Prob Set 2: due Thu night 22 Feb

Midterm date (poll) closes in 4 day(s)

A total of 77 vote(s) in 67 hours

- 21 (27% of users) - Mon 19 Mar (evening)
- 38 (49% of users) - Tue 20 Mar (class time)
- 20 (26% of users) - Wed 21 Mar (evening)
- 46 (60% of users) - Thu 22 Mar (class time)
"p-hacking"

https://xkcd.com/882/
Probability \((r \text{ successes in } N \text{ trials}) = \binom{N}{r} p^r (1 - p)^{N-r}\)

Overall probability for \(r\) successes is a competition between

\(\binom{N}{r} \): has maximum at \(r \sim N/2\)

\(p^r (1 - p)^{N-r} \): (large for small \(r\) when \(p < 1/2\); large for large \(r\) when \(p > 1/2\))

Consider rolling a standard six-sided die, with a roll of 6 considered a success, so \(p = 1/6\)

(See figures in notebook showing \(\binom{N}{r} p^r (1 - p)^{N-r}\) for \(N = 1, 2, 4, 10, 40, 80, 160, 320\) trials, with the number of successes \(r\) plotted along the horizontal axis for each value of \(N\).)

For a larger number \(N\) of trials, the distribution of expected number of successes becomes more narrowly peaked and more symmetrical about a fractional distance \(r = N/6\).
plt.figure(figsize=(12,16))
for i,N in enumerate((1,2,4,10,40,80,160,320)):
    plt.subplot(4,2,i+1)
    plt.title('Probability of r sixes in {} trials'.format(N))
    plt.xlabel('Number of sixes')
    plt.ylabel('Probability')
    plt.bar(range(N+1), [bern_prob(N,m) for m in range(N+1)])

from scipy.misc import comb
def bern_prob(N,m,p=1/6.):
    return comb(N,m) *p**m *(1-p)**(N-m)

plt.figure(figsize=(12,16))
for i,N in enumerate((1,2,4,10,40,80,160,320)):
    plt.subplot(4,2,i+1)
    plt.title('Probability of r sixes in {} trials'.format(N))
    plt.xlabel('Number of sixes')
    plt.ylabel('Probability')
    plt.bar(range(N+1), [bern_prob(N,m) for m in range(N+1)])
plt.figure(figsize=(18,12))
for i,N in enumerate((1,2,4,10,40,80,160,320)):
    plt.subplot(3,3,i+1)
    plt.title('Probability of r sixes in {} trials'.format(N))
    plt.xlabel('Number of sixes')
    plt.ylabel('Probability')
    plt.bar(range(N+1),[bern_prob(N,m) for m in range(N+1)])
    mean = Np = \sqrt{Np(1-p)} = 0.17
    stdev = \sqrt{Np(1-p)} = 0.37
    mean = Np = 0.33
    stdev = \sqrt{Np(1-p)} = 0.53
    mean = Np = 0.67
    stdev = \sqrt{Np(1-p)} = 0.75
    mean = Np = 1.67
    stdev = \sqrt{Np(1-p)} = 1.18
    mean = Np = 6.67
    stdev = \sqrt{Np(1-p)} = 2.36
    mean = Np = 13.33
    stdev = \sqrt{Np(1-p)} = 3.33
    mean = Np = 26.67
    stdev = \sqrt{Np(1-p)} = 4.71
    mean = Np = 53.33
    stdev = \sqrt{Np(1-p)} = 6.67
plt.figure(figsize=(18,12))
for i,N in enumerate((1,2,4,10,40,80,160,320)):
    plt.subplot(3,3,i+1)
    plt.title('Probability of r sixes in {} trials'.format(N))
    plt.xlabel('Number of sixes')
    plt.ylabel('Probability')
    plt.bar(range(N+1),[bern_prob(N,m) for m in range(N+1)])
    mean = Np = 0.17
    stdev = \sqrt{Np(1-p)} = 0.37
    mean = Np = 0.33
    stdev = \sqrt{Np(1-p)} = 0.53
    mean = Np = 0.67
    stdev = \sqrt{Np(1-p)} = 0.75
    mean = Np = 1.67
    stdev = \sqrt{Np(1-p)} = 1.18
    mean = Np = 6.67
    stdev = \sqrt{Np(1-p)} = 2.36
    mean = Np = 13.33
    stdev = \sqrt{Np(1-p)} = 3.33
    mean = Np = 26.67
    stdev = \sqrt{Np(1-p)} = 4.71
    mean = Np = 53.33
    stdev = \sqrt{Np(1-p)} = 6.67
Number of sixes for \( n \) rolls of a fair die, for \( n \) from 1 to 2048. Each "experiment" is repeated 10,000 times to collect statistics. Observed number of heads is near \( \mu = n/6 \), but with some variance about the mean value. The fractional standard deviation \( \sigma / \mu \) decreases as \( 1 / \sqrt{n} \) as \( n \) increases.

(see https://courses.cit.cornell.edu/info2950_2014fa/bsims.html)
Number of heads for n flips of a fair coin, for n from 1 to 2048. Each "experiment" is repeated 10,000 times to collect statistics. Observed number of heads is near $\mu=n/2$, but with some variance about the mean value. The fractional standard deviation ($\sigma/\mu$) decreases as $1/\sqrt{n}$ as n increases.

(see https://courses.cit.cornell.edu/info2950_2014fa/bsims.html)