**Feedback — Week 4 Quiz \*\*Please Note: No Grace Period\*\*** [**Help**](https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://courserahelp.zendesk.com/hc/articles/201523125-Quizzes)

Thank you. Your submission for this quiz was received.

You submitted this quiz on **Mon 26 Jan 2015 11:12 PM EST**. You got a score of **10.00** out of **10.00**.

Top of Form

**Question 1**

What is produced at the end of this snippet of R code?

set.seed(1)

rpois(5, 2)

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| A vector with the numbers 3.3, 2.5, 0.5, 1.1, 1.7 |  |  |  |
| A vector with the numbers 1, 4, 1, 1, 5 |  |  |  |
| A vector with the numbers 1, 1, 2, 4, 1 | Correct | 1.00 | Because the `set.seed()' function is used, `rpois()' will always output the same vector in this code. |
| It is impossible to tell because the result is random |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 2**

What R function can be used to generate standard Normal random variables?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| qnorm |  |  |  |
| dnorm |  |  |  |
| rnorm | Correct | 1.00 | Functions beginning with the `r' prefix are used to simulate random variates. |
| pnorm |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question Explanation**Standard probability distributions in R have a set of four functions that can be used to simulate variates, evaluate the density, evaluate the cumulative density, and evaluate the quantile function.

**Question 3**

When simulating data, why is using the set.seed() function important?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| It ensures that the sequence of random numbers is truly random. |  |  |  |
| It ensures that the sequence of random numbers starts in a specific place and is therefore reproducible. | Correct | 1.00 |  |
| It ensures that the random numbers generated are within specified boundaries. |  |  |  |
| It can be used to generate non-uniform random numbers. |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 4**

Which function can be used to evaluate the inverse cumulative distribution function for the Poisson distribution?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| ppois |  |  |  |
| dpois |  |  |  |
| rpois |  |  |  |
| qpois | Correct | 1.00 | Probability distribution functions beginning with the `q' prefix are used to evaluate the quantile (inverse cumulative distribution) function. |
| Total |  | 1.00 / 1.00 |  |

**Question 5**

What does the following code do?

set.seed(10)

x <- rep(0:1, each = 5)

e <- rnorm(10, 0, 20)

y <- 0.5 + 2 \* x + e

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Generate uniformly distributed random data |  |  |  |
| Generate random exponentially distributed data |  |  |  |
| Generate data from a Poisson generalized linear model |  |  |  |
| Generate data from a Normal linear model | Correct | 1.00 |  |
| Total |  | 1.00 / 1.00 |  |

**Question 6**

What R function can be used to generate Binomial random variables?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| qbinom |  |  |  |
| dbinom |  |  |  |
| rbinom | Correct | 1.00 |  |
| pbinom |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 7**

What aspect of the R runtime does the profiler keep track of when an R expression is evaluated?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| the working directory |  |  |  |
| the package search list |  |  |  |
| the global environment |  |  |  |
| the function call stack | Correct | 1.00 |  |
| Total |  | 1.00 / 1.00 |  |

**Question 8**

Consider the following R code

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()'?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| It is not possible to tell |  |  |  |
| 23% |  |  |  |
| 50% |  |  |  |
| 100% | Correct | 1.00 | When using `by.total' normalization, the top-level function (in this case, `lm()') always takes 100% of the time. |
| Total |  | 1.00 / 1.00 |  |

**Question 9**

When using 'system.time()', what is the user time?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| 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 |  |  |  |
| It is the time spent by the CPU evaluating an expression | Correct | 1.00 |  |
| Total |  | 1.00 / 1.00 |  |

**Question 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()'?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| elapsed time may be smaller than user time | Correct | 1.00 |  |
| user time is always smaller than elapsed time |  |  |  |
| user time is 0 |  |  |  |
| elapsed time is 0 |  |  |  |
| Total |  | 1.00 / 1.00 |  |

Bottom of Form