Can one Fake Randomness
Math 199 CHP

Chad Franzen, Rishabh Marya, Robert Weber

May 8, 2014

Abstract

If head/tail sequences obtained by tossing a coin two hundred times are lined up with similar "made up" sequences, can one distinguish the "fake" random sequences from the "real" ones? In most cases one can — humans are notoriously inefficient in mimicking true randomness, and there is a long history of psychological experiments that point towards this conclusion.

The goal of this project is to create interactive online tools to facilitate such experiments, and to develop a suite of randomness tests that is optimized to maximize its effectiveness as a "fake randomness detector." Possible applications range from educational uses at all levels to the testing of real-world data for randomness, and to fraud detection.

1 Introduction

Random numbers have uses far beyond deciding who kicks off a football game. Mathematicians and computer scientists have used random numbers in a wide variety of applications, from running weather simulations (Zhang, 2013) to estimating complicated integrals for financial applications (Aaronson, 2014). These applications rely on computer-generated numbers. Unlike computers, which can imitate randomness very well, humans are very poor at imitating randomness (Bakan, 1960) In this paper, we examine ways to differentiate human generated bits from those generated by a computer. This has many possible applications, ranging from education (what does a random sequence really look like?) to fraud detection (can human-faked data be identified?).

For our experiments, we focused on length-200 bit strings. We chose bit strings because humans have a real-world way of understanding random bit strings (flipping head/tail coins), and they are the simplest random object we could think of. Moreover, they are easy for computers to deal with. We asked humans to enter 200 "heads" or "tails" in a way consistent with flipping a coin. They used keyboards (entering 0's/1's). In the latest versions of our site, visitors enter "h" or "t" or can click on images of coins.

In Section 2, we describe our (informal) definition of randomness, and the differences between human-generated, computer-generated, and truly random numbers. In Section 3, we describe the tests that comprise our testing suite. Section 4 contains links to and descriptions of the final product. Section 5 has our results, including tests on actual humans and on real world data.

2 What Does it Mean to be Random?

2.1 True Randomness vs. Pseudorandomness

Randomness can be informally defined as previous results not having an impact on future ones. For example, if one flipped an unbiased coin ten times, the results of those ten flips would have no effect on the result of the eleventh flip. Computers are not capable of generating large sets of truly random numbers. Instead computers generate pseudo-random numbers. Pseudo-random numbers are generated through a deterministic process, that is with complete knowledge of the state of the system, it is possible to predict the next bit produced. For computers, this requires knowing the "seed" value that was input at the beginning of the generation process — a properly designed random number generator will not allow prediction of future results from only past ones, the way in which those results were created (i.e. the seed value) is required as well, and cannot be determined from just the numbers output. A good set of pseudo-random numbers from a computer will pass tests for randomness — that is they will be statistically indistinguishable from truly random numbers.

2.2 Human Randomness vs. Randomness

Humans are occasionally asked to imitate random numbers. Psychologists who studied these imitations have discovered that humans are generally incapable of imitating random sequences. Both in trivial ways (like overwhelmingly starting a H/T sequence with a head, as Bakan found), and in non-trivial (statistically significant) ways.

3 Our Tests

Since large numbers are required for many important applications (such as Monte Carlo integration) computer scientists have developed a wide variety of tests to determine when a sequence of bits is sufficiently random. These tests are designed to check large sets of numbers (thousands or possibly millions of bits), far more than a human would have the patience to generate. We have adapted these tests so that they can give us information after only 200 bits. We asked humans (students in Math 199) to generate a length 200 "H/T" sequence. We can view this as a length 200 bit string. We developed tests with the goal of differentiating these sequences from bit strings generated by computers.

3.1 Block Frequencies

We begin by separating the string into non-overlapping blocks of a set length. For our purposes, we use blocks of length 3 and 4. As an example, to test blocks of length 3, we would break the string 110101100 into the blocks 110, 101, and 100. There are 2^n possible blocks of length n, and in a random string each block would appear with a theoretical probability of $1/2^n$. Knowing this, we can use a chi-squared test with $2^n - 1$ degrees of freedom to see how well the sample string matches up with a random distribution. We then use a lookup table to match chi-squared values to their corresponding p-values.

3.2 Coupon Collector

This test looks specifically at blocks of length 4. We also allow overlapping blocks, so we obtain our data by looking at characters 1-4, characters 2-5, and so on. The test analyzes how long it takes for all 16 possible blocks of length 4 to appear in the sample string. Typically, it takes human strings much longer to collect all 16 blocks than it does for computer-generated strings. Using simulations (and computer-generated pseudo-random numbers) we can predict how long random strings take to collect all of the possible strings. In our simulations, 94% of computer strings strings would finish by halfway through the string, and 99.9% would collect all the substrings before the end of the 200 bits.

3.3 Gaps Test

This test analyzes the size of gaps between consecutive 1s in the sample string. Two 1s appearing right next to each other represents a gap of 0, two 1s with a single 0 in between represents a gap of 1, and so on. We only consider four gap sizes: $0, 1, 2$, and ≥ 3. Each of the four gap sizes has an associated theoretical probability — a gap of 0 happens with probability 1/2 (the next character is equally likely to be a 0 or 1), a gap of 1 happens with probability 1/4 (the next characters must be 01), etc. We perform a chi-squared test using the frequencies theoretically predicted.

3.4 Alternation Test

Unlike the other tests, this is not an adaptation of a test developed on random number generators. Instead, we developed this test based on our observations that humans tend to view nicely alternating values as random, and long runs of the same value as nonrandom. As a result, the blocks 0101 and 1010 tend to appear more frequently in human strings when compared to computer-generated strings. The alternation test simply counts the number occurrences of these two blocks in the sample string, and compares the extremeness of the result to theoretically expected values.

3.5 Calculating the Fakeness Score

Each of our above tests returns a p-value, representing the probability that a truly random string would produce a result as extreme as the one found. We combine these values into a single value by taking the harmonic mean of the p-values. To find the harmonic mean, which we call $\hat{p}$, we take the mean of the reciprocals of the p-values, and then take the reciprocal of that mean. The formal definition is as follows:

$$\hat{p} = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{p_i} \right)^{-1}$$

The harmonic mean has the effect of attaching more weight to the extreme p-values. For instance, a p-value of 0.05 has a reciprocal of 20, and therefore contributes more to the mean than a p-value of 0.8 with a reciprocal of 1.25. Therefore, if a string catastrophically fails a single test (a very rare occurrence for truly random strings), the string will likely produce a failing overall score even if it passes every other test. However, somewhat out of the ordinary results (those that a random string could get somewhat commonly) will not have too drastic of an effect. When we have obtained our value from the harmonic mean, we subtract it from 1 to produce the "fakeness score," the probability that the string is not random.

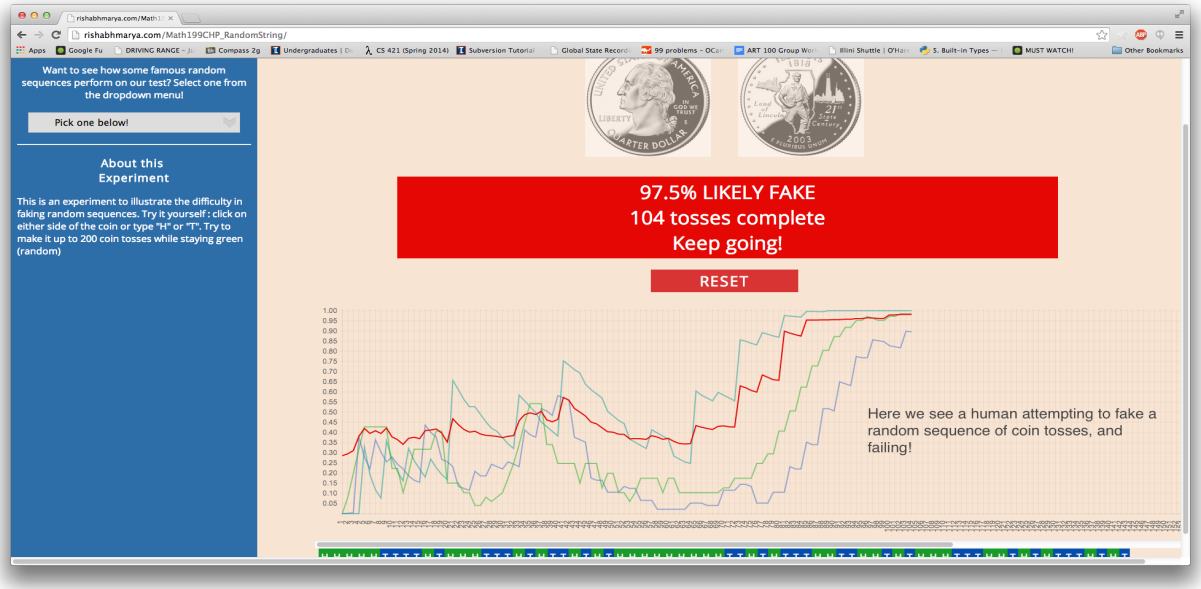
We define a fakeness score of .9 or greater to be significant. That is, a score of .9 or more is considered "likely fake." A score greater than .95 shows even more confidence that the string is fake (due to the lower false positive rate). A score below .9 indicates that our tests could not differentiate that string from a random string.

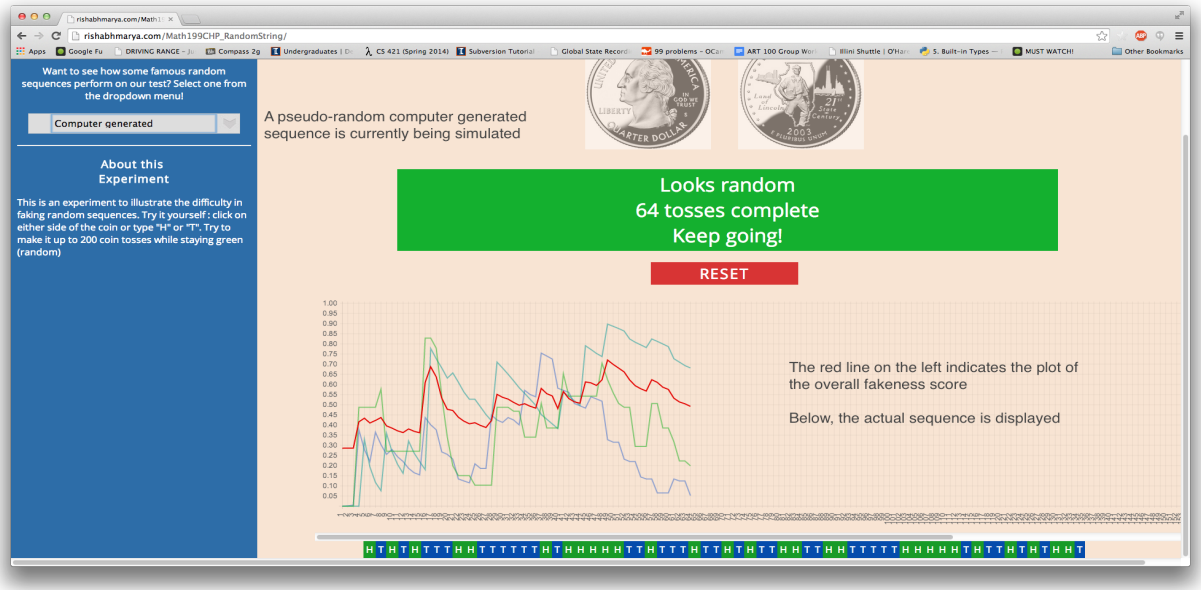
4 The Final Product

Our test suite is currently online, and available to everyone. The site includes the ability to test some famous sequences, as well as to enter your own sequence of heads and tails, to see how close to random your string is.

Visit: `rishabhmarya.com/Math199CHP_RandomString`

or use the QR code below:



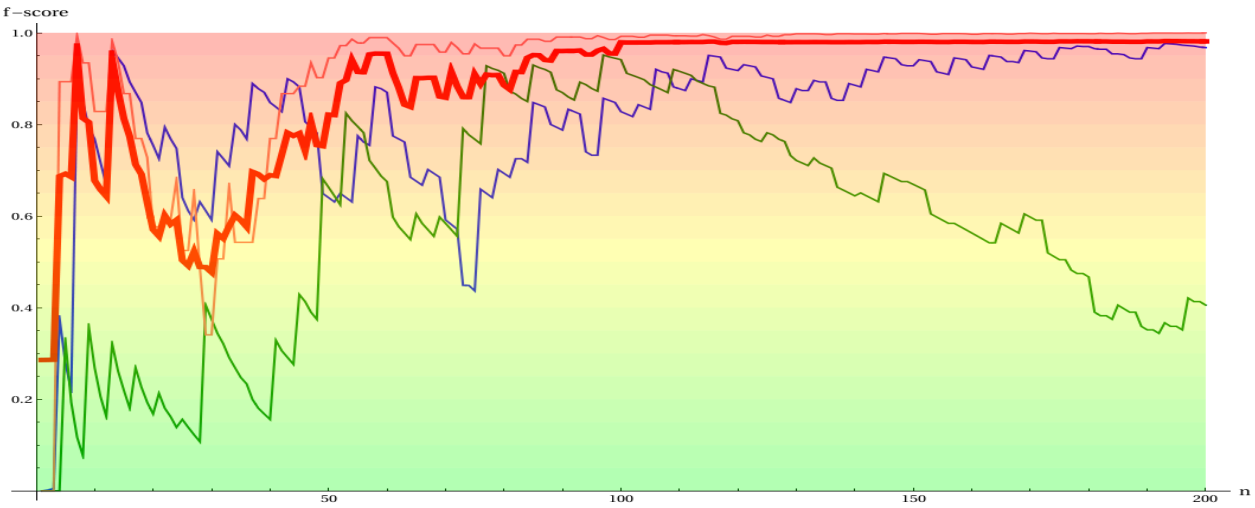


Note that users get instant feedback on how their string is doing with respect to the tests. The large bar at the top gives a color-coded summary. Green indicates the string (up to that point) passes our tests, yellow indicates failure at the .9 level, a score greater than .95 is shown as orange (indicating more confidence in being human), and a score greater than .975 is shown in red (indicating extremely strong confidence that the string was generated by a human). The graph at the bottom of the page shows the string's history with respect to the overall score as well as some of the individual tests. This allows users to learn what in their strings causes them to fail, hence its usefulness as a teaching tool.

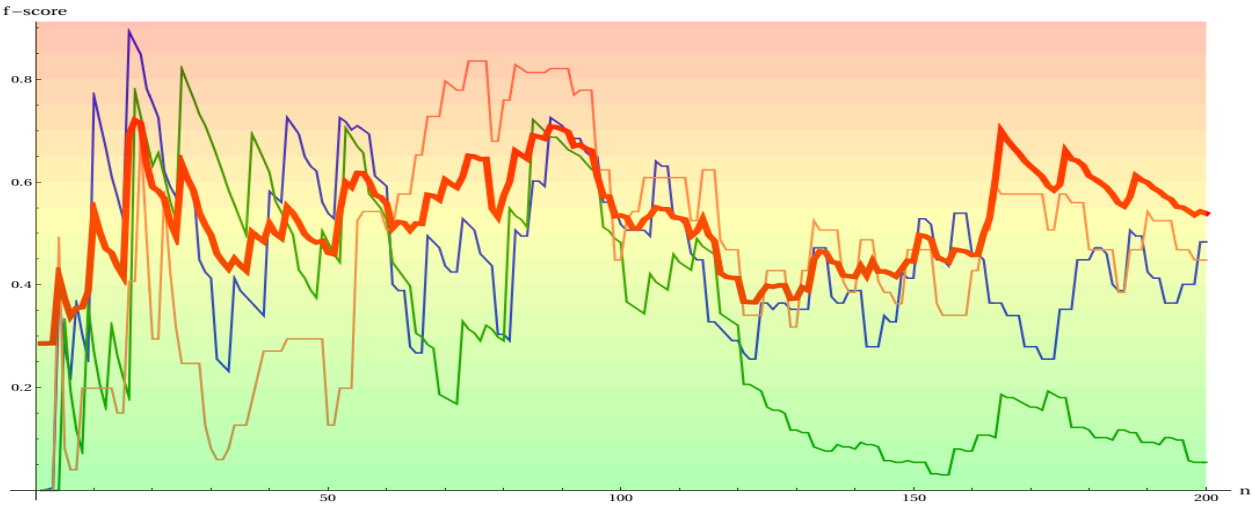
5 Results

5.1 Sample Runs

Below are graphs of some of the tests as they run on a human string and on a computer string. The graph shows $1 - p$ for each of the tests, as a function of the number of bits seen. The blue line is the block frequencies test, with block size 3. The green line is block frequencies, with block size 4, the orange line is the gaps test. The total fakeness score is the thick red line.



Notice that the human string is quickly identified as fake — by only a little more than halfway through the string, the tests have identified it as fake with extreme certainty. This is not uncommon for our tests.



The computer strings fakeness score fluctuates as the tests run across the string, but it generally stays below the significance line.

5.2 Catching Humans

Source	Fakeness Score	3-Blocks Test	4-Blocks Test	Gaps Test	Alternation Test	Coupon Test
Computer	0.146	0.097	0.262	0.008	0.161	0.160
Computer	0.164	0.004	0.306	0.271	0.045	0.106
Computer	0.169	0.035	0.012	0.341	0.191	0.179
Computer	0.232	0.124	0.549	0.022	0.045	0.075
Computer	0.280	0.059	0.306	0.104	0.161	0.531
Computer	0.365	0.021	0.459	0.223	0.469	0.454
Human	0.394	0.124	0.549	0.104	0.419	0.513
Computer	0.408	0.124	0.359	0.341	0.555	0.494
Computer	0.460	0.007	0.459	0.638	0.555	0.284
Computer	0.461	0.004	0.103	0.081	0.756	0.494
Computer	0.474	0.124	0.505	0.428	0.469	0.632
Computer	0.477	0.007	0.054	0.295	0.756	0.494
Computer	0.537	0.081	0.598	0.022	0.631	0.714
Computer	0.562	0.124	0.405	0.842	0.191	0.011
Computer	0.589	0.046	0.638	0.175	0.775	0.632
Computer	0.621	0.059	0.598	0.104	0.469	0.850
Computer	0.637	0.035	0.035	0.318	0.883	0.413
Human	0.769	0.870	0.306	0.175	0.631	0.884
Human	0.953	0.279	1.000	0.223	0.523	0.090
Human	0.966	0.925	0.220	0.968	0.992	0.413
Human	0.984	1.000	0.944	1.000	1.000	0.769
Human	0.985	0.999	0.924	1.000	1.000	0.950
Human	0.986	0.978	0.989	1.000	1.000	0.924
Human	0.988	1.000	1.000	1.000	1.000	0.371
Human	0.988	0.998	0.994	1.000	1.000	0.513
Human	0.988	1.000	1.000	1.000	1.000	0.601
Human	0.988	1.000	1.000	1.000	1.000	0.924
Human	0.989	1.000	1.000	1.000	1.000	0.976
Human	0.989	0.999	0.984	1.000	1.000	0.992
Human	0.990	1.000	1.000	1.000	1.000	1.000
Human	0.990	1.000	1.000	1.000	1.000	1.000
Human	0.990	1.000	1.000	1.000	1.000	0.993

The table shows the results of our tests on 16 human strings (generated by Math 199 students) and 16 computer-generated strings. The first column is the overall fakeness score, with each test showing $1 - p$ (i.e. high values indicate non-randomness). Two humans pass the test — one very easily, one with only some space between the cut-off and the fakeness score. The remaining 14 fail miserably, each reaching a fakeness score well above the .9 cutoff, and each spectacularly failing each test (p score of .01 or less — shown as .99 or more on the table.)

While not shown on the table, the tests will falsely identify random strings as fake. This is inevitable — every 200-bit string will eventually be created by a true random number generator, hence to identify a string as "likely human" we must also risk that a random number generator will eventually generate that string as well. The false positive rate is reasonably small — at the $f = .9$ level, simulations produced about 13% of strings as "likely fake." At the .95 level, 6% of the strings failed. These are each surprisingly close to the 10% and 5% rates which p-values usually indicate, especially considering the aggressive nature of the averaging process.

5.3 Real World Data

In addition to human generated data, we can run the tests on real-world bit strings to test them for randomness. Below are scores for some mathematical constants, as well as for data from sports teams. The mathematical constants are the first 200 bits in their binary expansions. These numbers are either hypothesized to be normal, or (for the Copeland-Erdos constant) constructed to be normal. The sports teams are the win-loss sequences of their seasons (i.e. "heads" or "1" represents a win, "tails" or "0" a loss — for hockey we considered merged overtime and regulation losses). The teams were selected for having approximately as many wins as losses.

Real World Data	Fakeness Score
Binary Expansion of Golden Ratio	0.796
Binary Expansion of √2	0.401
Binary Expansion of √3	0.644
Copeland Erdös Constant	0.599
2013 Arizona Diamondbacks	0.275
2011-2012 Ottawa Senators	0.709
2012-2013 Boston Celtics	0.306
Super Bowl Coin Toss	0.389

Each of the constants passes the tests (which is consistent with tests of normality done for these constants). The sports data indicates that win-loss sequences of sports teams are not like human-generated sequences, but more like a truly random sequence. This may be due to actual randomness in determining game results, or just indicate teams are more likely to have win streaks than humans are likely to enter consecutive "heads" or "tails"

6 Further applications

These tests have a wide range of possible applications. They can be used as an instruction tool when discussing randomness, especially with the instant feedback on which tests are failing and which are passing. This can be used to emphasize the difference between balance in a distribution and randomness. Secondly, they can be used in fraud detection — for binary sequences the tests can be applied instantly. Otherwise, one can map the data to a binary sequence (e.g. by taking generated numbers mod 2). The tests as written require that 0 and 1 are equally likely to appear, however they could be modified (by modifying the probabilities) to function for non-uniform distributions.

7 Summary

Humans generally fail at imitating random sequences. Starting with tests designed to check the abilities of random number generators, we have developed a set of tests, which catches the majority of human-generated strings, while having a reasonably low false positive rate in our initial testing. The tests have a range of applications, from teaching to fraud detection.

8 References

1. Aaronson, S. The Quest for Randomness. *American Scientist*, 102(3), 170-173. 2014.

2. Bakan, P. Response-tendencies in attempts to generate random binary series. *American Journal of Psychology*, **73**, 127-131. (1960).

3. Knuth, Donald Ervin. *The art of computer programming.* Pearson Education, (2005).

4. Marsaglia, George. "DIEHARD: a battery of tests of randomness." (1996).

5. Zhang, F., & Tao, D. Effects of Vertical Wind Shear on the Predictability of Tropical Cyclones. Journal Of The Atmospheric Sciences, 70(3), 975-983. (2013).