In [55]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [56]:
```
doc = pd.read_csv('Pro2_ mostrcntdata.csv')
```

In [57]:
```
doc.shape
```

Out[57]:
```
(7058, 1977)
```

In [58]:
```
doc.columns
```

Out[58]:
```
Index(['UNITID', 'OPEID', 'OPEID6', 'INSTNM', 'CITY', 'STABBR', 'ZIP',
       'ACCREDAGENCY', 'INSTURL', 'NPCURL',
       ...
       'OMAWDP8_NOTFIRSTTIME_POOLED_SUPP', 'OMENRUP_NOTFIRSTTIME_POOLED_SUPP',
       'OMENRYP_FULLTIME_POOLED_SUPP', 'OMENRAP_FULLTIME_POOLED_SUPP',
       'OMAWDP8_FULLTIME_POOLED_SUPP', 'OMENRUP_FULLTIME_POOLED_SUPP',
       'OMENRYP_PARTTIME_POOLED_SUPP', 'OMENRAP_PARTTIME_POOLED_SUPP',
       'OMAWDP8_PARTTIME_POOLED_SUPP', 'OMENRUP_PARTTIME_POOLED_SUPP'],
      dtype='object', length=1977)
```

In [59]:
```
doc
```

3	100706	105500	1055	University of Alabama in Huntsville	Huntsville	AL	35899	Southern Association of Colleges and Schools C...
4	100724	100500	1005	Alabama State University	Montgomery	AL	36104-0271	Southern Association of Colleges and Schools C...
5	100751	105100	1051	The University of Alabama	Tuscaloosa	AL	35487-0166	Southern Association of Colleges and Schools C...
6	100760	100700	1007	Central Alabama Community College	Alexander City	AL	35010	Southern Association of Colleges and Schools C...
7	100812	100800	1008	Athens State	Athens	AL	35611	Southern Association of

In [60]:
```
print(doc.columns.values)
```

```
['UNITID' 'OPEID' 'OPEID6' ... 'OMENRAP_PARTTIME_POOLED_SUPP'
 'OMAWDP8_PARTTIME_POOLED_SUPP' 'OMENRUP_PARTTIME_POOLED_SUPP']
```

In [61]:
```
doc.dtypes
```

Out[61]:
```
UNITID                      int64
OPEID                       int64
OPEID6                      int64
INSTNM                     object
CITY                       object
STABBR                     object
ZIP                        object
ACCREDAGENCY               object
INSTURL                    object
NPCURL                     object
SCH_DEG                   float64
HCM2                        int64
MAIN                        int64
NUMBRANCH                   int64
PREDDEG                     int64
HIGHDEG                     int64
CONTROL                     int64
ST_FIPS                     int64
REGION                      int64
LOCALE                    float64
LOCALE2                   float64
LATITUDE                  float64
LONGITUDE                 float64
CCBASIC                   float64
CCUGPROF                  float64
CCSIZSET                  float64
HBCU                      float64
PBI                       float64
ANNHI                     float64
TRIBAL                    float64
                           ...
OMAWDP8_NOTFIRSTTIME      float64
OMENRUP_NOTFIRSTTIME      float64
OMENRYP_FULLTIME          float64
OMENRAP_FULLTIME          float64
OMAWDP8_FULLTIME          float64
OMENRUP_FULLTIME          float64
OMENRYP_PARTTIME          float64
OMENRAP_PARTTIME          float64
OMAWDP8_PARTTIME          float64
```

```
OMENRUP_PARTTIME                    float64
OMENRYP_ALL_POOLED_SUPP              object
OMENRAP_ALL_POOLED_SUPP              object
OMAWDP8_ALL_POOLED_SUPP              object
OMENRUP_ALL_POOLED_SUPP              object
OMENRYP_FIRSTTIME_POOLED_SUPP        object
OMENRAP_FIRSTTIME_POOLED_SUPP        object
OMAWDP8_FIRSTTIME_POOLED_SUPP        object
OMENRUP_FIRSTTIME_POOLED_SUPP        object
OMENRYP_NOTFIRSTTIME_POOLED_SUPP     object
OMENRAP_NOTFIRSTTIME_POOLED_SUPP     object
OMAWDP8_NOTFIRSTTIME_POOLED_SUPP     object
OMENRUP_NOTFIRSTTIME_POOLED_SUPP     object
OMENRYP_FULLTIME_POOLED_SUPP         object
OMENRAP_FULLTIME_POOLED_SUPP         object
OMAWDP8_FULLTIME_POOLED_SUPP         object
OMENRUP_FULLTIME_POOLED_SUPP         object
OMENRYP_PARTTIME_POOLED_SUPP         object
OMENRAP_PARTTIME_POOLED_SUPP         object
OMAWDP8_PARTTIME_POOLED_SUPP         object
OMENRUP_PARTTIME_POOLED_SUPP         object
Length: 1977, dtype: object
```

In [62]:
```
doc.groupby
```

Out[62]:
```
<bound method NDFrame.groupby of        UNITID    OPEID  OPEID6  \
0      100654   100200    1002
1      100663   105200    1052
2      100690  2503400   25034
3      100706   105500    1055
4      100724   100500    1005
5      100751   105100    1051
6      100760   100700    1007
7      100812   100800    1008
8      100830   831000    8310
9      100858   100900    1009
10     100937   101200    1012
11     101028  1218200   12182
12     101073  1055400   10554
13     101116  1303906   13039
14     101143   101500    1015
15     101161   106000    1060
16     101189   100300    1003
17     101240   101700    1017
```

Q1> What is the most costly college? What is the cheapest?

In [63]:
```
# Combine annual cost and program cost to have one column for all institues tutuion
doc['COST'] = doc['COSTT4_A']
doc['COST'] = doc['COST'].fillna(doc['COSTT4_P'])
```

In [64]:
```
print ('The most expensive school is:')
print (doc.loc[doc['COST'].idxmax()].INSTNM)
print ('For the cost of:')
print (doc.loc[doc['COST'].idxmax()].COST)
```

```
The most expensive school is:
L3 Commercial Training Solutions Airline Academy
For the cost of:
105745.0
```

Q2> What is the average cost for college for colleges in different parts of the US?

In [65]:
```
doc.loc[doc['COST'] == 0.0]
```

Out[65]:

	UNITID	OPEID	OPEID6	INSTNM	CITY	STABBR	ZIP	ACCREDAGENCY	INSTURL	NPCURL	..
6589	490805	182700	1827	Purdue University Northwest	Hammond	IN	46323	Higher Learning Commission	www.pnw.edu	admissions.pnw.edu/financial-aid/net-price-cal...	..

1 rows × 1978 columns

In [66]:
```
tutionData = doc.dropna(subset=['COST'])
```

In [67]:
```
tutionData = round(tutionData.groupby(['STABBR'])['COST'].mean())
```

In [68]:
```
print ('The avarge tutition for each state')
print ('In descending order')
print  ('')
print (tutionData.sort_values(ascending=False))
```

```
The avarge tutition for each state
In descending order

STABBR
VT    39869.0
DC    36921.0
MA    35063.0
RI    34898.0
PA    29007.0
NH    28653.0
ME    28091.0
NY    27910.0
IN    26988.0
CT    26631.0
IA    25968.0
MD    25329.0
CA    25254.0
VA    25170.0
NJ    25149.0
MN    24887.0
WI    24553.0
DE    24394.0
NE    24390.0
SC    24323.0
OR    23797.0
FL    23676.0
GA    23428.0
OH    23277.0
IL    23245.0
CO    23154.0
NC    22907.0
TN    22750.0
MI    22736.0
NV    22694.0
MO    22621.0
KY    22480.0
WA    22177.0
```

```
KS    21793.0
TX    21573.0
AZ    21327.0
SD    21214.0
LA    20874.0
HI    20745.0
AL    20662.0
AK    20203.0
UT    20058.0
ID    19609.0
MS    19143.0
AR    18277.0
WV    18102.0
NM    18100.0
OK    17511.0
ND    17174.0
MT    17120.0
VI    16786.0
WY    14714.0
GU    12339.0
PR    11653.0
FM     9554.0
MH     8750.0
MP     8734.0
AS     7400.0
PW     6085.0
Name: COST, dtype: float64
```

Q3> What is the average cost for college for religious vs. secular institutions?

In [70]:

```
# create new data frame for religious tuition
# drop all rows where religious is NaN
religiousTuition = doc.dropna(subset=['RELAFFIL'])
# calculate the avarage tutition for religious schools
print ('The avarage tutition for religious schools is:')
print (round(religiousTuition.COST.mean()),'$')
```

```
The avarage tutition for religious schools is:
37389 $
```

In [73]:
```
# create new data frame for secular tuition
# drop all rows where religious is NaN
secularTuition = doc.loc[pd.isnull(doc).any(1),:]
# calculate the avarage annual tutition for religious schools
print ('The avarage annual tutition for secullar schools is:')
print (round(secularTuition.COST.mean()),'$')
```

```
The avarage annual tutition for secullar schools is:
23869 $
```

Q4> What percent of colleges have an open admission policy?

In [75]:
```
# From the Data Dictionary:
# Open admissions policy indicator:
# 1 Yes
# 2 No

# create a data frame with only open addmission schools:

# drop rows with NaN value
openAdmission = doc.dropna(subset=['OPENADMP'])
#drop rows with non-open admision
openAdmission = openAdmission[openAdmission.OPENADMP !=2]

# calculate the percentage of open admission out of all schools
print('There are', len(openAdmission), 'schools with open admission')
print('which make', round((len(openAdmission))/(len(doc))*100),
      '% of all', len(doc), 'schools')
```

```
There are 4063 schools with open admission
which make 58 % of all 7058 schools
```

Q5> What is the correlation (scatterplot) between admission rates and college cost?

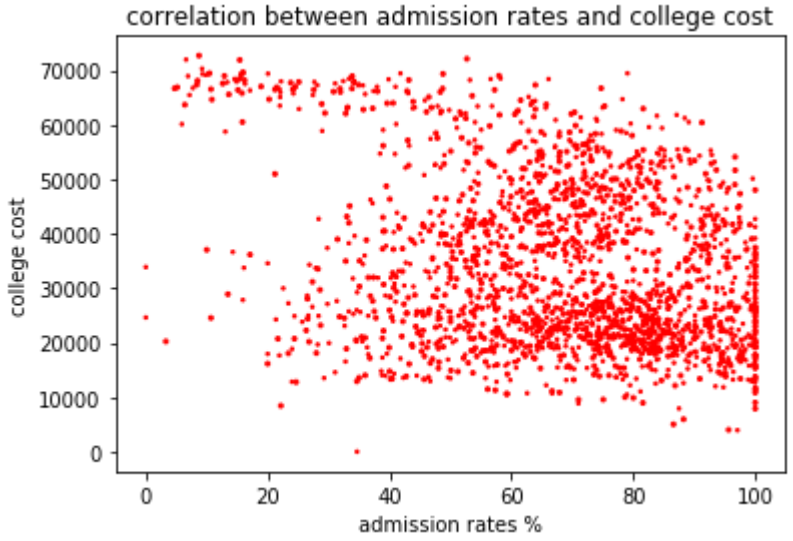
In [78]:
```
# 5.1. Annual costs
AdRt_COST = doc[['ADM_RATE', 'COST']].copy()
```

In [77]:
```
AdRt_COST.corr()
```

Out[77]:

	ADM_RATE	COST
ADM_RATE	1.000000	-0.301969
COST	-0.301969	1.000000

In [81]:
```
s = (4,2)
plt.scatter(doc.ADM_RATE*100, doc.COST, s, color='r')
plt.title("Correlation Between Admission Rates and College Cost")
plt.xlabel("admission rates %")
plt.ylabel("college cost")
plt.show()
```



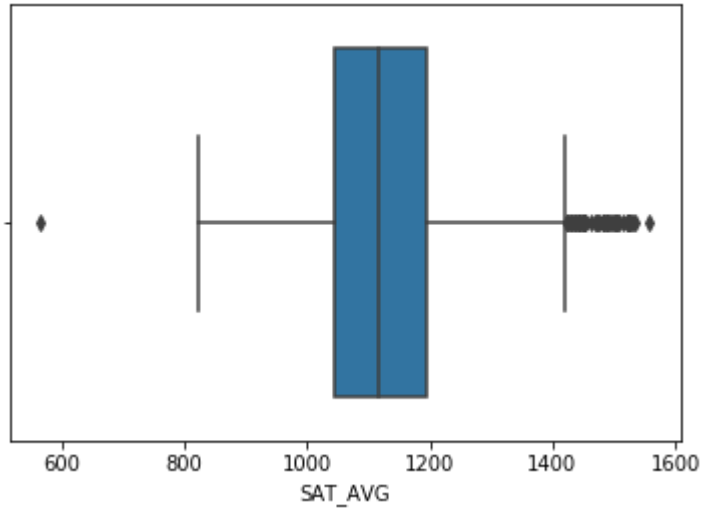
Q6 > What is the correlation between SAT scores and admission rates? Are there any outliers?

In [82]:
```
# Observation: There is one outlier just below 600
doc.SAT_AVG.sort_values(ascending=True).head(2)
```

Out[82]:
```
825     564.0
2513    822.0
Name: SAT_AVG, dtype: float64
```

In [83]:
```
sns.boxplot(doc.SAT_AVG)
```

Out[83]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x27295942f60>
```



In [85]:
```
# Observation: There are outliers close to 0.0
doc.ADM_RATE.sort_values(ascending=True).head(5)
```

Out[85]:
```
6065    0.000
5188    0.000
2960    0.000
6610    0.000
2964    0.033
Name: ADM_RATE, dtype: float64
```

Q6 >> What colleges have the highest and lowest family income averages?

How does that correlate with college costs?

In [86]:
```
# df['COST'] = df['COST'].fillna(df['COSTT4_P'])
doc['FAMINC'] = doc['FAMINC'].replace('PrivacySuppressed', np.nan)
doc['FAMINC'] = doc.FAMINC.astype(float)
```

In [96]:
```
print('The maximal family income registred is:  ', round(doc['FAMINC'].max()),'$') # max family income
print('Institution:',doc.iloc[doc['FAMINC'].idxmax].INSTNM,',', doc.iloc[doc
                                                              ['FAMINC'].idxmax].STABBR)
```

```
The maximal family income registred is:   174263.0 $
Institution: Jewish Theological Seminary of America , NY
```

In [91]:
```
print('The maximal family income registred is:  ', round(doc['FAMINC'].min()),'$')
print('Institution:',doc.iloc[doc['FAMINC'].idxmin].INSTNM,',', doc.iloc[doc['FAMINC'].idxmin].STABBR)
```

```
The maximal family income registred is:   321.0 $
Institution: J F Ingram State Technical College , AL
```

In [95]:
```
famIncm_Cost = doc[['FAMINC', 'COST']].copy() # corelation between max income and college cost
```

In [94]:
```
famIncm_Cost.corr()
```

Out[94]:

	FAMINC	COST
FAMINC	1.0000	0.6758
COST	0.6758	1.0000

In [ ]:
```
```