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, $\text{Var}(\bar{X})$?

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 A corgis have a significantly higher spread of heights than island B 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]

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?

4. **Entropy & Name2Age**

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