

Assignment 5: Data Visualization

Masha Edmondson

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/mashaedmondson/Desktop/Environmental_Data_Analytics_2020"

library(tidyverse)

## — Attaching packages — tidyverse
1.3.0 —

## ✓ ggplot2 3.2.1    ✓ purrr  0.3.3
## ✓ tibble  2.1.3    ✓ dplyr  0.8.3
## ✓ tidyr   1.0.0     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0
```

```
## — Conflicts
```

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

```
library(ggribes)  
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.nutrients.gathered <-  
  read_csv("Data/Processed/NTL-  
  LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
```

```
## Parsed with column specification:  
## cols(  
##   lakename = col_character(),  
##   year4 = col_double(),  
##   daynum = col_double(),  
##   month = col_double(),  
##   sampleddate = col_date(format = ""),  
##   depth = col_double(),  
##   nutrient = col_character(),  
##   concentration = col_double()  
## )
```

```
NEON_NIWO_Litter_Processed <-  
read_csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
```

```
## Parsed with column specification:  
## cols(  
##   plotID = col_character(),  
##   trapID = col_character(),  
##   collectDate = col_date(format = ""),  
##   functionalGroup = col_character(),  
##   dryMass = col_double(),  
##   qaDryMass = col_character(),  
##   subplotID = col_double(),
```

```
## decimalLatitude = col_double(),
## decimalLongitude = col_double(),
## elevation = col_double(),
## nlcdClass = col_character(),
## plotType = col_character(),
## geodeticDatum = col_character()
## )

#2
PeterPaul.chem.nutrients$sampldate <-
as.Date(PeterPaul.chem.nutrients$sampldate, format = "%Y-%m-%d")
PeterPaul.nutrients.gathered$sampldate <-
as.Date(PeterPaul.nutrients.gathered$sampldate, format = "%Y-%m-%d")

class(PeterPaul.chem.nutrients$sampldate)

## [1] "Date"

class(PeterPaul.nutrients.gathered$sampldate)

## [1] "Date"

class(NEON_NIWO_Litter_Processed$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 = "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 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.

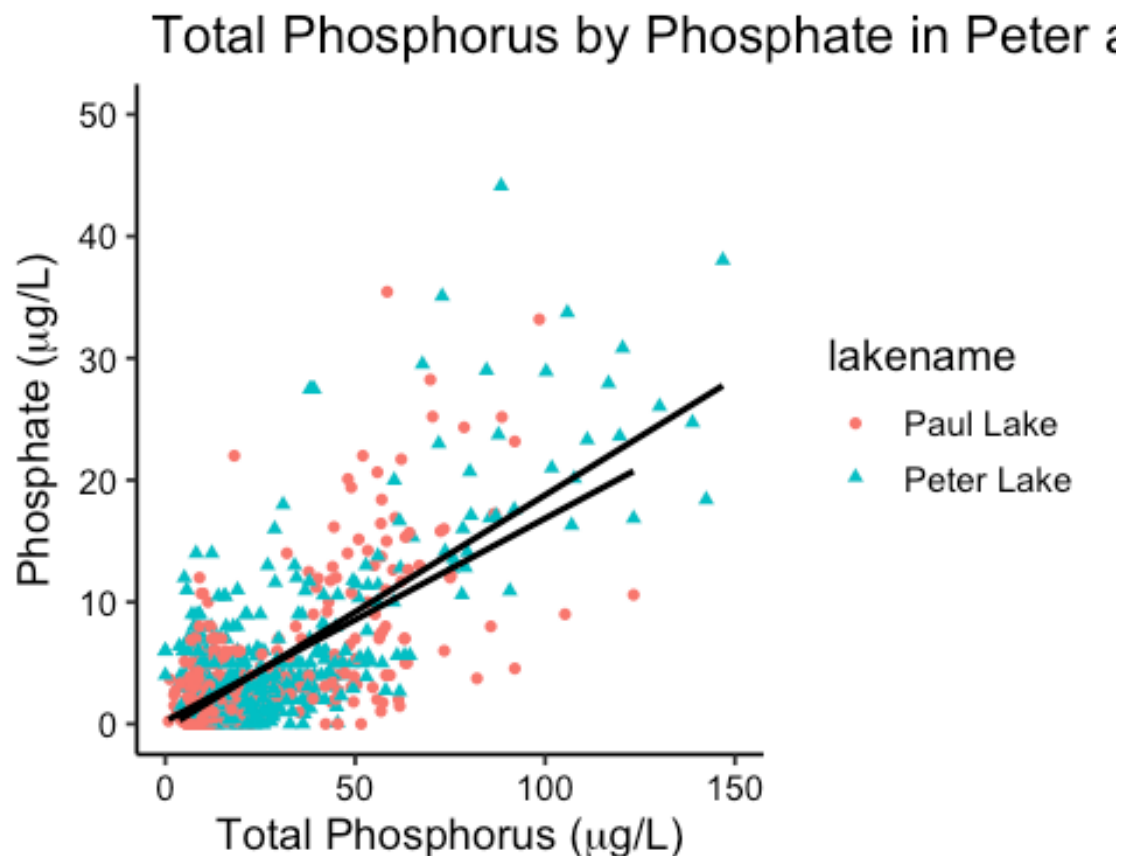
```
TP_PO4_plot <- ggplot(PeterPaul.chem.nutrients, aes(x= tp_ug, y = po4, shape=
lakename, color =lakename)) +
  geom_point(aes(x= tp_ug, y = po4, shape= lakename, color =lakename))+
  geom_smooth(aes(x= tp_ug, y = po4, shape= lakename, color =lakename),
method = "lm", se = FALSE, color="black") +
  labs(x = expression(paste("Total Phosphorus (", mu, "g/L)")), y =
expression(paste("Phosphate (", mu, "g/L)")), color= "lakename") +
```

```
xlim(0, 150)+
ylim(0, 50)+
ggtitle(" Total Phosphorus by Phosphate in Peter and Paul Lake")

## Warning: Ignoring unknown aesthetics: shape

print(TP_PO4_plot)

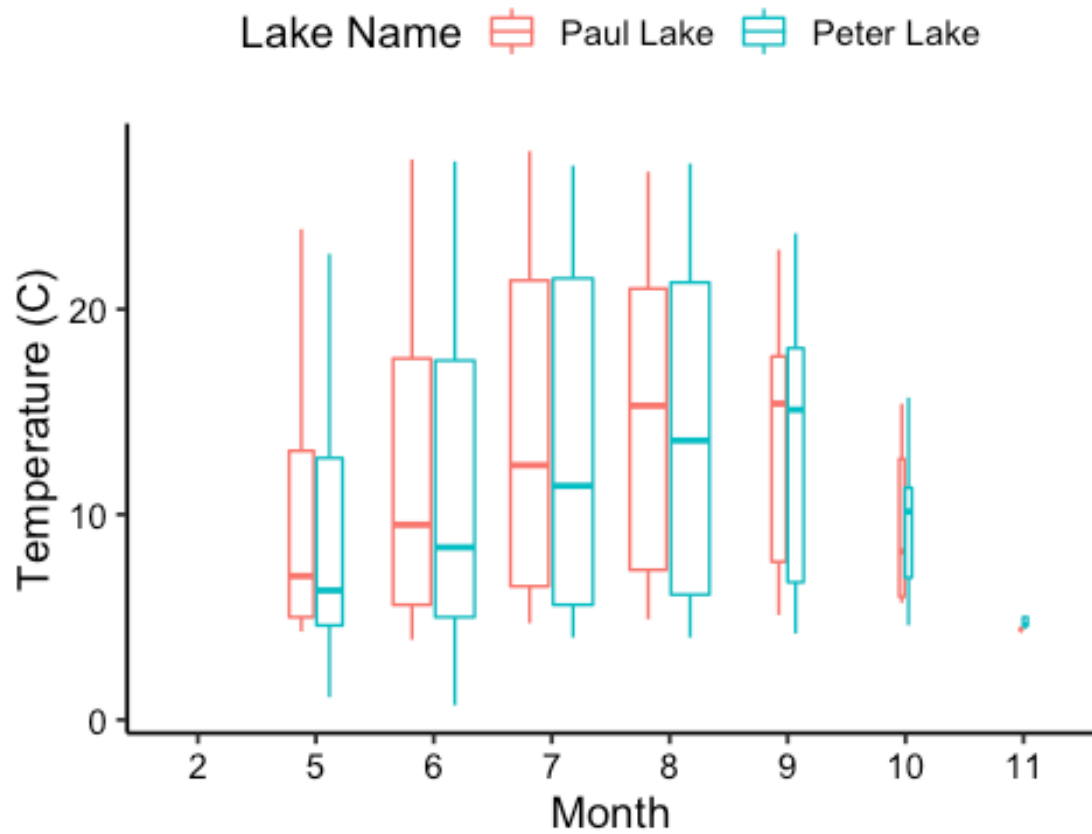
## Warning: Removed 21948 rows containing non-finite values (stat_smooth).
## Warning: Removed 21948 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_smooth).
```



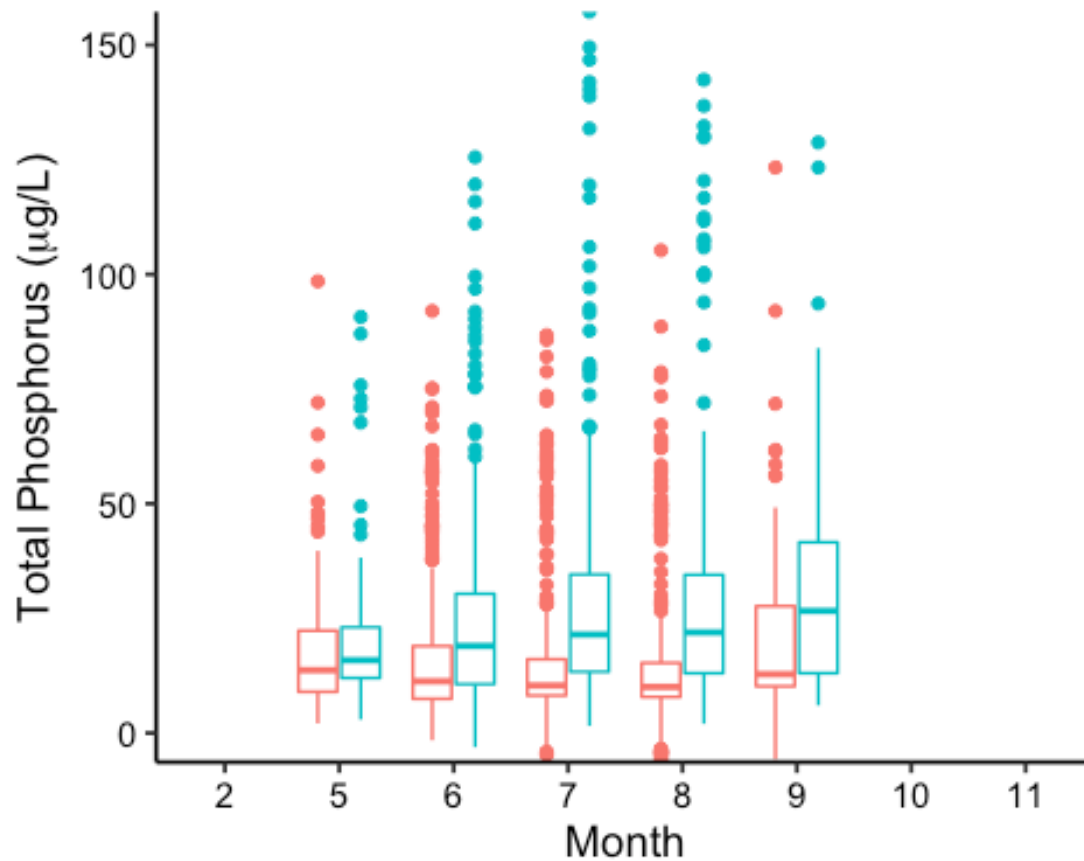
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
PeterPaul.temp.plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x= as.factor(month), y= temperature_C,
color = lakename))+
  geom_boxplot(varwidth = TRUE)+
  labs(x = "Month", y = "Temperature (C)", color = "Lake Name")+
```

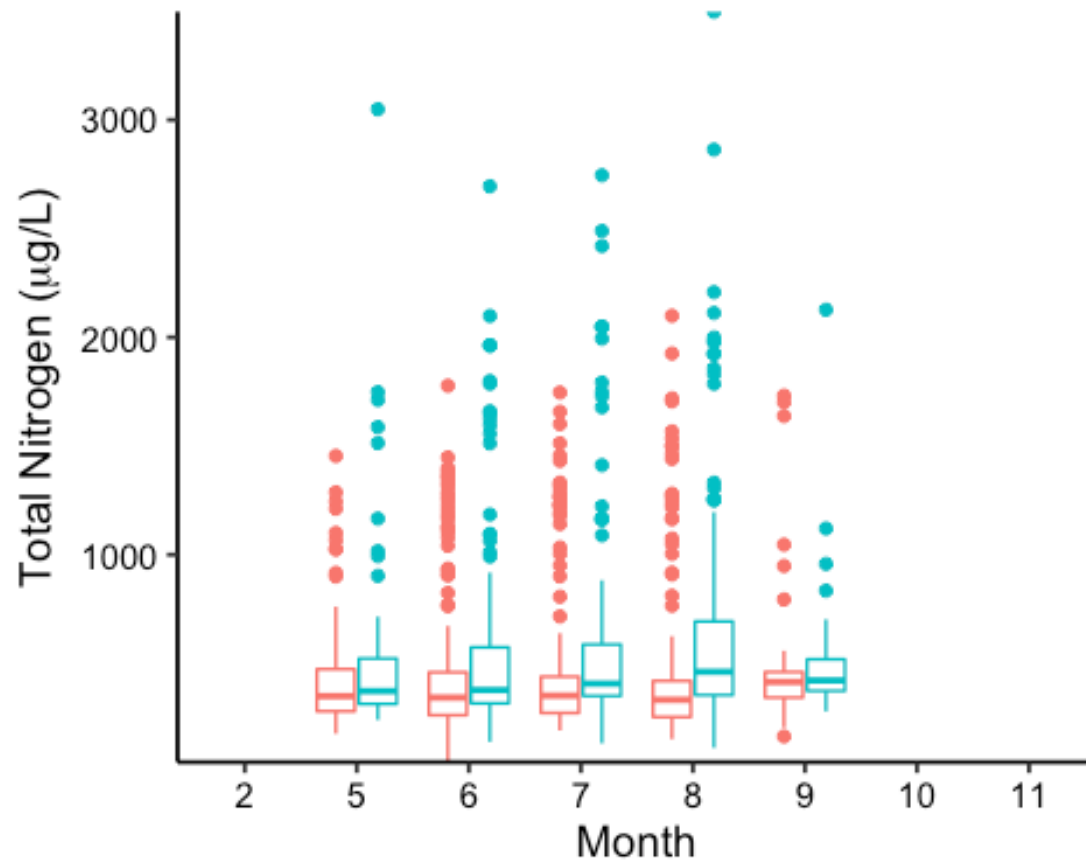
```
theme(legend.position = "top")
print(PeterPaul.temp.plot)
```



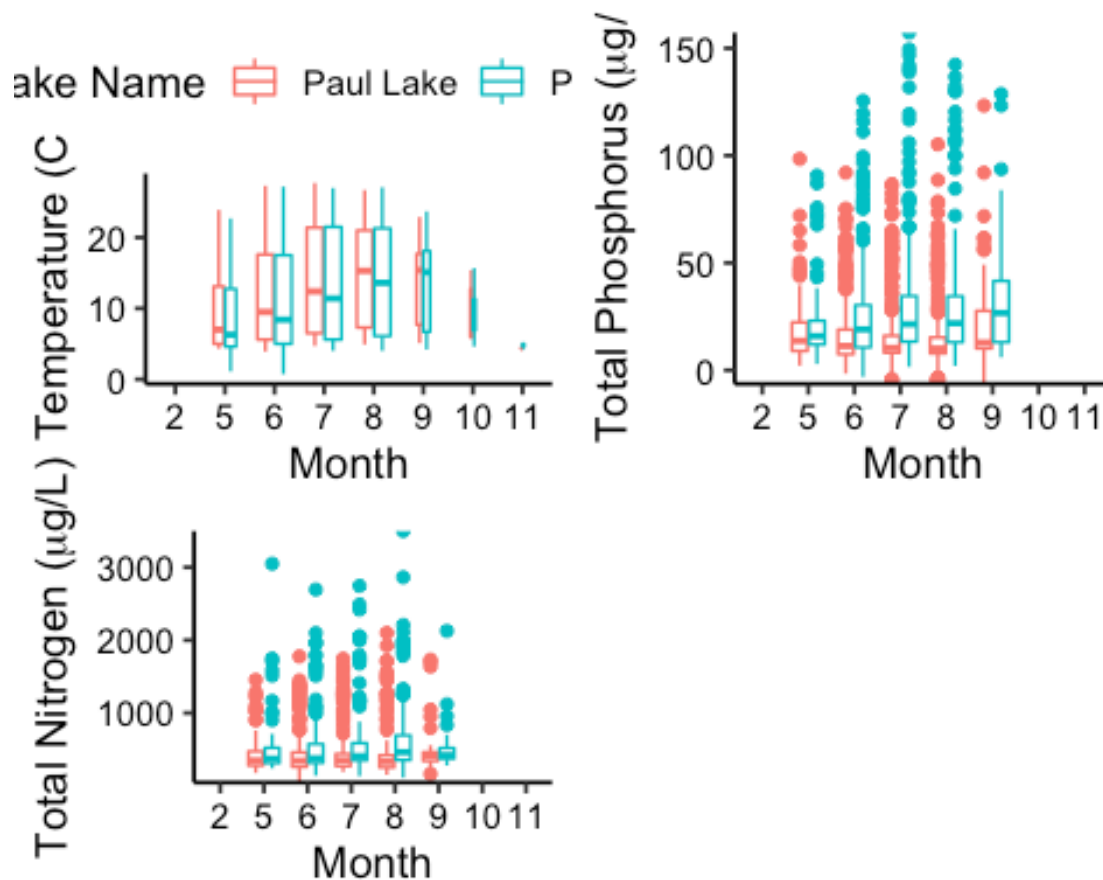
```
PeterPaul.tp.plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x= as.factor(month), y = tp_ug, color
= lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = expression(paste("Total Phosphorus (", mu, "g/L)")))
+
  scale_y_continuous(expand = c(0,0))+
  theme(legend.position = "none")
print(PeterPaul.tp.plot)
```



```
PeterPaul.tn.plot <-ggplot(PeterPaul.chem.nutrients, aes(x= as.factor(month),
y = tn_ug, color = lakenam)) +
  geom_boxplot() +
  labs(x = "Month", y = expression(paste("Total Nitrogen (", mu, "g/L)"))) +
  scale_y_continuous(expand = c(0,0))+
  theme(legend.position = "none")
print(PeterPaul.tn.plot)
```



```
plot_grid(PeterPaul.temp.plot, PeterPaul.tp.plot, PeterPaul.tn.plot, nrow=
3, axis = "b", align = "v", rel_heights = c(1.25, 1,1))
```



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

Answer: Through this graphic, we can see that temperature increases during months 7 and 8, which is consistent with summer time, and both lakes follow the same seasonal temperature ranges. It also illustrates that there are more outliers with both the total nitrogen and total phosphorus data. The median is lower for Paul Lake in regards to total phosphorus, but for Peter Lake total phosphorus has a slightly higher median with a larger inter quartile range. Total nitrogen seems to be relatively constant across the months for both lakes with concentrations and medians. Finally, through these graphs you can see a slight correlation between total nitrogen and total phosphorus decrease in medians over the summer months.

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

#6

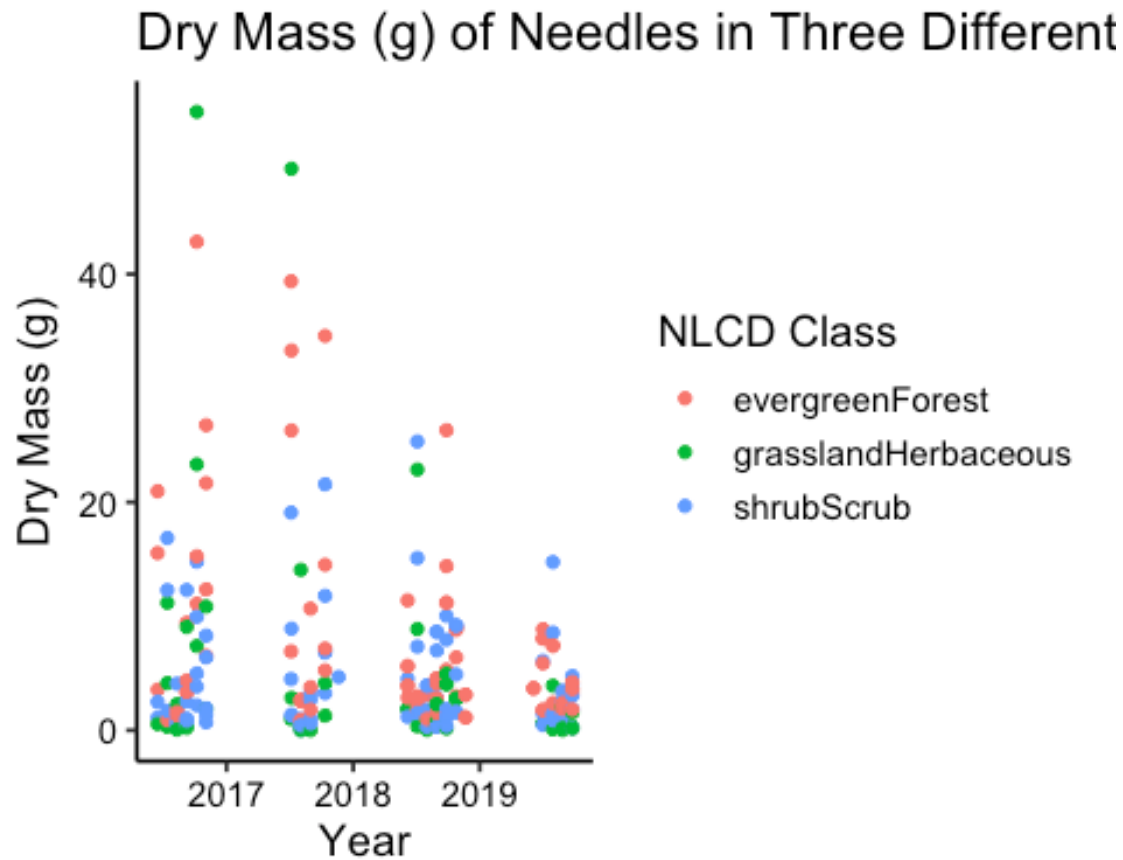
```
Needles.plot <- ggplot(subset(NEON_NIWO_Litter_Processed, functionalGroup ==
"Needles"), aes( x = as.Date(collectDate), y = dryMass, color = nlcdClass))+
  geom_point()+
  labs(x= "Year", y = expression("Dry Mass (g)"), color= "NLCD Class")+
  #scale_color_viridis(option = "magma", direction = -1, end = 0.8)+
```



```

theme(legend.position = "right")+
ggtitle("Dry Mass (g) of Needles in Three Different Classifications of
Forests")
print(Needles.plot)

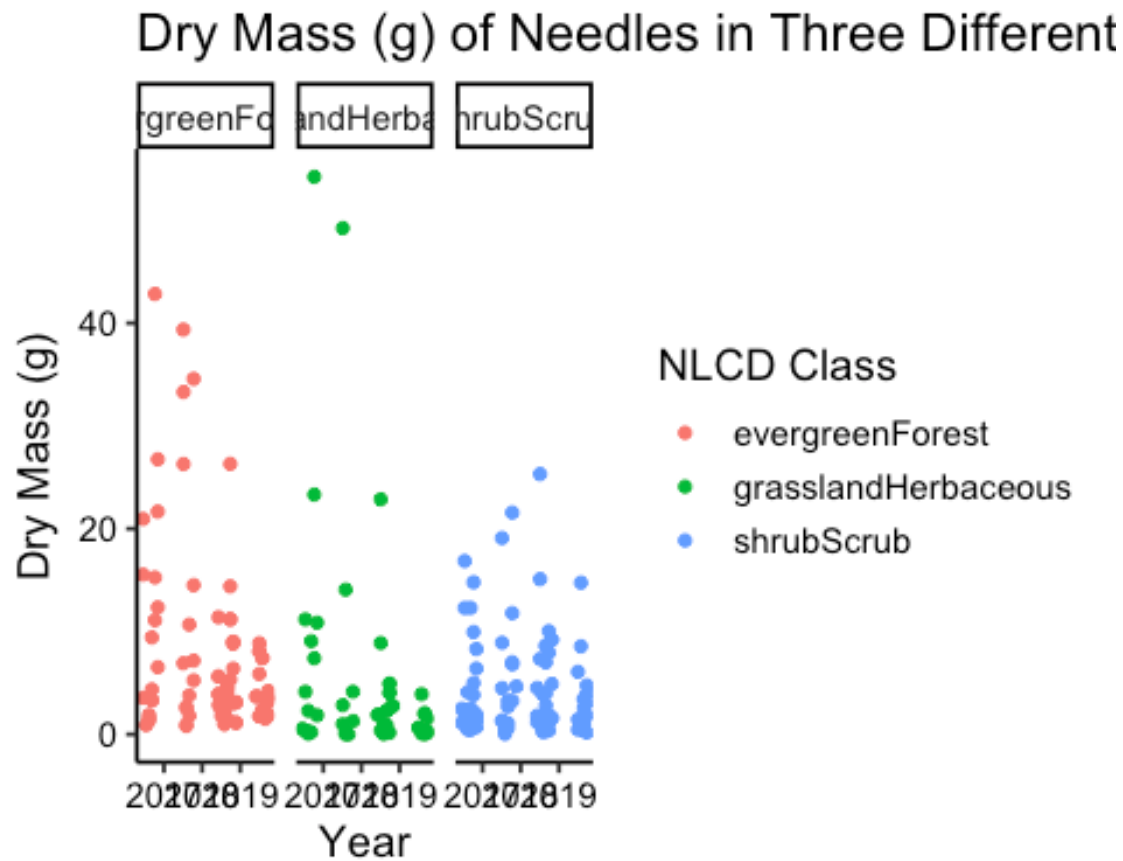
```



```

#7
facets.plot <-
  ggplot(subset(NEON_NIWO_Litter_Processed, functionalGroup == "Needles"),
aes(x= collectDate, y= dryMass, color= nlcdClass))+
  labs(x= "Year", y= expression ("Dry Mass (g)"), color = "NLCD Class")+
  geom_point()+
  theme(legend.position = "right")+
  facet_wrap(vars(nlcdClass))+
  ggtitle("Dry Mass (g) of Needles in Three Different Classifications of
Forests")
print(facets.plot)

```



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

Answer: I believe the facet plot is a better and more effective representation of the data because it clearly shows the different NLCD classes, how they relate to each other, how they relate across years, and clearly defines the range of dry mass in grams. In task six, separating the classes by color helped distinguish each individual class more; however, it leaves a confusing graphic that does not clearly define dry mass changes over the years and is hard to differentiate the class representation when dry mass is near 0 grams.

