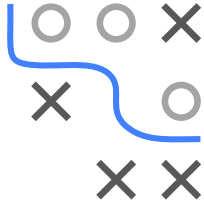
Algorithms and Data Structures

Random Numbers
Introduction to Random Numbers

Learning goals

- True random numbers
- Pseudo-random numbers
- Pseudo-random number generators (PRNG)

TRUE RANDOM NUMBERS



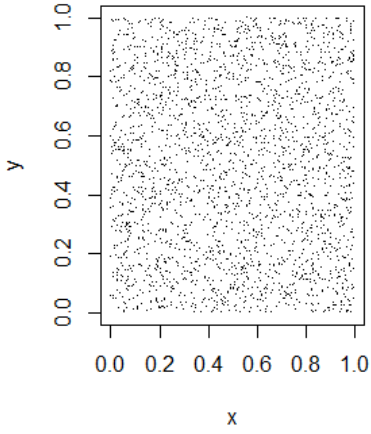
To source "true" random numbers there are several possibilities:

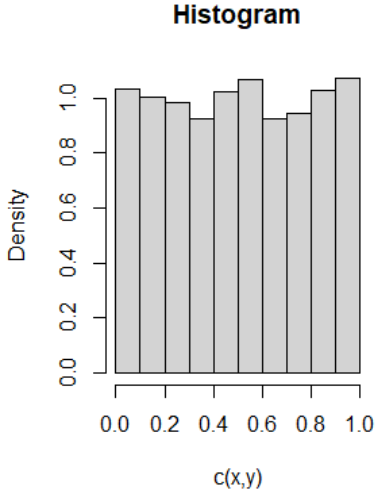
- Tossing a coin, dice, roulette wheel, ...
- Noise from electronic components (e.g. device drivers of a computer)
- Noise in the atmosphere (`http://random.org`)
- Radioactive decay (example HRNG: `https://bit.ly/2NZ8whF`)
- Response time of a user to a command prompt, time differences when typing on the keyboard (both measured in milliseconds or even more accurately), random mouse movement, . . .
- . . .

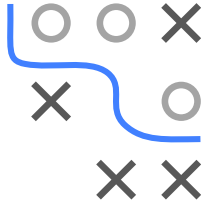
They all have in common that the creation of long sequences is very time-consuming or even impossible.

Also, an exact repetition of a "random experiment" is not possible.

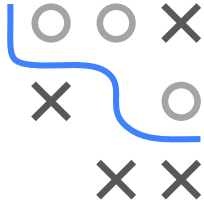
TRUE RANDOM NUMBERS / 2





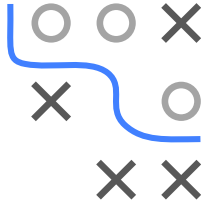


PSEUDO-RANDOM NUMBERS



- **Definition**: A sequence of pseudo-random numbers is a deterministic (!) sequence of numbers with the same relevant properties as a sequence of independent and identically distributed random variables.
- Usually numbers from discrete and continuous uniform distributions, other distributions result from transformations.
- Starting point: Let x_i be a sequence of natural numbers with discrete uniform distribution on the interval $[0, m]$. Then $u_i = x_i / m$ (with m being large) is an approximation of the continuous uniform distribution with numerical precision $1/m$.

PSEUDO-RANDOM NUMBERS / 2

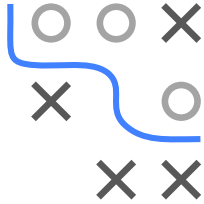


Initial value x_0: Must be specified, initialization is frequently done based on the system time. "Random" experiments with **Pseudo-random number generators (PRNG)** can be reproduced completely if the initial value is known ("seed value").

Period: Since there is only a finite set of numbers available, at some point $x_k = x_0$ must hold and the sequence repeats itself: $x_{i+k} = x_i$, $x_{i+k+1} = x_{i+1}$, ...

PRNG with periods as long as possible are desirable.

QUALITY CRITERIA FOR PRNG



Relevant properties of PRNG for uniformly distributed random numbers on $(0, 1)$:

- Actual uniform distribution on $(0, 1)$.
- Distributions of pairs, triplets, etc. are also random (especially for multidimensional uniform distribution).
- Period length. If the period is too short, then ...
  - ... many integers cannot be realized (problem due to uniform distribution)
  - ... there are not enough independent random numbers

The *relevance of each property* depends very much on the respective application.

TESTING PRNG

In order to assess the quality of PRNGs, they are subject to strict random number test suites. The best known is "Die Hard" from George Marsaglia (available for free):

- Considered individually, is each bit i.i.d. 0 or 1 with probability $1/2$?
- Spectral test: Are n-tuples uniformly distributed in the unit cube? Are there autocorrelations?
- Rank of binary 6×8 and 32×32 matrices.
- `R` package **RDieHarder**
  `http://cran.r-project.org/web/packages/RDieHarder/`