Lecture 13: Multiple Regression part III: Interactions

Zachary Marion

3/12/2018

** This lecture is based on chapter 7 of Statistical Rethinking by Richard McElreath.*

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

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

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);
  }
}
```

```
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
```

```
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="13.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] 0.72       0 0.04 0.63 0.72  0.80 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 0.40 0.51  0.62 4000.00    1
sigma    0.13       0 0.02 0.10 0.13  0.18 3342.14    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 `13.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="13.uniMod.stan", data=afrDat, iter=2000,
 chains=4, seed=867.5309)
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="13.uniMod.stan", data=nafrDat, iter=2000,
 chains=4, seed=867.5309)
nMu <- as.matrix(nafrMod, "mu")
```

```
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")
```

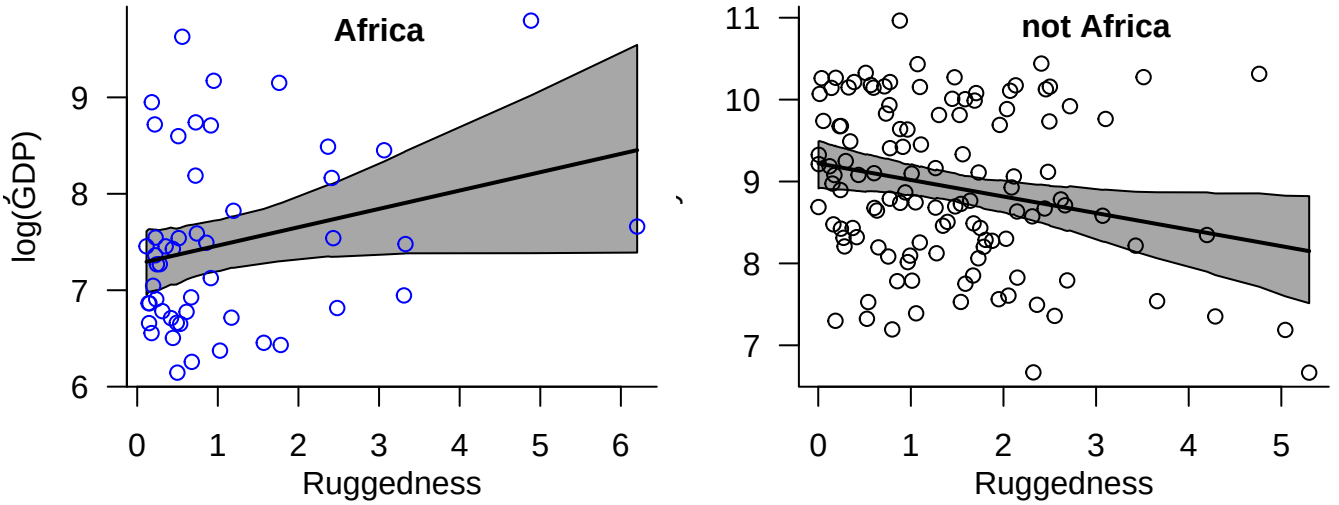
```r
# plot the data points and mean regression line
points(x, y, pch=1, col="blue")
lines(afrMean~x, col="black", lwd=2)
text(3, 9.7, "Africa", font=2)

### 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)

# 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 splitting the data in two, and 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 and 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 `13.multMod.stan`. How does singling out African nations affect our conclusions?

```
# 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="13.multMod.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)
```

```
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]
```

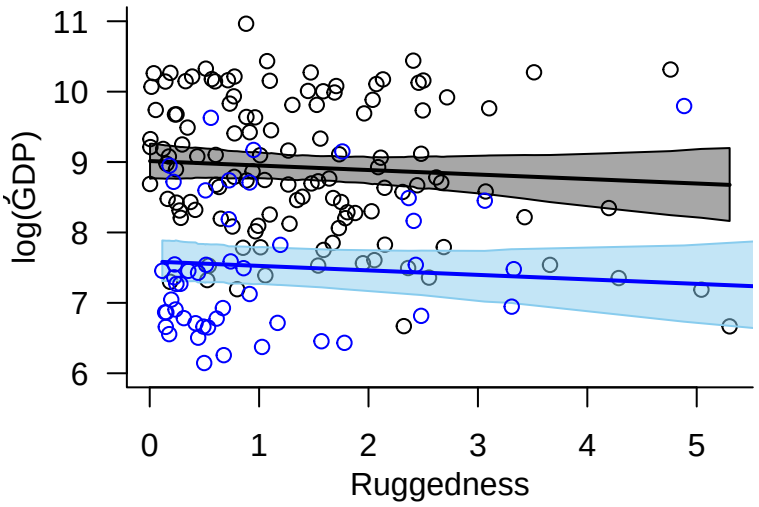
```r
# Make an empty plot
x <- nafrDat$xvar
y <- nafrDat$obs
plot(x, y, type="n", las=1, bty="l", ylim=c(6,11))
mtext(text = "Ruggedness", side=1, line = 2, cex=1)
mtext(text = "log(GDP)", side=2, line = 2.2, cex=1)

# 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.

The latter likelihood function is much easier to code though. There are about 5 different ways to code the first model. The easiest is to do the following:

```
data {
  int<lower=0> nObs;
  vector[nObs] obs;
  vector[nObs] R;
  vector[nObs] A;
  real<lower=0> aMu;       // mean of prior alpha
  real<lower=0> aSD;       // SD of prior alpha
  real<lower=0> bMu;       // mean of prior betas
  real<lower=0> bSD;       // SD of prior beta
  real<lower=0> sigmaSD;   // scale for sigma
}

parameters {
  real alpha;
  real betaR;
  real betaA;
  real betaAR;
  real<lower=0> sigma;
}

transformed parameters {
  vector[nObs] mu;
  vector[nObs] gamma;

  gamma = betaR + betaAR*A;
  // elementwise multiplication (.*)!
  mu = alpha + gamma .* R + betaA*A;
}

model {
  alpha ~ normal(aMu, aSD);
  betaR ~  normal(bMu, bSD);
  betaA ~  normal(bMu, bSD);
  betaAR ~  normal(bMu, bSD);
  sigma ~ cauchy(0, sigmaSD);
```

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

Could also do the following:

```
transformed parameters {
  vector[nObs] mu;
  vector[nObs] gamma;

  gamma = betaR + betaAR*A;
  mu = alpha + to_vector(gamma * R') + betaA*A;
}
```

The ' symbol means transpose and turns a `nObs` $\times$ 1 `vector` into a 1 $\times$ `nObs` vector for proper matrix multiplication (and results in a `nObs` $\times$ 1) output matrix).

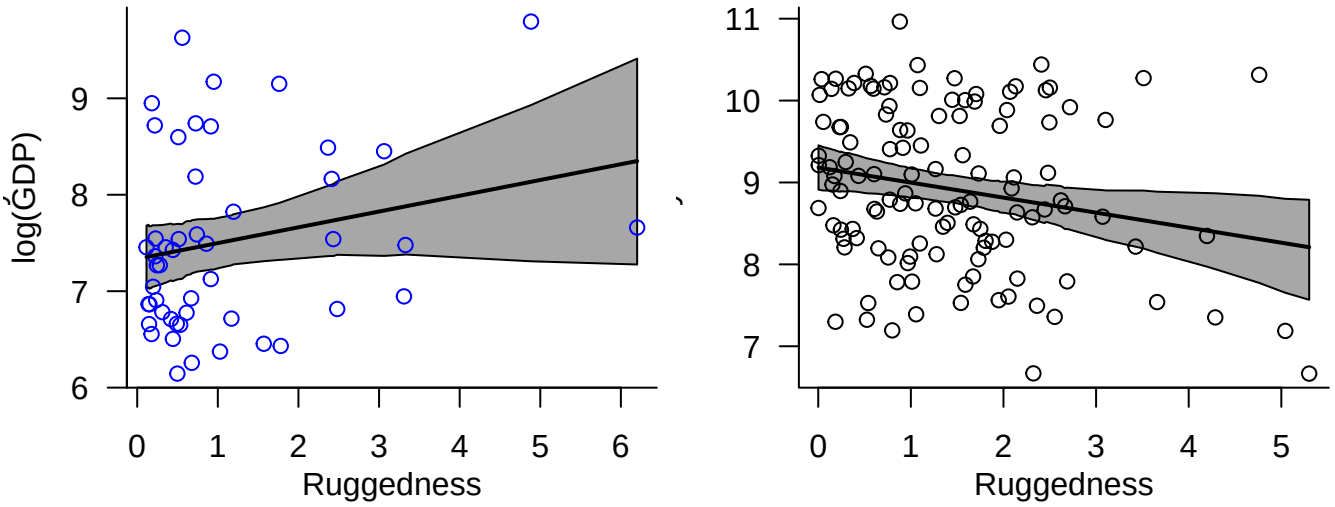
- Through the vagaries of Stan, you can't add together matrices and vectors. Therefore we coerce `gamma` $\times$ `R` into a vector using `to_vector()`.
- Alternatively we could just define `R` as a `row_vector` in the `data` block and skip the transpose.

Regardless of which model we use, we get the same results. If we run either model

```
# African linear regression
X <- model.matrix(~rugged*africa, data=rugged)[,2:4]

intDat <- list(nObs=nrow(rugged), obs=log(rugged$GDP),
  R=X[,"rugged"], A=X[,"africa"], aMu=0, aSD=20, bMu=0, bSD=1, sigmaSD=10)

intxnMod <- stan(file="13.intrxnMod.stan", data=intDat, iter=2000,
 chains=4, seed=867.5309)
mu <- as.matrix(intxnMod, "mu")
muHDI <- apply(mu, 2, HDI, credMass=0.95)
muMn <- colMeans(mu)
```



we find a positive relationship between `log(GDP)` and `ruggedness` for African nations and a negative

relationship for non-African nations.

Interpreting interaction terms

Interpreting interaction terms is not easy

- Plotting out the results is usually best for model inference.
- However, there are times when interpretation of the parameter estimates themselves is of value.

There are two reasons why interpreting tables of parameter estimates from models with interaction terms is challenging:

1. *Adding an interaction to the model changes parameter meanings.* Usually, the distribution of a "main effect" coefficient in an interaction model can not be directly compared to a term of the same name in a non-interaction model.
2. *Interpreting tables of interaction effects require thinking about parameter covariance.* This is a lot harder when the influence of a predictor depends on multiple parameters.

Adding an interaction to the model changes parameter meanings:

In a simple linear regression without interaction terms, each predictor variable is independent and directly measures that variable's influence.

- Not so for models with interaction terms

If we look at our likelihood function again,

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

a change in μ_i is dependent on a unit change in R_i is governed by γ_i.

- γ_i is a function of β_R, β_{AR}, and A_i; we need to know all three to interpret the effect of R_i on the outcome.
  - Only when $A_i = 0$ can we interpret the slope β_R because then $\gamma_i = \beta_R$.

Practically, this means interpreting tables of the estimates requires some math. If we want to estimate the effect of ruggedness on `log(GDP)` within Africa

```
round(summary(intxnMod, pars=c("alpha", "betaR", "betaA",
  "betaAR"), probs = c(0.025, 0.5, 0.975))$summary,2)
```

```
        mean se_mean   sd  2.5%   50% 97.5%   n_eff Rhat
alpha   9.18       0 0.14  8.91  9.18  9.45 1838.32    1
betaR  -0.18       0 0.08 -0.34 -0.18 -0.03 1954.19    1
betaA  -1.85       0 0.22 -2.27 -1.85 -1.40 1949.50    1
betaAR  0.35       0 0.13  0.10  0.34  0.60 2202.31    1
```

If we want to estimate the effect of ruggedness on `log(GDP)` within Africa,

$$\gamma = \beta_R + \beta_{AR}(1) = -0.18 + 0.35 = 0.17.$$

Outside of Africa,

$$\gamma = \beta_R + \beta_{AR}(0) = -0.18 = -0.18,$$

so the relationship is essentially reversed.

Interpreting tables of interaction effects require thinking about parameter covariance:

But those are only point estimates of the marginal values. If we really want to compare the effects between African and non-African countries, we have to consider the whole posterior distribution:

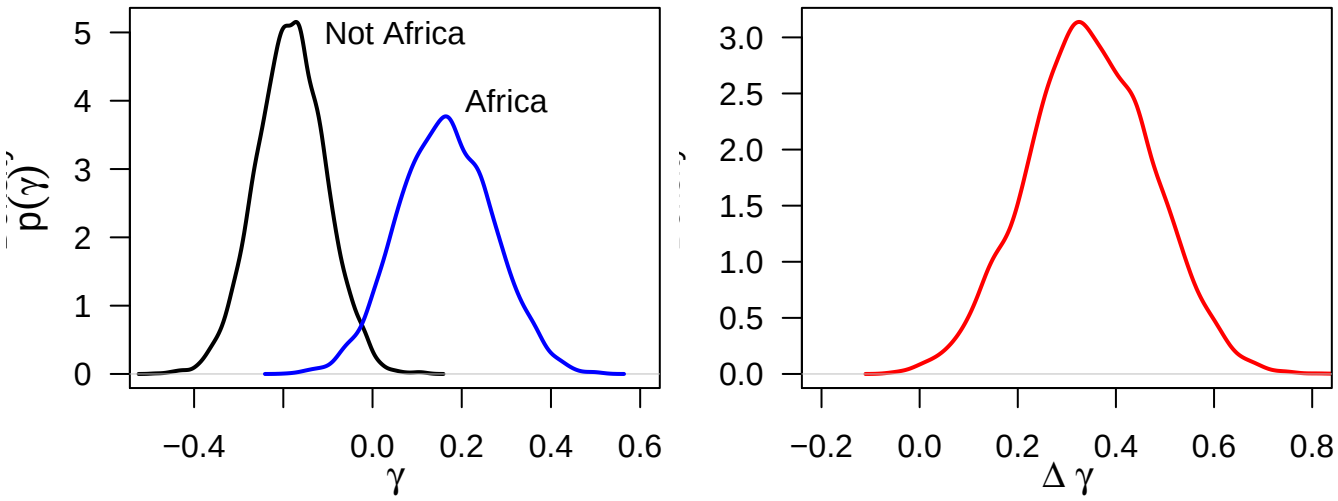
```
slopes <- as.matrix(intxnMod, pars=c("betaR", "betaAR"))
gammaA <- slopes[,"betaR"] + slopes[,"betaAR"]*1
gammaNA <- slopes[,"betaR"] + slopes[,"betaAR"]*0
mean(gammaA)
```

```
[1] 0.1635498
```

```
mean(gammaNA)
```

```
[1] -0.1834525
```

The means are almost identical to those above.



But we can also plot the full distributions together, or the difference in the distributions. Note that the proportion of the differences less than zero is only 0.00175

- much less than the overlap of the marginal distributions suggests.