PCA Exercise

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Data Representation

1,

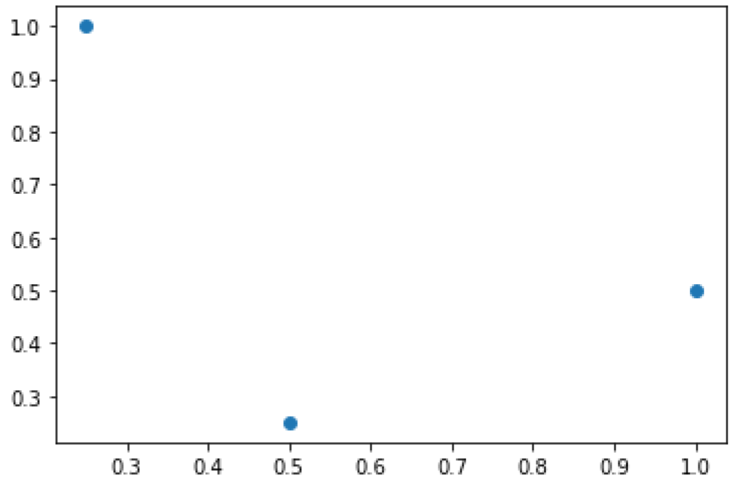
1.1

```
data = np.array([[0.25, 0.5, 1],[1, 0.25, 0.5]])
df = pd.DataFrame(data.transpose())
df
```

	0	1
0	0.25	1.00
1	0.50	0.25
2	1.00	0.50

```
plt.scatter(data[0],data[1])
```

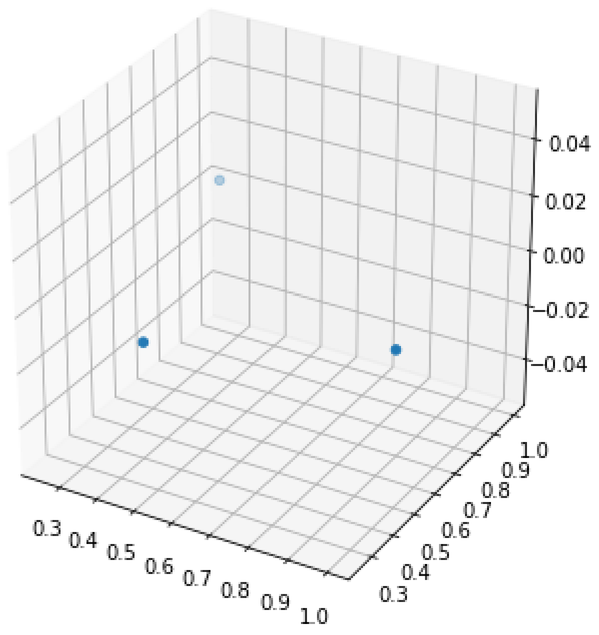
```
<matplotlib.collections.PathCollection at 0x7f47fb6ebb50>
```



1.2,

```
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(data[0], data[1])
```

```
    <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7f47fb70e850>
```



The figure was supposed to show vector from the origin (0,0,0) to the two points given, but I found that matplotlib is very bad at ploting vectors in 3D

1.3

```
w = np.array(([1/3, 1/2, 1/6]))
D = np.diag(w)
Ybar = data @ D @ np.ones(3)
Ybar
```

```
    array([0.5       , 0.54166667])
```

1.4,

```
X = data - np.outer(Ybar,np.ones(3))
covma = X @ D @ X.transpose()
covma
```

```
    array([[ 0.0625    , -0.04166667],
           [-0.04166667,  0.11284722]])
```

1.5,

```
M = np.diag(np.diagonal(covma)**-0.5)
M @ A @ M
```

```
array([[ 1.        , -0.49613894],
       [-0.49613894,  1.        ]])
```

2.3, Show Pearson coeff between 2 variables is in fact cosine difference between 2 variables divided by a scalar

```
def ac(x, w):
  D = np.diag(w)
  xc = x @ D @ np.ones(3)
  return xc
```

Pearson

```
import scipy.stats as scs
scs.pearsonr(df[0],df[1])[0]
```

```
-0.4999999999999999
```

Cosine diff

```
from numpy import linalg as l
np.dot(ac(df[0],w), ac(df[1],w))/(l.norm(ac(df[0],w))*l.norm(ac(df[1],w)))
```

```
1.0
```