In [1]:
```
import numpy as np
from sklearn import decomposition
X = np.array([[2,6],[1,7]])
```

In [2]:
```
print("Orginal Matrx X and its Shape")
```

```
Orginal Matrx X and its Shape
```

In [3]:
```
print(X)
print("Original Shape:",X.shape)
print("Original matrix\n\n")
```

```
[[2 6]
 [1 7]]
Original Shape: (2, 2)
Original matrix
```

In [4]:
```
pca = decomposition.PCA(n_components=2)
X_pca = pca.fit_transform(X)
```

In [5]:
```
print("Transformed Matrix and its Shape")
print(X_pca)
print("Transformed Shape:",X_pca.shape)
print("Transformed Matrix\n\n")
```

```
Transformed Matrix and its Shape
[[-7.07106781e-01  1.18606713e-17]
 [ 7.07106781e-01  1.18606713e-17]]
Transformed Shape: (2, 2)
Transformed Matrix
```

In [6]:
```
print("After Inverse Transform")
X_new=pca.inverse_transform(X_pca)
print(X_new)
print("After Inverse Transform\n\n\n")
```

```
After Inverse Transform
[[2. 6.]
 [1. 7.]]
After Inverse Transform
```

In [7]:
```
print('Explained variance\n')
print(pca.explained_variance_ratio_)
print('completed\n\n')
```

```
Explained variance

[1.00000000e+00 2.81351049e-34]
completed
```

In [8]:
```
print('Singular values')
print(pca.singular_values_)
print('completed\n\n')
```

```
Singular values
[1.00000000e+00 1.67735223e-17]
completed
```

In [9]:
```
import numpy as np
from numpy.linalg import eig
```

In [10]:
```
X = np.array([[3, 6], [4,7]])
```

In [11]:
```
print("Orginal Matrx X and its Shape")
print(X)
```

```
Orginal Matrx X and its Shape
[[3 6]
 [4 7]]
```

In [12]:
```
print("Original matrix Shape")
print("Original Shape:",X.shape)
```

```
Original matrix Shape
Original Shape: (2, 2)
```

In [13]:
```
M = np.mean(X.T, axis=1)
print("\nMean matrix")
print(M)
```

```
Mean matrix
[3.5 6.5]
```

In [14]:
```
C = X - M
print("\nCentre the matrix")
print(C)
```

```
Centre the matrix
[[-0.5 -0.5]
 [ 0.5  0.5]]
```

In [15]:
```
V = np.cov(C.T)
print("\nCovariance of the matrix\n")
print(V)
```

```
Covariance of the matrix

[[0.5 0.5]
 [0.5 0.5]]
```

In [16]:
```
values, vectors = eig(V)
print('\n Eigen vectors')

print(vectors)
```

```
 Eigen vectors
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
```

In [17]:
```
print('\n Eigen values')
print(values)
```

```
 Eigen values
[1.00000000e+00 1.11022302e-16]
```

In [18]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import KFold
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn import decomposition
import seaborn as sns
```

In [19]:
```
df = pd.read_csv("/home/machine/Downloads/buddymove_holidayiq.csv")
```

In [20]:
```
print(df.head(10))
```

```
   User Id  Sports  Religious  Nature  Theatre  Shopping  Picnic
0   User 1       2         77      79       69        68      95
1   User 2       2         62      76       76        69      68
2   User 3       2         50      97       87        50      75
3   User 4       2         68      77       95        76      61
4   User 5       2         98      54       59        95      86
5   User 6       3         52     109       93        52      76
6   User 7       3         64      85       82        73      69
7   User 8       3         54     107       92        54      76
8   User 9       3         64     108       64        54      93
9  User 10       3         86      76       74        74     103
```

In [21]:
```
df = df.drop(['User Id'],1)
```

In [22]:
```
df
```

Out[22]:

| | Sports | Religious | Nature | Theatre | Shopping | Picnic |
|---|---|---|---|---|---|---|
| **0** | 2 | 77 | 79 | 69 | 68 | 95 |
| **1** | 2 | 62 | 76 | 76 | 69 | 68 |
| **2** | 2 | 50 | 97 | 87 | 50 | 75 |
| **3** | 2 | 68 | 77 | 95 | 76 | 61 |
| **4** | 2 | 98 | 54 | 59 | 95 | 86 |
| **...** | ... | ... | ... | ... | ... | ... |
| **244** | 18 | 139 | 148 | 129 | 129 | 168 |
| **245** | 22 | 114 | 228 | 104 | 84 | 168 |
| **246** | 20 | 124 | 178 | 104 | 158 | 174 |
| **247** | 20 | 133 | 149 | 139 | 144 | 213 |
| **248** | 20 | 143 | 149 | 139 | 159 | 143 |

249 rows × 6 columns

In [23]:
```
array = df.values
X = array[:,0:4]
y = array[:,4]
```

In [24]:
```
X = array[:,0:4]
y = array[:,4]
kfold = KFold(n_splits=10)
model = KNeighborsClassifier(n_neighbors=3)
```

In [25]:
```
score = cross_val_score(model,X,y,cv=10)
```

```
/home/machine/anaconda3/lib/python3.7/site-packages/sklearn/model_s
election/_split.py:667: UserWarning: The least populated class in y
has only 1 members, which is less than n_splits=10.
  % (min_groups, self.n_splits)), UserWarning)
```

In [26]:
```
print('\n\n')
print("Cross score before applying PCA\n")
print(score.mean())
```

```
Cross score before applying PCA

0.08449999999999999
```

In [27]:
```
print("Apply PCA now...")
pca = decomposition.PCA(n_components=1)
X_pca = pca.fit_transform(X)
core = cross_val_score(model,X_pca,y,cv=10)
print('\n\n')
print("Cross score After applying PCA\n")
print(score.mean())
```

```
Apply PCA now...



Cross score After applying PCA

0.08449999999999999

/home/machine/anaconda3/lib/python3.7/site-packages/sklearn/model_s
election/_split.py:667: UserWarning: The least populated class in y
has only 1 members, which is less than n_splits=10.
  % (min_groups, self.n_splits)), UserWarning)
```

In [28]:
```
# calculate the mean of each column
M = np.mean(df.T, axis=1)
print("\nMean matrix")
print(M)
```

```
Mean matrix
Sports        11.987952
Religious    109.779116
Nature       124.518072
Theatre      116.377510
Shopping     112.638554
Picnic       120.401606
dtype: float64
```

In [29]:
```
# center columns by subtracting column means
C = df - M
print("\nCentre the matrix")
print(C)
```

```
Centre the matrix
        Sports  Religious      Nature   Theatre   Shopping     Picn
ic
0    -9.987952 -32.779116  -45.518072 -47.37751 -44.638554 -25.4016
06
1    -9.987952 -47.779116  -48.518072 -40.37751 -43.638554 -52.4016
06
2    -9.987952 -59.779116  -27.518072 -29.37751 -62.638554 -45.4016
06
3    -9.987952 -41.779116  -47.518072 -21.37751 -36.638554 -59.4016
06
4    -9.987952 -11.779116  -70.518072 -57.37751 -17.638554 -34.4016
06
..         ...        ...         ...       ...        ...
...
244   6.012048  29.220884   23.481928  12.62249  16.361446  47.5983
94
245  10.012048   4.220884  103.481928 -12.37751 -28.638554  47.5983
94
246   8.012048  14.220884   53.481928 -12.37751  45.361446  53.5983
94
247   8.012048  23.220884   24.481928  22.62249  31.361446  92.5983
94
248   8.012048  33.220884   24.481928  22.62249  46.361446  22.5983
94

[249 rows x 6 columns]
```

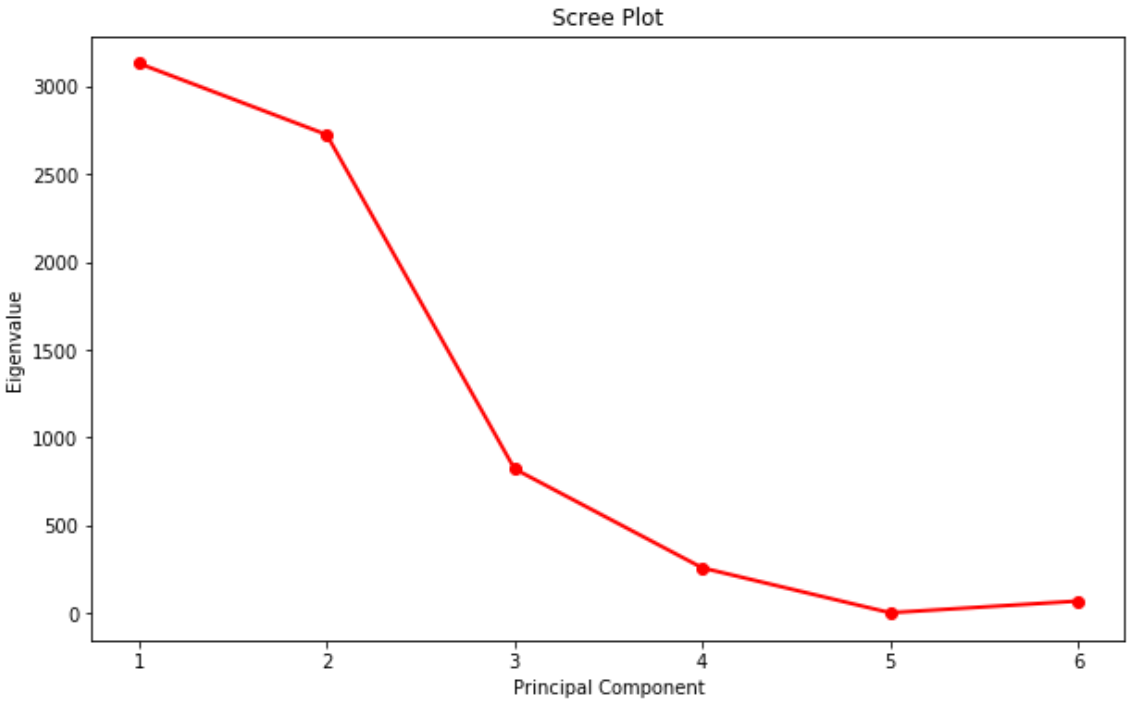
In [30]:
```
# calculate covariance matrix of centered matrix
V = np.cov(C.T)
print("\nCovariance of the matrix\n")
print(V)
```

```
Covariance of the matrix

[[  43.77808006  133.86426351  183.71191216  130.05698601  160.5883
6961
   172.25485814]
 [ 133.86426351 1053.26956212 -219.75607268  177.24098005 1208.1335
5033
   638.88342078]
 [ 183.71191216 -219.75607268 2082.95229304  626.05766615 -353.0619
899
   777.45642246]
 [ 130.05698601  177.24098005  626.05766615 1032.51013732  243.8749
0284
   236.90019756]
 [ 160.58836961 1208.13355033 -353.0619899   243.87490284 1727.4736
6887
   599.79090556]
 [ 172.25485814  638.88342078  777.45642246  236.90019756  599.7909
0556
  1064.93483612]]
```

In [31]:
```
# eigendecomposition of covariance matrix
values, vectors = eig(V)
print('\n Eigen vectors')
print(vectors)
print('\n Eigen values')
print(values)
```

```
 Eigen vectors
[[-0.1004277  -0.06012882  0.01976025 -0.01198683 -0.99260308 -0.02
244488]
 [-0.531791    0.19884391 -0.06902076  0.23449565  0.05528408 -0.78
412437]
 [-0.14649008 -0.84528475 -0.14567747 -0.43162925  0.0734506  -0.22
608325]
 [-0.22209214 -0.28832336  0.87372161  0.3040512   0.05150625  0.09
515899]
 [-0.66164543  0.30582912  0.09372179 -0.57991619  0.04941441  0.34
808839]
 [-0.44559363 -0.25627624 -0.44883238  0.57429468  0.03454163  0.45
089999]]

 Eigen values
[3.13050263e+03 2.72471335e+03 8.21529964e+02 2.57647181e+02
 1.99057792e+00 6.85348730e+01]
```

In [32]:
```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
figure=plt.figure(figsize=(10,6))
sing_vals=np.arange(len(values)) + 1
plt.plot(sing_vals,values, 'ro-', linewidth=2)
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Eigenvalue')
plt.show()
```




In [33]:
```
X = array[:,0:4]
y = array[:,4]
kfold = KFold(n_splits=10)
model = GaussianNB()
```

In [34]:
```
score = cross_val_score(model,X,y,cv=10)
print('\n\n')
print("Cross score before applying PCA\n")
print(score.mean())
```

```
Cross score before applying PCA

0.068

/home/machine/anaconda3/lib/python3.7/site-packages/sklearn/model_s
election/_split.py:667: UserWarning: The least populated class in y
has only 1 members, which is less than n_splits=10.
  % (min_groups, self.n_splits)), UserWarning)
```

In [35]:
```
print("Apply PCA now...")
pca = decomposition.PCA(n_components=1)
X_pca = pca.fit_transform(X)
core = cross_val_score(model,X_pca,y,cv=10)
print('\n\n')
print("Cross score After applying PCA\n")
print(score.mean())
```

```
Apply PCA now...



Cross score After applying PCA

0.068

/home/machine/anaconda3/lib/python3.7/site-packages/sklearn/model_s
election/_split.py:667: UserWarning: The least populated class in y
has only 1 members, which is less than n_splits=10.
  % (min_groups, self.n_splits)), UserWarning)
```

In [36]:
```
df = pd.read_csv("/home/machine/Downloads/Absenteeism.csv")
```

In [37]:
```
print(df.head(10))
```

```
   ID  Reason for absence  Month of absence  Day of the week  Seaso
ns  \
0  11                  26                 7                3
1
1  36                   0                 7                3
1
2   3                  23                 7                4
1
3   7                   7                 7                5
1
4  11                  23                 7                5
1
5   3                  23                 7                6
1
6  10                  22                 7                6
1
7  20                  23                 7                6
1
8  14                  19                 7                2
1
9   1                  22                 7                2
1

   Transportation expense  Distance from Residence to Work  Service
time  Age  \
0                     289                               36
13   33
1                     118                               13
18   50
2                     179                               51
18   38
3                     279                                5
14   39
4                     289                               36
13   33
5                     179                               51
18   38
6                     361                               52
3   28
7                     260                               50
11   36
8                     155                               12
14   34
9                     235                               11
14   37

   Work load Average/day   ...  Disciplinary failure  Education  So
n  \
0                239.554  ...                     0          1
2
1                239.554  ...                     1          1
1
2                239.554  ...                     0          1
0
3                239.554  ...                     0          1
2
4                239.554  ...                     0          1
```

```
2
5                 239.554  ...                    0          1
0
6                 239.554  ...                    0          1
1
7                 239.554  ...                    0          1
4
8                 239.554  ...                    0          1
2
9                 239.554  ...                    0          3
1

   Social drinker  Social smoker  Pet  Weight  Height  Body mass in
dex  \
0               1              0    1      90     172
30
1               1              0    0      98     178
31
2               1              0    0      89     170
31
3               1              1    0      68     168
24
4               1              0    1      90     172
30
5               1              0    0      89     170
31
6               1              0    4      80     172
27
7               1              0    0      65     168
23
8               1              0    0      95     196
25
9               0              0    1      88     172
29

   Absenteeism time in hours
0                          4
1                          0
2                          2
3                          4
4                          2
5                          2
6                          8
7                          4
8                         40
9                          8

[10 rows x 21 columns]
```

In [38]:
```
array = df.values
X = array[:,0:4]
y = array[:,4]
```

In [39]:
```
X = array[:,0:4]
y = array[:,4]
kfold = KFold(n_splits=10)
model = KNeighborsClassifier(n_neighbors=3)
```

In [40]:
```
score = cross_val_score(model,X,y,cv=10)
```

In [41]:
```
print('\n\n')
print("Cross score before applying PCA\n")
print(score.mean())
```

```
Cross score before applying PCA

0.727027027027027
```

In [42]:
```
print("Apply PCA now...")
pca = decomposition.PCA(n_components=1)
X_pca = pca.fit_transform(X)
core = cross_val_score(model,X_pca,y,cv=10)
print('\n\n')
print("Cross score After applying PCA\n")
print(score.mean())
```

```
Apply PCA now...


Cross score After applying PCA

0.727027027027027
```

In [43]:
```
# calculate the mean of each column
M = np.mean(df.T, axis=1)
print("\nMean matrix")
print(M)
```

```
Mean matrix
ID                                18.017568
Reason for absence                19.216216
Month of absence                   6.324324
Day of the week                    3.914865
Seasons                            2.544595
Transportation expense           221.329730
Distance from Residence to Work   29.631081
Service time                      12.554054
Age                               36.450000
Work load Average/day            271.490235
Hit target                        94.587838
Disciplinary failure               0.054054
Education                          1.291892
Son                                1.018919
Social drinker                     0.567568
Social smoker                      0.072973
Pet                                0.745946
Weight                            79.035135
Height                           172.114865
Body mass index                   26.677027
Absenteeism time in hours          6.924324
dtype: float64
```

In [44]:
```
# center columns by subtracting column means
C = df - M
print("\nCentre the matrix")
print(C)
```

```
Centre the matrix
            ID  Reason for absence  Month of absence  Day of the we
ek  \
0    -7.017568            6.783784          0.675676        -0.9148
65
1    17.982432          -19.216216          0.675676        -0.9148
65
2   -15.017568            3.783784          0.675676         0.0851
35
3   -11.017568          -12.216216          0.675676         1.0851
35
4    -7.017568            3.783784          0.675676         1.0851
35
..         ...                 ...               ...
...
735  -7.017568           -5.216216          0.675676        -0.9148
65
736 -17.017568           -8.216216          0.675676        -0.9148
65
737 -14.017568          -19.216216         -6.324324        -0.9148
65
738 -10.017568          -19.216216         -6.324324         0.0851
35
739  16.982432          -19.216216         -6.324324         2.0851
35

      Seasons  Transportation expense  Distance from Residence to W
ork  \
0   -1.544595               67.67027                         6.368
919
1   -1.544595             -103.32973                       -16.631
081
2   -1.544595              -42.32973                        21.368
919
3   -1.544595               57.67027                       -24.631
081
4   -1.544595               67.67027                         6.368
919
..        ...                    ...
...
735 -1.544595               67.67027                         6.368
919
736 -1.544595               13.67027                       -18.631
081
737 -1.544595             -103.32973                       -15.631
081
738 -0.544595                9.67027                         5.368
919
739  0.455405              -42.32973                        15.368
919

     Service time    Age  Work load Average/day   ...  Disciplinary
failure  \
```

```
0        0.445946  -3.45             -31.936235  ...             -
0.054054
1        5.445946  13.55             -31.936235  ...
0.945946
2        5.445946   1.55             -31.936235  ...             -
0.054054
3        1.445946   2.55             -31.936235  ...             -
0.054054
4        0.445946  -3.45             -31.936235  ...             -
0.054054
..            ...    ...                    ...  ...
...
735      0.445946  -3.45              -6.886235  ...             -
0.054054
736      1.445946   0.55              -6.886235  ...             -
0.054054
737      0.445946   3.55              -0.271235  ...             -
0.054054
738      1.445946   2.55              -0.271235  ...             -
0.054054
739      1.445946  16.55              -0.271235  ...             -
0.054054

     Education       Son  Social drinker  Social smoker       Pet
Weight  \
0    -0.291892  0.981081        0.432432      -0.072973  0.254054
10.964865
1    -0.291892 -0.018919        0.432432      -0.072973 -0.745946
18.964865
2    -0.291892 -1.018919        0.432432      -0.072973 -0.745946
9.964865
3    -0.291892  0.981081        0.432432       0.927027 -0.745946 -
11.035135
4    -0.291892  0.981081        0.432432      -0.072973  0.254054
10.964865
..         ...       ...             ...            ...       ...
...
735  -0.291892  0.981081        0.432432      -0.072973  0.254054
10.964865
736   1.708108 -0.018919       -0.567568      -0.072973  0.254054
8.964865
737  -0.291892 -0.018919        0.432432      -0.072973  7.254054
18.964865
738  -0.291892  0.981081        0.432432      -0.072973  1.254054
20.964865
739  -0.291892 -0.018919       -0.567568      -0.072973  0.254054
-2.035135

       Height  Body mass index  Absenteeism time in hours
0   -0.114865         3.322973                  -2.924324
1    5.885135         4.322973                  -6.924324
2   -2.114865         4.322973                  -4.924324
3   -4.114865        -2.677027                  -2.924324
4   -0.114865         3.322973                  -4.924324
..        ...              ...                        ...
735 -0.114865         3.322973                   1.075676
736 -0.114865         2.322973                  -2.924324
737 -2.114865         7.322973                  -6.924324
738 -2.114865         8.322973                  -6.924324
739  2.885135        -1.677027                  -6.924324
```

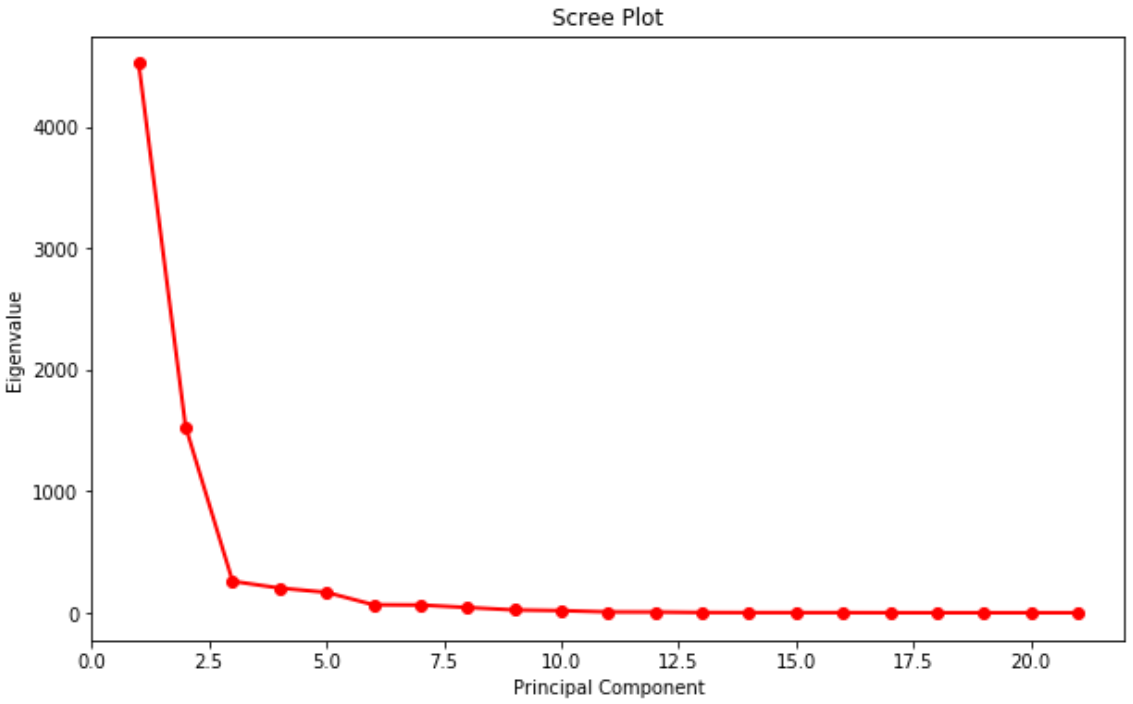
In [45]:
```
# calculate covariance matrix of centered matrix
V = np.cov(C.T)
print("\nCovariance of the matrix\n")
print(V)
```

```
Covariance of the matrix

[[ 1.21467891e+02 -5.97132721e+00 -1.64575943e-03  5.40063270e-01
   1.20692865e+00 -1.65409048e+02 -7.94968932e+01 -1.31788940e+01
   2.92036536e+00  3.97998511e+01  7.82622609e-01  1.12277365e-02
  -2.69004864e-01  3.34966902e-02 -2.46600592e+00 -3.10536518e-02
  -6.01755477e-01 -3.60966939e+01  5.07917017e+00 -1.44963482e+01
  -2.64413561e+00]
 [-5.97132721e+00  7.11223348e+01 -2.43016494e+00  1.39461654e+00
  -1.10573090e+00 -6.74069780e+01  2.02490217e+01  1.79073255e+00
  -4.29499323e+00 -4.06709115e+01  2.83483890e+00 -1.04011996e+00
  -2.68880518e-01 -5.12891782e-01  2.73598362e-01 -2.53958966e-01
  -6.21585049e-01 -2.92579454e-02 -4.03434151e+00  1.34462202e+00
  -1.94626413e+01]
 [-1.64575943e-03 -2.43016494e+00  1.18080679e+01 -3.18911604e-02
   1.55791245e+00  3.16398713e+01 -1.98186007e-01 -9.47189409e-01
  -3.38294993e-02 -2.28151183e+01 -5.97981202e+00  8.39337308e-02
  -1.52982482e-01  2.98321325e-01  9.57831986e-02 -3.45243755e-02
```

In [46]:
```
# eigendecomposition of covariance matrix
values, vectors = eig(V)
print('\n Eigen vectors')
print(vectors)
print('\n Eigen values')
print(values)
```

```
 Eigen vectors
[[-3.80887199e-02  3.09079871e-02  4.73530331e-01 -2.04194374e-01
   6.85141906e-02  8.02138826e-01  1.68772280e-01  2.06541530e-01
   1.18191932e-02  1.49882996e-02  7.82501819e-03  1.04897412e-01
   1.19395708e-02 -2.24167926e-03  1.12221453e-02 -8.43445482e-03
   1.92463882e-02 -2.84096391e-04 -3.75209802e-03 -1.56515402e-02
   6.02372468e-05]
 [-1.47940283e-02 -2.84938531e-02 -1.29116704e-01 -1.34373281e-01
   4.94948993e-02 -1.20508674e-01  9.57689266e-01 -1.53180280e-01
   1.30364989e-02 -7.34737669e-02 -5.04721244e-03 -1.89762795e-02
   1.66551233e-02 -4.61731158e-03  5.36847550e-03  9.75890384e-03
   3.77598481e-03 -1.43477948e-02 -6.17598683e-03  6.31433579e-04
  -1.25489296e-03]
 [ 6.98039573e-03 -1.49587466e-02  1.23743638e-02  1.53148092e-02
   6.86933640e-03  3.21198426e-02 -3.81144821e-02 -5.74619102e-03
  -2.43504929e-01 -5.74051256e-01 -7.51930088e-01 -5.35671185e-02
   3.31147055e-02  2.46917198e-02  7.74405814e-03 -1.94021656e-01
   9.27000948e-03 -4.86138203e-03  4.01486813e-03 -8.77952300e-03
```

In [47]:
```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
figure=plt.figure(figsize=(10,6))
sing_vals=np.arange(len(values)) + 1
plt.plot(sing_vals,values, 'ro-', linewidth=2)
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Eigenvalue')
plt.show()
```




In [ ]: