✓ **Félicitations ! Vous avez réussi !**

POUR RÉUSSIR 80 % ou plus

Continuer à apprendre

NOTE 100 %

Week 4 Quiz

NOTE DE LA SOUMISSION LA PLUS RÉCENTE

100%

1. What is produced at the end of this snippet of R code? **1 / 1 point**

```
set.seed(1)
rpois(5, 2)
```

- A vector with the numbers 1, 4, 1, 1, 5
- (selected) A vector with the numbers 1, 1, 2, 4, 1
- It is impossible to tell because the result is random
- A vector with the numbers 3.3, 2.5, 0.5, 1.1, 1.7

✓ **Correct**

Because the `set.seed()` function is used, `rpois()` will always output the same vector in this code.

2. What R function can be used to generate standard Normal random variables? **1 / 1 point**

- qnorm
- dnorm
- (selected) rnorm
- pnorm

✓ **Correct**

Functions beginning with the `r` prefix are used to simulate random variates.

3. When simulating data, why is using the set.seed() function important? Select all that apply. **1 / 1 point**

- [ ] It can be used to generate non-uniform random numbers.
- [ ] It ensures that the sequence of random numbers is truly random.
- [ ] It ensures that the random numbers generated are within specified boundaries.
- [x] It can be used to specify which random number generating algorithm R should use, ensuring consistency and reproducibility.

✓ **Correct**

4. Which function can be used to evaluate the inverse cumulative distribution function for the Poisson distribution? **1 / 1 point**

- (selected) qpois
- rpois
- dpois
- ppois

✓ **Correct**

Probability distribution functions beginning with the `q` prefix are used to evaluate the quantile (inverse cumulative distribution) function.

5. What does the following code do? **1 / 1 point**

```
set.seed(10)
x <- rep(0:1, each = 5)
y <- rnorm(10, 0, .08)
y <- 0.5 + 2 * x + e
```

- Generate data from a Poisson generalized linear model
- (selected) Generate data from a Normal linear model
- Generate uniformly distributed random data
- Generate random exponentially distributed data

✓ **Correct**

6. What R function can be used to generate Binomial random variables? **1 / 1 point**

- pbinom
- dbinom
- qbinom
- (selected) rbinom

✓ **Correct**

7. What aspect of the R runtime does the profiler keep track of when an R expression is evaluated? **1 / 1 point**

- (selected) the function call stack
- the global environment
- the package search list
- the working directory

✓ **Correct**

8. Consider the following R code **1 / 1 point**

```
library(datasets)
Rprof()
fit <- lm(y ~ x1 + x2)
Rprof(NULL)
```

(Assume that `y`, `x1`, and `x2` are present in the workspace.) Without running the code, what percentage of the run time is spent in the `lm` function, based on the `by.total` method of normalization shown in `summaryRprof()`?

- 23%
- It is not possible to tell
- 50%
- (selected) 100%

✓ **Correct**

When using `by.total` normalization, the top-level function (in this case, `lm()`) always takes 100% of the time.

9. When using 'system.time()', what is the user time? **1 / 1 point**

- (selected) It is the time spent by the CPU evaluating an expression
- It is the "wall-clock" time it takes to evaluate an expression
- It is the time spent by the CPU waiting for other tasks to finish
- It is a measure of network latency

✓ **Correct**

10. If a computer has more than one available processor and R is able to take advantage of that, then which of the following is true when using 'system.time()'? **1 / 1 point**

- user time is 0
- (selected) elapsed time may be smaller than user time
- elapsed time is 0
- user time is always smaller than elapsed time

✓ **Correct**