Assignment 5: Data Visualization

Jackie Van Der Hout

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] "C:/Users/Jackie/Box/Classes Spring 2022/Environmental Data Analytics/Environmental_Data_Analyti
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     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)
NTL_LTER <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", str
NEON_LITTER <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors

#2
class(NTL_LTER$sampledate)
```

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

```
NTL_LTER$sampledate <- as.Date(NTL_LTER$sampledate, format = "%Y-%m-%d")
class(NEON_LITTER$collectDate)
```

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

```
NEON_LITTER$collectDate <- as.Date(NEON_LITTER$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

```
#3
mytheme <- theme_light(base_size = 13) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
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.

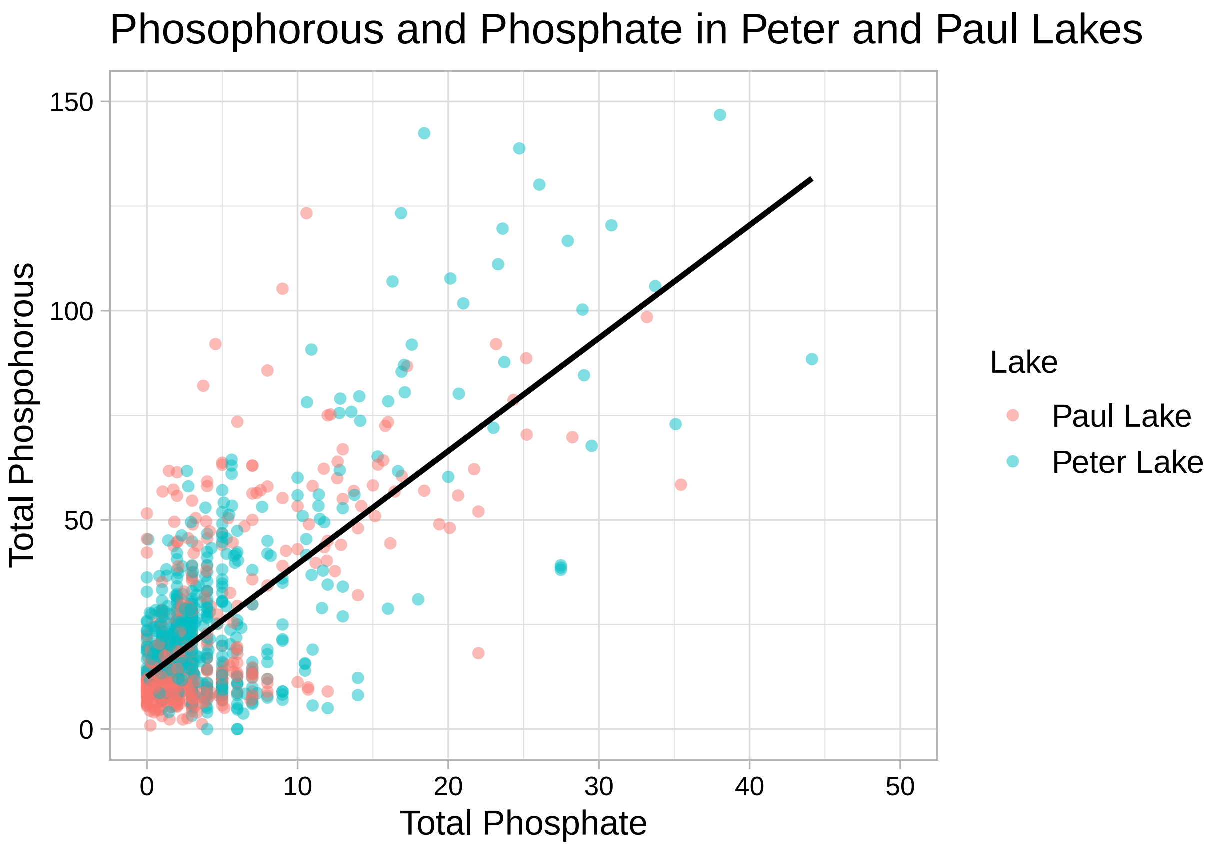
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and `ylim()`).

```
#4
tp_ugXpo4 <- ggplot(NTL_LTER, aes(x = po4, y = tp_ug, color = lakename))+
  geom_point(alpha = 0.5)+
  labs(x = "Total Phosphate", y = "Total Phospohorous", title = "Phosophorous and Phosphate in Peter an
  xlim(0,50)+
  ylim(0, 150)+
  geom_smooth(method = lm, color = "black", se = FALSE)+
  theme(legend.position = "right",
        legend.text = element_text(size = 12), legend.title = element_text(size = 12))
tp_ugXpo4
```

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

```
## Warning: Removed 21948 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21948 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
library(gridGraphics)
```

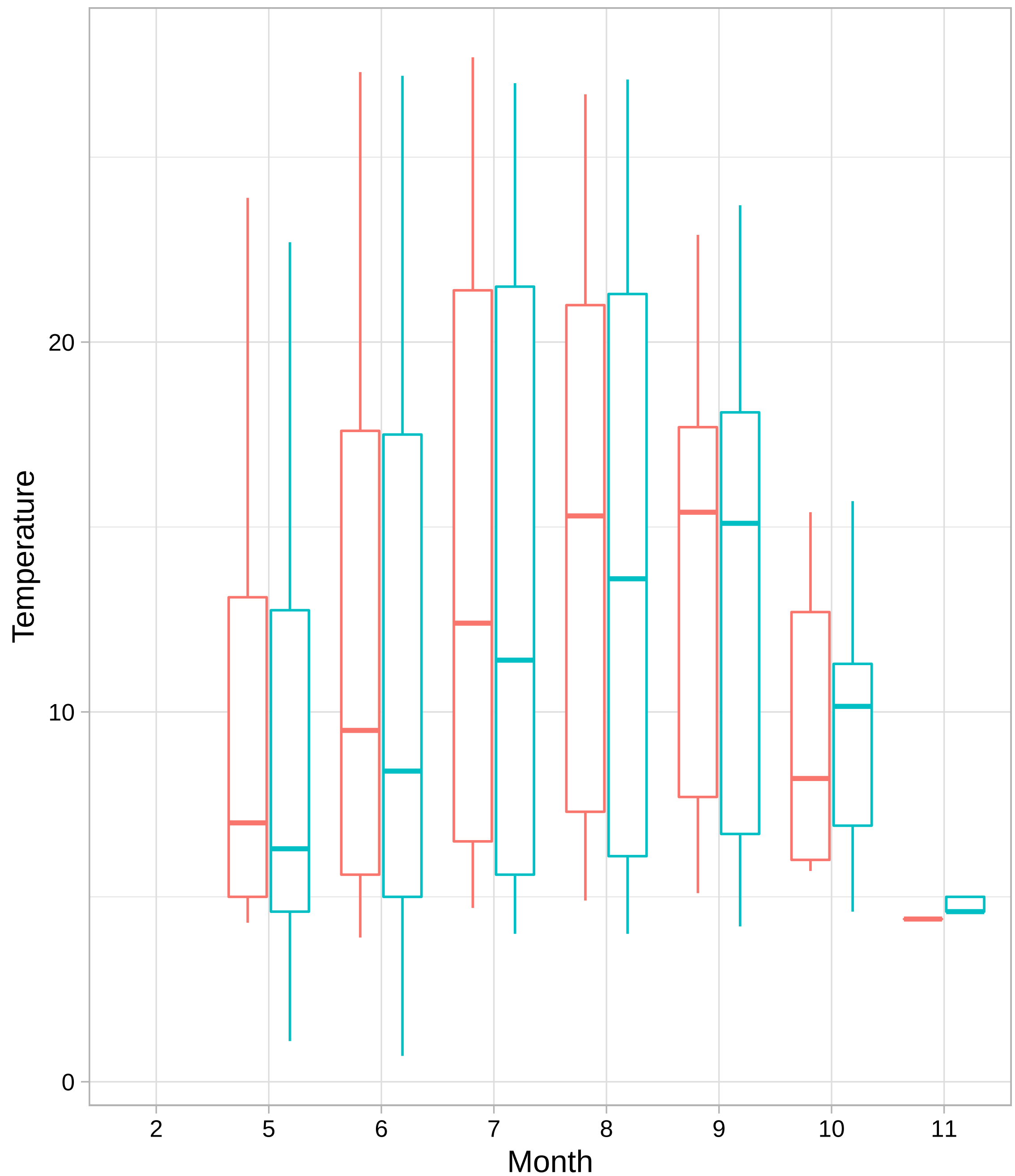
```
## Loading required package: grid
```

```
NTL_LTER$month <- as.factor(NTL_LTER$month)
class(NTL_LTER$month)
```

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

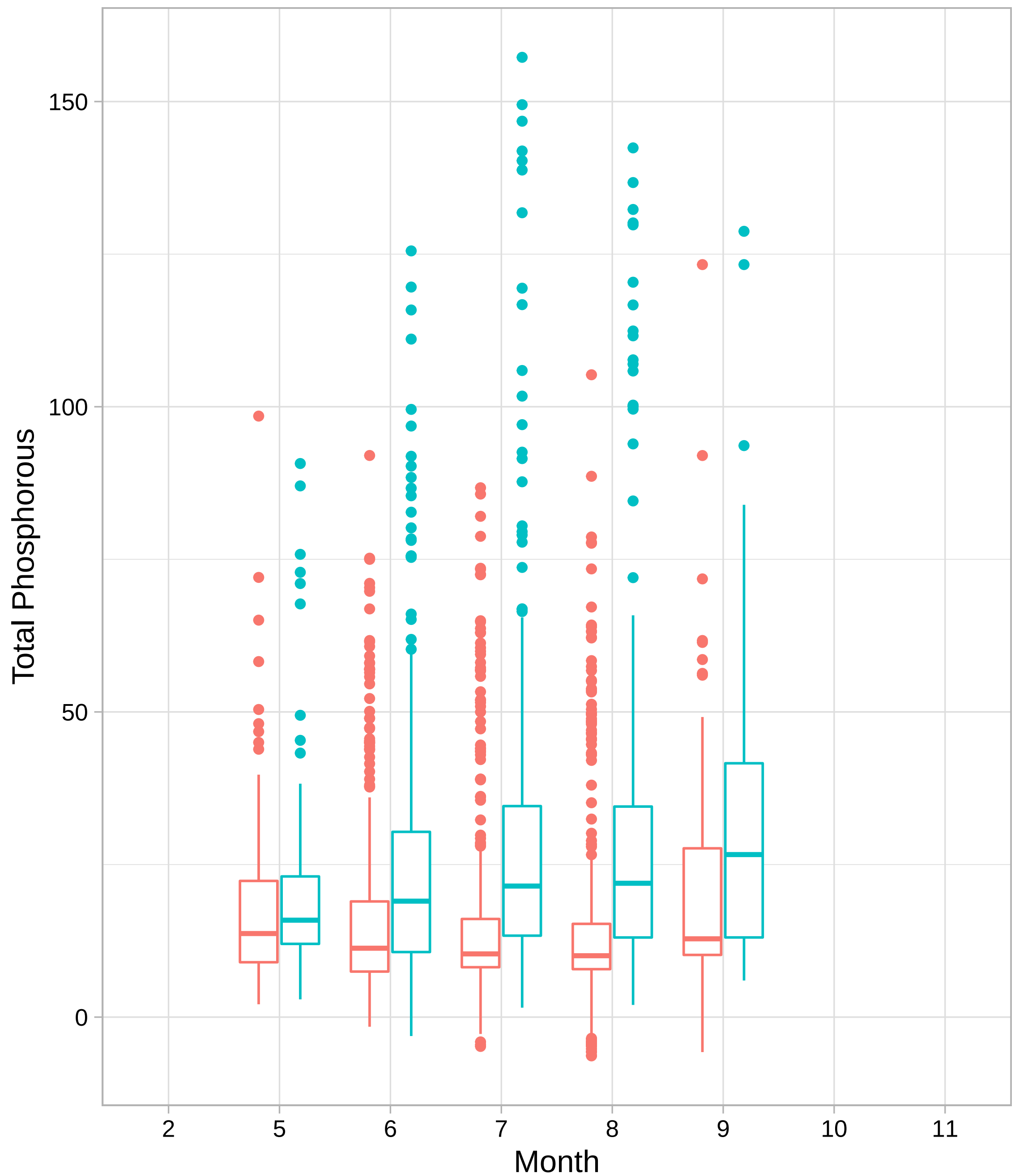
```
tempbox <-ggplot(NTL_LTER, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot()+
  labs(x = "Month", y = "Temperature", color = "Lake")+
  theme(legend.position = "none")
tempbox
```

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



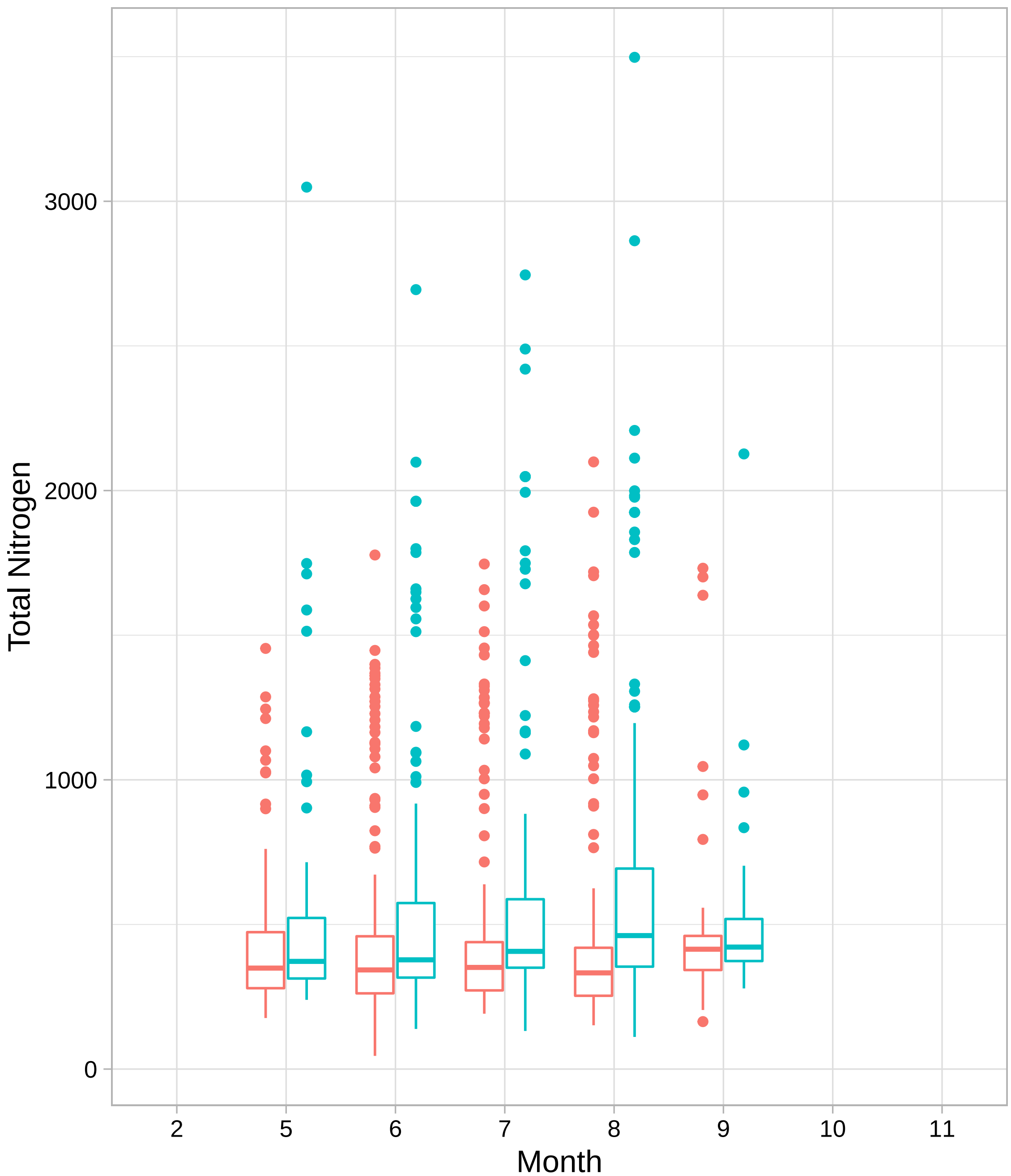

```
TPbox <- ggplot(NTL_LTER, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot()+
  labs(x = "Month", y = "Total Phosphorous", color = "Lake")+
  theme(legend.position = "none")
TPbox
```

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




```
TNbox <- ggplot(NTL_LTER, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot()+
  labs(x = "Month", y = "Total Nitrogen", color = "Lake")+
  theme(legend.position = "none")
TNbox
```

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




```
boxes <- plot_grid(
  tempbox + theme(legend.position = "none"),
  TPbox + theme(legend.position = "none"),
  TNbox + theme(legend.position = "none"),
  align = 'vh',
  hjust = -1,
  nrow = 1
)
```

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

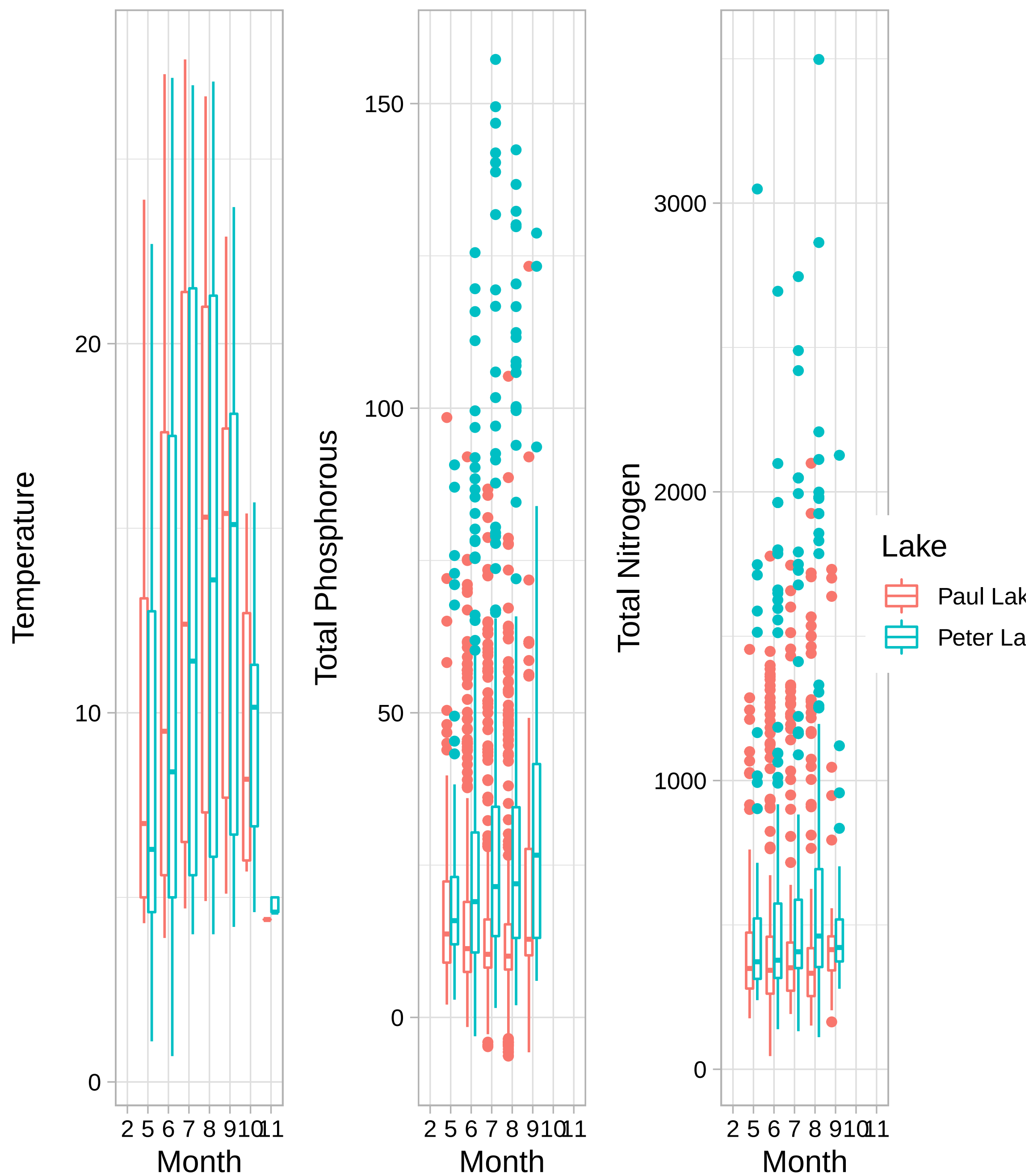
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).

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

```
fakelegend <- get_legend(tempbox +
  theme(legend.position = "right", legend.box.margin = margin(0, 0, 0, 1))
  )
```

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

```
plot_grid(boxes, fakelegend, ncol = 2, rel_widths = c(3, .4))
```



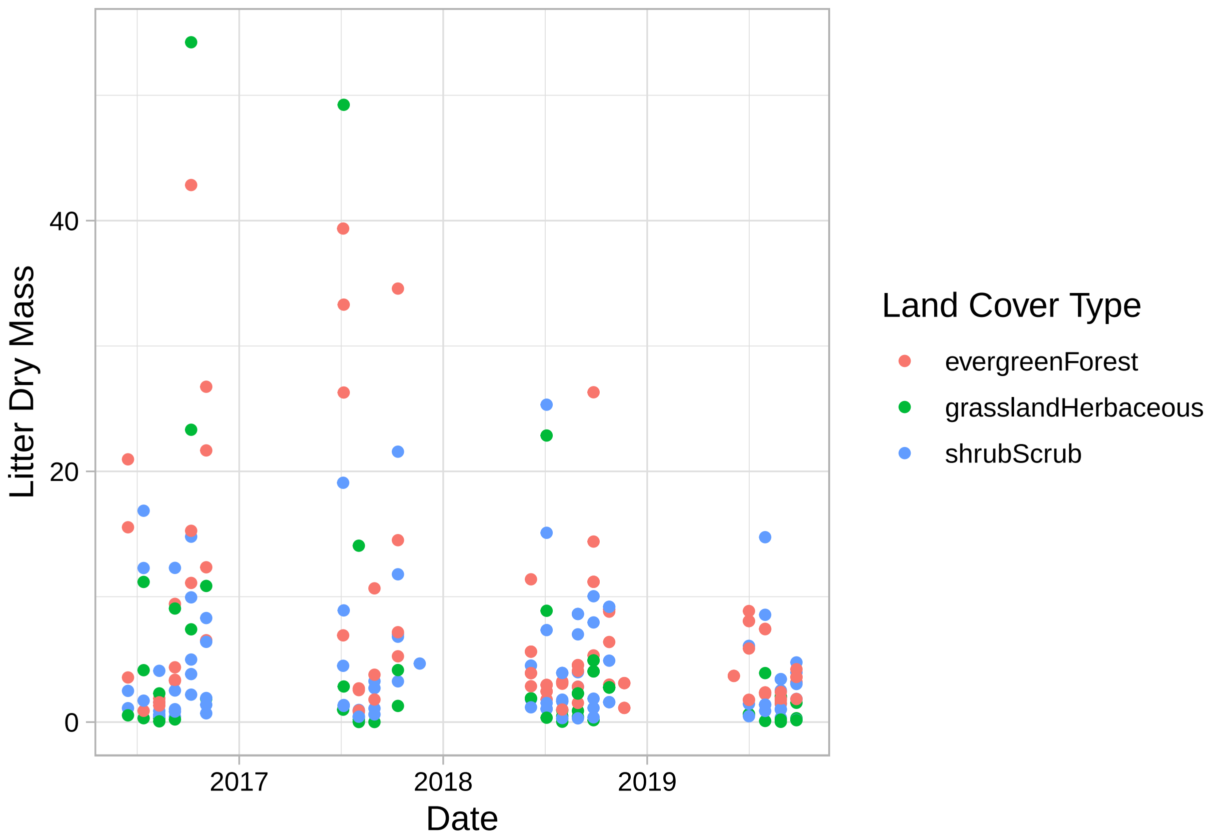
```
#this seems like a clunky way to do it but it more or less works
#however the output plot only looks good when expanded to full size
```

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

> Answer: Temperature has a wider range of variability than Phosphorous and Nitrogen, and is also the most sensitive to seasonal changes in both lakes. Both Peter and Paul lakes appear to display similar seasonal changes in temperature, with similar standard error ranges. However, Paul Lake seems to display a slightly higher mean temperature than Peter Lake until October, when Peter lake cools down more slowly. Inversely, Peter Lake appears to have a higher concentration of both Phosphorous and Nitrogen on average, as well as higher outlier peak values in nutrient concentration than Paul Lake. Nutrient concentrations appear to be seasonally highest in both lakes and in both nutrient categories during the late summer / early fall.

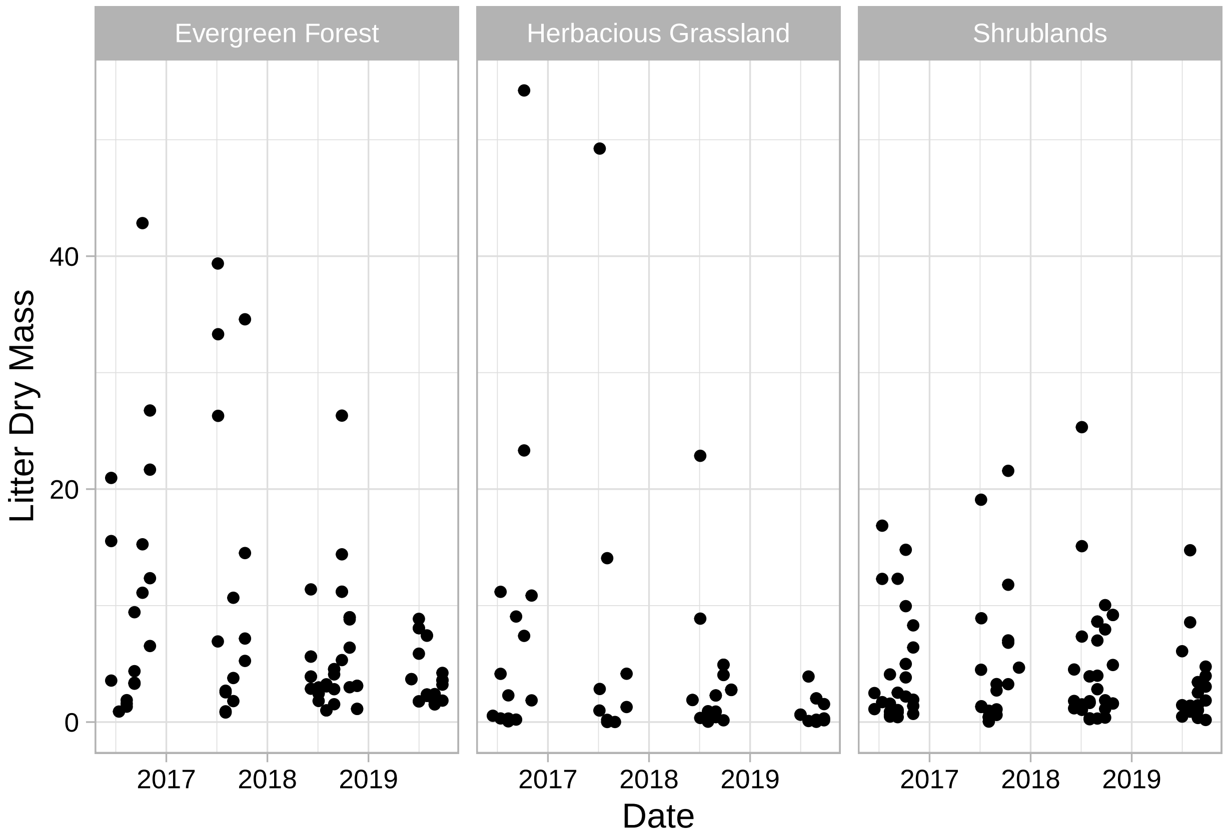
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes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

```
#6
needleplot <- ggplot(subset(NEON_LITTER, functionalGroup == "Needles"), aes(x = collectDate, y = dryMass
  geom_point()+
  labs(x = "Date", y = "Litter Dry Mass", color = "Land Cover Type")+
  theme(legend.position = "right")
needleplot
```




```
#7
NEON_LITTER$nlcdClass <- factor(NEON_LITTER$nlcdClass, levels = c("evergreenForest", "grasslandHerbaceo
                  labels = c("Evergreen Forest", "Herbacious Grassland", "Shrublands"))

needleplot2 <- ggplot(subset(NEON_LITTER, functionalGroup == "Needles"), aes(x = collectDate, y = dryMas
  geom_point()+
  labs(x = "Date", y = "Litter Dry Mass")+
  theme(legend.position = "right")+
  facet_wrap(vars(nlcdClass), nrow = 1)
needleplot2
```



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

> Answer: The facet wrapped plot in #7 provides an easier comparison of litter dry mass over both land cover type and time because of the side-by-side comparison with the same axes, instead of having all of the datapoints within one plot.