19: Sampling and the Bootstrap

Jerry Cain

May 10, 2021

Quick slide reference

Sampling definitions

Motivating example

You want to know the true mean and variance of happiness in Bhutan.

- But you can't ask everyone.
- You poll 200 random people.
- Your data looks like this:

Happiness = {72, 85, 79, 91, 68, ..., 71}

- The mean of all these numbers is 83.

Is this the **true mean happiness** of Bhutanese people?

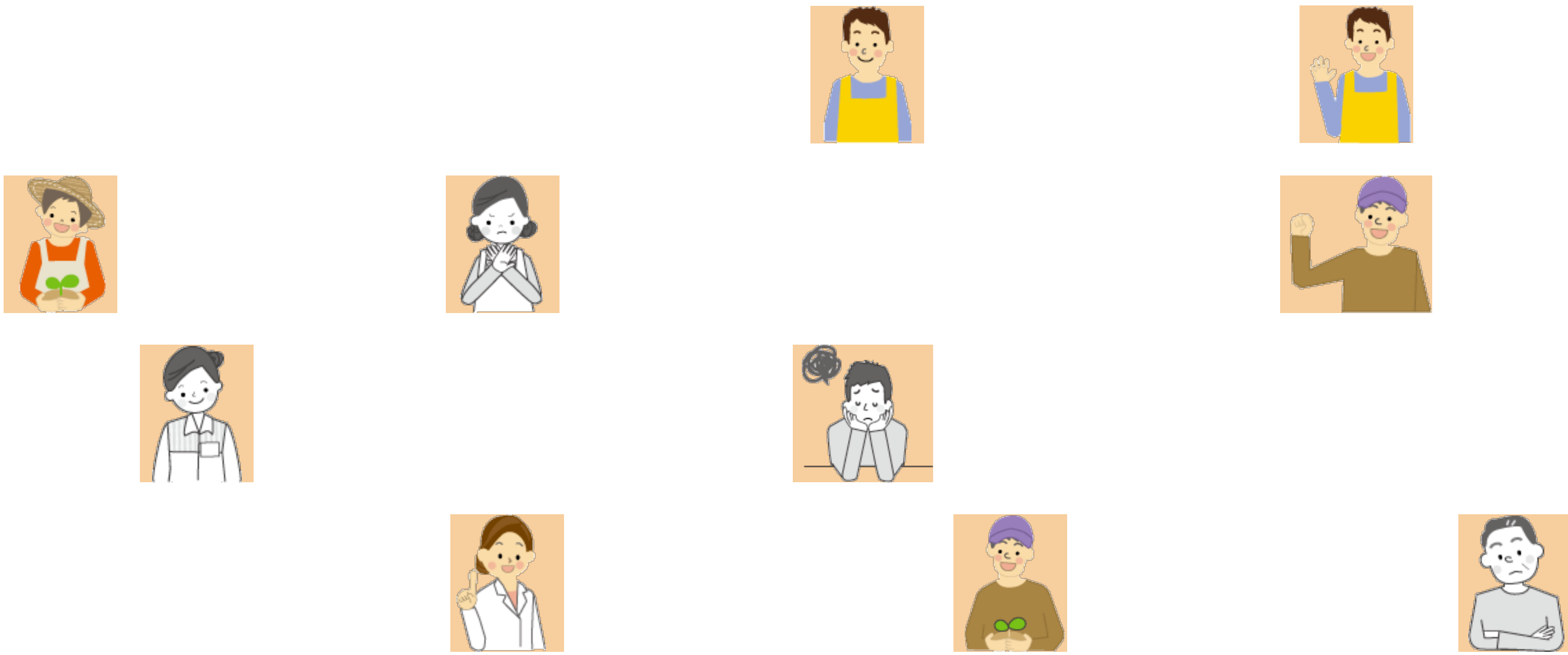
Population

This is a population.

Sample

A sample is selected from a population.

Sample



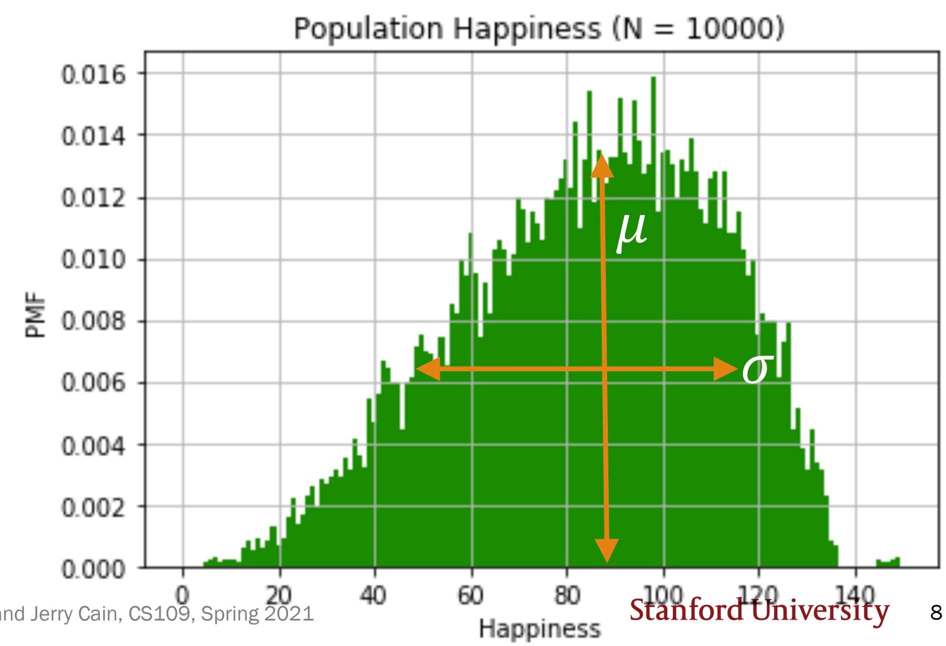
A **sample** is selected from a population.

A sample, mathematically

Consider n random variables $X_1, X_2, \dots, X_n$.

The sequence $X_1, X_2, \dots, X_n$ is a **sample** from distribution F if:

- X_i are all independent and identically distributed (i.i.d.)
- X_i all have same distribution function F (the **underlying distribution**), where $E[X_i] = \mu$, $\text{Var}(X_i) = \sigma^2$



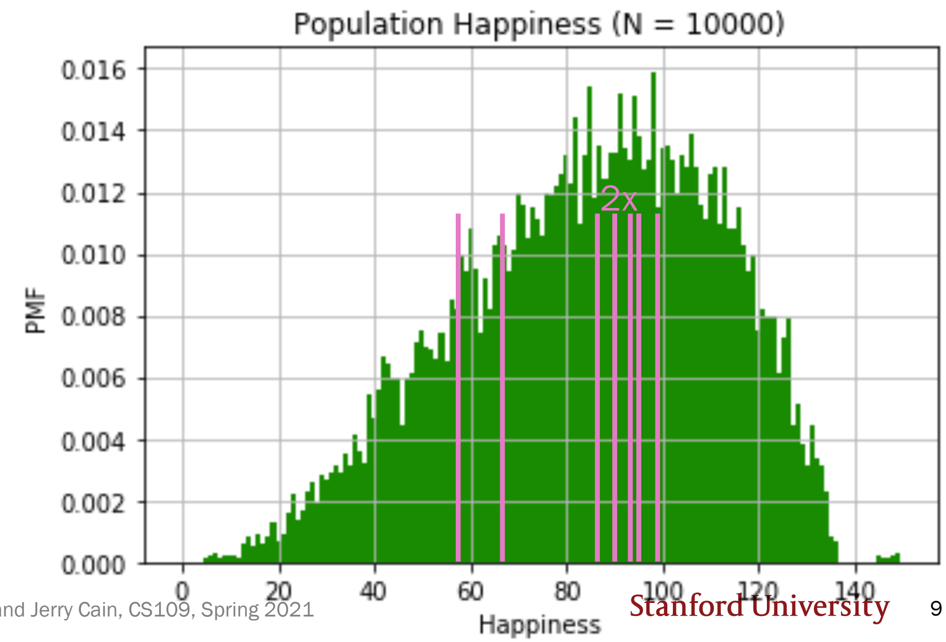
A sample, mathematically

A sample of **sample size** 8:

$$(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

A **realization** of a sample of size 8:

$$(59, 87, 94, 99, 87, 78, 69, 91)$$



A single sample

A happy
Bhutanese person

If we had a distribution F of our entire population, we could compute exact statistics about about happiness.

But we only have 200 people (a sample).

Today: If we only have a single sample,

- How do we report *estimated* statistics?
- How do we report estimated error of these estimates?
- How do we perform hypothesis testing?

Unbiased estimators

A single sample

A happy
Bhutanese person

If we had a distribution F of our entire population, we could compute exact statistics about about happiness.

But we only have 200 people (a sample).

So these population statistics are unknown:

- μ, the population mean
- σ^2, the population variance

A single sample

A happy
Bhutanese person

If we had a distribution F of our entire population, we could compute exact statistics about about happiness.

But we only have 200 people (a sample).

- From these 200 people, what is our best estimate of population mean and population variance?
- How do we define best estimate?

Estimating the population mean

1. What is our best estimate of μ, the **mean happiness** of Bhutanese people?

If we only have a sample, $(X_1, X_2, \dots, X_n)$:

The best estimate of μ is the **sample mean**:

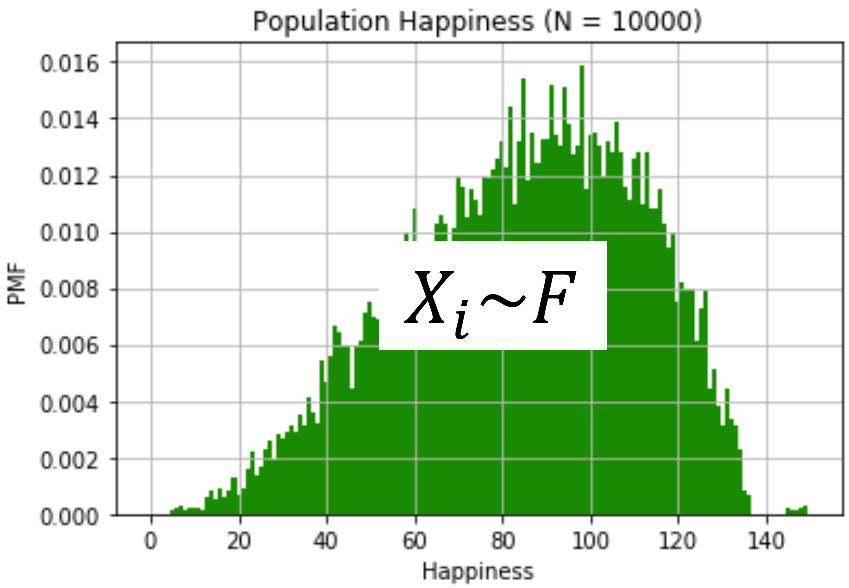
$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$\bar{X}$ is an <u>unbiased estimator</u> of the population mean μ. $E[\bar{X}] = \mu$

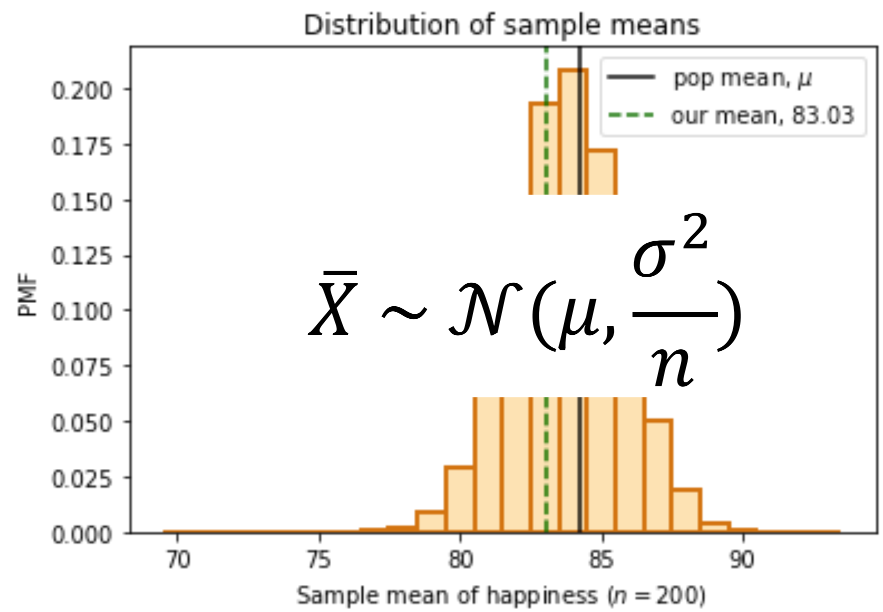
Intuition: By the CLT, $\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma^2}{n})$

If we could take *multiple* samples of size n:
1. For each sample, compute sample mean
2. On average, we would get the population mean

Sample mean



$$X_i \sim F$$



$$\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma^2}{n})$$

Even if we can't report μ, we can report our sample mean 83.03, which is an unbiased estimate of μ.

Estimating the population variance

2. What is σ^2, the variance of happiness of Bhutanese people?

If we knew the entire population $(x_1, x_2, \dots, x_N)$:

population mean

population variance $$\sigma^2 = E[(X-\mu)^2] = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

If we only have a sample, $(X_1, X_2, \dots, X_n)$:

sample mean

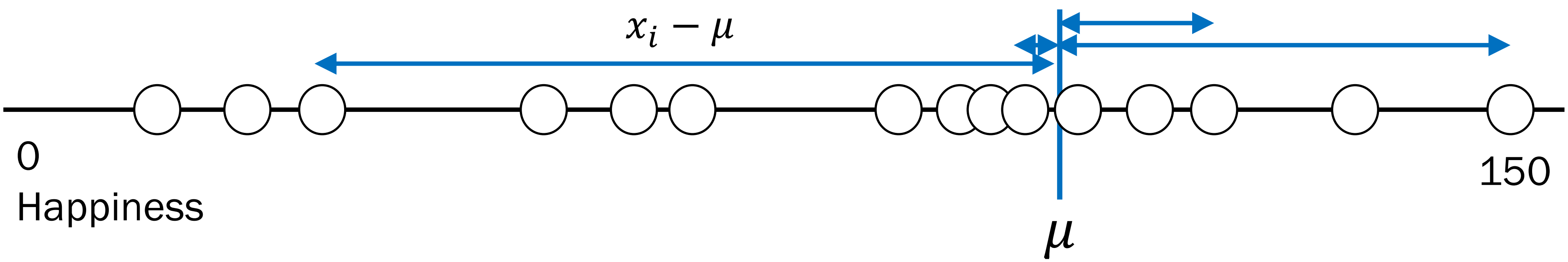
sample variance $$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

Intuition about the sample variance, S^2

<u>Actual, σ^2</u>

population mean

population variance $\quad \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$



Population size, N

Calculating population statistics **<u>exactly</u>** requires us knowing all N datapoints.

Intuition about the sample variance, S^2

Actual, σ^2

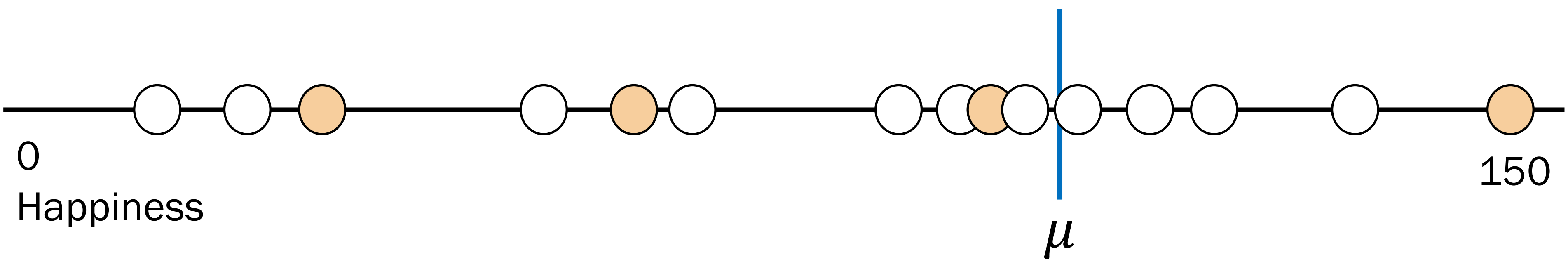
population mean

population variance $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$

Estimate, S^2

sample mean

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



Population size, N

Intuition about the sample variance, S^2

Actual, σ^2

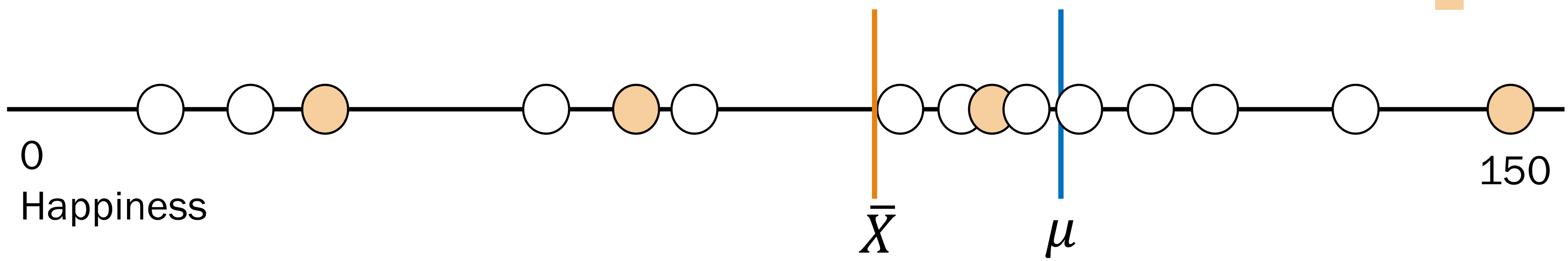
population mean

population variance $$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Estimate, S^2

sample mean

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



Population size, N

Intuition about the sample variance, S^2

Actual, σ^2

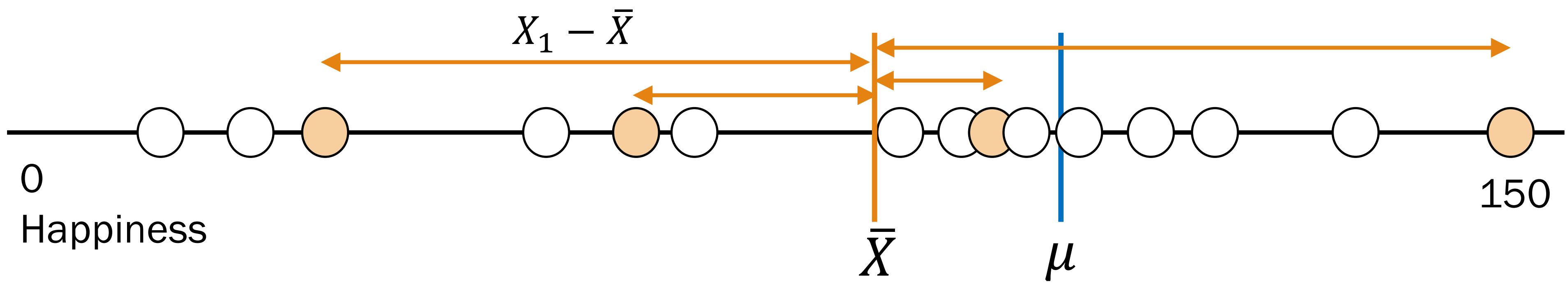
population mean

population variance $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$

Estimate, S^2

sample mean

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



Population size, N

Sample variance is an estimate using an estimate, so it needs additional scaling.

Estimating the population variance

2. What is σ^2, the variance of happiness of Bhutanese people?

If we only have a sample, $(X_1, X_2, \ldots, X_n)$:

The best estimate of σ^2 is the **sample variance:**

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

S^2 is an **unbiased estimator** of the population variance, σ^2.

$$E[S^2] = \sigma^2$$

Proof that S^2 is unbiased (just for reference)

$E[S^2] = \sigma^2$

$$E[S^2] = E\left[\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2\right] \quad \Rightarrow \quad (n-1)E[S^2] = E\left[\sum_{i=1}^{n}(X_i - \bar{X})^2\right]$$

$$(n-1)E[S^2] = E\left[\sum_{i=1}^{n}\left((X_i - \mu) + (\mu - \bar{X})\right)^2\right]$$

(introduce $\mu - \mu$)

$$= E\left[\sum_{i=1}^{n}(X_i - \mu)^2 + \sum_{i=1}^{n}(\mu - \bar{X})^2 + 2\sum_{i=1}^{n}(X_i - \mu)(\mu - \bar{X})\right]$$

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

$$2(\mu - \bar{X})\left(\sum_{i=1}^{n}X_i - n\mu\right)$$

$$2(\mu - \bar{X})n(\bar{X} - \mu)$$

$$-2n(\mu - \bar{X})^2$$

$$= E\left[\sum_{i=1}^{n}(X_i - \mu)^2 + n(\mu - \bar{X})^2 - 2n(\mu - \bar{X})^2\right]$$

$$= E\left[\sum_{i=1}^{n}(X_i - \mu)^2 - n(\mu - \bar{X})^2\right] = \sum_{i=1}^{n}E[(X_i - \mu)]^2 - nE[(\bar{X} - \mu)^2]$$

$$= n\sigma^2 - n\mathrm{Var}(\bar{X}) = n\sigma^2 - n\frac{\sigma^2}{n} = n\sigma^2 - \sigma^2 = (n-1)\sigma^2$$

Therefore $E[S^2] = \sigma^2$

19c_standard_error

Standard error

Estimating population statistics

A particular outcome

1. Collect a sample, $X_1, X_2, \ldots, X_n$. $(72, 85,79,79,91,68, \ldots, 71)$
   $n = 200$

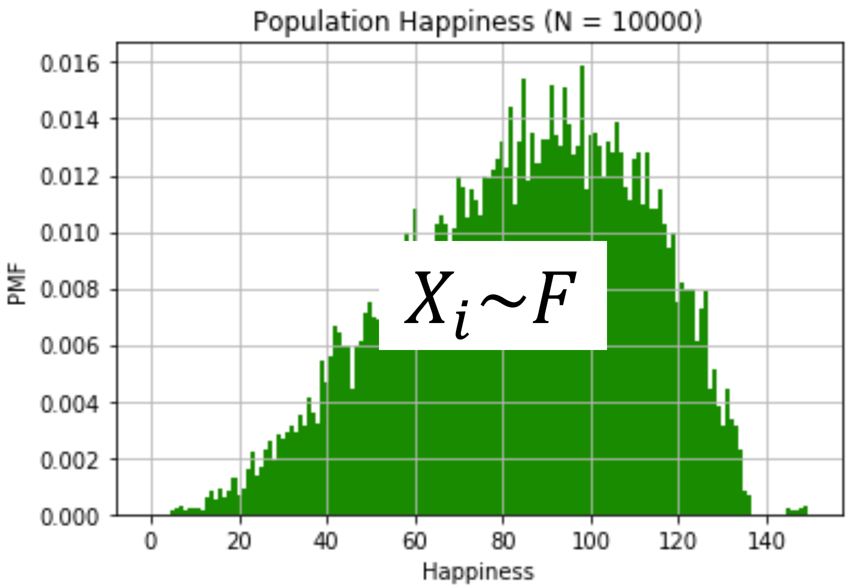
2. Compute **sample mean**, $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. $\bar{X} = 83$

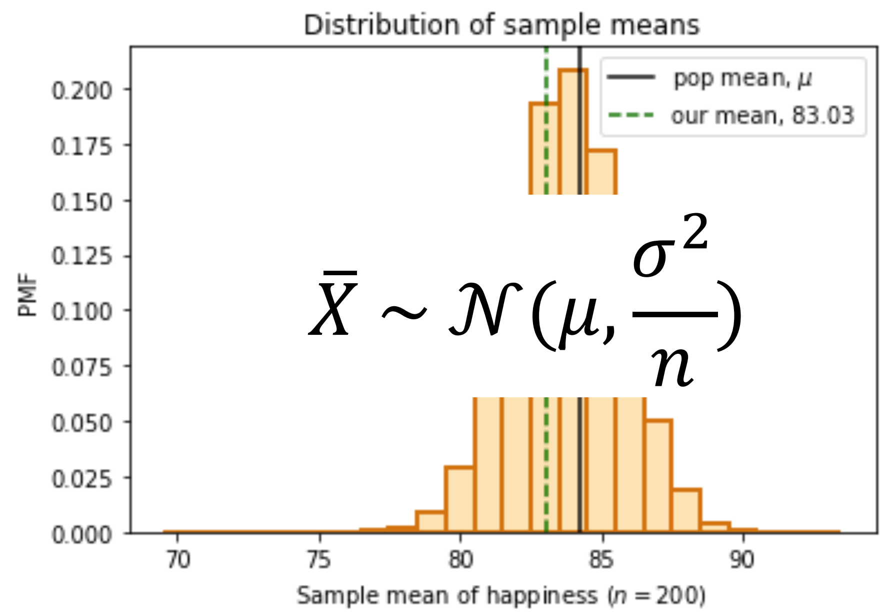
3. Compute sample deviation, $X_i - \bar{X}$. $(-11, 2, -4, -4,8, -15, \ldots, -12)$

4. Compute **sample variance**, $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$. $S^2 = 793$

How “close” are our estimates $\bar{X}$ and S^2?

Sample mean





- $\mathrm{Var}(\bar{X})$ is a measure of how "close" $\bar{X}$ is to μ.
- **How do we estimate** $\mathrm{Var}(\bar{X})$**?**

How “close” is our estimate $\bar{X}$ to μ?

$$E[\bar{X}] = \mu$$

$$\boxed{\text{Var}(\bar{X}) = \frac{\sigma^2}{n}}$$

We want to estimate this

def The **standard error** of the mean is an estimate of the standard deviation of $\bar{X}$.

$$SE = \sqrt{\frac{S^2}{n}}$$

Intuition:

- S^2 is an unbiased estimate of σ^2
- S^2/n is an unbiased estimate of $\sigma^2/n = \text{Var}(\bar{X})$
- $\sqrt{S^2/n}$ can estimate $\sqrt{\text{Var}(\bar{X})}$

More info on bias of standard error: [wikipedia]

Standard error of the mean

1. Mean happiness:

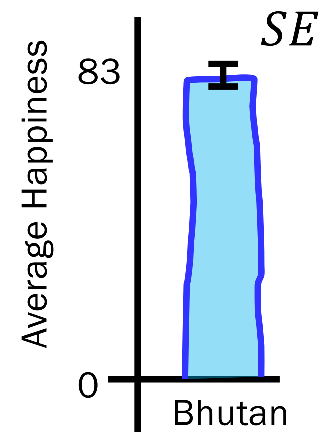
Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

this is our best estimate of μ

this is our estimate of how "close" we are

Closed form: $SE = \sqrt{\dfrac{S^2}{n}}$

error bars



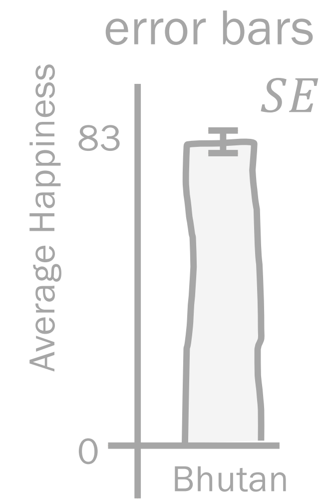
These 2 statistics give a sense of how the sample mean random variable $\bar{X}$ behaves.

Standard error of variance?

1. Mean happiness:

   Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

   Closed form: $SE = \sqrt{\dfrac{S^2}{n}}$



2. Variance of happiness:

   this is our best estimate of σ^2

   Claim: The variance of happiness of Bhutan is 793.

   Closed form: Not covered in CS109

   But how close are we?

Up next: Compute Statistics with code!

19d_bootstrap_mean

Bootstrap: Sample mean

Bootstrap

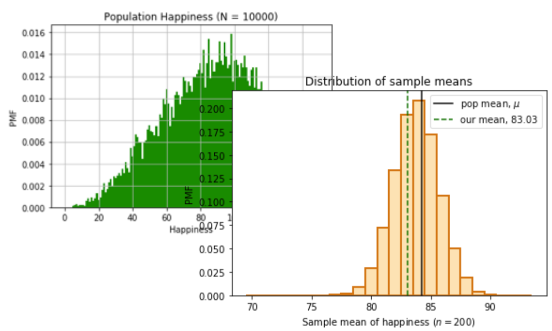
The Bootstrap:

Probability for Computer Scientists

Computing statistic of sample mean

What is the standard deviation of the sample mean $\bar{X}$? (sample size $n = 200$)

Population distribution
(we don't have this)



$$\frac{\sigma}{\sqrt{n}} = 1.886$$

Exact statistic
(we don't have this)

`1.869`

Simulated statistic
(we don't have this)

Sample distribution
(we do have this)

$$SE = \frac{S}{\sqrt{n}} = 1.992$$

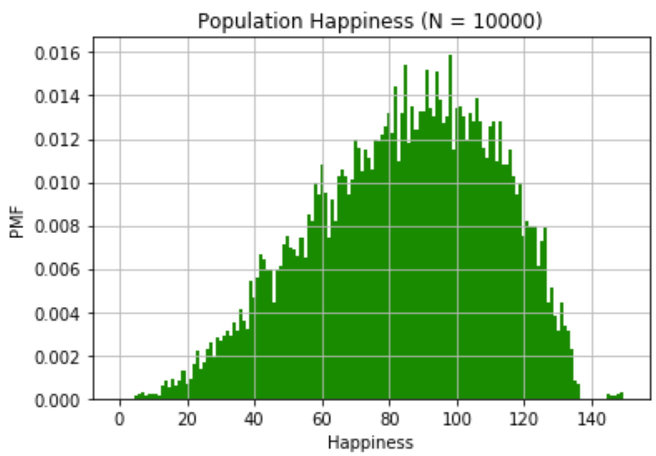
Estimated statistic, by formula, **standard error**

`???`

Simulated estimated statistic

Note: We don't have access to the population.
But Lisa is sharing the exact statistic with you.

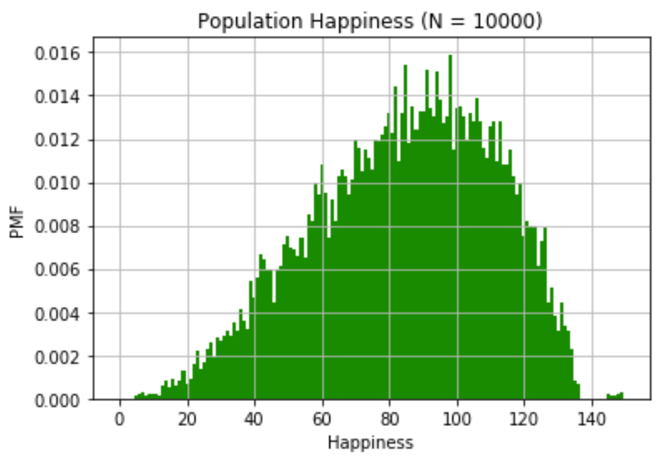
Bootstrap insight 1: Estimate the true distribution



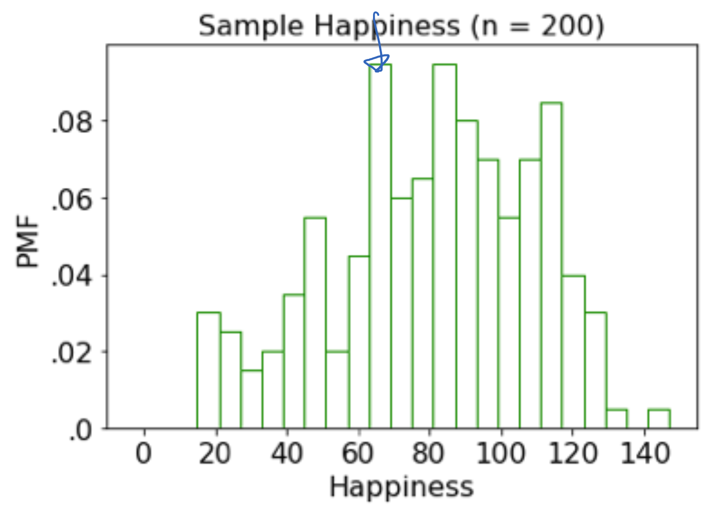
$\approx$

Bootstrap insight 1: Estimate the true distribution

You can estimate the PMF of the underlying distribution, using your sample.*



$\approx$



The underlying distribution

$$F \approx \hat{F}$$

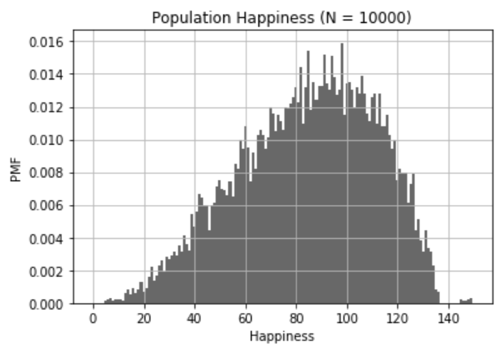
the sample distribution (aka the histogram of your data)

*This is just a histogram of your data!

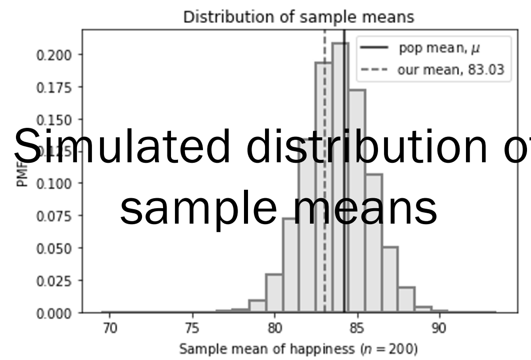
Bootstrap insight 2: Simulate a distribution

Approximate the procedure of simulating a distribution of a statistic, e.g., $\bar{X}$.

Population distribution
(we don't have this)



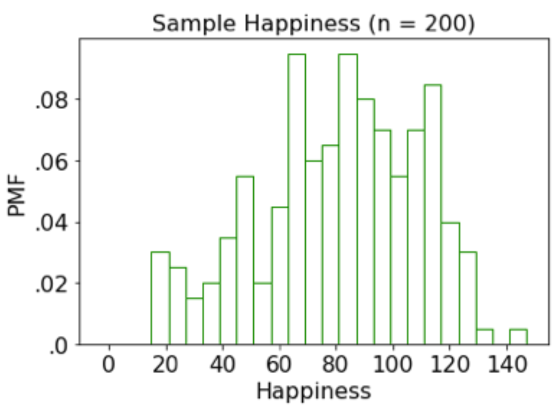
Distribution of $\bar{X}$



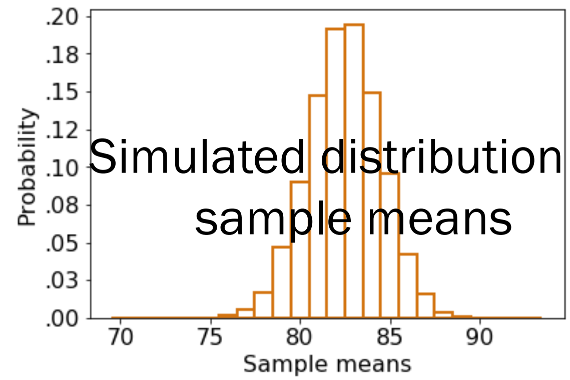
Simulated distribution of sample means

$\approx$

Sample distribution
(we do have this)

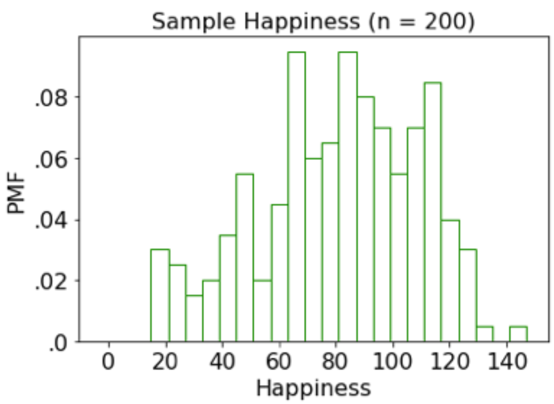


Bootstrap means



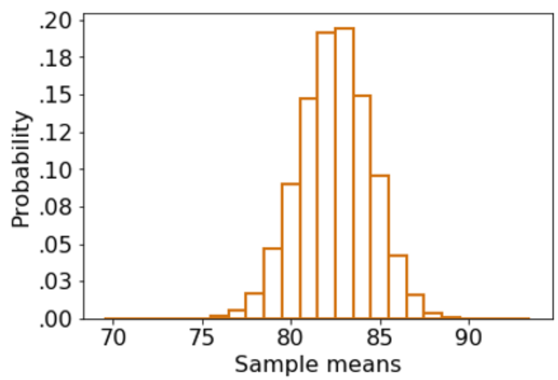
Simulated distribution of sample means

Bootstrapped sample means



Estimate the true PMF using our "PMF" (histogram) of our sample.

```
means = [84.7,
83.9, 80.6, 79.8,
90.3, …, 85.2]
```



...generate a whole bunch of sample means of this estimated distribution...

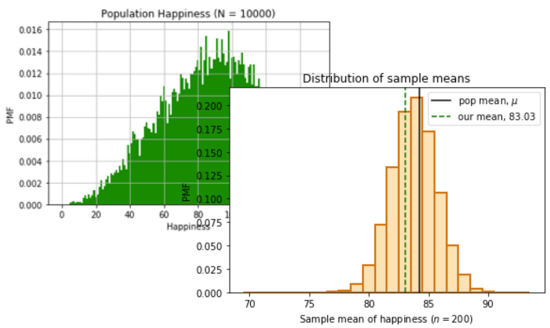
```
np.std(means)
2.003
```

...and compute the standard deviation of this distribution.

Computing statistic of sample mean

What is the standard deviation of the sample mean $\bar{X}$? (sample size $n = 200$)

Population distribution
(we don't have this)



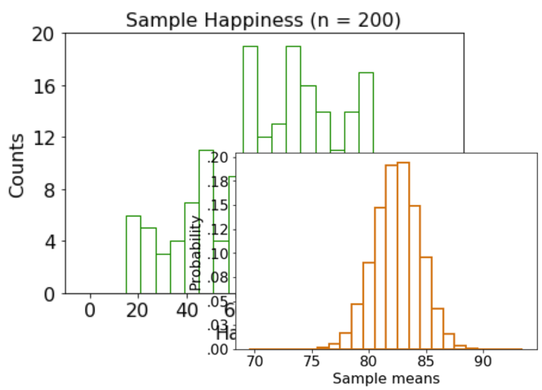
$$\frac{\sigma}{\sqrt{n}} = 1.886$$

Exact statistic
(we don't have this)

`1.869`

Simulated statistic
(we don't have this)

Sample distribution
(we do have this)



$$SE = \frac{S}{\sqrt{n}} = 1.992$$

Estimated statistic, by formula, **standard error**

`2.003`

Simulated estimated statistic, **bootstrapped standard error**

Bootstrap algorithm

```
Bootstrap Algorithm (sample):
1. Estimate the PMF using the sample
2. Repeat 10,000 times:
     a. Resample sample.size() from PMF
     b. Recalculate the sample mean on the resample
3. You now have a distribution of your sample mean
```

What is the distribution of your sample mean?

We'll talk about this algorithm in detail during live lecture!

Bootstrap algorithm

Bootstrap Algorithm (sample):

1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **statistic** on the resample
3. You now have a **distribution of your statistic**

What is the distribution of your statistic?

Bootstrapped sample variance

Bootstrap Algorithm (sample):

1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the sample variance on the resample
3. You now have a **distribution of your sample variance**

What is the distribution of your sample variance?

Even if we don't have a closed form equation,
we estimate statistics of sample variance with bootstrapping!

(live)
19: Sampling and the Bootstrap

Jerry Cain

May 10, 2021

Think

Slide 42 has a question to go over by yourself.

Post any clarifications here or in Zoom chat!

https://edstem.org/us/courses/5090/discussion/428950

Think by yourself: 2 min

Quick check

unknown value

1. μ, the population mean B
2. $(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$, a sample A (8-D)
3. σ^2, the population variance ??? B
4. $\bar{X}$, the sample mean A_{RV} $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$
5. $\bar{X} = 83$ C
6. $(X_1 = 59, X_2 = 87, X_3 = 94, X_4 = 99,$
   $X_5 = 87, X_6 = 78, X_7 = 69, X_8 = 91)$ C

A. Random variable(s)
B. Value
C. Event

Quick check

1. μ, the population mean

2. $(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$, a sample

3. σ^2, the population variance

4. $\bar{X}$, the sample mean

5. $\bar{X} = 83$

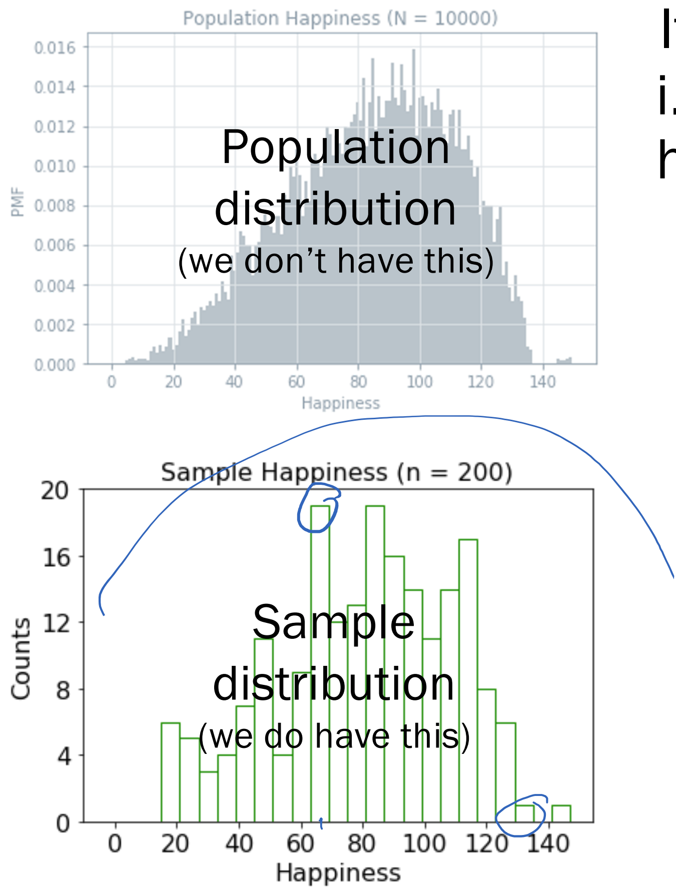
6. $(X_1 = 59, X_2 = 87, X_3 = 94, X_4 = 99,$
   $X_5 = 87, X_6 = 78, X_7 = 69, X_8 = 91)$

A. Random variable(s)
B. Value
C. Event

These are outcomes from your collected data.

Today: Crash course on (bootstrapped) statistics

Review



If we only have a single sample of RVs generated i.i.d. from the same unknown distribution, how can we perform statistical analysis?

- What is the probability that a Bhutanese peep is just straight up loving life?
- What is a good estimate of the population mean (and how "close" is the estimate)?
- What is a good estimate of the population variance (and how "close" is the estimate)?

Standard error

1. Mean happiness:

Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

this is our best estimate of μ

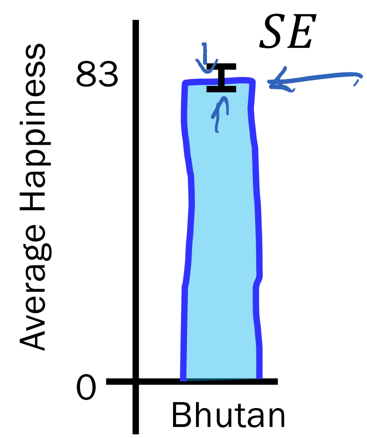
$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Closed form: $SE = \sqrt{\frac{S^2}{n}}$

this is how close we are

✅ Verified via bootstrap:

```
np.std(means)
  = 2.003
```



Standard error

1. Mean happiness:

   Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

   Closed form: $SE = \sqrt{\frac{S^2}{n}}$

2. Variance of happiness:

   this is our best estimate of σ^2

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

   Claim: The variance of happiness of Bhutan is 793.

   Closed form: Not covered in CS109

   But how close are we?

We can bootstrap for standard error of sample variance—a statistic of a statistic.

Bootstrap

Review

The Bootstrap:

Probability for Computer Scientists

Allows you to do the following:

- Calculate distributions over statistics
- Calculate p values null hypotheses

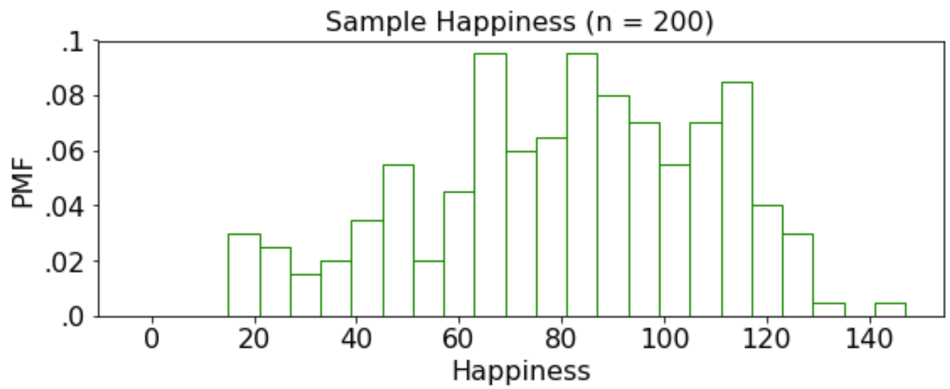
Bootstrapped sample variance

Bootstrap Algorithm (sample): $n=200$

1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the sample variance on the resample
3. You now have a **distribution of your sample variance**

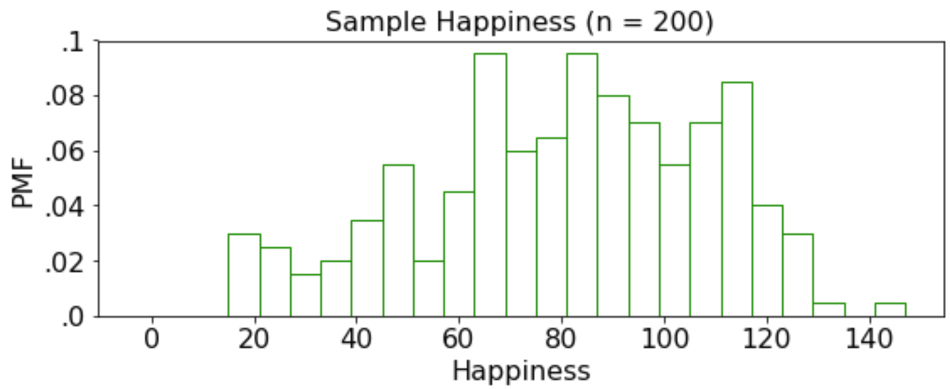
Goal What is the distribution of your sample variance?

Bootstrapped variance



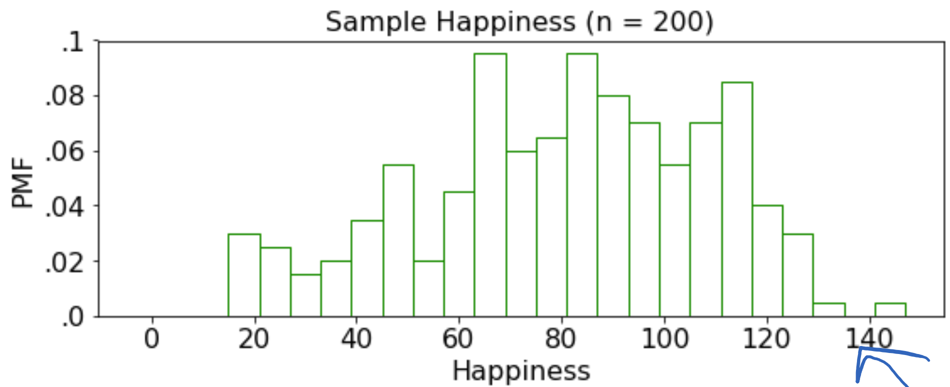
1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

Bootstrapped variance

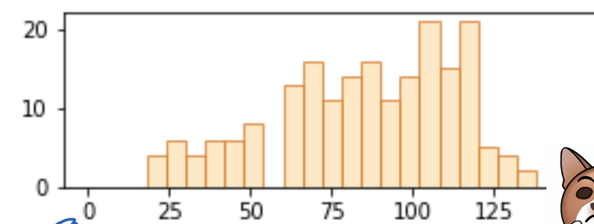


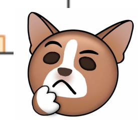
1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

Bootstrapped variance



[52, 38, 98, 107, ..., 94]



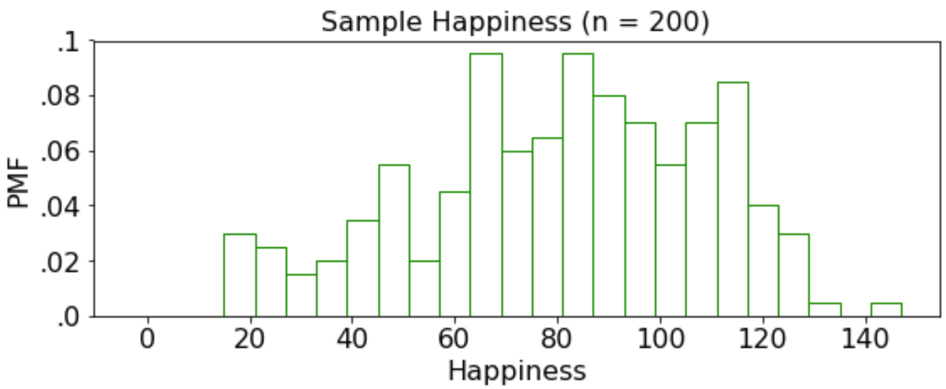


1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your**

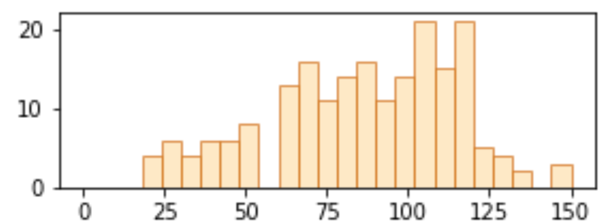
Why are these samples **different?**

This resampled sample is generated **with replacement.**

Bootstrapped variance



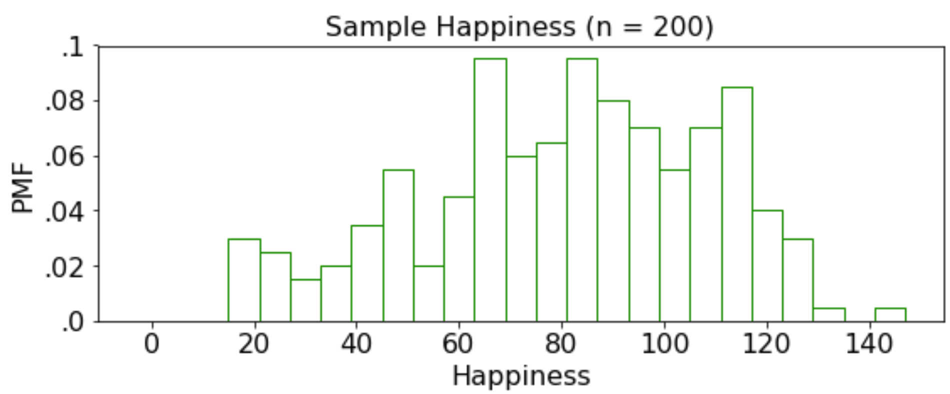
[52, 38, 98, 107, ..., 94]



1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

```
variances = [827.4]
```

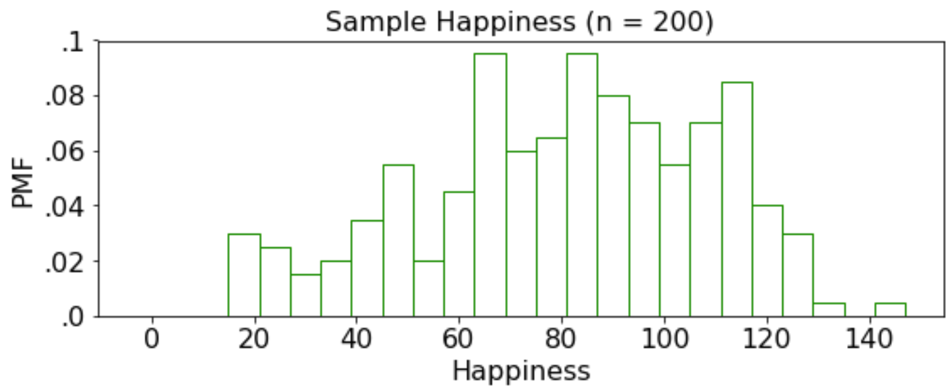
Bootstrapped variance



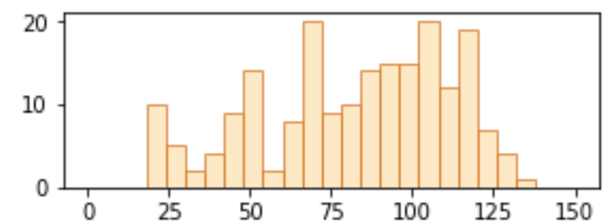
1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

```
variances = [827.4]
```

Bootstrapped variance



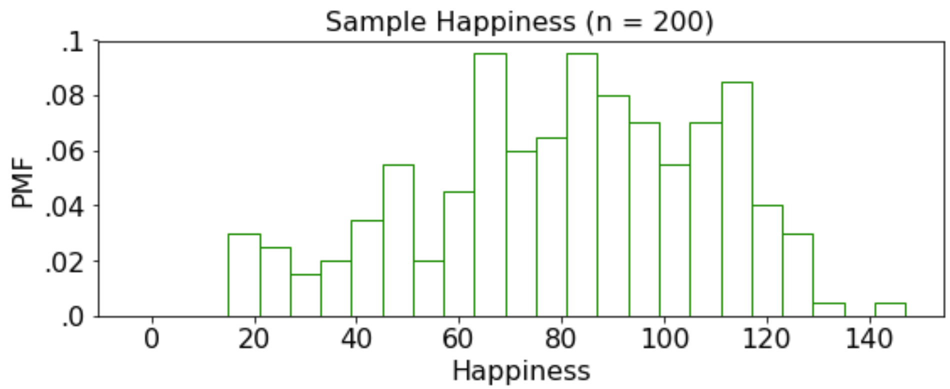
[116, 76, 132, 85, ..., 78]

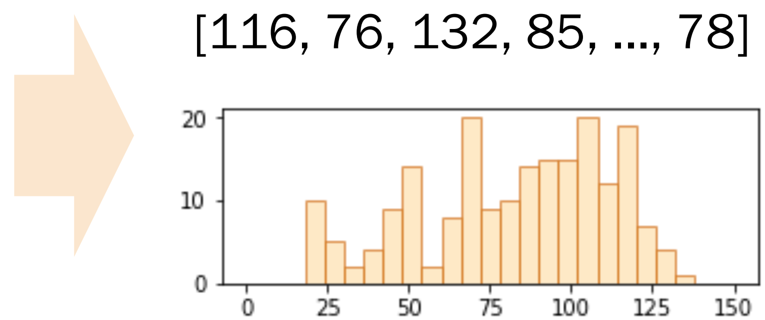


1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

```
variances = [827.4]
```

Bootstrapped variance

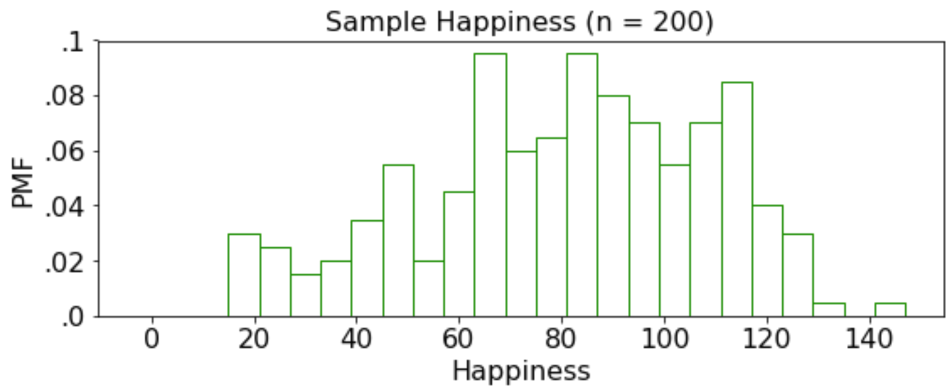




1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Resample **sample.size**() from PMF
   b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

```
variances = [827.4, 846.1]
```

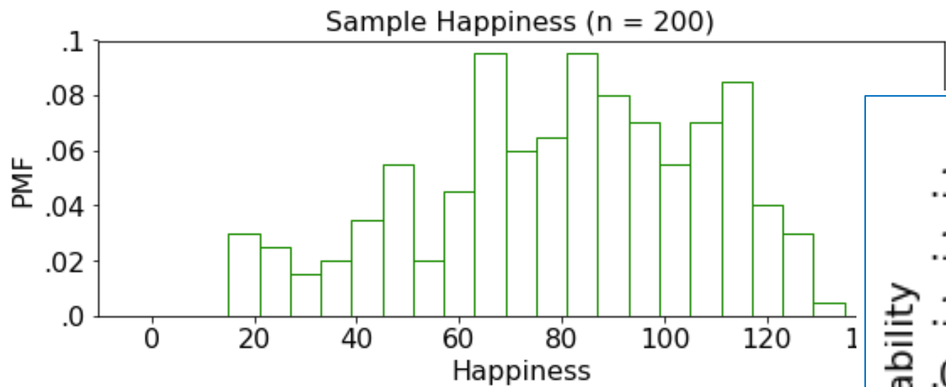
Bootstrapped variance

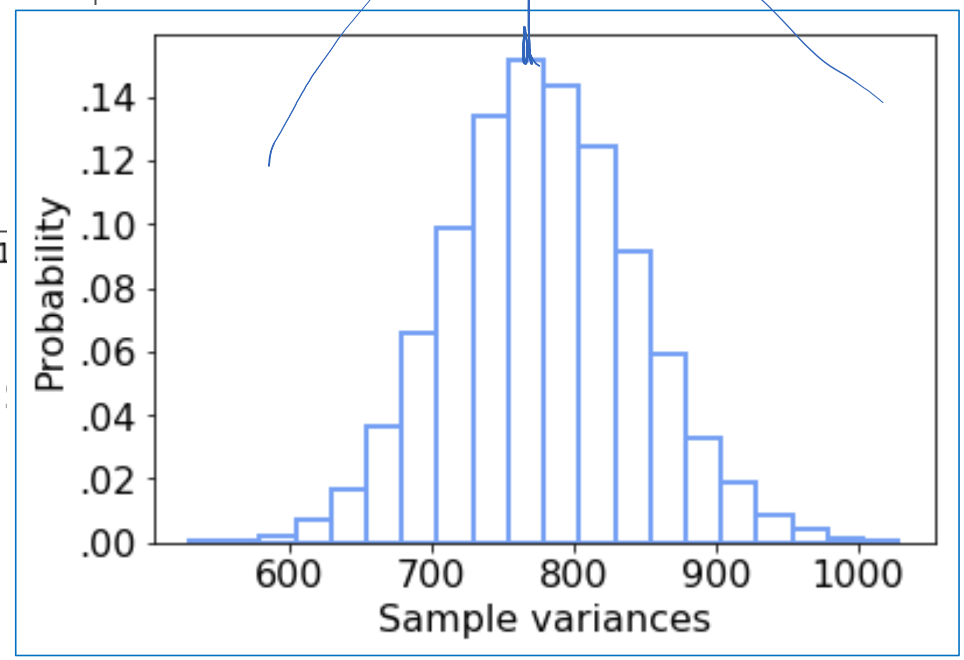


1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
    a. Resample **sample.size**() from PMF
    b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

```
variances = [827.4, 846.1]
```

Bootstrapped variance





1. Estimate the **PMF** using the

2. Repeat **10,000** times:

    a. Resample **sample.size**()

    b. Recalculate the **sample**

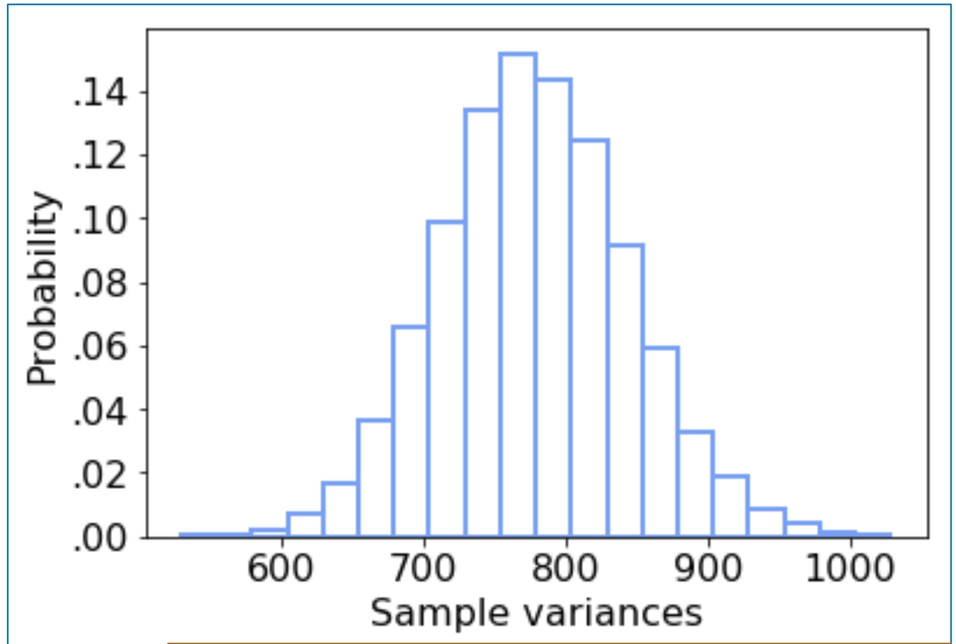
3. You now have a **distribution of your sample variance**

variances = [827.4, 846.1, 726.0, …, 860.7]

Bootstrapped variance

3. You now have a **distribution of your** sample variance

```
variances = [827.4,
     846.1, 726.0, …,
     860.7]
```



What is the bootstrapped standard error?

```
np.std(variances)
```

Bootstrapped standard error: 66.16

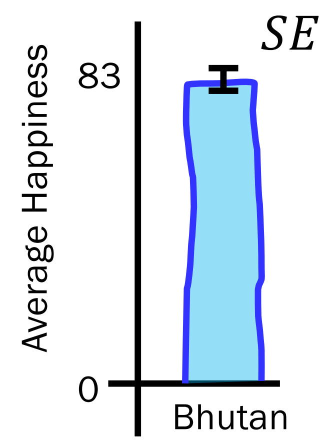
- Simulate a distribution of sample variances
- Compute standard deviation

Standard error

1. Mean happiness:

   Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

   Closed form: $SE = \sqrt{\frac{S^2}{n}}$



2. Variance of happiness:

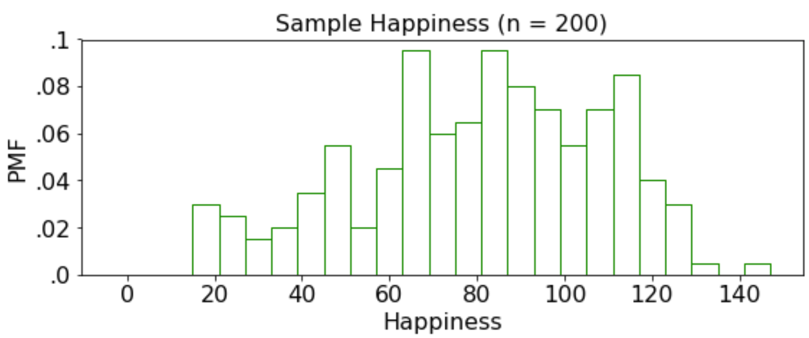
   S^2 is our best estimate of σ^2

   Claim: The variance of happiness of Bhutan is 793, with a **bootstrapped standard error of 66.16**.

   this is how close we are, calculated by bootstrapping

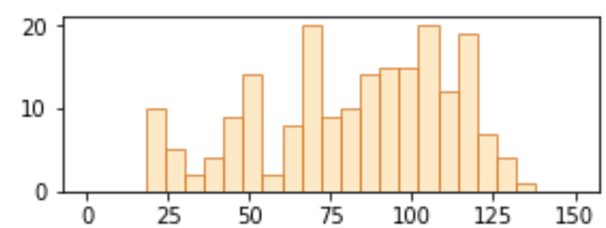
Algorithm in practice: Resampling

1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
    a. Resample **sample.size**() from PMF
    b. Recalculate the **statistic** on the resample
3. You now have a **distribution of your statistic**



?

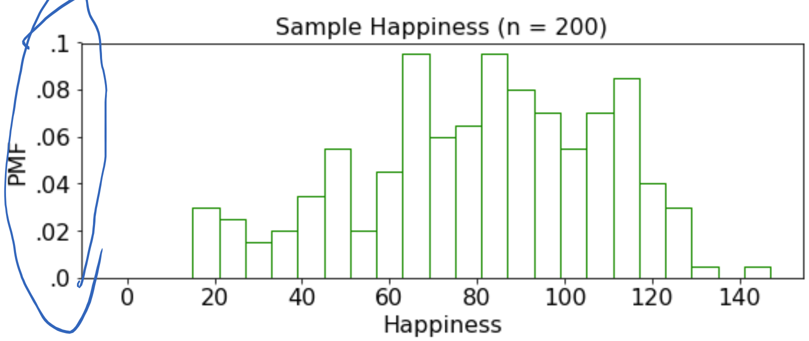
[116, 76, 132, 85, ..., 78]



$$P(X = k) = \frac{\#\text{ values in sample equal to } k}{n}$$

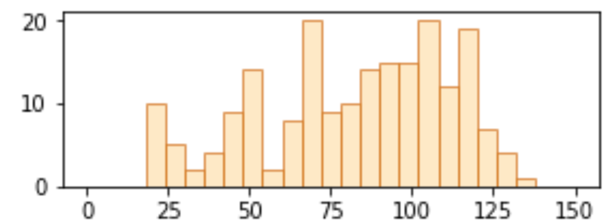
Algorithm in practice: Resampling

```python
def resample(sample, n):
    # estimate the PMF using the sample
    # draw n new samples from the PMF
    return np.random.choice(sample, n, replace=True)
```



?

[116, 76, 132, 85, ..., 78]



$$P(X = k) = \frac{\#\text{ values in sample equal to } k}{n}$$

This resampled sample is generated **with replacement.**

To the code!

Bootstrap provides a way to calculate probabilities of statistics using code.

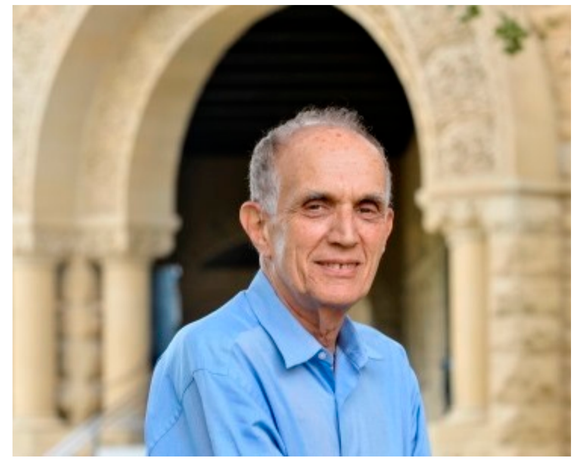
Bootstrapping works for any statistic*

*as long as your sample is i.i.d. and the underlying distribution does not have a long tail

Google colab notebook link
(we will use this in Breakout rooms)

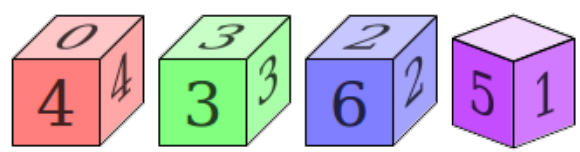
Bradley Efron

- Invented bootstrapping in 1979
- Still a professor at Stanford
- Won a National Science Medal



Efron's dice: 4 dice A, B, C, D such that

$$P(A > B) = P(B > C) = P(C > D) = P(D > A) = \frac{2}{3}$$



Interlude for announcements

Announcements

Problem Set 5

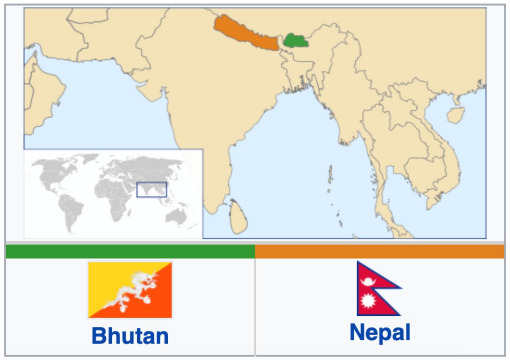
Out:	now
Due:	Friday 5/21 10:00am
Covers:	Up to and including today

Quiz #2

Time frame:	This Wednesday 5/12 11:00am – Friday 5/14 10:00am PT
Covers:	Up to end of Week 5 (including Lecture 15). PS3+PS4
Emma's Review session:	Tonight at 7pm PT (and will be recorded)
Info and practice:	http://web.stanford.edu/class/cs109/quizzes/

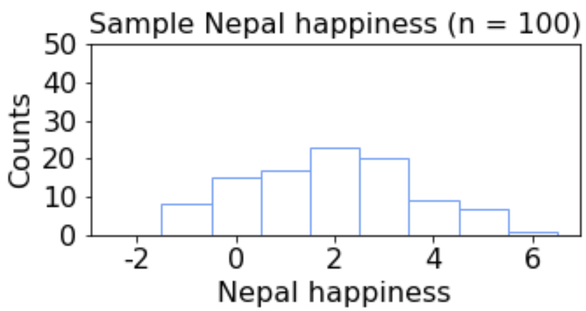
LIVE

Bootstrap: p-value

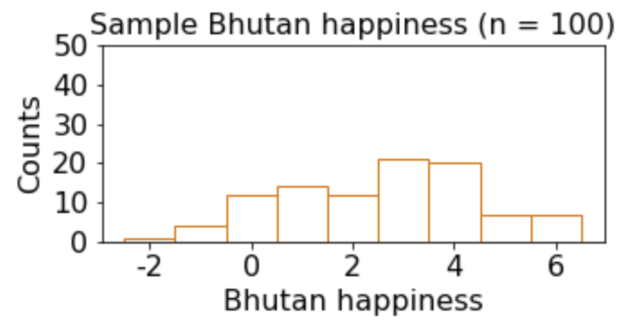


Null hypothesis test

Nepal Happiness
4.45
2.45
6.37
2.07
…
1.63



Bhutan Happiness
0.91
0.34
1.91
1.61
…
1.08



$\bar{X}_1 = 3.1$

$\bar{X}_2 = 2.4$

Claim: The difference in mean happiness between Nepal and Bhutan is 0.7 happiness points, and **this is significant**.

Null hypothesis test

def **null hypothesis** – Even if there is no pattern (i.e., the two samples are from identical distributions), your claim might have arisen by chance.

def **p-value** – What is the probability that the observed difference occurs under the null hypothesis?

Example:

- Flip some coin 100 times.
- Flip the same coin another 150 times.

Null hypothesis assumes we use the same coin

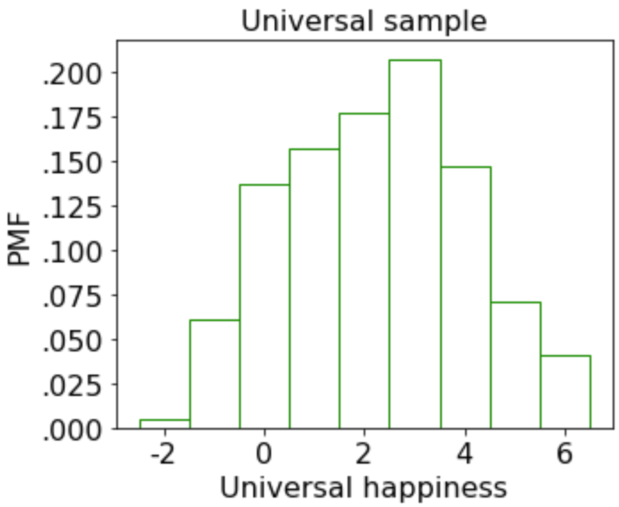
- Compute fraction of heads in both groups.
- There is a possibility we'll see the observed difference in these fractions even if we used the same coin

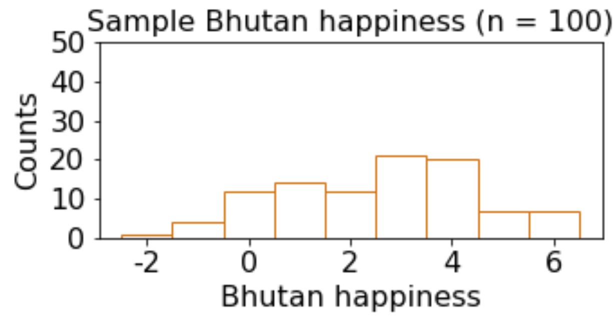
p-value

A **significant** p-value (< 0.05) means we reject the null hypothesis.

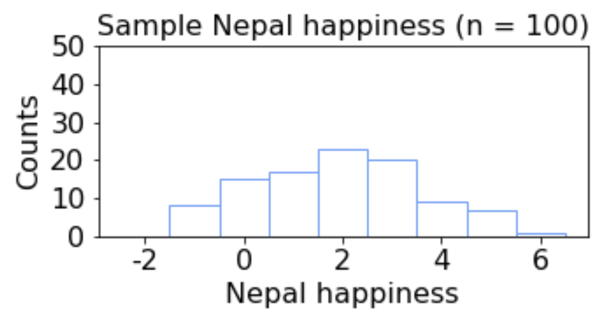
Universal sample

(this is what the null hypothesis assumes)





$\bar{X}_1 = 3.1$



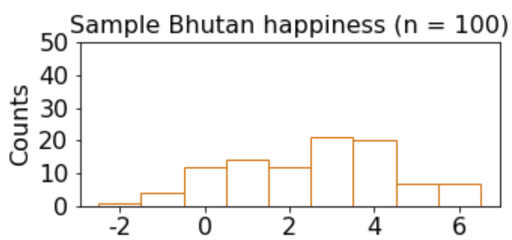
$\bar{X}_2 = 2.4$

Want p-value: probability $|\bar{X}_1 - \bar{X}_2| \overset{\geq}{\cancel{=}} |3.1 - 2.4|$ happens under null hypothesis

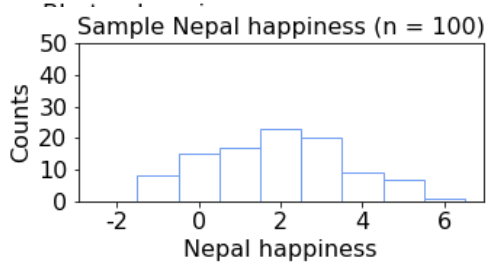
Bootstrap for p-values

1. Create a **universal sample** using your two samples

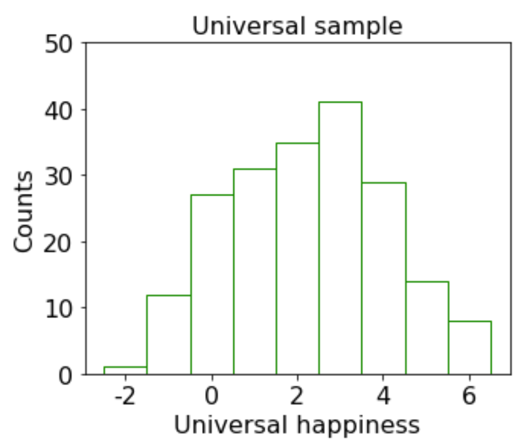
i.e., recreate the null hypothesis



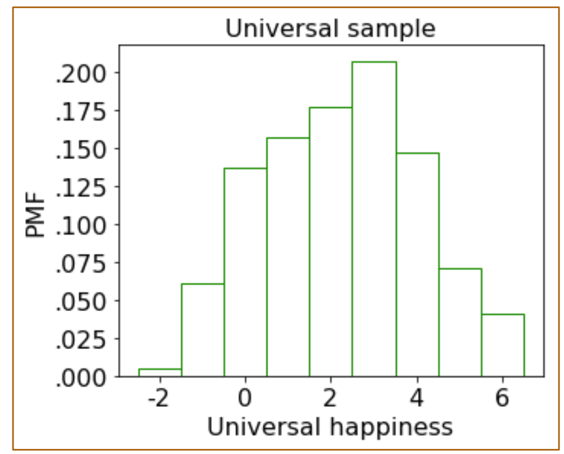
+



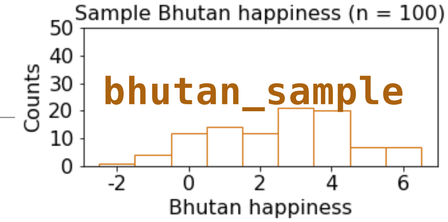
=

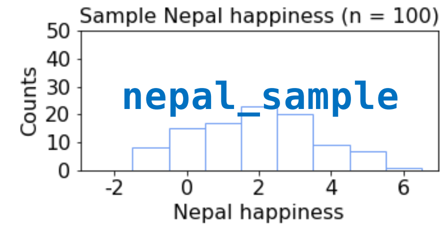


≈



Bootstrap for p-values





1. Create a **universal sample** using your two samples
2. Repeat **10,000** times:
   a. Resample **both samples**
   b. Recalculate the **mean difference** between the resamples
3. **p-value** = $\frac{\text{\# (mean diffs >= observed diff)}}{n}$

Probability that observed difference arose by chance

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal – muBhutan|
        if diff >= observed_diff:
            count += 1

pValue = count / 10,000
```

0.7

Bootstrap for p-values

1. Create a universal sample using your two samples

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal – muBhutan|
        if diff >= observed_diff:
            count += 1

pValue = count / 10,000
```

Bootstrap for p-values

2. a. Resample **both samples**

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal – muBhutan|
        if diff >= observed_diff:
            count += 1

pValue = count / 10,000
```

Bootstrap for p-values

2. b. Recalculate the **mean difference** b/t resamples

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal – muBhutan|
        if diff >= observed_diff:
            count += 1

pValue = count / 10,000
```

Bootstrap for p-values

3. **p-value** = $\frac{\text{\# (mean diffs > observed diff)}}{n}$

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal – muBhutan|
        if diff >= observed_diff:
            count += 1

    pValue = count / 10,000
```

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal – muBhutan|
        if diff >= observed_diff:
            count += 1

pValue = count / 10,000
```

with replacement!

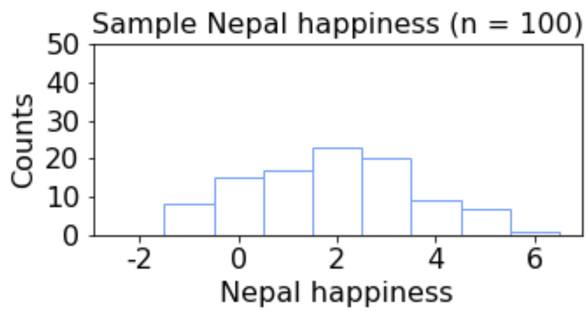
Bootstrap

Let’s try it!

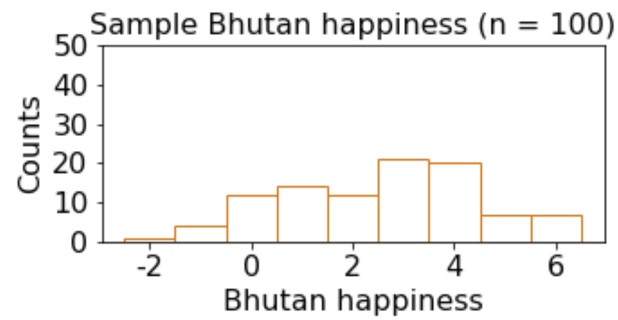
Google colab notebook link

Null hypothesis test

Nepal Happiness
4.45
2.45
6.37
2.07
…
1.63



Bhutan Happiness
0.91
0.34
1.91
1.61
…
1.08



$\bar{X}_1 = 3.1$

$\bar{X}_2 = 2.4$

Claim: The happiness of Nepal and Bhutan have a 0.7 difference of means, and this is significant ($p < 0.05$).