Simple Test for Randomness (STR Test)

What is the **probability** that the number **7** will be found in some 10-digit **decimal** string?

The probability that the number is NOT in the **first position** of the 10-digit string is 9 out of 10, or

$$\left(\frac{9}{10}\right)$$

The probability that the number is NEITHER in the **first** NOR in the **second** position will be

$$\left(\frac{9}{10}\right)^2$$

And, of course, the probability that the number is **not present** in the decimal ten-digit string is

$$\left(\frac{9}{10}\right)^{10} \approx 0.3486784 \approx 3$$

What we take it to mean, is that a **truly random** ten-digit decimal string will NOT contain about a third (or three) of all possible one-digit numbers and also that it **WILL contain** about seven out of ten one-digit decimal numbers from 0 to 9.

The following ten-digit strings would qualify as **random** in our test:

1234567777, 1111234567, 0102030456 etc.

What about longer strings?

Let's apply the same reasoning for a string that is about ten times longer. For a decimal string to contain **100** (one hundred) two-digit numbers it has to be at least 101 character long.

For example, the string [1.2] below contains the following two-digit numbers:

12, 23, 34, 45, 56, 67, 78 and 89.

1 2 3 4 5 6 7 8 9 [1.2]

It contains **eight** two-digit numbers, and it is **nine** character long.

What is the probability that a two-digit number, say 77, will be found in a 101 character long decimal string?

By the same logic, the probability that the two-digit (N = 2) number will NOT be present in the string is:

$$P(2) = \left(\frac{99}{100}\right)^{100} \approx 0.366032341$$

We could call **N -** the number of digits searched – or the **granularity** of the tested string.

So in a 101 long decimal string we should find approximately 63 two-digit numbers out of 100 to call that string a **truly random** string according to our test.

So, in general, the probability that an N-digit (N-granularity) number will NOT be present in a decimal string of the length of L, can be calculated as follows:

$$P(N) = \left(\frac{10^N - 1}{10^N}\right)^{10^N} \qquad [1.2]$$

Where N – the **granularity** of the tested string.

The length L required depends on the **granularity** N of the tested string and can be calculated as follows:

$$L(N) = 10^N + (N-1) \qquad [1.3]$$

For a string to contain 10^N N-digit numbers, it has to be at least $10^N + (N-1)$ character long. For example:

When N=1 (for one-digit numbers), then the length L of our tested string should be 10 digit long.

$L(1) = 10^1 + (1-1) = 10$

When N=2 (for two-digit numbers), then the length L of the tested string should be 101 digit long.

$L(2) = 10^2 + (2-1) = 101$
Etc.

When N is sufficiently large, then P (N) from [1.2] approaches $\frac{1}{e}$

$$P(N) = \lim N\text{-->}00 \left(\frac{10^N - 1}{10^N}\right)^{10^N} = \frac{1}{e} \quad [1.3]$$

Where e – is the Euler's number, the base of the natural logarithm.

e ≈ 2.718281828…

In case we have a string of a **given length L**, then the **granularity** N will be calculated as follows:

N = log L [1.4]

Where
L – given length of the string.
N – granularity of the tested string.
log – base-10 logarithm.

The calculated value of N should be **rounded up** to the nearest one, when the digit after the decimal point is more than zero. E.g. if N = 3.15 or N = 3.72, then we make granularity equal to 4, whereas when N = 3.09, we make N = 3.

Rounding up is required to improve the reliability of the test.

For example, if we take the string of, say, 120 characters long, then
N = log120 = 2.079, rounded to the nearest thousandth.

In this case we make the granularity N equal to two (N = 2).

When we have a string of 130 characters, then log N = 2.11, and we round up to make granularity N equal to three (N=3).

I wrote a simple VB and C# program to test the number Pi for randomness, taking one million digits of Pi from the site
http://www.piday.org/million/
and then one **billion** digits of Pi from the site
https://stuff.mit.edu/afs/sipb/contrib/pi/

The test results:

Number of digits N searched (Granularity)	**The length of the Pi string**	**P(N)**	**Theoretical prediction**	**Percentage Difference %**
3	1 002	0.367	0.368	0,272108844
4	10 003	0.3635	0.3679	1,203171999
5	100 004	0.36722	0.36788	0,179567406
6	1 000 005	0.368453	0.367879	0,155907933

As we see, we had:

367 out of 1000 three-digit numbers missing in the 1002 long Pi string,
3635 out of 10 000 four-digit numbers missing in the 10 003 long Pi string,
36722 out of 100 000 five-digit numbers missing in the 100 004 long Pi string, and
368453 out of 1 000 000 six-digit numbers missing in the 1 000 005 long Pi string.

All of these numbers are very close to the theoretical value of

$$\frac{1}{e} \approx 0.367879$$

So the number Pi is very close to being a truly random number according to the **STR test**.

The percentage difference in neither of them exceeds 1.3% of the theoretical value P(N).

Note: we have tested accordingly the first 1 002, 10 003, 100 004 and the first 1 000 005 digits of the decimal expansion of the Pi string.

I have also tested the 400 000 long random decimal string that was generously provided by the site (https://www.random.org/strings/)

And from the site - https://apod.nasa.gov/htmltest/gifcity/e.2mil tested the first 500 000 digits of the **Euler's** number e.

According to the STR tests, all these strings differed less than in one per cent from the theoretically calculated randomness value P(N).