Assignment 5: Data Visualization

Gaby Antonova

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Monday, February 14 at 7:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and 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).

```
getwd()
```

```
## [1] "C:/Users/gabri/OneDrive/Desktop/MPP Coursework/Spring 2022/ENVIRO 872/Environmental_Data_Analyti
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot)
library(RColorBrewer)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed <- read_csv("../Data/Processed/NTL-LTER_Lake_Chem
```

```
## Rows: 23008 Columns: 15
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): lakename
## dbl  (13): year4, daynum, month, depth, temperature_C, dissolvedOxygen, irra...
## date  (1): sampledate

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
NEON_NIWO_Litter_mass_trap_Processed <- read_csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processe
```

```
## Rows: 1692 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): plotID, trapID, functionalGroup, qaDryMass, nlcdClass, plotType, g...
## dbl  (5): dryMass, subplotID, decimalLatitude, decimalLongitude, elevation
## date (1): collectDate

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

```
#1
class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)
```

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

```
#2
class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
```

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

Define your theme

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

```r
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(mytheme)
theme_get()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : 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"
##   ..$ size         : 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 "black"
##   ..$ 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"
##   ..$ size         : NULL
##   ..$ 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 of 6
##   ..$ colour       : chr "black"
##   ..$ size         : 'rel' num 1
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "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
##   ..$ size         : 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()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "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 "top"
##  $ 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 "white"
##   ..$ colour       : logi NA
##   ..$ size         : 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 "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.grid.minor          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "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"
##   ..$ size         : 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 "white"
##   ..$ colour       : chr "black"
##   ..$ size         : 'rel' num 2
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ 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
```

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.

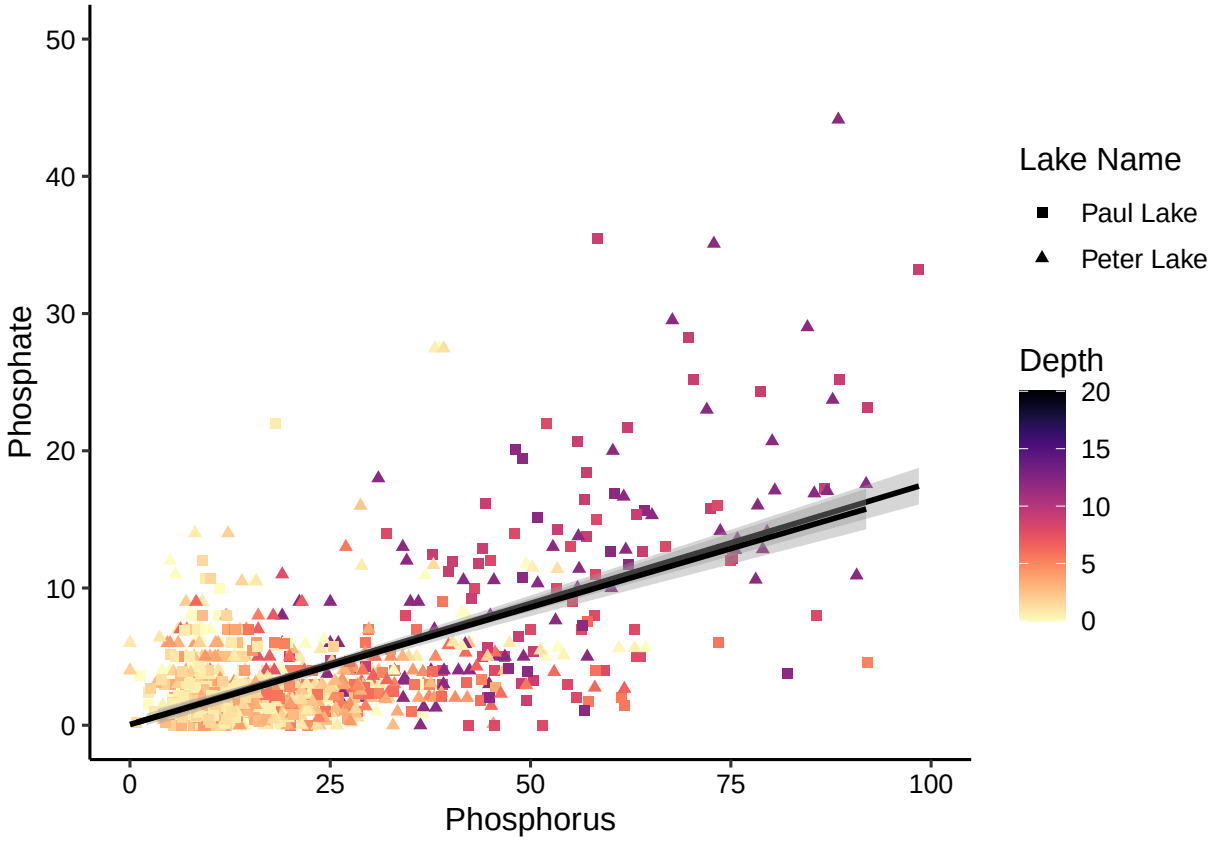
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and `ylim()`).

```r
#4
PhosphorusByPhosphate <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
aes(x= tp_ug, y= po4, shape=lakename, color=depth)) +
  geom_point()+
  xlim(0,100) +
  ylim(0,50)+
  scale_color_viridis(option = "magma", direction = -1)+
  ylab("Phosphate")+
  xlab("Phosphorus")+
  theme(legend.position = "right")+
  labs(color="Depth", shape="Lake Name")+
  geom_smooth(method = lm, color="black")+
  scale_shape_manual(values = c(15, 17))


print(PhosphorusByPhosphate)
```

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

```
## Warning: Removed 21964 rows containing non-finite values (stat_smooth).
```

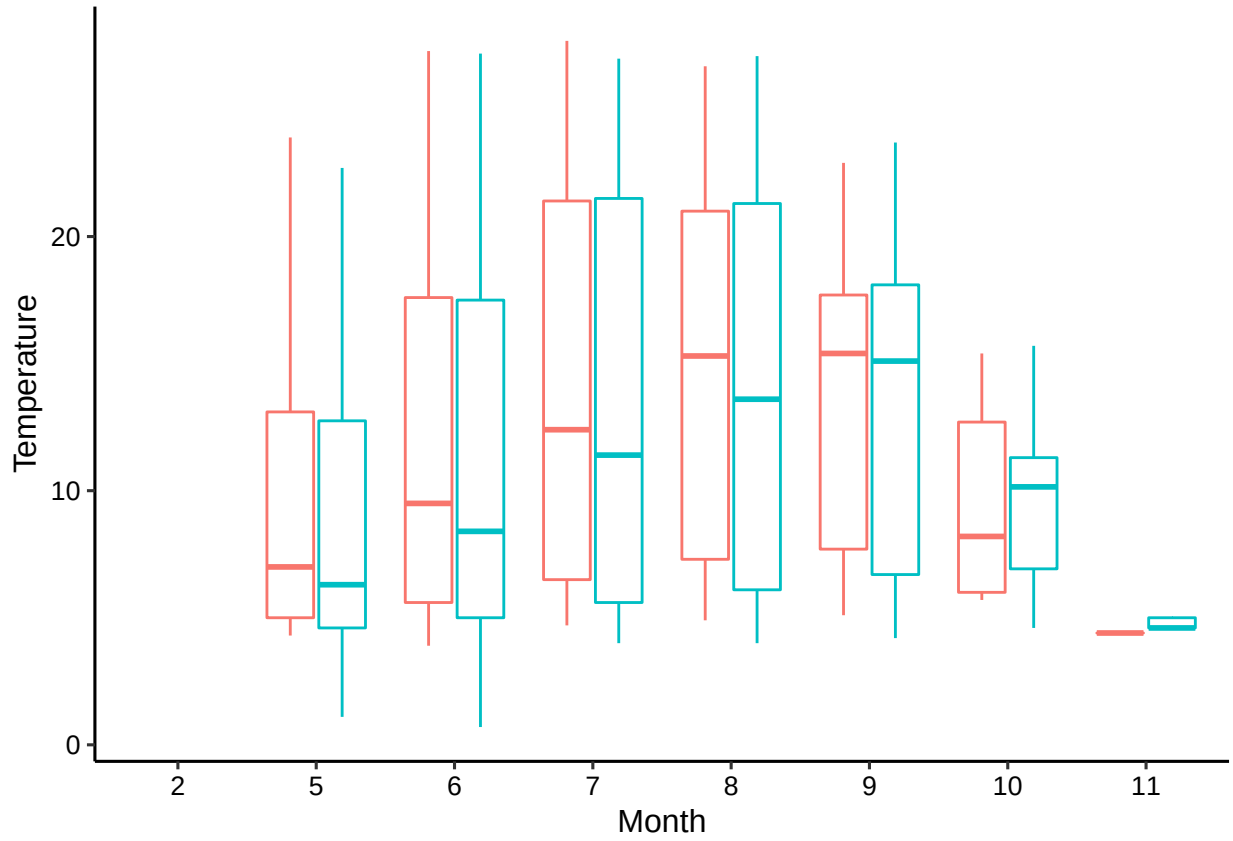
```
## Warning: Removed 21964 rows containing missing values (geom_point).
```



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.

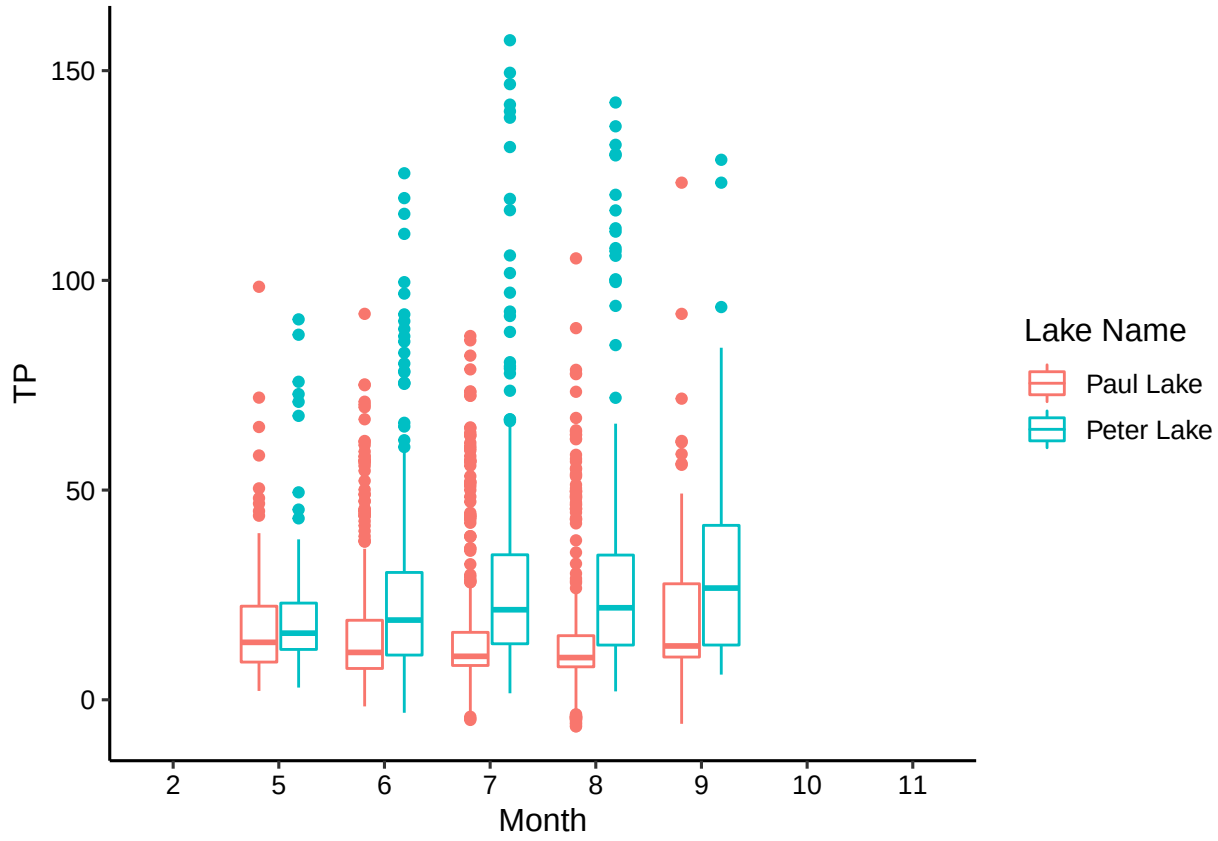
```
#5


temperature_boxplot <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
aes(x= as.factor(month), y=temperature_C, color=lakename)) +
  geom_boxplot()+
  ylab("Temperature")+
  xlab("Month")+
  labs(color="Lake Name")+
  theme(legend.position ="none")
print(temperature_boxplot)
```

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



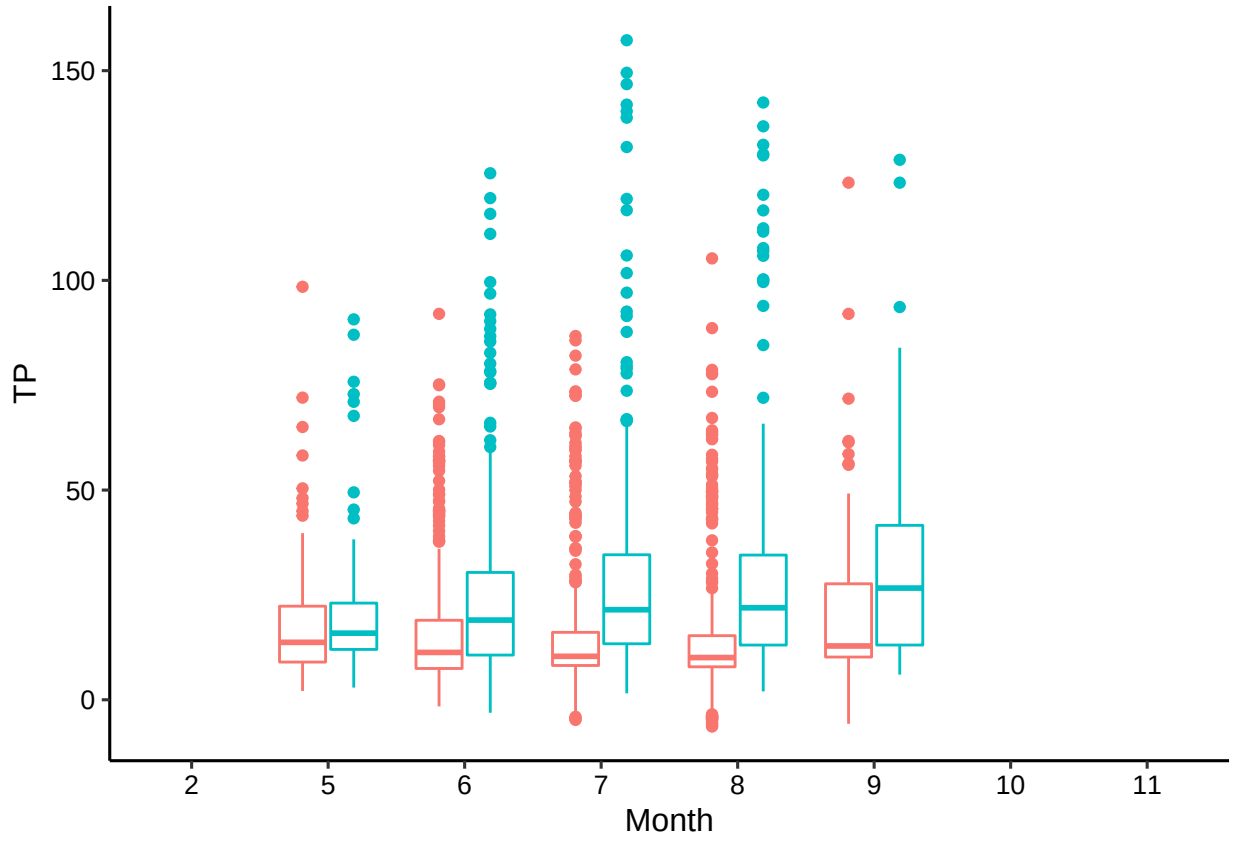

```
TP_boxplot_for_Legend <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
aes(x= as.factor(month), y=tp_ug, color=lakename)) +
  geom_boxplot()+
  xlab("Month")+
  ylab("TP")+
  labs(color="Lake Name")+
  theme(legend.position="right")
print(TP_boxplot_for_Legend)
```

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



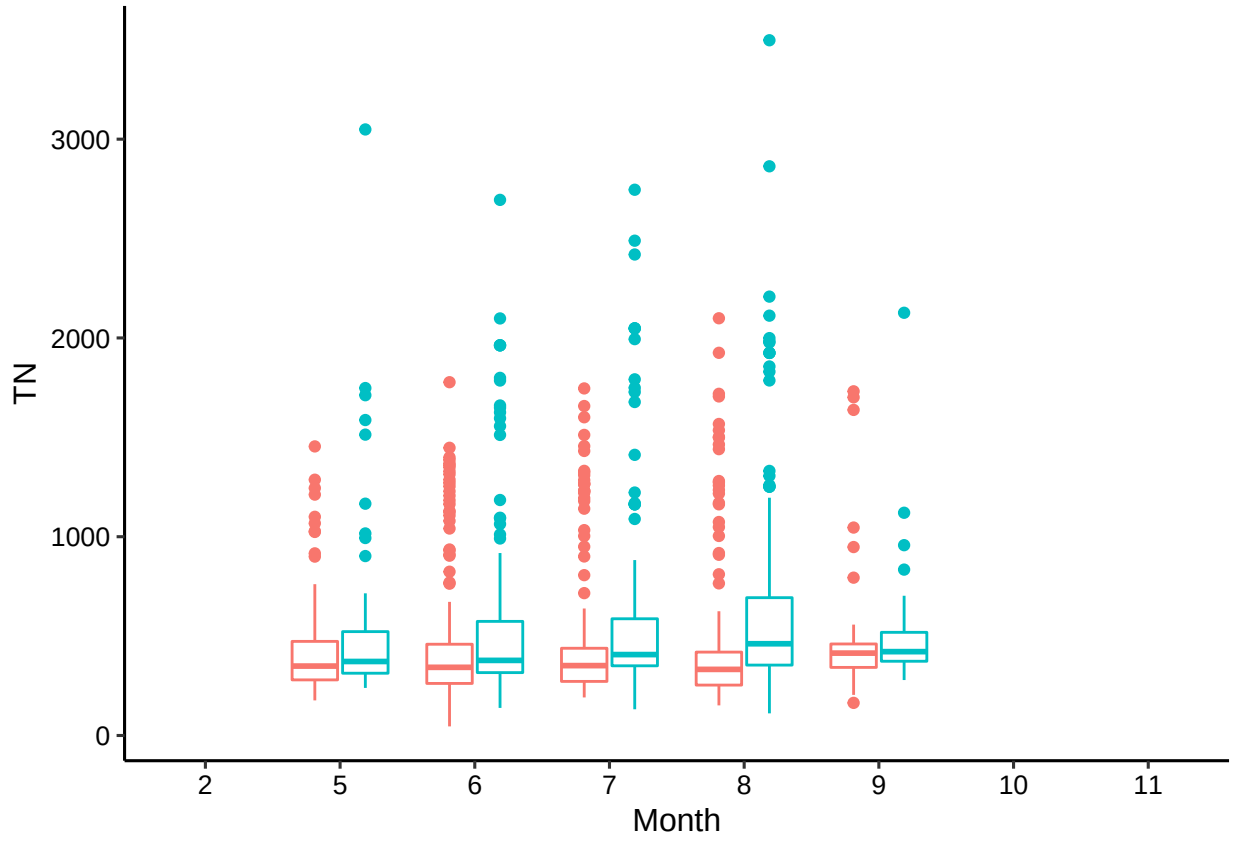

```
TP_boxplot <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
aes(x= as.factor(month), y=tp_ug, color=lakename)) +
  geom_boxplot()+
  xlab("Month")+
  ylab("TP")+
  labs(color="Lake Name")+
  theme(legend.position="none")
print(TP_boxplot)
```

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




```
TN_boxplot <-
  ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
  aes(x= as.factor(month), y=tn_ug, color=lakename)) +
  geom_boxplot()+
  ylab("TN")+
  xlab("Month")+
  labs(color="Lake Name")+
  theme(legend.position="none")
print(TN_boxplot)
```

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



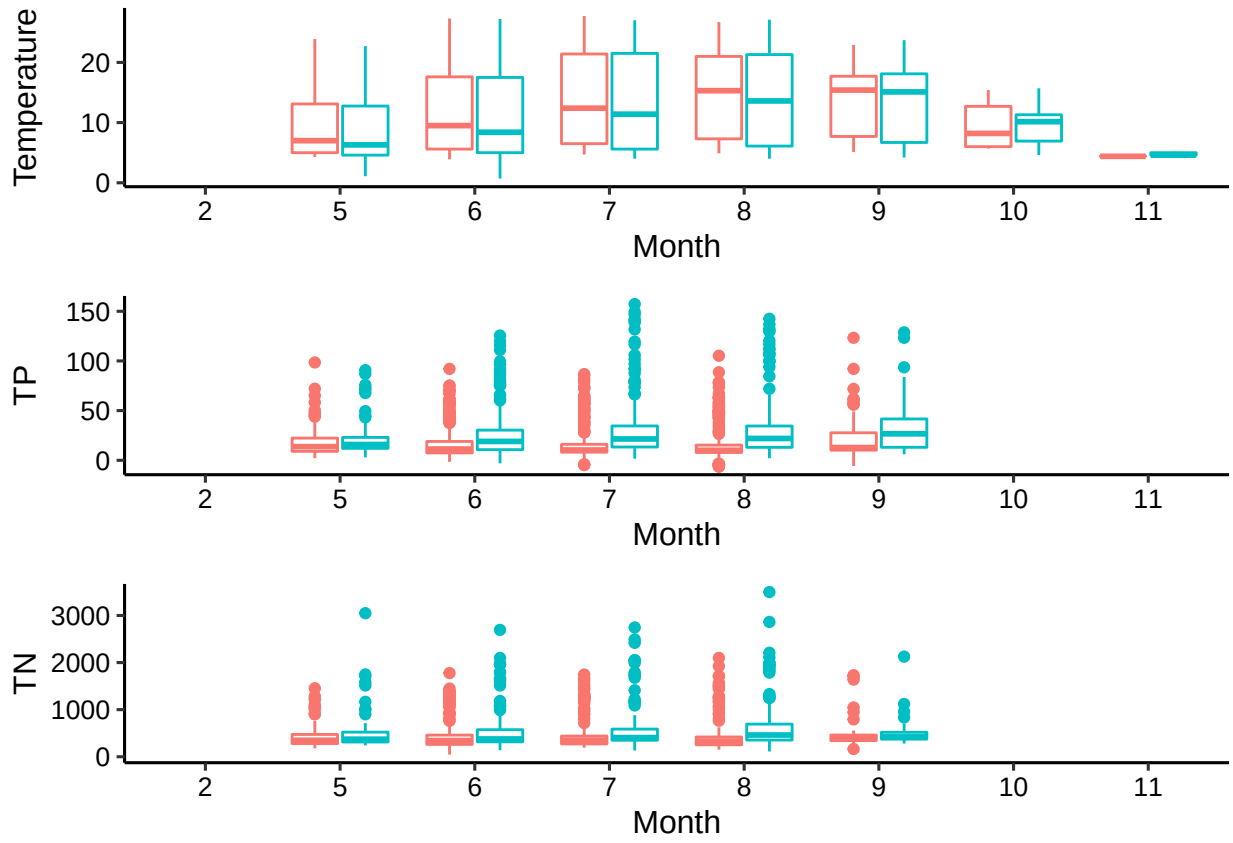

```
combined_plots <-
  plot_grid(temperature_boxplot, TP_boxplot, TN_boxplot, nrow = 3, align ="v")
```

```
## 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(combined_plots)
```



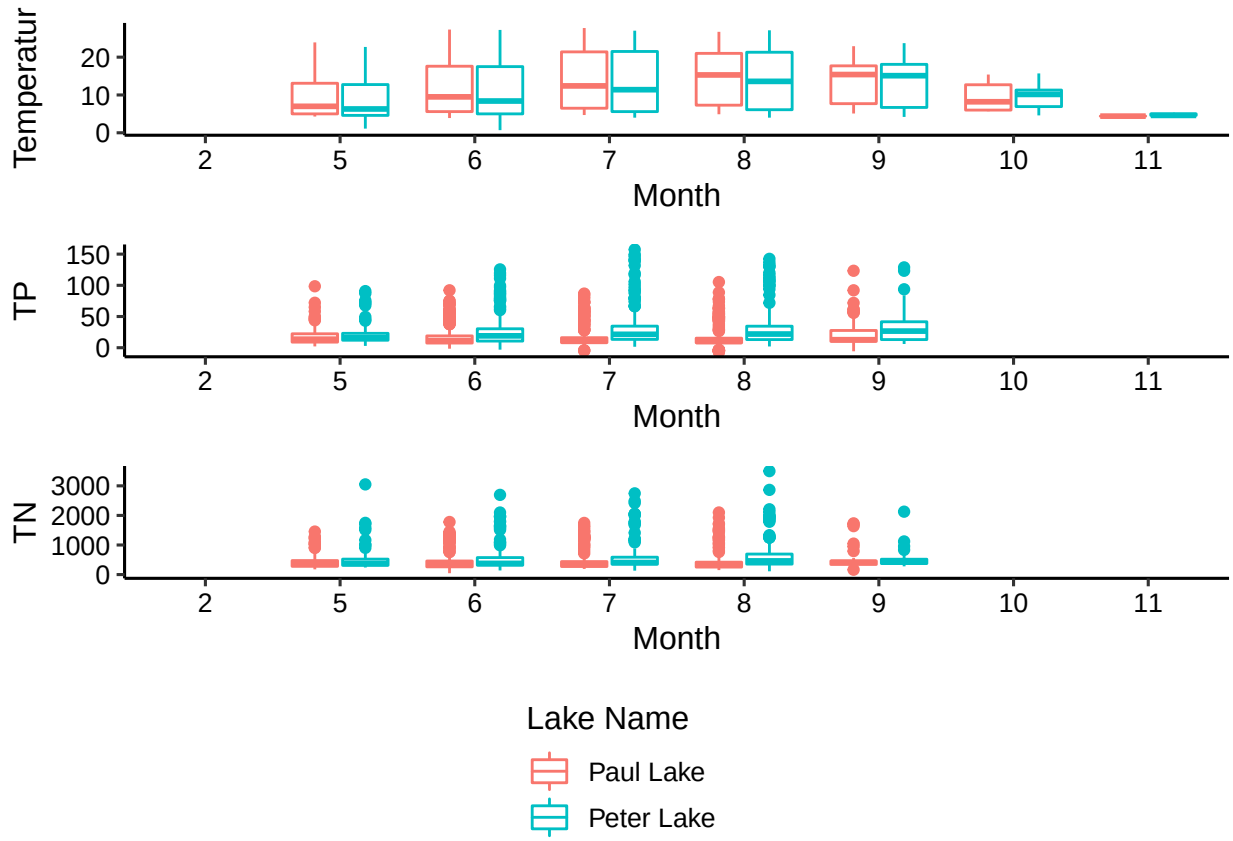

```
legend_plots <-
  get_legend(TP_boxplot_for_Legend)
```

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

```
print(legend_plots)
```

```
## TableGrob (5 x 5) "guide-box": 2 grobs
##                                     z     cells                 name
## 99_3e8bde505edc50a9718d7cf1ebfefe6d 1 (3-3,3-3)               guides
##                                     0 (2-4,2-4) legend.box.background
##                                                               grob
## 99_3e8bde505edc50a9718d7cf1ebfefe6d gtable[layout]
##                                     zeroGrob[NULL]
```

```
combined_plots_legend <-
  plot_grid(combined_plots, legend_plots, nrow = 2,  rel_heights = c(5, 1.5))
print(combined_plots_legend)
```



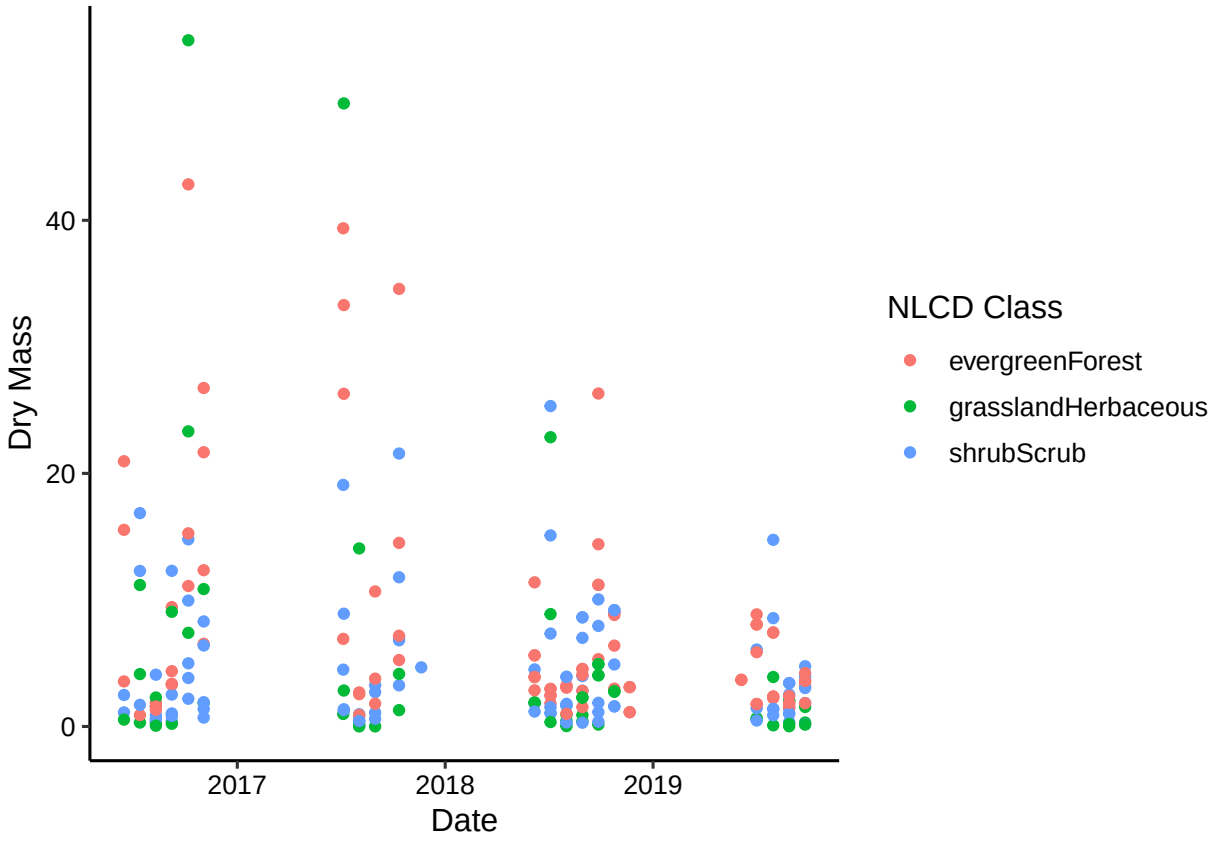
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: There appear to be more nutrients in the water when temperature is higher in the summer season. There are limited measurements taken in the winter. Peter lake seems to contain more TP and TN on average than Paul Lake, although it seems to be slightly cooler than Paul Lake. None of these results appear to be statistically significant.

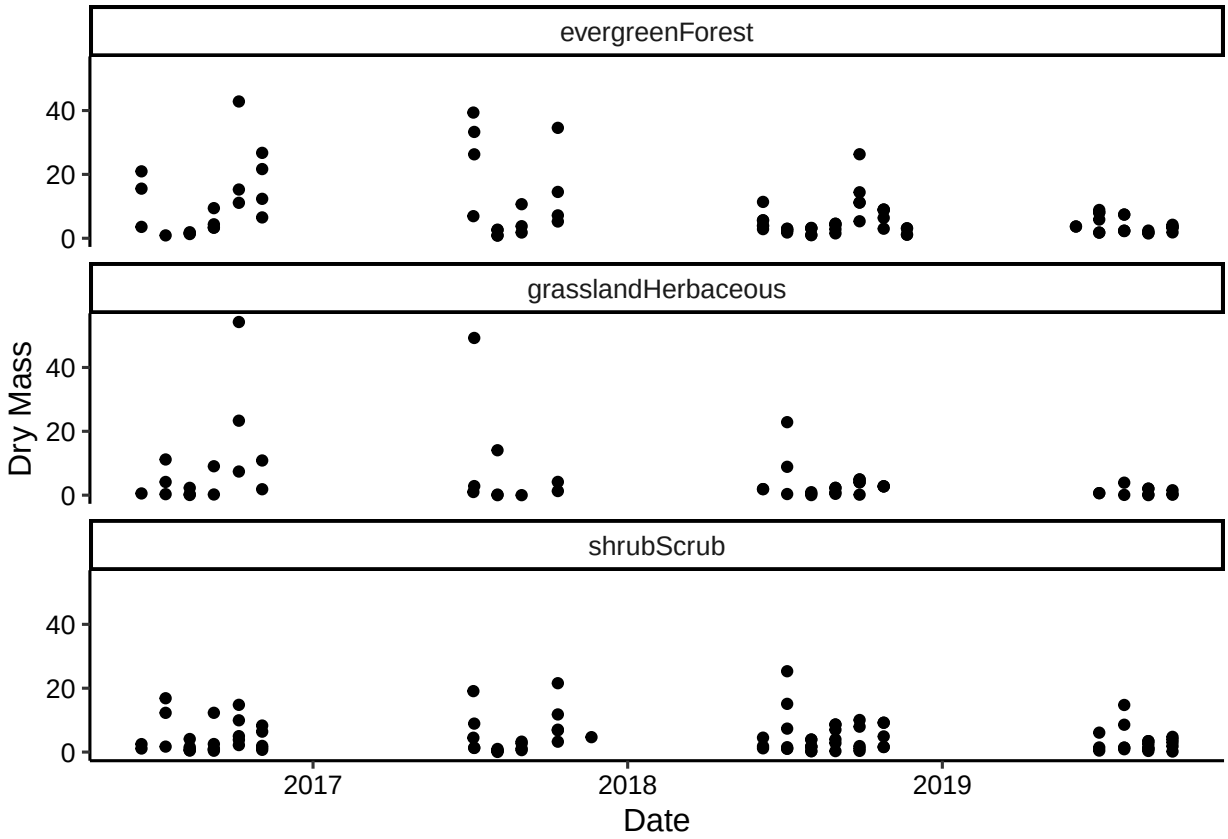
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

Niwot_Ridge_needles <-
ggplot(subset(NEON_NIWO_Litter_mass_trap_Processed,
functionalGroup == "Needles"), aes(x=collectDate, y=dryMass, color=nlcdClass))+
  geom_point() +
  theme(legend.position = "right")+
  ylab("Dry Mass")+
  xlab("Date")+
  labs(color = "NLCD Class")
print(Niwot_Ridge_needles)
```




```
#7
Niwot_Ridge_needles_facets <-
ggplot(subset(NEON_NIWO_Litter_mass_trap_Processed,
functionalGroup == "Needles"), aes(x=collectDate, y=dryMass))+
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3)+
  theme(legend.position = "right")+
  ylab("Dry Mass")+
  xlab("Date")
print(Niwot_Ridge_needles_facets)
```



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

Answer: I think plot 6 is more effective because it is much easier to compare the different NLCD classes since you can see them right next to each other. In plot 7, the differences in Dry Mass by NLCD class are much less obvious.