EXECUTIVE SUMMARY REPORT 2

ALY 6000 INTRODUCTION TO ANALYTICS

RAHUL AVINASH JADHAV

Northeastern University

College of Professional Studies, Northeastern University, Boston, MA 02115

Contact: jadhav.ra@northeastern.edu

Submitted to Professor: Prof. Dr Mary Donhoffner

Date of submission: 10/09/2021

Introduction

In this Executive Summary report, we are going to learn about how to:

- Install and import libraries
- Load dataset from libraries and also print records of the dataset
- make subsets from a dataset
- Visualize a dataset using different plots and make the visualization more interpretable based on categories to understand more information.

Key findings

a) Printing your name at the top of the script including the prefix: “Plotting Basics:”

```
> #Print your name at the top of the script
> Prefix <- "Plotting Basics:"
> name <- readline("Enter your last name : ")
Enter your last name : Jadhav
> print(paste(Prefix,name))
[1] "Plotting Basics: Jadhav"
```

In the above snippet, I have created a variable called 'Prefix' that stores the prefix provided by the instructor. Also, I have created a variable called 'name' that stores an input from a user using the readline() function.

The readline() function takes an input from the user through the terminal

The paste() function in R converts its argument to character strings which is why I have used this function so that I can concat the Prefix and last name and display them out as required.

b) Importing libraries including: FSA, FSAdata, magrittr, dplyr, plotrix, ggplot2, and moments :

```
#Installing and Import libraries
  #a)for installing single packages including dependencies
install.packages("FSA")

  #b)For installing multiple packages including dependencies at once
install.packages(c("FSA","FSAdata","ggplot2","moments"))

  #importing single library
library(FSAdata)

  #importing multiple libraries at once
lapply(c("FSA","magrittr","dplyr","plotrix","ggplot2","moments"),require, character.only = TRUE)
```

The above snippet shows how we can install and import single and multiple packages. While installing packages I have noticed that their dependencies packages will also get installed.

```
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/ralph/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
also installing the dependencies 'magrittr', 'dplyr', 'plotrix'
```

As you can see in the above image, While installing the FSA package, its dependencies ('magrittr', 'dplyr', 'plotrix') were also installed.

c) Load the BullTroutRML2 dataset

```
> #Load the BullTroutRML2 dataset
> BullTroutRMS2 <- BullTroutRML2
> BullTroutRMS2
   age  fl     lake     era
1   14 459 Harrison 1977-80
2   12 449 Harrison 1977-80
3   10 471 Harrison 1977-80
4   10 446 Harrison 1977-80
5    9 400 Harrison 1977-80
6    9 440 Harrison 1977-80
7    9 462 Harrison 1977-80
8    8 480 Harrison 1977-80
9    8 449 Harrison 1977-80
10   7 437 Harrison 1977-80
11   7 431 Harrison 1977-80
12   7 425 Harrison 1977-80
13   7 419 Harrison 1977-80
14   6 409 Harrison 1977-80
15   6 397 Harrison 1977-80
16   5 419 Harrison 1977-80
17   5 381 Harrison 1977-80
18   5 363 Harrison 1977-80
19   5 351 Harrison 1977-80
20   4 372 Harrison 1977-80
21   2 199 Harrison 1977-80
22   2 184 Harrison 1977-80
23   1  91 Harrison 1977-80
24  12 440 Harrison 1997-01
25  11 428 Harrison 1997-01
26  10 440 Harrison 1997-01
```

As the dataset was imported in the project while importing FSA and FSA data libraries, we can store them directly in the variable without using any function as shown above.

d) Print the first and last 3 records from the BullTroutRMS2 dataset

```
> headtail(BullTroutRMS2,n=3)
   age  fl     lake     era
1   14 459 Harrison 1977-80
2   12 449 Harrison 1977-80
3   10 471 Harrison 1977-80
94   4 298   Osprey 1997-01
95   3 279   Osprey 1997-01
96   3 273   Osprey 1997-01
```

in R programming language, The head() function returns the first 'n' rows, whereas the tail() function returns the last 'n' rows. Also, the headtail() function returns the first and last n rows. For this task, to print the first and the last n records from the dataset, we are using the headtail() function.

e) Remove all records except those from Harrison Lake

```
> #Remove all records except those from Harrison Lake
> Harrison_dataset <- filterD(BullTroutRMS2,lake=="Harrison")
Warning message:
'filter' is deprecated and will soon be removed from 'FSA'; please us
ost from 26-May-2021).
> Harrison_dataset
   age  fl     lake     era
1   14 459 Harrison 1977-80
2   12 449 Harrison 1977-80
3   10 471 Harrison 1977-80
4   10 446 Harrison 1977-80
5    9 400 Harrison 1977-80
6    9 440 Harrison 1977-80
7    9 462 Harrison 1977-80
8    8 480 Harrison 1977-80
9    8 449 Harrison 1977-80
10   7 437 Harrison 1977-80
11   7 431 Harrison 1977-80
12   7 425 Harrison 1977-80
13   7 419 Harrison 1977-80
```

For Removing all the records except those from Harrison lake, we will use the filterD(x,y) function. Where x is the dataset and y is the argument.

f) Display the first and last 5 records from the filtered BullTroutRML2 dataset

```
> #Display the first and last 5 records from the filtered BullTroutRML2 dataset
> headtail(Harrison_dataset,n=5)
   age  fl     lake     era
1   14 459 Harrison 1977-80
2   12 449 Harrison 1977-80
3   10 471 Harrison 1977-80
4   10 446 Harrison 1977-80
5    9 400 Harrison 1977-80
57   0  41 Harrison 1997-01
58   0  20 Harrison 1997-01
59   7 245 Harrison 1997-01
60   7 279 Harrison 1997-01
61   5 245 Harrison 1997-01
```

For displaying the first and the last records from the filtered dataset that is Harrison_dataset, We will use the same headtail() function explained above in task D.

g) Display the structure of the filtered BullTroutRML2dataset

```
> #Display the structure of the filtered BullTroutRML2 dataset
> str(Harrison_dataset)
'data.frame':   61 obs. of  4 variables:
 $ age : int  14 12 10 10 9 9 9 8 8 7 ...
 $ fl  : int  459 449 471 446 400 440 462 480 449 437 ...
 $ lake: Factor w/ 1 level "Harrison": 1 1 1 1 1 1 1 1 1 1 ...
 $ era : Factor w/ 2 levels "1977-80","1997-01": 1 1 1 1 1 1 1 1 1 1 ...
```

For displaying the structure of the filtered dataset that is Harrison_dataset, We will use the str() function which is used for fetching the structure of an R object.

h) Display the summary of the filtered BullTroutRML2dataset

```
> #Display the summary of the filtered BullTroutRML2dataset
> summary(Harrison_dataset)
      age              fl            lake         era
 Min.   : 0.000   Min.   : 20   Harrison:61   1977-80:23
 1st Qu.: 3.000   1st Qu.:221                 1997-01:38
 Median : 6.000   Median :372
 Mean   : 5.754   Mean   :319
 3rd Qu.: 8.000   3rd Qu.:425
 Max.   :14.000   Max.   :480
```

For displaying the summary of the filtered dataset that is Harrison_dataset, We will use the summary() function which is used for fetching the summary data of an R object.

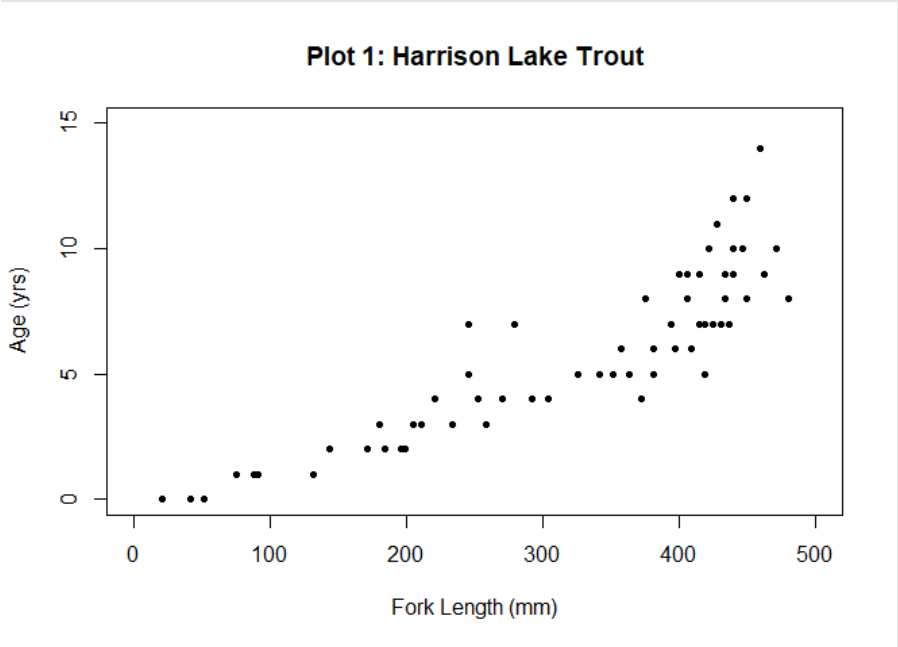
i) Create a scatterplot for “age” (y variable) and “fl” (x variable) with the following specifications:

- Limit of x axis is (0,500)
- Limit of y axis is (0,15)
- Title of graph is “Plot 1: Harrison Lake Trout
- Y axis label is “Age (yrs)”
- X axis label is “Fork Length (mm)”
- Use a small filled circle for the plotted data points

Code Snippet :

```
> #Create a scatterplot for “age” (y variable) and “fl” (x variable) with the provided specifications
> plot(Harrison_dataset$fl,
+      Harrison_dataset$age,
+      xlab ="Fork Length (mm)",
+      ylab = "Age (yrs)",
+      xlim=c(0,500),
+      ylim=c(0,15),
+      pch=20,
+      title("Plot 1: Harrison Lake Trout"))
```

Scatter plot :



In this task, we have plotted a scatter plot to check the relationship between the age and the fork length of the Bulltrout fish in Harrison lake. We have also used arguments to change: the limit of the x and the y axis, the Title, the label of the x and the y axis, and the shape of the data points.

As you can see from the plotting above, we can determine that the fork length increase as the age of the increases.

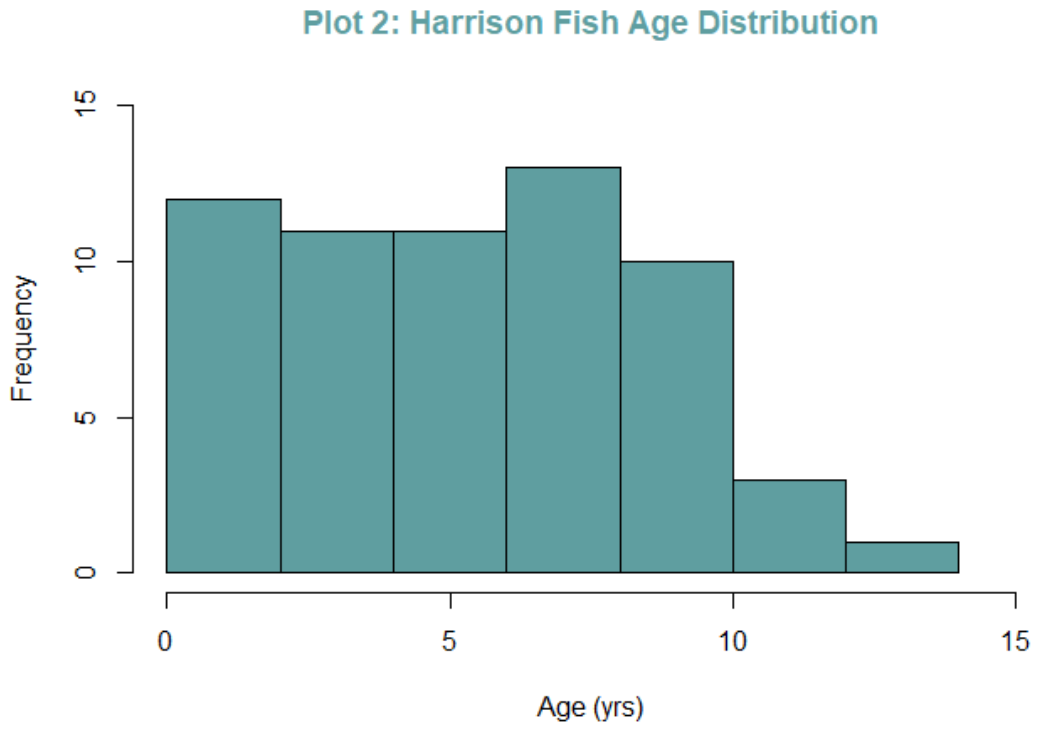
j) Plot an “Age” histogram with the following specifications:

- Y axis label is “Frequency”
- X axis label is “Age (yrs)”
- Title of the histogram is “Plot 2: Harrison Fish Age Distribution” X and Y axis limits is 0, 15
- The color of the frequency plots is “cadetblue”
- The color of the Title is “cadetblue

Console Snippet :

```
> #Plot an "Age" histogram with the provided specifications
> attach(Harrison_dataset)
The following objects are masked from Harrison_dataset (pos = 3):

    age, era, fl, lake

> hist(age,
+      xlab="Age (yrs)",
+      ylab = "Frequency",
+      main="Plot 2: Harrison Fish Age Distribution", col.main = "cadetblue",
+      xlim = c(0,15),
+      ylim = c(0,15),
+      col = "cadetblue",
+      )
```

Histogram :



In this task, we have plotted a histogram of age using hist() function and used arguments to change: the limit of the x and the y axis, the Title, the label of the x and the y axis, and the shape and color of the data points. We have also used attach() which is used to access the variables present in the data set without calling the data set

From the histogram plotting, we can see that:

- The frequency of trout fish of age 6 to 8 years is more than that of age 12 to 14.
- We can also say that frequency of BullTrout fish of age 8-14 years in Harrison lake is less.
- The data is right screwed.
- The lifespan of BullTrout fish is 14 years in Harrison lake.

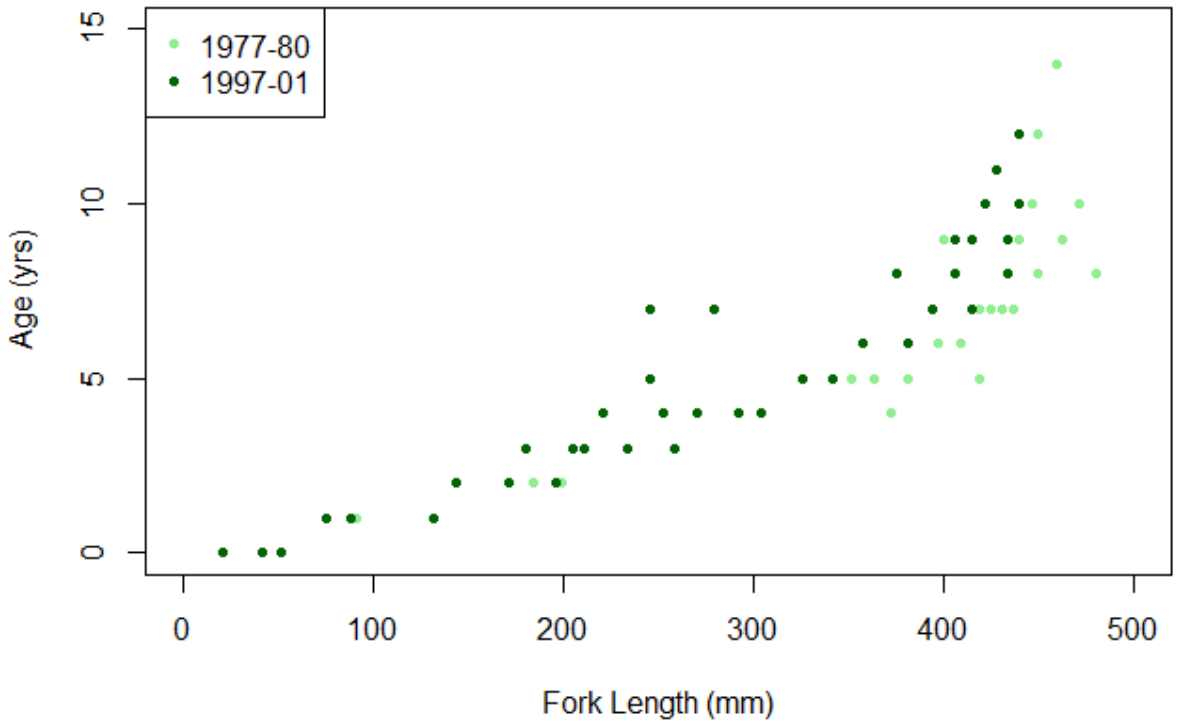
k) Create an overdense plot using the same specifications as the previous scatterplot :

- Title the plot “Plot 3: Harrison Density Shaded by Era”
- Y axis label is “Age (yrs)”
- Y axis limits are 0 to 15
- X axis label is “Fork Length (mm)”
- X axis limits are 0 to 500
- include two levels of shading for the “green” data points.
- Plot solid circles as data points

Console snippet:

```
> #creating the density plot
> cols <-c("lightgreen","darkgreen")
> attach(Harrison_dataset)
The following objects are masked from Harrison_dataset (pos = 3):

    age, era, fl, lake

The following objects are masked from Harrison_dataset (pos = 4):

    age, era, fl, lake

The following objects are masked from Harrison_dataset (pos = 5):

    age, era, fl, lake

The following objects are masked from Harrison_dataset (pos = 6):

    age, era, fl, lake

The following objects are masked from Harrison_dataset (pos = 7):

    age, era, fl, lake

> cols
[1] "lightgreen" "darkgreen"
> plot(fl,
+      age,
+      xlab ="Fork Length (mm)",
+      ylab = "Age (yrs)",
+      xlim=c(0,500),
+      ylim=c(0,15),
+      pch=20,
+      col=cols[era])
> legend(x="topleft",legend = paste(levels(era)),col=cols,pch=20)
```

Plotting :



To make the scatter plot of task I more understandable, we have given two shades of green to the data point to distinguish between the dataset from 1977-80 and 1997-01 to differentiate which data point belongs to which era.

l) Create a new object called “tmp” that includes the first 3 and last 3 records of the BullTroutRML2 data set.

```
> #Create a new object called "tmp" that includes the first 3 and last 3 records of the BullTroutRML2 data set
> tmp <- headtail(BullTroutRML2,n=3)
> tmp
   age  fl     lake     era
1   14 459 Harrison 1977-80
2   12 449 Harrison 1977-80
3   10 471 Harrison 1977-80
94   4 298   Osprey 1997-01
95   3 279   Osprey 1997-01
96   3 273   Osprey 1997-01
```

In this task we are going to create a new object ‘tmp’, which will contain first 3 and last 3 records of BullTroutRML2 data set. We have used the headtail() function that will return the first and last three records of the dataset and store that record to the tmp object.

m) Display the “era” column (variable) in the new “tmp” object

```
> #Display the “era” column (variable) in the new “tmp” object
> tmp$era
[1] 1977-80 1977-80 1977-80 1997-01 1997-01 1997-01
Levels: 1977-80 1997-01
```

We have displayed the content of column era from the tmp object by using the ‘$’ symbol. We use the ‘$’ symbol to select the variable or column from the dataset.

n) Create a pchs vector with the argument values for + and x.

```
> #Create a pchs vector with the argument values for + and x
> pchs <- c("+","x")
> pchs
[1] "+" "x"
```

We have created a vector called pchs with values ‘+’ and ‘x’ and displayed the vector.

o) Create a cols vector with the two elements “red” and “gray60”

```
> # Create a cols vector with the two elements “red” and “gray60”
> cols <- c("red","gray60")
> cols
[1] "red"    "gray60"
```

We have created a vector called cols with values “red” and “gray60” and displayed the vector

p) Convert the tmp era values to numeric values.

```
> # Convert the tmp era values to numeric values.
> #before
> tmp$era
[1] 1977-80 1977-80 1977-80 1997-01 1997-01 1997-01
Levels: 1977-80 1997-01
>
> #after
> tmp$era <- as.numeric(tmp$era)
> tmp$era
[1] 1 1 1 2 2 2
```

Here we have converted the tmp era values to numeric values using as.numeric() function. The as.numeric() function returns the converted numeric value.

q) Initialize the cols vector with the tmp era values

```
> #Initialize the cols vector with the tmp era values
> initialize(cols,tmp$era)
[1] "1" "1" "1" "2" "2" "2"
```

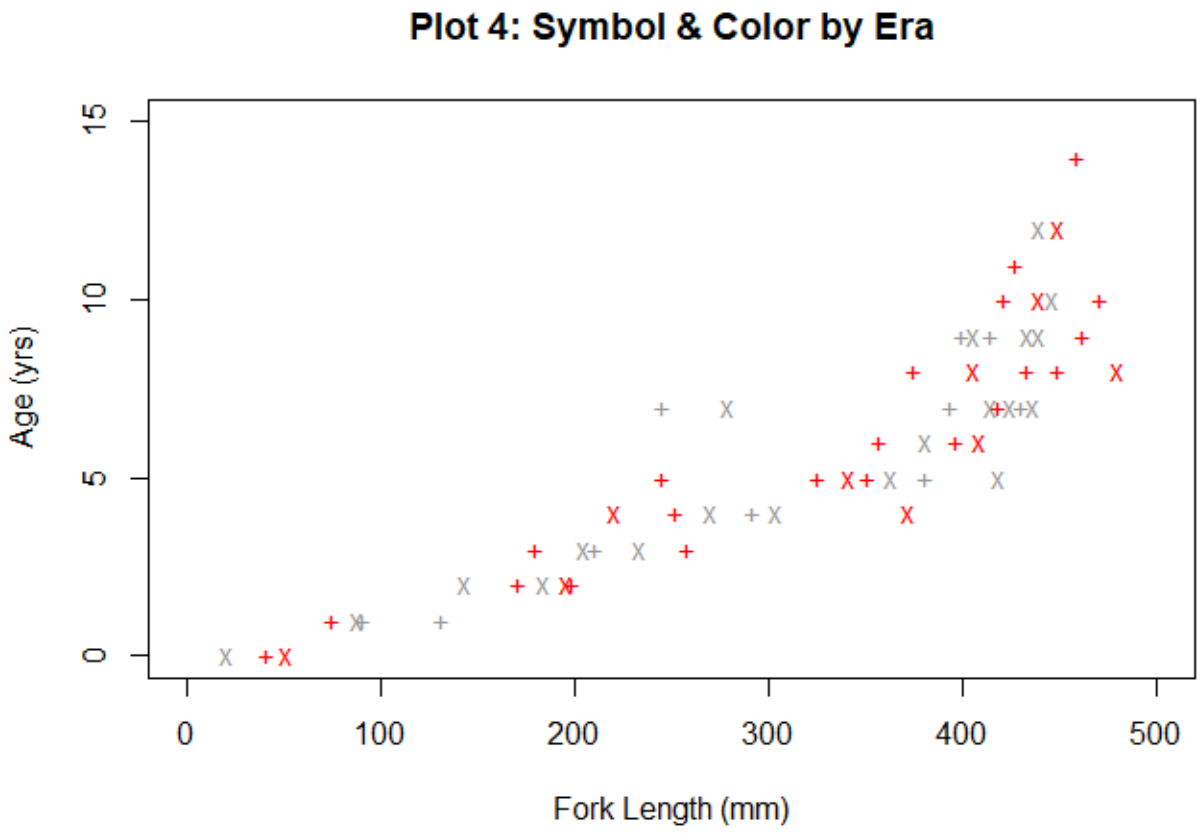
Here we have initialized the cols vector with tmp era values using initialize() function.

r) Create a plot of “Age (yrs)” (y variable) versus “Fork Length (mm)” (x variable) with the following specifications:

- Title of graph is “Plot 4: Symbol & Color by Era”
- Limit of x axis is (0,500)
- Limit of y axis is (0,15)
- X axis label is “Age (yrs)”
- Y axis label is “Fork Length (mm)”
- Set pch equal to pchs era values
- Set col equal to cols era values

```
> #Create a plot of “Age (yrs)” (y variable) versus “Fork Length (mm)” (x variable) with the following specifications
> plot(Harrison_dataset$fl,
+      Harrison_dataset$age,
+      xlab ="Fork Length (mm)",
+      ylab = "Age (yrs)",
+      xlim=c(0,500),
+      ylim=c(0,15),
+      pch=pchs,
+      col=cols[as.numeric(tmp$era)],
+      title("Plot 4: Symbol & Color by Era"))
```

Plotting:



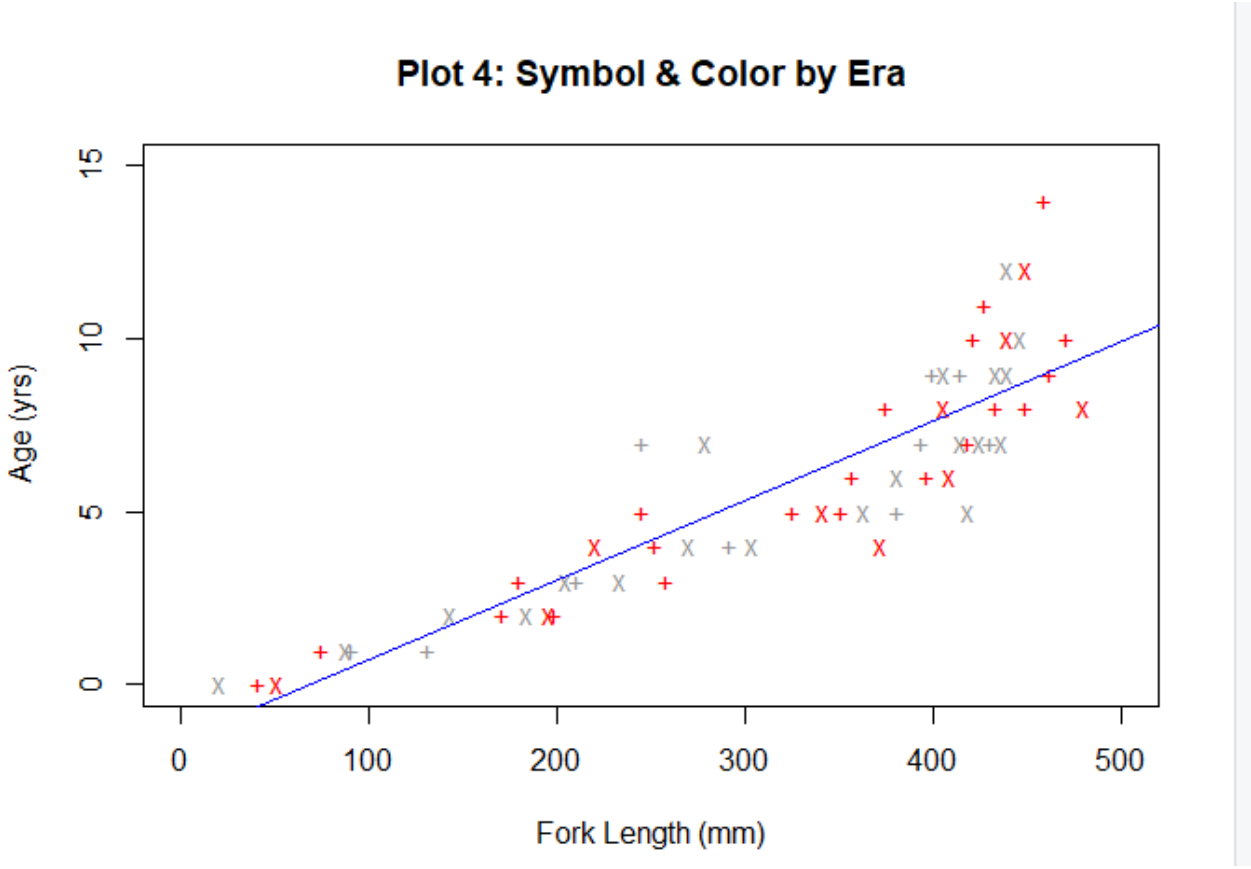
In this task, we have plotted a scatterplot using the plot() function and used arguments to change: the limit of the x and the y axis, the Title, the label of the x and the y axis. We have also used vector pchs and cols to set the shape and color of the dataset.

In the above plot, we can see that there are two symbols which are “x” and “+ ”and two colors which are “red” and “gray60” to differentiate the data points.

s) Plot a regression line overlay on Plot 4 and title the new graph “Plot 5: Regression Overlay”:

```
> plot(Harrison_dataset$fl,
+      Harrison_dataset$age,
+      xlab ="Fork Length (mm)",
+      ylab = "Age (yrs)",
+      xlim=c(0,500),
+      ylim=c(0,15),
+      pch=pchs,
+      col=cols[as.numeric(tmp$era)],
+      title("Plot 4: Symbol & Color by Era"))
> #Plot a regression line overlay on Plot 4 and title the new graph “Plot 5: Regression Overlay”
> abline(lm(Harrison_dataset$age ~ Harrison_dataset$fl, data = Harrison_dataset), col = "blue")
```

Plotting :



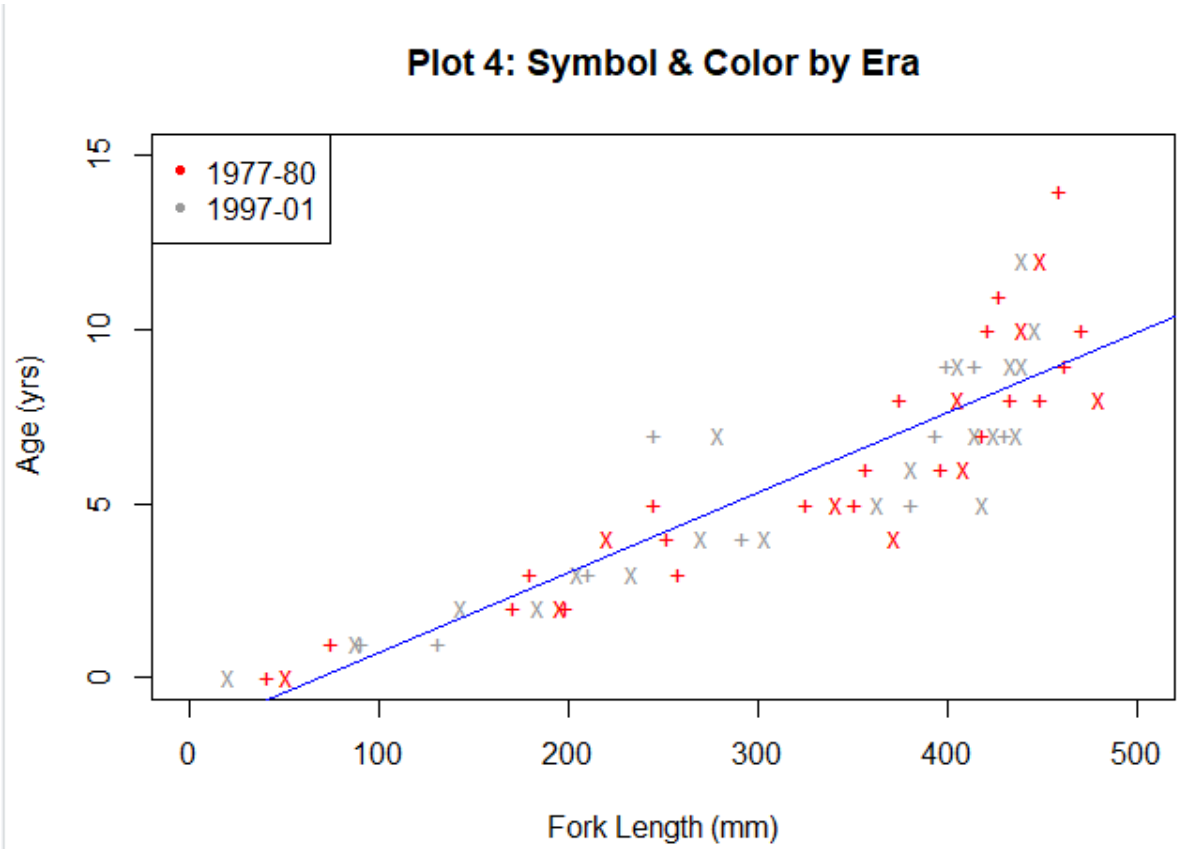
In this task, we are going to plot a regression line on the previous plot using the abline() function used to add vertical, horizontal, or regression lines to a graph. This line is the Regression line that helps to find the relation between the x and y variables.

From the plotting, We can say that the regression line is increasing as we go higher in the data set.

t) Place a legend of on Plot 5 and call the new graph “Plot 6: :Legend Overlay:

```
> plot(Harrison_dataset$fl,
+      Harrison_dataset$age,
+      xlab ="Fork Length (mm)",
+      ylab = "Age (yrs)",
+      xlim=c(0,500),
+      ylim=c(0,15),
+      pch=pchs,
+      col=cols[as.numeric(tmp$era)],
+      title("Plot 4: Symbol & Color by Era"))
> #Plot a regression line overlay on Plot 4 and title the new graph “Plot 5: Regression Overlay"
> abline(lm(Harrison_dataset$age ~ Harrison_dataset$fl, data = Harrison_dataset), col = "blue")
> # Place a legend of on Plot 5 and call the new graph “Plot 6: :Legend Overlay
> legend(x="topleft",legend = c("1977-80","1997-01"),col=cols,pch=20)
```

Plotting:



In this task, we will add a legend to the previous plot and change its title.

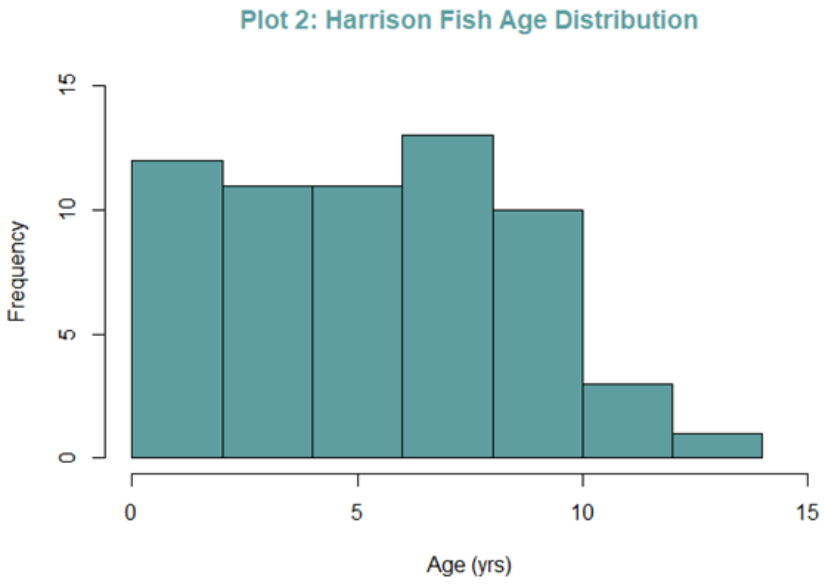
For adding a legend, we will use the legend() function which is used to place a legend with information describing the elements to the plot.

From the above plot, we can see that the data points are of two colors 'grey60' and 'red'. The 'grey60' represents data point from the 1997-01 era, and 'red' represent data points from the 1977-80 era.

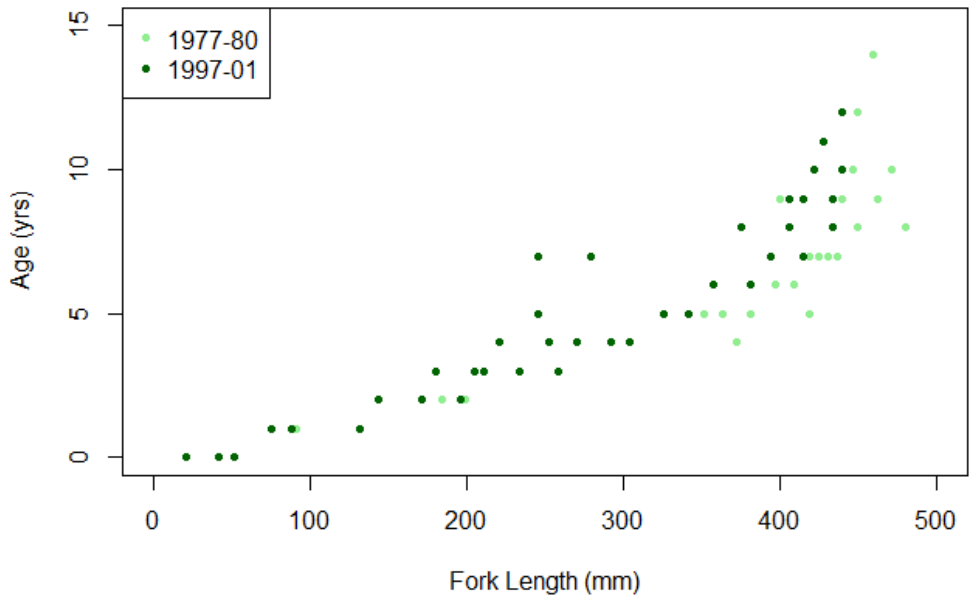
u) Summary

In this assignment, we have learned how to install packages and use them. We also learned to use the pre dataset BullTroutRML2, which is from the library name FSADATA.
The dataset was about the population of bull trout in different lakes and different periods.
The data is from the 1977-80 and 1997-01 era. In this assignment, we learned how to filter data and make a subset from the Main dataset. We also learned how to: check the structure and summary of the objects,t
types of plots like a scatterplot, histogram, density plot, and about the plotting arguments which used to make the plotting more understandable. Later we also learned how to use vector elements in the function argument that help to make the graphs interpretable to identify the data points based on categories present in the dataset. We also learned about how to draw a regression line that shows the linear relationship between 2 variables. Lastly, we learned about the legend() function used to describe the elements of the graph.

Data Analysis



- From the above Histogram, we can say that the average life span of Bull trout fish in Harrison lake is 6year's whereas the oldest age is 14 years.



- From the above scatter plot, we can also say that as the fish age increases, its fork length also increases. The average fork length is 319mm whereas the highest fork length is 480mm.
- we can also say that the tiniest fork length of fish in from 1997-01 era and the highest fork length of fish from 1977-80 era.
- The fish from 1977-80 is the oldest Bull trout fish in Harrison Lake with 14years of age.

Bibliography

- R programming: Take input from the user and display the values, print the version of R installation. w3resource. (n.d.). Retrieved October 4, 2021, from https://www.w3resource.com/r-programming-exercises/basic/r-programming-basic-exercise-1.php.
- R packages: How to download & install packages? DataCamp Community. (n.d.). Retrieved October 4, 2021, from https://www.datacamp.com/community/tutorials/r-packages-guide.
- Talagala, T. S. (2019, March 22). How to install and load multiple packages at once? THIYANGA TALAGALA. Retrieved October 4, 2021, from https://thiyanga.netlify.app/post/multiplepkg/.
- R_UserR_User 9, AlphaAlpha 77711 gold badge99 silver badges1414 bronze badges, TungTung 21.2k66 gold badges7272 silver badges8787 bronze badges, & stevecstevec 19.1k77 gold badges8585 silver badges130130 bronze badges. (1961, October 1). Colorize parts of the title in a plot. Stack Overflow. Retrieved October 4, 2021, from https://stackoverflow.com/questions/17083362/colorize-parts-of-the-title-in-a-plot.

Appendix

```
#Print your name at the top of the script
Prefix <- "Plotting Basics:"
name <- readline("Enter your last name : ")
print(paste(Prefix,name))

#checking version
R.Version()

#Installing and Import libraries
  #a)for installing single packages including dependencies
install.packages("FSA")

  #b)For installing multiple packages including dependencies at once
install.packages(c("FSA","FSAdata","ggplot2","moments"))

  #importing single library
library(FSAdata)

  #importing multiple libraries at once
lapply(c("FSA","magrittr","dplyr","plotrix","ggplot2","moments"),require, character.only
= TRUE)



#Load the BullTroutRML2 dataset
BullTroutRMS2 <- BullTroutRML2
BullTroutRMS2

#Print the first and last 3 records from the BullTroutRMS2 dataset
```

```
 #if dataset is loaded

headtail(BullTroutRMS2,n=3)

 #if dataset is imported to the project
headtail(BullTroutRML2,n=3)

#Remove all records except those from Harrison Lake
Harrison_dataset <- filterD(BullTroutRMS2,lake=="Harrison")
Harrison_dataset

#Display the first and last 5 records from the filtered BullTroutRML2 dataset
headtail(Harrison_dataset,n=5)

#Display the structure of the filtered BullTroutRML2 dataset
str(Harrison_dataset)

#Display the summary of the filtered BullTroutRML2dataset
summary(Harrison_dataset)

#Create a scatterplot for "age" (y variable) and "fl" (x variable) with the provided specifications
plot(Harrison_dataset$fl,
    Harrison_dataset$age,
    xlab ="Fork Length (mm)",
    ylab = "Age (yrs)",
    xlim=c(0,500),
    ylim=c(0,15),
    pch=20,
    title("Plot 1: Harrison Lake Trout"))
```

```
#Plot an "Age" histogram with the provided specifications
attach(Harrison_dataset)
hist(age,
    xlab="Age (yrs)",
    ylab = "Frequency",
    main="Plot 2: Harrison Fish Age Distribution", col.main = "cadetblue",
    xlim = c(0,15),
    ylim = c(0,15),
    col = "cadetblue",
    )

#creating the density plot
cols <-c("lightgreen","darkgreen")
attach(Harrison_dataset)
cols
plot(fl,
    age,
    xlab ="Fork Length (mm)",
    ylab = "Age (yrs)",
    xlim=c(0,500),
    ylim=c(0,15),
    pch=20,
    col=cols[era])
legend(x="topleft",legend = paste(levels(era)),col=cols,pch=20)


#Create a new object called "tmp" that includes the first 3 and last 3 records of the
BullTroutRML2 data set
```

```
tmp <- headtail(BullTroutRML2,n=3)
tmp

#Display the "era" column (variable) in the new "tmp" object
tmp$era

#Create a pchs vector with the argument values for + and x
pchs <- c("+","x")
pchs

# Create a cols vector with the two elements "red" and "gray60"
cols <- c("red","gray60")
cols

# Convert the tmp era values to numeric values.
#before
tmp$era

#after
tmp$era <- as.numeric(tmp$era)
tmp$era

#Initialize the cols vector with the tmp era values
initialize(cols,tmp$era)

#Create a plot of "Age (yrs)" (y variable) versus "Fork Length (mm)" (x variable) with the following specifications:
plot(Harrison_dataset$fl,
    Harrison_dataset$age,
    xlab ="Fork Length (mm)",
```

```
    ylab = "Age (yrs)",
    xlim=c(0,500),
    ylim=c(0,15),
    pch=pchs,
    col=cols[as.numeric(tmp$era)],
    title("Plot 4: Symbol & Color by Era"))


#Plot a regression line overlay on Plot 4 and title the new graph "Plot 5: Regression Overlay"
abline(lm(Harrison_dataset$age ~ Harrison_dataset$fl, data = Harrison_dataset), col = "blue")


# Place a legend of on Plot 5 and call the new graph "Plot 6: :Legend Overlay
legend(x="topleft",legend = c("1977-80","1997-01"),col=cols,pch=20)
```