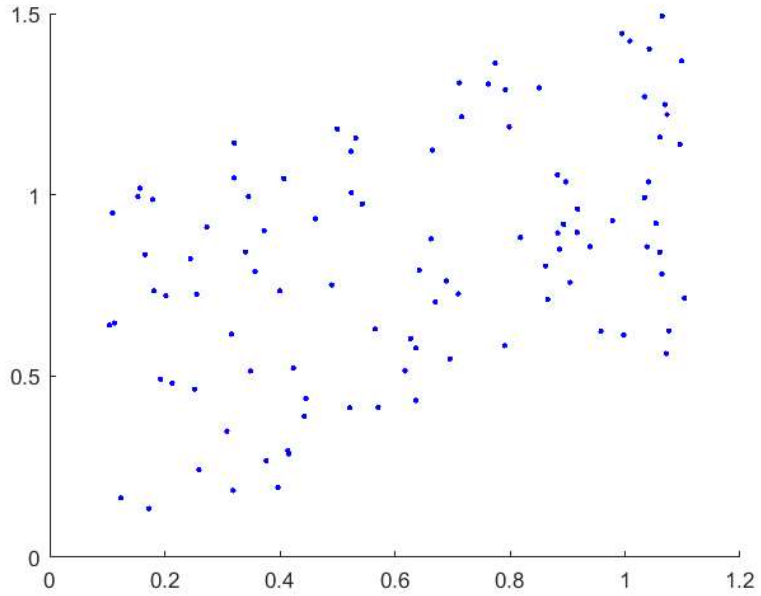
Pablo Vega Gallego, VC, FIB.

Sessió 1

En aquesta sessió realitzarem l'exercici proposat i mirarem de posar un exemple de per a què podria servir aquest codi.

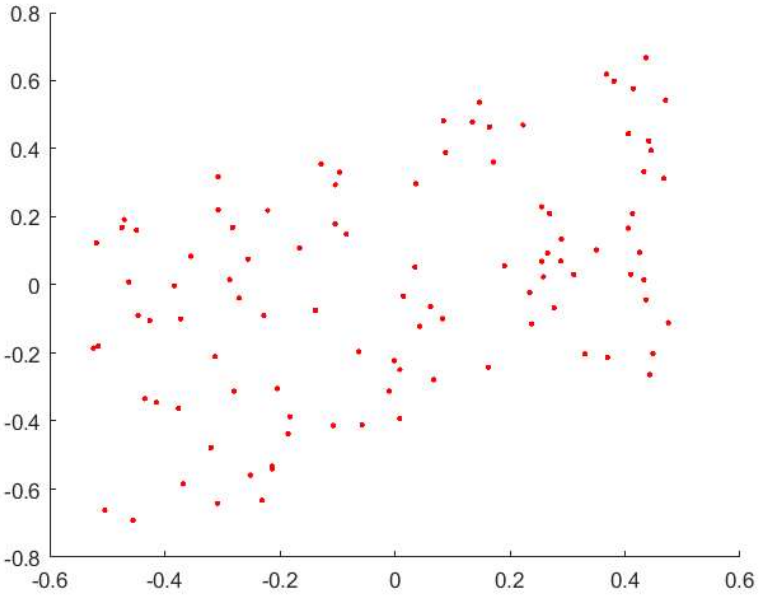
Comencem per la creació dels punts, en aquest cas hem decidit fer servir un núvol de punts totalment aleatori, fent servir la funció 'rand' de matlab.

```
x = rand(1, 100) + rand();
y = rand() .* x + rand(1, 100);
scatter(x, y, '.', 'b');
```



Després, centrarem aquests punts a l'origen de coordenades.

```
nx = x - mean(x);
ny = y - mean(y);
figure
scatter(nx, ny, '.', 'r');
```



Realitzarem el càlcul de la matriu de covariança per més tard trobar els eigenvector (vectors propis) i els eigenvalues (valors propis), per més tard poder trobar els eixos del nostre model. Aprofitarem que la matriu de covariança és simètrica i ens

asegurarem que podem trobar uns valors.

```
c = cov(nx, ny);
[evectors, evalues] = eig(c);
```

Valors de les variables...

```
disp(c);
```

```
    0.0998    0.0471
    0.0471    0.1056
```

```
disp(evectors);
```

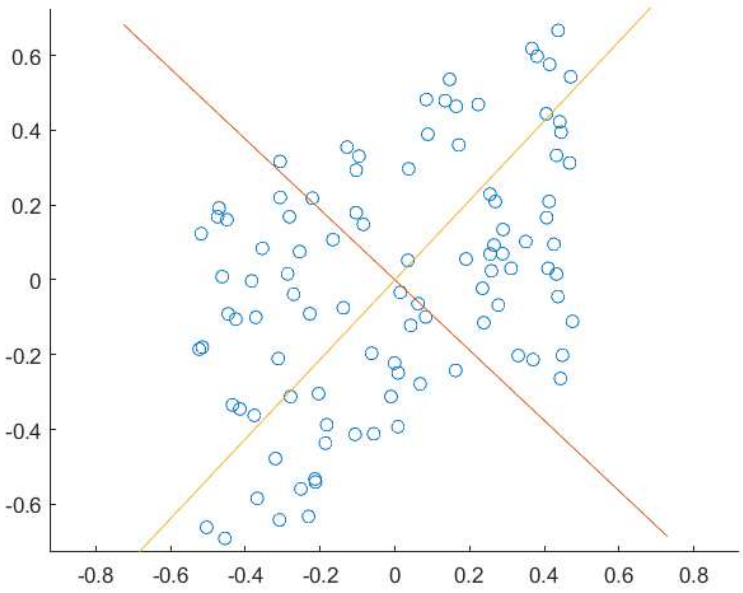
```
   -0.7284    0.6851
    0.6851    0.7284
```

```
disp(evalues);
```

```
    0.0555         0
         0    0.1499
```

Farem una impressió dels eixos i dels punts.

```
figure
hold on
escala = 1;
%axis equal
scatter(nx, ny);
axis equal
plot([evectors(1, 1)*-escala, evectors(1, 1)*escala], [evectors(1, 2)*-escala, evectors(1, 2)*escala]);
plot([evectors(2, 1)*-escala, evectors(2, 1)*escala], [evectors(2, 2)*-escala, evectors(2, 2)*escala]);
hold off
```



Ara, calcularem l'angle de rotació per tal de tenir els punts alineats amb els eixos i girarem fent servir la matriu de rotació.

```
[value, ind] = max(diag(evalues));
theta = -pi/2 - atan2(evectors(ind, 2), evectors(ind, 1));
disp(theta);
```

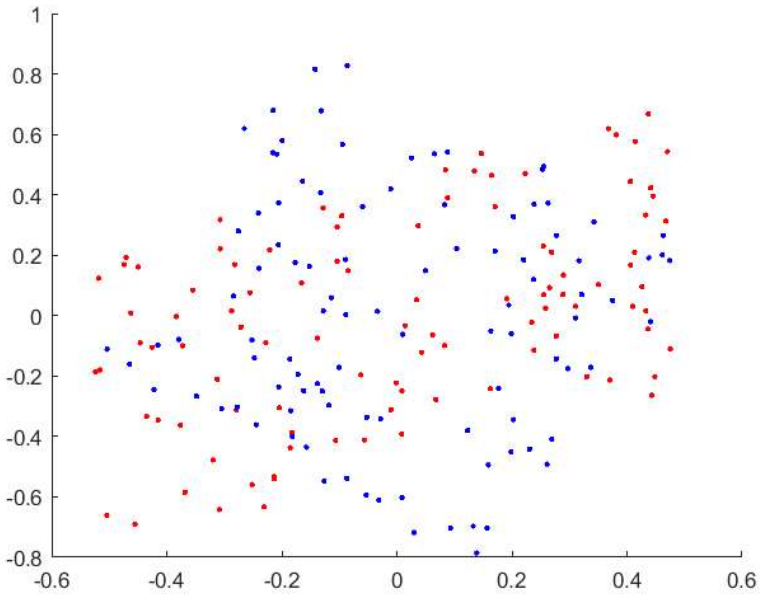
```
   -2.3868
```

```
R = [cos(-theta), sin(-theta); -sin(-theta), cos(-theta)];
rp = R * [nx; ny]
```

```
rp = 2×100
    0.0090   -0.1385   -0.0596   -0.1279    0.2774    0.4626    0.1988   -0.4155   -0.1320   -0.2091   -0.3484   -0.5037
   -0.6016   -0.2251    0.3617    0.0158    0.2646    0.2653   -0.0599   -0.0981    0.6782    0.5345   -0.2666   -0.1117
```

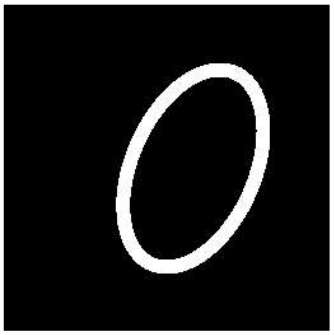
```
figure
```

```
hold on
scatter(nx, ny, '.', 'r');
scatter(rp(1, :), rp(2, :), '.', 'b');
hold off
```



Ara per tal de veure la utilitat d'aquest sistema carregarem una imatge com aquesta:

```
I = imread('oval.png');
circulo = im2bw(I, 0);
imshow(circulo);
```



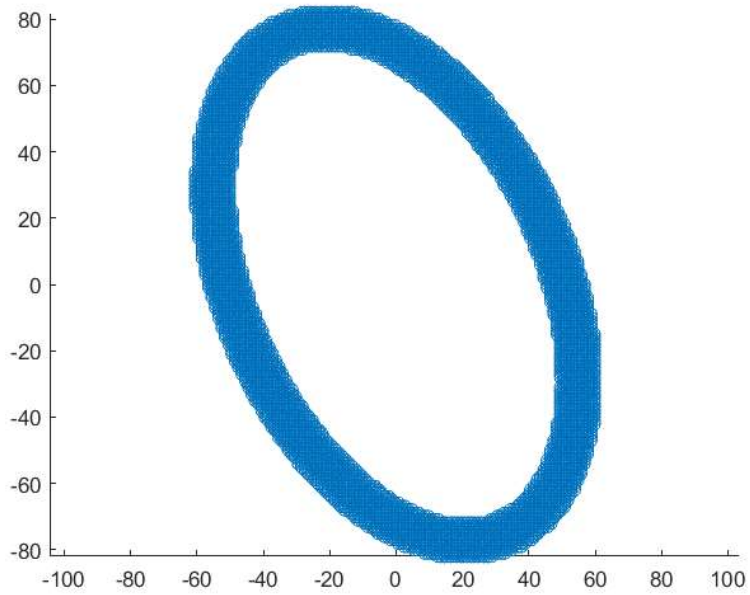
Ara tractarem la imatge com si fos un gràfic de punts com el vist anteriorment.

```
[y, x] = find(I);

nx = x - mean(x);
ny = y - mean(y);

nx = nx';
ny = ny';

figure
hold on
axis equal
scatter(nx, ny);
hold off
```



Aplicarem els mateixos mètodes que abans.

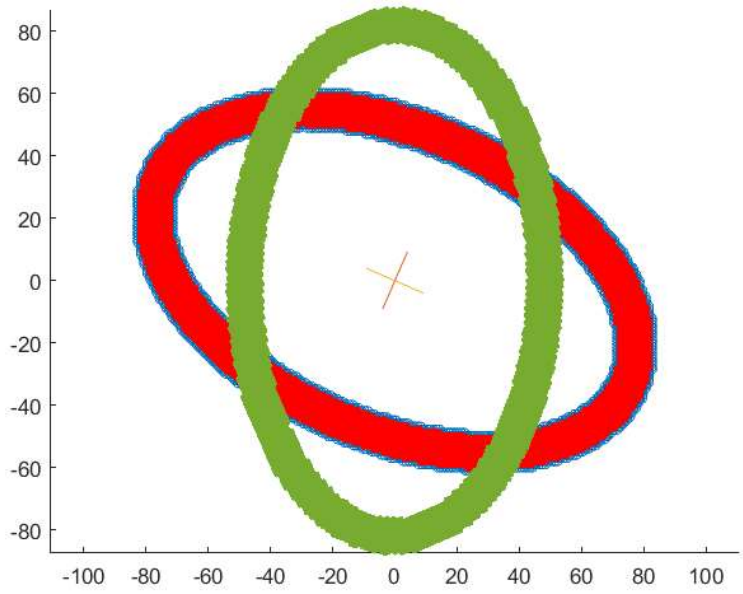
```
cc = cov(ncx, ncy);
[cevectors, cevalues] = eig(cc);

figure
hold on
cescala = 10;
axis equal
scatter(ncx, ncy);
axis equal
plot([cevectors(1, 1)*-cescala, cevectors(1, 1)*cescala], [cevectors(1, 2)*-cescala, cevectors(1, 2)*cescala]);
plot([cevectors(2, 1)*-cescala, cevectors(2, 1)*cescala], [cevectors(2, 2)*-cescala, cevectors(2, 2)*cescala]);
hold off
[cvalue, cind] = max(diag(cevalues));
ctheta = -pi/2 - atan2(cevectors(cind, 2), cevectors(cind, 1));

cR = [cos(-ctheta), sin(-ctheta); -sin(-ctheta), cos(-ctheta)];
crp = cR * [ncx; ncy]
```

```
crp = 2×4609
   46.2029   45.8048   45.4067   45.0086   44.6105   44.2123   43.8142   43.4161   43.0180   42.6199   42.2217   41.8236   4
   45.1370   46.0543   46.9716   47.8890   48.8063   49.7236   50.6410   51.5583   52.4756   53.3930   54.3103   55.2276   3
```

```
hold on
scatter(ncx, ncy, '.', 'r');
scatter(crp(1, :), crp(2, :), '.');
```


80
60
40
20
0
-20
-40
-60
-80
-100
-80
-60
-40
-20
0
20
40
60
80
100