

Variables aleatorias multidimensionales (ejemplo Matlab)

© 2020, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo en: <http://personales.upv.es/asala/YTV/vamdm.html>

Este código ejecutó correctamente en Matlab R2019b

Table of Contents

Generación de datos.....	1
Estudio de las distribuciones marginales.....	1
Media y Varianza.....	2
Momentos superiores marginales.....	3
Covarianza y correlación entre las variables.....	3

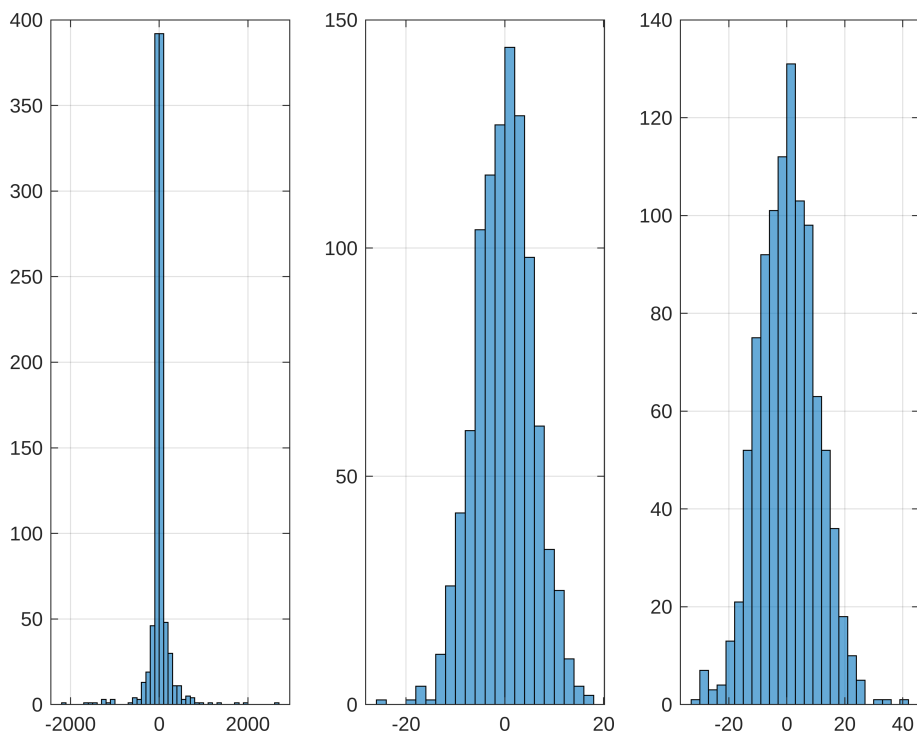
Generación de datos

```
N=1000;  
[x1,x2,x3]=formulasecreta(N)
```

```
x1 = 1×1000  
103 ×  
    -0.0167    -0.0141    1.3258    0.7727    0.0205    0.0646    0.2446    0.1340 ...  
x2 = 1×1000  
    -13.2538    -5.9825    5.1266    2.3186    -0.8204    3.4081    -9.5972    2.8402 ...  
x3 = 1×1000  
    -3.1956    -4.6547    25.8037    21.4535    6.8011    9.0681    17.4307    12.0926 ...
```

Estudio de las distribuciones marginales

```
%representemos graficamente los histogramas "marginales"  
subplot(1,3,1)  
histogram(x1), grid on  
subplot(1,3,2)  
histogram(x2), grid on  
subplot(1,3,3)  
histogram(x3), grid on
```



Media y Varianza

`%calculemos las medias marginales:`

```
medial=sum(x1)/N
```

```
medial = 6.0547
```

```
media2=sum(x2)/N
```

```
media2 = -0.0861
```

```
media3=sum(x3)/N
```

```
media3 = 0.3182
```

`%calculemos las variables incrementales:`

```
x1inc=x1-medial;
```

```
x2inc=x2-media2;
```

```
x3inc=x3-media3;
```

`%las varianzas (muestrales) marginales:`

```
var1=sum(x1inc.^2)/(N-1)
```

```
var1 = 5.8178e+04
```

```
var2=sum(x2inc.^2)/(N-1)
```

```
var2 = 32.8573
```

```
var3=sum(x3inc.^2)/(N-1)
```

```
var3 = 99.1970
```

```
var(x1) %también lo hace Matlab, claro!.
```

```
ans = 5.8178e+04
```

```
%desviación típica  
s1=std(x1)
```

```
s1 = 241.2017
```

```
s2=std(x2)
```

```
s2 = 5.7321
```

```
s3=std(x3)
```

```
s3 = 9.9598
```

```
%en distribución normal, el 98% de las muestras están entre +/- 2.33 veces  
%su desv. típica  
l11=-norminv(0.02/2)
```

```
l11 = 2.3263
```

```
length(find(-l11*s1<=x1inc & x1inc<=l11*s1))/N %tiene más outliers "extremos"...
```

```
ans = 0.9700
```

```
length(find(-l11*s2<=x2inc & x2inc<=l11*s2))/N
```

```
ans = 0.9780
```

```
length(find(-l11*s3<=x3inc & x3inc<=l11*s3))/N
```

```
ans = 0.9810
```

Momentos superiores marginales

```
% skewness (sesgo, asimetría, mediaxstd^3), kurtosis (prob. valores extremos, media xst  
% distrib. normal tiene sesgo=0, curtosis=3;  
skew=skewness([x1' x2' x3'])
```

```
skew = 1×3  
0.5738 -0.0834 0.0308
```

```
kur=kurtosis([x1' x2' x3'])
```

```
kur = 1×3  
40.0582 3.2577 3.2048
```

```
%la variable 1 tiene mucha más curtosis que la distr. normal.
```

Covarianza y correlación entre las variables

```
%la covarianza entre x1 y x2:
cov12=sum(x1inc.*x2inc)/(N-1)
```

```
cov12 = -396.6146
```

```
% mediante la matriz de datos:
X=[x1inc;x2inc;x3inc];
Sigma=X*X'/(N-1)
```

```
Sigma = 3×3
104 ×
    5.8178   -0.0397    0.1783
   -0.0397    0.0033   -0.0029
    0.1783   -0.0029    0.0099
```

```
cov(X') %Matlab lo tiene implementado... muestras en "filas"
```

```
ans = 3×3
104 ×
    5.8178   -0.0397    0.1783
   -0.0397    0.0033   -0.0029
    0.1783   -0.0029    0.0099
```

```
eig(Sigma)
```

```
ans = 3×1
104 ×
    0.0019
    0.0055
    5.8236
```

```
%si normalizamos todas las vbles a media 0, varianza 1:
x1incstd=x1inc/sqrt(var1);
x2incstd=x2inc/sqrt(var2);
x3incstd=x3inc/sqrt(var3);
Xstd=[x1incstd;x2incstd;x3incstd];
X2=zscore(X')'; %Matlab también lo tiene preparado.
%estodacero=X2-Xstd %esto da CERO
%obtenemos la "pairwise-correlation" matrix con variables normalizadas
Sigmastd=cov(Xstd')
```

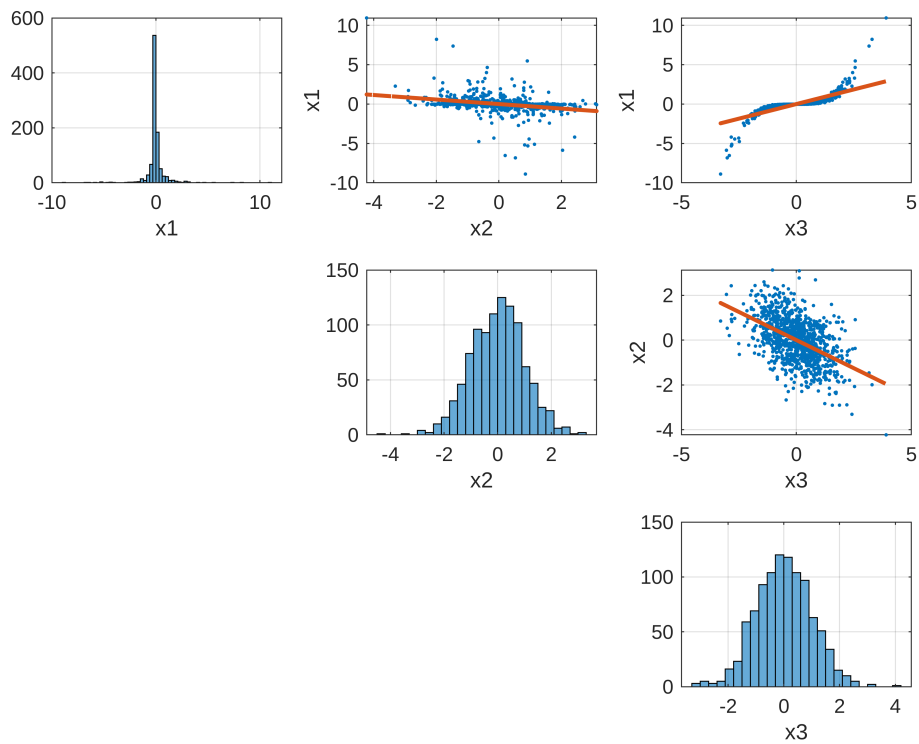
```
Sigmastd = 3×3
    1.0000   -0.2869    0.7423
   -0.2869    1.0000   -0.5026
    0.7423   -0.5026    1.0000
```

```
%dibujemos el pairwise scatterplot:
for i=1:3
    subplot(3,3,1+(i-1)*4)
    histogram(Xstd(i,:), xlabel('x'+string(i)), grid on)
    for j=(i+1):3
        subplot(3,3,(i-1)*3+j)
        plot(Xstd(j,:),Xstd(i,:),'.'), xlabel('x'+string(j)), ylabel('x'+string(i))
        hold on
        plot(Xstd(j,:),Xstd(j,:)*Sigmastd(i,j),'-.','LineWidth',2)
```

```

        grid on
        hold off
    end
end

```



```

function [x1,x2,x3]=formulasecreta(Nmuestras)
    p=randn(3,Nmuestras);
    q=[1 2 3;-2 -3 1;3 6 7]*p;
    x1=q(1,:).^3+randn(1,Nmuestras)*0.3;
    x2=q(2,:)*1.5;
    x3=q(3,:);
end

```