Assignment 5: Data Visualization

Mason Ibrahim

Fall 2024

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
#Load packages
library(tidyverse)
```

```
## -- 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
```

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

```
## here() starts at /home/guest/R/EDE_Fall2024
```

```
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
```

```
#Verify working directory
getwd()
```

```
## [1] "/home/guest/R/EDE_Fall2024"
```

```
#Read in data
df_peterpaul_chem_nutrients <-
  read.csv(
    here(
      "Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"
    ),
    stringsAsFactors = TRUE
  )
df_niwot_litter <- read.csv(
  here(
    "Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"
  ),
  stringsAsFactors = TRUE
)


#2
#Change date format
df_peterpaul_chem_nutrients$sampledate <- ymd(df_peterpaul_chem_nutrients$sampledate)
df_niwot_litter$collectDate <- ymd(df_niwot_litter$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
my_theme <- theme_base() +
  theme(
    rect = element_rect(color = "darkgreen"),
    text = element_text(color = "goldenrod"),
    plot.title = element_text(
      size = 16,
      face = "bold",
      color = "goldenrod"
    ),
    axis.title.x = element_text(size = 7, color = "goldenrod"),
    axis.title.y = element_text(size = 7, color = "goldenrod"),
    axis.text = element_text(size = 7, color = "palegreen4"),

    axis.ticks = element_line(color = "palegreen4"),
    panel.grid.major = element_line(color = "palegreen4"),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "white", color = NA),
    legend.key = element_rect(fill = "white", color = "goldenrod"),
    legend.background = element_rect(fill = "palegreen4", color = "goldenrod")
  )

complete = 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.

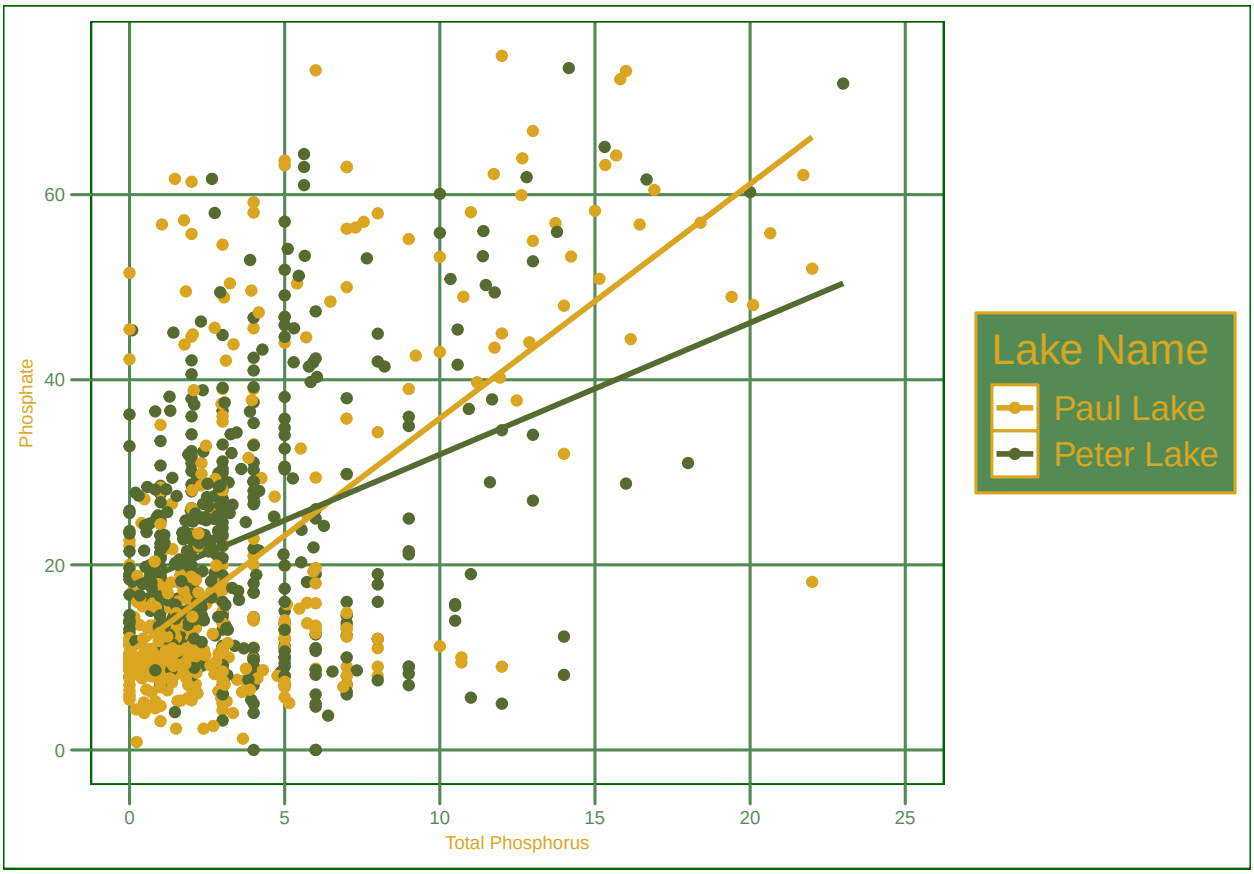
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
df_peterpaul_chem_nutrients %>%
  ggplot(aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  xlim(0, 25) +
  ylim(0, 75) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("goldenrod", "#556B2F")) +
  labs(x = "Total Phosphorus", y = "Phosphate", color = "Lake Name") +
  my_theme
```

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

## Warning: Removed 21996 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 21996 rows containing missing values or values outside the scale range
## (`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
#Get month abbs
df_peterpaul_chem_nutrients$month <-
  as.numeric(as.integer(df_peterpaul_chem_nutrients$month))
df_peterpaul_chem_nutrients$month <-
  factor(df_peterpaul_chem_nutrients$month,
         levels = 1:12,
```

```
        labels = month.abb)

#Box plot for temperature
temp_month <-
  df_peterpaul_chem_nutrients %>%
  ggplot(aes
         (x = month, color = lakename)) +
  geom_boxplot(aes
               (y = temperature_C)) +
  labs(x = "Month", y = "Temperature (°C)", color = "Lake Name") +
  my_theme +
  theme(
    axis.title.x = element_blank(),
    legend.position = "none" ,
    panel.grid.major = element_blank(),
    axis.text.x = element_text(
      angle = 90,
      hjust = 1,
      vjust = 0.5
    )
  ) +
  scale_color_manual(values = c("goldenrod", "#556B2F")) +
  scale_x_discrete(drop = FALSE)

#Box plot for TP
tp_month <-
  df_peterpaul_chem_nutrients %>%
  ggplot(aes
         (x = month, color = lakename)) +
  geom_boxplot(aes
               (y = tp_ug)) +
  labs(x = "Month", y = "Total Phosphorus") +
  my_theme +
  theme(
    axis.title.x = element_blank(),
    legend.position = "none" ,
    panel.grid.major = element_blank(),
    axis.text.x = element_text(
      angle = 90,
      hjust = 1,
      vjust = 0.5
    )
  ) +
  scale_color_manual(values = c("goldenrod", "#556B2F")) +
  scale_x_discrete(drop = FALSE)

#Box plot for TN
tn_month <- df_peterpaul_chem_nutrients %>%
  ggplot(aes
         (x = month, color = lakename)) +
  geom_boxplot(aes
               (y = tn_ug)) +
  labs(x = "Month", y = "Total Nitrogen") +
```

```
  my_theme +
  theme(
    axis.title.x = element_blank(),
    legend.position = "none" ,
    panel.grid.major = element_blank(),
    axis.text.x = element_text(
      angle = 90,
      hjust = 1,
      vjust = 0.5
    )
  ) +
  scale_color_manual(values = c("goldenrod", "#556B2F")) +
  scale_x_discrete(drop = FALSE)

#cowplot
combined_plot <- plot_grid(
  temp_month,
  tp_month,
  tn_month,
  nrow = 1,
  align = "h",
  axis = "b"
)
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

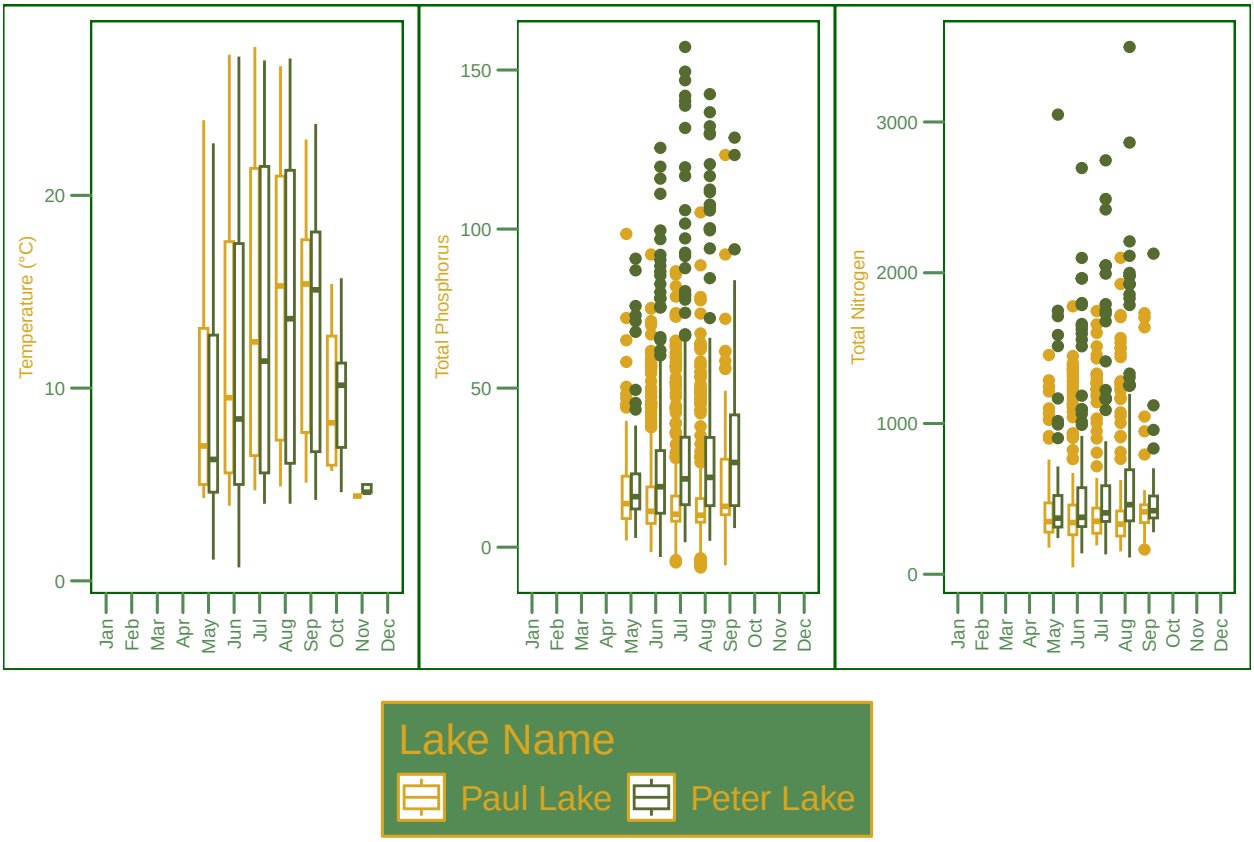
```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
legend <- get_legend(temp_month +
                       my_theme +
                       guides(color =
                              guide_legend(nrow = 1)) +
                       theme(legend.position = "right"))
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning in get_plot_component(plot, "guide-box"): Multiple components found;
## returning the first one. To return all, use `return_all = TRUE`.
```

```
#If I use anything other than right for legend position, it looks strange.
combined_plot <- plot_grid(combined_plot,
                           legend,
                           ncol = 1,
                           rel_heights = c(1, .3))
print(combined_plot)
```



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

> Answer: Peter Lake has higher amounts of phosphorus and nitrogen. Temperature goes up across both lakes in the summer.

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)

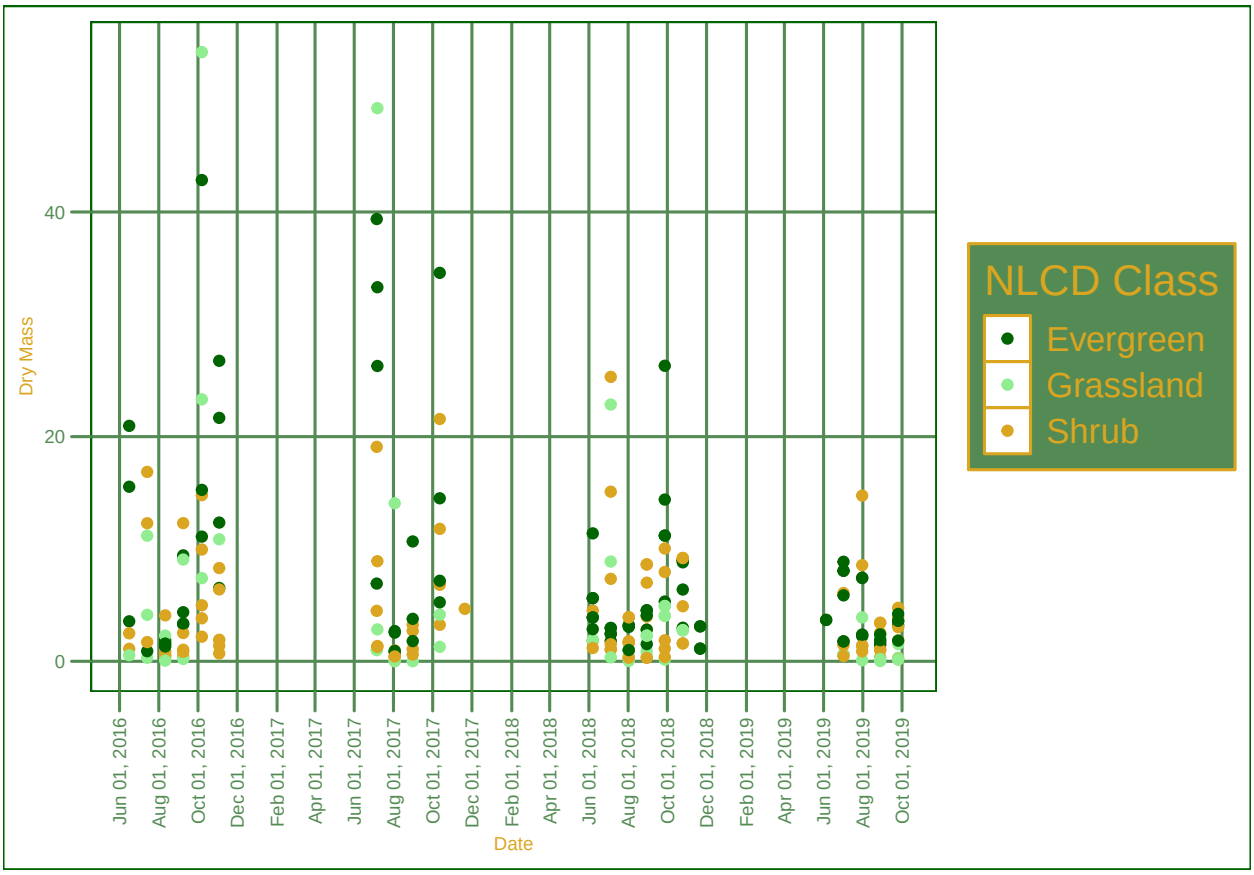
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

#Filter for Needles
df_niwot_needle <- df_niwot_litter %>%
  filter(functionalGroup == "Needles")

#Separate by Color
niwot_needle_plot <- df_niwot_needle %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  scale_color_manual(
    values = c("darkgreen", "lightgreen", "goldenrod"),
    labels = c("Evergreen", "Grassland", "Shrub")
  ) +
```

```
  my_theme +
  theme(axis.text.x = element_text(
    angle = 90,
    hjust = 1,
    vjust = 0.5
  )) +
  labs(x = "Date", y = "Dry Mass", color = "NLCD Class") +
  scale_x_date(date_labels = "%b %d, %Y",
               breaks = scales::date_breaks("2 month"))
#Show Plot
niwot_needle_plot
```

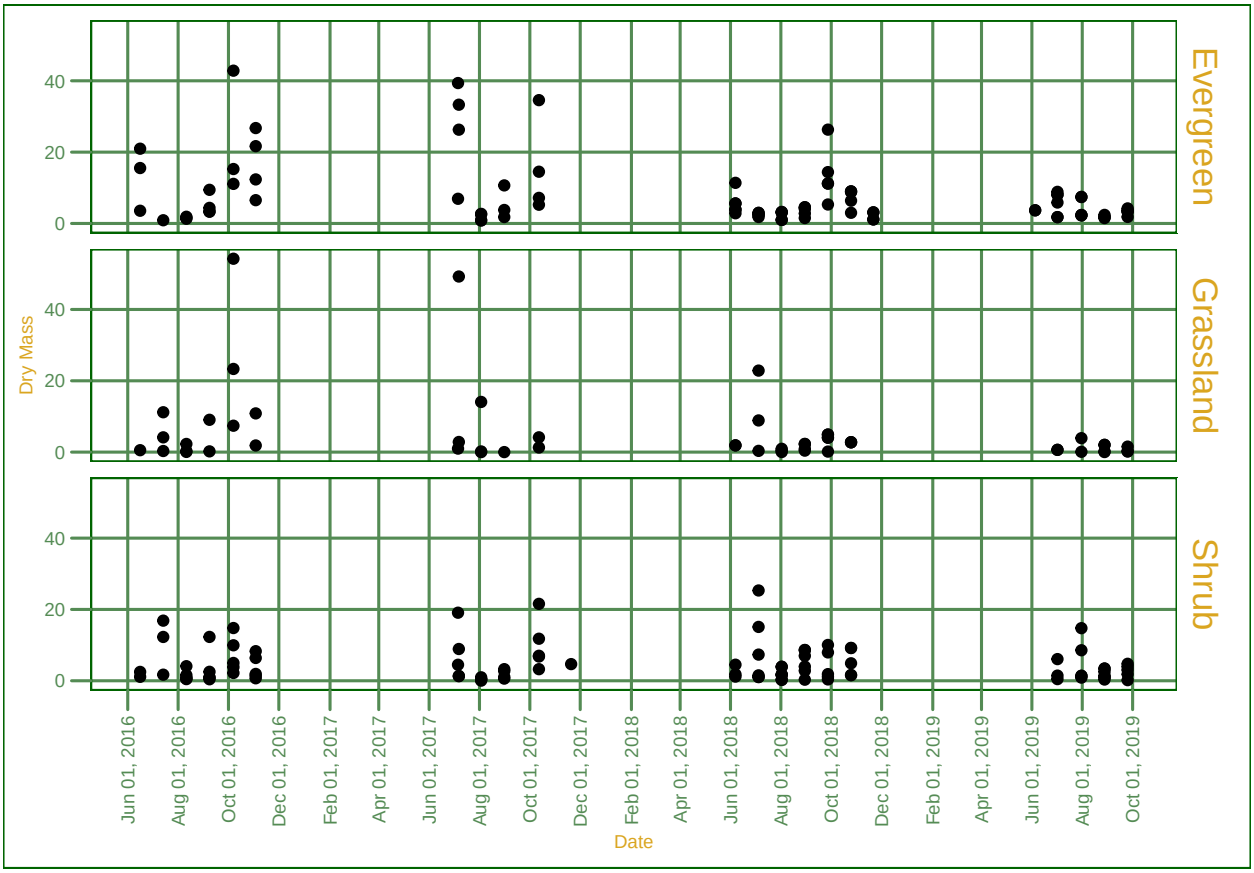



```
#7

#Separate by facet
niwot_needle_facet_plot <- df_niwot_needle %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  geom_point() +
  my_theme +
  theme(axis.text.x = element_text(
    angle = 90,
    hjust = 1,
    vjust = 0.5
  )) +
  labs(x = "Date", y = "Dry Mass") +
```

```
  scale_x_date(date_labels = "%b %d, %Y",
               breaks = scales::date_breaks("2 months")) +
  facet_grid(vars(nlcdClass), labeller = as_labeller(
    c(
      "evergreenForest" = "Evergreen",
      "grasslandHerbaceous" = "Grassland",
      "shrubScrub" = "Shrub"
    )
  ))

#Show plot
niwot_needle_facet_plot
```



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

> Answer: I beleive 7 was more effective because 6 covered up some of the other dots and was overwhelming to look at. Meanwhile, since 7 seperated the three NCLD classes, it made it easier to see each dot and compare it with the other classes.