Multivariate normal distribution: confidence interval, confidence rectangles, confidence ellipsoid (Matlab example)

*Executed in Matlab R2023b

Presentations in video:

http://personales.upv.es/asala/YT/V/cfrc1EN.html

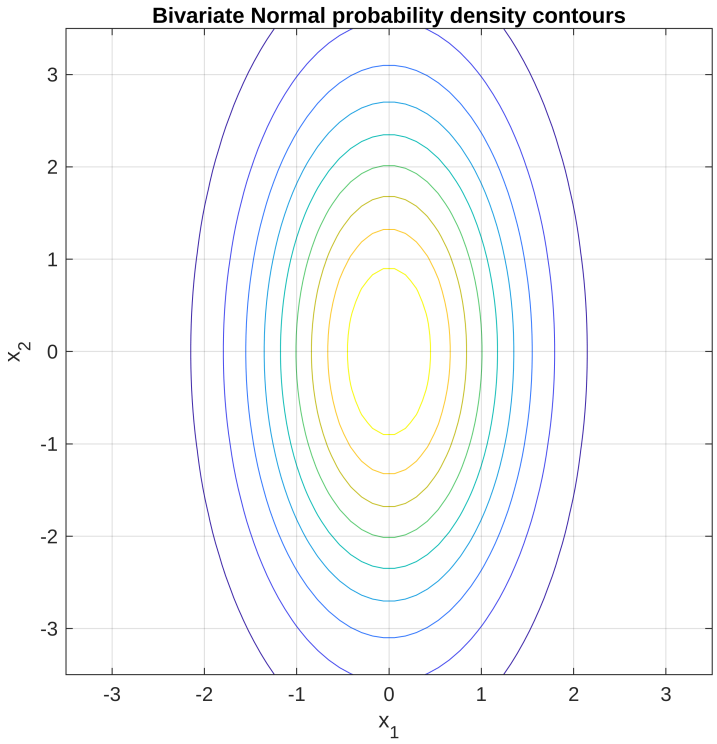
http://personales.upv.es/asala/YT/V/cfrcwrongEN.html

Goal: compare different rectangles, intervals and ellipsoids with a given probability of samples being there (confidence) for a multivariate normal.

```
syms x [2 1] real %for later manipulations
%CovMatrix=[1 1.1;1.1 4]; %correlated: CODE IS NOT VALID FOR THIS CASE
CovMatrix=diag([1 4]); %independent non standard, OK
%CovMatrix=eye(2); %standard normal, OK
[V,D]=eig(CovMatrix)
```

```
V = 2×2
     1     0
     0     1
D = 2×2
     1     0
     0     4
```

```
ezcontour(exp(-(0.5*x'*inv(CovMatrix)*x))/sqrt(2*pi*det(CovMatrix)),
[-3.5,3.5]), grid on, title("Bivariate Normal probability
density contours"), axis equal
```




```
sigma1=sqrt(CovMatrix(1,1))
```

```
sigma1 = 1
```

```
sigma2=sqrt(CovMatrix(2,2))
```

```
sigma2 = 2
```

```
errbnd=0.05;
confidence=1-errbnd %legend in plots may not be correct if changed
```

```
confidence = 0.9500
```

Marginal Conf. interval:

```
cfSingle=norminv(1-errbnd/2)
```

```
cfSingle = 1.9600
```

Joint confidence ellipsoid (ellipse in 2D) χ^2 bound:

```
dims=2;
chich=chi2inv(confidence,dims) %measured in "squared units"
```

```
chich = 5.9915
```

```
cfEllipse=sqrt(chich) %measured in "units"
```

```
cfEllipse = 2.4477
```

Area of an ellipse is product of semiaxis times π... our semiaxis are square root of covariance eigenvalues, so product is square root of determinant... times the zoom factor CfEllipse at each axis (so times "chi squared" bound).

```
Area=pi*sqrt(det(CovMatrix))*chich
```

```
Area = 37.6455
```

Confidence rectangle:

```
confrectequal=sqrt(confidence) %si fueran "iguales
```

```
confrectequal = 0.9747
```

```
confrect1=confrectequal;%minimal area rectangle, creo
%confrect1=0.974
confrect2=confidence/confrect1
```

```
confrect2 = 0.9747
```

```
cfboundeach1=norminv(1-(1-confrect1)/2)
```

```
cfboundeach1 = 2.2365
```

```
cfboundeach2=norminv(1-(1-confrect2)/2)
```

```
cfboundeach2 = 2.2365
```

```
AreaRect=cfboundeach2*cfboundeach1*4*sigma1*sigma2 %larger than the
confidence ellipse
```

```
AreaRect = 40.0146
```

If we "project" the ellipsoid, the "rectangle" containing the ellipsoid will have probability:

```
unused=normcdf(1.96)-normcdf(-1.96) %a reminder on what we did with single-
variable standard normal
```

```
unused = 0.9500
```

```
EachProjectionConf=normcdf(cfEllipse)-normcdf(-cfEllipse)
```

```
EachProjectionConf = 0.9856
```

```
(EachProjectionConf)^2
```

```
ans = 0.9715
```

Let us plot everything:

```
figure()
patch([-sigma1*cfEllipse   sigma1*cfEllipse sigma1*cfEllipse
-sigma1*cfEllipse],[-sigma2*cfEllipse -sigma2*cfEllipse sigma2*cfEllipse
sigma2*cfEllipse],[0.2 0.85 0.3])
```

```
hold on
patch([-sigma1*cfboundeach1   sigma1*cfboundeach1 sigma1*cfboundeach1
-sigma1*cfboundeach1],[-sigma2*cfboundeach2 -sigma2*cfboundeach2
sigma2*cfboundeach2 sigma2*cfboundeach2],[1 .7 .65])
patch([-sigma1*cfSingle   sigma1*cfSingle sigma1*cfSingle -sigma1*cfSingle],
[-sigma2*cfSingle -sigma2*cfSingle sigma2*cfSingle sigma2*cfSingle],[0.95
.87 .85])
fimplicit(x'*inv(CovMatrix)*x==cfEllipse^2,LineWidth=4), grid on
hold off
 xline(cfSingle*sigma1,'m',LineWidth=3),   xline(-
cfSingle*sigma1,'m',LineWidth=3)
 yline(cfSingle*sigma2,Color=[.5 .8 .6],LineWidth=3),   yline(-
cfSingle*sigma2,Color=[.3 .9 .8],LineWidth=3)

legend("97.15%Rect","95%Rect","90.25%Rect","95%Ellip","95%marg1","","95%marg2
", Location="best")
  axis equal
  m=axis; axis(1.15*m)
  title("Confidence regions")
```