Power Analysis

Jasper Slingsby

Power Analysis

No one ever does them…

…but they could save so much pain and suffering if they did!!!

Power Analysis

Statistical power is the probability of a hypothesis test finding an effect if there is an effect to be found.

Power analysis is a calculation typically used to estimate the smallest sample size needed for an experiment, given a required significance level, statistical power, and effect size.

- It is *normally conducted before the data collection*!

Why do power analysis?

Firstly, it helps you plan your analyses before you’ve done your data collection, which is always useful, because it will inform how you collect your data.

Secondly, not knowing the statistical power of your analysis can result in:

- missed findings (through Type II Error), or
- false findings (through Type I Error).

Why do power analysis?

Type II Error:

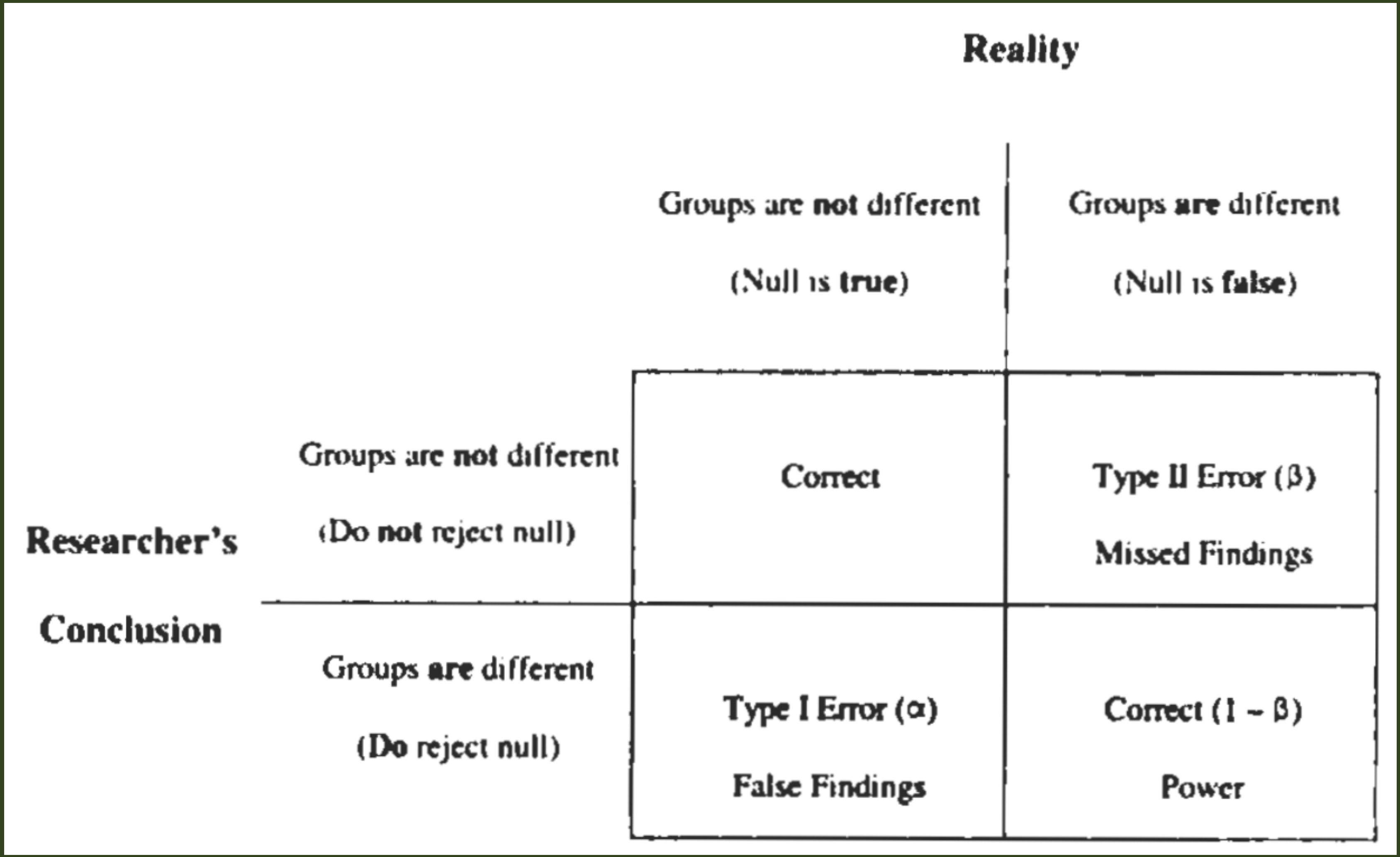
- occurs when the researcher erroneously concludes that there *is not* a difference between treatments, when in reality there is…
- this is a common outcome of low statistical power

Why do power analysis?

Type I Error:

- occurs when the researcher erroneously concludes that there *is* a difference between treatments, when in reality there is not…
- less likely when there is poor statistical power, but can happen with low sample sizes of highly variable subjects, or if there is bias in sampling…

Why do power analysis?



		Reality	
		Groups are not different (Null is true)	Groups are different (Null is false)
Researcher's Conclusion	Groups are not different (Do not reject null)	Correct	Type II Error (β) Missed Findings
	Groups are different (Do reject null)	Type I Error (α) False Findings	Correct (1 – β) Power

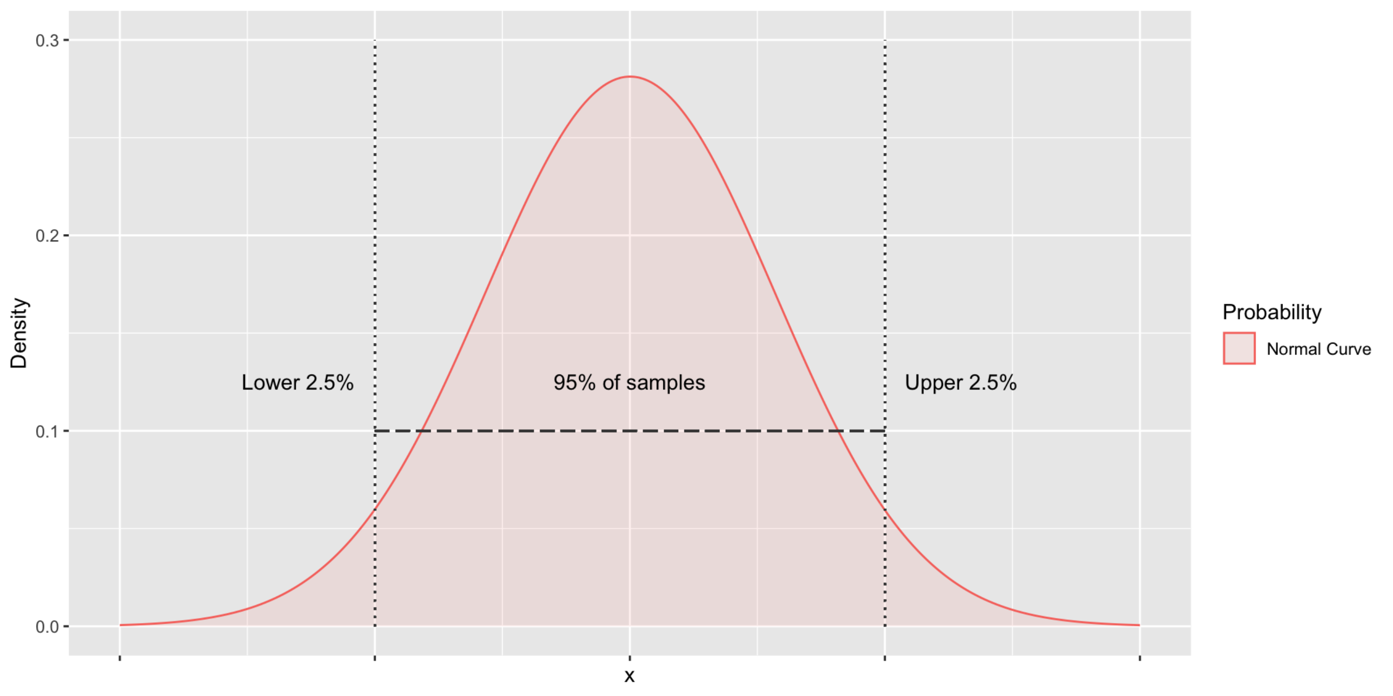
Type I and Type II Errors and how they result in false or missing findings, respectively. Image from Norton and Strube 2001, *JOSPT*.

Statistical Power

Is determined by the combination of the:

- α (“significance”) level required (e.g. $P < 0.05$)
- difference between group means (effect size)
- variability among subjects
- sample size (the factor we usually have most control over)

α (“significance”) level

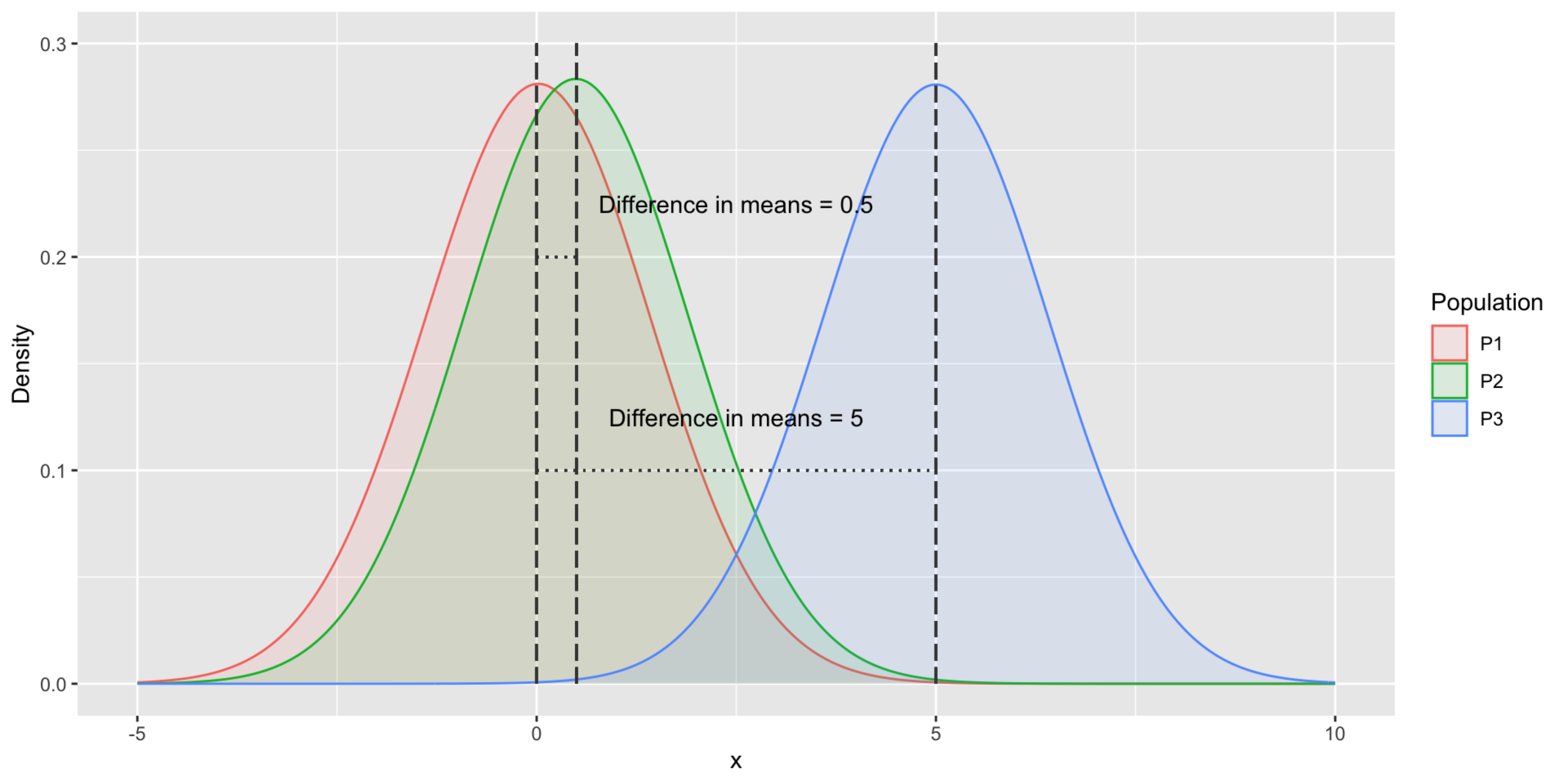


We usually use an α of 0.05 to indicate significant difference.

- i.e. the probability of the observation not being different to the null is less than 5% (i.e. $p < 0.05$), or the result should only be observed once or less for every 20 samples.

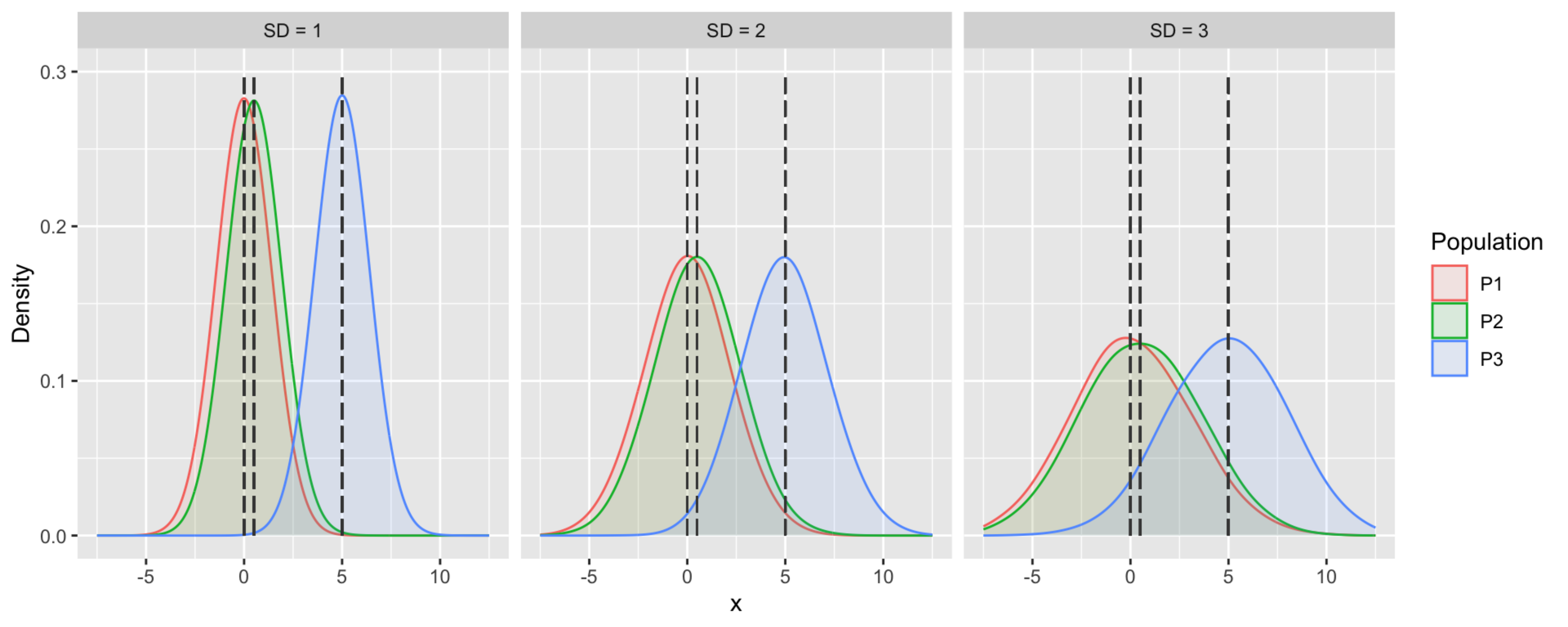
This is a subjective cut-off, but is generally accepted in the literature…

Difference between group means



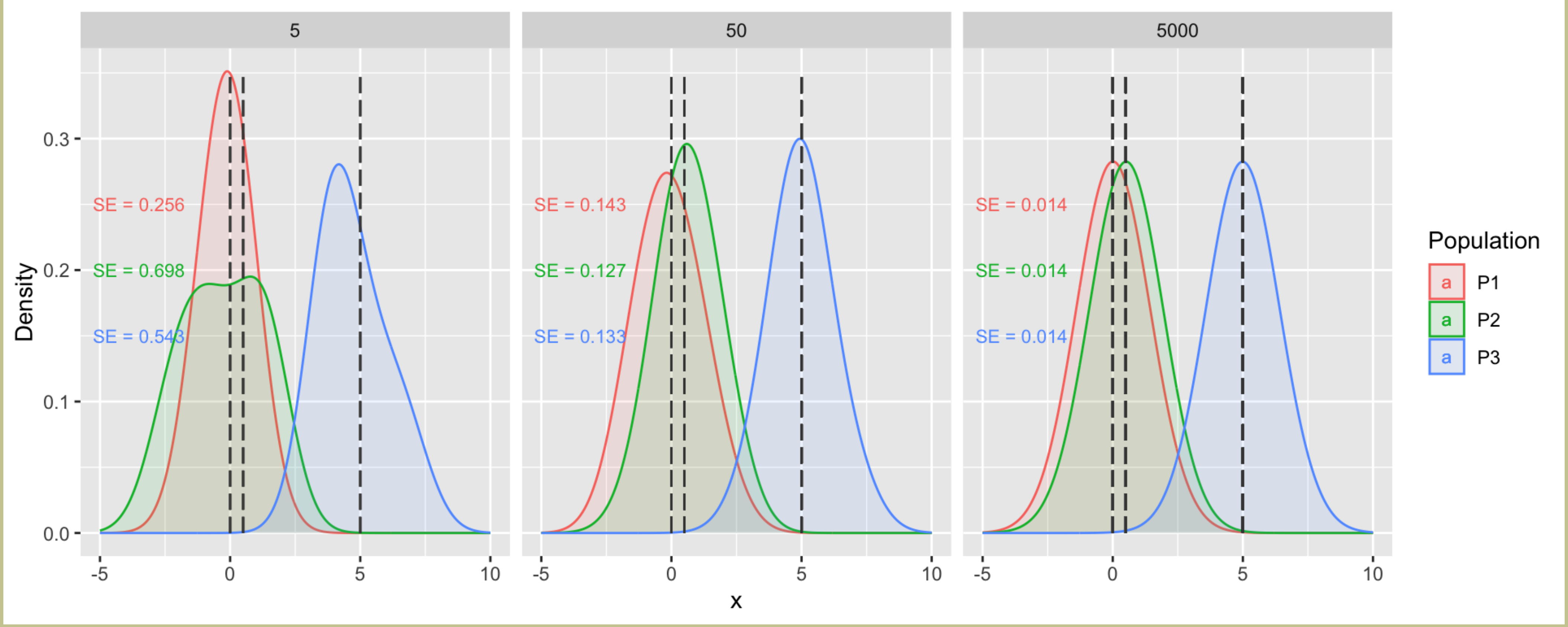
You have greater statistical power when you have greater differences in means (effect size). P1 vs P3 has greater power than either P1 vs P2 or P2 vs P3.

Variability among subjects



Greater variability among subjects results in larger standard deviations, reducing our ability to distinguish among groups (i.e. statistical power).

Sample size



Increasing sample size increases statistical power by improving the estimate of the mean and constricting the distribution of the test statistic (i.e. reducing the standard error (SE)). Dashed lines indicate the true population means.

How do we do power analysis?

Simulate the data you would expect to collect, varying the:

- difference between group means (effect size)
- variability among subjects
- sample size (the factor we usually have most control over)

…and test for significant difference using the appropriate statistical test (possibly varying the α (“significance”) level (e.g. $P < 0.05$) if justified).

Simulating data

First, we need to simulate some data.

If we believe our data are normally distributed, we can use the handy `rnorm()` function, like so:

```r
dat <- rnorm(n = 50, # set the sample size
             mean = 1, # set the mean to 1
             sd = 1) # set the standard deviation to 1
```

Type `?Distributions` into your R console for alternative distributions.

Simulating data

Now let’s look at our new data

- This is easier if we make it a data frame

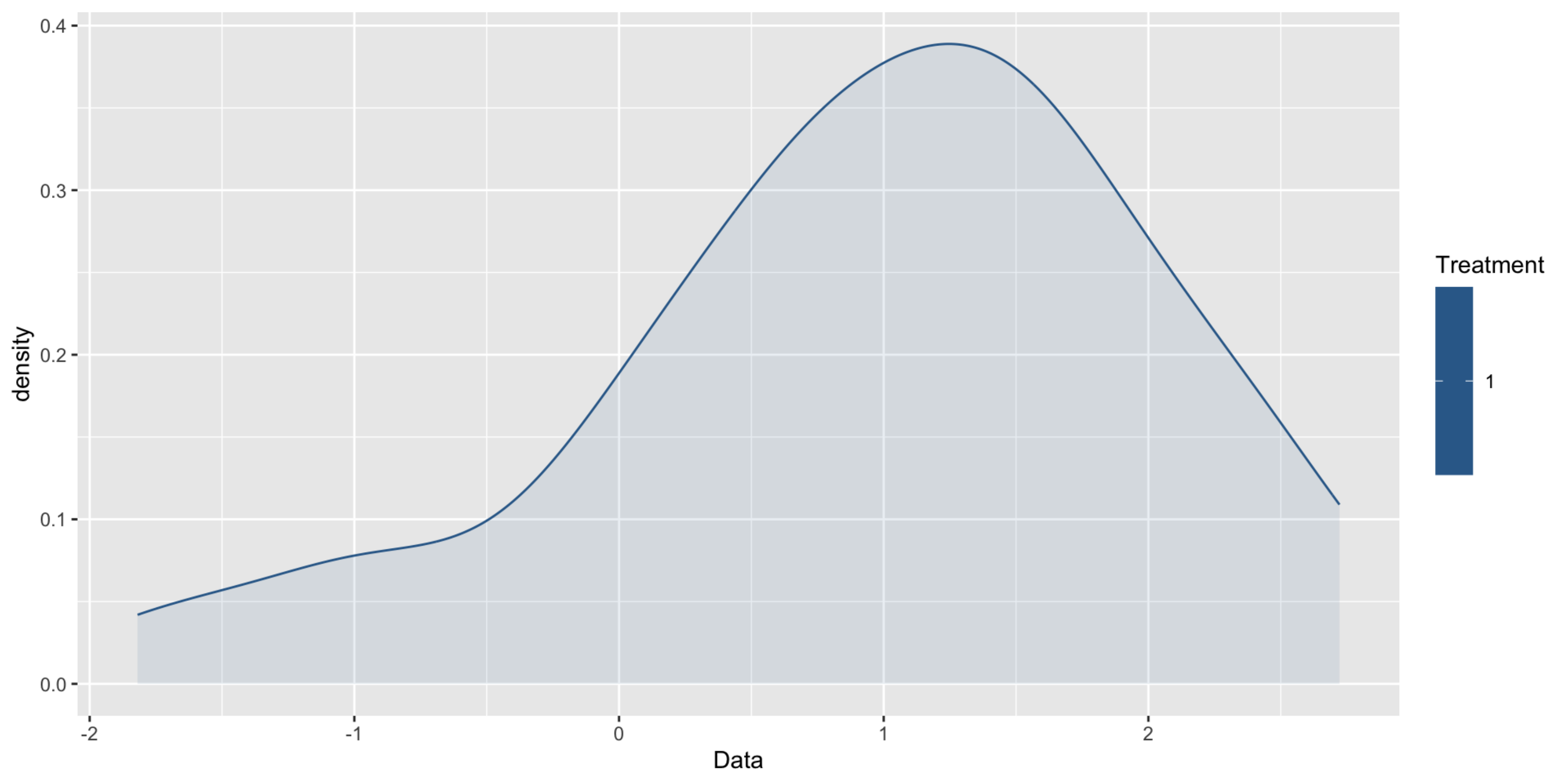
```r
df <- data.frame(Data = dat, Treatment = 1)

head(df)
```

```
       Data Treatment
1 0.7057159         1
2 0.4368053         1
3 1.3776016         1
4 0.8569171         1
5 1.2235321         1
6 1.4042901         1
```

Simulating data

We can plot it like so:

```r
ggplot(df, aes(Data, fill = Treatment, colour = Treatment)) +
  geom_density(alpha = 0.1)
```



One sample *t*-test

Tests the hypothesis that the mean of our population is a specific value (e.g. 0).

```r
t.test(x = df$Data, # set our vector of data values
       alternative = "two.sided", # specify the alternative hypothesis (which in this case is "not zero" so
       mu = 0) # set the "true value" of the mean
```

```

    One Sample t-test

data:  df$Data
t = 6.4457, df = 49, p-value = 4.793e-08
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.6609861 1.2598437
sample estimates:
mean of x
0.9604149
```

In this case, the difference is highly significant! P < 0.00000005!!!

One sample *t*-test

What if we fiddle with the α ("significance" level)?

- You usually wouldn't do this!!!

but

- With one-sample *t*-tests one effectively does when choosing your alternative hypothesis.
    - We set it to be "two-sided" because our alternative was that the mean is "not zero". This means the result is only significantly different if the observed mean is in the upper or lower 2.5% of the distribution.
- If our alternative hypothesis was that the observed mean was "greater" or "less" then we could set that and the result would only be significantly different if the observed mean is in either the upper or lower 5% of the distribution respectively.
    - i.e. setting the alternative to "greater" or "less" effectively makes the test more sensitive, similar to increasing the α

One sample *t*-test

Now let’s fiddle with the difference between ***group means (effect size)***.

In this case this is easiest done by shifting the *mu* to closer to the mean of our randomly generated data, like so

```r
t.test(x = df$Data, # set our vector of data values
       alternative = "two.sided", # specify the alternative hypothesis
       mu = 0.5) # set the "true value" of the mean
```

```

	One Sample t-test

data:  df$Data
t = 3.09, df = 49, p-value = 0.003295
alternative hypothesis: true mean is not equal to 0.5
95 percent confidence interval:
 0.6609861 1.2598437
sample estimates:
mean of x 
0.9604149 
```

Here we’ve reduced the effect size to from 1 to 0.5, but the result is still significantly different.

One sample *t*-test

Now let’s fiddle with ***variability among subjects***.

```r
# Make new data with greater variability (standard deviation = 2)
df <- data.frame(Data =
                  rnorm(n = 50, # set the sample size
                        mean = 1, # set the mean
                        sd = 2), # set bigger standard deviation
                 Treatment = 1)

# Run t-test
t.test(x = df$Data,
       alternative = "two.sided",
       mu = 0.5)
```

```

    One Sample t-test

data:  df$Data
t = 0.92312, df = 49, p-value = 0.3605
alternative hypothesis: true mean is not equal to 0.5
95 percent confidence interval:
 0.1965978 1.3189782
sample estimates:
mean of x
 0.757788
```

With double the variability (standard deviation), and an effect size of 0.5, the result is no longer significantly different…

One sample *t*-test

Now let’s increase the ***sample size***.

```r
# Make new data with greater sample size (n = 100)
df <- data.frame(Data =
                  rnorm(n = 100, # set the sample size
                        mean = 1, # set the mean
                        sd = 2), # set bigger standard deviation
                 Treatment = 1)

# Run t-test
t.test(x = df$Data,
       alternative = "two.sided",
       mu = 0.5)
```

```
    One Sample t-test

data:  df$Data
t = 3.8761, df = 99, p-value = 0.000191
alternative hypothesis: true mean is not equal to 0.5
95 percent confidence interval:
 0.8571468 1.6063205
sample estimates:
mean of x
 1.231734
```

Aha! By doubling our sample size, our result is significantly different once again…

Estimating the number of samples

Repeatedly rerunning our simulation with different sample size (***n***) would rapidly become tedious…

Fortunately, there’s a better way (for common tests…)!

library(pwr) allows us to input the effect size and power required and returns the required ***n***.

Estimating the number of samples

One sample *t*-test

```r
library(pwr)

pwr.t.test(d = 0.8,
           n = NULL,
           sig.level = 0.05,
           power = 0.8,
           type = "one.sample",
           alternative = "two.sided")
```

```

     One-sample t test power calculation 

              n = 14.30276
              d = 0.8
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

Which suggests we need 15 samples to achieve our desired statistical power.

Here we have set ***n*** = *NULL* because that's the property we want to estimate.

power = the power of the test (1 minus the Type II error probability), which in this case we have set to 80%

d = Cohen's *d* = a measure of effect size = the difference between the means divided by the pooled standard deviation

- i.e. you input the effect size and variability in one go

Estimating the number of samples

Cohen’s *d*

= the difference between the means divided by the pooled standard deviation (i.e. the standard deviation of the difference)

So, for our simulated data, assuming we’re comparing our data to 0:

```r
mean(df$Data)
```

```
[1] 1.231734
```

```r
sd(df$Data)
```

```
[1] 1.887832
```

```r
mean(df$Data)/sd(df$Data)
```

```
[1] 0.6524593
```

Estimating the number of samples

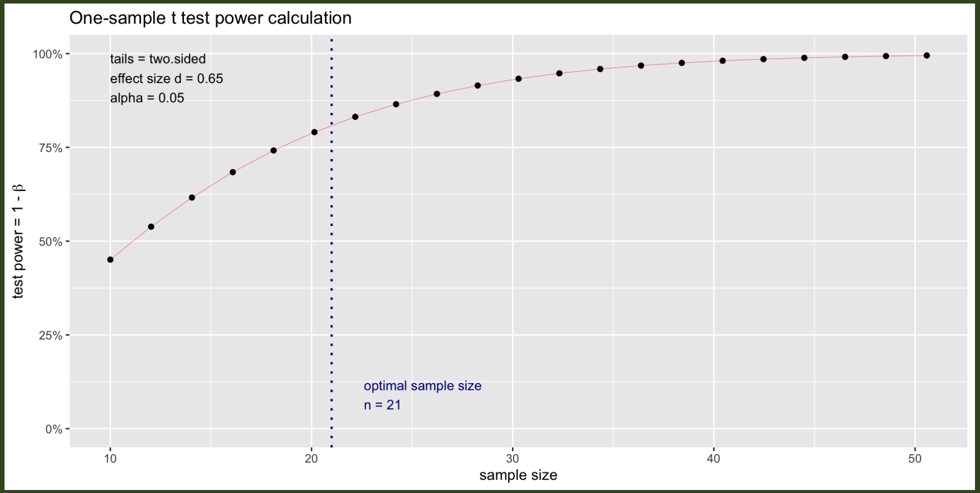
Rerun for our observed ***d***:

```
pwr.t.test(d = 0.65,
           n = NULL,
           sig.level = 0.05,
           power = 0.8,
           type = "one.sample",
           alternative = "two.sided")
```

```
     One-sample t test power calculation

              n = 20.58039
              d = 0.65
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

And plotted:

```
pwr.t.test(d = 0.65,
           n = NULL,
           sig.level = 0.05,
           power = 0.8,
           type = "one.sample",
           alternative = "two.sided") %>%
  plot()
```



library(pwr) functions:

- `pwr.p.test`: one-sample proportion test
- `pwr.2p.test`: two-sample proportion test
- `pwr.2p2n.test`: two-sample proportion test (unequal sample sizes)
- `pwr.t.test`: two-sample, one-sample and paired t-tests
- `pwr.t2n.test`: two-sample t-tests (unequal sample sizes)
- `pwr.anova.test`: one-way balanced ANOVA
- `pwr.r.test`: correlation test
- `pwr.chisq.test`: chi-squared test (goodness of fit and association)
- `pwr.f2.test`: test for the general linear model

library(pwr) functions:

A note on effect sizes…

Each test has a different metric of effect size, each calculated in a different way…

Fortunately, *library(pwr)* has a convenient function (`cohen.ES`) that can provide these for you for small, medium and large effect sizes.

```r
cohen.ES(test = "t", size = "medium")
```

```

     Conventional effect size from Cohen (1982) 

           test = t
           size = medium
    effect.size = 0.5
```

If you’re not sure of the effect size you’d expect, the conservative approach is to use “small”

library(pwr) functions:

A note on effect sizes…

You can pass the results of `cohen.ES` directly to the pwr function by calling the `effect.size` slot in the returned object:

```r
str(cohen.ES(test = "t", size = "medium"))
```

```
List of 4
 $ test       : chr "t"
 $ size       : chr "medium"
 $ effect.size: num 0.5
 $ method     : chr "Conventional effect size from Cohen (1982)"
 - attr(*, "class")= chr "power.htest"
```

```r
cohen.ES(test = "t", size = "medium")$effect.size
```

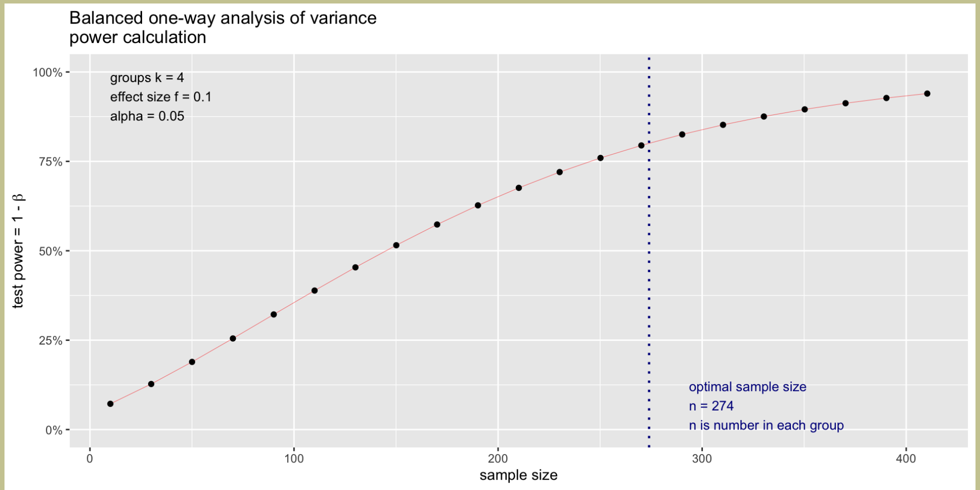
```
[1] 0.5
```

library(pwr) functions: ANOVA

For comparisons among 3 or more groups (k) for different effect sizes (f)

Small effect size

```r
pwr.anova.test(f = cohen.ES(test = "anov",
                            size = "small")$effect,
               k = 4,
               power = 0.80,
               sig.level = 0.05) %>%
  plot
```

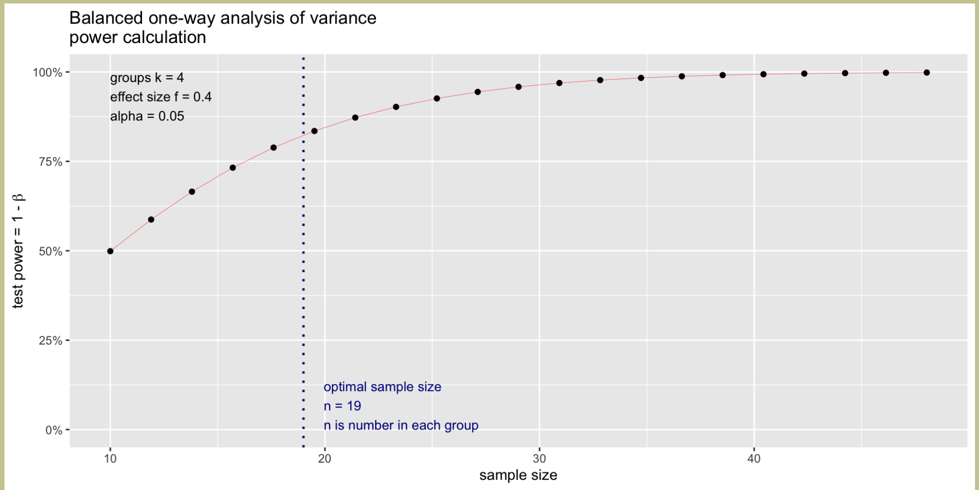

Balanced one-way analysis of variance
power calculation

groups k = 4
effect size f = 0.1
alpha = 0.05

optimal sample size
n = 274
n is number in each group

Large effect size

```r
pwr.anova.test(f = cohen.ES(test = "anov",
                            size = "large")$effect,
               k = 4,
               power = 0.80,
               sig.level = 0.05) %>%
  plot
```


Balanced one-way analysis of variance
power calculation

groups k = 4
effect size f = 0.4
alpha = 0.05

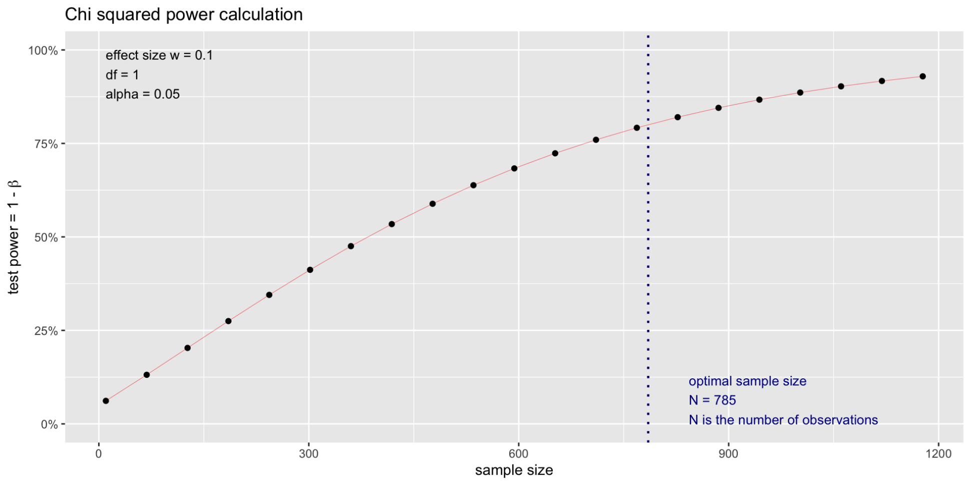
optimal sample size
n = 19
n is number in each group

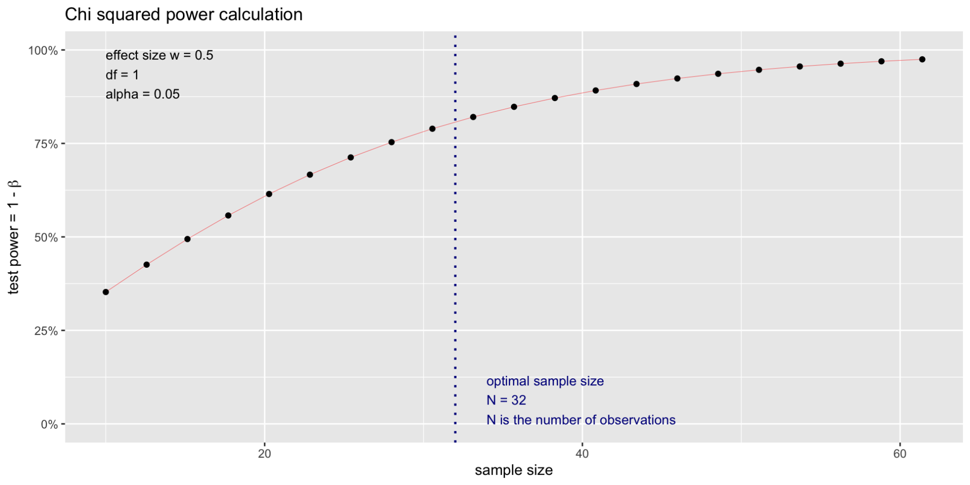
library(pwr) functions: Chi-squared

Tests whether 2 categorical variables (dimensions of a contingency table) are independent

```r
pwr.chisq.test(w = cohen.ES(test = "chisq",
                            size = "small")$effect,
               df = 1,
               power = 0.80,
               sig.level = 0.05) %>%
  plot()
```

```r
pwr.chisq.test(w = cohen.ES(test = "chisq",
                            size = "large")$effect,
               df = 1,
               power = 0.80,
               sig.level = 0.05) %>%
  plot()
```





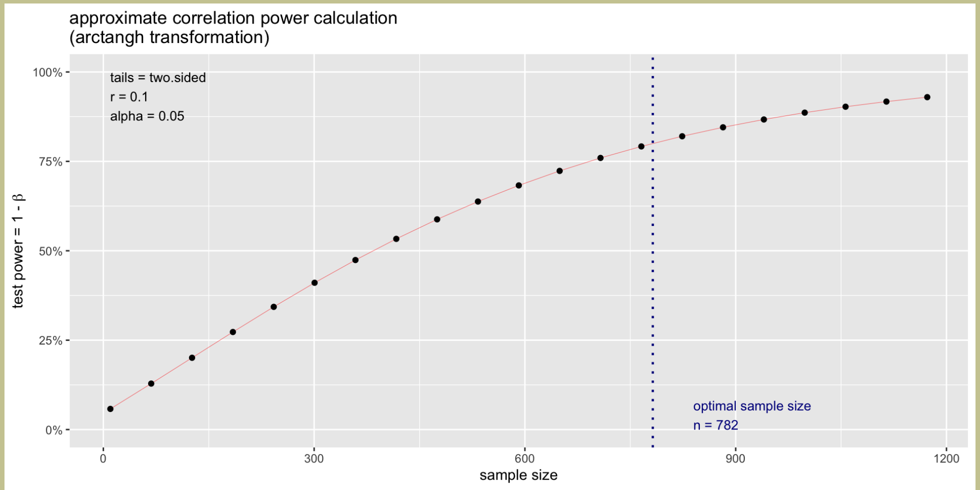
w = Effect size, df = Degrees of freedom

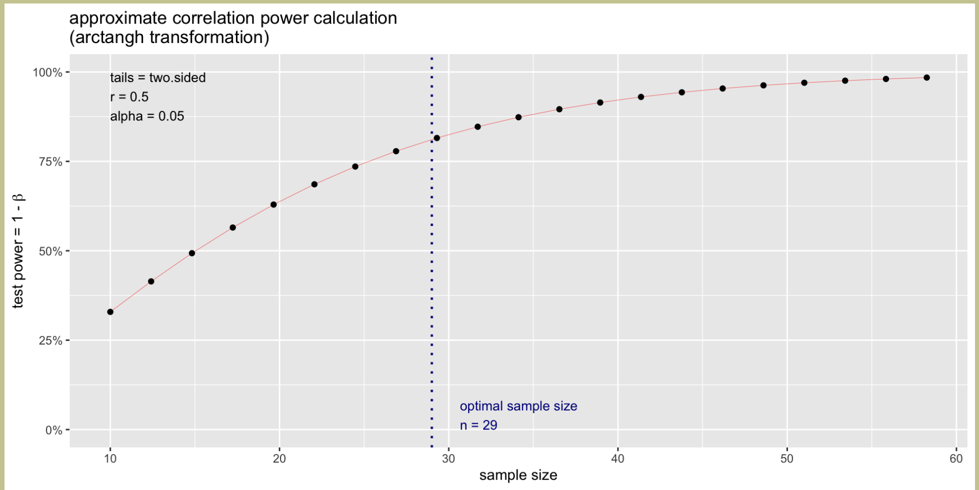
library(pwr) functions: Correlation

r = correlation coefficient (i.e. it is not R^2)

```r
pwr.r.test(r = cohen.ES(test = "r",
                        size = "small")$effect.siz
           power = 0.80,
           sig.level = 0.05,
           alternative = "two.sided") %>%
  plot()
```

```r
pwr.r.test(r = cohen.ES(test = "r",
                        size = "large")$effect.siz
           power = 0.80,
           sig.level = 0.05,
           alternative = "two.sided") %>%
  plot()
```





Note that for regression analysis you'd need to set the "alternative" to "greater" or "less", because it assumes that one variable is dependent on the other

library(pwr) functions: general linear model (i.e. multiple regression)

For regression analysis with multiple covariates (explanatory variables)

```r
pwr.f2.test(u = 1,
            v = NULL,
            f2 = cohen.ES(test = "f2", size = "sma
            power = 0.8,
            sig.level = 0.05)
```

```

     Multiple regression power calculation 

              u = 1
              v = 392.373
             f2 = 0.02
      sig.level = 0.05
          power = 0.8
```

```r
pwr.f2.test(u = 1,
            v = NULL,
            f2 = cohen.ES(test = "f2", size = "lar
            power = 0.8,
            sig.level = 0.05)
```

```

     Multiple regression power calculation 

              u = 1
              v = 22.50313
             f2 = 0.35
      sig.level = 0.05
          power = 0.8
```

f2 = Effect size

u = degrees of freedom for numerator (= number of groups - 1)

v = degrees of freedom for denominator (= total number of individuals across groups - the number of groups)

So…

1. Think carefully about the data you plan to collect and how to analyze them
2. Decide on your statistical analyses/tests
3. Do power analyses for each of your analyses/tests to determine necessary sample sizes
4. Include a description of your intended analyses and (ideally) present power analysis in your presentations on Thursday (not for marks, but very useful)
5. Include a power analysis in your write-ups!!! (for marks!)

Further resources

- library(simr) for generalised linear mixed effects models (GLMM), e.g. this demo.
- A blog with examples and R code
- Another blog/webpage with examples and R code