CoxAssignment02

Task: This assignment will take you through the process of understanding, implementing, and interpreting PCA.

Dataset: Please use the Universities.csv for this exercise.

In [3]:
```
# import required packages for this assignment
import pandas as pd
import numpy as np

from sklearn.decomposition import PCA
from sklearn import preprocessing
```

Principal Component Analysis

The dataset on American college and university rankings (available from www.dataminingbook.com) contains information on 1302 American colleges and universities offering an undergraduate program. For each university, there are 17 measurements that include continuous measurements (such as tuition and graduation rate) and categorical measurements (such as location by state and whether it is a private or a public school).

Feature	Description
Private	Public/private indicator
Apps	Number of applications received
Accept	Number of applicants accepted
Enroll	Number of new students enrolled
Top10perc	New students from top 10% of high school class
Top25perc	New students from top 25% of high school class
F.Undergrad	Number of full-time undergraduates
P.Undergrad	Number of part-time undergraduates
Outstate	Out-of-state tuition
Room.Board	Room and board costs
Books	Estimated book costs
Personal	Estimated personal spending
PhD	Percent of faculty with Ph.D.'s
Terminal	Percent of faculty with terminal degree

Feature	Description
S.F.Ratio	Student/faculty ratio
perc.alumni	Percent of alumni who donate
Expend	Instructional expenditure per student
Grad.Rate	Graduation rate

Data Exploration

In [12]:
```
# load the data and review
unis = pd.read_csv('Universities.csv')
unis.head(10)
```

Out[12]:

	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	i sta tuitic
0	Alaska Pacific University	AK	2	193.0	146.0	55.0	16.0	44.0	249.0	869.0	756(
1	University of Alaska at Fairbanks	AK	1	1852.0	1427.0	928.0	NaN	NaN	3885.0	4519.0	1742
2	University of Alaska Southeast	AK	1	146.0	117.0	89.0	4.0	24.0	492.0	1849.0	1742
3	University of Alaska at Anchorage	AK	1	2065.0	1598.0	1162.0	NaN	NaN	6209.0	10537.0	1742
4	Alabama Agri. & Mech. Univ.	AL	1	2817.0	1920.0	984.0	NaN	NaN	3958.0	305.0	170(
5	Faulkner University	AL	2	345.0	320.0	179.0	NaN	27.0	1367.0	578.0	560(
6	University of Montevallo	AL	1	1351.0	892.0	570.0	18.0	78.0	2385.0	331.0	222(
7	Alabama State University	AL	1	4639.0	3272.0	1278.0	NaN	NaN	4051.0	405.0	150(
8	Auburn University-Main Campus	AL	1	7548.0	6791.0	3070.0	25.0	57.0	16262.0	1716.0	210(
9	Birmingham-Southern College	AL	2	805.0	588.0	287.0	67.0	88.0	1376.0	207.0	1166(

Print the features and datatypes of them.

In [70]:
```
unis.dtypes
```

Out[70]:
```
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
```

Data Cleaning

Remove all categorical variables in the df and print the shape.

In [24]:
```
unis2 = unis.drop(['College Name','State', 'Public (1)/ Private (2)'], axis=1)
unis2.shape
```

Out[24]:
```
(1302, 17)
```

In [10]:
```
# put the new data in a dataframe and print the shape.
```

Out[10]:
```
(1302, 17)
```

Remove all records with missing numerical measurements from the dataset. Print the shape. How many values were dropped?

In [49]:
```
unis3 = unis2.dropna(axis=0)
unis3.shape
```

Out[49]:
```
(471, 17)
```

831 values were dropped.

In [11]:
```
# drop missing values and print the shape
```

Out[11]:
```
(471, 17)
```

Data Transformation

Conduct a principal components analysis without standardizing the cleaned data and comment on the results. Round to the fourth decimal.

In [103…
```
unis4 = preprocessing.normalize(unis3)
unis5 = pd.DataFrame(unis4, columns=unis3.columns)
unis5
```

Out[103]:

	# 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	
0	0.017128	0.012957	0.004881	0.001420	0.003905	0.022097	0.077119	0.670906	0.670906	0.1
1	0.021288	0.017059	0.012977	0.000583	0.003499	0.071737	0.269598	0.253996	0.761989	0.3
2	0.047597	0.034767	0.016970	0.003962	0.005203	0.081359	0.012239	0.689424	0.689424	0.1
3	0.051136	0.043735	0.010681	0.002187	0.003953	0.045249	0.010597	0.679570	0.679570	0.1
4	0.454218	0.154356	0.034473	0.003087	0.006174	0.093437	0.012246	0.583054	0.583054	0.1
...	...	...	...	...	...	...	...	...	...	
466	0.413893	0.326549	0.141384	0.001445	0.003757	0.615241	0.091318	0.159518	0.505428	0.1
467	0.011496	0.009681	0.005672	0.001286	0.003101	0.021328	0.001664	0.688249	0.688249	0.1
468	0.038389	0.032848	0.011345	0.001385	0.002902	0.052702	0.005145	0.690466	0.690466	0.1
469	0.357258	0.309917	0.169668	0.001184	0.005072	0.651198	0.197735	0.140333	0.430638	0.1
470	0.189415	0.141524	0.100169	0.002147	0.004294	0.703421	0.138910	0.178119	0.559002	0.1

471 rows × 17 columns

In [99]:
```
pcsSummary = pd.DataFrame({'Standard Deviation': np.sqrt(pcs.explained_variance_),
                           'Proportion of variance': pcs.explained_variance_ratio_,
                          'Cumulative proportion': np.cumsum(pcs.explained_variance_rat
pcsSummary.round(4)
```

Out[99]:

	Standard Deviation	Proportion of variance	Cumulative proportion
0	7430.9140	0.5614	0.5614
1	5987.9890	0.3645	0.9259
2	1854.6412	0.0350	0.9609
3	1192.5293	0.0145	0.9753
4	967.4279	0.0095	0.9848
5	679.6527	0.0047	0.9895
6	596.9761	0.0036	0.9932
7	580.6299	0.0034	0.9966
8	417.6136	0.0018	0.9984
9	318.1272	0.0010	0.9994
10	188.8676	0.0004	0.9997
11	155.6062	0.0002	1.0000
12	19.0491	0.0000	1.0000
13	12.5287	0.0000	1.0000
14	11.0184	0.0000	1.0000
15	5.3300	0.0000	1.0000
16	2.9059	0.0000	1.0000

Based on the results I would utilize the first 2 PCs as after the first 2 the cumulative proportion covers 92.59% of all the data and PC3 does not have a significant increase that is worth noting.

Print the loadings. Interpret the output. Why could understanding or interpretting this data be helpful to understanding or implementing PCA?

In [108…

```
unis5.transpose()
```

Out[108]:

	0	1	2	3	4	5	6	7	8
# appli. rec'd	0.017128	0.021288	0.047597	0.051136	0.454218	0.163639	0.056085	0.064224	0.062915
# appl. accepted	0.012957	0.017059	0.034767	0.043735	0.154356	0.114739	0.026458	0.056264	0.042117
# new stud. enrolled	0.004881	0.012977	0.016970	0.010681	0.034473	0.085417	0.013150	0.021382	0.029534
% new stud. from top 10%	0.001420	0.000583	0.003962	0.002187	0.003087	0.002186	0.003644	0.004058	0.002496
% new stud. from top 25%	0.003905	0.003499	0.005203	0.003953	0.006174	0.003187	0.005862	0.006789	0.005512
# FT undergrad	0.022097	0.071737	0.081359	0.045249	0.093437	0.633795	0.041984	0.074447	0.099936
# PT undergrad	0.077119	0.269598	0.012239	0.010597	0.012246	0.427812	0.014417	0.000468	0.010295
in-state tuition	0.670906	0.253996	0.689424	0.679570	0.583054	0.202159	0.684742	0.686723	0.665337
out-of-state tuition	0.670906	0.761989	0.689424	0.679570	0.583054	0.404318	0.684742	0.686723	0.665337
room	0.143766	0.366559	0.121211	0.116065	0.146535	0.176206	0.188692	0.151001	0.150788
board	0.221860	0.328066	0.143679	0.213627	0.158472	0.295043	0.121992	0.098326	0.231069
add. fees	0.011537	0.004957	0.007095	0.008411	0.043014	0.026499	0.009506	0.025362	0.015391
estim. book costs	0.070995	0.072904	0.023651	0.042053	0.102904	0.068297	0.039608	0.039018	0.041597
estim. personal $	0.133116	0.169428	0.053215	0.092516	0.144066	0.200338	0.063373	0.093644	0.140388
% fac. w/PHD	0.006745	0.005686	0.004375	0.005299	0.005763	0.008742	0.006258	0.006399	0.007071
stud./fac. ratio	0.001056	0.001385	0.000828	0.000959	0.001595	0.000610	0.000998	0.001022	0.001383
Graduation rate	0.001331	0.005686	0.004257	0.003701	0.004734	0.003005	0.004278	0.004916	0.007799

17 rows × 471 columns

Understanding this data can help you implament PCAs becuase this allows the user to see where the data is heading where the most variance is at. This also allows the user to focus the data around the mean of the data getting more value for the information compared to being more spread out.

Does standardizing the data effect the output? How?

Standardizing the data does effect the output by changing the mean to 0 and the standard deviation to 1.

Follow the same process as above. This time standardize the data. Ensure your output is in dataframe format. Round the dataframe to the fourth decimal place.

In [71]:
```
pcs = PCA()
pcs.fit(unis3)
```

Out[71]:
```
PCA()
```

In [72]:
```
pcsSummary = pd.DataFrame({'Standard Deviation': np.sqrt(pcs.explained_variance_),
                           'Proportion of variance': pcs.explained_variance_ratio_,
                          'Cumulative proportion': np.cumsum(pcs.explained_variance_rat
pcsSummary = pcsSummary.transpose()
pcsSummary.columns = ['PC1', 'PC2', 'PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10',
```

In [73]:
```
# you can use this statement to print the output
pcsSummary_df.columns = ['PC{}'.format(i) for i in range(1, len(pcsSummary_df.columns)
pcsSummary.round(4)
```

Out[73]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	P
Standard Deviation	7430.9140	5987.9890	1854.6412	1192.5293	967.4279	679.6527	596.9761	580.6299	417.6
Proportion of variance	0.5614	0.3645	0.0350	0.0145	0.0095	0.0047	0.0036	0.0034	0.0
Cumulative proportion	0.5614	0.9259	0.9609	0.9753	0.9848	0.9895	0.9932	0.9966	0.9

In [111…
```
pcsComponents_df = pd.DataFrame(pcs.components_.transpose(),
                                columns=pcsSummary_df.columns,
                                index=unis.iloc[:,3:].columns)
```

In [110…
```
pcsComponents_df.iloc[:]
```

Out[110]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	
# appli. rec'd	0.271883	0.551183	0.664458	0.129476	-0.034246	0.370333	-0.120305	-0.097471	-0.0
# appl. accepted	0.194107	0.321299	0.190957	-0.008357	-0.076674	-0.813924	0.353520	0.103440	0.0
# new stud. enrolled	0.084730	0.101590	-0.087451	-0.055253	-0.036068	-0.081429	0.019293	-0.039063	0.0
% new stud. from top 10%	-0.000898	0.001732	0.000136	-0.001906	0.001236	0.009145	-0.003462	-0.002851	-0.0
% new stud. from top 25%	-0.000811	0.001925	0.000040	-0.002352	0.001009	0.007166	-0.003192	-0.002603	-0.0
# FT undergrad	0.458121	0.492263	-0.635303	-0.284582	-0.080402	0.129196	-0.127077	0.011595	-0.0
# PT undergrad	0.108253	0.073410	-0.285353	0.942562	-0.051743	-0.039789	-0.018146	-0.073893	-0.0
in-state tuition	-0.670187	0.382489	-0.082787	-0.016972	-0.621759	0.000517	-0.060641	0.006407	-0.0
out-of-state tuition	-0.454535	0.428685	-0.129410	0.018657	0.748634	0.010286	0.141481	-0.091839	-0.0
room	-0.033420	0.055584	0.040113	0.065120	0.115354	-0.050083	-0.314426	0.873995	-0.3
board	-0.034236	0.040897	-0.008232	0.067313	0.006301	0.067317	-0.145646	0.276553	0.9
add. fees	0.013209	0.008746	0.032868	-0.012755	0.103097	-0.024987	-0.043534	0.068703	0.0
estim. book costs	-0.000058	0.003291	0.000316	0.010795	-0.005223	0.034097	0.011155	0.064045	0.0
estim. personal $	0.037557	0.001185	-0.054659	0.031666	-0.106952	0.407928	0.835394	0.339141	0.0
% fac. w/PHD	-0.000205	0.001564	-0.000995	-0.000055	0.004822	0.000925	-0.001184	-0.000218	0.0
stud./fac. ratio	0.000295	-0.000159	0.000025	-0.000135	-0.000201	-0.000748	-0.000397	-0.000359	0.0
Graduation rate	-0.001072	0.001397	0.000920	-0.002172	0.001129	0.001077	-0.001994	-0.000387	0.0

After standardizing, discuss the standard deviation, proportion of variance, and cumulative proportion.

The first 2 PCs yield the highest standard deviation, proportion of variance, and cumulative proportion. Leading me to beleive that since they cover 92.59% of the data that they are what I would continue forward with as using PC3 would increase the amount of information gained by

a significant amount. These results also alligned with the standard deviation, proportion of variance, and cumulative proportion as the normalized data set at the start of this assignment.

Discussion

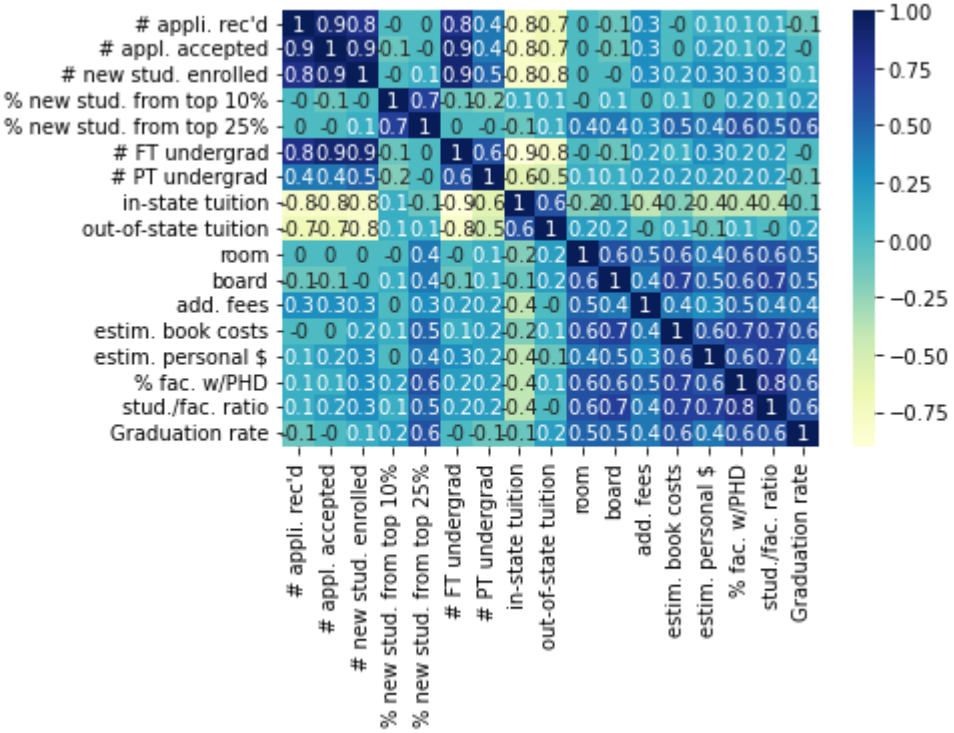
Should standardized data be used in this problem? If so, why or why not? How do you know which PCA implementation was more effective?

I feel that standardized data should be used for this problem because normalization is effected by outliers where standardized is not. Leading to more accurate findings. For my findings both PCA implementations lead to the same standard deviation, proportion of variance and cumulative proportion. That being said, if there were greater outliers that could change the results I would stick with the standardized data PCA implementation.

What "Topics" would you assign to each individual PC? You should be able to identify "Topics" that align with each PC. Each principle component may have a topic or two associated with it.

In [112…

```
import seaborn as sns
dp = sns.heatmap(unis5.corr().round(1), cmap="YlGnBu", annot=True)
```



-'PC1': # appli rec'd, # appl accepted
-'PC2': # new student enrolled, # FT undergrad
-'PC3': stud fac ratio, % fac w PHD
-'PC4': % new stud top 10%, % new stud top 25%
-'PC5': #PT undergrad, #FT undergrad

```
-'PC6': add fees, room
-'PC7': graduation rate, out of state tuition
-'PC8': estim book costs, # appl accepted
-'PC9': estim personal money, new stud top 10%
-'PC10': board, in state tuition
-'PC11': room, instate tuition
-'PC12': in state tuition, add fees
-'PC13': out of state tuition, #PT undergrad
-'PC14': #PT undergrad, in state tuition
-'PC15': out of state tution, new student enrolled
-'PC16': out of state tuition, #FT undergrad
-'PC17': instate tuition, #FT undergrad
```

Submission Instructions

1. Print your Notebook file as a PDF by completing the following: a. File>Print Preview b. Right-click>Print>Destination>Print as PDF
2. Save the PDF file as YourLastNameAssignment02.
3. Submit the PDF file to the dropbox “Assignment 02”.

Supplemental Resources

Use the following to help if you get stuck:

1. Google the error you may be receiving. At a higher level, google what you are expected to do. For example: How do I subtract variables in Python?
2. Navigate through the slides and use the find feature to locate keywords.
3. Rewatch a lecture video or two.
4. Check out YouTube. This is a great resource for students.
5. Post in the discussion board. Remember, I receive emails of postings and other students will as well. If you are not receiving emails, subscribe to the assignment questions discussion board.

In [ ]: