

# Assignment 3 - Confidence Interval and Hypothesis Testing

*Saha Debanshee Gopal*

*November 28, 2016*

## Confidence Interval and Hypothesis Testing:

1. In a filament cut test, a razor blade was tested six different times with ultimate forces corresponding to 8.5, 13.9, 7.4, 10.3, 15.7, and 4.0 g.

(a) Find a 95% confidence interval on the mean using the standard Student's t-distribution

```
set.seed(1)
x<-c(8.5, 13.9, 7.4, 10.3, 15.7, 4.0)
summary(x)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  4.000   7.675   9.400   9.967  13.000  15.700

c(var(x), sd(x))#variance and standard deviation of sample

## [1] 18.558667  4.307977

shapiro.test(x)#test for normality Shapiro - Wilk normality test data: x

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.97387, p-value = 0.9174

n <- length(x) #sample size
n

## [1] 6

mean(x) + qt(0.975, n - 1) * sd(x) * c(-1, +1)/sqrt(n) #student t based 95% CI

## [1]  5.445722 14.487611
```

(b) Find a 95% confidence interval on the mean using Efron's percentile method.

For the mean, we have a formula for the standard error, namely  $\frac{s}{\sqrt{n}}$ , where  $s$  is the population standard deviation and  $n$  is the sample size. This can be estimated for any sample by  $\hat{s} = s$ . This estimate should be acceptable for moderate to large sample sizes ( $n$  at least 30 or more):

```
set.seed(1) #for reproducibility
u0 <- mean(x) #mean of original sample
Sh <- sd(x)/sqrt(n)
```

```

thetas <- NULL
tstar <- NULL
for (i in 1:1000) { #bootstrap resampling
  xx <- sample(x, n, replace = TRUE) #new resample
  u <- mean(xx) #estimate
  thetas[i] <- u #save
  tstar[i] <- ((u-u0)/sd(xx)/sqrt(n)) #pivotal quantity
}
c(u0, mean(thetas)) #compare sample mean to mean of bootstrap sampling distribution

## [1] 9.966667 9.848217

c(Sh, sd(thetas)) #compare standard error from sample to standard error estimate from bootstrap distri

## [1] 1.758724 1.561886

summary(tstar)

##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -2.16700 -0.12880 -0.01265 -0.02317  0.10170  0.83810

quantile(tstar, probs = c(0.025,0.975)) #quantiles from bootstrap percentile t

##      2.5%      97.5%
## -0.4948938  0.4153047

qt(c(0.025,0.975), n - 1) #quantiles from student t distribution

## [1] -2.570582  2.570582

u0 + quantile(tstar, probs = c(0.025,0.975)) * Sh #bootstrap percentile t confidence interval

##      2.5%      97.5%
##  9.096285 10.697073

u0 + qt(c(0.025,0.975), n - 1) * Sh #student t confidence interval for comparison

## [1]  5.445722 14.487611

```

The mean of the sample (9.9666) is very close to the mean of the bootstrap sampling distribution (9.9263), because the mean is an unbiased estimator. The standard errors are not quite as close (1.7587 vs. 1.5955). The  $t^*$  quantiles (??? 0.4938 and +0.4117) are somewhat different from Student's  $t$  - quantiles (??? 2.57 and + 2.57), reflecting the asymmetry of the underlying distribution. This results in a very slightly different confidence interval estimate: (9.098, 10.690) for the bootstrap and (5.445, 14.48) for Student's  $t$  -based method assuming normality. The closeness of the two intervals suggests the accuracy of both.

The function " boott " from the package" bootstrap " is used for comparison:

```

set.seed(1) #reproducibility
require('bootstrap')

## Loading required package: bootstrap

boott(x, theta = mean, sdfun = function(x,nbootsd,theta)
{sqrt(var(x)/length(x))},
nboott = 1000, perc = c(0.025,0.975)) #bootstrap percentile t 95% C.I.

```

```
## $confpoints
##      0.025  0.975
## [1,] 5.54853 16.352
##
## $theta
## NULL
##
## $g
## NULL
##
## $call
## boott(x = x, theta = mean, sdfun = function(x, nbootsd, theta) {
##      sqrt(var(x)/length(x))
## }, nboott = 1000, perc = c(0.025, 0.975))
```

(c) Find a 95% confidence interval on the mean using the BCa method and the ABC method.

```
#BCa and ABC
set.seed(1) #for reproducibility
require('boot')

## Loading required package: boot

fboot<- function(x, i) mean(x[i]) #compute estimate given data x and index set i
bs<- boot(x, fboot, R = 1000) #generate bootstrap estimates
boot.ci(bs, type='bca' , conf = 0.95) #BCa method 95% C.I.

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bs, conf = 0.95, type = "bca")
##
## Intervals :
## Level      BCa
## 95%      ( 6.933, 13.417 )
## Calculations and Intervals on Original Scale

fabc <- function(x, w) w%*%x #ABC uses weighted average.
abc.ci(x, fabc, conf=0.95) #ABC method C.I.

## [1] 0.950000 6.863332 13.157697
```

The 95% confidence interval estimates for the BCa and ABC methods are (6.75, 13.30) and (6.863, 13.157), comparable to those found before. It should be noted that the ABC method is both very fast computationally and very accurate, typically giving intervals very close to those obtained from the BCa method.

(d) Find a 95% confidence interval on the mean using the percentile-t method

```
set.seed(1) #reproducibility
boott(x, theta = mean, nbootsd = 100, nboott = 1000,
perc = c(0.025,0.975)) #bootstrap percentile t 95% C.I. with nested bootstrap
```

```
## $confpoints
##      0.025      0.975
## [1,] 5.800209 16.10702
##
## $theta
## NULL
##
## $g
## NULL
##
## $call
## boott(x = x, theta = mean, nbootsd = 100, nboott = 1000, perc = c(0.025,
##      0.975))
```

(e) How do the intervals compare? Which intervals do you trust? What does this tell you about the benefits of parametric methods on small ( $n < 30$ ) samples and the problems of using bootstrap on such samples? What does it tell you about the percentile-t method compared with the other bootstrap methods, at least when a formula for the standard error is known?

Ans: As  $n_{\text{Size}}$  becomes large, the differences among methods shrink in size. For  $n_{\text{Size}}$  ( $n_{\text{Size}}$  = The sample size of the originating data that is to be bootstrapped) small (10 or less), the normal - t method performs best, probably because  $n_{\text{Size}}$  is too small for the generation of reasonable resampling distributions. Regardless of this, it should be noted that all methods have large coverage errors for  $n_{\text{Size}} = 100$  or less, and this does not improve much even for  $n_{\text{Size}}$  as large as 2000.

2. United States Environmental Protection Agency (USEPA) deems peanut butter adulterated if the mean aflatoxin residue is 20 ppb or more. The industry average for peanut was found to be 5.7 ppb in 1990 by consumer reports. In actual testing, 12 lots of peanut butter had aflatoxin residues in parts per billion of 4.94, 5.06, 4.53, 5.07, 4.99, 5.16, 4.38, 4.43, 4.93, 4.72, 4.92, and 4.96.

(a) Estimate an Efron percentile bootstrap 90% confidence interval on the mean aflatoxin residue. Use  $B = 1000$  resamples.

```
set.seed(1)
x <- c(4.94, 5.06, 4.53, 5.07, 4.99, 5.16, 4.38, 4.43, 4.93, 4.72, 4.92, 4.96)
summary(x)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      4.380   4.672   4.935   4.841   5.008   5.160
```

```
c(var(x), sd(x)) #variance and standard deviation of sample
```

```
## [1] 0.06866288 0.26203603
```

```
shapiro.test(x) #test for normality Shapiro - Wilk normality test data: x
```

```
##
## Shapiro-Wilk normality test
##
## data:  x
## W = 0.87092, p-value = 0.06713
```

```

n <- length(x) #sample size
n

## [1] 12

mean(x) + qt(0.950, n - 1) * sd(x) * c(- 1, + 1)/sqrt(n) #student t based 90% CI

## [1] 4.704987 4.976680

u0 <- mean(x) #mean of original sample
Sh <- sd(x)/sqrt(n)
thetas <- NULL
tstar <- NULL
for (i in 1:1000) { #bootstrap resampling
  xx <- sample(x, n, replace = TRUE) #new resample
  u <- mean(xx) #estimate
  thetas[i] <-u #save
  tstar[i] <- ((u-u0)/sd(xx)/sqrt(n)) #pivotal quantity
}
c(u0, mean(thetas)) #compare sample mean to mean of bootstrap sampling distribution

## [1] 4.840833 4.846526

c(Sh, sd(thetas)) #compare standard error from sample to standard error estimate from bootstrap distribution

## [1] 0.07564329 0.07083942

summary(tstar)

##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -0.33440 -0.03974  0.00745  0.02389  0.06949  0.83080

quantile(tstar, probs = c(0.05,0.95)) #quantiles from bootstrap percentile t

##           5%           95%
## -0.1185887  0.2045921

qt(c(0.050,0.950), n - 1) #quantiles from student t distribution

## [1] -1.795885  1.795885

u0 + quantile(tstar, probs = c(0.05,0.95)) * Sh #bootstrap percentile t confidence interval

##           5%           95%
## 4.831863 4.856309

u0 + qt(c(0.05,0.95), n - 1) * Sh #student t confidence interval for comparison

## [1] 4.704987 4.976680

#using nested bootstrap
set.seed(1) #reproducibility
require('bootstrap')
boott(x, theta = mean, sdfun = function(x,nbootsd,theta)
{sqrt(var(x)/length(x))},
nboott = 1000, perc = c(0.05,0.95)) #bootstrap percentile t 90% C.I.

## $confpoints
##           0.05           0.95
## [1,] 4.666061 4.950734
##

```

```
## $theta
## NULL
##
## $g
## NULL
##
## $call
## boott(x = x, theta = mean, sdfun = function(x, nbootsd, theta) {
##     sqrt(var(x)/length(x))
## }, nboott = 1000, perc = c(0.05, 0.95))
```

(b) Compare the alfatoxin level found with the industry average value of 5.7 ppm: Is the upper confidence limit less than 5.7 ppb, or is it equal or above? What does this imply about a hypothesis test of  $H_0 : \mu \geq 5.7$  ppb versus  $H_1 : \mu < 5.7$  ppb at the  $\alpha = 0.05$  significance level?

```
set.seed(1)
x <- c(4.94, 5.06, 4.53, 5.07, 4.99, 5.16, 4.38, 4.43, 4.93, 4.72, 4.92, 4.96)
summary(x)

##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##    4.380   4.672   4.935   4.841   5.008   5.160

c(var(x), sd(x))#variance and standard deviation of sample

## [1] 0.06866288 0.26203603

shapiro.test(x)#test for normality Shapiro - Wilk normality test data: x

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.87092, p-value = 0.06713

n <- length(x) #sample size
n

## [1] 12

mean(x) + qt(0.975, n - 1) * sd(x) * c(-1, +1)/sqrt(n) #student t based 90% CI

## [1] 4.674344 5.007323

set.seed(1) #for reproducibility
u0 <- mean(x) #mean of original sample
Sh <- sd(x)/sqrt(n)
thetas <- NULL
tstar <- NULL
for (i in 1:1000) { #bootstrap resampling
  xx <- sample(x, n, replace = TRUE) #new resample
  u <- mean(xx) #estimate
  thetas[i] <- u #save
  tstar[i] <- ((u-u0)/sd(xx)/sqrt(n)) #pivotal quantity
}
c(u0, mean(thetas)) #compare sample mean to mean of bootstrap sampling distribution
```

```
## [1] 4.840833 4.846526
c(Sh, sd(thetas)) #compare standard error from sample to standard error estimate from bootstrap distri

## [1] 0.07564329 0.07083942
summary(tstar)

##      Min.   1st Qu.   Median     Mean  3rd Qu.    Max.
## -0.33440 -0.03974  0.00745  0.02389  0.06949  0.83080
quantile(tstar, probs = c(0.025,0.975)) #quantiles from bootstrap percentile t

##      2.5%      97.5%
## -0.1466404  0.3268811
qt(c(0.025,0.975), n - 1) #quantiles from student t distribution

## [1] -2.200985  2.200985
u0 + quantile(tstar, probs = c(0.025,0.975)) * Sh #bootstrap percentile t confi dence interval

##      2.5%      97.5%
## 4.829741 4.865560
u0 + qt(c(0.025,0.975), n - 1) * Sh #student t confidence interval for comparison

## [1] 4.674344 5.007323
#using nested bootstrap
set.seed(1) #reproducibility
require('bootstrap')
boott(x, theta = mean, sdfun = function(x,nbootsd,theta)
{sqrt(var(x)/length(x))},
nboott = 1000, perc = c(0.025,0.975)) #bootstrap percentile t 95% C.I.

## $confpoints
##      0.025      0.975
## [1,] 4.584036 4.978569
##
## $theta
## NULL
##
## $g
## NULL
##
## $call
## boott(x = x, theta = mean, sdfun = function(x, nbootsd, theta) {
##      sqrt(var(x)/length(x))
## }, nboott = 1000, perc = c(0.025, 0.975))
set.seed(1) #reproducibility
boott(x, theta = mean, nbootsd = 100, nboott = 1000,
      perc = c(0.025,0.975)) #bootstrap percentile t 95% C.I. with nested bootstrap

## $confpoints
##      0.025      0.975
## [1,] 4.579057 4.972081
##
## $theta
```

```
## NULL
##
## $g
## NULL
##
## $call
## boott(x = x, theta = mean, nbootsd = 100, nboott = 1000, perc = c(0.025,
##      0.975))
```

The confidence points are 4.55 and 4.99. The upper confidence limit is less than 5.7 ppb. Thus, we reject the null hypothesis (H0).

(c) Find the P-value for the test in (b)

The p value can be found using the t-test since the population standard deviation is unknown.

Population mean (pm)= 5.7

Sample mean (sm)= 4.84

samples size (n)= 12

degree of freedom (df)= 11

Sample standard deviation (ssd)= 0.262

Formula :  $t: (sm - pm) / (ssd / \sqrt{n})$

```
x <-c(4.94, 5.06, 4.53, 5.07, 4.99, 5.16, 4.38, 4.43, 4.93, 4.72, 4.92, 4.96)
sm<-mean(x)
ssd<-sd(x)
pm <-5.7
n<-12
t<-(sm-pm)/(ssd/sqrt(12))
t
```

```
## [1] -11.35814
```

```
p<- 1*pt(-abs(t),df=n-1)
```

```
p
```

```
## [1] 1.021285e-07
```

**3.**

Find the observed Recall R, Precision P, and the figure of merit F2.

```
#Observed Recall
(r <- 123/(123+625))
```

```
## [1] 0.1644385
```



```

#Precision
(p <- 123/(123+27))

## [1] 0.82
#Figure of merit F2
(f2 <- p * r / (0.8 * p + 0.2 * r))

## [1] 0.1957352

```

Resample the  $2 \times 2$  contingency table  $B = 1000$  times.

```

#for reproducibility
set.seed(1)

#nt is total relevant
#nc is total irrelevant
#xt is relevant retrived
#xc is irrelevant retrived
#numresamp is number of resamples
nt <- 150
nc <- 7328
xt <- 123
xc <- 625
numresamp <- 1000
n <- nt + nc
number1s <- xt+xc
number0s <- n - number1s
pop <- c(rep(1,number1s), rep(0,number0s))
obspschat <- xt/nt
obspschat <- xc/nc
obsdiff <- obspschat - obspschat
simdiffs <- rep(0, numresamp)
for(i in 1:numresamp)
{
  resampt <- sample(pop, nt, replace = T)
  resampc <- sample(pop, nc, replace = T)
  simdiffs[i] <- sum(resampt)/nt - sum(resampc)/nc
}

#Resampled Data
head(simdiffs)

## [1] -0.037045124 -0.009286754 0.005138282 0.011552038 0.057127001
## [6] -0.049696143

```

Find 90% and 95% confidence intervals for the true F2 for the complete database using Efron's percentile method.

```

#Efron percentile based on 90% CI
quantile(simdiffs, probs = c(0.05,0.95))

##          5%          95%

```

```
## -0.03928894 0.04528157
```

```
#Efron percentile based on 95% CI
```

```
quantile(simdiffs, probs = c(0.025,0.975))
```

```
##          2.5%          97.5%
```

```
## -0.04589224 0.05249750
```

Conduct a test at the 0.05 significance level of  $H_0 : F_2 \leq 0.4$  versus  $H_1 : F_2 > 0.4$ . Should the search service be engaged for this lawsuit?

```
t.test(simdiffs, mu = 0.4, alternative = "greater")
```

```
##
```

```
## One Sample t-test
```

```
##
```

```
## data: simdiffs
```

```
## t = -498.44, df = 999, p-value = 1
```

```
## alternative hypothesis: true mean is greater than 0.4
```

```
## 95 percent confidence interval:
```

```
## -0.0004730462          Inf
```

```
## sample estimates:
```

```
## mean of x
```

```
## 0.0008453912
```