Assignment 1: Introduction

Catherine Otero

OVERVIEW

This exercise accompanies the lessons in Water Data Analytics 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 completing your assignment, fill out the assignment completion survey in Sakai.

Having trouble? See the assignment's answer key if you need a hint. Please try to complete the assignment without the key as much as possible - this is where the learning happens!

Target due date: 2022-01-18

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/CatEOtero/Water_Data_Analytics_2022.git

Data Visualization Exercises

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

```
library(tidyverse)
library(dataRetrieval)
library(zoo)

theme_set(theme_classic())
```

3. Upload discharge data for the Eno River at site 02096500 for the same dates as we studied in class (2012-01-01 through 2021-12-31). Obtain data for discharge. Rename the columns with informative titles, as we did in class.

```r
EnoDischarge <- readNWISdv(siteNumbers = "02096500",
                     parameterCd = "00060", # discharge (ft3/s)
                     startDate = "2012-01-01",
                     endDate = "2021-12-31")

#rename the 4th and 5th data columns
names(EnoDischarge)[4:5] <- c("Discharge_cfs", "Approval.Code")

#View attribute info from data retrieval
attr(EnoDischarge, "variableInfo")
```

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

```r
  # note: imperial, not metric
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
```

4. Build a plot called EnoPlot2. Use the base plot we made in class and make the following changes:

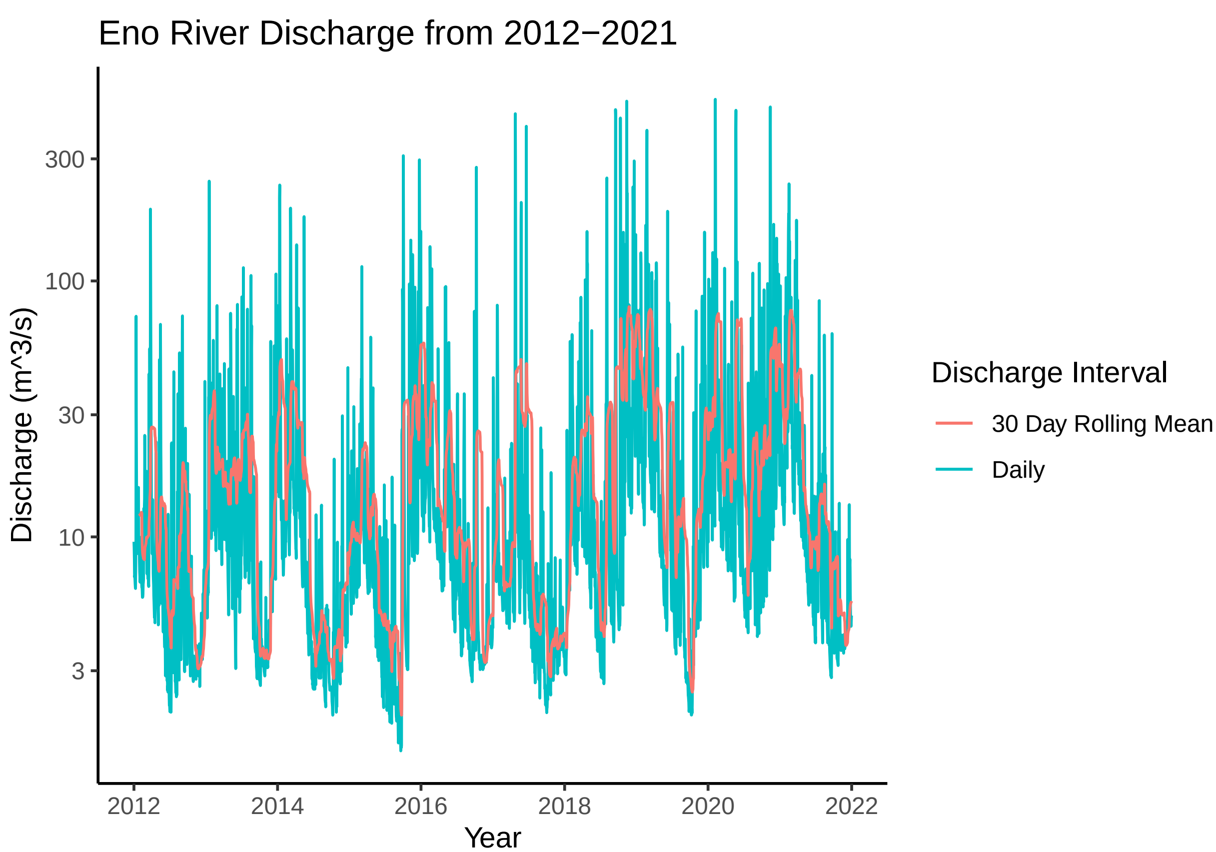
- Add a column to your data frame for discharge in meters cubed per second. hint: package dplyr in tidyverse includes a `mutate` function
- Add a column in your data frame for a 30-day rolling mean of the metric discharge. (hint: package dplyr in tidyverse includes a `mutate` function. hint: package zoo includes a `rollmean` function)
- Create two geom_line aesthetics, one for daily discharge (meters cubed per second) and one for rolling mean of discharge. Color these differently.
- Update your ggplot theme. I suggest "classic." (hint: https://ggplot2.tidyverse.org/reference/ggtheme.html)
- Update axis names
- Change the y axis from a linear to a log10 axis (hint: google "ggplot logged axis")
- Add a legend. (hint: Google "add legend two geom layers ggplot")

```r
#create function to convert cubic feet per second to cubic meters per second
CFS_TO_CMS <- function(x){
  CMS_converted <- x*0.02832
  return(CMS_converted)
}

#check if function is working
CMS <- CFS_TO_CMS(3)

CMS
```

```r
#mutate to create column with discharge in meters cubed per second. Also mutate with rollmean function

EnoDischarge_CMS <- EnoDischarge %>%
  mutate(Discharge_cms=CFS_TO_CMS(Discharge_cfs)) %>%
  mutate(Discharge_mean=rollmean(Discharge_cms, 30, align = 'right', fill=NA))

EnoDischarge_CMS
```

```r
#Plot the Discharge data
EnoPlot2 <- ggplot(EnoDischarge_CMS, aes(x = Date)) +
  geom_line(aes(y = Discharge_cms, color = "Daily")) +
  geom_line(aes(y = Discharge_mean, color = "30 Day Rolling Mean")) +
  labs(title = "Eno River Discharge from 2012-2021", x = "Year",
       y = "Discharge (m^3/s)") +
  scale_color_discrete(name = "Discharge Interval") +
  scale_y_continuous(trans = "log10")

EnoPlot2
```

```
## Warning: Removed 29 row(s) containing missing values (geom_path).
```



5. In what ways was the second plot a more effective visualization than the first?

   The second plot offered more detail and better labels, so a person knows what they are looking at without additional text blocks accompanying the plot. Showing the daily and 30 day average on one plot allows a person to compare the two also. With the increased detail of the second plot it's important to have clear labels to convey meaning and not just have a pretty uninterpretable graph.

6. What portions of the coding were challenging for you?

   ANSWER: The most challenging part of the code was figuring out how to add my own function into the mutate and then making sure all the plot parts came together.

7. Interpret the graph you made. What are the things you notice about within- and across-year variability, as well as the differences between daily values and 30-day rolling mean?

   ANSWER: Over the entire timeframe it appears that peak disharges have slightly increased. Within years, there is a lot of variability. The end of the year seems to have lower discharges. The 30-day rolling mean shows peak discharges follwing the daily peaks showing the trend for high disharges to add up over time.