# QUESTION 2

clear; close all;

%% data

%storing 100 realisations of the RV X; mu = 1, sigma = 2;

X = randn([100,1])\*sqrt(2)+1;

Y = 3.\*X.^2 + 5.\*X;

%Verification of my function

my\_ans = covariance(X,Y);

matlab\_ans = cov([X,Y],1);

%% Checking convergence of sigma xy upto 10^4 samples

n = 10^4;%sample size

sigxy = zeros(n,1);

N = zeros(n,1);

for i = 1:n

Xs = randn([i,1])\*sqrt(2)+1;

Ys = 3.\*Xs.^2 + 5.\*Xs;

mat = covariance(Xs,Ys);

sigxy(i) = mat(2,1);

N(i) = i;

end

plot(N,sigxy,N,(zeros(n,1))+22);

xlabel('Sample size');

ylabel('sigma xy');

%% Checking convergence in the range of 10^6 samples

n\_vals = 10^5:10^5:10\*10^6;

sigxy2 = zeros(length(n\_vals),1);

for i=1:length(n\_vals)

Xs = randn([n\_vals(i),1])\*sqrt(2)+1;

Ys = 3.\*Xs.^2 + 5.\*Xs;

mat = covariance(Xs,Ys);

sigxy2(i) = mat(2,1);

end

figure();

plot(n\_vals,abs(sigxy2-22));

title('Deviation from true value');

xlabel('Sample size');

ylabel('deviation');

%% For writing in assignment

n\_vals = [10,100,10^3,10^4,10^5,10^6,2\*10^6,3\*10^6];

sigxy3 = zeros(length(n\_vals),1);

for i=1:length(n\_vals)

Xs = randn([n\_vals(i),1])\*sqrt(2)+1;

Ys = 3.\*Xs.^2 + 5.\*Xs;

mat = covariance(Xs,Ys);

sigxy3(i) = mat(2,1);

end

del\_est = abs(sigxy3-22);

% figure();

% plot(n\_vals,sigxy2,'x');

% xlabel('Sample size');

% ylabel('sigma xy');

%% Covariance Matrix Function

function mat = covariance(x,y)

sigxy = sum((x-mean(x)).\*(y-mean(y)))./length(x);

sigx = sum((x-mean(x)).^2)/length(x);

sigy = sum((y-mean(y)).^2)/length(y);

mat = [sigx sigxy; sigxy sigy];

end

# QUESTION 4

clear; close all;

%Part a

R = 200;

N = 10000;

dof = 10;

X = chi2rnd(dof,[N,R]);

%Uncomment the next line to visualise sample distribution

%hist(X(:,1),1:1:max(max(X)))

x\_hat\_vec = 8:0.001:11;

L = length(x\_hat\_vec);

Jvals = zeros(L,1);

for i = 1:L

Jvals(i) = sum(sum(abs(X-x\_hat\_vec(i))))/(N\*R);

end

[min\_J, pos] = min(Jvals);

plot(x\_hat\_vec,Jvals);

title('Cost Function');

xlabel('X hat');

ylabel('J');

x\_hat = x\_hat\_vec(pos);

%Part b

Pr\_x\_star = chi2cdf(1.1\*x\_hat,10) - chi2cdf(0.9\*x\_hat,10);

[M,V] = chi2stat(10);

Pr\_x = chi2cdf(1.1\*M,10) - chi2cdf(0.9\*M,10);