

# Class 03

## DATA1220-55, Fall 2024

Sarah E. Grabinski

2024-09-04

# Load Packages for Today's Slides

```
# Contains the describe() function for comprehensive data s
library(Hmisc)
# Contains data sources used in our text book
library(openintro)
# Contains the palmer penguins dataset
library(palmerpenguins)
# For scatterplot matrices
library(GGally)
# Always load the tidyverse last
library(tidyverse)

# Set favorite ggplot2 theme for visualizations
theme_set(theme_bw())
```

# Homework is now due Monday

- ▶ Please take advantage of the extra time to attend office hours and/or post on Campuswire for help with remaining homework questions
- ▶ Late policy: “This homework is due by 6:00pm on Monday, 9/9/24. No credit will be lost for assignments received by 7:00pm to account for issues with uploading. 10% of the points will be deducted from assignments received by 9:00am on Tuesday, 9/10/24. Assignments turned in after this point are only eligible for 50% credit, so it benefits you to turn in whatever you have completed by the due date.”

# Let's talk about coding anxiety

- ▶ It's natural to be anxious about learning to code, but it has a bad reputation
- ▶ Older coding languages are less “readable” and required a lot of memorization
- ▶ Modern languages are more interpretable (i.e. function is named for what it does)
- ▶ Someone has probably answered the question you have somewhere on the internet

# What are your thoughts on ChatGPT?

It's a lying liar that lies. It will make up functions that don't exist, and troubleshooting its bugs has wasted countless of my hours. Using ChatGPT to write new code is risky at best. I do not recommend it.

That said, I have found ChatGPT occasionally useful for debugging code when I don't understand the error that is being generated. It is also useful for generating custom markdown templates. That has still led to dead ends and lost time though, so use at your own risk.

# My Approach to Coding

Don't waste time memorizing functions, package names, parameters, etc. Everything is just a quick Google search away. You should honestly be able to copy-paste a lot of your homework code from somewhere in the slides on that chapter. What you should focus on learning is how to recognize different types of data and which tools are best for analyzing that data. You won't need every tool in your toolkit all the time, but you should know how to find them when you need them.

# What is data science?

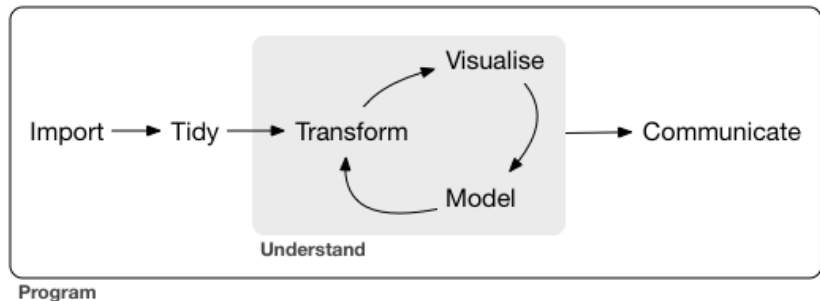


Figure 1: Data Science Pipeline

Source: Figure 1.1 in <https://r4ds.hadley.nz/intro.html>

# Chatfield's Six Rules for Data Analysis

1. *Do not attempt to analyze the data until you understand what is being measured and why.*
2. *Find out how the data were collected.*
3. *Look at the structure of the data.*
4. *Carefully examine the data in an exploratory way, before attempting a more sophisticated analysis.*
5. *Use your common sense at all times.*
6. *Report the results in a clear, self-explanatory way.*

*Chatfield, Chris (1996) Problem Solving: A Statistician's Guide, 2nd ed.*



# Chapter 1 Pipeline

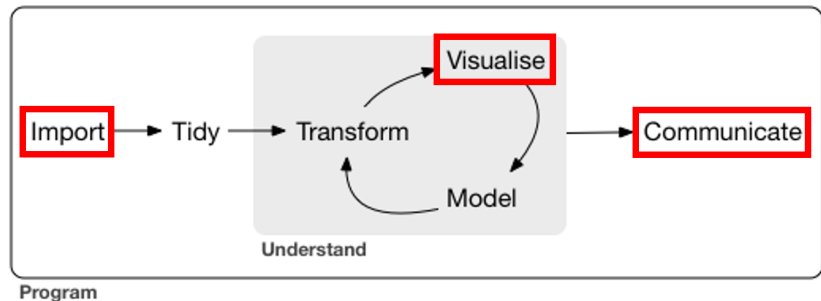


Figure 2: Data science pipeline priorities for Chapter 1

# Chapter 1 Objectives

- ▶ Get to know you better
- ▶ Set up R, RStudio, and Campuswire
- ▶ Describe how data was collected
  - ▶ Study, sample, and target populations
  - ▶ Sampling procedures and principles
- ▶ Identify what types of variables were measured
- ▶ Import and summarize raw data
- ▶ Create an exploratory visualization
- ▶ Communicate findings using a Quarto markdown document

# What we'll tackle today...

- ▶ Get to know you better
- ▶ Set up R, RStudio, and Campuswire
- ▶ Identify what types of variables were measured
- ▶ Import and summarize raw data
- ▶ Create an exploratory visualization
- ▶ Communicate findings using a Quarto markdown document

# What we'll tackle on Friday...

- ▶ Describe how data was collected
  - ▶ Study, sample, and target populations
  - ▶ Sampling procedures and principles
- ▶ Communicate findings using a Quarto markdown document (cont.)

# Introductory Survey

- ▶ DATA1220-55 Fall 2024 Intro Survey (link)
- ▶ “Getting to know you” exercise to help me serve you better
- ▶ Should take fewer than 10 minutes to complete
- ▶ 21 people have already responded – THANK YOU!
- ▶ Worth FIVE FREE POINTS on Homework 1

# Campuswire Forum

- ▶ Class Feed (link, bookmark this page!)
- ▶ Forum for homework issues, discussions, earning participation credit
- ▶ Point-based system for asking questions, crowdsourcing answers
- ▶ 22 people have completed registration – THANK YOU!
- ▶ Worth FIVE FREE POINTS on Homework 1

# Navigating Campuswire

The screenshot displays the Campuswire interface, which is organized into three main sections: a left sidebar, a central feed, and a right-hand detail panel.

**Left Sidebar:** Contains navigation icons and labels. At the top is a blue gear icon. Below it are 'Notifications' (with a red '99+' badge), 'DMs', 'Calendar', and 'Search'. A circular profile picture of Sarah is shown with a dropdown arrow. Below that is a blue 'Class feed' button. Further down are icons for '# Rooms', 'Files', and a profile picture of Sarah labeled 'Sarah Active'.

**Central Feed:** Titled 'Class feed' with a subtitle 'DATA1220-55: Elementary Statistics'. It includes a search bar, a category dropdown set to 'All categories', and a '+ New post' button. A list of recent posts follows:

- Updated Homework 1 Template** (#10): 'I've corrected some errors in the 'ho...' (0 likes, 39 minutes ago).
- Class 2 Slides PDF** (#8): 'A couple people had trouble downlo...' (0 likes, 2 hours ago).
- Does anyone understand the ho...** (#7): 'I've read through everything posted...' (2 likes, 3 hours ago).

A 'Last week' section is visible, featuring a highlighted post:

- Are you in??** (#5): 'Please comment here with your na...' (0 likes, 5 days ago).

At the bottom of the feed, it shows '2 online now' with profile pictures of two users.

**Right-hand Detail Panel:** Displays the details for the 'Are you in??' post. It shows the user 'Sarah Grabinski' who asked the question 5 days ago, with a 'Visible to: Everyone' setting. The post title is 'Are you in??' #5, categorized as 'Homework 1'. The question text is: 'Please comment here with your name, as you'd like others in class to use it, and your personal pronouns. For fun, please include a picture of your pet (or a pet you wish you had, if you don't currently have any). This will earn you your first participation points through Campuswire.' Engagement stats show 0 likes, 5 comments, 122 views, and 23 participants. A 'Resolved' status is indicated at the top right. Below the question, there are tabs for 'Answers' and 'Comments'. Under the 'Answers' tab, it shows '1' answer. The answer is by 'Caroline Cotter' (15 minutes ago), who answered 'Caroline Cotter she/her'. A blue 'Answer this question' button is located at the top right of the answer section.

Figure 3: The Campuswire main page, also called the Class Feed

# Making A New Post

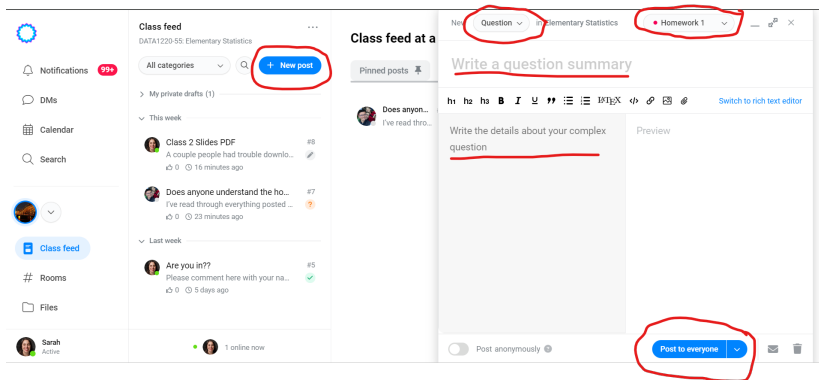


Figure 4: Post a question on Campuswire by selecting the blue “+ New post” button, creating a new ***\*\*question\*\**** using the drop-down menu, tagging the question topic, giving it a title and description, and posting. Be sure to post a new ***\*\*question\*\**** and not a note (default option) for full credit!



# Example Question

The screenshot displays the Campuswire interface. On the left is a sidebar with navigation options: Notifications (99+), DMs, Calendar, Search, a user profile for Sarah (idle), and a 'Class feed' button. The main content area is divided into two sections. The top section, titled 'Class feed' for 'DATA1220-55: Elementary Statistics', lists recent posts: 'Updated Homework 1 Template' (44 minutes ago), 'Class 2 Slides PDF' (3 hours ago), and a highlighted question 'Does anyone understand the ho...' (3 hours ago) with 7 answers. The bottom section shows 'Last week' posts, including 'Are you in??' (5 days ago). The right section shows a detailed view of the question 'Does anyone understand the homework?' asked by Ayden Dabney 31 minutes ago. The question text is 'I've read through everything posted on Canvas and I just am not understanding what I'm suppose to do with the template and how to work the RStudio.' It has 2 likes, 0 comments, 19 views, and 6 answers. The 'Answers' tab is selected, showing an 'Instructor answer' by Sarah Grabinski, who endorsed the answer and provided additional instructions on downloading R and RStudio. A blue 'Answer this question' button is visible at the bottom right of the question view.

**Class feed**  
DATA1220-55: Elementary Statistics

All categories

**Updated Homework 1 Template** #10  
I've corrected some errors in the 'ho...  
44 minutes ago

**Class 2 Slides PDF** #8  
A couple people had trouble downlo...  
3 hours ago

**Does anyone understand the ho...** #7  
I've read through everything posted ...  
3 hours ago

▼ Last week

**Are you in??** #5  
Please comment here with your na...  
5 days ago

2 online now

**Ayden Dabney** 31 asked a question 3 hours ago  
Visible to: Everyone

**Does anyone understand the homework?** #7

Homework 1

I've read through everything posted on Canvas and I just am not understanding what I'm suppose to do with the template and how to work the RStudio.

2 0 19 6

Answers Comments

**Instructor answers**

You endorsed this answer

**Sarah Grabinski** answered this question an hour ago

I've corrected a few errors in the template due to an update in R while I was away. Please redownload the files from Canvas. Here are some additional instructions.

- Make sure you have downloaded both R and RStudio. R is the language and needs to be installed first. RStudio is the interface, and requires an installation of R to run. (Class 1 slide 12)

Figure 5: A magnificent example of a brave student using Campuswire to crowdsource help on their homework

# Interacting With Questions

The screenshot displays the Canvas LMS interface. On the left is a sidebar with navigation links: Notifications (99+), DMs, Calendar, Search, a user profile dropdown, Class feed (highlighted), Rooms, and Files. The main content area is divided into two sections. The top section, 'Class feed', shows a list of posts: 'Updated Homework 1 Template' (#10), 'Class 2 Slides PDF' (#8), and a highlighted post 'Does anyone understand the homework?' (#7) by Ayden Dabney. The bottom section, 'Instructor answers', shows an answer by Sarah Grabinski. Red boxes highlight the 'Like' button (showing 2 likes) on the question post, the 'Answer this question' button, and the 'Upvote' button (showing 1 upvote) on the instructor's answer.

**Class feed**  
DATA1220-55: Elementary Statistics

All categories  + New post

**Updated Homework 1 Template** #10  
I've corrected some errors in the 'ho...  
👍 0 🕒 44 minutes ago

**Class 2 Slides PDF** #8  
A couple people had trouble downlo...  
👍 0 🕒 3 hours ago

**Does anyone understand the ho...** #7  
I've read through everything posted ...  
👍 2 🕒 3 hours ago

▼ Last week

**Are you in??** #5  
Please comment here with your na...  
👍 1 🕒 5 days ago

👤 2 online now

**Ayden Dabney** 31 asked a question 3 hours ago  
Visible to: Everyone

**Does anyone understand the homework?** #7  
Homework 1

I've read through everything posted on Canvas and I just am not understanding what I'm suppose to do with the template and how to work the RStudio.

👍 2 🗨 0 👁 19 👤 6

Answers Comments

**Answer this question**

**Instructor answers** ⚙ Best

👍 1 You endorsed this answer

**Sarah Grabinski** answered this question an hour ago

I've corrected a few errors in the template due to an update in R while I was away. Please redownload the files from Canvas. Here are some additional instructions.

- Make sure you have downloaded both R and RStudio. R is the language and needs to be installed first. RStudio is the interface and requires an installation of R to run. (Class 1 slide 19)

Figure 6: Interact with student-posed questions and discussion posts by liking the post, answering the question, or up-voting the answer(s) you think are best.


# Participation Points on Campuswire

The screenshot displays the Campuswire interface. On the left is a sidebar with navigation options: DMS, Calendar, Search, Class feed (highlighted), Rooms, Files, Grades, Settings, and Collapse. The main content area is divided into two sections. The top section, titled 'Class feed', shows a list of posts for 'DATA1220-55: Elementary Statistics'. The posts include 'Updated Homework 1 Template' (rank #10, 1 hour ago), 'Class 2 Slides PDF' (rank #8, 3 hours ago), 'Does anyone understand the ho...' (rank #7, 3 hours ago), and 'Are you in??' (rank #5, 5 days ago). The bottom section, titled 'Student answers', shows a post by 'Lauren Mezacapa' (rank 45, 5 days ago) with a photo of a black and white dog. The interface also shows a search bar at the top right and a 'Reply' button below the dog photo.

Figure 7: Participation scores will appear directly to the right of names with a bird icon indicating their level/status. Click on the icon to pull up how to earn participation points, the interactions needed to reach different ranks/levels, and your current participation status.

# Earning Participation Points

## Class reputation



In this class we'll be using the reputation system — Students will be rewarded for asking thoughtful questions, having engaging discussions and, most importantly, helping out other students by answering their questions

- Sarah Grabinski

Reputation levels

Reputation points

+2 POINTS	for each like you receive on a question posted on the class feed
+2 POINTS	for asking a question on the class feed
+5 POINTS	for each question you answer on the class feed
+10 POINTS	for each upvote you receive on your answers


Figure 8: The number of participation points (called “reputation points” on Campuswire) received for each type of interaction on the class feed.


# Participation Levels

## Class reputation

Reputation levels


Reputation points

 Level 0: N00b (you are here) Everyone starts out here

 Level 1: Starter ▼


☒ Answer 1 question on the class feed

☐ Receive 1 upvote from a classmate

 Level 2: Intermediate ▼

☐ Answer 5 question on the class feed

Figure 9: The first 3 participation levels, their corresponding icons, the interactions needed to reach each level, and your current progress.



# What's the difference between R and RStudio?

- ▶ R is an open source statistical programming language.
- ▶ R comes with it's own user interface called R Gui, but its functionality is limited.

 We do ***NOT*** want to use R Gui.

# What's the difference between R and RStudio?

RStudio is an integrated development environment (IDE) with a variety of tools for working with coding languages like R and Python.

- ▶ Smart code-highlighting for easy reading
- ▶ Direct and “chunkable” code execution
- ▶ Visualization capabilities
- ▶ Environment, workspace, and file management

# Downloading R v4.4.1

- ▶ Windows Installation of R-4.4.1
- ▶ macOS Installations
  - ▶ macOS 11 (**Big Sur**) and higher
  - ▶ Older, Intel Macs



This must be done before you can use RStudio.



# Installing RStudio Desktop

- ▶ Windows 10/11
- ▶ macOS 12+

**i** You may have to manually add an RStudio shortcut to your desktop.

How can I tell the difference?

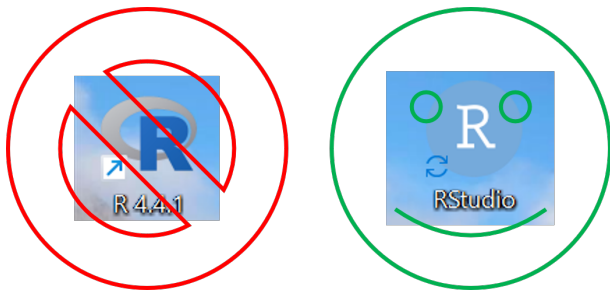


Figure 10: We want to work in RStudio.

# RStudio Files

We will work with 4 types of file in RStudio:

1. R scripts, ending in `.R`: text files containing only R code with no output
2. Quarto markdown documents, ending in `.qmd`: rich text files that combine R code with markdown language and YAML headers to format the document
3. HTML files, ending in `.html`: the rendered output of a Quarto markdown document that can be viewed in any standard web browser
4. PDF files, ending in `.pdf`: the rendered output of a Quarto markdown document that can be viewed in any standard PDF viewer

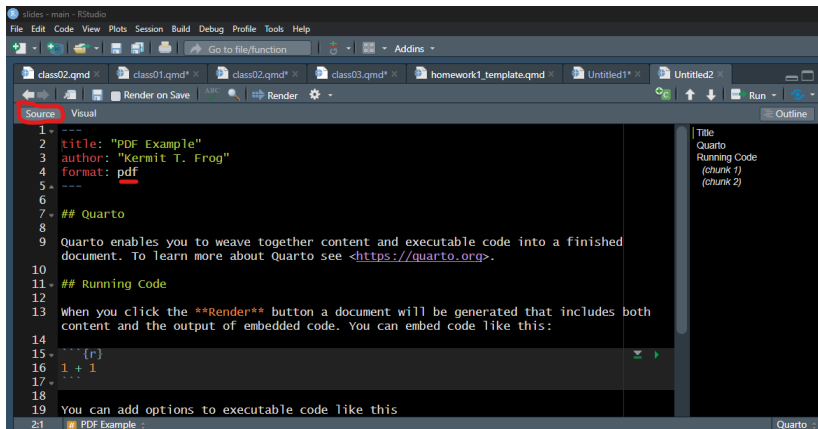
# Creating Raw Files in RStudio

Create raw text files by going to “File > New File” in RStudio and selecting...

- ▶ R Script for .R files
- ▶ Quarto Document for .qmd article-like documents
- ▶ Quarto Presentation for .qmd powerpoint-like presentations

# What does a raw .qmd file look like?

You can access the raw text, including the markdown language, by using the “Source” editor option.



```
1 ---
2 title: "PDF Example"
3 author: "Kermit T. Frog"
4 format: pdf
5 ---
6
7 ## Quarto
8
9 Quarto enables you to weave together content and executable code into a finished
10 document. To learn more about Quarto see <https://quarto.org>.
11
12 ## Running Code
13
14 When you click the Render button a document will be generated that includes both
15 content and the output of embedded code. You can embed code like this:
16
17 {r}
18 1 + 1
19
20 You can add options to executable code like this
```

2:1 PDF Example

Quarto  
Quarto  
Running Code  
(chunk 1)  
(chunk 2)

Figure 11: The “Source” editor shows the raw code and markup language without any preprocessing by RStudio. The file type is set as “PDF” in the YAML header at the top.

# Visual Editor for Raw Files

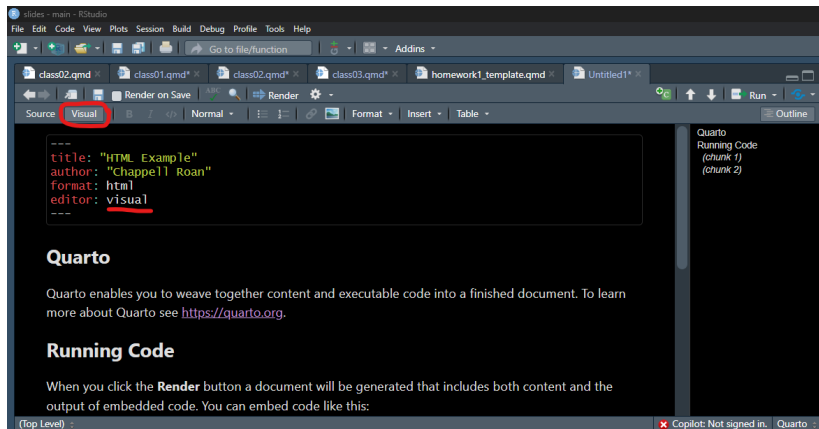


Figure 12: How to toggle between the “Source” and “Visual” editors in RStudio using the editor pane toolbar, along with how to set the default editor to “Visual” in the YAML header. The document type (“HTML”) is also specified in the YAML header.

! We want to live in the visual editor for now!

# Advantages of the Visual Editor

- ▶ Format text, insert links/images, create tables, and more from the toolbar (**no coding necessary!**)
- ▶ Preview document formatting without waiting for it to render over and over
- ▶ Insert executable cells (“code chunks”) and other special features directly into the document

# Creating Rendered Files in RStudio

Create rendered output files by (1) defining the document type in the YAML header and (2) running the “Render” process with the blue arrow icon at the top of the editor pane.

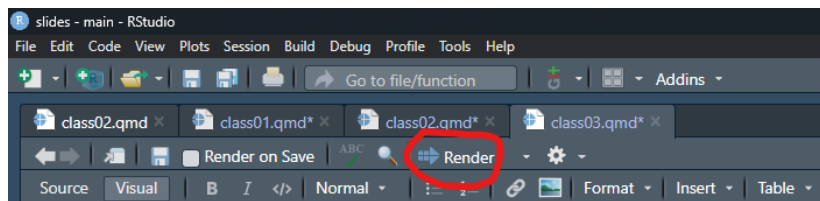


Figure 13: Run the “Render” process with the blue arrow icon to generate HTML and PDF files from Quarto markdown documents.

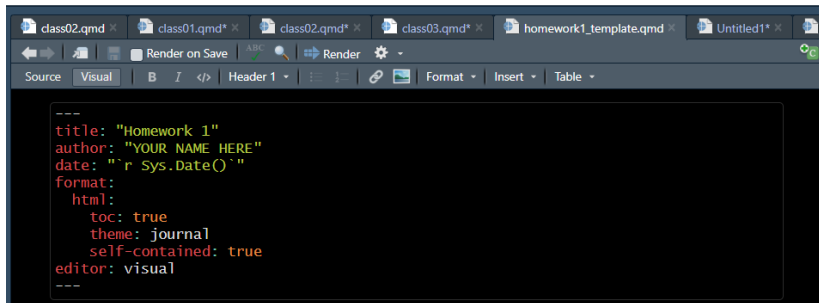


# WTH is a YAML?

- ▶ A text header bound by 3 dashes (---) at the top and bottom of a `.qmd` file
- ▶ Composed of key-value pairs, using the syntax `key: value` to define parameters
- ▶ Defines the document type, formatting, default options, and other *metadata* for your project

**i** I will write most of these for you in your homework templates.

# Example YAML Header from Homework 1

A screenshot of a web browser window displaying the Canvas LMS interface. The browser's address bar shows the URL 'https://canvas.libraries.psu.edu/'. The page title is 'Canvas LMS'. The main content area displays a file named 'homework1\_template.qmd' with a YAML header. The header is enclosed in a dark gray box with a light gray border. The YAML text is as follows:

```
---
title: "Homework 1"
author: "YOUR NAME HERE"
date: "`r Sys.Date()`"
format:
  html:
    toc: true
    theme: journal
    self-contained: true
editor: visual
---
```

Figure 14: The YAML header from the homework1\_template.qmd file in the Chapter 1 module on Canvas. It establishes the document metadata (title, author, date, etc.), defines the document type (HTML), and sets the default editor to “Visual”.

# Using Projects in RStudio

“Projects” in RStudio, files ending in `.Rproj`, allow you to divide your work into discrete containers for each project, keeping them separated with their own unique...

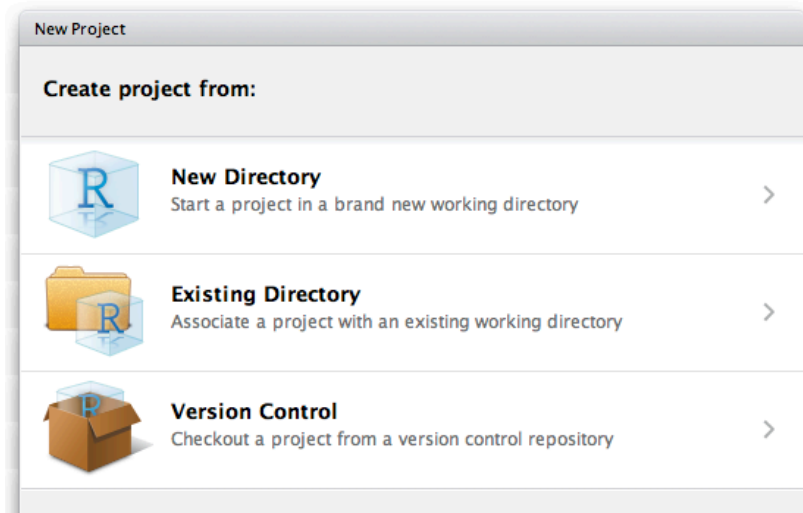
- ▶ Folder containing associated files (called the “working directory”)
- ▶ Data environment storing loaded packages, variables, and calculations (`.RData` files)
- ▶ Work history containing executed code (`.Rhistory` files)

## And the best parts?

- ▶ Projects will autosave your open documents so you have less to recover in the event of a crash
- ▶ Projects will store data and results objects in its “Global Environment” between sessions, so you don’t have to run the same calculations over and over each time
- ▶ Projects allow you to work on 2+ projects simultaneously across multiple RStudio sessions

# Creating a Project in RStudio

- ▶ File > New Project > New or Existing Directory
- ▶ RStudio Projects Tutorial



# Installing Code Packages

- ▶ Packages are collections of custom functions to use for statistical analyses
- ▶ Some are installed with base R, and some need to be installed manually
  - ▶ Install and update using the “Packages” tab in RStudio
  - ▶ Install in the console using the `install.packages('package_name')` function
- ▶ Packages are loaded into documents before any other code is written, using the `library('package_name')` function

## Coding Notes - Assignment Operator

You can declare a variable in R one of two ways: using an = or using the assignment operator <-. I prefer the latter, and that's what you'll see most often in my code. Both are acceptable.

```
# Example: defining a variable with =  
x = 5  
# Display the variable by declaring it  
x
```

```
[1] 5
```

```
# Example: defining a variable with <=  
y <- 27  
# Display the variable with the print() function  
print(y)
```

```
[1] 27
```

## Coding Notes - Pipe Operator

Create long chains of functions using the pipe operator `|>` to pass the results of one function as new input for the next function. We won't do much with this until you begin writing more complex code.

```
# Load the palmer penguins dataset and pipe on
penguins |>
  # Calculate a new variable and pipe on
  mutate(bill_length_cm = bill_length_mm / 10) |>
  # Rename a variable and pipe on
  rename(gender = sex) |>
  # Display specific variables from final result
  select(species, bill_length_cm, gender)
```



## Coding Notes - Pipe Operator

```
# A tibble: 344 x 3
  species bill_length_cm gender
  <fct>         <dbl> <fct>
1 Adelie         3.91 male
2 Adelie         3.95 female
3 Adelie         4.03 female
4 Adelie         NA    <NA>
5 Adelie         3.67 female
6 Adelie         3.93 male
7 Adelie         3.89 female
8 Adelie         3.92 male
9 Adelie         3.41 <NA>
10 Adelie        4.2   <NA>
# i 334 more rows
```

# Common Raw Data Formats

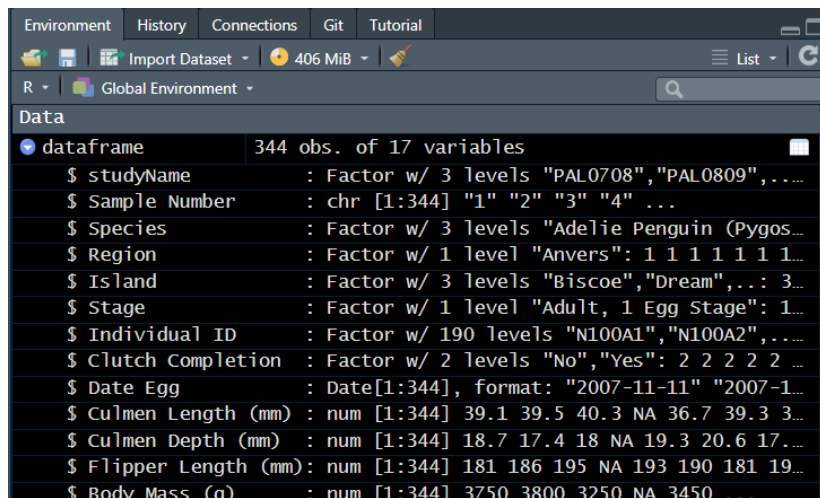
- ▶ Preformatted data sets in packages (e.g. openintro package from our textbook)
- ▶ R data files: `.rds` or `.rda` for single data objects, `.RData` for multiple data objects
- ▶ Delimited files: text files with character separators
  - ▶ Comma separated values (`.csv`)
  - ▶ Tab-separated values (`.tsv`)
- ▶ Excel spreadsheets (e.g. `.xls`, `.xlsx`)

# How do I get data files into R?

I will demonstrate the code for how to import common raw data formats as we work through examples in the course, rather than show them to you all at once.

- ▶ See Class 02, slides 17-19 for an example loading the palmerpenguins package data
- ▶ The Homework 1 template includes code to import a comma-separated values file (.csv)

# Once I import data, where does it go?



The screenshot shows the RStudio interface. The top menu bar includes 'Environment', 'History', 'Connections', 'Git', and 'Tutorial'. Below the menu bar, there's a toolbar with icons for file operations and a status bar showing 'Import Dataset', '406 MiB', and a search icon. The 'Environment' pane is active, showing the 'Global Environment'. A dataframe is loaded, containing 344 observations of 17 variables. The variables are listed with their data types and sample values.

dataframe	344 obs. of 17 variables
\$ studyName	: Factor w/ 3 levels "PAL0708","PAL0809",...
\$ Sample Number	: chr [1:344] "1" "2" "3" "4" ...
\$ Species	: Factor w/ 3 levels "Adelie Penguin (Pygos...
\$ Region	: Factor w/ 1 level "Anvers": 1 1 1 1 1 1 1...
\$ Island	: Factor w/ 3 levels "Biscoe","Dream",...: 3...
\$ Stage	: Factor w/ 1 level "Adult, 1 Egg Stage": 1...
\$ Individual ID	: Factor w/ 190 levels "N100A1","N100A2",...
\$ Clutch Completion	: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 ...
\$ Date Egg	: Date[1:344], format: "2007-11-11" "2007-1...
\$ Culmen Length (mm)	: num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 3...
\$ Culmen Depth (mm)	: num [1:344] 18.7 17.4 18 NA 19.3 20.6 17...
\$ Flipper Length (mm)	: num [1:344] 181 186 195 NA 193 190 181 19...
\$ Body Mass (g)	: num [1:344] 3750 3800 3250 NA 3450 ...

Figure 16: Imported data, user-defined variables, and calculated results will be stored in your project's “**Global Environment**”.

## What does data look like: Lists

```
# cbind-type list
```

```
c('object1', 'object2', 'object3')
```

```
[1] "object1" "object2" "object3"
```

```
# list-type list
```

```
list(1, 2, 3, 4, 5)
```

```
[[1]]
```

```
[1] 1
```

```
[[2]]
```

```
[1] 2
```

```
[[3]]
```

```
[1] 3
```

```
[[4]]
```

```
[1] 4
```

# What does data look like: Named lists

```
# named list  
list(name = 'Sabrina', age = 25,  
      major = 'Data Science', grad_year = 2026)
```

```
$name
```

```
[1] "Sabrina"
```

```
$age
```

```
[1] 25
```

```
$major
```

```
[1] "Data Science"
```

```
$grad_year
```

```
[1] 2026
```

## What does data look like: Matrices

```
# Input = a list of items
# Parameters = # of rows and/or columns
# Initialize cells by row = TRUE
matrix(data = seq(1:25), nrow = 5,
       ncol = 5, byrow = T)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	2	3	4	5
[2,]	6	7	8	9	10
[3,]	11	12	13	14	15
[4,]	16	17	18	19	20
[5,]	21	22	23	24	25

```
# Initialize cells by row = TRUE
matrix(data = seq(1:25), nrow = 5,
       ncol = 5, byrow = F)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	6	11	16	21
[2,]	2	7	12	17	22
[3,]	3	8	13	18	23
[4,]	4	9	14	19	24
[5,]	5	10	15	20	25

## What does data look like: Dataframes

```
# seq creates a list of numbers from low:high
# as.character changes variable type from numeric
data.frame(id = as.character(seq(1:5)),
            month = c('June', 'June', 'June',
                      'July', 'August'),
            # in rep, the 2nd parameter is the number of
            # times the 1st parameter is repeated in a list
            year = rep(2024, 5),
            state = c('OH', 'OH', 'IN', 'IN', 'CA'),
            # converts string variables to factors
            stringsAsFactors = T)
```

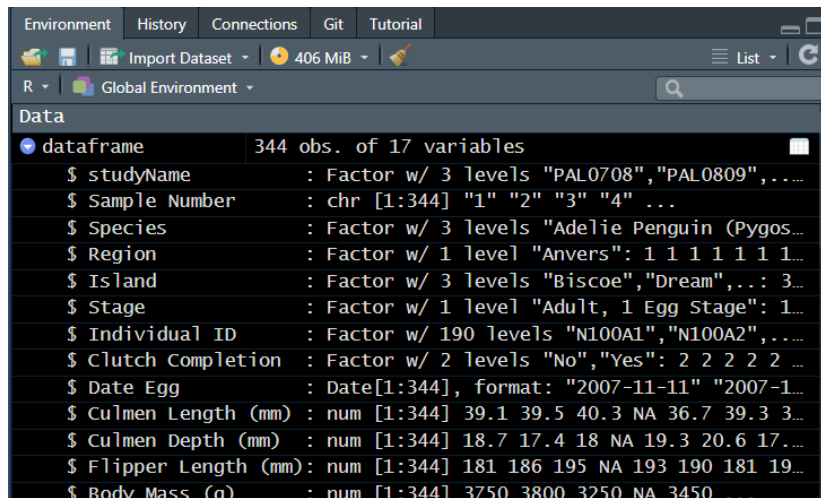
	id	month	year	state
1	1	June	2024	OH
2	2	June	2024	OH
3	3	June	2024	IN
4	4	July	2024	IN
5	5	August	2024	CA



# What is a dataframe?

- ▶ Data arranged in rows and columns like a typical spreadsheet
- ▶ Each row (ideally) contains 1 unique observation of the data for each of the measured variables
- ▶ Each column (ideally) contains all the observations for 1 unique variable that was measured

# What does a dataframe look like?



The screenshot shows the RStudio environment with the 'Data' pane selected. It displays a dataframe named 'dataframe' with 344 observations and 17 variables. A blue drop-down arrow next to the dataframe name allows for expanding the view to see variable details. The variables listed include studyName, Sample Number, Species, Region, Island, Stage, Individual ID, Clutch Completion, Date Egg, Culmen Length (mm), Culmen Depth (mm), Flipper Length (mm), and Body Mass (g), each with its data type and a preview of values.

dataframe	344 obs. of 17 variables
\$ studyName	: Factor w/ 3 levels "PAL0708","PAL0809",....
\$ Sample Number	: chr [1:344] "1" "2" "3" "4" ...
\$ Species	: Factor w/ 3 levels "Adelie Penguin (Pygos...
\$ Region	: Factor w/ 1 level "Anvers": 1 1 1 1 1 1 1...
\$ Island	: Factor w/ 3 levels "Biscoe","Dream",...: 3...
\$ Stage	: Factor w/ 1 level "Adult, 1 Egg Stage": 1...
\$ Individual ID	: Factor w/ 190 levels "N100A1","N100A2",....
\$ Clutch Completion	: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 ...
\$ Date Egg	: Date[1:344], format: "2007-11-11" "2007-1...
\$ Culmen Length (mm)	: num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 3...
\$ Culmen Depth (mm)	: num [1:344] 18.7 17.4 18 NA 19.3 20.6 17....
\$ Flipper Length (mm)	: num [1:344] 181 186 195 NA 193 190 181 19...
\$ Body Mass (g)	: num [1:344] 3750 3800 3250 NA 3450 ...

Figure 17: The compact view shows the name (dataframe), the number of rows (obs.), and the number of columns (variables). A preview of the variable names and data types can be accessed with the blue drop-down.

# What's a data dictionary?

A formatted document, often a table, which provides information about the variables such as...

- ▶ The variable names as seen in the raw data
- ▶ Description of the variable measured
- ▶ Units in which the variable was measured
- ▶ Number of observations
- ▶ Number of missing values

# Types of Data

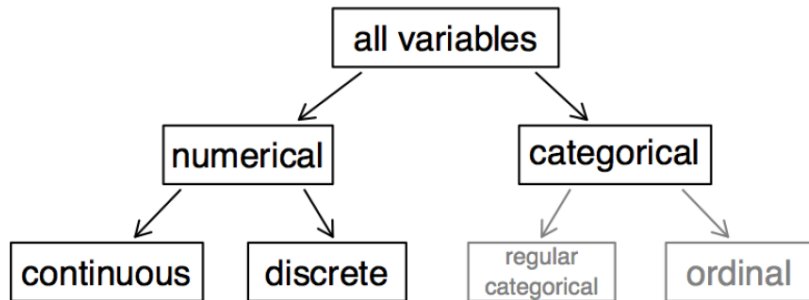


Figure 18: The two primary types of data we'll analyze are numerical and categorical variables.

# Numerical vs Categorical Data

- ▶ **Numerical** data is quantitative (measured) data.
- ▶ **Categorical** data is qualitative (descriptive) data.
  - ▶ **Binary** categorical variables only have 2 categories (e.g. 1 or 0)
  - ▶ **Multi-categorical** variables have 3+ categories (e.g.

# Numerical Variables: Continuous vs Discrete

- ▶ ***Continuous*** numeric variables can take any value imaginable within a given range
  - ▶ Examples: degrees Celsius, weight in grams, time elapsed in milliseconds
- ▶ ***Discrete*** numeric variables have a limited set of potential values
  - ▶ Examples: counts, time in months

# Categorical Variables: Nominal vs Ordinal

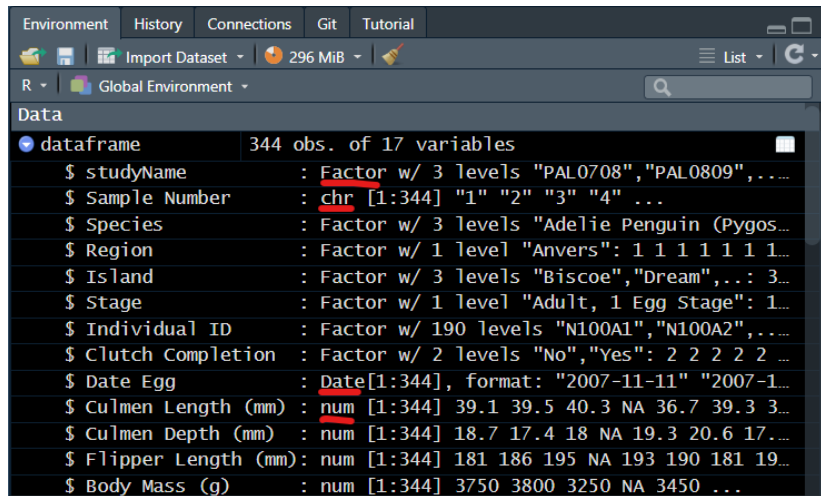
- ▶ **Nominal** categorical variables have no order
  - ▶ Rearranging the categories makes no difference
- ▶ **Ordinal** categorical variables have a direction
  - ▶ The order of the categories has meaning
  - ▶ Example: On a scale from 1-5..., Strongly Agree to Strongly Disagree

# How can I tell what kind of variable I have?

- ▶ Inspect the data in the global environment
- ▶ Print the data to the console or in a code chunk
- ▶ Use the `glimpse()` function for a quick summary
- ▶ Use the `describe()` function from the `Hmisc` package for a detailed summary



# Inspect the data in the global environment



The screenshot shows the RStudio interface with the 'Environment' pane selected. The 'Global Environment' is active, and a search bar is visible. The 'Data' section lists the 'dataframe' object, which contains 344 observations of 17 variables. The variables and their data types are as follows:

Variable	Data Type	Values (Sample)
studyName	Factor w/ 3 levels	"PAL0708", "PAL0809", ...
Sample Number	chr [1:344]	"1" "2" "3" "4" ...
Species	Factor w/ 3 levels	"Adelie Penguin (Pygos..."
Region	Factor w/ 1 level	"Anvers": 1 1 1 1 1 1 1...
Island	Factor w/ 3 levels	"Biscoe", "Dream", ...: 3...
Stage	Factor w/ 1 level	"Adult, 1 Egg Stage": 1...
Individual ID	Factor w/ 190 levels	"N100A1", "N100A2", ...
Clutch Completion	Factor w/ 2 levels	"No", "Yes": 2 2 2 2 2 ...
Date Egg	Date[1:344]	format: "2007-11-11" "2007-1..."
Culmen Length (mm)	num [1:344]	39.1 39.5 40.3 NA 36.7 39.3 3...
Culmen Depth (mm)	num [1:344]	18.7 17.4 18 NA 19.3 20.6 17....
Flipper Length (mm)	num [1:344]	181 186 195 NA 193 190 181 19...
Body Mass (g)	num [1:344]	3750 3800 3250 NA 3450 ...

Figure 19: Inspecting the object in the environment can provide details about what data types you have.

## Print the data in a code chunk

```
# the head command will print the first 5
# items in a list or rows in a dataframe
head(penguins_raw)
```

```
# A tibble: 6 x 17
  studyName `Sample Number` Species      Region Island
  <chr>          <dbl> <chr>      <chr> <chr>
1 PAL0708          1 Adelie Penguin ~ Anvers Torge~
2 PAL0708          2 Adelie Penguin ~ Anvers Torge~
3 PAL0708          3 Adelie Penguin ~ Anvers Torge~
4 PAL0708          4 Adelie Penguin ~ Anvers Torge~
5 PAL0708          5 Adelie Penguin ~ Anvers Torge~
6 PAL0708          6 Adelie Penguin ~ Anvers Torge~
# i 10 more variables: `Clutch Completion` <chr>, `Date Egg` <chr>,
#   `Culmen Length (mm)` <dbl>, `Culmen Depth (mm)` <dbl>,
#   `Flipper Length (mm)` <dbl>, `Body Mass (g)` <dbl>, Sex <chr>,
#   `Delta 15 N (o/oo)` <dbl>, `Delta 13 C (o/oo)` <dbl>, C
```

## Use glimpse() for a quick summary

```
glimpse(penguins_raw)
```

Rows: 344

Columns: 17

\$ studyName	<chr>	"PAL0708", "PAL0708", "PAL0708"
\$ `Sample Number`	<dbl>	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
\$ Species	<chr>	"Adelie Penguin (Pygoscelis a"
\$ Region	<chr>	"Anvers", "Anvers", "Anvers"
\$ Island	<chr>	"Torgersen", "Torgersen", "To"
\$ Stage	<chr>	"Adult, 1 Egg Stage", "Adult"
\$ `Individual ID`	<chr>	"N1A1", "N1A2", "N2A1", "N2A2"
\$ `Clutch Completion`	<chr>	"Yes", "Yes", "Yes", "Yes", "
\$ `Date Egg`	<date>	2007-11-11, 2007-11-11, 2007
\$ `Culmen Length (mm)`	<dbl>	39.1, 39.5, 40.3, NA, 36.7, 3
\$ `Culmen Depth (mm)`	<dbl>	18.7, 17.4, 18.0, NA, 19.3, 2
\$ `Flipper Length (mm)`	<dbl>	181, 186, 195, NA, 193, 190,
\$ `Body Mass (g)`	<dbl>	3750, 3800, 3250, NA, 3450, 3
\$ Sex	<chr>	"MALE", "FEMALE", "FEMALE", M
\$ `Delta 15 N (o/oo)`	<dbl>	NA, 8.04256, 8.26881, NA, 8

## Use `Hmisc::describe()` for a detailed summary

```
# describe is a common function name, so  
# it is a good habit to call this version  
# directly from the package using package_name::  
# to prevent conflicts and errors  
Hmisc::describe(penguins)
```

penguins

8 Variables      344 Observations

---

species

n	missing	distinct
344	0	3

Value	Adelie	Chinstrap	Gentoo
Frequency	152	68	124
Proportion	0.442	0.198	0.360

---

island

# Variable Terms

- ▶ ***Independent*** or ***explanatory*** variable
  - ▶ Typically on the x-axis
  - ▶ “Cause” variable
- ▶ ***Dependent*** or ***response*** variable
  - ▶ Typically on the y-axis
  - ▶ “Effect” variable

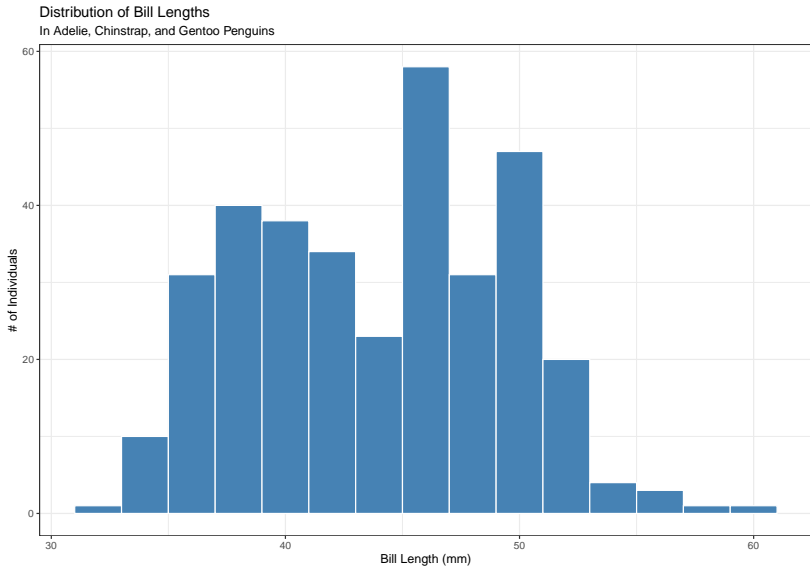
# Visualizing Data

- ▶ Distributions and numerical summaries of both explanatory and response variables
  - ▶ Histogram, bar plot
  - ▶ Density or violin plots
  - ▶ Boxplots
- ▶ Associations, relationships, correlations between explanatory and response variables
  - ▶ Scatter plots, regression
  - ▶ Scatterplot matrices

## Histogram - How common are certain ranges of values? (Discrete)

```
# Pipe data into ggplot2
penguins |>
  # Initialize the plot parameters with aes
  ggplot(aes(x = bill_length_mm)) + # ggplot2 only uses +!
  # add a histogram to the plot
  geom_histogram(binwidth = 2, # each bin spans 2 mm
                 fill = 'steelblue', # some color for fun
                 color = 'white') +
  # Add titles and axis labels
  labs(title = 'Distribution of Bill Lengths',
       subtitle = 'In Adelie, Chinstrap, and Gentoo Penguins',
       x = 'Bill Length (mm)',
       y = '# of Individuals')
```

# Histogram - How common are certain ranges of values? (Discrete)

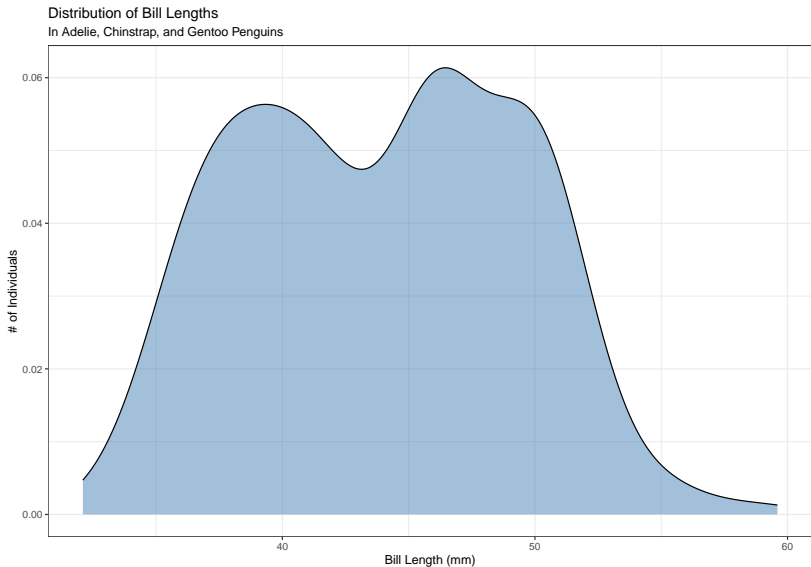




## Density Plot - How common are certain ranges of values? (Continuous)

```
# Pipe data into ggplot2
penguins |>
  # Initialize the plot parameters with aes
  ggplot(aes(x = bill_length_mm)) +
  # add a density curve to the plot
  geom_density(fill = 'steelblue', # add some color and make it semi-transparent
               alpha = 0.5) +
  # Add titles and axis labels
  labs(title = 'Distribution of Bill Lengths',
       subtitle = 'In Adelie, Chinstrap, and Gentoo Penguins',
       x = 'Bill Length (mm)',
       y = '# of Individuals')
```

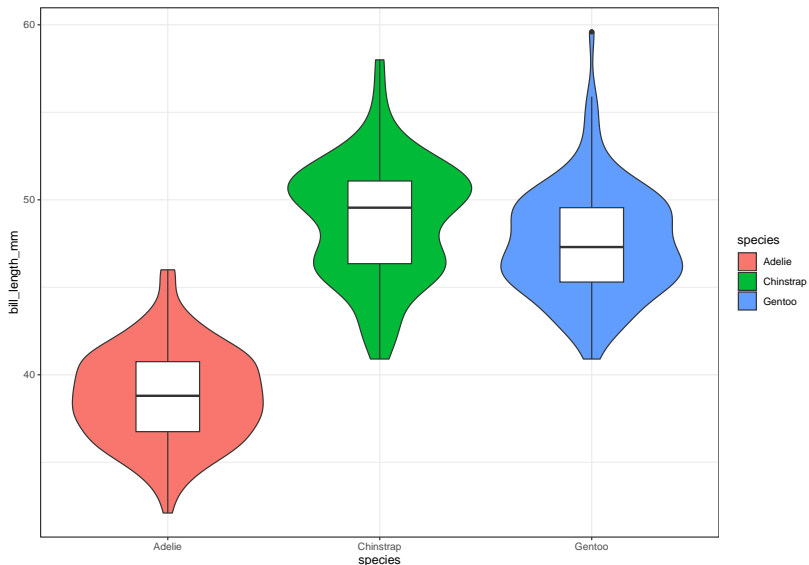
# Density Plot - How common are certain ranges of values? (Continuous)



## Boxplot + Violin - Numerical summary + density curves

```
# Pipe data into ggplot2
penguins |>
  # Initialize the plot parameters with aes
  ggplot(aes(x = species, y = bill_length_mm)) +
  # Add a violin plot as the base layer
  geom_violin(aes(fill = species)) +
  # Add a boxplot on top of the violin plot
  geom_boxplot(width = 0.3)
```

# Boxplot + Violin - Numerical summary + density curves



# Association vs. Independence

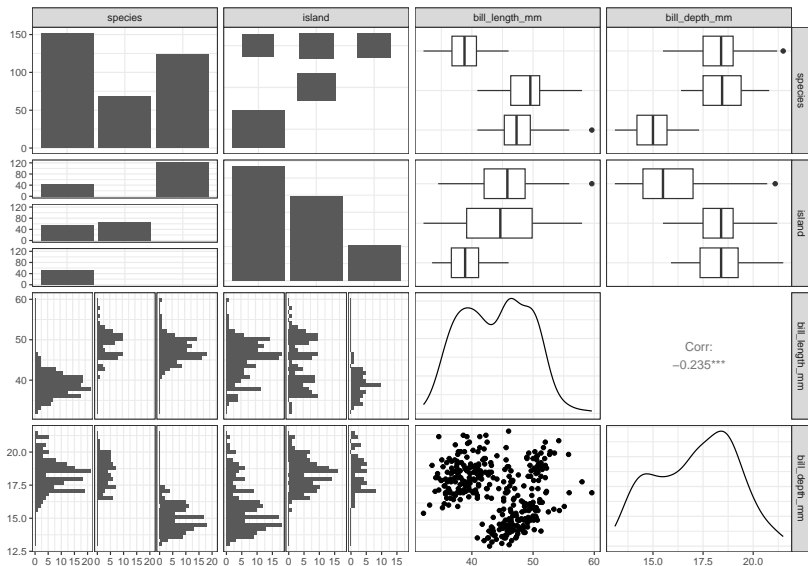
- ▶ When two variables show some connection with one another, they are called ***associated*** variables.
- ▶ If two variables are not associated, i.e. there is no evident connection between the two, then they are said to be ***independent***.

# Scatterplot Matrices - Quick Look at Many Relationships

This code will sometimes run slowly and generate lots of warning messages.

```
penguins |>
# Select variables of interest
  select(species, island, bill_length_mm,
          bill_depth_mm) |>
# send to ggpairs to create the matrix
ggpairs()
```

# Scatterplot Matrices - Quick Look at Many Relationships



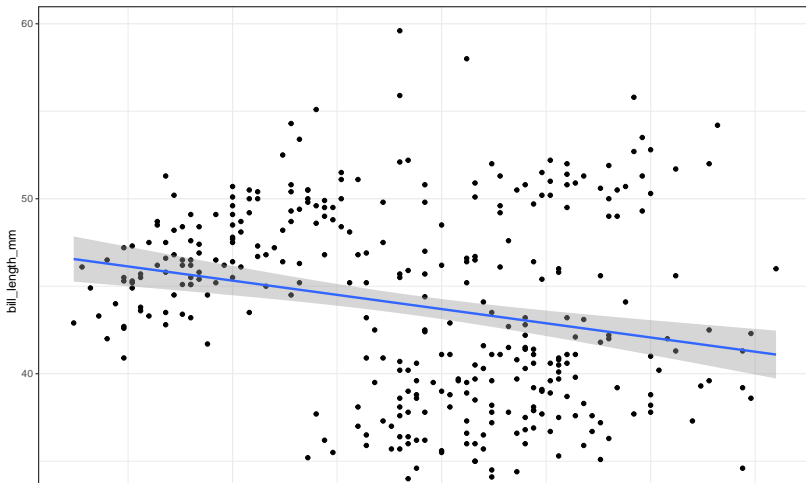
# Scatter plot + Linear Regression - Detailed Look at 1 Relationship

```
# Pipe data into ggplot2
penguins |>
  # Set x and y variables with aes
  ggplot(aes(x = bill_depth_mm,
              y = bill_length_mm)) +
  # add a scatterplot
  geom_point() +
  # add a linear model regression line
  geom_smooth(formula = y ~ x,
              # set method to lm
              method = 'lm',
              # keep standard error shading
              se = T)
```



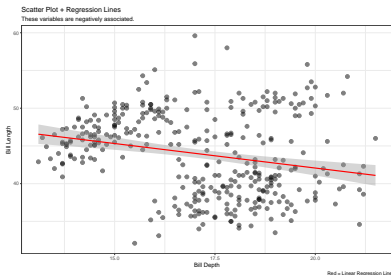
# Scatter plot + Linear Regression - Detailed Look at 1 Relationship

Does this look like this regression line accurately describes the relationship between bill depth and bill length? Do you see any patterns in the points?



# Negatively Correlated Variables

The regression line slopes downwards from the upper left-hand corner towards the lower right-hand corner.



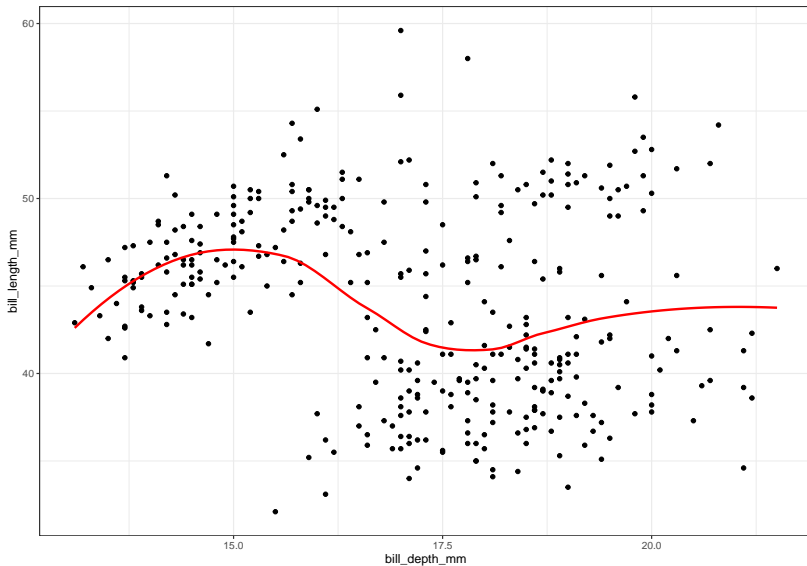
- ▶ Bill length is negatively associated with bill depth for all penguins sampled.
- ▶ Bill length is negatively correlated to bill depth.
- ▶ As bill depth increases, bill length decreases.

# Scatter plot + LOESS Regression - Detailed Look at 1 Relationship

Using the localized regression technique LOESS can help you identify trends in your data that traditional linear models can miss. What do you see here?

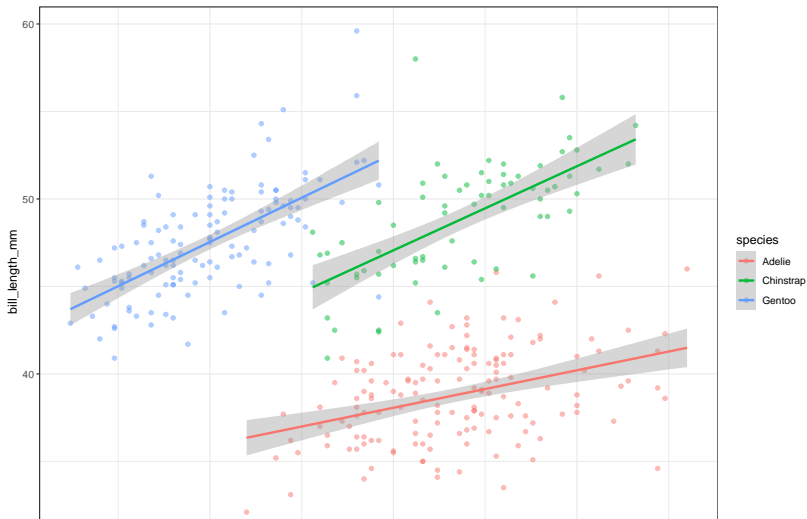
```
# Pipe data into ggplot2
penguins |>
  # Set x and y variables with aes
  ggplot(aes(x = bill_depth_mm,
             y = bill_length_mm)) +
  # add a scatterplot
  geom_point() +
  # add a LOESS regression line
  geom_smooth(formula = y ~ x,
             # set method to loess
             method = 'loess',
             # change the color
             color = 'red',
             # remove standard error shading
```

# Scatter plot + LOESS Regression - Detailed Look at 1 Relationship



# What happens when consider additional variables in the data?

How does this plot differ from the first linear regression analysis on data from penguins not grouped by species?

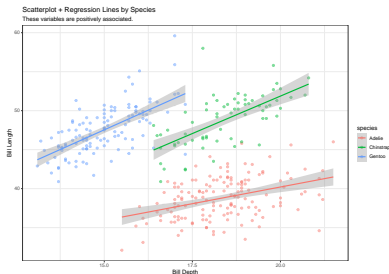


## What happens when consider additional variables in the data?

```
# Pipe data into ggplot2
penguins |>
  # Set x and y variables with aes
  ggplot(aes(x = bill_depth_mm,
             y = bill_length_mm,
             # group the data by species
             group = species,
             # color the points/lines by species
             color = species)) +
  # add a scatterplot
  geom_point(alpha = 0.5) +
  # add a linear model regression line
  geom_smooth(formula = y ~ x,
             # set method to lm
             method = 'lm',
             # keep standard error shading
             se = T)
```

# Positively Correlated Variables

The regression line slopes upwards from the bottom left-hand corner towards the upper right-hand corner.



- ▶ Bill length is positively associated with bill depth within each of the 3 penguin species.
- ▶ Bill length is negatively correlated to bill depth
- ▶ As bill depth increases, bill length decreases.

# Take-Home Lessons

- ▶ Conclusions are shaped by the assumptions we make during the analysis
- ▶ Context is important!
- ▶ A picture is worth a thousand words



## Next time...

- ▶ Please complete your Google surveys
- ▶ Please register for Campuswire
- ▶ Where does your data come from? (defining populations)
- ▶ Principles of sampling (Skittles activity??)
- ▶ DATA1220 pre-survey (FREE 2.5% of final grade)
  - ▶ Please contact me if you will not be in class Friday

## Session Info

At the end of every project, you should include your session info. This function prints out your computer's operating system, the installation of R you are using, and all of your installed packages plus version numbers. This is a good habit for producing reproducible research.

```
xfun::session_info()
```

```
R version 4.4.1 (2024-06-14 ucrt)
```

```
Platform: x86_64-w64-mingw32/x64
```

```
Running under: Windows 11 x64 (build 22631)
```

```
Locale:
```

```
LC_COLLATE=English_United States.utf8
```

```
LC_CTYPE=English_United States.utf8
```

```
LC_MONETARY=English_United States.utf8
```

```
LC_NUMERIC=C
```

```
LC_TIME=English_United States.utf8
```