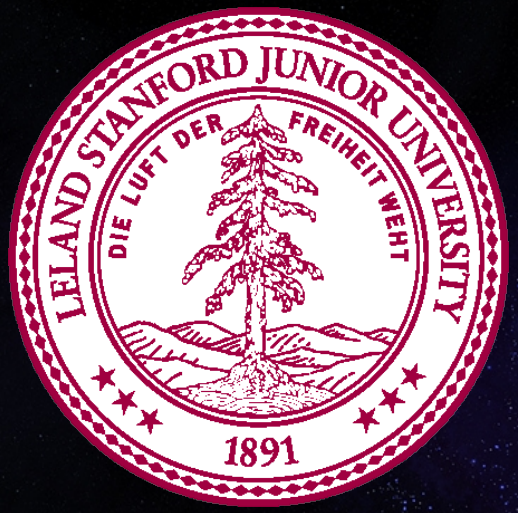
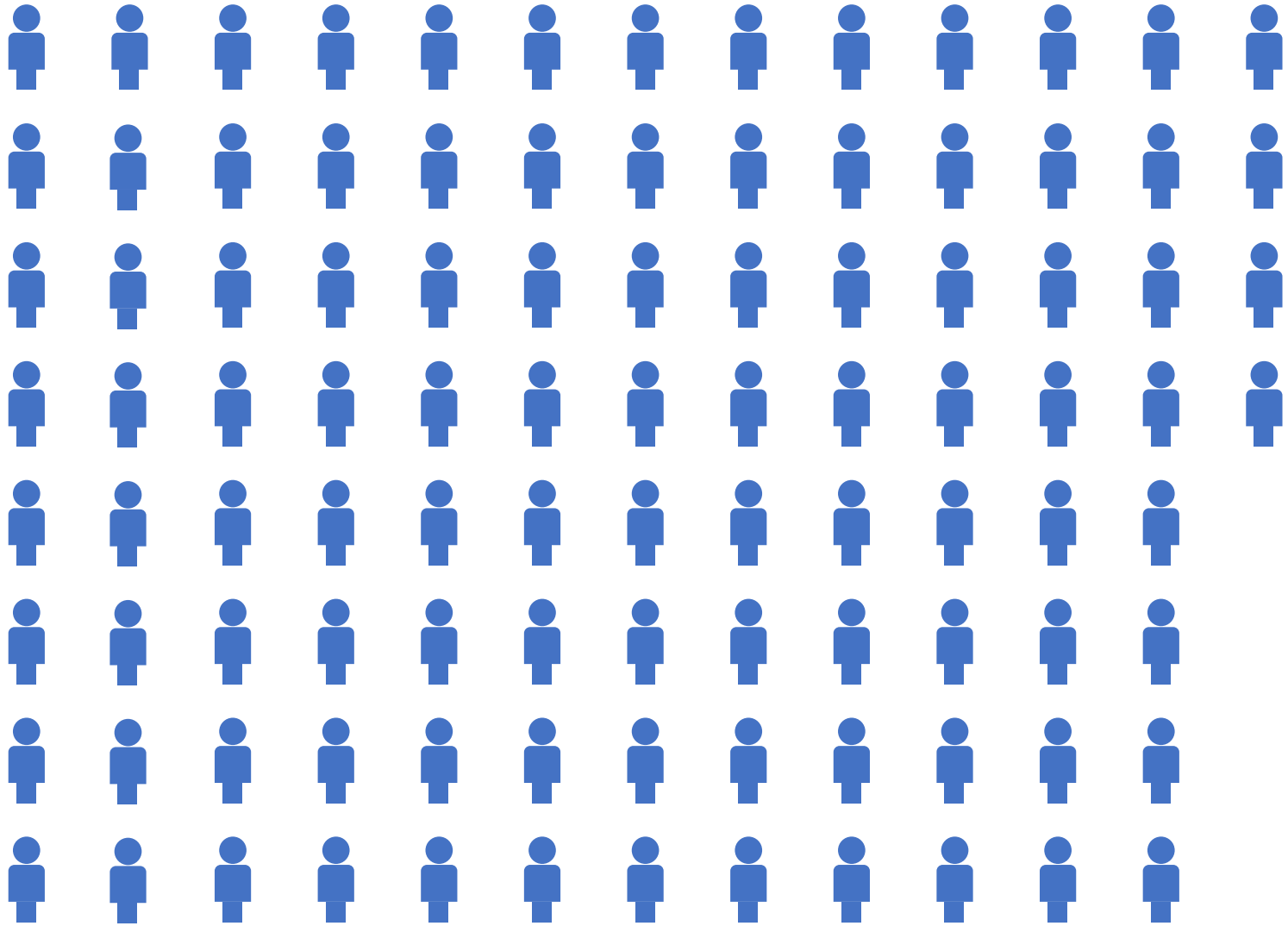


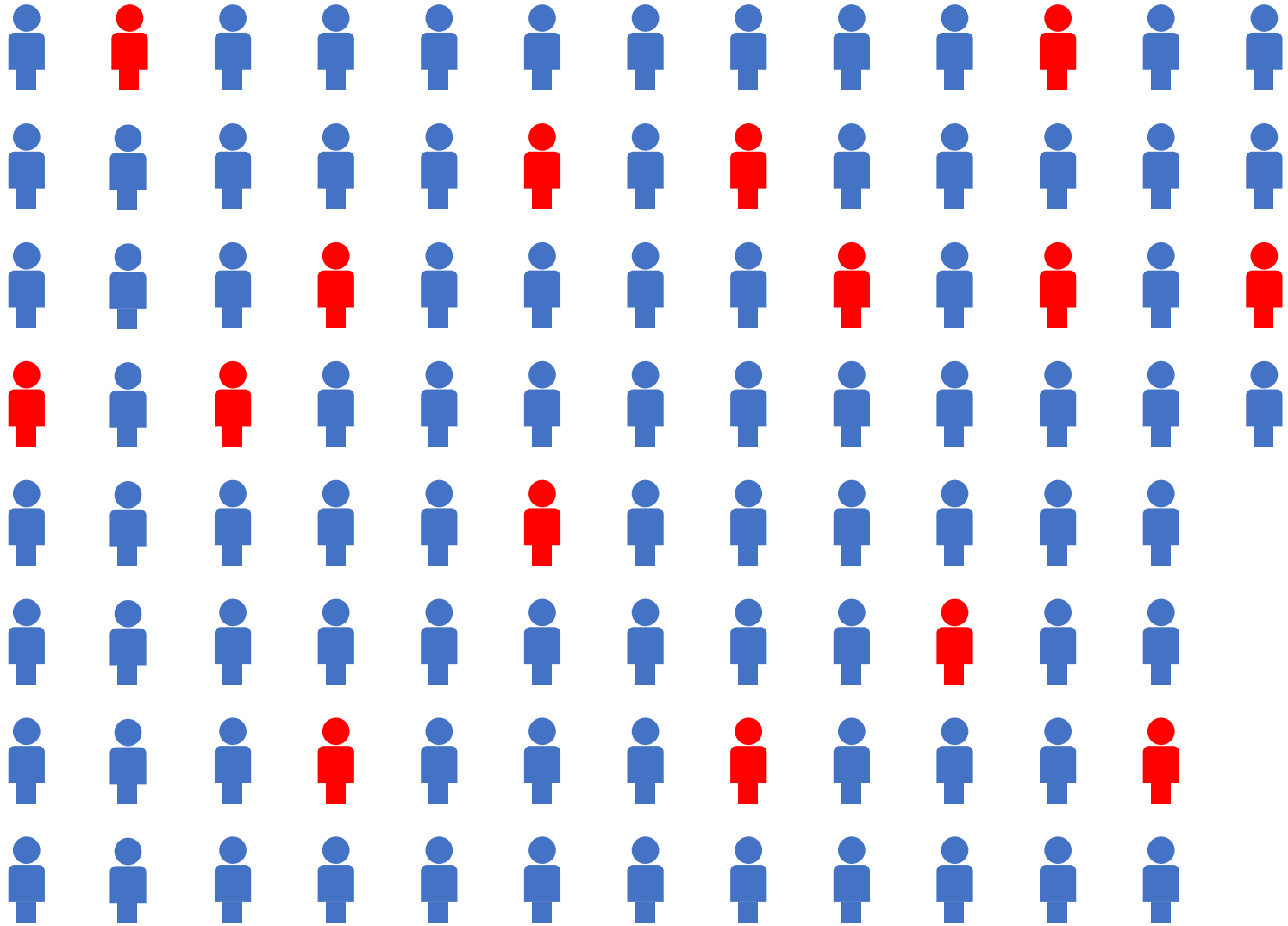
Central Theorems

Chris Piech
CS109, Stanford University

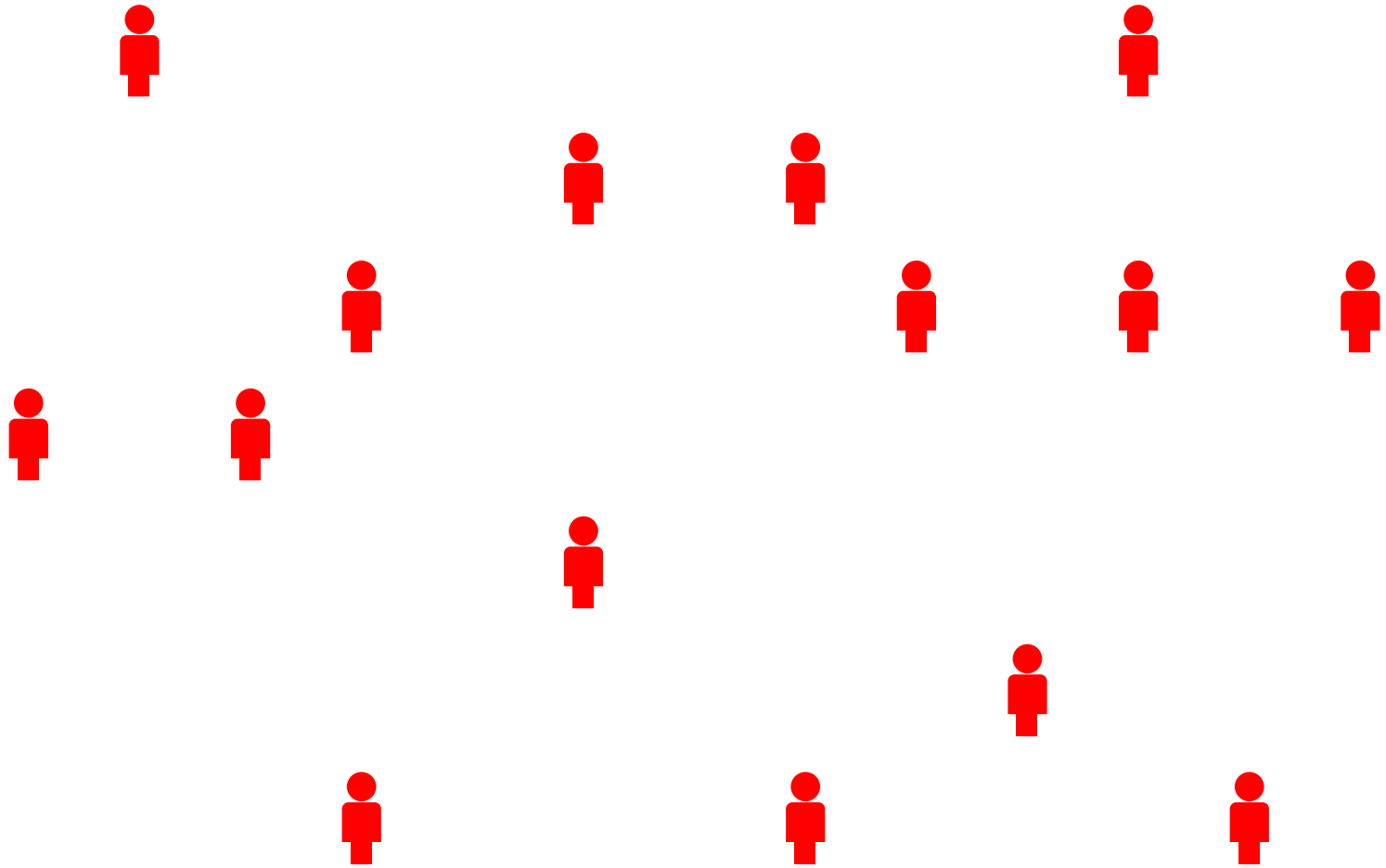
Population



Sample

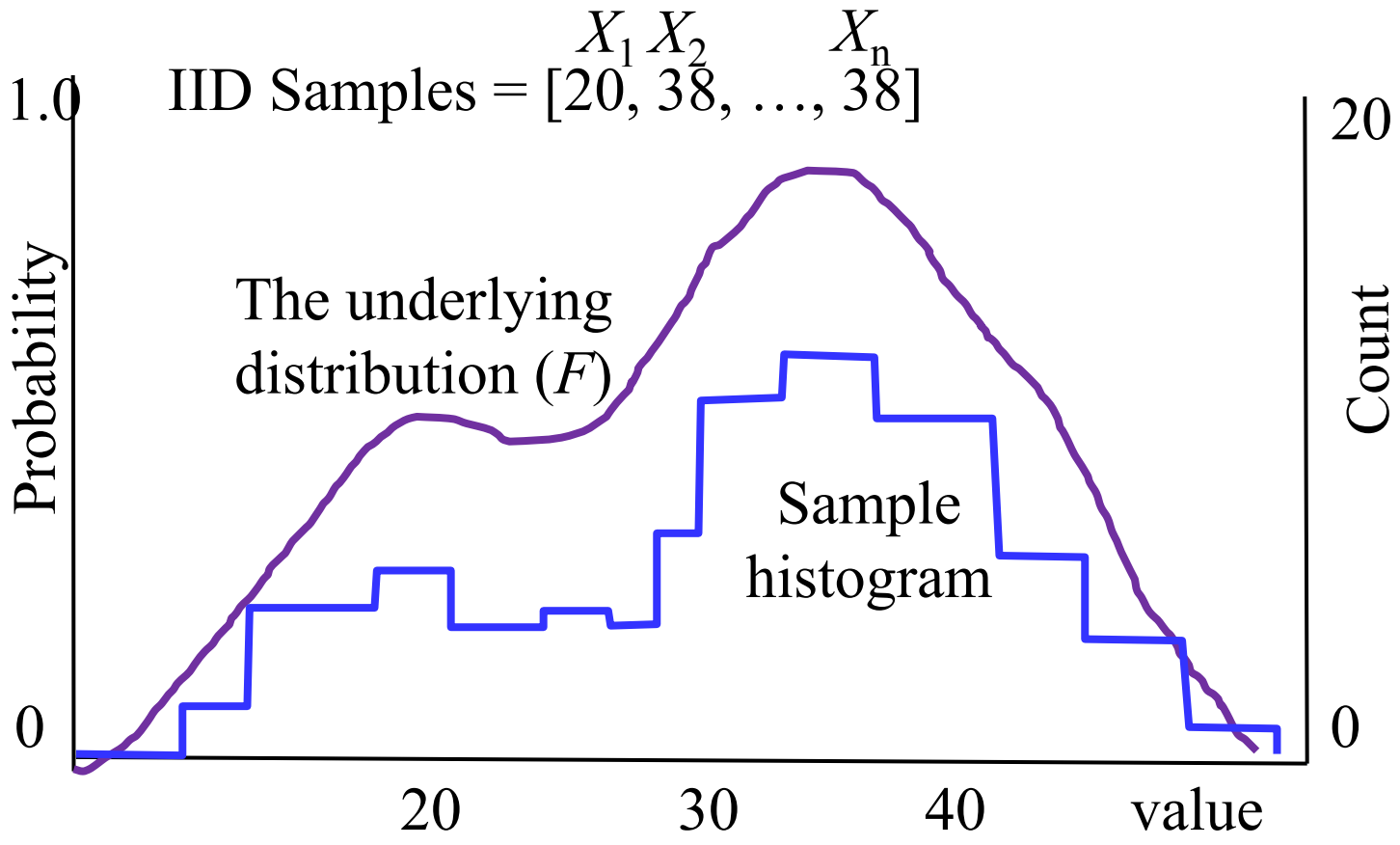


Sample



Collect one (or more) numbers from each person

Samples


X_1 X_2 X_n
IID Samples = [20, 38, …, 38]
1.0
0
Probability
The underlying distribution (F)
Sample histogram
20
Count
0
20
30
40
value

Sample Statistics

Sample Mean

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

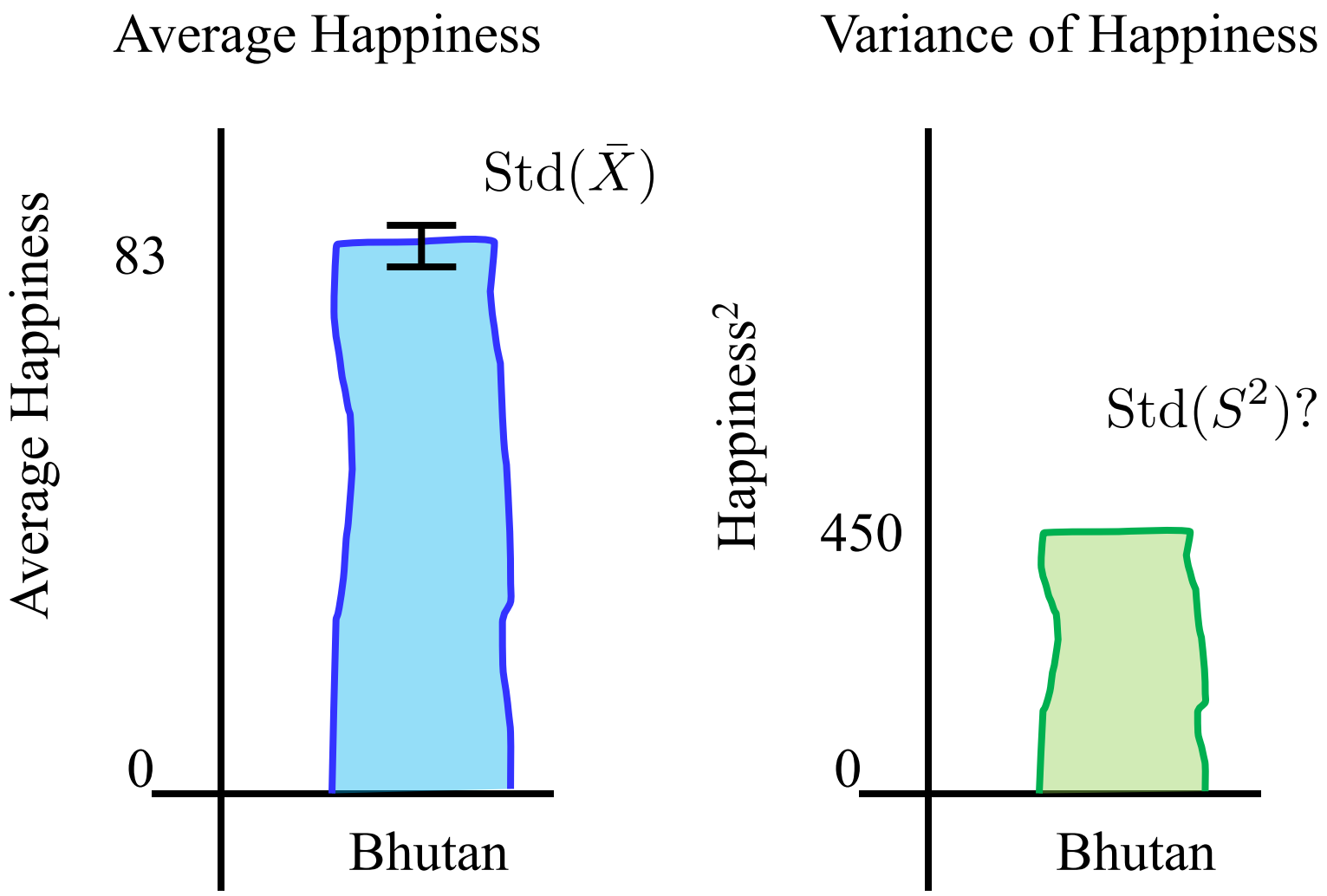
Sample Variance

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

Var of Sample Mean

$$\mathrm{Var}(\bar{X}) = \frac{S^2}{n}$$

Sample Mean



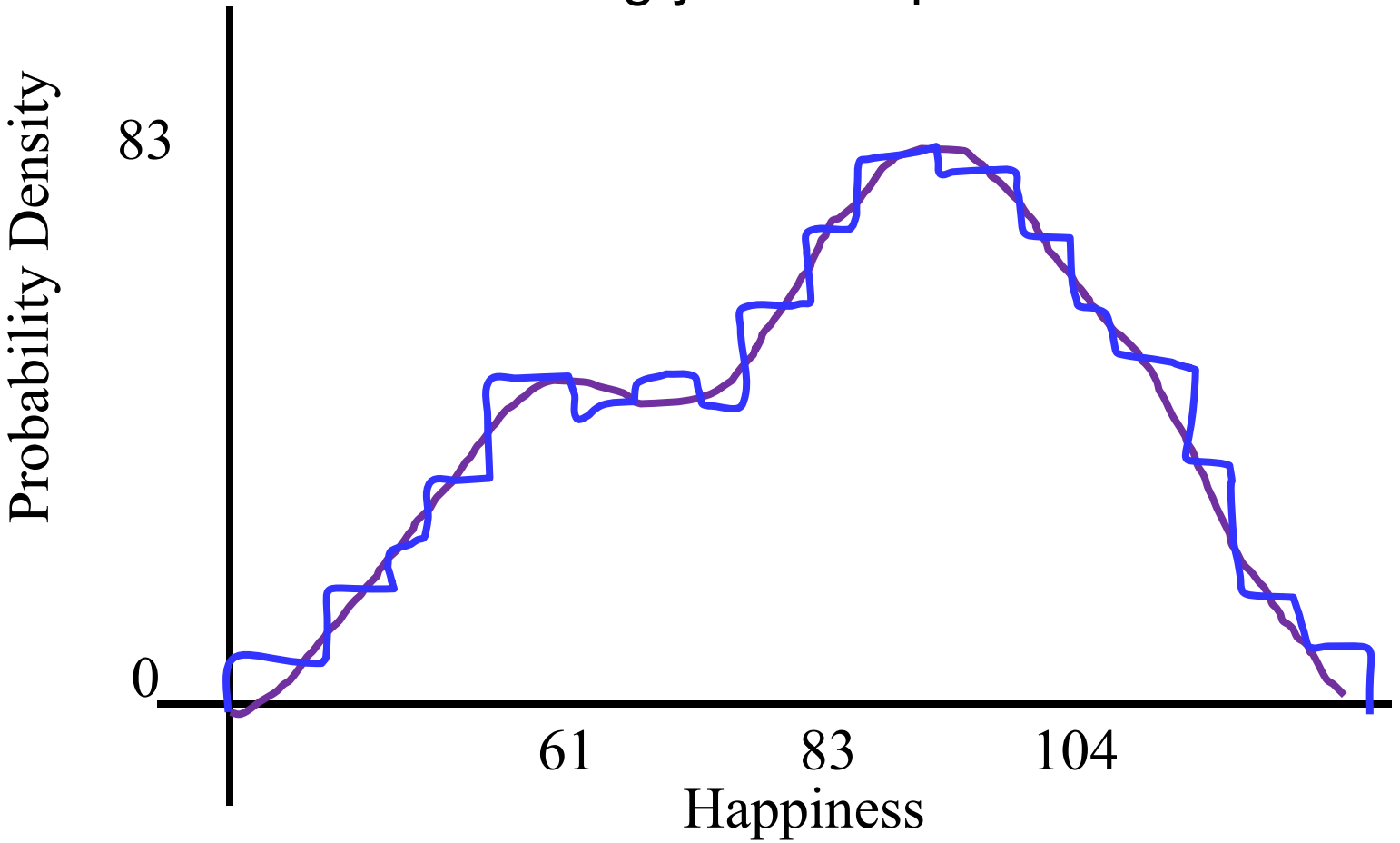
Claim: The average happiness of Bhutan is 83 ± 2

What if you want more?

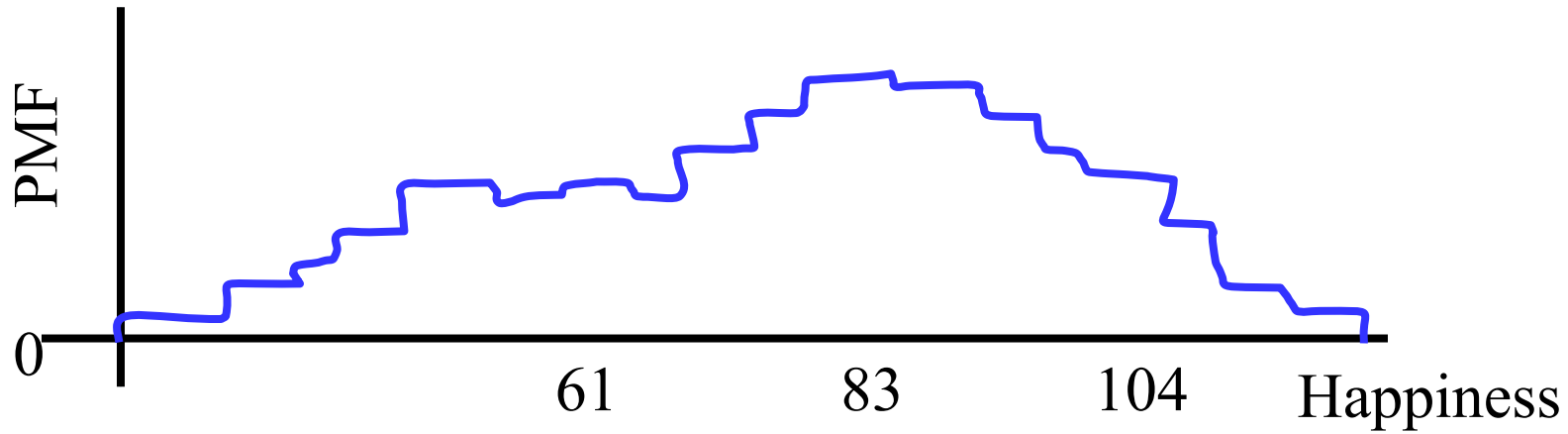
Bootstrap

Bootstrap Insight

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



Bootstrap of Means



Bootstrap Algorithm (sample):

1. Estimate the **PMF** using the sample
2. Repeat **10,000** times:
   a. Create a resample with **sample.size()** new samples from PMF
   b. **Recalculate the mean** on the resample
3. You now have a **distribution of your means**

Bootstrap for p values

Null Hypothesis Test

Population 1	Population 2
4.44	2.15
3.36	3.01
5.87	2.02
2.31	1.43
...	...
3.70	1.83
$\mu_1 = 3.1$	$\mu_2 = 2.4$

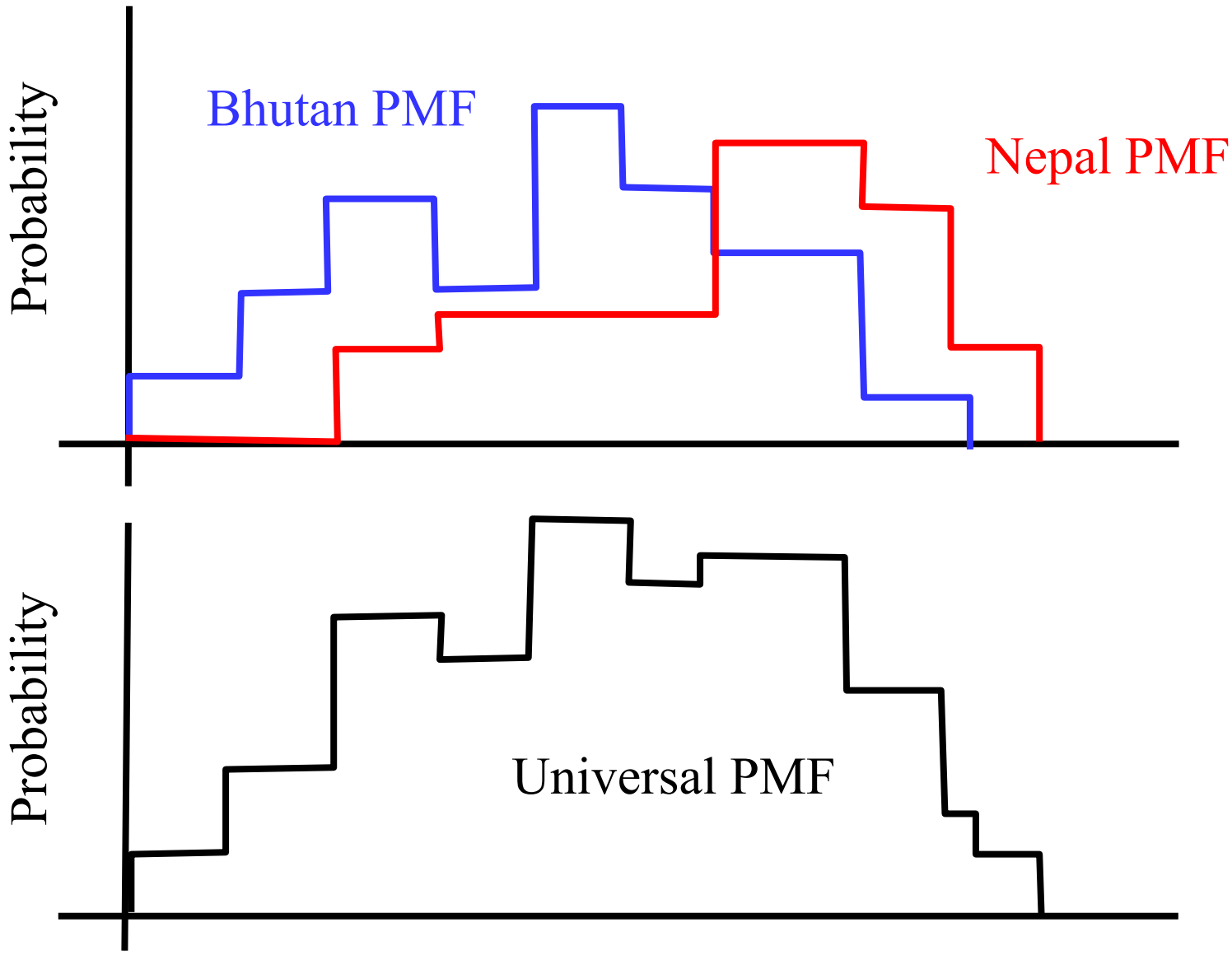
Null Hypothesis Test

Nepal Happiness	Bhutan Happiness
4.44	2.15
3.36	3.01
5.87	2.02
2.31	1.43
...	...
3.70	1.83
$\mu_1 = 3.1$	$\mu_2 = 2.4$

Claim: The difference in happiness between Nepal and Bhutan is 0.7 happiness points.

Null hypothesis: even if there is no pattern (ie the two samples are identically distributed) your results might have arisen by chance

Universal Sample



Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
   N = size of the bhutanSample
   M = size of the nepalSample

   universalSample = combine bhutanSamples and nepalSamples
   universalPmf = estimate the pmf of universalSample

   count = 0

   repeat 10,000 times:
      bhutanResample = draw N resamples from the universalPmf
      nepalResample = draw M resamples from the universalPmf
      muBhutan = sample mean of the bhutanResample
      muNepal = sample mean of the nepalResample
      meanDiff = |muNepal - muBhutan|
      if meanDiff > observedDifference:
         count += 1

   pValue = count / 10,000
```

Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
    N = size of the bhutanSample
    M = size of the nepalSample

    universalSample = combine bhutanSamples and nepalSamples
    universalPmf = estimate the pmf of universalSample

    count = 0

    repeat 10,000 times:
        bhutanResample = draw N resamples from the universalPmf
        nepalResample = draw M resamples from the universalPmf
        muBhutan = sample mean of the bhutanResample
        muNepal = sample mean of the nepalResample
        meanDiff = |muNepal - muBhutan|
        if meanDiff > observedDifference:
            count += 1

    pValue = count / 10,000
```

Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
    N = size of the bhutanSample
    M = size of the nepalSample

    universalSample = combine bhutanSamples and nepalSamples
    universalPmf = estimate the pmf of universalSample

    count = 0

    repeat 10,000 times:
        bhutanResample = draw N resamples from the universalPmf
        nepalResample = draw M resamples from the universalPmf
        muBhutan = sample mean of the bhutanResample
        muNepal = sample mean of the nepalResample
        meanDiff = |muNepal - muBhutan|
        if meanDiff > observedDifference:
            count += 1

    pValue = count / 10,000
```

Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
    N = size of the bhutanSample
    M = size of the nepalSample

    universalSample = combine bhutanSamples and nepalSamples
    universalPmf = estimate the pmf of universalSample

    count = 0

    repeat 10,000 times:
        bhutanResample = draw N resamples from the universalPmf
        nepalResample = draw M resamples from the universalPmf
        muBhutan = sample mean of the bhutanResample
        muNepal = sample mean of the nepalResample
        meanDiff = |muNepal - muBhutan|
        if meanDiff > observedDifference:
            count += 1

    pValue = count / 10,000
```

Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
   N = size of the bhutanSample
   M = size of the nepalSample

   universalSample = combine bhutanSamples and nepalSamples
   universalPmf = estimate the pmf of universalSample

   count = 0

   repeat 10,000 times:
      bhutanResample = draw N resamples from the universalPmf
      nepalResample = draw M resamples from the universalPmf
      muBhutan = sample mean of the bhutanResample
      muNepal = sample mean of the nepalResample
      meanDiff = |muNepal - muBhutan|
      if meanDiff > observedDifference:
         count += 1

   pValue = count / 10,000
```

Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
   N = size of the bhutanSample
   M = size of the nepalSample

   universalSample = combine bhutanSamples and nepalSamples
   universalPmf = estimate the pmf of universalSample

   count = 0

   repeat 10,000 times:
      bhutanResample = draw N resamples from the universalPmf
      nepalResample = draw M resamples from the universalPmf
      muBhutan = sample mean of the bhutanResample
      muNepal = sample mean of the nepalResample
      meanDiff = |muNepal - muBhutan|
      if meanDiff > observedDifference:
         count += 1

   pValue = count / 10,000
```

watch out for integer division

Bootstrap for P Values

```
def pvalueBootstrap(bhutanSample,nepalSample):
    N = size of the bhutanSample
    M = size of the nepalSample

    universalSample = combine bhutanSamples and nepalSamples
    universalPmf = estimate the pmf of universalSample

    count = 0

    repeat 10,000 times:
        bhutanResample = draw N resamples from universalSamples
        nepalResample = draw M resamples from universalSamples
        muBhutan = sample mean of the bhutanResample
        muNepal = sample mean of the nepalResample
        meanDiff = |muNepal - muBhutan|
        if meanDiff > observedDifference:
            count += 1

    pValue = count / 10,000
```

With replacement!!!

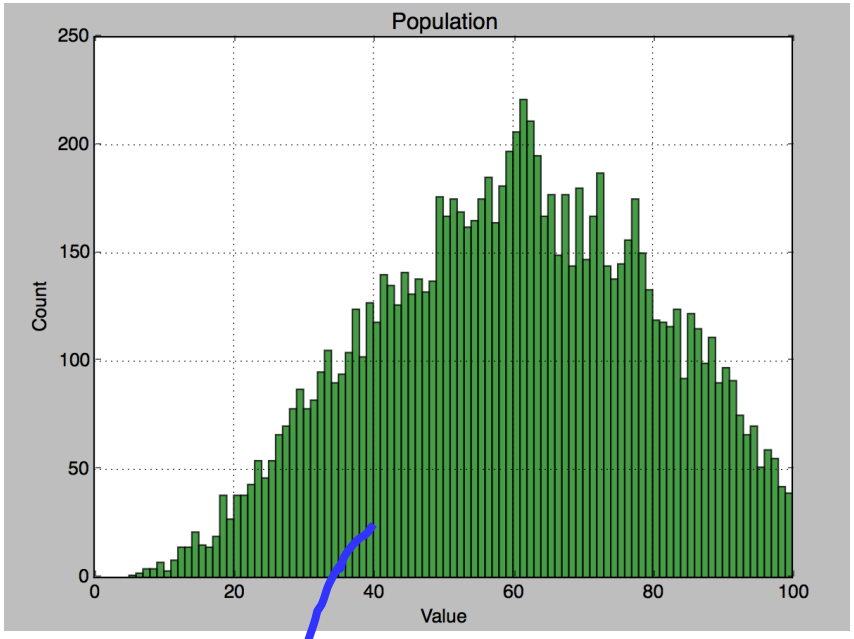
Bootstrap

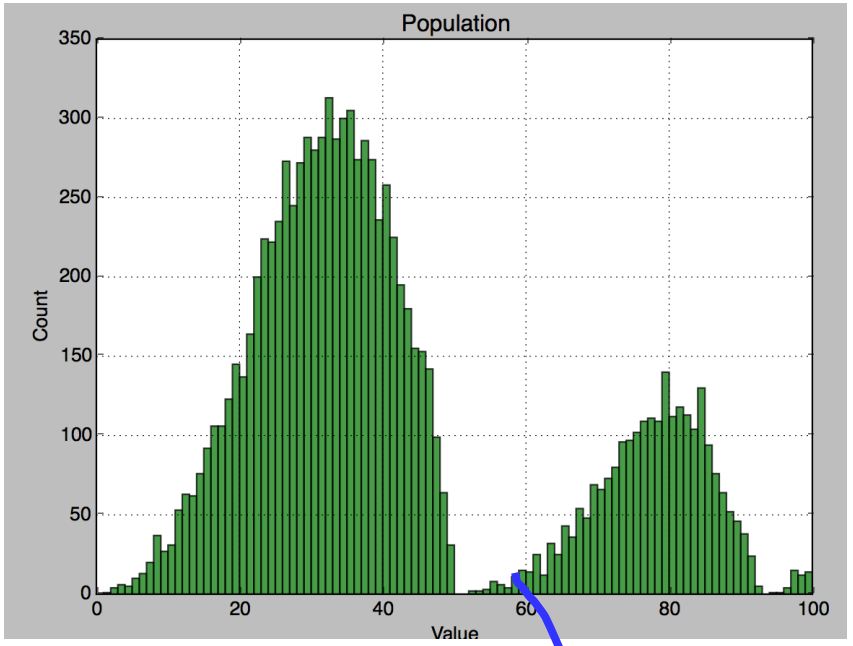
Lets try it!

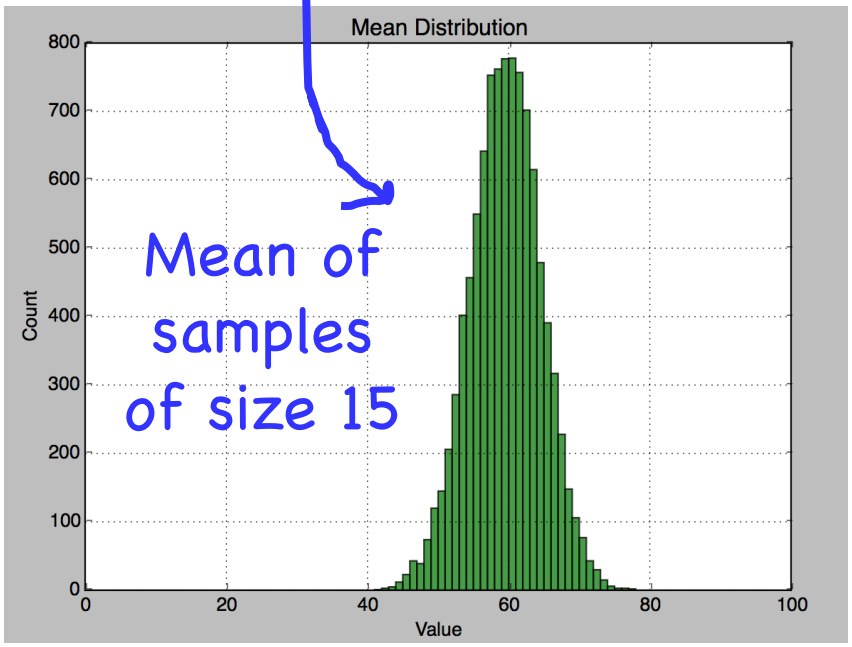
Null Hypothesis Test

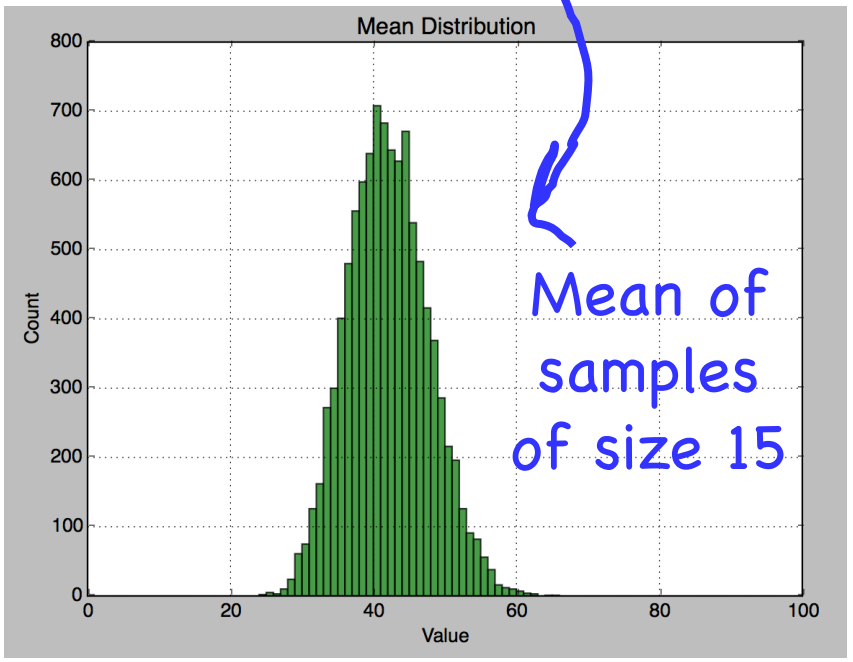
Nepal Happiness	Bhutan Happiness
4.44	2.15
3.36	3.01
5.87	2.02
2.31	1.43
...	...
3.70	1.83
$\mu_1 = 3.1$	$\mu_2 = 2.4$

Claim: The difference in happiness between Nepal and Bhutan is 0.7 happiness points (p = 0.008).


Population
Count
Value
0
50
100
150
200
250
0
20
40
60
80
100


Population
Count
Value
0
50
100
150
200
250
300
350
0
20
40
60
80
100


Mean Distribution
Count
Value
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100
Mean of samples of size 15


Mean Distribution
Count
Value
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100
Mean of samples of size 15

And now a moment of silence...

...before we present...

...a beautiful result of probability theory!

Central Limit Theorem

The Central Limit Theorem

- Consider I.I.D. random variables X_1, X_2, ...
  - X_i have distribution *F* with $E[X_i] = \mu$ and $Var(X_i) = \sigma^2$
  - Let: $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$
  - Central Limit Theorem:

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \quad \text{as} \quad n \to \infty$$

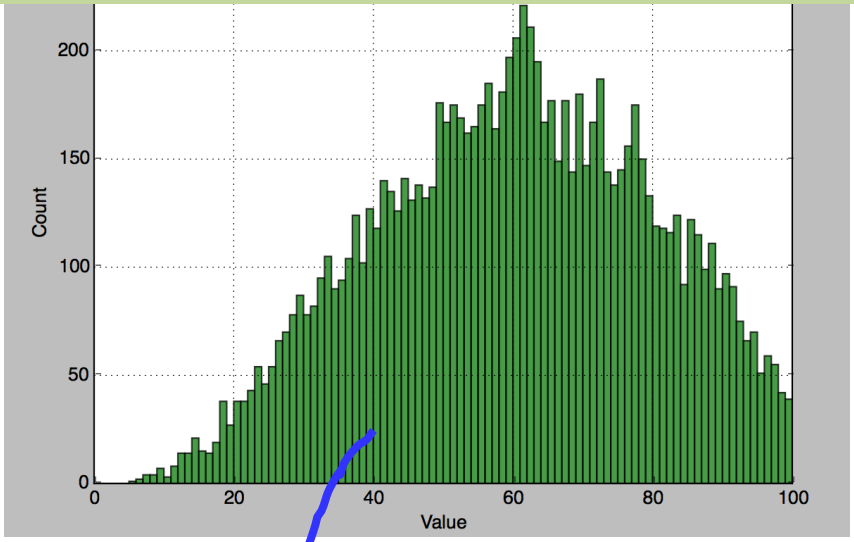
Demo

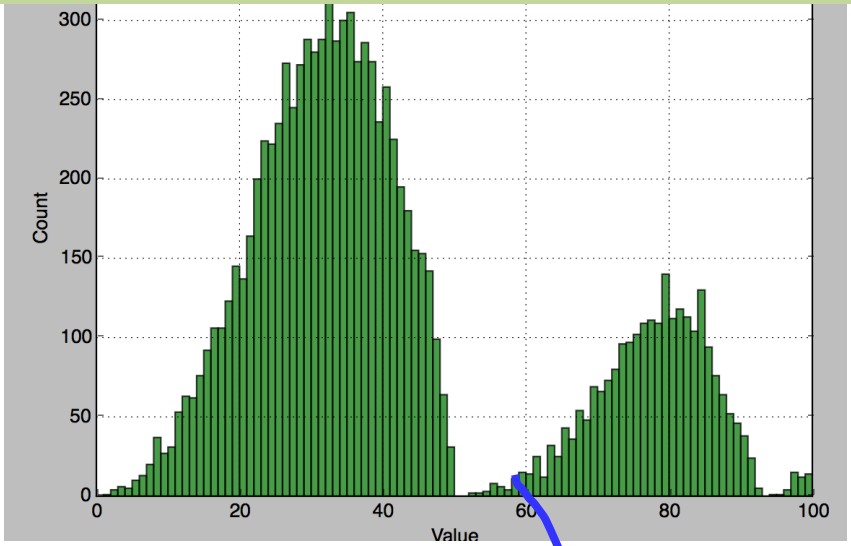
http://onlinestatbook.com/stat_sim/sampling_dist/

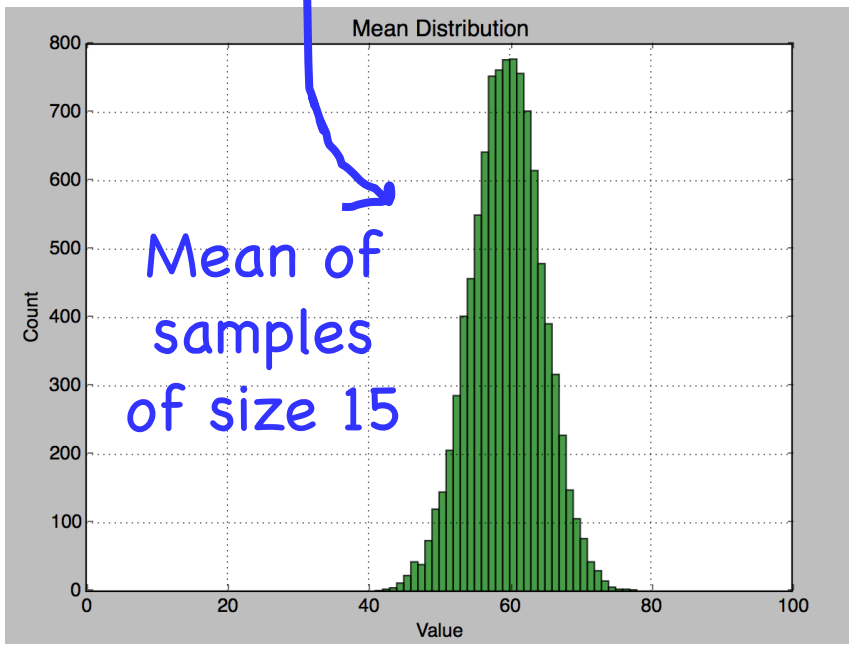
By the Central Limit Theorem, the sample mean of IID variables are distributed normally.

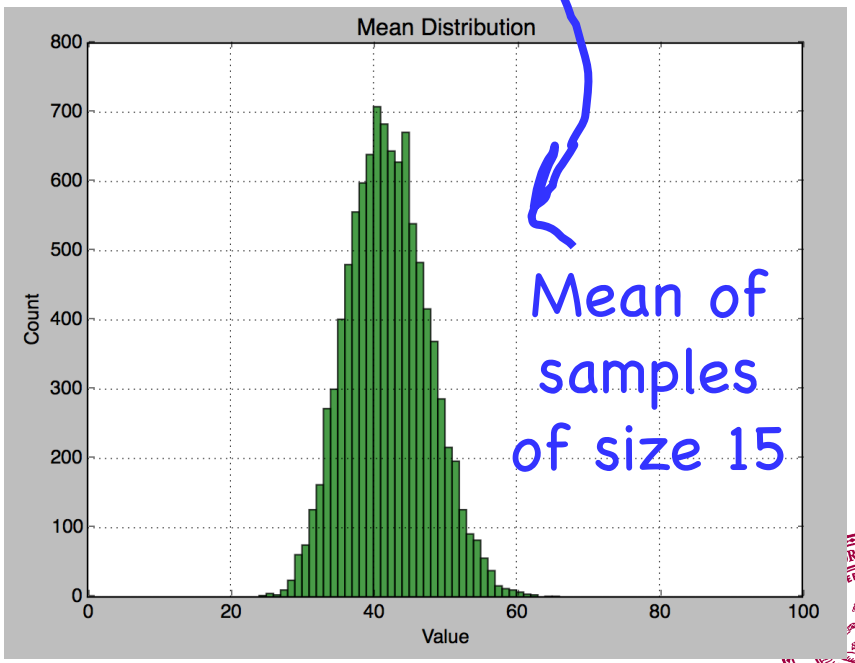
$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$$

C.L.T. Explains This









But Wait! There is More

- Consider I.I.D. random variables X_1, X_2, ...
  - X_i have distribution *F* with $E[X_i] = \mu$ and $Var(X_i) = \sigma^2$

$$\bar{X} = \frac{1}{n}\sum_{i}^{n} X_i \qquad \bar{X} \sim N(\mu, \frac{\sigma^2}{n})$$

$$Y = \sum_{i}^{n} X_i \qquad Y = n\bar{X}$$

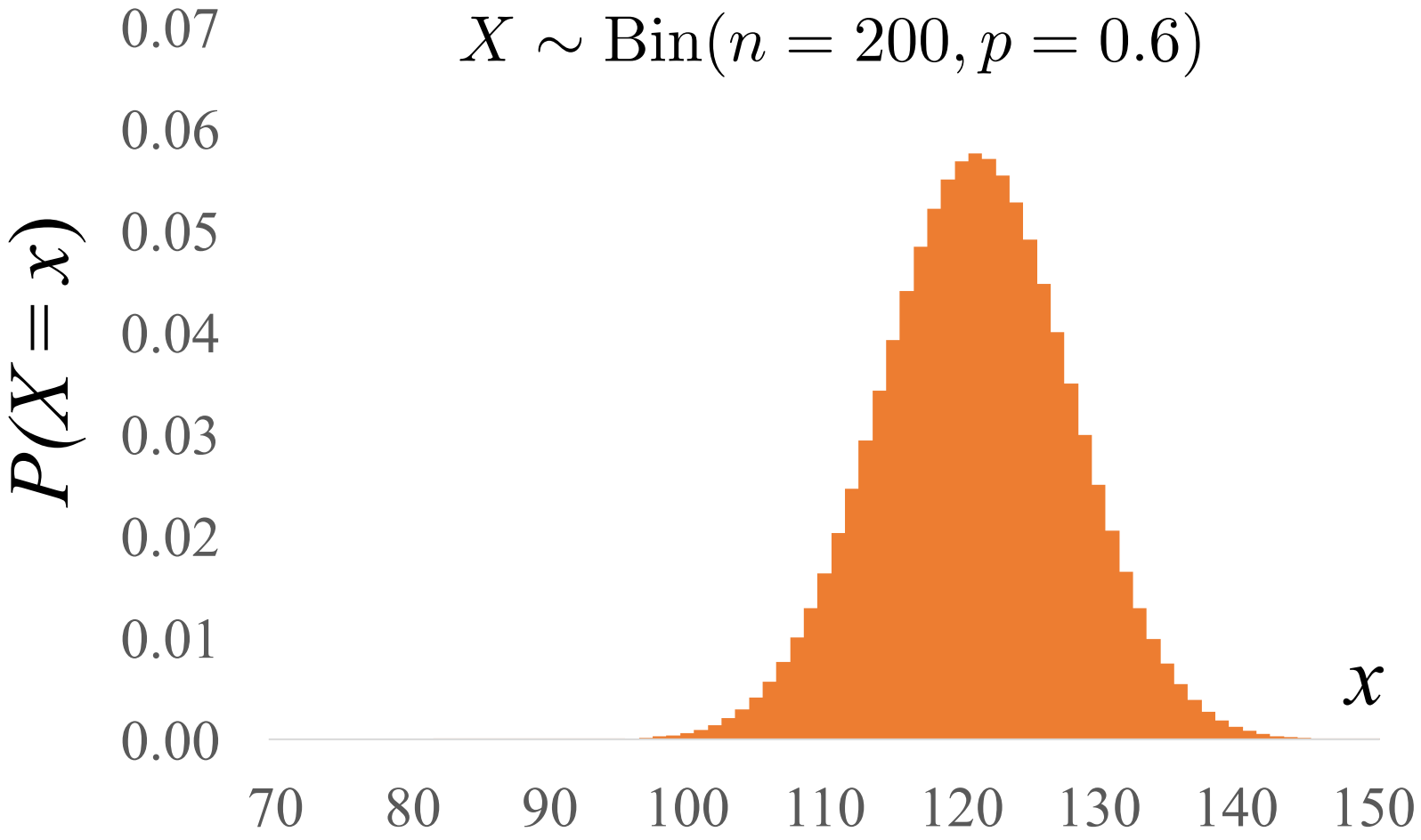
$$Y \sim N(n\mu, n^2\frac{\sigma^2}{n}) \qquad \text{Linear transform of a normal}$$

$$Y \sim N(n\mu, n\sigma^2) \qquad \text{Simplifying}$$

If sample mean of IID variables are distributed normally then so is the sum of IID variables

$$Y \sim N(n\mu, n\sigma^2)$$

C.L.T. Explains This



Binomial Approximation

- Consider I.I.D. Bernoulli variables X_1, X_2, ... With probability *p*
  - X_i have $E[X_i] = p$ and $Var(X_i) = p(1-p)$
  - Let: $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ Let: $Y = n\bar{X}$

Y is the sum of the Bernoullis

$$\bar{X} \sim N(\mu, \sigma^2) \text{ as } n \to \infty$$

Central Limit Theorem

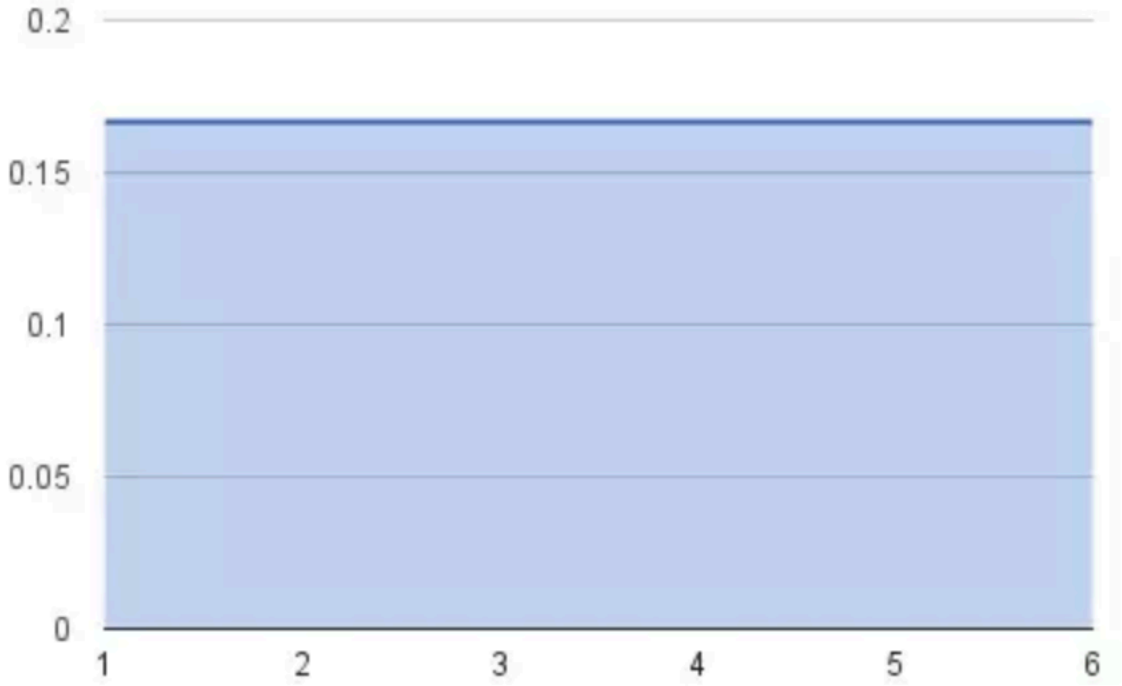
$$Y \sim N(n\mu, n\sigma^2)$$

$$Y \sim N(np, np(1-p))$$

Substituting mean and variance of Bernoilli

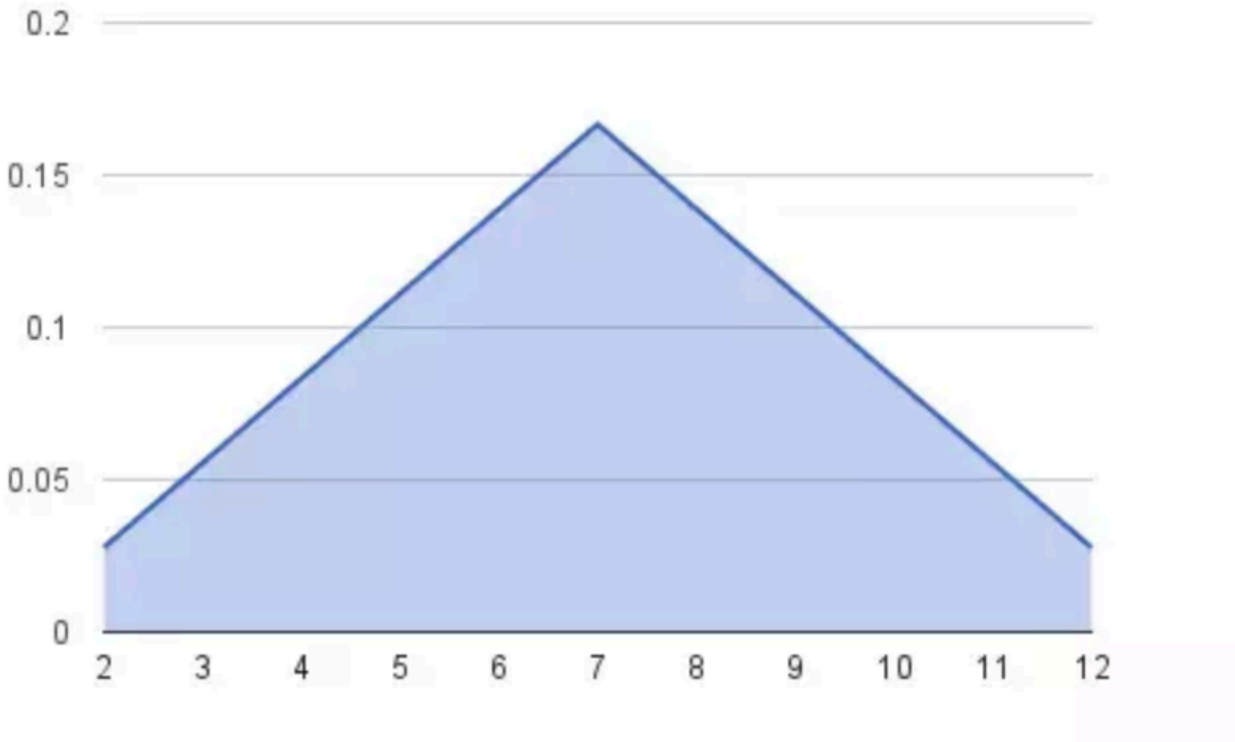
C.L.T. Intuition

This is the PMF of the sum of one dice



C.L.T. Intuition

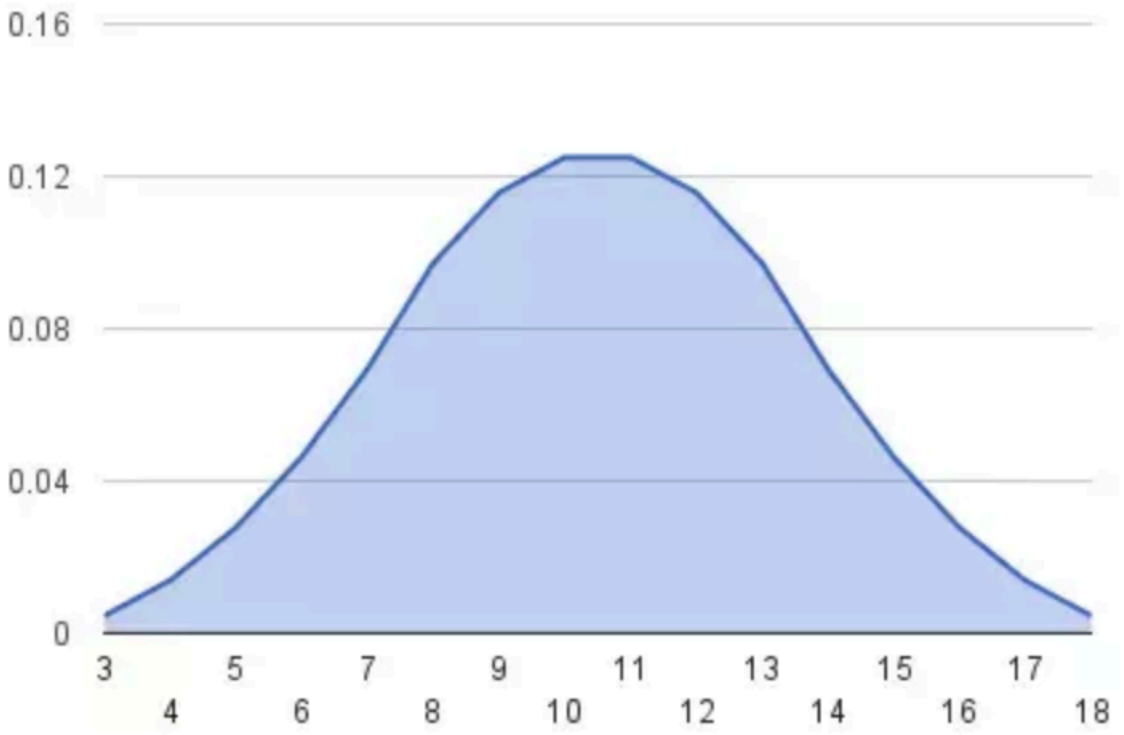
This is the PMF of the sum of two dice



Why is there more mass in the middle?

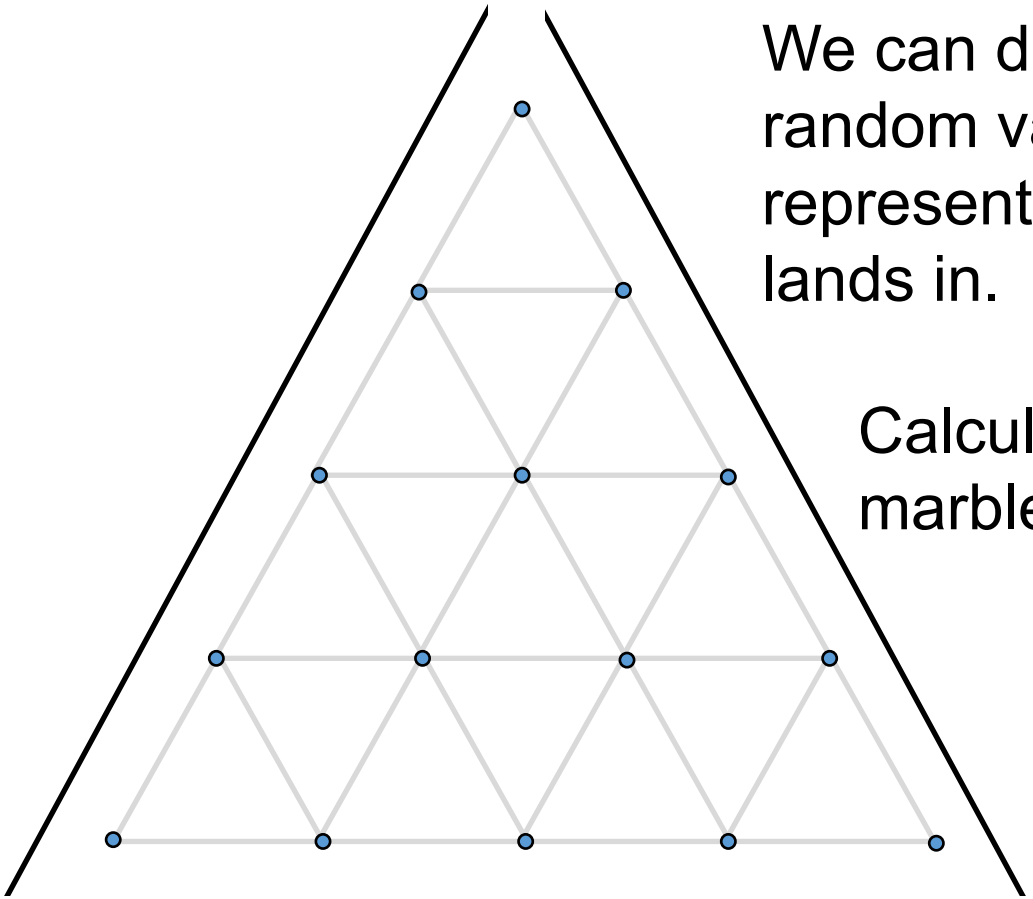
C.L.T. Intuition

This is the PMF of the sum of three dice



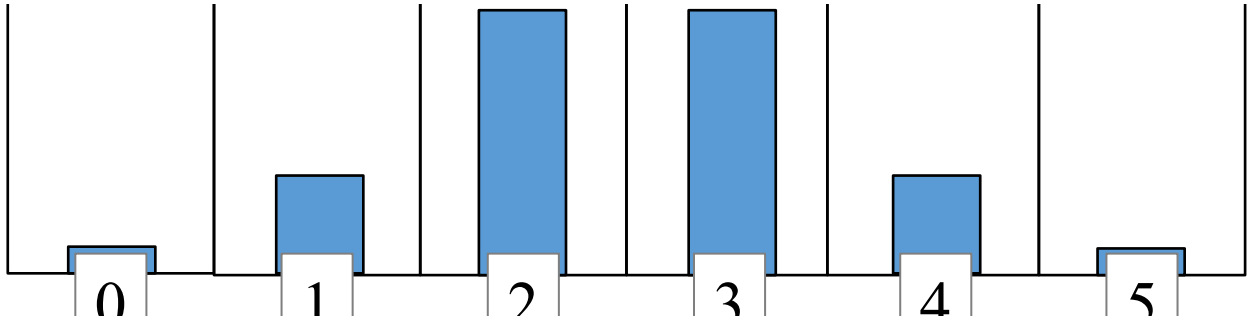
Why is there more mass in the middle?

C.L.T. Explains This



We can define an indicator random variable (***B***) which represents what bucket a marble lands in.

Calculate the probability of a marble landing in a bucket.



PDF

On the Proof of the CLT

- The proof of the CLT uses the Fourier transform of the probability mass of the sample distance from the mean, divided by standard deviation, and shows that this approaches an exponential function in the limit:

$$f(x) = e^{-\frac{x^2}{2}}$$

- That exponential function is in turn the Fourier transform of the Standard Normal. The Fourier transform of a probability density function is called a *Characteristic Function*.

- The proof is beyond the scope of CS109.

Central Limit Theorem in the Real World

- CLT is why some things in "real world" appear Normally distributed
  - Many quantities are sum of independent variables
  - Exams scores
    - Sum of individual problems on the SAT
    - Why does the CLT not apply to our midterm?
  - Election polling
    - Ask 100 people if they will vote for candidate X (p_1 = # "yes"/100)
    - Repeat this process with different groups to get p_1, ... , p_n
    - Will have a normal distribution over p_i
    - Can produce a "confidence interval"
      - How likely is it that estimate for true p is correct

Midterm on the Central Limit Theorem

- Start with 236 midterm scores: X_1, X_2, ..., X_{236}
  - E[X_i] = 84 and Var(X_i) = 370
  - Created 50 samples of size *n* = 10 with sums
  - ($Y_1, Y_2, .. , Y_{50}$)
  - Prediction by CLT: $\bar{Y}_i \sim N(84, 37)$

$$Z_i = \frac{\bar{Y}_i - 84}{\sqrt{37}}$$

$$\bar{Z} = \frac{1}{50}\sum_{i=1}^{50} Z_i = 4 \times 10^{-16}$$

$$\mathrm{Var}(\bar{Z}) = 0.997$$

Once Upon a Time…

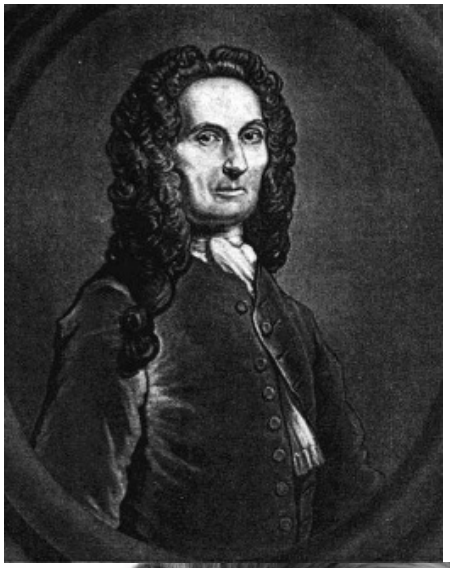
THE
DOCTRINE
OF
CHANCES:
OR,
A Method of Calculating the Probability
of Events in Play.

By *A. De Moivre.* F. R. S.

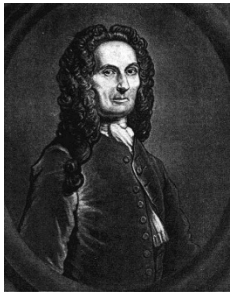
LONDON:
Printed by *W. Pearson,* for the Author. MDCCXVIII.

1733

Abraham De Moivre



Once Upon a Time…



- History of the Central Limit Theorem
  - 1733: CLT for X ~ Ber(1/2) postulated by Abraham de Moivre

  - 1823: Pierre-Simon Laplace extends de Moivre's work to approximating Bin(n, p) with Normal

  - 1901: Aleksandr Lyapunov provides precise definition and rigorous proof of CLT
  - 2016: Beyonce releases Lemonade

    - It was her 6th album, bringing her total number of songs to 214
    - Mean quality of subsamples of songs is Normally distributed (thanks to the Central Limit Theorem)

Estimating Clock Running Time

- Have new algorithm to test for running time
  - Mean (clock) running time: μ = *t* sec.
  - Variance of running time: σ^2 = 4 sec^2.
  - Run algorithm repeatedly (I.I.D. trials), measure time
    - How many trials s.t. estimated time = $t \pm 0.5$ with 95% certainty?
    - X_i = running time of *i*-th run (for $1 \leq i \leq n$), $\bar{X}$ is the sample mean

---

$$0.95 = P(-0.5 < \bar{X} - t < 0.5)$$

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \sim N(t, \frac{4}{n})$$ By CLT

$$\bar{X} - t \sim N(0, \frac{4}{n})$$ By linear transform of a normal

$$0.95 = P(-0.5 < \bar{X} - t < 0.5) \qquad \bar{X} - t \sim N(0, \frac{4}{n})$$

$$0.95 = P(\frac{-0.5 - 0}{\sqrt{\frac{4}{n}}} < \bar{X} - t < \frac{0.5 - 0}{\sqrt{\frac{4}{n}}})$$

$$= P(\frac{-0.5\sqrt{n}}{2} \leq Z \leq \frac{0.5\sqrt{n}}{2})$$

$$= \phi(\frac{\sqrt{n}}{n}) - \phi(-\frac{\sqrt{n}}{n})$$

$$= 2\phi(\frac{\sqrt{n}}{4}) - 1$$

$$0.95 = 2\phi(\frac{\sqrt{n}}{4}) - 1$$

$$0.975 = \phi(\frac{\sqrt{n}}{4})$$

$$\phi^{-1}(0.975) = \frac{\sqrt{n}}{4}$$

$$1.96 = \frac{\sqrt{n}}{4}$$

$$n = 61.4$$

William Sealy Gosset
(aka Student)

It’s play time!

Sum of Dice

- You will roll 10 6-sided dice (X_1, X_2, …, X_{10})
  - X = total value of all 10 dice = $X_1 + X_2 + \ldots + X_{10}$
  - Win if: $X \leq 25$ or $X \geq 45$
  - Roll!
- And now the truth (according to the CLT)…

Sum of Dice

- You will roll 10 6-sided dice ($X_1, X_2, \ldots, X_{10}$)
  - X = total value of all 10 dice = $X_1 + X_2 + \ldots + X_{10}$
  - Win if: $X \leq 25$ or $X \geq 45$

- Recall CLT: $X = \sum_{i}^{n} X_i \to N(n\mu, n\sigma^2) \qquad \text{As } n \to \infty$
  - Determine P(X ≤ 25 or X ≥ 45) using CLT:

$$\mu = E[X_i] = 3.5 \qquad \sigma^2 = \text{Var}(X_i) = \frac{35}{12} \qquad X \approx N(35, 29.2)$$

$$1 - P(25.5 < X < 44.5) = 1 - P\left(\frac{25.5 - 35}{\sqrt{29.2}} < Z < \frac{44.5 - 35}{\sqrt{29.2}}\right)$$

$$\approx 1 - (2\Phi(1.76) - 1) \approx 2(1 - 0.9608) = 0.0784$$

Wonderful Form of Cosmic Order

I know of scarcely anything so apt to impress the imagination as the wonderful form of cosmic order expressed by the ”[Central limit theorem]". The law would have been personified by the Greeks and deified, if they had known of it. It reigns with serenity and in complete self-effacement, amidst the wildest confusion. The huger the mob, and the greater the apparent anarchy, the more perfect is its sway. It is the supreme law of Unreason. Whenever a large sample of chaotic elements are taken in hand and marshalled in the order of their magnitude, an unsuspected and most beautiful form of regularity proves to have been latent all along.

-Sir Francis Galton