Assignment 6: Generalized Linear Models

Laurie Muzzy

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
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2
```

```
## -- Attaching packages ----------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0       v purrr   0.3.0
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.2       v stringr 1.3.1
## v readr   1.3.1       v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
## Warning: package 'tibble' was built under R version 3.4.4
```

```
## Warning: package 'tidyr' was built under R version 3.4.4
```

```
## Warning: package 'readr' was built under R version 3.4.4
```

```
## Warning: package 'purrr' was built under R version 3.4.4
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
## Warning: package 'stringr' was built under R version 3.4.4
```

```
## Warning: package 'forcats' was built under R version 3.4.3
```

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

```
getwd()
```

```
## [1] "/Users/laurie/Desktop/Envtl_Data_Analytics/MuzzyGitFile"
```

```
ECOTOX_Neonic <- read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv", header = TRUE) #header
library(readr)
NTL_LTER_Lake_ChemistryPhysics_Raw <- read_csv("Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv")
```

```
## Parsed with column specification:
## cols(
##   lakeid = col_character(),
##   lakename = col_character(),
##   year4 = col_double(),
##   daynum = col_double(),
##   sampledate = col_character(),
##   depth = col_double(),
##   temperature_C = col_double(),
##   dissolvedOxygen = col_double(),
##   irradianceWater = col_double(),
##   irradianceDeck = col_double(),
##   comments = col_logical()
## )
```

```
## Warning: 368 parsing failures.
##   row      col           expected                              actual
## 36649 comments 1/0/T/F/TRUE/FALSE DO Probe bad - Doesn't go to zero 'Data/Raw/NTL-LTER_Lake_Chemistry
## 36651 comments 1/0/T/F/TRUE/FALSE DO Probe bad - Doesn't go to zero 'Data/Raw/NTL-LTER_Lake_Chemistry
## 36653 comments 1/0/T/F/TRUE/FALSE DO Probe bad - Doesn't go to zero 'Data/Raw/NTL-LTER_Lake_Chemistry
## 36654 comments 1/0/T/F/TRUE/FALSE DO Probe bad - Doesn't go to zero 'Data/Raw/NTL-LTER_Lake_Chemistry
## 36655 comments 1/0/T/F/TRUE/FALSE DO Probe bad - Doesn't go to zero 'Data/Raw/NTL-LTER_Lake_Chemistry
## ..... ........ .................. ................................. .............................
## See problems(...) for more details.
```

```
#NTL-LTER_Lake_ChemistryPhysics_Raw <- read.csv("./Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv")

#2
A6theme <- theme_gray(base_size = 13)
theme(axis.text = element_text(color = "black"), legend.position = "right")
```

```
## List of 2
##  $ axis.text      :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "black"
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
```

```
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.position: chr "right"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

```
theme_set(A6theme)
```

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 how many chemicals are listed
summary(ECOTOX_Neonic$Chemical.Name) #(9 chemicals)
```

```
##  Acetamiprid Clothianidin  Dinotefuran Imidacloprid Imidaclothiz
##          136           74           59          695            9
##   Nitenpyram   Nithiazine  Thiacloprid Thiamethoxam
##           21           22          106          161
```

```
class(ECOTOX_Neonic$Pub..Year) #integer
```

```
## [1] "integer"
```

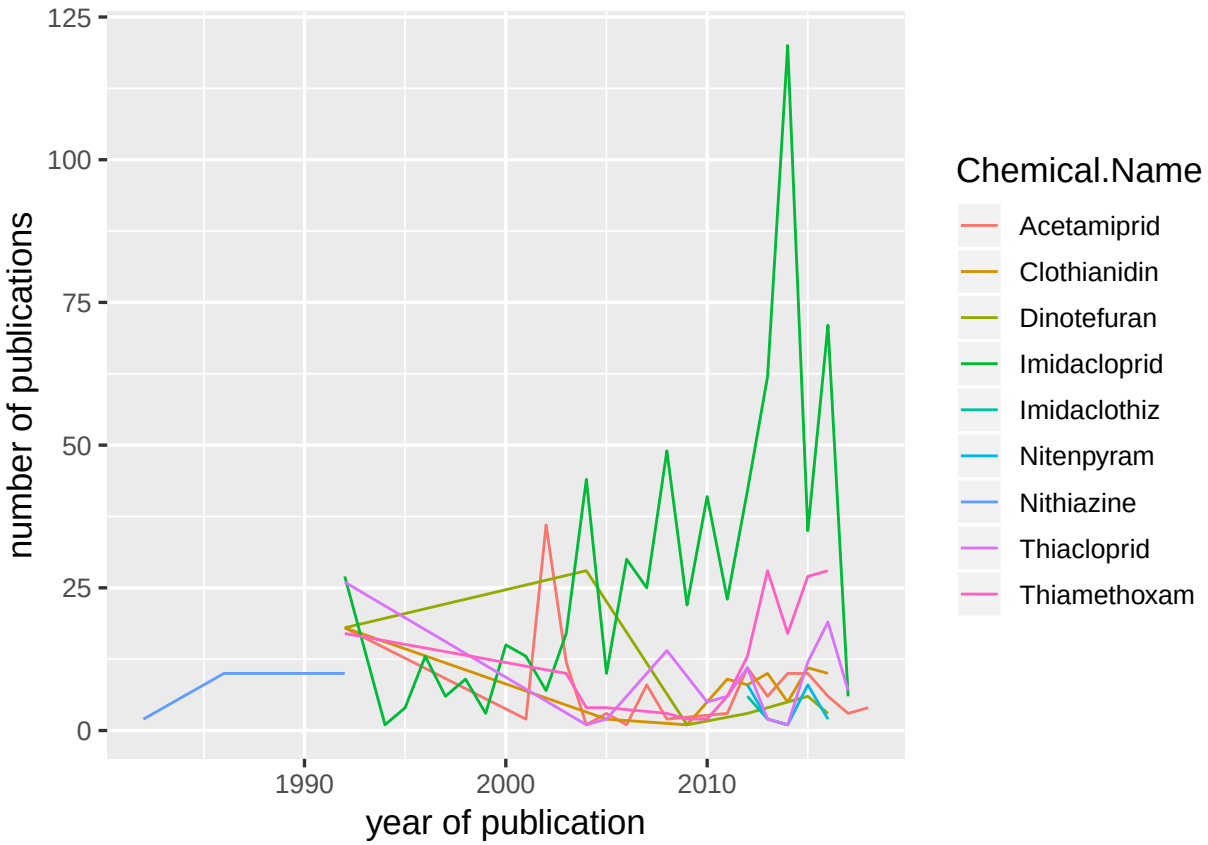
```
#4  see if it's a normal distr
#not numeric, so need other test like ANOVA

Chem.Name <- function(N) {ECOTOX_Neonic %>%
    filter(Chemical.Name == 'Acetamiprid') %>%
    pull(Pub..Year) %>%
    shapiro.test()
}

Chem.Name
```

```
## function(N) {ECOTOX_Neonic %>%
##     filter(Chemical.Name == 'Acetamiprid') %>%
##     pull(Pub..Year) %>%
##     shapiro.test()
## }
```

```
Ecotox.PubYr.norm <- ggplot(ECOTOX_Neonic) +
geom_freqpoly(aes(x = Pub..Year, color = Chemical.Name), stat = "count") +
  labs(x = "year of publication", y = "number of publications")
print(Ecotox.PubYr.norm)
```




```
#5 equal var in pub yrs for each chemical?

bartlett.test(ECOTOX_Neonic$Pub..Year ~ ECOTOX_Neonic$Chemical.Name)
```

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

```
#Bartlett's K-squared = 139.59, df = 8, p-value < 2.2e-16

# p <0.0001, so we can reject the null; the variance is not the same for all the chemicals.
```

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

   ANSWER: "Were studies on various neonicotinoid chemicals conducted in different years?" Kruskal-Wallis test, because it compares multiple groups and it's nonparametric.

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.

```
#7 test for studies of dif chemicals performed in dif years
#response ~ explanatory
range(ECOTOX_Neonic$Pub..Year)
```

```
## [1] 1982 2018
```

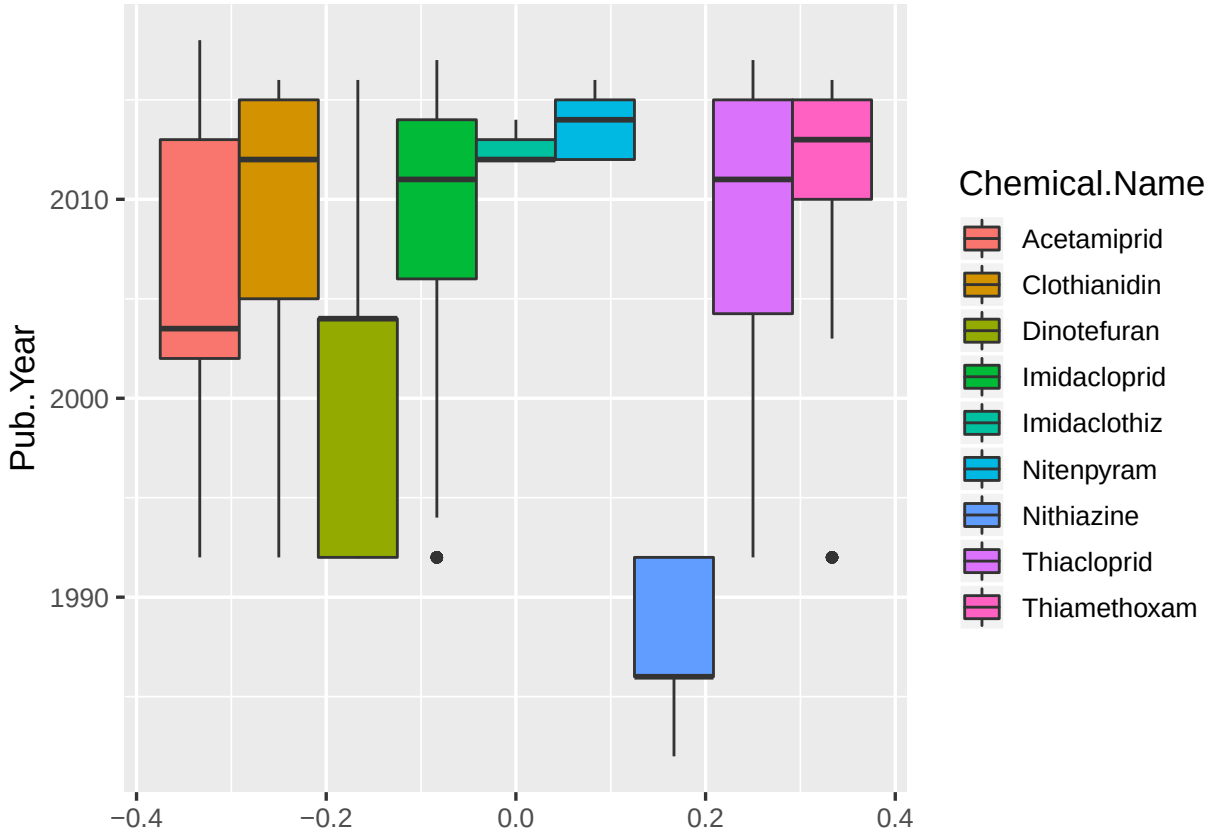
```
summary(ECOTOX_Neonic$Chemical.Name)
```

```
##  Acetamiprid Clothianidin  Dinotefuran Imidacloprid Imidaclothiz
##          136           74           59          695            9
##   Nitenpyram   Nithiazine  Thiacloprid Thiamethoxam
##           21           22          106          161
```

```
Chem.PubYr.kruskal <- kruskal.test(Pub..Year ~ Chemical.Name, ECOTOX_Neonic)
Chem.PubYr.kruskal #Kruskal-Wallis chi-squared = 134.15, df = 8, p-value < 2.2e-16
```

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

```
#8 boxplot of range of pub years for each chemical
#not informative enough: need better x axis, can't figure out units or numbers
Ecotox.PubYr.Chemicals <- ggplot(ECOTOX_Neonic, aes(stat = "count", y = Pub..Year )) +
geom_boxplot(aes(fill = Chemical.Name), position = "dodge") +
#labs(x = "number of publications", y = "publication year", title = "Publications on Neonicotinoids, 19
theme(legend.position = "right")
print(Ecotox.PubYr.Chemicals)
```



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: Used Kruskal test, p-val <0.05, indicating significant difference between the amount of publications for the different chemicals. (results: Kruskal-Wallis chi-squared = 134.15, df = 8,

p-value < 2.2e-16)

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.

```
#11 : dates in July: lakename, year4, daynum, depth, temperature_C, remove NAs (na.omit but only after
Lake.July.temps <- NTL_LTER_Lake_ChemistryPhysics_Raw %>%
filter(daynum >= 182 & daynum <= 212) %>%
select(lakename, year4, daynum, depth, temperature_C) %>%
na.omit()

#12 AIC

#Correlations close to -1 represent strong negative correlations, correlations close to zero represent
Lake.July.temps.AIC <- lm(data = Lake.July.temps, temperature_C ~ depth + daynum + year4)
step(Lake.July.temps.AIC)
```

```
## Start:  AIC=26016.31
## temperature_C ~ depth + daynum + year4
##
##          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 ~ depth + daynum + year4, data = Lake.July.temps)
##
## Coefficients:
## (Intercept)        depth       daynum        year4
##    -6.45556     -1.94726      0.04134      0.01013
```

```
Lake.July.temps.model <- lm(data = Lake.July.temps, temperature_C ~ year4 + daynum)
step(Lake.July.temps.model)
```

```
## Start:  AIC=39151.36
## temperature_C ~ year4 + daynum
##
##          Df Sum of Sq    RSS   AIC
## - year4   1      3.33 545046 39149
## <none>                545042 39151
## - daynum  1   1355.90 546398 39174
##
## Step:  AIC=39149.42
```

```
## temperature_C ~ daynum
##
##          Df Sum of Sq    RSS   AIC
## <none>                545046 39149
## - daynum  1    1356.6 546402 39172
```

```
##
## Call:
## lm(formula = temperature_C ~ daynum, data = Lake.July.temps)
##
## Coefficients:
## (Intercept)       daynum
##      4.4786       0.0417
```

```
summary(Lake.July.temps.model) #Residual standard error: 7.489 on 9719 degrees of freedom Multiple R-sq
```

```
##
## Call:
## lm(formula = temperature_C ~ year4 + daynum, data = Lake.July.temps)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.289  -7.138  -2.601   8.061  21.408
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.363637  16.976619   0.021    0.983
## year4        0.002060   0.008456   0.244    0.808
## daynum       0.041693   0.008479   4.917 8.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.489 on 9719 degrees of freedom
## Multiple R-squared:  0.002489,   Adjusted R-squared:  0.002284
## F-statistic: 12.13 on 2 and 9719 DF,  p-value: 5.503e-06
```

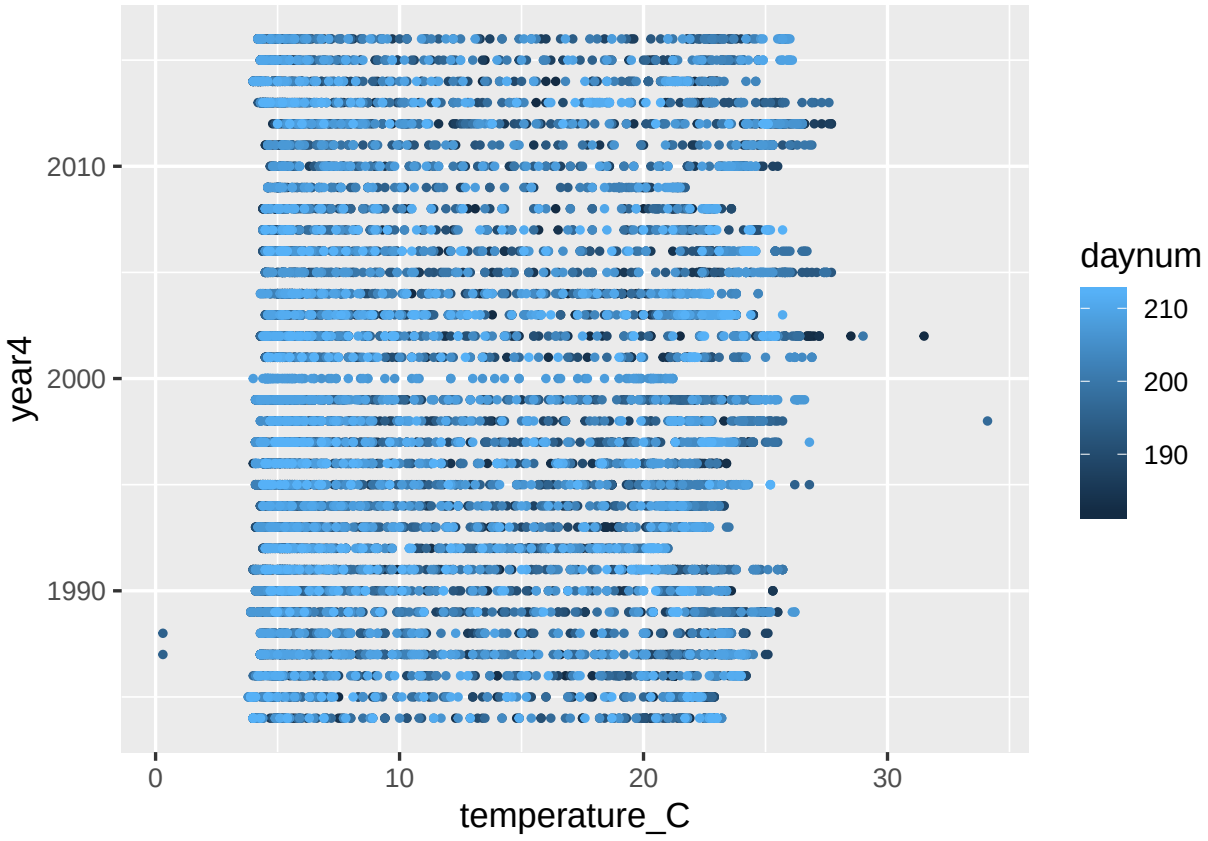
```
#weak correlation: only 0.2% of variance is accounted for by explan var

Lake.July.temps.regression <- lm(data = Lake.July.temps, temperature_C ~ year4 + daynum)
summary(Lake.July.temps.regression)
```

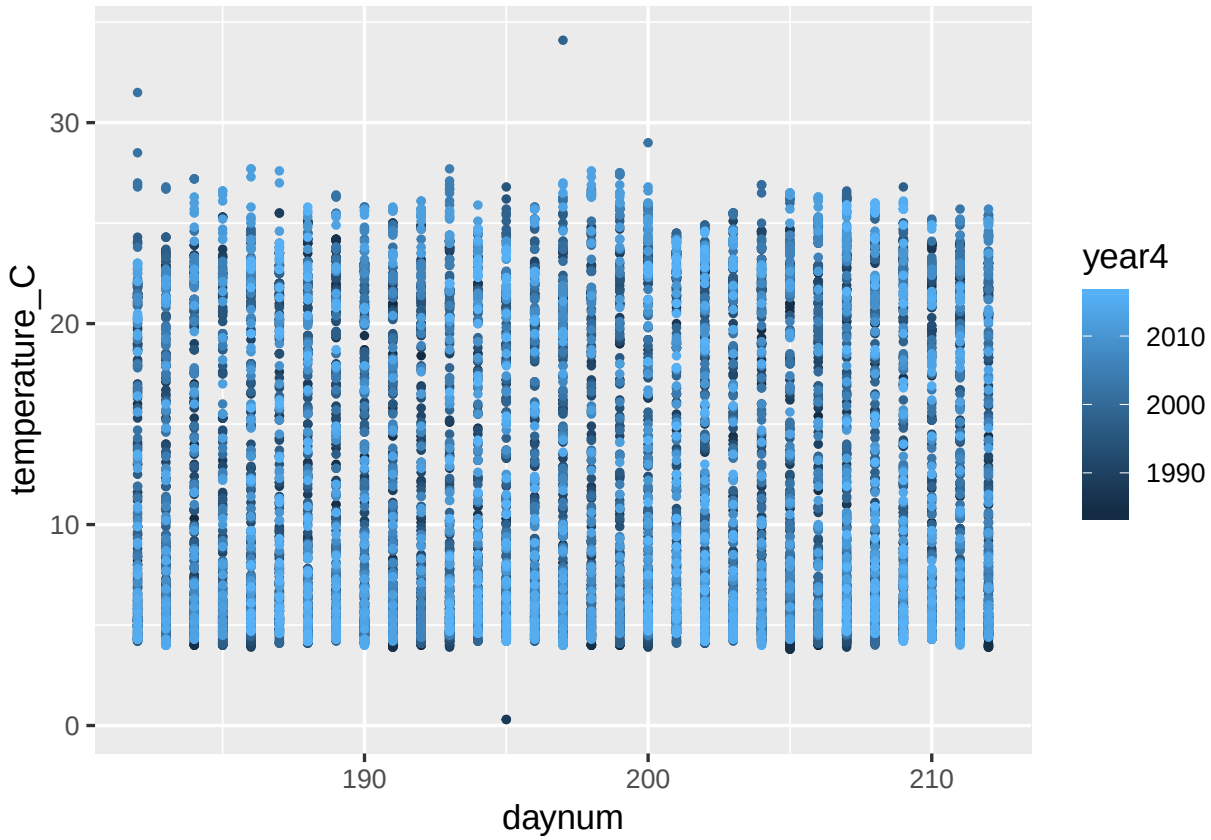
```
##
## Call:
## lm(formula = temperature_C ~ year4 + daynum, data = Lake.July.temps)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.289  -7.138  -2.601   8.061  21.408
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.363637  16.976619   0.021    0.983
## year4        0.002060   0.008456   0.244    0.808
## daynum       0.041693   0.008479   4.917 8.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.489 on 9719 degrees of freedom
## Multiple R-squared:  0.002489,   Adjusted R-squared:  0.002284
## F-statistic: 12.13 on 2 and 9719 DF,  p-value: 5.503e-06
```

```
Lake.July.temps.plot1 <- ggplot(Lake.July.temps,
                aes(x = temperature_C, y = year4, color = daynum)) +
  geom_point(size = 1)
print(Lake.July.temps.plot1)
```




```
Lake.July.temps.plot2 <- ggplot(Lake.July.temps,
                aes(x = daynum, y = temperature_C, color = year4)) +
  geom_point(size = 1)
print(Lake.July.temps.plot2)
```



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_C = 0.36 + 0.002(year4) + 0.04(daynum) + 16.9(E). This model only explains 0.2% of the variance, which is terrible.

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

```
#14 lm

Lake.July.temps.ancova <- lm(data = Lake.July.temps, temperature_C ~ lakename + depth)
summary(Lake.July.temps.ancova)
```

```
##
## Call:
## lm(formula = temperature_C ~ lakename + depth, data = Lake.July.temps)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1127 -3.0040 -0.2316  2.8312 15.1985
##
## Coefficients:
##                            Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                21.68826    0.32512   66.709  < 2e-16 ***
## lakenameCrampton Lake       4.52447    0.38213   11.840  < 2e-16 ***
## lakenameEast Long Lake     -1.45418    0.34530   -4.211 2.56e-05 ***
## lakenameHummingbird Lake -4.88905    0.46179  -10.587  < 2e-16 ***
## lakenamePaul Lake           0.91157    0.33264    2.740  0.00615 **
```

```
## lakenamePeter Lake        1.37937    0.33250    4.148 3.38e-05 ***
## lakenameTuesday Lake     -1.42651    0.33815   -4.219 2.48e-05 ***
## lakenameWard Lake        -0.68248    0.46187   -1.478  0.13954
## lakenameWest Long Lake   -0.20353    0.34392   -0.592  0.55400
## depth                    -1.96627    0.01095 -179.552  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.538 on 9712 degrees of freedom
## Multiple R-squared:  0.7775, Adjusted R-squared:  0.7773
## F-statistic:  3770 on 9 and 9712 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?

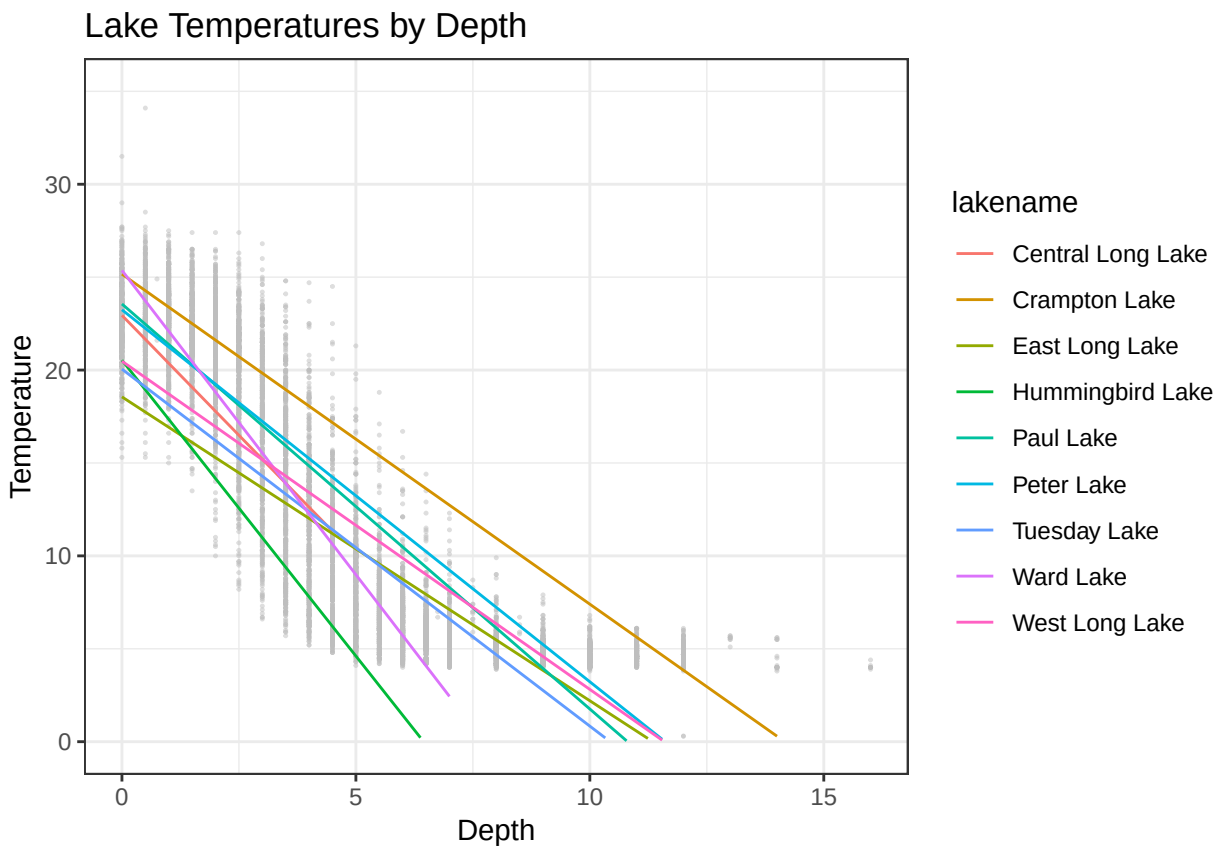
    ANSWER: There appears to be an interaction between depth and lakename (which makes sense: lakes are probably going to have different depths). It explains about 78% of the variance.

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 x=depth y=temp ?

Lakes.Temp.by.depth <- ggplot(Lake.July.temps, aes(x = depth, y = temperature_C), color = depth) +
  theme_bw() +
  geom_point(alpha = 0.5, size = 0.2, color = "gray") +
  ylim(0,35) +
  geom_smooth(aes(color = lakename), method = "lm", se = FALSE, size = 0.5) +
labs(x = "Depth", y = "Temperature", title = "Lake Temperatures by Depth")
print(Lakes.Temp.by.depth)
```

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


Lake Temperatures by Depth
Temperature
Depth
0
10
20
30
0
5
10
15
lakename
Central Long Lake
Crampton Lake
East Long Lake
Hummingbird Lake
Paul Lake
Peter Lake
Tuesday Lake
Ward Lake
West Long Lake