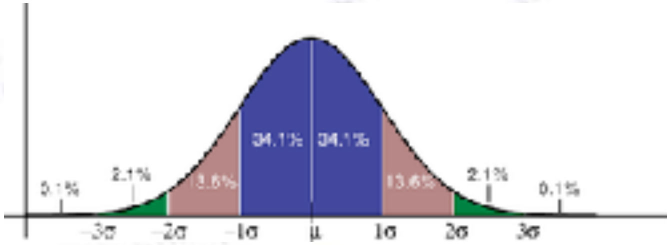
# Applied Statistics

## Testing random Number





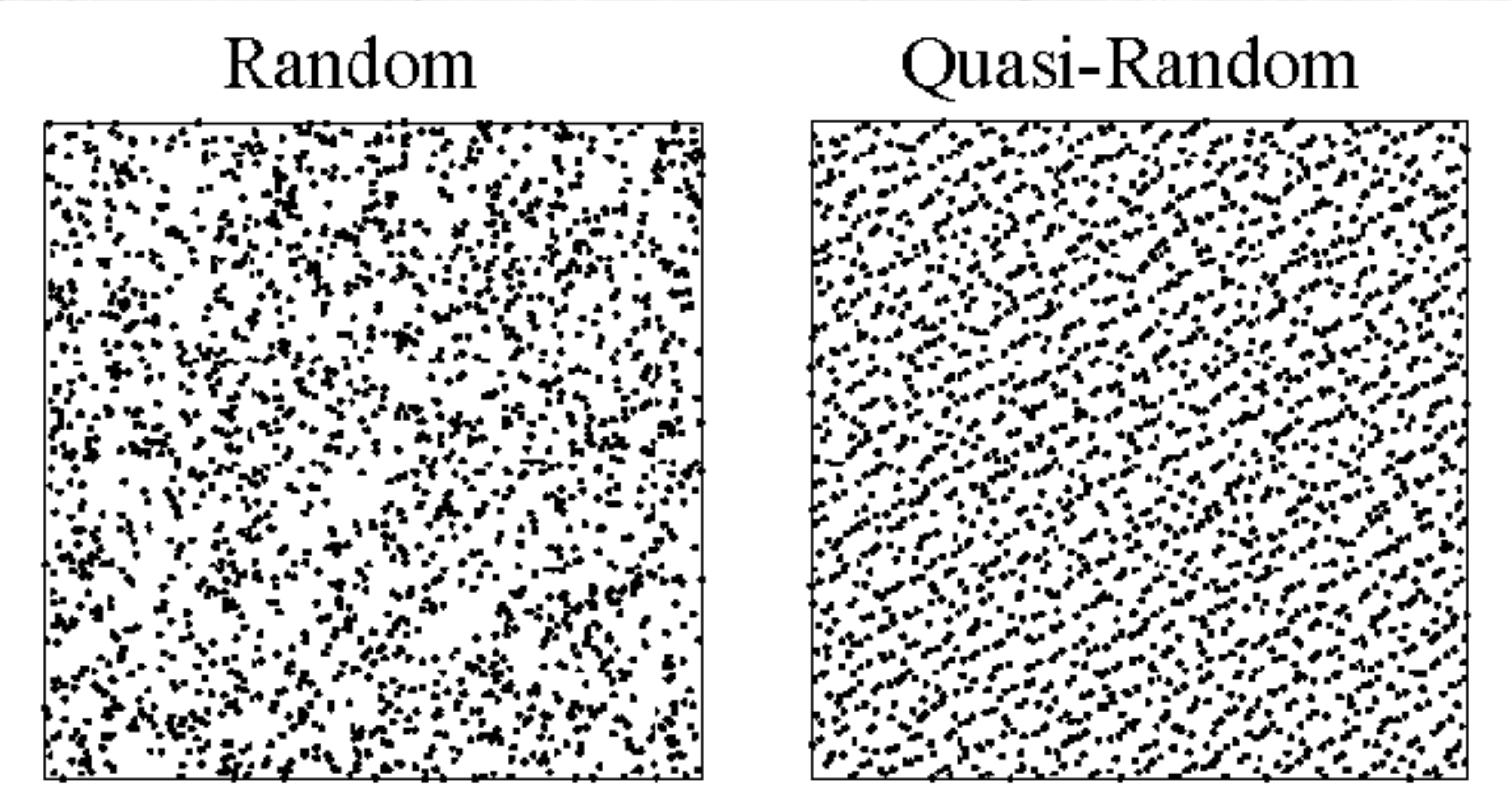

Troels C. Petersen (NBI)




*"Statistics is merely a quantisation of common sense"*

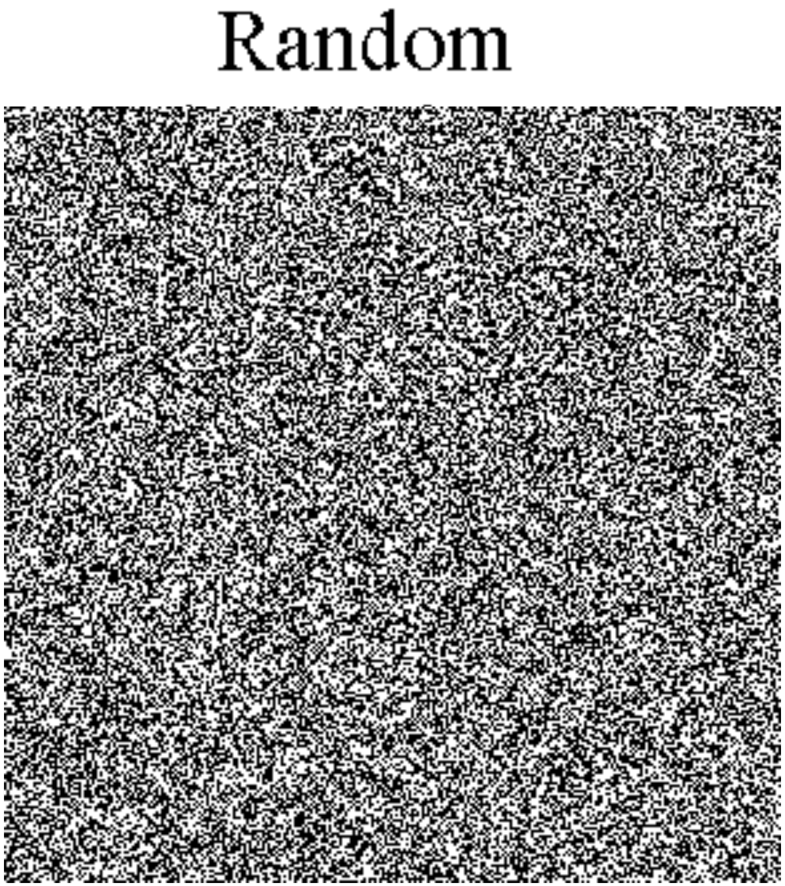
# Random numbers

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." [John Von Neumann]

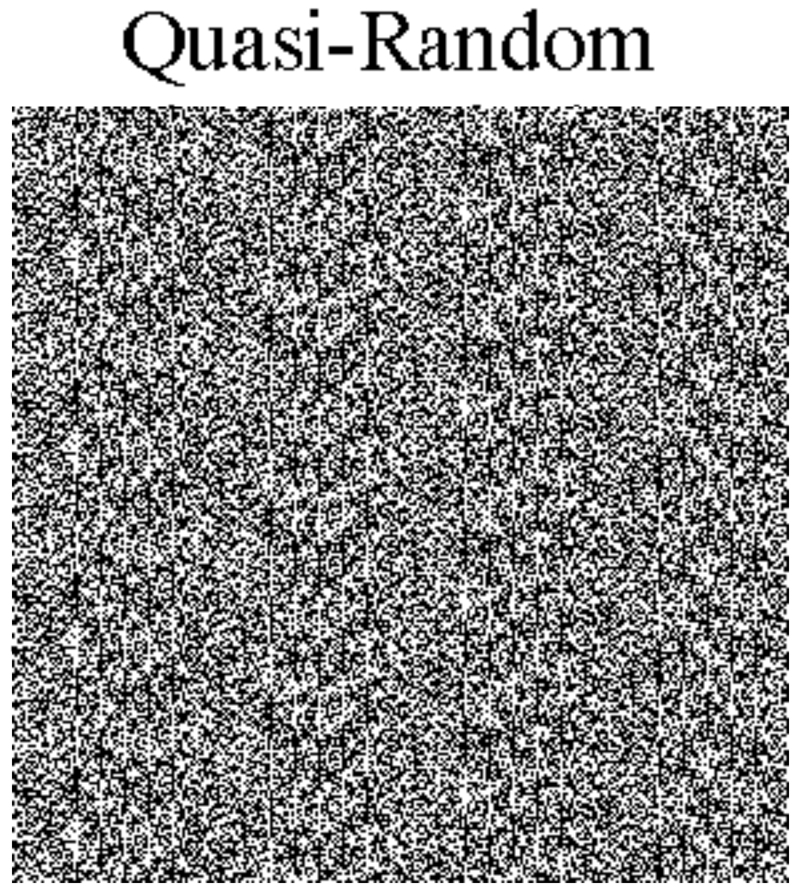


# Random numbers

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." [John Von Neumann]




RANDOM.ORG




PHP rand() on Microsoft Windows

The build-in random number generators are sometimes not optimal!

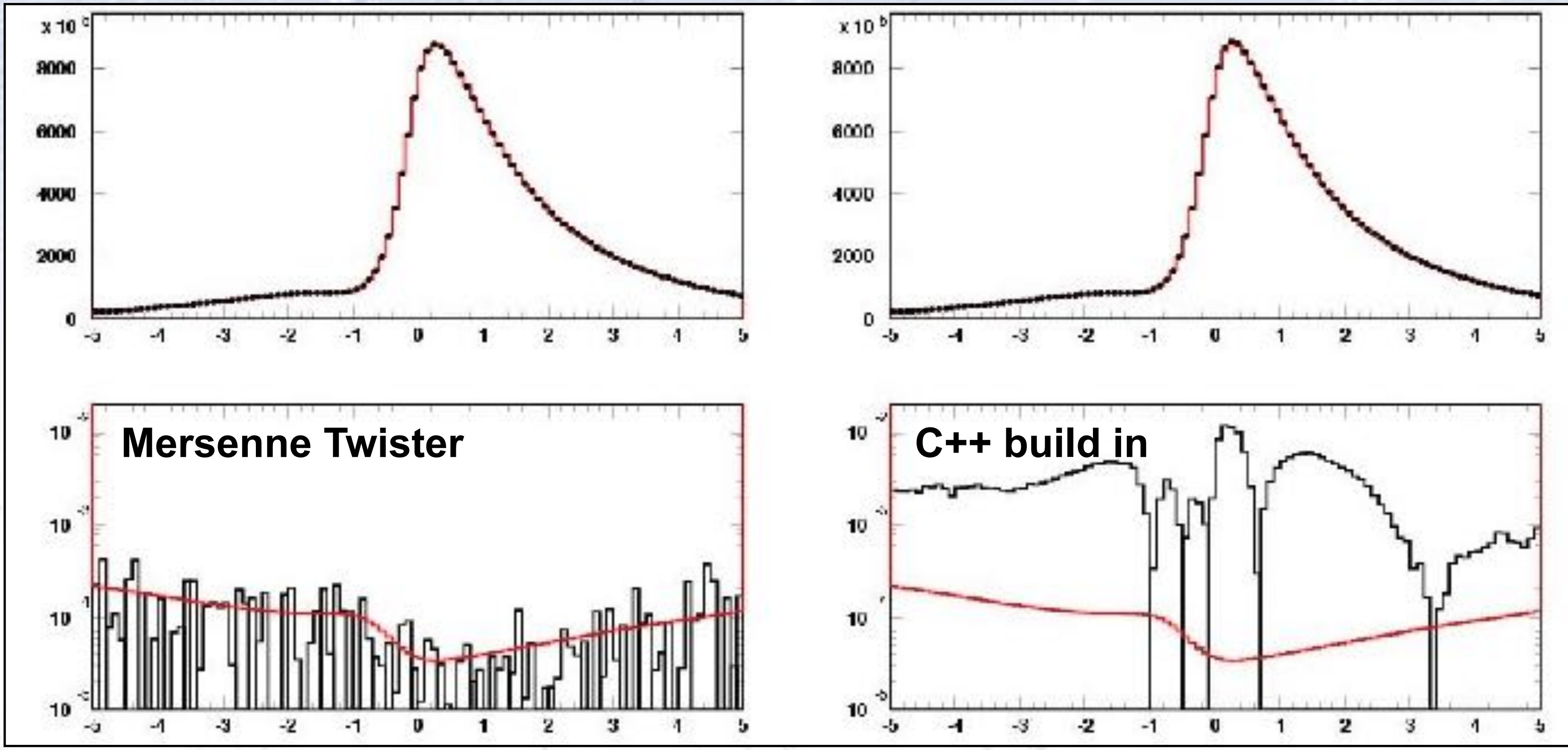



Figure 8.1: *To test the resolution function convolutions and the Monte Carlo generation, a large Monte Carlo sample of 25 billion events is generated, binned, and compared to the predictions of the convolutions. The actual function used is* $e^{-2|t|}\,[\cosh(1.5t) + 0.5\sinh(1.5t) + 0.1\cos(t) + 0.9\sin(t)]$ *convoluted with a GExp function with* $(\sigma, s, \tau) = (0.2, 0.7, 0.5)$. *In the upper plots, the red histograms show the predicted values and the black dots the number of accumulated Monte Carlo events. In the lower plots the relative difference is shown in a logarithmic plot. The red curve shows the expected statistical deviation and the black histogram the actual relative difference. In (a) the Monte Carlo was generated using a decent random number generator while in (b), it was generated using the standard built-in C++ routine. The agreement in (a) seems to be perfect within the statistical precision of 4-5 significant digits. In (b) on the other hand, the Monte Carlo deviates already at 2-3 significant digits, which is unacceptable.*

**T. Kittelmann, master thesis (2002)**

# Random numbers

# Random numbers

*"Most studies find that human subjects have some degree of non randomness when attempting to produce a random sequence of e.g. digits."*

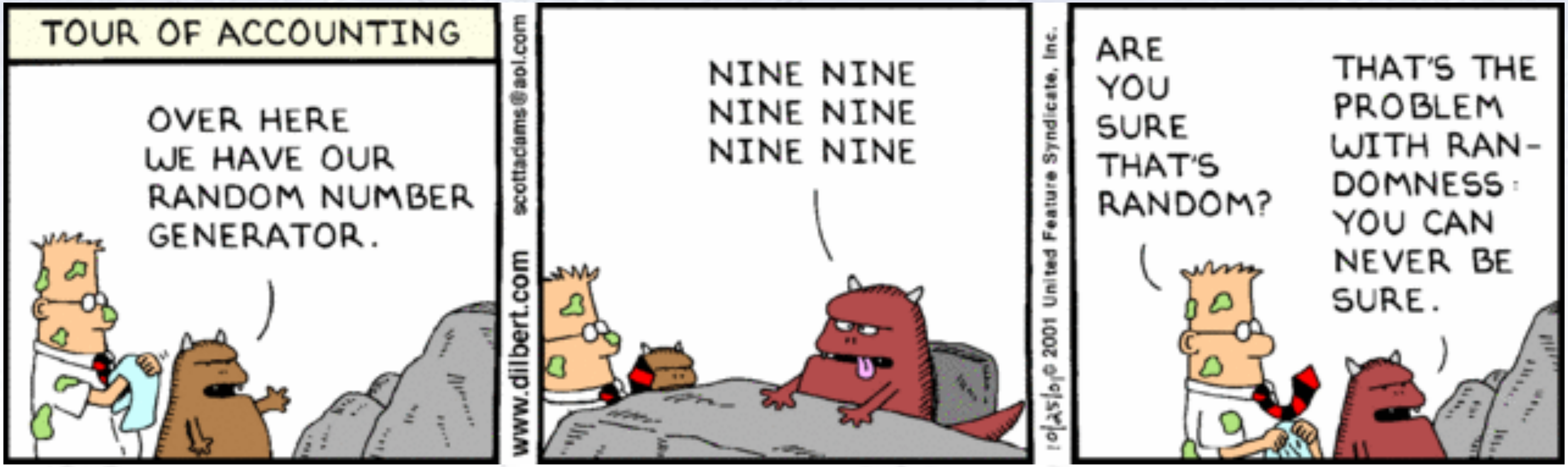
[Wikipedia, On random numbers produced by humans]

# Random numbers

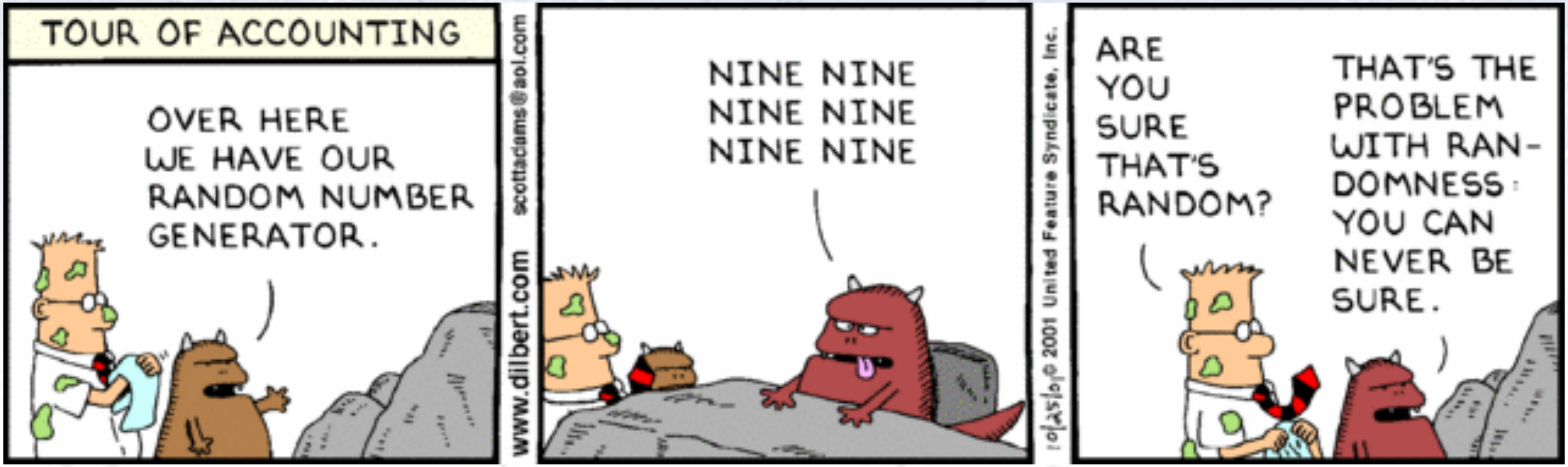
*"Most studies find that human subjects have some degree of non randomness when attempting to produce a random sequence of e.g. digits."*

[Wikipedia, On random numbers produced by humans]

# Testing random numbers

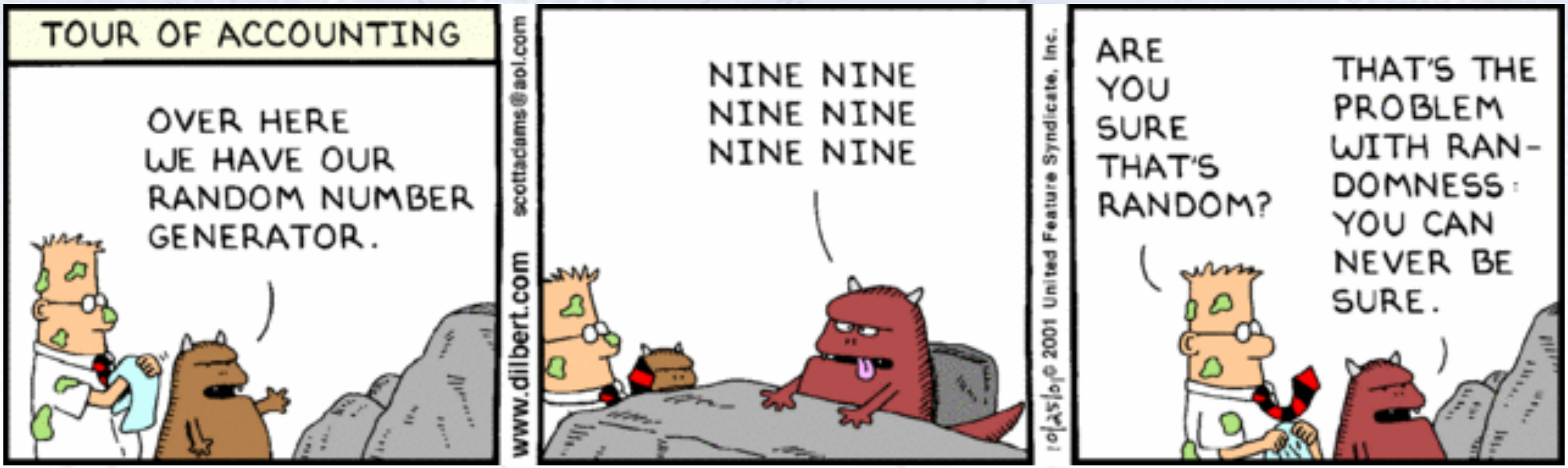


# Testing random numbers




Discuss with those sitting next to you (2-3 min), how you would go about testing, if a series of digits were random or not.

# Testing random numbers




In contrast to popular believe, there are ways of testing random numbers. But it is not easy! A recommended list (Louis Foley, Random.org, 2001) is:

- A chi-square test
- A test of runs above and below the median
- A reverse arrangements test
- An overlapping sums test
- A binary rank test for 32×32 matrices

# Binary Rank Test

A binary rank test for 31×31 matrices:

The leftmost 31 bits of 31 random integers from the test sequence are used to form a 31x31 binary matrix over the field {0,1}.

The rank is then determined. That rank can be from 0 to 31, but ranks < 28 are rare, and their counts are pooled with those for rank 28.

Ranks are found for 40,000 such random matrices and a chi-square test is performed on counts for ranks 31, 30, 29 and <=28.

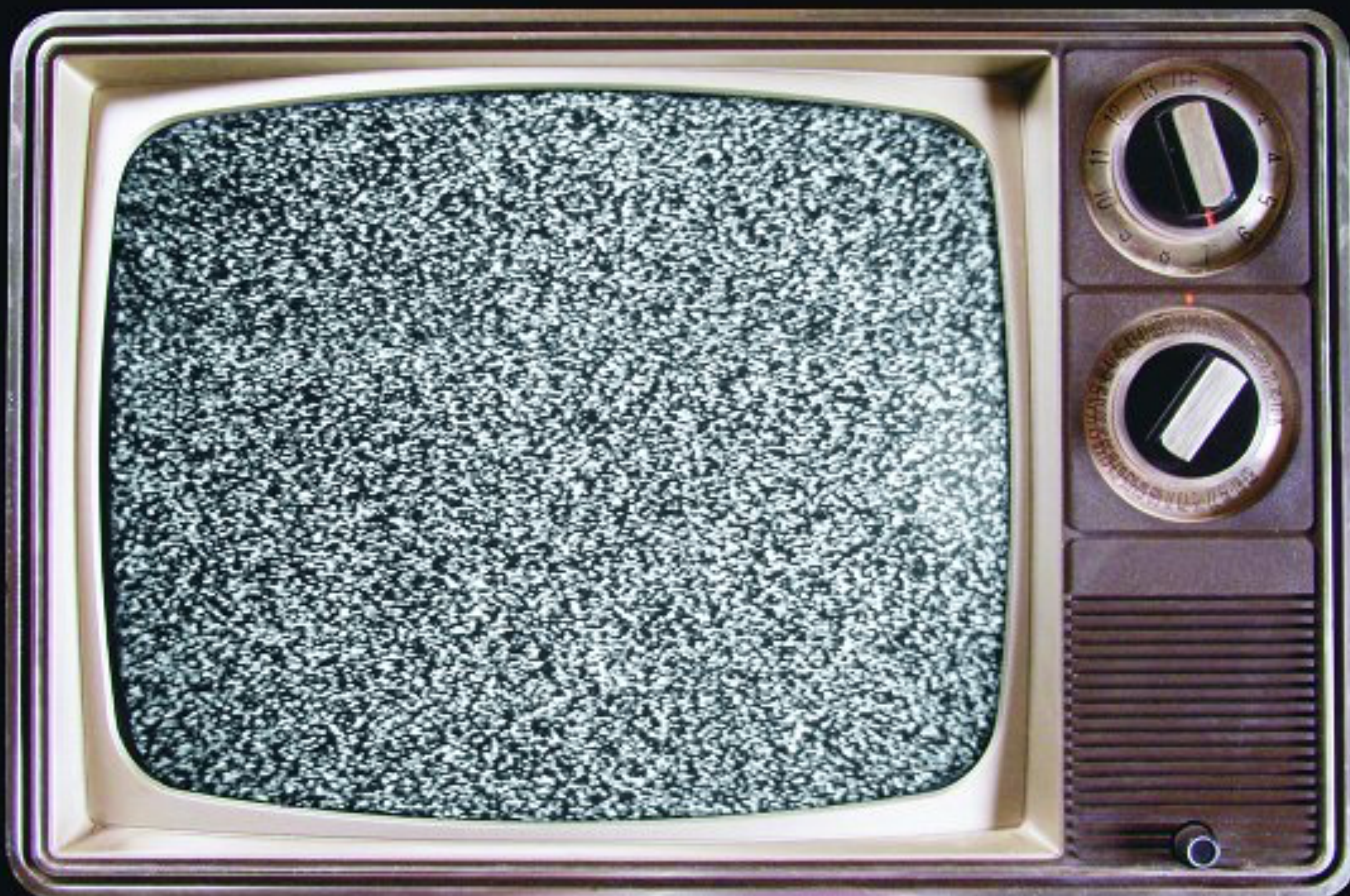
I.e. given that this distribution is known for truly random numbers, you can see if the distribution of ranks is the same!

# More tests for randomness

| NIST Statistical Test Suite | | |
|---|---|---|
| **Test** | **Defect Detected** | **Property** |
| Frequency (monobit) | Too many zeroes or ones | Equally likely (global) |
| Frequency (block) | Too many zeroes or ones | Equally likely (local) |
| Runs test | Oscillation of zeroes and ones too fast or too slow | Sequential dependence (locally) |
| Longest run of ones in a block | Oscillation of zeroes and ones too fast or too slow | Sequential dependence (globally) |
| Binary matrix rank | Deviation from expected rank distribution | Linear dependence |
| Discrete fourier transform (spectral) | Repetitive patterns | Periodic dependence |
| Non-overlapping template matching | Irregular occurences of a pre-specified template | Periodic dependence and equally likely |
| Overlapping template matching | Irregular occurences of a pre-specified template | Periodic dependence and equally likely |
| Maurer's universal statistical | Sequence is compressible | Dependence and equally likely |
| Linear complexity | Linear feedback shift register (LFSR) too short | Dependence |
| Serial | Non-uniformity in the joint distribution for m-length sequences | Equally likely |
| Approximate entropy | Non-uniformity in the joint distribution for m-length sequences | Equally likely |
| Cumulative sums (cusum) | Too many zeroes or ones at either an early or late stage in the sequence | Sequential dependence |
| Random excursions | Deviation from the distribution of the number of visits of a random walk to a certain state | Sequential dependence |
| Random excursions variants | Deviation from the distribution of the number of visits (across many random walks) to a certain state | Sequential dependence |

# What to use for random seed?

"Randomisation is too important to be left to chance." [ J. D. Petruccelli]



# NSA and Dual_EC_DRBG

Cryptography is based on random numbers. One of the four accepted random number generators (RNG) used for cryptography was Dual_EC_DRBG.

Several academics point out, that Dual_EC_DRBG was a very poor and possibly “back-doored” pseudorandom number generator! Nevertheless, one of the large cryptography companies in the US, RSA Security, continued using it.

# NSA and Dual_EC_DRBG

Cryptography is based on random numbers. One of the four accepted random number generators (RNG) used for cryptography was Dual_EC_DRBG.

Several academics point out, that Dual_EC_DRBG was a very poor and possibly “back-doored” pseudorandom number generator! Nevertheless, one of the large cryptography companies in the US, RSA Security, continued using it.

In 2013, Edward Snowden published papers that showed, that the NSA had put a backdoor in the Dual_EC_DRBG algorithm! The lack of perfect randomness allowed NSA to break the encryption.

Outputs of multiple independent RNGs can be combined (for example, using a bit-wise XOR operation) to provide a combined RNG at least as good as the best RNG used. This is referred to as software whitening.