Assignment 5: Data Visualization

Shivani Kuckreja

#install.packages("formatR")

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()
```

```
## [1] "/home/guest/R/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     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
```

```
PeterPaul_chem_nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")

NW_litter <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(PeterPaul_chem_nutrients$sampledate) #originally a character
```

```
## [1] "character"
```

```
# Change date to date object
PeterPaul_chem_nutrients$sampledate <- ymd(PeterPaul_chem_nutrients$sampledate)

class(NW_litter$collectDate) #originally a character
```

```
## [1] "character"
```

```
# Change date to date object
NW_litter$collectDate <- ymd(NW_litter$collectDate)
```

Define your theme

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

```
#3

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "blue"),
legend.position = "top") #alternative: legend.position + legend.justification

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 (`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()`).

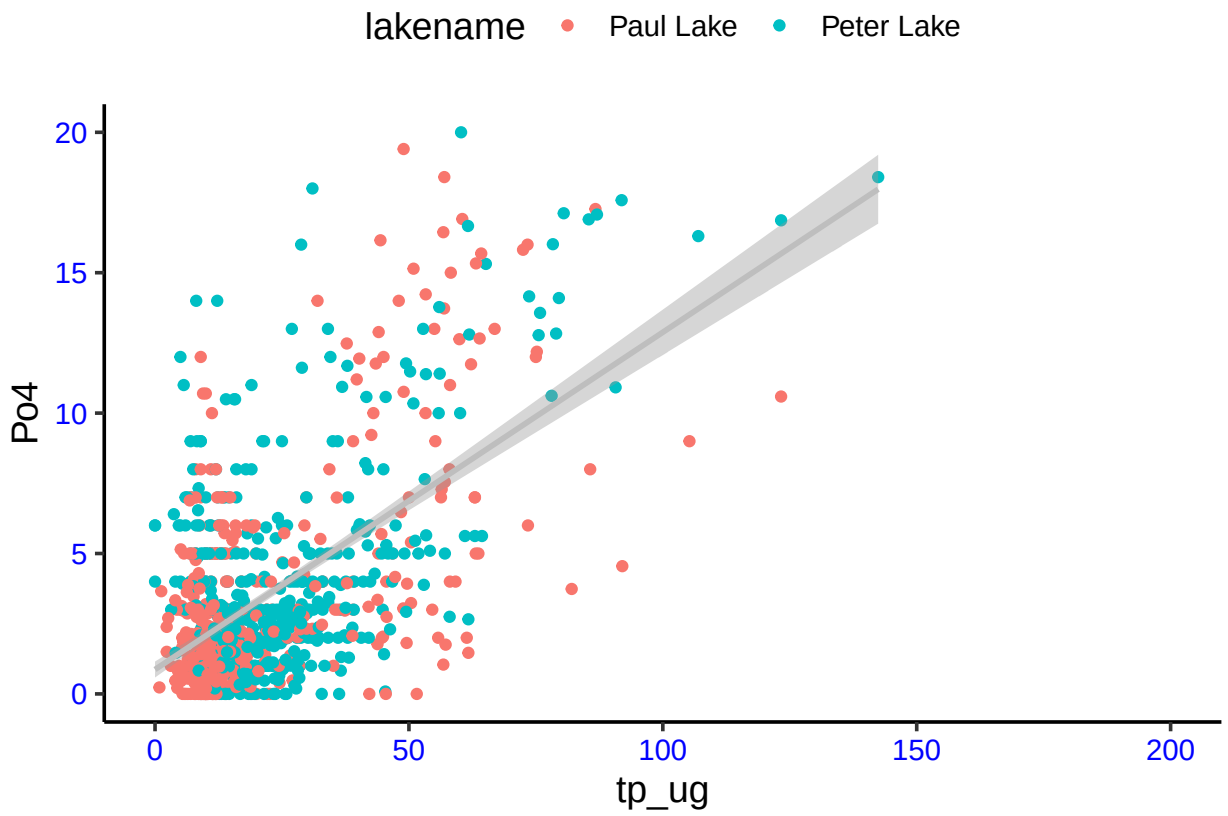
```
#4
NTLLTER_Plot <- ggplot(PeterPaul_chem_nutrients, aes(x = tp_ug, y = po4,
```

```
                                                    color=lakename)) +
  geom_point() +
  geom_smooth(method = lm, color = "grey") +
  ylab('Po4') +
  xlim(0, 200) +
  ylim(0, 20)
print(NTLLTER_Plot)
```

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

```
## Warning: Removed 21981 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21981 rows containing missing values (geom_point).
```



5. [NTL-LTER] Make three separate boxplots of (a) temperature,

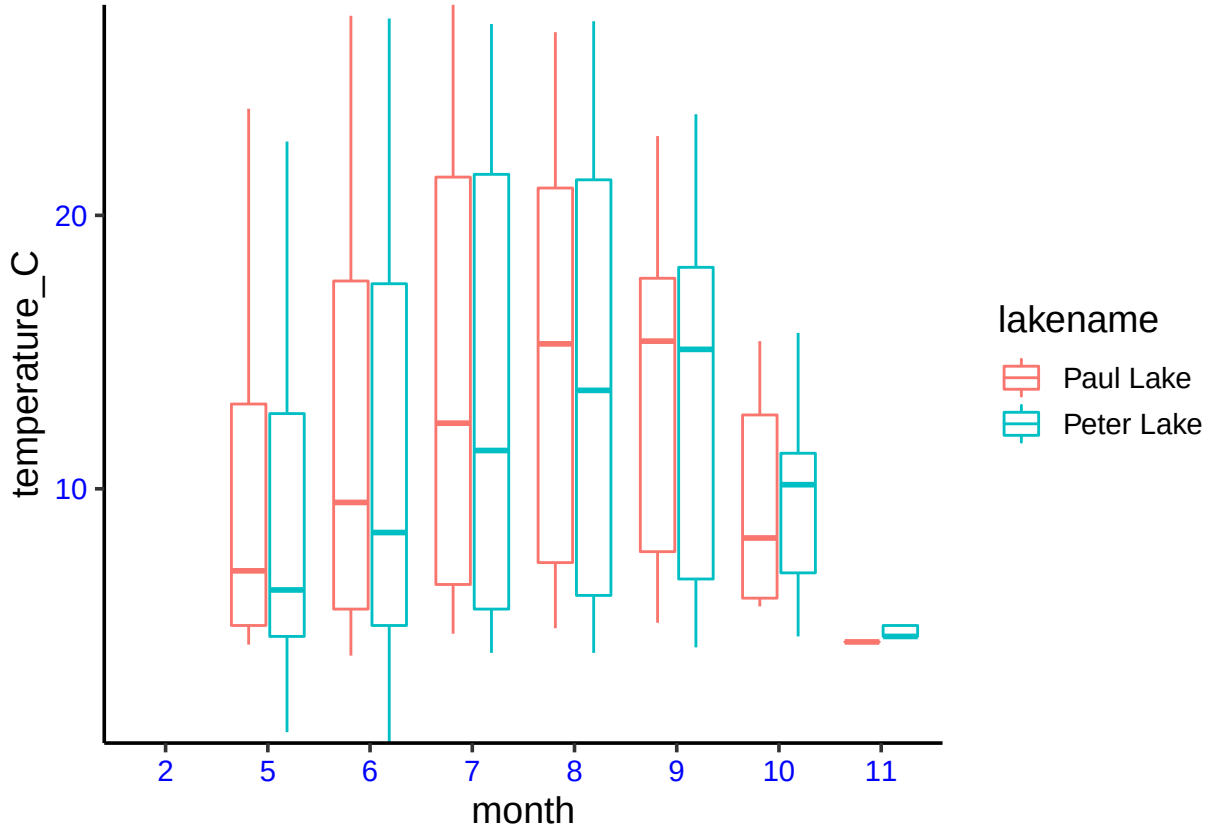
(b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
#5
PeterPaul_chem_nutrients$month <- as.factor(PeterPaul_chem_nutrients$month)

NTLLTER_BoxPlot_Temp <- ggplot(PeterPaul_chem_nutrients,
                               aes(x = month, y = temperature_C,
                                   color=lakename)) +
  geom_boxplot() +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position = "right")
```

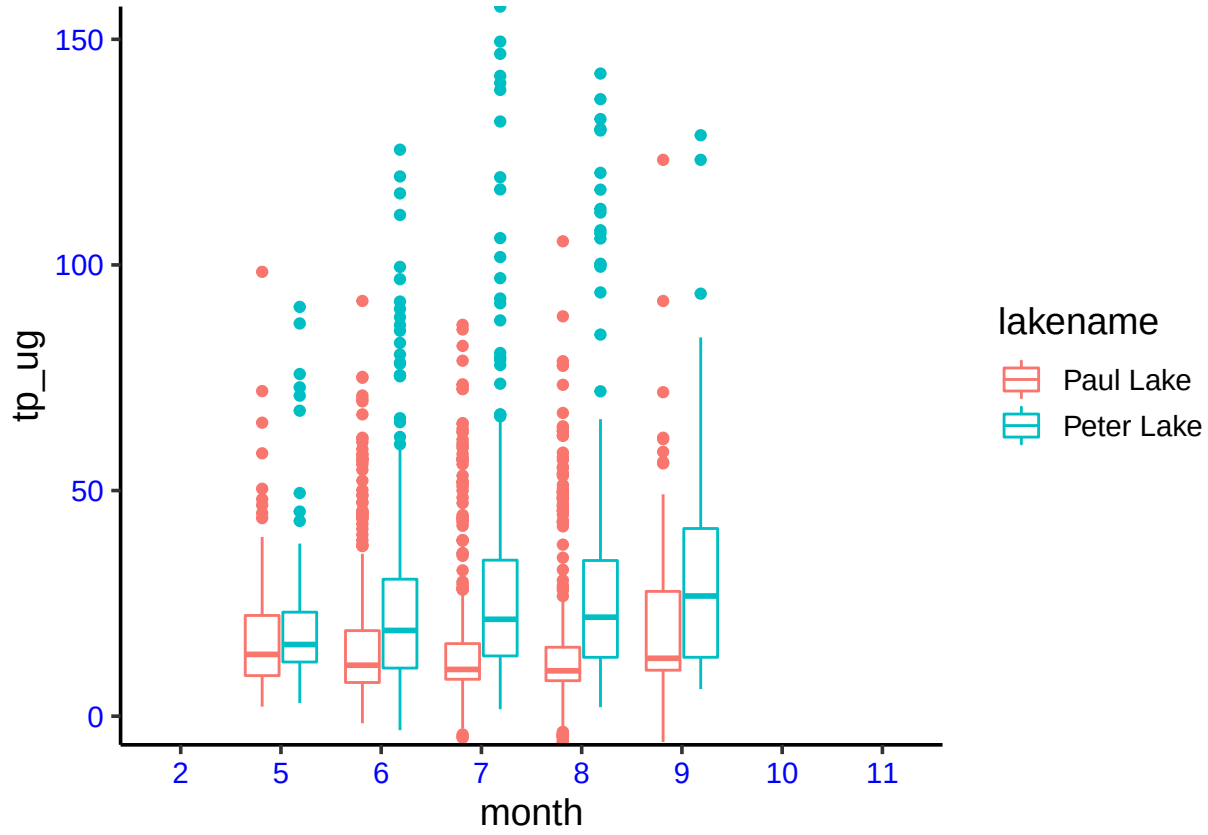
```
print(NTLLTER_BoxPlot_Temp)
```

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



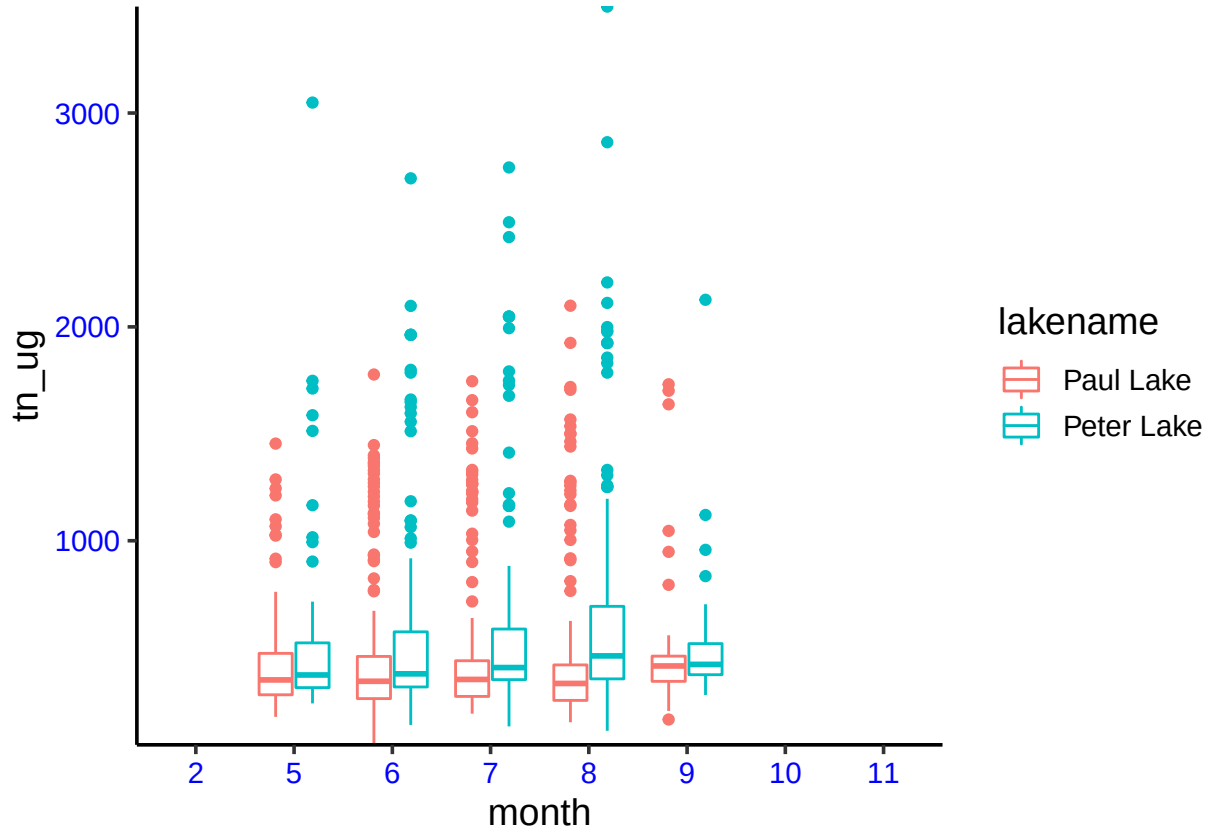

```
NTLLTER_BoxPlot_TP <- ggplot(PeterPaul_chem_nutrients, aes(x = month,
                                                           y = tp_ug,
                                                           color=lakename)) +
  geom_boxplot() +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position = "right")
print(NTLLTER_BoxPlot_TP)
```

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




```
NTLLTER_BoxPlot_TN <- ggplot(PeterPaul_chem_nutrients,
                             aes(x = month, y = tn_ug, color=lakename)) +
  geom_boxplot() +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position = "right")
print(NTLLTER_BoxPlot_TN)
```

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

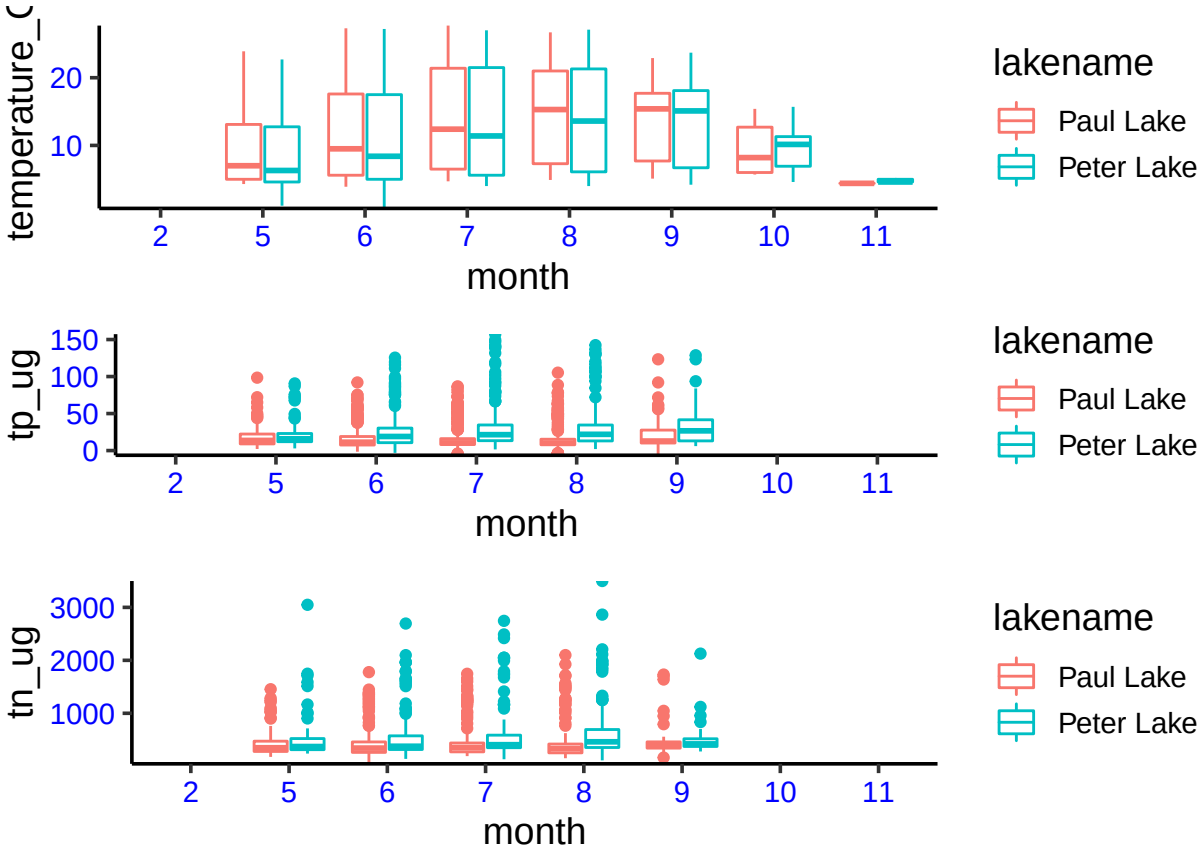



```
#install.packages("cowplot")
library(cowplot)
plot_grid(NTLLTER_BoxPlot_Temp, NTLLTER_BoxPlot_TP,
          NTLLTER_BoxPlot_TN, nrow = 3, align = 'h', rel_heights = c(1.25, 1))
```

```
## 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).
```



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

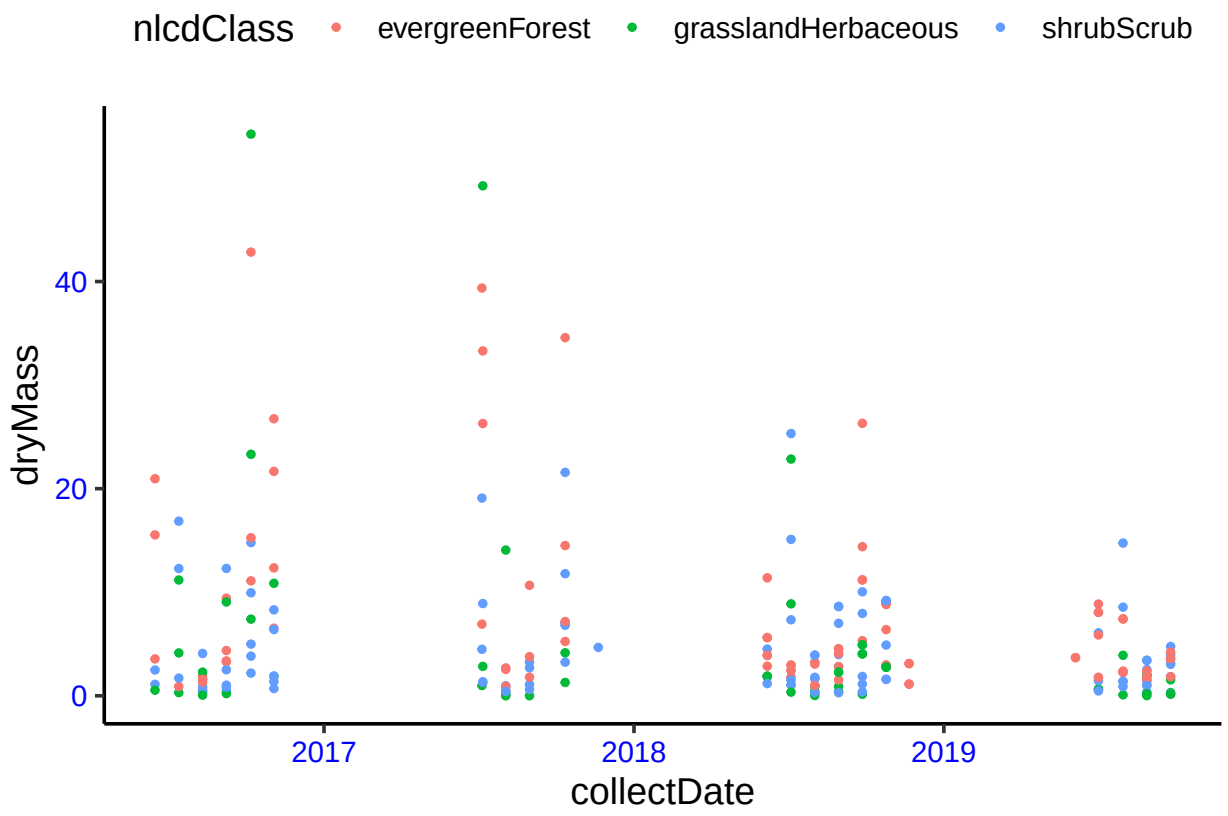
> Answer: tp_ug: There are a lot of outliers in this dataset, especially compared to the Temperature plots. There is low variability in the data for Paul Lake. tn_ug: There are a lot of outliers in this dataset, and Peter Lake seems to have significantly more outliers (of higher value) than does Paul Lake. temperature: Temperature over seasons and between lakes appears to vary very little.

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.

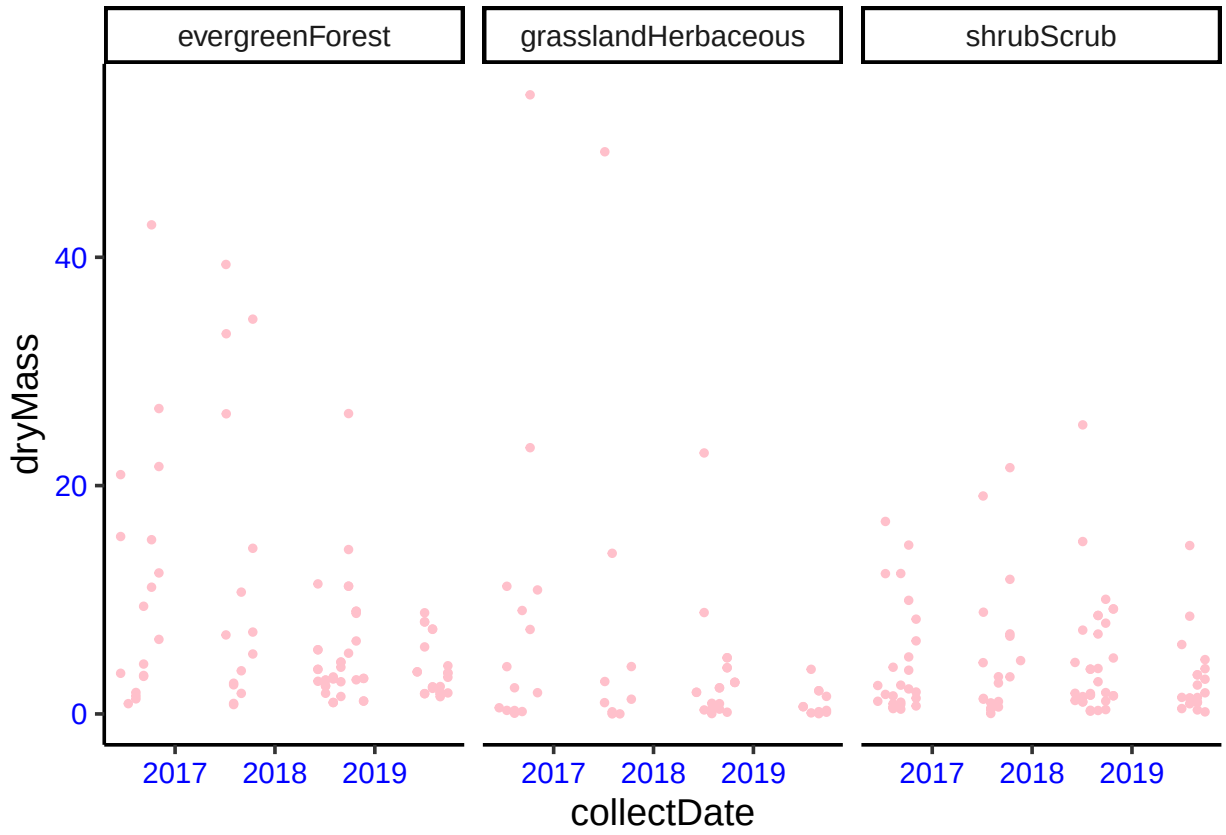
```
#6
NW_litter_filter <- NW_litter %>%
  filter(functionalGroup == "Needles")

ggplot(filter(NW_litter_filter)) +
  geom_point(aes(x = collectDate, y = dryMass,color=nlcdClass), size=1)
```




```
#7
NW_litter_filter <- NW_litter %>%
  filter(functionalGroup == "Needles")

ggplot(filter(NW_litter_filter)) +
  geom_point(aes(x = collectDate, y = dryMass), color="pink", size=1) +
facet_wrap('nlcdClass')
```



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

Answer: Plot 6 is more effective because it allows easier comparison across NLCD classes. Plot 7 only allows easier comparison by year within each NLCD class.