Lecture 7.3: Basic linear regression (cont.)

** This lecture is based on chapter 3 of Statistical Rethinking by Richard McElreath.*

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

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

We also need to read in the data. As a reminder, we are looking at the relationship between algal biomass (mg) and algal secondary metabolite concentrations (`terpenes`; $mg \cdot g^{-1}$).

```
algae <- read.csv("algae.csv")
biomass <- algae$biomass
terpenes <- algae$terpenes
```

Our first regression

At the end of the last lecture, we added `terpenes` to the model as a predictor variable.

By adding terpenes (x_i), the model was reformulated as:

$$\begin{aligned}
BM_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta x_i \\
\alpha &\sim \text{Normal}(150, 100) \\
\beta &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Cauchy}^+(0, 10)
\end{aligned} \tag{1}$$

μ was no longer a paremter to be estimated but instead is a deterministic function of α & β—the intercept and the slope.

Our model, `modLM.stan`, was formulated as follows (note that there are several ways to do this):

```
data {
  int<lower=0> nObs;          // No. obs.
  vector[nObs] BM;   // biomass observations
  vector[nObs] terpenes;
  real<lower=0> aMean;       // mean of prior alpha
  real<lower=0> aSD;         // SD of prior alpha
  real<lower=0> bSD;          // SD of prior beta
  real<lower=0> sigmaSD;      // scale for sigma
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}

transformed parameters {
    // can be useful for plotting purposes
  vector[nObs] mu;
  mu = alpha + beta*terpenes;
}

model {
  alpha ~ normal(aMean, aSD);
  beta ~  normal(0, bSD);
  sigma ~ cauchy(0, sigmaSD);

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

I am going to add a `generated quantities` block to the model and simulate some "new" data (`newBM`) for later for posterior prediction (see below). This model block occurs after the `model` block in Stan.

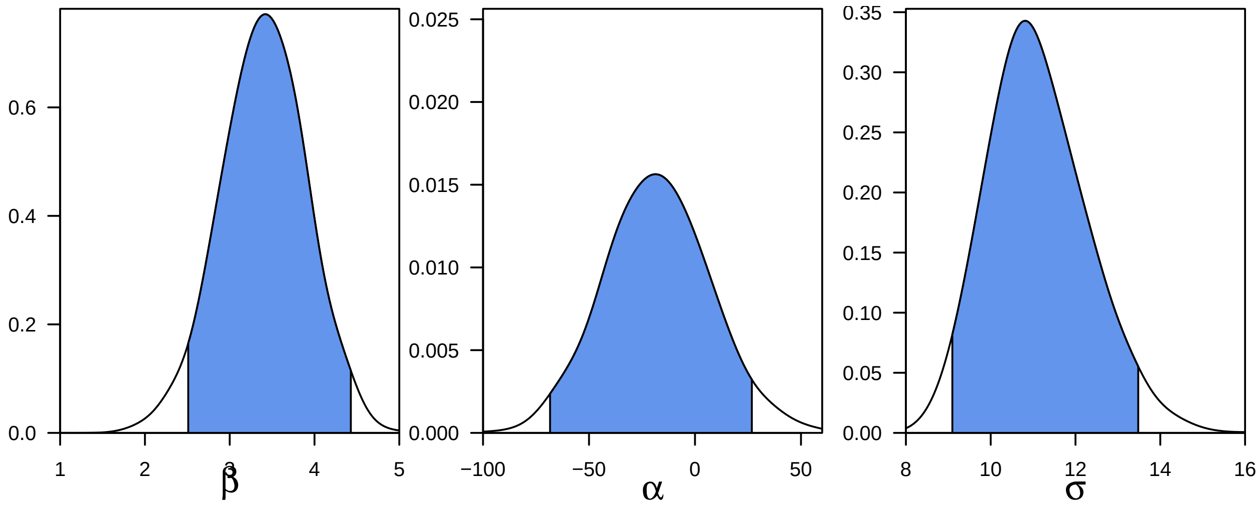
```
generated quantities {
  vector[nObs] newBM;

  for (n in 1:nObs)
    newBM[n] = normal_rng(mu[n], sigma);
}
```

```
datLM <- list(nObs=dim(algae)[1], BM=algae$biomass, terpenes=algae$terpenes,
              aMean=150, aSD=100, bSD=10, sigmaSD=10)

modLM <- stan(file="modLM.stan", data=datLM, iter=2000, chains=4, seed=3,
  pars=c("mu","newBM"), include=FALSE)

# extract posterior estimates
parLM <- as.matrix(modLM, pars=c("alpha", "beta", "sigma"))
```

The first thing we might want to do is print a summary of our results and look graphically.

- For brevity I am only going to look at the main parameters of interest and exclude μ.

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

plotInterval(parLM[,"beta"], HDI=TRUE, interval=0.95, xlims=c(1, 5),
  col="cornflowerblue", yOffset=0.01)
mtext(expression(paste(bold(beta))), side=1, line=2, cex=1.2)

plotInterval(parLM[,"alpha"], HDI=TRUE, interval=0.95, xlims=c(-100, 60),
  col="cornflowerblue", yOffset=0.01)
mtext(expression(paste(bold(alpha))), side=1, line=2, cex=1.2)

plotInterval(parLM[,"sigma"], HDI=TRUE, interval=0.95, xlims=c(8, 16),
  col="cornflowerblue", yOffset=0.01)
mtext(expression(paste(bold(sigma))), side=1, line=2, cex=1.2)
```



```
print(modLM, pars=c("alpha", "beta", "sigma"), digits.summary=2)
```

```
Inference for Stan model: modLM.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

        mean se_mean    sd   2.5%    25%    50%   75% 97.5% n_eff Rhat
alpha -17.80    0.70 24.33 -64.98 -34.67 -18.26 -1.72 32.58  1192    1
beta    3.40    0.01  0.49   2.37   3.08   3.41  3.74  4.35  1199    1
sigma  11.11    0.03  1.15   9.13  10.31  10.99 11.83 13.56  1446    1

Samples were drawn using NUTS(diag_e) at Mon Feb 27 12:42:03 2017.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

We have three parameters. Let's go through their interpretation one at a time.

1. The slope `beta`: a value of 3.4 can be interpreted as *an individual with 1 $mg \cdot g^{-1}$ higher terpene concentrations has 3.4 mg more biomass.
    - The 95% UI's suggest that β close to 2 or greater than ≈ 4.3 are improbable with these data and model.
    - Our prior was there was no relationship, but instead the model suggests a positive relationship between terpenes and biomass.
2. The intercept `alpha`: Our estimate suggests that a seaweed with no terpenes should have -17.8 mg biomass—impossible!
    - In it's raw form, the intercept is frequently uninterpretable without considering any $\beta's$.
3. The SD (σ): This tells us the widths of the distribution of biomass around the mean. One way to interpret is that ≈ 95% of the prob. of a normal distribution lies within 2 SD.
    - Thus, 95% of plausable biomasses lie within 22.22g of the mean biomass.
    - But there is uncertainty.

Centering

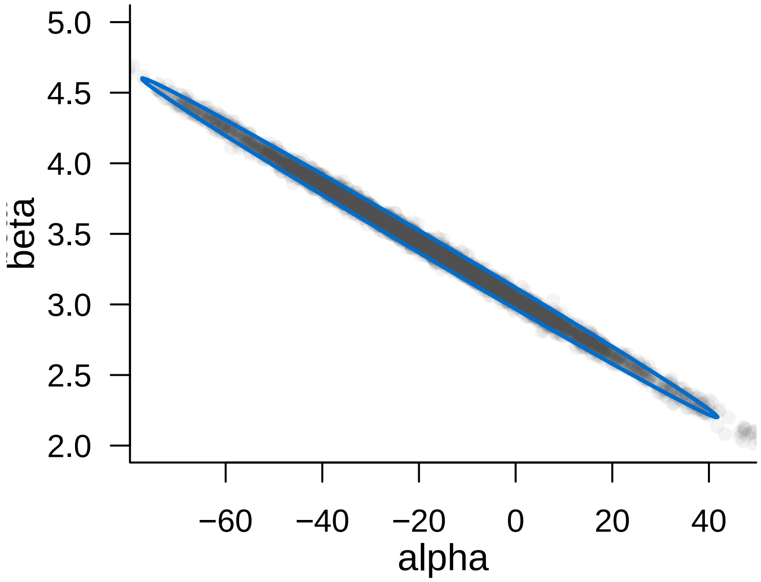
Notice that the uncertainty around α is considerable. When there are zero terpenes, the highest 95% of credible values for biomass lie between -65–33 g.

Investigating further, the correlation between α and β is almost perfect:

```r
round(cor(parLM[,1:2]), 3)
```

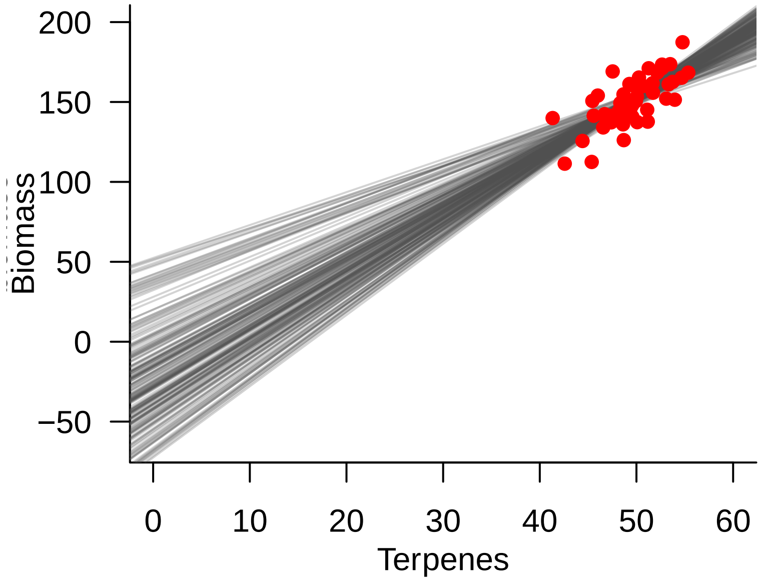
```
       alpha   beta
alpha  1.000 -0.998
beta  -0.998  1.000
```

```r
col <- "#50505010"

# Plot alpha and beta
plot(parLM[,1:2], pch=16, col=col, las=1, ylim=c(2,5),
      xlim=c(-75,45), bty="l")
dataEllipse(as.matrix(parLM[,1:2]),level=c(0.95), add=TRUE, labels=FALSE,
            plot.points=FALSE, center.pch=FALSE, col=c(col,"#006DCC"))
mtext(text = "beta", side=2, line=2.3, cex=1.2)
mtext(text = "alpha", side=1, line=2, cex=1.2)
```



If we plot a subset of the posterior estimates for our regression lines (μ), we can see why:

```
# Make an empty plot
plot(algae, type="n",xlim=c(0, 60), ylim=c(-65,200), las=1, bty="l")

# plot regression lines by looping through the first 200 posterior
# iterations of "alpha"
for(i in 1:200) {
  abline(a=parLM[i,"alpha"], b=parLM[i,"beta"], col="#50505040")
}
points(algae, pch=16, col="red")
mtext(text = "Biomass", side=2, line = 2.3, cex=1)
mtext(text = "Terpenes", side=1, line = 2, cex=1)
```



Because the means of the data are far from zero, small changes in the slope result in huge changes in the intercept.

- This leads to considerable uncertainty in the intercept and makes it's practical interpretation impossible.
- It also leads to less efficient models. Here it isn't too bad, but fitting complex models can become quite challenging.

One solution is to *center* the predictor variable(s) by subtracting each value from the mean.

```r
# center (but not scale terpenes)
terpC <- as.vector(scale(algae$terpenes, scale=FALSE))
datC <- list(nObs=dim(algae)[1], BM=algae$biomass,
  terpenes=terpC, aMean=150, aSD=100, bSD=10, sigmaSD=10)

modC <- stan(file="modLM.stan", data=datC, iter=2000, chains=4, seed=3)

# extract posterior estimates
parC <- as.matrix(modC, pars=c("alpha", "beta"))
mu <- as.matrix(modC, pars="mu")
newBM <- as.matrix(modC, pars="newBM")
```

If we look at the parameter estimates, β and σ haven't changed, but α now equals the mean `biomass` with much tighter UI's.

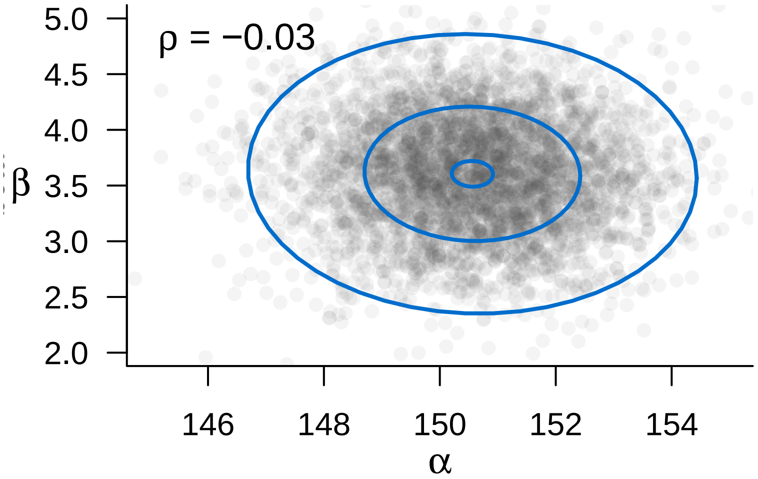
- Also note that the effective sample sizes more than doubled!

```r
print(modC, pars=c("alpha", "beta", "sigma"), digits_summary = 2)
```

```
Inference for Stan model: modLM.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

        mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
alpha 150.56    0.03 1.58 147.39 149.54 150.59 151.63 153.58  4000    1
beta    3.61    0.01 0.51   2.60   3.25   3.61   3.96   4.61  4000    1
sigma  11.08    0.02 1.14   9.13  10.29  11.00  11.76  13.62  4000    1

Samples were drawn using NUTS(diag_e) at Mon Feb 27 14:01:40 2017.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

If we plot the joint posterior of α & β, it is almost completely uncorrelated:

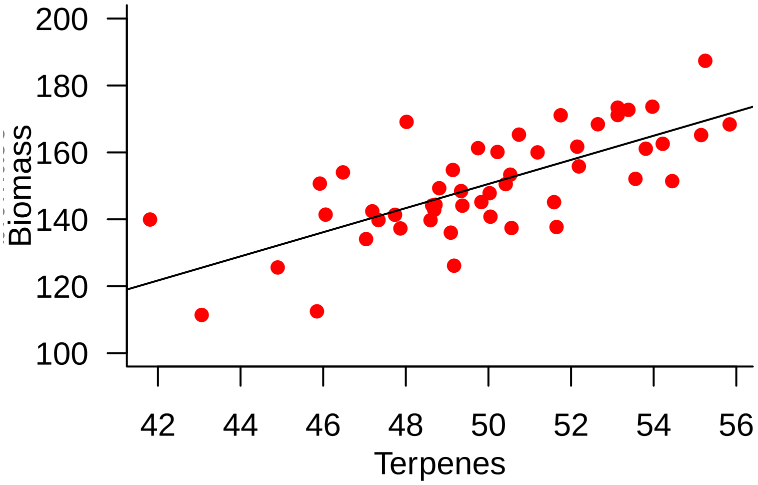
```r
# Plot alpha and beta
plot(parC, pch=16, col=col, las=1, ylim=c(2,5),
      xlim=c(145, 155), bty="l")
dataEllipse(parC,level=c(0.025, 0.5 ,0.95), add=TRUE, labels=FALSE,
            plot.points=FALSE, center.pch=FALSE, col=c(col,"#006DCC"))
mtext(text = expression(bold(alpha)), side=1, line=2, cex=1.2)
mtext(text = expression(bold(beta)), side=2, line=2.5, cex=1.2, las=1)
text(146.5, 4.8, expression(paste(rho, " = -0.03")),
  adj=c(0.5, 0.5),  cex=1.2)
```



Understanding uncertainty

To better understand how uncertainty plays into Bayesian regression, let's build up a plot a bit at a time. To start with, let's plot the data and our point estimate for the mean using the centered model `modC`.

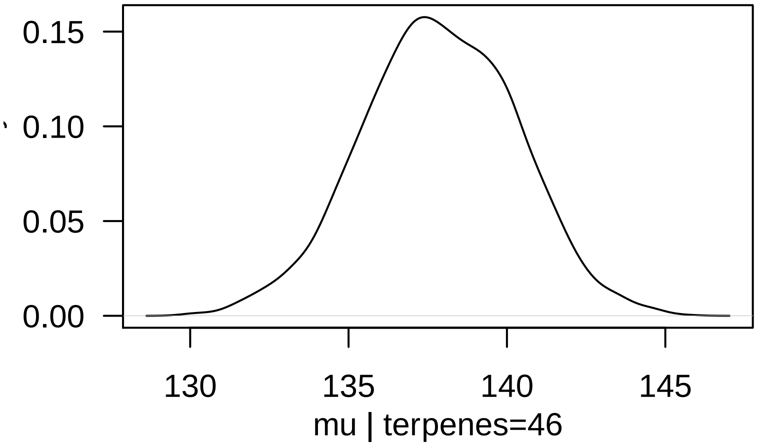
```
par(mar=c(3,3.2,0.1,0.5))

# Make an empty plot
plot(terpC, biomass, type="n", ylim=c(100,200), las=1, bty="l", xaxt="n")

# back transform the x-axis
axis(1, at=seq(-8, 6, by=2),
  labels=round((seq(-8, 6, by=2)+mean(terpenes))))
mtext(text = "Biomass", side=2, line = 2.3, cex=1)
mtext(text = "Terpenes", side=1, line = 2, cex=1)

# plot the data points and mean regression line
points(terpC, biomass, pch=16, col="red")
abline(a=mean(parC[,"alpha"]), b=mean(parC[,"beta"]))
```



However, this is just a point estimate. For each terpene measurement (terpene$_i$), we have a distribution of expected values of biomass$_i$ (i.e., μ_i).

- For example, terpene$_7 = 46$.

```
plot(density(mu[,7]), las=1, main="")
mtext(text = "mu | terpenes=46", side=1, line=2)
```

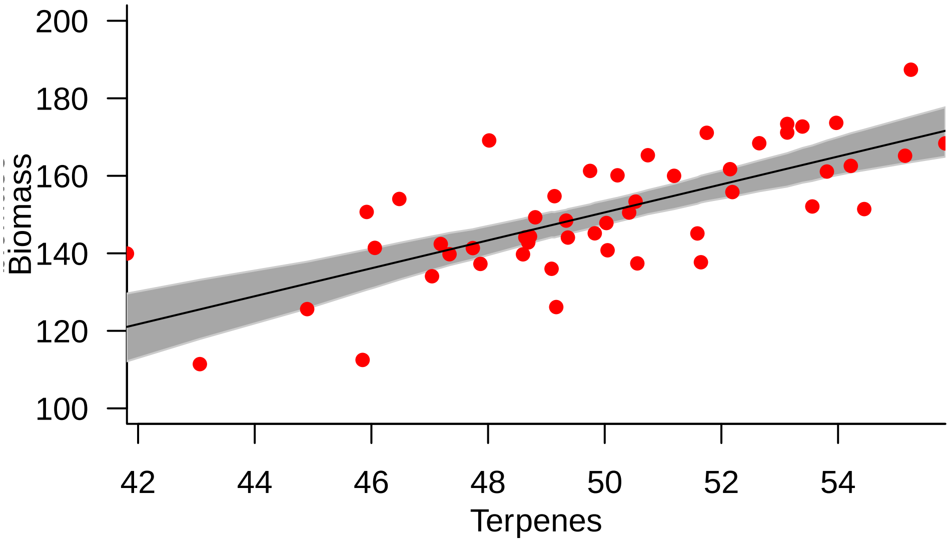


Because each posterior μ is a distribution, we can find intervals for it. Therefore, 95% of ways for the model to produce the data place the average biomass between ≈ 133–143, assuming the terpene concentration is 146 $mg \cdot g^{-1}$.

To draw the uncertainty around all of our $\mu's$, we just need to compute the upper and lower bounds:

```
muHDI <- apply(mu,2, HDI, credMass=0.95)

# Make an empty plot
plot(terpC, biomass, type="n", ylim=c(100,200), las=1,
  bty="l", xaxt="n", xaxs="i")

# back transform the x-axis
```

```
axis(1, at=seq(-8, 6, by=2),
  labels=round((seq(-8, 6, by=2)+mean(terpenes))))
mtext(text = "Biomass", side=2, line = 2.3, cex=1)
mtext(text = "Terpenes", side=1, line = 2, cex=1)

# plot uncertainty interval in mu as a polygon
polygon(x=c(terpC, rev(terpC)), y=c(muHDI[1,], rev(muHDI[2,])),
  col="#50505080", border="grey80")

# plot the data points and mean regression line
points(terpC, biomass, pch=16, col="red")
abline(a=mean(parC[,"alpha"]), b=mean(parC[,"beta"]))
```



Prediction intervals

The polygon we have added to the plot shows the 95% HDI for the expectation. In other words, *conditional on the assumption that biomass and terpene concentrations are related by a straight line we have plotted the most plausible line and the most plausible bounds.*

- This is just the average though.

To incorporate the uncertainty in the data (σ), we can use our estimates from the `generated quantities` part of our Stan model:

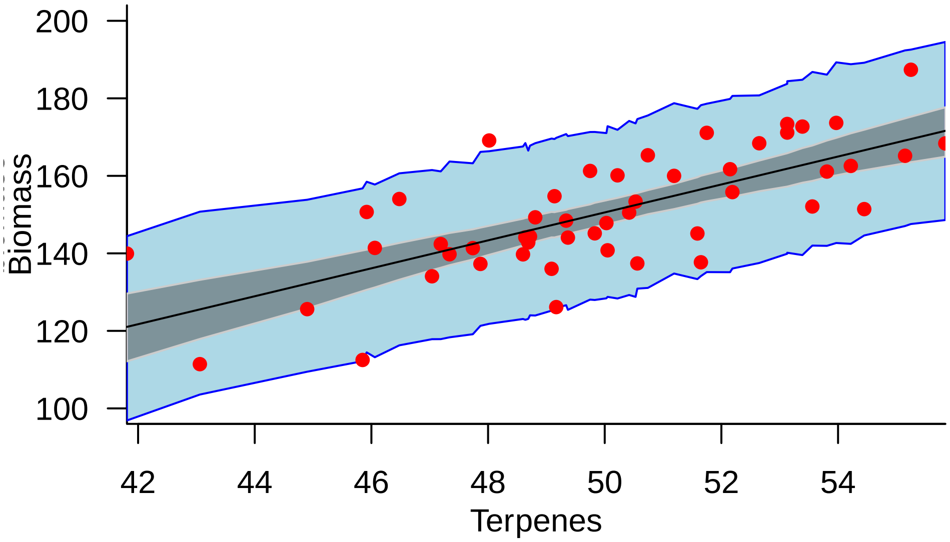
$$\tilde{y}_i \sim \text{Normal}(\mu_i, \sigma) \tag{2}$$

These are simulated biomass values that describe variation in the data, not average values.

```
bmHDI <- apply(newBM,2, HDI, credMass=0.95)
```

Now, we can build up the plot with our 1) point estimated line, 2) our 95% region of plausible μ, and 3) the boundries of simulated biomasses expected by the model.

```
# Make an empty plot
plot(terpC, biomass, type="n", ylim=c(100,200), las=1,
  bty="l", xaxt="n", xaxs="i")

# back transform the x-axis
axis(1, at=seq(-8, 6, by=2),
  labels=round((seq(-8, 6, by=2)+mean(terpenes))))
mtext(text = "Biomass", side=2, line = 2.3, cex=1)
mtext(text = "Terpenes", side=1, line = 2, cex=1)

# plot uncertainty interval in mu as a polygon
polygon(x=c(terpC, rev(terpC)), y=c(bmHDI[1,], rev(bmHDI[2,])),
  col="lightblue", border="blue")

# plot uncertainty interval in mu as a polygon
polygon(x=c(terpC, rev(terpC)), y=c(muHDI[1,], rev(muHDI[2,])),
  col="#50505080", border="grey80")

# plot the data points and mean regression line
points(terpC, biomass, pch=16, col="red")
abline(a=mean(parC[,"alpha"]), b=mean(parC[,"beta"]))
```



The difference between the two intervals is subtle but important. There are two distinct measures of uncertainty here even though they get blended together in the posterior simulation:

1. uncertainty in the parameter values

2. uncertainty in the sampling process.

The posterior distribution ranks the relative plausabilities of every combination of parameter values.

The distribution of simulated outcomes, however, is a distribution that incorporates sampling variation from some process that creates Gaussian random variables.