 13 13 ...
$ fishID : num [1:676] 16 23 30 44 50 65 66 68 69 70 ...
$ species: chr [1:676] "Bluegill" "Bluegill" "Bluegill" "Bluegill" ...
$ tl      : num [1:676] 61 66 70 38 42 54 27 36 59 39 ...
$ w       : num [1:676] 2.9 4.5 5.2 0.5 1 2.1 NA 0.5 2 0.5 ...
$ tag     : chr [1:676] NA NA NA NA ...
$ scale   : logi [1:676] FALSE FALSE FALSE FALSE FALSE FALSE ...
- attr(*, "spec")=
.. cols(
..   netID = col_double(),
..   fishID = col_double(),
..   species = col_character(),
..   tl = col_double(),
..   w = col_double(),
..   tag = col_character(),
..   scale = col_logical()
.. )
- attr(*, "problems")=<externalptr>
```

4. creating an object that counts and lists all the species records to do this I used a for loop on species categories which I learned in detail from (Neonscience.org, 2021)

```
#creating an object that counts and lists all the species records
ubio <- unique(bio$species[!is.na(bio$species)])
list <- vector()
counts <- vector()
for (i in ubio){
```

```

list <- c(list,i)
counts <- c(counts, length(which(bio$species==i)))
}
countlist = data.frame(lists = list,count = counts)

```

5. Displaying 8 names of species

```

#displaying names of species
disp <- headtail(countlist,n = 4)
print(disp$lists)

```

```

[1] "Bluegill"          "Bluntnose Minnow" "Iowa Darter"
[4] "Largemouth Bass"  "Pumpkinseed"      "Tadpole Madtom"
[7] "Yellow Perch"     "Black Crappie"

```

6. In this step we have to create a tmp object that stores the number of each species

```

tmp <- countlist

```

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

As it can be seen most of the data are Largemouth Bass

7. creating a subset of species

```

#Create a subset of just the species variable and display the first five records
tmp2 <- tmp$lists
head(tmp2,n=5)

```

```

[1] "Bluegill"          "Bluntnose Minnow" "Iowa Darter"
[4] "Largemouth Bass"  "Pumpkinseed"

```

8. creating w as a table form of tmp

```
#Create a table, <w>, of the species variable. Display the class of w
w <- table(tmp$lists)
class(w)
```

9. Converting <w> to a data frame named <t> and displaying the results to do this I used (Dummies.com, 2022)

```
# Convert <w> to a data frame named <t> and display the results
t <- as.data.frame(w)
print(t)
```

| | Var1 | Freq |
|---|------------------|------|
| 1 | Black Crappie | 1 |
| 2 | Bluegill | 1 |
| 3 | Bluntnose Minnow | 1 |
| 4 | Iowa Darter | 1 |
| 5 | Largemouth Bass | 1 |
| 6 | Pumpkinseed | 1 |
| 7 | Tadpole Madtom | 1 |
| 8 | Yellow Perch | 1 |

10. Frequency values are extracted and displayed from the <t> data frame

```
#Extract and display the frequency values from the <t> data frame
freq <- t$Freq
print(freq )
```

```
[1] 1 1 1 1 1 1 1 1
```

11. Using the bio species attribute (variable), create a table named <cSpec>, and verify that it shows the number of species in your dataset <bio>.in order to do this I used comparef() form the arsenal library. (Ebner, 2018) (Heinzen, 2017)

```
# creating cSpec
cSpec <- table(bio$species)
tcSpec <- as.data.frame(cSpec)
tcountlist <- as.data.frame(countlist)
tcountlist <- select(tcountlist,Var1 = lists,Freq = count)
comparedf (tcountlist,tcSpec)
```

Compare Object

Function Call:

```
comparedf(x = tcountlist, y = tcSpec)
```

Shared: 2 non-by variables and 8 observations.

Not shared: 0 variables and 0 observations.

Differences found in 2/2 variables compared.

0 variables compared have non-identical attributes.

The results show no difference between the two data frames confirming that they are identical.

12. Using the table class name <cSpecPct>, then create a table that displays the species and percentage of records for each species. And finally Ensuring the table class has been created. (Statistics Globe, 2021)

```
# Create a table named <cSpecPct>
cSpecPct <- table(bio$species)
cSpecPct <- cSpecPct/length(bio$species)
class(cSpecPct)
```

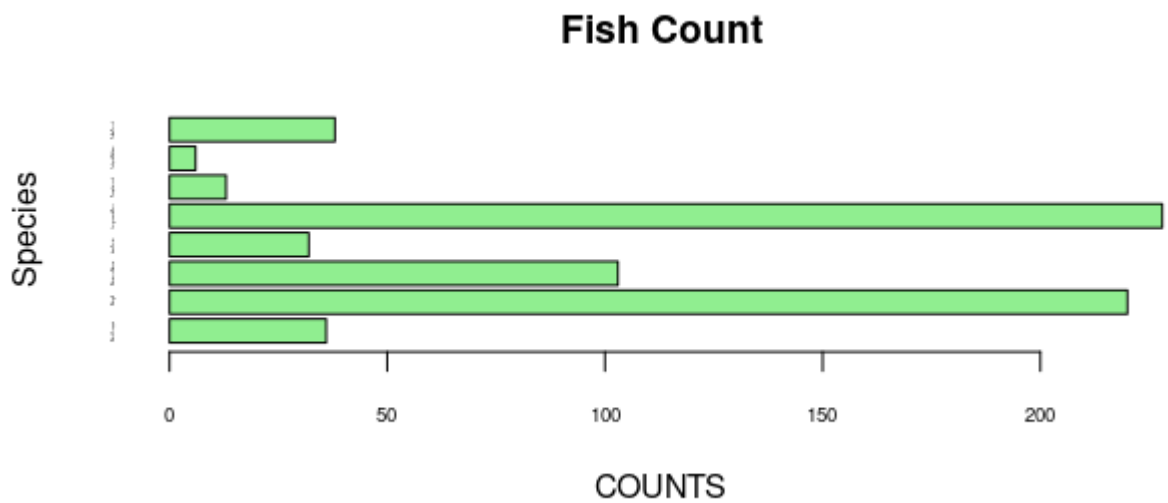
13. Table <cSpecPct> is converted into a data frame named <u> and confirmed that <u> is a data frame

```
#Convert the table, <cSpecPct>, to a data frame named <u>
u <- as.data.frame(cSpecPct)
class(u)
```

14.creating Fish count bar plot (in, 2010) (DataMentor, 2017)

```
#Create a barplot of <cSpec>
barplot(cSpec,
  main = "Fish Count",
  xlab = "COUNTS",
  ylab = "",
  cex.axis=0.6,
  cex.names=0.7,
  col = "Light Green",
  horiz = TRUE,
```

las = 1)

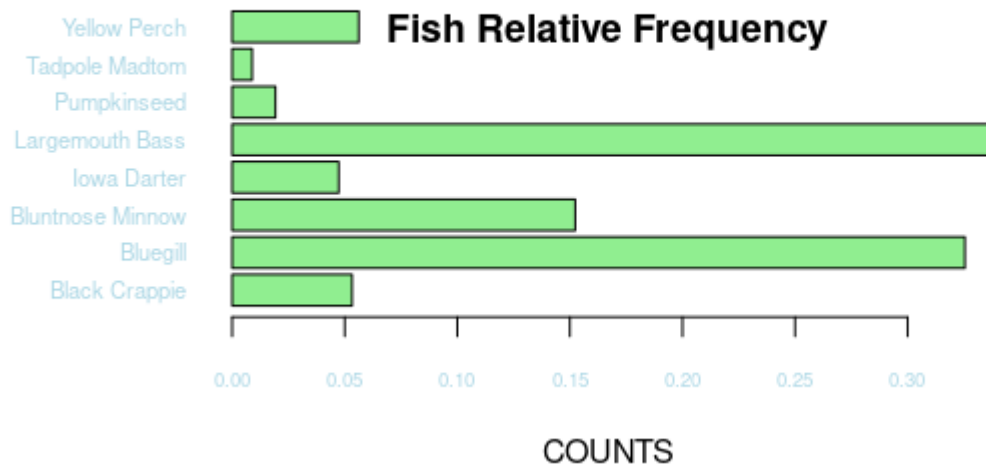


As it can be seen in the bar chart two species are far more counted in our sample than others

(Holtz, 2018)

15. creating Fish Relative Frequency bar chart

```
#Create a barplot of <cSpecPct>
par(mar=c(4,6,8,8))
barplot(cSpecPct,
        main = "Fish Relative Frequency",
        xlab = "COUNTS",
        ylab = "",
        ylim = range(0,4),
        cex.axis=0.6,
        cex.names=0.7,
        col = "Light Green",
        col.axis = "Light Blue",
        horiz = TRUE,
        las = 1)
```



As it can be seen Fish count and Fish Relative Frequency are pretty similar.

16. rearranging u as d (Datanovia, 2018)

```
#Rearrange the <u> cSpec Pct data frame in descending order as <d>
d <- u %>% arrange(desc(Freq))
```

17. renaming d columns

```
#remaing d cols
d <- select(d, Species = Var1 , RelFreq = Freq)
```

18. adding three columns to d (Tutorialspoint.com, 2020)

```
#adding new vars to d
d <- data.frame(d, cumfreq =
rep(c(NA), times=8), counts=rep(c(NA), times=8), cumcounts = rep(c(NA), times=8))
```

19. Create a parameter variable <def_par> to store parameter variables

```
def_par = c()
```

20. plotting Species Pareto (R-bloggers, 2012) (GeeksforGeeks, 2021) (R CODER, 2020)

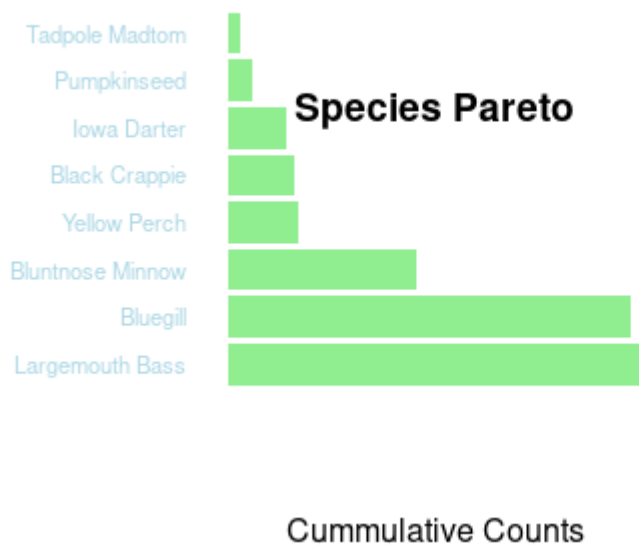
```
#creating <pc> bar chart
Pc <- barplot(d$RelFreq,
width = 1, space = 0.15,
main = "Species Pareto",
xlab = "Cummulative Counts",
ylab = "",
```



```

ylim = c(0,3.05),
names.arg = d$Species,
cex.axis=0.6,
cex.names=0.7,
col = "Light Green",
col.axis = "Light Blue",
horiz = TRUE,
las = 2
)

```



21. Adding a cumulative counts line to the <pc> plot (Amazonaws.com, 2022)

```

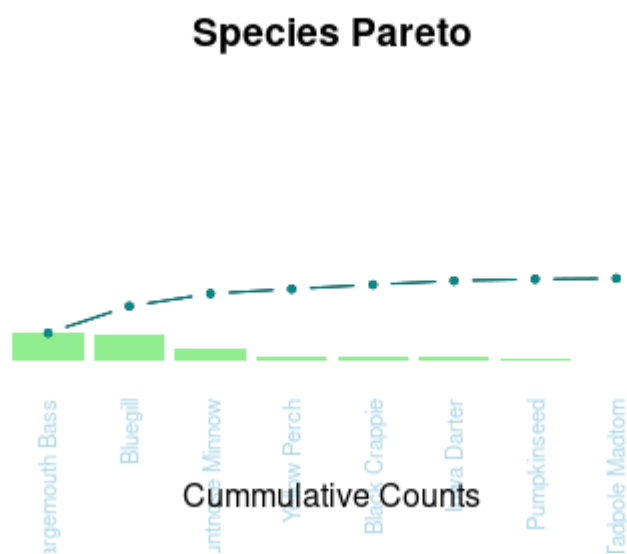
par(def_par)
par(mar=c(5,3,4,4))
barplot(d$RelFreq,
        width = 1, space = 0.15, axes = F, border = NA,
        main = "Species Pareto",
        xlab = "Cummulative Counts",
        ylab = "",
        ylim = c(0,3.05),
        names.arg = d$Species,
        cex.axis=0.6,
        cex.names=0.7,
        col = "Light Green",

```

```

col.axis = "Light Blue",
horiz = FALSE,
las = 2)
d <- arrange(d, desc(RelFreq)) %>%
mutate(
  cumsum = cumsum(RelFreq),
  freq = round(RelFreq / sum(RelFreq), 3),
  cum_freq = cumsum(freq)
)
lines(pc, d$cumsum, type = "b", cex = 0.7, pch = 20, col="cyan4")

```

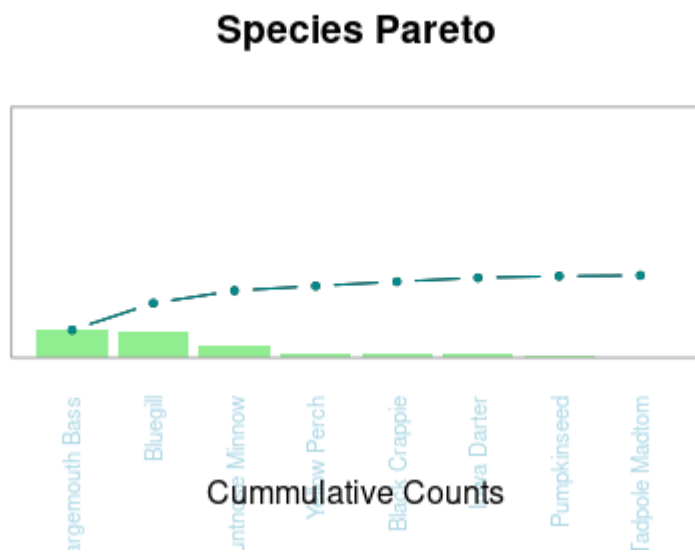


22. Place a grey box around the pareto plot

```

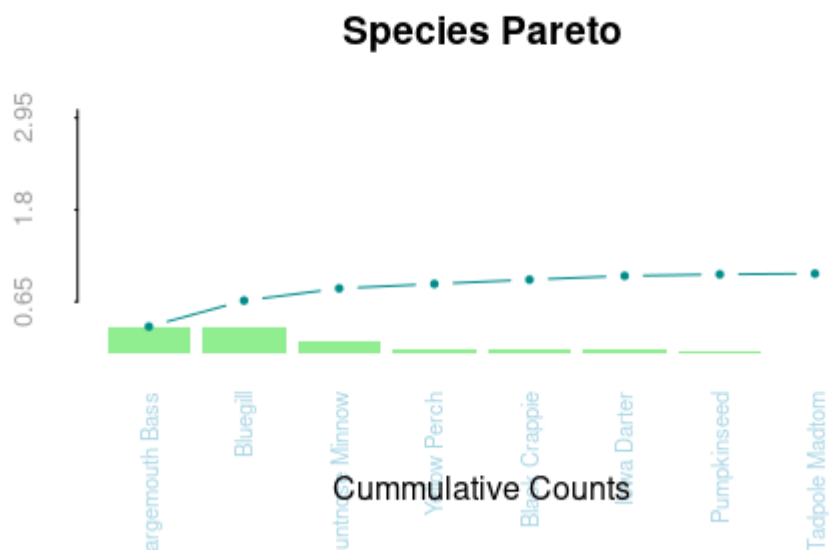
#placing a box around the chart
box(col = "grey62")

```



23. Add a left side axis (Statmethods.net, 2017)

```
# Add a left side axis
axis(2, at=pc, labels=pc, col.axis="grey62", cex.axis=0.8, tck=-.01, las=0)
```

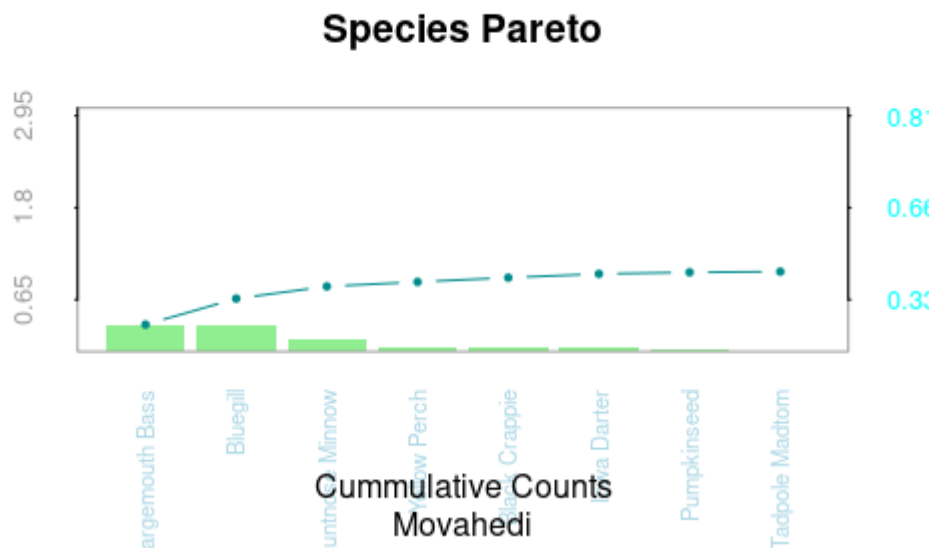


24. Add axis details on right side of box with the specifications

```
#Add axis details on right side of box with the specifications
```

```
axis(4, at=pc, labels=d$cum_freq,
     col.axis="cyan", las=2, cex.axis=0.8, tck=-.01)
```

25. displaying final result



Summery

In summery I got very deep knowledge about working with plots and R . I learned how to manipulate data with R and usage of dpleyr package

Bibliography

Willems, K. (2018). This R Data Import Tutorial Is Everything You Need. [online] DataCamp Community. Available at: <https://www.datacamp.com/community/tutorials/r-data-import-tutorial> [Accessed 4 Feb. 2022].

Neonscience.org. (2021). Filter, Piping, and GREPL Using R DPLYR - An Intro | NSF NEON | Open Data to Understand our Ecosystems. [online] Available at: <https://www.neonscience.org/resources/learning-hub/tutorials/grepl-filter-piping-dplyr-r> [Accessed 5 Feb. 2022].

Dummies.com. (2022). dummies - Learning Made Easy. [online] Available at: <https://www.dummies.com/article/technology/programming-web-design/r/how-to-convert-tables-to-a-data-frame-in-r-141550> [Accessed 5 Feb. 2022].

Ebner, J. (2018). How to rename columns in R. [online] Sharp Sight. Available at: <https://www.sharpsightlabs.com/blog/rename-columns-in-r/> [Accessed 5 Feb. 2022].

Heinzen, E. (2017). The comparedf function. [online] R-project.org. Available at: <https://cran.r-project.org/web/packages/arsenal/vignettes/comparedf.html> [Accessed 5 Feb. 2022].

Statistics Globe. (2021). Calculate Percentage in R (Example) | Convert Vector to Probability Table. [online] Available at: <https://statisticsglobe.com/calculate-percentage-in-r> [Accessed 5 Feb. 2022].

in (2010). How to increase size of label fonts in barplot. [online] Cross Validated. Available at: <https://stats.stackexchange.com/questions/3853/how-to-increase-size-of-label-fonts-in-barplot> [Accessed 5 Feb. 2022].

DataMentor. (2017). Data Mentor. [online] Available at: <https://www.datamentor.io/r-programming/bar-plot/> [Accessed 5 Feb. 2022].

Holtz, Y. (2018). Advanced R barplot customization. [online] R-graph-gallery.com. Available at: <https://www.r-graph-gallery.com/210-custom-barplot-layout.html> [Accessed 5 Feb. 2022].

Datanovia. (2018). Reorder Data Frame Rows in R - Datanovia. [online] Available at: <https://www.datanovia.com/en/lessons/reorder-data-frame-rows-in-r/> [Accessed 5 Feb. 2022].

Tutorialspoint.com. (2020). How to create a vector with repeated values in R? [online] Available at: <https://www.tutorialspoint.com/how-to-create-a-vector-with-repeated-values-in-r> [Accessed 5 Feb. 2022].

R-bloggers. (2012). Controlling Axes of R Plots | R-bloggers. [online] Available at: <https://www.r-bloggers.com/2012/12/controlling-axes-of-r-plots/> [Accessed 6 Feb. 2022].

GeeksforGeeks. (2021). Visualizing Missing Data with Barplot in R - GeeksforGeeks. [online] Available at: <https://www.geeksforgeeks.org/visualizing-missing-data-with-barplot-in-r/> [Accessed 6 Feb. 2022].

R CODER. (2020). BAR PLOTS in R [STACKED and GROUPED bar charts]. [online] Available at: <https://r-coder.com/barplot-r/> [Accessed 6 Feb. 2022].

Amazonaws.com. (2022). Pareto Chart With Base R Plotting System. [online] Available at: https://rstudio-pubs-static.s3.amazonaws.com/72023_670962b57f444c04999fd1a0a393e113.html [Accessed 6 Feb. 2022].

Statmethods.net. (2017). Quick-R: Axes and Text. [online] Available at: <https://www.statmethods.net/advgraphs/axes.html> [Accessed 6 Feb. 2022].

momova97 (2022). ALY6000_movahedi/Movahedi_M3_Project3.R at main · momova97/ALY6000_movahedi. [online] GitHub. Available at: https://github.com/momova97/ALY6000_movahedi/blob/main/Movahedi_M3_Project3.R [Accessed 6 Feb. 2022].

Appendix

```
#printing my name
print("mohammad Hossein Movahedi")
#importing and loading required libraries
install.packages("FSA")
install.packages("FSAdata")
install.packages("magrittr")
install.packages("dplyr")
install.packages("tidyr ")
install.packages("plyr")
install.packages("tidyverse")
install.packages()
#loading installed libraries
library(FSA)
library(FSAdata)
library(magrittr)
library(dplyr)
library(tidyr)
library(plyr)
library(tidyverse)
library(arsenal)
#Import the inchBio.csv and name the table bio
bio <- read_csv("/cloud/project/inchBio.csv")
#displaying head and tail and structure of bio
headtail(bio,n = 5)
str(bio)
#creating an object that counts and lists all the species records
ubio <- unique(bio$species[!is.na(bio$species)])
```

```

list <- vector()
counts <- vector()
for (i in ubio){
  list <- c(list,i)
  counts <- c(counts, length(which(bio$species==i)))
}
countlist = data.frame(lists = list,count = counts)
#displaying names of species
disp <- headtail(countlist,n = 4)
print(disp$lists)

#creating tmp
tmp <- countlist
#Create a subset of just the species variable and display the first five records
tmp2 <- tmp$lists
head(tmp2,n=5)
#Create a table, <w>, of the species variable. Display the class of w
w <- table(tmp$lists)
class(w)
# Convert <w> to a data frame named <t> and display the results
t <- as.data.frame(w)
print(t)
#Extract and display the frequency values from the <t> data frame
freq <- t$Freq
print(freq )
# creating cSpec
cSpec <- table(bio$species)
tcSpec <- as.data.frame(cSpec)
tcountlist <- as.data.frame(countlist)
tcountlist <- select(tcountlist,Var1 = lists,Freq = count)
comparedf (tcountlist,tcSpec)
# Create a table named <cSpecPct>
cSpecPct <- table(bio$species)
cSpecPct <- cSpecPct/length(bio$species)
class(cSpecPct)
#Convert the table, <cSpecPct>, to a data frame named <u>
u <- as.data.frame(cSpecPct)
class(u)
#Create a barplot of <cSpec>
par(mar=c(4,6,3,8))
barplot(cSpec,

```



```

    main = "Fish Count",
    xlab = "COUNTS",
    ylab = "",
    cex.axis=0.6,
    cex.names=0.7,
    col = "Light Green",
    horiz = TRUE,
    las = 1)
#Create a barplot of <cSpecPct>
par(mar=c(4,6,8,8))
barplot(d$counts,
    main = "Fish Relative Frequency",
    xlab = "COUNTS",
    ylab = "",
    ylim = range(0,4),
    cex.axis=0.6,
    cex.names=0.7,
    col = "Light Green",
    col.axis = "Species Pareto",
    horiz = TRUE,
    las = 1)
#Rearrange the <u> cSpec Pct data frame in descending order as <d>
d <- u %>% arrange(desc(Freq))
#remaing d cols
d <- select(d,Species = Var1 , RelFreq = Freq)

#adding new vars to d
d <- data.frame(d,cumfreq =
rep(c(NA),times=8),counts=rep(c(NA),times=8),cumcounts = rep(c(NA),times=8))
#Create a parameter variable <def_par> to store parameter variables
def_par = c()
#creating <pc> bar chart
pc <- barplot(d$RelFreq,
    width = 1, space = 0.15,axes = F,border = NA,
    main = "Species Pareto",
    xlab = "Cummulative Counts",
    ylab = "",
    ylim = c(0,3.05),
    names.arg = d$Species,
    cex.axis=0.6,
    cex.names=0.7,

```

```

    col = "Light Green",
    col.axis = "Light Blue",
    horiz = TRUE,
    las = 2,
  )
# Add a cumulative counts line to the <pc> plot
par(def_par)
par(mar=c(5,3,3,4))
barplot(d$RelFreq,
        width = 1, space = 0.15, axes = F, border = NA,
        main = "Species Pareto",
        sub = "Movahedi",
        xlab = "Cummulative Counts",
        ylab = "",
        ylim = c(0,3.05),
        names.arg = d$Species,
        cex.axis=0.6,
        cex.names=0.7,
        col = "Light Green",
        col.axis = "Light Blue",
        horiz = FALSE,
        las = 2,
      )
d <- arrange(d, desc(RelFreq)) %>%
  mutate(
    cumsum = cumsum(RelFreq),
    freq = round(RelFreq / sum(RelFreq), 3),
    cum_freq = cumsum(freq)
  )
lines(pc, d$cumsum, type = "b", cex = 0.7, pch = 20, col="cyan4")

#placing a box around the chart
box(col = "grey62")

# Add a left side axis
axis(2, at=pc, labels=pc, col.axis="grey62", cex.axis=0.8, tck=-.01, las=0)
#Add axis details on right side of box with the specifications

axis(4, at=pc, labels=d$cum_freq,
      col.axis="cyan", las=2, cex.axis=0.8, tck=-.01)

```