Joanna Masikowska

B9TB1710

CAPS08_B9TB1710.m

```
#logical indexing of matrices

x1=10; #number of trials {10,100,1000,10000,100000}
Po1=zeros(1,x1); #P of orange
for i=1:10
  B=rand(1,x1) < 0.4; #1 for red box; 0 for blue box
  Frnd=rand(1,x1); #probability of choosing a fruit
  F(B==1)=Frnd(B==1)<2/8; #1 for apple; 0 for orange
  F(B==0)=Frnd(B==0) < 3/4; #1 for apple; 0 for orange
  #P of red and orange #P of blue and orange
  Po1(1,i)=sum(F==0&B==1)/x1+sum(F==0&B==0)/x1;
endfor
P1=sum(Po1)/10
v1=var(Po1)


x2=100;
Po2=zeros(1,x2);
for i=1:10
  B=rand(1,x2) < 0.4; #1 for red box; 0 for blue box
  Frnd=rand(1,x2); #probability of choosing a fruit
  F(B==1)=Frnd(B==1)<2/8; #1 for apple; 0 for orange
  F(B==0)=Frnd(B==0) < 3/4; #1 for apple; 0 for orange
  #P of red and orange #P of blue and orange
  Po2(1,i)=sum(F==0&B==1)/x2+sum(F==0&B==0)/x2;
endfor
P2=sum(Po2)/10
v2=var(Po2)


x3=1000;
```

```
v2=var(Po2)


x3=1000;
Po3=zeros(1,x3);
for i=1:10
  B=rand(1,x3) < 0.4; #1 for red box; 0 for blue box
  Frnd=rand(1,x3); #probability of choosing a fruit
  F(B==1)=Frnd(B==1)<2/8; #1 for apple; 0 for orange
  F(B==0)=Frnd(B==0) < 3/4; #1 for apple; 0 for orange
  #P of red and orange #P of blue and orange
  Po3(1,i)=sum(F==0&B==1)/x3+sum(F==0&B==0)/x3;
endfor
P3=sum(Po3)/10
v3=var(Po3)


x4=10000;
Po4=zeros(1,x4);
for i=1:10
  B=rand(1,x4) < 0.4; #1 for red box; 0 for blue box
  Frnd=rand(1,x4); #probability of choosing a fruit
  F(B==1)=Frnd(B==1)<2/8; #1 for apple; 0 for orange
  F(B==0)=Frnd(B==0) < 3/4; #1 for apple; 0 for orange
  #P of red and orange #P of blue and orange
  Po4(1,i)=sum(F==0&B==1)/x4+sum(F==0&B==0)/x4;
endfor
P4=sum(Po4)/10
v4=var(Po4)
```

My code consists of 4 loops, each for one trial number being {10,100,1000,10000}. In each loop I generate a matrix of random numbers with 1 row and xi columns, where xi stands for number of trials. I compare each element of the matrix with number 0.4 (standing for probability of choosing red box). If the equality is true, the number 1 is assigned to the same index in the new matrix. Otherwise, number 0 is assigned. Resulting matrix with zeros and ones is called B. Next, I again generate a matrix of random numbers Frnd with 1 row and xi columns. It consist of probabilities of picking up orange or apple. For indices corresponding to red box in matrix B, I check if the probability point to apple or to orange for this box. I do analogically with blue box. I sum up the occurrence of each event (oranges from red box, apples from blue box, oranges from blue box, apples from red box) and divide by number of trials. The process is repeated 10 times and each time, probability of picking an orange from red box, is stored in matrix Po_{i}. By summing the values from the matrix and dividing by 10, I get the average probability corresponding to this number of trials. Then, I compute the variance of the data in matrix Po_{i}. My results are as below.

```
コマンドウィンドウ
GNU Octave, version 5.2.0
Copyright (C) 2020 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "x86_64-w64-mingw32".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> CAPS08_B9TB1710

P1 =  0.41000
v1 =  0.036556
P2 =  0.44400
v2 =  0.018205
P3 =  0.44630
v3 =  0.0019762
P4 =  0.44892
v4 =  0.00020138
>>
```

Number of trials x	P(o) after 10 times x trials	Variance
10	0.4100	0.036556
100	0.44400	0.018205
1000	0.44630	0.0019762
10000	0.44892	0.00020138

I can see that the more data I have, the smaller the variance is. In all cases probability is about ~0.44 which is almost identical to the value calculated by hand 9/20.