Back

Week 4 Quiz

Quiz, 10 questions

Question 1

1  
point

**1. Question 1**

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



1

2

set.seed(1)

rpois(5, 2)



A vector with the numbers 1, 4, 1, 1, 5



It is impossible to tell because the result is random



A vector with the numbers 1, 1, 2, 4, 1



A vector with the numbers 3.3, 2.5, 0.5, 1.1, 1.7

Question 2

1  
point

**2. Question 2**

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



pnorm



rnorm



qnorm



dnorm

Question 3

1  
point

**3. Question 3**

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



It can be used to generate non-uniform random numbers.



It ensures that the sequence of random numbers starts in a specific place and is therefore reproducible.



It ensures that the sequence of random numbers is truly random.



It ensures that the random numbers generated are within specified boundaries.

Question 4

1  
point

**4. Question 4**

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



dpois



rpois



qpois



ppois

Question 5

1  
point

**5. Question 5**

What does the following code do?



1

2

3

4

set.seed(10)

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

e <- rnorm(10, 0, 20)

y <- 0.5 + 2 \* x + e



Generate data from a Normal linear model



Generate uniformly distributed random data



Generate random exponentially distributed data



Generate data from a Poisson generalized linear model

Question 6

1  
point

**6. Question 6**

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



pbinom



rbinom



dbinom



qbinom

Question 7

1  
point

**7. Question 7**

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



the function call stack



the working directory



the global environment



the package search list

Question 8

1  
point

**8. Question 8**

Consider the following R code



1

2

3

4

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()'?



100%



23%



50%



It is not possible to tell

Question 9

1  
point

**9. Question 9**

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



It is a measure of network latency



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 the time spent by the CPU evaluating an expression

Question 10

1  
point

**10. 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()'?



user time is always smaller than elapsed time



elapsed time is 0



user time is 0



elapsed time may be smaller than user time

Upgrade to submit