Pseudo-Random Number Generation

Random Number Generation

- To a very high degree computers are deterministic and therefore are not a reliable source of significant amounts of random values. In general pseudo random number generators are used.
- The default algorithm in `R` is Mersenne-Twister but a long list of methods is available.
- See the help of `RNGkind()` to learn about random number generators.

Pseudo-Random Number Generation

`.Random.seed` is an integer vector, containing the **random number generator** (RNG) state for random number generation in `R`.
It can be saved and restored, but should not be altered by the user.

`RNGkind` is a more friendly interface to query or set the kind of RNG in use.

Pseudo-Random Number Generation

`RNGversion` can be used to set the random generators as they were in an earlier `R` version (for reproducibility).

`set.seed` is the recommended way to specify seeds.

Seed

A pseudo random number generator is an algorithm based on a starting point called "**seed**".
If you want to perform an exact replication of your program, you have to specify the seed using the function `set.seed()`. The argument of `set.seed()` has to be an integer.

```
> set.seed(1)
> runif(1)
[1] 0.2655087
> set.seed(1)
> runif(1)
[1] 0.2655087
```

Pseudo-Random Number Generation

Mersenne Twister

Mersenne Twister(MT) is a pseudorandom number generating algorithm developped by Makoto Matsumoto and Takuji Nishimura (alphabetical order) in 1996/1997.

- It is designed with consideration on the flaws of various existing generators.
- The algorithm is coded into a C-source downloadable below.
- Far longer period and far higher order of equidistribution than any other implemented generators.

Mersenne Twister

Mersenne Twister

- Fast generation.
  (Although it depends on the system, it is reported that MT is sometimes faster than the standard ANSI-C library in a system with pipeline and cache memory.)
- Efficient use of the memory.