Assignment 5: Data Visualization

Joanna Huertas

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
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.1.0
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)  #loading
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(here)  #loading
```

```
## here() starts at C:/Users/joann/Documents/EDA-Spring2023
```

```
library(cowplot)  #loading
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
# Add pre-set themes <--NEW
library(ggthemes)
```

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

```
getwd()  #Checking working directory
```

```
## [1] "C:/Users/joann/Documents/EDA-Spring2023/Assignments"
```

```
here()
```

```
## [1] "C:/Users/joann/Documents/EDA-Spring2023"
```

```
PeterPaul.nutrients <- read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Pr
    stringsAsFactors = TRUE)
```

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

```
# 2
```

```
class(PeterPaul.nutrients$sampledate)
```

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

```
class(NEON.litter$collectDate)
```

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

```
# change the format to date

PeterPaul.nutrients$sampledate <- ymd(PeterPaul.nutrients$sampledate)
NEON.litter$collectDate <- ymd(NEON.litter$collectDate)

# Verifying
class(PeterPaul.nutrients$sampledate)   #date format
```

```
## [1] "Date"
```

```
class(NEON.litter$collectDate)   #date format
```

```
## [1] "Date"
```

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

# Create a custom plot theme

my_theme <- theme_base() +
  theme(
    line = element_line(
      color='black',
      linewidth =1
    ),
    plot.background =   element_rect(
      color='yellow',
      fill='#FFFF99'
    ),
    axis.text = element_text(size = 11),
    plot.title =        element_text(
      size=14,
      color='black'
    ),
    axis.title = element_text(size = 14, face = "bold"),
```

```
    legend.background = element_rect(
      color='grey',
      fill = 'orange'
    ),
    legend.title = element_text(
      color='blue'
    )

 )
```

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()`).

```r
# 4


total.p.by.po4 <- PeterPaul.nutrients %>%
    ggplot(mapping = aes(x = tp_ug, y = po4, color = lakename)) +
    geom_point() + # Add a line of best fit and color it black geom_point()
    geom_point() + # Add a line of best fit and color it black +
    geom_point() + # Add a line of best fit and color it black #
    geom_point() + # Add a line of best fit and color it black Add
    geom_point() + # Add a line of best fit and color it black a
    geom_point() + # Add a line of best fit and color it black line
    geom_point() + # Add a line of best fit and color it black of
    geom_point() + # Add a line of best fit and color it black best
    geom_point() + # Add a line of best fit and color it black fit
    geom_point() + # Add a line of best fit and color it black and
    geom_point() + # Add a line of best fit and color it black color
    geom_point() + # Add a line of best fit and color it black it
    geom_point() + # Add a line of best fit and color it black black
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme va
```

```
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme v
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme v
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme v
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme v
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme v
geom_smooth(method = "lm", se = FALSE, color = "black") + # Adjust the x and y limits to hide extreme v
xlim(0, 125) + ylim(0, 50) + my_theme

print(total.p.by.po4)
```
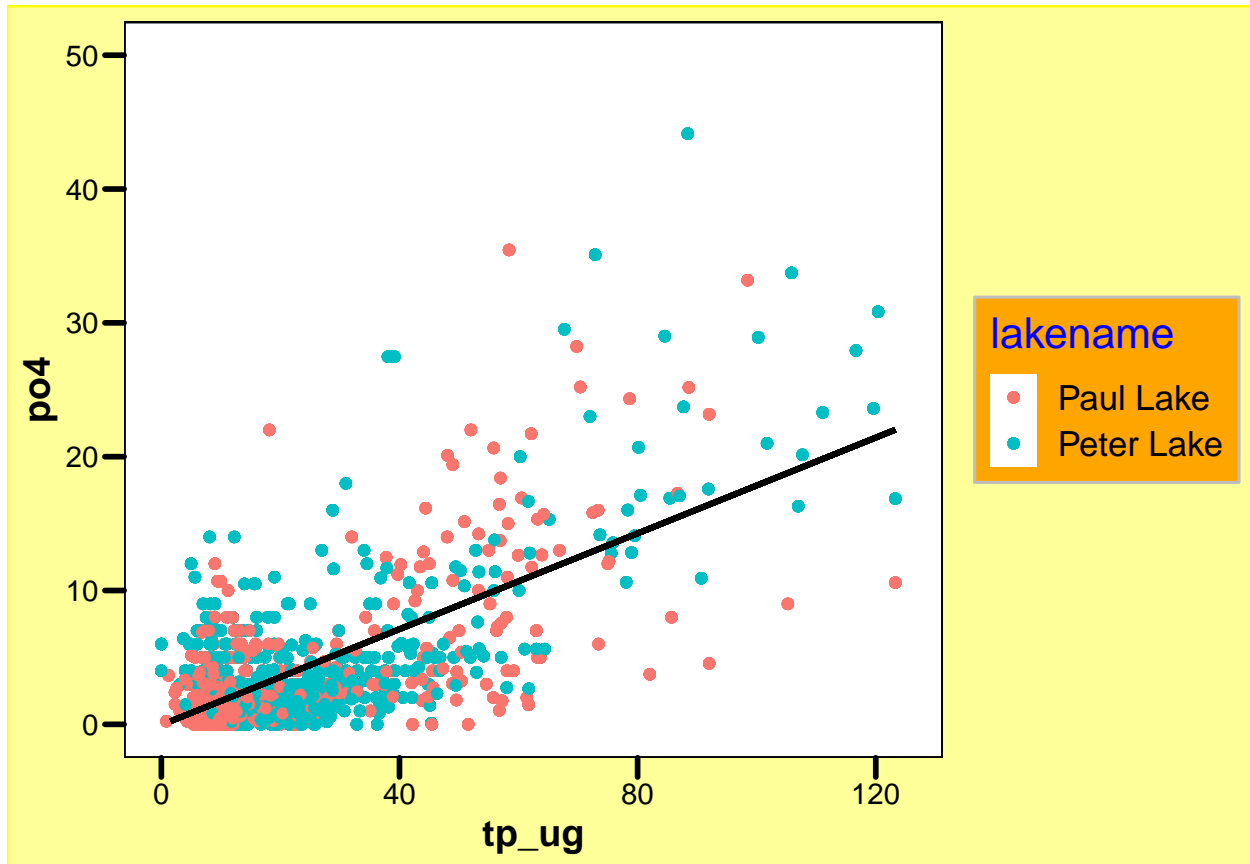

5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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 Convert month to a factor -- with 12 levels, labelled
# with month names
invisible(factor(PeterPaul.nutrients$month, levels = 1:12, labels = month.abb))

# a
Temperature.plot <- ggplot(PeterPaul.nutrients, aes(x = factor(PeterPaul.nutrients$month,
    levels = 1:12, labels = month.abb), y = temperature_C, color = lakename)) +
```

```
    geom_boxplot() + my_theme + labs(title = "Temperature vs months",
    x = "Month", y = "Temperature (C)", color = "Lakes")
print(Temperature.plot)
```
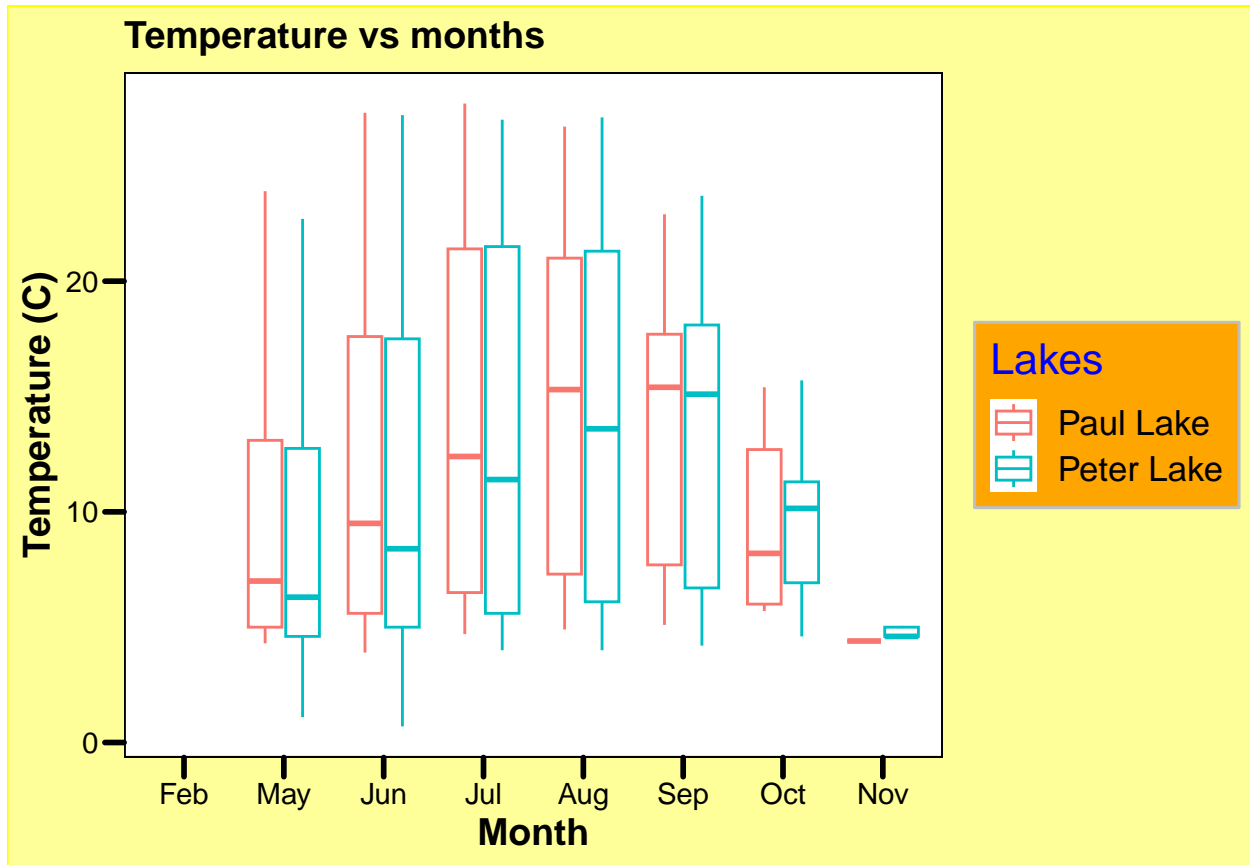


```
# b
TP.plot <- ggplot(PeterPaul.nutrients, aes(x = factor(PeterPaul.nutrients$month,
    levels = 1:12, labels = month.abb), y = tp_ug, color = lakename)) +
    geom_boxplot() + my_theme + labs(title = "TP vs months",
    x = "Month", color = "Lakes")
print(TP.plot)
```
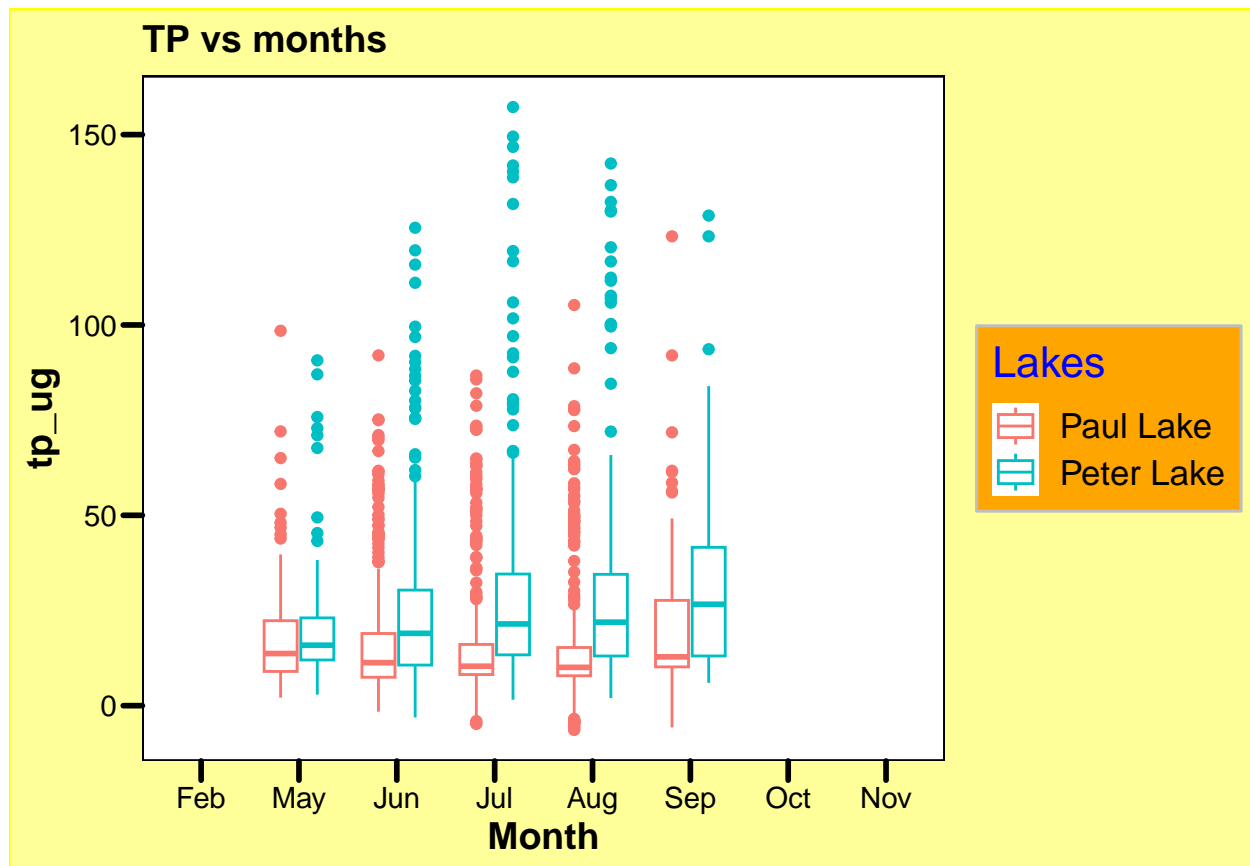
```
# C
TN.plot <- ggplot(PeterPaul.nutrients, aes(x = factor(PeterPaul.nutrients$month,
    levels = 1:12, labels = month.abb), y = tn_ug, color = lakename)) +
    geom_boxplot() + my_theme + labs(title = "TN vs months",
    x = "Month", color = "Lakes")
print(TN.plot)
```
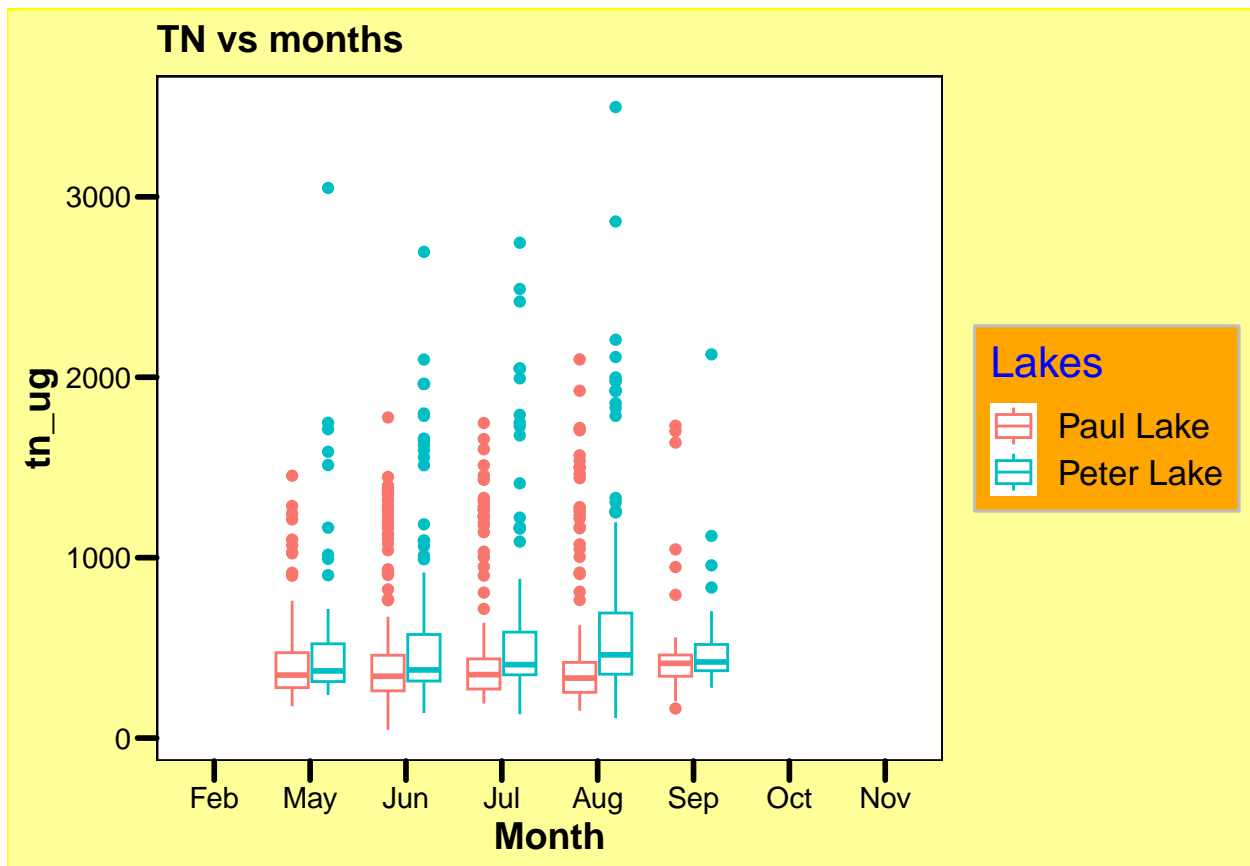
```
plot_grid(Temperature.plot + theme(legend.position = "none"),
    TP.plot + theme(legend.position = "none"), TN.plot + theme(legend.position = "bottom"),
    nrow = 3, align = "hv", axis = "l", rel_heights = c(1.5,
        1.5, 2))
```
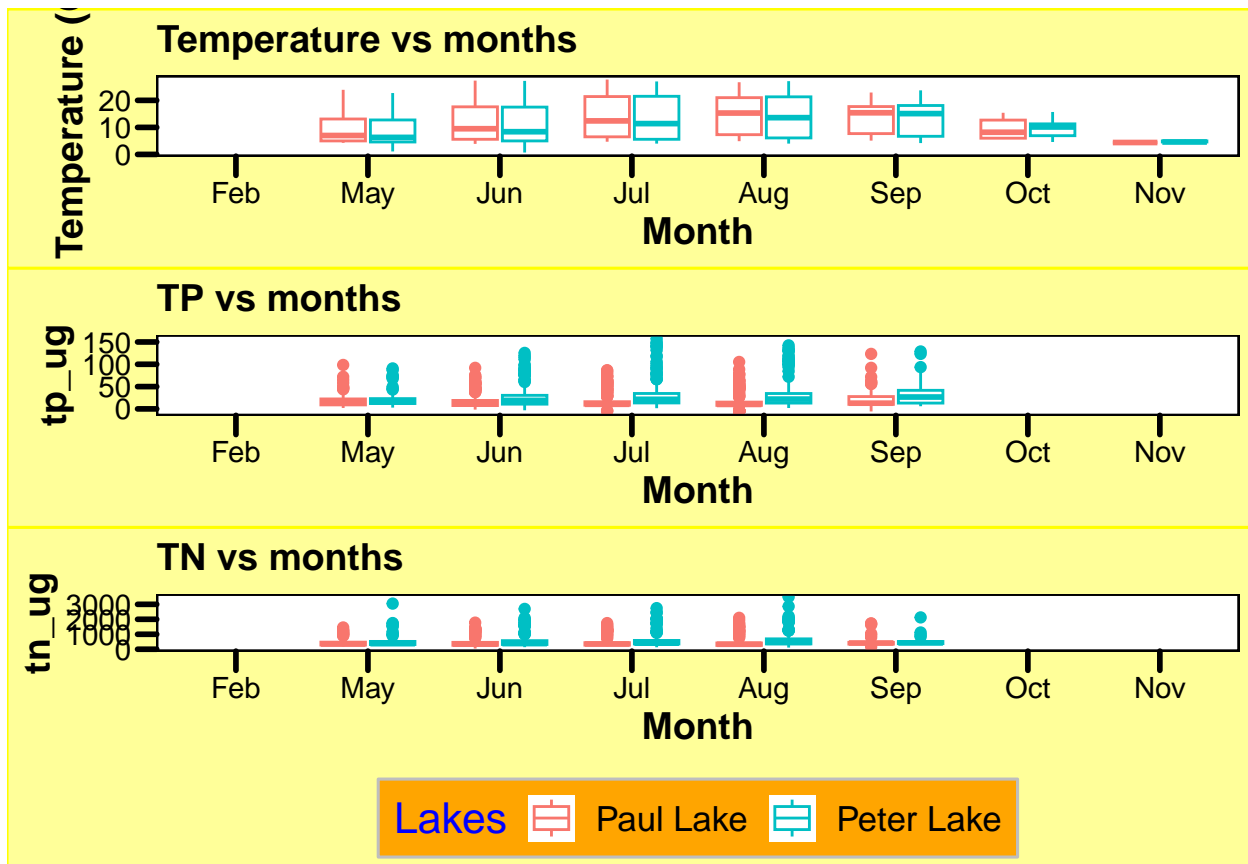
```
# extract a legend that is laid out horizontally legend <-
# get_legend( TN.plot + guides(color = guide_legend(nrow =
# 1)) + theme(legend.position = 'bottom') )
# plot_grid(Temperature.plot +
# theme(legend.position='none'), TP.plot +
# theme(legend.position='none'), TN.plot+
# theme(legend.position='none'), legend, nrow = 4, align =
# 'hv', axis = 'l', rel_heights = c(2, 2, 2, .5))
```

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

Answer: For temperature, I notice it is logical that the median is higher during the summer months. Between the two lakes, Paul Lake has a slightly higher median temperature than Peter Lake, most of the time. Also, we can see that the data is spread out. Regarding TP, the data overall is not very spread out, and Peter Lake has a higher median phosphorus concentration than Paul Lake in all months. As for TN, the data is not very spread out, and Peter Lake has a slightly higher median nitrogen concentration than Paul Lake in all months. The trend for TP and TN is very similar. During the summer months, Peter Lake seems to have a slightly higher concentration of TP and TN, but for Paul Lake, I don't see a trend.

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
Needles_vs_dryMass <- NEON.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
      color=nlcdClass)
    ) +
  geom_point(size=2, alpha=0.8)+ #alpha is transparency
  #facet_wrap(vars(lakename))+ #2plots 2 lakenames
  labs(title="Needles", x="Date", y="Dry Mass", color="NLCD class")+
  scale_x_date(date_breaks = "9 month", date_labels = "%Y-%m")+
  my_theme

print(Needles_vs_dryMass)
```
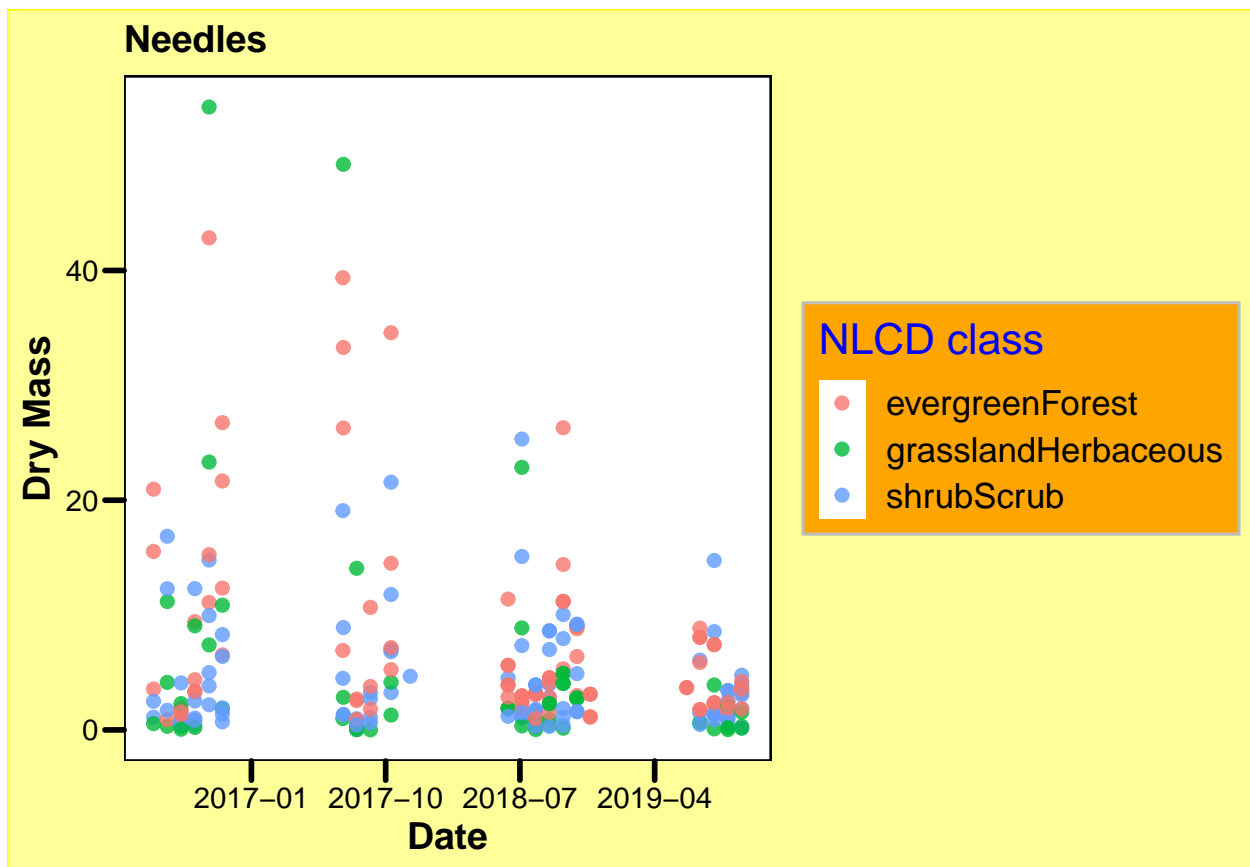


```
#7
Needles_dryMass_facets <- NEON.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
```

```
      color=nlcdClass)
    ) +
  geom_point(size=2, alpha=0.8)+ #alpha is transparency
  facet_wrap(vars(nlcdClass))+ #2plots 2 lakenames
  labs(title="Needles", x="Date (Years)", y="Dry Mass", color="NLCD class")+
  #scale_x_date(date_breaks = "16 month", date_labels = "%Y-%m")+
  my_theme

print(Needles_dryMass_facets)
```
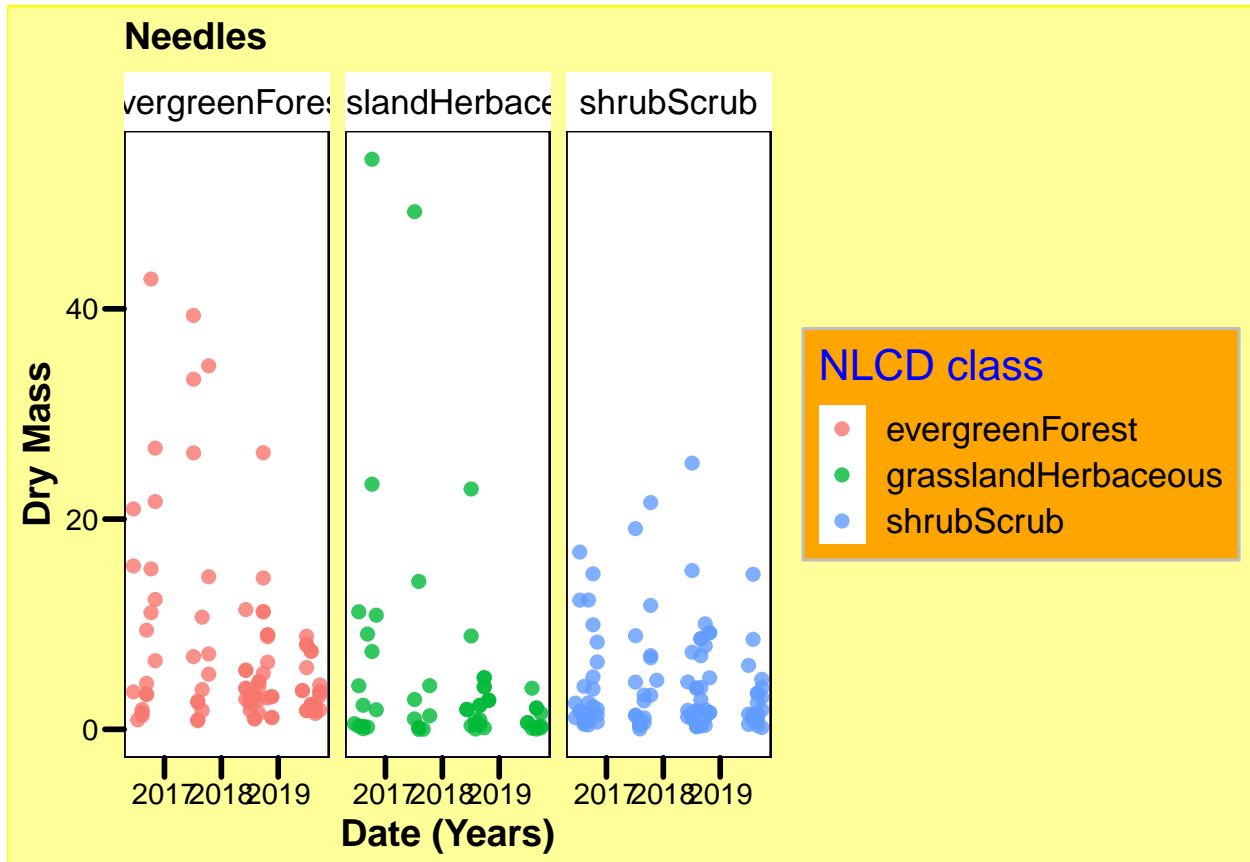

Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer: I think the first one (6) is better because it is easier for me to compare between each NLCD Class and knwo which one has a lower or higher dry mass at a specific point in time. While with the second plot I can see that all NLCD classes follow the same pattern, but it's difficult to compare individual points in time.