Math 3070 § 1.
Treibergs

Simulation of *t*-Confidence Intervals Example.

Name: Example
June 7, 2011

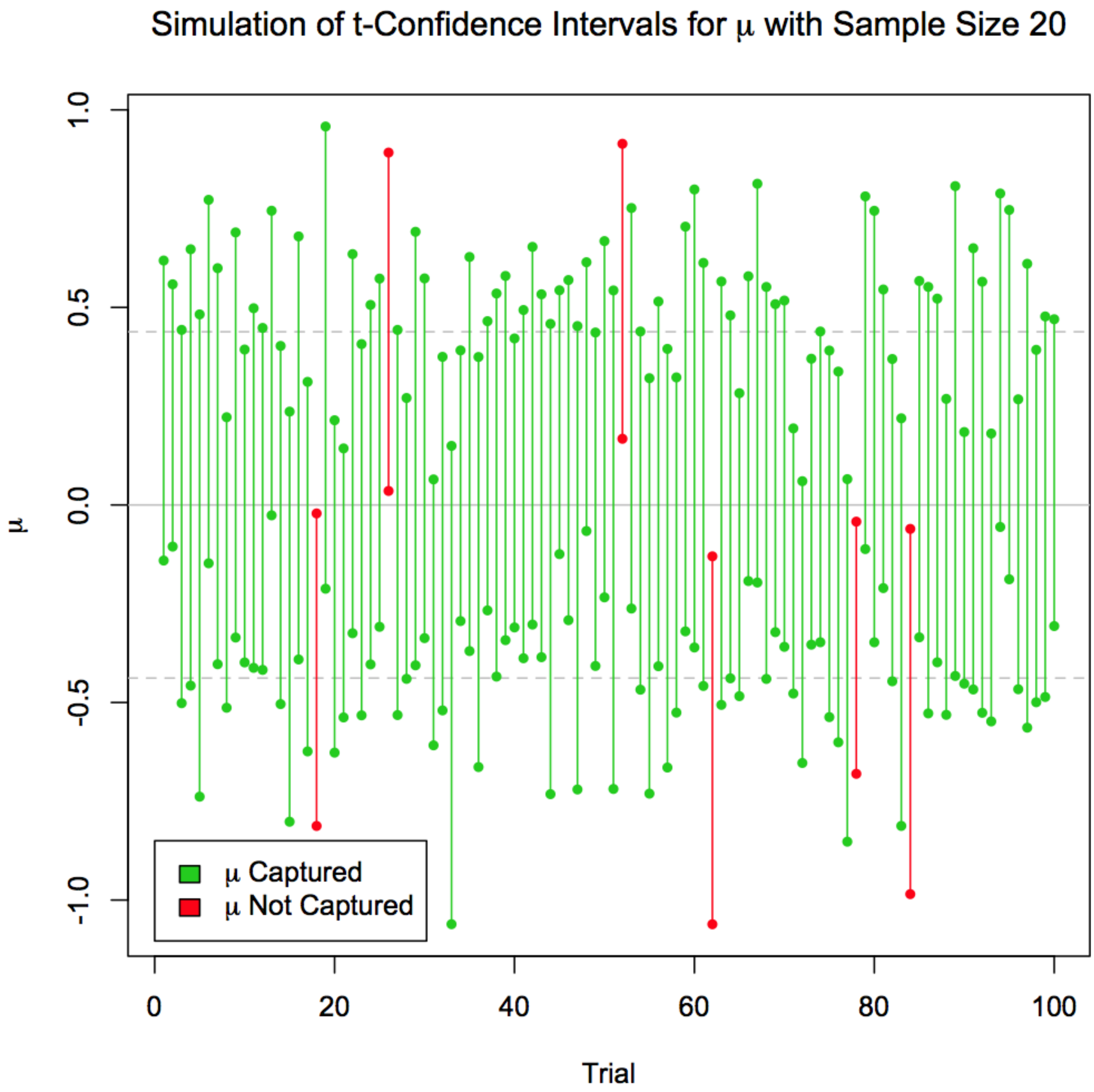
R Session:

```
R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type ’license()’ or ’licence()’ for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type ’contributors()’ for more information and
’citation()’ on how to cite R or R packages in publications.

Type ’demo()’ for some demos, ’help()’ for on-line help, or
’help.start()’ for an HTML browser interface to help.
Type ’q()’ to quit R.

[R.app GUI 1.34 (5589) i386-apple-darwin9.8.0]

[Workspace restored from /home/1004/ma/treibergs/.RData]

> ##################### SIMULATE t-CONFIDENCE INTERVALS ###############################
> # We take samples of size n=20 from a standard normal RV.
> # We repeat  m  times and count the number of times the CI captures the mean (mu=0)
> # We assume  mu is unknown for the sample so use the small sample t-distribution
> # based 2-sided CI.
>
> alpha <- .05
> n <- 20
> ta2 <- qt(alpha/2,df=n-1,lower.tail=FALSE);ta2
[1] 2.093024
>
> # The interval is   mu  in   ( xbar - ta2 * s / sqrt(n), xbar + ta2 * s / sqrt(n)
> # Equivalently, mu=0 is out of the CI if  xbar^2 = (xbar-mu)^2 >= ta2^2 * var / n
> # save the coefficient that is fixed in the loop
> c <- ta2^2/n

> # system.time({}) computes the time a procedure took to run. st[3] is elapsed time.
> m <- 100
> st <- system.time({
+                    noout <- 0
+                    for(j in 1:m){x <- rnorm(n)
+                                  if( mean(x)^2 >= c*var(x) ){ noout <- noout + 1 }
+                                 }
+                  })
```

```
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")


 Out of  100  samples of size  20 ,
 the CI failed to capture the mean  5  times.
 The proportion of intervals failing to capture is 0.05 .
 Time elapsed  0.017  seconds.

> st <- system.time({noout<- 0;
+                    for(j in 1:m){x<-rnorm(n);if(mean(x)^2 >= c*var(x)){noout<-noout+1}}})
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")


 Out of  100  samples of size  20 ,
 the CI failed to capture the mean  6  times.
 The proportion of intervals failing to capture is 0.06 .
 Time elapsed  0.018  seconds.

> st <- system.time({noout<- 0;
+                    for(j in 1:m){x<-rnorm(n);if(mean(x)^2 >= c*var(x)){noout<-noout+1}}})
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")



 Out of  100  samples of size  20 ,
 the CI failed to capture the mean  9  times.
 The proportion of intervals failing to capture is 0.09 .
 Time elapsed  0.023  seconds.

> m <- 1000
> st <- system.time({noout<- 0;
+                    for(j in 1:m){x<-rnorm(n);if(mean(x)^2 >= c*var(x)){noout<-noout+1}}})
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")



 Out of  1000  samples of size  20 ,
 the CI failed to capture the mean  45  times.
 The proportion of intervals failing to capture is 0.045 .
 Time elapsed  0.078  seconds.
```

```
> m <- 10000
> st <- system.time({noout<- 0;
+                     for(j in 1:m){x<-rnorm(n);if(mean(x)^2 >= c*var(x)){noout<-noout+1}}})
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")



 Out of  10000  samples of size  20 ,
 the CI failed to capture the mean  496  times.
 The proportion of intervals failing to capture is 0.0496 .
 Time elapsed  0.675  seconds.

> m <- 100000
> st <- system.time({noout<- 0;
+                     for(j in 1:m){x<-rnorm(n);if(mean(x)^2 >= c*var(x)){noout<-noout+1}}})
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")


 Out of  100000  samples of size  20 ,
 the CI failed to capture the mean  4915  times.
 The proportion of intervals failing to capture is 0.04915 .
 Time elapsed  6.628  seconds.

> m <- 1000000
> st <- system.time({noout<- 0;
+                     for(j in 1:m){x<-rnorm(n);if(mean(x)^2 >= c*var(x)){noout<-noout+1}}})
> cat("\n\n Out of ",format(m,scientific=FALSE)," samples of size ",n,
+ ",\n the CI failed to capture the mean ",noout,
+ " times.\n The proportion of intervals failing to capture is",noout/m,
+ ".\n Time elapsed ",st[3]," seconds. \n\n")



 Out of  1000000  samples of size  20 ,
 the CI failed to capture the mean  50003  times.
 The proportion of intervals failing to capture is 0.050003 .
 Time elapsed  66.401  seconds.
```

```
> ##################### PLOT SIMULATED CI'S ########################################
> m <- 100
> # I'll draw corresponding z-CI's  za2 / sqrt(n)  above and below  mu=zero.
> za2 <- qnorm(alpha/2,lower.tail=FALSE)
> za2
[1] 1.959964

> # Generate  m  CI's. This time store upper and lower confidence bounds, "low" and "high"
> # and flag whether the mean is captured in the logical vector "captured"
> # Since TRUE=1 and FALSE=0, sum(captured) is the number captured.
>
> low <-1:100
> high <- 1:100
> captured <- rep(TRUE,100)
> c1 <- ta2/sqrt(n)
> c12 <- c1^2
>  for(j in 1:m)
+               {
+                x <- rnorm(n)
+                xbar <- mean(x)
+                s <- sd(x)
+                captured[j] <- xbar^2 >= c12*s^2
+                low[j] <- xbar - c1*s
+                high[j] <- xbar + c1*s
+               }
> sum(captured)
[1] 6
> plot(c(low,high),type="n",xlim=c(1,100),xlab="Trial",ylab=expression(mu),pch=19)
> abline(h = c(0, za2/sqrt(n), -za2/sqrt(n)), lty = c(1,2,2), col = "gray")
> points(high, col = 3-captured, pch = 20)
> points(low, col = 3-captured, pch = 20)
> for(i in 1:100)
+                {
+                 lines(c(i,i), c(low[i],high[i]), col = 3-captured[i], pch = 19)
+                }
> title(expression(paste("Simulation of t-Confidence Intervals for ", mu,
+ " with Sample Size 20")))
> legend(0,-.85, legend = c(expression(paste(mu," Captured")),
+ expression(paste(mu," Not Captured"))), fill = c(3,2))
```


Simulation of t-Confidence Intervals for μ with Sample Size 20
μ
1.0
0.5
0.0
-0.5
-1.0
0
20
40
60
80
100
Trial
🟩 μ Captured
🟥 μ Not Captured