Assignment 5: Data Visualization

Njeri Kara

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
#Setting the working directory
setwd("C:/Users/jerik/OneDrive/Documents/Spring 2019 Semenster/Environmental Data Analytics/EDA_R_Work/I
#Confirming that it is the correct working directory
getwd()
```

```
## [1] "C:/Users/jerik/OneDrive/Documents/Spring 2019 Semenster/Environmental Data Analytics/EDA_R_Work/
```

```
#Loading necessary packages
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(knitr)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

```
#Uploading the required dataset.
NTL.Nutrients.PP.gathered.process.D <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")

NTL.Nutrients.PP.process.D <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv")

ECOTOX.Neonicotinoids.Mortality.raw.D <-
  read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv")

USGS.Flow.raw.D <-
  read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")

#Exploring the datasets to determine data columns and format
str
```

```
## function (object, ...)
## UseMethod("str")
## <bytecode: 0x0000000018579020>
## <environment: namespace:utils>
```

```
#2
#Exploring the datasets to determine data columns and format;
#changing date variable to date format
str(NTL.Nutrients.PP.process.D)
```

```
## 'data.frame':    2770 obs. of  13 variables:
##  $ lakeid    : Factor w/ 2 levels "L","R": 1 1 1 1 1 1 2 2 2 2 ...
##  $ lakename  : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 2 2 2 2 ...
##  $ year4     : int  1991 1991 1991 1991 1991 1991 1991 1991 1991 1991 ...
##  $ daynum    : int  140 140 140 140 140 140 140 140 140 140 ...
##  $ sampledate: Factor w/ 778 levels "1991-05-20","1991-05-27",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ depth_id  : int  1 2 3 4 5 6 1 2 3 4 ...
##  $ depth     : num  0 0.85 1.75 3 4 6 0 1 2.25 3.5 ...
##  $ tn_ug     : num  538 285 399 453 363 583 352 356 364 582 ...
##  $ tp_ug     : num  25 14 14 14 13 37 11 15 28 14 ...
##  $ 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 ...
##  $ comments  : logi  NA NA NA NA NA NA ...
```

```
NTL.Nutrients.PP.process.D$sampledate <- as.Date(NTL.Nutrients.PP.process.D$sampledate, format = "%Y-%m
```

```
str(NTL.Nutrients.PP.gathered.process.D)
```

```
## '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 ...
```

```
NTL.Nutrients.PP.gathered.process.D$sampledate <- as.Date(NTL.Nutrients.PP.gathered.process.D$sampledat
```

```
str(USGS.Flow.raw.D)
```

```
## '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 
##  $ 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 ...
```

```
#Changing the class of the datetime variable to date
USGS.Flow.raw.D$datetime <- as.Date(USGS.Flow.raw.D$datetime, format = "%m/%d/%y")
#changing the date format to be consistent with the other datasets
USGS.Flow.raw.D$datetime <- format(USGS.Flow.raw.D$datetime, "%y%m%d")
#correcting error in
#creating a function that specifies that if d is greater than 181231 (%y%m%d - format) then date should
date.correction.func <- (function(d) {
       paste0(ifelse(d > 181231,"19","20"),d)
       })
#running the created function with d as datetime for the dataset USGS.flow.data
USGS.Flow.raw.D$datetime <- date.correction.func(USGS.Flow.raw.D$datetime)

# formating the created datetime as a date
USGS.Flow.raw.D$datetime <- as.Date(USGS.Flow.raw.D$datetime, format = "%Y%m%d")
```

Define your theme

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

```
#3
#building a theme
NK.theme <- theme_light(base_size = 12) +
  theme(plot.background = element_rect(fill = "grey97"),panel.grid.major =element_line(linetype = "dott

#setting it as my default theme
theme_set(NK.theme)
```

Create graphs

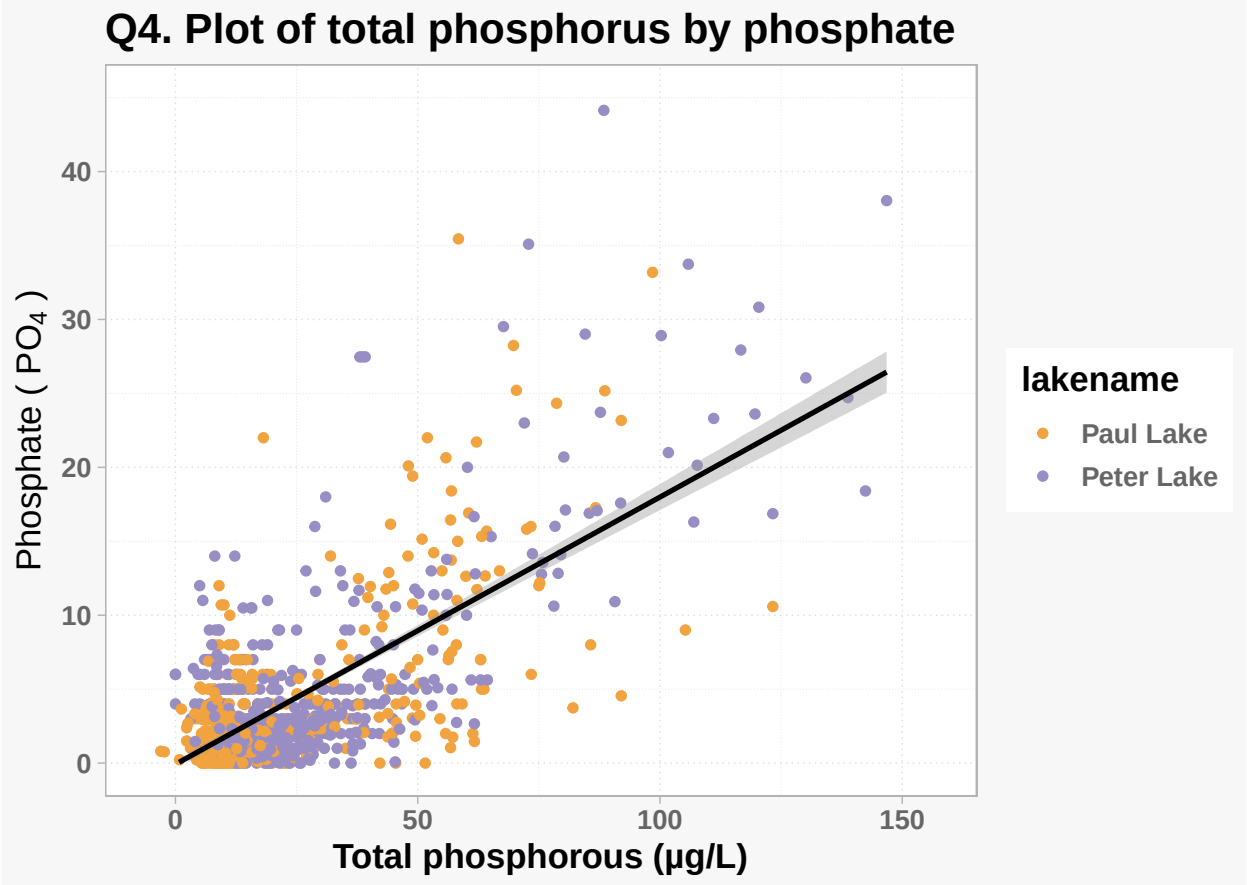
For numbers 4-7, create graphs that follow best practices for data visualization. To make your graphs "pretty," ensure your theme, color palettes, axes, and legends are edited to your liking.

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.

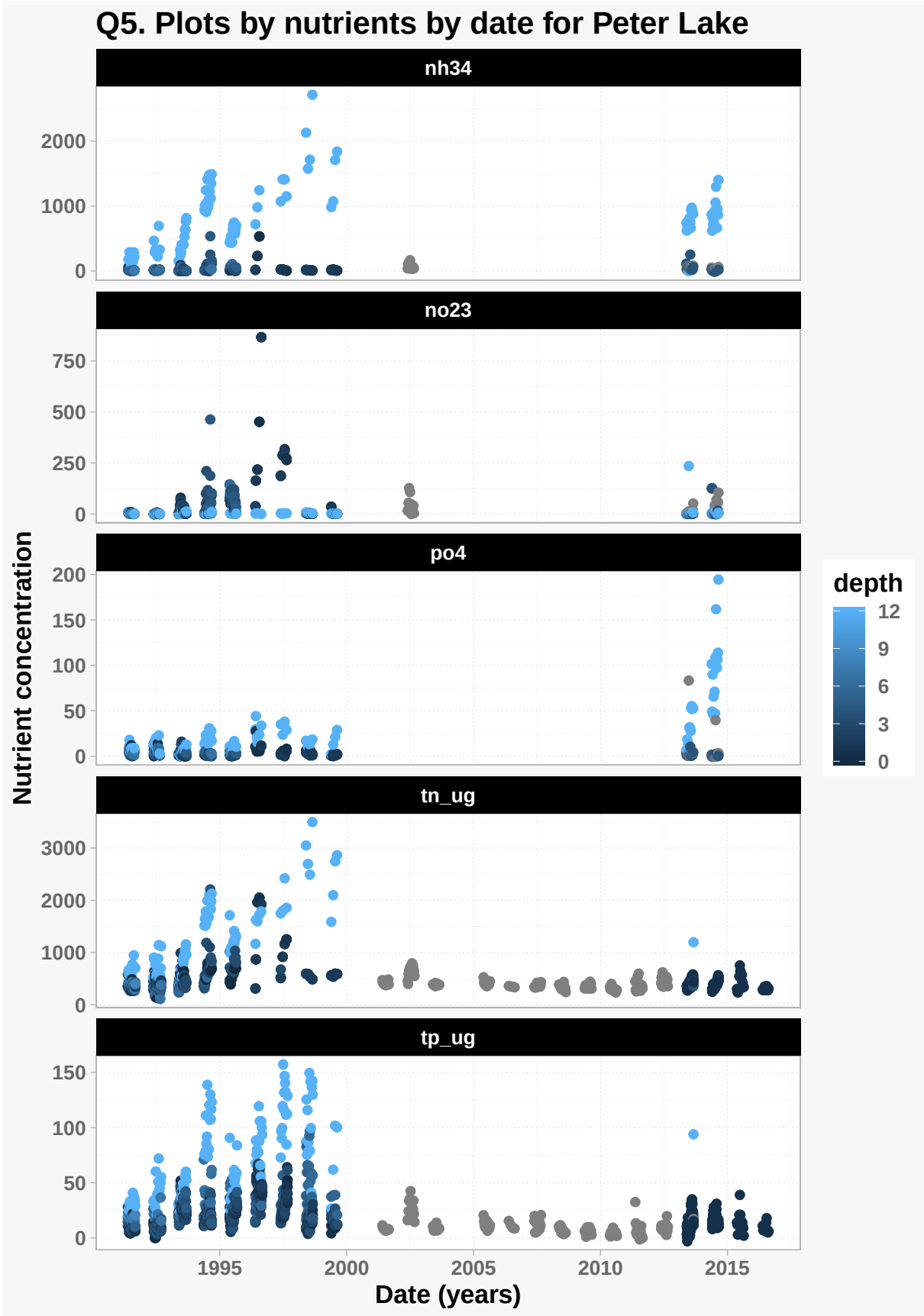
```
#4
NTL.LTER.Plot.Q4 <- ggplot(NTL.Nutrients.PP.process.D, aes(x = tp_ug, y = po4, color = lakename)) +
  geom_point() +
  geom_smooth(method = lm, color = "black") +
  scale_color_manual(values = c("#f1a340", "#998ec3")) +
  ylim(c(0,45)) +
  ggtitle("Q4. Plot of total phosphorus by phosphate") +
  xlab("Total phosphorous (\U003BCg/L)") +
  ylab(expression("Phosphate ( PO"[4]* " )"))

print(NTL.LTER.Plot.Q4)
```

```
## Warning: Removed 1708 rows containing non-finite values (stat_smooth).

## Warning: Removed 1708 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_smooth).
```



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

```
#5
#geting a subset of data for only peter lake
NTL.LTER.Peter.Q5.data <- subset(NTL.Nutrients.PP.gathered.process.D, lakename =="Peter Lake")
#ploting peter lake data
NTL.LTER.Peter.plot.Q5  <-
  ggplot(NTL.LTER.Peter.Q5.data, aes(x = sampledate, y = concentration , colour = depth, fill = depth))
  geom_point(size = 2) +
  facet_wrap(vars(nutrient), nrow = 5, scales = "free_y") +
  theme(strip.background = element_rect(fill = "black"), strip.text = element_text(color = "white")) +
  ggtitle("Q5. Plots by nutrients by date for Peter Lake") +
  xlab("Date (years)") +
  ylab("Nutrient concentration")

print(NTL.LTER.Peter.plot.Q5)
```

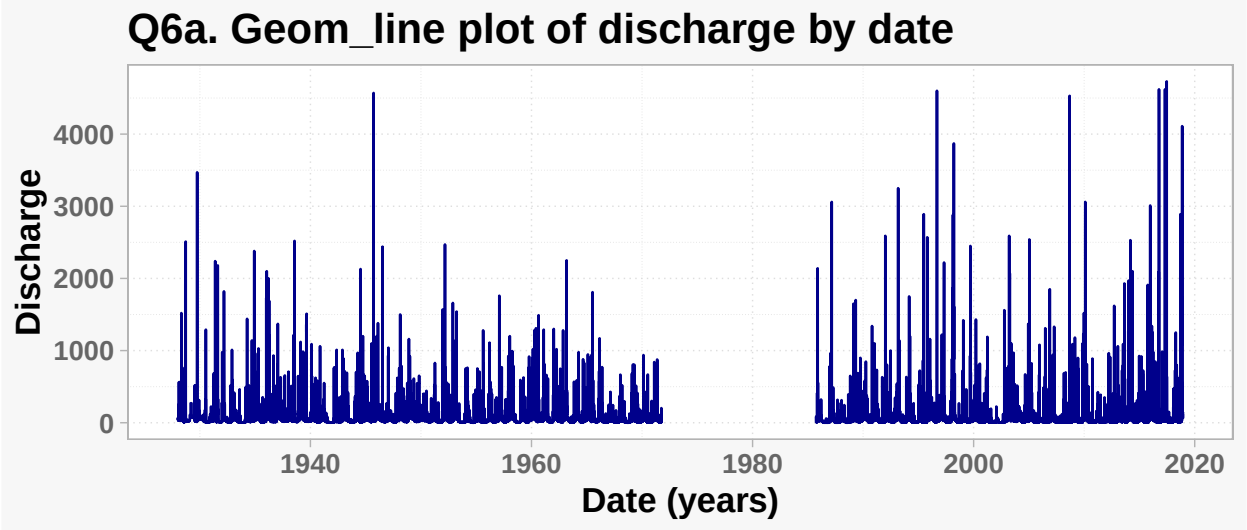

Q5. Plots by nutrients by date for Peter Lake
nh34
no23
po4
tn_ug
tp_ug
Nutrient concentration
Date (years)
depth
12
9
6
3
0

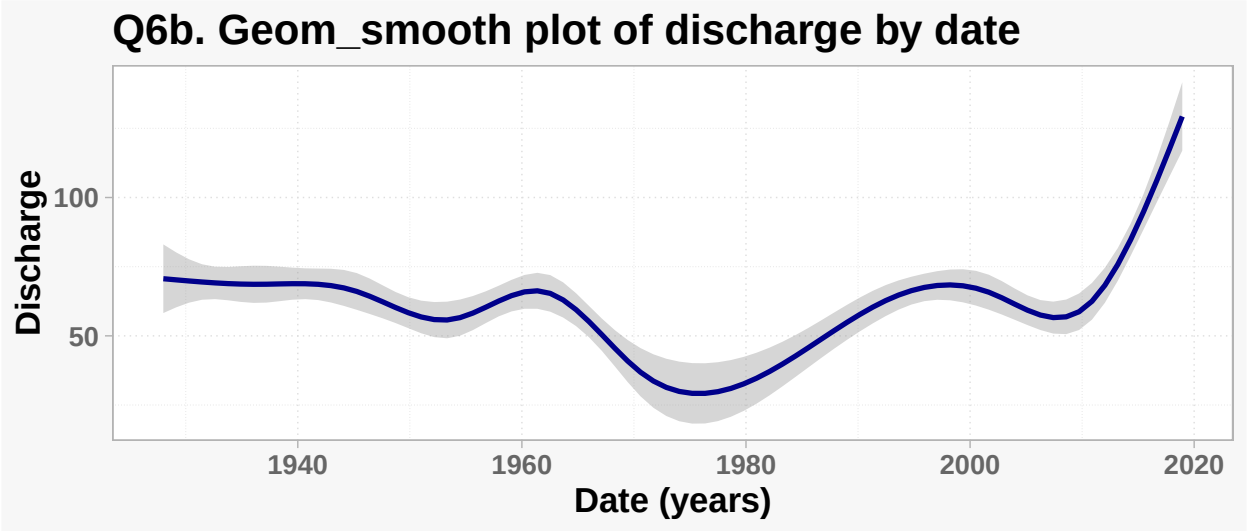
6. [USGS gauge] Plot discharge by date. Create two plots, one with the points connected with geom_line and one with the points connected with geom_smooth (hint: do not use method = "lm"). Place these graphs on the same plot (hint: ggarrange or something similar)

```
#6
#plot with geom_line
USGS.plot.Q6.line <- ggplot(USGS.Flow.raw.D, aes(x = datetime, y =X165986_00060_00001)) +
  geom_line(color = "darkblue") +
  ggtitle("Q6a. Geom_line plot of discharge by date") +
  xlab("Date (years)") +
  ylab("Discharge")

#plot with geom_smooth
USGS.plot.Q6.smooth <- ggplot(USGS.Flow.raw.D, aes(x = datetime, y =X165986_00060_00001)) +
  geom_smooth(color = "darkblue") +
  ggtitle("Q6b. Geom_smooth plot of discharge by date") +
  xlab("Date (years)") +
  ylab("Discharge")

#placing both graphs on the same plot
grid.arrange(USGS.plot.Q6.line, USGS.plot.Q6.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: Yes. The geom_line plot shows significant fluctuations in data across the years.The data range appears to be from 0 to about 5,000. It also clearly shows missing data in 1980. Data across years does not seam to have a distinguishable trend.

The geom_smooth plot on the other hand does show as much variation in data. The data range appears to be from about 25 to 125. It also does not show any indication of missing data and it change in data across years appears to have a trend.

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

```
#7
#selectring subset of data for concentrations in mg/L
Ecotox.plot.Q7 <- subset(ECOTOX.Neonicotinoids.Mortality.raw.D, Conc..Units..Std. =="AI mg/L")

#plotting graph
Ecotox.plot.Q7 <- ggplot(ECOTOX.Neonicotinoids.Mortality.raw.D, aes(x = Chemical.Name, y = Conc..Mean..
  geom_boxplot(aes(color = Chemical.Name)) +
  ylim(c(0,750)) +
  coord_flip() +
```

```
  theme(legend.position = "none") +
  ggtitle("Q7. Plot of concentration by chemical name") +
  xlab("Chemical name") +
  ylab("Concentration")

print(Ecotox.plot.Q7)
```

```
## Warning: Removed 77 rows containing non-finite values (stat_boxplot).
```