Assignment 5: Data Visualization

Susanna Jenkins

Spring 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. 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
library(tidyverse); library(lubridate); library(here); library(cowplot);
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.0
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## Loading required package: timechange
##
```

```
##
## Attaching package: 'lubridate'
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
##
## here() starts at /home/guest/EDA-Spring2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here()
```

```
## [1] "/home/guest/EDA-Spring2023"
```

```
PeterPaul.chem.nutrients <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"), string

NiwotRidge.litter <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"), stringsAsFactors = TRUE

#2

PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
NiwotRidge.litter$collectDate <- ymd(NiwotRidge.litter$collectDate)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```
#3

library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
susanna_theme <- theme_base() +
  theme(
    plot.title = element_text(
      color='purple'
    ),
    plot.background = element_rect(
      color = 'blue',
      fill = 'grey'
    ),
    legend.position = 'right'
    )
```

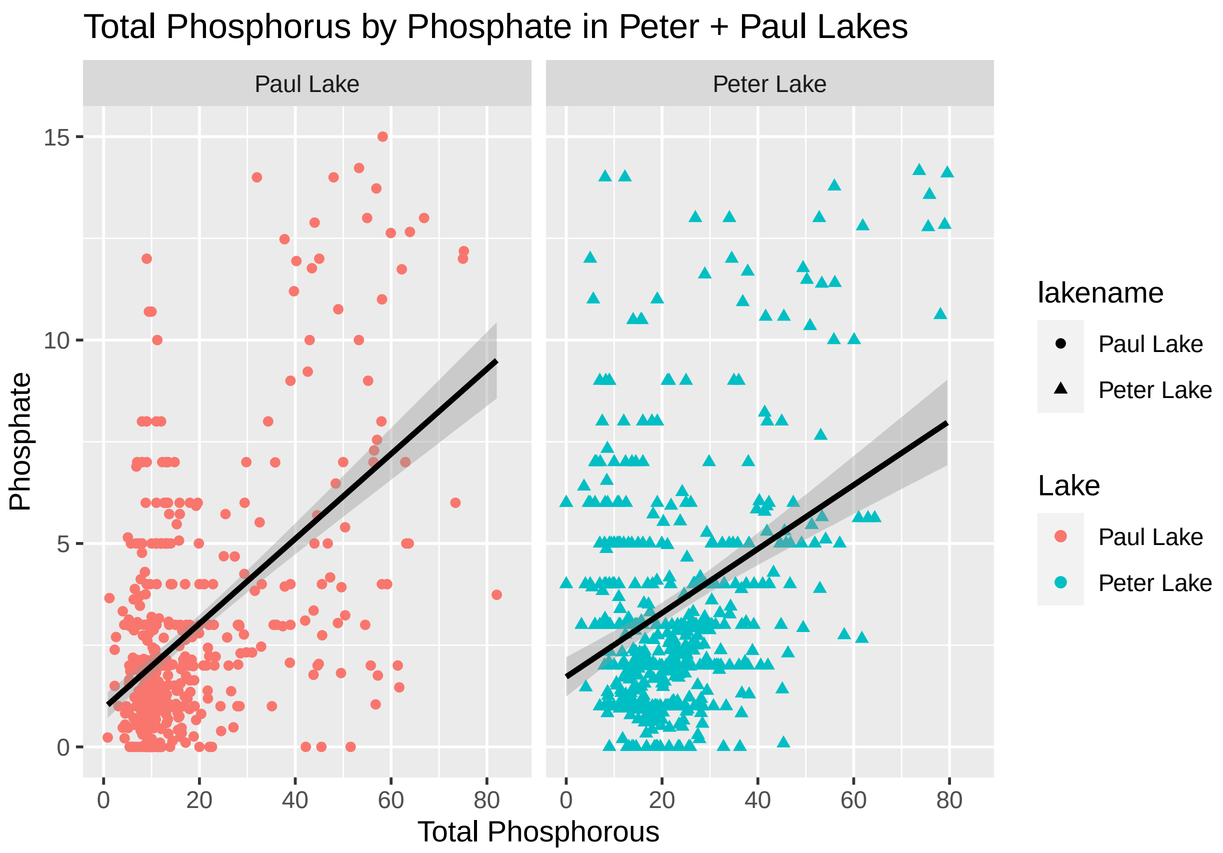
Create graphs

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 (`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/or `ylim()`).

```
#4

Plot4 <-
  ggplot(PeterPaul.chem.nutrients, aes(
    x = tp_ug,
    y = po4,
    color = lakename,
    shape = lakename)) +
  geom_point() +
  facet_wrap(vars(lakename)) +
  geom_smooth(method=lm, color = "black") +
  xlim(0, 85) +
  ylim(0, 15) +
  labs(x="Total Phosphorous", y="Phosphate", title = "Total Phosphorus by Phosphate in Peter + Paul Lake
print(Plot4)
```

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

## Warning: Removed 22010 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 22010 rows containing missing values (`geom_point()`).
```

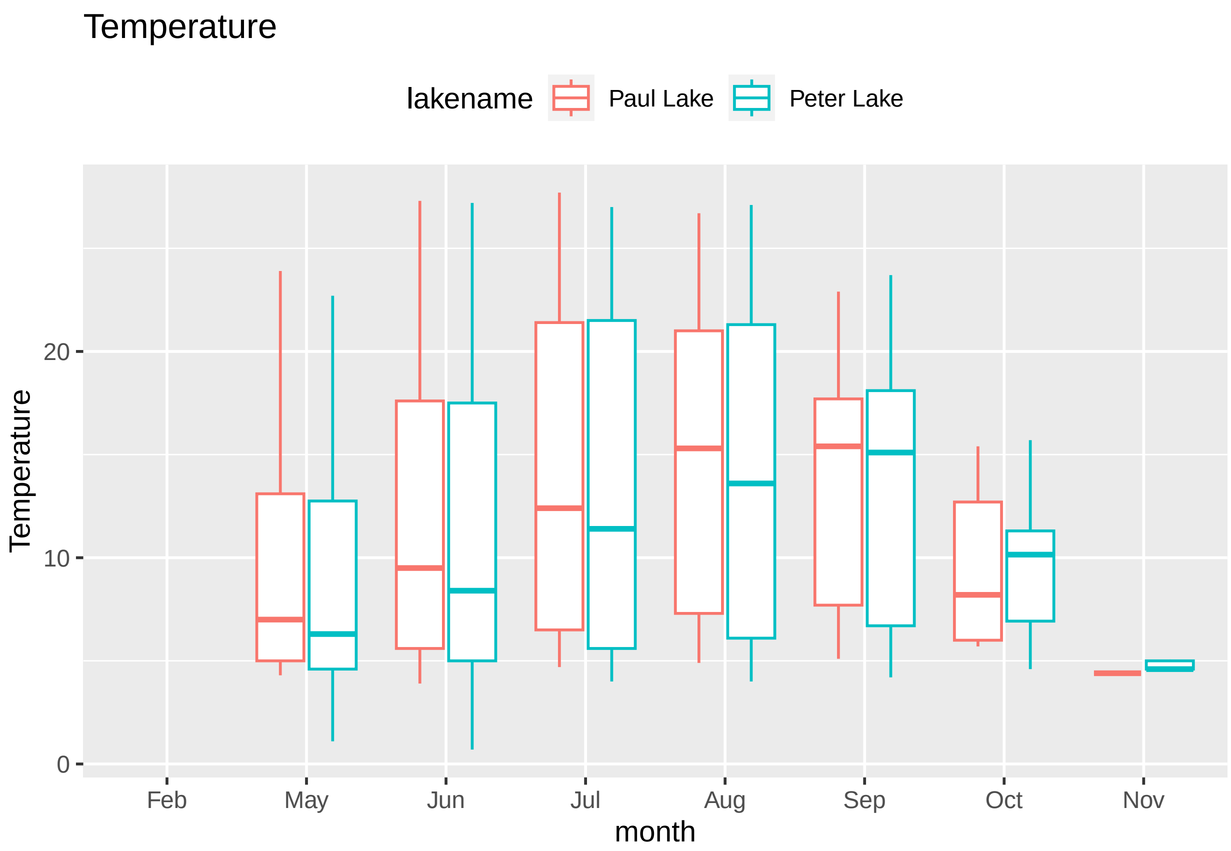


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: R has a build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

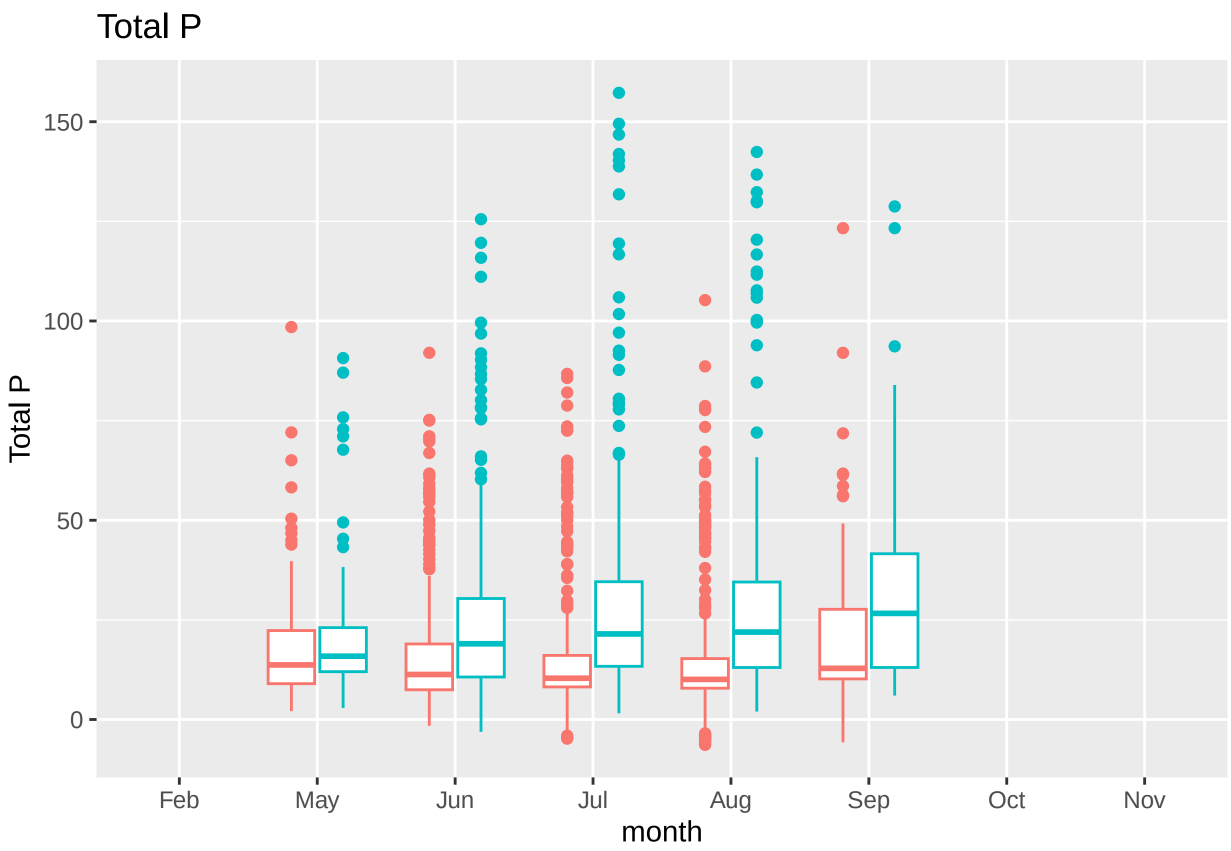
```
#5

#(a)

TemperaturePlot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = factor(month, levels = 1:12, labels=month.abb), y = temperatu
  geom_boxplot(aes(color = lakename)) +
  labs(
    x = "month",
    y = 'Temperature',
    title = 'Temperature',
    color = 'lakename'
  ) + theme(legend.position = "top")
print(TemperaturePlot)
```

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



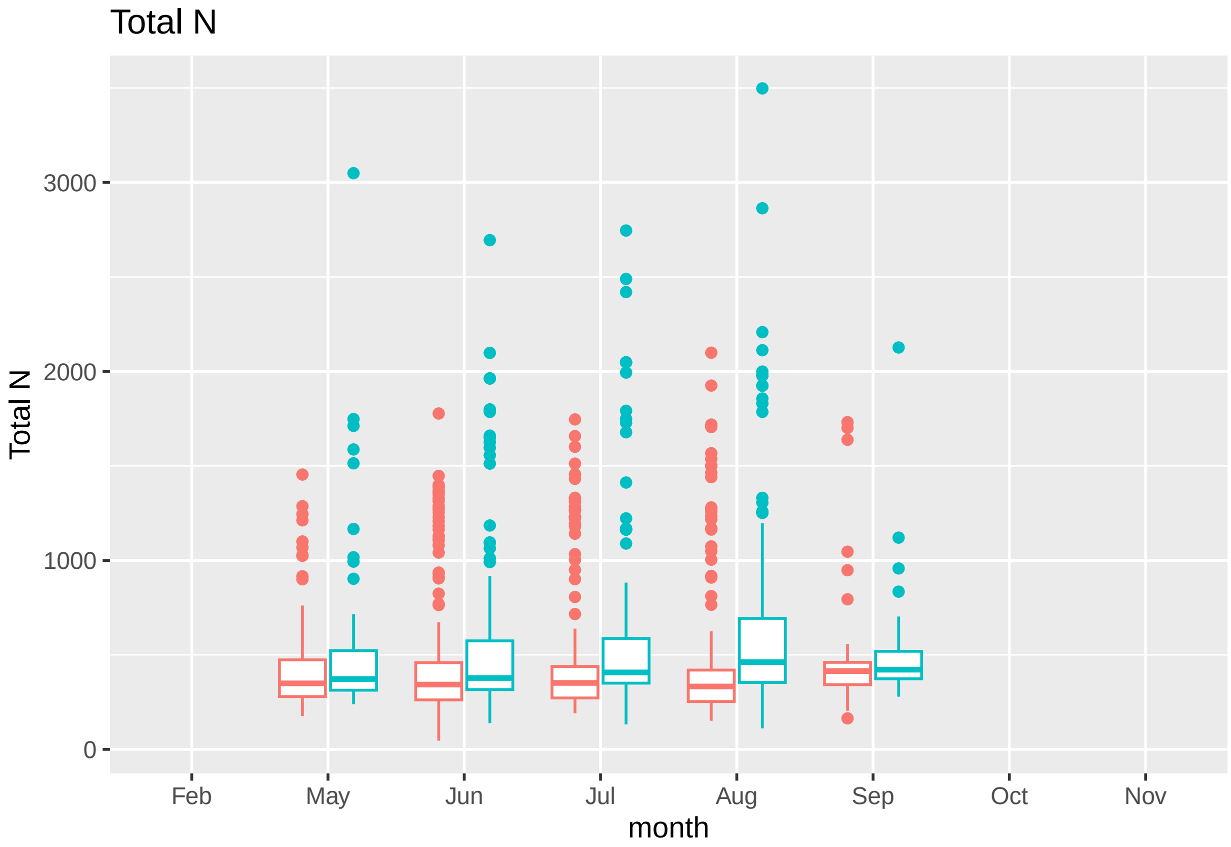
```
#(b)

TPPlot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = factor(month, levels = 1:12, labels=month.abb), y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(
    x = "month",
    y = 'Total P',
    title = 'Total P',
    color = 'lakename') + theme(legend.position =  "none")
print(TPPlot)
```

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



```
#(c)

TNPlot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = factor(month, levels = 1:12, labels=month.abb), y = tn_ug))
  geom_boxplot(aes(color = lakename)) +
  labs(
    x = "month",
    y = 'Total N',
    title = 'Total N',
    color = 'lakename'
  )  + theme(legend.position =  "none")
print(TNPlot)
```

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



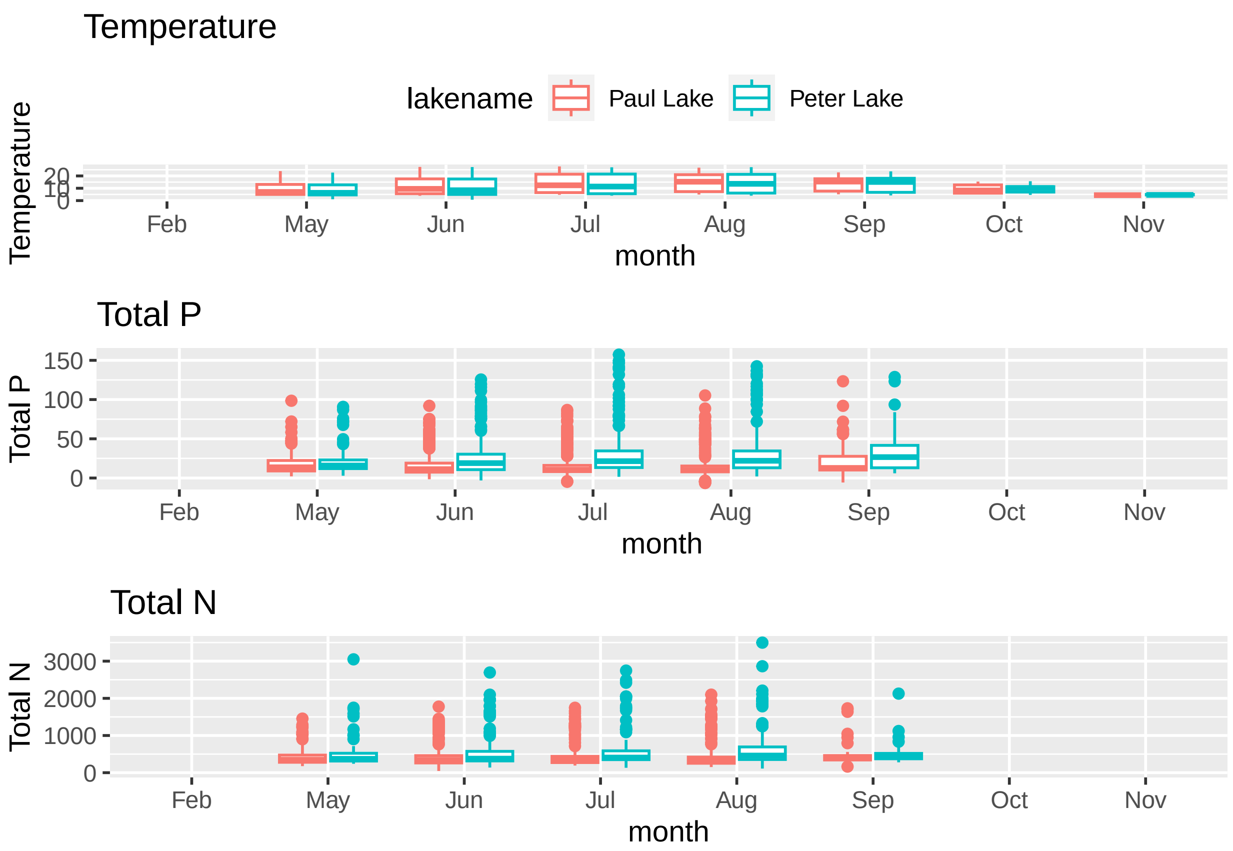
```
library(cowplot)

plot_grid(TemperaturePlot,TPPlot, TNPlot, nrow = 3, align = 'h')
```

```
## 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()‘).
```

```
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```



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

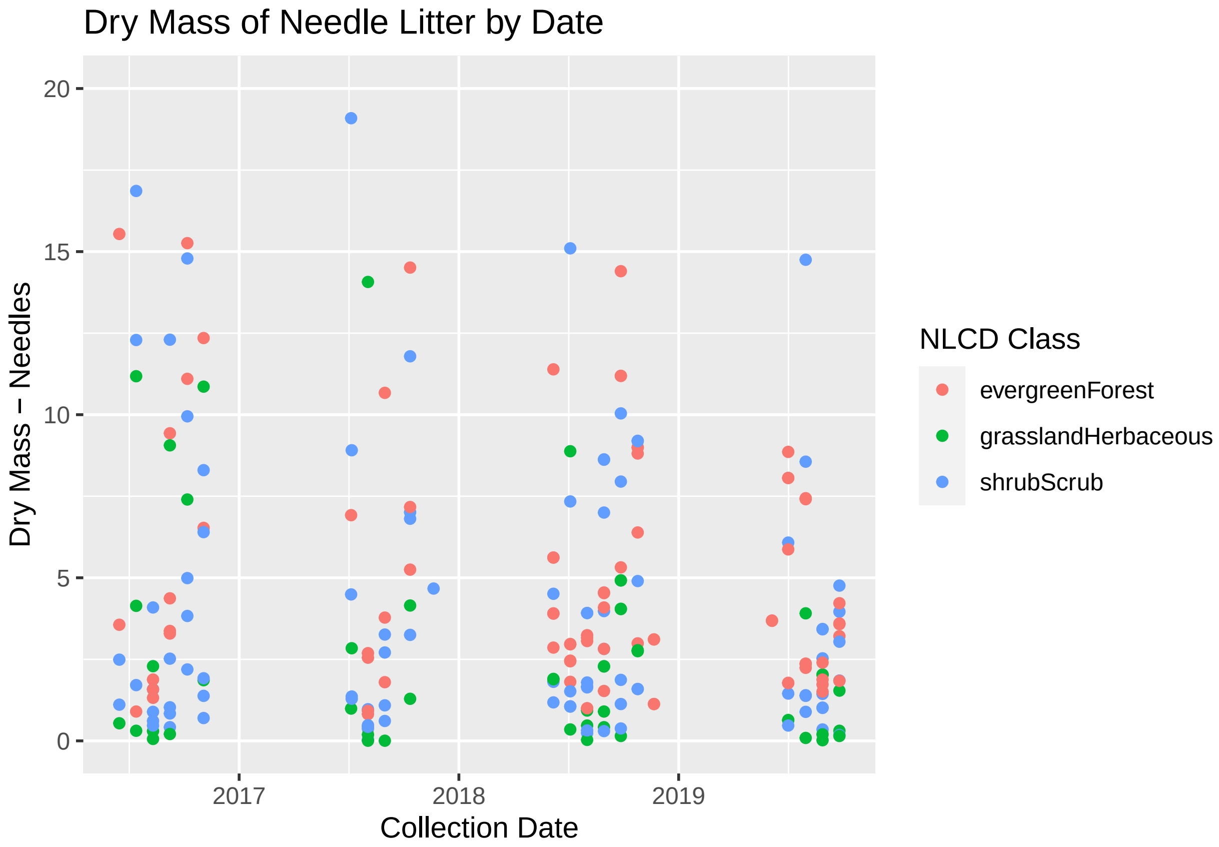
> Answer:

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.

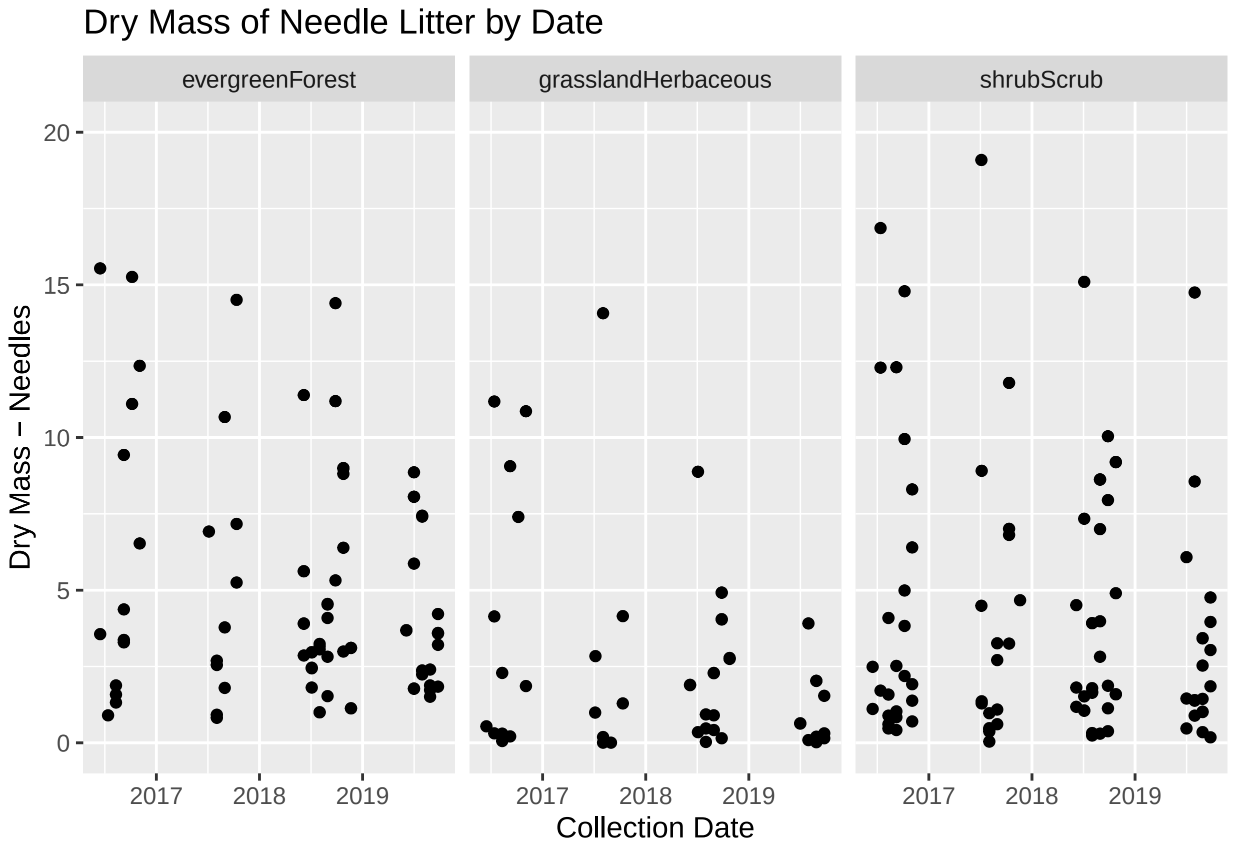
```
#6

PlotLitter <-
  ggplot(subset(NiwotRidge.litter, functionalGroup == "Needles"), aes(
    x = collectDate,
    y = dryMass,
    color = nlcdClass)) +
  geom_point() +
  labs(
    x = "Collection Date",
    y = 'Dry Mass - Needles',
    title = 'Dry Mass of Needle Litter by Date',
    color='NLCD Class') +
  ylim(0, 20)
print(PlotLitter)
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```



```
#7

PlotLitter_Facets <-
  ggplot(subset(NiwotRidge.litter, functionalGroup == "Needles"), aes(
    x = collectDate,
    y = dryMass)) +
  geom_point() +
  labs(
    x = "Collection Date",
    y = 'Dry Mass - Needles',
    title = 'Dry Mass of Needle Litter by Date',
    color='NLCD Class') +
  ylim(0, 20) +
  facet_wrap(vars(nlcdClass))
print(PlotLitter_Facets)
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```



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

> Answer: I personally find the first plot (6) more effective because it is allows for more visually obvious benchmarking of the litter collection by year between the 3 different land cover classes. Color helps me significantly.