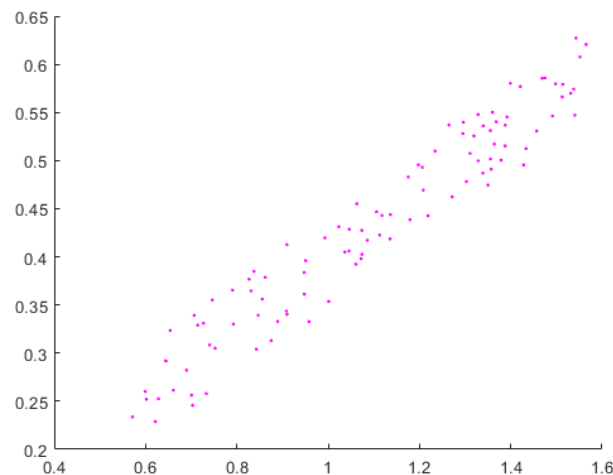


1. Creem un núvol de punts (x, y) aleatoris mitjançant la funció rand:

```
x = rand(1, 100) + rand();
y = rand().*x + rand(1,100)/10;
```

2. Per representar-los gràficament utilitzem la funció scatter, amb un marcador de punts de color magenta.

```
scatter(x, y, '.', 'm');
```



3. Per girar el núvol de punts (x, y) en relació al seu centre de masses amb un angle igual a la seva pendent. Primer de tot centrem el núvol de punts en el zero, restant la seva mitjana.

```
xp = x - mean(x);
yp = y - mean(y);
```

4. Calculem l'angle que té el núvol de punts amb l'eix horitzontal.

Primer calculem la covariància i obtenim els vectors i valors propis:

```
c = cov(xp, yp);
[evector, evalues] = eig(c);
```

Determinem quina dimensió té una major variància:

La variable val indica el valor màxim de la matriu diagonal de vectors propis.

La variable ind indica la posició dintre de la matriu 1x2 resultant d'aplicar la diagonal a la matriu de vector propis evalues.

```
[val, ind] = max(diag(evalues));
```

Extraiem l'angle amb el major eix:

Els valors substituïts per les XXXX de l'enunciat han sigut les posicions de columna que indica la variable ind, de les files 2 i 1 de la matriu dels vectors propis evector.

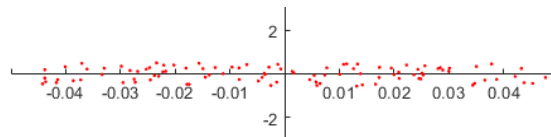
```
theta = -pi/2 - atan2(evector(2, ind), evector(1, ind));
```

5. Utilitzem la matriu de rotació R per girar els punts

```
R = [cos(theta), -sin(theta);
      sin(theta),  cos(theta)];
rp = R * [xp; yp];
```

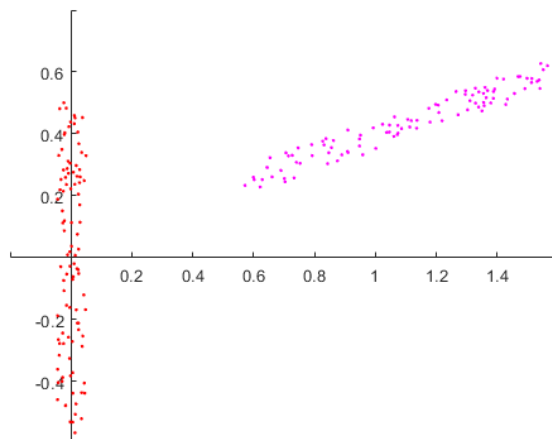
6. Fem servir la funció scatter, per representar el núvol de punts girats. Fem servir el marcador de punts de color vermell, establim límits a l'eix y perquè no es re-escalini les dades i ajustem l'eix d'abscisses i coordenades a l'origen.

```
scatter(rp(1, :), rp(2, :), '.', 'r');
ylim([-10, 10]);
set(gca, 'XAxisLocation', 'origin', 'YAxisLocation', 'origin')
```



Finalment juntem els dos núvols de punts en un sol gràfic per veure el resultat final

```
scatter(x, y, '.', 'm');
hold on
scatter(rp(1, :), rp(2, :), '.', 'r');
set(gca, 'XAxisLocation', 'origin', 'YAxisLocation', 'origin')
hold off
```



Es podrien representar de forma similar núvols de punts en 3D. Dependent de l'eix que volem rotar, la matriu R hauria de ser:

```
Rx = [ 1      0      0;
        0  cos(theta) -sin(theta);
        0  sin(theta)  cos(theta)];

Ry = [ cos(theta)  0  sin(theta);
        0          1  0;
       -sin(theta)  0  cos(theta)];

Rz = [ cos(theta) -sin(theta)  0;
        sin(theta)  cos(theta)  0;
         0          0          1];
```