In [37]:
```
import pandas as pd
import numpy as np

from sklearn.decomposition import PCA
from sklearn import preprocessing

import matplotlib.pylab as plt

import dmba
%matplotlib inline
```

In [ ]:
```

```

In [38]:
```
# load the data and review

universities_df = dmba.load_data('Universities.csv')
universities_df.head()
```

Out[38]:

	College Name	State	Public (1)/ Private (2)	# appli. rec'd	# appl. accepted	# new stud. enrolled	% new stud. from top 10%	% new stud. from top 25%	# FT undergrad	# PT undergrad	in-state tuition	out-of-state tuition	room	board	add. fees	estim. book costs	estim. personal $	% fac. w/PHD	Graduation rate
0	Alaska Pacific University	AK	2	193.0	146.0	55.0	16.0	44.0	249.0	869.0	7560.0	7560.0	1620.0	2500.0	130.0	800.0	1500.0	76.0	11.9	15.0
1	University of Alaska at Fairbanks	AK	1	1852.0	1427.0	928.0	NaN	NaN	3885.0	4519.0	1742.0	5226.0	1800.0	1790.0	155.0	650.0	2304.0	67.0	10.0	NaN
2	University of Alaska Southeast	AK	1	146.0	117.0	89.0	4.0	24.0	492.0	1849.0	1742.0	5226.0	2514.0	2250.0	34.0	500.0	1162.0	39.0	9.5	39.0
3	University of Alaska at Anchorage	AK	1	2065.0	1598.0	1162.0	NaN	NaN	6209.0	10537.0	1742.0	5226.0	2600.0	2520.0	114.0	580.0	1260.0	48.0	13.7	NaN
4	Alabama Agric & Mech. Univ.	AL	1	2817.0	1920.0	984.0	NaN	NaN	3958.0	305.0	1700.0	3400.0	1108.0	1442.0	155.0	500.0	850.0	53.0	14.3	40.0

In [39]:
```
var = list(universities_df.columns)
var
```

Out[39]:
```
['College Name',
 'State',
 'Public (1)/ Private (2)',
 '# appli. rec'd',
 '# appl. accepted',
 '# new stud. enrolled',
 '% new stud. from top 10%',
 '% new stud. from top 25%',
 '# FT undergrad',
 '# PT undergrad',
 'in-state tuition',
 'out-of-state tuition',
 'room',
 'board',
 'add. fees',
 'estim. book costs',
 'estim. personal $',
 '% fac. w/PHD',
 'stud./fac. ratio',
 'Graduation rate']
```

In [40]:
```
# variable data types
universities_df.dtypes
```

Out[40]:
```
College Name                  object
State                         object
Public (1)/ Private (2)        int64
# appli. rec'd                float64
# appl. accepted              float64
# new stud. enrolled          float64
% new stud. from top 10%      float64
% new stud. from top 25%      float64
# FT undergrad                float64
# PT undergrad                float64
in-state tuition              float64
out-of-state tuition          float64
room                          float64
board                         float64
add. fees                     float64
estim. book costs             float64
estim. personal $             float64
% fac. w/PHD                  float64
stud./fac. ratio              float64
Graduation rate               float64
dtype: object
```

1

In [41]:
```
# remove all three categorical variables
var.remove('College Name')
var.remove('State')
var.remove('Public (1)/ Private (2)')
```

In [42]:
```
all_numeric_df = universities_df[var]
all_numeric_df.shape
```

Out[42]:
```
(1302, 17)
```

In [43]:
```
# drop missing values
all_numeric_df = all_numeric_df.dropna(how='any')
universities_df = universities_df.dropna(how='any')
all_numeric_df.shape
```

Out[43]:
```
(471, 17)
```

2

In [44]:
```
# PCA

pcs = PCA()
pcs.fit(all_numeric_df)

# view the importance of principal components
pcsSummary_df = pd.DataFrame({'Standard deviation': np.sqrt(pcs.explained_variance_),
                              'Proportion of variance': pcs.explained_variance_ratio_,
                              'Cumulative proportion': np.cumsum(pcs.explained_variance_ratio_)})
pcsSummary_df = pcsSummary_df.transpose()
pcsSummary_df.columns = ['PC{}'.format(i) for i in range(1, len(pcsSummary_df.columns) + 1)]
pcsSummary_df.round(4)
```

Out[44]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17
Standard deviation	7430.9140	5987.9890	1854.6412	1192.5293	967.4279	679.6527	596.9761	580.6299	417.6136	318.1272	188.8676	155.6062	19.0491	12.5287	11.0184	5.33	2.9059
Proportion of variance	0.5614	0.3645	0.0350	0.0145	0.0095	0.0047	0.0036	0.0034	0.0018	0.0010	0.0004	0.0002	0.0000	0.0000	0.0000	0.00	0.0000
Cumulative proportion	0.5614	0.9259	0.9609	0.9753	0.9848	0.9895	0.9932	0.9966	0.9984	0.9994	0.9997	1.0000	1.0000	1.0000	1.0000	1.00	1.0000

In [45]:
```
# Components

print('\nComponents')
pcsComponents_df = pd.DataFrame(pcs.components_.transpose(),columns=pcsSummary_df.columns, index=[var])
pcsComponents_df
```

Out[45]:

Components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	
# appli. rec'd	0.271883	0.551183	0.664458	0.129476	-0.034246	0.370333	-0.120305	-0.097471	-0.035166	-0.009102	-0.016696	-0.008734	0.005788	0.000754	-0.002059	-0.001503	-0.0
# appl. accepted	0.194107	0.321299	0.190957	-0.008357	-0.076674	-0.813924	0.353520	0.103440	0.075971	-0.040233	0.103389	0.016789	-0.011578	-0.002453	0.003683	0.003584	0.0
# new stud. enrolled	0.084730	0.101590	-0.087451	-0.055253	-0.030608	-0.081429	0.019293	0.039063	0.030435	0.170403	-0.965233	-0.008420	0.013475	-0.002658	-0.005389	-0.006643	-0.0
% new stud. from top 10%	-0.000898	0.001732	0.000136	-0.001906	0.001236	0.009145	-0.003462	-0.002851	-0.001772	0.003426	-0.013176	0.004417	-0.511022	-0.251865	0.230605	0.781370	0.1
% new stud. from top 25%	-0.000811	0.001925	0.000040	-0.002352	0.001009	0.007166	-0.003192	-0.002603	-0.000749	0.001050	-0.006792	0.007512	-0.686812	-0.225707	0.320215	-0.610197	-0.0
# FT undergrad	0.458121	0.492263	-0.635303	-0.284582	-0.080402	0.129196	-0.127077	0.011595	-0.021579	-0.012258	0.152137	-0.000211	0.001405	0.001129	0.000611	0.000641	-0.0
# PT undergrad	0.108253	0.073410	-0.285353	0.942562	-0.051743	-0.039789	-0.018146	-0.073893	-0.044043	0.031981	0.001279	-0.003901	-0.003181	0.001234	0.001414	0.000195	0.0
in-state tuition	-0.670187	0.382489	-0.082787	-0.016972	-0.621759	0.000517	-0.060641	0.006407	-0.040511	0.070211	0.022783	-0.003362	0.000362	-0.000392	-0.002251	-0.000151	0.0
out-of-state tuition	-0.454535	0.429685	-0.129410	0.018657	0.749634	0.010286	0.141481	-0.091839	-0.000236	-0.056493	-0.016404	0.005916	0.004394	-0.000145	0.002655	-0.000479	0.0
room	-0.033420	0.055584	0.040113	0.065120	0.115354	-0.050083	-0.314426	0.873995	-0.310999	-0.053459	-0.065262	-0.050179	-0.001090	-0.000420	0.000894	-0.000278	0.0
board	-0.034236	0.040897	-0.008232	0.067313	0.006301	0.067317	-0.145646	0.276553	0.938293	-0.095646	-0.009815	-0.025704	-0.000158	-0.002073	0.002044	0.000795	-0.0
add. fees	0.013209	0.008746	0.032868	-0.012755	0.103097	-0.024987	-0.043534	0.068703	0.076206	0.972820	0.168601	0.001859	0.002655	-0.002461	0.002785	-0.001050	0.0
estim. book costs	-0.000058	0.003291	0.000316	0.010795	-0.005223	0.034097	0.011155	0.064045	0.006482	-0.002434	-0.014164	0.997041	0.005360	0.001904	-0.009205	0.000856	0.0
estim. personal $	0.037557	0.001185	-0.054659	0.031666	-0.106952	0.407928	0.835394	0.339141	0.002346	0.039237	-0.013501	-0.046075	-0.001610	0.000984	-0.000207	-0.000442	0.0
% fac. w/PHD	-0.000205	0.001564	-0.000995	-0.000055	0.004822	0.000925	-0.001184	0.000218	0.001612	0.002839	0.000403	-0.006167	-0.396144	-0.066826	-0.915632	-0.009347	-0.0
stud./fac. ratio	0.000295	-0.000159	0.000025	-0.000135	-0.000201	-0.000748	-0.000397	-0.000359	0.000470	-0.000638	0.000849	-0.000424	0.012433	0.033745	-0.014961	-0.112121	0.9
Graduation rate	-0.001072	0.001397	0.000920	-0.002172	0.001129	0.001077	-0.001994	-0.000387	0.001937	0.004057	-0.007366	0.000551	-0.331120	0.938075	0.074297	0.066329	-0.0

In [46]:
```
# PCA after normalization

pcs = PCA()
transformed_df =pcs.fit_transform(preprocessing.scale(all_numeric_df))

# view the importance of principal components
pcsSummary_df = pd.DataFrame({'Standard deviation': np.sqrt(pcs.explained_variance_),
                              'Proportion of variance': pcs.explained_variance_ratio_,
                              'Cumulative proportion': np.cumsum(pcs.explained_variance_ratio_)})
pcsSummary_df = pcsSummary_df.transpose()
pcsSummary_df.columns = ['PC{}'.format(i) for i in range(1, len(pcsSummary_df.columns) + 1)]
pcsSummary_df.round(4)
```

Out[46]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17
Standard deviation	2.2773	2.1449	1.0995	1.0336	0.9770	0.8728	0.8041	0.7736	0.7039	0.6629	0.6285	0.5503	0.4388	0.3042	0.2002	0.1745	0.1440
Proportion of variance	0.3044	0.2700	0.0710	0.0627	0.0560	0.0448	0.0380	0.0351	0.0291	0.0258	0.0232	0.0178	0.0113	0.0054	0.0024	0.0018	0.0012
Cumulative proportion	0.3044	0.5745	0.6454	0.7081	0.7642	0.8090	0.8469	0.8821	0.9111	0.9369	0.9601	0.9779	0.9892	0.9946	0.9970	0.9988	1.0000

In [47]:
```
# Components

print('\nComponents')
pcsComponents_df = pd.DataFrame(pcs.components_.transpose(),columns=pcsSummary_df.columns, index=[var])
pcsComponents_df
```

Out[47]:

Components

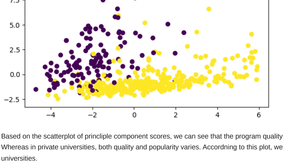
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	
# appli. rec'd	0.278361	0.420164	-0.031982	0.072621	-0.016694	0.112320	-0.268145	-0.093570	0.039628	-0.087361	-0.073021	-0.009995	0.602996	0.198790	0.346774	-0.344637	-0.2
# appl. accepted	0.023659	0.434471	-0.031423	0.118128	-0.089073	0.114381	-0.266285	-0.080991	0.022795	0.035197	-0.166046	-0.062100	0.251257	-0.240232	-0.452347	0.429830	0.3
# new stud. enrolled	-0.028802	0.445556	-0.038651	-0.031466	-0.075981	0.054079	-0.098870	-0.058138	0.096336	0.019353	-0.072613	0.013719	-0.486306	0.059301	-0.322663	-0.010969	-0.6
% new stud. from top 10%	0.354028	0.093547	-0.120129	-0.372457	0.162260	-0.004445	0.102709	-0.112334	0.028676	-0.326675	0.209275	-0.043489	-0.003825	0.646399	-0.185719	0.168396	0.1
% new stud. from top 25%	0.340496	0.118396	-0.142720	-0.385565	0.158187	0.092636	0.136409	-0.039927	-0.006007	-0.314110	0.234355	0.010823	0.037524	-0.685605	0.088571	-0.055470	-0.1
# FT undergrad	-0.049586	0.443583	-0.004012	-0.056459	-0.094781	0.043504	-0.043157	-0.043464	0.034858	-0.009057	-0.061392	0.050779	-0.512673	0.012862	0.441354	-0.217176	0.5
# PT undergrad	-0.106380	0.287700	0.265769	0.953495	-0.343681	-0.188041	0.509297	0.122490	0.172351	0.225459	0.531642	-0.107999	0.168015	-0.006459	-0.036556	0.009390	0.0
in-state tuition	0.379389	-0.150248	0.084350	0.041064	-0.172639	-0.000539	-0.129328	0.009974	0.092325	0.103905	-0.044406	-0.497755	-0.066563	-0.041638	-0.355983	-0.592824	0.1
out-of-state tuition	0.402555	-0.048728	0.051577	0.077658	0.158499	-0.044407	-0.077965	0.010688	0.044615	0.151510	-0.099283	-0.507936	-0.101073	0.006317	0.449517	0.507584	-0.1
room	0.273165	0.052271	0.250578	0.454416	0.004482	-0.015068	-0.122402	-0.091329	-0.680595	-0.180139	0.309060	0.153113	-0.120440	0.004846	-0.027487	-0.009710	-0.0
board	0.290437	0.010051	0.252096	0.301620	-0.199067	-0.038477	0.152138	0.466412	0.421139	-0.419434	-0.181506	0.302579	-0.004475	0.001880	-0.011574	0.033039	-0.0
add. fees	-0.012351	0.169499	-0.249747	0.446562	0.648920	-0.418437	0.082359	0.048174	0.205132	-0.013333	0.098383	-0.214784	-0.043950	-0.039396	-0.006340	-0.039115	0.0
estim. book costs	0.057302	0.056689	0.652241	-0.044356	0.518644	0.421195	0.190728	-0.130177	0.078975	0.170842	-0.172220	-0.033709	-0.006943	0.009861	-0.004617	0.003225	0.0
estim. personal $	-0.144908	0.156837	0.403735	-0.403709	0.103358	-0.466598	-0.289271	0.505883	-0.194258	0.000128	0.038711	-0.133197	0.051416	-0.013721	-0.012993	0.017390	-0.0
% fac. w/PHD	0.254201	0.196852	-0.189367	-0.074609	-0.017278	-0.180623	0.533349	0.145009	-0.404837	0.267768	-0.502748	0.096447	0.101835	0.027889	-0.950578	-0.073086	-0.0
stud./fac. ratio	-0.278542	0.101034	-0.187598	0.105222	0.003000	0.522154	0.211516	0.525627	-0.207194	-0.181720	0.136107	-0.416236	0.005232	0.065855	-0.006340	0.010918	-0.0
Graduation rate	0.325305	0.024264	-0.181888	-0.012600	0.109137	0.215325	-0.200105	0.389370	0.112811	0.598537	0.353414	0.333592	-0.009703	0.049079	0.006724	0.009623	0.0

Because the different variables are measured on different scales, we must first normalize the data, otherwise the variables with the largest scale will dominate the principal components. From the non-standardized output we see that the first component account for over 50% of the total variability (and the first 6 components accounts for over 98% of the total variability). When we standardize the data, those numbers drop to 30% and 81%.

The 1st component appears to capture program quality (e.g. student and faculty quality) and expenses (tuition, room, board). The 2nd component is related to the popularity of the program (as reflected by # application received, accepted, # new enrolled students, and # FT undergraduates). The 3rd component is mainly estimated book and personal costs, and the 4th contrasts room & board costs with student quality (% new students from top 10% and top 25%).

3

In [50]:
```
plot = plt.scatter(transformed_df[:,0],transformed_df[:,1], c = universities_df['Public (1)/ Private (2)'])
plot = plt.legend(handles=plot.legend_elements()[0], labels=list(universities_df['Public (1)/ Private (2)']))
```



Based on the scatterplot of principle component scores, we can see that the program quality varies quite a bit in public universities but the popularity does not vary that much. Whereas in private universities, both quality and popularity varies. According to this plot, we can potentially use the first two principal components to classify public and private universities.