Assignment 9

Ellen Bledsoe

2025-04-18

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 25 at midnight MST

Assignment Exercises

For many of the exercises in this week's assignment, we will actually be using a lot of the code that you have already written for Assignment 8. This time, however, all of your file paths will be different…

1. Set-Up (5 pts)

Now that we are working outside of Posit Cloud, we will need to first *install* our packages onto your computer before we can load them with the `library()` function.

We use the `install.packages()` function to download the package from the internet and create a local copy. Unlike with the `library()` function, the package name needs to be inside quotation marks.

Insert a code chunk and (a) install and (b) load `palmerpenguins`. Since we already installed the `tidyverse` during the lesson, you do not need to install it again. However, you do need to *load* it again.

Once you've installed `palmerpenguins`, comment out that line of code.

Note! If you haven't already, make sure you are working in an RStudio Project!

Your project should have sub-directories for raw data, clean data, output, docs, and scripts. The naming convention for these folders is up to you.

```
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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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 errors
```

```
library(palmerpenguins)
```

2. Portal Data Paths Review (25 points)

For this question, we are going to be using some of the code you've already written for Assignment 8, Question 2.

Click on the links below to download all 3 of the Portal files: surveys, species, and plots.

Then, move those three files from your Downloads folder into your equivalent of the raw data folder (yours might have a slightly different name). This assignment file should be in your scripts folder.

Now, let's begin to code!

a. Load the 3 data frames (surveys, species, plots) into R using `read_csv()`. Make sure your paths are *relative*.
b. Copy the answers from Week 8 Assignment, Questions 2d-f, into the code chunk below.
c. Save the output of the code from 2d as a new dataframe. Then, save that resulting dataframe as a csv file in the clean data sub-directory using the `write_csv()` function.
d. Save the ggplots from 2e and 2f into the outputs folder using the `ggsave()` function.

```
# 2a
surveys <- read_csv('../data_raw/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('../data_raw/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.
```

```r
plots <- read_csv('../data_raw/plots.csv')
```

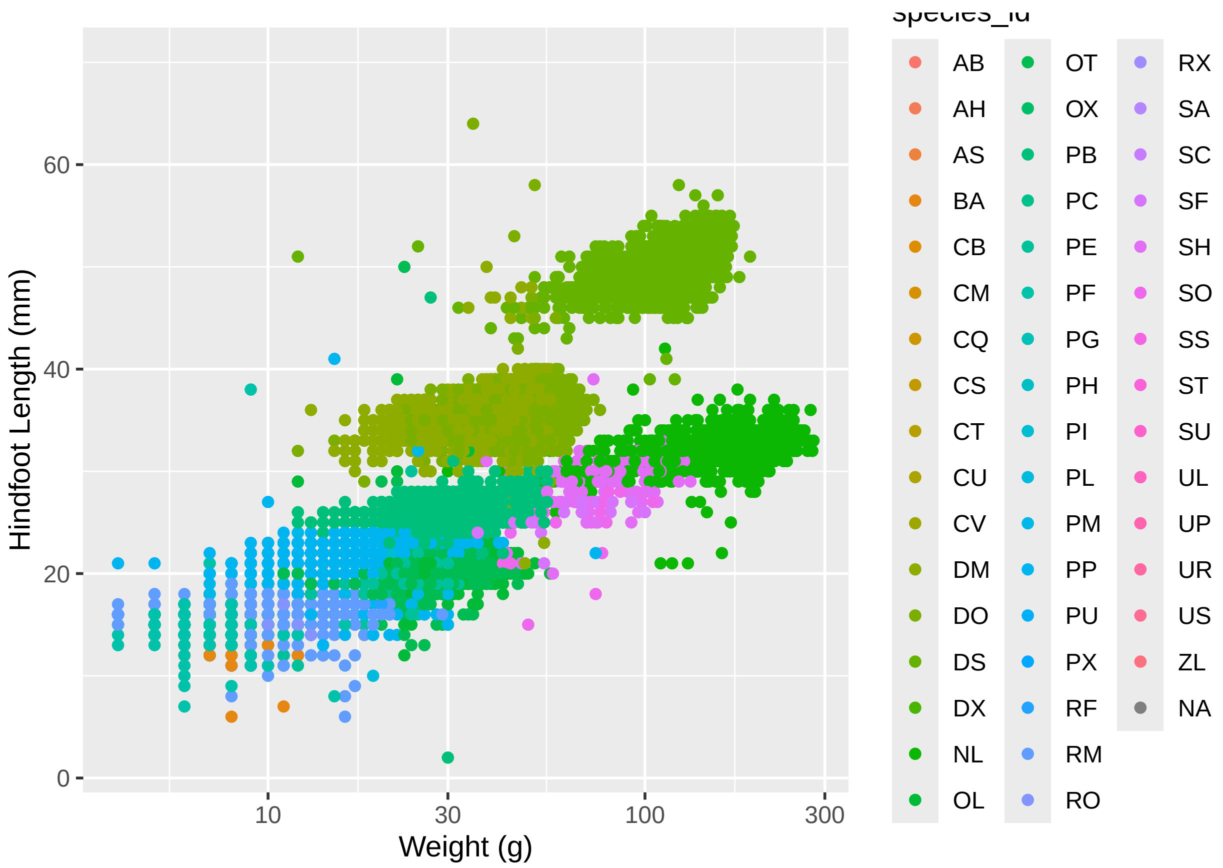
```
## Rows: 24 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): plot_type
## dbl (1): plot_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types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

```r
# d. Create a data frame with the `year`, `genus`, `species`, `weight` and `plot_type` for all cases wh
surveys %>%
  inner_join(species, by = "species_id") %>%
  inner_join(plots, by = "plot_id") %>%
  select(year, genus, species, weight, plot_type) %>%
  filter(genus == "Dipodomys")
```

```
## # A tibble: 16,167 x 5
##     year genus     species     weight plot_type
##    <dbl> <chr>     <chr>        <dbl> <chr>
##  1  1977 Dipodomys merriami        NA Control
##  2  1977 Dipodomys merriami        NA Rodent Exclosure
##  3  1977 Dipodomys merriami        NA Long-term Krat Exclosure
##  4  1977 Dipodomys merriami        NA Spectab exclosure
##  5  1977 Dipodomys merriami        NA Spectab exclosure
##  6  1977 Dipodomys spectabilis     NA Rodent Exclosure
##  7  1977 Dipodomys merriami        NA Rodent Exclosure
##  8  1977 Dipodomys merriami        NA Long-term Krat Exclosure
##  9  1977 Dipodomys merriami        NA Control
## 10  1977 Dipodomys merriami        NA Short-term Krat Exclosure
## # i 16,157 more rows
```

```r
# e. Make a scatter plot with `hindfoot_length` on the x-axis and `weight` on the y-axis. Color the poi
ggplot(data = surveys, mapping = aes(x = weight, y = hindfoot_length, color = species_id)) +
  geom_point() +
  scale_x_log10() +
  labs(x = "Weight (g)", y = "Hindfoot Length (mm)")
```

```
## Warning: Removed 4811 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
ggsave("../figures/plot_2e.png")
```

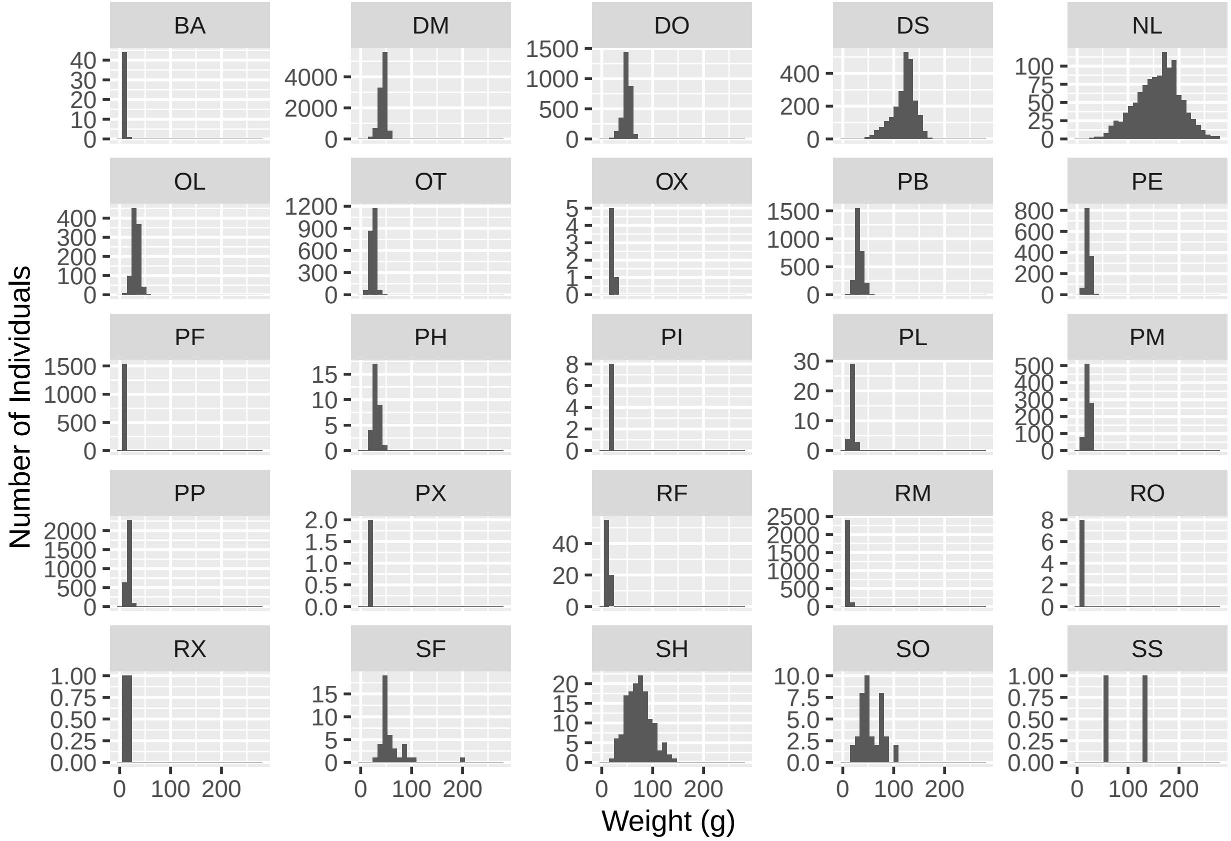
```
## Saving 6.5 x 4.5 in image

## Warning: Removed 4811 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# 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 to `"free_y"` so that the y-axes can vary.
# Include good axis labels.
surveys_with_weights <- filter(surveys, !is.na(weight))
surveys_with_weights
```

```
## # A tibble: 32,283 x 9
##    record_id month   day  year plot_id species_id sex   hindfoot_length weight
##        <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>           <dbl>  <dbl>
##  1        63     8    19  1977       3 DM         M                  35     40
##  2        64     8    19  1977       7 DM         M                  37     48
##  3        65     8    19  1977       4 DM         F                  34     29
##  4        66     8    19  1977       4 DM         F                  35     46
##  5        67     8    19  1977       7 DM         M                  35     36
##  6        68     8    19  1977       8 DO         F                  32     52
##  7        69     8    19  1977       2 PF         M                  15      8
##  8        70     8    19  1977       3 OX         F                  21     22
##  9        71     8    19  1977       7 DM         F                  36     35
## 10        74     8    19  1977       8 PF         M                  12      7
## # i 32,273 more rows
```

```r
ggplot(data = surveys_with_weights, mapping = 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`.
```



```r
ggsave("../figures/plot_2f.png")
```

```
## Saving 6.5 x 4.5 in image
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

3. Palmer Penguins and Path Files (15 points)

Like in Question 2 above, we will be recreating Question 4 from Week 8’s assignment but within our own R Project. As a reminder, this is the question that used the `palmerpenguins` data.

a. The code chunk below uses the `download.file()` function to go to a specific URL and then download the data at that URL. The location where the file is downloaded to is set by the `destfile` argument. Modify the path in the `destfile` argument for all three species datasets so that they are downloaded directly into your raw data folder.

```r
# Adelie penguin data
download.file(url = "https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.219.3&entit
              destfile = "../data_raw/adelie.csv")

# Gentoo penguin data
download.file(url = "https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.220.3&entit
              destfile = "../data_raw/gentoo.csv")

# Chinstrap penguin data
download.file(url = "https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.221.2&entit
              destfile = "../data_raw/chinstrap.csv")
```

b. Copy and then run the code that you wrote to combine the three above datasets and have the output match the penguins dataframe from the palmerpenguins dataframe.
c. Run the setdiff() function to make sure that your code worked (it shouldn't have any issues, but it is good to check!)
d. Save your version of the cleaned penguins data into your clean data folder.

```r
adelie <- read_csv("../data_raw/adelie.csv")
```

```
## 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.
```

```r
gentoo <- read_csv("../data_raw/gentoo.csv")
```

```
## 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.
```

```r
chinstrap <- read_csv("../data_raw/chinstrap.csv")
```

```
## 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.
```

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

penguins_clean <- penguins_raw %>%
  select(species = Species, island = Island, bill_length_mm = `Culmen Length (mm)`, bill_depth_mm = `Cul
  mutate(sex = na_if(sex, "."),
         species = str_extract(species, '\\w*'),
         sex = tolower(sex),
         year = lubridate::year(year))
```

```r
write_csv(penguins_clean, "../data_raw/penguins_clean.csv")
```

4. Add Version Control: Together in Class (15 points)

Add version control to this RProject that you are working in. We will do this together in class.

To get full points for this question, you will need to commit all of the current subdirectories and files to GitHub. This is particularly important, because I will be grading your assignment through the GitHub repo!

5. Create a Final Project Repository and RProject (20 points)

Follow the directions to create a new repository on GitHub in our class organization. This will be the repository that you will use for Final Project in the course.

Connect this GitHub repo with an RStudio Project, as we demonstrated in class. Once you have the R Project set up, add your subdirectories.

In D2L, you will find a file with a few questions about what dataset you are planning to use. If you don't have one in mind yet, don't worry! I'll provide a list with some options for you to explore.

Download the file from D2L and place it in the documents folder of your RProject.

Complete the questions in that document. Save the changes, commit the changes, and push the changes to GitHub.

Turning in Your Assignment

The way that you will be turning in assignments from now on is going to change now that we are no longer working in Posit Cloud.

Instead of turning in a PDF of your assignment, you will now be giving me a link to your GitHub repository. We will make sure all the permissions are take care of so that I have access.

It might be a good idea (though is not required) to add a bit of information in your README file for the GitHub repo to tell me where to look to find your assignment for any given week.