Assignment 5: Data Visualization

Lu Liu

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
#load packages
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/EDA_Spring2025_ForkCeleste
```

```
library(cowplot)
```

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

```
#get working directory
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025_ForkCeleste"
```

```
#read required data files
peter_paul_data <-
  read.csv(
    "Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
    stringsAsFactors = TRUE)

niwot_litter_data <- read.csv(
    "Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
    stringsAsFactors = TRUE)
#2
#check the category of date
class(peter_paul_data$sampledate)
```

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

```
class(niwot_litter_data$collectDate)
```

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

```
#change the format of date
peter_paul_data$sampledate <- as.Date(peter_paul_data$sampledate,format = "%Y-%m-%d")
niwot_litter_data$collectDate <- as.Date(niwot_litter_data$collectDate,format = "%Y-%m-%d")
```

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
#load package
library(ggplot2)
#define a custom theme
custom_theme <- theme(
  #customize plot backgroun
  plot.background = element_rect(fill="lightgray",color="black"),
  #light gray background with black border
  #customize plot title
  plot.title=element_text(size=20,face="bold", color="darkblue", hjust=0.5),
  #centered, bold, dark blue title
  #customize axis labels
  axis.title.x=element_text(size=14,color="darkred"),#dark red x-axis label
  axis.title.y=element_text(size=14,color="darkgreen"),
  #customize axis ticks and gridlines
  axis.ticks=element_line(color="black",size=1),#black, thicker ticks
  legend.position="bottom", #move legend to the bottom
  legend.background = element_rect(fill="white",color="black"),
  #white background with black border
  legend.title=element_text(face="bold",size=12), #bold legend title
  legend.text=element_text(size=10) #smaller legend text
  )
```

```
## 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.
```

```
#set the custom theme as the default
theme_set(custom_theme)
```

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.

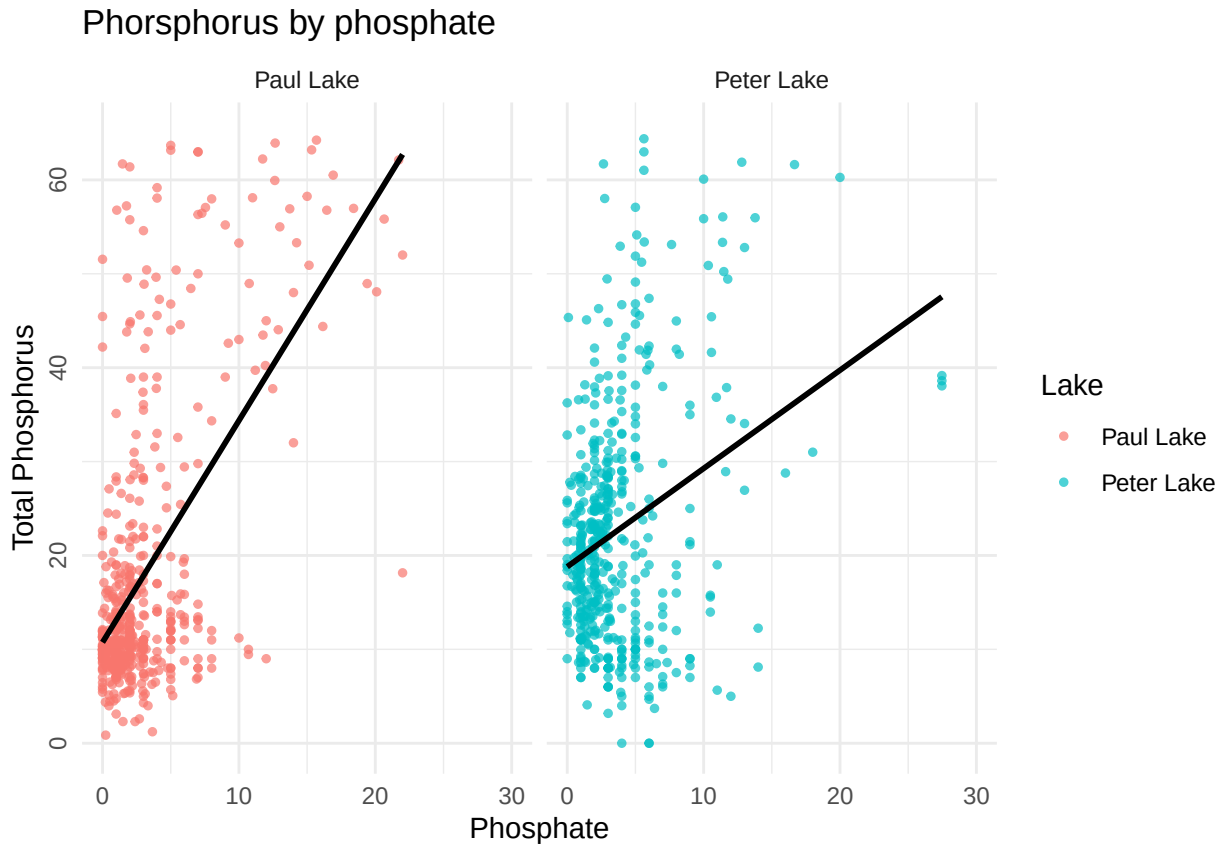
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
ggplot(peter_paul_data,aes(x=po4,y=tp_ug,color=lakename))+
  geom_point(size=1,alpha=0.7)+ #scatterplot with transparency
  geom_smooth(method="lm",formula = y ~ x,color="black",se=FALSE)+ #line of best fit black
  labs(
    title="Phorsphorus by phosphate ",
    x="Phosphate",
    y="Total Phosphorus",
```

```
    color="Lake",
  )+
   xlim(0, 30) +  # Adjust x-axis limits to hide extreme values
   ylim(0, 65)+ # Adjust y-axis limits to hide extreme values
  facet_wrap(~lakename, ncol=2)+# Create separate plots for Peter and Paul lakes
  theme_minimal() +  # use simple theme
  theme(
    axis.text.y = element_text(angle = 90, hjust = 0.5)  # rotate y-axis
  )
```

```
## Warning: Removed 22001 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

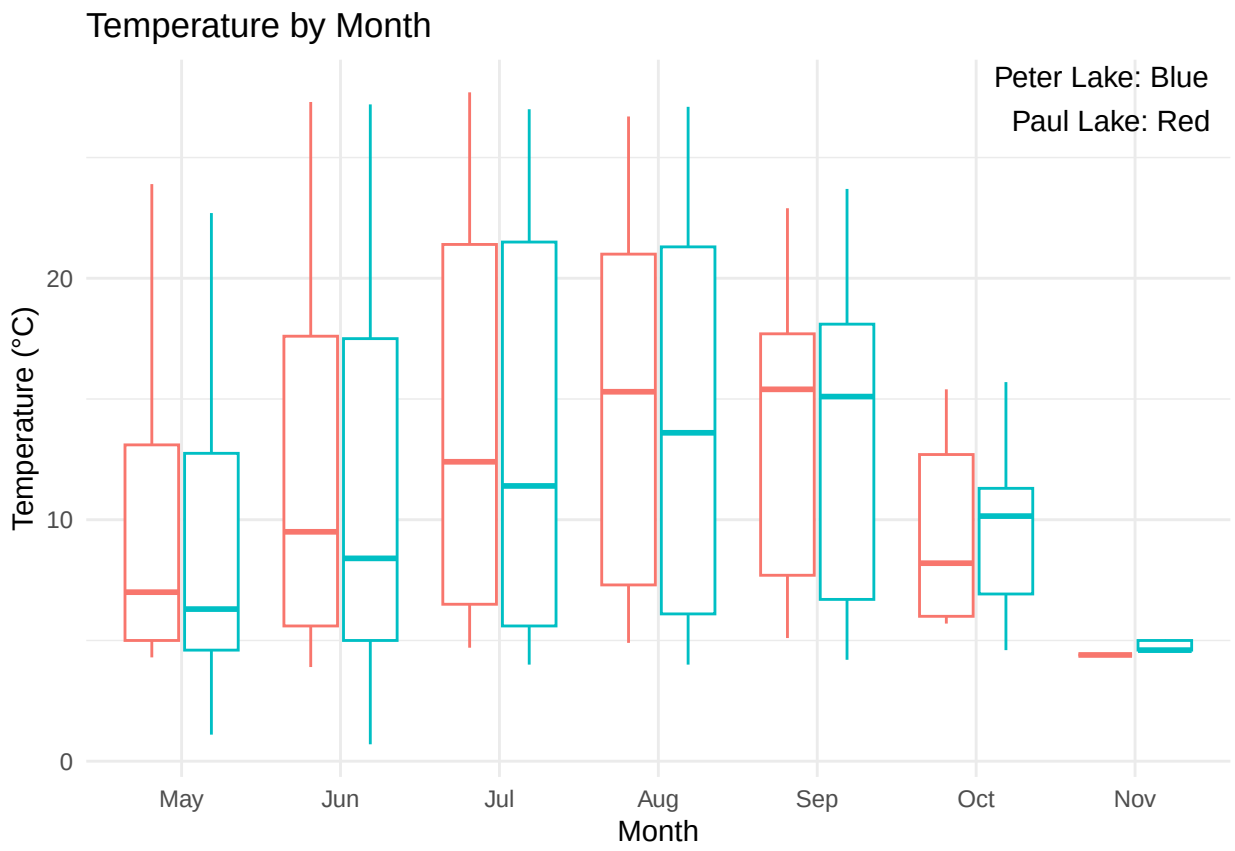
```
## Warning: Removed 22001 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



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.

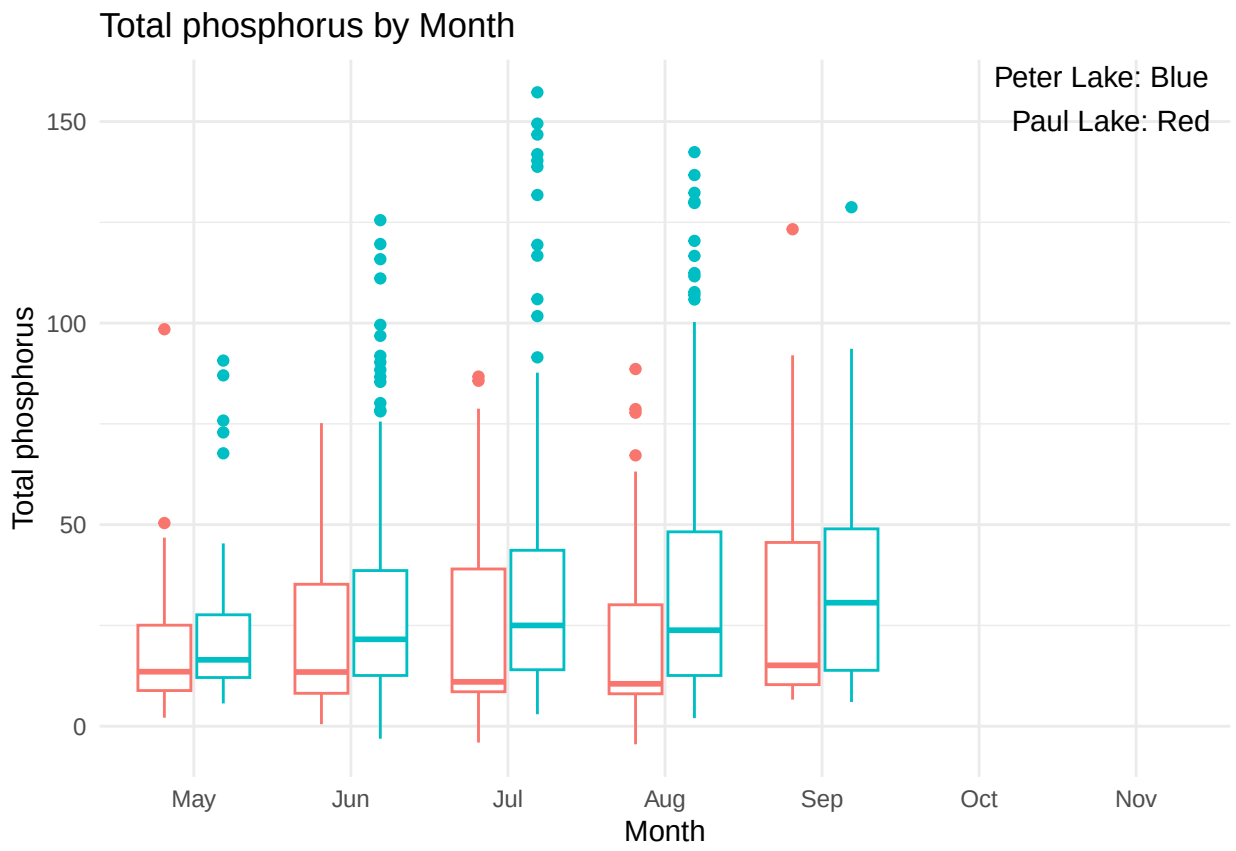
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a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

```r
#5
#load package
library(ggplot2)
library(dplyr)
library(cowplot)

#convert month to a factor
peter_paul_data_1 <-
  peter_paul_data %>%
  mutate(
    month=factor(month.abb[month],levels=month.abb))
#check result
view(peter_paul_data_1$month)

#remove missing values for temperature
peter_paul_data_1 <-
  peter_paul_data_1 %>%
    filter(!is.na(temperature_C))

#create boxplot for temperature
plot_temp <- ggplot(peter_paul_data_1, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  labs(
    title = "Temperature by Month",
    x = "Month",
    y = "Temperature (°C)",
    color = "Lakename"
  ) +
  theme_minimal() +
  theme(legend.position = "none")+  # Hide legend for individual plots
  annotate(
    "text",
    x = Inf, y = Inf,  # Position at the top-right corner
    label = "Peter Lake: Blue\nPaul Lake: Red",  # Text to display
    hjust = 1.1, vjust = 1.1,  # Adjust position
    color = "black", size = 4  # Customize text appearance
  )

# Display the plot
print(plot_temp)
```



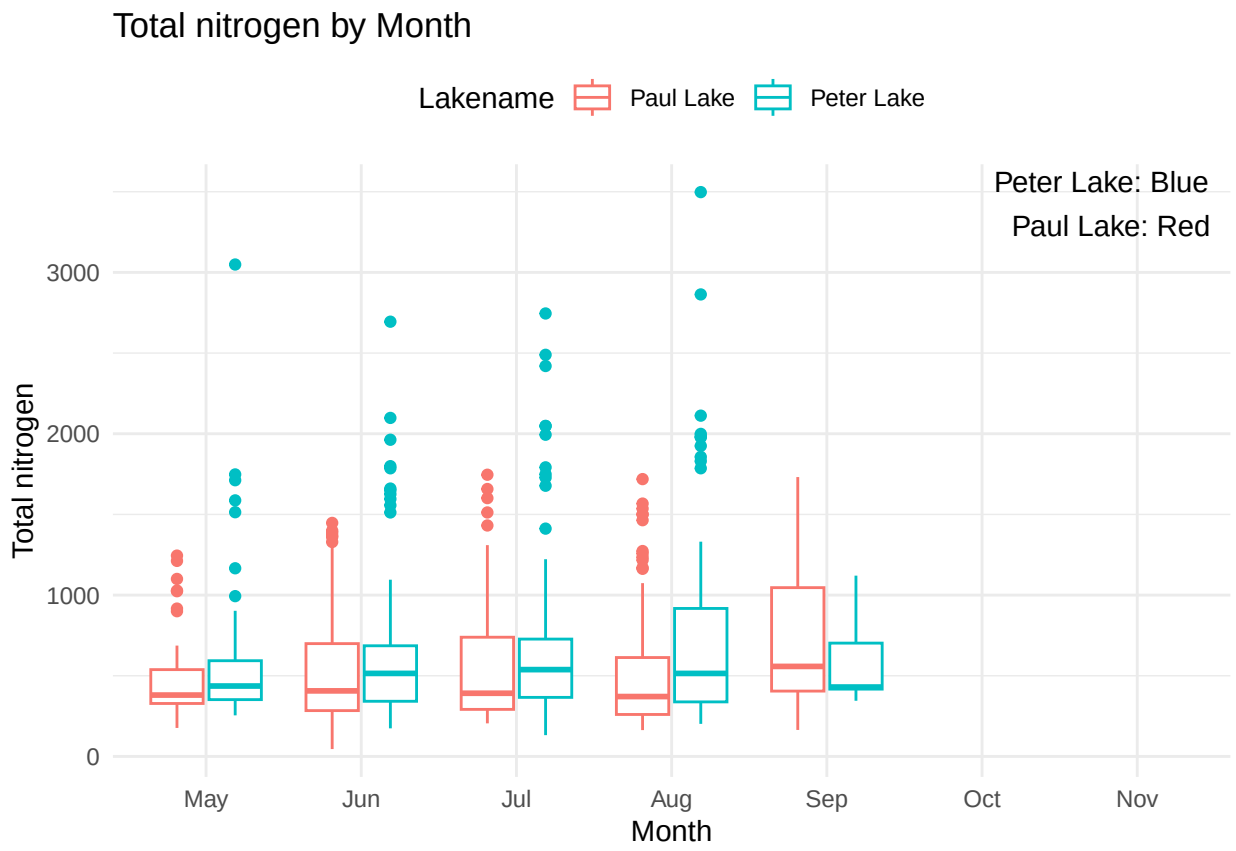

```
#create boxplot for total phosphorus
plot_pho <- ggplot(peter_paul_data_1, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(
    title = "Total phosphorus by Month",
    x = "Month",
    y = "Total phosphorus",
    color = "Lakename"
  ) +
  theme_minimal() +
  theme(legend.position = "none")+  # Hide legend for individual plots
  annotate(
    "text",
    x = Inf, y = Inf,  # Position at the top-right corner
    label = "Peter Lake: Blue\nPaul Lake: Red",  # Text to display
    hjust = 1.1, vjust = 1.1,  # Adjust position
    color = "black", size = 4  # Customize text appearance
  )

# Display the plot
print(plot_pho)
```

```
## Warning: Removed 18579 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
#create boxplot for total nitrogen
plot_ni <- ggplot(peter_paul_data_1, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  labs(
    title = "Total nitrogen by Month",
    x = "Month",
    y = "Total nitrogen",
    color = "Lakename"
  ) +
  theme_minimal() +
  theme(legend.position = "top")+  # Hide legend for individual plots
  annotate(
    "text",
    x = Inf, y = Inf,  # Position at the top-right corner
    label = "Peter Lake: Blue\nPaul Lake: Red",  # Text to display
    hjust = 1.1, vjust = 1.1,  # Adjust position
    color = "black", size = 4  # Customize text appearance
  )

# Display the plot
print(plot_ni)
```

```
## Warning: Removed 18861 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

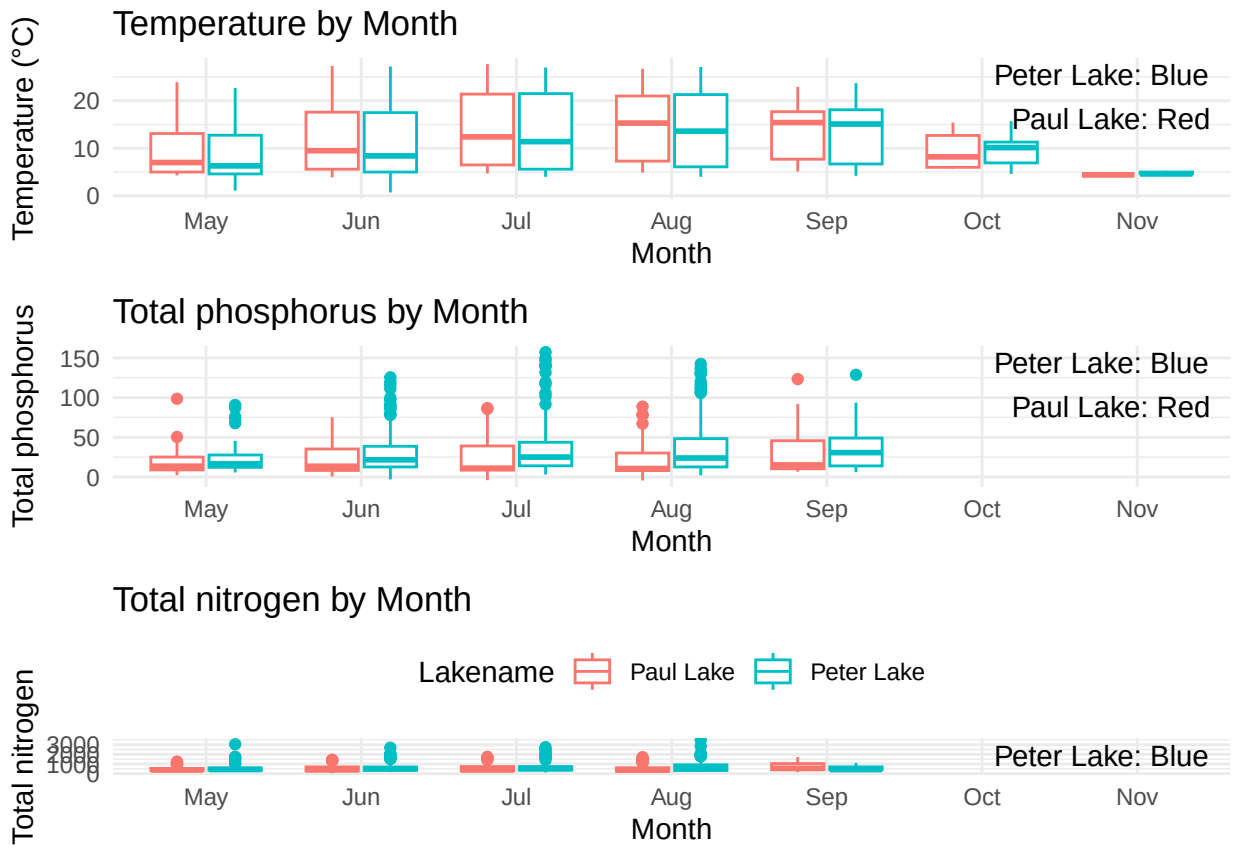



```
#combine the three boxplots
combined_plot <- plot_grid(plot_temp,plot_pho,plot_ni,
                           ncol = 1, align = "v",
                           rel_heights = c(1,1,1))
```

```
## Warning: Removed 18579 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 18861 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
print(combined_plot)
```



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

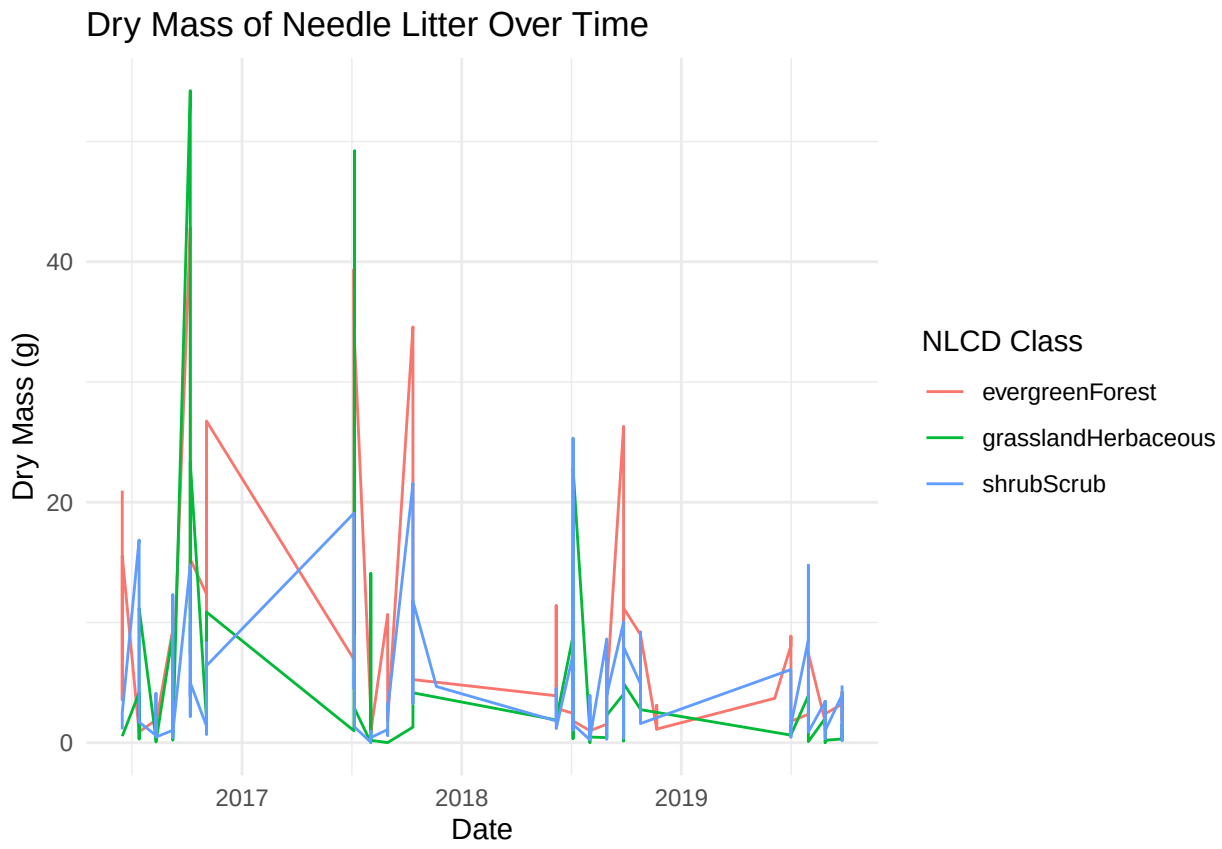
> Answer: We see from the graph that for temperature, Paul lake temperature is higher than Peter lake tempertaure except for October and November. For total phosphorus, Peter lake has higher phosphorus than Paul lake at every month. For nitrogen, Paul lake has lower nitrogen than peter lake excep September.

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)

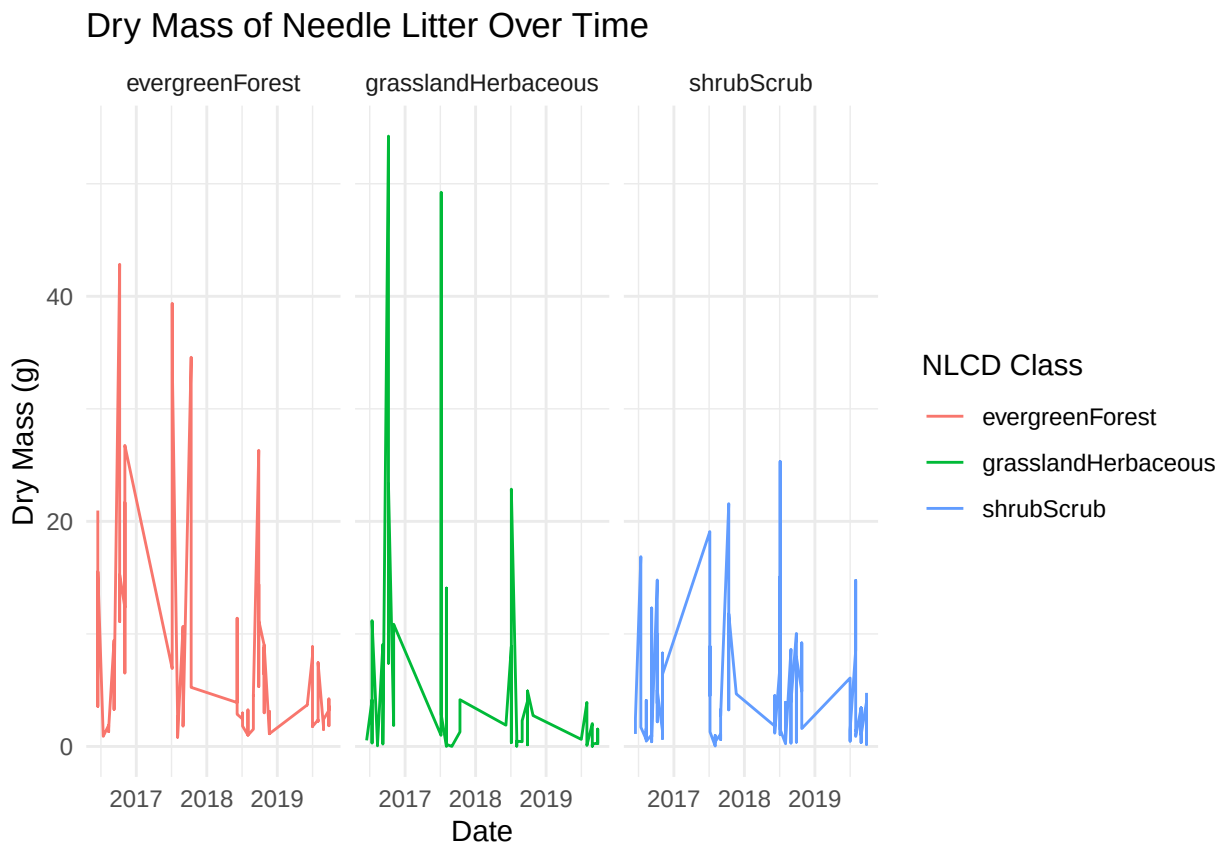
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
#subset the dataset to include only "needless"
needles_data <-
  niwot_litter_data %>%
  filter(functionalGroup=="Needles")
#create the plot
ggplot(needles_data,aes(x=collectDate,y=dryMass,color=nlcdClass))+
  geom_line()+
  labs(
    title = "Dry Mass of Needle Litter Over Time",
    x = "Date",
    y = "Dry Mass (g)",
```

```
    color = "NLCD Class"
  ) +
  theme_minimal()
```




```
#7
ggplot(needles_data,aes(x=collectDate,y=dryMass,color=nlcdClass))+
  geom_line(aes(color=nlcdClass))+
  labs(
    title = "Dry Mass of Needle Litter Over Time",
    x = "Date",
    y = "Dry Mass (g)",
    color = "NLCD Class"
  ) +
  theme_minimal()+
  facet_wrap(~nlcdClass, ncol = 3)  # Separate into three facets
```



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

Answer: If we aim to compare trends across NLCD classes, Plot 6 (with colored lines) is the more effective choice. On the other hand, if our focus is to examine trends within each NLCD class individually, Plot 7 (using facets) is the better option.