Assignment 8

Ellen Bledsoe

2024-03-18

Assignment Details

Purpose

The goal of this assignment is to practice problem decomposition and some best practices in reproducibility .

Task

Write R code to successfully answer each question below.

Criteria for Success

- Code is within the provided code chunks or new code chunks are created where necessary
- Code chunks run without errors
- Code chunks have brief comments indicating which code is answering which part of the question
- Code will be assessed as follows:
  - Produces the correct answer using the requested approach: 100%
  - Generally uses the right approach, but a minor mistake results in an incorrect answer: 90%
  - Attempts to solve the problem and makes some progress using the core concept, but returns the wrong answer and does not demonstrate comfort with the core concept: 50%
  - Answer demonstrates a lack of understanding of the core concept: 0%
- Any questions requiring written answers are answered with sufficient detail

Due Date

March 18 at midnight MST

Assignment Exercises

1. Set-Up (5 pts)

Load in the `tidyverse` to get started. If you haven't downloaded the files for the week (found at the beginning of the lesson), you will need to do so.

2. Portal Data Review (25 points)

For this question, we are using the Portal data to review many of the `dplyr`, `tidyr`, and `ggplot2` functions we have learned so far.

Load the 3 dataframes below into R using `read_csv()`.

- `surveys.csv`
- `species.csv`
- `plots.csv`

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
surveys <- read_csv("surveys.csv")
```

```
## Rows: 35549 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): species_id, sex
## dbl (7): record_id, month, day, year, plot_id, hindfoot_length, weight
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
species <- read_csv("species.csv")
```

```
## Rows: 54 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): species_id, genus, species, taxa
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
plots <- read_csv("plots.csv")
```

```
## Rows: 24 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): plot_type
## dbl (1): plot_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a. Create a data frame with only data for the `species_id` `DO`, with the columns `year`, `month`, `day`, `species_id`, and `weight`.

```
question_2a <- surveys %>%
  filter(species_id == "DO") %>%
  select(year, month, day, species_id, weight)
```

b. Create a data frame with only data for species IDs `PP` and `PB` and for years starting in 1995, with the columns `year`, `species_id`, and `hindfoot_length`, with no null values for `hindfoot_length`.

```
question_2b <- surveys %>%
  filter(species_id =="PP" | species_id =="PB") %>%
  filter(year >= 1995) %>%
  select(year, species_id, hindfoot_length) %>%
  filter(!is.na(hindfoot_length))
```

c. Create a data frame with the average `hindfoot_length` for each `species_id` in each `year` with no null values.

```
question_2c <- surveys %>%
  group_by(species_id, year) %>%
  filter(!is.na(hindfoot_length)) %>%
  select(species_id, year, hindfoot_length) %>%
  mutate(mean_hf = mean(hindfoot_length, na.rm=T))%>%
  arrange(species_id) %>%
  select(!hindfoot_length)
```

d. Create a data frame with the `year`, `genus`, `species`, `weight` and `plot_type` for all cases where the `genus` is `"Dipodomys"`.

```
question_2d <- surveys %>%
  filter(species_id == "DM" | species_id == "DX" | species_id == "DO" | species_id =="DS") %>%
  select(year, species_id, weight, plot_id)



question_2d <- question_2d %>%
  mutate(plot_type = ifelse(plot_id == 2, "Control", 0)) %>%
  mutate(plot_type = ifelse(plot_id == 1, "Spectab exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 9, "Spectab exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 4, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 8, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 11, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 12, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 14, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 17, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 22, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 3, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 15, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 19, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 21, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 6, "Short-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 13, "Short-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 18, "Short-term Krat Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 20, "Short-term Krat Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 5, "Rodent Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 7, "Rodent Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 10, "Rodent Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 16, "Rodent Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 23, "Rodent Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 24, "Rodent Exclosure", plot_type)) %>%
  select(year, species_id, weight, plot_type) %>%
  mutate(genus = "Dipodomys") %>%
  mutate(species = ifelse(species_id == "DS", "spectabilis", 0)) %>%
  mutate(species = ifelse(species_id == "DM", "merriami", species)) %>%
  mutate(species = ifelse(species_id == "DX", "sp.", species)) %>%
  mutate(species = ifelse(species_id == "DO", "ordii", species)) %>%
  select(year, genus, species, weight, plot_type)
```

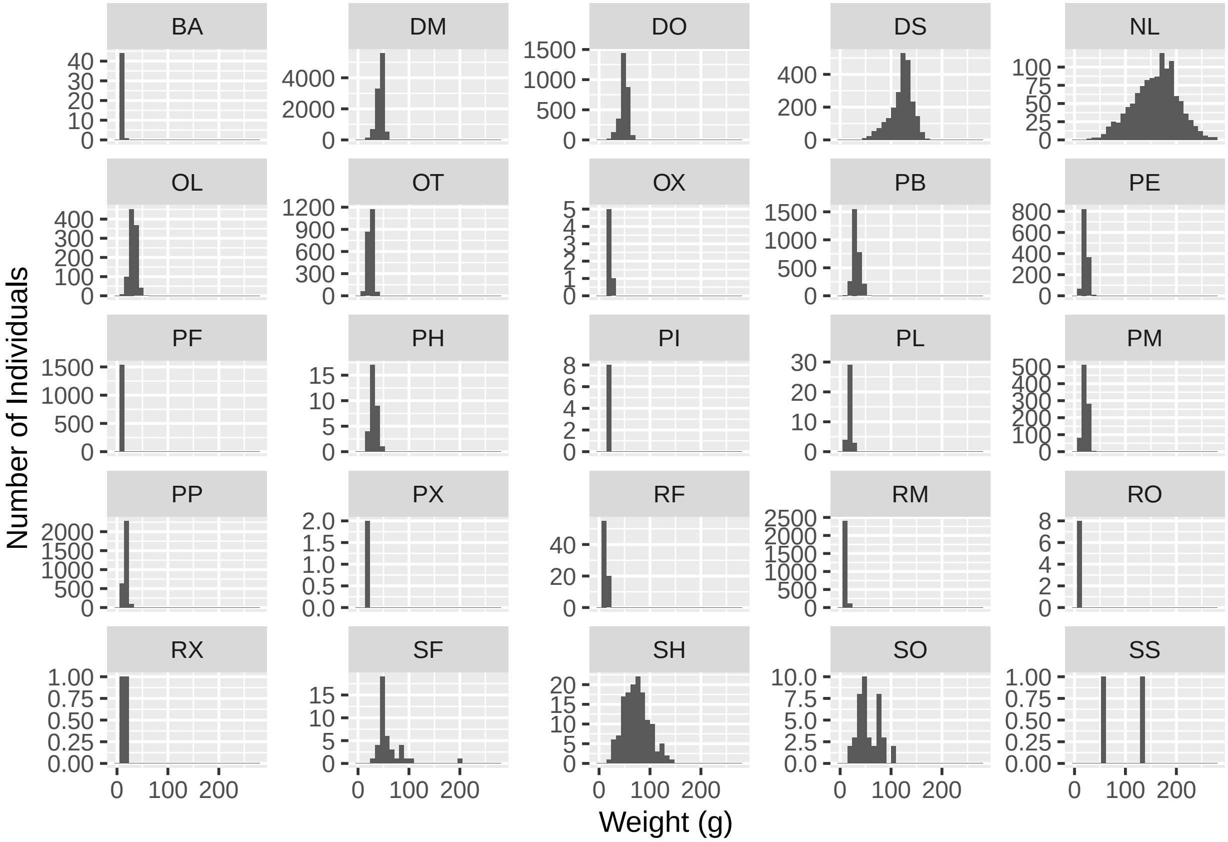
e. Make a scatter plot with `weight` on the x-axis and `hindfoot_length` on the y-axis. Use a `log10` scale on the x-axis. Color the points by `species_id`. Include good axis labels.

```
question_2e <- surveys %>%
  ggplot(aes(x=weight,y=hindfoot_length))+
  geom_point(aes(color=species_id))+
  scale_x_log10()+
  labs(x="Weight (g)", y="Hindfoot Length (mm)")
```

f. Make a histogram of weights with a separate subplot for each `species_id`. Do not include species with no weights. Set the `scales` argument in the `facet_wrap()` function to `"free_y"` so that the y-axes can vary. Include good axis labels.

```
surveys %>%
  filter(!is.na(weight)) %>%
  ggplot(aes(x=weight))+
  geom_histogram()+
  facet_wrap(~species_id, scales="free_y")+
  labs(x="Weight (g)", y= "Number of Individuals")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



g. (Challenge, optional) Make a plot with histograms of the weights of three species, `PP`, `PB`, and `DM`, colored by `species_id`, with a different facet (i.e., subplot) for each of three `plot_type`'s `Control`, `Long-term Krat Exclosure`, and `Short-term Krat Exclosure`. Include good axis labels and a title for the plot. Export the plot to a `png` file.

```
question_2g <- surveys %>%
  filter(species_id == "PP" | species_id == "PB" | species_id == "DM") %>%
  select(year, species_id, weight, plot_id)


question_2g <- question_2g %>%
  mutate(plot_type = ifelse(plot_id == 2, "Control", 0)) %>%
  mutate(plot_type = ifelse(plot_id == 1, "Spectab exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 9, "Spectab exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 4, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 8, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 11, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 12, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 14, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 17, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 22, "Control", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 3, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 15, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 19, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 21, "Long-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 6, "Short-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 13, "Short-term Krat Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 18, "Short-term Krat Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 20, "Short-term Krat Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 5, "Rodent Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 7, "Rodent Exclosure", plot_type)) %>%
  mutate(plot_type = ifelse(plot_id == 10, "Rodent Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 16, "Rodent Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 23, "Rodent Exclosure", plot_type))%>%
  mutate(plot_type = ifelse(plot_id == 24, "Rodent Exclosure", plot_type)) %>%
  select(species_id, plot_type, weight) %>%
  filter(plot_type =="Control" | plot_type == "Long-term Krat Exclosure" | plot_type == "Short-term Kra

question_2g <- question_2g %>%
  ggplot(aes(x=weight))+
  geom_histogram(aes(fill=species_id))+
  facet_wrap(~plot_type)+
  labs(title="Size distribution comparison across treatments", x= "Weight (g)", y= "Number of Individual
```

3. Megafaunal Extinction (35 points)

There were a relatively large number of extinctions of mammalian species roughly 10,000 years ago. To help understand why these extinctions happened scientists are interested in understanding if there were differences in the size of the species that went extinct and those that did not. You are going to reproduce the three main figures from one of the major papers on this topic Lyons et al. 2004.

You will do this using a large dataset of mammalian body sizes (`mammal-size-data-clean.txt`) that has data on the mass of recently extinct mammals as well as extant mammals (i.e., those that are still alive today).

a. Import the data into R. As with most real world data there are a some things about the dataset that you'll need to identify and address during the import process. Print out the structure of the resulting data frame.

```
mammal <- read_table("mammal-size-data-clean.txt", na = "NA")
```

```
##
```

```
## -- Column specification --------------------------------------------------------
## cols(
##   continent = col_character(),
##   status = col_character(),
##   order = col_character(),
##   family = col_character(),
##   genus = col_character(),
##   species = col_character(),
##   mass = col_double(),
##   reference = col_number()
## )
```

```
## Warning: 444 parsing failures.
## row col  expected    actual                         file
##   2  -- 8 columns 9 columns 'mammal-size-data-clean.txt'
##   3  -- 8 columns 9 columns 'mammal-size-data-clean.txt'
##  24  -- 8 columns 9 columns 'mammal-size-data-clean.txt'
##  25  -- 8 columns 9 columns 'mammal-size-data-clean.txt'
##  26  -- 8 columns 9 columns 'mammal-size-data-clean.txt'
## ... ... ......... ......... ............................
## See problems(...) for more details.
```

```
str(mammal)
```

```
## spc_tbl_ [5,731 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ continent: chr [1:5731] "AF" "AF" "AF" "AF" ...
##  $ status   : chr [1:5731] "extant" "extant" "extant" "extant" ...
##  $ order    : chr [1:5731] "Artiodactyla" "Artiodactyla" "Artiodactyla" "Artiodactyla" ...
##  $ family   : chr [1:5731] "Bovidae" "Bovidae" "Bovidae" "Bovidae" ...
##  $ genus    : chr [1:5731] "Addax" "Aepyceros" "Alcelaphus" "Ammodorcas" ...
##  $ species  : chr [1:5731] "nasomaculatus" "melampus" "buselaphus" "clarkei" ...
##  $ mass     : num [1:5731] 70000 52500 171002 28050 48000 ...
##  $ reference: num [1:5731] 60 63 63 60 75 60 1 2 -999 -999 ...
##  - attr(*, "problems")= tibble [444 x 5] (S3: tbl_df/tbl/data.frame)
##   ..$ row     : int [1:444] 2 3 24 25 26 27 28 29 30 32 ...
##   ..$ col     : chr [1:444] NA NA NA NA ...
##   ..$ expected: chr [1:444] "8 columns" "8 columns" "8 columns" "8 columns" ...
##   ..$ actual  : chr [1:444] "9 columns" "9 columns" "9 columns" "9 columns" ...
##   ..$ file    : chr [1:444] "'mammal-size-data-clean.txt'" "'mammal-size-data-clean.txt'" "'mammal-si
##  - attr(*, "spec")=
##   .. cols(
##   ..   continent = col_character(),
##   ..   status = col_character(),
##   ..   order = col_character(),
##   ..   family = col_character(),
##   ..   genus = col_character(),
##   ..   species = col_character(),
##   ..   mass = col_double(),
##   ..   reference = col_number()
##   .. )
```

b. Create a plot showing histograms of masses for mammal species that are still present and those that went extinct during the Pleistocene (`extant` and `extinct` in the `status` column). There should be one sub-plot for each continent and that sub-plot should show the histograms for both groups as a stacked histogram. To match the original analysis don't include islands (`Insular` and `Oceanic` in the `continent` column) and or the continent labeled `EA` (because `EA` had no species that went extinct in the Pleistocene). Scale the x-axis logarithmically and use 25 bins to roughly match the original figure. Use good axis labels.

```
unique(mammal$continent)
```

```
## [1] "AF"      "AUS"     "EA"      "Insular" NA        "Oceanic" "SA"
```

```
mammal %>%
  filter(is.na(continent))
```

```
## # A tibble: 779 x 8
##    continent status  order        family         genus  species   mass reference
##    <chr>     <chr>   <chr>        <chr>          <chr>  <chr>    <dbl>     <dbl>
##  1 <NA>      extant  Artiodactyla Antilocapridae Antil~ americ~ 4.61e4        60
##  2 <NA>      extinct Artiodactyla Antilocapridae Capro~ minor   1   e4         1
##  3 <NA>      extinct Artiodactyla Antilocapridae Capro~ mexica~ 1.5 e4        24
##  4 <NA>      extinct Artiodactyla Antilocapridae Stock~ conkli~ 5.10e4        27
##  5 <NA>      extinct Artiodactyla Antilocapridae Stock~ onusro~ 5.50e4        27
##  6 <NA>      extinct Artiodactyla Antilocapridae Tetra~ shuleri 6   e4        24
##  7 <NA>      extant  Artiodactyla Bovidae        Bison  bison   5.79e5        60
##  8 <NA>      extinct Artiodactyla Bovidae        Bison  latifr~ 9   e5        58
##  9 <NA>      extinct Artiodactyla Bovidae        Bison  priscus 9   e5        24
## 10 <NA>      extinct Artiodactyla Bovidae        Booth~ bombif~ 3   e5        26
## # i 769 more rows
```

```
mammal <- mammal %>%
  mutate(continent = ifelse(is.na(continent), "NA", continent))


question_3b <- mammal %>%
  filter(status == "extinct" | status == "extant") %>%
  filter(!continent == "EA") %>%
  filter(!continent == "Insular") %>%
  filter(!continent == "Oceanic") %>%
  ggplot(aes(x=mass, fill = status))+
  geom_histogram(position = "stack", bins =25)+
  facet_wrap(~continent)+
  scale_x_log10()+
  labs(x="Mass (g)", y="Number of Individuals")
```

c. The 2nd figure in the original paper looks in more detail at two orders, *Xenarthra* and *Carnivora*, which showed extinctions in North and South America. Create a figure similar to the one in Part 2, but that shows 4 sub-plots, one for each order on each of the two continents. Still scale the x-axis logarithmically, but use 19 bins to roughly match the original figure.

```
question_3c <- mammal %>%
  filter(order == "Xenarthra" | order == "Carnivora") %>%
  filter(status == "extinct" | status == "extant") %>%
  filter(continent == "NA" | continent == "SA") %>%
  ggplot(aes(x=mass, fill = status))+
  geom_histogram(position = "stack", bins =19)+
  facet_grid(rows =vars(order), cols = vars(continent))+
  scale_x_log10()+
  labs(x="Mass (g)", y="Number of Species")
```

d. The 3rd figure in the original paper explores Australia as a case study. Australia is interesting because there is good data on both Pleistocene extinctions (`extinct` in the `status` column) and more modern extinctions occurring over the last 300 years (`historical` in the `status` column). Make single stacked histogram that compares the sizes of `extinct`, `extant`, and `historical` statuses. Scale the x-axis logarithmically and use 25 bins to roughly match the original figure. Use good axis labels.

```
question_3d <- mammal %>%
  filter(continent == "AUS") %>%
  filter(status == "extinct" | status == "extant" | status == "historical") %>%
  ggplot(aes(x=mass, fill = status))+
  geom_histogram(position = "stack", bins =24)+
  scale_x_log10()+
  labs(x="Mass (g)", y="Number of Species")
```

e. (Challenge, optional) Instead of excluding continent `EA` by name in your analysis (in part 2), modify your code to determine from the data which continents had species that went extinct in the Pleistocene and only include those continents.

4. Palmer Penguins (35 points)

In this question, we are going to take some raw data and recreate a clean dataset. This is from the `palmerpenguins` R package, which has body size measurements from 3 species of Antarctic penguins from 2007-2009. First, we need to load in the package and take a look at the clean version of the data that we are trying to recreate.

```
library(palmerpenguins)

# because the data is from a package, it doesnt automatically show up in our environment unless with us
penguins <- penguins
head(penguins)
```

```
## # A tibble: 6 x 8
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
## 1 Adelie  Torgersen           39.1          18.7               181        3750
## 2 Adelie  Torgersen           39.5          17.4               186        3800
## 3 Adelie  Torgersen           40.3          18                 195        3250
## 4 Adelie  Torgersen           NA            NA                  NA          NA
## 5 Adelie  Torgersen           36.7          19.3               193        3450
## 6 Adelie  Torgersen           39.3          20.6               190        3650
## # i 2 more variables: sex <fct>, year <int>
```

Now, let’s bring in the original 3 datasets that were used to create this cleaned version (`penguins`)

```
# Adelie penguin data from: https://doi.org/10.6073/pasta/abc50eed9138b75f54eaada0841b9b86
url_adelie <- "https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.219.3&entityid=00
adelie <- read_csv(url_adelie)
```

```
## Rows: 152 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): studyName, Species, Region, Island, Stage, Individual ID, Clutch C...
## dbl  (7): Sample Number, Culmen Length (mm), Culmen Depth (mm), Flipper Leng...
## date (1): Date Egg
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Gentoo penguin data from: https://doi.org/10.6073/pasta/2b1cff60f81640f182433d23e68541ce
url_gentoo <- "https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.220.3&entityid=e03
gentoo <- read_csv(url_gentoo)
```

```
## Rows: 124 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): studyName, Species, Region, Island, Stage, Individual ID, Clutch C...
## dbl  (7): Sample Number, Culmen Length (mm), Culmen Depth (mm), Flipper Leng...
## date (1): Date Egg
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Chinstrap penguin data from: https://doi.org/10.6073/pasta/409c808f8fc9899d02401bdb04580af7
url_chinstrap <- "https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.221.2&entityid=
chinstrap <- read_csv(url_chinstrap)
```

```
## Rows: 68 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): studyName, Species, Region, Island, Stage, Individual ID, Clutch C...
## dbl  (7): Sample Number, Culmen Length (mm), Culmen Depth (mm), Flipper Leng...
## date (1): Date Egg
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Problem breakdown (both a and b are graded for completion, not accuracy)

a. Start by breaking down the problem into plain language. This stage shouldn't include any specific functions but is allowing you to talk through the steps conceptually.

Take the individual species data sets and make the 'penguins' dataset. Adjust column names into the correct case, add a species column with extracted info from the 'Species' column within each individual species data set, select only the columns we're interested in, do some of join to get the tables together, rename 'Culmen' to 'bill' for easy understanding, extract year from 'Date egg' column to make Year column, change capitalization of the 'sex' column, Once we get the tables joined together we're gonna need to pivot longer,

b. Make some predictions about the order in which you will want to accomplish this task, including which functions you will likely be using.

1. Use bind rows to combine all three columns
2. Select what columns we want
3. Rename columns/fix case/capitalization 3a. Rename "Culmen" to bill 3b. Make the values in sex column lowercase
4. Extract Year out of Date Egg
5. Extract first word of species column into new simplified column
6. Fix the "." to an NA

Coding

c. Recreate the clean dataset (`penguins`). Below are some tips (in no particular order) that will likely be helpful along the way

   - There is one instance in the sex column of one of the species where an unknown sex is marked with a `.` instead of `NA`

- You do not need to match up data types exactly (character and factors are mostly interchangeable; same with integer, numeric, and double)
- The year column is derived from the `Date Egg` column in the original 3 dataframes
- Culmen is basically a fancy word for a bird's bill
- I've taught you multiple ways to pull out a specific part of a character string, and you can choose whichever you prefer. Additional helpful hints are that the regex for extracting the first word in a string is `'\\w*'`; there is also a function called `word()` that is part of the `stringr` package.

You will know that you have successfully completed the task at hand if you run the code `setdiff(your_clean_df, penguins)`, and the result has 0 rows.

The `setdiff()` function takes 2 dataframes and looks for any differences. The output is a dataframe with rows that do not match up. If you have 0 rows that don't match, that means all rows do match!

```r
penguins_clean <- bind_rows(chinstrap, gentoo, adelie)

penguins_clean <- penguins_clean %>%
  select(Species, Island, `Culmen Length (mm)`,  `Culmen Depth (mm)`, `Flipper Length (mm)`, `Body Mass

penguins_clean <- penguins_clean %>%
  rename(species = Species, island = Island, bill_length_mm = `Culmen Length (mm)`, bill_depth_mm = `Cul

penguins_clean$sex <- tolower(penguins_clean$sex)

penguins_clean <- penguins_clean %>%
  mutate(year = year(year))

penguins_clean <- penguins_clean %>%
  mutate(species = str_extract(species, "(\\w+)"))

penguins_clean <- penguins_clean %>%
  mutate(sex = ifelse(sex ==".", NA, sex))

setdiff(penguins_clean, penguins)
```

```
## # A tibble: 0 x 8
## # i 8 variables: species <chr>, island <chr>, bill_length_mm <dbl>,
## #   bill_depth_mm <dbl>, flipper_length_mm <dbl>, body_mass_g <dbl>, sex <chr>,
## #   year <dbl>
```

5. Preparing for Next Week

This is not graded homework, but I want us to be as prepared for next week as we can be.

That means that I would like you to have R, RStudio, and git installed on your personal computer and you have registered for a GitHub account (If you haven't done any of this before, it might be a bit of a slog).

To do so, I would like you to follow the steps in Chapters 4, 5, and 6 from Happy git with R. D2L will have some additional resources in next week's module description, as well. Please don't hesitate to reach out if you're getting stuck!