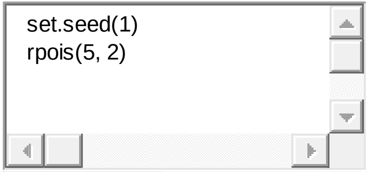
Coursera R Programming WEEK 1 Solutions

1.

Question 1

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

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



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

1 / 1 point

○

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

●

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

○

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

○

It is impossible to tell because the result is random

Correct

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

2.

Question 2

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

1 / 1 point

○

pnorm

●

rnorm

○

dnorm

○

qnorm

Correct

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

3.

Question 3

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

1 / 1 point

□

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

□

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

□

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

☑

It can be used to specify which random number generating algorithm R should use, ensuring consistency and reproducibility.

Correct

4.

Question 4

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

1 / 1 point

○

dpois

○

ppois

●

qpois

○

rpois

Correct

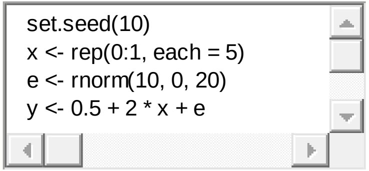
Probability distribution functions beginning with the `q' prefix are used to evaluate the quantile (inverse cumulative distribution) function.

5.

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
```



```
set.seed(10)
x <- rep(0:1, each = 5)
e <- rnorm(10, 0, 20)
y <- 0.5 + 2 * x + e
```

1 / 1 point

○

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

6.

Question 6

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

1 / 1 point

●

rbinom

○

dbinom

○

pbinom

○

qbinom

Correct

7.

Question 7

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

1 / 1 point

○

the global environment

●

the function call stack

○

the working directory

○

the package search list

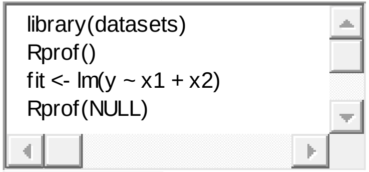
Correct

8.

Question 8

Consider the following R code

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



```
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()'?

1 / 1 point

○

23%

●

100%

○

50%

○

It is not possible to tell

Correct

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

9.

Question 9

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

1 / 1 point

●

It is the time spent by the CPU evaluating an expression

○

It is a measure of network latency

○

It is the time spent by the CPU waiting for other tasks to finish

○

It is the "wall-clock" time it takes to evaluate an expression

Correct

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()'?

1 / 1 point

●

elapsed time may be smaller than user time

○

user time is 0

○

user time is always smaller than elapsed time

○

elapsed time is 0

Correct