Executive Summary Report 3

Calculate data & Interpret data in graphs with R

ALY6000: Introduction to Analytics

Prepared by: Heejae Roh
Presented to: Professor Behzad Ahmadi

Date: Oct 13th, 2022

[Introduction]

In R, researcher must deal with data in raw format and need to know several methods that how to handle raw format data. Sometimes, process must be done before analyzing the characteristics and structure of the data. After raw format data is manipulated or calculated, researcher can explain the data. In this module I learned frequency data that is one of the most popular sorts of data.

[Key findings]

1. Name & Import libraries

```
print("Heejae Roh")
install.packages("easypackages")
install.packages(c("FSA","FSAdata"
,"magrittr","dplyr","tidyr","plyr"
,"tidyverse"))
library(easypackages)
libraries("FSA", "FSAdata",
"magrittr","dplyr","tidyr","plyr"
,"tidyverse")
```

1

2-3. Load <bio> & Display head, tail, str

```
> setwd("C:\\Users\\14083\\Desktop\\exacutive summary\\Project 3")
> bio <- read.csv("inchBio.csv", header=T,
+                 stringsAsFactors = T)
> headtail(bio,1)
    netID fishID       species  tl     w  tag scale
1      12     16      Bluegill  61   2.9      FALSE
676   129    879 Black Crappie 302 397.0 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: Factor w/ 8 levels "Black Crappie",..: 2 2 2 2 2 2 2 2
 $ 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    : Factor w/ 193 levels "","1014","1015",..: 1
 $ scale  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ..
```

2-3

Add 'stringsAsFactor=T' to make bio variables that have levels. No need to use factor().

4-5. Create <counts> that count species, Display just the 8 levels of the species (left)

```
> counts <- count(bio, "species")
> counts
           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
> levels(bio$species)
[1] "Black Crappie"    "Bluegill"         "Bluntnose Minnow"
[4] "Iowa Darter"      "Largemouth Bass"  "Pumpkin
[7] "Tadpole Madtom"   "Yellow Perch"
```

4-5

6-7. <tmp> displays species and number <tmp2> of just species variable and display 5

```
> tmp <- count(bio, "species")
> tmp
           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
> tmp2 <- select(bio, "species")
> head(tmp2, 5)
   species
1 Bluegill
2 Bluegill
3 Bluegill
4 Bluegill
5 Bluegill
```

6-7

Using count{plyr}, select{dplyr} is helpful in calculating raw format data to analyze frequency. According to '?count': if sort=TRUE, will show the largest groups at the top.

8-10. <w> display class/ convert to data frame <t>/ display frequency value (left)

```
> w <- table(bio$species)
> class(w)
[1] "table"
> 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
```

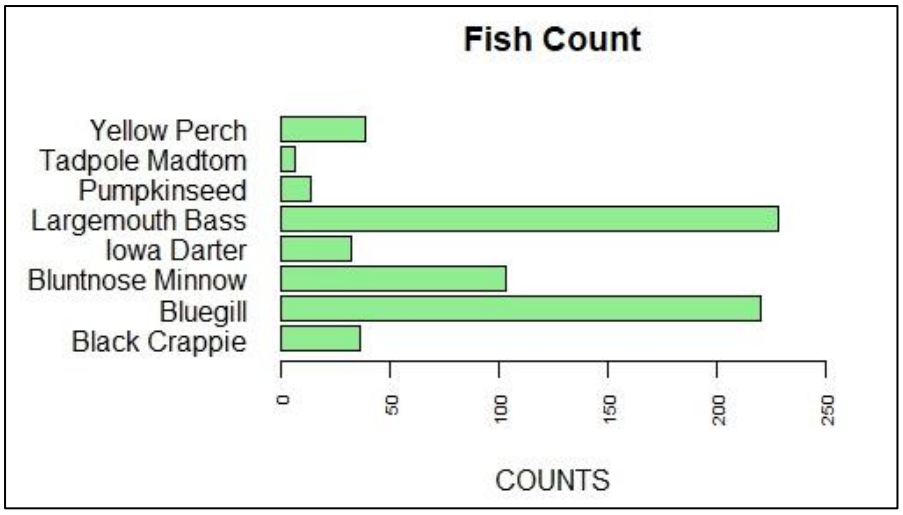
8-10

```
> cSpec <- w
> cSpec

   Black Crappie         Bluegill Bluntnose Minnow      Iowa Darter
              36              220              103               32
 Largemouth Bass       Pumpkinseed   Tadpole Madtom     Yellow Perch
             228               13                6               38
> cSpecPct <- table(bio$species) / length(bio$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"
> u <- as.data.frame(cSpecPct)
> class(u)
[1] "data.frame"
```

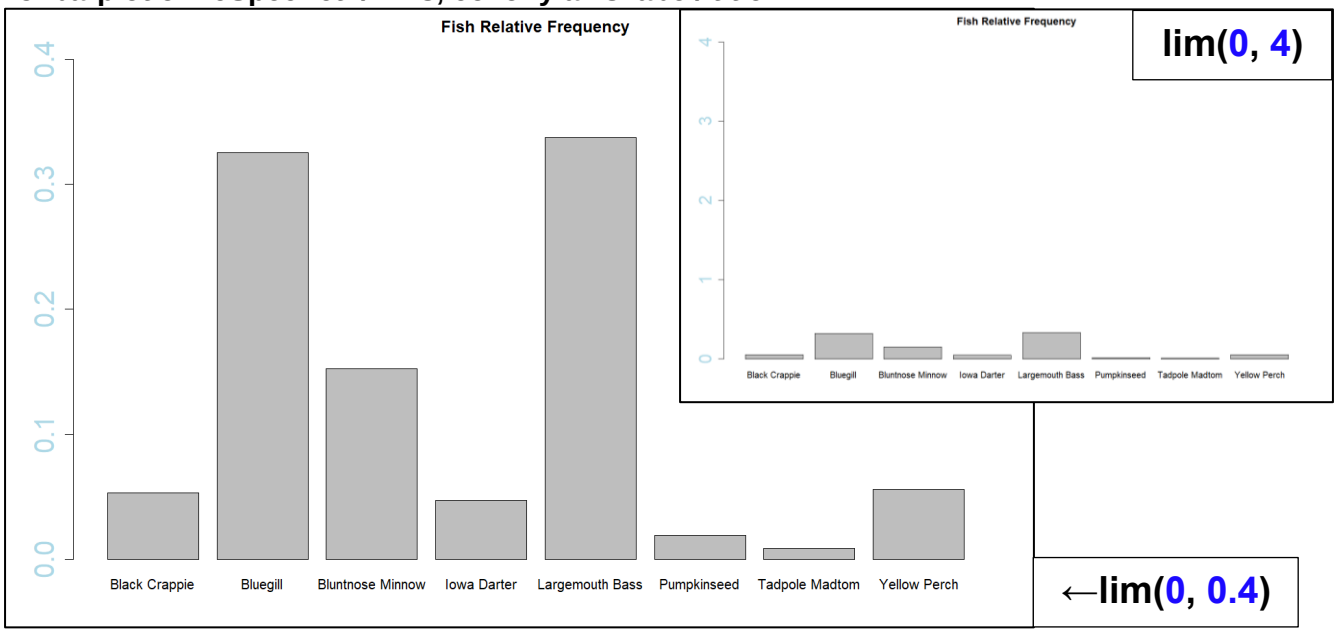
11-13

11-13. <cSpec>: number/ <u>: dataframe of <cSpecPct>: pct(%) of records

14. barplot of <cSpec>/ title, ylab, rotate y axis, set x axis font 60%



15. barplot of <cSpecPct>/ lims, col of y axis label/ title



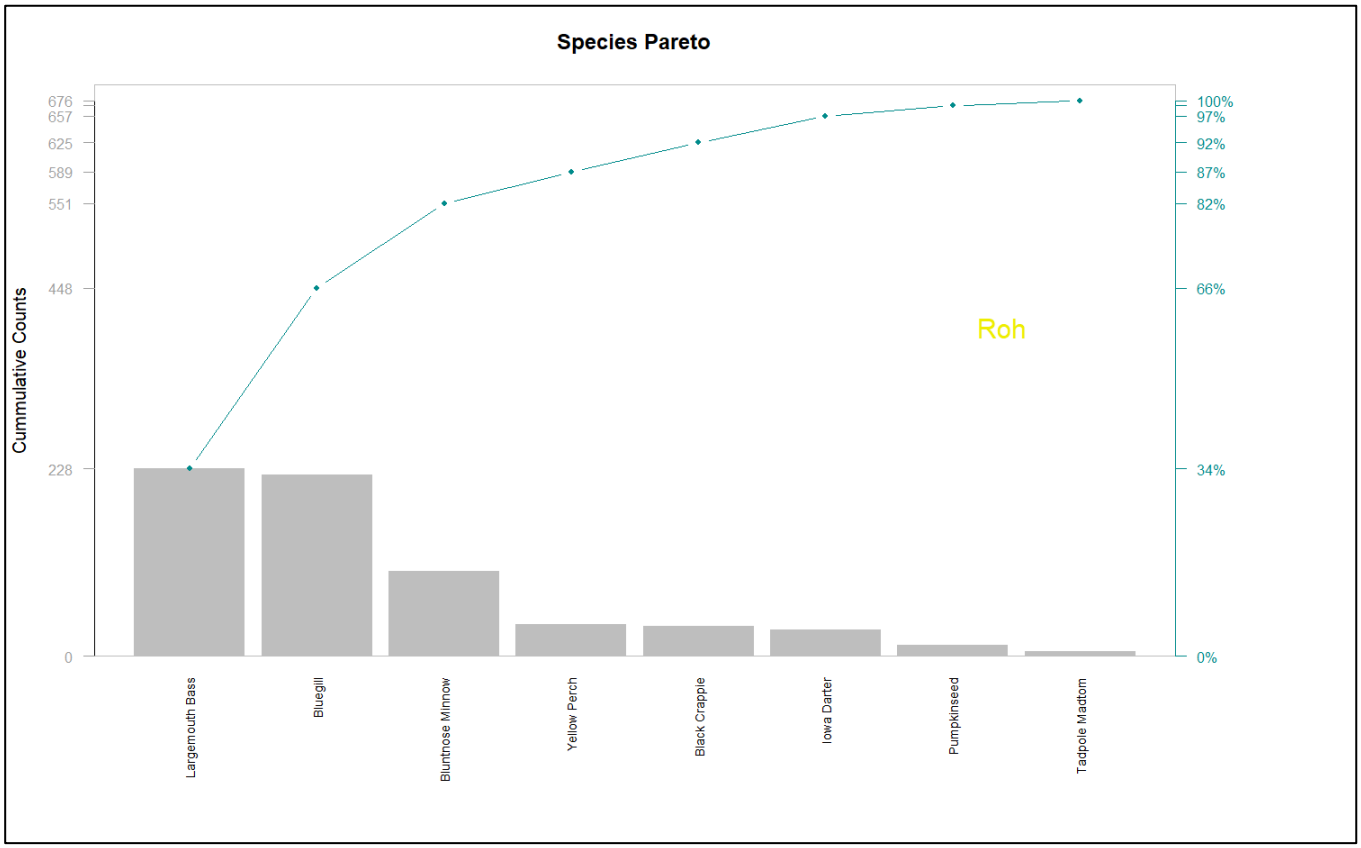
16-18. <u> descending order save as <d>/ rename col Var1 to Species Freq to RelFreq add variables: cumfreq, counts, cumcounts

```
> d <- u[order(u$Freq, decreasing = TRUE),]
> library(reshape)
> d <- rename(d, c(Var1 = "Species", Freq = "RelFreq"))
> d$cumfreq <- cumsum(d$RelFreq)
> d$counts <- d$RelFreq*676
> d$cumcounts <- cumsum(d$counts)
> d
```

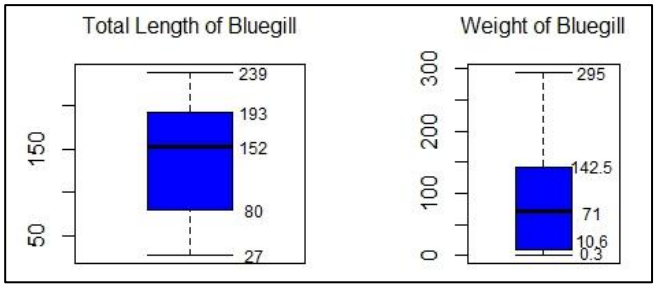
16-18

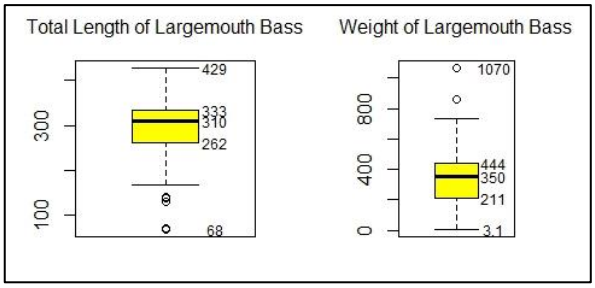
	Species	RelFreq	cumfreq	counts	cumcounts
5	Largemouth Bass	0.33727811	0.3372781	228	228
2	Bluegill	0.32544379	0.6627219	220	448
3	Bluntnose Minnow	0.15236686	0.8150888	103	551
8	Yellow Perch	0.05621302	0.8713018	38	589
1	Black Crappie	0.05325444	0.9245562	36	625
4	Iowa Darter	0.04733728	0.9718935	32	657
6	Pumpkinseed	0.01923077	0.9911243	13	670
7	Tadpole Madtom	0.00887574	1.0000000	6	676

19-25. plot of <pc>, add a cumulative lines, place grey box, add left&right side axis



Additional Plots & Analysis





These additional boxplots reveal the average length and weight of these fish. If I have a project with a fish processing company, I can discuss about factory equipment, based on the above boxplot. A bluegill needs a machine that covers a length of 239-152, and it can be explained that the fish will weigh approximately 71-295. I can make similar version of discussion for Largemouth Bass. This data could be used for optimization which is to select the best possible profits for the company.

Summary

In this module, I was able to calculate the frequency and cumulative number of raw data by using ‘cumsum()’ and ‘table() / length()’. It was found that Bluegill and Largemouth Bass were the two most frequent ‘species’. Those two accounted for 66.3% of the total fish counts. The cumulative graph showed the significant and highest jump of these two species.

Bibliography

Kabacoff, R. I. (2015). R In Action: Data Analysis and Graphics with R. Manning Publications.

Bluman, A. (2017). *Elementary Statistics: A Step By Step Approach* (10th ed.). McGraw-Hill Higher Education (US). https://reader2.yuzu.com/books/9781260042054

Iuchonacho. (2016, December 1). Load multiple packages at once. stackoverflow. Retrieved from https://stackoverflow.com/questions/8175912/load-multiple-packages-at-once

Eric Cai. (2015, February 3). How to Get the Frequency Table of a Categorical Variable as a Data Frame in R.R-BLOGGERS. Retrieved from https://www.r-bloggers.com/2015/02/how-to-get-the-frequency-table-of-a-categorical-variable-as-a-data-frame-in-r/

Joachim Schork. (n.d). Calculate Percentage in R. Statistics Globe. Retrieved from https://statisticsglobe.com/calculate-percentage-in-r

Shane. (2009, December 1). Rotating axis labels in R. stackoverflow. Retrieved from https://stackoverflow.com/questions/1828742/rotating-axis-labels-in-r

Sorting in R using order() Tutorial. (n.d.). datacamp. Retrieved from https://www.datacamp.com/tutorial/sorting-in-r

Erik Marsja. (2020, November 8). How to Add a Column to a Dataframe in R with tibble & dplyr. Blog of Erik. Retrieved from https://www.marsja.se/how-to-add-a-column-to-dataframe-in-r-with-tibble-dplyr/

hackedpersona. (2020, July 1). Levels function returning NULL. 1st quartile and last quartile?. stackoverflow. Retrieved from https://stackoverflow.com/questions/48654453/levels-function-returning-null

NPE. (2011, July 20). how to increase the limit for max.print in R. stackoverflow. Retrieved from https://stackoverflow.com/questions/6758727/how-to-increase-the-limit-for-max-print-in-r

hamed kamel. (2021, October 18). create a parameter variable <def_par> to store parameter variables. stackoverflow. Retrieved from https://stackoverflow.com/questions/69619366/create-a-parameter-variable-def-par-to-store-parameter-variables

Appendix: The R Script

```
print("Heejae Roh")
install.packages("easypackages")
install.packages(c("FSA","FSAdata","magrittr","dplyr","tidyr","plyr","tidyverse"))
library(easypackages)
libraries("FSA", "FSAdata", "magrittr","dplyr","tidyr","plyr","tidyverse")
setwd("C:\\Users\\14083\\Desktop\\exacutive summary\\Project 3")
bio <- read.csv("inchBio.csv", header=T,
                        stringsAsFactors = T)
options(max.print=999999)
bio
headtail(bio,1)
str(bio)
counts <- count(bio, "species")
counts
levels(bio$species)

tmp <- count(bio, "species")
tmp
tmp2 <- subset(bio, select = "species")
head(tmp2, 5)

w <- table(bio$species)
w
class(w)
t <- as.data.frame(w)
t

cSpec <- w
cSpec
cSpecPct <- table(bio$species) / length(bio$species)
cSpecPct
class(cSpecPct)

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

u
sum(u$Freq) #Double_Checking

opar <- par(no.readonly = TRUE)
par(fig=c(0.2, 1, 0, 1))
barplot(cSpec, main="Fish Count", xlim =c(0,250), col="lightgreen", xlab="COUNTS",
horiz=T, cex.axis=.6, las=2,)
par(opar)
```

```
opar <- par(no.readonly = TRUE)
barplot(cSpecPct, main="Fish Relative Frequency", ylim=c(0, 4), yaxt="n")
axis(2, col.axis = "lightblue", cex.axis = 2)
par(opar)

u
d <- u[order(u$Freq, decreasing = TRUE),]
d
install.packages("reshape")
library(reshape)
d <- rename(d, c(Var1 = "Species", Freq = "RelFreq"))
d$cumfreq <- cumsum(d$RelFreq)
d$counts <- d$RelFreq*676
d$cumcounts <- cumsum(d$counts)
d

attach(d)
def_par <- par(no.readonly = TRUE)
par(fig=c(0, 0.9, 0.1, 1))
pc <- barplot(height=d$counts, width=1, space=.15, border=NA, axes=F,
        ylim=c(0, 3.05*max(d$counts, na.rm=TRUE)), ylab="Cummulative Counts",
        cex.names=.7, names.arg=d$Species, main="Species Pareto", las=2)
lines(pc, cumcounts, type="b", cex=.7, pch=19, col="cyan4")
box(col="grey")
axis(side=2, at=c(0, cumcounts), col.ticks="grey62", col.axis="grey62", cex.axis=.8,
      las=2)
axis(side=4, at=c(0, cumcounts),
        labels=paste(c(0, round(d$cumfreq*100)), "%", sep=""), col.axis="darkcyan",
        col="cyan4", cex.axis=.8, las=2)
text(8, 400, "Roh",    cex=1.5, col="yellow2")
par(def_par)
detach(d)
```

#additional analysis

```
install.packages("sqldf")
library(sqldf)
bio

par(opar)

newdf <- sqldf("select * from bio where 'Bluegill'=species order by -tl",
                        row.names = TRUE)
newdf
opar <- par(no.readonly = TRUE)
par(fig=c(0, 0.5, 0, 1))
boxplot(newdf$tl, staplewex=1, col="blue")
mtext("Total Length of Bluegill", side=3, line=1)
text(y=fivenum(newdf$tl), labels=fivenum(newdf$tl), x=1.3, cex=0.75)
par(fig=c(0.5, 0.9, 0, 1), new = TRUE)
boxplot(newdf$w, staplewex=1, col="blue")
mtext("Weight of Bluegill", side=3, line=1)
fivenum(newdf$w)
text(1.35, 295, "295", cex=.75)
text(1.35, 145, "142.5", cex=.75)
text(1.35, 70, "71", cex=.75)
text(1.35, 25, "10.6", cex=.75)
text(1.35, 5, "0.3", cex=.75)

newdf2 <- sqldf("select * from bio where 'Largemouth Bass'=species order by -tl",
                        row.names = TRUE)
newdf2
str(newdf2)
newdf2.noNA <- na.omit(newdf2)
newdf2.noNA
str(newdf2.noNA)

opar <- par(no.readonly = TRUE)
par(fig=c(0, 0.5, 0, 1))
boxplot(newdf2$tl, staplewex=1, col="yellow")
mtext("Total Length of Largemouth Bass", side=3, line=1)
text(y=fivenum(newdf2$tl), labels=fivenum(newdf2$tl), x=1.3, cex=.75)
par(fig=c(0.5, 0.9, 0, 1), new = TRUE)
boxplot(newdf2.noNA$w, staplewex=1, col="yellow")
mtext("Weight of Largemouth Bass", side=3, line=1)
text(y=fivenum(newdf2.noNA$w), labels=fivenum(newdf2.noNA$w), x=1.35, cex=.75)
par(opar)
```