Confidence Intervals

Oliver d’Pug

## Bootstrap Confidence Intervals

The bootstrap can be used to find an approximate sampling distribution for a given estimator/statistic — the bootstrap distribution. Quantiles of this distribution provide an easy way to generate a confidence interval for the statistic.

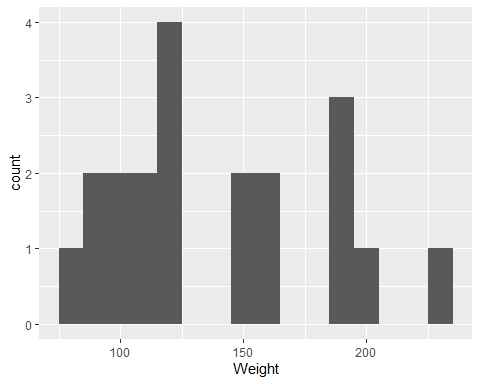
### The Mean

Consider the sample mean, . We can use the **boot** package to find the bootstrap distribution. We do so for the **Weight** data.

htwt <- read.csv("http://facweb1.redlands.edu/fac/jim\_bentley/data/Math%20312/regression/htwt.csv")  
 head(htwt)

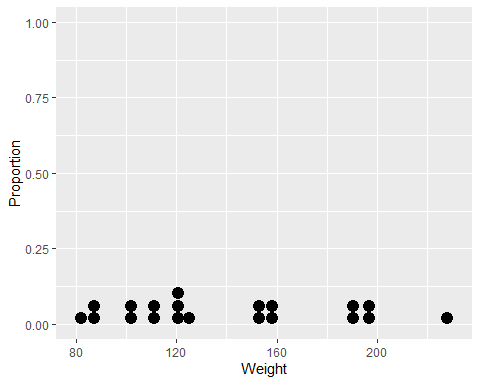
Height Weight Group  
1 64 159 1  
2 63 155 2  
3 67 157 2  
4 60 125 1  
5 52 103 2  
6 58 122 2

ggplot(htwt, aes(x=Weight)) + geom\_histogram(binwidth = 10)



ggplot(htwt, aes(x=Weight)) + geom\_dotplot() + ylab("Proportion")

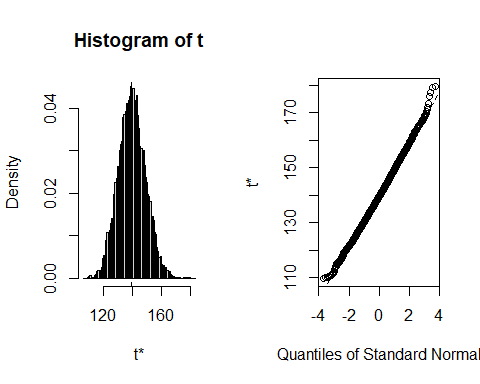
Bin width defaults to 1/30 of the range of the data. Pick better value with  
`binwidth`.



b.mean <- function(d, i){mean(d[i])}  
 wt.mean.boot <- boot(htwt$Weight, b.mean, R=9999)  
 wt.mean.boot

ORDINARY NONPARAMETRIC BOOTSTRAP  
  
  
Call:  
boot(data = htwt$Weight, statistic = b.mean, R = 9999)  
  
  
Bootstrap Statistics :  
 original bias std. error  
t1\* 139.6 0.1483898 9.400818

plot(wt.mean.boot)



While the raw data appear to be non-normally distributed — abnormal? — the distribution of *means* seems to be normal. Apparently the Central Limit Theorem has kicked in. Note that the estimator appears to be unbiased. The standard error is estimated to be 9.4008176. This is close to 9.6423954. All of this is interesting and useful — later.

To get the bootstrap confidence interval, we use a quantile approach. We can trap a proportion of ``plausible’’ values between two values found by trimming the required percentages off of each end.

quantile(wt.mean.boot$t, c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995))

0.5% 2.5% 5% 95% 97.5% 99.5%   
116.4990 121.9500 124.4000 155.4500 158.4000 164.6515

We see that a 95% CI for the mean is 121.95 to 158.4 and a 99% CI for the mean is 116.499 to 164.6515.

The confidence intervals from above, and in particular the Percentile method, are similar to those given by the **boot.ci** function below.

boot.ci(wt.mean.boot)

Warning in boot.ci(wt.mean.boot): bootstrap variances needed for studentized  
intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
Based on 9999 bootstrap replicates  
  
CALL :   
boot.ci(boot.out = wt.mean.boot)  
  
Intervals :   
Level Normal Basic   
95% (121.0, 157.9 ) (120.8, 157.2 )   
  
Level Percentile BCa   
95% (122.0, 158.4 ) (122.4, 159.3 )   
Calculations and Intervals on Original Scale

The Normal CI given above is calculated as or where , is the confidence level, and is the bootstrap standard error.

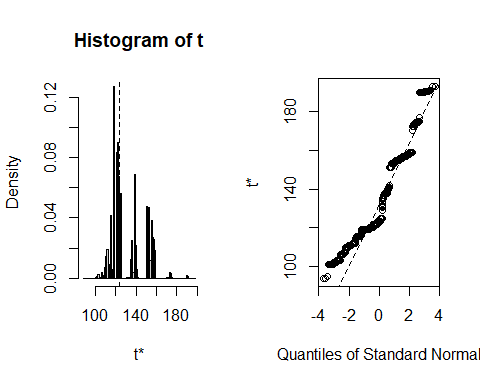
### The Median

Consider the sample median, , chosen such that . We can use the **boot** package to find the bootstrap distribution of the sample median. We do so using the **Weight** data from above.

b.median <- function(d, i){median(d[i])}  
 wt.median.boot <- boot(htwt$Weight, b.median, R=9999)  
 wt.median.boot

ORDINARY NONPARAMETRIC BOOTSTRAP  
  
  
Call:  
boot(data = htwt$Weight, statistic = b.median, R = 9999)  
  
  
Bootstrap Statistics :  
 original bias std. error  
t1\* 123.5 8.084658 15.97414

plot(wt.median.boot)



While the distribution of the bootstrap *means* seems to be normal, the distribution of the bootstrap *medians* does not. Since the Central Limit Theorem is a statement about the asymptotic distribution of *sums* of random variables (recalling the proof and the use of ), this should not be surprising. Note that the estimator appears to be biased. The standard error is estimated to be 15.974143.

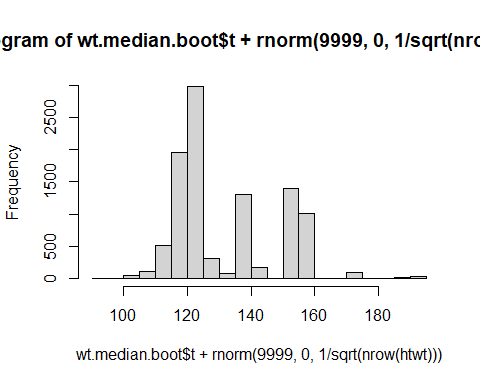
Because of the granularity of the bootstrap distribution we might be uncomfortable computing the bootstrap confidence interval for the median.

quantile(wt.median.boot$t, c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995))

0.5% 2.5% 5% 95% 97.5% 99.5%   
105.5 111.0 112.0 157.0 158.0 174.5

One suggested method for dealing with the granularity is to add a little random noise to smooth things out. Some authors suggest using

hist(wt.median.boot$t+rnorm(9999, 0, 1/sqrt(nrow(htwt))))



qqnorm(wt.median.boot$t+rnorm(9999, 0, 1/sqrt(nrow(htwt))))



quantile(wt.median.boot$t+rnorm(9999, 0, 1/sqrt(nrow(htwt))), c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995))

0.5% 2.5% 5% 95% 97.5% 99.5%   
105.5154 110.9115 112.1150 156.9491 158.1549 174.6958

The confidence intervals computed above can be compared to those generated by **boot.ci**.

boot.ci(wt.median.boot)

Warning in boot.ci(wt.median.boot): bootstrap variances needed for studentized  
intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
Based on 9999 bootstrap replicates  
  
CALL :   
boot.ci(boot.out = wt.median.boot)  
  
Intervals :   
Level Normal Basic   
95% ( 84.1, 146.7 ) ( 89.0, 136.0 )   
  
Level Percentile BCa   
95% (111, 158 ) (110, 157 )   
Calculations and Intervals on Original Scale

With an apparent lack of normality, it would be a waste of time to use a normal approach to computing a confidence interval.

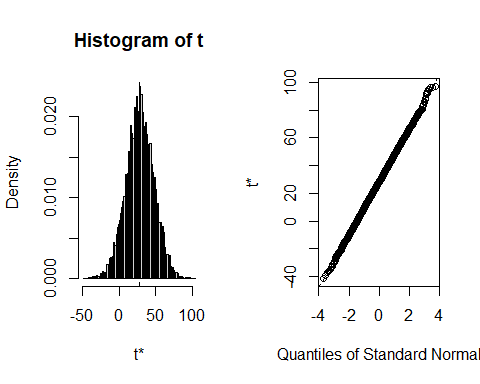
### Difference of Means

We can use the **boot** package to look at more interesting statistics. Consider creating a confidence interval for the difference of the means of two samples. As an example, we can look at the difference of **Group** **Weight**s using the data in the **htwt** data frame.

wtgrp <- htwt[,c("Weight", "Group")]  
 meanDiff <- function(x, i){  
 ### Compute group means  
 y <- tapply(x[i,1], x[i,2], mean)  
 ### Return the difference  
 y[1]-y[2]  
 }  
 wtgrp.meanDiff.boot <- boot(wtgrp, meanDiff, R=9999)  
 wtgrp.meanDiff.boot

ORDINARY NONPARAMETRIC BOOTSTRAP  
  
  
Call:  
boot(data = wtgrp, statistic = meanDiff, R = 9999)  
  
  
Bootstrap Statistics :  
 original bias std. error  
t1\* 28 0.3298177 19.04907

plot(wtgrp.meanDiff.boot)



boot.ci(wtgrp.meanDiff.boot)

Warning in boot.ci(wtgrp.meanDiff.boot): bootstrap variances needed for  
studentized intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
Based on 9999 bootstrap replicates  
  
CALL :   
boot.ci(boot.out = wtgrp.meanDiff.boot)  
  
Intervals :   
Level Normal Basic   
95% (-9.67, 65.01 ) (-9.72, 65.38 )   
  
Level Percentile BCa   
95% ( -9.38, 65.72 ) (-10.93, 64.14 )   
Calculations and Intervals on Original Scale

## Normal and t Confidence Intervals

The bootstrap distribution of a statistic/estimator is supposed to resemble the sampling distribution of the same statistic/estimator. So, the distribution of the mean found above should be representative of what we would see if we sampled all possible samples of size from the population.

We note that 95% of the sample means are within

quantile(wt.mean.boot$t, c(0.025, 0.975))

2.5% 97.5%   
121.95 158.40

Assuming normality (which is supported by the plots generated above), the CLT suggests that the sample mean is distributed normally with mean, and standard deviation 9.6423947. A plot of the normal (red/dot) and t (blue/dash) distributions for the weight data shows their differences.

(n <- length(htwt$Weight))

[1] 20

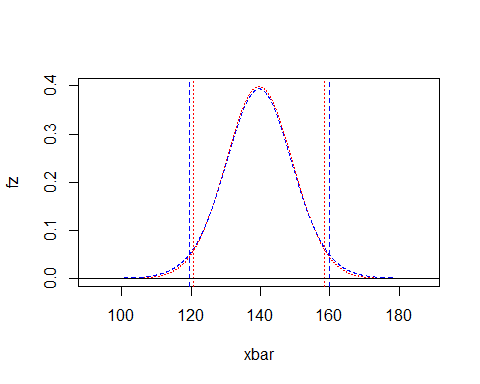
(mu <- mean(htwt$Weight))

[1] 139.6

(se <- sqrt(var(htwt$Weight)/n))

[1] 9.642395

xbar <- mu + seq(-5, 5, by=0.01)\*se  
 fz <- dnorm((xbar-mu)/se)  
 ft <- dt((xbar-mu)/se, n-1)  
 plot(xbar, fz, type="l", lty=3, col="red", ylim=range(c(fz, ft)))  
 lines(xbar, ft, lty=2, col="blue")  
 abline(v=qnorm(c(0.025,0.975), mu, se), lty=3, col="red")  
 abline(v=mu + qt(c(0.025,0.975), n-1)\*se, lty=2, col="blue")  
 abline(h=0)



Confidence intervals based upon the normal and t distributions also demonstrates the differences.

qnorm(c(0.025, 0.975), mu, se)

[1] 120.7013 158.4987

mu+qt(c(0.025, 0.975), n-1)\*se

[1] 119.4182 159.7818

If we look at the bootstrap means, we see that

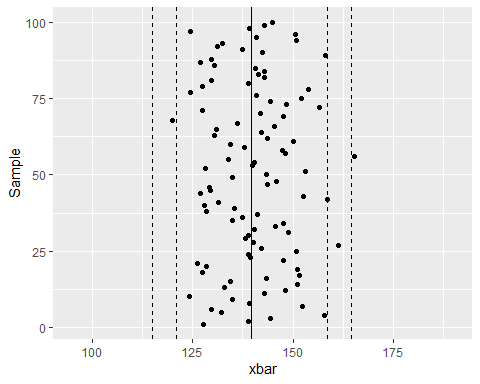
(within196se <- table(wt.mean.boot$t >= qnorm(0.025, mu, se) & wt.mean.boot$t <= qnorm(0.975, mu, se))/length(wt.mean.boot$t)\*100)

FALSE TRUE   
 4.250425 95.749575

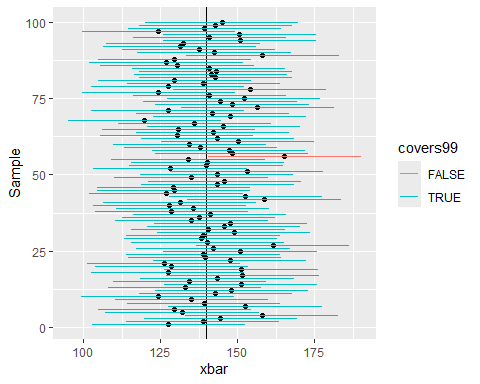
So, 95.75% of the means are within 1.96 standard errors of . We can turn this inside out and conclude that is within 1.96 standard errors of 95.75% of the sample means. Hence, if we use to create our intervals, 95% of the intervals will contain .

The plots below shows the proportion of means that fall within and outside of 95% and 99% CIs.

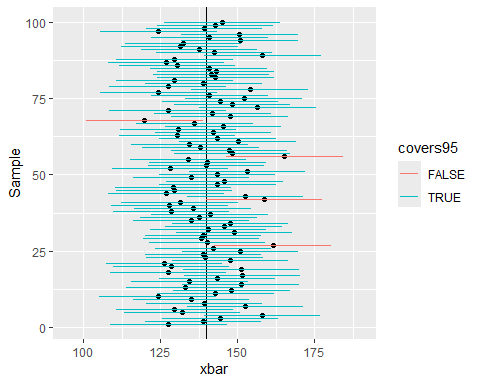
seed <- 47  
 nreps <- 100  
 Sample <- 1:nreps  
 xbar <- sample(wt.mean.boot$t, nreps)  
 mu <- mean(wt.mean.boot$t)  
 l95 <- xbar - 1.96 \* se  
 u95 <- xbar + 1.96 \* se  
 l99 <- xbar - 2.576 \* se  
 u99 <- xbar + 2.576 \* se  
 covers95 <- l95 <= mu & mu <= u95  
 covers99 <- l99 <= mu & mu <= u99  
 df <- data.frame(Sample, l99, l95, xbar, u95, u99)  
   
 p <- ggplot(df, aes(x=xbar, y=Sample)) + geom\_point() + geom\_vline(xintercept = mu) + xlim(range(c(l99,u99)))  
 p + geom\_vline(xintercept = mu + c(-2.576, -1.96, 1.96, 2.576)\*se, lty=2)



p + geom\_segment(aes(x = l99, y = Sample, xend = u99, yend = Sample, color = covers99))



p + geom\_segment(aes(x = l95, y = Sample, xend = u95, yend = Sample, color = covers95))



To formalize this approach we note that above we used the bootstrap to generate percentile confidence intervals. We also used the bootstrap standard error to create normal confidence intervals for those bootstrap distributions that appeared to be normal.

It turns out that because of the CLT, when we know the population standard deviation, , we can still use the normal approximation for the mean and sum. In this case, the standard error of the mean is . When the population standard deviation is not known, we can approximate it by using . When we make this substitution, we also substitute a t-distribution (on degrees of freedom) for the normal.

n <- length(htwt$Weight)  
 s.xbar <- sqrt(var(htwt$Weight)/n)  
 s.xbar

[1] 9.642395

s.boot <- sqrt(var(wt.mean.boot$t))  
 s.boot

[,1]  
[1,] 9.400818

mean(wt.mean.boot$t)

[1] 139.7484

wt.mean.boot

ORDINARY NONPARAMETRIC BOOTSTRAP  
  
  
Call:  
boot(data = htwt$Weight, statistic = b.mean, R = 9999)  
  
  
Bootstrap Statistics :  
 original bias std. error  
t1\* 139.6 0.1483898 9.400818

### Use the bootstrap std error  
 mean(htwt$Weight) + qnorm(c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995))\*s.boot

Warning in qnorm(c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995)) \* s.boot: Recycling array of length 1 in vector-array arithmetic is deprecated.  
 Use c() or as.vector() instead.

[1] 115.3851 121.1747 124.1370 155.0630 158.0253 163.8149

### Pretend that the sample std dev is the pop std dev  
 mean(htwt$Weight) + qnorm(c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995))\*s.xbar

[1] 114.7628 120.7013 123.7397 155.4603 158.4987 164.4372

### Compute the CI using the sample std dev  
 mean(htwt$Weight) + qt(c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995), n-1)\*s.xbar

[1] 112.0137 119.4182 122.9270 156.2730 159.7818 167.1863

boot.ci(wt.mean.boot)

Warning in boot.ci(wt.mean.boot): bootstrap variances needed for studentized  
intervals

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
Based on 9999 bootstrap replicates  
  
CALL :   
boot.ci(boot.out = wt.mean.boot)  
  
Intervals :   
Level Normal Basic   
95% (121.0, 157.9 ) (120.8, 157.2 )   
  
Level Percentile BCa   
95% (122.0, 158.4 ) (122.4, 159.3 )   
Calculations and Intervals on Original Scale

We see that the confidence intervals are slightly different. However, in this case, the differences are practically minimal — one or two pounds when looking at a mean of 139.6 pounds.

## Theoretical CIs

The old-school, theoretical approach to creating confidence intervals is based upon the CDF and its inverse. In what follows, remember that the confidence level is related to the hypothesis error type I error rate by .

### Proportion

For the proportion, we note that each of the outcomes can be coded as a “success” or “failure”. This leads us to the use of iid — or Bernoulli trials. The total number of “successes” is and . Further, and .

By the CLT, for “large” (variously *and* , or , etc.), . To obtain a level CI for , we approximate the variance using and look at

Thus, an equal tailed confidence interval for is .

### Mean

For iid and the CLT indicates that when is “large”, . When is known, and the or is large (), a level CI for can be computed as

Thus, a symmetric, two-tailed confidence interval for is .

### Variance

Recall that for iid , . With a bit of hand waving (note that and are ancillary, *etc.*), we see that . Thus

Thus, a two-tailed, confidence interval for is

## Sample Size Estimation

Prior to collecting data one might be required to determine a sample size that will support a certain margin of error (me) at a certain confidence level. A guess for can be made by noting that where is some measure of confidence based on an appropriate distribution and is the standard error of the estimator.

### Mean

The appropriate sample size for a confidence interval for when is known can be computed by recalling that . We need only solve for . Thus, we have

implies that we should choose at least as large as

### Proportion

For the proportion, , recall that

Again, solving for we get

Unfortunately, is unknown. Many statisticians plug in as this maximizes . Others prefer to use from a prior study or they run a pilot study.