SUMMER BOOTCAMP 2024

Data Science Project
KESHAV BISHT
Education Post 12th Data Analysis

S No.	Topic	Page No.
1	Cover Page	1
2	Index	2
3	List of Tables	3
4	List of Figures	3
5	Problem Statement	4
6	Data Exploration	5-7
7	Basic EDA	8-16
8	Application and Enrollment Analysis	17-21
9	Academic Excellence	22-24
10	Student Demographics	25-27
11	Cost of Spendings	28-31
12	Faculty Qualification	32-34
13	Student-Faculty Interaction	35
14	Alumini Engagement	36-37
15	Graduation Rates	38-40
16	Overall Insights	41-42

PROBLEM STATEMENT / OBJECTIVE

The objective of this analysis is to gain insights into the characteristics of colleges and answer key questions related to the educational landscape. By understanding the data, we aim to inform strategies for improving the quality of education and enhancing the overall college experience. The analysis will provide valuable insights and recommendations for stakeholders in the education sector.

IMPORTING THE LIBRARIES

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING THE DATA SET

In [2]:
```
df = pd.read_csv("/Users/keshavbisht/Downloads//1-Education_Post_12th_Standard_New.csv")
```

BASIC EXPLORATION

1. FIRST FIVE ROWS

In [15]:
```
df.head()
```

Out[15]:

	Names	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	Abilene Christian University	1660.0	1232	721.0	23.0	52	2885	537	7440	3300	450	2200.0	70	78	18.1	12	7041	60
1	Adelphi University	2186.0	1924	512.0	16.0	29	2683	1227	12280	6450	750	1500.0	29	30	?	16	10527	56
2	Adrian College	1428.0	1097	336.0	22.0	50	1036	99	11250	3750	400	1165.0	53	66	12.9	30	8735	54
3	Agnes Scott College	417.0	349	NaN	60.0	89	510	63	12960	5450	450	875.0	92	97	7.7	37	19016	59
4	Alaska Pacific University	193.0	146	55.0	16.0	44	249	869	7560	4120	800	1500.0	76	72	11.9	2	10922	15

2. INFO

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776
Data columns (total 18 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Names        777 non-null    object
 1   Apps         775 non-null    float64
 2   Accept       777 non-null    int64
 3   Enroll       775 non-null    float64
 4   Top10perc    773 non-null    float64
 5   Top25perc    777 non-null    int64
 6   F.Undergrad  777 non-null    int64
 7   P.Undergrad  777 non-null    int64
 8   Outstate     777 non-null    int64
 9   Room.Board   777 non-null    int64
 10  Books        777 non-null    int64
 11  Personal     774 non-null    float64
 12  PhD          777 non-null    int64
 13  Terminal     777 non-null    int64
 14  S.F.Ratio    777 non-null    object
 15  perc.alumni  777 non-null    int64
 16  Expend       777 non-null    int64
 17  Grad.Rate    777 non-null    int64
dtypes: float64(4), int64(12), object(2)
memory usage: 109.4+ KB
```

3. LAST 5 ROWS

In [17]:
```
df.tail()
```

Out[17]:

	Names	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
772	Worcester State College	2197.0	1515	543.0	4.0	26	3089	2029	6797	3900	500	1200.0	60	60	21	14	4469	40
773	Xavier University	1959.0	1805	695.0	24.0	47	2849	1107	11520	4960	600	1250.0	73	75	13.3	31	9189	83
774	Xavier University of Louisiana	2097.0	1915	695.0	34.0	61	2793	166	6900	4200	617	781.0	67	75	14.4	20	8323	49
775	Yale University	10705.0	2453	1317.0	95.0	99	5217	83	19840	6510	630	2115.0	96	96	5.8	49	40386	99
776	York College of Pennsylvania	2989.0	1855	691.0	28.0	63	2988	1726	4990	3560	500	1250.0	75	75	18.1	28	4509	99

4. SHAPE

In [19]:
```
df.shape
```

Out[19]: (777, 18)

5. STATISTICAL SUMMARY

In [20]:
```
df.describe()
```

Out[20]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	perc.alumni	
count	775.000000	777.000000	775.000000	773.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	774.000000	777.000000	777.000000	777.000000	77
mean	3007.592258	2018.804376	780.961290	27.620957	55.796654	3699.907336	855.298584	10440.669241	4357.526384	547.875161	1601.507752	72.660232	79.702703	22.743887	966
std	3873.414660	2451.113971	930.077779	17.645470	19.804778	4850.420531	1522.431887	4023.016484	1096.696416	167.426237	7369.594038	16.328155	14.722359	12.391801	522
min	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	1.000000	2340.000000	1780.000000	0.000000	50.000000	8.000000	24.000000	0.000000	318
25%	778.000000	604.000000	242.500000	15.000000	41.000000	992.000000	95.000000	7320.000000	3597.000000	465.000000	855.000000	62.000000	71.000000	13.000000	675
50%	1561.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	353.000000	9990.000000	4200.000000	500.000000	1200.000000	75.000000	82.000000	21.000000	837
75%	3635.000000	2424.000000	902.500000	35.000000	69.000000	4005.000000	967.000000	12925.000000	5050.000000	600.000000	1687.500000	85.000000	92.000000	31.000000	1083
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	21836.000000	21700.000000	8124.000000	2340.000000	205500.000000	103.000000	100.000000	64.000000	5623

6. NULL VALUES

In [21]:
```
df.isnull()
```

Out[21]:

	Names	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
772	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
773	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
774	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
775	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
776	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

777 rows × 18 columns

7. DUPLICATE VALUES

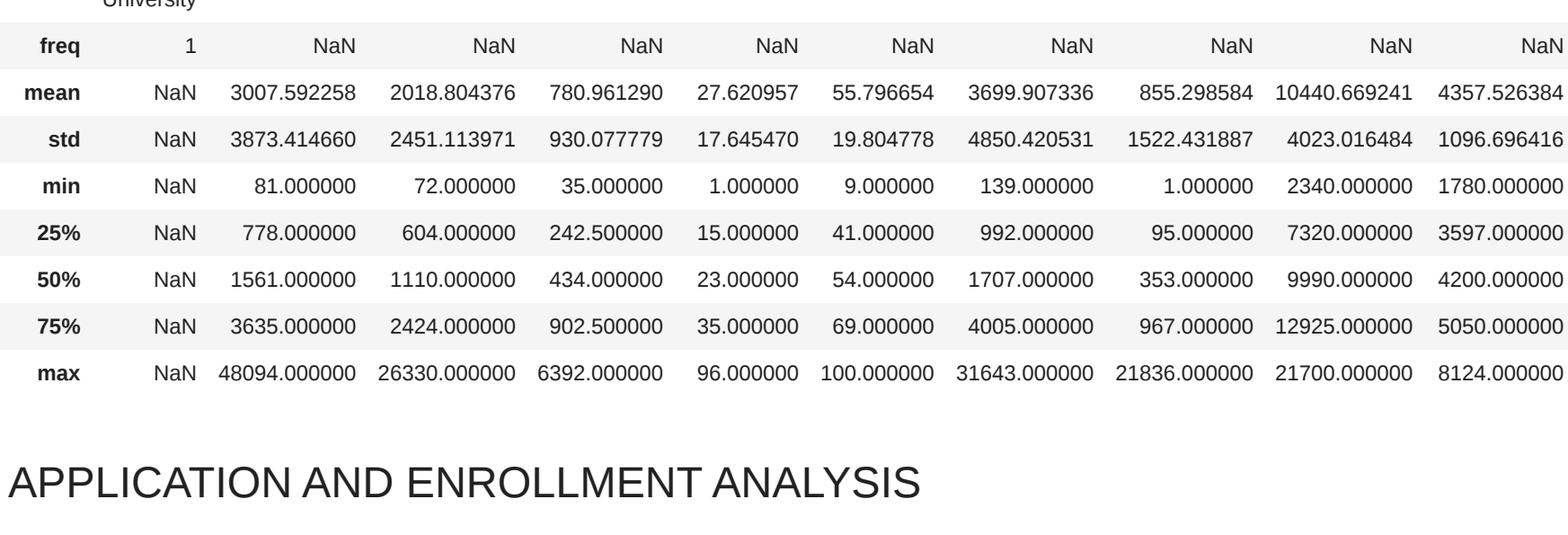
In [22]:
```
df.duplicated()
```

Out[22]:
```
0      False
1      False
2      False
3      False
4      False
       ...
772    False
773    False
774    False
775    False
776    False
Length: 777, dtype: bool
```

8. OUTLIERS AND THEIR AUTHTICITY

In [23]:
```
plt.figure(figsize =(10,3))
sns.boxplot(data = df)
```

Out[23]: <Axes: >



9 . ANOMALIES OR WRONG ENTRY

In [5]:
```
df.describe(include='all')
```

Out[5]:

	Names	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ra
count	777	775.000000	777.000000	775.000000	773.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	774.000000	777.000000	777.000000	7
unique	777	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
top	Abilene Christian University	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	NaN	3007.592258	2018.804376	780.961290	27.620957	55.796654	3699.907336	855.298584	10440.669241	4357.526384	547.875161	1601.507752	72.660232	79.702703	N
std	NaN	3873.414660	2451.113971	930.077779	17.645470	19.804778	4850.420531	1522.431887	4023.016484	1096.696416	167.426237	7369.594038	16.328155	14.722359	N
min	NaN	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	1.000000	2340.000000	1780.000000	0.000000	50.000000	8.000000	24.000000	N
25%	NaN	778.000000	604.000000	242.500000	15.000000	41.000000	992.000000	95.000000	7320.000000	3597.000000	465.000000	855.000000	62.000000	71.000000	N
50%	NaN	1561.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	353.000000	9990.000000	4200.000000	500.000000	1200.000000	75.000000	82.000000	N
75%	NaN	3635.000000	2424.000000	902.500000	35.000000	69.000000	4005.000000	967.000000	12925.000000	5050.000000	600.000000	1687.500000	85.000000	92.000000	N
max	NaN	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	21836.000000	21700.000000	8124.000000	2340.000000	205500.000000	103.000000	100.000000	N

APPLICATION AND ENROLLMENT ANALYSIS

1. WHAT IS THE AVERAGE NUMBER OF APPLICATIONS RECEIVED BY COLLEGES?

In [7]:
```
avg_apps = df['Apps'].mean()
avg_apps
```

Out[7]: 3007.592258064516

2. WHAT PERCENTAGE OF APPLICATIONS ARE ACCEPTED ON AVERAGE ACROSS ALL COLLEGES?

In [8]:
```
acceptance_rate = (df['Accept'] / df['Apps']).mean() * 100
acceptance_rate
```

Out[8]: 74.6620740888846

3. What is the average enrollment rate (number of students enrolled divided by number of applications accepted)?

In [9]:
```
enrollment_rate = (df['Enroll'] / df['Accept']).mean() * 100
enrollment_rate
```

Out[9]: 41.19383221625094

4. Which college has the highest number of applications received?

In [11]:
```
highest_apps_college = df[df['Apps'] == df['Apps'].max()]['Names'].values[0]
highest_apps_college
```

Out[11]: 'Rutgers at New Brunswick'

5. Average percentage of new students from the top 10% of their high school class

In [12]:
```
avg_top10perc = df['Top10perc'].mean()
avg_top10perc
```

Out[12]: 27.620957309184995

ACADEMIC EXCELLENCE

1. Average percentage of new students from the top 10% of their high school class

In [15]:
```
avg_top10perc = df['Top10perc'].mean()
avg_top10perc
```

Out[15]: 27.620957309184995

2. Average percentage of new students from the top 25% of their high school class

In [16]:
```
avg_top25perc = df['Top25perc'].mean()
avg_top25perc
```

Out[16]: 55.7966537966538

3. Correlation between the percentage of students from the top 10% and the top 25% of their high school class

In [17]:
```
correlation_top10_top25 = df['Top10perc'].corr(df['Top25perc'])
correlation_top10_top25
```

Out[17]: 0.8916010734346558

STUDENT DEMOGRAPHICS

1. What is the average number of full-time undergraduate students per college?

In [18]:
```
avg_full_time_undergrad = df['F.Undergrad'].mean()
avg_full_time_undergrad
```

Out[18]: 3699.907335907336

2. Average number of part-time undergraduate students per college

In [19]:
```
avg_part_time_undergrad = df['P.Undergrad'].mean()
avg_part_time_undergrad
```

Out[19]: 855.2985842985843

3. College with the highest number of out-of-state students

In [20]:
```
highest_outstate_college = df[df['Outstate'] == df['Outstate'].max()]['Names'].values[0]
highest_outstate_college
```

Out[20]: 'Bennington College'

COST OF SPENDING

1. What is the average cost of room and board across all colleges?

In [21]:
```
avg_room_board = df['Room.Board'].mean()
avg_room_board
```

Out[21]: 4357.526383526383

2. What is the average estimated book cost for a student?

In [22]:
```
avg_book_cost = df['Books'].mean()
avg_book_cost
```

Out[22]: 547.8751608751609

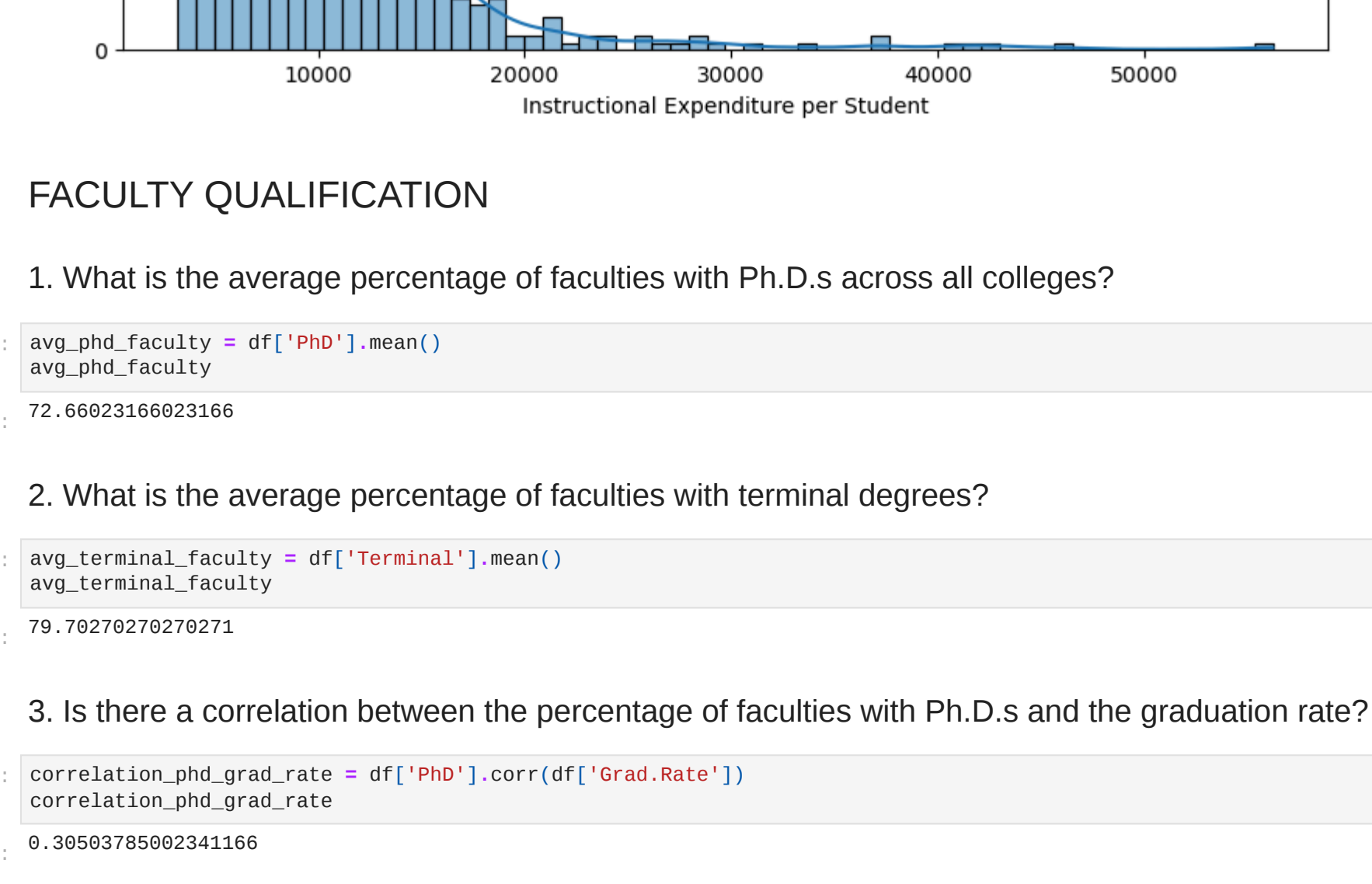
3. What is the average estimated personal spending for a student?

In [23]:
```
avg_personal_spending = df['Personal'].mean()
avg_personal_spending
```

Out[23]: 1601.5077519379845

4. How does the instructional expenditure per student vary across colleges?

In [24]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['Expend'], kde=True)
plt.title('Variation of Instructional Expenditure per Student Across Colleges')
plt.xlabel('Instructional Expenditure per Student')
plt.ylabel('Frequency')
plt.show()
```



FACULTY QUALIFICATION

1. What is the average percentage of faculties with Ph.D.s across all colleges?

In [25]:
```
avg_phd_faculty = df['PhD'].mean()
avg_phd_faculty
```

Out[25]: 72.66023166023166

2. What is the average percentage of faculties with terminal degrees?

In [26]:
```
avg_terminal_faculty = df['Terminal'].mean()
avg_terminal_faculty
```

Out[26]: 79.70270270270271

3. Is there a correlation between the percentage of faculties with Ph.D.s and the graduation rate?

In [27]:
```
correlation_phd_grad_rate = df['PhD'].corr(df['Grad.Rate'])
correlation_phd_grad_rate
```

Out[27]: 0.30503785002341166

STUDENT - FACULTY INTERACTION

1. Which college has the lowest student/faculty ratio?

In [42]:
```
lowest_student_faculty_ratio_college = df[df['S.F.Ratio'] == df['S.F.Ratio'].min()]['Names'].values[0]
lowest_student_faculty_ratio_college
```

Out[42]: 'Saint Olaf College'

ALUMNI ENGAGEMENT

1. What is the average percentage of alumni who donate across all colleges?

In [29]:
```
avg_alumni_donate = df['perc.alumni'].mean()
avg_alumni_donate
```

Out[29]: 22.743886743886744

2. Is there a correlation between the percentage of alumni who donate and the graduation rate?

In [30]:
```
correlation_alumni_donate_grad_rate = df['perc.alumni'].corr(df['Grad.Rate'])
correlation_alumni_donate_grad_rate
```

Out[30]: 0.4908975622847202

GRADUATION RATES

1. What is the average graduation rate across all colleges?

In [31]:
```
avg_grad_rate = df['Grad.Rate'].mean()
avg_grad_rate
```

Out[31]: 65.46332046332046

2. Which college has the highest graduation rate?

In [32]:
```
highest_grad_rate_college = df[df['Grad.Rate'] == df['Grad.Rate'].max()]['Names'].values[0]
highest_grad_rate_college
```

Out[32]: 'Cazenovia College'

3. Is there a correlation between the instructional expenditure per student and the graduation rate?

In [33]:
```
correlation_expend_grad_rate = df['Expend'].corr(df['Grad.Rate'])
correlation_expend_grad_rate
```

Out[33]: 0.3903426958593891

OVERALL INSIGHTS

1. Which factors (applications, acceptance rate, enrollment, academic excellence, costs, faculty qualifications, student/faculty ratio, alumni donations, expenditures) are most strongly associated with higher graduation rates?

In [48]:
```
correlation_matrix = df[['Apps', 'Accept', 'Enroll', 'Top10perc', 'Top25perc', 'Outstate', 'Room.Board',
                         'Books', 'Personal', 'PhD', 'Terminal', 'S.F.Ratio', 'perc.alumni', 'Expend', 'Grad.Rate']].corr()

correlation_with_grad_rate = correlation_matrix['Grad.Rate'].sort_values(ascending=False)
correlation_with_grad_rate
```

```
/var/folders/ck/kmw8dlf92gzbqwjfv8s3rgdr0000gn/T/ipykernel_65124/1794078562.py:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprec
ated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  'Books', 'Personal', 'PhD', 'Terminal', 'S.F.Ratio', 'perc.alumni', 'Expend', 'Grad.Rate']].corr()
```

Out[48]:
```
Grad.Rate      1.000000
Outstate       0.571290
Top10perc      0.492750
perc.alumni    0.490898
Top25perc      0.477281
Room.Board     0.424942
Expend         0.390343
PhD            0.305038
Terminal       0.289527
Apps           0.147149
Accept         0.067313
Personal       0.016145
Books          0.006532
Enroll        -0.022711
Name: Grad.Rate, dtype: float64
```

1. Out-of-state Tuition (Outstate): Strong positive correlation (0.571). Higher tuition often reflects better resources.

2. Top 10% of High School Class (Top10perc): Strong positive correlation (0.493). Attracting high-achieving students leads to higher graduation rates.

3. Alumni Donations (perc.alumni): Strong positive correlation (0.491). Alumni support often enhances resources and institutional quality.

4. Top 25% of High School Class (Top25perc): Strong positive correlation (0.477). High-achieving students contribute to higher graduation rates.

5. Room and Board Costs (Room.Board): Moderate positive correlation (0.425). Better living conditions support student success.

2. What recommendations can be made to colleges to improve their graduation rates based on the data analysis?

1. Recruit High-Achieving Students: Offer scholarships and honors programs.

2. Enhance Alumni Engagement: Strengthen alumni relations and encourage donations.

3. Improve Campus Living Conditions: Invest in quality housing and facilities.

4. Increase Instructional Spending: Allocate more resources to teaching and support services.

5. Improve Faculty Qualifications: Support faculty in obtaining higher degrees and professional development.