Assignment 7: GLMs week 2 (Linear Regression and beyond)

Kim Myers

OVERVIEW

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

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 25 at 1:00 pm.

Set up your session

1. Set up your session. Check your working directory, load the tidyverse, nlme, and piecewiseSEM packages, import the *raw* NTL-LTER raw data file for chemistry/physics, and import the processed litter dataset. You will not work with dates, so no need to format your date columns this time.

2. Build a ggplot theme and set it as your default theme.

```
#1
getwd()
```

```
## [1] "C:/Users/Temp/Documents/Duke/S20/DataAnalytics/Environmental_Data_Analytics_2020/Assignments"
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(piecewiseSEM)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
##
##   This is piecewiseSEM version 2.1.0.
##
##
##   Questions or bugs can be addressed to <LefcheckJ@si.edu>.
```

```
ntl <- read.csv("../Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv")
neon <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
mytheme <- theme_bw(base_size=12)+
   theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

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?

3. 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)

4. 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.

```
#3
ntl_filter <- ntl %>%
  filter(daynum > 181 & daynum < 213) %>%
  select(lakename:daynum, depth, temperature_C) %>%
  na.omit()

#4
ntl_lm <- lm(temperature_C~year4+daynum+depth, data=ntl_filter)
step(ntl_lm)
```

```
## 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_filter)
##
## Coefficients:
## (Intercept)        year4       daynum        depth
##    -6.45556      0.01013      0.04134     -1.94726
```

```r
summary(ntl_lm)
```

```
##
## Call:
## lm(formula = temperature_C ~ year4 + daynum + depth, data = ntl_filter)
##
## 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
```

```r
#the model with year, daynum, and depth is the best-suited to predict temperature
```

5. What is the final set of explanatory variables that predict temperature from your multiple regression? How much of the observed variance does this model explain?

   Answer: Year, day, and depth all predict 74.2% of variation in temperature.

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

```r
#6
library(agricolae)
ntl_int <- lm(temperature_C~lakename*depth, data=ntl_filter)
summary(ntl_int)
```

```
##
## Call:
## lm(formula = temperature_C ~ lakename * depth, data = ntl_filter)
##
## 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
```

7. Is there a significant interaction between depth and lakename? How much variance in the temperature observations does this explain?

   Answer: There is a significant interaction between depth and lake name which explains 78.6% of the variance in temperature.

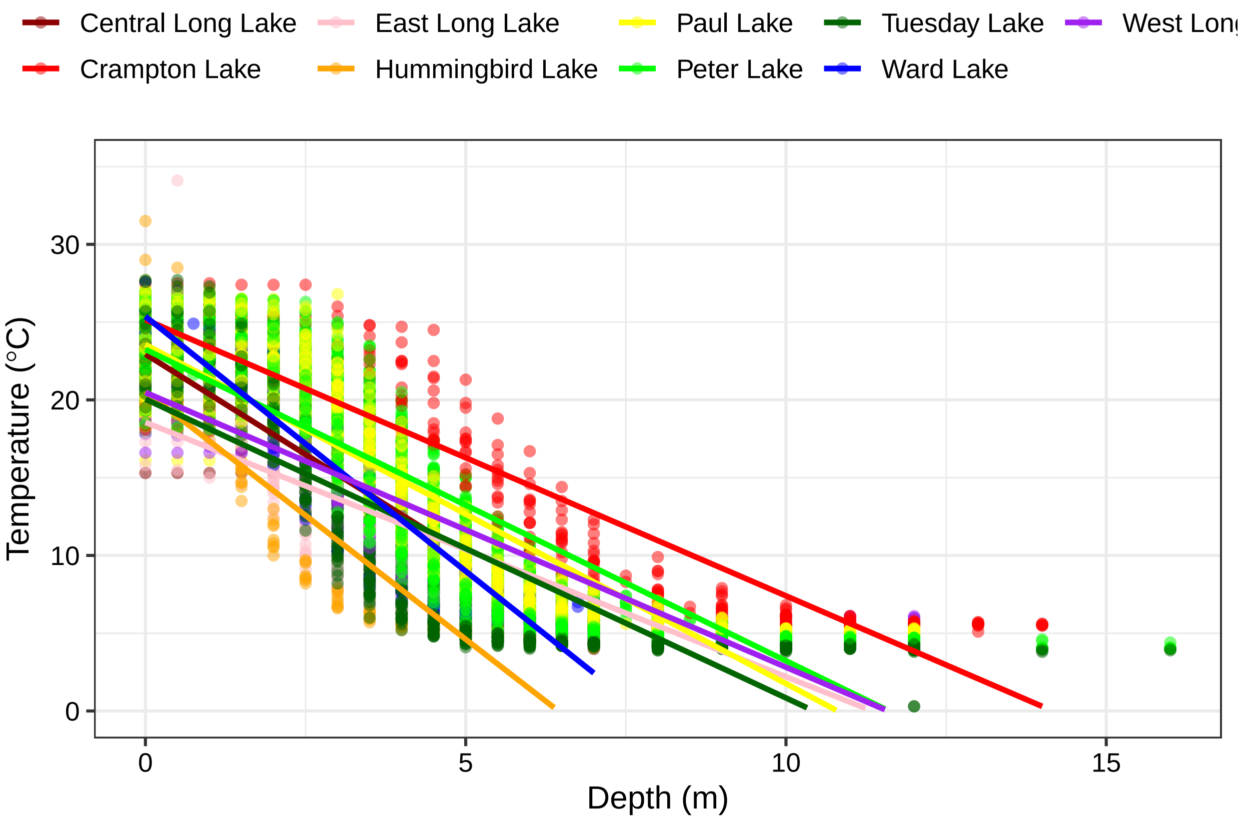
8. Create a graph that depicts temperature by depth, with a separate color for each lake. Add a geom_smooth (method = "lm", se = FALSE) for each lake. Make your points 50 % transparent. Adjust your y axis limits to go from 0 to 35 degrees. Clean up your graph to make it pretty.

```
#8
library(viridis)
```

```
## Loading required package: viridisLite
```

```
tdepth <- ggplot(ntl_filter,aes(y=temperature_C,x=depth,color=lakename)) +
  geom_point(alpha=0.5) +
  geom_smooth(method='lm',se=FALSE) +
  ylim(0,35) +
  scale_color_manual(values = c("darkred", "red", "pink", "orange", "yellow","green","darkgreen","blue"
  labs(y=expression("Temperature ("*degree*"C)"), x="Depth (m)",color="")
print(tdepth)
```

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



9. Run a mixed effects model to predict dry mass of litter. We already know that nlcdClass and functionalGroup have a significant interaction, so we will specify those two variables as fixed effects with an interaction. We also know that litter mass varies across plot ID, but we are less interested in the actual effect of the plot itself but rather in accounting for the variance among plots. Plot ID will be our random effect.

a. Build and run a mixed effects model.
b. Check the difference between the marginal and conditional R2 of the model.

```
library(nlme)
mem1 <- lme(data=neon, dryMass~nlcdClass*functionalGroup, random=~1|plotID)

rsquared(mem1) # marginal = 0.2465822, conditional = 0.2679023
```

```
##   Response   family     link method  Marginal Conditional
## 1  dryMass gaussian identity   none 0.2465822   0.2679023
```

b. continued… How much more variance is explained by adding the random effect to the model?

Answer: The conditional R-squared, which looks at variance explained by random effects in addition to fixed effects, explains 2% more variance than the marginal r-squared, which represents only fixed effects.

c. Run the same model without the random effect.
d. Run an anova on the two tests.

```
lm1 <- gls(data=neon, dryMass~nlcdClass*functionalGroup)
anova(mem1, lm1)
```

```
##      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## mem1     1 26 9038.575 9179.479 -4493.287
## lm1      2 25 9058.088 9193.573 -4504.044 1 vs 2 21.51338  <.0001
```

d. continued… Is the mixed effects model a better model than the fixed effects model? How do you know?

   Answer: The results of the anova show that the mixed effects model is better than the fixed effects model. This is the case because the models are significantly different (p<0.0001) and the AIC of the mixed effects model is lower than that of the fixed effects model.