Assignment 5: Data Visualization

Jacob Freedman

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_PeterPau`
version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed`
version).

2. Make sure R is reading dates as date format; if not change the format to date.

```
# 1
setwd("~/R/EDA-Fall2022")
getwd()
```

```
## [1] "/home/guest/R/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   1.2.0      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
```

```
library(cowplot)
```

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

```
PeterPaul_Processed <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed
    stringsAsFactors = TRUE)
NIWO_Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
    stringsAsFactors = TRUE)

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

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

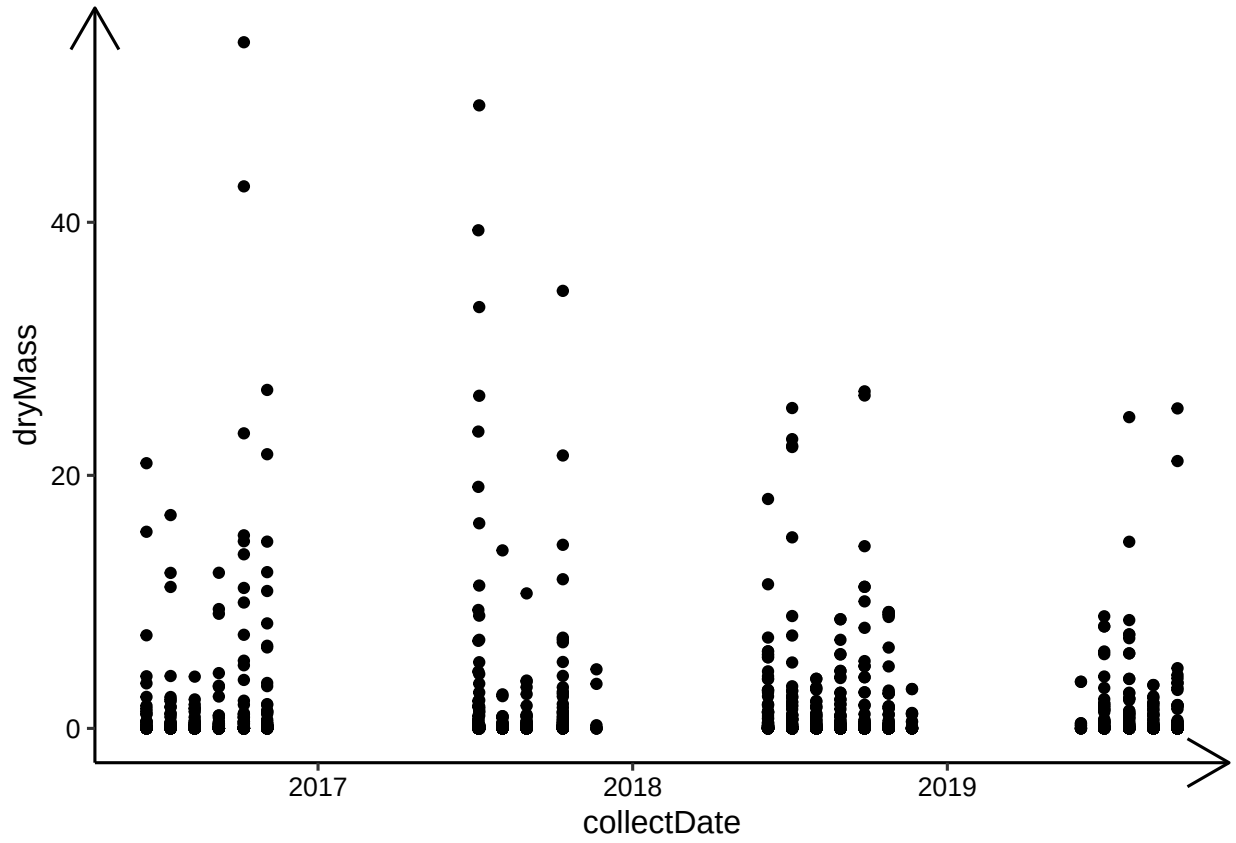
```
class(NIWO_Litter$collectDate)
```

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

Define your theme

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

```
# 3
theme1 <- theme_classic(base_size = 12) + theme(axis.text = element_text(color = "black"),
    legend.position = "left", axis.line = element_line(arrow = arrow()))
# Testing the appearance of my theme
ggplot(NIWO_Litter) + geom_point(aes(x = collectDate, y = dryMass)) + theme1
```




```
theme_set(theme1)
```

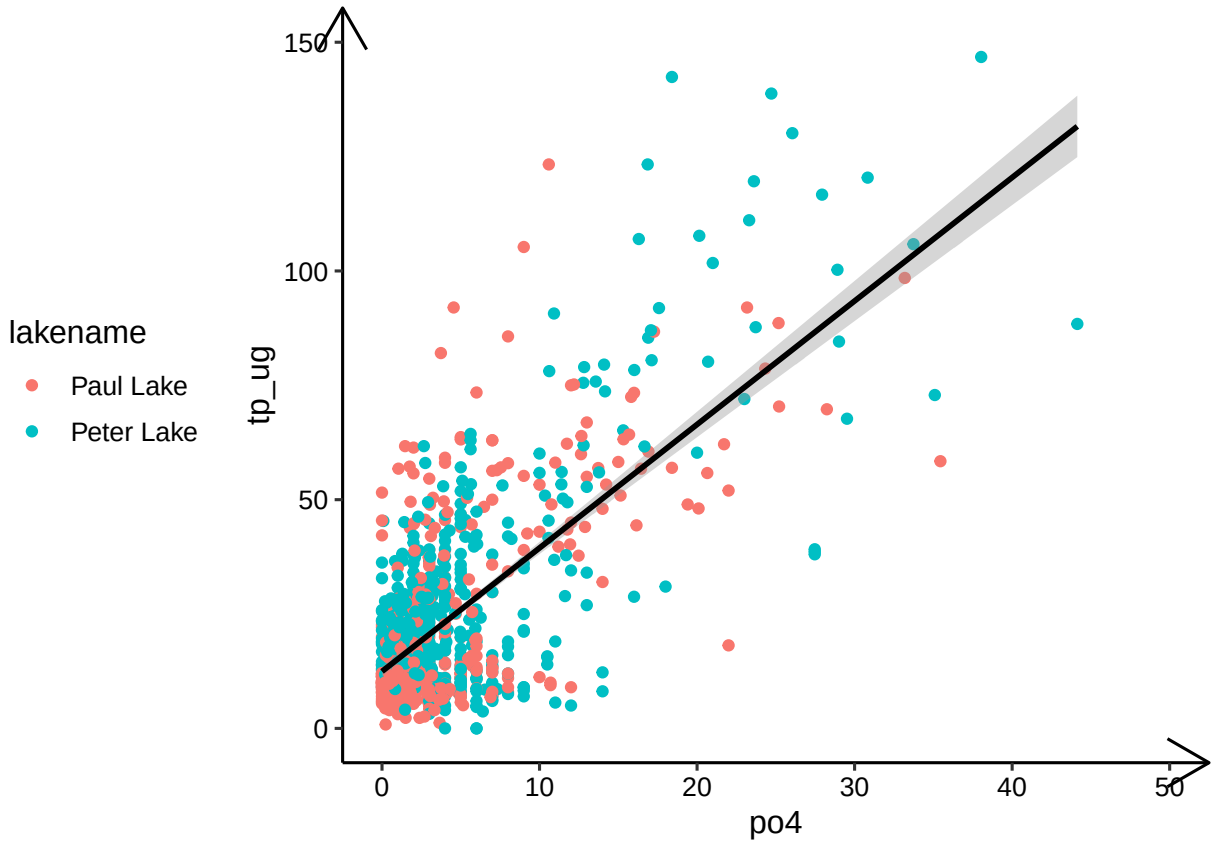
Create graphs

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 Removed one major outlier while adjusting axis ranges
Phosphates <- ggplot(PeterPaul_Processed) + geom_point(aes(x = po4, y = tp_ug, color = lakename)) +
    geom_smooth(aes(x = po4, y = tp_ug), method = lm, color = "black") + xlim(0,
    50) + ylim(0, 150) + theme1
print(Phosphates)
```

```
## `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).
```



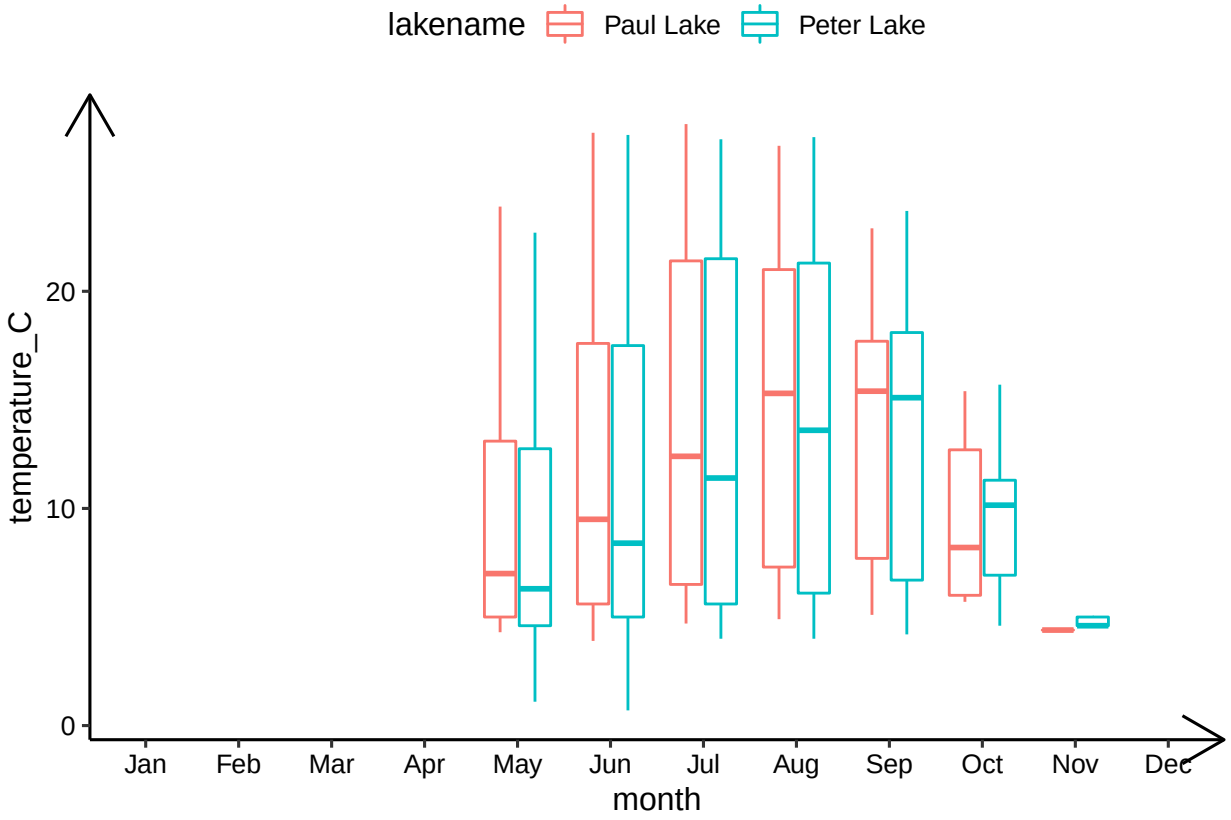
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

(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

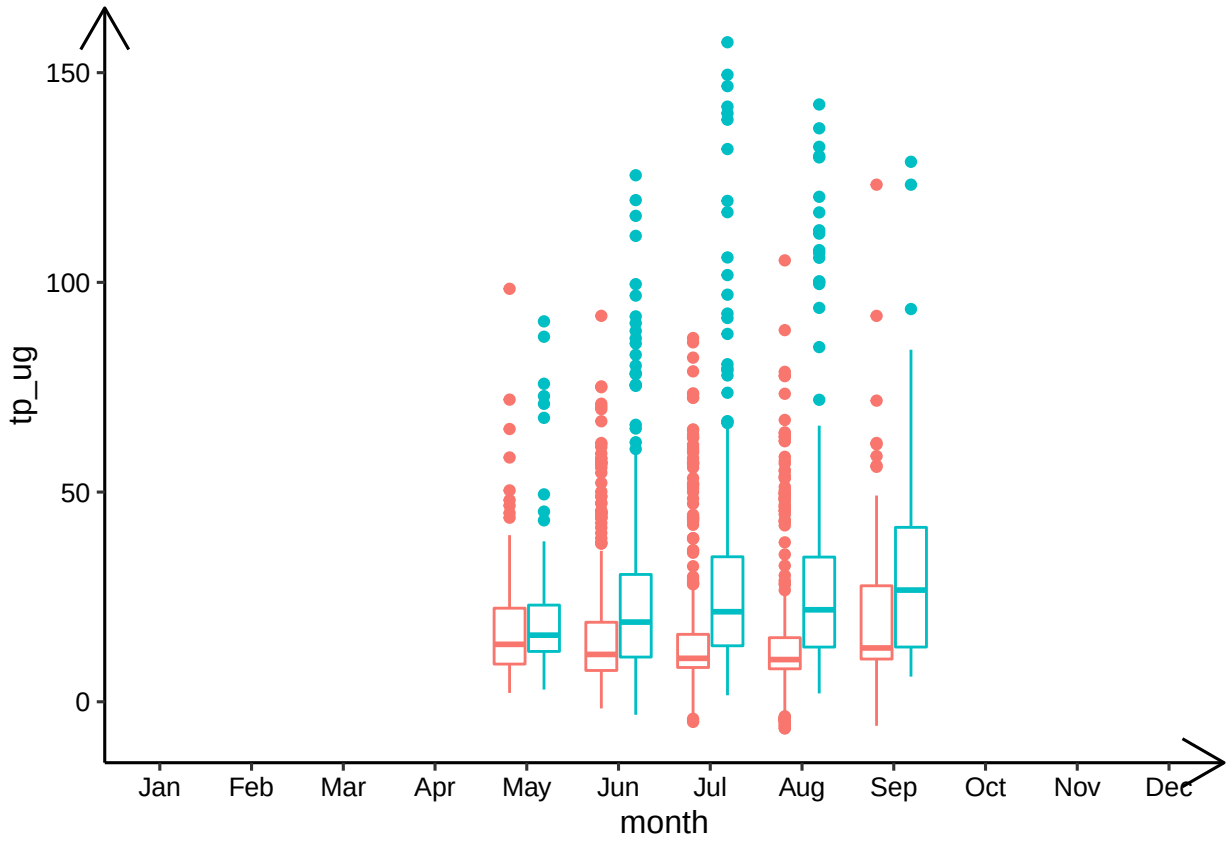
```
# 5
PeterPaul_Processed$month <- factor(PeterPaul_Processed$month, levels = c(1:12))

TempCBoxplot <- ggplot(PeterPaul_Processed, aes(x = month, y = temperature_C)) +
    geom_boxplot(aes(color = lakename)) + scale_x_discrete(label = month.abb, drop = FALSE) +
    theme1 + theme(legend.position = "top")
print(TempCBoxplot)
```

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



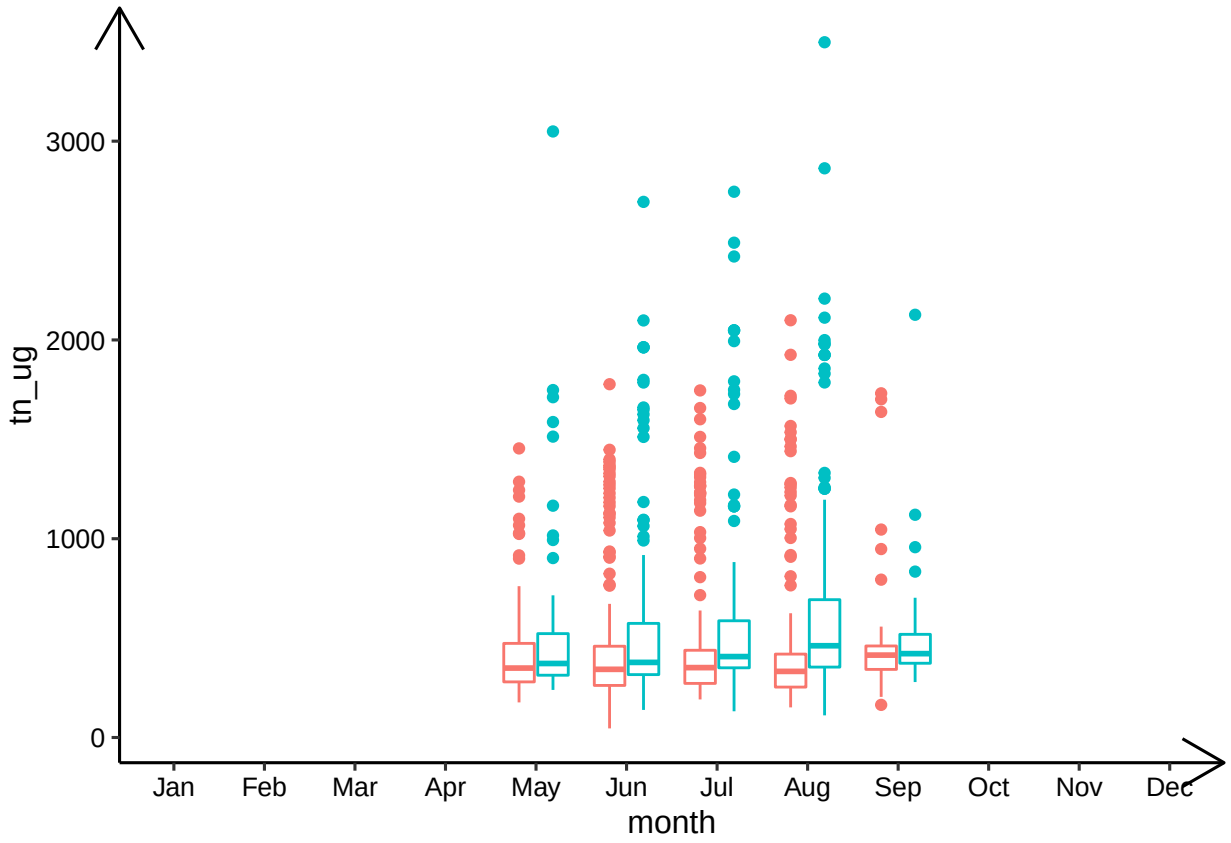

```
TPBoxplot <- ggplot(PeterPaul_Processed, aes(x = month, y = tp_ug)) + geom_boxplot(aes(color = lakename)
    scale_x_discrete(label = month.abb, drop = FALSE) + theme1 + theme(legend.position = "none")
print(TPBoxplot)
```

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




```
TNBoxplot <- ggplot(PeterPaul_Processed, aes(x = month, y = tn_ug)) + geom_boxplot(aes(color = lakename)
    scale_x_discrete(label = month.abb, drop = FALSE) + theme1 + theme(legend.position = "none")
print(TNBoxplot)
```

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

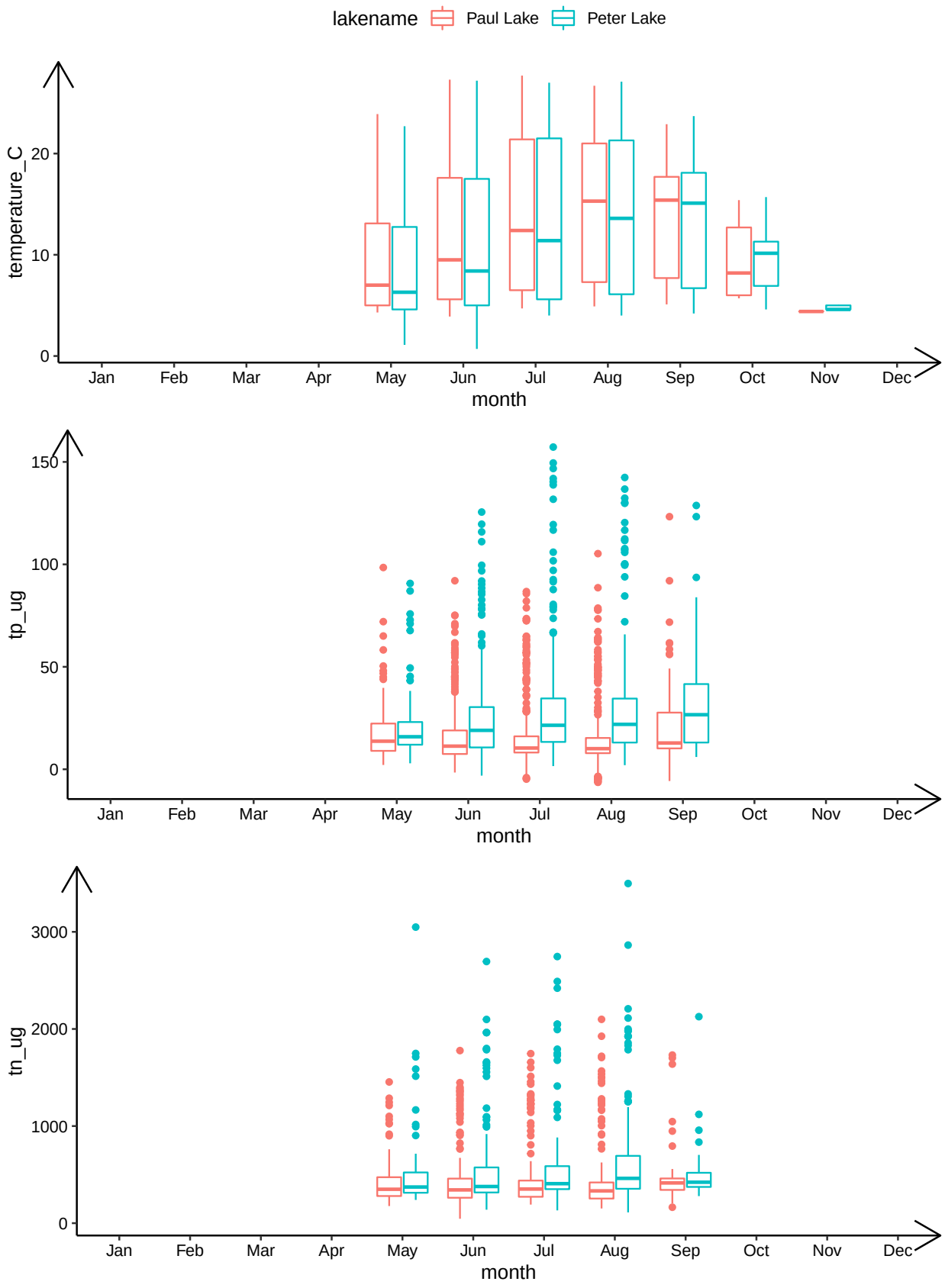



```
# Changed figure dimensions to better display cowplot
plot_grid(TempCBoxplot, TPBoxplot, TNBoxplot, nrow = 3, align = "w")
```

```
## 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).
```


lakename
Paul Lake
Peter Lake
temperature_C
0
10
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
month
tp_ug
0
50
100
150
month
tn_ug
0
1000
2000
3000
month

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

> Answer: Total phosphorus and nitrogen appear to be larger on average in Peter Lake than Paul Lake regardless of season. However, there are large ranges for both variables and the results may not be significant (we would need to run T tests based on a time period of choice). Temperature, unsurpisingly, is warmest in the summer and coldest in thr spring and fall for both lakes. Temperature appears relatively similar for both lakes.

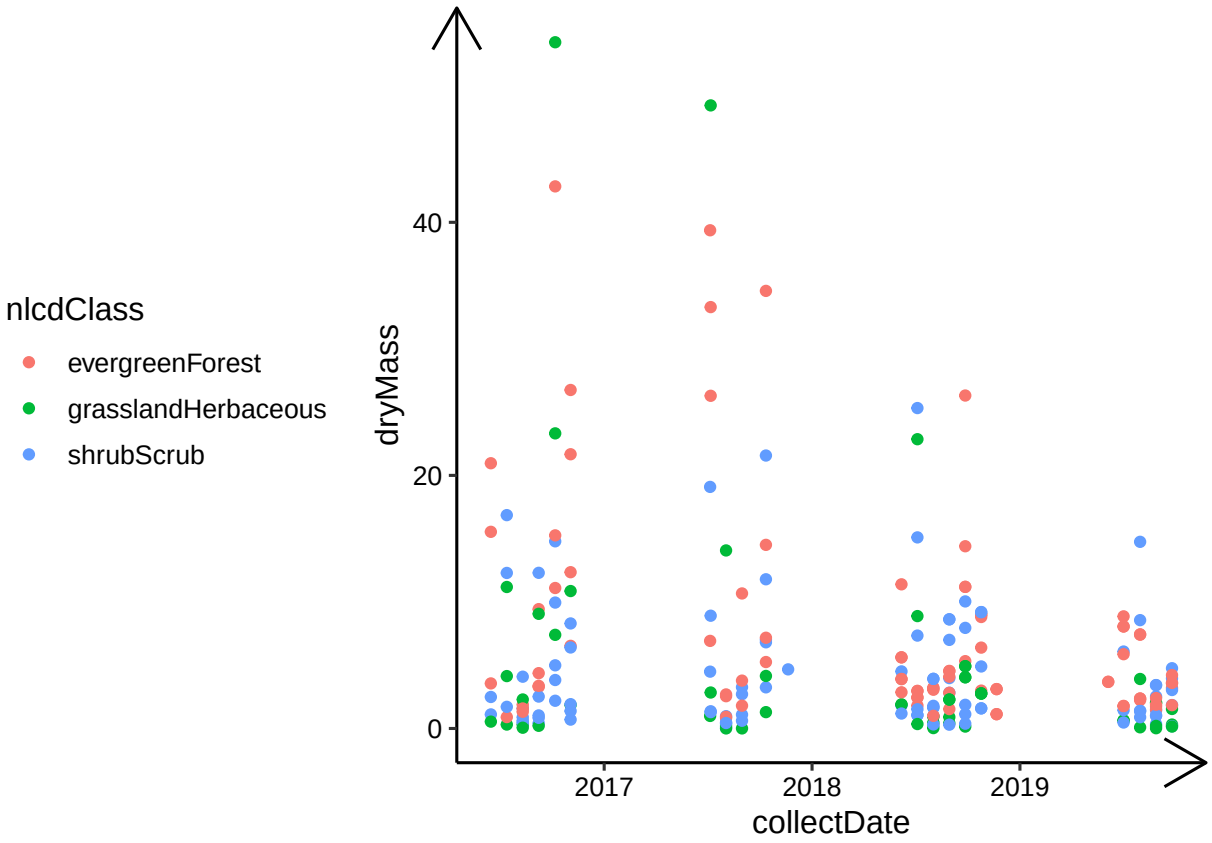
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

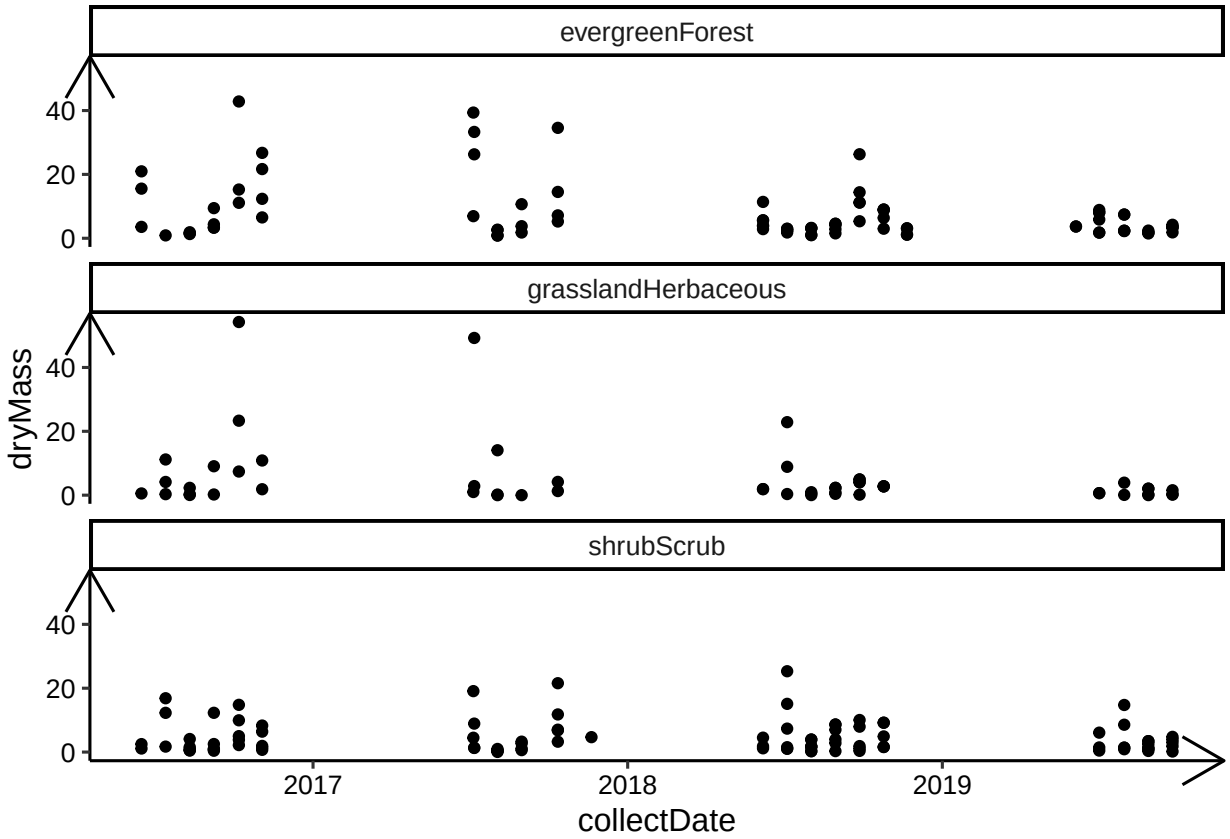
```
# 6
NIWO_Litter_Needles <- filter(NIWO_Litter, functionalGroup == "Needles")
str(NIWO_Litter_Needles)
```

```
## 'data.frame':    241 obs. of  13 variables:
##  $ plotID          : Factor w/ 12 levels "NIWO_040","NIWO_041",..: 7 4 6 11 8 9 3 5 4 8 ...
##  $ trapID          : Factor w/ 15 levels "NIWO_040_139",..: 9 5 8 13 10 11 4 6 5 10 ...
##  $ collectDate     : Date, format: "2016-06-16" "2016-06-16" ...
##  $ functionalGroup : Factor w/ 8 levels "Flowers","Leaves",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ dryMass         : num  1.11 0.54 20.96 3.56 15.54 ...
##  $ qaDryMass       : Factor w/ 2 levels "N","Y": 2 1 1 1 2 1 2 2 2 2 ...
##  $ subplotID       : int  32 40 31 32 41 31 40 31 40 41 ...
##  $ decimalLatitude : num  40 40.1 40 40 40 ...
##  $ decimalLongitude: num  -106 -106 -106 -106 -106 ...
##  $ elevation       : num  3446 3510 3382 3373 3413 ...
##  $ nlcdClass       : Factor w/ 3 levels "evergreenForest",..: 3 2 1 1 1 3 3 2 2 1 ...
##  $ plotType        : Factor w/ 1 level "tower": 1 1 1 1 1 1 1 1 1 1 ...
##  $ geodeticDatum   : Factor w/ 1 level "WGS84": 1 1 1 1 1 1 1 1 1 1 ...
```

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




```
# 7
NeedlePlot2 <- ggplot(NIWO_Litter_Needles) + geom_point(aes(x = collectDate, y = dryMass)) +
    facet_wrap(vars(nlcdClass), nrow = 3)
print(NeedlePlot2)
```



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

> Answer: Plot 7 is more effective because we can more easily see the visual differences in needle mass based on habitat type. In plot 6, data points are also overlain on top of one another and it is difficult to assess any patterns. When you look at plot 7, the needles masses by habitat type look relatively similar, but you can see slight differences (e.g. evergreen forest vs. shrubscrub in 2017).