Assignment 5: Data Visualization

Elise Harrigan

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.

```
#1
setwd("~/Desktop/Duke/Spring 2021/EnvDataAnalytics_872/Environmental_Data_Analytics_2021")
library(tidyverse)
```

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

## v ggplot2 3.3.1     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

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

```
library(cowplot)

PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", stringsAsFacto

PeterPaul.chem.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv", stringsAsFactors
Niwot.Litter <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = TRUE)

#2
Niwot.Litter$collectDate <- as.Date(Niwot.Litter$collectDate, format = "%Y-%m-%d")

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, f
```

Define your theme

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

```
mytheme <- theme_light(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top", legend.justification = "right")
theme_set(mytheme)
```

Create graphs

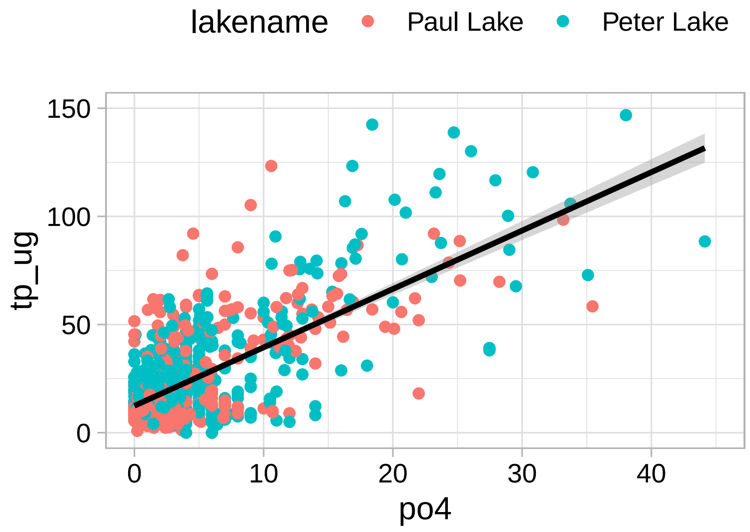
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus (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.

```
plot1_po4_tpug <-
  ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method=lm, color = "black") +
  xlim(0, 45) +
  ylim(0, 150)

print(plot1_po4_tpug)
```

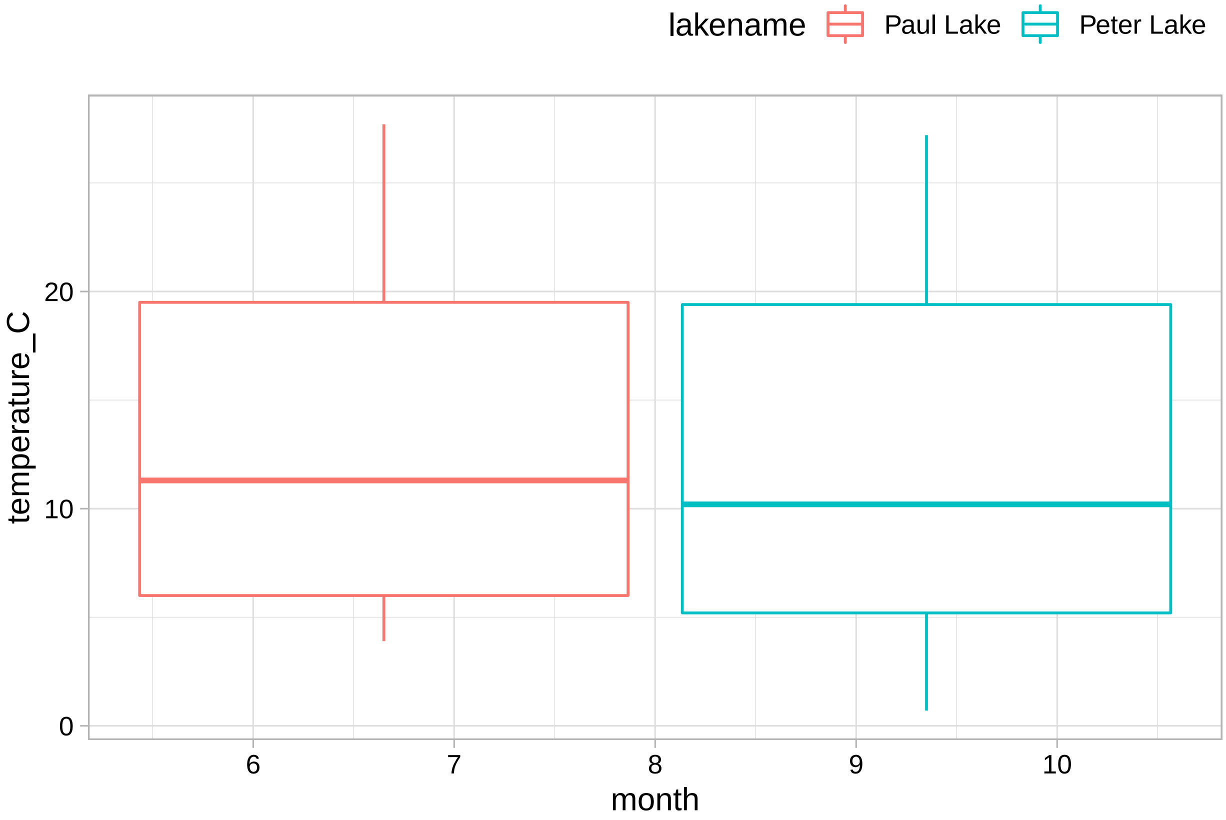
```
## `geom_smooth()` using formula 'y ~ x'
```



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.

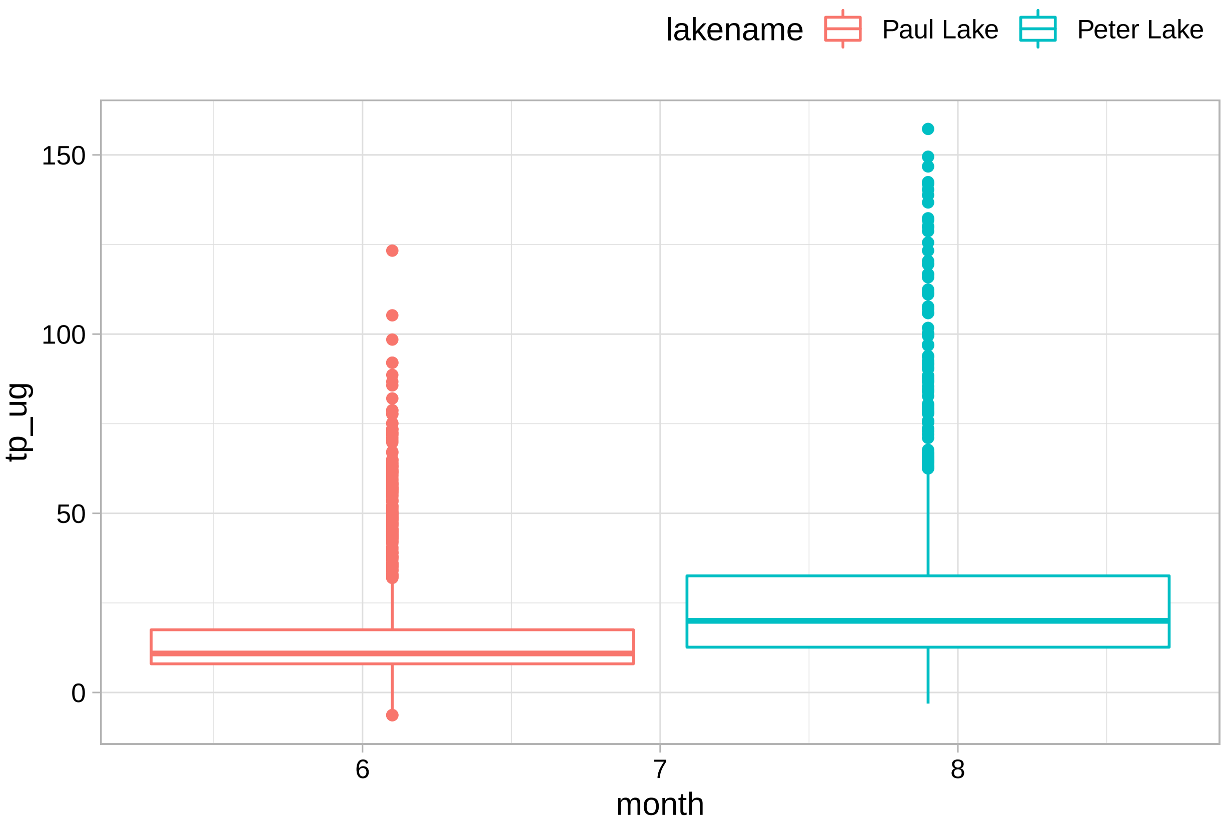
```
#3 separate boxplots
plot2 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename))
print(plot2)
```

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



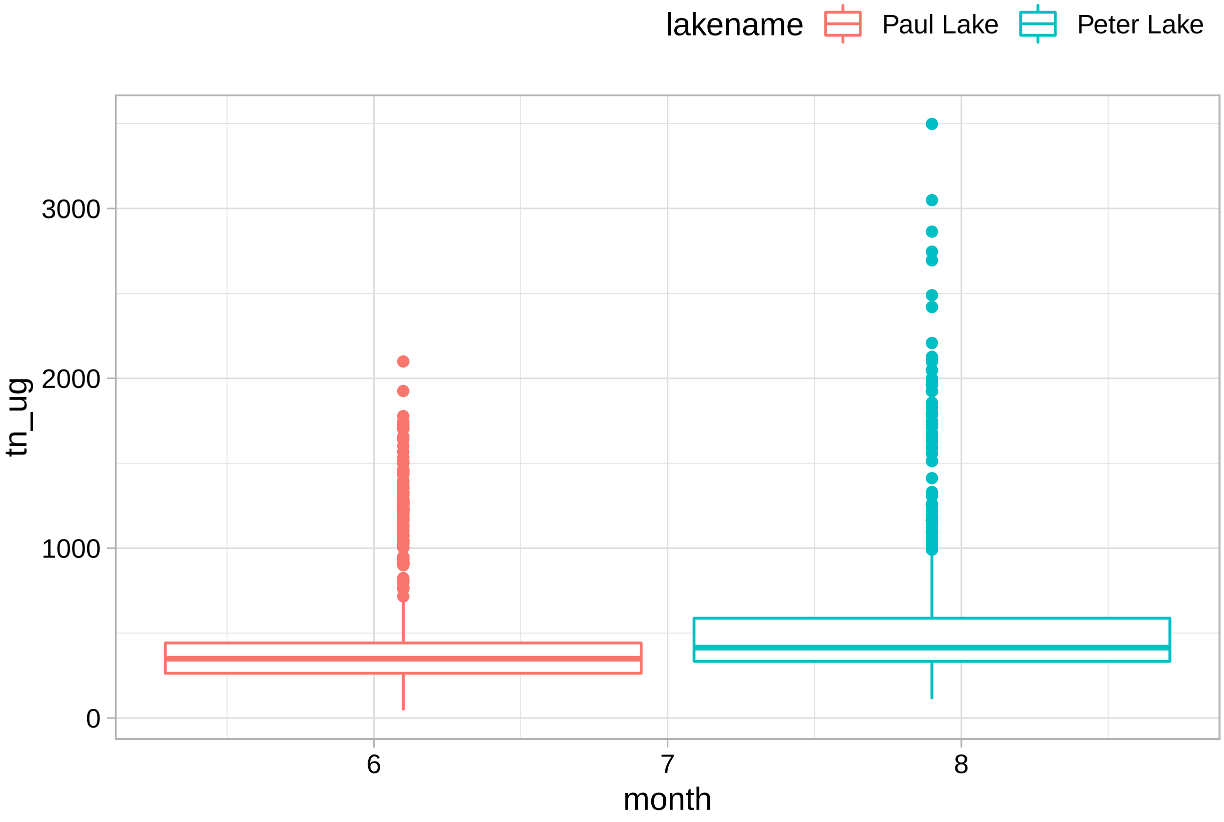
```
plot2.1 <-
ggplot(PeterPaul.chem.nutrients, aes(x=month, y= tp_ug)) +
  geom_boxplot(aes(color = lakename))
print(plot2.1)
```

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



```
plot2.2 <-
  ggplot(PeterPaul.chem.nutrients, aes(x=month, y=tn_ug))+
  geom_boxplot(aes(color = lakename))
print(plot2.2)
```

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



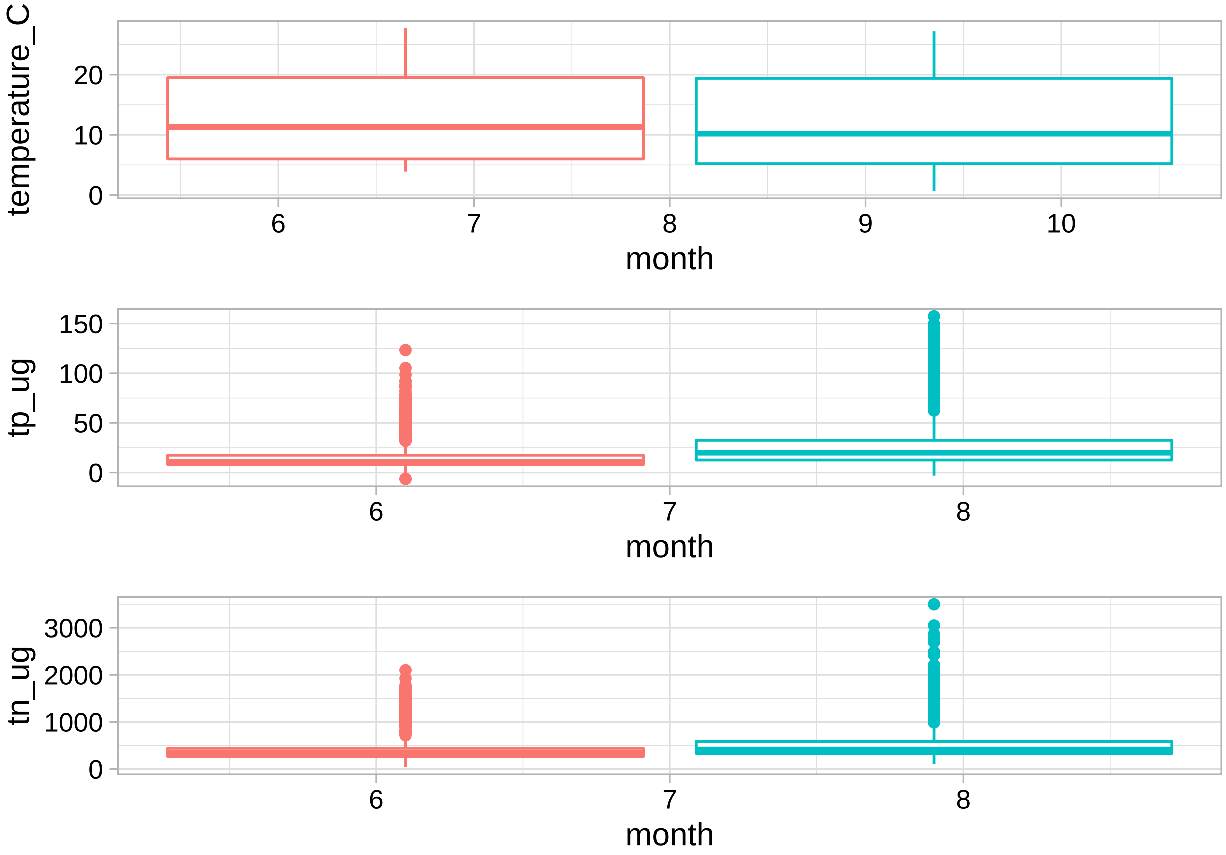
```
#cowplot to combine all three graphs
#create a stack of graphs without the legend - found wilkelab.org for help
pcol<- plot_grid(
  plot2 + theme(legend.position="none"),
  plot2.1 + theme(legend.position="none"),
  plot2.2 + theme(legend.position="none"),
align = 'vh',
hjust = -1,
ncol = 1
)
```

```
## 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).
```

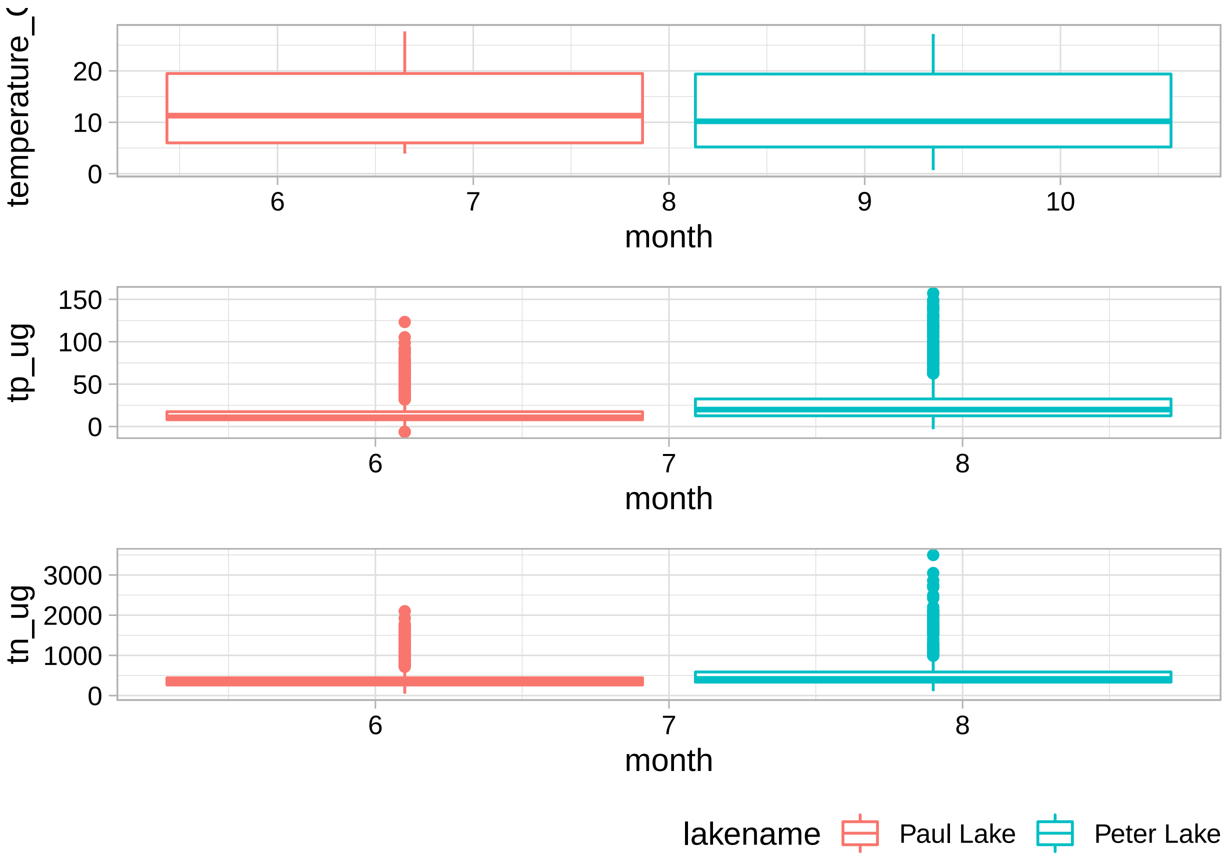
```
pcol
```



```
#extract the legend from one plot
legend <- get_legend(
  plot2 +
    guides (color = guide_legend(nrow=1)) +
    theme(legend.position = "bottom")
)
```

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

```
#add the legend to the col i made above. give it one-third of the width of one plot with rel_widths
plot_grid(pcol, legend, ncol = 1, rel_heights = c(1, .1))
```

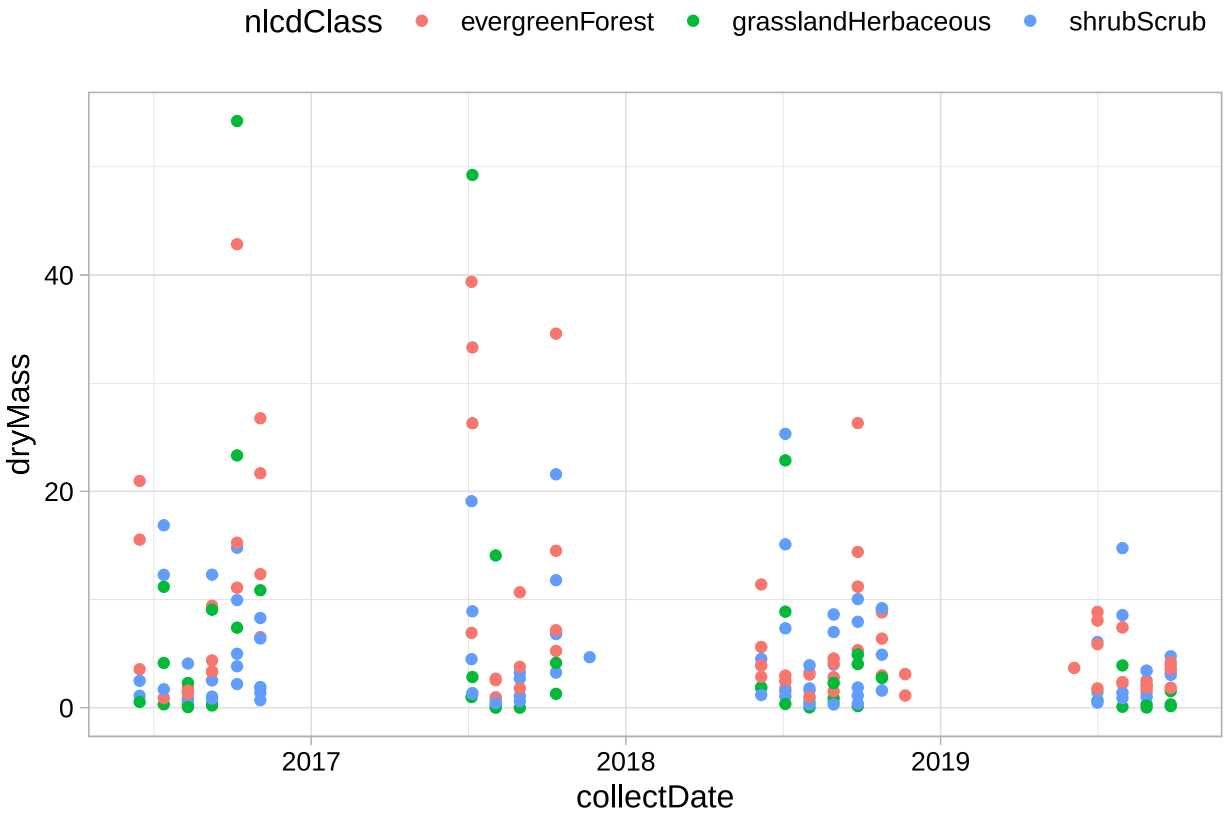


Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The temperature of Peter Lake get a little bit colder than Paul Lake which can be a little bit warmer. Later in the season, there is more TP and TN in Peter 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.

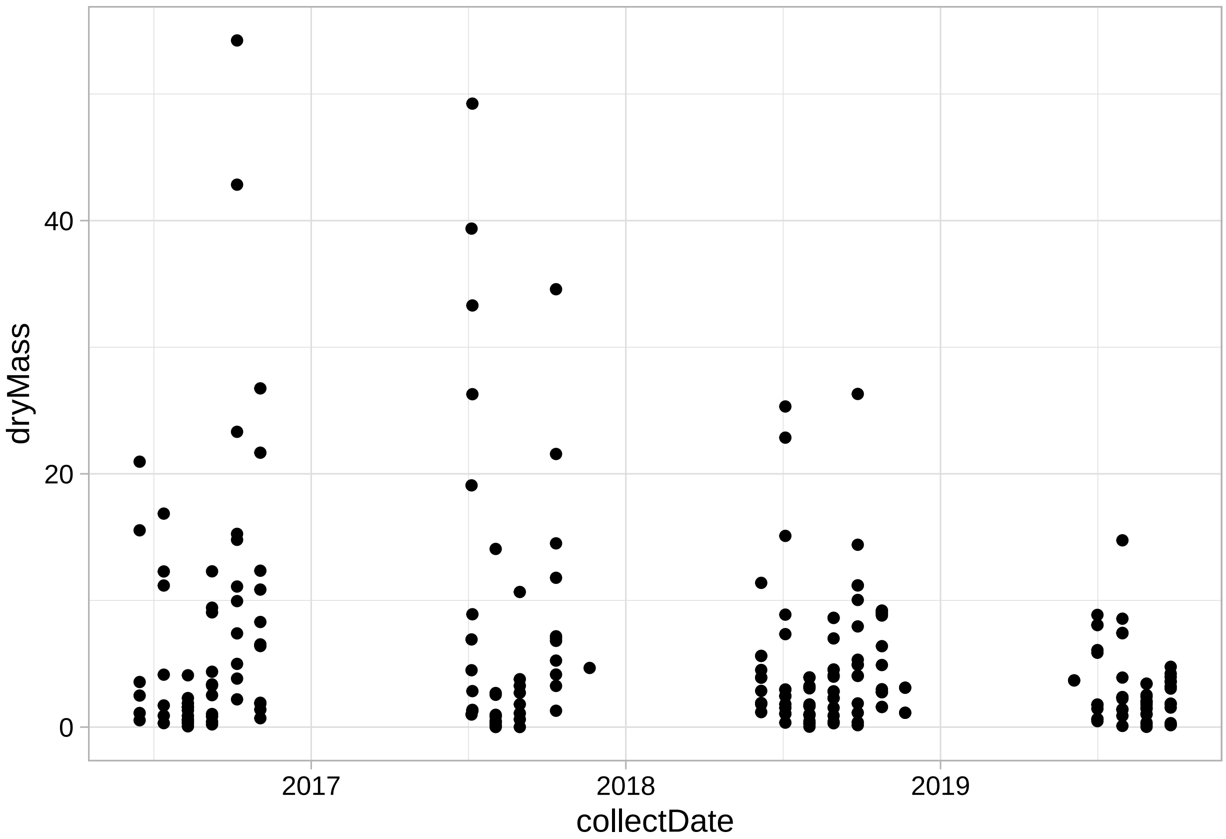
```
needles.p1 <-
  ggplot(subset(Niwot.Litter, functionalGroup == "Needles"),
         aes(x=collectDate, y=dryMass, color=nlcdClass)) +
  geom_point()
print(needles.p1)
```



```
needles.p2<-ggplot(subset(Niwot.Litter, functionalGroup == "Needles"),
         aes(x=collectDate, y=dryMass)) +
  geom_point()
  facet_wrap(vars(nlcdClass))
```

```
## <ggproto object: Class FacetWrap, Facet, gg>
##     compute_layout: function
##     draw_back: function
##     draw_front: function
##     draw_labels: function
##     draw_panels: function
##     finish_data: function
##     init_scales: function
##     map_data: function
##     params: list
##     setup_data: function
##     setup_params: function
##     shrink: TRUE
##     train_scales: function
##     vars: function
##     super:  <ggproto object: Class FacetWrap, Facet, gg>
```

```
print(needles.p2)
```



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

> Answer: Plot 6 (needles.p1) I think is more effective because you can see the different types of forest cover easily by color. the facet_wrap does not give you any additional information about the type of class.