In [1]:

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.preprocessing import scale
```

In [2]:

```
uni = pd.read_csv("C:/Users/Ashraf/Documents/Datafiles/PCA.csv")
uni.describe()
uni.head()
```

Out[2]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	Brown	1310	89	22	13	22704	94
1	CalTech	1415	100	25	6	63575	81
2	CMU	1260	62	59	9	25026	72
3	Columbia	1310	76	24	12	31510	88
4	Cornell	1280	83	33	13	21864	90

In [3]:

```
# Considering only numerical data
uni.data = uni.iloc[:,1:]
uni.data.head()
# Converting into numpy array
UNI = uni.data.values
UNI
```

```
C:\Users\Ashraf\AppData\Local\Temp/ipykernel_10480/3108596069.py:2: UserWarn
ing: Pandas doesn't allow columns to be created via a new attribute name - s
ee https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-acce
ss (https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-acc
ess)
  uni.data = uni.iloc[:,1:]
```

Out[3]:

```
array([[ 1310,    89,    22,    13, 22704,    94],
       [ 1415,   100,    25,     6, 63575,    81],
       [ 1260,    62,    59,     9, 25026,    72],
       [ 1310,    76,    24,    12, 31510,    88],
       [ 1280,    83,    33,    13, 21864,    90],
       [ 1340,    89,    23,    10, 32162,    95],
       [ 1315,    90,    30,    12, 31585,    95],
       [ 1255,    74,    24,    12, 20126,    92],
       [ 1400,    91,    14,    11, 39525,    97],
       [ 1305,    75,    44,     7, 58691,    87],
       [ 1380,    94,    30,    10, 34870,    91],
       [ 1260,    85,    39,    11, 28052,    89],
       [ 1255,    81,    42,    13, 15122,    94],
       [ 1081,    38,    54,    18, 10185,    80],
       [ 1375,    91,    14,     8, 30220,    95],
       [ 1005,    28,    90,    19,  9066,    69],
       [ 1360,    90,    20,    12, 36450,    93],
       [ 1075,    49,    67,    25,  8704,    67],
       [ 1240,    95,    40,    17, 15140,    78],
       [ 1290,    75,    50,    13, 38380,    87],
       [ 1180,    65,    68,    16, 15470,    85],
       [ 1285,    80,    36,    11, 27553,    90],
       [ 1225,    77,    44,    14, 13349,    92],
       [ 1085,    40,    69,    15, 11857,    71],
       [ 1375,    95,    19,    11, 43514,    96]], dtype=int64)
```

In [4]:

```
# Normalizing the numerical data
uni_normal = scale(UNI)
```

In [5]:

```
uni_normal
```

Out[5]:

```
array([[ 0.41028362,  0.6575195 , -0.88986682,  0.07026045, -0.33141256,
         0.82030265],
       [ 1.39925928,  1.23521235, -0.73465749, -1.68625071,  2.56038138,
        -0.64452351],
       [-0.06065717, -0.76045386,  1.02438157, -0.93346022, -0.16712136,
        -1.65863393],
       [ 0.41028362, -0.02520842, -0.78639393, -0.18066972,  0.29164871,
         0.14422904],
       [ 0.12771914,  0.34241431, -0.32076595,  0.07026045, -0.39084607,
         0.36958691],
       [ 0.69284809,  0.6575195 , -0.83813038, -0.68253005,  0.33778044,
         0.93298158],
       [ 0.4573777 ,  0.71003703, -0.47597528, -0.18066972,  0.29695528,
         0.93298158],
       [-0.10775125, -0.13024348, -0.78639393, -0.18066972, -0.51381683,
         0.59494478],
       [ 1.25797704,  0.76255456, -1.30375836, -0.43159988,  0.85874344,
         1.15833946],
       [ 0.36318954, -0.07772595,  0.24833493, -1.43532055,  2.21481798,
         0.0315501 ],
       [ 1.06960072,  0.92010716, -0.47597528, -0.68253005,  0.52938275,
         0.48226584],
       [-0.06065717,  0.44744937, -0.01034729, -0.43159988,  0.04698077,
         0.25690797],
       [-0.10775125,  0.23737924,  0.14486204,  0.07026045, -0.86787073,
         0.82030265],
       [-1.7466252 , -2.02087462,  0.76569936,  1.32491127, -1.21718409,
        -0.75720245],
       [ 1.02250664,  0.76255456, -1.30375836, -1.18439038,  0.20037583,
         0.93298158],
       [-2.46245521, -2.54604994,  2.6282113 ,  1.57584144, -1.29635802,
        -1.99667073],
       [ 0.88122441,  0.71003703, -0.9933397 , -0.18066972,  0.64117435,
         0.70762371],
       [-1.8031381 , -1.44318177,  1.43827311,  3.08142243, -1.32197103,
        -2.22202861],
       [-0.24903349,  0.97262469,  0.04138915,  1.07398111, -0.86659715,
        -0.98256032],
       [ 0.2219073 , -0.07772595,  0.55875358,  0.07026045,  0.77772991,
         0.0315501 ],
       [-0.81416244, -0.60290126,  1.49000956,  0.82305094, -0.84324827,
        -0.19380777],
       [ 0.17481322,  0.18486171, -0.16555662, -0.43159988,  0.01167444,
         0.36958691],
       [-0.39031573,  0.02730912,  0.24833493,  0.32119061, -0.99331788,
         0.59494478],
       [-1.70894994, -1.91583956,  1.541746  ,  0.57212078, -1.09888311,
        -1.77131286],
       [ 1.02250664,  0.97262469, -1.04507615, -0.43159988,  1.14098185,
         1.04566052]])
```

In [6]:

```
pca = PCA()
pca_values = pca.fit_transform(uni_normal)
```

In [7]:

```
pca_values
```

Out[7]:

```
array([[-1.00987445e+00, -1.06430962e+00,  8.10663051e-02,
         5.69506350e-02, -1.28754245e-01, -3.46496377e-02],
       [-2.82223781e+00,  2.25904458e+00,  8.36828830e-01,
         1.43844644e-01, -1.25961913e-01, -1.80703168e-01],
       [ 1.11246577e+00,  1.63120889e+00, -2.66786839e-01,
         1.07507502e+00, -1.91814148e-01,  3.45679459e-01],
       [-7.41741217e-01, -4.21874699e-02,  6.05008649e-02,
        -1.57208116e-01, -5.77611392e-01,  1.09163092e-01],
       [-3.11912064e-01, -6.35243572e-01,  1.02405189e-02,
         1.71363672e-01,  1.27261287e-02, -1.69212696e-02],
       [-1.69669089e+00, -3.44363283e-01, -2.53407507e-01,
         1.25643278e-02, -5.26606002e-02, -2.71661600e-02],
       [-1.24682093e+00, -4.90983662e-01, -3.20938196e-02,
        -2.05643780e-01,  2.93505340e-01, -7.80119838e-02],
       [-3.38749784e-01, -7.85168589e-01, -4.93584829e-01,
         3.98563085e-02, -5.44978619e-01, -1.55371653e-01],
       [-2.37415013e+00, -3.86538883e-01,  1.16098392e-01,
        -4.53365617e-01, -2.30108300e-01,  2.66983932e-01],
       [-1.40327739e+00,  2.11951503e+00, -4.42827141e-01,
        -6.32543273e-01,  2.30053526e-01, -2.35615124e-01],
       [-1.72610332e+00,  8.82371161e-02,  1.70403663e-01,
         2.60901913e-01,  2.33318380e-01,  2.38968449e-01],
       [-4.50857480e-01, -1.11329480e-02, -1.75746046e-01,
         2.36165626e-01,  2.63250697e-01, -3.14843521e-01],
       [ 4.02381405e-02, -1.00920438e+00, -4.96517167e-01,
         2.29298758e-01,  4.48031921e-01,  4.93921533e-03],
       [ 3.23373034e+00, -3.74580487e-01, -4.95372816e-01,
        -5.21237711e-01, -6.39294809e-01, -9.00477852e-02],
       [-2.23626502e+00, -3.71793294e-01, -3.98993653e-01,
         4.06966479e-01, -4.16760680e-01,  5.06186327e-02],
       [ 5.17299212e+00,  7.79915346e-01, -3.85912331e-01,
        -2.32211711e-01,  1.79286976e-01, -3.09046943e-02],
       [-1.69964377e+00, -3.05597453e-01,  3.18507851e-01,
        -2.97462682e-01, -1.63424678e-01,  1.14422592e-01],
       [ 4.57814600e+00, -3.47591363e-01,  1.49964176e+00,
        -4.54251714e-01, -1.91141971e-01,  1.04149297e-01],
       [ 8.22603117e-01, -6.98906146e-01,  1.42781145e+00,
         7.60778800e-01,  1.84260335e-01, -2.51103268e-01],
       [-9.77621343e-02,  6.50446454e-01,  1.00508440e-01,
        -5.00097185e-01,  4.87217823e-01,  2.19242132e-01],
       [ 1.96318260e+00, -2.24767561e-01, -2.55881433e-01,
        -4.84741049e-02,  8.22745655e-01,  1.52246521e-01],
       [-5.42288939e-01, -7.95888376e-02, -3.05393475e-01,
         1.31698758e-01,  5.27399148e-02, -3.67264440e-02],
       [ 5.32220920e-01, -1.01716720e+00, -4.23716362e-01,
         1.69535706e-01,  3.57813210e-01, -6.60989993e-02],
       [ 3.54869664e+00,  7.78461666e-01, -4.49363319e-01,
         3.23678618e-01, -3.58332564e-01, -7.74564151e-02],
       [-2.30590032e+00, -1.17704318e-01,  2.53988661e-01,
        -5.16183372e-01,  5.58940129e-02, -1.07932007e-02]])
```

In [8]:

```
pca = PCA(n_components = 6)
pca_values = pca.fit_transform(uni_normal)
```

In [9]:

```
pca_values
```

Out[9]:

```
array([[-1.00987445e+00, -1.06430962e+00,  8.10663051e-02,
         5.69506350e-02, -1.28754245e-01, -3.46496377e-02],
       [-2.82223781e+00,  2.25904458e+00,  8.36828830e-01,
         1.43844644e-01, -1.25961913e-01, -1.80703168e-01],
       [ 1.11246577e+00,  1.63120889e+00, -2.66786839e-01,
         1.07507502e+00, -1.91814148e-01,  3.45679459e-01],
       [-7.41741217e-01, -4.21874699e-02,  6.05008649e-02,
        -1.57208116e-01, -5.77611392e-01,  1.09163092e-01],
       [-3.11912064e-01, -6.35243572e-01,  1.02405189e-02,
         1.71363672e-01,  1.27261287e-02, -1.69212696e-02],
       [-1.69669089e+00, -3.44363283e-01, -2.53407507e-01,
         1.25643278e-02, -5.26606002e-02, -2.71661600e-02],
       [-1.24682093e+00, -4.90983662e-01, -3.20938196e-02,
        -2.05643780e-01,  2.93505340e-01, -7.80119838e-02],
       [-3.38749784e-01, -7.85168589e-01, -4.93584829e-01,
         3.98563085e-02, -5.44978619e-01, -1.55371653e-01],
       [-2.37415013e+00, -3.86538883e-01,  1.16098392e-01,
        -4.53365617e-01, -2.30108300e-01,  2.66983932e-01],
       [-1.40327739e+00,  2.11951503e+00, -4.42827141e-01,
        -6.32543273e-01,  2.30053526e-01, -2.35615124e-01],
       [-1.72610332e+00,  8.82371161e-02,  1.70403663e-01,
         2.60901913e-01,  2.33318380e-01,  2.38968449e-01],
       [-4.50857480e-01, -1.11329480e-02, -1.75746046e-01,
         2.36165626e-01,  2.63250697e-01, -3.14843521e-01],
       [ 4.02381405e-02, -1.00920438e+00, -4.96517167e-01,
         2.29298758e-01,  4.48031921e-01,  4.93921533e-03],
       [ 3.23373034e+00, -3.74580487e-01, -4.95372816e-01,
        -5.21237711e-01, -6.39294809e-01, -9.00477852e-02],
       [-2.23626502e+00, -3.71793294e-01, -3.98993653e-01,
         4.06966479e-01, -4.16760680e-01,  5.06186327e-02],
       [ 5.17299212e+00,  7.79915346e-01, -3.85912331e-01,
        -2.32211711e-01,  1.79286976e-01, -3.09046943e-02],
       [-1.69964377e+00, -3.05597453e-01,  3.18507851e-01,
        -2.97462682e-01, -1.63424678e-01,  1.14422592e-01],
       [ 4.57814600e+00, -3.47591363e-01,  1.49964176e+00,
        -4.54251714e-01, -1.91141971e-01,  1.04149297e-01],
       [ 8.22603117e-01, -6.98906146e-01,  1.42781145e+00,
         7.60778800e-01,  1.84260335e-01, -2.51103268e-01],
       [-9.77621343e-02,  6.50446454e-01,  1.00508440e-01,
        -5.00097185e-01,  4.87217823e-01,  2.19242132e-01],
       [ 1.96318260e+00, -2.24767561e-01, -2.55881433e-01,
        -4.84741049e-02,  8.22745655e-01,  1.52246521e-01],
       [-5.42288939e-01, -7.95888376e-02, -3.05393475e-01,
         1.31698758e-01,  5.27399148e-02, -3.67264440e-02],
       [ 5.32220920e-01, -1.01716720e+00, -4.23716362e-01,
         1.69535706e-01,  3.57813210e-01, -6.60989993e-02],
       [ 3.54869664e+00,  7.78461666e-01, -4.49363319e-01,
         3.23678618e-01, -3.58332564e-01, -7.74564151e-02],
       [-2.30590032e+00, -1.17704318e-01,  2.53988661e-01,
        -5.16183372e-01,  5.58940129e-02, -1.07932007e-02]])
```

In [10]:

```
# The amount of variance that each PCA explains is
var = pca.explained_variance_ratio_
var
```

Out[10]:

```
array([0.76868084, 0.13113602, 0.04776031, 0.02729668, 0.0207177 ,
       0.00440844])
```

In [11]:

```
# Cumulative variance
var1 = np.cumsum(np.round(var,decimals = 4)*100)
var1
```

Out[11]:

```
array([ 76.87,  89.98,  94.76,  97.49,  99.56, 100.  ])
```

In [12]:

```
pca.components_
```

Out[12]:

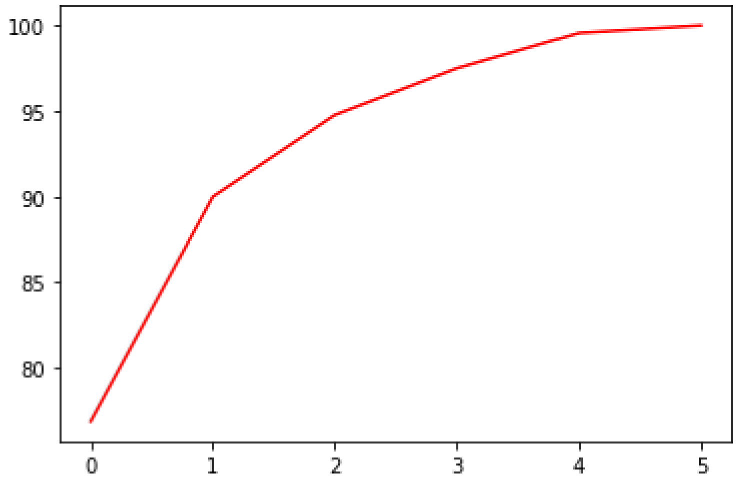
```
array([[-0.45774863, -0.42714437,  0.42430805,  0.39064831, -0.36252316,
        -0.37940403],
       [ 0.03968044, -0.19993153,  0.32089297, -0.43256441,  0.6344864 ,
        -0.51555367],
       [ 0.1870388 ,  0.49780855, -0.15627899,  0.60608085,  0.20474114,
        -0.53247261],
       [ 0.13124033,  0.37489567,  0.0612872 , -0.50739095, -0.62340055,
        -0.43863341],
       [ 0.02064583,  0.4820162 ,  0.8010936 ,  0.07682369,  0.07254775,
         0.33810965],
       [ 0.8580547 , -0.39607492,  0.21693361,  0.1720479 , -0.17376309,
        -0.00353754]])
```

In [13]:

```
# Variance plot for PCA components obtained
plt.plot(var1,color="red")
```

Out[13]:

```
[<matplotlib.lines.Line2D at 0x860355a3a0>]
```



In [14]:

```
pca_values[:,0:1]
```

Out[14]:

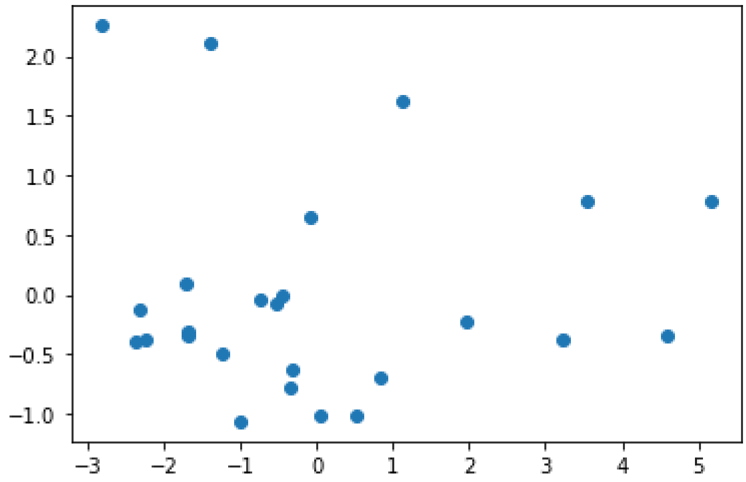
```
array([[-1.00987445],
       [-2.82223781],
       [ 1.11246577],
       [-0.74174122],
       [-0.31191206],
       [-1.69669089],
       [-1.24682093],
       [-0.33874978],
       [-2.37415013],
       [-1.40327739],
       [-1.72610332],
       [-0.45085748],
       [ 0.04023814],
       [ 3.23373034],
       [-2.23626502],
       [ 5.17299212],
       [-1.69964377],
       [ 4.578146  ],
       [ 0.82260312],
       [-0.09776213],
       [ 1.9631826 ],
       [-0.54228894],
       [ 0.53222092],
       [ 3.54869664],
       [-2.30590032]])
```

In [15]:

```
# plot between PCA1 and PCA2
x = pca_values[:,0:1]
y = pca_values[:,1:2]
#z = pca_values[:2:3]
plt.scatter(x,y)
```

Out[15]:

```
<matplotlib.collections.PathCollection at 0x860364ae50>
```



In [16]:

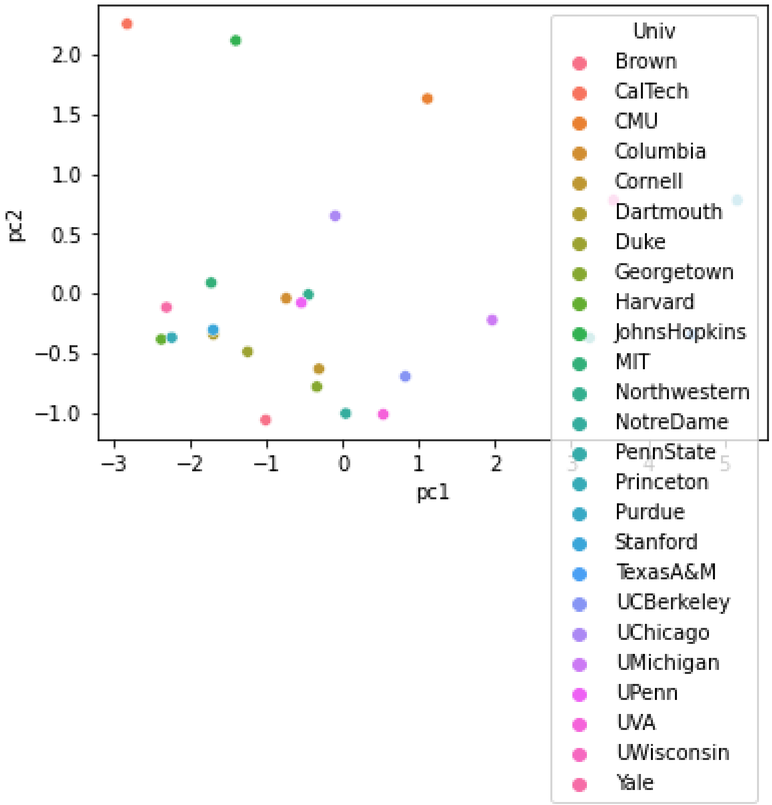
```
finalDf = pd.concat([pd.DataFrame(pca_values[:,0:2],columns=['pc1','pc2']), uni[['Univ']]],
```

In [17]:

```
import seaborn as sns
sns.scatterplot(data=finalDf,x='pc1',y='pc2',hue='Univ')
```

Out[17]:

```
<AxesSubplot:xlabel='pc1', ylabel='pc2'>
```



In [ ]:

```
```