

## Confidence Intervals

Suppose we observe data  $X_1, \dots, X_n$  and estimate the expected value using  $\bar{X}$ . There will be some error between  $\bar{X}$  and the estimation target  $EX$ . In order 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 called 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

$$\bar{X} \pm (\max(X_i) - \min(X_i))/2\sqrt{n}$$

as a confidence interval for  $EX$ . The following simulation estimates the coverage probability of this interval for exponential data.

```

## Keep track of coverage probabilities for different sample sizes here.
CP <- NULL

for (n in c(10,30,50,70,90,110,130))
{
  X <- array(rexp(n*1000), c(n,1000))

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

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

  ## Calculate the proportion of intervals that cover.
  cp <- mean(ci)

  CP <- c(CP, cp)
}

```

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.

Here are a couple of key points to understand about confidence intervals.

- For a given data set, a rule either covers the target value or it does not. In practice, we cannot know whether the interval covers in a particular instance. However may be able to estimate its coverage probability.
- Covering the true value is a good thing, but the goal is not simply to maximize coverage. If that were the case, we could use a very wide confidence

interval like  $\bar{X} \pm 1000$  or even  $\bar{X} \pm \infty$ . Such wide confidence intervals are of no practical use. Therefore we allow the interval to not cover the true value a small proportion of the time. If this proportion is 5%, the interval covers the true value 95% of the time.

- 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.

To construct a confidence interval for  $EX$ , begin by standardizing the sample mean:

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

This quantity has expected value zero and variance 1. If we are further willing to assume that this quantity is approximately normal in distribution (invoking the central limit theorem if the sample size is not too small), we can conclude that

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

Here we are using the fact that a standard normal random variable has probability 0.95 of falling between -1.96 and 1.96.

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.

```
## Keep track of coverage probabilities for different sample sizes here.
CP <- NULL

for (n in c(10,30,50,70,90,110,130))
{
  X <- array(rnorm(n*1e4), c(n,1e4))

  ## Construct the CI.
  M <- apply(X, 2, 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 <- mean(ci)

  CP <- c(CP, cp)
}
```

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.

It is desirable that the width of a confidence interval be as narrow as possible for a given level of coverage (e.g. 95%). The width of the confidence interval for  $\bar{X} \pm 1.96\sigma/\sqrt{n}$  is  $2 \cdot 1.96\sigma/\sqrt{n}$ . Nothing can be done about the constant  $2 \cdot 1.96$ . To get a shorter interval, either decrease  $\sigma$  (by using a more precise measuring instrument) or increase the sample size  $n$ .

## Nuisance parameters

The confidence interval

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

contains the population variance  $\sigma^2$ , which usually is not known. A parameter such as  $\sigma^2$  that is needed for a particular calculation but that is not a primary object of interest 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 (i.e. use  $\hat{\sigma}$  in place of  $\sigma$ ). Using the sample variance  $\hat{\sigma}^2$  in place of  $\sigma^2$  in a 95% confidence interval for  $EX$  yields

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

This “plug-in” approach may change the coverage properties of the interval. The “nominal coverage probability” (e.g. 95%) will generally differ from the actual coverage probability. This is more of a problem when the sample size is small, and the nuisance parameter estimate is likely to deviate a lot from the true value.

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

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

```

## Keep track of coverage probabilities for different sample sizes here.
CP <- NULL

for (n in c(10,30,50,70,90,110,130))
{
  X <- array(rnorm(n*1000), c(n,1000))

  ## Construct the CI.
  M <- apply(X, 2, mean)
  SD <- apply(X, 2, sd)
  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 <- mean(ci)

  CP <- c(CP, cp)
}

```

You will find that the coverage probability for sample size  $n = 10$  is lower than 0.95. For larger sample sizes, the coverage is about right.

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  $\hat{\sigma}$ . 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-distribution. To get this value in R, use `qt(0.975, n-1)`. Note that as  $n$  gets large, this quantile gets

extremely close to 1.96, which is the value used in the plug-in interval. However for smaller values of  $n$ ,  $qt(0.975, n-1)$  is larger than 1.96. The wider interval compensates for the uncertainty about  $\sigma^2$ .

The following simulation estimates the coverage probability.

```
## Keep track of coverage probabilities for different sample sizes here.
CP <- NULL

for (n in c(10,30,50,70,90,110,130))
{
  X <- array(rnorm(n*1000), c(n,1000))

  ## Construct the CI.
  M <- apply(X, 2, mean)
  SD <- apply(X, 2, sd)
  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 <- mean(ci)

  CP <- c(CP, cp)
}
```

Using the  $t$  distribution to compensate for uncertainty in  $\hat{\sigma}$  is mathematically justified if the data are normal. For most non-normal distributions, plug-in confidence intervals tend to have lower than nominal coverage for small sample sizes. Using a larger multiplier than 1.96 therefore makes sense. However there is no reason that for non-normal data, the particular multipliers given by the  $t$ -distribution will give the desired coverage.

## 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. For each artificial data set  $X^{(k)}$  we can compute the sample mean  $\bar{X}^{(k)}$ . Under replication, the value of  $\bar{X}$  would vary to some degree. The artificial data sets are constructed so that the  $\bar{X}^{(k)}$  vary to approximately the same degree. Once we have a reasonable number of simulated  $\bar{X}^{(k)}$  values, 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 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 1000 bootstrapped data sets from the list  $X$ , placing them in the columns of  $B$ .

```
p <- length(X)
B <- array(0, c(p,1000))

for (r in (1:1000))
{
  ii <- ceiling(p*runif(p))
  B[,r] <- X[ii]
}
```

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

```

p <- length(X)
ii <- ceiling(p*runif(p*1000))
B <- X[ii]
B <- array(B, c(p,1000))

```

To get a confidence interval from the bootstrapped data sets, the following would be used:

```

M <- apply(B, 2, mean)
M <- sort(M)
C <- c(M[25], M[975])

```

The bootstrap is easy to apply. But it is not a foregone conclusion that it has good coverage properties. Next we will investigate the coverage of bootstrap confidence intervals using simulations.

```

CP <- NULL

for (n in c(10,20,40,60))
{
  ## Keep track of how many times the interval covers the true value.
  nc <- 0

  for (k in (1:1000))
  {
    ## Simulate a data set.
    X <- rnorm(n)

    ## Generate 1000 bootstrap data sets from X.
    ii <- ceiling(n*runif(n*1000))
    B <- X[ii]
    B <- array(B, c(n,1000))

    ## Get the sample mean for each bootstrap data set.
    M <- apply(B, 2, 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 }
}

## Save the estimated coverage probability.
CP <- c(CP, nc/1000)
}

```

You will find that the coverage is close to the 95% nominal level, although it is a bit low for small sample sizes. This is known to be an issue with the bootstrap. You should also check the performance when the data are exponential.

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.

```

CP <- NULL

for (n in c(10,20,40,60))
{
  ## Keep track of how many times the interval covers the true value.
  nc <- 0

  for (k in 1:1000)
  {
    ## Simulate a data set.
    X <- rnorm(n)

    ## Generate 1000 bootstrap data sets from X.
    ii <- ceiling(n*runif(n*1000))
    B <- X[ii]
    B <- array(B, c(n,1000))

    ## Get the sample standard deviation for each bootstrap data set.
    M <- apply(B, 2, 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 }
  }

  ## Save the estimated coverage probability.
  CP <- c(CP, nc/1000)
}

```

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), but a statistic other than the expected value is of interest. In this case, no simple formula such as  $\sigma^2/n$  exists for producing a confidence interval.

In the parametric bootstrap, the data are used to estimate the parameters of a parametric distribution. Then artificial data are drawn from the parametric distribution. For example, if the data are assumed to be normal, the sample mean and sample variance of the actual data are first calculated. Then artificial data sets are generated from a normal distribution with the calculated mean and variance. The artificial data sets each have the same size as the actual data.

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.

```

CP <- NULL

for (n in c(10,20,40,60))
{
  ## Keep track of how many times the interval covers the true value.
  nc <- 0

  for (k in 1:1000)
  {
    ## Simulate a data set.
    X <- rnorm(n)

    ## Generate 1000 bootstrap data sets from X.
    mu <- mean(X)
    s <- sd(X)
    B <- mu + s*rnorm(1000*n)
    B <- array(B, c(n,1000))

    ## Get the sample median for each bootstrap data set.
    M <- apply(B, 2, 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 <- c(CP, nc/1000)
}

```

## Summary

How do we select which approach to use to construct a CI? The following is a good strategy.

- Estimating  $EX$ ?
  - Large sample size.
    - \*  $\bar{X} \pm 1.96\hat{\sigma}/\sqrt{n}$
  - Data are known to be approximately normal and sample size is small.
    - \*  $\bar{X} \pm q\hat{\sigma}/\sqrt{n}$ , where  $q$  is the 97.5 percentile of the  $t(n-1)$  distribution
  - Data are thought likely to follow a specific non-normal distribution (e.g. exponential) and the sample size is small.
    - \* Parametric bootstrap
  - Data are thought likely to be non-normal but not much is known about the specific distribution, and the sample size is small.
    - \* Non-parametric bootstrap
- Estimating a value other than  $EX$ ?
  - Data are likely to be non-normal.
    - \* Non-parametric bootstrap
  - Data distribution considered known (e.g. normal).
    - \* Parametric bootstrap
    - \* Search the literature for an appropriate formula.