Assignment 5: Data Visualization

Jack Eynon

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/jackeynon/Courses/EnvDataAnalytics/Environmental_Data_Analytics_2020/Assignments"
```

```
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())
```

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

```
lake.chemistry.nutrients <- read.csv("~/Courses/EnvDataAnalytics/Environmental_Data_Analytics_2020/Data/
lake.nutrients <- read.csv("~/Courses/EnvDataAnalytics/Environmental_Data_Analytics_2020/Data/Processed/
litter <- read.csv("~/Courses/EnvDataAnalytics/Environmental_Data_Analytics_2020/Data/Processed/NEON_NIW

#2
class(lake.chemistry.nutrients$sampledate) #factor; must convert to date
```

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

```
class(lake.nutrients$sampledate) # also factor
```

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

```
str(litter) ## several date columns, all set as factors (weighdate, setdate, collectdate, date (ignorin
```

```
## 'data.frame':    1692 obs. of  39 variables:
##  $ uid.x                    : Factor w/ 1692 levels "00156623-3635-4674-884a-03ae92b321e2",..: 1101 4
##  $ namedLocation.x          : Factor w/ 12 levels "NIWO_040.basePlot.ltr",..: 9 8 9 11 7 7 4 4 4 4 .
##  $ domainID.x               : Factor w/ 1 level "D13": 1 1 1 1 1 1 1 1 1 1 ...
##  $ siteID                   : Factor w/ 1 level "NIWO": 1 1 1 1 1 1 1 1 1 1 ...
##  $ plotID                   : Factor w/ 12 levels "NIWO_040","NIWO_041",..: 9 8 9 11 7 7 4 4 4 4 ...
##  $ trapID                   : Factor w/ 15 levels "NIWO_040_139",..: 11 10 11 13 9 9 5 5 5 5 ...
##  $ weighDate                : Factor w/ 25 levels "2013-09-26","2016-06-22",..: 2 2 2 2 2 2 2 2 2 2
##  $ setDate                  : Factor w/ 24 levels "2015-09-29","2016-06-16",..: 1 1 1 1 1 1 1 1 1 1
##  $ collectDate              : Factor w/ 24 levels "2016-06-16","2016-07-14",..: 1 1 1 1 1 1 1 1 1 1
##  $ ovenStartDate            : Factor w/ 28 levels "2016-06-16T23:20Z",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ovenEndDate              : Factor w/ 39 levels "2016-06-22T22:25Z",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ fieldSampleID            : Factor w/ 214 levels "NEON.LTR.NIWO040139.20161103",..: 141 123 141 178
##  $ massSampleID             : Factor w/ 1579 levels "NEON.LTR.NIWO040139.20161103.LVS",..: 1040 906 1
##  $ samplingProtocolVersion.x: Factor w/ 2 levels "NEON.DOC.001710vE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ functionalGroup          : Factor w/ 8 levels "Flowers","Leaves",..: 6 5 8 6 4 2 2 6 7 8 ...
##  $ dryMass                  : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass                : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 1 1 1 1 ...
##  $ remarks.x                : Factor w/ 8 levels "","actually 0.0039",..: 3 1 1 3 1 3 3 3 1 1 ...
##  $ measuredBy               : Factor w/ 13 levels "adignan@battelleecology.org",..: 4 4 4 4 4 4 4 4 4
##  $ uid.y                    : Factor w/ 15 levels "01551580-266c-4b37-b9eb-00f1f3078f02",..: 15 2 15
##  $ namedLocation.y          : Factor w/ 12 levels "NIWO_040.basePlot.ltr",..: 9 8 9 11 7 7 4 4 4 4 .
##  $ domainID.y               : Factor w/ 1 level "D13": 1 1 1 1 1 1 1 1 1 1 ...
##  $ subplotID                : int  31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude          : num  40.1 40 40.1 40 40 ...
##  $ decimalLongitude         : num  -106 -106 -106 -106 -106 ...
##  $ elevation                : num  3477 3413 3477 3373 3446 ...
##  $ nlcdClass                : Factor w/ 3 levels "evergreenForest",..: 3 1 3 1 3 3 2 2 2 2 ...
##  $ plotType                 : Factor w/ 1 level "tower": 1 1 1 1 1 1 1 1 1 1 ...
##  $ geodeticDatum            : Factor w/ 1 level "WGS84": 1 1 1 1 1 1 1 1 1 1 ...
##  $ coordinateUncertainty    : num  10.1 10.2 10.1 10.1 10.1 10.1 10.1 10.1 10.1 10.1 ...
##  $ elevationUncertainty     : num  0.1 0.4 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ date                     : Factor w/ 1 level "2015-07-15": 1 1 1 1 1 1 1 1 1 1 ...
##  $ samplingProtocolVersion.y: Factor w/ 1 level "NEON.DOC.001710vB": 1 1 1 1 1 1 1 1 1 1 ...
##  $ targetTaxaPresent        : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ trapType                 : Factor w/ 2 levels "Elevated","Ground": 1 1 1 1 1 1 1 1 1 1 ...
##  $ trapPlacement            : Factor w/ 1 level "Targeted": 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ trapSize                 : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ remarks.y                : Factor w/ 1 level "measuredBy, recordedBy, and setDate not reported": 
##  $ recordedBy               : Factor w/ 1 level "tdelpriore@neoninc.org": 1 1 1 1 1 1 1 1 1 1 ...
```

```
lake.chemistry.nutrients$sampledate <- as.Date(lake.chemistry.nutrients$sampledate, format = "%Y-%m-%d")
lake.nutrients$sampledate <- as.Date(lake.nutrients$sampledate, format = "%Y-%m-%d")
litter$weighDate <- as.Date(litter$weighDate, format = "%Y-%m-%d")
litter$setDate <- as.Date(litter$setDate, format = "%Y-%m-%d")
litter$collectDate <- as.Date(litter$collectDate, format = "%Y-%m-%d")
litter$date <- as.Date(litter$date, format = "%Y-%m-%d")
```

Define your theme

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

```
theme_eynon <- theme_bw(base_size = 12, base_line_size = 1) +
  theme(axis.ticks = element_line(color = "black"), axis.text = element_text(color = "darkmagenta"), axi

theme_set(theme_eynon)
```

Create graphs

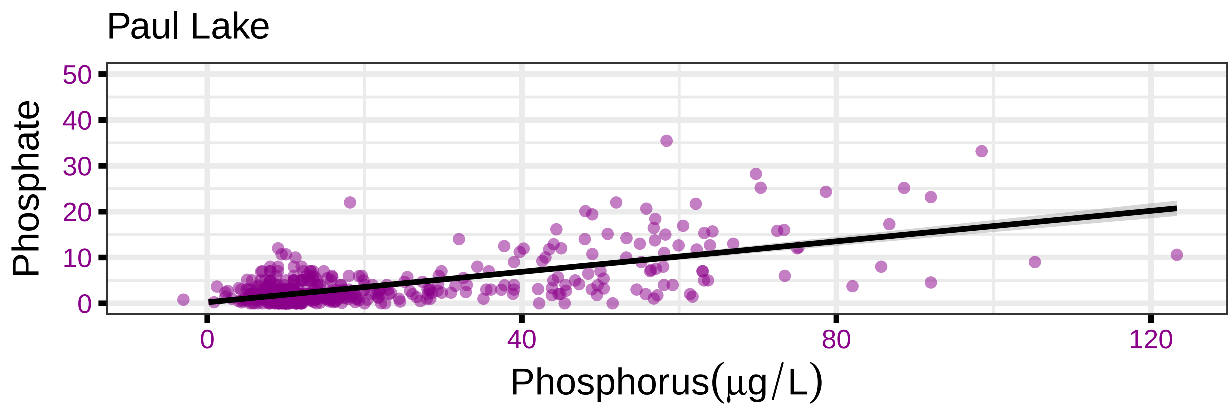
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

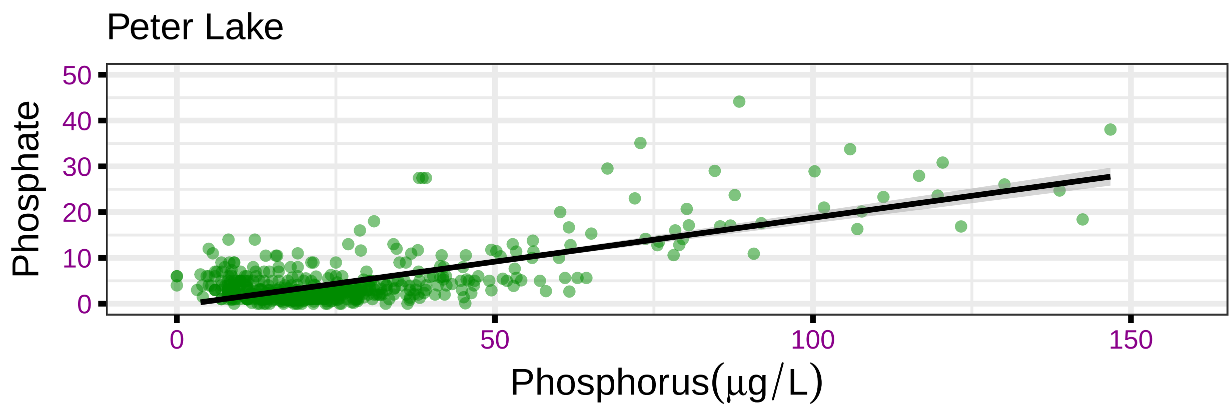
4. [NTL-LTER] Plot total phosphorus by phosphate, with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values.

```
## phosphate = po4, phosphorus = tp_ug
paul.lake.p04xtp <- ggplot(subset(lake.chemistry.nutrients, lakename=="Paul Lake"), aes(x=tp_ug, y=po4))
  labs(x=expression(Phosphorus (mu*g/L)), y = expression(Phosphate)) +
  theme(axis.title.x = element_text(face="bold", size = 14),
        axis.title.y = element_text(face="bold", size=14)) +
  geom_smooth(method = lm, color = "black") +
  ylim(0,50) + ggtitle("Paul Lake")

peter.lake.p04xtp <- ggplot(subset(lake.chemistry.nutrients, lakename=="Peter Lake"), aes(x=tp_ug, y=po4
  labs(x=expression(Phosphorus (mu*g/L)), y = expression(Phosphate)) +
  theme(axis.title.x = element_text(face="bold", size = 14),
        axis.title.y = element_text(face="bold", size=14)) +
  geom_smooth(method = lm, color = "black") +
  ylim(0,50) + ggtitle("Peter Lake")

question4 <- plot_grid(paul.lake.p04xtp, peter.lake.p04xtp, nrow=2)
```

```
## Warning: Removed 10524 rows containing non-finite values (stat_smooth).

## Warning: Removed 10524 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_smooth).

## Warning: Removed 11423 rows containing non-finite values (stat_smooth).

## Warning: Removed 11423 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_smooth).
```

```
print(question4)
```



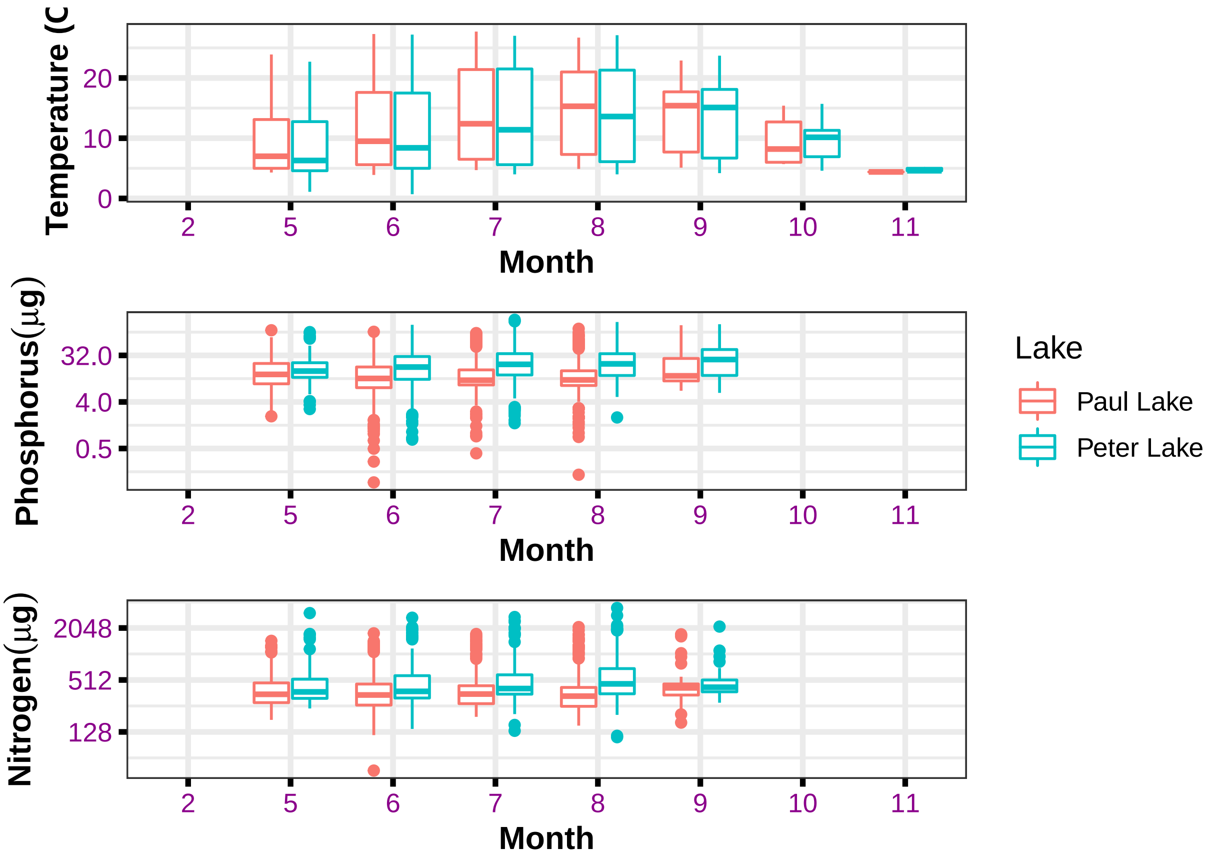


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
temp.box <- ggplot(data = lake.chemistry.nutrients, aes(x=as.factor(month), y=temperature_C, color = la
  geom_boxplot() +
  labs(x="Month", y="Temperature (C)") +
  theme(legend.position = "none")
tp.box <- ggplot(data = lake.chemistry.nutrients, aes(x=as.factor(month), y=tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(x="Month", y=expression(bold(paste(Phosphorus(mu*g)))), color = "Lake") +
   scale_y_continuous(trans = "log2") ## applied log scale to y-axis
tn.box <- ggplot(data = lake.chemistry.nutrients, aes(x=as.factor(month), y=tn_ug, color = lakename)) +
  geom_boxplot() +
  labs(x="Month", y=expression(bold(paste(Nitrogen(mu*g))))) +
  theme(legend.position = "none") +
   scale_y_continuous(trans = "log2") ## applied log scale to y-axis

question5 <- plot_grid(temp.box, tp.box, tn.box, ncol = 1, align = 'v', axis = "lr")
```

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

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 20773 rows containing non-finite values (stat_boxplot).

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

```r
print(question5)
```



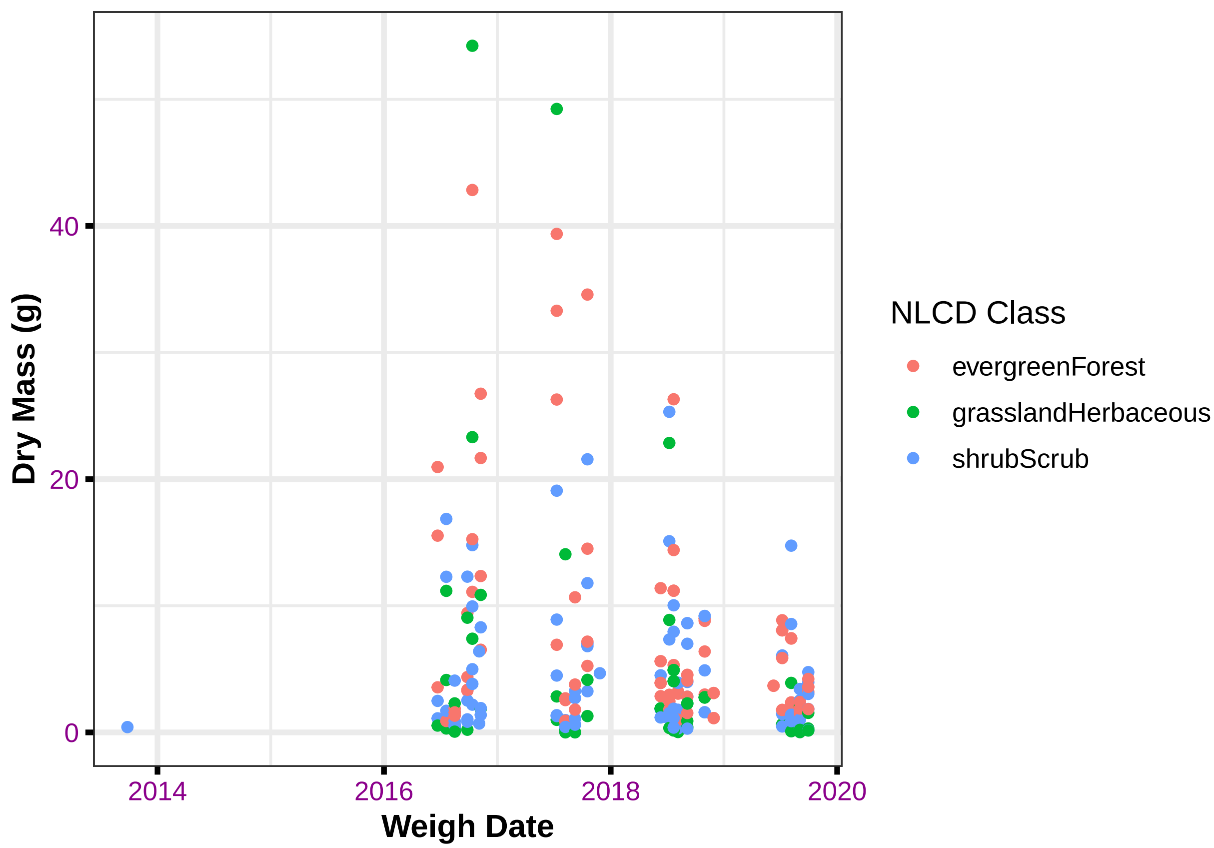
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature is predictably higher in the summer months. It is comparatively harder to discern a trend in nutrient levels over the seasons, although it seems both phosphorus and nitrogen concentrations may be slightly higher in the late summer/early fall. Peter lake seems to have higher nutrients loads in than Paul Lake in all seasons.

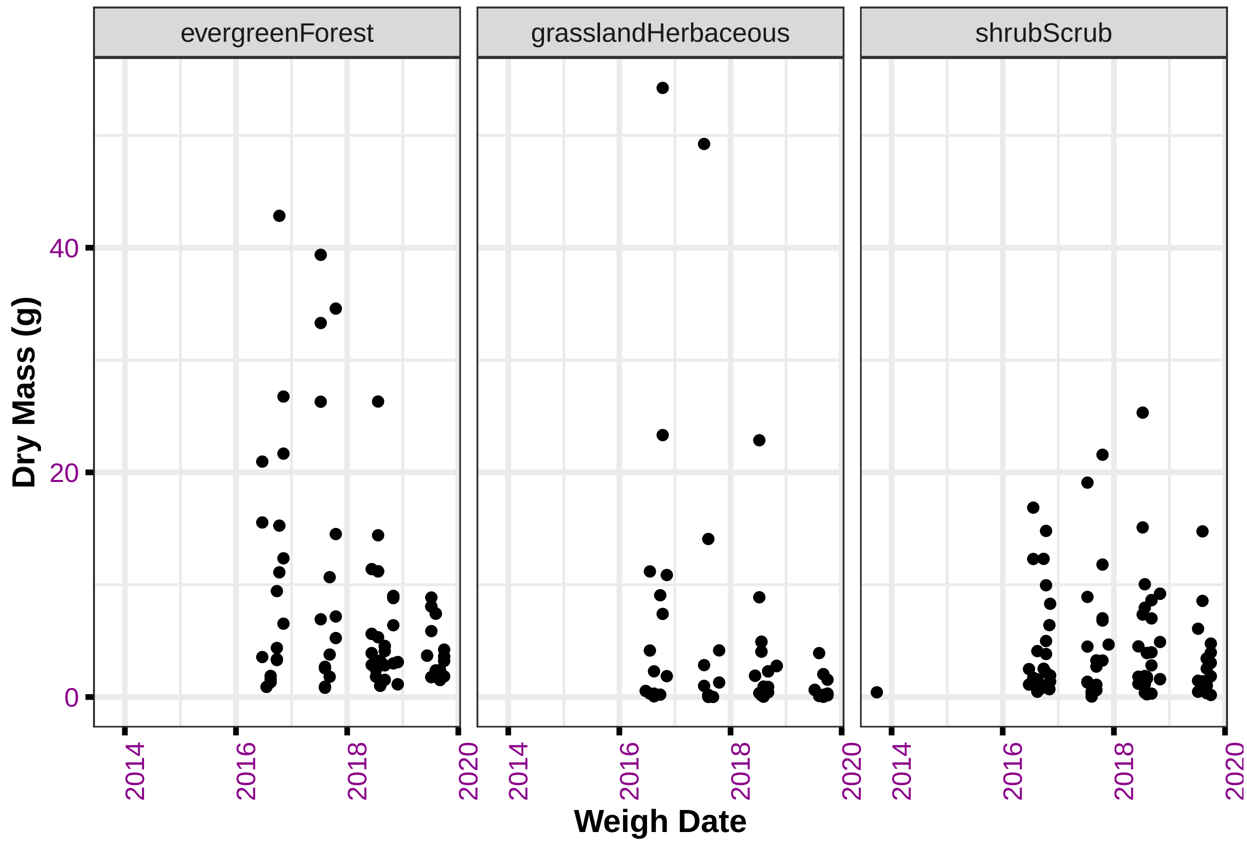
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```r
mass.plot.color <- ggplot(subset(litter, functionalGroup == "Needles"), aes(y=dryMass, x=weighDate, colo
  geom_point() +
  labs(x = "Weigh Date", y = "Dry Mass (g)", color = "NLCD Class")
print(mass.plot.color)
```




```
mass.plot.faceted <- ggplot(subset(litter, functionalGroup == "Needles"), aes(y=dryMass, x=weighDate)) +
  geom_point() +
  facet_wrap(vars(nlcdClass)) +
  labs(x = "Weigh Date", y = "Dry Mass (g)", color = "NLCD Class") +
  theme(axis.text.x = element_text(angle = 90))
print(mass.plot.faceted)
```



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

> Answer: I think 7, the faceted version, is more effective because the first plot is quite cluttered. With so many points plotted on one graph, it is difficult to get an idea of trends for each NLCD class. Although we might expect to be able to compare NLCD classes more easily when plotting them by color, this advantage is undermined by how busy the plot.