Assignment 1: Introduction

Yixin Wen

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/yixin311/Hydrologic_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: contaminants level in lake or river system; seasonal change in river flux

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

   ANSWER: Xinchen Li, Mengfan Li

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).

```
getwd()
```

```
## [1] "/Users/yixinwen/Box/Duke/2019 Fall/Hydrologic Data Analysis/Hydrologic_Data_Analysis/Assignments
```

```
library(tidyverse)
library(dataRetrieval)
library(zoo)
library(ggrepel)
library(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).

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

# Renaming columns
names(EnoDischarge)[4:7]<-c("Discharge","Approval.Code1","Gage_height","Approval.Code2")
```

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

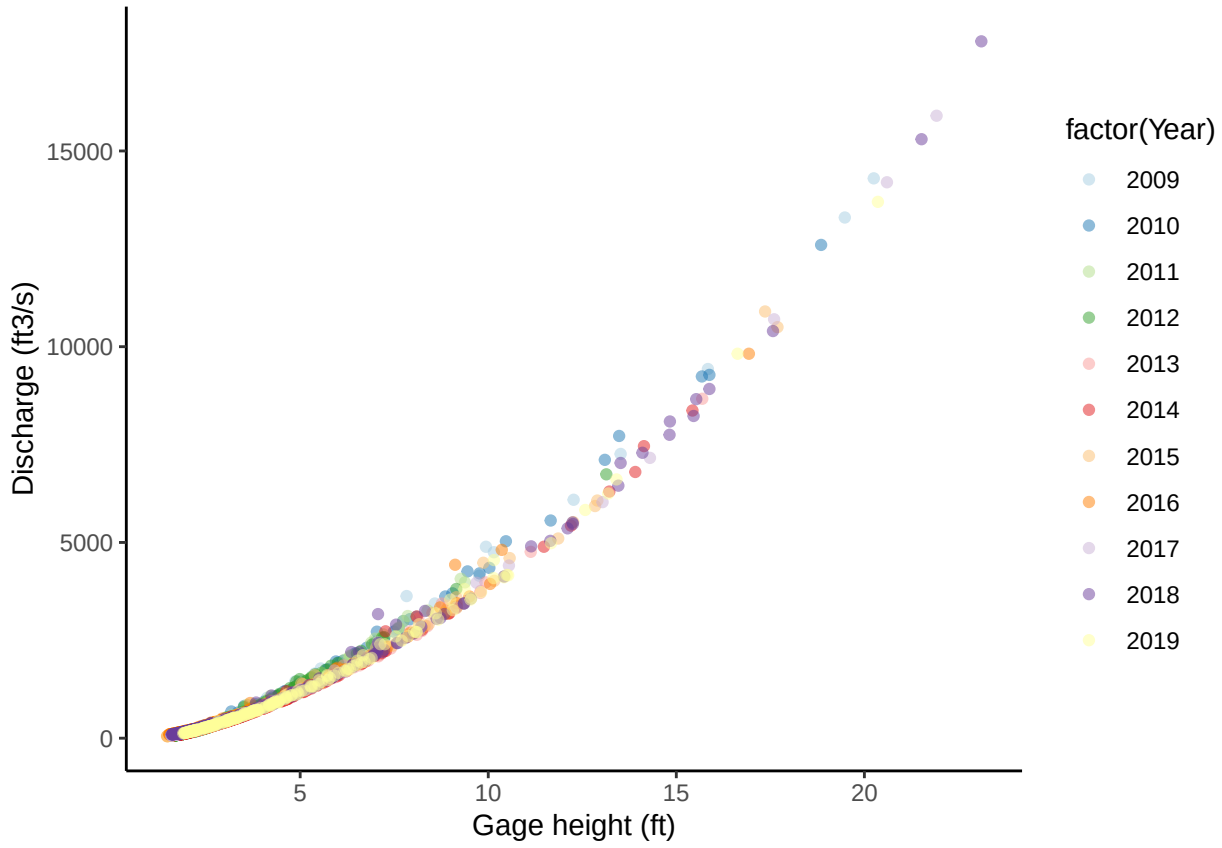
```
# add a "year" column to the original data frame
library(lubridate)
  new_column=EnoDischarge[,3] # apply the Date information to a new column
  Year=year(new_column) # extract the year of dates
  EnoDischarge <- cbind(EnoDischarge,Year) # add the Year column to EnoDischarge data frame
```

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

```
# Build a ggplot
EnoPlot1 <-
  ggplot(EnoDischarge)+
    geom_point(aes(x = Gage_height, y = Discharge, color = factor(Year)), alpha = 0.5) + # change the v
    xlab("Gage height (ft)") + # add units to axes
    ylab("Discharge (ft3/s)")+
    scale_color_brewer(palette = "Paired") # change the default color

print(EnoPlot1)
```

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



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

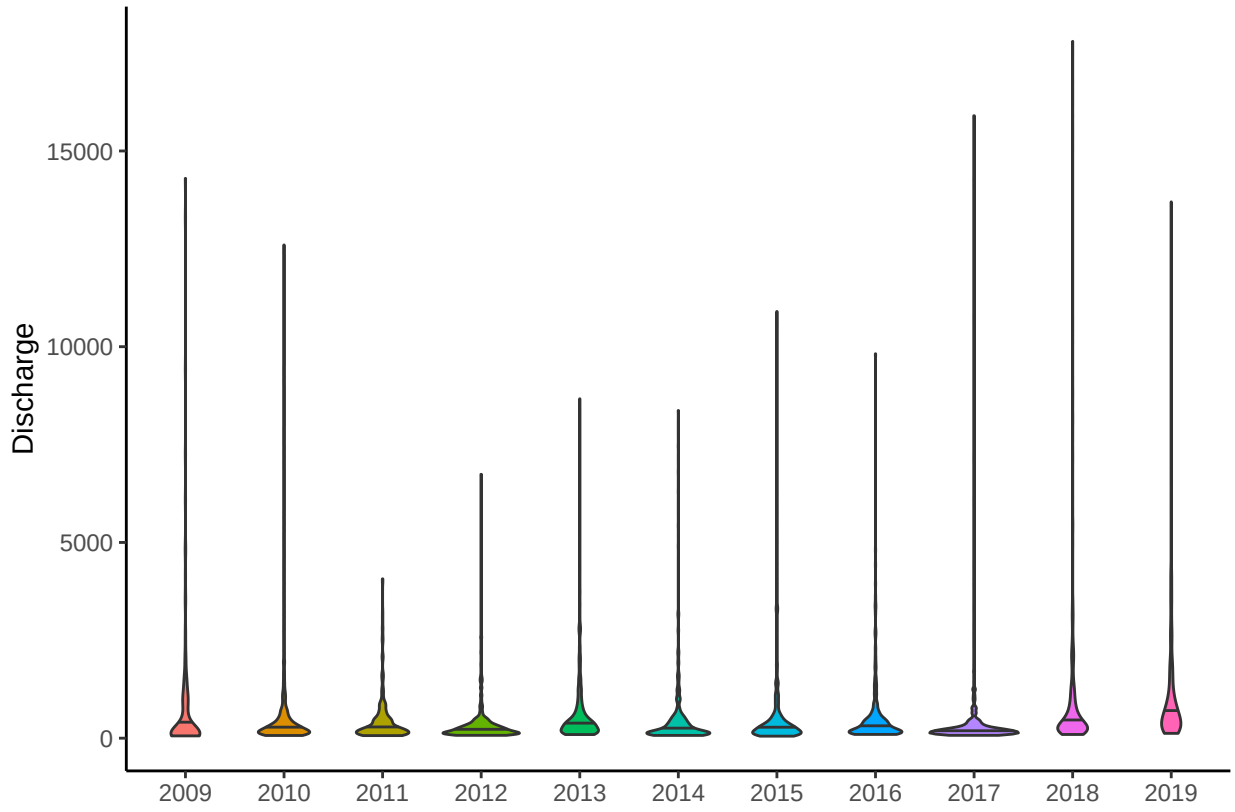
    ANSWER: From the plot, we can see that discharge is positively correlated with gage height. The value of discharge increases with the increase of gage height. 1. Since we need to color each point by year, we need to factorize the default continuous variable–"year", before coloring each point. 2. The range of alpha is between 0 to 1, which indicates completely transparent to opaque. 3. We can use scale_color_brewer function to change the default color palette in RColorBrewer package, but we need to notice that not every palette can meet the requirements. For example, in this problem we have 11 groups, which means we need to have 11 different colors. Thus we can use "Set3", "Paired", "Spectral", etc in this problem, while "Dark2", "Accent" cannot be used due to the lack of colors.

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)

```
# Build a violin plot

EnoDischarge$Year <- as.factor(EnoDischarge$Year) # factorize year in EnoDischarge
EnoPlot2 <-
  ggplot(EnoDischarge, aes(x = Year, y = Discharge, fill=Year))+
    geom_violin(draw_quantiles = 0.5)+ # add horizontal lines at 0.5 quantile
    xlab("")+ # remove x axis label
    theme(legend.position = "none")
print(EnoPlot2)
```

```
## 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: Violin plot is benefical in showning the highest value, lowest value and value distribution. From the violin plot, we can see that the median value of discharge in each year is below 1000.The small values of discharge have high frequency in each year. The highest value of discharge of the whole dataset is in 2018. 1. Since “year” is defaulted as a continous variable, we need to factorize it first to make divisions. 2. Draw_quantiles function can be used to draw horizontal lines of 0.25, 0.5, 0.75 quantiles. 3. In the violin plot, the axes labels are defaulted as the same in ase function, to remove axes lables, we can use xlab("“) or ylab(”").