Name – Patel Dhruvil Sunilbhai

NUID - 002955193

Date – 04/02/2022

Title – M3_Project

Instructor – Mohammad Shafiqul Islam (Shafiqul)

Module 3 – Project

Q1) Print your name at the top of the script and load these libraries: FSA, FSAdata, magrittr, dplyr, tidyr plyr and tidyverse

```
> print("Dhruvil Patel")
[1] "Dhruvil Patel"
```

```
> library(FSA)
## FSA v0.9.1. See citation('FSA') if used in publication.
## Run fishR() for related website and fishR('IFAR') for related book.
> library(FSAdata)
## FSAdata v0.3.8. See ?FSAdata to find data for specific fisheries analyses.
> library(magrittr)
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

> library(tidyr plyr)
Error: unexpected symbol in "library(tidyr plyr"
> library(tidyr)

Attaching package: 'tidyr'

The following object is masked from 'package:magrittr':

    extract

> library(plyr)
------------------------------------------------------------------------------
You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)
------------------------------------------------------------------------------

Attaching package: 'plyr'
```

```
> library(tidyverse)
-- Attaching packages ---------------------------------
v ggplot2 3.3.5     v purrr   0.3.4
v tibble  3.1.6     v stringr 1.4.0
v readr   2.1.1     v forcats 0.5.1
-- Conflicts ------------------------------------------
x plyr::arrange()    masks dplyr::arrange()
x purrr::compact()   masks plyr::compact()
x plyr::count()      masks dplyr::count()
x tidyr::extract()   masks magrittr::extract()
x plyr::failwith()   masks dplyr::failwith()
x dplyr::filter()    masks stats::filter()
x plyr::id()         masks dplyr::id()
x dplyr::lag()       masks stats::lag()
x plyr::mutate()     masks dplyr::mutate()
x plyr::rename()     masks dplyr::rename()
x purrr::set_names() masks magrittr::set_names()
x plyr::summarise()  masks dplyr::summarise()
x plyr::summarize()  masks dplyr::summarize()
>
```

Q2) Import the inchBio.csv and name the table

```
> bio <- read.csv(file.choose(), header=T)
> bio
    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
13     13     74         Bluegill  35    0.5          FALSE
14     13     75         Bluegill  32    1.0          FALSE
```

Q3) Display the head, tail and structure of bio

```
> head(bio)
  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
> tail(bio)
    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
> 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 ...
>
```

Q4) Create an object, , that counts and lists all the species records

```
> counts <- data.frame(bio)
> count(counts,"species")
           species 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
```

Q5) Display just the 8 levels (names) of the species

```
count(counts,"spe
         species
   Black Crappie
        Bluegill
Bluntnose Minnow
     Iowa Darter
 Largemouth Bass
     Pumpkinseed
  Tadpole Madtom
    Yellow Perch
```

Q6)

Create a <tmp> object that displays the different species and the number of record of each species in the dataset. Include this information in your report.

```
tmp<-table(bio$species)
data.frame(tmp)
                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
```

Q7) Create a subset, <tmp2>, of just the species variable and display the first five records

```
tmp2 <- subset(bio, select = species)
head(tmp2,5)
  species
1 Bluegill
2 Bluegill
3 Bluegill
4 Bluegill
5 Bluegill
class(tmp2)
[1] "data.frame"
```

Q8) Create a table, <w>, of the species variable. Display the class of w

```
> w <- table(bio$species)
> w

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

Q9) Convert <w> to a data frame named <t> and display the results

```
> t <- as.data.frame(w)
> t
              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
> class(t)
[1] "data.frame"
>
```

Q10) Extract and display the frequency values from the <t> data frame

```
> t$Freq
[1]  36 220 103  32 228  13   6  38
>
```

Q11) Create a table named <cSpec> from the bio species attribute (variable) and confirm that you created a table which displays the number of species in the dataset <bio>

```
> cSpec <- table(bio$species)
> cSpec

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

Q12) Create a table named <cSpecPct> that displays the species and percentage of records for each species. Confirm you created a table class.

```
> cSpecPct <- prop.table(table(bio$species))
> cSpecPct

   Black Crappie         Bluegill Bluntnose Minnow      Iowa Darter
      0.05325444       0.32544379       0.15236686       0.04733728
 Largemouth Bass      Pumpkinseed   Tadpole Madtom     Yellow Perch
      0.33727811       0.01923077       0.00887574       0.05621302
> class(cSpecPct)
[1] "table"
```

Q13) Convert the table, <cSpecPct>, to a data frame named <u> and confirm that <u> is a data frame

```
 u <- as.data.frame(cSpecPct)
 class(u)
1] "data.frame"
 u
             Var1       Freq
    Black Crappie 0.05325444
         Bluegill 0.32544379
 Bluntnose Minnow 0.15236686
      Iowa Darter 0.04733728
  Largemouth Bass 0.33727811
      Pumpkinseed 0.01923077
   Tadpole Madtom 0.00887574
     Yellow Perch 0.05621302
```

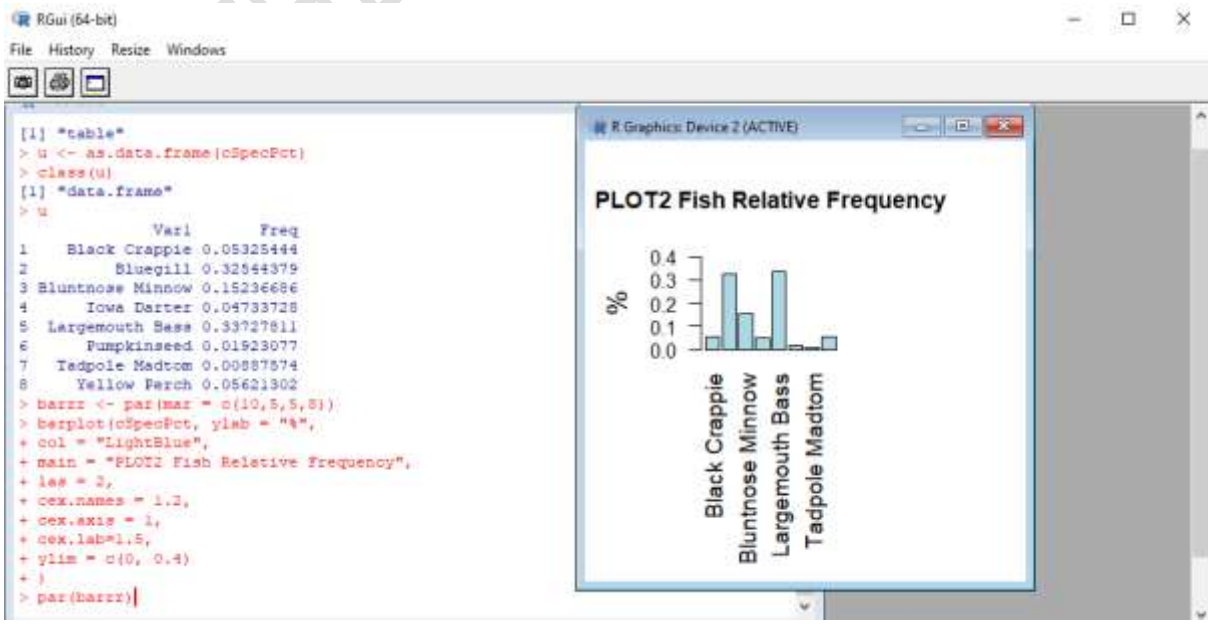
Q14) and Q15)

Create a barplot of <cSpec> with the following: titled Fish Count with the following specifications:

- Title: Fish Count
- Y axis is labeled “COUNTS”
- Color the bars Light Green
- Rotate Y axis to be horizontal
- Set the X axis font magnification to 60% of nominal

Create a barplot of <cSpecPct>, with the following specifications:

- Y axis limits of 0 to 4
- Y axis label color of Light Blue
- Title of “Fish Relative Frequency”



Q16) Rearrange the <u> cSpec Pct data frame in descending order of relative frequency. Save the rearranged data frame as the object <d>

```
> u
              Var1       Freq
1    Black Crappie 0.05325444
2         Bluegill 0.32544379
3 Bluntnose Minnow 0.15236686
4      Iowa Darter 0.04733728
5  Largemouth Bass 0.33727811
6      Pumpkinseed 0.01923077
7   Tadpole Madtom 0.00887574
8     Yellow Perch 0.05621302
> class(u)
[1] "data.frame"
> d <- u%>%
+
+ aa
Error in aa(.) : could not find function "aa"
> d <- u%>%
+ arrange(desc(Freq))
> d
              Var1       Freq
1  Largemouth Bass 0.33727811
2         Bluegill 0.32544379
3 Bluntnose Minnow 0.15236686
4     Yellow Perch 0.05621302
5    Black Crappie 0.05325444
6      Iowa Darter 0.04733728
7      Pumpkinseed 0.01923077
8   Tadpole Madtom 0.00887574
> class(d)
[1] "data.frame"
>
```

Q17) Rename the <d> columns Var 1 to Species, and Freq to RelFreq

```
names(d) <- c('Species','RelativeFreq')
d
         Species RelativeFreq
 Largemouth Bass   0.33727811
        Bluegill   0.32544379
Bluntnose Minnow   0.15236686
    Yellow Perch   0.05621302
   Black Crappie   0.05325444
     Iowa Darter   0.04733728
     Pumpkinseed   0.01923077
  Tadpole Madtom   0.00887574
```

Q18) Add new variables to <d> and call them cumfreq, counts, and cumcounts

```
d <- transform(d, cumfreq = cumsum(RelativeFreq))
d
          Species RelativeFreq   cumfreq
 Largemouth Bass   0.33727811 0.3372781
        Bluegill   0.32544379 0.6627219
Bluntnose Minnow   0.15236686 0.8150888
    Yellow Perch   0.05621302 0.8713018
   Black Crappie   0.05325444 0.9245562
     Iowa Darter   0.04733728 0.9718935
     Pumpkinseed   0.01923077 0.9911243
  Tadpole Madtom   0.00887574 1.0000000
```

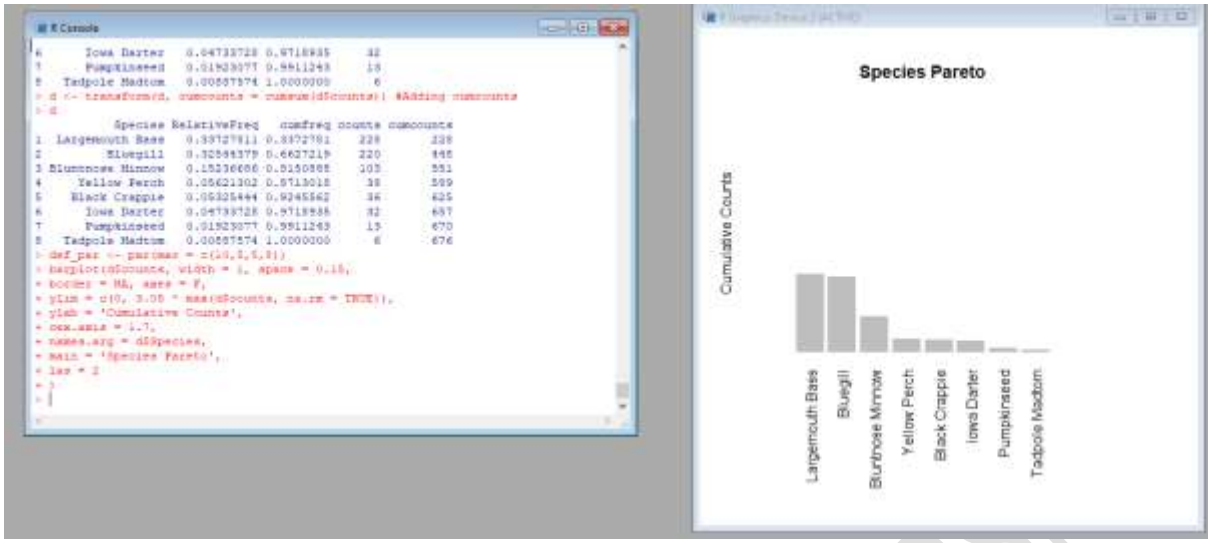
```
> d <- transform(d, counts = (RelativeFreq * nrow(temp2)))
> d
           Species RelativeFreq   cumfreq counts
1  Largemouth Bass   0.33727811 0.3372781    228
2         Bluegill   0.32544379 0.6627219    220
3 Bluntnose Minnow   0.15236686 0.8150888    103
4     Yellow Perch   0.05621302 0.8713018     38
5    Black Crappie   0.05325444 0.9245562     36
6      Iowa Darter   0.04733728 0.9718935     32
7      Pumpkinseed   0.01923077 0.9911243     13
8   Tadpole Madtom   0.00887574 1.0000000      6
>
```

```
d <- transform(d, cumcounts = cumsum(d$counts)) #Adding cumcounts
d
          Species RelativeFreq   cumfreq counts cumcounts
 Largemouth Bass   0.33727811 0.3372781    228       228
        Bluegill   0.32544379 0.6627219    220       448
Bluntnose Minnow   0.15236686 0.8150888    103       551
    Yellow Perch   0.05621302 0.8713018     38       589
   Black Crappie   0.05325444 0.9245562     36       625
     Iowa Darter   0.04733728 0.9718935     32       657
     Pumpkinseed   0.01923077 0.9911243     13       670
  Tadpole Madtom   0.00887574 1.0000000      6       676
```

Q19) Q20)

19. Create a parameter variable <def_par> to store parameter variables
20. Create a barplot, <pc>, with the following specifications:

- d$counts of width 1, spacing of .15
- no boarder
- Axes: F
- Yaxis limit 0,3.05*max
- d$counts na.rm is true
- y label is Cummulative Counts
- scale x axis to 70%
- names.arg: d$Species
- Title of the barplot is “Species Pareto”
- las: 2)

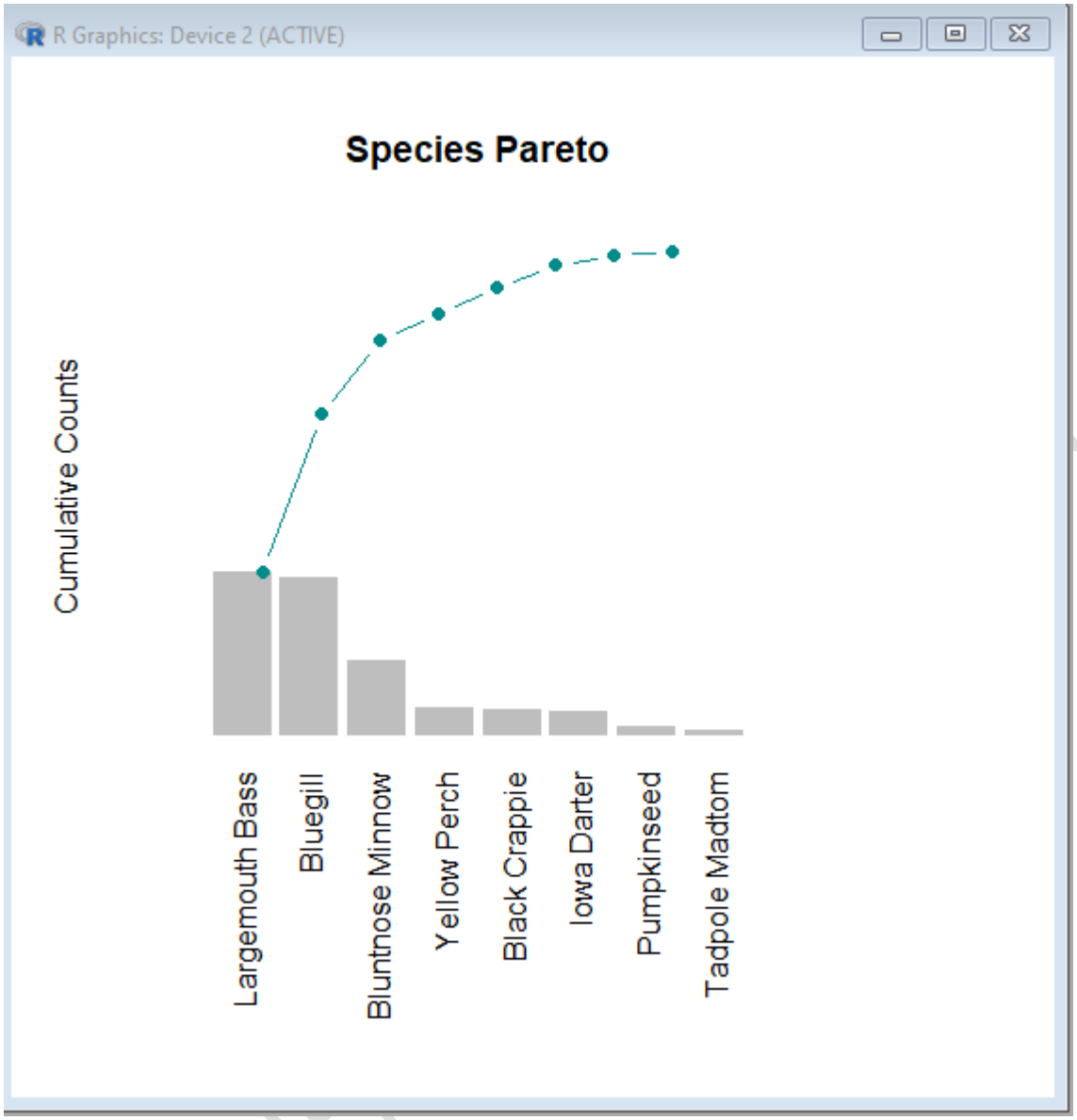


Q21)

Add a cumulative counts line to the <pc> plot with the following:

- Spec line type is b
- Scale plotting text at 70%
- Data values are solid circles with color cyan4

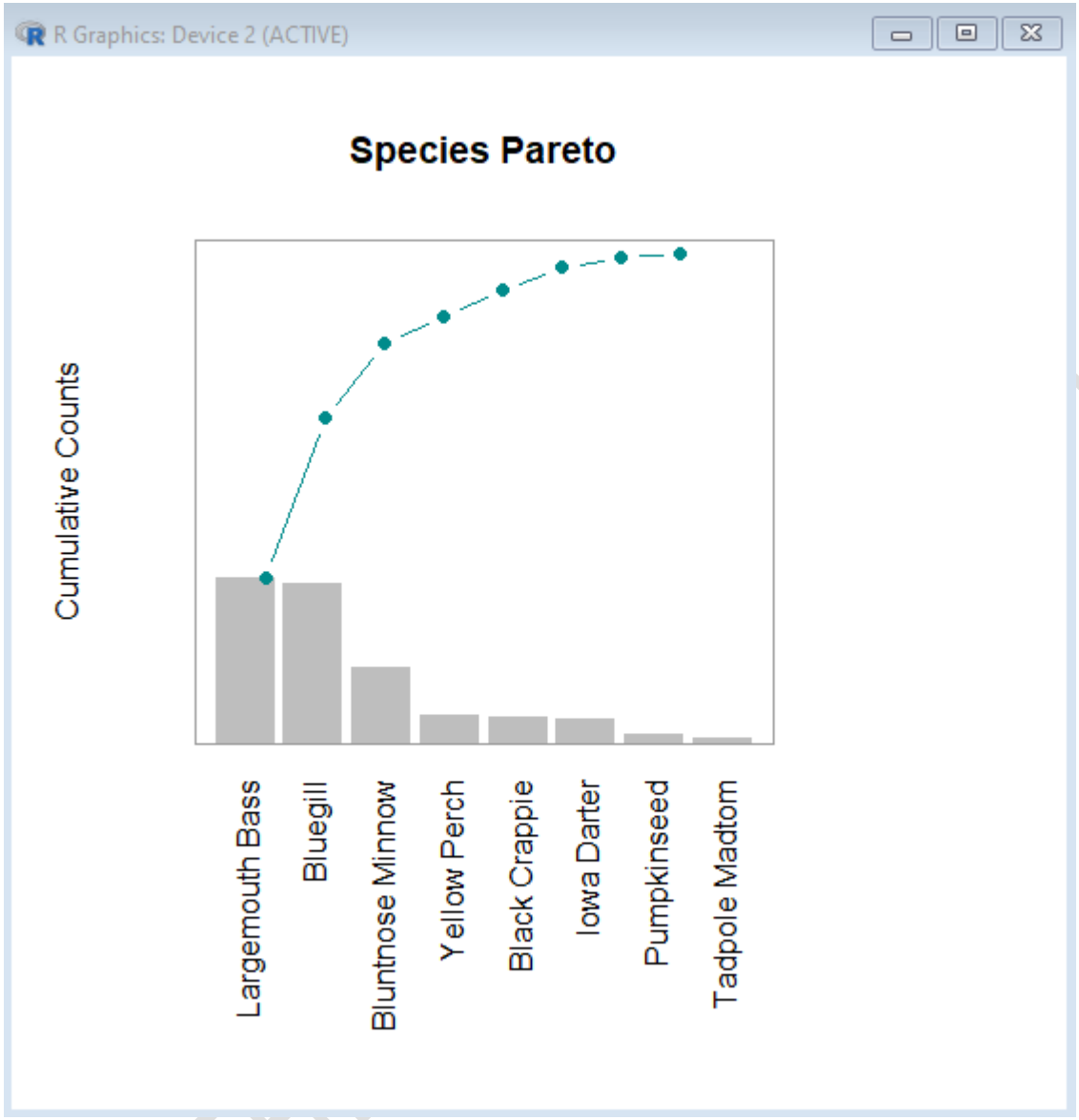
```
def_par <- par(mar = c(10,5,5,8))
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = F,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
names.arg = d$Species,
main = 'Species Pareto',
las = 2
)
lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4')
```



Q22)

Place a grey box around the pareto plot (hint: https://www.statmethods.net/advgraphs/parameters.html)

```
def_par <- par(mar = c(10,5,5,8))
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = F,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
names.arg = d$Species,
main = 'Species Pareto',
las = 2
)
lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4') #Cumulative counts li$
box(col = 'grey62')
```

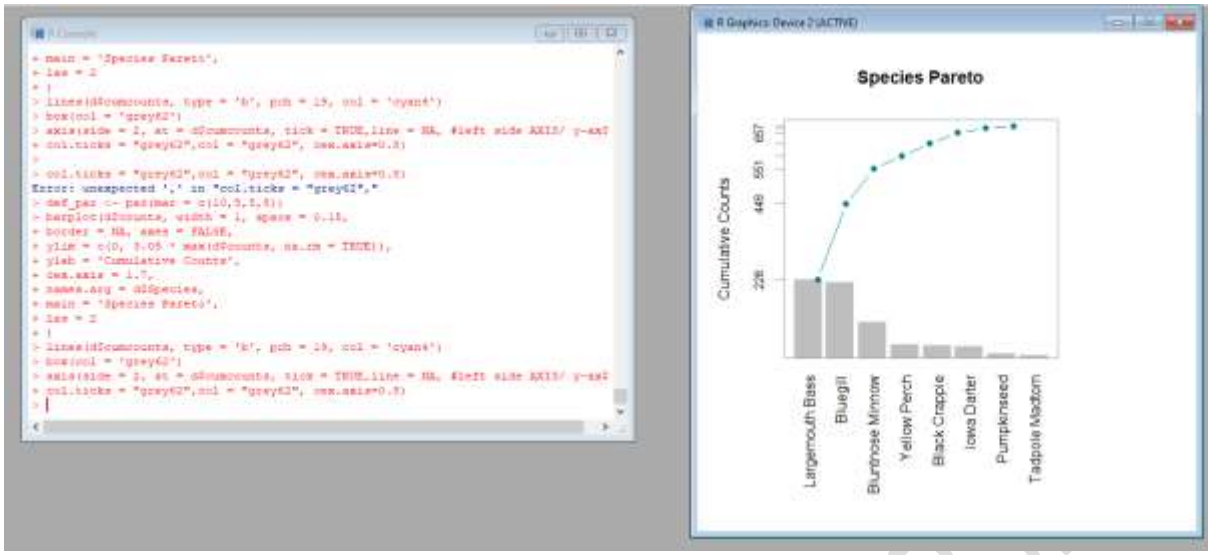


Q23)

Add a left side axis with the following specifications

- Horizontal values at tick marks at cumcounts on side 2
- Tickmark color of grey62
- Color of axis is grey62
- Axis scaled to 80% of normal

(hint: https://www.statmethods.net/advgraphs/axes.html)



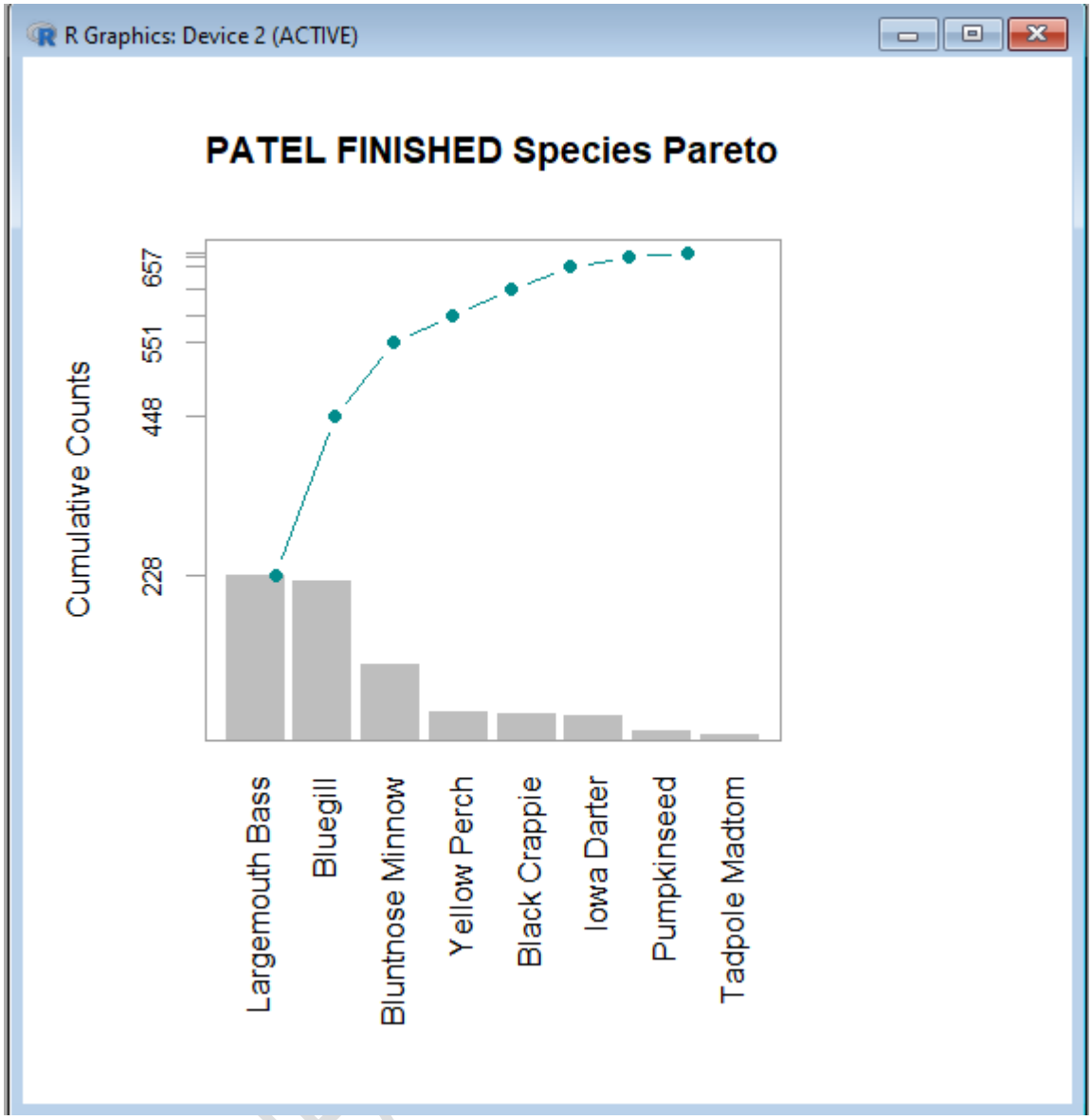
Q24) Q25)

Add axis details on right side of box with the specifications:

- Spec: Side 4
- Tickmarks at cumcounts with labels from 0 to cumfreq with %,
- Axis color of cyan5 and label color of cyan4
- Axis font scaled to 80% of nominal

25. Display the finished Species Pareto Plot (without the star watermarks). Have your last name on the plot

```
> def_par <- par(mar = c(10,5,5,8))
> barplot(d$counts, width = 1, space = 0.15,
- border = NA, axes = FALSE,
- ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
- ylab = 'Cumulative Counts',
- cex.axis = 1.7,
- names.arg = d$Species,
- main = 'PATEL FINISHED Species Pareto',
- las = 2
- )
> lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4')
> box(col = 'grey62')
> axis(side = 2, at = d$cumcounts, tick = TRUE,line = NA, #left side AXIS/ y-ax$
- col.ticks = "grey62",col = "grey62", cex.axis=0.8)
>
```



Q26) Github Link :

https://github.com/Dhruvilp7120/ALY6000-Patel

Summary:

dplyr is a package which provides a set of tools for efficiently manipulating datasets in R. Tidyverse packages are intended to make statisticians and data scientists more productive by guiding them through workflows that facilitate communication, and result in reproducible work products.

count() lets you quickly count the unique values of one or more variables

table() function in R Language is used to create a categorical representation of data with variable name and the frequency in the form of a table.

The class function in R helps us to understand the type of object, for example the output of class for a data frame is integer and the typeof of the same object is list because data frames are stored as list in the memory but they are represented as a data frame.

A data frame is the most common way of storing data in R and, generally, is the data structure most often used for data analyses.

barplot is used to display the relationship between a numeric and a categorical variable.

barplot(H,xlab,ylab,main, names.arg,col)

- H is a vector or matrix containing numeric values used in bar chart.
- xlab is the label for x axis.
- ylab is the label for y axis.
- main is the title of the bar chart.
- names.arg is a vector of names appearing under each bar.
- col is used to give colors to the bars in the graph.

A relative frequency table tells you how often certain values in a dataset occur relative to the total number of values in the dataset.

Cumsum(): The cumulative frequency can be computed by the summation of each frequency value from a frequency distribution table to include the sum of its predecessors.

cumcount: Cumulative count of strings. Return an integer vector counting the number of occurrences of each string up to that position in the vector.
A Pareto graph is a type of graph that displays the frequencies of the different categories with the cumulated frequencies of the categories.

Syntax:
pareto.chart(x, ylab = "Frequency", ylab2 = "Cumulative Percentage", xlab, cumperc = seq(0, 100, by = 25), ylim, main, col = heat.colors(length(x)))
Parameters:
x: a vector of values. names(x) are used for labelling the bars.
ylab: a string specifying the label for the y-axis.
ylab2: a string specifying the label for the second y-axis on the right side.
xlab: a string specifying the label for the x-axis.
cumperc: a vector of percentage values to be used as tickmarks for the second y-axis on the right side.
ylim: a numeric vector specifying the limits for the y-axis.
main: a string specifying the main title to appear on the plot.
col: a value for the color, a vector of colors, or a palette for the bars. See the help for colors and palette.

Bibliography:

- http://127.0.0.1:18828/library/vcd/html/00Index.html
- https://cran.r-project.org/mirrors.html
- http://127.0.0.1:18828/doc/html/packages.html
- https://rdrr.io/cran/vcd/
- https://r-forge.r-project.org/R/?group_id=351
- https://www.geeksforgeeks.org/r-bar-charts/
- https://www.tutorialspoint.com/how-to-count-the-number-of-values-that-satisfy-a-condition-in-an-r-vector
- https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/data.frame
- Book - R in action book

Appendix:

LINK - https://github.com/Dhruvilp7120/ALY6000-Patel

Code

Q1)

```
print("Dhruvil Patel")
```

Q2)

```
bio <- read.csv(file.choose(), header=T)
```

Q3)

```
head(bio)
tail(bio)
str(bio)
```

Q4)

```
counts <- data.frame(bio)
count(counts,"species")
```

Q5)

```
count(counts,"species")
```

Q6)

```
 tmp<-table(bio$species)
> data.frame(tmp)
```

Q7)

```
w <- table(bio$species)
w
class(w)
```

Q8)

```
t <- as.data.frame(w)
class(t)
t
```

Q9)

```
t <- as.data.frame(w)
class(t)
t
```

Q10)

```
t$Freq
```

Q11)

```
cSpec <- table(bio$species)
cSpec
class(cSpec)
```

Q12)

```
cSpecPct <- prop.table(table(bio$species))
cSpecPct
class(cSpecPct)

cSpecPct <- prop.table(table(bio$species))
cSpecPct
class(cSpecPct)
```

Q13)

```
u <- as.data.frame(cSpecPct)
```

```
class(u)
u
Q14) Q15)
barrr <- par(mar = c(10,5,5,8))
barplot(cSpecPct, ylab = "%",
col = "LightBlue",
main = "PLOT2 Fish Relative Frequency",
las = 2,
cex.names = 1.2,
cex.axis = 1,
cex.lab=1.5,
ylim = c(0, 0.4)
)
par(barrr)
Q16)
u
class(u)
d <- u %>%
arrange(desc(Freq))
d
class(d)
Q17)
names(d) <- c('Species','RelativeFreq')
> d
Q18)
d <- transform(d, cumfreq = cumsum(RelativeFreq)) #Adding cumfreq
d
d <- transform(d, counts = (RelativeFreq * nrow(temp2))) #Adding counts
d
d <- transform(d, cumcounts = cumsum(d$counts)) #Adding cumcounts
d
```

Q19)

```
def_par <- par(mar = c(10,5,5,8))
```

Q20)

```
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = F,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
names.arg = d$Species,
main = 'Species Pareto',
las = 2
)
```

Q21)

```
def_par <- par(mar = c(10,5,5,8))
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = F,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
names.arg = d$Species,
main = 'Species Pareto',
las = 2
)
lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4')
```

Q22)

```
def_par <- par(mar = c(10,5,5,8))
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = F,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
```

```
names.arg = d$Species,
main = 'Species Pareto',
las = 2
)
lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4') #Cumulative counts line
box(col = 'grey62')
```

Q23)

```
def_par <- par(mar = c(10,5,5,8))
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = FALSE,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
names.arg = d$Species,
main = 'Species Pareto',
las = 2
)
lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4')
box(col = 'grey62')
axis(side = 2, at = d$cumcounts, tick = TRUE,line = NA, #left side AXIS/ y-axis
col.ticks = "grey62",col = "grey62", cex.axis=0.8)
```

Q24)

```
def_par <- par(mar = c(10,5,5,8))
barplot(d$counts, width = 1, space = 0.15,
border = NA, axes = FALSE,
ylim = c(0, 3.05 * max(d$counts, na.rm = TRUE)),
ylab = 'Cumulative Counts',
cex.axis = 1.7,
names.arg = d$Species,
```

```
main = 'PATEL FINISHED Species Pareto',
las = 2
)
lines(d$cumcounts, type = 'b', pch = 19, col = 'cyan4')
box(col = 'grey62')
axis(side = 2, at = d$cumcounts, tick = TRUE,line = NA, #left side AXIS/ y-axis
col.ticks = "grey62",col = "grey62", cex.axis=0.8)
```