Reproducible Randomness in R

`set.seed()` and the Random Number Generator (RNG)

What does `set.seed()` do?

Core idea

Computers generate **pseudo-random** numbers using a **random number generator (RNG)**.

- The RNG produces a sequence that *looks* random.
- The sequence is determined by an internal **state**.
- `set.seed(s)` sets that state so you can reproduce the same sequence.

Why we care

Reproducibility: you (and others) can re-run your code and get identical results.

Quick demo: same seed ⇒ same output

```
set.seed(123)
runif(5)

set.seed(123)
runif(5) # identical to the previous runif(5)
```

- `runif`, `rnorm`, `sample`, etc. use the RNG.
- Setting the seed once near the top usually makes a script reproducible.

How to use seeds in practice

Typical patterns

- **Script / report:** set one seed near the top.
- **Functions:** usually *do not* set a seed inside a function (avoid surprising users).
- **Experiments:** store the seed you used so you can recreate the exact run.

Good habit (reproducible simulation)

`set.seed(42)` ⇒ same simulated data each run

Summary

- `set.seed(s)` makes random outputs reproducible.
- Same seed + same code path ⇒ same random draws.