جامعـة نيويورك أبـوظـبي

NYU | ABU DHABI

PSYCH-UH 1004Q: Statistics for Psychology

Class 14: Effect sizes and statistical power (part 2 - practical)

Prof. Jon Sprouse
Psychology

A brief review

Statistical Significance vs Practical Significance

Scientists make a distinction between two types of “significance”:

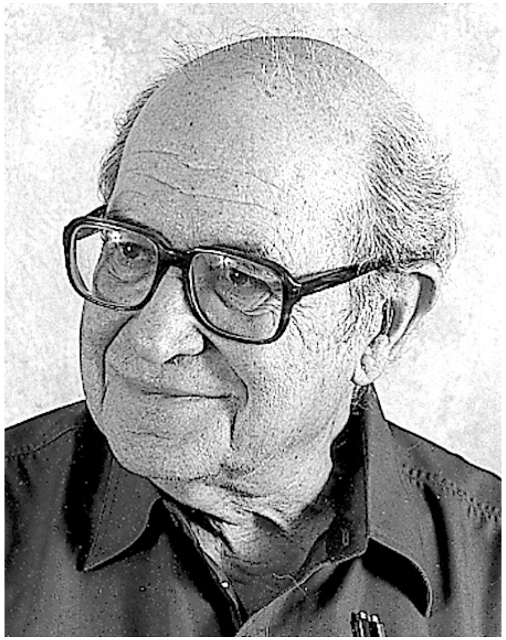
Statistical significance: Achieving a *p*-value below a critical alpha level. This tells us that the effect that we detected would be relatively rare if the null hypothesis is true.

Practical significance: Achieving an effect size that is in line with our theory. This could be the size directly predicted by our theory, or it could a size that has meaningful consequences for our decisions about what to do if our theory is correct.

As scientists, we want **both**. We want to demonstrate that our effect is rare if the null hypothesis is true, and we want to demonstrate that the effect size is in line with our theory (that it is practically significant). We have learned a lot about statistical significance so far. Today we will see how to make the idea of effect size mathematically rigorous… and how to quantify our ability to detect effects of certain sizes through a measure called statistical power.

Standardized effect size is called Cohen’s d

Jacob Cohen is another famous statistician, this time from the second half of the 20th century. He got his PhD at NYU and worked there his entire career!



1923-1998

Cohen’s d is a population parameter. It describes the size of the difference between two population means in terms of the standard deviation of the populations. It typically assumes that the two populations have the same standard deviation:

$$d = \frac{\mu_1 - \mu_2}{\sigma}$$

It is a rare case of a population parameter that uses the Latin alphabet (not Greek).

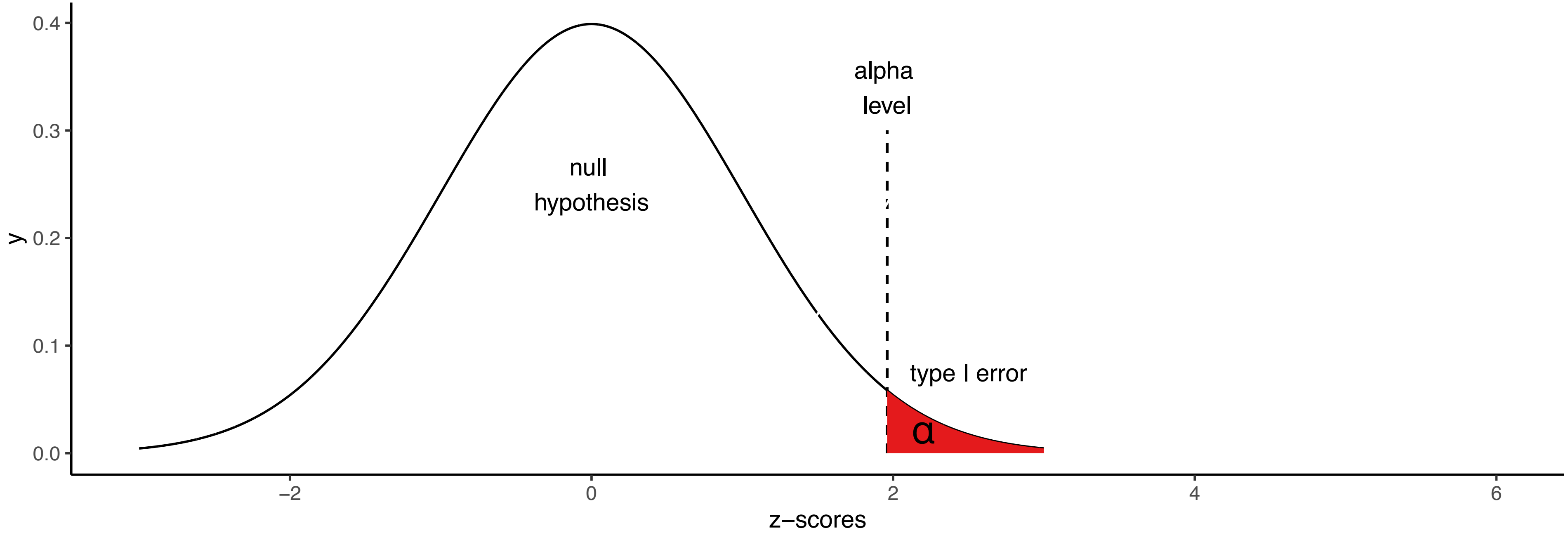
In practice, you may not know the population parameters to put into the equation. In that case, you can estimate it from your sample statistics, using the square root of the pooled variance. Our textbook recommends calling this *g*, but sometimes people still call it d or estimated d.

$$g = \frac{\bar{x} - \bar{x}}{s_p}$$

H_0 is…	True	False
Rejected	Type I error (false positive) probability = α	correct decision (true positive) probability = 1-β
Not Rejected	correct decision (true negative) probability = 1-α	Type II error (false negative) probability = β

If your test statistic falls on the left of the alpha criterion, you'll fail to reject H_0, which is a **correct decision**.

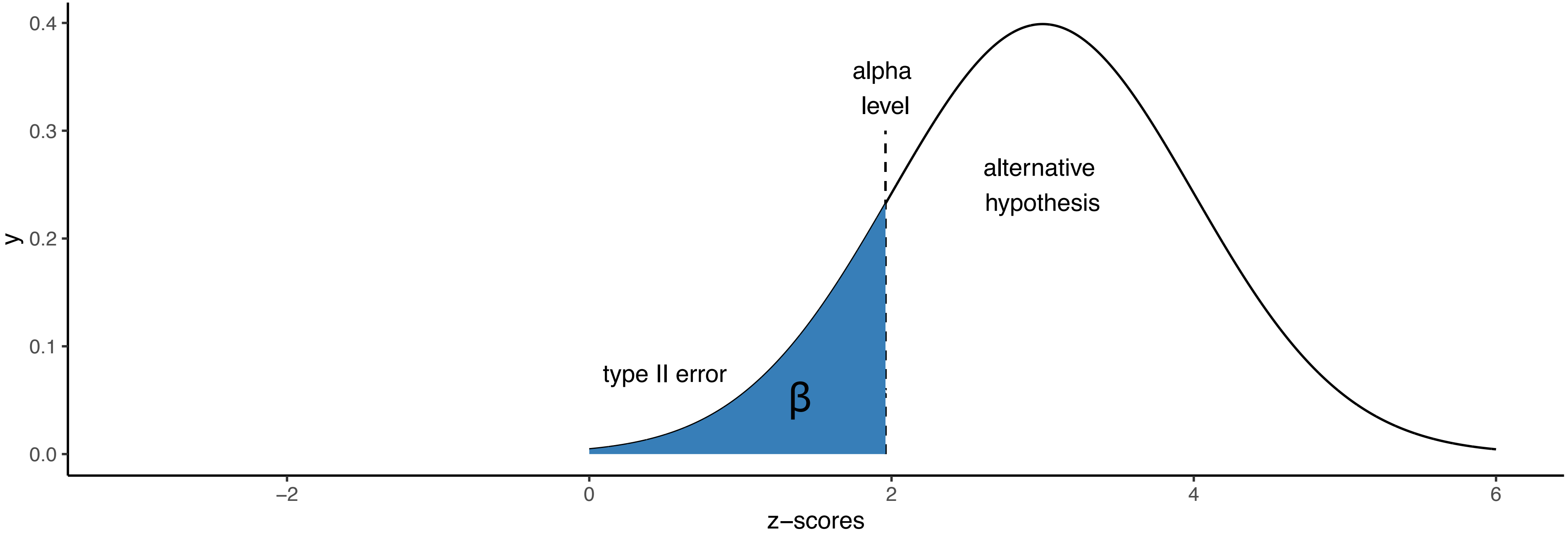
If your test statistic falls on the right of the alpha criterion, you'll reject H_0, which is a type I error (false positive).



H_0 is…	True	False
Rejected	Type I error (false positive) probability = α	correct decision (true positive) probability = 1-β
Not Rejected	correct decision (true negative) probability = 1-α	Type II error (false negative) probability = β

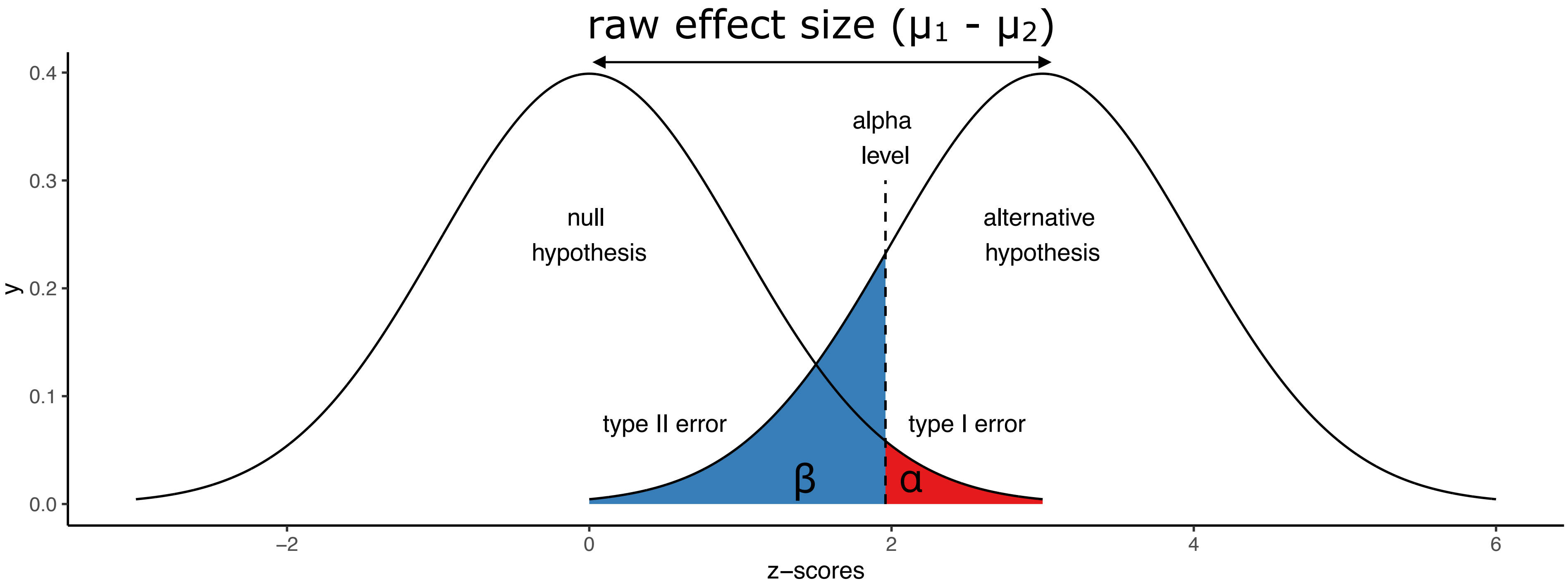
If your test statistic falls on the left of the alpha criterion, you'll fail to reject H_0, which is a type II error (false negative).

If your test statistic falls on the right of the alpha criterion, you'll reject H_0, which is a **correct decision**.



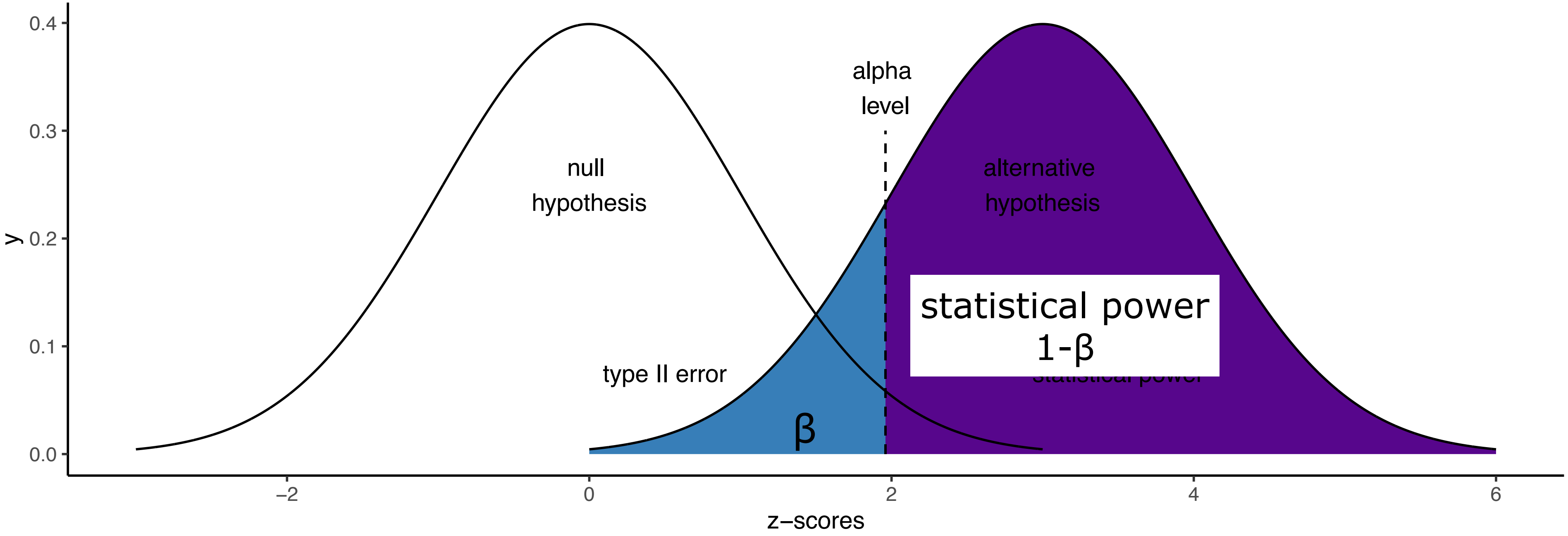
H_0 is…	True	False
Rejected	Type I error (false positive) probability = α	correct decision (true positive) probability = 1-β
Not Rejected	correct decision (true negative) probability = 1-α	Type II error (false negative) probability = β

Remember that we never know whether H_0 is true or false. So we have to imagine **both possibilities**. We will **make a decision**. That decision will be one of the four outcomes:



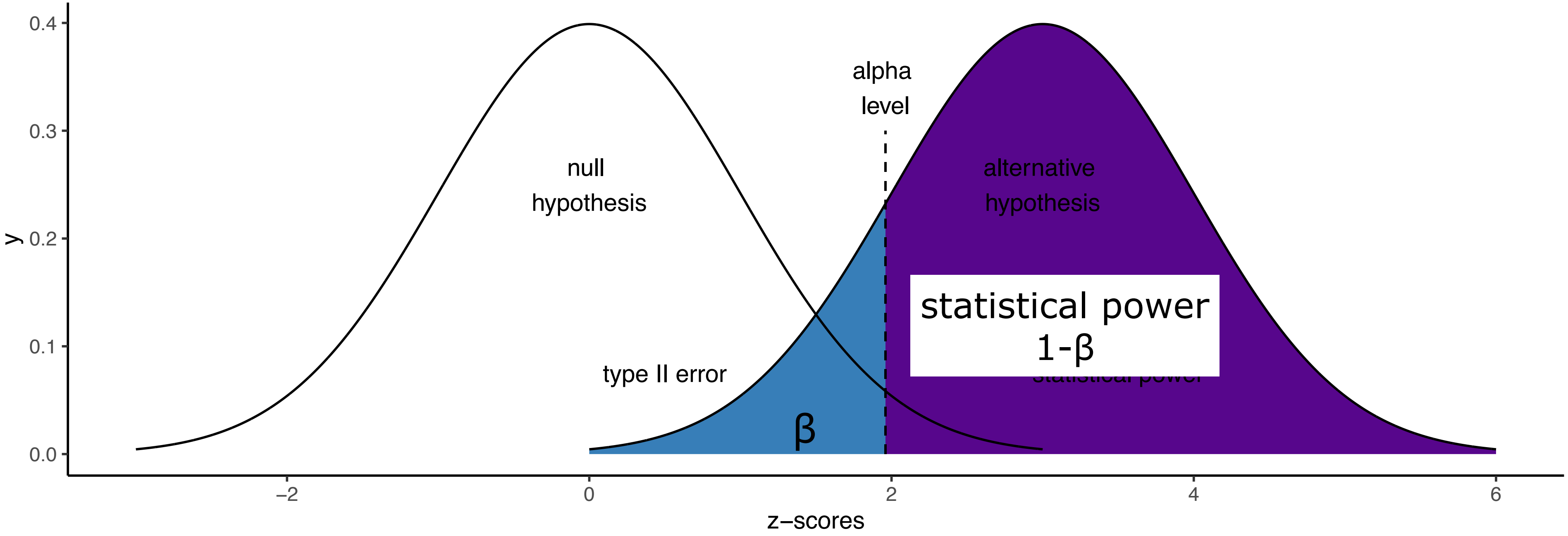
H_0 is…	True	False
Rejected	Type I error (false positive) probability = α	correct decision (true positive) probability = 1-β
Not Rejected	correct decision (true negative) probability = 1-α	Type II error (false negative) probability = β

In precise NHT terms, statistical power is the probability of rejecting the null hypothesis when the null hypothesis is false. It is when you make a correct decision, but could have made a Type II error.



H_0 is…	True	False
Rejected	Type I error (false positive) probability = α	correct decision (true positive) probability = 1-β
Not Rejected	correct decision (true negative) probability = 1-α	Type II error (false negative) probability = β

The rule of thumb (so, a suggestion) from Cohen is that we should have statistical power of .8, that means an 80% chance of detecting the effect. He chose this because he said that Type I errors (α =.05) are 4 times worse than Type II error, so β should equal .2, which yields power (1-β) of .8.

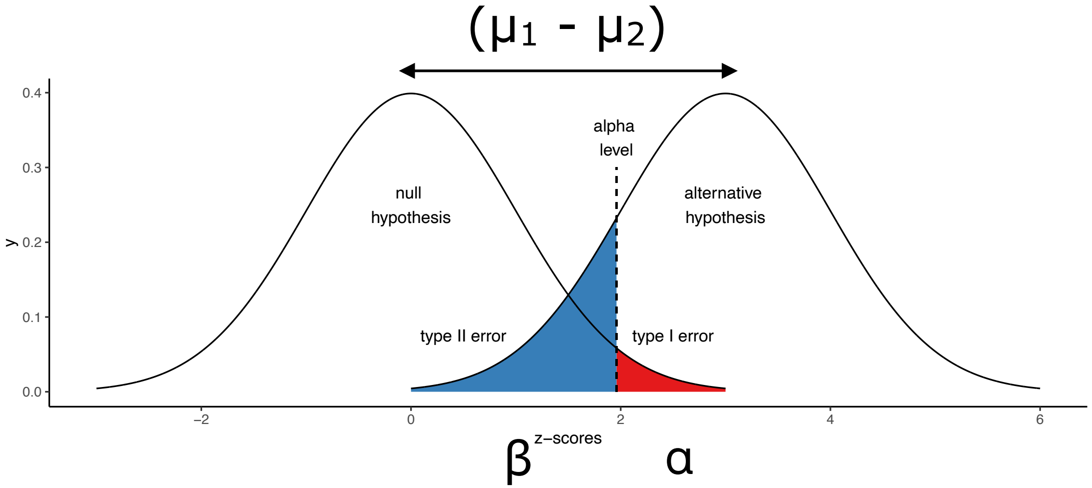


Which properties matter?

The relationships here are complex. But we can identify 4 properties that dictate the relationship

1. **The choice of alpha (α)**. This will move the ratio of Type I and Type II errors.

2. **The raw effect size (μ_1 - μ_2).** This will move the two distributions away from each other.



3. **The standard deviation (variability)** in the distributions. This will create less (or more) overlap between the two distributions.

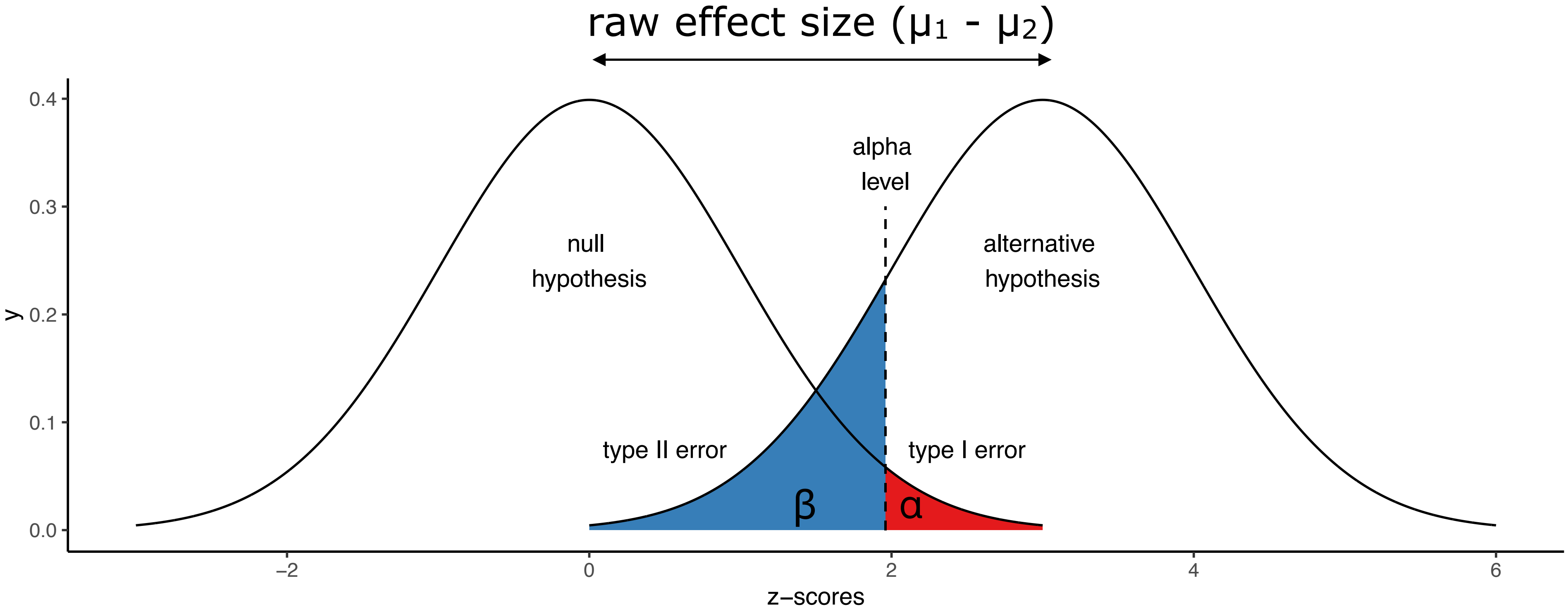
4. **The sample size (n)**. This will also change the width of the distributions. This is because they are sampling distributions of the mean, so their standard deviation is the standard error. As the sample size increases, the standard error decreases!

$$s_{\bar{x}} = \frac{s}{\sqrt{n}}$$

An interactive demonstration

Here is an interactive demonstration of this same plot, so we can see the consequences of the four properties that dictate the error rates.

https://rpsychologist.com/d3/nhst/



You should take time to really play with this to build an intuition about how these values are related.

What can we change in practice?

There really is only one of these that we can change in practice:

1. **The choice of alpha (α)**. This will move the ratio of Type I and Type II errors.

← This is set by the field! We can’t change it in practice.

2. **The raw effect size (μ_1 - μ_2)**. This will move the two distributions away from each other.

← This is an inherent property of the phenomenon you are studying. The universe sets this!

3. **The standard deviation** in the distributions. This will create less (or more) overlap between the two distributions.

← This is an inherent property of your measurement. The universe sets this!

4. **The sample size (n)**. This will also change the width of the distributions.

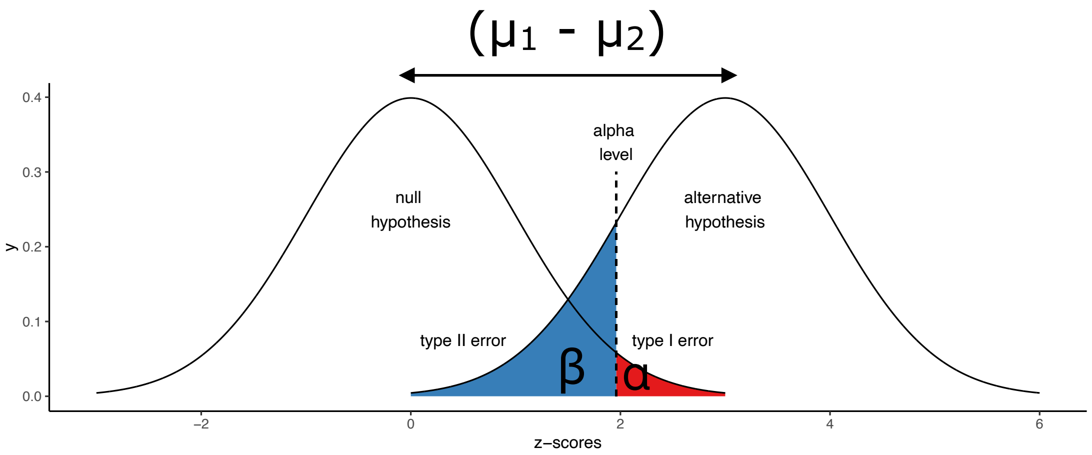
← You choose this!

Calculating the sample size that you need for good statistical power (typically .8)

Power is a complex function

As we have seen, statistical power is a function that takes 4 arguments:

1. **The choice of alpha (α)**.
2. **The raw effect size (μ_1 - μ_2).**
3. **The standard deviation.**
4. **The sample size (n)**.



Therefore we need a complex function to compute it. R gives us those functions. For t-tests, that function is power.t.test(). If you look at the help file for this function, you will see this. Let's unpack it!

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE, tol = .Machine$double.eps^0.25)
```

The logic of the function

The way the power.t.test() function works is that it has 5 arguments: the 4 that determine power and an argument for power itself. You fill in 4 of them, and leave one of those 5 empty (NULL), and it calculates the one that you left null. It is really very flexible that way!

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE, tol = .Machine$double.eps^0.25)
```

n is the sample size. This is what we usually want to calculate, so we usually leave it null.

delta is the difference between means.

sd is the standard deviation. It has a default value of 1 (we will see soon that this is because it assumes you want to enter Cohen's d into delta!).

sig.level is the alpha criterion. You will keep this at 0.05, which is the default.

power is the power level. You should set it 0.8, but you can choose others.

The logic of the function

The only other thing to remember is that there are different kinds of t-tests. So you have to tell this function what kind of t-test you want to run.

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE, tol = .Machine$double.eps^0.25)
```

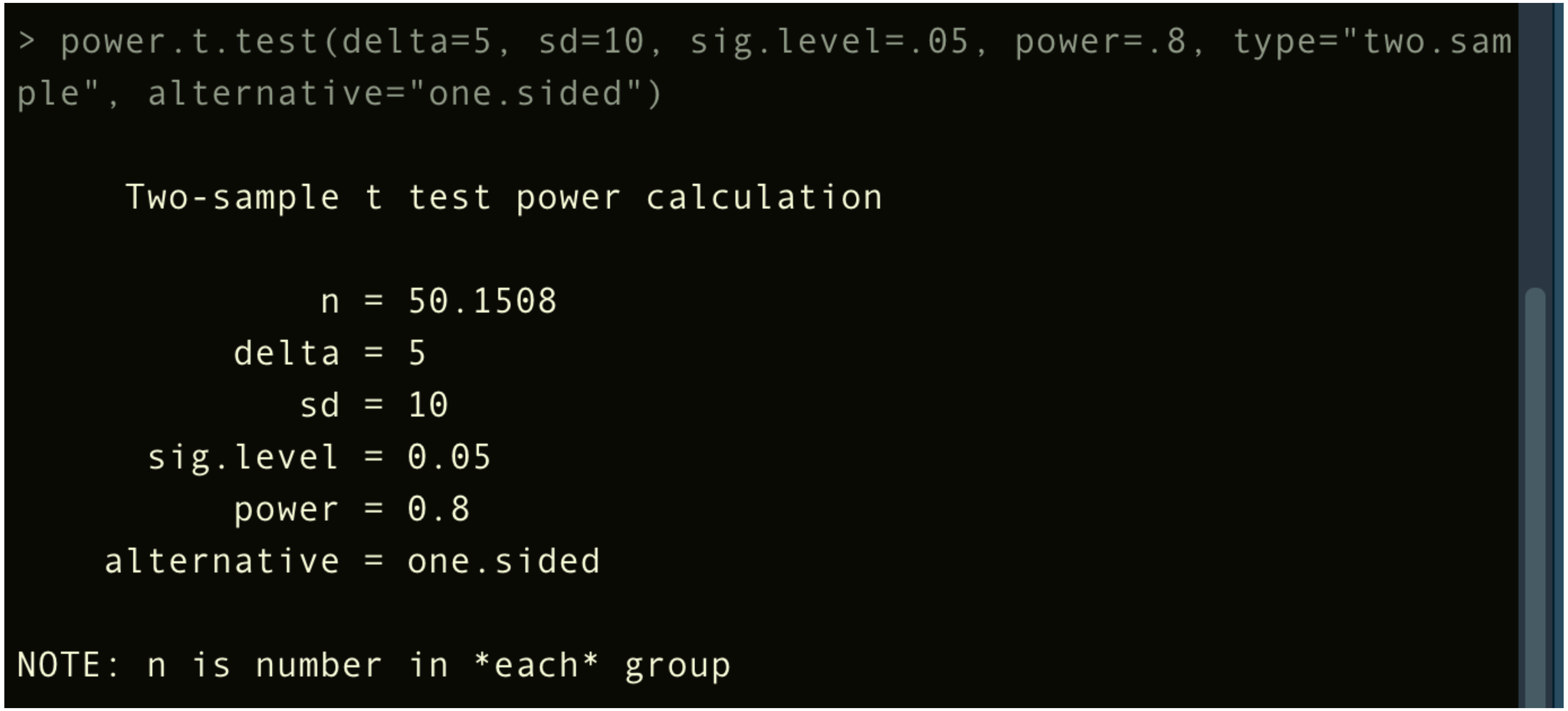
The type argument tells it if you are running a two-sample test (independent samples), a one-sample test, or a paired test (we haven’t seen that one yet!).

The alternative argument tells R if you are running a two-tailed (“two.sided”) or one-tailed (“one.sided”) test.

Let’s try it

Let’s say we are working with our height data set. We want to test for a difference between our sample means of 5cm, a standard deviation of 10cm, and we want power of .8 with an alpha of .05. We want to do this with two samples (an independent samples t-test) and we have a one-tailed hypothesis. How many observations/participants do we need to collect in our samples?

To calculate this, all we need to do is fill in the correct options for each of the arguments, and leave the n argument NULL so that it will calculate n.

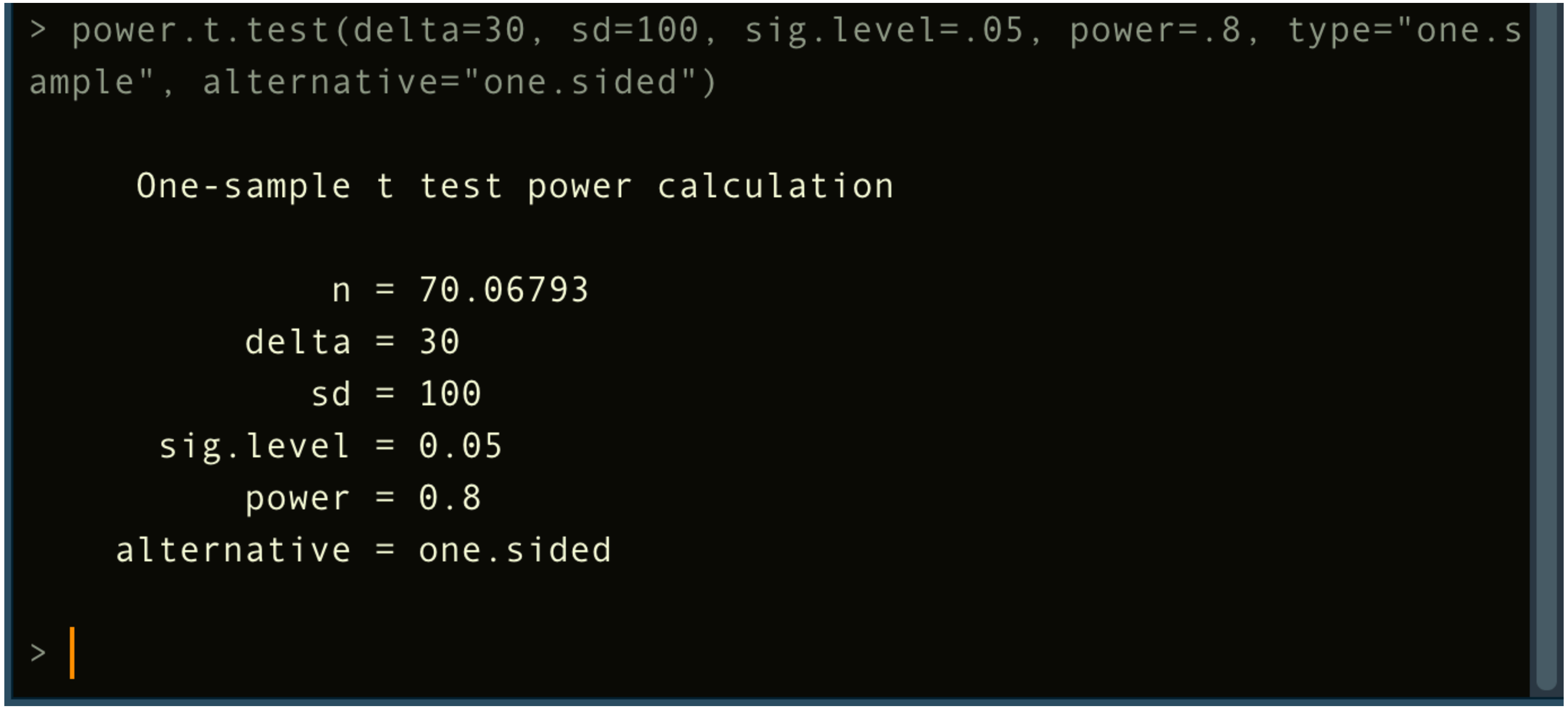


```
> power.t.test(delta=5, sd=10, sig.level=.05, power=.8, type="two.sam
ple", alternative="one.sided")

     Two-sample t test power calculation 

              n = 50.1508
          delta = 5
             sd = 10
      sig.level = 0.05
          power = 0.8
    alternative = one.sided

NOTE: n is number in *each* group
```

Let’s try it again.

Let’s say we are working with a new data set. We want to test for a difference between our means of 30, a standard deviation of 100, and we want power of .8 with an alpha of .05. We want to do this with one sample, and we have a one-tailed hypothesis. How many observations/participants do we need to collect in our sample?

To calculate this, all we need to do is fill in the correct options for each of the arguments, and leave the n argument NULL so that it will calculate n.



```
> power.t.test(delta=30, sd=100, sig.level=.05, power=.8, type="one.s
ample", alternative="one.sided")

     One-sample t test power calculation 

              n = 70.06793
          delta = 30
             sd = 100
      sig.level = 0.05
          power = 0.8
    alternative = one.sided

> 
```

What if we only know Cohen’s d?

The power.t.test() function asks for the raw difference between means in the delta argument and standard deviation in the sd argument. But what if you don’t know the raw difference between means or the standard deviation, but instead know Cohen’s d?

Don’t worry. You can still use the power.t.test() function. All you need to do is remember that Cohen’s d is the difference between means divided by the standard deviation. So we have all the information we need. We just need to figure out how to enter it into the function.

$$d = \frac{\mu_1 - \mu_2}{\sigma}$$

Cohen’s d is usually given as a single number, like 0.5. But we can view this as fraction set over 1, like this.

$$d = 0.5 =$$

And now we can set this equal to the typical equation to see both a difference between conditions and an sd:

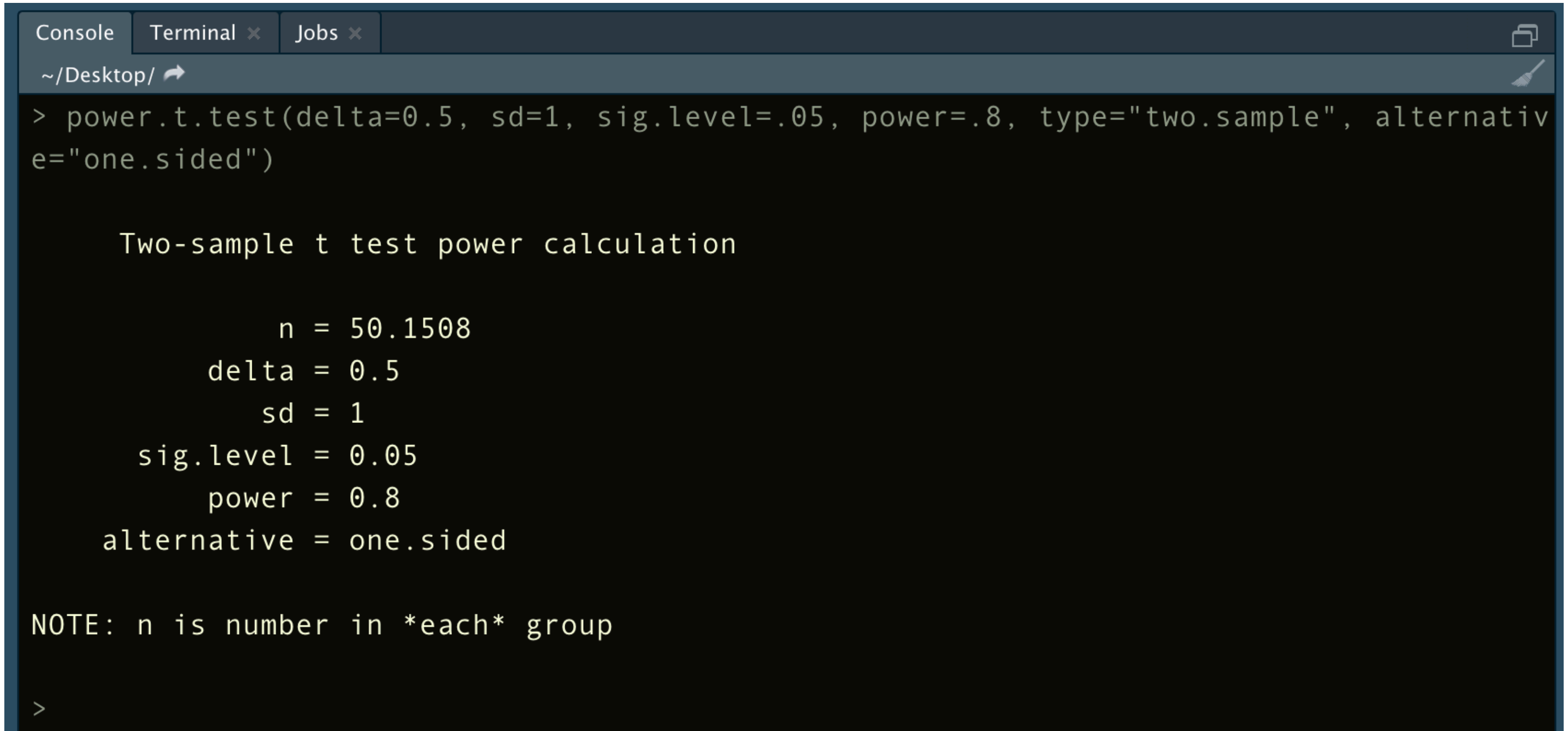
$$d = \frac{0.5}{1} = \frac{\mu_1 - \mu_2}{\sigma} \quad \text{so:} \quad \begin{aligned} \mu_1 - \mu_2 &= 0.5 \\ \sigma &= 1 \end{aligned}$$

So we put 0.5 into delta, and 1 into sd. Since 1 is the default for sd, this also means that you simply put Cohen’s d into delta!

Let’s try it one with Cohen’s d.

Let’s say we want to test for an effect with a Cohen’s d of 0.5, and we want power of .8 with an alpha of .05. We want to do this with two samples (independent samples t-test), and we have a one-tailed hypothesis. How many observations/participants do we need to collect in our sample?

Notice that we put Cohen’s d into delta directly, and we set the sd to 1. Very easy!

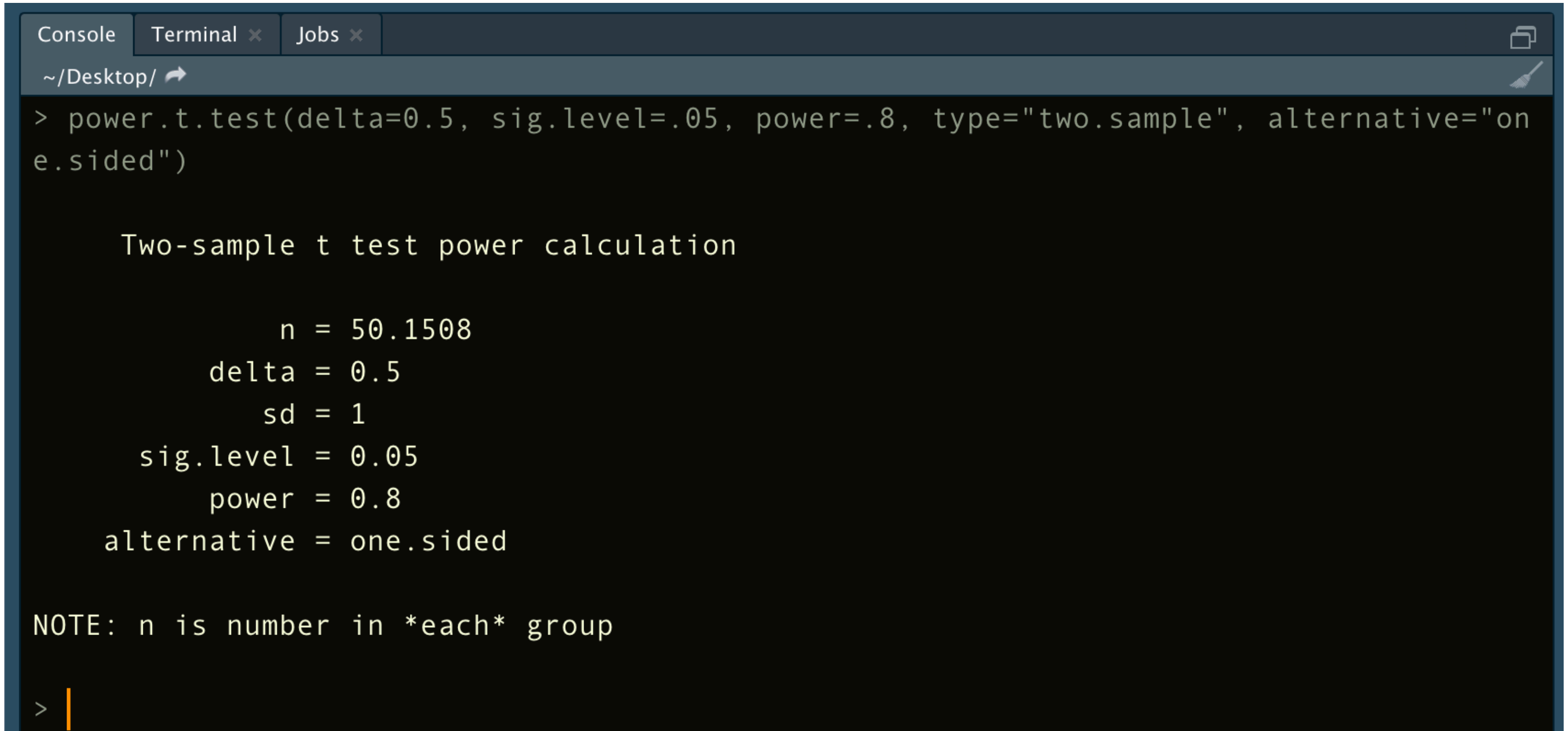


```
> power.t.test(delta=0.5, sd=1, sig.level=.05, power=.8, type="two.sample", alternativ
e="one.sided")

     Two-sample t test power calculation 

              n = 50.1508
          delta = 0.5
             sd = 1
      sig.level = 0.05
          power = 0.8
    alternative = one.sided

NOTE: n is number in *each* group

>
```

Let’s try it one with Cohen’s d.

Let’s say we want to test for an effect with a Cohen’s d of 0.5, and we want power of .8 with an alpha of .05. We want to do this with two samples (independent samples t-test), and we have a one-tailed hypothesis. How many observations/participants do we need to collect in our sample?

We can also leave the sd argument out entirely because its default value is 1. So when we are working with Cohen’s d, we can just enter that in delta, and move on!



```
> power.t.test(delta=0.5, sig.level=.05, power=.8, type="two.sample", alternative="on
e.sided")

     Two-sample t test power calculation 

              n = 50.1508
          delta = 0.5
             sd = 1
      sig.level = 0.05
          power = 0.8
    alternative = one.sided

NOTE: n is number in *each* group

> 
```

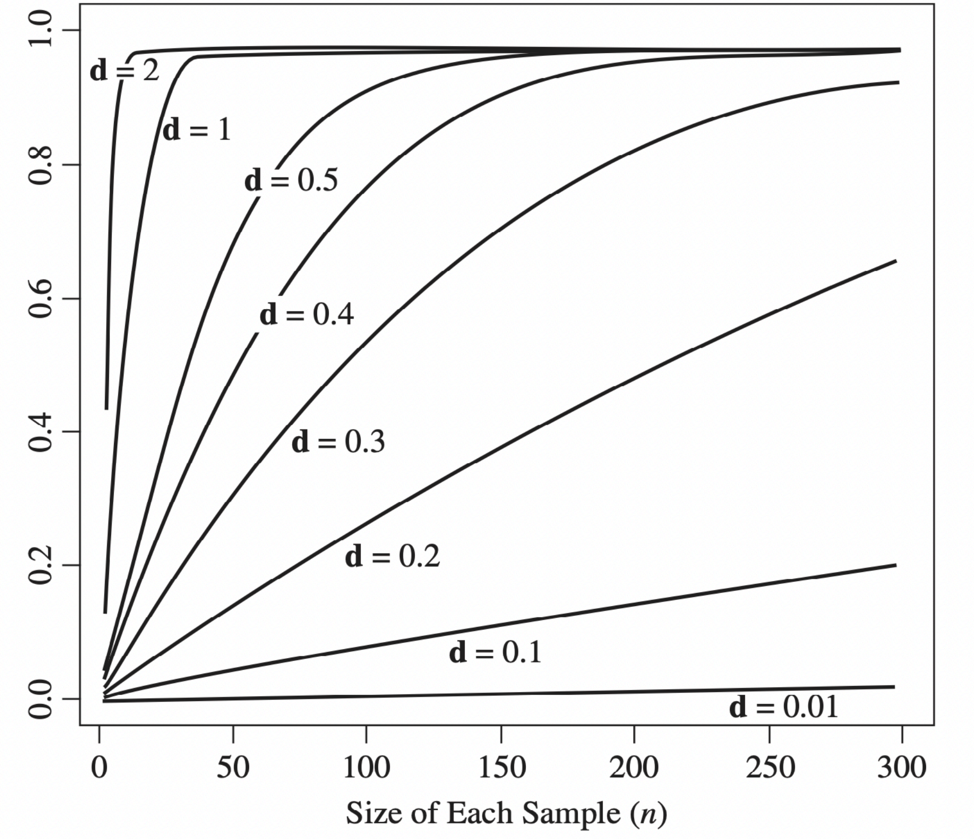
But where do you get the effect size from?

It is about your theory

As we discussed last time, effect sizes come from your theory. It is up to you, as a scientist, to use your scientific knowledge to say what the expected effect size is that is practically significant for your theory.

If you can use your theory to determine the Cohen's d for your effect, you can use a plot like the one in your book to simply look up the power that you would obtain at different sample sizes.

Let's take a moment to read this plot. It shows how power changes as a function of sample size for several different effect sizes.



Cohen suggested rules of thumb (which are not rules!) for determining if an effect size is small or large: 0.2 is small, 0.5 is medium, and 0.8 is large.

You can also estimate it from previous work

If previous research exists on your topic, you can use the formulas in the book to calculate an estimated effect size, called *g* in our book, from the numbers that people report in their articles.

We've already seen one way to calculate *g*. You can use the sample means and standard deviations that are reported in a paper, like this:

Cohen's d

$$d = \frac{\mu_1 - \mu_2}{\sigma}$$

Estimate

$$g = \frac{\bar{y} - \bar{x}}{s_p}$$

Another option is to use a *t*-statistic that is reported in a paper. Here is the formula for this:

$$d = \delta\sqrt{\frac{2}{n}}$$

$$g = t\sqrt{\frac{2}{n}}$$

You do not need to memorize these. The basic formula will probably stick in your mind because it is very similar to the formula for d. But the estimate using a *t*-statistic is rarely used. You can just look it up if you need it. (And, if you are curious about where it comes from, you can see the derivation in the book — it comes from the formula for *t*.)

Less common uses for the functions that calculate power

You can use it to calculate any of the 5 values

The power.t.test() function is very flexible. It can calculate any of the 5 arguments in the function. You just need to fill in 4 of them, and leave 1 null:

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE, tol = .Machine$double.eps^0.25)
```

So what else could you do?

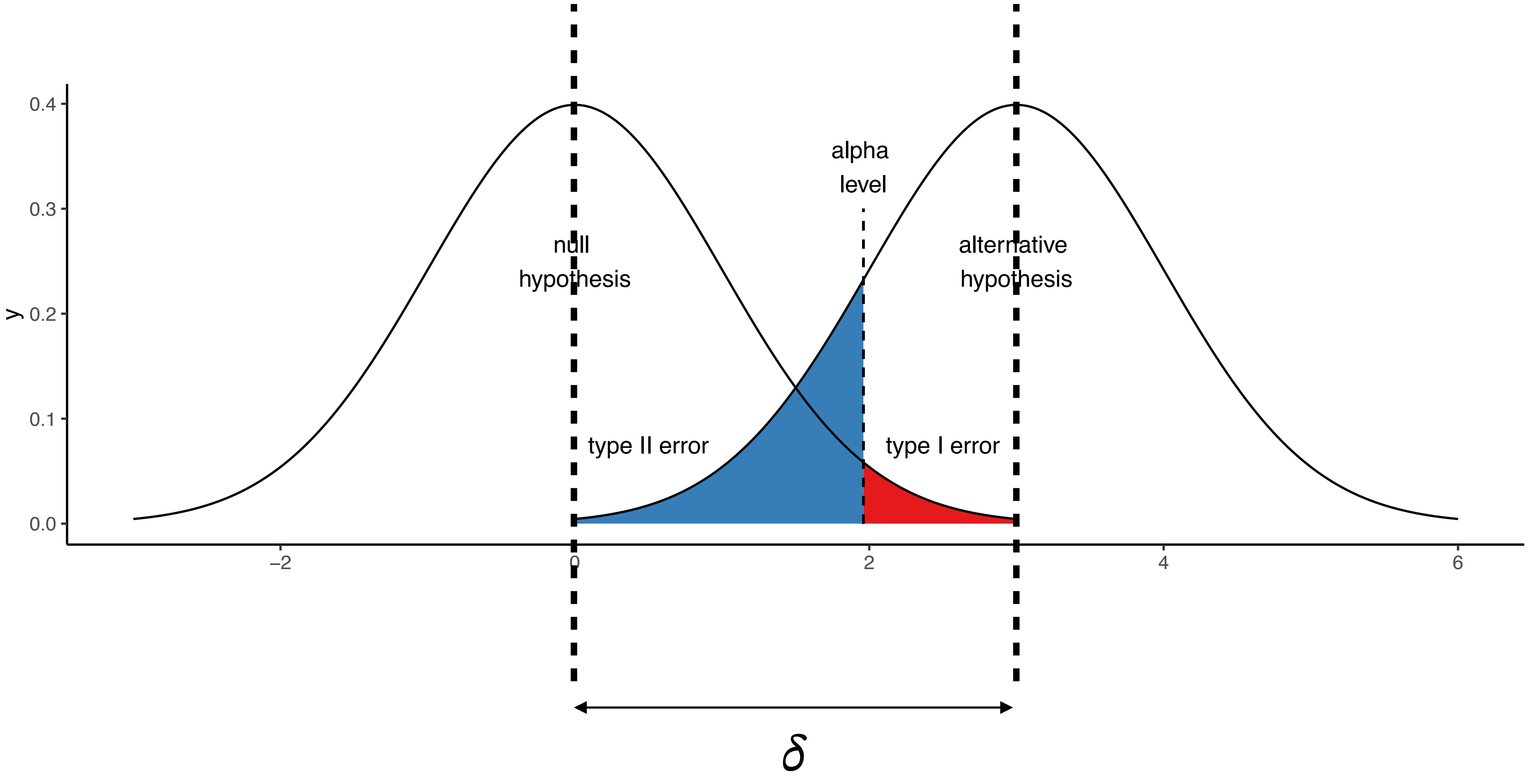
1. You can look at a study that has already been run, and determine how much power it had to detect an effect of that size. You fill in the n, delta, sd, and sig.level, and leave out the power.

2. You can look at a study that has already been run, and determine what the smallest effect size is that it could have detected with good power. You fill in the n, sd, sig.level, and power, and leave out the delta.

What is going on with δ in the book chapter?

This is called the non-centrality parameter

The non-centrality parameter is a way to describe the distribution of the alternative hypothesis. It is the distance between the central tendency of the null hypothesis distribution and the central tendency of the alternative hypothesis distribution. The term non-central means not centered on zero!



the non-centrality parameter

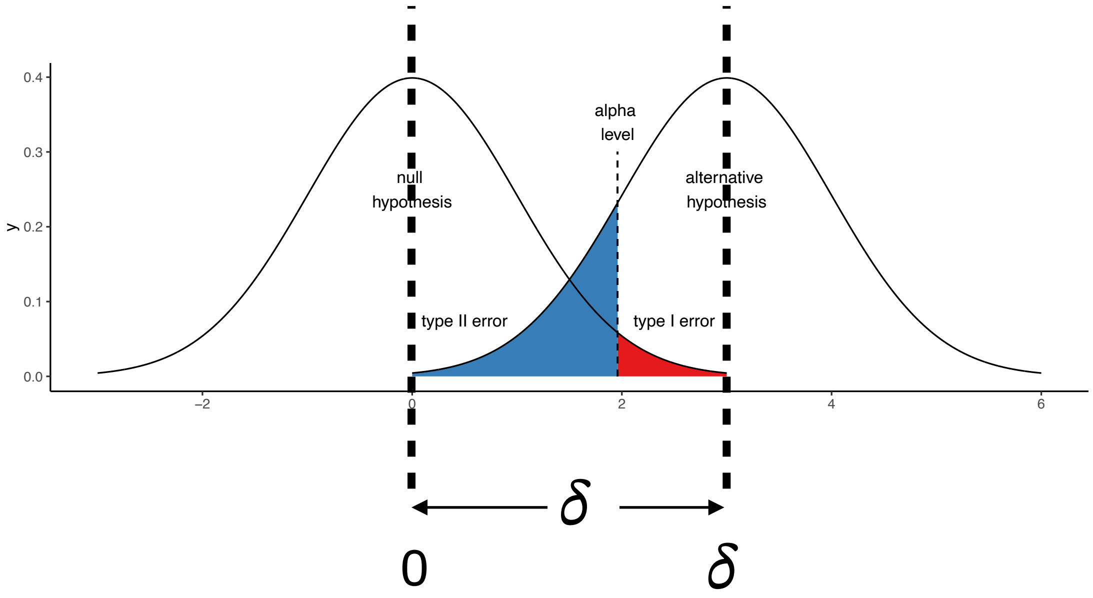
Using the non-centrality parameter is simpler

The function for computing power is complicated. It has 4 arguments. It is difficult to put 4 arguments in a table. So, if you want to create a table of power, you need to reduce the number of arguments. The non-centrality parameter helps us collapse three of these arguments together: the effect size, the standard deviation, and the size of the sample. You can see this in the formula - it contains d (effect size and sd) and n:

$$\delta = d\sqrt{\frac{n}{2}}$$

The formula derives from the formula for a t-test. You can see this in the book if you like.

One way to think of delta is as the *t*-value that you would expect if the alternative hypothesis is true. It is the mean *t*-statistic of the alternative hypothesis distribution, so it is the most common value. This is because the null distribution is centered on 0, so the distance between them becomes the value.



Though interesting, I don’t think you need it

The non-centrality parameter is an interesting concept in statistics. And it is certainly nice to reduce the power function to a table.

But, as a practical matter, you are very unlikely to use the non-centrality parameter for anything in your actual statistical analyses other than calculating power from a table. So, unlike other concepts that we teach you, this one seems limited in practical use for us. And the R function power.t.test() is much more flexible than the tables in the book. So I don’t want you to memorize delta. I won’t ask you to calculate anything with delta. I want you to learn how to use the power.t.test() function (which, confusingly, calls the effect size “delta”, because “delta” in math just means “difference”). This function will help you more in your work.

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE, tol = .Machine$double.eps^0.25)
```