**Hypothesis Tests for Randomness**

Now we’ll consider random sequences of the outcomes of a binary event. We’ll take a coin flip as our illustrative example. The sequence of outcomes would be something like,

HTHHTHTTTHTHHTHHHTTHT

We’d like to see if the outcomes are purely random, or if there is evidence for non-randomness in the sequence. We’ll presume still, that the probability of each outcome is p = 0.5. And so by ‘random’ we don’t mean that each outcome is not equiprobable. For instance, we could get the following sequence:

HHHHHHHHHHTTTTTTTTTTTT

which would be consistent with p = 0.5, but not likely a random sequence.

**Null Hypothesis**

The null hypothesis is that the sequence is random, and implicitly, the probability of each possibility is the same, p = 0.5.

**Alternative Hypothesis**

This would be that the sequence is not random.

**The test**

Now I’d like to consider hypothesis tests regarding random sequences. We’ll consider binary sequences, like a coin flip, where ostensibly each outcome is equally likely. One such sequence would be this.

HTHHTHTTTHTHHTHHHTTHT

Our test for randomness will concern the number of runs in the sequence. A run is a sequence of consecutive identical values. In the example above we have 7 runs of T’s and 7 runs of H’s. The minimum number of total runs we can have is rmin = 1, like this.

HHHHHHHHHHHHHHHHHHH

And next we can have 2 total runs,

TTTTTTTTHHHHHHHHHHHH

And we can have 3, 4, etc. total runs. We’ll note that the number of runs of H’s and T’s can differ from each other by no more than 1. For instance, below we have 7 runs of H, and 8 runs of T. Or 7 runs of H and 7 runs of T. Or 7 runs of H and 6 runs of T.

THTHHTHTTTHTHHTHHHTTHT

HTHHTHTTTHTHHTHHHTTHT

HTHHTHTTTHTHHTHHHTTH

The maximum number of runs we can have is when the letters alternate. We can have 10 runs of H and 11 runs of T. Or say 10 runs of H and 10 runs of T. Or 10 runs of H and 9 runs of T. So in general, the maximum number of runs we can have is the same as the number of outcomes rmax = n.

THTHTHTHTHTHTHTHTHTHTHT

THTHTHTHTHTHTHTHTHTHTH

HTHTHTHTHTHTHTHTHTHTH

Since the outcomes (T, or H) are equally likely, every equal lengthed sequence is equally likely, regardless of the number of runs it contains. Nonetheless sequences with a large or small number of runs are much less likely than sequences with a medium number of runs because there are far less ways to create sequences with large or small runs than there are ways to create those with medium number of runs. Basically, the multiplicity of large or small run sequences is smaller than that of medium run sequences. This is clearly related to entropy. Now let’s say we have r = rH + rT runs in this sequence. What is its multiplicity? Consider the sequence below with rH = 7, and rT = 8.

THTHHTHTTTHTHHTHHHTTHT

And let’s put the runs in cells,



First let’s observe that there is no other way to organize the sequence of runs if we want rH = 7 and rT = 8, i.e., no other way to position the cells. They have to go blue-grey-blue-grey-blue-grey-blue-grey-blue-grey-blue-grey-blue-grey-blue. What we *can* do is change the number of T’s in the blue cells, and number of H’s in the grey cells. For instance we could put 3 T’s in the first blue cell and 2 H’s in the first grey cell. And then put 2 T’s in the second blue cell, and 1 H in the second grey cell, etc. Of course the total number of T’s and H’s in all the cells cannot change. So if we think about it, the multiplicity of this run is the number of distinguishable ways we can distribute the nT = 11 T’s into the rT = 8 boxes, multiplied by the number of distinguishable ways we can distribute the nH = 11 H’s into the rH = 7 boxes (note nH doesn’t have to equal nR, but it does in this illustration). This is:



respectively. Have to think about that sometime. Well, they say to think of it this way. Suppose we have our nT T’s lined up. Presently, they are partitioned like so below, in the sequence above. Meaning, we have 1 T assigned to the first box, 1 T to the second, 3 T’s to the third, etc.



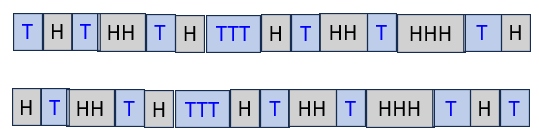
A new assignment can be constructed by rearranging the positions of the rT vertical red bars. There are nT – 1 positions for the bars to go into. And we have to choose which of these our red rT – 1 red bars will go to. So the multiplicity is the number of distinct ways we can assign rT – 1 bars to nT – 1 total positions. Or could think of it like this,



We have nT – 1 total bars. And we choose a subset rT – 1 of these to be red. The red bars delineate the groups/cells. And there would be (nT – 1)!/(rT – 1)!(nT – rT)! ways to do this. So there. Okay, well then the multiplicity would be:



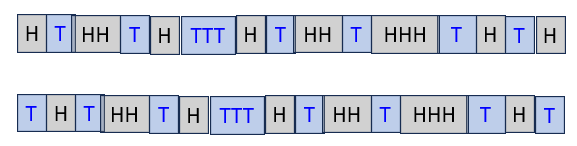
Now we actually want the multiplicity of r = rT + rH runs total. Recalling that |rT – rH| = 0, 1 only, we can add up the multiplicities associated with these possibilities. If r is even, then we must have rT = rH, but we can lead the sequence off with either the T’s or the H’s. Something like this:



So we’d have:



This complicated notation means that either rT,H = (r/2, r/2), or rT,H = (r/2, r/2). Of course these are the same. So we just get double of either case. But I’m writing it this way to make the r = odd case more intelligible. Speaking of, if r is odd, then we can have rT = rH – 1, or rT = rH + 1. But rT + rH must still add up to the same r, either way. Something like,



and the multiplicity would be:



But what are the probababilities of the number of runs then? Well, if we have nT tails and nH heads, how many distinguishable sequences can we make? This is:



So we have:



Finally! Does this simplify any? Let’s do r = even, and nH = nT = n/2. Then the multiplicity would be:



We’ll do Stirling’s approximation,



Taking the ln of both sides, and just going out to nln(n) – n, for simplicity,



Now want to maximize w/r to r,



So the most likely number of runs is in the middle, rμ = n/2 + 1. Let’s expand ln(W) about r = rμ. So we’ll say r = rμ + δ. Plugging this into Wolfram Alpha, we get the following Taylor series,



Exponentiating, and ignoring the constant stuff in brackets, we have:



We can therefore divine that Pr is something like,



This evinces Pr is of Gaussian shape, at least near the peak, with a variance 2σ2 = (n-1)/2 -> σ2 = (n-1)/4. Knowing the normalization constant for a Gaussian, we can say,



FWIW, this is also very close to a binomial approximation for x = number of ‘successes’ given p = 0.5, q = 0.5. There <x> = np = n/2, and <x>2 = npq = n/4. Since the number of runs of a given letter must be roughly rμ/2 ≈ n/4, and the number of instances of a given letter is roughly n/2, our results suggest that the number of runs of a letter (H, or T) should be equal to half the number of instances of that letter. So the average run size for a letter should be 2. Okay. So say we have r\* runs. And we want to know if this is unusual, i.e., not likely random. So we’d do a two-tailed test. And we’d calculate our p-value as:



Basically, we’d be trying to add up the probabilities for runs as extreme or more than r\*. If n is small, then we should probably do the finite Pr equivalent.

**Example**

Suppose a professor makes up answers to a True-False test. The answers are:

T F F T F T F T T F T F F T F T F T T F

and she wants to ascertain whether this sequence looks random or not. So we’ll calculate a p-value. So n = 20, rμ = 11, σ2 = 19/4 = 4.75 -> σ = 2.18. And our number of runs r\* = 16. So |r\* - rμ| = 5. This is greater than 2 std’s away. So can already tell it would not be random at the 95% significance level. Can also see that there 10 T’s. And 8 runs of T’s. So the average size of the T runs is 10/8 = 1.25. This is pretty far from what the average is supposed to be, i.e., roughly 2. Let’s get a p-value,



Actually did this with scipy,



Apparently the correct answer, had we used the finite Pr, is p = 0.038. The correct answer would be obtained by,



And we’d have to use the even/odd formulas for Pr as appropriate. It makes sense that our answer was lower. The Gaussian approximation does worse near the tails of the distribution – it underestimates it. But I’m okay with that. I wonder if approximating Pr with a Student’s T distribution would be better?

**Example**

An analyst has been watching a stock price fluctuate up and down for the past couple months. It generally has an equal chance of going up or going down, every day. However, in the past two weeks, the sequence of ups and downs looked like this:

ududduduududdududuudd

Based on the number of runs in the sequence, is there evidence at the 5% confidence level that this sequence is not random?



So we’d have to form the test statistic,



So yes.