Assignment 5: Data Visualization

Emma Wellbaum

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 Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) and the processed data file for the Niwot Ridge litter dataset.

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

```
#1

# get working directory
getwd()
```

```
## [1] "C:/Users/emmaw/Documents/ENV872/Environmental_Data_Analytics_2021"
```

```
# install.packages(tidyverse)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# install.packages(cowplot)
library(cowplot)
```

```
# Upload processed datasets
PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
           stringsAsFactors = TRUE)
PeterPaul.chem.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
Litter <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2

# Change the format of date columns to the date format
PeterPaul.chem.nutrients$sampledate <- as.Date(
  PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d")
PeterPaul.chem.nutrients.gathered$sampledate <- as.Date(
  PeterPaul.chem.nutrients.gathered$sampledate, format = "%Y-%m-%d")
Litter$collectDate <- as.Date(Litter$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

```
# Create custom theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text= element_text(color = "black"),
        legend.position = "right",
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 12))
# Set as default
theme_set(mytheme)
```

Create graphs

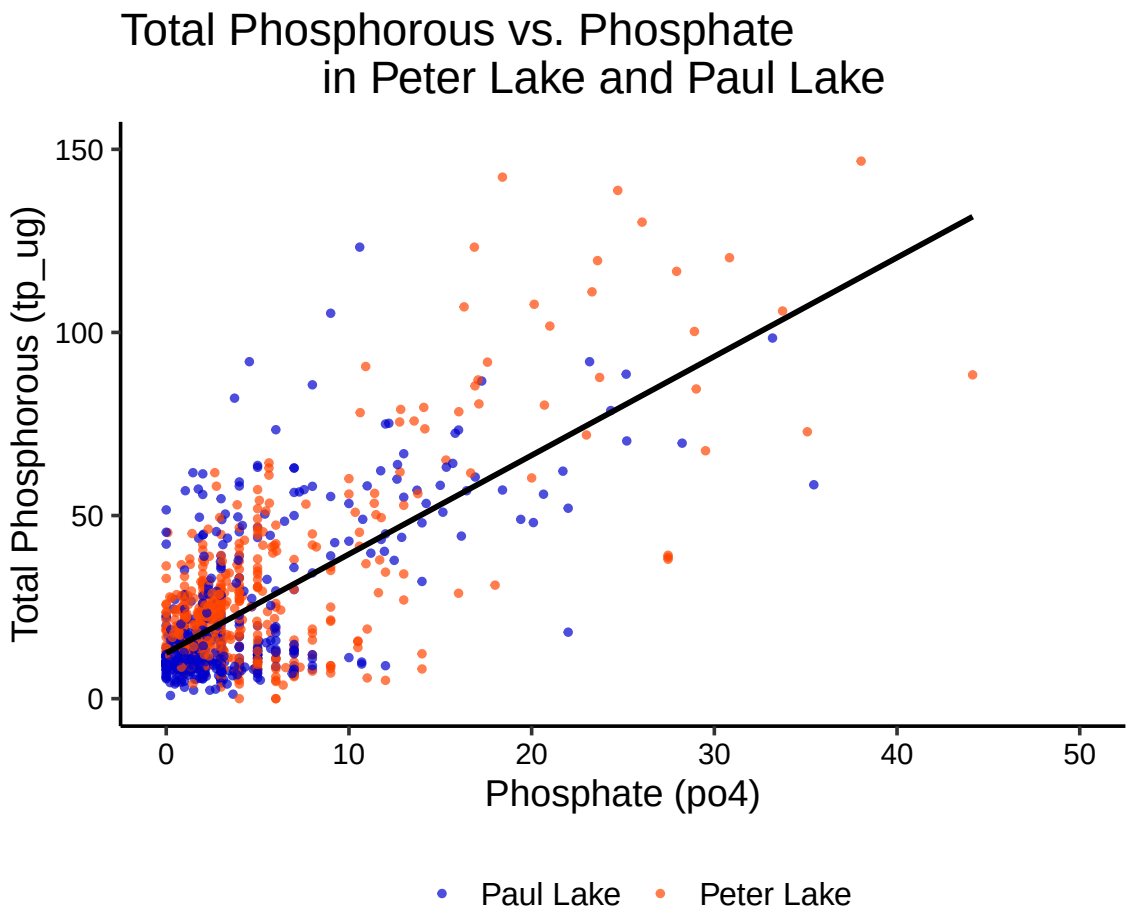
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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.

```
nutrient.plot <-
  # Plot total phosphorous by phosphate with different colors for each lake
  ggplot(PeterPaul.chem.nutrients, aes(x=po4, y=tp_ug, color=lakename)) +
  geom_point(size = 1, alpha = 0.7)+
  # Set the colors
  scale_color_manual(values = c("mediumblue", "orangered")) +
  # Adjust axes to hide extreme values
  xlim(0, 50) +
  ylim(0, 150) +
  # Add a (black) line of best fit
  geom_smooth(method = lm, se = FALSE, color = "black") +
  # Add/alter the title and axis labels
  labs(title = "Total Phosphorous vs. Phosphate
                in Peter Lake and Paul Lake",
       y = "Total Phosphorous (tp_ug)",
```

```
       x ="Phosphate (po4)",
       color = " ") +
  # Position the legend below the plot
  theme(legend.position = "bottom")
print(nutrient.plot)
```

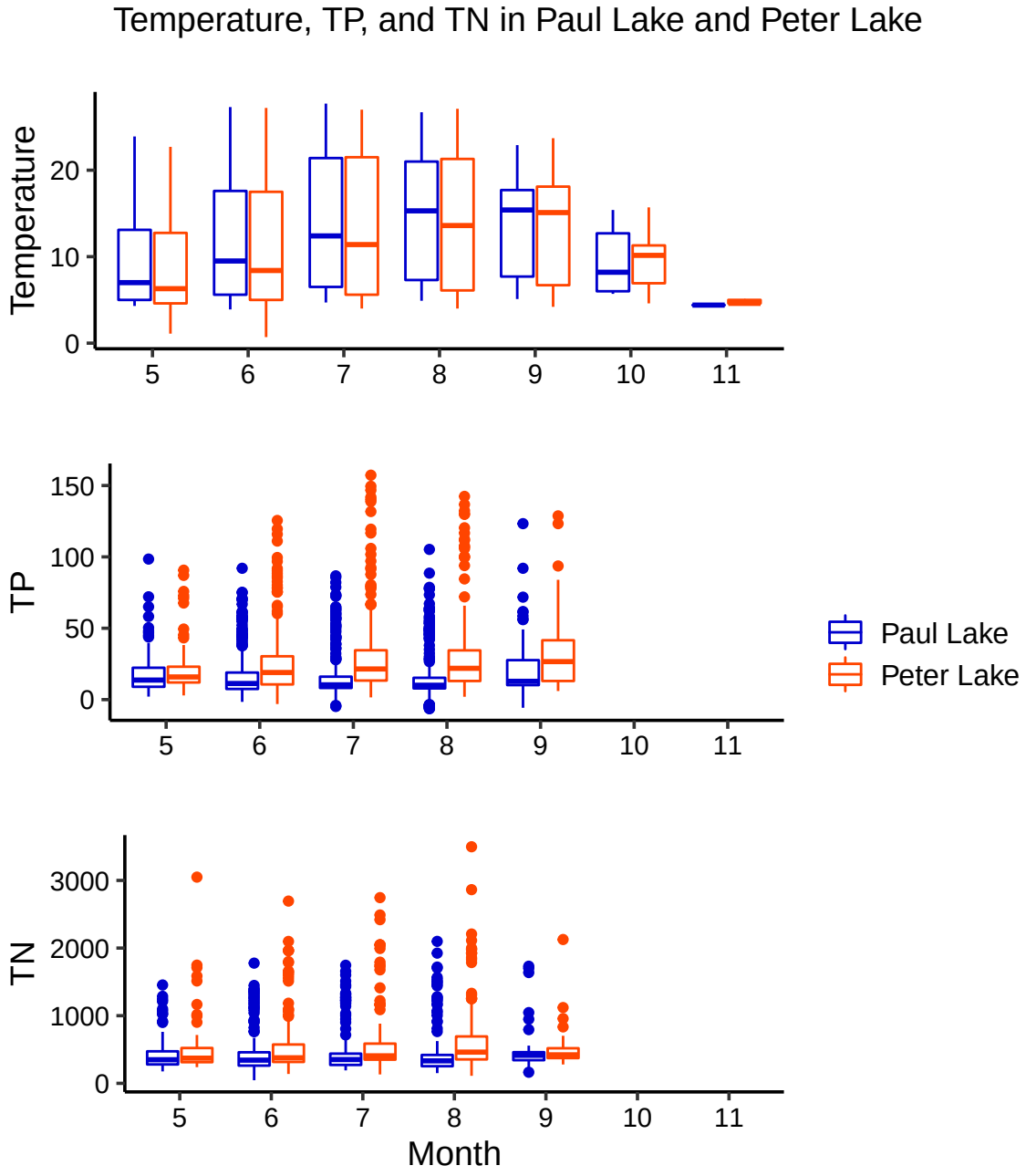
```
## `geom_smooth()` using formula 'y ~ x'
```



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.

```
# Temperature Boxplot
temp.plot <-
  # Filter the dataset to exclude months with no data, color by lake name,
  # and set month to be a factor so that we can compare nutrient levels seasonally.
  ggplot(filter(PeterPaul.chem.nutrients, month !=2),
         aes(x = as.factor(month), y=temperature_C, color=lakename)) +
  geom_boxplot() +
  # Set the colors
  scale_color_manual(values = c("mediumblue", "orangered")) +
  # Label the y axis and hide the x axis label (only needs to be labeled once in cowplot)
  labs(x = "", y = "Temperature") +
  # Hide the legend (cowplot only needs one legend)
```

```
  theme(legend.position = "none")

# TP Boxplot
TP.plot <-
  # Repeat filtering process with TP
  ggplot(filter(PeterPaul.chem.nutrients, month !=2),
         aes(x=as.factor(month), y=tp_ug, color=lakename)) +
  geom_boxplot() +
  # Set the colors
  scale_color_manual(values = c("mediumblue", "orangered")) +
  # Label the y axis only
  labs(x = "", y = "TP") +
  theme(legend.position = "none")

# TN Boxplot
TN.plot <-
  # Repeat fitlering process with TN
  ggplot(filter(PeterPaul.chem.nutrients, month !=2),
         aes(x=as.factor(month), y=tn_ug, color=lakename)) +
  geom_boxplot() +
  # Set the colors
  scale_color_manual(values = c("mediumblue", "orangered")) +
  # Label the x axis and legend (to be used in the cowplot)
  labs(color = "", x = "Month", y = "TN") +
  # Position legend to the right of the plot
  theme(legend.position = "right")


# Combine the temperature, TP, and TN plots into a single cowplot.
combo.plot <-
  plot_grid(temp.plot,
            TP.plot,
            TN.plot + theme(legend.position = "none"), # hide TN.plot legend
            nrow = 3, # three rows
            align = 'h', # aligned horizontally
            rel_heights = c(1, 1, 1)) # all with the same relative height

# Extract the legend from TN.plot and add it combo.plot
legend <- get_legend(TN.plot + theme(legend.box.margin = margin(0, 0, 0, 0)))
combo.plot <-
  plot_grid(combo.plot, legend, rel_widths=c(1, 0.3))

# Create cowplot title and add it to combo.plot
title <- ggdraw() +
  draw_label("Temperature, TP, and TN in Paul Lake and Peter Lake")
combo.plot <-
  plot_grid(title, combo.plot, ncol=1, rel_heights=c(0.1, 1))
# print combo.plot
print(combo.plot)
```

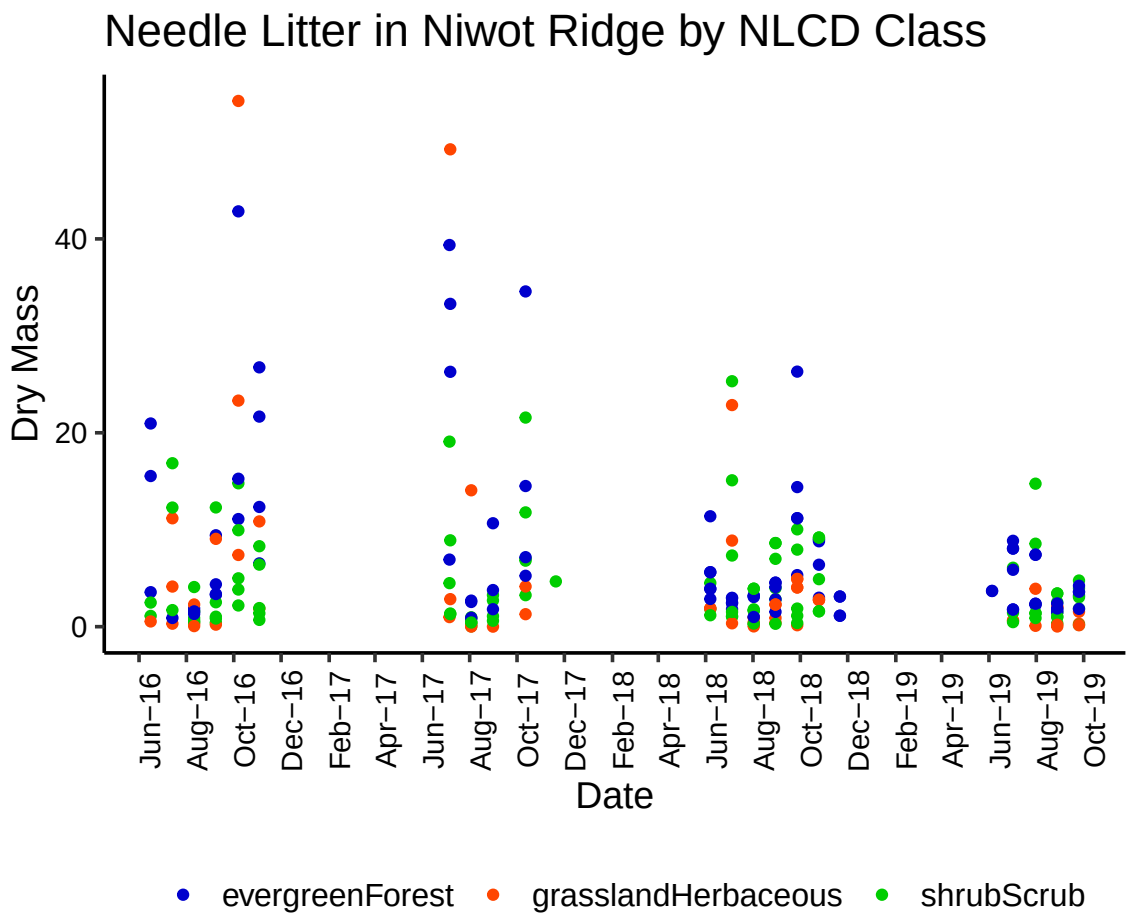


Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: We can observe a correlation between temperature increase and an increase in total phosphorous and total nitrogen (this is expected). It also looks like Peter Lake has (much?) higher concentrations of both phosphorous and nitrogen, which may point to water quality issues. It would be interested to see if total phosphorous and nitrogen levels were collected for Peter and Paul Lake in the Fall and Winter months as well as the Spring and Summer months.

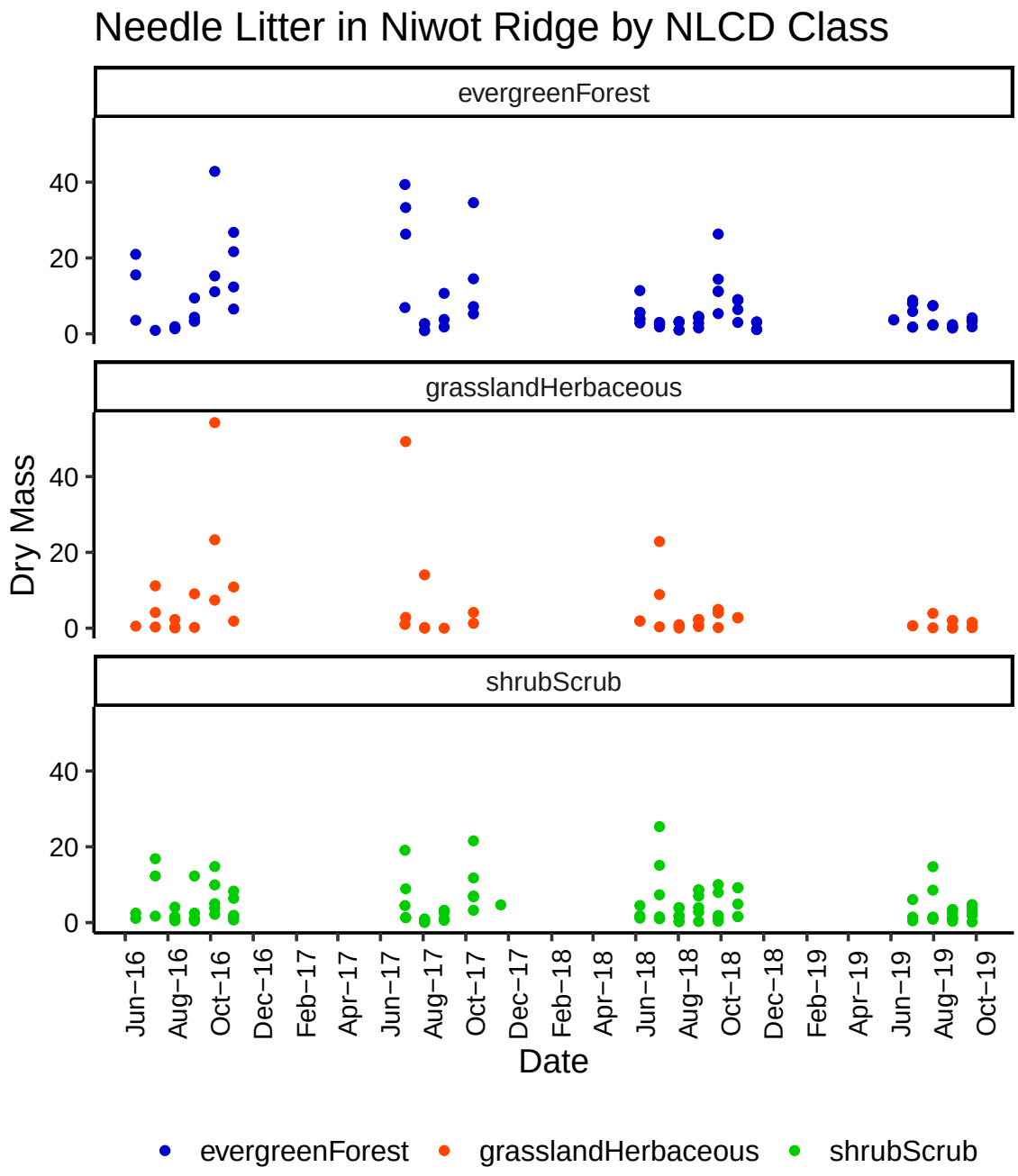
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

```r
#6
Needle.Color.Plot <-
  # Filter the dataset to values where the functional group equals "Needles"
  ggplot(filter(Litter, functionalGroup == "Needles"),
         aes(collectDate, y=dryMass, color=nlcdClass)) +
  geom_point() +
  # Position the legend below the plot
  theme(legend.position = "bottom") +
  # Set the colors
  scale_color_manual(values = c("mediumblue", "orangered", "green3")) +
  # Scale & reformat the x axis tick marks and labels
  scale_x_date(date_breaks = "2 months", date_labels = "%b-%y") +
  # Rotate the x axis tick mark labels so they are vertical
  theme(axis.text.x = element_text(angle=90)) +
  # Label axes and legend and add title
  labs(color = "", x = "Date", y = "Dry Mass",
       title = "Needle Litter in Niwot Ridge by NLCD Class")
print(Needle.Color.Plot)
```



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

```r
#7
Needle.Facet.Plot <-
  ggplot(filter(Litter, functionalGroup == "Needles"),
         aes(x=collectDate, y=dryMass, color=nlcdClass)) +
  geom_point() +
  # Separate the plot into a 3-row facet
  facet_wrap(vars(nlcdClass), nrow = 3) +
  # Position the legend below the plot
  theme(legend.position = "bottom") +
  # Set the colors
  scale_color_manual(values = c("mediumblue", "orangered", "green3")) +
  # Scale & reformat the x axis tick marks and labels
  scale_x_date(date_breaks = "2 months",
               date_labels = "%b-%y") +
  # Rotate the x axis tick mark labels so they are vertical
  theme(axis.text.x = element_text(angle=90)) +
  # Label axes and legend and add title
  labs(color = "", x = "Date", y = "Dry Mass",
       title = "Needle Litter in Niwot Ridge by NLCD Class")
print(Needle.Facet.Plot)
```



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

Answer: The faceted plot created for question 7 is more effective. It's much easier to read. It's difficult to tell what the spread and distribution of each NLCD class is when they are separated by color alone. It's much easier to interpret the annual changes within and between the classes in the faceted plot.