

```
In [1]: import pandas as pd
import numpy as np
df = pd.read_csv("cell_samples.csv")
df.head()
```

```
Out[1]:
```

	ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
0	1000025	5	1	1	1	2	1	3	1	1	2
1	1002945	5	4	4	5	7	10	3	2	1	2
2	1015425	3	1	1	1	2	2	3	1	1	2
3	1016277	6	8	8	1	3	4	3	7	1	2
4	1017023	4	1	1	3	2	1	3	1	1	2

```
In [17]: df.describe()
```

```
Out[17]:
```

	ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BlandChrom	NormNucl	Mit	Class
count	6.990000e+02	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000	699.000000
mean	1.071704e+06	4.417740	3.134478	3.207439	2.806867	3.216023	3.437768	2.866953	1.589413	2.689557
std	6.170957e+05	2.815741	3.051459	2.971913	2.855379	2.214300	2.438364	3.053634	1.715078	0.951273
min	6.163400e+04	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000
25%	8.706885e+05	2.000000	1.000000	1.000000	1.000000	2.000000	2.000000	1.000000	1.000000	2.000000
50%	1.171710e+06	4.000000	1.000000	1.000000	1.000000	2.000000	3.000000	1.000000	1.000000	2.000000
75%	1.238298e+06	6.000000	5.000000	5.000000	4.000000	4.000000	5.000000	4.000000	1.000000	4.000000
max	1.345435e+07	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	4.000000

```
In [4]: import matplotlib.pyplot as plt
plt.scatter(df["Class"]==2,df["Clump"],df["UnifSize"],label='malignant')
plt.scatter(df["Class"]==4,df["Clump"],df["UnifSize"],label='benign')
plt.legend()
plt.show()
```

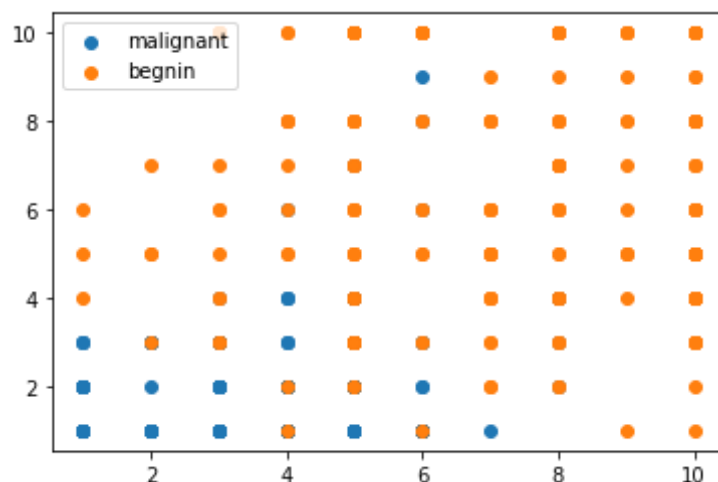
```
File "<ipython-input-4-7952512f0ae6>", line 3
plt.scatter(df["Class"]==4,df["Clump"],df["UnifSize"],label='benign')
```

SyntaxError: EOL while scanning string literal

```
In [10]: yy = df[df["Class"]==2]
         tt = df[df["Class"]==4]

         # plt.scatter(df["Clump"],df["UnifSize"],label='malignant')
```

```
In [11]: import matplotlib.pyplot as plt
         plt.scatter(yy["Clump"],yy["UnifSize"],label='malignant')
         plt.scatter(tt["Clump"],tt["UnifSize"],label='benign')
         plt.legend()
         plt.show()
```



```
In [37]: df = df[pd.to_numeric(df['BareNuc'], errors='coerce').notnull()]
         df['BareNuc'] = df['BareNuc'].astype('int')
         x = df.drop(["ID", "Class"],axis=1)
         x.head()
```

```
Out[37]:
```

	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit
0	5	1	1	1	2	1	3	1	1
1	5	4	4	5	7	10	3	2	1

	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit
2	3	1	1	1	2	2	3	1	1
3	6	8	8	1	3	4	3	7	1
4	4	1	1	3	2	1	3	1	1

```
In [38]: y = df["Class"]
y.head()
```

```
Out[38]: 0    2
1    2
2    2
3    2
4    2
Name: Class, dtype: int64
```

```
In [35]: df.isnull().sum()
```

```
Out[35]: ID          0
Clump            0
UnifSize         0
UnifShape        0
MargAdh          0
SingEpiSize      0
BareNuc          0
BlandChrom       0
NormNucl         0
Mit              0
Class            0
dtype: int64
```

```
In [39]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=1)
```

```
In [40]: x_test.shape
```

```
Out[40]: (137, 9)
```

```
In [41]: from sklearn.svm import SVC
model = SVC()
```

```
rissh = model.fit(x_train,y_train)
```

```
In [43]: y_pred =rissh.predict(x_test)
y_pred[0:5]
```

```
Out[43]: array([2, 2, 2, 4, 2], dtype=int64)
```

```
In [45]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
Out[45]: array([[90,  0],
               [ 0, 47]], dtype=int64)
```

```
In [48]: from sklearn.metrics import jaccard_score
jaccard_score(y_test,y_pred,pos_label=2)*100
```

```
Out[48]: 100.0
```

```
In [50]: from sklearn.metrics import f1_score
f1_score(y_test,y_pred,average="weighted")*100
```

```
Out[50]: 100.0
```

```
In [ ]:
```