Elementary statistics

BBM 101 - Introduction to Programming I

Hacettepe University
Fall 2015

Fuat Akal, Aykut Erdem, Erkut Erdem, Vahid Garousi

Slides based on material prepared by Ruth Anderson, Michael Ernst and Bill Howe in the course CSE 140 at University of Washington and John Guttag in the course MIT 6.00.2X at MIT

A dice-rolling game

- Two players each roll a die
- The higher roll wins
  - Goal: roll as high as you can!
- Repeat the game 6 times

Hypotheses regarding the outcome

- Luck
- Fraud
  - loaded die
  - inaccurate reporting

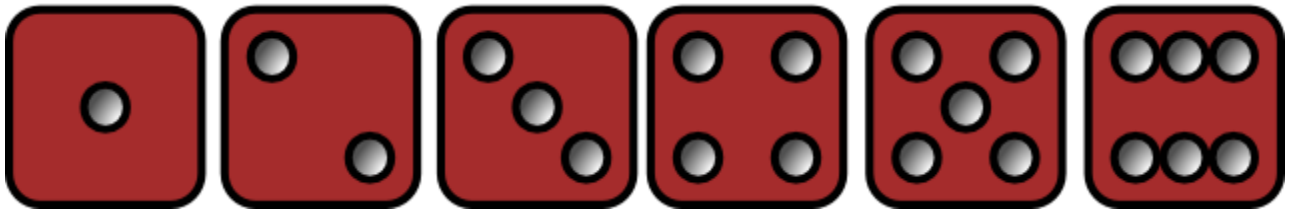
- How likely is luck?
- How do we decide?

Questions that statistics can answer

- I am flipping a coin. Is it fair?
  How confident am I in my answer?
- I have two bags of beans, each containing some black and some white beans. I have a handful of beans. Which bag did the handful come from?
- I have a handful of beans, and a single bag. Did the handful come from that bag?

- Does this drug improve patient outcomes?
- Which website design yields greater revenue?
- Which baseball player should my team draft?
- What premium should an insurer charge?
- Which chemical process leads to the best-tasting beer?

What can happen when you roll a die?

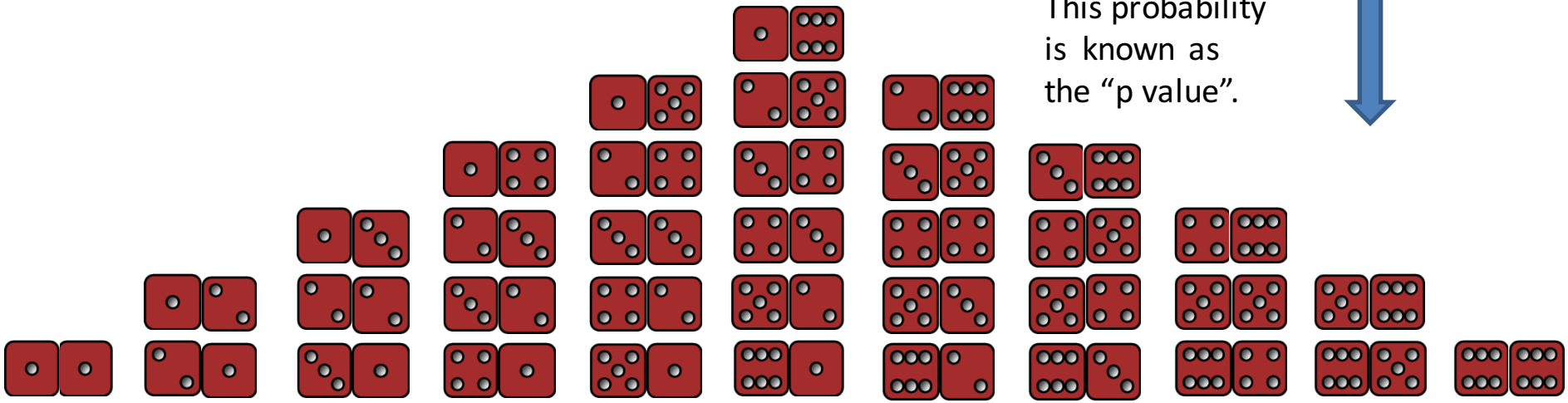


What is the likelihood of each?

What can happen when you roll two dice?

How likely are you to roll *11 or higher*?

This probability is known as the "p value".



2 3 4 5 6 7 8 9 10 11 12

How to compute p values

- Via a statistical formula
  - Requires you to make assumptions and know which formula to use
- Computationally (simulation)
  - Run many experiments
  - Count the fraction with a better result
    - Requires a metric/measurement for “better”
  - Requires you to be able to run the experiments
  - We will use this approach exclusively

Analogy between hypothesis testing and mathematical proofs

"The underlying logic [of hypothesis testing] is similar to a proof by contradiction. To prove a mathematical statement, A, you assume temporarily that A is false. If that assumption leads to a contradiction, you conclude that A must actually be true."

From the book *Think Statistics* by Allen Downey

Interpreting p values

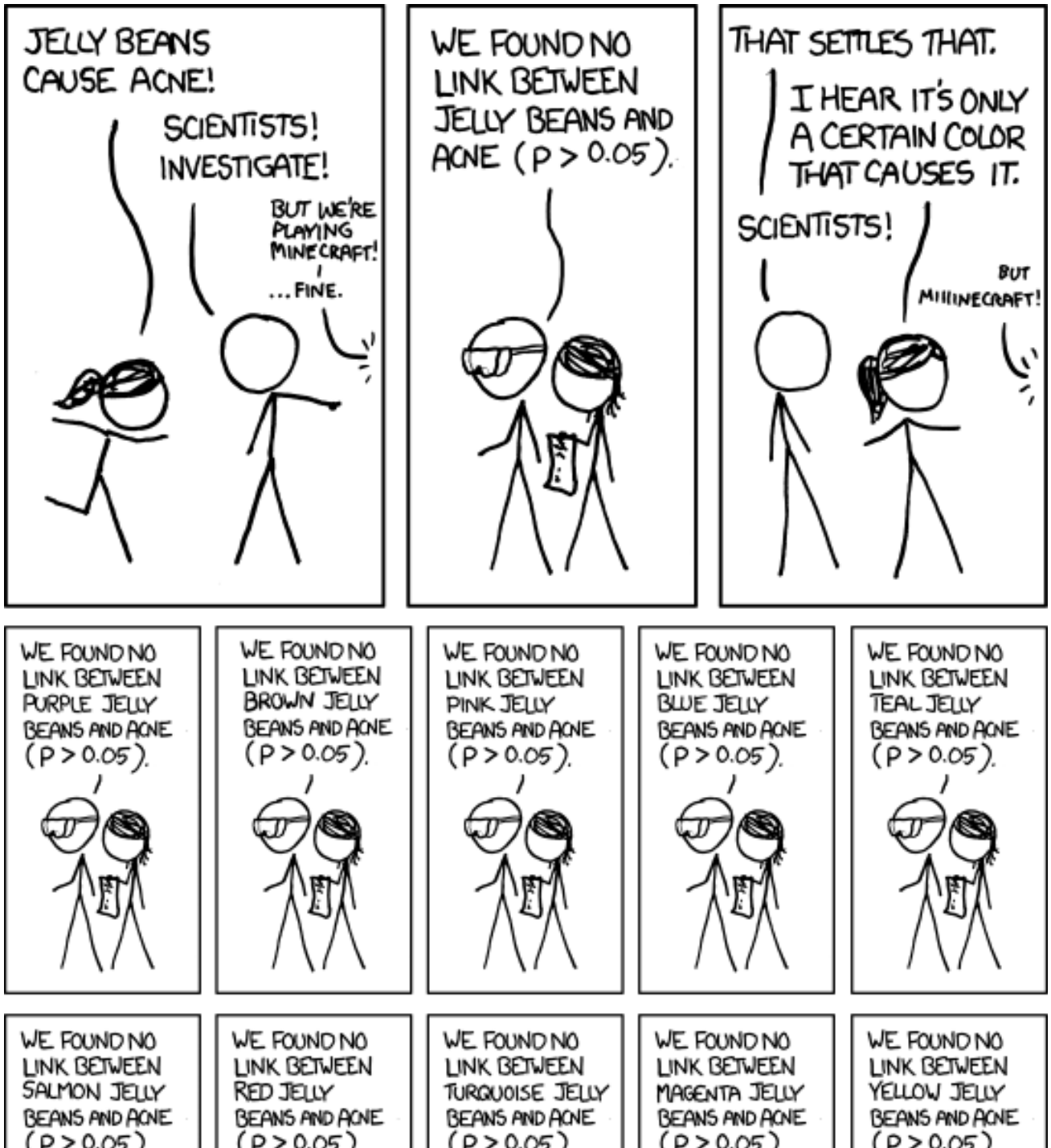
p value of 5% or less = statistically significant

- This is a *convention*; there is nothing magical about 5%

Two types of errors may occur in statistical tests:

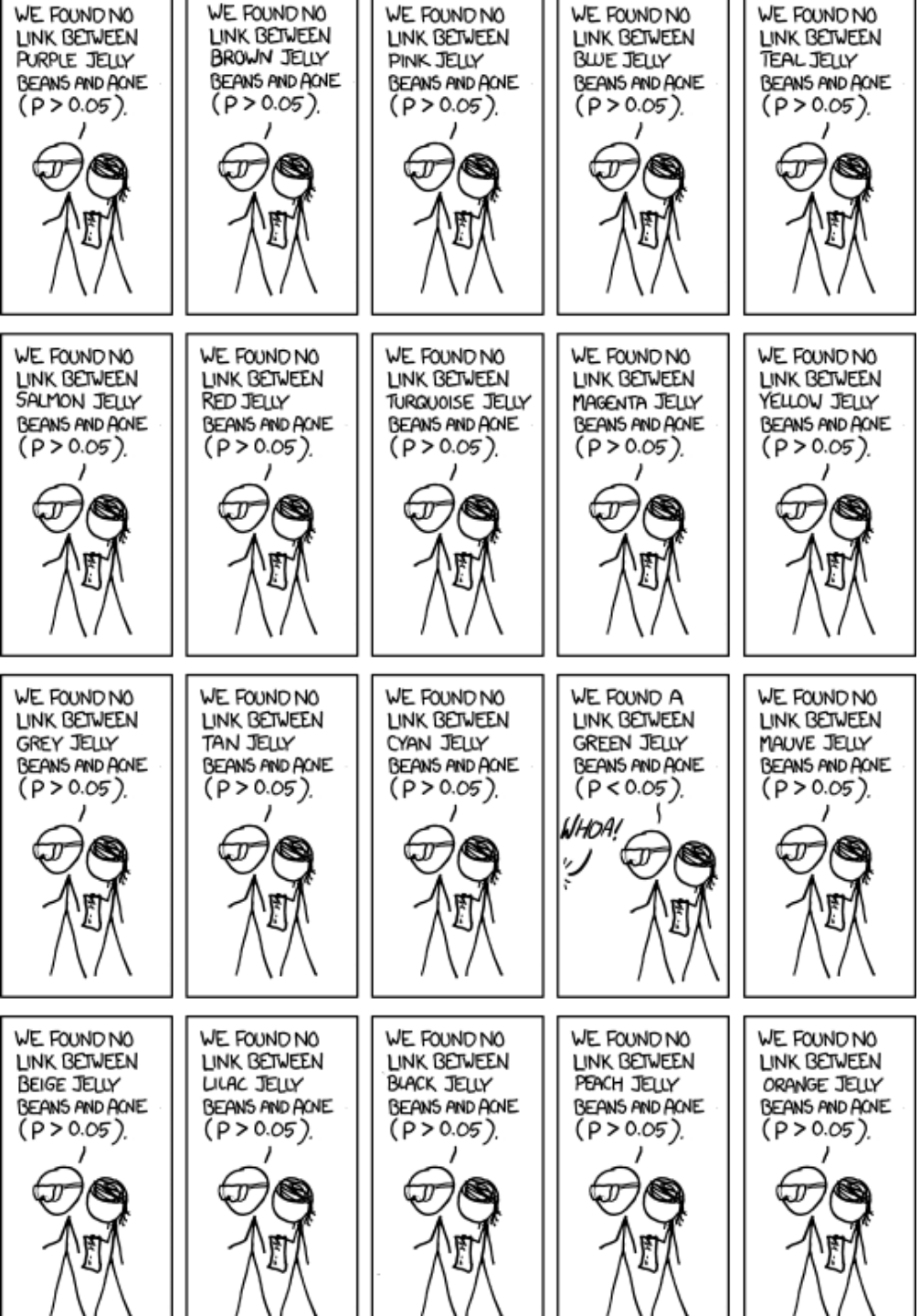
- false positive (or false alarm or Type I error): no real effect, but report an effect (through good/bad luck or coincidence)
  - If no real effect, a false positive occurs about 1 time in 20
- false negative (or miss or Type II error): real effect, but report no effect (through good/bad luck or coincidence)

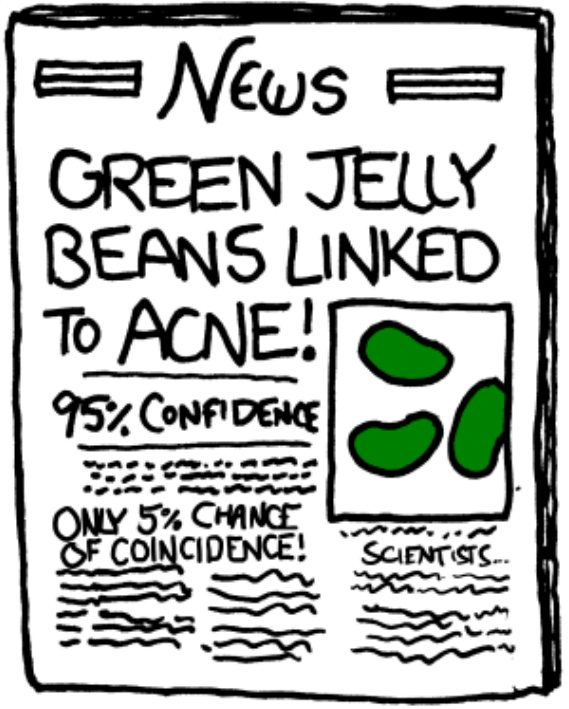
The *larger* the sample, the *less the likelihood* of a false positive or negative



A false positive

http://xkcd.com/882/





http://xkcd.com/882/

Summary of statistical methodology

1. Decide on a metric (bigger value = better)
2. Observe what you see in the real world
3. Hypothesize that what you saw is normal/typical

   This is the "null hypothesis"
4. Simulate the real world many times
5. How different is what you observed from the simulations?

   What percent of the simulation values are the real world values bigger than?
6. If the percentage is 95% or more, reject the null hypothesis

A common error

1. Observe what you see in the real world
2. Decide on a metric (bigger value = better)

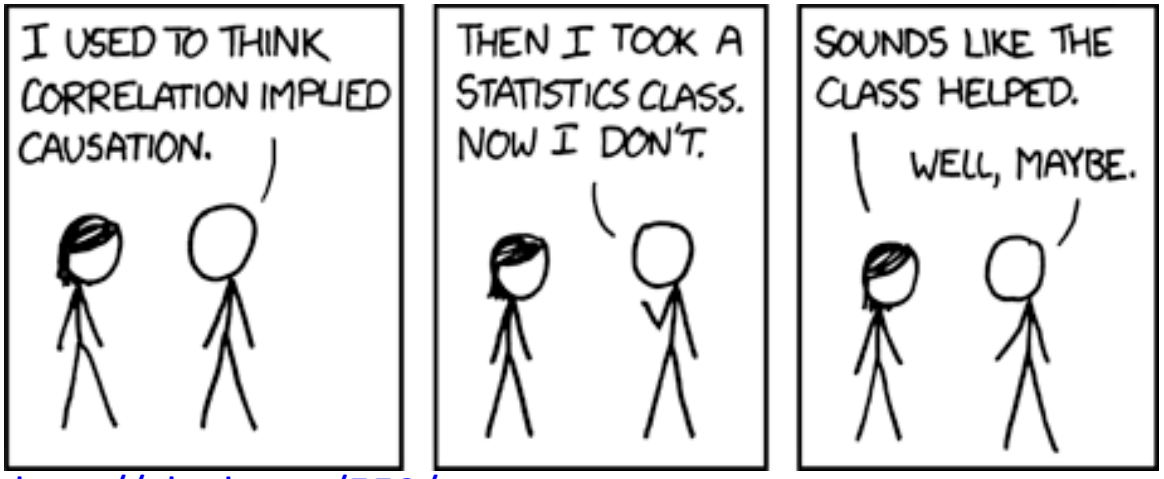
This is *backwards*

For any observation, there is something unique about it.

Example: Roll dice, then be amazed because what are the odds you would get exactly that combination of rolls?

Correlation ≠ causation

Ice cream sales and murder rates are correlated



http://xkcd.com/552/

Statistical significance ≠ practical importance

Don’t trust your intuition

- People have very bad statistical intuition
- It’s much better to follow the methodology and do the experiments

The Law of Large Numbers

The Law of Large Numbers

- In repeated **independent tests** with the same **actual probability** *p* of a particular outcome in each test,
- the chance that the **fraction of times** that outcome occurs differs from *p* converges to zero as the number of trials goes to infinity.

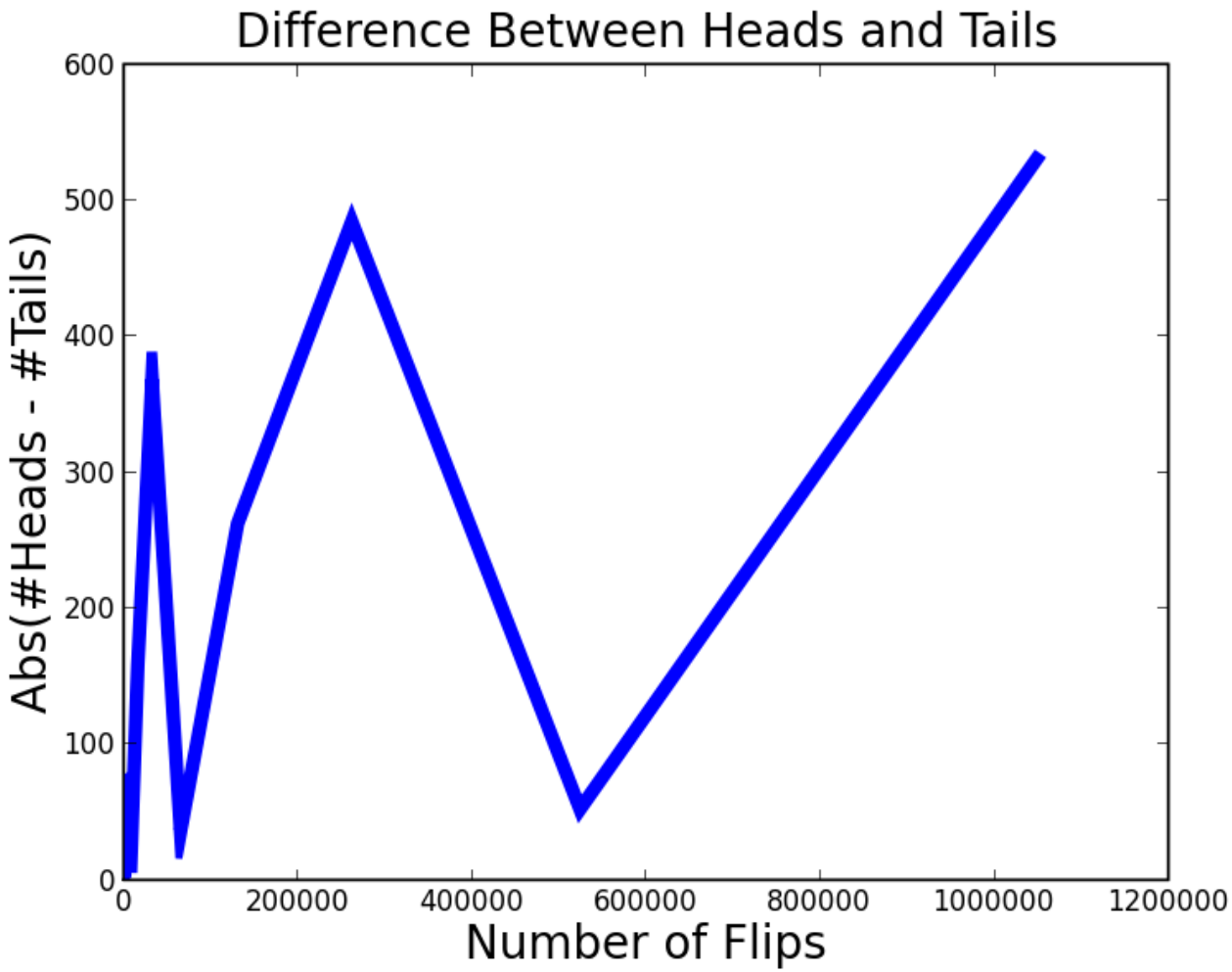
Gambler's Fallacy

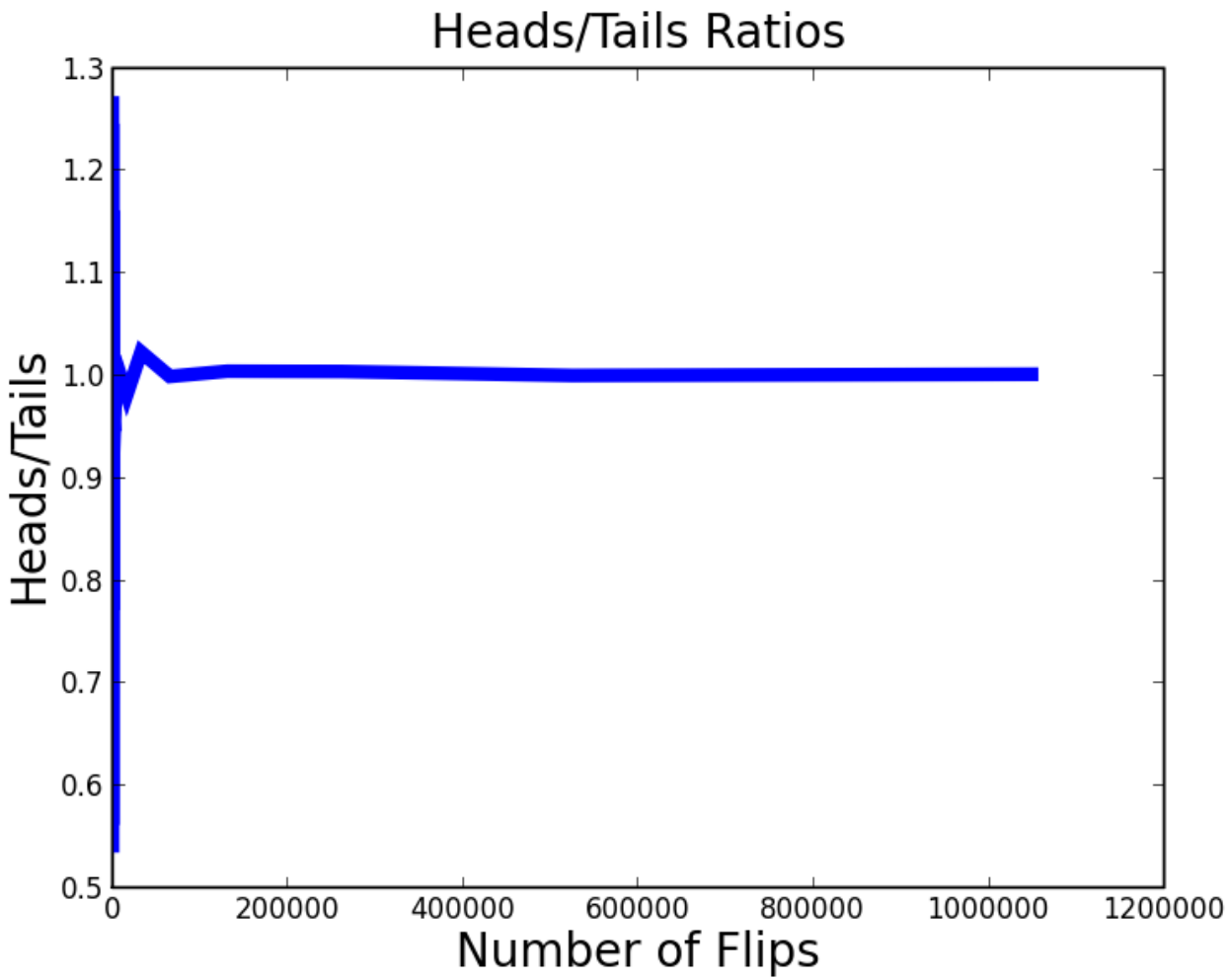
- If deviations from expected behavior occur, these deviations are likely to be evened out by opposite deviations in the future.

```
def flipPlot(minExp, maxExp):
    """Assumes minExp and maxExp positive
                integers; minExp < maxExp
       Plots results of 2**minExp to
                2**maxExp coin flips"""

    ratios = []
    diffs = []
    xAxis = []
    for exp in range(minExp, maxExp + 1):
        xAxis.append(2**exp)

...
```

```
for numFlips in xAxis:
        numHeads = 0
        for n in range(numFlips):
            if random.random() < 0.5:
                numHeads += 1
        numTails = numFlips - numHeads
        ratios.append(numHeads/float(numTails))
        diffs.append(abs(numHeads - numTails))

...
```

```
pylab.title('Difference Between Heads and Tails')
pylab.xlabel('Number of Flips')
pylab.ylabel('Abs(#Heads - #Tails)')
pylab.plot(xAxis, diffs)

pylab.figure()
pylab.title('Heads/Tails Ratios')
pylab.xlabel('Number of Flips')
pylab.ylabel('Heads/Tails')
pylab.plot(xAxis, ratios)
```


Difference Between Heads and Tails
Abs(#Heads - #Tails)
Number of Flips
0
100
200
300
400
500
600
0
200000
400000
600000
800000
1000000
1200000


Heads/Tails Ratios
Heads/Tails
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0
200000
400000
600000
800000
1000000
1200000
Number of Flips

How Much is Enough?

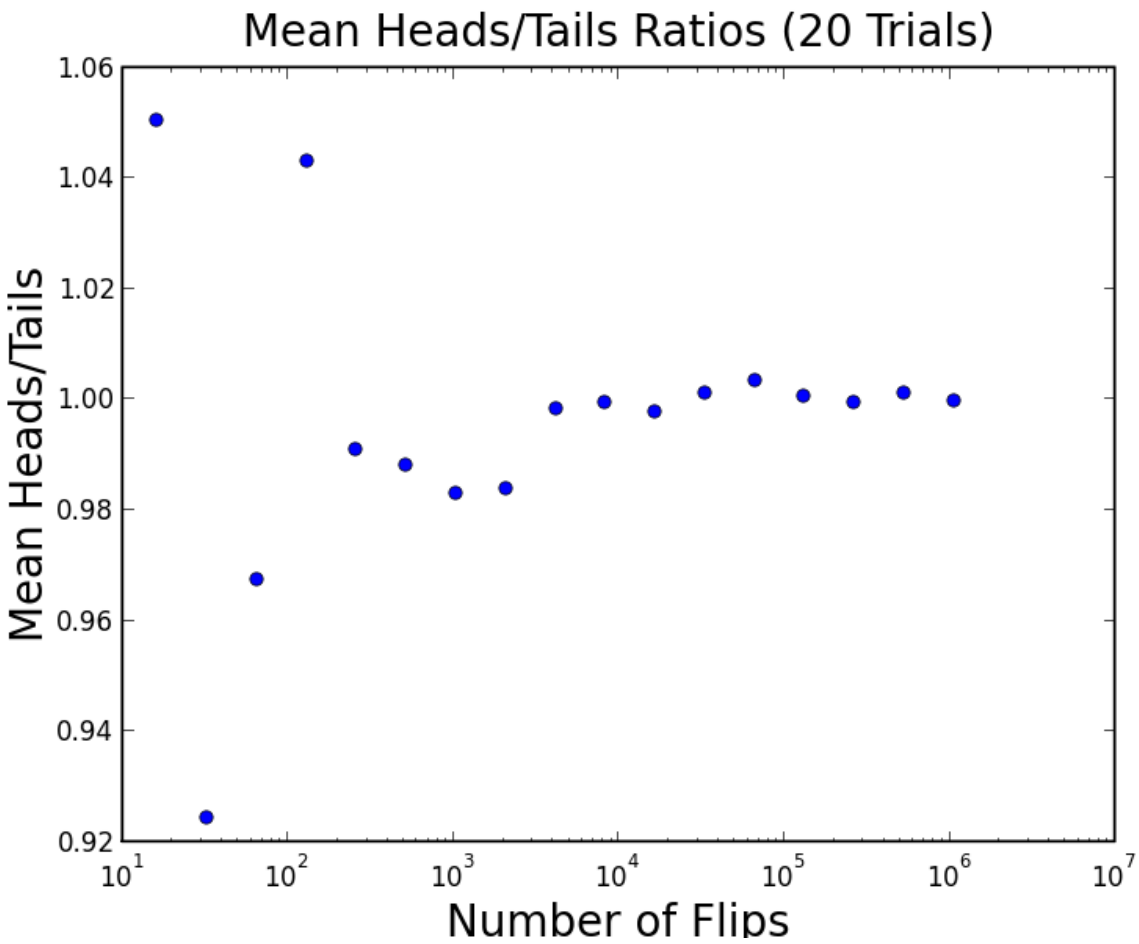
- How many samples do we need to look at in order to have a justified confidence that something that is true about the population of samples is also true about the population from which the samples were drawn?

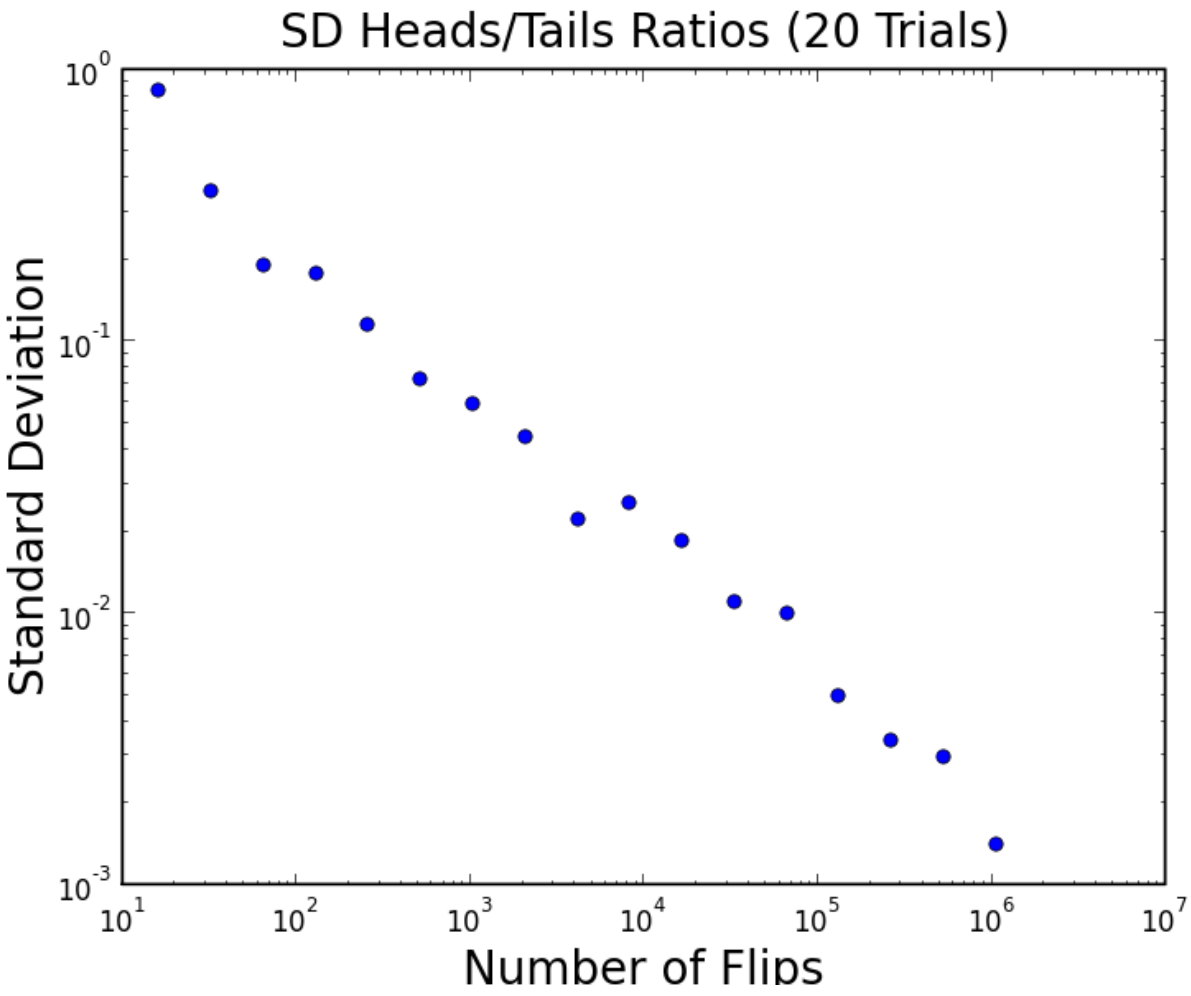
- Depends upon the **variance** in the underlying distribution

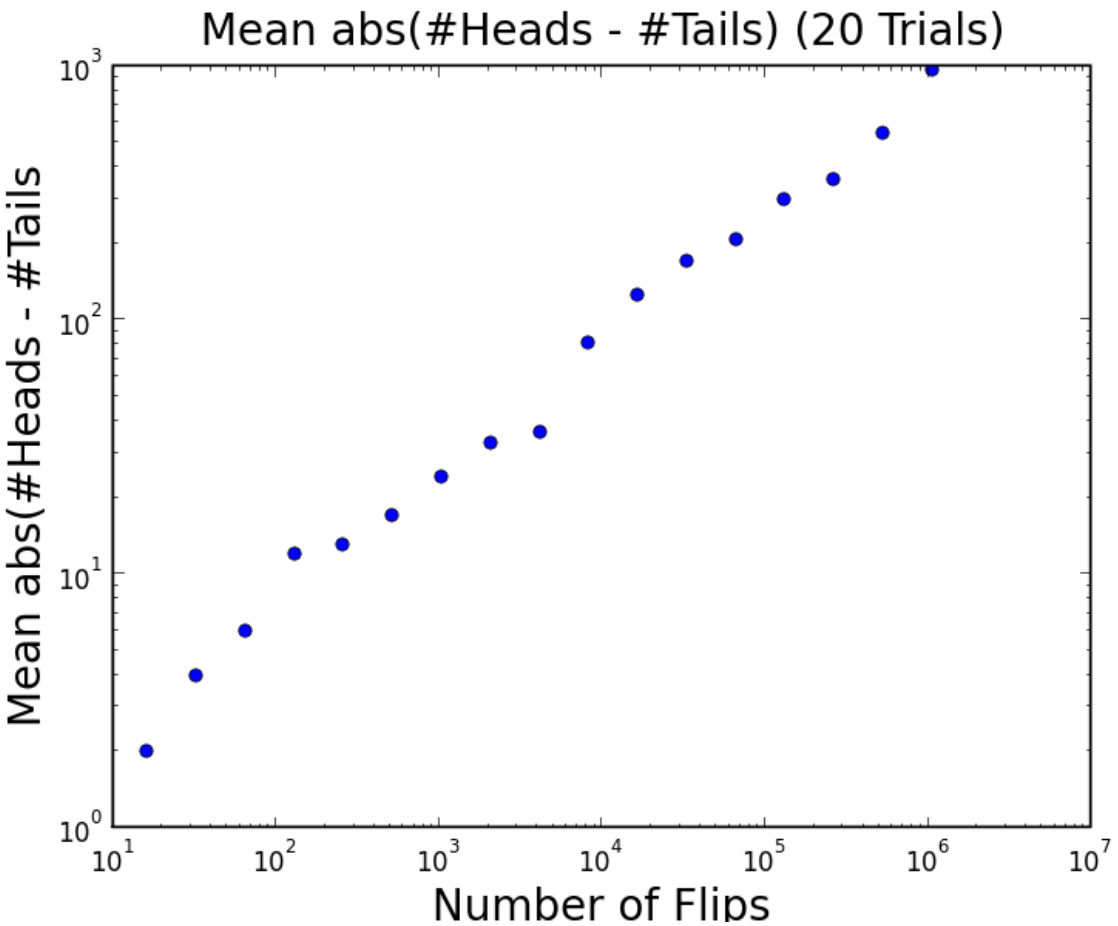
Variance

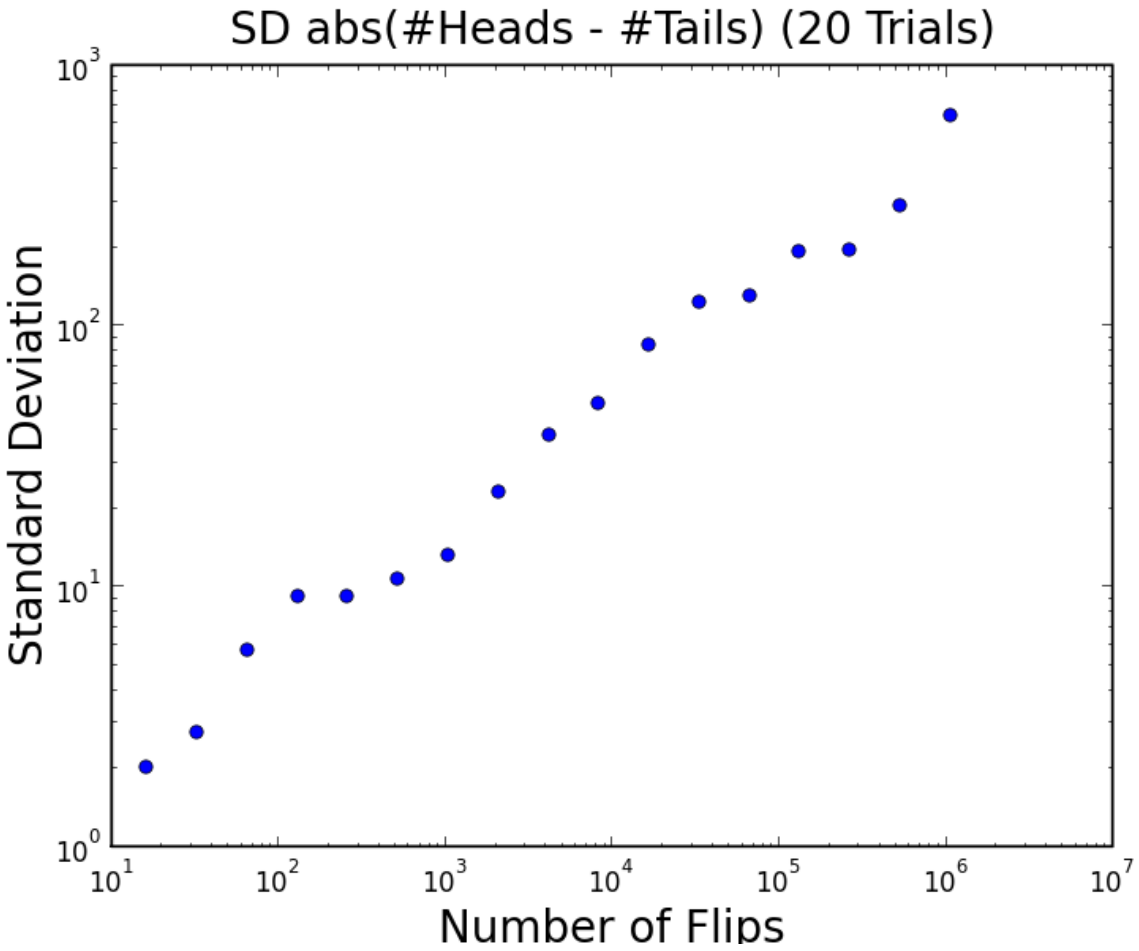
- We measure the amount of variance in the outcomes of multiple trials.

```
def stdDev(X):
    mean = sum(X)/float(len(X))
    tot = 0.0
    for x in X:
        tot += (x - mean)**2
    return (tot/len(X))**0.5
```


Mean Heads/Tails Ratios (20 Trials)
Mean Heads/Tails
Number of Flips
1.06
1.04
1.02
1.00
0.98
0.96
0.94
0.92
10^1
10^2
10^3
10^4
10^5
10^6
10^7


SD Heads/Tails Ratios (20 Trials)
Standard Deviation
Number of Flips
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
10^{7}


Mean abs(#Heads - #Tails) (20 Trials)
Mean abs(#Heads - #Tails
Number of Flips
10^0
10^1
10^2
10^3
10^1
10^2
10^3
10^4
10^5
10^6
10^7


SD abs(#Heads - #Tails) (20 Trials)
Standard Deviation
Number of Flips
10^0
10^1
10^2
10^3
10^1
10^2
10^3
10^4
10^5
10^6
10^7

Coefficient of Variation

- The **coefficient of variation** is the standard deviation divided by the mean.

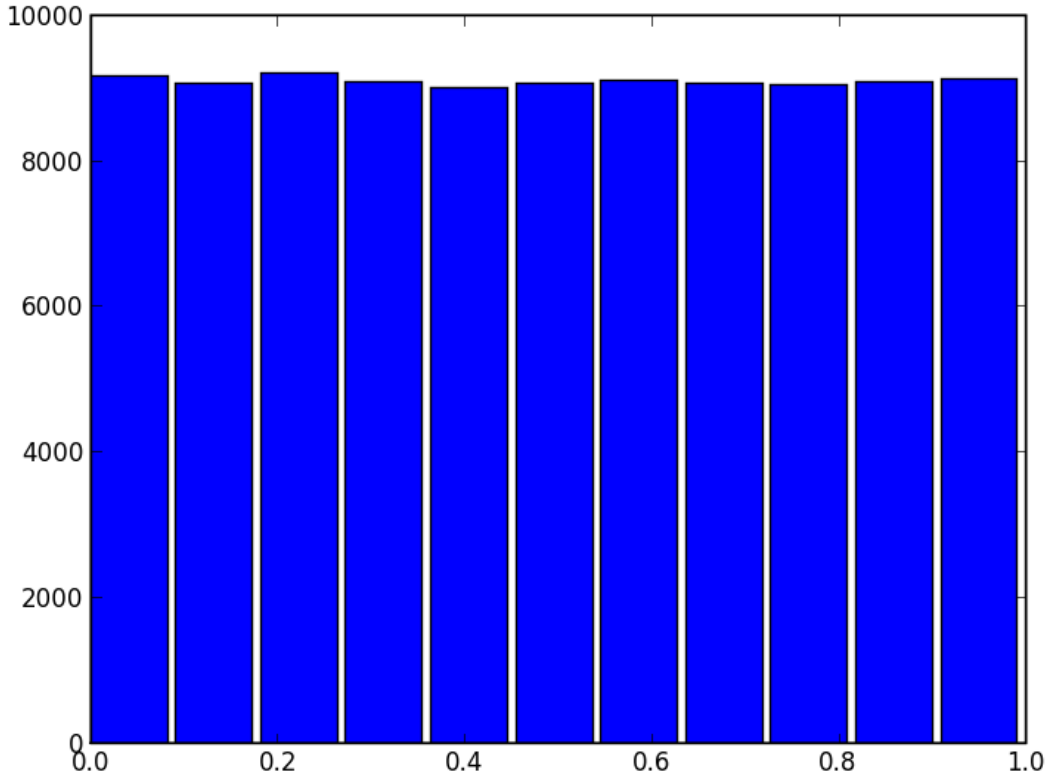
```
def CV(X):
    mean = sum(X)/float(len(X))
    try:
        return stdDev(X)/mean
    except ZeroDivisionError:
        return float('Nan')
```

```
def flip(numFlips):
def flipSim(numFlipsPerTrial, numTrials):
def labelPlot(nf, nt, mean, sd):
def makePlots(nf1, nf2, nt):
  """nt = number of trials per experiment
     nf1 = number of flips 1st experiment
     nf2 = number of flips 2nd experiment"""
```

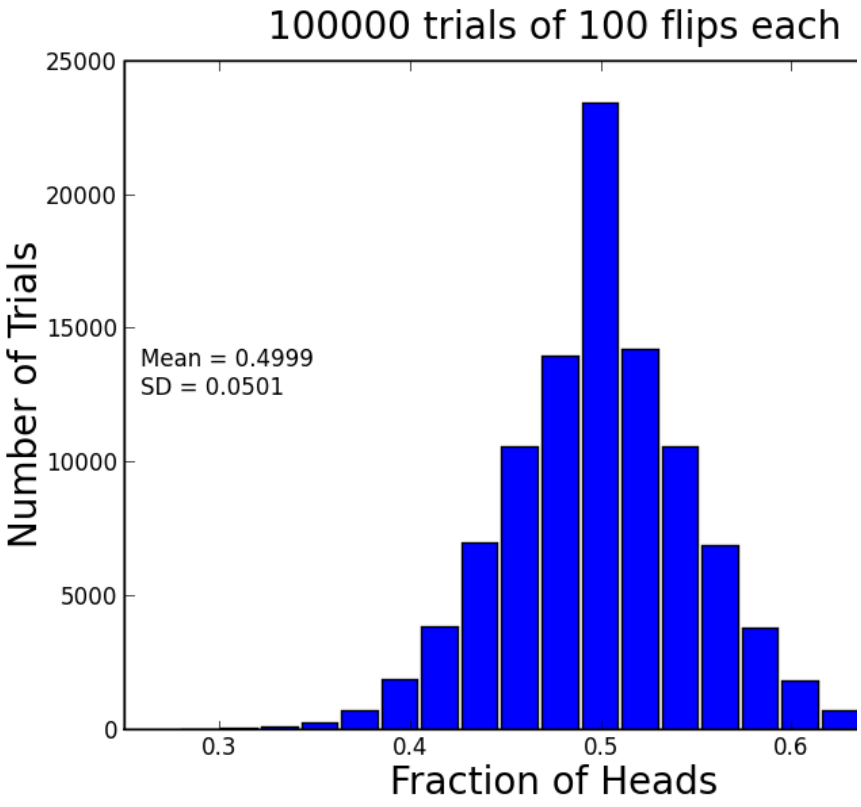
makePlots

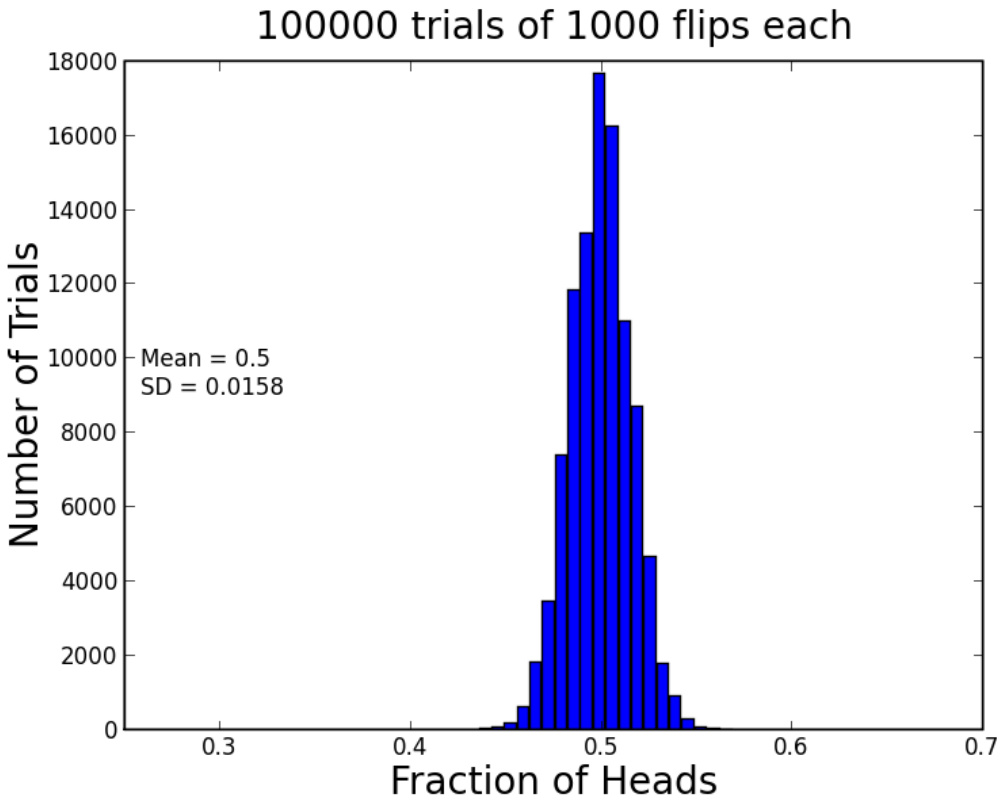
```
def makePlots(numFlips1, numFlips2, numTrials):
    val1, mean1, sd1 = flipSim(numFlips1, numTrials)
    pylab.hist(val1, bins = 20)
    xmin,xmax = pylab.xlim()
    ymin,ymax = pylab.ylim()
    labelPlot(numFlips1, numTrials, mean1, sd1)
    pylab.figure()
    val2, mean2, sd2 = flipSim(numFlips2, numTrials)
    pylab.hist(val2, bins = 20)
    pylab.xlim(xmin, xmax)
    ymin, ymax = pylab.ylim()
    labelPlot(numFlips2, numTrials, mean2, sd2)
```

```
vals = []
for i in range(100000):
    num = random.random()
    vals.append(num)
pylab.hist(vals, bins = 11)
```



```
vals = []
for i in range(100000):
    num = random.random()
    vals.append(num)
pylab.hist(vals, bins = 11)
xmin, xmax = pylab.xlim()
ymin, ymax = pylab.ylim()
print 'x-range =', xmin, '-', xmax
print 'y-range =', ymin, '-', ymax
pylab.figure
pylab.hist(vals, bins = 11)
#pylab.xlim(-1.0, 2.0)
```


100000 trials of 100 flips each
Number of Trials
Fraction of Heads
Mean = 0.4999
SD = 0.0501
0
5000
10000
15000
20000
25000
0.3
0.4
0.5
0.6


100000 trials of 1000 flips each
Number of Trials
Fraction of Heads
Mean = 0.5
SD = 0.0158
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
0.3
0.4
0.5
0.6
0.7