# Problem Description

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.

**Data Description**

• **Names**: Names of various university and colleges

• **Apps**: Number of applications received

• **Accept**: Number of applications accepted

• **Enroll**: Number of new students enrolled

• **Top10perc**: Percentage of new students from top 10% of Higher Secondary class

• **Top25perc**: Percentage of new students from top 25% of Higher Secondary class

• **F.Undergrad**: Number of full-time undergraduate students

• **P.Undergrad**: Number of part-time undergraduate students

• **Outstate**: Number of students for whom the particular college or university is Out-of-state tuition

• **Room.Board**: Cost of Room and board

• **Books**: Estimated book costs for a student

• **Personal**: Estimated personal spending for a student

• **PhD**: Percentage of faculties with Ph.D.’s

• **Terminal**: Percentage of faculties with terminal degree

• **S.F.Ratio**: Student/faculty ratio

• **perc.alumni**: Percentage of alumni who donate

• **Expend**: The Instructional expenditure per student

• **Grad.Rate**: Graduation rate

## 

## Importing the necessary libraries

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns

# Loading Dataset

df=pd.read\_csv(r"C:\Users\Desktop\All Projects\1-Education\_Post\_12th\_Standard\_New.csv")

# Basic Exploration

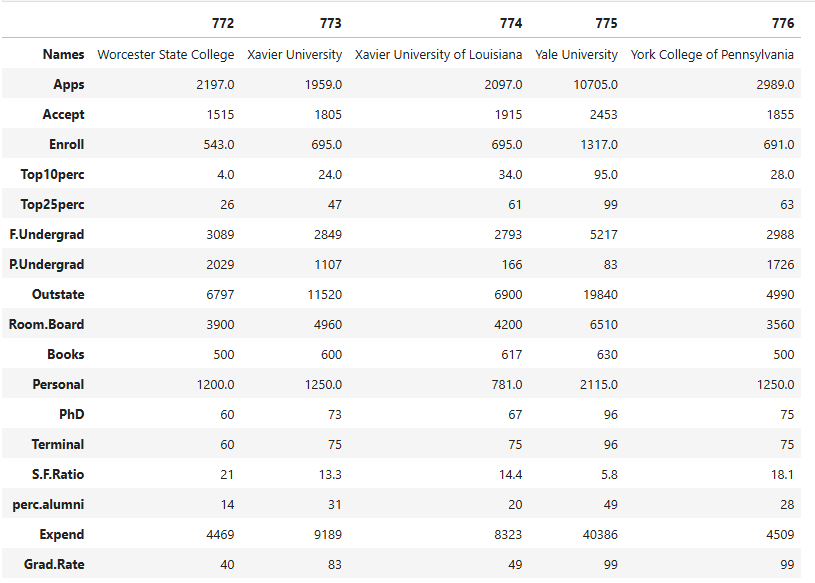
df.head()

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

* *Based on the above result we can observe that the following columns of 'Enroll' and 'S.F.Ratio' have some missing value and incorrect value as:*
* *'Enroll'* have a value of *'NaN'* on index *3*.
* *'S.F.Ratio'* have a incorrect value as *'?'* on index *'1'*.

Replacing **?** with **NaN**

df.replace('?',np.nan,inplace=True)



## displaying last 5 rows

df.tail()

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

df.shape

(777, 18)

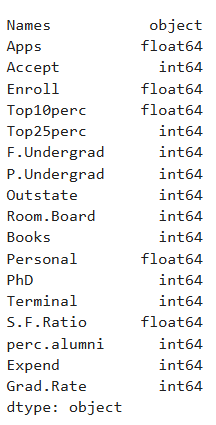
Dataset contains **777 rows** and **18 columns**

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 774 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

Data type of **S.F.Ratio** should be **Float**

df['S.F.Ratio']=df['S.F.Ratio'].astype(float)



df.isnull().sum()

Names 0  
Apps 2  
Accept 0  
Enroll 2  
Top10perc 4  
Top25perc 0  
F.Undergrad 0  
P.Undergrad 0  
Outstate 0  
Room.Board 0  
Books 0  
Personal 3  
PhD 0  
Terminal 0  
S.F.Ratio 3  
perc.alumni 0  
Expend 0  
Grad.Rate 0  
dtype: int64

df.duplicated().sum()

0

df.describe()

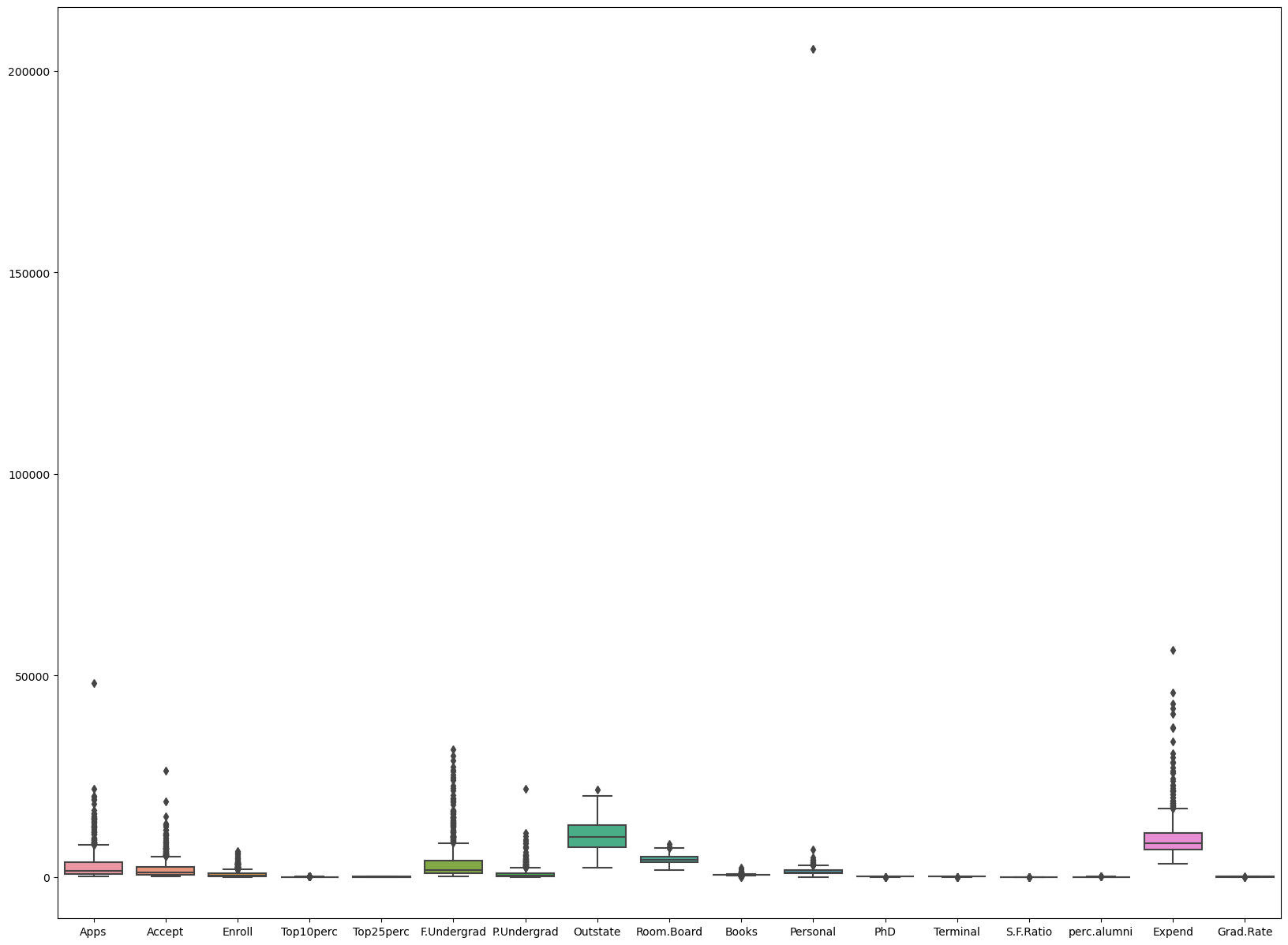
Apps Accept Enroll Top10perc Top25perc \  
count 775.000000 777.000000 775.000000 773.000000 777.000000   
mean 3007.592258 2018.804376 780.961290 27.620957 55.796654   
std 3873.414660 2451.113971 930.077779 17.645470 19.804778   
min 81.000000 72.000000 35.000000 1.000000 9.000000   
25% 778.000000 604.000000 242.500000 15.000000 41.000000   
50% 1561.000000 1110.000000 434.000000 23.000000 54.000000   
75% 3635.000000 2424.000000 902.500000 35.000000 69.000000   
max 48094.000000 26330.000000 6392.000000 96.000000 100.000000   
  
 F.Undergrad P.Undergrad Outstate Room.Board Books \  
count 777.000000 777.000000 777.000000 777.000000 777.000000   
mean 3699.907336 855.298584 10440.669241 4357.526384 547.875161   
std 4850.420531 1522.431887 4023.016484 1096.696416 167.426237   
min 139.000000 1.000000 2340.000000 1780.000000 0.000000   
25% 992.000000 95.000000 7320.000000 3597.000000 465.000000   
50% 1707.000000 353.000000 9990.000000 4200.000000 500.000000   
75% 4005.000000 967.000000 12925.000000 5050.000000 600.000000   
max 31643.000000 21836.000000 21700.000000 8124.000000 2340.000000   
  
 Personal PhD Terminal S.F.Ratio perc.alumni \  
count 774.000000 777.000000 777.000000 774.000000 777.000000   
mean 1601.507752 72.660232 79.702703 14.090310 22.743887   
std 7369.594038 16.328155 14.722359 3.949257 12.391801   
min 50.000000 8.000000 24.000000 2.500000 0.000000   
25% 855.000000 62.000000 71.000000 11.500000 13.000000   
50% 1200.000000 75.000000 82.000000 13.600000 21.000000   
75% 1687.500000 85.000000 92.000000 16.500000 31.000000   
max 205500.000000 103.000000 100.000000 39.800000 64.000000   
  
 Expend Grad.Rate   
count 777.000000 777.00000   
mean 9660.171171 65.46332   
std 5221.768440 17.17771   
min 3186.000000 10.00000   
25% 6751.000000 53.00000   
50% 8377.000000 65.00000   
75% 10830.000000 78.00000   
max 56233.000000 118.00000

min value of **Books** and **perc.alumni** is 0, which is need to be checked

**Box Plot**

plt.figure(figsize=(20,15))  
sns.boxplot(data = df)

<Axes: >



**Removing outliers**

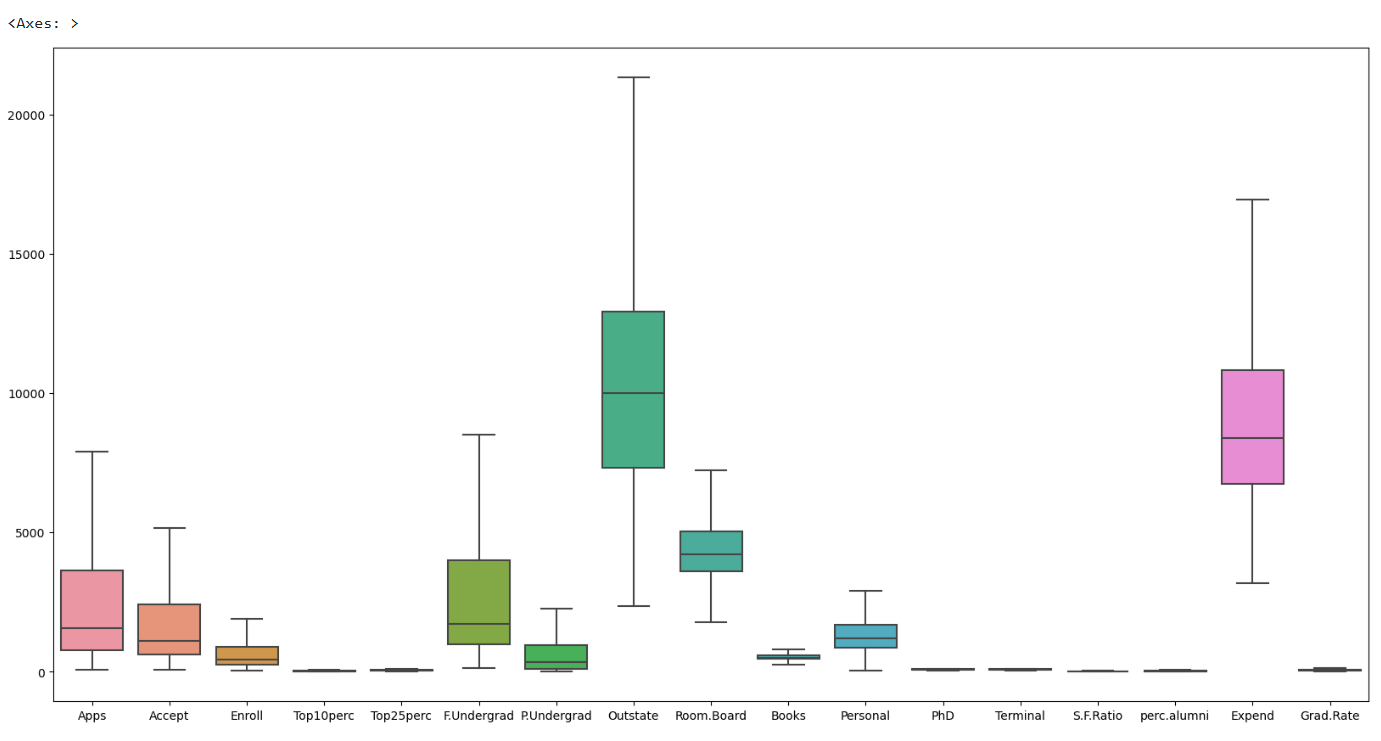
**Since outliers are present so replacing NULL values in Numerical columns with Median**

df[df.isnull().sum(axis=1)>0]

Names Apps Accept Enroll Top10perc \  
1 Adelphi University 2186.0 1924 512.0 16.0   
3 Agnes Scott College 417.0 349 NaN 60.0   
9 Alderson-Broaddus College NaN 498 172.0 21.0   
41 Bellarmine College NaN 707 308.0 39.0   
81 Campbell University 2087.0 1339 NaN 20.0   
102 Central Connecticut State University 4158.0 2532 902.0 NaN   
103 Central Missouri State University 4681.0 4101 1436.0 NaN   
128 College of Notre Dame 344.0 264 97.0 NaN   
129 College of Notre Dame of Maryland 457.0 356 177.0 NaN   
166 Dillard University 1998.0 1376 651.0 41.0   
175 Earlham College 1358.0 1006 274.0 35.0   
177 East Tennessee State University 3330.0 2730 1303.0 15.0   
241 Gwynedd Mercy College 380.0 237 104.0 30.0   
  
 Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books \  
1 29 2683 1227 12280 6450 750   
3 89 510 63 12960 5450 450   
9 44 799 78 10468 3380 660   
41 63 1198 605 8840 2950 750   
81 54 3191 1204 7550 2790 600   
102 24 6394 3881 5962 4444 500   
103 35 8094 1596 4620 3288 300   
128 42 500 331 12600 5520 630   
129 61 667 1983 11180 5620 600   
166 88 1539 45 6700 3650 500   
175 63 1028 13 15036 4056 600   
177 36 6706 2640 5800 3000 600   
241 56 716 1108 11000 5550 500   
  
 Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate   
1 1500.0 29 30 NaN 16 10527 56   
3 875.0 92 97 7.7 37 19016 59   
9 1800.0 40 41 11.5 15 8991 52   
41 1290.0 74 82 13.1 31 6668 84   
81 500.0 77 77 NaN 34 3739 63   
102 985.0 69 73 16.7 4 4900 49   
103 2250.0 69 80 19.7 4 5501 50   
128 2250.0 77 80 10.4 7 9773 43   
129 700.0 64 64 11.5 32 7477 75   
166 NaN 52 52 14.1 12 7566 61   
175 NaN 90 94 10.6 46 14634 78   
177 NaN 73 75 14.0 9 9825 42   
241 500.0 36 41 NaN 22 7483 96

median1=df['Apps'].median()  
median2=df['Enroll'].median()  
median3=df['Top10perc'].median()  
median4=df['Personal'].median()  
median5=df['S.F.Ratio'].median()  
  
df['Apps'].replace(np.nan,median1,inplace=True)  
df['Enroll'].replace(np.nan,median2,inplace=True)  
df['Top10perc'].replace(np.nan,median3,inplace=True)  
df['Personal'].replace(np.nan,median4,inplace=True)  
df['S.F.Ratio'].replace(np.nan,median5,inplace=True)

**Box Plot after outlier treatment**



# Application and Enrollment Analysis

#### Calculating the average number of Application recieved by colleges.

# Calculating the average number of Application recieved by colleges.  
avg = round(df['Apps'].mean(),3)  
print("average number of applications received by colleges:",avg)

average number of applications received by colleges: 3003.869

#### Calculating the percentage of applications accepted across all colleges.

# Calculating the percentage of applications accepted across all colleges.  
df['Accept\_Rate']=(df['Accept']/df['Apps'])\*100  
avg = round(df['Accept\_Rate'].mean(),2)  
print("percentage of applications are accepted on average across all colleges:",avg)

percentage of applications are accepted on average across all colleges: 74.57

#### Calculating the Average Enrollment Rate.

# Calculating the Average Enrollment Rate.  
df['Enroll\_rate']=(df['Enroll']/df['Accept'])\*100  
avg =round(df['Enroll\_rate'].mean(),2)  
print("Average Enrollment Rate:",avg)

Average Enrollment Rate: 41.29

# Calculating the Highest number of Applications Recieved.  
df['Accept'].max()

26330

df[df['Accept']==26330]

Names Apps Accept Enroll Top10perc Top25perc \  
483 Rutgers at New Brunswick 48094.0 26330 4520.0 36.0 79   
  
 F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD \  
483 21401 3712 7410 4748 690 2009.0 90   
  
 Terminal S.F.Ratio perc.alumni Expend Grad.Rate Accept\_Rate \  
483 95 19.5 19 10474 77 54.746954   
  
 Enroll\_rate   
483 17.16673

## Academic Excellence

#### Calculate the average percentage of new students from the top 10% of their higher secondary class

# Calculate the average percentage of new students from the top 10% of their higher secondary class  
avg\_top10perc = df['Top10perc'].mean()  
print(f"Average percentage of new students from the top 10% of their higher secondary class: {avg\_top10perc:.2f}")

Average percentage of new students from the top 10% of their higher secondary class: 27.60

#### Calculate the average percentage of new students from the top 25% of their higher secondary class

# Calculate the average percentage of new students from the top 25% of their higher secondary class  
avg\_top25perc = df['Top25perc'].mean()  
print(f"Average percentage of new students from the top 25% of their higher secondary class: {avg\_top25perc:.2f}")

Average percentage of new students from the top 25% of their higher secondary class: 55.80

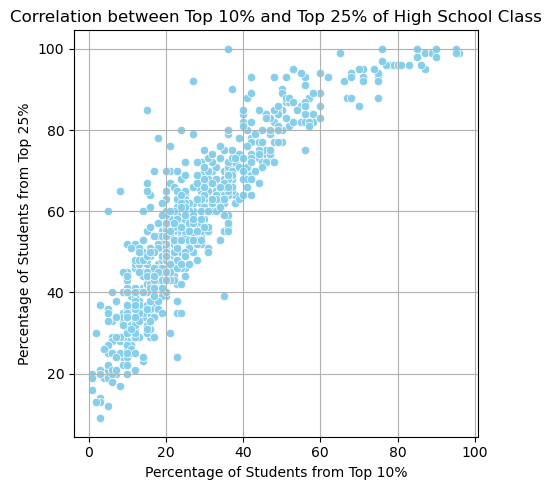
#### Compute the correlation coefficient between 'Top10perc' and 'Top25perc'

# Compute the correlation coefficient between 'Top10perc' and 'Top25perc'  
correlation = df['Top10perc'].corr(df['Top25perc'])  
print(f"The correlation coefficient between the percentage of students from the top 10% and the top 25% of their higher secondary class is {correlation:.2f}")

The correlation coefficient between the percentage of students from the top 10% and the top 25% of their higher secondary class is 0.89

#### Plotting the scatter plot for Top10perc vs Top25perc

# Plotting the scatter plot for Top10perc vs Top25perc  
plt.figure(figsize=(5, 5))  
sns.scatterplot(x='Top10perc', y='Top25perc', data=df, color='skyblue')  
plt.title('Correlation between Top 10% and Top 25% of High School Class')  
plt.xlabel('Percentage of Students from Top 10%')  
plt.ylabel('Percentage of Students from Top 25%')  
plt.grid(True)  
plt.tight\_layout()  
plt.show()



* The scatter plot shows a positive correlation between Top10perc and Top25perc. This means that colleges with a higher percentage of students from the top 10% of their high school class also tend to have a higher percentage of students from the top 25% of their high school class. This indicates that colleges attracting top-performing students (top 10%) are likely also attracting a broader range of high-performing students (top 25%).

## Student Demographics

#### Calculate the average number of full-time undergraduate students per college

# Calculate the average number of full-time undergraduate students per college  
fulltime = df['F.Undergrad'].mean()  
print(f"Average number of full-time undergraduate students per college: {fulltime:.2f}")

Average number of full-time undergraduate students per college: 3699.91

#### Calculate the average number of part-time undergraduate students per college

# Calculate the average number of part-time undergraduate students per college  
parttime = df['P.Undergrad'].mean()  
print(f"Average number of part-time undergraduate students per college:{parttime:.2f}")

Average number of part-time undergraduate students per college:855.30

#### Find the college with the highest number of out-of-state students

# Find the college with the highest number of out-of-state students  
outstate\_students = df.loc[df['Outstate'].idxmax(), 'Names']  
print("The College with the highest number of out-of-state students is: ",outstate\_students)

The College with the highest number of out-of-state students is: Bennington College

## Cost and Spending

#### Calculate the average cost of room and board across all colleges

# Calculate the average cost of room and board across all colleges  
avg\_personal = df['Personal'].mean()  
print(f"Avg. estimated personal spending: {avg\_personal:.2f}")

Avg. estimated personal spending: 1599.96

#### Calculate the average estimated book cost for a student

# Calculate the average estimated book cost for a student  
avg\_books = df['Books'].mean()  
print(f"Avg. estimated book cost: {avg\_books:.2f}")

Avg. estimated book cost: 547.88

#### Calculate the average estimated personal spending for a student

# Calculate the average estimated personal spending for a student  
avg\_personal = df['Personal'].mean()  
print(f"Avg. estimated personal spending: {avg\_personal:.2f}")

Avg. estimated personal spending: 1599.96

#### Calculate the variation of instructional expenditure per student across colleges

# Calculate the variation of instructional expenditure per student across colleges  
instructional\_expenditure\_variation = df['Expend'].describe()  
print("Variation of instructional expenditure per student:")  
print(instructional\_expenditure\_variation)

Variation of instructional expenditure per student:  
count 777.000000  
mean 9660.171171  
std 5221.768440  
min 3186.000000  
25% 6751.000000  
50% 8377.000000  
75% 10830.000000  
max 56233.000000  
Name: Expend, dtype: float64

* This reveals a wide range of instructional expenditures per student across different colleges. This indicates significant variability in how much colleges invest in instructional costs per student.

## Faculty Qualification

#### Average percentage of faculties with Ph.D.s across all colleges

# Average percentage of faculties with Ph.D.s across all colleges  
avg\_phd = df['PhD'].mean()  
print(f"Avg.percentage of faculties with Ph.D.s: {avg\_phd:.2f}%")

Avg.percentage of faculties with Ph.D.s: 72.66%

#### Average percentage of faculties with terminal degrees across all colleges

# Average percentage of faculties with terminal degrees across all colleges  
avg\_terminal = df['Terminal'].mean()  
print(f"Avg. percentage of faculties with terminal degrees: {avg\_terminal:.2f}%")

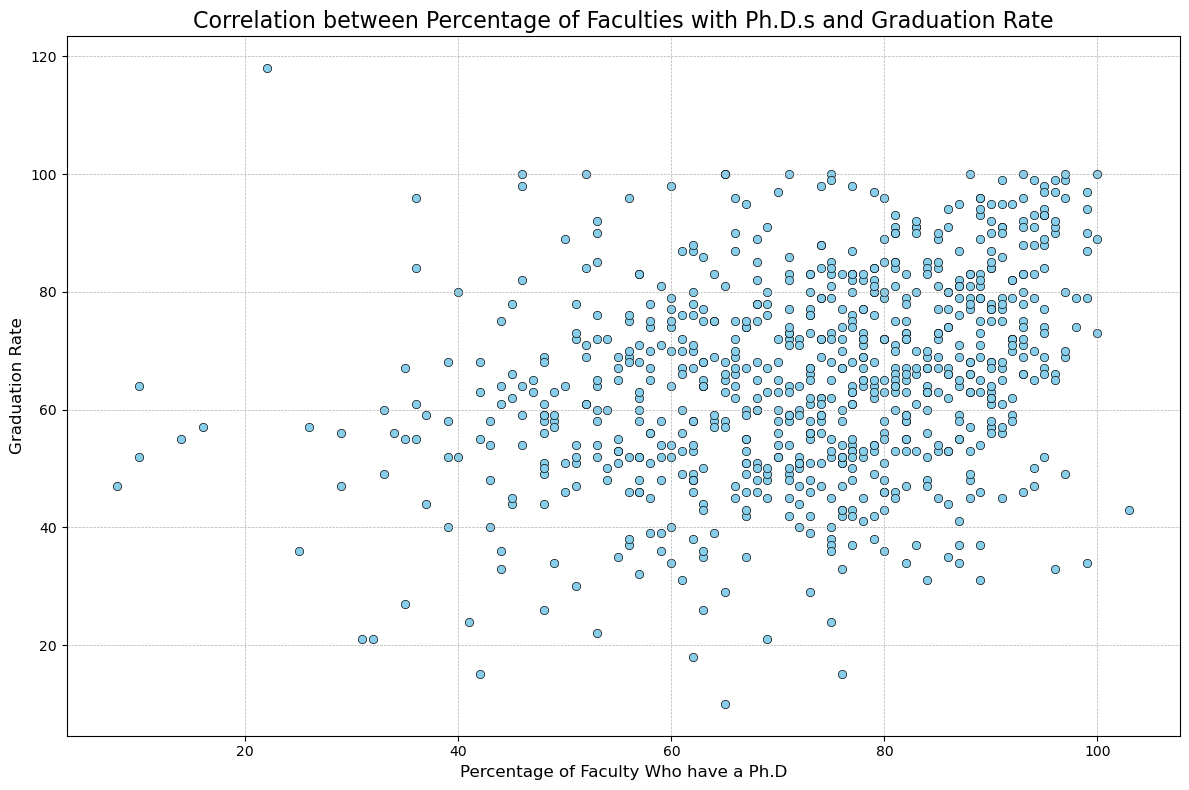
Avg. percentage of faculties with terminal degrees: 79.70%

#### Correlation between the percentage of faculties with Ph.D.s and the graduation rate

# Correlation between the percentage of faculties with Ph.D.s and the graduation rate  
corr\_phd\_grad = df['PhD'].corr(df['Grad.Rate'])  
print(f"Correlation between faculties with Ph.D.s and graduation rate: {corr\_phd\_grad:.2f}")

Correlation between faculties with Ph.D.s and graduation rate: 0.31

plt.figure(figsize=(12, 8))  
sns.scatterplot(x='PhD', y='Grad.Rate', data=df, color='skyblue', edgecolor='black')  
plt.title('Correlation between Percentage of Faculties with Ph.D.s and Graduation Rate', fontsize=16)  
plt.xlabel('Percentage of Faculty Who have a Ph.D', fontsize=12)  
plt.ylabel('Graduation Rate', fontsize=12)  
plt.grid(True, linestyle='--', linewidth=0.5)  
plt.tight\_layout()  
plt.show()



* The scatter plot demonstrates a positive relationship between the percentage of faculty with Ph.D.s and the graduation rate. This indicates that as the proportion of Ph.D. faculty rises, the graduation rate generally increases.
* There is significant variation in graduation rates even among colleges with high percentages of Ph.D. faculty. This suggests that while a higher percentage of Ph.D. faculty may contribute to higher graduation rates, it is not the sole factor.

## Student-Faculty Interaction

#### Average student/faculty ratio across all colleges

# Average student/faculty ratio across all colleges  
avg\_ratio = df['S.F.Ratio'].mean()  
print(f"Avg. student/faculty ratio: {avg\_ratio:.2f}")

Avg. student/faculty ratio: 14.09

#### College with the lowest student/faculty ratio

# College with the lowest student/faculty ratio  
lowest\_sfratio\_college = df.loc[df['S.F.Ratio'].idxmin(), 'Names']  
print(f"College with lowest student/faculty ratio: {lowest\_sfratio\_college}")

College with lowest student/faculty ratio: University of Charleston

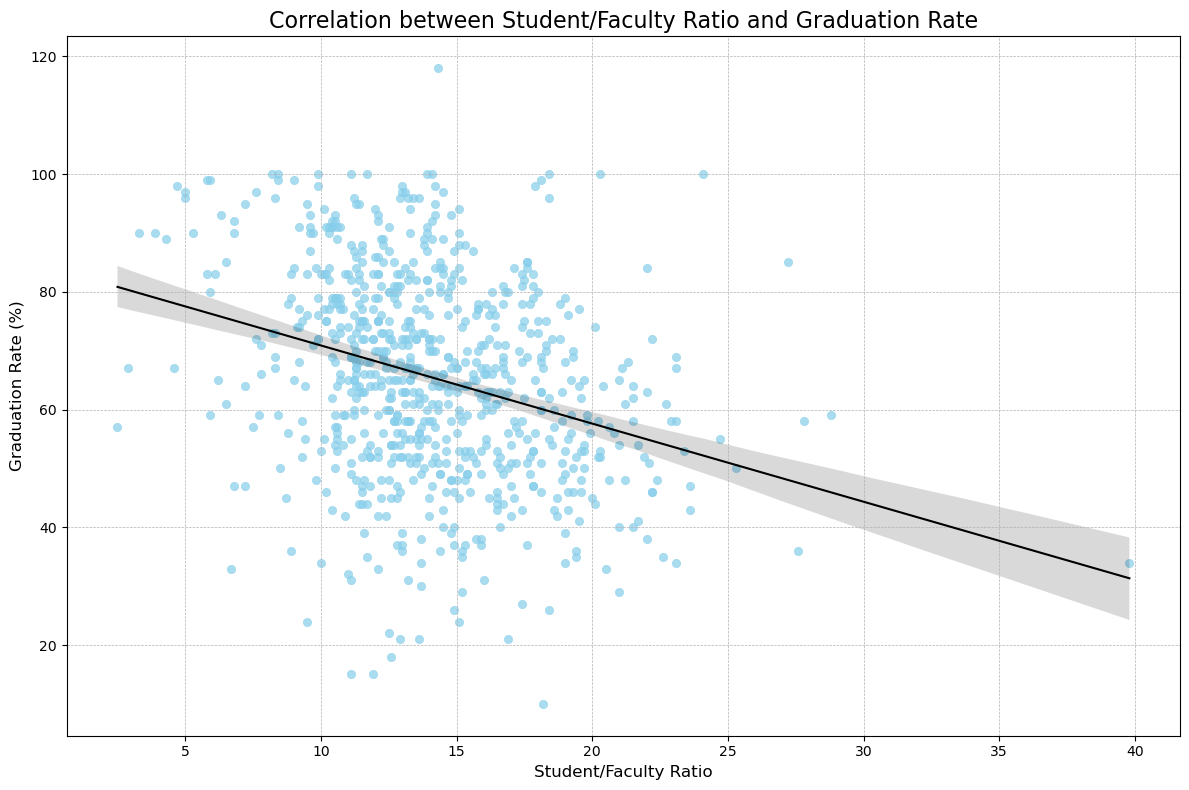
#### Correlation between the student/faculty ratio and the graduation rate

# Correlation between the student/faculty ratio and the graduation rate  
corr\_sfratio\_grad = df['S.F.Ratio'].corr(df['Grad.Rate'])  
print(f"Correlation between student/faculty ratio and graduation rate: {corr\_sfratio\_grad:.2f}")

Correlation between student/faculty ratio and graduation rate: -0.30

#### Plotting the scatter plot for student/faculty ratio vs graduation rate

# Plotting the scatter plot for student/faculty ratio vs graduation rate  
plt.figure(figsize=(12, 8))  
sns.scatterplot(x='S.F.Ratio', y='Grad.Rate', data=df, color='skyblue', alpha=0.7, edgecolor=None)  
sns.regplot(x='S.F.Ratio', y='Grad.Rate', data=df, scatter=False, color='black', line\_kws={'linewidth': 1.5})  
plt.title('Correlation between Student/Faculty Ratio and Graduation Rate', fontsize=16)  
plt.xlabel('Student/Faculty Ratio', fontsize=12)  
plt.ylabel('Graduation Rate (%)', fontsize=12)  
plt.xticks(fontsize=10)  
plt.yticks(fontsize=10)  
plt.grid(True, linestyle='--', linewidth=0.5)  
plt.tight\_layout()  
plt.show()



## Alumni Engagement

#### Average percentage of alumni who donate across all colleges

# Average percentage of alumni who donate across all colleges  
avg\_alumni = df['perc.alumni'].mean()  
print(f"Average alumni donation rate: {avg\_alumni:.2f}%")

Average alumni donation rate: 22.74%

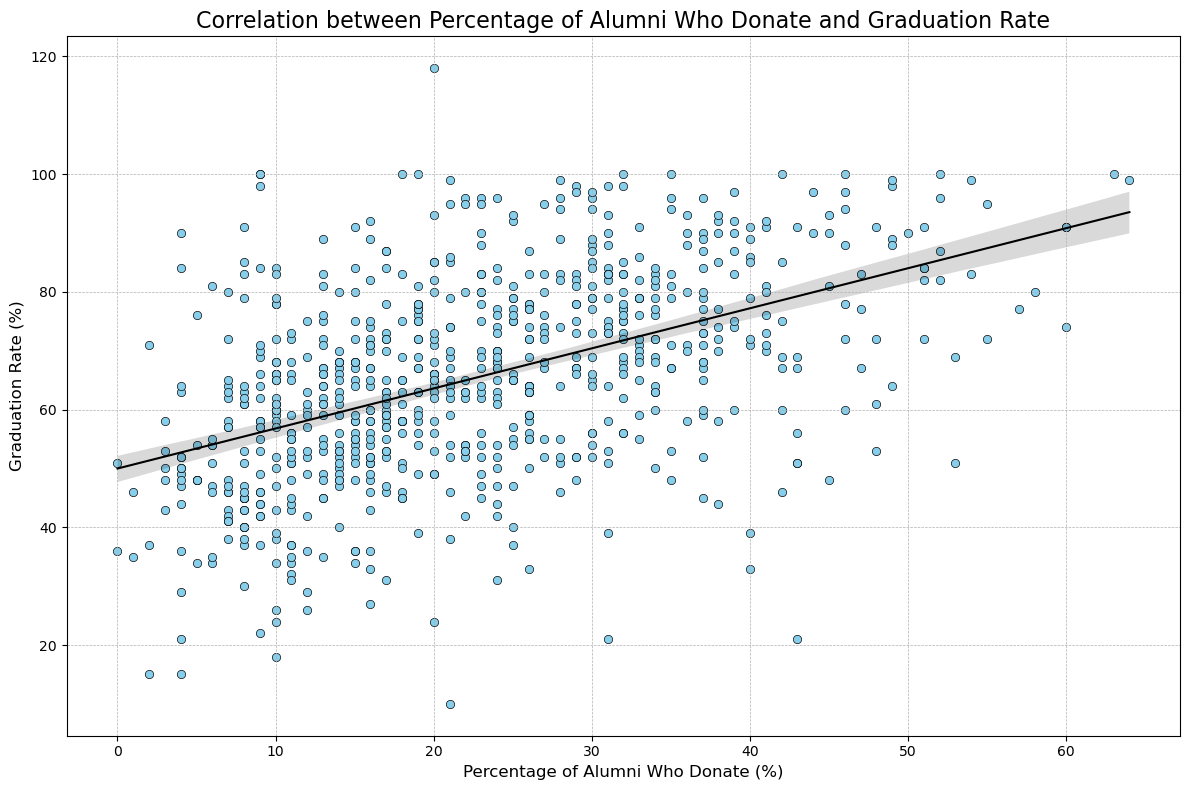
#### Correlation between the percentage of alumni who donate and the graduation rate

# Correlation between the percentage of alumni who donate and the graduation rate  
corr\_alumni\_grad = df['perc.alumni'].corr(df['Grad.Rate'])  
print(f"Correlation between alumni donation rate and graduation rate: {corr\_alumni\_grad:.2f}")

Correlation between alumni donation rate and graduation rate: 0.49

#### Plotting the scatter plot for percentage of alumni who donate vs graduation rate

# Plotting the scatter plot for percentage of alumni who donate vs graduation rate  
plt.figure(figsize=(12, 8))  
sns.scatterplot(x='perc.alumni', y='Grad.Rate', data=df, color='skyblue', edgecolor='black')  
sns.regplot(x='perc.alumni', y='Grad.Rate', data=df, scatter=False, color='black', line\_kws={'linewidth': 1.5})  
plt.title('Correlation between Percentage of Alumni Who Donate and Graduation Rate', fontsize=16)  
plt.xlabel('Percentage of Alumni Who Donate (%)', fontsize=12)  
plt.ylabel('Graduation Rate (%)', fontsize=12)  
plt.xticks(fontsize=10)  
plt.yticks(fontsize=10)  
plt.grid(True, linestyle='--', linewidth=0.5)  
plt.tight\_layout()  
plt.show()



* There is considerable variability in graduation rates among colleges with similar alumni donation percentages. While the overall trend shows that higher percentages of alumni donations are linked to higher graduation rates, some colleges with lower alumni donation percentages still exhibit high graduation rates.

## Graduation Rates

#### Average graduation rate across all colleges

# Average graduation rate across all colleges  
average\_graduation\_rate = df['Grad.Rate'].mean()  
print(f"Average graduation rate: {average\_graduation\_rate:.2f}%")

Average graduation rate: 65.46%

#### College with the highest graduation rate

# College with the highest graduation rate  
top\_col = df.loc[df['Grad.Rate'].idxmax(), 'Names']  
print(f"College with highest graduation rate: {top\_col}")

College with highest graduation rate: Cazenovia College

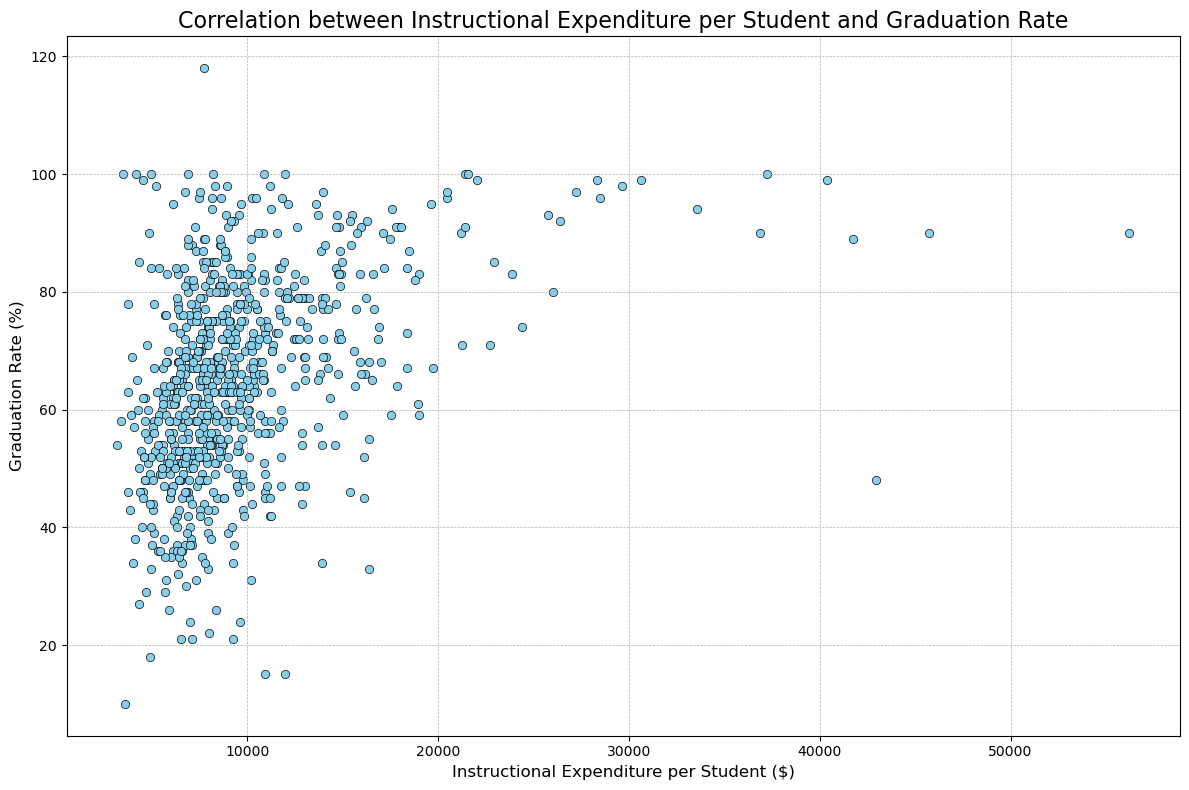
#### Correlation between the instructional expenditure per student and the graduation rate

# Correlation between the instructional expenditure per student and the graduation rate  
expend\_grad\_corr = df['Expend'].corr(df['Grad.Rate'])  
print(f"Correlation between instructional expenditure per student and graduation rate: {expend\_grad\_corr:.2f}")

Correlation between instructional expenditure per student and graduation rate: 0.39

#### Plotting the scatter plot for instructional expenditure per student vs graduation rate

# Plotting the scatter plot for instructional expenditure per student vs graduation rate  
plt.figure(figsize=(12, 8))  
sns.scatterplot(x='Expend', y='Grad.Rate', data=df, color='skyblue', edgecolor='black')  
plt.title('Correlation between Instructional Expenditure per Student and Graduation Rate', fontsize=16)  
plt.xlabel('Instructional Expenditure per Student ($)', fontsize=12)  
plt.ylabel('Graduation Rate (%)', fontsize=12)  
plt.grid(True, linestyle='--', linewidth=0.5)  
plt.tight\_layout()  
plt.show()



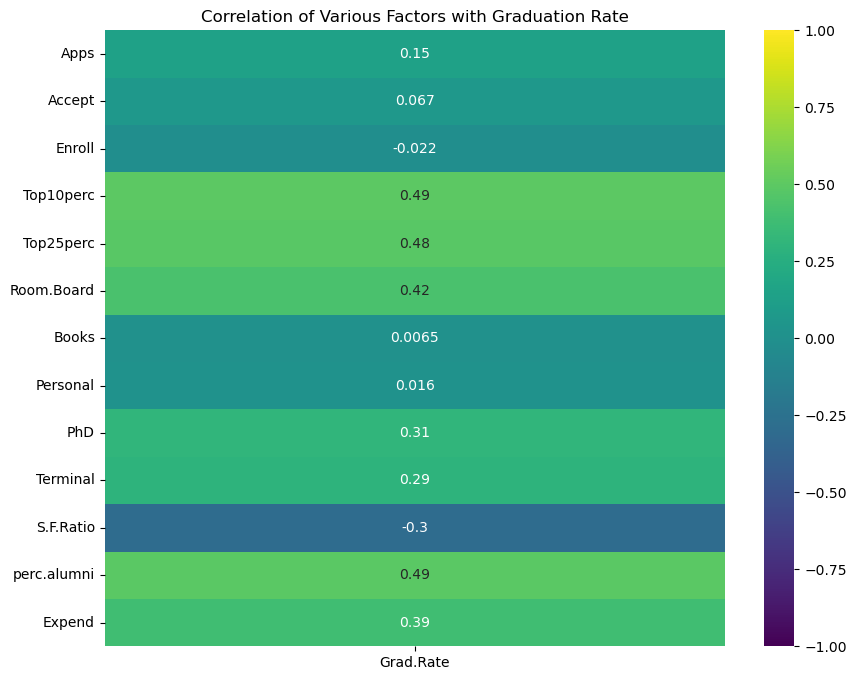
## Overall Insights

#### Calculate correlations between various factors

#### Extract the correlations with graduation rate

#### Plotting the heatmap

# Calculate correlations between various factors  
correlation\_matrix = df[['Apps', 'Accept', 'Enroll', 'Top10perc', 'Top25perc', 'Room.Board', 'Books', 'Personal', 'PhD', 'Terminal', 'S.F.Ratio', 'perc.alumni', 'Expend', 'Grad.Rate']].corr()  
  
# Extract the correlations with graduation rate  
correlation\_with\_gradrate = correlation\_matrix[['Grad.Rate']].drop('Grad.Rate')  
  
# Plotting the heatmap  
plt.figure(figsize=(10, 8))  
sns.heatmap(correlation\_with\_gradrate, annot=True, cmap='viridis', cbar=True, vmin=-1, vmax=1)  
plt.title('Correlation of Various Factors with Graduation Rate')  
plt.show()



**Instructional Expenditure (Expend)** The correlation analysis indicates a positive relationship between the instructional expenditure per student and the graduation rate. This implies that higher investments in instructional resources may play a role in improving student outcomes and graduation rates.

**Percentage of Alumni Who Donate (perc.alumni)** There is a noticeable positive correlation between the percentage of alumni who donate and the graduation rate. This suggests that a higher level of alumni engagement and donations could be associated with better institutional support and resources, leading to improved graduation rates.

**Student/Faculty Ratio (S.F.Ratio)** An inverse relationship is observed between the student/faculty ratio and the graduation rate. This indicates that lower student/faculty ratios, allowing for more individualized attention and support, tend to be linked with higher graduation rates.

**Faculty Qualifications (PhD, Terminal)** There is a positive correlation between the percentage of faculty with Ph.D.s and terminal degrees and the graduation rate. This implies that institutions with higher percentages of faculty with advanced degrees may provide better educational quality, potentially contributing to higher graduation rates.

**Academic Excellence (Top10perc, Top25perc)** Positive correlations are found between the percentages of students in the top 10% and top 25% of their high school classes and the graduation rate. This suggests that colleges with higher proportions of academically successful students generally have higher graduation rates.

**Acceptance and Enrollment Metrics (Apps, Accept, Enroll)** While there are correlations between the number of applications, acceptance rates, and enrollment numbers and the graduation rate, these correlations are not as strong as those with other factors. This implies that these metrics may have less direct impact on graduation outcomes compared to factors like expenditure and faculty qualifications.

**Out-of-State Students (Outstate)** A slight positive correlation is identified between the number of out-of-state students and the graduation rate. This may indicate that institutions attracting a diverse student body from different regions can create environments that support higher graduation rates.

**Room and Board Costs (Room.Board)** There is a slight positive correlation between room and board costs and graduation rates. This could suggest that institutions with higher room and board costs may provide better facilities and support, potentially leading to higher graduation rates.

# Suggestions

* **Enhance Academic Excellence**: Focus on admitting students who excelled in their higher secondary classes, as indicated by high correlations with Top10perc and Top25perc.
* **Increase Faculty Qualifications**: Improve graduation rates by hiring more faculty with Ph.D.s and terminal degrees, as suggested by high correlations with PhD and Terminal.
* **Improve Student Support Services**: Invest in support services such as tutoring, counseling, and career guidance to help students navigate challenges and stay on track. While the correlations with Accept, Enroll, and Apps are weaker, indicating factors beyond enrollment numbers play a role in graduation outcomes.
* **Improve Student-to-Faculty Ratio**: Maintain a lower student-to-faculty ratio to provide more personalized attention, positively impacting graduation rates, as indicated by a high correlation with S.F.Ratio.
* **Boost Alumni Engagement**: Increase donations and engagement from alumni to enhance facilities and programs for students, thus improving graduation rates, as suggested by a strong positive correlation with perc.alumni.
* **Focus on Out-of-State Student Recruitment**: Given the slight positive correlation with Outstate, colleges could focus on attracting more out-of-state students to create a diverse and potentially more engaged student body, which could positively influence graduation rates.
* **Increase Instructional Expenditure**: Improve graduation rates by investing more in instructional materials and resources per student, as indicated by a high correlation with Expend.