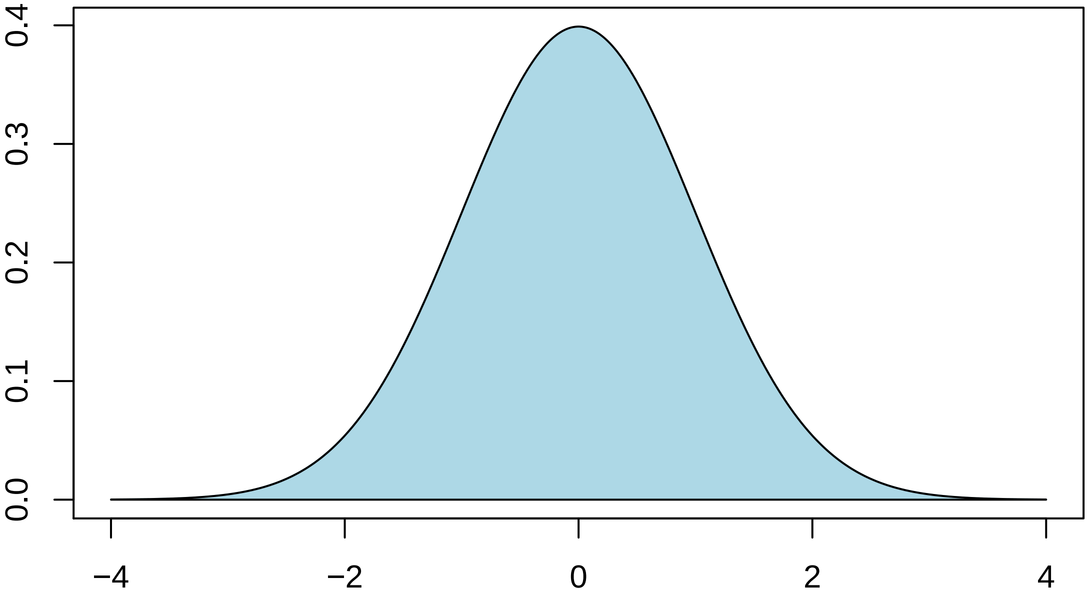
Exam 1 STAT 4600 Take 2

Cody Frisby

October 7, 2016

Part B

First, I display the population we will do our simulation and sampling from. It is the standard normal distribution, $N(0,1)$



Procedure:

I take 1000 samples of size 10. After the samples have been taken we calculate a 95% t test confidence interval for all 1000 samples. Of these 1000 confidence intervals, I then determine how many of them contain μ. This is our accuracy measure.

With the re-sampling procedure, we start with one sample of size 10 from the standard normal distribution. I then re sample from this sample 10^4 times with replacement producing 1000 samples of size 10. Of these samples I then compute 1000 means and compute a 95% quantile. We then repeat 1000 times. After this is done I determine which quantiles contain μ and calculate that as a percent of the total. Here are the results:

$$t\ test\ accuracy = 0.947$$

$$resample\ accuracy = 0.897$$

A few thoughts about the above procedure: We could just as easily do this procedure using different values for the quantiles and similarly for the t test confidence interval. We could also do this procedure with sampling from other distributions.

I now perform the same simulation as above, only changing the distribution to an exponential with rate = 1.

Comparing the accuracy of the 95% *t test* confidence interval with the percent the bootstrap procedure gets correct our results are

$$t.test\ accuracy = 0.91$$

$$bootstrap\ accuracy = 0.869$$

and surprisingly, the t test wins, even when the data is nowhere near normally distributed.

Appendix I

R Code:

```r
curve(dnorm, col="lightblue", from = -4, to = 4)
cord.x <- c(-4,seq(-4, 4, 0.01),4)
cord.y <- c(0,dnorm(seq(-4, 4, 0.01),0,1),0)
polygon(cord.x,cord.y, col="lightblue")
# simulate data from normal with mean=0 and sd=1
# then take n samples and comple a t.test confidence interval
# and then bootrap n*b times and compute 95% quantile.
# we then compare the two results' accuracy.
n <- 10; b <- 1000
x <- rnorm(n*b)
######### t test ##########
a <- matrix(sample(x, n*b), b, n)
f <- function(x){t.test(x)$conf.int} # create t conf intervals
d <- t(apply(a, 1, f)) # b t.test confidence intervals
y <- sign(d[,1]) * sign(d[,2]) # test for mu in interval
t.accuracy <- sum(y<0) / length(y) # how many contain mu?
######### end t test ###############
# what if we only have one sample from the normal?
######### bootstrap one random sample #############
w <- matrix(nrow=b, ncol=2)
for(i in 1:b){ # I can't figure this part out without a loop, so...
  B <- matrix(sample(a[i,], n*b, replace = TRUE), b, n)
  #bootstrap b samples n times, and b times in the loop
  v <- apply(B, 1, mean) # here we get 1000 means
  w[i,] <- quantile(v, c(0.025, 0.975)) # here we get the quantile...
}                                       # for the b means
z <- sign(w[,1]) * sign(w[,2]) # test to see if mu is in interval
boot.accuracy <- sum(z<0) / length(z) # how many times is mu there?
t.accuracy # print to console
boot.accuracy # print to console
```

Appendix II

Here I display a random sample of 40 of the confidence intervals produced with the bootstrap method for reference.

-0.0375064	0.9627800
-0.0617930	0.8439100
-0.5606944	0.4017437
-1.0736969	0.2189132
-0.7419435	0.1994336
-0.8774227	-0.0382338
-0.9848874	-0.1584344
-0.4625829	0.4221391
-0.5239363	0.7292809
-0.3674979	0.6234636
-0.4861170	0.8009912
-0.7239397	0.6442392
-1.3965175	-0.2293653
-0.4573730	0.8711241
-0.2873928	0.4801573
-0.5833455	0.3659781
-0.1408714	0.5011627
-0.7053115	0.1361192
-0.6611437	0.1576853
-0.9989670	0.2308147
-0.7480305	0.4622392
-0.7881986	0.3563441
0.3240158	1.2185117
-0.0927732	1.0994637
-0.4633545	0.5478588
-0.1857117	0.9325677
-0.8650031	-0.1481795
-1.0939889	0.6846951
-0.3664722	0.7065527
0.0552071	1.2323954
-0.2621860	0.5801771
-0.5727129	1.3281692
-0.1956189	0.5628363
-0.0745596	1.2039203
-0.2895863	0.7612244
-0.2478381	0.6950661
-0.4996893	0.6525592
-0.4225238	0.5906813
-0.5301806	0.3184615
-0.0489084	1.0513349