

If you ever need to figure out if a function call in R generated a random number or not, here is a simple trick that you can use in an interactive R session. Add the following to your

`~/Rprofile(*)`:

```
if (interactive()) {
  invisible(addTaskCallback(local({
    last <- .GlobalEnv$.Random.seed

    function(...) {
      curr <- .GlobalEnv$.Random.seed
      if (!identical(curr, last)) {
        msg <- "NOTE: .Random.seed changed"
        if (requireNamespace("crayon", quietly=TRUE)) msg <-
crayon::blurred(msg)
        message(msg)
        last <-<- curr
      }
      TRUE
    }
  })), name = "RNG tracker"))
}
```

It works by checking whether or not the state of the random number generator (RNG), that is, `.Random.seed` in the global environment, was changed. If it has, a note is produced. For example,

```
> sum(1:100)
[1] 5050
> runif(1)
[1] 0.280737
NOTE: .Random.seed changed
>
```

It is not always obvious that a function generates random numbers internally. For instance, the `rank()` function may or may not update the RNG state depending on argument `ties` as illustrated in the following example:

```
> x <- c(1, 4, 3, 2)
> rank(x)
[1] 1.0 2.5 2.5 4.0
> rank(x, ties.method = "random")
[1] 1 3 2 4
NOTE: .Random.seed changed
>
```

For some functions, it may even depend on the input data whether or not random numbers are generated, e.g.

```
> y <- matrixStats::rowRanks(matrix(c(1,2,2), nrow=2, ncol=3),
ties.method = "random")
NOTE: .Random.seed changed
> y <- matrixStats::rowRanks(matrix(c(1,2,3), nrow=2, ncol=3),
```

```
ties.method = "random")  
>
```

I have this RNG tracker enabled all the time to learn about functions that unexpectedly draw random numbers internally, which can be important to know when you run statistical analysis in parallel.

As a bonus, if you have the [crayon](#) package installed, the RNG tracker will output the note with a style that is less intrusive.

(*) If you use the [startup](#) package, you can add it to a new file `~/.Rprofile.d`
`/interactive=TRUE/rng_tracker.R...`