---

title: “Assignment 1: Introduction”
author: “Ethan Ready”
output: pdf_document
geometry: margin=2.54cm
editor_options:
chunk_output_type: console

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/eready22/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: Environmental Justice, Natural Resource Management

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

   ANSWER: I know Gabi and Theo outside of class and would be happy to work with either of them, but I'm also happy to work with people I don't know

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).

```
## get working directory
getwd()
```

```
## [1] "/Users/ethanready/Hydrologic_Data_Analysis/Assignments"
```

```
##load necessary packages
library(tidyverse)
library(dataRetrieval)
library(lubridate)
library(RColorBrewer)

## set ggplot theme
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).

```r
# Import data
EnoDischarge <- readNWISdv(siteNumbers = "02096500",
                     parameterCd = c("00060","00065"), # discharge (ft3/s)
                     startDate = "2009-08-01",
                     endDate = "2019-07-31")

# Renaming columns
names(EnoDischarge)[4:7] <- c("Discharge", "Approval.Code.Discharge", "Gage.Height", "Approv
head(EnoDischarge)
```

```
##   agency_cd  site_no       Date Discharge Approval.Code.Discharge
## 1      USGS 02096500 2009-08-01       186                       A
## 2      USGS 02096500 2009-08-02       129                       A
## 3      USGS 02096500 2009-08-03       123                       A
## 4      USGS 02096500 2009-08-04       118                       A
## 5      USGS 02096500 2009-08-05        84                       A
## 6      USGS 02096500 2009-08-06       112                       A
##   Gage.Height Approval.Code.Gage
## 1        2.13                  A
## 2        1.89                  A
## 3        1.85                  A
## 4        1.84                  A
## 5        1.66                  A
## 6        1.77                  A
```

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

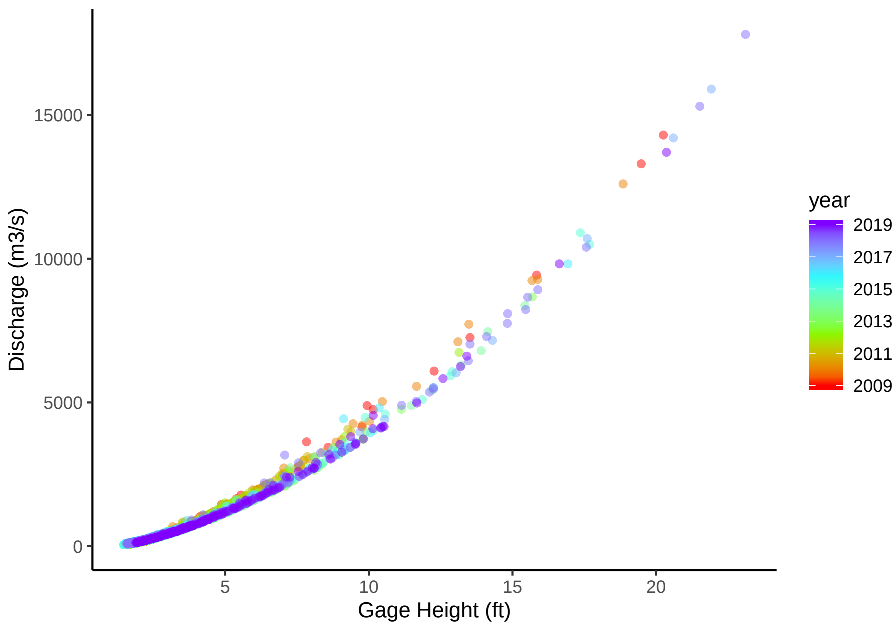
```r
yrs<- as.Date(EnoDischarge$Date)
EnoDischarge<-mutate(EnoDischarge, year=year(yrs))
```

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

```r
plot<-ggplot(EnoDischarge, aes(x=Gage.Height, y=Discharge, color=year))+
  geom_point(alpha=0.5)+
  scale_color_gradientn(colours=rainbow(4))+
  xlab("Gage Height (ft)")+
  ylab("Discharge (m3/s)")
plot
```

```
## 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: This summer while working with USGS I used a similar graph to estimate the total discharge of a stream based on the gage height, so we didn't have to take a discharge measurement every day. We needed the total discharge in order to calculate how much sea lamprey pheromone we would add to the stream to achieve our desired concentration. This graph shows a positive relation between Gage Height and Discharge. It appears to resemble y=e^x curvature.

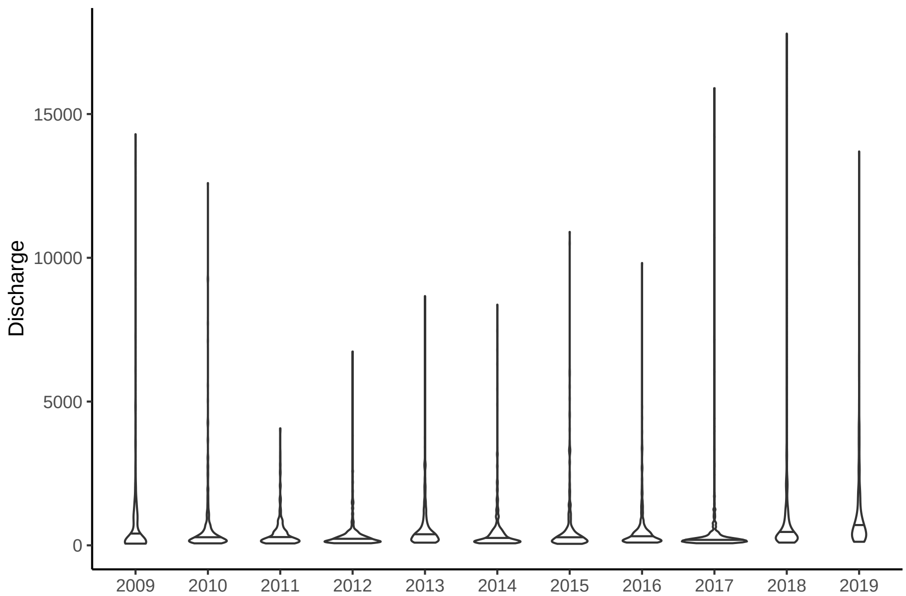
11. Create a ggplot violin plot of discharge, divided by year. (Hint: in your aesthetics, specify year as a factor rather than a continuous variable). Make the following edits to follow good data visualization practices:

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

```r
plt<-ggplot(EnoDischarge, aes(y=Discharge,x=as.factor(year)))+
  geom_violin(draw_quantiles = 0.5)+
  xlab("")
plt
```

```
## 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 plot shows the relative distribution of the total discharge for each year of the data. It shows that for every year, the large majority of total discharge measurements are well below the maximum measurement. This suggests a river with low daily flow that experiences large flooding events at least once, but not more than a few times, each year.