

Resampling Methods

Clay Ford

November 2014

Traditional Statistical Inference

To estimate the mean of a population:

- ▶ Take a random sample from the population
- ▶ Calculate the mean of the sample to estimate population mean
- ▶ Assess variability of sample mean by calculating standard error: $\frac{s}{\sqrt{n}}$, where s is the standard deviation of the sample

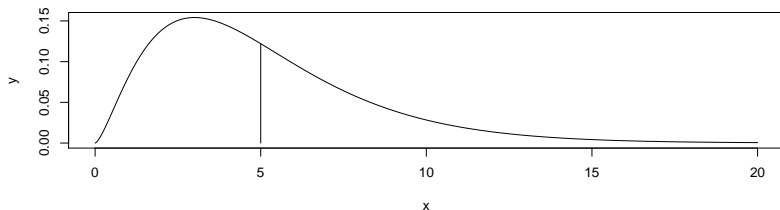
The standard error gives us an idea of how accurate our estimate is.

Two items of note about the standard error:

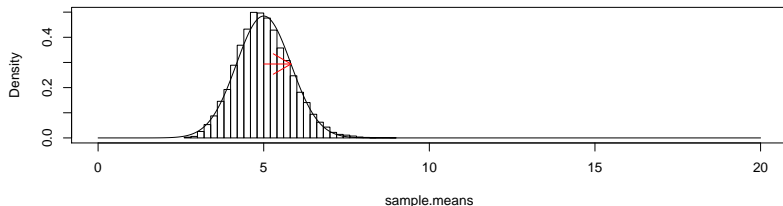
1. the concise formula
2. it assumes the distribution of the mean is normal

Illustration of Sampling Distribution of mean

A Skewed Population with mean=5



Simulated Sampling Distribution of mean for n=15



red arrow is standard deviation of sampling distribution; the standard error estimates it

What about other statistics?

For example, median, trimmed mean, ratio, correlation, standard deviation, error rate, . . .

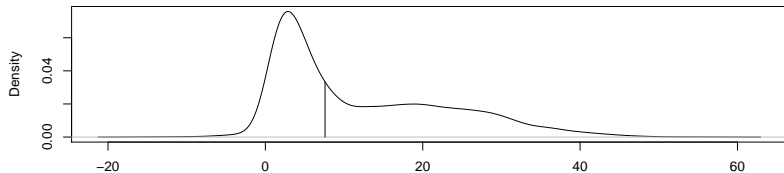
Sample means are essentially the only statistic with an easy-to-obtain standard error formula.

Deriving standard error formulas for other statistics can be tedious or even impossible.

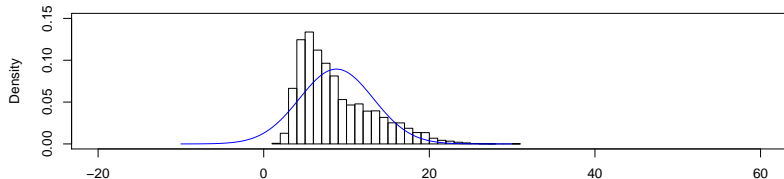
Derivation of standard error formulas also requires making (possibly incorrect) assumptions about the shape of the sampling distribution.

Sampling Distribution of Median

A skewed population



Simulated Sampling Distribution of median for n=15



Sampling distribution is not normally distributed. How do we calculate standard error?

Introducing Resampling Methods

Replace complicated or inaccurate approximations with *computer simulations*. Also known as “Bootstrapping Methods”.

- ▶ Use computer to resample from *original random sample* to create replicate datasets (bootstrap samples)
- ▶ Calculate statistic of interest for *each* replicate dataset (bootstrap replications)
- ▶ Summarize bootstrap replications (example: take standard deviation to estimate standard error)

Resampling methods can assess other accuracy measures such as bias, prediction errors, and confidence intervals.

Example of Bootstrap Sample

1. *resample with replacement*
2. resample the same amount as original data

Example using R:

```
# original sample data (n=10)
myData <- c(18,18,29,20,11,12,16,25,24,21)
# resample data (n=10)
sample(myData, replace=TRUE)
```

```
## [1] 29 29 29 11 20 21 18 12 24 21
```

Example of Bootstrap Replication

Calculate statistic of interest for each bootstrap sample.

```
# median of original sample  
median(myData)
```

```
## [1] 19
```

```
# median of resample  
median(sample(myData, replace=TRUE))
```

```
## [1] 20.5
```

We want to do this at least 200 times; in practice we use 999 or more.

Example of 200 Bootstrap Replications

```
# resample data 200 times, take median each time  
bout <- replicate(200,  
                  median(sample(myData, replace=T)))  
# bootstrap estimate of standard error:  
sd(bout)
```

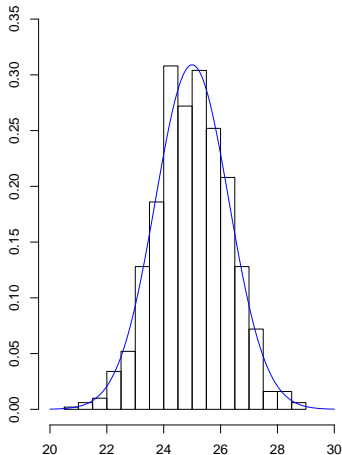
```
## [1] 2.231856
```

Note we'll get a slightly different answer each time we do this.

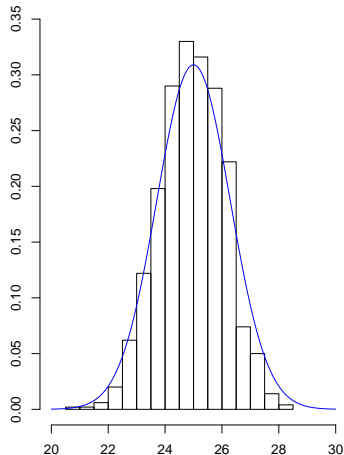
This was a toy example. We'll use the boot package in R for formal resampling procedures.

Sampling Distribution vs. Bootstrap Distribution

**Simulated Sampling Distribution
of mean from $N(25,5)$
 $n=15$**



**Bootstrap Distribution
of mean using one sample of $n=15$
from $N(25,5)$**



Limitations of the Bootstrap

The bootstrap requires that the sample serve as a surrogate for the population.

The bootstrap **will fail** if:

- ▶ the original sample is not random
- ▶ the original sample is biased
- ▶ the sample objects are not independent of one another

Again, the original sample must be representative of the population!

Bootstrap Algorithm for Estimating Standard Errors

1. Select B independent bootstrap samples $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \dots, \mathbf{x}^{*B}$ each consisting of n data values drawn with replacement from **the original data set**
2. Evaluate the bootstrap replication corresponding to each bootstrap sample,

$$\hat{\theta}^*(b) = s(\mathbf{x}^{*b}) \quad b = 1, 2, \dots, B$$

3. Estimate the standard error by the sample standard deviation of the B replications:

$$\hat{se}_B = \sqrt{\sum_{b=1}^B [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2 / (B - 1)}$$

where $\hat{\theta}^*(\cdot) = \sum_{b=1}^B \hat{\theta}^*(b) / B$

Implementing Bootstrap Algorithms in R

Use the `boot` package.

- ▶ Comes with base R installation.
- ▶ Returns a convenient `boot` object with associated `print` and `plot` methods
- ▶ preserves bootstrap replicates
- ▶ allows easy calculation of confidence intervals
- ▶ implements variants on the bootstrap (stratified sampling, time series, censored data)

One catch: user must supply a function.

Let's go to R.

Bias

Another useful measure of statistical accuracy is *Bias*.

Bias is the difference between the expectation of an estimator $\hat{\theta}$ and the quantity θ being estimated.

Large bias is almost always undesirable.



Biased



Unbiased

Bootstrap Estimate of Bias

The bootstrap estimate of bias is

$$\widehat{bias}_B = \hat{\theta}^*(\cdot) - \hat{\theta}$$

where $\hat{\theta}^*(\cdot) = \sum_{b=1}^B \hat{\theta}^*(b) / B$

In other words, take the mean of the bootstrap replications and subtract the original estimate.

As a rule of thumb, bias less than 0.25 standard errors can be ignored. ($|\widehat{bias} / \widehat{se}| < 0.25$)

Large bias may be an indication that your statistic is not an appropriate estimator of the population parameter.

Let's go to R.

The Jackknife

The jackknife, first proposed in the 1950's, is the original computer-based method for estimating biases and standard errors.

The jackknife focuses on the samples that *leave out one observation at a time*:

$$\mathbf{x}_{(i)} = (x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$$

The i th jackknife sample consists of the data set with the i th observation removed. For example, $\mathbf{x}_{(1)}$ is the data set with the first observation removed.

$\hat{\theta}_{(i)}$ is the i th jackknife replication of $\hat{\theta}$. $\hat{\theta}_{(1)}$ is the jackknife estimate with the first observation removed.

The Jackknife Estimates of Bias and Standard Error

The jackknife estimate of bias:

$$\widehat{bias}_{jack} = (n - 1)(\hat{\theta}_{(\cdot)} - \hat{\theta})$$

where $\hat{\theta}_{(\cdot)} = \sum_{i=1}^n \hat{\theta}_{(i)} / n$.

The jackknife estimate of standard error:

$$\widehat{se}_{jack} = \sqrt{\frac{n-1}{n} \sum (\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)})^2}$$

The jackknife provides a simple approximation to the bootstrap for estimation of bias and standard errors.

Let's go to R.

The Jackknife-after-Bootstrap Plot

A common use of the jackknife is for bootstrap diagnostics.

The jackknife-after-bootstrap plot looks at bootstrap samples in which the i th point did not appear.

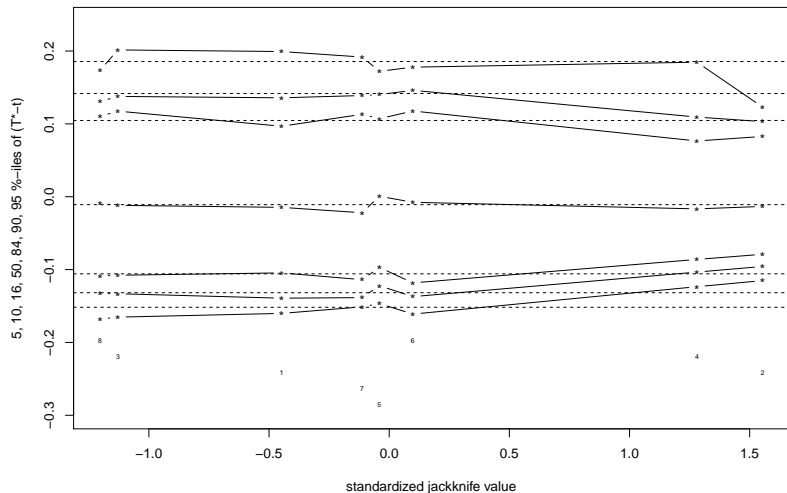
The plot shows the sensitivity of the statistic and of the percentiles of its bootstrapped distribution to deletion of individual observations.

Centered quantiles are generated for all jackknife samples and plotted against *standardized jackknife influence values*:

$$I_{jack,j} = (n - 1)(\hat{\theta}_{(.)} - \hat{\theta}_{(i)}) / \sqrt{\text{var}(\hat{\theta}_{(i)})}$$

R Syntax: `jack.after.boot(boot.object)` or
`plot(boot.object, jack=TRUE)`

The Jackknife-after-Bootstrap Plot - Example



Let's go to R.

Confidence Intervals

Standard errors are often used to assign approximate confidence intervals to a parameter of interest.

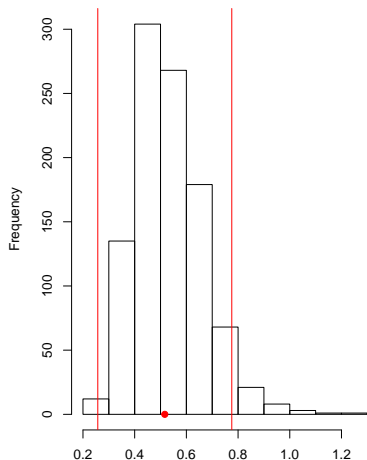
For example, given a parameter estimate $\hat{\theta}$ and an estimated standard error \hat{se} , the usual 95% confidence interval for θ is

$$\hat{\theta} \pm 1.96 \cdot \hat{se}$$

This gives us a best guess for θ and how far in error that guess might be (*assuming a symmetric distribution*).

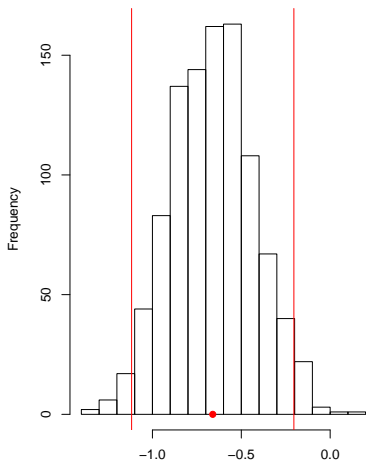
Symmetric versus Asymmetric Distributions

Asymmetric (requires transformation)



Standard CI not accurate

Symmetric (after transformation)



Standard CI works well

Bootstrap Confidence Intervals

A goal of bootstrap methods is to produce dependable confidence intervals regardless of the distribution shape.

- ▶ match classic confidence intervals when such intervals are appropriate
- ▶ provide accurate coverage without transforming data

We'll look at two types:

1. Percentile
2. BC_a

The Percentile Interval

How it works: use percentiles of the bootstrap histogram to define confidence limits.

Example: For a 95% percentile interval, the lower bound is the 2.5% percentile and the upper bound the 97.5% percentile.

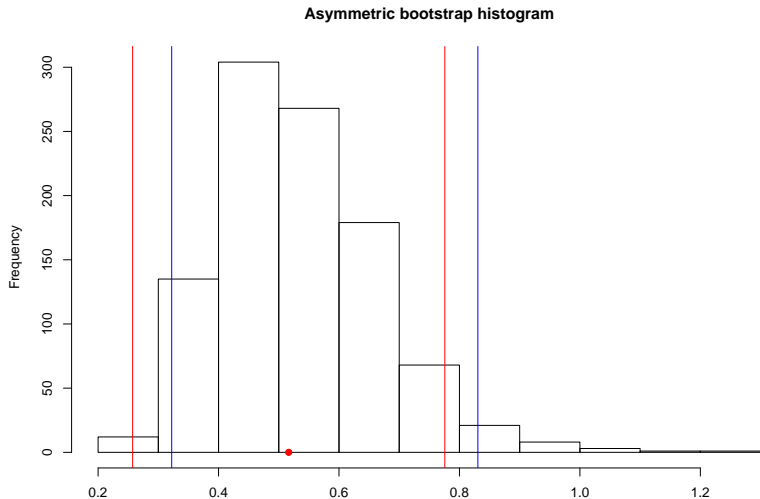
If the bootstrap distribution is roughly normal, then the classic (standard normal) and percentile intervals will nearly agree.

The percentile method provides good coverage for asymmetric distributions and doesn't require transforming data.

R code

```
boot.ci(boot.out, conf = 0.95, type = "perc")
```

The Percentile Interval - Example



red lines = classic CI interval; blue lines = percentile interval

The BC_a Interval

An improved version of the percentile method. Stands for *Bias-corrected and accelerated*.

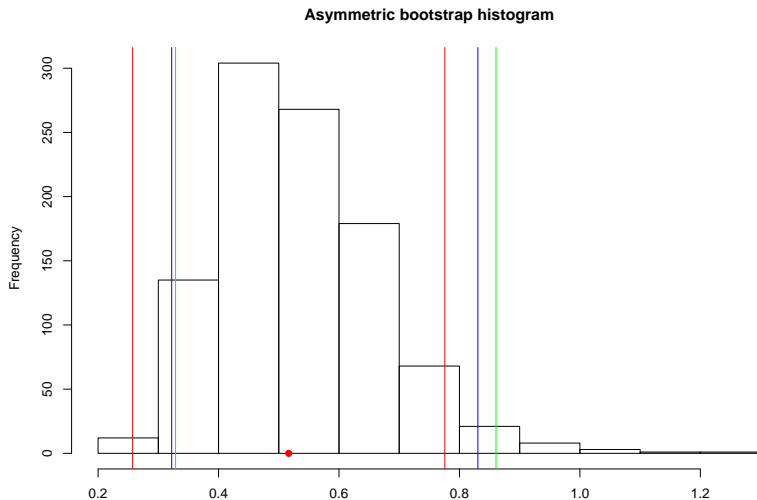
Provides better coverage than the percentile method, automatically corrects for bias in the point estimate, and is preferred in practice.

More difficult to explain than the percentile interval, but not much more difficult to calculate.

R code

```
boot.ci(boot.out, conf = 0.95, type = "bca")
```

The BC_a Interval - Example



red lines = classic CI interval; blue lines = percentile interval; green lines = bca interval

The BC_a method

The BC_a interval endpoints are also given by percentiles of the bootstrap distribution, but the percentiles are computed using a sophisticated method.

$$BC_a : (\hat{\theta}^{*(\alpha_1)}, \hat{\theta}^{*(\alpha_2)})$$

where

$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^\alpha}{1 - \hat{a}(\hat{z}_0 + z^\alpha)}\right)$$

α_2 is same as α_1 except $1 - \alpha$ replaces α .

\hat{a} is the acceleration and \hat{z}_0 is the bias-correction.

The BC_a method - continued

- ▶ Acceleration refers to the rate of change of the SE of $\hat{\theta}$ with respect to θ .
- ▶ Bias refers to proportion of bootstrap replications less than $\hat{\theta}$.

The formulas for the acceleration and the bias-correction are complicated. See Efron, B. (1987) Better bootstrap confidence intervals (with Discussion). *Journal of the American Statistical Association*, **82**, 171-200.

Let's go to R.

Bootstrapping Regressions

We can bootstrap standard errors of regression coefficients. Two ways:

1. *Case resampling*: select R bootstrap samples of the data (i.e. the rows) and fit a model to each sample.
2. *Residual resampling*: fit a model to original data set and select R bootstrap samples of residuals to create new responses. We resample residuals and add to fitted response values matched to original corresponding observed predictors.

Bootstrapping regressions makes sense for models fit with methods other than least-squares.

Example of Case Resampling

```
# cars: speed of cars and the distances taken to stop  
head(cars, n=4)
```

```
##    speed dist  
## 1      4    2  
## 2      4   10  
## 3      7    4  
## 4      7   22
```

```
s <- sample(nrow(cars), replace=TRUE)  
cars[s[1:4],]
```

```
##    speed dist  
## 28     16   40  
## 5      8   16  
## 30     17   40  
## 39     20   32
```

Example of Residual Resampling

```
mod <- lm(dist ~ speed, data = cars) # fit model
rr  <- resid(mod)[s] # resample residuals
cbind(cars$speed[1:4], fitted(mod)[1:4] + rr[1:4])
```

```
##      [,1]      [,2]
## 1      4 -7.1889051
## 2      4  0.2703650
## 3      7  0.6759124
## 4      7 -19.1213139
```

Case Resampling vs. Residual Resampling

If predictors are fixed (not random) values, may make more sense to use residual resampling.

Residual resampling enforces the strong assumption that errors are identically distributed.

Case resampling is less sensitive to assumptions, therefore it may be preferable in some cases.

The car package provides a handy function for bootstrapping regression models: `Boot`

```
Boot(model, R=1000, method="case")
```

```
Boot(model, R=1000, method="residual")
```

Let's go to R.

Cross-Validation

Cross-validation is a resampling method for estimating prediction error.

Prediction error measures how well a model works.

Assessing this with data used to build model leads to underestimating error rate.

However we don't usually have new data to test the model.

To get around this, cross-validation uses part of the available data to fit the model and a different part to test it.

How Cross-Validation Works

1. Randomly split data into K roughly equal-sized parts.
(Typically $K = 5$ or 10)
2. Hold out the k th part and fit the model with the other $K - 1$ parts.
3. Calculate the prediction error of the fitted model using the k th part.
4. Do the above for $k = 1, 2, \dots, K$ and combine the K estimates of prediction error for regression and classification, respectively:
 - ▶ $CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i$ where $MSE_i = \frac{1}{n_i} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ (Mean Square Error)
 - ▶ $CV_{(k)} = \frac{1}{k} \sum_{i=1}^k Err_i$ where $Err_i = \frac{1}{n_i} \sum_{i=1}^n I(y_i \neq \hat{y}_i)$ (Classification Error Rate)

Cross-Validation Notes

When $K = n$ we call this “leave-one-out” cross-validation (LOOCV).

Using $K = 5$ or $K = 10$ often gives more accurate estimates of prediction error and requires less computation.

The `boot` package provides the `cv.glm` function to calculate K-fold cross-validation prediction error, which requires fitting a model using `glm`.

Cross validation also works for more complex modeling methods.

Let's go to R.

References

Canty, A.J. (2002) *Resampling methods in R: the boot package*. R News. 2/3, 2-7.

Davison, A.C. and Hinkley, D.V. (1997) *Bootstrap methods and their applications*. Cambridge University Press, Cambridge.

Davison, A.C. and Kuonen, D. (2002) *An introduction to the Bootstrap with applications in R*. Statistical Computing & Statistical Graphics Newsletter. 13, 6-11.

Diaconis, P. and Efron, B. (1983) *Computer-Intensive Methods in Statistics*. Scientific American. Vol 248, No 5, 116-130.

Efron, B. and Tibshirani, R.J. (1993) *An Introduction to the Bootstrap*. Chapman and Hall, London, New York.

References (cont'd)

Fox, J. and Weisberg, S. (2012) *Bootstrapping regression models in R, an appendix to an R Companion to Applied Regression*, 2nd ed. Sage, London.

James, G. et al. (2013) *An Introduction to Statistical Learning*. Springer, New York.

Maindonald, J. and Braun, J.W. (2010) *Data Analysis and Graphics Using R*, 3rd Ed. Cambridge University Press, Cambridge.

Thanks for coming today!

For help and advice with your data analysis, contact the StatLab to set up an appointment: statlab@virginia.edu

Sign up for more workshops or see past workshops:
<http://data.library.virginia.edu/statlab/>

Register for the Research Data Services newsletter to stay up-to-date on StatLab events and resources:
<http://data.library.virginia.edu/newsletters/>