Assignment 5: Data Visualization

Shirley Fontanié

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 Monday, February 14 at 7:00 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 (use the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed.csv`] version).

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
#getwd()
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

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

```
##
## Attaching package: 'lubridate'
```

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

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)

#2

Nutrients <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", str

Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                   stringsAsFactors = TRUE, header = TRUE)

class(Nutrients$sampledate)
```

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

```
class(Litter$collectDate)
```

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

```
Nutrients$sampledate <- as.Date(Nutrients$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.

```
#3
ShirleysStylishTheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(ShirleysStylishTheme)
```

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 (hint: change the limits using `xlim()` and `ylim()`).

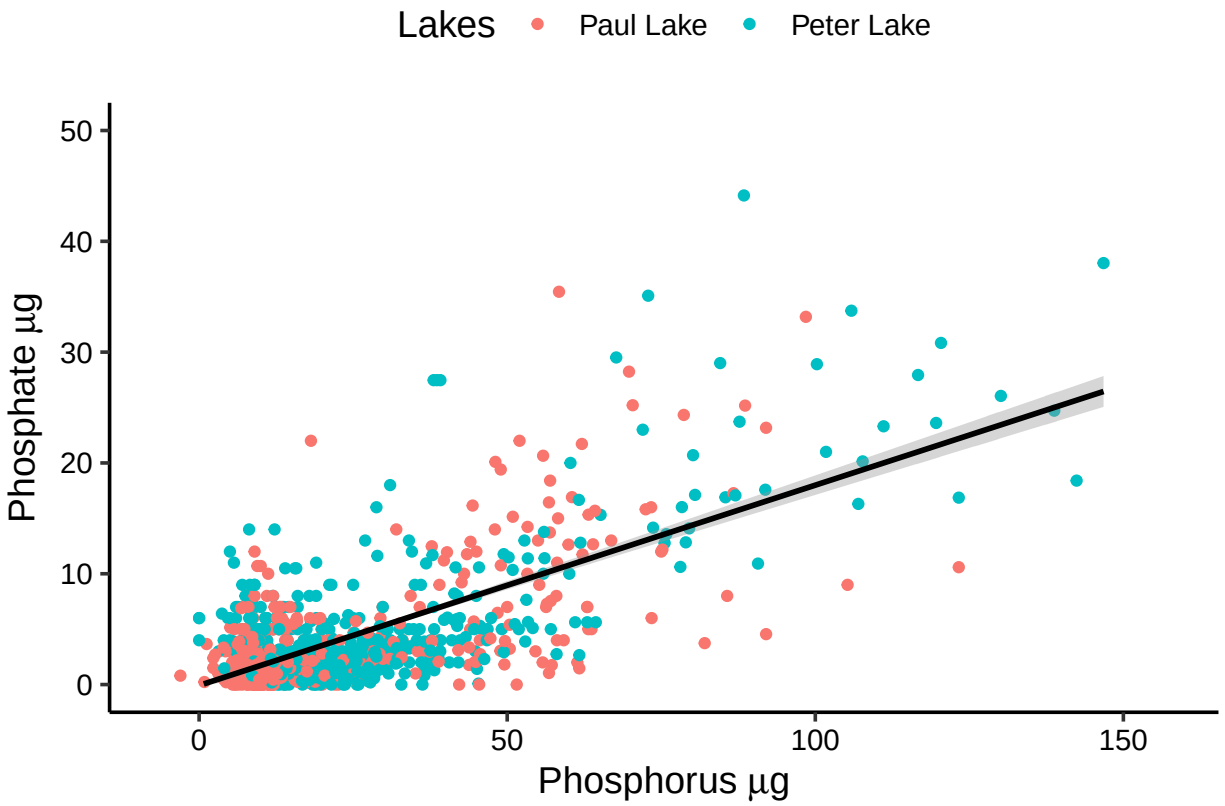
```
#4
PeterPaulPlot1 <-
  ggplot(Nutrients, aes(x = tp_ug, y = po4, color = lakename))+
  geom_point()+
  geom_smooth(method=lm, color = "black")+
  xlab(expression(paste("Phosphorus ", mu, "g")))+
  ylab(expression(paste("Phosphate " , mu, "g")))+
  ylim(0,50)+
  scale_color_discrete(name = "Lakes")
print(PeterPaulPlot1)
```

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

```
## Warning: Removed 21947 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21947 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.

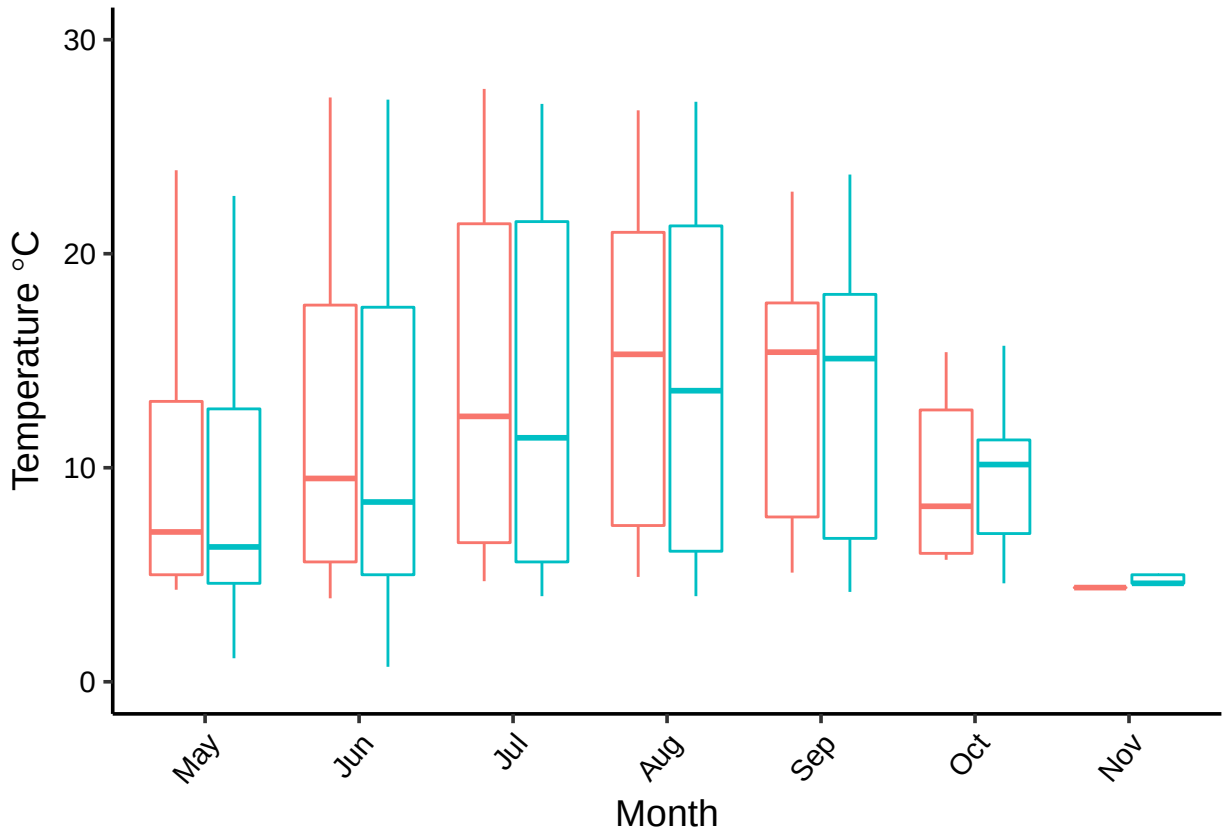
```
#5

# how to fix months on x axis.
# Luana recommends changing class of month from numeric to factor
Nutrients$month <- as.factor(Nutrients$month)
# you can remove month label, but leave a comment explaining


Temperatureplot <-
  ggplot(Nutrients, aes(x = month, y = temperature_C, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  ylab(expression(paste("Temperature " , degree, "C")))+
  ylim(0,30)+
  #scale_color_discrete(names = "")+
  scale_x_discrete(limits=factor("5":"11"),
  labels=c("May","Jun","Jul","Aug","Sep","Oct","Nov"))+
  theme(axis.text.x = element_text(angle = 50,  hjust = 1))+
  theme(legend.position = "none")
print(Temperatureplot)
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
```

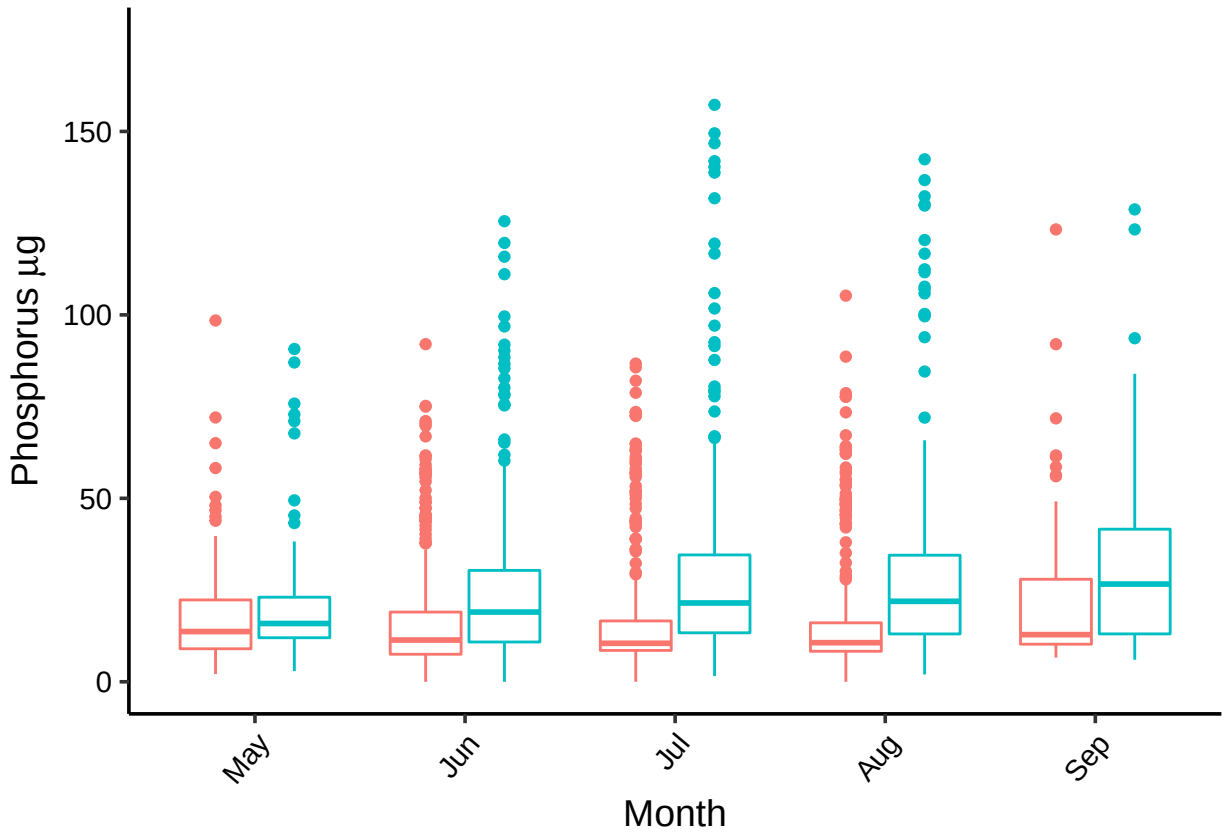
```
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
```




```
TPplot <-
  ggplot(Nutrients, aes(x = month, y = tp_ug, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  ylab(expression(paste("Phosphorus ", mu, "g")))+
  ylim(0,175)+
  scale_color_discrete(name = "Lakes")+
  scale_x_discrete(limits=factor("5":"9"),
  labels=c("May","Jun","Jul","Aug","Sep"))+
  theme(axis.text.x = element_text(angle = 50,  hjust = 1))+
  theme(legend.position = "none")
print(TPplot)
```

```
## Warning: Removed 205 rows containing missing values (stat_boxplot).
```

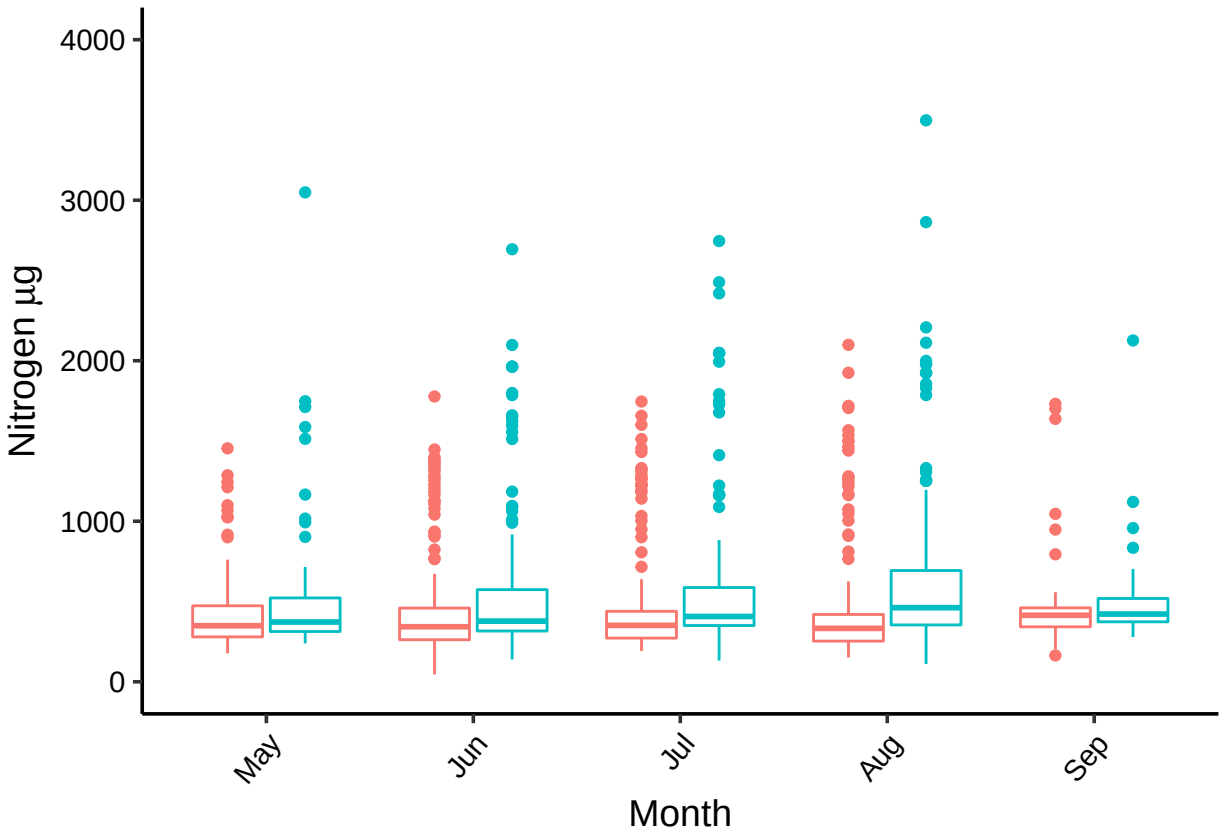
```
## Warning: Removed 20561 rows containing non-finite values (stat_boxplot).
```




```
TNPlot <-
  ggplot(Nutrients, aes(x = month, y = tn_ug, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  ylab(expression(paste("Nitrogen ", mu,"g")))+
  ylim(0, 4000)+
  scale_color_discrete(name= "Lakes")+
  scale_x_discrete(limits=factor("5":"9"),
  labels=c("May","Jun","Jul","Aug","Sep"))+
  theme(axis.text.x = element_text(angle = 50,  hjust = 1))+
  theme(legend.position = "none")
print(TNPlot)
```

```
## Warning: Removed 205 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 21378 rows containing non-finite values (stat_boxplot).
```



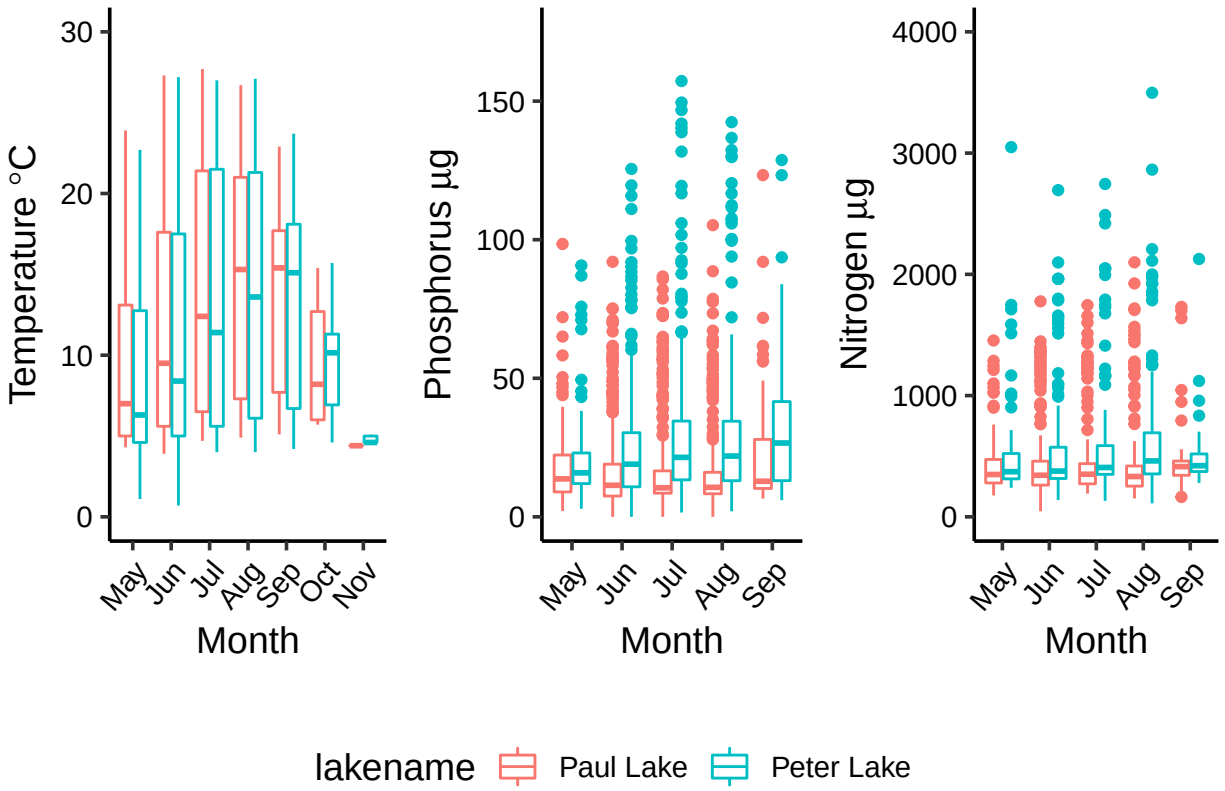

```
PRACTICE.PLOT <- plot_grid(Temperatureplot,TPplot,TNPlot, nrow = 1)
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
## Warning: Removed 205 rows containing missing values (stat_boxplot).
## Warning: Removed 20561 rows containing non-finite values (stat_boxplot).
## Warning: Removed 205 rows containing missing values (stat_boxplot).
## Warning: Removed 21378 rows containing non-finite values (stat_boxplot).
```

```
TriplePlotLegend <- get_legend(Temperatureplot + theme(legend.position = "bottom"))
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
```

```
FINAL.PLOT <- plot_grid(PRACTICE.PLOT, TriplePlotLegend, nrow = 2, rel_heights = c(2, 0.5))
print(FINAL.PLOT)
```




```
# 3 graphs -> remove legends for the graphs
# put together 3 graphs without legend, use plotgrid function, legendposition= NONE
# make an object that is just legend, use get_legend
# get legend from one of the plots, set legend position, there
# use plotgrid again, combining 3 plots, then combine legend object
```

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

> Answer: Over seasons, as summer progresses Peter Lake seems to have higher nitrogen and phosphorus levels than Paul Lake. While Paul Lake seems to have slightly higher median temperature in degrees Celsius than Peter Lake. Also the range of the levels between phosphorus and nitrogen is significant. Phosphorus reaches levels of >150 mu grams, while nitrogen reaches levels of >3,000 mu grams.

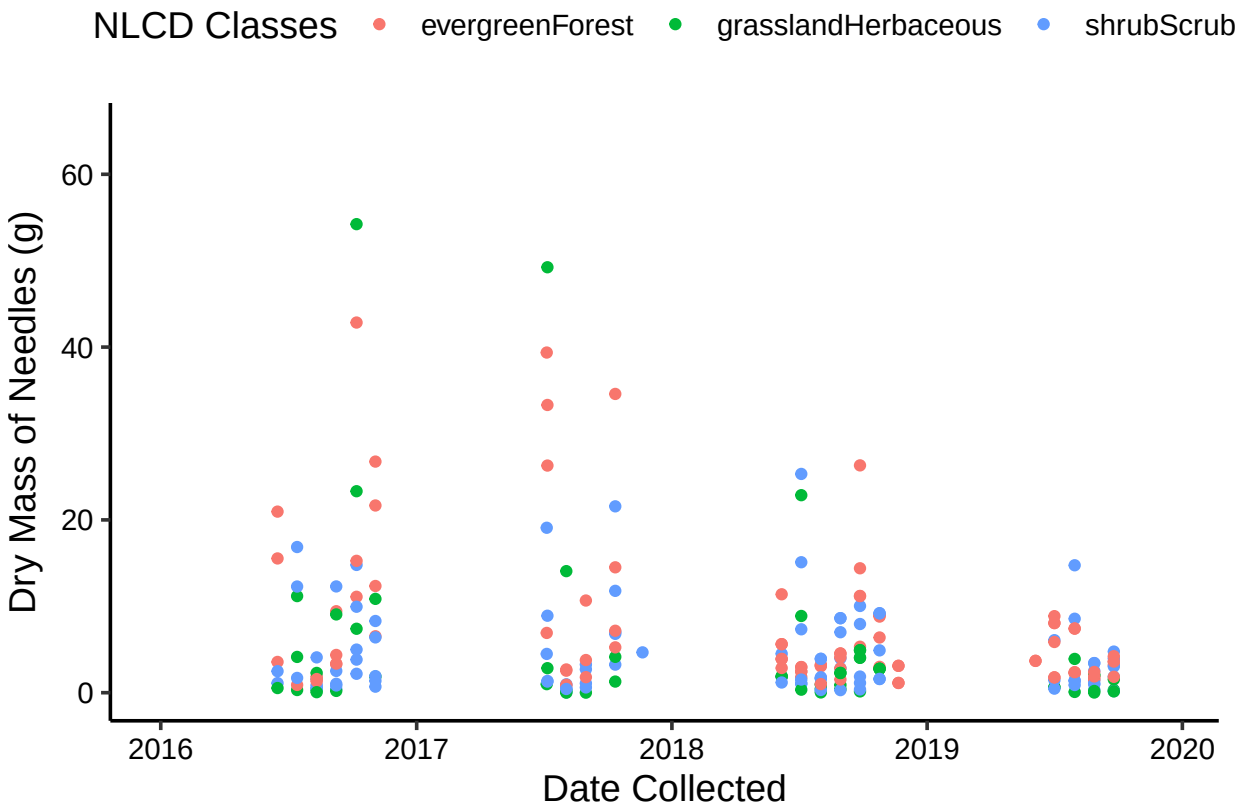
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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"),
  aes(x = collectDate, y = dryMass, color = nlcdClass))+
  geom_point()+
  xlab("Date Collected")+
  ylab(expression(paste("Dry Mass of Needles ","(g)")))+
  scale_x_date(limits= as.Date(c("2016-01-01","2019-12-12"),
  labels=c("2016","2017","2018","2019")))+
```

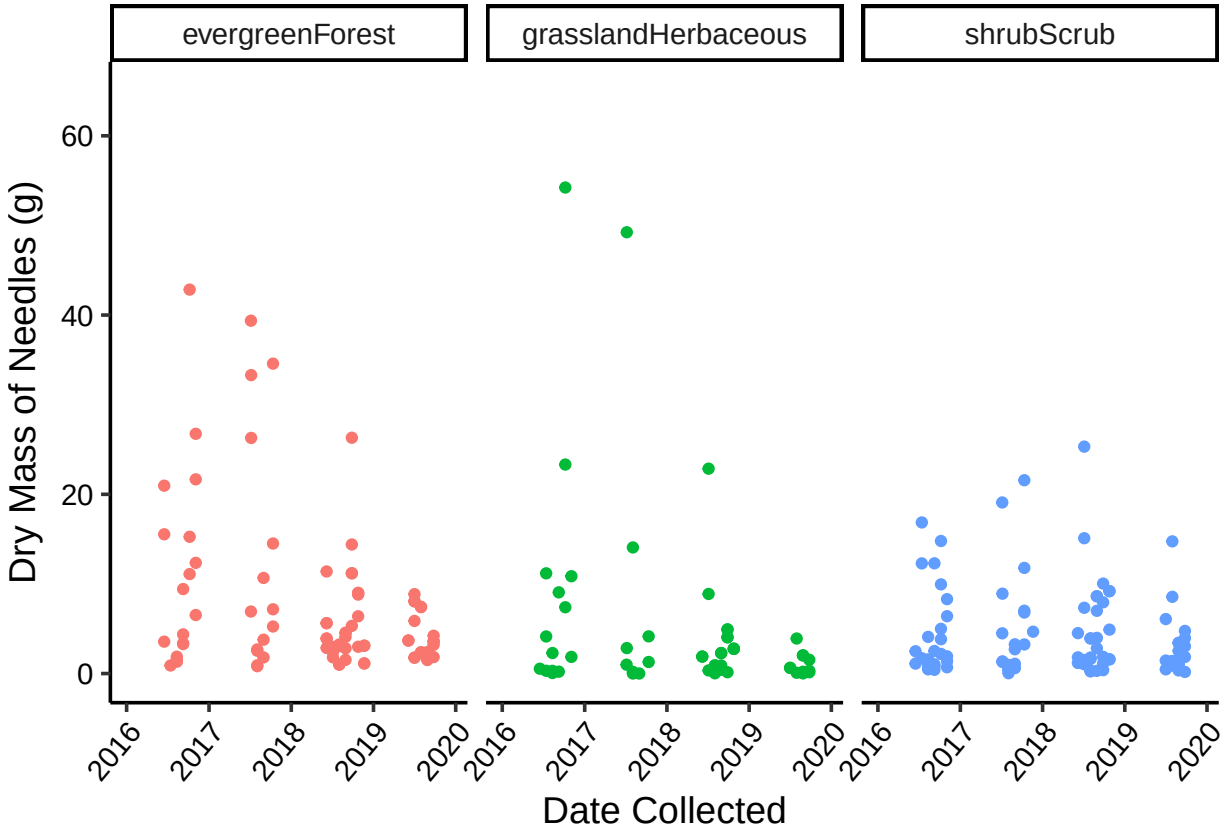
```
 ylim(0,65)+
 scale_color_discrete(name= "NLCD Classes")
print(NeedlesPlot)
```




```
# need to fix legend spelling and size!
# ("evergreenForest"="Evergreen Forest","grasslandHerbaceous"="Grassland Herbaceous","shrubScrub"="Shr

#7

Faceted.NeedlesPlot <-
  ggplot(subset(Litter, functionalGroup == "Needles"),
 aes(x = collectDate, y = dryMass, color = nlcdClass))+
 geom_point()+
 xlab("Date Collected")+
 ylab(expression(paste("Dry Mass of Needles ","(g)")))+
 scale_x_date(limits= as.Date(c("2016-01-01","2019-12-12"),
 labels=c("2016","2017","2018","2019")))+
 ylim(0,65)+
 facet_wrap(vars(nlcdClass), nrow = 1)+
 theme(axis.text.x = element_text(angle = 50,  hjust = 1))+
 theme(legend.position = "none")
print(Faceted.NeedlesPlot)
```




```
# I deleted the legend,
# because it was redundant,
# now that the classes were separated
```

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

> Answer: Plot 7 is more effective, because you can see the difference in dry mass of needles in each land class separately. Plot 6 combines land classes together. The layers on top of one another can take more time to draw conclusions from the data.