Assignment 5: Data Visualization

Max Hermanson

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 23 at 11:59 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and cowplot packages. Upload the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) and the processed data file for the Niwot Ridge litter dataset.

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

```
library(cowplot)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

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

```
library(ggridges)


getwd()
```

```
## [1] "/Users/mothership/Desktop/EDA_21/Environmental_Data_Analytics_2021_0215/Assignments"
```

```
lake_processed <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
lake_gathered <- read.csv("../Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
```

```
litter.df <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

class(lake_processed)
```

```
## [1] "data.frame"
```

Define your theme

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

```
maxsTheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

#theme_set(maxsTheme)
```

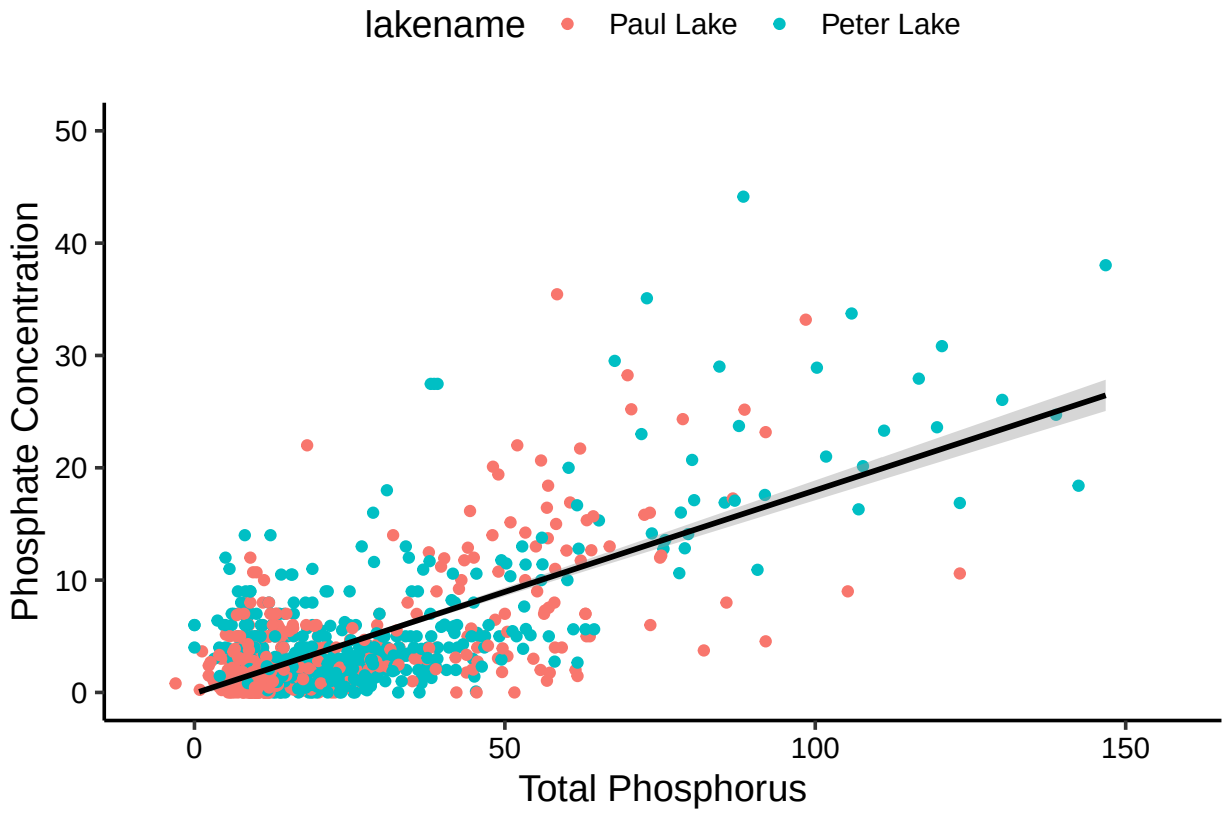
Create graphs

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.

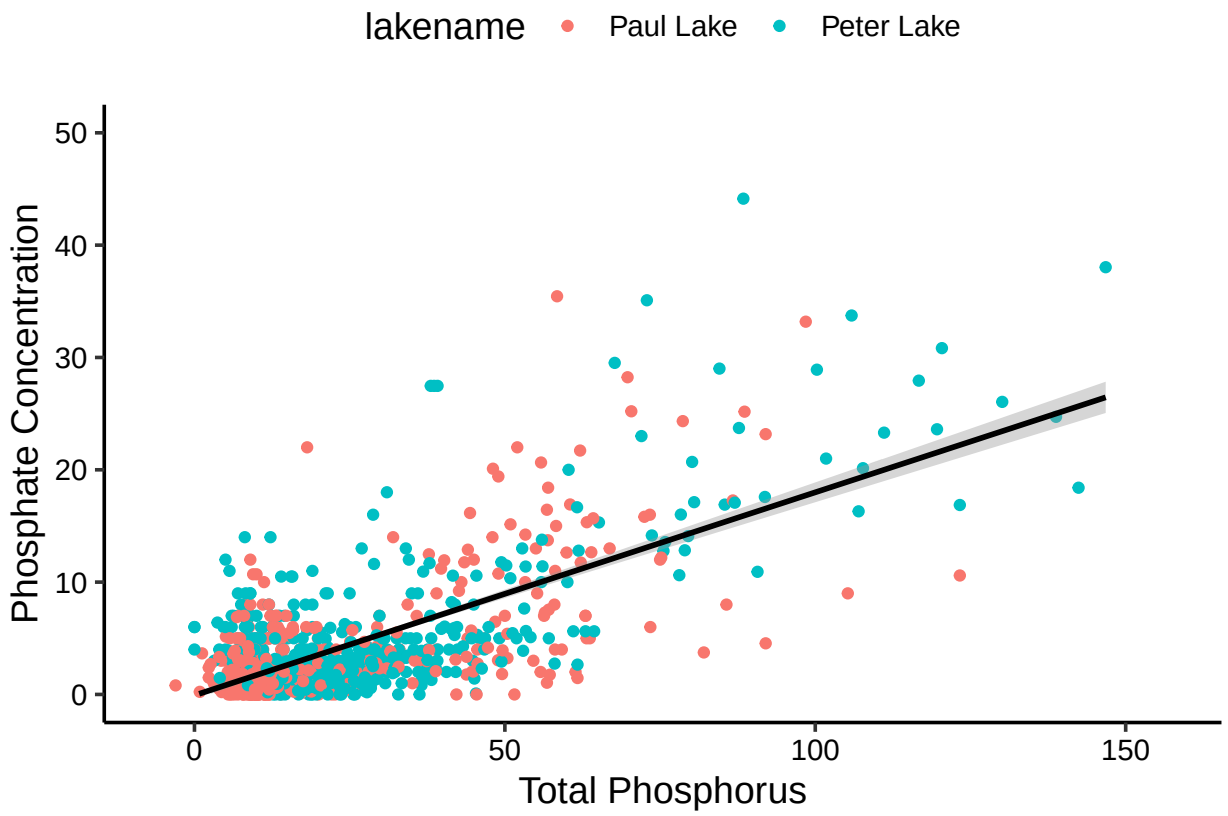
```
phosphorus_scatter <- ggplot(lake_processed,aes(x= tp_ug, y=po4))+
         geom_point(aes(color = lakename))+
  maxsTheme+
  xlab("Total Phosphorus")+
  ylab("Phosphate Concentration")+
  geom_smooth(method=lm, color = "black")+
  ylim(0,50)+
  labs(lakename = "Lake Name:") #cant get legend title channge to work

print(phosphorus_scatter)
```

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

## Warning: Removed 21947 rows containing non-finite values (stat_smooth).

## Warning: Removed 21947 rows containing missing values (geom_point).

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




```
phosphorus_scatter+ labs()
```

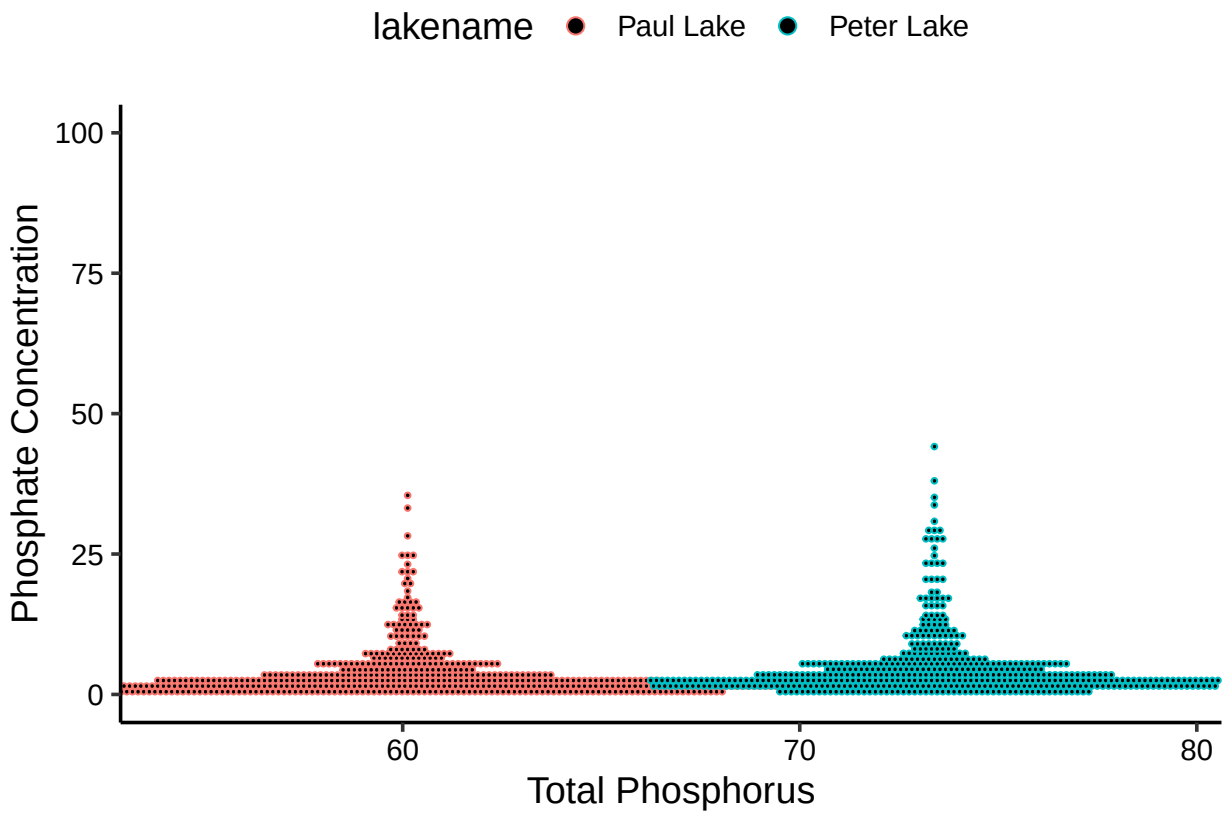
```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 21947 rows containing non-finite values (stat_smooth).
## Warning: Removed 21947 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_smooth).
```




```
#ignore the next two chunks
phosphorus_dot <- ggplot(lake_processed,aes(x= tp_ug, y=po4))+
         geom_dotplot(aes(color = lakename), binaxis = "y", binwidth = 1 , stackdir = "center", positio
  maxsTheme+
  xlab("Total Phosphorus")+
  ylab("Phosphate Concentration")+
  ylim(0,100)
print(phosphorus_dot)
```

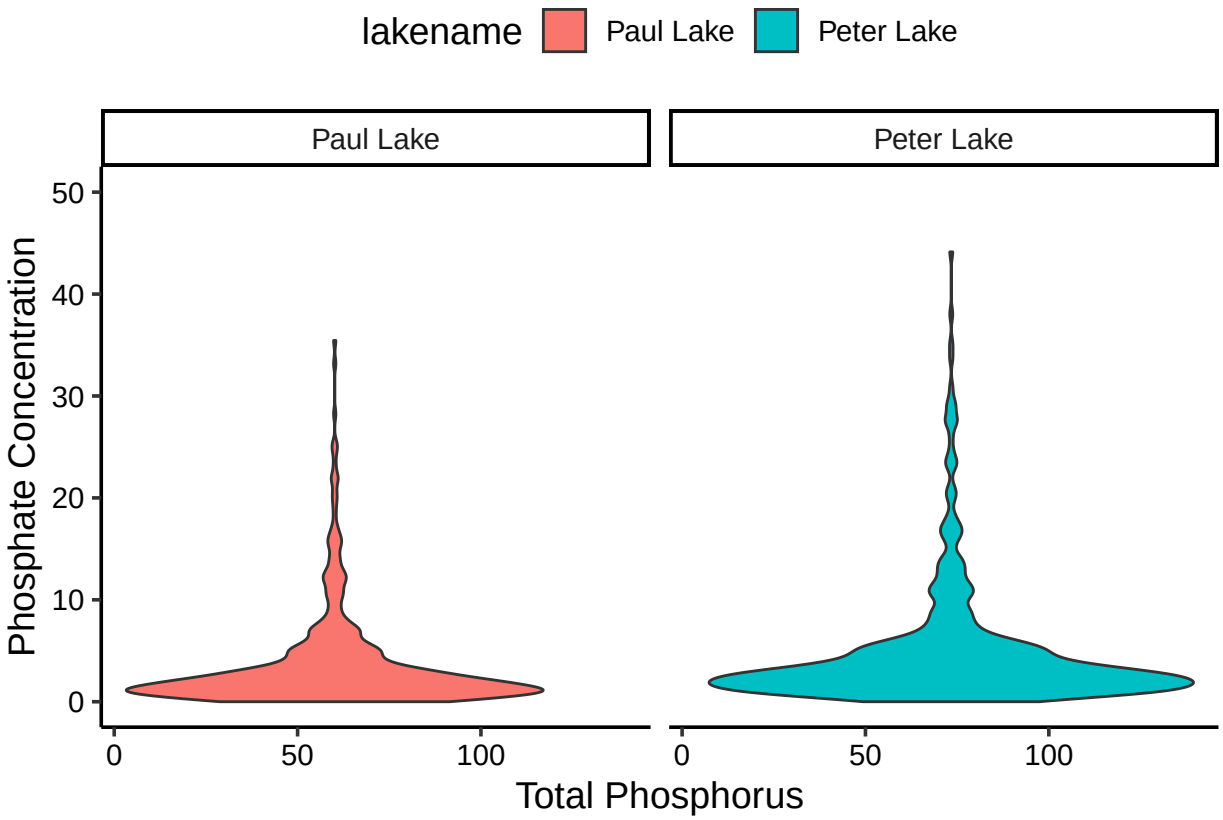
```
## Warning: Removed 21839 rows containing non-finite values (stat_bindot).
```

```
## Warning: Removed 108 rows containing non-finite values (stat_bindot).
```




```
#my choice
phosphorus_violin <- ggplot(lake_processed,aes(x= tp_ug, y=po4, fill = lakename))+
  geom_violin()+
  ylim(0,50)+
  facet_wrap(vars(lakename), nrow = 1)+
  xlab("Total Phosphorus")+
  ylab("Phosphate Concentration")+
  maxsTheme
print(phosphorus_violin)
```

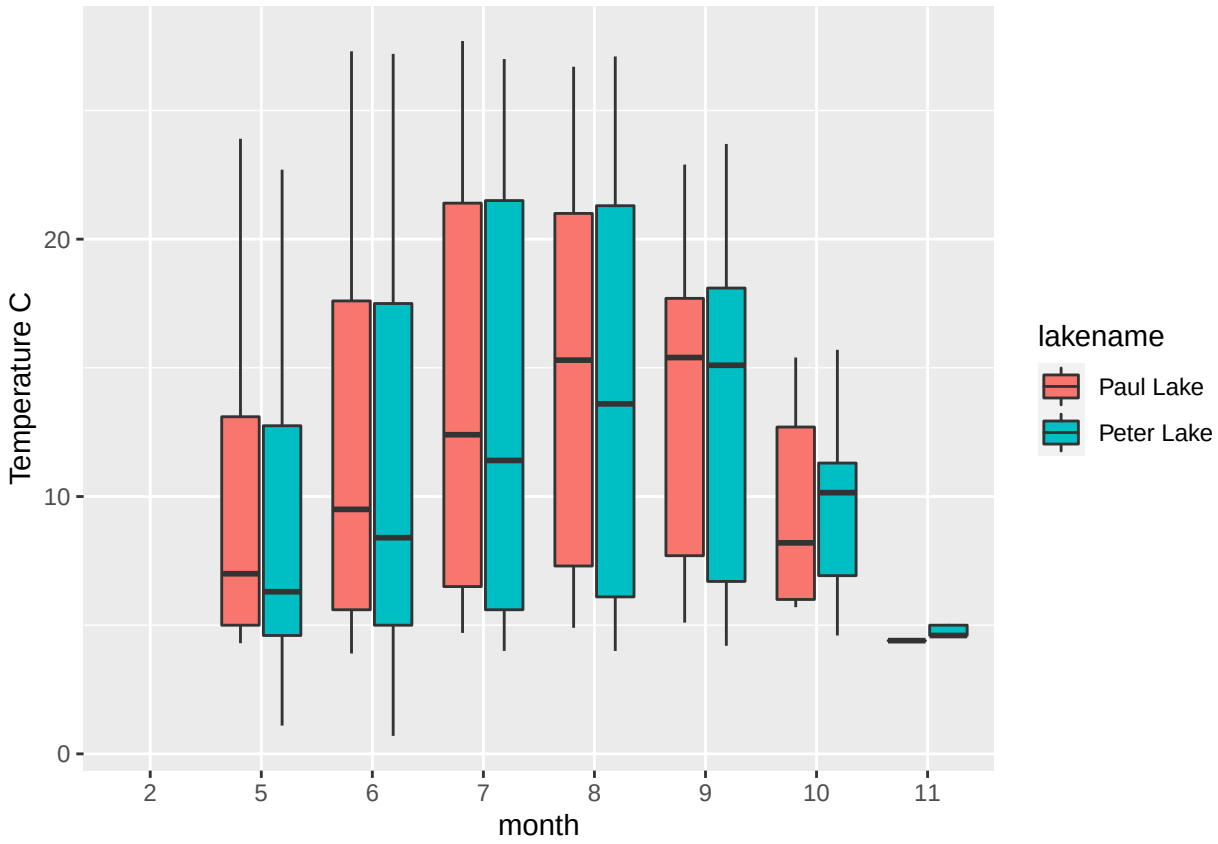
```
## Warning: Removed 21947 rows containing non-finite values (stat_ydensity).
```



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.

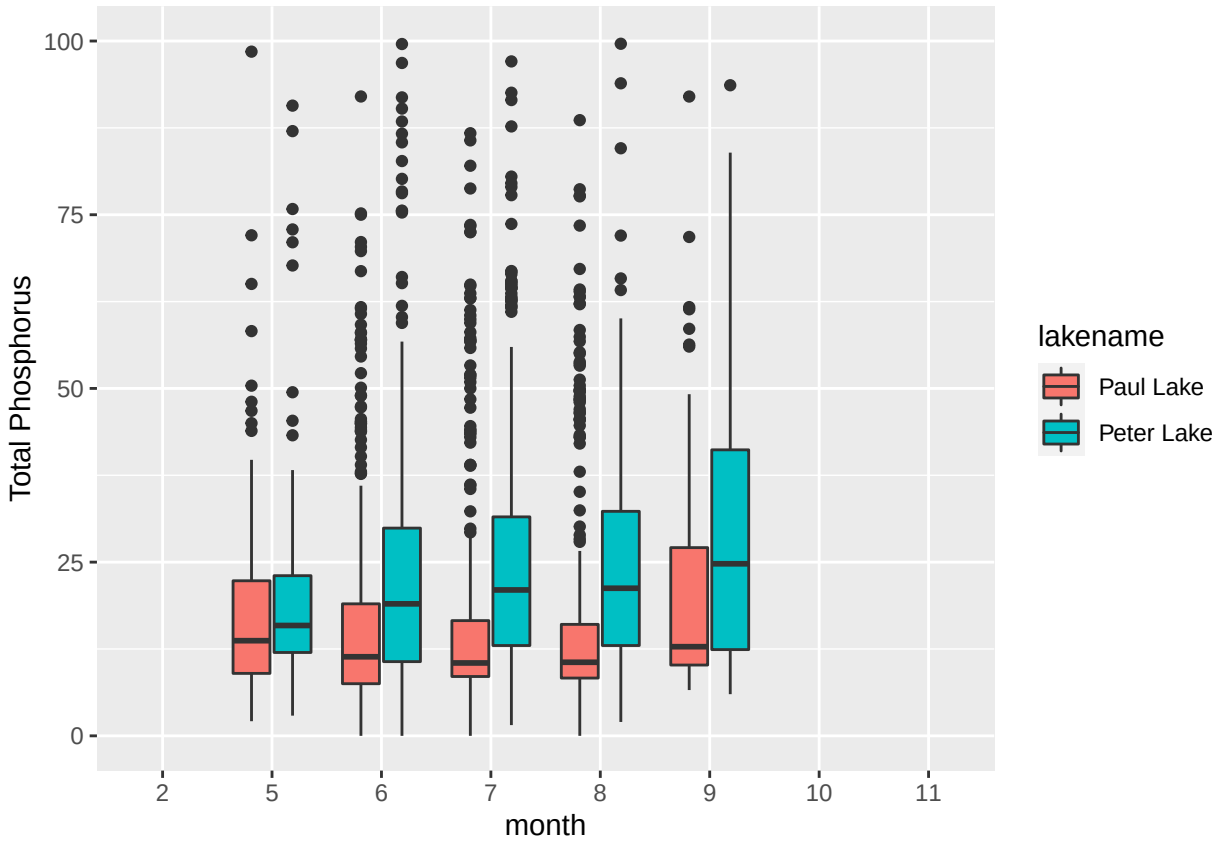
```
temp_box <- ggplot(lake_processed, aes(x = as.factor(month), y = temperature_C))+
  geom_boxplot(aes(fill = lakename ))+
  xlab("Month")+
  ylab("Temperature C")+
  scale_x_discrete("month")

print(temp_box)
```

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



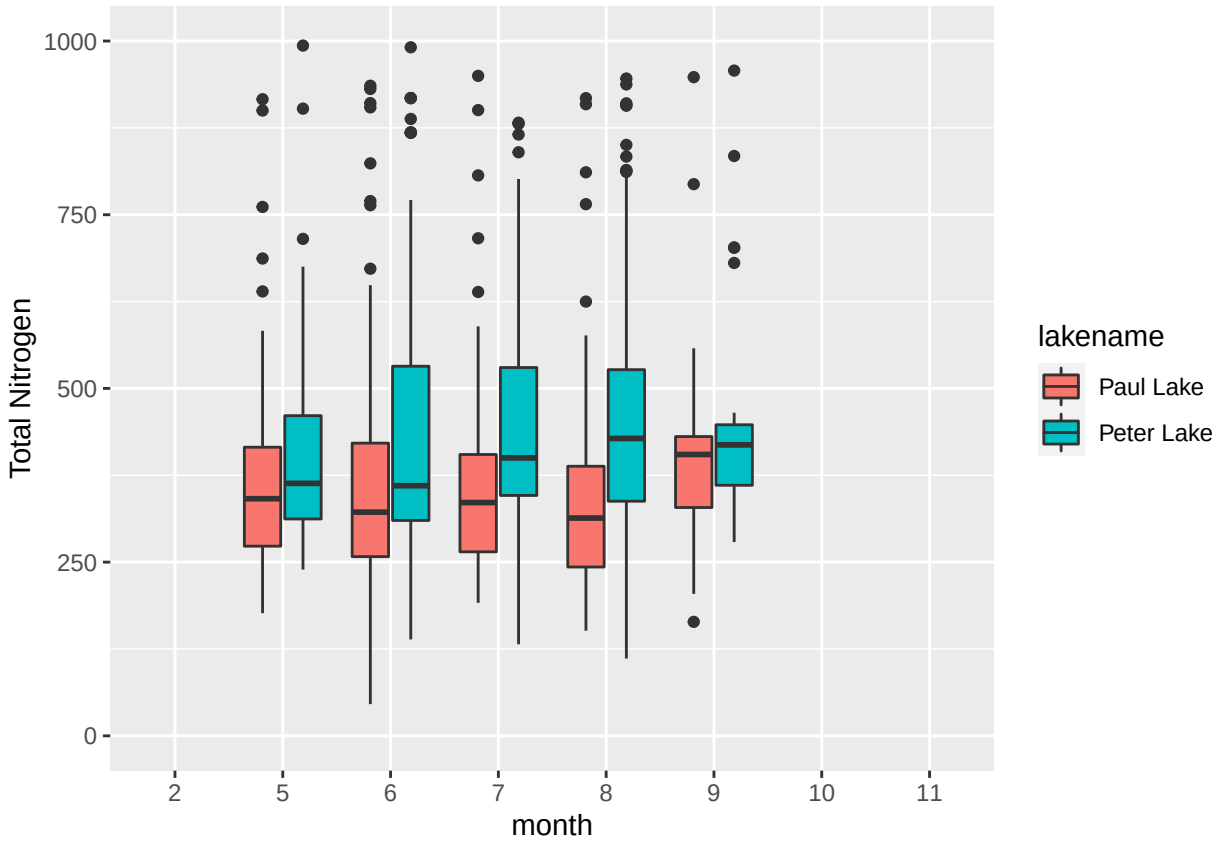

```
TP_box <- ggplot(lake_processed, aes(x = as.factor(month), y = tp_ug))+
  geom_boxplot(aes(fill = lakename))+
  xlab("Month")+
  ylab("Total Phosphorus")+
  scale_x_discrete("month")+
  ylim(0,100)


print(TP_box)
```

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




```
TN_box <- ggplot(lake_processed, aes(x = as.factor(month), y = tn_ug))+
  geom_boxplot(aes(fill = lakename))+
  ylim(0, 1000)+
  xlab("Month")+
  ylab("Total Nitrogen")+
  scale_x_discrete("month")



print(TN_box)
```

```
## Warning: Removed 21730 rows containing non-finite values (stat_boxplot).
```




```
library(cowplot)
combined_cow <- plot_grid(temp_box+theme(legend.position="none"), TP_box+theme(legend.position = "none"
```

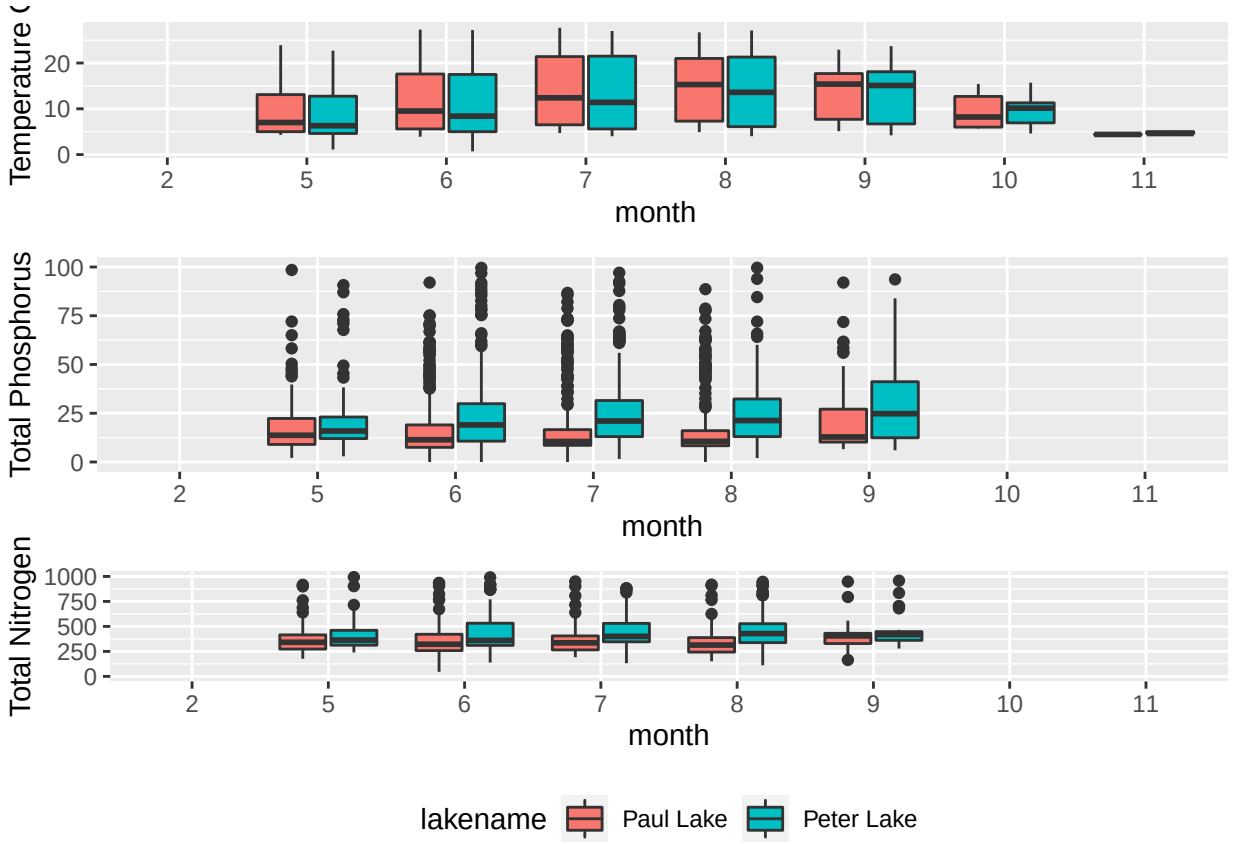
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```

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

```
## Warning: Removed 21730 rows containing non-finite values (stat_boxplot).
```

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

```
print(combined_cow)
```




```
ggsave("../Output/A05_cowplot.png", combined_cow, dpi = 300)
```

```
## Saving 6.5 x 4.5 in image
```

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

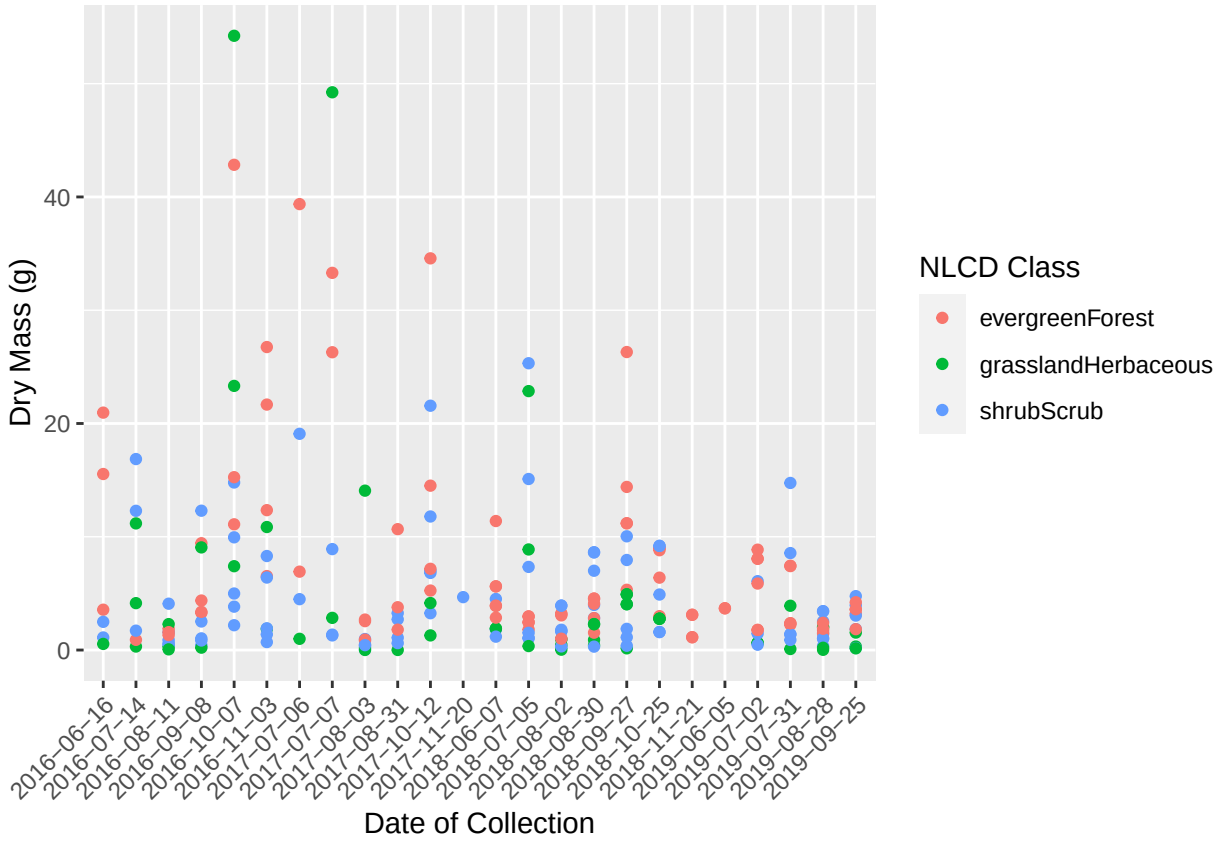
> Answer: temperature varies most and is highest during summer months, whereas temperature is lowest and most constant during winter months. The temperature is very similar between the two lakes. Total P reaches its peak in september and remains relatively constant in other months for both Paul and Peter Lake. Paul Lake had lower total P than Peter Lake every month. Trends in total nitrogen (TN): Peter Lake had higher TN levels than Paul Lake. THe only other discernabile trend in the TN data was that TN gradually increased in Peter Lake from May to October, and then sharply dropped; also the range of TN values (within the box and whiskers) was smallest in the Fall.

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)

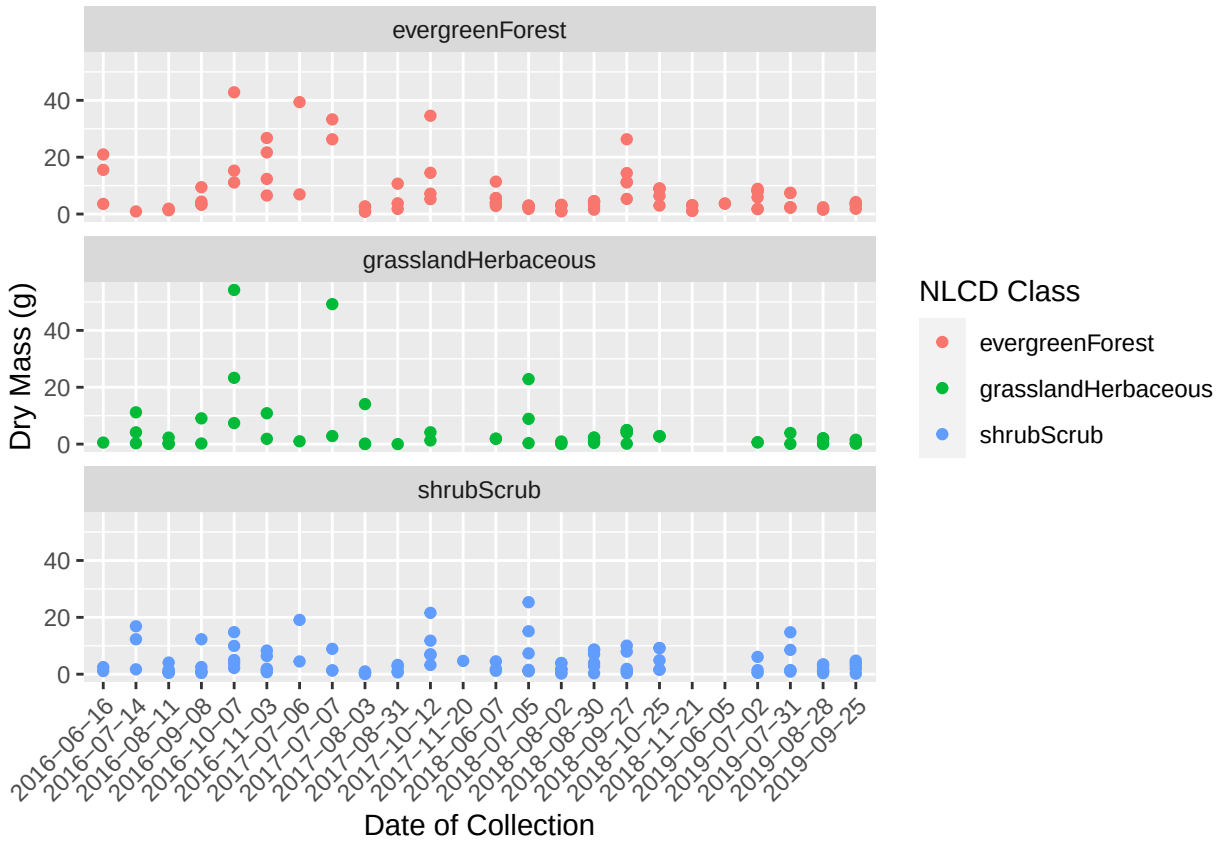
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
# Dry mass = Y axis, Collect Date = X-Axis? > facet wrap problem: make sure you use vars command
# Faced grid, more than 1 variable
# Facet wrap,


litter.df_needles <- litter.df %>%
  filter(functionalGroup == "Needles")
```

```
needle_plot <- ggplot(litter.df_needles, aes(x= collectDate, y = dryMass))+
  geom_point(aes(color = nlcdClass))+
  theme(axis.text.x=element_text(angle = 45, hjust = 1 ))+
  xlab("Date of Collection")+
  ylab("Dry Mass (g)")+
  scale_color_discrete(name="NLCD Class")
needle_plot
```




```
needle_plot2 <- ggplot(litter.df_needles, aes(x= collectDate, y = dryMass))+
  geom_point(aes(color = nlcdClass))+
  theme(axis.text.x=element_text(angle = 45, hjust = 1 ))+
  xlab("Date of Collection")+
  ylab("Dry Mass (g)")+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  scale_color_discrete(name="NLCD Class")   # creates legend
needle_plot2
```




```
#example_plot <- ggplot(df, aes(x= variable2, y=variable3))+
 # geom_point()
  #facet_wrap(vars(variable1), nrow = 3)+
#example_plot + labs(variable1 = "Legend")
```

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

> Answer: Plot 7. There were far too many (overlapping) data points in Plot 6 to effectively discern patterns/differences between the nlcd classes. The facet approach allows for easier comparison of similarities and differences at each individual date and across time across NLCD classes, AND makes it easier to examine trends within individual nlcd classes.