Assignment 5: Data Visualization

Wynona Curaming

Spring 2024

OVERVIEW

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

---

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)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/ENV 872/EDA_Spring2024
```

```
library(cowplot)
```

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

```
library(ggthemes)
```

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

```
getwd()
```

```
## [1] "/home/guest/ENV 872/EDA_Spring2024"
```

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

Niwot.litter.data <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = T)

#2

PeterPaul.chem.nutrients.data$sampledate<-ymd(PeterPaul.chem.nutrients.data$sampledate)
Niwot.litter.data$collectDate<-ymd(Niwot.litter.data$collectDate)

class(PeterPaul.chem.nutrients.data$sampledate)
```

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

```
class(PeterPaul.chem.nutrients.data$sampledate)
```

```
## [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

```r
#3
my_theme <- theme_base() + theme(line=element_line(color='black',linewidth=1),
                                  plot.background = element_rect(color='beige', fill='beige'),
                                  plot.title = element_text(color='black', size=12),
                                  legend.background = element_rect(
      color='darkgrey',
      fill= 'darkgrey'),legend.text = element_text(color='white',size=12),
      axis.title = element_text(size=12),
      panel.grid.major = element_line(color = "lightgray", linewidth = 0.5),
    panel.grid.minor = element_line(color = "lightgray", linewidth = 0.25)
    )
```

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 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()`).

```r
#4
colnames(PeterPaul.chem.nutrients.data)
```

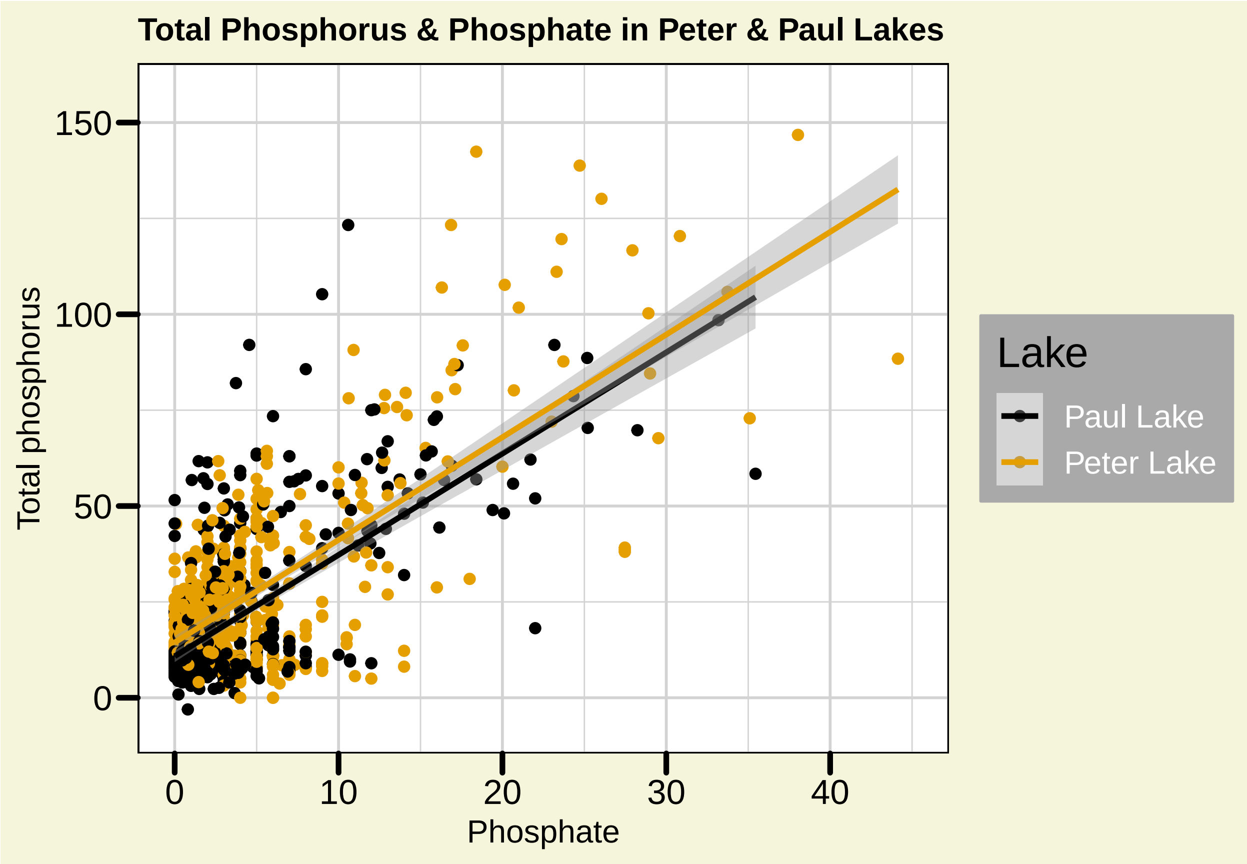
```
##  [1] "lakename"        "year4"           "daynum"          "month"
##  [5] "sampledate"      "depth"           "temperature_C"   "dissolvedOxygen"
##  [9] "irradianceWater" "irradianceDeck"  "tn_ug"           "tp_ug"
## [13] "nh34"            "no23"            "po4"
```

```r
ggplot(PeterPaul.chem.nutrients.data,
       aes(x=po4, y=tp_ug,color=lakename))+
  geom_point()+geom_smooth(method=lm)+xlim(0,45)+
  my_theme+ labs(x="Phosphate",y="Total phosphorus",color="Lake")+
  scale_color_colorblind()+ggtitle("Total Phosphorus & Phosphate in Peter & Paul Lakes")
```

```
## `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.

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
bp.temp<-ggplot(PeterPaul.chem.nutrients.data,
                aes(x=factor(month),y=temperature_C,color=lakename))+
  geom_boxplot()+ylab(label='Temperature')+
  my_theme+theme(legend.position = 'none',
                 axis.title.x=element_blank())+
  scale_color_colorblind()+ggtitle("Lake Conditions Over Time")

bp.tp<-ggplot(PeterPaul.chem.nutrients.data,
              aes(x=factor(month),y=tp_ug))+geom_boxplot(aes(color=lakename))+
  my_theme+theme(legend.position = 'none',
                 axis.title.x=element_blank())+
  ylab(label="Phosphate")+
  scale_color_colorblind()

bp.tn<-ggplot(PeterPaul.chem.nutrients.data,
```

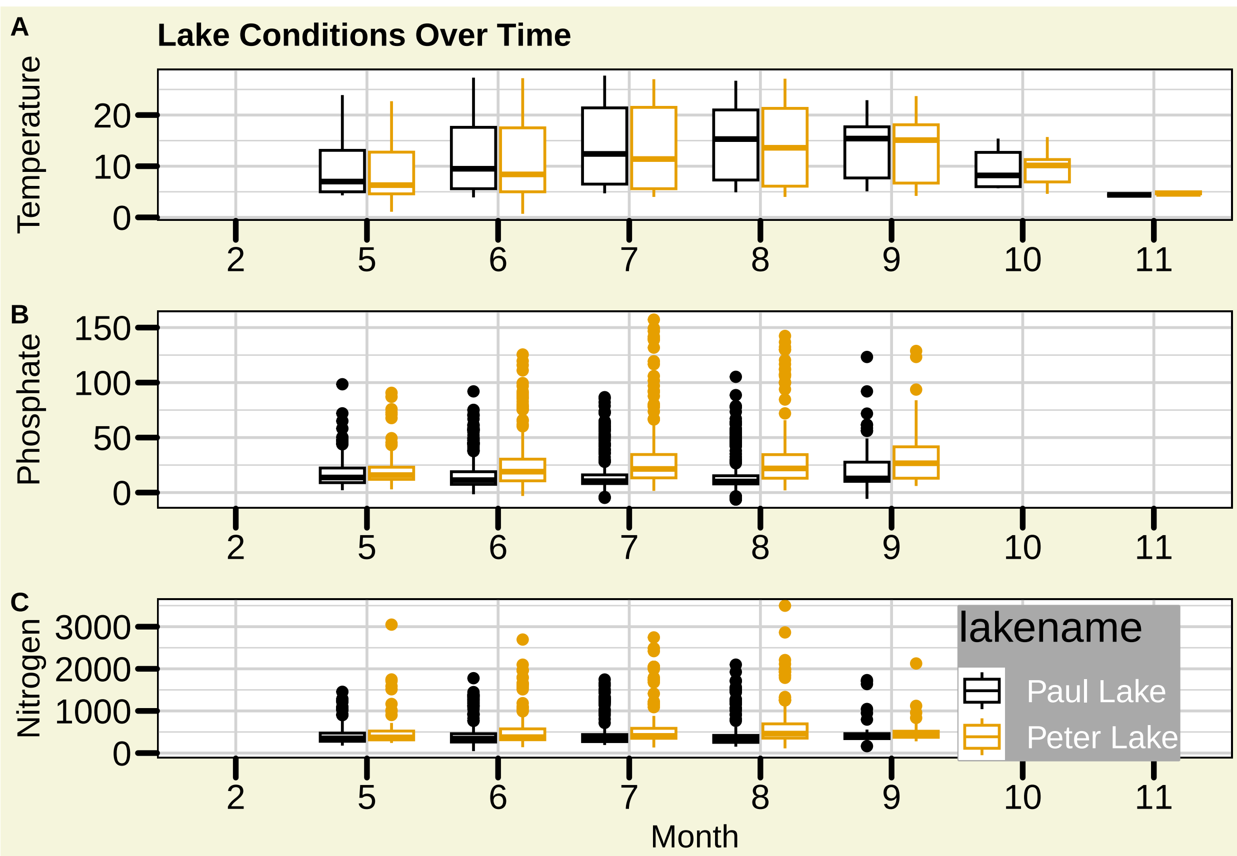
```r
              aes(x=factor(month),y=tn_ug))+geom_boxplot(aes(color=lakename))+
  my_theme+theme(legend.position = c(.95, .95),
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(0, 0, 0, 0))+
  xlab(label="Month")+ylab(label="Nitrogen")+
  scale_color_colorblind()

#install.packages(cowplot)
library(cowplot)
plot_grid(bp.temp, bp.tp, bp.tn, ncol=1, align="v",
          labels = c('A', 'B', 'C'), label_size = 10)
```

```
## 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: There is sparse/no data in winter months (Oct, Nov, Dec, Jan, and Feb). I think this is because nitrogen and phosphate largely come from fertilizer runoff and perhaps during these months, farmers near these lakes aren't growing crops (not adding fertilizer to fields). As the

seasons get hotter (eg. approaching summer), the lake temperature also gets hotter. There is a generally a slightly higher TP and TN content in Peter Lake (yellow) than Paul Lake (black). Phosphate content on average increases from May to Sep in Peter Lake whereas it is fairly constant for 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
colnames(Niwot.litter.data)
```

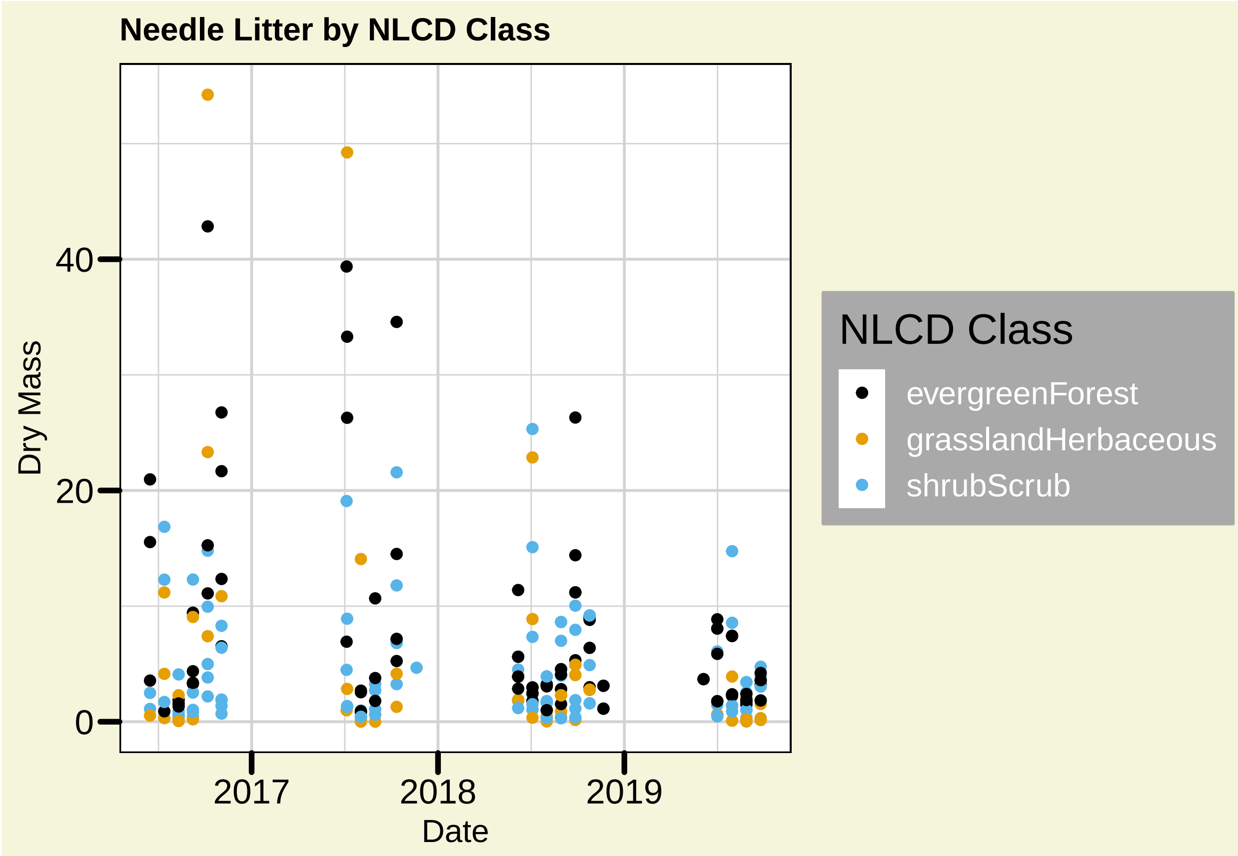
```
##  [1] "plotID"           "trapID"           "collectDate"      "functionalGroup"
##  [5] "dryMass"          "qaDryMass"        "subplotID"        "decimalLatitude"
##  [9] "decimalLongitude" "elevation"        "nlcdClass"        "plotType"
## [13] "geodeticDatum"
```

```
Niwot.litter.data$collectDate<-ymd(Niwot.litter.data$collectDate)
class(Niwot.litter.data$collectDate)
```

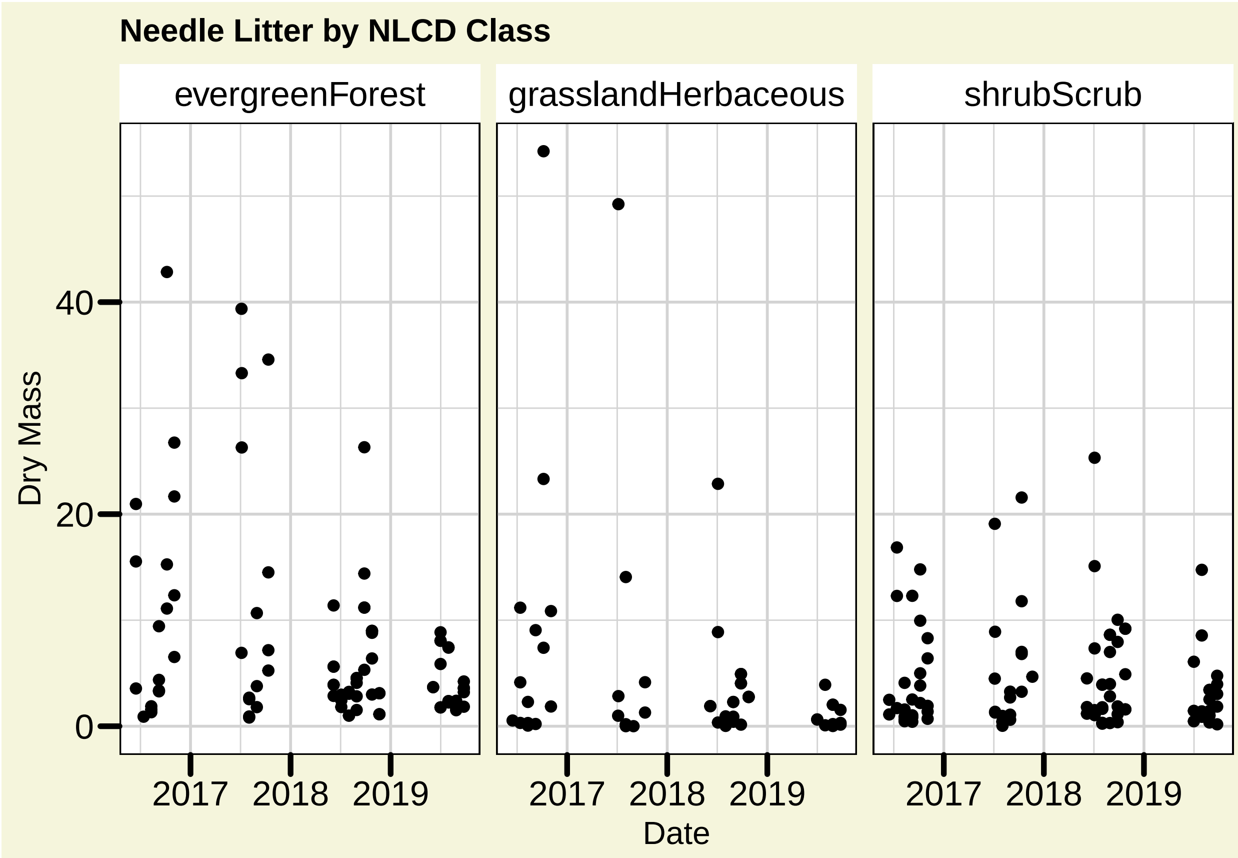
```
## [1] "Date"
```

```
needles<-filter(Niwot.litter.data, functionalGroup=="Needles")
ggplot(needles, aes(x=collectDate, y=dryMass))+
  geom_point(aes(color=nlcdClass))+my_theme+
  labs(x="Date", y="Dry Mass",color="NLCD Class")+
  scale_color_discrete(labels = c("Evergreen Forest", "Grassland", "Shrub"))+
  scale_color_colorblind()+ggtitle("Needle Litter by NLCD Class")
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```




```
#7
ggplot(needles, aes(x=collectDate, y=dryMass))+geom_point()+
  facet_wrap(needles$nlcdClass,
             labeller = labeller(nlcdClass=c("Evergreen Forest","Grassland","Shrub")))+
  my_theme+ labs(x="Date", y="Dry Mass")+
  scale_color_colorblind()+ggtitle("Needle Litter by NLCD Class")
```



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

> Answer: Plot 7 is more effective because it is easier to see how the dry mass each NLCD class changes over time.