Class and Coding Expectations (CWRU, Pitt, UCF, UTRGV)

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

27 September, 2022

Contents

4.2.2.1 Class Readings, Assignments, Syllabus Topics

4.2.2.1.1 Reading, Lab Exercises, SemProjects

- Readings:
  - For today: R4DS 1-3
  - For next class: OIS5, (EDA 32-58)
- Laboratory Exercises:
  - LE2 : Due today at midnight
  - LE3 : Given out today
- Office Hours: (Class Canvas Calendar for Zoom Link)
  - Wednesday @ 4:00 PM to 5:00 PM, Will Oltjen
  - Saturday @ 3:00 PM to 4:00 PM, Kristen Hernandez

  - **Office Hours are on Zoom, and recorded**
- Semester Projects
  - DSCI 451 Students Biweekly Update 2 due 9/23
  - DSCI 451 Students
    - Next **Report Out #1 is Due Friday September 30th**
  - All DSCI 351/351M/451 Students:
    - **Peer Grading of Report Out #1 is Due October 11th, 2022**
  - Exams
    - MidTerm: Tuesday October 18th, in class or remote, 11:30 - 12:45 PM
    - Final: Monday December 19, 2022, 12:00PM - 3:00PM, Nord 356 or remote

4.2.2.2 Introduction

- Lets go over some of our expectations regarding this course.

This document outlines general expectations moving forth.

Please read through this document carefully and

- if you have any questions,
- do message us on Slack.

4.2.2.2.1 LE Grading Rubric, as an example Global Points

- -1 pts per day late

Question Points

- 1-1 Completion (½ pt)
- 1-2 Structure Completion (1 pt)
  - Give the Class of Each Column (1/4) pts
  - Subset using versicolor (1/4) pts
  - Mean and Median (1/4) pts
  - lm () & plot (1/4) pts
- 1-3 OIS Completion (1) pt
  - Cases (1/3) pts
  - Numerical Variables (1/3) pts
  - Categorical (1/3) pts
- 1-4 Distributions (1) pt
  - 1/8 pts answer
  - 1/8 pts for explanation
  - 1/8 pts answer
  - 1/8 pts for explanation
  - 1/8 pts answer
  - 1/8 pts for explanation
  - 1/8 pts answer
  - 1/8 pts for explanation
- 1-5 OIS 2.9 (1 pt)
  - 1/8 pts answer
  - 1/8 pts for explanation
  - 1/8 pts answer
  - 1/8 pts for explanation
  - 1/8 pts answer
  - 1/8 pts for explanation
  - 1/8 pts answer
  - 1/8 pts for explanation
- 1-6 For Loops (1/2) pt

  - 1/4 pt Attempt
  - 1/4 pt Success
- 1-7 Heart Transplants (2) pts
  - 1/3 pts Answer + Reason
  - 1/3 pts Answer + Reason
  - 1/3 pts Answer + Reason
  - 1 pts
    * 1/4 pts Answer
    * 1/4 pts Answer
    * 1/2 pts Answer + Reason

4.2.2.3 General Expectations

- Please follow these points carefully as you work on your lab exercises.

The grading will be done based on these points.

To provide an overview of the points:-

- Remember that the knitted PDF/HTML shows the first 80 characters in the code
  - as the output and omits the rest.
- Make sure you format your code chunks properly.
  - Please verify this before submission.
- Since we'll be grading your work, be sure to comment your code.
  - Suppose you use a function that is not common (for example, unite()),
  - explain what it does in a short comment and which package you used.
- It would also help us learn about new functions in the process.

There are multiple approaches to the same question.

- We'll take that into account while grading.

Please answer questions outside the code chunk

- (next to ANSWER -> or Answer).
- Suppose, the question asks you to compare trends,
  - don't type it as a comment in a code chunk.

Be creative while you work on plots.

- As you will learn along, ggplot2 package has amazing functionalities.
  - Spend some time formatting plots to make them more understandable.
- For example,
  - adding a plot title,
  - axes labels (specify units wherever required),
  - changing theme specifications will earn full points.

Include meaningful variable names.

- For a dataset containing billboard lyrics,
  - billboard_lyrics is too long and bbl is too short.
- Find a balance between the two extremes like 'songs'.

4.2.2.4 Slack

- Please use Slack as your primary mode of communication with us,
  - especially for asking questions.
  - This way everyone can learn from the discussion

Before you ask a question,

- please verify if the question has already been asked
- by other students by scrolling up.

Don't forget to tag us by using @

- because it's highly probable that your messages
  - might get lost before the submission.

While asking questions,

- refer to the question number while typing your question
- and show a minimum working example (MWE)
  - (explain what you did and send a screenshot).

If you are facing an error

- or have difficulty in debugging or troubleshooting,
- please send us a screenshot.

It'll be really nice to ask questions within threads

- and post it on the main channel
- so that we won't miss them.

Our handles are -

- TAs:
  - @Will Oltjen (CWRU, TA)
  - @Kristen Hernandez (CWRU, TA)
  - @Mingxuan Li
- Professor:
  - @Roger French
  - @Paul Leu

Try your best to ask your questions in the class channel #dsci351-351m-451.

- This way others can also benefit and help with your answers.
- It'll be a win-win situation!

If something has to be privately discussed with the TAs or Prof. French,

- please DM directly on Slack.
  - or contact by email
- Please do not message LE questions by DMs.

While we will try our best to reply as soon as possible,

- please allow for some potential delays in our response rate.
- We cannot guarantee answering questions
  - 10 minutes before the submission deadline,
  - so make sure you ask questions earlier
  - in order to receive a response from us.

Please be aware that we are a large group of people

- and there are few TAs.
- Please feel free to form your own study groups
  - and discuss among yourselves.
- The TAs, cannot be your study buddies
  - and cannot organize personal Zoom sessions
  - unless it is crucial.

In the Slack channel,

- please don't wait for the TAs to respond to all the questions.
- If you browse through the Slack channel
  - and you know you can answer a question,
  - we encourage you to do that.

We won't be available late at night

- just before the submission deadline.
- Last-minute questions before the deadline
  - will probably not be answered until the next morning.

4.2.2.5 Office Hours

- We have four office hours in the week:-
  - CWRU: Mondays and Wednesdays at 4:30-5:30 pm
  - UPitt: Mondays 11 am-12 pm and Tuesdays 1-2 pm

Please attend the office hours if you have questions and any other concerns.

- These sessions ARE recorded.

Because we have about 10-15 students attending the office hours

- and there are fewer TAs,
- please be patient and ask questions one by one.

Please make sure to give others the time to ask their questions.

- Some students ask several questions at once
  - without giving others time to ask questions.

4.2.2.6 Submission Time

- Typically lab exercises are expected to be submitted by midnight
  - of the deadline day.

Requests made on the day of submission

- will probably not be accommodated.

Make sure you are able to knit your documents earlier before submission.

If you are facing any challenges or problems

- that prevent you from turning in the lab exercise on time,
  - please contact us with an explanation of why you need extra time
  - and we'll see if your request can be accommodated.

4.2.2.7 Git Commands

- **NOTE: NOT FOLLOWING THE GIT COMMANDS COULD LEAD TO SEVERAL PROBLEMS.**

I cannot emphasize enough why this is important.

Sameera's personal story:

- 'when I was a DSCI student, I wasn't regular at git pushing and git pulling.
- At one point, I lost all of my files from the ODS Desktop computer
  - and thankfully, it was possible to retrieve
  - after spending a few hours from the CWRU Network.

- All of this happened just before the midterm exam!
  - It was definitely scary.
- Please don't land in such situations!'

Git pushing and pulling regularly

- will save your work virtually
  - and in events of your computer crashing for no particular reason,
  - you can always clone your fork.
- And remember **every hard drive or usb stick will fail**.
  - You can rely on that

4.2.2.8 OnDemand (Markov, RStudio Server-4.1.1) Versus Your Own Local Computer

- There are several reasons why we encourage you to use ondemand.case.edu.
  - Not just because it is fancy to do virtual computing
    - but also because your session can be recovered
    - (given you input a longer duration).

The packages would be updated from time to time

- and it's easier to reach out to help@case.edu about the problems.

In fact, we can try diagnosing the issues from our end too.

- We understand that because of spotty internet connection,
  - it's can be hard to use virtual computing.
  - But the Rstudio Server's cache keystrokes, which helps
- In such cases, make sure you have the Git Bash app set up
  - and do git pushing/pulling regularly.

If you ever face any issue, please contact us.

Additionally, you can send a ticket to:-

- ODS Desktop problems: help@case.edu
- Markov problems: hpcsupport@case.edu

4.2.2.9 Coding Style

- Code styling is important!
  - As data scientists, part of our job is
    - to effectively communicating our process
    - and results to a broader audience.
  - Martin Fowler: "Any fool can write code that a computer can understand.
    - Good programmers write code that humans can understand."
  - Good coding style helps you find bugs and will save you time in the long run.
  - You may collaborate with others on coding.
    - It is important that others can understand your code.
  - **YOU MUST BE ACTIVELY MAKING YOUR CODE READABLE**.
    - As you are coding, use the below practices to make your code easy to understand.
  - After your code is functional,
    - do several passes over your code to clean up the code,
    - simplify it, and ensure it is something another programmer can understand.

We expect your code to be

- compact, well-commented and effectively answers
  - the question you are trying to solve.

There needs to be a whitespace between symbols and characters.

- `hits %>% filter(Artist=="madonna")` is not the ideal style
- `hits %>% dplyr::filter(Artist == "madonna")` is!

There are multiple methods to solve a question

- and new approaches to solving a question would always be appreciated.

If you are using a new function that is not regularly used,

- please use package::function approach
  - and say what it does.
- It can be a comment in a code chunk.

Going forward, increasing scrutiny will be placed on the following things

- (including the general expectations)
  - and points will be deducted accordingly.

NOTE

When looking at the .Rmd file you can notice something interesting.

Every sentence I write is ended by a **PERIOD** a **SPACE** and a **ENTER** key.

Although it creates a new line of code,

- when it is knitted they will all be in one block of text.

4.2.2.9.1 General

- Use RStudio Diagnostics. Diagnostics can be enabled and set within the `Global Options -> Code -> Diagnostics editing pane:`

Do not use magic numbers.

- [Magic numbers](https://en.wikipedia.org/wiki/Magic_number_(programming) are numbers that appear directly in computations.

- The use of magic numbers
  - obscures the designer's intent in choosing that number,
  - increases possibilities for error,
  - and makes it more difficult to extend and adapt code in the future.

- Magic numbers should be replaced with constants or variables
  - to make the program easier to maintain, understand, and read.
- Use `students_per_group <- 4`
  - `num_groups <- 15`
  - `students <- students_per_group * num_groups`
- Avoid `students <- 4 * 15`

Eliminate code duplication.

- Any block of code appearing in more than one m-file
  - should be considered for packaging as a function.
- It is much easier to manage changes and debug if code appears in only one file.

- Do not copy and paste blocks of code.
  - If you find yourself cutting and pasting code,

  - consider how you can create a new function or subfunction of this block of code.

Make your code versatile.

- Consider how your code could be applied to different scenarios
- Make sure there are error checks
- Test your code (see below)
- Use parentheses. Parentheses help clarify operator precedence.

- Be careful of floating point comparisons

Here is an example.

- You can use the `round` or `eps` function in R
  - to deal with these issues.

```
# This is true
short_side <- 3
long_side <- 5
other_side <- 4
long_side ** 2 == (short_side ** 2 + other_side ** 2)
```

```
## [1] TRUE
```

```
# But this is false
scale_factor <- 0.01
(scale_factor * long_side) ** 2 == ((scale_factor * short_side)
                                    ** 2 + (scale_factor * other_side) ** 2)
```

```
## [1] FALSE
```

Write a test script for every function.

- Use the prefix `test_` for your test function
  - and make use of the `assert` function.

- Write a test script for every function.
  - This practice will improve the quality of the initial version

  - and the reliability of changed versions.
- Consider that any function too difficult to test is probably too difficult to write.

Boris Beizer: "More than the act of testing,

- the act of designing tests is one of the best bug preventers known."
- For example, to test the function fibonacci,
  - you should write `TestFibonacci()`

Functions should be kept short

- (guideline of under 30 lines of code).

Modularize

- Modularization enhances readability, understanding and testing
  - by reducing the amount of text which must be read
  - to see what the code is doing.

- Code longer than two editor screens is a candidate for partitioning.
  - Small well designed functions are more likely
  - to be usable in other applications.
- Use existing functions.

- It may be quicker or surer to find an existing function
  - that provides some or all of the required functionality.

Vectorize

- If you can program the same operations without the use of a `for` loop,
  - then the code should run a lot faster.
- Make use of `lapply()`, `sapply()`, and `tapply()`
- Or better yet, use the Tidyverse

4.2.2.9.2 Naming

- Consistency is very important in naming.
  - The style of a name immediately indicates what the entity is:
    * a function, variable, etc…

Use as descriptive a name as possible.

- Variables should be nouns,
  - while functions should be "command" verbs.

Do not try to save horizontal space,

- as it is far more important to make your code
  - immediately understandable to a reader.
- Use `num_errors` or `num_completed_connections`
- Avoid `n` or `nerr`
- Function names should be imperative (that is they should be commands).
- Use `OpenFile()`

Names of dimensioned variables and constants

- should usually have a units suffix.
- Adding a unit suffix helps to avoid confusion
- Use `incident_angle_radians` not `incident_angle`

NASA lost its $125 million Mars Climate Orbiter

- because of issues with units in their data

Make sure there are no abbreviations in names

- Use whole words to reduce ambiguity
- Use `ComputeArrivalTime()`
  - Avoid `ComputeArr()`
- Specific phrases that are naturally known through abbreviations
  - should be kept abbreviated (e.g., cm).
- Even these cases might benefit from a defining comment
  - at their first appearance.

Names should be pronounceable

- Pronounceable names are easier to remember

4.2.2.9.3 Functions

- CamelCase is preferred for function names.

The prefix `Is` should be used for boolean functions

- `IsOverpriced()`
- There are a few alternatives to the `Is` prefix
  - that fit better in some situations.
- These include the `Has`, `Can`, and `Should` prefixes:
  - `HasLicense()`, `CanEvaluate()`, `ShouldSort()`

Functions should have meaningful names

- Make your filenames very specific.
  - For example, use `http_sever_logs` rather than `logs`
- Use `ComputeTotalWidth()` not `CompWid()`
- An exception is the use of abbreviations or acronyms widely used.
  - Functions with such abbreviations or acronyms
  - should have the complete word in the first header line for clarity
- Functions with a single output can be named for the output
  - Examples are `Mean()`, `StandardError()`
- Functions with no output should be named for what they do
  - An example is `Plot()`

The prefix `Find` can be used in methods where something is looked up.

- Examples are `FindOldestRecord()`; `FindHeaviestElement()`
- Complement names should be used for complement operations
  - Get/Set,
  - Add/Remove/Create/Destroy,
  - Start/Stop,
  - Insert/Delete,
  - Increment/Decrement,
  - Old/New,
  - Begin/End,
  - First/Last,
  - Up/Down,
  - Min/Max,
  - Next/Previous,
  - Open/Close,
  - Show/Hide,
  - Suspend/Resume,
  - etc…

Use explicit `return`

- Use
  - `AddValues <- function(x, y) {`
  - `return(x+y)`
  - `}`
- Not
  - `AddValues <- function(x, y) {`
  - `x+y`
  - `}`

4.2.2.9.4 Variable Names

- The names of variables should document their meaning or use.

The prefix `num` should be used

- for variables representing the number of objects.
  - `num_files`, `num_segments`

The suffix `array` should be used

- for the pluralization of a variable
  - `point_array`

Variables representing a single entity number

- can be suffixed by Number.
  - `table_number`, `employeee_number`

Iterator variables should be named or prefixed with i, j, k etc.

- `for (i_file in Mylist) {}`

Avoid negated Boolean operators

- Use `is_found` not `is_not_found`

4.2.2.10 Comments in your actual code blocks

- Comments are good and should be used.
  - But they can sometimes be used to cover up for badly written code.
  - Many times, heavily commented code is being used
    - only because it is compensating for bad style.
  - If you write your code properly,
    - code comments are might actually superfluous.
  - Before commenting your code,
    - see if you can write your code in a way
    - so that the comments aren't even needed.

Example:

- The following code uses magic numbers,
  - and then explains what the magic numbers are through comments:

```
solar_decom <- lapply(time_series, stlplus, s.window = "periodic", n.p = 24*4)
# The season for the data is one day because the cycle repeats over the course
# of 24 hours. Therefore, the frequency would be 24 hours *  4 measurements
# per hour for a total of 96 measurements in a day.
```

Instead, the following code

- is more readable
  - and reduces the need for comments:

```
hours_per_day <- 24
measurements_per_hour <- 4 # every 15 minutes
solar_decom <- lapply(time_series, stlplus, s.window = "periodic", n.p = hours_per_day*measurements_per
```

4.2.2.11 Comments

- In order for us to better understand our own work
  - and to better to be able to share knowledge with your "customer"
  - is it important to comment your code!

Even with the most stringent of commenting,

- we have returned to scripts with confusion and frustration.

As we like to say it: be kind to your future self!

Therefore, from now on **YOU MUST BE ACTIVELY COMMENTING YOUR CODE**.

We will also be checking your adherence to the 80 character line limit.

A comment is not helpful if it gets cut off halfway through explaining your work.

NOTE: DO NOT WRITE THE TEXT-BASED ANSWER AS A COMMENT!

4.2.2.12 Questions and Answering Style

- One word answers will now not receive full credit.
  - But, you do not need to write an excessive amount for each question.
  - Your comments and visuals should help us understand what you have done
    - to solve the problem
  - and the text portion is for you
    - to explain the results / reasoning from the analysis.

To make your life easier,

- we have included 'ANSWER ->' in places where
  - we expect you to type your reasoning.

We want you to fully explain your reasoning

- and how the data/analysis leads you to these conclusions.

If you feel like you need an additional 'ANSWER ->'

- feel free to add an additional space
- for your explanations! Be thorough!

This will enable us to give partial credit where possible.

Make sure to show the relative code outputs to the questions asked.

Ask yourself if your submission can be read by anyone!

NOTE: DO NOT WRITE A PARAGRAPH FOR AN ANSWER.

It would be easier if you have a text answer right below the question/sub-question.

If you decide to answer everything in one place,

- make sure you are referring to the correct sub-question.

We have observed that some students

- are not able to identify the sub-questions to be answers
- because they are lost in between the description of the question.

Please double check before you submit

- and use `Ctrl+ Shift+ O`
- to view all the important sections and sub-sections.

NOTE

The code blocks and comments in the assignment R Markdown files

- are merely a suggestion of what you can do.

The commments are there to guide your process.

However, there are many approaches to problems as mentioned before

- so your method may not always match and that's okay!

4.2.2.13 Plots

- Data visualization is one of the key components of data science and
  - it is critical that you use proper graphical representation of your analysis.

While this is a topic that will be dealt with in the next few weeks,

- it is a good idea to refer back to this section at a later stage.

While you can use base functions in R like plot(),

- for quick and dirty EDA plots for your personal use
- we want you to be creative and
  - customize professional-looking plots in your submissions.

You won't get full credit

- for the base output of the ggplot2:ggplot() function either.

Experiment with different aspects of the ggplot2 package

- and make your plots more colorful and easy to follow.

The expectations of a good plot include:

- adding a plot title,
- axes labels (with units, if any),
- changing theme specifications,
- adding a meaningful axes scale
- and using good color schemes with appropriate legends.

An example of a professional plot is included below.

NOTE

We do not mean to make you think

- that the more complicated your plot is,
  - the better.

Some things we will be looking for are

- axis labels,
- units,
- titles,
- legends, etc.

Visuals should be purposeful.

If your visuals are too complicated

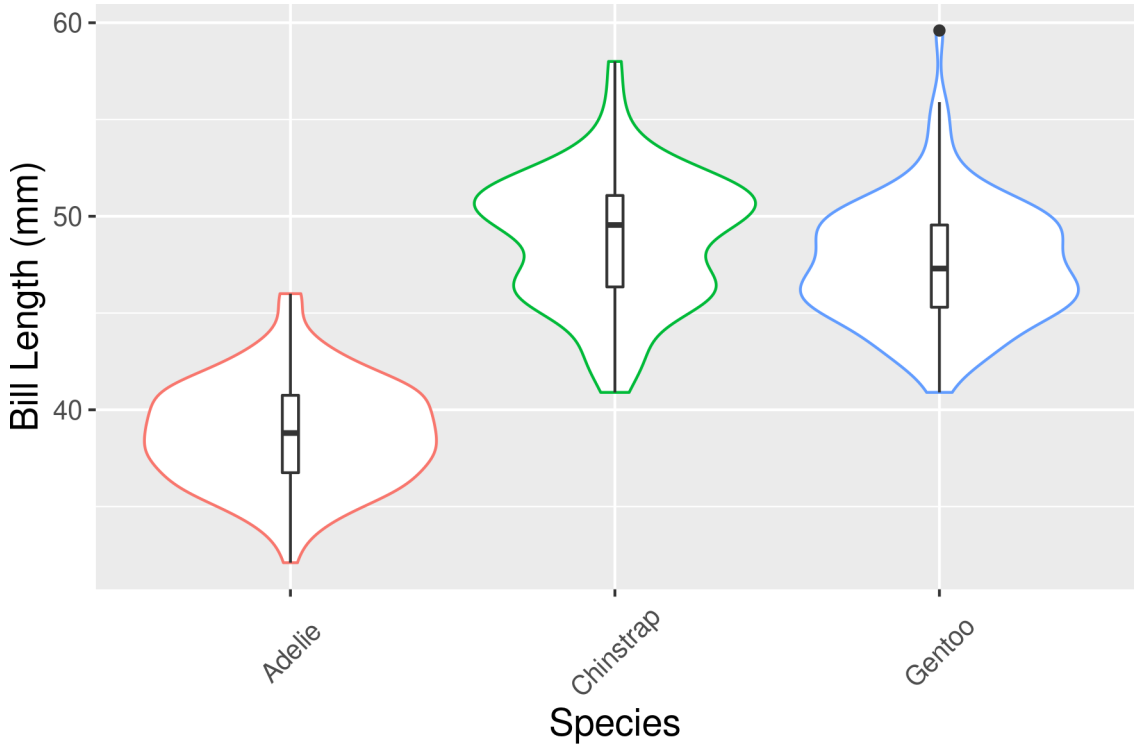
- they can become distracting.

Infact, you can start collecting useful code snippets

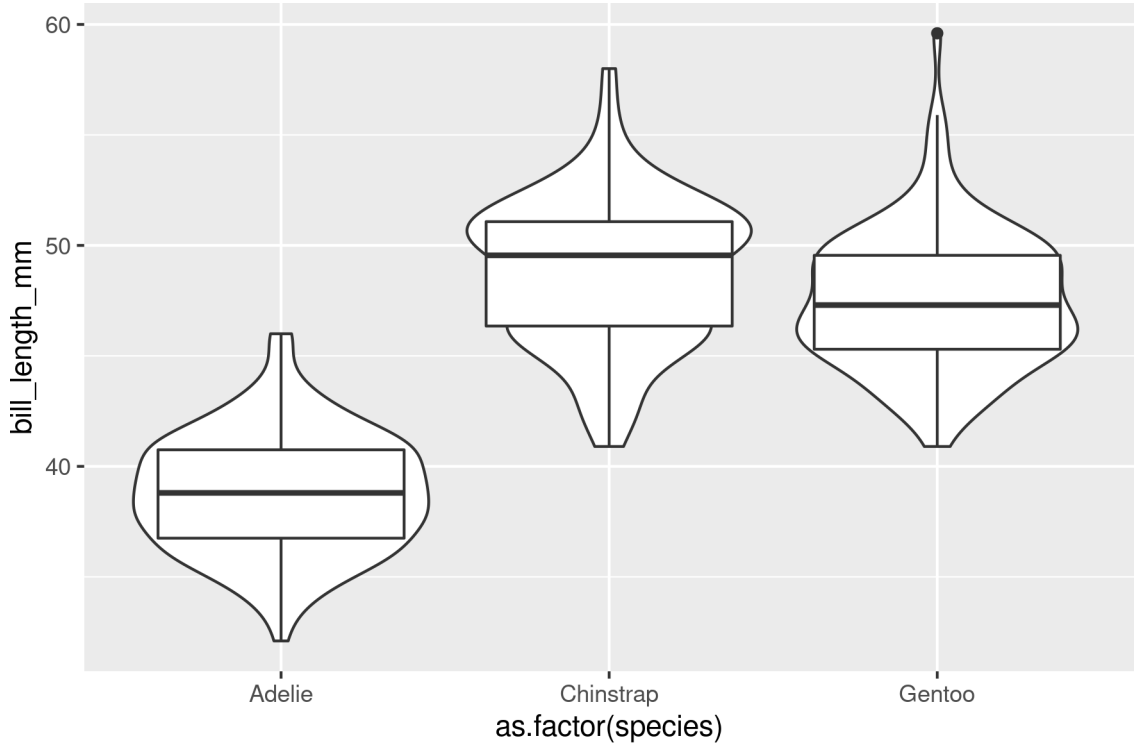
- as either, .R scripts
- or better yet, R Markdown .Rmd files.

```
library(ggplot2)

#using penguins dataset
peng_df <- palmerpenguins::penguins
```

```
mytheme <- theme(
  text = element_text(size = rel(3.5)),
  strip.text.y = element_text(size = rel(3.5)),
  strip.text.x = element_text(size = rel(3.5))
)
#sets a relative size for all labels
#Strip text is used when plots are faceted

mytheme2 <-
  theme(
    axis.text.x = element_text(
      size = 10,
      angle = 45,

      vjust = 0.5
    ),
    axis.text.y = element_text(size = 10),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14)
  )
#sets individual sizes
#ngle determines the angle of the text


ggplot(peng_df) + #calls df
  geom_violin(aes(
    x = as.factor(species),
    y = bill_length_mm,
    color = as.factor(species)
  )) +
  #creates vioplot
  geom_boxplot(aes(x = as.factor(species),
                   y = bill_length_mm),
               width = .05) +
  #Notice I did not add a color here
  #I think it looks bad
  #creates boxplot narrow
  labs(x = 'Species',
       y = 'Bill Length (mm)') + #x & y labs
  mytheme2 +
  #12pt 45degreex .5vjustx
  theme(legend.position = 'none')
```




```
## Versus
ggplot(peng_df) +
  geom_violin(aes(x = as.factor(species),
                  y = bill_length_mm)) +
  geom_boxplot(aes(x = as.factor(species),
                   y = bill_length_mm))
```



4.2.2.14 Grading

- With this information in mind,
  - we would go through each and every code block
  - and read through your comments as well as your answers.

Please be aware that you submit the right set of files

- as mentioned on Canvas submission page.

We find it difficult to follow up individually

- about individual submissions
  - and tracking re-submissions.
- Please avoid such situations.

We would be following an absolute grading system

- in which we grade solely on the answers
- of what a person has included in their submission.

Your score is reflective of your performance on the lab exercise.

Please be mindful that the grading is done on a curve,

- given the diversity of this course.

That way, an A does not mean 90+ points on the course!

4.2.2.15 Shortcuts in RStudio

- To view the document outline
  - and what all questions to answer (as mentioned earlier):
    - `Ctrl + Shift + O`

To reformat code:

- `Ctrl + Shift + A`

Please make sure you use these shortcuts each time you work on the code.

4.2.2.16 Problems with Virtual Computing

- As mentioned earlier,
  - please send a ticket explaining your problem
    - and the class details
    - (that you are a student of DSCI353/353M/453) to:-
    - help@case.edu: if you are facing issues with the ODS VDI

hpcsupport@case.edu:

- if you are facing issues with Markov/RStudio Server

4.2.2.17 Final Thoughts

- The best way to learn this course
  - is not only by learning on your own
    - but also by talking to your peers
    - and communicating with the professor as well as the TAs.

While discussing ideas with your peers is highly encouraged,

- plagiarism will not be tolerated.

We want you to make the best use of this course

- by working sincerely
- and being open to learning new concepts.

As data science concepts can be applied to any industry you can think of,

- please make sure you are paying attention to the coursework
- and learning as much as you can.

Please contact us if you face any difficulty in the course

- and we'd be more than happy to help you out
  - (this encompasses a range of problems
  - from course difficulty to RStudio issues).

Even if you think your questions are silly and not worth asking, please ask.

- We'll try our best to answer them!

We will be on Slack most of the times and have regular office hours.

We'll try and be around on weekends if you have questions.

Before you go, here's the last section you'd want to pay attention to

- (caution: there will be some scrolling
  - but you'll see why it can be annoying
  - to have the entire dataset printed).

4.2.2.18 Code Chunks

- We will first be more closely looking at the outputs of code chunks.

Let's take a look at the following.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(palmerpenguins)

df <- palmerpenguins::penguins

as.data.frame(df)
```

```
##       species    island bill_length_mm bill_depth_mm flipper_length_mm
## 1      Adelie Torgersen           39.1          18.7               181
## 2      Adelie Torgersen           39.5          17.4               186
## 3      Adelie Torgersen           40.3          18.0               195
## 4      Adelie Torgersen             NA            NA                NA
## 5      Adelie Torgersen           36.7          19.3               193
## 6      Adelie Torgersen           39.3          20.6               190
```

```
## 7     Adelie Torgersen           38.9          17.8               181
## 8     Adelie Torgersen           39.2          19.6               195
## 9     Adelie Torgersen           34.1          18.1               193
## 10    Adelie Torgersen           42.0          20.2               190
## 11    Adelie Torgersen           37.8          17.1               186
## 12    Adelie Torgersen           37.8          17.3               180
## 13    Adelie Torgersen           41.1          17.6               182
## 14    Adelie Torgersen           38.6          21.2               191
## 15    Adelie Torgersen           34.6          21.1               198
## 16    Adelie Torgersen           36.6          17.8               185
## 17    Adelie Torgersen           38.7          19.0               195
## 18    Adelie Torgersen           42.5          20.7               197
## 19    Adelie Torgersen           34.4          18.4               184
## 20    Adelie Torgersen           46.0          21.5               194
## 21    Adelie    Biscoe           37.8          18.3               174
## 22    Adelie    Biscoe           37.7          18.7               180
## 23    Adelie    Biscoe           35.9          19.2               189
## 24    Adelie    Biscoe           38.2          18.1               185
## 25    Adelie    Biscoe           38.8          17.2               180
## 26    Adelie    Biscoe           35.3          18.9               187
## 27    Adelie    Biscoe           40.6          18.6               183
## 28    Adelie    Biscoe           40.5          17.9               187
## 29    Adelie    Biscoe           37.9          18.6               172
## 30    Adelie    Biscoe           40.5          18.9               180
## 31    Adelie     Dream           39.5          16.7               178
## 32    Adelie     Dream           37.2          18.1               178
## 33    Adelie     Dream           39.5          17.8               188
## 34    Adelie     Dream           40.9          18.9               184
## 35    Adelie     Dream           36.4          17.0               195
## 36    Adelie     Dream           39.2          21.1               196
## 37    Adelie     Dream           38.8          20.0               190
## 38    Adelie     Dream           42.2          18.5               180
## 39    Adelie     Dream           37.6          19.3               181
## 40    Adelie     Dream           39.8          19.1               184
## 41    Adelie     Dream           36.5          18.0               182
## 42    Adelie     Dream           40.8          18.4               195
## 43    Adelie     Dream           36.0          18.5               186
## 44    Adelie     Dream           44.1          19.7               196
## 45    Adelie     Dream           37.0          16.9               185
## 46    Adelie     Dream           39.6          18.8               190
## 47    Adelie     Dream           41.1          19.0               182
## 48    Adelie     Dream           37.5          18.9               179
## 49    Adelie     Dream           36.0          17.9               190
## 50    Adelie     Dream           42.3          21.2               191
## 51    Adelie    Biscoe           39.6          17.7               186
## 52    Adelie    Biscoe           40.1          18.9               188
## 53    Adelie    Biscoe           35.0          17.9               190
## 54    Adelie    Biscoe           42.0          19.5               200
## 55    Adelie    Biscoe           34.5          18.1               187
## 56    Adelie    Biscoe           41.4          18.6               191
## 57    Adelie    Biscoe           39.0          17.5               186
## 58    Adelie    Biscoe           40.6          18.8               193
## 59    Adelie    Biscoe           36.5          16.6               181
## 60    Adelie    Biscoe           37.6          19.1               194
```

```
## 61     Adelie    Biscoe          35.7          16.9               185
## 62     Adelie    Biscoe          41.3          21.1               195
## 63     Adelie    Biscoe          37.6          17.0               185
## 64     Adelie    Biscoe          41.1          18.2               192
## 65     Adelie    Biscoe          36.4          17.1               184
## 66     Adelie    Biscoe          41.6          18.0               192
## 67     Adelie    Biscoe          35.5          16.2               195
## 68     Adelie    Biscoe          41.1          19.1               188
## 69     Adelie Torgersen          35.9          16.6               190
## 70     Adelie Torgersen          41.8          19.4               198
## 71     Adelie Torgersen          33.5          19.0               190
## 72     Adelie Torgersen          39.7          18.4               190
## 73     Adelie Torgersen          39.6          17.2               196
## 74     Adelie Torgersen          45.8          18.9               197
## 75     Adelie Torgersen          35.5          17.5               190
## 76     Adelie Torgersen          42.8          18.5               195
## 77     Adelie Torgersen          40.9          16.8               191
## 78     Adelie Torgersen          37.2          19.4               184
## 79     Adelie Torgersen          36.2          16.1               187
## 80     Adelie Torgersen          42.1          19.1               195
## 81     Adelie Torgersen          34.6          17.2               189
## 82     Adelie Torgersen          42.9          17.6               196
## 83     Adelie Torgersen          36.7          18.8               187
## 84     Adelie Torgersen          35.1          19.4               193
## 85     Adelie     Dream          37.3          17.8               191
## 86     Adelie     Dream          41.3          20.3               194
## 87     Adelie     Dream          36.3          19.5               190
## 88     Adelie     Dream          36.9          18.6               189
## 89     Adelie     Dream          38.3          19.2               189
## 90     Adelie     Dream          38.9          18.8               190
## 91     Adelie     Dream          35.7          18.0               202
## 92     Adelie     Dream          41.1          18.1               205
## 93     Adelie     Dream          34.0          17.1               185
## 94     Adelie     Dream          39.6          18.1               186
## 95     Adelie     Dream          36.2          17.3               187
## 96     Adelie     Dream          40.8          18.9               208
## 97     Adelie     Dream          38.1          18.6               190
## 98     Adelie     Dream          40.3          18.5               196
## 99     Adelie     Dream          33.1          16.1               178
## 100    Adelie     Dream          43.2          18.5               192
## 101    Adelie    Biscoe          35.0          17.9               192
## 102    Adelie    Biscoe          41.0          20.0               203
## 103    Adelie    Biscoe          37.7          16.0               183
## 104    Adelie    Biscoe          37.8          20.0               190
## 105    Adelie    Biscoe          37.9          18.6               193
## 106    Adelie    Biscoe          39.7          18.9               184
## 107    Adelie    Biscoe          38.6          17.2               199
## 108    Adelie    Biscoe          38.2          20.0               190
## 109    Adelie    Biscoe          38.1          17.0               181
## 110    Adelie    Biscoe          43.2          19.0               197
## 111    Adelie    Biscoe          38.1          16.5               198
## 112    Adelie    Biscoe          45.6          20.3               191
## 113    Adelie    Biscoe          39.7          17.7               193
## 114    Adelie    Biscoe          42.2          19.5               197
```

```
## 115    Adelie    Biscoe           39.6          20.7              191
## 116    Adelie    Biscoe           42.7          18.3              196
## 117    Adelie Torgersen           38.6          17.0              188
## 118    Adelie Torgersen           37.3          20.5              199
## 119    Adelie Torgersen           35.7          17.0              189
## 120    Adelie Torgersen           41.1          18.6              189
## 121    Adelie Torgersen           36.2          17.2              187
## 122    Adelie Torgersen           37.7          19.8              198
## 123    Adelie Torgersen           40.2          17.0              176
## 124    Adelie Torgersen           41.4          18.5              202
## 125    Adelie Torgersen           35.2          15.9              186
## 126    Adelie Torgersen           40.6          19.0              199
## 127    Adelie Torgersen           38.8          17.6              191
## 128    Adelie Torgersen           41.5          18.3              195
## 129    Adelie Torgersen           39.0          17.1              191
## 130    Adelie Torgersen           44.1          18.0              210
## 131    Adelie Torgersen           38.5          17.9              190
## 132    Adelie Torgersen           43.1          19.2              197
## 133    Adelie     Dream           36.8          18.5              193
## 134    Adelie     Dream           37.5          18.5              199
## 135    Adelie     Dream           38.1          17.6              187
## 136    Adelie     Dream           41.1          17.5              190
## 137    Adelie     Dream           35.6          17.5              191
## 138    Adelie     Dream           40.2          20.1              200
## 139    Adelie     Dream           37.0          16.5              185
## 140    Adelie     Dream           39.7          17.9              193
## 141    Adelie     Dream           40.2          17.1              193
## 142    Adelie     Dream           40.6          17.2              187
## 143    Adelie     Dream           32.1          15.5              188
## 144    Adelie     Dream           40.7          17.0              190
## 145    Adelie     Dream           37.3          16.8              192
## 146    Adelie     Dream           39.0          18.7              185
## 147    Adelie     Dream           39.2          18.6              190
## 148    Adelie     Dream           36.6          18.4              184
## 149    Adelie     Dream           36.0          17.8              195
## 150    Adelie     Dream           37.8          18.1              193
## 151    Adelie     Dream           36.0          17.1              187
## 152    Adelie     Dream           41.5          18.5              201
## 153    Gentoo    Biscoe           46.1          13.2              211
## 154    Gentoo    Biscoe           50.0          16.3              230
## 155    Gentoo    Biscoe           48.7          14.1              210
## 156    Gentoo    Biscoe           50.0          15.2              218
## 157    Gentoo    Biscoe           47.6          14.5              215
## 158    Gentoo    Biscoe           46.5          13.5              210
## 159    Gentoo    Biscoe           45.4          14.6              211
## 160    Gentoo    Biscoe           46.7          15.3              219
## 161    Gentoo    Biscoe           43.3          13.4              209
## 162    Gentoo    Biscoe           46.8          15.4              215
## 163    Gentoo    Biscoe           40.9          13.7              214
## 164    Gentoo    Biscoe           49.0          16.1              216
## 165    Gentoo    Biscoe           45.5          13.7              214
## 166    Gentoo    Biscoe           48.4          14.6              213
## 167    Gentoo    Biscoe           45.8          14.6              210
## 168    Gentoo    Biscoe           49.3          15.7              217
```

```
## 169    Gentoo    Biscoe          42.0          13.5          210
## 170    Gentoo    Biscoe          49.2          15.2          221
## 171    Gentoo    Biscoe          46.2          14.5          209
## 172    Gentoo    Biscoe          48.7          15.1          222
## 173    Gentoo    Biscoe          50.2          14.3          218
## 174    Gentoo    Biscoe          45.1          14.5          215
## 175    Gentoo    Biscoe          46.5          14.5          213
## 176    Gentoo    Biscoe          46.3          15.8          215
## 177    Gentoo    Biscoe          42.9          13.1          215
## 178    Gentoo    Biscoe          46.1          15.1          215
## 179    Gentoo    Biscoe          44.5          14.3          216
## 180    Gentoo    Biscoe          47.8          15.0          215
## 181    Gentoo    Biscoe          48.2          14.3          210
## 182    Gentoo    Biscoe          50.0          15.3          220
## 183    Gentoo    Biscoe          47.3          15.3          222
## 184    Gentoo    Biscoe          42.8          14.2          209
## 185    Gentoo    Biscoe          45.1          14.5          207
## 186    Gentoo    Biscoe          59.6          17.0          230
## 187    Gentoo    Biscoe          49.1          14.8          220
## 188    Gentoo    Biscoe          48.4          16.3          220
## 189    Gentoo    Biscoe          42.6          13.7          213
## 190    Gentoo    Biscoe          44.4          17.3          219
## 191    Gentoo    Biscoe          44.0          13.6          208
## 192    Gentoo    Biscoe          48.7          15.7          208
## 193    Gentoo    Biscoe          42.7          13.7          208
## 194    Gentoo    Biscoe          49.6          16.0          225
## 195    Gentoo    Biscoe          45.3          13.7          210
## 196    Gentoo    Biscoe          49.6          15.0          216
## 197    Gentoo    Biscoe          50.5          15.9          222
## 198    Gentoo    Biscoe          43.6          13.9          217
## 199    Gentoo    Biscoe          45.5          13.9          210
## 200    Gentoo    Biscoe          50.5          15.9          225
## 201    Gentoo    Biscoe          44.9          13.3          213
## 202    Gentoo    Biscoe          45.2          15.8          215
## 203    Gentoo    Biscoe          46.6          14.2          210
## 204    Gentoo    Biscoe          48.5          14.1          220
## 205    Gentoo    Biscoe          45.1          14.4          210
## 206    Gentoo    Biscoe          50.1          15.0          225
## 207    Gentoo    Biscoe          46.5          14.4          217
## 208    Gentoo    Biscoe          45.0          15.4          220
## 209    Gentoo    Biscoe          43.8          13.9          208
## 210    Gentoo    Biscoe          45.5          15.0          220
## 211    Gentoo    Biscoe          43.2          14.5          208
## 212    Gentoo    Biscoe          50.4          15.3          224
## 213    Gentoo    Biscoe          45.3          13.8          208
## 214    Gentoo    Biscoe          46.2          14.9          221
## 215    Gentoo    Biscoe          45.7          13.9          214
## 216    Gentoo    Biscoe          54.3          15.7          231
## 217    Gentoo    Biscoe          45.8          14.2          219
## 218    Gentoo    Biscoe          49.8          16.8          230
## 219    Gentoo    Biscoe          46.2          14.4          214
## 220    Gentoo    Biscoe          49.5          16.2          229
## 221    Gentoo    Biscoe          43.5          14.2          220
## 222    Gentoo    Biscoe          50.7          15.0          223
```

```
## 223    Gentoo    Biscoe           47.7          15.0             216
## 224    Gentoo    Biscoe           46.4          15.6             221
## 225    Gentoo    Biscoe           48.2          15.6             221
## 226    Gentoo    Biscoe           46.5          14.8             217
## 227    Gentoo    Biscoe           46.4          15.0             216
## 228    Gentoo    Biscoe           48.6          16.0             230
## 229    Gentoo    Biscoe           47.5          14.2             209
## 230    Gentoo    Biscoe           51.1          16.3             220
## 231    Gentoo    Biscoe           45.2          13.8             215
## 232    Gentoo    Biscoe           45.2          16.4             223
## 233    Gentoo    Biscoe           49.1          14.5             212
## 234    Gentoo    Biscoe           52.5          15.6             221
## 235    Gentoo    Biscoe           47.4          14.6             212
## 236    Gentoo    Biscoe           50.0          15.9             224
## 237    Gentoo    Biscoe           44.9          13.8             212
## 238    Gentoo    Biscoe           50.8          17.3             228
## 239    Gentoo    Biscoe           43.4          14.4             218
## 240    Gentoo    Biscoe           51.3          14.2             218
## 241    Gentoo    Biscoe           47.5          14.0             212
## 242    Gentoo    Biscoe           52.1          17.0             230
## 243    Gentoo    Biscoe           47.5          15.0             218
## 244    Gentoo    Biscoe           52.2          17.1             228
## 245    Gentoo    Biscoe           45.5          14.5             212
## 246    Gentoo    Biscoe           49.5          16.1             224
## 247    Gentoo    Biscoe           44.5          14.7             214
## 248    Gentoo    Biscoe           50.8          15.7             226
## 249    Gentoo    Biscoe           49.4          15.8             216
## 250    Gentoo    Biscoe           46.9          14.6             222
## 251    Gentoo    Biscoe           48.4          14.4             203
## 252    Gentoo    Biscoe           51.1          16.5             225
## 253    Gentoo    Biscoe           48.5          15.0             219
## 254    Gentoo    Biscoe           55.9          17.0             228
## 255    Gentoo    Biscoe           47.2          15.5             215
## 256    Gentoo    Biscoe           49.1          15.0             228
## 257    Gentoo    Biscoe           47.3          13.8             216
## 258    Gentoo    Biscoe           46.8          16.1             215
## 259    Gentoo    Biscoe           41.7          14.7             210
## 260    Gentoo    Biscoe           53.4          15.8             219
## 261    Gentoo    Biscoe           43.3          14.0             208
## 262    Gentoo    Biscoe           48.1          15.1             209
## 263    Gentoo    Biscoe           50.5          15.2             216
## 264    Gentoo    Biscoe           49.8          15.9             229
## 265    Gentoo    Biscoe           43.5          15.2             213
## 266    Gentoo    Biscoe           51.5          16.3             230
## 267    Gentoo    Biscoe           46.2          14.1             217
## 268    Gentoo    Biscoe           55.1          16.0             230
## 269    Gentoo    Biscoe           44.5          15.7             217
## 270    Gentoo    Biscoe           48.8          16.2             222
## 271    Gentoo    Biscoe           47.2          13.7             214
## 272    Gentoo    Biscoe             NA            NA              NA
## 273    Gentoo    Biscoe           46.8          14.3             215
## 274    Gentoo    Biscoe           50.4          15.7             222
## 275    Gentoo    Biscoe           45.2          14.8             212
## 276    Gentoo    Biscoe           49.9          16.1             213
```

```
## 277 Chinstrap     Dream           46.5          17.9               192
## 278 Chinstrap     Dream           50.0          19.5               196
## 279 Chinstrap     Dream           51.3          19.2               193
## 280 Chinstrap     Dream           45.4          18.7               188
## 281 Chinstrap     Dream           52.7          19.8               197
## 282 Chinstrap     Dream           45.2          17.8               198
## 283 Chinstrap     Dream           46.1          18.2               178
## 284 Chinstrap     Dream           51.3          18.2               197
## 285 Chinstrap     Dream           46.0          18.9               195
## 286 Chinstrap     Dream           51.3          19.9               198
## 287 Chinstrap     Dream           46.6          17.8               193
## 288 Chinstrap     Dream           51.7          20.3               194
## 289 Chinstrap     Dream           47.0          17.3               185
## 290 Chinstrap     Dream           52.0          18.1               201
## 291 Chinstrap     Dream           45.9          17.1               190
## 292 Chinstrap     Dream           50.5          19.6               201
## 293 Chinstrap     Dream           50.3          20.0               197
## 294 Chinstrap     Dream           58.0          17.8               181
## 295 Chinstrap     Dream           46.4          18.6               190
## 296 Chinstrap     Dream           49.2          18.2               195
## 297 Chinstrap     Dream           42.4          17.3               181
## 298 Chinstrap     Dream           48.5          17.5               191
## 299 Chinstrap     Dream           43.2          16.6               187
## 300 Chinstrap     Dream           50.6          19.4               193
## 301 Chinstrap     Dream           46.7          17.9               195
## 302 Chinstrap     Dream           52.0          19.0               197
## 303 Chinstrap     Dream           50.5          18.4               200
## 304 Chinstrap     Dream           49.5          19.0               200
## 305 Chinstrap     Dream           46.4          17.8               191
## 306 Chinstrap     Dream           52.8          20.0               205
## 307 Chinstrap     Dream           40.9          16.6               187
## 308 Chinstrap     Dream           54.2          20.8               201
## 309 Chinstrap     Dream           42.5          16.7               187
## 310 Chinstrap     Dream           51.0          18.8               203
## 311 Chinstrap     Dream           49.7          18.6               195
## 312 Chinstrap     Dream           47.5          16.8               199
## 313 Chinstrap     Dream           47.6          18.3               195
## 314 Chinstrap     Dream           52.0          20.7               210
## 315 Chinstrap     Dream           46.9          16.6               192
## 316 Chinstrap     Dream           53.5          19.9               205
## 317 Chinstrap     Dream           49.0          19.5               210
## 318 Chinstrap     Dream           46.2          17.5               187
## 319 Chinstrap     Dream           50.9          19.1               196
## 320 Chinstrap     Dream           45.5          17.0               196
## 321 Chinstrap     Dream           50.9          17.9               196
## 322 Chinstrap     Dream           50.8          18.5               201
## 323 Chinstrap     Dream           50.1          17.9               190
## 324 Chinstrap     Dream           49.0          19.6               212
## 325 Chinstrap     Dream           51.5          18.7               187
## 326 Chinstrap     Dream           49.8          17.3               198
## 327 Chinstrap     Dream           48.1          16.4               199
## 328 Chinstrap     Dream           51.4          19.0               201
## 329 Chinstrap     Dream           45.7          17.3               193
## 330 Chinstrap     Dream           50.7          19.7               203
```

```
## 331 Chinstrap     Dream          42.5          17.3               187
## 332 Chinstrap     Dream          52.2          18.8               197
## 333 Chinstrap     Dream          45.2          16.6               191
## 334 Chinstrap     Dream          49.3          19.9               203
## 335 Chinstrap     Dream          50.2          18.8               202
## 336 Chinstrap     Dream          45.6          19.4               194
## 337 Chinstrap     Dream          51.9          19.5               206
## 338 Chinstrap     Dream          46.8          16.5               189
## 339 Chinstrap     Dream          45.7          17.0               195
## 340 Chinstrap     Dream          55.8          19.8               207
## 341 Chinstrap     Dream          43.5          18.1               202
## 342 Chinstrap     Dream          49.6          18.2               193
## 343 Chinstrap     Dream          50.8          19.0               210
## 344 Chinstrap     Dream          50.2          18.7               198
##     body_mass_g    sex year
## 1          3750   male 2007
## 2          3800 female 2007
## 3          3250 female 2007
## 4            NA   <NA> 2007
## 5          3450 female 2007
## 6          3650   male 2007
## 7          3625 female 2007
## 8          4675   male 2007
## 9          3475   <NA> 2007
## 10         4250   <NA> 2007
## 11         3300   <NA> 2007
## 12         3700   <NA> 2007
## 13         3200 female 2007
## 14         3800   male 2007
## 15         4400   male 2007
## 16         3700 female 2007
## 17         3450 female 2007
## 18         4500   male 2007
## 19         3325 female 2007
## 20         4200   male 2007
## 21         3400 female 2007
## 22         3600   male 2007
## 23         3800 female 2007
## 24         3950   male 2007
## 25         3800   male 2007
## 26         3800 female 2007
## 27         3550   male 2007
## 28         3200 female 2007
## 29         3150 female 2007
## 30         3950   male 2007
## 31         3250 female 2007
## 32         3900   male 2007
## 33         3300 female 2007
## 34         3900   male 2007
## 35         3325 female 2007
## 36         4150   male 2007
## 37         3950   male 2007
## 38         3550 female 2007
## 39         3300 female 2007
```

```
## 40         4650   male 2007
## 41         3150 female 2007
## 42         3900   male 2007
## 43         3100 female 2007
## 44         4400   male 2007
## 45         3000 female 2007
## 46         4600   male 2007
## 47         3425   male 2007
## 48         2975   <NA> 2007
## 49         3450 female 2007
## 50         4150   male 2007
## 51         3500 female 2008
## 52         4300   male 2008
## 53         3450 female 2008
## 54         4050   male 2008
## 55         2900 female 2008
## 56         3700   male 2008
## 57         3550 female 2008
## 58         3800   male 2008
## 59         2850 female 2008
## 60         3750   male 2008
## 61         3150 female 2008
## 62         4400   male 2008
## 63         3600 female 2008
## 64         4050   male 2008
## 65         2850 female 2008
## 66         3950   male 2008
## 67         3350 female 2008
## 68         4100   male 2008
## 69         3050 female 2008
## 70         4450   male 2008
## 71         3600 female 2008
## 72         3900   male 2008
## 73         3550 female 2008
## 74         4150   male 2008
## 75         3700 female 2008
## 76         4250   male 2008
## 77         3700 female 2008
## 78         3900   male 2008
## 79         3550 female 2008
## 80         4000   male 2008
## 81         3200 female 2008
## 82         4700   male 2008
## 83         3800 female 2008
## 84         4200   male 2008
## 85         3350 female 2008
## 86         3550   male 2008
## 87         3800   male 2008
## 88         3500 female 2008
## 89         3950   male 2008
## 90         3600 female 2008
## 91         3550 female 2008
## 92         4300   male 2008
## 93         3400 female 2008
```

```
## 94         4450   male 2008
## 95         3300 female 2008
## 96         4300   male 2008
## 97         3700 female 2008
## 98         4350   male 2008
## 99         2900 female 2008
## 100        4100   male 2008
## 101        3725 female 2009
## 102        4725   male 2009
## 103        3075 female 2009
## 104        4250   male 2009
## 105        2925 female 2009
## 106        3550   male 2009
## 107        3750 female 2009
## 108        3900   male 2009
## 109        3175 female 2009
## 110        4775   male 2009
## 111        3825 female 2009
## 112        4600   male 2009
## 113        3200 female 2009
## 114        4275   male 2009
## 115        3900 female 2009
## 116        4075   male 2009
## 117        2900 female 2009
## 118        3775   male 2009
## 119        3350 female 2009
## 120        3325   male 2009
## 121        3150 female 2009
## 122        3500   male 2009
## 123        3450 female 2009
## 124        3875   male 2009
## 125        3050 female 2009
## 126        4000   male 2009
## 127        3275 female 2009
## 128        4300   male 2009
## 129        3050 female 2009
## 130        4000   male 2009
## 131        3325 female 2009
## 132        3500   male 2009
## 133        3500 female 2009
## 134        4475   male 2009
## 135        3425 female 2009
## 136        3900   male 2009
## 137        3175 female 2009
## 138        3975   male 2009
## 139        3400 female 2009
## 140        4250   male 2009
## 141        3400 female 2009
## 142        3475   male 2009
## 143        3050 female 2009
## 144        3725   male 2009
## 145        3000 female 2009
## 146        3650   male 2009
## 147        4250   male 2009
```

```
## 148        3475 female 2009
## 149        3450 female 2009
## 150        3750   male 2009
## 151        3700 female 2009
## 152        4000   male 2009
## 153        4500 female 2007
## 154        5700   male 2007
## 155        4450 female 2007
## 156        5700   male 2007
## 157        5400   male 2007
## 158        4550 female 2007
## 159        4800 female 2007
## 160        5200   male 2007
## 161        4400 female 2007
## 162        5150   male 2007
## 163        4650 female 2007
## 164        5550   male 2007
## 165        4650 female 2007
## 166        5850   male 2007
## 167        4200 female 2007
## 168        5850   male 2007
## 169        4150 female 2007
## 170        6300   male 2007
## 171        4800 female 2007
## 172        5350   male 2007
## 173        5700   male 2007
## 174        5000 female 2007
## 175        4400 female 2007
## 176        5050   male 2007
## 177        5000 female 2007
## 178        5100   male 2007
## 179        4100   <NA> 2007
## 180        5650   male 2007
## 181        4600 female 2007
## 182        5550   male 2007
## 183        5250   male 2007
## 184        4700 female 2007
## 185        5050 female 2007
## 186        6050   male 2007
## 187        5150 female 2008
## 188        5400   male 2008
## 189        4950 female 2008
## 190        5250   male 2008
## 191        4350 female 2008
## 192        5350   male 2008
## 193        3950 female 2008
## 194        5700   male 2008
## 195        4300 female 2008
## 196        4750   male 2008
## 197        5550   male 2008
## 198        4900 female 2008
## 199        4200 female 2008
## 200        5400   male 2008
## 201        5100 female 2008
```

```
## 202        5300   male 2008
## 203        4850 female 2008
## 204        5300   male 2008
## 205        4400 female 2008
## 206        5000   male 2008
## 207        4900 female 2008
## 208        5050   male 2008
## 209        4300 female 2008
## 210        5000   male 2008
## 211        4450 female 2008
## 212        5550   male 2008
## 213        4200 female 2008
## 214        5300   male 2008
## 215        4400 female 2008
## 216        5650   male 2008
## 217        4700 female 2008
## 218        5700   male 2008
## 219        4650   <NA> 2008
## 220        5800   male 2008
## 221        4700 female 2008
## 222        5550   male 2008
## 223        4750 female 2008
## 224        5000   male 2008
## 225        5100   male 2008
## 226        5200 female 2008
## 227        4700 female 2008
## 228        5800   male 2008
## 229        4600 female 2008
## 230        6000   male 2008
## 231        4750 female 2008
## 232        5950   male 2008
## 233        4625 female 2009
## 234        5450   male 2009
## 235        4725 female 2009
## 236        5350   male 2009
## 237        4750 female 2009
## 238        5600   male 2009
## 239        4600 female 2009
## 240        5300   male 2009
## 241        4875 female 2009
## 242        5550   male 2009
## 243        4950 female 2009
## 244        5400   male 2009
## 245        4750 female 2009
## 246        5650   male 2009
## 247        4850 female 2009
## 248        5200   male 2009
## 249        4925   male 2009
## 250        4875 female 2009
## 251        4625 female 2009
## 252        5250   male 2009
## 253        4850 female 2009
## 254        5600   male 2009
## 255        4975 female 2009
```

```
## 256        5500   male 2009
## 257        4725   <NA> 2009
## 258        5500   male 2009
## 259        4700 female 2009
## 260        5500   male 2009
## 261        4575 female 2009
## 262        5500   male 2009
## 263        5000 female 2009
## 264        5950   male 2009
## 265        4650 female 2009
## 266        5500   male 2009
## 267        4375 female 2009
## 268        5850   male 2009
## 269        4875   <NA> 2009
## 270        6000   male 2009
## 271        4925 female 2009
## 272          NA   <NA> 2009
## 273        4850 female 2009
## 274        5750   male 2009
## 275        5200 female 2009
## 276        5400   male 2009
## 277        3500 female 2007
## 278        3900   male 2007
## 279        3650   male 2007
## 280        3525 female 2007
## 281        3725   male 2007
## 282        3950 female 2007
## 283        3250 female 2007
## 284        3750   male 2007
## 285        4150 female 2007
## 286        3700   male 2007
## 287        3800 female 2007
## 288        3775   male 2007
## 289        3700 female 2007
## 290        4050   male 2007
## 291        3575 female 2007
## 292        4050   male 2007
## 293        3300   male 2007
## 294        3700 female 2007
## 295        3450 female 2007
## 296        4400   male 2007
## 297        3600 female 2007
## 298        3400   male 2007
## 299        2900 female 2007
## 300        3800   male 2007
## 301        3300 female 2007
## 302        4150   male 2007
## 303        3400 female 2008
## 304        3800   male 2008
## 305        3700 female 2008
## 306        4550   male 2008
## 307        3200 female 2008
## 308        4300   male 2008
## 309        3350 female 2008
```

```
## 310        4100   male 2008
## 311        3600   male 2008
## 312        3900 female 2008
## 313        3850 female 2008
## 314        4800   male 2008
## 315        2700 female 2008
## 316        4500   male 2008
## 317        3950   male 2008
## 318        3650 female 2008
## 319        3550   male 2008
## 320        3500 female 2008
## 321        3675 female 2009
## 322        4450   male 2009
## 323        3400 female 2009
## 324        4300   male 2009
## 325        3250   male 2009
## 326        3675 female 2009
## 327        3325 female 2009
## 328        3950   male 2009
## 329        3600 female 2009
## 330        4050   male 2009
## 331        3350 female 2009
## 332        3450   male 2009
## 333        3250 female 2009
## 334        4050   male 2009
## 335        3800   male 2009
## 336        3525 female 2009
## 337        3950   male 2009
## 338        3650 female 2009
## 339        3650 female 2009
## 340        4000   male 2009
## 341        3400 female 2009
## 342        3775   male 2009
## 343        4100   male 2009
## 344        3775 female 2009
```

Here, we have forced the knitted file to express the entire dataset, a full 79 pages!

Additionally when tidyverse was used in R

- it expressed a bunch of stuff
  - we don't want in our final report!

We do not want to see either of these examples in a submitted assignment anymore!

If you need to print a data frame or value

- make sure that it does not take up pages.

We suggest you use dplyr::glimpse() or dplyr::tibble()

- which both natively suppress base::print() statements
  - to just a few rows.

Notice that we are declaring the namespace when we code.

- There are thousands of packages,
- so it is often necessary what package's function you are using!

We can suppress WARNINGS and MESSAGES in our finished report

- by adding it to the beginning of our code blocks like so:-

```
library(tidyverse)
library(palmerpenguins)
df <- palmerpenguins::penguins

df %>%
  dplyr::group_by(year) %>%
  dplyr::tally()
```

```
## # A tibble: 3 x 2
##    year     n
##   <int> <int>
## 1  2007   110
## 2  2008   114
## 3  2009   120
```

```
##Print Statements should go at the end of a code block
dplyr::tibble(df)
```

```
## # A tibble: 344 x 8
##    species island    bill_length_mm bill_depth_mm flipper_~1 body_~2 sex    year
##    <fct>   <fct>              <dbl>         <dbl>      <int>   <int> <fct> <int>
##  1 Adelie  Torgersen           39.1          18.7        181    3750 male   2007
##  2 Adelie  Torgersen           39.5          17.4        186    3800 fema~  2007
##  3 Adelie  Torgersen           40.3          18          195    3250 fema~  2007
##  4 Adelie  Torgersen           NA            NA           NA      NA <NA>   2007
##  5 Adelie  Torgersen           36.7          19.3        193    3450 fema~  2007
##  6 Adelie  Torgersen           39.3          20.6        190    3650 male   2007
##  7 Adelie  Torgersen           38.9          17.8        181    3625 fema~  2007
##  8 Adelie  Torgersen           39.2          19.6        195    4675 male   2007
##  9 Adelie  Torgersen           34.1          18.1        193    3475 <NA>   2007
## 10 Adelie  Torgersen           42            20.2        190    4250 <NA>   2007
## # ... with 334 more rows, and abbreviated variable names 1: flipper_length_mm,
## #   2: body_mass_g
```

```
dplyr::glimpse(df)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

That's it! Hope this gives you a better idea of our expectations early on.

Happy learning and hope you all do well!