Assignment 5: Data Visualization

Julia Kagiliery

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
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 1.9.4     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(here)
```

```
## here() starts at /Users/juliakagiliery/Library/Mobile Documents/com~apple~CloudDocs/GitHub Links/EDA
```

```
library(cowplot)
```

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

```
print(getwd())
```

```
## [1] "/Users/juliakagiliery/Library/Mobile Documents/com~apple~CloudDocs/GitHub Links/EDAClas2025"
```

```
NTL_LTER_Lake_Chemistry <- read.csv(here("Data/Processed/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients

NEON_NIWO_Litter <- read.csv(here("Data/Processed/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv

#2

NEON_NIWO_Litter <- NEON_NIWO_Litter |> #make dates be read as dates
  mutate(collectDate = ymd(collectDate))

NTL_LTER_Lake_Chemistry <- NTL_LTER_Lake_Chemistry |>
  mutate(sampledate = ymd(sampledate))

class(NTL_LTER_Lake_Chemistry$sampledate) #confirm date works
```

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

```
class(NEON_NIWO_Litter$collectDate)
```

```
## [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

library(ggridges)
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(RColorBrewer)
library(colormap)
library(ggthemes)
```

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

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

```r
#I want a barbie theme, all different texts are various colors of pink or grey, messed with size and li
pink_theme <- theme_base() +
  theme(
    line = element_line(color = 'hotpink', linewidth = 1.5),
    legend.title = element_text(color = 'deeppink', face = "italic"),
    legend.text = element_text(color = 'darkgrey'),
    plot.background = element_rect(color = 'pink'),
    panel.grid.major = element_line(color = 'lightpink', linewidth = .5),
    panel.grid.minor = element_line(color = 'lightpink', linewidth = 0.25),
    axis.title.x = element_text(color = 'deeppink', size = 14),
    axis.title.y = element_text(color = 'deeppink', size = 14),
    plot.title = element_text(
      color = 'deeppink',
      face = "bold"
    ),
    axis.text.x = element_text(color = 'darkgrey'),
    axis.text.y = element_text(color = 'darkgrey'),
    strip.text = element_text(color = 'darkgrey'),
  )

# default
theme_set(pink_theme)

#could not figure out how to add this above but this function just means that when I use colors to indi
pink_palette <- c("#FFC0CB", "#FF69B4", "#FF1493", "#DB7093", "#C71585")

scale_color_pink <- function() {
  scale_color_manual(values = pink_palette)
}
```

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()`).

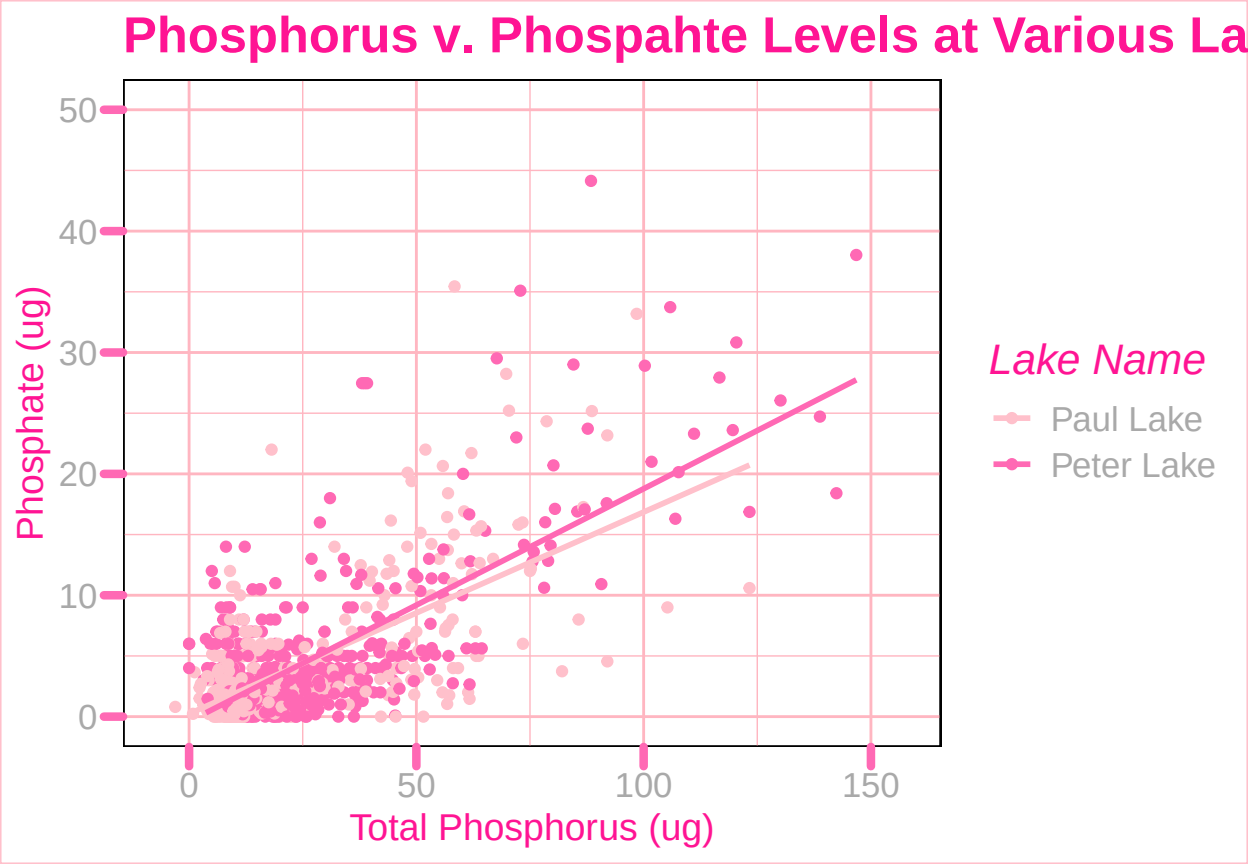
```
#4
NTL_LTER_Lake_Chemistry |>
  ggplot(aes(
    x = tp_ug, #phosporous
    y = po4, #phosphate
    color = as.factor(lakename) #this is what creates a legend because it gives the colors meaning
  )) +
  geom_point() +
  scale_color_pink() + #make the scatter plots pink
  ylim(0, 50) +
  labs(
    x = "Total Phosphorus (ug)",
    y = "Phosphate (ug)",
    color = "Lake Name",
    title = "Phosphorus v. Phospahte Levels at Various Lakes"
  ) +
  geom_smooth(method = lm, se = FALSE)
```

```
## `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()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_smooth()`).
```



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`.

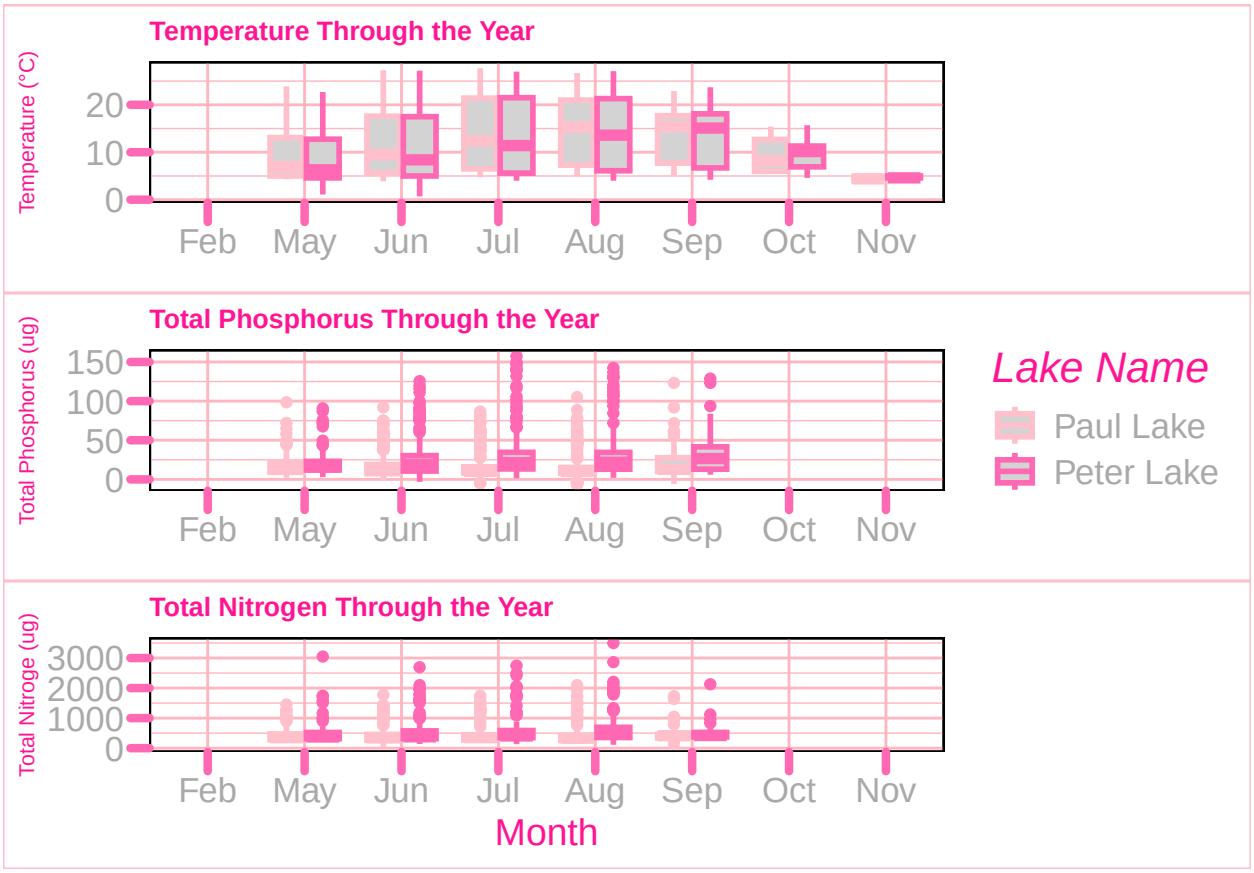
```
#5
TempPlot <- NTL_LTER_Lake_Chemistry |>
  ggplot(aes(y = temperature_C, color = as.factor(lakename))) +
  geom_boxplot(aes(x = factor(
    month, levels = 1:12, labels = month.abb #this is what tunrs it from 1 to Jan
  )), fill = "lightgray", size = 1) +  # fill in the bars bc it is easier to read in the pink hell i cr
  scale_color_pink() +
  labs(
    x = element_blank(), # only want one x label
    y = "Temperature (°C)",
    color = "Lake Name",
    title = "Temperature Through the Year"
  ) +
  theme(legend.position = "none", # only want one legend
        plot.title = element_text(size = 10),    # I had to pess with these because it rendered weird in
        axis.title.y = element_text(size = 8)
```

```
  )

#same as above
PhosPlot <- NTL_LTER_Lake_Chemistry |>
  ggplot(aes(y = tp_ug, color = as.factor(lakename))) +
  geom_boxplot(aes(x = factor(
    month, levels = 1:12, labels = month.abb
  )), fill = "lightgray", size = 1) +
  scale_color_pink() +
  labs(
    x = element_blank(),
    y = "Total Phosphorus (ug)",
    color = "Lake Name",
    title = "Total Phosphorus Through the Year"
  ) +
  theme(legend.position = "right",
        plot.title = element_text(size = 10),
        axis.title.y = element_text(size = 8)
  )

#same as above
NitrogenPlot <- NTL_LTER_Lake_Chemistry |>
  ggplot(aes(y = tn_ug, color = as.factor(lakename))) +
  geom_boxplot(aes(x = factor(
    month, levels = 1:12, labels = month.abb
  )), fill = "lightgray", size = 1) +
  scale_color_pink() +
  labs(
    x = "Month",
    y = "Total Nitroge (ug)",
    color = "Lake Name",
    title = "Total Nitrogen Through the Year"
  ) +
  theme(legend.position = "none",
        plot.title = element_text(size = 10),
        axis.title.y = element_text(size = 8)
  )

# three plots, stacked and cenntered
plot_grid(TempPlot, PhosPlot, NitrogenPlot,
          ncol = 1, nrow = 3, align = 'v')
```

```
## 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()`).
```



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

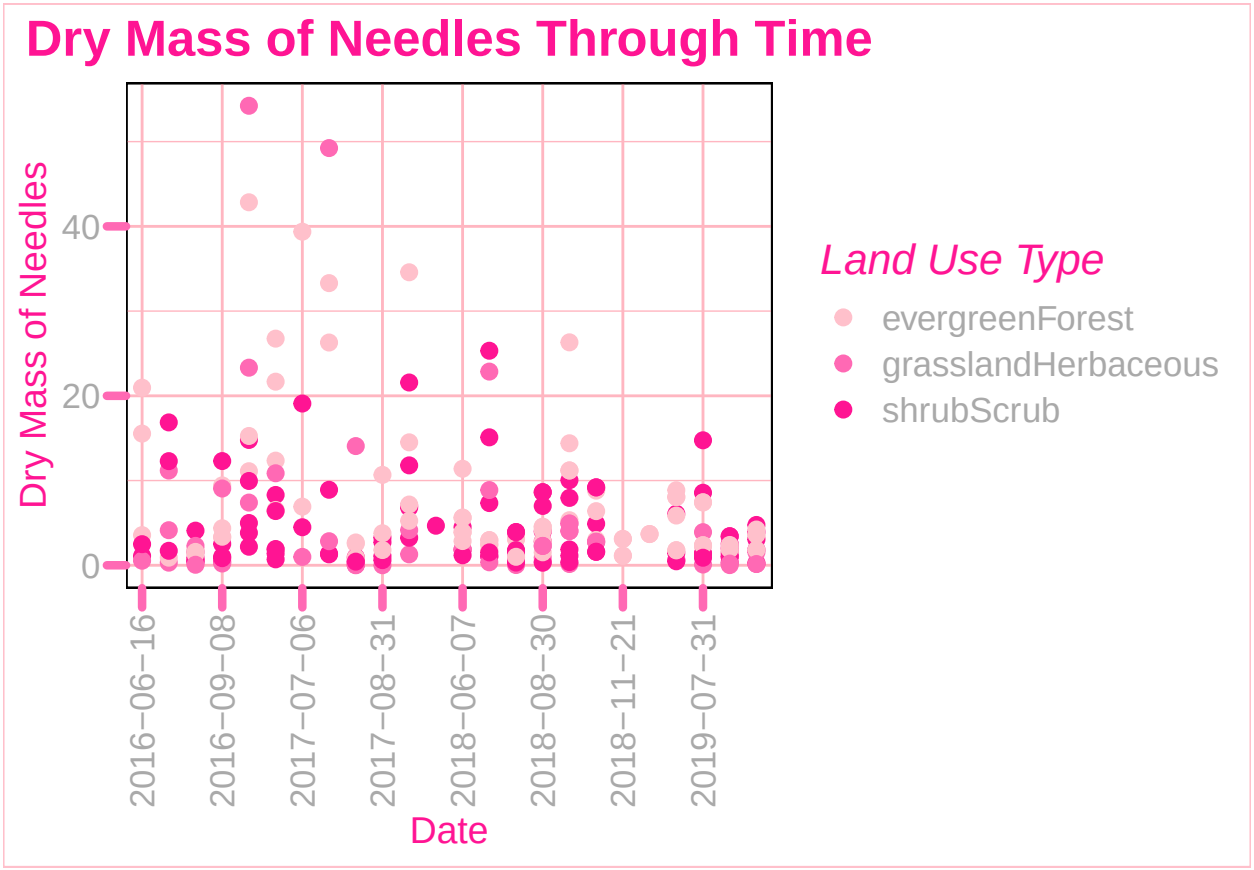
> Answer: These variables all seem to be seasonal such that they peak in the summer (in higher temepratures) and are lower in the fall and spring. There is no data in the winter. Further more, the temperature is very similar between lakes but the total phosphourus and nitrogen seem higher (or at least more variable) in Peter Lake 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.

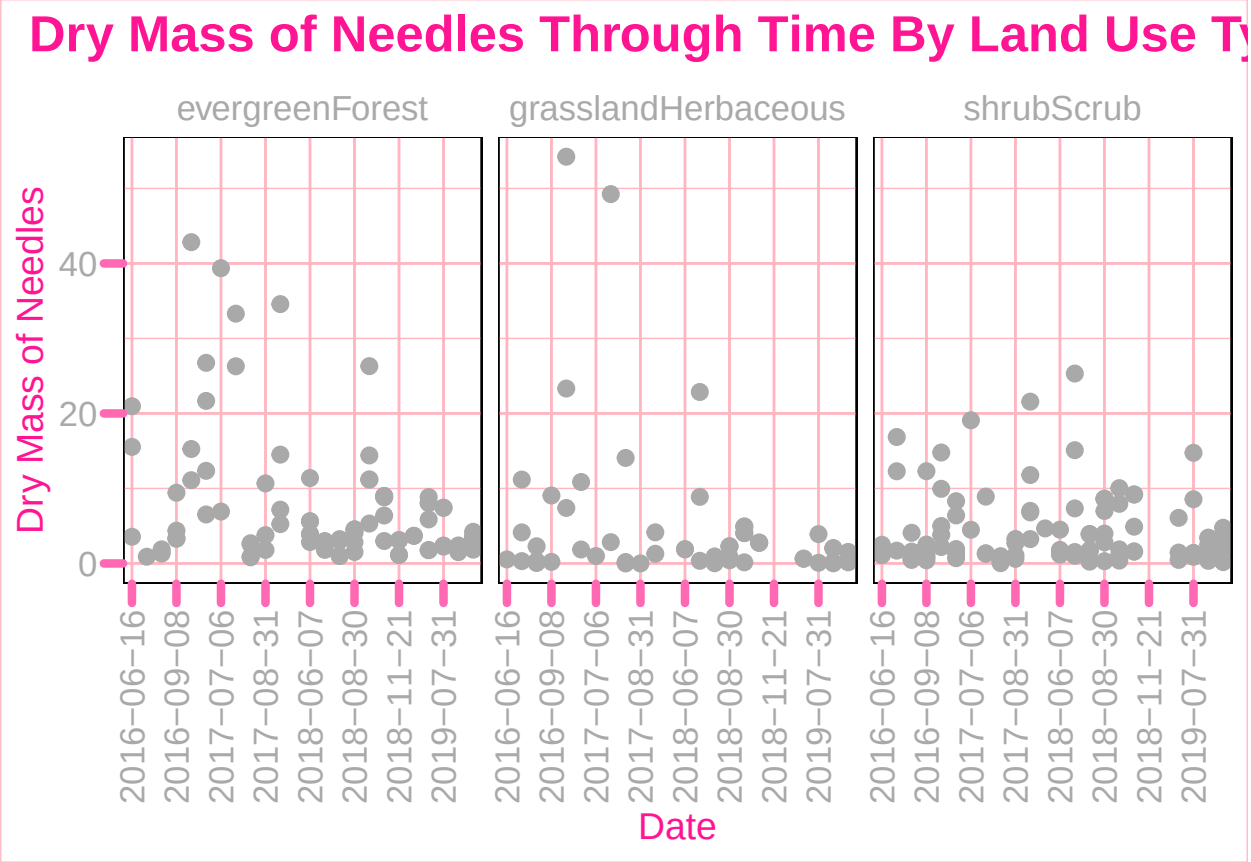
```
#6
NEON_NIWO_LitterNeddles <- NEON_NIWO_Litter |>
  filter(functionalGroup == "Needles") #needles only per instructions

NEON_NIWO_LitterNeddles |>
  ggplot(aes(
    y = dryMass,
    x = as.factor(collectDate),
    color = nlcdClass
  )) +
  scale_color_pink() +
  geom_point(size = 2.5) +
```

```
    scale_x_discrete(breaks = function(x) x[seq(1, length(x), by = 3)]) + #label every 3rd date to make
  labs(
    x = "Date",
    y = "Dry Mass of Needles",
    color = "Land Use Type",
    title = "Dry Mass of Needles Through Time"
  ) +
  theme(
    axis.text.x = element_text( #couldn't read it in the render, make it smaller and rotate it
      angle = 90,
      vjust = 0.5,
      hjust = 0.5
    ),
    plot.title = element_text(hjust = 0.5)
  )
```




```
#7
NEON_NIWO_LitterNeddles |>
  ggplot(aes(y = dryMass, x = as.factor(collectDate))) +
  geom_point(color = "darkgrey", size = 2.5) +
  facet_wrap( ~ nlcdClass) +
  labs(x = "Date", y = "Dry Mass of Needles", title = "Dry Mass of Needles Through Time By Land Use Type
  scale_x_discrete(breaks = function(x) x[seq(1, length(x), by = 3)]) + #label every 3rd date to make i
  theme(
    axis.text.x = element_text(
```

```
    angle = 90,
    vjust = 0.5,
    hjust = 0.5
  ),
  plot.title = element_text(hjust = 0.5)
)
```



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

Answer: I think plot 7 is more effective/ There are so many points, that faceting by land use type allows for you to more acurately see trends in the data without having to process the color differences in plot 6. The frequency of somapling and number of points is pretty high, I do think that the plots could certainly be made more clear by condensive that data (such as reporting the mean for each day rather than every value) or generating a box plot.Furthermore, We could adjust the number of time periods reported (maybe take the average of two months which would have the number of observations).