Assignment 5: Data Visualization

Queenie Wei

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
#reading in all the necessary packages, making sure the wd is right, and reading in the csvs
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.3.1
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.1
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.2     v readr   2.1.4
## v forcats 1.0.0     v stringr 1.5.0
## v ggplot2 3.4.2     v tibble  3.2.1
## v purrr   1.0.1     v tidyr   1.3.0
```

```
## -- 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(here)
```

```
## Warning: package 'here' was built under R version 4.3.1
```

```
## here() starts at C:/Users/ziyaw/Downloads/EDE_Fall2023
```

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.3.1
```

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

```
getwd()
```

```
## [1] "C:/Users/ziyaw/Downloads/EDE_Fall2023"
```

```
here()
```

```
## [1] "C:/Users/ziyaw/Downloads/EDE_Fall2023"
```

```
LTER_PerterPaul <- read.csv(here("Data/Processed_KEY/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_Pe
                      stringsAsFactors = TRUE)
Litter_trap <- read.csv(here("Data/Processed_KEY/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv
                      stringsAsFactors = TRUE)

#2
class(LTER_PerterPaul$sampledate)
```

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

```r
class(Litter_trap$collectDate)
```

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

```r
#setting the dates to the date format
LTER_PerterPaul$sampledate <- as.Date(LTER_PerterPaul$sampledate,
                                      format = "%Y-%y-%d")
Litter_trap$collectDate <- as.Date(Litter_trap$collectDate,
                                   format = "%Y-%y-%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

```r
#3
fudge <- theme_bw(base_size = 15) +
  theme(axis.text = element_text(color = "pink3"),
        plot.background = element_rect(fill = "lightgray"),
        axis.text.y = element_text(color = "blue4"),
        axis.text.x = element_text(color = "blue4"),
        axis.title = element_text(color = "darkred"),
        legend.position = "bottom")
```

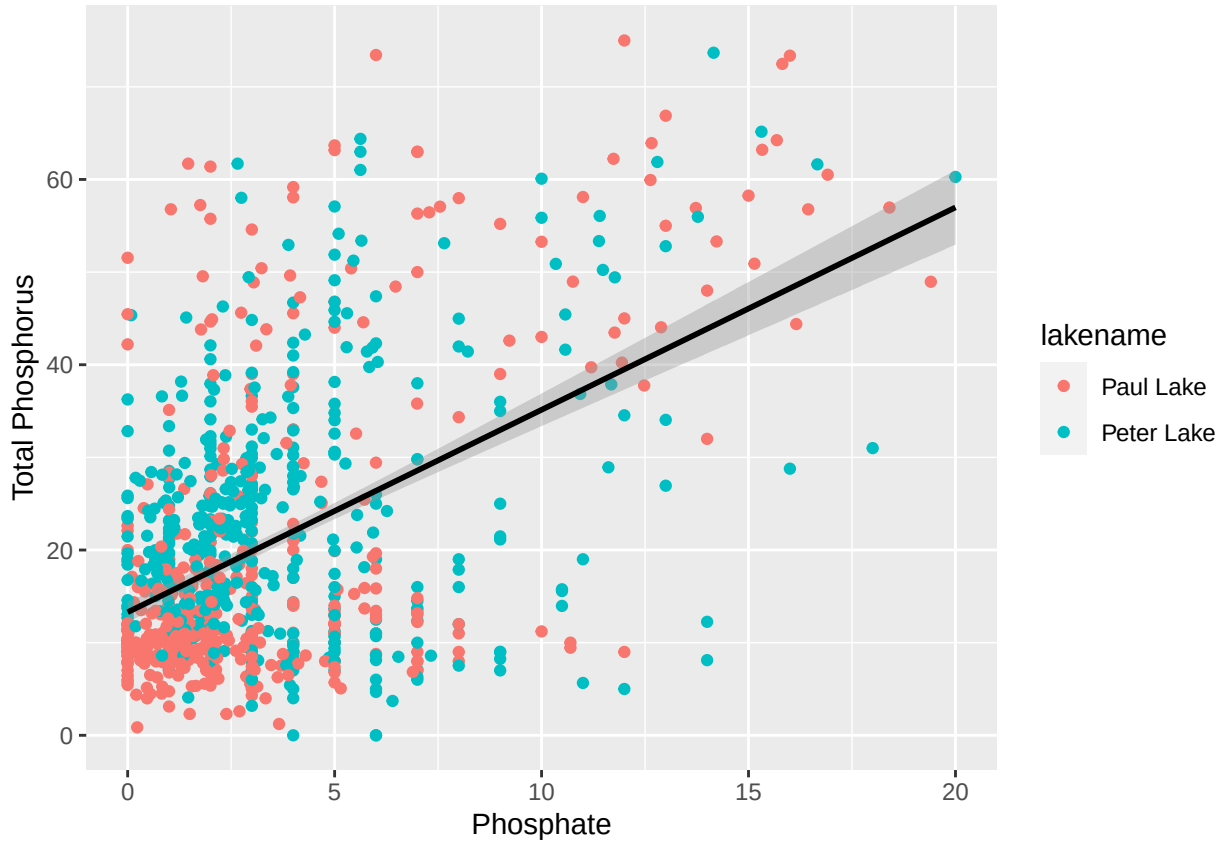
Create graphs

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()`).

```r
#4

NTL_LTER_Adjusted <-
  ggplot(LTER_PerterPaul, aes(x = po4, y = tp_ug)) +
  geom_point(aes(color = lakename)) +
  geom_smooth(method = lm, color = "black") +
  ylab("Total Phosphorus") +
  xlab("Phosphate") +
  xlim(0, 20) +
  ylim(0, 75)
print(NTL_LTER_Adjusted)
```

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

## Warning: Removed 22002 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 22002 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.

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

library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.3.1

##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:cowplot':
##
##     get_legend
```

```
library(gridGraphics)
```

```
## Warning: package 'gridGraphics' was built under R version 4.3.1
```

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

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.3.1
```

```
##
## Attaching package: 'gridExtra'
```

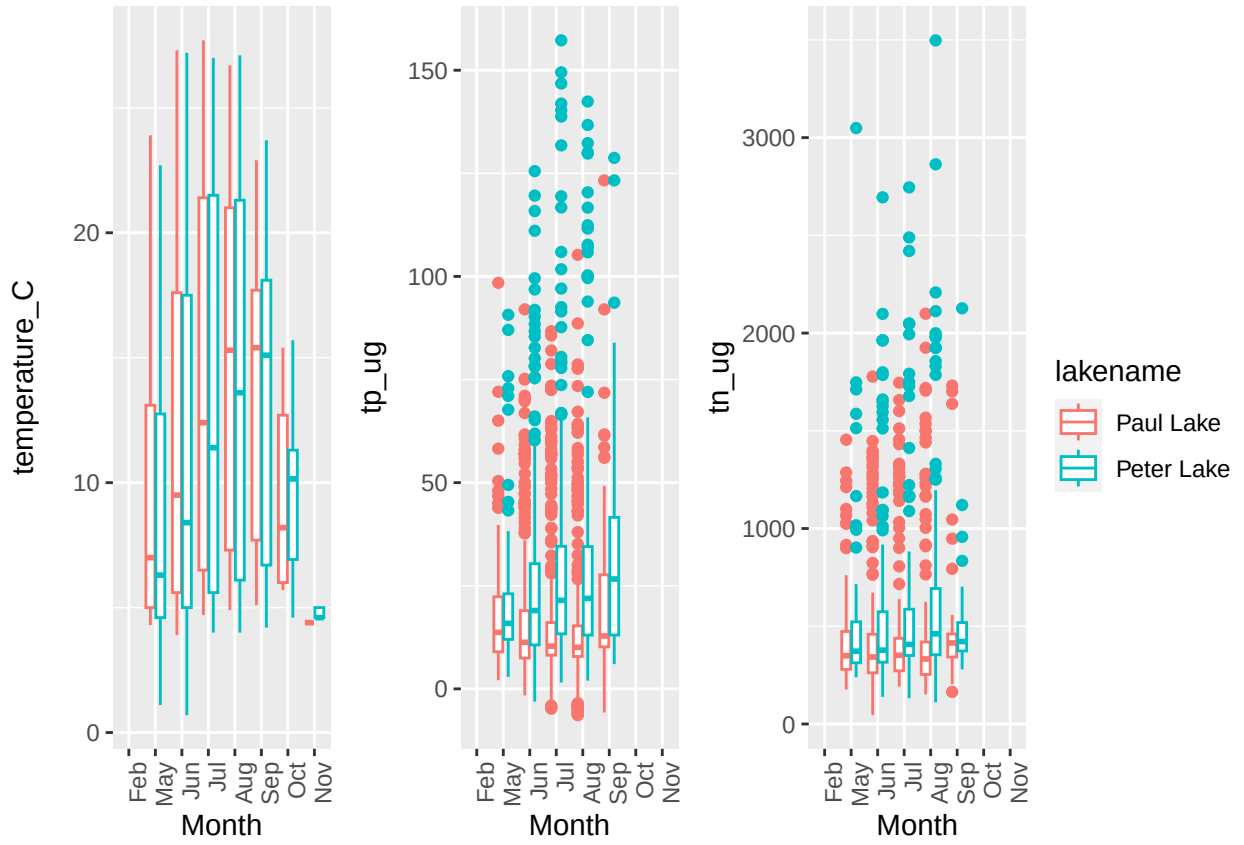
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# month.abb[mth]
# temp <-
#   ggplot(LTER_PerterPaul, aes(x = month.abb[mth], y = temperature_C)) +
#   geom_boxplot(aes(color = lakename)) +
#   theme(legend.position = "none")
# print(temp)
#
# TP <-
#   ggplot(LTER_PerterPaul, aes(x = month.abb[mth], y = tp_ug)) +
#   geom_boxplot(aes(color = lakename)) +
#   theme(legend.position = "none")
#
# print(TP)
#
# TN <-
#   ggplot(LTER_PerterPaul, aes(x = month.abb[mth], y = tn_ug)) +
#   geom_boxplot(aes(color = lakename))
#
# plot_grid(temp,TP, TN, nrow = 1, align = 'h')
##original code


mth <- LTER_PerterPaul$month
##
month_order <- rev(month.abb)

# Create the temp plot
temp <- ggplot(LTER_PerterPaul, aes(x = factor(month.abb[mth],
                                               levels = rev(month_order)),
                                    y = temperature_C)) +
    xlab("Month") +
  geom_boxplot(aes(color = lakename)) +
 guides(color = "none")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```r
# Create the TP plot
TP <- ggplot(LTER_PerterPaul, aes(x = factor(month.abb[mth],
                                             levels = rev(month_order)),
                                  y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  xlab("Month") +
 guides(color = "none")+
theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Create the TN plot
TN <- ggplot(LTER_PerterPaul, aes(x = factor(month.abb[mth],
                                             levels = rev(month_order)),
                                  y = tn_ug)) +
    xlab("Month") +
  geom_boxplot(aes(color = lakename)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

final_plot <- ggpubr::ggarrange(temp, TP, TN,
                  common.legend = TRUE,
                  legend = "right",
                  align = "hv",
                  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()`).

## 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()`).
```

```r
# Display the final_plot_with_legend
final_plot
```




```
#######
####works cited: ChatGPT
###promt: I fed it my original code and used the prompt "arrange the x axis so
#that they are in the order of months".
### the code it gave me got rid of all of the legends, so I had to
#manually delete one of the "theme(legend.position = "none")"
###the original answer Chatgpt gave me also had the months in a reverse order,
#so I had to tell it to reverse the months
```

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

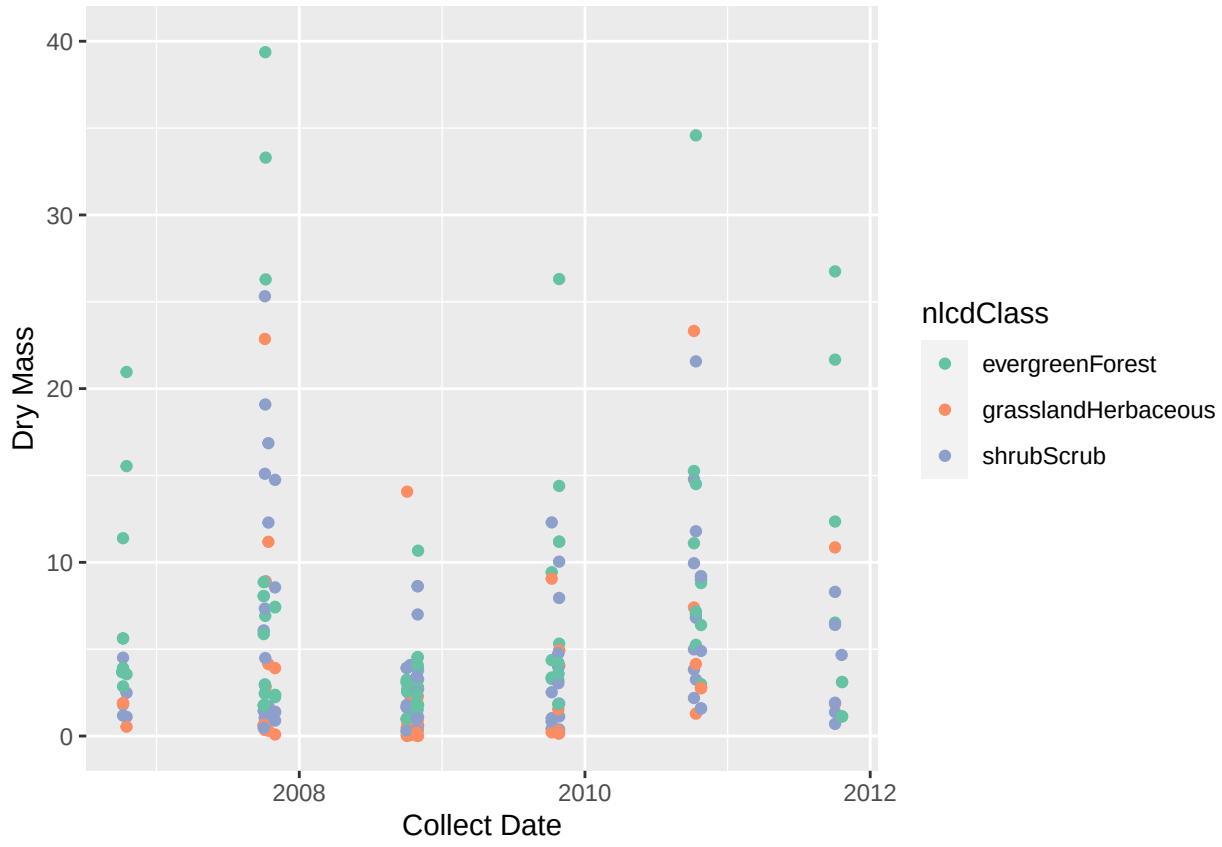
> Answer: Temperature tends to peak around July and August (the summer), and gets pretty low in the winter. Phosphorous demonstrates different trends in the two lakes. in Paul Lake, there seems to be a decline from May to August, following by an increase in September. For Peter Lake, it tends to increase from May to September. Phosphate: there isnt a general trend for Paul lake, except for an increase in September. For Peter Lake, there is a positive trend from May to August, fowlling by a decline in 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

####creates a subset of the litter dataset with only the needles functional group. This plots the dry m

library(ggplot2)

# Create the boxplot
library(ggplot2)

# Create the boxplot
boxplot_plot <- ggplot(subset(Litter_trap, functionalGroup == "Needles"),
                       aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  xlab("Collect Date") +
  ylab("Dry Mass") +
  ylim(0, 40) +
  scale_color_brewer(palette = "Set2")

boxplot_plot
```

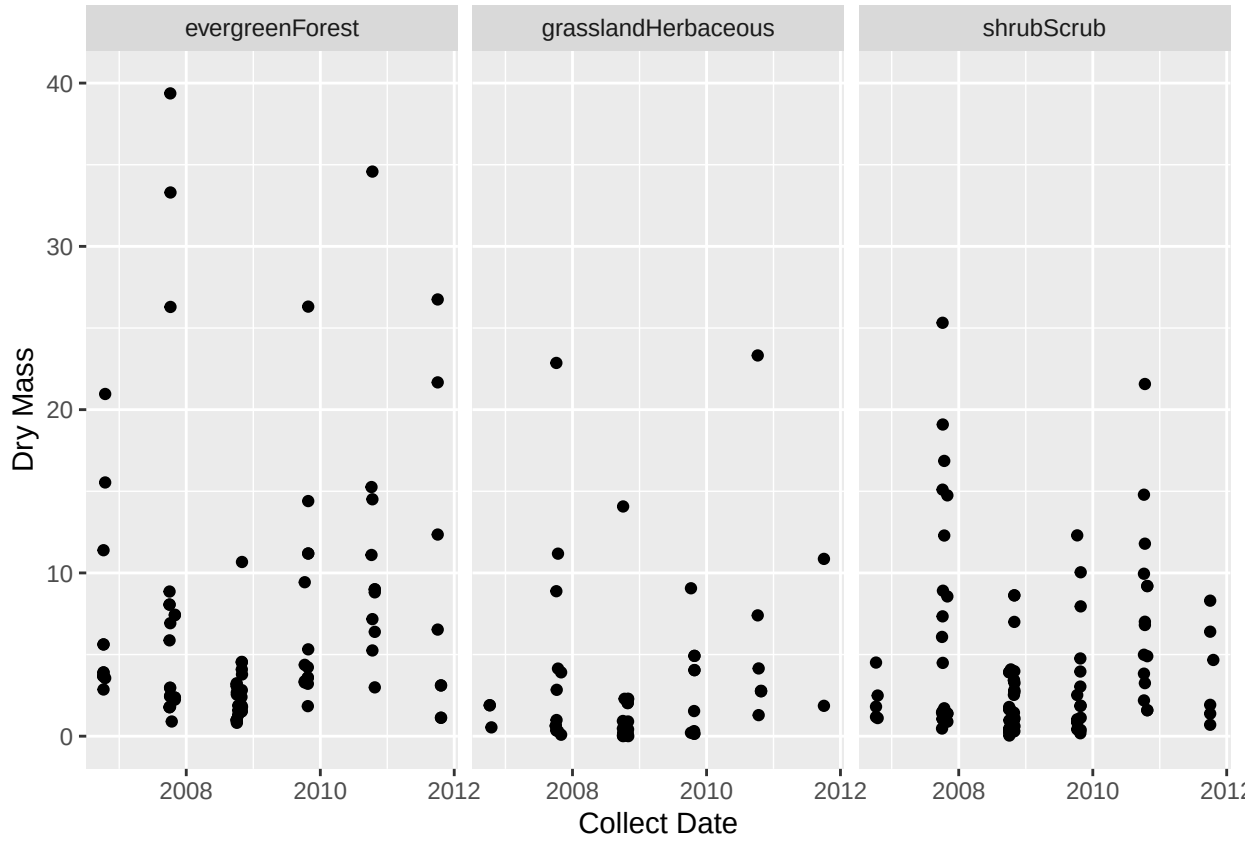
```
## Warning: Removed 3 rows containing missing values (‘geom_point()‘).
```




```
#7
boxplot_plot2 <- ggplot(subset(Litter_trap, functionalGroup == "Needles"),
                       aes(x = collectDate, y = dryMass)) +
```

```
  geom_point() +
  xlab("Collect Date") +
  ylab("Dry Mass") +
  ylim(0, 40) +
  facet_wrap(vars(nlcdClass), nrow = 1)

boxplot_plot2
```

```
## Warning: Removed 3 rows containing missing values (`geom_point()`).
```




```
######additional fun stuff im trying out

boxplot_plot3 <- ggplot(Litter_trap,
                        aes(x = factor(year(collectDate)), y = dryMass,
                            color = nlcdClass)) +
  geom_boxplot(outlier.shape = NA) +  # Remove outlier points from the boxplots
  geom_jitter(position = position_jitter(width = 0.2), alpha = 0.2, size = 1) +  # Add individual obser
  xlab("Collect Year") +
  ylab("Dry Mass") +
  ylim(0, 8.5) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_wrap(~nlcdClass, nrow = 1)

# Print the plot
print(boxplot_plot3)
```

```
## Warning: Removed 64 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 319 rows containing missing values (`geom_point()`).
```


```
#####Works cited:chatgpt
#### I fed it my code from #7 and used the prompt "instead of this, create boxplots for each nlcdclass
#### this graph could also use some colors :) and then I added color = nlcdClass to make it prettier
```

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

> Answer: I believe plot 7 is more effective at communicating how the distribution of dry mass differs among the three facetwrap groups since the different points are seperated by groups. However, I would appreciate it if the different groups have different colors, and having a boxplot would also be nice too. (like the one I did in the end)