In [1]:

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
% matplotlob inline
```

```
UsageError: Line magic function `%` not found.
```

In [3]:

```python
from sklearn.datasets import load_breast_cancer
cancer=load_breast_cancer()
```

In [8]:

```python
print(cancer['DESCR'])
print(cancer['target_names'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
```

In [9]:

```python
df=pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
df.head()
```

Out[9]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

5 rows × 30 columns

In [17]:
```
from sklearn.preprocessing import StandardScaler
Scaler=StandardScaler()
Scaler.fit(df)
Scaled_data=Scaler.transform(df)
```

In [18]:
```
from sklearn.decomposition import PCA
pca=PCA(n_components=2)
pca.fit(Scaled_data)
```

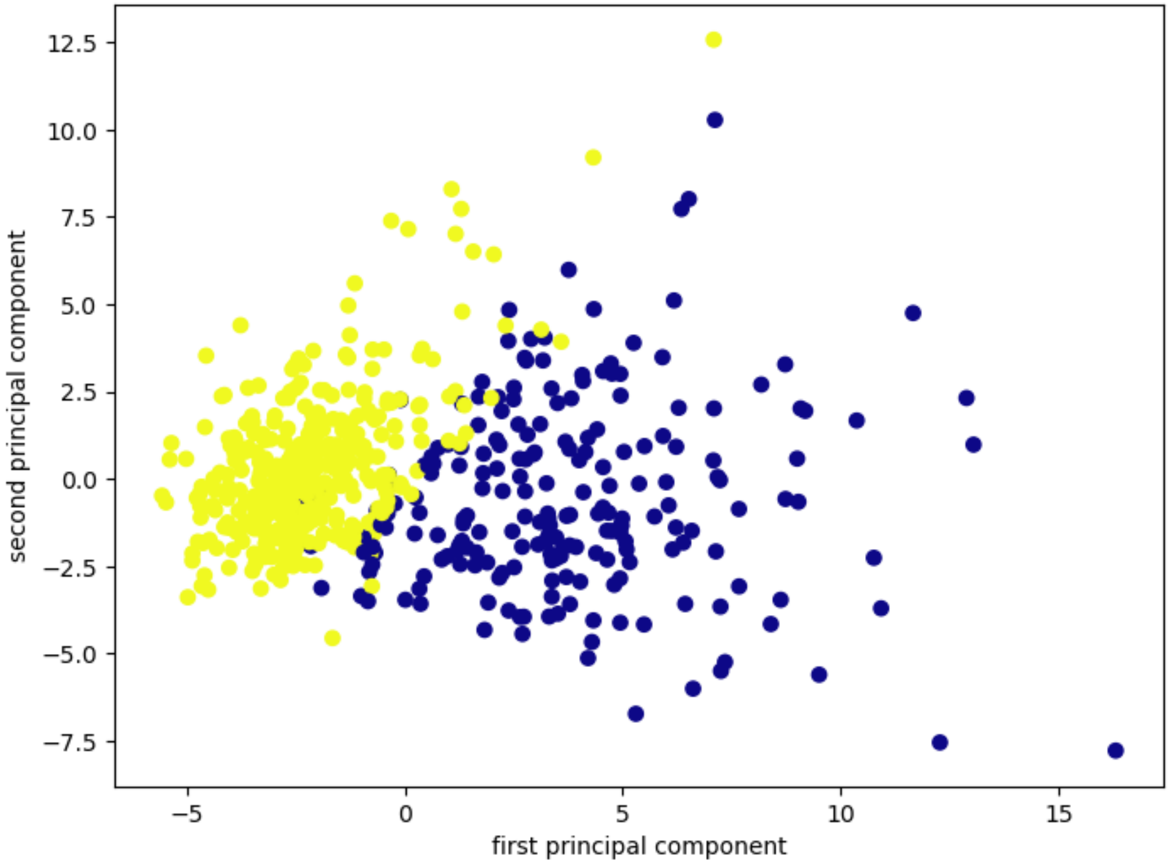
Out[18]:
```
PCA(n_components=2)
```

In [20]:
```
x_pca=pca.transform(Scaled_data)
print(Scaled_data.shape)
print(x_pca.shape)
```

```
(569, 30)
(569, 2)
```

In [21]:
```
plt.figure(figsize=(8,6))
plt.scatter(x_pca[:,0],x_pca[:,1],c=cancer['target'],cmap='plasma')
plt.xlabel('first principal component')
plt.ylabel('second principal component')
```

Out[21]:
```
Text(0, 0.5, 'second principal component')
```

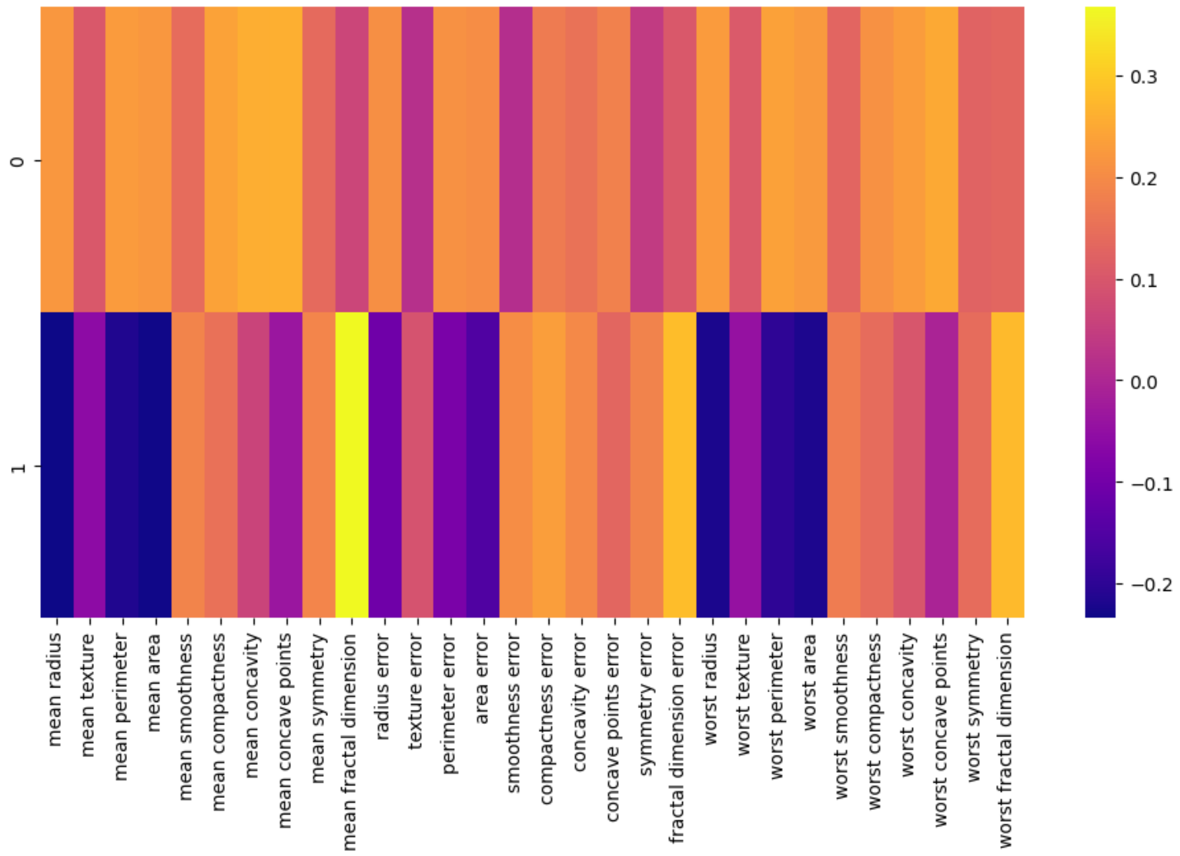


```
In [22]: pca.components_
```

```
Out[22]: array([[ 0.21890244,  0.10372458,  0.22753729,  0.22099499,  0.14258969,
                  0.23928535,  0.25840048,  0.26085376,  0.13816696,  0.06436335,
                  0.20597878,  0.01742803,  0.21132592,  0.20286964,  0.01453145,
                  0.17039345,  0.15358979,  0.1834174 ,  0.04249842,  0.10256832,
                  0.22799663,  0.10446933,  0.23663968,  0.22487053,  0.12795256,
                  0.21009588,  0.22876753,  0.25088597,  0.12290456,  0.13178394],
                [-0.23385713, -0.05970609, -0.21518136, -0.23107671,  0.18611302,
                  0.15189161,  0.06016536, -0.0347675 ,  0.19034877,  0.36657547,
                 -0.10555215,  0.08997968, -0.08945723, -0.15229263,  0.20443045,
                  0.2327159 ,  0.19720728,  0.13032156,  0.183848  ,  0.28009203,
                 -0.21986638, -0.0454673 , -0.19987843, -0.21935186,  0.17230435,
                  0.14359317,  0.09796411, -0.00825724,  0.14188335,  0.27533947]])
```

```
In [26]: df_comp=pd.DataFrame(pca.components_,columns=cancer['feature_names'])
         plt.figure(figsize=(12,6))
         sns.heatmap(df_comp,cmap='plasma',)
```

```
Out[26]: <AxesSubplot:>
```


mean radius
mean texture
mean perimeter
mean area
mean smoothness
mean compactness
mean concavity
mean concave points
mean symmetry
mean fractal dimension
radius error
texture error
perimeter error
area error
smoothness error
compactness error
concavity error
concave points error
symmetry error
fractal dimension error
worst radius
worst texture
worst perimeter
worst area
worst smoothness
worst compactness
worst concavity
worst concave points
worst symmetry
worst fractal dimension