Assignment 5: Data Visualization

Felipe Raby Amadori

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).

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, eval=TRUE)

#1
getwd()
```

```
## [1] "C:/Users/Felipe/OneDrive - Duke University/1. DUKE/1. Ramos 2 Semestre/EOS-872 Env. Data Analyti
```

```
#install.packages('ggpubr')
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   0.8.1     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(RColorBrewer)
library(ggpubr)
```

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

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(colormap)

PeterPaul.ChemPhy.Tidy <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv")
PeterPaul.ChemPhy.Gathered <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Proc
USGS_stream_gauge <- read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")
EPA_Ecotox_Neonicotinoids <- read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv")

#2

class(PeterPaul.ChemPhy.Tidy$sampledate)
```

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

```
class(PeterPaul.ChemPhy.Gathered$sampledate)
```

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

```
class(USGS_stream_gauge$datetime)
```

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

```
#EPA_Ecotox does not have dates

PeterPaul.ChemPhy.Tidy$sampledate <- as.Date(PeterPaul.ChemPhy.Tidy$sampledate, format = "%Y-%m-%d")
PeterPaul.ChemPhy.Gathered$sampledate <- as.Date(PeterPaul.ChemPhy.Gathered$sampledate, format = "%Y-%m
USGS_stream_gauge$datetime <- as.Date(USGS_stream_gauge$datetime, format = "%m/%d/%y")

USGS_stream_gauge$datetime <- format(USGS_stream_gauge$datetime, "%y%m%d")

# Changes 20 with 19
create.early.dates <- (function(d) {
       paste0(ifelse(d > 181231,"19","20"),d)
       })
# use the function
USGS_stream_gauge$datetime <- create.early.dates(USGS_stream_gauge$datetime)

# fix the format
USGS_stream_gauge$datetime <- as.Date(USGS_stream_gauge$datetime, format = "%Y%m%d")
```

```
#Check the correct format
class(PeterPaul.ChemPhy.Tidy$sampledate)
```

```
## [1] "Date"
```

```
class(PeterPaul.ChemPhy.Gathered$sampledate)
```

```
## [1] "Date"
```

```
class(USGS_stream_gauge$datetime)
```

```
## [1] "Date"
```

Define your theme

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

```
#3
felipe_theme <- theme_light(base_size = 12) +
  theme(axis.text = element_text(color = "grey8"),
        legend.position = "right")
theme_set(felipe_theme)
```

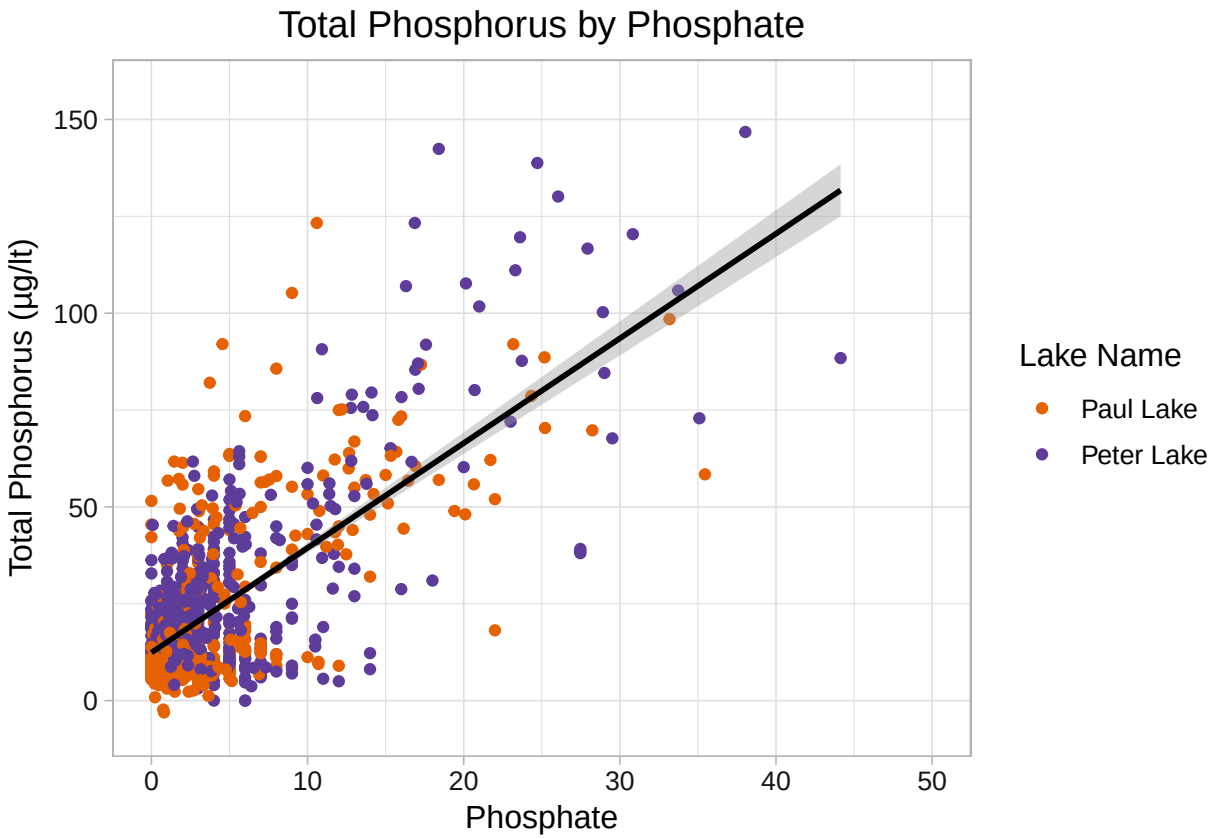
Create graphs

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.

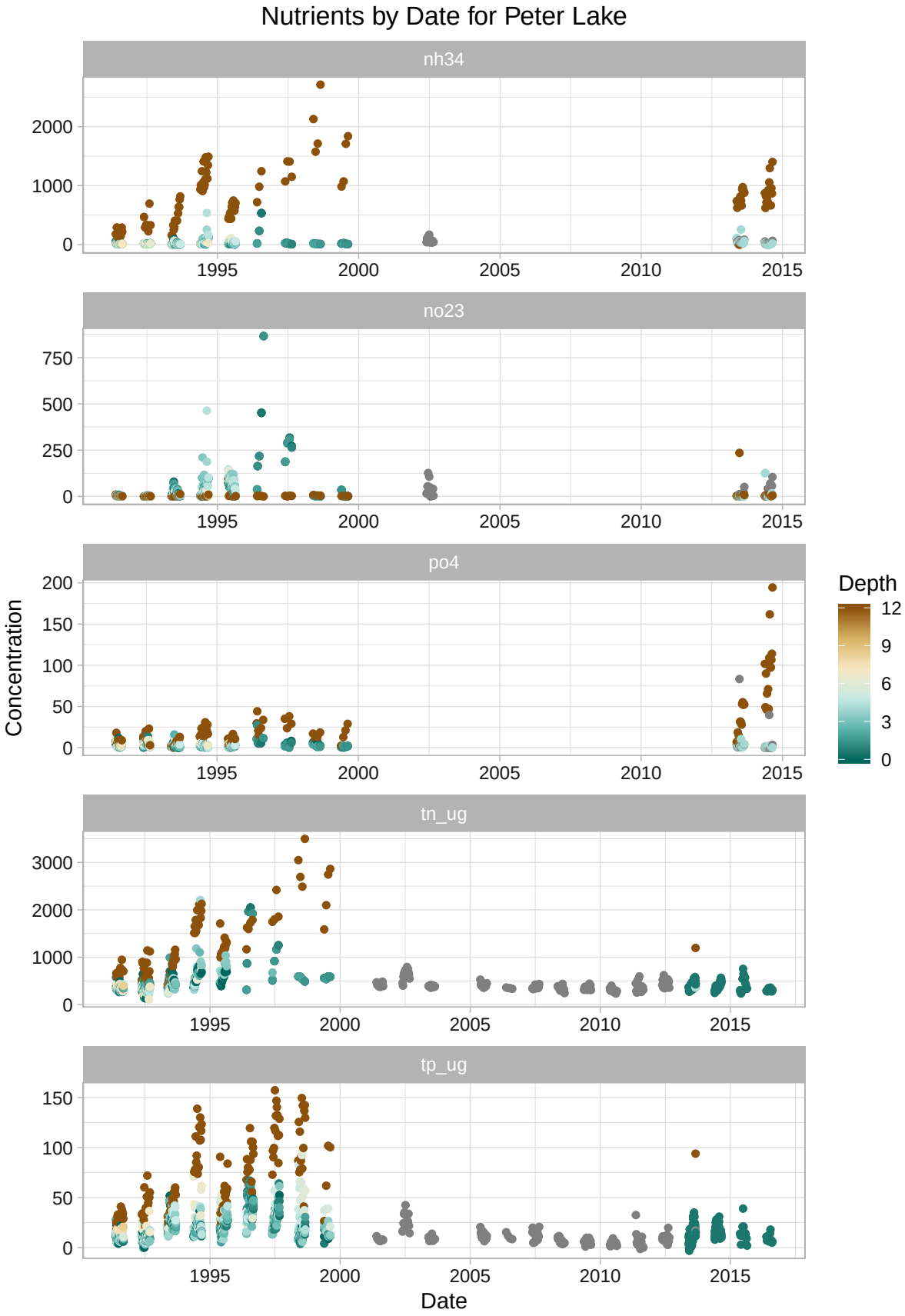
4. [NTL-LTER] Plot total phosphorus by phosphate, with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black.

```
#4
phophorus_ate_PLOT <- ggplot(PeterPaul.ChemPhy.Tidy, aes(x = po4, y = tp_ug)) +
  geom_point(aes(color = lakename)) +
  xlim(0, 50) +
  geom_smooth(method = lm, color = "black") +
  xlab(expression("Phosphate")) +
  ylab(expression("Total Phosphorus (\U003BCg/lt)"))+
  labs(color = 'Lake Name') +
  scale_color_manual(values = c("#e66101", "#5e3c99")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Total Phosphorus by Phosphate")


print(phophorus_ate_PLOT)
```



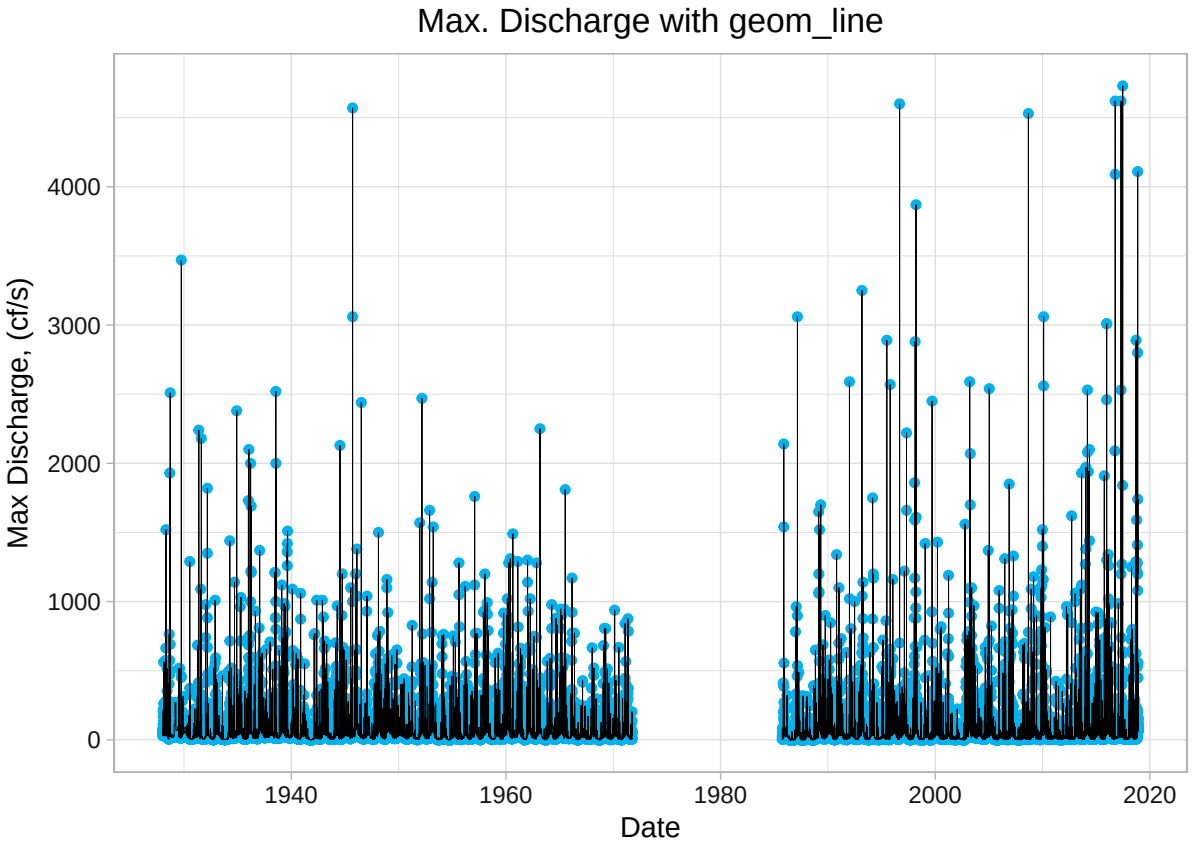
5. [NTL-LTER] Plot nutrients by date for Peter Lake, with separate colors for each depth. Facet your graph by the nutrient type.

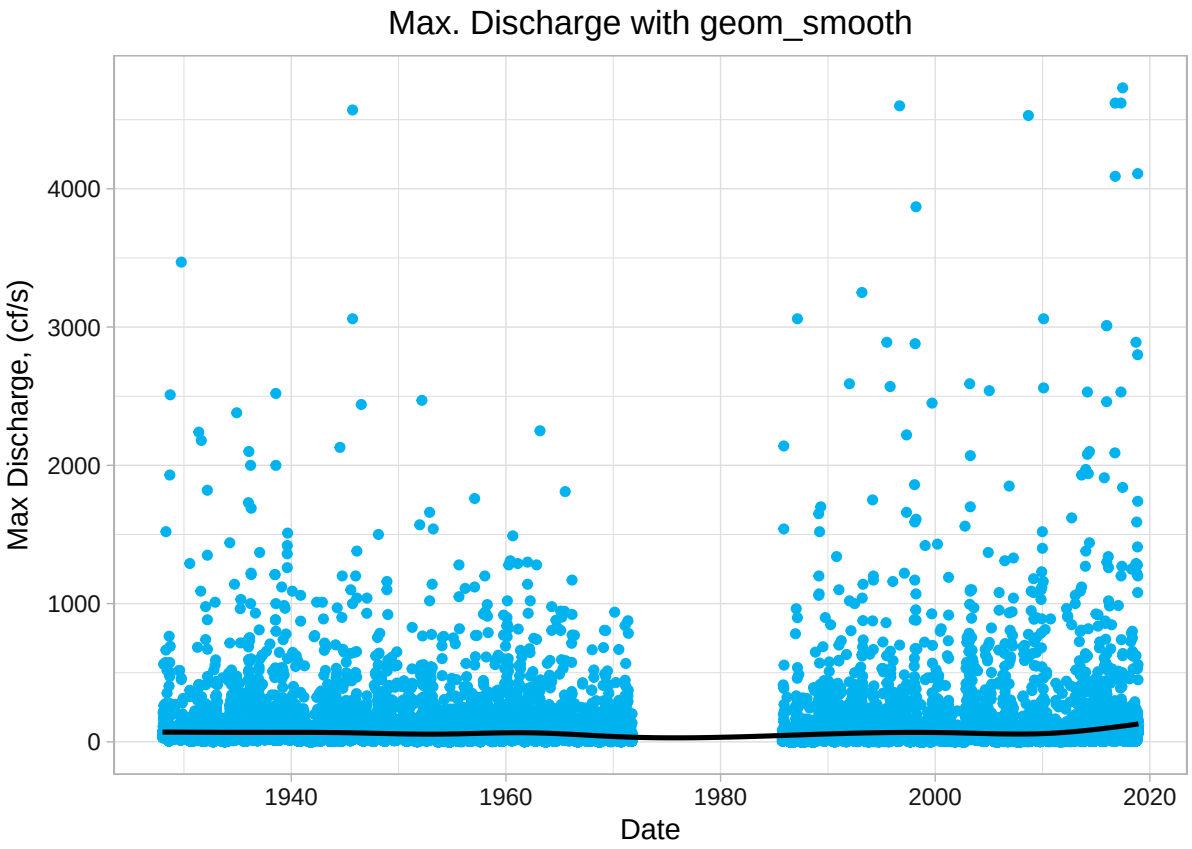
```
#5
Nutr_date_PLOT <- ggplot(subset(PeterPaul.ChemPhy.Gathered, lakename %in% c("Peter Lake")), aes(x = sam
  geom_point(aes(color = depth)) +
  xlab(expression("Date")) +
  ylab(expression("Concentration"))+
  labs(color = 'Depth') +
  scale_color_distiller(palette = "BrBG") +
  facet_wrap(vars(nutrient), nrow = 5, scales = "free") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Nutrients by Date for Peter Lake")

print(Nutr_date_PLOT)
```


Nutrients by Date for Peter Lake
nh34
no23
po4
tn_ug
tp_ug
Concentration
Date
Depth
12
9
6
3
0
1995
2000
2005
2010
2015

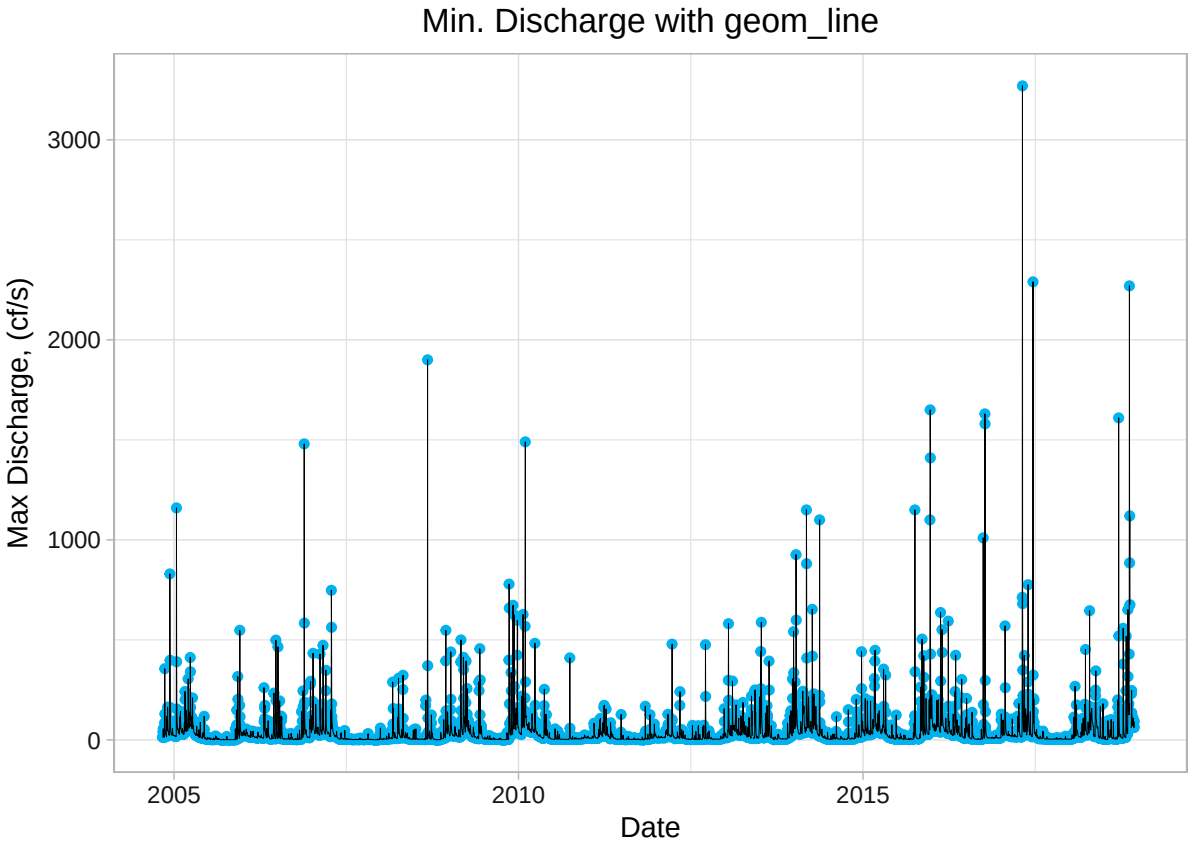
6. [USGS gauge] Plot discharge by date. Create two plots, one with the points connected with geom_line and one with the points connected with geom_smooth (hint: do not use method = "lm"). Place these graphs on the same plot (hint: ggarrange or something similar)

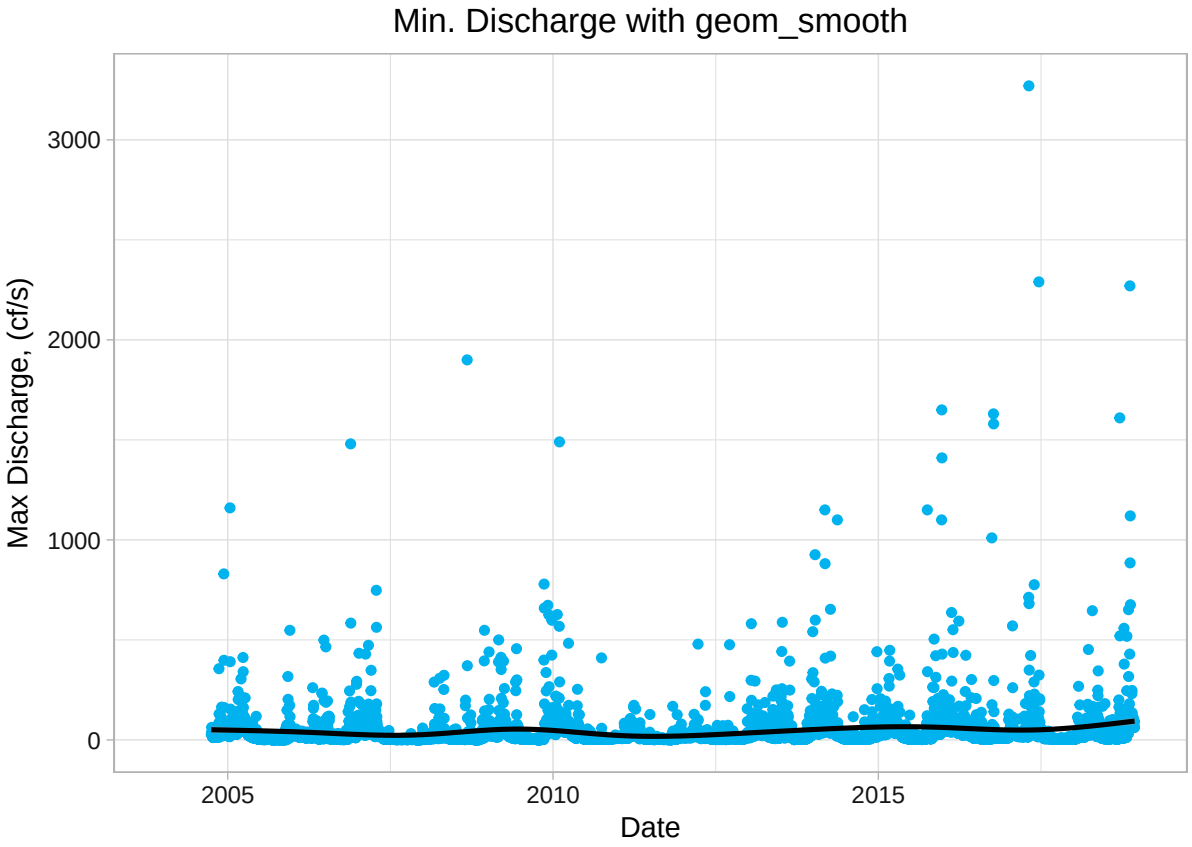
```
#6
# First for the max. discharge rate
Discharge_date_line_PLOT <- ggplot(USGS_stream_gauge,
                                   aes(x = datetime, y = X165986_00060_00001)) +
  geom_point(color = "deepskyblue2") +
  geom_line(size=0.1) +
  xlab(expression("Date")) +
  ylab(expression("Max Discharge, (cf/s)")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Max. Discharge with geom_line")

Discharge_date_smooth_PLOT <- ggplot(USGS_stream_gauge, aes(x = datetime, y = X165986_00060_00001)) +
  geom_point(color = "deepskyblue2") +
  geom_smooth(color = "black") +
  xlab(expression("Date")) +
  ylab(expression("Max Discharge, (cf/s)")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Max. Discharge with geom_smooth")

ggarrange(Discharge_date_line_PLOT, Discharge_date_smooth_PLOT, nrow = 2)
```


Max. Discharge with geom_line
Max Discharge, (cf/s)
0
1000
2000
3000
4000
1940
1960
1980
2000
2020
Date

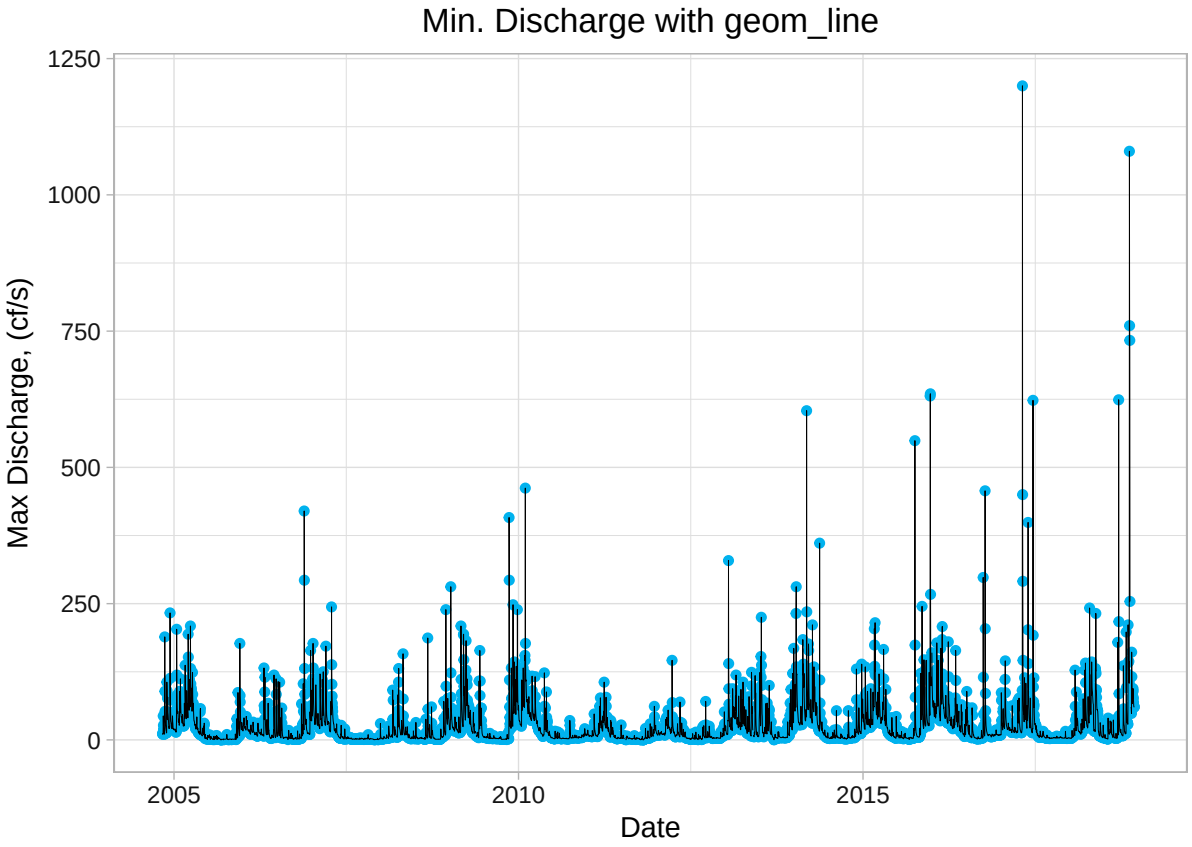

Max. Discharge with geom_smooth
Max Discharge, (cf/s)
0
1000
2000
3000
4000
1940
1960
1980
2000
2020
Date

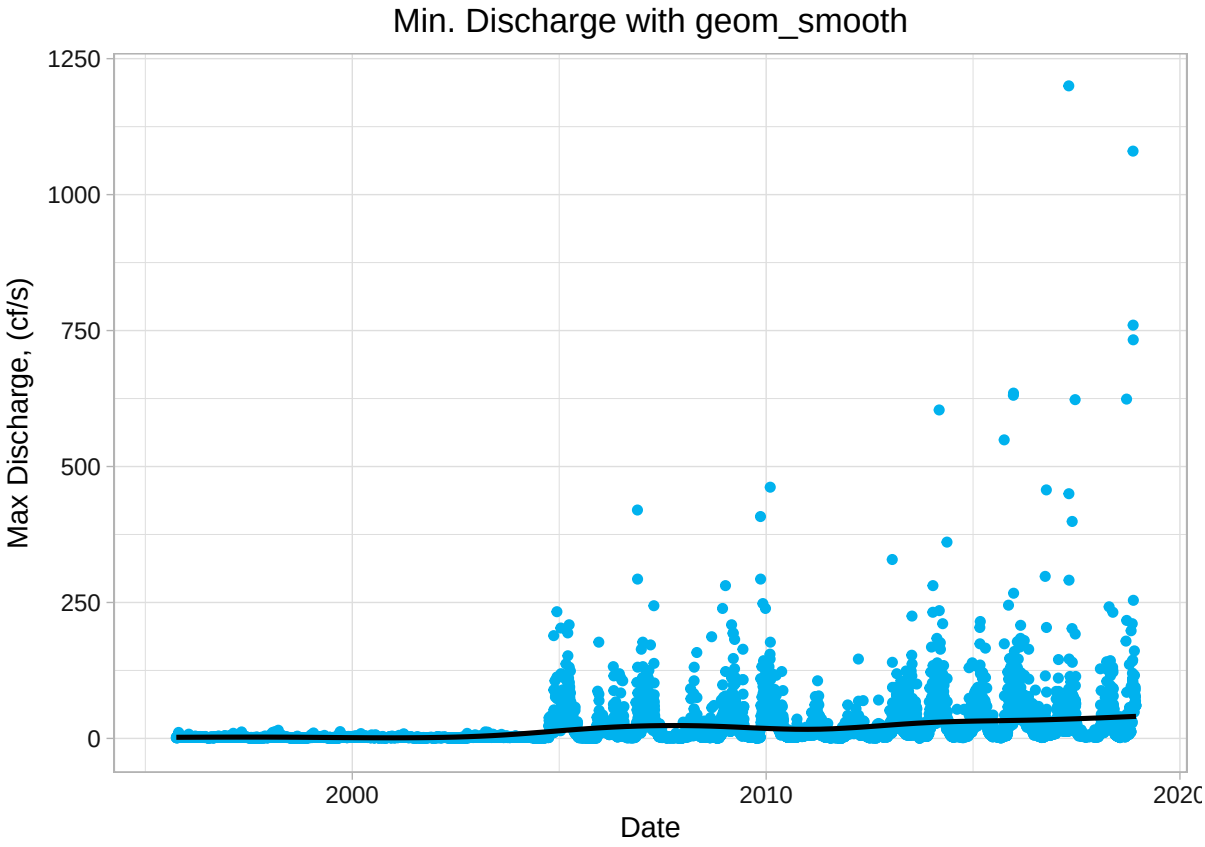
```r
# Second for the mean discharge rate

Discharge_date_line_PLOT2 <- ggplot(USGS_stream_gauge, aes(x = datetime, y = X84936_00060_00003)) +
  geom_point(color = "deepskyblue2") +
  geom_line(size=0.1) +
  xlab(expression("Date")) +
  ylab(expression("Max Discharge, (cf/s)")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Min. Discharge with geom_line") +
  scale_x_date(limits = as.Date(c("2004-11-01", "2018-12-31")))

Discharge_date_smooth_PLOT2 <- ggplot(USGS_stream_gauge, aes(x = datetime, y = X84936_00060_00003)) +
  geom_point(color = "deepskyblue2") +
  geom_smooth(color = "black") +
  xlab(expression("Date")) +
  ylab(expression("Max Discharge, (cf/s)")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Min. Discharge with geom_smooth") +
  scale_x_date(limits = as.Date(c("2004-01-01", "2018-12-31")))

ggarrange(Discharge_date_line_PLOT2, Discharge_date_smooth_PLOT2, nrow = 2)
```


Min. Discharge with geom_line
Max Discharge, (cf/s)
3000
2000
1000
0
2005
2010
2015
Date


Min. Discharge with geom_smooth
Max Discharge, (cf/s)
3000
2000
1000
0
2005
2010
2015
Date

```r
# Third for the min. discharge rate
Discharge_date_line_PLOT3 <- ggplot(USGS_stream_gauge,
                                    aes(x = datetime, y = X165987_00060_00002)) +
  geom_point(color = "deepskyblue2") +
  geom_line(size=0.1) +
  xlab(expression("Date")) +
  ylab(expression("Max Discharge, (cf/s)")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Min. Discharge with geom_line") +
  scale_x_date(limits = as.Date(c("2004-11-01", "2018-12-31")))


Discharge_date_smooth_PLOT3 <- ggplot(USGS_stream_gauge, aes(x = datetime, y = X165987_00060_00002)) +
  geom_point(color = "deepskyblue2") +
  geom_smooth(color = "black") +
  xlab(expression("Date")) +
  ylab(expression("Max Discharge, (cf/s)")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Min. Discharge with geom_smooth") +
  scale_x_date(limits = as.Date(c("1995-06-01", "2018-12-31")))


ggarrange(Discharge_date_line_PLOT3, Discharge_date_smooth_PLOT3, nrow = 2)
```


Min. Discharge with geom_line
Max Discharge, (cf/s)
1250
1000
750
500
250
0
2005
2010
2015
Date

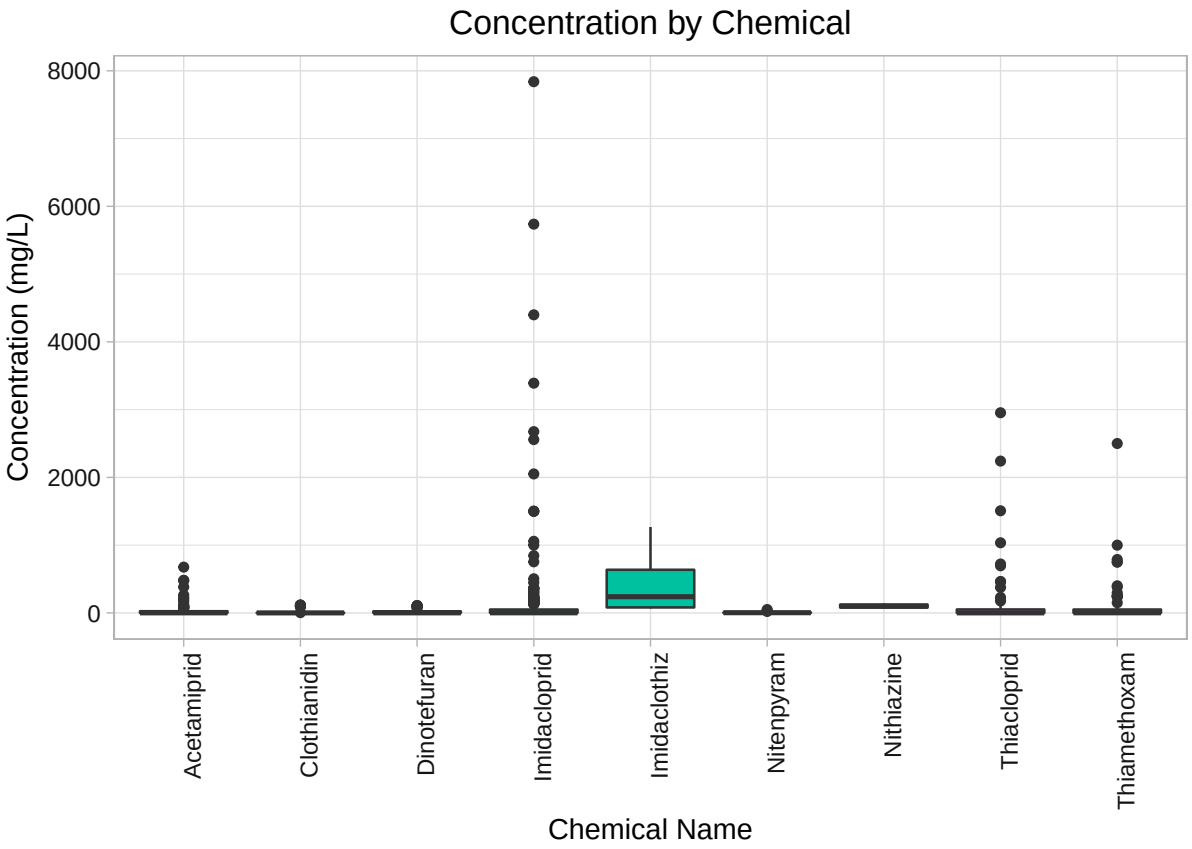

Min. Discharge with geom_smooth
Max Discharge, (cf/s)
1250
1000
750
500
250
0
2000
2010
202(
Date

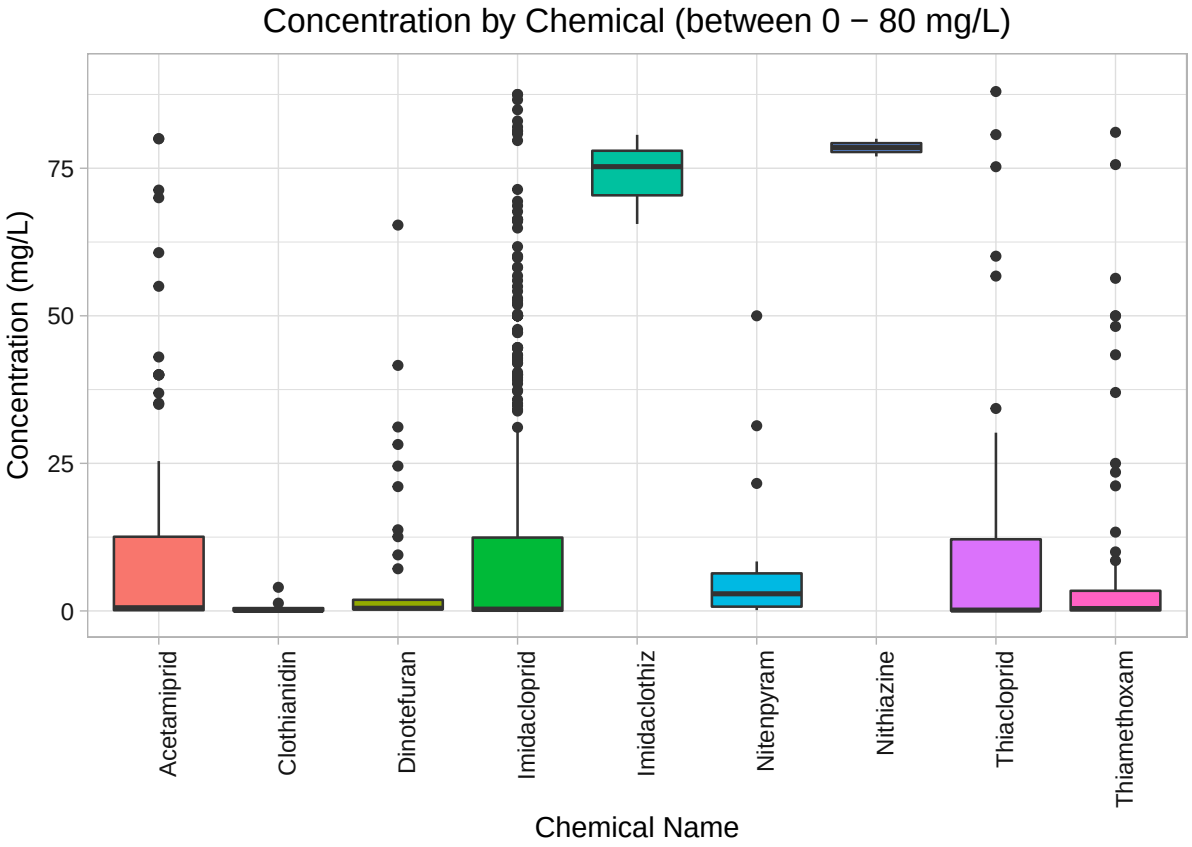
Question: How do these two types of lines affect your interpretation of the data?

Answer: The geom_line connects and goes through all the points in the data, showing clearly the extremes values of it, specially the peaks or max values of the data and are useful when trying to visualize these kind of behaviors; whereas, the geom_smooth line shows the global trends of the data, and is much less affected by the extremes. This kind of line is useful to interpret longer term trends in the data.

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

```
#7

#Full Plot
Ecotox_Conc_PLOT3 <- ggplot(subset(EPA_Ecotox_Neonicotinoids, Conc..Units..Std.
                                   %in% c("AI mg/L", "mg/L")),
                            aes(x = Chemical.Name, y = Conc..Mean..Std., fill = Chemical.Name)) +
  geom_boxplot() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_color_brewer(palette = "Set1") +
  guides(fill=FALSE) +
  xlab(expression("Chemical Name")) +
  ylab(expression("Concentration (mg/L)")) +
  ggtitle("Concentration by Chemical")


#Zooming 0-90 mg/L
Ecotox_Conc_PLOT2 <- ggplot(subset(EPA_Ecotox_Neonicotinoids, Conc..Units..Std.
                                   %in% c("AI mg/L", "mg/L")),
                            aes(x = Chemical.Name, y = Conc..Mean..Std., fill = Chemical.Name)) +
  geom_boxplot() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_color_brewer(palette = "Set1") +
  ylim(0, 90) +
  guides(fill=FALSE) +
  xlab(expression("Chemical Name")) +
  ggtitle("Concentration by Chemical (between 0 - 80 mg/L)") +
  ylab(expression("Concentration (mg/L)"))


ggarrange(Ecotox_Conc_PLOT3, Ecotox_Conc_PLOT2, nrow = 2)
```


Concentration by Chemical
Concentration (mg/L)
8000
6000
4000
2000
0
Acetamiprid
Clothianidin
Dinotefuran
Imidacloprid
Imidaclothiz
Nitenpyram
Nithiazine
Thiacloprid
Thiamethoxam
Chemical Name


Concentration by Chemical (between 0 − 80 mg/L)
Concentration (mg/L)
75
50
25
0
Acetamiprid
Clothianidin
Dinotefuran
Imidacloprid
Imidaclothiz
Nitenpyram
Nithiazine
Thiacloprid
Thiamethoxam
Chemical Name