Chris Piech
CS 109

Section #7

Section 7: Uncertainty Theory Part 2

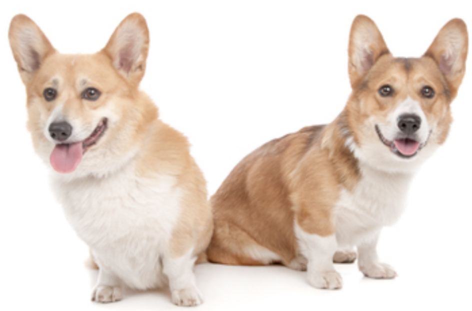
1. **Warmup**: *populations vs. samples*

   What is the difference between the population variance, σ^2, and sample variance, S^2? What is the difference between sample variance, S^2, and variance of the sample mean, $\text{Var}(\bar{X})$?

- 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. **Variance of Height among Island Corgis**

   A colleague has collected samples of heights of corgis that live on two different islands, A and B. The colleague collects 50 samples from each island.



The sample mean is the same for both groups: 10 inches. However, island B has a **sample variance** that is 3.1 in^2 **greater** than island A. The colleague wants to make the claim that island B corgis have a significantly higher spread of heights than island A corgis. You are skeptical. It's possible that heights are identically distributed across both islands, and the observed difference in variance is just a result of chance and small sample size, i.e. the **null hypothesis**.

Write code that uses **bootstrapping** to calculate the probability of the null hypothesis. Here is the data. Each number is the height, in inches, of an independently sampled corgi:

Island A Corgi Heights (S^2 = 6): [13, 12, 7, 16, 9, 11, 7, 10, 9, 8, 9, 7, 16, 7, 9, 8, 13, 10, 11, 9, 13, 13, 10, 10, 9, 7, 7, 6, 7, 8, 12, 13, 9, 6, 9, 11, 10, 8, 12, 10, 9, 10, 8, 14, 13, 13, 10, 11, 12, 9]

Island B Corgi Heights (S^2 = 9.1): [8, 8, 16, 16, 9, 13, 14, 13, 10, 12, 10, 7, 14, 8, 13, 14, 7, 13, 7, 9, 4, 11, 7, 12, 8, 9, 12, 8, 11, 10, 12, 6, 10, 15, 11, 12, 3, 8, 11, 10, 10, 8, 12, 9, 11, 6, 7, 10, 9, 7]

```
def bootstrap(pop1, pop2):
    # make the universal population, combining the two lists
    totalPop = pop1 + pop2

    # Run bootstrapping
    countDiffGreaterThanObserved = 0
    for i in range(10000):
        # resample and recalculate the statistic
        sample1 = resample(totalPop, len(pop1), replace=True)
        sample2 = resample(totalPop, len(pop2), replace=True)

        sampleStat1 = calcSampleVariance(sample1)
        sampleStat2 = calcSampleVariance(sample2)

        diff = abs(sampleStat2 - sampleStat1)

        # count how many times the statistic is more extreme
        if diff >= 3.1:
            countDiffGreaterThanObserved += 1

    # compute the p-value
    p = countDiffGreaterThanObserved / 10000
    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."

How would this calculation be different if you were interested in looking at the statistical significance of the difference in sample mean? 95th percentile?

3. **Binary Tree**: Consider the following function for constructing binary trees:

```
def random_binary_tree(p):
    """
    Returns a dictionary representing a random binary tree structure.
    The dictionary can have two keys, "left" and "right".
    """
    if random_bernoulli(p):   # returns true with probability p
        new_node = {}  # initialize one new node
        new_node["left"] = random_binary_tree(p)
        new_node["right"] = random_binary_tree(p)
        return new_node
    else:
        return None
```

The `if` branch is taken with probability p (and the `else` branch with probability $1 - p$). A tree with no nodes is represented by `None`; so a tree node with no left child has `None` for the `left` field (and the same for the right child).

Let X be the number of nodes in a tree returned by `random_binary_tree(p)`. You can assume $0 < p < 0.5$. What is $E[X]$, in terms of p?

The number of nodes in the tree depends on whether or not the if statement is true or false. It is true with probability p and false with probability 1 - p. This suggests that in order to find $E[X]$, we need to define a background random variable, Y, corresponding to the result of `random_bernoulli(p)`, where $P(Y = 1) = p$ and $P(Y = 0) = 1 - p$. Then, we can apply the Law of Total Expectation:

$$E[X] = p \cdot E[X \mid \text{if}] + (1 - p)E[X \mid \text{else}]$$

$$E[X] = p \cdot E[X \mid Y = 1] + (1 - p)E[X \mid Y = 0]$$

$E[X \mid Y = 0] = 0$ because if the `else` statement is executed, we return a tree with no nodes.

Let X_1 and X_2 be number of nodes returned by the left and right calls to `random_binary_tree`. We can write $E[X \mid Y = 1]$ as $E[1 + X_1 + X_2]$ because that represents the total number of nodes that are added to the tree if the `if` statement is true. Then, because the recursive call is identical to the original function call, we know that $E[X_1] = E[X_2] = E[X]$, so

$$E[X \mid Y = 1] = E[1 + X_1 + X_2] = 1 + E[X] + E[X] = 1 + 2E[X]$$

Putting this all together:

$$\begin{aligned}
E[X] &= p \cdot E[X \mid Y = 1] + (1 - p)E[X \mid Y = 0] \\
&= p \cdot E[1 + X_1 + X_2] + (1 - p) \cdot 0 \\
&= p \cdot (1 + 2E[X]]) \\
(1 - 2p)E[X] &= p && \text{(getting } E[X] \text{ alone on the LHS)} \\
E[X] &= \frac{p}{1 - 2p}
\end{aligned}$$

Extra Challenge Q: Why did we need to assume that p is less than 0.5?

4. **Entropy & Name2Age**

   See the Colab notebook at: `https://web.stanford.edu/class/cs109/section/7/`

   Solutions are available through a link on the course website page.