

Assignment 5: Data Visualization

Kamil Burak Karayel

Spring 2025

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. Read in 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 in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the NEON_NIWO_Litter_mass_trap_Processed.csv version, again from the Processed_KEY folder).
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 core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
## here() starts at /home/guest/EDA_Spring2025_kbk
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
here()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
PeterPaul.chem.nutrients <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)
```

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

```
#2
class(PeterPaul.chem.nutrients$sampleddate)
```

```
## [1] "factor"
```

```
#Class of sampleddate is factor
PeterPaul.chem.nutrients$sampleddate <- ymd(PeterPaul.chem.nutrients$sampleddate)
#Change to date

class(Niwot.Ridge.litter$collectDate)
```

```
## [1] "factor"
```

```
#Class of collectDate is factor
Niwot.Ridge.litter$collectDate <- ymd(Niwot.Ridge.litter$collectDate)
#Change to 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

```
library(ggthemes) #knit couldn't see theme_base, added for that error
```

```
##
```

```
## Attaching package: 'ggthemes'
```

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

```
##
```

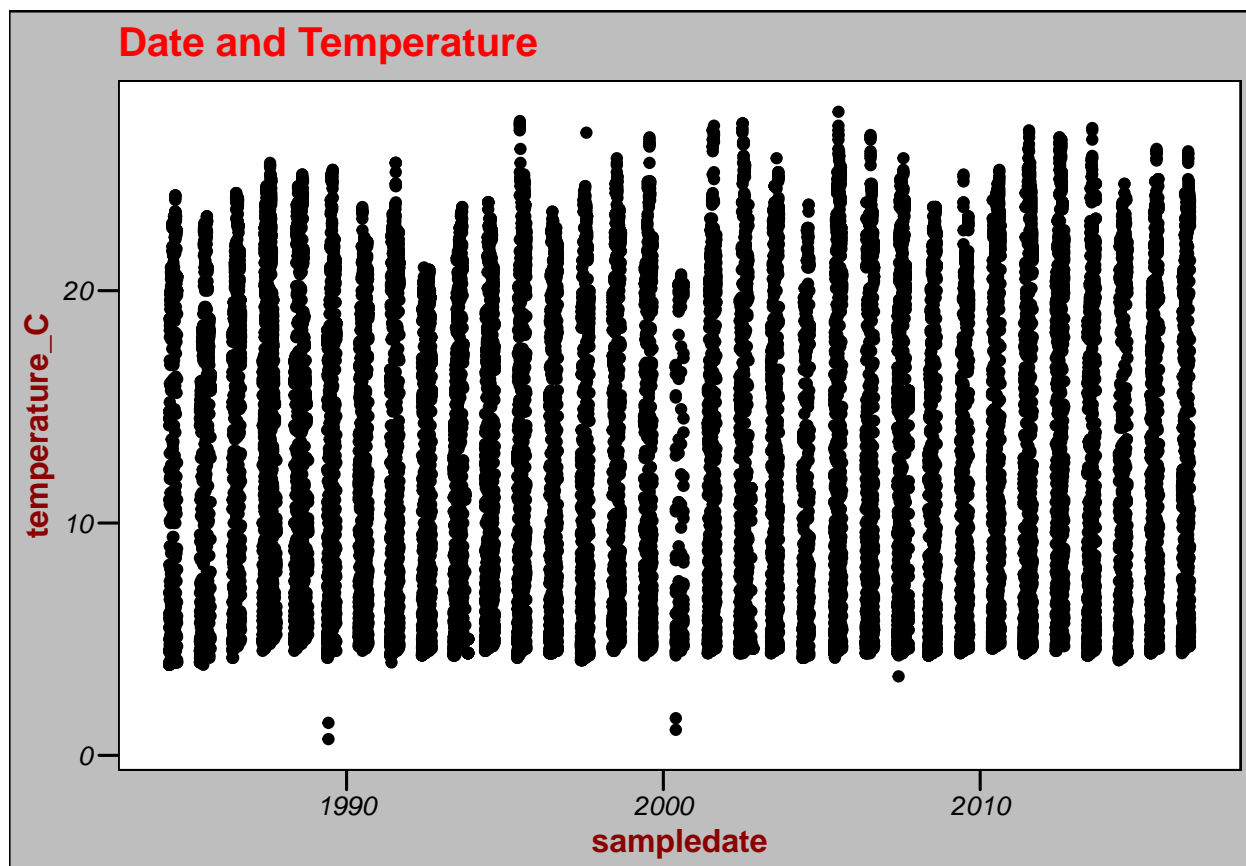
```
## theme_map
```

```
theme_kbk <- theme_base() +
  theme(
    plot.background = element_rect(colour = 'black', fill = 'grey'),
    #background is grey and frame is black
    plot.title = element_text(size = 15, colour = 'red'),
    #title of the plot is red and size of 15
    axis.title = element_text(size = 12, face = "bold", colour = "darkred"),
    #axis labels are dark red, bold and size of 12
    axis.text = element_text(size = 10, face = "italic"),
    #indicators of the axis are italic and size of 10
    legend.position = 'bottom'
    #legend will be at the bottom of the plot
  )
```

```
#test plot for the new theme_kbk
```

```
ggplot(PeterPaul.chem.nutrients, aes(x = sampleddate, y = temperature_C)) +
  geom_point() + labs(title = "Date and Temperature") + theme_kbk
```

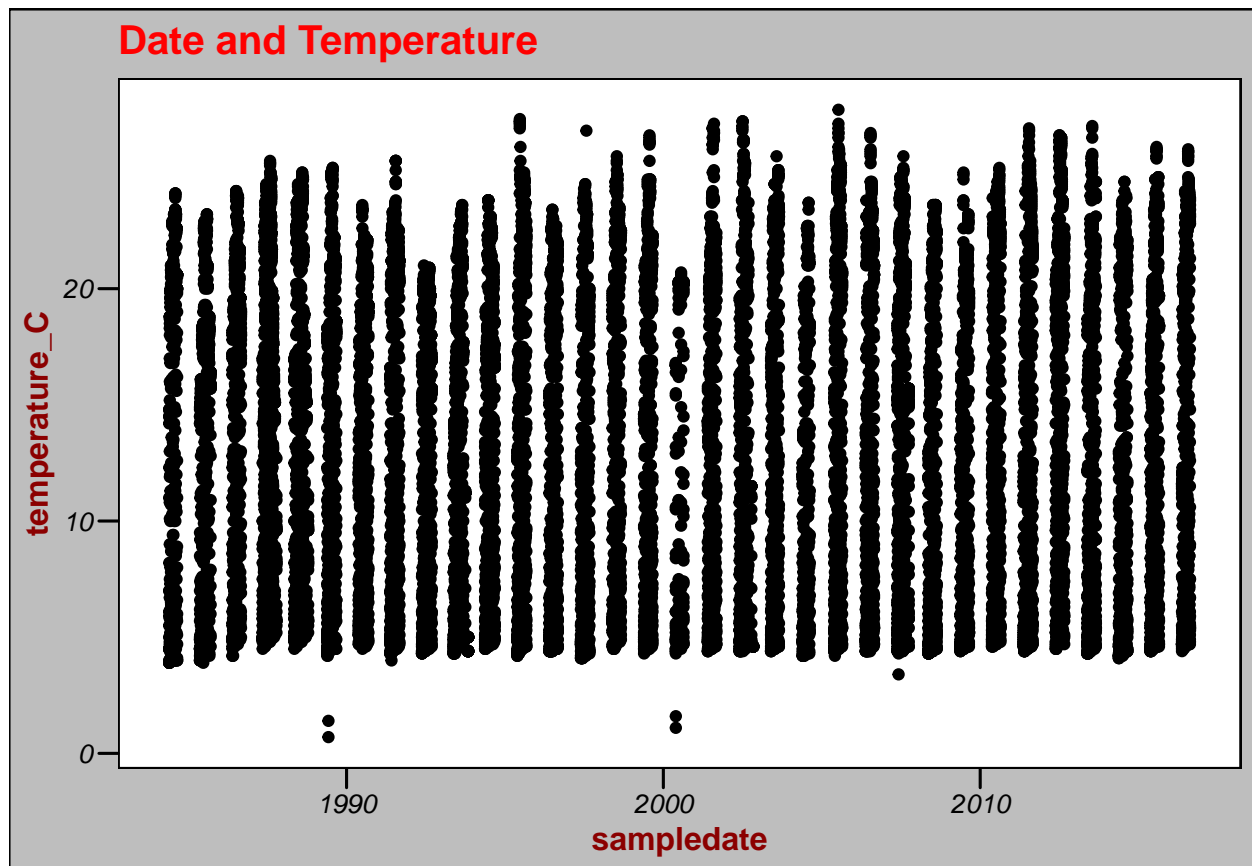
```
## Warning: Removed 3566 rows containing missing values or values outside the scale range
## ('geom_point()').
```



```
#Set new theme_kbk as the default theme
theme_set(theme_kbk)
```

```
#check for the default theme whether it is theme_kbk
ggplot(PeterPaul.chem.nutrients, aes(x = sampledate, y = temperature_C)) +
  geom_point() + labs(title = "Date and Temperature")
```

```
## Warning: Removed 3566 rows containing missing values or values outside the scale range
## ('geom_point()').
```



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.

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

```
#4
library(dplyr)

#Want to see the min and max points to decide extreme points to drop out
summary(PeterPaul.chem.nutrients$tp_ug)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
## -6.349   9.194  14.401  22.159  27.746  157.250   20729
```

```
#Want to see the min and max points to decide extreme points to drop out
summary(PeterPaul.chem.nutrients$po4)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
## -0.233   1.000   2.324   5.919   5.000  373.836   21822
```

```

ggplot(PeterPaul.chem.nutrients,
       aes(x = po4, y = tp_ug, color = lakename)) + #Add color as separate aesthetics for lakes
geom_point() +
labs(title = str_wrap("Phosporus and Phosphate Levels of Peter and Paul Lakes", width = 30),
     #Very long title, I had to cut it at 30 characters
     x = "Phosphate",      #title for x axis
     y = "Phosphorus",     #title for y axis
     color = 'Lake') +     #Write "Lake" instead of "lakename" in legend
geom_smooth(method = "lm", color = "black") +      #Add line for lakes to see the relation between
                                                    #Phosphorus and Phosphate
xlim(-1,45) +
                                                    #-1 and 45 are enough to see the minimum
                                                    #and maximum values of Phosphate,
                                                    #no need to hide extreme values for Phosphorus,
                                                    #so no need for ylim function

facet_wrap(vars(lakename), ncol = 2) +          #Divide into two graphs for Peter and Paul lakes.
theme(legend.position = "right")                #Legend at the bottom is seen bad for two columns,

```

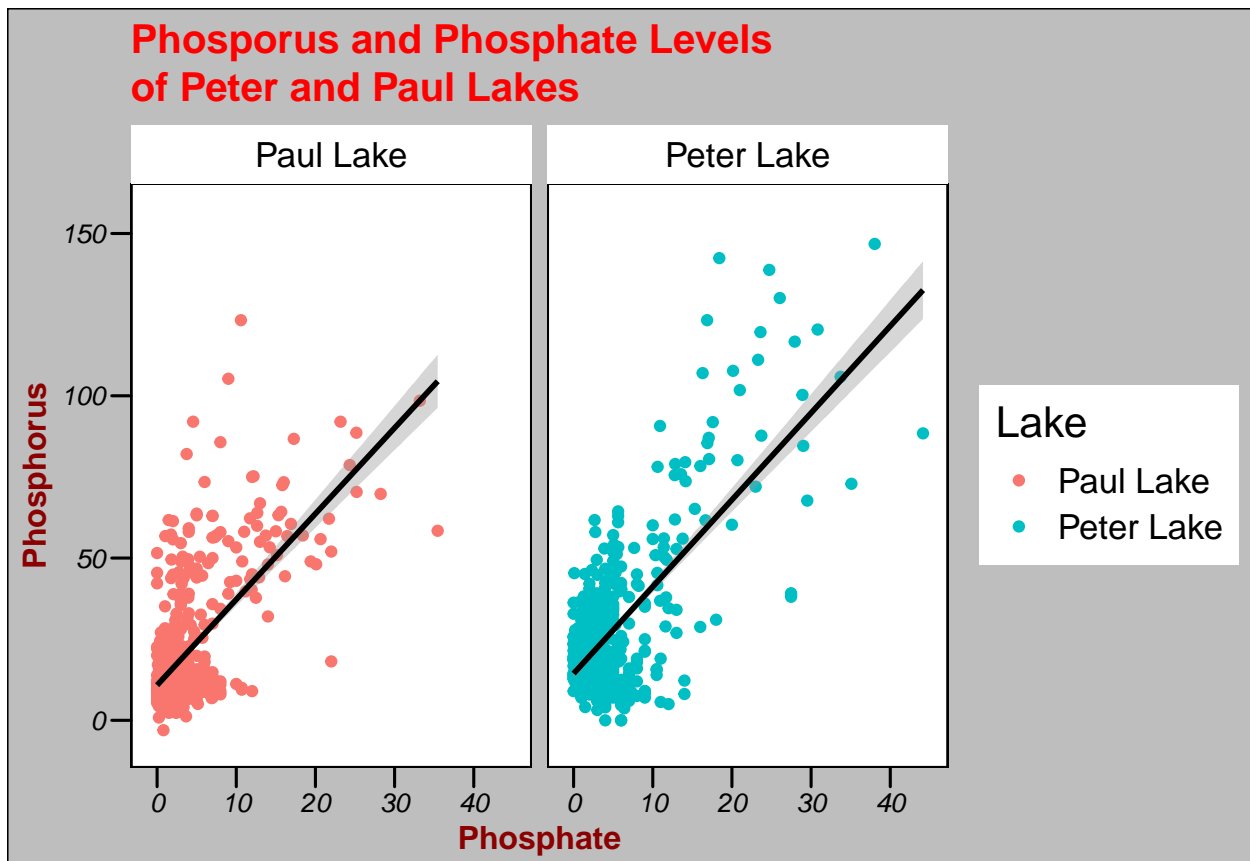
```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 21947 rows containing non-finite outside the scale range
```

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

```
## Warning: Removed 21947 rows containing missing values or values outside the scale range
```

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



#I moved the legend to right.

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.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
#Create new column called Jan2Dec showing month names by using data in month column.
#This column will be used in X axis.
PeterPaul.chem.nutrients <- PeterPaul.chem.nutrients %>%
  mutate(Jan2Dec = factor(month.abb[month], levels = month.abb))

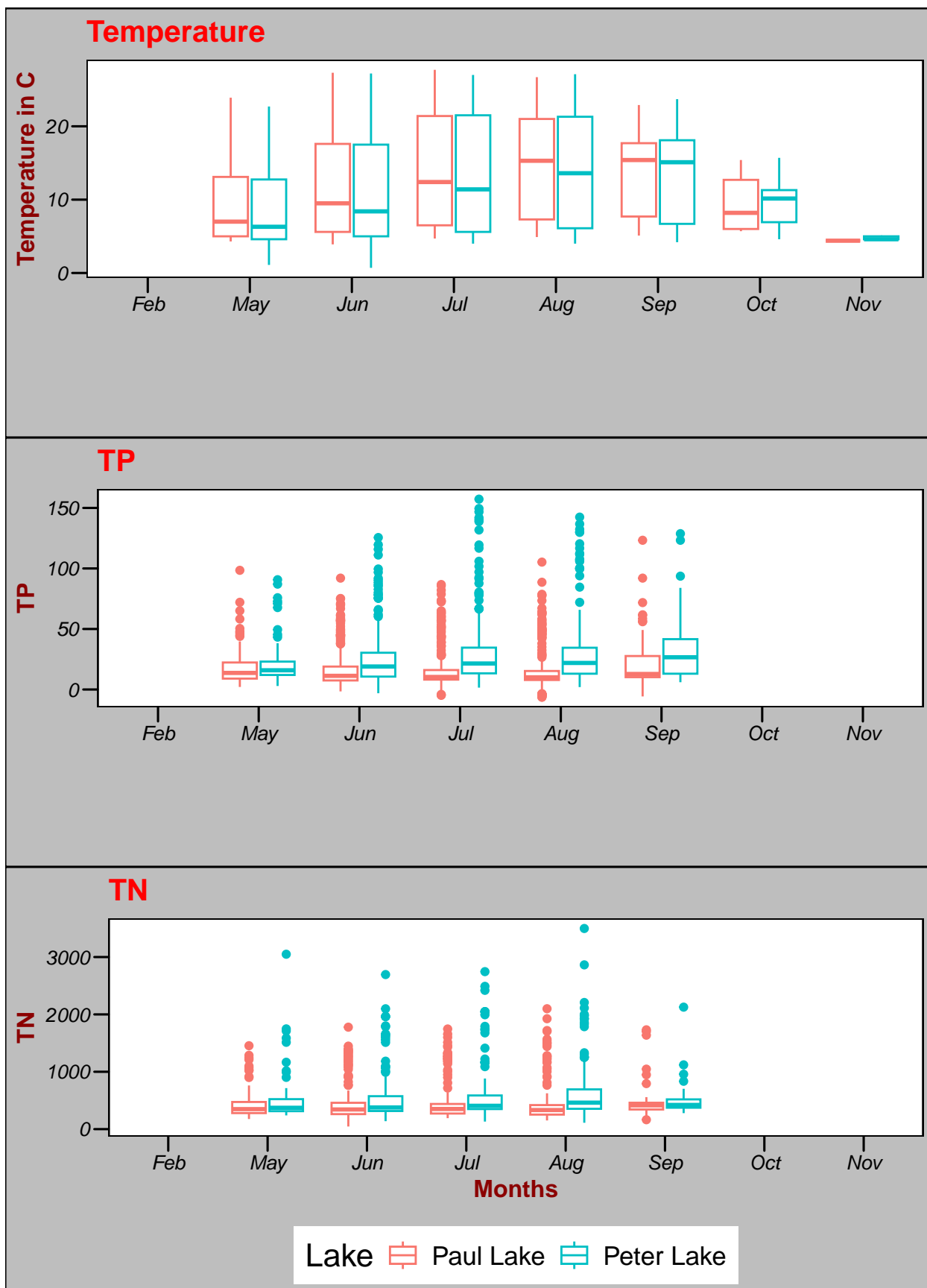
a_Temp <- ggplot(PeterPaul.chem.nutrients, #Plot temperature with month for two lakes in different colors
  aes(x = Jan2Dec, y = temperature_C, color = lakename)) +
  geom_boxplot(na.rm = TRUE) + #drop NA values from the plot not to get error
  labs(title = "Temperature",
    x = "",
    y = "Temperature in C",
    color = 'Lake') +
  theme(legend.position = "none")

b_TP <- ggplot(PeterPaul.chem.nutrients, #Plot TP with month for two lakes in different colors
  aes(x = Jan2Dec, y = tp_ug, color = lakename)) +
  geom_boxplot(na.rm = TRUE) + #drop NA values from the plot not to get error
  labs(title = "TP",
    x = "",
    y = "TP",
    color = 'Lake')+
  theme(legend.position = "none")

c_TN <- ggplot(PeterPaul.chem.nutrients, #Plot TN with month for two lakes in different colors
  aes(x = Jan2Dec, y = tn_ug, color = lakename)) +
  geom_boxplot(na.rm = TRUE) + #drop NA values from the plot not to get error
  labs(title = "TN",
    x = "Months",
    y = "TN",
    color = 'Lake')+
  theme(legend.position = "bottom")

#Merge three plot for Temperature, TP and TN.
Q5_plot <- plot_grid(
  a_Temp,
  b_TP,
  c_TN,
  nrow = 3, align = 'h'
)

Q5_plot
```



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

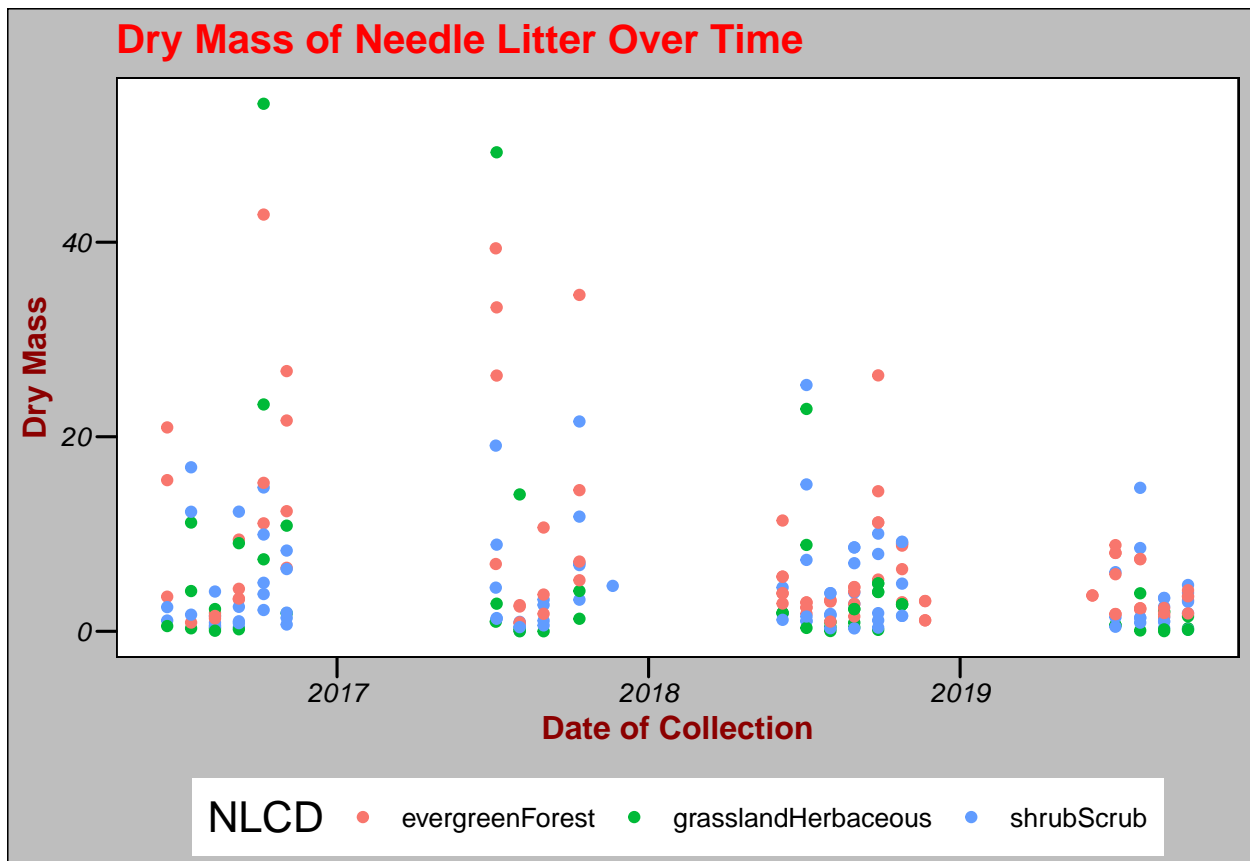
Answer: The observations for TP and TN are measured between May and September. Temperature is observed from May to November. While temperature levels are at the similar level for both lakes, TP and TN levels for Peter Lake is slightly higher than Paul Lake.

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 <- Niwot.Ridge.litter %>%
  filter(functionalGroup == "Needles") #Create a new dataset with only Needles functional group.

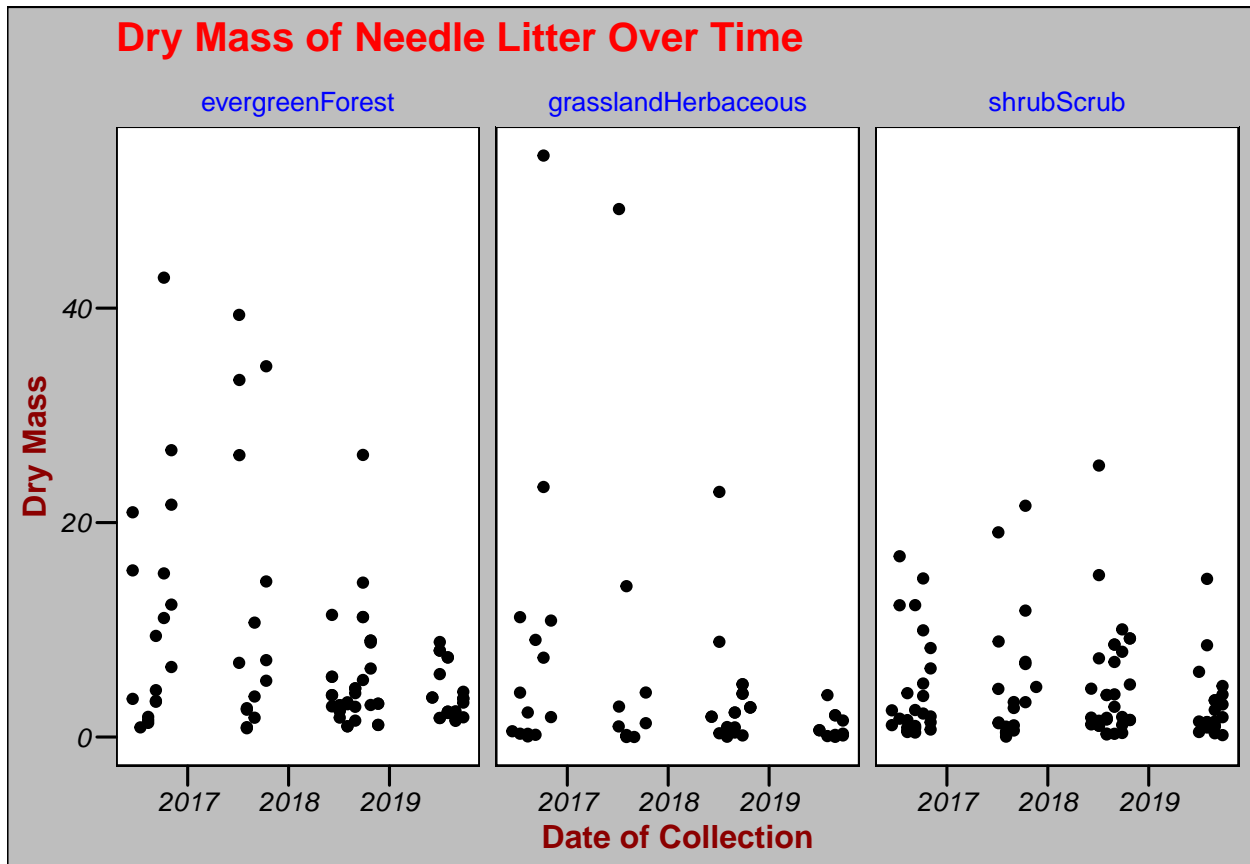
Q6_Plot <- ggplot(Niwot.Ridge.needles, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(title = "Dry Mass of Needle Litter Over Time",
       x = "Date of Collection",
       y = "Dry Mass",
       color = "NLCD") +
  theme(legend.text = element_text(size = 10)) #Change the text size in order to fit.

Q6_Plot
```



```
#7
Q7_facet <- ggplot(Niwot.Ridge.needles, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), ncol = 3) +
  labs(title = "Dry Mass of Needle Litter Over Time",
       x = "Date of Collection",
       y = "Dry Mass") +
  theme(strip.text = element_text(size = 10, color = "blue"), #Change title size of facets to fit.
        strip.background = element_rect(fill = "grey")) #Change facet title background to comply with m

Q7_facet
```



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

Answer: I think faceted graph is more effective in this case. Even the point for NLCD Class is shown in different colors, every dot in the same plot seems confusing for me.