Annexure- I

Summer Internship Report

PYTHON, MySQL, DSA, MACHINE LEARNING FOR DATA SCIENTIST

From Edyoda digital University

[By EDYODA]

A Project Report

Submitted in partial fulfillment of the requirements for the award of degree of

Master of Computer Application

[DATA SCIENTIST]

Submitted to

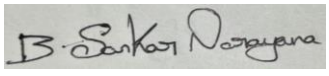
LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

From 06/02/23 to 08/8/23

SUBMITTED BY

Sankar Narayana
Reg. No.: 12217027

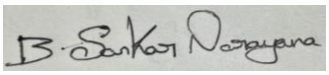


Annexure-II: Student Declaration

To whom so ever it may concern

I, **Sankar Narayana, 12217027,** hereby declare that the work done by me on "**PYTHON, MySQL, DSA, MACHINE LEARNING FOR DATA SCIENTIST**" from **June, 2023** to **July, 2023**, is a record of original work for the partial fulfillment of the requirements for the award of the degree, **Master of Computer Application.**

Name of the Student (Registration Number): Sankar Narayana [12217027]

Signature of the student: 

ACKNOWLEDGMENTS

I have successful completion of this summer internship report would not have been possible without the help and guidance of many individuals and organizations (EDYODA). The tutor “Prof. Pratyush Srivastava” feel especially blessed to have gotten this during my certification period. The tutor would like to take this opportunity to offer my earnest admiration to every one of them.

All thanks to my dear parents for their intense help and support during the period of this certification.

The tutor of Coursera is thankful to my learned and repudiated tutor for his unbeatable personality, kindness; animated support to help readably and greatly improve the quality of my summer Internship Report and brought up to its present status. The tutor whose work have used in this report to depend on different national and international publications for the completion of the certification program through Edyoda.

Thanks to my colleagues who helped me directly or indirectly to manage my work. I am especially grateful to **Prof. Awantik das**(Prof, Edyoda University), has also been a positive and encouraging tutor to help to learn the concept of Python, SQL, and Machine learning.

Finally, I would like to say thank my colleagues and lecturer who helped me a lot in collecting information, data, and guidance me from time to time during this summer internship program, they gave me different ideas in making this project unique.

ABSTRACT

This report is the reflection and the journey of my one-month summer internship program along with the highlights of what I learned through errors, work responsibilities, and the most importance of this summer internship program in Edyoda. The knowledge I have achieved as a developer on front end development, and how to work in an office environment is elegant. My work was to learn and focus on Data scientist assignments which were provided by the tutor of the **Edyoda university.**

In this report, I have focused my work and explain my new learning thing which I have got during my summer internship period at **Edyoda university** The challenge linked with web development to that the tools and techniques used to analysing the data from the data sets. slowly and so the analyst needs to slowly be knowledge that how the field is developing.

This report provides an overview of data scientist and its related technologies. This report includes a discussion of the best practices for data analysis projects and an overview of the tools and technologies used by data analysing in the industry.

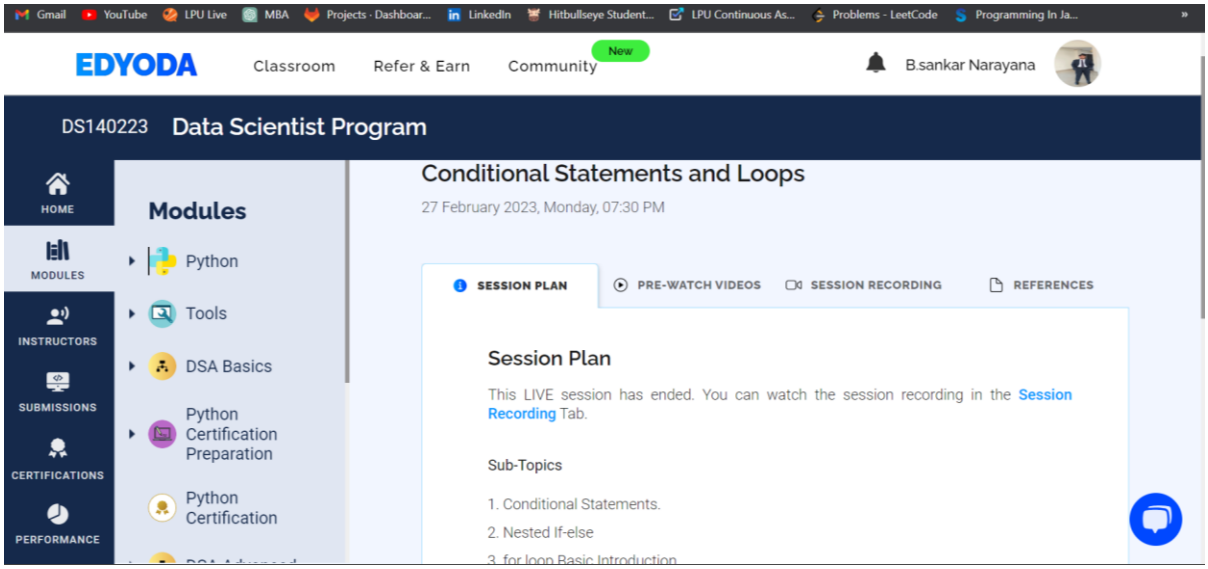
This summer internship that the tutor worked in certainly helped him by increasing the knowledge of Python, SQL, and Machine learning.

Table of contents

1. Introduction

Introduction of the Course:

Data science is an in-demand career path for people with an aptitude for research, programming, math, and computers. Discover real-world applications and job opportunities in data science and what it takes to work in this exciting field.Data science is an interdisciplinary field that uses algorithms, procedures, and processes to examine large amounts of data in order to uncover hidden patterns, generate insights, and direct decision-making. To create prediction models, data scientists use advanced machine learning algorithms to sort through, organize, and learn from structured and unstructured data. As a fast-growing field with applications across numerous industries, data science offers a variety of job opportunities—from researching to computing. In this article, you will learn about how data science is used in the real world, the job outlook for the field, its required skills, and what credentials you need to land a job.



2.Data scientist:

A data scientist is a professional who employs various techniques and methodologies to analyze and interpret complex data sets in order to extract valuable insights, make informed decisions, and solve real-world problems. Data scientists use a combination of skills from different fields, including statistics, computer science, mathematics, domain knowledge, and data visualization, to uncover patterns, trends, and relationships within data.

The main responsibilities of a data scientist typically include:

1. Data Collection and Cleaning: Gathering relevant data from various sources, including databases, APIs, and other data repositories, and ensuring that the data is accurate, consistent, and properly formatted.

2.Data Exploration and Visualization: Using statistical and visualization tools to explore the data, identify patterns, correlations, and anomalies, and present the findings in a visually understandable manner.

3. Statistical Analysis: Applying statistical techniques and models to draw meaningful insights from the data, and using these insights to make predictions and informed decisions.

4. Machine Learning: Developing and implementing machine learning models to predict outcomes, classify data, and solve complex problems. This involves tasks such as feature selection, model training, evaluation, and tuning.

5. Data Interpretation: Translating the technical findings into actionable insights that can guide business strategies, policy decisions, or other relevant actions.

6. Domain Expertise:Understanding the specific domain or industry in which the data is being analyzed, in order to contextualize findings and apply relevant domain-specific knowledge.

7. Communication: Effectively communicating findings and insights to both technical and non-technical stakeholders through reports, presentations, and discussions.

8. Collaboration: Working closely with other team members, such as data engineers, domain experts, and business analysts, to ensure that the data analysis aligns with organizational goals.

Data scientists often work with large and complex datasets, leveraging programming languages like Python or R, along with tools and libraries for data manipulation, analysis, and machine learning, such as Pandas, NumPy, scikit-learn, and TensorFlow. The insights generated by data scientists can have a significant impact on business decisions, product development, research, and various other area

3.Internship Certificate

(As given by MOOC or Organization in original)

Link of certificate: --
https://github.com/Sankardot/python_assignment/commit/8a22cda3b9ae6abe35ec58add1e46e8691f98244

4.Project: Problem for Covid 19 Data Analysis Project using Python

Analysing COVID-19 data for a report requires a more in-depth and structured approach. Here's a detailed outline of how you might structure your COVID-19 data analysis project for a comprehensive report:

1. Executive Summary

Provide a concise overview of the report's objectives, methods, and key findings.

2. Introduction

-Introduce the importance of analyzing COVID-19 data and its relevance to public health and policy-making.

3. Data Sources and Methodology

-Describe the sources of COVID-19 data used in the analysis (e.g., WHO, CDC, national health agencies).

- Explain the methodology for data collection, cleaning, and preprocessing.

- Discuss any limitations of the data, such as reporting delays or inconsistencies.

4. Descriptive Analysis

- Present essential descriptive statistics of COVID-19 data, including total cases, deaths, recoveries, and active cases.

- Use tables and visualizations (bar charts, pie charts) to showcase these statistics.

5. Temporal Trends Analysis

- Display time series graphs illustrating the progression of COVID-19 cases, deaths, and recoveries over time.

- Highlight any significant events or policy changes that may have impacted the trends.

6. Geographic Distribution Analysis

- Utilize maps or heatmaps to showcase the geographic spread of COVID-19 cases, deaths, or other relevant metrics.

- Analyze regional variations and discuss potential factors influencing these disparities.

7. Demographic and Socioeconomic Analysis

- Examine how different age groups, genders, and socioeconomic factors are correlated with COVID-19 outcomes.

- Present infection rates, hospitalizations, mortality rates, etc., for various demographic categories.

8. Testing and Positivity Rates

- Discuss the importance of testing and present trends in testing efforts.

- Calculate and visualize positivity rates over time to understand the progression of the pandemic.

9. Healthcare System Impact

- Analyze hospitalization rates, ICU admissions, ventilator usage, and strain on healthcare systems.

- Discuss the capacity of healthcare facilities and potential implications for medical resources.

10. Vaccination Progress Analysis

- Detail the progress of COVID-19 vaccination campaigns.

- Present vaccination rates, coverage by demographics, and potential effects on disease spread.

11. Comparative Analysis

- Compare COVID-19 data across different countries, regions, or states.

- Analyze differences in policies, healthcare infrastructure, and outcomes.

12. Predictive Analysis (if applicable)

- If you've conducted predictive modeling, explain the methodology and present the results.

- Discuss the accuracy of your predictions and their implications for future scenarios.

13. Key Insights and Implications

- Summarize the main insights derived from your analysis.

- Discuss implications for public health strategies, policy decisions, and future pandemic preparedness.

14. Recommendations

- Provide actionable recommendations based on your analysis.

- Suggest strategies for managing the pandemic, improving healthcare responses, and vaccination campaigns.

15. Conclusion

- Recap the key points of your report.

- Emphasize the ongoing importance of data analysis in understanding and addressing the pandemic.

16. References

- List all the sources of data, research papers, and references used in your analysis.

17. Appendices

- Include additional detailed charts, graphs, or statistical analyses that support your main findings.

Remember to maintain a clear and organized writing style throughout the report. Use headings and subheadings to structure the content, and include visualizations to enhance understanding. Provide thorough explanations for your analysis methods and findings to ensure the report's credibility.

Code of Mini Project:

Problem for Covid 19 Data Analysis Project using Python
Dataset Link:
Url : = https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv
Q1.Import the dataset using Pandas from above mentioned url.

In [11]:

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

```
data =
pd.read_csv('https://raw.githubusercontent.com/SR1608/Datasets/main/covid-
data.csv',sep = ',')
```

Q2. High Level Data Understanding:

In [10]:

```
#a.Find no. of rows & columns in the dataset

num_rows = data.shape[0]
num_columns = data.shape[1]

print("Number of rows:", num_rows)
print("Number of columns:", num_columns)
```

```
Number of rows: 57394
Number of columns: 49
```

In [8]:

```
#b. Data types of columns.

data.dtypes
```

Out[8]:

```
iso_code                              object
continent                             object
location                              object
date                                  object
total_cases                          float64
new_cases                            float64
new_cases_smoothed                   float64
total_deaths                         float64
new_deaths                           float64
new_deaths_smoothed                  float64
total_cases_per_million              float64
new_cases_per_million                float64
new_cases_smoothed_per_million       float64
total_deaths_per_million             float64
new_deaths_per_million               float64
new_deaths_smoothed_per_million      float64
reproduction_rate                    float64
icu_patients                         float64
icu_patients_per_million             float64
hosp_patients                        float64
hosp_patients_per_million            float64
weekly_icu_admissions                float64
weekly_icu_admissions_per_million    float64
weekly_hosp_admissions               float64
weekly_hosp_admissions_per_million   float64
total_tests                          float64
new_tests                            float64
total_tests_per_thousand             float64
new_tests_per_thousand               float64
new_tests_smoothed                   float64
new_tests_smoothed_per_thousand      float64
tests_per_case                       float64
positive_rate                        float64
stringency_index                     float64
population                           float64
```

```
population_density                      float64
median_age                              float64
aged_65_older                           float64
aged_70_older                           float64
gdp_per_capita                          float64
extreme_poverty                         float64
cardiovasc_death_rate                   float64
diabetes_prevalence                     float64
female_smokers                          float64
male_smokers                            float64
handwashing_facilities                  float64
hospital_beds_per_thousand              float64
life_expectancy                         float64
human_development_index                 float64
dtype: object
```

In [9]:

```
# c. Info & describe of data in dataframe.
data.info()
data.describe()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57394 entries, 0 to 57393
Data columns (total 49 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   iso_code                              57071 non-null  object
 1   continent                             56748 non-null  object
 2   location                              57394 non-null  object
 3   date                                  57394 non-null  object
 4   total_cases                           53758 non-null  float64
 5   new_cases                             56465 non-null  float64
 6   new_cases_smoothed                    55652 non-null  float64
 7   total_deaths                          44368 non-null  float64
 8   new_deaths                            56465 non-null  float64
 9   new_deaths_smoothed                   55652 non-null  float64
 10  total_cases_per_million               53471 non-null  float64
 11  new_cases_per_million                 56401 non-null  float64
 12  new_cases_smoothed_per_million        55587 non-null  float64
 13  total_deaths_per_million              44096 non-null  float64
 14  new_deaths_per_million                56401 non-null  float64
 15  new_deaths_smoothed_per_million       55587 non-null  float64
 16  reproduction_rate                     37696 non-null  float64
 17  icu_patients                          4490 non-null   float64
 18  icu_patients_per_million              4490 non-null   float64
 19  hosp_patients                         5005 non-null   float64
 20  hosp_patients_per_million             5005 non-null   float64
 21  weekly_icu_admissions                 357 non-null    float64
 22  weekly_icu_admissions_per_million     357 non-null    float64
 23  weekly_hosp_admissions                645 non-null    float64
 24  weekly_hosp_admissions_per_million    645 non-null    float64
 25  total_tests                           22017 non-null  float64
 26  new_tests                             21787 non-null  float64
 27  total_tests_per_thousand              22017 non-null  float64
 28  new_tests_per_thousand                21787 non-null  float64
 29  new_tests_smoothed                    24612 non-null  float64
 30  new_tests_smoothed_per_thousand       24612 non-null  float64
 31  tests_per_case                        22802 non-null  float64
```

```
 32  positive_rate                        23211 non-null  float64
 33  stringency_index                     47847 non-null  float64
 34  population                           57071 non-null  float64
 35  population_density                   54371 non-null  float64
 36  median_age                           51034 non-null  float64
 37  aged_65_older                        50265 non-null  float64
 38  aged_70_older                        50768 non-null  float64
 39  gdp_per_capita                       50367 non-null  float64
 40  extreme_poverty                      33571 non-null  float64
 41  cardiovasc_death_rate                51013 non-null  float64
 42  diabetes_prevalence                  52881 non-null  float64
 43  female_smokers                       39669 non-null  float64
 44  male_smokers                         39156 non-null  float64
 45  handwashing_facilities               24176 non-null  float64
 46  hospital_beds_per_thousand           45936 non-null  float64
 47  life_expectancy                      56336 non-null  float64
 48  human_development_index              49247 non-null  float64
dtypes: float64(45), object(4)
memory usage: 21.5+ MB
```

Out[9]:

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_million	total_deaths_per_million	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
count	5.375800e+04	56465.000000	55652.000000	4.436800e+04	56465.000000	55652.000000	56401.000000	55587.000000	44096.000000		...	50367.000000	33571.000000	51013.000000	52881.000000	39669.000000	39156.000000	24176.000000	45936.000000	56336.000000	49247.000000

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_million	total_deaths_per_million	. . .	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
	04	00	000	04	00	000	000	00				000	000	000	000	000	000	000		000	
mean	1.677974e+05	1953.576941	1920.431953	6.858639e+03	47.054317	46.835439	3139.099882	39.344804	38.403211	95.335293	. . .	20620.172071	12.435453	252.646642	8.070269	10.741569	32.642686	52.089636	3.089724	73.937780	0.722223
std	1.693038e+	18269.650	17777.391	5.578081e+	3900.8537	3783.2727	6183.4552	133.98515	101.64944	180.7216	. . .	20310.9998	19.427924	117.52234	4.189605	10.470743	13.453566	31.645306	2.513193	7.397016	0.153261

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_million	total_deaths_per_million	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
	06	40	85	04								32									
min	1.000000e+00	-826	-552.000000	1.000000e+00	-1918.000000	-232.143000	0.001000	-2212.545000	-269.978000	0.000000	...	661.240000	0.100000	79.370000	0.990000	0.100000	7.700000	1.188000	0.100000	53.280000	0.354000
25%	1.800000e+	0.000000	0.857000	1.300000e+	0.000000	0.000000	90.033000	0.000000	0.260000	3.977750	...	5321.4440	0.500000	156.139000	5.310000	1.900000	21.400000	21.220000	1.300000	69.870000	0.601000

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_million	total_deaths_per_million	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
	02			01								00									
50%	2.070000e+03	14.000000	19.429000	8.400000e+01	0.000000	0.286000	593.663000	2.194000	3.863000	20.383000	...	13913.839000	2.000000	238.339000	7.110000	6.400000	31.400000	52.232000	2.500000	75.345000	0.752000
75%	2.356756e+	235.000000	245.286000	7.270000e+	4.000000	4.000000	3487.050000	25.941000	29.678500	90.571250	...	31400.840	18.100000	318.991000	10.390000	19.600000	40.900000	83.741000	4.200000	79.380000	0.847000

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_million	total_deaths_per_million	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
04			02									00									
max	5.515465e+07	64628.000000	58498.570000	1.328537e+06	10600.000000	9027.714000	765541.772000	86526.580000	2472.188000	124800.014000	...	116935.600000	77.600000	724.417000	30.530000	44.000000	78.100000	98.999000	13.800000	86.750000	0.953000

8 rows × 45 columns

Q3. Low Level Data Understanding :

In [6]:

```
# a. Find count of unique values in location column.
len(data.location.unique())
```

Out[6]:

```
216
```

In [7]:

```
#b. Find which continent has maximum frequency using values counts.
```

```
data['continent'].value_counts().max()
```

Out[7]:

```
14828
```

In [8]:

```
#c. Find maximum & mean value in 'total_cases'.

max_total_cases = data['total_cases'].max()

mean_total_cases = data['total_cases'].mean()

print("Maximum total_cases:", max_total_cases)
print("Mean total_cases:", mean_total_cases)
Maximum total_cases: 55154651.0
Mean total_cases: 167797.3688753302
```

In [10]:

```
#d. Find 25%,50% & 75% quartile value in 'total_deaths'.

quartiles = data['total_deaths'].describe(percentiles=[0.25,0.5,0.75])

print("25th quartile value:", quartiles['25%'])
print("50th quartile value :", quartiles['50%'])
print("75th quartile value:", quartiles['75%'])
25th quartile value: 13.0
50th quartile value : 84.0
75th quartile value: 727.0
```

In [17]:

```
# e. Find which continent has maximum 'human_development_index'.

continent_max_hdi = data.loc[data['human_development_index'].idxmax(),
'continent']

print(continent_max_hdi)
Europe
```

In [19]:

```
# f. Find which continent has minimum 'gdp_per_capita'.
minimum_continents = data.groupby('continent')['gdp_per_capita'].min()
print(minimum_continents)
continent
Africa            661.240
Asia             1479.147
Europe           5189.972
North America    1653.173
Oceania          2205.923
South America    6885.829
Name: gdp_per_capita, dtype: float64
```

Q4. Filter the dataframe with only this columns ['continent','location','date','total_cases','total_deaths','gdp_per_ca pita','human_development_index'] and update the data frame.

In [21]:

```
dataframe = data.loc[:, ['continent', 'location', 'date', 'total_cases',
'total_deaths', 'gdp_per_capita', 'human_development_index']]
dataframe
```

Out[21]:

	continent	location	date	total_cases	total_deaths	gdp_per_capita	human_development_index
0	Asia	Afghanistan	31/12/19	NaN	NaN	1803.987	0.498
1	Asia	Afghanistan	01/01/20	NaN	NaN	1803.987	0.498
2	Asia	Afghanistan	02/01/20	NaN	NaN	1803.987	0.498
3	Asia	Afghanistan	03/01/20	NaN	NaN	1803.987	0.498
4	Asia	Afghanistan	04/01/20	NaN	NaN	1803.987	0.498
...	...	...	...	...	...	...	...
57389	NaN	International	13/11/20	696.0	7.0	NaN	NaN
57390	NaN	International	14/11/20	696.0	7.0	NaN	NaN
57391	NaN	International	15/11/20	696.0	7.0	NaN	NaN
57392	NaN	International	16/11/20	696.0	7.0	NaN	NaN
57393	NaN	International	17/11/20	696.0	7.0	NaN	NaN

57394 rows × 7 columns

Q5. Data Cleaning

In [11]:

```
#a. Remove all duplicates observations

data_no_duplicates = data.drop_duplicates()

print(data_no_duplicates)
```

```
      iso_code continent       location      date  total_cases  new_cases
\
0          AFG      Asia    Afghanistan  31/12/19          NaN        0.0
1          AFG      Asia    Afghanistan  01/01/20          NaN        0.0
2          AFG      Asia    Afghanistan  02/01/20          NaN        0.0
3          AFG      Asia    Afghanistan  03/01/20          NaN        0.0
4          AFG      Asia    Afghanistan  04/01/20          NaN        0.0
...        ...       ...            ...       ...          ...        ...
57389      NaN       NaN  International  13/11/20        696.0        NaN
57390      NaN       NaN  International  14/11/20        696.0        NaN
57391      NaN       NaN  International  15/11/20        696.0        NaN
57392      NaN       NaN  International  16/11/20        696.0        NaN
57393      NaN       NaN  International  17/11/20        696.0        NaN

       new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  .
..  \
0                     NaN           NaN         0.0                  NaN  .
..
1                     NaN           NaN         0.0                  NaN  .
..
2                     NaN           NaN         0.0                  NaN  .
..
3                     NaN           NaN         0.0                  NaN  .
..
4                     NaN           NaN         0.0                  NaN  .
..
...                   ...           ...         ...                  ...  .
..
57389                 NaN           7.0         NaN                  NaN  .
..
57390                 NaN           7.0         NaN                  NaN  .
..
57391                 NaN           7.0         NaN                  NaN  .
..
57392                 NaN           7.0         NaN                  NaN  .
..
57393                 NaN           7.0         NaN                  NaN  .
..

       gdp_per_capita  extreme_poverty  cardiovasc_death_rate  \
0            1803.987              NaN                597.029
1            1803.987              NaN                597.029
2            1803.987              NaN                597.029
3            1803.987              NaN                597.029
4            1803.987              NaN                597.029
...               ...              ...                    ...
57389             NaN              NaN                    NaN
57390             NaN              NaN                    NaN
57391             NaN              NaN                    NaN
```

```
57392             NaN              NaN                    NaN
57393             NaN              NaN                    NaN

       diabetes_prevalence  female_smokers  male_smokers  \
0                     9.59             NaN           NaN
1                     9.59             NaN           NaN
2                     9.59             NaN           NaN
3                     9.59             NaN           NaN
4                     9.59             NaN           NaN
...                    ...             ...           ...
57389                  NaN             NaN           NaN
57390                  NaN             NaN           NaN
57391                  NaN             NaN           NaN
57392                  NaN             NaN           NaN
57393                  NaN             NaN           NaN

       handwashing_facilities  hospital_beds_per_thousand  life_expectancy
\
0                      37.746                         0.5            64.83
1                      37.746                         0.5            64.83
2                      37.746                         0.5            64.83
3                      37.746                         0.5            64.83
4                      37.746                         0.5            64.83
...                       ...                         ...              ...
57389                    NaN                         NaN              NaN
57390                    NaN                         NaN              NaN
57391                    NaN                         NaN              NaN
57392                    NaN                         NaN              NaN
57393                    NaN                         NaN              NaN

       human_development_index
0                        0.498
1                        0.498
2                        0.498
3                        0.498
4                        0.498
...                        ...
57389                    NaN
57390                    NaN
57391                    NaN
57392                    NaN
57393                    NaN

[57394 rows x 49 columns]
```

In [14]:

```
# b. Find missing values in all columns

missing_values = data.isnull().sum()
print(missing_values)
```

```
iso_code                              323
continent                             646
location                                0
date                                    0
total_cases                          3636
new_cases                             929
new_cases_smoothed                   1742
```

```
total_deaths                            13026
new_deaths                                929
new_deaths_smoothed                      1742
total_cases_per_million                  3923
new_cases_per_million                     993
new_cases_smoothed_per_million           1807
total_deaths_per_million                13298
new_deaths_per_million                    993
new_deaths_smoothed_per_million          1807
reproduction_rate                       19698
icu_patients                            52904
icu_patients_per_million                52904
hosp_patients                           52389
hosp_patients_per_million               52389
weekly_icu_admissions                   57037
weekly_icu_admissions_per_million       57037
weekly_hosp_admissions                  56749
weekly_hosp_admissions_per_million      56749
total_tests                             35377
new_tests                               35607
total_tests_per_thousand                35377
new_tests_per_thousand                  35607
new_tests_smoothed                      32782
new_tests_smoothed_per_thousand         32782
tests_per_case                          34592
positive_rate                           34183
stringency_index                         9547
population                                323
population_density                       3023
median_age                               6360
aged_65_older                            7129
aged_70_older                            6626
gdp_per_capita                           7027
extreme_poverty                         23823
cardiovasc_death_rate                    6381
diabetes_prevalence                      4513
female_smokers                          17725
male_smokers                            18238
handwashing_facilities                  33218
hospital_beds_per_thousand              11458
life_expectancy                          1058
human_development_index                  8147
dtype: int64
```

In [15]:

```
#c.Remove all observations where continent column value is missing

data.dropna(subset=['continent'],inplace = True)
data
```

Out[15]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN	597.029	9.59	NaN	NaN	37.746	0.5	64.83	0.498
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN	597.029	9.59	NaN	NaN	37.746	0.5	64.83	0.498
2	AFG	Asia	Afghani	02/01/	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.9	NaN	597.029	9.59	NaN	NaN	37.746	0.5	64.83	0.498

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
			stan	20								87									
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN	597.029	9.59	NaN	NaN	37.746	0.5	64.83	0.498
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN	597.029	9.59	NaN	NaN	37.746	0.5	64.83	0.498

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56743	ZWE	Africa	Zimbabwe	13/11/20	8696.0	29.0	36.000	2555.0	0.0	1.000	...	1899.775	214	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56744	ZWE	Africa	Zimbabwe	14/11/20	8765.0	69.0	42.000	2557.0	2.0	1.000	...	1899.775	214	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56745	ZWE	Africa	Zimbab	15/11/	8786.0	21.0	41.143	2557.0	0.0	0.857	...	1899.775	214	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
			we	20																	
56746	ZWE	Africa	Zimbabwe	16/11/20	8786.0	0.0	36.429	257.0	0.0	0.571	...	1899.775	214.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56747	ZWE	Africa	Zimbabwe	17/11/20	8897.0	111.0	48.000	257.0	0.0	0.429	...	1899.775	214.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535

56748 rows × 49 columns

In [16]:

```
# d. Fill all missing values with 0

data.fillna(0,inplace = True)
data
```

Out[16]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
0	AFG	Asia	Afghanistan	31/12/19	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498
1	AFG	Asia	Afghanistan	01/01/20	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498
2	AFG	Asia	Afghani	02/01/	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.9	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
			stan	20								87									
3	AFG	Asia	Afghanistan	03/01/20	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498
4	AFG	Asia	Afghanistan	04/01/20	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56743	ZWE	Africa	Zimbabwe	13/11/20	8696.0	29.0	36.000	2555.0	0.0	1.000	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56744	ZWE	Africa	Zimbabwe	14/11/20	8765.0	69.0	42.000	2557.0	2.0	1.000	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56745	ZWE	Africa	Zimbab	15/11/	8786.0	21.0	41.143	2557.0	0.0	0.857	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
			we	20																	
56746	ZWE	Africa	Zimbabwe	16/11/20	8786.0	0.0	36.429	257.0	0.0	0.571	...	1899.775	214.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56747	ZWE	Africa	Zimbabwe	17/11/20	8897.0	111.0	48.000	257.0	0.0	0.429	...	1899.775	214.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535

56748 rows × 49 columns

Q6.Date time format :

In [18]:

```
#a.Convert date column in datetime format using pandas.to_datetime

data['date'] = pd.to_datetime(data['date'])
data
```

```
C:\Users\vijay\AppData\Local\Temp\ipykernel_11648\2688995731.py:3: UserWarn
ing: Could not infer format, so each element will be parsed individually, f
```

```
alling back to `dateutil`. To ensure parsing is consistent and as-expected,
please specify a format.
  data['date'] = pd.to_datetime(data['date'])
```

Out[18]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
0	AFG	Asia	Afghanistan	2019-12-31	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498
1	AFG	Asia	Afghanistan	2020-01-01	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
2	AFG	Asia	Afghanistan	2020-02-01	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498
3	AFG	Asia	Afghanistan	2020-03-01	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.987	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498
4	AFG	Asia	Afghani	2020-04	0.0	0.0	0.000	0.0	0.0	0.000	...	1803.9	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
			stan	-01								87									
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56743	ZWE	Africa	Zimbabwe	2020-11-13	8696.0	29.0	36.000	255.0	0.0	1.000	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56744	ZWE	Africa	Zimbabwe	2020-11-	8765.0	69.0	42.000	257.0	2.0	1.000	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
				14																	
56745	ZWE	Africa	Zimbabwe	2020-11-15	8786.0	21.0	41.143	2577.0	0.0	0.857	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
56746	ZWE	Africa	Zimbabwe	2020-11-16	8786.0	0.0	36.429	2577.0	0.0	0.571	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535
567	ZWE	Afri	Zimba	2020-	8897	111	48.000	257	0.0	0.429	...	1899.7	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index
47	ca	bwe	11-17	.0	.0			.0				75									

56748 rows × 49 columns

In [19]:

```
#b.Create new column month after extracting month data from date column.

data['month'] = data['date'].dt.month
data
```

Out[19]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
0	AFG	Asia	Afghanistan	2019	0.0	0.0	0.000	0.0	0.0	0.000	...	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498	12

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
	ia	anistan	-12-31										029						83		
1	AFG	Asia	Afghanistan	2020-01-01	0.0	0.0	0.000	0.0	0.0	0.000	...	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498	1
2	AFG	Asia	Afghanistan	2020-02-01	0.0	0.0	0.000	0.0	0.0	0.000	...	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498	2

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
3	AFG	Asia	Afghanistan	2020-03-01	0.0	0.0	0.000	0.0	0.0	0.000	...	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498	3
4	AFG	Asia	Afghanistan	2020-04-01	0.0	0.0	0.000	0.0	0.0	0.000	...	0.0	597.029	9.59	0.0	0.0	37.746	0.5	64.83	0.498	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
56743	ZWE	Africa	Zimbabwe	2020-11-13	8696.0	29.0	36.000	255.0	0.0	1.000	...	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535	11
56744	ZWE	Africa	Zimbabwe	2020-11-14	8765.0	69.0	42.000	257.0	2.0	1.000	...	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535	11
56745	ZWE	Africa	Zimbabwe	2020-11-	8786.0	21.0	41.143	257.0	0.0	0.857	...	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535	11

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
56746	ZWE	Africa	Zimbabwe	2020-11-15	8786.0	0.0	36.429	257.0	0.0	0.571	...	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535	11
56747	ZWE	Africa	Zimbabwe	2020-11-17	8897.0	111.0	48.000	257.0	0.0	0.429	...	21.4	307.846	1.82	1.6	30.7	36.791	1.7	61.49	0.535	11

56748 rows × 50 columns

Q7. Data Aggregation:

In [20]:

```
#a. Find max value in all columns using groupby function on 'continent' column
```

```
data.groupby('continent').max().reset_index()
```

Out[20]:

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
0	Africa	ZWE	Zimbabwe	2020-12-11	752269.0	13944.0	12583.714	20314.0	572.0	297.429	...	77.6	525.432	22.02	9.7	65.8	89.827	6.30	76.88	0.797	12
1	Asia	YEM	Yemen	2020-12-11	887429.0	9789.4	9319.857	13051.9	200.3	1168.000	...	30.3	724.417	17.72	26.9	78.1	98.999	13.05	84.86	0.933	12
2	Europe	VAT	Vatican	2020-12-12	19912.3	868.52	5486.857	521.47	20.0	1101.000	...	5.7	539.849	10.08	44.0	58.3	97.719	13.80	86.75	0.953	12

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
				-11	.0																
3	NorthAmerica	VIR	UnitedStatesVirginIslands	2020-12-11	1120548 6.0	1848 13.0	15641 9.143	2472 20.0	4928 .0	2715 .143	...	23.5	430. 548	17.11	19.1	53.3	90.650	5.80	83. 92	0.926	12

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	month
4	Oceania	WLF	Wallis and Futuna	2020-12-11	277500.0	1384.0	551.714	907.0	59.0	22.000	...	25.1	561.494	30.53	23.5	48.8	82.502	3.84	83.44	0.939	12
5	South America	VEN	Venezuela	2020-12-11	587646464.0	69074.0	46393.000	16014.0	3935.0	1096.714	...	7.1	373.159	12.54	34.2	42.9	80.635	5.00	81.44	0.843	12

6 rows × 50 columns

In [24]:

```
#b. Store the result in a new dataframe named 'df_groupby'.
# (Use df_groupby dataframe for all further analysis)
```

```
df_groupby = data.groupby('continent').max().reset_index()

print(df_groupby)
       continent iso_code                      location       date  \
0         Africa      ZWE                      Zimbabwe 2020-12-11
1           Asia      YEM                         Yemen 2020-12-11
2         Europe      VAT                       Vatican 2020-12-11
3  North America      VIR  United States Virgin Islands 2020-12-11
4        Oceania      WLF              Wallis and Futuna 2020-12-11
5  South America      VEN                     Venezuela 2020-12-11

   total_cases  new_cases  new_cases_smoothed  total_deaths  new_deaths  \
0     752269.0    13944.0           12583.714       20314.0       572.0
1    8874290.0    97894.0           93198.571      130519.0      2003.0
2    1991233.0    86852.0           54868.571       52147.0      2004.0
3   11205486.0   184813.0          156419.143      247220.0      4928.0
4      27750.0     1384.0             551.714         907.0        59.0
5    5876464.0    69074.0           46393.000      166014.0      3935.0

   new_deaths_smoothed  ...  cardiovasc_death_rate  diabetes_prevalence  \
0              297.429  ...                525.432                22.02
1             1168.000  ...                724.417                17.72
2             1101.000  ...                539.849                10.08
3             2715.143  ...                430.548                17.11
4               22.000  ...                561.494                30.53
5             1096.714  ...                373.159                12.54

   female_smokers  male_smokers  handwashing_facilities  \
0             9.7          65.8                  89.827
1            26.9          78.1                  98.999
2            44.0          58.3                  97.719
3            19.1          53.3                  90.650
4            23.5          48.8                  82.502
5            34.2          42.9                  80.635

   hospital_beds_per_thousand  life_expectancy  human_development_index  \
0                        6.30            76.88                    0.797
1                       13.05            84.86                    0.933
2                       13.80            86.75                    0.953
3                        5.80            83.92                    0.926
4                        3.84            83.44                    0.939
5                        5.00            81.44                    0.843

   month  total_deaths_to_total_cases
0     12                     1.000000
1     12                     1.000000
2     12                     0.196204
3     12                     1.000000
4     12                     0.250000
5     12                     1.000000

[6 rows x 51 columns]
```

Q8.Feature Engineering :

In [23]:

```
#a.. Create a new feature 'total_deaths_to_total_cases' by ratio of
'total_deaths' column to 'total_cases'

df_groupby = data['total_deaths_to_total_cases'] = data['total_deaths'] /
data['total_cases']
df_groupby
```

Out[23]:

```
0             NaN
1             NaN
2             NaN
3             NaN
4             NaN
           ...
56743    0.029324
56744    0.029321
56745    0.029251
56746    0.029251
56747    0.028886
Length: 56748, dtype: float64
```

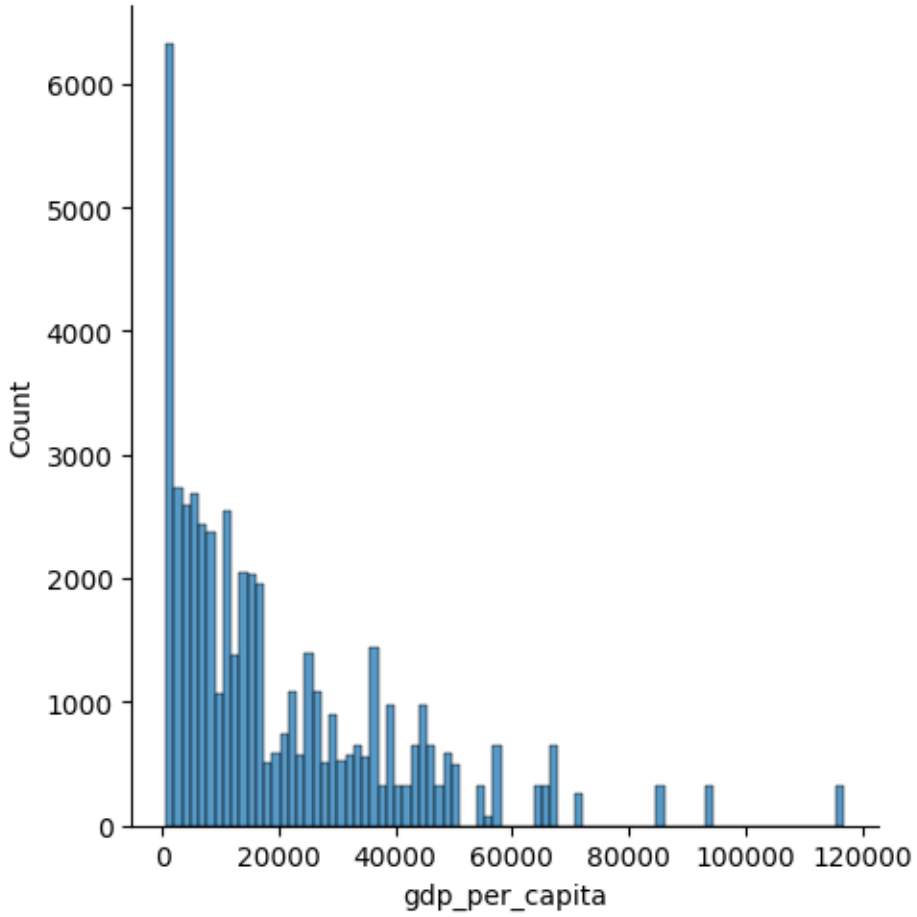
Q9.Data Visualization :

In [7]:

```
#a. Perform Univariate analysis on 'gdp_per_capita' column by plotting
histogram using seaborn dist plot.

df_groupby = sns.displot(data['gdp_per_capita'])
df_groupby
C:\Users\vijay\AppData\Local\Programs\Python\Python311\Lib\site-packages\se
aborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

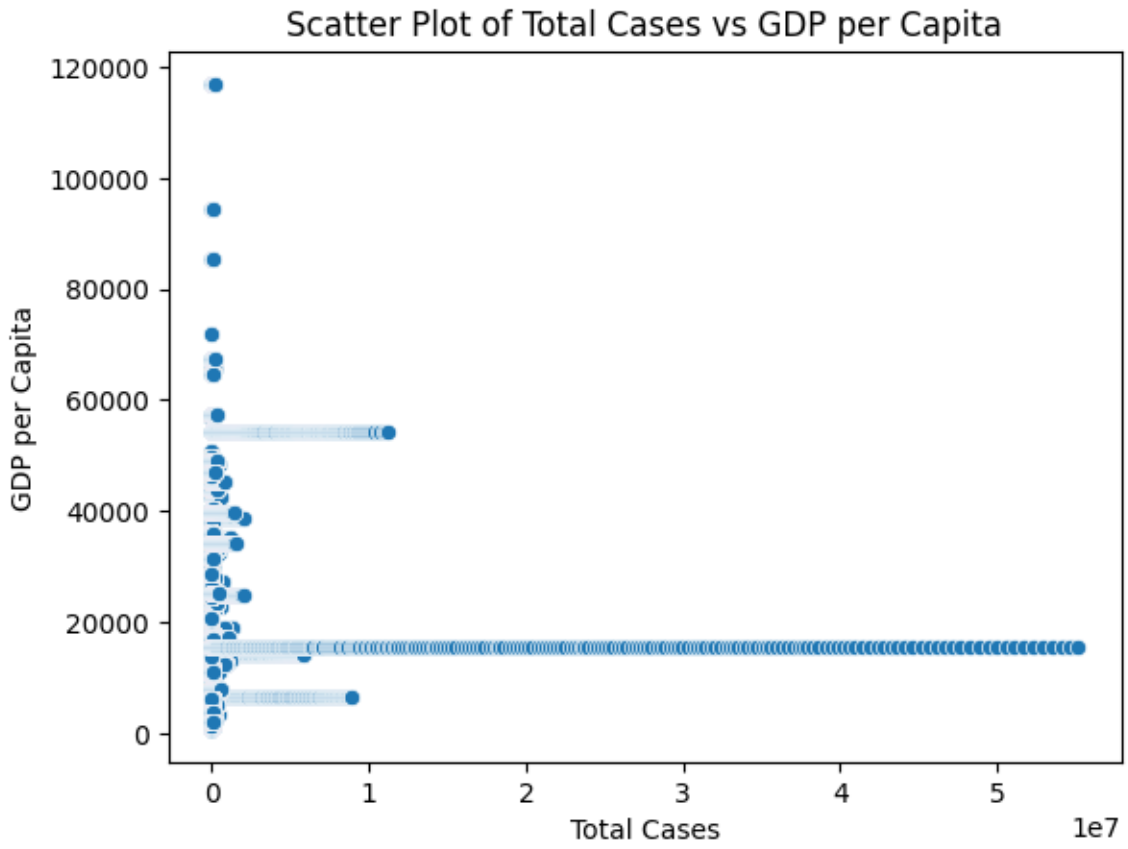
Out[7]:

```
<seaborn.axisgrid.FacetGrid at 0x22980be3d90>
```



In [17]:

```
#b. Plot a scatter plot of 'total_cases' & 'gdp_per_capita'

sns.scatterplot(data=data, x='total_cases', y='gdp_per_capita')

plt.xlabel('Total Cases')
plt.ylabel('GDP per Capita')
plt.title('Scatter Plot of Total Cases vs GDP per Capita')

print(plt.show())
```



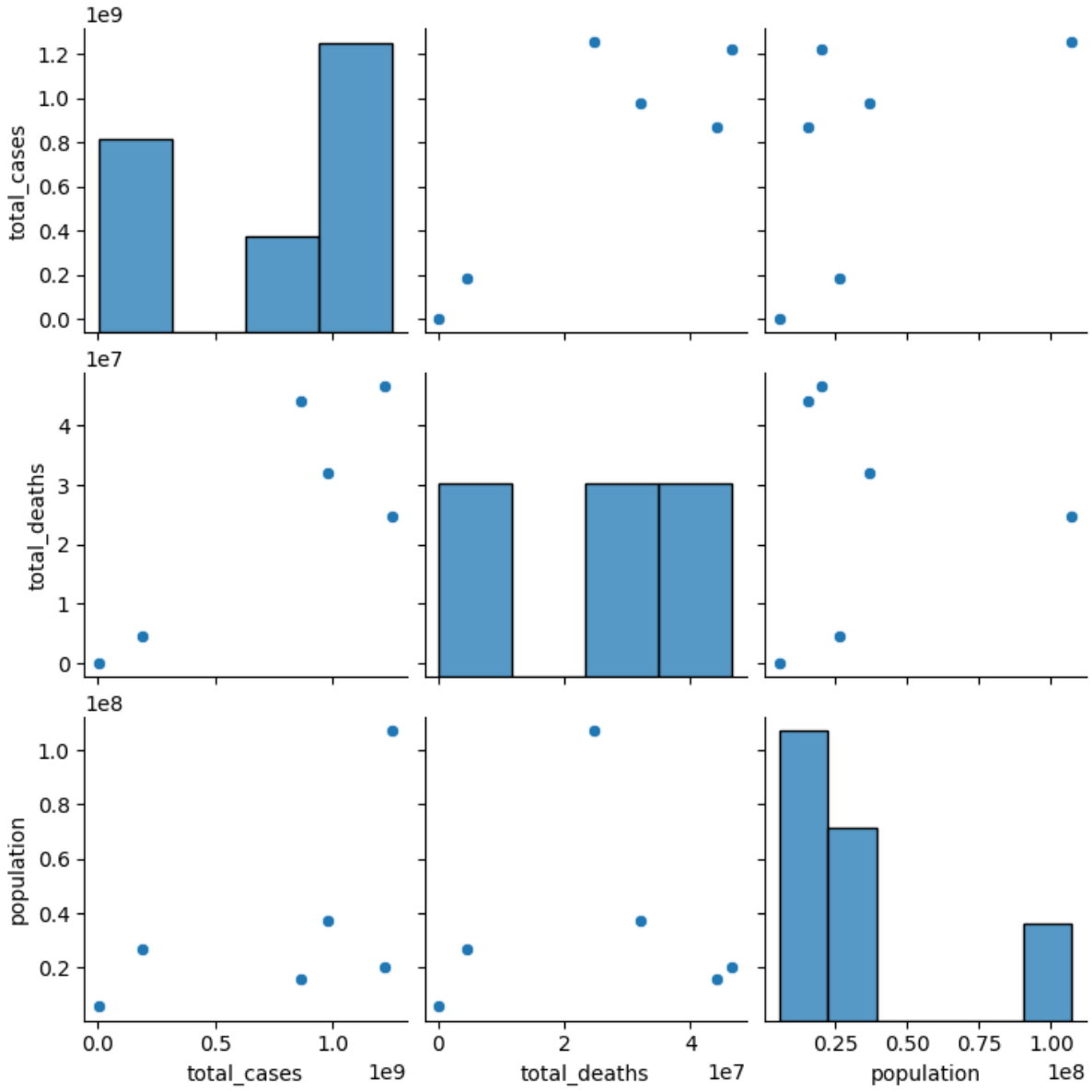

```
None
```

In [19]:

```
#c. Plot Pairplot on df_groupby dataset.

df_groupby = data.groupby(["continent"]).agg({"total_cases":"sum",
"total_deaths":"sum", "population":"mean"}).reset_index()
sns.pairplot(df_groupby)
C:\Users\vijay\AppData\Local\Programs\Python\Python311\Lib\site-packages\se
aborn\axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```

Out[19]:

```
<seaborn.axisgrid.PairGrid at 0x22982a3c250>
```

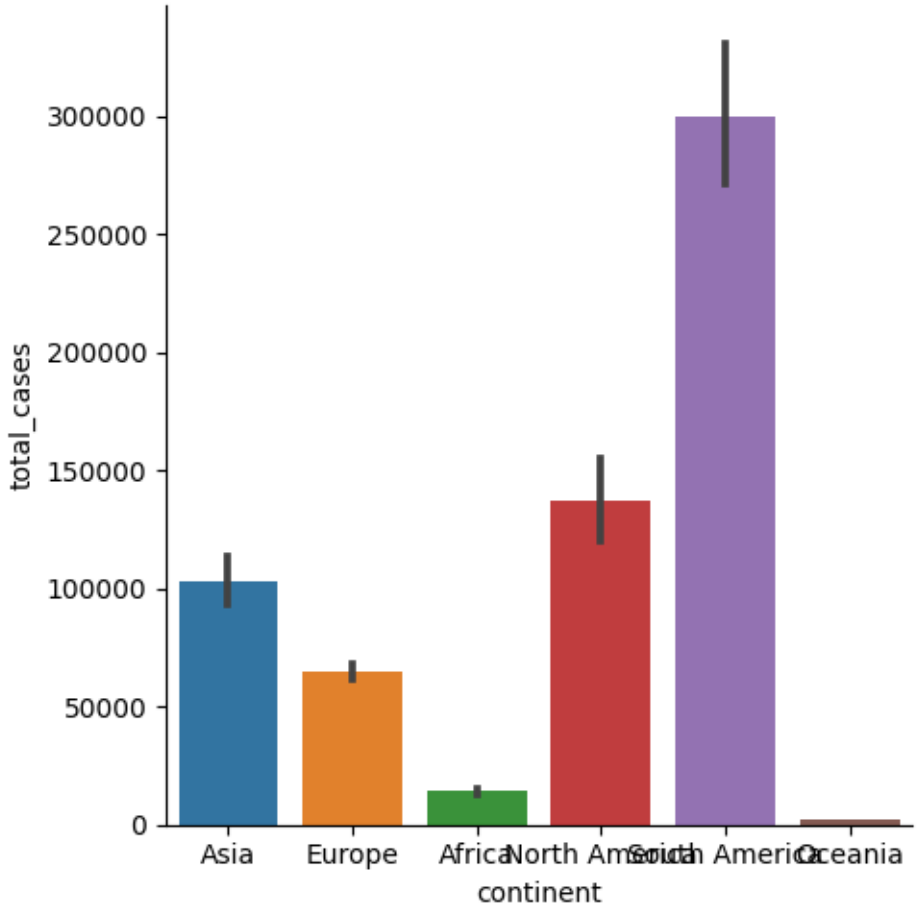


In [21]:

```
#d. Plot a bar plot of 'continent' column with 'total_cases' .
# (Tip : using kind='bar' in seaborn catplot)

df_groupby = sns.catplot(data= data, kind='bar', x='continent',
y='total_cases')
df_groupby
C:\Users\vijay\AppData\Local\Programs\Python\Python311\Lib\site-packages\se
aborn\axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```

Out[21]:

```
<seaborn.axisgrid.FacetGrid at 0x2298253a8d0>
```



Q10.Save the df_groupby dataframe in your local drive using pandas to_csv function .

In [ ]:

```
df_groupby.to_csv('grouped_data.csv')
```

6.Code Snippets:

1.Data Loading and preprocessing

```
import pandas as pd

# Load COVID-19 data from a CSV file
data = pd.read_csv('covid_data.csv')

# Data preprocessing: Drop unnecessary columns, handle missing values
data = data.drop(['unnecessary_column'], axis=1)
data = data.dropna()

# Convert date column to datetime format
data['date'] = pd.to_datetime(data['date'])
```

2.Descriptive Analysis

```
# Calculate total cases, deaths, and recovered
total_cases = data['cases'].sum()
total_deaths = data['deaths'].sum()
total_recovered = data['recovered'].sum()
print("Total Cases:", total_cases)
```

```
print("Total Deaths:", total_deaths)
print("Total Recovered:", total_recovered)
```

3.Demographic Analysis

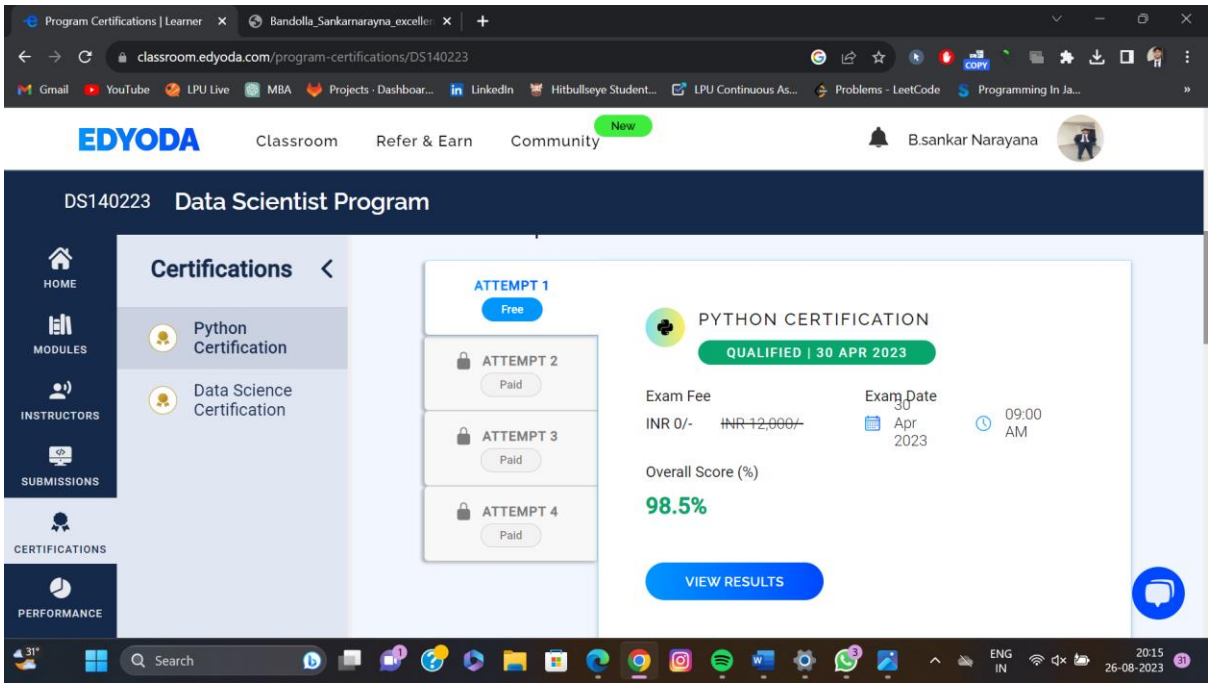
```
import seaborn as sns

# Create a box plot to analyze COVID-19 cases by age group
plt.figure(figsize=(10, 6))
sns.boxplot(x='age_group', y='cases', data=data)
plt.xlabel('Age Group')
plt.ylabel('Cases')
plt.title('COVID-19 Cases by Age Group')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

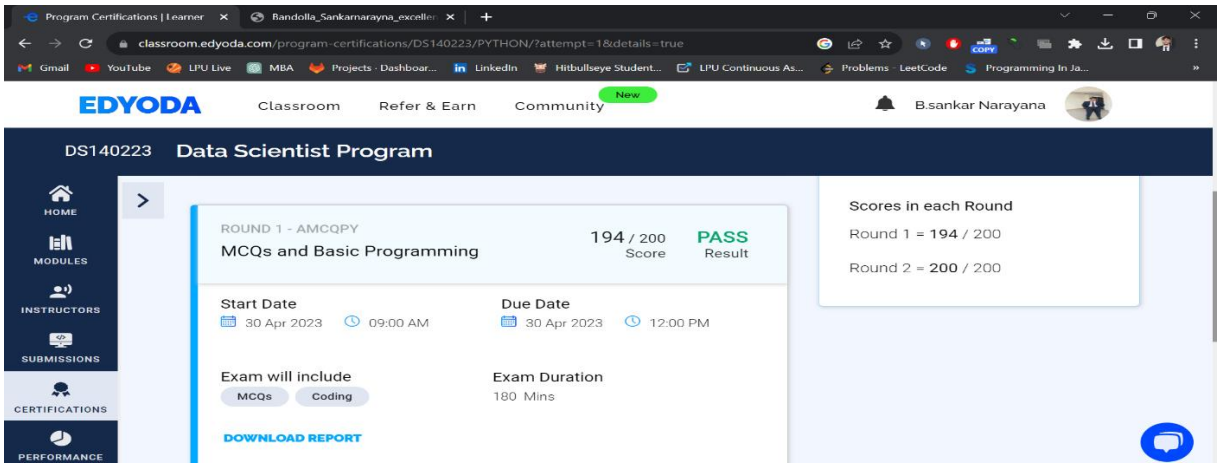
4.Testing and positive rates:

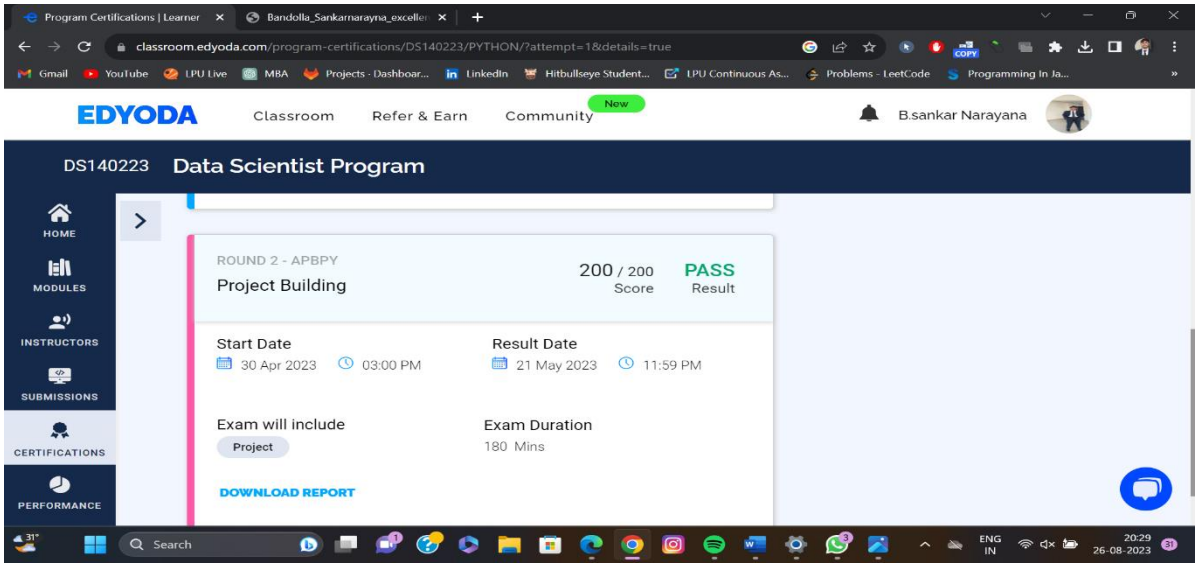
```
# Calculate daily testing and positivity rate
data['testing_rate'] = data['tests'] / data['population']
data['positivity_rate'] = data['cases'] / data['tests']

# Plot trends in testing and positivity rate
plt.figure(figsize=(10, 6))
plt.plot(data['date'], data['testing_rate'], label='Testing Rate')
plt.plot(data['date'], data['positivity_rate'], label='Positivity Rate')
plt.xlabel('Date')
plt.ylabel('Rate')
plt.title('Testing and Positivity Rates')
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

5. Grade sheet of assignments/ marks card from the MOOC

The grade of my assignments/ marks card of MOOC snap shots is given below







6.Bibliography for References: --

eBooks: --

➢Data Science for Beginners by Andrew park

➢Data science for Dummies

➢Data Science from Scratch by Joel Grus

➢Introduction to probability

Web links: --

1. [Kaggle Datasets](https://www.kaggle.com/datasets): Kaggle hosts a variety of datasets that you can use for data science projects and competitions. You can also explore the "Notebooks" section to see how others have analyzed and visualized these datasets.

2. [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/index.php): This repository provides a collection of databases, domain theories, and data generators that are used by the machine learning community for empirical studies.

3. [Awesome Data Science Projects](https://github.com/academic/awesome-datascience#projects): This GitHub repository lists a variety of data science projects across different domains. It's a great resource for inspiration.

4. [DataQuest Projects](https://www.dataquest.io/projects/): DataQuest offers guided projects on various data science topics. These projects provide step-by-step instructions to help you complete real-world tasks.

5. [Towards Data Science](https://towardsdatascience.com/): This Medium publication often features articles with detailed explanations of data science projects. You can find inspiration and learn from the projects showcased here.

6. [GitHub](https://github.com/): Search for "data science projects" or specific topics on GitHub to find repositories where developers share their project code and documentation.

7. [Data Science Central](https://www.datasciencecentral.com/): This community platform occasionally features data science projects and case studies shared by members.