The Jackknife and Bootstrap

1 Introduction

One of the most important applications of Jackknife and bootstrapping is to compute the standard error or standard deviation of 'any' statistical estimate. This would translate to estimating the variance of the estimator. In other words, we need to estimate how variability in the data influences the variability of the estimator.

Applying the central limit theorem (CLT), we can, in general, derive asymptotic normality of the estimator, and this allows us to compute the standard error and also the confidence intervals. However, the derivation of CLT may often be very hard. Instead of relying on CLT, bootstrapping attempts to learn the distribution of a parameter approximately and then use the approximate distribution to derive its standard deviation.

Now, how can we learn an approximate distribution? Here is an example.

If $P(\theta)$ does not follow any known parametric distribution family. We then approximately learn the distribution $P(\theta)$ using samples $\theta_1, \ldots, \theta_K$ such that $\theta_i \sim P(\theta)$. To see that this is indeed valid. If θ follows a Beta distribution with parameters 19 and 133 ($\theta \sim \text{Beta}(19, 133)$), we want to learn the distribution of $\log(\theta)$.

```
theta <- rbeta(10000, 19, 133)

#Density of log(\theta) using Jacobian is given below
#densijacobian <- (1/beta(19,133)) * exp(log(theta))^19 * (1-exp(log(theta)))^(133-1)
#Use that to compute density values in following grid

thetagrid <- (1:1000)/1000 #seq(range(theta)[1], range(theta)[2], length.out = 1000)
denjacobian <- (1/beta(19,133)) * exp(log(thetagrid))^19 * (1-exp(log(thetagrid)))^(133-1)

plot(density(log(theta)), col=1, type = 'l') #ploting density using Monte Carlo method,
                                             #just transformed the generated data in the
                                             #first line and computed numeric desity
points(log(thetagrid), denjacobian, col=2, type = 'l') #Jacobian computed densities

#Some standard distributions:

plot(density(theta), col=1, type = 'l') #ploting density using Monte Carlo method
                                        #(computed numeric density)
points(thetagrid, dbeta(thetagrid, 19, 133), col=2, type = 'l')#r function computed density


x <- rnorm(1000, 0, 1)
xgrid <- seq(-3,3,length.out = 1000)
```

```
plot(density(x), col=1, type = 'l') #ploting density using Monte Carlo method
                                    #(computed numeric density)
points(xgrid, dnorm(xgrid, 0, 1), col=2, type = 'l') #r function computed density
```

Above examples show numerically computed densities from samples are matching with exact densities.

Hence, if we have samples of the parameter of interest, we can learn its distribution. We can imagine a dataset as a sample of an unknown data generating mechanism. Ok, that makes a dataset one single sample. But we need multiple samples (or datasets). 'Resampling' then comes in action. We will resample from the same dataset multiple times to create these fake data.

If $x_1, \ldots, x_n \sim F$, some distribution, let $\mu = \mathbb{E}(x)$ be our parameter of interest. An unbiased estimate of μ in most cases is sample mean $\hat{\mu} = \frac{1}{n}\sum_i x_i = \bar{x}$. Now, variance of this estimate $V(\hat{\mu}) = V(\frac{1}{n}\sum_i x_i) = \frac{\sigma^2}{n}$, where $V(x_i) = \sigma^2$. When σ^2 is also unknown, we can plug-in the unbiased sample estimate $\hat{\sigma}^2 = \frac{1}{n-1}\sum_i (x_i - \bar{x})^2$.

For mean, standard error or variance estimation is easy. However, for sample median or mode, this is not so easy. Even when we want a variance of $\hat{\sigma}^2$, it is still complicated. More generally, say, T_n be a sample statistic which is a function of the data $\{x_1, \ldots, x_n\}$. Thus $T_n = g(x_1, \ldots, x_n)$.

2 The Jackknife

2.1 Jackknife definitions: bias and standard error

Suppose $X = (X_1, \ldots, X_n)^\top \sim F$ is a random sample from an unknown distribution F. Let $\theta = T(F)$ be a parameter of interest and let

$$\hat{\theta} = s(x)$$

be an estimator computed from the observed data $x = (x_1, \ldots, x_n)^\top$. The estimator $s(x)$ need not be a plug-in estimator of the form $T(\hat{F})$.

The jackknife studies the sensitivity of $\hat{\theta}$ to individual observations by systematically leaving out one observation at a time.

Leave-one-out estimates. Let $x_{(i)}$ denote the sample obtained by removing the ith observation:

$$x_{(i)} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n).$$

Define the leave-one-out estimator

$$\hat{\theta}_{(i)} = s(x_{(i)}), \qquad i = 1, \ldots, n.$$

The average of the leave-one-out estimates is

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

If $\hat{\theta}$ is stable, then $\hat{\theta}_{(i)}$ should be close to $\hat{\theta}$ for all i. Large deviations indicate sensitivity to individual observations.

Jackknife bias estimate. The jackknife estimate of bias is

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

This formula arises from a first-order Taylor expansion of $\hat{\theta}$ around the empirical distribution. The factor $(n-1)$ rescales the leave-one-out discrepancy to match the scale of the full-sample estimator.

Example: bias of the sample variance. Let

$$\hat{\theta} = s^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2,$$

the biased sample variance. The jackknife bias estimate recovers the familiar finite-sample bias and leads to the unbiased variance estimator

$$\frac{n}{n-1}s^2.$$

Jackknife standard error estimate. The jackknife estimate of the standard error of $\hat{\theta}$ is

$$\widehat{\text{se}}_{\text{jack}}(\hat{\theta}) = \left[\frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\right)^2\right]^{1/2}.$$

This is simply the sample standard deviation of the leave-one-out estimates, with a scaling factor $(n-1)/n$ to account for dependence among the $\hat{\theta}_{(i)}$.

Example: standard error of the sample mean. For the sample mean $\bar{X}$,

$$\hat{\theta}_{(i)} = \frac{1}{n-1}\sum_{j\neq i} X_j.$$

A direct calculation shows that

$$\widehat{\text{se}}_{\text{jack}}(\bar{X}) = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(X_i - \bar{X})^2},$$

which coincides with the classical standard error of the mean.

Thus, for linear statistics, the jackknife is exact.

Example: failure for the sample median. For the sample median, the leave-one-out estimates $\hat{\theta}_{(i)}$ often change only when the deleted observation is near the center of the data. As a result:

- many $\hat{\theta}_{(i)}$ are identical,
- the jackknife variance estimate can be severely biased downward,
- the jackknife standard error may be inconsistent.

This is a well-known failure case motivating the bootstrap.

Why it fails? The delete-1 jackknife relies on a linear (first-order) approximation: $\hat{\theta}_{(i)} \approx \hat{\theta} - \frac{1}{n}\text{IF}(X_i)$, where $\text{IF}(X_i)$ stands for the influence function evaluated at observation x_i. This works well when the estimator is smooth. Unless the estimator is sufficiently smooth as a function of the data, the above approximation will not hold well. Specifically, if IF is discontinuous or zero for most points, the jackknife breaks.

Influence function (IF). Let $T(F)$ be a statistical functional and F the underlying distribution. The influence function (IF) of T at point x is defined as

$$\mathrm{IF}(x; T, F) = \lim_{\varepsilon \downarrow 0} \frac{T\big((1-\varepsilon)F + \varepsilon \delta_x\big) - T(F)}{\varepsilon},$$

where δ_x denotes a point mass at x. The IF quantifies the first-order sensitivity of the estimator to an infinitesimal contamination at x.

Example: Median Let $\theta = \mathrm{median}(F)$ and assume F has density f with $f(\theta) > 0$. The influence function of the median is

$$\mathrm{IF}(x) = \frac{1}{2f(\theta)} \big\{ \mathbf{1}(x \le \theta) - \mathbf{1}(x > \theta) \big\}.$$

This IF is bounded but discontinuous at θ, which explains why linear approximations such as the delete-one jackknife fail for the sample median.

2.2 Derivation via jackknife pseudovalues

A classical way to derive the jackknife variance formula is through *pseudovalues*. This approach avoids an explicit influence-function calculations and provides an intuitive interpretation of the jackknife as a variance estimator based on approximately independent contributions.

Definition of pseudovalues. Let $\hat{\theta} = s(x_1, \ldots, x_n)$ be the full-sample estimator and $\hat{\theta}_{(i)} = s(x_{(i)})$ the leave-one-out estimator with observation i deleted. The ith jackknife pseudovalue is defined by

$$\tilde{\theta}_i := n\hat{\theta} - (n-1)\hat{\theta}_{(i)}, \qquad i = 1, \ldots, n. \tag{PV}$$

Heuristically, $\tilde{\theta}_i$ represents the contribution of the ith observation to the full estimator $\hat{\theta}$. For smooth estimators, the pseudovalues behave approximately like i.i.d. observations with mean θ.

Jackknife estimator as an average of pseudovalues. A key identity is that the average pseudovalue equals the jackknife bias-corrected estimator:

$$\tilde{\theta}_{(\cdot)} := \frac{1}{n} \sum_{i=1}^{n} \tilde{\theta}_i = n\hat{\theta} - (n-1)\hat{\theta}_{(\cdot)},$$

where

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

For many smooth estimators, $\tilde{\theta}_{(\cdot)}$ is approximately unbiased for θ, and

$$\hat{\theta} \approx \frac{1}{n} \sum_{i=1}^{n} \tilde{\theta}_i.$$

Variance estimate from pseudovalues. If the pseudovalues $\tilde{\theta}_1, \ldots, \tilde{\theta}_n$ are treated as an approximately independent sample, then the variance of their average is estimated by

$$\widehat{\mathrm{Var}}(\hat{\theta}) = \frac{1}{n} \cdot \frac{1}{n-1} \sum_{i=1}^{n} \big(\tilde{\theta}_i - \tilde{\theta}_{(\cdot)}\big)^2.$$

This is simply the usual variance-of-the-mean formula applied to the pseudovalues.

Reduction to the standard jackknife formula. Now observe that

$$\tilde{\theta}_i - \tilde{\theta}_{(\cdot)} = \Big[n\hat{\theta} - (n-1)\hat{\theta}_{(i)}\Big] - \Big[n\hat{\theta} - (n-1)\hat{\theta}_{(\cdot)}\Big] = -(n-1)\big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big).$$

Substituting this into (2.2) gives

$$\widehat{\mathrm{Var}}(\hat{\theta}) = \frac{1}{n}\cdot\frac{1}{n-1}\sum_{i=1}^{n}\Big[(n-1)^2\big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2\Big]$$

$$= \frac{n-1}{n}\sum_{i=1}^{n}\big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2.$$

Thus we arrive at the usual delete-1 jackknife variance estimator:

$$\boxed{\widehat{\mathrm{Var}}_{\mathrm{jack}}(\hat{\theta}) = \frac{n-1}{n}\sum_{i=1}^{n}\big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2.}$$

Taking square roots yields the jackknife standard error:

$$\boxed{\widehat{\mathrm{se}}_{\mathrm{jack}}(\hat{\theta}) = \left[\frac{n-1}{n}\sum_{i=1}^{n}\big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2\right]^{1/2}.}$$

The pseudovalue derivation highlights the jackknife's interpretation as a variance estimator based on n approximately independent "data contributions." This works well for smooth estimators, but fails for non-smooth functionals (e.g. sample quantiles), where pseudovalues are not stable.

2.3 Derivation of the jackknife standard error for asymptotic linearity

We now give a standard derivation of the delete-1 jackknife variance formula based on the influence-function (asymptotic linearity) representation.

Asymptotic linearity assumption. Let $\hat{\theta} = T(\hat{F})$ be an estimator of a scalar parameter $\theta = T(F)$. Assume that $\hat{\theta}$ is *asymptotically linear*, meaning that there exists an influence function $\mathrm{IF}(X)$ such that

$$\hat{\theta} - \theta = \frac{1}{n}\sum_{j=1}^{n}\mathrm{IF}(X_j) + o_p(n^{-1/2}), \tag{1}$$

where $\mathbb{E}[\mathrm{IF}(X)] = 0$ and $\mathbb{E}[\mathrm{IF}(X)^2] < \infty$. (You can imagine $o_p(n^{-1/2})$ is something which becomes negligible with increasing sample size.) Then

$$\mathrm{Var}(\hat{\theta}) = \frac{1}{n}\mathrm{Var}(\mathrm{IF}(X)) + o(n^{-1}).$$

Leave-one-out expansion. Let $\hat{\theta}_{(i)}$ denote the estimator computed with observation i deleted. Since $\hat{\theta}_{(i)}$ is based on $n-1$ points, the same expansion gives

$$\hat{\theta}_{(i)} - \theta = \frac{1}{n-1}\sum_{j\neq i}\mathrm{IF}(X_j) + o_p(n^{-1/2}).$$

Subtracting (1) yields

$$\hat{\theta}_{(i)} - \hat{\theta} = \frac{1}{n-1}\sum_{j\neq i} \mathrm{IF}(X_j) - \frac{1}{n}\sum_{j=1}^{n} \mathrm{IF}(X_j) + o_p(n^{-1/2}).$$

Let $S = \sum_{j=1}^{n} \mathrm{IF}(X_j)$. Then $\sum_{j\neq i} \mathrm{IF}(X_j) = S - \mathrm{IF}(X_i)$, so

$$\hat{\theta}_{(i)} - \hat{\theta} = \frac{S - \mathrm{IF}(X_i)}{n-1} - \frac{S}{n} + o_p(n^{-1/2}) = \frac{S}{n(n-1)} - \frac{\mathrm{IF}(X_i)}{n-1} + o_p(n^{-1/2}).$$

Centering at the jackknife mean. Define the jackknife mean

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

Averaging the above expansion over i gives

$$\hat{\theta}_{(\cdot)} - \hat{\theta} = o_p(n^{-1/2}),$$

so to first order $\hat{\theta}_{(\cdot)}$ and $\hat{\theta}$ coincide.

Therefore,

$$\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)} = -\frac{\mathrm{IF}(X_i)}{n-1} + o_p(n^{-1/2}). \tag{2}$$

Jackknife variance. Squaring (2) and summing over i gives

$$\sum_{i=1}^{n} \big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2 = \frac{1}{(n-1)^2}\sum_{i=1}^{n} \mathrm{IF}(X_i)^2 + o_p(1).$$

Multiplying by the scaling factor $(n-1)/n$ yields

$$\frac{n-1}{n}\sum_{i=1}^{n} \big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2 = \frac{1}{n(n-1)}\sum_{i=1}^{n} \mathrm{IF}(X_i)^2 + o_p(n^{-1}).$$

Since $\frac{1}{n}\sum_{i=1}^{n} \mathrm{IF}(X_i)^2$ consistently estimates $\mathbb{E}[\mathrm{IF}(X)^2]$, we conclude that

$$\widehat{\mathrm{Var}}_{\mathrm{jack}}(\hat{\theta}) := \frac{n-1}{n}\sum_{i=1}^{n} \big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2 \quad \xrightarrow{p} \quad \mathrm{Var}(\hat{\theta}).$$

Thus, the jackknife standard error is

$$\boxed{\widehat{\mathrm{se}}_{\mathrm{jack}}(\hat{\theta}) = \left[\frac{n-1}{n}\sum_{i=1}^{n} \big(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}\big)^2\right]^{1/2}.}$$

The key step is the first-order approximation $\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)} \approx -(n-1)^{-1}\mathrm{IF}(X_i)$. Hence, the jackknife is valid precisely when the estimator admits an influence function expansion, i.e. when it is sufficiently smooth.

2.4 Numerical examples

Jackknife-based confidence intervals. Let $\hat{\theta} = T(\hat{F})$ be an estimator of a scalar parameter θ based on data $X_1, \ldots, X_n$, and let $\hat{\theta}_{(i)}$ denote the leave-one-out estimator computed with the ith observation removed. Define the jackknife mean

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

The jackknife variance estimator is

$$\widehat{\mathrm{Var}}_{\mathrm{jack}}(\hat{\theta}) = \frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}_{(i)} - \bar{\theta}_{(\cdot)}\right)^2,$$

with standard error $\widehat{\mathrm{SE}}_{\mathrm{jack}}(\hat{\theta}) = \sqrt{\widehat{\mathrm{Var}}_{\mathrm{jack}}(\hat{\theta})}$.

If $\hat{\theta}$ admits a first-order linear expansion

$$\hat{\theta} - \theta = \frac{1}{n} \sum_{i=1}^{n} \mathrm{IF}(X_i) + o_p(n^{-1/2}),$$

then $\widehat{\mathrm{SE}}_{\mathrm{jack}}(\hat{\theta})$ is consistent and $\hat{\theta}$ is asymptotically normal. A $(1-\alpha)$ jackknife confidence interval is therefore given by

$$\boxed{\hat{\theta} \ \pm \ z_{1-\alpha/2}\, \widehat{\mathrm{SE}}_{\mathrm{jack}}(\hat{\theta})}.$$

This normal-based jackknife confidence interval is valid for smooth estimators (e.g., means, smooth M-estimators, and U-statistics), but fails for non-smooth estimators such as sample quantiles.

With Bootstrap, we can do even better.

Example 1: Jackknife standard error of the mean (`iris`). Consider the famous `iris` dataset. Let

$$\hat{\theta} = \bar{x}, \qquad x_i = \texttt{Sepal.Length}_i, \qquad n = 150.$$

The leave-one-out means are

$$\hat{\theta}_{(i)} = \frac{1}{n-1} \sum_{j \neq i} x_j, \qquad i = 1, \ldots, n.$$

The jackknife standard error is

$$\widehat{\mathrm{se}}_{\mathrm{jack}}(\bar{x}) = \left[\frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)} \right)^2 \right]^{1/2}.$$

R code:

```
data(iris)
x <- iris$Sepal.Length
n <- length(x)

theta_hat <- mean(x)
```

```
theta_i   <- sapply(1:n, function(i) mean(x[-i]))

theta_dot <- mean(theta_i)

se_jack <- sqrt((n-1)/n * sum((theta_i - theta_dot)^2))
theta_hat
se_jack
```

For the sample mean, the jackknife matches the classical standard error exactly.

Example 2: Jackknife standard error of a regression coefficient (`mtcars`). Let us estimate the slope in the regression model

$$\texttt{mpg} = \beta_0 + \beta_1\, \texttt{wt} + \varepsilon$$

using the dataset `mtcars` ($n = 32$).

The estimator of interest is

$$\hat{\theta} = \hat{\beta}_1.$$

The jackknife recomputes the slope after deleting each observation.

R code:

```
data(mtcars)

y <- mtcars$mpg
x <- mtcars$wt
n <- length(y)

# full-sample slope
fit_full <- lm(y ~ x)
theta_hat <- coef(fit_full)[2]

# leave-one-out slopes
theta_i <- sapply(1:n, function(i) {
  coef(lm(y[-i] ~ x[-i]))[2]
})

theta_dot <- mean(theta_i)

se_jack <- sqrt((n-1)/n * sum((theta_i - theta_dot)^2))

theta_hat
se_jack
```

Because regression coefficients are smooth functionals, the jackknife performs very well here.

Example 3: Failure for the sample median (`faithful`). Now consider the eruption times in the dataset `faithful`. Let

$$\hat{\theta} = \text{median}(x).$$

Unlike the mean, the median is not smooth, and many leave-one-out medians are identical.

R code:

```
data(faithful)
x <- faithful$eruptions
n <- length(x)

theta_hat <- median(x)

theta_i <- sapply(1:n, function(i) median(x[-i]))
theta_dot <- mean(theta_i)

se_jack <- sqrt((n-1)/n * sum((theta_i - theta_dot)^2))

theta_hat
se_jack
table(theta_i)   # many repeated values
```

In this case, the jackknife standard error is often too small, reflecting the well-known inconsistency of the delete-1 jackknife for non-smooth estimators such as quantiles.

3 The Bootstrap

Bootstrapping is any test or metric that uses random sampling with replacement (e.g. mimicking the sampling process), and falls under the broader class of resampling methods (like jackknife, Permutation tests, cross validation, etc.). The bootstrap was published by Bradley Efron in "Bootstrap methods: another look at the jackknife" (1979), inspired by earlier work on the jackknife. Subsequently, he wrote several papers on this, exploring many directions and improving its performance. The term 'bootstrap' is due to the idiom 'pull oneself up by one's bootstraps'.

Monte Carlo method: We can draw multiple samples from the same data generating mechanism (this is also called replication!). For each sample, we can compute the T_n and thus we get a sample of T_n's.

However, for real data, we only have one sample. Thus, Monte Carlo method will not work. Although, we can empirically estimate the distribution F_n of F using the data $x_1, \ldots, x_n$ and generate multiple replications of the data.

By definition, $F(x) = P(X \leq x)$ is a cumulative distribution function. We can estimate F with the empirical distribution function F_n, the cdf that puts mass $1/n$ at each data point x_i.

$$F_n(x) = \frac{\text{Number of samples in the data} \leq x}{n}.$$

The above definition of empirical distribution is itself complicated when we are in the multivariate regime, as there is no clear definition of the cumulative distribution function. Hence, resampling-based methods are easier to use.

Parametric bootstrap: If we know the parametric class of F, we can consider the parametric bootstrap. Let us assume that we know $x_1, \ldots, x_n$ follows a Poisson distribution. We first estimate the model parameters and then generate samples from the distribution with the estimated parameter.

```
lambdahat = mean(x)
for b = 1,...,B
sample xb_1,...,xb_n from Poisson(lambdahat).
Compute Tb
```

```
end;
barT=mean(Tb)
VarT=V(Tb) #this is sample variance
```

Non-parametric bootstrap: what if the distribution is not known at all? Hence, we need to rely on empirically estimated distribution. Then, how to sample from the empirical distribution? Drawing $\{x_1^*, \ldots, x_n^*\}$ from the empirical distribution F_n is equivalent to draw n observations, with replacement from the original data $\{x_1, \ldots, x_n\}$. Therefore, Bootstrapping sampling is also described as resampling data.

```
for b = 1,...,B
sample xb_1,...,xb_n from {x_1,...,x_n} with replacement.
Compute Tb
end;
barT=mean(Tb)
VarT=V(Tb) #this is sample variance
```

Here Tb could be sample median or mode of the sampled data xb_1,...,xb_n. V(Tb)=$\frac{1}{B-1}\sum_b (Tb - \text{mean}(Tb))^2$.

Bootstrap confidence interval

Bootstrapping can further help to construct confidence intervals. There are two ways to do that using the bootstrap samples. Let $T_1, \ldots, T_B$ be the estimates collected from B many resamples.

Normal approximation: In this approach, we assume normality of the generated samples $\boldsymbol{T} = T_1, \ldots, T_B$. And thus the interval will be $(mean(\boldsymbol{T}) - Z_{1-\alpha/2} sd(\boldsymbol{T}), mean(\boldsymbol{T}) + Z_{\alpha/2} sd(\boldsymbol{T}))$ for $100(1-\alpha)\%$ Confidence interval.

Empirical quantile: We can apply the quantile function on $\boldsymbol{T}$ to compute the confidence interval empirically. Specifically, it will look like $quantile(\boldsymbol{T}, probs = c(0.025, 0.975))$.

Connection to $\log(\theta)$ example: We need to generate samples of T_n which is a function of the data $g(x_1, \ldots, x_n)$ (which is like $\log(\theta)$). There, we generated samples of θ. Here we need to get samples of the data i.e. $\{x_1^{(k)}, \ldots, x_n^{(k)}\}_{k=1}^K$. And our samples of $T_n = \{T_n^{(1)}, \ldots, T_n^{(K)}\}$ are $T_n^{(k)} = g(x_1^{(k)}, \ldots, x_n^{(k)})$.

Why bootstrapping works?: A numerical illustration that resampling preserves the original distribution in 'Expectation': Here we use the statistical concept that the cumulative distribution function $F(m) = P(X \leq m)$ 'uniquely' characterizes a distribution. Since we check this numerically, we consider the empirical cumulative distribution function (ecdf) for testing. First, the code:

```
n <- 100
x <- rnorm(n)

###Number of bootstrap samples is N
N <- 1000
```

```
####Store the bootstrap samples
y <- matrix(0, N, n)
for(i in 1:N){
  y[i,] <- x[sample(1:n, replace = T)]
}

m <- 0.7

##True empirical cumulative distribution
mean(x <= m)

##Average empirical cumulative distributions across the resamples
avgecdf <- apply(y, 1, FUN=function(z){mean(z <= m)})
mean(avgecdf)

sd(avgecdf)

###Empirical confidence interval
quantile(avgecdf, probs = c(0.025,0.975))



####################For standard deviation as estimator############
sd(x)

##Average empirical cumulative distributions across the resamples
avgsd <- apply(y, 1, sd)
mean(avgsd)

sd(avgsd)

###Empirical confidence interval
quantile(avgsd, probs = c(0.025,0.975))
```

We can vary N and n to see their influences. We see that the average ecdf from resamples match with the true ecdf. Note that `mean(x <= m)` or mathematically, $F_n(m) = \frac{\text{Number of samples in the data} \leq m}{n}$ is an estimator of $F(X \leq m)$. Hence, by the bootstrap theory, `sd(avgecdf)` quantifies the variance of the estimator $F_n(m)$. If we have sufficiently large N, in general, the mean and sd of the estimator are primarily influenced by n only. However, if one wants to compute the confidence intervals empirically, it is important to set N large enough to be able to approximate the two tail probabilities sufficiently well.

3.1 Bootstrap overview

The bootstrap is a computer-based resampling procedure to assess statistical accuracy:

- It can compute standard error or bias of a statistic, approximate the sampling distribution of a statistic, or construct confidence intervals.

- It does not require an explicit mathematical expression for bias or standard error.

3.2 Bootstrap setup and notation

Let $X = (X_1, \ldots, X_n)^T \sim F$.

- $\theta = T(F)$: parameter of interest.
- $\hat{\theta} = s(x)$: estimator based on observed sample $x = (x_1, \ldots, x_n)^T$.
- $\widehat{F}$: empirical distribution function (EDF) of x:

$$\widehat{F}(t) = \frac{1}{n} \sum_{i=1}^{n} \overset{\text{ind}}{\sim} (x_i \leq t).$$

- A bootstrap sample $X^* = (X_1^*, \ldots, X_n^*)$ is a random sample of size n drawn *with replacement* from $\widehat{F}$.
- A bootstrap replication of $\hat{\theta}$ is

$$\hat{\theta}^* = s(x^*).$$

- Repeat R times to obtain $\hat{\theta}^{*1}, \ldots, \hat{\theta}^{*R}$.

3.3 Bootstrap standard error

Draw R bootstrap samples $X^{*1}, \ldots, X^{*R}$ and compute $\hat{\theta}^{*r} = s(x^{*r})$. Then the bootstrap estimate of standard error is

$$\widehat{\text{se}}_{\text{boot}}(\hat{\theta}) = \left[\frac{\sum_{r=1}^{R} (\hat{\theta}^{*r} - \hat{\theta}^{*(\cdot)})^2}{R-1} \right]^{1/2}, \qquad \hat{\theta}^{*(\cdot)} = \frac{1}{R} \sum_{r=1}^{R} \hat{\theta}^{*r}.$$

Rule of thumb stated: $R \approx 50$ can be enough for a reasonable estimate of $\text{se}(\hat{\theta})$, but *much larger* R is required for confidence intervals.

3.4 Bootstrap percentile confidence interval

Order the bootstrap replicates

$$\hat{\theta}^*_{(1)} \leq \hat{\theta}^*_{(2)} \leq \cdots \leq \hat{\theta}^*_{(R)}.$$

Let

$$m = \lfloor (\alpha/2)R \rfloor \quad (\text{or } \lfloor (\alpha/2)R \rfloor \text{ as stated}),$$

and interpret the $(1-\alpha)100\%$ bootstrap percentile CI for θ as

$$\left(\hat{\theta}^*_{(m)},\ \hat{\theta}^*_{(R-m)} \right).$$

Guidance stated: choose $R = 1000$ or larger.

4 Bootstrap Hypothesis Testing: One-sample Location Problem

Consider the one-sample location problem:

- $X = (X_1, \ldots, X_n)^T \sim F$, observed sample $x = (x_1, \ldots, x_n)^T$.
- Hypotheses: $H_0 : \theta = \theta_0$ vs $H_a : \theta \neq \theta_0$.
- Let $T(X)$ be a test statistic (need not be an estimator). Le $T(x) = \hat{\theta} = \bar{x}$ be an example.

To generate bootstrap samples *under* H_0, recenter:

$$\{x_1 - \hat{\theta} + \theta_0, \ \ldots, \ x_n - \hat{\theta} + \theta_0\},$$

and resample from this set. Let $\hat{\theta}^{*r}$ be the statistic computed on the rth bootstrap sample. A one-sided p-value is written as

$$\text{P-value} = \frac{\#\{\hat{\theta}^{*r} \geq \hat{\theta}\}}{R},$$

with the appropriate modification for two-sided tests.

5 Assessing Error in Bootstrap Estimates

Bootstrap estimates are not exact (nearly unbiased but can have substantial variance). Two sources of variability are highlighted:

5.1 Sampling variability

We have only a sample of size n rather than the entire population.

5.2 Resampling variability

We take only R bootstrap samples rather than the total number of distinct bootstrap samples.

6 Example: Aspirin and Heart Attacks (Bootstrap)

6.1 Hypothesis and experiment

Hypothesis: small aspirin doses prevent heart attacks in healthy middle-aged men.

Controlled, randomized, double-blinded study:

- One half received aspirin, other half placebo.
- Define $X_i = 1$ if a heart attack is observed and $X_i = 0$ otherwise.

Data summary:

```
labels = c("nattacks","nsubjects")
aspirin = c(104, 11037)
placebo = c(189, 11034)
data = data.frame(aspirin, placebo)
rownames(data) = labels
data
```

6.2 Estimator: ratio of rates

Define the attack rate within a group as $\text{rate} = \frac{\text{nattacks}}{\text{nsubjects}}$. Estimate

$$\hat{\theta} = \frac{\text{rate}_{\text{aspirin}}}{\text{rate}_{\text{placebo}}}.$$

R code shown:

```
ratio = function(r) { r[1] / r[2] }
theta.hat = ratio(data$aspirin) / ratio(data$placebo)
theta.hat
```

Interpretation: in the sample, aspirin-takers have about 55% as many heart attacks as placebo-takers.

6.3 Bootstrap to assess uncertainty

Construct binary samples reflecting each group:

```
sample.aspirin = c(rep(1, times = data["nattacks","aspirin"]),
                   rep(0, times = (data["nsubjects","aspirin"] - data[
                      "nattacks","aspirin"])))
table(sample.aspirin)

sample.placebo = c(rep(1, times = data["nattacks","placebo"]),
                   rep(0, times = (data["nsubjects","placebo"] - data[
                      "nattacks","placebo"])))
table(sample.placebo)
```

Define a bootstrap replicate of the ratio of heart-attack rates:

```
bootstrap.sample = function() {
  boot.sam.aspirin = sample(sample.aspirin, replace = TRUE)
  boot.sam.placebo = sample(sample.placebo, replace = TRUE)
  h.rate.aspirin = sum(boot.sam.aspirin)/length(boot.sam.aspirin)
  h.rate.placebo = sum(boot.sam.placebo)/length(boot.sam.placebo)
  return(h.rate.aspirin/h.rate.placebo)
}
```

Run bootstrap:

```
R = 10000
theta.boot = replicate(R, bootstrap.sample())
hist(theta.boot, breaks=100)
abline(v=theta.hat, col="red", lwd=4)

theta.lower = sort(theta.boot)[R*.025]
theta.upper = sort(theta.boot)[R*.975]
abline(v=c(theta.lower, theta.upper), lwd=3)
```

Percentile CI via quantiles:

```
quantile(theta.boot, probs = c(.025, .975))
```

Conclusion stated: aspirin is significantly beneficial (CI below 1).

7 Complete Enumeration and the Exhaustive Bootstrap

7.1 Counting bootstrap samples

Number of bootstrap samples (ordered with replacement) of size n from n observed points is n^n, but many such samples correspond to the same multiset.

Characterize a bootstrap sample by a weight vector

$$k = (k_1, \dots, k_n), \qquad \sum_{i=1}^{n} k_i = n, \quad k_i \in \{0, 1, 2, \dots\}.$$

Let

$$C_n = \{k = (k_1, \dots, k_n) : \ k_1 + \cdots + k_n = n, \ k_i \geq 0, \ k_i \in \mathbb{Z}\}$$

be the space of compositions of n into at most n parts.

7.2 Size of the composition space

The size is

$$|C_n| = \binom{2n-1}{n-1},$$

with the standard "stars and bars" interpretation: distribute n balls into n boxes using $n-1$ separators among $2n-1$ positions.

Each bootstrap sample corresponds to sampling weights

$$k \sim \text{Multinomial}(n, p), \qquad p = (p_1, \dots, p_n), \qquad p_i = \frac{1}{n}.$$

7.3 Exhaustive bootstrap distribution

The exhaustive bootstrap distribution of a statistic $T(X)$:

- compute each of the $\binom{2n-1}{n-1}$ statistics (one per composition in C_n),
- associate each with a weight $k \sim \text{Multinomial}(n, p)$.

This shift from resamples to C_n can yield large computational savings; for example, for $n = 10$, enumerations reduce from 10^{10} to 92378.

8 Monte Carlo Bootstrap vs Exhaustive Bootstrap: LSAT–GPA Example

Data: LSAT scores and GPA (classic Efron bootstrap illustration).

```
library(bootstrap)
data(law)
t(law)
```

Plug-in estimate of correlation:

```
theta.hat = cor(law$LSAT, law$GPA)
theta.hat
```

Monte Carlo bootstrap replicates for the correlation:

```
draw.bootstrap.samples = function(df) {
  n = dim(df)[1]
  ind = sample(n, replace = TRUE)
  cor.bootstrap.replicate = cor(df[ind, "LSAT"], df[ind, "GPA"])
  return(cor.bootstrap.replicate)
}
R = 10000
theta.hat.star = replicate(R, draw.bootstrap.samples(law))
```

Bootstrap SE:

```
sd(theta.hat.star)
```

Exhaustive bootstrap via compositions:

```
library(partitions)
n = 15
allCompositions = compositions(n, n)
dim(allCompositions)[2] == choose((2*n-1), (n-1))
```

9 Gray Codes to Speed Up Enumeration

Enumeration can be sped up by changing only one coordinate at a time using Gray codes. Suggested reading mentioned: Diaconis and Holmes (1994), *Gray Codes for Randomization Procedures*.

10 Why Bootstrap Works (Functional/Variance View)

A functional viewpoint for the consistency of the bootstrap variance estimator.

Let $\hat{\theta}_n = T_{\text{target}}(\widehat{F}_n)$ be a statistic (functional of the EDF). Its distribution is determined by F and n, hence

$$\text{Var}(\hat{\theta}_n) = \text{Var}\big(T_{\text{target}}(\widehat{F}_n)\big) = v_{n,\text{target}}(F).$$

Often we have a scaling

$$v_{n,\text{target}}(F) \approx \frac{1}{n}\, v_{1,\text{target}}(F),$$

for some functional $v_{1,\text{target}}$.

Conditionally on $\widehat{F}_n$, the bootstrap estimator is $\hat{\theta}^*_n = T_{\text{target}}(\widehat{F}^*_n)$ and

$$\text{Var}(\hat{\theta}^*_n \mid \widehat{F}_n) = \text{Var}\big(T_{\text{target}}(\widehat{F}^*_n) \mid \widehat{F}_n\big) = v_{n,\text{target}}(\widehat{F}_n) \approx \frac{1}{n}\, v_{1,\text{target}}(\widehat{F}_n).$$

When $v_{1,\text{target}}(\widehat{F}_n) \approx v_{1,\text{target}}(F)$, bootstrap variance estimates the true sampling variance well.

10.1 Worked example: the mean

For the mean functional $T_{\text{mean}}(F) = \int z\, dF(z)$, we have:

$$\hat{\mu}_n = T_{\text{mean}}(\widehat{F}_n) = \int z\, d\widehat{F}_n(z) = \bar{X}_n,$$

and the bootstrap mean

$$\hat{\mu}_n^* = T_{\text{mean}}(\widehat{F}_n^*) = \int z \, d\widehat{F}_n^*(z) = \bar{X}_n^*.$$

In this case,

$$\text{Var}(T_{\text{mean}}(\widehat{F}_n)) = \text{Var}(\bar{X}_n) = \frac{1}{n}\text{Var}_F(X),$$

and the bootstrap conditional variance is

$$\text{Var}(T_{\text{mean}}(\widehat{F}_n^*) \mid \widehat{F}_n) = \frac{1}{n}\text{Var}_{\widehat{F}_n}(X).$$

11 Example When Bootstrap Fails

A nonsmooth functional at a boundary point.

Let $X_1, X_2, \ldots, X_n \overset{\text{iid}}{\sim} F$ with $\sigma^2 = \text{Var}_F(X) = 1$ and consider $g(x) = |x|$. Define

$$T_n = \sqrt{n}\,\big(g(\bar{X}) - g(\mu)\big).$$

If the true mean is $\mu = 0$, then by CLT and continuous mapping,

$$T_n = \sqrt{n}\,|\bar{X}| \xrightarrow{d} |Z|, \qquad Z \sim N(0, \sigma^2).$$

However, the bootstrap counterpart

$$T_n^* = \sqrt{n}\,\big(|\bar{X}^*| - |\bar{X}|\big)$$

does not converge to the same limit distribution when $\mu = 0$; rather it converges to a different distribution (as shown in the slide derivation), so the sequence of bootstrap CDFs is not consistent at $\mu = 0$.

12 When Bootstrap Will Work

Statement emphasized: bootstrap will work if the estimator is *asymptotically linear*. That is, if

$$\hat{\theta}_n - \theta = \frac{1}{n}\sum_{i=1}^{n} \psi(X_i) + o_p(n^{-1/2})$$

for some influence function / score-like ψ, then the bootstrap typically gives a consistent approximation to the sampling distribution.