



Minor Project Report

on

[University Ranking and Accreditation Analysis]

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

in partial fulfilment of the requirements for the award of degree of

Master of Computer Applications

Submitted By

Mohd Shah Wasif 12223538

Anmol Gupta 12223484

Laxman Rao 12223537

Supervised By

Rafiq Sir

LOVELY FACULTY OF TECHNOLOGY & SCIENCES

LOVELY PROFESSIONAL UNIVERSITY

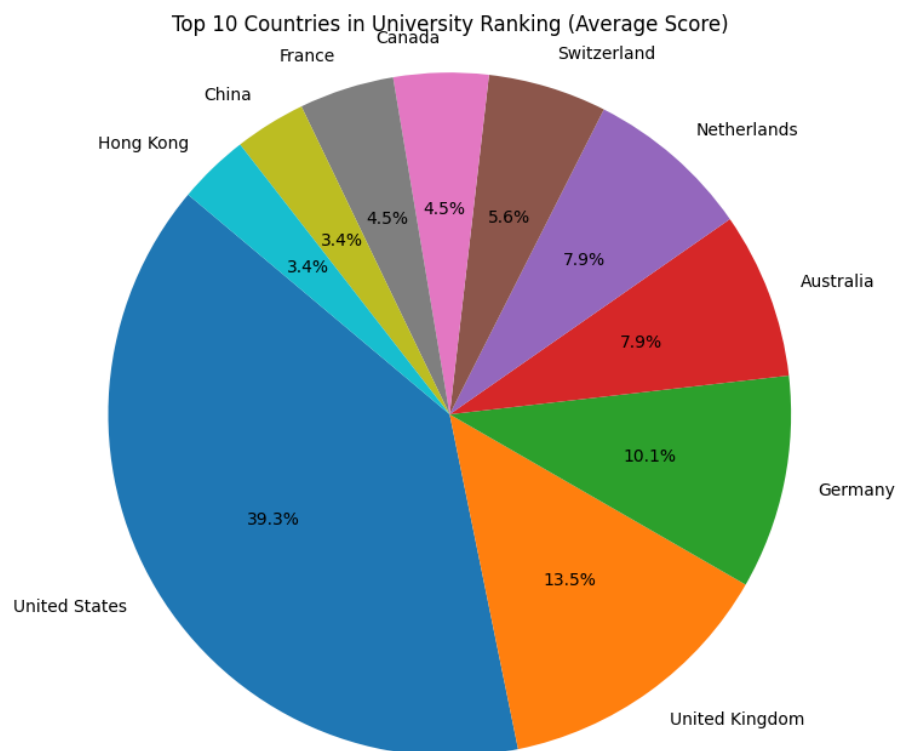
PUNJAB

[November 2023]

Table of Content		
<i>Sr. No.</i>	<i>Content</i>	<i>Page Index</i>
1	Introduction about project	1-3
2	Project Modules and its description	3-6
3	Coding	6-27
4	Screenshot output	27-28

Introduction

In this project we analyse the World University Rankings 2023. The Times Higher Education World University Rankings 2023 includes 1,799 universities across 104 countries and regions, making them the largest and most diverse university rankings to date.



This dataset includes the following 13 columns:

1. **University Rank:** The global ranking of the university.
2. **Name of University:** The name of the university.
3. **Location:** The country where the university is located.
4. **No of student:** The total number of students at the university.
5. **No of student per staff:** The ratio of students to staff at the university.
6. **International Student:** The percentage of students who are international.
7. **Female: Male Ratio:** The ratio of female to male students at the university.
8. **Overall Score:** The overall score of the university, presumably out of 100.
9. **Teaching Score:** The university's score for teaching, presumably out of 100.
10. **Research Score:** The university's score for research, presumably out of 100.
11. **Citations Score:** The university's score for citations, presumably out of 100.
12. **Industry Income Score:** The university's score for industry income, presumably out of 100.
13. **International Outlook Score:** The university's score for international outlook, presumably out of 100.

The goal of this project is to take a deep dive into the dataset and answer the following four questions:

Which country has the highest number of universities in the top 100 of the World University Rankings 2023

What is the university ranking distribution of the top 10 scoring Countries?

What is the correlation between each score and overall ranking? Which factor is the most important?

Project Module

1. **University Rank:** There are 162 unique university ranks in the dataset. The most frequent rank is 'Reporter', which appears 512 times.
2. **Name of University:** There are 2233 unique universities in the dataset. The most frequent university is the University of Oxford, which appears once.
3. **Location:** The universities are located in 116 unique locations. The most frequent location is the United States, which is the location of 173 universities in the dataset.

4. **No of student:** The number of students ranges from 17,080 to an unknown maximum value. The most frequent number of students is 17,080, which appears twice.
5. **No of student per staff:** The ratio of students to staff ranges from 0.4 to 232.2. The average ratio is approximately 19. The standard deviation is approximately 12.13, indicating a wide spread of the data.
6. **International Student:** There are 79 unique values for the percentage of international students. The most frequent value is 0%, which appears 336 times.
7. **Female:Male Ratio:** There are 87 unique values for the female-to-male ratio. The most frequent ratio is 58:42, which appears 95 times.
8. **OverAll Score:** There are 160 unique overall scores. The most frequent score is 18.4–24.3, which appears 305 times.
9. **Teaching Score:** The teaching score ranges from 11.6 to 94.8, with an average of approximately 27. The standard deviation is approximately 13.28, indicating a wide spread of the data.
10. **Research Score:** The research score ranges from 7.4 to 99.7, with an average of approximately 23. The standard deviation is approximately 16.76, indicating a wide spread of the data.
11. **Citations Score:** The citations score ranges from 0.8 to 100, with an average of approximately 48.5. The standard deviation is approximately 27.97, indicating a wide spread of the data.
12. **Industry Income Score:** The industry income score ranges from 36.9 to 100, with an average of approximately 47.1. The standard deviation is approximately 15.09, indicating a wide spread of the data.
13. **International Outlook Score:** The international outlook score ranges from 14.1 to 99.7, with an average of approximately 46.9. The standard deviation is approximately 22.58, indicating a wide spread of the data.

Code

```

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

university = pd.read_csv('World University Rankings 2023.csv')
university.head()
university.info()

missing_values = university.isnull().sum()
print("Missing Values:\n", missing_values)
missing_values = university.isnull().sum()
print("Missing Values:\n", missing_values)
university.drop_duplicates(inplace=True)
university['Location'].value_counts()

plt.figure(figsize=(12, 8))
sns.countplot(data=university, x='Location', order=university['Location'].value_counts().index)
plt.title('Regional Distribution of Universities')
plt.xlabel('Region')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()

# Get the top ten countries with the most universities in the top 100
top_100 = university.head(100)
countries = top_100['Location']

# Find rows with null values in the 'Location' column
null_locations = top_100[top_100['Location'].isnull()]

# Display the rows with null values
print(null_locations)

#fill in the missing country data
universities_to_countries = {
    'Tsinghua University': 'China',
    'Peking University': 'China',
    'National University of Singapore': 'Singapore',

```

```

'Technical University of Munich': 'Germany',
'University of Hong Kong': 'Hong Kong',
'LMU Munich': 'Germany',
'KU Leuven': 'Belgium',
'Universität Heidelberg': 'Germany',
'Chinese University of Hong Kong': 'Hong Kong',
'McGill University': 'Canada',
'The University of Queensland': 'Australia',
'University of Manchester': 'United Kingdom',
'The Hong Kong University of Science and Technology': 'Hong Kong',
'Zhejiang University': 'China',
'UNSW Sydney': 'Australia',
'University of Science and Technology of China': 'China',
'University of Groningen': 'Netherlands',
'University of Bristol': 'United Kingdom',
'Leiden University': 'Netherlands',
'Yonsei University (Seoul campus)': 'South Korea',
'Hong Kong Polytechnic University': 'Hong Kong',
'Erasmus University Rotterdam': 'Netherlands',
'University of Glasgow': 'United Kingdom',
'McMaster University': 'Canada',
'University of Adelaide': 'Australia',
'City University of Hong Kong': 'Hong Kong'
}

```

```

# Iterate through the universities and assign the corresponding country
for university_name, country in universities_to_countries.items():
    top_100.loc[top_100['Name of University'] == university_name, 'Location'] = country

# number of universities in each country
country_counts = countries.value_counts()

# top ten countries
top_ten_countries = country_counts.head(10)

```

```

print(top_ten_countries)

# Convert 'OverAll Score' column to float using .loc for assignment
top_100.loc[:, 'OverAll Score'] = top_100['OverAll Score'].astype(float)

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(top_ten_countries, labels=top_ten_countries.index, autopct='%1.1f%%', startangle=140)
plt.title('Top 10 Countries in University Ranking (Average Score)')
plt.axis('equal')
# Show the pie chart
plt.show()

# top ten countries and their count of universities in the top 100
top_ten_countries_df = top_ten_countries.reset_index()

# Rename the columns
top_ten_countries_df.columns = ['Country', 'University Count']

#create the barchart
plt.figure(figsize=(10, 6))
bar_chart = sns.barplot(x='Country', y='University Count', data=top_ten_countries_df,
palette='viridis')
bar_chart.set_title('Top 10 Countries with Universities in Top 100')
bar_chart.set_xlabel('Country')
bar_chart.set_ylabel('Number of Universities')
plt.xticks(rotation=45, ha='right')
plt.show()

# Filter the data to include only the top 10 countries
filtered_data = top_100[top_100['Location'].isin(top_ten_countries.index)]

# Create a distribution box plot
plt.figure(figsize=(10, 6))

```

```
sns.boxplot(x='Location', y='OverAll Score', data=filtered_data)
plt.title('Distribution of University Rankings in Top 10 Countries')
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('OverAll Score')
plt.show()
```

```
# Scatterplot for Teaching Score vs. OverAll Score
plt.scatter(top_100['Teaching Score'], top_100['OverAll Score'])
plt.title('Teaching Score vs. OverAll Score')
plt.xlabel('Teaching Score')
plt.ylabel('OverAll Score')
plt.show()
```

```
# Scatterplot for Research Score vs. OverAll Score
plt.scatter(top_100['Research Score'], top_100['OverAll Score'])
plt.title('Research Score vs. OverAll Score')
plt.xlabel('Research Score')
plt.ylabel('OverAll Score')
plt.show()
```

```
# Scatterplot for Citations Score vs. OverAll Score
plt.scatter(top_100['Citations Score'], top_100['OverAll Score'])
plt.title('Citations Score vs. OverAll Score')
plt.xlabel('Citations Score')
plt.ylabel('OverAll Score')
plt.show()
```

```
# Scatterplot for Industry Income Score vs. OverAll Score
plt.scatter(top_100['Industry Income Score'], top_100['OverAll Score'])
plt.title('Industry Income Score vs. OverAll Score')
plt.xlabel('Industry Income Score')
plt.ylabel('OverAll Score')
plt.show()
```



```
# Scatterplot for International Outlook Score vs. OverAll Score
plt.scatter(top_100['International Outlook Score'], top_100['OverAll Score'])
plt.title('International Outlook Score vs. OverAll Score')
plt.xlabel('International Outlook Score')
plt.ylabel('OverAll Score')
plt.show()
```

```
top_n = 50
top_scores_teaching_rank = university.nlargest(top_n, 'Teaching Score')
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x='Name of University', y='Teaching Score', data=top_scores_teaching_rank)
plt.title(f'Top {top_n} Universities by Teaching Scores')
plt.xlabel('University Name')
plt.xticks(rotation=90)
plt.ylabel('Teaching Score Rank')
plt.show()
```

```
top_n = 50
top_scores_research_rank = university.nlargest(top_n, 'Research Score')
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x='Name of University', y='Research Score', data=top_scores_research_rank,
color='green')
plt.title(f'Top {top_n} Universities by Research Scores')
plt.xlabel('University Name')
plt.xticks(rotation=90)
plt.ylabel('Research Score')
plt.show()
```

```
top_n = 50
top_scores_citation = university.nlargest(top_n, 'Citations Score')
```

```
plt.figure(figsize=(10, 6))  
sns.barplot(x='Name of University', y='Citations Score', data=top_scores_research_rank,  
color='skyblue')  
plt.title(f'Top {top_n} Universities by Citations Scores')  
plt.xlabel('Name of Universitys')  
plt.xticks(rotation=90)  
plt.ylabel('Citations Score')  
plt.show()
```

Screenshot Output

Data Collection

```
[3]: #import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
[4]: university = pd.read_csv('World_University_Rankings_2023.csv')
```

```
[5]: university.head()
```

	Rank	University	Location	No of student	student per staff	International Student	Female:Male Ratio	OverAll Score	Teaching Score	Research Score	Citations Score	Industry Income Score	International Outlook Score
0	1	University of Oxford	United Kingdom	20,965	10.6	42%	48:52	96.4	92.3	99.7	99.0	74.9	96.2
1	2	Harvard University	United States	21,887	9.6	25%	50:50	95.2	94.8	99.0	99.3	49.5	80.5
2	3	University of Cambridge	United Kingdom	20,185	11.3	39%	47:53	94.8	90.9	99.5	97.0	54.2	95.8
3	3	Stanford University	United States	16,164	7.1	24%	46:54	94.8	94.2	96.7	99.8	65.0	79.8
4	5	Massachusetts Institute of Technology	United States	11,415	8.2	33%	40:60	94.2	90.7	93.6	99.8	90.9	89.3

```
[26]: university.info()
```

```
Data columns (total 13 columns):
# Column Non-Null Count Dtype
---
0 University Rank 2312 non-null object
1 Name of University 2233 non-null object
2 Location 2073 non-null object
3 No of student 2209 non-null object
4 No of student per staff 2208 non-null float64
5 International Student 2209 non-null object
6 Female:Male Ratio 2128 non-null object
7 OverAll Score 1799 non-null object
8 Teaching Score 1799 non-null float64
9 Research Score 1799 non-null float64
10 Citations Score 1799 non-null float64
11 Industry Income Score 1799 non-null float64
12 International Outlook Score 1799 non-null float64
dtypes: float64(6), object(7)
memory usage: 252.9+ KB
```

Data Cleaning

```
[27]: missing_values = university.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
University Rank      0
Name of University   79
Location             239
No of student        103
No of student per staff 104
International Student 103
Female:Male Ratio    184
OverAll Score        513
Teaching Score       513
Research Score       513
Citations Score      513
Industry Income Score 513
International Outlook Score 513
dtype: int64
```

```
[29]: university.dropna(inplace=True)
```

```
[30]: missing_values = university.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
University Rank      0
Name of University   0
Location             0
No of student        0
No of student per staff 0
International Student 0
Female:Male Ratio    0
OverAll Score        0
Teaching Score       0
Research Score       0
Citations Score      0
Industry Income Score 0
International Outlook Score 0
dtype: int64
```

```
[31]: university.drop_duplicates(inplace=True)
```

```
[36]: university['Location'].value_counts()
```

```
Location
United States    166
Japan            112
United Kingdom   92
India            65
Brazil           60
```

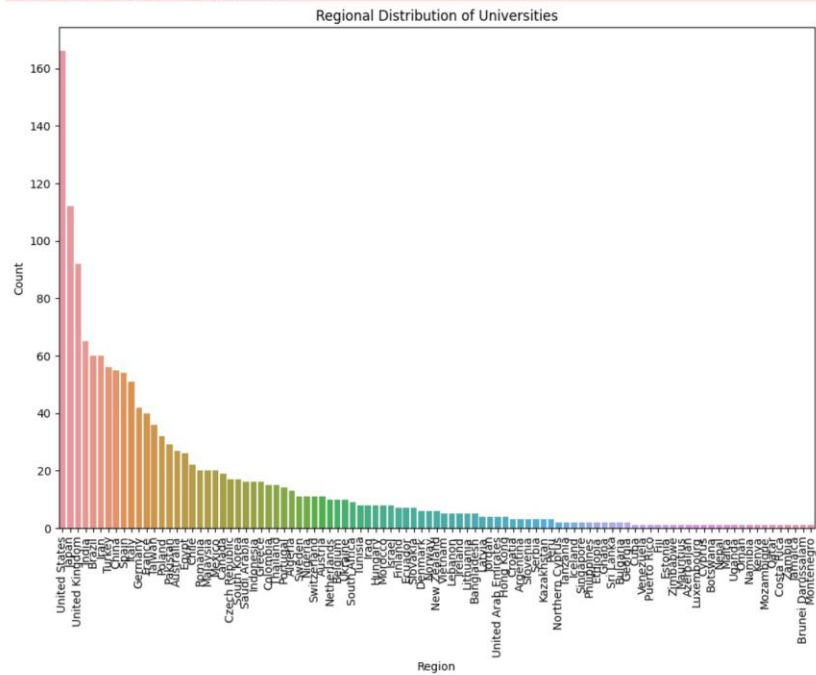
```

Costa Rica      1
Zambia         1
Jamaica        1
Brunei Darussalam 1
Montenegro     1
Name: count, Length: 99, dtype: int64

[37]: plt.figure(figsize=(12, 8))
sns.countplot(data=university, x='Location', order=university['Location'].value_counts().index)
plt.title('Regional Distribution of Universities')
plt.xlabel('Region')
plt.ylabel('Count')
plt.xticks(rotations=90)
plt.show()

C:\Users\mevho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mevho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):

```



```
# Iterate through the universities and assign the corresponding country
for university_name, country in universities_to_countries.items():
    top_100.loc[top_100["Name of University"] == university_name, "Location"] = country
```

Data Analysis

```
[41]: # number of universities in each country
country_counts = countries.value_counts()

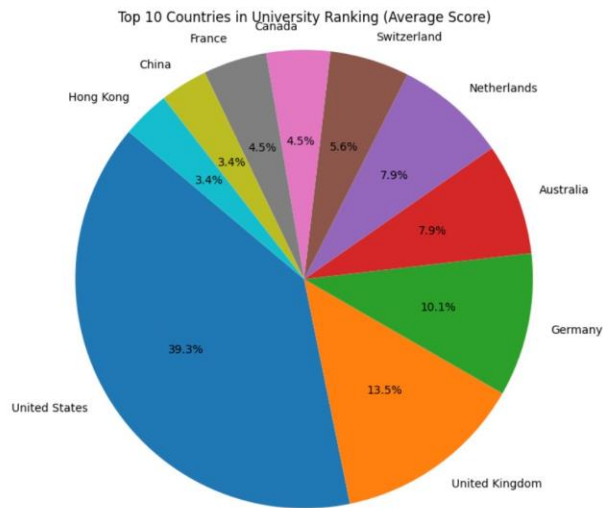
# top ten countries
top_ten_countries = country_counts.head(10)

print(top_ten_countries)

Location
United States    35
United Kingdom   12
Germany           9
Australia         7
Netherlands       7
Switzerland       5
Canada            4
France            4
China             3
Hong Kong        3
Name: count, dtype: int64

[42]: # Convert 'OverAll Score' column to float using .loc for assignment
top_100.loc[:, 'OverAll Score'] = top_100['OverAll Score'].astype(float)

[43]: # Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(top_ten_countries, labels=top_ten_countries.index, autopct='%1.1f%%', startangle=140)
plt.title('Top 10 Countries in University Ranking (Average Score)')
plt.axis('equal')
# Show the pie chart
plt.show()
```



```
# Top 10 Countries and their Count of Universities in the top 100

[44]: # top ten countries and their count of universities in the top 100
top_ten_countries_df = top_ten_countries.reset_index()

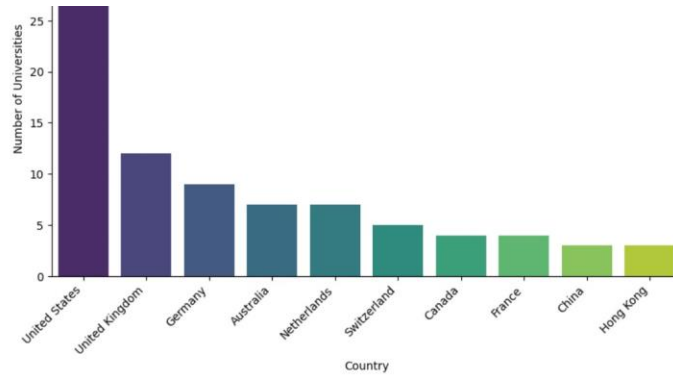
# Rename the columns
top_ten_countries_df.columns = ['Country', 'University Count']

# Create the bar chart
plt.figure(figsize=(10, 6))
bar_chart = sns.barplot(x='Country', y='University Count', data=top_ten_countries_df, palette='viridis')
bar_chart.set_title('Top 10 Countries with Universities in Top 100')
bar_chart.set_xlabel('Country')
bar_chart.set_ylabel('Number of Universities')
plt.xticks(rotation=45, ha='right')
plt.show()
```

C:\Users\me\who\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\me\who\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\me\who\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):

Top 10 Countries with Universities in Top 100

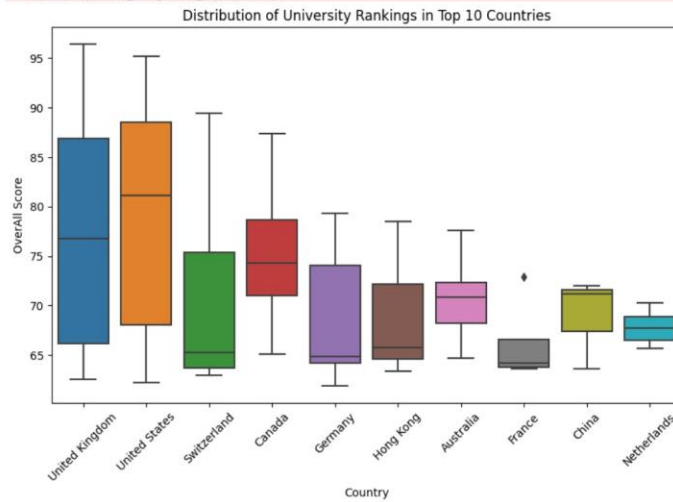




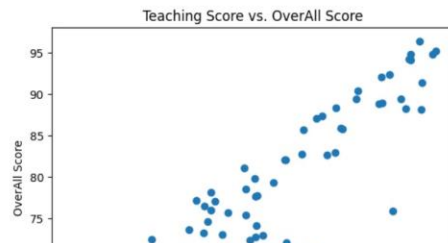
```
[45]: # Filter the data to include only the top 10 countries
filtered_data = top_100[top_100['Location'].isin(top_ten_countries.index)]

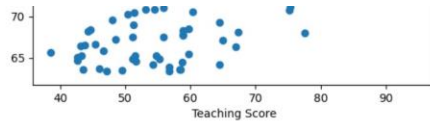
# Create a distribution box plot
plt.figure(figsize=(10, 6))
sns.boxplot(x='Location', y='OverAll Score', data=filtered_data)
plt.title('Distribution of University Rankings in Top 10 Countries')
plt.xticks(rotations=45)
plt.xlabel('Country')
plt.ylabel('OverAll Score')
plt.show()

C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

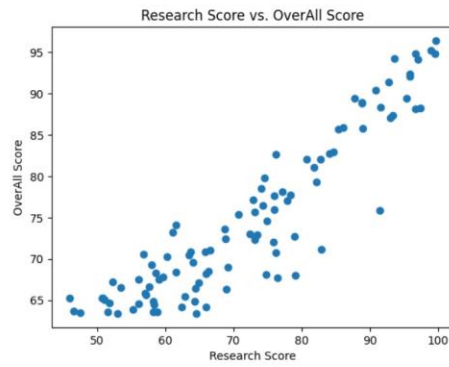


```
[19]: # Scatterplot for Teaching Score vs. OverAll Score
plt.scatter(top_100['Teaching Score'], top_100['OverAll Score'])
plt.title('Teaching Score vs. OverAll Score')
plt.xlabel('Teaching Score')
plt.ylabel('OverAll Score')
plt.show()
```

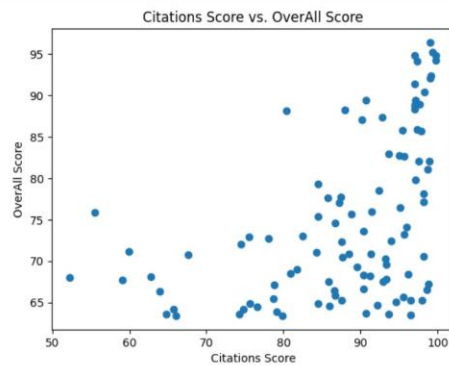




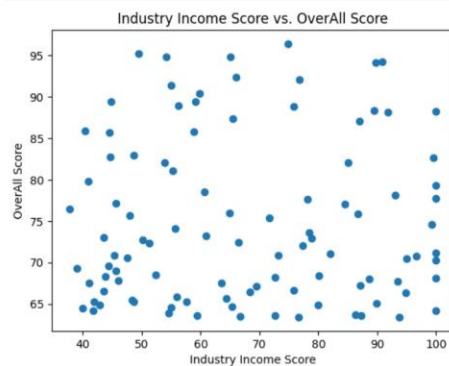
```
[20]: # Scatterplot for Research Score vs. OverAll Score
plt.scatter(top_100['Research Score'], top_100['OverAll Score'])
plt.title('Research Score vs. OverAll Score')
plt.xlabel('Research Score')
plt.ylabel('OverAll Score')
plt.show()
```



```
[21]: # Scatterplot for Citations Score vs. OverAll Score
plt.scatter(top_100['Citations Score'], top_100['OverAll Score'])
plt.title('Citations Score vs. OverAll Score')
plt.xlabel('Citations Score')
plt.ylabel('OverAll Score')
plt.show()
```

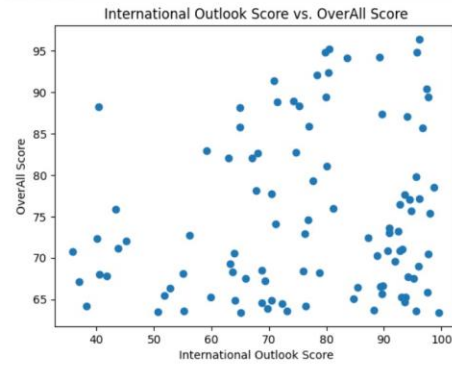


```
[22]: # Scatterplot for Industry Income Score vs. OverAll Score
plt.scatter(top_100['Industry Income Score'], top_100['OverAll Score'])
plt.title('Industry Income Score vs. OverAll Score')
plt.xlabel('Industry Income Score')
plt.ylabel('OverAll Score')
plt.show()
```



```
[23]: # Scatterplot for International Outlook Score vs. OverAll Score
plt.scatter(top_100['International Outlook Score'], top_100['OverAll Score'])
```

```
plt.title('International Outlook Score vs. OverAll Score')
plt.xlabel('International Outlook Score')
plt.ylabel('OverAll Score')
plt.show()
```

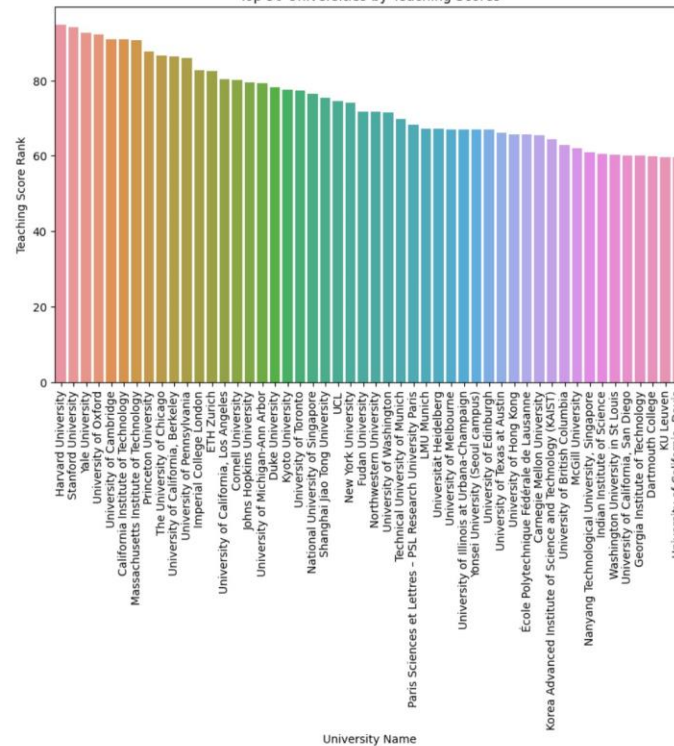


```
[60]: top_n = 50
top_scores_teaching_rank = university.nlargest(top_n, 'Teaching Score')

plt.figure(figsize=(10, 6))
sns.barplot(xs='Name of University', ys='Teaching Score', data=top_scores_teaching_rank)
plt.title(f'Top {top_n} Universities by Teaching Scores')
plt.xlabel('University Name')
plt.xticks(rotation=90)
plt.ylabel('Teaching Score Rank')
plt.show()
```

C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):

Top 50 Universities by Teaching Scores



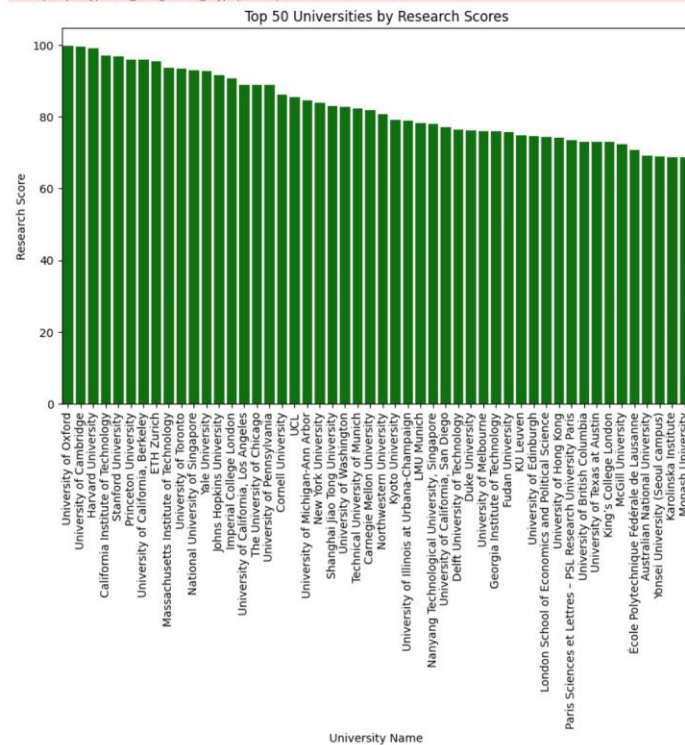
```
[59]: top_n = 50
top_scores_research_rank = university.nlargest(top_n, 'Research Score')

plt.figure(figsize=(10, 6))
sns.barplot(xs='Name of University', ys='Research Score', data=top_scores_research_rank, color='green')
plt.title(f'Top {top_n} Universities by Research Scores')
plt.xlabel('University Name')
plt.xticks(rotation=90)
```



```
plt.ylabel('Research Score')
plt.show()

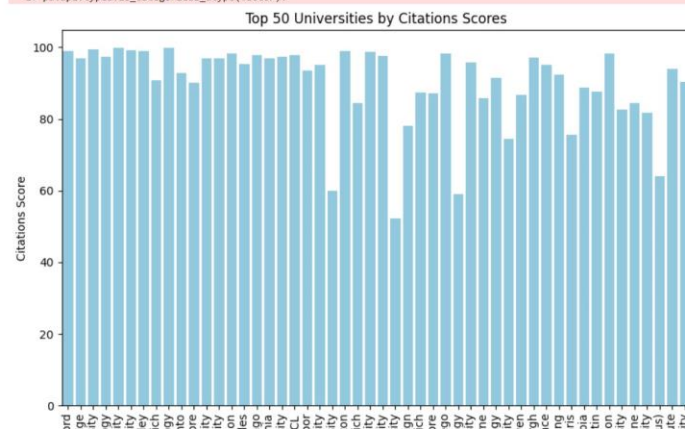
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



```
[61]: top_n = 50
top_scores_citation = university.nlargest(top_n, 'Citations Score')

plt.figure(figsize=(10, 6))
sns.barplot(x='Name of University', y='Citations Score', data=top_scores_research_rank, color='skyblue')
plt.title('Top (top_n) Universities by Citations Scores')
plt.xlabel('Name of Universitys')
plt.xticks(rotation=90)
plt.ylabel('Citations Score')
plt.show()
```

```
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\mewho\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



University of Oxford
University of Cambridge
University of Toronto
California Institute of Technology
Stanford University
Princeton University
University of California Berkeley
ETH Zurich
Massachusetts Institute of Technology
National University of Singapore
Tsinghua University
Johns Hopkins University
Imperial College London
University of California, Los Angeles
The University of Chicago
University of Pennsylvania
Cornell University
University of Michigan-Ann Arbor
Harvard University
Shanghai Jiao Tong University
University of Washington
Technical University of Munich
Carnegie Mellon University
Northwestern University
Kyoto University
University of Illinois at Urbana-Champaign
LMU Munich
Nanyang Technological University, Singapore
University of Science and Technology of China
Delft University of Technology
University of Melbourne
Georgia Institute of Technology
Fudan University
KU Leuven
University of Edinburgh
London School of Economics and Political Science
University of Vienna
University of Tokyo
Paris Sciences et Lettres - PSL Research University Paris
University of British Columbia
University of Texas at Austin
King's College London
University of London
Ecole Polytechnique Fédérale de Lausanne
Australian National University
Yonsei University (Seoul campus)
Karolinska Institute
Monash University

Name of Universities

[]: