Assignment 5: Data Visualization

Camber Vincent

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
#loading in all packages
library(tidyverse)
library(lubridate)
library(here)
library(cowplot)

getwd() #verify home directory
```

```
## [1] "/Users/cambervincent/EDA_Fall_2024/Assignments"
```

```
#read in csv files
base_path<-"/Users/cambervincent/EDA_Fall_2024/Data/Processed_KEY/"
lake_chem_nutrients<-read.csv(file.path(base_path,
                                "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))
litter<-read.csv(file.path(base_path,
                 "NEON_NIWO_Litter_mass_trap_Processed.csv"))

#2
#checking class of dates in data samples
class(lake_chem_nutrients$sampledate)
```

```
## [1] "character"
```

```
class(litter$collectDate)
```

```
## [1] "character"
```

```
#changing format to date
lake_chem_nutrients$sampledate<-as.Date(lake_chem_nutrients$sampledate)
litter$collectDate<-as.Date(litter$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

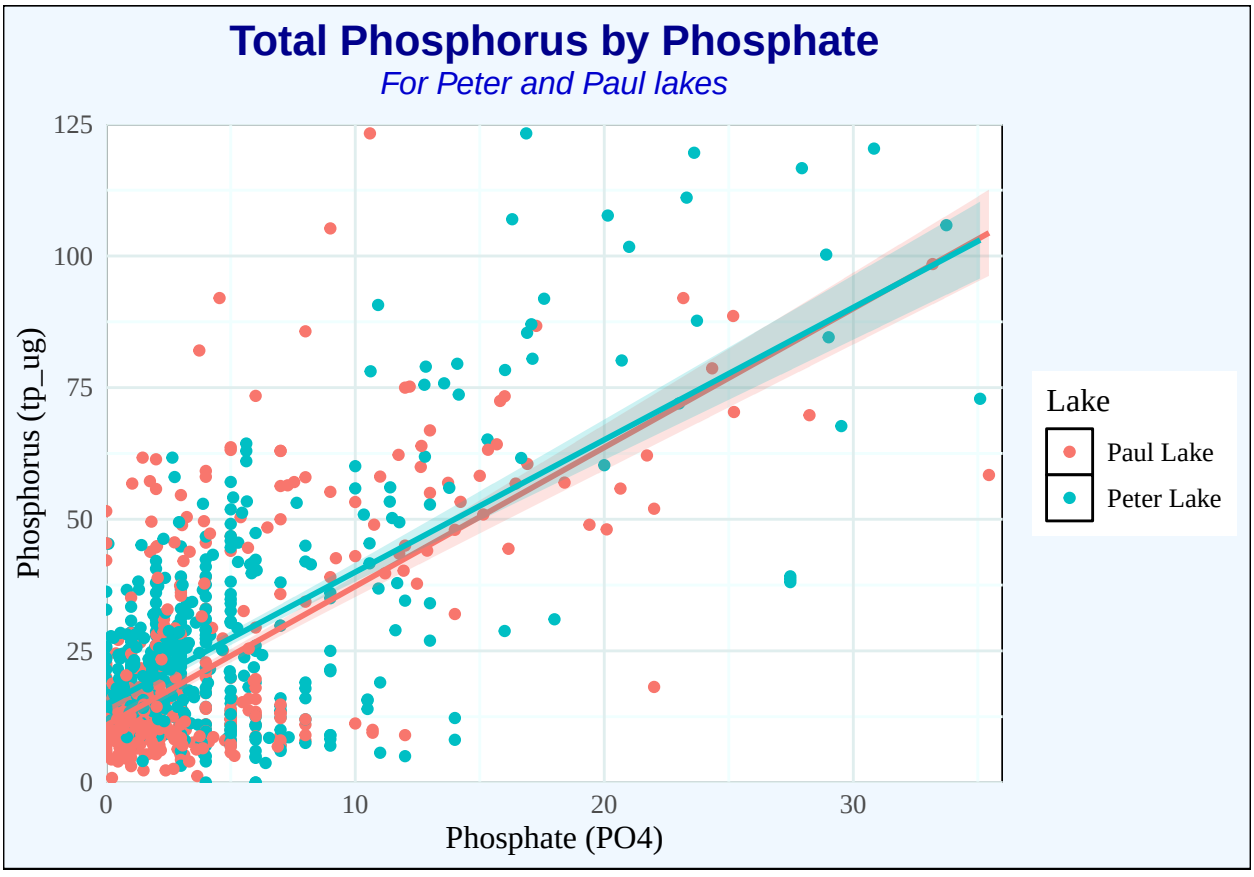
```
#3
theme_standard<-theme(

  text=element_text(family="Times",size=12), #setting base font to Times New Roman
  plot.title=element_text(family="Helvetica",face="bold",size=16,  #title text theme
                          color="darkblue",
                          margin=margin(b=3)), #added margin for visual clarity
  plot.subtitle=element_text(family="Helvetica",face="italic",size=12, #subtitle text theme
                             color="blue3",
                             margin=margin(b=10)), #added margin for visual clarity

  plot.background=element_rect(fill="aliceblue"), #background set to light blue
  panel.background=element_rect(fill="white"), #graph background set to white

  panel.grid.major=element_line(color="azure2"), #recolor gridlines
  panel.grid.minor=element_line(color="azure1"),
  axis.ticks=element_blank() #turn off ticks

)

theme_set(theme_standard) #set custom theme as the base theme for all graphs
```

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()`).

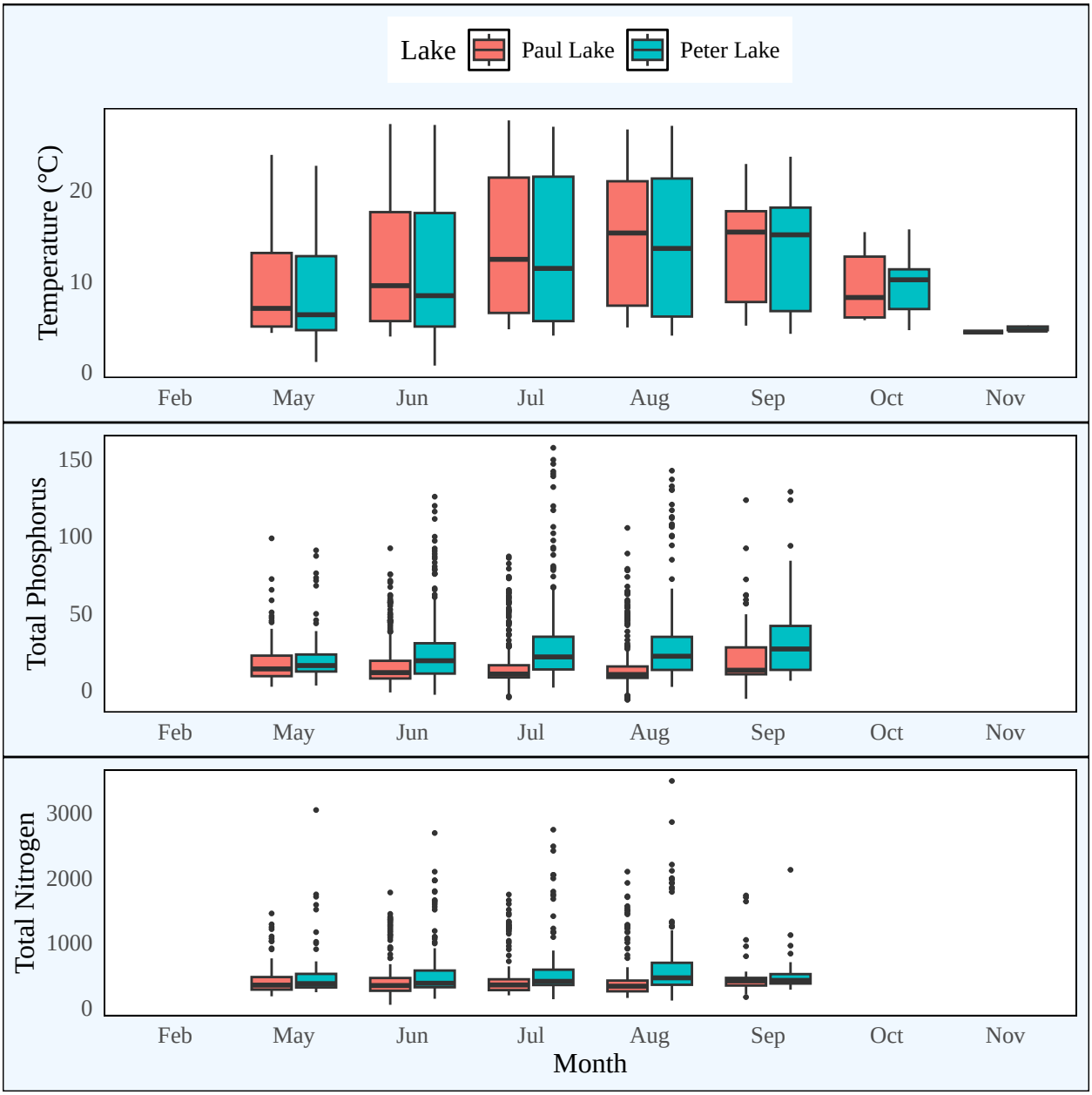
```
#4
ggplot(lake_chem_nutrients,aes(x=po4,y=tp_ug,color=lakename))+
  geom_point()+
  geom_smooth(aes(fill=lakename),#set color of confidence range to the same as the line
              method="lm",
              alpha=0.2, #made the confidence range semi-transparent
              show.legend=FALSE)+ #turned off additional legend created
  scale_x_continuous(expand=c(0,0), #center x-axis at 0-x
                     limits=c(0,36))+ #set x-axis limits for visual clarity
  scale_y_continuous(expand=c(0,0), #center y-axis at 0-x
                     limits=c(0,125))+ #set y-axis limit for visual clarity
  labs(title="Total Phosphorus by Phosphate",
       subtitle="For Peter and Paul lakes",
       x="Phosphate (PO4)",
       y="Phosphorus (tp_ug)",
       color="Lake")
```



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
#creating plot for temperature
p1<-ggplot(lake_chem_nutrients,aes(x=factor(month,levels=1:12,labels=month.abb),
                                   y=temperature_C,
                                   fill=lakename))+
  geom_boxplot(outlier.size=0.5)+ #reduce outlier size for visual clarity
  labs(y="Temperature (°C)",fill="Lake")+
  theme(panel.grid.major=element_blank(), #removing gridlines for clarity on boxplot
        panel.grid.minor=element_blank(),
        axis.title.x=element_blank(),legend.position="top") #removes x-axis titles and legend

#creating plot for total phosphorus
p2<-ggplot(lake_chem_nutrients,aes(x=factor(month,levels=1:12,labels=month.abb),
                                   y=tp_ug,
                                   fill=lakename))+
  geom_boxplot(outlier.size=0.5)+ #reduce outlier size for visual clarity
  labs(y="Total Phosphorus",fill="Lake")+
  theme(panel.grid.major=element_blank(), #removing gridlines for clarity on boxplot
        panel.grid.minor=element_blank(),
        axis.title.x=element_blank(),legend.position="none") #removes x-axis titles and legend

#creating plot for total nitrogen
p3<-ggplot(lake_chem_nutrients,aes(x=factor(month,levels=1:12,labels=month.abb),
                                   y=tn_ug,
                                   fill=lakename))+
  geom_boxplot(outlier.size=0.5)+ #reduce outlier size for visual clarity
  labs(x="Month",y="Total Nitrogen",fill="Lake")+
  theme(panel.grid.major=element_blank(), #removing gridlines for clarity on boxplot
        panel.grid.minor=element_blank(),
        legend.position="none") #keep x axis title here

#cowplot combined graph
combined_plot<-plot_grid(p1,p2,p3,ncol=1,
                         align="v", #align graphs vertically
                         axis="lr", #align axes left to right
                         rel_heights=c(2.5,2,2)) #set relative heights to account for legend

#display the final plot
combined_plot
```



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

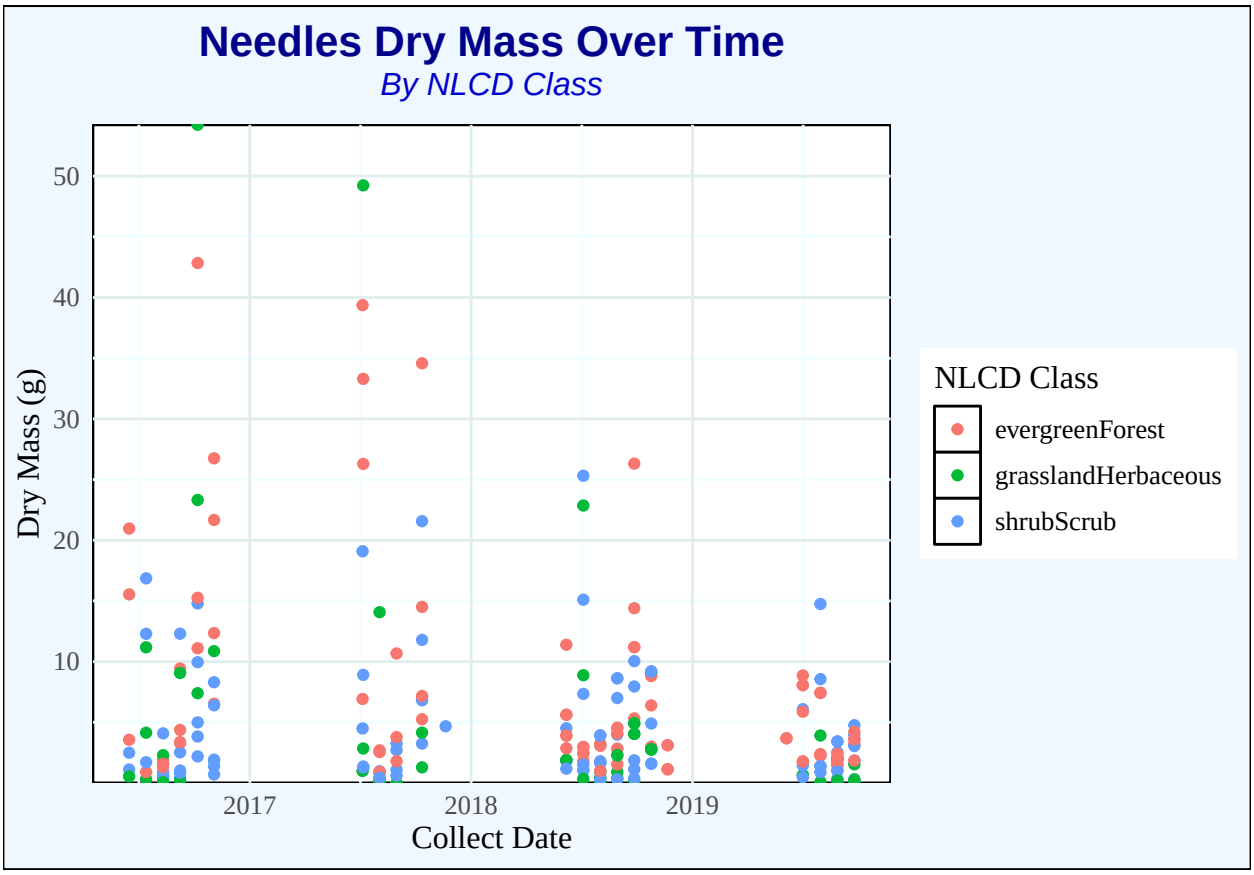
Answer: Median tempearture for Paul Lake appears to be consistently higher than median temperature for Peter lake across all months, with the exception of October and November. However, both lakes exhibit roughly the same range of temperatures ad appear to have roughly equivalent 25th and 75th percentiles. Total phosphorus shows much more variability, with significant numbers of outliers for both lakes across several months. Peter Lake has a far higher median across most months for which data is available, with the Peter Lake median total phosphorus appearing higher than the Paul Lake 75th percentile phosphorus amount for June, July, August, and September. Similarly, Peter Lake appears to have a generally higher amount of total nitrogen across all months for which data is available. The consistent range of total nitrogen across both lakes is generally smaller with far more varaiability in outliers.

6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group.

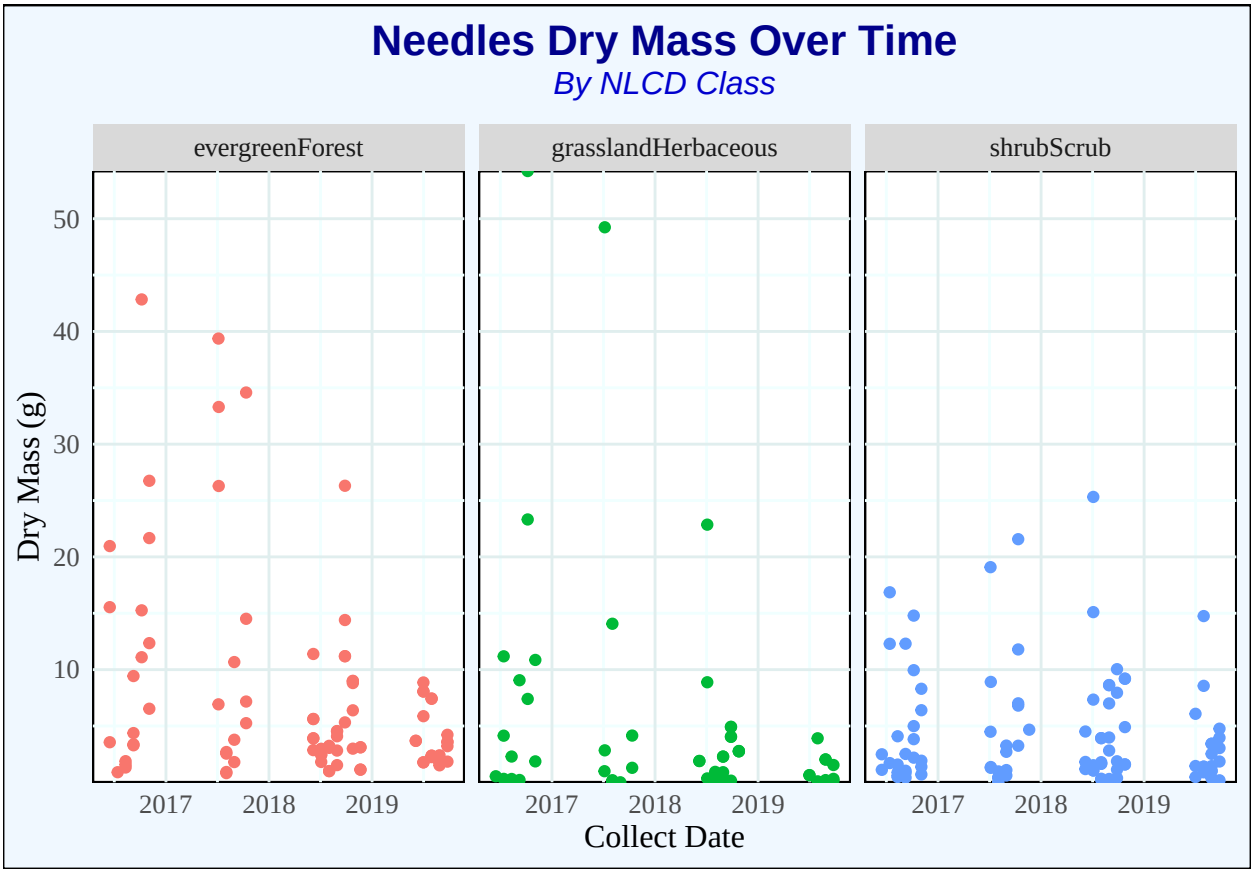
Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
needles_data<-litter%>%
  filter(functionalGroup=="Needles") #subset litter data to Needles

ggplot(needles_data,aes(x=collectDate,y=dryMass,color=nlcdClass))+ #set up graph as described
  geom_point()+
  labs(title="Needles Dry Mass Over Time",subtitle="By NLCD Class",
       x="Collect Date",y="Dry Mass (g)",color="NLCD Class")+
  scale_y_continuous(expand=c(0,0)) #added to center graph at 0-y
```




```
#7
ggplot(needles_data,aes(x=collectDate,y=dryMass,color=nlcdClass))+
  geom_point()+
  labs(title="Needles Dry Mass Over Time",subtitle="By NLCD Class",
       x="Collect Date",y="Dry Mass (g)",color="NLCD Class")+
  facet_wrap(~nlcdClass,ncol=3)+ #adding a facet wrap by NLCD Class
  theme(legend.position="none")+
  scale_y_continuous(expand=c(0,0)) #added to center graph at 0-y
```



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

Answer: I think that plot 7 with the facet wrap included is more effective for visualizing the data. The data in plot 6 is visually hard to distinguish between the different NLCD classes even with a color aspect applied. When the data is separated by NLCD class into different facets, trends are more easily visualized. For example, it is easier to pick out that needle mass in the Evergreen Forest class is consistently higher with more variability than the Grassland Herbaceous and Shrub Scrub classes. The visual clarity provided by the facet wrap makes plot 7 more effective for data visualization.