Second Class: R 4 Beginners

Maria Jose Rodriguez Barrera

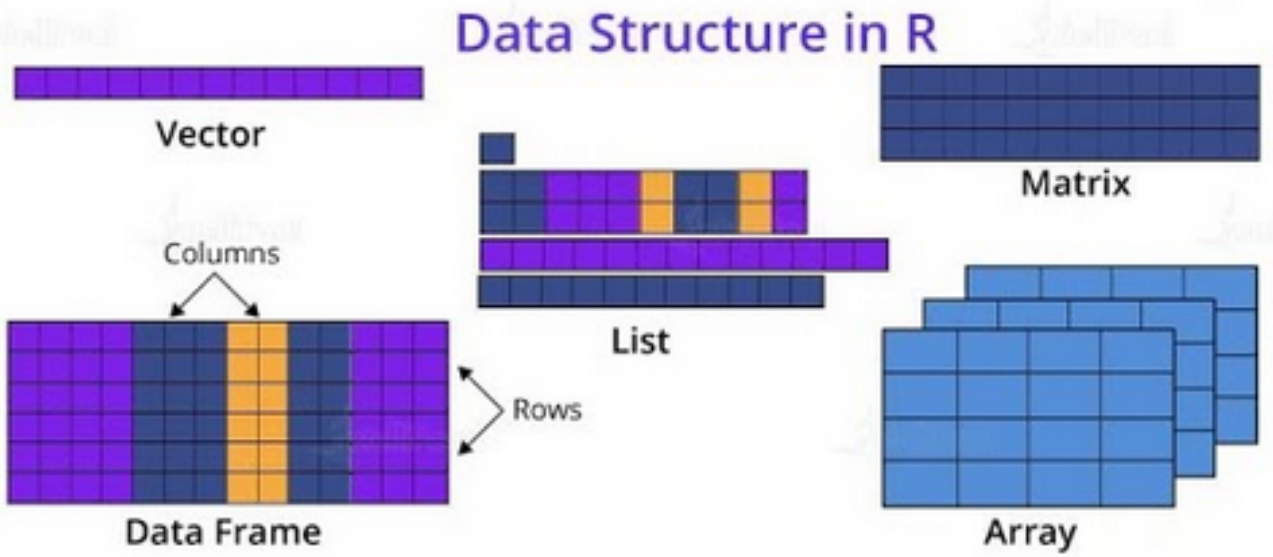
2024-11-4

Class Github

- Link to Class Github: https://github.com/MajoRB15/StanfordCourse_R4beginners

We're working in our R project

Types of Data in R



Comparison Operators

Symbols	Use
`<`	`Less than`
`>`	`Bigger than`
`==`	`Left operand the same as the right operand?`
`!=`	`Left operand different from the right operand?`
`<=`	`Less than or equal to`
`>=`	`Bigger than or equal to`

Outputs are logical values (TRUE or FALSE) for all

Logical Operators

Symbols	Use
&	AND, both left and rightshould be TRUE
|	Or, if either of the conditionals is TRUE
!	NOT, it returns the opposite of a given condition

Data Frames

Creation

- Two dimensions

```
#How to create one
#data.frame()
data1= data.frame(
  names= c("Andy","Mia","Greg"),
  age= c(14, 50, 5)
) #It will create a data frame named data1 with 2 columns,
  #one called names and the other called age
```

Data Frames

Acess to data

- Take into account that we have columns that mean different things
- Reminder: Dataframes are two dimensions

```
#--Accessing to the information of column 1-> names --
data1[,1] #dataframe[lines, columns]
data1$names #dataframe$column_name

#-- Acessing to specific data--
data1[2,1] #dataframe[lines, columns] #it will print "Mia"
data1$names=="Mia" #It will print "Mia"
```

Data Frames

Some commands

```
#---- Rows ----
nrow(data1) #Number of rows
row.names(data1) #Rows Names

#---- Columns ----
ncol(data1) #Number of columns
colnames(data1) #Column names
```

Lists

Lists

- You can have different types of data
- You can also have different dimensions!
- Access to data is different (as always)

Lists

Creation

```
#Creating a List
list1= list(c("Apple", "Melon"), "Majo", 1:6)
#It creates a list of sublists (one vector of fruit words,
            #a character and a vector of numbers)
```

Lists

Creation

```
#---- Better organization ----
#First a create my sublists
myvector <- 1:10
mymatrix <- matrix(1:4, nrow = 2) #R would create a column
                                  #if we dont give any number to ncol
mydf  <- data.frame("names" = c("Andy", "Mia", "Greg"),
                    "age"= c(14, 50, 5))

#I merged them in a list and add names to the sublists
mylist= list(
  vector= myvector,
  matrix= mymatrix,
  dataset= mydf
)
```

Lists

Acess to the data

```
# Printing the first element/sublist in my List
mylist[[1]] #List[[sublist]]

#Printing the fourth element in the first sublist
mylist[[1]][4]  #List[[sublist]][element]

#You can also do:
mylist$vector[4] #List$sublist[element]
```

Eliminating data from a List

```
#Eliminating a sublist
mylist[[1]]=NULL
```

Exercises: DataSets and Lists

- From the variable "data1" add another column named "Color" that has the information for favorite color
- Change the name of the third column (Color) to "fav.color"

Data frame should look like this:

names	**age**	**fav.color**
Andy	14	Black
Mia	50	Yellow
Greg	5	Green

Answers

```
# Adding favorite color to data1
data1= data.frame(data1, Color=c("Black","Yellow","Green"))

#Changing the column name to fav.color
colnames(data1)[3]="fav.color"
```

Basic Useful Fuctions in R

Function	Use
`na.omit()`	Eliminate NA values
`subset()`	To keep data given a condition
`unique()`	Unique Values
`duplicated()`	Returns duplicated data
`cbind()`	Binds columns of the same dimensions
`rbind()`	Binds rows that have same columns
`table()`	Counting values in a table format
`dim()`	Obtaining the dimensions of data
`length()`	Obtaining the Length of a vector
`which()`	Obtaining positions for a value given a condition

Google for Math fuctions! There are plenty, some are: sum(), mean(), min(), max(), var(), etc

Penguins Data set

- 344 penguins
- 3 different species of penguins
- Penguins origin from 3 islands in the Palmer Archipelago, Antarctica.

Group Exercise

Working with Penguins Data Set

- Install the palmerpenguins package:

```
install.packages("palmerpenguins")
```

- Upload palmerpenguin dataset:

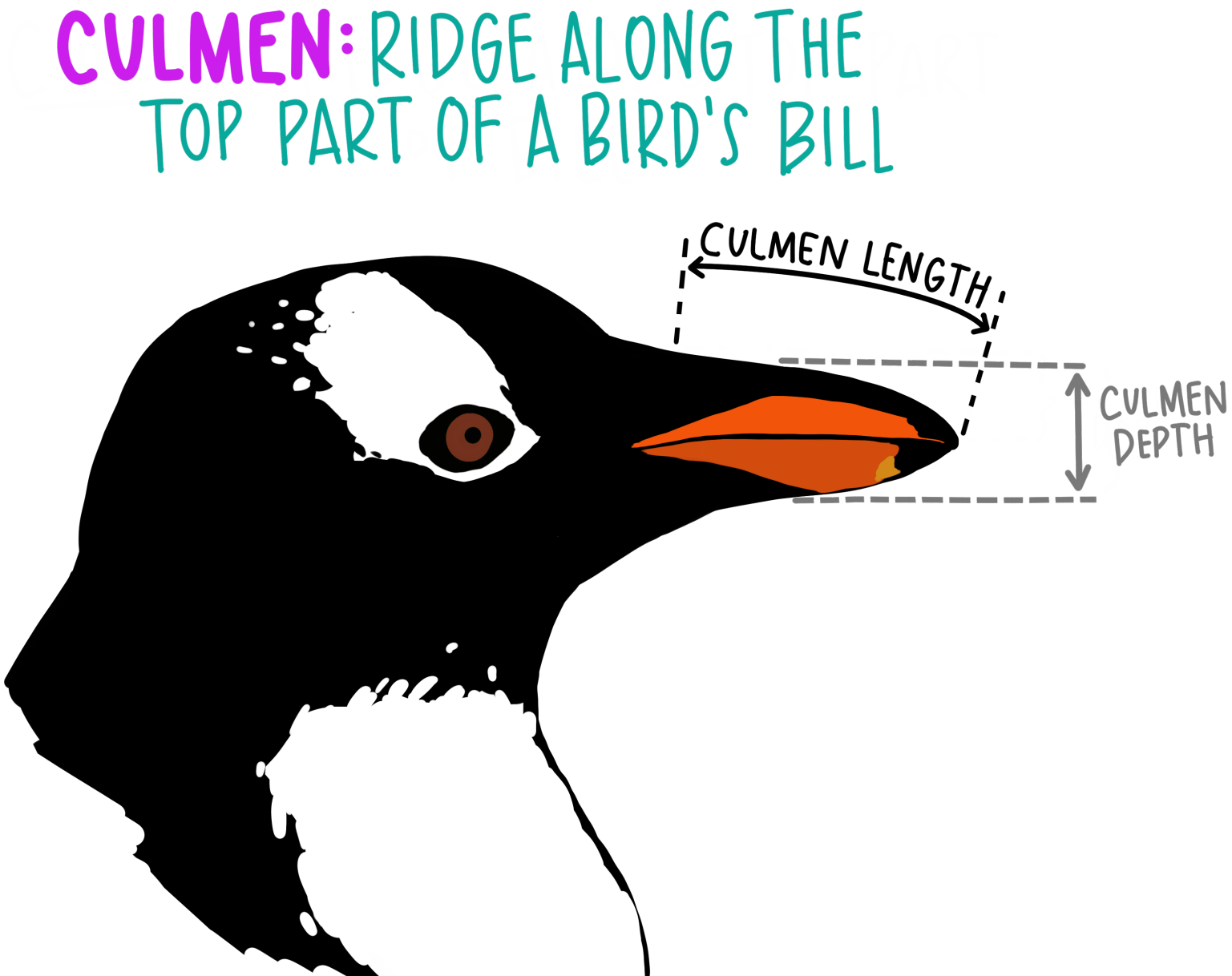
```
library(palmerpenguins)
data(package = 'palmerpenguins')
```

- **Get in touch with the data**

Penguins Data

```r
#How does data looks like?
head(penguins) #Printing the head of the data
head(penguins, 5) #printing the first 5 lines

#What types of variables we have?
#Do these variables make sense?
str(penguins)

#Data Dimensions
dim(penguins)
```

Understanding Data



Exercises

- Get rid of individuals with NA values
- How many male and females penguins do we have?
- Print the names of the 3 Species the data set has: “Adelie Gentoo Chinstrap”
- Save penguins sampled by 2007 in a variable called “Year2007”
- Obtain the positions for the penguins that are male AND were sampled in 2009. How many are them?

Answers

```r
#Eliminating penguins that have NA values
penguins = na.omit(penguins)

#Amount of males and females
table(penguins$sex)

#Names of species in the data set
unique(penguins$species)
table(penguins$species)
```

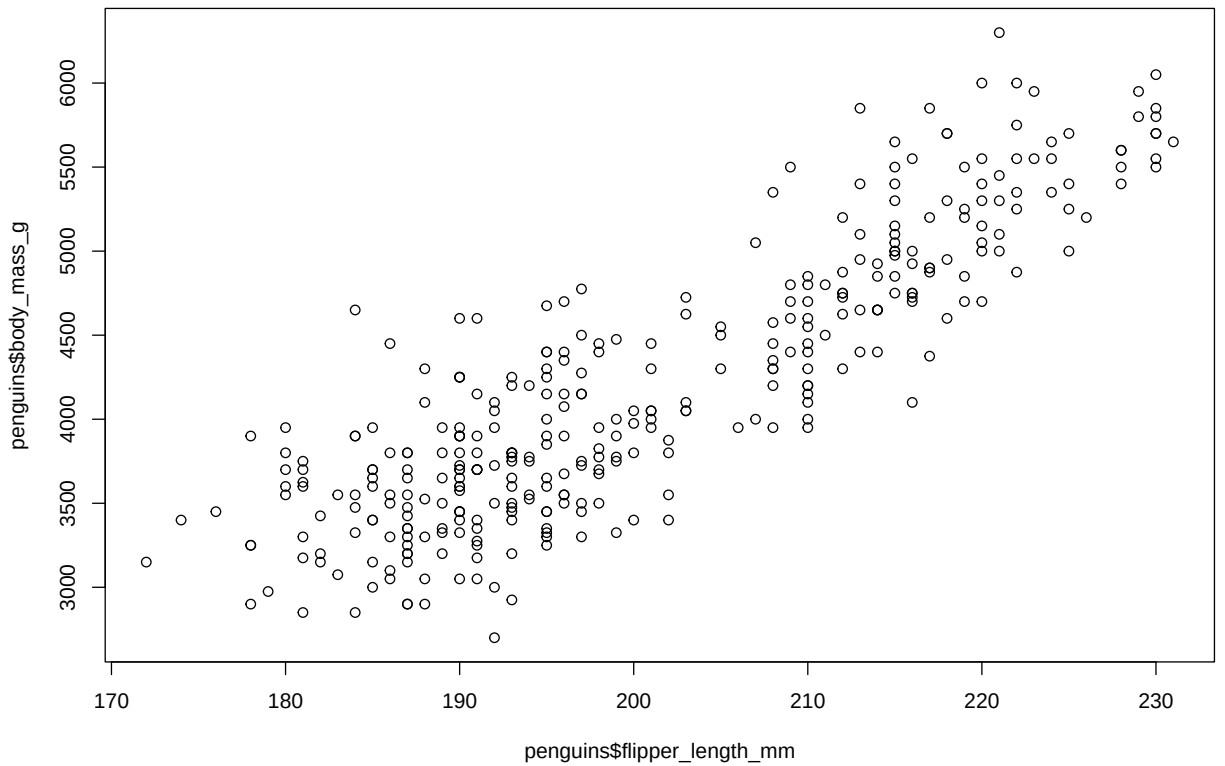
Answers

```r
#Positions for the penguins that are male AND were sampled
which(penguins$sex== "male" & penguins$year== 2009)
#Amount of them:
length(which(penguins$sex== "male" & penguins$year== 2009))

#Subset for 2007 samples
subset1= subset(penguins, penguins$year==2007)
```

Easy Plots

Linear Plot

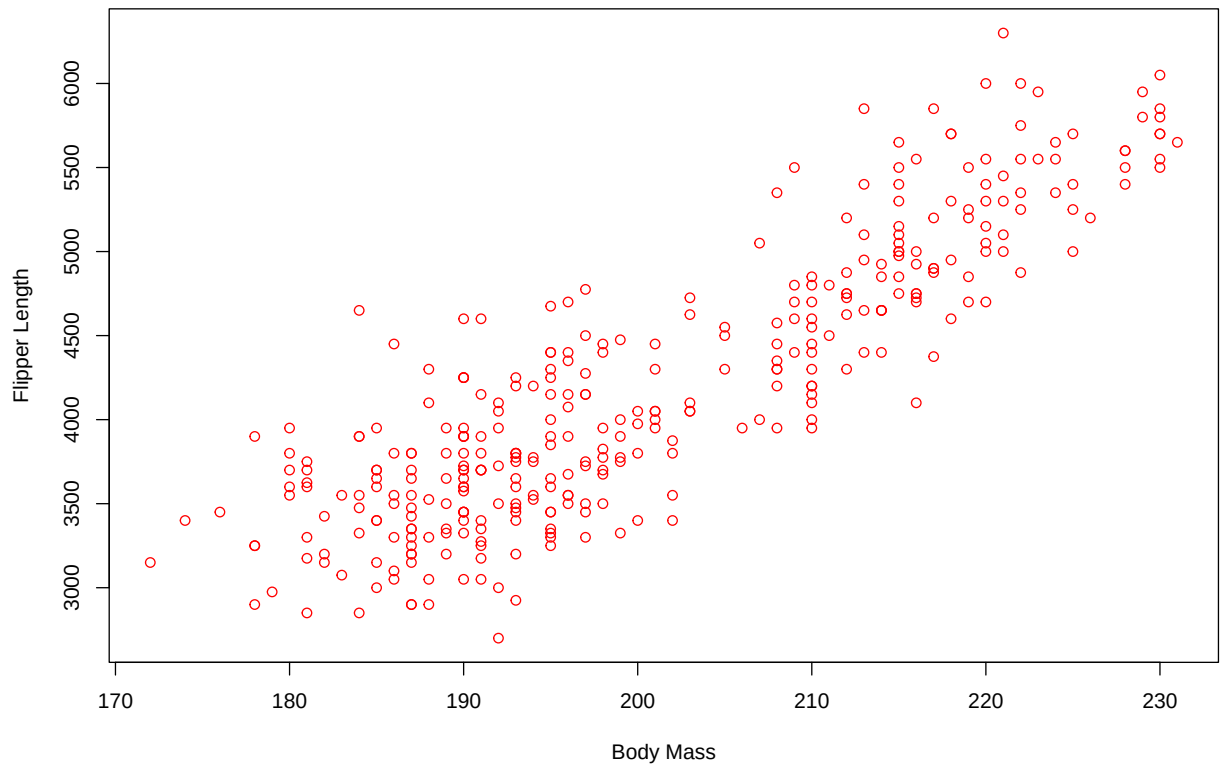
```
plot(penguins$flipper_length_mm, penguins$body_mass_g)
```



Easy Plots

Customize our Linear Regression

```
plot(penguins$flipper_length_mm, penguins$body_mass_g,
     xlab="Body Mass", ylab="Flipper Length", col= "red")
```

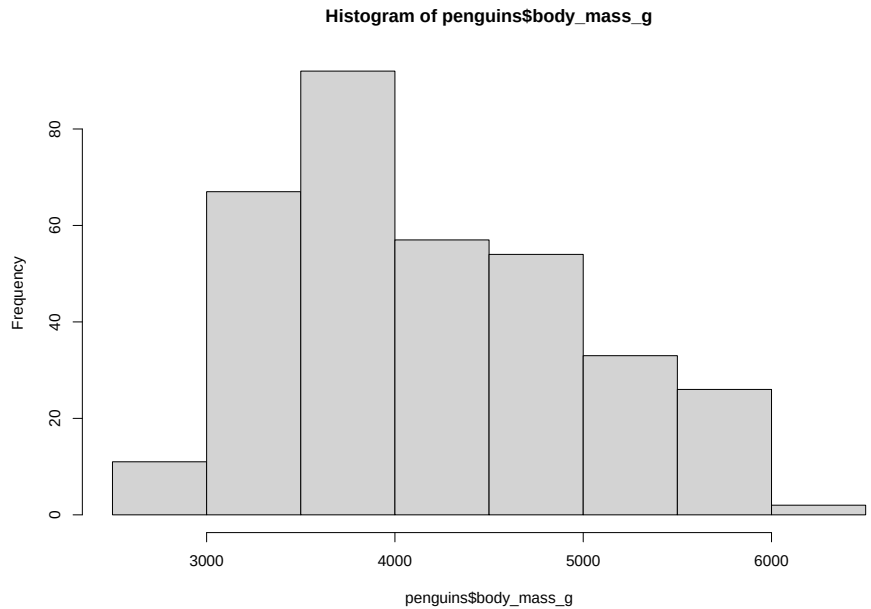


Easy Plots

Histogram

Graph that uses bars to display the distribution of numerical data

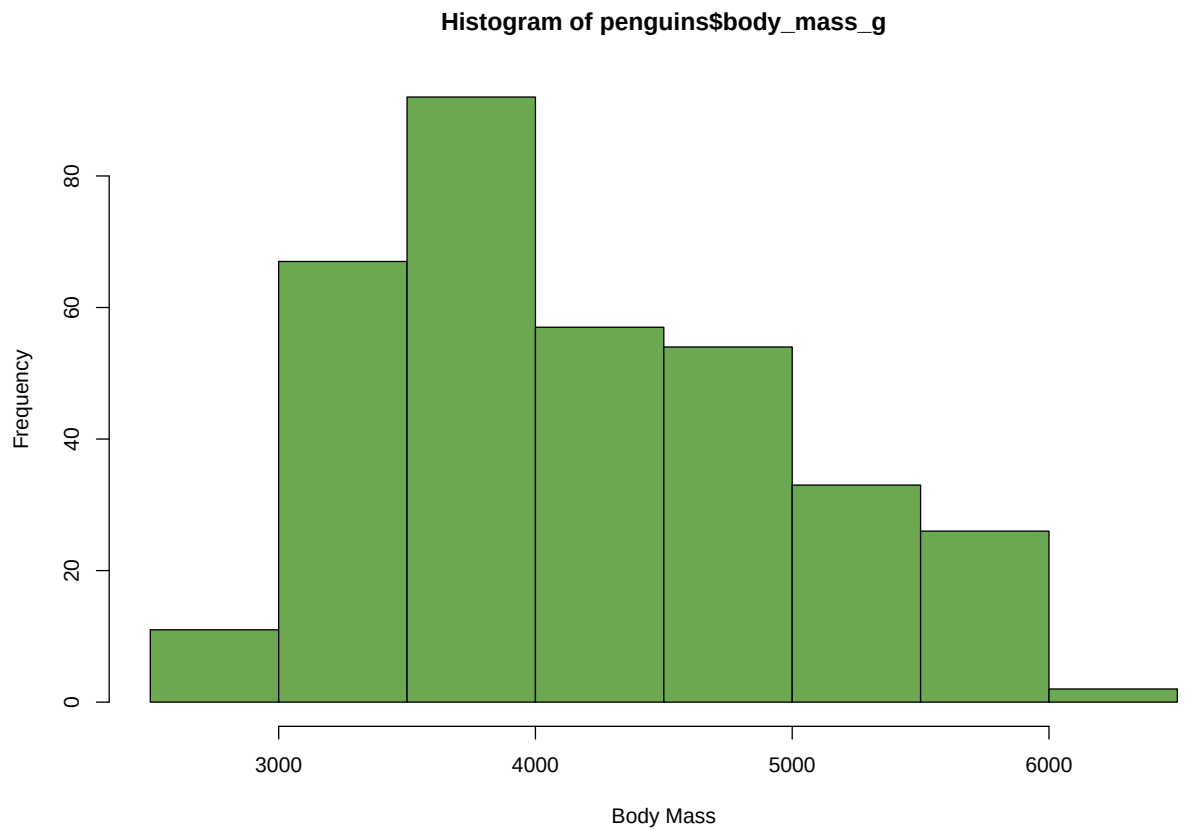
```
#Function
#hist()

#Histogram of body mass
hist(penguins$body_mass_g)
```



Easy Plots

Histogram Customization

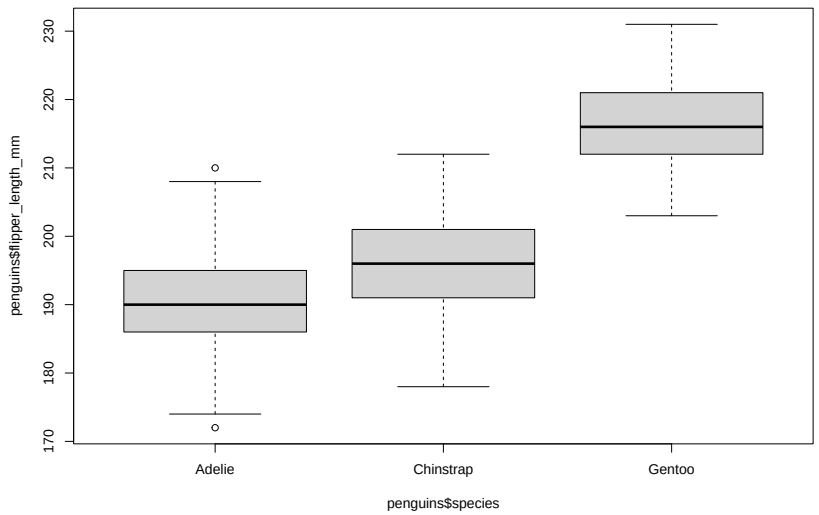
```
# Adding color and Labs
hist(penguins$body_mass_g, col= "#6ba851",
     xlab= "Body Mass", ylab="Frequency")
```



Easy Plots

Boxplot

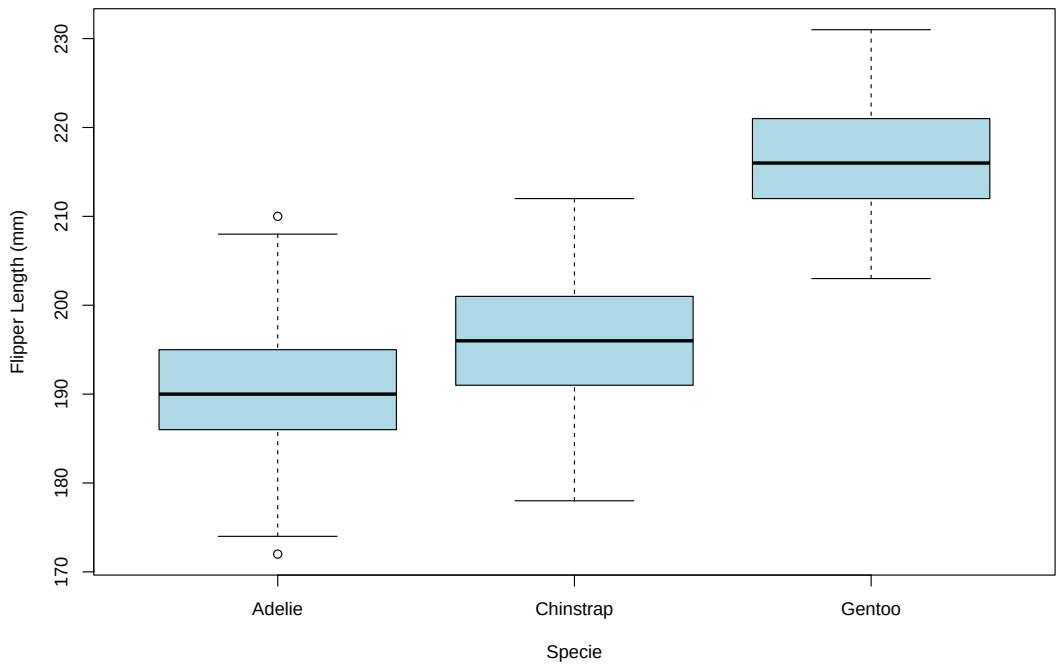
```
#Function
#boxplot()
#boxplot(formula, data = NULL, ...,)

#Plotting the flipper length in terms of the Species
boxplot(penguins$flipper_length_mm ~ penguins$species,
        penguins)
```



Easy Plots

Customizing Boxplots

```
#Adding color and labs
boxplot(penguins$flipper_length_mm ~ penguins$species,
        penguins, col= "lightblue",
        xlab="Specie",ylab= "Flipper Length (mm)")
```

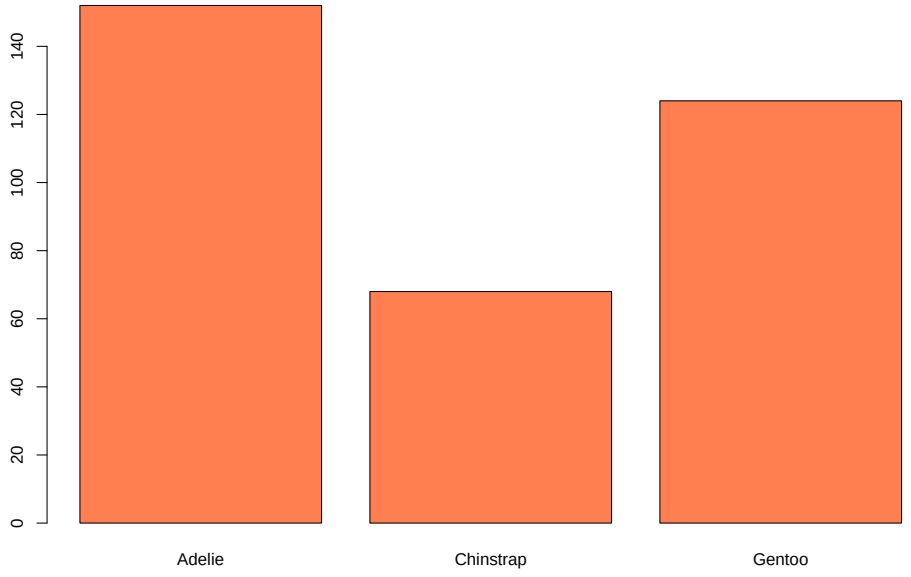


Easy Plots

Barchart

A barchart/barplot is used to display the relationship between a numeric and a categorical variable

```
#barplot()
freq=table(penguins$species)
barplot(freq, col="coral")
```



R colors

- HEX colors
- RBG format rgb() function

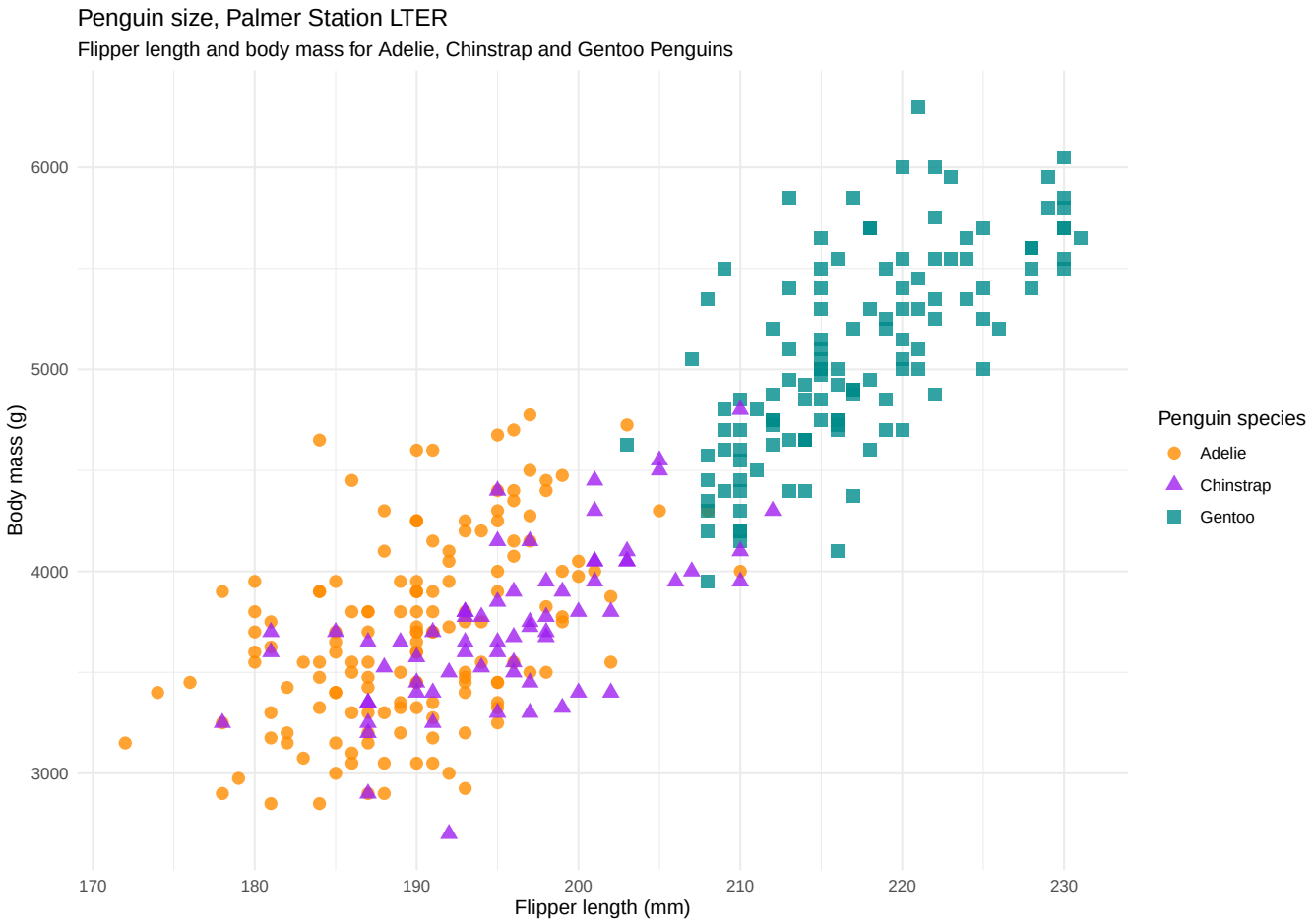
For color names:
http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

Google for hex codes!

Ggplot

For fancy plots we use *ggplot*. I highly encourage you to learn how to use it!

An example to what you can do:



Aknowledgements

Horst AM, Hill AP, Gorman KB (2020). palmerpenguins: Palmer Archipelago (Antarctica) **penguin data**. R package version 0.1.0. https://allisonhorst.github.io/palmerpenguins/. doi: 10.5281/zenodo.3960218

Special thanks to Bioinformatics Fridays at @LIIGH-UNAM !!