

Confidence Intervals

Suppose we observe data X_1, \dots, X_n and estimate the expected value using \bar{X} .

There will be some estimation error between \bar{X} and the estimation target EX .

To provide a more complete description of the information in the data about EX , it is important to define a range of values around \bar{X} that is likely to contain EX .

This is a confidence interval.

The key concept behind a confidence interval is coverage.

Suppose we devise some procedure for constructing a confidence interval leading to the interval $\bar{X} \pm c$.

The coverage probability of this interval is

$$P(\bar{X} - c \leq EX \leq \bar{X} + c).$$

In words, this is the frequency over many replications that the interval contains the target value.

For example, suppose we use

$$\bar{X} \pm \text{range}(X_1, \dots, X_n) / 2\sqrt{n}$$

as a confidence interval for EX , where

$$\text{range}(X_1, \dots, X_n) = \max(X_i) - \min(X_i)$$

The following simulation estimates the coverage probability of this interval for standard exponential data.

```

## Sample sizes
N = c(10,30,50,70,90,110,130)

## Coverage probabilities for each sample size.
CP = array(0, length(N))

for (k in 1:length(N))
{
  n = N[k]
  X = array(rexp(n*1000), c(1000,n))

  ## Construct the CI.
  M = apply(X, 1, mean)
  MX = apply(X, 1, max)
  MN = apply(X, 1, min)
  C = (MX - MN)/(2*sqrt(n))

  ## Determine which intervals cover EX=1.
  ci = ((M-C < 1) & (M+C > 1))
}

```

```
## Calculate the proportion of intervals that cover.  
CP[k] = mean(ci)  
}
```

The results indicate that the coverage is around 0.77 for sample size 10, and increases to almost 0.99 for sample size 130.

A confidence interval should have the same coverage probability regardless of the sample size, so this procedure is not working well.

Confidence interval widths

The CI $\bar{X} \pm c$ has width $2c$.

The width of a confidence interval is related to its coverage probability – wider confidence intervals have higher coverage probabilities, narrower confidence intervals have lower coverage probabilities.

Constructing a CI for EX

Begin by standardizing the sample mean

$$Z = \sqrt{n} \frac{\bar{X} - \mu}{\sigma}.$$

This expected value of Z is zero and its variance is 1.

If \bar{X} is approximately normal, then Z is approximately normal.

The central limit theorem (CLT) tells us that \bar{X} will be approximately normal if the sample size is not too small.

If

$$Z = \sqrt{n} \frac{\bar{X} - \mu}{\sigma}.$$

is approximately normal, then

$$P(-1.96 \leq \sqrt{n} \frac{\bar{X} - \mu}{\sigma} \leq 1.96) = 0.95.$$

Rearranging yields

$$P(-1.96\sigma/\sqrt{n} \leq \bar{X} - \mu \leq 1.96\sigma/\sqrt{n}) = 0.95,$$

and finally

$$P(\bar{X} - 1.96\sigma/\sqrt{n} \leq \mu \leq \bar{X} + 1.96\sigma/\sqrt{n}) = 0.95.$$

Thus the rule

$$\bar{X} \pm 1.96\sigma/\sqrt{n}$$

provides an approximate 95% confidence interval as long as the variance is known and the sample size is large enough for the central limit theorem to apply.

If a coverage level other than 95% is desired, only the constant 1.96 need be changed. For example, to get 90% coverage use

$$\bar{X} \pm 1.64\sigma/\sqrt{n}.$$

The following simulation estimates the coverage probability of this interval for normal data.

```
## Sample sizes.
N = c(10,30,50,70,90,110,130)

## Coverage probabilities.
CP = NULL

## Loop over the sample sizes.
for (k in 1:length(N))
{
  n = N[k]
  X = array(rnorm(n*1e4), c(1e4,n))

  ## Construct the CI.
  M = apply(X, 1, mean)
  C = 1.96/sqrt(n)

  ## Determine which intervals cover.
  ci = ((M-C < 0) & (M+C > 0))
}
```

```
## Calculate the proportion of intervals that cover.  
CP[k] = mean(ci)  
}
```

The results in a particular run of this program were

0.948 0.948 0.944 0.939 0.949 0.957 0.951

indicating that the coverage is quite accurate.

If the same program is run using the exponential distribution in place of the normal distribution (remember that the target value is 1 rather than 0 in that case), the coverage probabilities will continue to be around 0.95.

Nuisance parameters

The confidence interval

$$\bar{X} \pm 1.96\sigma/\sqrt{n}$$

requires the population variance σ^2 , which usually is not known.

A parameter such as σ^2 that is needed for constructing the CI but that is not the value being estimated is called a nuisance parameter.

A standard approach for handling nuisance parameters is to choose a reasonable estimate, then substitute the estimate in place of the population value. This is called the “plug-in” approach.

For example, we could plug-in the usual estimate $\hat{\sigma}$ in place of σ .

Plug-in CI's work well if the sample size is large.

If the sample size is small or moderate, they tend to cover at less than the "nominal level." For example a CI with nominal 95% coverage may have 88% coverage if the plug-in method is used.

The following simulation calculates coverage probabilities for the the plug-in confidence interval

$$\bar{X} \pm 1.96\hat{\sigma}/\sqrt{n}.$$

```
## Sample sizes.  
N = c(5,10,30,50,70,90,110,130)  
  
## Coverage probabilities.  
CP = NULL  
  
for (k in 1:length(N))  
{  
  n = N[k]  
  X = array(rnorm(n*1000), c(1000,n))  
  
  ## Estimate the standard deviation from the data.  
  SD = apply(X, 1, sd)  
  
  ## Construct the CI.
```

```
M = apply(X, 1, mean)
C = 1.96*SD/sqrt(n)

## Determine which intervals cover.
ci = ((M-C < 0) & (M+C > 0))

## Calculate the proportion of intervals that cover.
CP[k] = mean(ci)
}
```

Based on the above simulation, you will find that the coverage probabilities for sample size $n = 5$ and $n = 10$ are much lower than 0.95. For larger sample sizes, the coverage is about right.

What can be done to improve the performance for small sample sizes?

In the special case where the data are approximately normal, it is possible to mathematically compensate for the effects of plugging in the estimated value $\hat{\sigma}$.

The CLT does not say anything about whether data are normal.

For approximately normal data it turns out that

$$\sqrt{n} \frac{\bar{X} - \mu}{\hat{\sigma}}$$

has a t-distribution with $n - 1$ degrees of freedom. Thus

$$P(\bar{X} - q\sigma/\sqrt{n} \leq \mu \leq \bar{X} + q\sigma/\sqrt{n}) = 0.95,$$

where q is set to the 97.5 percentile of the appropriate t_{n-1} distribution.

To get this value in R, use `qt(0.975, n-1)`.

As the degrees of freedom gets larger, the p^{th} quantile of the t distribution gets closer to the p^{th} quantile of the standard normal distribution.

```
## Demonstrate that t-distribution quantiles get closer to the
## standard normal quantile as the degrees of freedom increases.

M = array(0, c(100,2))

q = 0.975  ## The quantile to analyze.

## Cycle through various degrees of freedom.
for (d in 1:100) {

  ## The qth quantile of the t-distribution with d degrees of freedom.
  M[d,1] = qt(q, d)

  ## The qth quantile of the standard normal distribution
  ## (doesn't depend on d).
  M[d,2] = qnorm(q)
}
```

The following simulation estimates the coverage probability.

```
## Sample sizes.
N = c(5,10,30,50,70,90,110,130)

## Coverage probabilities.
CP = NULL

for (k in 1:length(N))
{
  n = N[k]

  X = array(rnorm(n*1e4), c(1e4,n))

  ## Estimate sigma from the data.
  SD = apply(X, 1, sd)

  ## Construct the CI, correctly adjusting for the
  ## uncertainty in SD.
  M = apply(X, 1, mean)
  C = qt(0.975,n-1)*SD/sqrt(n)
```

```
## Determine which intervals cover.
```

```
ci = ((M-C < 0) & (M+C > 0))
```

```
## Calculate the proportion of intervals that cover.
```

```
CP[k] = mean(ci)
```

```
}
```

Bootstrap confidence intervals

The idea behind the bootstrap is to use the data we have to produce artificial data sets that are similar to what we would get under replication.

If we want a CI for EX , then for each artificial data set $X^{(k)}$ we can compute the sample mean $\bar{X}^{(k)}$.

The artificial data sets must be constructed so that the variation in $\bar{X}^{(k)}$ approximates the sampling distribution of \bar{X} under actual replication.

Once we have achieved this, we can use the sample 2.5 and 97.5 percentiles of the $\bar{X}^{(k)}$ as the lower and upper bounds of a confidence interval.

For the standard bootstrap, the data values are sampled with replacement from the actual data (this is called “resampling”).

For example, if the actual data were 1, 2, 3, and 4, the rows of the following table would each be a possible bootstrapped data set:

Artificial data set 1:	2	4	4	1
Artificial data set 2:	4	1	2	3
Artificial data set 3:	1	3	2	3
Artificial data set 4:	3	3	3	3
Artificial data set 5:	2	1	4	1
Artificial data set 6:	3	1	2	1

In the bootstrap, the artificial data sets are the same size as the actual data set. In any given data set, some of the actual data values may be repeated, and others may be missing.

The following code fragment generates 10,000 bootstrapped data sets from the list X, placing them in the columns of B.

```
p = length(X) ## The sample size of the actual data.
nrep = 1e4 ## The number of bootstrap samples to generate.
B = array(0, c(nrep,p)) ## Each row of B holds a bootstrap sample.

for (r in 1:nrep)
{
  ii = ceiling(p*runif(p))
  B[r,] = X[ii]
}
```

This is a bit slow in R due to the loop. The following approach is much faster.

```
nrep = 1e4
p = length(X)
ii = ceiling(p*runif(p*nrep))
B = X[ii]
B = array(B, c(nrep,p))
```

The following code gives you a 95% confidence interval from the bootstrapped data sets:

```
M = apply(B, 1, mean)
M = sort(M)
C = c(M[25], M[975])
```

The bootstrap is easy to apply. But it does not necessarily have good coverage properties. We need to check.

```
## Sample sizes.
N = c(10,20,40,60)

nrep = 1000 ## Number of simulation replications per sample size value.
nboot = 1000 ## The number of bootstrap data sets.

## Coverage probabilities.
CP = NULL

for (j in 1:length(N))
{
  ## Keep track of how many times the interval covers the true value.
  nc = 0

  n = N[j]

  for (k in 1:nrep)
  {
```

```
## Simulate a data set.
X = rnorm(n)

## Generate bootstrap data sets from X.
ii = ceiling(n*runif(n*nboot))
B = X[ii]
B = array(B, c(nboot,n))

## Get the sample mean for each bootstrap data set.
M = apply(B, 1, mean)
M = sort(M)

## Get the confidence interval lower and upper bound.
C = c(M[25], M[975])

## Check for coverage.
if ( (C[1] < 0) & (C[2] > 0) ) { nc = nc+1 }
}

CP[j] = nc/nrep
```

}

A major advantage of the bootstrap is that it can be applied to any estimation problem, not just estimation of the expected value.

The following simulation assesses the performance of bootstrap confidence intervals for the population standard deviation based on the sample standard deviation. Note that only two lines differ from the previous program.

```
## Sample sizes.
N = c(10,20,40,60)

nrep = 1000 ## Number of simulation replications per sample size value.
nboot = 1000 ## The number of bootstrap data sets.

## Coverage probabilities.
CP = NULL

for (j in 1:length(N))
{
  ## Keep track of how many times the interval covers the true value.
  nc = 0
```

```
n = N[j]

for (k in 1:nrep)
{
  ## Simulate a data set.
  X = rnorm(n)

  ## Generate bootstrap data sets from X.
  ii = ceiling(n*runif(n*nboot))
  B = X[ii]
  B = array(B, c(nboot,n))

  ## Get the sample standard deviation for each bootstrap data set.
  M = apply(B, 1, sd)
  M = sort(M)

  ## Get the confidence interval lower and upper bound.
  C = c(M[25], M[975])

  ## Check for coverage.
  if ( (C[1] < 1) & (C[2] > 1) ) { nc = nc+1 }
```

```
}  
CP[j] = nc/nrep  
}
```

The parametric bootstrap

The parametric bootstrap is a variant of the “non-parametric” bootstrap, presented above.

The parametric bootstrap is used if the distributional family of the data is considered known (e.g. normal, exponential).

Like the non-parametric bootstrap, the parametric bootstrap is most useful when a statistic other than the expected value is of interest. In this case, no simple formula such as σ^2/n exists for producing a confidence interval.

There are three main steps to the parametric bootstrap:

1. Use the observed data to estimate the parameters of the parametric distribution. For example, if the data are thought to be normally distributed, then the mean and variance must be estimated.
2. A large number of artificial data are drawn from the estimated parametric distribution.
3. Calculate your statistic of interest on each artificial data set. Then sort the values and use the appropriate quantiles to define your CI.

The following code fragment generates 1000 normal parametric bootstrap samples from the data in **X**:

```
p = length(X)
mu = mean(X)
s = sd(X)
B = mu + s*rnorm(1000*p)
B = array(B, c(p,1000))
```

Here is a simulation that assesses the coverage properties of the parametric bootstrap for the population median.

```
## Sample sizes.
N = c(5,10,20,40,60)

nrep = 1000 ## Number of simulation replications
nboot = 1000 ## Number of bootstrap samples

## The coverage probabilities.
CP = NULL

for (j in 1:length(N))
{
  n = N[j]

  ## Keep track of how many times the interval covers the true value.
  nc = 0

  for (k in 1:nrep)
  {
```

```
## Simulate a data set.  
X = rnorm(n)  
  
## Generate 1000 bootstrap data sets from X.  
mu = mean(X)  
s = sd(X)  
B = mu + s*rnorm(nboot*n)  
B = array(B, c(nboot,n))  
  
## Get the sample median for each bootstrap data set.  
M = apply(B, 1, median)  
M = sort(M)  
  
## Get the confidence interval lower and upper bound.  
C = c(M[25], M[975])  
  
## Check for coverage.  
if ( (C[1] < 0) & (C[2] > 0) ) { nc = nc+1 }  
}  
  
## Save the estimated coverage probability.
```

```
CP[j] = nc/nrep  
}
```