Assignment 6: GLMs week 1 (t-test and ANOVA)

Samantha Burch

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on t-tests and ANOVAs.

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_A06_GLMs_Week1.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 18 at 1:00 pm.

Set up your session

1. Check your working directory, load the `tidyverse`, `cowplot`, and `agricolae` packages, and import the NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv dataset.
2. Change the date column to a date format. Call up `head` of this column to verify.

```
#1
getwd()
```

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

```
library(tidyverse)
library(cowplot)
library(agricolae)
library(ggplot2)

Litter <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv")

#2
Litter$sampledate <- as.Date(Litter$sampledate , format = "%Y-%m-%d")
class(Litter$sampledate)
```

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

Wrangle your data

3. Wrangle your dataset so that it contains only surface depths and only the years 1993-1996, inclusive. Set month as a factor.

```
Litter.Wrangle <- Litter %>%
  filter(depth == 0 & year4 > 1992 & year4 < 1997)
```

```r
Litter.Wrangle$month <- as.factor(Litter.Wrangle$month)
```

Analysis

Peter Lake was manipulated with additions of nitrogen and phosphorus over the years 1993-1996 in an effort to assess the impacts of eutrophication in lakes. You are tasked with finding out if nutrients are significantly higher in Peter Lake than Paul Lake, and if these potential differences in nutrients vary seasonally (use month as a factor to represent seasonality). Run two separate tests for TN and TP.

4. Which application of the GLM will you use (t-test, one-way ANOVA, two-way ANOVA with main effects, or two-way ANOVA with interaction effects)? Justify your choice.

   Answer: I would run a two-way ANOVA first and then with interaction effects, because I will eventually want to examine both the individual effects and interactions of the explanatory variables. Moreover, I want to see if seasonality has an effect on nutrient levels of TN and TP.

5. Run your test for TN. Include examination of groupings and consider interaction effects, if relevant.

6. Run your test for TP. Include examination of groupings and consider interaction effects, if relevant.

```r
#5 TN, When we run aov for TN w/ lakename and month, we do not see a significant interaction between la
Litter.2wayanova.TN <- aov(data = Litter.Wrangle, Litter.Wrangle$tn_ug ~ lakename + month)
summary(Litter.2wayanova.TN)
```

```
##              Df  Sum Sq Mean Sq F value   Pr(>F)    
## lakename      1 2468595 2468595   36.32 2.75e-08 ***
## month         4  459542  114885    1.69    0.158    
## Residuals   101 6864107   67961                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 23 observations deleted due to missingness
```

```r
str(Litter.Wrangle)
```

```
## 'data.frame':    130 obs. of  14 variables:
##  $ lakeid    : Factor w/ 2 levels "L","R": 2 1 2 1 2 1 2 1 2 1 ...
##  $ lakename  : Factor w/ 2 levels "Paul Lake","Peter Lake": 2 1 2 1 2 1 2 1 2 1 ...
##  $ year4     : int  1993 1993 1993 1993 1993 1993 1993 1993 1993 1993 ...
##  $ daynum    : int  139 140 146 147 153 154 160 161 167 168 ...
##  $ month     : Factor w/ 5 levels "5","6","7","8",..: 1 1 1 1 2 2 2 2 2 2 ...
##  $ sampledate: Date, format: "1993-05-19" "1993-05-20" ...
##  $ depth_id  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ depth     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ tn_ug     : num  373 249 313 258 415 ...
##  $ tp_ug     : num  18.5 10.7 16.8 7 24.6 ...
##  $ nh34      : num  0 0 9.7 8.66 10.64 ...
##  $ no23      : num  5.5 1.15 9.93 3.21 11.54 ...
##  $ po4       : num  1 2 2 1 3 1 4 2 5 0 ...
##  $ comments  : logi  NA NA NA NA NA NA ...
```

```r
TukeyHSD(Litter.2wayanova.TN)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Litter.Wrangle$tn_ug ~ lakename + month, data = Litter.Wrangle)
##
```

```
## $lakename
##                          diff      lwr      upr p adj
## Peter Lake-Paul Lake 303.796 203.8026 403.7894     0
##
## $month
##          diff        lwr      upr     p adj
## 6-5 132.58168 -104.53533 369.6987 0.5307817
## 7-5 196.50011  -47.94924 440.9495 0.1761663
## 8-5 208.77984  -32.91447 450.4741 0.1238871
## 9-5 160.08048 -220.97835 541.1393 0.7701126
## 7-6  63.91843 -123.99128 251.8281 0.8785969
## 8-6  76.19815 -108.11330 260.5096 0.7803543
## 9-6  27.49879 -320.00718 375.0048 0.9994732
## 8-7  12.27972 -181.37388 205.9333 0.9997809
## 9-7 -36.41964 -388.96950 316.1302 0.9984948
## 9-8 -48.69936 -399.34457 301.9458 0.9952369
```

```r
#6 TP
Litter.2wayanova.TP <- aov(data = Litter.Wrangle, tp_ug ~ lakename + month)
summary(Litter.2wayanova.TP)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## lakename      1  10228   10228  94.453 <2e-16 ***
## month         4    813     203   1.876  0.119
## Residuals   123  13320     108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

```r
TukeyHSD(Litter.2wayanova.TP)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = tp_ug ~ lakename + month, data = Litter.Wrangle)
##
## $lakename
##                          diff      lwr      upr p adj
## Peter Lake-Paul Lake 17.80939 14.18208 21.43669     0
##
## $month
##           diff        lwr       upr     p adj
## 6-5  6.3451786  -3.012727 15.703084 0.3350273
## 7-5  8.8661326  -0.491773 18.224038 0.0723646
## 8-5  4.8191843  -4.469970 14.108339 0.6055077
## 9-5  5.4951391  -6.998304 17.988582 0.7410806
## 7-6  2.5209540  -4.366278  9.408186 0.8487741
## 8-6 -1.5259943  -8.319518  5.267530 0.9713266
## 9-6 -0.8500395 -11.618033  9.917954 0.9994865
## 8-7 -4.0469483 -10.840472  2.746576 0.4691480
## 9-7 -3.3709935 -14.138987  7.397000 0.9084852
## 9-8  0.6759548 -10.032345 11.384255 0.9997883
```

```r
#Interaction effects
Litter.2wayanova.TP.int <- aov(data = Litter.Wrangle, tp_ug ~ lakename * month) #only thing that change
summary(Litter.2wayanova.TP.int)
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)
## lakename          1  10228   10228  98.914 <2e-16 ***
## month             4    813     203   1.965 0.1043
## lakename:month    4   1014     254   2.452 0.0496 *
## Residuals       119  12305     103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

```
TP.int <- with(Litter.Wrangle, interaction(lakename, month))
TP.anova.int <- aov(data = Litter.Wrangle, tp_ug ~ TP.int)

TP.groups <- HSD.test(TP.anova.int, "TP.int", group = TRUE) #we do this bc it is significant, p value i

Litter.2wayanova.TN.int <- aov(data = Litter.Wrangle, tn_ug ~ lakename * month) #only thing that change
summary(Litter.2wayanova.TN.int)
```

```
##                Df  Sum Sq Mean Sq F value   Pr(>F)
## lakename        1 2468595 2468595  36.414 2.91e-08 ***
## month           4  459542  114885   1.695    0.157
## lakename:month  4  288272   72068   1.063    0.379
## Residuals      97 6575834   67792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 23 observations deleted due to missingness
```

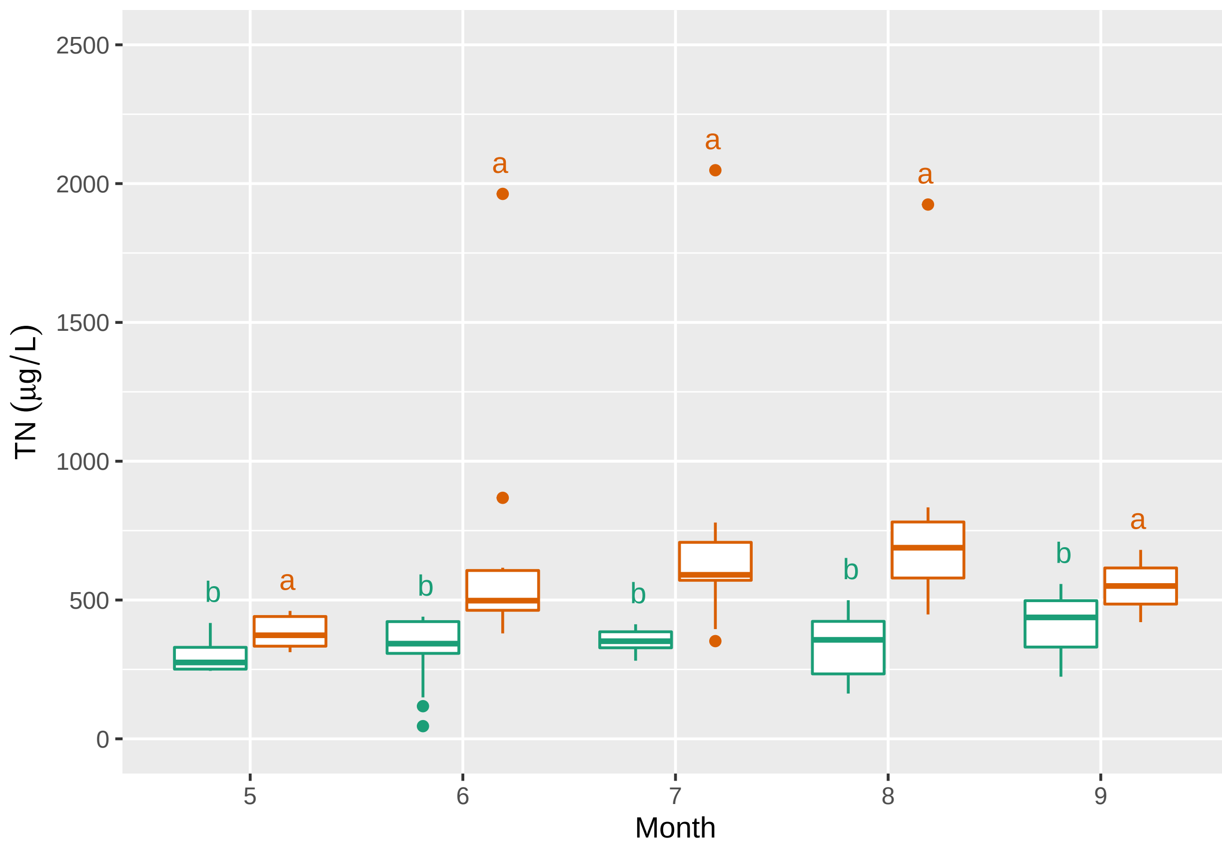
7. Create two plots, with TN (plot 1) or TP (plot 2) as the response variable and month and lake as the predictor variables. Hint: you may use some of the code you used for your visualization assignment. Assign groupings with letters, as determined from your tests. Adjust your axes, aesthetics, and color palettes in accordance with best data visualization practices.

8. Combine your plots with cowplot, with a common legend at the top and the two graphs stacked vertically. Your x axes should be formatted with the same breaks, such that you can remove the title and text of the top legend and retain just the bottom legend.

```
#7.1 TN boxplot
TNplot <-
    ggplot(Litter.Wrangle, aes(y = tn_ug, x = as.factor(month), color = lakename)) +
  geom_boxplot() +
  labs(x = expression(paste("Month")),
       y = expression(paste("TN" ~ (mu*g / L)))) +
  scale_color_brewer(palette = "Dark2") +
  ylim(0,2500) +
  stat_summary(geom = "text", fun.y = max, vjust = -1, size = 4, position = position_dodge(.7), label =
  theme(legend.position = "none")
print(TNplot)
```

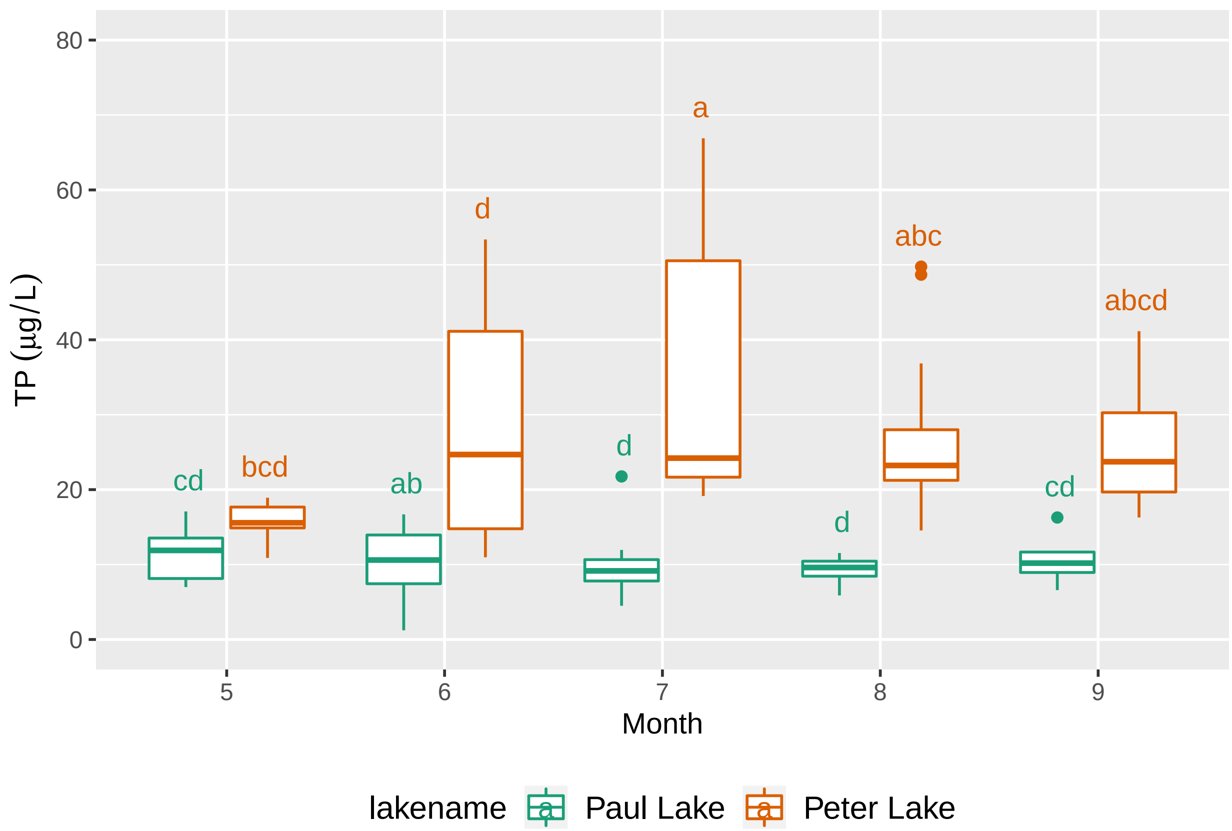
```
## Warning: Removed 23 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 23 rows containing non-finite values (stat_summary).
```




```r
#7.2 TP boxplot
TPplot <-
    ggplot(Litter.Wrangle, aes(y = tp_ug, x = as.factor(month), color = lakename)) +
  geom_boxplot() +
  labs(x = expression(paste("Month")),
       y = expression(paste("TP" ~ (mu*g / L)))) +
  scale_color_brewer(palette = "Dark2") +
  ylim(0,80) +
  stat_summary(geom = "text", fun.y = max, vjust = -1, size = 4, position = position_dodge(.7), label =
  theme(legend.position = "bottom",
        legend.text = element_text(size = 12), legend.title = element_text(size = 12))
print(TPplot)
```

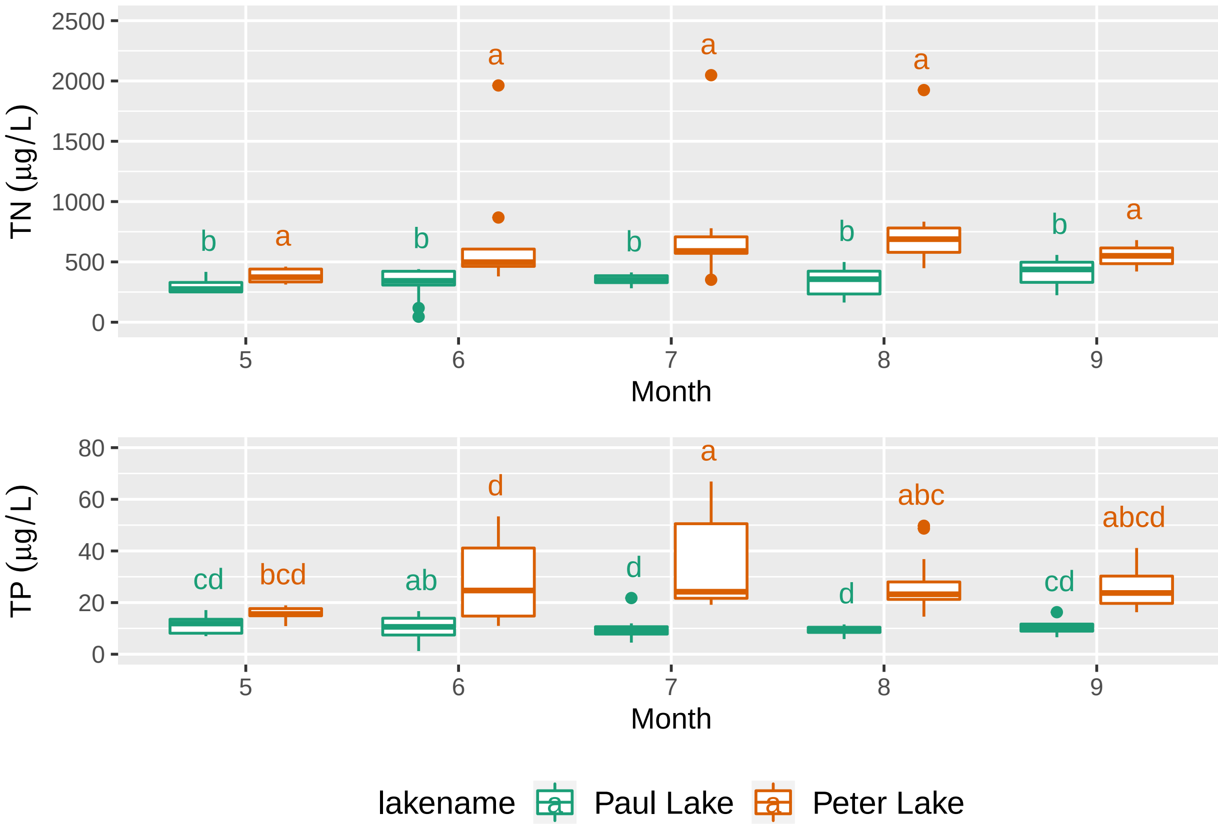
```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## Warning: Removed 1 rows containing non-finite values (stat_summary).
```




```r
#8
TP_TN_plots <- plot_grid(TNplot, TPplot,
                    align = "vh", ncol = 1)
```

```
## Warning: Removed 23 rows containing non-finite values (stat_boxplot).

## Warning: Removed 23 rows containing non-finite values (stat_summary).

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## Warning: Removed 1 rows containing non-finite values (stat_summary).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```r
print(TP_TN_plots)
```


TN (µg/L)
2500
2000
1500
1000
500
0
b
a
b
a
b
a
b
a
b
a
5
6
7
8
9
Month
TP (µg/L)
80
60
40
20
0
cd
bcd
ab
d
d
a
d
abc
cd
abcd
5
6
7
8
9
Month
lakename
Paul Lake
Peter Lake