Assignment 5: Data Visualization

Alicia Zhao

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.

```
getwd()
```

```
## [1] "/Users/mac/Desktop/Data Analytics/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## 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 ------------------------------------------------------------------------------------
## 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"
```

```
# R is not reading dates as date format, so will need to convert them.

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")
```

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 = "gray35"),
        legend.position = "right")
```

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.

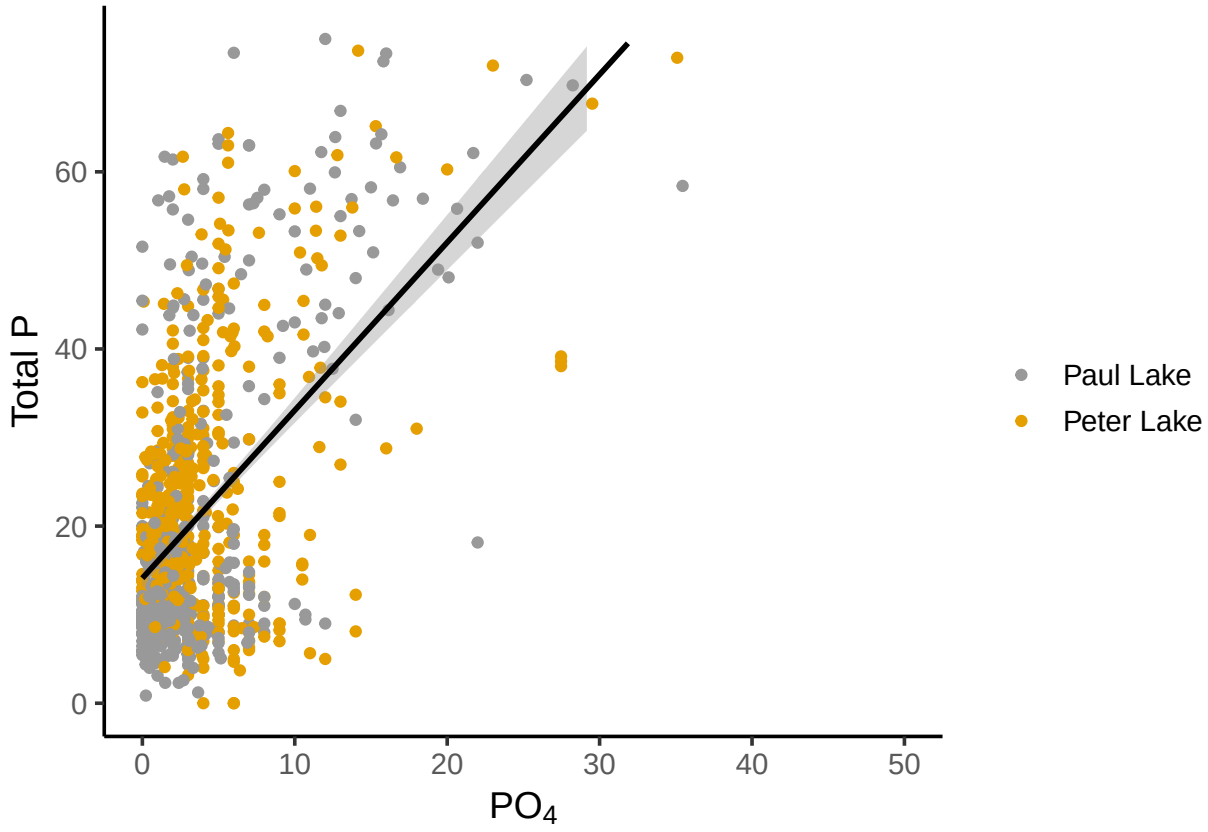
```
TPvP <- ggplot(PeterPaul.chem.nutrients, aes(y=tp_ug, x=po4, color=lakename)) +
  geom_point() +
  geom_smooth(method = lm, color="black") +
  mytheme +
  xlim(0, 50) +
  ylim(0,75) +
  labs(x = expression("PO"[4]) , y = "Total P", color="") +
  scale_color_manual(values = c("#999999", "#E69F00"))
```

```
print (TPvP)
```

```
## Warning: Removed 21988 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21988 rows containing missing values (geom_point).
```

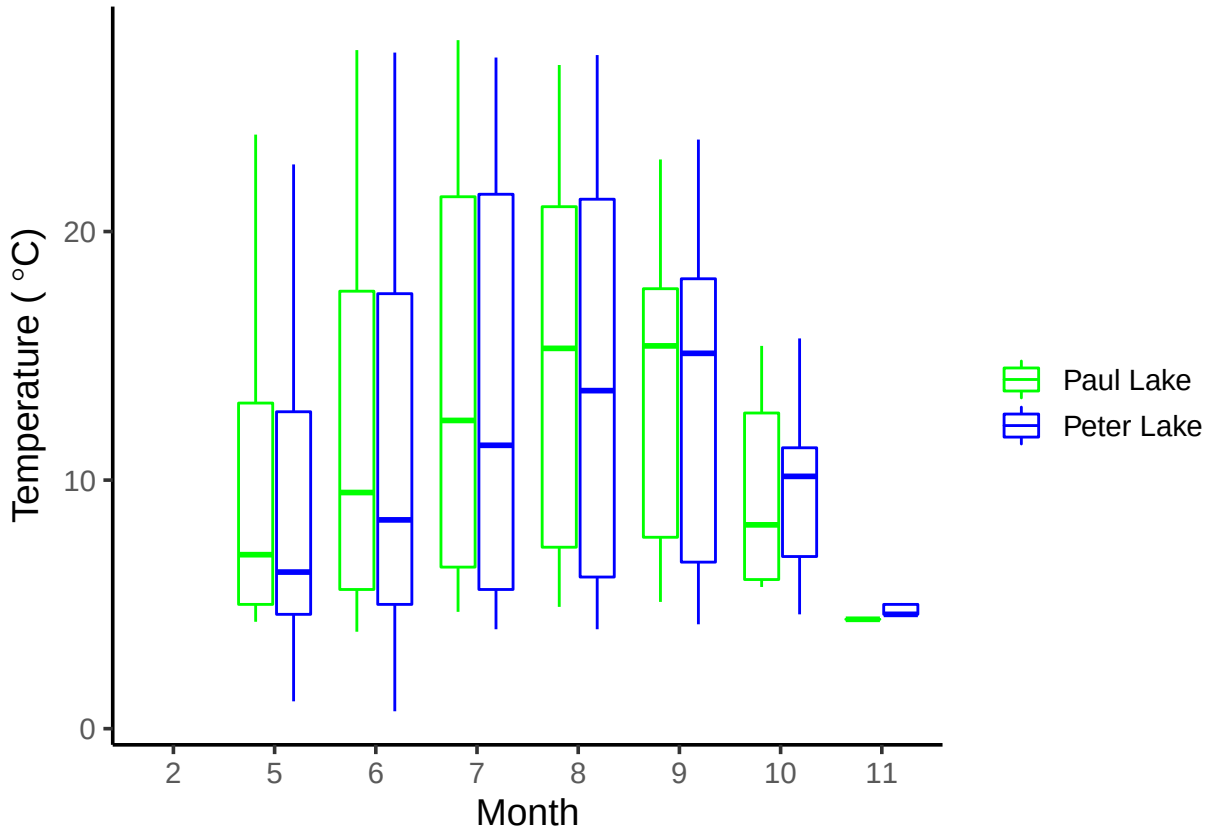
```
## Warning: Removed 8 rows containing missing values (geom_smooth).
```



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.

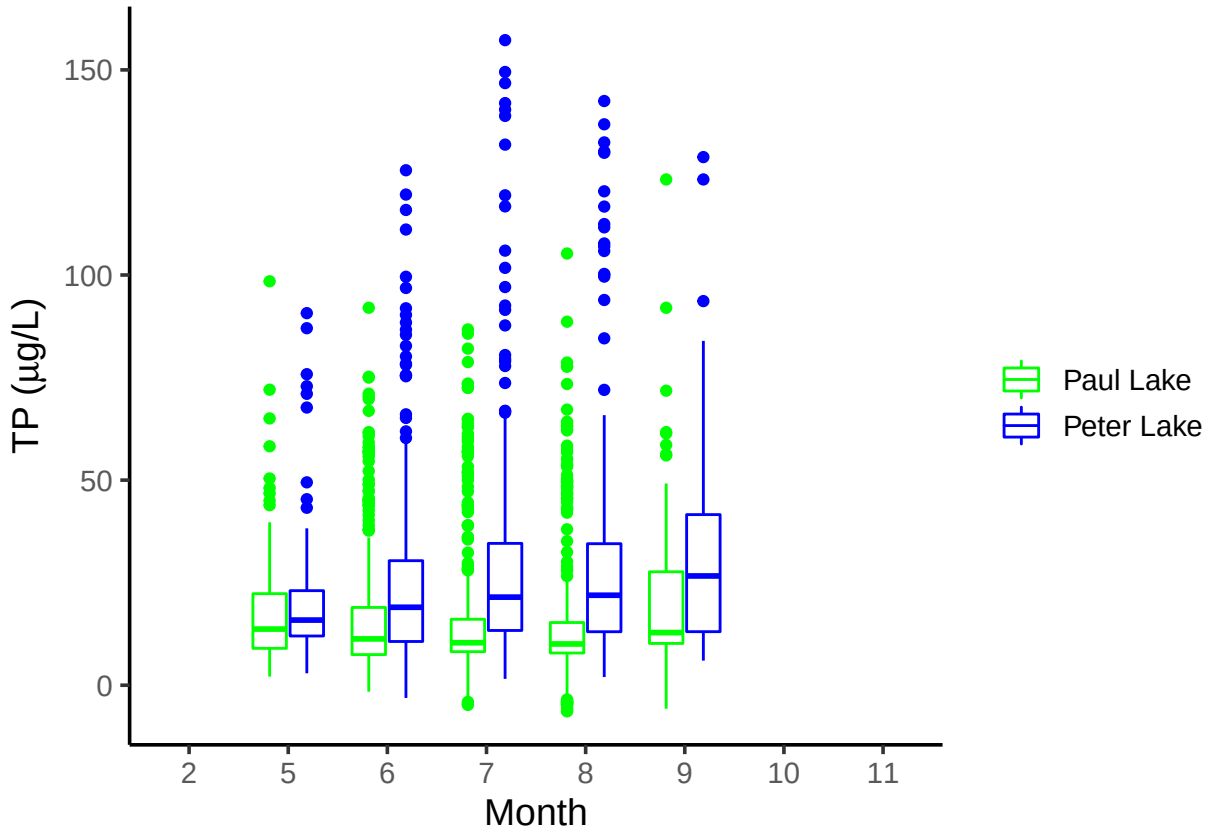
```
Temp.boxplot <- ggplot (PeterPaul.chem.nutrients, aes(y = temperature_C, x = as.factor(month), color = l
  geom_boxplot() +
  mytheme +
  labs (x = "Month", y = expression(paste("Temperature (", ~degree, "C)")), color="") +
  scale_color_manual(values=c("green", "blue"))
print(Temp.boxplot)
```

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



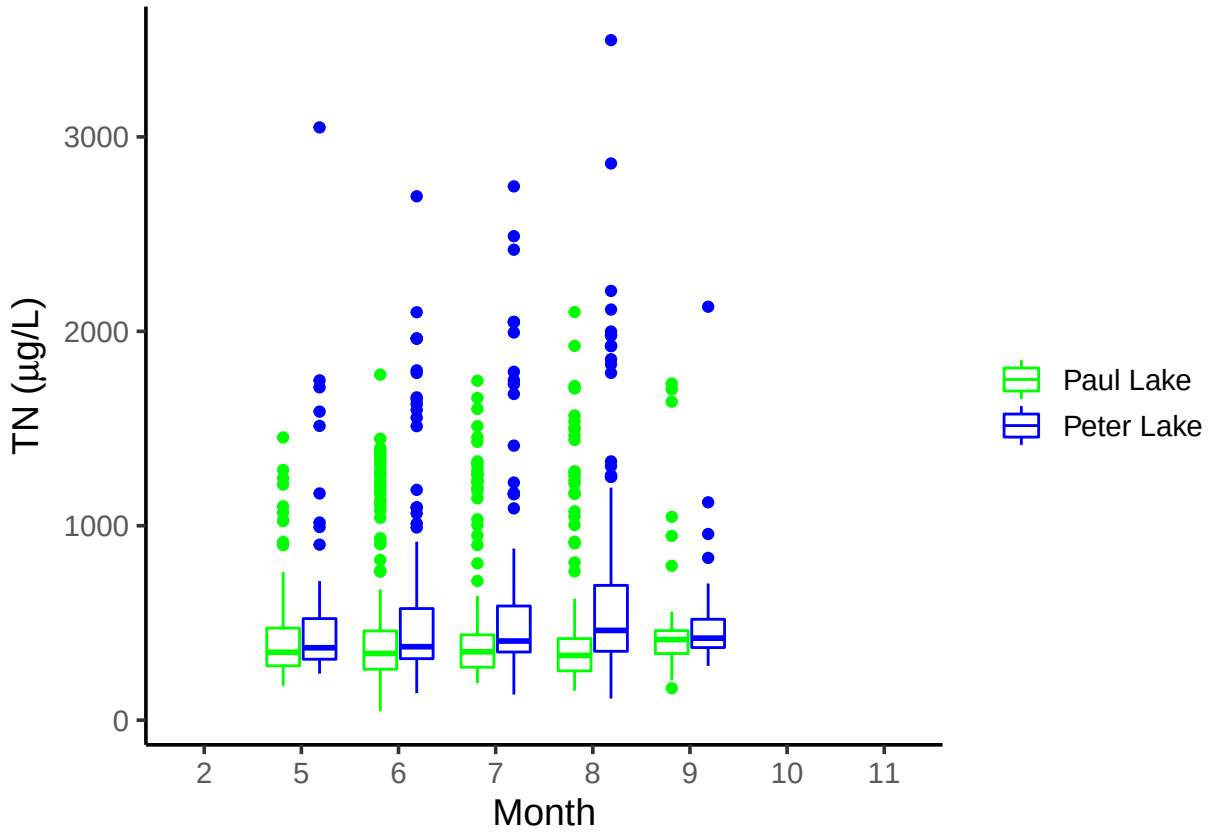

```
TP.boxplot <- ggplot (PeterPaul.chem.nutrients, aes(y = tp_ug, x = as.factor(month))) +
  geom_boxplot(aes(color=lakename)) +
  mytheme +
  labs (x = "Month", y = expression(paste("TP (", mu, "g/L)")), color="") +
  scale_color_manual(values=c("green", "blue"))
print(TP.boxplot)
```

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




```
TN.boxplot <- ggplot (PeterPaul.chem.nutrients, aes(y = tn_ug, x = as.factor(month))) +
  geom_boxplot(aes(color=lakename)) +
  mytheme +
  labs (x = "Month", y = expression(paste("TN (", mu, "g/L)")), color="")  +
  scale_color_manual(values=c("green", "blue"))
print(TN.boxplot)
```

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




```
joint.boxplot <- plot_grid(
  Temp.boxplot + theme (legend.position = "none") + labs(x=""),
  TP.boxplot + theme (legend.position = "none") + labs(x=""),
  TN.boxplot + theme (legend.position = "none"),
  nrow = 3, align = 'v')
```

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

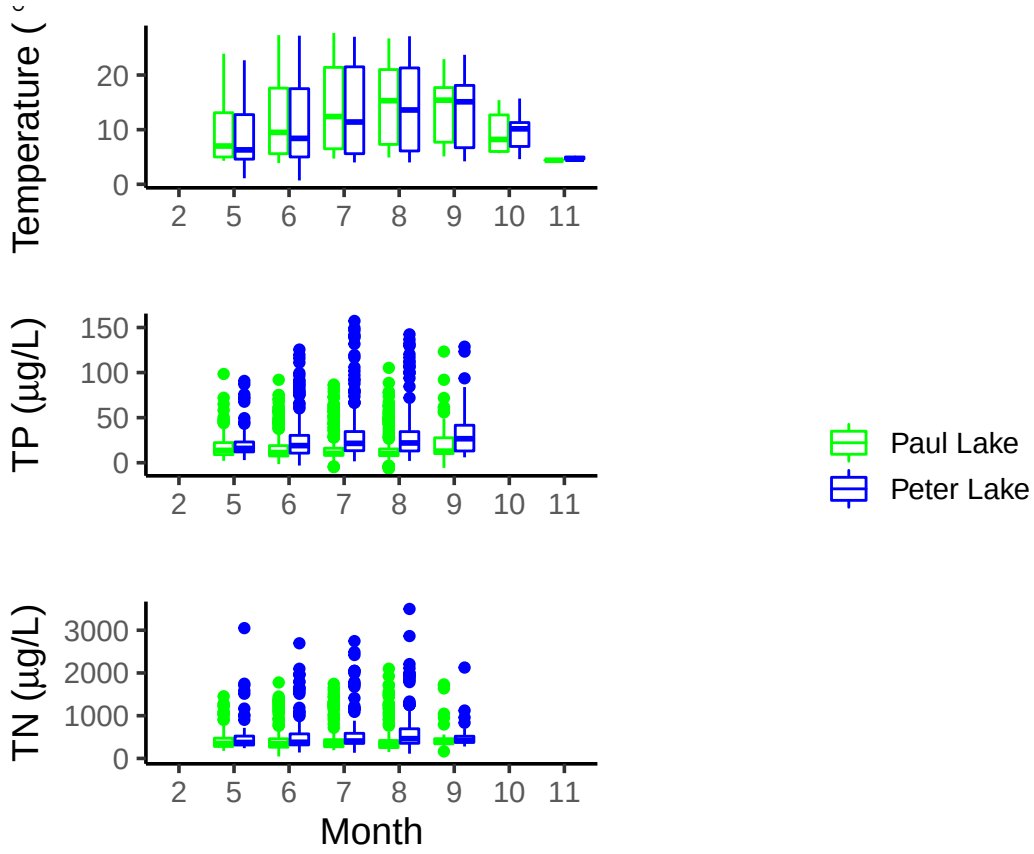
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```

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

```
legend <- get_legend(Temp.boxplot)
```

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

```
plot_grid (joint.boxplot, legend)
```



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

> Answer: As one would expect, temperature is higher during the summer seasons (especially August and September) and lower during the other seasons for both lakes. Peter Lake has a lower median temperature than Paul Lake over all months except for October and November. Paul Lake has lower total phosphorous (TP) than Peter Lake over all months. TP concentrations for Peter Lake increase from May through September. TP concentrations for Paul Lake do not exhibit the same pattern, and remain approximately the same across months. Paul Lake has lower total nitrogen (TN) than Peter Lake overall. TN concentrations for Peter Lake increases from May through August, and then drops slightly in September. TN Concentrations for PAul Lake remain approximately the same across months, with a slight increase in September.

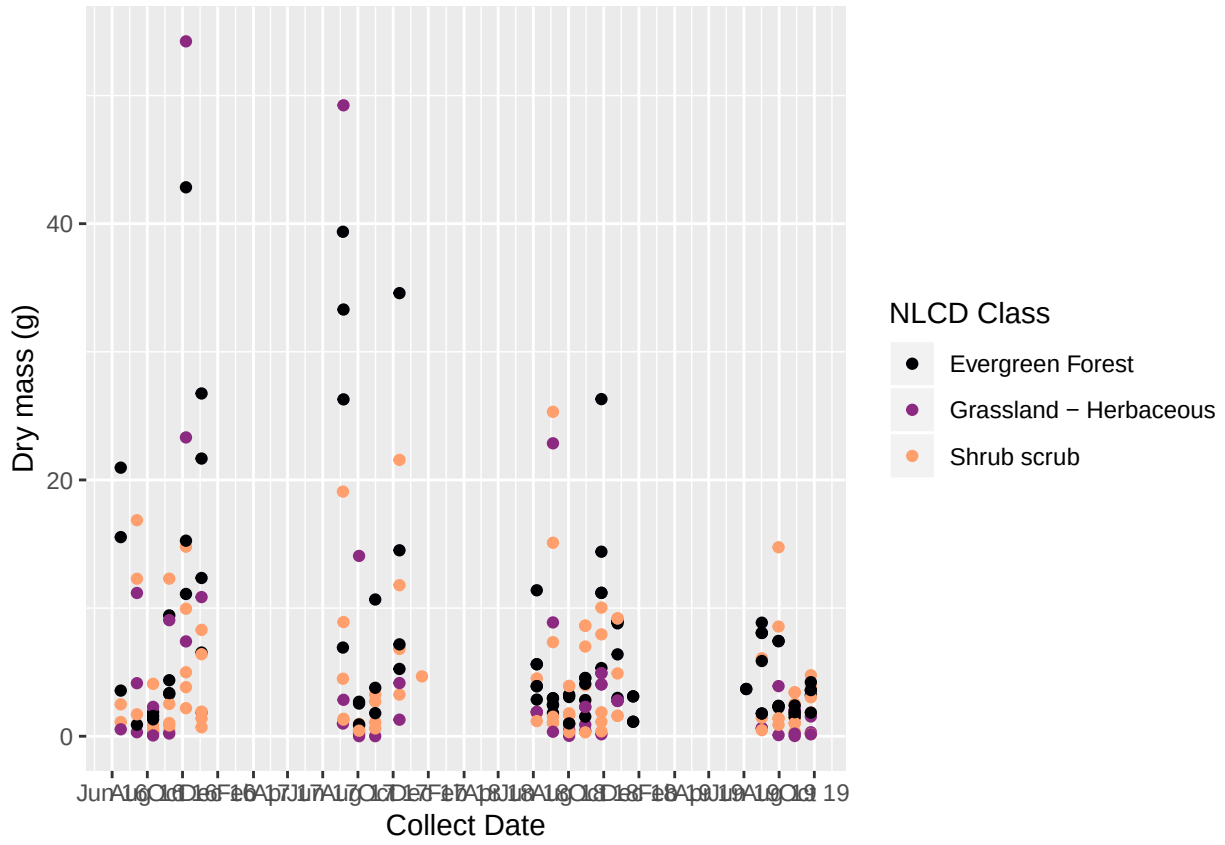
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

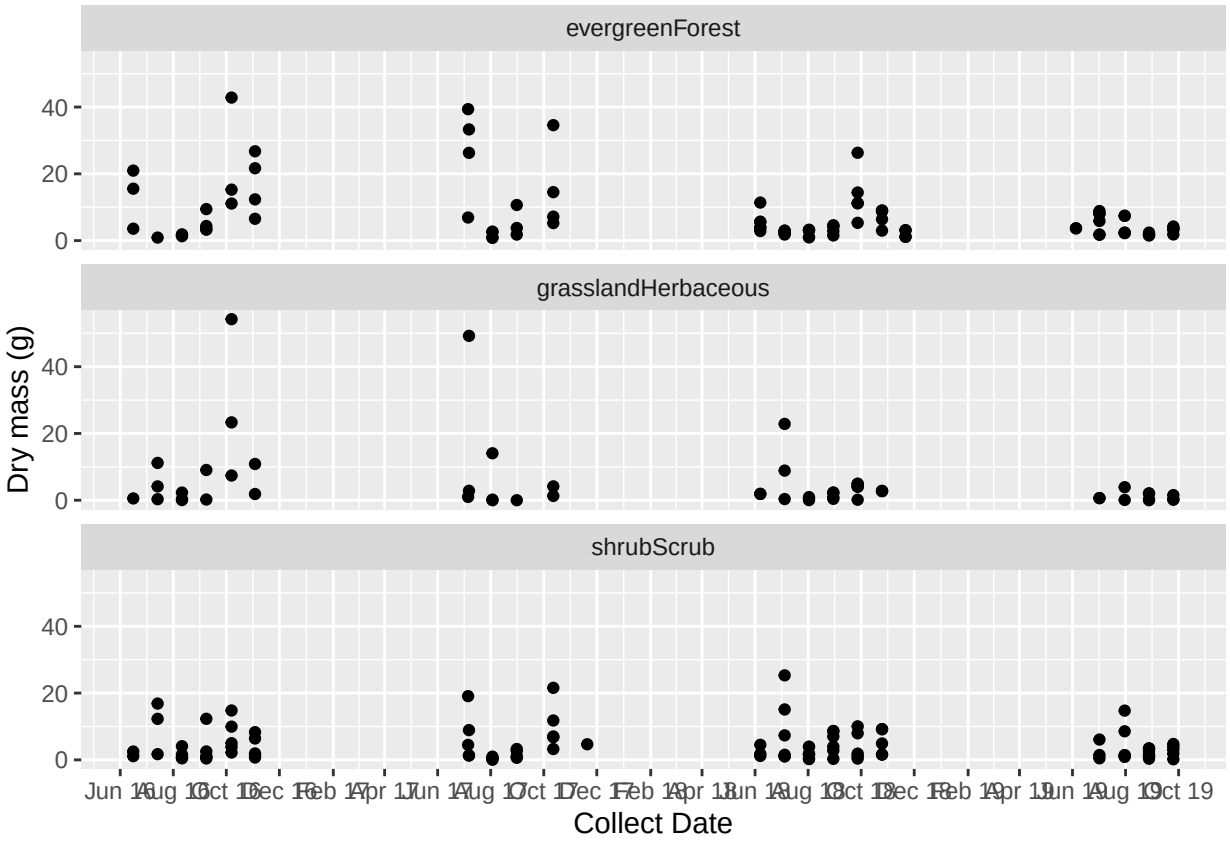
```
library(viridis)
```

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

```
ggplot((subset(Litter, functionalGroup == "Needles"))) +
  geom_point(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  labs (x ="Collect Date", y ="Dry mass (g)", color = "NLCD Class") +
    scale_x_date(date_breaks = "2 months", date_labels = "%b %y") +
    scale_color_viridis(discrete = TRUE, option = "magma", end = 0.8,
                        labels = c("Evergreen Forest", "Grassland - Herbaceous", "Shrub scrub"))
```




```
ggplot((subset(Litter, functionalGroup == "Needles"))) +
  geom_point(aes(x = collectDate, y = dryMass)) +
  labs (x ="Collect Date", y ="Dry mass (g)") +
    scale_x_date(date_breaks = "2 months", date_labels = "%b %y") +
    scale_color_viridis(discrete = TRUE, option = "magma", end = 0.8,
                        labels = c("Evergreen Forest", "Grassland - Herbaceous", "Shrub scrub"))+
  facet_wrap(vars(nlcdClass), nrow=3)
```



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

Answer: Plot 6 is more effective because it overlays the threeclasses in different colors, which allows us to more easily discern differences in distribution. Plot 7 presents the 3 classes separately, but it is more difficult to tell how they differ from each other.