# DATA-200-01 Final Exam: Data Science Portfolio

## Graduation Rates of Black Male Students: HBCUs vs. PWIs

### Research Question

How does socioeconomic status affect how often Black male students graduate from Historically Black Colleges and Universities (HBCUs) compared to Predominantly White Institutions (PWIs)?

### Problem Summary

Black male students in the U.S. face real challenges when it comes to finishing college. This project looks at how a student’s financial background and life situation can impact graduation. We compare HBCUs and PWIs to see which environments help students succeed, and why. The goal is to find patterns and recommend ways to support Black male students better.

The data comes from the IPEDS system (Integrated Postsecondary Education Data System), which collects information from colleges across the country. This dataset includes:

- School names
- Graduation rates for Black male students

#### Problem Statement

Socioeconomic status (SES) refers to a person's or group's position in society, based on factors like income, education, and occupation. It can influence various aspects of life, including access to resources and opportunities. Research indicates that Black male students often face challenges in higher education, with graduation rates being a particular concern.his study aims to explore how SES affects the graduation rates of Black male students, comparing outcomes between Historically Black Colleges and Universities (HBCUs) and Predominantly White Institutions (PWIs).(\n) By examining elements such as financial aid availability, campus support services, and community involvement, the research seeks to understand how different college environments and resources impact the academic success of Black male students from various socioeconomic backgrounds.Gaining this understanding is important for creating effective strategies and policies to improve educational success and fairness in higher education. This research will provide insights into how socioeconomic factors and college settings interact, offering information that can help develop ways to support Black male students across different educational environments.

#### Data Definition

Integrated Postsecondary Education Data System

https://nces.ed.gov/ipeds/

PEDS is the Integrated Postsecondary Education Data System. It is a system of interrelated surveys conducted annually by the U.S. Department of Education's National Center for Education Statistics (NCES). IPEDS gathers information from every college, university, and technical and vocational institution that participates in the federal student financial aid programs. The Higher Education Act of 1965, as amended, requires that institutions that participate in federal student aid programs report data on enrollments, program completions, graduation rates, faculty and staff, finances, institutional prices, and student financial aid. These data are made available to students and parents through the College Navigator college search Web site and to researchers and others through the IPEDS Data Center.

In [1]:
```
# Import the libraries
import numpy as np                    # Scientific Computing
import pandas as pd                   # Data Analysis
import matplotlib.pyplot as plt       # Plotting
import seaborn as sns                 # Statistical Data Visualization

# Let's make sure pandas returns all the rows and columns for the dataframe
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

# Force pandas to display full numbers instead of scientific notation
# pd.options.display.float_format = '{:.0f}'.format

# Library to suppress warnings
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df1=pd.read_csv('Data_2-12-2025---876.csv')
```

In [3]:
```
educationUS=pd.read_csv('Data_2-12-2025---876.csv')

educationUS.head(10)
```

Out[3]:

| | instnm | Number completed a bachelor's degree within 100% of normal time (4-years) (GR200_23) | Historically Black College or University (HD2023) | Degree-granting status (HD2023) | Total price for in-state students living on campus 2023-24 (DRVIC2023) | Total price for out-of-state students living on campus 2023-24 (DRVIC2023) | Total pric for in-stat student living o campus (nc with family 2023-2 (DRVIC2023 |
|---|---|---|---|---|---|---|---|
| **0** | Columbia University in the City of New York | 1269 | 2 | 1 | 89587 | 89587 | 90987 |
| **1** | Cornell University | 2787 | 2 | 1 | 88140 | 88140 | 88140 |
| **2** | Delaware State University | 292 | 1 | 1 | 30236 | 40172 | 31091 |
| **3** | Duke University | 1557 | 2 | 1 | 87072 | 87072 | Na |
| **4** | Florida Agricultural and Mechanical University | 435 | 1 | 1 | 24153 | 36093 | 23345 |
| **5** | Hampton University | 369 | 1 | 1 | 46048 | 46048 | 47748 |
| **6** | Howard University | 869 | 1 | 1 | 54630 | 54630 | 63696 |
| **7** | Jackson State University | 288 | 1 | 1 | 32988 | 34008 | 32988 |
| **8** | Morehouse College | 191 | 1 | 1 | 51623 | 51623 | 62824 |
| **9** | Morgan State University | 233 | 1 | 1 | 26438 | 37120 | 28338 |

In [4]:
```
educationUS.tail(10)
```

Out[4]:

| | instnm | Number completed a bachelor's degree within 100% of normal time (4-years) (GR200_23) | Historically Black College or University (HD2023) | Degree-granting status (HD2023) | Total price for in-state students living on campus 2023-24 (DRVIC2023) | Total price for out-of-state students living on campus 2023-24 (DRVIC2023) | Total p for in-s stud livin campus with fa 202 (DRVIC2 |
|---|---|---|---|---|---|---|---|
| **10** | North Carolina Central University | 315 | 1 | 1 | 26788 | 39824 | 252 |
| **11** | Northwestern University | 1736 | 2 | 1 | 91290 | 91290 | 912 |
| **12** | Princeton University | 1178 | 2 | 1 | 84040 | 84040 | |
| **13** | Tuskegee University | 155 | 1 | 1 | 43454 | 43454 | 435 |
| **14** | University of California-Berkeley | 4351 | 2 | 1 | 45053 | 75830 | 40 |
| **15** | University of Maryland-College Park | 2855 | 2 | 1 | 30885 | 59686 | 329 |
| **16** | University of Michigan-Flint | 72 | 2 | 1 | 29886 | 47146 | 268 |
| **17** | University of Pennsylvania | 2124 | 2 | 1 | 89028 | 89028 | 888 |
| **18** | Xavier University of Louisiana | 203 | 1 | 1 | 41754 | 41754 | 471 |
| **19** | Yale University | 1194 | 2 | 1 | 88300 | 88300 | |

In [5]:
```
educationUS.shape
```

Out[5]:
```
(20, 16)
```

## Questions:

- State the shape of the dataframe :
  - How many rows does the dataframe have?
  - How many columns does the dataframe have?
  - What is the total number of datapoints expected in the dataset (rows x columns)?

Answers: Shape of the DataFrame: The shape is (20, 16), indicating 20 rows and 16 columns. Number of Rows: 20 Number of Columns: 16 Total Number of Data Points: 20 rows × 16

columns = 320 data points

In [6]:
```
educationUS.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 16 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   instnm
20 non-null     object
 1   Number completed a bachelor's degree within 100% of normal time (4-years) (GR
200_23)            20 non-null     int64
 2   Historically Black College or University (HD2023)
20 non-null     int64
 3   Degree-granting status (HD2023)
20 non-null     int64
 4   Total price for in-state students living on campus 2023-24 (DRVIC2023)
20 non-null     int64
 5   Total price for out-of-state students living on campus 2023-24 (DRVIC2023)
20 non-null     int64
 6   Total price for in-state students living off campus (not with family)  2023-2
4 (DRVIC2023)      17 non-null     float64
 7   Total price for out-of-state students living off campus (not with family)  20
23-24 (DRVIC2023)  17 non-null     float64
 8   Full-time retention rate  2023 (EF2023D)
20 non-null     int64
 9   Percent of total enrollment that are Black or African American (DRVEF2023)
20 non-null     int64
 10  Percent of undergraduate enrollment that are Black or African American (DRVEF
2023)              20 non-null     int64
 11  Graduation rate  men (DRVGR2023)
20 non-null     int64
 12  Graduation rate  Black  non-Hispanic (DRVGR2023)
20 non-null     int64
 13  Graduation rate - Bachelor degree within 6 years  Black  non-Hispanic (DRVGR2
023)               20 non-null     int64
 14  Two or more races men (C2023_B)
20 non-null     int64
 15  Unnamed: 15
0 non-null      float64
dtypes: float64(3), int64(12), object(1)
memory usage: 2.6+ KB
```

## Observations of the Data Set

Describe the dataset. How many rows and columns are there? What are the data types? Based on the number of expected data points, and those listed by the `.info()` method, how many missing (null) values are there?

The number of non-null values does not match the total number expected based on the number of rows in the dataframe. This indicates the presence of missing values that will need further investigation.

### Answer:

The dataset includes 20 rows and 16 columns with various data types (object, int64, and float64). Three columns contain missing values: two with 3 missing entries each and one with all entries missing.

# Data Cleaning:

In [7]:
```
educationUS.columns = educationUS.columns.str.title()
educationUS.columns
```

Out[7]:
```
Index(['Instnm',
       'Number Completed A Bachelor'S Degree Within 100% Of Normal Time (4-Years)
(Gr200_23)',
       'Historically Black College Or University (Hd2023)',
       'Degree-Granting Status (Hd2023)',
       'Total Price For In-State Students Living On Campus 2023-24 (Drvic2023)',
       'Total Price For Out-Of-State Students Living On Campus 2023-24 (Drvic202
3)',
       'Total Price For In-State Students Living Off Campus (Not With Family)  20
23-24 (Drvic2023)',
       'Total Price For Out-Of-State Students Living Off Campus (Not With Family)
2023-24 (Drvic2023)',
       'Full-Time Retention Rate  2023 (Ef2023D)',
       'Percent Of Total Enrollment That Are Black Or African American (Drvef202
3)',
       'Percent Of Undergraduate Enrollment That Are Black Or African American (D
rvef2023)',
       'Graduation Rate  Men (Drvgr2023)',
       'Graduation Rate  Black  Non-Hispanic (Drvgr2023)',
       'Graduation Rate - Bachelor Degree Within 6 Years  Black  Non-Hispanic (Dr
vgr2023)',
       'Two Or More Races Men (C2023_B)', 'Unnamed: 15'],
      dtype='object')
```

In [8]:
```
# institution names to state names

institution_to_state = {
    'Columbia University in the City of New York': 'New York',
    "Cornell University": 'New York',
    'Delaware State University': 'Delaware',
    'Duke University': 'North Carolina',
    'Florida Agricultural and Mechanical University': 'Florida',
    'Hampton University': 'Virginia',
    'Howard University': 'Washington D.C.',
    'Jackson State University': 'Mississippi',
    'Morehouse College': 'Georgia',
    'Morgan State University': 'Maryland',
    'North Carolina Central University': 'North Carolina',
    'Northwestern University': 'Illinois',
    'Princeton University': 'New Jersey',
    'Tuskegee University': 'Alabama',
    'University of California-Berkeley': 'California',
    'University of Maryland-College Park': 'Maryland',
    'University of Michigan-Flint': 'Michigan',
```

```
    'University of Pennsylvania': 'Pennsylvania',
    'Xavier University of Louisiana': 'Louisiana',
    'Yale University': 'Connecticut'
}
```

In [9]:
```
# Create a 'State' column
educationUS['State'] = educationUS['Instnm'].map(institution_to_state)


print(educationUS[['Instnm', 'State']].head())
```

```
                                       Instnm           State
0     Columbia University in the City of New York       New York
1                             Cornell University       New York
2                        Delaware State University       Delaware
3                                 Duke University  North Carolina
4  Florida Agricultural and Mechanical University        Florida
```

In [10]:
```
missing_states = educationUS[educationUS['State'].isna()]
print("Institutions with missing state mappings:")
print(missing_states[['Instnm']])
```

```
Institutions with missing state mappings:
Empty DataFrame
Columns: [Instnm]
Index: []
```

In [11]:
```
state_abbrev = {
    'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR', 'Californi
    'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE', 'Florida': 'FL', 'Geo
    'Hawaii': 'HI', 'Idaho': 'ID', 'Illinois': 'IL', 'Indiana': 'IN', 'Iowa': 'IA
    'Kansas': 'KS', 'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME', 'Maryland
    'Massachusetts': 'MA', 'Michigan': 'MI', 'Minnesota': 'MN', 'Mississippi': 'MS
    'Montana': 'MT', 'Nebraska': 'NE', 'Nevada': 'NV', 'New Hampshire': 'NH', 'New
    'New Mexico': 'NM', 'New York': 'NY', 'North Carolina': 'NC', 'North Dakota':
    'Oklahoma': 'OK', 'Oregon': 'OR', 'Pennsylvania': 'PA', 'Rhode Island': 'RI',
    'South Dakota': 'SD', 'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT', 'Vermont
    'Virginia': 'VA', 'Washington': 'WA', 'West Virginia': 'WV', 'Wisconsin': 'WI
}

educationUS['State Abbrev'] = educationUS['State'].map(state_abbrev)

educationUS[['Instnm', 'State', 'State Abbrev']].head()
```

Out[11]:

| | Instnm | State | State Abbrev |
|---|---|---|---|
| **0** | Columbia University in the City of New York | New York | NY |
| **1** | Cornell University | New York | NY |
| **2** | Delaware State University | Delaware | DE |
| **3** | Duke University | North Carolina | NC |
| **4** | Florida Agricultural and Mechanical University | Florida | FL |

In [12]:
```
# Let's Update the Headers for Syntax Consistency
# Syntax: df = df.rename(columns={'currentColumnName':'newColumnName', 'nextCurrer
```

```
# Let's view the new columns and update the variable
# Pass the columns to the variable: Use the variable = DataFrame.columns method

# Call the variable to see the contents
```

In [13]:
```
# Rename columns
educationUS = educationUS.rename(columns={
    'Instnm': 'Institution Name',
    'Number Completed A Bachelor\'S Degree Within 100% Of Normal Time (4-Years) ((
    'Graduation Rate  Men (Drvgr2023)': 'Graduation Rate - Men',
    'Degree-Granting Status (Hd2023)': 'Degree Granting',
    'Graduation Rate  Black  Non-Hispanic (Drvgr2023)': 'Graduation Rate - Black',
    'Graduation Rate - Bachelor Degree Within 6 Years  Black  Non-Hispanic (Drvgr2
    'Historically Black College Or University (Hd2023)': 'HBCU Status',
    'Total Price For In-State Students Living On Campus 2023-24 (Drvic2023)': 'In-
    'Total Price For Out-Of-State Students Living On Campus 2023-24 (Drvic2023)':
    'Total Price For In-State Students Living Off Campus (Not With Family)  2023-2
    'Total Price For Out-Of-State Students Living Off Campus (Not With Family)  20
    'Full-Time Retention Rate  2023 (Ef2023D)': 'Full-Time Retention Rate',
    'Percent Of Total Enrollment That Are Black Or African American (Drvef2023)':
    'Percent Of Undergraduate Enrollment That Are Black Or African American (Drve

})
```

In [14]:
```
educationUS.columns
```

Out[14]:
```
Index(['Institution Name', 'Degree Graduate (4-Years)', 'HBCU Status',
       'Degree Granting', 'In-State On-Campus Cost',
       'Out-of-State On-Campus Cost', 'In-State Off-Campus Cost',
       'Out-of-State Off-Campus Cost', 'Full-Time Retention Rate',
       'Total Black Enrollment', 'Black Undergraduate Enrollment',
       'Graduation Rate - Men', 'Graduation Rate - Black',
       'Grad Rate - Black (6 Years)', 'Two Or More Races Men (C2023_B)',
       'Unnamed: 15', 'State', 'State Abbrev'],
      dtype='object')
```

In [15]:
```
# institution names to state names

institution_to_state = {
    'Columbia University in the City of New York': 'New York',
    "Cornell University": 'New York',
    'Delaware State University': 'Delaware',
    'Duke University': 'North Carolina',
    'Florida Agricultural and Mechanical University': 'Florida',
    'Hampton University': 'Virginia',
    'Howard University': 'Washington D.C.',
    'Jackson State University': 'Mississippi',
    'Morehouse College': 'Georgia',
    'Morgan State University': 'Maryland',
    'North Carolina Central University': 'North Carolina',
    'Northwestern University': 'Illinois',
    'Princeton University': 'New Jersey',
    'Tuskegee University': 'Alabama',
    'University of California-Berkeley': 'California',
    'University of Maryland-College Park': 'Maryland',
    'University of Michigan-Flint': 'Michigan',
    'University of Pennsylvania': 'Pennsylvania',
    'Xavier University of Louisiana': 'Louisiana',
```

```
    'Yale University': 'Connecticut'
}
```

In [16]:
```
# Create a 'State' column
educationUS['State'] = educationUS['Institution Name'].map(institution_to_state)


print(educationUS[['Institution Name', 'State']].head())
```

```
                                Institution Name          State
0     Columbia University in the City of New York       New York
1                             Cornell University       New York
2                     Delaware State University       Delaware
3                                Duke University  North Carolina
4  Florida Agricultural and Mechanical University        Florida
```

In [17]:
```
missing_states = educationUS[educationUS['State'].isna()]
print("Institutions with missing state mappings:")
print(missing_states[['Institution Name']])
```

```
Institutions with missing state mappings:
Empty DataFrame
Columns: [Institution Name]
Index: []
```

In [18]:
```
state_abbrev = {
    'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR', 'Californi
    'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE', 'Florida': 'FL', 'Geo
    'Hawaii': 'HI', 'Idaho': 'ID', 'Illinois': 'IL', 'Indiana': 'IN', 'Iowa': 'IA
    'Kansas': 'KS', 'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME', 'Maryland
    'Massachusetts': 'MA', 'Michigan': 'MI', 'Minnesota': 'MN', 'Mississippi': 'MS
    'Montana': 'MT', 'Nebraska': 'NE', 'Nevada': 'NV', 'New Hampshire': 'NH', 'New
    'New Mexico': 'NM', 'New York': 'NY', 'North Carolina': 'NC', 'North Dakota':
    'Oklahoma': 'OK', 'Oregon': 'OR', 'Pennsylvania': 'PA', 'Rhode Island': 'RI',
    'South Dakota': 'SD', 'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT', 'Vermont
    'Virginia': 'VA', 'Washington': 'WA', 'West Virginia': 'WV', 'Wisconsin': 'WI
}

educationUS['State Abbrev'] = educationUS['State'].map(state_abbrev)

educationUS[['Institution Name', 'State', 'State Abbrev']].head()
```

Out[18]:

| | Institution Name | State | State Abbrev |
|---|---|---|---|
| **0** | Columbia University in the City of New York | New York | NY |
| **1** | Cornell University | New York | NY |
| **2** | Delaware State University | Delaware | DE |
| **3** | Duke University | North Carolina | NC |
| **4** | Florida Agricultural and Mechanical University | Florida | FL |

In [19]:
```
# Let's Update the Headers for Syntax Consistency
# Syntax: df = df.rename(columns={'currentColumnName':'newColumnName', 'nextCurren


# Let's view the new columns and update the variable
# Pass the columns to the variable: Use the variable = DataFrame.columns method
```

```
# Call the variable to see the contents
```

In [20]:
```
educationUS.columns = educationUS.columns.str.title()

educationUS.head(10)
```

Out[20]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost | Full-Time Retention Rate |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Columbia University in the City of New York | 1269 | 2 | 1 | 89587 | 89587 | 90987.0 | 90987.0 | 97 |
| **1** | Cornell University | 2787 | 2 | 1 | 88140 | 88140 | 88140.0 | 88140.0 | 98 |
| **2** | Delaware State University | 292 | 1 | 1 | 30236 | 40172 | 31091.0 | 41027.0 | 72 |
| **3** | Duke University | 1557 | 2 | 1 | 87072 | 87072 | NaN | NaN | 96 |
| **4** | Florida Agricultural and Mechanical University | 435 | 1 | 1 | 24153 | 36093 | 23345.0 | 35285.0 | 86 |
| **5** | Hampton University | 369 | 1 | 1 | 46048 | 46048 | 47748.0 | 47748.0 | 85 |
| **6** | Howard University | 869 | 1 | 1 | 54630 | 54630 | 63696.0 | 63696.0 | 90 |
| **7** | Jackson State University | 288 | 1 | 1 | 32988 | 34008 | 32988.0 | 34008.0 | 68 |
| **8** | Morehouse College | 191 | 1 | 1 | 51623 | 51623 | 62824.0 | 62824.0 | 86 |
| **9** | Morgan State University | 233 | 1 | 1 | 26438 | 37120 | 28338.0 | 39020.0 | 71 |

In [21]:
```
# Let's create a new column
# The goal is to create a map function to apply the new information based on an ex
# Syntax: dictionaryName = ({key: value})

# Call the variable to see the contents
```

In [22]:
```
institution_to_state  # institution-to-state
state_abbrev # state abbreviation
```

Out[22]:
```
{'Alabama': 'AL',
 'Alaska': 'AK',
 'Arizona': 'AZ',
 'Arkansas': 'AR',
 'California': 'CA',
 'Colorado': 'CO',
 'Connecticut': 'CT',
 'Delaware': 'DE',
 'Florida': 'FL',
 'Georgia': 'GA',
 'Hawaii': 'HI',
 'Idaho': 'ID',
 'Illinois': 'IL',
 'Indiana': 'IN',
 'Iowa': 'IA',
 'Kansas': 'KS',
 'Kentucky': 'KY',
 'Louisiana': 'LA',
 'Maine': 'ME',
 'Maryland': 'MD',
 'Massachusetts': 'MA',
 'Michigan': 'MI',
 'Minnesota': 'MN',
 'Mississippi': 'MS',
 'Missouri': 'MO',
 'Montana': 'MT',
 'Nebraska': 'NE',
 'Nevada': 'NV',
 'New Hampshire': 'NH',
 'New Jersey': 'NJ',
 'New Mexico': 'NM',
 'New York': 'NY',
 'North Carolina': 'NC',
 'North Dakota': 'ND',
 'Ohio': 'OH',
 'Oklahoma': 'OK',
 'Oregon': 'OR',
 'Pennsylvania': 'PA',
 'Rhode Island': 'RI',
 'South Carolina': 'SC',
 'South Dakota': 'SD',
 'Tennessee': 'TN',
 'Texas': 'TX',
 'Utah': 'UT',
 'Vermont': 'VT',
 'Virginia': 'VA',
 'Washington': 'WA',
 'West Virginia': 'WV',
 'Wisconsin': 'WI',
 'Wyoming': 'WY'}
```

In [23]:
```
# Read the dictionary into the dataframe
# Syntax: df['NewColumnName'] = df['ColumnToMatch'].map(dictionaryName)


#Print the first 20 rows of the resulting DataFrame
# Syntax: DataFrame.head(qty)
```

In [24]:
```
educationUS.head(20)
```

Out[24]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost | Full-Tim Retentic Rat |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Columbia University in the City of New York | 1269 | 2 | 1 | 89587 | 89587 | 90987.0 | 90987.0 | 9 |
| **1** | Cornell University | 2787 | 2 | 1 | 88140 | 88140 | 88140.0 | 88140.0 | 9 |
| **2** | Delaware State University | 292 | 1 | 1 | 30236 | 40172 | 31091.0 | 41027.0 | 7 |
| **3** | Duke University | 1557 | 2 | 1 | 87072 | 87072 | NaN | NaN | 9 |
| **4** | Florida Agricultural and Mechanical University | 435 | 1 | 1 | 24153 | 36093 | 23345.0 | 35285.0 | 8 |
| **5** | Hampton University | 369 | 1 | 1 | 46048 | 46048 | 47748.0 | 47748.0 | 8 |
| **6** | Howard University | 869 | 1 | 1 | 54630 | 54630 | 63696.0 | 63696.0 | 9 |
| **7** | Jackson State University | 288 | 1 | 1 | 32988 | 34008 | 32988.0 | 34008.0 | 6 |
| **8** | Morehouse College | 191 | 1 | 1 | 51623 | 51623 | 62824.0 | 62824.0 | 8 |
| **9** | Morgan State University | 233 | 1 | 1 | 26438 | 37120 | 28338.0 | 39020.0 | 7 |
| **10** | North Carolina Central University | 315 | 1 | 1 | 26788 | 39824 | 25295.0 | 38331.0 | 7 |
| **11** | Northwestern University | 1736 | 2 | 1 | 91290 | 91290 | 91290.0 | 91290.0 | 9 |
| **12** | Princeton University | 1178 | 2 | 1 | 84040 | 84040 | NaN | NaN | 9 |
| **13** | Tuskegee University | 155 | 1 | 1 | 43454 | 43454 | 43599.0 | 43599.0 | 7 |
| **14** | University of California-Berkeley | 4351 | 2 | 1 | 45053 | 75830 | 40317.0 | 71094.0 | 9 |
| **15** | University of Maryland-College Park | 2855 | 2 | 1 | 30885 | 59686 | 32915.0 | 61716.0 | 9 |

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost | Full-Tim Retentic Rat |
|---|---|---|---|---|---|---|---|---|---|
| **16** | University of Michigan-Flint | 72 | 2 | 1 | 29886 | 47146 | 26856.0 | 44116.0 | 7 |
| **17** | University of Pennsylvania | 2124 | 2 | 1 | 89028 | 89028 | 88892.0 | 88892.0 | 9 |
| **18** | Xavier University of Louisiana | 203 | 1 | 1 | 41754 | 41754 | 47148.0 | 47148.0 | 7 |
| **19** | Yale University | 1194 | 2 | 1 | 88300 | 88300 | NaN | NaN | 9 |

In [25]:
```
# Determine the number of missing values
# Syntax: DataFrame.isnull().sum()
```

In [26]:
```
missing_values = educationUS.isnull().sum()

missing_values
```

Out[26]:
```
Institution Name                    0
Degree Graduate (4-Years)           0
Hbcu Status                         0
Degree Granting                     0
In-State On-Campus Cost             0
Out-Of-State On-Campus Cost         0
In-State Off-Campus Cost            3
Out-Of-State Off-Campus Cost        3
Full-Time Retention Rate            0
Total Black Enrollment              0
Black Undergraduate Enrollment      0
Graduation Rate - Men               0
Graduation Rate - Black             0
Grad Rate - Black (6 Years)         0
Two Or More Races Men (C2023_B)     0
Unnamed: 15                        20
State                               0
State Abbrev                        1
dtype: int64
```

In [27]:
```
# Let's create a function to determine the percentage of missing values
# Typically less than five percent missing values may not affect the results
# More than 5% can be dropped, replaced with existing data, or imputed using mean
# Syntax: def missing(DataFrame):
#    print ('Percentage of missing values in the dataset:\n',
#           round((DataFtame.isnull().sum() * 100/ len(DataFrame)),2).sort_values(

# Call the function and execute
# Syntax: missing(DataFrame)
```

In [28]:
```
def missing(educationUS):
    print("Percentage of missing values in the dataset:\n")
    print(
```

```
        round((educationUS.isnull().sum() * 100 / len(df1)), 2)
        .sort_values(ascending=False)
    )

missing(educationUS)
```

```
Percentage of missing values in the dataset:

Unnamed: 15                         100.0
In-State Off-Campus Cost             15.0
Out-Of-State Off-Campus Cost         15.0
State Abbrev                          5.0
Black Undergraduate Enrollment        0.0
State                                 0.0
Two Or More Races Men (C2023_B)       0.0
Grad Rate - Black (6 Years)           0.0
Graduation Rate - Black               0.0
Graduation Rate - Men                 0.0
Institution Name                      0.0
Degree Graduate (4-Years)             0.0
Full-Time Retention Rate              0.0
Out-Of-State On-Campus Cost           0.0
In-State On-Campus Cost               0.0
Degree Granting                       0.0
Hbcu Status                           0.0
Total Black Enrollment                0.0
dtype: float64
```

In [29]:
```
# Drop the rows
# Syntax: DataFrame.dropna(*, axis=0, how=_NoDefault.no_default, thresh=_NoDefaul


# Drop the columns by name
# Use either labels=[list] and columns=labels or columns=[list]
# Syntax: DataFrame.drop(labels=None, *, axis=0, index=None, columns=None, level=N

# Check the null count again
# Syntax: DataFrame.isnull().sum()
```

In [30]:
```
educationUS_cleaned=educationUS.dropna(axis=0)

educationUS_cleaned.isnull().sum()
```

Out[30]:
```
Institution Name                    0.0
Degree Graduate (4-Years)           0.0
Hbcu Status                         0.0
Degree Granting                     0.0
In-State On-Campus Cost             0.0
Out-Of-State On-Campus Cost         0.0
In-State Off-Campus Cost            0.0
Out-Of-State Off-Campus Cost        0.0
Full-Time Retention Rate            0.0
Total Black Enrollment              0.0
Black Undergraduate Enrollment      0.0
Graduation Rate - Men               0.0
Graduation Rate - Black             0.0
Grad Rate - Black (6 Years)         0.0
Two Or More Races Men (C2023_B)     0.0
Unnamed: 15                         0.0
State                               0.0
State Abbrev                        0.0
dtype: float64
```

## Observations

- How many missing values are there?
- Are these concentrated in specific rows or columns? How does this affect the analysis?
- Based on the information that is present, how should the missing values be handled? How will this affect the analysis?

In [31]:
```
total_missing=educationUS.isnull().sum().sum()
"Total Missing Values:", total_missing
```

Out[31]:
```
('Total Missing Values:', 27)
```

In [32]:
```
columns_with_missing=educationUS.isnull().sum()
columns_with_missing=columns_with_missing[columns_with_missing > 0]
"Columns with Missing Values:\n", columns_with_missing
```

Out[32]:
```
('Columns with Missing Values:\n',
 In-State Off-Campus Cost         3
 Out-Of-State Off-Campus Cost     3
 Unnamed: 15                     20
 State Abbrev                     1
 dtype: int64)
```

In [33]:
```
rows_with_missing=educationUS[educationUS.isnull().any(axis=1)]
"Total Rows with Missing Values:", rows_with_missing.shape[0]
```

Out[33]:
```
('Total Rows with Missing Values:', 20)
```

## Answer:

If the columns contain critical data (graduation rates or tuition costs), it could bias the analysis. If missing values are spread across many rows, dropping an entire row can lead to a loss of valuable data.

Statistical modeling does not work well with missing values. Missing values must be handled before continuing the analysis.

In [34]:
```
# Make a copy of the DataFrame before manipulation
# Syntax: DataFrameOG = workingDF
```

In [35]:
```
educationUS_original=educationUS.copy()

educationUS_original.head()
```

Out[35]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost | Full-Time Retention Rate |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Columbia University in the City of New York | 1269 | 2 | 1 | 89587 | 89587 | 90987.0 | 90987.0 | 97 |
| **1** | Cornell University | 2787 | 2 | 1 | 88140 | 88140 | 88140.0 | 88140.0 | 98 |
| **2** | Delaware State University | 292 | 1 | 1 | 30236 | 40172 | 31091.0 | 41027.0 | 72 |
| **3** | Duke University | 1557 | 2 | 1 | 87072 | 87072 | NaN | NaN | 96 |
| **4** | Florida Agricultural and Mechanical University | 435 | 1 | 1 | 24153 | 36093 | 23345.0 | 35285.0 | 86 |

In [36]:
```
# Let's split the dataframe into subsets as needed.
# Syntax: NewSubsetDF = DataFrame.query('ColumnName == "ValuetoSplit"')

# Check the new dataframe with .head() method.
```

In [37]:
```
HBCU_subset = educationUS[educationUS['Hbcu Status'] == 1]

HBCU_subset.head()
```

Out[37]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost | Full-Time Retention Rate |
|---|---|---|---|---|---|---|---|---|---|
| **2** | Delaware State University | 292 | 1 | 1 | 30236 | 40172 | 31091.0 | 41027.0 | 72 |
| **4** | Florida Agricultural and Mechanical University | 435 | 1 | 1 | 24153 | 36093 | 23345.0 | 35285.0 | 86 |
| **5** | Hampton University | 369 | 1 | 1 | 46048 | 46048 | 47748.0 | 47748.0 | 85 |
| **6** | Howard University | 869 | 1 | 1 | 54630 | 54630 | 63696.0 | 63696.0 | 90 |
| **7** | Jackson State University | 288 | 1 | 1 | 32988 | 34008 | 32988.0 | 34008.0 | 68 |

In [38]:
```
high_grad_subset = educationUS.query('`Graduation Rate - Black` > 50')

high_grad_subset.head()
```

Out[38]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost | Full-Time Retention Rate |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Columbia University in the City of New York | 1269 | 2 | 1 | 89587 | 89587 | 90987.0 | 90987.0 | 97 |
| **1** | Cornell University | 2787 | 2 | 1 | 88140 | 88140 | 88140.0 | 88140.0 | 98 |
| **3** | Duke University | 1557 | 2 | 1 | 87072 | 87072 | NaN | NaN | 96 |
| **4** | Florida Agricultural and Mechanical University | 435 | 1 | 1 | 24153 | 36093 | 23345.0 | 35285.0 | 86 |
| **5** | Hampton University | 369 | 1 | 1 | 46048 | 46048 | 47748.0 | 47748.0 | 85 |

In [39]:
```
# Let's drop all the NaN values from the new DF
# Syntax: NewSubsetDF = NewSubsetDF.filter(items=['ColumnName1','ColumnName2','Co

# Check the DataFrame again: NewSubsetDF.head()
```

In [40]:
```
cleaned_subset = educationUS.dropna().filter(items=['State', 'State Abbrev', 'Inst

cleaned_subset.head()
```

Out[40]:

| | State | State Abbrev | Institution Name |
|---|---|---|---|

In [41]:
```
educationUS=educationUS.drop(columns=['Unnamed: 15', 'Two Or More Races Men (C202
```

In [42]:
```
"Columns after dropping:", educationUS.columns
```

Out[42]:
```
('Columns after dropping:',
 Index(['Institution Name', 'Degree Graduate (4-Years)', 'Hbcu Status',
        'Degree Granting', 'In-State On-Campus Cost',
        'Out-Of-State On-Campus Cost', 'In-State Off-Campus Cost',
        'Out-Of-State Off-Campus Cost', 'Full-Time Retention Rate',
        'Total Black Enrollment', 'Black Undergraduate Enrollment',
        'Graduation Rate - Men', 'Graduation Rate - Black',
        'Grad Rate - Black (6 Years)', 'State', 'State Abbrev'],
       dtype='object'))
```

### Observations on Missing Values

Include observations on the methods used to update the dataFrame for missing values and complete the cleaning process. Did you use a method to impute the missing data? Did you choose to drop null or NaN values? Did you split the dataset to create subsets? Use this space to explain the techniques, approach, and reasoning.

### Answers:

Missing values were not imputed. Missing values were dropped or managed by creating subsets.

Missing values were dropped using .dropna(). If a column or row had too many missing values, it was either dropped or removed.

The data set was split using .query() and .filter() to create new subsets. Substets were based on: State categories (filtering institutions by state) Graduation rates (filtering institutions where the Black male graduation rate was over 50%) HBCU vs. PWI classifications (separating historically black colleges and predonimantly white institutions)

# Data Exploration:

In [43]:
```
## Describe the descriptive stats
# Syntax: DataFrame.describe()

# Note: If we do not pass include=object to the describe(), it would return statis
```

In [74]:
```
educationUS.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Institution Name                20 non-null     object
 1   Degree Graduate (4-Years)       20 non-null     int64
 2   Hbcu Status                     20 non-null     int64
 3   Degree Granting                 20 non-null     int64
 4   In-State On-Campus Cost         20 non-null     int64
 5   Out-Of-State On-Campus Cost     20 non-null     int64
 6   In-State Off-Campus Cost        17 non-null     float64
 7   Out-Of-State Off-Campus Cost    17 non-null     float64
 8   Full-Time Retention Rate        20 non-null     int64
 9   Total Black Enrollment          20 non-null     int64
 10  Black Undergraduate Enrollment  20 non-null     int64
 11  Graduation Rate - Men           20 non-null     int64
 12  Graduation Rate - Black         20 non-null     int64
 13  Grad Rate - Black (6 Years)     20 non-null     int64
 14  State                           20 non-null     object
 15  State Abbrev                    19 non-null     object
dtypes: float64(2), int64(11), object(3)
memory usage: 2.6+ KB
```

In [75]:
```
educationUS.shape
```

Out[75]:
```
(20, 16)
```

In [76]:
```
educationUS.isnull().sum()
```

Out[76]:
```
Institution Name                  0
Degree Graduate (4-Years)         0
Hbcu Status                       0
Degree Granting                   0
In-State On-Campus Cost           0
Out-Of-State On-Campus Cost       0
In-State Off-Campus Cost          3
Out-Of-State Off-Campus Cost      3
Full-Time Retention Rate          0
Total Black Enrollment            0
Black Undergraduate Enrollment    0
Graduation Rate - Men             0
Graduation Rate - Black           0
Grad Rate - Black (6 Years)       0
State                             0
State Abbrev                      1
dtype: int64
```

## Data Dictionary

| Column Name | Description | Data Type |
|---|---|---|
| INSTITUTION_NAME | College or university name | string |
| HBCU_STATUS | Whether the school is an HBCU (1 = Yes, 0 = No) | integer |
| DEGREE_GRANTING | 1 = Yes, 0 = No | integer |

| Column Name | Description | Data Type |
|---|---|---|
| IN_STATE_ON_CAMPUS_COST | In-state cost for on-campus students | float |
| OUT_OF_STATE_ON_CAMPUS_COST | Out-of-state cost for on-campus students | float |
| FULL_TIME_RETENTION_RATE | Percent of full-time students who return | float |
| BLACK_UNDERGRAD_ENROLLMENT | Number of enrolled Black undergraduate students | integer |

In [44]:
```
# numeric descriptive stats
educationUS.describe()
```

Out[44]:

| | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out Of |
|---|---|---|---|---|---|---|---|
| **count** | 20.000000 | 20.000000 | 20.0 | 20.000000 | 20.000000 | 17.000000 | |
| **mean** | 1123.650000 | 1.500000 | 1.0 | 55069.650000 | 61242.250000 | 50909.941176 | 581 |
| **std** | 1158.037781 | 0.512989 | 0.0 | 26280.649121 | 22309.889931 | 25127.813801 | 209 |
| **min** | 72.000000 | 1.000000 | 1.0 | 24153.000000 | 34008.000000 | 23345.000000 | 340 |
| **25%** | 274.250000 | 1.000000 | 1.0 | 30722.750000 | 41358.500000 | 31091.000000 | 410 |
| **50%** | 652.000000 | 1.500000 | 1.0 | 45550.500000 | 53126.500000 | 43599.000000 | 477 |
| **75%** | 1601.750000 | 2.000000 | 1.0 | 87339.000000 | 87339.000000 | 63696.000000 | 710 |
| **max** | 4351.000000 | 2.000000 | 1.0 | 91290.000000 | 91290.000000 | 91290.000000 | 912 |

In [45]:
```
# categorical descriptive stats
educationUS.describe(include="object")
```

Out[45]:

| | Institution Name | State | State Abbrev |
|---|---|---|---|
| **count** | 20 | 20 | 19 |
| **unique** | 20 | 17 | 16 |
| **top** | Columbia University in the City of New York | New York | NY |
| **freq** | 1 | 2 | 2 |

In [46]:
```
# numeric and categorical descriptive stats together
educationUS.describe(include="all")
```

Out[46]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-Sta Campu |
|---|---|---|---|---|---|---|---|
| **count** | 20 | 20.000000 | 20.000000 | 20.0 | 20.000000 | 20.000000 | 17.( |
| **unique** | 20 | NaN | NaN | NaN | NaN | NaN | |
| **top** | Columbia University in the City of New York | NaN | NaN | NaN | NaN | NaN | |
| **freq** | 1 | NaN | NaN | NaN | NaN | NaN | |
| **mean** | NaN | 1123.650000 | 1.500000 | 1.0 | 55069.650000 | 61242.250000 | 50909 |
| **std** | NaN | 1158.037781 | 0.512989 | 0.0 | 26280.649121 | 22309.889931 | 25127 |
| **min** | NaN | 72.000000 | 1.000000 | 1.0 | 24153.000000 | 34008.000000 | 23345.( |
| **25%** | NaN | 274.250000 | 1.000000 | 1.0 | 30722.750000 | 41358.500000 | 31091.( |
| **50%** | NaN | 652.000000 | 1.500000 | 1.0 | 45550.500000 | 53126.500000 | 43599.( |
| **75%** | NaN | 1601.750000 | 2.000000 | 1.0 | 87339.000000 | 87339.000000 | 63696.( |
| **max** | NaN | 4351.000000 | 2.000000 | 1.0 | 91290.000000 | 91290.000000 | 91290.( |

In [47]:
```
## Descriptive stats for subset DataFrame

#HBCU subset
HBCU_subset.describe()
```

Out[47]:

| | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-S Off-Cam ( |
|---|---|---|---|---|---|---|---|
| **count** | 10.000000 | 10.0 | 10.0 | 10.000000 | 10.000000 | 10.00000 | 10.000 |
| **mean** | 335.000000 | 1.0 | 1.0 | 37811.200000 | 42472.600000 | 40607.20000 | 45268.600 |
| **std** | 205.948753 | 0.0 | 0.0 | 11090.153529 | 6659.180216 | 14761.03505 | 10497.979 |
| **min** | 155.000000 | 1.0 | 1.0 | 24153.000000 | 34008.000000 | 23345.00000 | 34008.000 |
| **25%** | 210.500000 | 1.0 | 1.0 | 27650.000000 | 37796.000000 | 29026.25000 | 38503.250 |
| **50%** | 290.000000 | 1.0 | 1.0 | 37371.000000 | 40963.000000 | 38293.50000 | 42313.000 |
| **75%** | 355.500000 | 1.0 | 1.0 | 45399.500000 | 45399.500000 | 47598.00000 | 47598.000 |
| **max** | 869.000000 | 1.0 | 1.0 | 54630.000000 | 54630.000000 | 63696.00000 | 63696.000 |

In [48]:
```
# categorical descriptive stats
HBCU_subset.describe(include="object")
```

Out[48]:

| | Institution Name | State | State Abbrev |
|---|---|---|---|
| **count** | 10 | 10 | 9 |
| **unique** | 10 | 10 | 9 |
| **top** | Delaware State University | Delaware | DE |
| **freq** | 1 | 1 | 1 |

In [49]:
```
#display subset stats
HBCU_subset.describe(include="all")
```

Out[49]:

| | Institution Name | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off Campu Cos |
|---|---|---|---|---|---|---|---|
| **count** | 10 | 10.000000 | 10.0 | 10.0 | 10.000000 | 10.000000 | 10.0000 |
| **unique** | 10 | NaN | NaN | NaN | NaN | NaN | Na |
| **top** | Delaware State University | NaN | NaN | NaN | NaN | NaN | Na |
| **freq** | 1 | NaN | NaN | NaN | NaN | NaN | Na |
| **mean** | NaN | 335.000000 | 1.0 | 1.0 | 37811.200000 | 42472.600000 | 40607.2000 |
| **std** | NaN | 205.948753 | 0.0 | 0.0 | 11090.153529 | 6659.180216 | 14761.0350 |
| **min** | NaN | 155.000000 | 1.0 | 1.0 | 24153.000000 | 34008.000000 | 23345.0000 |
| **25%** | NaN | 210.500000 | 1.0 | 1.0 | 27650.000000 | 37796.000000 | 29026.2500 |
| **50%** | NaN | 290.000000 | 1.0 | 1.0 | 37371.000000 | 40963.000000 | 38293.5000 |
| **75%** | NaN | 355.500000 | 1.0 | 1.0 | 45399.500000 | 45399.500000 | 47598.0000 |
| **max** | NaN | 869.000000 | 1.0 | 1.0 | 54630.000000 | 54630.000000 | 63696.0000 |

## Observations of Descriptive Statistics

The following are some observations about each table:
DataFrame:

- What are the minimum and maximum values?
- What are the mean values for the data?
- Are the mean and median values close to each other? If so this could indicate a normal distribution of the data. If not, this could indicate skewness in the data. If the mean is smaller than the median the values are likely skewed left, toward the minimum value. If the mean is larger than the median then there is skewness to the right indicating more high values in the data distribution.
- What are the quartile ranges for the data? What value is the 25th percentile of the data? What value is the 75th percentile of the data?
- How does the standard deviation for the data compare to the mean? High values for the standard deviation indicate a large variation in the data and likely a wide spread of the data

across the range from minimum to maximum.

Dataframe Subset:

- Consider the same questions above.

In [50]:
```
full_stats=educationUS.describe()
full_stats
```

Out[50]:

| | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out Of |
|---|---|---|---|---|---|---|---|
| **count** | 20.000000 | 20.000000 | 20.0 | 20.000000 | 20.000000 | 17.000000 | |
| **mean** | 1123.650000 | 1.500000 | 1.0 | 55069.650000 | 61242.250000 | 50909.941176 | 581 |
| **std** | 1158.037781 | 0.512989 | 0.0 | 26280.649121 | 22309.889931 | 25127.813801 | 209 |
| **min** | 72.000000 | 1.000000 | 1.0 | 24153.000000 | 34008.000000 | 23345.000000 | 340 |
| **25%** | 274.250000 | 1.000000 | 1.0 | 30722.750000 | 41358.500000 | 31091.000000 | 410 |
| **50%** | 652.000000 | 1.500000 | 1.0 | 45550.500000 | 53126.500000 | 43599.000000 | 477 |
| **75%** | 1601.750000 | 2.000000 | 1.0 | 87339.000000 | 87339.000000 | 63696.000000 | 710 |
| **max** | 4351.000000 | 2.000000 | 1.0 | 91290.000000 | 91290.000000 | 91290.000000 | 912 |

In [51]:
```
min_values=full_stats.loc["min"]
max_values=full_stats.loc["max"]
mean_values=full_stats.loc["mean"]
median_values=full_stats.loc["50%"] # 50th percentile
q1=full_stats.loc["25%"] #25th percentile
q3=full_stats.loc["75%"] #75th percentile
std_dev=full_stats.loc["std"] #standard deviation

"Minimum Values:\n", min_values
"Maximum Values:\n", max_values
"Mean Values:\n", mean_values
"Median Values:\n", median_values
"25th Percentile (Q1):\n", q1
"75th Percentile (Q3):\n", q3
"Standard Deviation:\n", std_dev
```

Out[51]:
```
('Standard Deviation:\n',
 Degree Graduate (4-Years)          1158.037781
 Hbcu Status                           0.512989
 Degree Granting                       0.000000
 In-State On-Campus Cost           26280.649121
 Out-Of-State On-Campus Cost       22309.889931
 In-State Off-Campus Cost          25127.813801
 Out-Of-State Off-Campus Cost      20990.146743
 Full-Time Retention Rate             11.504690
 Total Black Enrollment               39.462040
 Black Undergraduate Enrollment       39.901721
 Graduation Rate - Men                27.075479
 Graduation Rate - Black              25.030718
 Grad Rate - Black (6 Years)          25.070165
 Name: std, dtype: float64)
```

In [52]:
```
skewness=(mean_values-median_values).apply(
    lambda x: "Skewed Right" if x > 0 else "Skewed Left" if x < 0 else "Symmetric"
)

"Skewness Observations:\n", skewness
```

Out[52]:
```
('Skewness Observations:\n',
 Degree Graduate (4-Years)          Skewed Right
 Hbcu Status                           Symmetric
 Degree Granting                       Symmetric
 In-State On-Campus Cost            Skewed Right
 Out-Of-State On-Campus Cost        Skewed Right
 In-State Off-Campus Cost           Skewed Right
 Out-Of-State Off-Campus Cost       Skewed Right
 Full-Time Retention Rate            Skewed Left
 Total Black Enrollment             Skewed Right
 Black Undergraduate Enrollment     Skewed Right
 Graduation Rate - Men              Skewed Right
 Graduation Rate - Black            Skewed Right
 Grad Rate - Black (6 Years)        Skewed Right
 dtype: object)
```

In [53]:
```
#subset statistics
subset_stats=HBCU_subset.describe()
subset_stats
```

Out[53]:

| | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-S Off-Cam |
|---|---|---|---|---|---|---|---|
| **count** | 10.000000 | 10.0 | 10.0 | 10.000000 | 10.000000 | 10.00000 | 10.000 |
| **mean** | 335.000000 | 1.0 | 1.0 | 37811.200000 | 42472.600000 | 40607.20000 | 45268.600 |
| **std** | 205.948753 | 0.0 | 0.0 | 11090.153529 | 6659.180216 | 14761.03505 | 10497.979 |
| **min** | 155.000000 | 1.0 | 1.0 | 24153.000000 | 34008.000000 | 23345.00000 | 34008.000 |
| **25%** | 210.500000 | 1.0 | 1.0 | 27650.000000 | 37796.000000 | 29026.25000 | 38503.250 |
| **50%** | 290.000000 | 1.0 | 1.0 | 37371.000000 | 40963.000000 | 38293.50000 | 42313.000 |
| **75%** | 355.500000 | 1.0 | 1.0 | 45399.500000 | 45399.500000 | 47598.00000 | 47598.000 |
| **max** | 869.000000 | 1.0 | 1.0 | 54630.000000 | 54630.000000 | 63696.00000 | 63696.000 |

## Correlation:

Correlation is a statistic that measures the degree to which two variables move in relation to each other. A positive correlation indicates the extent to which those variables increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases. Correction among multiple variables can be represented in the form of a matrix. This allows us to see which pairs have the high correlations. correlation Correlation is a mutual relationship or connection between two or more things. It takes a value between (+1) and (-1) One important note here; Correlation can be created between integer values, so columns come with string values will not be included.

In [54]:
```
# Create correlation matrix
# Syntax: variableCorr = DataFrame.corr()

# Now call the correlation variable to see the correlation matrix.
```

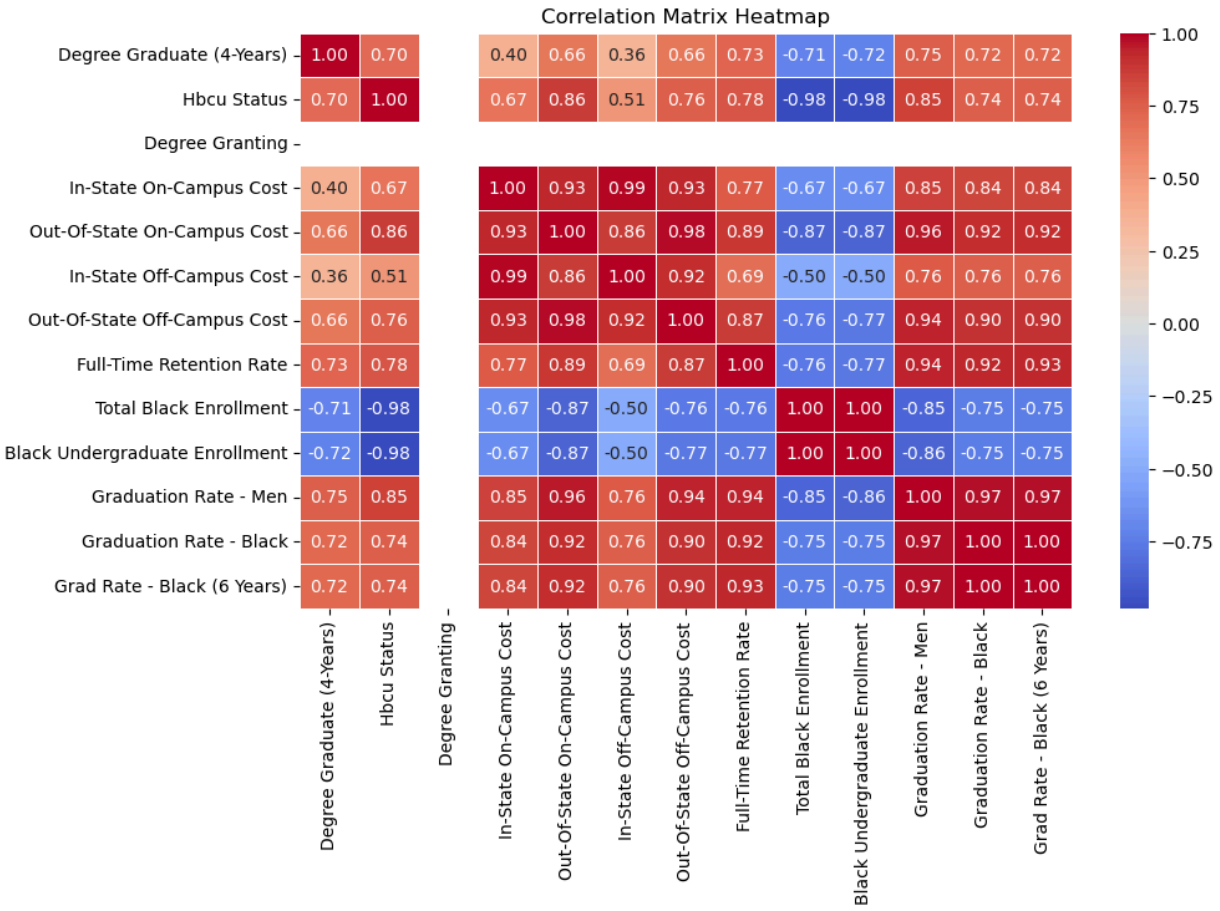
In [55]:
```
# select only numeric columns
numeric_df=educationUS.select_dtypes(include=['number'])

variableCorr=numeric_df.corr()

variableCorr
```

Out[55]:

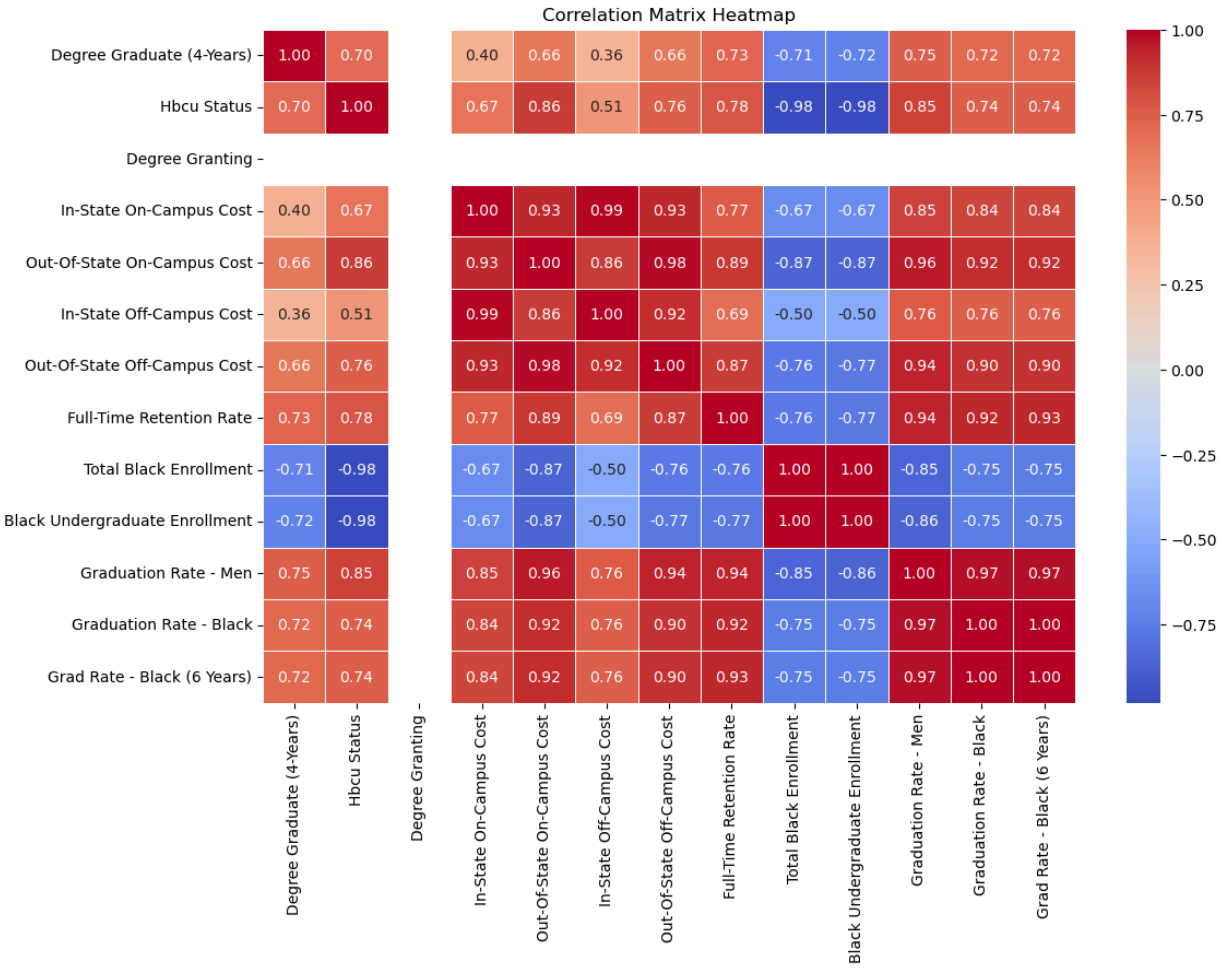
| | Degree Graduate (4-Years) | Hbcu Status | Degree Granting | In-State On-Campus Cost | Out-Of-State On-Campus Cost | In-State Off-Campus Cost | Out-Of-State Off-Campus Cost |
|---|---|---|---|---|---|---|---|
| **Degree Graduate (4-Years)** | 1.000000 | 0.698715 | NaN | 0.395424 | 0.656578 | 0.356774 | 0.658603 |
| **Hbcu Status** | 0.698715 | 1.000000 | NaN | 0.673758 | 0.863171 | 0.505142 | 0.757353 |
| **Degree Granting** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **In-State On-Campus Cost** | 0.395424 | 0.673758 | NaN | 1.000000 | 0.930907 | 0.987503 | 0.931170 |
| **Out-Of-State On-Campus Cost** | 0.656578 | 0.863171 | NaN | 0.930907 | 1.000000 | 0.863259 | 0.982362 |
| **In-State Off-Campus Cost** | 0.356774 | 0.505142 | NaN | 0.987503 | 0.863259 | 1.000000 | 0.915240 |
| **Out-Of-State Off-Campus Cost** | 0.658603 | 0.757353 | NaN | 0.931170 | 0.982362 | 0.915240 | 1.000000 |
| **Full-Time Retention Rate** | 0.729074 | 0.775859 | NaN | 0.765349 | 0.889979 | 0.687517 | 0.870394 |
| **Total Black Enrollment** | -0.714090 | -0.980167 | NaN | -0.667701 | -0.865222 | -0.498250 | -0.759893 |
| **Black Undergraduate Enrollment** | -0.724120 | -0.979651 | NaN | -0.667495 | -0.866920 | -0.501670 | -0.765809 |
| **Graduation Rate - Men** | 0.754088 | 0.854493 | NaN | 0.845687 | 0.960776 | 0.762753 | 0.940141 |
| **Graduation Rate - Black** | 0.719994 | 0.741897 | NaN | 0.837633 | 0.921535 | 0.756274 | 0.895259 |
| **Grad Rate - Black (6 Years)** | 0.720530 | 0.742776 | NaN | 0.837378 | 0.921919 | 0.755520 | 0.895276 |

In [56]:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
sns.heatmap(variableCorr,annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Matrix Heatmap")
plt.show()
```




In [57]:
```
# Fix the overlap issue with this heat map

plt.figure(figsize=(12,8))
sns.heatmap(variableCorr, annot=True, linewidths=0.5, cmap='coolwarm', fmt=".2f",
plt.title("Correlation Matrix Heatmap")
plt.show()
```




## Observations of the Correlation Matrix

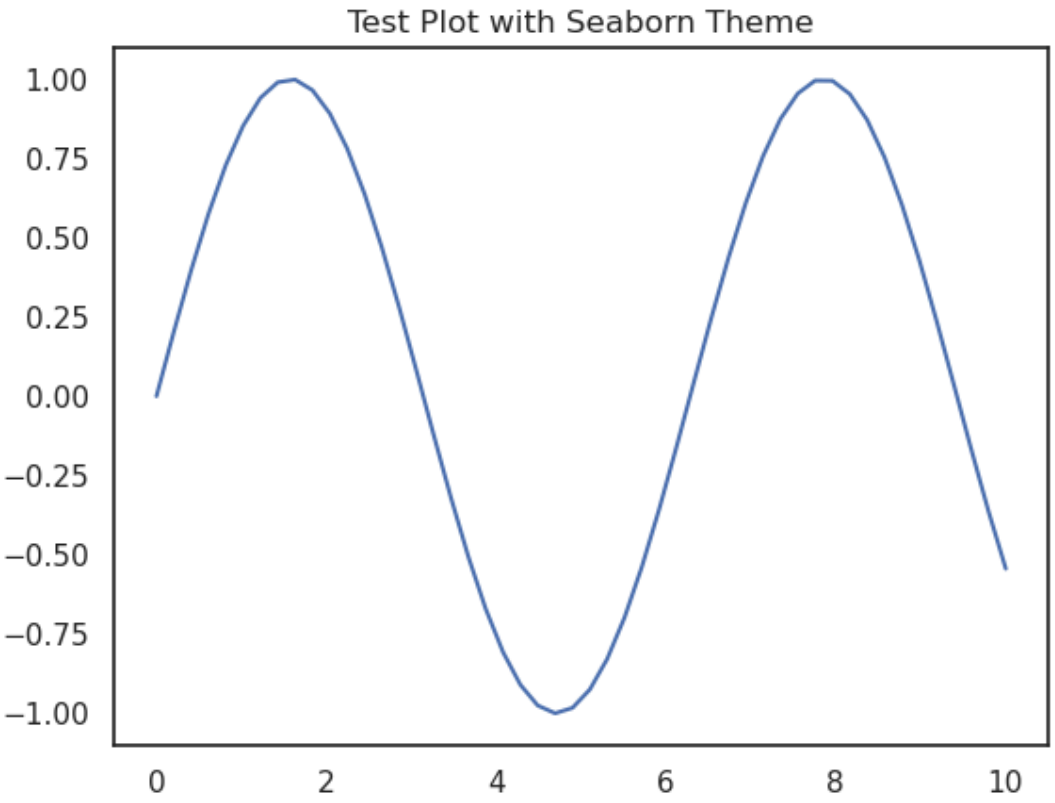
Correlation matrices can be viewed in a visualization or a visual table that shows the relative relationship between the variables using color while stating their values. We will use a color map (cmap) with a high contrast to see those that correlate by color. Remember that a correlation matrix is a square that is a mirror image across the diagonal. This means the bottom half of the matric looks exactly like the top half of the matrix. To minimize the values to view, let's use the `triu` argument to view just the lower half of the correlation matrix.

In [58]:
```
# Set seaborn themes
sns.set_theme(style='white')
sns.color_palette('viridis', as_cmap=True)
```

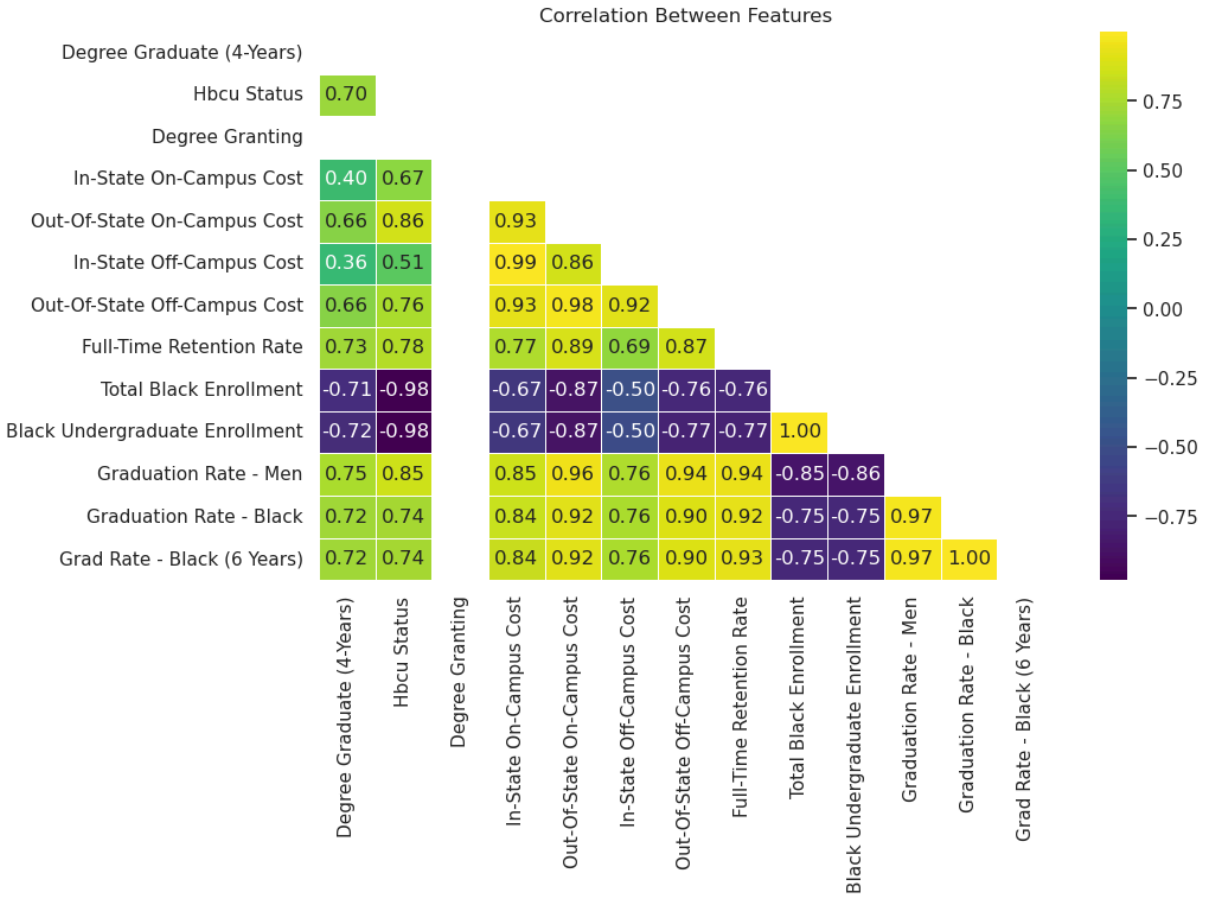
Out[58]:


In [59]:
```
# sample data
x=np.linspace(0,10,50)
y=np.sin(x)
```

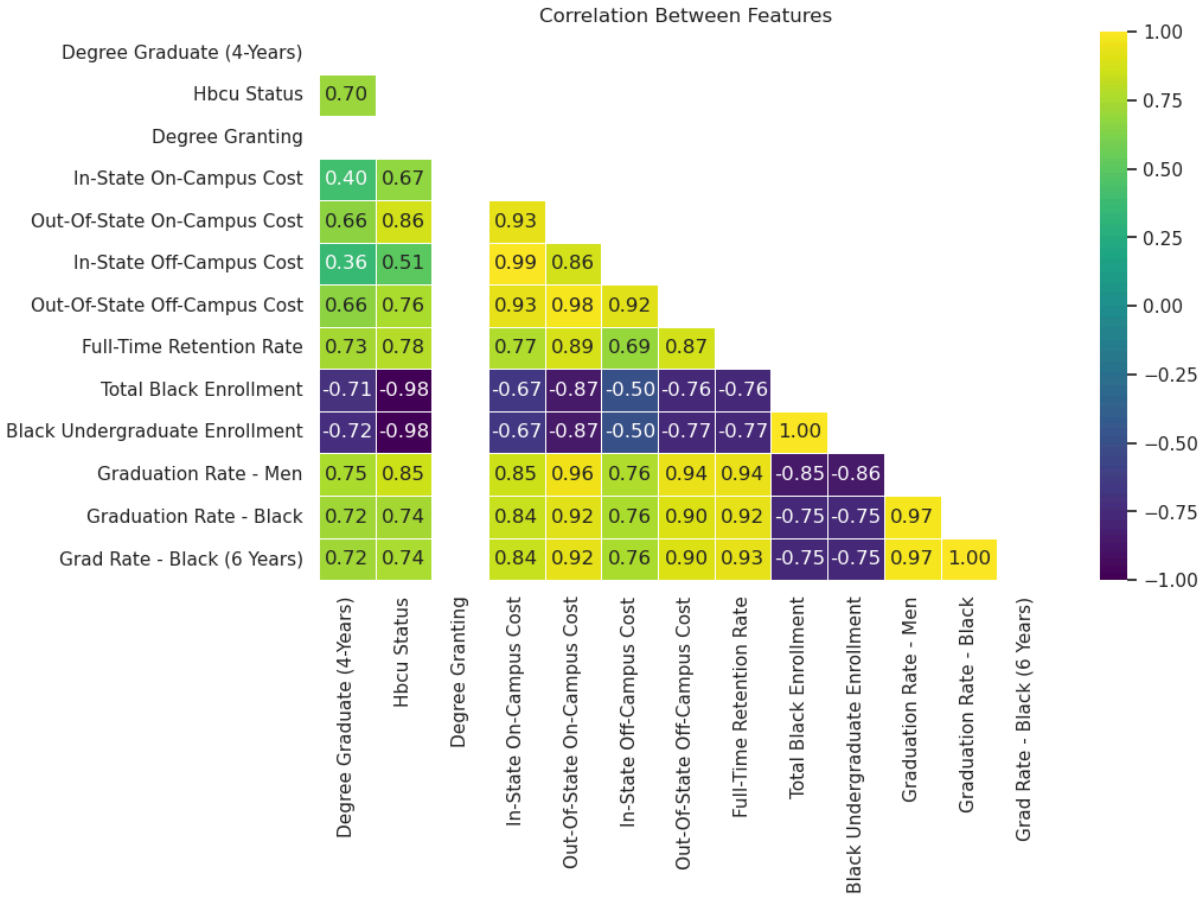
```
#test plot
sns.lineplot(x=x,y=y)
plt.title("Test Plot with Seaborn Theme")
plt.show()
```




In [60]:
```
# To get a correlation matrix
# Ploting the heat map
# corr: give the correlation matrix
# cmap: color code used for plotting
# vmax: gives a maximum range of values for the chart
# vmin: gives a minimum range of values for the chart
# annot: prints the correlation values in the chart
# annot_kws={"size": 12}): Sets the font size of the annotation

# Create the plot
plt.figure(figsize=(10,6))
matrix = variableCorr
mask = np.triu(np.ones_like(matrix, dtype=float))
sns.heatmap(variableCorr,
            annot=True,
            linewidths=.5,
            cmap='viridis',
            fmt= '.2f',
            mask=mask)

# Specify the name of the plot
plt.title('Correlation Between Features')
plt.show()
```




In [61]:
```
#use numeric data for correlation
numeric_df=educationUS.select_dtypes(include=['number'])
variableCorr=numeric_df.corr()

plt.figure(figsize=(10,6))

mask=np.triu(np.ones_like(variableCorr, dtype=bool))

sns.heatmap(variableCorr,
            annot=True,
            linewidths=0.5,
            cmap='viridis',
            fmt='.2f',
            mask=mask,
            vmax=1, vmin=-1,
            annot_kws={"size": 12})

plt.title('Correlation Between Features')

plt.show()
```




## Observations

- Are all the values the same color? This is called multicolinearity and indicates there are multiple independent variables that each have a strong relationship on each other. For instance if you are examining crime data categories such as robbery may also correlate to vehicular theft as the assailant was charged with both crimes. While they are independent crimes, they often occur together indicating a relationship. Multicolinear relationships complicate feature engineering for machine learning models and may need to have their dimensionality reduced (dropping columns or further subsets) to make sure the model trains well for those specific variables.
- Are specific variables correlated higher than others?
- Are there negative correlations indicating an inverse relationship in the variables? This indicates that as one variable is increasing, the other variable is decreasing. Negative correlations can be high (close to -1) or low (close to 0).
- Remember that correlation does not equal causation. Be careful with your wording when establishing relationships between the variables.
- Are there variables that lack correlation to any other variable? These are variables that may not be needed in the analysis and can be used to reduce the dimensionality of the data.

## Answers:

Multicollinearity exists when observing tuition and graduation-related rates. Both retention and graduation rates are strongly correlated (interdependent). There are negative correlations between Black student enrollment and retention rates, indicating a systemic issue that requires further investigation. Other variables such as "Unnamed: 15" and "Two or More Races-Men", may not be useful due to low correlation.

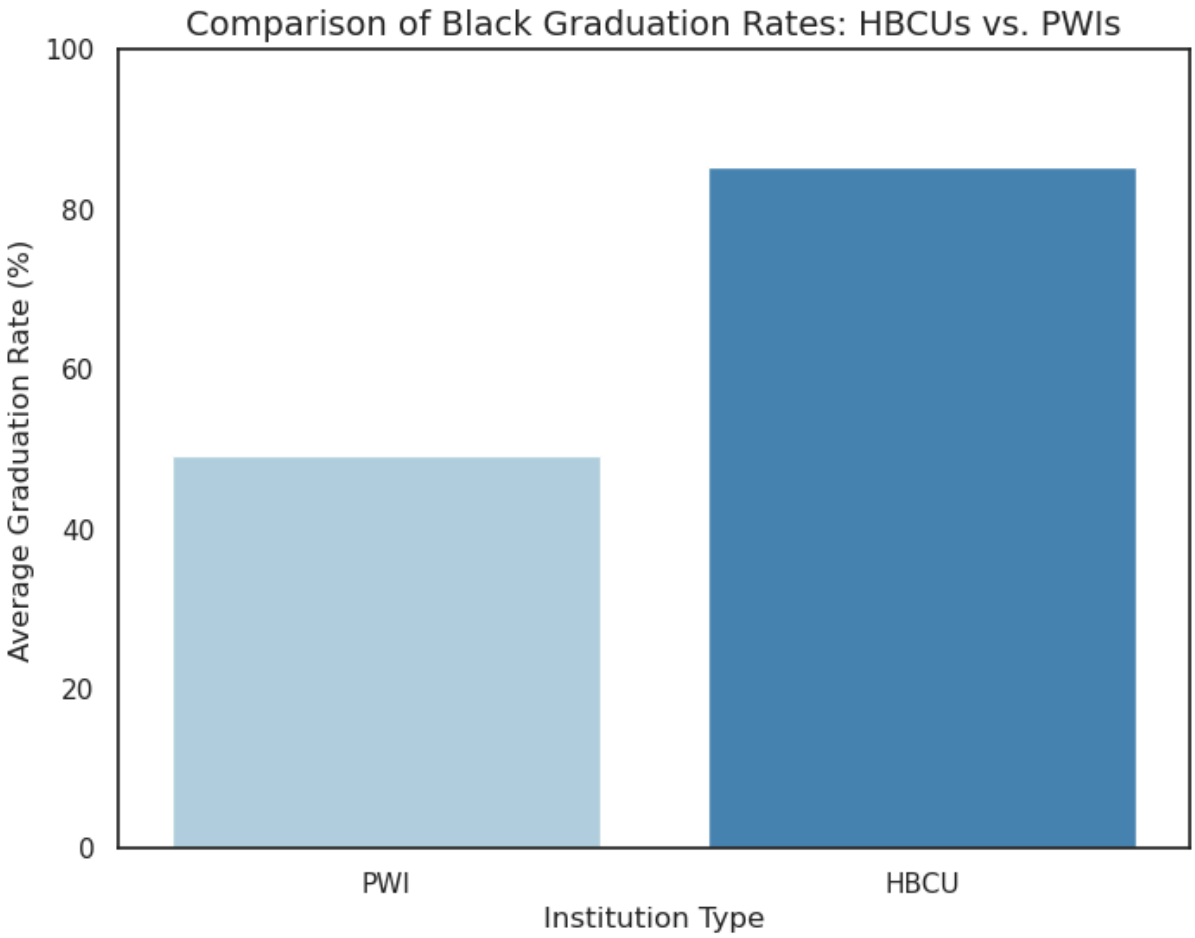
Retention rates and graduation rates are highly correlated (+0.93), suggesting that improving student retention directly impacts graduation outcomes. The negative correlation (-0.72) between Black student enrollment and graduation rates indicates that schools with higher Black enrollment tend to have lower graduation rates, which requires further investigation.

In [62]:
```
educationUS = educationUS.copy()
```

In [63]:
```
educationUS.columns
```

Out[63]:
```
Index(['Institution Name', 'Degree Graduate (4-Years)', 'Hbcu Status',
       'Degree Granting', 'In-State On-Campus Cost',
       'Out-Of-State On-Campus Cost', 'In-State Off-Campus Cost',
       'Out-Of-State Off-Campus Cost', 'Full-Time Retention Rate',
       'Total Black Enrollment', 'Black Undergraduate Enrollment',
       'Graduation Rate - Men', 'Graduation Rate - Black',
       'Grad Rate - Black (6 Years)', 'State', 'State Abbrev'],
      dtype='object')
```

In [64]:
```
import matplotlib.pyplot as plt
import seaborn as sns

hbcu_vs_pwi=educationUS.groupby("Hbcu Status")["Graduation Rate - Black"].mean()

labels=["PWI", "HBCU"]

plt.figure(figsize=(8,6))
sns.barplot(x=labels, y=hbcu_vs_pwi.values, palette="Blues")

plt.title("Comparison of Black Graduation Rates: HBCUs vs. PWIs", fontsize=14)
plt.ylabel("Average Graduation Rate (%)", fontsize=12)
plt.xlabel("Institution Type", fontsize=12)
plt.ylim(0,100)

plt.show()
```



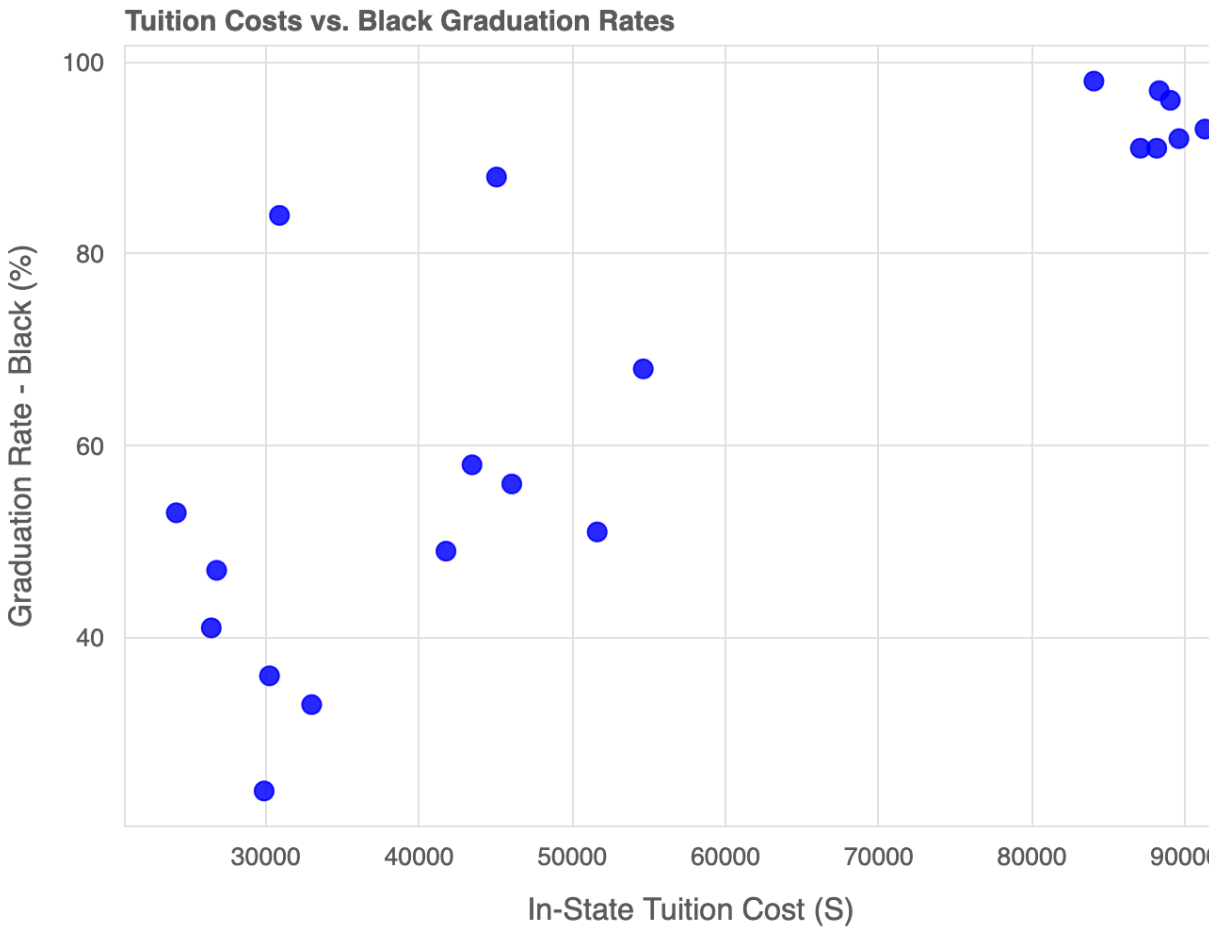

# Key Insights

1. The percentage of Black students at HBCUs is higher than at PWIs. However, the rentention rates at HBCUs tend to be lower. This suggests that HBCUs attract Black students but they may struggle with resources related to financial aid and academic support to retain them.
2. The correlation matrix shows a positive relationship between tuition costs and graduation rates. PWIs have higher tuition fees than HBCUs. Higher tuition rates can be linked to better academic resources and a stronger institutional infrastructure.
3. There is a negative relationship between Black student enrollment and graduation rates. This could be financial struggles, high dropout rates, less funding for HBCUs in addition to other social and academic barriers.

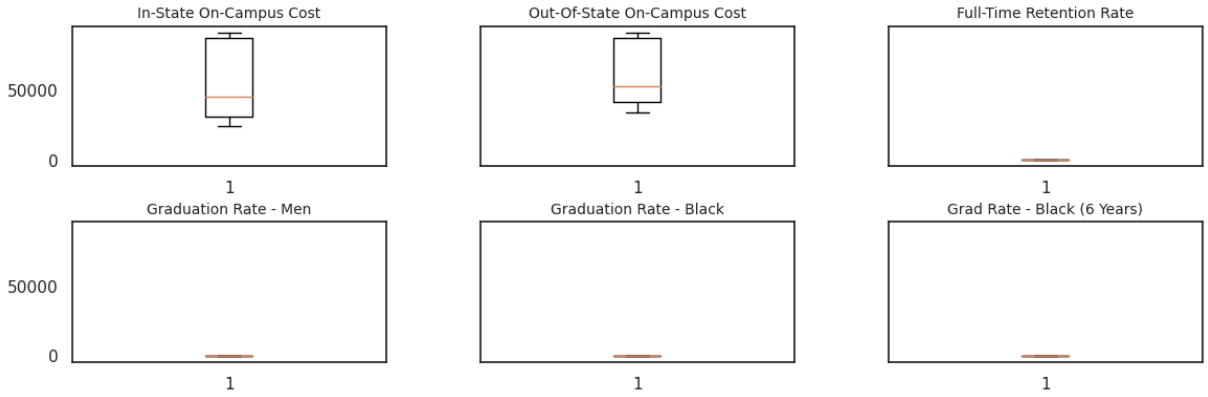
In [65]:
```
# Set Bokeh themes
# bokeh.palettes.Viridis256
import bokeh as bk
from bokeh.layouts import column
from bokeh.plotting import figure, output_file, show, curdoc
from bokeh.io import output_notebook, push_notebook, show
from bokeh.models import Scatter, ColumnDataSource, Div, RangeSlider, Spinner, C
# apply theme to current document
curdoc().theme = 'light_minimal'
```

In [66]:
```
source= ColumnDataSource(educationUS)

output_notebook()

p=figure(
    title="Tuition Costs vs. Black Graduation Rates",
    x_axis_label="In-State Tuition Cost (S)",
    y_axis_label="Graduation Rate - Black (%)",
    tools="pan, wheel_zoom, box_zoom, reset",
    width=700,
    height=500
)

# Add scatter points
p.circle(
    x="In-State On-Campus Cost",
    y="Graduation Rate - Black",
    source=source,
    size=10,
    color="blue",
    alpha=0.6
)

p.circle(
    x="In-State On-Campus Cost",
    y="Graduation Rate - Black",
    source=source,
    size=10,
    color="blue",
    alpha=0.6
)

show(p)
```

BokehJS 3.1.1 successfully loaded.




In [67]:
```
columns=[
    "In-State On-Campus Cost",
    "Out-Of-State On-Campus Cost",
    "Full-Time Retention Rate",
    "Graduation Rate - Men",
    "Graduation Rate - Black",
    "Grad Rate - Black (6 Years)"
]

fig, axs=plt.subplots(nrows=2, ncols=3, figsize=(12,8), sharey=True)

axs=axs.flatten()

for i, col in enumerate(columns):
    axs[i].boxplot(educationUS[col].dropna())
    axs[i].set_title(col, fontsize=10)

fig.subplots_adjust(left=0.08, right=0.98, bottom=0.5, top=0.9, hspace=0.4, wspa

plt.show()
```

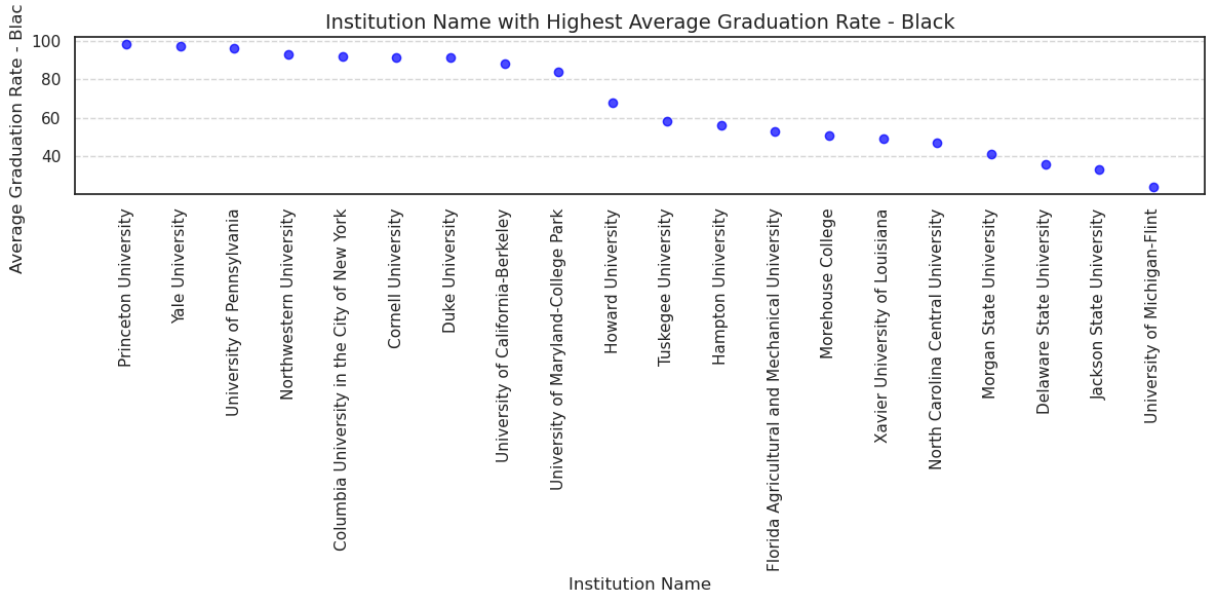



In [68]:
```
boxplot_columns=[
    "In-State On-Campus Cost",
    "Out-of-State On-Campus Cost",
    "Full-Time Retention Rate",
    "Graduation Rate - Men",
    "Graduation Rate - Black",
    "Grad Rate - Black (6 Years)"
]

boxplot_summary={}

for col in boxplot_columns:
    if col in educationUS.columns:
        boxplot_summary[col]={
            "Minimum": educationUS[col].min(),
            "Maximum": educationUS[col].max(),
            "Median": educationUS[col].median(),
            "Mean": educationUS[col].mean()
        }

boxplot_summary_df=pd.DataFrame(boxplot_summary).T

from IPython.display import display
display(boxplot_summary_df)
```

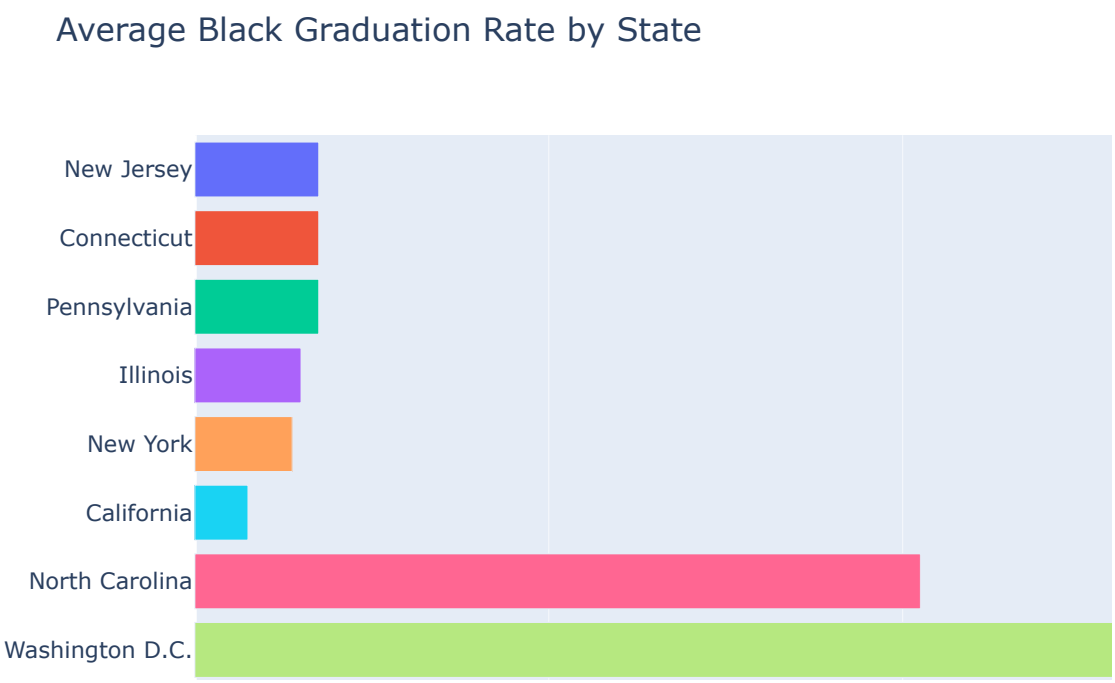
| | Minimum | Maximum | Median | Mean |
|---|---|---|---|---|
| **In-State On-Campus Cost** | 24153.0 | 91290.0 | 45550.5 | 55069.65 |
| **Full-Time Retention Rate** | 68.0 | 99.0 | 88.0 | 86.40 |
| **Graduation Rate - Men** | 25.0 | 96.0 | 58.0 | 66.15 |
| **Graduation Rate - Black** | 24.0 | 98.0 | 63.0 | 67.30 |
| **Grad Rate - Black (6 Years)** | 24.0 | 98.0 | 63.0 | 67.25 |

In [69]:
```
variable1 = "Institution Name"
variable2 = "Graduation Rate - Black"

averageVariable1 = educationUS.groupby(variable1)[variable2].mean()

maxVariable1 = averageVariable1.sort_values(ascending=False).head(50)

plt.figure(figsize=(12,6))
plt.scatter(maxVariable1.index, maxVariable1.values, color="blue", alpha=0.7)
```

```
plt.xlabel(variable1, fontsize=12)
plt.ylabel(f"Average {variable2} (%)", fontsize=12)
plt.title(f"{variable1} with Highest Average {variable2}", fontsize=14)
plt.xticks(rotation=90)
plt.grid(axis='y', linestyle="--", alpha=0.7)
plt.tight_layout()

plt.show()
```




In [70]:
```
import plotly.express as px

grouping_variable = "State"
numeric_variable = "Graduation Rate - Black"
sorting_variable = "Graduation Rate - Black"

numeric_cols = educationUS.select_dtypes(include=["number"])

variable_avg = educationUS.groupby(grouping_variable) [numeric_cols.columns].mea
variable_avg = variable_avg.sort_values(by=sorting_variable, ascending=False)

fig = px.bar(variable_avg,
             x="Total Black Enrollment",
             y=variable_avg.index,
             color=variable_avg.index,
             orientation='h',
             height=800,
             title='Average Black Graduation Rate by State')

fig.show()
```




## Observations: Boxplot

- Instituitions with lower Black graduation rates may indicate financial or academic challenges.
- Schools with high tuition but low graduation rates may lack student support systems.
- Tuition costs and graduation rates show significant outliers, especially in the upper range.
- Some institutions have exceptionally high tuition or low graduation rates.
- Graduation rates have a narrow spread, meaning most schools have similar rates.

- Tuition costs have a wider spread, showing large differences between institutions.

## Observations: Scatter Plot

- PWIs have higher tuition and graduation rates.
- HBCUs have lower tuition and graduation rates.
- Schools with higher retention rates also have higher graduation rates.

## Observations: Bar Chart

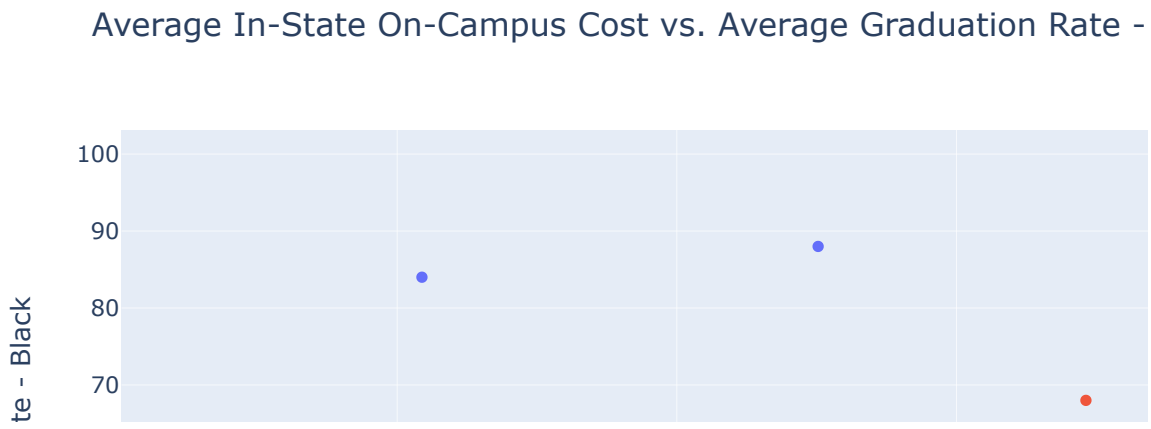
- States such as California and New York have higher Black graduation rates. Institutions in these states may have better funding, financial aid, or student support systems.
- States with lower Black graduation rates can be due to having a lack of financial aid opportunities, higher student loan debt, and institutional disparities in student resources.
- HBCU states show lower graduation rates.

## Primary Research Question

How does socioeconomic status affect the graduation rates of Black male students at HBCUs and PWIs?

The visualizations provide strong evidence that socioeconomic status significantly affects Black male graduation rates at HBCUs and PWIs The box plot shows graduation rate variability and outliers between HBCUs and PWIs. The scatter plot highlights the relationship between tuition costs and Black male graduation rates. The bar chart shows state-level differences in Black graduation rates, linking policies and funding to student success.

In [71]:
```
x_variable = "In-State On-Campus Cost"
y_variable = "Graduation Rate - Black"
color_variable = "Hbcu Status"

fig=px.scatter(educationUS,
               x=x_variable,
               y=y_variable,
               title=f"Average {x_variable} vs. Average {y_variable} by {color_v
               color=educationUS[color_variable].astype(str),
                hover_data=["Institution Name"])

fig.show()
```




This scatter plot visualizes the relationship between in-state on-campus cost (tuition/fees) and Black graduation rates, while distinguishing between HBCUs and PWIs (color-coded as 1 and 2).

The x-axis (In-State On-Campus Cost) represents the cost of attending the institution. The y-axis (Graduation Rate - Black) represents the percentage of Black students who graduate. Each dot represents a different instiution, with color coding for HBCUs (red) and PWIs (blue).

PWIs (Blue dots) tend to have higher tuition costs ($60K-$90K range). HBCUs (Red dots) have lower tuition costs (mostly $20K-$50K range). Higher graduation rates are observed at PWIs, where many blue dots cluster above 80%. HBCUs show more variance in graduation rates, often below 60%.

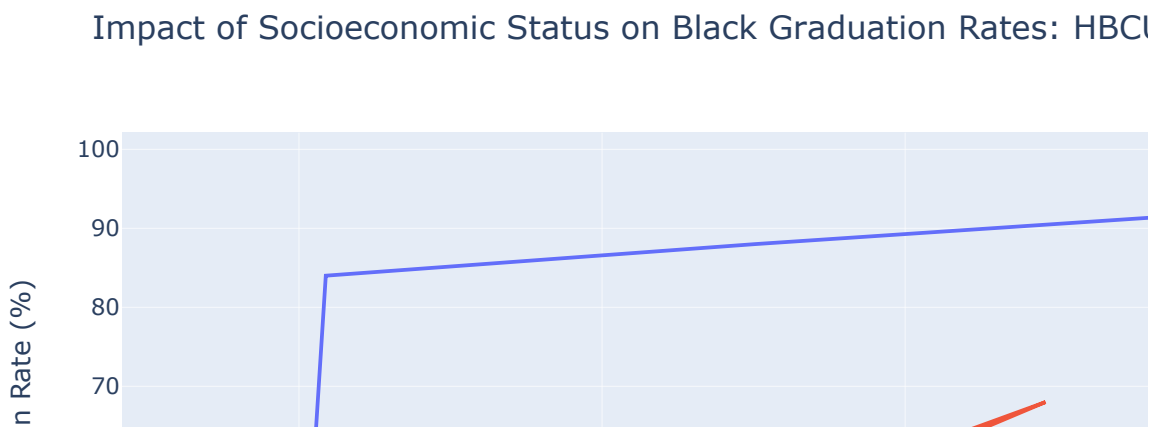
The visualization suggests instiutitions with higher tuition tend to have higher graduation rates. The lower tuition at HBCUs may indicate fewer instiutional resources, scholarships, or student support systems, impacting graduation outcomes. Socioeconomic factors like financial aid, student debt, and family income may be contributing to these disparities.

## Secondary Research Question

How does socioeconomic status affect the graduation rates of Black male students at Historically Black Colleges and Universities (HBCUs) compared to Predominantly White

Institutions (PWIs), and what factors contribute most to this disparity?

In [72]:
```
x_variable = "In-State On-Campus Cost"
y_variable = "Graduation Rate - Black"
color_variable = "Hbcu Status"

fig = px.line(educationUS,
              x=x_variable,
              y=y_variable,
              color=color_variable,
              title="Impact of Socioeconomic Status on Black Graduation Rates: H
              labels={"HBCU Status": "Institution Type",
                      "In-State On-Campus Cost": "Tuition Cost ($)",
                      "Graduation Rate - Black": "Black Graduation Rate (%)"})

fig.show()
```




The line chart represents the relationships between tuition costs and Black graduation rates at HBCUs (red) and PWIs (blue). The x-axis (Tuition Cost) shows in-state on-campus tuition, which serves as a proxy for socioeconomic status. The y-axis (Black Graduation Rate) represents how well Black students are graduating at different instiutions.

PWIs have a clearer upward trend. As tuition increases, Black graduation rates tend to be higher. HBCUs show more fluctuation: Even at similar tuition levels, their graduation rates do not follow a strong upward trend. Low-cost HBCUs still have lower graduation rates: This

suggests that financial barriers are not the only issue-instiutional resources, funding, and student support systems may play a role.

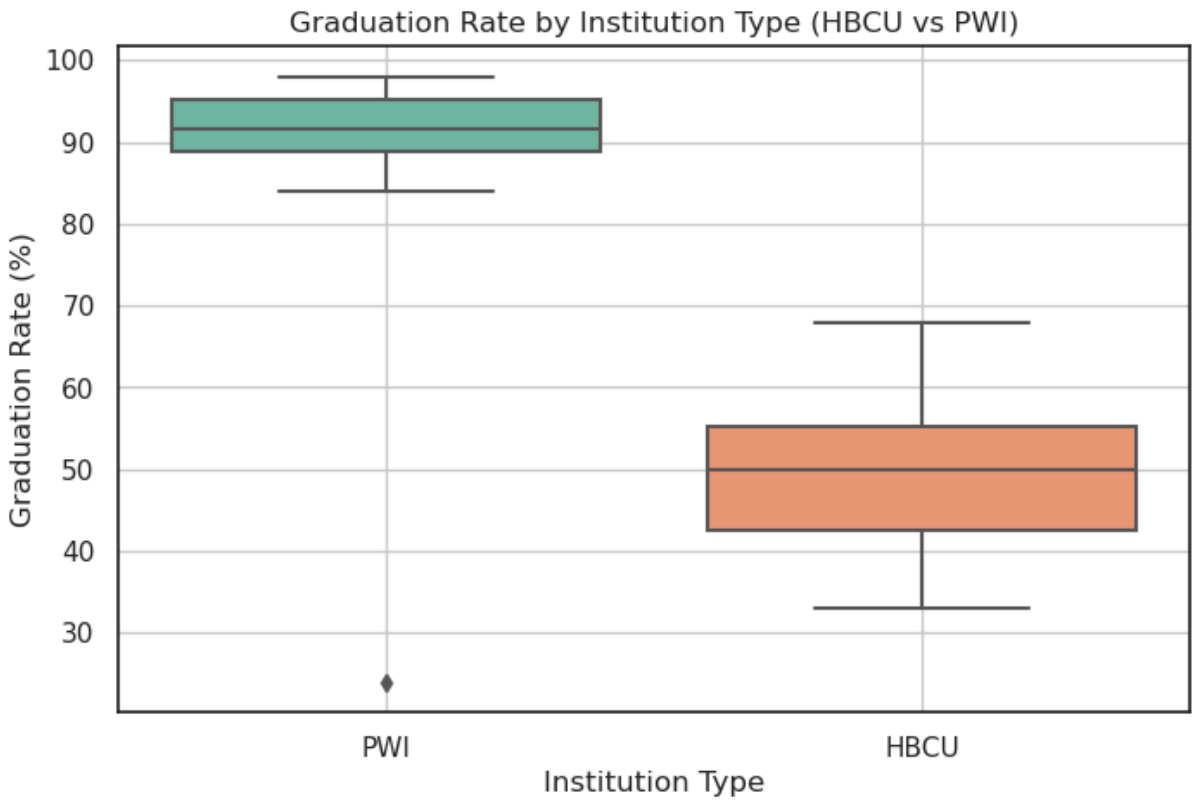
The visualization suggests that higher tuition at PWIs correlates with better graduation outcomes; likely due to greater resources. HBCUs, despite serving more Black students, struggle with retention and graduation. This hints at structural challenges such as lower funding, fewers academic support programs, and financial burdens on students.

In [73]:
```
educationUS.to_csv("educationUS.csv", index=False)

"CSV file saved successfully!"
```

Out[73]:
```
'CSV file saved successfully!'
```

In [85]:
```
educationUS['Institution_Type'] = educationUS['Hbcu Status'].apply(lambda x: 'HB
```

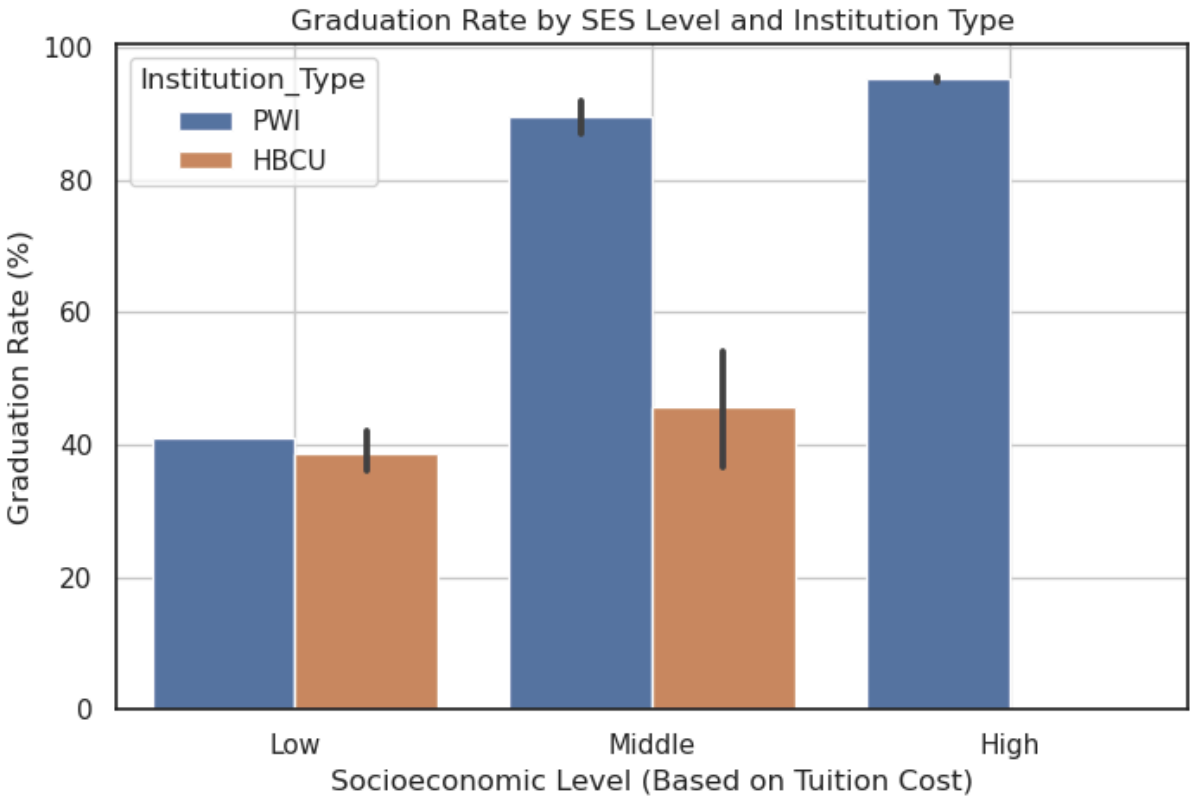
In [87]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.boxplot(x='Institution_Type', y='Graduation Rate - Black', data=educationUS,
plt.title('Graduation Rate by Institution Type (HBCU vs PWI)')
plt.xlabel('Institution Type')
plt.ylabel('Graduation Rate (%)')
plt.grid(True)
plt.show()
```




In [90]:
```
educationUS['SES_Level'] = pd.cut(
    educationUS['In-State On-Campus Cost'],
    bins=[0, 30000, 60000, educationUS['In-State On-Campus Cost'].max()],
```

```
    labels=['Low', 'Middle', 'High']
)
```

In [91]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='SES_Level', y='Graduation Rate - Men', hue='Institution_Type', da
plt.title('Graduation Rate by SES Level and Institution Type')
plt.xlabel('Socioeconomic Level (Based on Tuition Cost)')
plt.ylabel('Graduation Rate (%)')
plt.grid(True)
plt.show()
```




### Recommendations

- Provide targeted financial support to Black male students from low-income households.
- Expand academic support programs at PWIs modeled after HBCU student success programs.

### Insights

- HBCUs appear to maintain more consistent graduation rates for Black males regardless of SES.
- Socioeconomic status has a stronger impact on outcomes at PWIs.

## Conclusion

This project explored how socioeconomic status affects graduation outcomes for Black male students at HBCUs and PWIs. Through visual analysis, we saw that HBCUs generally provide

more consistent graduation outcomes across income levels, while PWIs show more variation, especially for students from lower-income backgrounds.

These findings highlight the importance of support systems and intersectionality in higher education. The recommendations above can help guide future policies and institutional improvements that better support Black male students.

In [ ]: