

# Assignment 5: Data Visualization

Shidi Dai

## 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 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
getwd()

## [1] "/home/guest/R/EDA-Fall2022/Assignments"

# install.packages(tidyverse)
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()

# install.packages(lubridate)
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
NTL_LTER_Processed_Data <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)
Niwot_Ridge_Litter_Processed_Data <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)

# 2
class(NTL_LTER_Processed_Data$sampldate)

## [1] "factor"
NTL_LTER_Processed_Data$sampldate <- as.Date(NTL_LTER_Processed_Data$sampldate,
  format = "%Y-%m-%d")
class(Niwot_Ridge_Litter_Processed_Data$collectDate)

## [1] "factor"
Niwot_Ridge_Litter_Processed_Data$collectDate <- as.Date(Niwot_Ridge_Litter_Processed_Data$collectDate,
  format = "%Y-%m-%d")
```

## Define your theme

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

```
# 3
mytheme <- theme_classic(base_size = 14) + theme(axis.text = element_text(color = "black"),
  legend.position = "top")
theme_set(mytheme)
```

## Create graphs

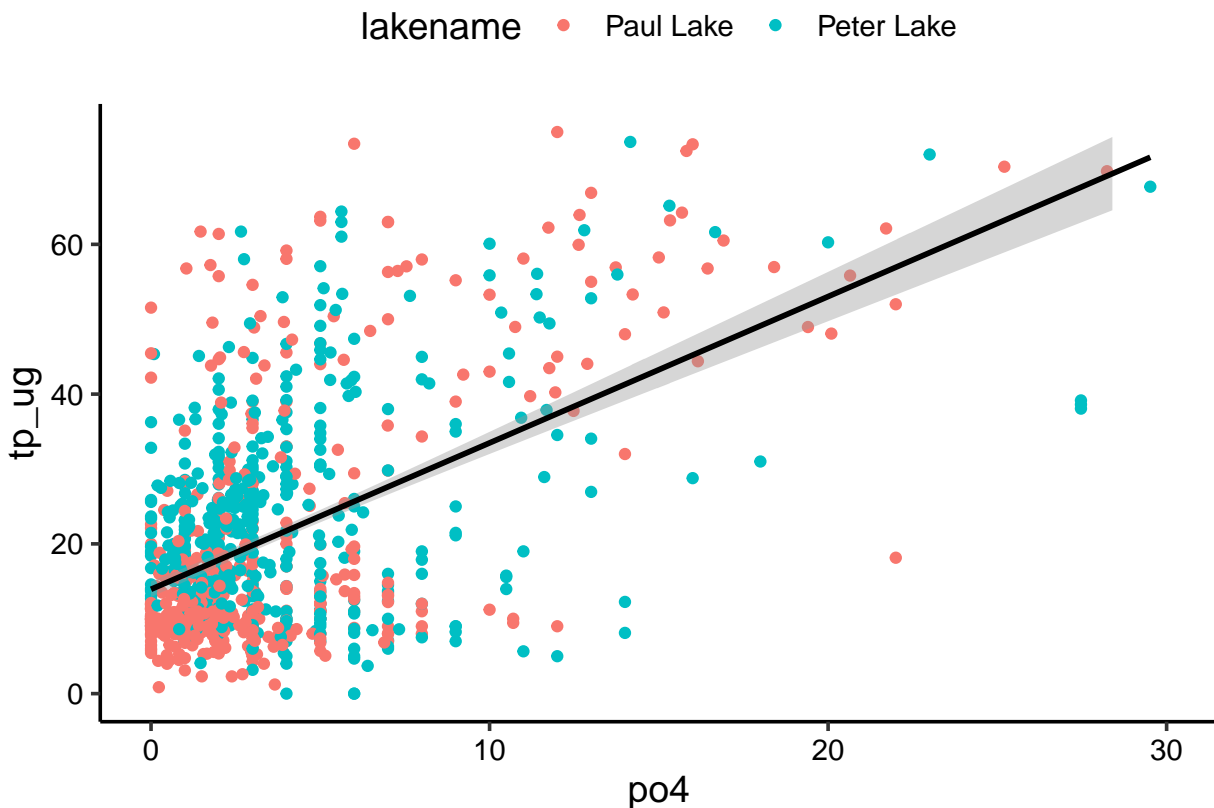
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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
plot04 <- ggplot(NTL_LTER_Processed_Data, aes(x = po4, y = tp_ug,
  color = lakename)) + geom_point() + xlim(0, 30) + ylim(0,
  75) + geom_smooth(method = lm, color = "black")
print(plot04)

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

```
## Warning: Removed 21990 rows containing non-finite values (stat_smooth).
## Warning: Removed 21990 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 built-in variable called `month.abb` that returns a list of months; see <https://r-lang.com/month-abb-in-r-with-example>

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

```
##      [1] 5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5
##     [25] 5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  6  6  6  6  6  6  6  6
##     [49] 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
##     [73] 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
##     [97] 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
##    [121] 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
##    [145] 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
##    [169] 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7
##    [193] 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
##    [217] 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
##    [241] 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
##    [265] 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
##    [289] 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
```

```

## [313] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [337] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [361] 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [385] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [409] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [433] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [457] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [481] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [505] 8 8 8 8 8 8 8 8 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [529] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [553] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [577] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [601] 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [625] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [649] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [673] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [697] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [721] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [745] 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [769] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [793] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [817] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [841] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [865] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [889] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [913] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [937] 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8
## [961] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [985] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [1009] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [1033] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [1057] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [1081] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9
## [1105] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [1129] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [1153] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [1177] 9 9 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [1201] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6
## [1225] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1249] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1273] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1297] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1321] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1345] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1369] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [1393] 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1417] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1441] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1465] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1489] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1513] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1537] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [1561] 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8
## [1585] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

```

[illegible]

[illegible]

[illegible]

###	[5497]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
###	[5521]	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5545]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5569]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5593]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5617]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5641]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5665]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5689]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
###	[5713]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	8
###	[5737]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
###	[5761]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
###	[5785]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
###	[5809]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
###	[5833]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
###	[5857]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
###	[5881]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	9	9	9
###	[5905]	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
###	[5929]	9	9	9	9	9	9	9	9	9	9	9	9	5	5	5	5	5	5	5	5	5
##	[5953]	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
##	[5977]	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
##	[6001]	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6
##	[6025]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[6049]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[6073]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[6097]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[6121]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[6145]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[6169]	6	6	6																		



[illegible]

##	[8089]	9	9	9	9	9	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
##	[8113]	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
##	[8137]	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	6	6
##	[8161]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8185]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8209]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8233]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8257]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8281]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8305]	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
##	[8329]	6	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7
##	[8353]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
##	[8377]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
##	[8401]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
##	[8425]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
##	[8449]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
##	[8473]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
##	[8497]	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	8	8	8	8	8	8
##	[8521]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8545]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8569]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8593]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8617]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8641]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8665]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8689]	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
##	[8713]	8	8	8	8	8	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
##	[8737]	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	5
##	[8761]	5	5	5																		

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

```

## [21049] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [21073] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [21097] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [21121] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [21145] 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6
## [21169] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21193] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21217] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21241] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21265] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21289] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21313] 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21337] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21361] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21385] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21409] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21433] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21457] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21481] 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [21505] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [21529] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [21553] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [21577] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [21601] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 5 5 5 5 5 5 5 5 5
## [21625] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21649] 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21673] 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8
## [21697] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 5
## [21721] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6
## [21745] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21769] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7
## [21793] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21817] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21841] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [21865] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9
## [21889] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 5 5 5 5 5 5 5 5
## [21913] 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [21937] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7
## [21961] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [21985] 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22009] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9
## [22033] 9 9 9 9 9 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6
## [22057] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22081] 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22105] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8
## [22129] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22153] 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 5
## [22177] 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22201] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22225] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22249] 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22273] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22297] 8 8 9 9 9 9 9 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6
## [22321] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7

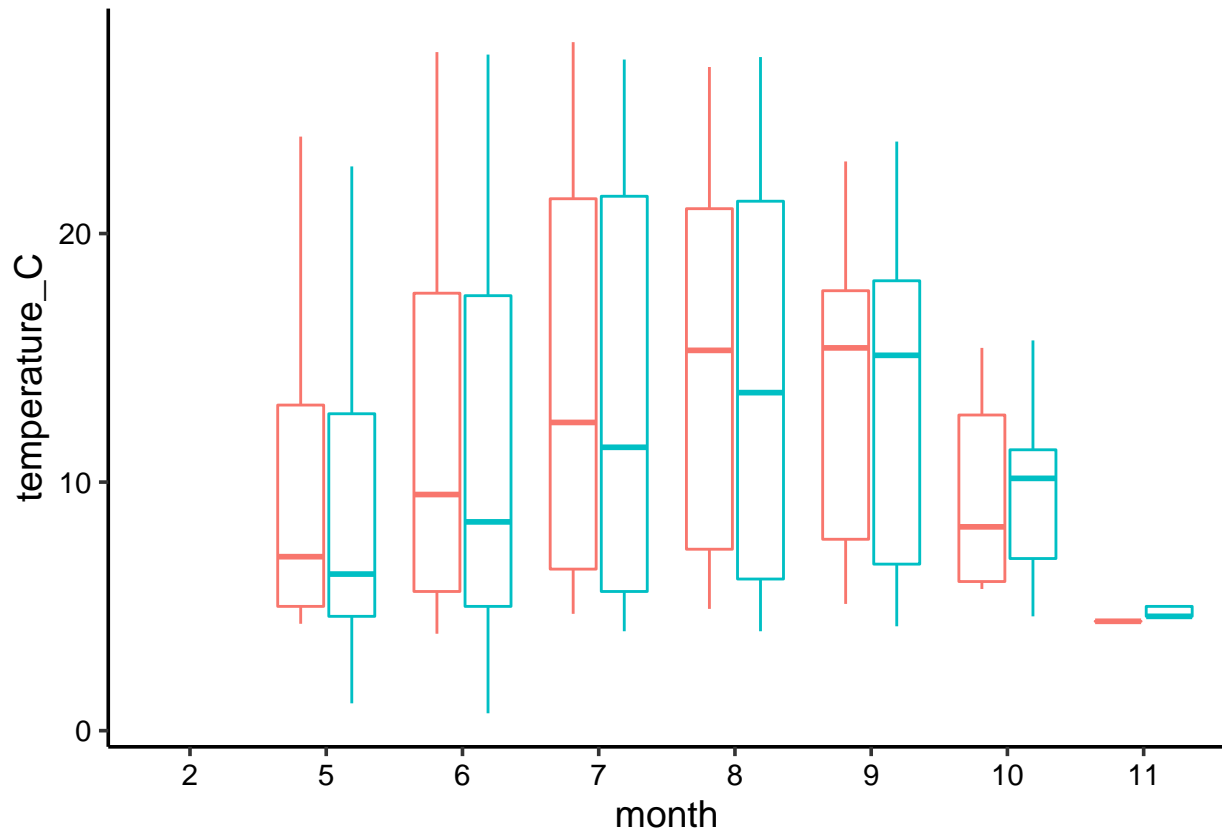
```

```
## [22345] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22369] 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22393] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22417] 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22441] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7
## [22465] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22489] 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22513] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22537] 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22561] 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7
## [22585] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8
## [22609] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22633] 8 8 8 8 8 8 8 8 8 8 8 8 5 5 5 5 5 5 6 6 6 6
## [22657] 6 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 5 5 5 5
## [22681] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22705] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22729] 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22753] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22777] 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22801] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22825] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 5 5 5
## [22849] 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22873] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
## [22897] 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7
## [22921] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
## [22945] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8
## [22969] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [22993] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## Levels: 1 2 3 4 5 6 7 8 9 10 11 12
```

```
# boxplot of temperature
```

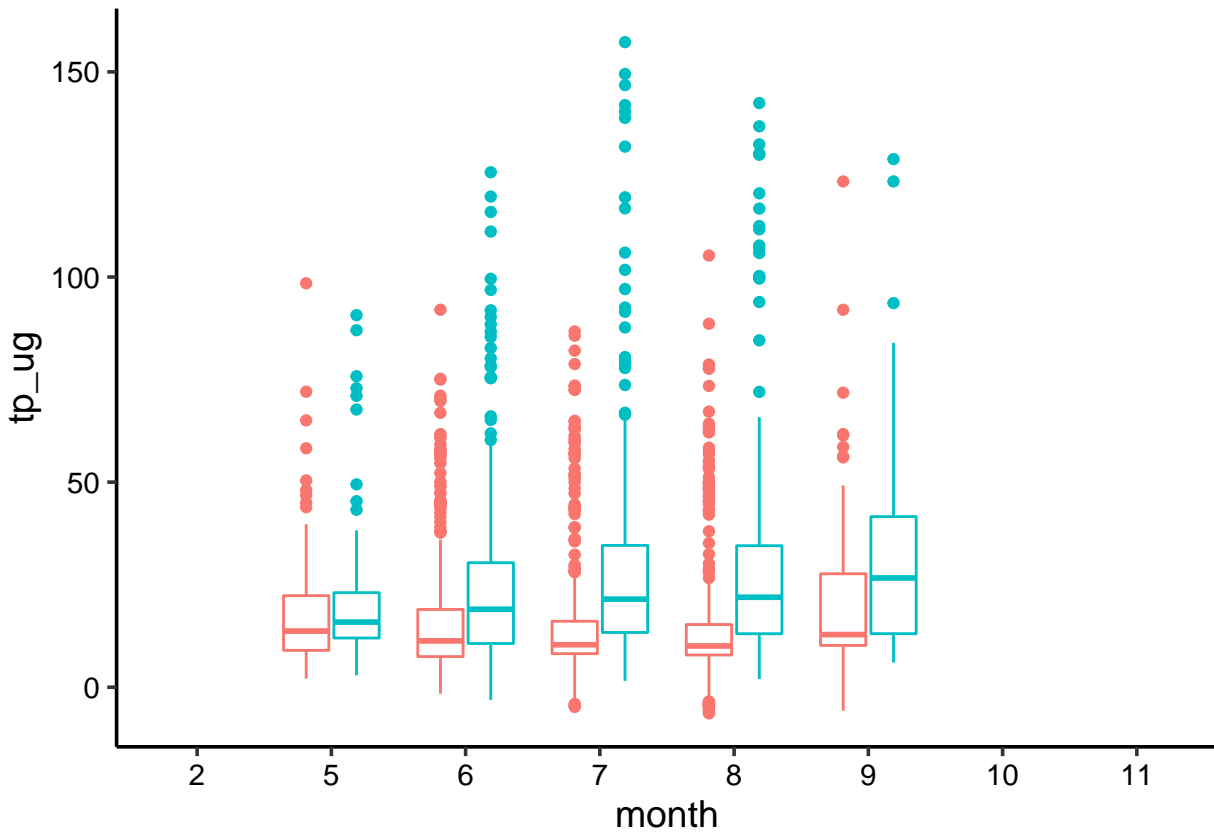
```
Temperature_boxplot <- ggplot(NTL_LTER_Processed_Data, aes(x = month,
  y = temperature_C)) + geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(drop = FALSE) + theme(legend.position = "none")
print(Temperature_boxplot)
```

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



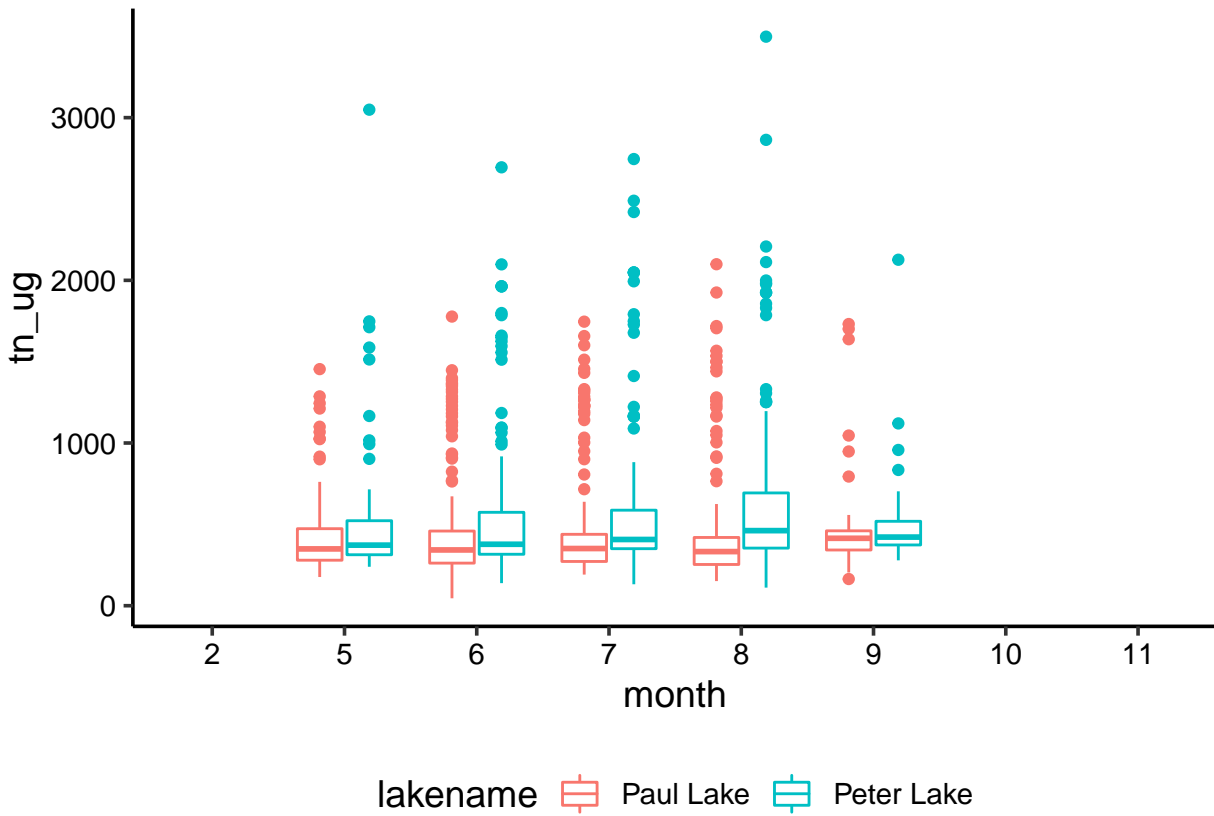
```
# boxplot of TP
TP_boxplot <- ggplot(NTL_LTER_Processed_Data, aes(x = month,
  y = tp_ug)) + geom_boxplot(aes(color = lakename)) + scale_x_discrete(drop = FALSE) +
  theme(legend.position = "none")
print(TP_boxplot)
```

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



```
# boxplot of TN
TN_boxplot <- ggplot(NTL_LTER_Processed_Data, aes(x = month,
  y = tn_ug)) + geom_boxplot(aes(color = lakename)) + scale_x_discrete(drop = FALSE) +
  theme(legend.position = "bottom")
print(TN_boxplot)
```

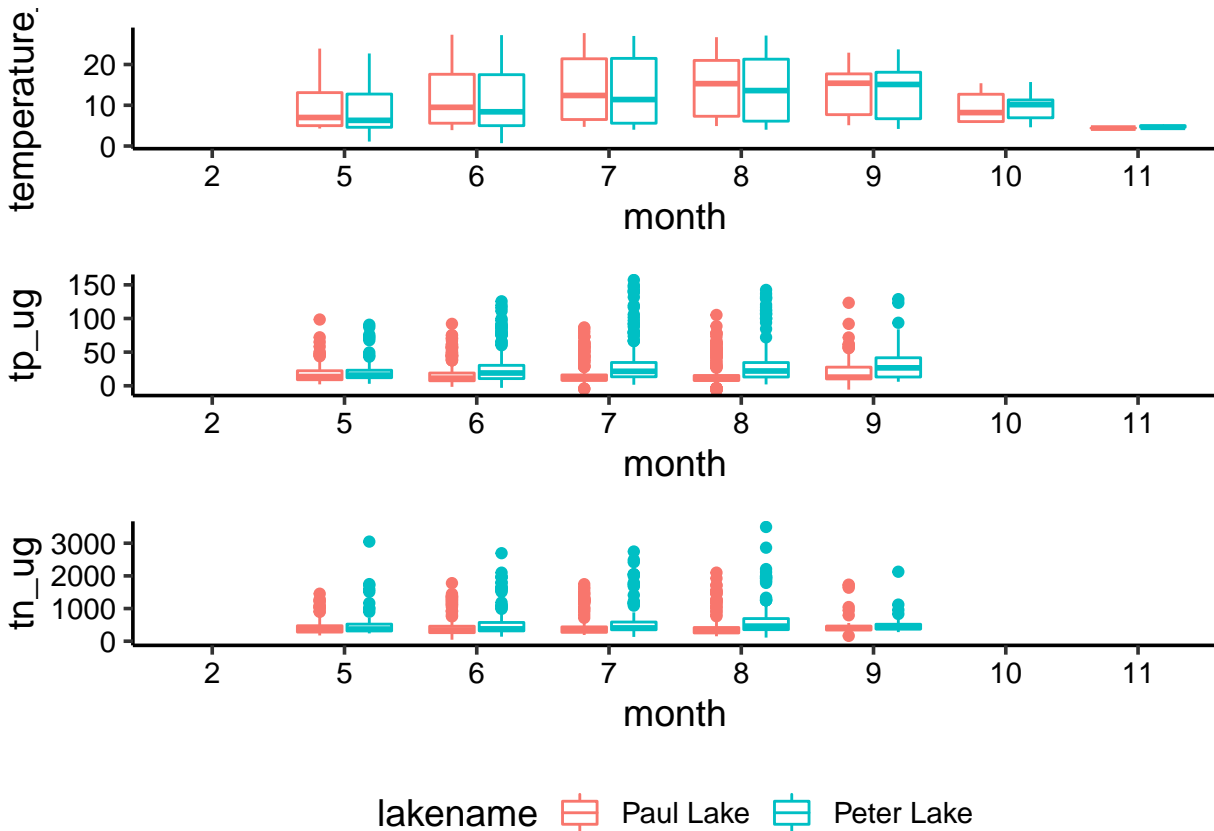
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```



```
library(cowplot)
plot_grid(Temperature_boxplot, TP_boxplot, TN_boxplot, nrow = 3,
  align = "v", rel_heights = c(2, 2, 3))
```

```
## 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).
```



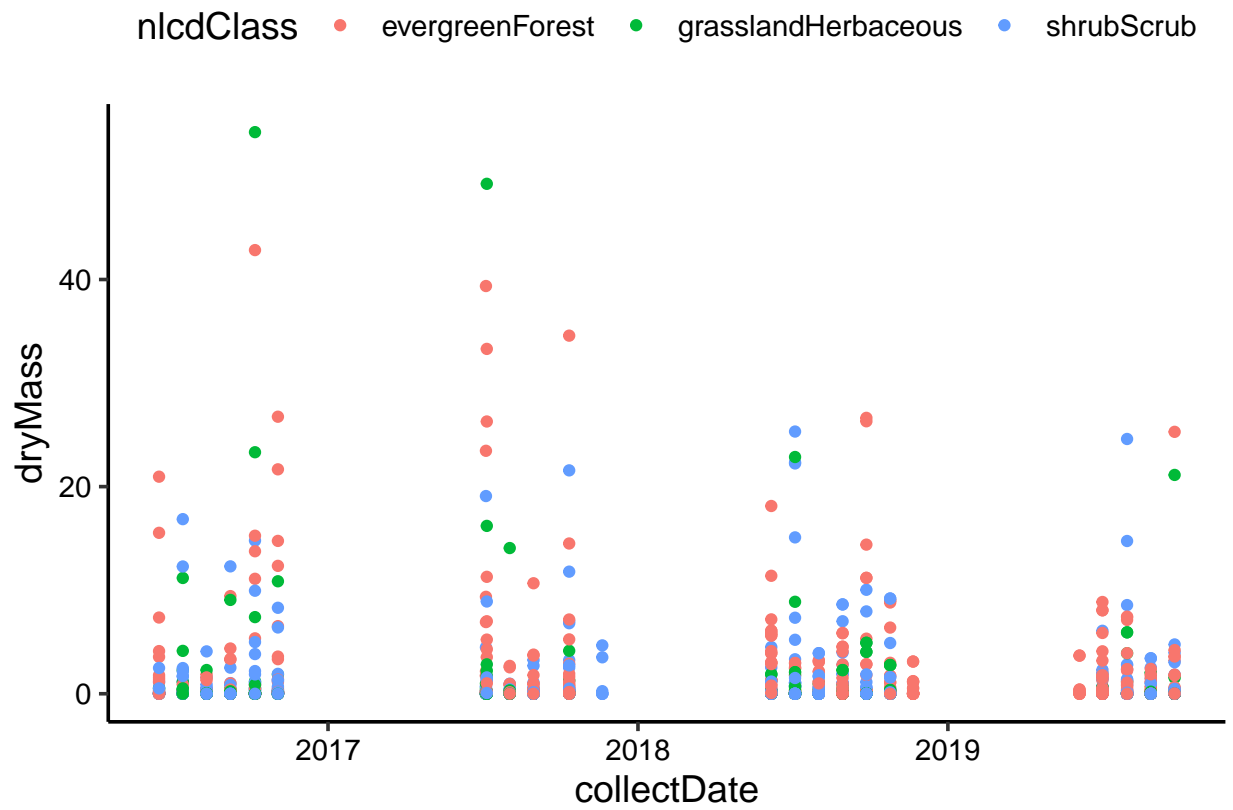


Question: What do you observe about the variables of interest over seasons and between lakes?

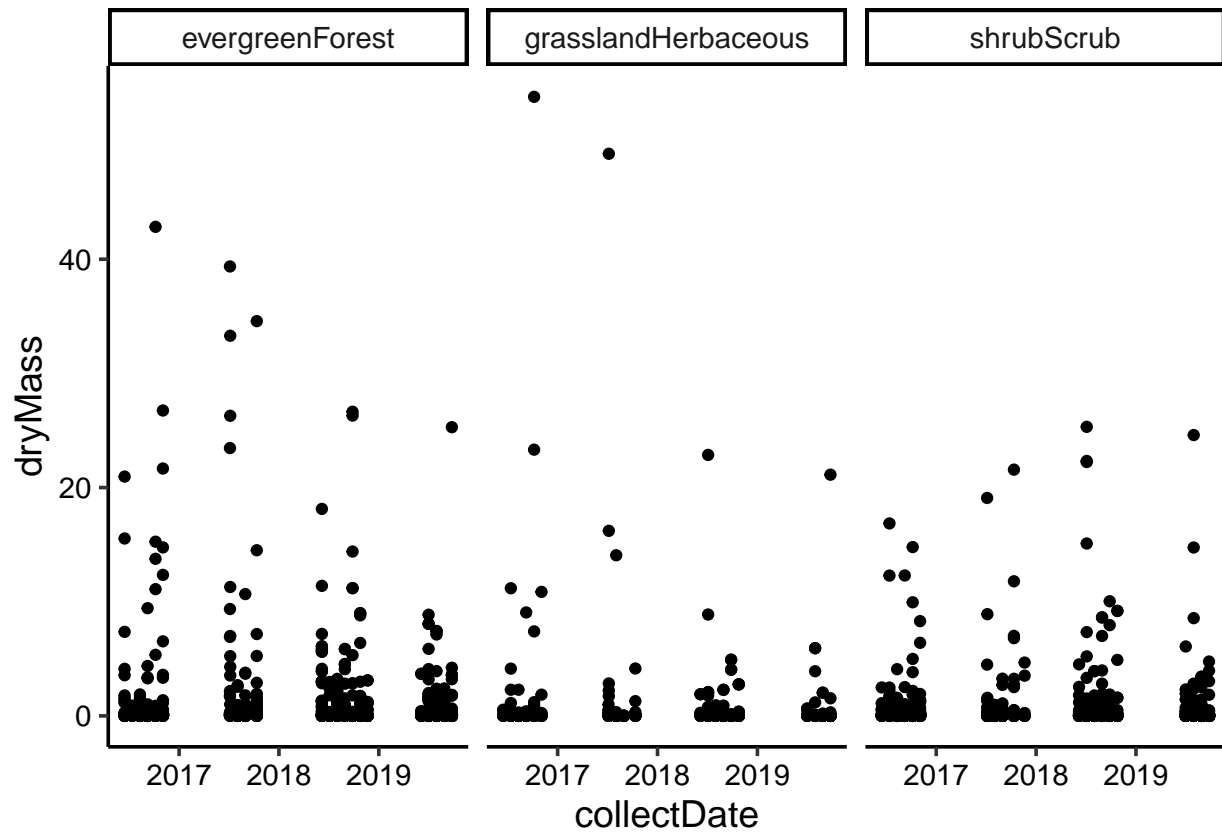
Answer: Temperatures are higher in both lakes in the summer (July, August, September) and relatively lower before summer in May and June, and lower in November. The variable `tp_ug` is always higher in Peter Lake than in Paul lake from May till September. The variable `tn_ug` is also higher in Peter lake than in Paul lake from May to September. The difference in `tn_ug` between two lakes are bigger in July and August (summer). `Tn_ug` and `tp_ug` in the same lake are approximately the same throughout 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.

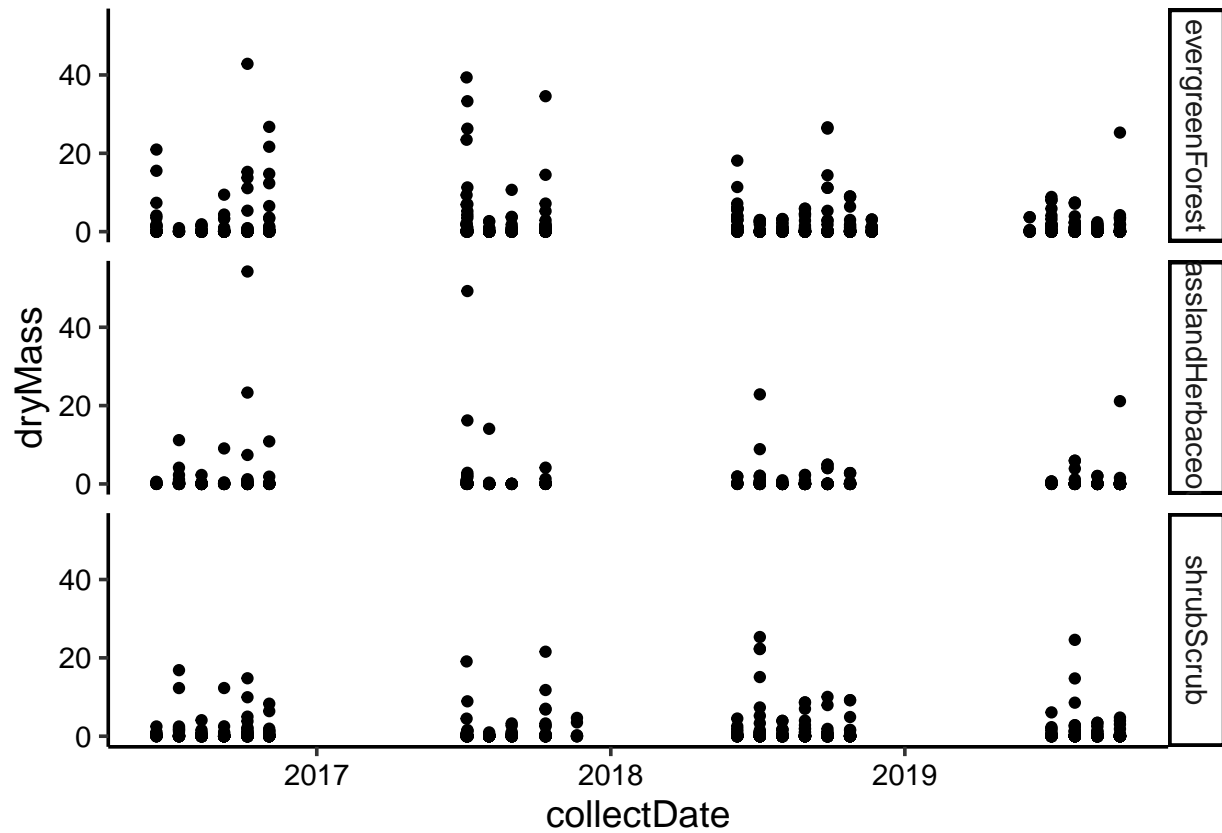
```
# 6
Needles_plot <- ggplot(subset(Niwot_Ridge_Litter_Processed_Data,
  functionalGroup = "Needles"), aes(x = collectDate, y = dryMass,
  color = nlcdClass)) + geom_point()
print(Needles_plot)
```



```
# 7
Needles_faceted <- ggplot(subset(Niwot_Ridge_Litter_Processed_Data,
  functionalGroup = "Needles"), aes(x = collectDate, y = dryMass)) +
  geom_point() + facet_wrap(vars(nlcdClass), ncol = 3)
print(Needles_faceted)
```



```
Needles_faceted2 <- ggplot(subset(Niwot_Ridge_Litter_Processed_Data,
  functionalGroup = "Needles"), aes(x = collectDate, y = dryMass)) +
  geom_point() + facet_grid("nlcdClass")
print(Needles_faceted2)
```



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

Answer: I think plot 7, my `Needles_faceted` plot is more effective. It separated the three nlcdclass, evergreenforest, grasslandherbaceous, and shrubscrub into three rows to show their distribution of drymass by collectdate. It is clearer than three colors messing together in the same graph (plot 6), which is hard to see the distribution of each nlcdclass. With plots in 7, we can easily compare the drymass and distributions between the three nlcdclasses.