

WORLD UNIVERSITY RANKINGS 2023 ANALYSIS

The dataset likely includes information about universities from around the world and their respective rankings for the year 2023. Each row in the dataset probably represents a university, with columns representing various attributes such as the university name, country, ranking position, scores in different categories (e.g., teaching quality, research output, international outlook), and possibly other relevant factors.

IMPORT LIBRARY

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


Upload Csv File

```
data = pd.read_csv ("/content/16_World University Rankings 2023 Analysis.csv")
```

Data Reprocessing

- .head() head is used show to the By default = 5 rows in the dataset

```
data.head()
```




	University Rank	Name of University	Location	No of student	No of student per staff	International Student	OverAll Score	Teaching Score	Research Score	
0	1	University of Oxford	United Kingdom	20,965	10.6	42%	96.4	92.3	99.7	
1	2	Harvard University	United States	21,887	9.6	25%	95.2	94.8	99.0	
2	3	University of Cambridge	United Kingdom	20,185	11.3	39%	94.8	90.9	99.5	
3	3	Stanford University	United States	16,164	7.1	24%	94.8	94.2	96.7	
4	5	Massachusetts Institute of Technology	United States	11,415	8.2	33%	94.2	90.7	93.6	

.tail()

tail is used to show last rows

```
data.tail()
```



	University Rank	Name of University	Location	No of student	No of student per staff	International Student	Female:Male Ratio	OverAll Score	Teaching Score	Research Score	Citations Score	Industry Income Score
2336	-	University of the West of Scotland	NaN	NaN	NaN	NaN	NaN	34.0–39.2	24.1	15.5	61.5	37.9
2337	-	University of Windsor	NaN	NaN	NaN	NaN	NaN	34.0–39.2	35.1	29.4	34.5	44.2
2338	-	University of Wolverhampton	NaN	NaN	NaN	NaN	NaN	34.0–39.2	18.2	14.3	68.8	37.3
2339	-	University of Wuppertal	NaN	NaN	NaN	NaN	NaN	34.0–39.2	26.4	26.7	52.8	52.1
2340	-	Xi'an Jiaotong-Liverpool University	NaN	NaN	NaN	NaN	NaN	34.0–39.2	17.8	14.8	68.2	38.2

.shape

It show the total no of rows & Column in the dataset

```
data.shape
```

```
(2341, 13)
```

```
.Columns
```

It show the no of each Column

```
data.columns
```

```
Index(['University Rank', 'Name of University', 'Location', 'No of student',
      'No of student per staff', 'International Student', 'Female:Male Ratio',
      'OverAll Score', 'Teaching Score', 'Research Score', 'Citations Score',
      'Industry Income Score', 'International Outlook Score'],
      dtype='object')
```

```
.dtypes
```

This Attribute show the data type of each column

```
data.dtypes
```

```
University Rank      object
Name of University   object
Location             object
No of student        object
No of student per staff  float64
International Student object
Female:Male Ratio     object
OverAll Score         object
Teaching Score        float64
Research Score         float64
Citations Score        float64
Industry Income Score float64
International Outlook Score float64
dtype: object
```

```
.unique()
```

In a column, It show the unique value of specific column.

```
data["Location"].unique()
```

```
array(['United Kingdom', 'United States', 'Switzerland', nan, 'Canada',
      'Australia', 'Singapore', 'Japan', 'France', 'Sweden', 'China',
      'South Korea', 'Netherlands', 'Germany', 'Belgium', 'Finland',
      'Denmark', 'Austria', 'Norway', 'New Zealand', 'Spain', 'Italy',
      'Saudi Arabia', 'Luxembourg', 'Qatar', 'Brazil', 'Israel',
      'Ireland', 'Taiwan', 'India', 'United Arab Emirates',
      'Brunei Darussalam', 'Iceland', 'Lebanon', 'Philippines',
      'Portugal', 'Iran', 'Malaysia', 'Poland', 'Egypt', 'Turkey',
      'Greece', 'Vietnam', 'Algeria', 'Nigeria', 'Tanzania', 'Chile',
      'Pakistan', 'Ukraine', 'Romania', 'Czech Republic', 'South Africa',
      'Northern Cyprus', 'Hong Kong', 'Ethiopia', 'Jordan', 'Serbia',
      'Sri Lanka', 'Jamaica', 'Zambia', 'Iraq', 'Costa Rica', 'Cyprus',
      'Bangladesh', 'Mozambique', 'Colombia', 'Kenya', 'Namibia', 'Peru',
      'Latvia', 'Oman', 'Thailand', 'Lithuania', 'Slovenia', 'Uganda',
      'Malta', 'Nepal', 'Kazakhstan', 'Mexico', 'Botswana', 'Slovakia',
      'Ghana', 'Morocco', 'Georgia', 'Tunisia', 'Mauritius', 'Hungary',
      'Puerto Rico', 'Ecuador', 'Fiji', 'Croatia', 'Estonia', 'Zimbabwe',
      'Indonesia', 'Argentina', 'Bulgaria', 'Venezuela', 'Azerbaijan',
      'Cuba', 'Montenegro', 'Uzbekistan', 'Palestine', 'Kuwait',
      'Somalia', 'Libya', 'Moldova', 'Kyrgyzstan', 'Malawi', 'Paraguay',
      'Mongolia', 'Armenia', 'Sudan', 'Turkmenistan', 'Uruguay',
      'Albania', 'Cambodia', 'Kosovo'], dtype=object)
```

```
.nunique()
```

It will show the total no of unque value from whole data frame

```
data.nunique()
```


```
University Rank      162
Name of University   2233
Location             116
No of student        2136
No of student per staff  420
International Student    79
Female:Male Ratio      87
OverAll Score         160
Teaching Score         437
Research Score         477
Citations Score        820
```

```
Industry Income Score      391
International Outlook Score 694
dtype: int64
```

.describe()

It show the Count, mean , median etc

```
data.describe()
```




	No of student per staff	Teaching Score	Research Score	Citations Score	Industry Income Score	International Outlook Score
count	2208.000000	1799.000000	1799.000000	1799.000000	1799.000000	1799.000000
mean	19.000408	27.018010	23.016898	48.495887	47.104558	46.880378
std	12.132224	13.282243	16.763819	27.967185	15.093682	22.582401
min	0.400000	11.600000	7.400000	0.800000	36.900000	14.100000
25%	12.600000	18.000000	11.300000	23.100000	37.800000	27.900000
50%	16.600000	22.700000	17.000000	47.200000	40.500000	42.100000
75%	22.200000	31.850000	28.900000	72.350000	48.300000	62.100000
max	232.200000	94.800000	99.700000	100.000000	100.000000	99.700000

• .value counts

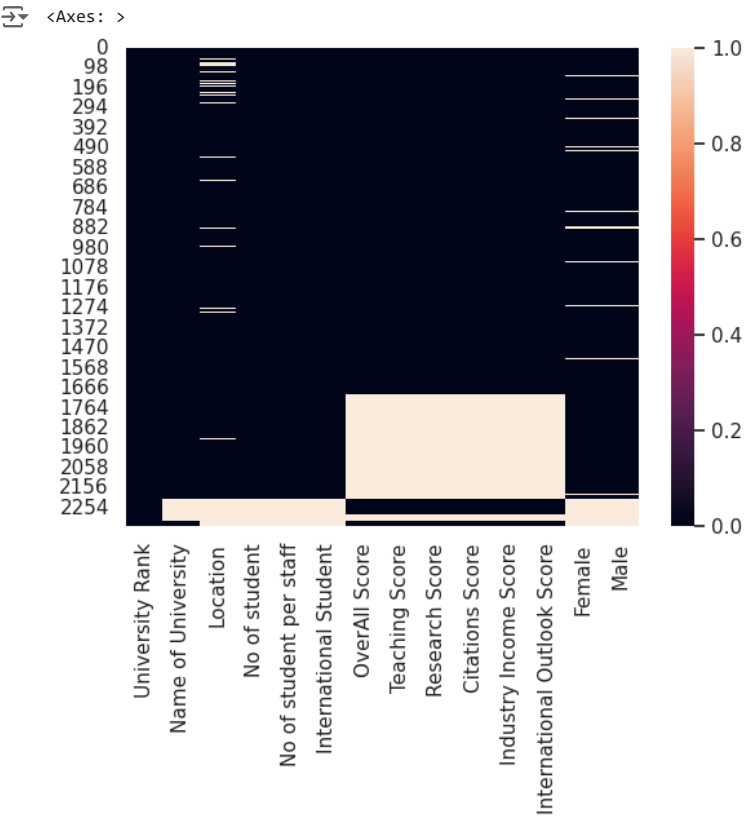
It Shows all the unique values with their count

```
data["Location"].value_counts()
```



Location	
United States	173
Japan	150
United Kingdom	149
India	91
China	82
...	
Puerto Rico	1
Mozambique	1
Mauritius	1
Namibia	1
Kosovo	1
Name: count, Length: 116, dtype: int64	

```
sns.heatmap(data.isnull())
```



`.isnull()`
It shows the how many null values

`data.isnull()`

<Axes: >

	University Rank	Name of University	Location	No of student	No of student per staff	International Student	Female:Male Ratio	OverAll Score	Teachir Scor
0	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	Fals
...	
2336	False	False	True	True	True	True	True	False	Fals
2337	False	False	True	True	True	True	True	False	Fals
2338	False	False	True	True	True	True	True	False	Fals
2339	False	False	True	True	True	True	True	False	Fals
2340	False	False	True	True	True	True	True	False	Fals

2341 rows x 13 columns

```
data['International Student'] = data['International Student'].str.replace('%', ' ')
data['International Student'] = pd.to_numeric(data['International Student'], errors='coerce')

data['International Student']= data['International Student']/100
```

```
import plotly.express as px

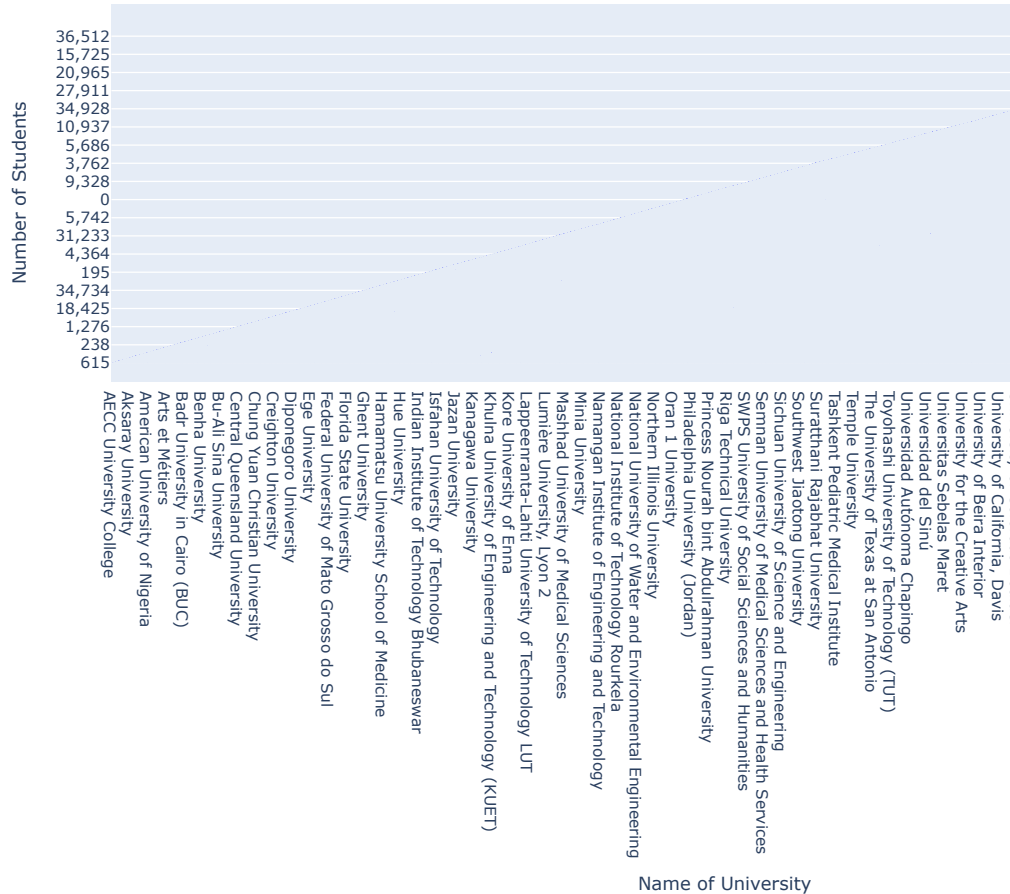
monthly_trends = data.groupby('Name of University')['No of student'].sum().reset_index()

fig_monthly_trends = px.bar(
    monthly_trends,
    x='Name of University',
    y='No of student',
    labels={'Name of University': 'Name of University', 'No of student': 'Number of Students'},
    title='Number of Students by each University',
    height=800
)

fig_monthly_trends.show()
```



Number of Students by each University



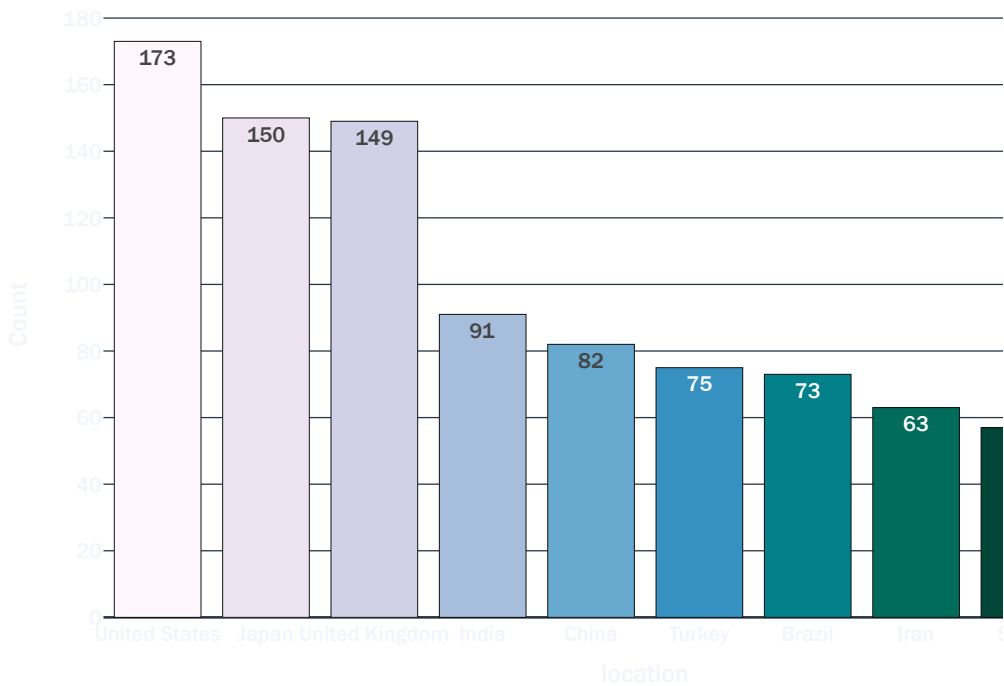
```
top10_job_title = data['Location'].value_counts()[:10]
# creation of bar plot
fig = px.bar(
    y=top10_job_title.values,
    x=top10_job_title.index,
    color=top10_job_title.index,
    color_discrete_sequence=px.colors.sequential.PuBuGn, # Set color palette
    text=top10_job_title.values,
    title='Top 10 Location ',
    template='plotly_dark'
)

fig.update_layout(
    title_text='Distribution of location',
    height=650,
    xaxis_title="location",
    yaxis_title="Count",
    font=dict(size=17, family="Franklin Gothic")
)

# Display the plot
fig.show()
```



Distribution of location



```
import plotly.express as px

# Remove any rows with null values in the 'Location' column
data_cleaned = data.dropna(subset=['Location'])

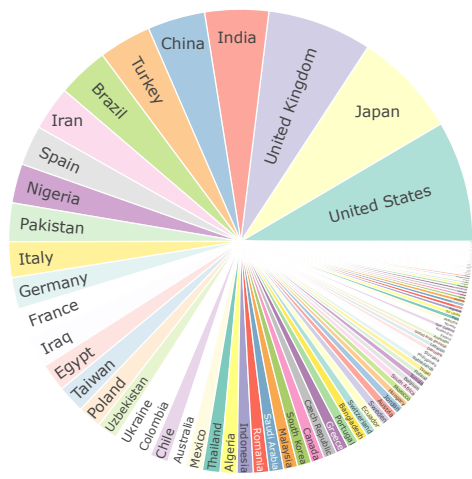
# Create a sunburst chart
fig2 = px.sunburst(
    data_cleaned,
    path=['Location'],
    color_discrete_sequence=px.colors.qualitative.Set3
)

# Update the layout with the desired title and height
fig2.update_layout(
    title_text='Distribution of Students by Location',
    height=500
)

# Display the sunburst chart
fig2.show()
```



Distribution of Students by Location

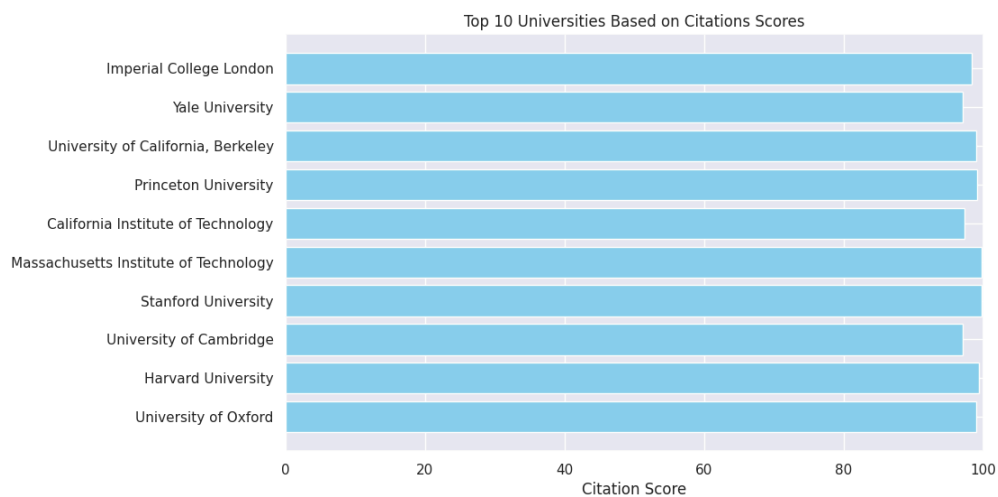


```

top_10_universities = data["Name of University"][:10]
top_10_scores = data["Citations Score"][:10]

# Create a bar chart for the top 10 universities
plt.figure(figsize=(10, 6))
plt.barh(top_10_universities[::-1], top_10_scores[::-1], color='skyblue')
plt.xlabel('Citation Score')
plt.title('Top 10 Universities Based on Citations Scores')
plt.xlim(0, 100) # Customize the score range as needed
plt.gca().invert_yaxis() # Reverse the order to show the top university at the top
plt.show()

```

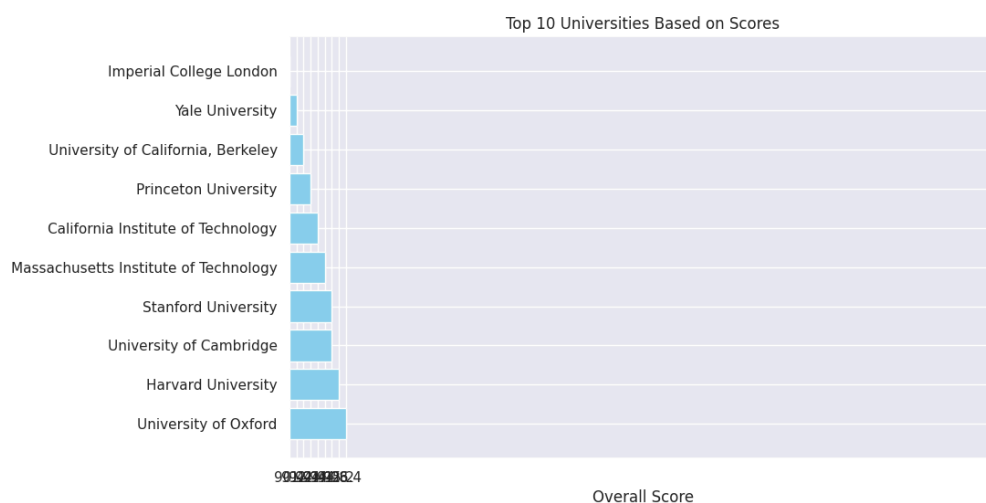


```

top_10_universities = data["Name of University"][:10]
top_10_scores = data["OverAll Score"][:10]

# Create a bar chart for the top 10 universities
plt.figure(figsize=(10, 6))
plt.barh(top_10_universities[::-1], top_10_scores[::-1], color='skyblue')
plt.xlabel('Overall Score')
plt.title('Top 10 Universities Based on Overall Scores')
plt.xlim(0, 100) # Customize the score range as needed
plt.gca().invert_yaxis() # Reverse the order to show the top university at the top
plt.show()

```

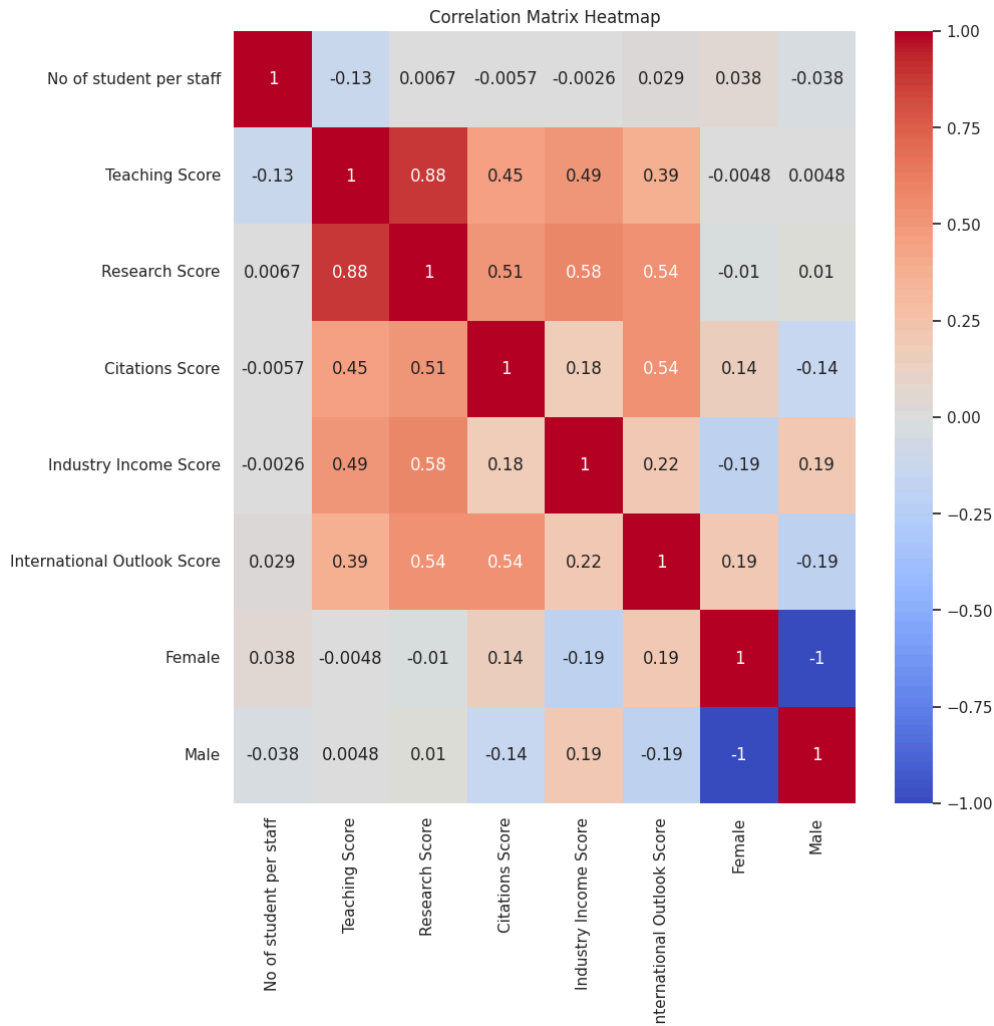


```
numerical_columns = data.select_dtypes(include=['float64', 'int64']).columns
```

```
if not numerical_columns.any():
    print("No numerical columns found for correlation calculation.")
else:
```

```
    # Calculate the correlation matrix
    Corr_Matrix = data[numerical_columns].corr()
```

```
    # Set up the figure and plot the heatmap
    plt.figure(figsize=(10, 10))
    sns.heatmap(Corr_Matrix, annot=True, cmap='coolwarm', center=0)
    plt.title('Correlation Matrix Heatmap')
    plt.show()
```



```
# splitting the dataset
```

```
X = data.drop(columns=['University Rank', 'Name of University', 'Female', 'OverAll Score', 'Teaching Score', 'Research Score', 'Citations Score', 'Male'])
y = data['Female']
```