Assignment 5: Data Visualization

Meghan Seyler

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).

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

```
#1
getwd()
```

```
## [1] "Z:/ENV872/Environmental_Data_Analytics_2022/Assignments"
```

```
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)

PeterPaul.chem.nutrients <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Pro
stringsAsFactors = TRUE)
```

```
NEON_NIWO_Litter<-read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
stringsAsFactors = TRUE)


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

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

```
#sampledate is a factor
class(NEON_NIWO_Litter$collectDate)
```

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

```
#collectdate is a factor
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#load lubridate to change date columns from factor to date

PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
#change sampledate to date in year month day order
NEON_NIWO_Litter$collectDate<-ymd(NEON_NIWO_Litter$collectDate)
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
#sampledate changed from factor to date
class(NEON_NIWO_Litter$collectDate)
```

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

```
#collectdate changed from factor to date

PeterPaul.chem.nutrients$month<- as.factor(PeterPaul.chem.nutrients$month)
class(PeterPaul.chem.nutrients$month)
```

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

```
#Change month to factor so that the boxplots in question four are created with a boxplot per month
```

Define your theme

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

```
#3
customtheme<-theme_light(base_size = 14) +
#increasing axis font size from 11 to 14
  theme(axis.text = element_text(color = "black"),
```

```
  #changing axis titles to a bolder black font
        legend.position = "right")
        #position the legend on the right


theme_set(customtheme)
#setting customtheme as default
```

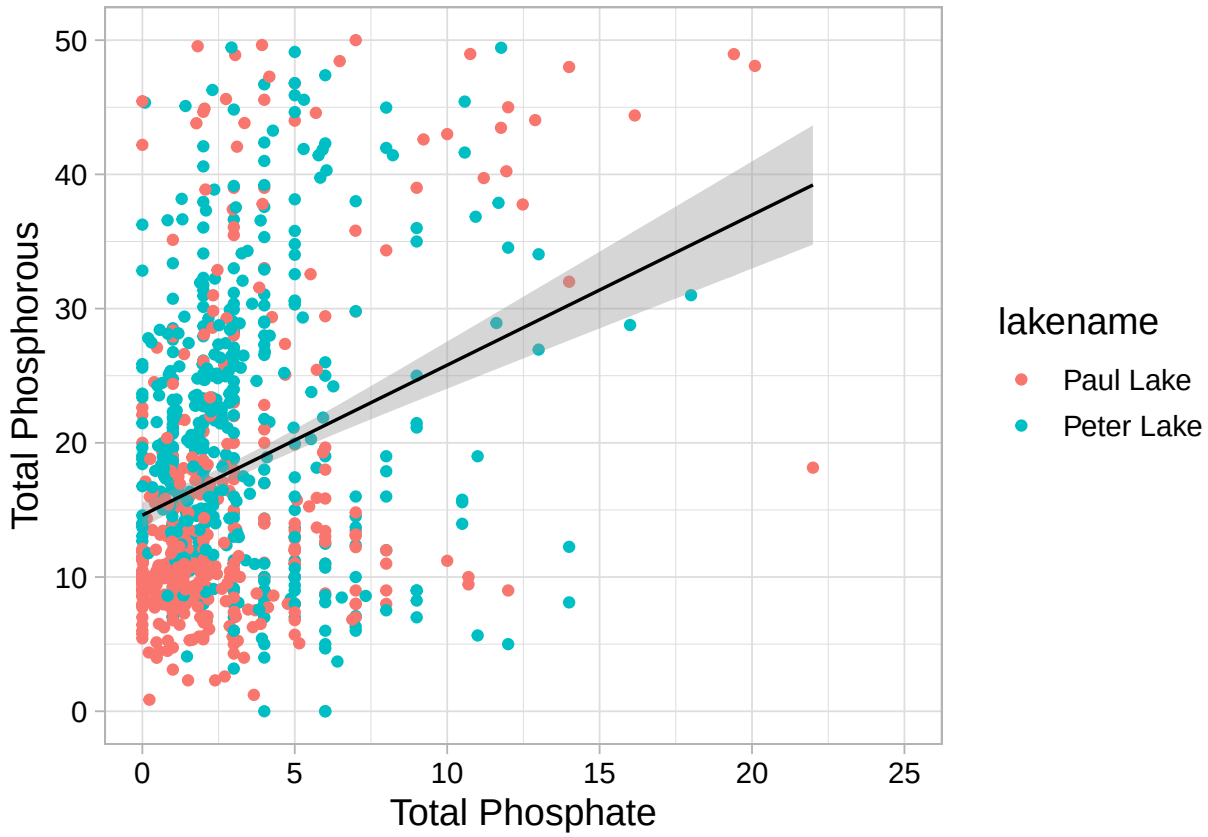
Create graphs

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()`).

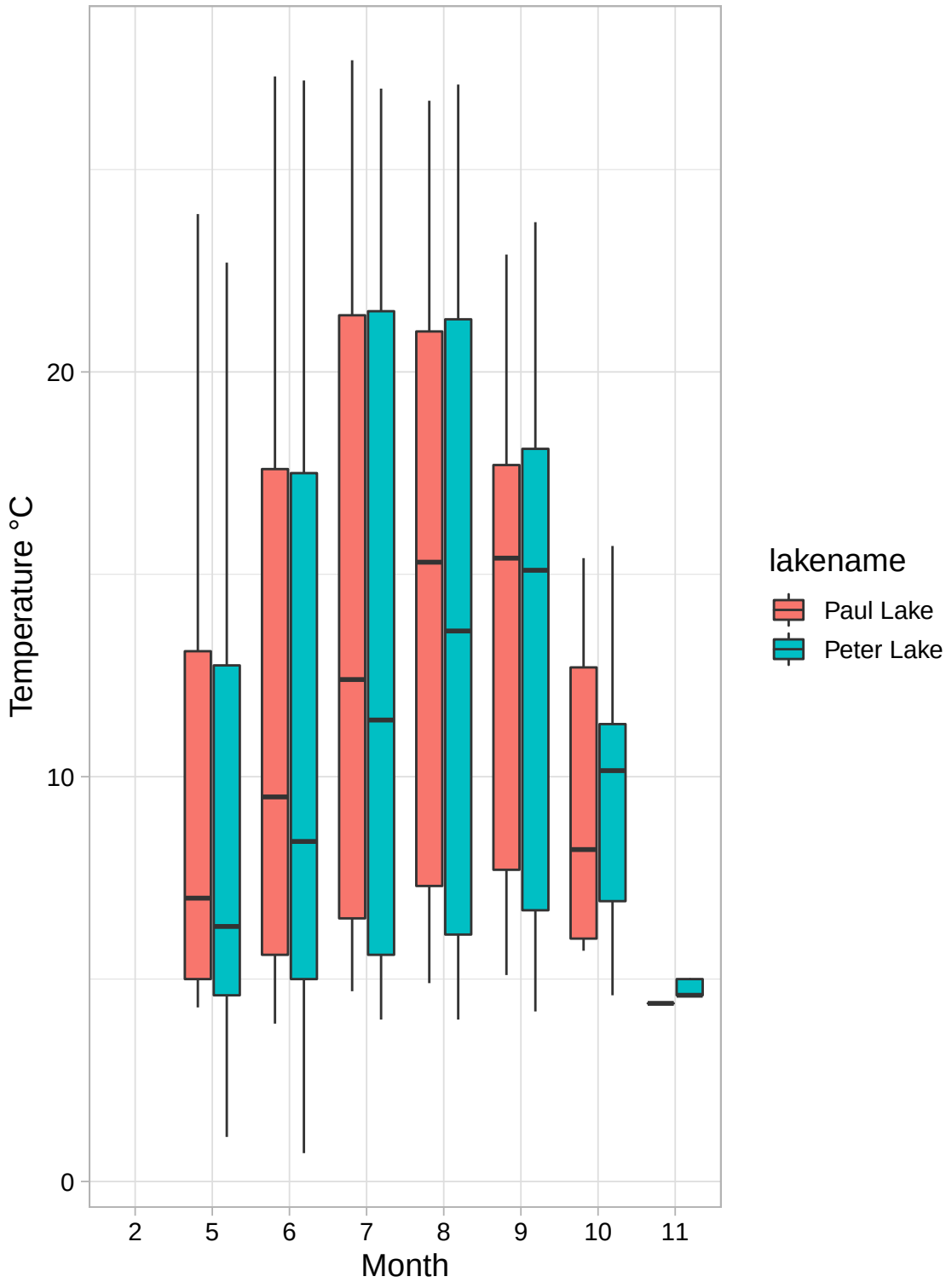
```
#4
ggplot(filter(PeterPaul.chem.nutrients), aes(x=po4, y=tp_ug, color= lakename)) +
  geom_point() +
  #separate aesthetics for Peter and Paul lakes
  geom_smooth(method = "lm", color = "black" , size = .60) +
  xlim(0, 25) +
  #Adjusting x axis to hide extremes, I chose to zoom in this much to really differentiate the points n
  ylim(0,50)+
  #adjusting y axis to hide extremes, I chose to zoom in this much to really differentiate the points n
  ylab("Total Phosphorous") +
  xlab ("Total Phosphate")
```

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



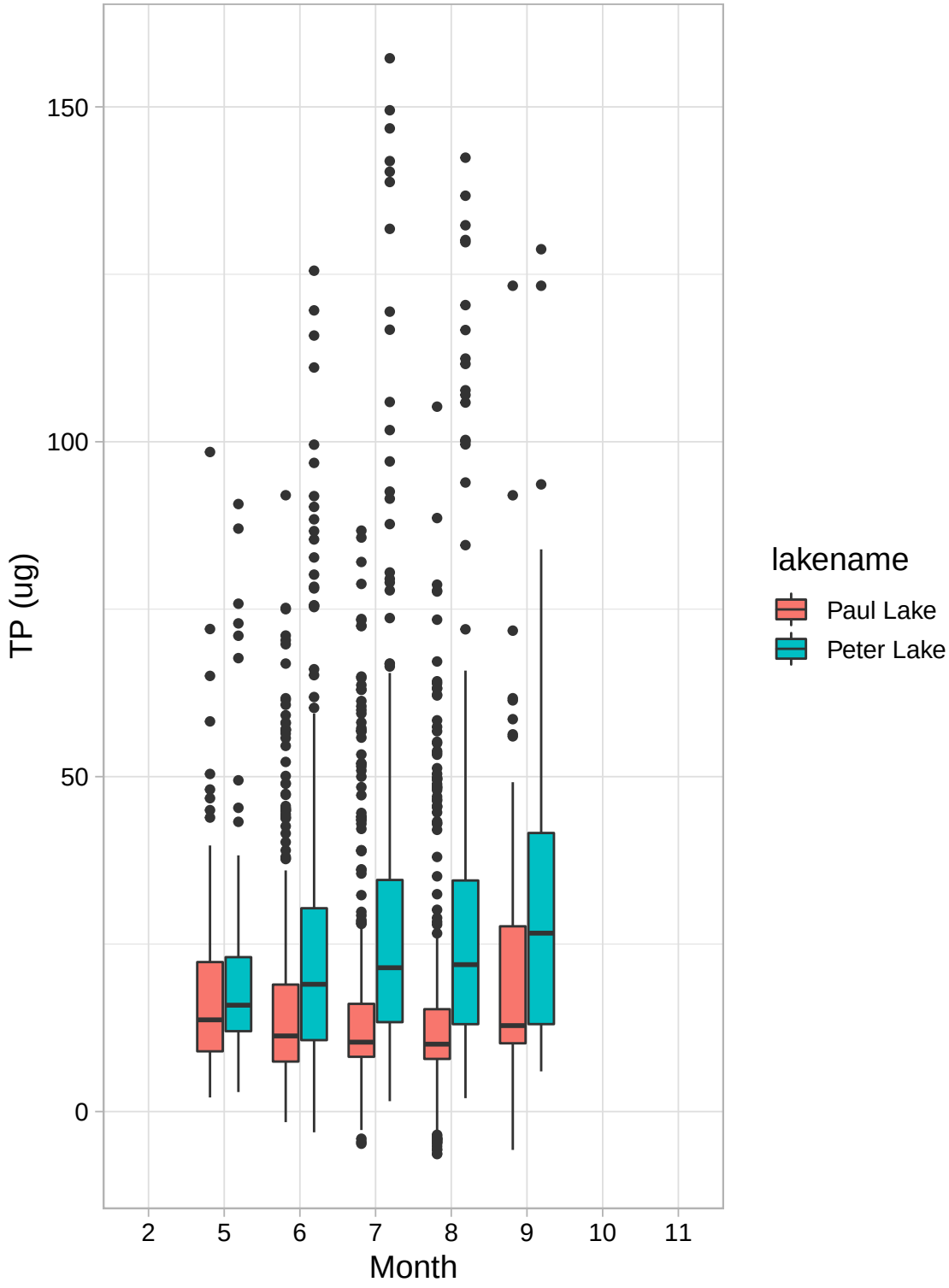
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
#5
#how to add scale_x_date(date_labels = "%b")
library(cowplot)
temperature_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month,
  y = temperature_C, fill = lakename)) +
  #making the boxpots fill according to lake name
  geom_boxplot() + #make boxplot
  ylab("Temperature °C")+ #change y axis label
  xlab("Month") #change x axis label
print(temperature_plot)
```



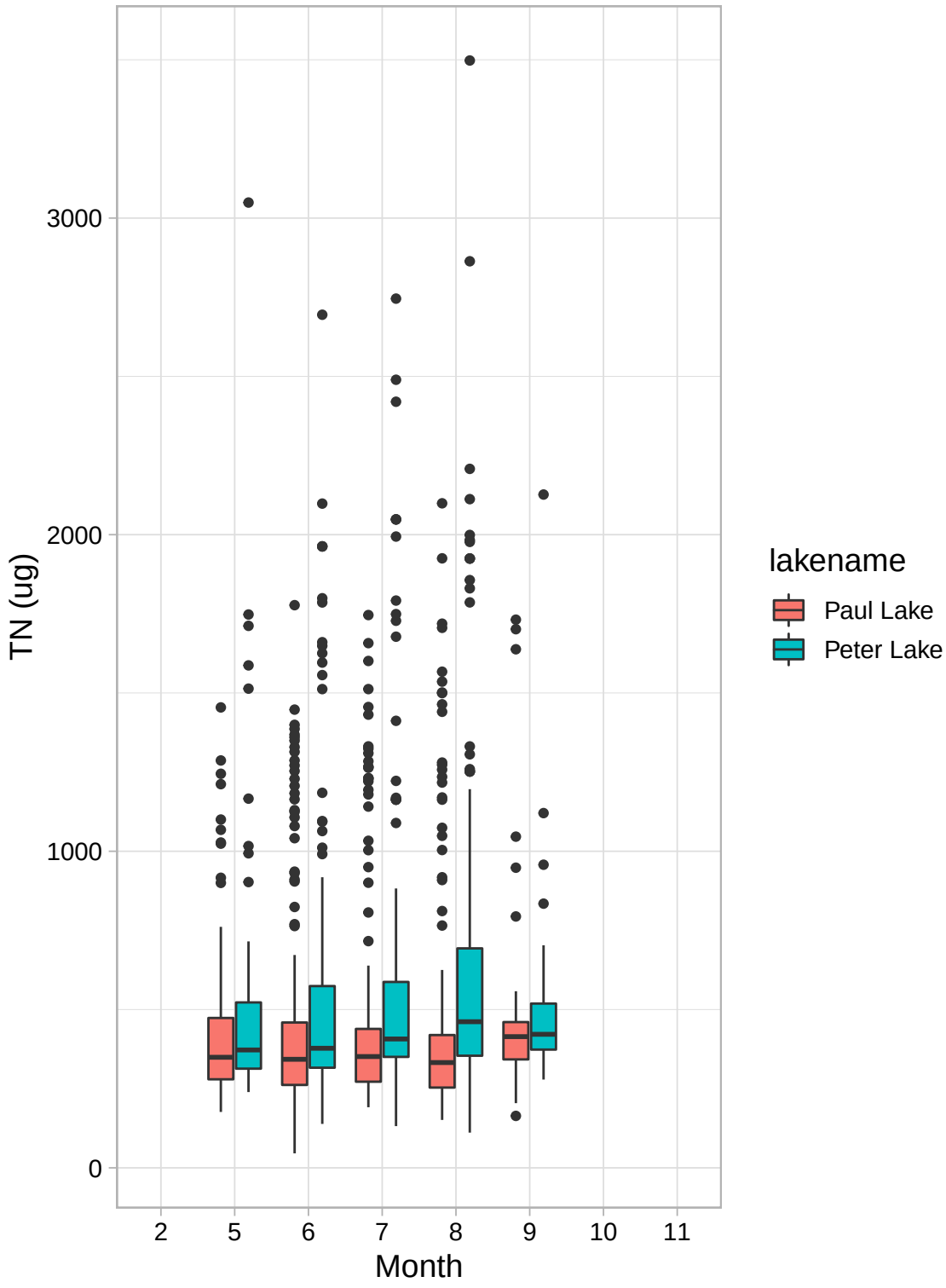

```
tp_ug_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug, fill = lakename)) +
  geom_boxplot() +
  ylab("TP (ug)")+
  xlab("Month")
```

```
print(tp_ug_plot)
```

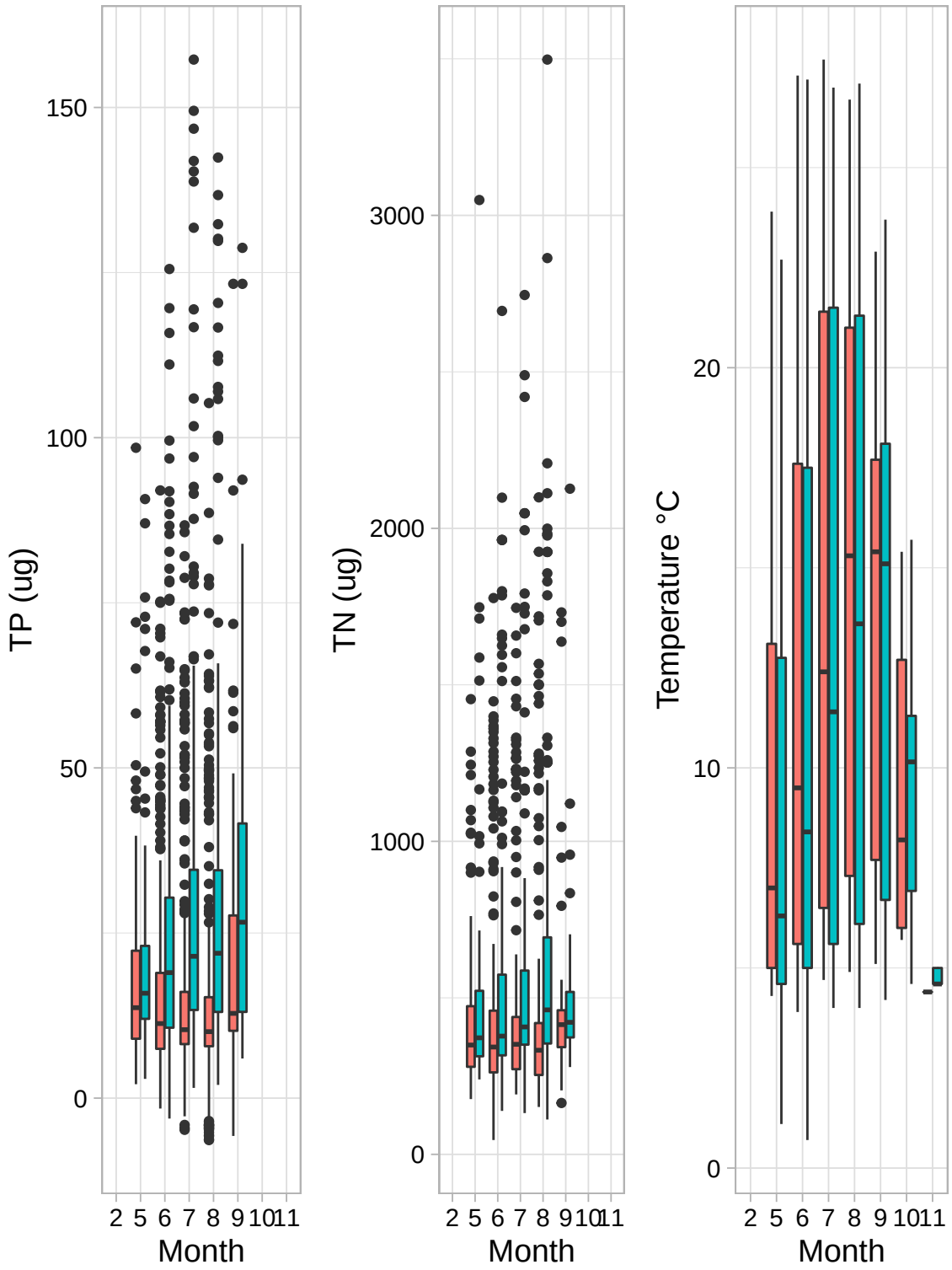



```
tn_ug_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug, fill = lakename)) +
  geom_boxplot() +
```

```
  ylab("TN (ug)")+
  xlab("Month")
print(tn_ug_plot)
```

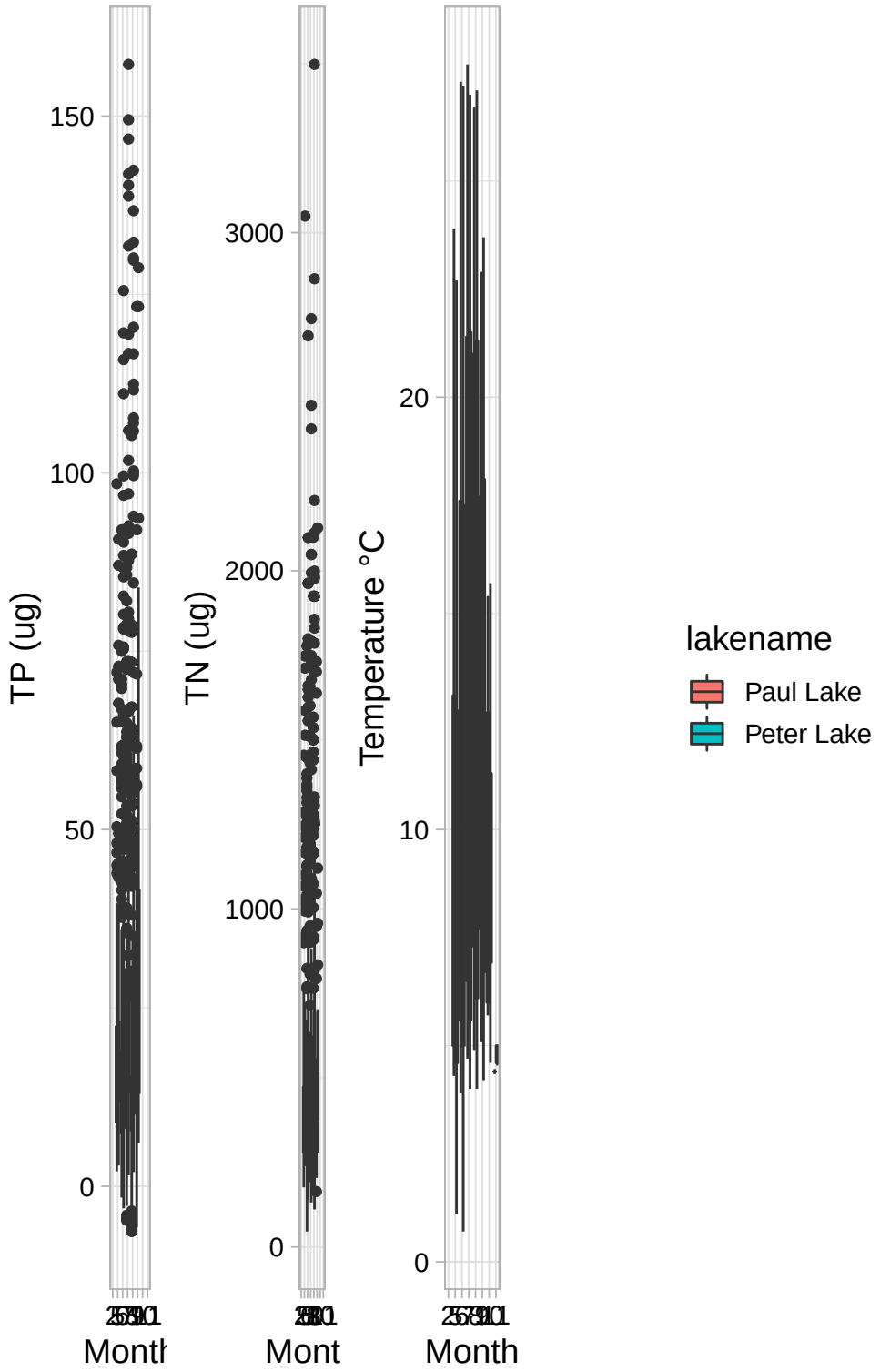


```
temp_tp_tn_box<-plot_grid(
  tp_ug_plot + theme(legend.position = "none"),
  #removing all legends
  tn_ug_plot + theme(legend.position = "none"),
  temperature_plot + theme (legend.position = "none"),
  nrow =1,
  #placing all three plots in one row
  rel_heights = c(2,0.1 ))
  #aligning graphs horizontally and vertically

print(temp_tp_tn_box)
```




```
legend<-get_legend(temperature_plot)
#creating seperate legend

temp_tp_tn_box_legend<-plot_grid(
  temp_tp_tn_box,
```

```
  legend,
  rel_heights = c(2,0.1 ),
  nrow = 1,
  align = "hv"
  )
#combining combined box plots and legend
print(temp_tp_tn_box_legend)
```



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

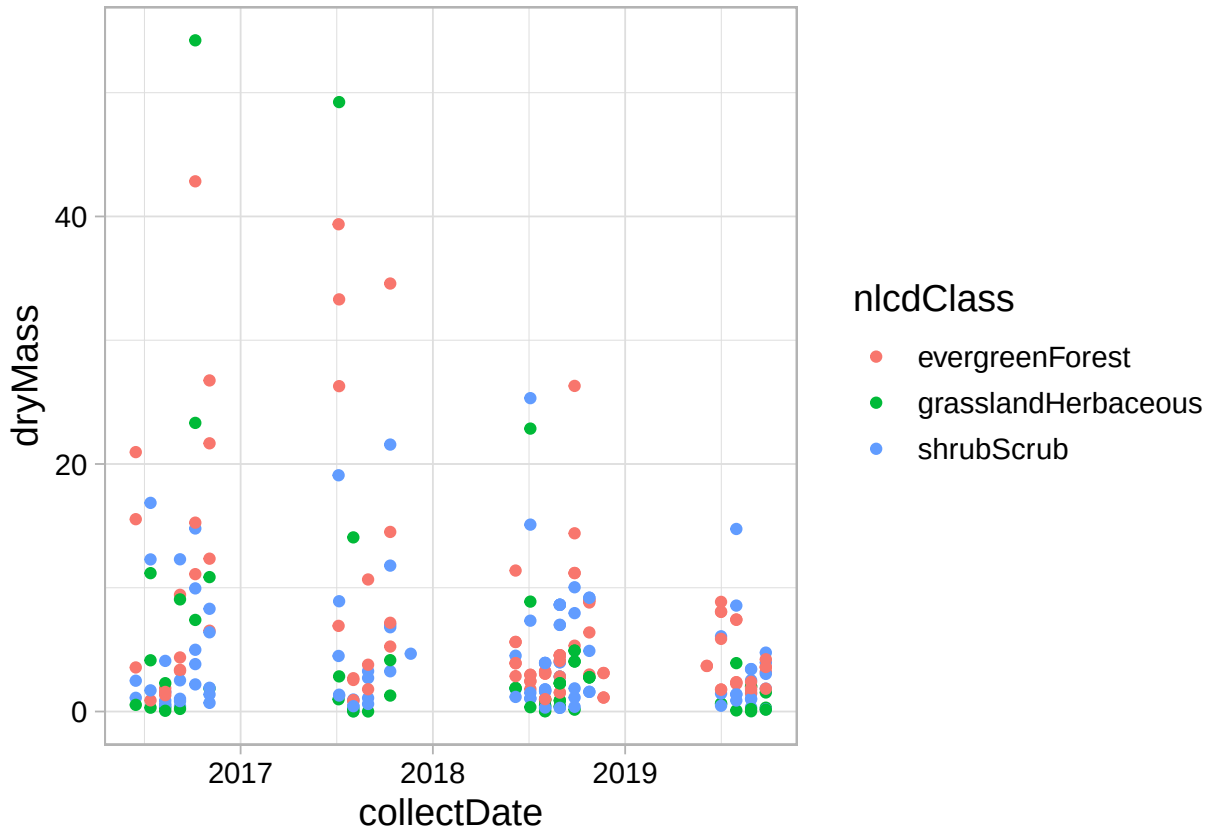
Answer: Temperature - Paul lake has a higher median temperature than Peter Lake. TP - Paul lake has a lower median total phosporus (ug) than Peter Lake. TN - Paul lake has a lower median total nitrogen (ug) than Peter Lake.

Also, it is interesting that Nitrogen and phosphorous are only measured during the summer months and temperature is measured over summer and fall.

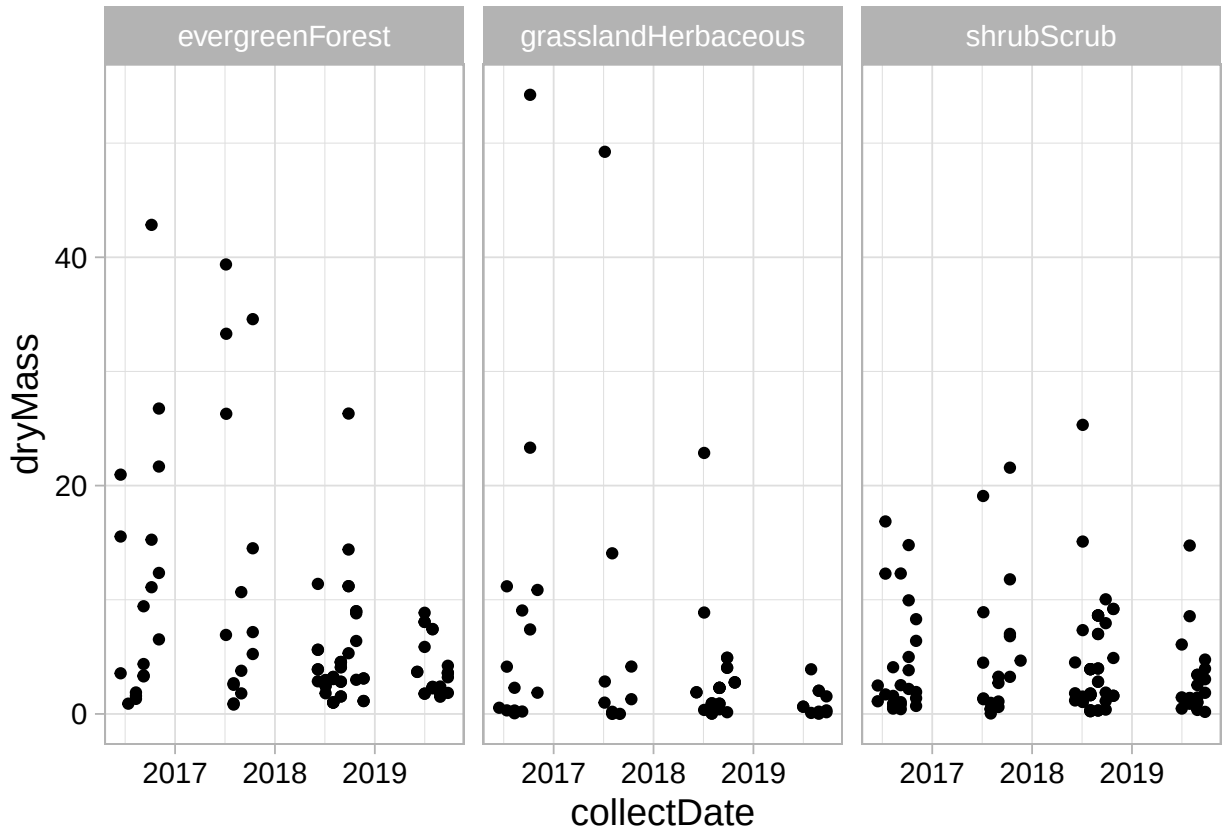
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
NEON_NIWO_Litter_filterNeedle<-NEON_NIWO_Litter %>%
  filter(functionalGroup %in% c("Needles"))

Needle_point<-ggplot(NEON_NIWO_Litter_filterNeedle)+
  geom_point( aes(x  = collectDate, y = dryMass, color = nlcdClass ))

print(Needle_point)
```




```
#7
NEON_NIWO_Litter_facet<-ggplot(filter(NEON_NIWO_Litter_filterNeedle))+
  geom_point(aes(x  = collectDate, y = dryMass))+
  facet_wrap('nlcdClass')
#making three different plots based on nlcdClass
print(NEON_NIWO_Litter_facet)
```



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

> Answer: Plot 6 is more effective than plot 7 because it makes it easier to compare the dry mass of each class each year. When the classes are seperated it is more dificult to interpret how they stack up against one another.However, if you were looking to specifically analyze each nlcd class individually #7 is more effective because it has less overlapping points.