

Assignment 5: Data Visualization

Ashton Cloer

Fall 2023

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 setting up session and reading in data files
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.3      v readr      2.1.4
## v forcats    1.0.0      v stringr    1.5.0
## v ggplot2    3.4.3      v tibble     3.2.1
## v lubridate  1.9.2      v tidyr      1.3.0
## 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(here)
```

```
## here() starts at /Users/ashtoncloer/EDE_Fall2023
```

```
library(cowplot)
```

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

```
library(ggthemes)
```

```
##  
## Attaching package: 'ggthemes'  
##  
## The following object is masked from 'package:cowplot':  
##  
## theme_map
```

```
getwd()
```

```
## [1] "/Users/ashtoncloer/EDE_Fall2023"
```

```
here()
```

```
## [1] "/Users/ashtoncloer/EDE_Fall2023"
```

```
PeterPaul.chem.nutrients <-  
  read.csv(  
    here(  
      "Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),  
    stringsAsFactors = TRUE)
```

```
NiwotRidge.litter <-  
  read.csv(  
    here(  
      "Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),  
    stringsAsFactors = TRUE)
```

```
#2 changing to date format
```

```
PeterPaul.chem.nutrients$sampldate <- ymd(  
  PeterPaul.chem.nutrients$sampldate)  
NiwotRidge.litter$collectDate <- ymd(  
  NiwotRidge.litter$collectDate)
```

```
class(PeterPaul.chem.nutrients$sampldate)
```

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

```
class(NiwotRidge.litter$collectDate)
```

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

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 building plot theme by changing the axis text color,  
#the background, base text size  
#and font type and the legend position
```

```
mytheme1 <- theme_economist(  
  base_size = 14,  
  base_family = "Times") +  
  theme(  
    axis.text = element_text(  
      color = "gray"),  
    legend.position = "bottom")
```

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 a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4  
PhosphorusVsPhosphate <-  
  ggplot(PeterPaul.chem.nutrients,  
    aes(x = po4, y = tp_ug, #plotting total phosphorus by phosphate  
      color = lakename)) + #setting color of points by lakename  
  geom_point() +  
  xlim(0, 40) +  
  scale_color_manual(  
    values = c(  
      "#E7B800", "#FC4E07")) + #changing colors of points  
  geom_smooth(method = "lm", #adding line of best fit  
    color="black") + #coloring line of best fit black  
  ggtitle(
```

```

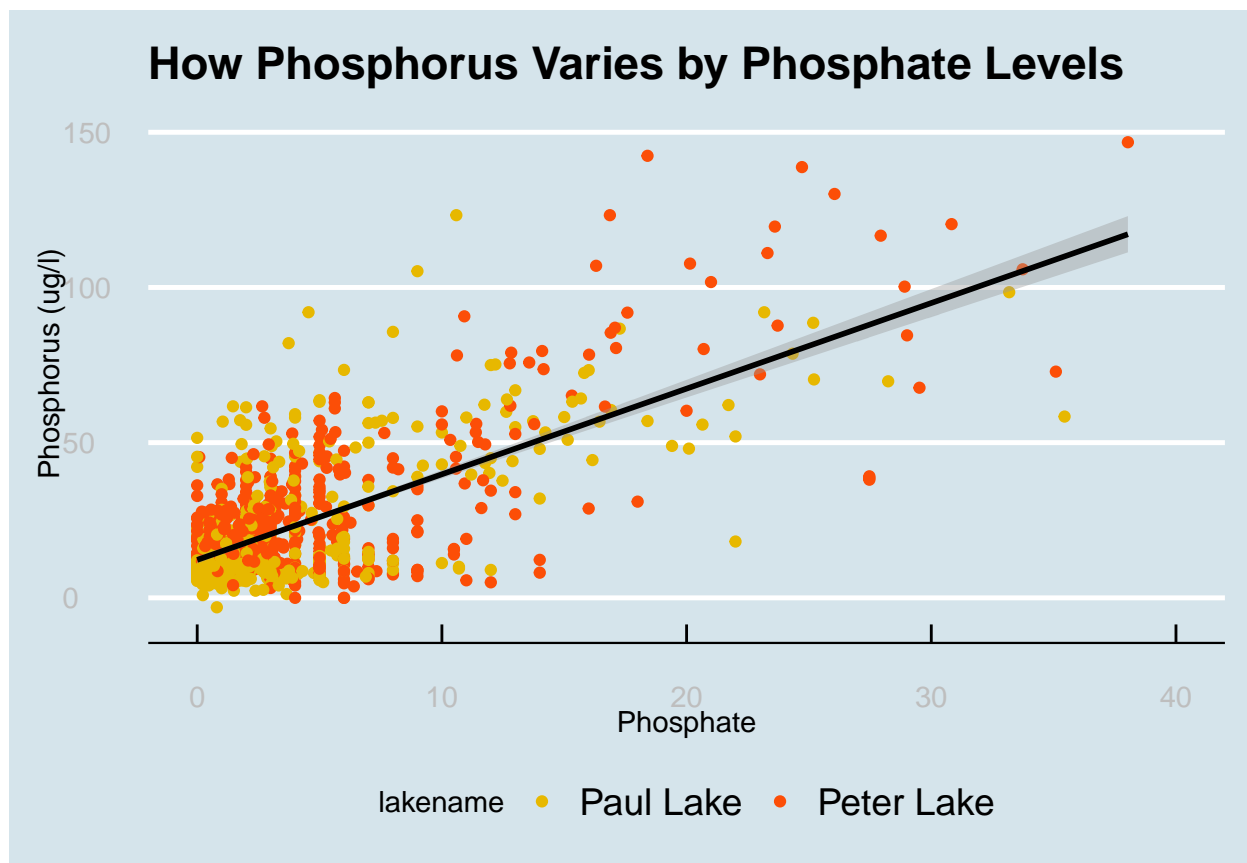
  "How Phosphorus Varies by Phosphate Levels") + #adding title
labs(
  x = "Phosphate", #adding labels to xaxis
  y = "Phosphorus (ug/l)" + #adding labels to yaxis
  mytheme1 #adding my set theme
print(PhosphorusVsPhosphate)

```

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

```
## Warning: Removed 21948 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 21948 rows containing missing values ('geom_point()').
```



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.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months; see <https://r-lang.com/month-abb-in-r-with-example>

```

#5
factor(
  PeterPaul.chem.nutrients$month,
  levels=1:12,
  labels=month.abb) #converting month to factor

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

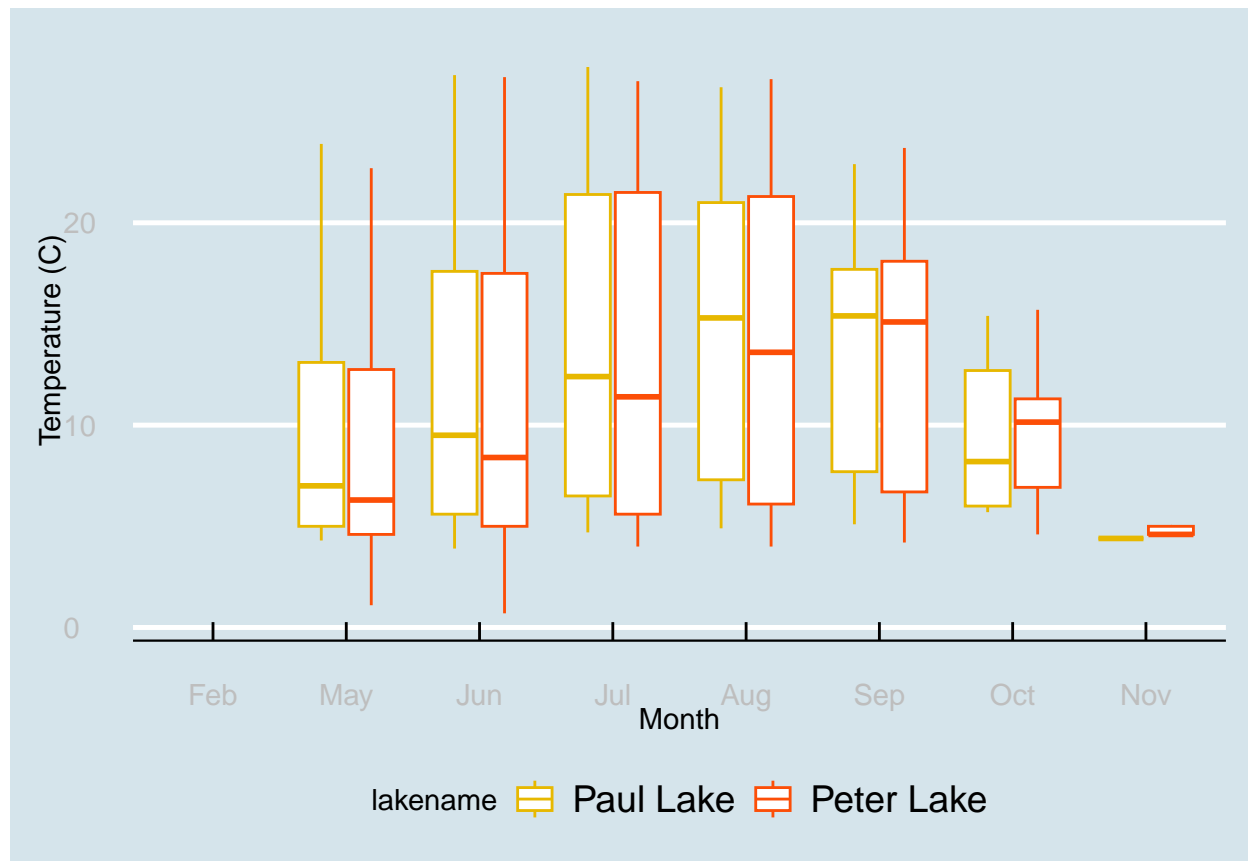
```
## [22357] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Aug Aug Aug Aug
## [22375] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22393] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Sep Sep Sep Sep
## [22411] Sep Sep Sep Sep Sep May May May May May May May Jun Jun Jun Jun Jun Jun Jun Jun
## [22429] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22447] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jul Jul Jul Jul Jul Jul Jul
## [22465] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22483] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Aug Aug Aug Aug
## [22501] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22519] Aug Aug Aug Aug Aug Aug Aug Aug Aug Sep Sep Sep Sep Sep Sep Sep Sep May May May
## [22537] May May May Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22555] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jul Jul
## [22573] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22591] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Aug Aug Aug Aug Aug Aug Aug
## [22609] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22627] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug May May May
## [22645] May May May May Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jul Jul Jul Jul Jul
## [22663] Jul Jul Aug Aug Aug Aug Aug Aug Aug Aug Aug May May May May May May May May May
## [22681] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22699] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22717] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jul
## [22735] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22753] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22771] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Aug Aug Aug Aug
## [22789] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22807] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22825] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Sep Sep May
## [22843] May May May May May May May May May Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22861] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22879] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun
## [22897] Jun Jun Jun Jun Jun Jun Jun Jun Jun Jun Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22915] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22933] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul
## [22951] Jul Jul Jul Jul Jul Jul Jul Jul Jul Jul Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22969] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [22987] Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug Aug
## [23005] Aug Aug Aug Aug
## Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

```
temp <-
  ggplot(PeterPaul.chem.nutrients,
    aes(x = factor(
      PeterPaul.chem.nutrients$month,
      levels=1:12,
      labels=month.abb), y = temperature_C)) +
  geom_boxplot(
    aes(color = lakename)) +
    #scale_x_discrete(
      #name="month", drop=FALSE) +
  scale_color_manual(
    values = c(
      "#E7B800", "#FC4E07")) +
  labs(x = "Month",
    y = "Temperature (C)") +
```

```
mytheme1
print(temp)
```

```
## Warning: Use of 'PeterPaul.chem.nutrients$month' is discouraged.
## i Use 'month' instead.
```

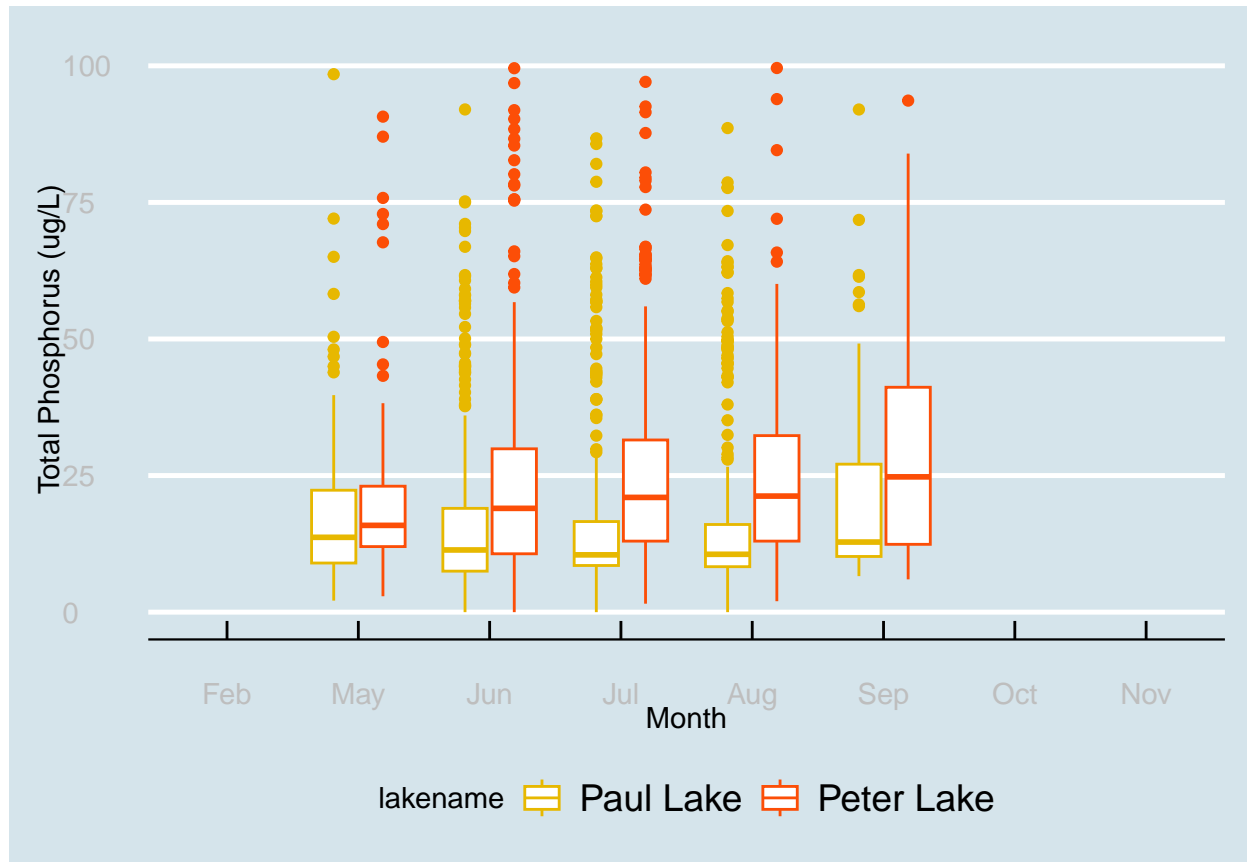
```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```



```
TP <-
ggplot(PeterPaul.chem.nutrients, aes(
  x = factor(
    PeterPaul.chem.nutrients$month,
    levels=1:12,
    labels=month.abb), y = tp_ug)) +
geom_boxplot(aes(color = lakename)) +
  #scale_x_discrete(
    #name="month", drop=FALSE) +
scale_color_manual(
  values = c(
    "#E7B800", "#FC4E07")) +
labs(x = "Month",
  y = "Total Phosphorus (ug/L)") +
ylim(0, 100) +
mytheme1
print(TP)
```

```
## Warning: Use of 'PeterPaul.chem.nutrients$month' is discouraged.
## i Use 'month' instead.
```

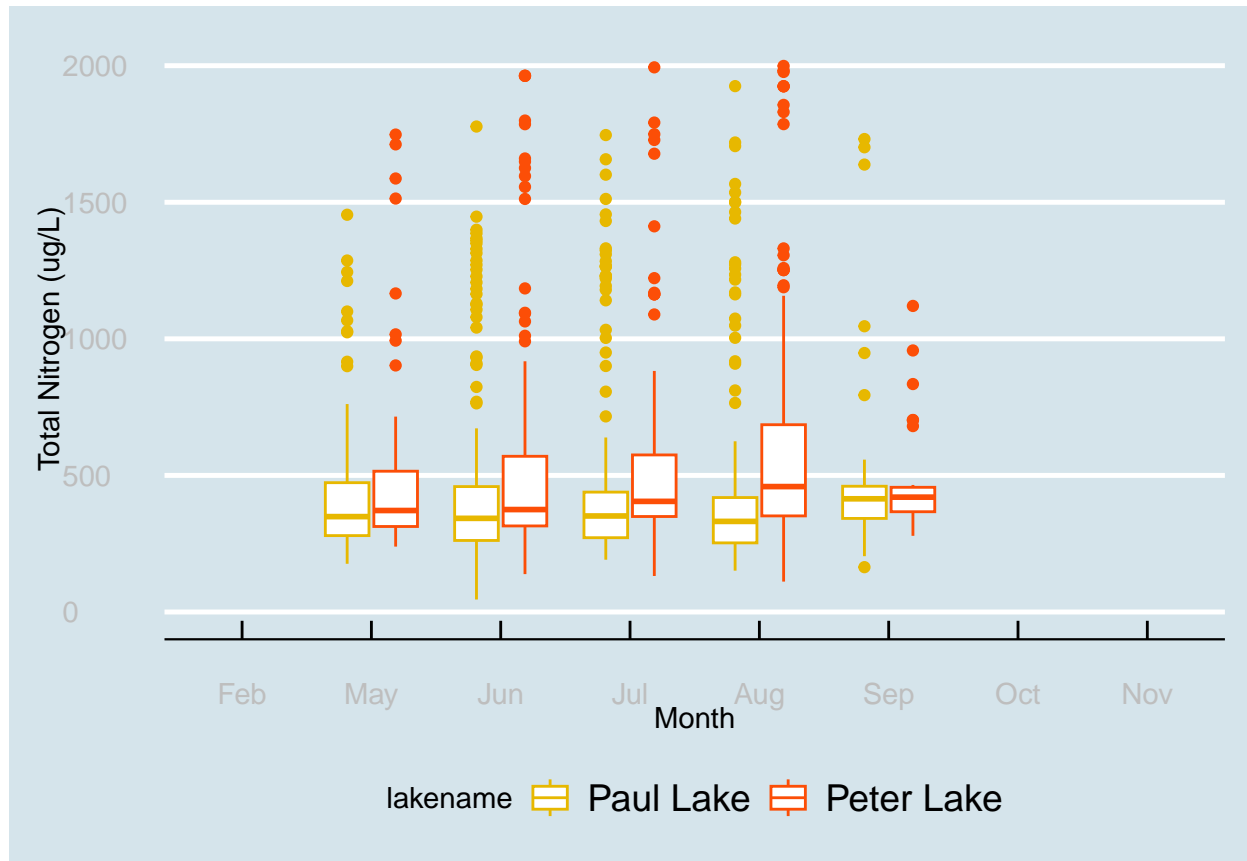
```
## Warning: Removed 20799 rows containing non-finite values ('stat_boxplot()').
```



```
TN <-
  ggplot(PeterPaul.chem.nutrients, aes(
    x = factor(
      PeterPaul.chem.nutrients$month,
      levels=1:12,
      labels=month.abb), y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  #scale_x_discrete(
    #name="month", drop=FALSE) +
  scale_color_manual(
    values = c(
      "#E7B800", "#FC4E07")) +
  labs(x = "Month",
    y = "Total Nitrogen (ug/L)") +
  ylim(0, 2000) +
  mytheme1
print(TN)
```

```
## Warning: Use of 'PeterPaul.chem.nutrients$month' is discouraged.
## i Use 'month' instead.
```

```
## Warning: Removed 21598 rows containing non-finite values ('stat_boxplot()').
```



```
combined_plot <- plot_grid(
  TN + theme(legend.position = "none"),
  TP + theme(legend.position = "none"),
  temp,
  align = "vh",
  nrow = 3, rel_heights=c(1, 1, 1.25)
)
```

```
## Warning: Use of 'PeterPaul.chem.nutrients$month' is discouraged.
## i Use 'month' instead.
## Removed 21598 rows containing non-finite values ('stat_boxplot()').
```

```
## Warning: Use of 'PeterPaul.chem.nutrients$month' is discouraged.
## i Use 'month' instead.
```

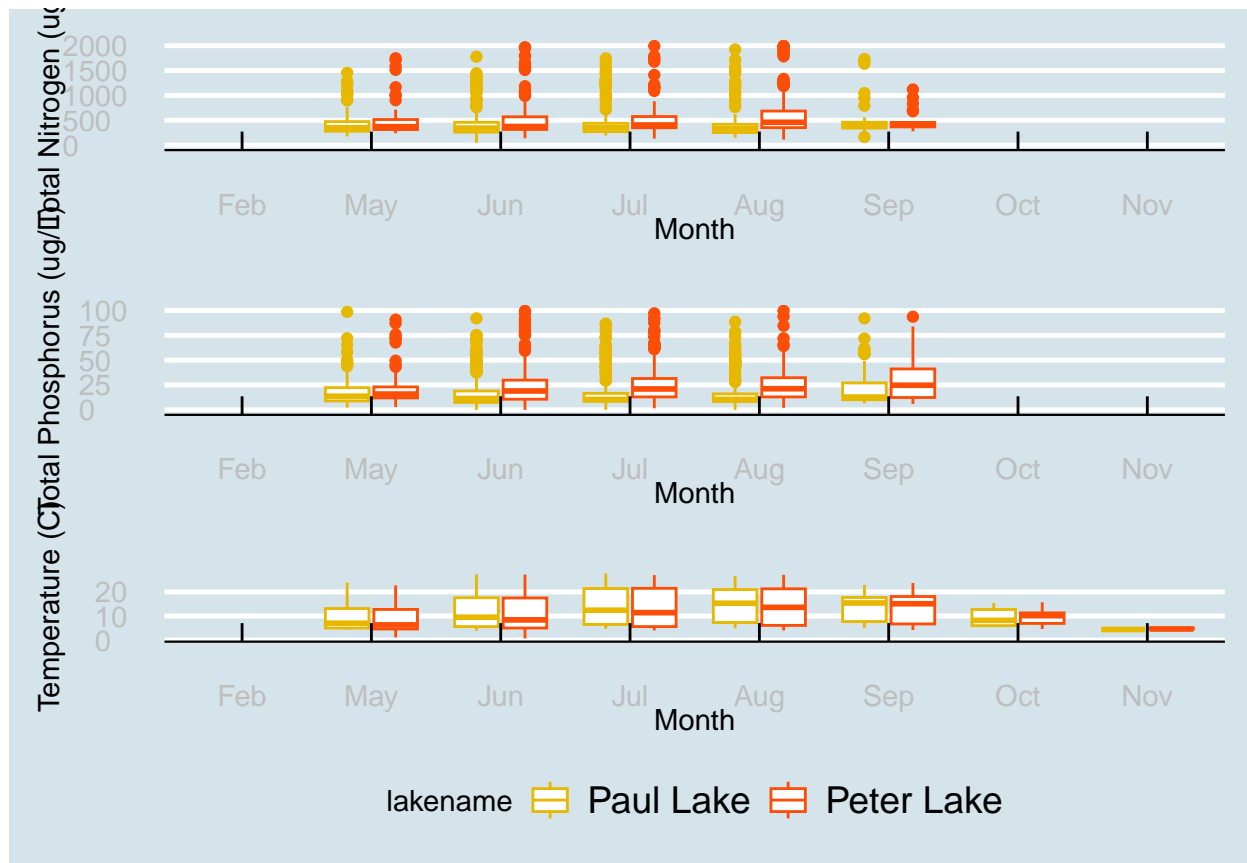
```
## Warning: Removed 20799 rows containing non-finite values ('stat_boxplot()').
```

```
## Warning: Use of 'PeterPaul.chem.nutrients$month' is discouraged.
## i Use 'month' instead.
```

```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```

```
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```
print(
  combined_plot)
```



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

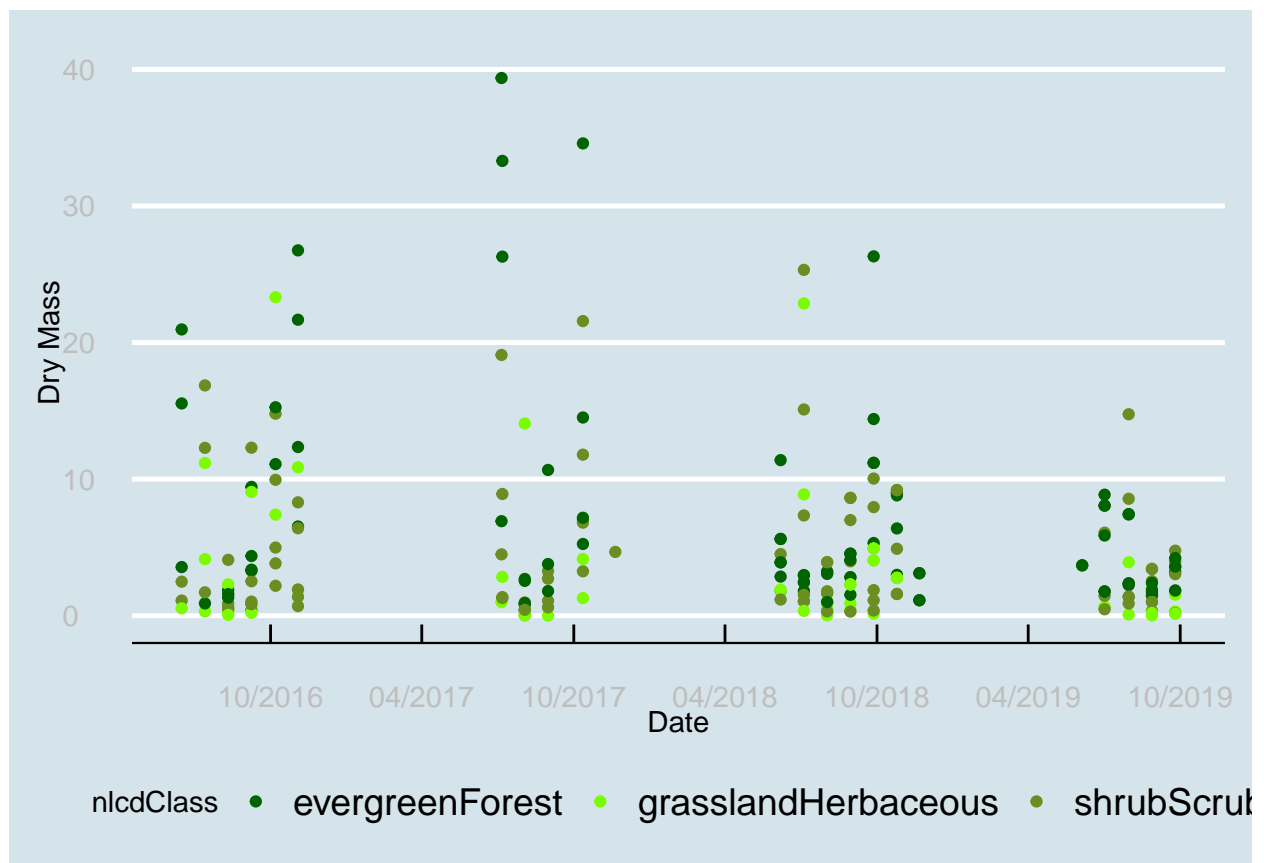
Answer: The temperature at Peter and Paul lakes both seem to follow the same general pattern, increasing in May and June to a maximum temperature in July and August, followed by a decrease in temperature through November. For total phosphorus, Peter lake seems to generally increase from May to a maximum in September. However, Paul lake seems to be highest in September and then generally decreases until it reaches a minimum value in July and August. For total nitrogen, Peter lake seems to increase starting in May until reaching a maximum in August and then a minimum in September. Paul lake is generally highest in May and then reaches a minimum in August and begins increasing again in September.

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
needles_litter <-
  ggplot(subset(NiwotRidge.litter,
               functionalGroup == "Needles"),
         aes(x = collectDate,
             y = dryMass,
             color = nlcdClass)) +
  geom_point() +
  scale_color_manual(
    values = c(
      "#006400",
      "#7CFC00",
      "#6B8E23")) +
  labs(x = "Date",
       y = "Dry Mass") +
  scale_x_date(
    date_breaks = "6 months",
    date_labels = "%m/%Y") +
  ylim(0, 40) +
  theme1

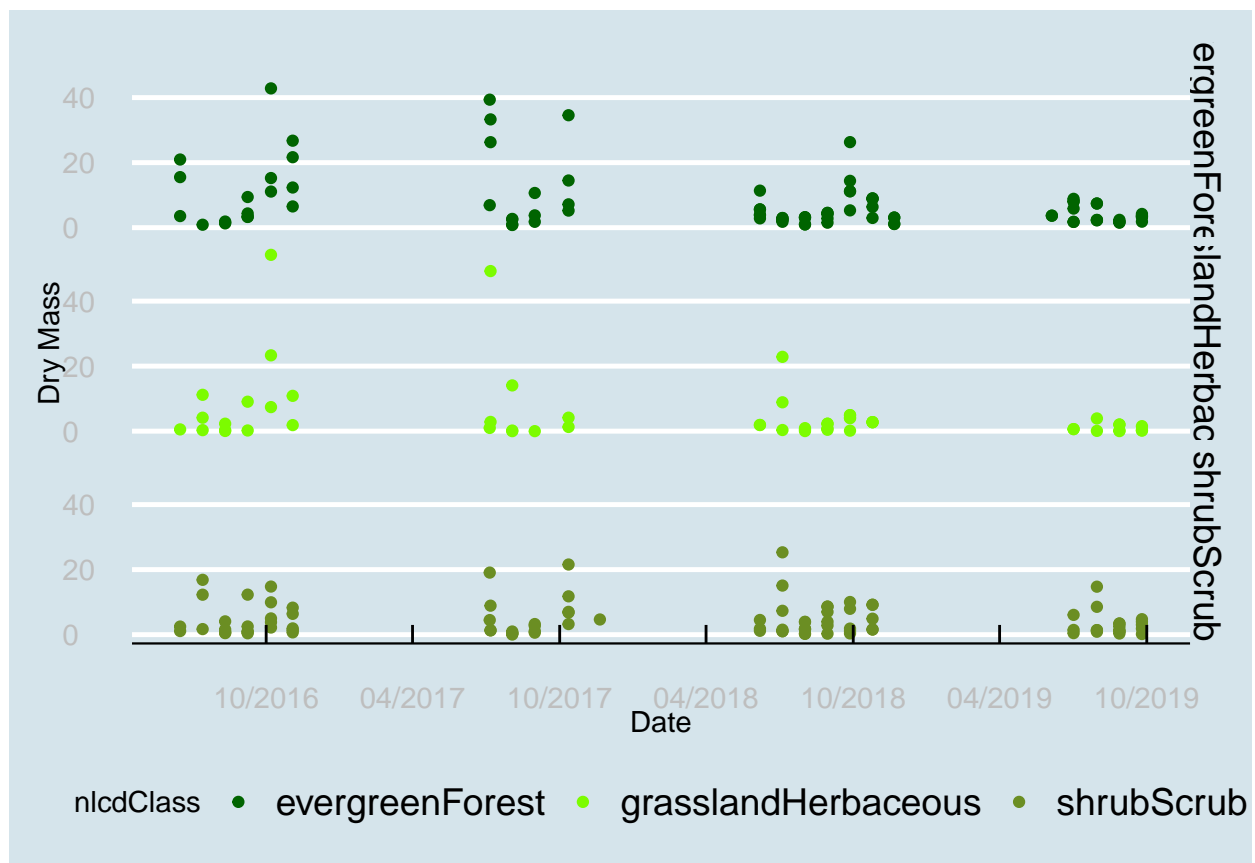
print(needles_litter)
```

```
## Warning: Removed 3 rows containing missing values ('geom_point()').
```



```
#7
needles_litter2 <-
  ggplot(subset(
    NiwotRidge.litter,
    functionalGroup == "Needles"),
    aes(x = collectDate,
        y = dryMass,
        color = nlcdClass)) +
  geom_point() +
  facet_grid(vars(nlcdClass)) +
  scale_color_manual(
    values = c(
      "#006400",
      "#7CFC00",
      "#6B8E23")) +
  labs(x = "Date",
       y = "Dry Mass") +
  scale_x_date(
    date_breaks = "6 months",
    date_labels = "%m/%Y") +
  theme_minimal

print(needles_litter2)
```



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

Answer: Plot 6 is more effective for directly comparing the drymass of the various classes, since

each class is plotted on the same graph. For example, in October of 2016, you can directly compare the drymass values to see that Evergreen forest had the highest drymass occurrence in that month and grasslands had the smallest. However, plot 7 is more effective for identifying trends in the individual classes. For example, plot 7 makes it easier to see that the drymass of the grassland class peaked in 2016, followed by a decrease in 2017 and another peak in 2018.