In [1]:
```
1 from sklearn.datasets import load_iris
2 import pandas as pd
```

```
C:\Users\CompLab27\anaconda3\lib\site-packages\scipy\__init__.py:138: UserWarnin
g: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (d
etected version 1.24.4)
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion} is requ
ired for this version of "
```

In [3]:
```
1 dataset=load_iris() #constructor
2 cols=dataset.feature_names
3 df=pd.DataFrame(dataset.data,columns=cols)
4 df
```

Out[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [4]:
```
1 df.head()
```

Out[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [6]:
```
df["target"]=dataset.target  #Load new column
df["target"]
```

Out[6]:
```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: target, Length: 150, dtype: int32
```

In [7]:
```
df.head()
```

Out[7]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [8]:
```
df.tail()
```

Out[8]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

In [9]:
```
from sklearn.preprocessing import StandardScaler #Full data on same scale
```

In [10]:
```
1 x=df.iloc[:,:-1]
2 x
```

Out[10]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [20]:
```
1 y=df.iloc[:,-1]
2 y
```

Out[20]:
```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: target, Length: 150, dtype: int32
```

In [21]:
```
x=StandardScaler().fit_transform(x)
x
```

Out[21]:
```
array([[-9.00681170e-01,  1.01900435e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00, -1.31979479e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.38535265e+00,  3.28414053e-01, -1.39706395e+00,
        -1.31544430e+00],
       [-1.50652052e+00,  9.82172869e-02, -1.28338910e+00,
        -1.31544430e+00],
       [-1.02184904e+00,  1.24920112e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-5.37177559e-01,  1.93979142e+00, -1.16971425e+00,
        -1.05217993e+00],
       [-1.50652052e+00,  7.88807586e-01, -1.34022653e+00,
        -1.18381211e+00],
       [-1.02184904e+00,  7.88807586e-01, -1.28338910e+00,
        -1.31544430e+00],
       [-1.74885626e+00, -3.62176246e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00,  9.82172869e-02, -1.28338910e+00,
```

In [22]:
```
x=pd.DataFrame(x)
x
```

Out[22]:

	0	1	2	3
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444
3	-1.506521	0.098217	-1.283389	-1.315444
4	-1.021849	1.249201	-1.340227	-1.315444
...	...	...	...	...
145	1.038005	-0.131979	0.819596	1.448832
146	0.553333	-1.282963	0.705921	0.922303
147	0.795669	-0.131979	0.819596	1.053935
148	0.432165	0.788808	0.933271	1.448832
149	0.068662	-0.131979	0.762758	0.790671

150 rows × 4 columns

In [23]:
```
from sklearn.decomposition import PCA
```

In [24]:
```
pca=PCA()
x_pca=pca.fit_transform(x)
x_pca=pd.DataFrame(x_pca)
x_pca.head()
```

Out[24]:

	0	1	2	3
0	-2.264703	0.480027	-0.127706	-0.024168
1	-2.080961	-0.674134	-0.234609	-0.103007
2	-2.364229	-0.341908	0.044201	-0.028377
3	-2.299384	-0.597395	0.091290	0.065956
4	-2.389842	0.646835	0.015738	0.035923

In [25]:
```
#Varince covered by each person
print("Explained variance",pca.explained_variance_)
print("Proportions of explained variance",pca.explained_variance_ratio_)

import numpy as np
print("Cumulative proportion of explained variance",np.cumsum(pca.explained_v
x_pca["target"]=y
x_pca.columns=["PC1","PC2","PC3","PC4","target"]
x_pca.head()
```

```
Explained variance [2.93808505 0.9201649  0.14774182 0.02085386]
Proportions of explained variance [0.72962445 0.22850762 0.03668922 0.00517871]
Cumulative proportion of explained variance [0.72962445 0.95813207 0.99482129 1.
]
```

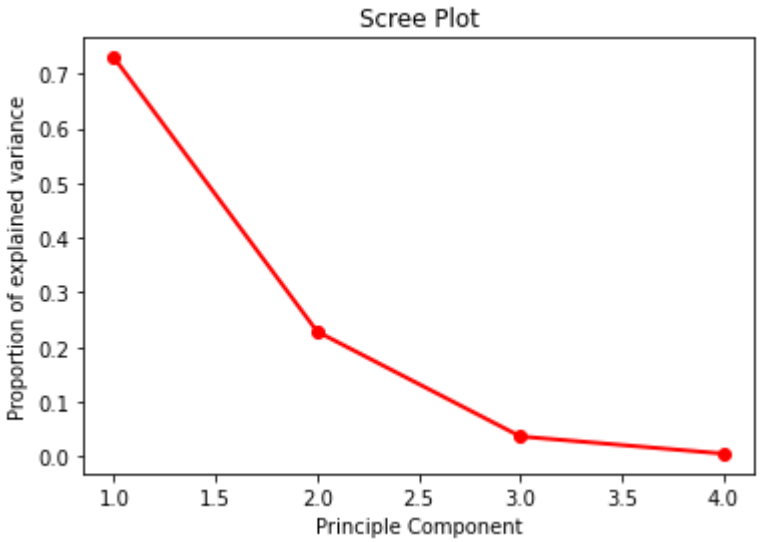
Out[25]:

	PC1	PC2	PC3	PC4	target
0	-2.264703	0.480027	-0.127706	-0.024168	0
1	-2.080961	-0.674134	-0.234609	-0.103007	0
2	-2.364229	-0.341908	0.044201	-0.028377	0
3	-2.299384	-0.597395	0.091290	0.065956	0
4	-2.389842	0.646835	0.015738	0.035923	0

In [27]:
```
import matplotlib.pyplot as plt
```

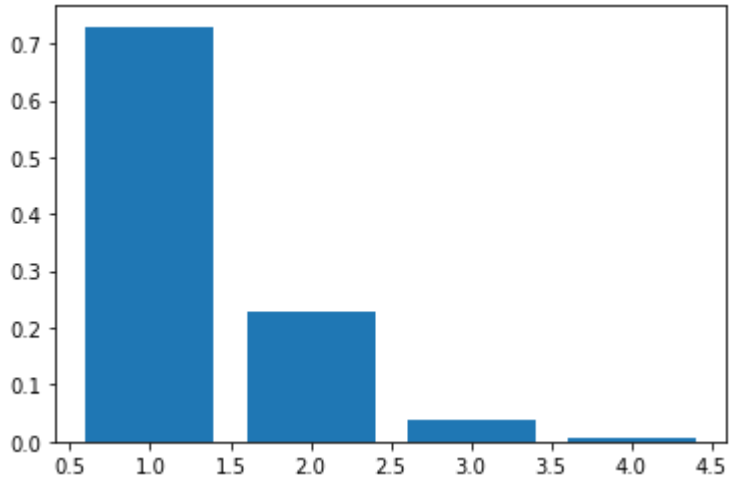
In [35]:
```
PCA_values=np.arange(pca.n_components_)+1

plt.plot(PCA_values,pca.explained_variance_ratio_,"ro-",linewidth=2)
plt.title("Scree Plot")
plt.xlabel("Principle Component")
plt.ylabel("Proportion of explained variance")
plt.show()
```



In [34]:
```
plt.bar(PCA_values,pca.explained_variance_ratio_)
```

Out[34]: <BarContainer object of 4 artists>



In [ ]:
```
1
```