Here are four possible models for the bats and birds.

- We will focus attention on model 2.
- Later we will compare inference via simulation with classical inference.

```r
> bats = read.table("bats.txt", header = T)
> str(bats)
'data.frame': 20 obs. of 4 variables:
$ species: Factor w/ 16 levels "ColumbaLivia",..: 14 15 7 4 9 10 16 5 6 3 ...
$ mass : num 779 628 258 315 24.3 35 72.8 120 213 275 ...
$ type : Factor w/ 3 levels "bird","eBat",..: 3 3 3 3 1 1 1 1 1 1 ...
$ energy : num 43.7 34.8 23.3 22.4 2.46 3.93 9.15 13.8 14.6 22.8 ...

> bats0.lm = lm(log(energy) ~ 1, bats)
> bats1.lm = lm(log(energy) ~ log(mass), bats)
> bats2.lm = lm(log(energy) ~ log(mass) + type, bats)
> bats3.lm = lm(log(energy) ~ log(mass) * type, bats)
```

An ANOVA would prefer the regression model without type.

- But this does not help us to estimate things.

```r
> anova(bats0.lm, bats1.lm, bats2.lm, bats3.lm)
Analysis of Variance Table
Model 1: log(energy) ~ 1
Model 2: log(energy) ~ log(mass)
Model 3: log(energy) ~ log(mass) + type
Model 4: log(energy) ~ log(mass) * type
   Res.Df RSS Df Sum of Sq F Pr(>F)
Model 1: 19 0.5829
Model 2: 18 0.5533 1 0.0296 0.4100 0.6713
Model 3: 16 0.5049 2 0.0484 0.6718 0.5265
Model 4: 14 0.5049 2 0.0484 0.6718 0.5265
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 ' ' 1
```
Summary of Model 2

> display(bats2.lm, digits = 3)

```r
lm(formula = log(energy) ~ log(mass) + type, data = bats)
coef.est coef.se
(Intercept) -1.474 0.239
log(mass) 0.815 0.045
typeeBat -0.024 0.158
typenBat -0.102 0.114
```

---

n = 20, k = 4
residual sd = 0.186, R-Squared = 0.98

Energy Cost of Echolocation

- For any given mass, the estimated difference in log energy use for echolocating bats versus non-echolocating bats is:

\[
-0.024 - (-0.102) = 0.078
\]

- This implies the cost of echolocation is about 8%.

Specifically,

\[
\log(eBat energy) - \log(nBat energy) = 0.078
\]

\[
\log\left(\frac{eBat energy}{nBat energy}\right) = 0.078
\]

\[
\frac{eBat energy}{nBat energy} = e^{0.078} = 1.081
\]

- What is a confidence interval for this estimate?

Using the `sim()` Function

- The `arm` library has a function `sim()` that will use simulation to take samples of probable model parameters, given the observed data.

- The result of `sim()` is a list with two components:
  - `$beta` is a matrix of regression coefficients corresponding to the model matrix.
  - `$sigma` is an estimate of the standard deviation of the normal error.

- For most purposes, a simulation of about 1000 realizations of the model parameters is sufficient.

R Example

```r
> bats2.sim = sim(bats2.lm, 1000)
> dim(bats2.sim$beta)
[1] 1000 4
> length(bats2.sim$sigma)
[1] 1000
> bats2.sim$beta[1:3, ]
     (Intercept) log(mass) typeeBat typenBat
[1,] -1.9228092 0.8949050 0.3866206 -0.008195968
[2,] -1.8814576 0.8767998 0.3474210 -0.144067220
[3,] -0.9706024 0.7105570 -0.1658778 0.074913204
> bats2.sim$sigma[1:3]
[1] 0.1648946 0.2218280 0.2387390
```
Uncertainty in Regression Coefficients

- The difference between the third and fourth columns of `bats2.sim$beta` are estimates of the difference in log energy use for the two types of bats.
- We calculate this difference and summarize it in various ways.

```r
> mean(bats2.diff)
[1] 0.08070429
> exp(mean(bats2.diff))
[1] 1.08405
> quantile(bats2.diff, c(0.025, 0.975))
2.5%   97.5%  
-0.3231736  0.4872756
> exp(quantile(bats2.diff, c(0.025, 0.975)))
[1] 0.907 1.084
```

We can check if the estimated standard errors are similar to simulation estimates.

```r
apply() applies a function to the rows (1) or columns (2) of a matrix.

> display(bats2.lm, digits = 3)

lm(formula = log(energy) ~ log(mass) + type, data = bats)

Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -1.474 | 0.239 | -6.167 1.35e-05 *** |
| log(mass) | 0.815 | 0.046 | 18.297 3.76e-12 *** |
| typeeBat | -0.024 | 0.158 | -0.150 0.883 |
| typenBat | -0.102 | 0.114 | -0.906 0.364 |

---

Residual standard error: 0.186 on 16 degrees of freedom
Multiple R-Squared: 0.9815, Adjusted R-squared: 0.9781

> 2 * sum(bats2.sim$beta[, 3] > 0)/1000
[1] 0.866
> 2 * sum(bats2.sim$beta[, 4] > 0)/1000
[1] 0.376
```
### Prediction Intervals

- Suppose we wanted to know a 95% prediction interval for the energy use of a 150 gram bird.
- We could use `predict()` or the simulation.

```r
x.new = data.frame(mass = 150, type = "bird")
pred.1 = bats2.sim$beta %*% x.new + rnorm(1000, 0, bats2.sim$sigma)
```

```
  fit  lwr  upr
[1,]  2.609357  2.198517 3.020196
```

```r
exp(predict(bats2.lm, x.new, interval = "prediction"))
```

```
  fit  lwr  upr
[1,] 13.59030  9.011636 20.49532
```

### Simulation for Prediction

- Multiply the $1000 \times 4$ beta matrix by the $4 \times 1$ predictor vector and add to this random error using the simulated $\sigma$.

```r
x.1 = c(1, log(150), 0, 0)
pred.1 = bats2.sim$beta %*% x.1 + rnorm(1000, 0, bats2.sim$sigma)
quantile(pred.1, c(0.025, 0.975))
```

```
       2.5%        97.5%    
   2.020283    2.989556
```

### Other Predictions

- Suppose we had a new 400 g bird species and a new 400 g non-echolocating bat and we wanted to predict the difference in energy use.

```r
x.2 = c(1, log(400), 0, 0)
x.3 = c(1, log(400), 0, 1)
pred.2 = exp(bats2.sim$beta %*% x.2 + rnorm(1000, 0, + bats2.sim$sigma))
pred.3 = exp(bats2.sim$beta %*% x.3 + rnorm(1000, 0, + bats2.sim$sigma))
quantile(pred.2 - pred.3, c(0.025, 0.975))
```

```
  2.5%        97.5%    
-16.167886    20.13797
```

### Graph

#### 400 g Bird – 400 g non-echolocating Bat

![Graph showing the difference in energy use between a 400 g bird and a 400 g non-echolocating bat. The graph displays a density plot with the difference in energy use ranging from -40 to 40 watts, and the density values range from 0.00 to 0.05. The plot includes a normal distribution curve representing the predicted energy use.]
Confidence Intervals for Regression

- We can also use the simulation for confidence intervals of the regression lines.
- Here is how to find a 95% confidence interval for the energy use of a 400 g bird.

```r
> x.new = data.frame(mass = 400, type = "bird")
> exp(predict(bats2.lm, x.new, interval = "confidence"))

     fit lwr  upr
[1,] 30.22564 26.41756 34.58266

> conf.2 = exp(bats2.sim$beta %*% x.2)
> quantile(conf.2, c(0.5, 0.025, 0.975))

     50%    2.5%   97.5%
30.22653 26.34265 34.44575
```

Graph

Mean of 400 g Bird

```
> mean_energy = bats2.sim$beta %*% x.2
> density_energy = density(mean_energy)

> plot(density_energy, main = "Mean of 400 g Bird", xlab = "Energy (watts)", ylab = "Density")
```

```r
summarizing_linear_regression <- data.frame(energy = bats2.sim$beta %*% x.2, density = density(mean_energy)
```