Assignment 5: Data Visualization

Sam Tolbert

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.

```r
#1

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
```

```r
library(lubridate)
library(cowplot)
```

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

```r
library(here)
```

```
## here() starts at /home/guest/EDA Fall 2024
```

```r
Peter.Paul.ChemistryNutrients<-read.csv(here(                   'Data/Processed_KEY/NTL-LTER_Lake_Chemis

NEON.LitterMass<-read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors=T)


#2

class(Peter.Paul.ChemistryNutrients$sampledate) #oh no! its a factor
```

```
## [1] "factor"
```

```r
class(NEON.LitterMass$collectDate)
```

```
## [1] "factor"
```

```r
Peter.Paul.ChemistryNutrients$sampledate<-ymd(Peter.Paul.ChemistryNutrients$sampledate)
NEON.LitterMass$collectDate<-ymd(NEON.LitterMass$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

```r
#3

Sam_Theme<-theme(
  panel.background = element_rect (fill="beige"),
  axis.line.y = element_line (color ="lightblue"),
  axis.line.x= element_line (color= "lightblue"),
  panel.grid.major = element_line (color ="lightblue"),
  panel.grid.minor = element_line (color= "lightblue"),

  plot.title= element_text(size= 16, hjust=.5),
)
```

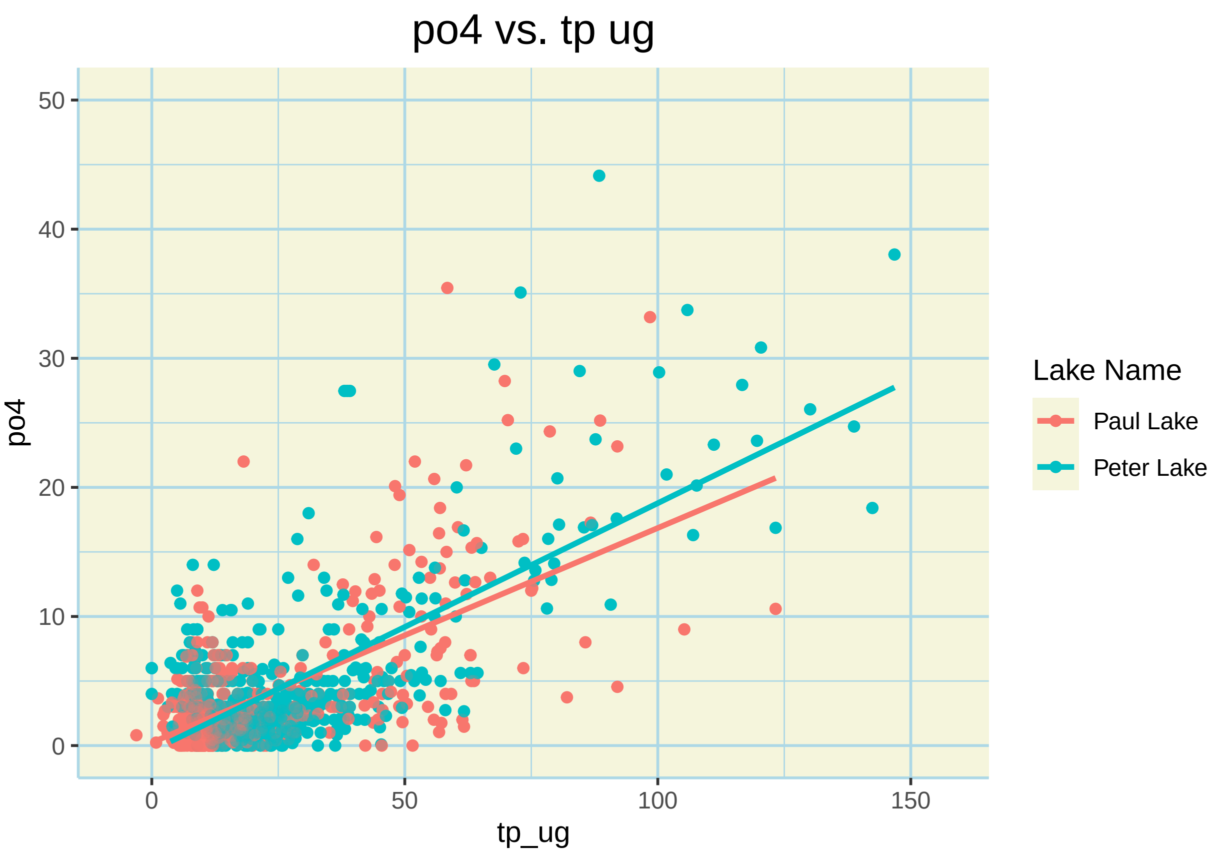
Create graphs

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

Peter.Paul.ChemistryNutrients %>%
  ggplot(aes(x=tp_ug,
             y=po4,
             color=lakename))+
  labs(title="po4 vs. tp ug", color="Lake Name")+
  geom_point() +
  ylim(y=0,50) +
  geom_point(alpha=.2)+
  geom_smooth(method="lm", se=0)+
  Sam_Theme
```

```
## ‘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()‘).
## 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.

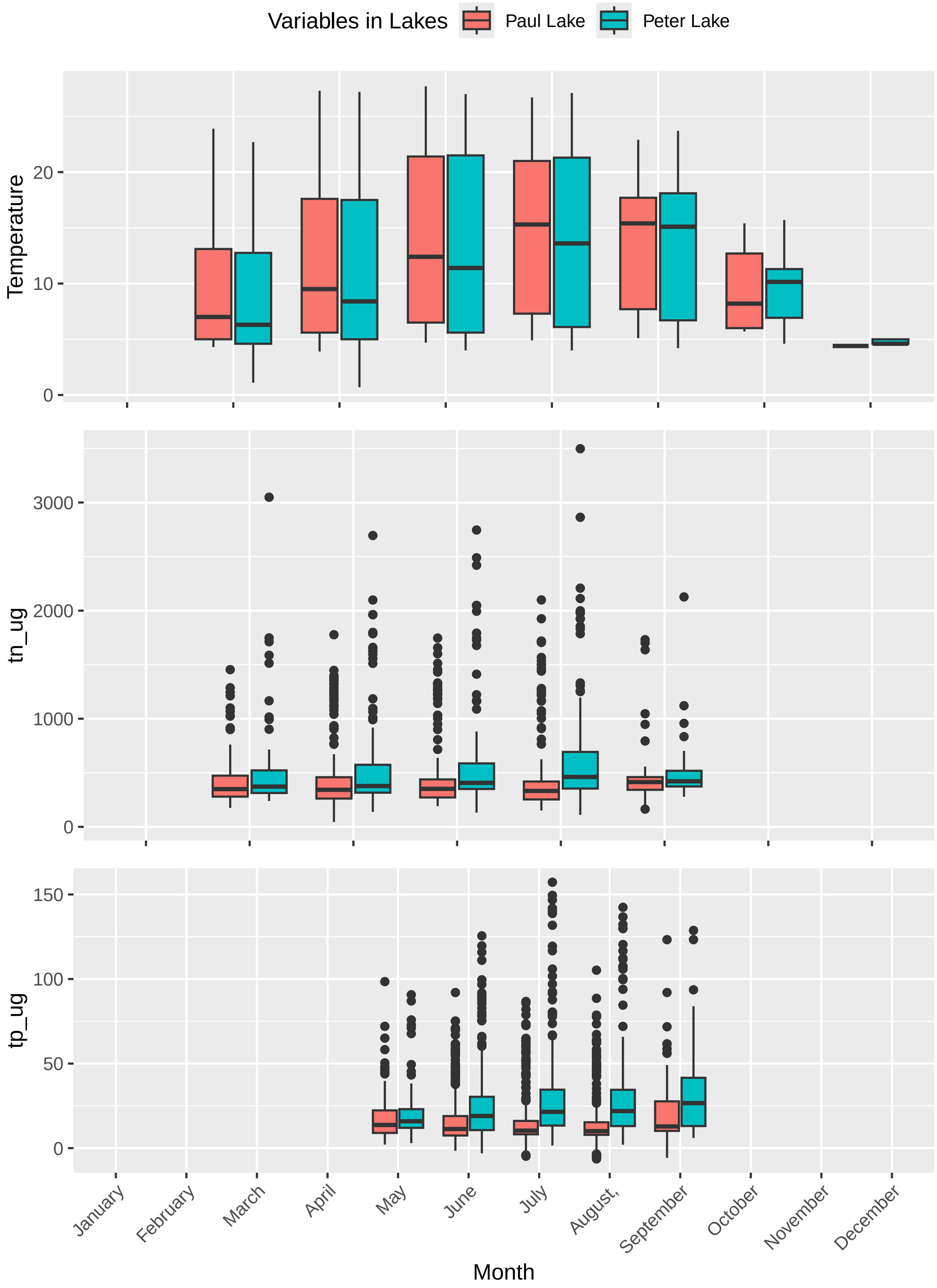
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a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

```
#5

Peter.Paul.ChemistryNutrients$month<-as.factor(Peter.Paul.ChemistryNutrients$month)


PPTemp<-Peter.Paul.ChemistryNutrients %>%
  ggplot(aes(x=month,
             y=temperature_C,
             fill= lakename))+
  geom_boxplot()+
  labs(fill= "Variables in Lakes", y= "Temperature")+
  theme(
        legend.position= "top",
        axis.title.x = element_blank(),
        axis.text.x= element_blank())
```

```
PPTpug<-Peter.Paul.ChemistryNutrients %>%
ggplot(aes(x=month,
             y=tp_ug,
             fill=lakename))+
  geom_boxplot()+
  labs(x="Month")+
    theme(axis.text.x=element_text(angle = 45, hjust = 1),
          legend.position = "none")+
  scale_x_discrete(
  limits= c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"),
  labels= c("January", "February", "March", "April", "May", "June", "July", "August,", "September", "Oc

PPTnug<-Peter.Paul.ChemistryNutrients %>%
  ggplot(aes(x=month,
             y=tn_ug,
             fill=lakename))+
geom_boxplot()+
    theme(axis.text.x=element_blank(),
          legend.position="none",
          axis.title.x = element_blank())

CombinedPPDataGrid<-plot_grid(PPTemp,
PPTnug,
PPTpug,
  ncol=1, align="y") #vertical alignment isn't quite as clear in regards to
```

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

## 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()`).
```

```
#relationship between temp and chemical but ncol=3
#horizontal alignment on my tiny screen is impossible to see so bear with me pls

print(CombinedPPDataGrid)
```

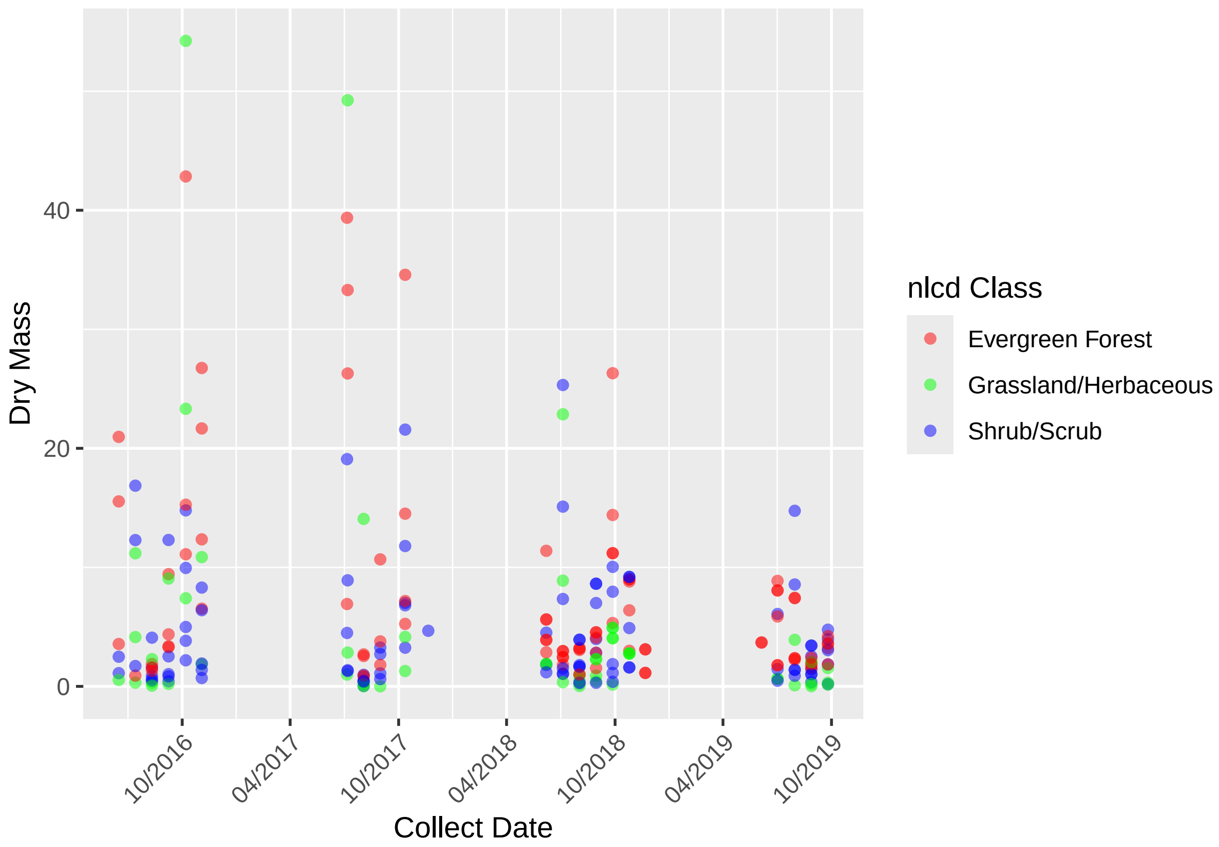

Variables in Lakes
Paul Lake
Peter Lake
Temperature
0
10
20
tn_ug
0
1000
2000
3000
tp_ug
0
50
100
150
January
February
March
April
May
June
July
August,
September
October
November
December
Month

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

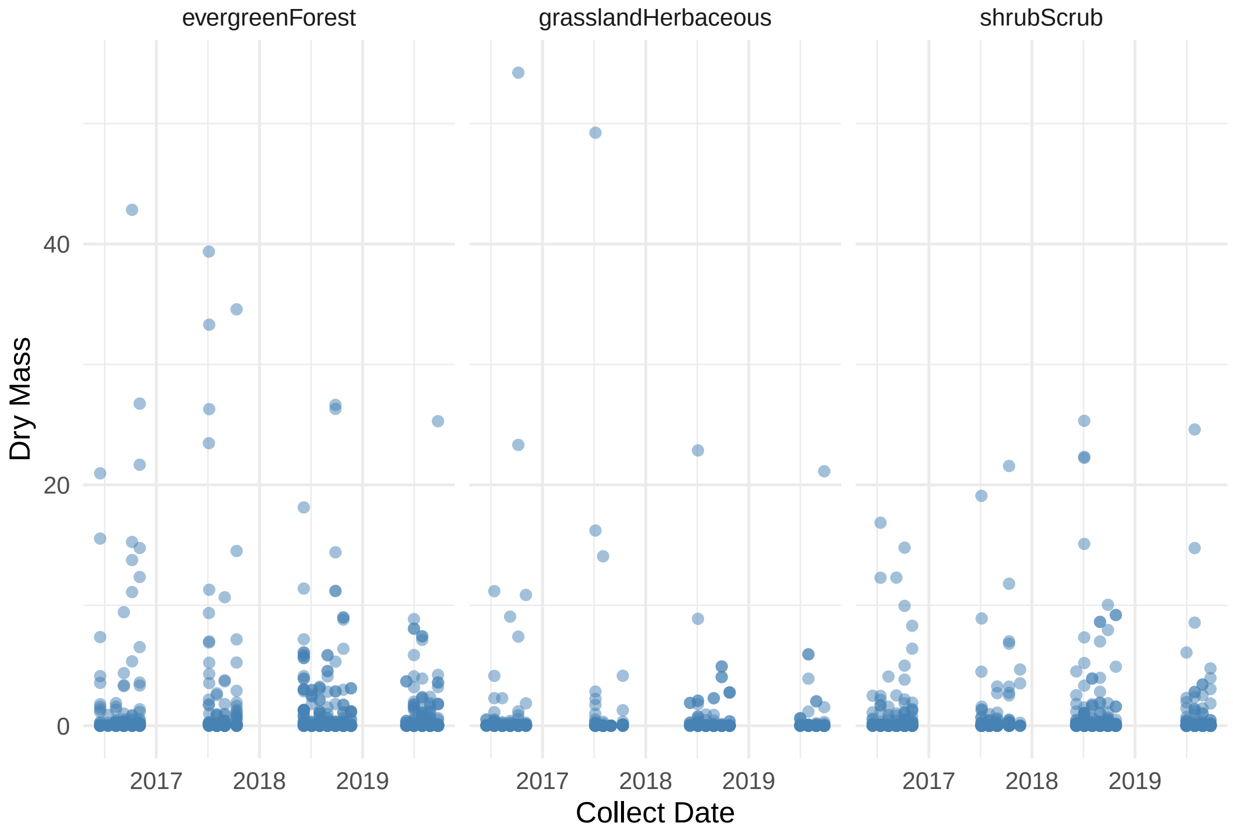
> Answer: The chemicals tend to increase as the year goes on.

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
NEON.LitterMass_OnlyNeedles<-NEON.LitterMass %>%
                                filter(functionalGroup=="Needles")

NEON.Scatter<-NEON.LitterMass_OnlyNeedles %>%
  ggplot(aes(x=collectDate,
             y= dryMass,
             color=nlcdClass))+
  geom_point(alpha=.5)+
  labs(color ="nlcd Class",
       y="Dry Mass",
       x= "Collect Date"
       )+
  scale_x_date(date_breaks = "6 months", date_labels = "%m/%Y")+
  scale_color_manual(
    values= c("evergreenForest"= "red", "grasslandHerbaceous"="green", "shrubScrub"= "blue" ),
    labels= c("Evergreen Forest", "Grassland/Herbaceous", "Shrub/Scrub")
)+
    theme(
      axis.text.x = element_text(angle=45, hjust=1))

print(NEON.Scatter)
```



```
#7

NEON.Facet<-NEON.LitterMass %>%
  ggplot(aes(x=collectDate,
             y=dryMass))+
  geom_point(color="steelblue", alpha=.5 ) +
  facet_wrap(~ nlcdClass)+
  theme_minimal()+
  labs (x= "Collect Date", y="Dry Mass")

print(NEON.Facet)
```



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

Answer:It's an interesting question. I also messed around with alpha to see if it made a differene. Ultimately I think that the faceted graph provides a little more clarity, as the jumble of colors in the joint plot doesn't provide as much clarity where the Mass quantities are concentrated towards the mean of each nlcd class. Any strenght in comparison in the outliers in the joint plot is captured by the faceted chart, and the faceted chart also provides a little more clarity of where each nlcd class stands towards it mean, as there's a little less jumble (again, opacity helps with that too).