**Mega Project:**

**Data Base Management System using Python and SQL**

This management project will give you the idea of how to interact the python with Databases. Where you have to perform whole end to end Operations using SQL , Python and Power BI

**Task 1:-**

Import the Dataset in python Its Called World Ranking Universities where you have several columns like

1. world\_rank
2. institution
3. country
4. national\_rank
5. quality\_of\_education
6. alumini\_employment
7. quality\_of\_faculty
8. publications
9. influence
10. citations
11. broad\_impact
12. patents
13. score
14. year

**Task 2 :-**

Data Cleaning

Find the Missing values

Remove the duplicates from the datasets

Remove the inconsistencies from the datasets

**Task 3 :-**

Import the necessary libraries like

import pyodbc as odbc

# pip install pypyodbc

#DRIVER\_NAME = 'ODBC Driver 17 for SQL Server'

#SERVER\_NAME = 'DESKTOP-DF403AL\SQLEXPRESS'

#DATABASE\_NAME = 'Hitesh'

import numpy as np # linear algebra

import pandas as pd

# data processing, CSV file I/O (e.g. pd.read\_csv)

import sqlite3

import matplotlib.pyplot as plt

Make a Connection String with python

Connection\_string = (

r'DRIVER={SQL Server};'

r'SERVER= your SERVER Name;'

r'DATABASE=Give your Database Name in which Database you want to work;'

r'Trusted\_Connection=yes;'

)

Conn = odbc.connect(Connection\_string)

print(Conn)

After Connecting your Database you will get this type of message given below

<pyodbc.Connection object at 0x0000019EC4A98FA0>

After that use below Code inorder to connect with MS-SQL Server

from sqlalchemy.engine import URL

connection\_url = URL.create("mssql+pyodbc", query={"odbc\_connect": Connection\_string})

from sqlalchemy import create\_engine

engine = create\_engine(connection\_url)

Now take a new cell and import below Statement inorder to start with SQL Database

import sqlalchemy as sa

with engine.begin() as conn:

df = pd.read\_sql\_query(sa.text("SELECT TOP 10 \* FROM WorldUniversity"), conn)

print(df)

**Task 4:-**

Congratulations you have Successfully Connected your SQL Server with Python and ready to solve the Problems

If you have to solve the Query then you have to write this given below line everytime

with engine.begin() as conn:

df = pd.read\_sql\_query(sa.text("Your SQL Query"), conn)

print(df)

Questions using SQL:

*Code snippet I used to connect sql with python*

***import pandas as pd***

***import sqlalchemy as sa***

***import urllib.parse***

***connection\_string = (***

***r'DRIVER=ODBC Driver 17 for SQL Server;'***

***r'SERVER=HARIPRIYA\SQLEXPRESS;'***

***r'DATABASE=University;'***

***r'Trusted\_Connection=yes;'***

***)***

***connection\_url = f"mssql+pyodbc:///?odbc\_connect={urllib.parse.quote\_plus(connection\_string)}"***

***engine = sa.create\_engine(connection\_url)***

***sql\_query = """***

***select institution from world\_ranking\_universities***

***where quality\_of\_education <=5 and year=2014***

***"""***

***with engine.begin() as conn:***

***df = pd.read\_sql\_query(sql\_query, conn)***

***for col in df.select\_dtypes(include='float').columns:***

***df[col] = df[col].astype(int)***

***pd.set\_option('display.max\_rows', None)***

***pd.set\_option('display.max\_columns', None)***

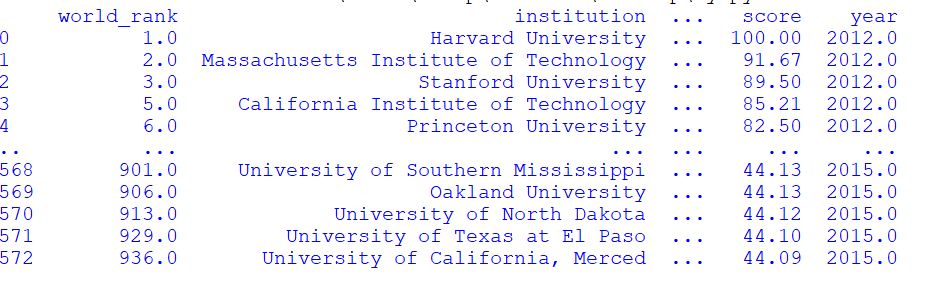
***print(df)***

1. Retrieve all columns for universities in the USA.

*with engine.begin() as conn:*

*df = pd.read\_sql\_query("****select \* from world\_ranking\_universities where country in ('USA')****", conn)*

*print(df)*

**

1. Find the top 10 universities with the highest scores in 2012.

*sql\_query = """*

*select top(10) score,institution from world\_ranking\_universities*

*where year=2012*

*order by score desc*

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

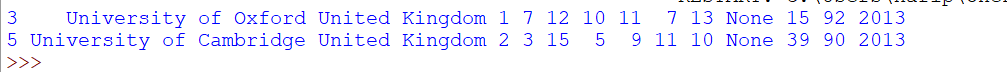
1. List universities in the United Kingdom with a score above 80 in 2013.

*with engine.begin() as conn:*

*df = pd.read\_sql\_query("****select \* from world\_ranking\_universities where country = 'United Kingdom' and year=2013 and score > 80****", conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*

1. Count the number of universities in each country.

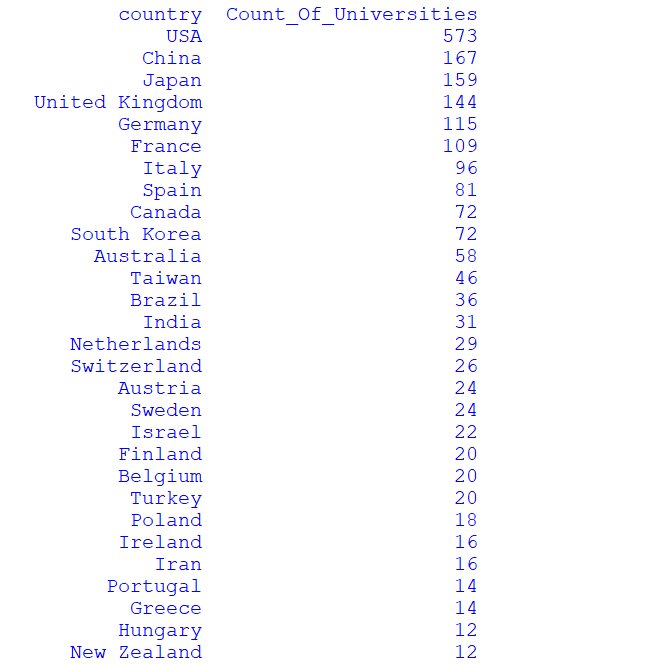
*With engine.begin() as conn:*

*df = pd.read\_sql\_query("****select country,count(\*) as Count\_Of\_Universities from world\_ranking\_universities group by country order by Count\_Of\_Universities desc****", conn)*

*print(df)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*

1. Calculate the average score for universities in each country in 2014.

*with engine.begin() as conn:*

*df = pd.read\_sql\_query("****select round(avg(score),2) as Average\_score,country from world\_ranking\_universities where year=2014 group by country*** *", conn)*

*print(df)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*

**

1. Find universities with a quality\_of\_education score greater than 20.

*with engine.begin() as conn:*

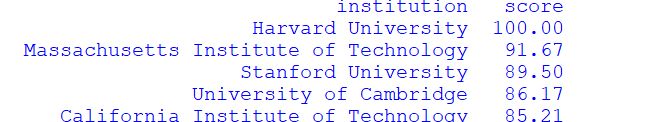
*df = pd.read\_sql\_query("****select institution,score from world\_ranking\_universities where score > 20*** *", conn)*

*print(df)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*



1. Retrieve universities with a score between 70 and 80.

*with engine.begin() as conn:*

*df = pd.read\_sql\_query("****select institution,score from world\_ranking\_universities where score between 70 and 80*** *", conn)*

*print(df)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*

**

1. List the top 5 universities with the highest alumni employment scores in 2012

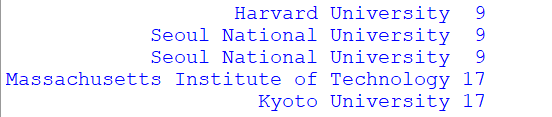
*with engine.begin() as conn:*

*df = pd.read\_sql\_query("****select top 5 institution,alumni\_employment from world\_ranking\_universities where alumni\_employment in (select alumni\_employment from world\_ranking\_universities where year=2012)****", conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*

**

1. Find the university with the highest quality\_of\_faculty in 2013

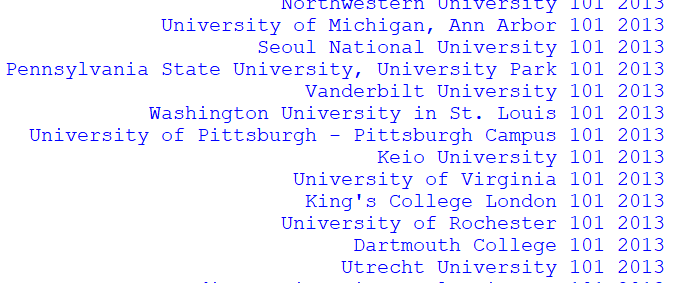
*with engine.begin() as conn:*

*df = pd.read\_sql\_query("****select institution,quality\_of\_faculty,year from world\_ranking\_universities where quality\_of\_faculty in (select max(quality\_of\_faculty) as Highest\_quality\_of\_faculty from world\_ranking\_universities where year=2013)and year=2013****", conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*print(df.to\_string(index=False, header=False))*

**

1. Count the number of universities that have a national\_rank less than 5.

*sql\_query = """*

***SELECT COUNT(\*) AS Number\_of\_Universities,institution***

***FROM world\_ranking\_universities***

***WHERE national\_rank < 5***

***group by institution***

***order by Number\_of\_Universities***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

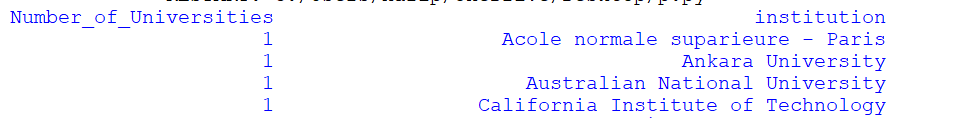
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Retrieve universities with a quality\_of\_education rank equal to 1

*sql\_query = """*

***select institution,quality\_of\_education from world\_ranking\_universities***

***where quality\_of\_education=1***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

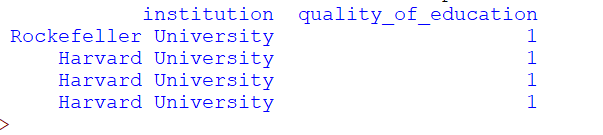
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. List the top 10 universities with the highest citations in 2014

*sql\_query = """*

***select top 10 institution,citations,year from world\_ranking\_universities***

***where citations in (select max(citations) as Max\_Citations from world\_ranking\_universities***

***where year=2014****)"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Calculate the average influence score for universities in the USA.

*sql\_query = """*

***select institution,round(avg(influence),0) as Average\_influence ,round(avg(score),1) as Average\_Score from world\_ranking\_universities***

***where country = 'USA'***

***group by institution***

***order by Average\_influence desc****"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Find universities with a broad\_impact rank less than or equal to 50.

*sql\_query = """*

***select institution,broad\_impact from world\_ranking\_universities***

***where broad\_impact <= 50***

***order by broad\_impact***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

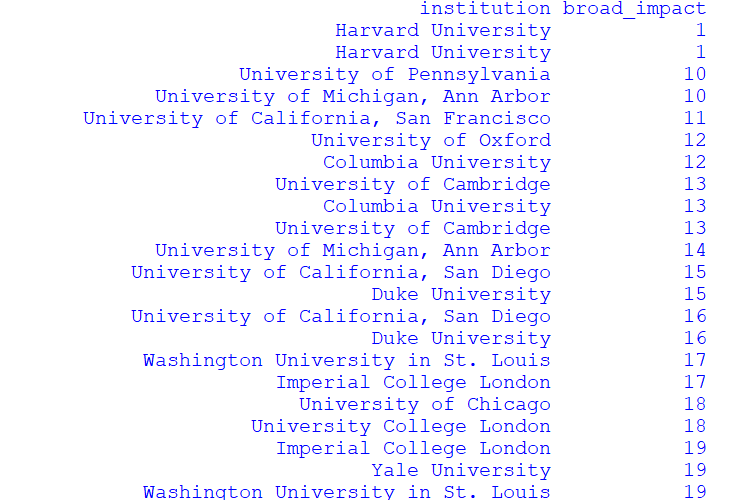
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Retrieve universities in Japan with a score greater than 60 in 2012.

*sql\_query = """*

***select institution,Country,score,year from world\_ranking\_universities***

***where Country = 'Japan' and score > 60 and year=2012***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

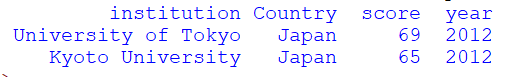
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. List the top 5 universities with the highest patents in 2013.

*sql\_query = """*

***select top 5 institution,patents as Highest\_patents,year from world\_ranking\_universities***

***where patents in (select max(patents) from world\_ranking\_universities where year=2013)***

***and year=2013***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

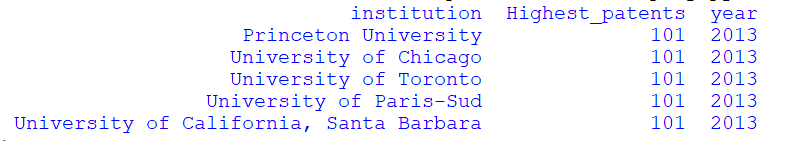
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Count the number of universities with a quality\_of\_faculty score between 5 and 10.

*sql\_query = """*

***select institution,count(institution) as No\_Of\_Universities,quality\_of\_faculty from world\_ranking\_universities***

***where quality\_of\_faculty between 5 and 10***

***group by institution,quality\_of\_faculty***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

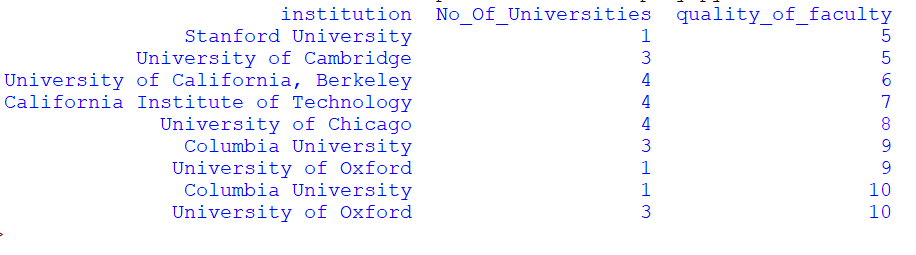
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*



1. Calculate the average score for universities in the United Kingdom in 2014.

*sql\_query = """*

***select round(avg(score),2) as Average\_score, institution from world\_ranking\_universities***

***where country='United Kingdom' and year=2014***

***group by institution***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

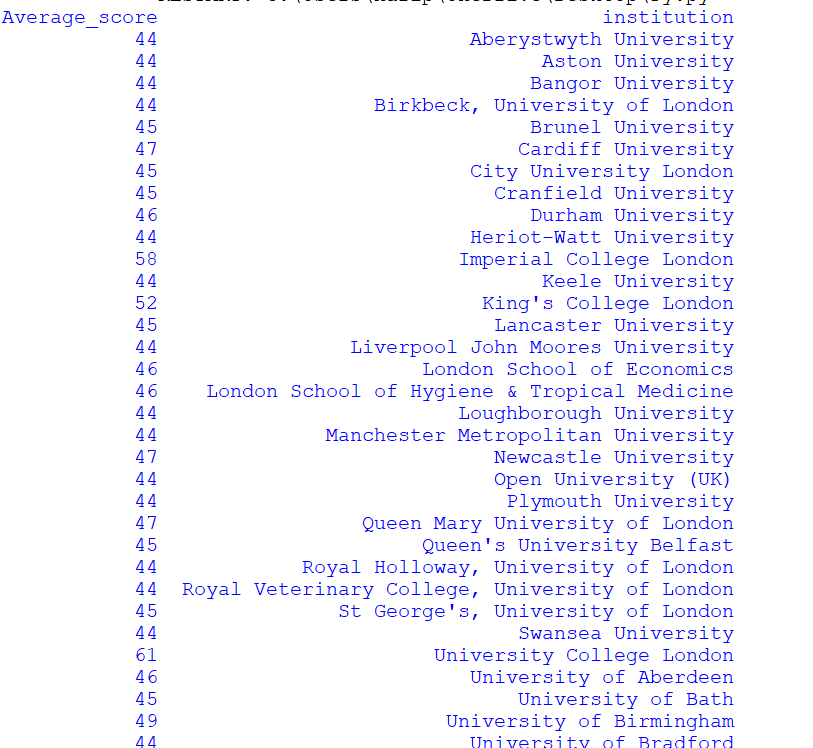
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Find universities with a national\_rank between 1 and 3 in 2012.

*sql\_query = """*

***select institution,national\_rank from world\_ranking\_universities***

***where national\_rank between 1 and 3 and year=2012***

***order by national\_rank desc***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

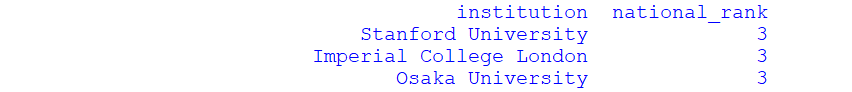
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*



1. List universities with a citations rank less than 10.

*sql\_query = """*

***select institution,citations from world\_ranking\_universities***

***where national\_rank < 10***

***order by citations***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Retrieve the university with the highest alumni\_employment in 2013.

*sql\_query = """*

***SELECT TOP 1 institution, alumni\_employment AS Highest\_alumni\_employment, 2013 AS year***

***FROM world\_ranking\_universities***

***WHERE year = 2013***

***ORDER BY alumni\_employment DESC***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

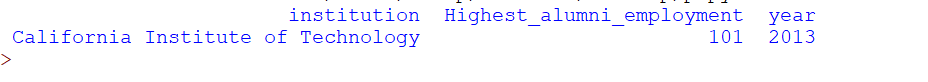
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Find the top 5 universities with the highest publications in 2012.

*sql\_query = """*

***select top 5 institution,publications,year from world\_ranking\_universities***

***where publications in (select max(publications) from world\_ranking\_universities where year=2012)***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

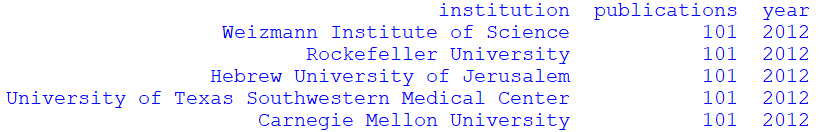
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Count the number of universities in each year.

*sql\_query = """*

***select count(institution) as Number\_of\_universities,year from world\_ranking\_universities***

***group by year***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

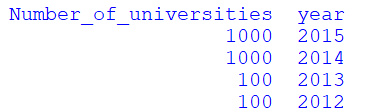
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Calculate the average alumni\_employment score for universities in the USA.

*sql\_query = """*

***select round(avg(alumni\_employment),2) as Average\_alumni\_employment ,round(avg(score),2) as Average\_Score from world\_ranking\_universities***

***where country='USA'***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Retrieve universities with a broad\_impact score greater than 70.

*sql\_query = """*

***select institution from world\_ranking\_universities***

***where broad\_impact > 70***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

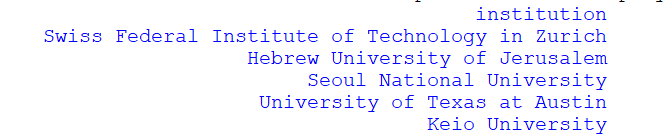
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. List the top 10 universities with the highest influence in 2014.

*sql\_query = """*

***select top 10 institution from world\_ranking\_universities***

***where influence in (select max(influence) from world\_ranking\_universities where year=2014)***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

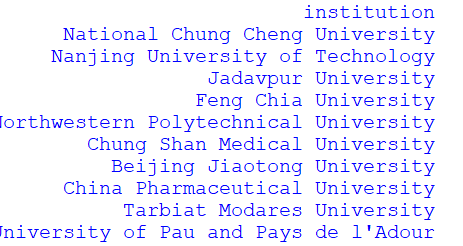
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Find universities with a patents rank equal to 1.

*sql\_query = """*

***select institution from world\_ranking\_universities***

***where patents = 1***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

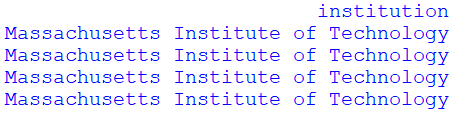
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

1. Count the number of universities with a broad\_impact between 20 and 30.

*sql\_query = """*

***select count(institution) as No\_Of\_Universities from world\_ranking\_universities***

***where broad\_impact between 20 and 30***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

29)Calculate the average quality\_of\_faculty score for universities in the United Kingdom.

*sql\_query = """*

***select round(avg(quality\_of\_faculty),2) as Average\_quality\_of\_faculty,round(avg(score),0) as Average\_Score from world\_ranking\_universities***

***where country = 'United Kingdom'***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

30)Find universities with an influence rank less than 5.

*sql\_query = """*

***select institution,influence from world\_ranking\_universities***

***where influence < 5***

***order by influence***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

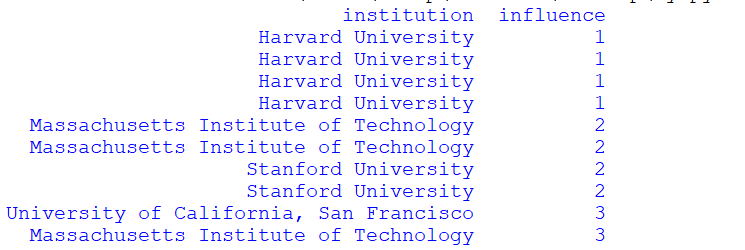
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

31)Retrieve universities with a score greater than 75 and a national\_rank less than or equal to 10 in 2013.

*sql\_query = """*

***select institution,score,national\_rank,year from world\_ranking\_universities***

***where score > 75 and national\_rank <= 10 and year=2013***

***order by national\_rank***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

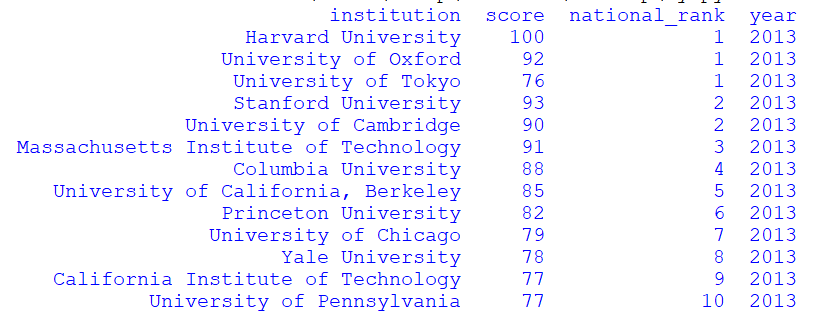
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

32)List the top 5 universities with the highest patents in 2014.

*sql\_query = """*

***select top 5 institution,patents,year from world\_ranking\_universities***

***where patents in (select max(patents) from world\_ranking\_universities where year=2014)***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

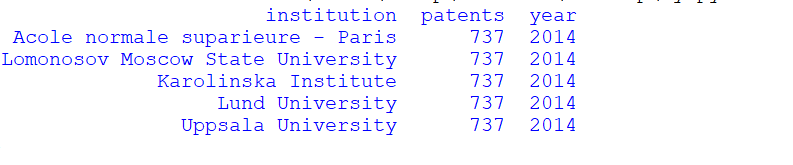
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

33)Count the number of universities with an alumni\_employment score greater than 50.

*sql\_query = """*

***select count(institution) as Number\_of\_universities from world\_ranking\_universities***

***where alumni\_employment > 50 and score > 50***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

34)Calculate the average quality\_of\_education score for universities in Japan.

*sql\_query = """*

***select round(avg(quality\_of\_education),2) as Average\_quality\_of\_education ,round(avg(score),2) as Average\_score from world\_ranking\_universities***

***where country='Japan'***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

35)Find universities with a quality\_of\_education rank equal to 1 in 2014.

*sql\_query = """*

***select institution,quality\_of\_education,year from world\_ranking\_universities***

***where quality\_of\_education=1 and year=2014***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

36)Retrieve universities with a score greater than 80 and an alumni\_employment rank less than 5.

*sql\_query = """*

***select institution,score,alumni\_employment from world\_ranking\_universities***

***where score > 80 and alumni\_employment < 5***

***order by score***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

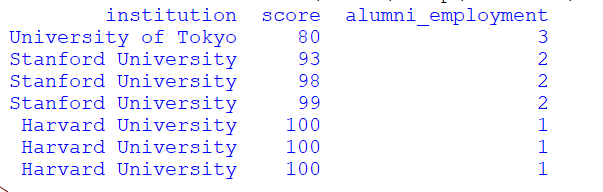
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

37)List the top 10 universities with the highest publications in 2013.

*sql\_query = """*

***select top 10 institution,publications,year from world\_ranking\_universities***

***where publications in (select max(publications) from world\_ranking\_universities where year=2013)***

***and year=2013***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

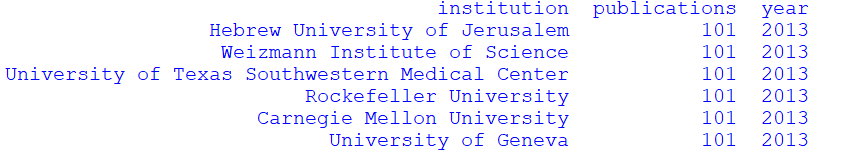
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

38)Count the number of universities with a broad\_impact score between 40 and 50.

*sql\_query = """*

***select count(institution) as Number\_of\_universities from world\_ranking\_universities***

***where broad\_impact between 40 and 50 and score between 40 and 50***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

39)Calculate the average score for universities in Australia.

*sql\_query = """*

***select round(avg(score),2) as Average\_score from world\_ranking\_universities***

***where country='Australia'***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

40)Find universities with an influence rank equal to 1 in 2012.

*sql\_query = """*

***select institution,influence,year from world\_ranking\_universities***

***where influence=1 and year=2012***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

41)Retrieve universities with a quality\_of\_faculty score greater than 15.

*sql\_query = """*

***select institution from world\_ranking\_universities***

***where quality\_of\_faculty > 15 and score >15***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

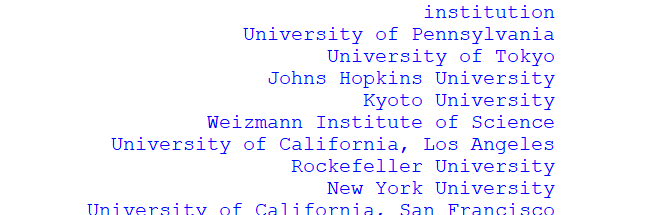
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

42)List the top 5 universities with the highest alumni\_employment in 2014.

*sql\_query = """*

***select top 5 institution,alumni\_employment,year from world\_ranking\_universities***

***where alumni\_employment in (select max(alumni\_employment) from world\_ranking\_universities where year=2014)***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

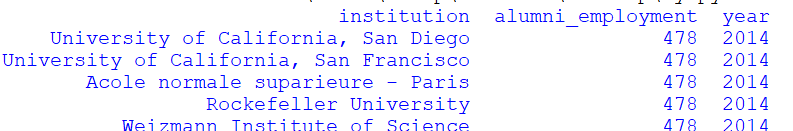
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

43)Count the number of universities with a national\_rank less than 20 in 2013.

*sql\_query = """*

***select count(institution) as Number\_of\_universities from world\_ranking\_universities***

***where national\_rank < 20 and year=2013***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

44)Calculate the average citations score for universities in Germany.

*sql\_query = """*

***select round(avg(citations),2) as Average\_citations,round(avg(score),2) as Average\_score from world\_ranking\_universities***

***where country='Germany'***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

45)Find universities with a publications rank equal to 1.

*sql\_query = """*

***select institution from world\_ranking\_universities***

***where publications=1***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

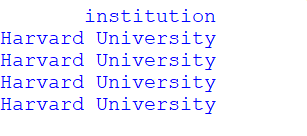
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

46)Retrieve universities with a broad\_impact score greater than 60 and a national\_rank less than or equal to 5.

*sql\_query = """*

*select institution from world\_ranking\_universities*

*where broad\_impact > 60 and national\_rank <= 5*

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

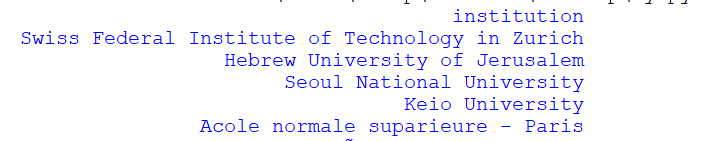
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

47)List the top 10 universities with the highest quality\_of\_education in 2012.

*sql\_query = """*

***select top 10 institution from world\_ranking\_universities***

***where quality\_of\_education in (select max(quality\_of\_education) from world\_ranking\_universities where year=2012)***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

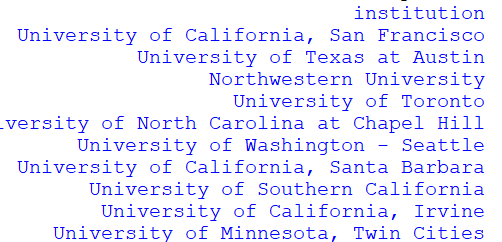
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

48)Count the number of universities with a score greater than 90.

*sql\_query = """*

***select count(institution) as Number\_of\_universities from world\_ranking\_universities***

***where score > 90***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

49)Calculate the average influence score for universities in the United Kingdom.

*sql\_query = """*

***select round(avg(influence),2) as Average\_influence,round(avg(score),2) as Average\_score from world\_ranking\_universities***

***where country='United Kingdom'***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*

**

50)Find universities with a quality\_of\_education rank less than or equal to 5 in 2014.

*sql\_query = """*

***select institution from world\_ranking\_universities***

***where quality\_of\_education <=5 and year=2014***

*"""*

*with engine.begin() as conn:*

*df = pd.read\_sql\_query(sql\_query, conn)*

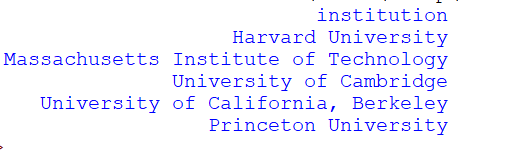
*for col in df.select\_dtypes(include='float').columns:*

*df[col] = df[col].astype(int)*

*pd.set\_option('display.max\_rows', None)*

*pd.set\_option('display.max\_columns', None)*

*print(df)*



**TASK 5 :-**

Questions Using Python

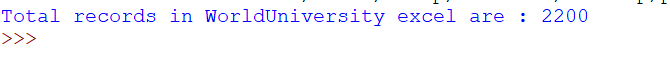
1. What is the total number of records in the dataset?

*import pandas as pd*

*WorldUniversity=pd.read\_excel('WorldUniversity.xlsx')*

*total\_records=len(WorldUniversity)*

*print("Total records in WorldUniversity excel are :",total\_records)*



1. How many columns are there in the dataset?

*import pandas as pd*

*WorldUniversity=pd.read\_excel('WorldUniversity.xlsx')*

*No\_of\_columns = WorldUniversity.columns.tolist()*

*print("Total columns in WorldUniversity excel are :",len(No\_of\_columns))*

**

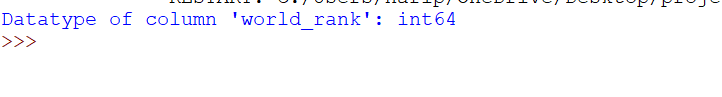
1. What is the datatype of the "world\_rank" column?

*import pandas as pd*

*WorldUniversity=pd.read\_excel('WorldUniversity.xlsx')*

*datatype=WorldUniversity['world\_rank'].dtype*

*print("Datatype of column 'world\_rank':", datatype)*

**

1. Which country is represented the most in the dataset?

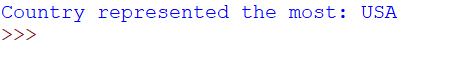
*import pandas as pd*

*WorldUniversity=pd.read\_excel('WorldUniversity.xlsx')*

*country\_counts = WorldUniversity['country'].value\_counts()*

*most\_common\_country = country\_counts.idxmax()*

*print("Country represented the most:", most\_common\_country)*

**

1. What is the average "quality\_of\_education" across all institutions?

*import pandas as pd*

*WorldUniversity=pd.read\_excel('WorldUniversity.xlsx')*

*Average = WorldUniversity['quality\_of\_education'].mean()*

*print("Average of quality\_of\_education across all institutions is:",round(Average,2))*

**

1. Find the institution with the highest "alumni\_employment" value.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*max\_alumni\_employment\_row = WorldUniversity['alumni\_employment'].idxmax()*

*institution\_name = WorldUniversity.loc[max\_alumni\_employment\_row,'institution']*

*print("Institution with the highest alumni employment value is:", institution\_name)*



1. In which year does the dataset end?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*last\_year = WorldUniversity['year'].tail(1).iloc[0]*

*print("Year in which dataset ends is :",last\_year)*

**

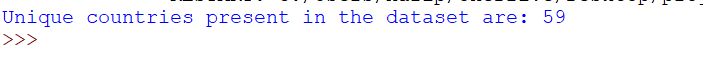
1. How many unique countries are present in the dataset?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*Unique\_conutry = WorldUniversity['country'].nunique()*

*print("Unique countries present in the dataset are:",Unique\_conutry)*



1. Which institution has the highest "score" in the year 2012?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year=WorldUniversity[WorldUniversity['year']==2012]*

*max\_score=year['score'].idxmax()*

*institution=year.loc[max\_score,'institution']*

*print("Institution with the highest score in the year 2012 is :",institution)*

1. What is the national rank of Harvard University in the year 2012?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] ==2012]*

*institution = year[year['institution'] == 'Harvard University']*

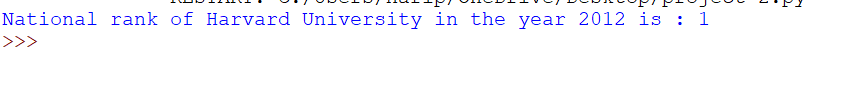
*if not institution.empty:*

*National\_rank = institution['national\_rank'].iloc[0]*

*print("National rank of Harvard University in the year 2012 is :",National\_rank)*

*else:*

*print("No data available for Harvard University in the year 2012.")*

**

1. Find the average "publications" value for institutions in the United Kingdom.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*country = WorldUniversity[WorldUniversity['country'] == 'United Kingdom']*

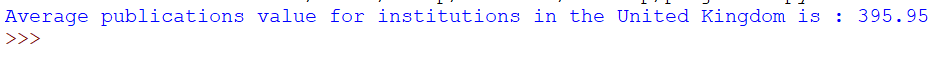
*if not country.empty:*

*Average\_publications = country['publications'].mean()*

*print("Average publications value for institutions in the United Kingdom is :",round(Average\_publications,2))*

*else:*

*print("No data available for publications in United Kingdom")*

**

1. Identify the institution with the highest "influence" in the year 2013.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2013]*

*max\_influence = year['influence'].idxmax()*

*institution = year.loc[max\_influence,'institution']*

*print("The institution with the highest influence in the year 2013 is :",institution)*

**

1. What is the minimum "broad\_impact" value in the dataset?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*minimum = WorldUniversity['broad\_impact'].min()*

*print("The minimum broad\_impact value in the dataset is :",int(minimum))*

**

1. How many institutions are from Japan in the year 2012?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year= WorldUniversity[WorldUniversity['year']==2012]*

*country = year[year['country'] == 'Japan']*

*if not country.empty:*

*institution = country['institution'].nunique()*

*print("Number of institutions from Japan in the year 2012 are : ",institution)*

*else:*

*print("No institutions are from Japan in the year 2012 ")*

**

1. What is the average "patents" value for institutions in the USA in the year 2013?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year= WorldUniversity[WorldUniversity['year']==2013]*

*country = year[year['country'] == 'USA']*

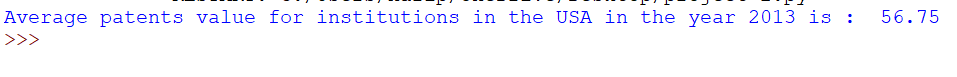
*if not country.empty:*

*patent = country['patents'].mean()*

*print("Average patents value for institutions in the USA in the year 2013 is : ",round(patent,2))*

*else:*

*print("No data available for patents for USA in the year 2013 ")*

**

1. Find the top 5 institutions with the highest "citations" in the year 2015.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

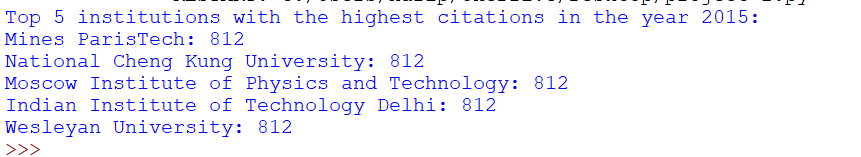
*year\_2015 = WorldUniversity[WorldUniversity['year'] == 2015]*

*top\_5\_institutions = year\_2015.nlargest(5, 'citations')*

*print("Top 5 institutions with the highest citations in the year 2015:")*

*for \_, row in top\_5\_institutions.iterrows():*

*print(f"{row['institution']}: {row['citations']}")*

**

1. What is the median "international\_students" percentage across all institutions? – *no relevant data noticed*
2. Identify the institution with the lowest "score" in the year 2014.

*import pandas as pd*

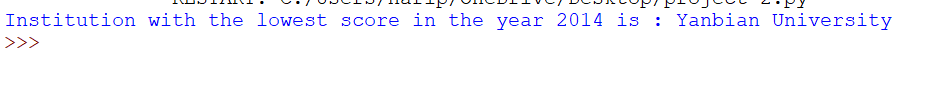
*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

*score = year['score'].idxmin()*

*institution = year.loc[score,'institution']*

*print("Institution with the lowest score in the year 2014 is :",institution)*

**

1. How many missing values are there in the "score" column?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

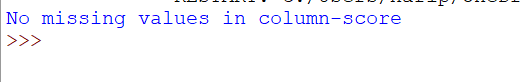
*Missing\_values = WorldUniversity['score'].isnull().any()*

*if Missing\_values:*

*print("Missing values in column- score are:",Missing\_values)*

*else:*

*print("No missing values in column-score")*

**

1. Find the top 3 countries with the most institutions in the dataset

*import pandas as pd*

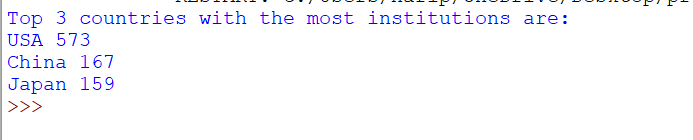
*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*country\_counts = WorldUniversity['country'].value\_counts()*

*top\_3\_countries = country\_counts.head(3)*

*print("Top 3 countries with the most institutions are:")*

*for country, country\_counts in top\_3\_countries.items():*

*print(f"{country}: {country\_counts}")*

1. Calculate the percentage of institutions with a score greater than 70.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

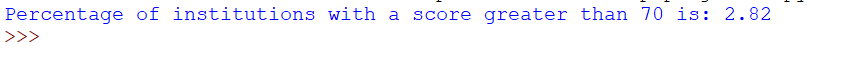
*score=WorldUniversity[WorldUniversity['score'] > 70]*

*total\_institutions\_greater\_than\_70= len(score)*

*total\_institutions = len(WorldUniversity)*

*percentage = (total\_institutions\_greater\_than\_70 / total\_institutions ) \* 100*

*print("Percentage of institutions with a score greater than 70 is:",round(percentage,2))*



1. What is the difference in score between MIT and Stanford University in 2016?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2016]*

*mit\_stanford = year[year['institution'].isin(['Massachusetts Institute of Technology', 'Stanford University'])]*

*if not mit\_stanford.empty:*

*mit\_score = mit\_stanford[mit\_stanford['institution'] == 'Massachusetts Institute of Technology']['score'].iloc[0]*

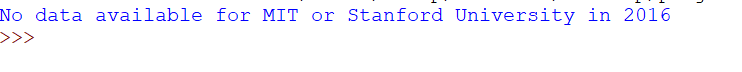
*stanford\_score = mit\_stanford[mit\_stanford['institution'] == 'Stanford University']['score'].iloc[0]*

*difference = mit\_score - stanford\_score*

*print("Difference in score between MIT and Stanford University in 2016:", difference)*

*else:*

*print("No data available for MIT or Stanford University in 2016")*

**

1. How many institutions have a "world\_rank" between 50 and 100 in the year 2011?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2011]*

*if not year.empty:*

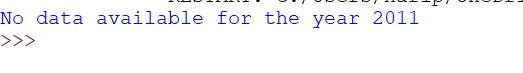
*world\_rank = year[(year['world\_rank'] >= 50) & (year['world\_rank'] <= 100)]*

*institutions = world\_rank['institution'].nunique()*

*print("Number of institutions having world\_rank between 50 and 100 in the year 2011:",institutions)*

*else:*

*print("No data available for the year 2011")*

**

1. Find the country with the highest average "industry\_income" across all years.—*No relevant data in excel*
2. What is the standard deviation of scores for institutions in the United States?

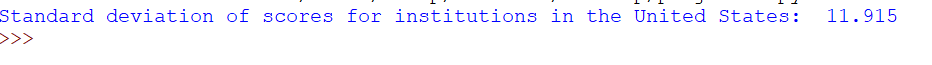
*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*country = WorldUniversity[WorldUniversity['country'] == 'USA' ]*

*score = country['score'].std()*

*print("Standard deviation of scores for institutions in the United States: ",round(score,3))*



1. Identify the institution with the highest "alumni\_employment" in the year 2015.

*import pandas as pd*

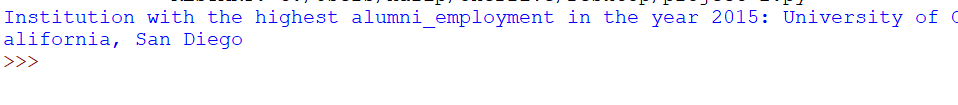
*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2015]*

*alumni\_employment = year['alumni\_employment'].idxmax()*

*institution = year.loc[alumni\_employment,'institution']*

*print("Institution with the highest alumni\_employment in the year 2015:" ,institution)*



1. Calculate the correlation between "score" and "patent" across all years.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*score\_patent = WorldUniversity[['score','patents']]*

*correlation = score\_patent.corr().iloc[0,1]*

*print("Correlation between 'score' and 'patent' across all years:", correlation)*



1. How many institutions have a "broad\_impact" greater than 800 in the year 2014?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] == 2014 ]*

*if not year.empty:*

*broad\_impact= year[year['broad\_impact'] > 800]*

*if not broad\_impact.empty:*

*institutions= broad\_impact['institution'].nunique()*

*print("Number of institutions have broad\_impact greater than 800 in the year 2014 :",institutions)*

*else:*

*print("No institutions have broad\_impact greater than 800 in the year 2014")*

*else:*

*print("No data avaiable for year 2014")*

**

1. Find the average "international\_students" percentage for institutions in Australia.—*no relevant data in excel*
2. Identify the institution with the highest " score" in the year 2016.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] == 2016 ]*

*if not year.empty:*

*score = year['score'].idxmax()*

*institution = year.loc[score,'institution']*

*print("institution with the highest score in the year 2016:",institution)*

*else:*

*print("No data available for year 2016")*

**

1. What is the percentage of missing values in the "broad\_impact" column?

*import pandas as pd*

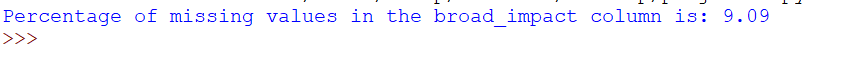
*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*broad\_impact=WorldUniversity['broad\_impact'].isnull().sum()*

*total\_values = len(WorldUniversity['broad\_impact'])*

*percentage = (broad\_impact / total\_values) \* 100*

*print("Percentage of missing values in the broad\_impact column is:",round(percentage,2))*

**

1. How many institutions have a "score" greater than 90 in the year 2013?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] ==2013]*

*if not year.empty:*

*score = year[year['score'] > 90]*

*if not score.empty:*

*institutions = score['institution'].nunique()*

*print("Institutions have score greater than 90 in the year 2013:",institutions)*

*else:*

*print("No institution scored greater than 90 in the year 2013")*

*else:*

*print("No data avialble for year 2013")*

**

1. Calculate the average score for institutions in the top 10 of in 2015.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] ==2015]*

*if not year.empty:*

*sort = year.sort\_values(by='score',ascending=False)*

*top\_10 = sort.head(10)*

*average=top\_10['score'].mean()*

*print("Average score for institutions in the top 10 in 2015:",round(average,2))*

*else:*

*print("No data avialble for year 2015")*

*or*

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year= WorldUniversity[WorldUniversity['year']== 2015]*

*if not year.empty:*

*top\_10 = year.nlargest(10,'score')*

*average=top\_10['score'].mean()*

*print("Average score for institutions in the top 10 in 2015:",round(average,2))*

*else:*

*print("No data available for year 2015")*

**

1. Identify the institution with the lowest "international\_students" percentage in the year 2012.—*no relevant data in excel*
2. What is the range of "score" values in the dataset?

*import pandas as pd*

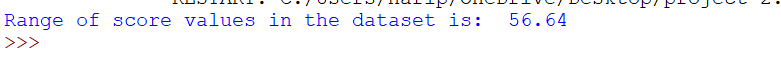
*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*min\_score = WorldUniversity['score'].min()*

*max\_score = WorldUniversity['score'].max()*

*range = max\_score-min\_score*

*print("Range of score values in the dataset is: " ,range)*

**

1. How many institutions have a "national\_rank" of 1 in the year 2014?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] ==2014]*

*if not year.empty:*

*national\_rank = year[year['national\_rank'] == 1]*

*if not national\_rank.empty:*

*institutions = national\_rank['institution'].nunique()*

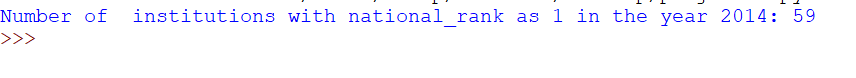
*print("Number of institutions with national\_rank as 1 in the year 2014:",institutions)*

*else:*

*print("No National\_rank for 1 is available")*

*else:*

*print("No data available for year 2014")*

**

1. Find the country with the highest average "international\_students" percentage.—No *relevant data in excel*
2. Calculate the average score for institutions in the top 5 of "world\_rank" in 2016.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] ==2016]*

*if not year.empty:*

*sort = year.sort\_values(by='world\_rank',ascending=False)*

*top\_5 = sort.head(5)*

*average = top\_5['score'].mean()*

*print("Average score for institutions in the top 5 of world\_rank in 2016:",round(average,2))*

*else:*

*print("No data available for year 2016")*

**

1. Identify the institution with the highest score in the year 2014.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] ==2014]*

*if not year.empty:*

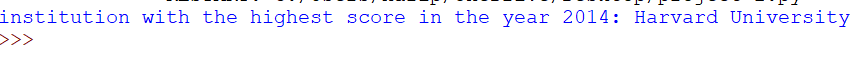
*score=year['score'].idxmax()*

*institution=year.loc[score,'institution']*

*print("institution with the highest score in the year 2014:",institution)*

*else:*

*print("No data available for year 2014")*

**

1. How many institutions have a "world\_rank" greater than 200 in the year 2015?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year =WorldUniversity[ WorldUniversity['year'] ==2015]*

*if not year.empty:*

*world\_rank= year[year['world\_rank'] > 200]*

*if not world\_rank.empty:*

*institutions = world\_rank['institution'].nunique()*

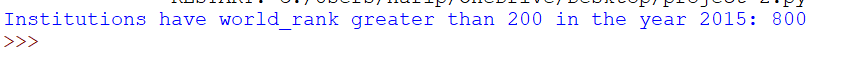
*print("Institutions have world\_rank greater than 200 in the year 2015:",institutions)*

*else:*

*print("No institutions have world\_rank greater than 200")*

*else:*

*print("No data available for year 2015")*

**

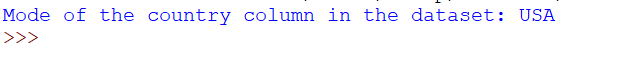
1. What is the mode of the "country" column in the dataset?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*country = WorldUniversity['country'].mode()[0]*

*print("Mode of the country column in the dataset:",country)*

**

1. Identify the institution with the highest "patent" in the year 2011

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year=WorldUniversity[WorldUniversity['year']==2011]*

*if not year.empty:*

*patent=year['patents'].idxmax()*

*if not pd.isna(patent):*

*institution=year.loc[patent,'institution']*

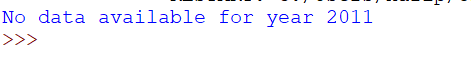
*print("Institution with the highest patent in the year 2011 is:",institution)*

*else:*

*print("No patents in year 2011")*

*else:*

*print("No data available for year 2011")*

**

1. Calculate the average "citations" score for institutions in the bottom 10 of "world\_rank" in 2013.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year=WorldUniversity[WorldUniversity['year']==2013]*

*if not year.empty:*

*world\_rank=year.sort\_values(by='world\_rank',ascending=False)*

*bottom\_10=world\_rank.tail(10)*

*average=bottom\_10['citations'].mean()*

*print("Average citations for institutions in the bottom 10 of world\_rank in 2013 are:",average)*

*else:*

*print("No data available for year 2011")*

**

1. How many institutions have a "score" greater than 80 in the year 2016?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year=WorldUniversity[WorldUniversity['year']==2016]*

*if not year.empty:*

*score=year[year['score'] > 80]*

*institutions=score['institution'].nunique()*

*print("Number of institutions have a score greater than 80 in the year 2016:",institutions)*

*else:*

*print("No data available for year 2016")*

**

1. Find the country with the lowest average "score" across all years.

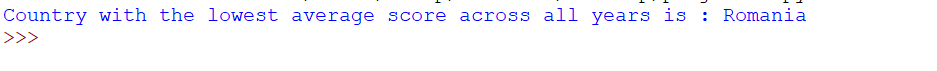
*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*average = WorldUniversity.groupby('country')['score'].mean()*

*low\_average = average.idxmin()*

*print("Country with the lowest average score across all years is :",low\_average)*

**

1. Calculate the average "international\_students" percentage for institutions in the top 5 of "alumni\_employment" in 2014.—*no relevant data in excel*
2. Identify the institution with the lowest score in the year 2015.

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year=WorldUniversity[WorldUniversity['year'] == 2015 ]*

*if not year.empty:*

*score=year['score'].idxmin()*

*if not pd.isna(score):*

*institution=year.loc[score,'institution']*

*print("Institution with the lowest score in the year 2015:",institution)*

*else:*

*print("No scores aviailable")*

*else:*

*print("No data aviailable for year 2015")*

**

1. How many missing values are there in the "broad\_impact" column?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*missing=WorldUniversity['broad\_impact'].isnull().sum()*

*print("Number of missing values are there in the broad\_impact column are:",missing)*

**

1. Find the average "female\_male\_ratio" for institutions in the United States.—*no relevant data in excel*
2. What is the highest "influence" score in the dataset?

*import pandas as pd*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*influence=WorldUniversity['influence'].max()*

*print("Highest influence score in the dataset:",influence)*



**Task 6 :-**

Questions for Visualizations:-

1. How can you use matplotlib to create a bar chart showing the top 10 universities by their scores in 2012?

Answer: Use matplotlib's bar function to plot the scores of the top 10 universities in 2012.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2012]*

*score = year.sort\_values(by='score', ascending=False).head(10)*

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

*plt.bar(score['institution'], score['score'], color='skyblue')*

*plt.xlabel('Institution')*

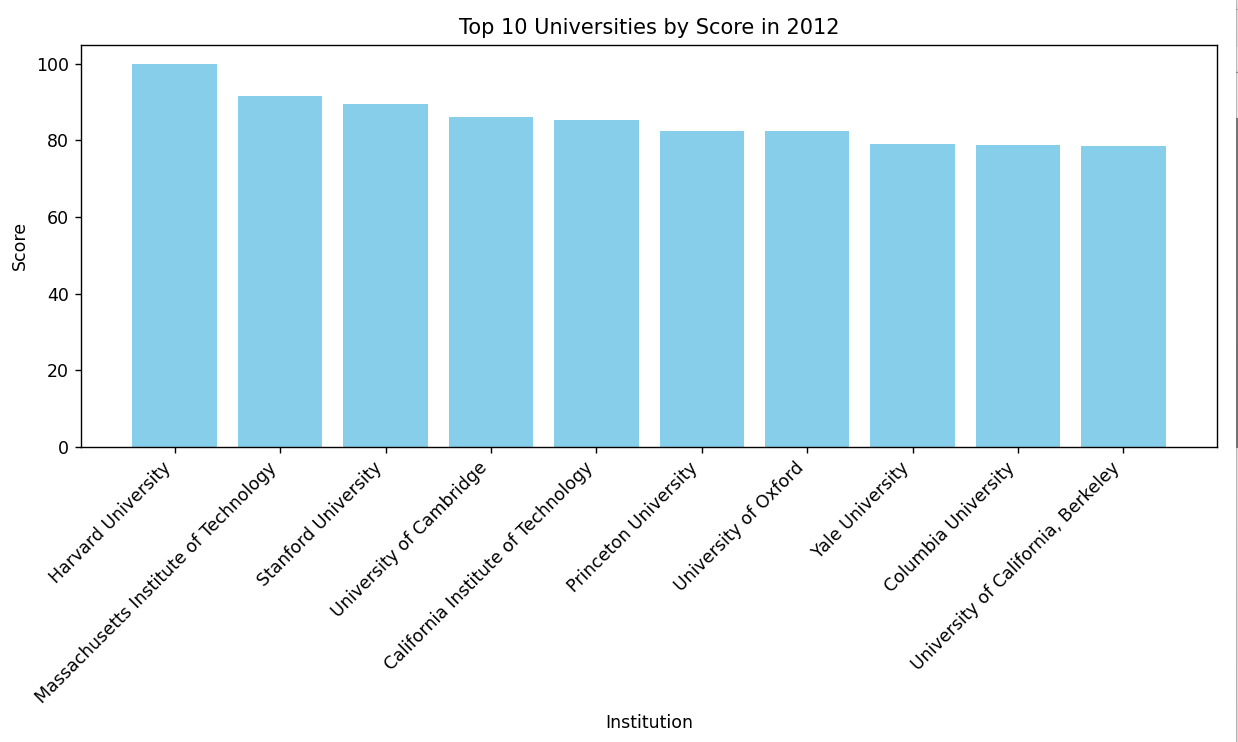
*plt.ylabel('Score')*

*plt.title('Top 10 Universities by Score in 2012')*

*plt.xticks(rotation=45, ha='right')*

*plt.tight\_layout()*

*plt.show()*



1. Create a line plot using seaborn to visualize the trend of scores for the University of Tokyo from 2012 to 2014.

Answer: Use seaborn's lineplot with the appropriate filter for the University of Tokyo to show the trend over the years.

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*tokyo\_data = WorldUniversity[(WorldUniversity['institution'] == 'University of Tokyo') & WorldUniversity['year'].isin([2012, 2013, 2014])]*

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

*sns.lineplot(data=tokyo\_data, x='year', y='score', marker='o', color='skyblue')*

*plt.title('Trend of Scores for University of Tokyo (2012-2014)')*

*plt.xlabel('Year')*

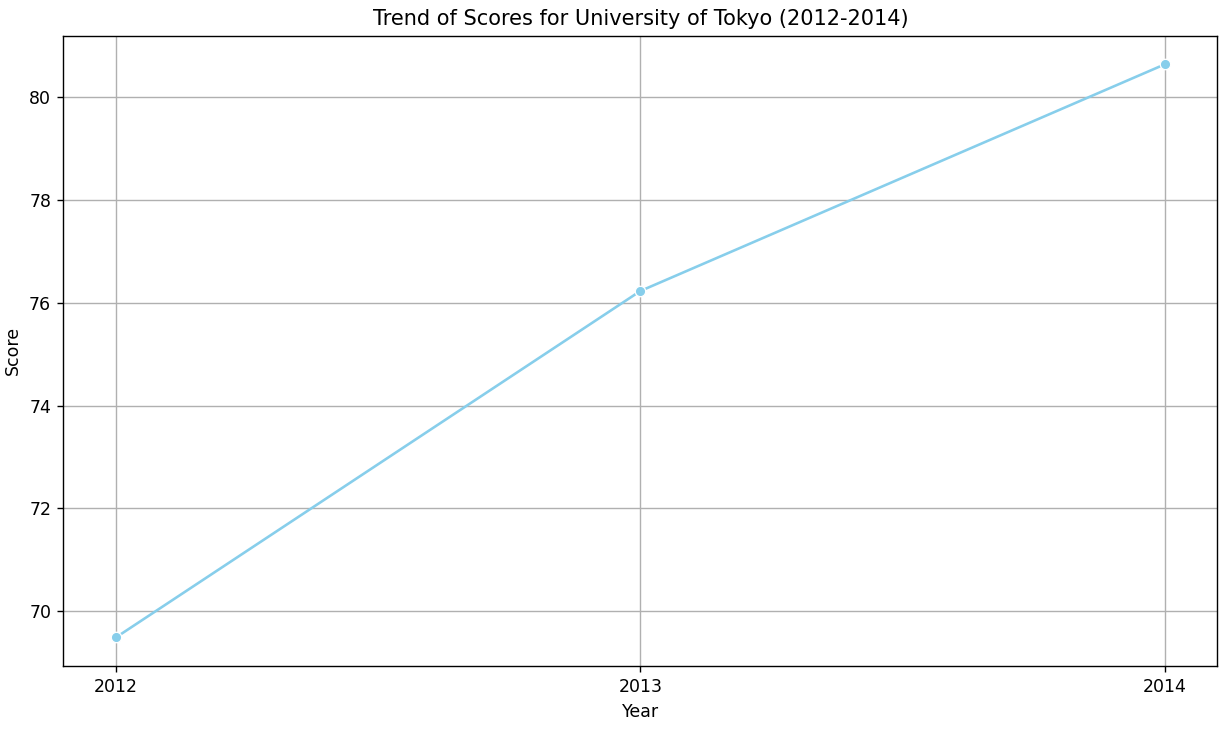
*plt.ylabel('Score')*

*plt.grid(True)*

*plt.xticks(range(2012, 2015))*

*plt.tight\_layout()*

*plt.show()*

**

1. How can you use Altair to create a scatter plot comparing the quality of faculty and alumni employment for all universities in 2013?

Answer: Use Altair's Chart and encode functions to create a scatter plot with quality of faculty on the x-axis and alumni employment on the y-axis.

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2013]*

*scatter\_plot = alt.Chart(year).mark\_circle(size=60).encode(*

*x='quality\_of\_faculty',*

*y='alumni\_employment',*

*tooltip=['institution', 'quality\_of\_faculty', 'alumni\_employment']*

*).properties(*

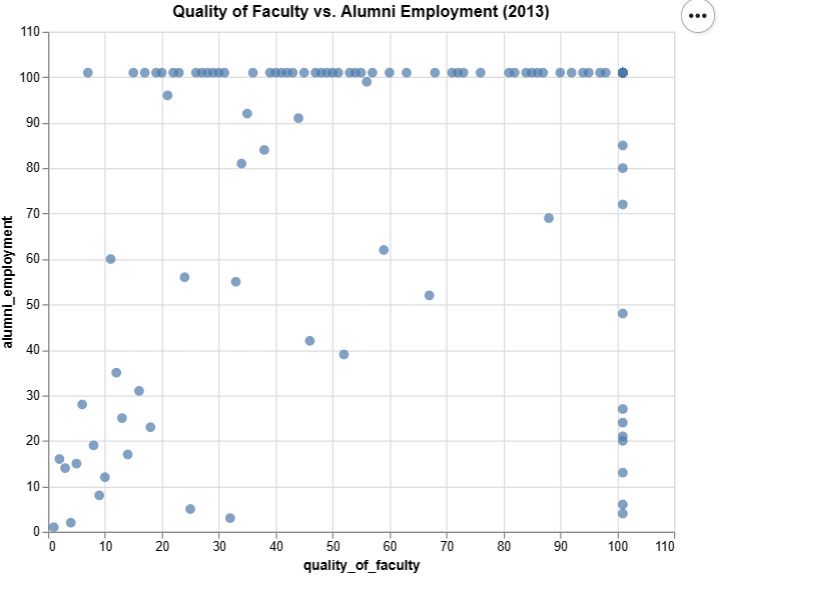
*width=500,*

*height=400,*

*title='Quality of Faculty vs. Alumni Employment (2013)'*

*).interactive()*

*scatter\_plot.save('C:/Users/harip/OneDrive/Desktop/scatter\_plot.html')*

**

1. Using matplotlib, create a horizontal bar chart to display the top 10 universities in terms of influence in 2014.

Answer: Utilize matplotlib's barh function to create a horizontal bar chart for the top 10 universities based on influence in 2014.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

*top\_10 = year.sort\_values(by='influence',ascending=False).head(10)*

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

*plt.barh(top\_10['institution'], top\_10['influence'], color='skyblue')*

*plt.xlabel('Influence')*

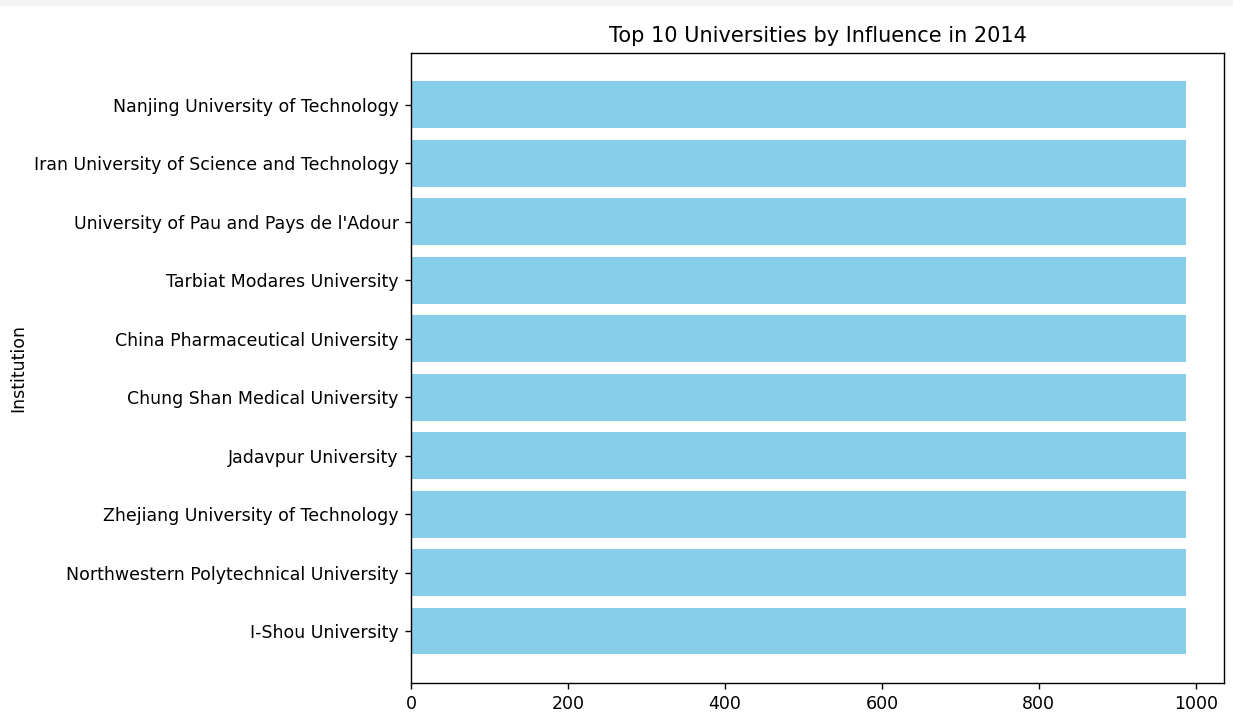
*plt.ylabel('Institution')*

*plt.title('Top 10 Universities by Influence in 2014')*

*plt.gca().invert\_yaxis()*

*plt.tight\_layout()*

*plt.show()*



1. How can you use seaborn to create a boxplot for the distribution of scores among universities in 2012?

Answer: Employ seaborn's boxplot function, specifying the dataset and the variable to be plotted (scores) to visualize the distribution among universities in 2012.

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2012]*

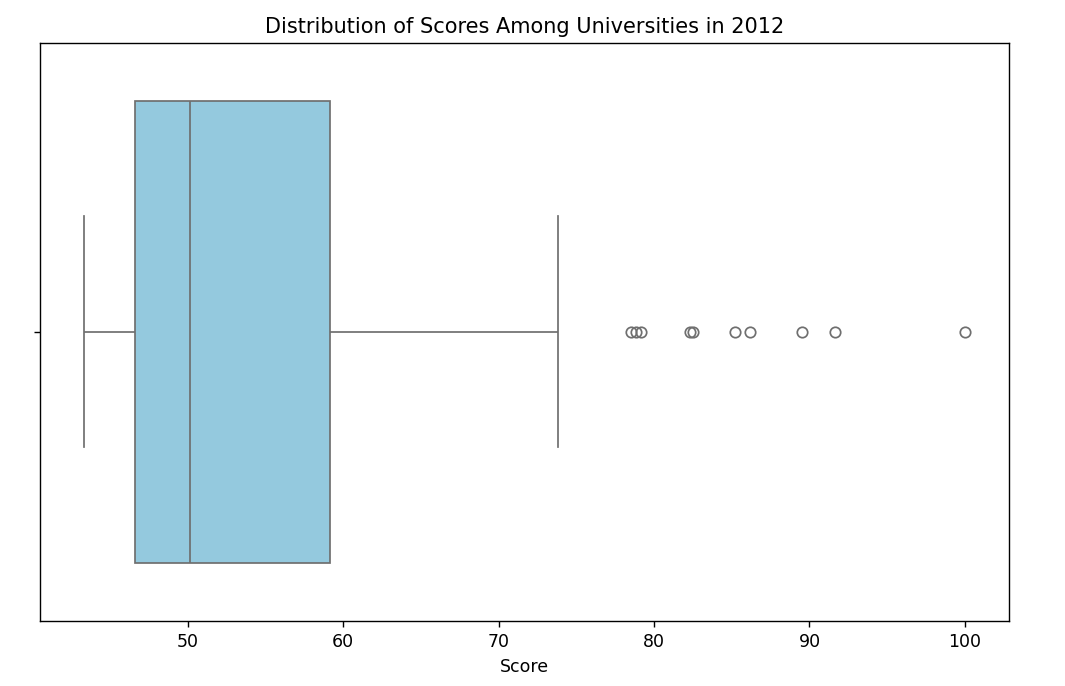
*plt.figure(figsize=(10, 6))*

*sns.boxplot(x='score', data=year, color='skyblue')*

*plt.xlabel('Score')*

*plt.title('Distribution of Scores Among Universities in 2012')*

*plt.show()*

**

1. Create a stacked area plot using Altair to represent the change in scores for the top 5 universities from 2012 to 2014.

Answer: Use Altair to create a layered area plot with the scores on the y-axis, years on the x-axis, and different colors for each university.

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*top\_5 = WorldUniversity[WorldUniversity['year'] == 2012].nlargest(5, 'score')*

*top\_5\_universities = WorldUniversity[WorldUniversity['institution'].isin(top\_5['institution'])]*

*bar\_chart = alt.Chart(top\_5\_universities).mark\_bar().encode(*

*x=alt.X('year:O', title='Year'),*

*y=alt.Y('score:Q', title='Score', stack='normalize'),*

*color='institution:N',*

*tooltip=['institution', 'year', 'score']*

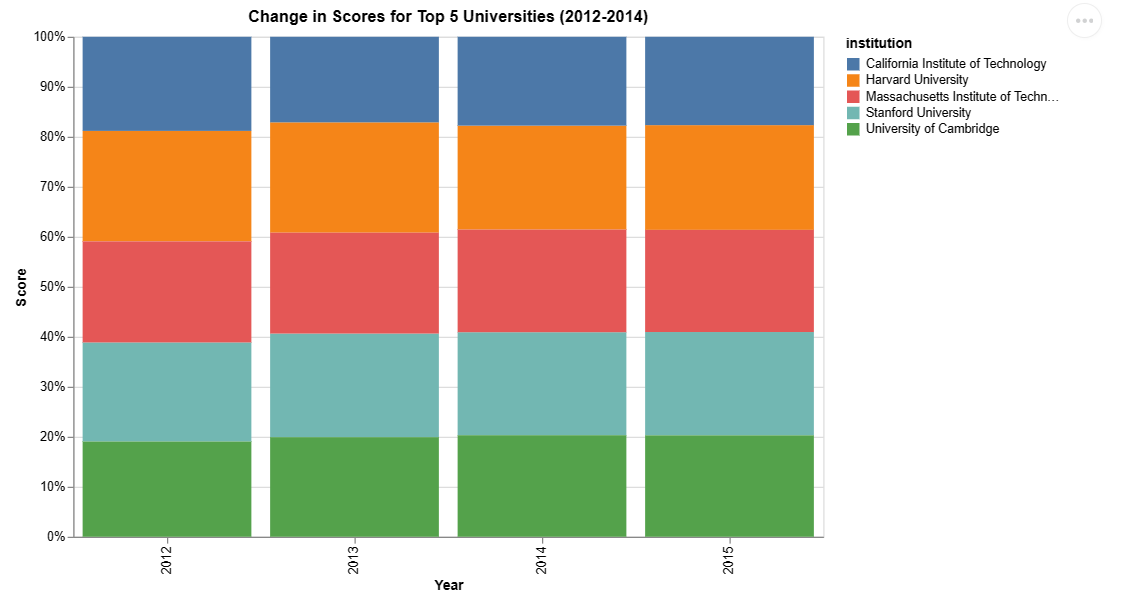
*).properties(*

*width=600,*

*height=400,*

*title='Change in Scores for Top 5 Universities (2012-2014)'*

*).interactive()*

*bar\_chart.save('stacked\_bar\_chart.html')*

1. How can you use matplotlib to create a pie chart illustrating the distribution of universities in the United States and other countries in 2013?

Answer: Use matplotlib's pie function to create a pie chart with data grouped by country for the year 2013.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2013]*

*country = year['country'].value\_counts()*

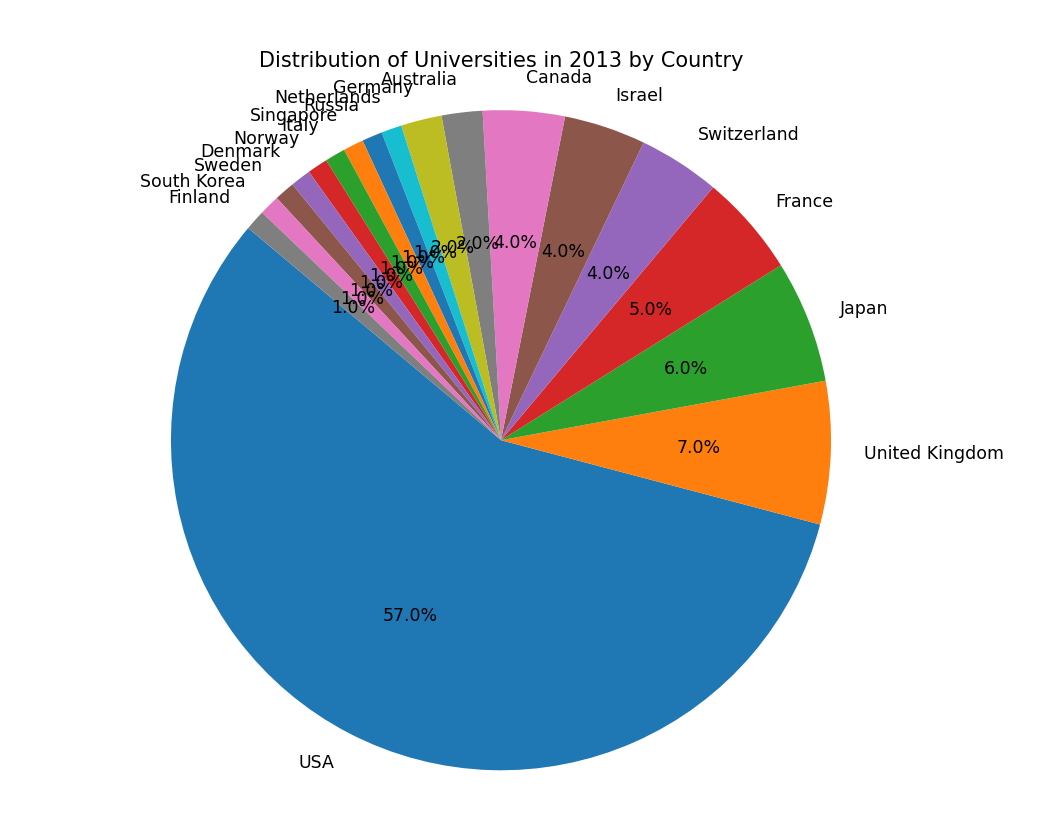
*plt.figure(figsize=(10, 8))*

*plt.pie(country, labels=country.index, autopct='%1.1f%%', startangle=140)*

*plt.title('Distribution of Universities in 2013 by Country')*

*plt.axis('equal')*

*plt.show()*

**

1. Using seaborn, create a violin plot to show the distribution of alumni employment scores among universities in 2014.

Answer: Employ seaborn's violinplot with alumni employment scores as the variable to show the distribution among universities in 2014.

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

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

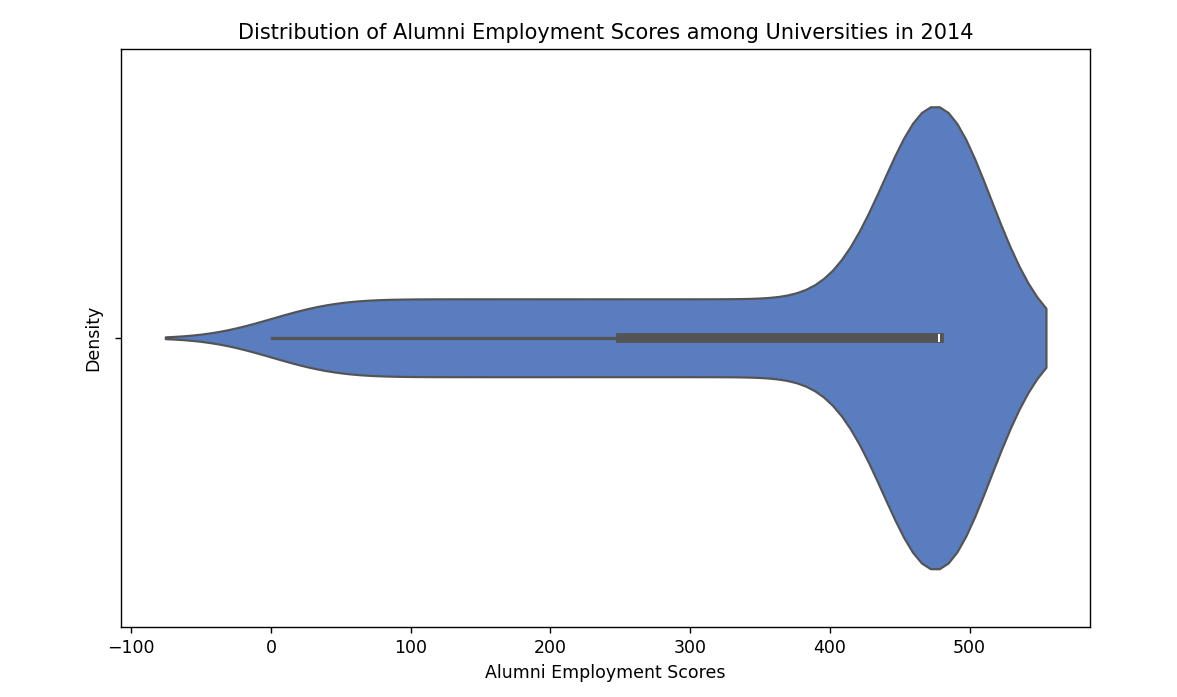
*sns.violinplot(x='alumni\_employment', data=year, palette='muted',legend=False)*

*plt.xlabel('Alumni Employment Scores')*

*plt.ylabel('Density')*

*plt.title('Distribution of Alumni Employment Scores among Universities in 2014')*

*plt.show()*

**

1. How can you use Altair to create a bar chart for the top 10 universities with the highest scores in 2014?

Answer: Use Altair to create a bar chart by encoding the university names on the x-axis and scores on the y-axis, then sorting the data by scores and selecting the top 10.

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

*top\_10 = year.nlargest(10, 'score')*

*bar\_chart = alt.Chart(top\_10).mark\_bar().encode(*

*x='score:Q',*

*y=alt.Y('institution:N', sort='-x'),*

*tooltip=['institution', 'score']*

*).properties(*

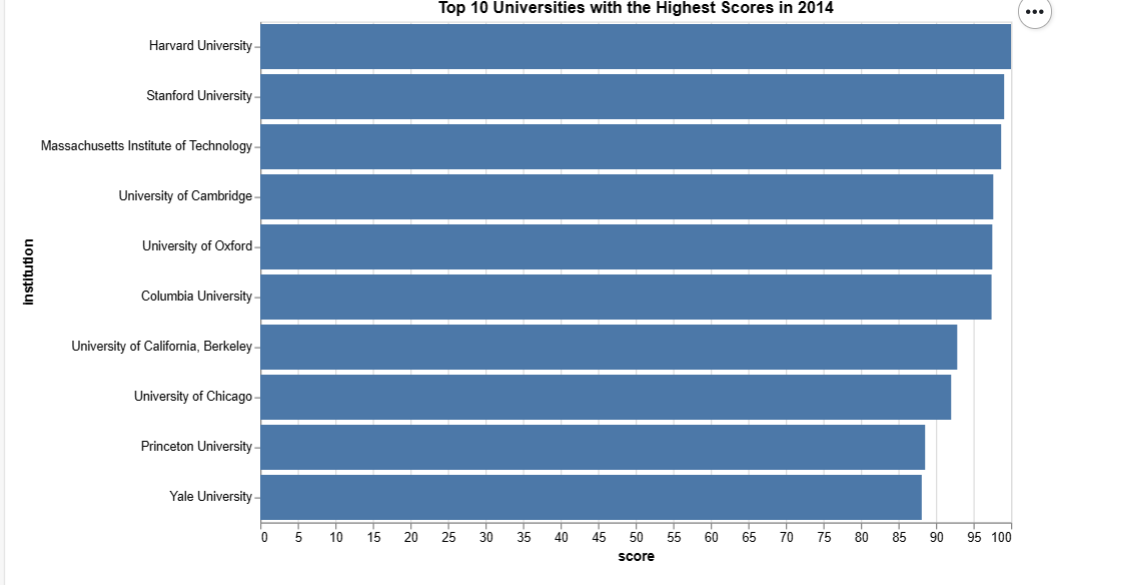
*width=600,*

*height=400,*

*title='Top 10 Universities with the Highest Scores in 2014'*

*)*

*bar\_chart.save('top\_10\_universities\_2014.html')*



1. Create a scatter plot using matplotlib to visualize the correlation between the quality of education and the quality of faculty for all universities in 2012.

Answer: Use matplotlib's scatter function to plot the quality of education against the quality of faculty for each university in 2012.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2012]*

*quality\_education = year['quality\_of\_education']*

*quality\_faculty = year['quality\_of\_faculty']*

*plt.figure(figsize=(8, 6))*

*plt.scatter(quality\_education, quality\_faculty, color='blue', alpha=0.5)*

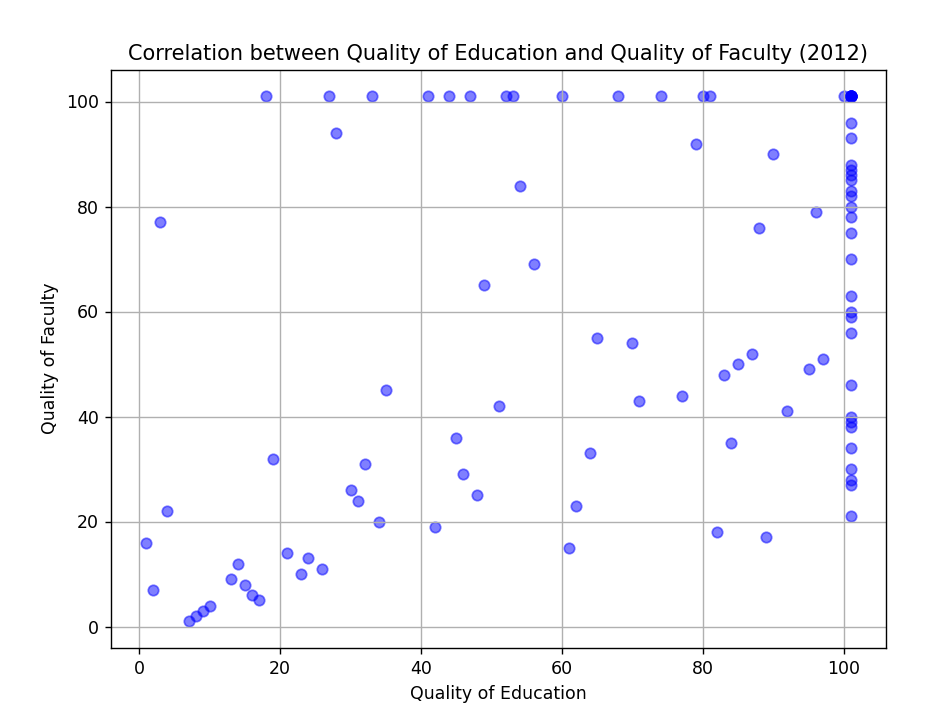
*plt.title('Correlation between Quality of Education and Quality of Faculty (2012)')*

*plt.xlabel('Quality of Education')*

*plt.ylabel('Quality of Faculty')*

*plt.grid(True)*

*plt.show()*

**

1. How can you use seaborn to create a pair plot to visualize the relationships between the scores, influence, and citations for the top 10 universities in 2013?

Answer: Use seaborn's pairplot with the relevant columns filtered for the top 10 universities in 2013.

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2013]*

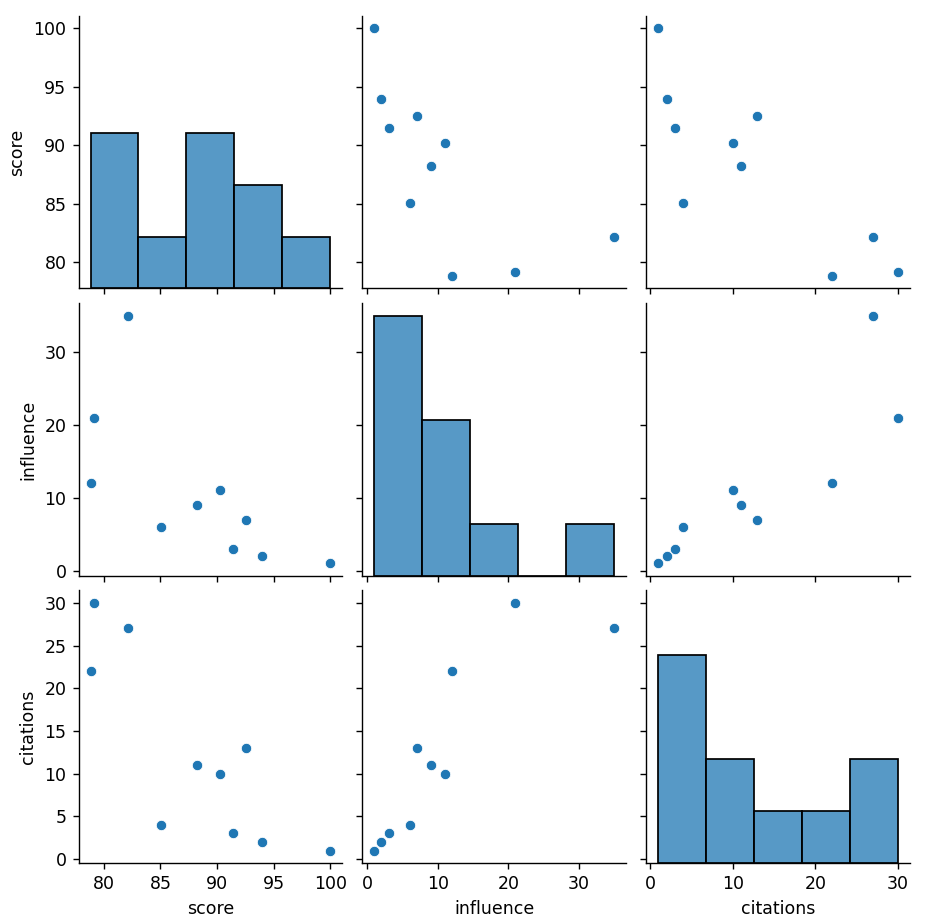
*top\_10\_universities = year.nlargest(10, 'score')*

*data = top\_10\_universities[['score', 'influence', 'citations']]*

*sns.pairplot(data)*

*plt.suptitle('Pair Plot of Scores, Influence, and Citations for Top 10 Universities (2013)', y=1.02)*

*plt.show()*

**

1. Using Altair, create a bar chart to compare the scores of universities in the United States and the United Kingdom in 2014.

Answer: Use Altair to create a bar chart with universities on the x-axis, scores on the y-axis, and color encoding for differentiating between the United States and the United Kingdom.

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

*us\_uk\_universities = year[year['country'].isin(['United States', 'United Kingdom'])]*

*bar\_chart = alt.Chart(us\_uk\_universities).mark\_bar().encode(*

*x='score:Q',*

*y=alt.Y('institution:N', sort='-x'),*

*color='country:N',*

*tooltip=['institution', 'score']*

*).properties(*

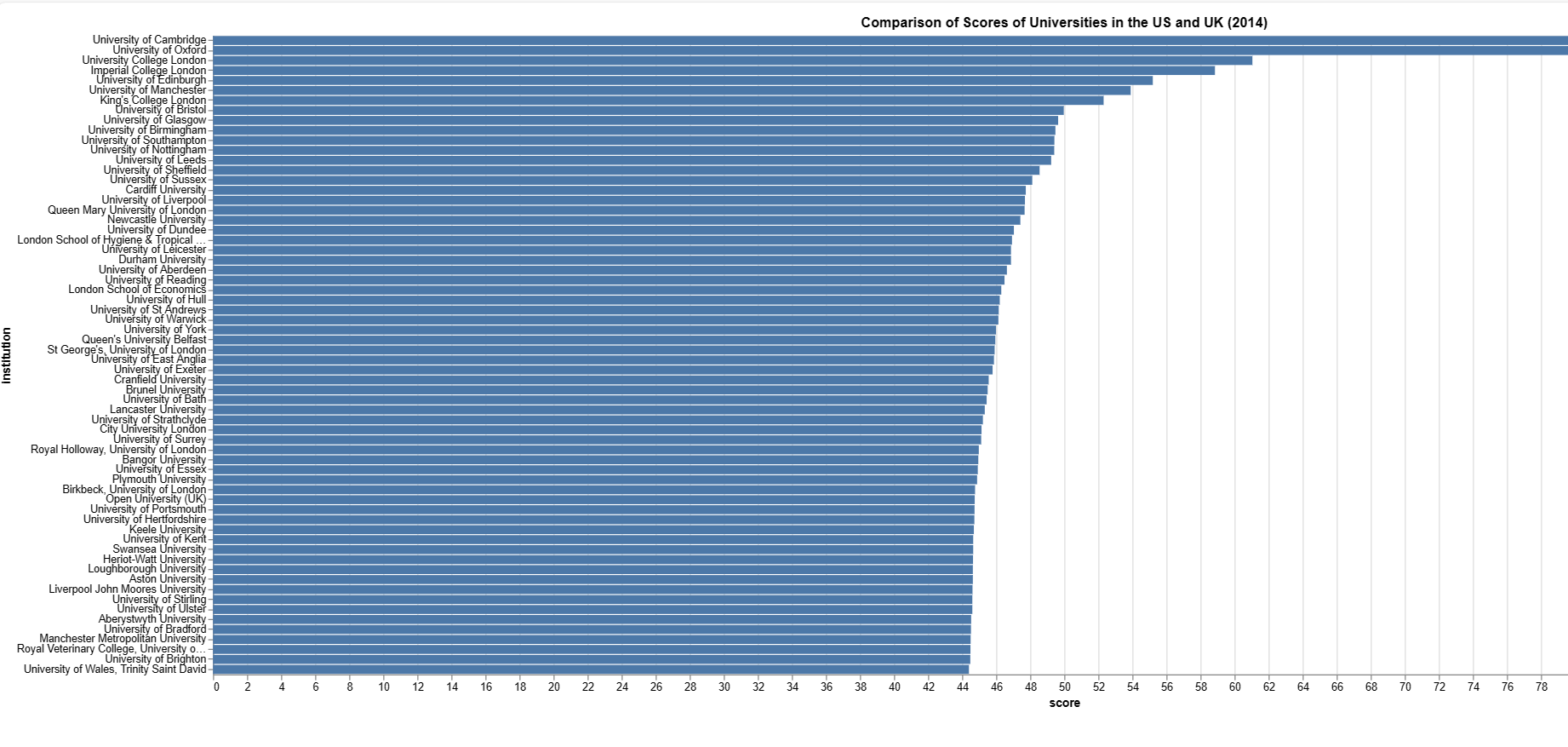
*width=600,*

*height=400,*

*title='Comparison of Scores of Universities in the US and UK (2014)'*

*)*

*bar\_chart.save('us\_uk\_universities\_2014.html')*



1. How can you use matplotlib to create a stacked bar chart illustrating the distribution of universities in different countries in 2012?

Answer: Utilize matplotlib's bar function with the bottom parameter to create a stacked bar chart representing the distribution of universities in different countries in 2012.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2012]*

*country\_counts = year['country'].value\_counts()*

*countries = country\_counts.index.tolist()*

*counts = country\_counts.values.tolist()*

*bottom = None*

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

*for country, count in zip(countries, counts):*

*plt.bar(country, count, bottom=bottom)*

*if bottom is None:*

*bottom = count*

*else:*

*bottom += count*

*plt.xlabel('Country')*

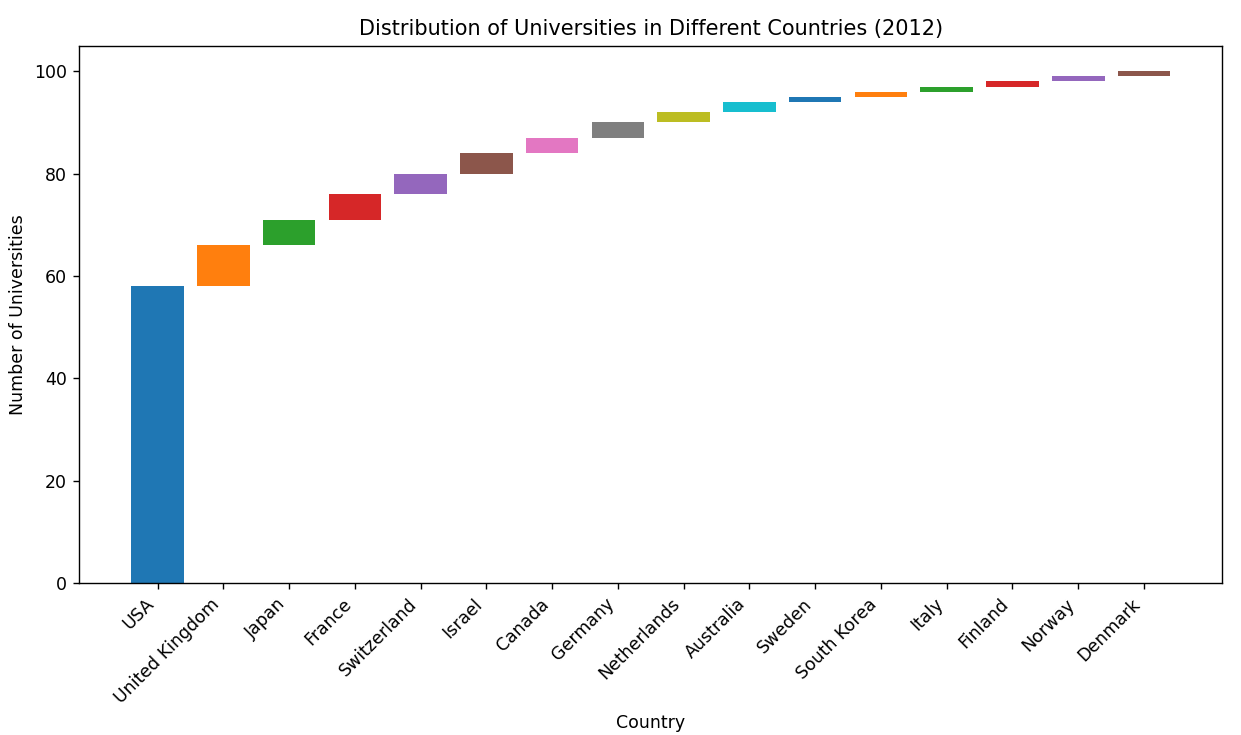
*plt.ylabel('Number of Universities')*

*plt.title('Distribution of Universities in Different Countries (2012)')*

*plt.xticks(rotation=45, ha='right')*

*plt.tight\_layout()*

*plt.show()*



1. Create a heatmap using seaborn to visualize the correlation matrix of the numerical columns in the dataset.

Answer: Use seaborn's heatmap function to plot the correlation matrix of the numerical columns in the dataset.

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*numeric\_columns = WorldUniversity.select\_dtypes(include='number')*

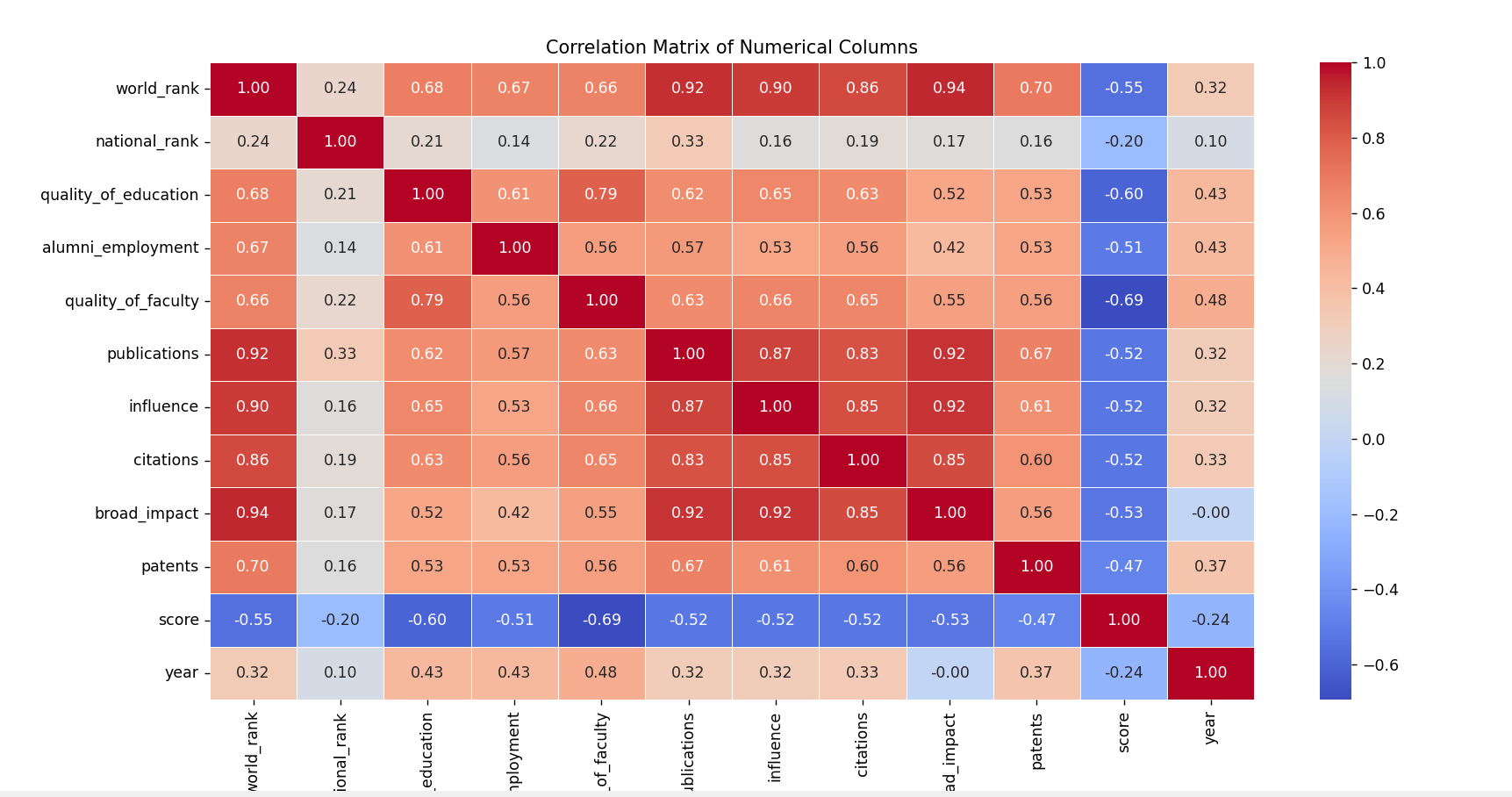
*corr\_matrix = numeric\_columns.corr()*

*plt.figure(figsize=(12, 10))*

*sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)*

*plt.title('Correlation Matrix of Numerical Columns')*

*plt.show()*



1. How can you use Altair to create a line chart showing the trend of scores for the top 5 universities in 2014?

Answer: Use Altair to create a line chart with years on the x-axis, scores on the y-axis, and different lines for each of the top 5 universities.

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

*top\_5 = year.sort\_values(by='score', ascending=False).head(5)*

*line\_chart = alt.Chart(top\_5).mark\_line().encode(*

*x='year:N', # Assuming 'year' is the column name for years*

*y='score:Q', # Assuming 'score' is the column name for scores*

*color='institution:N' # Assuming 'institution' is the column name for university names*

*).properties(*

*width=600,*

*height=400,*

*title='Trend of Scores for Top 5 Universities in 2014'*

*)*

*line\_chart.save('top\_5\_universities\_scores\_2014.html')*

1. Using matplotlib, create a bar chart to show the average scores for universities in each country in 2013.

Answer: Use matplotlib's bar function to create a bar chart with countries on the x-axis, average scores on the y-axis, and data grouped by country for the year 2013.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year\_2013 = WorldUniversity[WorldUniversity['year'] == 2013]*

*average\_scores = year\_2013.groupby('country')['score'].mean().reset\_index()*

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

*plt.bar(average\_scores['country'], average\_scores['score'])*

*plt.xlabel('Country')*

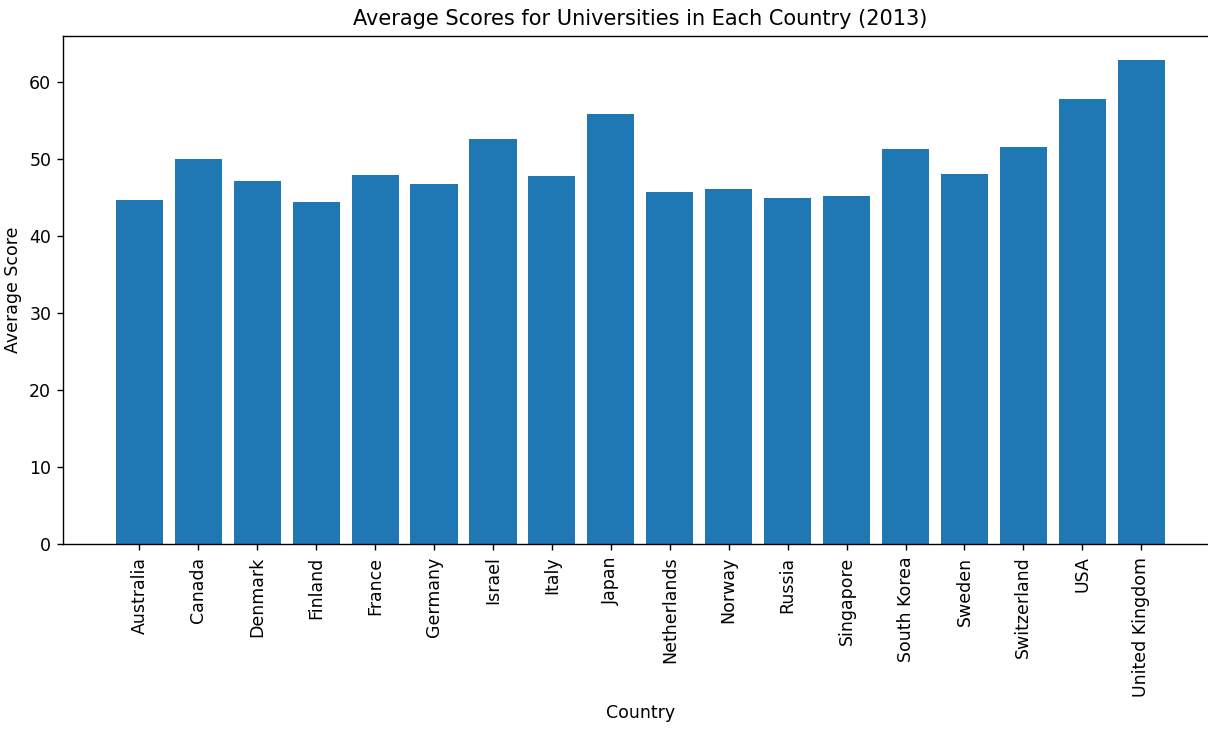
*plt.ylabel('Average Score')*

*plt.title('Average Scores for Universities in Each Country (2013)')*

*plt.xticks(rotation=90)*

*plt.tight\_layout()*

*plt.show()*

**

1. How can you use seaborn to create a swarm plot to visualize the distribution of scores among universities in 2014?

Answer: Utilize seaborn's swarmplot with scores as the variable to show the distribution among universities in 2014.

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

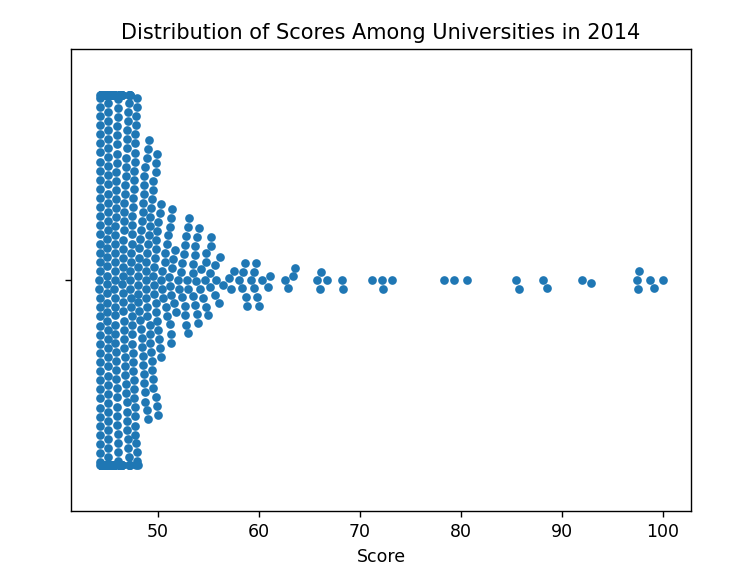
*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*sns.swarmplot(x='score', data=WorldUniversity[WorldUniversity['year'] == 2014])*

*plt.title('Distribution of Scores Among Universities in 2014')*

*plt.xlabel('Score')*

*plt.show()*

**

1. Create a treemap using Altair to represent the proportion of universities in each country in 2012.

Answer: Use Altair to create a treemap with countries as the hierarchy and the size of each square representing the proportion of universities in each country in 2012.

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*data\_2012 = WorldUniversity[WorldUniversity['year'] == 2012]*

*country\_counts = data\_2012['country'].value\_counts().reset\_index()*

*country\_counts.columns = ['country', 'count']*

*bar\_chart = alt.Chart(country\_counts).mark\_bar().encode(*

*x='count:Q',*

*y=alt.Y('country:N', sort='-x'),*

*tooltip=['country:N', 'count:Q']*

*).properties(*

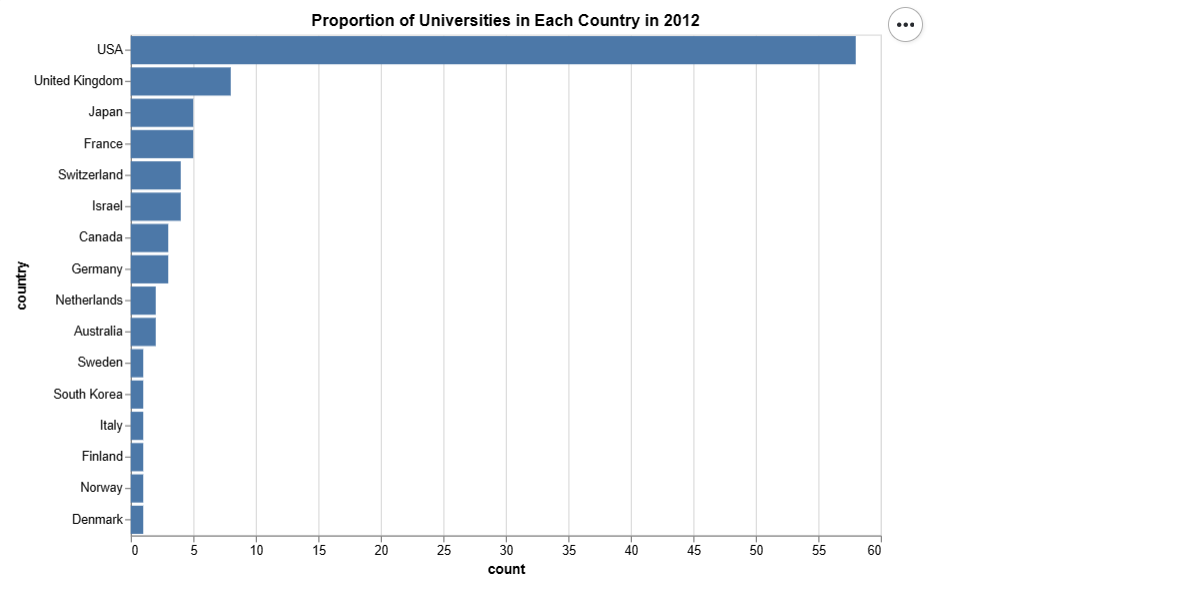
*width=600,*

*height=400,*

*title='Proportion of Universities in Each Country in 2012'*

*)*

*bar\_chart.save('hierarchical\_bar\_chart\_universities\_2012.html')*



1. How can you use matplotlib to create a histogram showing the distribution of scores for all universities in 2014?

Answer: Use matplotlib's hist function to create a histogram representing the distribution of scores for all universities in 2014.

*import pandas as pd*

*import matplotlib.pyplot as plt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[WorldUniversity['year'] == 2014]*

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

*plt.hist(year['score'], bins=20, color='skyblue', edgecolor='black')*

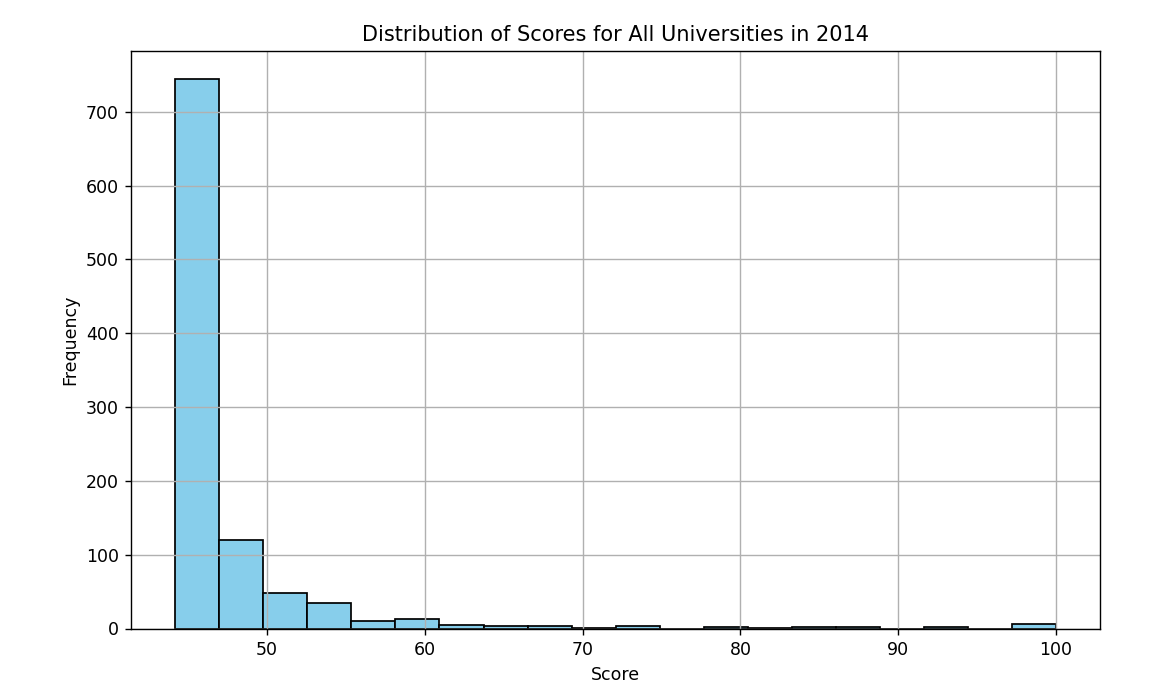
*plt.title('Distribution of Scores for All Universities in 2014')*

*plt.xlabel('Score')*

*plt.ylabel('Frequency')*

*plt.grid(True)*

*plt.show()*



1. Using Altair, create a bar chart to compare the influence scores of universities in Canada and Australia in 2013.

Answer: Use Altair to create a bar chart with universities on the x-axis, influence scores on the y-axis, and color encoding for differentiating between Canada and Australia for the year 2013

*import pandas as pd*

*import altair as alt*

*WorldUniversity = pd.read\_excel('WorldUniversity.xlsx')*

*year = WorldUniversity[(WorldUniversity['year'] == 2013) &*

*(WorldUniversity['country'].isin(['Canada', 'Australia']))]*

*bar\_chart = alt.Chart(year).mark\_bar().encode(*

*x=alt.X('institution:N', title='University'),*

*y=alt.Y('influence:Q', title='Influence Score'),*

*color='country:N',*

*tooltip=['institution', 'influence']*

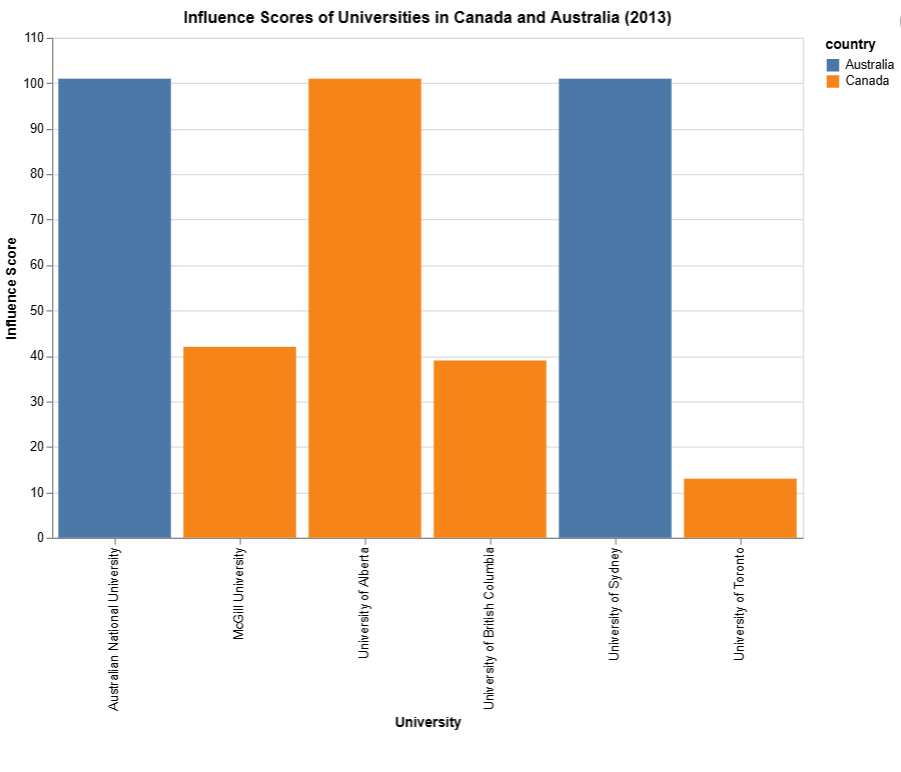
*).properties(*

*width=600,*

*height=400,*

*title='Influence Scores of Universities in Canada and Australia (2013)'*

*)*

*bar\_chart.save('influence\_scores\_CA\_AU\_2013.html')*

**Task 7 :-**

**Solve the Questions Using Dax and Power BI Visuals**

Open the Power BI , Import the given Datasets and Solve the below Questions:-

Q1) What is the overall trend of university scores over the years?

Q2)How does the number of publications vary across the top 10 universities in 2012?

Total Publications = SUM('world\_ranking\_universities'[publications])

Rank = RANKX(ALL('world\_ranking\_universities'[institution]), [Total Publications], , DESC)

Top 10 Institutions =

TOPN(10,

     FILTER(

         ALL('world\_ranking\_universities'[institution]),

         [Rank] <= 10

     ),

     [Total Publications], DESC

)

Q3) What is the average influence score for universities in the United States in 2013?

*Applied filters for US and year as 2013*

AverageInfluenceScore = AVERAGE('world\_ranking\_universities (2)'[influence])

Q4) How does the quality of faculty correlate with alumni employment?

Calc column:

Standardized\_Employment =

VAR Avg\_Alumni\_Employment = AVERAGE('world\_ranking\_universities'[alumni\_employment])

VAR StdDev\_Alumni\_Employment = STDEV.P('world\_ranking\_universities'[alumni\_employment])

RETURN

    DIVIDE('world\_ranking\_universities'[alumni\_employment] - Avg\_Alumni\_Employment, StdDev\_Alumni\_Employment)

Standardized\_Quality =

VAR Avg\_Quality\_of\_Faculty = AVERAGE('world\_ranking\_universities'[quality\_of\_faculty])

VAR StdDev\_Quality\_of\_Faculty = STDEV.P('world\_ranking\_universities'[quality\_of\_faculty])

RETURN

    DIVIDE('world\_ranking\_universities'[quality\_of\_faculty] - Avg\_Quality\_of\_Faculty, StdDev\_Quality\_of\_Faculty)

DAX:

Correlation =

VAR Avg\_Quality\_of\_Faculty = AVERAGE('world\_ranking\_universities'[Standardized\_Quality])

VAR Avg\_Alumni\_Employment = AVERAGE('world\_ranking\_universities'[Standardized\_Employment])

RETURN

    DIVIDE(

        SUMX(

            'world\_ranking\_universities',

            ('world\_ranking\_universities'[Standardized\_Quality] - Avg\_Quality\_of\_Faculty) \* ('world\_ranking\_universities'[Standardized\_Employment] - Avg\_Alumni\_Employment)

        ),

        SQRT(

            SUMX('world\_ranking\_universities', POWER('world\_ranking\_universities'[Standardized\_Quality] - Avg\_Quality\_of\_Faculty, 2)) \*

            SUMX('world\_ranking\_universities', POWER('world\_ranking\_universities'[Standardized\_Employment] - Avg\_Alumni\_Employment, 2))

        )

    )

Q5) Which country has the highest average score in 2014?

*Used filter pane to get the result*

Q6) What is the percentage change in citations from 2012 to 2013 for the top 5 universities?

RankByCitations2012 =

RANKX(

    FILTER('world\_ranking\_universities', 'world\_ranking\_universities'[year] = 2012),

    CALCULATE(SUM('world\_ranking\_universities'[citations]), 'world\_ranking\_universities'[year] = 2012),

    ,

    DESC

)

CitationsIn2013\_Top5 =

CALCULATE(

    SUM('world\_ranking\_universities'[citations]),

    TOPN(5, 'world\_ranking\_universities', [RankByCitations2012], DESC),

    'world\_ranking\_universities'[year] = 2013

)

PercentageChange =

DIVIDE(

    [CitationsIn2013\_Top5] - CALCULATE(SUM('world\_ranking\_universities'[citations]), 'world\_ranking\_universities'[year] = 2012),

    CALCULATE(SUM('world\_ranking\_universities'[citations]), 'world\_ranking\_universities'[year] = 2012)

) \* 100

Q7) Which university has the highest quality of education in 2014?

AvgQualityOfEducation2014 =

CALCULATE(

    AVERAGE('world\_ranking\_universities'[quality\_of\_education]),

    'world\_ranking\_universities'[year] = 2014

)

UniversityWithHighestQuality2014 =

MAXX(

    TOPN(1,

        FILTER('world\_ranking\_universities', 'world\_ranking\_universities'[year] = 2014),

        'world\_ranking\_universities'[quality\_of\_education], DESC

    ),

    'world\_ranking\_universities'[institution]

)

Q8) What is the average score for universities in the United States in each year?

AverageScoreUSByYear =

CALCULATE(

    AVERAGE('world\_ranking\_universities'[score]),

    FILTER('world\_ranking\_universities', 'world\_ranking\_universities'[country] = "USA"),

    ALLEXCEPT('world\_ranking\_universities', 'world\_ranking\_universities'[year])

)

Q9) How many universities from the United Kingdom are in the top 20 in 2013?

UKUniversitiesTop20\_2013 =

CALCULATE(

    COUNTROWS('world\_ranking\_universities'),

    FILTER(

        'world\_ranking\_universities',

        'world\_ranking\_universities'[year] = 2013 &&

        'world\_ranking\_universities'[country] = "United Kingdom" &&

        'world\_ranking\_universities'[world\_rank] <= 20

    )

)

Q10) What is the rank of the University of Tokyo in 2014?

RankUniversityOfTokyo\_2014 =

CALCULATE(

    FIRSTNONBLANK('world\_ranking\_universities'[world\_rank], 1),

    FILTER(

        'world\_ranking\_universities',

        'world\_ranking\_universities'[year] = 2014 &&

        'world\_ranking\_universities'[institution] = "University of Tokyo"

    )

)

Q11) Which country has the most universities in the top 50 in 2012?

CountryWithMostTop50\_2012 =

MAXX(

    TOPN(

        1,

        SUMMARIZE(

            FILTER(

                'world\_ranking\_universities',

                'world\_ranking\_universities'[year] = 2012 &&

                'world\_ranking\_universities'[world\_rank] <= 50

            ),

            'world\_ranking\_universities'[country],

            "Count", COUNTROWS('world\_ranking\_universities')

        ),

        [Count], DESC

    ),

    [country]

)

Q12) What is the average score for universities with a high number of publications in 2014?

AverageScoreUSByYear =

CALCULATE(

    AVERAGE('world\_ranking\_universities'[score]),

    FILTER('world\_ranking\_universities', 'world\_ranking\_universities'[country] = "USA"),

    ALLEXCEPT('world\_ranking\_universities', 'world\_ranking\_universities'[year])

)

Q13) How does the number of patents correlate with the broad impact score?

Covariance =

VAR PatentsAvg = AVERAGE('world\_ranking\_universities'[patents])

VAR BroadImpactAvg = AVERAGE('world\_ranking\_universities'[broad\_impact])

VAR Deviations =

    SUMX(

        'world\_ranking\_universities',

        ('world\_ranking\_universities'[patents] - PatentsAvg) \* ('world\_ranking\_universities'[broad\_impact] - BroadImpactAvg)

    )

RETURN

    Deviations / COUNTROWS('world\_ranking\_universities')

CorrelationCoefficient =

DIVIDE(

    SUMX(

        FILTER(

            'world\_ranking\_universities',

            NOT(ISBLANK('world\_ranking\_universities'[broad\_impact]))

        ),

        ('world\_ranking\_universities'[patents] - AVERAGE('world\_ranking\_universities'[patents])) \* ('world\_ranking\_universities'[broad\_impact] - AVERAGE('world\_ranking\_universities'[broad\_impact]))

    ),

    SQRT(

        AVERAGEX(

            FILTER(

                'world\_ranking\_universities',

                NOT(ISBLANK('world\_ranking\_universities'[broad\_impact]))

            ),

            POWER('world\_ranking\_universities'[patents] - AVERAGE('world\_ranking\_universities'[patents]), 2)

        )

        \*

        AVERAGEX(

            FILTER(

                'world\_ranking\_universities',

                NOT(ISBLANK('world\_ranking\_universities'[broad\_impact]))

            ),

            POWER('world\_ranking\_universities'[broad\_impact] - AVERAGE('world\_ranking\_universities'[broad\_impact]), 2)

        )

    )

)

Q14) What is the percentage change in world rank for the top 10 universities from 2012 to 2014?

Ranking2012 =

RANKX(

    FILTER('world\_ranking\_universities', 'world\_ranking\_universities'[year] = 2012),

    CALCULATE(MAX('world\_ranking\_universities'[world\_rank]))

)

Ranking2014 =

RANKX(

    FILTER('world\_ranking\_universities', 'world\_ranking\_universities'[year] = 2014),

    CALCULATE(MAX('world\_ranking\_universities'[world\_rank]))

)

PercentageChange =

VAR Rank2012 = [Ranking2012]

VAR Rank2014 = [Ranking2014]

RETURN

DIVIDE(Rank2014 - Rank2012, ABS(Rank2012)) \* 100

Q15) Which universities have shown an improvement in influence score from 2013 to 2014?

InfluenceScore2013 =

CALCULATE(

    AVERAGE('world\_ranking\_universities'[influence]),

    'world\_ranking\_universities'[year] = 2013

)

InfluenceScore2014 =

CALCULATE(

    AVERAGE('world\_ranking\_universities'[influence]),

    'world\_ranking\_universities'[year] = 2014

)

ImprovementInfluence =

IF(

    [InfluenceScore2014] > [InfluenceScore2013],

    "Improved",

    "Not Improved"

)

Q16) What is the total number of universities included in the dataset?

TotalNumberOfUniversities = DISTINCTCOUNT('world\_ranking\_universities'[institution])

Q17) Which universities have a score above the average score in 2013?

UniversitiesAboveAverage2013 =

FILTER(

    'world\_ranking\_universities',

    'world\_ranking\_universities'[year] = 2013 &&

    'world\_ranking\_universities'[score] > CALCULATE(AVERAGE('world\_ranking\_universities'[score]), 'world\_ranking\_universities'[year] = 2013)

)

Q18) How does the number of alumni employed change over the years for the top 5 universities?

AlumniEmploymentRank = RANKX(ALL('world\_ranking\_universities'[institution]), CALCULATE(SUM('world\_ranking\_universities'[alumni\_employment])), , DESC)

Top5Universities = IF('world\_ranking\_universities'[AlumniEmploymentRank] <= 5, 1, 0)

Q19) What is the average percentage change in quality of education for the top 10 universities from 2012 to 2013?

AveragePercentageChange =

VAR Top10Universities2012 =

    TOPN(

        10,

        'world\_ranking\_universities',

        'world\_ranking\_universities'[quality\_of\_education],

        DESC,

        'world\_ranking\_universities'[year] = 2012

    )

VAR Top10Universities2013 =

    TOPN(

        10,

        'world\_ranking\_universities',

        'world\_ranking\_universities'[quality\_of\_education],

        DESC,

        'world\_ranking\_universities'[year] = 2013

    )

VAR PercentageChange =

    AVERAGEX(

        Top10Universities2012,

        DIVIDE(

            'world\_ranking\_universities'[quality\_of\_education],

            CALCULATE(

                MAX('world\_ranking\_universities'[quality\_of\_education]),

                FILTER(

                    Top10Universities2013,

                    'world\_ranking\_universities'[institution] = EARLIER('world\_ranking\_universities'[institution])

                )

            )

        ) - 1

    )

RETURN

    PercentageChange

Q20) Which countries have universities with an average score above 70 in 2014?

CountriesAbove70 =

VAR Above70 =

    FILTER (

        VALUES('world\_ranking\_universities'[country]),

        CALCULATE (

            AVERAGE('world\_ranking\_universities'[score]),

            'world\_ranking\_universities'[year] = 2014

        ) > 70

    )

RETURN

    IF (

        COUNTROWS(Above70) > 0,

        Above70,

        "No countries have universities with an average score above 70 in 2014."

    )

**\*\*\*\*\*\*\*\*\*\*\*Thank You\*\*\*\*\*\*\*\*\*\*\***