Assignment 5: Data Visualization

Yeeun Kim

Spring 2025

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
# Import needed 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
```

```
library(cowplot)
```

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

```
library(ggplot2)
library(dplyr)

# Verify my home directory
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
# Read the files
NTL_litter <- read.csv(
  file=here("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)
NIWOT_litter <- read.csv(
  file=here("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
NTL_litter$sampledate <- as.Date(NTL_litter$sampledate, format="%Y-%m-%d")
NIWOT_litter$collectDate <- as.Date(NIWOT_litter$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
my_theme <- theme(
  plot.background = element_rect(fill="white"),
  plot.title = element_text(size=16, hjust=0.5),
  legend.background = element_rect(fill="white", color="black")
)
```

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.

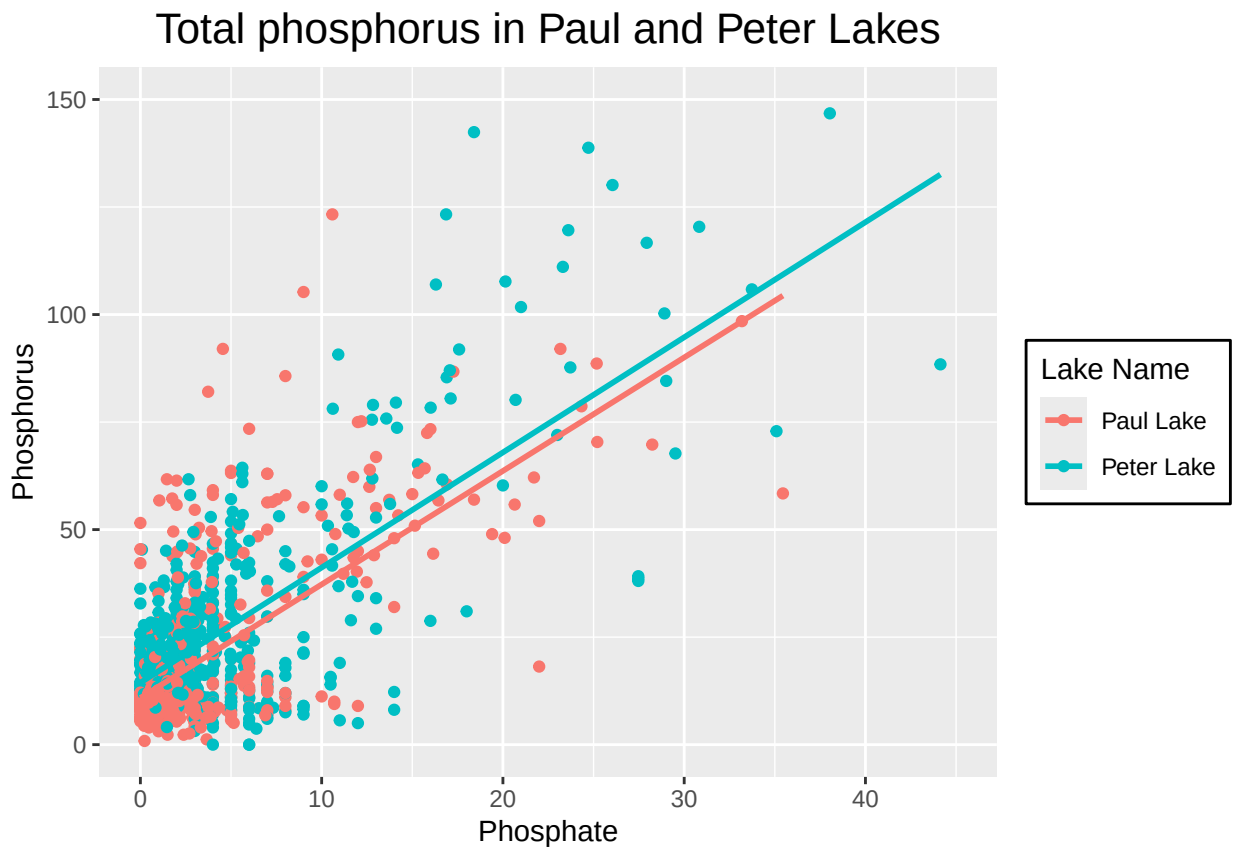
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
NTL_plot <-
  ggplot(NTL_litter, aes(x=po4, y=tp_ug, color=lakename)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  labs(
    title = "Total phosphorus in Paul and Peter Lakes",
    x= "Phosphate",
    y= "Phosphorus",
    color="Lake Name"
  ) +
  xlim(0,45) +
  ylim(0,150) +
  my_theme

print(NTL_plot)
```

```
## `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
# Convert month as a factor
NTL_litter$month <- factor(NTL_litter$month,
       levels=1:12, labels = month.abb)

# Making a boxplot for temperature
temp_plot <-
  ggplot(NTL_litter, aes(x=factor(month), y=temperature_C, fill=lakename)) +
  scale_x_discrete(drop=F) +
  geom_boxplot() +
  labs(y="Temperature", x=NULL) +
  my_theme +
  theme(
    legend.position = "none",
    axis.title.x = element_blank()
  )
```

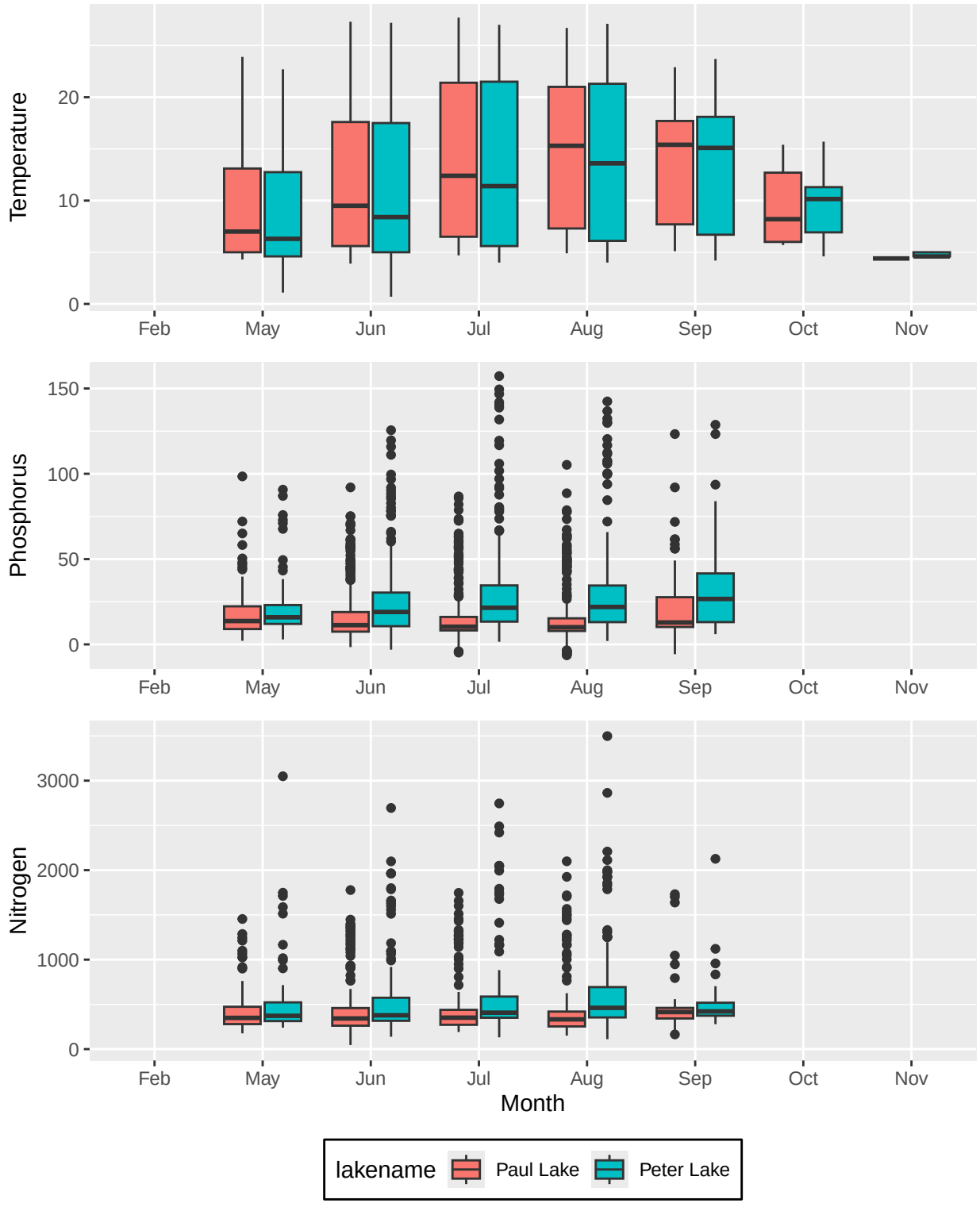
```r
# Making a boxplot for tp
TP_plot <-
  ggplot(NTL_litter, aes(x=factor(month), y=tp_ug, fill=lakename)) +
  scale_x_discrete(drop=F) +
  geom_boxplot() +
  labs(y="Phosphorus", x=NULL) +
  my_theme +
  theme(
    legend.position = "none",
    axis.title.x = element_blank()
  )

# Making a boxplot for TN
TN_plot <-
  ggplot(NTL_litter, aes(x=factor(month), y=tn_ug, fill=lakename)) +
  scale_x_discrete(drop=F) +
  geom_boxplot() +
  labs(y="Nitrogen", x="Month") +
  my_theme +
  theme(
    legend.position = "bottom",
  )

# Integrating three boxplots
combined_plot <- plot_grid(
  temp_plot, TP_plot, TN_plot,
  align ="v",
  ncol=1,
  rel_heights = c(1,1,1.4)
)
```

```
## 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()‘).
```

```r
print(combined_plot)
```



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

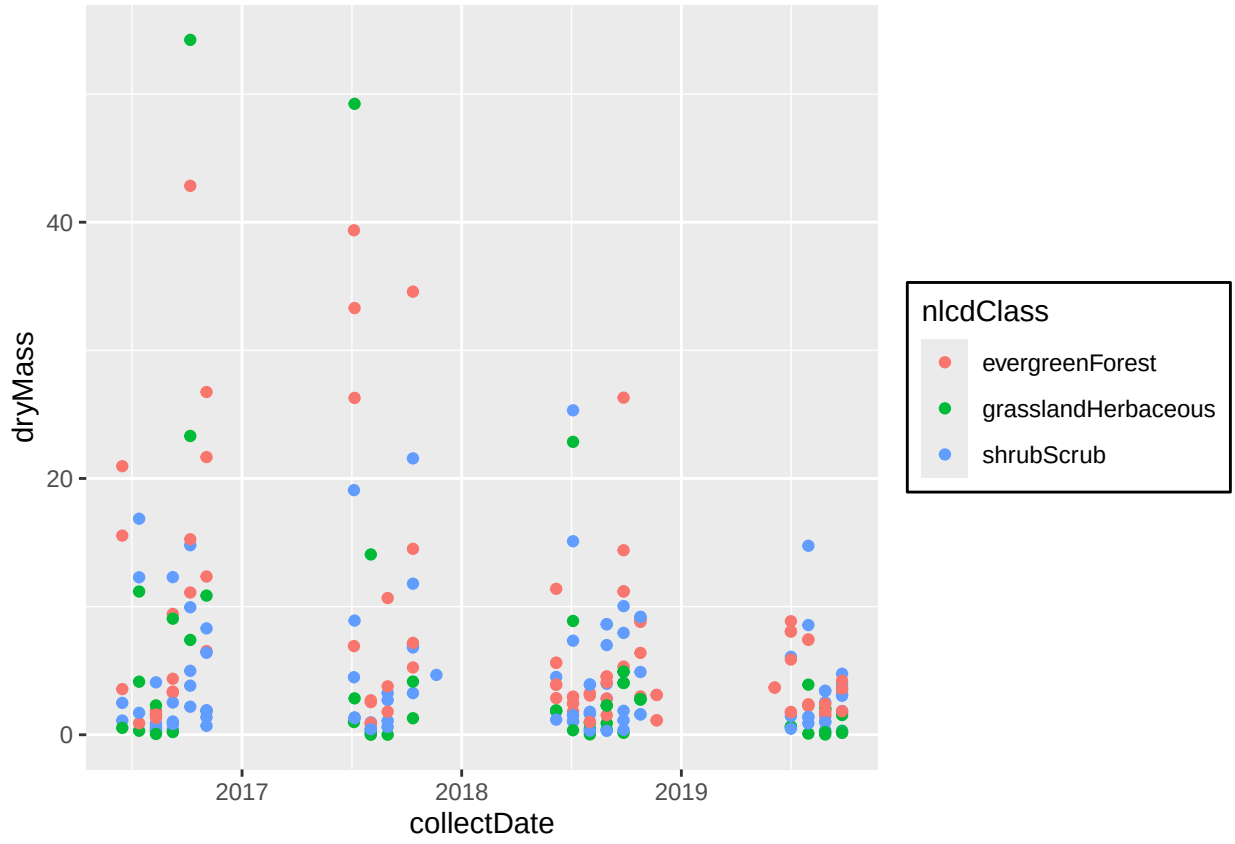
Answer: Temperature fluctuates seasonally, increasing in summer and decreasing in winter. In Paul Lake, phosphorus levels appear to follow an inverse trend to temperature, decreasing as temperature rises. In contrast, phosphorus in Peter Lake shows a steady increase over time, sug-

gesting it is less influenced by temperature. For nitrogen, Paul Lake exhibits minimal variation, while Peter Lake shows a pattern that appears to be influenced by temperature changes.

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)

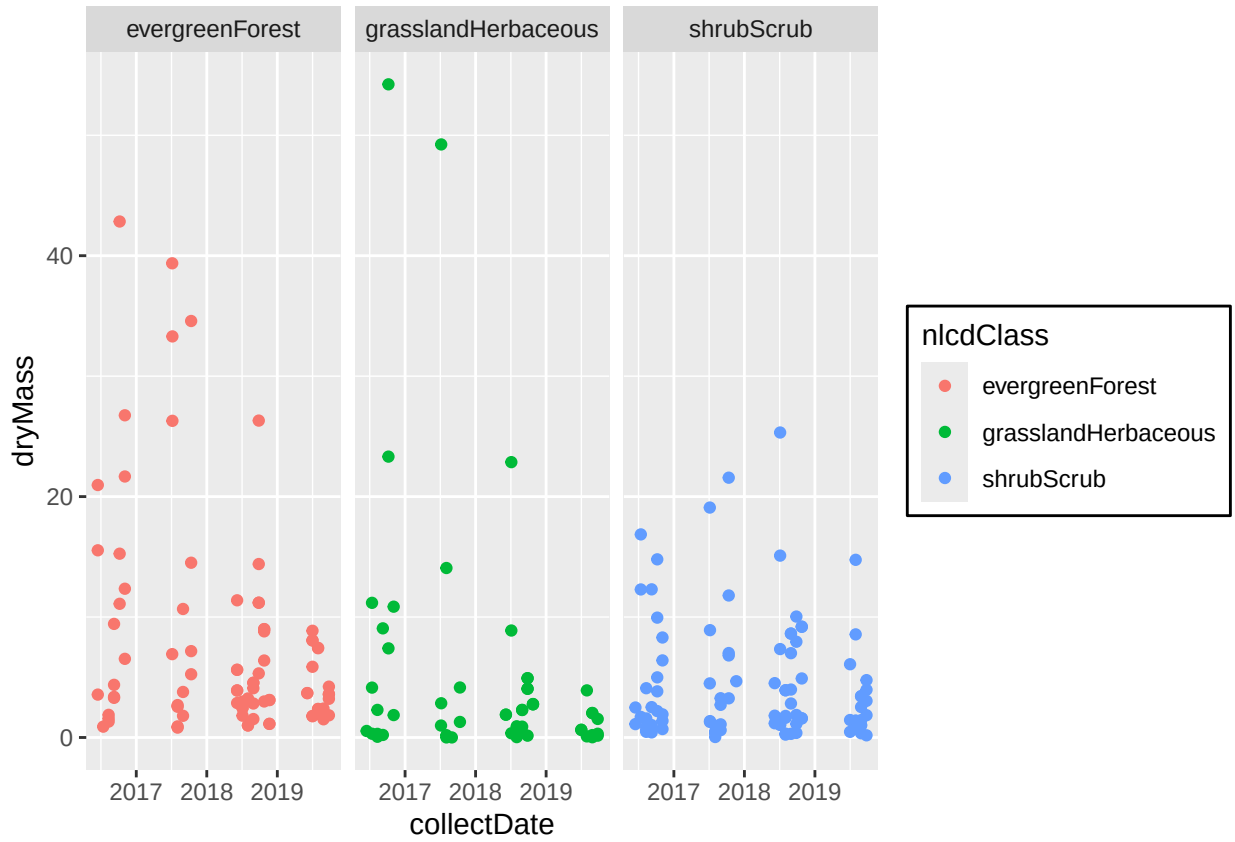
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
# Define a dataframe for a subset of the litter dataset
subset_needles <- subset(NIWOT_litter, functionalGroup =="Needles")

# Plot the data
needles_plot <-
  ggplot(subset_needles,aes(x=collectDate, y=dryMass, color=nlcdClass))+
  geom_point()+
  my_theme

print(needles_plot)
```




```
#7
# Plot the data
needles_seperate_plot <-
```

```
  ggplot(subset_needles, aes(x=collectDate, y=dryMass, color=nlcdClass))+
  geom_point()+
  facet_wrap(~nlcdClass, ncol = 3)+
  my_theme

print(needles_seperate_plot)
```



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

Answer: It depends on the purpose of data analysis. The plot from 6 is better when compaing trends across classes during the same time periods. The plot from 7 is better when examining individual nlcd class trends and its pattern.