Lecture 8.3: Multiple regression part III: categorical variables and interactions

** This lecture is based on chapter 5 of Statistical Rethinking by Richard McElreath.*

As always, we need to load some packages and set some options prior to running any models:

```r
library(rstan)
library(shinystan)
library(car)
library(xtable)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
source("../utilityFunctions.R")
```

In lecture 8.2, we had a brief think about dummy coding as on-off switches for categorical variables. Now we are going to expand that thought process to encompass many categories.

Binary categories are easy to think about because we only need one dummy variable. The other category is the intercept α.

When there are more than 2 categories, we need $k - 1$ dummy variables, where $k =$ the number of groups.

- Each $k - 1$ dummy variable represents, with value 1, a unique category
- The kth category ends up again as the intercept

Let's go back to the milk dataset again (full data; `milkFull.csv`), where we were interested in the energy content (`kcal`$\cdot g^{-1}$) of primates.

- Now we are interested in the `clade` variable, which encompasses broad taxonomic categories of the primate species.

```r
milk <- read.csv("milkFull.csv")
unique(milk$clade)
```

```
[1] Strepsirrhine    New World Monkey Old World Monkey Ape
Levels: Ape New World Monkey Old World Monkey Strepsirrhine
```

There are four categories and none of them has yet been coded as dummy variables.

To make dummy variables for the `Streppsirrhine` category:

```r
(strep <- ifelse(milk$clade == "Strepsirrhine", 1, 0))
```

```
 [1] 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

only those rows (observations) where `clade == "Strepsirrhine` get a 1.

We can make the rest of the dummy variables with the same strategy and `cbind` them together to make our design matrix:

```r
nwm <- ifelse(milk$clade == "New World Monkey", 1, 0)
owm <- ifelse(milk$clade == "Old World Monkey", 1, 0)
clade <- cbind(nwm, owm, strep)
```

Note that we don't need for an `Ape` dummy variable because it will be the default intercept category.

- Including a dummy variable for `Ape` as well will result in a non-identified model. *Why?*

Another (easy) way to create a design matrix is to use the `model.matrix` function in `R`.

```r
X <- model.matrix(~clade, data=milk)
X <- as.matrix(X) # needed to get rid of the attribute list
```

By default `model.matrix` orders categories alphabetically, and the first category is a vector of 1's for the intercept.

- If we want to specify the intercept separately (as α) then drop the first column. Otherwise we could just have a k-length vector of $\beta' s$.

Our model is then specified as

$$\begin{aligned}
obs_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \sum_{j=2}^{k} \beta_k x_{ji} \\
&= \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_k x_{ki} \\
\alpha &\sim \text{Normal}(a_\mu, a_{SD}) \\
\beta_j &\sim \text{Normal}(b_\mu, b_{SD}) \\
\sigma &\sim \text{Cauchy}^+(0, \sigma_{SD}).
\end{aligned} \tag{1}$$

Here is how dummy variables assign individual parameters based on category:

Category	NWM$_i$	OWM$_i$	Strep$_i$	μ_i
Ape	0	0	0	$\mu_i = \alpha$
NW Monkey	1	0	0	$\mu_i = \alpha + \beta_1$
OW Monkey	0	1	0	$\mu_i = \alpha + \beta_2$
Strepsirrhine	0	0	1	$\mu_i = \alpha + \beta_3$

Fitting the model (`multMod8.2.stan`) is identical to last time.

```r
nObs <- nrow(milk)
nVar <- ncol(clade)
obs <- milk$kcal
X   <- clade
aMu <- bMu <- 0.6
aSD <- sigmaSD <- 10
bSD <- 1

dat <- list(nObs=nObs, nVar=nVar, obs=obs, X=X, aMu=aMu, aSD=aSD,
  bMu=bMu, bSD=bSD, sigmaSD=sigmaSD)

milkMod <- stan(file="multMod8.2.stan", data=dat, iter=2000,
 chains=4, seed=867.5309, pars="mu", include=FALSE)

round(summary(milkMod, pars=c("alpha", "beta", "sigma"),
  probs = c(0.025, 0.5, 0.975))$summary,2)
```

```
         mean se_mean   sd  2.5%   50% 97.5%   n_eff Rhat
alpha    0.54       0 0.04  0.46  0.54  0.63 1355.94    1
beta[1]  0.17       0 0.06  0.05  0.17  0.29 1689.74    1
beta[2]  0.25       0 0.07  0.11  0.25  0.39 1897.09    1
beta[3] -0.03       0 0.07 -0.17 -0.03  0.11 1793.07    1
sigma    0.13       0 0.02  0.10  0.13  0.18 2571.07    1
```

The marginal posterior estimate α is the average milk energy for apes.

- All other categories are deviations from `Ape`.
- To get their posterior distributions of average milk energy for each category:

```r
post <- as.matrix(milkMod, pars=c("alpha","beta"))
mu <- cbind(post[,1],sweep(post[,2:4], MARGIN=1, STATS = post[,1], FUN='+'))
means <- colMeans(mu)
SD <- apply(mu, 2, sd)
muHDI <- apply(mu, 2, HDI, credMass=0.95)
sumTab <- data.frame(Mean=means, SD=SD, lower.95=muHDI[1,],
  upper.95=muHDI[1,], row.names=c("Ape", "NWM", "OWM", "Strep"))
```

% latex table generated in R 3.3.2 by xtable 1.8-2 package % Wed Mar 8 15:42:36 2017

	Mean	SD	lower.95	upper.95
Ape	0.54	0.04	0.45	0.45
NWM	0.71	0.04	0.63	0.63
OWM	0.79	0.05	0.68	0.68
Strep	0.51	0.06	0.40	0.40

If we are interested in whether there is a difference between `New World Monkeys` and `Old`

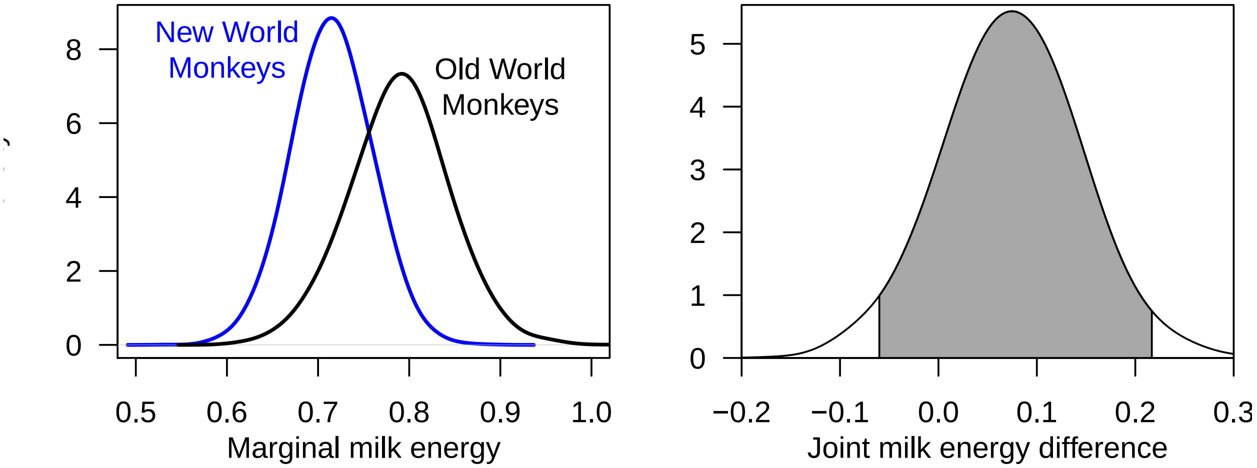
`World Monkeys`, it isn't enough to look for overlap in the marginal 95% uncertainty intervals or whether one overlaps with zero a lot, while the other is far away from zero.

- This isn't just unique to Bayesian models either.
- Instead, we need to compute a *contrast* of the difference.

For example, suppose β_1 has a mean and standard deviation of 0.15 ± 0.02—reliably different from zero—and $\beta_2' s$ is 0.02 ± 0.10—not so different. The difference is $(0.15 - 0.02) \pm \sqrt{0.02^2 + 0.10^2} \approx 0.13 \pm 0.10$.

- β_1 is reliably different from zero but we cannot say that β_1 and β_2 are different.
- Instead, conduct a contrast. Because we are working with samples from the posterior, we get a posterior distribution of the difference; none of the uncertainty is lost. Yay Bayes!

```r
par(mar=c(3,3.2,0.1,0.5))
par(mfrow=c(1,2))
plot(density(mu[,2], adj=2), las=1, col="blue", lwd=2, main="",
  xlim=c(0.5, 1))
lines(density(mu[,3], adj=2), col="black", lwd=2)
mtext(text = "Marginal milk energy", side=1, line = 2, cex=1)
text(0.6,8, "New World\nMonkeys", col="blue")
text(0.9,7, "Old World\nMonkeys", col="black")

# Contrast plot
dif <- mu[,3]-mu[,2]
plotInterval(dif, HDI = TRUE, interval = 0.95, xlims=c(-0.2,0.3), col="#50505080")
mtext(text = "Joint milk energy difference", side=1, line = 2, cex=1)
```



Going back to the `New World` vs. `Old World` comparison, plotting the marginal densities suggests substantial overlap between the two categories.

However, plotting the joint distribution of the difference suggests that the two types of monkeys differ substantially in their milk energies (HDI: -0.06, 0.22). The proportion of differences below zero is only 0.14; much less than the marginal distributions would suggest.

Unique intercepts

Another way to model categorical variables is to construct a vector of intercept parameters, one for each category and then use index variables.

- This is very similar to what we did earlier when we made hierarchical models.

```
data {
  int<lower=0> nObs;
  int<lower=0> nVar;      // no. vars
  vector[nObs] obs;
  int x[nObs];
  real<lower=0> aMu;      // mean of prior alpha
  real<lower=0> aSD;      // SD of prior alpha
  real<lower=0> sigmaSD;  // scale for sigma
}

parameters {
  vector[nVar] alpha;
  real<lower=0> sigma;
}

model {
  alpha ~ normal(aMu, aSD);
  sigma ~ cauchy(0, sigmaSD);
  {
    vector[nObs] mu;
    mu = alpha[x];

    obs ~ normal(mu, sigma);
  }
}
```

```
obs <- milk$kcal
x   <- as.integer(milk$clade)
nObs <- nrow(milk)
nVar <- max(x)
aMu <- 0.6
aSD <- sigmaSD <- 10

dat <- list(nObs=nObs, nVar=nVar, obs=obs, x=x, aMu=aMu, aSD=aSD,
  sigmaSD=sigmaSD)
```

```
intMod <- stan(file="interceptMod.stan", data=dat, iter=2000,
 chains=4, seed=867.5309)

round(summary(intMod, pars=c("alpha", "sigma"),
  probs = c(0.025, 0.5, 0.975))$summary,2)
```

```
         mean se_mean   sd 2.5%  50% 97.5%   n_eff Rhat
alpha[1] 0.55       0 0.04 0.46 0.55  0.63 4000.00    1
alpha[2] 0.71       0 0.04 0.63 0.71  0.79 4000.00    1
alpha[3] 0.79       0 0.05 0.68 0.79  0.90 4000.00    1
alpha[4] 0.51       0 0.06 0.39 0.51  0.63 4000.00    1
sigma    0.13       0 0.02 0.10 0.13  0.18 2957.52    1
```

Interactions

The next example (from Nunn & Puga, 2011) is lame because it isn't ecological, but the thought process is nice nonetheless. The data consist of the gross domestic product (`GDP`) for a number of countries around the world in relation to geography (specifically ruggedness).

- We are going to focus on GDP comparisons between African countries and non-African countries (artificial I know).
  - We will use the logarithm of GDP because wealth tends to be an exponential process and we are interested in the *magnitude* (i.e., wealth begets more wealth)
- Terrain ruggedness (`rugged`) is often related to bad economies—at least outside of Africa.
  - Here `rugged` is a Terrain Ruggedness Index that quantifies topographic heterogeneity of landscapes.

Let's read in the data and fit regression models to African vs non-African countries separately at first using the `uniMod.stan` model.

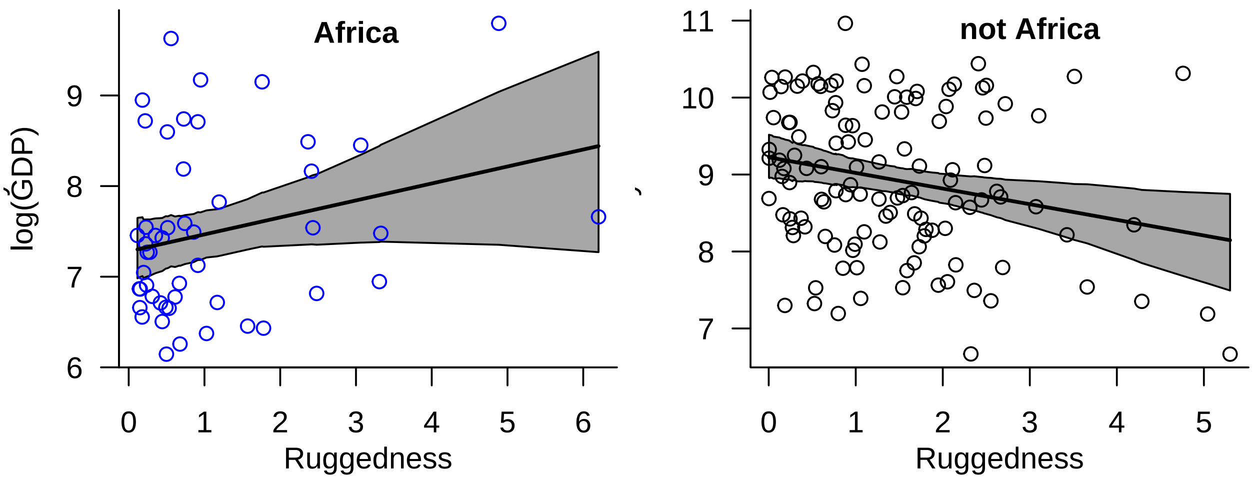
```
rugged <- read.csv("RuggedGDP.csv")
rugged <- rugged[order(rugged$rugged),]
afr <- rugged[rugged$africa == 1, ] # African dataset
nafr <- rugged[rugged$africa == 0, ] # non-African dataset
afr <- afr[order(afr$rugged),]
nafr <- nafr[order(nafr$rugged),]
```

```
# African linear regression
afrDat <- list(nObs=nrow(afr), obs=log(afr$GDP), xvar=afr$rugged, aSD=20,
   bSD=1, sigmaSD=10)

afrMod <- stan(file="uniMod.stan", data=afrDat, iter=2000,
 chains=4, seed=867.5309)
```

```r
afrMu <- as.matrix(afrMod, "mu")

# NonAfrican linear regression
nafrDat <- list(nObs=nrow(nafr), obs=log(nafr$GDP), xvar=nafr$rugged, aSD=20,
   bSD=1, sigmaSD=10)

nafrMod <- stan(file="uniMod.stan", data=nafrDat, iter=2000,
 chains=4, seed=867.5309)
nMu <- as.matrix(nafrMod, "mu")
```

```r
par(mar=c(3,3.2,0.1,0.5))
par(mfrow=c(1,2))

### AFRICA
# Mean & HDI
afrHDI <- apply(afrMu,2, HDI, credMass=0.95)
afrMean <- colMeans(afrMu)

# Make an empty plot
x <- afrDat$xvar
y <- afrDat$obs
plot(x, y, type="n", las=1, bty="l")
mtext(text = "Ruggedness", side=1, line = 2, cex=1)
mtext(text = "log(GDP)", side=2, line = 2.2, cex=1)

# plot uncertainty interval in mu as a polygon
polygon(x=c(x, rev(x)), y=c(afrHDI[1,],
  rev(afrHDI[2,])), col="#50505080", border="black")

# plot the data points and mean regression line
points(x, y, pch=1, col="blue")
lines(afrMean~x, col="black", lwd=2)
text(3, 9.7, "Africa", font=2)

### NONAFRICA
nHDI <- apply(nMu,2, HDI, credMass=0.95)
nMean <- colMeans(nMu)

# Make an empty plot
x <- nafrDat$xvar
y <- nafrDat$obs
plot(x, y, type="n", las=1, bty="l")
mtext(text = "Ruggedness", side=1, line = 2, cex=1)
```

```r
# plot uncertainty interval in mu as a polygon
polygon(x=c(x, rev(x)), y=c(nHDI[1,],
  rev(nHDI[2,])), col="#50505080", border="black")

# plot the data points and mean regression line
points(x, y, pch=1, col="black")
lines(nMean~x, col="black", lwd=2)
text(3, 10.9, "not Africa", font=2)
```



It might make sense that ruggedness would be associated with poor countries.

- Rugged terrain makes travel challenging, which might impede the movement of goods and services to market, thus depressing wealth.
- But why is the situation reversed in Africa?
  - One hypothesis is that rugged regions served as a barrier to the slave trade, which was predominately based on the coasts. Those regions continue to suffer economically in many regards.

Regardless, irregardless even, of the reversal between African and non-African countries, how do we model this? Here we are cheating by spliting the data in two, but there are obvious drawbacks to that.

1. There are some parameters that are independent of African identity (e.g., σ).
   - By doing a no-pooling model, we are decreasing the accuracy of the estimate for those parameters are making two less-accurate estimates rather than pooling all the evidence into one.
   - also make a strong assumption that the variance differs between African and non-African countries.
2. We are not making any probability statements about the difference between African and non-African countries because we are not including that variable in the model.
   - Assuming there is no uncertainty in discriminating between African/non-African

countries.

3. Later on, we may want to use information criteria to compare models that treat all the data as belonging to one posterior distribution as opposed to a model that allows different slopes.
    - We have to include all the data in a model to do so.

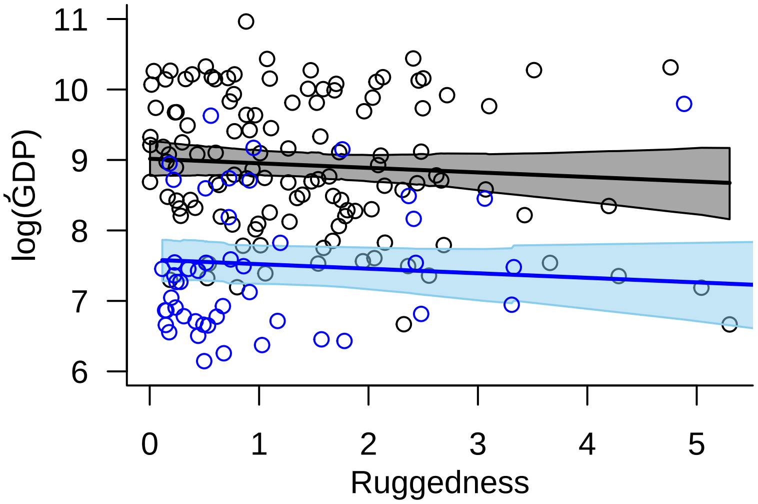
Adding a dummy variable

So let's add Africa as an indicator variable and use model `multMod8.2.stan`. How does singling out African nations affect our conclusions?

```r
# African linear regression
X <- cbind(rugged$rugged, rugged$africa)
fullDat <- list(nObs=nrow(rugged), nVar=ncol(X), obs=log(rugged$GDP), X=X,
  aMu=0, aSD=20, bMu=0, bSD=1, sigmaSD=10)

fullMod <- stan(file="multMod8.2.stan", data=fullDat, iter=2000,
 chains=4, seed=867.5309)
mu <- as.matrix(fullMod, "mu")
muHDI <- apply(mu, 2, HDI, credMass=0.95)
muMn <- colMeans(mu)
```

```r
par(mar=c(3,3.2,0.1,0.5))
par(mfrow=c(1,1))
### AFRICA
# Mean & HDI
afrHDI <- muHDI[,rugged$africa==1]
afrMean <- muMn[rugged$africa==1]

### not AFRICA
# Mean & HDI
nHDI <- muHDI[,rugged$africa==0]
nMean <- muMn[rugged$africa==0]

# Make an empty plot
x <- nafrDat$xvar
y <- nafrDat$obs
plot(x, y, type="n", las=1, bty="l", ylim=c(6,11))
mtext(text = "Ruggedness", side=1, line = 2, cex=1)
mtext(text = "log(GDP)", side=2, line = 2.2, cex=1)

# plot uncertainty interval in mu as a polygon
polygon(x=c(x, rev(x)), y=c(nHDI[1,],
  rev(nHDI[2,])), col="#50505080", border="black")
```

```
# plot the data points and mean regression line
points(x, y, pch=1, col="black")
lines(nMean~x, col="black", lwd=2)

### AFRICA
x <- afrDat$xvar
y <- afrDat$obs
# plot uncertainty interval in mu as a polygon
polygon(x=c(x, rev(x)), y=c(afrHDI[1,],
  rev(afrHDI[2,])), col="#88CCEE80", border="#88CCEE")

# plot the data points and mean regression line
points(x, y, pch=1, col="blue")
lines(afrMean~x, col="blue", lwd=2)
```



According to this figure, there is a slightly negative relationship between GDP and ruggedness overall.

- Including the dummy variable for Africa has allowed the model to predict a lower mean GDP for African nations relative to non-African nations.
- But the slopes are parallel! What's going on?

Adding a linear interaction

Unsurprisingly perhaps, we need an interaction effect.

The likelihood function for the previous model was essentially:

$$\begin{aligned} obs_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha + \beta_R R_i + \beta_A A_i \end{aligned}$$

where R is `rugged` and A is `africa`.

This linear model is constructed by specifying the mean μ as a linear function of new parameters and data.

Interactions will extend this approach. Now we want the relationship between *obs* and R to vary as a function of A.

- In the previous model, this relationship was measured by β_R.
  - What we want to do is make β_R a linear function itself—one that includes A.

$$\begin{aligned}
obs_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \gamma_i R_i + \beta_A A_i \\
\gamma_i &= \beta_R + \beta_{AR} A_i
\end{aligned}$$

Now our Bayesian model has two linear models in it, but it's essentially the same as every regression model thus far.

1. The first line is the same Gaussian likelihood we all know and love.
2. The second line is the same additive definition of μ_i.
3. The third line uses γ as a placeholder for our new linear function that defines the slope between `log(GDP)` and `rugged`.
   - γ_i is the linear interaction effect of ruggedness and African nations

γ_i explicitly models the hypothesis that the slope between `GDP` and ruggedness is *conditional* on whether a nation is on the African continent with β_{AR} describing the strength of that dependence.

- If $\beta_{AR} = 0$, then we get our original likelihood function back.
  - For any nation not in Africa, $A_i = 0$ and so β_{AR} has no effect.
- If $\beta_{AR} > 1$, African nations have a more positive slope between GDP and ruggedness.
- if $\beta_{AR} < 1$, African nations have a more negative slope

We could also rewrite this equation using the conventional notation by substituting in γ_i:

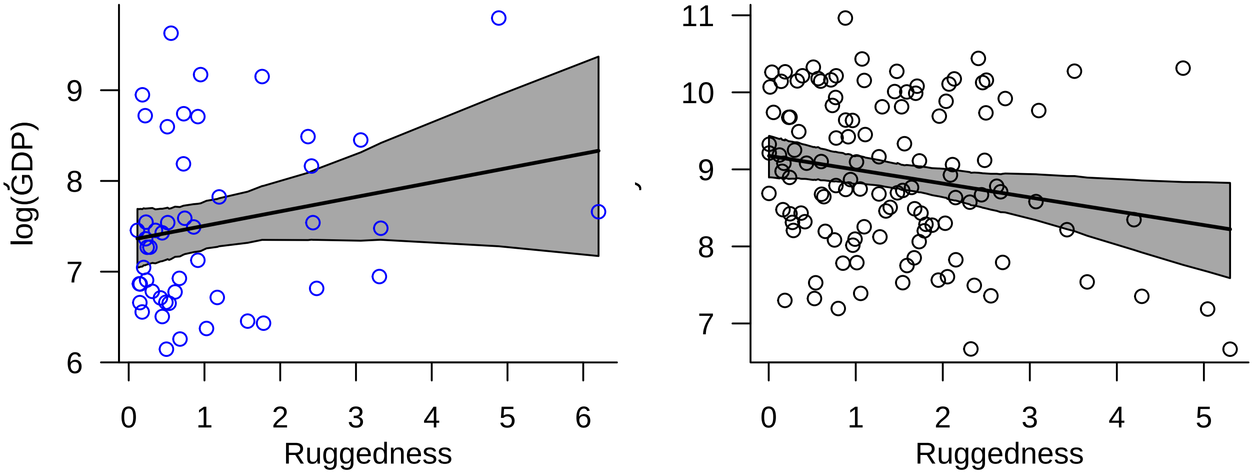
$$\begin{aligned}
\gamma_i &= \beta_R + \beta_{AR} A_i \\
\mu_i &= \alpha + \gamma_i R_i + \beta_A A_i \\
&= \alpha + (\beta_R + \beta_{AR} A_i) R_i + \beta_A A_i \\
&= \alpha + \beta_R R_i + \beta_{AR} A_i R_i + \beta_A A_i.
\end{aligned}$$

The former is more explicit and understanding it will be key to understanding (and building) more complex hierarchical models later. I leave it as an exercise for the class to create a stan model that runs using this form.

- Note that Stan will most likely throw an error unless you use either elementwise multiplication (`.*`) or some creative parameter/variable definitions.
  - If you successfully do the latter, you will have a much greater understanding of how to build Stan models using `matrices`, `vectors`, and `row_vectors`.
  - Look at the Stan manual for help (specifically sections II, IV, & VII).

The latter likelihood function is much easier to code though.

```
# African linear regression
X <- model.matrix(~rugged*africa, data=rugged)[,2:4]
intDat <- list(nObs=nrow(rugged), nVar=ncol(X), obs=log(rugged$GDP), X=X,
  aMu=0, aSD=20, bMu=0, bSD=1, sigmaSD=10)

intMod <- stan(file="multMod8.2.stan", data=intDat, iter=2000,
 chains=4, seed=867.5309)
mu <- as.matrix(intMod, "mu")
muHDI <- apply(mu, 2, HDI, credMass=0.95)
muMn <- colMeans(mu)
```



If we look at a plot of the results, now the slopes reverse direction for within vs. outside of Africa.