Lab #1 – how accurate is testing?

Goals of the lab:

In this lab, we'll

- Learn about the statistics involved in testing – for Covid or for any other disease, and why it becomes more difficult to trust your test results if the disease is rare.
- Write simple Python code to implement test analysis. It requires using variables, simple arithmetic expressions and printing.

The parameters of testing

There are a few important numbers that describe tests. The first is called *sensitivity*. It tells what fraction of sick people do in fact test positive. Of course, we would like our test to detect *all* sick people (i.e., sensitivity=1), but that's rarely possible in real life. When our sensitivity is less than one, that means we get some *false negatives* – i.e., the test gives a negative result but it's mistaken.

The second important number is the *specificity*. It tells what fraction of healthy people test negative. Of course, we want to have specificity=1, but again that's unlikely in real life. When specificity<1, we get the occasional *false positive* – a healthy person winds up mistakenly testing positive.

A well-run site using a high-quality Covid test (like we have at Tufts) may easily have sensitive and specificity of .95 or higher. Does that mean our results are quite accurate? That's what this lab will tell us.

How the statistics work

Let's take an example. Assume that there are 10000 people in our world and 1% of them are infected. Assume that we use a test that has 98% sensitivity and 97% specificity. Then consider the following table:

	Healthy	Infected
Test positive	297	98
Test negative	9603	2

We first compute that 1% of our 10000 people – i.e., 100 – are infected. Of the 100 infected people, our 98% sensitivity says that 98 people will correctly test positive and 2 will mistakenly test negative (i.e., a false negative). Of our 9900 healthy people, the 97% specificity says that 9603 will correctly test negative and 297 will result in a false positive.

Looking at these numbers, we see that 98+297=395 people test positive. Of these, 98 are true positives (i.e., are actually infected). Thus, if you receive a positive test, then your odds of being infected are 98/395 or roughly 24%. Surprisingly low! (Of course, if, in addition to receiving a positive test, you also feel lousy, are running a fever and have lost your sense of smell then your odds of being infected are probably a whole lot higher).

If somebody tells you to spend two weeks of your life quarantining based on having a 24% chance of being infected, you might reasonably push back. Perhaps you should ask for a retest?

Let's assume that all of the 395 people who test positive ask for a retest. In this new sub-world, we have 395 people and we know that 98 of them are actually positive. We can use this data to make a new table, assuming that the sensitivity and specificity numbers are unchanged. This will indeed be part two of the assignment.

Also note that Chapter 9 of *Calling Bullshit* talks about the "prosecutor's fallacy," which is quite related to this lab.

Details

Your scripts should start by defining SENSITIVITY=.98 and SPECIFICITY=.97, and also N_PEOPLE=200000. The number of infected people should remain at 1% of the total population.

Then do computations similar to the ones described above, but using the variables SENSITIVITY and SPECIFICITY rather than the actual numbers .98 and .97. After those computations, print out a line like the following (your numbers will be different):

> Of the 395 total positives, 25% are true and 75% are false.

Next, retest all of the people who tested positive, and print out a line like (again, your numbers will be different):

> Retest results: of the 105 total positives, 92% are true.

You will notice below that 10% of your grade comes from "code clarity." Essentially, this means following the guidelines in Think Python 2e for variable names and comments, following our guidelines of using all caps for constants, and generally following the rule to "keep everything as simple as possible but no simpler."

Questions to answer

1. You will have noted that the retest was quite effective. Intuitively, why was this? Hint – do you think the retest numbers would have look so good if we had retested *everybody* rather than just those people who originally tested positive?
2. Explain how Chapter 9 of Calling Bullshit (specifically the section on the "Prosecutor's fallacy") is really the same issue as this lab.

Challenge problems

Each lab will include a few ungraded challenge problems intended to stretch your coding and analytical skills. If you found this lab easy, take some time to work on these!

1. Given the sensitivity and specificity defined above, how many tests do you need in order to be 99.99% sure that you don't have Covid?
2. Suppose you get tested every three days, and the test has 98% sensitivity. After a few negative tests, you'll be very sure you don't have Covid. But suppose that each day you also have a 1% chance of getting infected without knowing it. What is your limit of certainty? That is, on any given day, how sure can you be that you do not have the illness? You can solve this analytically or with code; we'd love to see both!
3. Conveniently, the numbers in this simulation worked out so that a whole number of people tested positive and negative. But suppose that our initial sample had some other number, giving a fractional number of positive tests. Clearly it doesn't make any sense for 120.78 people to test positive, so what is the right thing to do with these numbers?

How would you modify the simulation so that it reflects the reality that individuals test either positive or negative?

What to turn in:

- Your *testing.py* and a .pdf file with the answers to the questions.

***Grading*:**

- Code correctness: 50 pts
- Code clarity: 10 pts (as per the note above)
- Questions: 20 pts each