Assignment 5: Data Visualization

Jun Hu

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.

```r
#1
library(tidyverse);library(lubridate);library(here);library(cowplot)
```

```
## -- 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()
```

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

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

## here() starts at /Users/jun/Desktop/EDA/EDA-Spring2023

##
## Attaching package: 'cowplot'

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

```
here()
```

```
## [1] "/Users/jun/Desktop/EDA/EDA-Spring2023"
```

```
processed_data = "Data/Processed_KEY"
PeterPaul.chem.nutrients <- read.csv(
  here(processed_data,"NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)
NEON.data <- read.csv(
  here(processed_data,"NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)
#2
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
NEON.data$collectDate <- ymd(NEON.data$collectDate)
```

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
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
```

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.

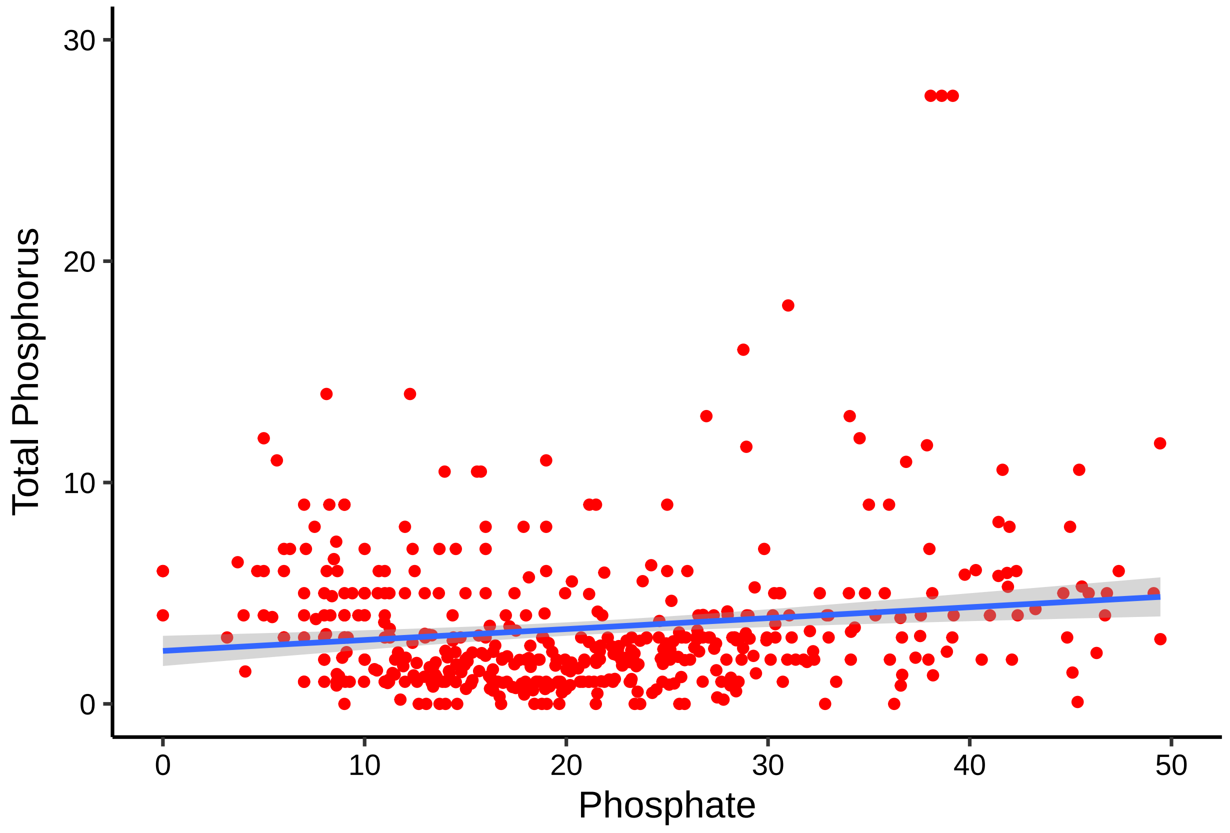
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
Peter_phosphorous_plot <- PeterPaul.chem.nutrients %>%
  filter(lakename == "Peter Lake") %>%
  ggplot(aes(
      x=tp_ug,
      y=po4))+
  geom_point( color="red" ) +
  xlim(0,50)+
  ylim(0,30)+
  ylab(expression ("Total Phosphorus"))+
  xlab(expression("Phosphate"))+
  geom_smooth(method = lm) +
  mytheme
print(Peter_phosphorous_plot)
```

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

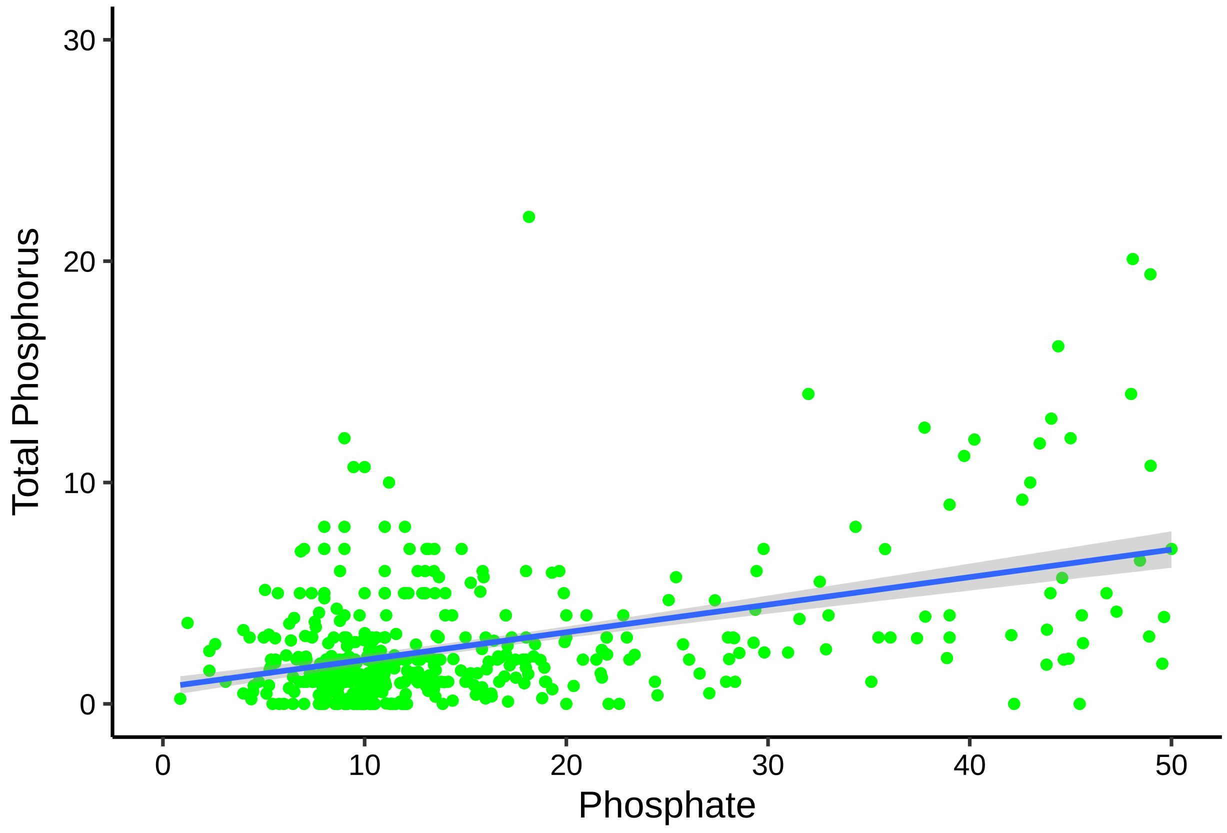
```
## Warning: Removed 11480 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 11480 rows containing missing values (geom_point).
```



```
Paul_phosphorous_plot <- PeterPaul.chem.nutrients %>%
  filter(lakename == "Paul Lake") %>%
  ggplot(aes(
      x=tp_ug,
      y=po4))+
  geom_point( color="green" ) +
  xlim(0,50)+
  ylim(0,30)+
  ylab(expression ("Total Phosphorus"))+
  xlab(expression("Phosphate"))+
  geom_smooth(method = lm) +
  mytheme
print(Paul_phosphorous_plot)
```

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

## Warning: Removed 10582 rows containing non-finite values (stat_smooth).

## Warning: Removed 10582 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 build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

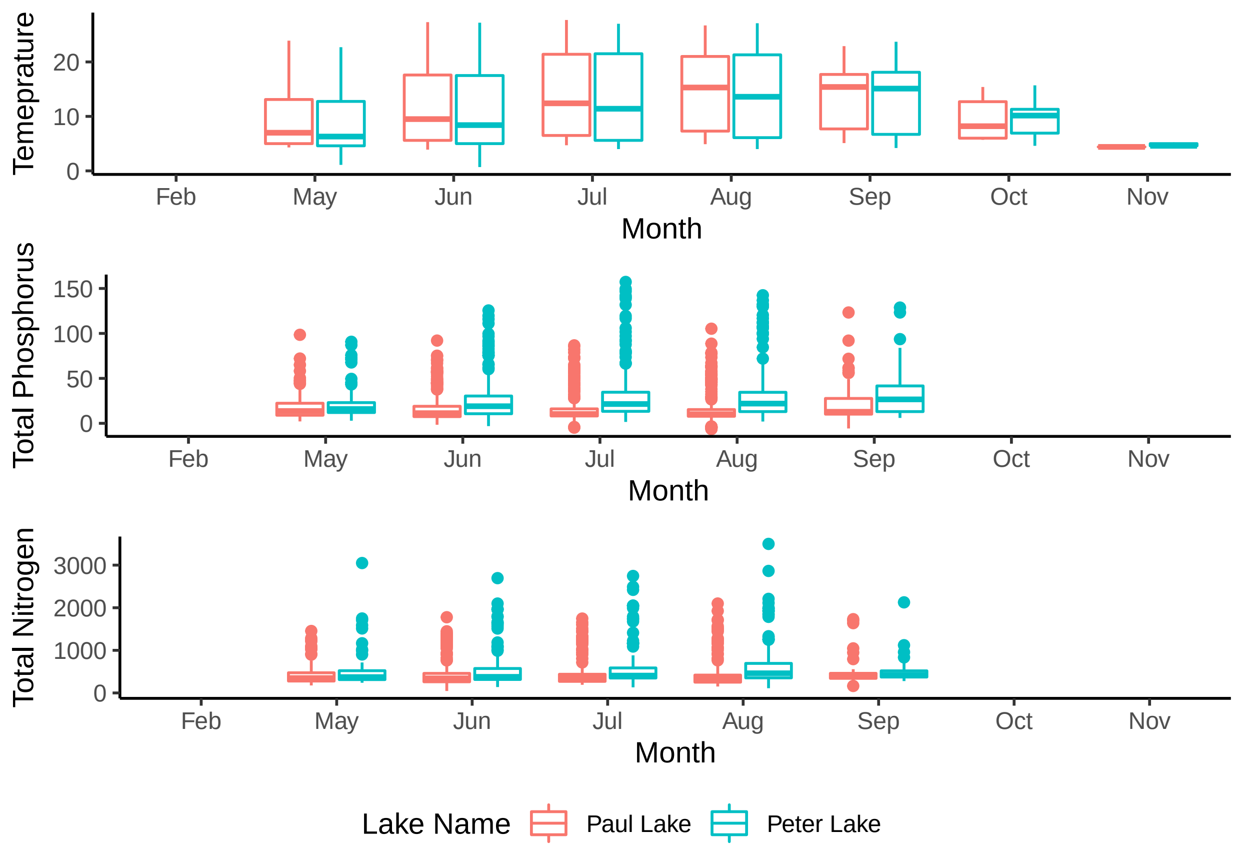
```
#5

Box_temp_plot <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
      (x=factor(month,level=1:12, label= month.abb)),
      y=temperature_C, color= lakename)) +
  geom_boxplot() +
  ylab(expression ("Temeprature"))+
  xlab(expression ("Month"))+
  theme_classic()+
  theme(legend.position="none")


Box_TP_plot <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
      (x=factor(month,level=1:12, label= month.abb)),
      y=tp_ug, color= lakename)) +
  geom_boxplot() +
  ylab(expression ("Total Phosphorus"))+
  xlab(expression ("Month"))+
  theme_classic()+
  theme(legend.position="none")


Box_TN_plot <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
      (x=factor(month,level=1:12, label= month.abb)),
      y=tn_ug, color= lakename)) +
  geom_boxplot() +
  ylab(expression ("Total Nitrogen"))+
  xlab(expression ("Month"))+
  labs(color="Lake Name")+
  theme_classic()+
  theme(legend.position="none")

combined_plot<-plot_grid(Box_temp_plot, Box_TP_plot,Box_TN_plot,ncol=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).
```

```
legend <- get_legend(
  Box_TN_plot +
    guides(color = guide_legend(nrow = 1)) +
    theme(legend.position = "bottom")
)
```

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

```
plot_grid(combined_plot, legend,ncol=1,rel_heights = c(1, .1))
```

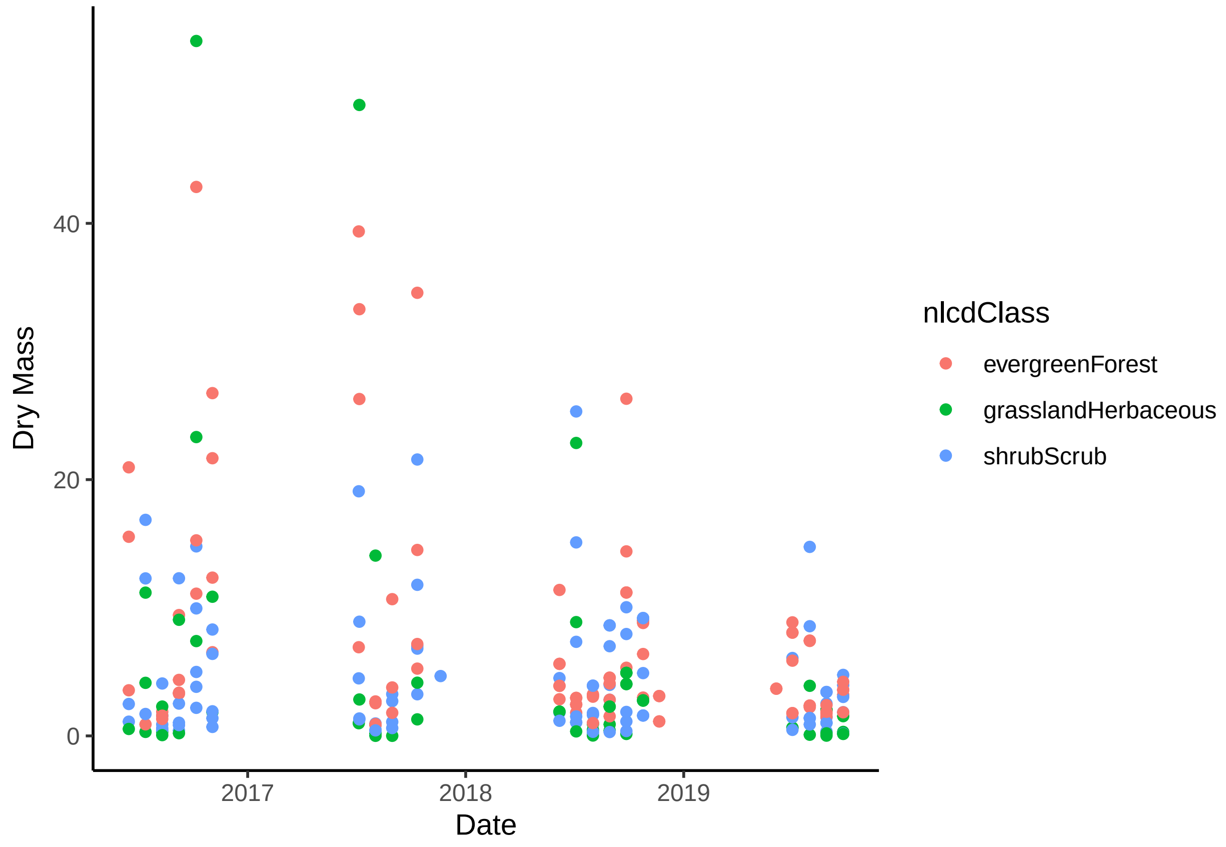


Question: What do you observe about the variables of interest over seasons and between lakes?

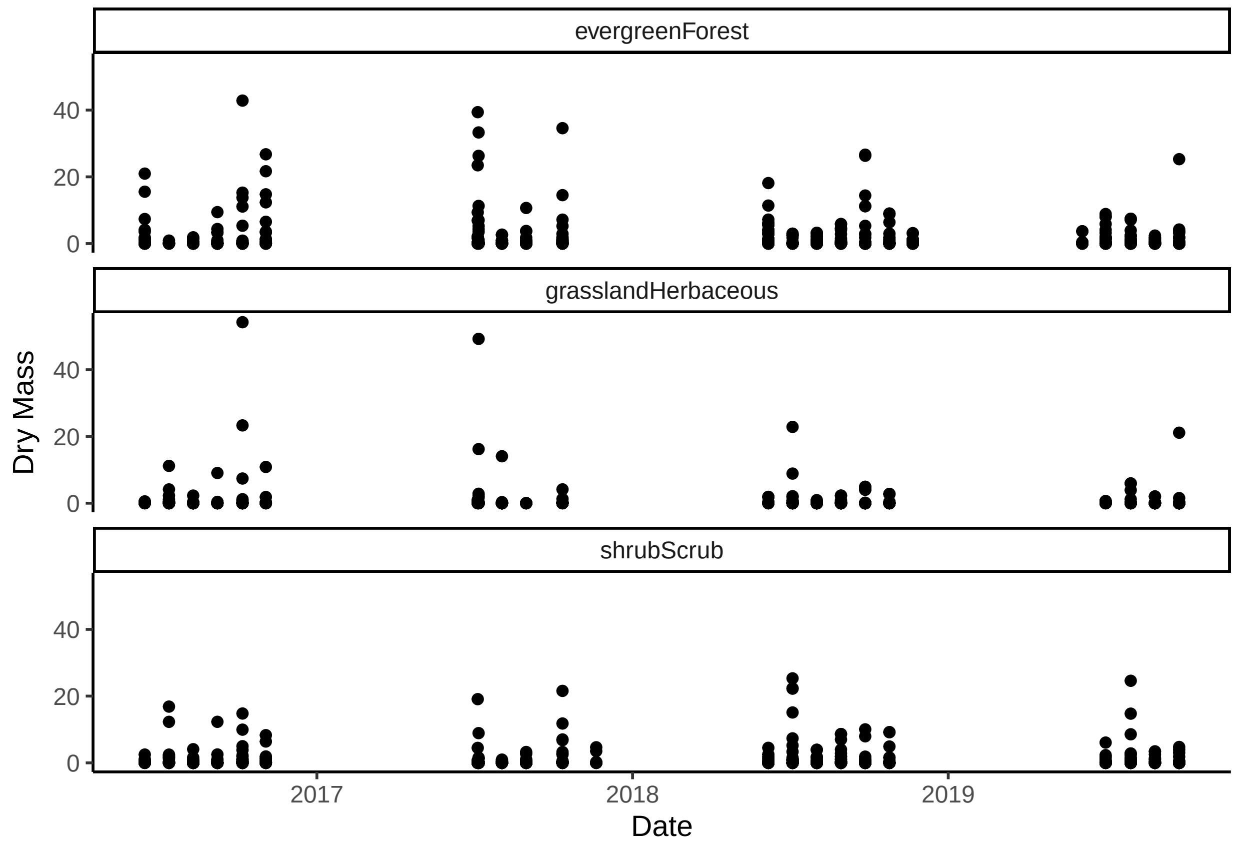
> Answer: During the summer seasons, the water temeprature, total phosphorus, and total nitrogen are higher than other seasons. Through May to September, Paul lake has higher temeperature than Peter Lake. Peter lake show higher total phosphorus and nitrogen than Paul lake through all the years.

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
Needles_plot <- NEON.data %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    aes(
      x=collectDate,
      y=dryMass,
      color=nlcdClass)
    ) +
  geom_point() +
  ylab(expression ("Dry Mass"))+
  xlab(expression ("Date"))+
  theme_classic()
print(Needles_plot)
```



```
#7
Needles_plot.faceted <-
  ggplot(NEON.data, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  ylab(expression ("Dry Mass"))+
  xlab(expression ("Date"))+
  theme_classic()
print(Needles_plot.faceted)
```



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 the comparision is more evident among three NLCD class by date