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

Emily McNamara

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/emilymcnamara/Desktop/Env Data Analytics/Environmental_Data_Analytics_2020"
```

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

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

#2

PP.nutrients$sampledate <- as.Date(PP.nutrients$sampledate, format = "%Y-%m-%d")
class(PP.nutrients$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.

```
PP.Depths <- PP.nutrients %>%
  filter(depth == 0 & year4 > 1992 & year4 < 1997)
```

```r
PP.Depths$month <- as.factor(PP.Depths$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: Two-way ANOVA because we want to examine the effects of two categorical explanatory/predictable variables (Lakes and month) on a continuous response variable (nutrients).

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.anova.2way <- aov(data = PP.Depths, tn_ug ~ lakename + month)
summary(TN.anova.2way)
```

```
##              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
TukeyHSD(TN.anova.2way)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = tn_ug ~ lakename + month, data = PP.Depths)
##
## $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.anova.2way <- aov(data = PP.Depths, tp_ug ~ lakename + month)
summary(TN.anova.2way)
```

```
##              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
TukeyHSD(TP.anova.2way)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = tp_ug ~ lakename + month, data = PP.Depths)
##
## $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
#6.1 Interaction Effects

TP.anova.2way3 <- aov(data = PP.Depths, tp_ug ~ lakename * month)
summary(TP.anova.2way3)
```

```
##                 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
```

```r
TP.interaction <- with(PP.Depths, interaction(lakename, month))
TP.anova.2way4 <- aov(data = PP.Depths, tp_ug ~ TP.interaction)

TP.groups <- HSD.test(TP.anova.2way4, "TP.interaction", group = TRUE)
TP.groups
```

```
## $statistics
##    MSerror  Df     Mean      CV
##   103.4055 119 19.07347 53.3141
##
## $parameters
##    test         name.t ntr StudentizedRange alpha
##   Tukey TP.interaction  10         4.560262  0.05
##
## $means
##                   tp_ug       std  r    Min    Max     Q25     Q50      Q75
## Paul Lake.5  11.474000  3.928545  6  7.001 17.090  8.1395 11.8885 13.53675
## Paul Lake.6  10.556118  4.416821 17  1.222 16.697  7.4430 10.6050 13.94600
## Paul Lake.7   9.746889  3.525120 18  4.501 21.763  7.8065  9.1555 10.65700
## Paul Lake.8   9.386778  1.478062 18  5.879 11.542  8.4495  9.6090 10.45050
## Paul Lake.9  10.736000  3.615978  5  6.592 16.281  8.9440 10.1920 11.67100
## Peter Lake.5 15.787571  2.719954  7 10.887 18.922 14.8915 15.5730 17.67400
## Peter Lake.6 28.357889 15.588507 18 10.974 53.388 14.7790 24.6840 41.13000
## Peter Lake.7 34.404471 18.285568 17 19.149 66.893 21.6640 24.2070 50.54900
## Peter Lake.8 26.494000  9.829596 19 14.551 49.757 21.2425 23.2250 27.99350
## Peter Lake.9 26.219250 10.814803  4 16.281 41.145 19.6845 23.7255 30.26025
##
## $comparison
## NULL
##
## $groups
##                   tp_ug groups
## Peter Lake.7 34.404471      a
## Peter Lake.6 28.357889     ab
## Peter Lake.8 26.494000    abc
## Peter Lake.9 26.219250   abcd
## Peter Lake.5 15.787571    bcd
## Paul Lake.5  11.474000     cd
## Paul Lake.9  10.736000     cd
## Paul Lake.6  10.556118      d
## Paul Lake.7   9.746889      d
## Paul Lake.8   9.386778      d
##
## attr(,"class")
## [1] "group"
```

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.

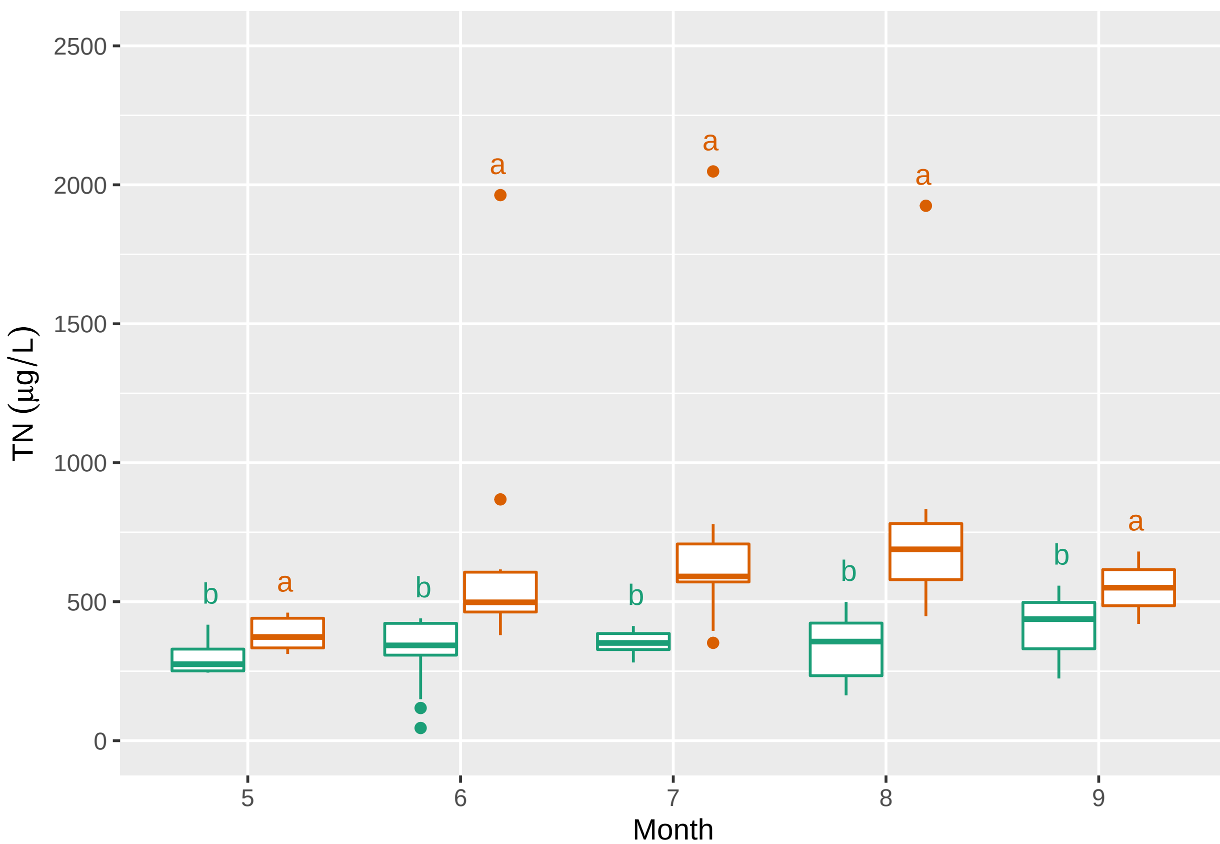
8. Combine your plots with cowplot, with a common legend at the top and the two graphs stacked vertically. Your x axes should be formatted with the same breaks, such that you can remove the title and text of the top legend and retain just the bottom legend.

```
#7

# TN Boxplot

TN.boxplot <-
```

```
    ggplot(PP.Depths, 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 = c("a", "b", "a", "b", "a", "b", "a", "b", "a", "b")) +
      theme(legend.position = "none")
print(TN.boxplot)
```

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

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

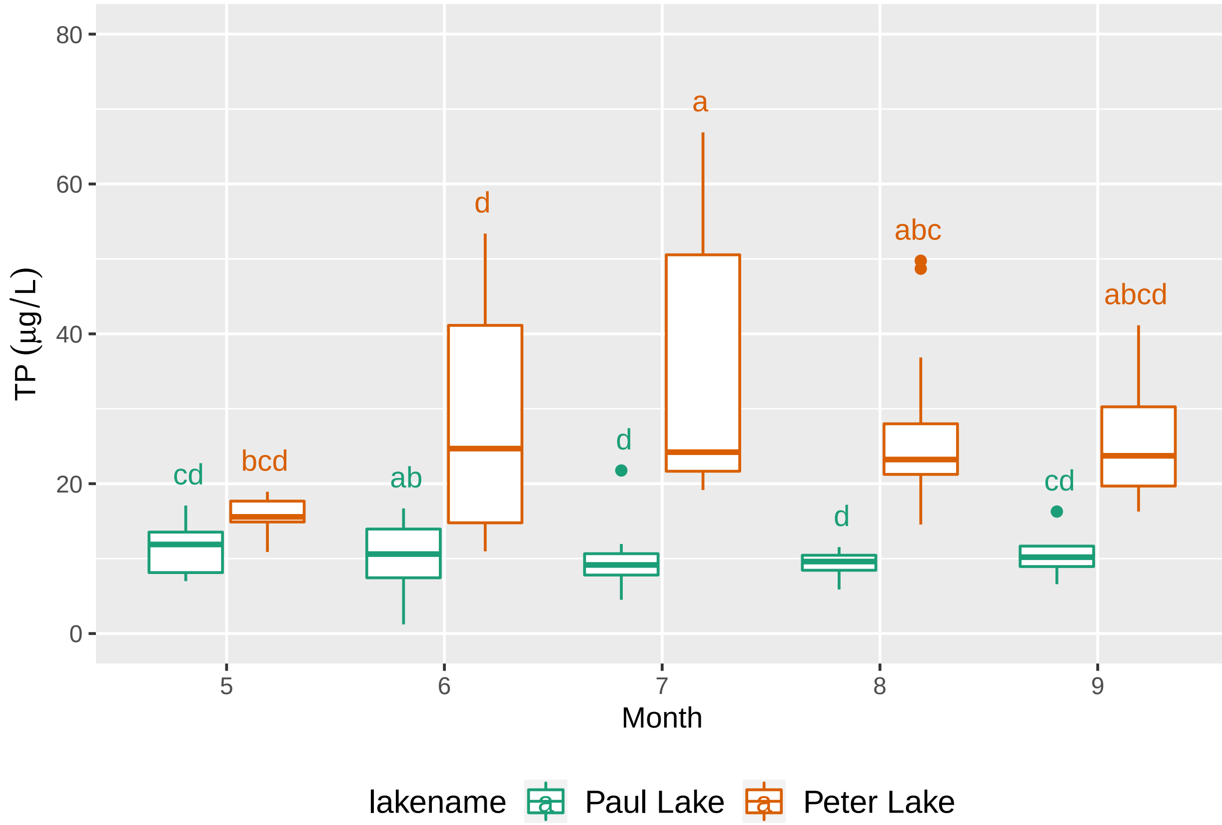



```
# TP Boxplot
TP.boxplot <-
    ggplot(PP.Depths, aes(y = tp_ug, x = as.factor(month),  color = lakename)) +
      geom_boxplot() +
  labs(x = expression(paste("Month")),
       y = expression(paste("TP"~ (mu*g / L)))) +
  scale_color_brewer(palette = "Dark2") +
  ylim(0,80) +
      stat_summary(geom = "text", fun.y = max, vjust = -1, size = 4, position = position_dodge(.7),
                   label = c("bcd", "cd", "d", "ab", "a", "d", "abc", "d", "abcd", "cd")) +
      theme(legend.position = "bottom",
        legend.text = element_text(size = 12), legend.title = element_text(size = 12))
```

```r
print(TP.boxplot)
```

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

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




```r
#8

TNTP.plots <- plot_grid(TN.boxplot, TP.boxplot,
                        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.
```

```
print(TNTP.plots)
```