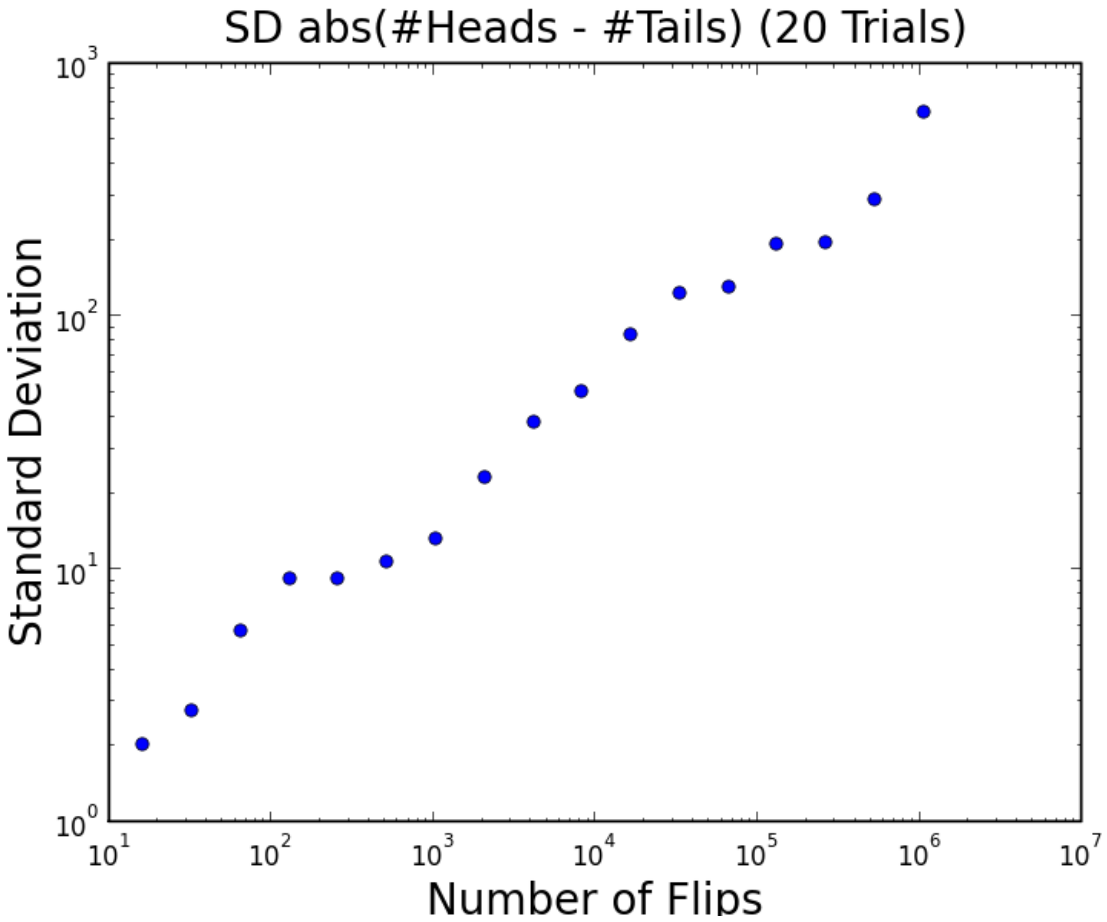
Standard Deviations and Histograms

Lecturer: John Guttag

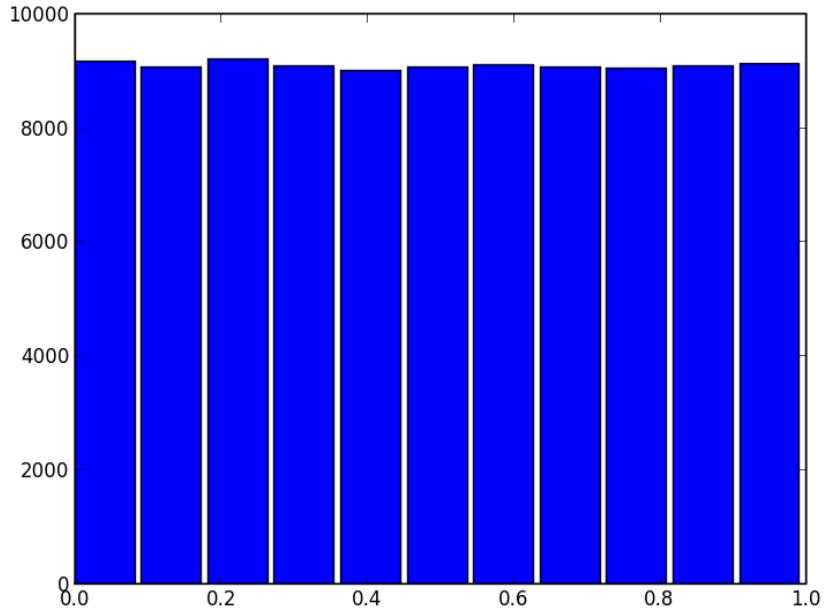

SD abs(#Heads - #Tails) (20 Trials)
Standard Deviation
Number of Flips
10^0
10^1
10^2
10^3
10^1
10^2
10^3
10^4
10^5
10^6
10^7

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

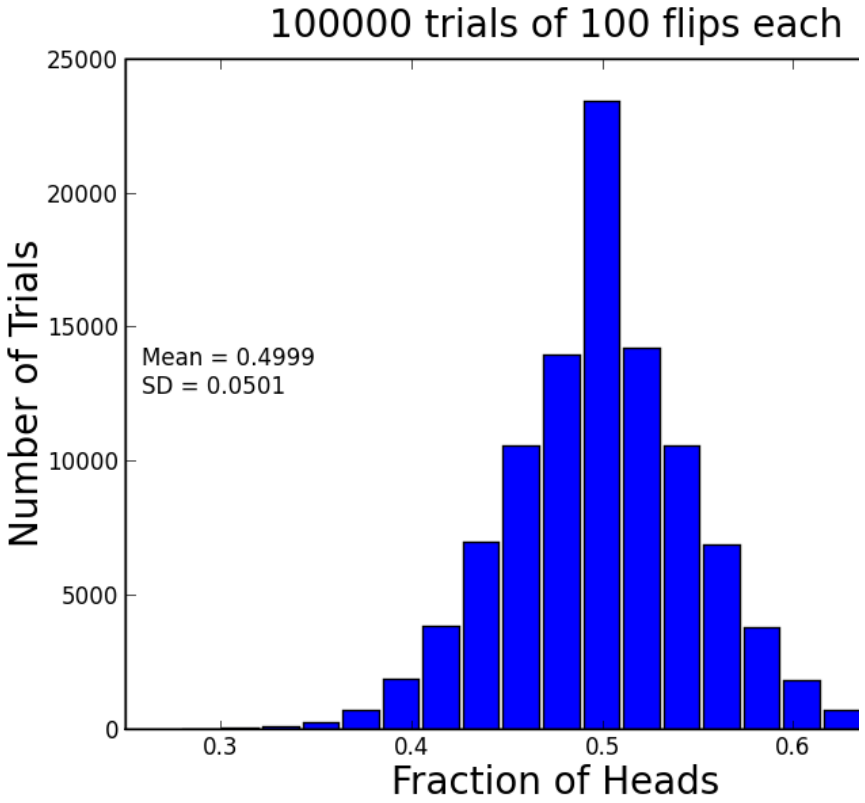
```
def flip(numFlips):

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"""
```

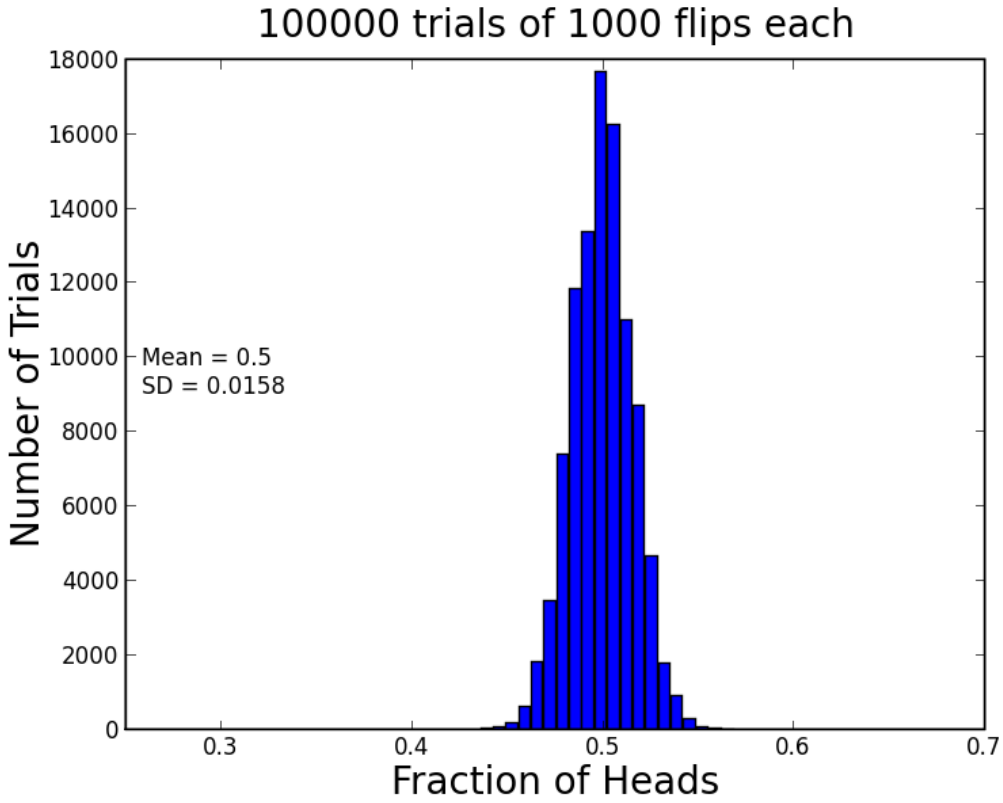
```
def makePlots(numFlips1, numFlips2, numTrials):
    val1, mean1, sd1 = flipSim(numFlips1, numTrials)
    pylab.hist(val1, bins = 20)
    xmin,xmax = pylab.xlim()
    ymin,ymax = pylab.ylim()
    labelPlot(numFlips1, numTrials, mean1, sd1)
    pylab.figure()
    val2, mean2, sd2 = flipSim(numFlips2, 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
25000
20000
15000
10000
5000
0
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
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0.3
0.4
0.5
0.6
0.7