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

Student Name

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/amandabraun/Documents/Classes Spring 2020/Data Analytics/Environmental_Data_Analytics_20
```

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

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

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

```
## [1] "1991-05-20" "1991-05-20" "1991-05-20" "1991-05-20" "1991-05-20"
## [6] "1991-05-20"
```

```
class(PP_Nutrient_Processed$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.

```
NT.PeterPaul.SurfaceDepths.1993.1996 <- PP_Nutrient_Processed  %>%
  filter(depth == 0) %>%
  filter(year4 == "1993" | year4 == "1994" | year4 == "1995" | year4 == "1996")

NT.PeterPaul.SurfaceDepths.1993.1996$month <-factor (NT.PeterPaul.SurfaceDepths.1993.1996$month)

#factor makes the numbers a category, so you're able to use month as categories as predictors for a con
```

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: A two-way anova with interactions should be used to investigate the interaction of two different categorical variables (months to represent seasonality, and lake name) on the continuous response variable (total nitrogen or total phosphorous). We are testing to see if there is a significant interaction between the two categorial variables.

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.

```
#5
TN.SurfaceDepths.2wayanova.nitrogen <- aov(data = NT.PeterPaul.SurfaceDepths.1993.1996, tn_ug ~ lakenam
summary(TN.SurfaceDepths.2wayanova.nitrogen)
```

```
##                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
```

```
TukeyHSD(TN.SurfaceDepths.2wayanova.nitrogen)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = tn_ug ~ lakename * month, data = NT.PeterPaul.SurfaceDepths.1993.1996)
##
## $lakename
##                          diff      lwr      upr p adj
## Peter Lake-Paul Lake 303.796 203.8773 403.7146     0
##
## $month
##          diff       lwr      upr     p adj
## 6-5 132.58168 -104.4173 369.5807 0.5296645
## 7-5 196.50011  -47.8276 440.8278 0.1755245
## 8-5 208.77984  -32.7942 450.3539 0.1234174
## 9-5 160.08048 -220.7887 540.9497 0.7692917
```

```
## 7-6  63.91843 -123.8978 251.7346 0.8780820
## 8-6  76.19815 -108.0216 260.4179 0.7795574
## 9-6  27.49879 -319.8343 374.8318 0.9994702
## 8-7  12.27972 -181.2775 205.8370 0.9997797
## 9-7 -36.41964 -388.7941 315.9548 0.9984863
## 9-8 -48.69936 -399.1701 301.7714 0.9952106
##
## $`lakename:month`
##                                    diff         lwr       upr     p adj
## Peter Lake:5-Paul Lake:5    84.42736 -384.695091 553.54981 0.9998802
## Paul Lake:6-Paul Lake:5     23.61297 -376.795278 424.02122 1.0000000
## Peter Lake:6-Paul Lake:5   308.53119  -95.128061 712.19044 0.2949521
## Paul Lake:7-Paul Lake:5     53.12257 -358.325034 464.57018 0.9999929
## Peter Lake:7-Paul Lake:5   409.37327   -6.794730 825.54127 0.0577843
## Paul Lake:8-Paul Lake:5     35.99664 -375.450962 447.44425 0.9999998
## Peter Lake:8-Paul Lake:5   445.47177   38.159418 852.78411 0.0206524
## Paul Lake:9-Paul Lake:5    105.82450 -490.419726 702.06873 0.9998933
## Peter Lake:9-Paul Lake:5   249.95650 -438.527028 938.44003 0.9743614
## Paul Lake:6-Peter Lake:5   -60.81439 -439.493476 317.86470 0.9999541
## Peter Lake:6-Peter Lake:5  224.10383 -158.011173 606.21883 0.6694487
## Paul Lake:7-Peter Lake:5   -31.30479 -421.638257 359.02869 0.9999999
## Peter Lake:7-Peter Lake:5  324.94591  -70.360160 720.25198 0.2042224
## Paul Lake:8-Peter Lake:5   -48.43071 -438.764185 341.90276 0.9999950
## Peter Lake:8-Peter Lake:5  361.04441  -24.927657 747.01648 0.0870846
## Paul Lake:9-Peter Lake:5    21.39714 -560.477640 603.27193 1.0000000
## Peter Lake:9-Peter Lake:5  165.52914 -510.548261 841.60655 0.9985431
## Peter Lake:6-Paul Lake:6   284.91822   -8.787028 578.62346 0.0650344
## Paul Lake:7-Paul Lake:6     29.50960 -274.811140 333.83034 0.9999994
## Peter Lake:7-Paul Lake:6   385.76030   75.087182 696.43342 0.0043241
## Paul Lake:8-Paul Lake:6     12.38367 -291.937068 316.70441 1.0000000
## Peter Lake:8-Paul Lake:6   421.85880  123.152702 720.56489 0.0005774
## Paul Lake:9-Paul Lake:6     82.21153 -445.831232 610.25429 0.9999647
## Peter Lake:9-Paul Lake:6   226.34353 -403.998878 856.68594 0.9761624
## Paul Lake:7-Peter Lake:6  -255.40862 -563.994320  53.17709 0.1964898
## Peter Lake:7-Peter Lake:6  100.84208 -214.009961 415.69412 0.9891274
## Paul Lake:8-Peter Lake:6  -272.53454 -581.120248  36.05116 0.1316086
## Peter Lake:8-Peter Lake:6  136.94058 -166.109506 439.99066 0.9029804
## Paul Lake:9-Peter Lake:6  -202.70669 -733.218875 327.80550 0.9642843
## Peter Lake:9-Peter Lake:6  -58.57469 -690.987190 573.83782 0.9999996
## Peter Lake:7-Paul Lake:7   356.25070   31.473618 681.02778 0.0200027
## Paul Lake:8-Paul Lake:7    -17.12593 -335.831873 301.58002 1.0000000
## Peter Lake:8-Paul Lake:7   392.34920   79.000035 705.69836 0.0038467
## Paul Lake:9-Paul Lake:7     52.70193 -483.760115 589.16397 0.9999994
## Peter Lake:9-Paul Lake:7   196.83393 -440.577960 834.24582 0.9916222
## Paul Lake:8-Peter Lake:7  -373.37663 -698.153706 -48.59955 0.0116944
## Peter Lake:8-Peter Lake:7   36.09850 -283.423597 355.62059 0.9999978
## Paul Lake:9-Peter Lake:7  -303.54877 -843.639684 236.54215 0.7209271
## Peter Lake:9-Peter Lake:7 -159.41677 -799.885807 481.05227 0.9983429
## Peter Lake:8-Paul Lake:8   409.47512   96.125963 722.82428 0.0020552
## Paul Lake:9-Paul Lake:8     69.82786 -466.634186 606.28990 0.9999924
## Peter Lake:9-Paul Lake:8   213.95986 -423.452032 851.37175 0.9849047
## Paul Lake:9-Peter Lake:8  -339.64727 -872.944314 193.64978 0.5579223
## Peter Lake:9-Peter Lake:8 -195.51527 -830.265716 439.23518 0.9917740
## Peter Lake:9-Paul Lake:9   144.13200 -625.615985 913.87999 0.9998333
```

```r
#6
TP.SurfaceDepths.2wayanova <- aov(data = NT.PeterPaul.SurfaceDepths.1993.1996, tn_ug ~ lakename * month)
summary(TP.SurfaceDepths.2wayanova)
```

```
##                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
```

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

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = tn_ug ~ lakename * month, data = NT.PeterPaul.SurfaceDepths.1993.1996)
##
## $lakename
##                          diff      lwr      upr p adj
## Peter Lake-Paul Lake 303.796 203.8773 403.7146     0
##
## $month
##          diff       lwr      upr     p adj
## 6-5 132.58168 -104.4173 369.5807 0.5296645
## 7-5 196.50011  -47.8276 440.8278 0.1755245
## 8-5 208.77984  -32.7942 450.3539 0.1234174
## 9-5 160.08048 -220.7887 540.9497 0.7692917
## 7-6  63.91843 -123.8978 251.7346 0.8780820
## 8-6  76.19815 -108.0216 260.4179 0.7795574
## 9-6  27.49879 -319.8343 374.8318 0.9994702
## 8-7  12.27972 -181.2775 205.8370 0.9997797
## 9-7 -36.41964 -388.7941 315.9548 0.9984863
## 9-8 -48.69936 -399.1701 301.7714 0.9952106
##
## $`lakename:month`
##                                    diff         lwr       upr     p adj
## Peter Lake:5-Paul Lake:5     84.42736 -384.695091 553.54981 0.9998802
## Paul Lake:6-Paul Lake:5      23.61297 -376.795278 424.02122 1.0000000
## Peter Lake:6-Paul Lake:5    308.53119  -95.128061 712.19044 0.2949521
## Paul Lake:7-Paul Lake:5      53.12257 -358.325034 464.57018 0.9999929
## Peter Lake:7-Paul Lake:5    409.37327   -6.794730 825.54127 0.0577843
## Paul Lake:8-Paul Lake:5      35.99664 -375.450962 447.44425 0.9999998
## Peter Lake:8-Paul Lake:5    445.47177   38.159418 852.78411 0.0206524
## Paul Lake:9-Paul Lake:5     105.82450 -490.419726 702.06873 0.9998933
## Peter Lake:9-Paul Lake:5    249.95650 -438.527028 938.44003 0.9743614
## Paul Lake:6-Peter Lake:5    -60.81439 -439.493476 317.86470 0.9999541
## Peter Lake:6-Peter Lake:5   224.10383 -158.011173 606.21883 0.6694487
## Paul Lake:7-Peter Lake:5    -31.30479 -421.638257 359.02869 0.9999999
## Peter Lake:7-Peter Lake:5   324.94591  -70.360160 720.25198 0.2042224
## Paul Lake:8-Peter Lake:5    -48.43071 -438.764185 341.90276 0.9999950
## Peter Lake:8-Peter Lake:5   361.04441  -24.927657 747.01648 0.0870846
```

```
## Paul Lake:9-Peter Lake:5    21.39714 -560.477640 603.27193 1.0000000
## Peter Lake:9-Peter Lake:5  165.52914 -510.548261 841.60655 0.9985431
## Peter Lake:6-Paul Lake:6   284.91822   -8.787028 578.62346 0.0650344
## Paul Lake:7-Paul Lake:6     29.50960 -274.811140 333.83034 0.9999994
## Peter Lake:7-Paul Lake:6   385.76030   75.087182 696.43342 0.0043241
## Paul Lake:8-Paul Lake:6     12.38367 -291.937068 316.70441 1.0000000
## Peter Lake:8-Paul Lake:6   421.85880  123.152702 720.56489 0.0005774
## Paul Lake:9-Paul Lake:6     82.21153 -445.831232 610.25429 0.9999647
## Peter Lake:9-Paul Lake:6   226.34353 -403.998878 856.68594 0.9761624
## Paul Lake:7-Peter Lake:6  -255.40862 -563.994320  53.17709 0.1964898
## Peter Lake:7-Peter Lake:6  100.84208 -214.009961 415.69412 0.9891274
## Paul Lake:8-Peter Lake:6  -272.53454 -581.120248  36.05116 0.1316086
## Peter Lake:8-Peter Lake:6  136.94058 -166.109506 439.99066 0.9029804
## Paul Lake:9-Peter Lake:6  -202.70669 -733.218875 327.80550 0.9642843
## Peter Lake:9-Peter Lake:6  -58.57469 -690.987190 573.83782 0.9999996
## Peter Lake:7-Paul Lake:7   356.25070   31.473618 681.02778 0.0200027
## Paul Lake:8-Paul Lake:7    -17.12593 -335.831873 301.58002 1.0000000
## Peter Lake:8-Paul Lake:7   392.34920   79.000035 705.69836 0.0038467
## Paul Lake:9-Paul Lake:7     52.70193 -483.760115 589.16397 0.9999994
## Peter Lake:9-Paul Lake:7   196.83393 -440.577960 834.24582 0.9916222
## Paul Lake:8-Peter Lake:7  -373.37663 -698.153706 -48.59955 0.0116944
## Peter Lake:8-Peter Lake:7   36.09850 -283.423597 355.62059 0.9999978
## Paul Lake:9-Peter Lake:7  -303.54877 -843.639684 236.54215 0.7209271
## Peter Lake:9-Peter Lake:7 -159.41677 -799.885807 481.05227 0.9983429
## Peter Lake:8-Paul Lake:8   409.47512   96.125963 722.82428 0.0020552
## Paul Lake:9-Paul Lake:8     69.82786 -466.634186 606.28990 0.9999924
## Peter Lake:9-Paul Lake:8   213.95986 -423.452032 851.37175 0.9849047
## Paul Lake:9-Peter Lake:8  -339.64727 -872.944314 193.64978 0.5579223
## Peter Lake:9-Peter Lake:8 -195.51527 -830.265716 439.23518 0.9917740
## Peter Lake:9-Paul Lake:9   144.13200 -625.615985 913.87999 0.9998333
```

```
Lake.Season.interaction <- with(NT.PeterPaul.SurfaceDepths.1993.1996, interaction(lakename, month))
Peter.Paul.anova.phosphorous <- aov(data = NT.PeterPaul.SurfaceDepths.1993.1996, tp_ug ~ Lake.Season.in
Peter.Paul.Phosphorous.groups <- HSD.test(Peter.Paul.anova.phosphorous, "Lake.Season.interaction", grou
Peter.Paul.Phosphorous.groups
```

```
## $statistics
##    MSerror  Df     Mean      CV
##   103.4055 119 19.07347 53.3141
##
## $parameters
##    test                  name.t ntr StudentizedRange alpha
##   Tukey Lake.Season.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"
```

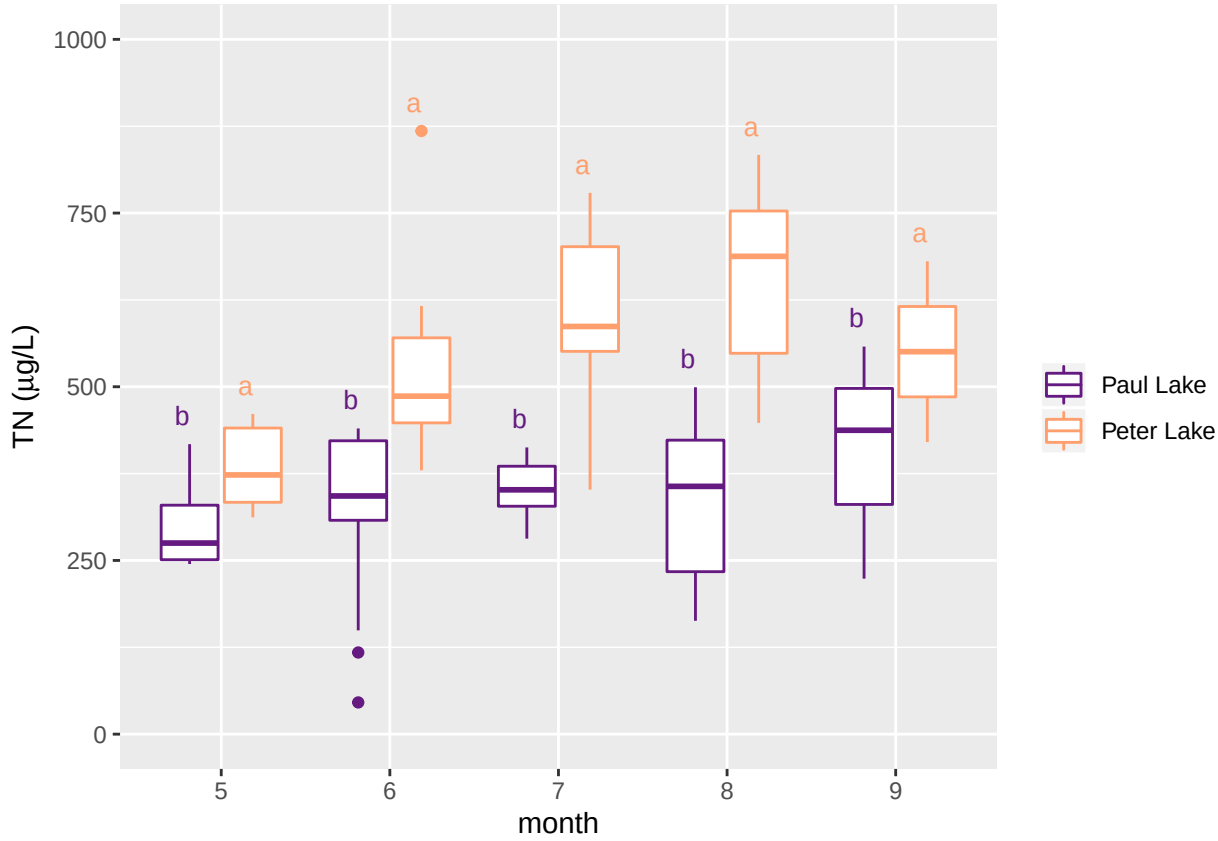
7. Create two plots, with TN (plot 1) or TP (plot 2) as the response variable and month and lake as the predictor variables. Hint: you may use some of the code you used for your visualization assignment. Assign groupings with letters, as determined from your tests. Adjust your axes, aesthetics, and color 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
tnplot <- ggplot(NT.PeterPaul.SurfaceDepths.1993.1996,  aes(x = as.factor(month), y = tn_ug, colour = la
  geom_boxplot() +
  labs(x = "month", y = expression(paste("TN (", mu, "g/L)")), colour = " ") +
  scale_color_viridis_d(option = "magma", begin = 0.3, end = 0.8) +
  ylim(0, 1000) +
  theme(legend.position = "right") +
    stat_summary(geom = "text", fun.y = max, vjust = -1, hjust = 1, size = 3.5,
              label = c("a", "b", "a", "b", "a", "b",
                        "a", "b", "a", "b"), position = position_dodge(0.75), show.legend = FALSE)
print(tnplot)
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```

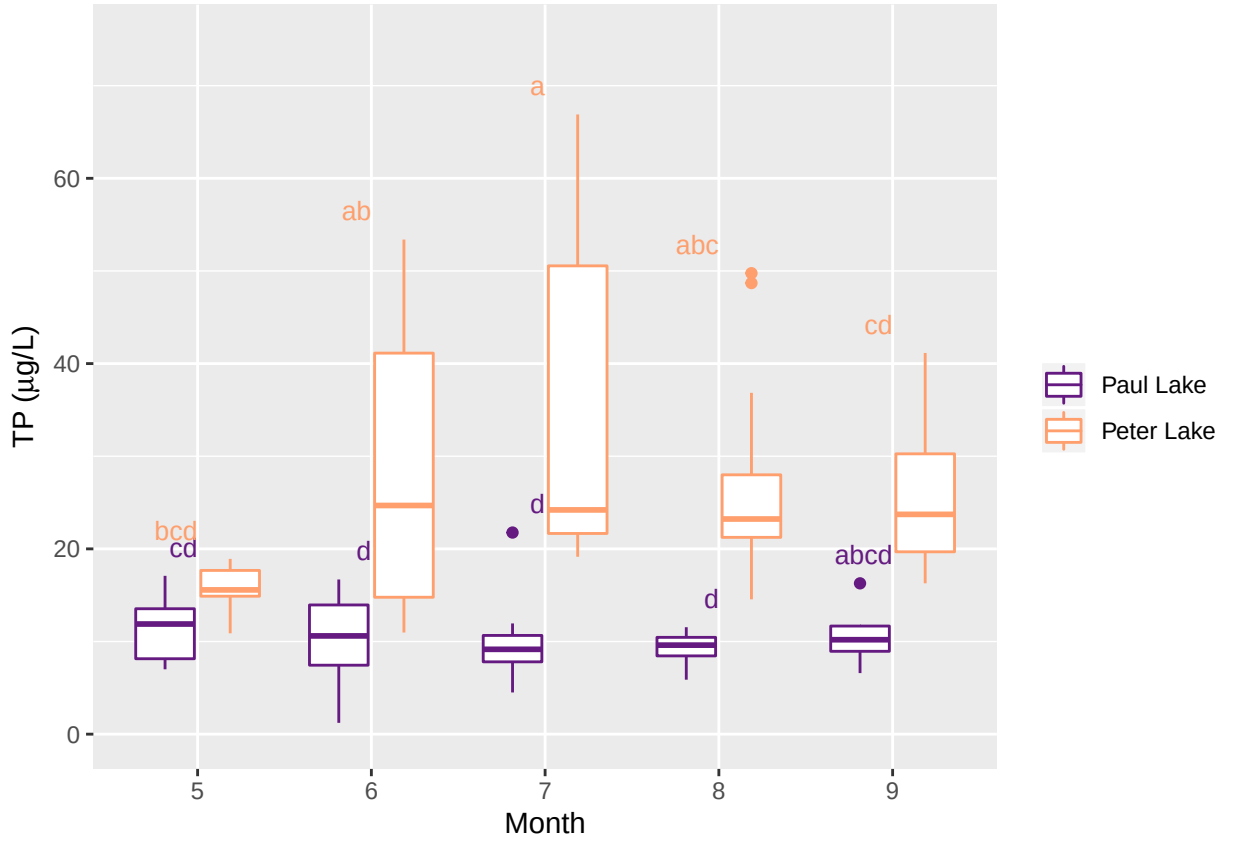
```
## Warning: Removed 26 rows containing non-finite values (stat_summary).
```




```
tpplot <- ggplot(NT.PeterPaul.SurfaceDepths.1993.1996,  aes(x = month, y = tp_ug, colour = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = expression(paste("TP (", mu, "g/L)")), colour = " ") +
  scale_color_viridis_d(option = "magma", begin = 0.3, end = 0.8) +
  ylim(0, 75) +
  theme(legend.position = "right") +
  stat_summary(geom = "text", fun.y = max, vjust = -1, hjust = 1, size = 3.5,
               label = c("cd", "bcd", "d", "ab", "d", "a", "d", "abc", "abcd", "cd"), show.legend = FALS
print(tpplot)
```

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

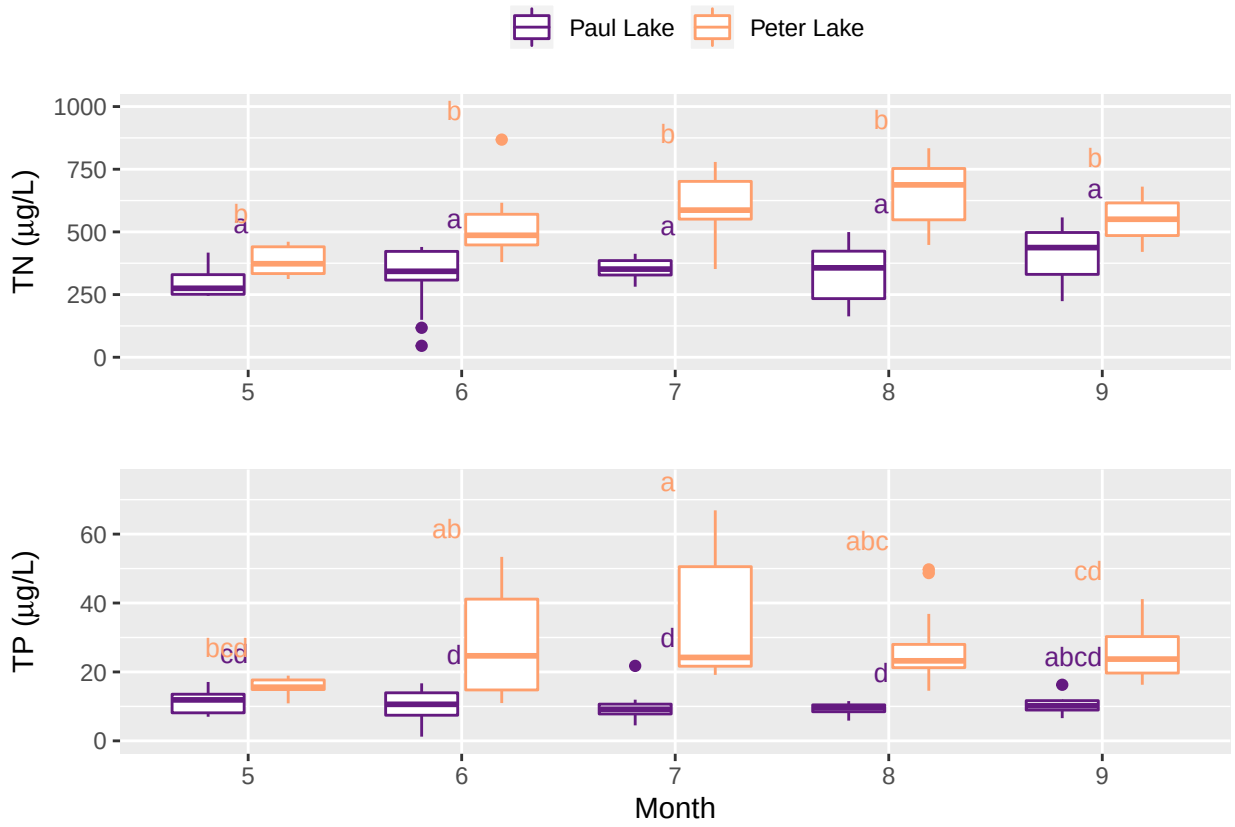
```
## Warning: Removed 1 rows containing non-finite values (stat_summary).
```




```
#8

tnplot2 <- ggplot(NT.PeterPaul.SurfaceDepths.1993.1996,  aes(x = as.factor(month), y = tn_ug, colour = l
  geom_boxplot() +
  labs(x = "", y = expression(paste("TN (", mu, "g/L)")), colour = " ") +
  scale_color_viridis_d(option = "magma", begin = 0.3, end = 0.8) +
  ylim(0, 1000) +
  theme(legend.position = "top") +
    stat_summary(geom = "text", fun.y = max, vjust = -1, hjust = 1, size = 3.5,
              label = c("a", "b", "a", "b", "a", "b",
                        "a", "b", "a", "b"), show.legend = FALSE)

tpplot2 <- ggplot(NT.PeterPaul.SurfaceDepths.1993.1996,  aes(x = month, y = tp_ug, colour = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = expression(paste("TP (", mu, "g/L)")), colour = " ") +
  scale_color_viridis_d(option = "magma", begin = 0.3, end = 0.8) +
  ylim(0, 75) +
  theme(legend.position = "none") +
  stat_summary(geom = "text", fun.y = max, vjust = -1, hjust = 1, size = 3.5,
               label = c("cd", "bcd", "d", "ab", "d", "a", "d", "abc", "abcd", "cd"), show.legend = FALS

plot_grid(tnplot2, tpplot2,  nrow = 2 , align = "v", rel_heights = c(1.25, 1, 1))
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
## Warning: Removed 26 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).
```




```
print(plot_grid)
```

```
## function (..., plotlist = NULL, align = c("none", "h", "v", "hv"),
##     axis = c("none", "l", "r", "t", "b", "lr", "tb", "tblr"),
##     nrow = NULL, ncol = NULL, rel_widths = 1, rel_heights = 1,
##     labels = NULL, label_size = 14, label_fontfamily = NULL,
##     label_fontface = "bold", label_colour = NULL, label_x = 0,
##     label_y = 1, hjust = -0.5, vjust = 1.5, scale = 1, greedy = TRUE,
##     cols = NULL, rows = NULL)
## {
##     plots <- c(list(...), plotlist)
##     num_plots <- length(plots)
##     if (!is.null(cols)) {
##         warning("Argument 'cols' is deprecated. Use 'ncol' instead.")
##     }
##     if (!is.null(rows)) {
##         warning("Argument 'rows' is deprecated. Use 'nrow' instead.")
##     }
##     scale <- rep_len(scale, num_plots)
##     if (sum(scale <= 0) > 1) {
##         stop("Argument 'scale' needs to be greater than 0.")
##     }
##     if (!is.null(ncol)) {
```

```
##         cols <- ncol
##     }
##     if (!is.null(nrow)) {
##         rows <- nrow
##     }
##     grobs <- align_plots(plotlist = plots, align = align, axis = axis,
##         greedy = greedy)
##     if (is.null(cols) && is.null(rows)) {
##         cols <- ceiling(sqrt(num_plots))
##         rows <- ceiling(num_plots/cols)
##     }
##     if (is.null(cols))
##         cols <- ceiling(num_plots/rows)
##     if (is.null(rows))
##         rows <- ceiling(num_plots/cols)
##     if ("AUTO" %in% labels)
##         labels <- LETTERS[1:num_plots]
##     else if ("auto" %in% labels)
##         labels <- letters[1:num_plots]
##     hjust <- rep_len(hjust, length(labels))
##     vjust <- rep_len(vjust, length(labels))
##     label_x <- rep_len(label_x, length(labels))
##     label_y <- rep_len(label_y, length(labels))
##     rel_heights <- rep(rel_heights, length.out = rows)
##     rel_widths <- rep(rel_widths, length.out = cols)
##     x_deltas <- rel_widths/sum(rel_widths)
##     y_deltas <- rel_heights/sum(rel_heights)
##     xs <- cumsum(rel_widths)/sum(rel_widths) - x_deltas
##     ys <- 1 - cumsum(rel_heights)/sum(rel_heights)
##     p <- ggdraw()
##     col_count <- 0
##     row_count <- 1
##     for (i in 1:(rows * cols)) {
##         if (i > num_plots)
##             break
##         x_delta <- x_deltas[col_count + 1]
##         y_delta <- y_deltas[row_count]
##         x <- xs[col_count + 1]
##         y <- ys[row_count]
##         p_next <- grobs[[i]]
##         if (!is.null(p_next)) {
##             p <- p + draw_grob(p_next, x, y, x_delta, y_delta,
##                 scale[i])
##         }
##         if (i <= length(labels)) {
##             p <- p + draw_plot_label(labels[i], x + label_x[i] *
##                 x_delta, y + label_y[i] * y_delta, size = label_size,
##                 family = label_fontfamily, fontface = label_fontface,
##                 colour = label_colour, hjust = hjust[i], vjust = vjust[i])
##         }
##         col_count <- col_count + 1
##         if (col_count >= cols) {
##             col_count <- 0
##             row_count <- row_count + 1
```

```
##         }
##     }
##     p
## }
## <bytecode: 0x7fab8d870ce8>
## <environment: namespace:cowplot>
```