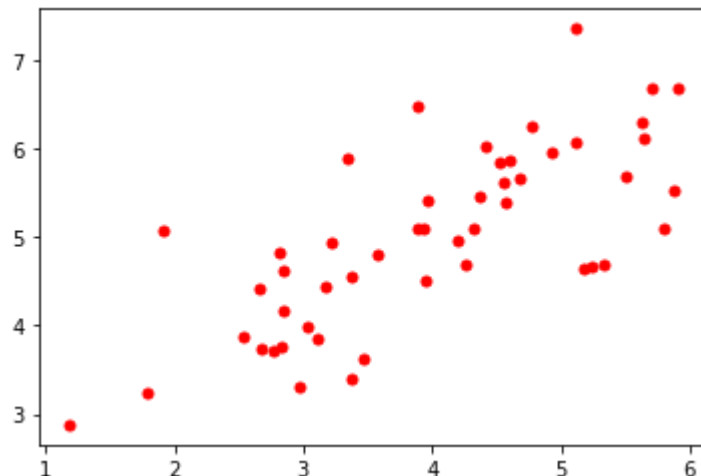


## 1 数据展示

PYTHON

```
from scipy.io import loadmat
import matplotlib.pyplot as plt
import numpy as np
from sklearn.decomposition import PCA
import pandas as pd
data = loadmat('/root/pycharmDemo/ML/data/MLData/7/ex7data1.mat')
datapd = pd.DataFrame(data['X'], dtype=float)
plt.scatter(datapd[0], datapd[1], s=25, c='r')
```



## 2 PCA降维

PYTHON

```
data = PCA(n_components=1).fit_transform(datapd[[0,1]])
data.shape
```

(5000, 1)

## 3 人脸数据

PYTHON

```
data = loadmat('/root/pycharmDemo/ML/data/MLData/7/ex7faces.mat')
datapd = pd.DataFrame(data['X'])
```

```

X = np.array(datapd)
X = np.array([x.reshape((32, 32)).T.reshape(1024) for x in data.get('X')])
def plot_n_image(X, n):
    """ plot first n images
    n has to be a square number
    """
    pic_size = int(np.sqrt(X.shape[1]))
    grid_size = int(np.sqrt(n))

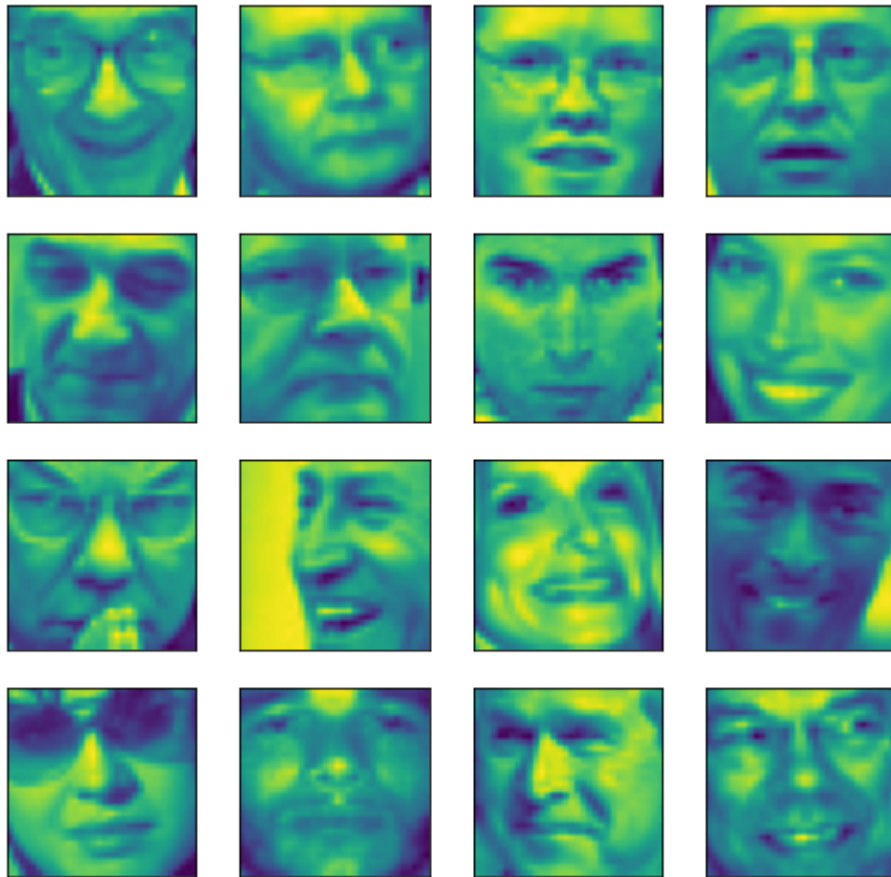
    first_n_images = X[:n, :]

    fig, ax_array = plt.subplots(nrows=grid_size, ncols=grid_size,
                                  sharey=True, sharex=True, figsize=(
    10, 10))

    for r in range(grid_size):
        for c in range(grid_size):
            ax_array[r, c].imshow(first_n_images[grid_size * r + c].re
            plt.xticks(np.array([]))
            plt.yticks(np.array([]))

plot_n_image(X,16)

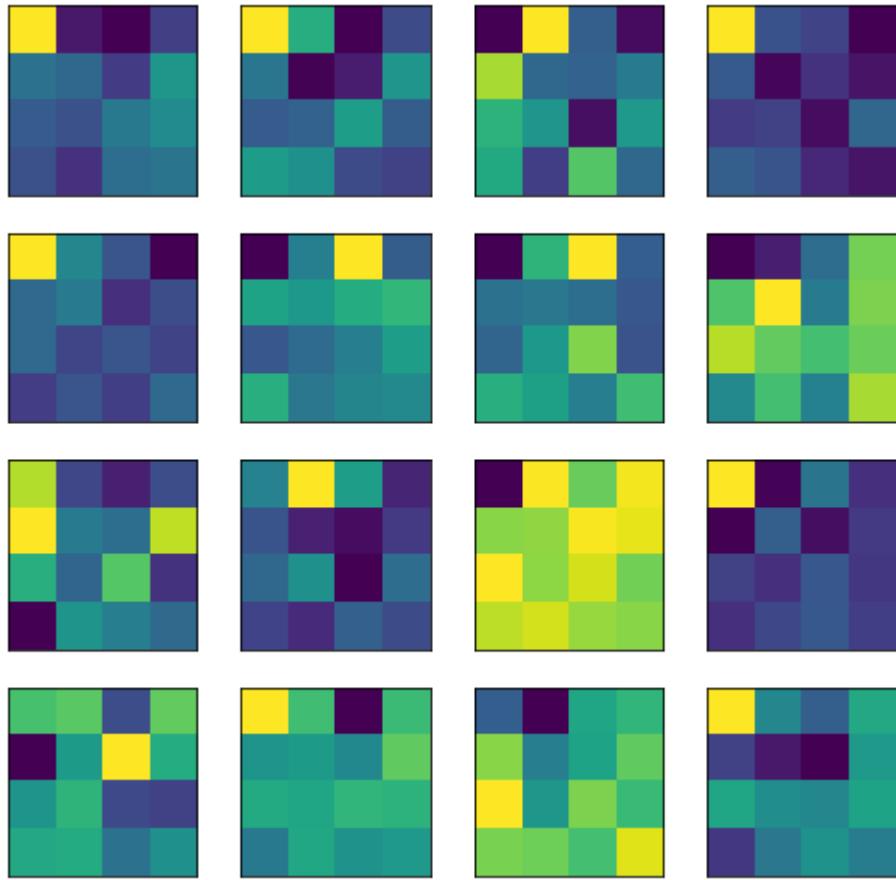
```



## 4 降维

PYTHON

```
model = PCA(n_components=16).fit(X)
Y = model.transform(X)
plot_n_image(Y,16)
```



PYTHON

# 特征向量

```
model.components_.shape
```

```
X2 = model.inverse_transform(Y)
```

```
plot_n_image(X,16)
```

(16, 1024)

