Statistics Module

Homework 1.

Instructions: *This is an optional homework; turning it in or not will not affect your grade. If you do turn it in on time, we will grade it with feedback. Solutions will not be posted. Your answers must be handwritten; you can either turn in a hardcopy to Prof. Cowley's office (Freeman 113) or e-mail Prof. Cowley, the TAs, and Razan with a scan. Any use of resources is permitted.*

This homework is due by **Friday, September 12, 11:59pm**.

Problem 1.

Compute the expected value for the exponential distribution:
$\text{Exp}(x; \lambda) = \lambda e^{-\lambda x}$ (Hint: Integration by parts.)

Problem 2.

You are taking a medical test for a serious but rare disease. The test is not perfectly reliable, returning a positive result 99% of the time if the patient has the disease. If the subject has no disease, the test will return a negative result 80% of the time. Only 1 out of 1,000 people have the disease.
You take the test, and it returns positive. How likely are you that you have the disease?

Problem 3.

You are a neuroscience graduate student, training a mouse to lick for reward every time you optogenetically stimulate a neuron. You want to estimate the mouse's probability of licking $p(\text{lick})$ with as few stimulations as possible. In other words, you want to stop recording behavior when you are confident in your estimate of $p(\text{lick})$, stopping when there is a 90% chance that your estimate is only 0.01 off from the true probability.

Part A:
Let's say for 50 stimulations, the mouse licked 10 times. Should you stop recording?

Part B:
How many stimulations do you expect to record behavior when the true $p(\text{lick}) = 0.5$?

Part C:
Plot the minimum number of stimulations needed (y-axis) for different values of the true $p(\text{lick})$ (x-axis; 0, 0.1, 0.2,...,1.0).

Part D:
Write pseudocode of your algorithm to stop recording given an observed number of heads and tails.

```
def stop_record(num_heads, num_tails):
```

```
return: True if stop_flag==True else False
```

Problem 4.

Write pseudocode of a function that performs a permutation test to compute the p-value of a difference of means, where X is a 1-d array of N samples and Y is a 1-d array of M samples.

```
def permutation_test(X_data, Y_data, num_runs):
```

```
return: p-value
```

Problem 5.

You are interested in finding a cure for cancer. You plan to test the efficacy of 1,000 drugs. Draw a plot of density (y-axis) versus p-value (x-axis) that you'd expect to see if none of the drugs were effective. Then, draw a plot of p-value (y-axis) versus test index, sorted (x-axis) that you'd expect to see if only one drug was truly effective.