Bootstrap Confidence Intervals

Key ideas from last class

Algorithm:

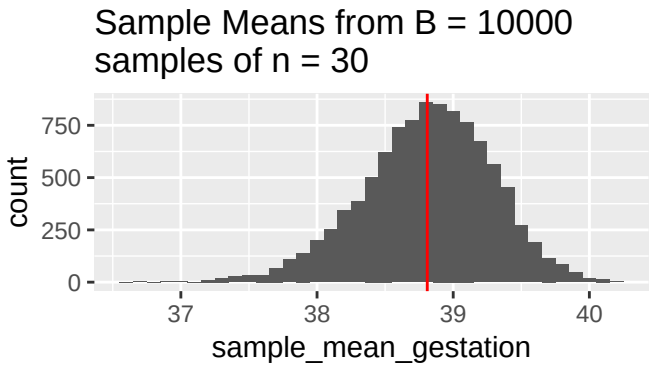
1. For $b = 1, \ldots, B$:
   a. Draw a bootstrap sample of size n **with replacement** from the observed data
   b. Calculate the estimate $\hat{\theta}_b$ based on that bootstrap sample (a number)
2. The distribution of estimates $\{\hat{\theta}_1, \ldots, \hat{\theta}_B\}$ from different simulated samples approximates the sampling distribution of the estimator $\hat{\theta}$ (the random variable).

Notation:

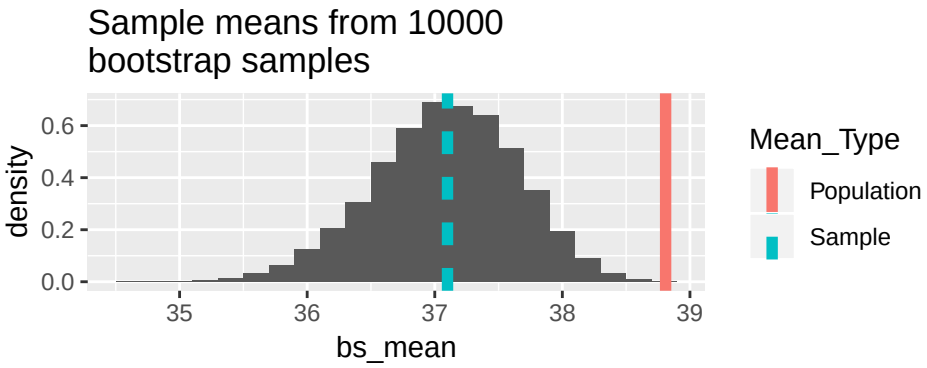
- θ is the unknown parameter to estimate
- $\hat{\theta}$ is the estimate from our sample
- $\hat{\theta}_b$ is the estimate from boostrap resample number b

Compare the approximations from sampling directly from the population and from bootstrap resampling:

Many means, based on samples from the population:



Many means, based on bootstrap resamples with replacement from the sample:

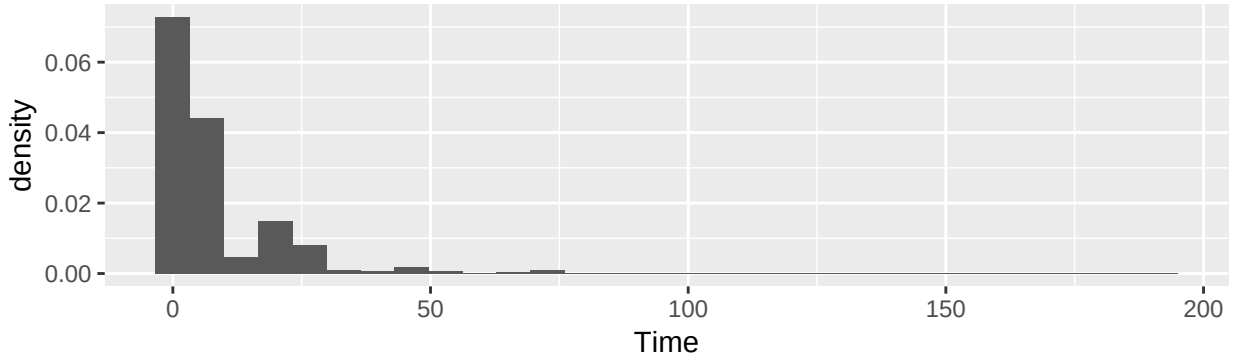


- The relationship of $\hat{\theta}$ to θ is like the relationship of $\hat{\theta}_b$ to $\hat{\theta}$:
  - Bootstrap distribution **reproduces shape and bias** of actual sampling distribution
  - Bootstrap distribution **does not reproduce mean** of actual sampling distribution
    - E.g., centered at sample mean instead of population mean

Verizon Repair Times

This example is taken from Hesterberg (2014). We have data on the amount of time it took Verizon to repair problems in their telephone lines in the state of New York. They were brought to court, accused of taking longer than their contractual obligation to do repairs. For legal reasons, there was interest in estimating the mean repair time.

```
library(readr)
library(dplyr)
library(ggplot2)

verizon <- read_csv("http://www.evanlray.com/data/chihara_hesterberg/Verizon.csv")
verizon_ilec <- verizon %>%
  filter(Group == "ILEC")

ggplot(data = verizon_ilec, mapping = aes(x = Time)) +
  geom_histogram(mapping = aes(y = ..density..))
```



Let's compare three interval estimates for the mean repair time:

- t, from normal theory (intro stats)
- bootstrap percentile
- bootstrap t

In all cases the estimate is $\hat{\theta} = 8.412$

```
mean(verizon_ilec$Time)
```

```
## [1] 8.411611
```

Confidence Interval Idea #1: Bootstrap Percentile CI

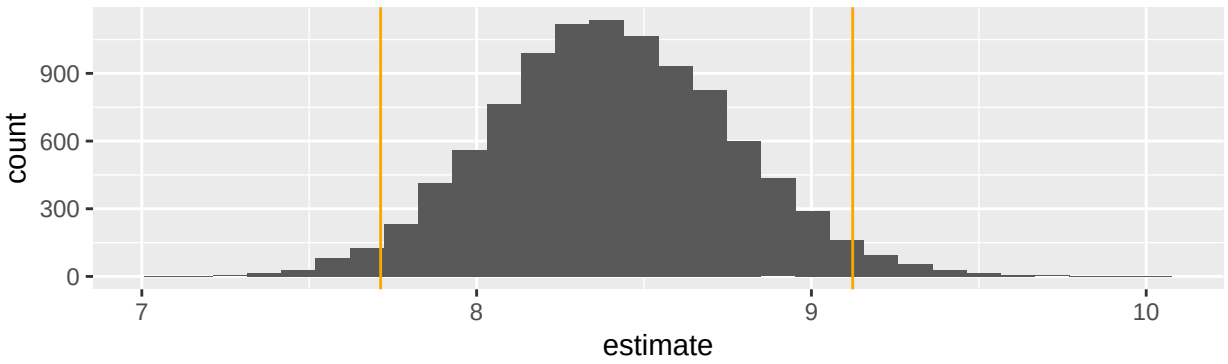
Take the bootstrap sampling distribution and chop off the left and right tails.

- Probably what you have seen previously
- Essentially unjustifiable unless sampling distribution is symmetric
- Regardless of no great formal justification, tends to work well for moderate to large sample sizes
- For small n, actual coverage rate $\neq$ nominal coverage rate

```
# sample size
n <- nrow(verizon_ilec)

# how many bootstrap samples to take, and storage space for the results
num_samples <- 10^4
bs_percentile_results <- data.frame(
  estimate = rep(NA, num_samples)
)

# draw many samples from the observed data and calculate mean of each simulated sample
for(i in seq_len(num_samples)) {
  ## Draw a bootstrap sample of size n with replacement from the observed data
  bs_sample <- verizon_ilec %>%
    sample_n(size = n, replace = TRUE)

  ## Calculate mean of bootstrap sample
  bs_percentile_results$estimate[i] <- mean(bs_sample$Time)
}

# 95% Bootstrap Percentile Interval
bs_percentile_interval <- quantile(bs_percentile_results$estimate, prob = c(0.025, 0.975))
bs_percentile_interval
```

```
##     2.5%    97.5%
## 7.713295 9.123312
```

```
# Plot
ggplot(data = bs_percentile_results, mapping = aes(x = estimate)) +
  geom_histogram() +
  geom_vline(xintercept = bs_percentile_interval, color = "orange")
```



Confidence Interval Idea #2: Bootstrap t

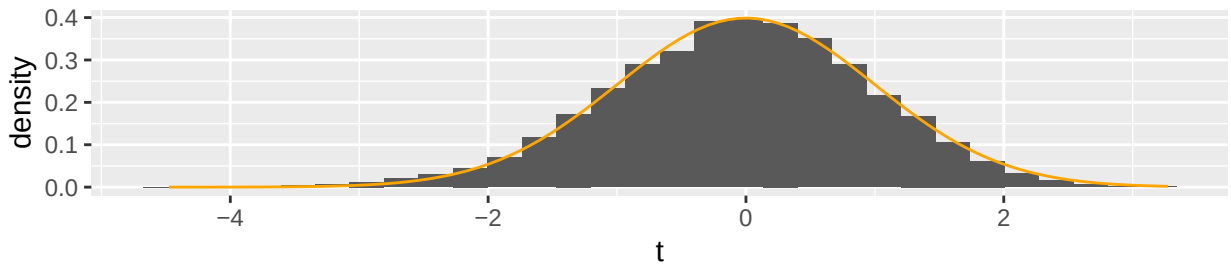
- The t statistic is $t = \frac{\hat{\theta}-\theta}{SE(\hat{\theta})}$.
- If $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \text{Normal}(\theta, \sigma^2)$ (or similar) then the exact distribution of t can be derived.
- Otherwise, approximate the sampling distribution of the t statistic using the bootstrap
- Algorithm:

1. For $b = 1, \ldots, B$:
   a. Draw a bootstrap sample of size n **with replacement** from the observed data
   b. Calculate $t_b = \frac{\hat{\theta}_b - \hat{\theta}}{SE(\hat{\theta}_b)}$ based on that bootstrap sample 2. The distribution of t statistics $\{t_1, \ldots, t_B\}$ from different simulated samples approximates the sampling distribution of the t statistic.
2. Form a confidence interval as $[\hat{\theta} - q(1-\frac{\alpha}{2})SE(\hat{\theta}), \hat{\theta} - q(\frac{\alpha}{2})SE(\hat{\theta})]$, where $q(a)$ is the a'th quantile of the bootstrap t distribution.

```
# sample size
n <- nrow(verizon_ilec)

# how many bootstrap samples to take, and storage space for the results
num_samples <- 10^4
bs_t_results <- data.frame(
  t = rep(NA, num_samples)
)

# draw many samples from the observed data and calculate mean of each simulated sample
for(i in seq_len(num_samples)) {
  ## Draw a bootstrap sample of size n with replacement from the observed data
  bs_sample <- verizon_ilec %>%
    sample_n(size = n, replace = TRUE)

  ## Calculate t statistic based on bootstrap sample
  bs_t_results$t[i] <- (mean(bs_sample$Time) - mean(verizon_ilec$Time)) /
    (sd(bs_sample$Time) / sqrt(n))
}

# 95% Bootstrap t Interval
bs_t_interval <- c(
  mean(verizon_ilec$Time) -
    quantile(bs_t_results$t, prob = 0.975) * sd(verizon_ilec$Time) / sqrt(n),
  mean(verizon_ilec$Time) -
    quantile(bs_t_results$t, prob = 0.025) * sd(verizon_ilec$Time) / sqrt(n)
)

bs_t_interval
```

```
##    97.5%     2.5%
## 7.750814 9.190865
```

```
# Plot to compare bootstrap estimate of distribution of t statistic to theoretical result
ggplot(data = bs_t_results, mapping = aes(x = t)) +
  geom_histogram(mapping = aes(y = ..density..)) +
  stat_function(fun = dt, args = list(df = n - 1), color = "orange")
```



Q: What if we didn't have a formula for $SE(\hat{\theta}_b)$?

A: Bootstrap standard error, *nested within the outer bootstrap.*

This is really really slow.

Here's a statement of our full algorithm (omitting pre-allocation of storage space for clarity, though that is an important implementation detail to keep it from being too slow in R):

1. For $b = 1, ..., B_1$
   i. Draw a bootstrap sample from the original data, with replacement
   ii. For $j = 1, ..., B_2$
      a. Draw a bootstrap sample from the bootstrap sample obtained in step 1 i, with replacement
      b. Calculate $\hat{\theta}_j$ based on the bootstrap sample from step 1 ii a
   iii. Calculate $SE(\hat{\theta}_b) = \sqrt{\frac{1}{B_2-1}\sum_{j=1}^{B_2}(\hat{\theta}_j - \frac{1}{B_2}\sum_{k=1}^{B_2}\hat{\theta}_k)^2}$
   iv. Calculate $t_b = \frac{\hat{\theta}_b - \hat{\theta}}{SE(\hat{\theta}_b)}$
2. For $b = 1, \ldots, B_3$
   i. Draw a bootstrap sample from the original data, with replacement
   ii. Calculate $\hat{\theta}_b$ based on the bootstrap sample from step 3 i
3. Calculate $SE(\hat{\theta}) = \sqrt{\frac{1}{B_3-1}\sum_{b=1}^{B_3}(\hat{\theta}_b - \frac{1}{B_3}\sum_{c=1}^{B_3}\hat{\theta}_c)^2}$
4. Calculate $\hat{\theta}$ based on the original observed data
5. Calculate the confidence interval as $[\hat{\theta} - q(1-\frac{\alpha}{2})SE(\hat{\theta}), \hat{\theta} - q(\frac{\alpha}{2})SE(\hat{\theta})]$, where $q(1-\frac{\alpha}{2})$ and $q(\frac{\alpha}{2})$ are quantiles of the bootstrap estimate of the sampling distribution of the t statistic obtained in step 1, $\hat{\theta}$ was computed in step 2, and $SE(\hat{\theta})$ was computed in step 4.

```
run_time <- system.time({
  # sample size
  n <- nrow(verizon_ilec)

  # how many bootstrap samples to take, and storage space for the results
  num_samples <- 10^3
  bs_t_results <- data.frame(
    t = rep(NA, num_samples)
  )

  num_inner_samples <- 10^3 # fewer to make this take an achievable amount of time
  inner_bs_se_results <- data.frame(
    theta = rep(NA, num_inner_samples)
  )

  # Step 1
  for(b in seq_len(num_samples)) {
    # Step 1 i: Draw a bootstrap sample of size n with replacement from the observed data
    bs_sample <- verizon_ilec %>%
      sample_n(size = n, replace = TRUE)

    # Step 1 ii: Use a nested bootstrap to estimate SE(\hat{\theta}_b), based on this bootstrap sample
    for(j in seq_len(num_inner_samples)) {
      # Step 1 ii a
      inner_bs_sample <- bs_sample %>%
        sample_n(size = n, replace = TRUE)

      # Step 1 ii b
      inner_bs_se_results$theta[j] <- mean(inner_bs_sample$Time)
    }
    # Step 1 iii and iv
    bs_se <- sd(inner_bs_se_results$theta)

    # Step 1 iv: Calculate t statistic based on bootstrap sample
```

```
    bs_t_results$t[b] <- (mean(bs_sample$Time) - mean(verizon_ilec$Time)) / bs_se
  }
})

# Step 2
bs_se_results <- data.frame(
  theta = rep(NA, num_samples)
)
for(b in seq_len(num_samples)) {
  # Step 2 i
  bs_sample <- verizon_ilec %>%
    sample_n(size = n, replace = TRUE)

  # Step 2 ii
  bs_se_results$theta[b] <- mean(bs_sample$Time)
}
# Step 3
bs_se <- sd(bs_se_results$theta)

# Steps 4 and 5: 95% Bootstrap t Interval
bs_t_interval <- c(
  mean(verizon_ilec$Time) -
    quantile(bs_t_results$t, prob = 0.975) * bs_se,
  mean(verizon_ilec$Time) -
    quantile(bs_t_results$t, prob = 0.025) * bs_se
)

bs_t_interval
```

```
##    97.5%     2.5%
## 7.739686 9.126088
```

```
run_time
```

```
##    user  system elapsed
## 217.224   1.779 220.075
```