

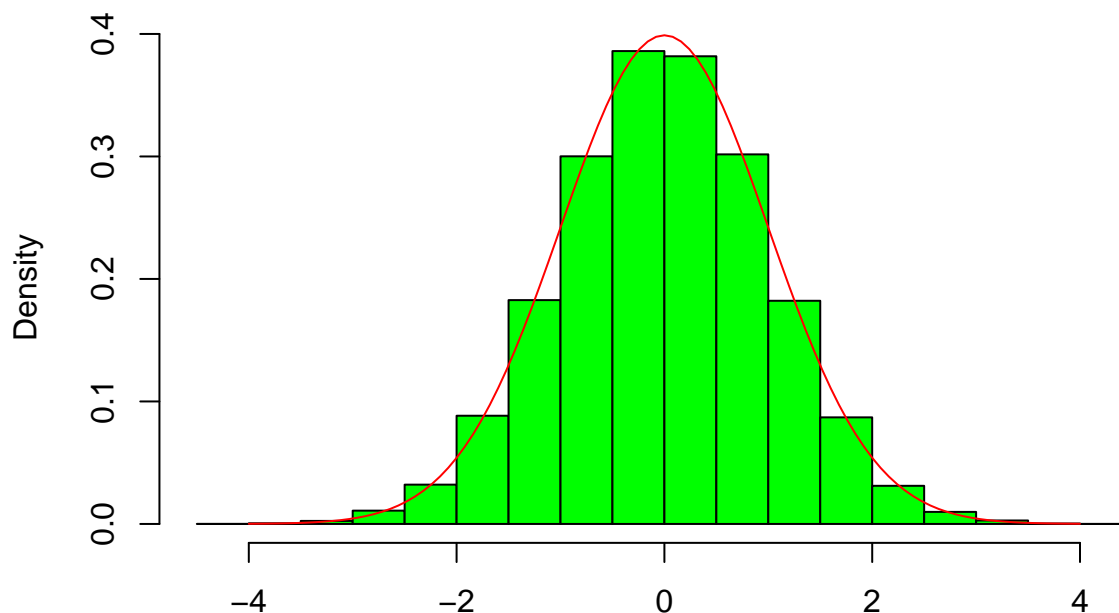
# Exam 1 STAT 4600

*Cody Frisby*

*October 7, 2016*

## Part B

First, I display a histogram of the population we will do our simulation and sampling from.



## Procedure:

We begin with the standard normal distribution,  $\mu = 0, \sigma = 1$ . I take 10 random samples  $10^4$  times. After the samples have been taken we calculate a 95% t test confidence interval for all  $10^4$  samples. Of these  $10^4$  confidence intervals, I then determine how many of them contain  $\mu$ . This is our accuracy measure.

With the re-sampling procedure, we start with only one sample of size 10 from the standard normal distribution. I then re sample from this sample  $10^5$  times with replacement producing  $10^4$  samples of size 10. Of these samples I then take a 95% quantile,  $\text{quantile}(x, c(0.025, 0.975))$ . I then determine which samples contain  $\mu$  and calculate that as a percent of the total. Here are the results:

$$t \text{ test accuracy} = 0.9485$$

$$\text{resample accuracy} = 0.9938$$

## 90% Confidence Intervals:

Comparing the accuracy of both methods using the formula for 90% confidence we get

$$t \text{ test accuracy} = 0.8988$$

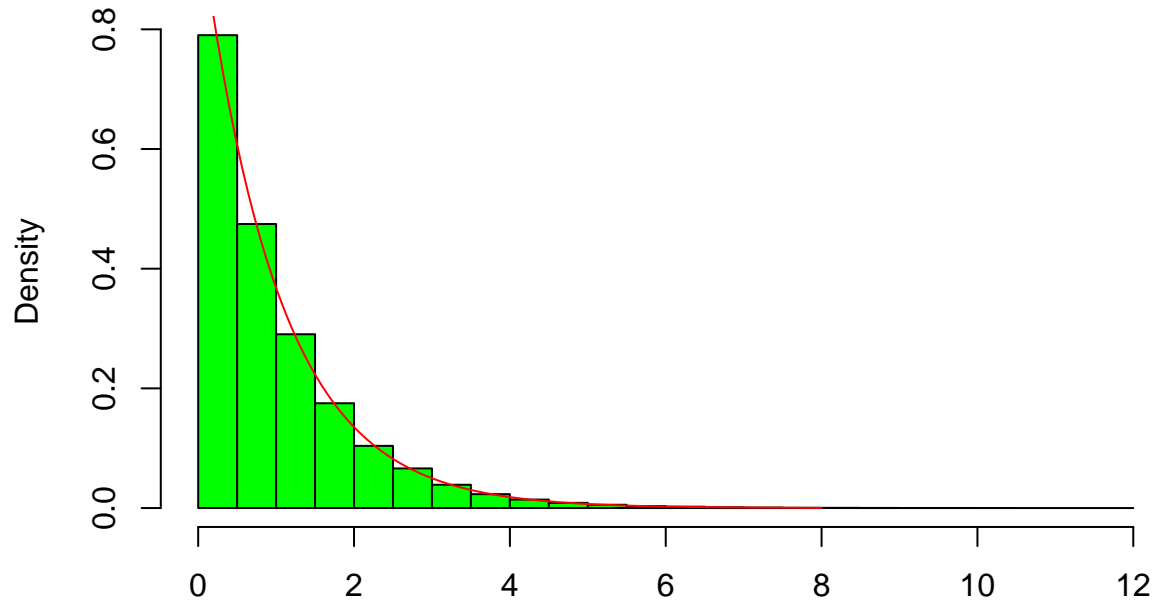
$$\text{bootstrap accuracy} = 0.9928$$

This is a very interesting result. While the  $t$  test appears to be accurate 95% and 90% of the time (which is what we would expect when sampling from a normal population) the re-sampling method appears to be correct a greater proportion of the time in both cases.

---

Next I perform the same simulation as above, but instead of sampling from the standard normal distribution I will sample from an exponential distribution where  $\beta = 1$ .

First, take a look at the population histogram.



Comparing the percent the 95%  $t$  test confidence interval gets correct with the percent the bootstrap procedure gets correct we get

$$t.test\ accuracy = 0.9004$$

$$bootstrap\ accuracy = 0.8952$$

A few thoughts about the above procedure: The resampling procedure is highly dependent on the sample that is drawn from the population. In this case the samples is of size 10. If by chance we get 10 samples that are far from the true mean of the population, then our resampling procedure will not be highly accurate. However, if we compute a  $t$ .test confidence interval from the same sample, then we would not be very accurate as well. When running one such scenario where our sample produced a  $t$  test confidence interval of  $[-1.11, -0.076]$  and we can see that  $\mu = 0$  is not contained in this interval. And when running our resampling procedure over this same sample, our accuracy measure was 0.6524. Neither of these results are very desirable, but at least with the resampling procedure we are correct 65% of the time.

---

**R Code:**

```

n <- 100000; b <- 10
x <- rnorm(n*b)
a <- matrix(sample(x, n*b, replace=TRUE), n, b)
f <- function(x){t.test(x)$conf.int} #create t conf intervals
d <- t(apply(a, 1, f)) # n t.test confidence intervals
# what if we only have one sample from our population?
B <- matrix(sample(a[1,], n*b, replace = TRUE), b, n)
h <- function(x){quantile(x, c(0.025, 0.975))}
j <- t(apply(B,2,h)) #bootstrap conf intervals
y <- sign(d[,1]) * sign(d[,2])
t.accuracy <- sum(y<0) / length(y)
z <- sign(j[,1]) * sign(j[,2])
boot.accuracy <- sum(z<0) / length(z)
# Trying some other things, bootstrapping my n "synthetic" samples, size 10
g <- function(x){sample(x, n, replace = TRUE)}
C <- matrix(apply(B, 2, g), n, n)
jj <- t(apply(C,2,h)) #bootstrap conf intervals
zz <- sign(jj[,1]) * sign(jj[,2])
boot.zz <- sum(zz<0) / length(zz)
hist(x, col="green", freq = FALSE, xlab="", main="")
curve(dnorm, col="red", add=TRUE, from = -4, to = 4)
# we could also take a look at other confidence intervals:
m <- function(x){t.test(x, conf.level = 0.9)$conf.int} # t.test interval
q <- function(x){quantile(x, c(0.05, 0.95))} #bootstrap 90% confidence interval
o <- t(apply(B,2,q)) # generate n confidence intervals from B
z <- sign(o[,1]) * sign(o[,2]) # is the true mean in our interval?
boot.accuracy <- sum(z < 0) / length(z) # this is our accuracy
p <- t(apply(a,1,m)) # generate n confidence intervals
y <- sign(p[,1]) * sign(p[,2]) # Is the mean in the interval?
t.accuracy <- sum(y<0) / length(y) # how accurate are we?
x <- rexp(n*b)
a <- matrix(sample(x, n*b, replace=TRUE), n, b)
f <- function(x){t.test(x)$conf.int} #create t conf intervals
d <- t(apply(a, 1, f)) # t confidence intervals
# what if we only have one sample from our population??
B <- matrix(sample(a[1,], n*b, replace = TRUE), b, n)
h <- function(x){quantile(x, c(0.025, 0.975))}
j <- t(apply(B,2,h)) #bootstrap conf intervals
y <- d[,1] < 1 & d[,2] > 1
t.accuracy <- sum(y) / length(y)
z <- j[,1] < 1 & j[,2] > 1
boot.accuracy <- sum(z) / length(z)
hist(x, col="green", freq = FALSE, xlab="", main="")
curve(dexp, col="red", add=TRUE, from=0, to=8)

```