Week 2

Santiago Barreda

Contents

1 Comparing two groups

In the previous chapter, I focused mostly on outlining the logic of estimating parameter values and credible intervals for these parameters. I focused on investigating values from a single group, which is the simplest possible data you can deal with. In this chapter I am going to talk a bit more about `brms` and Bayesian multilevel models, with the goal of comparing data from two groups to see how similar/different they really are. We are going to be using the same vowel data, but this time we are going to focus on f0 measurements for adult females and girls between 10-12 years of age.

1.1 Data and research questions

Below I load in the Hillenbrand et al. data from last time, and add a new variable that indicates whether the talker is an adult or a child. I also split the data up by gender. The variable `uspeaker` is a speaker number that is unique across all speaker groups, and `type` indicates speaker group from among `b` (boys), `g` (girls), `m` (men), and `w` (women).

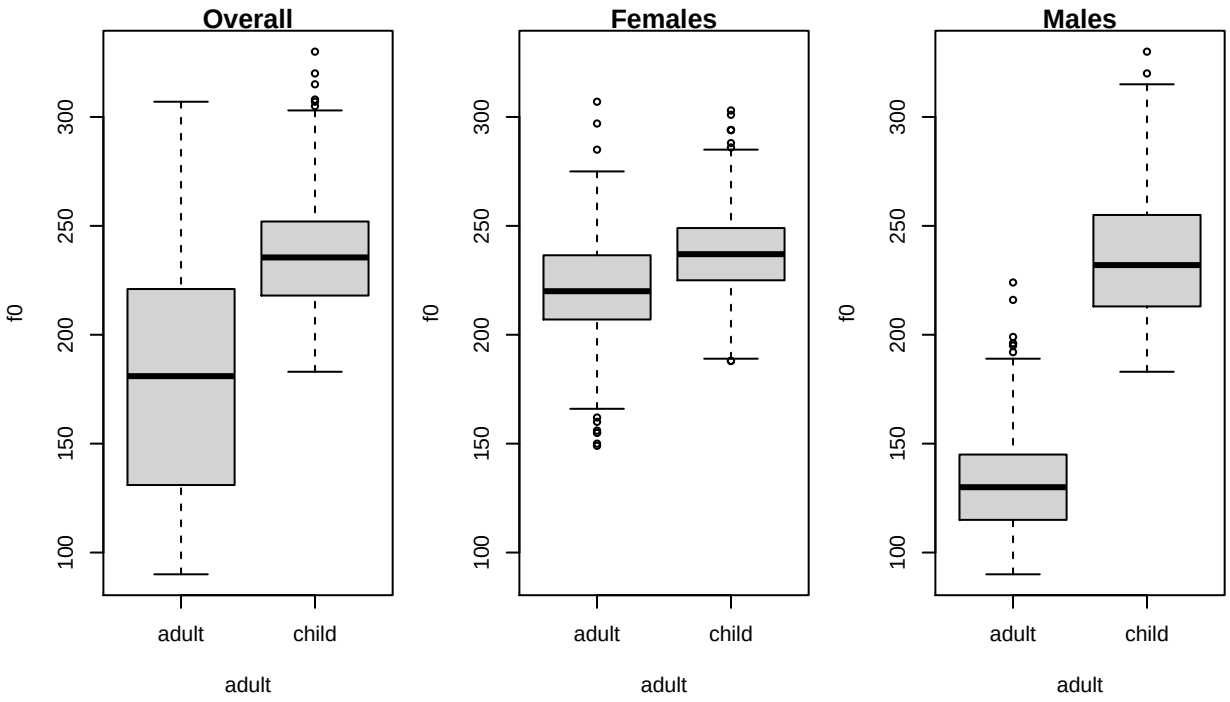
```r
url1 = "https://raw.githubusercontent.com/santiagobarreda"
url2 = "/stats-class/master/data/h95_vowel_data.csv"
h95 = read.csv (url(paste0 (url1, url2)))
## make variable that indicates if the talker is an adult
h95$adult = ""
h95$adult[h95$type %in% c('w','m')] = "adult"
h95$adult[h95$type %in% c('g','b')] = "child"

## split up data by into male and female groups
males = h95[h95$type %in% c('m','b'),]
females = h95[h95$type %in% c('w','g'),]
# re-factor to remove excluded subjects
males$uspeaker = factor (males$uspeaker)
# re-factor to remove excluded subjects
females$uspeaker = factor (females$uspeaker)
```

One of the simplest questions a researcher can ask (from a modeling perspective) is: Are two groups of observations different or are they the same? For example in phonetics researcher ask questions like, have these vowels merged in this dialect (are these two things different)? Does visual information speed up speech perception (are two sets of reaction times the same)?

A good way to isolate a single difference is to create groups that differ primarily according to the characteristic you are interested in testing. For the reaction-time example above, I might present listeners with very similar words in the same conditions, but half with and half without visual information. The difference in reaction times between the groups should reflect the advantage given by visual information. The question would then be, are reaction times with visual information *the same thing* as reaction times without visual information?

In our data we have female talkers of the same dialect, who differ primarily in terms of age. Basically, these groups differ mostly in terms of adultness, and so we can use the characteristics of these groups to investigate the effect of 'adultness' on average f0. Below I present boxplots of the effect of adultness on f0, first overall and then divided by speaker gender.

```r
par (mfrow = c(1,3))
boxplot (f0 ~ adult, data = h95, main = "Overall", ylim = c(90,330))
boxplot (f0 ~ adult, data = females, main = "Females", ylim = c(90,330))
boxplot (f0 ~ adult, data = males, main = "Males", ylim = c(90,330))
```

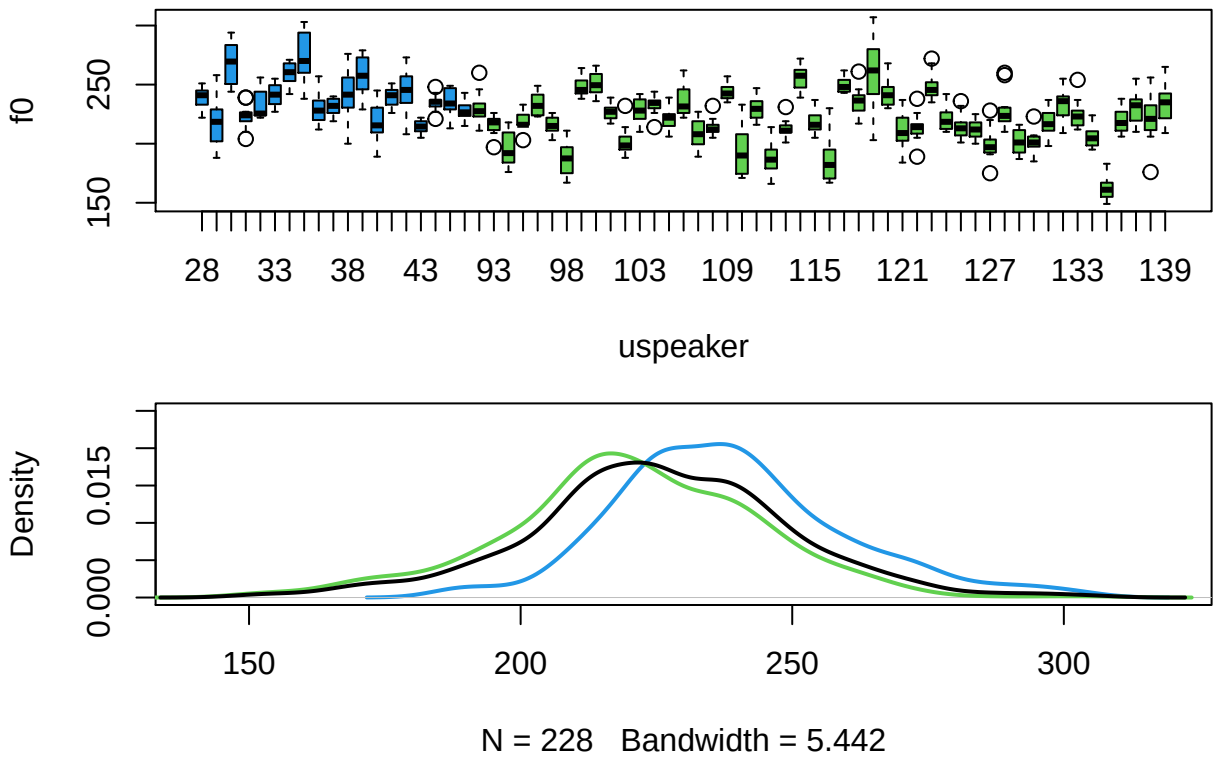


Clearly there is a difference in f0 between children and adults, but the difference also seems to be gender-dependent: there is more of a difference between men and boys than between women and girls.

We are going to focus on the female data for now. Below, the figure on the left highlights both between- and within-speaker variation in f0 by women (blue) and girls (green). On the right, we see the overall distribution (black) compared to the distribution of f0 for women (blue) and girls (green).

Obviously the distributions seem a bit different, but they are also not *that* different. We can also clearly see that there is substantial overlap between individual girls and women (in the figure on the left), so that even if the groups are 'different' in some sense, that does not mean that they are entirely different. We are going to fit a model that can help us quantify all of the variability shown in the figures below.

```
colors = c(3,4)[ apply (table(females$uspeaker, females$adult),1,which.max) ]

par (mfrow = c(2,1))
boxplot (f0 ~ uspeaker, data=females, col = colors)

plot (density (females$f0[females$adult=="child"]), lwd = 2, col = 4, main = "",
      xlim = c(140,320), ylim = c(0,0.025))
lines (density (females$f0[females$adult=="adult"]), lwd = 2, col = 3)
lines (density (females$f0), lwd = 2, col = 1)
```



1.2 The model

In chapter 1 the model we fit had the simplest possible formula, just an intercept. Here, we need to extend this to include an actual predictor: a vector indicating 'adultness'. Remember that formulas look like this `variable ~ predictor`. So, if we want to predict f0 based on whether the talker is an adult or not, our model is going to have a formula that looks basically like this `f0 ~ adult` (we can omit the `1` when we include predictors in the formula). You can think of this meaning something like 'We expect the distribution to vary based on whether the talker is an adult or not'. We can fit this model for the female speakers below.

```
library (brms)
```

```
set.seed (1)
female_model =
  brm (f0 ~ adult + (1|uspeaker), data = females, chains = 1, cores = 1,
       iter = 4000, refresh = 0)
```

```
## load pre-fit model
female_model = readRDS ("2_female_model.RDS")
```

Note that the model also calculates an intercept (mean value) for each speaker, but does **not** include `adult` inside the `(1|uspeaker)` section. This is because we are estimating a mean for each speaker, but not the effect for 'adultness' for each speaker. Each speaker is only either an adult or a child, and so we cannot

estimate the effect for 'adultness' for each speaker. Doing things like that that don't 'make sense' from the model's perspective will cause it to crash or return strange values.

We can inspect the model below:

```
female_model ## inspect the model
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: f0 ~ adult + (1 | uspeaker)
##    Data: females (Number of observations: 804)
## Samples: 1 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 2000
##
## Group-Level Effects:
## ~uspeaker (Number of levels: 67)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    19.13      1.77    16.01    22.88 1.00      177      289
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    220.26      2.75   215.12   225.97 1.00       69      108
## adultchild    18.10      5.21     8.15    28.55 1.00       94      254
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.95      0.33    12.30    13.60 1.00     1772     1478
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

The output is mostly familiar, but there is a new predictor in the model. In addition to the 'Intercept' term, we now get estimates for a term called 'adultchild'. Admittedly, this is weird, but its how R handles predictors that are words (called 'factors' in R). What this tells us is that this is the estimate for the 'child' level of the 'adult' factor.

A couple of questions arise. First, the 'Intercept' term in the model above seems to correspond to the mean f0 for adult females. We can check this:

```
## calculate means of f0 based on values of adult vector
tapply (females$f0, females$adult, mean)
```

```
##    adult    child
## 220.4010 238.3509
```

And see that the intercept the is adult female mean. Except the estimate for children is 17 Hz? This is obviously not the mean f0 for the girls in our sample. Why not? To understand model coefficients we are going to have to talk about contrasts

1.2.1 Contrasts

Contrasts are the numerical implementation of factors (variables like adult vs. child that are not numerical) in your model. You may initially think that we can separately estimate the women's average, the girl's average, and the the overall mean. However, our models can't actually do this. The general problem is this: if you have two groups you can't independently calculate:

1) group 1 mean.
2) group 2 mean.
3) the overall mean.

Why not? Because once you know 2 of those things you know the 3rd. For example, if the group 1 mean is 5 and the overall mean is 6, obviously the group 2 mean **must** be 7. Why does this matter? Because when things are entirely predictable based on each other, they are not actually separate things, even though they may seem that way to us. When things are entirely predictable in this way we say they are linearly dependent, and regression models don't like this.

For example, imagine you were trying to predict a person's weight from their height. You want to include height in centimeters *and* height in meters in your model, and you want to independently estimate effects for both predictors. Since height in centimeters = height in meters / 100, that is obviously not going to be possible. The effect of one must be 100 times the effect of the other! Even though it may be less transparent, this is the same reasons why we can't estimate all the group means *and* the overall mean.

Another way to think of it is that with two groups you can estimate one difference, not two. To estimate both group means and the overall mean, you would need to estimate *two* distances if each group could really be a different distance from the mean. Instead, we are really only in a position to estimate *one* difference.

Here's another way to think about why we can't get the overall mean. When we had one group we obviously could only estimate one mean. We couldn't get the overall mean independently from the sample mean. Adding 1 more group allows us to calculate 1 more mean. Why would it let us calculate 2 more means? That would mean adding a second group (with 1 mean) somehow contributed twice as much information as the first group did.

1.2.2 Treatment coding

The coding scheme determines how your model represents the differences it encodes. In the model above we used 'treatment' coding (the default in R). In treatment coding, a 'reference' level is chosen to be the intercept, and all group effects reflect the difference between the group mean and the mean for the reference level.

By default, R chooses the alphabetically-lowest level to be the reference level. That is why the Intercept in our model is equal to the mean of the 'adult' group, the average for adult females. The effect for child ('adultchild') represents the difference between the child mean and the adult mean. This means that our credible intervals also represent the *difference* in the means and not the means themselves. So, we expect the *difference* between girls and women in this sample to be about 17 Hz, and we think there is a 95% chance that the *difference* between the means is between 7.8 and 27.7 Hz in magnitude.

We can see how the effects estimates in our model resemble the means, or differences between means, in our sample.

```
# calculate group means
tapply (females$f0, females$adult, mean)
```

```
##    adult    child
## 220.4010 238.3509
```

```
# find the difference between them
diff (tapply (females$f0, females$adult, mean))
```

```
##    child
## 17.94984
```

So, to interpret treatment coded coefficients remember:

- The reference category mean is the 'Intercept' in the model.
- The value of the coefficients of any other group mean will be equal to `group mean - reference group mean`.
- To recover the mean estimate for any other group, we add `group mean + reference group mean`.

1.3 Interpreting the model

I'm going to explain how to get all kinds of information out of this model, and how this can be used to answer research questions.

1.3.1 Interpreting the model print statement

Just typing the model name into the console and hitting enter prints the information seen above. The first part:

```
Family: gaussian
  Links: mu = identity; sigma = identity
Formula: f0 ~ adult + (1 | uspeaker)
   Data: females (Number of observations: 804)
Samples: 1 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup samples = 2000
```

Just tells you technical details that we don't have to worry about for now (though some are obvious). The next part tells about random variation at the second-level (between-speaker variation). Right now it only includes means but soon we will fit models with more terms here:

```
Group-Level Effects:
~uspeaker (Number of levels: 67)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)    19.01      1.77    15.90    22.61 1.00      185      208
```

Above we see that the standard deviation of speaker-specific means is 19 Hz. This means that if you randomly select a girl or woman from our sample, we expect them to be about 19 Hz away from their group mean (not the overall mean), on average. We also get information about credible intervals for this estimate, and information about ESS for the parameter. This approach assumes that women and girls are distributed in the same way around their group means, since it estimates only a single standard deviation term.

Next we see the effects estimated for out predictors at the lowest level, the data level. These are 'population' level effects because they are not specific to any given speaker in our sample.

```
Population-Level Effects:
           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    220.65      2.89   215.01   226.63 1.00       94      191
adultchild    18.11      5.35     7.68    29.27 1.02       96      144
```

For models that deal with normally-distributed data (like ours), the last section presents information about random variation at the lowest level. In this case, this is the within-speaker variation, which we are treating as 'error'. Of course, there are many systematic reasons why f0 varies within-speaker. However, our model does not contain any predictors that explain them. As a result, from the perspective of our model, only between-group and between-speaker vatiation can be explained, and *all* within-speaker variation is considered to be error.

The sigma term tells you: if you take a random observation from a random person, how far do you expect the token f0 to be from the person's mean f0, on average.

```
Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma    12.96      0.33    12.31    13.60 1.00     1236     1414
```

This last section is just boilerplate and contains some basic reminders. This text will look the same after all models.

```
Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

1.3.2 Reporting values and differences

The `brms` package has several functions that make getting information from these models simple. The `fixef` function gets you means and 95% credible intervals for all your 'population level' parameters (sometimes called 'fixed' effects).

```
fixef (female_model)
```

```
##             Estimate Est.Error       Q2.5     Q97.5
## Intercept  220.26363  2.746034 215.115266 225.97120
## adultchild  18.09525  5.213419   8.149107  28.54532
```

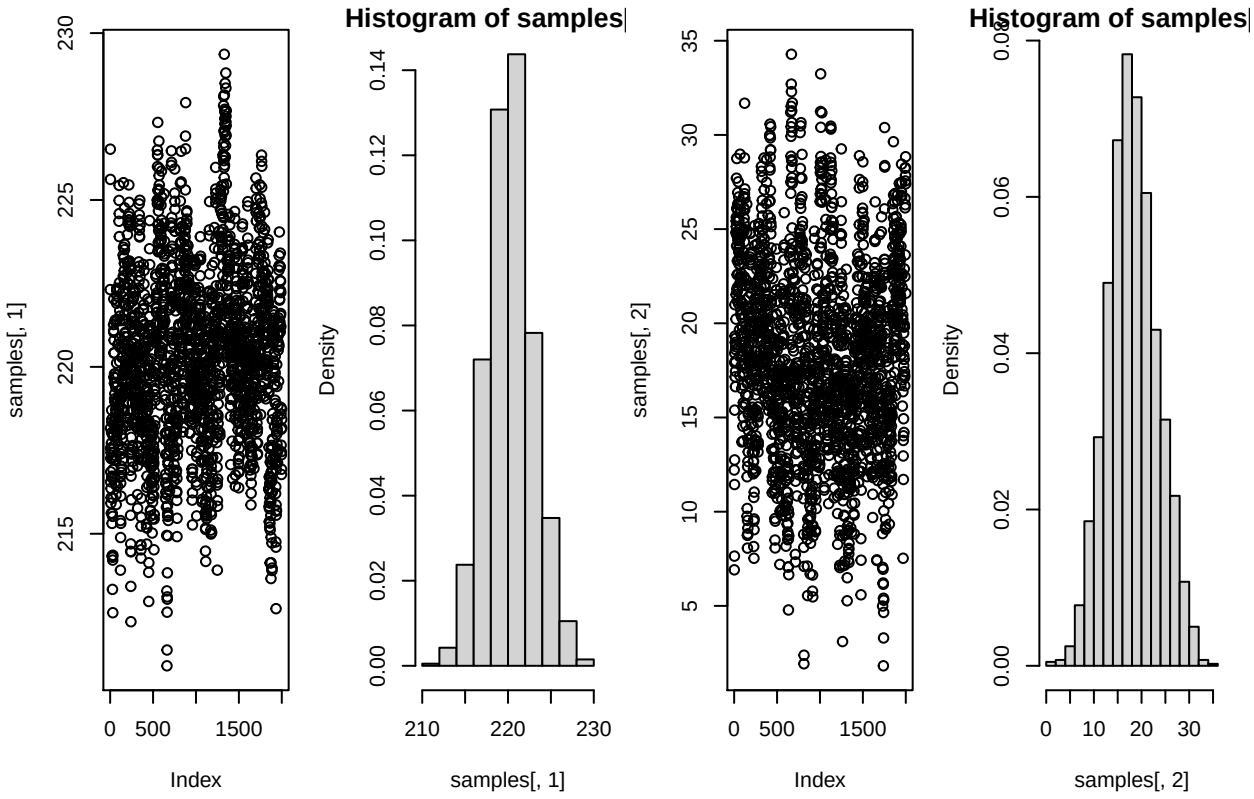
To recover the child mean, remember that we need to combine the group mean and the `adultchild` parameter. We're not actually allowed to just add the values you see above (for good, but technical reasons).

What you actually need to do is add *the individual samples for the two parameters*, and then inspect the output. You can see the individual samples by falling the `fixef` function and setting `summary` to `FALSE`.

```
samples = fixef (female_model, summary = FALSE)
head(samples, 10)
```

```
##           parameters
## iterations Intercept adultchild
##       [1,]  222.3155  12.212789
##       [2,]  226.5228   6.913820
##       [3,]  225.6168   7.641884
##       [4,]  218.1615  11.441671
##       [5,]  217.5661  12.747467
##       [6,]  216.5565  15.390580
##       [7,]  216.8537  18.414595
##       [8,]  217.2630  17.908500
##       [9,]  215.9170  20.987388
##      [10,]  216.3574  19.320220
```

```
par (mfrow = c(1,4))
plot (samples[,1])  ## plot the intercept
hist (samples[,1], freq=FALSE)  ## histogram of the intercept
plot (samples[,2])  ## plot the adultchild parameter
hist (samples[,2], freq=FALSE)  ## histogram of the adultchild parameter
```



We can then summarize the sum of the parameters using the `posterior_summary` function, resulting in a mean, standard deviation, and credible interval for the new parameter:

```
## calculate child mean
child_mean = samples[,1] + samples[,2]
## report mean and spread of samples
posterior_summary (child_mean)
```

```
##      Estimate Est.Error     Q2.5    Q97.5
## [1,] 238.3589  4.275597 229.7993 246.3781
```

Luckily, there is a function in `brms` called `hypothesis` that helps us add terms very easily. You can ask it to add terms in your model (spelled just as they are in the print statement), and to compare the result to some number. If you compare the result to 0, it just tells you about the result of the terms you added.

```
hypothesis(female_model, "Intercept + adultchild = 0")
```

```
## Hypothesis Tests for class b:
##                 Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob Star
## 1 (Intercept+adultc... = 0   238.36      4.28    229.8   246.38         NA        NA    *
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

The output above tells us what our estimate is for `Intercept+adultchild`, which is the mean value for girls in our sample. The credible interval provided is now for the actual girl's mean, not for the difference between the means.

We can also use this approach to find an estimate for the overall mean. Remember that the overall mean has to be exactly between the women's mean and the girls mean. Since the `adultchild` parameter represents the difference between these two means, half the value of this parameter will represent the halfway point. So, we can use:

```
hypothesis(female_model, "Intercept + adultchild/2 = 0")
```

```
## Hypothesis Tests for class b:
##                 Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob Star
## 1 (Intercept+adultc... = 0   229.31      2.47   224.23   233.88         NA        NA    *
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

To find an estimate of our overall mean, and a credible interval for this parameter. If I were writing this is in a paper, at this point I could present this information in a paragraph. I would say something like:

"The overall mean f0 across all speakers was 229 Hz (sd = 2.47, 95% CI = 224, 234). Adult female mean f0 was 220 Hz (sd = 2.75, 95% CI = [215, 226]), while the mean f0 for girls was 228 Hz (sd = 4.28, 95% CI = [230, 246]). The difference between the group means was 17.9 Hz on average (sd = 5.32, 95% CI = [7, 28]), suggesting a small but noisy difference between groups, on average."

As seen in the box plot above, much of the variability in the estimate seems to be inherent, and arises from between-speaker variation in the sample.

1.3.3 'Random' Effects

Above we discussed finding and reporting the 'fixed' population level effects. These effects are common to all subjects in the experiment. Our model also includes what are sometimes called 'random' effects: the speaker-specific means. Our model really does treat these as a random variable, going so far as to estimate a standard deviation for these parameters (reported in the print statement, as described above).

The terms 'fixed' and 'random' effects have several inconsistent and sometimes contradictory definitions. For now, its enough to say that 'random' effects are those for which we estimate a standard deviation term. So, we treat speaker average f0 as a variable, and estimate the characteristics of the distribution of this variable. We are legitimately interested in this standard distribution because it represents between-speaker variation and tells us about our population of speakers. For example, we can make inferences about the *other* speakers that we did not observe in the sample (as discussed in Chapter 1).

The `brms` package also has several functions to help understand our 'random' effect, our speaker-specific intercepts. There is a function called `ranef` which returns random effects estimates in the same way that `fixef` provides fixed effects estimates. The leftmost column of the output below represents the estimated effects associated with each speaker average value.

```
## I am telling it to give me the 'uspeaker' Intercepts, but only the first
## 10 rows. This is just so it doesn't take up the whole page.
ranef (female_model)$uspeaker[1:10,,"Intercept"]
```

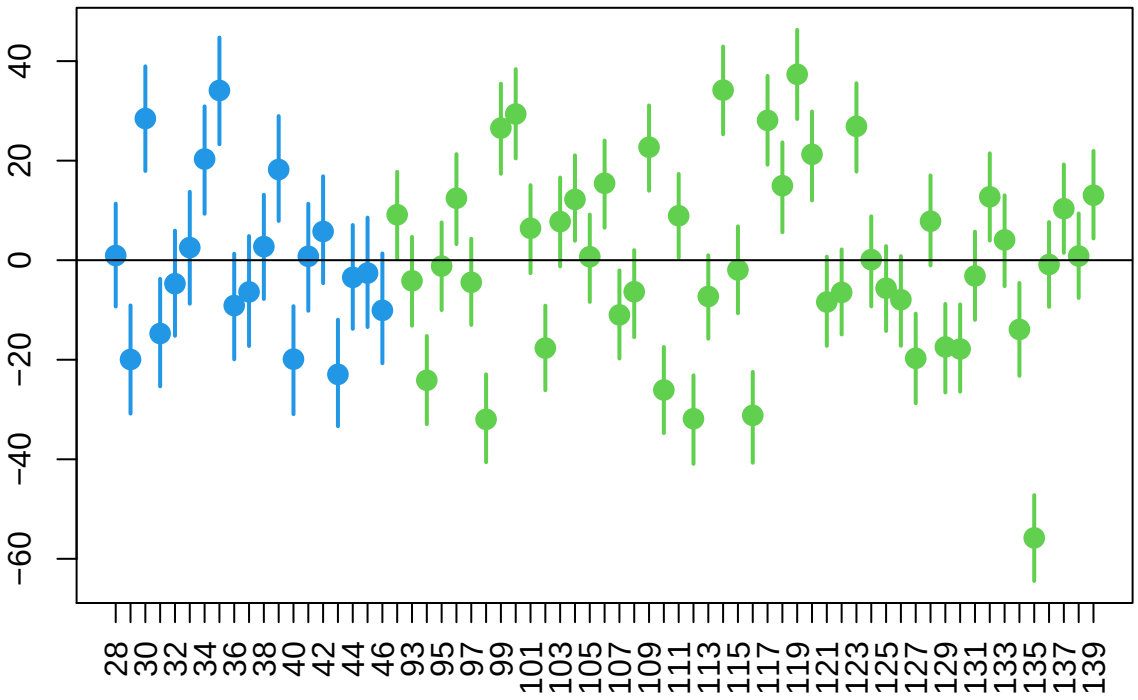
```
##      Estimate Est.Error       Q2.5     Q97.5
## 28   0.938288  5.343401  -9.312154 11.364519
## 29 -19.948653  5.518901 -30.818788 -9.085841
## 30  28.460902  5.505425  17.922695 38.949702
## 31 -14.728995  5.412624 -25.348544 -3.748185
## 32  -4.726466  5.416382 -15.183818  5.941205
## 33   2.537247  5.699444  -8.737180 13.747747
## 34  20.327908  5.499930   9.329407 30.910764
## 35  34.103775  5.570118  23.284610 44.733556
## 36  -9.110753  5.373997 -19.908470  1.296369
## 37  -6.371143  5.575863 -17.235411  4.844565
```

Notice that the speaker averages vary around 0, and some are even negative? That is because regression models encode **differences**, not average values directly. This is the same issue that came up previously about how to represent the difference in f0 between women and girls. So, what the speaker-specific mean terms tell us is: what is the average value for this speaker, relative to the average for their group?

For example, we see that the mean for the second speaker above is -19 Hz. This means that they have a lower f0 than 'expected', and their mean f0 is 19 Hz lower than their group mean f0. In contrast, the first speaker has a speaker mean effect that is nearly zero (0.93). That tells you that this speaker's mean f0 was nearly the same as that of the average for their category.

We can use a function I wrote to look at the distribution of speaker-averages represented in the table above:

```
par (mfrow = c(1,1))
brmplot( ranef (female_model)[["uspeaker"]][,,"Intercept"], col = colors)
abline (h=0)
```



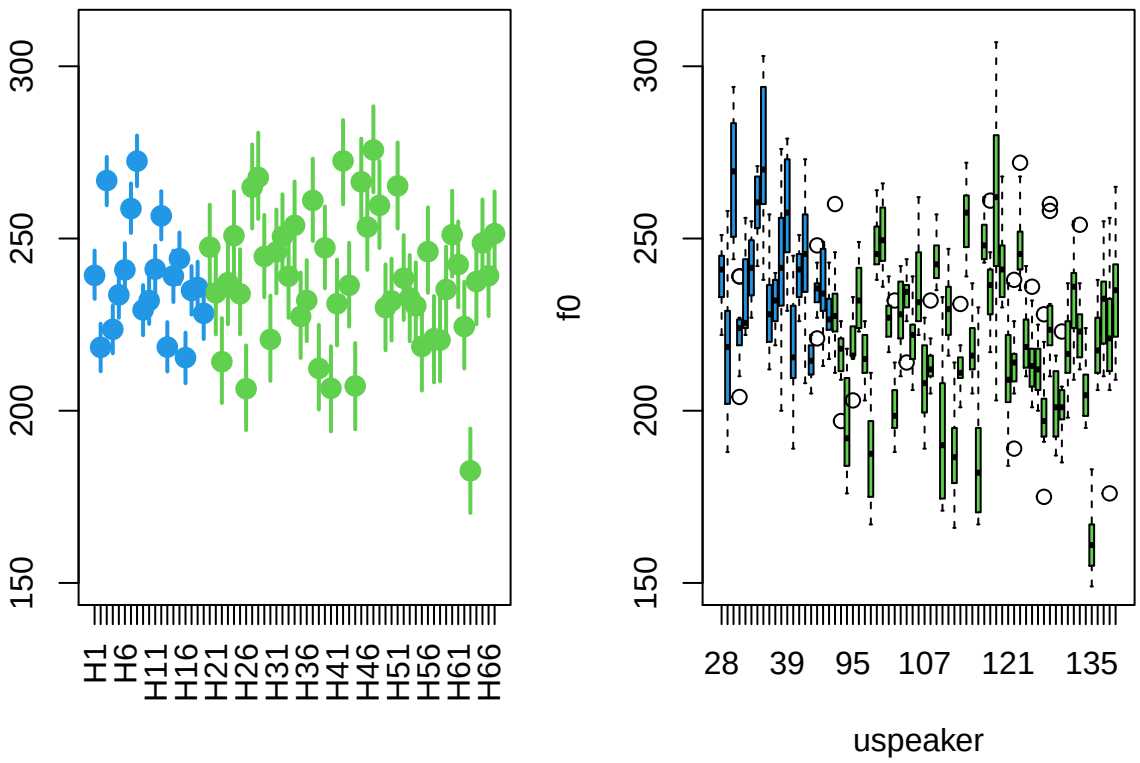
Obviously, it would be useful to get the *actual* speaker means, and credible intervals for these means. To do this, we need to find the group mean, and add the subject-specific effect to it. This is a similar challenge to that presented by recovering the mean for girls outlined above. We can use the `hypothesis` function to help us do this too, as shown below:

```
## get speaker averages
speaker_means = hypothesis(female_model, "Intercept + adultchild = 0",
                           group = "uspeaker", scope = "coef")
```

The above expands on our previous use of `hypothesis` by telling the function to include the random variation according to `uspeaker` in our mean estimates. This provides as output as before, but since it has a row for each of 96 subject, I won't print it.

We can compare the estimates of speaker-specific means to the distribution of actual data arranged by subject. There is a close correspondence.

```
## plot comparison of estimates of speaker means to actual data
par (mfrow = c(1,2))
brmplot( posterior_summary (speaker_means$samples), col = colors,
         ylim = c(150,310))
boxplot (f0 ~ uspeaker, data=females, col = colors, ylim = c(150,310))
```



1.3.4 Thinking of models as sums of effects

Regression models try to break up values into their components. This is why the effects are expressed in terms of differences. For example, if we say that that average is 220 Hz and the effect is 0 Hz, then it has no effect (i.e., it causes no difference). On the other hand something that *does* cause a difference does have an effect, and we can express the effect in terms of the difference it causes. More generally, we can think of any variable as the sum of a bunch of independent effects.

For example, according to our model, the observed f0 for each token is equal to:

f0 = group mean + speaker effect + random error

Meaning that we expect and given observed f0 to be equal to the average value for the group, plus some speaker-specific difference, plus some random error. For example if we observe a value of 256 for an adult female that produced an average f0 of 240 Hz:

256 = 220 (adult female mean) + 20 (speaker effect) + 16 (error)

This reflects that this speaker's average f0 is 20 Hz above the average for adult female's and that this production was 16 Hz above the speaker's mean. Another observation from this talker might be:

237 = 220 (adult female mean) + 20 (speaker effect) - 3 (error)

Indicating an error of -3 since the production is 3 Hz *below* the speaker average.

Our `brm` model above encodes each of these sources of variation:

- the population parameters (`Intercept`, `adultchild`) represent variation in group means.

- the `uspeaker` random Intercepts represent variation in speaker means.
- the `sigma` parameter represents the random error.

1.4 Checking the model fit and specifying priors

Look at the output of the model we fit above:

```
female_model
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: f0 ~ adult + (1 | uspeaker)
##    Data: females (Number of observations: 804)
## Samples: 1 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 2000
##
## Group-Level Effects:
## ~uspeaker (Number of levels: 67)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    19.13      1.77    16.01    22.88 1.00      177      289
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    220.26      2.75   215.12   225.97 1.00       69      108
## adultchild    18.10      5.21     8.15    28.55 1.00       94      254
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.95      0.33    12.30    13.60 1.00     1772     1478
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

I want to mention two things that are especially extremely important for 'real' work when you want to be confident in your model: checking that the model is reliable, and specifying appropriate priors.

1.4.1 Checking the model fit

Remember that our model parameter estimates consist of a set of samples from the parameter likelihood. If we don't take enough of these samples, our parameter estimates will be unreliable.

For this reason, it's important to look at the ESS values (the 'expected sample size'), and the 'Rhat' values. ESS tells you about how many independent samples you have taken from the likelihood. Bulk ESS is how many samples the sampler took in the thick part of the density, and Tail ESS reflects how much time the sampler spent in the thin part, the 'tails'. Rhat tells you about whether your 'chains' have converged (more on this later). As noted above, values of Rhat near 1 are good, and values higher than around 1.1 are a bad sign.

We haven't really taken many samples here, so we can't be confident in our parameter estimates. Ideally we would like several hundred samples (at least) for mean estimates, and thousands to be confident in the 95% confidence intervals.

To get more samples we can run the model longer, or we can use more *chains*. A chain is basically a separate set of samples for your parameter values. Just imagine you had estimated the model 4 times in a row and mixed your estimations. A model can be fit in parallel across several chains, and then the estimates can be merged across chains. When you do this across multiple cores, you can get N times as many samples when you use N cores.

Below, I refit the same model from above but run it on 4 chains, and on 4 cores at once. This doesn't take any longer but it does give us a higher ESS. Just make sure you leave a couple of cores free on your computer when you fit a model!

```
set.seed (1)
female_model_multicore =
  brm (f0 ~ adult + (1|uspeaker), data = females, chains = 4, cores = 4,
       iter = 4000)
```

```
female_model_multicore = readRDS ('2_female_model_multicore.RDS')
```

If we compare the ESS for this new model to the previous model, we see that using 4 chains has substantially increased our ESS. Notice that the value of Rhat has also gone up a bit. This is because Rhat measures the 'differentness' of the 4 chains. If they are all accurate, they should be very similar, and Rhat should equal nearly 1. When our model only had one chain, Rhat was not telling us anything.

```
# model run on 1 chain only
female_model
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: f0 ~ adult + (1 | uspeaker)
##    Data: females (Number of observations: 804)
## Samples: 1 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 2000
##
## Group-Level Effects:
## ~uspeaker (Number of levels: 67)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    19.13      1.77    16.01    22.88 1.00      177      289
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    220.26      2.75   215.12   225.97 1.00       69      108
## adultchild    18.10      5.21     8.15    28.55 1.00       94      254
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.95      0.33    12.30    13.60 1.00     1772     1478
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# model run on 4 chains
female_model_multicore
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: f0 ~ adult + (1 | uspeaker)
##    Data: females (Number of observations: 804)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~uspeaker (Number of levels: 67)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    19.10      1.80    16.03    23.15 1.00      783     1586
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    220.27      2.91   214.67   226.01 1.02      381      935
## adultchild    18.19      5.42     7.86    28.92 1.01      588     1084
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.94      0.34    12.29    13.61 1.00     5212     5483
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

1.4.2 Specifying prior probabilities

If you don't specify prior probabilities for your parameters, a 'flat' prior is assumed. This is not only illogical in many cases, but can also cause problems for `brms`. Basically, the sampler has a harder time figuring out the most likely values when you tell it to look anywhere from positive to negative infinity. Even a bit of guidance can help.

`brms` makes it easy to specify prior probabilities for specific parameters or whole groups of parameters. First we figure out the overall mean and the standard deviation of the data.

```
mean(females$f0)
```

```
## [1] 225.4913
```

```
sd(females$f0)
```

```
## [1] 23.98959
```

In the example below, I use this information to set reasonable bounds on the parameters in the model. I do this by class of parameter:

- Intercept: this is a unique class, only for intercepts.
- b: this is for all the non-intercept effects terms. In our example this is only `adultchild`.

- sd: this is for all the higher-level standard deviation parameters. In our example this is only `sd(Intercept)` for `uspeaker`.

```
set.seed (1)
female_model_priors =
  brm (f0 ~ adult + (1|uspeaker), data = females, chains = 4, cores = 4,
       iter = 4000, prior = c(set_prior("student_t(3, 225, 50)", class = "Intercept"),
                              set_prior("student_t(3, 0, 50)", class = "b"),
                              set_prior("student_t(3, 0, 50)", class = "sd")))
```

```
female_model_priors = readRDS ('2_female_model_priors.RDS')
```

The output of this model can be compared to our original model. We see that the priors have resulted in a higher ESS, while also a better Rhat value than the multicore model without specified priors. In addition, specifying a prior has has no noticeable effect on our results. This is because the prior matters less and less when you have a lot of data, and because we have set a wide prior that is mostly appropriate for our data.

```
# model run on 1 chain only, no priors
female_model
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: f0 ~ adult + (1 | uspeaker)
##    Data: females (Number of observations: 804)
## Samples: 1 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 2000
##
## Group-Level Effects:
## ~uspeaker (Number of levels: 67)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    19.13      1.77    16.01    22.88 1.00      177      289
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    220.26      2.75   215.12   225.97 1.00       69      108
## adultchild    18.10      5.21     8.15    28.55 1.00       94      254
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.95      0.33    12.30    13.60 1.00     1772     1478
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# model run on 4 chains, with priors
female_model_priors
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: f0 ~ adult + (1 | uspeaker)
##    Data: females (Number of observations: 804)
```

```
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~uspeaker (Number of levels: 67)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    19.19      1.81    16.02    23.25 1.01      716      868
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    220.61      2.71   215.34   225.84 1.00      352      735
## adultchild    17.45      5.18     7.48    27.76 1.01      452      891
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    12.95      0.34    12.30    13.64 1.00     5775     6025
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

1.5 But what does it all mean?

Ok so are the f0s produced by women and girls different? A detailed look at our results (and figures) above suggests that:

- the magnitude of between speaker variation is **larger** than the difference between girls and women (19 Hz vs 18/17 Hz). This means that random people drawn from the two groups are largely to overlap.
- the magnitude of within-speaker variation (12 Hz) is almost as large as the group difference and the between-speaker difference! This means that two random productions from two different people might be the same, even when the speakers average f0s are quite different.

And yet:

- the mean f0 **is** reliably different between women and girls, and there are well-known anatomical reasons for this (i.e., this was expected a priori).
- the between-speaker differences are 'random' from person to person, but systematic for a given person.
- even the within-speaker variation may be systematic given a more-complicated model.

So are they different yes or no? Statistics aside, a fair assessment of our data suggests that neither binary conclusions is fully supported, they are distinguishable but overlapping substantially. So, our results basically mean whatever we want them to mean, as long as reviewers (and readers in general) will believe us. The meaning is in our heads and not in the model. If you want to use this model to highlight the differences between girls and women, I think it is valid. I also think it would be valid to use this data to highlight between and within-speaker variation. Both are true! The model is simply a reflection of the relationships in our data, and the interpretation is up to us.

This may sound like I am advocating that people should feel free to draw any conclusions given their data, but keep in mind that the "as long as reviewers (and readers in general) will believe us" component is crucial. The results of the model will need to be interpreted in the larger context of the work it is presented in, and in terms of scientific and general knowledge that readers have. The results of any model will need to 'make

sense' given this, and a statistical result on its own will not be enough to make people believe outlandish claims.

The model is not reality and should not be confused with reality. This is a very important point! A statistical finding does not *prove* that something is *true*. This kind of thinking has caused many problems for scientists recently. We can imagine that 10 people might have approached the question of woman-girl f0 differences in different ways, a concept known as researcher degrees of freedom. This would cause in slight differences in their results, resulting in a sort of 'sampling distribution' for their results.

For example, we know for a fact that f0 varies across vowel categories, a concept known as (intrinsic f0)[https://www.sciencedirect.com/science/article/abs/pii/S0095447095801650]. A model that included vowel category as a within-speaker predictor would reduce the apparent error in our model, and might affect the precision of our other estimates. Would this new model invalidate our current model? This thinking is problematic because there is *always* a better model that could then invalidate our current model.

The solution is to think of your model not as a mathematical implementation of *reality* but instead as a mathematical implementation of your research questions. Your model should include the information and structure that you think are necessary to represent and investigate your questions.

Using a different model can result in different results given the same data, but asking a different question can also obviously lead to different results given then same data! One of my favorite phrases to use is "given our data and model structure". This phrase is helpful because it communicates that you understand that your results are contingent on:

1) the data you collected. Given other data you may have come to other conclusions.

2) the model you chose. Given another model you may have come to other conclusions.