

MA386 Statistical Programming HW 4

Due Friday, April 11, 3 PM

Instructions: You should document your functions in this assignment, especially your function for problem 1. In addition you should, as usual, use indentation to improve readability.

1: Many probability problems are difficult to solve analytically but easy to solve using simulation. In this problem you will use simulation to analyze a two-dimensional random walk problem in which a particle moves randomly about a 3 by 3 array or matrix M of locations. Initially, at iteration 1, the particle is in the center of the matrix, i.e., at location $M[2, 2]$. By the next iteration, iteration 2, the particle has randomly moved to one of the 8 adjacent elements. By iteration 3, it has again jumped to an adjacent element. At each iteration, the particle has an equally likely chance of jumping to any of the elements adjacent to its current location. For example, if the particle is at location $M[1, 1]$ then there are three adjacent locations, $M[1, 2]$, $M[2, 2]$, and $M[2, 1]$. Thus there is a $1/3$ probability it will jump from $M[1, 1]$ to $M[2, 2]$ and so on.

You are to write a function called `randwalk` which has one argument, `numiter`, the number of iterations `randwalk` is to simulate. The return value of `randwalk` is a 3 by 3 matrix P where element $P[i, j]$ contains the fraction of iterations in which the particle was in location $M[i, j]$. So, for example, if you call `randwalk` with `numiter = 2` and the particle ends up in location $M[2, 3]$ on iteration 2 then `randwalk` returns P where $P[2, 2] = P[2, 3] = 1/2$ and the remaining elements of P are zero.

E-mail me a copy of the file defining `randwalk` **plus** a Word (.docx) or a notepad (.txt) file containing your values for P based on 10000 iterations.

2: In this problem you will construct a function to do a two-sided, two-sample permutation test comparing the means of two populations/processes and then assess its performance by doing a simulation study. Here is the outline of your function

```
permtest <- function(x,y,N=10000){  
  ...  
  commands  
  ...  
  return(pval)  
}
```

where the vectors `x` and `y` contain the two samples, `N` is the number of permutations, and `pval` is the permutation test p-value. (The mechanics of how to compute the permutation p-value are provided on the next page.) To keep the focus on ideas rather than minor details, we will assume that there are no missing values (NA's) in `x` or `y`. Once you have coded `permtest`, do the simulation study described on the next page.

2 continued: permtest simulation study.

1. Generate two samples of size 20 from a standard normal distribution 1000 times using `rnorm()`. For each of these 1000 pairs of samples, compute the corresponding p-value for testing $H_o: \mu_x = \mu_y$ vs. $H_a: \mu_x \neq \mu_y$ using `permtest`. Determine the fraction of these 1000 sample pairs for which the p-value is less than or equal to 0.05.
2. Repeat step 1 but when creating the second sample of size 20 (the vector `y`), use `rnorm(20,mean=1)`.
3. Repeat step 2 but when creating the second sample of size 20 (the vector `y`), use `rnorm(20,mean=2)`.

If the simulation study behaves as expected, you should get a p-value of less than or equal to 0.05 for about 5% of the 1000 sample pairs in step 1. In step 2 should get a greater percentage of small p-values and an even greater percentage in step 3.

Once you have finished the simulation study, send me an e-mail providing your `permtest` code plus your simulation study results.

Permutation p-value recipe:

Here is the algorithm for computing the permutation p-value:

1. Let the values in vector `x` have label “1” (`x` contains sample 1 values) and values in vector `y` have label “2” (`y` contains sample 2 values).
2. Compute $|\bar{X}_1 - \bar{X}_2|$, the absolute value of the difference between the sample mean for sample 1, \bar{X}_1 , and the sample mean for sample 2, \bar{X}_2 .
3. Compute $|\bar{X}_1^* - \bar{X}_2^*|$, the absolute value of the difference between the sample means for a random permutation of the sample labels (1’s and 2’s).
4. Repeat step 3 $N - 1$ additional times where N is a large number, 10000 by default.
5. Determine the number of permutations N_p for which

$$|\bar{X}_1^* - \bar{X}_2^*| \geq |\bar{X}_1 - \bar{X}_2|$$

6. The permutation p-value is then $\text{pval} = N_p/N$.