Test Statistics, Null and Alternative Distributions

Type II errors, Power, Effect Size, and Non-central Distributions

Bruce Dudek

2024-11-14

Contents

1 Preface

This document arose as a core document in an introductory graduate level statistics course in a discipline (psychology). As such, it has some references to the course sequence of topics. However, it is also structured so that it can be a standalone document useful to others outside of the course. The course is algebra-based rather than calculus-based and the required background would be the basics of the null hypothesis significance testing framework, and review is also provided. The value of the document is a simulation based approach to the introduction and usage of a non-central t distribution and the logical extension to power concepts and sample size planning for researchers.

2 Background

Prior course work introduced the concepts of Type I and II errors in some detail, but in a limited context. The Type II error concept is consistently defined across NHST methods as the probability of failing to reject the null hypothesis when it is false (and a specific alternative hypothesis is specified). This document does several things. First, it reviews, with simulation, the earlier classroom development of the Type II error concept in the context of a one-sample location test where the population variance is known. Then the document extends this idea

to a one sample "t-tests" where population variances are unknown and are estimated. That sets the stage for further extension to other methods later in the course sequence. The most important additional concept that requires development is that of the non-centrality parameter and non-central distributions such as the non-central t. The approach is anchored in simulation rather than mathematical derivation, and it is facilitated by visualizations.

An additional goal of the document is the consolidation of a general understanding that can permit the student to use such tools as the **pwr** package in R or the GPower software that is widely used. Both can be important tools for sample size planning, but require a modicum of understanding of effect sizes, non-central distributions, and power.

3 R Essentials

Several R packages are required in this document. Both base R and the **ggplot2** package are used for visualizations in multiple places and the **ggfortify** package adds useful capabilities for distribution plotting.

```
library(ggplot2)
library(psych)
library(car)
library(ggfortify)
library(knitr)
library(gt)
library(pwr)
```

Package citations for packages loaded here (in the above order): **ggplot2** (Wickham et al., 2018), **psych** (Revelle, 2019), **car** (Fox, Weisberg, & Price, 2018), **ggfortify** (Horikoshi & Tang, 2019), **knitr** (Xie, 2018), **gt** (Iannone, Cheng, & Schloerke, 2019), **pwr**(Champely, 2018)

4 Review of power concepts using the Z_M NHST method

The introduction to traditional null hypothesis significance testing is most readily accomplished with the single sample "Z of a mean" test. The null hypothesis specifies the mean (μ_o) of a distribution. The alternative typically specifies that the true mean (μ_1)is either different than the null value or deviant higher or lower when the test is a one tailed test. The test is called a "Z of a mean" test because the observed sample mean is standardized, relative to the null value

$$\left(\bar{X} - \mu_0\right) / \left(\sigma_X / \sqrt{n}\right).$$

The standard error in the denominator is the theoretical std deviation of the sampling distribution of the mean, based on the population RV std deviation and sample size ($^{\sigma_x}/_{\sqrt{n}}$). Our interest is in how this sample mean behaves (if repeated samples are taken from that null distribution) and we can re-scale the sample mean in terms of std error units when we compute the "Z of a mean"

This Z of a mean taken on our one real sample is also called our test statistic. We compare it to values expected if the null is true by using a standard normal distribution. This document explores how means and test statistics behave under various hypotheses and other specifications. The goals are:

- furthering understanding of Type II errors.
- reinforcing the understanding that power is computed "against a specific alternative".
- laying the foundation for consideration of non-central versions of the t, chi-square and F distributions.

4.1 Visualizing sampling distributions under different hypotheses. Part 1: The null hypothesis distribution.

First, let's specify that sample(s) are drawn from a hypothetical population distribution of the RV where $\mu_0 = 100$ (the null hypothesis value) and $\sigma_x = 15$ (a common occurence for many normed test instruments). Further assume that our samples are each comprised of 25 observations (n=25). The theoretical sampling distribution of the mean based on these specifications has a standard deviation of 3: $\left(^{\sigma_x}/_{\sqrt{n}}\right)$.

We can simulate this by randomly drawing 10000 sample means from this sampling distribution and showing the first few and last few produced -

($\mu_0 = 100$ and $\sigma_x = 15$)

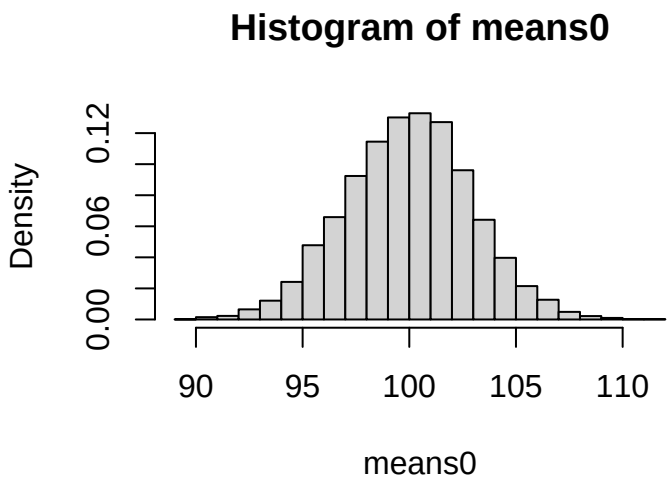
so, ($\mu_{\bar{X}} = 100,\ and\ \sigma_{\bar{X}} = 3$) when H_0 is true.

```
set.seed= 1000
means0 <- round(rnorm(10000,mean=100,sd=3),digits=2)
psych::headTail(means0) # shows the first few and last few elements of the object
```

```
  [,1]            [,2]            [,3]            [,4]
h "97.51"         "101.59"        "98.16"         "99.44"
  "...       ..." "...       ..." "...       ..." "...       ..."
t "103.62"        "102.29"        "95.33"         "99.62"
```

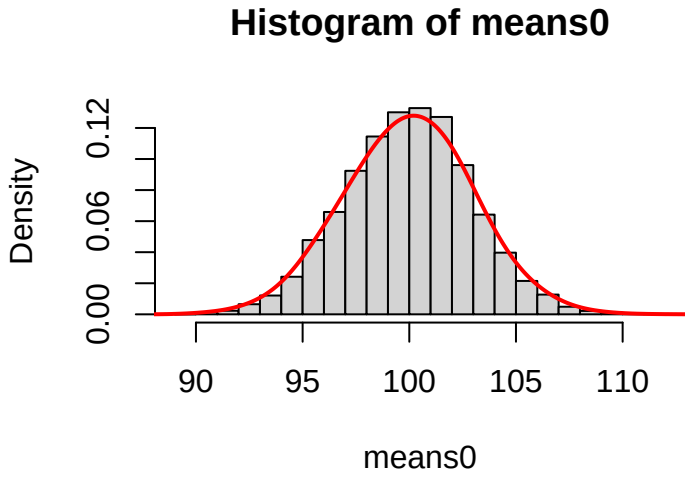
Next, we can draw a frequency histogram of those 10,000 means.

```
hist(means0,breaks=30, prob="TRUE")
```



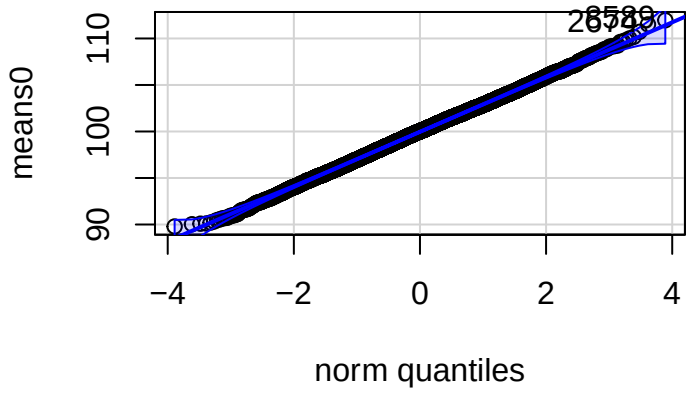
And we can add a kernel density function to better visualize the shape.

```
hist(means0,breaks=30, prob="TRUE")
lines(density(means0,bw=1),col="red",lwd=2)
```



This simulated distribution looks quite normal. Lets check with a qq plot.

```
car::qqPlot(means0) # the default qqPlot is a qq normal plot
```




```
[1] 8589 2674
```

What are the mean and std deviation of these simulated means? Are they close to the 100 and 3 values expected?

```
mean(means0)
```

```
[1] 99.97235
```

```
sd(means0)
```

```
[1] 2.960721
```

We can thus see how the $\bar{X}$ statistic "behaves" when the null is true and samples are drawn from that null hypothesis distribution.

But what if we converted our 10,000 means each to a "Z of a mean"? We need to subtract 100 from each and divide by the std error (3). We can do that operation with the whole vector of 10000 simulated means.

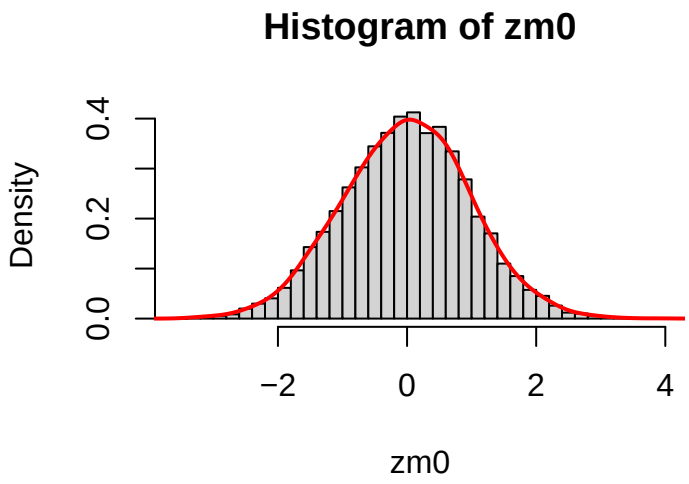
Since we are doing this with the sample means drawn from the null hypothesis distribution, we are simply doing a standardization of that sampling distribution. Assuming normality of the original sampling distribution, this distribution should also approximate a std normal.

```
zm0 <- round((means0-100)/3,digits=4)
psych::headTail(zm0)
```

```
  [,1]            [,2]            [,3]            [,4]
h "-0.83"         "0.53"          "-0.6133"       "-0.1867"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "1.2067"        "0.7633"        "-1.5567"       "-0.1267"
```

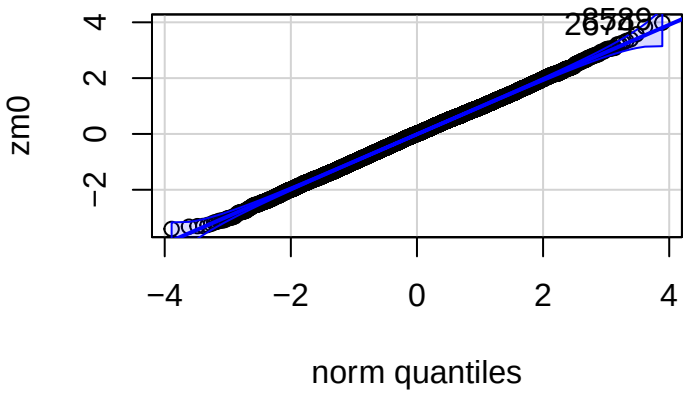
And we can visualize this standardized distribution in the same ways.

```
hist(zm0,breaks=30,prob="T")
lines(density(zm0,bw=.2),col="red",lwd=2)
```



It also shows the expected normality.

```
car::qqPlot(zm0)
```

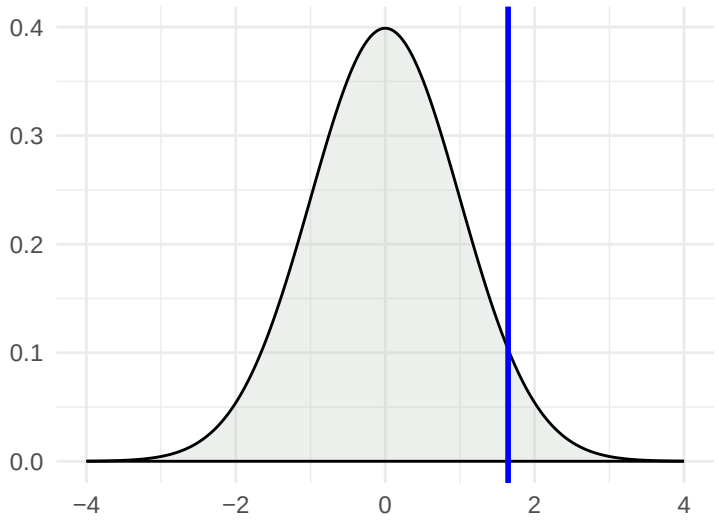



```
[1] 8589 2674
```

The central idea here is that when a. the null is true and b. we take one sample, compute a mean, and standardize it, that Z of a mean can be viewed as coming from a standard normal Z distribution. Therefore, our TEST STATISTIC behaves in this specified way when the null is true (and the normality assumption is satisfied)

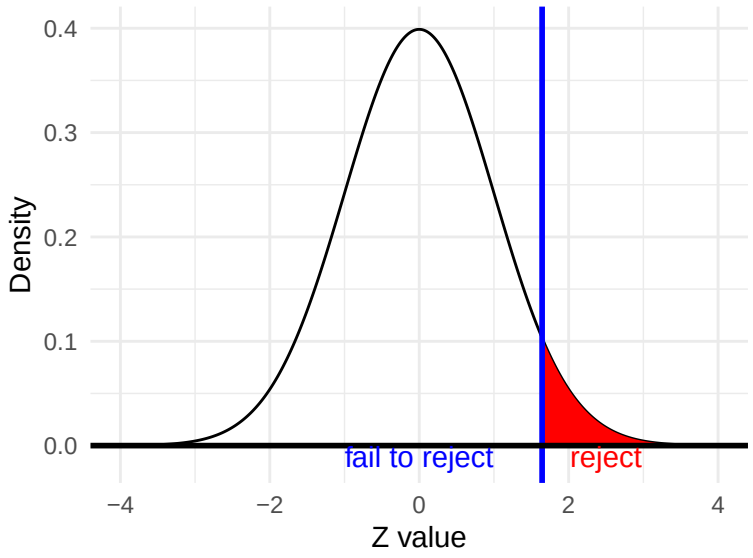
If we draw this theoretical null hypothesis distribution (the one that we just simulated) using 'ggplot2' tools we can also identify a "critical value" for a one-tailed (upper tail) test based on an alpha of .05 - thus "seeing" the Type I error rate. The first plot of this distribution uses the 'ggdistribution' function which makes the plot easy to draw, but does not permit some of the additional attributes that are found on the second plot. The 'ggdistribution' function can plot a distribution using any of the built-in probability functions that R has by using the "d" prefix for density.

```
# library(ggfortify)
p <- ggdistribution(dnorm, seq(-4, 4, .002), fill = 'honeydew3')
p +     geom_vline(aes(xintercept=qnorm(.05,0,1,lower.tail=F)),
                   linetype="solid", linewidth=1, colour="blue") +
        theme_minimal()
```



The second way of doing this plot uses straight 'ggplot2' methods and controls many attributes of the plot.

```
#Plot prob dist in ggplot (not with ggfortify)
x <- seq(-4, 4, .002)
y <- dnorm(x,0,1)
dat <- cbind.data.frame(x,y)
p <- ggplot(dat, aes(x = x, y = y)) +
  geom_line() +
  geom_area(mapping = aes(x =
                            ifelse(x>qnorm(.05,0,1,lower.tail=F),
                                   x, qnorm(.05,0,1,lower.tail=F))),
                            fill = "red") +
  xlim(-4,4) + ylim(-.015,.4) +
  geom_vline(aes(xintercept=qnorm(.05,0,1,lower.tail=F)),
             linetype="solid", linewidth=1, colour="blue") +
  geom_hline(aes(yintercept=0),
             linetype="solid", linewidth=1, colour="black") +
  annotate("text", label = "fail to reject", x = 0, y = -.01, size = 4, colour = "blue") +
  annotate("text", label = "reject", x = 2.5, y = -.01, size = 4, colour = "red") +
  labs(x="Z value", y="Density") +
  theme_minimal()
p
```



4.2 Visualizing sampling distributions under different hypotheses. Part 2: The alternative hypothesis distribution.

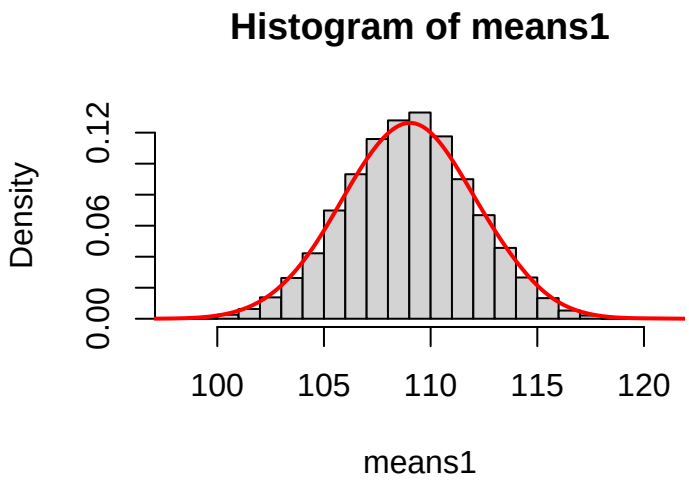
If the null hypothesis is false, then means will not be distributed according to the specifications defined above for the first simulation. When the inferential test is performed, we typically specify "inexact" alternative hypotheses. But the understanding of Type II errors and power requires spcecification of an exact alternative. This does not change whether the test is done as a 1 or 2-tailed test with inexact alternative(s). We say that power and type II errors are found "against a specific alternative". For example, if the true distribution that samples are drawn from has a mean higher than 100 (say 109), then the theoretical sampling distribution of the mean will be shifted nine units higher (or 3 standard errors). We can simulate this here.

```
set.seed= 1000
means1 <- round(rnorm(10000,mean=109,sd=3),digits=2)
psych::headTail(means1)
```

```
  [,1]            [,2]            [,3]            [,4]
h "110.92"        "110.31"        "111.96"        "108.1"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "111.19"        "108.46"        "105.28"        "107.78"
```

If we plot these simulated means in a histogram and add a density function, the shift to a location of 109 can be visualized, but note that the shape is still normal.

```
hist(means1,breaks=30, prob="TRUE")
lines(density(means1,bw=1),col="red",lwd=2)
```



If we standardize (relative to our null value of 100) our simulated set of means from this second simulation, we have another set of Zm's. It's shape is also normal, but note that the mean is at a Z value of approximately 3. This is because 109 is 3 standard errors above the null value that we standardized it to. This shift of 3 Z units can be described as an effect size (muo -mu1/std dev).

```
zm2 <- (means1-100)/3
mean(zm2)
```

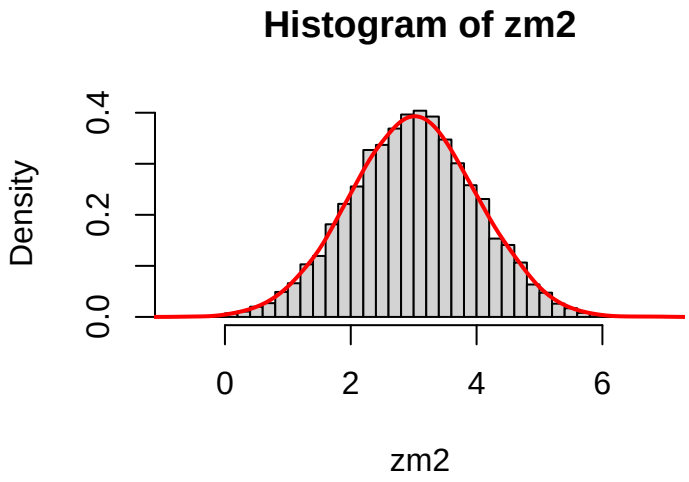
```
[1] 2.997742
```

```
sd(zm2)
```

```
[1] 0.9980483
```

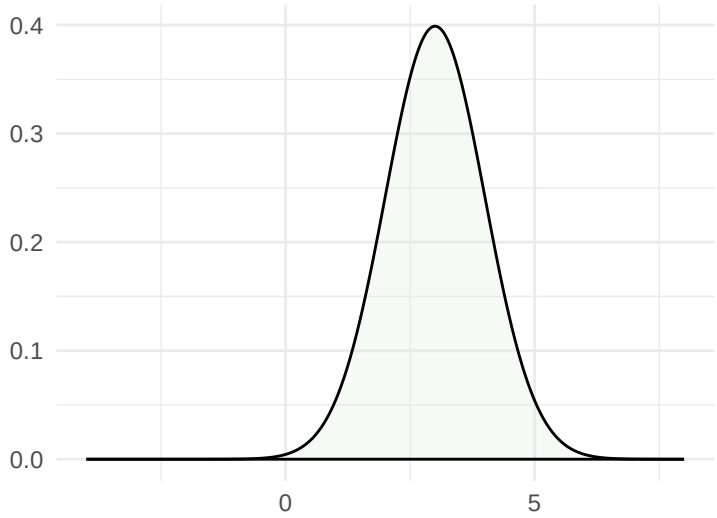
Plotting these standardized Zm values also reveals the expected normal shape (central limit theorem still in play).

```
hist(zm2,breaks=30, prob="TRUE")
lines(density(zm2, bw=.2),col="red",lwd=2)
```



Drawing a more accurate depiction of this theoretical sampling distribution with ggplot2 tools can reinforce the impression that the alternative hypothesis sampling distribution is normal.

```
p <- ggdistribution(dnorm, seq(-4, 8, .002), mean=3, sd=1, fill = 'honeydew2') +
        theme_minimal()
p
```

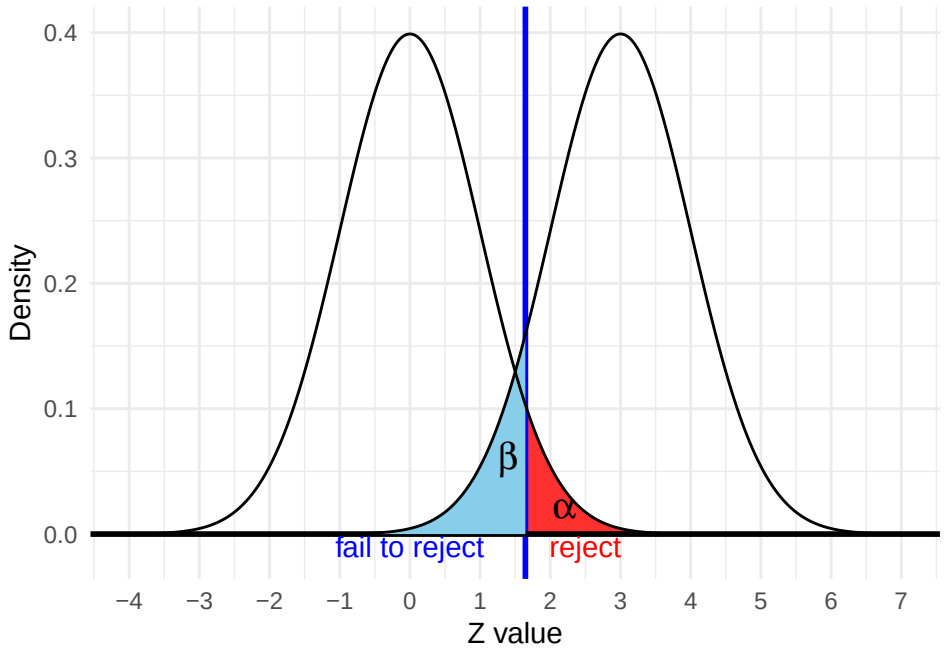


4.3 Visualizing sampling distributions under different hypotheses. Part 3: Visualize both the null and alternative distributions

If we plot both of these theoretical sampling distributions of the Zm statistic on the same graph, and also include the visual indicator of the critical value (at the vertical line), then the familiar visualization of both alpha and beta/power is possible. Students will recognize the similarity of this visualization from the earlier introduction to this topic and with use of the SHINY app, except that the plots here are of the test statistic (Z_M) rather than the raw sample means

The key elements are that 1) alpha is defined as an area based on the null distribution (it is .05) and 2) beta is found under the alternative distribution (it equals about .09). Another point to reinforce is that both sampling distributions are normal. This is because when means from the alternative distribution are standardized, each mean has a constant subtracted (the null mean) and this difference is also divided by a constant (the standard error). This latter point will not be operative when considering sampling from population distributions where the variance is unknown and estimated from sample data (next section).

```
#Plot prob dist in ggplot (not with ggfortify)
x <- seq(-4, 7, .002)
y0 <- dnorm(x,0,1)
y1 <- dnorm(x,3,1)
dat <- cbind.data.frame(x,y0,y1)
p3 <- ggplot(dat, aes(x = x, y = y0)) +
  #geom_line() +
  geom_area(mapping = aes(x = ifelse(x>qnorm(.05,0,1,lower.tail=F), x, qnorm(.05,0,1,lower.ta
        xlim(-4,7) + ylim(-.015,.4) +
  geom_vline(aes(xintercept=qnorm(.05,0,1,lower.tail=F)),
             linetype="solid", size=1, colour="blue") +
  geom_hline(aes(yintercept=0),
             linetype="solid", size=1, colour="black") +
  annotate("text", label = "fail to reject", x = 0, y = -.01, size = 4, colour = "blue") +
  annotate("text", label = "reject", x = 2.5, y = -.01, size = 4, colour = "red") +
  annotate("text", label = expression(alpha), x = 2.2, y = .02, size = 5, colour = "black") +
  scale_x_continuous(breaks=seq(-4,7,1)) +
  labs(x="Z value", y="Density") +
  theme_minimal()
p3 +
  geom_area(mapping = aes(x = ifelse(x<qnorm(.05,0,1,lower.tail=F), x, 0), y=y1), fill = "sky
  geom_line(mapping=aes(x=x,y=y1)) +
    geom_line(mapping=aes(x=x,y=y0)) +
    annotate("text", label = expression(beta), x = 1.4, y = .06, size = 5, colour = "black")
```


0.4
0.3
0.2
0.1
0.0
Density
β
α
fail to reject
reject
−4
−3
−2
−1
0
1
2
3
4
5
6
7
Z value

5 Power for tests using the t distributon: Intro to the Non-central t distribution

When the standard normal ("Z") distribution is not used for NHST tests other distributions such as the t, F and Chi-squared are often employed. A nice attribute of Z tests is that under Alternative hypotheses, sampling distributions of tests statistics such as Z_M are also normal. This leads to the ability to evaluate Type II error rates and power using the standard normal distribution as we saw above. Other tests, such as "t-tests" do not have this characteristic of alternative hypothesis sampling distributions. The typical t-distributions that we have considered can be called central t-distributions. But when the null is false, not only are the sampling distributions shifted from a mean of zero, they also change shape. This shape change, in the case of t-tests, is known to follow what are called non-central t-distributions and are skewed. This section introduces non-central t-distributions using a simulation approach that parallels what was done above for the Z_M test. First, we will simulate the null hypothesis sampling distribution and show that it follows the expected t distribution. Then we will simmulate the test statistic's sampling distribution when the null is false in order to introduce the non-central t-distribution. Initially these examples are done with the same sampling situation as above, the one-sample test of a mean.

5.1 Visualizing the test statistic distribution for a One-sample t-test of a mean. Part I: Null hypothesis true

When evaluating a null hypothesis regarding the value of a single population mean, we reviewed (above) Type II errors and Power for the situation where the population variance is known. This leads to use of the Z_M test. Now we evaluate the one-sample t-test for situations where the population variance is not known and must be estimated from the sample variance and standard deviation. The test statistic was developed previously and is

$$\frac{\bar{X} - \mu_0}{s_x/\sqrt{n}}$$

.

This expression mimics the Z_M construction except for the use of the sample standard deviation in the denominator. This means that each test statistic (in repeated samples) would be expected to vary in both numerator and denominator - each of the two sample statistics coming from its own sampling distribution - independently if the RV is normally distributed. This has implications for the shape of the sampling distribution of the test statistic, as we will see with the simulations below.

The approach to simulation of multiple samples is also different than how it was done above for the Z_M test. The R code for creating a large number of simulated replicate samples, each with their own mean and standard deviation, is more involved. For each replicate sample, N

scores are sampled from a normal distribution and from those scores, the sample mean, the sample standard deviation, the one-sample test statistic ($\frac{\bar{X}-\mu_0}{s_x/\sqrt{n}}$) and the p-value for that test are all computed.

We begin by assuming that the null hypothesis is true: $H_0 : \mu = \mu_0$ In addition we will duplicate the characteristics of the Z_M test above: $\mu_0 = 100$ *and* $\sigma_X = 15$ In order to do the simulation the population variance (or sd) has to be specified so that the n scores can be randomly simulated. But in an actual application of this test, these parameters are unknown and estimated from the sample data. So, the sample standard deviation is found in each replicate simulation, along with the sample mean.

The initial code chunk here creates 10000 replicate samples, each of n=25. It returns the means, sd's, t values, and p values for each of the samples in vectors that are used in following code chunks. The 'headTail' function from the **psych** package returns the first and last four values of the 10000 replicate samples for the sample mean and t-values as an illustration of success of the simulation.

```
# establish basic characteristics of the sampling situation
n <- 25      # sample size
mu <- 100  # true mean of the population distribution of X (null is true)
sigma <- 15  # true SD of the population distribution of X
mu0 <- 100   # mean under the null hypothesis
reps <- 10000 # number of simulations

## initialize a few vectors:
#xvals <- as.matrix(NA,ncol=n,nrows=reps)
xbars <- numeric(reps)
stdevs <- numeric(reps)
tvals <- numeric(reps)
pvalues <- numeric(reps)

set.seed(14) # for reproducibility
for (i in 1:reps) {
  x <- rnorm(n, mu, sigma)
  xbars[i] <- mean(x)
  stdevs[i] <- sd(x)
  tvals[i] <- (mean(x) - mu0)/(sd(x)/sqrt(n))
  pvalues[i] <- 2*(1 - pt(abs(tvals[i]), n-1))
  # alternatively: pvalues[i] <- t.test(x, mu = mu0)$p.value
}
psych::headTail(round(xbars, 2)) # round to two decimal places
```

```
  [,1]            [,2]            [,3]            [,4]
```

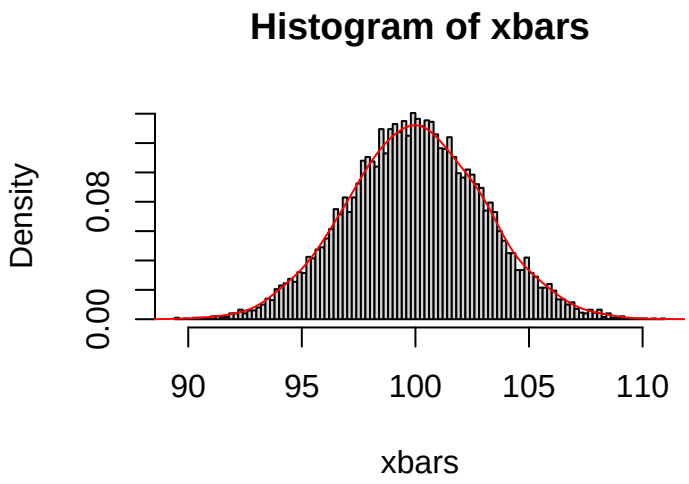
```
h "107.81"        "98.31"         "98.27"         "98.2"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "101.39"        "105.62"        "100.65"        "104.8"
```

```
psych::headTail(round(tvals, 2)) # round to two decimal places
```

```
  [,1]            [,2]            [,3]            [,4]
h "2.96"          "-0.65"         "-0.63"         "-0.72"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "0.43"          "1.96"          "0.24"          "1.72"
```

Since the simulation sampled values from a normal distribution with a mean of 100, this simulated sampling distribution of $\bar{X}$ should also center on 100 and its shape should be normal. The 'describe' function shows that the mean of the simulated sampling distribution is close to 100, its standard deviation is close to the value of 3 expected ($\sigma_x / \sqrt{n}$). Note that the skewness is close to zero, as expected for the sampling distribution of a mean based on sizeable n.

```
#hist(pvalues,breaks=100,prob=T)
#lines(density(pvalues), col="red")
hist(xbars, breaks=100, prob=TRUE)
lines(density(xbars, bw=.6), col="red")
```

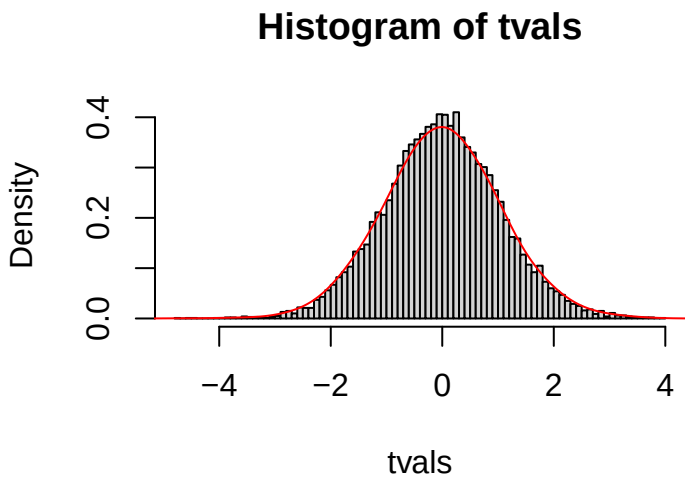


```
d1 <- describe(xbars)
d1b <- d1[,c(3,4,11)]
gt(d1b)
```

mean	sd	skew
99.99266	2.984081	0.01455882

Next we can examine the simulated sampling distribution of the t test statistic. It has the expected center near zero, and should look like a t distribution with df=24. Theoretically, it should also be symmetrical and we see that the skewness is close to zero

```
hist(tvals, breaks=100, prob=TRUE)
lines(density(tvals, bw=.3), col="red")
```




```
d2 <- describe(tvals)
d2b <- d2[,c(3,4,11)]
gt(d2b)
```

mean	sd	skew
-0.0037544	1.0398	0.009814452

A better way to evaluate the simulated sampling distribution of the t values is to submit those values to a qq plot. Although we have only examined qqnormal plots in prior work, it is possible to compare a variable to any probability distribution. The 'qqPlot' function permits this. First, in this next code chunk, we look at the empirical 5th and 95th percentile values. They are close to the exact values found with the 'qt' function for an actual t distribution with df=24. In addition, the qq plot verifies a pretty good match of the 100000 simulated values with the theoretical t distribution. It is a little bit off in the extremities of the tails, so even 10000 replicates generates a few outliers. Changing the seed for random number generation would modify this outlier behavior and increasing the number of replicates would minimize them. Thus when the null is true, the t test statistic behaves as expected. Also note that the 1.710882 value would be the critical value establishing the rejection region in a one-tailed t-test with df=24 and $\alpha = .05$.

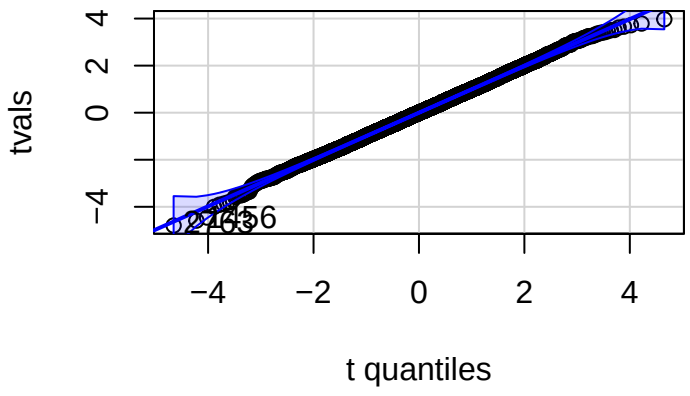
```
quantile(tvals, prob=c(.05,.95))
```

```
       5%       95%
-1.708343  1.723015
```

```
qt(c(.05,.95), df=24)
```

```
[1] -1.710882  1.710882
```

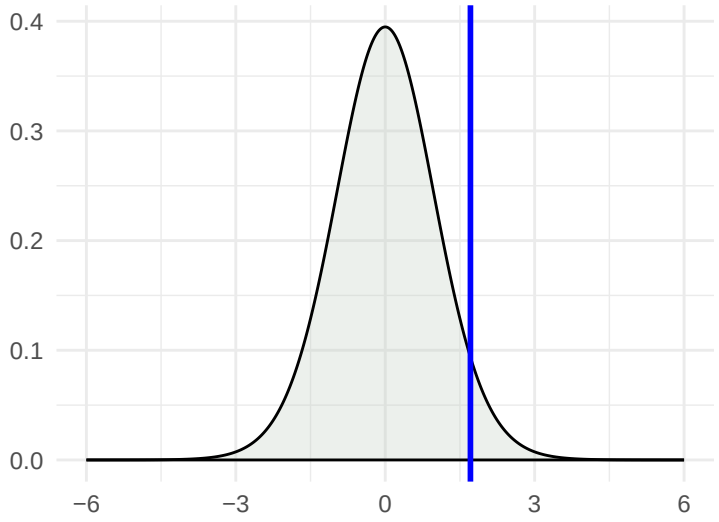
```
car::qqPlot(tvals,distribution="t",df=24) # qqPlot also returns the case numbers of the extre
```



```
[1] 2763 1456
```

Also note that the 1.710882 value seen just above would be the critical value establishing the rejection region in a one-tailed t-test with df=24, as visualized here.

```
# library(ggfortify)
p <- ggdistribution(dt, seq(-6, 6, .002), df=24, fill = 'honeydew3')
p +     geom_vline(aes(xintercept=qt(.05,df=24,lower.tail=F)),
                   linetype="solid", linewidth=1, colour="blue") +
        theme_minimal()
```



5.2 Visualizing the test statistic distribution for a One-sample t-test of a mean. Part II: Null hypothesis False

Next we will repeat the same type of simulation of the one-sample t-test, assuming that the null is false. For our purposes here let's use the same alternative population mean that we did in the Z_M simulation above ($\mu_1 = 109$). Note that 109 is three standard errors above the null hypothesis mean of 100 (recalling that the theoretical standard error is coincidentally also 3).

```
# establish basic characteristics of the sampling situation
n <- 25      # sample size
mu <- 109  # true mean of the population distribution of X (null is true)
sigma <- 15  # true SD of the population distribution of X
```

```
mu0 <- 100   # mean under the null hypothesis
reps <- 10000 # number of simulations

## initialize a few vectors:
#xvals <- as.matrix(NA,ncol=n,nrows=reps)
xbars <- numeric(reps)
stdevs <- numeric(reps)
tvals <- numeric(reps)
pvalues <- numeric(reps)

set.seed(14) # for reproducibility
for (i in 1:reps) {
  x <- rnorm(n, mu, sigma)
  xbars[i] <- mean(x)
  stdevs[i] <- sd(x)
  tvals[i] <- (mean(x) - mu0)/(sd(x)/sqrt(n))
  pvalues[i] <- 2*(1 - pt(abs(tvals[i]), n-1))
  # alternatively: pvalues[i] <- t.test(x, mu = mu0)$p.value
}
headTail(round(xbars, 2)) # round to two decimal places
```

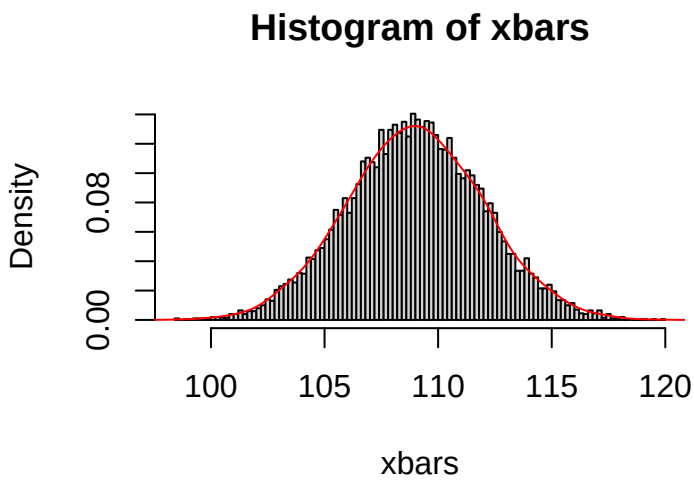
```
  [,1]            [,2]            [,3]            [,4]
h "116.81"        "107.31"        "107.27"        "107.2"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "110.39"        "114.62"        "109.65"        "113.8"
```

```
headTail(round(tvals, 2)) # round to two decimal places
```

```
  [,1]            [,2]            [,3]            [,4]
h "6.37"          "2.79"          "2.66"          "2.87"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "3.18"          "5.1"           "3.5"           "4.95"
```

We would no longer expect the simulated sampling distribution of the mean to be centered at 100. It is close to the expected value of 109, and still normal in shape. It also has the expected standard deviation of 3.

```
#hist(pvalues,breaks=100,prob=T)
#lines(density(pvalues), col="red")
hist(xbars, breaks=100, prob=TRUE)
lines(density(xbars, bw=.6), col="red")
```

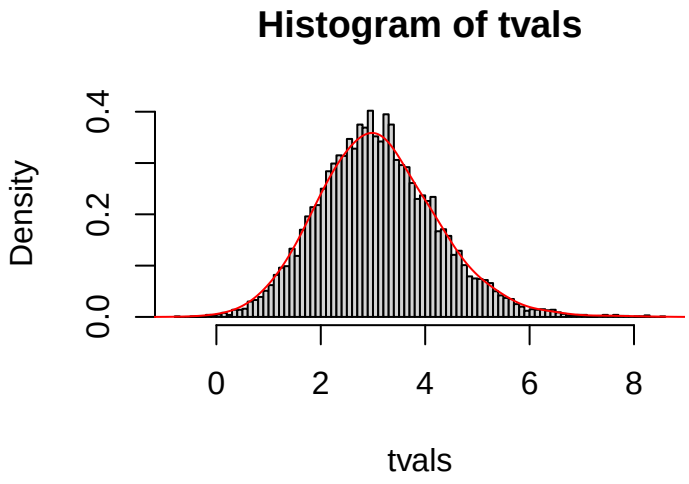



```
d1 <- describe(xbars)
d1b <- d1[,c(3,4,11)]
gt(d1b)
```

mean	sd	skew
108.9927	2.984081	0.01455882

Next we examine the simulated sampling distribution of the t statistics. They should be located at about 3 standard errors above the null value of 100 as discussed above, and they are (mean is about 3). But a close examination of the plot and results here suggests that the simulated sampling distribution is no longer symmetrical. Some positive skewness can be seen in the histogram/density plot and the skewness coefficient is a positive value somewhat deviant from zero.

```
hist(tvals, breaks=100, prob=TRUE)
lines(density(tvals, bw=.3), col="red")
```

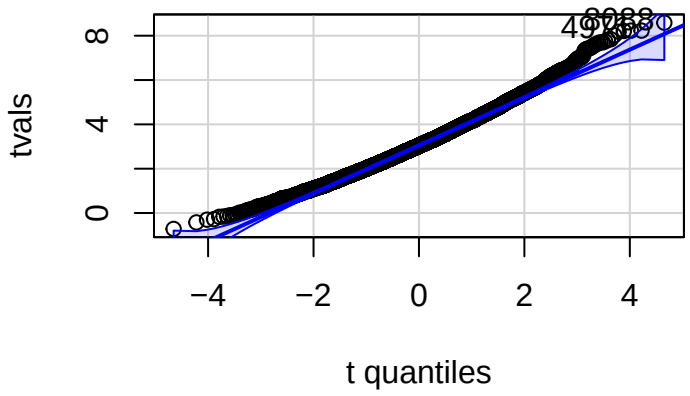



```
d2 <- describe(tvals)
d2b <- d2[,c(3,4,11)]
gt(d2b)
```

mean	sd	skew
3.094814	1.136048	0.4008286

The positive skewness of this simulated sampling distribution can also be revealed with the qqplot. The somewhat concave shape of the 10000 points in the plot reflects this skewness.

```
qqPlot(tvals,distribution="t",df=24) # qqPlot also returns the case numbers of the extreme ca
```




```
[1] 8088 4971
```

At this point, we can conclude that the sampling distribution of the test statistic behaves differently under the alternative than under the null. This is not simply a matter of a shift in its location, but it's shape is no longer symmetrical and thus does not follow a t distribution. Since Type II errors and power are found as regions of the alternative distribution, their calculation is problematic if there is not a knowable probability distribution that guides their calculation. But before taking that step, lets change a couple of aspects of the sampling simulation so that the asymmetry of the distribution is more clear.

The degree of positive skewness depends on a two characteristics, one of which is the df in the situation. So in this next simulation we will set n=9 rather than the 25 used above; df is now 8.

In order to make the arithmetic more direct/visible, another aspect of the initial population distribution of the RV (X) is also changed. Lets establish the population standard deviation (σ_X) as equal to 12.0, keeping the null hypothesis mean at 100. Let's also establish the alternative hypothesis mean as 112. With these characteristics, we would expect the sampling distribution of the t test statistic (under the alternative) to center at about 3.0 since 112 is 3 standard errors above 100 (the std error of the mean is now 12/3 or 4.0). This shift of 3 units is what we will call the non-centrality parameter, described in more detail below. The greek letter lambda (λ) is often used for this quantity. The value of lambda is the second characteristic that determines the degree of skewness of the distribution.

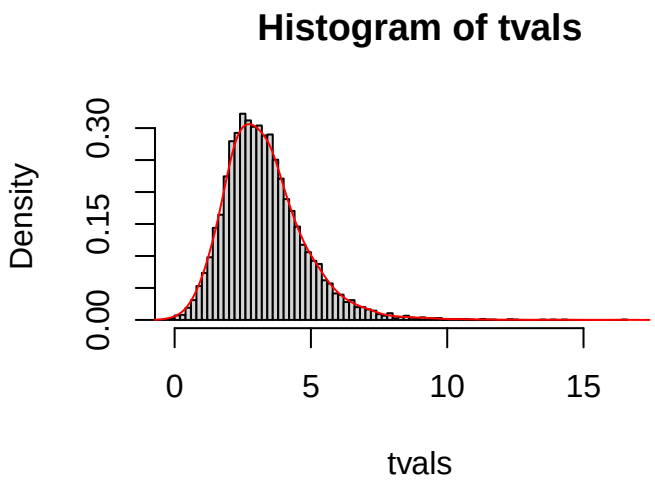
```r
# establish basic characteristics of the sampling situation
n <- 9      # sample size
mu <- 112  # true mean of the population distribution of X (null is true)
sigma <- 12  # true SD of the population distribution of X
mu0 <- 100   # mean under the null hypothesis
reps <- 10000 # number of simulations

## initialize a few vectors:
#xvals <- as.matrix(NA,ncol=n,nrows=reps)
xbars <- numeric(reps)
stdevs <- numeric(reps)
tvals <- numeric(reps)
pvalues <- numeric(reps)

set.seed(111) # for reproducibility
for (i in 1:reps) {
  x <- rnorm(n, mu, sigma)
  xbars[i] <- mean(x)
  stdevs[i] <- sd(x)
  tvals[i] <- (mean(x) - mu0)/(sd(x)/sqrt(n))
  pvalues[i] <- 2*(1 - pt(abs(tvals[i]), n-1))
  # alternatively: pvalues[i] <- t.test(x, mu = mu0)$p.value
}
psych::headTail(round(tvals, 2)) # round to two decimal places
```

```
  [,1]            [,2]            [,3]            [,4]
h "1.13"          "3.14"          "3.18"          "0.91"
  "..."       ..." "..."       ..." "..."       ..." "..."       ..."
t "5.27"          "1.88"          "3.32"          "2.21"
```

```r
hist(tvals, breaks=100, prob=TRUE, xlim=c(0,18))
lines(density(tvals, bw=.3), col="red")
```




```
d2 <- describe(tvals)
d2b <- d2[,c(3,4,11)]
gt(d2b)
```

mean	sd	skew
3.304349	1.479758	1.198309

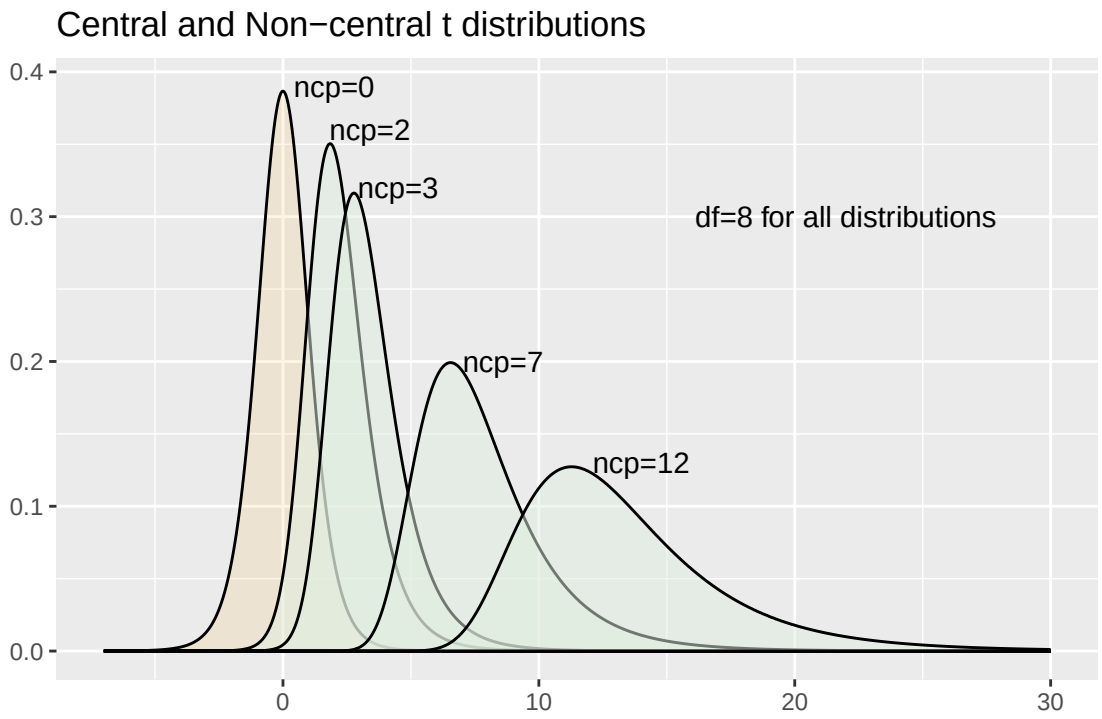
The degree of skewness is now more visible in the histogram/density plot, and the skewness coefficient for our simulated distribution is over 1.0. Notice that the distribution is shifted above where it was (zero) for the sampling situation when the null was specified as true. It is close to the value of 3 discussed above as what the shift was expected to be. Below, there is a discussion of why the mean of this empirical sampling distribution of the t statistic is not closer to the expecte 3.0, but for now lets assume that it is close enough.

This positively skewed empirical sampling distribution is actually best described as a non-central t distribution. The t distribution centered on zero is symmetrical for all df and it is called the central t distribution. We now see that the original family of t distributions that we have introduced previously can be called central t distributions. The characteristics of non-central t's are well understood, but for our purposes, the two primary characteristics are lambda (λ), the non-centrality parameter and the degree of skewness. The non-centrality parameter is the degree of shift that the non-central t moves away from the zero value that is the center of the central t distribution, for those same df. In our example, we expected the shift to be 3 units since 112 was 3 standard errors above the null value of 100, thus $\lambda = 3.00$.

5.2.1 Visualize Non-central t distributions

We can view several non-central t distributions at once using the `ggdistribution` function and compare to the central t distribtion. In this visualization, the df is kept at the same value as our simulation, df=8.

```
p <- ggdistribution(dt, seq(-7, 30, .002),df=8,ncp=0,fill = 'orange', alpha=.1) # the central
p2 <- ggdistribution(dt, seq(-7, 30, .002),df=8,ncp=2,fill = 'honeydew2',p=p, alpha=.5)
p3 <- ggdistribution(dt, seq(-7, 30, .002),df=8,ncp=3,fill = 'honeydew2',p=p2, alpha=.5)
p4 <- ggdistribution(dt, seq(-7, 30, .002),df=8,ncp=7,fill = 'honeydew2',p=p3, alpha=.5)
p5 <- ggdistribution(dt, seq(-7, 30, .002),df=8,ncp=12,fill = 'honeydew2',p=p4, alpha=.5)
p5 +  annotate("text", label = "ncp=2", x = 3.4, y = .36, size = 4, colour = "black") +
  annotate("text", label = "ncp=3", x = 4.5, y = .32, size = 4, colour = "black") +
  annotate("text", label = "ncp=7", x = 8.6, y = .2, size = 4, colour = "black") +
  annotate("text", label = "ncp=12", x = 14, y = .13, size = 4, colour = "black") +
  annotate("text", label = "ncp=0", x = 2, y = .39, size = 4, colour = "black") +
  annotate("text", label = "df=8 for all distributions", x = 22, y = .3, size = 4, colour = "
  labs(title="Central and Non-central t distributions")
```

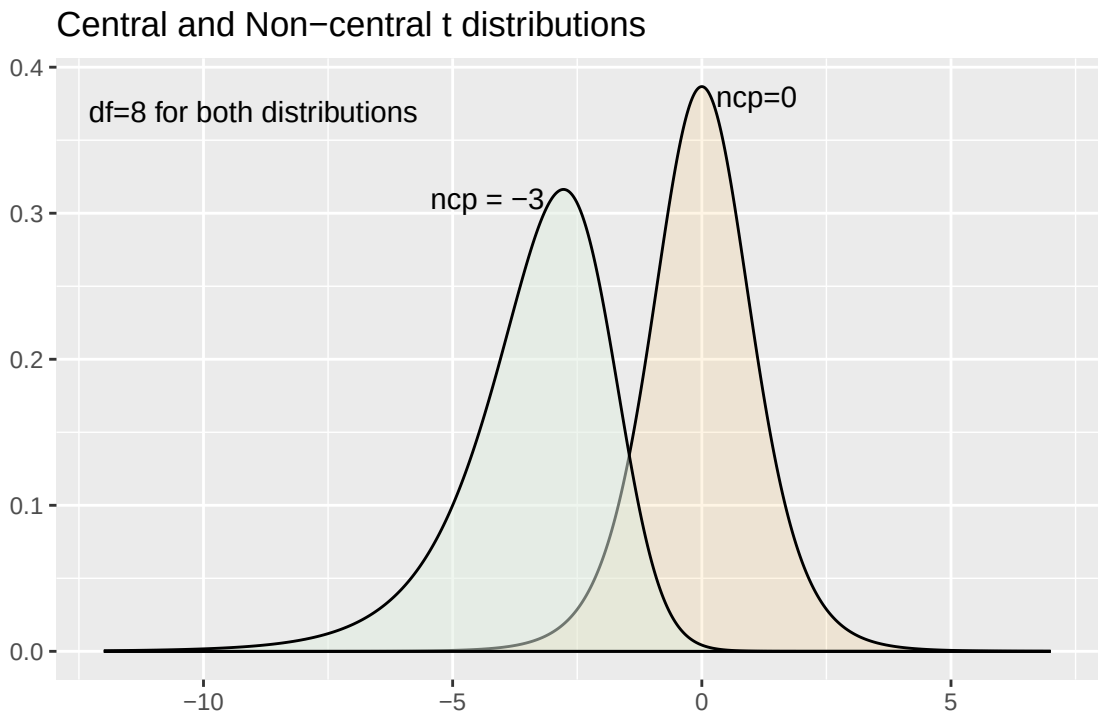


Intuitively, the mean of any non-central t would be expected to be equal to the non-centrality parameter. But our simulation showed that the mean of our 10000 simulated t values, under

the alternative of 112, was slightly above 3.00. This characteristic of non-central t's is known. The larger the df, the closer the mean of a non-central t will be to λ. For small df, the discrepancy can be substantial. Hogben, et al (1961) have derived the exact value of moments of a non-central t distribution for varying df. For our 8 df model, a correction factor of 1.10 can be applied. Multiplying the lambda by this correction (3*1.10778) gives a value of 3.32 which is very close to the mean of our simulated sampling distribution of the t statistic under the alternative as calculated above.

Finally, realize that the example/figure here only depicted non-central t distributions where λ was positive. This is because our illustration used a directional alternative in the upper tail, and thus was a one-tailed test/question. Alternatives where the mean can be lower than the null hypothesis mean would lead to λ values that would be negative and the skewness of those non-central t's would be negative, as illustrated here:

```
p <- ggdistribution(dt, seq(-12, 7, .002),df=8,ncp=0,fill = 'orange', alpha=.1) # the central
p2 <- ggdistribution(dt, seq(-12, 7, .002),df=8,ncp=-3,fill = 'honeydew2',p=p, alpha=.5)
p2 +  annotate("text", label = "ncp = -3", x = -4.3, y = .31, size = 4, colour = "black") +
    annotate("text", label = "ncp=0", x =1.1, y = .38, size = 4, colour = "black") +
    annotate("text", label = "df=8 for both distributions", x = -9, y = .37, size = 4, colou
 labs(title="Central and Non-central t distributions")
```



5.3 The non-central t as a probability distribution.

The purpose of this simulation and the introduction of the non-central t distribution was to find a way to understand Type II error and power concepts in this one sample test of a mean when the population variance is not known. In order to do this, we can use R's capability to provide probability information on non-central t's. We can use the same suite of probability functions that we have already used with the t distribution (`pt` and `qt`). We just need to add an argument that specifies λ (the argument is called ncp). For example, to find the quantile that specifies the upper 5% of a non-central t, we can use `qt`. Use the non-central t visualizations above (the one with positive ncp values) to compare (the ncp=12 distribution is the one farthest to the right in the plot with positive values of λ shown above). The `qt` function can take an argument for the non-centrality parameter and otherwise is used as we have previously. This calculated value appears to be consistent with the visualization - about five percent of the non-central t curve (ncp=12) sits above 20.83.

```
qt(.05, df=8, ncp=12,lower.tail=F)
```

```
[1] 20.82887
```

And finding the probability of a (lower tail) region less than 9 with `pt` works analogously.

```
pt(9,df=8,ncp=12,lower.tail=T)
```

```
[1] 0.0931195
```

5.4 Visualization of Type II error regions with the non-central t distribution in the df=8 example

We can now take the final step and find the expected Type II error rate in our simulated situation. For a one-sample test of the mean, with a one-tailed test in the upper region, we specified:

$H_0 : \mu = 100$

$H_1 : \mu = 112$

$\sigma_X = 12$

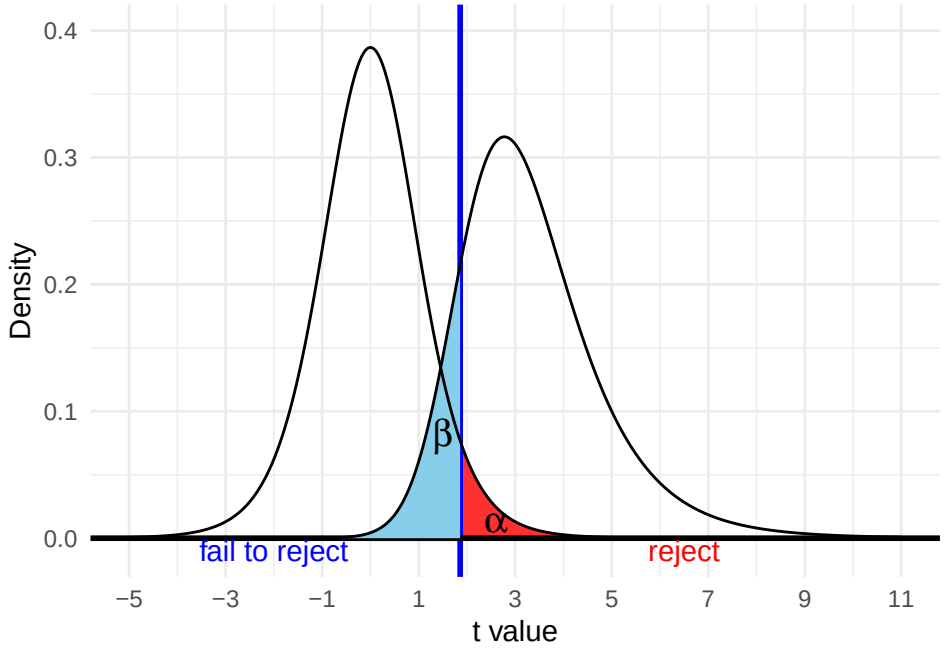
n=9

and let's set $\alpha = .05$

Note that the alternative mean of 112 is three standard errors above the null mean (std error is four - 12/3), so a ncp value of 3 is employed.

By overlaying the non-central t with the central t, and adding the location of the critical value establishing the region of rejection, we can visualize β (the Type II error rate) as an area under the non-central t distribution (the blue area). This plot is the familiar way of simultaneously viewing Type I and II error regions in the one-tailed test.

```
x <- seq(-5, 11, .002)
y0 <- dt(x,df=8)
y1 <- dt(x,df=8,ncp=3)
dat <- cbind.data.frame(x,y0,y1)
p3 <- ggplot(dat, aes(x = x, y = y0)) +
  #geom_line() +
  geom_area(mapping = aes(x = ifelse(x>qt(.05,df=8,lower.tail=F), x,
  xlim(-8,16) + ylim(-.01,.4) +
  geom_vline(aes(xintercept=qt(.05,df=8,lower.tail=F)),
             linetype="solid", size=1, colour="blue") +
#  geom_segment(aes(x = qt(.05,df=3,lower.tail=F), y = 0,
#                   xend = qt(.05,df=3,lower.tail=F), yend = .04, colour = "blue"),data=dat)
  geom_hline(aes(yintercept=0),
             linetype="solid", size=1, colour="black") +
  annotate("text", label = "fail to reject", x = -2, y = -.01, size = 4, colour = "blue") +
  annotate("text", label = "reject", x = 6.5, y = -.01, size = 4, colour = "red") +
  annotate("text", label = expression(alpha), x = 2.6, y = .012, size = 5, colour = "black")
  scale_x_continuous(breaks=seq(-5,11,2)) +
  labs(x="t value", y="Density") +
  theme_minimal()
p3 +
  geom_area(mapping = aes(x = ifelse(x<qt(.05,df=8,lower.tail=F), x, 0), y=y1), fill = "skybl
  geom_line(mapping=aes(x=x,y=y1)) +
  geom_line(mapping=aes(x=x,y=y0)) +
  annotate("text", label = expression(beta), x = 1.5, y = .08, size = 5, colour = "black")
```



We can also compute the Type II probability exactly, using the critical value that is established under the null hypothesis central t distribution. Visually, this computation looks correct - about 14% of the non-central t curve falls below the CV (the blue area).

```
pt(qt(.05,df=8, lower.tail=F), df=8,ncp=3, lower.tail=T)
```

```
[1] 0.1381863
```

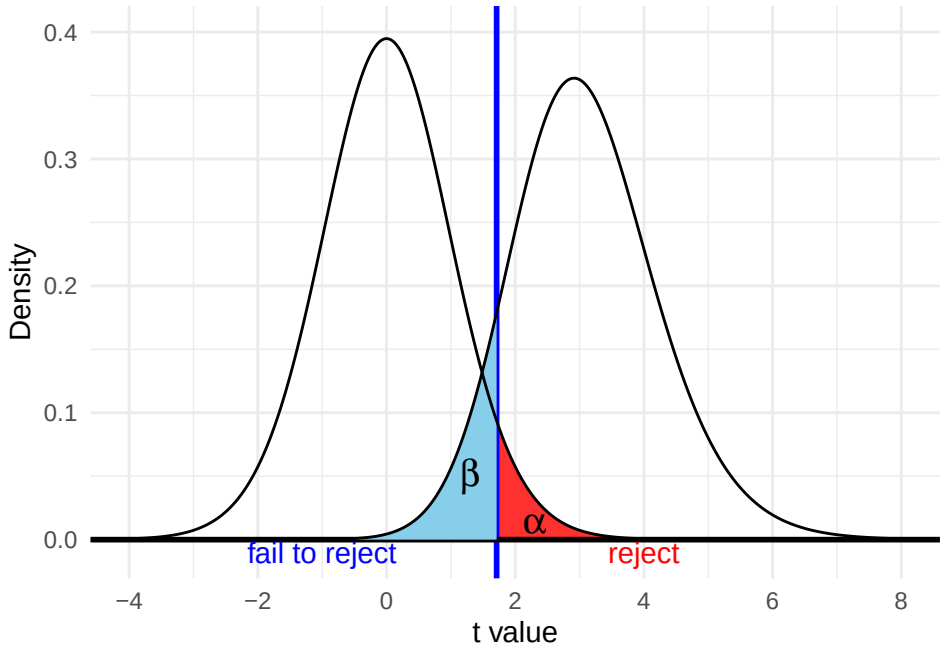
5.5 Visualization of Type II error regions in the original df=24 example

We originally did Z_M and one-sample t-test simulations with an n=25 sampling situation with different alternative means (109) and standard deviations (15). The switch to the smaller sample size was done so that the skewness of the non-central t simulation could be better visualized. But now we can return to the earlier example characteristics and find β using the tools developed just above. With these parameters, the mean of 109 is 3 standard errors of the mean above the null value of 100. Therefore, λ=3.00.

First, the visualization:

```
x <- seq(-4, 8, .002)
y0 <- dt(x,df=24)
y1 <- dt(x,df=24,ncp=3)
```

```r
dat <- cbind.data.frame(x,y0,y1)
p3 <- ggplot(dat, aes(x = x, y = y0)) +
  #geom_line() +
  geom_area(mapping = aes(x = ifelse(x>qt(.05,df=24,lower.tail=F), x,
  xlim(-8,16) + ylim(-.01,.4) +
  geom_vline(aes(xintercept=qt(.05,df=24,lower.tail=F)),
             linetype="solid", size=1, colour="blue") +
#  geom_segment(aes(x = qt(.05,df=3,lower.tail=F), y = 0,
#                   xend = qt(.05,df=3,lower.tail=F), yend = .04, colour = "blue"),data=dat)
  geom_hline(aes(yintercept=0),
             linetype="solid", size=1, colour="black") +
  annotate("text", label = "fail to reject", x = -1, y = -.01, size = 4, colour = "blue") +
  annotate("text", label = "reject", x = 4, y = -.01, size = 4, colour = "red") +
  annotate("text", label = expression(alpha), x = 2.3, y = .012, size = 5, colour = "black")
  scale_x_continuous(breaks=seq(-4,8,2)) +
  labs(x="t value", y="Density") +
  theme_minimal()
p3 +
  geom_area(mapping = aes(x = ifelse(x<qt(.05,df=24,lower.tail=F), x, 0), y=y1), fill = "skyl
  geom_line(mapping=aes(x=x,y=y1)) +
  geom_line(mapping=aes(x=x,y=y0)) +
  annotate("text", label = expression(beta), x = 1.3, y = .05, size = 5, colour = "black")
```



The exact value of β is found again with `pt` by passing the critical value, based on the null, as the first argument. The returned value of about .10 is visualized as the blue area in the above figure. Power in this situation would be approximately 90%.

```
pt(qt(.05,df=24,lower.tail=F),df=24,ncp=3, lower.tail=T)
```

```
[1] 0.1022361
```

6 Why is the non-central t asymmetrical?

The short, less mathematical, answer to the question of the origin of the asymmetry of the non-central t is that it derives from the fact that the t is a ratio of one RV (sample mean) to another RV (sample standard deviation) - each adjusted by a constant (μ_0) and n, respectively. The Z_M is the ratio of a random variable (sample mean) to a constant, so its distribution would follow the shape of the RV (the sample mean) as we saw in the simulation. But for the t statistic, the numerator is normally distributed (sampling distribution of the mean) but the denominator is not. We have covered the idea that the sample variance can be seen as a Chi-square derived value and Chi-squared distributions are positively skewed (moreso for lower df). So, the sample standard deviation is also positively skewed (especially for low df) as we visualized in a shiny app. So one of the reasons for the asymmetry comes from the math of ratios of normal RV's to non-normal RV's. But a second reason comes from the fact that the actual mean of a non-central t is not exactly equal to the value of λ, and this discrepancy is larger for lower df and larger λ (this was addressed above). These two things produce the skewness of the non-central t.

7 Non-centrality parameter, effect size and t statistic

In order to extend fully our understanding of power for purposes of sample size planning, and to incorporate the role of the non-centrality parameter, we also need to connect these concepts to the concept of an effect size. In order to accomplish this, let's revisit the non-centrality parameter more formally. Although some treatment of this is in the Howell (2013) textbook, this section is heavily influenced by Cumming and Finch (2001). In sections above, we discussed how when an alternative hypothesis is correct, that we would expect the sampling distribution of our one-sample test statistic to be shifted away from the zero-centered value of the central t distribution. The amount of this shift was described as lambda and shown with simulation. We can see this in the relevant expressions. The test statistic for the one-sample t-test is:

$$\frac{\bar{X} - \mu_0}{s_x / \sqrt{n}}$$

.

When the null is true, $\bar{X}$ deviates above and below the null mean in such as way that the expected value of the test statistic is zero. But when an alternative hypothesis is true, $\bar{X}$ deviates from the specified μ_0 with larger deviations and the sampling distribution of the test statistic will deviate above or below the zero value expected under the null. The amount of this shift depends on the true value of the population mean. If the true value of μ is equal to an alternative hypothesis mean (μ_1) then we can rewrite the numerator of the test statistic by breaking the deviation into two components:

$$\left(\bar{X} - \mu_1\right) + (\mu_1 - \mu_0)$$

This reflects the fact that the total deviation from the null mean is composed of a deviation of the sample mean from the true population mean plus a component that reflects the shift of the sampling distribution from a center of μ_0 to a center of μ_1. So the test statistic now can be rewritten to reflect this:

$$\frac{\left(\bar{X} - \mu_1\right) + (\mu_1 - \mu_0)}{s_x / \sqrt{n}} \sim t_{n-1,\lambda}$$

What this expression tells us is that our test statistic will be distributed as a non-central t with n-1 df, and with a non-centrality parameter of λ. The non-centrality emerges from the amount of shift described as the second phrase in the numerator $(\mu_1 - \mu_0)$ but lambda is actually this deviation standardized relative to the true standard error in the system:

$$\lambda = \frac{\mu_1 - \mu_0}{\sigma_x / \sqrt{n}}$$

With that simulation and formal defninitions of lambda now in place we need to review the effect size concept as the next step to using power to plan sample sizes. In prior work, we had introduced the standardized effect size statistic called Cohen's d in the two-sample location or independent samples t-test context. There is an analogous effect size statistic for the one-sample test emphasized above throughout this document. It simply indexes the discrepancy of the observed sample mean from the null hypothesis mean relative to the variation of the variable (as a standard deviation):

$$d = \frac{\bar{X} - \mu_0}{s_x}$$

This sample statistic should be seen as an estimate of the true effect size if some alternative hypothesis mean were the true mean. This population parameter takes several different symbols in different texts, but we will use the greek lower case delta here:

$$\delta = \frac{\mu_1 - \mu_0}{\sigma_x}$$

Since this δ expression has most of the same quantities as the expression for λ, we can see that the relative size of the two is dependent on a scalar that is the square root of sample size:

$$\delta = \frac{\lambda}{\sqrt{n}} \ and \ \lambda = \delta\sqrt{n}$$

Note that these are defined as "true" population quantities and are thus constants, even though these are theoretical population distributions. At this point we can see that both λ and δ are important components of power determinations and their distinction is the sample size quantity that we would like to solve for in planning research studies.

We saw with the simulations above that the true λ in a population system drives the expected value of the t test statistic. It should not be surprising that there is also a relationship between the observed t and Cohen's d:

$$d = \frac{t_{obs}}{\sqrt{n}}$$

This is a slightly different relationship than we saw for Cohen's d and the observed t for the two-sample t-test, but it is only because of the number of samples taken.

In the figures above where overlapping central and non-central t distributions were drawn to depict the Type I and II areas, the drawings could only be done by specifying n (and in turn, df). But if n is unknown and the value that we are seeking prospectively, then it is not obvious how to calculate it such that the combination of δ, α, and β parameters all fit the situation. In order to choose sample size, based on a fixed α, and a chosen power $(1 - \beta)$ we realize that need an effect size estimate. With those things in place (and the algebraic relationships above) it is possible to calculate precisely the sample size required by using the non-central t distribution as depicted in the above figures showing the overlapping central and non-central t distributions. For any a priori choice of α and β, there is only one sample size (therefore df) that locates the critical value in the correct place. Since λ depends on df (n) the simplest way to find this unkown n is via successive approximations or iterations. There are mathematical and computer algorithms that do this efficiently and fortunately, software implements this process quickly, as described in the next section.

7.0.1 A Note on Notation

Statistics texts and articles are notorious for using multiple different notation conventions. In this document, we have used d as the notation for the sample effect size, and that is fairly standard (Cohen's d). It is also somewhat common for the population parameter that is estimated by Cohen's d to be called δ. But the notation for the non-centrality parameter is not at all standardized. In this document, the ncp is called λ but this is not necessarily the most common labeling. Sometimes the ncp is called δ and sometimes it is called Δ. Other sources use $\delta^{'}$. Confusion is likely unless one is aware.

8 Using power software

Using power functions and power software to choose sample sizes in the planning stages of experimental design makes quick work of the concepts developed above and may even obscure the role that is played by the non-central distribution. Illustrating a few functions in R plus G Power is done here for the one-sample t-test emphasized above in this document, but it can serve as a model for how those approaches can also be used for all of the other basic tests encountered in the 510/511 courses.

8.1 Power facilities in R

In R, the base system package called **stats** has a set of functions that do power analysis: `power.prop.test` for proportions, `power.anova.test` for ANOVA designs, and `power.t.test` for t-tests of means. Power and sample size analyses for both one and two sample t-tests can be done as well as the paired samples (dependent samples) t-test can be done, as well as one-tailed or two tailed versions. The function takes arguments of n, delta (effect size as a standardized mean difference such as Cohen's d), alpha, power, and an estimate of the population standard deviation. If one of these parameters is left out or set to null, then the function solves for it, based on the information from the others. Here, we ask for sample size for our one-sample study based on a moderate sized Cohen's d, a directional alternative and power of .8.

```
d<-.5
power.t.test(d=d,sig.level=0.05,type="one.sample",alternative="one.sided", power=.8,strict=T
```

```
     One-sample t test power calculation

              n = 26.13751
          delta = 0.5
             sd = 1
```

```
      sig.level = 0.05
          power = 0.8
    alternative = one.sided
```

A more extensive set of power-related functions can be found in the **pwr** package. For our one-sample t-test situation a similar set of arguments is passed to the `pwr.t.test` function as was the case for `power.t.test` shown above.

```
d<-.5
pwr.t.test(d=d,sig.level=0.05,type="one.sample",alternative="greater", power=.8)
```

```
     One-sample t test power calculation

              n = 26.13753
              d = 0.5
      sig.level = 0.05
          power = 0.8
    alternative = greater
```

Before going on to GPower, let's use `pwr.t.test` for a two-sample (independent samples) t-test sample size choice. The results here can be compared to a spreadsheet-computed set of values shown when power was first introduced earlier in the course.

```
pwr.t.test(d=.75, sig.level=.05, type="two.sample", alternative="two.sided", power=.9)
```

```
     Two-sample t test power calculation

              n = 38.34604
              d = 0.75
      sig.level = 0.05
          power = 0.9
    alternative = two.sided

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

Here is a snippet from that spreadsheet showing the sample size calculation for the same set of parameters. Note that the sample size estimate is slightly lower than the one from `pwr.t.test`.

This is because the earlier spreadsheet approach is an approximation and did not use the non-central t distribution to calculate the Type II error exactly. Instead, a standard normal was used as an approximation. For small df, the approximation is poorer.

13										
14	**mu1-mu2**	**WG SD**	**WG Var**		**varBG=**	**varTOT**	**d**	**omega squared**	**n required per group**	
15	1	4	16		0.25	16.25000	0.2500	0.015385	335.9	
16	1.25	4	16		0.39	16.39063	0.3125	0.023832	215.0	
17	1.5	4	16		0.56	16.56250	0.3750	0.033962	149.3	
18	1.75	4	16		0.77	16.76563	0.4375	0.045666	109.7	
19	2	4	16		1.00	17.00000	0.5000	0.058824	84.0	
20	2.5	4	16		1.56	17.56250	0.6250	0.088968	53.7	
21	3	4	16		2.25	18.25000	0.7500	0.123288	37.3	←
22	4	4	16		4.00	20.00000	1.0000	0.200000	21.0	
23	5	4	16		6.25	22.25000	1.2500	0.280899	13.4	
24	6	4	16		9.00	25.00000	1.5000	0.360000	9.3	

8.2 GPower application for power

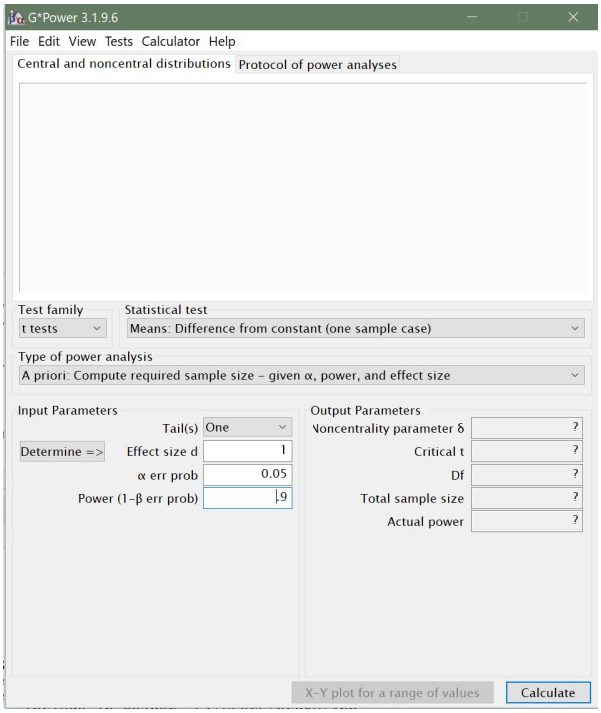
While the **pwr** package has good capability, many researchers have come to rely on GPower software. Among the nice features of GPower are a) a visualization of the central and non-central distributions along with α and β depictions, b) facilities for estimating δ and using it in the power/sample size calculations, and c) depiction of how changes in sample size influence power. GPower implements power related calculations for a wide variety of basic statistical methods.

GPower can be found at the following URL:

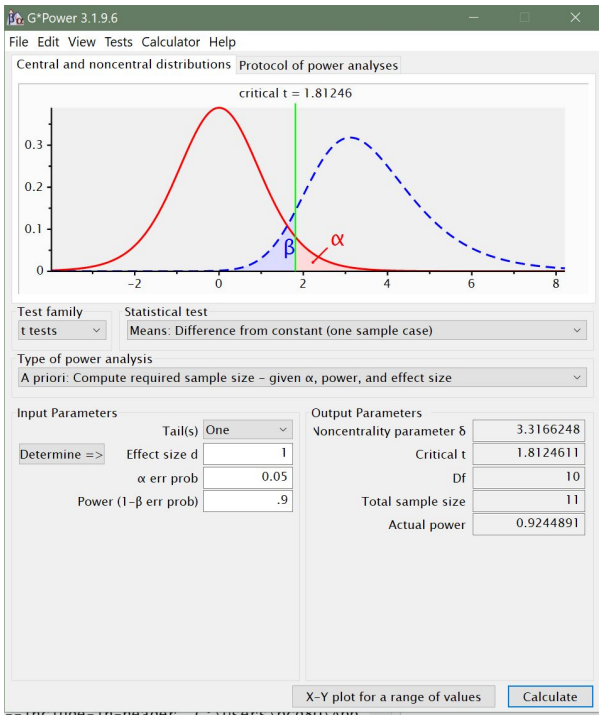
http://www.psychologie.hhu.de/arbeitsgruppen/allgemeine-psychologie-und-arbeitspsychologie/gpower.html

Initially, lets examine how to use GPower for sample size planning in our one-sample location t-test. The situation we will work with is the first t-test alternative from above where the alternative mean was 112, the null mean was 100, and the population standard deviation was 12. This means that we were expecting the effect size (δ) to be 1.0. In our example, we specified n=9 but for purposes here, lets assume that we want to have a power of .9 and need to know what sample size would produce that power (also given an alpha of .05 and a one-tailed, upper-tailed test). In that illustration above, we found that power was approximately .86, so we would expect that GPower would find a sample size not too far from n=9 would produce the specified power of .9.

GPower requires us to choose the test, set the type of analysis (we want the a priori sample size planning capability), enter the effect size as 1, α=.05, and β=.10.

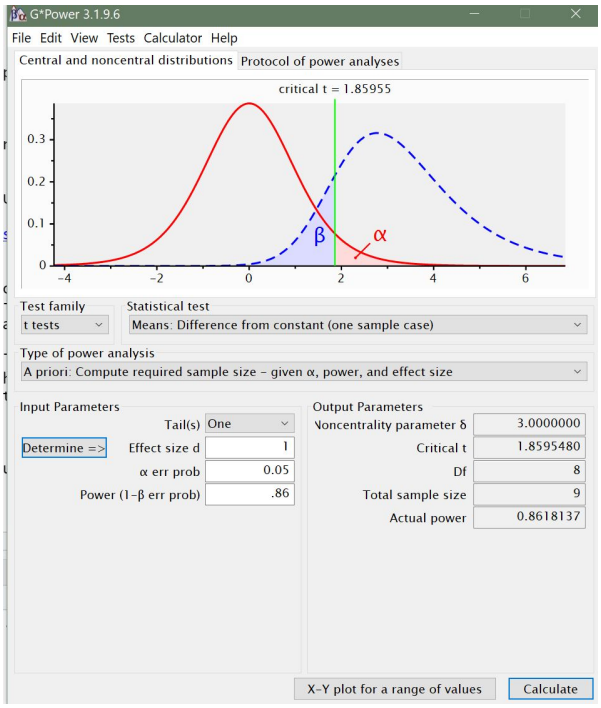

G*Power 3.1.9.6
File Edit View Tests Calculator Help
Central and noncentral distributions
Protocol of power analyses
Test family
t tests
Statistical test
Means: Difference from constant (one sample case)
Type of power analysis
A priori: Compute required sample size – given α, power, and effect size
Input Parameters
Tail(s)
One
Determine =>
Effect size d
1
α err prob
0.05
Power (1–β err prob)
,9
Output Parameters
Noncentrality parameter δ
?
Critical t
?
Df
?
Total sample size
?
Actual power
?
X-Y plot for a range of values
Calculate

Clicking on the calculate button produces the following result:



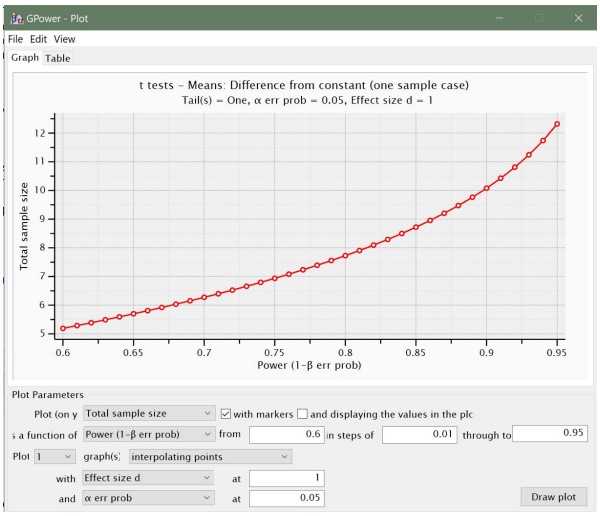
Note that GPower rounds up to whole number sample size values (n=11 here), so the power in this situation is not exactly .90 as we specified. It is the smallest power value above our specification of .90 that leads to a whole number sample size. Lambda is also slightly larger than the 3.0 that we delineated in the exposition above, since power was also larger than .90.

What if we set power to .86, the value that we calculated above when the λ =3.0?



Now we see that GPower found the sample size to be the exact value of n=9 that we used in the original demonstration where we visualized β=.14 and power=.86. The drawing produced by GPower also looks very similar to the depiction that we created above when first visualizing the non-central t distribution overlapping with the central t.

Finally, pressing the X-Y plot button yields a plot that relates power to varying sample sizes for the set of parameters specified.



This section is not intended to be a comprehensive overview of GPower. Substantial information on its usage can be found on the URL given above. Here, this quick overview was included to reinforce the concepts developed with the one-sample t-test in the simulation and narrative above.

9 The two-sample (Independent-samples) t-test of means

For the independent samples t-test, it is helpful to remember that power (and thus a priori sample size planning) is dependent on the expected form of the sampling distribution of the test statistic, under the null and under a specified alternative. From this point of view, the core concept is no different that outlined above for the one-sample situation. The sampling distribution of the test statistic, under an alternative is once again distributed as a non-central t and power if found in the analagous fashion. Therefore, in this section, we will not do the same sequence of simulations as was done above. Rather, the core test statistic expressions as well as δ and λ be reviewed as a reminder and extension of things already covered when the two-sample t-test was introduced.

The test statistic for the independent samples t-test (assuming homogeneity of poulation variances) was learned as:

$$\frac{\left(\bar{X}_1 - \bar{X}_2\right) - \left(\mu_1 - \mu_2\right)}{\sqrt{s_P^2\left(\frac{n_1+n_2}{n_1 n_2}\right)}} \sim t_{df=n_1+n_2-2}$$

,

where the second phrase in the numerator is almost always assumed to be zero, under the null, and s_P^2 is the pooled sample variance that estimates the common population variance σ_C^2, which assumes homogeneity of within-group variances ($\sigma_1^2 = \sigma_2^2 = \sigma_C^2$).

The population entity that the test statistic estimates would be:

$$\frac{\mu_1 - \mu_2}{\sqrt{\sigma_C^2\left(\frac{n_1+n_2}{n_1 n_2}\right)}}$$

and this quantity is equal to the non-centrality parameter, λ

When the null is true, the numerator is zero, and the test statistic would be distributed as a t with $df = n_1 + n_2 - 2$. When the null is false, the numerator would be non-zero and the test statistic would be distributed as a non-central t with a non-centrality parameter driven by the size of δ and it would also have $df = n_1 + n_2 - 2$.

Cohen's d for sample data is now:

$$d = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{s_P^2}}$$

The population parameter estimated by this d statistic is:

$$\delta = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_C^2}}$$

The relationship between the observed t statistic and d is:

$$t_{obs} = \frac{d}{\sqrt{\frac{n_1+n_2}{n_1 n_2}}}$$

And this reduces to

$$t_{obs} = \frac{d}{\sqrt{\frac{2}{n}}}$$

when the group sample sizes are equal (n).

λ, as defined above, can also be defined in terms of the effect size in an analogous manner to how it was defined above for the one-sample case:

$$\lambda = \frac{\delta}{\sqrt{\frac{n_1+n_2}{n_1 n_2}}}$$

With this background, informed use of either `pwr` or GPower for a priori sample size planning is just as simple as for the one-sample situation. Again, assuming a one-tailed test, and a desire for power of at least .9, with the same large effect size used for the one-sample illustration being the minimal relevant difference in means, we can use either software.

```
pwr.t.test(d=1,sig.level=.05, power=.9, type="two.sample", alternative="greater")
```

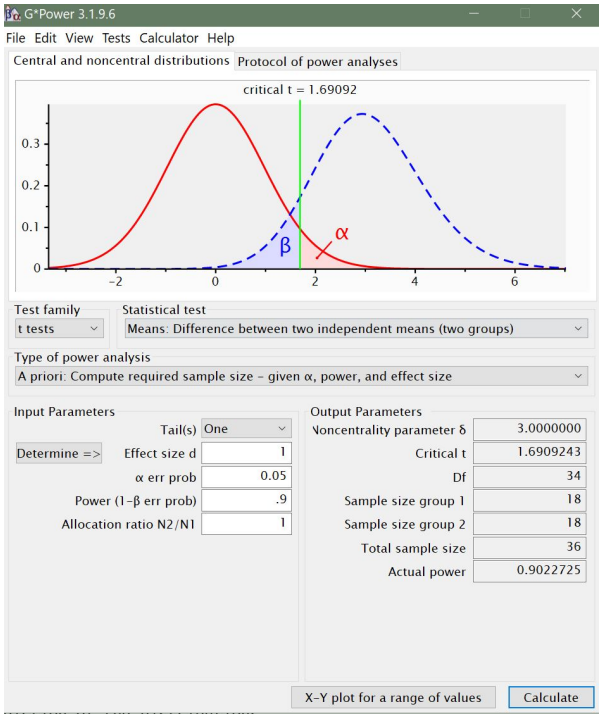
```

     Two-sample t test power calculation

              n = 17.84712
              d = 1
      sig.level = 0.05
          power = 0.9
    alternative = greater

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

Recall that GPower rounds up to the nearest whole number sample size, so compared to `pwr.t.test`, the sample sizes per group are slightly larger and power is also slightly larger than .9.



The visualization from GPower reminds us that power is found under the alternative hypothesis test statistic sampling distribution (the non-central t) and that the null hypothesis sampling distribution (central t), is used to establish the critical value.

Let's use our tools in R to verify the quantities produced by GPower for this situation. With an effect size of 1, α=.05, and df=34, we can find β:

```
# first find the critial value under the null distribution
cv <- qt(.05, df=34, lower.tail=F)
cv
```

```
[1] 1.690924
```

```
# now use that cv value to find beta under the alternative distribution
beta <- pt(cv,df=34,ncp=3,lower.tail=T)
beta
```

```
[1] 0.09772751
```

```
# and find power
1-beta
```

```
[1] 0.9022725
```

Both the critical value and the power quantity match what was found in GPower.

10 Extensions of the concept of a non-central distribution

With the basics put in place here with regard to test statistic sampling distributions and non-centrality under alternatives, we can now easily extend this understanding to a host of situations not covered in this document.

10.1 The dependent samples t-test (paired, or related measurement test)

When a two-sample location test is performed on paired samples, this is typically called the dependent-samples t-test. When we covered that test previously, we took an approach that derived difference scores for each pair of DV values. The mechanics of the test were then approached exactly as we would have done a one-sample t-test with the single sample of difference scores. It follows that Type II errors, λ, and power would be approached exactly as was outlined in the one-sample illustration above in this document.

The `power.t.test` and `pwr.t.test` functions both permit power calculation and sample size planning for the dependent samples test (called "paired" in the type argument). GPower handles the dependent samples test situation directly as well.

10.2 Two-tailed tests

For two-tailed tests, there is no additional complexity beyond what we already saw for the Z_M test in the original lecture presentation. The only item to recall is that there are now two "regions of rejection" and finding β needs to take that into account. Once the non-centrality parameter is specified, β can be found.

10.2.1 Visualization of Type II error in a two-tailed (one-sample) t-test: the df=8 example

Here, we return to the small sample size illustration and redo the simulation of the expected Type II error rate but doing a two-tailed test (still keeping α=.05). For a one-sample test of the mean, with a two-tailed test we specify:

$H_0 : \mu = 100$

$H_1 : \mu = 112$

$\sigma_X = 12$

n=9

and let's set $\alpha = .05$

The same two distributions, central and non-central t's are still germane. But since we are doing a two-tailed test, two regions of rejection exist, one in each tail. β (the Type II error rate) will exist as a probability found from the non-central t in region between these two critical values. By overlaying the non-central t with the central t, and adding the location of the critical values establishing the regions of rejection, we can once again visualize β as an area under the non-central t distribution (the blue area). This plot is the familiar way of simultaneously viewing Type I and II error regions in the two-tailed test. The impact of this second rejection region in the lower tail of the central t distribution on power/β is negligible. This is because the question of what would be expected if the alternative were 12 units above the null distribution mean results in very little of that distribution falling in the lower region of rejection - it is not even visually detectable with this plot.

First, we need to find the new critical values based on the null (from the central t):

```
qt(.025, df=8)
```

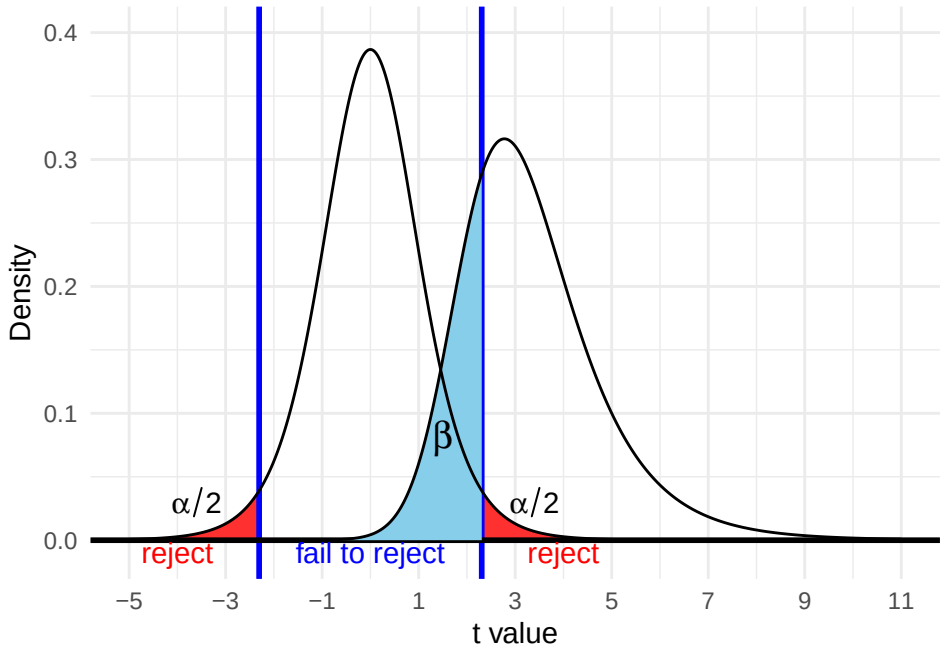
```
[1] -2.306004
```

```
qt(.025, df=8, lower.tail=F)
```

```
[1] 2.306004
```

These +/- 2.31 values can now be visualized, along with the new size of β.

```r
x <- seq(-5, 11, .002)
y0 <- dt(x,df=8)
y1 <- dt(x,df=8,ncp=3)
dat <- cbind.data.frame(x,y0,y1)
p3 <- ggplot(dat, aes(x = x, y = y0)) +
  #geom_line() +
  geom_area(mapping = aes(x = ifelse(x>qt(.025,df=8,lower.tail=F), x,
  xlim(-8,16) + ylim(-.01,.4) +
    geom_area(mapping = aes(x = ifelse(x<qt(.025,df=8), x, qt(.025,df=8) )), fill = "firebri
  xlim(-8,16) + ylim(-.01,.4) +
  geom_vline(aes(xintercept=qt(.025,df=8,lower.tail=F)),
             linetype="solid", size=1, colour="blue") +
    geom_vline(aes(xintercept=qt(.025,df=8)),
             linetype="solid", size=1, colour="blue") +
#  geom_segment(aes(x = qt(.05,df=3,lower.tail=F), y = 0,
#                   xend = qt(.05,df=3,lower.tail=F), yend = .04, colour = "blue"),data=dat)
  geom_hline(aes(yintercept=0),
             linetype="solid", size=1, colour="black") +
  annotate("text", label = "fail to reject", x = 0, y = -.01, size = 4, colour = "blue") +
  annotate("text", label = "reject", x = -4, y = -.01, size = 4, colour = "red") +
  annotate("text", label = "reject", x = 4, y = -.01, size = 4, colour = "red") +
  annotate("text", label = expression(alpha/2), x = 3.4, y = .03, size = 4, colour = "black"
  annotate("text", label = expression(alpha/2), x = -3.6, y = .03, size = 4, colour = "black
  scale_x_continuous(breaks=seq(-5,11,2)) +
  labs(x="t value", y="Density") +
  theme_minimal()
p3 +
  geom_area(mapping = aes(x = ifelse(x<qt(.025,df=8,lower.tail=F), x, 0), y=y1), fill = "sky
  geom_line(mapping=aes(x=x,y=y1)) +
  geom_line(mapping=aes(x=x,y=y0)) +
  annotate("text", label = expression(beta), x = 1.5, y = .08, size = 5, colour = "black")
```



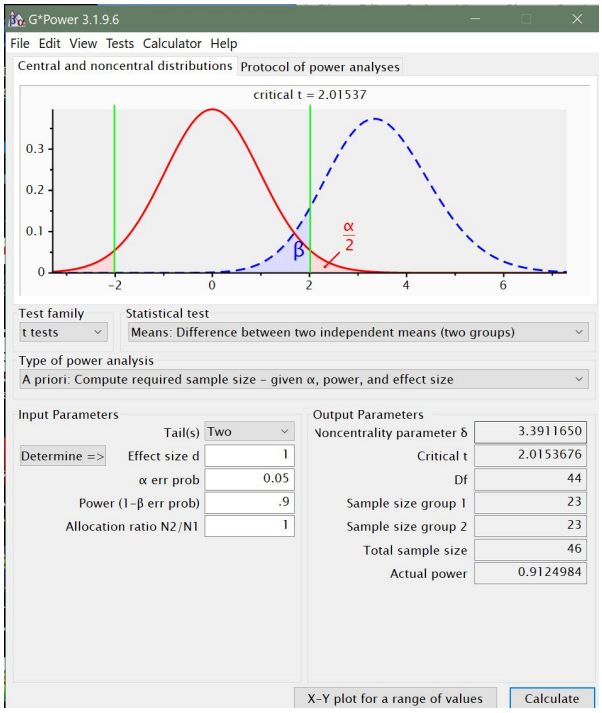
Realize that the alternative curve (the non-central t) extends to and beyond the lower critical value even though it cannot be seen in this example because the ncp, and thus the shift of the alternative distribution, is so large. β, the blue region, does not extend past this lower critical value and is thus found only between the two critical values. We can also compute the Type II probability exactly, using those critical values established under the null hypothesis central t distribution. Visually, this computation looks correct.

```
# calculate the full left tail region under the alternative and subtract the unseen part belo
pt(qt(.025,df=8, lower.tail=F), df=8,ncp=3, lower.tail=T) - pt(qt(.025,df=8), df=8,ncp=3, lou
```

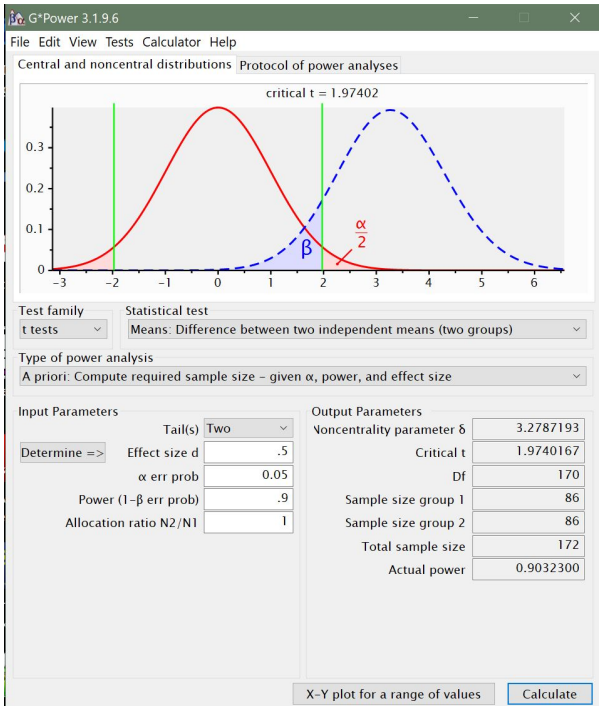
```
[1] 0.2519829
```

10.2.2 Two-tailed tests and GPower

The software applications reviewed above handle two-tailed tests with ease. GPower also provides the helpful visual depiction of the distributions. Here is a GPower result for the independent samples t-test but switching to a two-tailed test. The software was queried about sample size needed when the Cohen's d was 1.0 (a large effect size) and a power of at least .90 was desired. Note how the figure at the- top of the app panel shows both these regions of rejection, which are taken into account.



At this point it would be useful to recall the admonition that research studies are often radically under-powered. This topic has been discussed heavily in the past decade or so in the context of the "reproducibility crisis" and a large amount of that literature can be found in the toolkit bibliography. One example is Maxwell (2004). If we re-run GPower in exactly the same situation as just above, but reduce the effect size to a "moderate" value of .5, sample sizes are considerably larger than most studies employ:



10.3 Confidence Intervals

There is no need for a consideration of the non-central t distribution in computation of confidence intervals for a single sample mean or the difference between two independently drawn means. Those expressions appropriately use a critical value from central t distributions.

In contrast, confidence intervals for an effect size statistic such as Cohen's d (δ) do require a non-central t distribution. See the Cumming and Finch (2001) article.

10.4 Other tests

NHST tests for other applications such as correlation, regression, goodness of fit, ANOVA, etc most commonly use test statistics that employ the t, F, or Chi-square distributions. All have non-central complements to the central versions initially learned with these tests. The power algorithms in R and GPower handle this range of applications and more. The non-centrality idea that we developed for the t distribution is directly analogous to the non-central F and Chi-square distributions.

Finally, Bayesian Inference will require understanding of non-central distributions as well. Power, as we have addressed it here is strictly an NHST concept. But there are power topics associated with Bayesian tests and the background from this document should help facilitate that learning curve.

11 R Documentation and Reproducibility

Document version 1.2 - 11-01-24

```
sessionInfo()
```

```
R version 4.4.1 (2024-06-14 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 22631)

Matrix products: default


locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] pwr_1.3-0        gt_0.11.0        knitr_1.48       ggfortify_0.4.17
[5] car_3.1-2        carData_3.0-5    psych_2.4.6.26   ggplot2_3.5.1

loaded via a namespace (and not attached):
 [1] gtable_0.3.5      jsonlite_1.8.8    dplyr_1.1.4       compiler_4.4.1
 [5] tidyselect_1.2.1  xml2_1.3.6        stringr_1.5.1     parallel_4.4.1
 [9] gridExtra_2.3     tidyr_1.3.1       scales_1.3.0      yaml_2.3.10
[13] fastmap_1.2.0     lattice_0.22-6    R6_2.5.1          labeling_0.4.3
[17] generics_0.1.3    tibble_3.2.1      munsell_0.5.1     pillar_1.9.0
[21] rlang_1.1.4       utf8_1.2.4        stringi_1.8.4     xfun_0.46
[25] cli_3.6.3         withr_3.0.1       magrittr_2.0.3    digest_0.6.36
[29] grid_4.4.1        rstudioapi_0.16.0 lifecycle_1.0.4   nlme_3.1-165
[33] vctrs_0.6.5       mnormt_2.1.1      evaluate_0.24.0   glue_1.7.0
[37] farver_2.1.2      abind_1.4-5       fansi_1.0.6       colorspace_2.1-1
[41] purrr_1.0.2       rmarkdown_2.27    tools_4.4.1       pkgconfig_2.0.3
```

```
[45] htmltools_0.5.8.1
```

References

Champely, S. (2018). *Pwr: Basic functions for power analysis*. Retrieved from https://CRAN.R-project.org/package=pwr

Cumming, G., & Finch, S. (2001). A primer on the understanding, use, and calculation of confidence intervals that are based on central and noncentral distributions. *Educational and Psychological Measurement*, *61*, 532–574. Journal Article.

Fox, J., Weisberg, S., & Price, B. (2018). *Car: Companion to applied regression*. Retrieved from https://CRAN.R-project.org/package=car

Hogben, D., Pinkham, R. S., & Wilk, M. B. (1961). The moments of the non-central t-distribution. *Biometrika*, *48*, 465–468. Journal Article.

Horikoshi, M., & Tang, Y. (2019). *Ggfortify: Data visualization tools for statistical analysis results*. Retrieved from https://CRAN.R-project.org/package=ggfortify

Howell, D. C. (2013). *Statistical methods for psychology* (8th ed., pp. xix, 770 p.). Book, Belmont, CA: Wadsworth Cengage Learning.

Iannone, R., Cheng, J., & Schloerke, B. (2019). *Gt: Easily create presentation-ready display tables*. Retrieved from https://github.com/rstudio/gt

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods*, *9*, 147–163. Journal Article.

Revelle, W. (2019). *Psych: Procedures for psychological, psychometric, and personality research*. Retrieved from https://CRAN.R-project.org/package=psych

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., & Woo, K. (2018). *ggplot2: Create elegant data visualisations using the grammar of graphics*. Retrieved from https://CRAN.R-project.org/package=ggplot2

Xie, Y. (2018). *Knitr: A general-purpose package for dynamic report generation in r*. Retrieved from https://CRAN.R-project.org/package=knitr