Getting started

Exploring the data

A simple statistical test

Multiple linear regression

More Practice

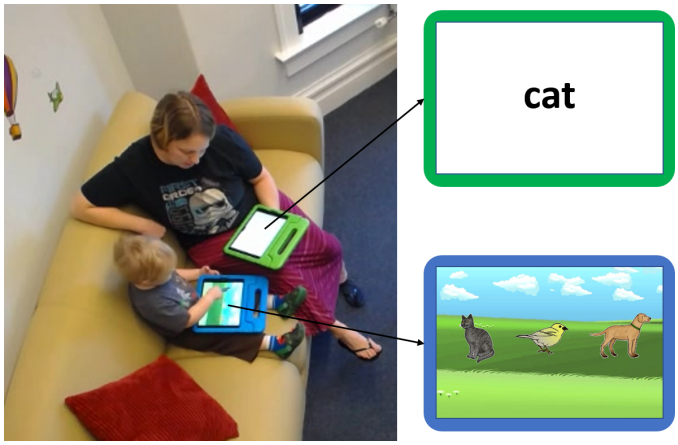
Choosing a model

Your reproducible lab report: Before you get started, download the R Markdown template for this lab. Remember all of your code and answers go in this document:

```
download.file("https://dyurovsky.github.io/85309/post/rmd/lab11.Rmd",
              destfile = "lab11.Rmd")
```

One of the most striking developments in children's first few years of life is just how quickly they learn language. This is despite the fact that children's abiltiies to control their attention and remmeber information are still developing. One potential explanation for why children learn so quickly despite their immature cognitive systems is that the language they hear is different from the language that adults hear. If you think about how we talk to children, you'll probably notice a number of ways in which we simplify our speech. One hypothesis about child-directed directed speech is that it is not just simpler that adult-directed speech across the board, but that it might be specially calibarted to children's developing linguistic and cognitive abilities: Caregivers might tune the information they provide to keep it at the right level of complexity.

In this lab we'll be looking at data from a recent paper from my lab (https://callab.github.io/publication/leung-2021-parents/leung-2021-parents.pdf) in which we asked from parents and their children playing a simple reference game. On each round of the game, children saw three animals on the screen of an iPad, and parents goal was to use language to communicate to their child which target animal they should choose (see below). The primary measure of interest in this study was how many words parents used to communicate to their child. In line with the tuning hypothesis, we predicted that parents should use more words to talk to their children about animals that they are less likey to know.



Getting started

As usual, we're going to load the `tidyverse` package for data manipulation. We'll also be reading in a dataset to work with just like we usually do. We'll also load in the data from the paper.

```
library(tidyverse)
data <- read_csv("https://dyurovsky.github.io/85309/data/lab11/animal_game.csv")
```

The data

The data consists of the lengths of parents utterances for each trial of the game for 41 parent-child dyads, as well as some relevant predictor measures.

The meaning of each variable can be found in the codebook below:

Codebook

- `subj_group` : A variable just for this lab, I split the dataset into two halves (training and testing)
- `subj` : A unique identifier for each dyad
- `trial` : Which round of the game it is.
- `trial_target` : The animal that was the target
- `appearance` : Each animal was the target twice. So each time it is it's first or second apperance
- `avg_known` : The proportion of children in the sample whose parents reported that they knew each animal
- `known` : Whether this parent reported that their child knew the target animal
- `length` : The number of words the parent said on this round of the game

---

Exploring the data

Let's start by visualizing the dependent measure to see what what kinds of statistical tools might be appropriate for thinking about it.

Exercise 1 Make a subset of the data called `train_data` that contains only the subjects whose `subj_group` is `train`. We're going to use to use a comparison between these and the `test` subjects later to think about generalization. You'll use this subset for Exercises 2-8.

Exercise 2 Plot and describe the distribution of `length` for the `train_data`. Is the distribution skewed? Is that what you would have expected? Why might length look like that?

Exercise 3 Plot and describe the relationship between `length` and the categorical variable `appearance`. Does it like there is a relationship between them?

A simple statistical test

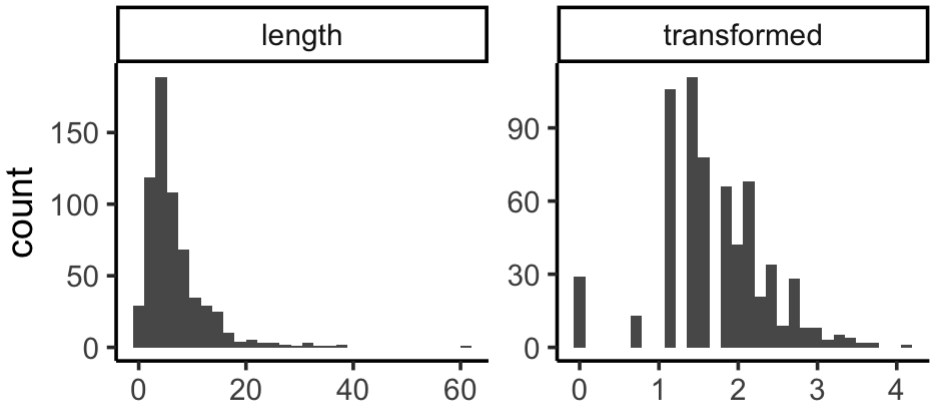
You've learned a number of methods for Null Hypothesis tests over the course of the semester. Let's remind ourselves what they are, and also look at the similarities and differences among them.

Exercise 4 Use simulations (like in Lab 4 (https://dyurovsky.github.io/85309/post/labs/hypothesis_testing.html)) to determine if there is a relationship between `length` and `appearance`. Write out the Null and Alternative Hypotheses you are considering and say whether you rejected the Null. Hint: Thank carefully about how you want to group the data.

Exercise 5 Use a t-test to answer the same question. Hint: Think carefully about what kind of t-test is appropriate here.

Exercise 6 Finally, use a simple linear regression to answer the same question. Does the estimate of the slope correspond to the estimate you got from your t-test. Does the p-value correspond? Why or why not?

You hopefully noticed in Exercise 1 that the distribution of lengths is definitely not Normal. Let's try transforming it to see how that impacts our statistical test. You want to get from the distribution on the left to the distribution on the right.



Exercise 7 Now repeat one of these methods, but transform the length variable using the appropriate transformation to make the disribution more normal. What changed about your model's output? What do you think is causing the change?

Multiple linear regression

Now that we're warmed up, let's try predict `length` of parents' referring expressions from all of the measures we have available using multiple regression except for `subj` and `trial_target`. Before you do this, make sure you have transformed the length variable appropriately. The easiest thing to do is to make a new `tibble` called `transformed_train` which has all of the same data as the `train` tibble but also a new column called `transformed_length` with the appropriate transformation.

Exercise 8 Fit a multiple regression model predicting transformed length from all of the potentially relevant variables in the dataset and store it in a variable called `full_model`. Interpret the output of this model to tell me what predicts the length of parents' referring expressions (Hint: use the `summary` function).

Exercise 9 Use stepwise regression (the `step` function) to do model selection to find the model that is the "best" model using the AIC method. Store this model in a variable called `step_model`. Which variables are still in the model? Did anything about your interpretation change?

More Practice

Prediction and Generalization

Let's see if our model selection was any good. One way to think about what you are trying to do when you leave variables out of your model is that you are trying to keep just the variables that will generalize to other samples from the same population. These variables should be useful for predicting what new data will look like, while other variables are not.

In the next question, we're going to use the `predict` function which takes a model as input and spits out it's predictions for a dataset. Remember, a model is just an equation for estimating the dependent variable from the values of independent variables.

First, let's get predictions from the model for our training data that we used to estimate it. We can do that like this:

```
predicted_train <- transformed_train %>%
  mutate(predicted_full = predict(full_model),
         predicted_step = predict(step_model))
```

Exercise 10 Use the `cor` function to compute the correlations between the predictions of both the full and stepwise models and the transformed length in the training data. Do these values map on as you would expect to the output of the models?

Now let's try generalizing to new data.

Exercise 11 Make a new tibble called `transformed_test` that contains only the subjects in the `test` `subj_group` from the original dataset that has a column called `transformed_length` applying the same transformation to length as you did in the train data.

We'll use the predict function again, but this time with a new argument `newdata` which will tell it to make predictions using values for the predictor variables found in a new tibble.

```
predicted_test <- transformed_test %>%
  mutate(predicted_full = predict(full_model, newdata = .),
         predicted_step = predict(step_model, newdata = .))
```

Exercise 12 Use the `cor` function again to compute the correlation between model predictions for both the full and stepwise model and this new dataset. Which model predicted the new data better?

Exercise 13 Finally, apply stepwise regression to the full dataset (`data`). Did you end up with a different model than when you were looking at just the training dataset? If so, why do you think that is?