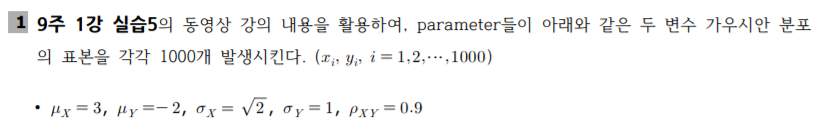
**Generation of Bivariate Gaussian Random Samples**

**and Estimation of Statistical Parameters**

**김지영**



- MATLAB Code

clear; clf;

mx=3; sx=sqrt(2); my=-2; sy=1; rho=0.9;

N=1000;

xy=fn\_Bivariate\_Gauss(mx, my, sx, sy, rho, N);

for i=1:N

plot(xy(i,1), xy(i,2), '.'); hold on;

end

axis ([-10 10 -6 6]); grid on;

function xy=fn\_Bivariate\_Gauss(mx, my, sx, sy, rho, N)

x=gaussrv(mx, sx, N);

tilde\_my=my+rho\*(sy/sx)\*(x-mx);

tilde\_sy=sqrt((sy^2)\*(1-rho^2));

y=gaussrv(tilde\_my, tilde\_sy, N);

xy=[x y];

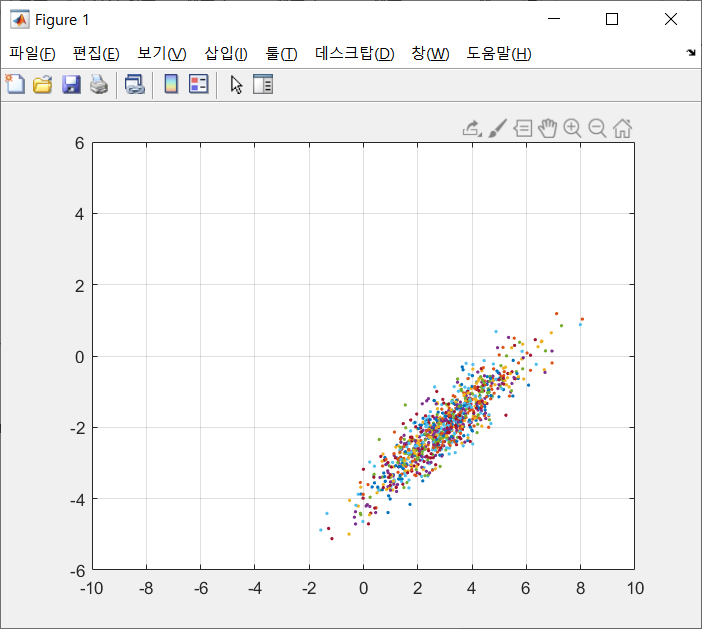
end

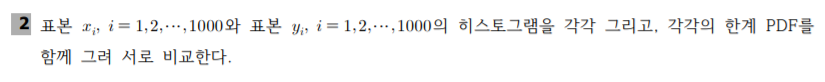
function x=gaussrv(mu,sigma,m)

x=mu +(sigma\*randn(m,1));

end

- Figure





- MATLAB Code

clear; clf;

mx=3; sx=sqrt(2); my=-2; sy=1; rho=0.9;

N=1000;

x=gaussrv(mx, sx, N);

y=gaussrv(my, sy, N);

x\_marginalpdf=gausspdf(mx, sx, x);

y\_marginalpdf=gausspdf(my, sy, y);

subplot(121), hold on, histogram(x, 'Normalization', 'pdf');

subplot(122), hold on, histogram(y, 'Normalization', 'pdf');

x\_pdf=floor(mx-4\*sx):0.01:ceil(mx+4\*sx);

y\_pdf=floor(my-4\*sx):0.01:ceil(mx+4\*sy);

subplot(121), plot(x\_pdf, gausspdf(mx, sx, x\_pdf), '-r', 'LineWidth', 2);

subplot(122), plot(y\_pdf, gausspdf(my, sy, y\_pdf), '-r', 'LineWidth', 2);

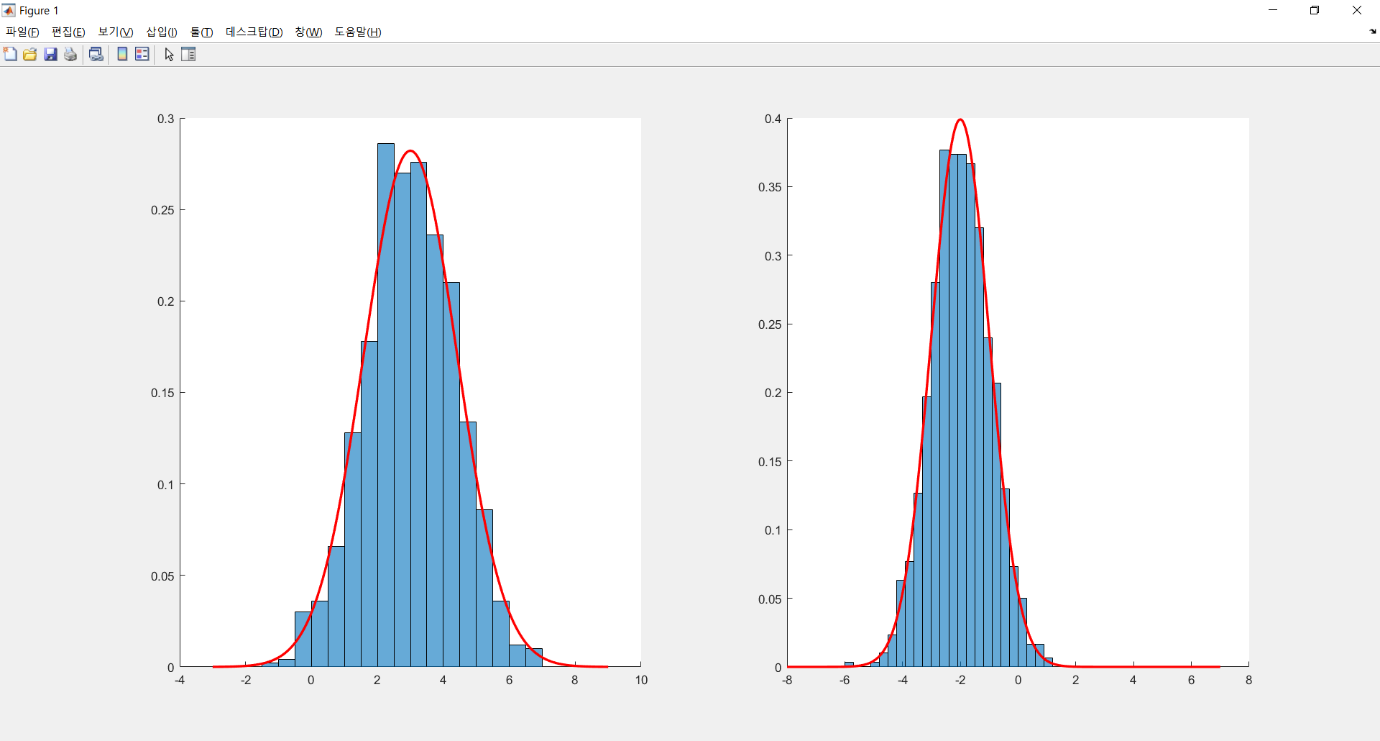
function f=gausspdf(mu,sigma,x)

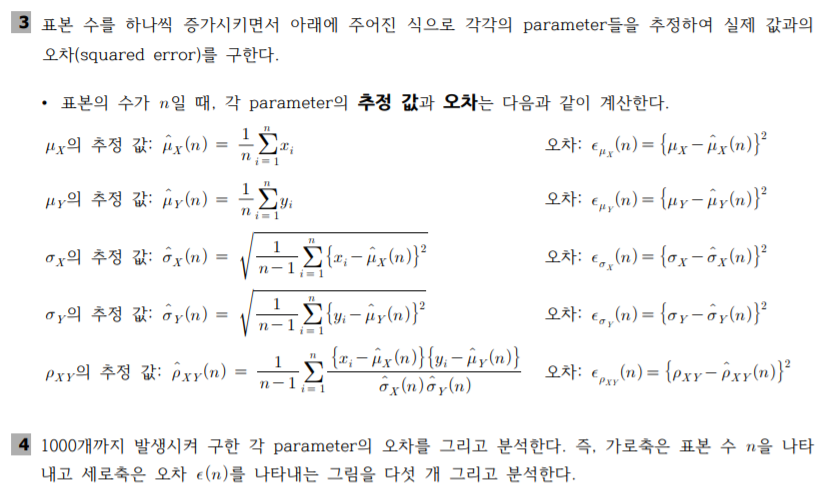
f=exp(-(x-mu).^2/(2\*sigma^2))/...

sqrt(2\*pi\*sigma^2);

end

- Figure

=> x, y 각각 히스토그램과 실제 한계 pdf분포가 비슷하게 나옴을 확인할 수 있습니다.



- MATLAB Code

clear; clf;

mx=3; sx=sqrt(2); my=-2; sy=1; rho=0.9;

n=1000;

x=gaussrv(mx, sx, n);

y=gaussrv(my, sy, n);

%--- mx추정

sum1=0;

for i=1:n

sum1=x(i,1)+sum1;

mx\_hat(i)=sum1./i;

error\_mx(i)=(mx-mx\_hat(i)).^2;

end

%--- my추정

sum2=0;

for i=1:n

sum2=y(i,1)+sum2;

my\_hat(i)=sum2./i;

error\_my(i)=(mx-mx\_hat(i)).^2;

end

%--- sigma x추정

sum3=0;

for i=2:n

sum3=(x(i,1)-mx\_hat(i)).^2+sum3;

sx\_hat(i)=sqrt((1/(i-1)).\*sum3);

error\_sx(i)=(sx-sx\_hat(i)).^2;

end

%--- sigma y추정

sum4=0;

for i=2:n

sum4=(y(i,1)-mx\_hat(i)).^2+sum4;

sy\_hat(i)=sqrt((1/(i-1)).\*sum4);

error\_sy(i)=(sy-sy\_hat(i)).^2;

end

%--- rho xy추정

sum5=0;

for i=2:n

sum5=((x(i,1)-mx\_hat(i))).\*((y(i,1)-my\_hat(i)))./(sx\_hat(i).\*sy\_hat(i))+sum5;

rho\_hat(i)=(1/(i-1)).\*sum5;

error\_rho(i)=(rho-rho\_hat(i)).^2;

end

N=1:n;

figure(1), plot(N, error\_mx(N), 'r'); title('mx추정'); hold on; grid on;

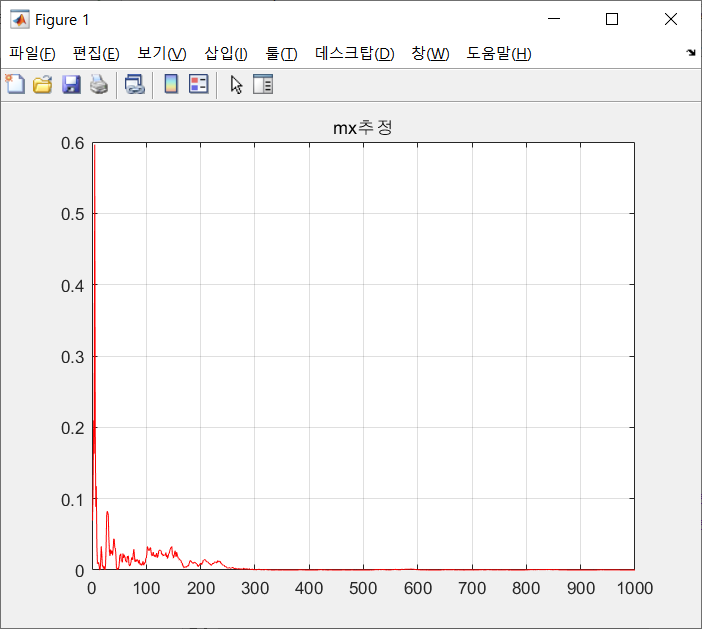
figure(2), plot(N, error\_my(N), 'r'); title('my추정'); hold on; grid on;

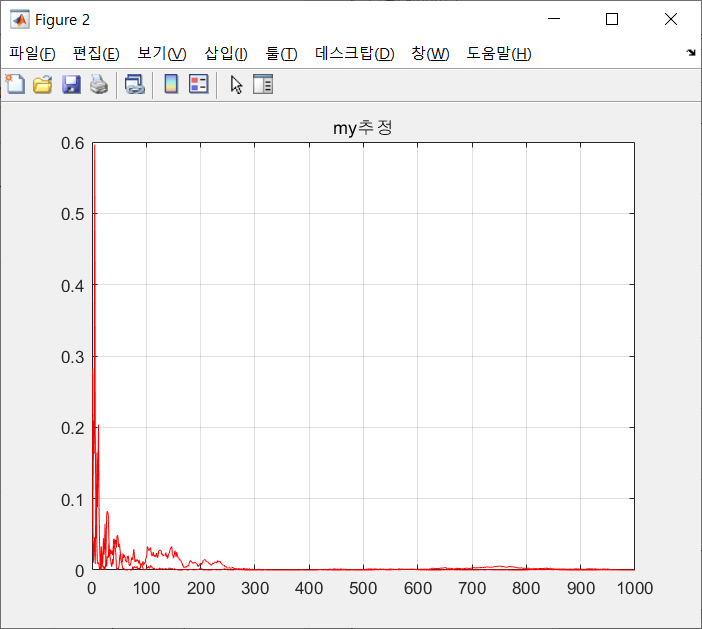
figure(3), plot(N, error\_sx(N), 'r'); title('sigma x추정'); hold on; grid on;

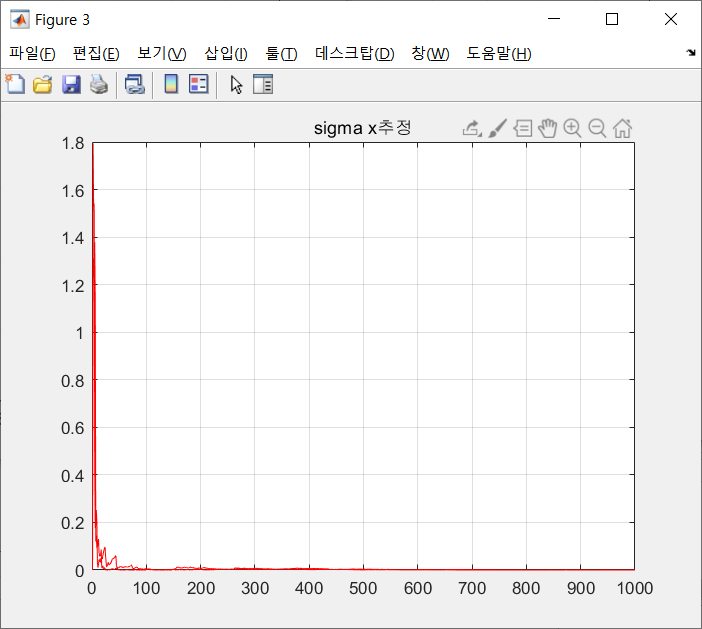
figure(4), plot(N, error\_sy(N), 'r'); title('sigma y추정'); hold on; grid on;

figure(5), plot(N, error\_rho(N), 'r'); title('rho xy추정'); hold on; grid on;

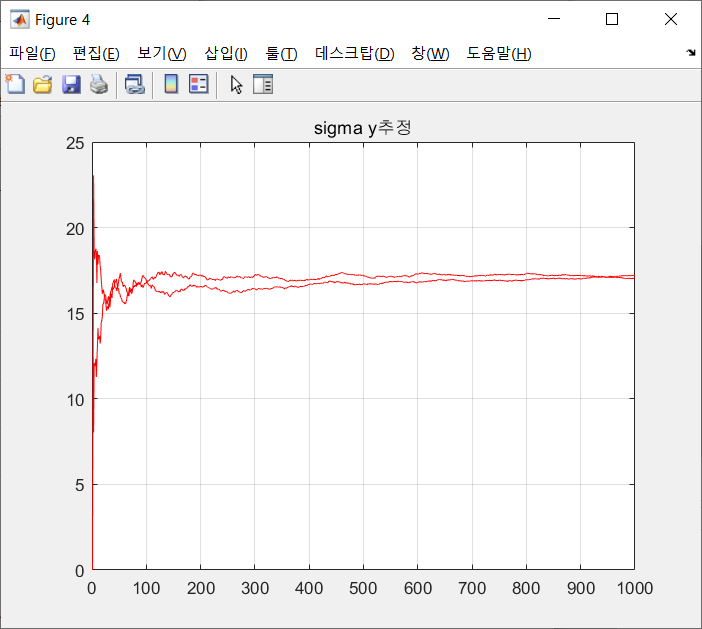
- Figure

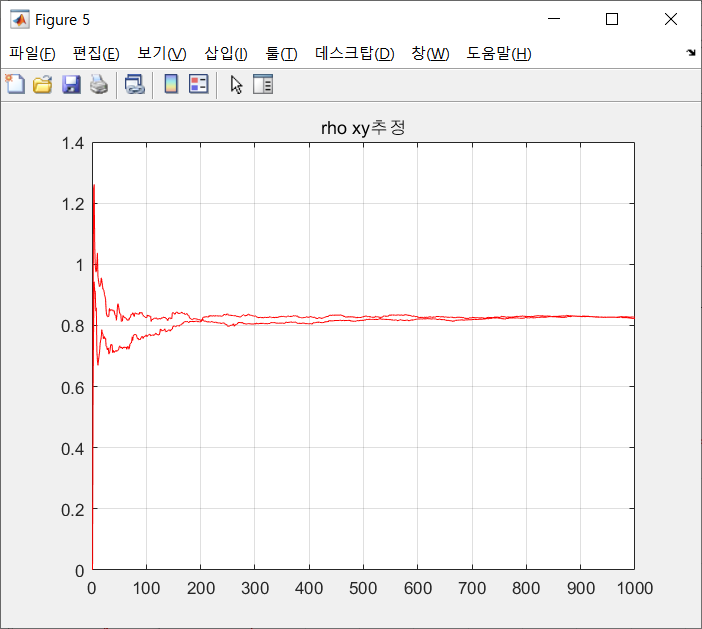




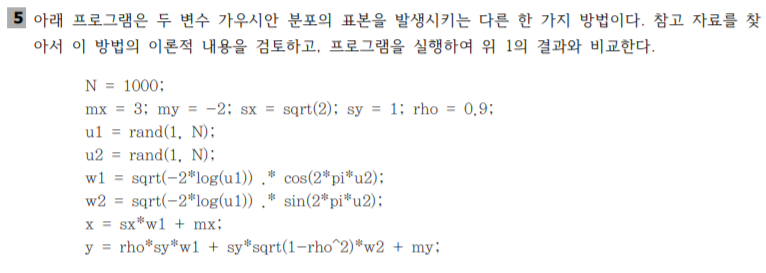


=> mx, my, sigma x를 추정한 결과, 표본 수가 증가할 수록 오차가 줄어들다 못해 거의 없습니다.





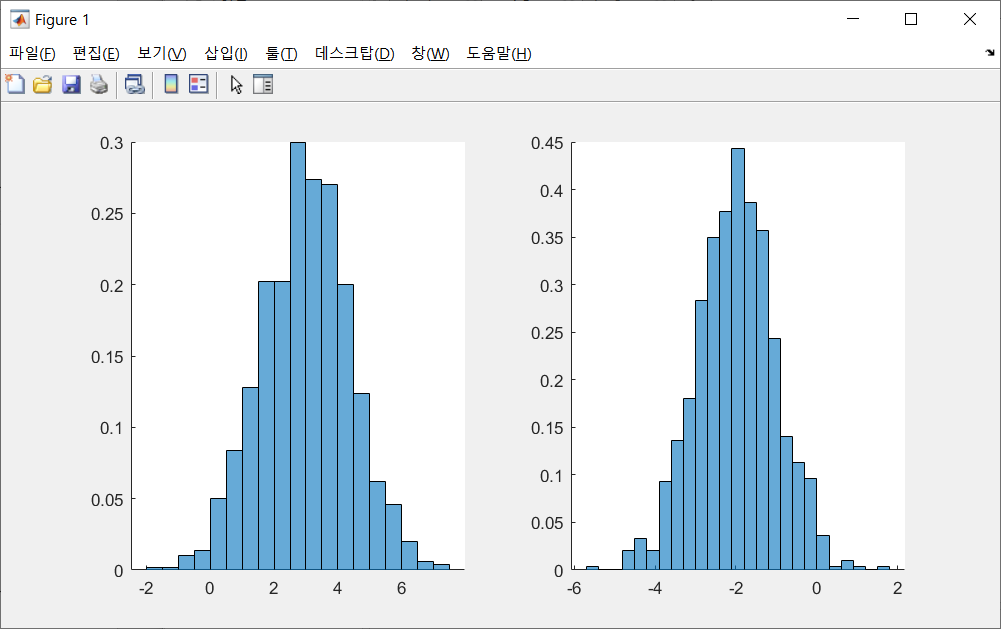
=> 반면, sigma y와 rho xy를 추정한 결과, 표본 수가 증가해도 특정 오차값이 계속 유지되며 변함이 없습니다.



subplot(121), hold on, histogram(x, 'Normalization', 'pdf');

subplot(122), hold on, histogram(y, 'Normalization', 'pdf');

- Figure



=> 문제2의 결과와 비슷하게 나온 모습을 확인할 수 있습니다.