

Mango

Christopher Hughes

17 February 2017

R Coding Test

1. Create a vector of length 26, containing samples from a normal distribution

```
norm_26 <- rnorm(26)
```

2. Print the minimum, maximum, median, 25% and 75% quantiles of this data.

```
# Exclude the mean - fourth element in the vector  
summary(norm_26)[-4]
```

```
##      Min. 1st Qu.  Median 3rd Qu.    Max.  
## -1.4800 -0.5862 -0.1587  0.3502   1.0010
```

3. Create a 10x10 matrix of samples from a Poisson distribution (lambda=1).

```
n <- 100  
d <- 10  
pois_100 <- matrix(rpois(n, 1), d)
```

4. Print the maximum of each row of the matrix.

```
# max columns randomly gives the index when there is a tie  
max_i <- max.col(pois_100)  
# index the original values with max_i nice to have the index  
pois_100[cbind(seq(10),max_i)]
```

```
## [1] 3 4 2 2 2 4 2 4 3 2
```

```
# Or use apply - tidier solution  
apply(X = pois_100, MARGIN = 1, FUN = max)
```

```
## [1] 3 4 2 2 2 4 2 4 3 2
```

```
# Use apply with an anonymous function - flexible  
row_max <- apply(X = pois_100, MARGIN = 1, FUN = function(x){max(x)})
```

5. Store the created vector and matrix in a “list” structure

```
list_r <- list(pois_100, row_max)
```

6. Print the length of each component of the list.

```
lapply(list_r,FUN = length)
```

```
## [[1]]  
## [1] 100  
##  
## [[2]]  
## [1] 10
```

7. Create a function to output the maximum of each row of a matrix. Then use this function to re-run the question 4.

```
row_max <- function(x){
  apply(X = x, MARGIN = 1, FUN = function(x){max(x)})
}
row_max(pois_100)
```

```
## [1] 3 4 2 2 2 4 2 4 3 2
```

Extension

1. Write a function that will simulate numbers from a given distribution. The function should take one argument for the number of samples and a second argument which specifies the distribution (Normal, Poisson or Binomial). The function should be able to handle additional parameters depending on the distribution chosen, e.g. a 'lambda' for Poisson samples.

```
sim_values <- function(n, dist = "Normal", ...) {

  if (tolower(dist) == "normal") {
    mean <- list(...)$mean
    sd <- list(...)$sd
    if (is.null(mean)) {
      mean <- 0
    }
    if (is.null(sd)) {
      sd <- 1
    }
    rnorm(n, mean, sd)
  }

  else if (tolower(dist) == "poisson"){
    lambda <- list(...)$lambda
    if (is.null(lambda)) {
      lambda <- 1
    }
    rpois(n, lambda)
  }

  else if (tolower(dist) == "binomial") {
    size <- list(...)$size
    prob <- list(...)$prob
    if (is.null(size)) {
      size <- 10
    }
    if (is.null(prob)) {
      prob <- 0.5
    }
    rbinom(n, size, prob)
  }

}
```

We can see the function in action:

```
library(ggplot2)
df_sim <- data.frame(Values = sim_values(1000, dist = "NORMAL", mean = 5))

ggplot2::ggplot() + ggplot2::geom_histogram(data = df_sim, mapping = aes(Values), bins = 30, fill = "red")
```

