Assignment 5: Data Visualization

Ying Wei Jong

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
PeterPaulTidy <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
PeterPaulGathered <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv
USGS.data <- read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv", header=T)
Ecotox <- read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv")

#2
str(PeterPaulTidy)
```

```
## 'data.frame':    23372 obs. of  14 variables:
##  $ lakename       : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 1 1 1 1 ...
##  $ daynum         : int  148 148 148 148 148 148 148 148 148 148 ...
##  $ year4          : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ sampledate     : Factor w/ 1105 levels "1984-05-27","1984-05-28",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ depth          : num  0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num  14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num  9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num  1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num  1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
```

```
##  $ tn_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
str(PeterPaulGathered)
```

```
## 'data.frame':    7997 obs. of  7 variables:
##  $ lakename     : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 2 2 2 2 ...
##  $ daynum       : int  140 140 140 140 140 140 140 140 140 140 ...
##  $ year4        : int  1991 1991 1991 1991 1991 1991 1991 1991 1991 1991 ...
##  $ sampledate   : Factor w/ 778 levels "1991-05-20","1991-05-27",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ depth        : num  0 0.85 1.75 3 4 6 0 1 2.25 3.5 ...
##  $ nutrient     : Factor w/ 5 levels "nh34","no23",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ concentration: num  538 285 399 453 363 583 352 356 364 582 ...
```

```
str(USGS.data)
```

```
## 'data.frame':    33216 obs. of  15 variables:
##  $ agency_cd             : Factor w/ 1 level "USGS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ site_no               : int  2085000 2085000 2085000 2085000 2085000 2085000 2085000 2085000 2085
##  $ datetime              : Factor w/ 33216 levels "1/1/00","1/1/01",..: 20 1021 2022 2295 2386 2477 1
##  $ X165986_00060_00001   : num  74 61 56 54 48 47 44 41 44 57 ...
##  $ X165986_00060_00001_cd: Factor w/ 4 levels "","A","A:e","P": 2 2 2 2 2 2 2 2 2 2 ...
##  $ X165987_00060_00002   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X165987_00060_00002_cd: Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ X84936_00060_00003    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X84936_00060_00003_cd : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ X84937_00065_00001    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X84937_00065_00001_cd : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ X84938_00065_00002    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X84938_00065_00002_cd : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ X84939_00065_00003    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X84939_00065_00003_cd : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
```

```
str(Ecotox)
```

```
## 'data.frame':    1283 obs. of  13 variables:
##  $ CAS.No.          : int  138261413 111988499 138261413 138261413 111988499 111988499 111988499 111
##  $ Chemical.Name    : Factor w/ 9 levels "Acetamiprid",..: 4 8 4 4 8 8 8 8 4 4 ...
##  $ Species.Name     : Factor w/ 172 levels "Acipenser transmontanus",..: 54 86 54 43 54 54 54 54 43 9
##  $ Common.Name      : Factor w/ 124 levels "Alderfly","Alfalfa Plant Bug",..: 68 97 68 68 68 68 68 68
##  $ Effect           : Factor w/ 1 level "Mortality": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Measurement      : Factor w/ 1 level "Mortality": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Endpoint         : Factor w/ 23 levels "EC10","EC50",..: 5 23 9 5 5 5 5 9 9 20 ...
##  $ Dur..Std.        : num  28 7 28 28 21 28 14 28 28 4 ...
##  $ Conc..Type       : Factor w/ 3 levels "Active ingredient",..: 2 1 2 2 1 1 1 1 2 1 ...
##  $ Conc..Mean..Std. : num  0.000041 0.00007 0.000195 0.000235 0.00024 0.00027 0.0003 0.0003 0.000316
##  $ Conc..Units..Std.: Factor w/ 16 levels "AI mg/kg bdwt",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Pub..Year        : int  2013 2017 2013 2013 2016 2016 2016 2016 2013 1992 ...
##  $ Citation         : Factor w/ 198 levels "Aaen,S.M., L.A. Hamre, and T.E. Horsberg. A Screening of
```

```
#Need to change date for PeterPaulTidy, PeterPaulGathered and USGS.data
#It is funny that I did not realize until now, but in the as.Date function, I need to make sure that th
PeterPaulTidy$sampledate <- as.Date(PeterPaulTidy$sampledate, format = "%m/%d/%y")
PeterPaulGathered$sampledate <- as.Date(PeterPaulGathered$sampledate, format = "%Y-%m-%d")
```

```
USGS.data$datetime <- as.Date(USGS.data$datetime, format = "%m/%d/%y")
USGS.data$datetime <- format(USGS.data$datetime, format = "%y%m%d")
create.early.dates <- (function(d) {
       paste0(ifelse(d > 181231,"19","20"),d)
       })
USGS.data$datetime <- create.early.dates(USGS.data$datetime)
USGS.data$datetime <- as.Date(USGS.data$datetime, format = "%Y%m%d")
```

Define your theme

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

```
#3
library(ggplot2)
my.theme <- theme_bw(base_size = 12) +
  theme(axis.text=element_text(color="gray0"), legend.position = "right")
theme_set(my.theme)
```

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.

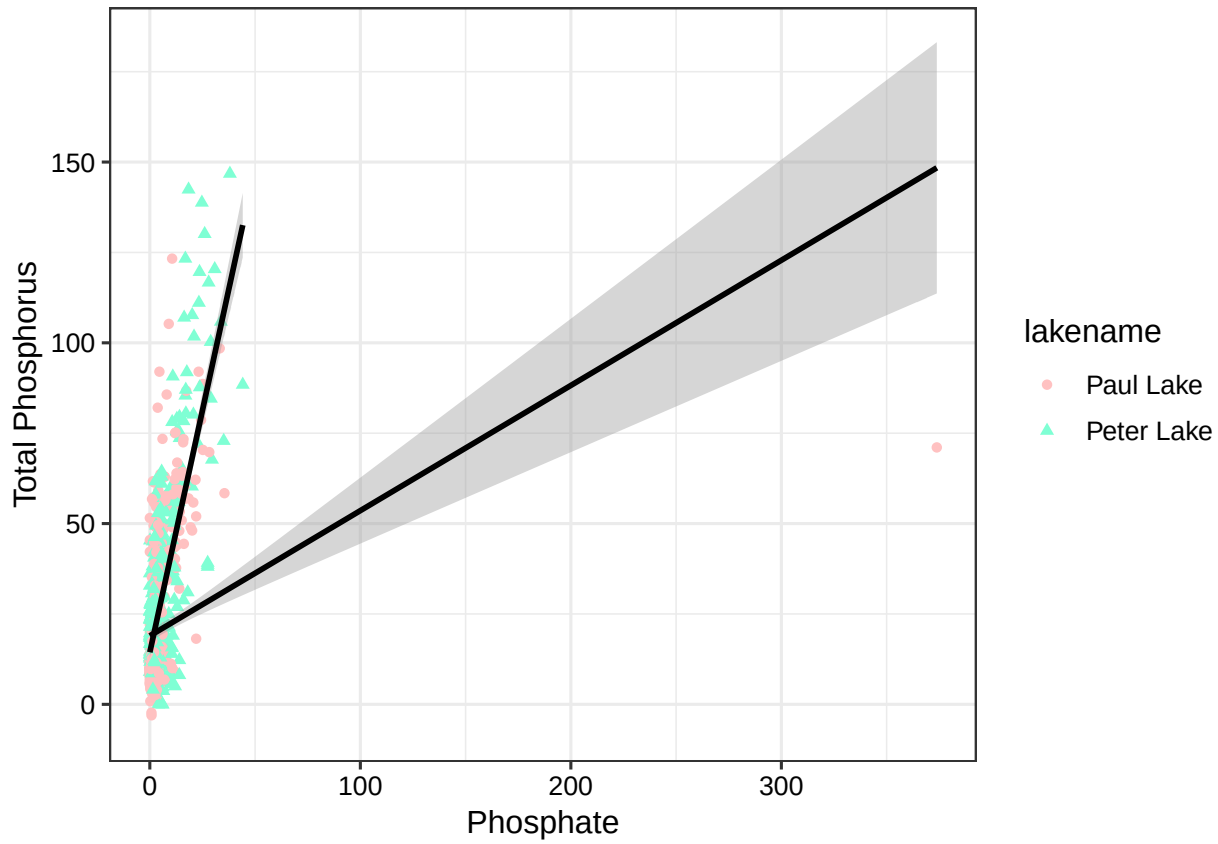
Hint: a good way to build graphs is to make them ugly first and then create more code to make them pretty.

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(PeterPaulTidy, aes(y=tp_ug, x=po4,color = lakename,shape=lakename)) +
  geom_point() +
  xlab("Phosphate") +
  ylab("Total Phosphorus") +
  scale_color_manual(values = c("rosybrown1", "aquamarine")) +
  geom_smooth(method = lm, formula = y~x,col="black")
```

```
## Warning: Removed 22309 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 22309 rows containing missing values (geom_point).
```




```
#Shall I take out that outlier? Why are there two lines?
```

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

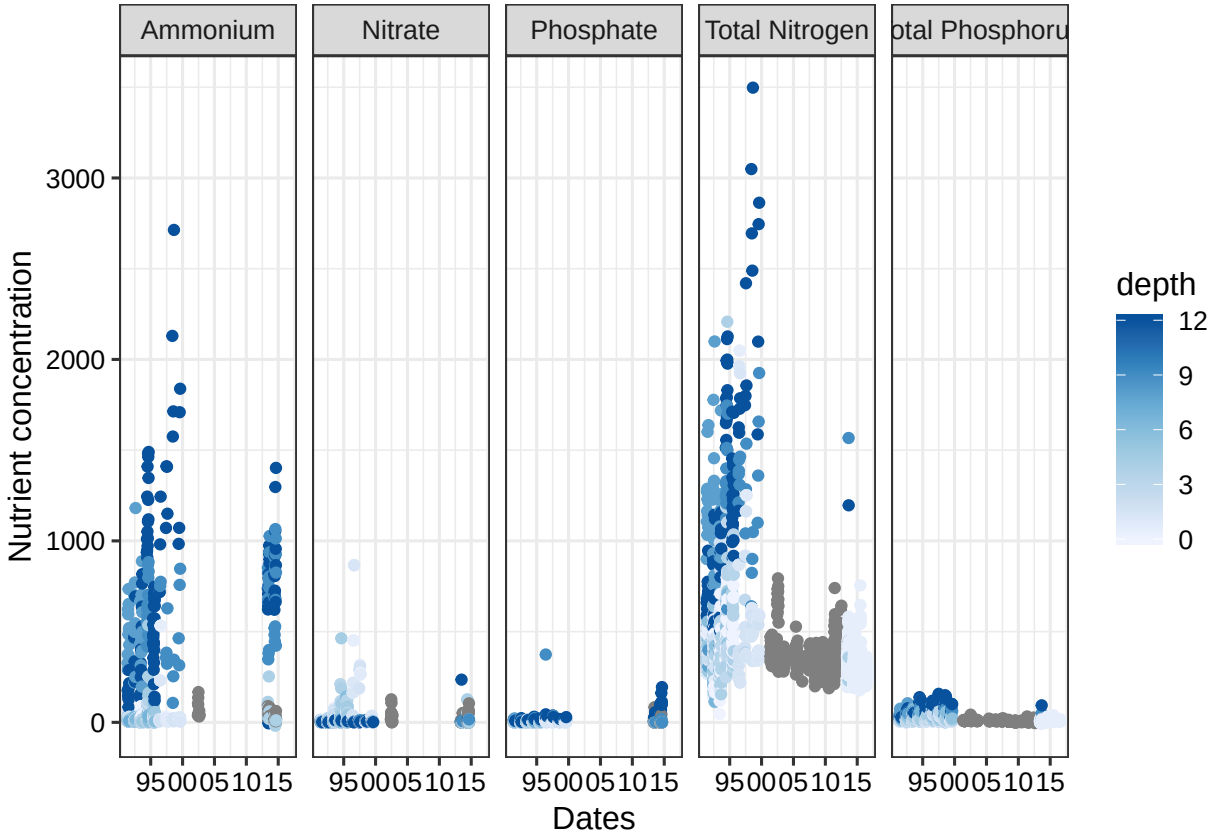
```
#5
library(RColorBrewer)

nutrient_names <- list(
  "nh34"="Ammonium",
  "no23"="Nitrate",
  "po4"="Phosphate",
  "tn_ug"="Total Nitrogen",
  "tp_ug"="Total Phosphorus"
)

nutrient_labeller <- function(variable,value){
  return(nutrient_names[value])
}
ggplot(PeterPaulGathered, aes(y=concentration, x=sampledate, color=depth))+
  geom_point()+
  facet_grid(PeterPaulGathered$nutrient, labeller = nutrient_labeller)+
  #facet_wrap(vars(nutrient), nrow=5) +
  xlab("Dates")+
  ylab("Nutrient concentration")+
  scale_color_distiller(palette = "Blues", direction = 1)+
  scale_x_date(
```

```
    date_breaks = "5 year", date_labels = "%y")
```

```
## Warning: The labeller API has been updated. Labellers taking `variable`and
## `value` arguments are now deprecated. See labellers documentation.
```




```
?label_value
```

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 I chose the mean discharge instead of max discharge
library(gridExtra)
library(ggpubr)
```

```
## Loading required package: magrittr
```

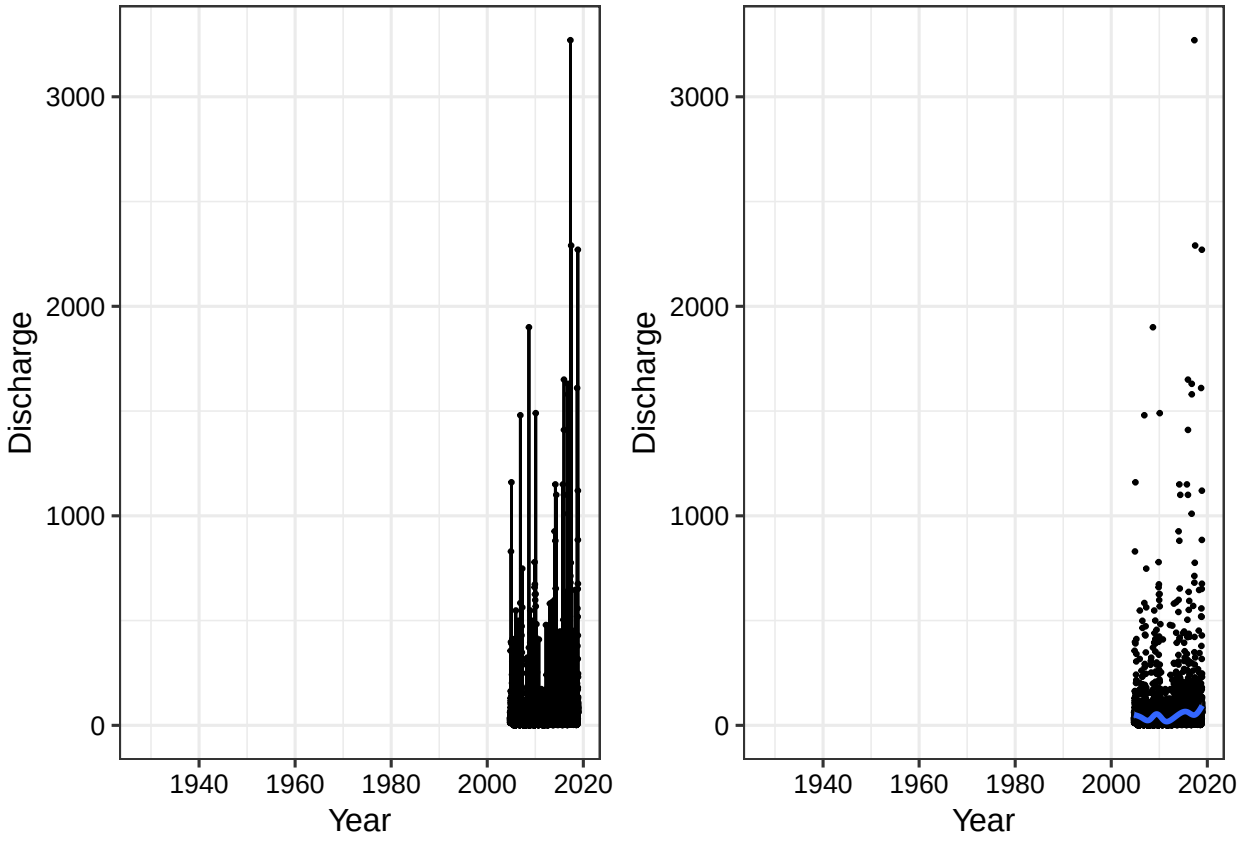
```
Plot1 <- ggplot(USGS.data,aes(x=datetime, y=X84936_00060_00003))+
  geom_point(size=0.5)+
  geom_line()+
  #xlim(2000,2020)+ When i add this line, I get error message "Error in as.Date.numeric(value) : 'origi
  ylab("Discharge")+
  xlab("Year")

Plot2 <- ggplot(USGS.data,aes(x=datetime, y=X84936_00060_00003))+
  geom_point(size=0.5)+
  geom_smooth(method="auto")+
  #xlim(2000,2020)+
```

```
  ylab("Discharge")+
  xlab("Year")

ggarrange(Plot1, Plot2, nrow=1, ncol=2)
```

```
## Warning: Removed 28049 rows containing missing values (geom_point).
## Warning: Removed 28033 rows containing missing values (geom_path).
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 28049 rows containing non-finite values (stat_smooth).
## Warning: Removed 28049 rows containing missing values (geom_point).
```



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

> Answer: The first plot where points are connected by lines does not really help in interpreting the data because I cannot see any patterns from the lines. The second plot is slightly better as i can see the general trend of where most data points lie.

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

```
#7
ggplot(Ecotox,aes(y=`Conc..Mean..Std.`, x=Chemical.Name, col=Chemical.Name))+
  geom_violin() +
  ylab("Concentration")+
  xlab("Chemical Types")+
  theme_bw(base_size = 9) +
```

```
  theme(legend.position="none")
```