Assignment 5: Data Visualization

Claire Pajka

Fall 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. 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);library(lubridate);library(here); library(cowplot)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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
```

```
## here() starts at C:/Users/cepaj/OneDrive/Documents/EDE_Fall2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here()
```

```
## [1] "C:/Users/cepaj/OneDrive/Documents/EDE_Fall2023"
```

```
Lake_Chemistry_Nutrients_PeterPaul_Processed <- read.csv(
  file=here(
    "C:/Users/cepaj/OneDrive/Documents/EDE_Fall2023/Data/Processed_KEY/Processed_KEY/NTL-LTER_Lake_Chemi
  stringsAsFactors = TRUE)
Lake_ChemistryPhysics_PeterPaul_Processed <- read.csv(
  file=here(
    "C:/Users/cepaj/OneDrive/Documents/EDE_Fall2023/Data/Processed_KEY/Processed_KEY/NTL-LTER_Lake_Nutri
  stringsAsFactors = TRUE)
Litter_mass_trap_processed <- read.csv(
  file=here(
    "C:/Users/cepaj/OneDrive/Documents/EDE_Fall2023/Data/Processed_KEY/Processed_KEY/NEON_NIWO_Litter_ma
  stringsAsFactors = TRUE)

#2
class(Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
```

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

```
Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate <-
  ymd(Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
class(Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
```

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

```
Litter_mass_trap_processed$collectDate <-
  ymd(Litter_mass_trap_processed$collectDate)
class(Litter_mass_trap_processed$collectDate)
```

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

```
class(Lake_ChemistryPhysics_PeterPaul_Processed$sampledate)
```

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

```
Lake_ChemistryPhysics_PeterPaul_Processed$sampledate <-
  ymd(Lake_ChemistryPhysics_PeterPaul_Processed$sampledate)
class(Lake_ChemistryPhysics_PeterPaul_Processed$sampledate)
```

```
## [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
mytheme <- theme_classic(base_size=12)+
  theme(
    plot.title= element_text(size= 12, color = "darkblue",
                             face="bold", hjust = 0.5),
    axis.text = element_text(size = 12, color = "black"),
    legend.position = 'bottom',
    legend.background = element_blank(),
    legend.box.background = element_rect(colour = "black"),
    plot.background = element_rect(color = "black"),
    axis.line = element_line(size = 0.65, color = "black"))
```

```
## 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.
```

```
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
Peterpaul_PO4plot <-
  ggplot(Lake_Chemistry_Nutrients_PeterPaul_Processed,
         aes(x = tp_ug, y = po4, color = lakename))+
```

```
  geom_point()+
  geom_smooth(method = lm, size = .75, color = "black")+
  xlim(0,150)+
  ylim(0,40)+
  labs(title="Total Phosphorus and Phosphate in Peter Lake and Paul Lake",
       fontface = "bold",
       color = "Lake Name")+
  ylab("Phosphate")+
  xlab("Total Phosphorus")+
  scale_shape_manual(values = c(15,17))
```

```
## 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.
```

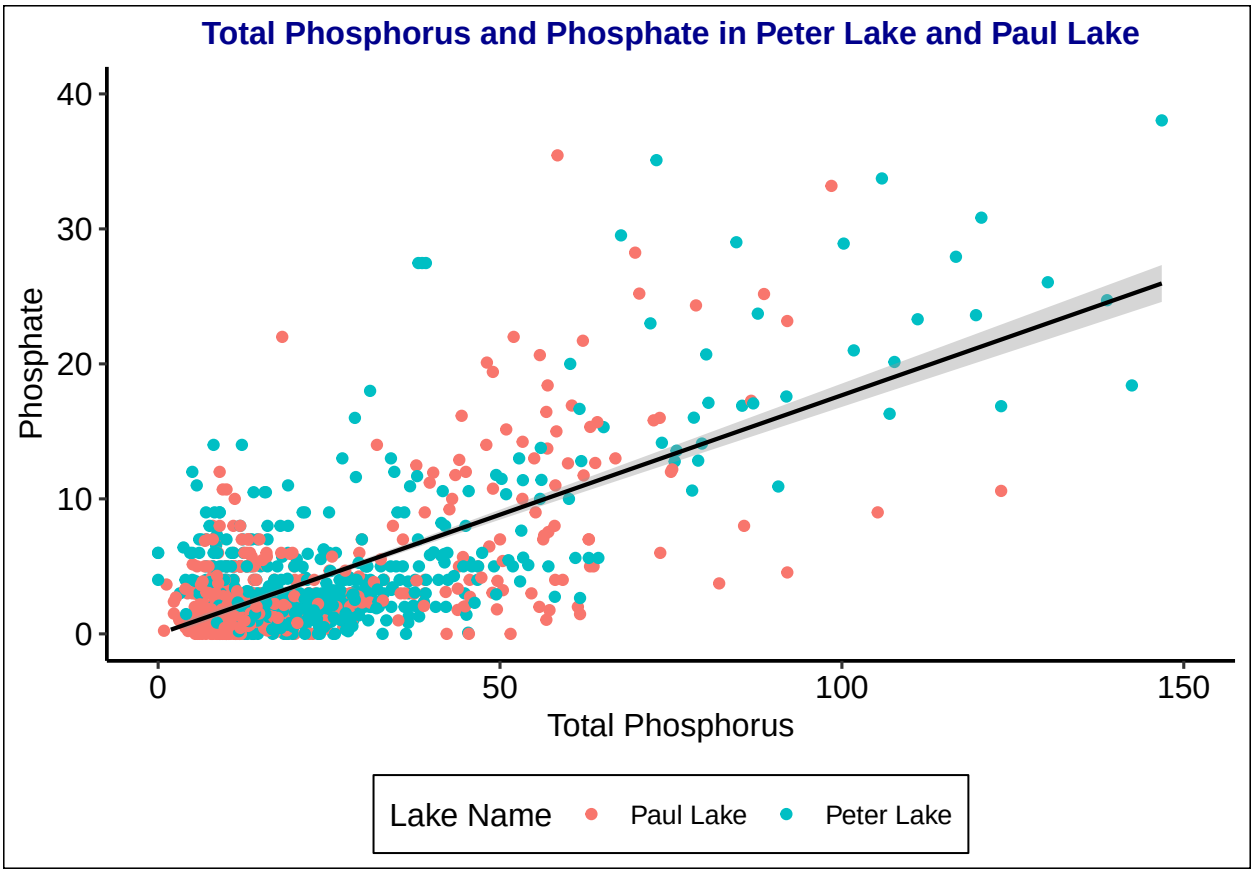
```
print(Peterpaul_PO4plot)
```

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

## Warning: Removed 21949 rows containing non-finite values ('stat_smooth()').

## Warning: Removed 21949 rows containing missing values ('geom_point()').

## Warning: Removed 1 rows containing missing values ('geom_smooth()').
```

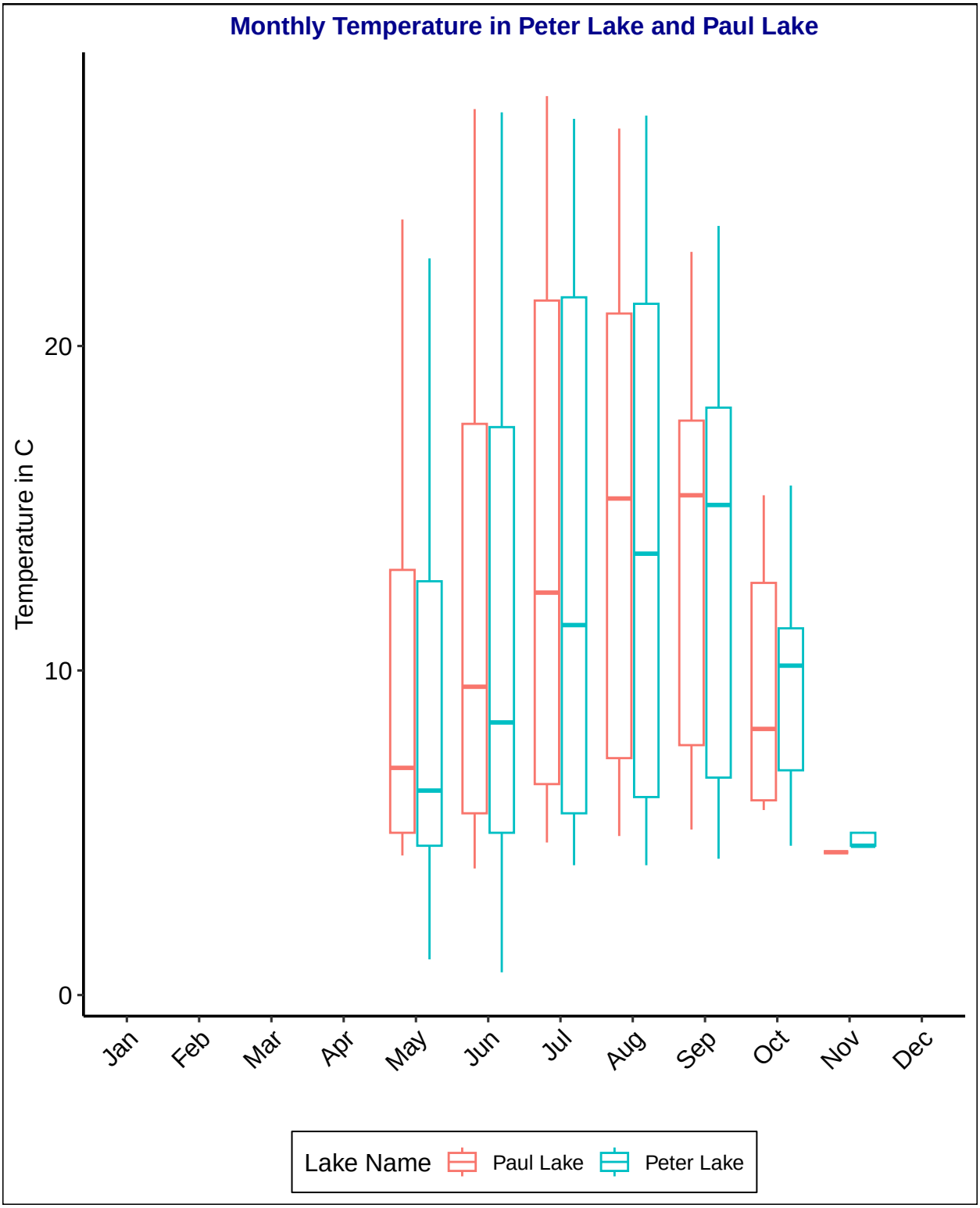


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
Monthly_temp_plot <-
  ggplot(Lake_Chemistry_Nutrients_PeterPaul_Processed,
         aes(x=factor(month, levels = 1:12, labels = month.abb),
             y = temperature_C))+
  geom_boxplot(aes(color = lakename))+
  labs(title="Monthly Temperature in Peter Lake and Paul Lake",
       fontface = "bold",
       color = "Lake Name",
       x= "",
       y= "Temperature in C")+
  scale_x_discrete(name="", drop=FALSE)+
  theme(axis.text.x = element_text(angle = 45, hjust=1))
print(Monthly_temp_plot)
```

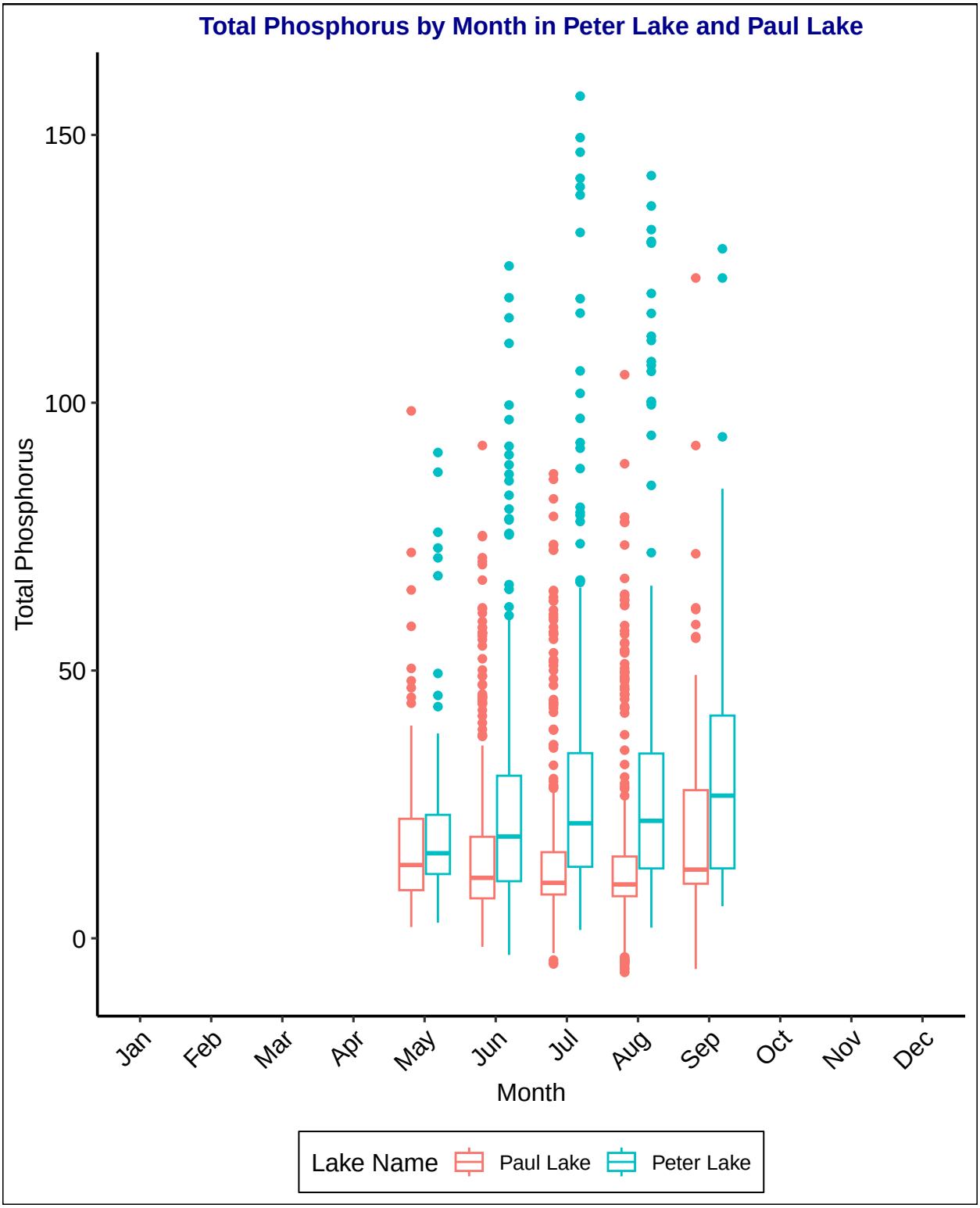
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```




```
Monthly_Tp_plot <-
  ggplot(Lake_Chemistry_Nutrients_PeterPaul_Processed,
         aes(x=factor(month, levels = 1:12, labels = month.abb),
             y = tp_ug))+
  geom_boxplot(aes(color = lakename))+
```

```
  labs(title="Total Phosphorus by Month in Peter Lake and Paul Lake",
       fontface = "bold", color = "Lake Name",
       x= "Month",
       y= "Total Phosphorus")+
  scale_x_discrete(name="Month", drop=FALSE)+
  theme(axis.text.x = element_text(angle = 45, hjust=1))
print(Monthly_Tp_plot)
```

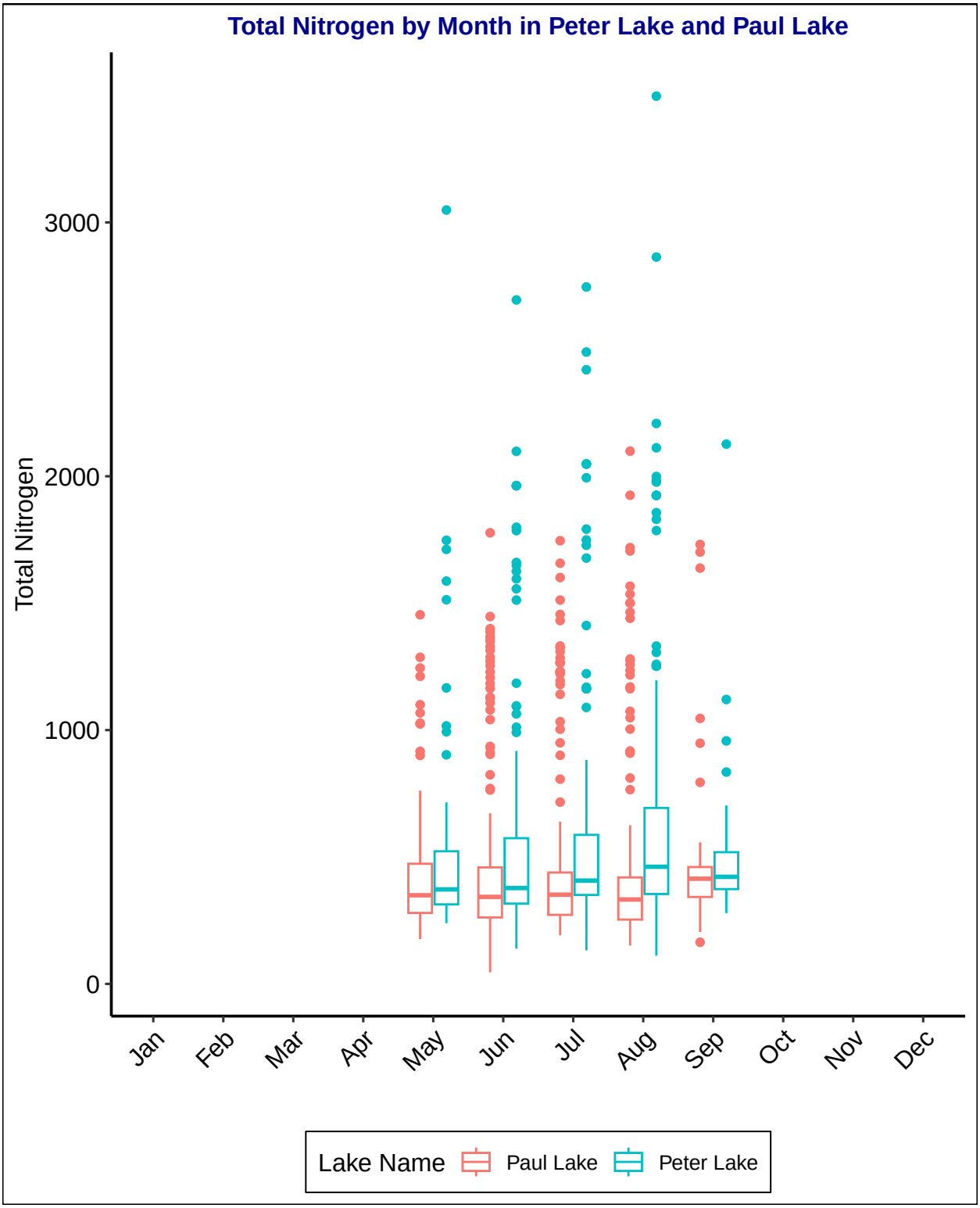
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
```




```
Monthly_Tn_plot <-
  ggplot(Lake_Chemistry_Nutrients_PeterPaul_Processed,
         aes(x=factor(month, levels = 1:12, labels = month.abb),
             y = tn_ug))+
  geom_boxplot(aes(color = lakename))+
```

```
  labs(title="Total Nitrogen by Month in Peter Lake and Paul Lake",
       fontface = "bold", color = "Lake Name",
       x= "",
       y= "Total Nitrogen")+
  scale_x_discrete(name="", drop=FALSE)+
  theme(axis.text.x = element_text(angle = 45, hjust=1))
print(Monthly_Tn_plot)
```

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




```
combined_plot <-
  plot_grid(
    Monthly_temp_plot +theme(legend.position = "none"),
    Monthly_Tn_plot+ theme(legend.position = "none"),
    Monthly_Tp_plot+ theme(legend.position = "bottom"),
```

```
    ncol=1, nrow=3, align = 'v', axis = 'l', rel_heights = c(1,1,1.4),
  theme(
    axis.text.x = element_text(angle = 45, hjust=1)))
```

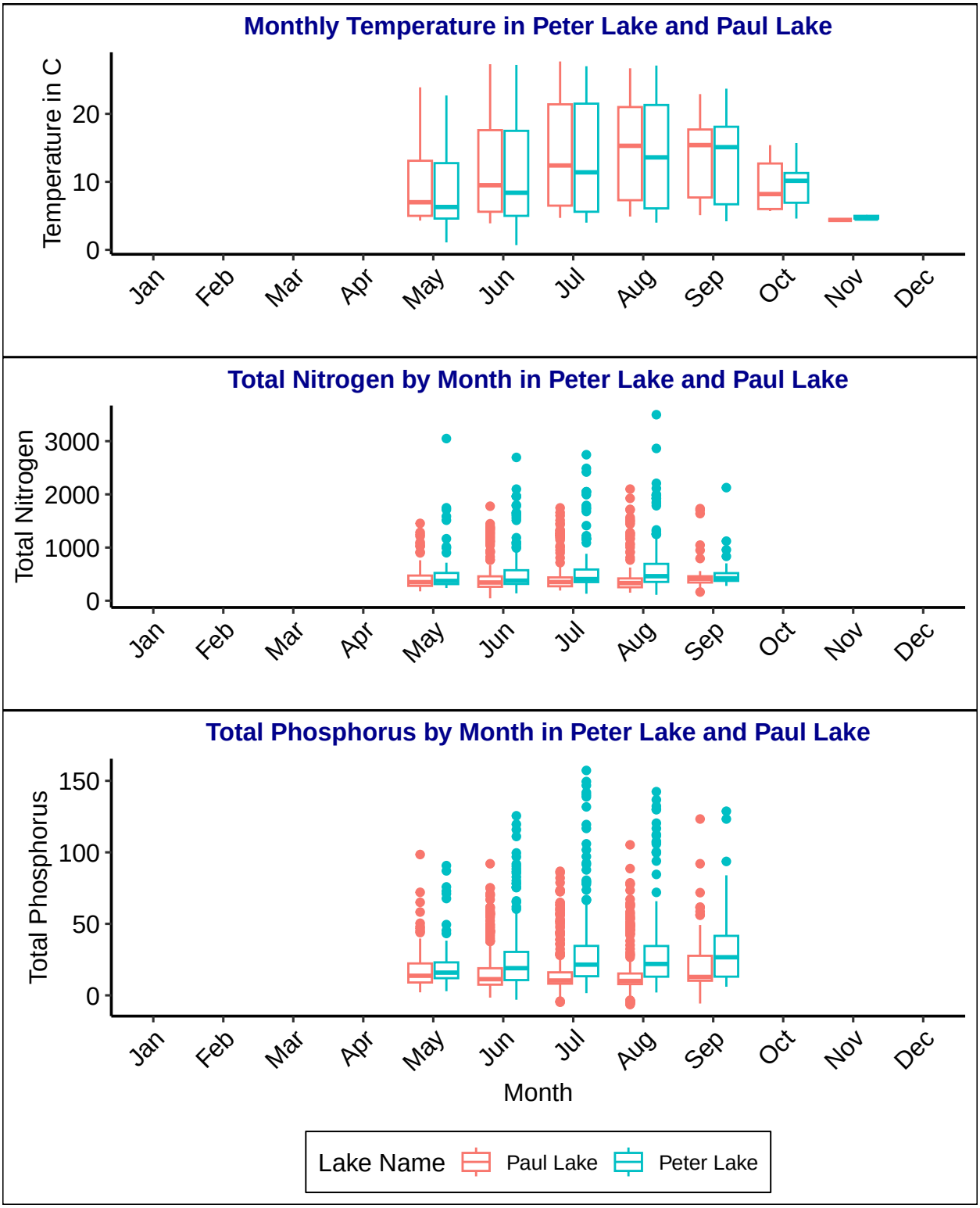
```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).
```

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

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

```
## Warning in as_grob.default(plot): Cannot convert object of class themegg into a
## grob.
```

```
print(combined_plot)
```



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

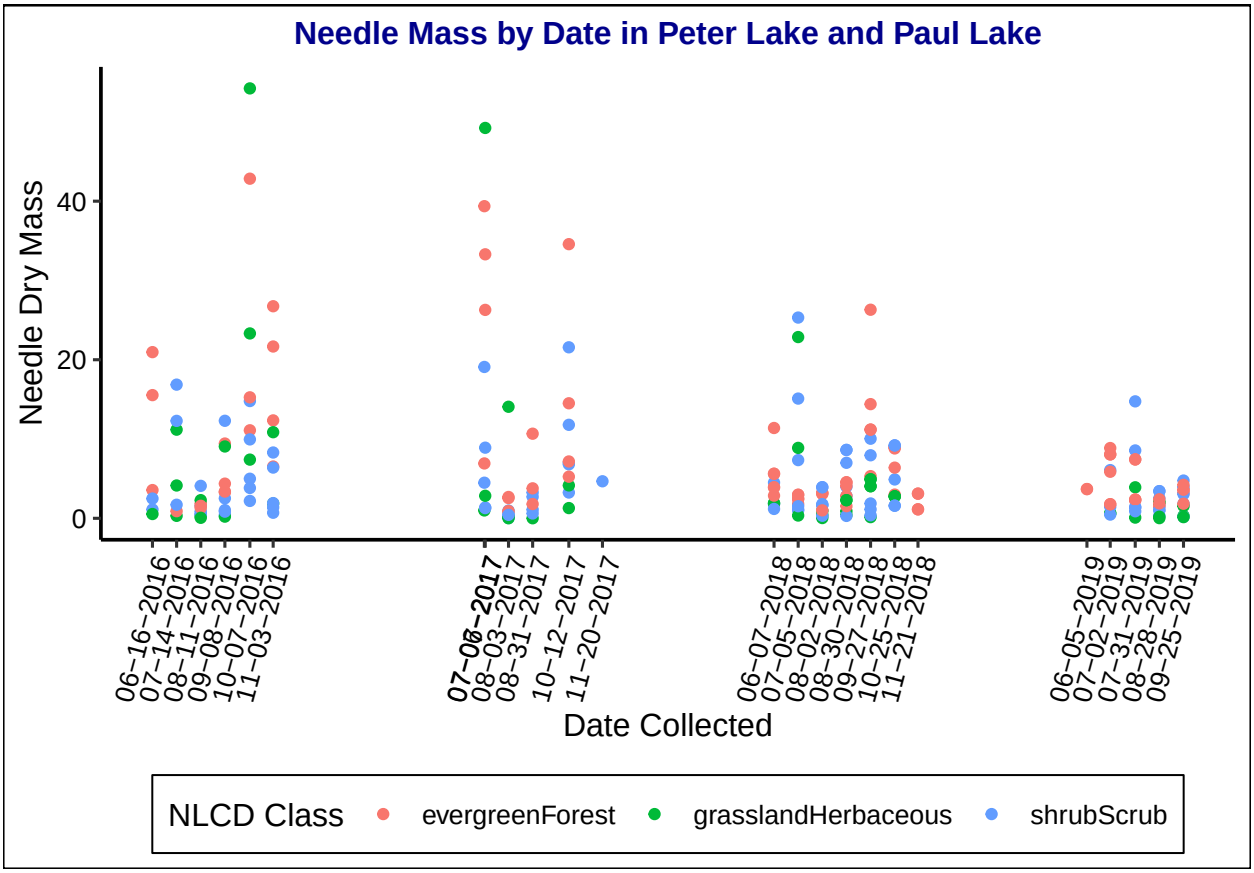
Answer: From the number and further distance of outliers, it seems that the variability in Nitrogen and Phosphorus increases from June to August. This could be due to many reasons, including summer storm runoff bringing nitrogen and phosphorus into the lakes in the form of

agricultural or lawn fertilizers.The monthly temperatures seem to be relatively close, though Paul lake tends to be slightly warmer in May through Septemeber, though is notably less warm than Peter Lake in October. Because both Peter and Paul lake are dimictic, this is not an unsurprising finding (around October or November, the lakes circulate colder water up, and warmer water down. In this instance, it appears that Paul Lake mixed before Peter Lake did.

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)

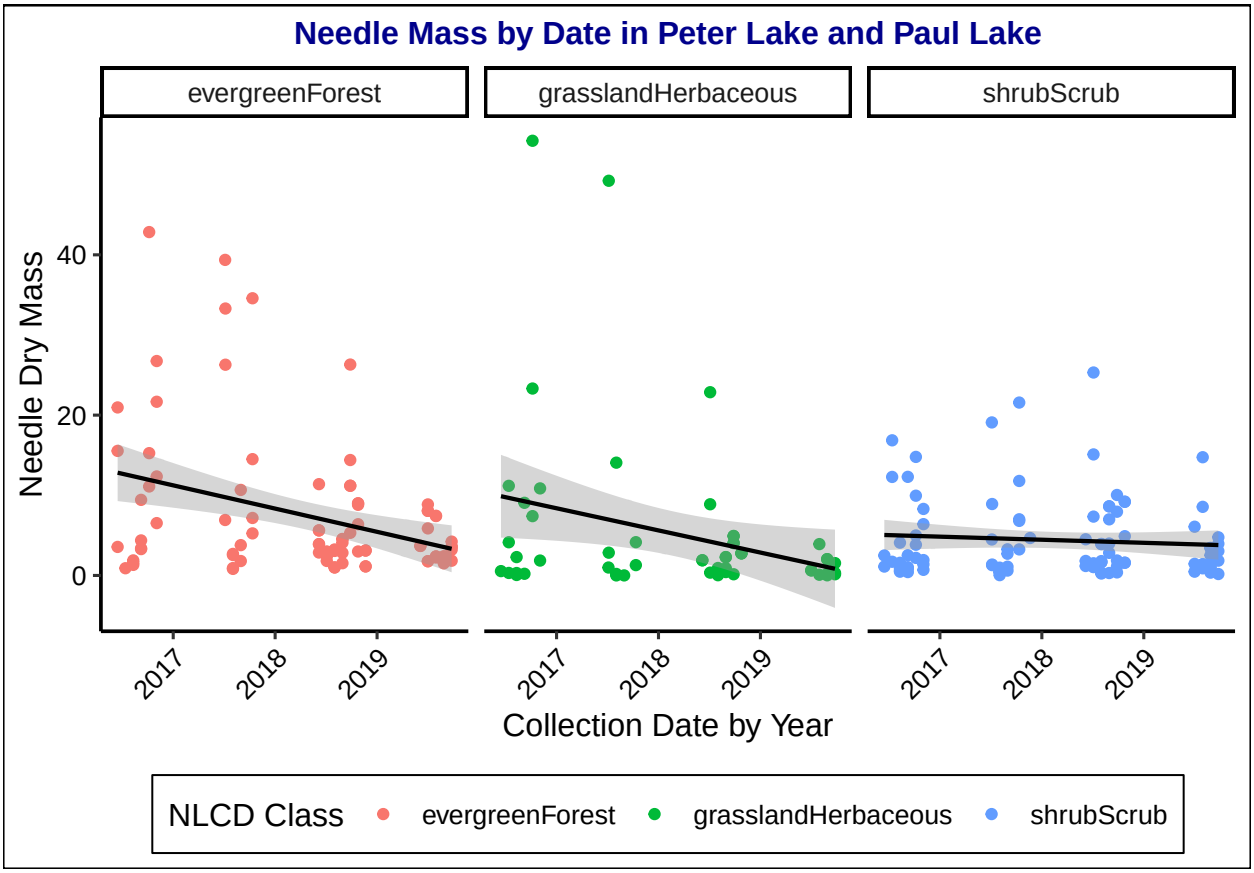
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
Needle_Weight_drymass_plot <- Litter_mass_trap_processed %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate,
             y=dryMass,
             color = nlcdClass))+
  geom_point()+
  labs(title="Needle Mass by Date in Peter Lake and Paul Lake",
       fontface = "bold", color = "NLCD Class")+
  ylab("Needle Dry Mass")+
  xlab("Month")+
  scale_x_date(name="Date Collected", date_labels ="%m-%d-%Y", breaks = unique(Litter_mass_trap_processe
  theme(axis.text = element_text(size = 10))+
  theme(axis.text.x = element_text(angle = 75, hjust=1))
print(Needle_Weight_drymass_plot)
```




```
#7
Needlemass_facet <-
  ggplot(filter(Litter_mass_trap_processed, functionalGroup == "Needles"),
       aes(y=dryMass,
           x=collectDate,
           color = nlcdClass)) +
  facet_wrap(vars(nlcdClass), nrow=1)+
  geom_point()+
  geom_smooth(method = lm, color = "black", size = .75, se=TRUE)+
  labs(title="Needle Mass by Date in Peter Lake and Paul Lake",
       fontface = "bold", color = "NLCD Class")+
  ylab("Needle Dry Mass")+
  xlab("Collection Date by Year")+ #separating by month makes the graphs cluttered & hard to interpret
  scale_x_date(
    name="Collection Date by Year", date_labels ="%Y")+
  theme(axis.text = element_text(size = 10))+
  theme(axis.text.x = element_text(angle = 45, hjust=1))
print(Needlemass_facet)
```

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



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

Answer: I think graph 7 is more effective, because it displays the trends in each different land class area more clearly, and you can see trends over time for each individual land use. For example, in graph 7, you can easily see that the amount of Needle dry mass steadily decreases from 2017 to 2020 by inserting a trendline for each land class, but in graph 6, even though the land use types are distinct by color, a trendline is not able to adequately show trends.