Assignment 1: Introduction

Lindsay Roth

OVERVIEW

This exercise accompanies the lessons in Hydrologic Data Analysis on introductory material.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document (marked with >).
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "FILENAME") prior to submission.

The completed exercise is due on 2019-09-04 before class begins.

Course Setup

1. Post the link to your forked GitHub repository below. Your repo should include one or more commits and an edited README file.

   Link: https://github.com/lhr12/Hydrologic_Data_Analysis

2. Complete the Consent Form in Sakai. You must choose to either opt in or out of the research study being conducted in our course.

Did you complete the form? (yes/no)

> yes

Course Project

3. What are some topics in aquatic science that are particularly interesting to you?

   ANSWER: hydrology, hydrogeomorphology/fluvial geomorphology, water quality/toxicology, harmful algal blooms

4. Are there specific people in class who you would specifically like to have on your team?

   ANSWER: If I'm working on a project related to HABs, it might be helpful to have members of my MP on my team (Caroline Watson, Tristen Townsend, Jake Greif), but if it is a different topic I'm open to working with anyone in the class.

5. Are there specific people in class who you would specifically *not* like to have on your team?

   ANSWER: No.

Data Visualization Exercises

6. Set up your work session. Check your working directory, load packages `tidyverse`, `dataRetrieval`, and `lubridate`. Set your ggplot theme as theme_classic (you may need to look up how to set your theme).

```
library(tidyverse)
#install.packages("dataRetrieval")
library(dataRetrieval)
library(lubridate)
mytheme <- theme_set(theme_classic())
```

7. Upload discharge data for the Eno River at site 02096500 for the same dates as we studied in class (2009-08-01 through 2019-07-31). Obtain data for discharge and gage height (you will need to look up these parameter codes). Rename the columns with informative titles. Imperial units can be retained (no need to change to metric).

```
EnoDischargeGage <- readNWISdv(siteNumbers = "02096500",
                     parameterCd = c("00060", "00065"), # discharge (ft3/s) & gage height (ft)
                     startDate = "2009-08-01",
                     endDate = "2019-07-31")

names(EnoDischargeGage)[4:7] <- c("Discharge", "Approval.Code.Discharge",
                              "Gage.Height","Approval.Code.Gage")
```

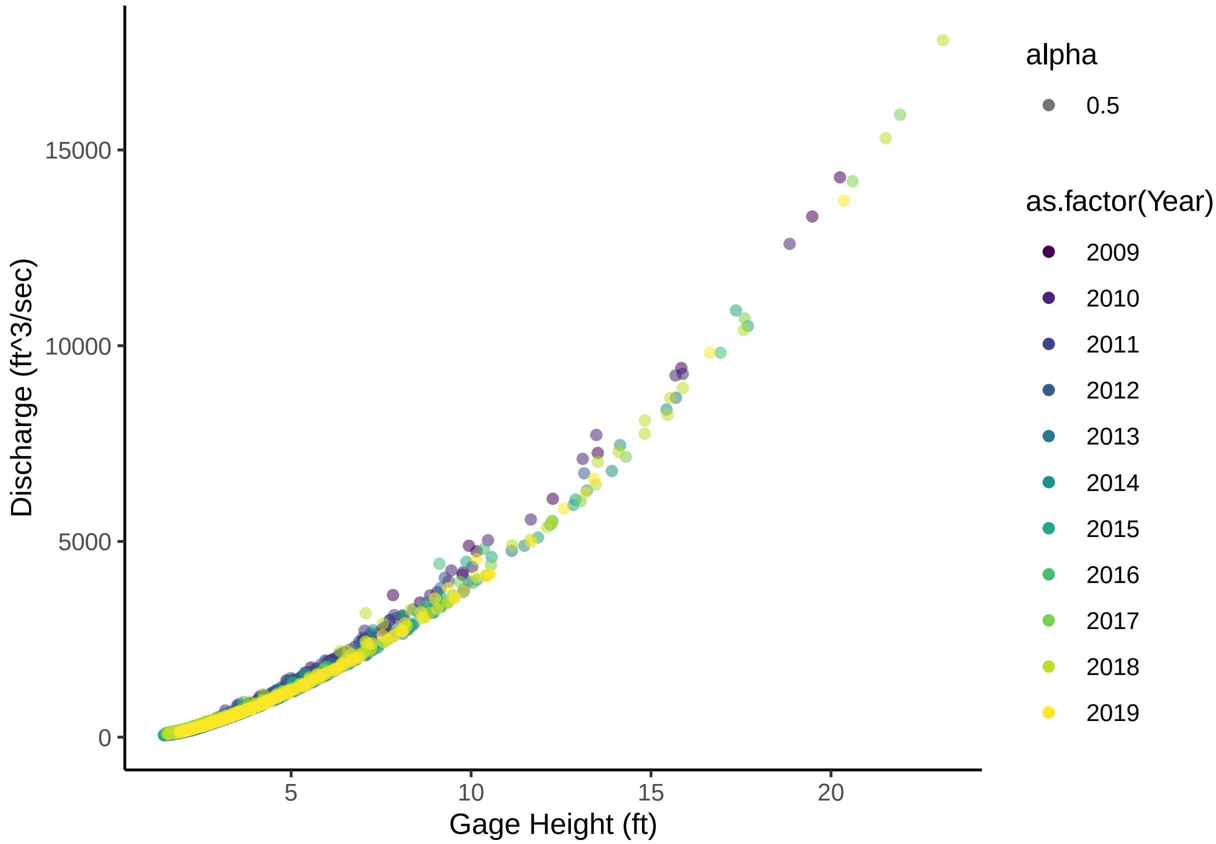
8. Add a "year" column to your data frame (hint: lubridate has a `year` function).

```
class(EnoDischargeGage$Date)

## [1] "Date"

EnoDischargeGage <- EnoDischargeGage %>%
  mutate(Year = year(Date))
```

9. Create a ggplot of discharge vs. gage height, with gage height as the x axis. Color each point by year. Make the following edits to follow good data visualization practices:

- Edit axes with units
- Change color palette from ggplot default
- Make points 50 % transparent

```
EnoPlotDG <- ggplot(EnoDischargeGage) +
  geom_point(aes(x = Gage.Height, y = Discharge,
               color = as.factor(Year), alpha = 0.5)) +
  scale_color_viridis_d()+
  labs(x = "Gage Height (ft)", y = "Discharge (ft^3/sec)")

print(EnoPlotDG)

## Warning: Removed 9 rows containing missing values (geom_point).
```



10. Interpret the graph you made. Write 2-3 sentences communicating the main takeaway points.

    ANSWER: As gage height increases, discharge also increases. The relationship is not linear, and discharge increases faster with increasing gage height. The highest discharge and gage height event occurred in 2017.

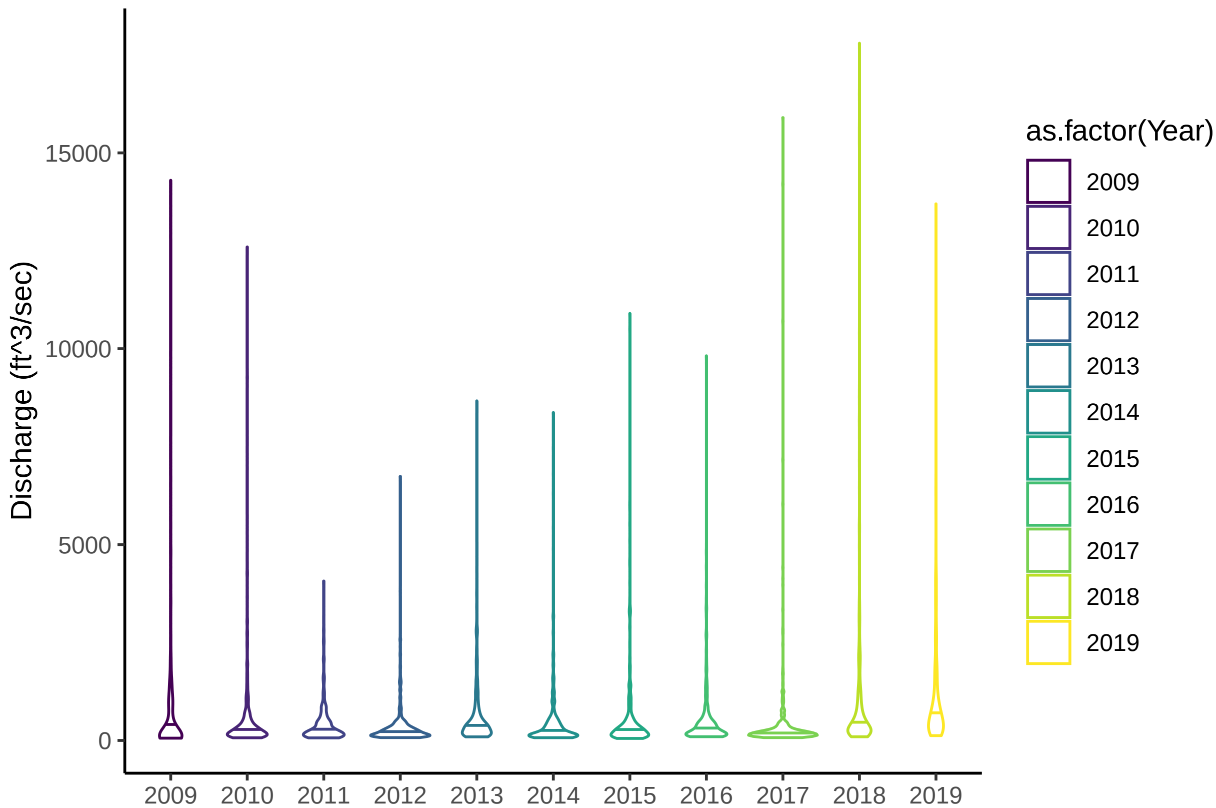
11. Create a ggplot violin plot of discharge, divided by year. (Hint: in your aesthetics, specify year as a factor rather than a continuous variable). Make the following edits to follow good data visualization practices:

- Remove x axis label
- Add a horizontal line at the 0.5 quantile within each violin (hint: draw_quantiles)

```
EnoViolin <- ggplot(EnoDischargeGage) +
  geom_violin(aes(x = as.factor(Year), y = Discharge, color = as.factor(Year)),
              draw_quantiles = 0.5) +
  labs(x = "", y = "Discharge (ft^3/sec)") +
  scale_color_viridis_d()


print(EnoViolin)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```



12. Interpret the graph you made. Write 2-3 sentences communicating the main takeaway points.

ANSWER: This graph shows that most of the discharge for the Eno River occurs far below 5000 cubic feet per second because the 50% quantile is low on the y-axis for all 11 years. There are a few extreme events in each year that act as outliers causing the violin graphs to stretch up the y-axis. 2011 had the smallest extreme event, while 2018 had the largest extreme event.