Assignment 6: Generalized Linear Models

Walker Grimshaw

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on generalized linear models.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A06_GLMs.pdf") prior to submission.

The completed exercise is due on Tuesday, 26 February, 2019 before class begins.

Set up your session

1. Set up your session. Upload the EPA Ecotox dataset for Neonicotinoids and the NTL-LTER raw data file for chemistry/physics.
2. Build a ggplot theme and set it as your default theme.

```
#1
getwd()
```

```
## [1] "/Users/walkergrimshaw/Documents/Duke/Courses/Spring_2019/Environmental_Data_Analytics/Assignmen
```

```
suppressMessages(library(tidyverse))
library(lubridate) # easy date manipulation
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(gridExtra) # multiple plots in a figure
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(RColorBrewer)
library(FSA) # dunn.test after Kruskal Wallace
```

```
## ## FSA v0.8.22. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```
## EPA Ecotox
EPA.Ecotox.Raw <- read.csv("../Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv",
                           header = T)
## NTL-LTER
NTL.chem.phys.raw <- read.csv("../Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv",
                              header = T)

#2
WalkersTheme <- theme_bw(base_size = 12) +
  theme(legend.position = "top")

theme_set(WalkersTheme)
```

Neonicotinoids test

Research question: Were studies on various neonicotinoid chemicals conducted in different years?

3. Generate a line of code to determine how many different chemicals are listed in the Chemical.Name column.
4. Are the publication years associated with each chemical well-approximated by a normal distribution? Run the appropriate test and also generate a frequency polygon to illustrate the distribution of counts for each year, divided by chemical name. Bonus points if you can generate the results of your test from a pipe function. No need to make this graph pretty.
5. Is there equal variance among the publication years for each chemical? Hint: var.test is not the correct function.

```
#3
count.chemicals <- length(levels(EPA.Ecotox.Raw$Chemical.Name))
#4

## normality test

tapply(EPA.Ecotox.Raw$Pub..Year, EPA.Ecotox.Raw$Chemical.Name, FUN = shapiro.test)
```

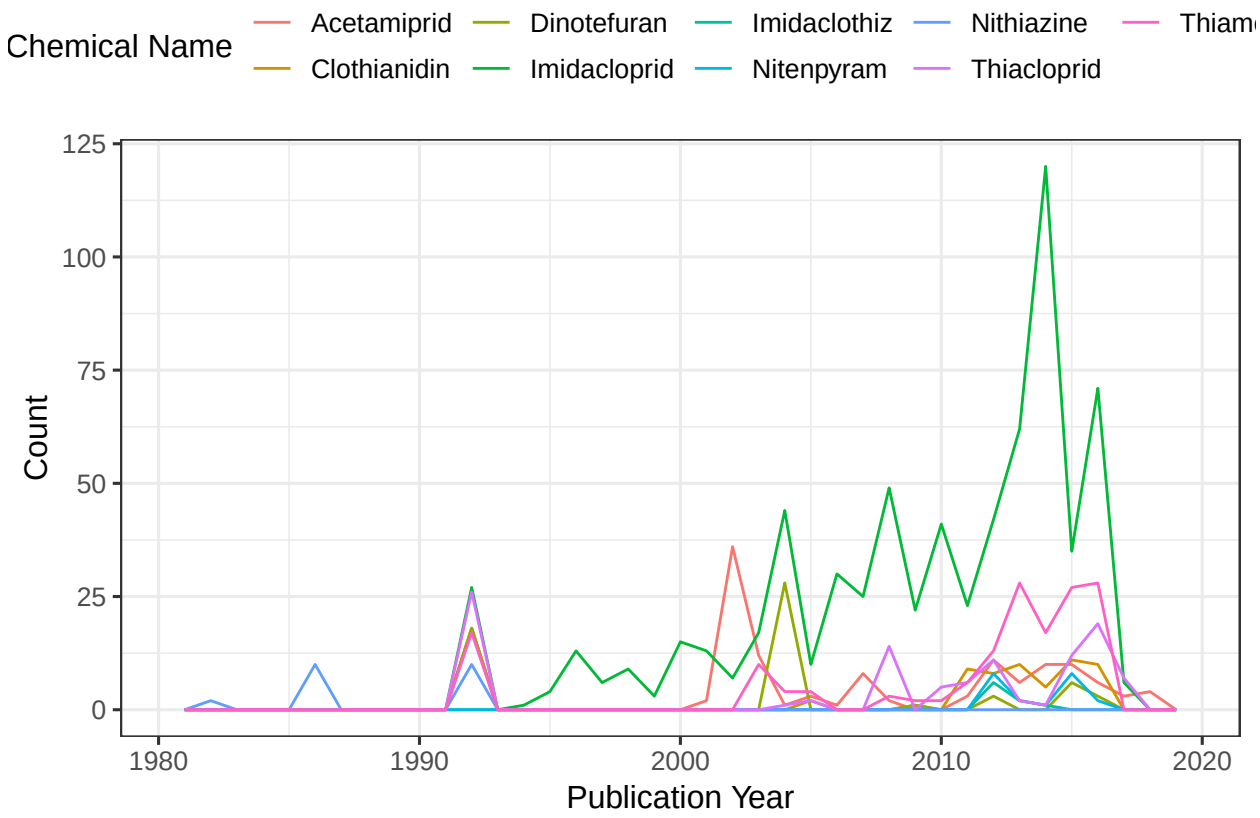
```
## $Acetamiprid
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.90191, p-value = 5.706e-08
##
##
## $Clothianidin
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
```

```
## W = 0.69577, p-value = 4.287e-11
##
##
## $Dinotefuran
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.82848, p-value = 8.83e-07
##
##
## $Imidacloprid
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.88178, p-value < 2.2e-16
##
##
## $Imidaclothiz
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.68429, p-value = 0.00093
##
##
## $Nitenpyram
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.79592, p-value = 0.0005686
##
##
## $Nithiazine
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.75938, p-value = 0.0001235
##
##
## $Thiacloprid
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.7669, p-value = 1.118e-11
##
##
## $Thiamethoxam
##
##  Shapiro-Wilk normality test
```

```
##
## data:  X[[i]]
## W = 0.7071, p-value < 2.2e-16
```

```
# Each chemical has a p-value less than 0.001, indicating non-normal distributions

## frequency polygon
ggplot(EPA.Ecotox.Raw, aes(color = Chemical.Name, x = Pub..Year)) +
  geom_freqpoly(binwidth = 1) +
  labs(x = "Publication Year", y = "Count", color = "Chemical Name")
```




```
#5
## bartlett test to compare variances among all groups
bartlett.test(EPA.Ecotox.Raw$Pub..Year ~ EPA.Ecotox.Raw$Chemical.Name)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  EPA.Ecotox.Raw$Pub..Year by EPA.Ecotox.Raw$Chemical.Name
## Bartlett's K-squared = 139.59, df = 8, p-value < 2.2e-16
```

```
## the variances are significantly different from one another
```

6. Based on your results, which test would you choose to run to answer your research question?

   ANSWER: Kruskal-Wallis Test, the non-parametric equivalent of ANOVA.

7. Run this test below.

8. Generate a boxplot representing the range of publication years for each chemical. Adjust your graph to make it pretty.

```r
#7
kruskal.test(EPA.Ecotox.Raw$Pub..Year ~ EPA.Ecotox.Raw$Chemical.Name)
```

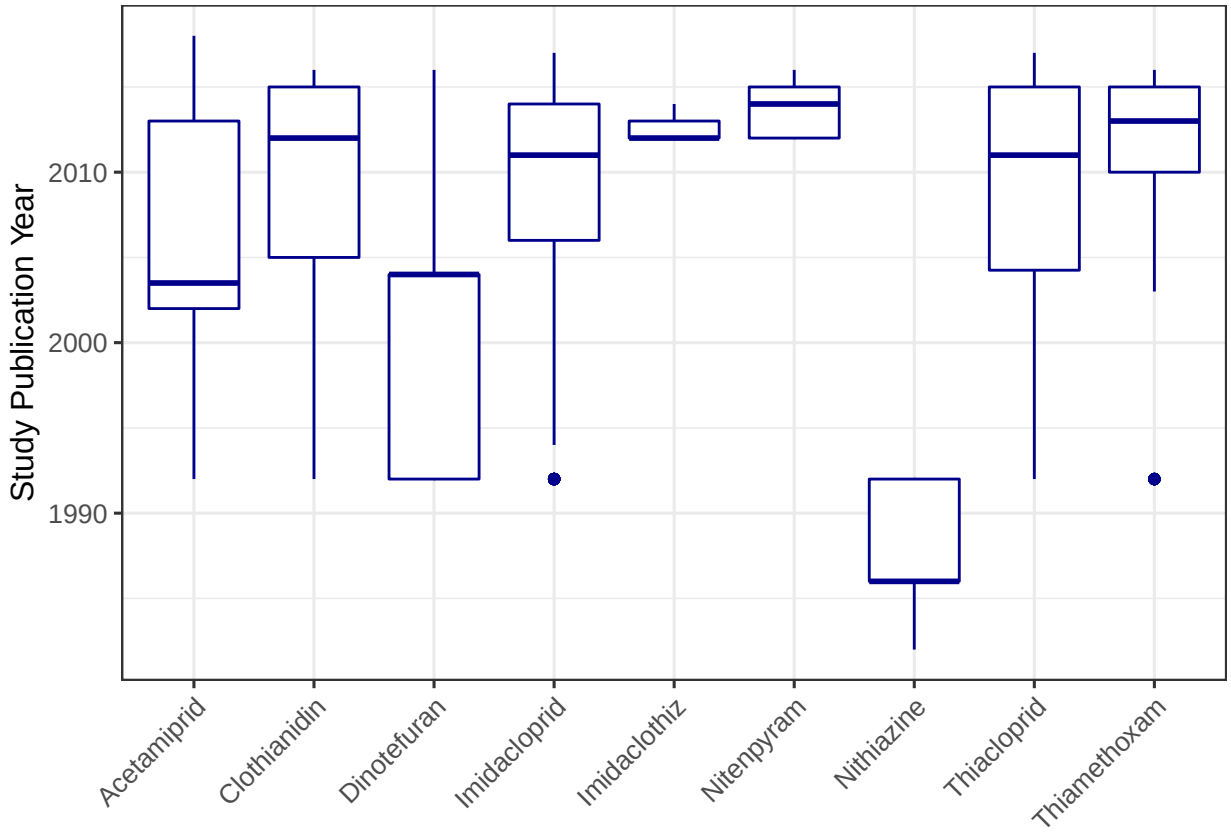
```
##
##  Kruskal-Wallis rank sum test
##
## data:  EPA.Ecotox.Raw$Pub..Year by EPA.Ecotox.Raw$Chemical.Name
## Kruskal-Wallis chi-squared = 134.15, df = 8, p-value < 2.2e-16
```

```r
## p-value << 0.01 indicates publication year does vary by chemical

## dunn test to compare publication years pairwise
dunnTest(EPA.Ecotox.Raw$Pub..Year, EPA.Ecotox.Raw$Chemical.Name)
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Holm method.
##                      Comparison          Z      P.unadj        P.adj
## 1   Acetamiprid - Clothianidin -3.0388079 2.375163e-03 4.037777e-02
## 2    Acetamiprid - Dinotefuran  2.1172089 3.424212e-02 4.109054e-01
## 3   Clothianidin - Dinotefuran  4.4060765 1.052598e-05 2.420975e-04
## 4   Acetamiprid - Imidacloprid -4.0204987 5.807507e-05 1.277651e-03
## 5  Clothianidin - Imidacloprid  0.5068899 6.122321e-01 1.000000e+00
## 6   Dinotefuran - Imidacloprid -5.2140290 1.847826e-07 4.989129e-06
## 7   Acetamiprid - Imidaclothiz -1.8052932 7.102881e-02 7.813169e-01
## 8  Clothianidin - Imidaclothiz -0.5166649 6.053901e-01 1.000000e+00
## 9   Dinotefuran - Imidaclothiz -2.6586494 7.845456e-03 1.176818e-01
## 10 Imidacloprid - Imidaclothiz -0.7284284 4.663514e-01 1.000000e+00
## 11    Acetamiprid - Nitenpyram -4.5018639 6.736012e-06 1.616643e-04
## 12   Clothianidin - Nitenpyram -2.4936264 1.264456e-02 1.770238e-01
## 13    Dinotefuran - Nitenpyram -5.4527796 4.958852e-08 1.388479e-06
## 14   Imidacloprid - Nitenpyram -3.0634837 2.187761e-03 3.937970e-02
## 15   Imidaclothiz - Nitenpyram -1.0897204 2.758363e-01 1.000000e+00
## 16    Acetamiprid - Nithiazine  5.6425299 1.675694e-08 4.859513e-07
## 17   Clothianidin - Nithiazine  7.1473251 8.848514e-13 2.831524e-11
## 18    Dinotefuran - Nithiazine  3.8693508 1.091255e-04 2.291636e-03
## 19   Imidacloprid - Nithiazine  7.7286349 1.087060e-14 3.804708e-13
## 20   Imidaclothiz - Nithiazine  4.8473136 1.251445e-06 3.253758e-05
## 21     Nitenpyram - Nithiazine  7.7099812 1.258363e-14 4.278434e-13
## 22   Acetamiprid - Thiacloprid -3.2225618 1.270497e-03 2.413945e-02
## 23  Clothianidin - Thiacloprid  0.1414916 8.874816e-01 8.874816e-01
## 24   Dinotefuran - Thiacloprid -4.6025295 4.173904e-06 1.043476e-04
## 25  Imidacloprid - Thiacloprid -0.3888712 6.973714e-01 1.000000e+00
## 26  Imidaclothiz - Thiacloprid  0.5870686 5.571576e-01 1.000000e+00
## 27    Nitenpyram - Thiacloprid  2.6709745 7.563140e-03 1.210102e-01
## 28    Nithiazine - Thiacloprid -7.3166886 2.541647e-13 8.387437e-12
## 29  Acetamiprid - Thiamethoxam -5.8898861 3.864618e-09 1.159385e-07
## 30 Clothianidin - Thiamethoxam -1.7587256 7.862413e-02 7.862413e-01
## 31  Dinotefuran - Thiamethoxam -6.6762123 2.451967e-11 7.601098e-10
## 32 Imidacloprid - Thiamethoxam -3.5327039 4.113329e-04 8.226657e-03
## 33 Imidaclothiz - Thiamethoxam -0.1886278 8.503846e-01 1.000000e+00
## 34   Nitenpyram - Thiamethoxam  1.5927766 1.112103e-01 1.000000e+00
## 35   Nithiazine - Thiamethoxam -8.7224129 2.723352e-18 9.804067e-17
## 36  Thiacloprid - Thiamethoxam -2.1461156 3.186376e-02 4.142288e-01
```

```r
#8
ggplot(EPA.Ecotox.Raw, aes(x = Chemical.Name, y = Pub..Year)) +
  geom_boxplot(color = "darkblue") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(y = "Study Publication Year", x = NULL)
```



9. How would you summarize the conclusion of your analysis? Include a sentence summarizing your findings and include the results of your test in parentheses at the end of the sentence.

   ANSWER: Studies were conducted in different years for different neonicotinoid chemicals (Kruskal-Wallace; chi-squared = 134.15, df = 8, $p < 0.001$).

NTL-LTER test

Research question: What is the best set of predictors for lake temperatures in July across the monitoring period at the North Temperate Lakes LTER?

11. Wrangle your NTL-LTER dataset with a pipe function so that it contains only the following criteria:

- Only dates in July (hint: use the daynum column). No need to consider leap years.
- Only the columns: lakename, year4, daynum, depth, temperature_C
- Only complete cases (i.e., remove NAs)

12. Run an AIC to determine what set of explanatory variables (year4, daynum, depth) is best suited to predict temperature. Run a multiple regression on the recommended set of variables.

```r
#11
NTL.July <- NTL.chem.phys.raw %>%
  filter(daynum >= 182 & daynum <= 212) %>%
```

```
  select(lakename, year4, daynum, depth, temperature_C) %>%
  na.omit
#12
# AIC
temp.AIC <- lm(data = NTL.July, temperature_C ~ year4 + daynum + depth)
step(temp.AIC)
```

```
## Start:  AIC=26016.31
## temperature_C ~ year4 + daynum + depth
##
##          Df Sum of Sq    RSS   AIC
## <none>                141118 26016
## - year4   1        80 141198 26020
## - daynum  1      1333 142450 26106
## - depth   1    403925 545042 39151
```

```
##
## Call:
## lm(formula = temperature_C ~ year4 + daynum + depth, data = NTL.July)
##
## Coefficients:
## (Intercept)        year4       daynum        depth
##    -6.45556      0.01013      0.04134     -1.94726
```

```
## the full model is the minimum adequate model
temp.lm <- temp.AIC
```

13. What is the final linear equation to predict temperature from your multiple regression? How much of the observed variance does this model explain?

    ANSWER: Temperature = -6.46 + 0.01(year) + 0.04(day number) - 1.95(depth) This model explains 74% of the observed variance.

14. Run an interaction effects ANCOVA to predict temperature based on depth and lakename from the same wrangled dataset.

```
#14
temp.interaction <- lm(data = NTL.July, temperature_C ~ depth*lakename)
summary(temp.interaction)
```

```
##
## Call:
## lm(formula = temperature_C ~ depth * lakename, data = NTL.July)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6455 -2.9133 -0.2879  2.7567 16.3606
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     22.9455     0.5861  39.147  < 2e-16 ***
## depth                           -2.5820     0.2411 -10.711  < 2e-16 ***
## lakenameCrampton Lake            2.2173     0.6804   3.259  0.00112 **
## lakenameEast Long Lake          -4.3884     0.6191  -7.089 1.45e-12 ***
## lakenameHummingbird Lake        -2.4126     0.8379  -2.879  0.00399 **
## lakenamePaul Lake                0.6105     0.5983   1.020  0.30754
## lakenamePeter Lake               0.2998     0.5970   0.502  0.61552
```

```
## lakenameTuesday Lake            -2.8932     0.6060  -4.774 1.83e-06 ***
## lakenameWard Lake                2.4180     0.8434   2.867  0.00415 **
## lakenameWest Long Lake          -2.4663     0.6168  -3.999 6.42e-05 ***
## depth:lakenameCrampton Lake      0.8058     0.2465   3.268  0.00109 **
## depth:lakenameEast Long Lake     0.9465     0.2433   3.891  0.00010 ***
## depth:lakenameHummingbird Lake  -0.6026     0.2919  -2.064  0.03903 *
## depth:lakenamePaul Lake          0.4022     0.2421   1.662  0.09664 .
## depth:lakenamePeter Lake         0.5799     0.2418   2.398  0.01649 *
## depth:lakenameTuesday Lake       0.6605     0.2426   2.723  0.00648 **
## depth:lakenameWard Lake         -0.6930     0.2862  -2.421  0.01548 *
## depth:lakenameWest Long Lake     0.8154     0.2431   3.354  0.00080 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.471 on 9704 degrees of freedom
## Multiple R-squared:  0.7861, Adjusted R-squared:  0.7857
## F-statistic:  2097 on 17 and 9704 DF,  p-value: < 2.2e-16
```

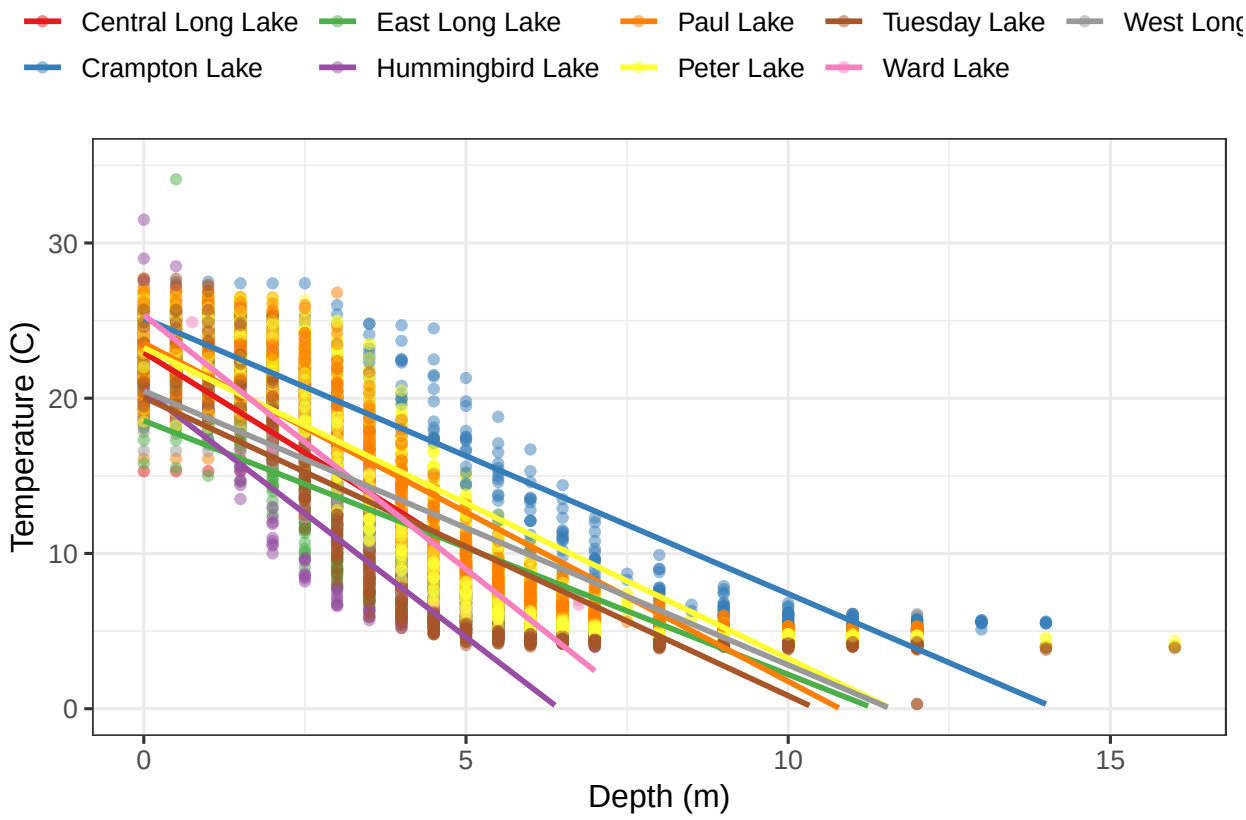
15. Is there an interaction between depth and lakename? How much variance in the temperature observations does this explain?

    ANSWER: There is an interaction between depth and lakename in predicting temperature. Together, depth and lakename explain 78.6% of temperature variance.

16. Create a graph that depicts temperature by depth, with a separate color for each lake. Add a geom_smooth (method = "lm", se = FALSE) for each lake. Make your points 50 % transparent. Adjust your y axis limits to go from 0 to 35 degrees. Clean up your graph to make it pretty.

```
#16
ggplot(NTL.July, aes(x = depth, y = temperature_C, color = lakename)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  ylim(0,35) +
  scale_color_brewer(palette = "Set1") +
  labs(x = "Depth (m)", y = "Temperature (C)", color = NULL)
```

```
## Warning: Removed 73 rows containing missing values (geom_smooth).
```


Central Long Lake
East Long Lake
Paul Lake
Tuesday Lake
West Long
Crampton Lake
Hummingbird Lake
Peter Lake
Ward Lake
Temperature (C)
30
20
10
0
0
5
10
15
Depth (m)