Assignment 5: Data Visualization

Siying Chen

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 Tuesday, 19 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()
```

```
## [1] "/Users/Sylvia/Downloads/ENV872/ENV872"
```

```
library(tidyverse)
library(viridis)
library(RColorBrewer)
library(gridExtra)

# Import datasets
PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
USGS_flow_raw <- read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")
Ecotox_raw <-read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv")

#2
# Rename columns
```

```r
colnames(USGS_flow_raw) <- c("agency_cd", "site_no", "datetime",
                              "discharge.max", "discharge.max.approval",
                              "discharge.min", "discharge.min.approval",
                              "discharge.mean", "discharge.mean.approval",
                              "gage.height.max", "gage.height.max.approval",
                              "gage.height.min", "gage.height.min.approval",
                              "gage.height.mean", "gage.height.mean.approval")
# Check date format
class(USGS_flow_raw$datetime)
```

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

```r
class(PeterPaul.chem.nutrients$sampledate)
```

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

```r
class(PeterPaul.nutrients.gathered$sampledate)
```

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

```r
# Change date format for USGS flow data
USGS_flow_raw$datetime <- as.Date(USGS_flow_raw$datetime, format = "%m/%d/%y")

# Rearrange the datetime format as year month date
USGS_flow_raw$datetime <- format(USGS_flow_raw$datetime, "%y%m%d")

# Create a function that if d is greater than 181231, give 19, if not give 20, ammend them in the cell
create.early.dates <- (function(d) {
       paste0(ifelse(d > 181231,"19","20"),d)
       })

# Assign the new dates above as the new datetime
USGS_flow_raw$datetime <- create.early.dates(USGS_flow_raw$datetime)

# Rearrange the format as year month date
USGS_flow_raw$datetime <- as.Date(USGS_flow_raw$datetime, format = "%Y%m%d")

# Repeat for PeterPaul chemistry and physics data
PeterPaul.chem.nutrients$sampledate <-
  as.Date(PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d")
PeterPaul.nutrients.gathered$sampledate <-
  as.Date(PeterPaul.nutrients.gathered$sampledate, format = "%Y-%m-%d")
```

Define your theme

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

```r
#3
mytheme <- theme_bw(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom",
        panel.grid.major = element_line(size = 0.5, linetype = 'solid'),
        panel.grid.minor = element_line(size = 0.25, linetype = 'dashed'))

theme_set(mytheme)
```

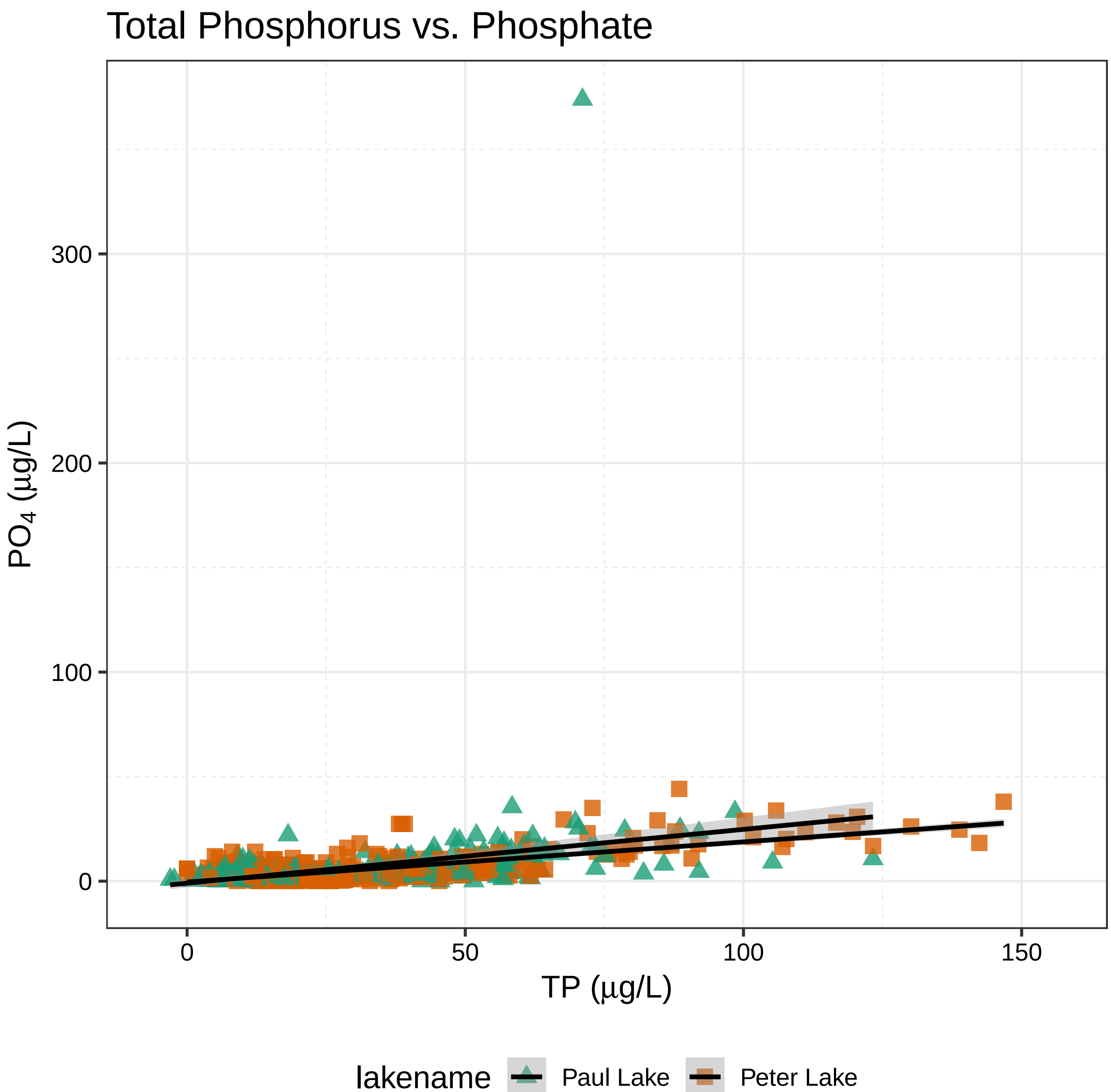
Create graphs

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.

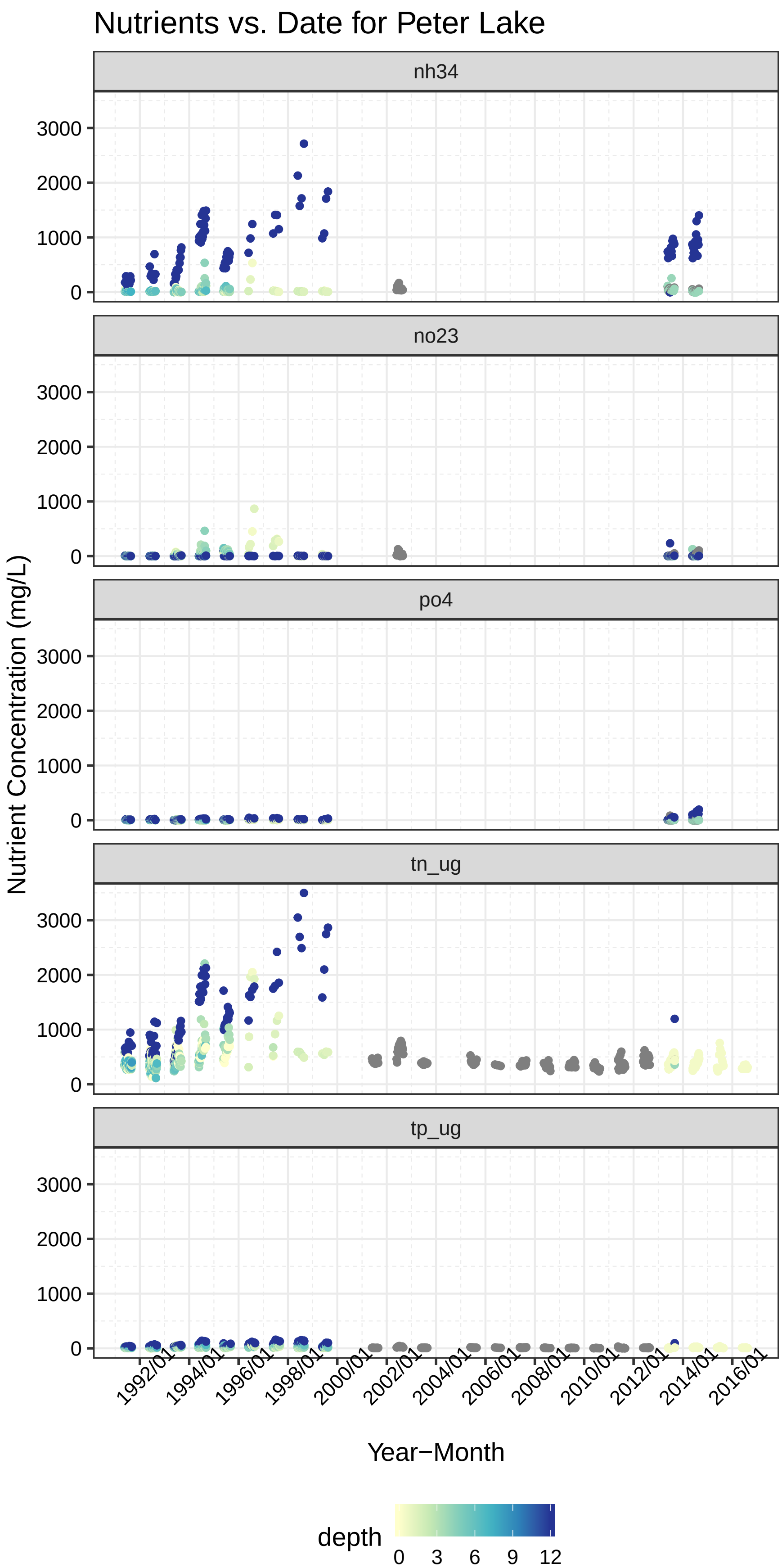
```r
#4
TP_PO4 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4, color = lakename, shape = lakename)) +
  geom_point(alpha = 0.8, size = 3) +
  labs(x = expression(paste("TP", " (", mu, "g/L)")), y = expression(paste(PO[4], " (", mu, "g/L)")),
       title = "Total Phosphorus vs. Phosphate") +
  scale_shape_manual(values = c(17, 15)) +
  scale_color_brewer(palette = "Dark2") +
  geom_smooth(method = lm, color = "black", show.legend = TRUE)

print(TP_PO4)
```



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

```
#5
Nutrients_Date <-
  ggplot(subset(PeterPaul.nutrients.gathered, lakename == "Peter Lake"),
         aes(x = sampledate, y = concentration, color = depth)) +
  geom_point() +
  labs(x = "Date", y = "Nutrient Concentration (mg/L)",
       title = "Nutrients vs. Date for Peter Lake") +
  scale_color_distiller(palette = "YlGnBu", direction = 1) +
  facet_wrap(vars(nutrient), nrow = 5) +
  scale_x_date(breaks = "2 years", date_labels = "%Y/%m", "Year-Month") +
  theme(axis.text.x = element_text(angle=45))
```

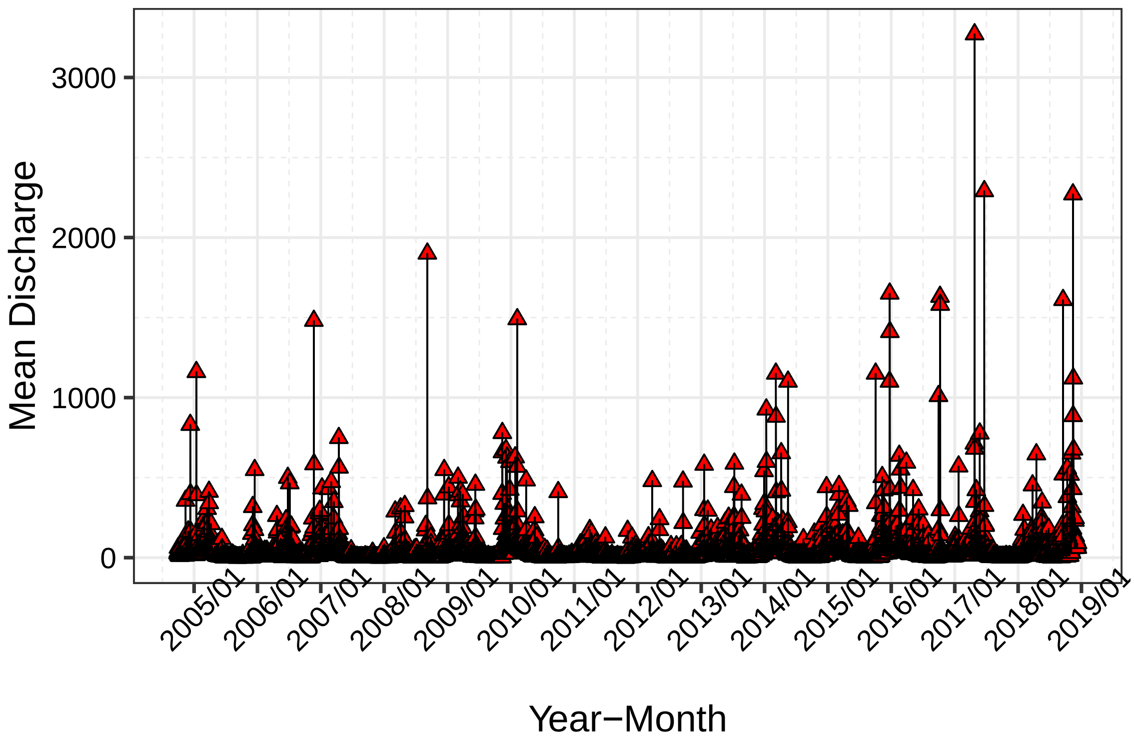
```r
print(Nutrients_Date)
```

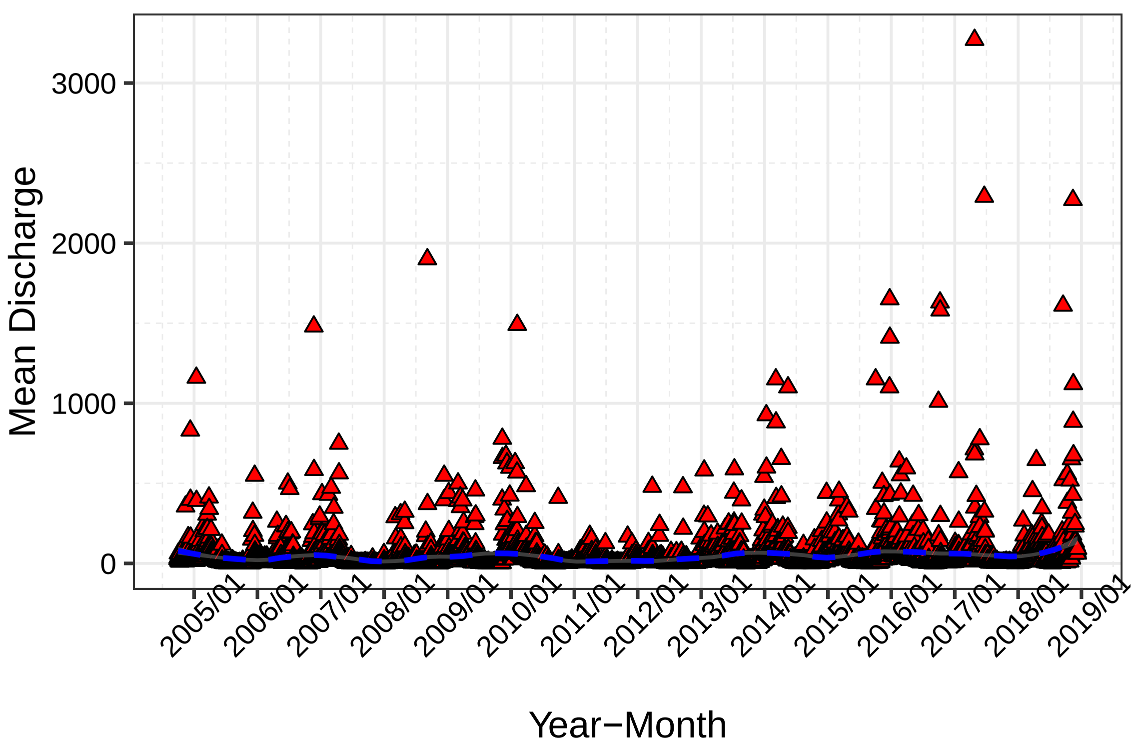


Nutrients vs. Date for Peter Lake
nh34
no23
po4
tn_ug
tp_ug
3000
2000
1000
0
Nutrient Concentration (mg/L)
1992/01
1994/01
1996/01
1998/01
2000/01
2002/01
2004/01
2006/01
2008/01
2010/01
2012/01
2014/01
2016/01
Year−Month
depth
0
3
6
9
12

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_Date1 <-
  ggplot(subset(USGS_flow_raw, !is.na(USGS_flow_raw$discharge.mean)),
         aes(x = datetime, y = discharge.mean)) +
  geom_point(shape = 24, fill = "red", size = 2) +
  labs(x = "Year-Month", y = "Mean Discharge") +
  scale_x_date(breaks = "1 year", date_labels = "%Y/%m") +
  geom_line(size = 0.3) +
  theme(axis.text.x = element_text(angle=45))

Discharge_Date2 <-
  ggplot(subset(USGS_flow_raw, !is.na(USGS_flow_raw$discharge.mean)),
         aes(x = datetime, y = discharge.mean)) +
  geom_point(shape = 24, fill = "red", size = 2) +
  labs(x = "Year-Month", y = "Mean Discharge") +
  scale_x_date(breaks = "1 year", date_labels = "%Y/%m") +
  geom_smooth(method = "loess", color = "blue", linetype = "dashed", span = 0.2) +
  theme(axis.text.x = element_text(angle=45))

grid.arrange(Discharge_Date1, Discharge_Date2)
```





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

Answer: geom_line gives you the exact data trend since it's connecting all the data points, it may shows a lot of variabilities or even "exaggerate" the extremes of the data. In this graph, I think the high discharge points are more apparent, which would affect our interpretation of the data. geom_smooth is a best-fit line of the data, it would give you a general trend of the it's less affected by the outliers/extremes of the data. In this graph, we can see the overall trend of the majority of the data without focusing too much on the outliers.

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

```
#7
Ecotox <-
  ggplot(Ecotox_raw, aes(x = Chemical.Name, y = Conc..Mean..Std., color = Chemical.Name)) +
  geom_boxplot() +
  labs(x = "Chemical Name", y = "Concentration", title = "Chemical Name vs. Concentration") +
  scale_color_brewer(palette = "Set1")

print(Ecotox)
```