Assignment 5: Data Visualization

Kendall Fitzgerald

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Read in the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (use the tidy `NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv` version in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
# loading packages required for this assignment
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(ggplot2)

# assigning a variable to processed data location
processed_data = "./Data/Processed_KEY"

# reading in data
PeterPaul.chem.nutrients <- read.csv(
  here(
    processed_data,"NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

NIWO.litter <- read.csv(
  here(processed_data,"NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
# fix format of dates
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
NIWO.litter$collectDate <- ymd(NIWO.litter$collectDate)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```
#3
#setting a custom theme as default theme
custom_theme <- theme_gray() +
  theme(
    plot.title = element_text(face = "bold"),
    legend.position = "top")
theme_set(custom_theme)
```

Create graphs

For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

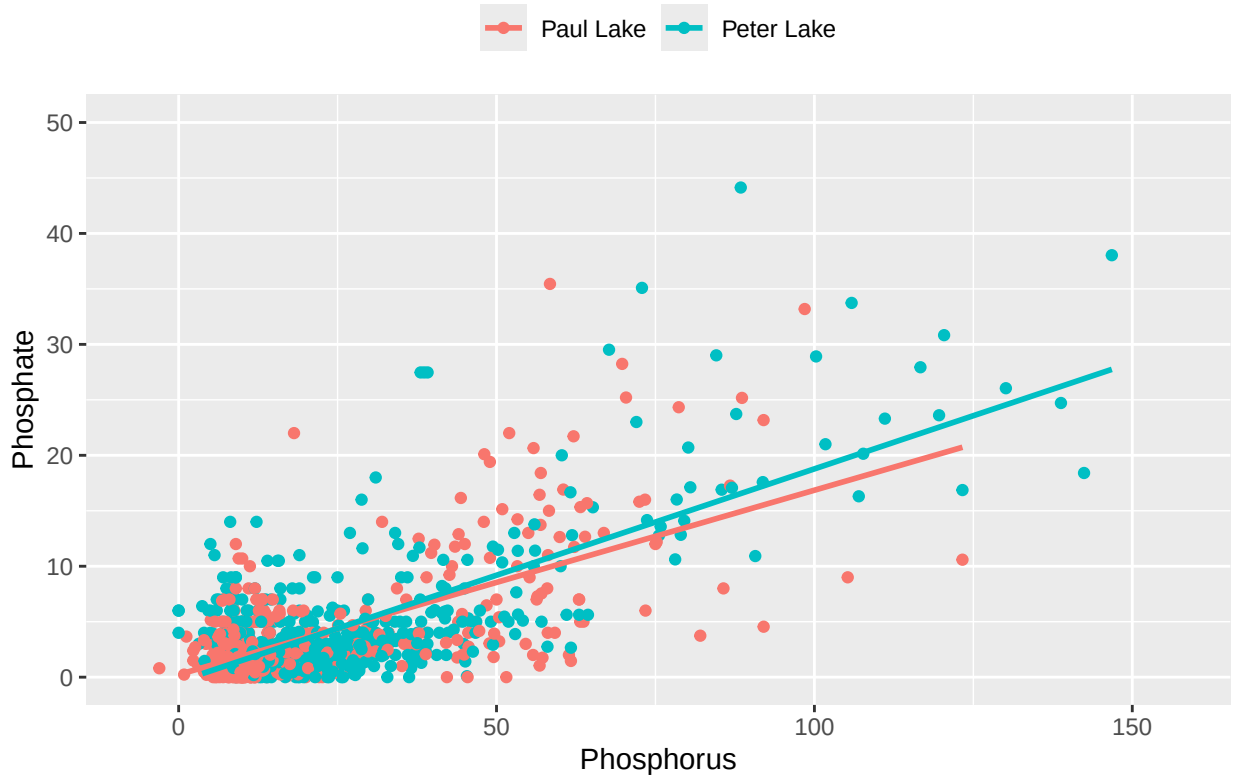
```
#4
#creating plot with lines of best fit and outlier limits
Phos.plot <- PeterPaul.chem.nutrients %>%
  ggplot(aes(x = tp_ug, y = po4,color = lakename))+
  geom_point()+
  ylim(0,50)+
  geom_smooth(method = "lm", se = FALSE)

#adding titles and fixing aesthetics
Phos.plot +
  labs(title = "Total Phosphorus by Phosphate Levels in Peter the & Paul Lakes",
       y = "Phosphate",
       x = "Phosphorus")+
  theme(
    plot.title = element_text(size = 14),
    legend.title = element_blank()
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 21947 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21947 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_smooth()`).
```



5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
#creating boxplot with temperature as y-axis
boxplot_temp <-
  ggplot(PeterPaul.chem.nutrients, aes(x= month,
                                       y=temperature_C,
                                       color=lakename))+
  geom_boxplot()

boxplot_temp1 <-
  boxplot_temp +
  labs(y="Temperature",
       x="Month")+
  theme(legend.title = element_blank())


#creating boxplot with TP as y-axis
```

```
boxplot_tp <-
  ggplot(PeterPaul.chem.nutrients, aes(x= month,
                                       y=tp_ug,
                                       color=lakename))+
  geom_boxplot()

boxplot_tp1 <-
  boxplot_tp +
  labs(y="Total Phosphorus",
       x="Month")


#creating boxplot with TN as y-axis
boxplot_tn <-
  ggplot(PeterPaul.chem.nutrients, aes(x= month,
                                       y=tn_ug,
                                       color=lakename))+
  geom_boxplot()

boxplot_tn1 <-
  boxplot_tn +
  labs(y="Total Nitrogen",
       x="Month")


#combining boxplots together and fixing aesthetics
cowplot1 <- plot_grid(
  boxplot_tn1 + theme(legend.position = "none"),
  boxplot_tp1+ theme(legend.position = "none"),
  boxplot_temp1+ theme(legend.position = "none"),
  align = 'vh',
  hjust = -1,
  nrow = 1)
```

```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

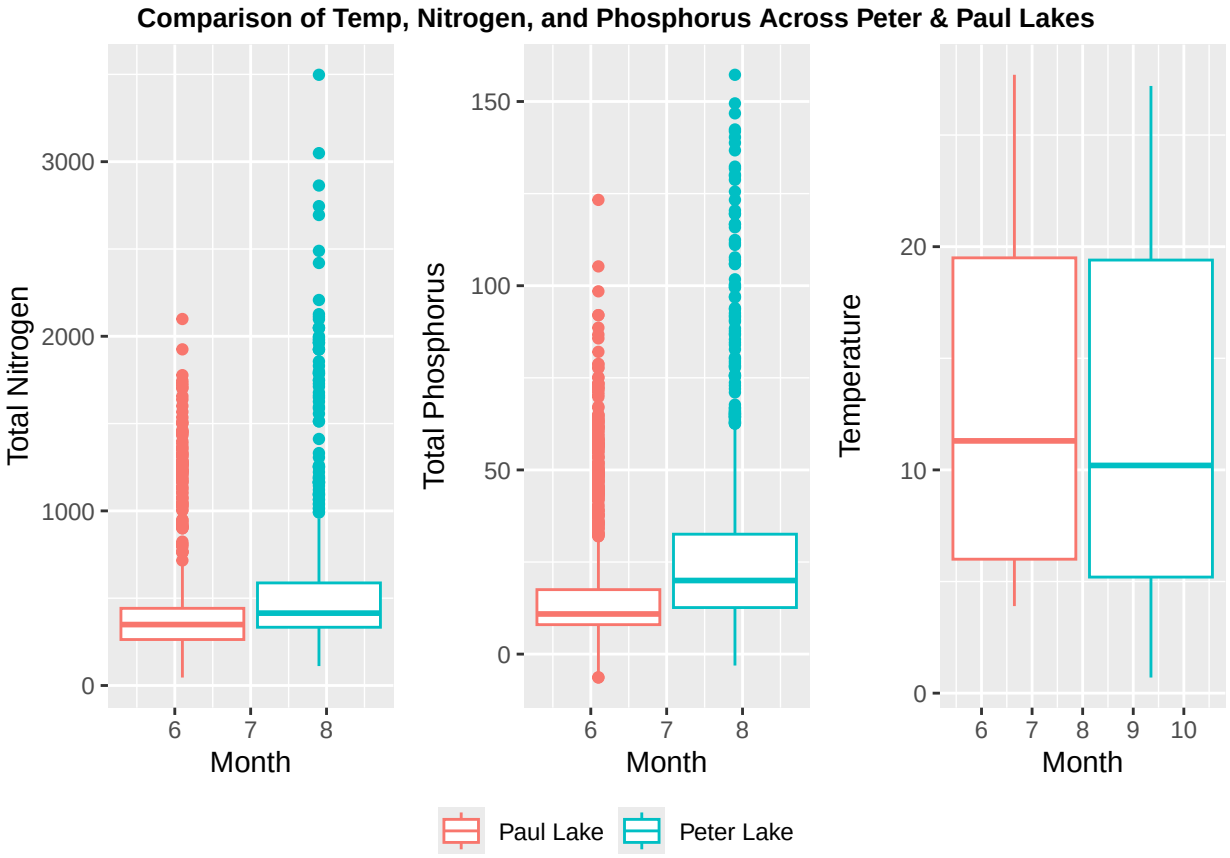
```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
#extracting legend from one of boxplots
legend <- get_legend(
  boxplot_temp1+
     guides(color = guide_legend(nrow = 1))+
     theme(legend.position = "right")
)
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning in get_plot_component(plot, "guide-box"): Multiple components found;
## returning the first one. To return all, use `return_all = TRUE`.
```

```r
#creating a new cowplot with new legend position
cowplot2 <-
  plot_grid(cowplot1, legend, ncol = 1, rel_heights = c(1, .1))

#creating a new cowplot with an aesthetic title
cowplot2 +
  labs(title = "Comparison of Temp, Nitrogen, and Phosphorus Across Peter & Paul Lakes")+
  theme(
     plot.title = element_text(size = 10.25, face = "bold")
  )
```



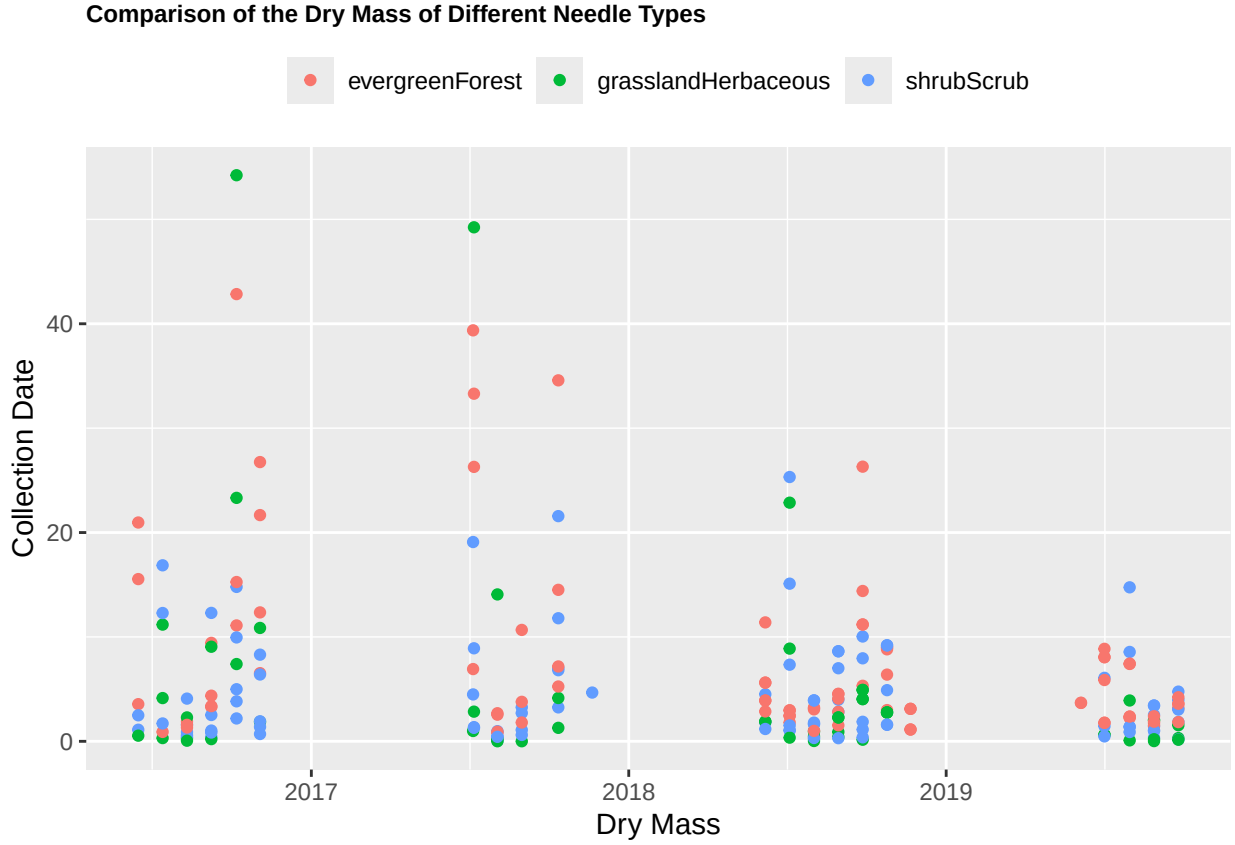
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: While the catalogued temperature between both lakes seemed fairly similar according to the months recorded, total phosphorus and total nitrogen levels were both higher in Peter Lake than in Paul Lake.

6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

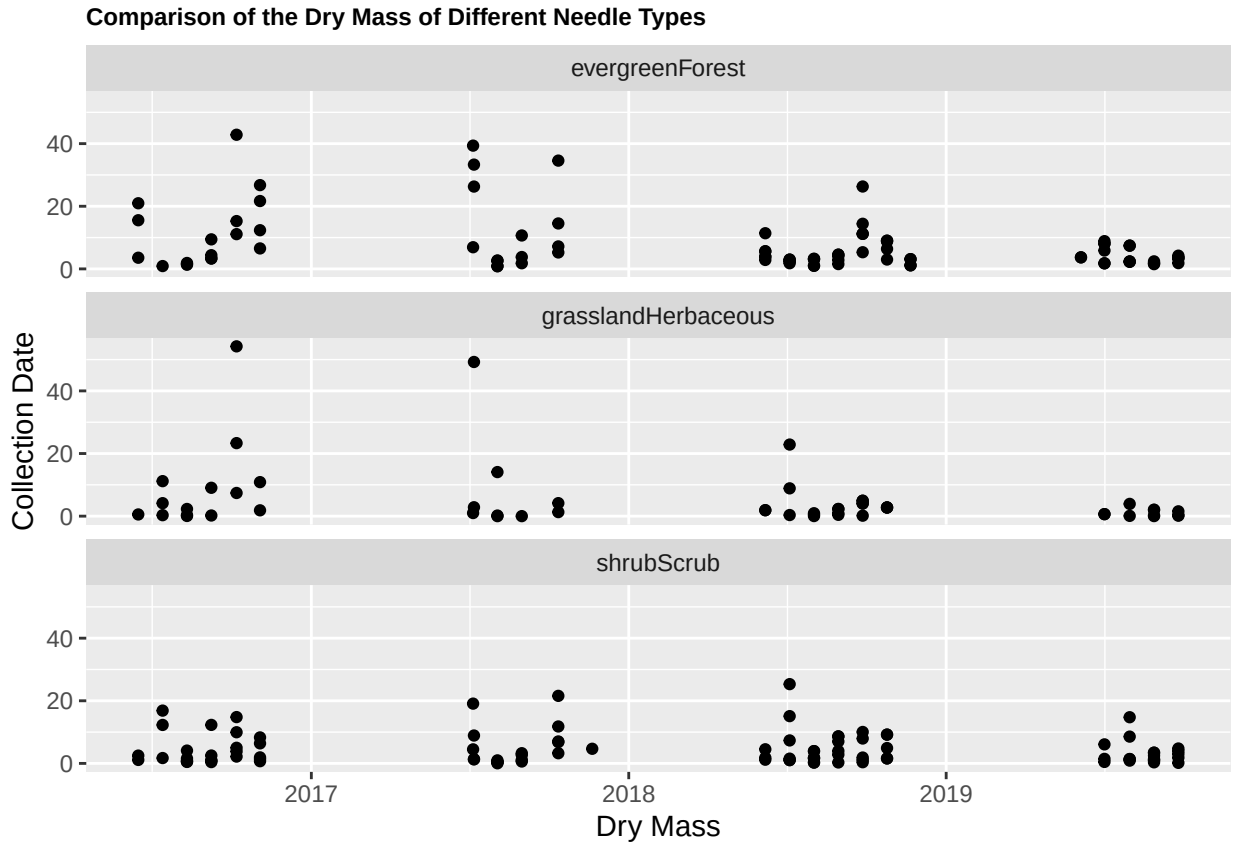
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
# plotting Needles functional group by date and dry mass
Litter1 <- NIWO.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
      color=nlcdClass)
  )+
  geom_point()

#adjusting aesthetics and adding titles
Litter1 +
  labs(
    title="Comparison of the Dry Mass of Different Needle Types",
    x="Dry Mass",
    y="Collection Date")+
  theme(legend.title = element_blank(),
        plot.title=element_text(size=9))
```




```
#7
#mapping separated NLCD Classes
Litter3 <- NIWO.litter %>%
  filter(functionalGroup == "Needles") %>%
```

```
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass)
  )+
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow=3)

#adjusting aesthetics and adding titles
Litter3 +
  labs(
    title="Comparison of the Dry Mass of Different Needle Types",
    x="Dry Mass",
    y="Collection Date")+
  theme(legend.title = element_blank(),
        plot.title=element_text(size=9))
```



Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer: I think plot 6 is more effective because of the fact that all three NLCD Classes are plotted together. I think the different colors being plotted on the same graph allow for a better visualization of the comparison between all three classes.