Assignment 5: Data Visualization

Rachel Landman

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., "Salk_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 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 (tidy and gathered) 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
getwd()
```

```
## [1] "/Users/rmlandman/Desktop/Data Analytics/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyverse
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------ tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.chem.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
Niwot.Litter <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")


#2
class(Niwot.Litter$collectDate)
```

```
## [1] "factor"
```

```
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")
Niwot.Litter$collectDate <-
  as.Date(Niwot.Litter$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

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

theme_set(mytheme)
```

Create graphs

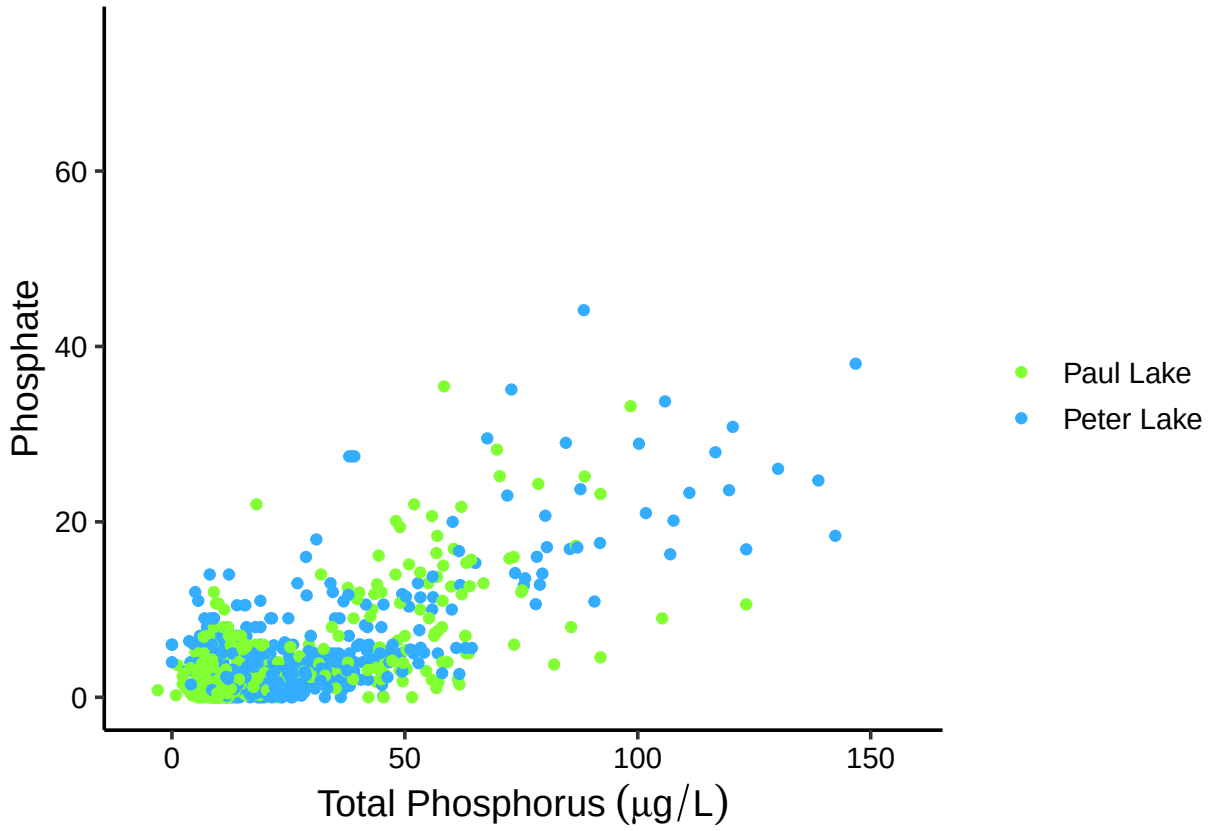
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus by phosphate, 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.

```
Phosphorus_Phosphateplot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4, color = lakename))+
  geom_point()+
  labs (x = expression("Total Phosphorus" ~ (mu*g / L)), y = "Phosphate", color = "")+
  ylim(0,75)+
  scale_color_manual(values = c("#82FF33", "#33ADFF"))+
   theme(legend.position = "right")

print(Phosphorus_Phosphateplot)
```

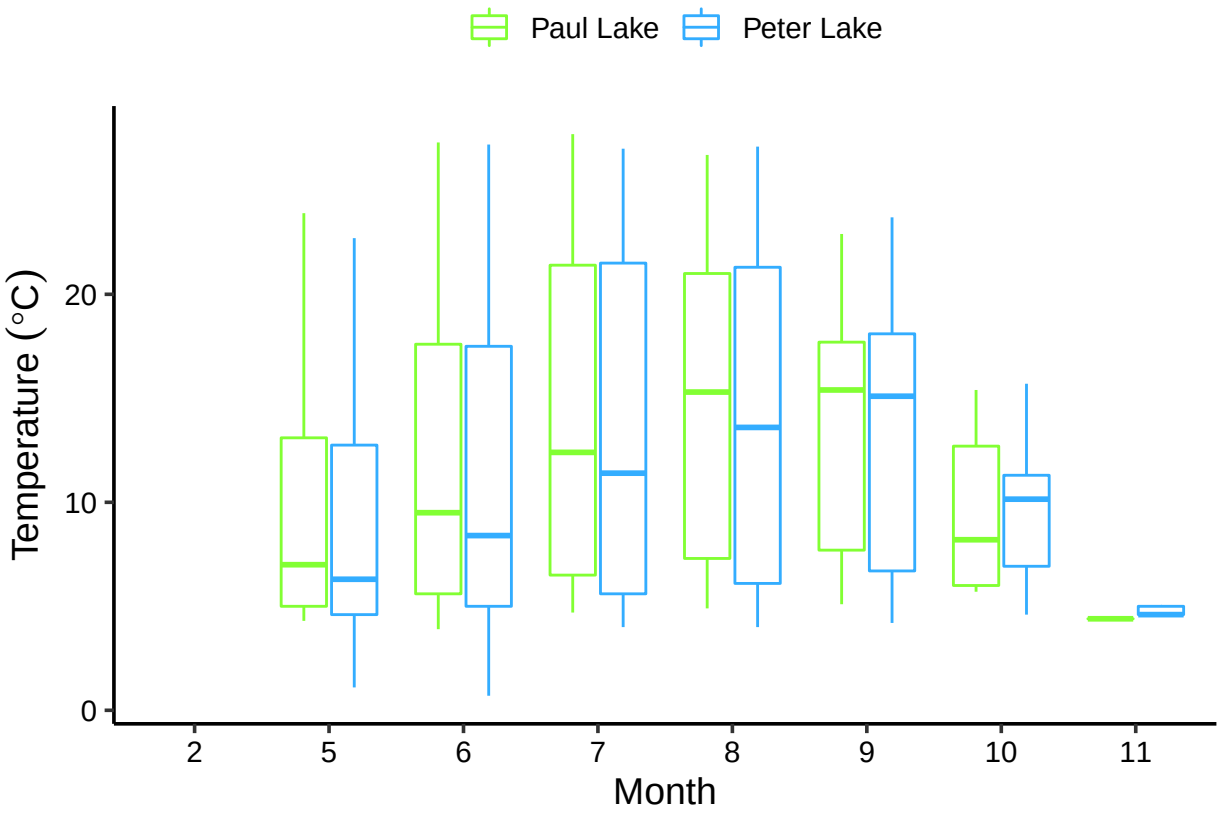
```
## Warning: Removed 21947 rows containing missing values (geom_point).
```



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.

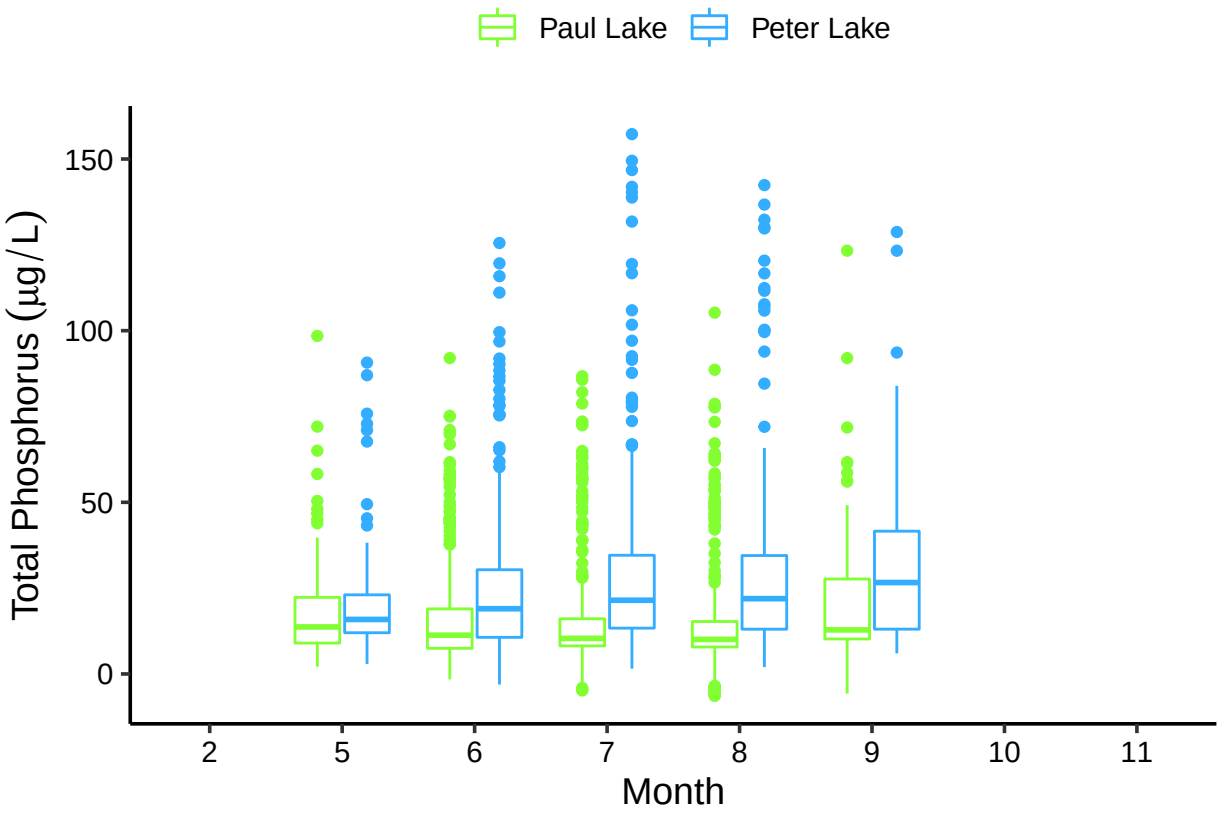
```
Temp_boxplot <-
  ggplot(PeterPaul.chem.nutrients, aes(y = temperature_C, x = as.factor(month), color = lakename))+
  geom_boxplot()+
  labs(x = "Month", y = expression("Temperature " ( degree*C)), color = "")+
   scale_color_manual(values = c("#82FF33", "#33ADFF"))
print(Temp_boxplot)
```

```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```



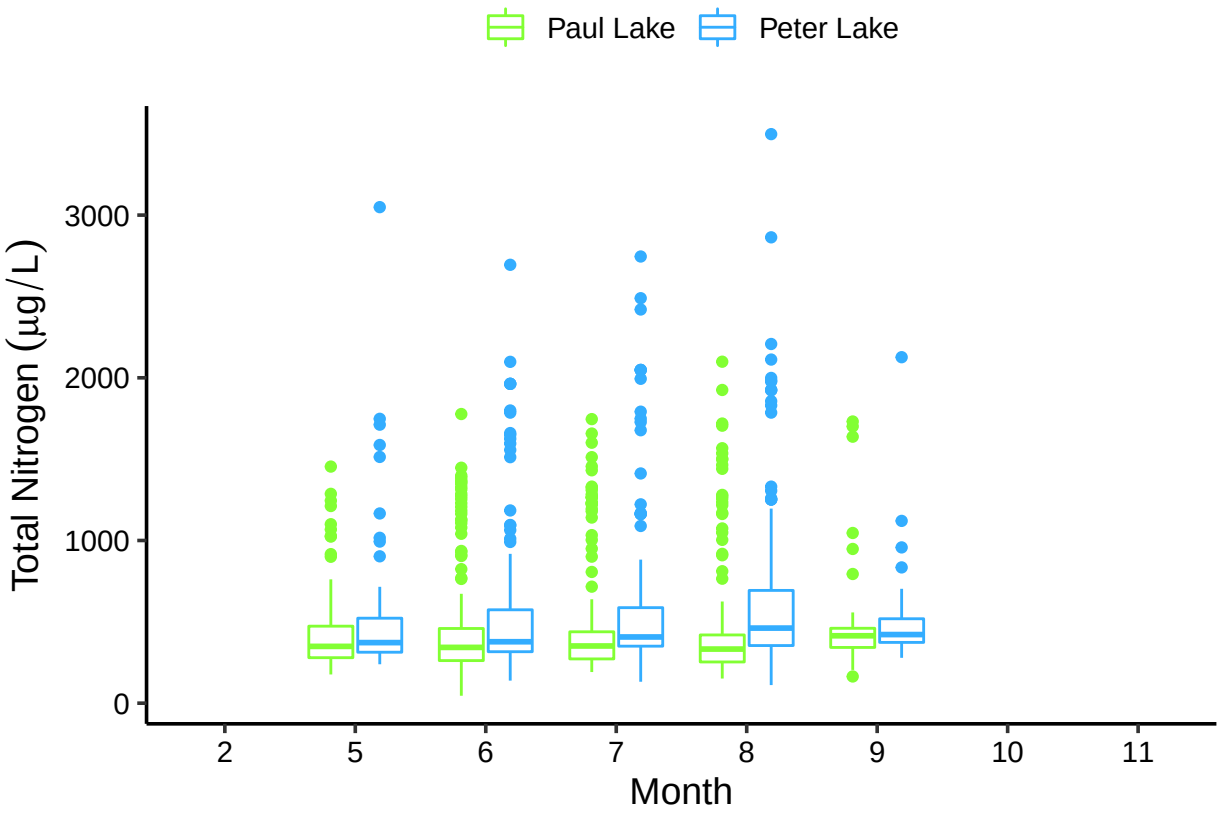

```
TP_boxplot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = as.factor(month), y = tp_ug, color = lakename))+
  geom_boxplot()+
  labs(x = "Month", y = expression("Total Phosphorus" ~ (mu*g / L)), color = "")+
  scale_color_manual(values = c("#82FF33", "#33ADFF"))
print(TP_boxplot)
```

```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```




```
TN_boxplot <-
  ggplot(PeterPaul.chem.nutrients, aes(y = tn_ug, x = as.factor(month), color = lakename))+
  geom_boxplot()+
   labs(x = "Month", y = expression("Total Nitrogen" ~ (mu*g / L)), color = "")+
  scale_color_manual(values = c("#82FF33", "#33ADFF"))
print(TN_boxplot)
```

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




```
### had to replot each to remove legend so that only one legend would show in cowplot

Temp_boxplot1 <-
  ggplot(PeterPaul.chem.nutrients, aes(y = temperature_C, x = as.factor(month), color = lakename))+
  geom_boxplot()+
  labs(x = "Month", y = expression("Temperature " ( degree*C)), color = "")+
   scale_color_manual(values = c("#82FF33", "#33ADFF"))+
  theme(legend.position = "none")

TP_boxplot1 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = as.factor(month), y = tp_ug, color = lakename))+
  geom_boxplot()+
  labs(x = "Month", y = expression("Phosphorus" ~ (mu*g / L)), color = "")+
  scale_color_manual(values = c("#82FF33", "#33ADFF"))+
   theme(legend.position = "none")

TN_boxplot1 <-
  ggplot(PeterPaul.chem.nutrients, aes(y = tn_ug, x = as.factor(month), color = lakename))+
  geom_boxplot()+
   labs(x = "Month", y = expression("Nitrogen" ~ (mu*g / L)), color = "")+
  scale_color_manual(values = c("#82FF33", "#33ADFF"))+
   theme(legend.position = "none")


library(cowplot)
```

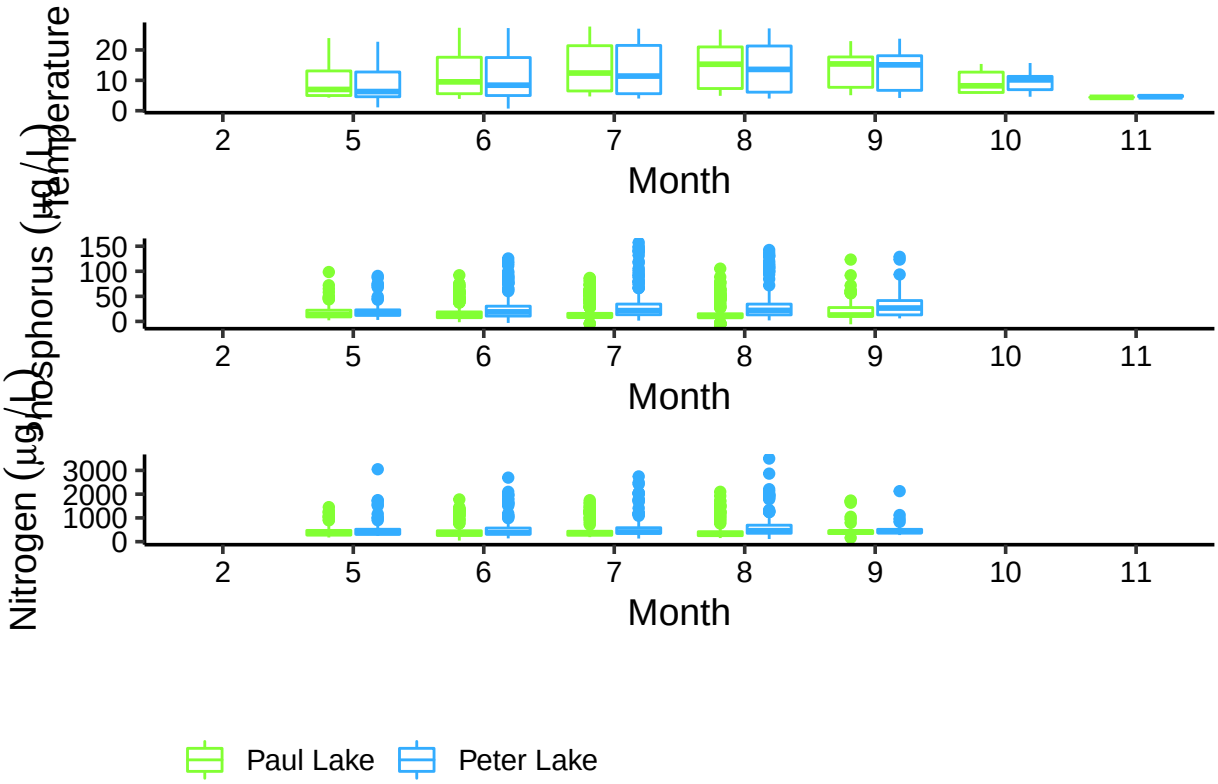
```
##
## ************************************************************
## Note: As of version 1.0.0, cowplot does not change the
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## ************************************************************
```

```
legend <- get_legend(Temp_boxplot + theme(legend.position = "top"))
```

```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```

```
plot_grid(Temp_boxplot1, TP_boxplot1, TN_boxplot1, ncol = 1, align = "v", axis = "ltb", legend)
```

```
## 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).
```



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

> Answer: Temperature follows a predictable seasonal pattern with increased T in July, August, September. The median temp of Paul lake is slightly higher than the median T of Peter Lake. Both phosphorus and nitrogen peak in the summer, around August and seem to have a similar pattern to T. The median concentration of both P and N is higher slightly in Peter Lake and Peter has higher 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)

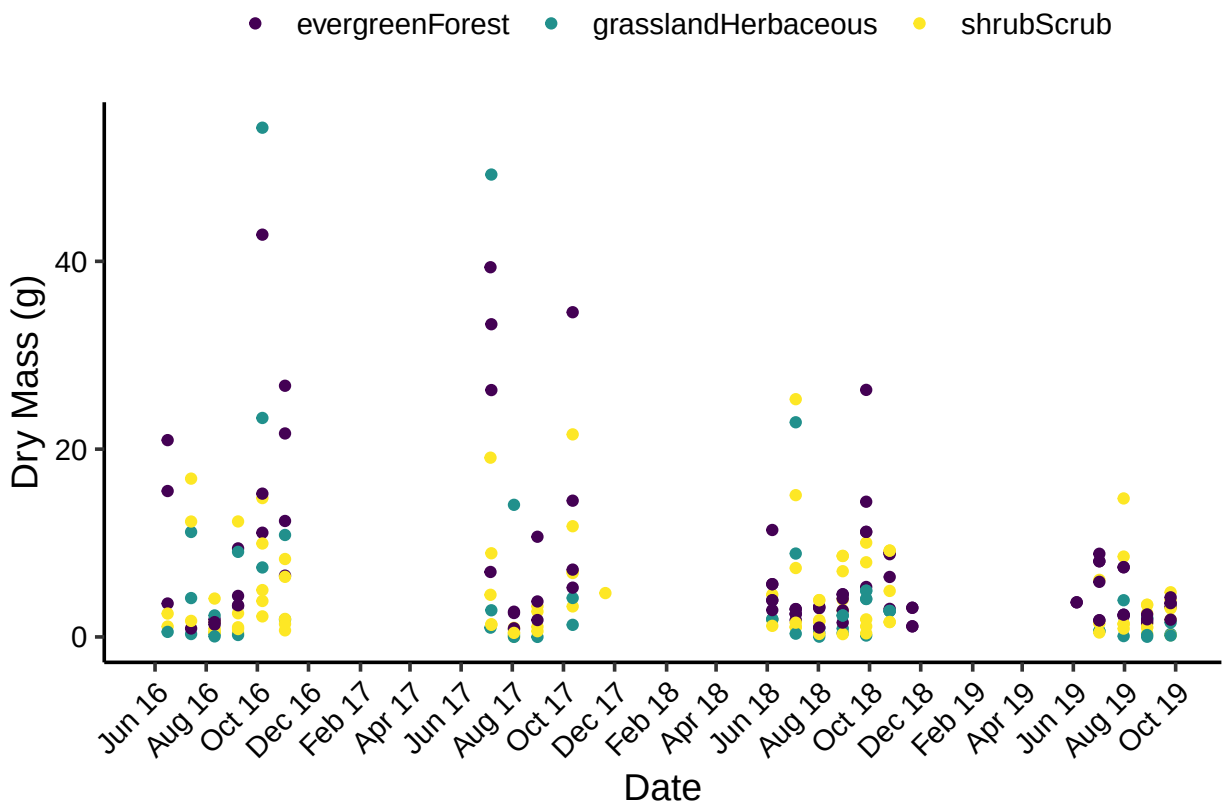
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
library(viridis)
```

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

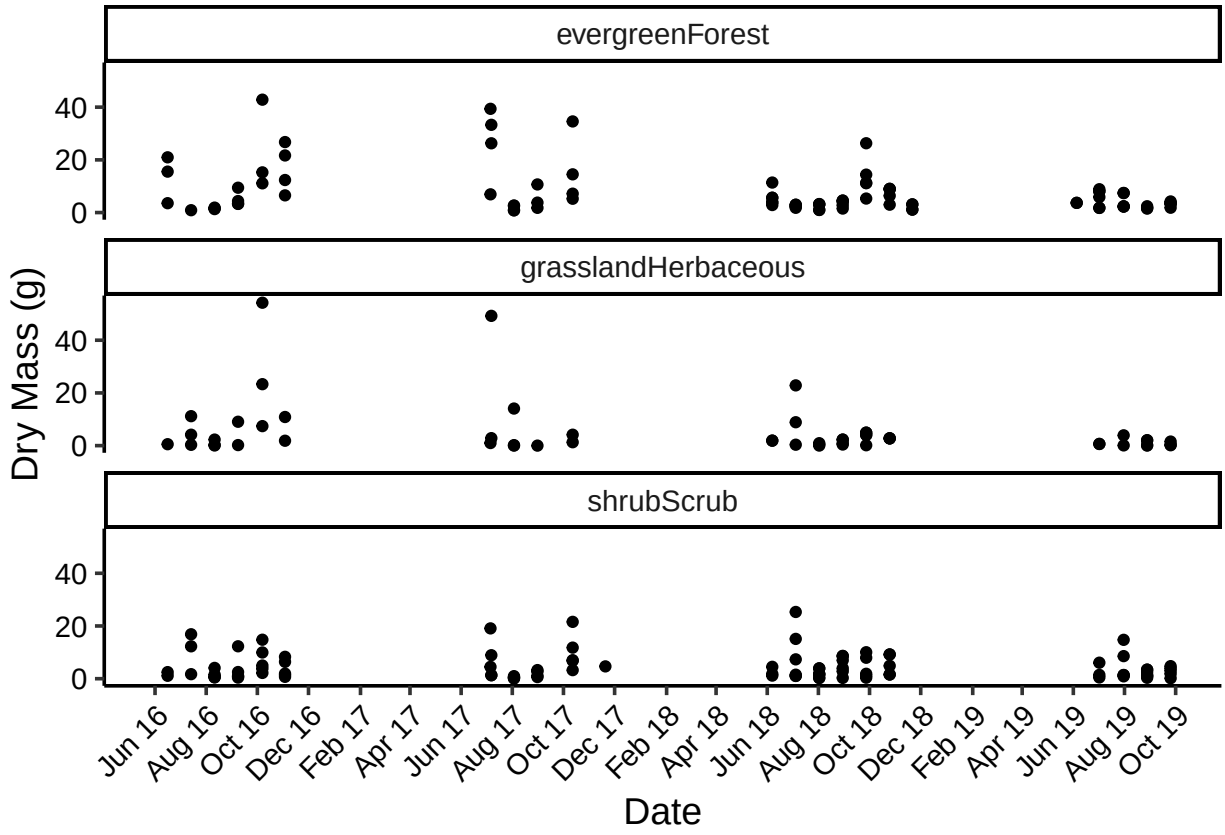
```
library(RColorBrewer)
library(colormap)

LitterPlot_color <-
  ggplot(subset(Niwot.Litter, functionalGroup == "Needles"))+
  aes(x = collectDate, y = dryMass, color = nlcdClass)+
  geom_point()+
  scale_x_date(limits = as.Date(c("2016-06-01", "2019-09-31")),
    date_breaks = "2 months", date_labels = "%b %y")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  labs(x = "Date", y = "Dry Mass (g)", color = "")+
  scale_color_viridis(discrete=TRUE)
print(LitterPlot_color)
```




```
LitterPlot_facet <-
  ggplot(subset(Niwot.Litter, functionalGroup == "Needles"))+
  aes(x = collectDate, y = dryMass)+
```

```
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  scale_x_date(limits = as.Date(c("2016-06-01", "2019-09-31")),
    date_breaks = "2 months", date_labels = "%b %y")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  labs(x = "Date", y = "Dry Mass (g)")
print(LitterPlot_facet)
```



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

> Answer: 7 is a better representation because you can clearly see the difference between the three different landcovers. In 6, a lot of the points overlap so eventhough they are different colors it is difficult to distinguish trends.