Bias, MSE, Studentized Intervals and The Jackknife

Mark M. Fredrickson (mfredric@umich.edu)

Computational Methods in Statistics and Data Science (Stats 406)

Estimating Bias and MSE using Bootstrap

Definitions of Bias and MSE

For a **parameter** θ and an **estimator** $\hat{\theta}$ the **bias** is defined as:

$$\mathsf{E}\left(\hat{\theta} - \theta\right)$$

The **mean squared error** is defined as:

$$\mathsf{E}\left((\hat{\theta} - \theta)^2\right)$$

When doing Monte Carlo estimates of bias and MSE, we **fixed the true parameter value** and then drew samples.

Key idea of bootstrap: Use $\hat{\theta}$ for θ, and $\hat{\theta}^*$ (bootstrap replications) for sampling distribution.

Estimation

To be more explicit:

- Bias:

$$\frac{1}{B}\sum_{j=1}^{B}(\hat{\theta}_j^* - \hat{\theta})$$

- MSE:

$$\frac{1}{B}\sum_{j=1}^{B}(\hat{\theta}_j^* - \hat{\theta})^2$$

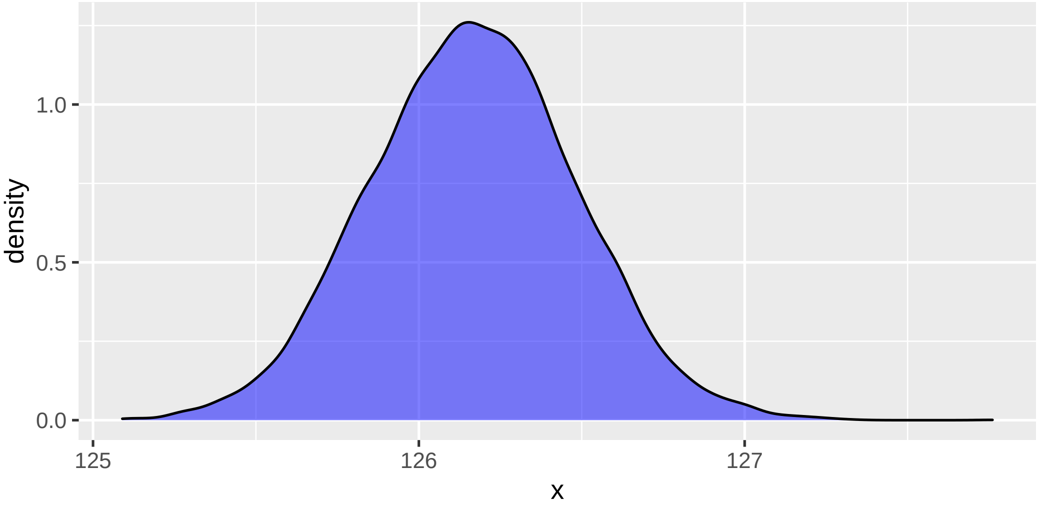
Bootstrapping Trimmed Mean (NHANES)

```
> (observed_trim <- trimmed_mean(sys_mean, p = 0.2))

[1] 126.2

> B <- 10000
> n <- length(sys_mean)
> bootstrap_samples <- rerun(B,
+                           sample(sys_mean, size = n, replace = TRUE))
> bootstrap_trims <- map_dbl(bootstrap_samples, trimmed_mean, p = 0.2)
```

Trimmed Mean Bootstrap Distribution



Estimating Bias/MSE for Trimmed Mean

```
> ## bias
> mean(bootstrap_trims - observed_trim)
```

```
[1] -0.07389
```

```
> ## MSE
> mean((bootstrap_trims - observed_trim)^2)
```

```
[1] 0.1053
```

For comparison, the bootstrapped **sample mean** estimated MSE:

```
> mean((bootstrap_means - observed_mean)^2)
```

```
[1] 0.09719
```

Studentized Bootstrap CIs

Studentization

Suppose $X_1, \ldots, X_n \sim N(\mu, \sigma^2)$ (independent). Then

$$W = \frac{\bar{X} - \mu}{(S^2/n)^{1/2}}$$

has a Student's t-distribution with $n - 1$ degrees of freedom.

More generally we say that a statistic is **studentized** if we subtract off a hypothesized location parameter and divide by an estimate of the standard deviation of the estimator.

Bootstrap-t (percentile) confidence intervals

Define the "studentized" bootstrap replicate

$$W^* = \frac{T^* - T}{\hat{\sigma}^*} \approx \frac{T^* - \theta}{\sigma}$$

Then we have the following percentile type interval:

$$P(0 \in [W^*_{\alpha/2}, W^*_{1-\alpha/2}]) > \alpha \quad \text{(0 because } (T - \theta)/\sigma \approx 0\text{)}$$

Undo the studentization to get back to the T scale:

$$\alpha/2 = P(W^* \leq W_{\alpha/2}) = P\left(\frac{T^* - \theta}{\sigma} \leq W_{\alpha/2}\right) = P(\theta \leq T^* - \sigma W_{\alpha/2})$$

Sticking in our estimates of T for θ and $\hat{\sigma}$ for σ:

$$[T - \hat{\sigma} W^*_{1-\alpha/2}, T - \hat{\sigma} W^*_{\alpha/2}]$$

Variance Estimators

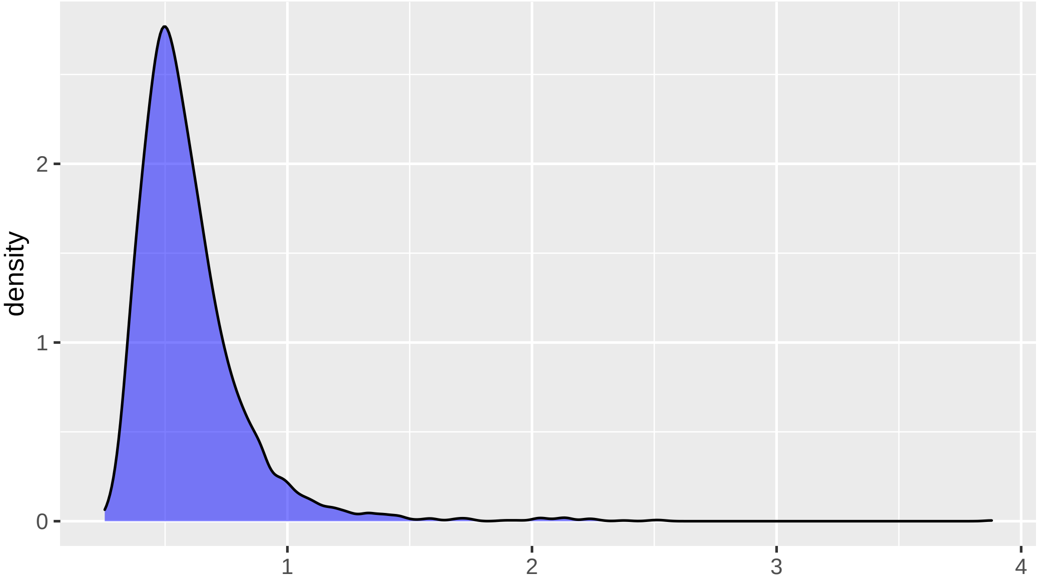
In the previous algorithm, we used **two different variance estimators** (for notational ease, I'm going to write these using standard deviations instead):

- $\hat{\sigma}^*$: estimates $\text{Var}(T^*)^{1/2}$ based on a particular bootstrap sample
- $\hat{\sigma}$: estimates $\text{Var}(T)^{1/2}$ based on the original sample

For either of these we could use

- Theoretical sample variance (e.g., variance of the sample mean)
- Bootstrap estimate of variance ("nested bootstrap")
- The Jackknife (which we'll discuss a bit later)

Log Ratio of Systolic to Diastolic



Bootstrapping the mean

```
> library(boot)
> mean_boot <- function(x, index) { mean(x[index]) }
> boot_mean <- boot(log(sysdia_ratio), statistic = mean_boot, R = 1000)
```

```
> boot.ci(boot_mean, type = c("norm", "basic", "perc"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot_mean, type = c("norm", "basic", "perc"))

Intervals :
Level      Normal              Basic              Percentile
95%   ( 0.5932,  0.6088 )   ( 0.5928,  0.6088 )   ( 0.5931,  0.6091 )
Calculations and Intervals on Original Scale
```

Variance of Sample Mean

$$
\begin{aligned}
\mathrm{Var}\left(\bar{X}\right) &= \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) && \text{(definition)} \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} \mathrm{Var}\left(X_i\right) + 2\sum_{i<j} \mathrm{Cov}\left(X_i, X_j\right)\right) && \text{(variance of a sum)} \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} \mathrm{Var}\left(X_i\right)\right) && \text{(independence} \Rightarrow \text{no covariance)} \\
&= \frac{1}{n^2} n\mathrm{Var}\left(X\right) && \text{(identical)} \\
&= \frac{1}{n}\mathrm{Var}\left(X\right)
\end{aligned}
$$

We estimate $\mathrm{Var}\left(X\right)$ using

$$
S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2
$$

Bootstrap-t: sample mean with sample variance estimator

```
> B <- 1000
> lsdr <- log(sysdia_ratio)
> n <- length(lsdr)
> est_t <- mean(lsdr)
> est_var_t <- var(lsdr) / n
```

```
> boot_sv <- replicate(B, {
+     xstar <- sample(lsdr, replace = TRUE)
+     (mean(xstar) - est_t) / sqrt(var(xstar) / n)
+ })
> (boot_ci_svar <- est_t - sqrt(est_var_t) *
+     quantile(boot_sv, c(0.975, 0.025)))

    97.5%      2.5%
0.5939491 0.6091313
```

Nested bootstrap

The variance of the sampling distribution for the sample mean is relatively easy to find.

For other statistics, is more difficult (e.g., trimmed mean). But we can use **a bootstrap estimate of variance** (like we did when forming asymptotic intervals for the trimmed mean).

Algorithm: for each bootstrap sample $\mathbf{X}^*$, run a separate bootstrap for the variance **by resampling from $\mathbf{X}^*$** (bootstrap within bootstrap/nested bootstrap).

Bootstrap-t: Nested bootstrap

```
> boot_for_var <- replicate(100, {
+     xstar <- sample(lsdr, replace = TRUE)
+     mean(xstar)
+ })
> boot_var_est <- var(boot_for_var)
```

```
> boot_boot <- replicate(100, {
+     xstar <- sample(lsdr, replace = TRUE)
+     xstar_boot <- replicate(100, {
+         xstarstar <- sample(xstar, replace = TRUE)
+         mean(xstarstar)
+     })
+     (mean(xstar) - est_t) / sd(xstar_boot)
+ })
> (boot_ci_boot <- est_t - sqrt(boot_var_est) *
+      quantile(boot_boot, c(0.975, 0.025)))

    97.5%      2.5%
0.5923056 0.6085139
```

Using the `boot` package

If we return **two values**, the `boot` package will treat the first as T^* and the second as $\hat{\sigma}^2_*$.

```
> mean_var <- function(x, index) {
+   xstar <- x[index]
+   n <- length(xstar)
+   c(mean(xstar), var(xstar) / n)
+ }
> boot_both <- boot(lsdr, mean_var, R = 1000)
```

```
> boot.ci(boot_both, type = 'stud')

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot_both, type = "stud")

Intervals :
Level    Studentized
95%   ( 0.5939,  0.6088 )
Calculations and Intervals on Original Scale
```

Comparing CIs

	Low	High	Rel. Width
Basic	0.593633389	0.608242475	1.000000000
Percentile	0.593693701	0.608302787	1.000000000
Studentized	0.593913162	0.608812719	1.019882847

Nested bootstrap of the median

With long tailed data (like the log-ratio were using), **the median** may be a better measure of **central tendency** than the mean.

We can't use sample variance estimate for the median, so we'll use **nested bootstrap**.

```
> median_idx <- function(data, idx) {
+   median(data[idx])
+ }
> median_nested <- function(data, idx) {
+     xstar <- data[idx]
+     meds <- boot(xstar, median_idx, R = 100)$t # just the T values
+     return(c(median(xstar), var(meds)))
+ }
```

Bootstrapping with Parallel Library

```
> library(parallel)
> cl <- makeCluster(detectCores())
> ## load the nested bootstrap components on the cluster
> ignore <- clusterEvalQ(cl, library(boot))
> clusterExport(cl, c("median_idx", "median_nested"))
> boot_median <- boot(lsdr, median_nested, R = 1000,
+   parallel = "snow", cl = cl, ncpus = detectCores())
> stopCluster(cl)
```

```
> boot.ci(boot_median, type = c("stud", "basic"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot_median, type = c("stud", "basic"))

Intervals :
Level      Basic              Studentized
95%   ( 0.5412,  0.5541 )   ( 0.5430,  0.5548 )
Calculations and Intervals on Original Scale
```

The Jackknife

Variance Estimation

We often have need to **compute or estimate the variance of an estimator** T.
(Note: we are referring to the variance of the **sampling distribution**.)

Why?

- To create **Normal theory confidence intervals**: $T \pm z_{\alpha/2}\sigma_T^2$.
- To compare **efficiency of estimators**.
- Combine with **bias** to get **Mean Squared Error**.
- Studentized confidence intervals: $(T^* - T)/\sigma^*$

Subsets of the data

We could use the **variance of the bootstrap distribution** to estimate the variance of T, but this is often **computationally burdensome**.

Idea: can we be more computationally efficient when $n < B$ (sample size less than number of bootstrap replications)?

With the bootstrap, we **generate new samples of size** n from the original sample.

Could use **subsets of the data** to investigate the variance of T?

Dropping one observation

The easiest possible subset is **dropping the jth observation**:

$$T_j = T(X_1, X_2, \ldots, X_{j-1}, X_{j+2}, \ldots, X_n)$$

Goal: when $T(X_1, \ldots, X_n)$ is the **sample mean**, combine the T_j to get the usual **sample variance estimate** (recall, $\mathrm{Var}\left(\bar{X}\right) = \mathrm{Var}\left(X\right)/n$):

$$S_X^2/n = \frac{1}{n(n-1)} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

What is T_j and $\bar{T}$?

The mean, dropping one observation:

$$T_j = \frac{1}{n-1} \sum_{i \neq j} X_i = \frac{n\bar{X} - X_j}{n-1}$$

What is $\bar{T} = n^{-1} \sum_{j=1}^{n} T_j$?

$$\bar{T} = \frac{1}{n} \sum_{j=1}^{n} \frac{n\bar{X} - X_j}{n-1} = \frac{1}{n(n-1)} \left(n^2 \bar{X} - n\bar{X} \right) = \frac{1}{n(n-1)} \left(\bar{X}(n(n-1)) \right) = \bar{X}$$

Variance of the T_j

The plugging the T_j values into the usual **sample variance estimator**,

$$
\begin{aligned}
S_T^2 &= \frac{1}{n-1}\sum_{j=1}^{n}(T_j - \bar{T})^2 \\
&= \frac{1}{n-1}\sum_{j=1}^{n}\left(\frac{n\bar{X} - X_j}{n-1} - \bar{X}\right)^2 \\
&= \frac{1}{(n-1)^3}\sum_{j=1}^{n}\left(n\bar{X} - X_j - (n-1)\bar{X}\right)^2 \\
&= \frac{1}{(n-1)^3}\sum_{j=1}^{n}\left(\bar{X} - X_j\right)^2 \\
&= \frac{1}{(n-1)^2} S_X^2
\end{aligned}
$$

Adjusting

We found that $S_T^2 = S_X^2/(n-1)^2$, but we wanted S_X^2/n. So correct by $(n-1)^2/n$,

$$v_J = \frac{(n-1)^2}{n} S_T^2 = \frac{n-1}{n} \sum_{i=1}^{n} (T_i - \bar{T})^2$$

We call this the (delete-1) **jackknife estimator of variance** and we can use it for "smooth" functions other than the sample mean.

Alternative motivation: estimating bias

We saw that we can use the bootstrap to **estimate the bias of an estimator**.

To do so, we compared the **mean of the bootstrap replications** to the **original sample statistic**.

Suppose we want to estimate the bias by leaving out one observation, so we get the bias estimate:

$$\hat{b} = (n-1)(\bar{T} - T)$$

(where $T = T(X_1, \ldots, X_n)$)

Naturally we might want to adjust our estimate using this estimated bias:

$$T_{\text{adj}} = T - \hat{b} = nT - (n-1)\bar{T}$$

Variance of bias adjusted estimate

From the last slide:

$$T_{\text{adj}} = T - \hat{b} = nT - (n-1)\bar{T}$$

Use this form to think about bias due to i:

$$W_i = nT - (n-1)T_i$$

If we ignored the dependence between the W_i, estimating the variance of $\bar{W}$,

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}(W_i - \bar{W})^2 = \frac{n-1}{n}\sum_{i=1}^{n}(T_i - \bar{T})^2 = v_J$$

Jackknife Notes

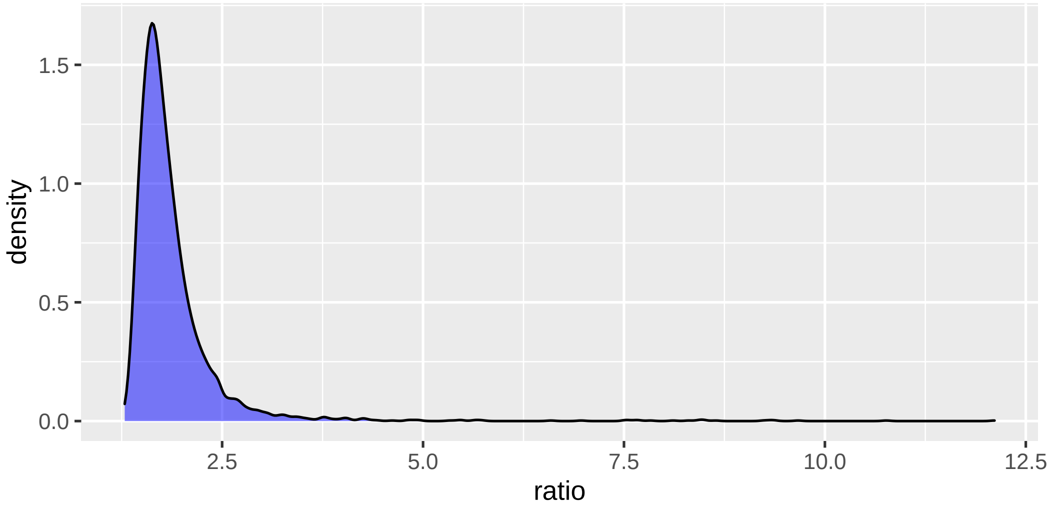
Personally, I find the direct variance estimate a little easier to think about than the bias correction motivation.

BUT! You may encounter jackknife bias correction, so it's good to know about.

For the sample mean, the $v_j = S^2/n$.

The jackknife can work for other statistics, provided they are sufficiently smooth.

We can also perform **delete-k** jackknife, but this requires evaluating more subsets.


1.5
1.0
0.5
0.0
density
2.5
5.0
7.5
10.0
12.5
ratio

Jackknife estimate of variance for sys-dia ratios

A useful R convention: **negative indexes get dropped**:

```
> n <- dim(nhanes)[1]
> tj <- map_dbl(1:n, function(i) { mean(log(nhanes$ratio)[-i])})

> (n - 1) / n * sum((tj - mean(tj))^2)

[1] 1.427501e-05

> (1/n) * var(log(nhanes$ratio))

[1] 1.427501e-05
```

Estimate of variance of the sample correlation

```
> tj <- map_dbl(1:n, function(i) {
+   with(nhanes, cor(sys_mean[-i], dia_mean[-i]))
+ })
> (vcor_J <- (n - 1) / n * sum((tj - mean(tj))^2))
```

```
[1] 0.0004447666
```

Studentized bootstrap intervals with the jackknife

```
> cor_vj <- function(x, index) {
+     n <- dim(x)[1]
+     sys <- x$sys_mean[index]
+     dia <- x$dia_mean[index]
+
+     tj <- map_dbl(1:n, function(i) {
+         cor(sys[-i], dia[-i], use = "complete")
+     })
+
+     c(cor(sys, dia, use = "complete"), (n - 1) / n * sum((tj - mean(tj))^2
+ }
> boot_cor_vj <- boot(nhanes, cor_vj, 1000, parallel = "snow", cl = cl, ncp
```

```
> boot.ci(boot_cor_vj, type = c("basic", "perc", "stud"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot_cor_vj, type = c("basic", "perc", "stud"))

Intervals :
Level      Basic              Studentized          Percentile
95%   ( 0.2566,  0.3399 )   ( 0.2547,  0.3377 )   ( 0.2534,  0.3367 )
Calculations and Intervals on Original Scale
```

Estimating Bias

Recall the definition of a **bias** (for T estimating θ):

$$\text{bias}(T) = E(T) - \theta$$

We've talked about **estimating bias of an estimator** using **bootstrap replications** T^*:

$$\hat{b} = \bar{T}^* - T$$

A theme of statistics is **any estimate should include a measure of uncertainty**. Let's add one to $\hat{b}$.

Jackknife-after-bootstrap

How could we (naively) use the jackknife to estimate the variance of $\hat{b}$?

- For $j = 1, \ldots, n$, drop the jth observation.
- Take B bootstrap samples from the other $n - 1$ observations and compute $\hat{b}_j$
- Compute $\frac{n-1}{n} \sum_{i=1}^{n} (\hat{b}_j - \bar{b})^2$

Computationally expensive!

Jackknife after bootstrap: **estimate** $\hat{b}_j$ using the bootstrap samples that **omit** X_j.

JAB for correlation

```
> cor_stat <- function(x, index) {
+     cor(x[index, 1], x[index, 2])
+ }
> cor_boot <- boot(nhanes[, c("sys_mean", "dia_mean")], cor_stat, R = 1000
```

```
> cor_array <- boot.array(cor_boot)
> cor_array[1:5, 1:5]

     [,1] [,2] [,3] [,4] [,5]
[1,]    4    0    0    0    1
[2,]    1    2    2    1    1
[3,]    0    1    1    2    2
[4,]    0    0    0    2    2
[5,]    0    0    4    3    1
```

```
> cor_sample <- with(nhanes, cor(sys_mean, dia_mean))
> bs <- apply(cor_array, 2, function(xi_used) {
+     excluded <- cor_boot$t[xi_used == 0]
+     mean(excluded) - cor_sample
+ })
> n <- dim(nhanes)[1]
> (jab_est_var <- (n - 1)/n * sum((bs - mean(bs))^2))

[1] 0.003110607
```

Interpreting results

Estimated bias is small relative to the estimated standard deviation

```
> (mean(cor_boot$t) - cor_sample) / sqrt(jab_est_var)
[1] -0.01457004
```

Generally, we don't worry much if this ratio is less than 0.25, and we are far below that.

Aside: it is tempting to use the estimated variance to create a confidence interval. I was unable to find justification for this in general.

Summary

- Key features of the **sampling distribution** of an estimator include **bias, variance, and MSE**.
- We can estimate these from the **full bootstrap distribution**, but sometimes it is useful to have computationally convenient estimators.
- **The jackknife** is a method for **dropping observations** to estimate bias and variance.
- In addition to comparing estimators on **bias and MSE**, variance is useful for creating CIs, particularly **studentized CIs**.
- Several extensions combine these tools (jackknife in studentized intervals, jackknife after bootstrap).