

Where did Table 15.1 in Fundamentals text come from.

I generated Table 15.1 (and Appendix D.5) in *Fundamental Statistics for the Behavioral Sciences* many years ago, and when I was recently asked about how I generated it I had to think for a while. But actually that is pretty easy after a bit of thought.

The table that I refer to is shown below.

Table 15.1 Abbreviated Version of Table E.5, Power as a Function of δ and Significance Level

δ	Alpha for Two-Tailed Test			
	.10	.05	.02	.01
1.00	0.26	0.17	0.09	0.06
1.10	0.29	0.20	0.11	0.07
1.20	0.33	0.22	0.13	0.08
1.30	0.37	0.26	0.15	0.10
1.40	0.40	0.29	0.18	0.12
1.50	0.44	0.32	0.20	0.14
1.60	0.48	0.36	0.23	0.17
1.70	0.52	0.40	0.27	0.19
1.80	0.56	0.44	0.30	0.22
1.90	0.60	0.48	0.34	0.25
2.00	0.64	0.52	0.37	0.28
2.10	0.68	0.56	0.41	0.32
2.20	0.71	0.60	0.45	0.35
2.30	0.74	0.63	0.49	0.39
2.40	0.78	0.67	0.53	0.43
2.50	0.80	0.71	0.57	0.47
2.60	0.83	0.74	0.61	0.51
2.70	0.85	0.77	0.65	0.55
2.80	0.88	0.80	0.68	0.59
2.90	0.90	0.83	0.72	0.65
3.00	0.91	0.85	0.75	0.66

First, keep in mind that the approach in that chapter is based on an approximation to power. That approximation is based on the standard normal distribution (z). (I will give a more exact solution below, but the normal approximation is really good enough for just

about any purpose we want.) The advantage of the normal distribution is that the shape of the distribution does not depend on sample size, so I don't have to worry about my degrees of freedom. When I come to using the (noncentral) t distribution to give a more exact solution I will have to pay attention to the degrees of freedom. The distribution will still be symmetric, but it will not be normal.

I will start with an example that I expect to have a small effect size ($d = 0.20$). Imagine that I want to study to examine a product advertised to help you gain weight. If the product did not work I would expect a mean weight gain of 0 pounds (my null hypothesis), but from previous research I expect that the mean weight gain from 25 participants will be 3 pounds with a standard deviation of 15. (I have such a large standard deviation because many people will probably lose weight or even gain a large amount of weight even though the average gain is 3.) I want to compute the power of this experiment to find a significant mean weight gain if, as I expect, the true mean gain is 3 pounds with a standard deviation of 15.

$$d = \frac{\mu_1 - \mu_0}{\sigma} = \frac{3 - 0}{15} = .20$$

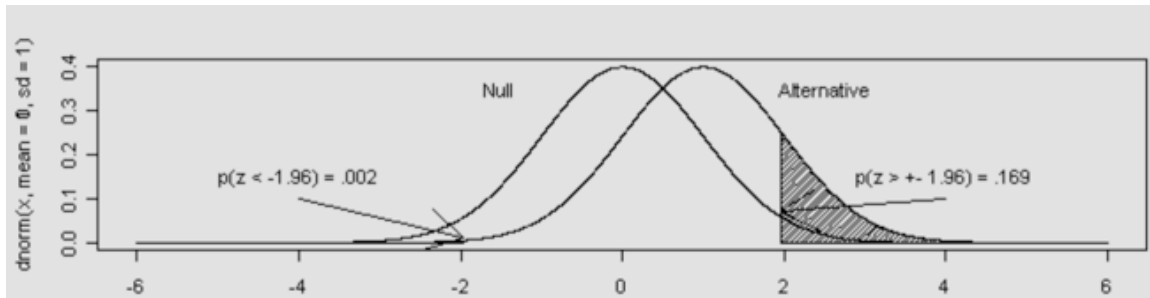
$$\delta = d\sqrt{n} = .20\sqrt{25} = 1.00$$

Notice that I have an effect size of .20, which Cohen would think of as a small effect. When we combine the effect size and the sample size to produce δ , which is required for use of Table 15.1, we have $\delta = 1.00$. (When we come to working with t instead of z , δ will be known as the “noncentrality parameter.”)

From Table 15.1 we see that for a two tailed test as $\alpha = .05$ the predicted power for our experiment is only .17, meaning that if we are correct in our expectations, only 17% of the time that we run this experiment will we get a significant result. But where did that .17 come from?

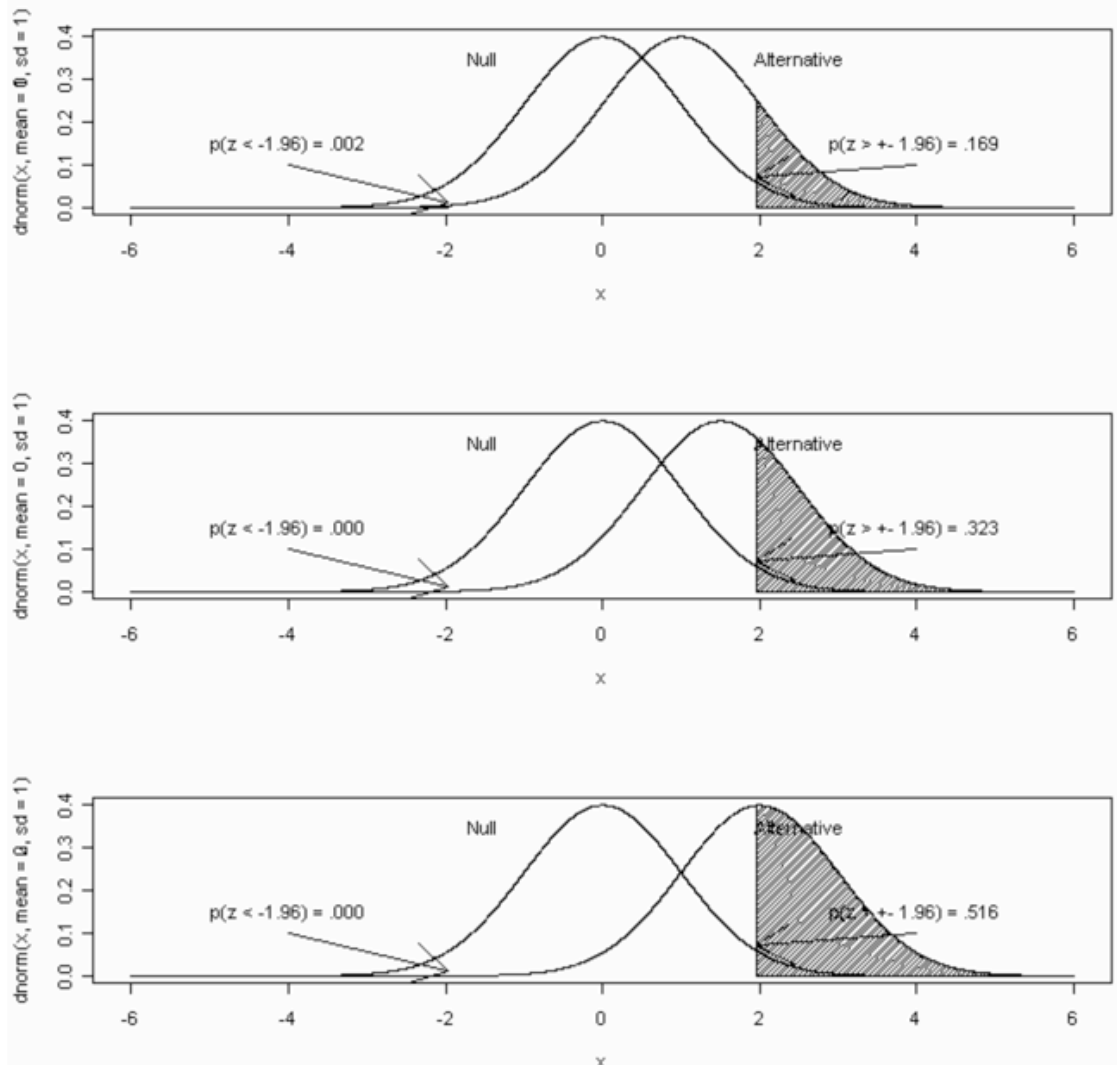
First of all remember that with a two-tailed z test (see Chapter 12, Section 12.2) the critical value at $\alpha = .05$ is ± 1.96 . In other words we will reject the null whenever we calculate a z on our sample data in excess of ± 1.96 . Also, if the μ_1 really equals 3 with a standard deviation of 5, as we expect, the distribution of z values for a huge number of replications of this experiment would be have a standard normal distribution around $\delta = 1$ with a standard deviation of 5. So all we really need to ask is “How often would we get an obtained value of z greater than +1.96 from a normal distribution with mean = 1 and standard deviation = 5?”

The diagram that follows illustrates that probability. Notice that on the left I have drawn the distribution of expected z values under the null, and on the right I have drawn expected z values under the alternative hypothesis. I have also given the areas under the right hand distribution for values outside the range of +1.96. If you add these two areas you get $.002 + .169 = .171$, which is the value given in the table for $\delta = 1$, $\alpha = .05$, two-tailed.



What about other values of δ ? I now want to repeat the above but consider other values of δ , which is the effect size adjusted for n . I could alter δ in several ways. I could change the expected mean, the expected standard deviation, or the sample size (n). I am going to do the latter, and I will aim for $\delta = 1.50$ and 2.00 , which will require sample sizes of 56 and 100, respectively.

The logic is exactly the same as it was for the first example except that I substitute the new values of δ . This will cause the alternative hypothesis to be displaced to the right, with more and more of the area under the alternative exceeding $+ 1.96$, resulting in greater power.



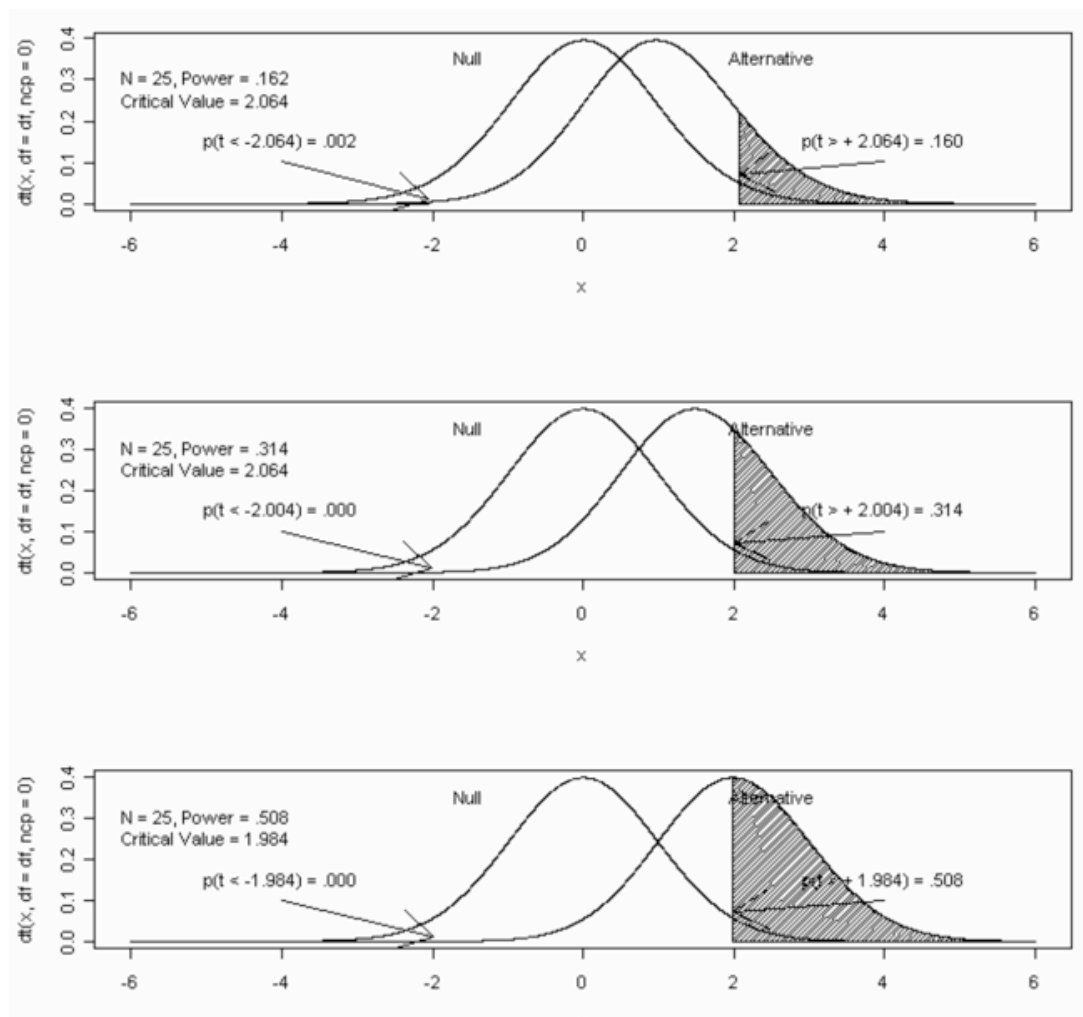
The results of these runs are shown in the figure below, where I have included the case with $\delta = 1$ simply to make it easier to compare. If you add the areas in both tails of the distribution you will get .32 and .52 for $\delta = 1.5$ and 2.0 , respectively. These agree with the values shown in Table 15.1.

But I Want to be More Exact—I Don't Want an Approximation.

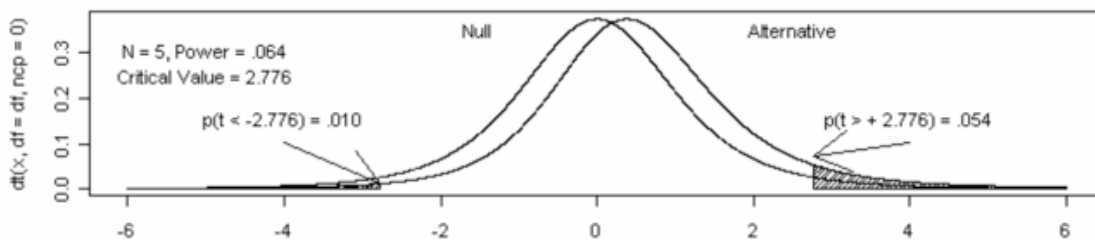
The calculations given above assume that you are going to use the normal distribution to approximate the power of our test. But that means that the critical values are always $+1.96$ regardless of the

sample size. The correct way to deal with this problem is to use the (noncentral) t distribution. It is called the noncentral t because it will be centered over the noncentrality parameter (δ) instead of over 0, as it is in the normal t tables.

Because I will get a different distribution for each value of sample size and for each value of δ , I could generate a huge number of figures similar to the ones above. Instead I am going to stick with $d = 0.20$ and vary n from 25 to 56 to 100. This will give me values of 1.0, 1.5, and 2.0. The only real change that I have to make to my program, other than to change the critical values in line with the changing degrees of freedom, is to replace commands to draw normal distributions with commands to draw t distributions.



From this figure you can see how power increases as sample size increases. You can also see that in the top plot power is .162, which is close to what we obtained with our approximation. However, the approximation is not good for very small sample sizes. Below I have included the same plot as above but with $n = 5$. Here you can see that the true power is .064 rather than the .17 that our approximation gives. However when you are working with such low power values you probably should not run that experiment to begin with.



The R program that generated these plots is shown below, but it was assembled quickly and is far from elegant.

```
#####
```

```
# Generate a sequence of x values
x <- seq(-6, 6, by = 0.001)
par(mfrow = c(3,1))
# Plot normal curve over x with mean at 1.0
plot(x, dnorm(x, mean = 0, sd = 1), type = "l")
par(new = T)
plot(x, dnorm(x, mean = 1, sd = 1), type = "l")
# Define left side boundary using min(x)
```

```

# and CritVal using alpha = 0.05
alpha <- 0.05
LCrit <- qnorm(alpha / 2)    #LCir = -1.96
UCrit <- qnorm(1-alpha/2)
xl <- seq(min(x), LCrit, length = 100)
yl <- c(dnorm(xl,1,1), 0, 0)
PUpper <- 1-pnorm(UCrit, 1,1) ; PUpper
PLower <- pnorm(LCrit,1,1); PLower
PTwotailed <- PUpper+PLower; PTwotailed
# add CritVal, min(x) to complete polygon
xl <- c(xl, LCrit, min(x))
# draw and fill left region
polygon(xl, yl, density = 50)
arrows(-4,.1,-1.96,.01)
arrows(4,.1,1.96,.07)
text(-4.0, .15, "p(z < -1.96) = .002")
text(4.0,.15,"p(z > +- 1.96) = .169")
text(-1.5,.35,"Null" )
text(2.5,.35,"Alternative")
# Do the same for right side boundary using max(x)
# and Crit Val

```



```
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dnorm(xu,1,1), 0,0)
# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)
# draw and fill left region
polygon(xu, yu, density = 50)
```

Now put the mean of the normal distribution at 1.5 and then 2.0

```
# Generate a sequence of x values
x <- seq(-6, 6, by = 0.001)

# Plot normal curve over x with mean at 1.0
plot(x, dnorm(x, mean = 0, sd = 1), type = "l")
par(new = T)
plot(x, dnorm(x, mean = 1.5, sd = 1), type = "l", ylab = "")
# Define left side boundary using min(x)
# and CritVal using alpha = 0.05
alpha <- 0.05
LCrit <- qnorm(alpha / 2)    #LCir = -1.96
```

```

UCrit <- qnorm(1-alpha/2)
xl <- seq(min(x), LCrit, length = 100)
yl <- c(dnorm(xl,1.5,1), 0, 0)
PUpper <- 1-pnorm(UCrit, 1.5,1) ; PUpper
PLower <- pnorm(LCrit,1.5,1); PLower
PTwotailed <- PUpper+PLower; PTwotailed
# add CritVal, min(x) to complete polygon
xl <- c(xl, LCrit, min(x))
# draw and fill left region
polygon(xl, yl, density = 50)
arrows(-4,.1,-1.96,.01)
arrows(4,.1,1.96,.07)
text(-4.0, .15, "p(z < -1.96) = .000")
text(4.0,.15,"p(z > +- 1.96) = .323")
text(-1.5,.35,"Null" )
text(2.5,.35,"Alternative")
# Do the same for right side boundary using max(x)
# and Crit Val
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dnorm(xu,1.5,1), 0,0)

```

```

# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)

# draw and fill left region
polygon(xu, yu, density = 50)

#####

# Generate a sequence of x values
x <- seq(-6, 6, by = 0.001)

# Plot normal curve over x with mean at 1.0
plot(x, dnorm(x, mean = 0, sd = 1), type = "l")
par(new = T)
plot(x, dnorm(x, mean = 2, sd = 1), type = "l")

# Define left side boundary using min(x)
# and CritVal using alpha = 0.05
alpha <- 0.05
LCrit <- qnorm(alpha / 2)    #LCir = -1.96
UCrit <- qnorm(1-alpha/2)

xl <- seq(min(x), LCrit, length = 100)
yl <- c(dnorm(xl,2,1), 0, 0)

PUpper <- 1-pnorm(UCrit, 2,1) ; PUpper
PLower <- pnorm(LCrit,2,1); PLower

```

```

PTwotailed <- PUpper+PLower; PTwotailed
# add CritVal, min(x) to complete polygon
xl <- c(xl, LCrit, min(x))
# draw and fill left region
polygon(xl, yl, density = 50)
arrows(-4,.1,-1.96,.01)
arrows(4,.1,1.96,.07)
text(-4.0, .15, "p(z < -1.96) = .000")
text(4.0,.15,"p(z > +- 1.96) = .516")
text(-1.5,.35,"Null" )
text(2.5,.35,"Alternative")
# Do the same for right side boundary using max(x)
# and Crit Val
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dnorm(xu,2,1), 0,0)
# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)
# draw and fill left region
polygon(xu, yu, density = 50)

#####

```

```

# This is the beginning of using the t distribution

# Now hold delta at 1.5, let n = 25, 56, and 100, and use noncentral
t distribution.

# Generate a sequence of x values

df <- 24

ncp <- 1

x <- seq(-6, 6, by = 0.001)

par(mfrow = c(3,1))

# Plot t curve over x with mean at 1.0

plot(x, dt(x, df = df, ncp = 0), type = "l")

par(new = T)

plot(x, dt(x, df = df, ncp = ncp), type = "l", ylab = "", yaxt = "n")

# Define left side boundary using min(x)

# and CritVal using alpha = 0.05

alpha <- 0.05

LCrit <- qt(alpha / 2, df = df, ncp = 0)

UCrit <- qt((1-alpha/2),df = df, ncp = 0)

xl <- seq(min(x), LCrit, length = 100)

yl <- c(dt(xl,df = df, ncp = ncp), 0, 0)

PUpper <- 1-pt(UCrit, df = df, ncp = ncp) ; PUpper

PLower <- pt(LCrit,df = df, ncp = ncp); PLower

PTwotailed <- PUpper+PLower; PTwotailed

```

```

# add CritVal, min(x) to complete polygon
xl <- c(xl, LCrit, min(x))

# draw and fill left region
polygon(xl, yl, density = 50)

arrows(-4,.1,-2.064,.01)
arrows(4,.1,2.064,.07)

text(-4.0, .15, "p(t < -2.064) = .002")
text(4.0,.15,"p(t > + 2.064) = .160")
text(-1.5,.35,"Null" )
text(2.5,.35,"Alternative")

# Do the same for right side boundary using max(x)
# and Crit Val
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dt(xu,df = df, ncp = ncp), 0,0)

# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)

# draw and fill left region
polygon(xu, yu, density = 50)

text(-5,.3,"N = 25, Power = .162")
text(-5, .25, "Critical Value = 2.064")

```

```
#####
```

```
# Plot
```

```
df <- 55
```

```
ncp = 1.5
```

```
plot(x, dt(x, df = df, ncp = 0), type = "l")
```

```
par(new = T)
```

```
plot(x, dt(x, df = df, ncp = ncp), type = "l", ylab = "", yaxt = "n")
```

```
# Define left side boundary using min(x)
```

```
# and CritVal using alpha = 0.05
```

```
alpha <- 0.05
```

```
LCrit <- qt(alpha / 2, df = df, ncp = 0)
```

```
UCrit <- qt((1-alpha/2),df = df, ncp = 0)
```

```
x1 <- seq(min(x), LCrit, length = 100)
```

```
y1 <- c(dt(x1,df = df, ncp = ncp), 0, 0)
```

```
PUpper <- 1-pt(UCrit, df = df, ncp = ncp) ; PUpper
```

```
PLower <- pt(LCrit,df = df, ncp = ncp); PLower
```

```
PTwotailed <- PUpper+PLower; PTwotailed
```

```
# add CritVal, min(x) to complete polygon
```

```
x1 <- c(x1, LCrit, min(x))
```

```

# draw and fill left region
polygon(xl, yl, density = 50)
arrows(-4,.1,-2.004,.01)
arrows(4,.1,2.004,.07)
text(-4.0, .15, "p(t < -2.004) = .000")
text(4.0,.15,"p(t > + 2.004) = .314")
text(-1.5,.35,"Null" )
text(2.5,.35,"Alternative")
# Do the same for right side boundary using max(x)
# and Crit Val
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dt(xu,df = df, ncp = ncp), 0,0)
# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)
# draw and fill left region
polygon(xu, yu, density = 50)
text(-5,.3,"N = 25, Power = .314")
text(-5, .25, "Critical Value = 2.064")
#####
# Plot t curve over x with mean at 1.0
plot(x, dt(x, df = df, ncp = 0), type = "l")

```



```

df = 99

ncp = 2.0

par(new = T)

plot(x, dt(x, df = df, ncp = ncp), type = "l", ylab = "", yaxt = "n")

# Define left side boundary using min(x)

# and CritVal using alpha = 0.05

alpha <- 0.05

LCrit <- qt(alpha / 2, df = df, ncp = 0)

UCrit <- qt((1-alpha/2),df = df, ncp = 0)

xl <- seq(min(x), LCrit, length = 100)

yl <- c(dt(xl,df = df, ncp = ncp), 0, 0)

PUpper <- 1-pt(UCrit, df = df, ncp = ncp) ; PUpper

PLower <- pt(LCrit,df = df, ncp = ncp); PLower

PTwotailed <- PUpper+PLower; PTwotailed

# add CritVal, min(x) to complete polygon

xl <- c(xl, LCrit, min(x))

# draw and fill left region

polygon(xl, yl, density = 50)

arrows(-4,.1,-1.984,.01)

arrows(4,.1,1.984,.07)

text(-4.0, .15, "p(t < -1.984) = .000")

```

```

text(4.0,.15,"p(t > + 1.984) = .508")
text(-1.5,.35,"Null" )
text(2.5,.35,"Alternative")
# Do the same for right side boundary using max(x)
# and Crit Val
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dt(xu,df = df, ncp = ncp), 0,0)
# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)
# draw and fill left region
polygon(xu, yu, density = 50)
text(-5,.3,"N = 25, Power = .508")
text(-5, .25, "Critical Value = 1.984")
#####Now with sample size of only 5. ncp = .447
# Generate a sequence of x values
df <- 4
ncp <- .447
x <- seq(-6, 6, by = 0.001)
par(mfrow = c(3,1))
# Plot t curve over x with mean at 1.0
plot(x, dt(x, df = df, ncp = 0), type = "l")

```

```

par(new = T)
plot(x, dt(x, df = df, ncp = ncp), type = "l", ylab = "", yaxt = "n")
# Define left side boundary using min(x)
# and CritVal using alpha = 0.05
alpha <- 0.05
LCrit <- qt(alpha / 2, df = df, ncp = 0)
UCrit <- qt((1-alpha/2),df = df, ncp = 0)
xl <- seq(min(x), LCrit, length = 100)
yl <- c(dt(xl,df = df, ncp = ncp), 0, 0)
PUpper <- 1-pt(UCrit, df = df, ncp = ncp) ; PUpper
PLower <- pt(LCrit,df = df, ncp = ncp); PLower
PTwotailed <- PUpper+PLower; PTwotailed
# add CritVal, min(x) to complete polygon
xl <- c(xl, LCrit, min(x))
# draw and fill left region
polygon(xl, yl, density = 50)
arrows(-4,.1,-2.776,.01)
arrows(4,.1,2.776,.07)
text(-4.0, .15, "p(t < -2.776) = .010")
text(4.0,.15,"p(t > + 2.776) = .054")
text(-1.5,..35,"Null" )

```

```
text(2.5,.35,"Alternative")
# Do the same for right side boundary using max(x)
# and Crit Val
xu <- seq(UCrit, max(x), length = 100)
yu <- c(dt(xu,df = df, ncp = ncp), 0,0)
# add max(x) and CritVal to complete polygon
xu <- c(xu, max(x) , UCrit)
# draw and fill left region
polygon(xu, yu, density = 50)
text(-5,.3,"N = 5, Power = .064")
text(-5, .25, "Critical Value = 2.776")
```

dch: 7/6/2015