

Assignment 5: Data Visualization

Andrew Barfield

Fall 2024

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. Read in 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 in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the NEON_NIWO_Litter_mass_trap_Processed.csv version, again from the Processed_KEY folder).
2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2     3.5.1      v tibble    3.2.1
## v lubridate   1.9.3      v tidyr     1.3.1
## v purrr       1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
library(cowplot)
```

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

```
library(ggthemes)
```

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

```
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
PeterPaul.Chemistry.Nutrients.Processed <-
  read.csv(here
    ("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
    stringsAsFactors = TRUE)
```

```
Niwot.Ridge.Litter <-
  read.csv(here
    ("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
    stringsAsFactors = TRUE)
```

```
#2
class(PeterPaul.Chemistry.Nutrients.Processed$sampldate)
```

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

```
class(Niwot.Ridge.Litter$collectDate)
```

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

```
PeterPaul.Chemistry.Nutrients.Processed$sampldate <-
  ymd(PeterPaul.Chemistry.Nutrients.Processed$sampldate)
Niwot.Ridge.Litter$collectDate <-
  ymd(Niwot.Ridge.Litter$collectDate)

class(PeterPaul.Chemistry.Nutrients.Processed$sampldate)
```

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

```
class(Niwot.Ridge.Litter$collectDate)
```

```
## [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
my_theme <- theme_base() +
  theme(
    legend.position = 'bottom',
    plot.background = element_rect(fill = 'lightgrey'),
    legend.background = element_rect(fill='lightblue',
    size = 10, linetype = "solid"),
    legend.key = element_rect (fill = 'lightblue'),
    plot.title = element_text(size = 15, color = 'black',
    hjust=0.5),
    axis.text = element_text(color = 'black', size = 8),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    legend.title = element_blank()
  )
```

```
## Warning: The 'size' argument of 'element_rect()' is deprecated as of ggplot2 3.4.0.
## i Please use the 'linewidth' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

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 line(s) of best fit using the **lm** method. Adjust your axes to hide extreme values (hint: change the limits using **xlim()** and/or **ylim()**).

```

#4
PeterPaul.Chemistry.Nutrients.Processed %>%
  ggplot(aes
    (x=po4,
     y=tp_ug,
     color=lakename)) +
  geom_smooth(
    method = "lm",
    color = 'black',
    size = 0.5,
    se = FALSE) +
  geom_point() +
  my_theme +
  xlim (0,50) +
  ylim(0,200) +
  ggtitle("Total Phosphorus vs. Phosphate Peter and Paul Lake") +
  xlab("Phosphate") +
  ylab("Total Phosphorus")

```

```

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

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

```

```

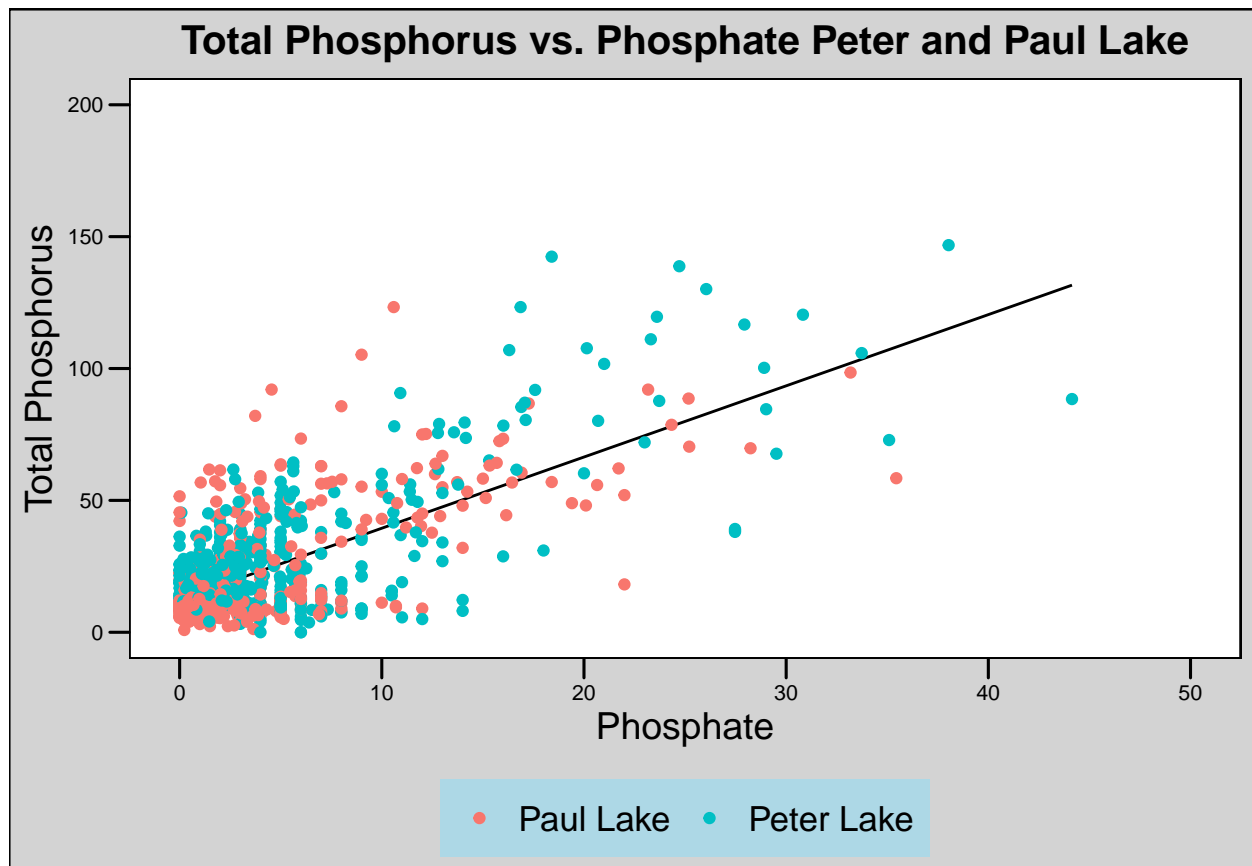
## Warning: Removed 21948 rows containing non-finite outside the scale range
## ('stat_smooth()').

```

```

## Warning: Removed 21948 rows containing missing values or values outside the scale range
## ('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.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
PeterPaul.Chemistry.Nutrients.Processed$month <-
  factor(PeterPaul.Chemistry.Nutrients.Processed$month,
    levels = 1:12,
    labels = month.abb)

Temp.Boxplot <-
  PeterPaul.Chemistry.Nutrients.Processed %>%
  ggplot(aes(
    x=month,
    y=temperature_C,
    color=lakename)) +
  geom_boxplot() +
  my_theme +
  theme(panel.grid.major = element_line(
    color = 'lightgrey', size = 0.25)) +
```

```

theme(panel.grid.minor = element_line(
  color = 'lightgrey', size = 0.25)) +
theme(panel.grid.minor = element_line()) +
theme(axis.title.x = element_blank()) +
theme(legend.position = 'none') +
ggtitle ("Temperature per Month Peter Paul Lake") +
xlab("Month") +
ylab("Temp (C)")

```

```

## Warning: The 'size' argument of 'element_line()' is deprecated as of ggplot2 3.4.0.
## i Please use the 'linewidth' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

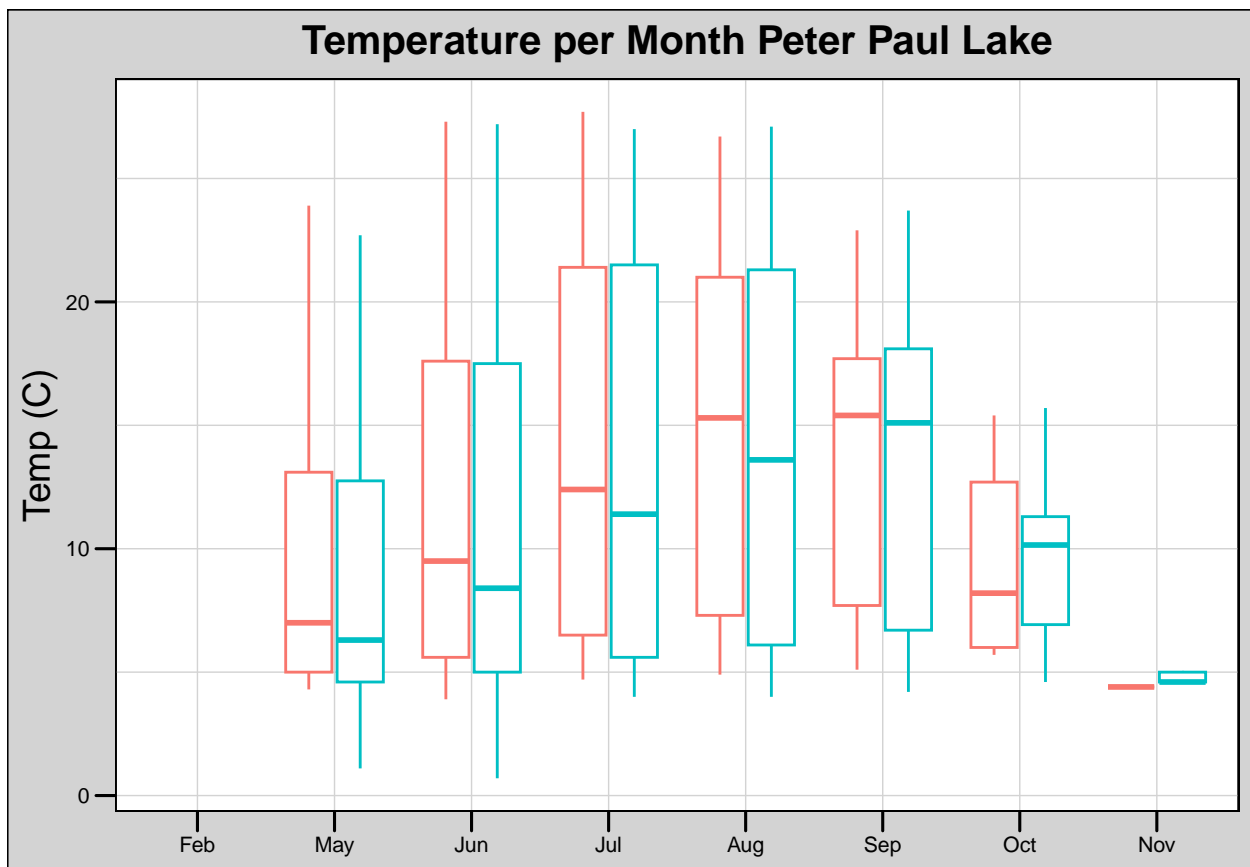
print(Temp.Boxplot)

```

```

## Warning: Removed 3566 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```



```

TP.Boxplot <-
  PeterPaul.Chemistry.Nutrients.Processed %>%
  ggplot(aes(

```

```

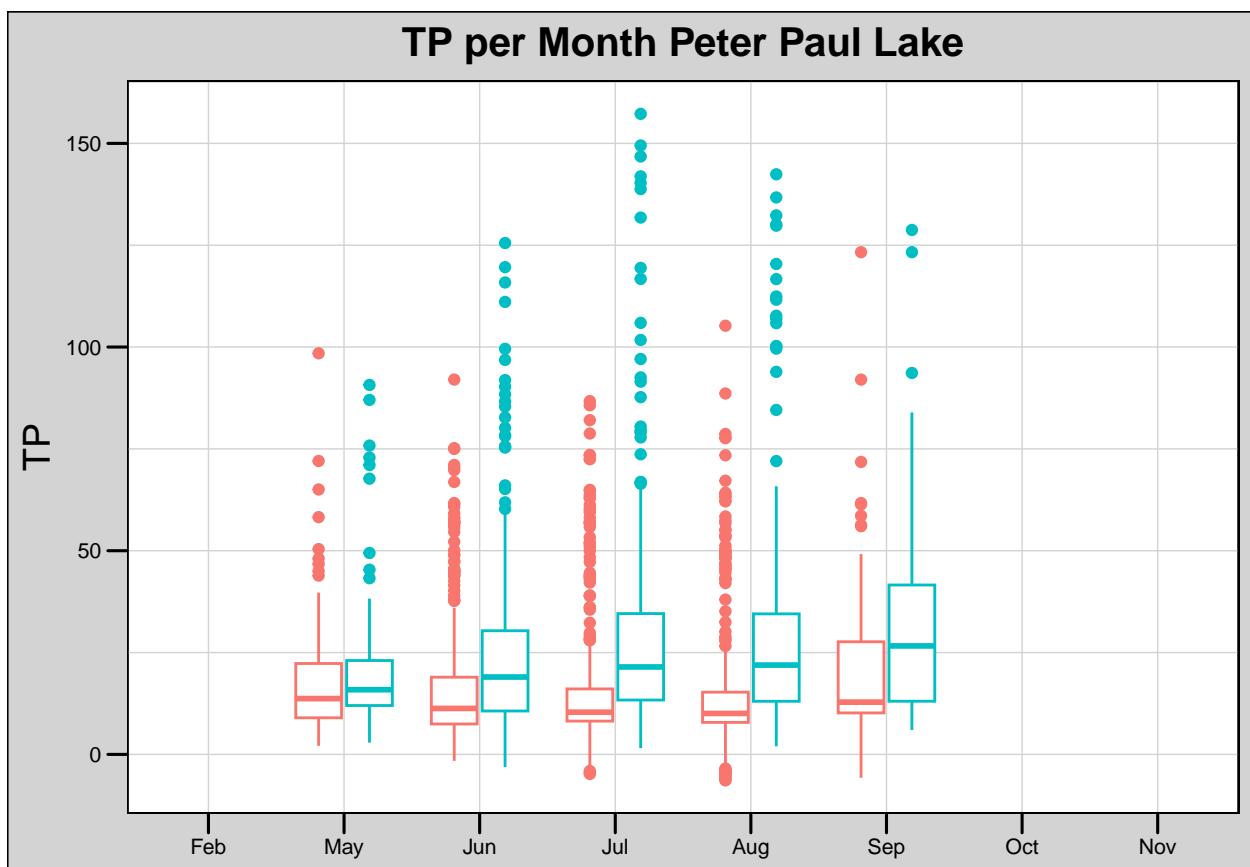
x=month,
y=tp_ug,
color=lakename))+
geom_boxplot() +
my_theme +
theme(panel.grid.major = element_line(
color = 'lightgrey', size = 0.25)) +
theme(panel.grid.minor = element_line(
color = 'lightgrey', size = 0.25)) +
theme(panel.grid.minor = element_line()) +
theme(axis.title.x = element_blank()) +
theme(legend.position = 'none') +
ggtitle ("TP per Month Peter Paul Lake") +
xlab("Month") +
ylab("TP")
print(TP.Boxplot)

```

```

## Warning: Removed 20729 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```



```

TN.Boxplot <-
PeterPaul.Chemistry.Nutrients.Processed %>%
ggplot(aes(
x=month,
y=tn_ug,

```

```

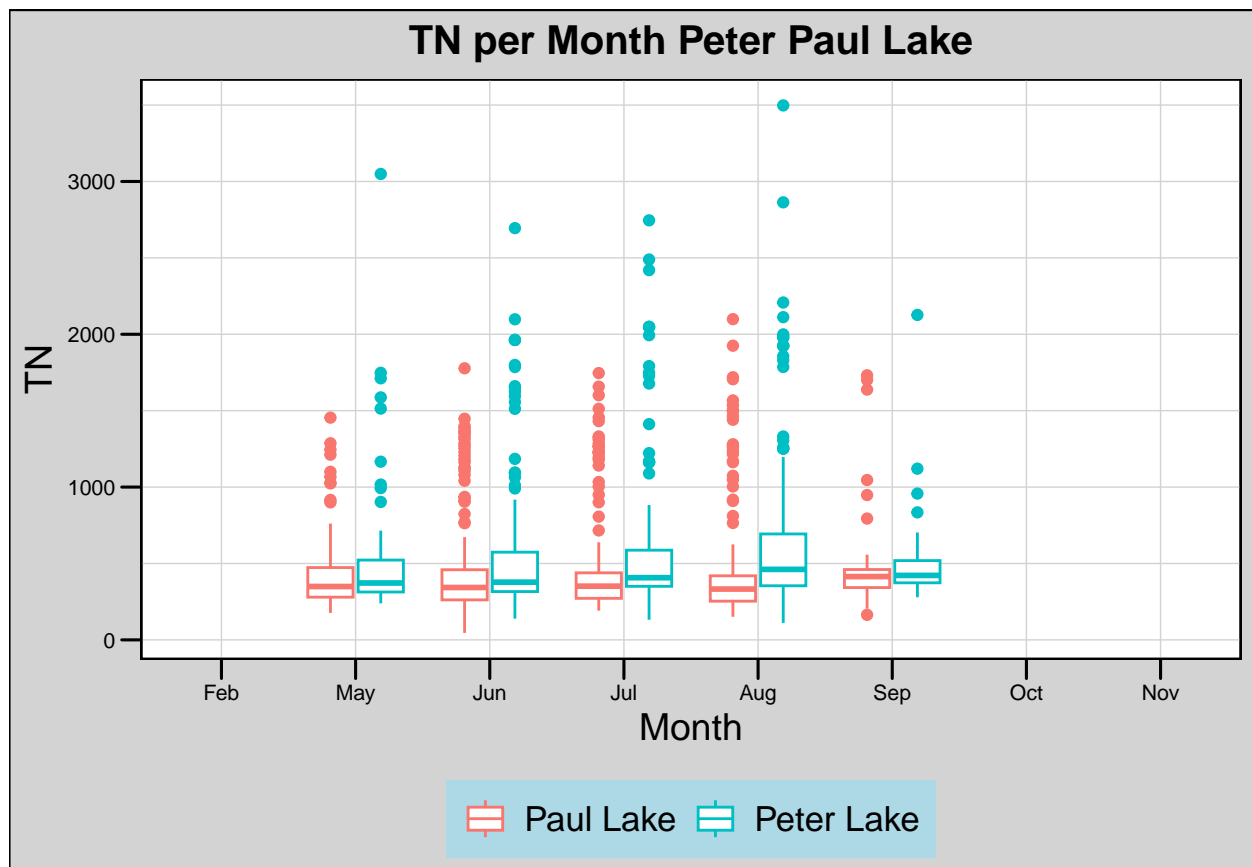
    color=lakename)) +
geom_boxplot() +
my_theme +
theme(panel.grid.major = element_line(
  color = 'lightgrey', size = 0.25)) +
theme(panel.grid.minor = element_line(
  color = 'lightgrey', size = 0.25)) +
theme(panel.grid.minor = element_line()) +
ggtitle ("TN per Month Peter Paul Lake") +
xlab("Month") +
ylab("TN")
print(TN.Boxplot)

```

```

## Warning: Removed 21583 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```



```

plot_grid(Temp.Boxplot, TP.Boxplot, TN.Boxplot,
  align = 'v', rel_heights = c(1,1.25,2),
  nrow = 3)

```

```

## Warning: Removed 3566 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```

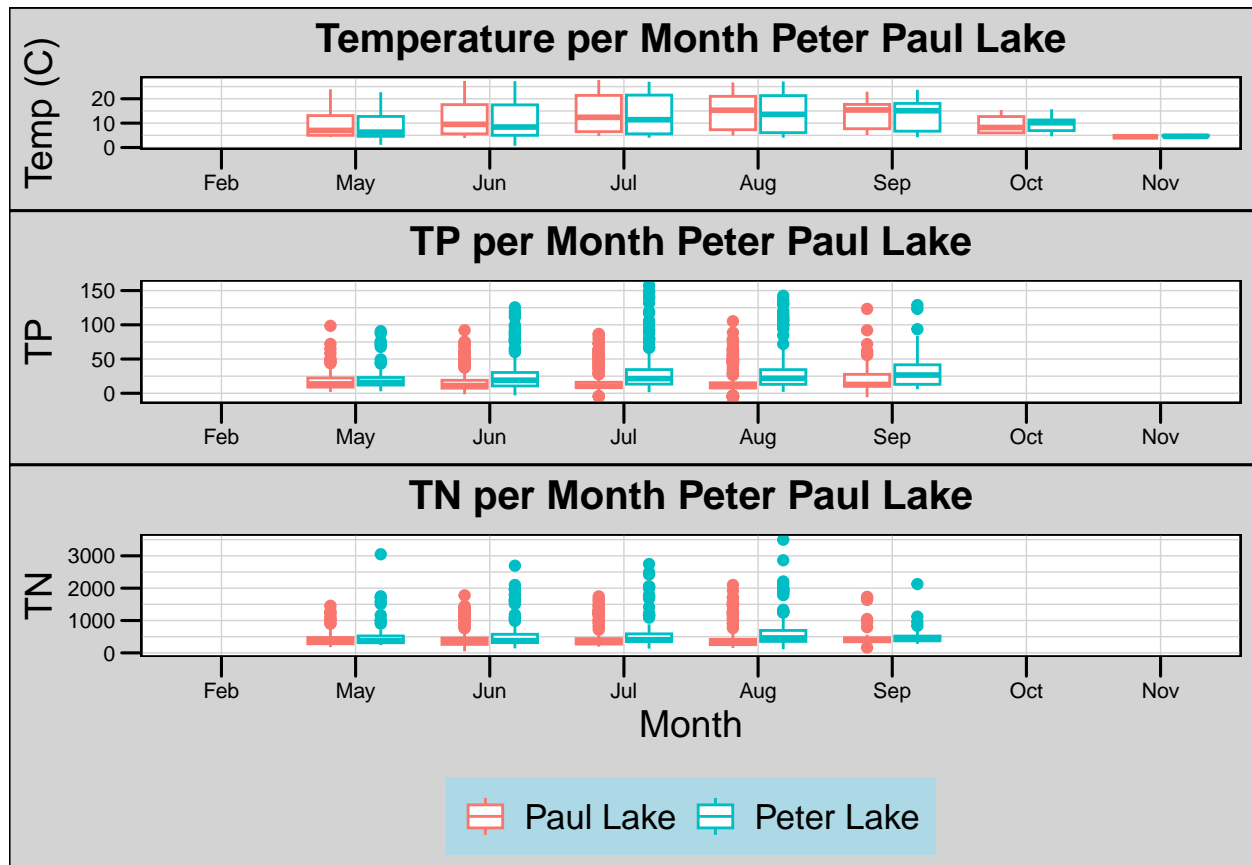
```

## Warning: Removed 20729 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```



```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```



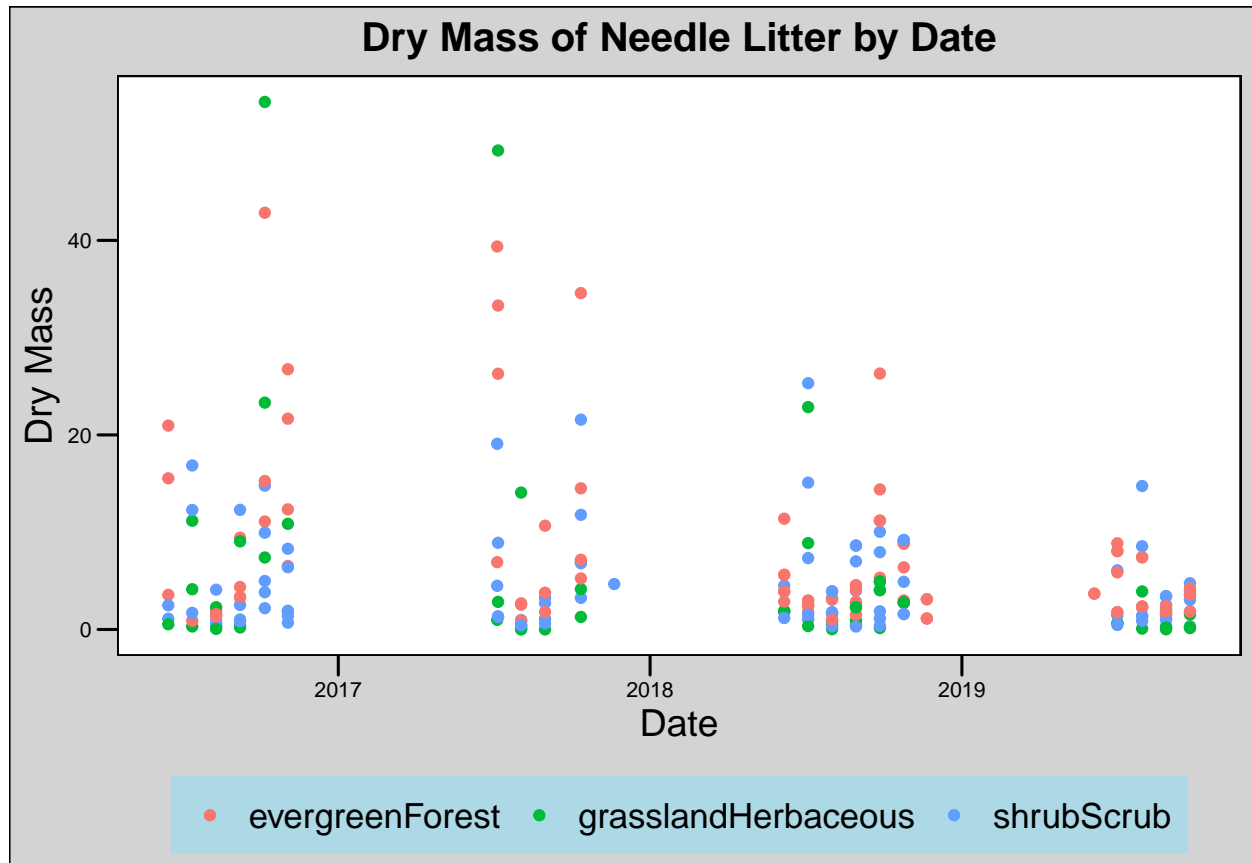
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Total phosphorus seems to have much higher average concentrations at Peter Lake than Paul Lake. Peter Lake also seems to have higher total nitrogen concentrations on average, although the difference is not as great as it is for phosphorus. Phosphorus concentrations seem to increase with temperature at both sites, although that increase is much more profound at Peter Lake than Paul Lake. Nitrogen concentrations seem to be much more uniform across each location with very little variation in relation to temperature.

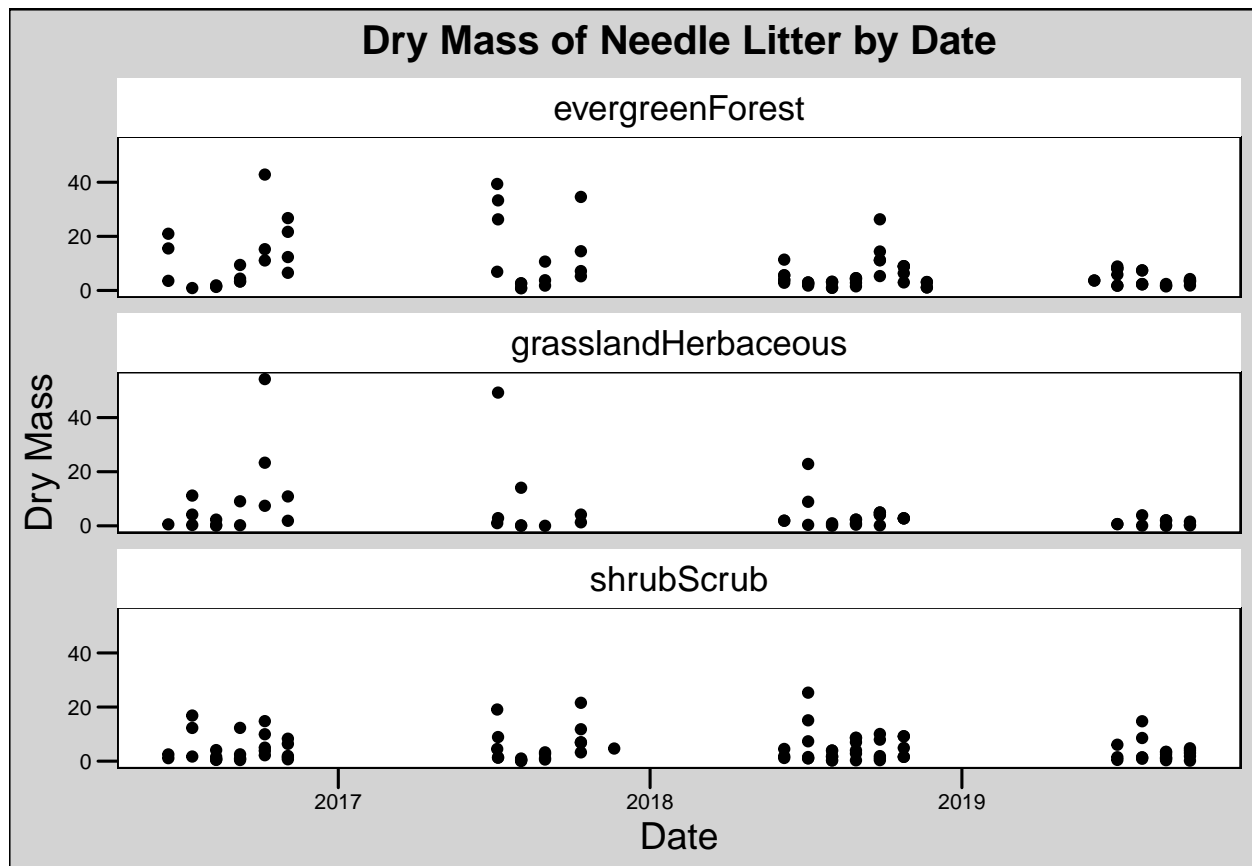
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the “Needles” functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
Needles.Boxplot <-
ggplot(subset(Niwot.Ridge.Litter,
  functionalGroup == "Needles"),
  aes(x=collectDate,
    y=dryMass,
    color=nlcdClass)) +
```

```
geom_point() +
my_theme +
xlab("Date") +
ylab("Dry Mass") +
ggtitle("Dry Mass of Needle Litter by Date")
print(Needles.Boxplot)
```



```
#7
Needles.Facet <-
  ggplot(subset(Niwot.Ridge.Litter,
    functionalGroup == "Needles"),
    aes(x=collectDate,
      y=dryMass)) +
  geom_point() +
  my_theme +
  xlab("Date") +
  ylab("Dry Mass") +
  ggtitle("Dry Mass of Needle Litter by Date") +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(Needles.Facet)
```



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

Answer: I think that the second plot (faceted) is more effective because many of the data points of the three different NLCD classes have very similar values and distributions. This makes it difficult to evaluate each class individually when their data points are often stacked on top of each other.