Bootstrap Confidence Intervals

Math 445, Spring 2016

One Sample Bootstrap

Motivating example

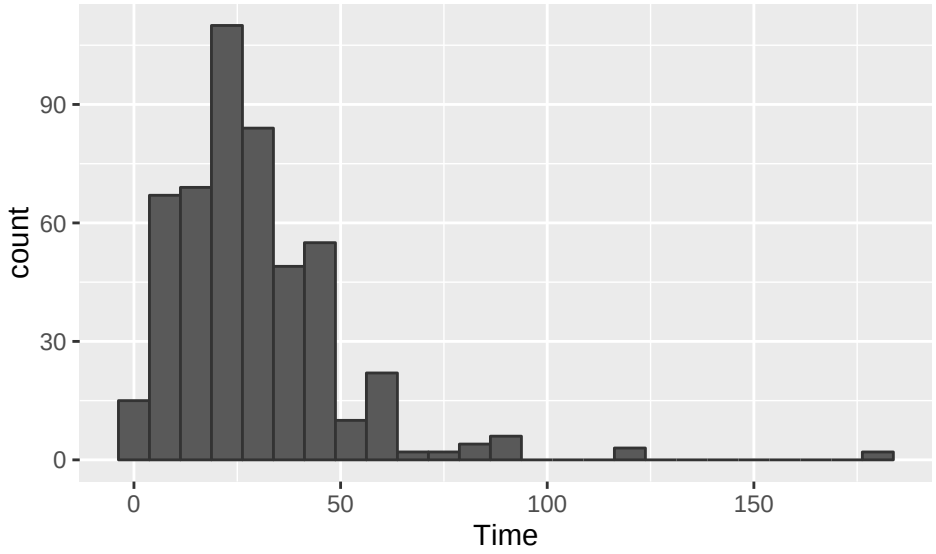
The file `CommuteAtlanta.csv` contains consists of a random sample of 500 people living in Atlanta, and provides information on their commute. The data were obtained from the U.S. Census Bureau's American Housing Survey (AHS), which contains information about housing and living conditions for samples from certain metropolitan areas. This sample includes only individuals that worked somewhere outside of their home.

```
commute <- read.csv("data/CommuteAtlanta.csv")
str(commute)
```

```
## 'data.frame':    500 obs. of  5 variables:
##  $ City    : Factor w/ 1 level "Atlanta": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Age     : int  19 55 48 45 48 43 48 41 47 39 ...
##  $ Distance: int  10 45 12 4 15 33 15 4 25 1 ...
##  $ Time    : int  15 60 45 10 30 60 45 10 25 15 ...
##  $ Sex     : Factor w/ 2 levels "F","M": 2 2 2 1 1 2 2 1 2 1 ...
```

Below is a histogram of the commute times:

```
ggplot(data = commute) +
  geom_histogram(mapping = aes(x = Time), colour = "grey20", binwidth = 7.5)
```



Suppose that you are interested in moving to Atlanta and want to better understand the average commute time. It is easy to compute an average commute time from this sample, but what does this tell you about all commuters in Atlanta (i.e. the population)?

```r
# sample mean
mean(commute$Time)
```

```
## [1] 29.11
```

```r
# sample standard deviation
sd(commute$Time)
```

```
## [1] 20.71831
```

Target: a plausible range of values for the average commute time of all commuters in Atlanta, GA.

Algorithm

Given a sample of size n from a population,

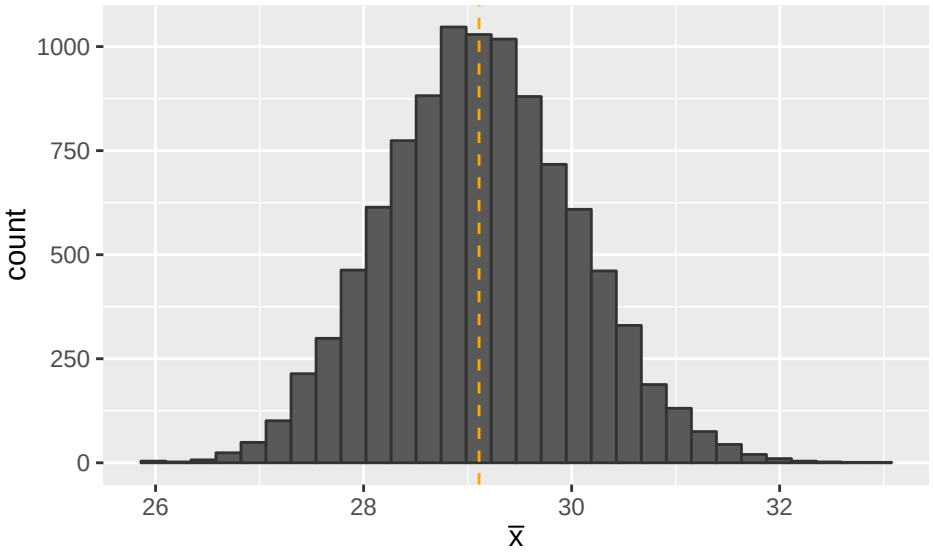
1. Draw a resample of size n with replacement from the sample. Compute a statistic that describes the sample, such as the sample mean.
2. Repeat this resampling process many times, say 10,000.
3. Construct the bootstrap distribution of the statistic. Inspect its spread, bias, and shape.

Example implementation

The below code snippet boostraps our sample of 500 commuters in Atlanta in an effort to better understand the average commute time.

```r
n <- nrow(commute)
N <- 10^4
time_mean <- numeric(N)
for (i in 1:N) {
  x <- sample(commute$Time, size = n, replace = TRUE)
  time_mean[i] <- mean(x)
}
```

Bootstrap distribution



```r
# bootstrap mean
mean(time_mean)
```

```
## [1] 29.12476
```

```r
# bias
mean(time_mean) - mean(commute$Time)
```

```
## [1] 0.014756
```

```r
# standard error
sd(time_mean)
```

```
## [1] 0.9179816
```

Bootstrap percentile confidence intervals

```r
quantile(time_mean, probs = c(.025, .975))
```

```
##     2.5%    97.5%
## 27.38195 30.98005
```

Two Sample Bootstrap

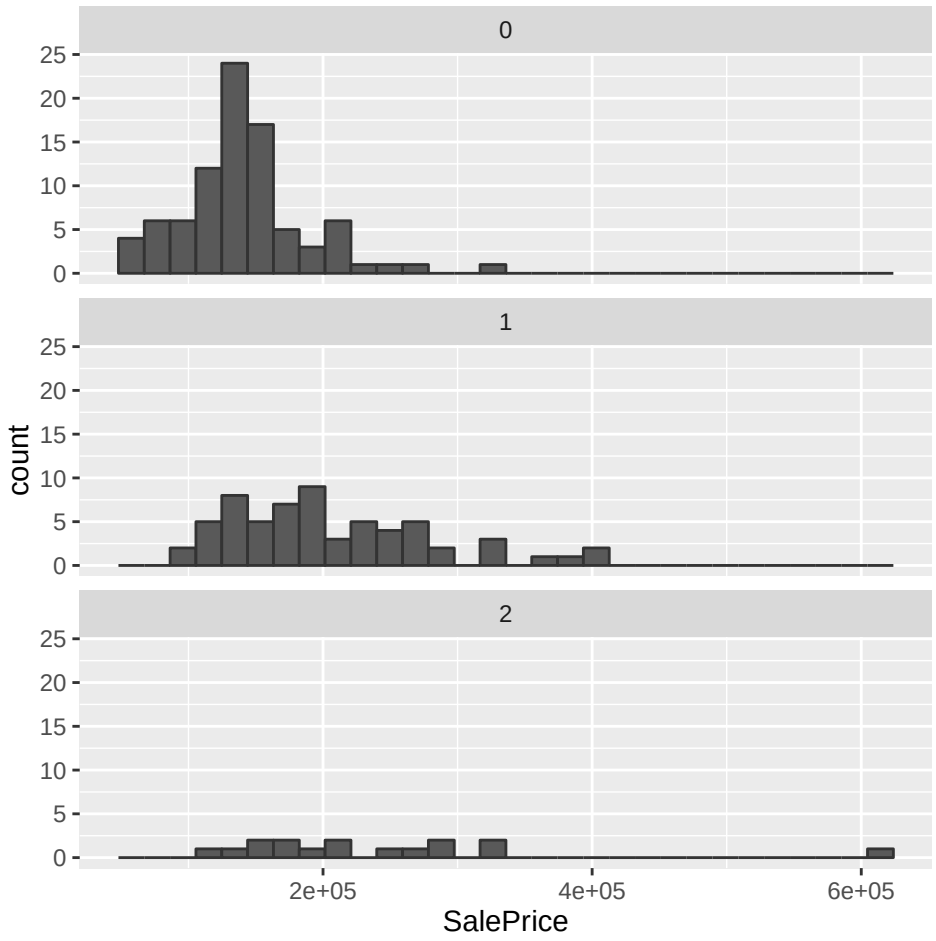
Motivating example

Are houses with fireplaces more expensive? The file `AmesHousing2010.csv` contains a random sample of houses sold in Ames, IA during 2010. The data set contains details of the sales and the property. Suppose that you are interested in estimating the difference in average price between houses with and without a fireplace. This is a situation where the **two sample bootstrap** can be used to construct a confidence interval.

```r
housing <- read.csv("data/AmesHousing2010.csv")
```

First, let's see what the data look like

```r
ggplot(data = housing) +
  geom_histogram(mapping = aes(x = SalePrice), colour = "gray20") +
  facet_wrap(~Fireplaces, ncol = 1)
```



Suppose further that you are only interested in houses with either no fireplace or one fireplace

```
library(dplyr)
sub_housing <- filter(housing, Fireplaces <= 1)

# Quick summary stats
price_stats <-
  sub_housing %>%
  group_by(Fireplaces) %>%
  summarise(min = min(SalePrice),
            Q1 = quantile(SalePrice, .25),
            median = median(SalePrice),
            Q3 = quantile(SalePrice, .75),
            max = max(SalePrice),
            mean = mean(SalePrice),
            sd = sd(SalePrice),
            n = n())
price_stats
```

```
## Source: local data frame [2 x 9]
##
##   Fireplaces   min     Q1 median     Q3    max     mean       sd     n
##        (int) (int)  (dbl)  (dbl)  (dbl)  (int)    (dbl)    (dbl) (int)
## 1          0 55000 117750 136500 155000 328000 140676.4 46175.91    87
```

```
## 2          1 99500 144875 189000 251675 410000 205717.8 75590.35    62
```

So we see that our **point estimate** is

```
obs_diff_means <- price_stats$mean[2] - price_stats$mean[1]
obs_diff_means
```

```
## [1] 65041.4
```

Algorithm

Given independent samples of sizes m and n from two populations,

1. Draw a resample of size m with replacement from the first sample and a separate resample of size n from the second sample. Compute a statistic that compares the two groups, such as the difference between the two sample means.
2. Repeat this resampling process many times, say 10,000.
3. Construct the bootstrap distribution of the statistic. Inspect its spread, bias, and shape.

Example implementation

The below code snippet performs the two sample bootstrap in order to construct a confidence interval for the difference in average sales price of homes in Ames, IA with and without fireplaces.

Since we have many extra columsn in our data set, it's easier to first select only the two columns of interest.

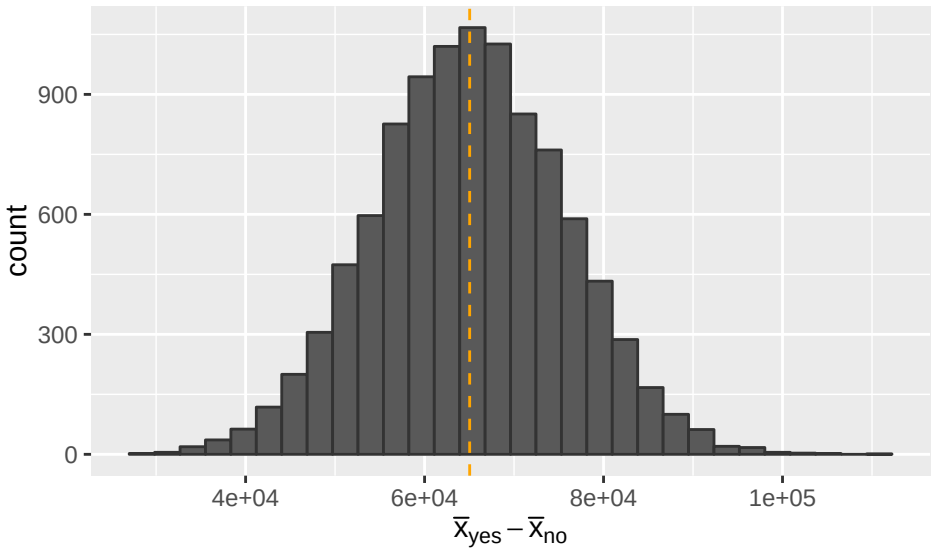
```
boot_df <- select(sub_housing, SalePrice, Fireplaces)
```

```
N <- 10^4

no_fp <- subset(boot_df, select = SalePrice, subset = Fireplaces == 0, drop = TRUE)
fp <- subset(boot_df, select = SalePrice, subset = Fireplaces == 1, drop = TRUE)

price_diff_mean <- numeric(N)
for (i in 1:N) {
  no_fp_sample <- sample(no_fp, replace = TRUE)
  fp_sample <- sample(fp, replace = TRUE)
  price_diff_mean[i] <- mean(fp_sample) - mean(no_fp_sample)
}
```

Bootstrap distribution




```
# bootstrap mean
mean(price_diff_mean)
```

```
## [1] 64996.84
```

```
# bias
mean(price_diff_mean) - obs_diff_means
```

```
## [1] -44.56705
```

```
# standard error
sd(price_diff_mean)
```

```
## [1] 10660.71
```

```
# bias/se
(mean(price_diff_mean) - obs_diff_means) / sd(price_diff_mean)
```

```
## [1] -0.004180496
```

Bootstrap percentile confidence intervals

```
quantile(price_diff_mean, probs = c(.05, .95))
```

```
##       5%      95%
## 47452.09 82551.83
```

Other statistics

Instead of focusing on the difference in means, we could instead focus on the ratio of means.

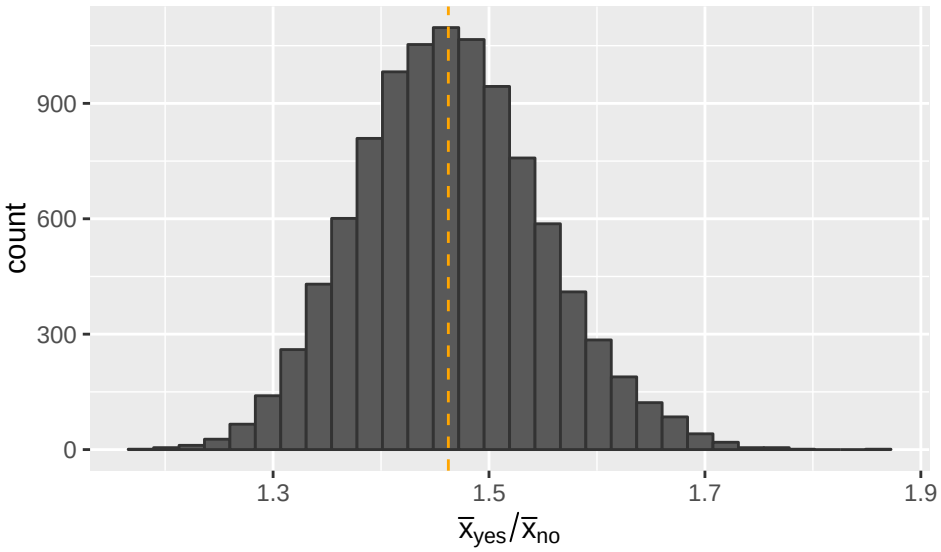
Implementation

```
N <- 10^4

no_fp <- subset(boot_df, select = SalePrice, subset = Fireplaces == 0, drop = TRUE)
fp <- subset(boot_df, select = SalePrice, subset = Fireplaces == 1, drop = TRUE)

price_ratio_mean <- numeric(N)
for (i in 1:N) {
  no_fp_sample <- sample(no_fp, replace = TRUE)
  fp_sample <- sample(fp, replace = TRUE)
  price_ratio_mean[i] <- mean(fp_sample) / mean(no_fp_sample)
}
```

Bootstrap distribution




```
# bootstrap mean
mean(obs_ratio_mean)
```

```
## [1] 1.462348
```

```
# bias
mean(price_ratio_mean) - obs_ratio_mean
```

```
## [1] 0.001793249
```

```
# standard error
sd(price_ratio_mean)
```

```
## [1] 0.08500633
```

```r
# bias/se
(mean(price_ratio_mean) - obs_ratio_mean) / sd(price_ratio_mean)
```

```
## [1] 0.02109547
```

Bootstrap percentile confidence intervals

```r
quantile(obs_ratio_mean, probs = c(.05, .95))
```

```
##       5%      95%
## 1.462348 1.462348
```