1.14 Multivariate normal density function

- Let X be a vector R.V.

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

mean

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}$$

- Covariance (matrix) :

$$X' = X^T \ : \ \text{transpose of } X$$

when m = 0

$$\begin{aligned} E(XX') &= E\left(\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \begin{bmatrix} X_1 & X_2 & \cdots & X_n \end{bmatrix} \right) \\ &= \begin{bmatrix} E(X_1X_1) & E(X_1X_2) & \cdots & E(X_1X_n) \\ E(X_2X_1) & E(X_2X_2) & \cdots & E(X_2X_n) \\ \vdots & \vdots & \ddots & \vdots \\ E(X_nX_1) & E(X_nX_2) & \cdots & E(X_nX_n) \end{bmatrix} \end{aligned}$$

covariance (general case)

- covariance (when m is not zero vector)

$$\mathbf{C} = \begin{bmatrix} E[(X_1 - m_1)^2] & E[(X_1 - m_1)(X_2 - m_2)] & \cdots \\ E[(X_2 - m_2)(X_1 - m_1)] & \ddots & \\ \vdots & & E[(X_n - m_n)^2] \end{bmatrix}$$

- Exercise: Let X_1 and X_2 be independent R.V.s with $X_1 \sim N(0,1)$ and $X_2 \sim N(1,4)$.
  - find the covariance of

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Exercise: covariance

- Let X_1 and X_2 be independent R.V.s with $X_1 \sim N(0,1)$ and $X_2 \sim N(1,4)$.
- covariance matrix $\begin{bmatrix} E(X_1^2) & E(X_1(X_2-1)) \\ E(X_1(X_2-1)) & E((X_2-1)^2) \end{bmatrix}$
- X, Y : independent

$$E(X_1X_2) = E(X_1)E(X_2) \Rightarrow E(X_1(X_2-1)) = E(X_1X_2-X_1) = E(X_1)E(X_2)-E(X_1) = 0$$

- covariance

$$\begin{bmatrix} E(X_1^2) & E(X_1(X_2-1)) \\ E(X_1(X_2-1)) & E((X_2-1)^2) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

jointly normal or jointly Gaussian R.V.s

- R.V.s $X_1, X_2, \ldots, X_n$ are said to be *jointly normal* or *jointly Gaussian* if their joint p.d.f. is given by

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left\{-\frac{1}{2}\left[(\mathbf{x}-\mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})\right]\right\} \qquad (1.14.5)$$

  - m is mean, C is covariance

- Example:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \qquad (1.14.6)$$

and

$$\mathbf{C} = \begin{bmatrix} E[(X_1-m_1)^2] & E[(X_1-m_1)(X_2-m_2)] \\ E[(X_1-m_1)(X_2-m_2)] & E[(X_2-m_2)^2] \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

$$f_{x_1x_2}(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(x_1-m_1)^2}{\sigma_1^2} - \frac{2\rho(x_1-m_1)(x_2-m_2)}{\sigma_1\sigma_2} + \frac{(x_2-m_2)^2}{\sigma_2^2}\right]\right\}$$

$$\text{Correlation coefficient} = \rho = \frac{\text{Cov of } X \text{ and } Y}{\sqrt{\text{Var } X}\sqrt{\text{Var } Y}} = \frac{E[(X-m_X)(Y-m_Y)]}{\sqrt{\text{Var } X}\sqrt{\text{Var } Y}}$$

1.15 Linear transformation of normal R.V.s

- normal R.V.: $X \sim N(m_X, C_X)\ (X \in R^n, m_X \in R^n, C_X \in R^{n\times n})$

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_X|^{1/2}} \exp\left\{-\frac{1}{2}\left[(\mathbf{x}-\mathbf{m}_X)^T \mathbf{C}_X^{-1}(\mathbf{x}-\mathbf{m}_X)\right]\right\}$$

- R.V. Y is defined by : $Y = Ax + b\ (A \in R^{n\times n}, b \in R^n)$
  - Question: is Y Gaussian? → Answer: Yes
  - What is mean and covariance of Y?

$$m_Y = Am_X + b, \quad C_Y = AC_XA'$$

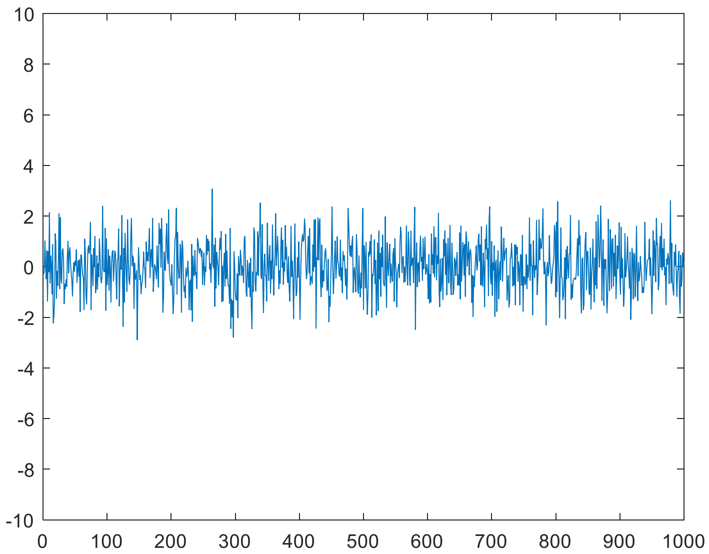
matlab example (1)

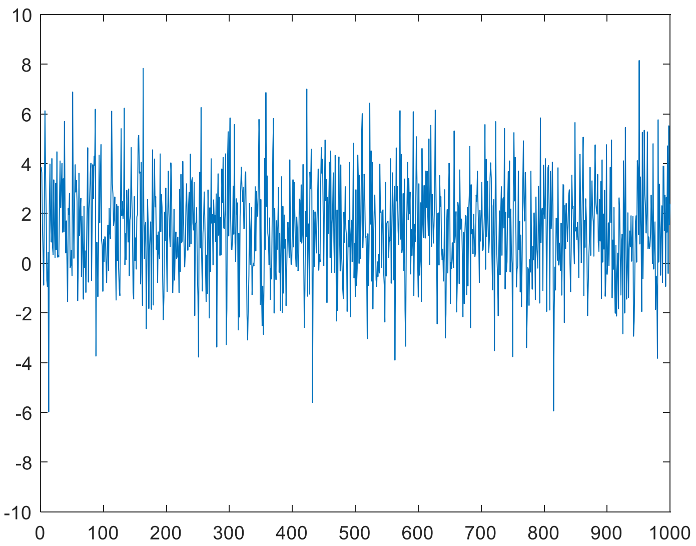
- X~N(0,1)

```
x1 = randn(1,1000);
plot(x1)
axis([0 1000 -10 10]);
```

X~N(1.5,4)

```
x1 = 2*randn(1,1000) + 1.5;
plot(x1)
axis([0 1000 -10 10]);
```



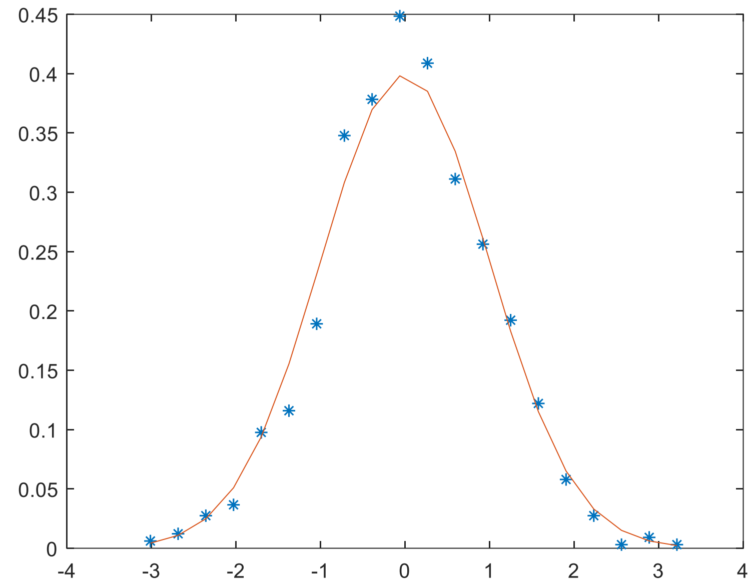


울산대학교
UNIVERSITY OF ULSAN

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$$

matlab example

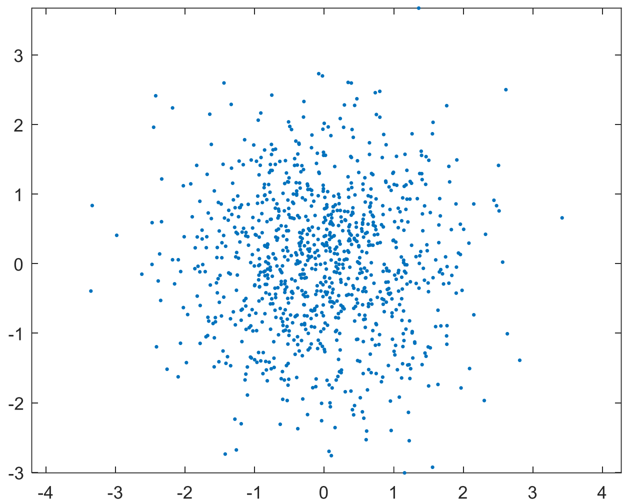
```
sigma = 1;
mu = 0;

Ntotal = 1000;
x1 = randn(1,Ntotal);

[counts,x] = hist(x1,20);
h = x(2) - x(1);
N = length(x);
plot(x,counts  /(h*Ntotal),'*');
```




```
y = zeros(1,N);
for i = 1:N
    y(i) = (1/ ( sqrt(2*pi) * sigma)) * exp( - (1/(2*sigma^2)) * ( x(i) - mu)^2);
end
hold on
plot(x,y);
hold off
```

2D Gaussian

```
Ntotal =1000;
x1 = randn(1,Ntotal);
x2 = randn(1,Ntotal);
plot(x1,x2,'.');
```



- What is the covariance matrix of $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$

2D Gaussian

- What is the covariance matrix of $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$

$$X_1 \sim N(0,1), \quad X_2 \sim N(0,2)$$

$$E(XX') = E(\begin{bmatrix} E(X_1^2) & E(X_1X_2) \\ E(X_1X_2) & E(X_2^2) \end{bmatrix}) = E(\begin{bmatrix} 1 & ? \\ ? & 1 \end{bmatrix}) = E(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$$

  - X1 and X2 are uncorrelated

Gaussian Probability Density Functions: Properties and Error Characterization

Similarly to what was considered for a Gaussian random variable, it is also useful for a variety of applications and for a second order Gaussian random vector, to evaluate the locus (x, y) for which the pdf is greater or equal a specified constant, K_1, i.e.,

$$\left\{(x,y) : \frac{1}{2\pi\sqrt{det\Sigma}} \exp\left[-\frac{1}{2}[x - m_X \; y - m_Y]\Sigma^{-1}[x - m_X \; y - m_Y]^T\right] \geq K_1\right\} \tag{2.15}$$

which is equivalent to

$$\left\{(x,y) : [x - m_X \; y - m_Y]\Sigma^{-1}\begin{bmatrix} x - m_X \\ y - m_Y \end{bmatrix} \leq K\right\} \tag{2.16}$$

with

$$K = -2\ln(2\pi K_1\sqrt{det\Sigma}).$$

Probability

$$n = 1; \quad Pr\{\text{x inside the ellipsoid}\} = -\frac{1}{\sqrt{2\pi}} + 2erf(\sqrt{K})$$
$$n = 2; \quad Pr\{\text{xinside the ellipsoid}\} = 1 - e^{-K/2}$$
$$n = 3; \quad Pr\{\text{x inside the ellipsoid}\} = -\frac{1}{\sqrt{2\pi}} + 2erf(\sqrt{K}) - \sqrt{\frac{2}{\pi}}\sqrt{K}e^{-K/2} \tag{4.4}$$

where n is the dimension of the random vector. Numeric values of the above expression for $n = 2$ are presented in the following table

Probability	K
50%	1.386
60%	1.832
70%	2.408
80%	3.219
90%	4.605

$$\{x : [x - m_X]^T \Sigma_X^{-1} [x - m_X] \leq K\}$$

$$q(x) = 0.5\mathbf{erf}(\frac{x}{\sqrt{2}})$$

- error function defined in the pdf is different from the usual definition (matlab)

$$q(x) = erf(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \exp(-\frac{y^2}{2})\, dy$$

$$\mathbf{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$

n	$\chi^2_{0.995}$	$\chi^2_{0.99}$	$\chi^2_{0.975}$	$\chi^2_{0.95}$	$\chi^2_{0.90}$	$\chi^2_{0.75}$	$\chi^2_{0.50}$	$\chi^2_{0.25}$	$\chi^2_{0.10}$	$\chi^2_{0.05}$
1	7.88	6.63	5.02	3.84	2.71	1.32	0.455	0.102	0.0158	0.0039
2	10.6	9.21	7.38	5.99	4.61	2.77	1.39	0.575	0.211	0.103
3	12.8	11.3	9.35	7.81	6.25	4.11	2.37	1.21	0.584	0.352
4	14.9	13.3	11.1	9.49	7.78	5.39	3.36	1.92	1.06	0.711

From this table we can conclude, for example, that for a third-order Gaussian random vector, $n = 3$,

$$Pr\{K \leq 6.25\} = Pr\{[x - m_X]^T \Sigma^{-1} [x - m_X] \leq 6.25\} = 0.9$$

covariance: diagonal case

Case 1 - Diagonal covariance matrix

When $\rho = 0$, i.e., the variables X and Y are uncorrelated, the covariance matrix is diagonal,

$$\Sigma = \begin{bmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_Y^2 \end{bmatrix} \tag{2.19}$$

and the eigenvalues particularize to $\lambda_1 = \sigma_X^2$ and $\lambda_2 = \sigma_Y^2$. In this particular case, illustrated in Figure 2.5, the locus (2.16) may be written as

$$\left\{(x, y) : \frac{(x - m_X)^2}{\sigma_X^2} + \frac{(y - m_Y)^2}{\sigma_Y^2} \leq K\right\} \tag{2.20}$$

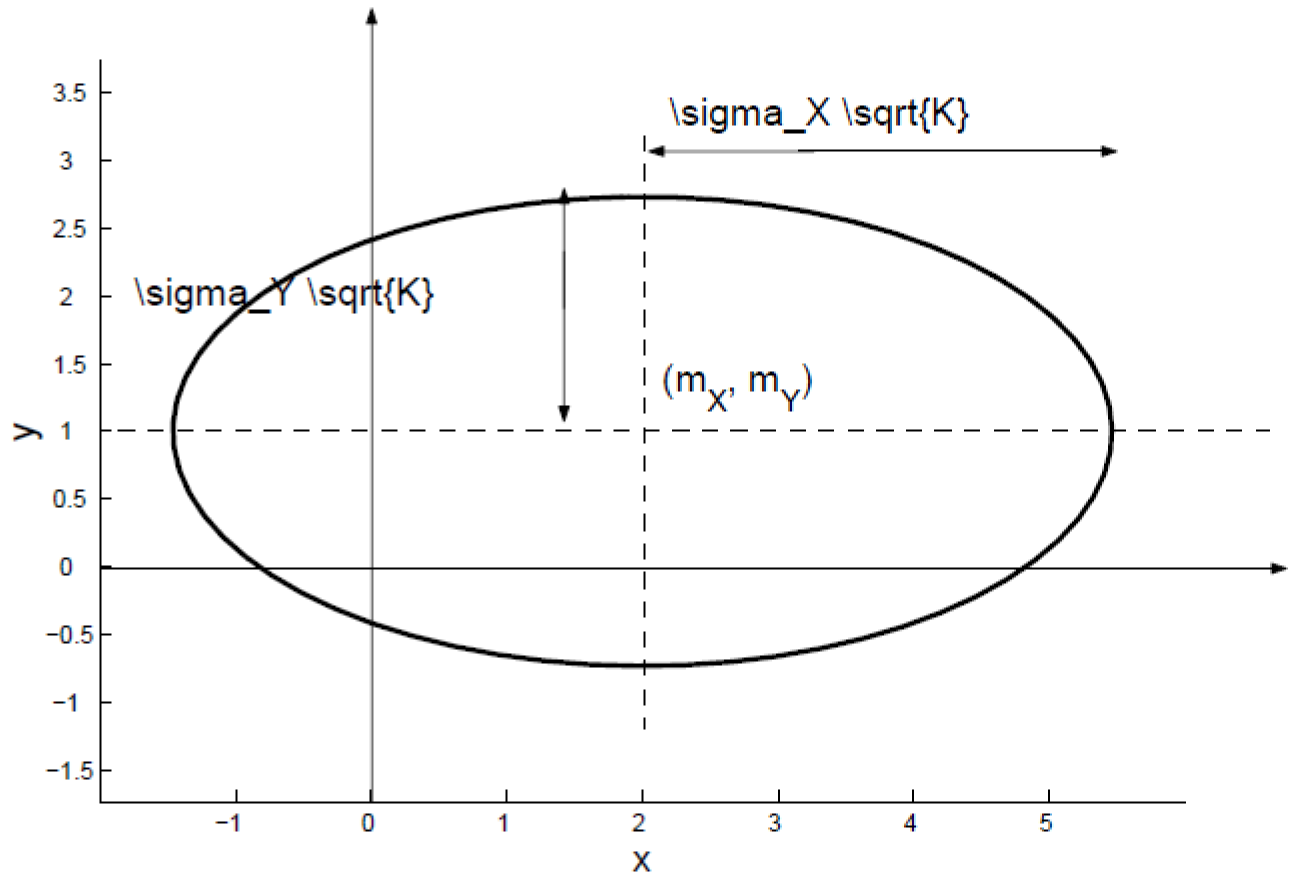
or also,

$$\left\{(x, y) : \frac{(x - m_X)^2}{K\sigma_X^2} + \frac{(y - m_Y)^2}{K\sigma_Y^2} \leq 1\right\}. \tag{2.21}$$

Figure 2.5 represents the ellipse that is the border of the locus in (2.21) having:

- x-axis with length $2\sigma_X\sqrt{K}$
- y-axis with length $2\sigma_Y\sqrt{K}$.

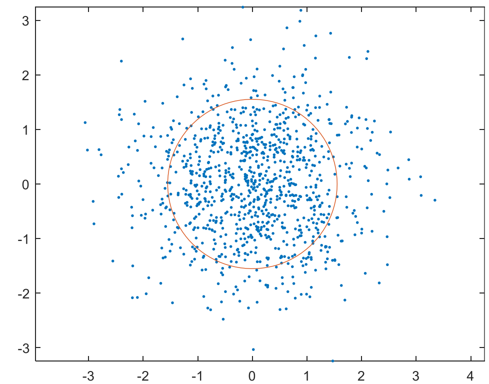
covariance: diagonal case



2D Gaussian

```
Ntotal =1000;
x1 = randn(1,Ntotal);
x2 = randn(1,Ntotal);
K = 2.408;

M = 100;
theta = 0:(2*pi/(M-1)):2*pi;
x = zeros(1,M);
y = zeros(1,M);
for i = 1:M
    x(i) = sqrt(K) * cos(theta(i));
    y(i) = sqrt(K) * sin(theta(i));
end
```

```
count = 0;
for i = 1:Ntotal
    if ( x1(i)^2 + x2(i)^2 < K )
        count = count + 1;
    end
end
fprintf('The number of points
inside the circle %d₩n',count)
```

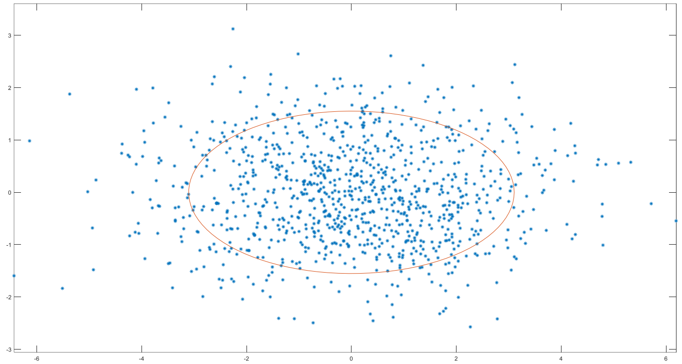


Exercise:

- Suppose X ~ N(m, C)

$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

  – draw the random point plot with 70% circle



Exercise: matlab code

```
Ntotal =1000;
x1 = 2 * randn(1,Ntotal);
x2 = randn(1,Ntotal);
K = 2.408;

M = 100;
theta = 0:(2*pi/(M-1)):2*pi;
x = zeros(1,M);
y = zeros(1,M);
for i = 1:M
    x(i) = ________________________;
    y(i) = ________________________;
end
```

```
count = 0;
for i = 1:Ntotal
    if ( (x1(i)/2)^2 + x2(i)^2 < K )
        count = count + 1;
    end
end
fprintf('The number of points
inside the circle %d₩n',count)
```


울산대학교
UNIVERSITY OF ULSAN

Exercise: matlab code

```
Ntotal =1000;
x1 = 2 * randn(1,Ntotal);
x2 = randn(1,Ntotal);
K = 2.408;

M = 100;
theta = 0:(2*pi/(M-1)):2*pi;
x = zeros(1,M);
y = zeros(1,M);
for i = 1:M
    x(i) = 2 * sqrt(K) * cos(theta(i));
    y(i) = sqrt(K) * sin(theta(i));
end
```

```
count = 0;
for i = 1:Ntotal
    if ( (x1(i)/2)^2 + x2(i)^2 < K )
        count = count + 1;
    end
end
fprintf('The number of points
inside the circle %d₩n',count)
```

울산대학교
UNIVERSITY OF ULSAN

covariance: non-diagonal case

- the ellipse has center in (m_X, m_Y) and its axes not aligned with the coordinate frame (let's assume the center is at the origin)

$$\Sigma = \begin{bmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{bmatrix}$$

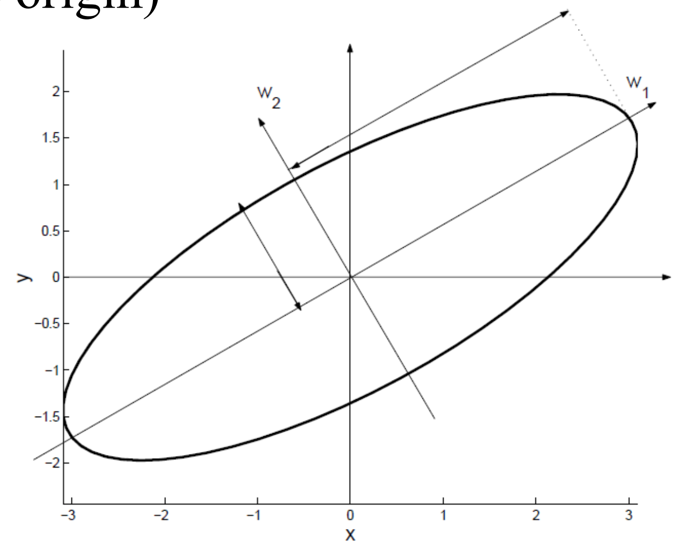
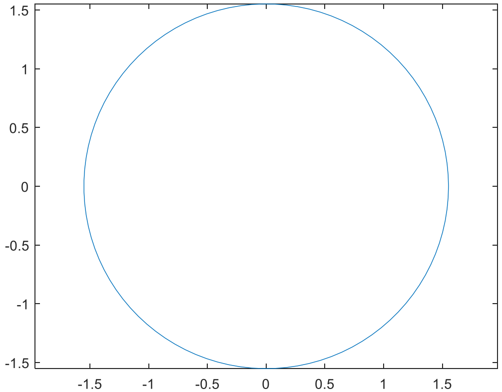


Figure 2.6: Ellipses non-aligned with the coordinate axis x and y.

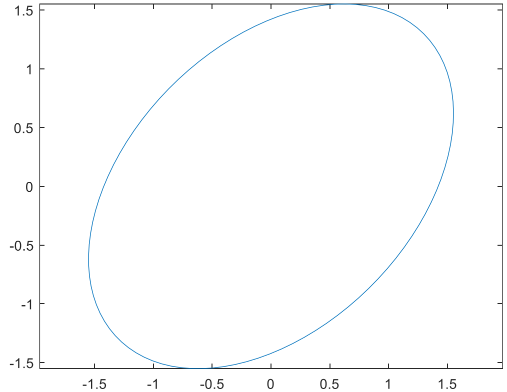
$$\begin{bmatrix} x & y \end{bmatrix} \Sigma^{-1} \begin{bmatrix} x \\ y \end{bmatrix} = K \quad \Rightarrow \quad \sigma_X^2 x^2 + \sigma_Y^2 y^2 + 2\rho\sigma_X\sigma_Y xy = K$$

Gaussian with non-diagonal covariance

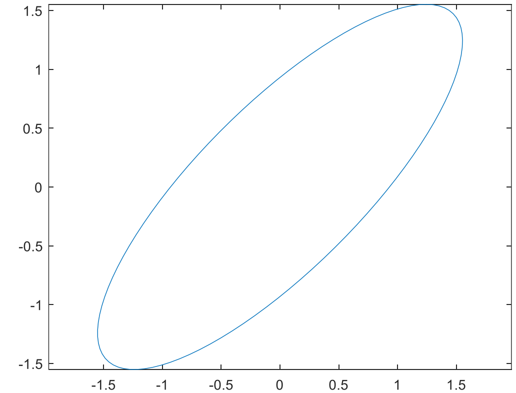
$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



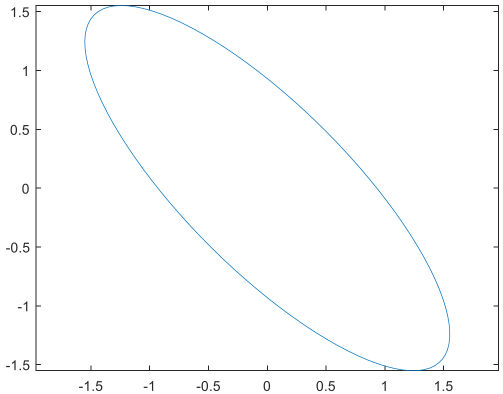
$$C = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$$



$$C = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$



$$C = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$



non diagonal case

- the length of the ellipse axis is related with the eigenvalues of the covariance matrix

$$\lambda_1 = \frac{1}{2}\left[\sigma_X^2 + \sigma_Y^2 + \sqrt{(\sigma_X^2 - \sigma_Y^2)^2 + 4\sigma_X^2\sigma_Y^2\rho^2}\right], \qquad (2.17)$$

$$\lambda_2 = \frac{1}{2}\left[\sigma_X^2 + \sigma_Y^2 - \sqrt{(\sigma_X^2 - \sigma_Y^2)^2 + 4\sigma_X^2\sigma_Y^2\rho^2}\right]. \qquad (2.18)$$

- decomposition of covariance matrix

$$\Sigma = T\ D\ T^{-1} \qquad (2.23)$$

where

$$T = [\ v_1 \quad | \quad v_2\ ], \quad D = diag(\lambda_1,\ \lambda_2)$$

and v_1, v_2 are the unit-norm eigenvectors of Σ associated with λ_1 and λ_2. Replacing (2.23) in (2.22) yields

$$\left\{(x, y) : [x \quad y] T D^{-1} T^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \leq K \right\}. \qquad (2.24)$$

non-diagonal case

Denoting

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = T^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.25}$$

and given that $T^T = T^{-1}$, it is immediate that (2.24) can be expressed as

$$\left\{ (w_1, w_2) : [w_1 \quad w_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}^{-1} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \leq K \right\} \tag{2.26}$$

- the coordinate transformation defined by (2.25) corresponds to a rotation of the coordinate system (x,y) around its origin by an angle

$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{2\rho\sigma_X\sigma_Y}{\sigma_X^2 - \sigma_Y^2} \right), \quad -\frac{\pi}{4} \leq \alpha \leq \frac{\pi}{4}, \quad \sigma_X \neq \sigma_Y .$$

ellipse equation

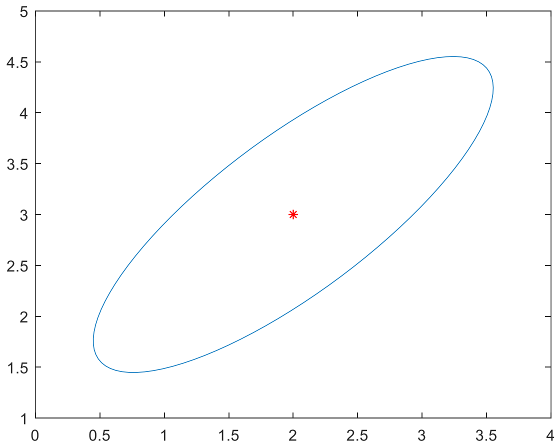
$$\left\{ (w_1, w_2) : \frac{w_1^2}{K\lambda_1} + \frac{w_2^2}{K\lambda_2} \leq 1 \right\} \tag{2.28}$$

that corresponds to an ellipse having

- w_1-axis with length $2\sqrt{K\lambda_1}$
- w_2-axis with length $2\sqrt{K\lambda_2}$

- the algorithm is implemented in “draw_ellipse.m”

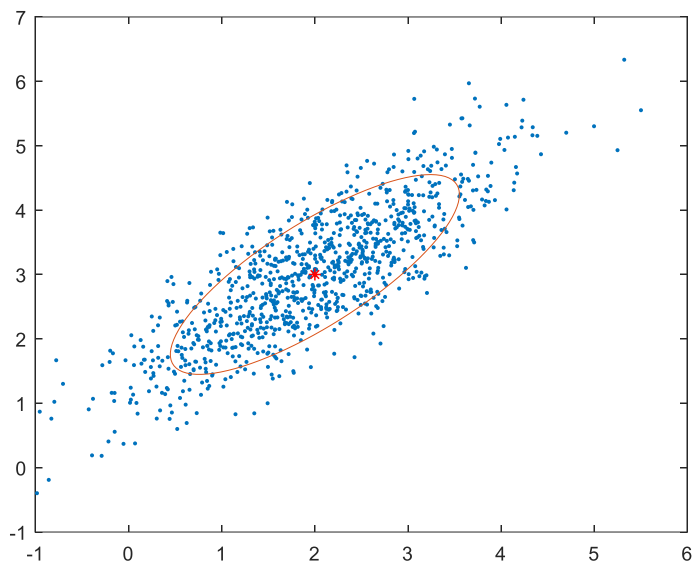
```
C = [ 1 0.8 ; 0.8 1];
m = [ 2 ; 3];
K = 2.408     % 70%

[r] = draw_ellipse(C,m,K);
plot(r(1,:),r(2,:),m(1),m(2),'r*');
```



simulation data generation

- matlab function : sqrtm

```
N = 10000;
a = sqrtm(C) * randn(2,N) + m;
plot(a(1,:),a(2,:),'.')
hold on
plot(r(1,:),r(2,:),m(1),m(2),'r*');
hold off
```



  - X = sqrtm(A) returns the principal square root of the matrix A, that is,
    - X*X = A.

Example: robot's position estimation (1)

- Suppose you want to estimate 2D position of a mobile robot with some sensor
  - there are two measurements

$$y_1 = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix}, \quad y_2 = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$$

  - let r be the true position of the robot.
  - What is your estimate of r based on two measurements?

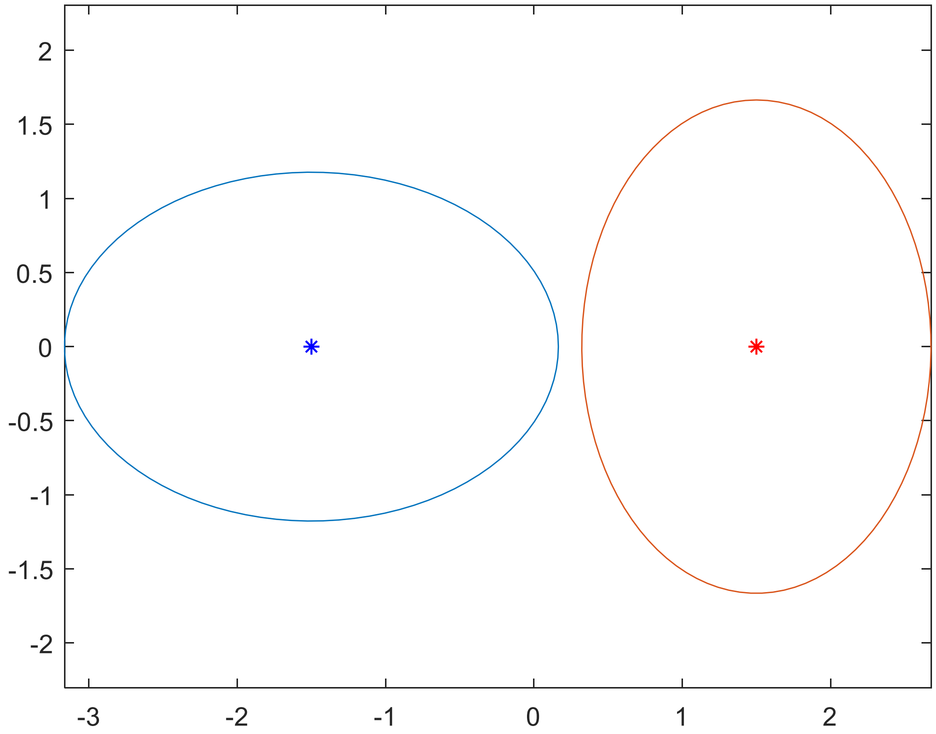
$$\hat{r} = \frac{y_1 + y_2}{2}$$

  - Note: measurement model (v1 and v2 are uncorrelated)

$$\begin{aligned} y_1 &= r + v_1 \\ y_2 &= r + v_2 \end{aligned} \qquad v_1 \sim N(0, C_1), \quad v_2 \sim N(0, C_2)$$

Example: robot's position estimation (2)

$$C_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$



50% ellipse

Optimal estimator?

- Suppose the estimator is the following form:

$$\hat{r} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} y_1 + \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} y_2$$

- estimator error $e = r - \hat{r} = -\begin{bmatrix} k_1 & 0 \\ 0 & k_1 \end{bmatrix} v_1 - \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} v_2$

- estimation error covariance

$$\begin{aligned} \mathrm{E}\{ee'\} &= \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \mathrm{E}\{v_1 v_1'\} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} + \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} \mathrm{E}\{v_1 v_1'\} \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} \\ &= \begin{bmatrix} 2k_1^2 & 0 \\ 0 & k_2^2 \end{bmatrix} + \begin{bmatrix} (1-k_1)^2 & 0 \\ 0 & 2(1-k_2)^2 \end{bmatrix} \\ &= \begin{bmatrix} 2k_1^2 + (1-k_1)^2 & 0 \\ 0 & k_2^2 + 2(1-k_2)^2 \end{bmatrix} \end{aligned}$$

- optimization problem

$$\min_{k_1, k_2} \mathrm{Tr}\, \mathrm{E}\{ee'\}$$

Optimal estimator

- minimization $\quad \mathrm{Tr}\,\mathrm{E}\{ee'\} = 2k_1^2 + (1-k_1)^2 + k_2^2 + 2(1-k_2)^2$

- k_1, k_2 minimization

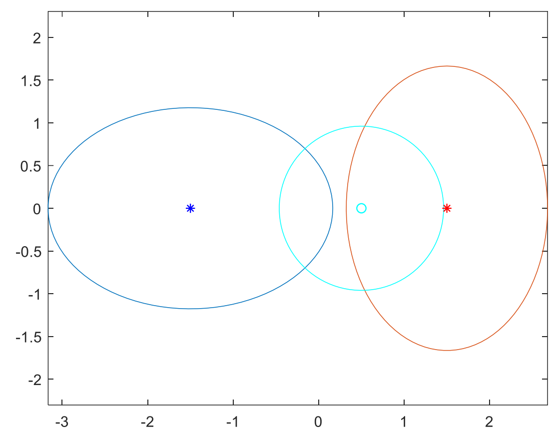
$$f_1(k_1) = 2k_1^2 + (1-k_1)^2 = 3k_1^2 - 2k_1 + 1 \;\Rightarrow\; f_1'(k_1) = 0 \;\Rightarrow k_1 = \frac{1}{3}$$

$$f_2(k_2) = k_2^2 + 2(1-k_2)^2 = 3k_2^2 - 4k_2 + 1 \;\Rightarrow\; f_2'(k_2) = 0 \;\Rightarrow k_2 = \frac{2}{3}$$

- optimal estimator

$$\hat{r} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{2}{3} \end{bmatrix} y_1 + \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & \frac{1}{3} \end{bmatrix} y_2 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$\mathrm{E}\{ee'\} = \begin{bmatrix} 2k_1^2 + (1-k_1)^2 & 0 \\ 0 & k_2^2 + 2(1-k_2)^2 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & \frac{2}{3} \end{bmatrix}$$



Another estimator

- Suppose the estimator is the following form:

$$\hat{r} = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} y_1 + \begin{bmatrix} 1-k & 0 \\ 0 & 1-k \end{bmatrix} y_2$$

- estimator error

$$e = r - \hat{r} = -\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} v_1 - \begin{bmatrix} 1-k & 0 \\ 0 & 1-k \end{bmatrix} v_2$$

- estimation error covariance

$$\begin{aligned} \mathrm{E}\{ee'\} &= \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \mathrm{E}\{v_1 v_1'\} \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} + \begin{bmatrix} 1-k & 0 \\ 0 & 1-k \end{bmatrix} \mathrm{E}\{v_1 v_1'\} \begin{bmatrix} 1-k & 0 \\ 0 & 1-k \end{bmatrix} \\ &= \begin{bmatrix} 2k^2 & 0 \\ 0 & k^2 \end{bmatrix} + \begin{bmatrix} (1-k)^2 & 0 \\ 0 & 2(1-k)^2 \end{bmatrix} \\ &= \begin{bmatrix} 2k^2 + (1-k)^2 & 0 \\ 0 & k^2 + 2(1-k)^2 \end{bmatrix} \end{aligned}$$

- optimization problem

$$\min_k \mathrm{Tr}\, \mathrm{E}\{ee'\} = 3(k^2 + (1-k)^2) = 3(2k^2 - 2k + 1) \Rightarrow k = \frac{1}{2}$$

Another estimator (2)

- estimator

$$\hat{r} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} y_1 + \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} y_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathrm{E}\{ee'\} = \begin{bmatrix} 2k^2 + (1-k)^2 & 0 \\ 0 & k^2 + 2(1-k)^2 \end{bmatrix} = \underline{\begin{bmatrix} \frac{3}{4} & 0 \\ 0 & \frac{3}{4} \end{bmatrix}}$$

larger error covariance

Exercise: robot's position estimation

- There are two measurements

$$y_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathrm{E}\{v_1\} = 0, \mathrm{E}\{v_1 v_1'\} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$y_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \mathrm{E}\{v_2\} = 0, \mathrm{E}\{v_2 v_2'\} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

- Suppose the estimator has the following form

$$\hat{r} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} y_1 + \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} y_2$$

- Find the optimal estimator and its error covariance
  - draw 50% error covariance radius

Exercise: robot's position estimation

- estimation error covariance

$$
\begin{aligned}
\mathrm{E}\{ee'\} &= \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \mathrm{E}\{v_1 v_1'\} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} + \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} \mathrm{E}\{v_1 v_1'\} \begin{bmatrix} 1-k_1 & 0 \\ 0 & 1-k_2 \end{bmatrix} \\
&= \begin{bmatrix} 2k_1^2 & 0 \\ 0 & k_2^2 \end{bmatrix} + \begin{bmatrix} 2(1-k_1)^2 & 0 \\ 0 & (1-k_2)^2 \end{bmatrix} \\
&= \begin{bmatrix} 2(k_1^2 + (1-k_1)^2) & 0 \\ 0 & k_2^2 + (1-k_2)^2 \end{bmatrix}
\end{aligned}
$$

- minimizing parameters $k_1 = \frac{1}{2}, \quad k_2 = \frac{1}{2}$

- estimator and its covariance

$$
\hat{r} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} y_1 + \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} y_2 = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}
$$

$$
\mathrm{E}\{ee'\} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}
$$