

How to calculate size and power

James Curran

Univariate case

Let's assume that I am interested in the size and the power of the two-sample t -test. The size is the Type I Error rate - the proportion of time that we falsely reject the null hypothesis when in fact it is true. If we are using an α value of 0.05, then we would like the size to be as close as possible to 0.05. If $\alpha = 0.01$ then we want the size to be as close as possible to 0.01 and so on.

So how do we do this? At the simplest level, we take two samples from the **same** normal distribution multiple times and count the number of times the P -value is less than α . There are two factors that could have an effect on this. Firstly, there is the issue of sample size. In general, we expect hypothesis tests to perform better as the sample size increases. Secondly, the pooled (or standard) t -test assumes that the samples originate from distributions that have the same standard deviation (or variance). Rather than mess around with Welch's t - we will just assume that that assumption holds - because this is what is assumed in the multivariate case.

```
set.seed(123)
n = expand.grid(n1 = 2:10, n2 = 2:10) ## range of sample sizes we will explore
N = 1000 ## the number of samples we will take for each
      ## sample size combination
alpha = 0.05
size = rep(0, nrow(n))

for(i in 1:nrow(n)){
  n1 = n[i,1]
  n2 = n[i,2]

  count = 0

  ## Note I could vectorise and use apply
  ## but I haven't because it is harder to understand
```

```

for(r in 1:N){
  x = rnorm(n1)
  y = rnorm(n2)
  p = t.test(x, y)$p.value

  if(p <= alpha){
    count = count + 1
  }
}

size[i] = count / N
}

```

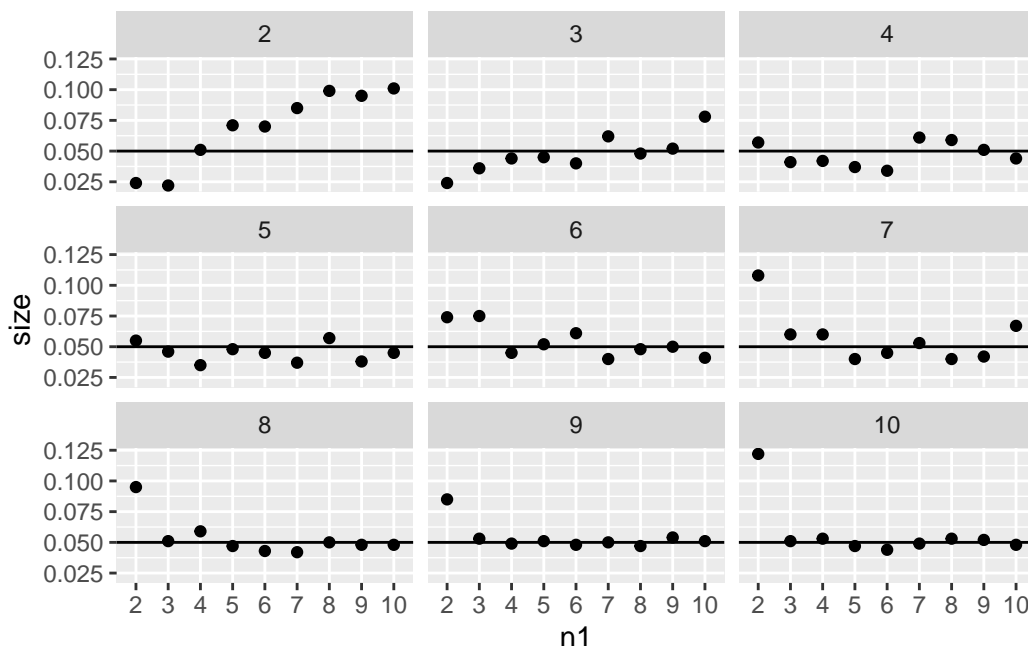
It is useful to see the results graphically

```

library(ggplot2)
plot.df = data.frame(n1 = factor(n[,1]),
                     n2 = factor(n[,2]),
                     size = size)

g = plot.df |>
  ggplot(aes(x = n1, y = size)) +
  geom_point() +
  geom_hline(yintercept = alpha) +
  facet_wrap(vars(n2), nrow = 3)
g

```



Power is equal to $1 - \beta$ where β is Type II Error – i.e. the probability that we will fail to reject the null hypothesis when it is actually false. What we need, on top of changing sample size, is a systematic way of making the null hypothesis false. We can do this by taking the first sample from $N(\mu_1, \sigma^2)$, and the second sample from $N(\mu_2, \sigma^2)$ where $\mu_2 = \mu_1 + \delta\sigma$, and $\delta \in \mathbb{R}^+$. This means we sample from two distributions which have the same standard deviation, but are δ standard deviations apart. We expect that as δ increases, our ability to reject the null hypothesis should increase. So how do we do that in R? It is very similar to the size experiments, but we have an additional parameter delta. That is,

```
set.seed(123)
sim.params = expand.grid(n1 = 2:10,
                        n2 = 2:10,
                        delta = seq(0.5, 3.5, by = 0.5))
N = 1000 ## the number of samples we will take for each
        ## sample size combination
alpha = 0.05
power = rep(0, nrow(sim.params))

for(i in 1:nrow(sim.params)){
  n1 = sim.params$n1[i]
  n2 = sim.params$n2[i]
  delta = sim.params$delta[i]
```

```

count = 0

## Note I could vectorise and use apply
## but I haven't because it is harder to understand
for(r in 1:N){
  x = rnorm(n1)
  y = rnorm(n2, delta)
  p = t.test(x, y)$p.value

  if(p <= alpha){
    count = count + 1
  }
}

power[i] = count / N
}

```

I have made the grid for δ very coarse because this is a lot of computation – there are 567,000 sets of samples taken in just this small run. What do we expect to see here? Power should increase as delta increases and sample size increases. You probably want to change the colours on this – I don't have enough time left in my life to know all of `ggplot2` (or care :-))

```
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

```
filter, lag
```

The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

```

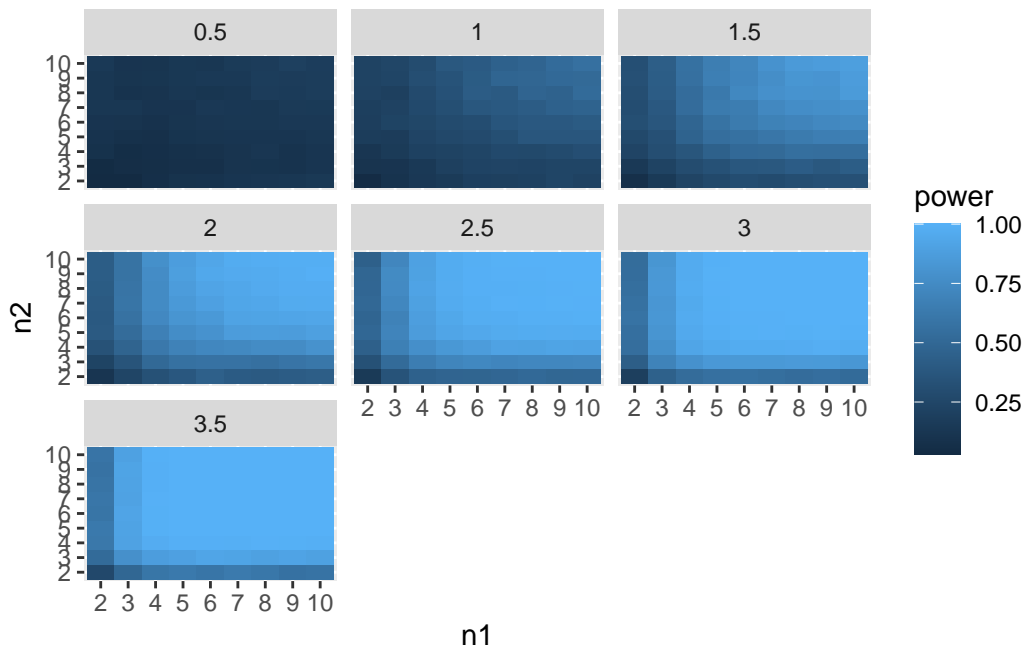
plot.df = sim.params |>
  mutate(
    n1 = factor(n1),
    n2 = factor(n2),
    delta = factor(delta),

```

```

    power = power
  )
g = plot.df |>
  ggplot(aes(x = n1, y = n2, fill = power)) +
  geom_tile() +
  facet_wrap(vars(delta), nrow = 3)
g

```



Multivariate

So I think this should be fairly obvious. Use the package `rmvtnorm`, and specifically `rmvtnorm` to generate your samples. You could repeat the experiments I have above using Hotelling's T^2 - but that isn't what we are trying to do. To do size, you should:

1. Compute the **average** the variance-covariance matrix from your database. That is - compute $\hat{\Sigma}$ for each source in the database, and then average all of those.
2. On each iteration
 - Generate two *new* samples using `rmvtnorm` with a mean vector of $\mathbf{0}$ and the variance-covariance matrix you calculated in the previous step
 - Run this through your grouping algorithm. If it says there is one groups, then do nothing. If there is a significant split, then count it.

One issue I think we will run into is how to deal with sample size.

To do the power experiments, use the square root of the diagonal of the variance-covariance matrix you calculated in Step 1 as your Δ (I'm using capitals because it is a vector). Add some fraction of Δ to the mean of the second group.