MS&E 226: Fundamentals of Data Science

Lecture 10: The bootstrap

Ramesh Johari

Recall: Limitations of asymptotic theory

In Lecture 9, we used asymptotic normality to construct standard errors and confidence intervals for estimators like the sample mean, OLS coefficients, and logistic regression coefficients.

But this approach has limitations:

- Asymptotic theory requires “large” sample sizes; with small n, the normal approximation may be poor.
- Assumptions like homoskedasticity (constant error variance) may be violated.
- For complex estimators, closed-form expressions for standard errors may not exist.

Question: Is there a more flexible, assumption-light approach to quantifying uncertainty?

The bootstrap: A simulation-based approach

The *bootstrap* provides a practical alternative to asymptotic theory.

Key idea: Instead of relying on theoretical approximations, we use the data itself to simulate the sampling distribution.

This lecture introduces the bootstrap as a flexible tool for computing standard errors and confidence intervals in practice.

Resampling and the bootstrap

Idea behind the bootstrap

The basic idea behind the bootstrap is straightforward:

- The data $\mathbf{Y}$ are a sample from the population model.

Idea behind the bootstrap

The basic idea behind the bootstrap is straightforward:

- The data $\mathbf{Y}$ are a sample from the population model.
- If we *resample* (with replacement) from $\mathbf{Y}$, this “mimics” sampling from the population model.

Idea behind the bootstrap

The basic idea behind the bootstrap is straightforward:

- The data $\mathbf{Y}$ are a sample from the population model.
- If we *resample* (with replacement) from $\mathbf{Y}$, this “mimics” sampling from the population model.

In the bootstrap, we draw B new samples (of size n) from the original data, and treat each of these as a “parallel universe.”

We can then pretend this is the sampling distribution, and compute what we want (e.g., standard errors, confidence intervals, etc.).

Why bootstrap?

We've already seen asymptotic normality can give us standard errors and confidence intervals.

Why do we need the bootstrap?

Why bootstrap?

We've already seen asymptotic normality can give us standard errors and confidence intervals.

Why do we need the bootstrap?

- *Small samples*: When n is small, asymptotic approximations may be inaccurate.

Why bootstrap?

We’ve already seen asymptotic normality can give us standard errors and confidence intervals.

Why do we need the bootstrap?

- *Small samples*: When n is small, asymptotic approximations may be inaccurate.
- *Violated assumptions*: Model assumptions (e.g., homoskedasticity, normality) may not hold.

Why bootstrap?

We've already seen asymptotic normality can give us standard errors and confidence intervals.

Why do we need the bootstrap?

- *Small samples*: When n is small, asymptotic approximations may be inaccurate.
- *Violated assumptions*: Model assumptions (e.g., homoskedasticity, normality) may not hold.
- *Complex estimators*: For estimators beyond simple means or regression coefficients (e.g., medians, quantiles, ratios), closed-form standard errors may not exist.

Why bootstrap?

We've already seen asymptotic normality can give us standard errors and confidence intervals.

Why do we need the bootstrap?

- *Small samples*: When n is small, asymptotic approximations may be inaccurate.
- *Violated assumptions*: Model assumptions (e.g., homoskedasticity, normality) may not hold.
- *Complex estimators*: For estimators beyond simple means or regression coefficients (e.g., medians, quantiles, ratios), closed-form standard errors may not exist.

The bootstrap provides a practical, general-purpose tool that requires minimal assumptions.

The bootstrap algorithm

The bootstrap algorithm

We are given a sample $\mathbf{Y}$ of n observations.

We want to estimate the *sampling distribution* of a *statistic* $T(\mathbf{Y})$, i.e., a quantity that depends on the data.

For $1 \leq b \leq B$:

- Draw n samples uniformly at random, with replacement, from $\mathbf{Y}$. Denote by $\mathbf{Y}^{(b)}$ the samples in the b'th "parallel universe."
- Compute value of T in the b'th "parallel universe"; call this T_b.

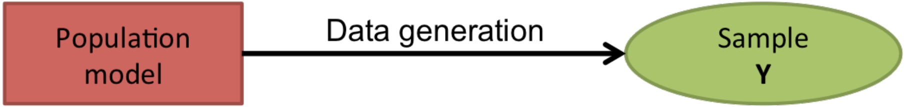
The histogram (i.e., empirical distribution) of $\{T_b, 1 \leq b \leq B\}$ is an estimate of the sampling distribution of T. We call this the *bootstrap distribution*.

The bootstrap algorithm

A picture:

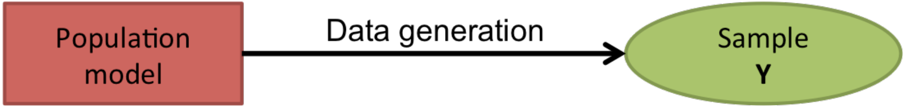
An analogy

The following analogy is helpful to keep in mind. For the sampling distribution we have:

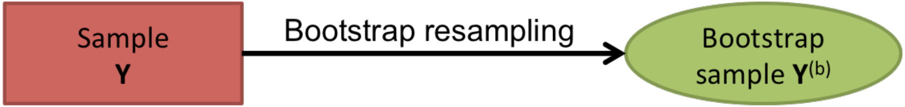


An analogy

The following analogy is helpful to keep in mind. For the sampling distribution we have:



Bootstrapping treats the sample $\mathbf{Y}$ as if it represents the true population model:



Limitations and assumptions

For the bootstrap distribution to accurately approximate the sampling distribution:

Limitations and assumptions

For the bootstrap distribution to accurately approximate the sampling distribution:

- *Representative sample*: The original sample $\mathbf{Y}$ should be i.i.d. draws from the population, with no sampling bias, and n large enough to capture the population's structure.

Limitations and assumptions

For the bootstrap distribution to accurately approximate the sampling distribution:

- *Representative sample*: The original sample $\mathbf{Y}$ should be i.i.d. draws from the population, with no sampling bias, and n large enough to capture the population's structure.
- *Sufficient bootstrap samples*: B should be large enough that resampling variability is negligible (typically $B \geq 10{,}000$).

Limitations and assumptions

For the bootstrap distribution to accurately approximate the sampling distribution:

- *Representative sample*: The original sample $\mathbf{Y}$ should be i.i.d. draws from the population, with no sampling bias, and n large enough to capture the population's structure.
- *Sufficient bootstrap samples*: B should be large enough that resampling variability is negligible (typically $B \geq 10{,}000$).
- *Appropriate statistic*: The bootstrap can fail for extreme value statistics (e.g., maximum, minimum) or when the statistic is highly sensitive to outliers.

Limitations and assumptions

For the bootstrap distribution to accurately approximate the sampling distribution:

- *Representative sample*: The original sample $\mathbf{Y}$ should be i.i.d. draws from the population, with no sampling bias, and n large enough to capture the population's structure.
- *Sufficient bootstrap samples*: B should be large enough that resampling variability is negligible (typically $B \geq 10{,}000$).
- *Appropriate statistic*: The bootstrap can fail for extreme value statistics (e.g., maximum, minimum) or when the statistic is highly sensitive to outliers.

Despite these limitations, for standard tasks like estimating standard errors of means, medians, or regression coefficients, the bootstrap is reliable and widely used.

Standard errors from the bootstrap

We use the bootstrap distribution just like we would use the true sampling distribution.

The bootstrap estimate of the standard error is:

$$\widehat{\mathrm{SE}}_{\mathrm{BS}} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(T_b - \bar{T})^2},$$

where $\bar{T} = (1/B)\sum_{b=1}^{B} T_b$ is the mean of the bootstrap distribution.

In other words, $\widehat{\mathrm{SE}}_{\mathrm{BS}}$ is simply the sample standard deviation of the bootstrap distribution.

Confidence intervals

Two approaches to building 95% confidence intervals from the bootstrap:

Confidence intervals

Two approaches to building 95% confidence intervals from the bootstrap:

- *The normal interval*: $[T(\mathbf{Y}) - 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}, T(\mathbf{Y}) + 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}]$. This approach assumes that the sampling distribution of $T(\mathbf{Y})$ is normal, and uses the standard error accordingly.

Confidence intervals

Two approaches to building 95% confidence intervals from the bootstrap:

- *The normal interval*: $[T(\mathbf{Y}) - 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}, T(\mathbf{Y}) + 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}]$. This approach assumes that the sampling distribution of $T(\mathbf{Y})$ is normal, and uses the standard error accordingly.
- *The percentile interval*: Let T_q be the q'th quantile of the bootstrap distribution. Then the 95th percentile bootstrap interval is: $[T_{0.025}, T_{0.975}]$.

  In general the percentile interval is preferred when the sampling distribution is symmetric, but not necessarily normal. (Many other types of intervals as well...)

Example 1: Mean of flight arrival delays

I sampled 500 flights from all flights in 2024 that had an arrival delay recorded with the Bureau of Transportation Statistics.

The mean delay in this sample was $\hat{\mu} = \overline{Y} = 11.68$ minutes, and the sample standard deviation was $\hat{\sigma} = 74.45$.

If we use the central limit theorem (see last lecture), the resulting sample mean standard error is approximately $\hat{\sigma}/\sqrt{n} = 3.329$.

What does the bootstrap suggest?

Example 1: Mean of flight arrival delays

Code to run bootstrap in R (using `boot` library):

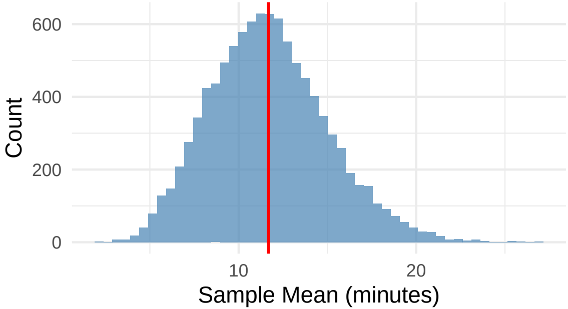
```
# Using the boot package
mean_boot <- function(data, indices) {
  return(mean(data[indices]))
}

# Run bootstrap with B = 10000
boot_mean_result <- boot(flight_sample, mean_boot, R = 10000)
```

`boot_mean_result$t` contains the means across parallel universes. Use `boot.ci` for confidence intervals.

Example 1: Mean of flight arrival delays

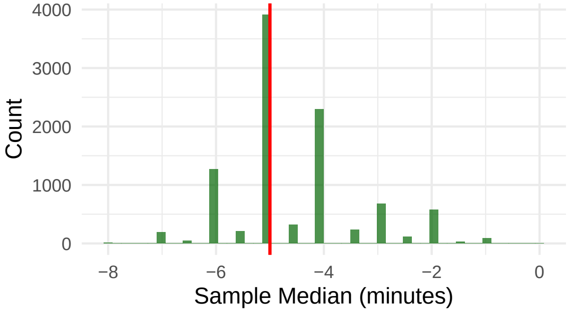
Results:



$\widehat{\mathrm{SE}}_{\mathrm{BS}} = 3.315$, very close to the normal approximation (because the central limit theorem is pretty good here).

Example 2: Median of flight arrival delays

For the median, we don't have a closed form expression for the standard error, but we can use the bootstrap. Results:



$\widehat{\text{SE}}_{\text{BS}} = 1.163$. This is much smaller than the sample mean estimated SE (why?)

Note also that the actual sample median was -5 (matching the population median).

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit “Purchase”.

In practice, customers might *abandon* before completing these activities.

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit “Purchase”.

In practice, customers might *abandon* before completing these activities.

Suppose you’ve collected data on n customers, where $Y_i \in \{1, 2, 3\}$ denotes the latest stage the customer completed.

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit "Purchase".

In practice, customers might *abandon* before completing these activities.

Suppose you've collected data on n customers, where $Y_i \in \{1, 2, 3\}$ denotes the latest stage the customer completed.

Let γ_i be the probability that a customer that completes stage s will also complete stage $s + 1$, for $s = 1, 2$. We estimate these as follows:

$$\hat{\gamma}_1 = \frac{|\{i : Y_i \geq 2\}|}{|\{i : Y_i \geq 1\}|}; \; \hat{\gamma}_2 = \frac{|\{i : Y_i = 3\}|}{|\{i : Y_i \geq 2\}|}.$$

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit "Purchase".

In practice, customers might *abandon* before completing these activities.

Suppose you've collected data on n customers, where $Y_i \in \{1, 2, 3\}$ denotes the latest stage the customer completed.

Let γ_i be the probability that a customer that completes stage s will also complete stage $s + 1$, for $s = 1, 2$. We estimate these as follows:

$$\hat{\gamma}_1 = \frac{|\{i : Y_i \geq 2\}|}{|\{i : Y_i \geq 1\}|}; \ \hat{\gamma}_2 = \frac{|\{i : Y_i = 3\}|}{|\{i : Y_i \geq 2\}|}.$$

But standard errors are not easy to compute, since these are quotients; the bootstrap is an easy approach to get standard errors.

Example 4: Block bootstrap

This example came up when working on a project with Netflix; it is a common example of computing standard errors correctly when you have *clustered observations*.

It is also a useful example to see how you can sometimes apply the bootstrap even in settings where the data is not perfectly independent and identically distributed.

Example 4: Block bootstrap

Suppose we collect data on viewing behavior of $n = 1000$ users. Each user i has k_i *sessions*; and the average *bit rate* of session j of user i is r_{ij}.

So our data is $\{r_{ij}, 1 \le i \le 1000, 1 \le j \le k_i\}$.

I generated synthetic data where each user generated 10 sessions; for each i, user i's session had mean rate that is $\mu_i \sim \text{Exp}(1/1000)$; and each of the k_i sessions of user i are of rate $\mathcal{N}(\mu_i, 1)$.

We estimate the average delivered bit rate as:

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k_i} r_{ij}}{\sum_{i=1}^{n} k_i}.$$

This estimate is 960.6.

Example 4: Block bootstrap

What is the standard error of this estimate? *Naive approach:* take sample standard deviation of the per session bit rates, divided by square root of number of sessions. For my synthetic dataset this gave $\widehat{\mathrm{SE}} = 9.35$.

This is not quite right however, because sessions are not independent: sessions belonging to the same user are likely to have similar bit rate.

How to adjust for this correlation?

Example 4: Block bootstrap

Recreate the data generating process with a *block bootstrap*:

- Sample *users* ($n = 1000$) with replacement.
- Each time a user is sampled, include *all* of their sessions.
- The bootstrap distribution is the resulting histogram of $\hat{\mu}^{(b)}$, over each bootstrap sample b.

Using this approach gave a standard error of $\widehat{\text{SE}}_{\text{BS}} = 22.67$ (much larger).

Bootstrap for regression

Linear regression via OLS

Given data $\mathbf{X}$ and $\mathbf{Y}$, suppose we want bootstrap standard errors for OLS coefficients.

For $1 \leq b \leq B$:

- Draw n samples (outcome *and* corresponding covariates) uniformly at random, with replacement, from $(\mathbf{X}, \mathbf{Y})$. (This is called *case resampling*.)
- Given the resulting data in the b'th sample, run OLS and compute the resulting coefficient vector $\hat{\boldsymbol{\beta}}^{(b)}$.

This gives the bootstrap distribution of $\hat{\boldsymbol{\beta}}$, and we can use it to, e.g., compute standard errors or confidence intervals (or even bias) for each of the coefficients.

Example 5: Heteroskedastic data

Suppose for $1 \leq i \leq 100$, $X_i \sim \mathcal{N}(0, 1)$, and $Y_i = X_i + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, X_i^4)$. This data exhibits *heteroskedasticity*: the error variance is not constant.

In this case the linear normal model assumptions are violated. What is the effect on the standard error that R reports?

We use the bootstrap to find out.

Example 5: Heteroskedastic data

Here is the code to run the bootstrap:

```
df = data.frame(X,Y)

coef_boot <- function(data, indices) {
  fit <- lm(Y ~ 1 + X, data = data[indices, ])
  return(coef(fit))
}

# Run bootstrap
boot_reg_result <- boot(df, coef_boot, R = 10000)
```

Example 5: Heteroskedastic data

If we run `lm(data = df, Y ~ 1 + X)`, then R reports a standard error of 0.209 on the coefficient on X, which is 1.1167.

But using the bootstrap approach, we compute a standard error of 0.423!

Why the difference? R's standard error assumes homoskedasticity (constant error variance). When this assumption is violated, as here, the reported standard error is too optimistic.

The bootstrap correctly accounts for the heteroskedasticity by resampling the actual data, giving a more accurate (and larger) standard error.

Other applications

The bootstrap is an incredibly flexible tool, with many variations developed to make it applicable in a wide range of settings.

For example, it can be used to measure variability in model estimates for more other modeling strategies (e.g., ridge, lasso).

In fact, it can even be used to estimate test error in a prediction setting (instead of cross validation).[1]

[1] The basic idea here is that bias and variance are measured over "parallel universes" of the training data; the bootstrap let's us simulate these parallel universes.