

Assignment 5: Data Visualization

Rachel Gonsenhauser

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., “Salk_A05_DataVisualization.Rmd”) prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 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 (tidy and gathered) 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.

```
#1
getwd()

## [1] "/Users/rachelgonsenhauser/Documents/Environmental_Data_Analytics_2020"

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.2.1      v purrr   0.3.3
## v tibble  2.1.3      v dplyr  0.8.3
## v tidyr   1.0.0      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

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

library(cowplot)

##
## *****
## Note: As of version 1.0.0, cowplot does not change the
## default ggplot2 theme anymore. To recover the previous
```

```

##   behavior, execute:
##   theme_set(theme_cowplot())

## *****

PeterPaul.chem.nutrients <-
  read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.chem.nutrients.gathered <-
  read.csv("../Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
Litter <-
  read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(PeterPaul.chem.nutrients$sampledate)

## [1] "factor"
class(PeterPaul.chem.nutrients.gathered$sampledate)

## [1] "factor"
class(Litter$collectDate)

## [1] "factor"
PeterPaul.chem.nutrients$sampledate <-
  as.Date(PeterPaul.chem.nutrients$sampledate,
    format = "%Y-%m-%d")
PeterPaul.chem.nutrients.gathered$sampledate <-
  as.Date(PeterPaul.chem.nutrients.gathered$sampledate,
    format = "%Y-%m-%d")
Litter$collectDate <-
  as.Date(Litter$collectDate, format = "%Y-%m-%d")

class(PeterPaul.chem.nutrients$sampledate)

## [1] "Date"
class(PeterPaul.chem.nutrients.gathered$sampledate)

## [1] "Date"
class(Litter$collectDate)

## [1] "Date"

```

Define your theme

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

```

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

theme_set(mytheme)

```

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 by phosphate, 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.

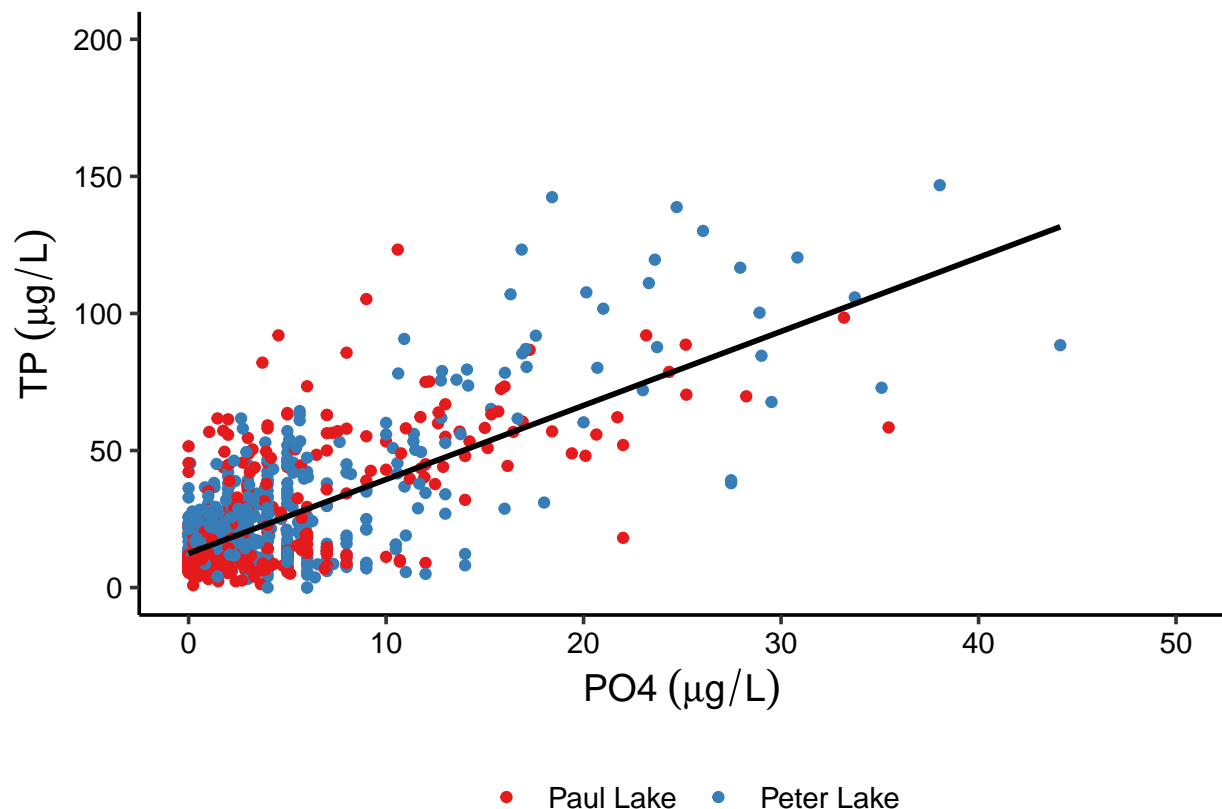
```
library(viridis)

## Loading required package: viridisLite

library(RColorBrewer)
library(colormap)

Pplot1 <- ggplot(PeterPaul.chem.nutrients) +
  geom_point(aes(x = po4, y = tp_ug, color = lakename)) +
  geom_smooth(aes(x = po4, y = tp_ug), color = "black",
              method = lm, se = FALSE) +
  labs(x = expression(PO4 ~ (mu*g / L)), y = expression(TP ~ (mu*g / L)),
       color = "") +
  scale_color_brewer(palette = "Set1") +
  xlim(0, 50) +
  ylim(0, 200)
print(Pplot1)

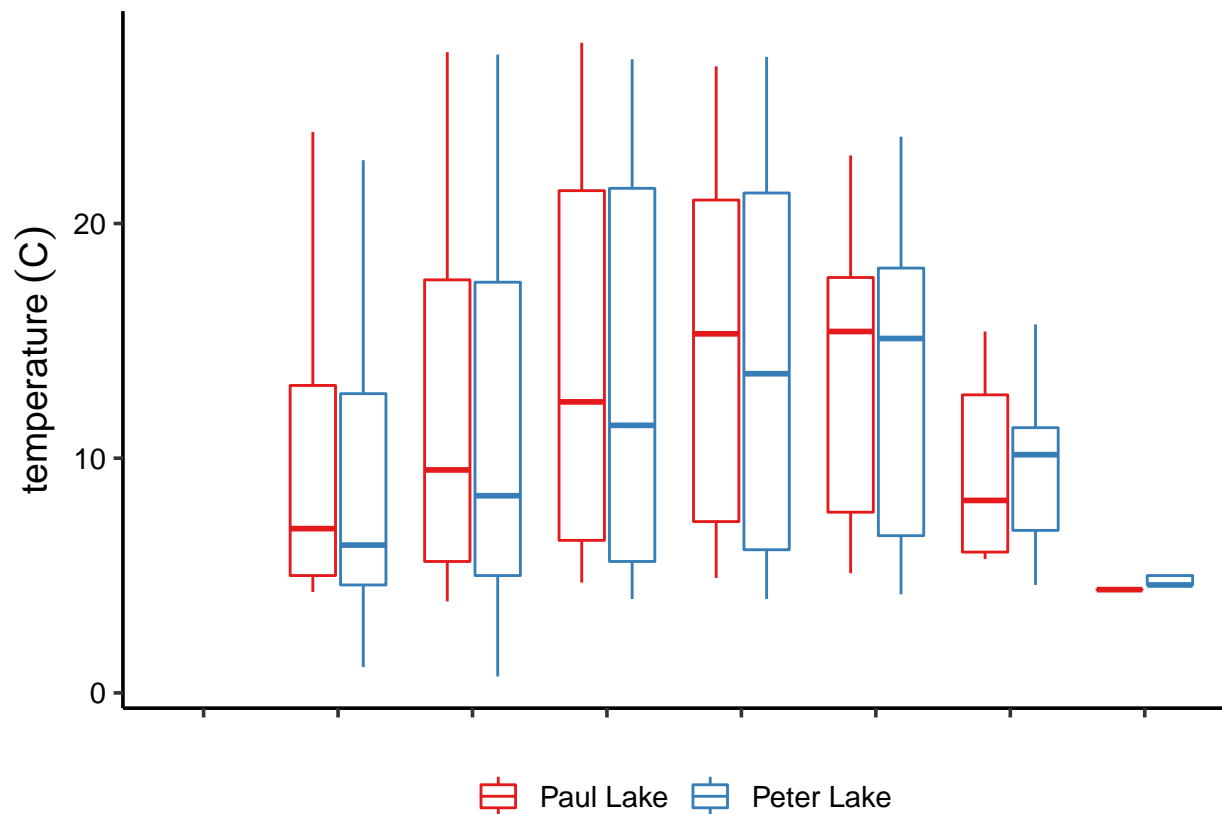
## Warning: Removed 21948 rows containing non-finite values (stat_smooth).
## Warning: Removed 21948 rows containing missing values (geom_point).
```



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.

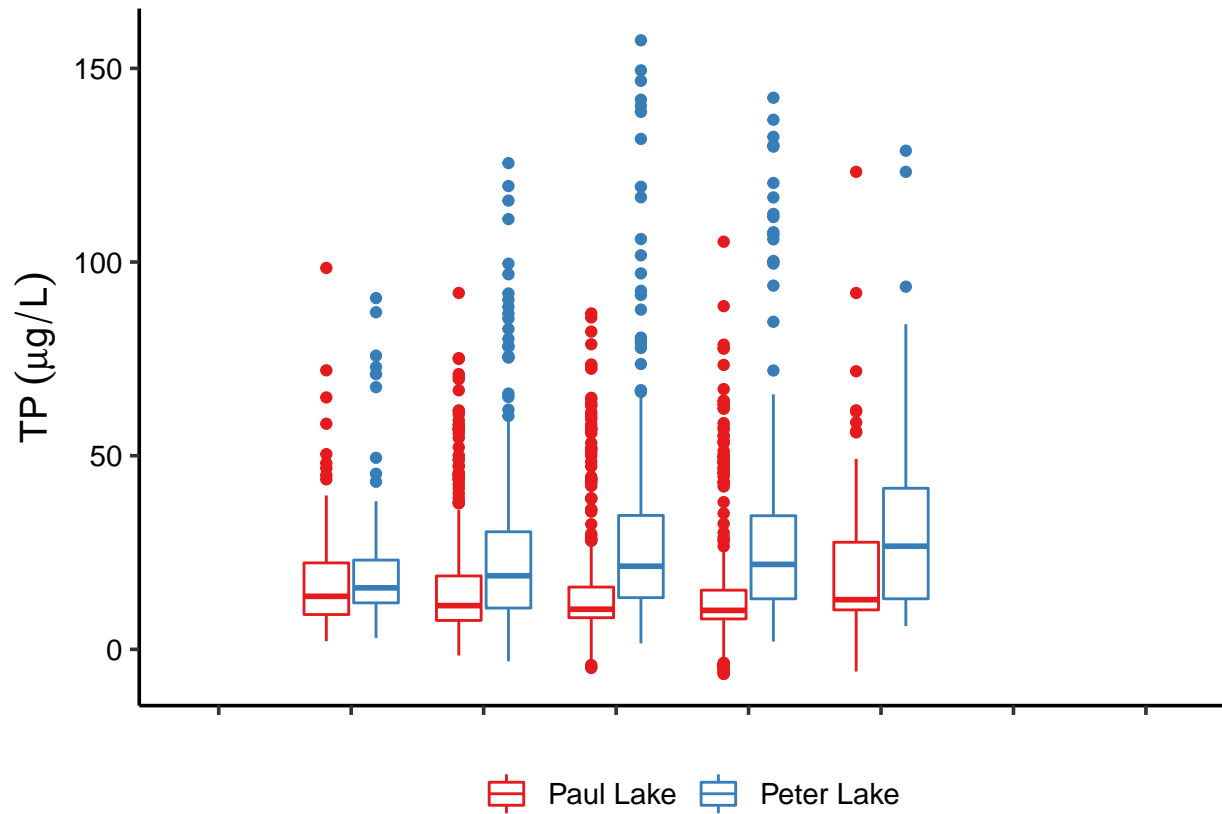
```
TempPlot <- ggplot(PeterPaul.chem.nutrients) +
  geom_boxplot(aes(x = as.factor(month), y = temperature_C, color = lakename)) +
  labs(x = expression(month), y = expression(temperature ~ (C)),
       color = "") +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank()) +
  scale_color_brewer(palette = "Set1")
print(TempPlot)
```

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



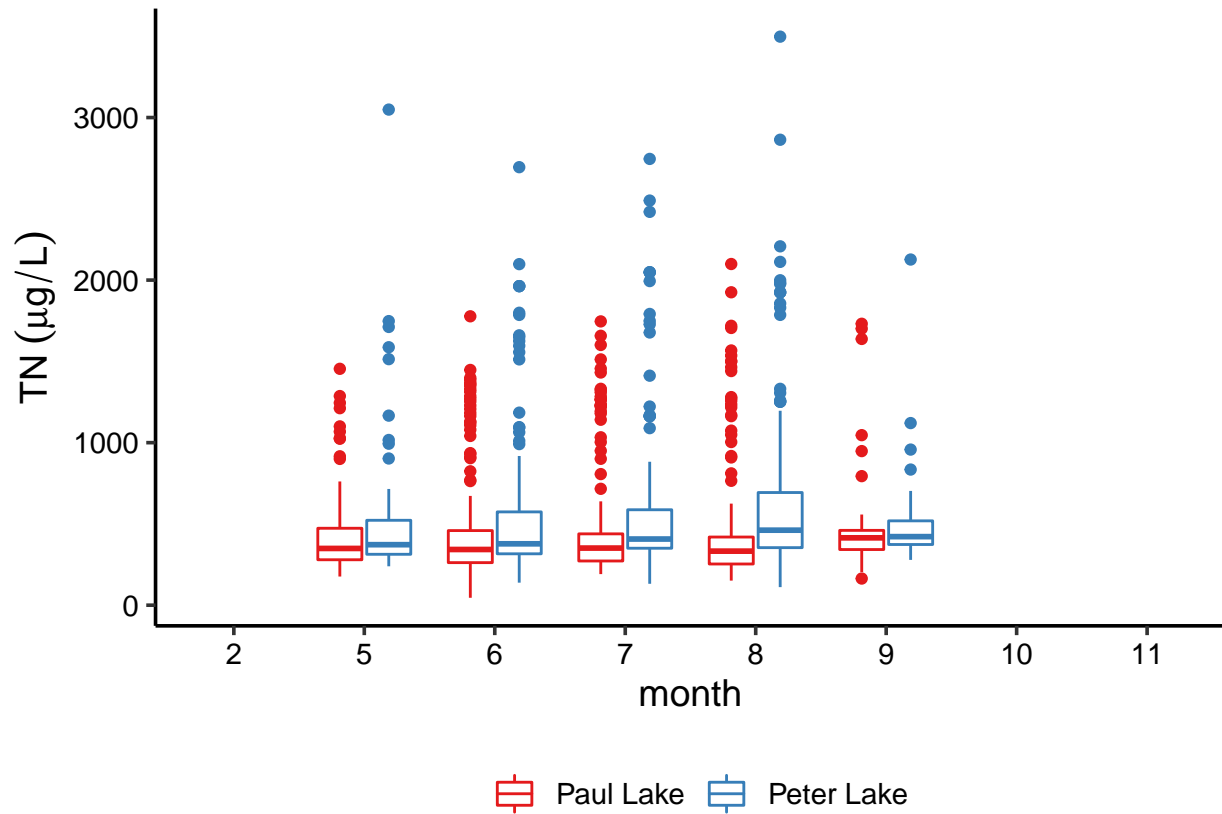
```
TPPlot <- ggplot(PeterPaul.chem.nutrients) +
  geom_boxplot(aes(x = as.factor(month), y = tp_ug, color = lakename)) +
  labs(x = expression(month), y = expression(TP ~ (mu*g / L)),
       color = "") +
  scale_color_brewer(palette = "Set1") +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())
print(TPPlot)
```

Warning: Removed 20729 rows containing non-finite values (stat_boxplot).



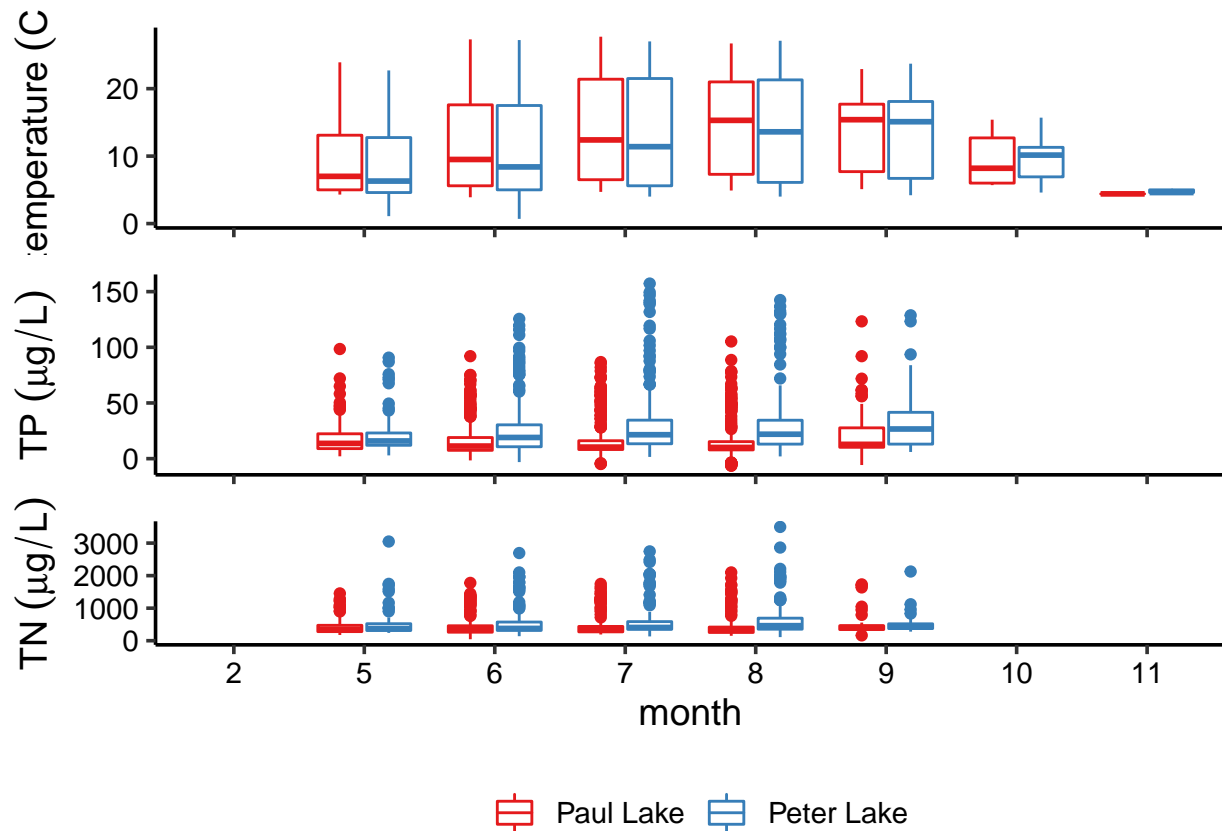
```
TNPlot <- ggplot(PeterPaul.chem.nutrients) +
  geom_boxplot(aes(x = as.factor(month), y = tn_ug, color = lakename)) +
  labs(x = expression(month), y = expression(TN ~ (mu*g / L)),
    color = "") +
  scale_color_brewer(palette = "Set1")
print(TNPlot)
```

```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```



```
library(cowplot)
plot_grid((TempPlot + theme(legend.position = "none")), (TPPlot + theme(legend.position = "none")), (TNPlot + theme(legend.position = "none")))

## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```



```
print
```

```
## function (x, ...)
## UseMethod("print")
## <bytecode: 0x7fea9e7bec48>
## <environment: namespace:base>
```

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

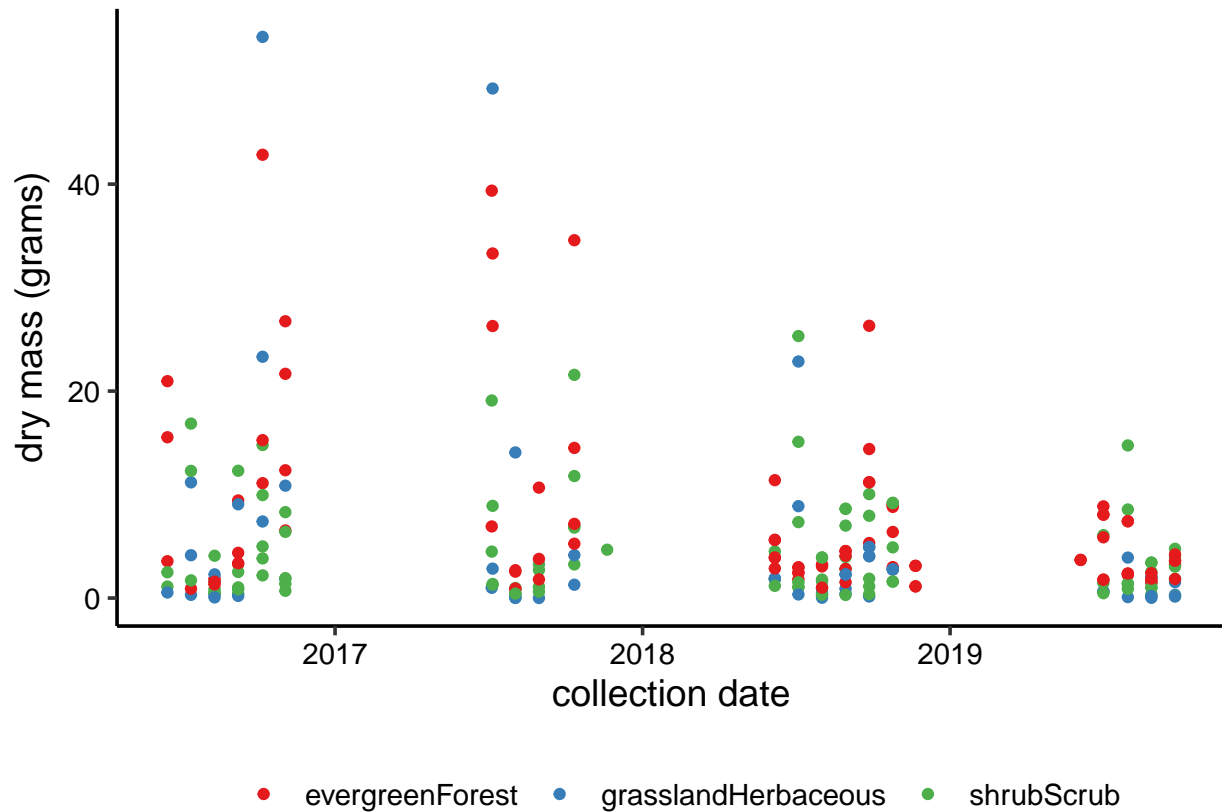
Answer: Temperature increases during the summer and early fall months. There is not a marked difference in temperature between the two lakes. Total phosphorous does not appear to have much seasonal variability and there are many outliers (particularly higher value outliers) for observations in both lakes. Peter Lake consistently has higher median values of total phosphorous across seasons. Total nitrogen, like total phosphorous, does not appear to have much seasonal variability and there are many upper-value outliers. Peter Lake, again, has higher median values of total nitrogen across seasons.

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)

```
Needles <- filter(Litter, functionalGroup %in%
  c("Needles"))

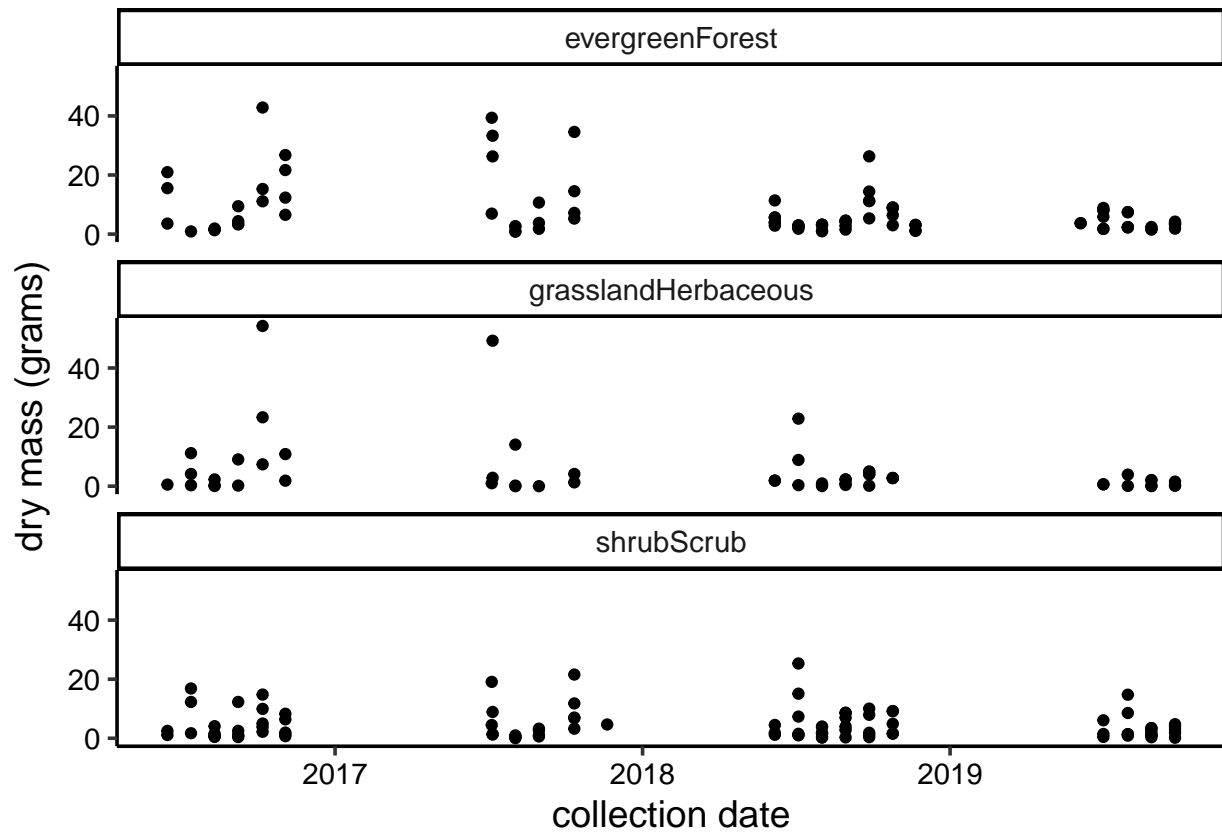
NeedlesPlot <- ggplot(Needles) +
  geom_point(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  labs(x = expression("collection date"), y = expression("dry mass (grams)"),
    color = "") +
```

```
scale_color_brewer(palette = "Set1")
print(NeedlesPlot)
```



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

```
NeedlesPlot2 <- ggplot(Needles) +
  geom_point(aes(x = collectDate, y = dryMass)) +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  labs(x = expression("collection date"), y = expression("dry mass (grams)"),
       color = "") +
  scale_color_brewer(palette = "Set1")
print(NeedlesPlot2)
```

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

Answer: I think plot 7 is more effective because it allows you to look at changes in dry mass over time for a given NLCD class and it also allows you to look vertically to compare dry mass values across NLCD classes, at a given collection date. While plot 6 differentiates NLCD classes using a color aesthetic, there are too many overlapping data points for the color aesthetic to aid the viewer in looking at differences in dry mass between NLCD classes.