**Homework Monte Carlo methods**

**Student: Hongxing NIU; Matricule ULB: 000342366**

1. **Monte Carlo methods**

Consider the ratio of integrals,

(a). Use **rejection sampling** to generate data X from and estimate by

*PROCEDURE*

(1). Generate a random number from a normal distribution

(2). Compute

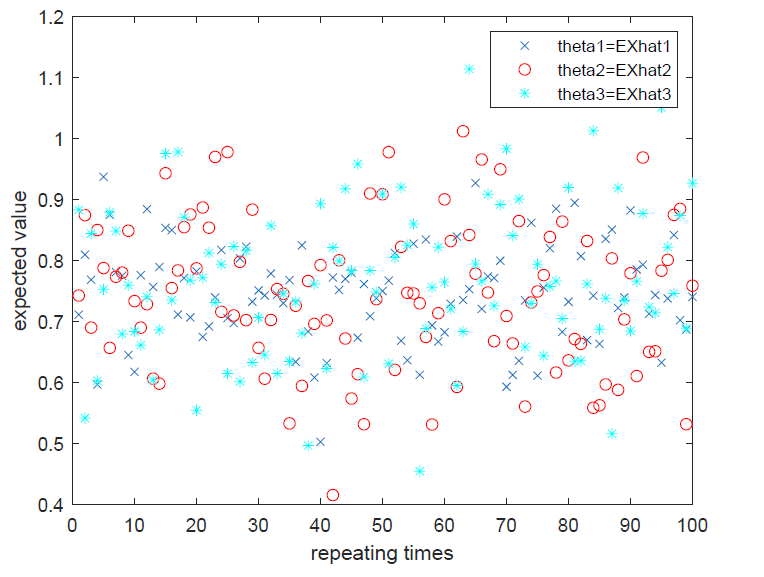
(3). Generate a uniform random number

(4). If accept , else reject and go to step 1

(b). Generate normal data and estimate *EX* by with

(c). Generate Cauchy data and estimate *EX* by with

The simulation results using above three samplers are shown by the following figure.



Here, we took n = 2000 samples and repeated 100 times for each of the three samplers;

other parameters were a = -3, b = 2, stdev = 4, mu = 5.

In Matlab,

;

;

;

;

;

-----------------------------

Sample Size Variance

-----------------------------

1 100 0.0057

2 100 0.0185

3 100 0.0171

-----------------------------

* Perform Levene’s Test for homogeneity/equality of Variances, i.e, whether

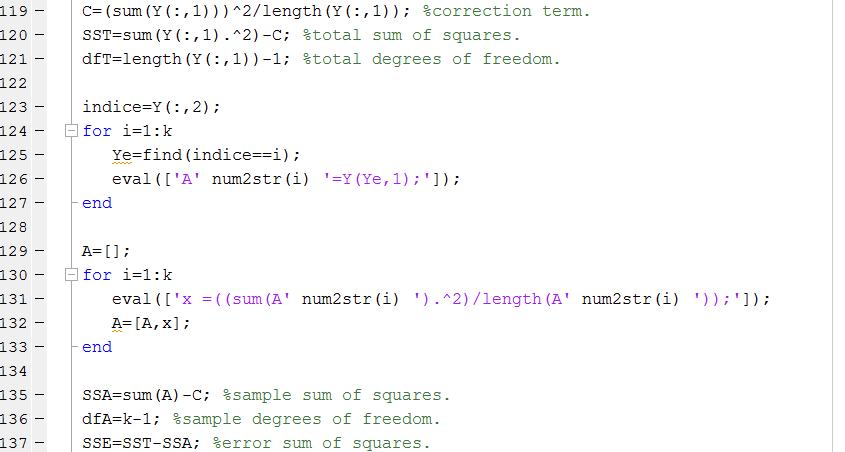
The test statistic is

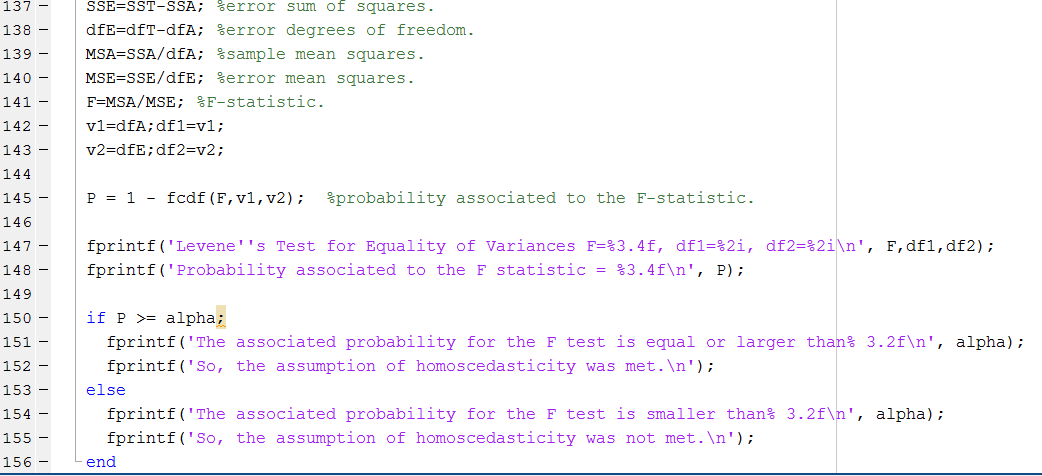
where *Ni* is the sample size of the *i*th group, and *k* is the number of groups.

In the Levene's test, the data are transforming to and uses the *F* distribution performing an one-way ANOVA using Z as the dependent variable (Brownlee, 1965; Miller, 1986)].

In our case, k=3, N=3\*100=300, Ni=100,

The Matlab code for Levene's Test for Equality of Variances is shown below,





The results of Levene's are as follows,

F=6.2700, df1= 2, df2=297;

Probability associated to the F statistic = 0.0022;

The associated probability for the F test is smaller than 0.05;

So, the assumption of homoscedasticity was not met, i.e., the null hypothesis of is rejected.

**2. Markov Chain Monte Carlo sampling**

The matlab file *randgammaMCMC.m* implements a Metropolis-Hastings algorithm for the generation of pseudo random variables following a Gamma distribution. It is an alternative to the routine *randgamma.m*, which uses rejection sampling. The file *testrandgammaMCMC.m* compares both generators.

(a). Knowing this, explain how to arrive at the lines of code in *randgammaMCMC.m*

*PROCEDURE – METROPOLIS-HASTINGS SAMPLER*

(1). Initialize the chain to *X=zeros(m,1)* and set *Y = randexp(1, r, lambda), X(1) = sum(Y),* where *m=sample size=10,000, r = floor(alfa)*. Set *k=2*.

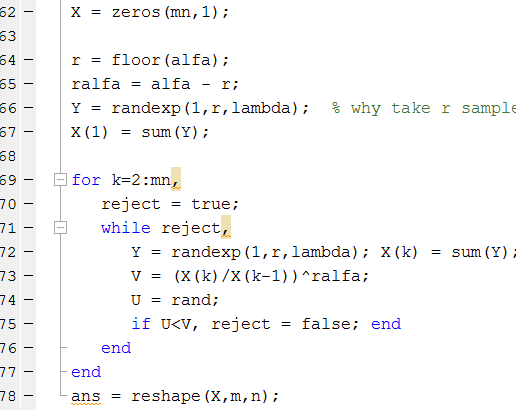
(2). Generate a candidate point *X(k) = sum(Y)* from a proposal distribution *Y = randexp(1, r, lambda)*. Note, the sum of *r* exponential (*lambda*) random variables is a random variable following an Erlang/Gamma (*r*, *lambda*) distribution, *i.e, .*

(3). Generate *U=rand* from a uniform distribution.

(4). If , then accept *X(k)*, else keep the value of *X(k-1)*.

(5). Set *k=2+1* and repeat steps (2) through (5) until *k≤m*.

In Matlab, above procedure is implemented by



(b). Run the file *testrandgammaMCMC.m* and explain in one or two sentences what you see. What could be a drawback of *randgammaMCMC.m* (MCMC sampling) compared to *randgamma.m* (rejection sampling).

Below figure compares the random variables by the method of rejection sampling and by the Metropolis-Hastings algorithm (MH). The MH is an MCMC technique that draws samples from a probability distribution where direct sampling is difficult.

The *cons* of MH are that (i) the simulations *x(k)* are correlated, hence less informative than i.i.d. simulations; (ii) the validation of the method is only asymptotic, hence there is an approximation in considering *x(k)* for a fixed *k* as a realization of  The *pros* of rejection sampling are that there is no approximation in the method: the outcome is truly an i.i.d. sample from .

