Assignment 5: Data Visualization

Xueying Feng

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/ethel/Desktop/Environ 872/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyverse 1
```

```
## 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_conflic
## 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())
```

```
## ************************************************************
```

```
library(ggridges)

PeterPaul.chem <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.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
PeterPaul.chem$sampledate <- as.Date(PeterPaul.chem$sampledate, format = "%Y-%m-%d")
PeterPaul.gathered$sampledate <- as.Date(PeterPaul.gathered$sampledate, format = "%Y-%m-%d")
Litter$collectDate <- as.Date(Litter$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

```
mytheme <- theme_minimal(base_size = 12, base_family = "Times") +
  theme(axis.text.x = element_text(color = "DarkGrey"),
        legend.position = "top")

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.

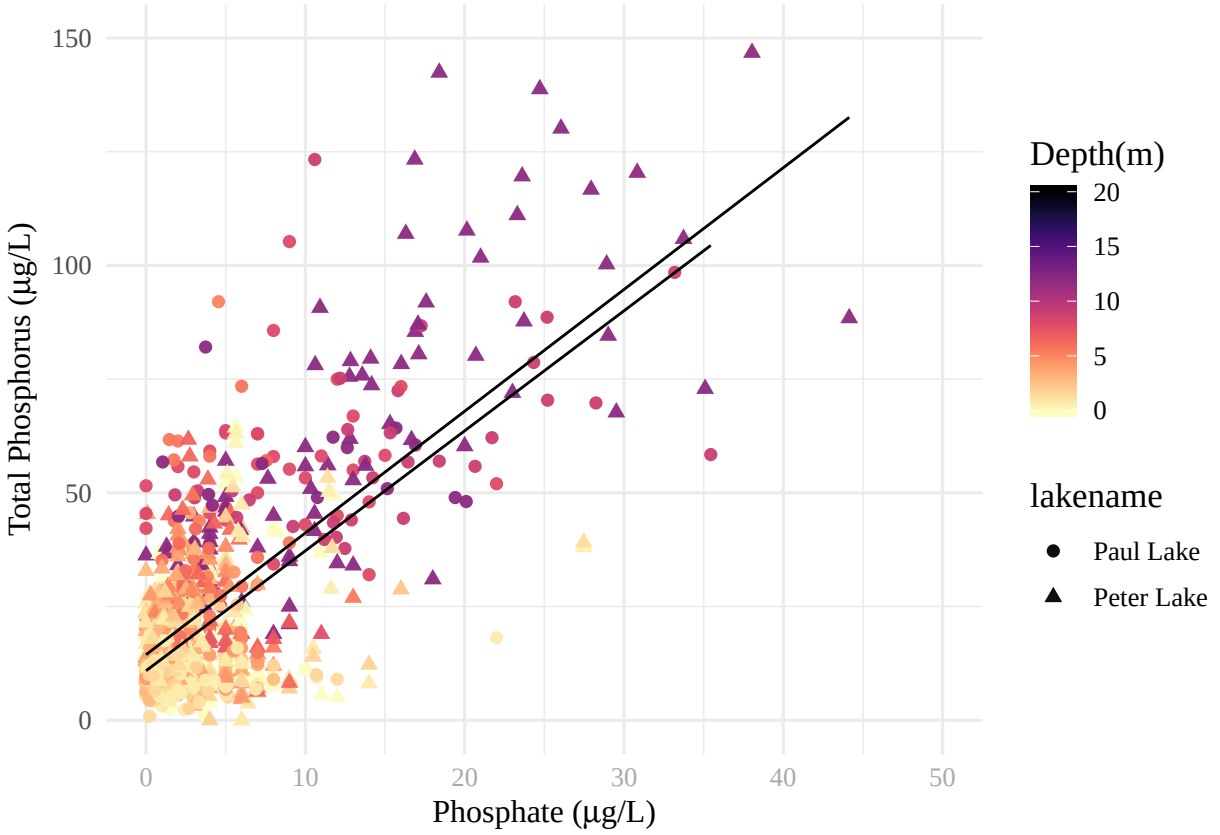
```
#install.packages("viridis")
library(viridis)
```

```
## Loading required package: viridisLite
```

```
NvsP <-
  ggplot(PeterPaul.chem, aes(x = po4, y = tp_ug, color = depth, shape = lakename)) +
  geom_point(alpha = 0.95, size = 2) +
  labs(x=expression(paste("Phosphate (",mu,"g/L)"))) +
  labs(y=expression(paste("Total Phosphorus (",mu,"g/L)")))+
  labs(color="Depth(m)")+
  scale_shape_manual(values = c(16,17)) +
  scale_color_viridis(option = "A", direction = -1) +
  theme(legend.position = "right",
        legend.text = element_text(size = 10), legend.title = element_text(size = 13))+
  geom_smooth(method = "lm", colour="black", size=0.5, se = FALSE)+
  xlim(0, 50) +
  ylim(0, 150)

print(NvsP)
```

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

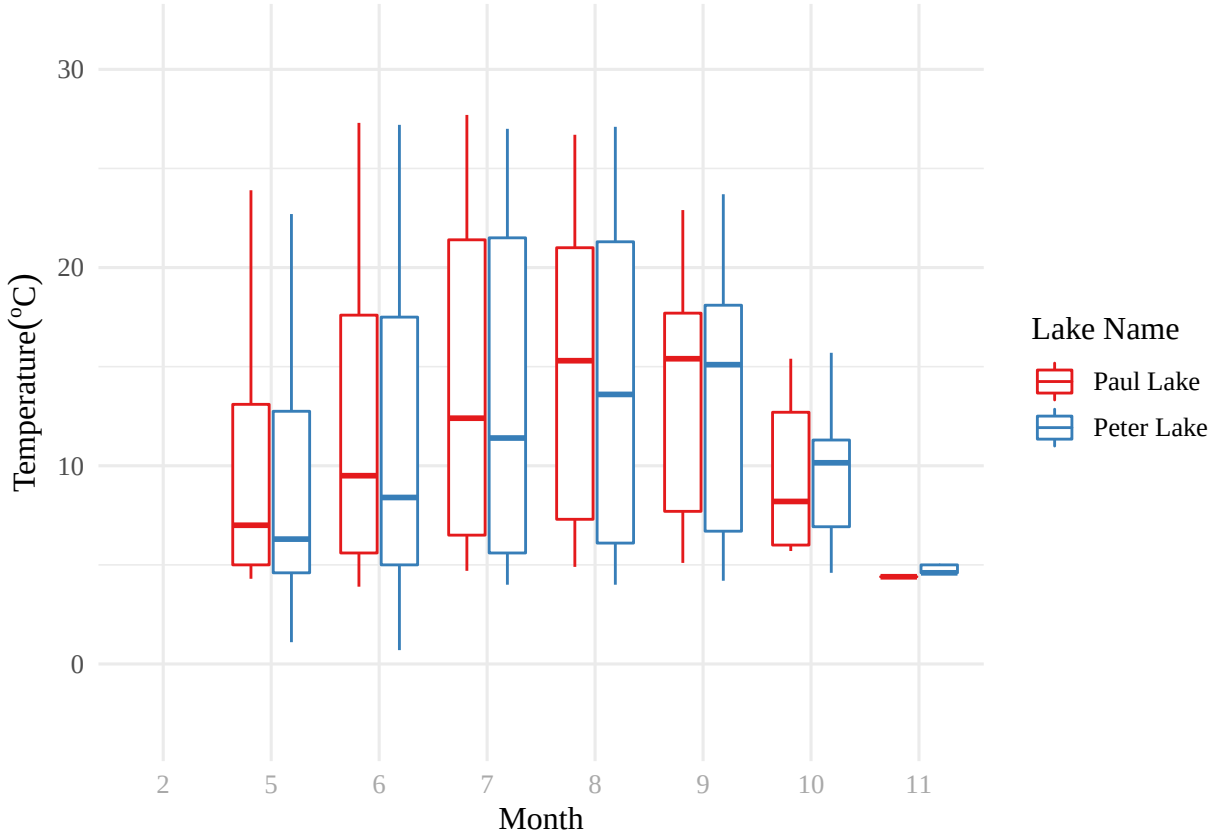
```
## Warning: Removed 21948 rows containing missing values (geom_point).
```



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.

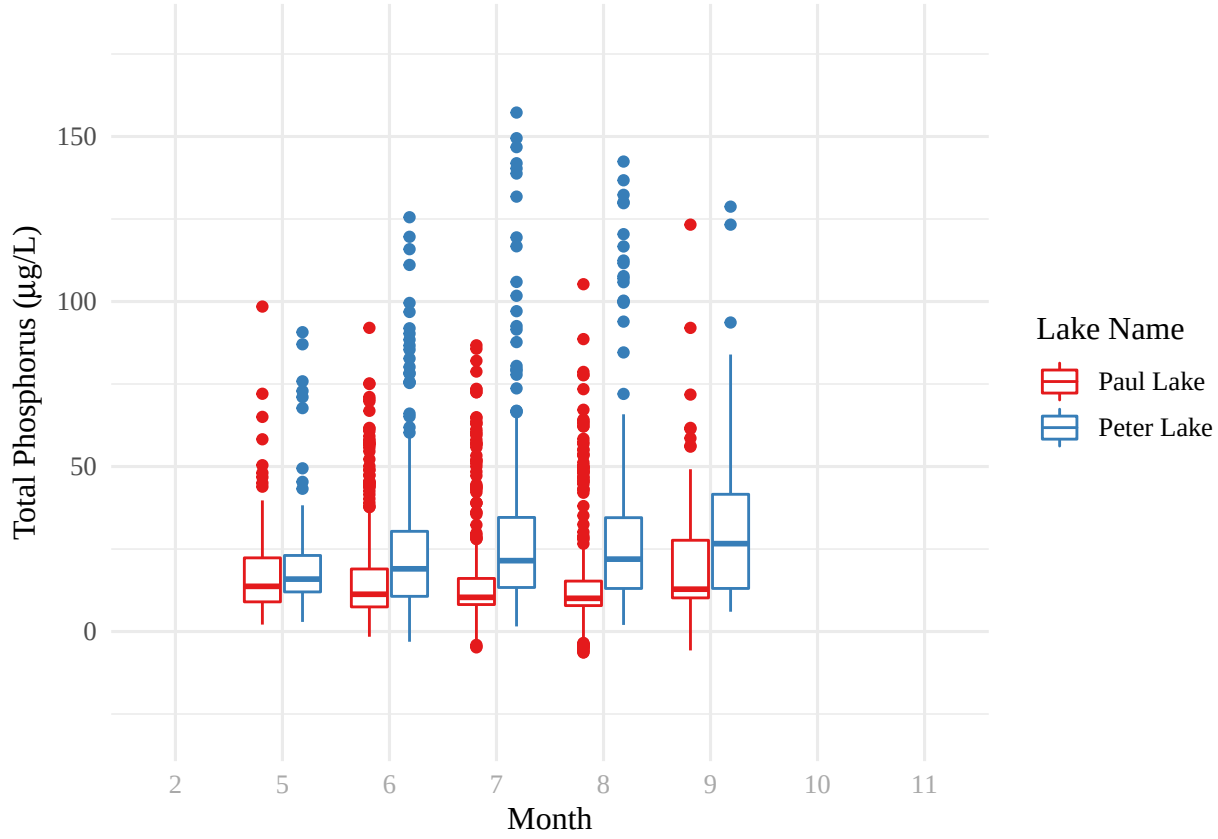
```
#install.packages("RColorBrewer")
library(RColorBrewer)

library(ggplot2)
# (a)
Tempplot <-
  ggplot(PeterPaul.chem, aes(x = as.factor(month), y = temperature_C, color = lakename)) +
  #facet_wrap("lakename")+
  geom_boxplot() +
  labs(x=expression(paste("Month"))) +
  labs(y=expression(Temperature (''^o*C)))+
  labs(color="Lake Name")+
  scale_y_continuous(expand = c(0.2, 0.2)) +
  scale_color_brewer(type = 'qual', palette = 'Set1')+
  theme(legend.position = "right")
print(Tempplot)
```

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



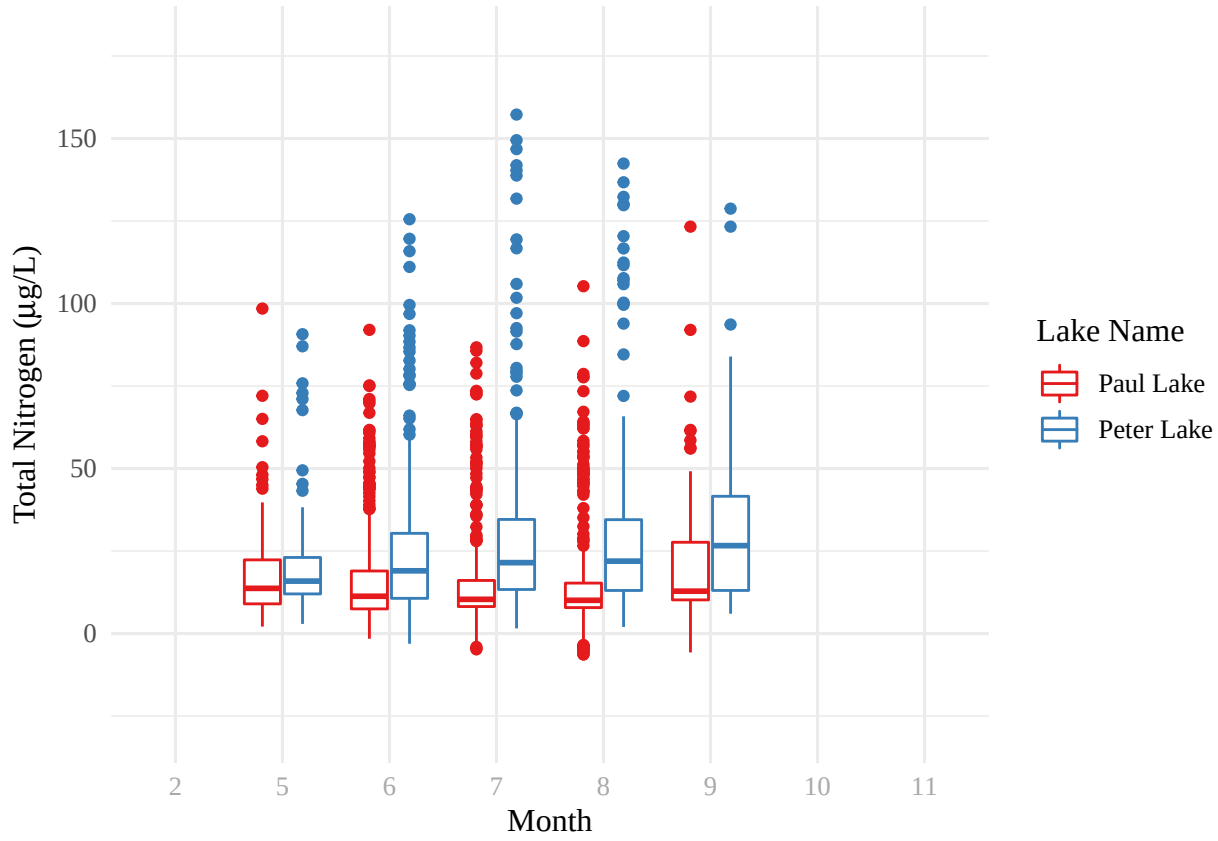

```
# (b)
TPplot <-
  ggplot(PeterPaul.chem, aes(x=as.factor(month), y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(x=expression(paste("Month"))) +
  labs(y=expression(paste("Total Phosphorus (",mu,"g/L)")))+
  labs(color="Lake Name")+
  scale_y_continuous(expand = c(0.2, 0.2)) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "right")
print(TPplot)
```

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




```
# (c)
TNplot <-
  ggplot(PeterPaul.chem, aes(x = as.factor(month), y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(x=expression(paste("Month"))) +
  labs(y=expression(paste("Total Nitrogen (",mu,"g/L)")))+
  labs(color="Lake Name")+
  scale_y_continuous(expand = c(0.2, 0.2)) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "right")
print(TNplot)
```

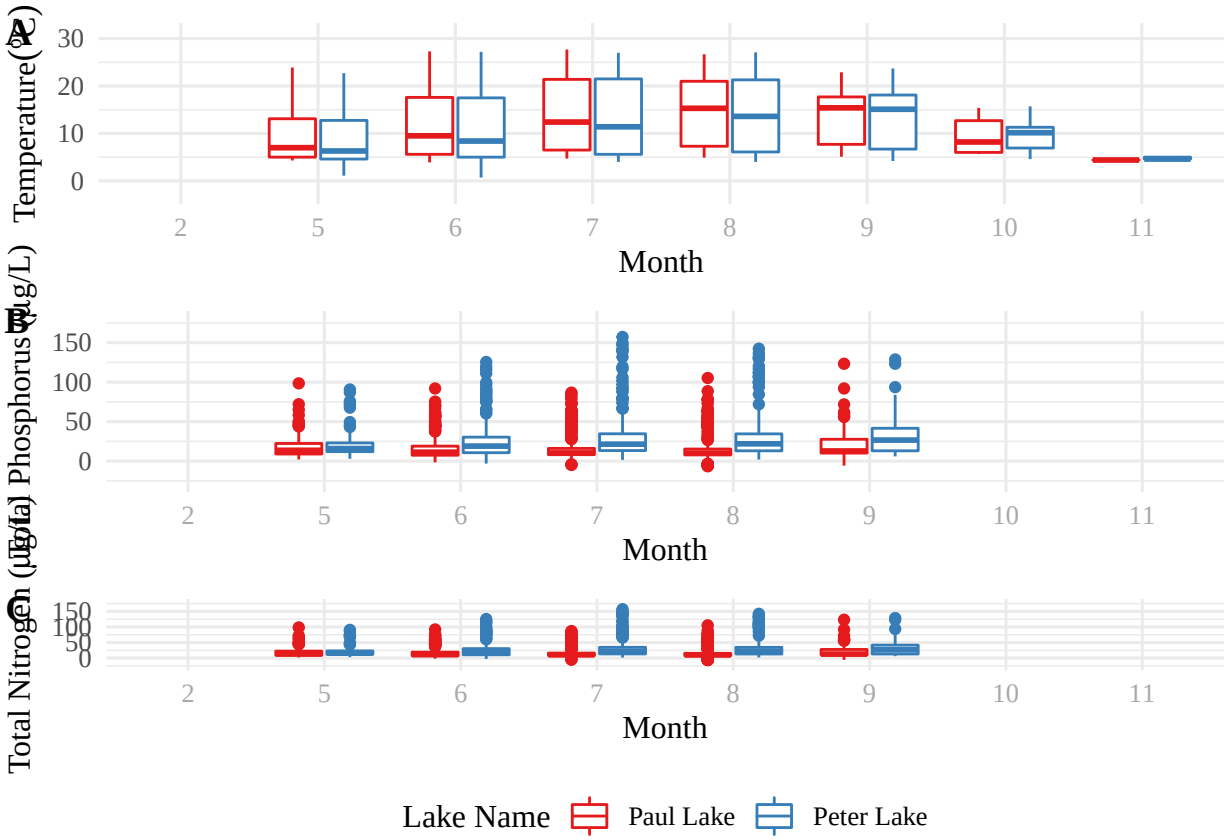
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```




```
#combined three graph three graph
library(cowplot)
CombinedPlot<-  plot_grid(Tempplot+theme(legend.position="none"),
                          TPplot+theme(legend.position="none"),
                          TNplot+theme(legend.position="bottom"),
                          nrow=3, labels=c('A', 'B', 'C'))
```

```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```

```
print(CombinedPlot)
```




```
#dev.off()  #cowplot" sometimes does not work, stack overflow give this methods
```

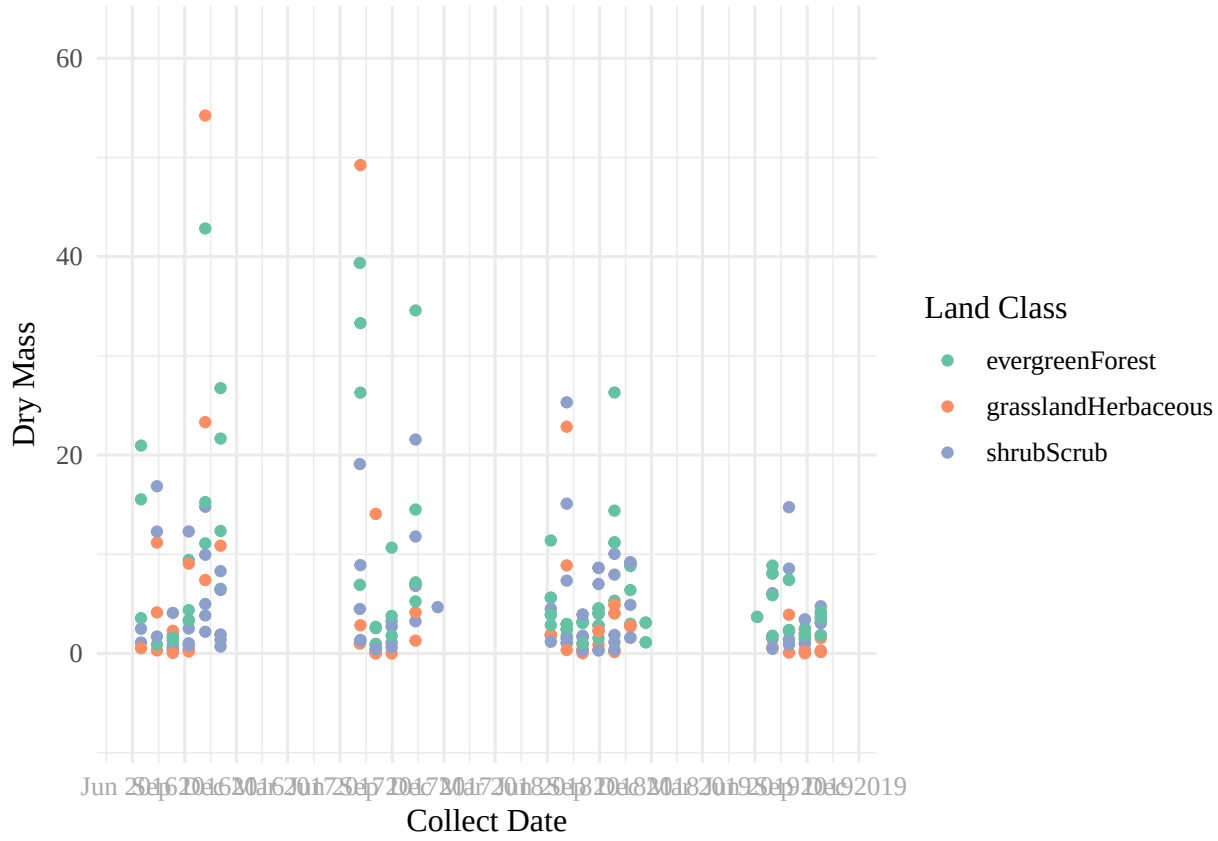
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer:

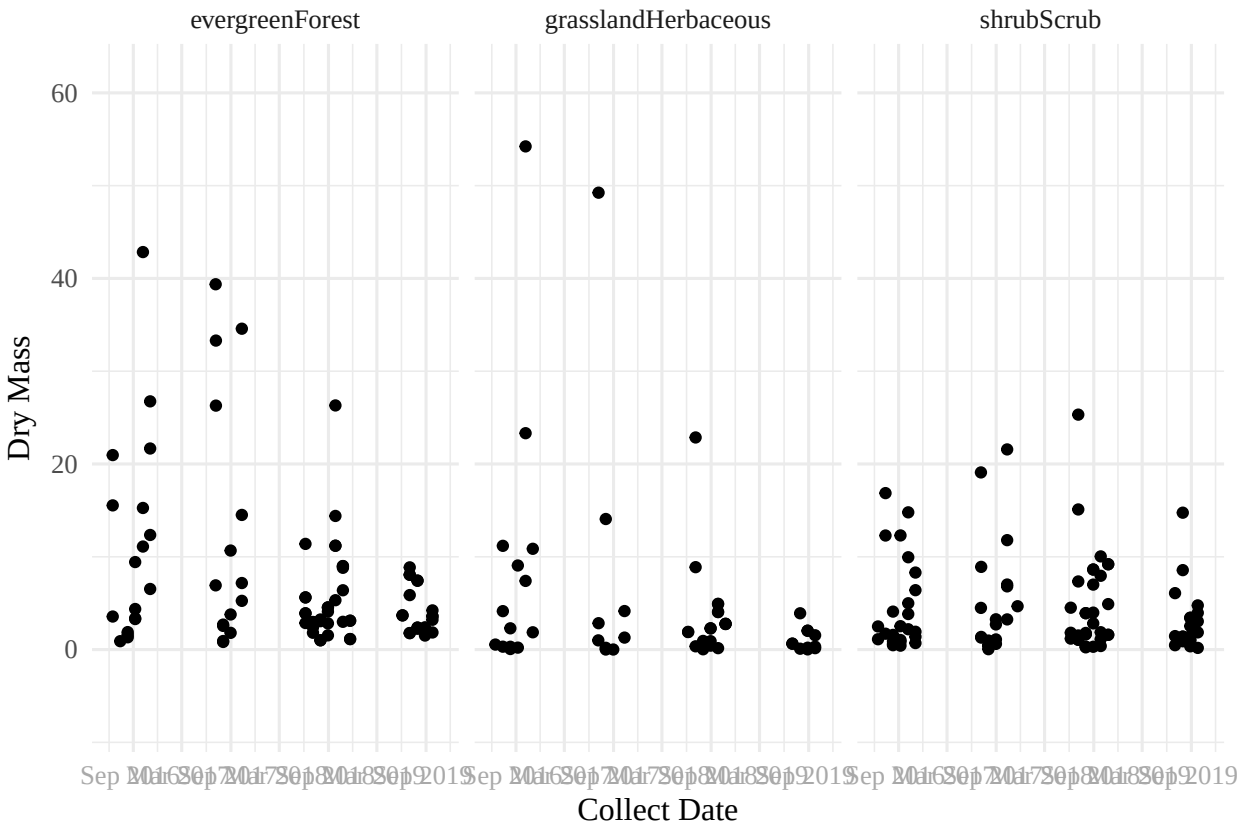
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
NeedlesPlot <-
  ggplot(subset(Litter, functionalGroup=="Needles")) +
  geom_point(aes(x = collectDate, y = dryMass, color=nlcdClass))+
  scale_x_date(limits = as.Date(c("2016-06-01", "2019-10-31")),
               date_breaks = "3 months", date_labels = "%b %Y ") +
  labs(x=expression(paste("Collect Date"))) +
  labs(y=expression(paste("Dry Mass")))+
  labs(color="Land Class")+
  scale_y_continuous(expand = c(0.2, 0.2)) +
  scale_color_brewer(type = 'qual', palette = 'Set2') +
  theme(legend.position = "right")
print(NeedlesPlot)
```




```
#7
NeedlesPlot_facet <-
  ggplot(subset(Litter, functionalGroup=="Needles")) +
  geom_point(aes(x = collectDate, y = dryMass))+
  scale_x_date(limits = as.Date(c("2016-06-01", "2019-10-31")),
               date_breaks = "6 months", date_labels = "%b %Y ") +
  labs(x=expression(paste("Collect Date"))) +
  labs(y=expression(paste("Dry Mass")))+
  scale_y_continuous(expand = c(0.2, 0.2)) +
  facet_wrap(~ nlcdClass, ncol=3)
print(NeedlesPlot_facet)
```



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

Answer: I think #6 is more effective, because it is more clear to compare dry mass of each class at same time. In #7, all classes data seperately, which is hard to compare