Random number generation in R

We'll begin with a brief discussion of the random number generator (RNG) in R.

The RNG can be *seeded* with `set.seed(n)` for any integer `n`. The values that the RNG outputs will depend on its seed, and setting the seed "resets" it to the initial state corresponding to that particular seed.

- Try using `set.seed()` in conjunction with `runif(5)` to get a sense of how this works.

This is important because we may want, *e.g.*, to generate the same random partitioning of our data consistently, in which case we would put a `set.seed(1)` call before our shuffling of the indices with `sample()`.[1] Or, alternatively, you may want a sequence of reproducibly different calls of `sample()`, etc.

[1] In some cases, you may want to (1) save the state of the RNG for later, (2) set the seed to something specific and generate a consistent splitting of the data, and (3) change the RNG back to its saved state. This is possible using `.Random.seed` and is described in Cookbook for R – we won't need this for this lesson, but it's important to be aware of (as it will eventually surely come up).