6.00 Handout, Lecture 15
(Not intended to make sense outside of lecture)

```
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

def flipPlot(minExp, maxExp, numTrials):
    meanRatios = []
    meanDiffs = []
    ratiosSDs =  []
    diffsSDs =  []
    xAxis = []
    for exp in range(minExp, maxExp + 1):
        xAxis.append(2**exp)
    for numFlips in xAxis:
        ratios = []
        diffs = []
        for t in range(numTrials):
            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))
        meanRatios.append(sum(ratios)/numTrials)
        meanDiffs.append(sum(diffs)/numTrials)
        ratiosSDs.append(stdDev(ratios))
        diffsSDs.append(stdDev(diffs))
    pylab.plot(xAxis, meanRatios, 'bo')
    pylab.title('Mean Heads/Tails Ratios ('
                + str(numTrials) + ' Trials)')
    pylab.xlabel('Number of Flips')
    pylab.ylabel('Mean Heads/Tails')
    pylab.semilogx()
    pylab.figure()
    pylab.title('Mean abs(#Heads - #Tails) ('
                + str(numTrials) + ' Trials)')
    pylab.xlabel('Number of Flips')
    pylab.ylabel('Mean abs(#Heads - #Tails')
    pylab.plot(xAxis, meanDiffs, 'bo')
    pylab.semilogx()
    pylab.semilogy()
    ...
```

```
def flip(numFlips):
    heads = 0.0
    for i in range(numFlips):
        if random.random() < 0.5:
            heads += 1.0
    return heads/numFlips

def flipSim(numFlipsPerTrial, numTrials):
    fracHeads = []
    for i in range(numTrials):
        fracHeads.append(flip(numFlipsPerTrial))
    return fracHeads

def labelPlot(nf, nt, mean, sd):
    pylab.title(str(nt) + ' trials of '
                + str(nf) + ' flips each')
    pylab.xlabel('Fraction of Heads')
    pylab.ylabel('Number of Trials')
    xmin, xmax = pylab.xlim()
    ymin, ymax = pylab.ylim()
    pylab.text(xmin + (xmax-xmin)*0.02, (ymax-ymin)/2,
               'Mean = ' + str(round(mean, 6))
               + '\nSD = ' + str(round(sd, 6)))

def makePlots(nf1, nf2, nt):
    """nt = number of trials per experiment
       nf1 = number of flips 1st experiment
       nf2 = number of flips 2nd experiment"""
    fracHeads1 = flipSim(nf1, nt)
    mean1 = sum(fracHeads1)/float(len(fracHeads1))
    sd1 = stdDev(fracHeads1)
    pylab.hist(fracHeads1, bins = 20)
    xmin,xmax = pylab.xlim()
    ymin,ymax = pylab.ylim()
    labelPlot(nf1, nt, mean1, sd1)
    pylab.figure()
    fracHeads2 = flipSim(nf2, nt)
   . . .

def poll(n, p):
    votes = 0.0
    for i in range(n):
        if random.random() < p/100.0: votes += 1
    return votes

def testErr(n = 1000, p = 46.0, numTrials = 1000):
    results = []
    for t in range(numTrials):
        results.append(poll(n, p))
    print 'std = ' + str((stdDev(results)/n)*100) + '%'
    results = pylab.array(results)/n
    pylab.hist(results)
    pylab.xlabel('Fraction of Votes')
    pylab.ylabel('Number of Polls')
```

6.00SC Introduction to Computer Science and Programming
Spring 2011