Assignment 5: Data Visualization

Walker Grimshaw

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on data wrangling.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A04_DataWrangling.pdf") prior to submission.

The completed exercise is due on Thursday, 14 February, 2019 before class begins.

Set up your session

1. Set up your session. Upload the NTL-LTER processed data files for chemistry/physics for Peter and Paul Lakes (tidy and gathered), the USGS stream gauge dataset, and the EPA Ecotox dataset for Neonicotinoids.

2. Make sure R is reading dates as date format, not something else (hint: remember that dates were an issue for the USGS gauge data).

```
#1
getwd(); # Course project directory, but appears to change upon knitting
```

```
## [1] "/Users/walkergrimshaw/Documents/Duke/Courses/Spring_2019/Environmental_Data_Analytics/Assignment
```

```
# setwd("/Users/walkergrimshaw/Documents/Duke/Courses/Spring_2019/Environmental_Data_Analytics/Assignme
suppressMessages(library(tidyverse))
library(lubridate) # easy date manipulation
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(gridExtra) # multiple plots in a figure
```

```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(RColorBrewer)

## Peter Paul physchem processed
PP.chem.phys <-
  read.csv("../Data/Processed/NTL-LTER_Lake_ChemistryPhysics_PeterPaul_Processed.csv",
           header = T)
## Peter Paul Nutrients Processed
PP.chem.nutrients <-
  read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
           header = T)
## Peter Paul Nutrients Processed and gathered
PP.nutrients.gathered <-
  read.csv("../Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv",
           header = T)

## USGS Stream gauge
USGS.raw <- read.csv("../Data/Raw/USGS_Site02085000_Flow_Raw.csv", header = T)
## EPA Ecotox
EPA.Ecotox.Raw <- read.csv("../Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv",
                           header = T)

#2
class(EPA.Ecotox.Raw$Pub..Year); # only contains the year, so leave as integer
```

```
## [1] "integer"
```

```r
class(PP.chem.phys$sampledate); # factor
```

```
## [1] "factor"
```

```r
## change PeterPaul sampledate column to be formatted as date
PP.chem.phys$sampledate <- as.Date(PP.chem.phys$sampledate, format = "%m/%d/%y")
PP.chem.nutrients$sampledate <- as.Date(PP.chem.nutrients$sampledate, format = "%m/%d/%y")
PP.nutrients.gathered$sampledate <- as.Date(PP.nutrients.gathered$sampledate,
                                            format = "%Y-%m-%d")

class(USGS.raw$datetime); # factor
```

```
## [1] "factor"
```

```r
## format to date, then correct the dates formatted to the future
USGS.raw$datetime <- as.Date(USGS.raw$datetime, format = "%m/%d/%y")
## if date is in the future, correct format to 1900s
USGS.raw$datetime <- as.Date(ifelse(USGS.raw$datetime > Sys.Date(),
                                    format(USGS.raw$datetime, "19%y-%m-%d"),
                                    format(USGS.raw$datetime)))
```

Define your theme

3. Build a theme and set it as your default theme.

```
#3
WalkersTheme <- theme_bw(base_size = 12) +
  theme(legend.position = "top")

## Make my newly created theme the default theme for this assignment
theme_set(WalkersTheme)
```

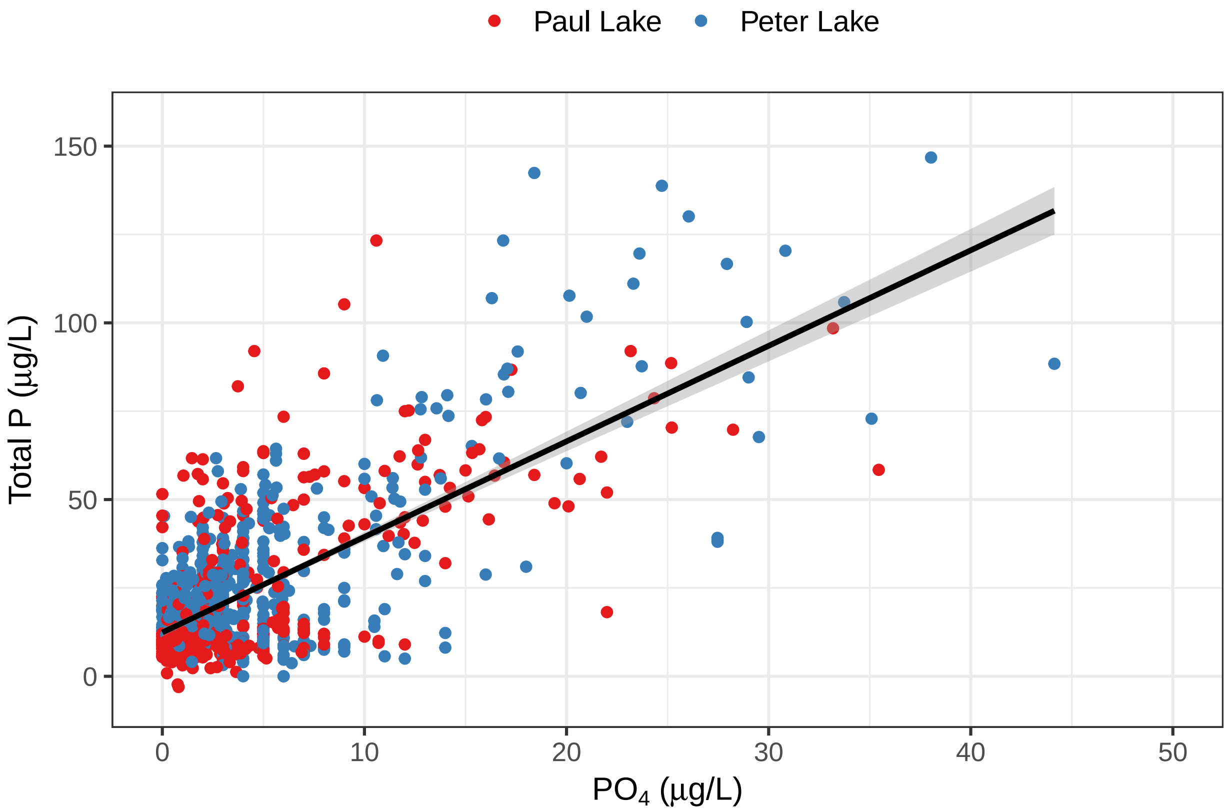
Create graphs

For numbers 4-7, create graphs that follow best practices for data visualization. To make your graphs "pretty," ensure your theme, color palettes, axes, and legends are edited to your liking.

Hint: a good way to build graphs is to make them ugly first and then create more code to make them pretty.

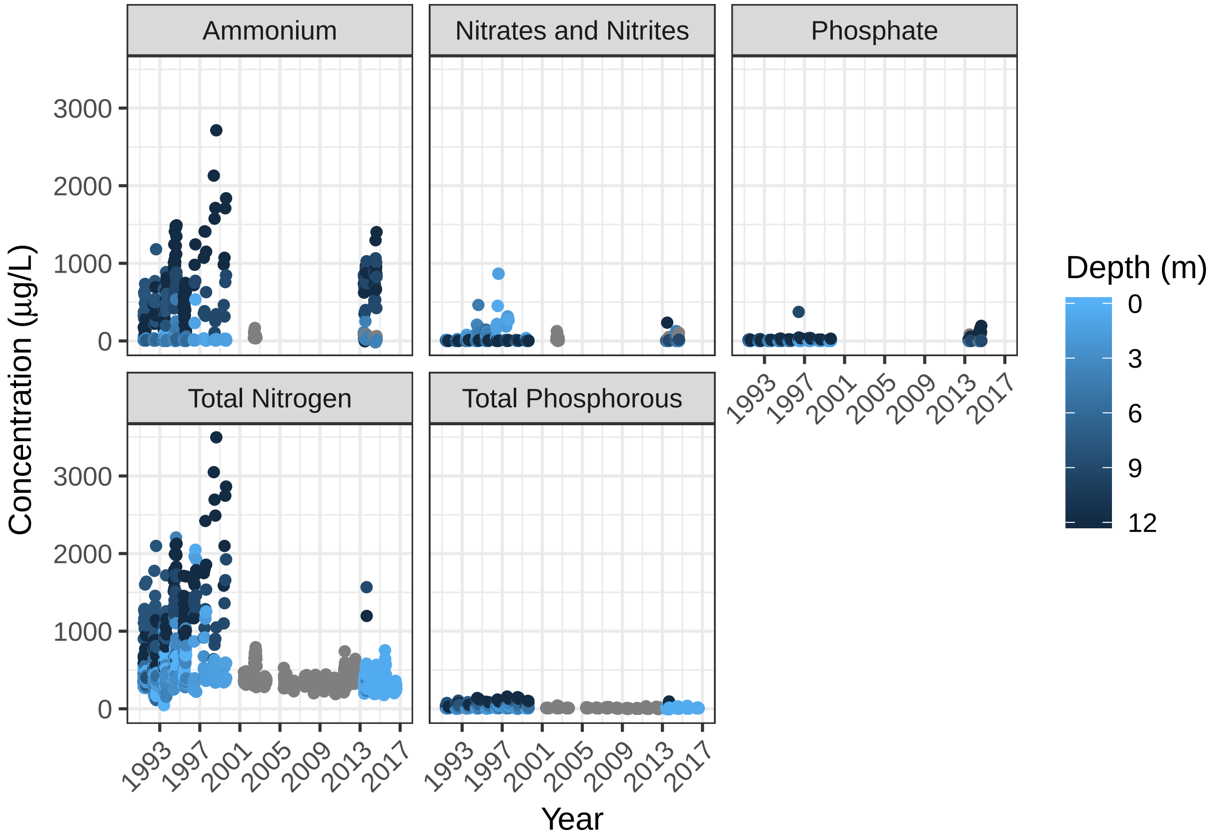
4. [NTL-LTER] Plot total phosphorus by phosphate, with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black.

```
#4

ggplot(PP.chem.nutrients, aes(x = po4, y = tp_ug)) +
  geom_point(aes(color = lakename)) +
  labs(x = expression(paste("PO"[4]* " (", mu, "g/L)")),
       y = expression(paste("Total P (", mu, "g/L)")),
       color = NULL) +
  geom_smooth(method = lm, color = "black") +
  ## non-default colors
  scale_color_brewer(palette = "Set1") +
  ## lower the x limit to 50 to get rid of what appears to be an error value at above 300
  xlim(0,50) +
  ## slightly increase size of legend text
  theme(legend.text=element_text(size=11))
```



5. [NTL-LTER] Plot nutrients by date for Peter Lake, with separate colors for each depth. Facet your graph by the nutrient type.

```
#5
## facet labels
nutrient.labels <- c(nh34 = "Ammonium", no23 = "Nitrates and Nitrites",
                     po4 = "Phosphate", tn_ug = "Total Nitrogen",
                     tp_ug = "Total Phosphorous")

ggplot(PP.nutrients.gathered) +
  geom_point(aes(x = sampledate, y = concentration, color = depth)) +
  ## dates from 1991 to 2016, break every 4 years
  scale_x_date(limits = as.Date(c("1991-01-01", "2016-12-31")),
               date_breaks = "4 years", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Year",
       y = expression(paste("Concentration (", mu, "g/L)")),
       color = "Depth (m)") +
  facet_wrap( ~ nutrient, labeller = labeller(nutrient = nutrient.labels)) +
  ## put legend on right to be intuitive with depth
  theme(legend.position = "right") +
  ## flip legend color so darker is deeper
  scale_color_continuous(trans = "reverse")
```

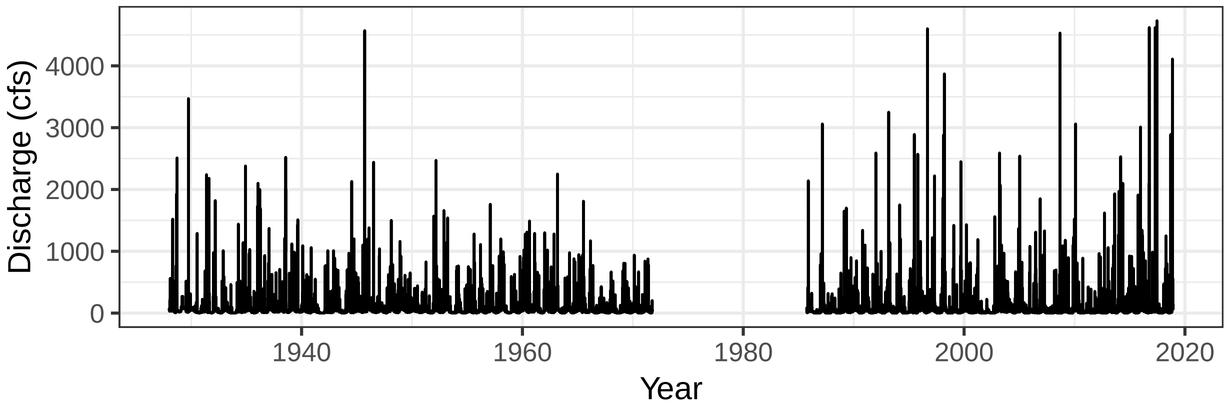


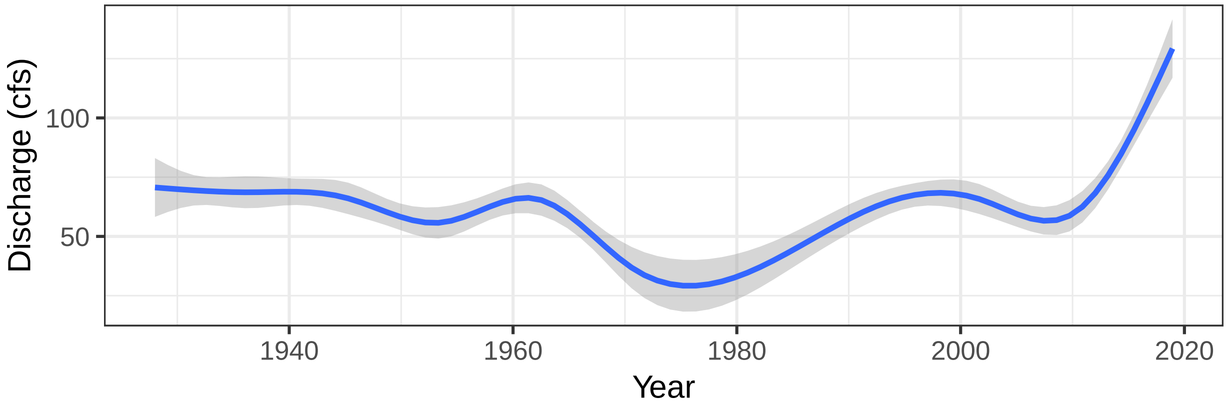
6. [USGS gauge] Plot discharge by date. Create two plots, one with the points connected with geom_line and one with the points connected with geom_smooth (hint: do not use method = “lm”). Place these graphs on the same plot (hint: ggarrange or something similar)

```
#6
## discharge in cubic feet per second

discharge.line <-
  ggplot(USGS.raw, aes(x = datetime, y = X165986_00060_00001)) +
  geom_line() +
  labs(x = "Year", y = "Discharge (cfs)")

discharge.smooth <-
  ggplot(USGS.raw, aes(x = datetime, y = X165986_00060_00001)) +
  geom_smooth() +
  labs(x = "Year", y = "Discharge (cfs)")

## print the two figures together
grid.arrange(discharge.line, discharge.smooth)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 5113 rows containing non-finite values (stat_smooth).
```



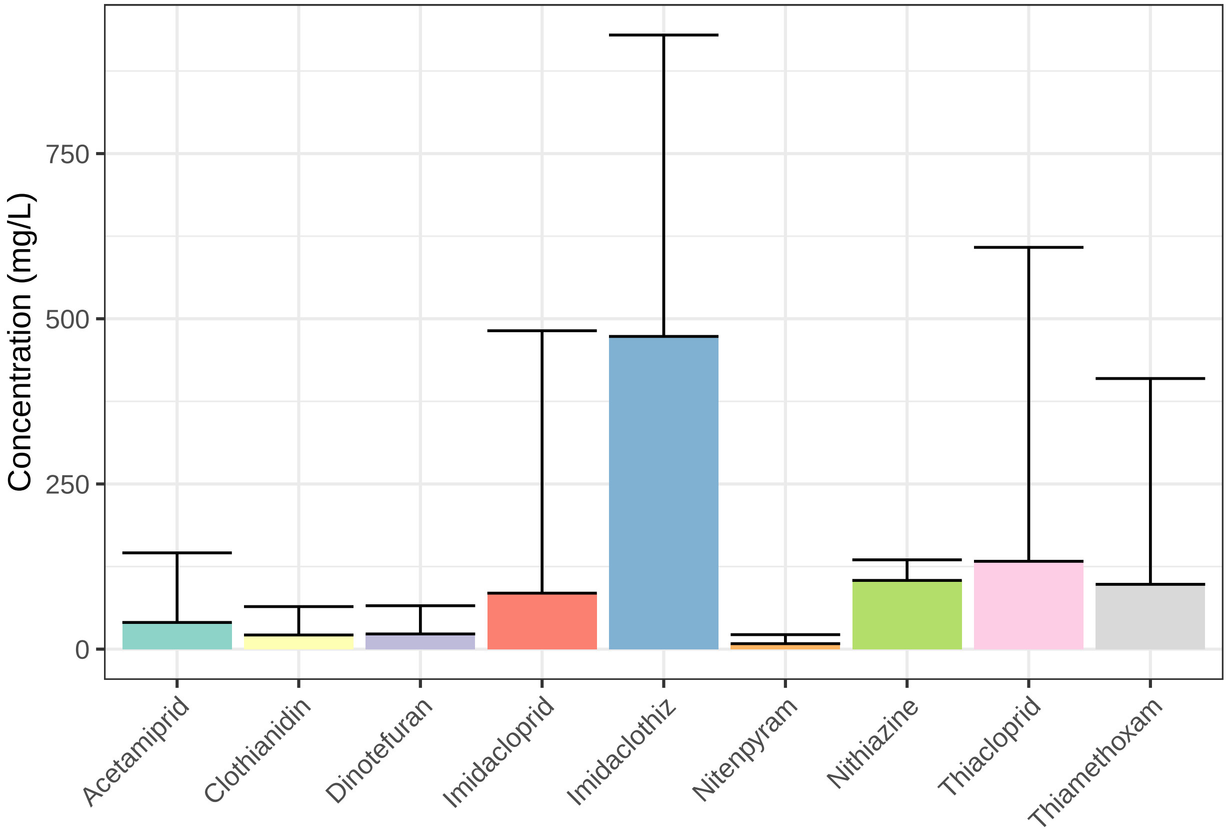


Question: How do these two types of lines affect your interpretation of the data?

> Answer: The first plot shows that the majority of discharge measurements are relatively low, and it allows the viewer to roughly determine the frequency and magnitude of flood events. The second line shows average trends over time. Though it erroneously shows an apparent drought in the 70s and 80s due to a lack of data, it also shows that stream discharge is increasing since 2010. However, the viewer is unable to see single events in the second plot.

7. [ECOTOX Neonicotinoids] Plot the concentration, divided by chemical name. Choose a geom that accurately portrays the distribution of data points.

```
#7
## Use filter to choose all rows with units of AI mg/L
EPA.Ecotox.mgL <- filter(EPA.Ecotox.Raw, Conc..Units..Std. == "AI mg/L")

## geometries commented out below did not show distribution well because of range of outliers of Imidacl

# ggplot(EPA.Ecotox.mgL, aes(x = Chemical.Name, y = Conc..Mean..Std.)) +
#   geom_boxplot()
#
# ggplot(EPA.Ecotox.mgL, aes(x = Chemical.Name, y = Conc..Mean..Std.)) +
#   geom_violin()
#
# ggplot(EPA.Ecotox.mgL, aes(x = Conc..Mean..Std., color = Chemical.Name, stat(density))) +
#   geom_freqpoly(bins = 50) +
#   facet_wrap(~Chemical.Name)

## try geom_bar with error bars
EPA.Ecotox.Summary <- EPA.Ecotox.mgL %>%
  group_by(Chemical.Name) %>%
  summarize(sd = sd(Conc..Mean..Std.),
            mean = mean(Conc..Mean..Std.))

ggplot(EPA.Ecotox.Summary, aes(x = Chemical.Name, y = mean, fill = Chemical.Name)) +
  geom_bar(stat = "identity") +
  ## error bars extend below 0, so only show max error bar
  geom_errorbar(aes(ymin = mean, ymax = mean+sd)) +
  labs(x = NULL, y = "Concentration (mg/L)") +
  theme(legend.position = "none") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")
```


Concentration (mg/L)
Acetamiprid
Clothianidin
Dinotefuran
Imidacloprid
Imidaclothiz
Nitenpyram
Nithiazine
Thiacloprid
Thiamethoxam