HW 06: Wrangling II

Graphical Analysis of Biological Data

By the end of this assignment, you should demonstrate an ability to

- mutate, summarize, and group data;
- Use a typical workflow to wrangle and plot data.

Click on any blue text to visit the external website.

This assignment has two parts.

Note: If you contact me for help or (better yet) open an issue in the public discussion forum, please include the code that is not working and also tell me what you have tried.

Preparation

- Open your `.Rproj` project file in RStudio.
- Create an `hw06` folder inside the same folder as your project file.
- Right-click to save this file as `feathers.csv` in your data folder. Make sure the file ends with `.csv` and not `.txt`.

Part 1

- Download this R Notebook file. Right-click to save the file as `<lastname>_hw06a.Rmd` inside the `hw06` folder. Make sure the file ends with `.Rmd` and not `.txt`.

Read the following sections. As you read the asigned sections, insert a code chunk in your notebook and recreate *every* example in that section, except for those I tell you in the notebook to skip. *Type the code!* In part 2, you will have to type the code from scratch so typing the code will help you remember it better. The point here is *learn* how to do this!

- Answer the questions in each section, except where I tell you to skip them. You will have to enter code chunks to answer the quwstions and also to write text outside of the code chunks. In other words, you will take advantage of R Notebook features.
- For each code chunk you add, write a brief description of what that chunk does. That includes examples and questions. Writing these descriptions will help you learn and understand the code.
- Some answers will require you to run code and enter text. Other questions will require only one or the other. The questions should give you the context you need.
- **Stage and commit regularly!** A good habit would be to stage and commit after you complete each section. You should also push each time but at least push your completed notebook to your remote repo.

Part 2: Wrangling II

Your task here is simple. Write code chunks that

- import data,
- wrangle the data as specified, and
- graph the data.

Then, describe the results briefly. Identify patterns and trends, potential outliers, and other results you find interesting. Think like the scientist you are!

You may need to review your Introduction to R coding exercise to meet some requirements, especially factors and how to select one column from a dataframe, and how to change one value to another.

Remember to put all the data files in your `data` folder.

A few things:

- Create a new notebook file called `<lastname>_06b.Rmd`. Edit the YAML file as you have for previous assignments.
- Load the `tidyverse` and `knitr` libraries.
- **Install the `smatr` package and then load the `smatr` library.**
- Define a variable with the path to your `data` folder. This should be the same data path you used last time.
- Make sure the feathers.csv file is in your `data` folder.
- Commit early. Commit often. Push regularly but at least push your completed assignment.

Darters in riffles

Import data

- Read in `darters.txt` like you did for the last assignment.
- Use `filter()` to remove `tetrazonum` and `zonale`.
- Do *not* delete the `minsub` column like you did in the previous assignment.

Wrangle data

The substrate was measured at each trap site. The relative proportion of the two most commmon substrate types (sand, fine gravel, small gravel, large gravel, and cobble) was estimated. The relative abundance of the most abundant substrate type was recorded as `majsub`. The less abundant substrate type was recorded as `minsub`. Together, `majsub + minsub = 1`. If they do not sum to 1, then there was a data recording error.

- Use `mutate()` to perform these wrangling tasks
  - Sum together the `majsub` and `minsub` into a new column called `total_substrate`. Does `total_substrate` sum to 1 for all observations?
  - Change the riffle values of `1` and `2` to `Riffle 1` and `Riffle 2`.
  - Change the length data from centimeters to millimeters. It is more common to use mm for small fishes.
  - Change the sex values of `f` and `m` to `Female` and `Male`.
- Only `total_substrate` needs a new column. The other changes should use their own columns.

Summarize data

Use `group_by()` and `summarize()` to summarize the mean length, depth, and velocity for each species for each riffle. *Save this summary to a different object* than the original data. You will need both in the following plot. Use `facet_wrap` on riffles like you did in the last assignment.

Table results

- Use Make a table with the summary means.

Graph data

Plot 1

- Make a plot that uses the `stat_summary()` technique shown in R4ds section 3.7, reason 3 (scroll down, just before 3.7.1 Exercises) but use `x = species` and `y = length`.

  This will plot a point to show the median length, and draw a line from the shortest (`min`) to the longest (`max`) length to show the range of lengths for each species.

- Remember that `ggplot2` uses a "layered grammar of graphics." Add a `geom_point()` layer to add the mean length of each species to the plot. Use a filled square shape. *You will have to use the summarized data set to get the mean for this geom.*

Note: If you do not get a point in the next plot showing the mean for *E. blennioides,* then you did something wrong. What was it?

- Turn the plot so that the species are listed on the left axis and length on bottom axis.
- Add `facet_wrap()` to separate the two riffles.
- Add `labs()` so that length includes the appropriate unit of measurement in parentheses and that the species axis is not labeled (`x = NULL`).

How does the relationship of the median, mean, and range change between the two riffles? In other words, how does the size distribution change between the two riffles?

Plot 2

You probably noticed that the location of the median is not centered between the minimum and maximum values, even though the median is the middle value.

You decide that you could display the data more effectively. You want to show the individual measurements and emphasize the mean over the median. You also decide you want to try a slightly different way of graphing the data.

- Make a new plot with a `geom_point()` layer of length for each species. Change the point color to a lighter shade in the range of `gray50` to `gray70`. The points should contrast against the overall gray background but they are not the main point of the graph.
- Add a `stat_summary()` layer with these arguments:
  - `fun.y = median`
  - `geom = "point"`
  - `size = 2.2`
- Add another `stat_summary()`. Change the y function to `mean`, size to 3, and add the `color = "15"maroon"` argument.
- Facet, label, and flip the graph as you did above.

Stress and corticosterones in birds

Beaugeard et al. 2018 (open access) studied the effects of stress from urbaniztion on House Sparrow (*Passer domesticus*).

Import data

- load the `smatr` library when you load `tidyverse`.
- Data file: `feathers.csv`.
- Are the data tidy?
- The data were recorded by a team of French scientists, so decimal numbers use a decimal comma instead of decimal point (e.g., 2,31 instead of 2.31). Use the `locale = locale(<argument>)` argument

in the `read_csv()` function to specify the decimal mark. Type `?locale` to read the help file. The only argument you need to supply to `locale()` is for the decimal mark.
- Their column names include spaces and units of measurement, generally not a good idea because you have to use backticks to reference the columns. Rename the columns when you import them, or rename them after you import the data using the `rename()` function. Use these names, in order from left to right:
  - capture_date, day, site, urban_score, cortF, cortB, sex, smi, tarsus, mass, basal_cort, stress_cort
- *Hint:* If you rename the columns when you read in the data, you will need to use the `skip` argument to remove the row of column names in the file.

Wrangle data

The authors use an index called the "scaled mass index" (SMI), included as a column. However, you will use R to recreate that value. The equation to calculate SMI for each individual (i) is,

$$\text{SMI}_i = M_i \times (L_0/L_i)^b,$$

where L_0 is the mean tarsus length of all birds in the sample, M_i and L_i are the mass and tarsus[1] length of each individual, and b is the slope estimate of the standardized major axis regression on the log-transformed mass and log-transformed tarsus length.

Fortunately, you do not need to know the details of the analysis. You can get the values you need with a few easy steps. Let's first get b, the slope of the regression.

- Calculate the mean tarsus length of all individuals in the data set, and store it in av ariable called `mean_tarsus`. You should know how to calculate the mean of all values in a column.
- Use `mutate()` to log-transform (`log()`) the body mass (`mass`) and the tarsus length (`tarsus`). Save these in new columns as `log_mass` and `log_tarsus`, respectively.
- Run the following function `major_axis <- sma(log_mass ~ log_tarsus, data = <data>)`. Substitute the name of your imported data for `<data>`. The `sma()` function calculates the standardized major axis regression.
- Apply the `summary()` funciton to `major_axis`.
  - The value you want for b is the slope of the estimate.
- Apply the `coef()` function to `major_axis`. The output is a named numeric vector with two elements. The value you want is the second element. Store that value in the variable `b`.

At this point, you now have the information you need to calculate "SMI" for each bird. L_0 = `mean_tarsus`, b = `b`, and M_i and L_i are the mass and tarsus columns. **Note:** Do *not* use the log-transformed values for this calculation.

- Use mutate to calculate SMI and store the values in a new column called `new_smi`. Compare your new column with the column you imported. Your values should be nearly identical, differing by no more than 0.1. *Hint (optional):* Use the `select()` function of `dplyr` to rearrange your columns so that `smi` and `new_smi` are side by side. Or, print them out together. **Note:** If your numbers are not correct, figure out where you made a mistake and correct it.

Summarie data

Next, summarize the basal and stress corticosterone data.

- Group by site, and calculate the mean and standard error of the mean (SE_Y) for `basal_cort` and `stress_cort`. Standard error of the mean is calculated as,

[1] The tarsus looks like part of the bird's leg but is homologous to the tarsal bones in the vertebrate foot.

$$SE_Y = \frac{s}{\sqrt{n}},$$

where s is the standard deviation and n is the number of individuals. You can calculate the standard deviation in R with the `sd()` function. Do not forgot to include `na.rm = TRUE` in the arguments for `mean()` and `sd()`, as shown in the text.

Graph data

Plot 1

- Make a boxplot of `new_smi` for each site. Do you see any clear differences among the four sites?

Plot 2 A common graph is to plot the means with error bars based on the standard errors of the means.

- Make a plot of the basal corticosterone concentration (ng/ml) for each site. Use `geom_point()`. Add a `geom_errorbar()` layer. Use the `?` function and the interwebs if you need, but here are a few hints:
  - x should be the sites,
  - `geom_errorbar()` needs `ymin` and `ymax` to draw the lower and upper ends of the error bars. Here, `ymin` should be one standard error below the mean and `ymax` should be one standard error above the mean. How could you do that? Good. Now do it.

Plot 3

I am not a fan of the Star Wars Tie Fighter look of the default error bars. The horizontal lines do not add any information about the data. In other words, they decrease the data-ink ratio.

Repeat the above plot but with two differences. Use the `mean_stress` and its standard error, and use `geom_linerange()` instead of `geom_errorbar`. Linerange and errorbar use the same arguments.

Graduate and Honors students

You must modify one of the above graphs with the following requirements:

- use size of 3 for the point located at the mean,
- color the point maroon,
- place the error bars behind the points so that a black line does not cur across the face of each point.
- Assume you like the Tie Fighter look but you do not want them so exaggerated. Size them to something much smaller but still present. The exact size depends on your tastes. **Note:** I encourage undergrads to attempt this but this does not count for or against your grade.

---

et Vóila