Stochastic simulations

Today's agenda:

- Discuss simulations
- Do some simple simulations

What are simulations?

Experimentation with the help of a model

- We choose (or build) a model, select parameters, and see what happens

Uses of simulations

- Theory development (what happens when I add a new feature?)
- Model design (does this model capture the intended dynamics?)
- Study design (how many replicates do I need?)
- Methods development (does this metric tell us what we want?)

How to simulate?

1) Select a model
2) Choose one or more sets of parameters
3) Simulate using the model and parameters
4) Examine the output
5) Revise 1 and repeat as needed (optionally)

Simple linear function with normal error

- Let’s pick a simple line as our model (y = a + bx)
- For our parameters:
  - A = 2
  - B = 1
  - X = 1:20

```
a <- 2
b <- 1
x <- 1:20
y_det <- a + b * x
plot(x = x, y = y_det)
```

Simple linear function with normal error

- Let’s make it a bit more realistic by adding some uncertainty

```
a <- 2
b <- 1
x <- 1:20
y_det <- a + b * x
y_stoch <- rnorm(n = 20, mean = y_det, sd = 2)

plot(x=x, y=y_stoch)
```

Simple linear function with normal error

- Try re-running these lines a few times and see what happens

```
y_stoch <- rnorm(n = 20, mean = y_det, sd = 2)
plot(x=x, y=y_stoch)
```

Simple linear function with normal error

Each time you re-run `rnorm()` you get a different set of points

- A random draw from a normal distribution
- Each draw represents a different possible outcome for the same model/parms

Simple linear function with correlated error

What if the variation increases with mean?

- E.g., variation in body size might increase with mean body size

```
a <- 2
b <- 1
x <- 1:20
sd_multiplier <- 0.3

y_det <- a + b * x
y_stoch <- rnorm(n = 20, mean = y_det, sd = y_det*sd_multiplier)

plot(x=x, y=y_stoch)
```

Simple linear function with correlated error

Do a few more replicates, comparing them with the deterministic bit

```
y_det <- a + b * x
y_stoch <- rnorm(n = 20, mean = y_det, sd = y_det*sd_multiplier)

plot(x = x, y = y_stoch)
points(x, y_det, col = "blue")
```

Hyperbolic function with negative binomial error

Model: y = a*b/(b+x)

Error: negative binomial

Hyperbolic function with negative binomial error

Model: y = a*b/(b+x)

Error: negative binomial

```
a <- 20
b <- 1
k <- 5
x <- runif(50, min=0, max=5) #How does this differ from x <- 0:5?

y_det <- a/(b+x)

plot(x = x, y = y_det)
```

What is the shape of the plotted line?

Why doesn't a normal error term make sense?

Hyperbolic function with negative binomial error

Adding the negative binomial error:

```
y <- rnbinom(n = 50,mu = y_det, size = k)
plot(x = x, y = y)
```

Run these a few times to get a sense of the variation

What does the mu parameter control?

Categorical differences

Model: y = a*b/(b+x)

Adding multiple categories:

- Assume species differ in *a* and *b* parameters
- Assume they have the same error size parameter

Categorical differences

Model: y = a*b/(b+x)

Adding multiple categories:

- Assume species differ in *a* and *b* parameters
- Assume they have the same error size parameter

We need different parameters and a way of accessing them

Categorical differences

Create a grouping variable:

```
g <- factor(rep(1:2, each = 25))
```

This will be used to divide up the 50 observations into 2 groups

Categorical differences

Specify parameters as vectors

```
a <- c(20,10)
b <- c(1,2)
```

Define y_det with the different variables:

```
y_det <- a[g]/(b[g]+x)
```

Which values will apply to group 2? Why?

(Hint: try a[1] or a[2] to see what happens)

Categorical differences

Plot the two sets of variables:

```
plot(x = x, y = y_det)
```

Categorical differences

Now add in the variation and plot again:

```
y <- rnbinom(n = 50, size = k,mu = y_det)
plot(x = x, y = y)
```

Does that look right?

Categorical differences

```
plot(x = x, y = y)
```

Difficult to tell the clusters apart

Need to improve the plot

- `pch` controls plot symbols
- `col` controls plot colors
- Both are designed to work with numbers

Categorical differences

We can have R give each group a different color

```
plot(x = x, y = y, col = g)
```

Try this out!

`pch` works in the same way, give it a shot!

Also try having both `pch` and `col` correspond to the group

Using simulations

Once we have the simulated data we can, e.g. :

- Compare to expectations/existing data
- Compare to data with other parameter values
- Check whether groups can be differentiated
- Check whether downstream analyses do what we expect

Using simulations

For our categorical trait simulation:

- Try changing a and b for one or both groups
- Can you find parameter values where the two groups seem very different?
- Or where they seem identical?
- Are there some parameter values where they overlap in some regions but not in others?

Adding more

Simulations and models can range from very simple to very complex

- Simpler models are easier to validate and debug
- More complex models add more realism
- Start simple and add parameters as needed

See 5.2.2.2 in the book for a more complex example

Before next class:

- Read 5.3 (especially important to understand if you plan experiments!)