**Analytical Methods – mySums explained**

We will create a dataset of random numbers uniformly distributed from 0 to 1. We will create a new dataset as follows: For each row, we need to take the sum of the first three columns and the last three columns. Whichever set of numbers gives the highest sum should go in the first three columns. So, there is a chance that we will have to swap columns 1, 2 and 3 for columns 4, 5, and 6 in the new dataset. **How to code this?**

|  |  |
| --- | --- |
| **Code** | **Explanation** |
| N=100;  X=rand(N,6); | We start by deciding how big our sample will be (100). Next, we will create the random numbers we need for our exercise. This is done via the function rand. |
| S1=X(:,1:3);  S2=X(:,4:6); | We have two sets of data within X. It’s best if we separate them into two different groups from the beginning for clarification. We will call these two groups:   * S1 (columns 1 through 3 of X) * S2 (columns 4 through 6 of X) |
| Xnew=zeros(size(X)); | We will manipulate the values inside X (now conveniently divided into S1, S2) to create a new dataset. We can just overwrite the values inside X, but this is problematic because it’s hard to keep track of coding problems that way.  Instead, let’s create an empty matrix, of identical size as X, but filled with zeroes. We will fill Xnew after we compare the values inside S1 and S2. |
| for i=1:1:N    S1chunk=S1(i,:);  S2chunk=S2(i,:); | For every observation, we need to compare the sum of the cells in S1 with the sum of the cells in S2. This part of the code separates these “chunks” for ease of access. |
| if sum(S1chunk)>sum(S2chunk)  Xnew(i,1:3)=S1chunk;  Xnew(i,4:6)=S2chunk; | There can be three scenarios when comparing the sums from the S1 and S2 chunks.  **Scenario 1:** *The sum of S1’s chunk is larger than the sum of S2’s chunk*  This is the case shown here. If so, the new dataset will have S1 and S2’s chunks in the same position. |
| elseif sum(S2chunk)>sum(S1chunk)  Xnew(i,1:3)=S2chunk;  Xnew(i,4:6)=S1chunk; | **Scenario 2:** *The sum of S2’s chunk is larger than the sum of S1’s chunk.*  If our data follows this scenario, then we need to move the S2 chunk in the first position of the new dataset, and conversely for S1. |
| else  judge=rand();  if judge<=.5  Xnew(i,1:3)=S1chunk;  Xnew(i,4:6)=S2chunk;  else  Xnew(i,1:3)=S2chunk;  Xnew(i,4:6)=S1chunk;  end  end  end | **Scenario 3:** *The sums are equal*  In this case it’s up to the researcher. One can simply leave the data as is. To make the problem more challenging we can define the following rule:  *Flip a coin. If Tails, then S1 goes first, then S2.*  *If Heads, then S2 goes first, then S2.*  To “flip a coin” in MATLAB, one can draw a random number (Uniform) from 0 to 1. We will call it “judge”. This number can take any number from 0 to 1 with **the same probability.** Thus, there is half a chance that judge is between 0 and .5 (Tails), and half a chance it’s bigger (Heads). Depending on judge’s value, we will assign S1 and S2’s chunks accordingly. |

☺ Code complete! ☺