Outline

1. Re-parametrizations
   - Different formulas for the same model
   - How to get different coefficients in R

2. Interactions
   - Two-way interactions between a factor and another predictor
   - Two-way interactions between two numerical predictors

3. Transformations
   - Linear transformations
   - Log transformations
   - Polynomial regression
Different formulas for the same model

With categorical predictors, there are several choices for the \( m - 1 \) coefficients. General formula:

\[
Y = \mu + \sum_i \alpha_i \mathbb{1}_{X=i} + e
\]

Many combinations of \( \mu \) and \( \alpha_i \)'s represent the same model.

Example: different parametrizations of the population effect, for mussels in the presence of Asian crabs:

\[
\begin{align*}
\text{STI}_{\text{Asian}} &= 0.679 - 0.034 \mathbb{1}_{\text{South}} + e \\
\text{STI}_{\text{Asian}} &= 0.645 + 0.034 \mathbb{1}_{\text{North}} + e \\
\text{STI}_{\text{Asian}} &= 0.662 - 0.017 \mathbb{1}_{\text{South}} + 0.017 \mathbb{1}_{\text{North}} + e
\end{align*}
\]

are all equivalent.
Different formulas for the same model

Southern mussels: different parametrizations of the crab effects

\[
\begin{align*}
\text{STI}_{\text{North}} &= 0.679 - 0.014 \mathbb{1}_{\text{control}} + 0.021 \mathbb{1}_{\text{green}} + e \\
\text{STI}_{\text{North}} &= 0.665 + 0.014 \mathbb{1}_{\text{asian}} + 0.035 \mathbb{1}_{\text{green}} + e \\
\text{STI}_{\text{North}} &= 0.700 - 0.021 \mathbb{1}_{\text{asian}} - 0.035 \mathbb{1}_{\text{control}} + e \\
\text{STI}_{\text{North}} &= 0.6813 + 0.0187 \mathbb{1}_{\text{green}} - 0.0023 \mathbb{1}_{\text{asian}} \\
&\quad - 0.0163 \mathbb{1}_{\text{control}} + e
\end{align*}
\]

are all equivalent. Each one has a constraint: \( \alpha_{\text{asian}} = 0 \) or \( \alpha_{\text{green}} = 0 \) or \( \alpha_{\text{control}} = 0 \) or \( \alpha_{\text{asian}} + \alpha_{\text{green}} + \alpha_{\text{control}} = 0. \)
Different formulas for the same model

Birds and bats example: different parametrization of the species effect

\[
\text{energy} = 6.02 - 3.43 \mathbb{1}_{\text{nBat}} - 4.60 \mathbb{1}_{\text{eBat}} + 0.057 \times \text{mass} + \epsilon \\
\text{energy} = 2.59 + 3.43 \mathbb{1}_{\text{bird}} - 1.17 \mathbb{1}_{\text{eBat}} + 0.057 \times \text{mass} + \epsilon
\]

Bottom-line:

- There are different formulas for the same model.
- Equivalent formulas all use the same number of coefficients, but these coefficients have different meanings.
- Equivalent formulas predict the exact same mean response, when given the same input of predictor values.

What does R use by default?
Default ‘treatment’ parametrization in R

By default: formula corresponding to ‘treatment contrasts’, i.e.
with $\alpha_1 = 0$ and alphabetical ordering of the factor’s levels:

```r
# to read in the data:
dat = read.table("freemanByers.txt", header=T)
# check structure of data:
str(dat)
'data.frame': 72 obs. of 3 variables:
$ pop : Factor w/ 2 levels "North","South": 1 1 1 1 1 1 1 1 1 1 ...
$ crab : Factor w/ 3 levels "asian","control",..: 2 2 2 2 2 2 2 2 2 2 ...
$ shell: num 0.63 0.698 0.696 0.631 0.673 0.67 0.663 0.648 0.66 0.694 ...

# to see the Southern subset of the data:
subset(dat, pop=="North")
# anova for this subset:
lm(shell ~ crab, subset(dat, pop=="North"))
```

```
(Intercept) crabcontrol crabgreen
 0.67892  -0.01383   0.02108
```
How to change the reference level in R

```r
> dat$crab
... Levels: asian control green
> relevel(dat$crab, "control")
... Levels: control asian green

> # to set 'control' as the reference level:
> dat$crab = relevel(dat$crab, "control")
> # check structure of data to see the change:
> str(dat)
...
$ crab : Factor w/ 3 levels "control","asian",..: 1 1 1 ...

> # same anova, but with new parametrization:
> lm(shell ~ crab, subset(dat, pop=="North"))
... (Intercept) crabasian crabgreen
  0.66508  0.01383  0.03492
Outline

1. Re-parametrizations
   - Different formulas for the same model
   - How to get different coefficients in R

2. Interactions
   - Two-way interactions between a factor and another predictor
   - Two-way interactions between two numerical predictors

3. Transformations
   - Linear transformations
   - Log transformations
   - Polynomial regression
Two-way interactions

There is an interaction between two predictors when the **effect of one predictor** on the response **depends** on the value of the other predictor.

**Example** (intuitive): Exposure to Asian crab has no effect on Northern mussels’ shell thickening, while exposure to Asian crabs causes Southern mussels to thicken their shells.

**Warning**: There can be an interaction between two completely independent predictors.

Schematic representation:
Interaction between a factor and a numerical predictor

Prediction formula if predictors $X_1$ is a factor (i.e. categorical) and $X_2$ is numerical:

$$\text{mean of } Y = \mathbb{E}Y = \beta_0 + \sum_i \alpha_i \mathbbm{1}_{X_1=i} + \beta_2 X_2 + \sum_i \gamma_i X_2 \mathbbm{1}_{X_1=i}$$

where:

- $\alpha_1, \ldots, \alpha_k$ are the effect of $X_1$ when $X_2 =$
- $\beta_2$ is the slope for $Y$ on $X_2$ whenever the group $X_1 = i$ is such that $\gamma_i = 0$
- $\gamma_1, \ldots, \gamma_k$ are interaction coefficients: they correct for varying slopes.

Ex: $X_1 =$ type (bird & bats), $X_2 =$ mass.
Interaction between two factors

Prediction formula if predictors $X_1$ and $X_2$ are factors:

$$\mathbb{E} Y = \mu + \sum_i \alpha_i 1_{X_1=i} + \beta_j 1_{X_2=j} + \sum_{ij} \gamma_{ij} 1_{X_1=i, X_2=j}$$

where:

- $\alpha_1, \ldots, \alpha_{k_1}$ are the ‘main’ effects of $X_1$,
- $\beta_1, \ldots, \beta_{k_2}$ are the ‘main’ effects of $X_2$,
- $\gamma_{ij}$, are interaction coefficients to correct for varying effects.
Interaction plots: if no interaction between Crab and Population effects, then lines should be parallel.

dat = read.table("freemanByers.txt", header=T)

# plot of individual observations:
layout(matrix(1:2, 1,2))
plot(shell ~ crab, data=subset(dat, pop=="North"), xlab="North")
plot(shell ~ crab, data=subset(dat, pop=="South"), xlab="South")
# but caution: North and South are plotted on different scales

# plot the group means:
with(dat,
   interaction.plot(crab, pop, shell, type="b", pch=c("N","S"))
)
with(dat,
   interaction.plot(pop, crab, shell, type="b", pch=c("c","a","g"))
)
Fitting and visualizing a model with no interaction

```r
> fit = lm(shell ~ pop + crab, data=dat)
> coef(fit)  # Only main effects: 1 coef (pop) + 2 coefs (crab)
(Intercept) popSouth crabcontrol crabgreen
 0.68858333 -0.05333333 -0.02512500 0.00337500

> newM = data.frame(
>   crab = rep( c("control","green","asian"), 2),
>   pop = rep( c("North","South"), each=3)
> )
> newM

   crab  pop
1 control North
2  green North
3   asian North
4  control South
5   green South
6   asian South
> predict(fit, newM)
 1 2 3 4 5 6
0.6634583 0.6919583 0.6885833 0.6101250 0.6386250 0.6352500

> interaction.plot(newM$crab, newM$pop, predict(fit, newM))
> interaction.plot(newM$pop, newM$crab, predict(fit, newM))
```
Fitting and visualizing a model with interaction

```r
> fit = lm(shell ~ pop * crab, data=dat)
> coef(fit)  # Many more coefficients
     (Intercept) popSouth  crabcontrol crabgreen popSouth:crabcontrol popSouth:crabgreen
 0.6789167   -0.03400000  -0.01383333  0.02108333  -0.02258333  -0.03541667

> newM
  crab  pop
1 control North
2    green North
3     asian North
4   control South
5      green South
6      asian South
> predict(fit, newM)
     1     2     3     4     5     6
0.6650833 0.7000000 0.6789167 0.6085000 0.6305833 0.6449167

> interaction.plot(newM$crab, newM$pop, predict(fit, newM))
> interaction.plot(newM$pop, newM$crab, predict(fit, newM))
```

How many coefficients total? How many groups?
Interaction between two numerical predictors

Prediction formula if predictors $X_1$ and $X_2$ are numerical:

$$\text{mean of } Y = \mathbb{E} Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

where:

- $\beta_1$ is the slope for $Y$ on $X_1$ when $X_2 = 0$
- $\beta_2$ is the slope for $Y$ on $X_2$ when $X_1 = 0$
- $\beta_3$ is the interaction coefficient to correct for varying slopes.

Examples in homework assignments.
Re-parametrizations
- Different formulas for the same model
- How to get different coefficients in R

Interactions
- Two-way interactions between a factor and another predictor
- Two-way interactions between two numerical predictors

Transformations
- Linear transformations
- Log transformations
- Polynomial regression
Linear transformations

- do not change the quality of a fit ($R^2$, p-value for slope, residual plot).
- can make parameters easier to interpret.

Example: explain energy by mass, in birds only.

```r
> bats = read.table("bats.txt", header = T)
> tapply(bats$mass, bats$type, mean)
bird eBat nBat
   263  29  495
> birds = subset(bats, type=="bird")

> birds0.lm = lm(energy ~ mass, data=birds)
> birds1.lm = lm(energy ~ I(mass-260), data=birds)
```
Compare output

- $R^2$ and residual SD are the same for both fits.
- intercepts are different.

```r
> summary(birds0.lm)

Call: lm(formula = energy ~ mass, data = birds)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.3167     3.2988   1.01   0.340
mass          0.0678     0.0107   6.31  8.8e-05 ***

Residual standard error: 5.89 on 10 degrees of freedom
Multiple R-squared: 0.799, Adjusted R-squared: 0.779
F-statistic: 39.8 on 1 and 10 DF, p-value: 8.82e-05

> summary(birds1.lm)

Call: lm(formula = energy ~ I(mass - 260), data = birds)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   20.9381     1.6995  12.32   2.3e-07 ***
I(mass - 260)  0.0678     0.0107   6.31  8.8e-05 ***

Residual standard error: 5.89 on 10 degrees of freedom
Multiple R-squared: 0.799, Adjusted R-squared: 0.779
F-statistic: 39.8 on 1 and 10 DF, p-value: 8.82e-05
```
Predictions

- Use the `predict()` function to make predictions.
- The predicted values are the same.

```r
> predict(birds0.lm)
> predict(birds1.lm)

   5     6     7     8     9    10    11    12    13    14
4.96  5.69  8.25 11.45 17.75 21.95 28.39 29.34 33.27 31.24

   15    16
25.68 35.85

> predict(birds0.lm, list(mass=c(200, 260, 320)))
> predict(birds1.lm, list(mass=c(200, 260, 320)))

     1     2     3
16.872 20.938 25.005```
Models with interactions

Transformations by **centering** are especially helpful for interpreting interaction coefficients.

```r
> bats1.lm = lm(energy ~ I(mass-260) * type, data=bats)
> library(arm)
> display(bats1.lm, digits=3)

lm(formula = energy ~ I(mass - 260) * type, data = bats)

<table>
<thead>
<tr>
<th></th>
<th>coef.est</th>
<th>coef.se</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>20.938</td>
<td>1.455</td>
</tr>
<tr>
<td>I(mass - 260)</td>
<td>0.068</td>
<td>0.009</td>
</tr>
<tr>
<td>typeBat</td>
<td>2.861</td>
<td>15.996</td>
</tr>
<tr>
<td>typenBat</td>
<td>0.701</td>
<td>3.997</td>
</tr>
<tr>
<td>I(mass - 260):typeeBat</td>
<td>0.022</td>
<td>0.069</td>
</tr>
<tr>
<td>I(mass - 260):typenBat</td>
<td>-0.028</td>
<td>0.015</td>
</tr>
</tbody>
</table>

---

n = 20, k = 6
residual sd = 5.041, R-Squared = 0.90
Short quiz

> coef(bats1.lm)
  (Intercept) I(mass - 260) typeeBat typenBat
  20.938149  0.067775  2.861360  0.701114
  I(mass - 260):typeeBat I(mass - 260):typenBat
  0.021862  -0.027729

1. Interpret each intercept coefficient.
2. Interpret each slope coefficient.
3. Predict the energy for a 20 g echo-locating bat.
4. Predict the energy for a 220 g bird.
5. Predict the energy for a 420 g non-echo-locating bat.

> predict(bats1.lm, list(mass=c(20, 220, 420),
+                             type=factor(c("eBat","bird","nBat"))))

  2.2867 18.2272 28.0466
Log transformations

- **do change** the goodness of fit ($R^2$, p-values, residual plots).
  can make a linear model fit better –or worse.
- require variables with **positive** numerical values;
- especially useful when large values are more variable than small values

How to do it:

- The function \( \log() \) takes the **natural** logarithm.
- Use the \( \exp() \) function to go back to the original units.
- Since \( \exp(x) \approx 1 + x \) for \( x \) near 0, interpretations of coefficients after a log transformation of the outcome variable are easier to interpret than if using \( \log_{10} \).
Log transformations: interpretation

- if $X$ is categorical and $\log(Y) = \beta_0 + \beta_1 \mathbb{1}_{eBat} + e$, a change of $X$ from the reference level to $eBat$ results in a change of $Y$ by a factor $(1 + \beta_1)$ approximately, or in an increase of $Y$ by $\beta_1 \times 100\%$.

- if $X$ is numerical and $\log(Y) = \beta_0 + \beta_1 X + e$, an increase of $X$ by a small amount $d$ results in a change of $Y$ by a factor $(1 + \beta_1 d)$, or in an increase of $Y$ by $\beta_1 d \times 100\%$.

- if $X$ is numerical and $\log(Y) = \beta_0 + \beta_1 \log(X) + e$, $\log(x) + d \approx \log(x \times (1 + d))$ and an increase of $X$ by a small amount $d \times 100\%$ results in an increase of $Y$ by about $\beta_1 d \times 100\%$. 
Birds and bats example

Plot the data on the original and log scale:

```r
> plot(log(energy) ~ log(mass), data=bats, col=type)
> plot( energy ~ mass , data=bats, col=type)
```

Linear regression on the log scale, mass centered:

```r
> with(bats, tapply(log(mass), type, mean) )
   bird  eBat  nBat
      5.2290 2.6388 6.1015
> bats2.lm = lm(log(energy) ~ I(log(mass)-5.2)+type, data=bats)
```

Check the residual plot to decide which regression is best:

```r
> plot(resid(bats2.lm) ~ fitted(bats2.lm))
> abline(h=0)
```
# is much better than before transformation:
```r
> plot(resid(bats1.lm) ~ fitted(bats1.lm))
> abline(h=0)
```
Birds and bats example

```r
> summary(bats2.lm)

lm(formula = log(energy) ~ I(log(mass) - 5.2) + type, data = bats)

Coefficients:

                  Estimate Std. Error t value  Pr(>|t|)
(Intercept)        2.7637     0.0537   51.47  < 2e-16 ***
I(log(mass) - 5.2)  0.8150     0.0445   18.30 3.8e-12 ***
type<eBat         -0.0236     0.1576   -0.15     0.88
type<enBat        -0.1023     0.1142   -0.90     0.38

Residual standard error: 0.186 on 16 degrees of freedom
Multiple R-squared: 0.982,     Adjusted R-squared: 0.978
```

- In a comparison of an echo-locating bat to a bird of equal mass, we expect the energy use to be about 2% less.
- In a comparison of a non-echo-locating bat to a bird of equal mass, we expect the energy use to be \(\sim 10\%\) less.
- An increase of mass by 1% is expected to result in an increase of energy by about 0.8%.
- Are these numbers significantly different from 0?
Predictions

```r
> newdata = data.frame(mass=c(20,20,400,400),
+                        type=c("bird","eBat","bird","nBat"))
> predict(bats2.lm, newdata)         # this is on log scale
> exp( predict(bats2.lm, newdata) )  # original scale: Watts

Plot data and predictions on the log-log scale:

mass.x = seq(1,801,by=10)
newbird = data.frame( mass=mass.x, type="bird")
neweBat = data.frame( mass=mass.x, type="eBat")
newnBat = data.frame( mass=mass.x, type="nBat")

plot(log(energy) ~ log(mass) , data=bats, col=type, pch=16)
lines(log(mass.x), predict(bats2.lm, newbird) )
lines(log(mass.x), predict(bats2.lm, neweBat), col=2 )
lines(log(mass.x), predict(bats2.lm, newnBat), col=3 )

Plot data and predictions on the original scale (Watts/grams):

plot(energy ~ mass , data=bats, col=type, pch=16)
lines(mass.x, exp(predict(bats2.lm, newbird)) )
lines(mass.x, exp(predict(bats2.lm, neweBat)), col=2 )
lines(mass.x, exp(predict(bats2.lm, newnBat)), col=3 )
```
Polynomial regression

- Model: \( y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k + e_i \) and 
  \( e_i \sim \mathcal{N}(0, \sigma^2) \) are supposed independent.

- Caveat: the predictors \( x_i, x_i^2, \ldots \) are usually highly correlated, so the inference of slopes is unstable (large uncertainty) but it is not a problem for prediction within the range of the data (more later).

- Hierarchical structure: if we include \( x^3 \) for instance, then we also include \( x^2 \).

  Useful if other transformations do not work, if there is a curved pattern with homogeneous variance.
Example: quadratic and cubic regressions

```r
> bats3.lm = lm(energy~mass + I(mass^2), data=bats)
> bats4.lm = lm(energy~mass + I(mass^2) + I(mass^3), data=bats)
> anova(bats3.lm, bats4.lm)
Analysis of Variance Table
Model 1: energy ~ mass + I(mass^2)
Model 2: energy ~ mass + I(mass^2) + I(mass^3)
  Res.Df RSS Df Sum of Sq    F  Pr(>F)
1   17 347
2   16 332  1     14  0.69 0.42

> summary(bats4.lm)
... Coefficients:
            Estimate Std. Error    t value  Pr(>|t|)
(Intercept) -0.0105    0.227  -0.050 0.9636
mass        0.1240     0.036  3.423  0.0035 **
I(mass^2)   -0.00017   0.001  -1.283 0.2205
I(mass^3)   9.27e-08   1.11e-07 0.832 0.4171

> summary(bats3.lm)
... Coefficients:
            Estimate Std. Error   t value  Pr(>|t|)
(Intercept)   0.8774    1.92  0.464 0.6540
mass          0.0960    0.013  7.208 1.5e-06 ***
I(mass^2)    -0.00057   0.001  -2.994 0.0082 **
...