Linear Regressions

Here you will compute simple least-squares linear regressions in a few ways:

- classic linear model (lm) – which estimates Y given a known X
- glmmTMB - an advance package that allows much fancier models later
- smatr for standardized major axis regression – which estimates Y given an *estimated* X

You will also evaluate model assumptions using performance (which you used last week). Below is a mixture of information, questions, and **`code.`** As always, retain and annotate scripts for your future use.

First install packages (lm is already part of base R) and then load them:

```
install.packages('glmmTMB') # enables generalized linear mixed effect models
install.packages('smatr') # enables standardized major axis regression
install.packages('performance') # enables efficient assumptions tests - already installed?

library(glmmTMB)
library(smatr)
library(performance)
```

Caterpillar-tannins experiment: In the ongoing war between insects and plants, tannins produced by plants interfere with insect nutrient absorption. Some plants (e.g., oaks) have high tannin content, whereas other plant species do not. But one cannot simply feed different plant species to caterpillars to test for tannin effects – other factors matter, too (leaf toughness, etc.). So here tannins were added to a standard food to answer this question: How much does tannin concentration (mg/g dry weight) depress caterpillar growth (mm length) during the 2-week experiment? Each entry represents an experimental unit containing ad lib food and 30 caterpillars, for which mean growth and tannin level were recorded.

1. Get the `tancat.txt` from the course web site and attach it.
2. Compute a linear regression of growth as a function of tannin:

```
tanmodel1 <- lm(growth~tannin)
summary(tanmodel1)
plot(growth~tannin)   # plots the data and regression model
abline(tanmodel1)
```

Is there a clear relationship? How strong is it?

3. Let's test assumptions required for this model: Are they OK?

```
check_model(tanmodel1)
```

Now try that same model again in glmmTMB – we *should* get the same result:

```
tanmodel2 <- glmmTMB(growth~tannin)
summary(tanmodel2)
check_model(tanmodel2)
```

Both of these packages use classic linear regression – estimate Y (with error) assuming a precisely known X (without error). But lm uses classic frequentist methods, whereas glmmTMB uses maximum likelihood (a big difference in details). Either way, this can be used to interpolate

new Y estimates within the range. It is not intended to estimate X given a Y (though it is often used that way).

But what if both Y and X are estimates with error? What if one wished to calculate a "true" model that could be used to estimate Y given X, or estimate X given Y? In that case, we use standardized major axis regression, often used in allometry (read Warton et al. 2012 for details).

```
tanmodel3 <- sma(growth~tannin, method="SMA")
summary(tanmodel3)
check_model(tanmodel3)
plot(growth~tannin)  # plots the data and regression model
abline(tanmodel3)
```

Notice that:

- performance doesn't work with sma models
- the sma slope will always be steeper than the lm slope
- but the same R^2 is obtained

SMA models only work with one predictor (X). For multiple regressions we have to use lm or fancy versions (e.g., glmmTMB).

Now we let you decide how to analyze a new data set (commands above may help):

Mercury in Fish of FL Lakes: Fish are regularly sampled for mercury (Hg) contamination. This data set represents average Hg load in fish (mg/kg) in fish from 53 Florida lakes. Also reported are basic water quality measures – alkalinity (alk), pH, calcium concentration (Ca), and chlorophyll *a* (Chla – a measure of algal concentration in water). Each of these factors may predict Hg load in fish, related to chemical complexing of Hg and food web uptake.

Your mission: *find the best **single** predictor of Hg* load in fish based on these four potential predictors. You decide: should you use lm (or glmmTMB) or smatr?

4. Get the `FLHg.txt` file from the course web site.

```
detach tancat # to detach (unattach) the tancat data set
attach FLHg
```

5. Try this panel of graphs with smoothed model curves to visualize the data set:

```
pairs(FLHg[2:6], panel=panel.smooth) #[2:6] picks the numeric columns
```

6. Repeat the needed `lm` or `glmmTMB` commands used above for tannins, but here first evaluate `alk` as a predictor of Hg. Name your model to suit the predictor (e.g., alkmodel).

Does this regression comply with assumptions?
Try a square-root transform of Hg and then repeat the regression and assumptions tests. Better?

7. *Now repeat and rinse* for the other three predictor variables to obtain a suite of single-predictor models for Hg in fish. Keep renaming your models to suit.

Now use model selection (based on AICc) to compare regression models:

7. Install the `bbmle` package (if it is not already in packages), and then load it.
8. Enter this command, where model names are assumed below (edit as needed):

```
library(bbmle) # for AIC calculations, should already be installed
AICctab(alkmodel, pHmodel, Camodel, Chlamodel, sort=TRUE, base=TRUE,  delta=TRUE,
weights=TRUE)
```

The resulting table shows you a model-selection approach to answer the question (Which ONE...) above. Bottom line = the model with the greatest weight is the most plausible model among those listed. How much better? Delta AICc (`dAICc`) shows you how much better a model is compared to the next one, and weight tells you the probability that a model is most plausible, given the list of models.

Based on AIC output, which ONE predictor variable would you recommend be used to predict Hg load in fish of Florida lakes?

Now look at your most-efficient model with summary and plot.

How well does the most-efficient model predict fish [Hg]?
How does that compare to your expectation based on the initial grid of graphs?

If you had to tell your boss at the FL EPA about this, how would you explain your results?