

# PSTAT 5LS Lab 1

TA Name Here

Week of September 30, 2024

# Section 1

## Learning Objectives

# R Learning Objectives

- ① Learn how to import data into R
- ② Learn how to find the five-number summary of a variable, and find a specific numeric summary (statistic) in R
- ③ Learn how to make a histogram in R
- ④ Learn how to make a box plot in R
- ⑤ Learn how to make side-by-side box plots in R

# Statistical Learning Objectives

- 1 Understand when to histogram
- 2 Understand when to make a box plot
- 3 Understand when to make a side-by-side box plot and how to use this type of comparison
- 4 Be able to use these graphical and numerical summaries to discuss data

# Functions covered in this lab

- ① `read.csv()`
- ② `head()`
- ③ `str()`
- ④ `summary()`
- ⑤ `hist()`
- ⑥ `min()`, `mean()`, `median()`, `max()`, `sd()`, `IQR()`
- ⑦ `boxplot()`

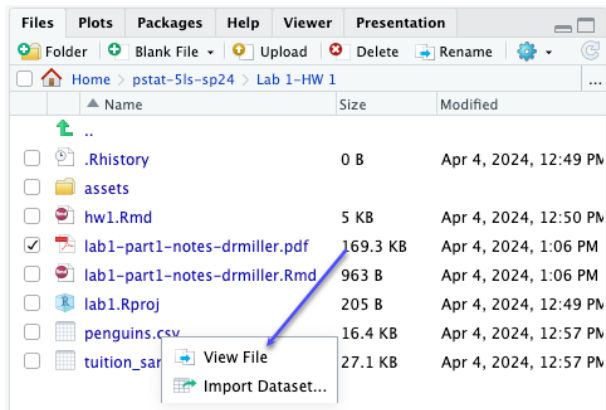
## Section 2

### Lab Tutorial

# CSV files: a common way to store data

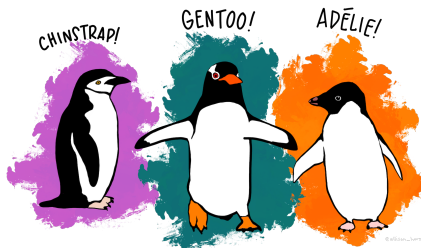
One common way to store data is to store it in a CSV file. CSV stands for “comma separated values”.

Open the file “penguins.csv” from the files pane (lower right) to see what a .csv file looks like:



# Palmer Penguins Data

We're going to start by working with a data set with data on 333 penguins collected from 3 islands in the Palmer Archipelago in Antarctica. Data were collected and made available by Dr. Kristen Gorman and the Palmer Station, Antarctica LTER, a member of the Long Term Ecological Research Network, and the data were prepared by Dr. Allison Horst.





# Reading Data into R

We can **read** data into R using a function called `read.csv()`. The first argument to `read.csv()` is the name of a `.csv` file (here, `penguins.csv`), in quotes. We then store the results of `read.csv()` as an object called `penguins`.

```
penguins <- read.csv("penguins.csv", stringsAsFactors = TRUE)
```

The argument `stringsAsFactors = TRUE` has been added to the `read.csv()` function. This is important because we want to be able to distinguish between something called a *string*, which is a phrase or word, and a **factor**, which represents the levels of a categorical variable. Basically, by setting `stringsAsFactors = TRUE`, we are letting R know to expect that any words or phrases in the data actually relate to categorical variables.

Go ahead and run the `loadPenguins` chunk of your `lab1-notes.Rmd` markdown file, and verify that the `penguins` data is in your environment in the top right corner of your RStudio project.

# Peeking at the Data

Let's see what's in the data. We can peek at the first few (6, specifically) rows of the data using the `head()` function:

```
head(penguins)
```

```
##   species      island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## 1  Adelie Torgersen      39.1          18.7           181           3750
## 2  Adelie Torgersen      39.5          17.4           186           3800
## 3  Adelie Torgersen      40.3          18.0           195           3250
## 4  Adelie Torgersen      36.7          19.3           193           3450
## 5  Adelie Torgersen      39.3          20.6           190           3650
## 6  Adelie Torgersen      38.9          17.8           181           3625
##      sex year
## 1   male 2007
## 2 female 2007
## 3 female 2007
## 4 female 2007
## 5   male 2007
## 6 female 2007
```

# Peeking at the Data

We read that line of code as “*head of penguins*”. Remember that `penguins` is what we named our data set. We can see that `penguins` contains a number of *variables*, like `species`, `island`, and more.

Use the `tryIt1` chunk in your notes to peek at the first 6 rows of the `penguins` data file. The function is `head(penguins)`.

# Data

Variable name	Description
species	Penguin species (Adélie, Chinstrap, and Gentoo)
island	Island in Palmer Archipelago, Antarctica, on which the penguin was observed (Biscoe, Dream, or Torgersen)
bill_length_mm	A number denoting bill length (in millimeters)
bill_depth_mm	A number denoting bill depth (in millimeters)

# Data Continued

Variable name	Description
flipper_length_mm	A whole number denoting flipper length (in millimeters)
body_mass_g	A whole number denoting penguin body mass (in grams)
sex	Penguin sex (female, male)
year	Study year (2007, 2008, 2009)

# Another Way

We can also peek at the data using a function called `str()` (pronounced “stir”, short for “structure”):

```
str(penguins)
```

```
## 'data.frame':   333 obs. of  8 variables:
## $ species      : Factor w/ 3 levels "Adelie","Chinstrap",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ island       : Factor w/ 3 levels "Biscoe","Dream",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ bill_length_mm : num  39.1 39.5 40.3 36.7 39.3 38.9 39.2 41.1 38.6 34.6 ...
## $ bill_depth_mm : num  18.7 17.4 18 19.3 20.6 17.8 19.6 17.6 21.2 21.1 ...
## $ flipper_length_mm: int  181 186 195 193 190 181 195 182 191 198 ...
## $ body_mass_g    : int  3750 3800 3250 3450 3650 3625 4675 3200 3800 4400 ...
## $ sex            : Factor w/ 2 levels "female","male": 2 1 1 1 2 1 2 1 2 2 ...
## $ year           : int   2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

Use the `tryIt2` chunk in your notes examine the structure of the penguins data file.

# How to Find Help in R

R has built-in “documentation” for every function. If you want to find that documentation, you can Google it, but that takes too long. So it’s better to use R’s built in help! In the R console, just type a question mark `?` followed by the name of the function you want help with, then hit enter.

For example, `?hist` will bring up the documentation for the `hist()` function.

Let’s try this in the `tryit3` code chunk in your notes file.

At the end of the help file you *may* find an example of how to use the function. These examples are generally super helpful! You can directly run them using the `example()` function – e.g., `example(hist)`.

The most useful feature of help in R is a list of a function’s arguments and a quick explanation of what each argument does. You may not be able to fully understand some of the terms in the documentation just yet, but try it out and your TA will be able to help!

# Numerical Summaries

Let's start with the `flipper_length_mm` variable. Is this a categorical or quantitative variable? How do you know?

We can use R to summarize data numerically. We'll use the `summary()` function to do that for a given variable. Here, we'll summarize the `flipper_length_mm` variable, which is the length of the penguins' flippers (in millimeters). Try this code out by running the `tryit4` code chunk in your `lab1-notes`.

```
summary(penguins$flipper_length_mm)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	172	190	197	201	213	231



# Numerical Summaries

You might have noticed that the `summary()` function doesn't give you the standard deviation of the variable. To get the standard deviation, use the `sd()` function in addition to the `summary()` function.

Summarize the `flipper_length_mm` variable again adding the code to get the standard deviation as well. Use the `tryit5` code chunk to have R give you the standard deviation in addition to the other summary statistics.

```
summary(penguins$flipper_length_mm)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      172     190     197     201     213     231
```

```
sd(penguins$flipper_length_mm)
```

```
## [1] 14.01577
```

## Number Summaries continued

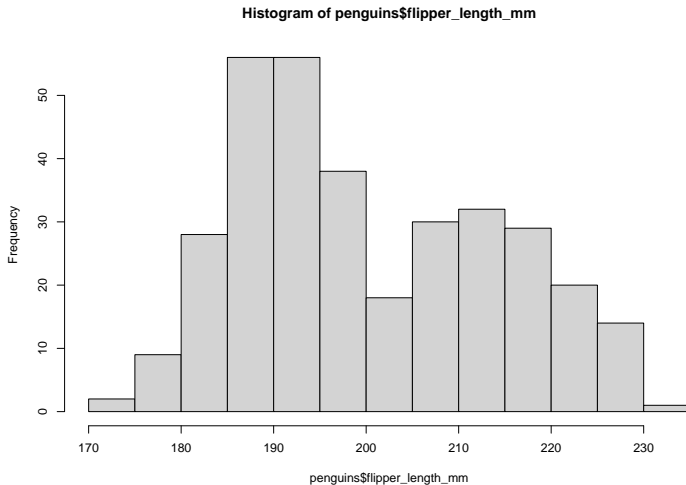
You can always get just the one numerical summary you're looking for using the function for that specific summary. Try these out in the `tryit6` code chunk.

```
min(penguins$flipper_length_mm)
mean(penguins$flipper_length_mm)
median(penguins$flipper_length_mm)
max(penguins$flipper_length_mm)
sd(penguins$flipper_length_mm)
IQR(penguins$flipper_length_mm)
```

# Histograms in R

One type of graphical display for a quantitative variable is a histogram. Histograms in R are also pretty easy – you just use the `hist()` function.

# Histogram of Flipper Length of Penguins



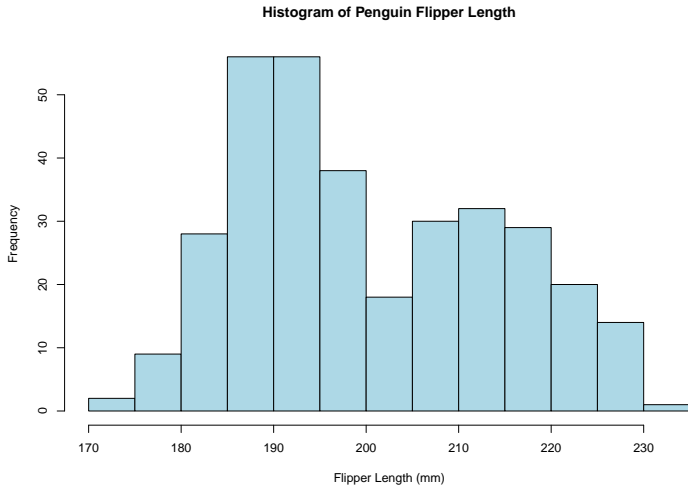
# Don't Forget Labels and Titles!

```
hist(penguins$flipper_length_mm)
```

So here we've got a histogram. Notice that we didn't provide the `main`, `xlab`, and `ylab` arguments that we'd normally use for a plot title and an axis label, but R still gave us a title and labels. This is nice, but the labels are *horrible*: nobody (other than you) knows what `penguin$flipper_length_mm` means, so we don't want to use that as a title or axis label.

The moral of the story is to **always provide `main`, `xlab`, and `ylab` arguments when making a plot!**

# That Same Histogram with Labels and a Title



# Histogram Code

Try this code out in the tryit7 code chunk. Watch out for the dreaded typos! If you get an error message, try to debug it yourself before asking for help!

```
hist(penguins$flipper_length_mm,  
     main = "Histogram of Penguin Flipper Length",  
     xlab = "Flipper Length (mm)",  
     col = "lightblue")
```

# Describing Histograms

Recall from lecture that we describe distributions by addressing four aspects:

- 1 Shape (number of modes + symmetry or lack thereof)
- 2 Center
- 3 Spread/Variability
- 4 Outliers

A handy mnemonic to remember what to comment about is **SOCS**:

**S**hape **O**utliers **C**enter **S**pread

**Note:** Be sure to mention whether there are or are not outliers. Not saying anything doesn't let us know that you know to check for outliers.



# Using Histograms to Describe Distributions

Describe the distribution of flipper lengths.

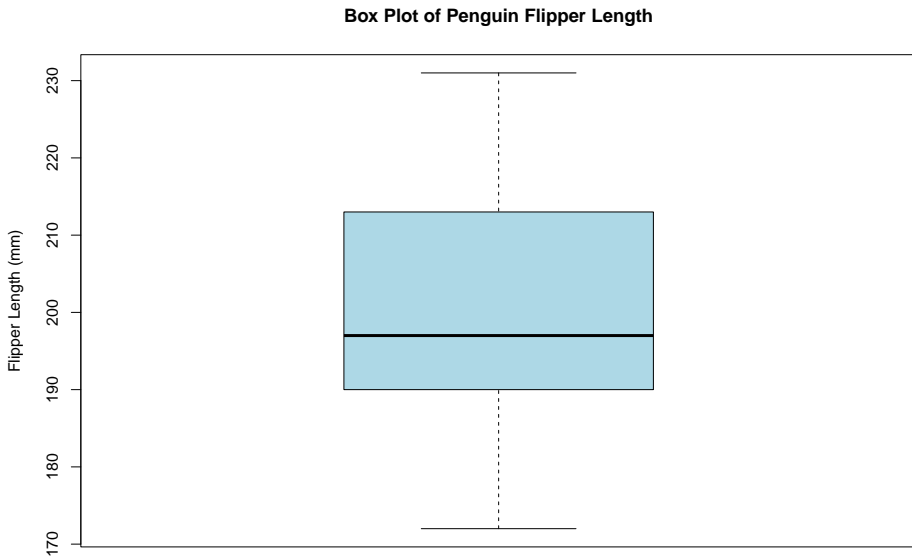
# Using Histograms to Describe Distributions

Do you think that the mean is the best measure of center for the flipper lengths? Why or why not?

# Box Plots in R

Another type of graphical display for a quantitative variable is a box plot. The command for making a box plot in R is pretty simple: it's just `boxplot()`. To make a box plot of a single variable, just give R the name of the data set, a dollar sign (\$), then the name of the variable. Also provide the arguments `main` and `ylab` for a plot title and y-axis label.

# Box Plot of the Flipper Length of Penguins



# Code for the Box Plot of the Flipper Length of Penguins

Try this code out yourself in the tryit8 code chunk. Watch out for the dreaded typos! If you get an error message, try to debug it yourself before asking for help!

```
boxplot(penguins$flipper_length_mm,  
        main = "Box Plot of Penguin Flipper Length",  
        ylab = "Flipper Length (mm)",  
        col = "lightblue")
```

# Describing Distributions with Box Plots

True or False:

The box plot of flipper lengths appears to be unimodal and symmetric.

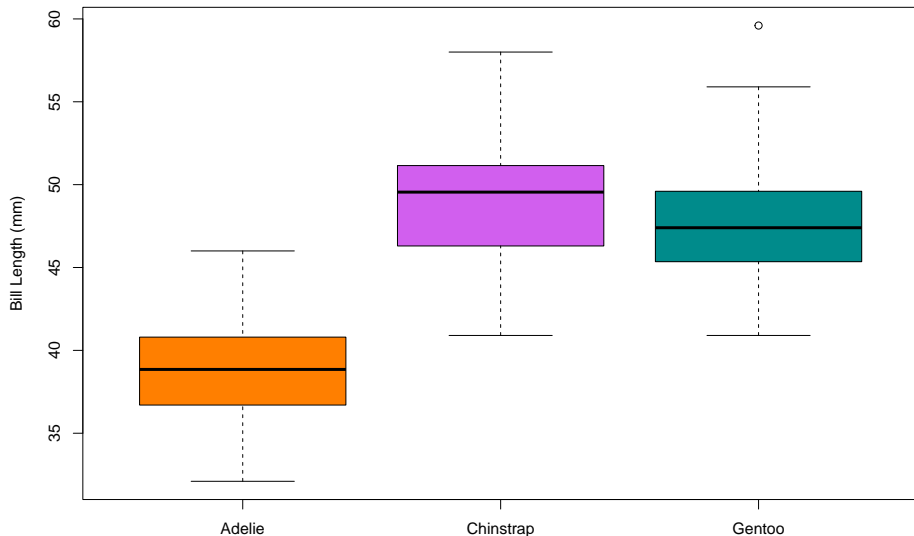
# Side-by-side Box Plots

Sometimes we're interested in comparing two or more groups using "side-by-side" box plots. We can compare the different species of penguins' bill lengths in this way, still using the `boxplot` function. We have provided the code in the `tryit9` code chunk.

```
boxplot(penguins$bill_length_mm ~ penguins$species,  
        main = "Box Plots of Penguin Bill Length by Species",  
        ylab = "Bill Length (mm)",  
        xlab = "Species",  
        col = c("darkorange1", "mediumorchid2", "darkcyan"))
```

# Side-by-side Box Plots Continued

Box Plots of Penguin Bill Length by Species





# Penguin Bill Length By Species

Does it appear that a penguin's bill length is related to its species, for the penguins in Palmer Archipelago? Why or why not?