Week 14: Bootstrap

Marcelo Coca Perraillon

University of Colorado
Anschutz Medical Campus

Health Services Research Methods I
HSMP 7607
2018

Updated notes are here: https://clas.ucdenver.edu/marcelo-perraillon/teaching/health-services-research-methods-i-hsmp-7607

Outline

- Review of standard errors
- The magic of bootstrapping
- Caveats

Standard error, reminder

- Back in the days (week 4 of this class) we covered **standard errors**
- We have an estimator, say, the mean of a sample $\bar{X} = \sum_{i=1}^{n} X_i/n$ or a proportion $\hat{p}$
- The parameter estimate has some error and a distribution (not the same as the standard deviation of the data)
- If we know the distribution of the parameter and its standard error, then we can build confidence intervals and do hypothesis testing
- In the context of linear regression, we know that $\hat{\beta}_j$ distributes normal and we have a formula for its standard error (the **variance-covariance** matrix, really). We know this because of **theory**

Derivation

- We use **statistical theory** to derive standard errors
- In the linear model, we use the central limit theorem, the law of large numbers, and the assumption of iid errors that are normally distributed
- We needed all that to come up with formulas for the standard error
- The logic of standard errors is a lot easier to understand using simulations

Example

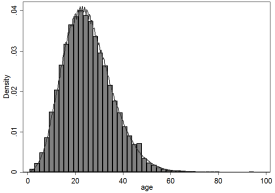
- Say that we have a population of 40,000 observations
- We will take a sample of 150 observations out of the 40,000 (recall, that in theory, we assume that the population is **infinitely** large)
- We will take the mean of the 150 observations
- If we could **repeat this experiment many times**, we could calculate the mean many times and see how it distributes (that's why this way of thinking about statistics is called **frequentist**)
- Keep in mind: we want to understand how the mean distributes, not the distribution of the 150 observations or the 40,000
- We will do this 1,000 times. In a real life example, we can't do this. We just get a sample of 150 observations. We can't repeat the experiments many times

Example

- We will create a population of 40,000 people and simulate their "age" with $(N(5,1))^2$
- I take the square to avoid negative ages; it will distribute Chi-square. Also, I'll remove the decimals

```
clear
set seed 1234567
set obs 40000
gen age = int((rnormal(5,1))^2)

sum age
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |     40,000    25.49595    10.12632          1         95
```



Example

- Next, we take a sample of 150 and calculate the mean
- We repeat 1,000 times so we have a distribution for the mean and calculate the standard deviation of the means (i.e. the standard error)
- It will take a while...

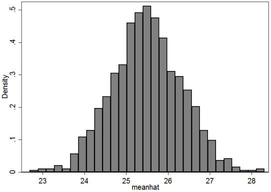
```
postfile buffer meanhat using sampmean, replace
   forvalues i=1/1000 {
   preserve
   sample 150, count
   qui sum age
   post buffer (r(mean))
  restore
}
postclose buffer
```

- **I'm not bootstrapping here**. This is about understanding the standard error

Example

- We can see how the means distribute and what is the standard deviation (standard error)

```
use sampmean, clear
hist meanhat, saving(med.gph, replace)
sum meanhat
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     meanhat |      1,000    25.49305    .8420507   22.69333   28.30667
graph export med.png, replace
```



So what did we learn?

- Even though the data does not distribute normal (it had a Chi-square distribution) the means of the 150 do distribute normal
- The standard error (the standard deviation of the means) is 0.84
- With that information we could do hypothesis testing. For example, we know that 95% of the values are within 2 standard deviations
- If say, I use the first sample of 150 with mean 26.73, the 95 CI is $[26.73 - 2 * 0.84, 26.73 + 2 * 0.84] = [25.05, 28.41]$. We would reject the null that the mean is 29, for example
- (Recall, though, that we use the t-distribution because we have to estimate the standard error)

So what did we learn?

- Of course, we don't do simulations in practice since we can't and know that the **theoretical** SE of the mean is $\frac{\hat{\sigma}}{\sqrt{N}}$, where $\hat{\sigma}$ is the standard deviation of the sample
- For example, we can use just 1 sample to get an approximation:

```
 Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        age |       150    26.73333     9.95673          4         54
di  9.957649 /sqrt(150)
.81303864
```

- Theory gives us a formula for the standard error and a distribution. With simulations, we found that it was 0.84. With theory, we got 0.81

What if we don't have theory?

- What happens when we don't have theory to tell us what is the standard error?
- We collect a sample and have an estimator but we don't know its standard error either because we don't know how to derive the theoretical SE or because there is no formula for it
- We **can't** use simulations because we do not know the true model; we just used simulations to understand the logic behind the theory
- This is when the **bootstrap** is truly like *magic*

Nonparametric bootstrap

- Suppose a new situation (that is slightly more realistic)
- We have a sample of 150 people and we calculate mean age but let's **assume that we do not know the formula** for the standard error of the mean
- How could we come up with an approximation for the standard error using the data?
- Enters the **bootstrap**
- I'll show you how the bootstrap works before we **try** to understand **why it works**

Nonparametric bootstrap

- We won't simulate from any distribution. We will **resample with replacement**. We will resample our sample of 150 observations
- We will use the 150 observations and obtain a sample with replacement so we have another set of 150 observations
- We will take the mean of the 150 observations and save it
- We will repeat this process 3000 times and use the 3000 means to calculate their standard deviation and distribution

Sampling with replacement

- Sampling with replacement can be confusing
- Suppose you have ten numbers: 2, 4, 6, 10, 3, 11, 20, 40, 13,1
- If we sample 10 numbers with replacement, we could get: 2, 4, 4, 4, 11, 1, 20, 6, 6, 2
- In other words, just a combination of the **same** numbers, some of them repeated but most likely **not the same numbers**
- Sampling 10 numbers out of those 10 numbers **without** replacement would imply getting the same exact 10 numbers

Stop here for a bit

- Make sure you understand what is different here from the simulations
- We are not drawing a random sample from a distribution
- We are using our **sample** to take other samples of the **same size**
- It can be hard to understand this distinction and even harder to understand why it works

Example bootstrap

- I saved one sample of 150 in a dataset called s150.dta
- We want to calculate the SE of the mean because we are **pretending we don't know the formula for the standard error**

```
use s150,clear
sum age
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |        150    25.49595    10.70073          1         95

* Theoretical SE
. di 10.70073/sqrt(150)
.87371095
```

- In this sample, the theoretical error is 0.87

Example bootstrap

- Resample from the 150 with replacement to get another sample of size 150
- Again, it's not going to be the same 150 observations, each will be different
- Take the mean and save it; repeat 3000 times

```
postfile buffer meanhat using sampmean_b, replace
forvalues i=1/3000 {
   preserve
   bsample 150
   qui sum age
   post buffer (r(mean))
restore
}
postclose buffer

use sampmean_b, clear
sum
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     meanhat |      3,000     25.4795    .8323291      22.52   28.56667
```

- Our boostrapped SE is .8323291, which is close to theoretical SE. **MAGIC**

Another example

- You don't need to write your own program most of the time
- Stata has a bootstrap command
- We will use the auto dataset

Auto dataset

- Auto dataset

```
 sysuse auto, clear
(1978 Automobile Data)

reg price mpg turn

      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(2, 71)        =     10.08
       Model |   140436412         2  70218206.1   Prob > F        =    0.0001
    Residual |   494628984        71  6966605.41   R-squared       =    0.2211
-------------+----------------------------------   Adj R-squared   =    0.1992
       Total |   635065396        73  8699525.97   Root MSE        =    2639.4

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |  -259.6967   76.84886    -3.38   0.001     -412.929   -106.4645
        turn |  -38.03857   101.0624    -0.38   0.708    -239.5513    163.4742
       _cons |   13204.27   5316.186     2.48   0.015       2604.1    23804.45
------------------------------------------------------------------------------
```

- We do have theory and we do have a formula for the SEs here...

Auto dataset

- Let's bootstrap them anyway

```
 bootstrap, reps(1000): regress price mpg turn
(running regress on estimation sample)

Bootstrap replications (1000)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
...
..................................................  2000

Linear regression                               Number of obs     =         74
                                                Replications      =      2,000
                                                Wald chi2(2)      =      14.53
                                                Prob > chi2       =     0.0007
                                                R-squared         =     0.2211
                                                Adj R-squared     =     0.1992
                                                Root MSE          =  2639.4328

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
       price |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |  -259.6967   104.8474    -2.48   0.013    -465.1939   -54.19961
        turn |  -38.03857   129.3878    -0.29   0.769    -291.6339    215.5568
       _cons |   13204.27   7012.439     1.88   0.060    -539.8537     26948.4
------------------------------------------------------------------------------
```

- **MAGIC!!!**

Not impressed?

- Perhaps you are not too impressed because the SEs are not that close in this example
- But check the data more carefully; the auto dataset has only **74** observations
- What about if we try the same with more data?
- But first, why more data would be better? The way I make sense of it is that with more data we have more realizations or examples of values so it is like we were drawing random samples repeatedly, just like what we did getting at the beginning of the class
- Let's use the beauty dataset

Beauty dataset

- Beauty dataset has more observations

```
reg lwage abvavg exper looks union

      Source |       SS           df       MS      Number of obs   =     1,260
-------------+----------------------------------   F(4, 1255)      =     48.43
       Model |  59.4988269         4  14.8747067   Prob > F        =    0.0000
    Residual |  385.481145     1,255  .307156291   R-squared       =    0.1337
-------------+----------------------------------   Adj R-squared   =    0.1310
       Total |  444.979972     1,259  .353439215   Root MSE        =    .55422

------------------------------------------------------------------------------
       lwage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      abvavg |  -.1600523   .0618286    -2.59   0.010    -.2813512   -.0387534
       exper |   .0152176   .0013286    11.45   0.000      .012611    .0178241
       looks |   .1874543   .0413931     4.53   0.000      .106247    .2686616
       union |   .1986142   .0353455     5.62   0.000     .1292715    .2679569
       _cons |   .7791512   .1212421     6.43   0.000     .5412917    1.017011
------------------------------------------------------------------------------
```

Beauty dataset

- Bootstrap the SEs

```
  bootstrap, reps(2000): reg lwage abvavg exper looks union
(running regress on estimation sample)

Bootstrap replications (2000)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
...
..................................................  2000

Linear regression                               Number of obs     =      1,260
                                                Replications      =      2,000
                                                Wald chi2(4)      =     216.74
                                                Prob > chi2       =     0.0000
                                                R-squared         =     0.1337
                                                Adj R-squared     =     0.1310
                                                Root MSE          =     0.5542

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
       lwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      abvavg |  -.1600523   .0632131    -2.53   0.011    -.2839476    -.036157
       exper |   .0152176   .0013484    11.29   0.000     .0125747    .0178604
       looks |   .1874543   .0429811     4.36   0.000     .1032128    .2716958
       union |   .1986142   .0322598     6.16   0.000     .1353861    .2618423
       _cons |   .7791512   .1255793     6.20   0.000     .5330203    1.025282
------------------------------------------------------------------------------
```

- As I said, **MAGIC!**

When do we use bootstrapped SEs?

- We use them when we don't have theory to guide us
- The classic example: no theoretical SE for the median
- You will use them next semester when doing some versions of instrumental variables and propensity scores
- There are several variants of bootstrap (jackknife, parametric)
- **Why does it work**? Well, because, apparently, resampling from a sample with replacement is like sampling from a population; it works better when the sample itself is not small
- In other words, resampling with replacement from the 150 observations is like sampling from the 40,000 observations (the population)
- **Active area of research**

Median

- As I said, there is no good theoretical formula for the standard error of the median
- So bootstrapping is a good option. I'll do it the longer way, making my own program

```
seed seed 12354
sysuse auto, clear

* Write a command call mymedian
program mymedian, rclass
   version 14.2
   args x
   qui sum `x', det
   return scalar med = r(p50)
end

bootstrap r(med), reps(1000): mymedian price
```

Median

- Output

```
Bootstrap replications (1000)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
...
.................................................. 1000

Bootstrap results                               Number of obs     =         74
                                                Replications      =      1,000

      command:  mymedian price
        _bs_1:  r(med

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _bs_1 |     5006.5   270.0553    18.54   0.000     4477.201    5535.799
------------------------------------------------------------------------------
```

Median - the shortest way

- Using the summarize command directly

```
 bootstrap r(p50), reps(1000): summarize price, detail
(running summarize on estimation sample)
Bootstrap replications (1000)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
......
..................................................  1000

Bootstrap results                               Number of obs     =         74
                                                Replications      =      1,000

      command:  summarize price, detail
        _bs_1:  r(p50)

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _bs_1 |     5006.5   257.0563    19.48   0.000     4502.679    5510.321
------------------------------------------------------------------------------
```

Pesky details

- But what about the distribution of the estimate? We know the SE but we need to know how it distributes as well

```
. estat bootstrap, all

Bootstrap results                               Number of obs     =         74
                                                Replications      =       1000

      command:  summarize price, detail
        _bs_1:  r(p50)

------------------------------------------------------------------------------
             |    Observed               Bootstrap
             |       Coef.       Bias    Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _bs_1 |      5006.5      7.681   257.05629    4502.679   5510.321   (N)
             |                                         4603.5     5708.5   (P)
             |                                           4647       5719  (BC)
------------------------------------------------------------------------------
(N)    normal confidence interval
(P)    percentile confidence interval
(BC)   bias-corrected confidence interval
```

Summary

- Active area of research
- Extremely useful in some situations, although for most applied research we don't need it because there is theory
- The idea is so simple yet so powerful