

Cmd> #

The Stat5303 package includes two functions of use here: `power.anova.test()` and `sample.size.anova()`. You won't be surprised to learn that the first computes power and the second computes sample sizes, both for ANOVA.

In order to compute power you need to be able to compute noncentrality, which depends on means, sample sizes, and the error variance. In the power function, you can specify the noncentrality in two ways. You can either give the means, the sample sizes, and the error variance, or you can give the noncentrality, the numerator degrees of freedom, and the denominator degrees of freedom. You also need to provide the type one error rate, but by default it is assumed to be .05. The value for `power.anova.test` is simply the power.

Sample size is computed to obtain a particular power. In this case you must specify the power that you wish to obtain and the type I error rate. You also have to help it out with how to compute noncentrality. You do this by providing either treatment means and error variance or "n=1" noncentrality and number of groups. The value for `sample.size.anova` is a list with the required sample sizes and the achieved power.

When finding sample size with the means specified, you can also provide an argument called "propvec", which is short hand for proportionality vector. The propvec should have the same number of entries as there are means. If you provide a propvec, then the computed sample sizes will roughly be proportional to the entries of propvec.

The noncentrality parameter is  $\sum_{i=1}^g n_i \alpha_i^2 / \sigma^2$ , where the  $\alpha_i$ s here use the restriction that  $\sum n_i \alpha_i = 0$ . The  $n = 1$  "noncentrality parameter" is  $\sum_{i=1}^g \alpha_i^2 / \sigma^2$ .

> **power.anova.test(means=c(6, 6, 5, 4, 4, 5), ns=rep(3, 6), sigma2=1, alpha=.05)**

OK, suppose that we have six groups, and we think that an interesting alternative would be two with mean 4, two with mean 5, and two with mean 6. We think that the error variance is 1, and we're testing at the .05 level. What is the power?

[1] 0.5529867

> **power.anova.test(means=c(1, 1, 0, -1, -1, 0), ns=rep(3, 6), sigma2=1, alpha=.05)**

You can add or subtract a constant from all the means, and it won't change the power. It's only the differences between means that matter.

[1] 0.5529867

> **power.anova.test(means=c(1, 1, 0, -1, -1, 0), ns=3, sigma2=1, alpha=.05)**

If all the sample sizes are the same, then you can just put in one value for the sample size.

[1] 0.5529867

> **power.anova.test(means=c(1, 1, 0, -1, -1, 0), ns=2, sigma2=1, alpha=.05)**

Smaller sample size means less power.

[1] 0.2715161

> **power.anova.test(means=c(1, 1, 0, -1, -1, 0), ns=4, sigma2=1, alpha=.05)**

Bigger sample size means more power.

[1] 0.7640361

```
> (4*1+4*1+4*0+4*1+4*1+4*0)/1
```

Compute the noncentrality parameter when all sample sizes are four. The treatment effects are either plus or minus one, so squaring just gives us one. And the error variance is also one. All in all, this is a pretty boring noncentrality parameter.

```
[1] 16
```

```
> power.anova.test(ncp=16, df1=5, df2=18, alpha=.05)
```

This is the alternate form for power using noncentrality and degrees of freedom. While this is a little more trouble to use, it's actually a lot more flexible and can be used in more complex situations.

```
[1] 0.7640361
```

```
> power.anova.test(means=c(6, 6, 5, 4, 4, 5), ns=4, sigma2=1, alpha=.01)
```

A smaller type I error leads to lower power.

```
[1] 0.4812791
```

```
> power.anova.test(means=c(1, 1, 0, -1, -1, 0), ns=c(4, 3, 2, 4, 3, 2), sigma2=1, alpha=.05)
```

Here we have the same total sample size (18) as three in every group, but we put more replications on the treatments in the extremes and fewer on those in the middle. More data on means in the extremes increases power more than more data on those in the center. Of course, you have to know where the extremes are to make this work, and that is very unrealistic.

```
[1] 0.6283721
```

```
> sample.size.anova(.4, .05, means=c(6, 6, 5, 4, 4, 5), sigma2=1)
```

Here we find the smallest sample size that will give us at least power .4 when testing at the .05 level with the means and error variance as specified. It shows us the  $n_i$ s and the achieved power. Note that we can't hit the requested power right spot on, we've overshot. Above we saw that  $n=2$  gave us power .27, so we need  $n=3$ .

```
$nis
```

```
[1] 3 3 3 3 3 3
```

```
$power
```

```
[1] 0.5529867
```

```
> sample.size.anova(.4, .05, ncpl=4, ngrps=6)
```

Alternatively, we can specify the noncentrality parameter for  $n=1$  and the number of groups.

```
$nis
```

```
[1] 3 3 3 3 3 3
```

```
$power
```

```
[1] 0.5529867
```

```
> sample.size.anova(.9, .05, means=c(6, 6, 5, 4, 4, 5), sigma2=1)
```

Asking for more power means that you'll need more data.

```
$nis
```

```
[1] 6 6 6 6 6 6
```

```
$power
```

```
[1] 0.9520672
```

```
> sample.size.anova(.9, .01, means=c(6, 6, 5, 4, 4, 5), sigma2=1)
```

Testing at a lower type I error rate means that you'll need more data to keep the same power.

```
$nis
```

```
[1] 7 7 7 7 7 7
```

```
$power
```

```
[1] 0.9112071
```

```
> sample.size.anova(.9, .01, means=c(6, 6, 5, 4, 4, 5), sigma2=2)
```

Larger error variance (which gives you a smaller noncentrality parameter) means you need more data to get the required power.

```
$nis
```

```
[1] 13 13 13 13 13 13
```

```
$power
```

```
[1] 0.9220537
```

```
> sample.size.anova(.4, .05, ncp1=2^2/2/1, ngrps=6)
```

Often you can't say what you think all the treatment effects will be. If you can give a value  $D$ , such that any treatment differences of  $D$  or greater are "interesting," then the smallest possible noncentrality with differences of  $D$  is  $nD^2/(2\sigma^2)$ . Designing for this noncentrality gives us worst case power for differences of size  $D$ . So if we get the power we want for this worst case noncentrality, then we'll get at least that much power for any other noncentrality from the same error variance and differences of at least  $D$ .

Suppose we have a  $D$  of 2 ( $=6-4$ ) and our error variance is still 1. The minimal  $n=1$  noncentrality in this case is 2 (instead of the 4 we had in examples above which had more groups spread out). We now need at least 4 in each group. To get at least power .4

```
$nis
```

```
[1] 4 4 4 4 4 4
```

```
$power
```

```
[1] 0.4322055
```

```
> sample.size.anova(.4, .05, means=c(-1, 1, 0, 0, 0, 0), sigma2=1)
```

Same thing but directly using means that show the difference  $D$ .

```
$nis
```

```
[1] 4 4 4 4 4 4
```

```
$power
```

```
[1] 0.4322055
```

```
> sample.size.anova(.9, .01, means=c(6, 6, 5, 4, 4, 5), sigma2=2, propvec=c(1, 1, 2, 2, 2, 2))
```

Finally, the sample size function has a limited ability to find unequal sample sizes using the propvec argument. For the sake of argument, assume that treatments 1 and 2 are more expensive than the other treatments, so in general we would like to use fewer replications there. In this example we ask for (roughly) twice as many replications in treatments 3 through 6.

```
$nis
```

```
[1] 9 9 17 17 17 17
```

```
$power
```

```
[1] 0.9047962
```

```
> power.anova.test(ncp=16, df1=1, df2=18, alpha=.05)
```

Here is a useful fact: for the same amount of noncentrality and total number of data, it is easier to detect a difference with fewer groups than with more groups. In some sense it is easier to find the difference if there is only one direction to look in.

```
[1] 0.9655275
```

```
> power.anova.test(ncp=16, df1=3, df2=16, alpha=.05)
```

```
[1] 0.8556421
```

```
> power.anova.test(ncp=16, df1=4, df2=15, alpha=.05)
```

```
[1] 0.791441
```

```
> power.anova.test(ncp=16, df1=9, df2=10, alpha=.05)
```

```
[1] 0.4608024
```

```
> ntotry <- 2:20
```

It is sometimes useful to compute power for a sequence of sample sizes and then make a plot.

```
> power <- ntotry
```

```
> for(i in 1:length(ntotry)) {
```

```
+ power[i] <- power.anova.test(means=c(6, 6, 5, 4, 4, 5), ns=ntotry[i], sigma2=1, alpha=.05)
```

```
+ }
```

```
> power
```

```
[1] 0.2715161 0.5529867 0.7640361 0.8888004 0.9520672 0.9807802
```

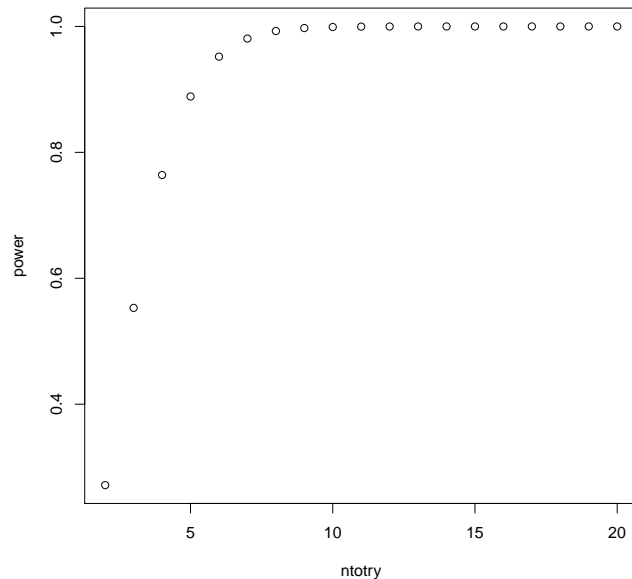
```
[7] 0.9927432 0.9973962 0.9991059 0.9997045 0.9999056 0.9999707
```

```
[13] 0.9999912 0.9999974 0.9999993 0.9999998 0.9999999 1.0000000
```

```
[19] 1.0000000
```

> **plot (ntotry, power)**

What should be obvious is that there are serious diminishing returns. Once the power gets pretty big, we can add a lot of additional sample size with very little actual increase in power (it can't be bigger than 1.0).



> #

Here is a serious example. We want to find the effect of certain diets on the blood concentration of estradiol in premenopausal women. We have historical data on six subjects. For each woman, estradiol was measured twice midfollicular diet 1, twice midfollicular diet 2, twice early follicular diet 1, and twice early follicular diet 2. A cursory inspection of the data reveals that variance changes with mean. On the log scale, variance seems fairly constant at about .109 and does not seem to change across diet or cycle phase.

The response of interest is post-diet minus pre-diet change in blood estradiol. This difference should have a variance of about  $2 \times .109 = .218$ .

> **sample.size.anova (.95, .05, ncp1=.5^2/2/.218, ngrps=2)**

We want 95% power to detect a change of 0.5 log units (roughly 65%), testing at the 0.05 level.

\$nis

[1] 24 24

\$power

[1] 0.9526565

> **sample.size.anova (.95, .05, ncp1=.75^2/2/.218, ngrps=2)**

To detect a larger shift (slightly more than a doubling), we need fewer subjects.

\$nis

[1] 12 12

\$power

[1] 0.9638958