Chris Piech
CS109

Section #6
Nov 7, 2018

Section #6: Samples Solution

---

1. **Warmup**:

   - Population variance, σ^2: true variance of a population (or random variable).
   - Sample variance, S^2: unbiased estimate of true variance based on a random subsample.
   - Variance of sample mean, $\text{Var}(\bar{X})$: Amount of spread in the estimation of the true mean.

2. **Beta Sum**:

   By the Central Limit Theorem, the sum of equally weighted IID random variables will be Normally distributed. We calculate the expectation and variance of X_i using the beta formulas:

$$
\begin{aligned}
E(X_i) &= \frac{a}{a+b} && \text{Expectation of a Beta} \\
&= \frac{3}{7} \approx 0.43 \\
\text{Var}(X_i) &= \frac{ab}{(a+b)^2(a+b+1)} && \text{Variance of a Beta} \\
&= \frac{3 \cdot 4}{(3+4)^2(3+4+1)} \\
&= \frac{12}{49 \cdot 8} \approx 0.03
\end{aligned}
$$

$$
\begin{aligned}
X &\sim N(\mu = n \cdot E[X_i], \sigma^2 = n \cdot \text{Var}(X_i)) \\
&\sim N(\mu = 43, \sigma^2 = 3)
\end{aligned}
$$

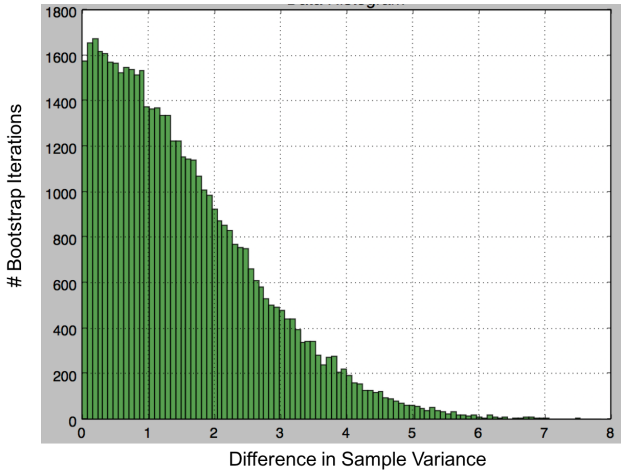
3. **Variance of Height among Island Corgis**:

```
def bootstrap(pop1, pop2):
   # make the universal population
   totalPop = copy.deepcopy(pop1)
   totalPop.extend(pop2)

   # Run a bootstrap experiment
   countDiffGreaterThanObserved = 0
   print 'starting bootstrap'
   for i in range(50000):
      # resample and recalculate the statistic
```

```
        sample1 = resample(totalPop, len(pop1))
        sample2 = resample(totalPop, len(pop2))
        sampleStat1 = calcSampleVariance(sample1)
        sampleStat2 = calcSampleVariance(sample2)
        diff = abs(sampleStat2 - sampleStat1)
        # count how many times the statistic is more extreme
        if diff >= 3:
            countDiffGreaterThanObserved += 1
    # compute the p-value
    p = float(countDiffGreaterThanObserved) / 50000
    print 'p-value:', p
```

For this data, the two-tailed (eg using absolute value) test returns a null hypothesis probability **p = 0.12**. There is a pretty decent chance that the observed difference in sample variance was random chance – and it doesn't fall under what scientists often call "statistically significant." Here is a histogram of all the diff values from the bootstrap experiment:



4. Traffic Lights:

a. The key insight here is that since time = $\frac{\text{dist}}{\text{speed}}$, we can find a CDF for time. If speed $\sim N(\mu = 0.16667\frac{\text{mi}}{\text{min}}, \sigma = 0.0333\frac{\text{mi}}{\text{min}})$ and distance = 2 mi we know that $\frac{1}{\text{time}} \sim N(\mu = 0.0833333 \text{ min}^{-1}, \sigma = 0.016667 \text{ min}^{-1})$. We represent $\frac{1}{\text{time}}$ as a Gaussian random variable τ.
$P(\text{time} < 10 \text{ min}) = P(\tau > 0.1 \text{ min}^{-1}) = 1 - \Phi(\frac{0.1-0.83333}{0.016667}) = 1 - \Phi(1) = 0.1587.$

b. We can use the Central Limit Theorem here. Our wait time PDF for one traffic light is 0.5 if $x = 0$ and 0.125 if $0 < x < 4$ (the uniform distribution whose probability sums to 0.5). Thus we have a total wait time $W \sim N(\mu = 10min, \sigma = \sqrt{\frac{50}{3}})$. $P(W < 8) = \Phi(\frac{8-10}{\sqrt{\frac{50}{3}}}) = \Phi(-.4899) = 0.3121.$

c. Where T = total time, T_W = wait time at lights, S = biking speed, F_S = CDF for biking speed:

$$
\begin{aligned}
P(T < 20 | T_W = w) &= P(S > \frac{2}{20 - w}), \text{ so:} \\
P(T < 20) &= \int_{w=-\infty}^{\infty} f_W(w) P(T < 20 | T_W = w) dw \\
&= \int_{w=0}^{40} f_W(w) P(S > \frac{2}{20 - w}) dw \\
&= \int_{w=0}^{40} f_W(w)(1 - F_S(\frac{2}{20 - w})) dw \\
&\approx \sum_{w=0}^{19} f_W(w)(1 - F_S(\frac{2}{20 - w})) dw \\
&\approx 0.3054
\end{aligned}
$$

Code for Riemann approximation:

```
from scipy.stats import norm
from math import sqrt

def f_W(w):
        return norm.pdf(w, 10, sqrt(50.0/3))

def F_S(s):
        return norm.cdf(s, 0.1666666667, 0.03333333)

# add rects of width 1
sum = 0
for w in range(20):
        sum += f_W(w) * (1 - F_S(2.0 / (20 - w)))
        print w, sum
```