

17.9: Bayesian ANOVA

As you can tell, the `BayesFactor` package is pretty flexible, and it can do Bayesian versions of pretty much everything in this book. In fact, it can do a few other neat things that I haven't covered in the book at all. However, I have to stop somewhere, and so there's only one other topic I want to cover: Bayesian ANOVA.

quick refresher

As with the other examples, I think it's useful to start with a reminder of how I discussed ANOVA earlier in the book. First, let's remind ourselves of what the data were. The example I used originally is the `clin.trial` data frame, which looks like this

```
load("../rbook-master/data/clinicaltrial.Rdata")
head(clin.trial)
```

```
##      drug      therapy mood.gain
## 1 placebo no.therapy      0.5
## 2 placebo no.therapy      0.3
## 3 placebo no.therapy      0.1
## 4 anxifree no.therapy      0.6
## 5 anxifree no.therapy      0.4
## 6 anxifree no.therapy      0.2
```

To run our orthodox analysis in earlier chapters we used the `aov()` function to do all the heavy lifting. In Chapter 16 I recommended using the `Anova()` function from the `car` package to produce the ANOVA table, because it uses Type II tests by default. If you've forgotten what "Type II tests" are, it might be a good idea to re-read Section 16.10, because it will become relevant again in a moment. In any case, here's what our analysis looked like:

```
library(car)
```

```
## Loading required package: carData
```

```
model <- aov( mood.gain ~ drug * therapy, data = clin.trial )
Anova(model)
```

```
## Anova Table (Type II tests)
##
## Response: mood.gain
##              Sum Sq Df F value    Pr(>F)
## drug           3.4533  2 31.7143 1.621e-05 ***
## therapy         0.4672  1  8.5816  0.01262 *
## drug:therapy    0.2711  2  2.4898  0.12460
## Residuals       0.6533 12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

That's pretty clearly showing us evidence for a main effect of `drug` at $p < .001$, an effect of `therapy` at $p < .05$ and no interaction.

Bayesian version

How do we do the same thing using Bayesian methods? The `BayesFactor` package contains a function called `anovaBF()` that does this for you. It uses a pretty standard `formula` and `data` structure, so the command should look really familiar. Just like we did with regression, it will be useful to save the output to a variable:

```
models <- anovaBF(
  formula = mood.gain ~ drug * therapy,
  data = clin.trial
)
```

The output is quite different to the traditional ANOVA, but it's not too bad once you understand what you're looking for. Let's take a look:

```
models
```

This looks very similar to the output we obtained from the `regressionBF()` function, and with good reason. Remember what I said back in Section 16.6: under the hood, ANOVA is no different to regression, and both are just different examples of a linear model. Because of this, the `anovaBF()` reports the output in much the same way. For instance, if we want to identify the best model we could use the same commands that we used in the last section. One variant that I find quite useful is this:

```
models/max(models)
```

```
## Bayes factor analysis
## -----
## [1] drug                : 0.3521042    ±0.94%
## [2] therapy            : 0.001047568 ±0.94%
## [3] drug + therapy     : 1                ±0%
## [4] drug + therapy + drug:therapy : 0.978514    ±1.29%
##
## Against denominator:
##   mood.gain ~ drug + therapy
## ---
## Bayes factor type: BFlinearModel, JZS
```

By “dividing” the `models` output by the best model (i.e., `max(models)`), what R is doing is using the best model (which in this case is `drugs + therapy`) as the denominator, which gives you a pretty good sense of how close the competitors are. For instance, the model that contains the interaction term is almost as good as the model without the interaction, since the Bayes factor is 0.98. In other words, the data do not clearly indicate whether there is or is not an interaction.

Constructing Bayesian Type II tests

Okay, that's all well and good, you might be thinking, but what do I report as the alternative to the p-value? In the classical ANOVA table, you get a single p-value for every predictor in the model, so you can talk about the significance of each effect. What's the Bayesian analog of this?

It's a good question, but the answer is tricky. Remember what I said in Section 16.10 about ANOVA being complicated. Even in the classical version of ANOVA there are several different “things” that ANOVA might correspond to. Specifically, I discussed how you get different p-values depending on whether you use Type I tests, Type II tests or Type III tests. To work out which Bayes factor is analogous to “the” p-value in a classical ANOVA, you need to work out which version of ANOVA you want an analog for. For the purposes of this section, I'll assume you want Type II tests, because those are the ones I think are most sensible in general. As I discussed back in Section 16.10, Type II tests for a two-way ANOVA are reasonably straightforward, but if you have forgotten that section it wouldn't be a bad idea to read it again before continuing.

Assuming you've had a refresher on Type II tests, let's have a look at how to pull them from the Bayes factor table. Suppose we want to test the main effect of `drug`. The null hypothesis for this test corresponds to a model that includes an effect of `therapy`, but no effect of `drug`. The alternative hypothesis is the model that includes both. In other words, what we want is the Bayes factor corresponding to this comparison:

```
knitr::kable(tibble::tribble(
  ~V1, ~V2,
  "Null model:", "`mood.gain ~ therapy`",
  "Alternative model:", "`mood.gain ~ therapy + drug`"
), col.names = c("", ""))
```

Null model:	mood.gain ~ therapy
Alternative model:	mood.gain ~ therapy + drug

As it happens, we can read the answer to this straight off the table because it corresponds to a comparison between the model in line 2 of the table and the model in line 3: the Bayes factor in this case represents evidence *for* the null of 0.001 to 1. Or, more helpfully, the odds are about 1000 to 1 against the null.

The main effect of `therapy` can be calculated in much the same way. In this case, the null model is the one that contains only an effect of `drug`, and the alternative is the model that contains both. So the relevant comparison is between lines 2 and 1 in the table. The odds in favour of the null here are only 0.35 to 1. Again, I find it useful to frame things the other way around, so I'd refer to this as evidence of about 3 to 1 in favour of an effect of `therapy`.

Finally, in order to test an interaction effect, the null model here is one that contains both main effects but no interaction. The alternative model adds the interaction. That is:

```
knitr::kable(tibble::tribble(
  ~V1, ~V2,
  "Null model:", "`mood.gain ~ drug + therapy`",
  "Alternative model:", "`mood.gain ~ drug + therapy + drug:therapy`"
), col.names = c("", ""))
```

Null model:	mood.gain ~ drug + therapy
Alternative model:	mood.gain ~ drug + therapy + drug:therapy

If we look those two models up in the table, we see that this comparison is between the models on lines 3 and 4 of the table. The odds of 0.98 to 1 imply that these two models are fairly evenly matched.

You might be thinking that this is all pretty laborious, and I'll concede that's true. At some stage I might consider adding a function to the `lsr` package that would automate this process and construct something like a "Bayesian Type II ANOVA table" from the output of the `anovaBF()` function. However, I haven't had time to do this yet, nor have I made up my mind about whether it's really a good idea to do this. In the meantime, I thought I should show you the trick for how I do this in practice. The command that I use when I want to grab the right Bayes factors for a Type II ANOVA is this one:

```
max(models)/models
```

```
##              denominator
## numerator      drug  therapy drug + therapy
## drug + therapy 2.840068 954.5918             1
##              denominator
## numerator      drug + therapy + drug:therapy
## drug + therapy              1.021958
```

The output isn't quite so pretty as the last one, but the nice thing is that you can read off everything you need. The best model is `drug + therapy`, so all the other models are being compared to that. What's the Bayes factor *for* the main effect of `drug`? The relevant null hypothesis is the one that contains only `therapy`, and the Bayes factor in question is 954:1. The main effect of `therapy` is weaker, and the evidence here is only 2.8:1. Finally, the evidence *against* an interaction is very weak, at 1.01:1.

Reading the results off this table is sort of counterintuitive, because you have to read off the answers from the “wrong” part of the table. For instance, the evidence for an effect of `drug` can be read from the column labelled `therapy`, which is pretty damned weird. To be fair to the authors of the package, I don't think they ever intended for the `anovaBF()` function to be used this way. My understanding²⁷³ is that their view is simply that you should find the best model and report that model: there's no inherent reason why a Bayesian ANOVA should try to follow the exact same design as an orthodox ANOVA.²⁷⁴

In any case, if you know what you're looking for, you can look at this table and then report the results of the Bayesian analysis in a way that is pretty closely analogous to how you'd report a regular Type II ANOVA. As I mentioned earlier, there's still no convention on how to do that, but I usually go for something like this:

A Bayesian Type II ANOVA found evidence for main effects of drug (Bayes factor: 954:1) and therapy (Bayes factor: 3:1), but no clear evidence for or against an interaction (Bayes factor: 1:1).

This page titled [17.9: Bayesian ANOVA](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Danielle Navarro](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.