Module 5 Lab Examples

Introduction

This lab covers the following topics.

1. Confidence and prediction intervals
2. Manually performing a Sum of Squares F-test
3. Using `anova()` to perform a Sum of Squares F-test
4. Using model selection criteria functions `AIC()` and `BIC()` to compare models
5. Exploring the usefulness of R^2 for model comparison

You'll explore these concepts in the context of a study looking at the energy costs of echo-location in bats. The data is available in the `Sleuth3` package, in the object `case1002`.

```
case1002
```

```
##     Mass                   Type Energy
## 1  779.0  non-echolocating bats  43.70
## 2  628.0  non-echolocating bats  34.80
## 3  258.0  non-echolocating bats  23.30
## 4  315.0  non-echolocating bats  22.40
## 5   24.3 non-echolocating birds   2.46
## 6   35.0 non-echolocating birds   3.93
## 7   72.8 non-echolocating birds   9.15
## 8  120.0 non-echolocating birds  13.80
## 9  213.0 non-echolocating birds  14.60
## 10 275.0 non-echolocating birds  22.80
## 11 370.0 non-echolocating birds  26.20
## 12 384.0 non-echolocating birds  25.90
## 13 442.0 non-echolocating birds  29.50
## 14 412.0 non-echolocating birds  43.70
## 15 330.0 non-echolocating birds  34.00
## 16 480.0 non-echolocating birds  27.80
## 17  93.0      echolocating bats   8.83
## 18   8.0      echolocating bats   1.35
## 19   6.7      echolocating bats   1.12
## 20   7.7      echolocating bats   1.02
```

```
?case1002
```

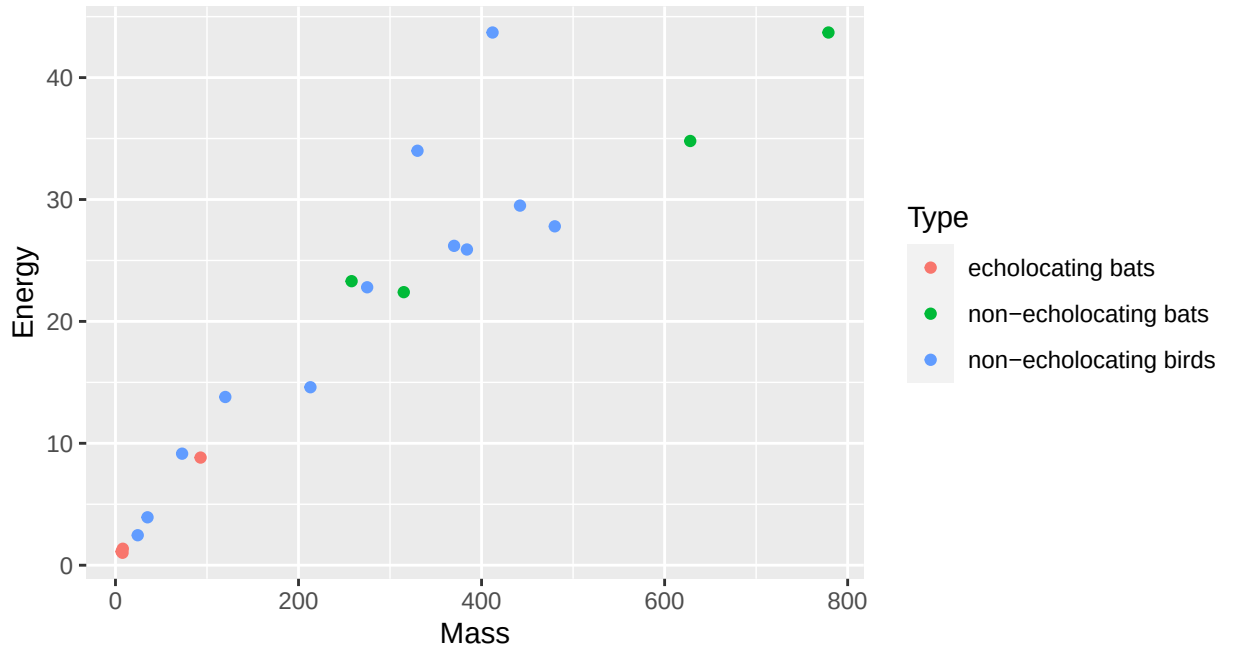
Here's the context: Some bats use echolocation to orient themselves. Zoologists have hypothesized the energy costs of echolocation during flight are the sum of the energy cost of flight plus the energy cost of echolocation. If they can show bats that echolocate use about the same energy as non-echolocating bats, they have evidence these bats have evolved to echo-locate efficiently. But there is a complication, the energy costs of flight depend on how heavy you are.

The data are on in–flight energy expenditure and body mass from 20 energy studies on three types of flying vertebrates: echolocating bats, non–echolocating bats and non–echolocating birds.

Exploratory Analysis

Let's take a quick look at what we have:

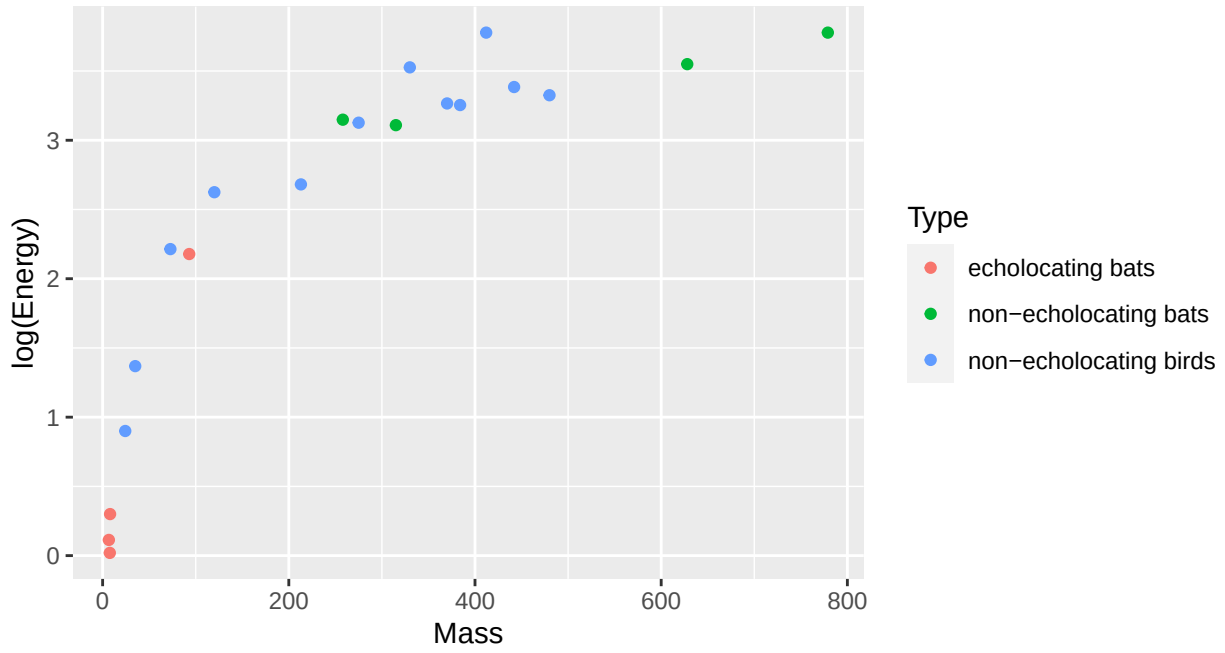
```
qplot(Mass, Energy, data = case1002, color = Type)
```



Notice the general relationship, that increasing mass is associated with increasing energy costs. It's a little hard to see any substantial difference between the three types. You might notice there seems to be a lot more variation in energy with higher mass, this is often a sign a transformation might result in a cleaner relationship. Let's try looking at `log(Energy)` instead.

Plot `Mass` vs. `log(Energy)`.

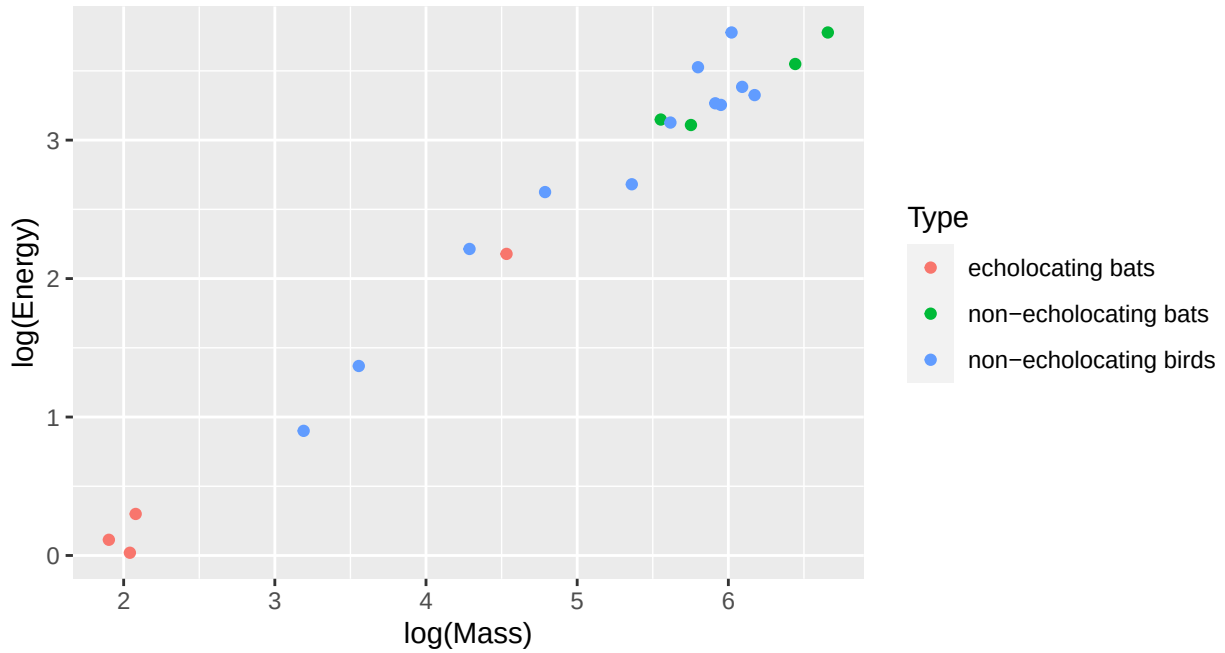
```
qplot(Mass, log(Energy), data = case1002, color = Type)
```



The variation looks a little more constant, but the relationship between energy and mass looks rather non-linear. Let's try `log(Mass)` as well:

Plot `log(Mass)` vs. `log(Energy)`.

```
qplot(log(Mass), log(Energy), data = case1002, color = Type)
```



Much better: nice linear relationship, and constant variance.

Multiple Linear Regression

We are going to be interested in three models.

Model 1: Both birds and non-echolocating bats have possibly different energy costs in flight to echolocating bats, after accounting for a linear relationship between log energy and log mass,

$$\log(Energy_i) = \beta_0 + \beta_1 \log(Mass_i) + \beta_2 non\text{-}ebat_i + \beta_3 bird_i + \epsilon_i$$

Model 2: The energy costs for non-echolocating bats and echolocating bats is the same, but possibly different to birds, after accounting for a linear relationship between log energy and log mass,

$$\log(Energy_i) = \beta_0 + \beta_1 \log(Mass_i) + \beta_3 bird_i + \epsilon_i$$

Model 3: All types have the same energy costs in flight, after accounting for a linear relationship between log energy and log mass,

$$\log(Energy_i) = \beta_0 + \beta_1 \log(Mass_i) + \epsilon_i$$

where $non\text{-}ebat_i$ is an indicator variable that the `Type` of the i^{th} observation is a non-echo-locating bat, and $bird_i$ is an indicator variable that the `Type` of the i^{th} observation is a bird.

Fit all three of the models specified above using `lm()` and store them in the variables `mod1`, `mod2`, and `mod3`, respectively.

Note: Model 2 is a little tricky, we need to create a new indicator that is `TRUE` only for `"non-echolocating birds"`.

```
mod1 <- lm(log(Energy) ~ log(Mass) + Type, data = case1002)
mod2 <- lm(log(Energy) ~ log(Mass) +
    I(Type == "non-echolocating birds"), data = case1002)
mod3 <- lm(log(Energy) ~ log(Mass), data = case1002)
```

We don't actually have to fit model 2 to answer our question of interest. Since Model 2 only differs from Model 1 by dropping the β_2 term, we can ask if $\beta_2 = 0$ in Model 1, to decide if Model 2 is more appropriate.

We fit it here to demonstrate the difference between the fit of the two models, which we'll do by examining their predictions.

Prediction and Confidence Intervals

If you want prediction and/or confidence intervals for the mean response variable value at new explanatory variable values, the procedure is similar to the one we used for SLR. You need to specify a new data frame containing the values of the variables of interest, then use `predict()`. Let's compare the predictions between Model 1 and Model 2, for echo-locating and non-echo-locating bats of the same weight:

First we will make a new data frame to use as new data.

```
same_weight_bats <- data.frame(
  Type = rep(c("echolocating bats", "non-echolocating bats"), 3),
  Mass = rep(c(100, 400, 700), each = 2)
)
same_weight_bats
```

```
##                    Type Mass
## 1     echolocating bats  100
## 2 non-echolocating bats  100
```

```
## 3     echolocating bats  400
## 4 non-echolocating bats  400
## 5     echolocating bats  700
## 6 non-echolocating bats  700
```

You must be careful to keep the column names exactly the same as those in the input data to `lm()` and any categorical variables must use the same values. OK, let's take a look at the predictions for `mod1`:

```
cbind(same_weight_bats, pred_log_energy = predict(mod1, newdata = same_weight_bats))
```

```
##                     Type Mass pred_log_energy
## 1     echolocating bats  100        2.255321
## 2 non-echolocating bats  100        2.176658
## 3     echolocating bats  400        3.385092
## 4 non-echolocating bats  400        3.306429
## 5     echolocating bats  700        3.841155
## 6 non-echolocating bats  700        3.762492
```

In the first model, notice first that the predictions for echo and non-echo locating bats of the same weight are different, and second they always differ by the same amount (i.e. `2.255321-2.176658 = 3.385092-3.306429 = 0.078663`). This amount is our estimate of β_2, which you can confirm by examining `coef(mod1)`.

Do the same as above for `mod2`: make predictions for the log-energy-usage using `mod2`.

```
cbind(same_weight_bats, pred_log_energy = predict(mod2, newdata = same_weight_bats))
```

```
##                     Type Mass pred_log_energy
## 1     echolocating bats  100        2.212898
## 2 non-echolocating bats  100        2.212898
## 3     echolocating bats  400        3.324432
## 4 non-echolocating bats  400        3.324432
## 5     echolocating bats  700        3.773134
## 6 non-echolocating bats  700        3.773134
```

In the second model, β_2 has been forced to zero and the predictions for echo and non-echo locating bats of the same weight are always the same.

These predictions are on the transformed response scale. To go back to energy (rather than log energy) you can backtransform the predictions (and confidence interval endpoints if needed):

```
cbind(same_weight_bats, pred_energy = exp(predict(mod1, newdata = same_weight_bats)))
```

```
##                     Type Mass pred_energy
## 1     echolocating bats  100    9.538359
## 2 non-echolocating bats  100    8.816789
## 3     echolocating bats  400   29.520721
## 4 non-echolocating bats  400   27.287500
## 5     echolocating bats  700   46.579266
## 6 non-echolocating bats  700   43.055579
```

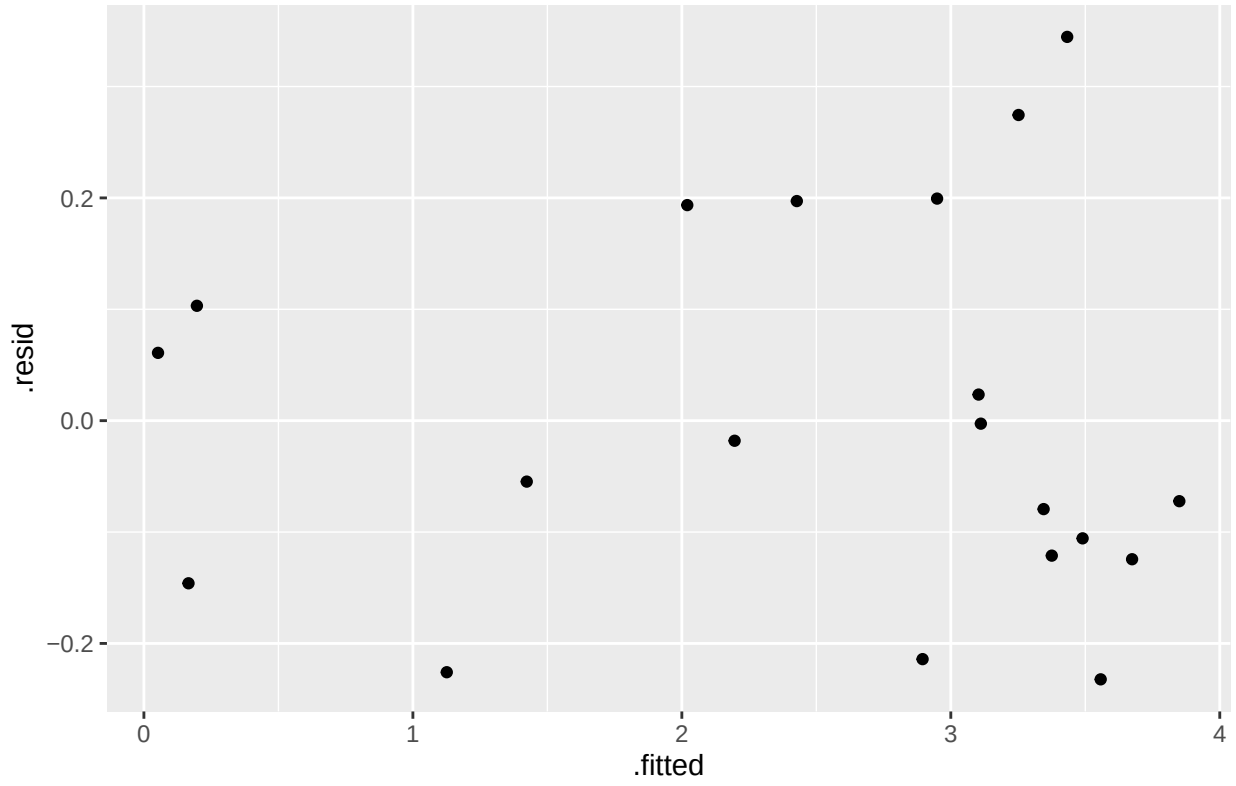
If you need a confidence interval or prediction interval, you can add the `interval` argument as usual.

Diagnostics

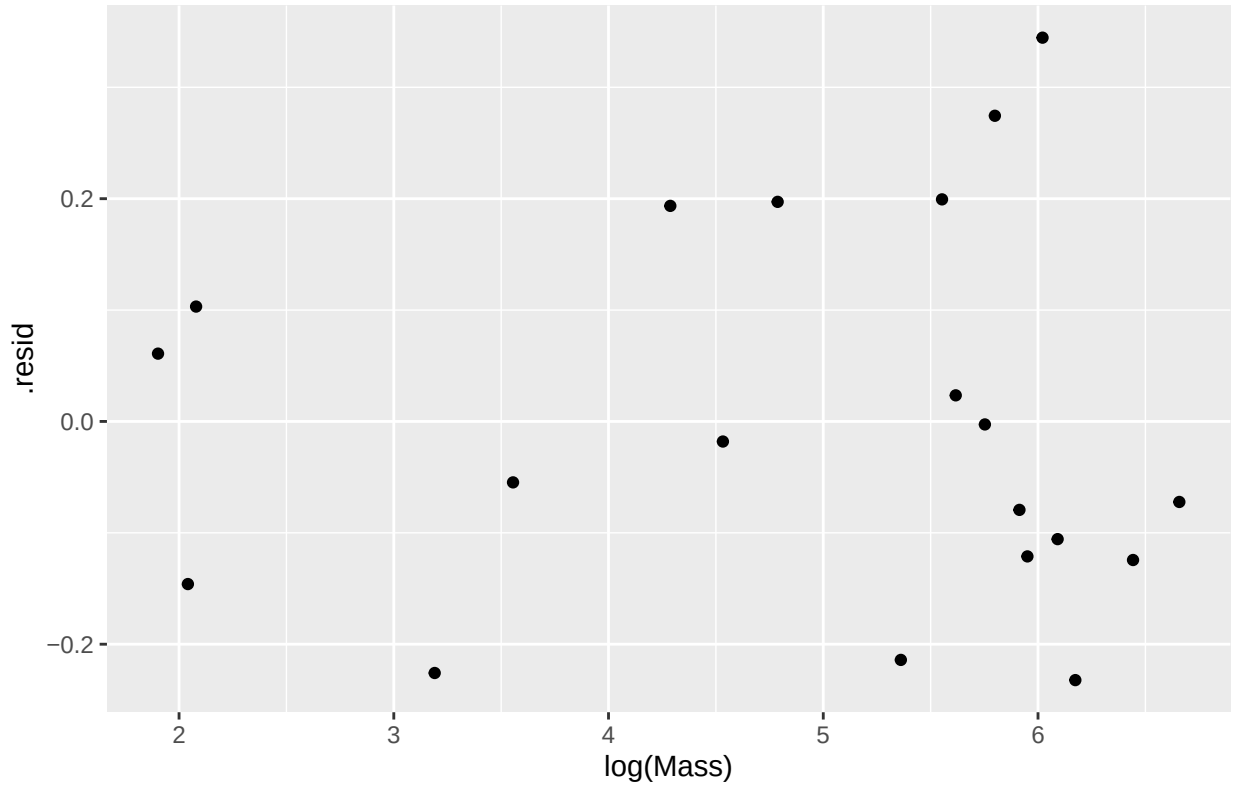
Hopefully you are convinced that we can answer the scientific question of interest with inference on β_2 in Model 1. Before we do, we should take a look at the fit of the models. You should generally do these checks on the most complicated model you are considering.

**Use `augment()` on `mod1` to create a data frame with diagnostics. Save it to the variable `case1002_diags`. Then plot each `.fitted`, `'log(Mass)`, and `Type` against `.resid`.

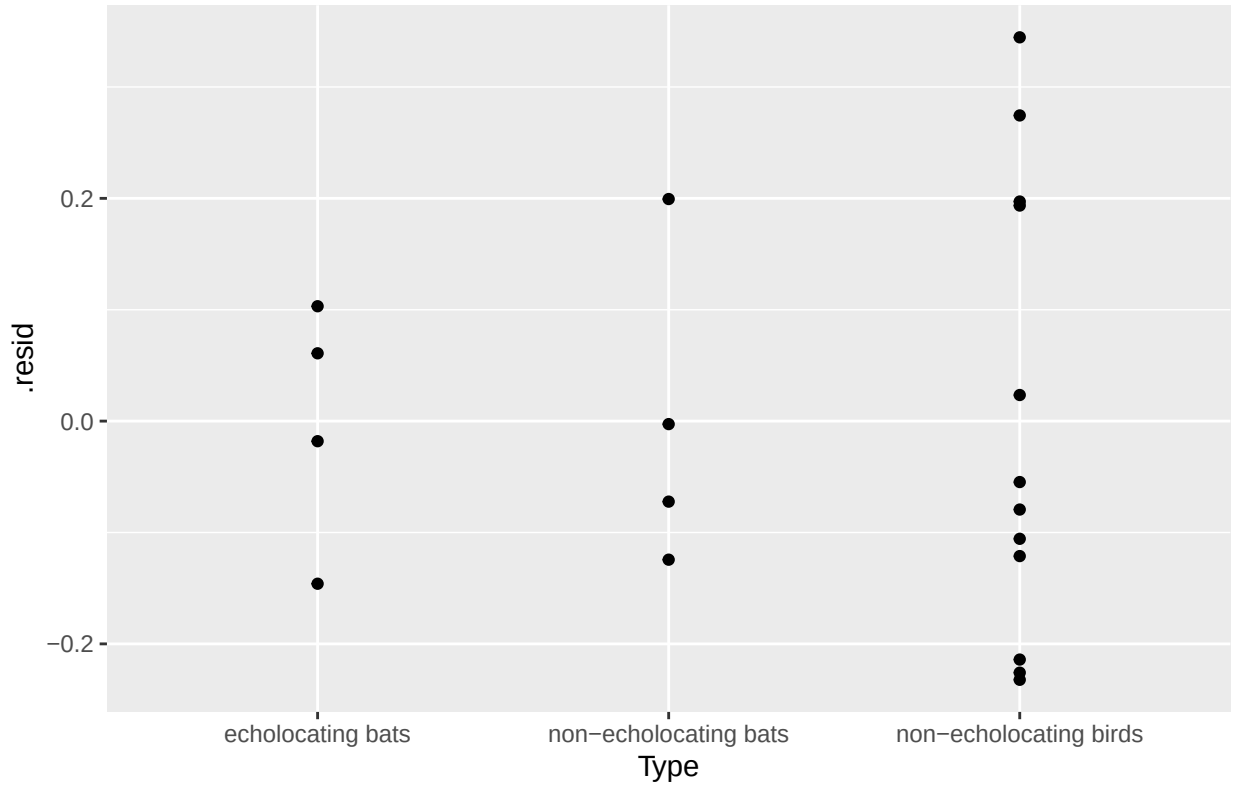
```
case1002_diag <- augment(mod1, case1002)
qplot(.fitted, .resid, data = case1002_diag)
```




```
qplot(log(Mass), .resid, data = case1002_diag)
```




```
qplot(Type, .resid, data = case1002_diag)
```



To review, we are looking for signs that our model assumptions–linearity, normality, independence, constant variance–may have been violated. If they have, it could show up as systematic patterns in the residual plots.

The one plot that may raise eyebrows in this example is the residuals versus Type. It looks like the residuals for non-echolocating birds are much more spread out. However, there is not too much cause for concern: since there are many more birds than either kind of bats, you are more likely to see extremes just because you have more observations for that group.

Inference on single parameters

Take a look at the summary of Model 1

```
summary(mod1)
```

```
##
## Call:
## lm(formula = log(Energy) ~ log(Mass) + Type, data = case1002)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23224 -0.12199 -0.03637  0.12574  0.34457
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.49770    0.14987  -9.993 2.77e-08 ***
## log(Mass)                   0.81496    0.04454  18.297 3.76e-12 ***
## Typenon-echolocating bats  -0.07866    0.20268  -0.388    0.703
## Typenon-echolocating birds  0.02360    0.15760   0.150    0.883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.186 on 16 degrees of freedom
## Multiple R-squared:  0.9815, Adjusted R-squared:  0.9781
## F-statistic: 283.6 on 3 and 16 DF,  p-value: 4.464e-14
```

The output includes the t-tests for the null hypothesis that each parameter is zero. This isn't always the hypothesis of interest, but in this case, the scientific question can be answered by examining the row for β_2, the parameter associated with the non-echolocating bat indicator variable.

There is no evidence that non-echolocating bats have a different energy cost in flight to echo-locating bats, after accounting for body mass (p-value = 0.70 from t-test echo-locating indicator variable in a regression of log energy on log mass and type).

Confidence intervals for model parameters can be obtained with the `confint()` function. **Use it on `mod1`.**

```
confint(mod1)
```

```
##                                2.5 %     97.5 %
## (Intercept)                -1.8154046 -1.1799884
## log(Mass)                   0.7205339  0.9093811
## Typenon-echolocating bats  -0.5083245  0.3509972
## Typenon-echolocating birds -0.3104999  0.3576964
```

With 95% confidence, the mean log energy cost for non-echolocating bats is between 0.51 units lower and 0.35 units higher than echolocating bats.

Inference on more than one parameter

You might be interested in comparing Model 1 to Model 3. Since these two models differ by two parameters the appropriate comparison uses and Extra Sum of Squares F-test.

A Sum of Squares F-test allows us to compare any two nested models, (which we typically call the "reduced model" and a more complex "full model"), where the reduced model is "nested" in the full model. Nested means that the full model has all predictors the reduced model has, plus additional predictors. Put another way, the predictors in the reduced model are a subset of the predictors in the full model. Note that this is the only setting where we can use an F-test.

Sum of Squares F-test

Basically, we want to compare the proportion of total variability explained by each model, and ask if the increase in explained variability is sufficient to justify adding the additional term/s into the model. Equivalently, is the *decrease* in *unexplained* variability sufficient to justify the more complex model? The F-statistic contains the information we need, and its distribution has an answer.

$$F = \frac{(RSS_{red} - RSS_{full})/(df_{red} - df_{full})}{RSS_{full}/df_{full}} \sim F_{(df_{red}-df_{full},\, df_{full})}$$

"RSS" stands for Residual Sum of Squares, "df" stands for degrees of freedom, and the subscripts denote the full and reduced model. The following code calculates the necessary pieces, and then performs a Sum of Squares F-test.

```
rss1 <- deviance(mod1) # Model 1 RSS
rss3 <- deviance(mod3) # Model 3 RSS
df1 <- df.residual(mod1) # Model 1 Residual Degrees of Freedom
df3 <- df.residual(mod3) # Model 3 Residual Degrees of Freedom
fstat <- ((rss3 - rss1)/(df3 - df1))/(rss1/df1) # F-statistic
1 - pf(fstat, df3 - df1, df3) # p-value
```

```
## [1] 0.6585362
```

The first two lines extract the residuals sum of squares from each model fit. The next two lines extract the residual degrees of freedom from each model. The next line calculates the F-statistic, and the final line calculates the p-value. To review, the p-value is the probability of seeing an F-statistic this unlikely, or more unlikely, if the null hypothesis (simple model) is true. What do we conclude?

There is no evidence for the more complicated model. The extra terms aren't worth it. You can probably guess, there is an easier way to perform the same test.

```
anova(mod3, mod1) # Sum of Squares F-test
```

```
## Analysis of Variance Table
##
## Model 1: log(Energy) ~ log(Mass)
## Model 2: log(Energy) ~ log(Mass) + Type
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     18 0.58289
## 2     16 0.55332  2  0.029574 0.4276 0.6593
```

The analysis of variance table contains all the information we just calculated. Conveniently, the two models being compared are provided in the output (note that `Model 1` is just the first model object you provided–in this case, `mod3`; likewise, `Model 2` is the second model object you provided–in this case, `mod1`). After the summary of the models being compared, the output displays the ANOVA table. The columns give the residual degrees of freedom, residual sum of squares, difference in degrees of freedom (numerator degrees of freedom), difference in sum of squares between the two models, F-statistic, and p-value.

Perform the Sum of Squares F-test on `mod2` and `mod1`.

```
anova(mod2, mod1)
```

```
## Analysis of Variance Table
##
## Model 1: log(Energy) ~ log(Mass) + I(Type == "non-echolocating birds")
## Model 2: log(Energy) ~ log(Mass) + Type
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     17 0.55853
## 2     16 0.55332  1 0.0052094 0.1506  0.703
```

What other ways can we compare two or more models?

Model Comparison Criteria

An analyst may want additional model comparison criteria. Two commonly used tools are the Akaike information criterion (AIC), and the Bayesian information criterion (BIC). The details of these measures are beyond the scope of this lab, but suffice it to say that the lower the score for each of these two criteria, the better the fit (**discussion of "better" below). A strength of AIC and BIC, in contrast to the F-test, is that not only can we compare more than two models simultaneously, but we can compare models that are not nested.

```
AIC(mod1, mod2, mod3)
```

```
##      df       AIC
## mod1  5 -4.993569
## mod2  4 -6.806154
## mod3  3 -7.952201
```

```
BIC(mod1, mod2, mod3)
```

```
##      df         BIC
## mod1  5 -0.01490754
## mod2  4 -2.82322519
## mod3  3 -4.96500445
```

Both criteria agree with the F-test: the simplest of the three models is sufficient. We know this because `mod3` has the lowest AIC and BIC scores.

**What does "better" mean to AIC and BIC? There is not a simple answer, but there are a few ideas worth mentioning. AIC and BIC place value on model "parsimony," in the Occam's Razor sense of the word. If two models explain variation in the response equally well, then AIC and BIC tend to prefer the simpler model (with fewer terms). AIC and BIC implement this preference by penalizing a model (that is, giving it a higher, less "desirable" score) for extra predictors. This preference aids interpretation; the greater the

number of terms in a model, typically the more difficult it is to understand and interpret the meaning of those terms. For example, imagine interpreting a marginally useful squared three-way interaction term! However, this preference for simpler, more interpretable models makes AIC and BIC somewhat ill-suited to model selection for prediction. If prediction is your ultimate goal, and that squared three-way interaction term improves prediction, then so be it!

R^2 - Coefficient of Determination

What about R^2 as a model comparison metric? R^2 is sometimes called the "Coefficient of Determination," and described as the proportion of total variation in the response explained by the model.

Revisiting the models we fit, R^2 did increase with the addition of `Type`:

```
summary(mod1) # Multiple R-squared:  0.9815
```

```
##
## Call:
## lm(formula = log(Energy) ~ log(Mass) + Type, data = case1002)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23224 -0.12199 -0.03637  0.12574  0.34457
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.49770    0.14987  -9.993 2.77e-08 ***
## log(Mass)                   0.81496    0.04454  18.297 3.76e-12 ***
## Typenon-echolocating bats  -0.07866    0.20268  -0.388    0.703
## Typenon-echolocating birds  0.02360    0.15760   0.150    0.883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.186 on 16 degrees of freedom
## Multiple R-squared:  0.9815, Adjusted R-squared:  0.9781
## F-statistic: 283.6 on 3 and 16 DF,  p-value: 4.464e-14
```

```
summary(mod3) # Multiple R-squared:  0.9806
```

```
##
## Call:
## lm(formula = log(Energy) ~ log(Mass), data = case1002)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21143 -0.14422 -0.04284  0.09681  0.37695
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.46826    0.13716  -10.71  3.1e-09 ***
## log(Mass)    0.80861    0.02684   30.13  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.18 on 18 degrees of freedom
## Multiple R-squared:  0.9806, Adjusted R-squared:  0.9795
## F-statistic: 907.6 on 1 and 18 DF,  p-value: < 2.2e-16
```

Does this mean `mod1` is superior to `mod3`? To explore this answer, consider adding another explanatory variable, but this time a completely meaningless one. We will generate Normal random variables with the same mean as the response variable `log(Energy)`, and a standard deviation of one. What will happen to R^2 if we include this predictor in the model?

```
set.seed(12345)
noise <- rnorm(nrow(case1002), mean(log(case1002$Energy)), 1)
mod4 <- lm(log(Energy) ~ log(Mass) + Type + noise, data = case1002)
summary(mod4) # Multiple R-squared:  0.9823
```

```
##
## Call:
## lm(formula = log(Energy) ~ log(Mass) + Type + noise, data = case1002)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22849 -0.10940 -0.05392  0.12741  0.36124
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.40198    0.19327  -7.254 2.81e-06 ***
## log(Mass)                   0.81863    0.04529  18.076 1.36e-11 ***
## Typenon-echolocating bats  -0.08586    0.20521  -0.418    0.682
## Typenon-echolocating birds  0.01406    0.15986   0.088    0.931
## noise                      -0.04162    0.05213  -0.798    0.437
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1881 on 15 degrees of freedom
## Multiple R-squared:  0.9823, Adjusted R-squared:  0.9776
## F-statistic:   208 on 4 and 15 DF,  p-value: 6.077e-13
```

R^2 increased! Did we really *explain* more of the variation in our data? Of course not. In fact, R^2 will never decrease when you add more terms; it can only increase. AIC and BIC, on the other hand, take into consideration the number of terms included in a model.

Check AIC and BIC for `mod1` and `mod4`.

```
AIC(mod1, mod4)
```

```
##      df       AIC
## mod1  5 -4.993569
## mod4  6 -3.825931
```

```
BIC(mod1, mod4)
```

```
##      df         BIC
## mod1  5 -0.01490754
## mod4  6  2.14846307
```

Note that AIC and BIC are both smaller for `mod1` than for `mod4`, so `mod1` would be the preferred model according to both of these criteria. The AIC and BIC do a better (but of course not perfect) job of identifying models that contain useful variables vs. models that contain "noise" (useless) variables.