Assignment 5: Data Visualization

Laura Brockington

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) #loading in necessary packages
library(lubridate)
library(here)
library(cowplot)
here() #verifying my home directory
```

```
## [1] "/Users/laura/Desktop/EDA/EDA"
```

```
PeterPaul.chem.nutrients <-

  →   read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  →
```

```
              stringsAsFactors = TRUE) #loading in datasets
Niwot.litter.mass <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
                        stringsAsFactors = TRUE)


#2
#checking format of date columns
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
class(Niwot.litter.mass$collectDate)
```

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

```
#changing date columns to read as dates, not factors
PeterPaul.chem.nutrients$sampledate <-
  ymd(PeterPaul.chem.nutrients$sampledate)
Niwot.litter.mass$collectDate <-
  ymd(Niwot.litter.mass$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
mytheme <- theme_gray(base_size = 11) + #creating my custom plot theme
  theme(axis.text = element_text(family = "serif",
                                 color = "black"),
        axis.title = element_text(family = "serif"),
        axis.ticks = element_line(color = "lightblue3",
                                  linewidth = 0.5),
        plot.title = element_text(family = "serif",
                                  face = "bold",
                                  color = "black"),
        panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(color = "lightblue3",
                                        linewidth = 0.5,
                                        linetype = "solid"),
        panel.grid.minor = element_line(color = "lightblue2",
                                        linewidth = 0.25,
                                        linetype = "dashed"),
        plot.background = element_rect(color = "lightblue1"),
```

```
        legend.position = "right",
        legend.key = element_rect(color = "white"),
        legend.text = element_text(family = "serif"),
        legend.title = element_text(family = "serif"),
        legend.title.align = 0.5)
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/or `ylim()`).

```
#4
tp_ug.po4 <- ggplot( #creating scatterplot of phosphorus by phosphate
  PeterPaul.chem.nutrients,
  aes(x=po4,
      y=tp_ug,
      color=lakename)
) +
  geom_point(size=0.85, alpha=0.7) +
  xlim(-1, 40) +
  ylim(-1, 100) +
  geom_smooth(method=lm) +
  labs(
    title = "Total Phosphorus vs Phosphate in Peter and Paul Lakes",
    x = "Phosphate",
    y = "Total Phosphorus",
    col = "Lake") +
  scale_color_manual(values=c("darkolivegreen3", "cyan4"))

print(tp_ug.po4)
```
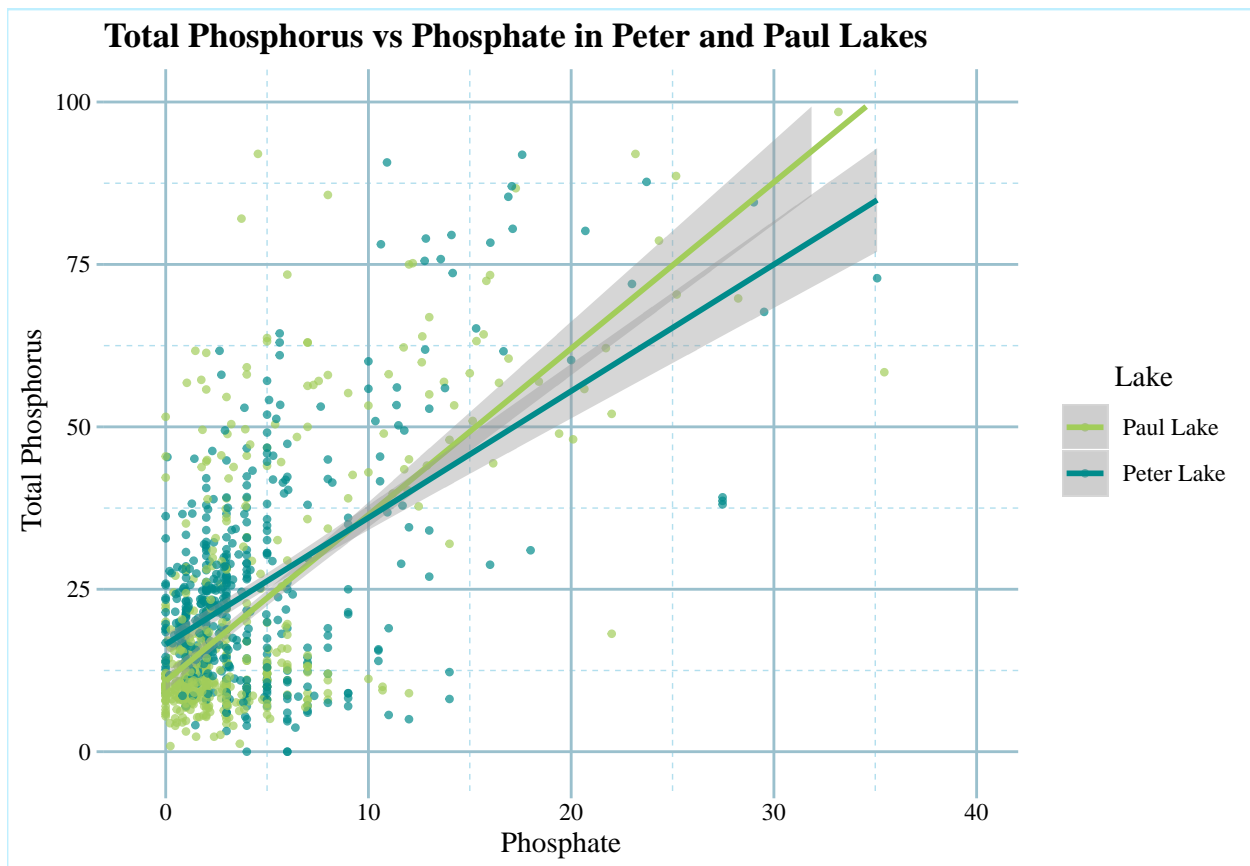
```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21965 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21965 rows containing missing values (`geom_point()`).

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

3

## Total Phosphorus vs Phosphate in Peter and Paul Lakes

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
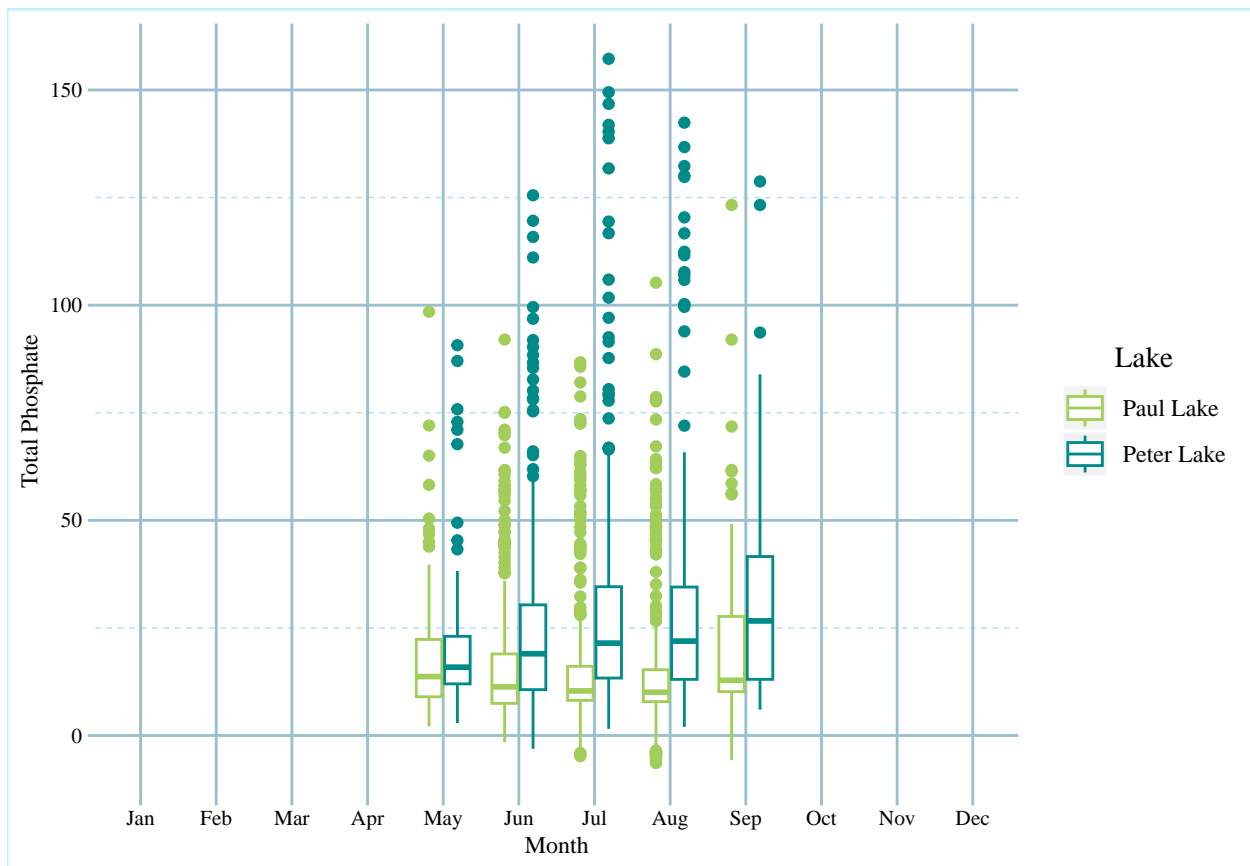
```
#5
temp_month <- #boxplot of temperature by month
  ggplot(
    PeterPaul.chem.nutrients,
    aes(x = (factor(month, levels = 1:12, labels = month.abb)),
        y = temperature_C,
        color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(drop = F,
                   name = "Month") +
  labs(x = "Month",
       y = "Temperature (C)",
       col = "Lake") +
  scale_color_manual(values = c("darkolivegreen3", "cyan4")) +
  theme(axis.text = element_text(size = 8),
        axis.title = element_text(size = 9))
print(temp_month)
```

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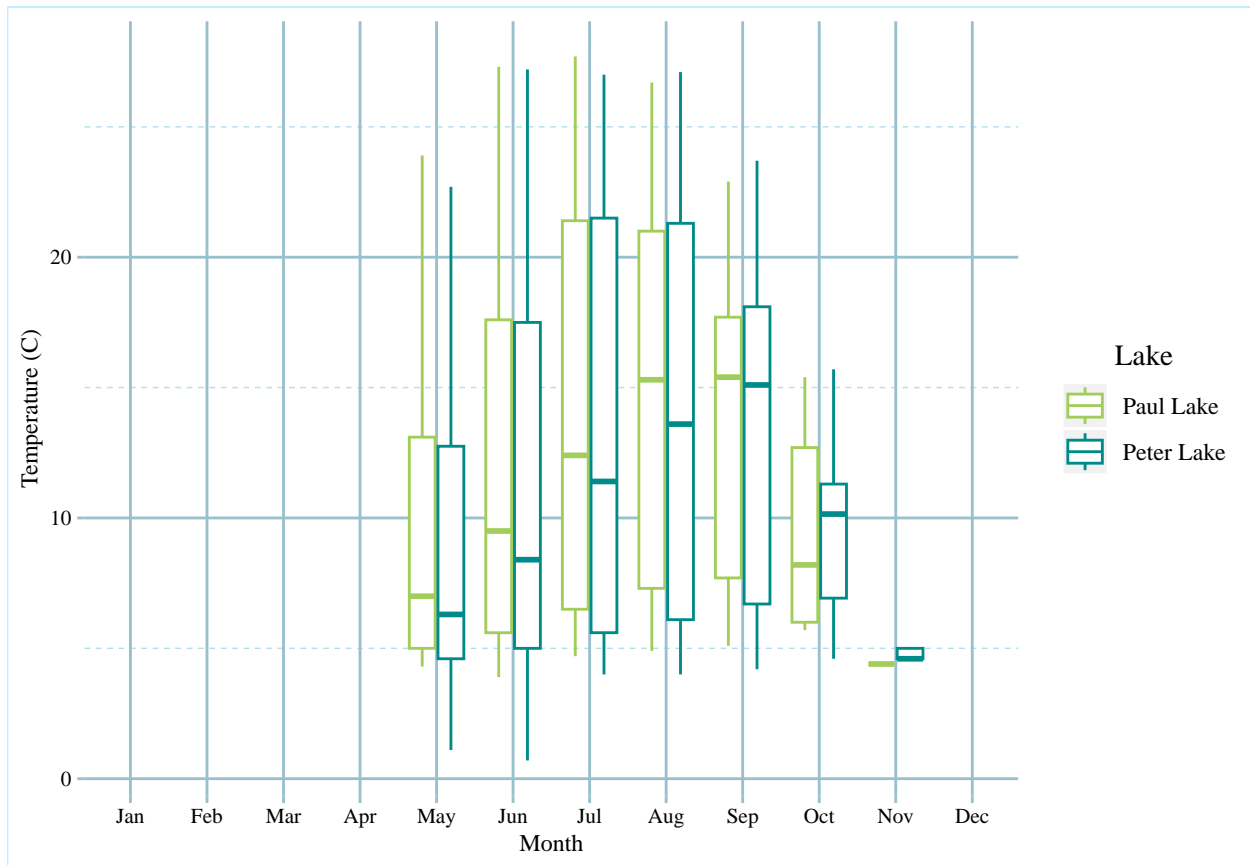


```
tp_month <- #boxplot of total phosphorus by month
  ggplot(
    PeterPaul.chem.nutrients,
    aes(x = (factor(month, levels = 1:12, labels = month.abb)),
        y = tp_ug,
        color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(drop = F,
                   name = "Month") +
  labs(x = "Month",
       y = "Total Phosphate",
       col = "Lake") +
  scale_color_manual(values = c("darkolivegreen3", "cyan4")) +
  theme(axis.text = element_text(size = 8),
        axis.title = element_text(size = 9))
print(tp_month)
```

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

```
tn_month <- #boxplot of total nitrogen by month
  ggplot(
    PeterPaul.chem.nutrients,
    aes(x = (factor(month, levels = 1:12, labels = month.abb)),
        y = tn_ug,
        color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(drop = F,
                   name = "Month") +
  labs(x = "Month",
       y = "Total Nitrogen",
       col = "Lake") +
  scale_color_manual(values = c("darkolivegreen3", "cyan4")) +
  theme(axis.text = element_text(size = 8),
        axis.title = element_text(size = 9))
print(tn_month)
```

```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```
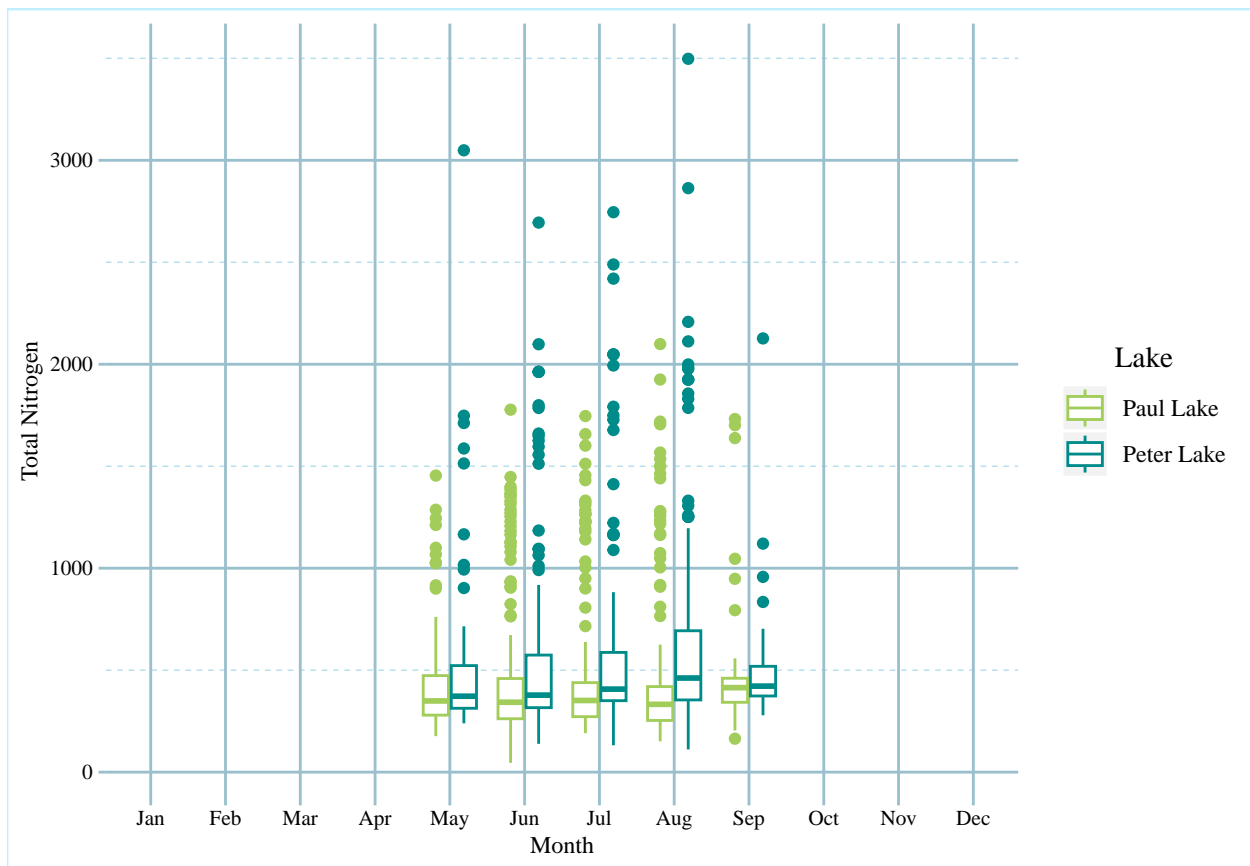
```r
#creating cowplot with all 3 boxplots
cowplot_no_legend <- plot_grid(temp_month +
                                    theme(legend.position="none"),
          tp_month + theme(legend.position="none"),
          tn_month + theme(legend.position="none"),
          nrow = 3,
          align = 'hv')
```

```
## 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()`).
```

```r
legend <- get_legend(
  tp_month +
    theme(legend.position = "bottom",
          legend.text = element_text(size = 8),
          legend.title = element_text(size = 9)))
```

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

```r
#creating cowplot of all 3 boxplots with only 1 legend
cowplot <- plot_grid(cowplot_no_legend, legend,
                     ncol = 1, rel_heights = c(1.2, 0.1))
print(cowplot)
```
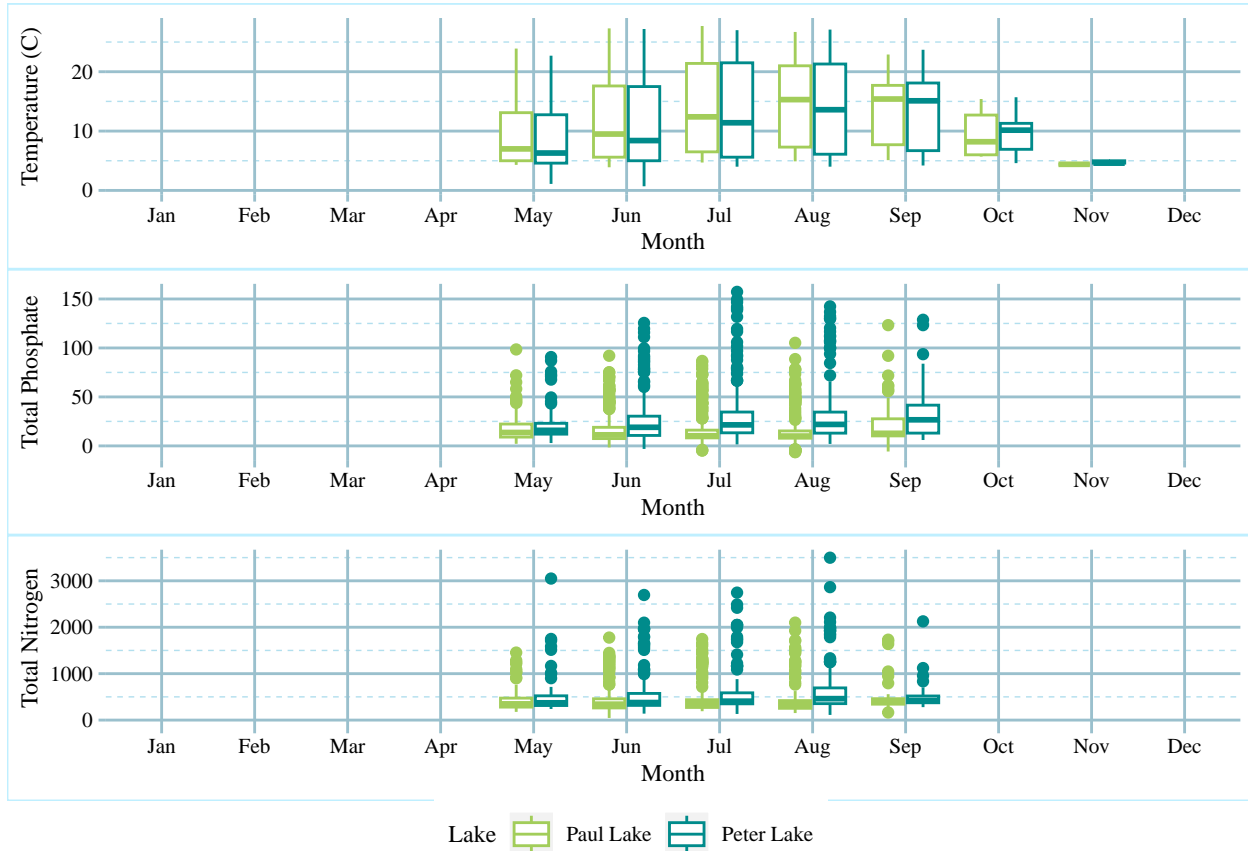

Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: In both total phosphate and total nitrogen of Peter Lake, there appears to be more dipersion in warmer months. In Paul Lake, that difference is not as apparent. Throughout the year, there appears to be less total phosphate and total nitrgoen in Paul Lake than in Peter Lake.

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.
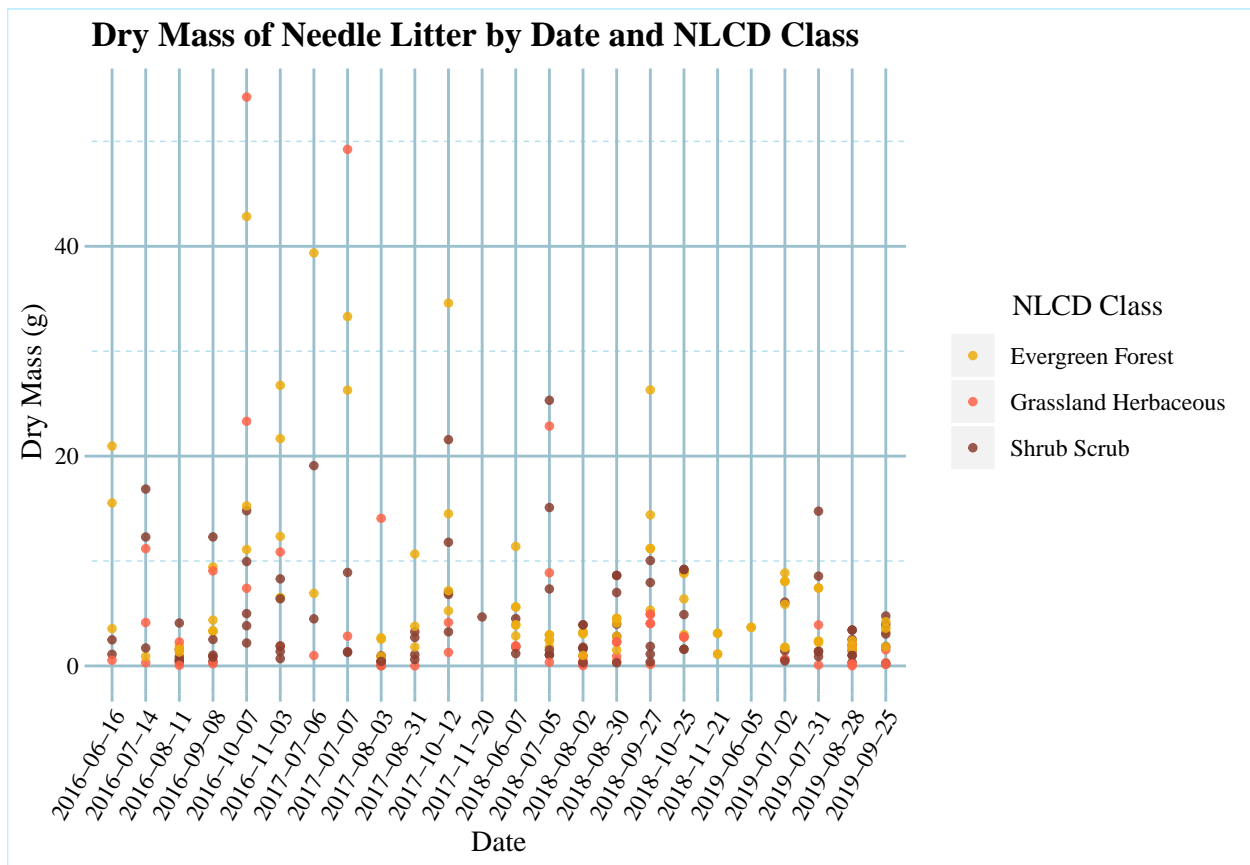
```r
#6
needles_drymass <- Niwot.litter.mass %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    aes(x = factor(collectDate),
        y = dryMass,
        color = nlcdClass)) +
```

```
  geom_point(size = 1, alpha = 0.85) +
  labs(
    title = "Dry Mass of Needle Litter by Date and NLCD Class",
    x = "Date",
    y = "Dry Mass (g)",
    col = "NLCD Class") +
  scale_color_manual(values = c("darkgoldenrod2", "tomato1", "coral4"),
                     labels = c("Evergreen Forest",
                                "Grassland Herbaceous",
                                "Shrub Scrub")) +
  theme(axis.text.x = element_text(angle=60, hjust=1))
print(needles_drymass)
```


Dry Mass of Needle Litter by Date and NLCD Class

```
#7
needles_drymass_faceted <- Niwot.litter.mass %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    aes(x = factor(collectDate),
        y = dryMass)) +
  geom_point(size = 1, alpha = 0.85, col = "darkgoldenrod2") +
  facet_wrap(vars(nlcdClass),
             labeller = as_labeller(c("evergreenForest" =
                                        "Evergreen Forest",
                                      "grasslandHerbaceous" =
                                        "Grassland Herbaceous",
```

```
                                       "shrubScrub" =
                                         "Shrub Scrub"))) +
  labs(
    title = "Dry Mass of Needle Litter by Date and NLCD Class",
    x = "Date",
    y = "Dry Mass (g)") +
  theme(axis.text.x = element_text(angle=60, hjust=1),
        axis.text = element_text(size = 6))
print(needles_drymass_faceted)
```


Dry Mass of Needle Litter by Date and NLCD Class
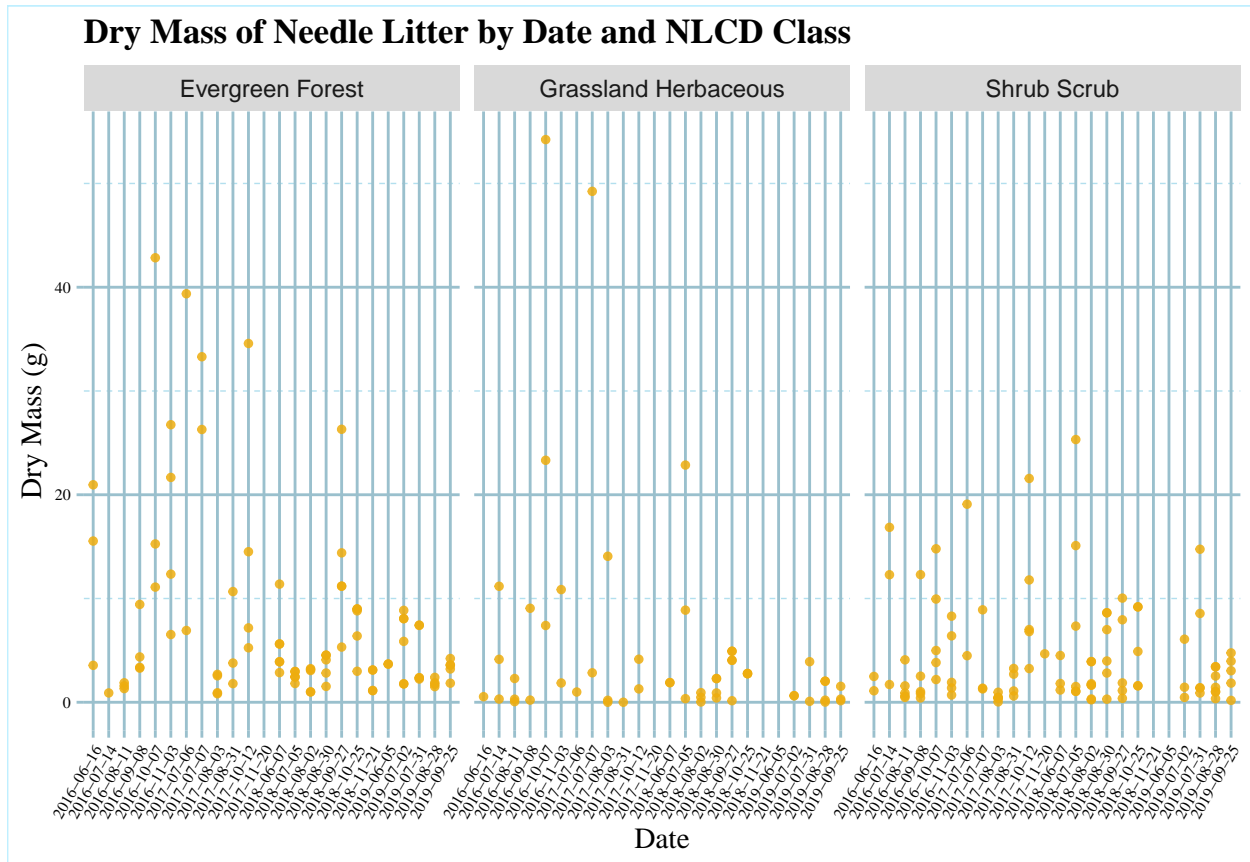
Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer: I find the plot in #6 to be more effective. It is much easier to compare the dry mass of litter on a certain date between NLCD classes when they are on the same line. I also find it less crowded and easier to read than the graph in #7. #7 has an unneccesary repition of axis.