

# Assignment 5: Data Visualization

Sam Campbell

Spring 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. 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
library(tidyverse);library(lubridate);library(here);library(cowplot)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## Loading required package: timechange
##
```

```
##
## Attaching package: 'lubridate'
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
## here() starts at /home/guest/R/EDA-Spring2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp

getwd()

## [1] "/home/guest/R/EDA-Spring2023"

NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"), stringsAsFactors = FALSE)

NEON_NIWO_Litter_mass_trap_Processed <- read.csv(
  here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)

## [1] "factor"

class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)

## [1] "factor"

#change dates from factor to date format
NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate <- ymd(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)

NEON_NIWO_Litter_mass_trap_Processed$collectDate <- ymd(NEON_NIWO_Litter_mass_trap_Processed$collectDate)

#check class again
class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)

## [1] "Date"
```

```
class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)
```

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

## 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
samtheme <- theme_dark(base_size = 12) +
  theme(axis.text = element_text(color = "blue"),
        legend.position = "left")
```

## 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

#first try without separate aesthetics for 2 lakes
#plot4 <-
  #ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
          #aes(x=po4,
               #y=tp_ug))) +
  #geom_point()
#print(plot4)

#second try with faceting, successful
#plot4 <-
  #ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
          #aes(x=po4,
               #y=tp_ug))) +
  #geom_point() +
  #facet_wrap(vars(lakename), nrow = 3) +
#print(plot4)

#third try with samtheme
#plot4 <-
  #ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
```

```

                # (aes(x=po4,
                        #y=tp_ug))) +
#geom_point() +
#facet_wrap(vars(lakename), nrow = 3)
#samtheme
#print(plot4)

#4th try with line of best fit
#plot4 <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
          # (aes(x=po4,
                  #y=tp_ug))) +
#geom_point() +
#geom_smooth(method = lm,
              #color="black") +
#facet_wrap(vars(lakename), nrow = 3)
#samtheme
#print(plot4)

#5th version with adjusted axes
#plot4 <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
          # (aes(x=po4,
                  #y=tp_ug))) +
#geom_point() +
#xlim(0, 50) +
#geom_smooth(method = lm,
              #color="black") +
#facet_wrap(vars(lakename), nrow = 3)
#samtheme
#print(plot4)

#Final version with lake colors instead of facets per John's feedback
plot4 <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
          (aes(x=po4,
                y=tp_ug,
                color=lakename))) +
  geom_point() +
  xlim(0, 50) +
  geom_smooth(method = lm,
              color="black")
  samtheme

```

```

## List of 94
## $ line                               :List of 6
## ..$ colour                          : chr "black"
## ..$ linewidth                       : num 0.545
## ..$ linetype                        : num 1
## ..$ lineend                         : chr "butt"
## ..$ arrow                          : logi FALSE
## ..$ inherit.blank                  : logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_line" "element"

```

```

## $ rect                                     :List of 5
## ..$ fill                                : chr "white"
## ..$ colour                             : chr "black"
## ..$ linewidth                          : num 0.545
## ..$ linetype                           : num 1
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ text                                     :List of 11
## ..$ family                             : chr ""
## ..$ face                               : chr "plain"
## ..$ colour                             : chr "black"
## ..$ size                               : num 12
## ..$ hjust                              : num 0.5
## ..$ vjust                              : num 0.5
## ..$ angle                              : num 0
## ..$ lineheight                         : num 0.9
## ..$ margin                             : 'margin' num [1:4] 0points 0points 0points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug                              : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ title                                 : NULL
## $ aspect.ratio                         : NULL
## $ axis.title                           : NULL
## $ axis.title.x                         :List of 11
## ..$ family                             : NULL
## ..$ face                               : NULL
## ..$ colour                             : NULL
## ..$ size                               : NULL
## ..$ hjust                              : NULL
## ..$ vjust                              : num 1
## ..$ angle                              : NULL
## ..$ lineheight                         : NULL
## ..$ margin                             : 'margin' num [1:4] 3points 0points 0points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug                              : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.top                    :List of 11
## ..$ family                             : NULL
## ..$ face                               : NULL
## ..$ colour                             : NULL
## ..$ size                               : NULL
## ..$ hjust                              : NULL
## ..$ vjust                              : num 0
## ..$ angle                              : NULL
## ..$ lineheight                         : NULL
## ..$ margin                             : 'margin' num [1:4] 0points 0points 3points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug                              : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.bottom                  : NULL
## $ axis.title.y                         :List of 11

```

```

## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : NULL
## ..$ vjust       : num 1
## ..$ angle       : num 90
## ..$ lineheight  : NULL
## ..$ margin      : 'margin' num [1:4] 0points 3points 0points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.y.left      : NULL
## $ axis.title.y.right     :List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : NULL
## ..$ vjust       : num 0
## ..$ angle       : num -90
## ..$ lineheight  : NULL
## ..$ margin      : 'margin' num [1:4] 0points 0points 0points 3points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text           :List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : chr "blue"
## ..$ size        : 'rel' num 0.8
## ..$ hjust       : NULL
## ..$ vjust       : NULL
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : NULL
## ..$ debug       : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x         :List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : NULL
## ..$ vjust       : num 1
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : 'margin' num [1:4] 2.4points 0points 0points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE

```

```

##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.top           :List of 11
##   ..$ family               : NULL
##   ..$ face                 : NULL
##   ..$ colour               : NULL
##   ..$ size                 : NULL
##   ..$ hjust                : NULL
##   ..$ vjust                : num 0
##   ..$ angle                : NULL
##   ..$ lineheight           : NULL
##   ..$ margin               : 'margin' num [1:4] 0points 0points 2.4points 0points
##   ..- attr(*, "unit")= int 8
##   ..$ debug                : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.bottom       : NULL
## $ axis.text.y              :List of 11
##   ..$ family               : NULL
##   ..$ face                 : NULL
##   ..$ colour               : NULL
##   ..$ size                 : NULL
##   ..$ hjust                : num 1
##   ..$ vjust                : NULL
##   ..$ angle                : NULL
##   ..$ lineheight           : NULL
##   ..$ margin               : 'margin' num [1:4] 0points 2.4points 0points 0points
##   ..- attr(*, "unit")= int 8
##   ..$ debug                : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.y.left         : NULL
## $ axis.text.y.right        :List of 11
##   ..$ family               : NULL
##   ..$ face                 : NULL
##   ..$ colour               : NULL
##   ..$ size                 : NULL
##   ..$ hjust                : num 0
##   ..$ vjust                : NULL
##   ..$ angle                : NULL
##   ..$ lineheight           : NULL
##   ..$ margin               : 'margin' num [1:4] 0points 0points 0points 2.4points
##   ..- attr(*, "unit")= int 8
##   ..$ debug                : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.ticks                :List of 6
##   ..$ colour               : chr "grey20"
##   ..$ linewidth            : 'rel' num 0.5
##   ..$ linetype              : NULL
##   ..$ lineend               : NULL
##   ..$ arrow                 : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ axis.ticks.x              : NULL

```

```

## $ axis.ticks.x.top          : NULL
## $ axis.ticks.x.bottom      : NULL
## $ axis.ticks.y             : NULL
## $ axis.ticks.y.left        : NULL
## $ axis.ticks.y.right       : NULL
## $ axis.ticks.length         : 'simpleUnit' num 3points
##   .- attr(*, "unit")= int 8
## $ axis.ticks.length.x       : NULL
## $ axis.ticks.length.x.top   : NULL
## $ axis.ticks.length.x.bottom : NULL
## $ axis.ticks.length.y       : NULL
## $ axis.ticks.length.y.left  : NULL
## $ axis.ticks.length.y.right : NULL
## $ axis.line                 : list()
##   .- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.line.x               : NULL
## $ axis.line.x.top           : NULL
## $ axis.line.x.bottom        : NULL
## $ axis.line.y               : NULL
## $ axis.line.y.left          : NULL
## $ axis.line.y.right         : NULL
## $ legend.background         :List of 5
##   ..$ fill                  : NULL
##   ..$ colour                 : logi NA
##   ..$ linewidth              : NULL
##   ..$ linetype               : NULL
##   ..$ inherit.blank: logi TRUE
##   .- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ legend.margin              : 'margin' num [1:4] 6points 6points 6points 6points
##   .- attr(*, "unit")= int 8
## $ legend.spacing             : 'simpleUnit' num 12points
##   .- attr(*, "unit")= int 8
## $ legend.spacing.x           : NULL
## $ legend.spacing.y           : NULL
## $ legend.key                  :List of 5
##   ..$ fill                   : chr "grey50"
##   ..$ colour                 : logi NA
##   ..$ linewidth              : NULL
##   ..$ linetype               : NULL
##   ..$ inherit.blank: logi TRUE
##   .- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ legend.key.size             : 'simpleUnit' num 1.2lines
##   .- attr(*, "unit")= int 3
## $ legend.key.height           : NULL
## $ legend.key.width            : NULL
## $ legend.text                 :List of 11
##   ..$ family                 : NULL
##   ..$ face                   : NULL
##   ..$ colour                 : NULL
##   ..$ size                   : 'rel' num 0.8
##   ..$ hjust                  : NULL
##   ..$ vjust                  : NULL
##   ..$ angle                  : NULL
##   ..$ lineheight             : NULL

```



```

## ..$ margin      : NULL
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ legend.text.align      : NULL
## $ legend.title           :List of 11
## ..$ family             : NULL
## ..$ face                : NULL
## ..$ colour              : NULL
## ..$ size                : NULL
## ..$ hjust               : num 0
## ..$ vjust               : NULL
## ..$ angle               : NULL
## ..$ lineheight          : NULL
## ..$ margin              : NULL
## ..$ debug               : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ legend.title.align     : NULL
## $ legend.position        : chr "left"
## $ legend.direction       : NULL
## $ legend.justification   : chr "center"
## $ legend.box              : NULL
## $ legend.box.just        : NULL
## $ legend.box.margin      : 'margin' num [1:4] 0cm 0cm 0cm 0cm
## ..- attr(*, "unit")= int 1
## $ legend.box.background  : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.box.spacing     : 'simpleUnit' num 12points
## ..- attr(*, "unit")= int 8
## $ panel.background       :List of 5
## ..$ fill                 : chr "grey50"
## ..$ colour               : logi NA
## ..$ linewidth            : NULL
## ..$ linetype              : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ panel.border           : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ panel.spacing          : 'simpleUnit' num 6points
## ..- attr(*, "unit")= int 8
## $ panel.spacing.x        : NULL
## $ panel.spacing.y        : NULL
## $ panel.grid              :List of 6
## ..$ colour               : chr "grey42"
## ..$ linewidth            : NULL
## ..$ linetype              : NULL
## ..$ lineend               : NULL
## ..$ arrow                 : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.grid.major        :List of 6
## ..$ colour               : NULL
## ..$ linewidth            : 'rel' num 0.5

```

```

## ..$ linetype      : NULL
## ..$ lineend       : NULL
## ..$ arrow         : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.grid.minor      :List of 6
## ..$ colour         : NULL
## ..$ linewidth      : 'rel' num 0.25
## ..$ linetype       : NULL
## ..$ lineend        : NULL
## ..$ arrow          : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.grid.major.x    : NULL
## $ panel.grid.major.y    : NULL
## $ panel.grid.minor.x    : NULL
## $ panel.grid.minor.y    : NULL
## $ panel.ontop           : logi FALSE
## $ plot.background       :List of 5
## ..$ fill            : NULL
## ..$ colour          : chr "white"
## ..$ linewidth       : NULL
## ..$ linetype        : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ plot.title            :List of 11
## ..$ family           : NULL
## ..$ face             : NULL
## ..$ colour           : NULL
## ..$ size             : 'rel' num 1.2
## ..$ hjust            : num 0
## ..$ vjust            : num 1
## ..$ angle            : NULL
## ..$ lineheight       : NULL
## ..$ margin           : 'margin' num [1:4] 0points 0points 6points 0points
## ..- attr(*, "unit")= int 8
## ..$ debug            : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.title.position   : chr "panel"
## $ plot.subtitle         :List of 11
## ..$ family           : NULL
## ..$ face             : NULL
## ..$ colour           : NULL
## ..$ size             : NULL
## ..$ hjust            : num 0
## ..$ vjust            : num 1
## ..$ angle            : NULL
## ..$ lineheight       : NULL
## ..$ margin           : 'margin' num [1:4] 0points 0points 6points 0points
## ..- attr(*, "unit")= int 8
## ..$ debug            : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"

```

```

## $ plot.caption          :List of 11
##   ..$ family           : NULL
##   ..$ face              : NULL
##   ..$ colour            : NULL
##   ..$ size              : 'rel' num 0.8
##   ..$ hjust             : num 1
##   ..$ vjust             : num 1
##   ..$ angle             : NULL
##   ..$ lineheight        : NULL
##   ..$ margin            : 'margin' num [1:4] 6points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug             : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.caption.position : chr "panel"
## $ plot.tag              :List of 11
##   ..$ family           : NULL
##   ..$ face              : NULL
##   ..$ colour            : NULL
##   ..$ size              : 'rel' num 1.2
##   ..$ hjust             : num 0.5
##   ..$ vjust             : num 0.5
##   ..$ angle             : NULL
##   ..$ lineheight        : NULL
##   ..$ margin            : NULL
##   ..$ debug             : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.tag.position     : chr "topleft"
## $ plot.margin           : 'margin' num [1:4] 6points 6points 6points 6points
##   ..- attr(*, "unit")= int 8
## $ strip.background      :List of 5
##   ..$ fill              : chr "grey15"
##   ..$ colour            : logi NA
##   ..$ linewidth         : NULL
##   ..$ linetype          : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ strip.background.x    : NULL
## $ strip.background.y    : NULL
## $ strip.clip            : chr "inherit"
## $ strip.placement       : chr "inside"
## $ strip.text            :List of 11
##   ..$ family           : NULL
##   ..$ face              : NULL
##   ..$ colour            : chr "grey90"
##   ..$ size              : 'rel' num 0.8
##   ..$ hjust             : NULL
##   ..$ vjust             : NULL
##   ..$ angle             : NULL
##   ..$ lineheight        : NULL
##   ..$ margin            : 'margin' num [1:4] 4.8points 4.8points 4.8points 4.8points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug             : NULL

```

```

## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.x : NULL
## $ strip.text.y :List of 11
## ..$ family : NULL
## ..$ face : NULL
## ..$ colour : NULL
## ..$ size : NULL
## ..$ hjust : NULL
## ..$ vjust : NULL
## ..$ angle : num -90
## ..$ lineheight : NULL
## ..$ margin : NULL
## ..$ debug : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.switch.pad.grid : 'simpleUnit' num 3points
## ..- attr(*, "unit")= int 8
## $ strip.switch.pad.wrap : 'simpleUnit' num 3points
## ..- attr(*, "unit")= int 8
## $ strip.text.y.left :List of 11
## ..$ family : NULL
## ..$ face : NULL
## ..$ colour : NULL
## ..$ size : NULL
## ..$ hjust : NULL
## ..$ vjust : NULL
## ..$ angle : num 90
## ..$ lineheight : NULL
## ..$ margin : NULL
## ..$ debug : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi TRUE
## - attr(*, "validate")= logi TRUE

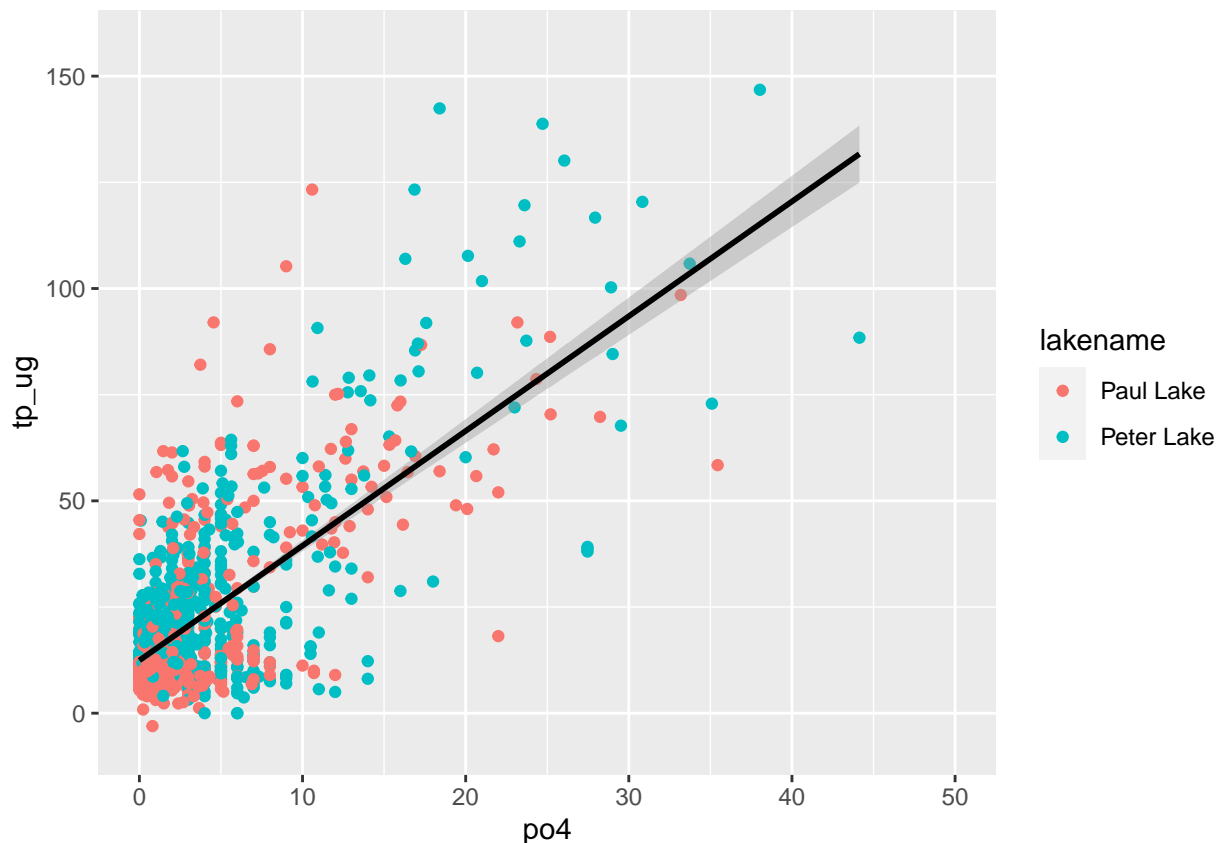
```

```
print(plot4)
```

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

```
## Warning: Removed 21947 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 21947 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.

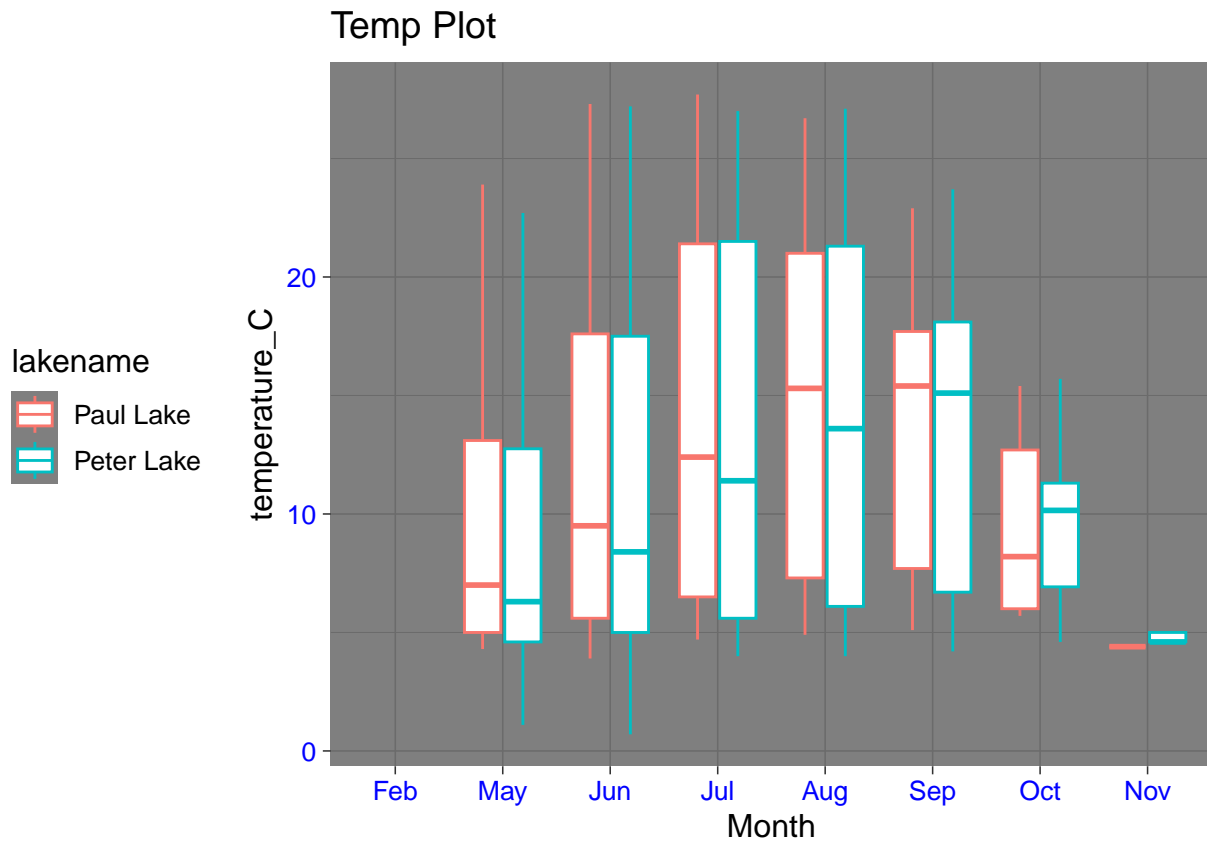
Tip: R has a built in variable called `month.abb` that returns a list of months; see <https://r-lang.com/month-abb-in-r-with-example>

```
#5

#Changing class of month from an integer to a factor prior to plot creation
monthnames <- factor(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$month,
  levels = 1:12,
  labels = month.abb)

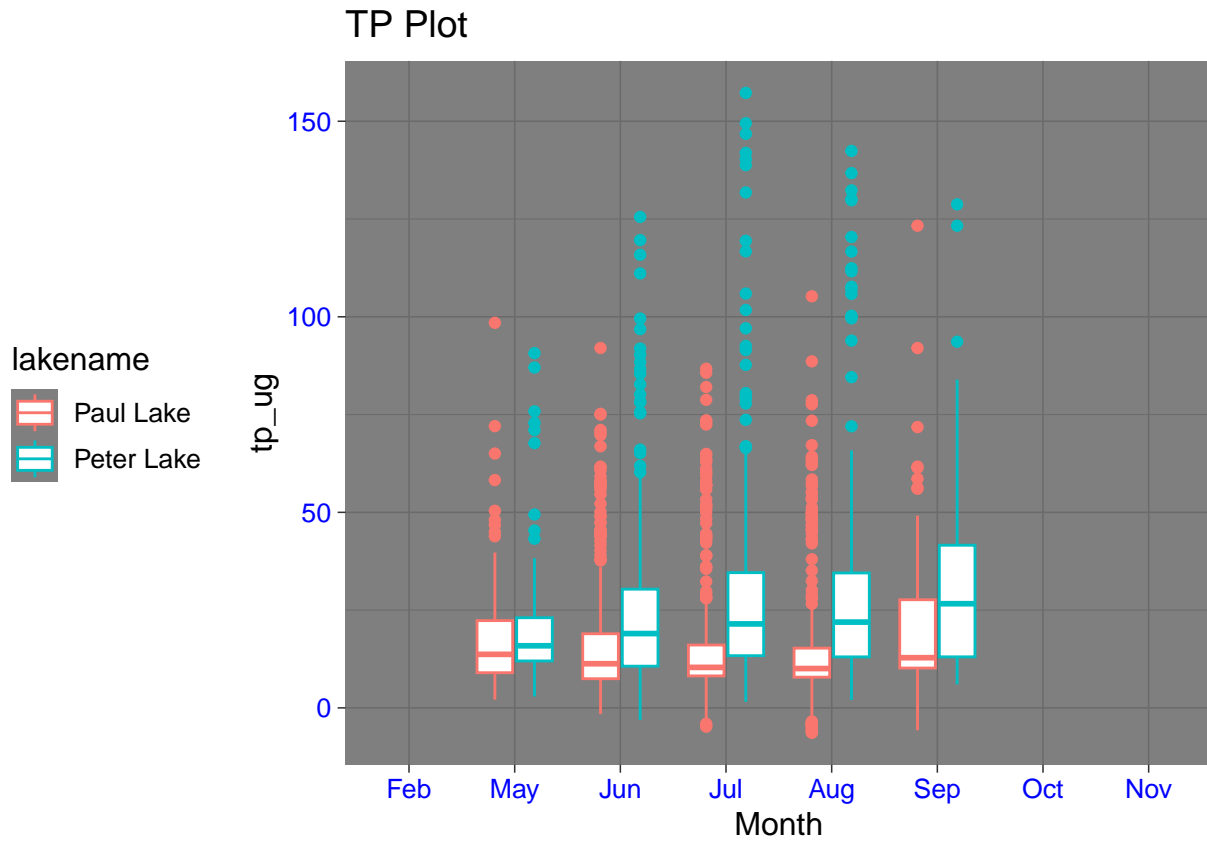
#boxplot for temperature after change month from integer to factor
plot5a <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
    aes(x=monthnames,
        y=temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  labs(
    title="Temp Plot",
    x="Month") +
  samtheme
print(plot5a)
```

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



```
#boxplot for TP
plot5b <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
    aes(x=monthnames,
        y=tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(
    title="TP Plot",
    x="Month") +
  samtheme
print(plot5b)
```

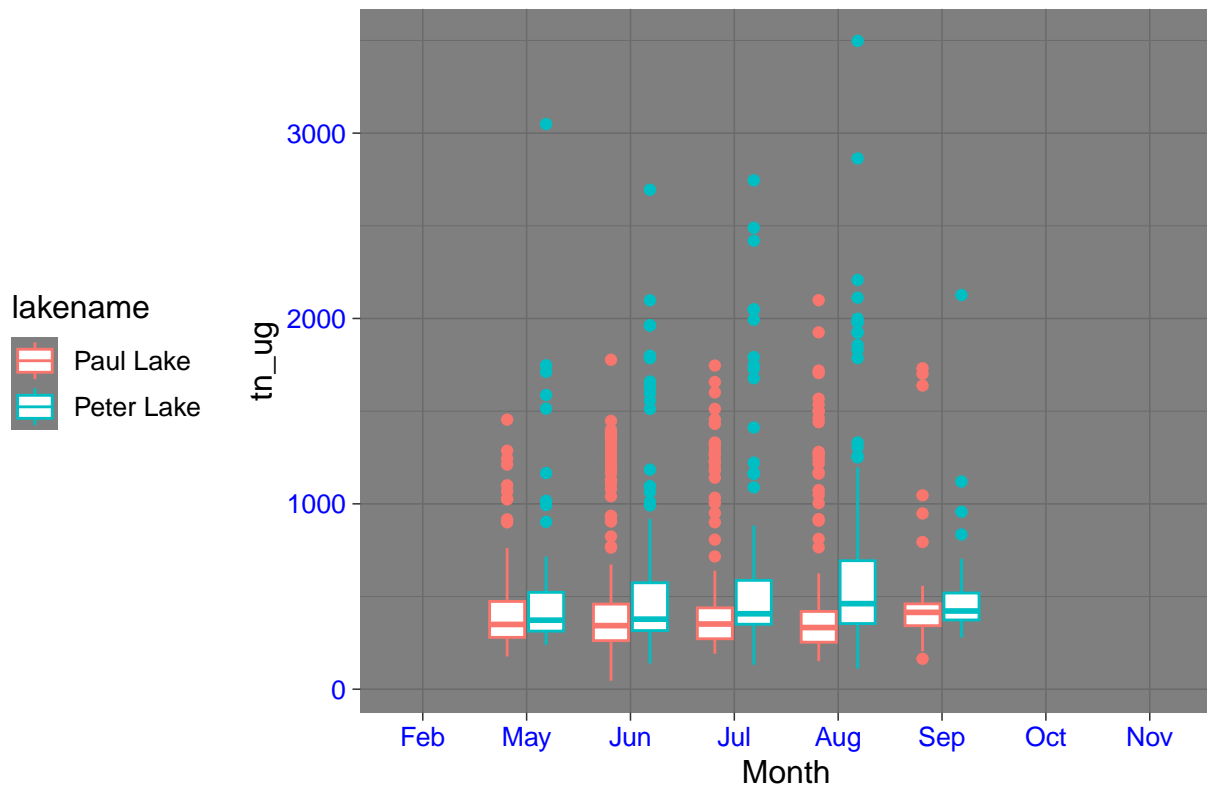
```
## Warning: Removed 20729 rows containing non-finite values ('stat_boxplot()').
```



```
#boxplot for TN
plot5c <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
    aes(x=monthnames,
        y=tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(
    title="TN Plot",
    x="Month") +
  samtheme
print(plot5c)
```

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

TN Plot



*#1st try for cowplot combining temperature, TP, and TN; creates large side-by-side plots w/ 3 individual*

```
library(cowplot)
```

```
cowplot5 <-
```

```
plot_grid(plot5a, plot5b, plot5c, nrow = 1, align = "h", axis = "tb")
```

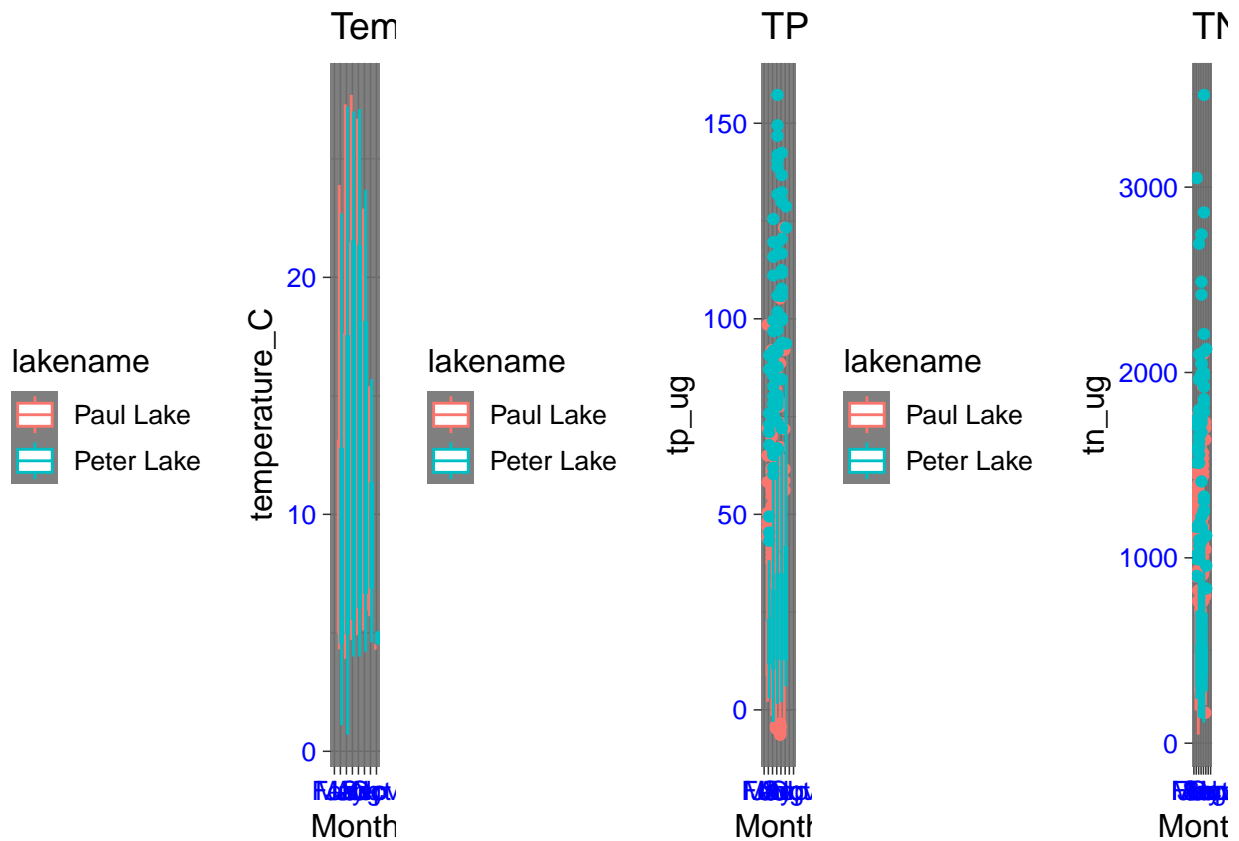
```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```

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

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

```
print(cowplot5)
```





```
#extract one legend
legend <- get_legend(plot5a)
```

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

```
#2nd try for cowplot; shows one legend
cowplot5 <-
  plot_grid(
    plot5a + theme(legend.position="none"),
    plot5b + theme(legend.position="none"),
    plot5c + theme(legend.position="none"),
    legend,
    align = 'vh',
    hjust = -1,
    nrow = 1
  )
```

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

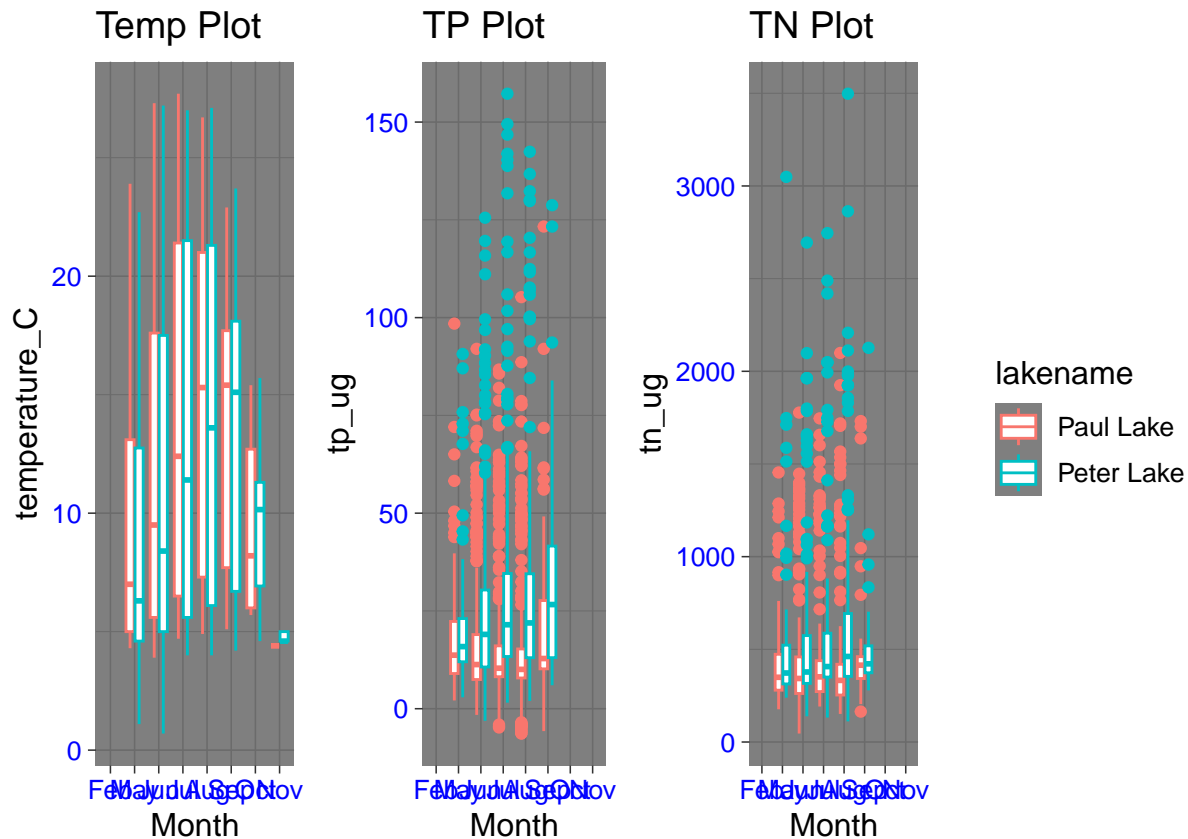
```
## Warning: Removed 20729 rows containing non-finite values ('stat_boxplot()').
```

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

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

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

```
print(cowplot5)
```



```
#Final cowplot with aligned axes
```

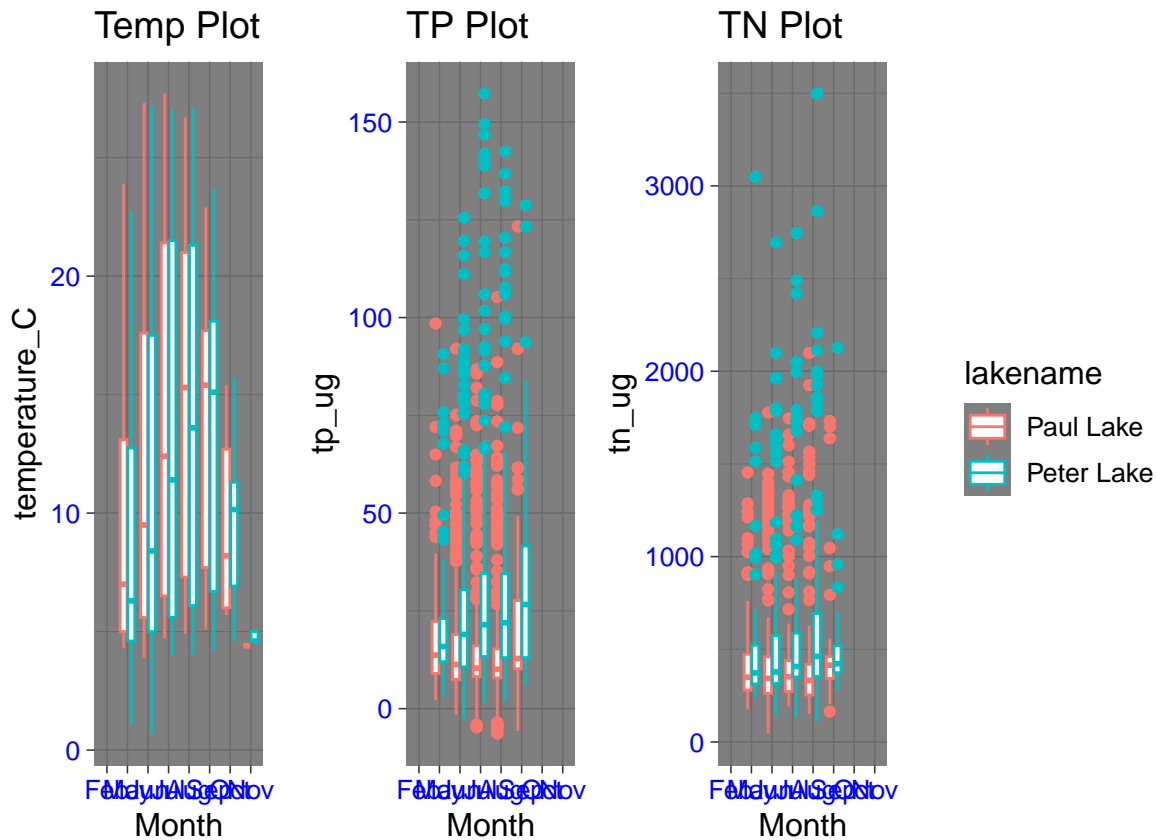
```
cowplot5 <-
  plot_grid(
    plot5a + theme(legend.position="none"),
    plot5b + theme(legend.position="none"),
    plot5c + theme(legend.position="none"),
    legend,
    align = 'vh',
    axis = "btlr",
    hjust = -1,
    nrow = 1
  )
```

```
## 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()').
```

```
print(cowplot5)
```



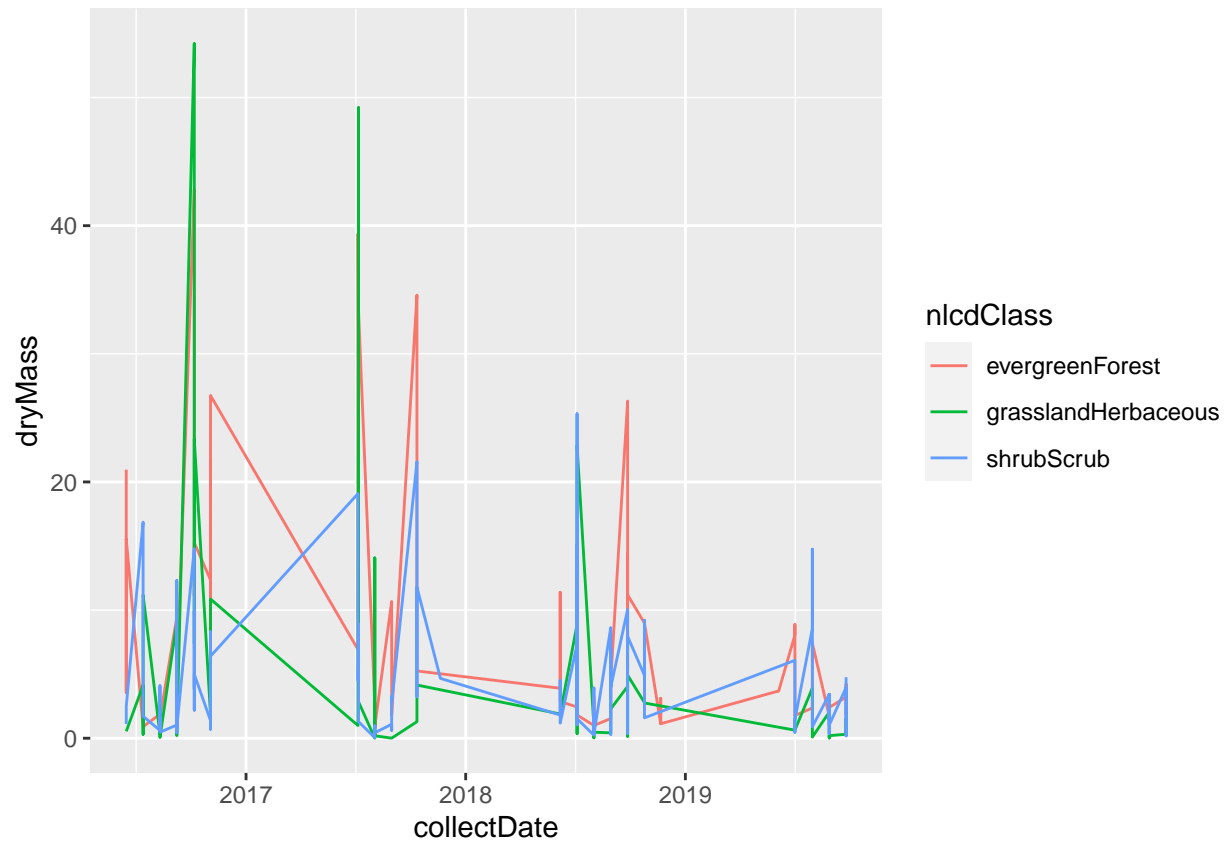
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: The 3 variables seem to be aligned over seasons and between lakes, each reaching a peak during the summer months.

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.

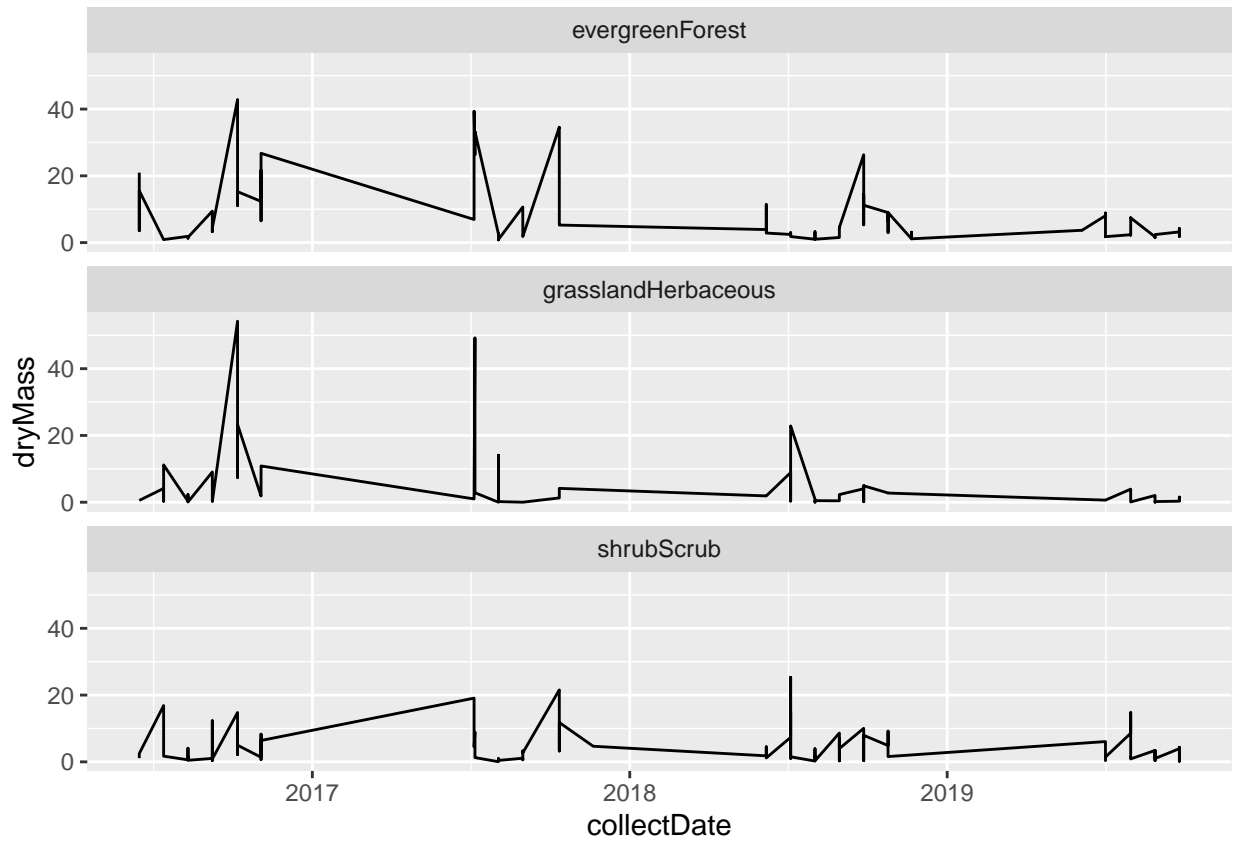
```
#6

#Plot needle litter dry mass b date with NLCD class color aesthetic
plot6 <-
  ggplot(subset(NEON_NIWO_Litter_mass_trap_Processed, functionalGroup == "Needles"),
    aes(x = collectDate,
        y = dryMass,
        color = nlcdClass)) +
  geom_line()
print(plot6)
```



```
#7

#Same plot as 6, but with NLCD classes as facets rather than colors
plot7 <-
  ggplot(subset(NEON_NIWO_Litter_mass_trap_Processed, functionalGroup == "Needles"),
    aes(x = collectDate,
        y = dryMass)) +
  geom_line() +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(plot7)
```



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

Answer: I think plot 7 is more effective. The lines on the plot get jumbled up and hard to see like spaghetti when all displayed on the same plot, but when separated into 3 facets it becomes easier to see the changes in each nlcdClass over time, and to compare the classes across the 3 facets.