Assignment 5: Data Visualization

Sam Vanasse

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] "/Users/samvanasse/Desktop/ENV872-R/Environmental_Data_Analytics_2022/Assignments"
```

```
library(tidyverse)
```

```
## -- 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)

PeterPaul.chem.nutrients <-
  read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
NiwotRidge.litter <-
```

```
  read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
PeterPaul.chem.nutrients$sampledate <- as.Date(
  PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d")
NiwotRidge.litter$collectDate <- as.Date(
  NiwotRidge.litter$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

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

theme_set(def.theme)
```
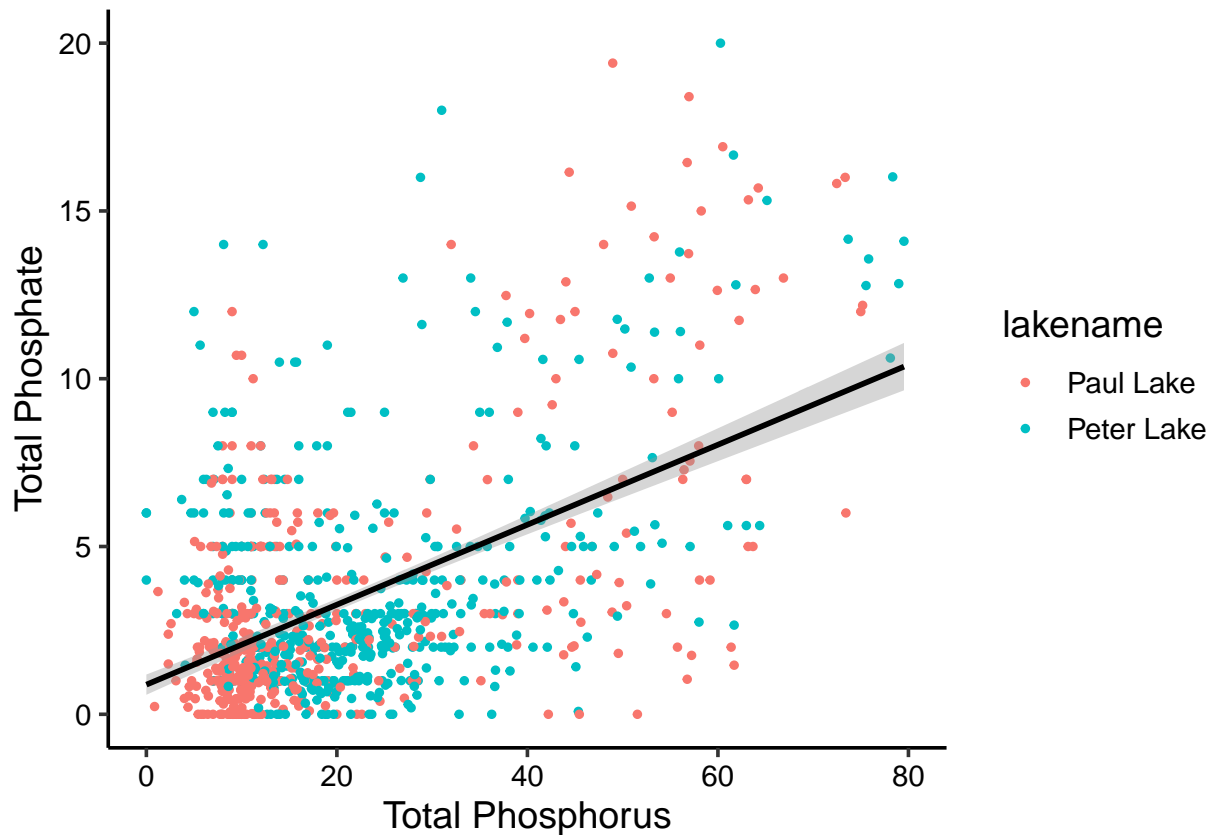
Create graphs

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
PeterPaulPlot.phosphorus.phosphate <-
  ggplot(PeterPaul.chem.nutrients, aes(x=tp_ug, y=po4, color = lakename)) +
  geom_point(size =1) +
  geom_smooth(method = lm, color = "black") +
  ylab("Total Phosphate") +
  xlab("Total Phosphorus") +
  xlim(0, 80) +
  ylim(0, 20)
print(PeterPaulPlot.phosphorus.phosphate)
```

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

```
## Warning: Removed 21995 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21995 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
temp.boxplot <-
  ggplot(PeterPaul.chem.nutrients,
         aes(x = month, y = temperature_C, shape = as.factor(month),
             color = lakename)) +
  geom_boxplot()+
  ylab("Temperature (C)") +
  xlab("Month")
print(temp.boxplot)
```
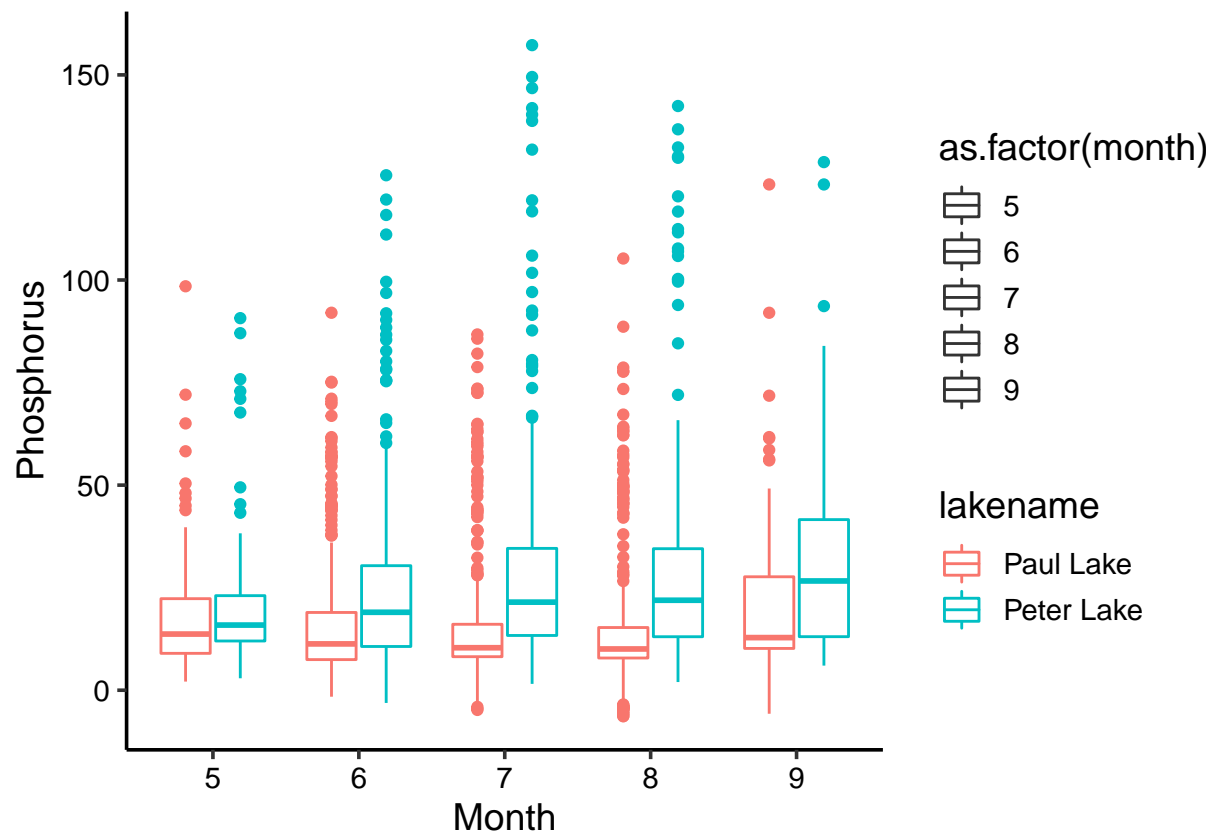
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```
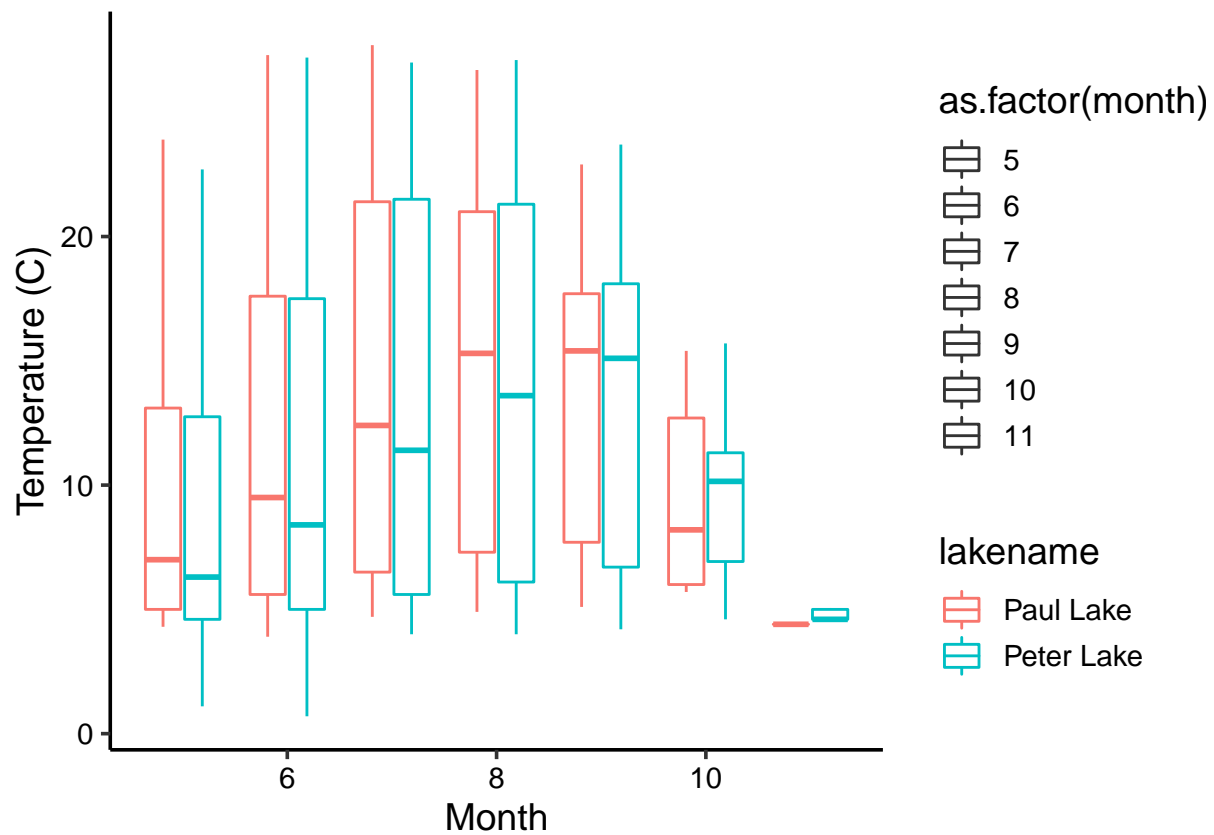
```
TP.boxplot <-
  ggplot(PeterPaul.chem.nutrients,
         aes(x = month, y = tp_ug, shape = as.factor(month),
             color = lakename)) +
  geom_boxplot() +
  ylab("Phosphorus") +
  xlab("Month")
print(TP.boxplot)
```

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

```
TN.boxplot <-
  ggplot(PeterPaul.chem.nutrients,
         aes(x = month, y = tn_ug, shape = as.factor(month),
             color = lakename)) +
  geom_boxplot() +
  ylab("Nitrogen") +
  xlab("Month")
print(TN.boxplot)
```

Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
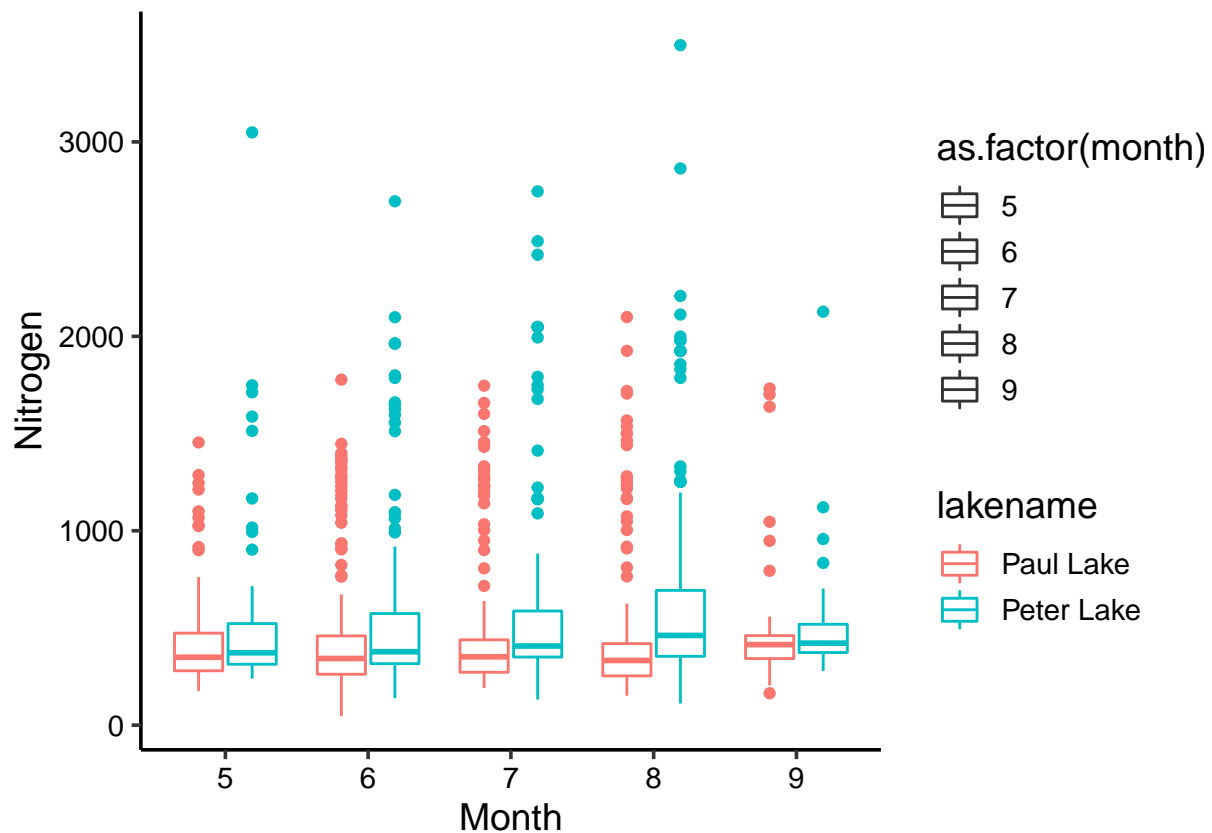
```
## Combining the graphs
Total.plot <-
  plot_grid(temp.boxplot + theme(legend.position='none'),
                   TP.boxplot + theme(legend.position='none'),
                   TN.boxplot + theme(legend.position='none'),
                   nrow = 3, align = 'v', axis = "l")
```
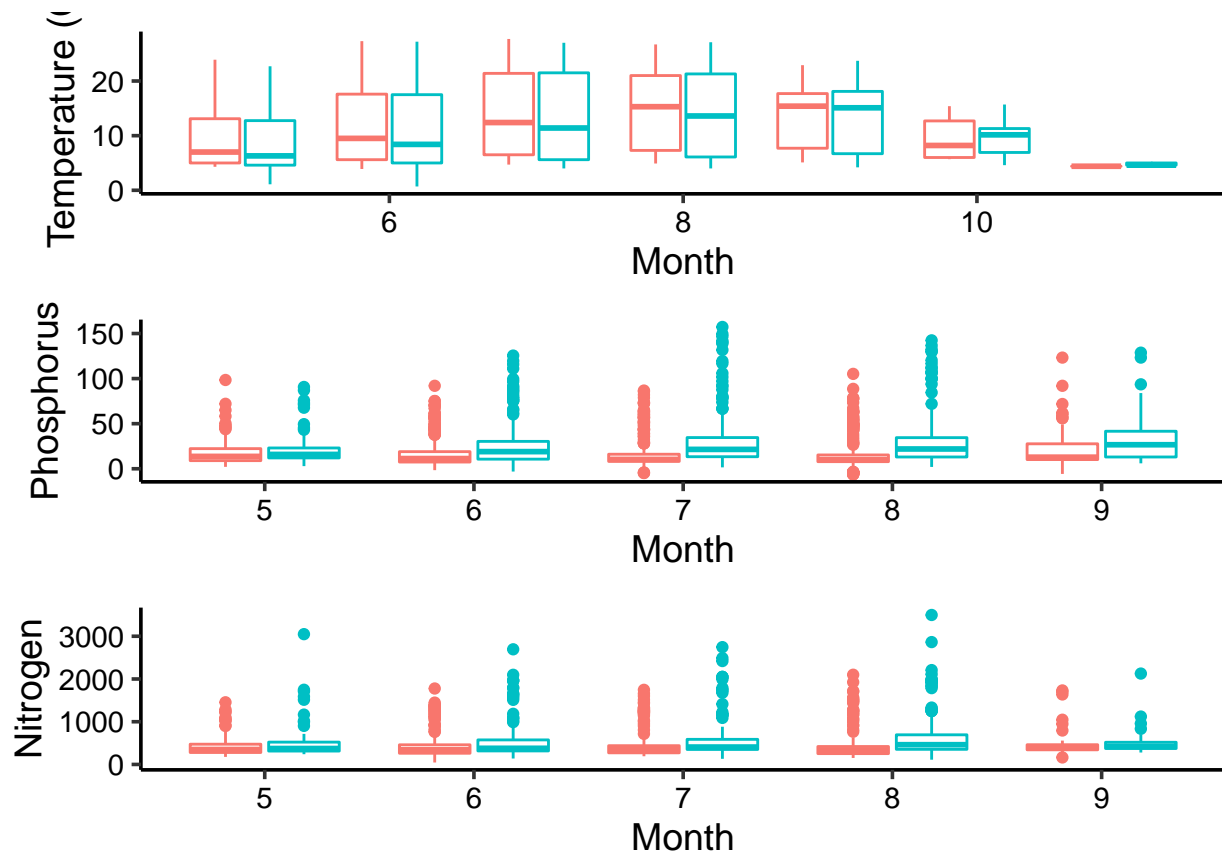
Warning: Removed 3566 rows containing non-finite values (stat_boxplot).

Warning: The shape palette can deal with a maximum of 6 discrete values because
more than 6 becomes difficult to discriminate; you have 7. Consider
specifying shapes manually if you must have them.

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

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

```
print(Total.plot)
```

```r
legend <- get_legend(
  temp.boxplot + theme(legend.box.margin = margin(0, 1, 0, 1)))
```

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

Warning: The shape palette can deal with a maximum of 6 discrete values because
more than 6 becomes difficult to discriminate; you have 7. Consider
specifying shapes manually if you must have them.

```r
Total.plot.legend <-
  plot_grid(Total.plot, legend, rel_widths = c(1, .4))
print(Total.plot.legend)
```
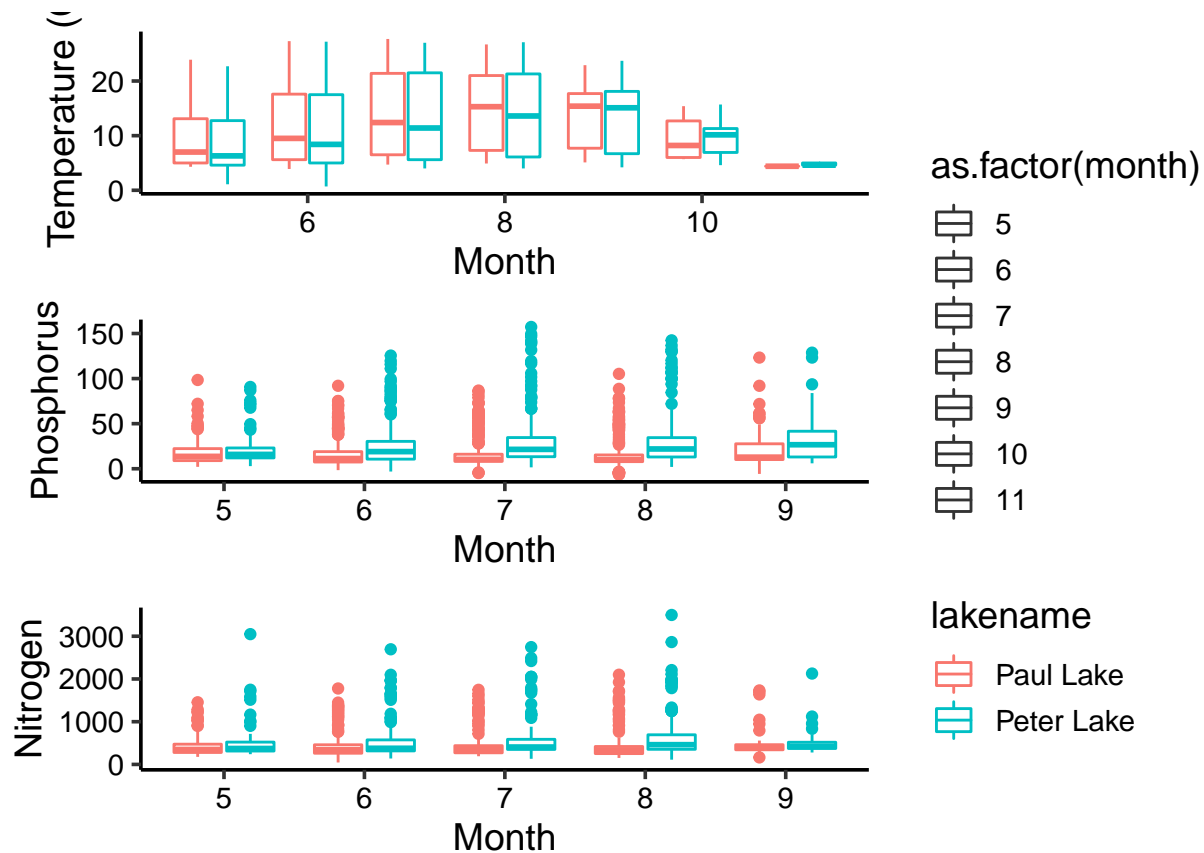
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: The temperatures seem to be relatively consistant between the two lakes, rising in the summer months. The phosphorus and nitrogen levels, however, seem to be higher for Peter Lake and also spike in the warmer summer months.

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
NiwotRidge.litter.needles.color <-
  ggplot(subset(NiwotRidge.litter, functionalGroup == "Needles"),
         aes(x=collectDate, y=dryMass, color = nlcdClass)) +
  geom_point(size =1)
print(NiwotRidge.litter.needles.color)
```
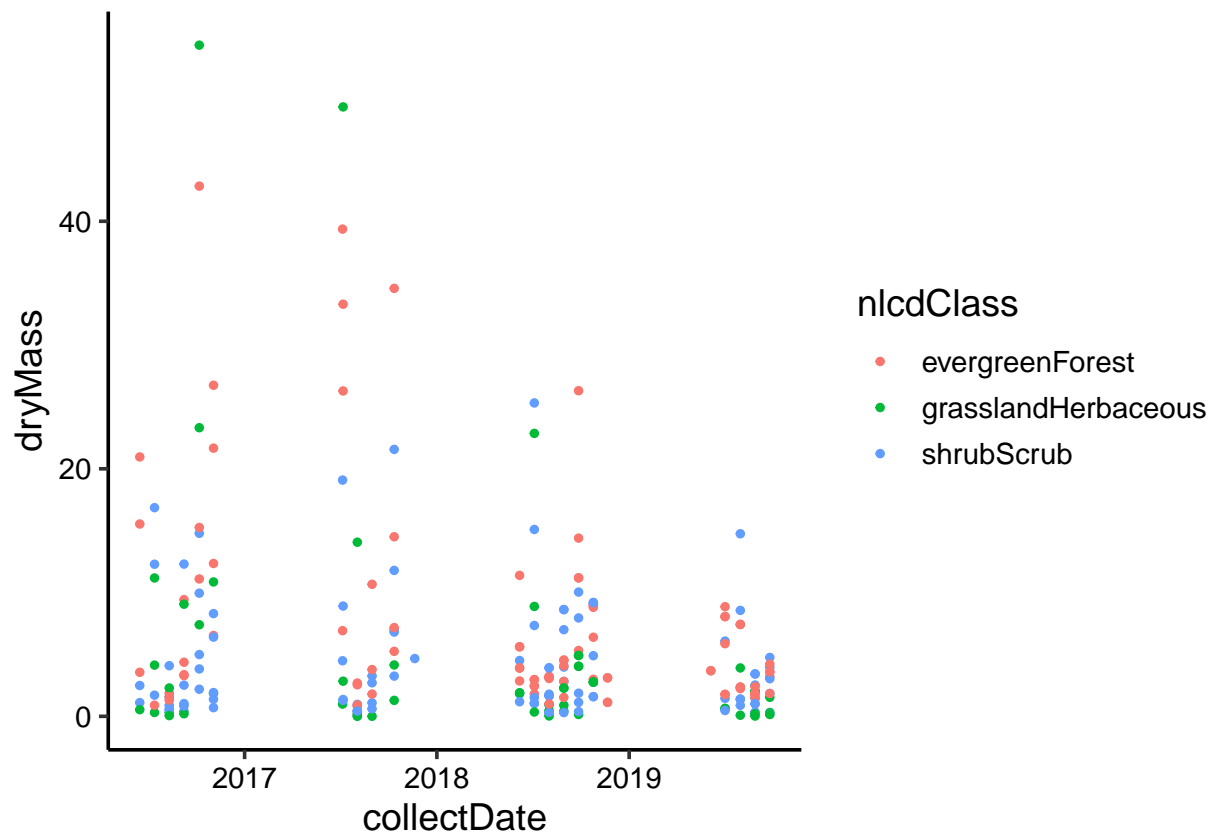
```
#7
NiwotRidge.litter.needles.facets <-
  ggplot(subset(NiwotRidge.litter, functionalGroup == "Needles"),
         aes(x=collectDate, y=dryMass)) +
  geom_point(size =1)+
  facet_wrap(vars(nlcdClass), nrow = 3)
print(NiwotRidge.litter.needles.facets)
```
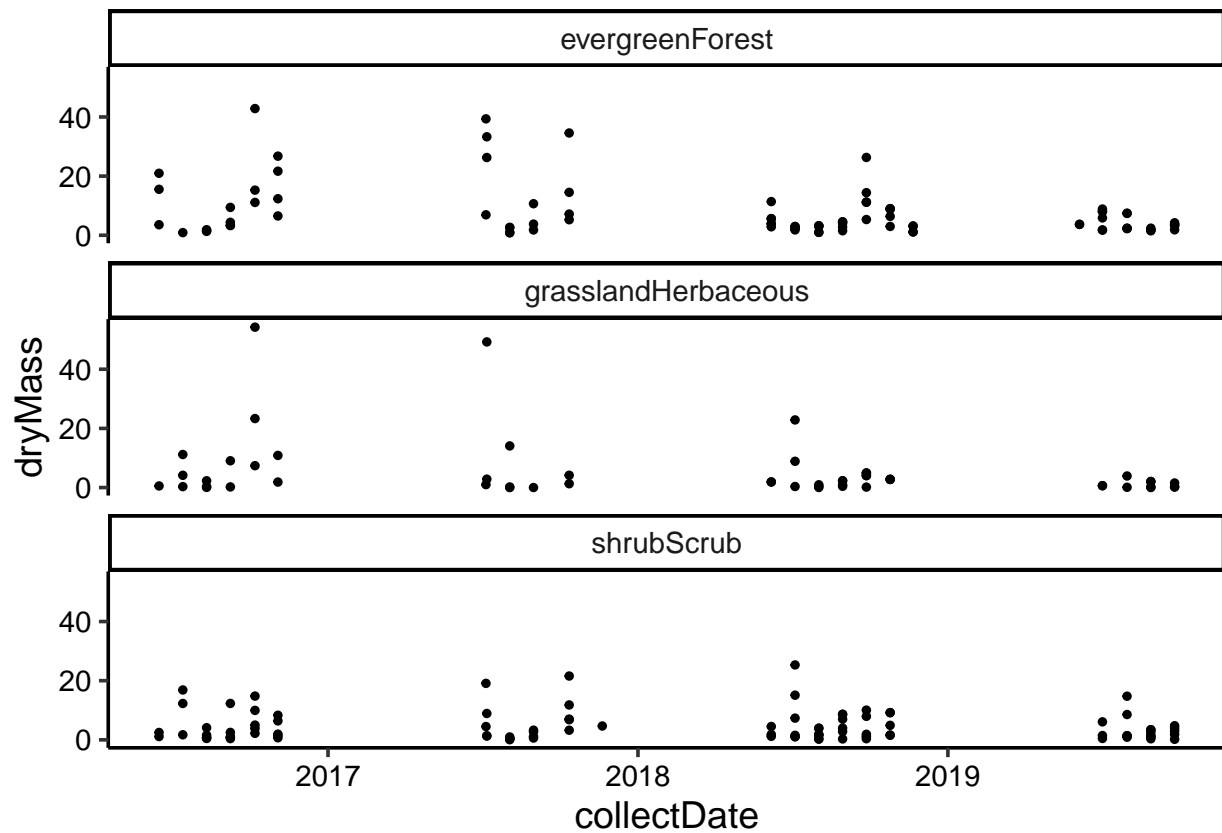
Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer: It depends on the purpose of the plot. Plot 7 is a clearer view of the individual NLCD classes, however, plot 6 allows for an easier direct comparison between the NLCD classes. If it is expected that one NLCD class is corrolated with higher dry mass yields, then plot 6 would allow for an easier assessment of that relatioonship.