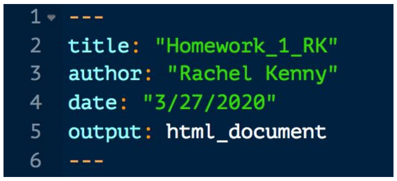
HOMEWORK #2

1. Prepare your document

- At this point, decisions about the organization of your files are up to you. You can either make a new project file and folder every time you work on a homework, or you can keep all of your homework assignments associated with the same project and folder. However, I would suggest creating a separate markdown document for each homework.
- Create your new rmarkdown file, and save it in the associated folder with your Rproj file and data. Remember to delete all the code that comes in the default file after line 10. Alternatively, you can just duplicate last week's homework file and delete all of the code after you load your packages and data, then edit the date and title in the first five lines of code (see screenshot above right for reference).



```
---
title: "Homework_1_RK"
author: "Rachel Kenny"
date: "3/27/2020"
output: html_document
---
```

- Load the agua chinon data and the oak data that we used last time.

2. Refer to specific columns

- Within the oak dataset:
  - Display two different ways that you can refer to the height column
  - Display how you would refer to the columns for location_uml through sampling group
  - Display how you would refer to columns 2, 5, 6, 7
  - Display how you would refer to every column except for 3 and 8

3. Change the names of columns & observations

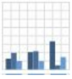
- Create a new dataframe with the oak data named oak_data2
  - Within the new dataframe, change all observations called "L_C" to "lower_closed", change all observations called "M_O" to "middle_open", and so forth for all six groups in the samplling group column (L_C, M_C, U_C, L_O, M_O, U_O). These are combinations of the experimental conditions (upper/middle/lower and open/closed)
  - Change the name of the seventh column to "supplementary_watering"

4. Create new columns

- Using the agua chinon data:
  - Create a new column using the $ operator that divides the pin number by 2
  - Create a new column using the mutate() function that returns a "1" for native species and a "0" for anything else (non-native or no native)
  - Create a new column (any method) that rounds the number for the column you made in the first step

5. **Create tables**
    - Recreate the “diversity1” table - filter to include only species from point-intercept data. Then tally the number of observations per species code, grouped by polygon id, transect, and native/non native. *(Hint - you can either create a dataframe with a filter in advance, or you can filter your data within the same code as your table creation by using the pipe operator)*
    - Recreate the “functional1” table - filter your data to include only native species from point-intercept data, and then tally the number of observations per functional group grouped by polygon id and transect.

6. **Practice with ggplots**
    - *Hint - in order to complete these successfully, you will need to refer to the data visualization cheat sheet. If it takes you more than an hour to figure it out feel free to give me a call :)*
    - Create appropriately labeled plots
        1) Side-by-side bar graph displaying the count of observations grouped by layer, then subdivided by functional group.

        2) Stacked bar graph (normalized height) displaying a count of observations grouped by polygon ID then subdivided by native/non-native

    - Customize:
        - ○ Change the colors
        - ○ Adjust the x-axis labels as needed
        - ○ Change the font size on any text elements
        - ○ Make the title bold