Assignment 5: Data Visualization

Laura Martinez

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

The completed exercise is due on Friday, Oct 14th @ 5:00pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse, lubridate, & 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 check working directory, load packages, and load data
setwd("~/Documents/EDA-Fall2022")
getwd()
```

```
## [1] "/Users/laura/Documents/EDA-Fall2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
```

```
##     date, intersect, setdiff, union
```

```
# install.packages('cowplot')
library(cowplot)
```

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

```
LTER_Nutrients <-  LTER_Nutrients <-
read.csv("~/Documents/EDA-Fall2022/Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.cs
    stringsAsFactors = TRUE)

NIWO_MassTrap <-  NIWO_MassTrap <-
read.csv("~/Documents/EDA-Fall2022/Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
    stringsAsFactors = TRUE)

# 2
LTER_Nutrients$sampledate <- as.Date(LTER_Nutrients$sampledate,
    format = "%Y-%m-%d")
NIWO_MassTrap$collectDate <- as.Date(NIWO_MassTrap$collectDate,
    format = "%Y-%m-%d")
```

Define your theme

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

```
# 3 Build & set theme

Laurastheme <- theme_classic(base_size = 12) + theme(axis.text = element_text(color = "plum4"),
    legend.position = "bottom")
theme_set(Laurastheme)
```

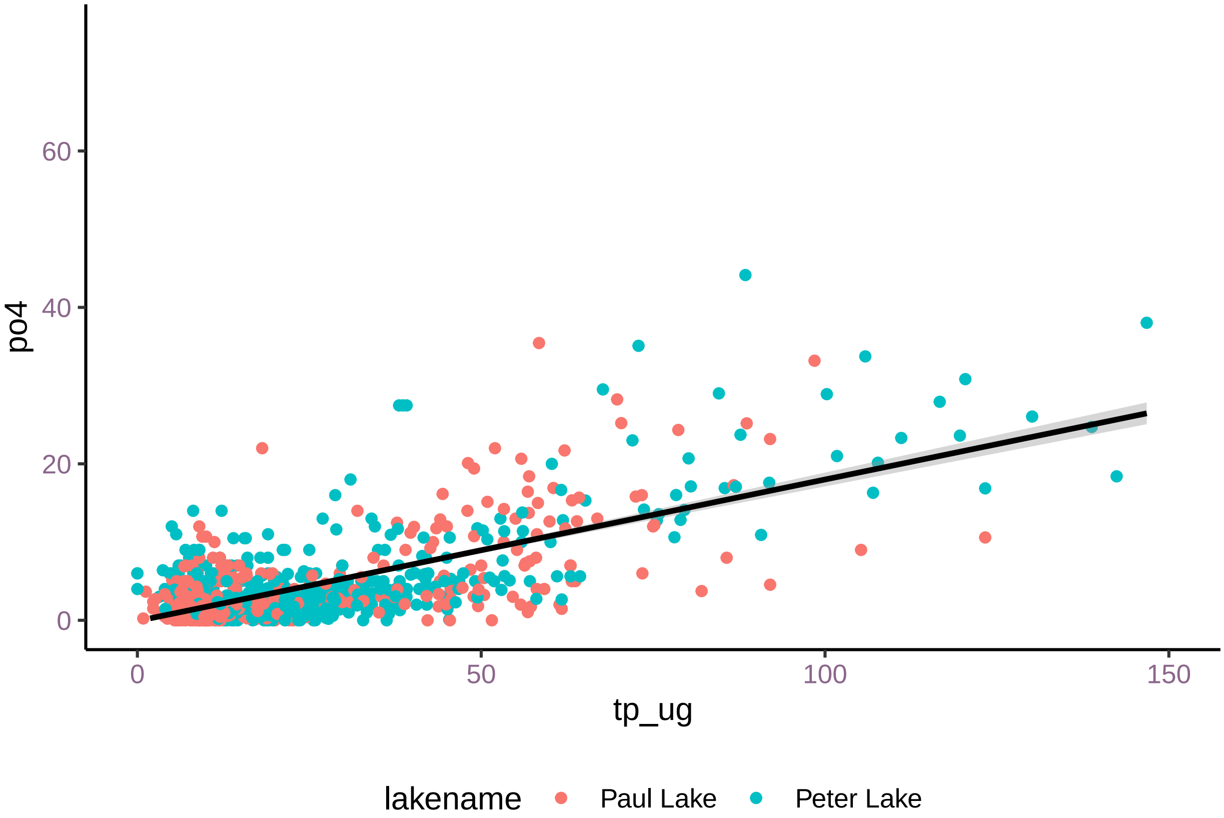
Create graphs

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
Pbypo4 <- ggplot(LTER_Nutrients, aes(x = tp_ug, y = po4)) + geom_point(aes(color = lakename)) +
    geom_smooth(method = lm, color = "black") + xlim(0, 150) +
    ylim(0, 75)
print(Pbypo4)
```

```
## `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).

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



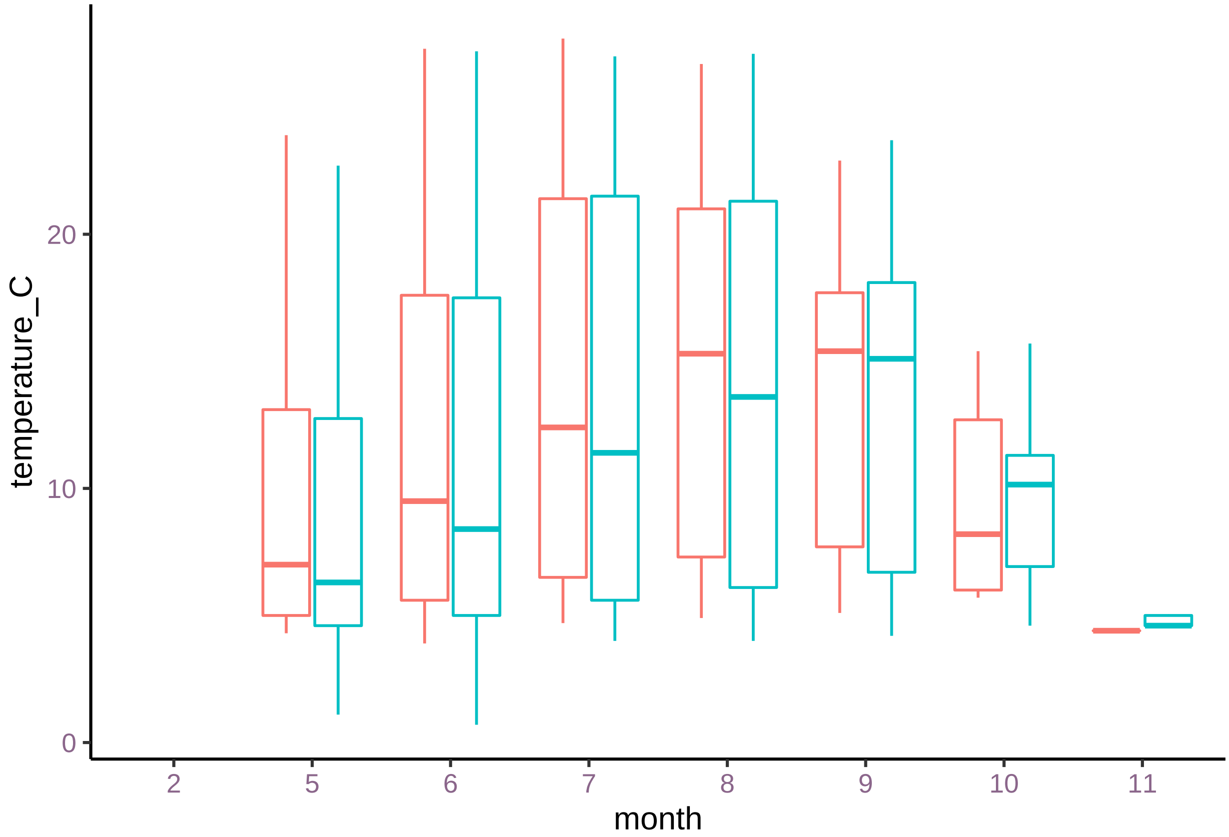
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

(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: R has a build in variable called `month.abb` that returns a list of months; see https://r-lang.com/month-abb-in-r-with-example

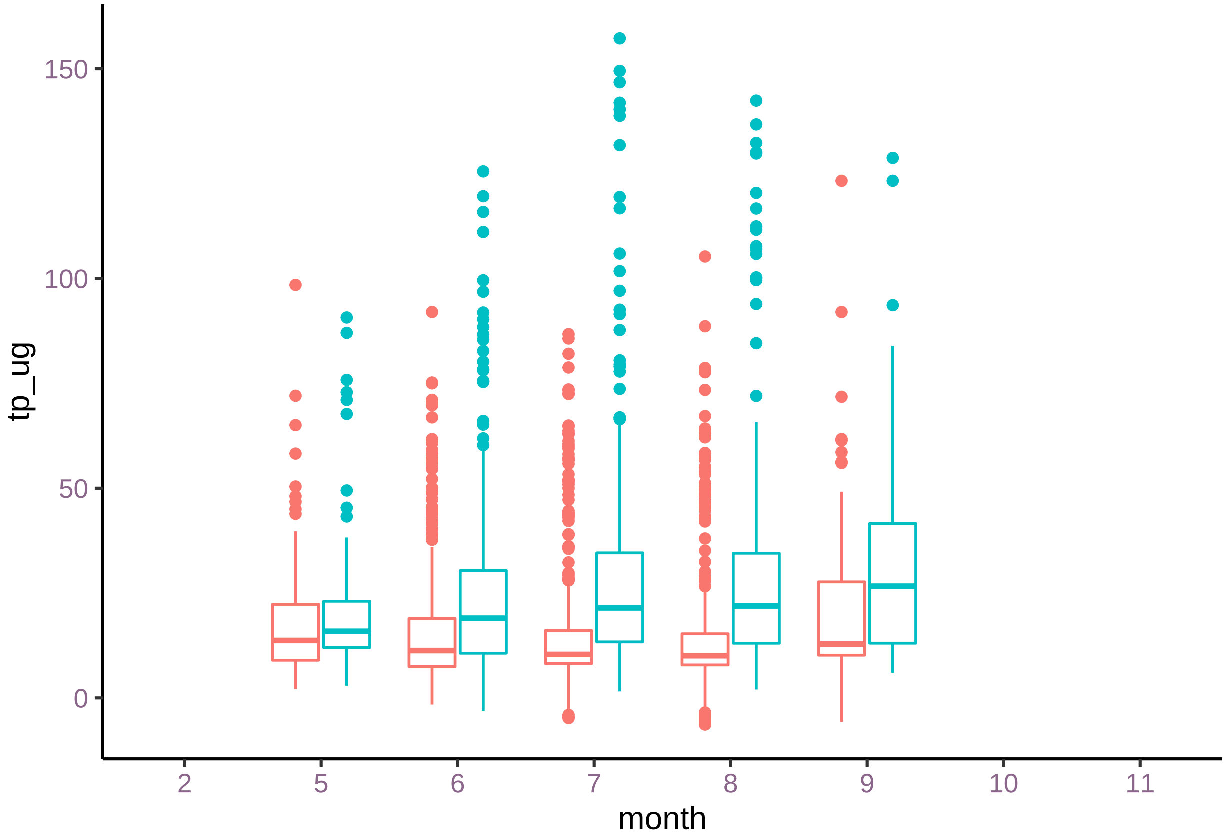
```
#Recast month as factors
LTER_Nutrients$month <- as.factor(LTER_Nutrients$month)

#5
Temp.boxplot <-
  ggplot(LTER_Nutrients, aes(x=month, y=temperature_C)) +
  geom_boxplot(aes(color=lakename)) +
  theme(legend.position="none")
print(Temp.boxplot)
```

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



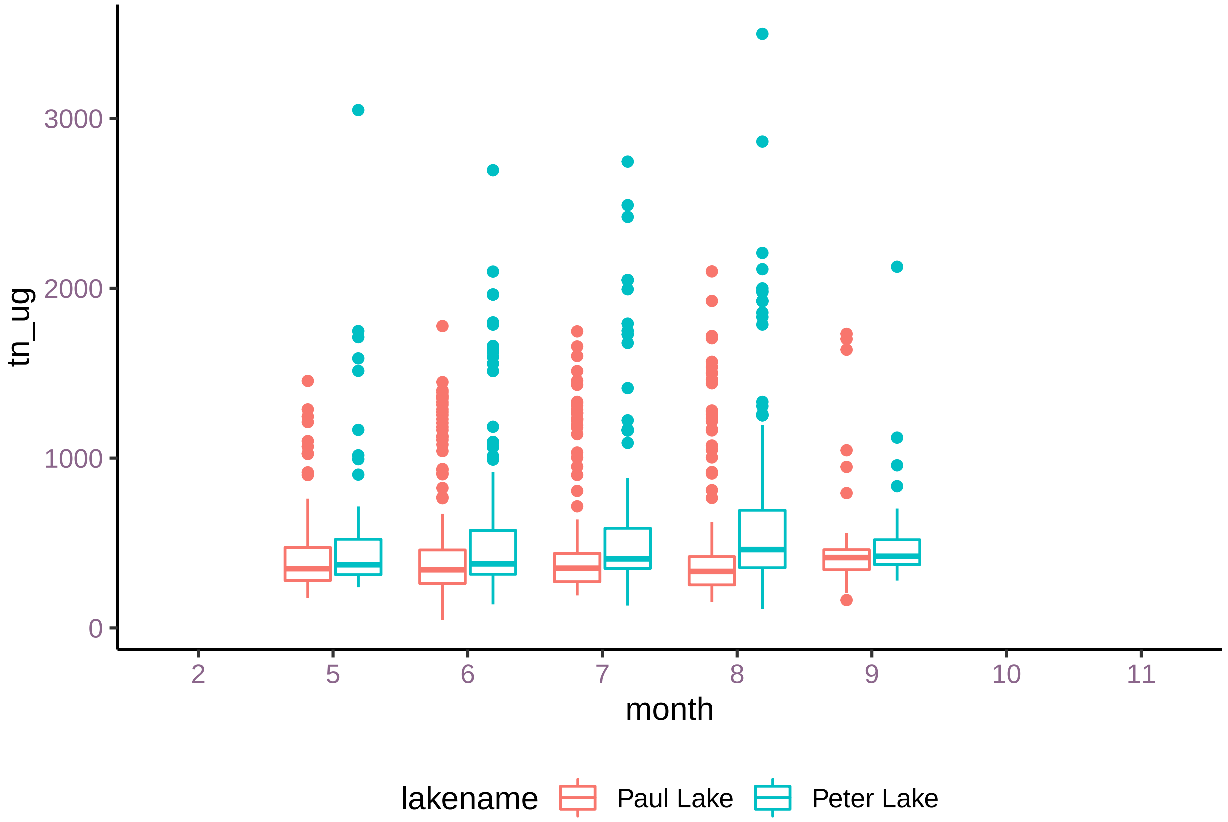
```
TP.boxplot <-
  ggplot(LTER_Nutrients, aes(x=month, y=tp_ug)) +
  geom_boxplot(aes(color=lakename)) +
  theme(legend.position="none")
print(TP.boxplot)
```

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



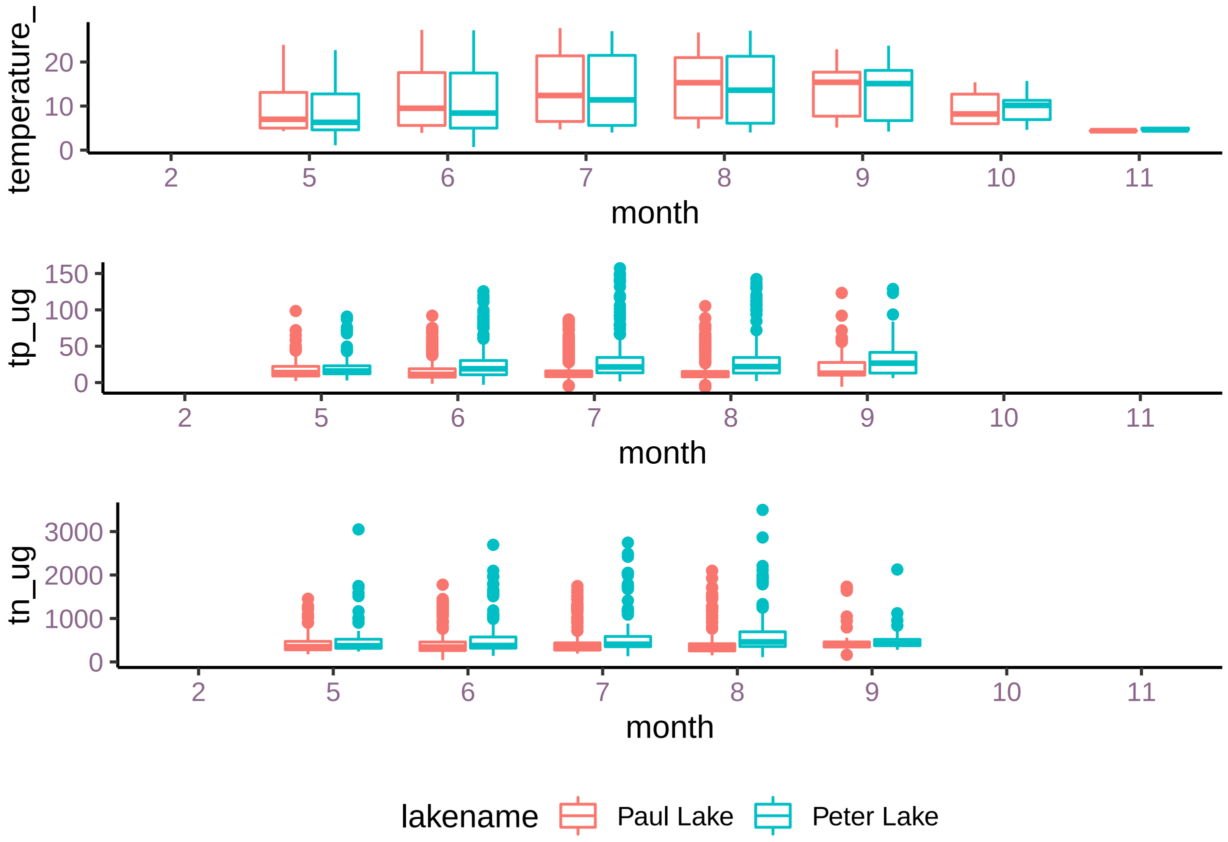
```
TN.boxplot <-
  ggplot(LTER_Nutrients, aes(x=month, y=tn_ug)) +
  geom_boxplot(aes(color=lakename))
print(TN.boxplot)
```

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



```
plot_grid(Temp.boxplot, TP.boxplot, TN.boxplot, ncol = 1, align = 'h',

rel_heights = c(2.5, 2.5, 4))
```

```
## 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: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

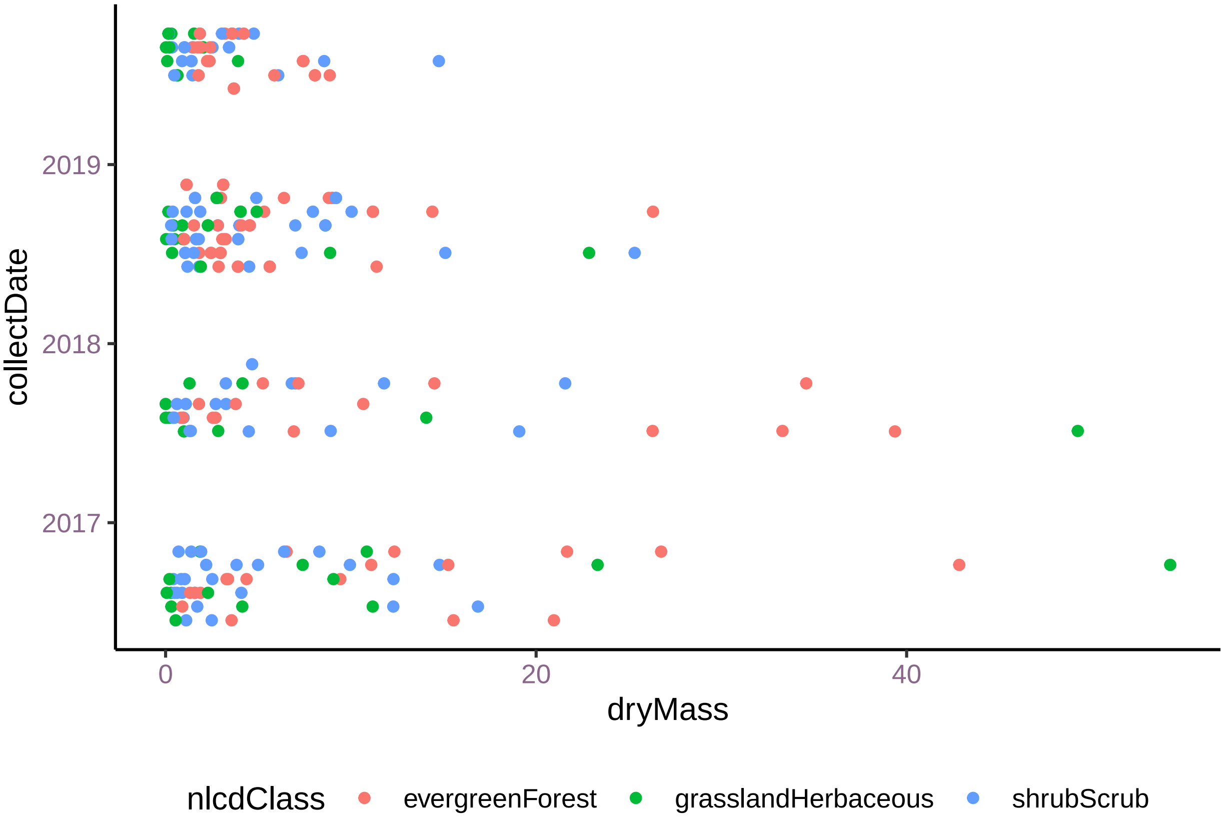


Question: What do you observe about the variables of interest over seasons and between lakes?

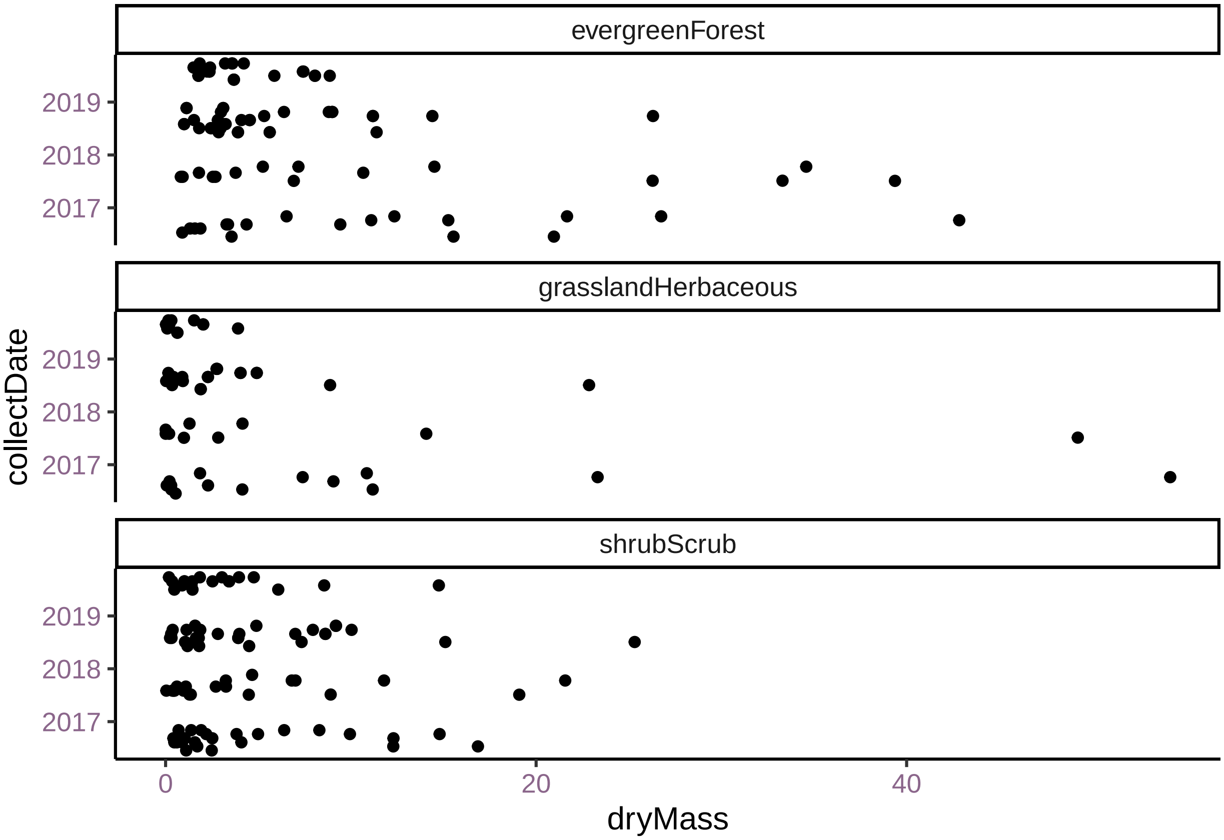
> Answer: Both lakes follow a normal distribution for mean temperature, whereas mean tp_ug shows a more consistent trend across the months and seasons. Similarly, mean tn_ug follows a consistent pattern across months and seasons. One difference is that temperature does not show a lot of outlier data, where tp_ug and tn_ug have a lot of data skewing the IQR into a higher range. While the means are similar for Peter and Paul Lake, the tp_ug and tn_ug plots show that Paul Lake tends to have lower mean values across the year.

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
subsetNIWOT <- filter(NIWO_MassTrap, functionalGroup %in% c("Needles"))

Needles1 <- ggplot(subsetNIWOT, aes(x = dryMass, y = collectDate)) +
    geom_point(aes(color = nlcdClass))
print(Needles1)
```



```
# 7
Needles2 <- ggplot(subsetNIWOT, aes(x = dryMass, y = collectDate)) +
    geom_point() + facet_wrap(vars(nlcdClass), nrow = 3)
print(Needles2)
```



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

Answer: The first graph is better at showing the overall trends in dry mass from year to year as an aggregate of the different NLCD classes. The second graph, is showing the data more teased apart.For visualization purposes, it is easier to see the dry mass trends from year to year for each seperate NLCD class. I think one way to make this easier to understand for audiences, would be to use the second graph and also color each NCLD class to make it visually easier to differentiate between evergreen, grassland, and shrub data.