Assignment 6: Generalized Linear Models

Rachel Bash

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/rachelbash/Documents/DUKE/Data Analytics/Environmental_Data_Analytics/Assignments"
```

```
suppressMessages(library(tidyverse))
suppressMessages(library(viridis))
suppressMessages(library(RColorBrewer))
suppressMessages(library(gridExtra))

Ecotox <- read.csv("../Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv")
ChemPhysics <- read.csv("../Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv")

#2
Rachel_theme <- theme_bw(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
theme_set(Rachel_theme)
```

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
nlevels(Ecotox$Chemical.Name)
```

```
## [1] 9
```

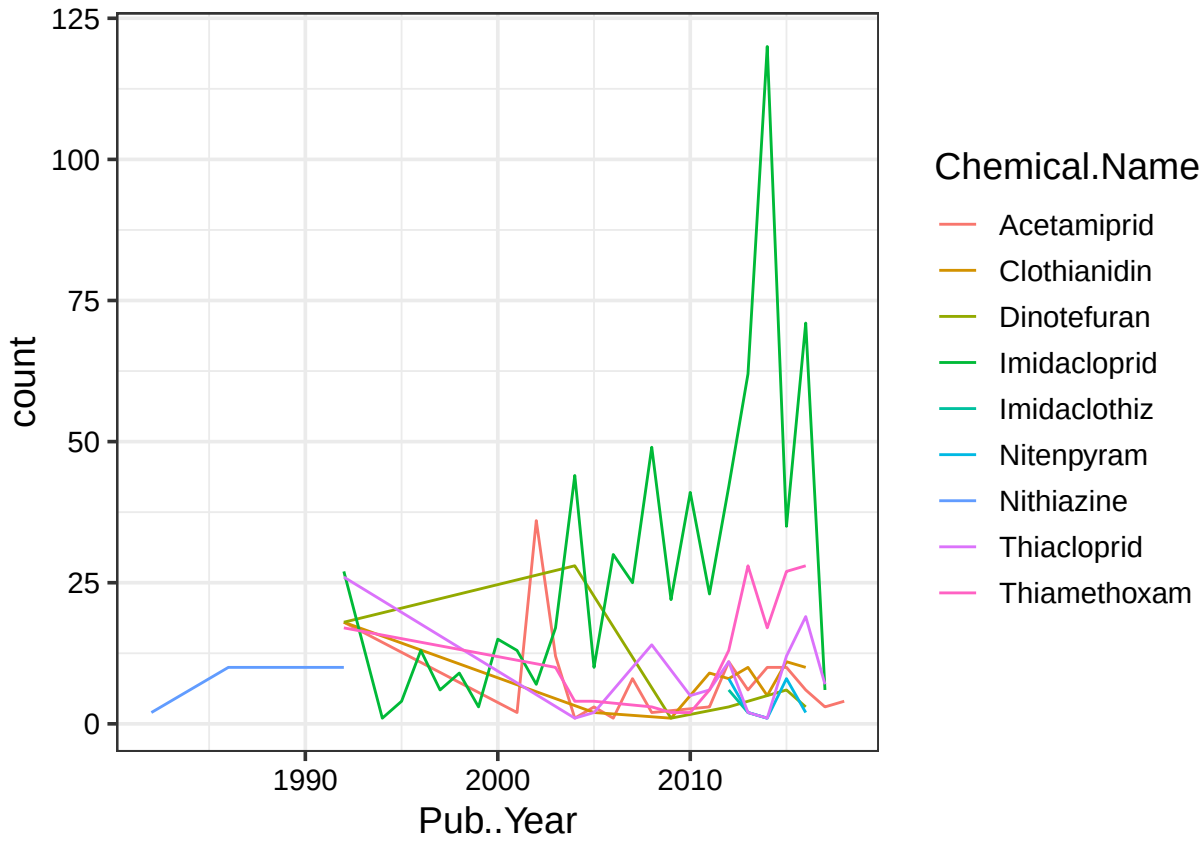
```
levels(Ecotox$Chemical.Name)
```

```
## [1] "Acetamiprid"  "Clothianidin" "Dinotefuran"  "Imidacloprid"
## [5] "Imidaclothiz" "Nitenpyram"   "Nithiazine"   "Thiacloprid"
## [9] "Thiamethoxam"
```

```
#4
Shap.test <- Ecotox %>%
  group_by(Chemical.Name) %>%
  summarise(
  statistic = shapiro.test(Pub..Year)$statistic,
  p.value = shapiro.test(Pub..Year)$p.value)
print(Shap.test)
```

```
## # A tibble: 9 x 3
##   Chemical.Name statistic  p.value
##   <fct>             <dbl>    <dbl>
## 1 Acetamiprid       0.902 5.71e- 8
## 2 Clothianidin      0.696 4.29e-11
## 3 Dinotefuran       0.828 8.83e- 7
## 4 Imidacloprid      0.882 1.38e-22
## 5 Imidaclothiz      0.684 9.30e- 4
## 6 Nitenpyram        0.796 5.69e- 4
## 7 Nithiazine        0.759 1.24e- 4
## 8 Thiacloprid       0.767 1.12e-11
## 9 Thiamethoxam      0.707 1.57e-16
```

```
ggplot(Ecotox, aes(x = Pub..Year, color = Chemical.Name)) +
  geom_freqpoly(stat = "count")
```




```
#as seen by the p-values and the graph, it is clear that none of these are normally distributed!


#5
bartlett.test(Ecotox$Pub..Year ~ Ecotox$Chemical.Name)
```

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

```
#no equal variance
```

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

   ANSWER: Non-parametric equivalent of Anova, the Kruskal-Wallis test.

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.

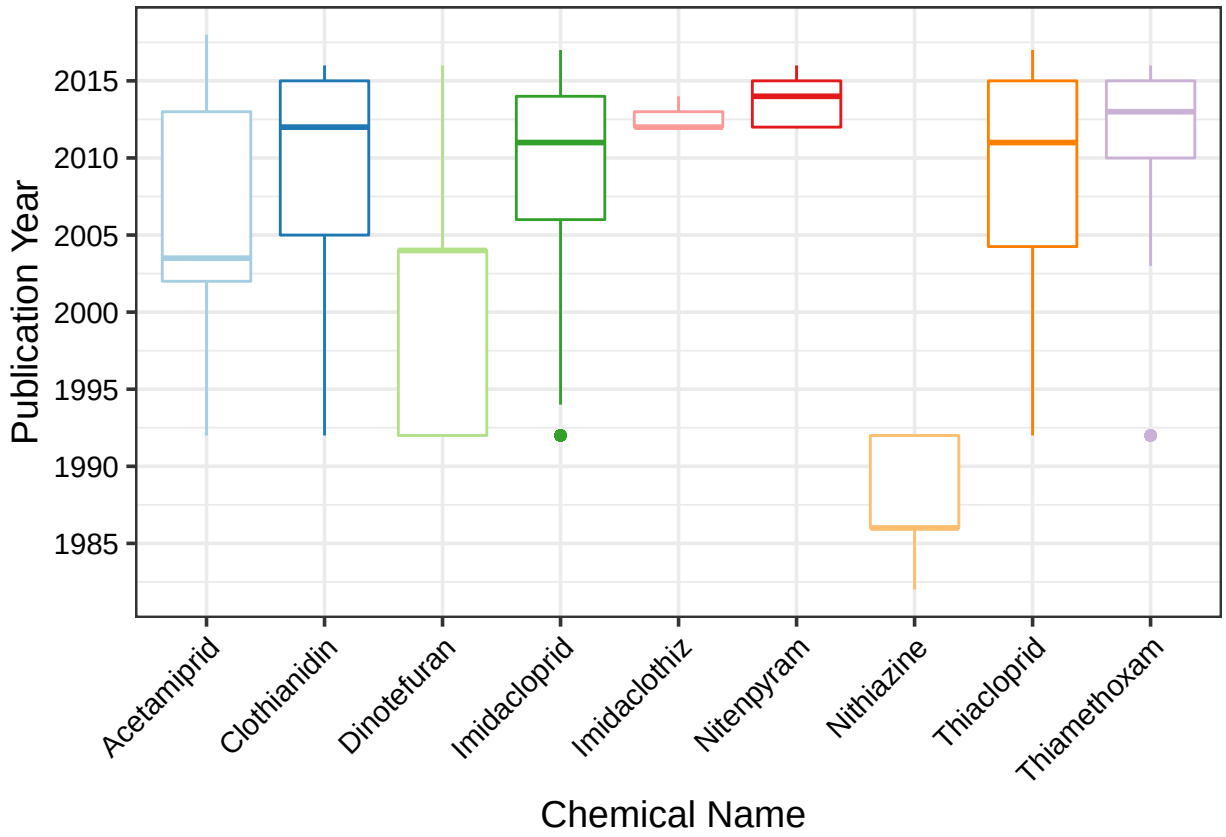
```
#7
Ecotox$Pub..Year <- as.integer(Ecotox$Pub..Year)
class(Ecotox$Pub..Year)
```

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

```
NameYear.kw <- kruskal.test(Ecotox$Pub..Year ~ Ecotox$Chemical.Name)
NameYear.kw
```

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

```r
#8
NameYearPlot <-
  ggplot(Ecotox, aes(x = Chemical.Name, y = Pub..Year, color = Chemical.Name)) +
  geom_boxplot() +
  scale_y_continuous(breaks = c(1985, 1990, 1995, 2000, 2005, 2010, 2015)) +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1), legend.position = "none") +
  labs(x = "Chemical Name ", y = "Publication Year") +
  scale_color_brewer(palette = "Paired")
print(NameYearPlot)
```



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: Publication year by Chemical name is not approximated by a normal distribution. This is shown by the highly significant p-values for each Chemical Name (Shap.test table), meaning that the null hypothesis(there is normality for each) is rejected. In addition, the bartlett test shows that variances among the Chemical Names for publication year is not equal. Therefore, a non-parametric Kruskal Wallis test was utilized. This test shows that there is no significant relationship between publication year and chemical (Kruskal-Wallis; p-value < 0.0001, df $= 8$; chi-squared $= 134.15$).

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
Temp.Pred.July <-
  ChemPhysics %>%
  filter(daynum %in% c(182:212)) %>%
  select(lakename, year4, daynum, depth, temperature_C) %>%
  filter(!is.na(temperature_C) & !is.na(depth) & !is.na(year4) &
           !is.na(daynum) & !is.na(lakename))


#12
Temp.Pred.July.AIC <- lm(data = Temp.Pred.July, temperature_C ~ depth + daynum +
              year4)
step(Temp.Pred.July.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 = Temp.Pred.July)
##
## Coefficients:
## (Intercept)        depth       daynum        year4
##    -6.45556     -1.94726      0.04134      0.01013
```

```
July.model <- lm(data = Temp.Pred.July, temperature_C ~ depth + daynum + year4)
summary(July.model)
```

```
##
## Call:
## lm(formula = temperature_C ~ depth + daynum + year4, data = Temp.Pred.July)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6517 -2.9937  0.0855  2.9692 13.6171
##
## Coefficients:
```

```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -6.455560   8.638808   -0.747   0.4549
## depth       -1.947264   0.011676 -166.782   <2e-16 ***
## daynum       0.041336   0.004315    9.580   <2e-16 ***
## year4        0.010131   0.004303    2.354   0.0186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.811 on 9718 degrees of freedom
## Multiple R-squared:  0.7417, Adjusted R-squared:  0.7417
## F-statistic:  9303 on 3 and 9718 DF,  p-value: < 2.2e-16
```

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: See final linear equation below (p-value < 0.0001, df = 3, 9718, $R^2 = 0.74$). The model explains 74% of the observed variance, which is pretty good.

$$Temperature = -6.46 - 1.95(depth) + 0.04(daynum) + 0.01(year) + \epsilon$$

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

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

```
##
## Call:
## lm(formula = temperature_C ~ lakename * depth, data = Temp.Pred.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 ***
## 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                            -2.5820     0.2411 -10.711  < 2e-16 ***
## lakenameCrampton Lake:depth       0.8058     0.2465   3.268  0.00109 **
## lakenameEast Long Lake:depth      0.9465     0.2433   3.891  0.00010 ***
## lakenameHummingbird Lake:depth   -0.6026     0.2919  -2.064  0.03903 *
## lakenamePaul Lake:depth           0.4022     0.2421   1.662  0.09664 .
## lakenamePeter Lake:depth          0.5799     0.2418   2.398  0.01649 *
## lakenameTuesday Lake:depth        0.6605     0.2426   2.723  0.00648 **
## lakenameWard Lake:depth          -0.6930     0.2862  -2.421  0.01548 *
## lakenameWest Long Lake:depth      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: Yes, there is significant interaction between depth and lakename on temperature of lake (linear regression, ANCOVA; p-value < 0.0001, df $= 17, 9704$, $R^2 = 0.78$). Not all interactions (specifically with Peter and Paul lakes) were significant, but the overall model is significant. 78% of the variance in temperature is explained by this model.

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
Temp.Pred.July.plot <- ggplot(Temp.Pred.July, aes(x = depth, y = temperature_C, color = lakename)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  ylim(0, 35) +
  labs(x = "Depth (m)", y="Temperature (C)", color = "Lake") +
  scale_color_viridis(option = "viridis", discrete = TRUE) +
  theme(legend.position = "bottom", legend.text=element_text(size=9),
        legend.key.size = unit(0.1, "line"))
print(Temp.Pred.July.plot)
```