

Lab 4

T-Tests: Theoretical Aspects
Psychology 310

Instructions. Work through the lab, saving the output as you go. If you work in Microsoft Word, you can easily copy any graph to Word via the clipboard. Numerical output may also be copied easily by highlighting, moving it to the clipboard, then copying into Word. However, you should format R output in TrueType Courier New font so that it is *monospaced*. Output from this lab is to be handed in by Monday, November 15. Your output file should be named `LAST_FIRST_LAB4.DOC`, where `LAST` is your last name, and `FIRST` is your first name. Any additional files should have the same naming scheme, except the file extension should be correct. You may add any description text you wish after `LAB4` in the file name.

Preamble. Today's assignment involves looking at the distribution of the sample mean, using both theoretical results and simulation. A related homework assignment will deal with the more practical aspects of using *t*-tests on experimental data.

1 The Distribution of the Sample Mean

Tests on means are built on the assumption that the sample mean \bar{X}_n is based on N independent observations from a population with mean μ and variance σ^2 . From linear combination theory, we have derived that, so long as the N observations are independent, \bar{X}_n will have a mean of μ and a variance of σ^2/N .

What about the shape of the sampling distribution?

If, in addition, we can specify that the population distribution is exactly normal, then the distribution of \bar{X}_n will be exactly normal.

However, even if the population distribution is *not* normal, we can often act as if it is without introducing serious error, because of the *Central Limit Theorem (CLT)*. The CLT implies that, under rather general conditions, the sample mean will have an asymptotically normal distribution even when the population distribution is not normal.

With any asymptotic distributional result, a key question is *rate of convergence*, that is, how fast the actual distribution of the statistic converges to the asymptotic result. If convergence is slow, the distribution of \bar{X}_n may

not be close to a normal distribution at the sample sizes we wish to use. If convergence is fast, the distribution of \bar{X}_\bullet may be quite close to a normal distribution at moderate sample sizes.

1.1 Simulating the Sampling Distribution

We can easily use R to simulate sampling distributions. Start off by setting the random number seed to 12345, so we all get the same numbers.

```
> set.seed(12345)
```

To create a random sample of size $N = 25$ from a normal distribution with mean $\mu = 100$ and standard deviation $\sigma = 15$, use the command

```
> data <- rnorm(25,100,15)
```

These data are a simulated sample from a population with known characteristics. Of course, the sample estimates from this sample will not be equal to the population quantities they are estimating. In this case,

```
> xbar <- mean(data)
```

```
> s <- sd(data)
```

```
> xbar
```

```
[1] 99.98234
```

```
> s
```

```
[1] 14.17671
```

By chance, \bar{X}_\bullet was quite close to the actual μ . The sampling error was only

```
> sampling.error <- xbar - 100
```

```
> sampling.error
```

```
[1] -0.0176593
```

Because we can create our own statistical universe, with known properties, in this case we know precisely what sampling error was. Of course, in the real world, we will not know μ , and we will not know the sampling error. Instead, we will have to work with long run probabilities.

We already know from our work in Psychology 310 lectures that, when the population is normal, $\mu = 100$, $\sigma = 15$, and $N = 25$, the sample mean \bar{X}_\bullet has a normal distribution with mean 100 and standard deviation $\sigma_{\bar{X}_\bullet} = 3$.

Let's verify that with a statistical simulation.

The following code will take a sample of size 25 and compute the sample mean. (Notice how I reset the seed, in case we somehow got out of synch.)

```
> set.seed(12345)
> mean(rnorm(25,100,15))
[1] 99.98234
```

I could do this 10 times using the `replicate` command.

```
> set.seed(12345)
> replicate(10,mean(rnorm(25,100,15)))
[1] 99.98234 105.40465 102.95220 106.37264 103.18856 96.42129
[7] 99.03329 104.07084 98.52455 103.15902
```

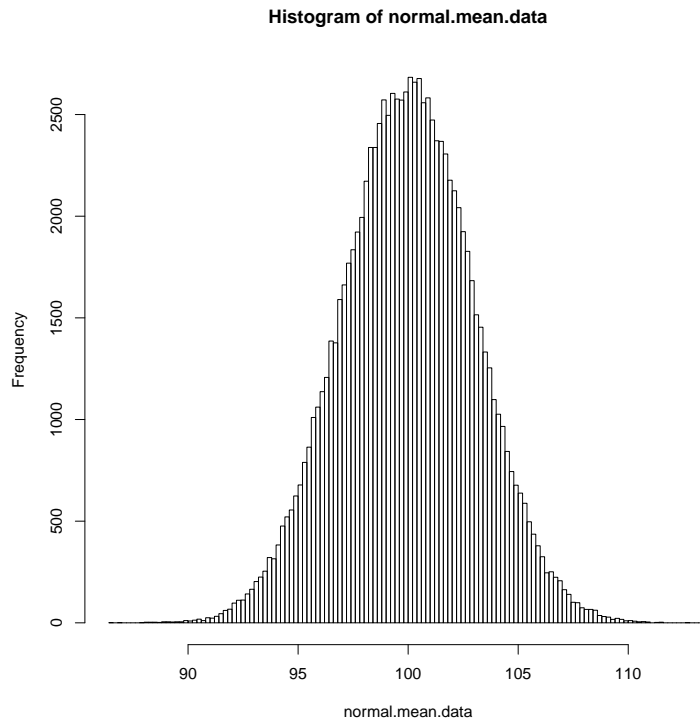
R is pretty fast. Let's do 100,000 means, compute some basic statistics, and save the data for further analysis.

```
> set.seed(12345)
> normal.mean.data <- replicate(100000,mean(rnorm(25,100,15)))
> mean(normal.mean.data)
[1] 99.9954
> sd(normal.mean.data)
[1] 2.999217
```

Those results are very close to our theoretical expectation, aren't they!

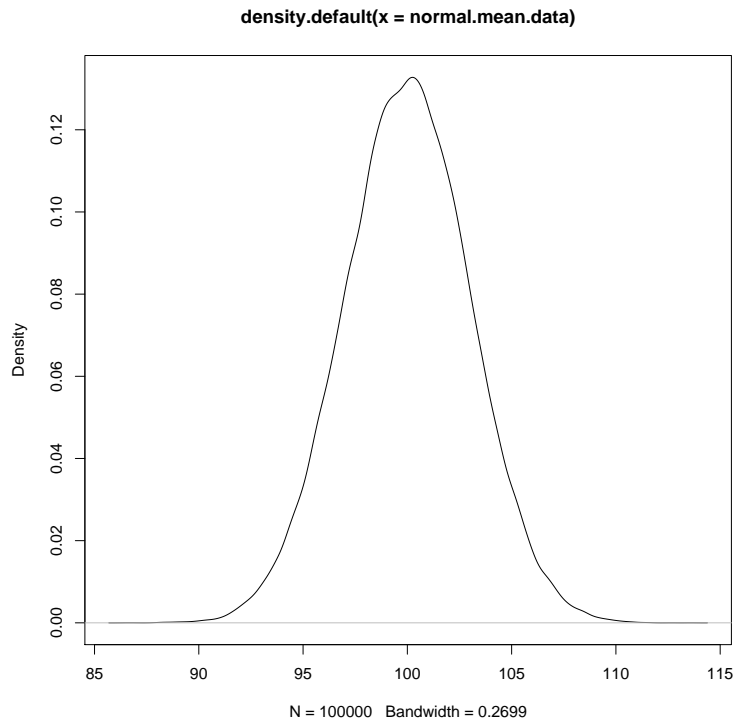
So far, we've been portraying distributions primarily with histograms. Let's look at histogram of our means.

```
> hist(normal.mean.data,breaks=100)
```



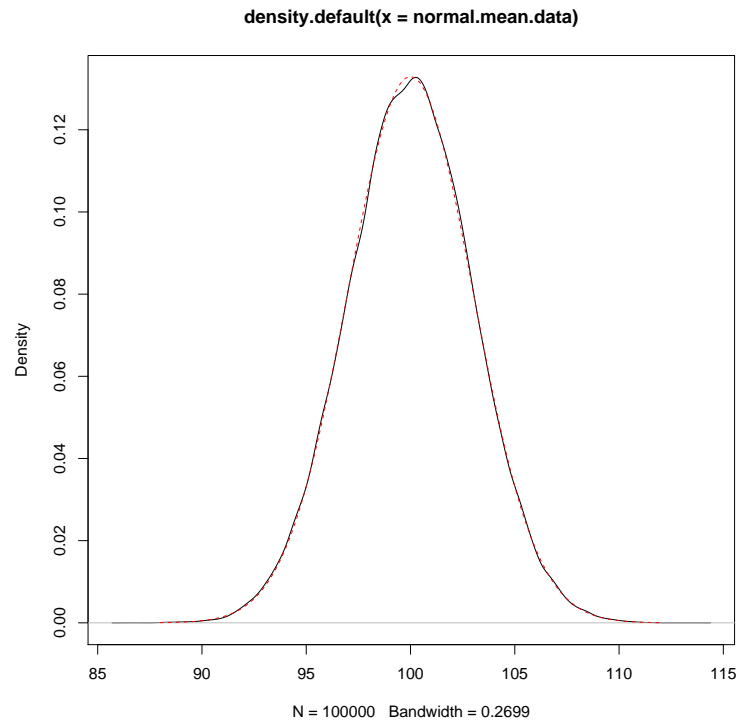
It certainly looks normal, doesn't it. Let's use a more precise and advanced technique to plot the distribution. This technique is called "kernel density estimation" (KDE). It produces a smoothed estimate of the actual probability density function. The `density()` function produces a density object, and the function `plot.density` actually plots it.

```
> plot.density(density(normal.mean.data))
```



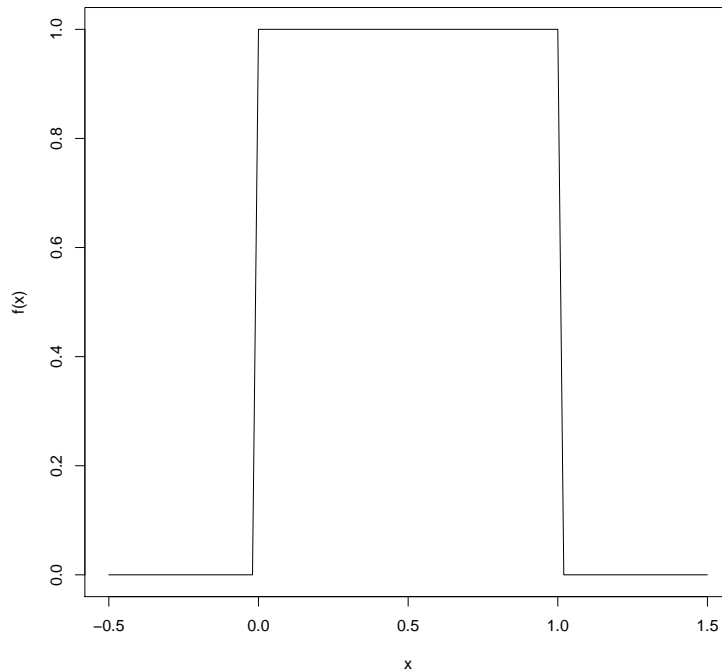
Except for some minor aberrations, that looks suspiciously like a normal distribution with mean 100 and standard deviation 3. How close is it? We can easily induce R to superimpose the precise probability density function on top of this graph. I'm making my line dotted red.

```
> curve(dnorm(x,100,3),88,112,col='red',lty=2,add=TRUE)
```



Now, let's look at the sampling distribution of \bar{X}_n when the population is not normally distributed. Let's take a classic example, the uniform (0,1) distribution. Here is a plot of the density function.

```
> curve(dunif(x,0,1),-.5,1.5,xlab="x",ylab="f(x)")
```



Clearly this population is not normally distributed. What will the distribution of the sample mean look like? As long as we know the population mean μ and the population standard deviation σ , and we can stipulate that observations are independent, we can deduce the mean and standard deviation of the sampling distribution of \bar{X} .

In the case of the uniform distribution, the general formulas for a $U(a, b)$ distribution are

$$\begin{aligned}\mu &= \frac{b+a}{2} \\ \sigma &= \frac{b-a}{\sqrt{12}}\end{aligned}$$

Often we wish to specify a uniform distribution in terms of its mean and standard deviation, rather than its limits. We can use the following result that you will prove on your homework. A continuous uniform random variable with mean μ and standard deviation σ has limits of $\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma$.

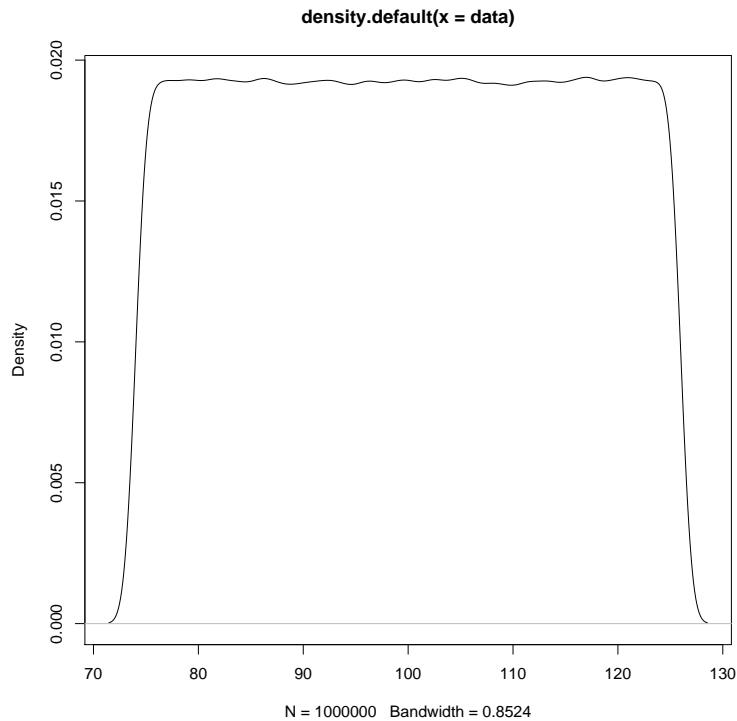
So, for example, suppose we wanted to create observations that are uniform with a mean of 100 and a standard deviation of 15. We would generate

our observations from the $U(100 - \sqrt{675}, 100 + \sqrt{675})$ distribution. It makes sense to simply create our own functions:

```
> my.runif <- function(N,mean,sd)
+ {
+ dist <- sqrt(3*sd^2)
+ runif(N, mean - dist, mean + dist)
+ }
> my.dunif <- function(x,mean,sd)
+ {
+ dist <- sqrt(3*sd^2)
+ dunif(x,mean-dist,mean+dist)
+ }
```

Let's check this out by taking a sample of 1,000,000 observations from a uniform distribution with a mean of 100 and a standard deviation of 15.

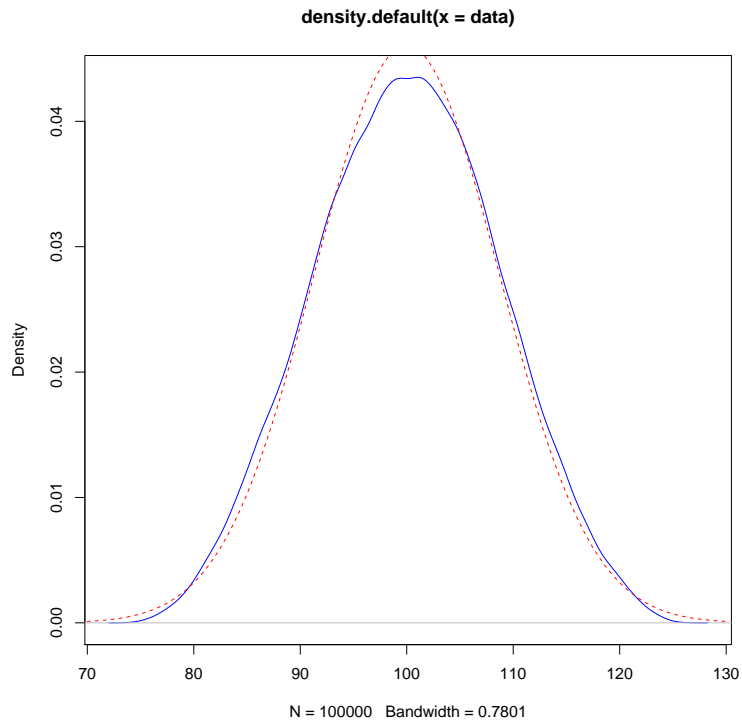
```
> data <- my.runif(1000000,100,15)
> mean(data)
[1] 100.0223
> sd(data)
[1] 14.98693
> plot.density(density(data))
```



Now let's investigate the distribution of the sample mean when we are taking our samples from a uniform distribution with a mean of 100 and a standard deviation of 15.

Let's start with a really small sample size, i.e., $N = 3$. We'll sample 100,000 means, compute the mean and standard deviation of our data, plot the KDE estimate of the data in blue, and compare it with a plot (in dotted red) of the density of the normal distribution with a mean of 100 and a standard deviation of $15/\sqrt{3}$.

```
> set.seed(12345)
> data <- replicate(100000,mean(my.runif(3,100,15)))
> mean(data)
[1] 100.0237
> sd(data)
[1] 8.667335
> plot.density(density(data),col="blue")
> curve(dnorm(x,100,15/sqrt(3)),70,130,lty=2,col="red",add=T)
```



Even with an N of only 3, the convergence to the asymptotic normal form is pretty far advanced. The actual distribution has slightly shorter tails than the normal distribution.

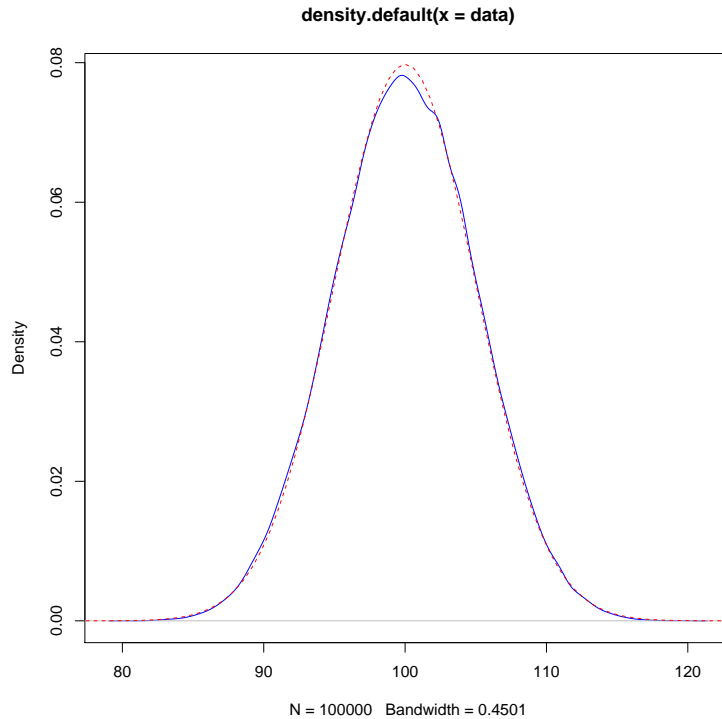
Now, let's try an N of 9.

On your own. Show your code in what you hand in

Make the appropriate changes to the code you just entered, do 100,000 sample means, and generate the plot. Reset the seed to 12345 before starting. Below are the mean, standard deviation, and the distribution plot you should get:

[1] 99.99902

[1] 5.001615



As you can see, the distribution of the sample mean is remarkably close to the normal distribution.

The uniform distribution is symmetric, and, in general, the mean from a symmetric distribution converges rather quickly to normality. Now, let's try a skewed distribution, the lognormal.

The lognormal distribution is a two parameter family. We discuss it in some detail in the next homework assignment. For now, let's examine the behavior of a lognormal variable with parameters 0.2938933 and 1.268636.

According to theory, such a variable has a mean of 3 and a standard deviation of 6. Let's check.

```
> set.seed(12345)
> data <- rlnorm(10000000, .2938933, 1.268636)
> mean(data)
[1] 3.002705
> sd(data)
[1] 6.036241
```

Let's transform this variable linearly to have a new variable Y mean of 100 and a standard deviation of 15. Since the current mean is 3 and standard deviation is 6, we simply multiply by 2.5 then add 92.5.

To hand in! Reset the seed to 12345, then take 100,000 means based on samples of size 9 from the transformed lognormal discussed in the preceding paragraph. These means will have a sampling distribution with a mean of 100 and a standard deviation of 5. Sample the means, then plot the KDE estimated density, and superimpose a normal density on top of it. Notice how I compute the mean and standard deviation of the means to make sure I computed the correct values for the sampling distribution.

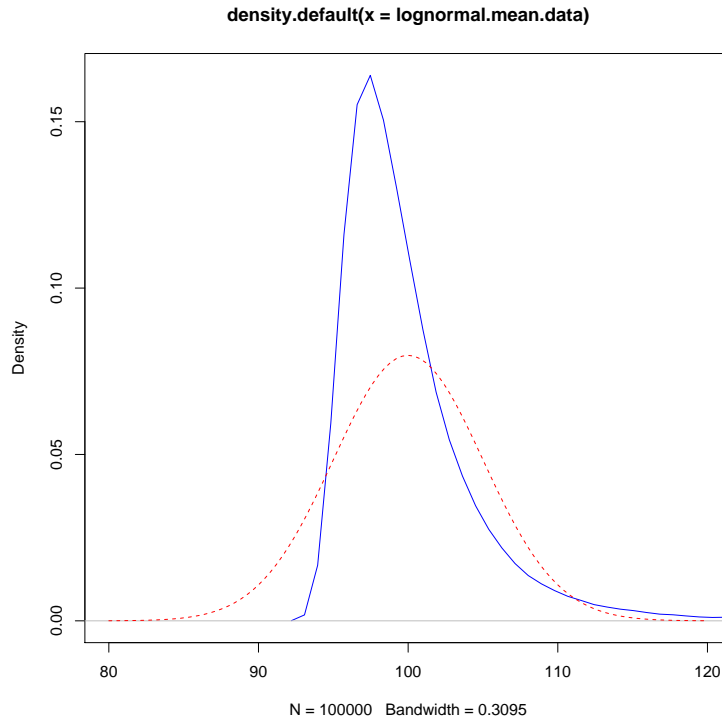
```
> set.seed(12345)
> lognormal.mean.data <- replicate(100000,
+   mean(2.5*rlnorm(9,.2938933,1.268636)+92.5))
> mean(lognormal.mean.data)

[1] 100.021

> sd(lognormal.mean.data)

[1] 5.157599

> plot.density(density(lognormal.mean.data),
+   col="blue",xlim=c(80,120))
> curve(dnorm(x,100,5),from=80,to=120,lty=2,col="red",add=T)
```



As you can see, at a sample size where the means sampled from a population that is uniform have a sampling distribution that is very close to a normal distribution, the sampling distribution for a lognormal distribution has not converged. It is still far from normal.

To hand in! Modify the above code and take a look at the sampling distribution the mean for the transformed lognormal variate when $N = 100$. Superimpose the correct normal distribution for comparison. How do things look now?

Here is a question for you to answer. In the above simulations, we used the theoretical mean and standard deviation of the sampling distribution (for example, 100 and 5). Some people might do it slightly differently, i.e., generate the data from a specific seed, compute the mean and standard deviation of the data, then plot the density but add a normal distribution *with the mean and standard deviation of the data* instead of the theoretical values. In what sense might this be better? Why doesn't it matter much in our simulations?

In our simulations so far, we've learned that convergence to a normal sampling distribution occurs at varying rates, according to the shape of the

population distribution.

2 Why Can We Substitute a Consistent Estimator for the Sampling Variance?

2.1 Comparing Z and t Numerical Outcomes

Consider the 1-sample test that $\mu = 100$. In class, we first derived a test statistic

$$Z = \frac{\bar{X}_{\bullet} - 100}{\sigma/\sqrt{N}} \quad (1)$$

This statistic assumes that σ is somehow known, and, as we mentioned several times, this is unlikely to be true in practice. I then asserted that it really doesn't matter that much, and that, as N gets reasonably large, the difference between a statistic containing the true σ and one containing a consistent estimator of σ from the sample data gets smaller and smaller.

Let's use simulation to try to understand why that's true.

To keep life simple, let's sample from a normal distribution with $\mu = 100$ and $\sigma = 15$. We'll start every simulation run by setting the seed to 12345, so our results will be reproducible.

Let's pick 3 sample sizes to represent small, medium, and large. How about 16, 100, and 225. Replicating a Z -statistic is easy. I'm going to start by creating two functions that calculate a Z statistic, and the corresponding t statistic which is just like the Z , but has the sample standard deviation in the denominator.

```
> my.z <- function(data,mu0,sigma)
+ {
+ (mean(data)-mu0)/(sigma/sqrt(length(data)))
+ }
> my.t <- function(data,mu0)
+ {
+ (mean(data)-mu0)/(sd(data)/sqrt(length(data)))
+ }
```

We already know that the Z statistic has a normal distribution. The question is, how different is the Z from the t in practice, when they are applied to the same data?

Here's 100000 replications with $N = 16$. For each data set, I compute the Z , the t , and the difference.

```

> set.seed(12345)
> z.output.16 <- replicate(100000,my.z(rnorm(16,100,15),100,15))
> set.seed(12345)
> t.output.16 <- replicate(100000,my.t(rnorm(16,100,15),100))
> diff.16 <- t.output.16 - z.output.16

```

Here are some questions for you to answer:

1. What is the correlation between the two statistics?
2. What does the scatterplot look like? Be *sure* to use `abline` plot an identity line (a line with a slope of 1 and an intercept of zero) in dotted red. This will help you see what is going on. Plot the Z statistic on the horizontal axis and the t statistic on the vertical. If the t was “working perfectly”, what would you expect to see? For a given value of Z , does t tend to be too high or too low in absolute value? (Pay special attention to performance around the typical rejection points.)
3. What does the distribution of the differences look like? (You may have to play around with the range of the density plot.)

Repeat the simulation experiment for an N of 100 and an N of 225. What do you see?

2.2 The Source of the Difference

The formulas for our two test statistics can be written in a slightly non-standard way as follows:

$$\begin{aligned}
 Z &= \frac{1}{\sigma} \times \sqrt{N}(\bar{X}_{\bullet} - \mu_0) \\
 t &= \frac{1}{s} \times \sqrt{N}(\bar{X}_{\bullet} - \mu_0)
 \end{aligned}$$

If we apply these statistics to the same data, the ratio t/Z is equal to σ/s . It depends solely on the distribution of σ , and is not affected at all by variations in the sample mean! As the ratio of σ/s moves toward 1 in expected value, and shows reduced variability, the difference between t and Z will vanish.

Let’s examine the behavior of the ratio $(Z/t)^2$. Do the following:

Set up a simulation experiment with 20,000 replications for each of the following sample sizes: 10,20,50,100,200,500. For each of these sample sizes,

- Save t and Z data.
- Then compute the ratio Z^2/t^2
- Plot its distribution.
- Compute its mean and variance.
- Across the 6 sample size, plot the mean as a function of N .
- Plot the variance as a function of N .
- What do you see in this final pair of plots?
- Does this help you understand why, as N gets increasingly large, the substitution of s for σ has less and less effect?

Clearly, it is the ability to characterize the behavior of the ratio of s to σ that leads to an understanding of how the t statistic deviates in its behavior from the Z statistic. It was a difficult problem to crack, but Student was able to do it. A key step is understanding the distribution of s^2/σ^2 . It turns out that this ratio has a distribution over repeated samples that is related to the χ^2 distribution, which we will study later in the course.