Assignment 1: Introduction

Haoyu Zhang

```
knitr::opts_chunk$set(echo = T, message = F)
knitr::opts_chunk$set(fig.width=6, fig.height=6, fig.align = "center")
```

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/forkhala/ENV-710-Data-Analysis.git

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: Water cycle; how plant communities are involved in the process; what are some potential impacts of climate change on hydrological cycle; or any topics concerning the role of forest in hydrology.

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

   ANSWER: No, I should be fine with anyone (hopefully).

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).

```
rm(list = ls())
getwd()
```

```
## [1] "D:/William/Duke/Study/EOS 722/Hydrologic_Data_Analysis/Assignments/Assignment 1"
```

```
setwd("D:/William/Duke/Study/EOS 722/Hydrologic_Data_Analysis/Assignments/Assignment 1")
require(tidyverse)
require(dataRetrieval)
require(lubridate)

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).

```
EnoDischarge <- readNWISdv(siteNumbers = "02096500",
                     parameterCd = c("00060","00065"),
                     # Discharge (ft3/s); Gage height, feet
                     startDate = "2009-08-01",
                     endDate = "2019-07-31")
head(EnoDischarge)
```

```
##   agency_cd  site_no       Date X_00060_00003 X_00060_00003_cd
## 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
##   X_00065_00003 X_00065_00003_cd
## 1          2.13                A
## 2          1.89                A
## 3          1.85                A
## 4          1.84                A
## 5          1.66                A
## 6          1.77                A
```

```
attr(EnoDischarge, "variableInfo")
```

```
##   variableCode           variableName              variableDescription
## 1        00060 Streamflow, ft&#179;/s Discharge, cubic feet per second
## 2        00065        Gage height, ft                Gage height, feet
##       valueType  unit options noDataValue
## 1 Derived Value ft3/s    Mean          NA
## 2 Derived Value    ft    Mean          NA
```

```
attr(EnoDischarge, "siteInfo")
```

```
##                   station_nm  site_no agency_cd timeZoneOffset
## 1 HAW RIVER AT HAW RIVER, NC 02096500      USGS         -05:00
##   timeZoneAbbreviation dec_lat_va dec_lon_va       srs siteTypeCd    hucCd
## 1                  EST   36.08722  -79.36611 EPSG:4326         ST 03030002
##   stateCd countyCd network
## 1      37    37001    NWIS
```

```
names(EnoDischarge)[c(4, 6)] <- c("Discharge", "Gageheight")
names(EnoDischarge)
```

```
## [1] "agency_cd"        "site_no"          "Date"
## [4] "Discharge"        "X_00060_00003_cd" "Gageheight"
## [7] "X_00065_00003_cd"
```

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

```
EnoDischarge$Year <- year(EnoDischarge$Date)
head(EnoDischarge)
```

```
##   agency_cd  site_no       Date Discharge X_00060_00003_cd Gageheight
## 1      USGS 02096500 2009-08-01       186                A       2.13
## 2      USGS 02096500 2009-08-02       129                A       1.89
## 3      USGS 02096500 2009-08-03       123                A       1.85
## 4      USGS 02096500 2009-08-04       118                A       1.84
## 5      USGS 02096500 2009-08-05        84                A       1.66
## 6      USGS 02096500 2009-08-06       112                A       1.77
##   X_00065_00003_cd Year
## 1                A 2009
## 2                A 2009
## 3                A 2009
## 4                A 2009
## 5                A 2009
## 6                A 2009
```

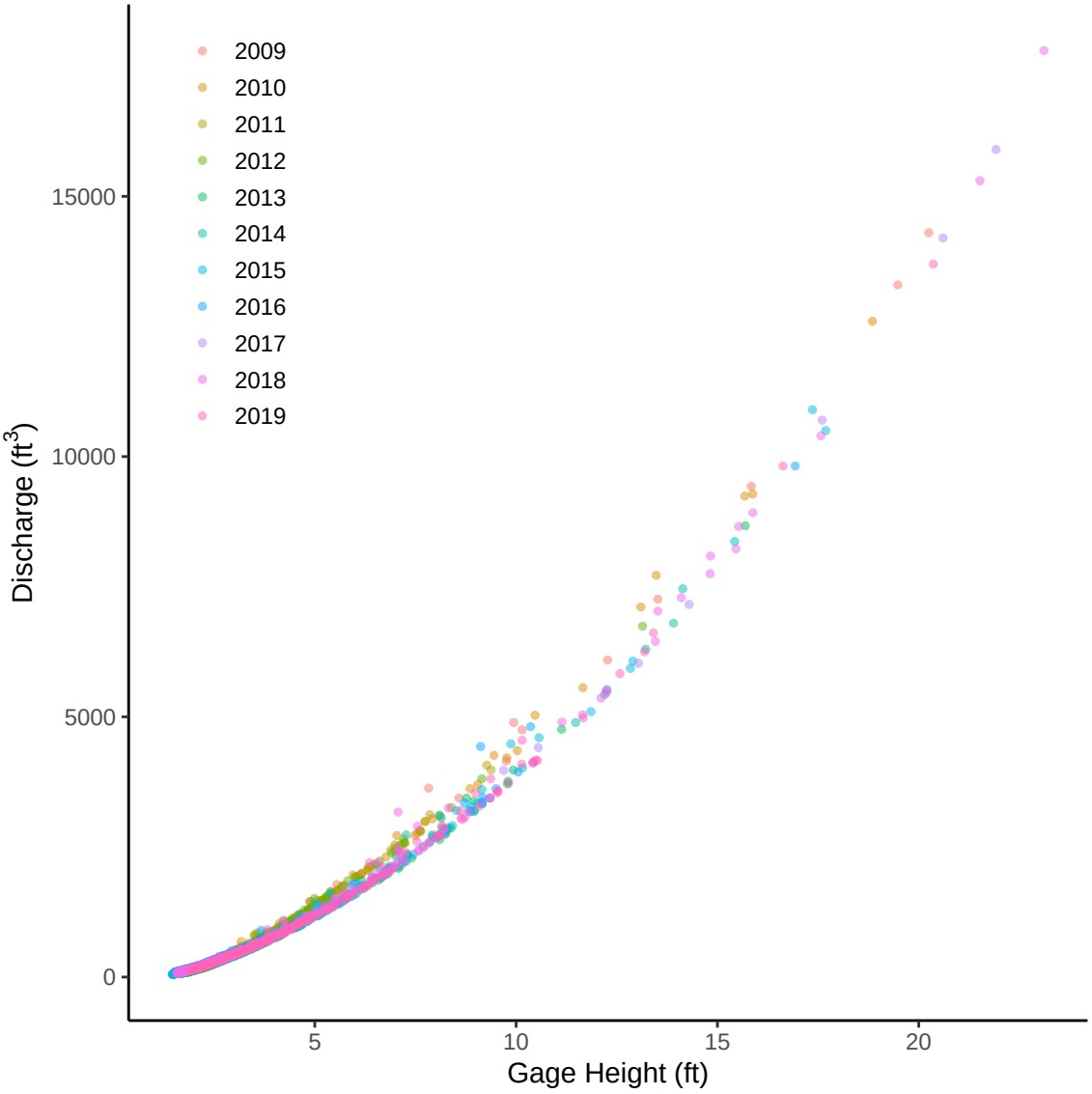
```
unique(EnoDischarge$Year)
```

```
##  [1] 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
```

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

```
ggplot(data = EnoDischarge, aes(x = Gageheight, y = Discharge)) +
  geom_point(na.rm = T, aes(color = as.factor(EnoDischarge$Year)), size = 1, alpha = 0.5 ) +
  xlab("Gage Height (ft)") + ylab(expression("Discharge (ft"^3*")")) +
  theme(
    legend.position = c(0.05, 1), legend.justification = c(0.05, 1),
    legend.title = element_blank(), legend.key.size = unit(0.5, "cm"),
    legend.box.margin = margin(0,0,0,0,"cm"))
```



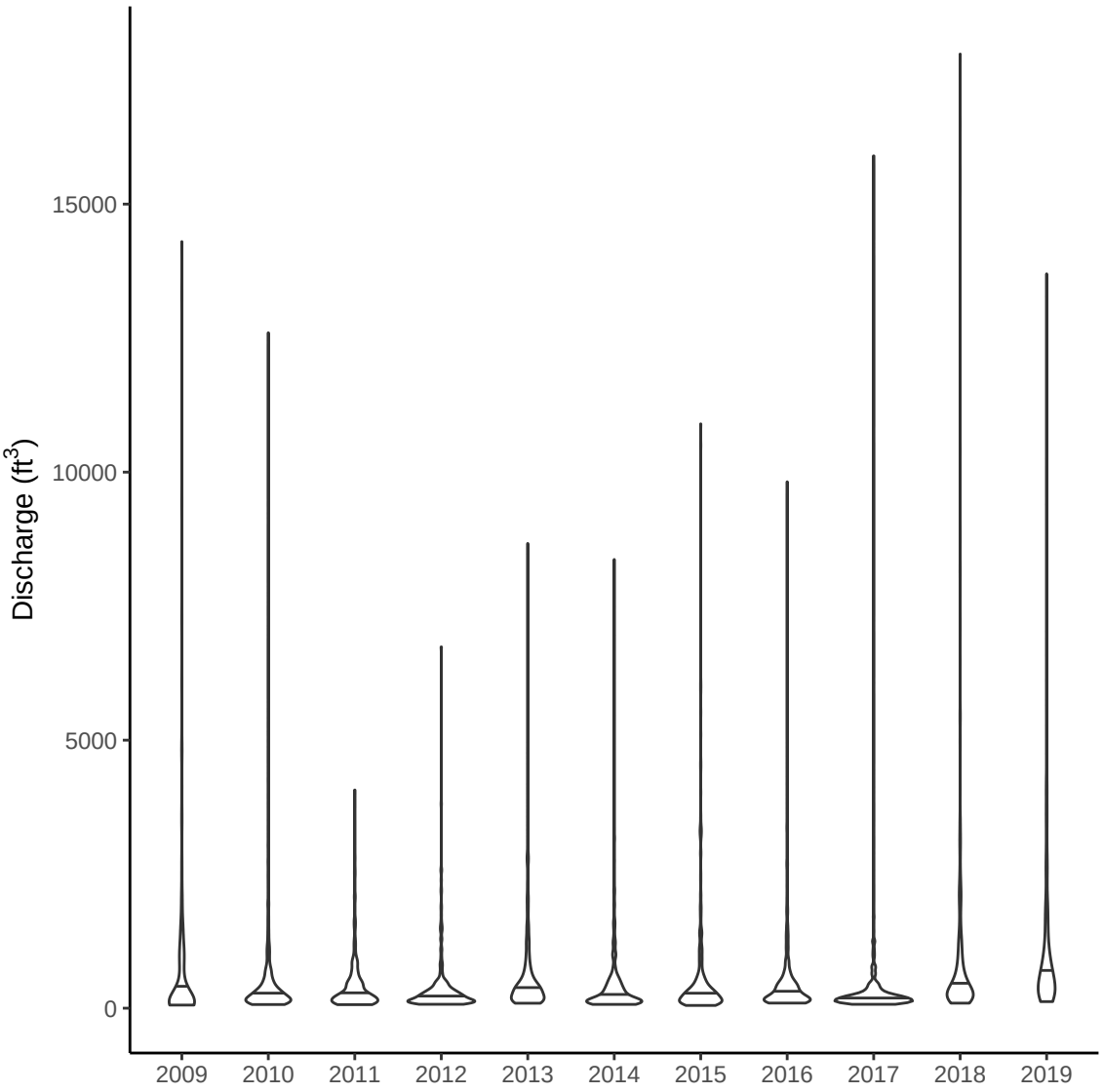
10. Interpret the graph you made. Write 2-3 sentences communicating the main takeaway points.

    ANSWER: Despite that the data span more than 10 years, there is a consistent and strong correlation between gage height and discharge rate. As gage height increases, discharge rate increases exponentially.

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:

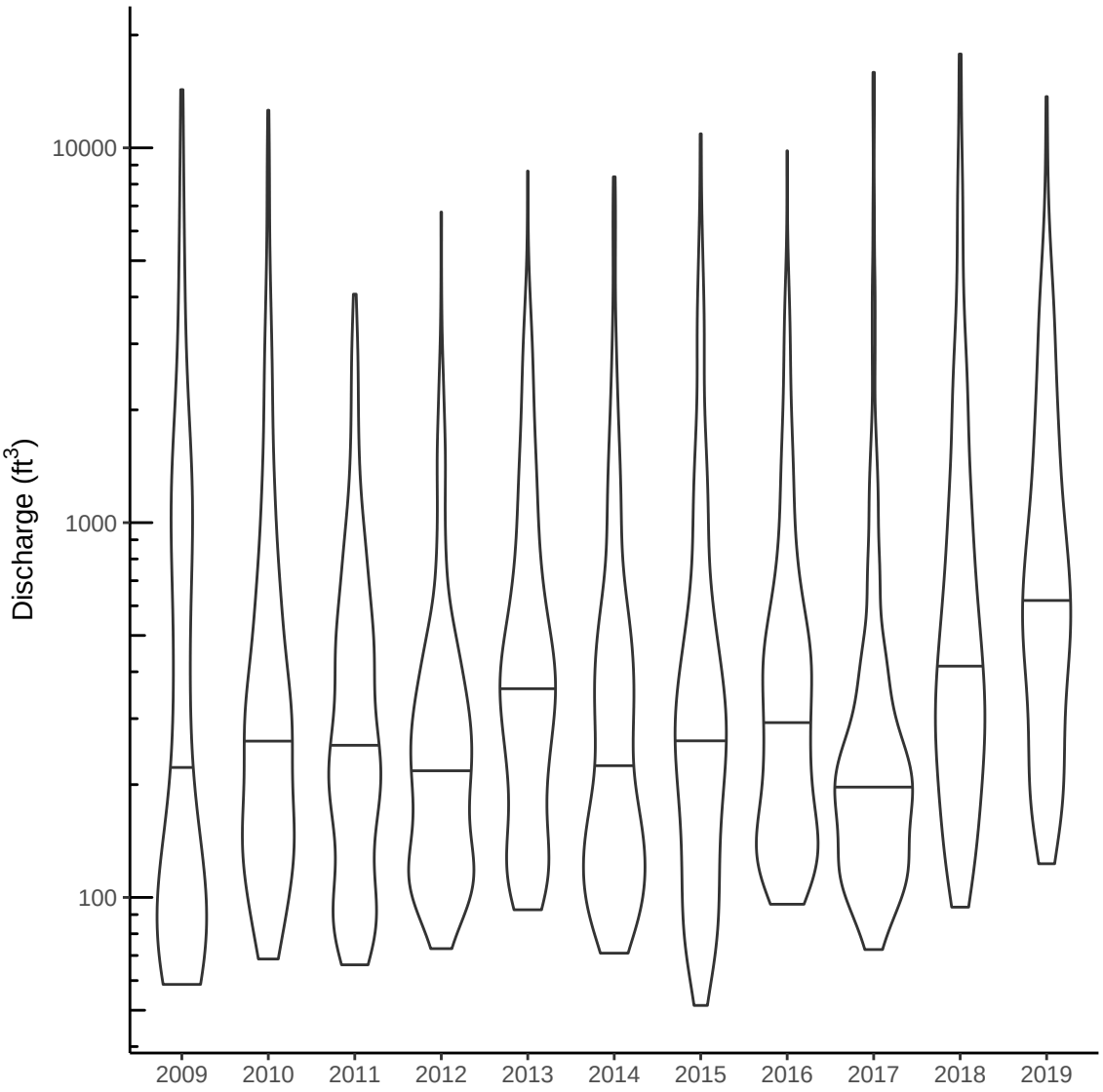
- Remove x axis label
- Add a horizontal line at the 0.5 quantile within each violin (hint: draw_quantiles)

```
ggplot(data = EnoDischarge, aes(x = as.factor(EnoDischarge$Yea), y = Discharge)) +
  geom_violin(na.rm = T, draw_quantiles = 0.5) +
  xlab(element_blank()) + ylab(expression("Discharge (ft"^3*")"))
```




```
# Original discharge extremely skewed; needs log transformation

ggplot(data = EnoDischarge, aes(x = as.factor(EnoDischarge$Yea), y = Discharge)) +
  geom_violin(na.rm = T, draw_quantiles = 0.5) +
  xlab(element_blank()) +
  scale_y_log10(name =  expression("Discharge (ft"^3*")")) +
  annotation_logticks(base = 10, scaled = T, side = "l")
```



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

    ANSWER: Discharge is highly positively skewed for all years, so log10 transformation was performed for better visualization. The median seems to remain relatively stable over the time period, although there might be a slightly upward trend. Variance within each year is high, but variance among years seems to be smaller based on log10 transformation.