Assignment 6: Generalized Linear Models

Jiaqi Li

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 Set up
setwd("/Users/ljq/Desktop/Blue Devils/Data Analysis/ENV872_02")
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.7
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
## -- Conflicts ---------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
tox <- read.csv("./Data/Raw/ECOTOX_Neonicotinoids_Mortality_raw.csv")
lake <- read.csv("./Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv")

#2 Set plot theme
tox.theme <- theme_light() +
  theme(axis.title.x=element_blank(), axis.text.x=element_blank(),
```

```
        axis.ticks.x=element_blank())
theme_set(tox.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 Cheminal names
unique(tox$Chemical.Name)
```

```
## [1] Imidacloprid Thiacloprid  Thiamethoxam Acetamiprid  Clothianidin
## [6] Dinotefuran  Nitenpyram   Nithiazine   Imidaclothiz
## 9 Levels: Acetamiprid Clothianidin Dinotefuran ... Thiamethoxam
```

```
#4 # Testing normalization
shapiro.test(subset(tox, Chemical.Name == 'Imidacloprid')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Imidacloprid")$Pub..Year
## W = 0.88178, p-value < 2.2e-16
```

```
shapiro.test(subset(tox, Chemical.Name == 'Thiacloprid')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Thiacloprid")$Pub..Year
## W = 0.7669, p-value = 1.118e-11
```

```
shapiro.test(subset(tox, Chemical.Name == 'Thiamethoxam')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Thiamethoxam")$Pub..Year
## W = 0.7071, p-value < 2.2e-16
```

```
shapiro.test(subset(tox, Chemical.Name == 'Acetamiprid')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Acetamiprid")$Pub..Year
## W = 0.90191, p-value = 5.706e-08
```

```r
shapiro.test(subset(tox, Chemical.Name == 'Clothianidin')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Clothianidin")$Pub..Year
## W = 0.69577, p-value = 4.287e-11
```

```r
shapiro.test(subset(tox, Chemical.Name == 'Dinotefuran')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Dinotefuran")$Pub..Year
## W = 0.82848, p-value = 8.83e-07
```

```r
shapiro.test(subset(tox, Chemical.Name == 'Nitenpyram')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Nitenpyram")$Pub..Year
## W = 0.79592, p-value = 0.0005686
```

```r
shapiro.test(subset(tox, Chemical.Name == 'Nithiazine')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Nithiazine")$Pub..Year
## W = 0.75938, p-value = 0.0001235
```

```r
shapiro.test(subset(tox, Chemical.Name == 'Imidaclothiz')$Pub..Year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(tox, Chemical.Name == "Imidaclothiz")$Pub..Year
## W = 0.68429, p-value = 0.00093
```

```r
# Frequency polygon
year.summary <-
  ggplot(tox, aes(x = Pub..Year, color = Chemical.Name)) +
  geom_freqpoly()

print(year.summary)
```

```r
#5 Variance test
bartlett.test(tox$Pub..Year ~ tox$Chemical.Name)
```

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

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

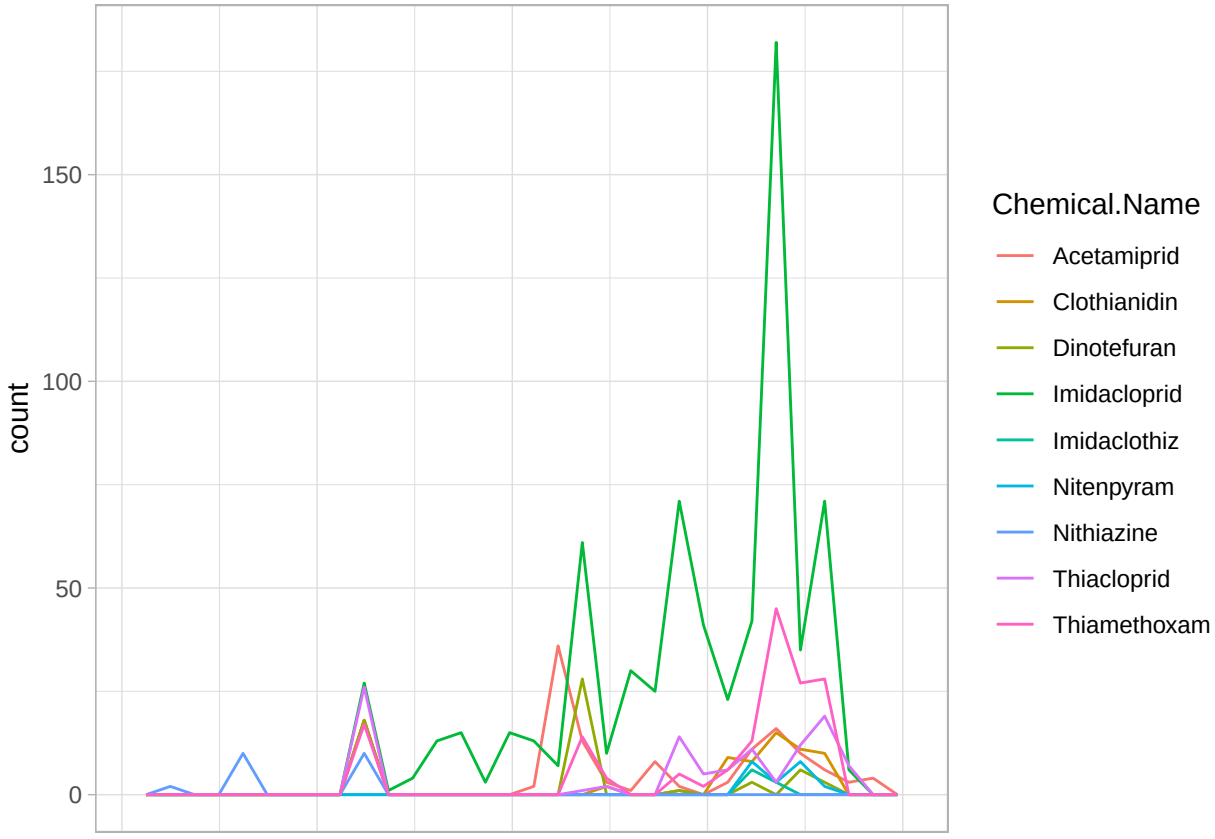

Figure 1: Frequency polygon for publication year

ANSWER: One-way ANOVA, because the response variable *Publish Year* is a continuous variable and the explanatory variable *Chemical Name* is categorical.

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 One-way ANOVA
tox.anova <- lm(tox$Pub..Year ~ tox$Chemical.Name)
summary(tox.anova)
```

```
##
## Call:
## lm(formula = tox$Pub..Year ~ tox$Chemical.Name)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.366  -3.993   1.889   4.889  13.441
##
## Coefficients:
##                                Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                  2005.9926     0.6082 3298.222  < 2e-16 ***
## tox$Chemical.NameClothianidin    2.0479     1.0246    1.999  0.04584 *
## tox$Chemical.NameDinotefuran    -3.4333     1.1057   -3.105  0.00194 **
## tox$Chemical.NameImidacloprid    3.1181     0.6651    4.689 3.05e-06 ***
## tox$Chemical.NameImidaclothiz    6.4518     2.4412    2.643  0.00832 **
## tox$Chemical.NameNitenpyram      7.7216     1.6630    4.643 3.78e-06 ***
## tox$Chemical.NameNithiazine    -17.6290     1.6299  -10.816  < 2e-16 ***
## tox$Chemical.NameThiacloprid     1.6394     0.9190    1.784  0.07467 .
## tox$Chemical.NameThiamethoxam    4.3738     0.8261    5.295 1.40e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.093 on 1274 degrees of freedom
## Multiple R-squared:  0.1726, Adjusted R-squared:  0.1674
## F-statistic: 33.21 on 8 and 1274 DF,  p-value: < 2.2e-16
```

```
#8 Boxplot
tox.box <-
  ggplot(tox, aes(x = Chemical.Name, y = Pub..Year)) +
  geom_boxplot(aes(color =Chemical.Name)) +
  ylab("Publish Year") +
  scale_color_brewer(palette = "Spectral", name = "Chemical Name")

print(tox.box)
```

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: The publish years for each neonicotinoid chemical are statistically significant different from each other (One-way ANOVA; $F = 33.21$, $df = 1274$, $p < 0.0001$). Papers on Nithiazine were published earlist and papers on Nitenpyram were published most recently.

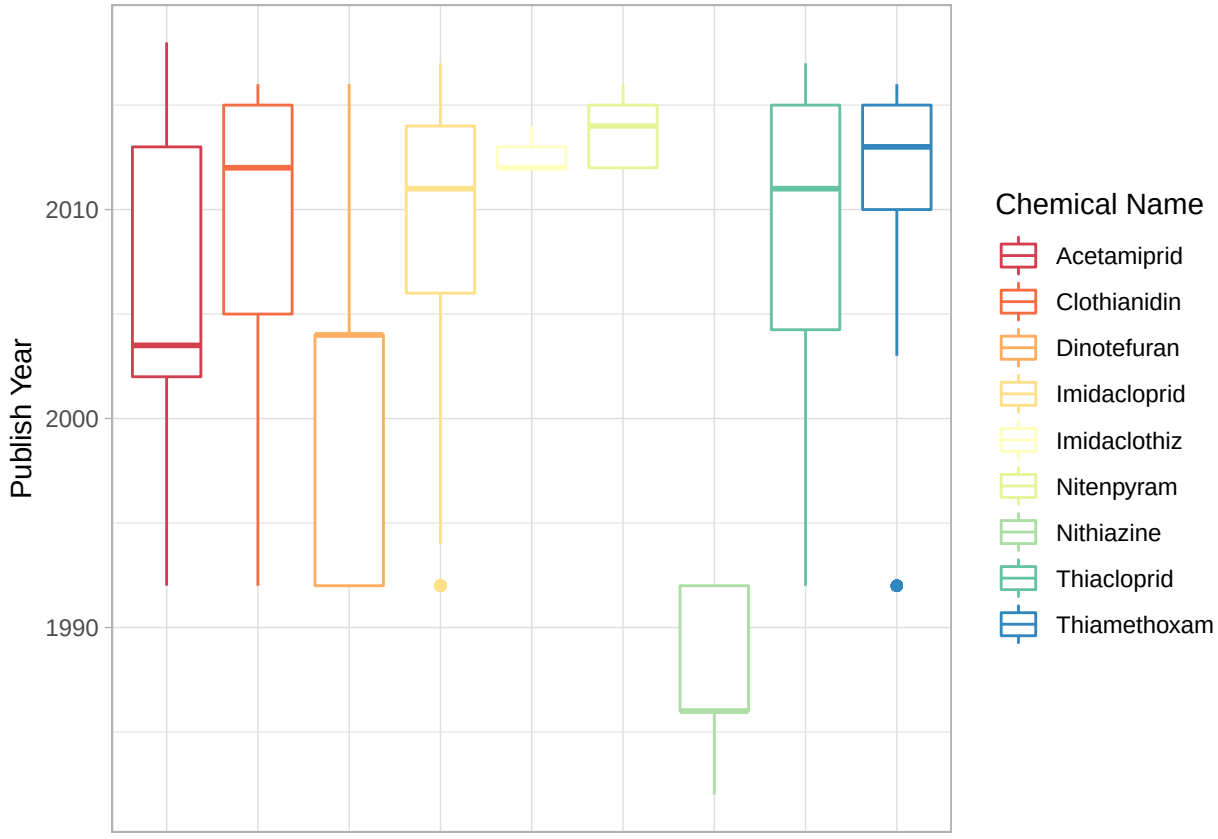

Figure 2: Boxplot of publish year for different neonicotinoid chemicals

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 Tidy the dataset
lake.tidy <- lake %>%
  filter(daynum >= 182 & daynum <= 212) %>%
  select(lakename, year4, daynum, depth, temperature_C) %>%
  na.omit()

#12 AIC test
lakeAIC <- lm(data = lake.tidy, temperature_C ~ year4 + daynum + depth)
step(lakeAIC)
```

```
## 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 = lake.tidy)
##
## Coefficients:
## (Intercept)        year4       daynum        depth
##    -6.45556      0.01013      0.04134     -1.94726
```

```
lake.model <- lm(data = lake.tidy, temperature_C ~ year4 + daynum + depth)
summary(lake.model)
```

```
##
## Call:
## lm(formula = temperature_C ~ year4 + daynum + depth, data = lake.tidy)
##
## 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
## year4        0.010131   0.004303    2.354   0.0186 *
## daynum       0.041336   0.004315    9.580   <2e-16 ***
```

```
## depth       -1.947264   0.011676 -166.782   <2e-16 ***
## ---
## 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: The final linear equation to predict temperature is:

    $$temperature = -6.46 + 0.01 * year + 0.04 * day - 1.95 * depth + \epsilon$$

    The model explains 74.17% of the observed variance (Multiple linear regression; F = 9303, df = 9718, p-value < 0.001).

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

```r
#14 ANCOVA test
lake.interact <- lm(data = lake.tidy, temperature_C ~ lakename * depth)
summary(lake.interact)
```

```
##
## Call:
## lm(formula = temperature_C ~ lakename * depth, data = lake.tidy)
##
## 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: There are interactions between depth and most of the lakes, except Paul Lake. This interaction model explains 78.57% of the variance (ANCOVA; F = 2097, df = 9704, p-value < 0.001).

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 Temperature plot
light.theme <- theme_light(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")

temp.plot <-
  ggplot(lake.tidy, aes(x = depth, y = temperature_C, color = lakename)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  ylim(c(0, 35)) +
  ylab(expression('Temperature ('*~degree*C*')')) +
  xlab("Depth (m)") +
  scale_color_brewer(palette = "YlGnBu", name = "Lake Name") +
  light.theme

print(temp.plot)
```

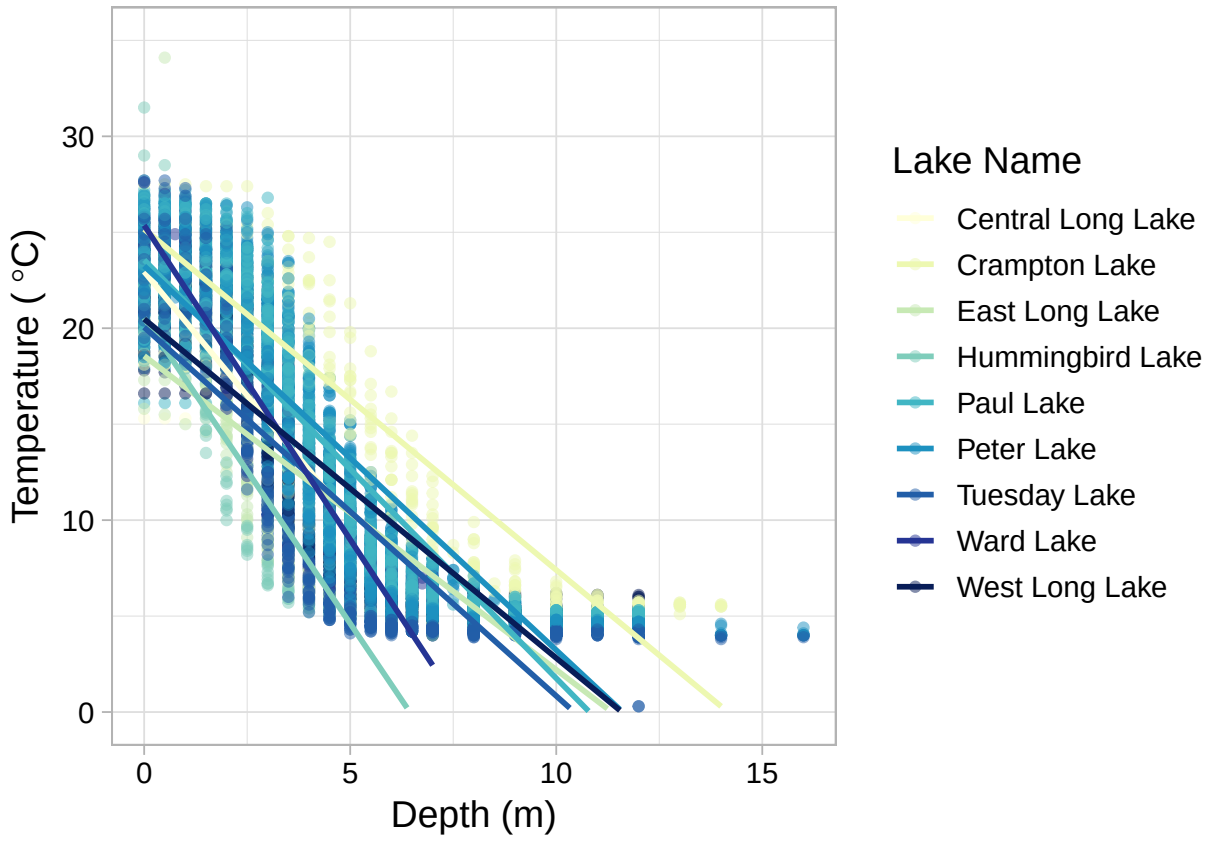

Figure 3: Temperature of lake by depth