LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

Course Code: MGN342	**Course Title: BUSINESS ANALYTICS**
Course Instructor: Dr. Pritpal Singh	
Academic Task No.: 1	**Academic Task Title: Assignment**
Date of Allotment:	**Date of submission: 20/02/2025**
Student's Roll no: 10	**Student's Reg. no: 12208114**
Evaluation Parameters: (Parameters on which student is to be evaluated- To be mentioned by students as specified at the time of assigning the task by the instructor)	

Learning Outcomes: (Student to write briefly about learnings obtained from the academic tasks)

Declaration:

I declare that this Assignment is my individual work. I have not copied it from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for me by any other person.

Student signature: Sumit Sarkar

Evaluator'scomments (For Instructor's use only)

General Observations	Suggestions for Improvement	Best part of Assignment

Evaluator's Signature and Date:

Marks Obtained: _____ Max. Marks: _____

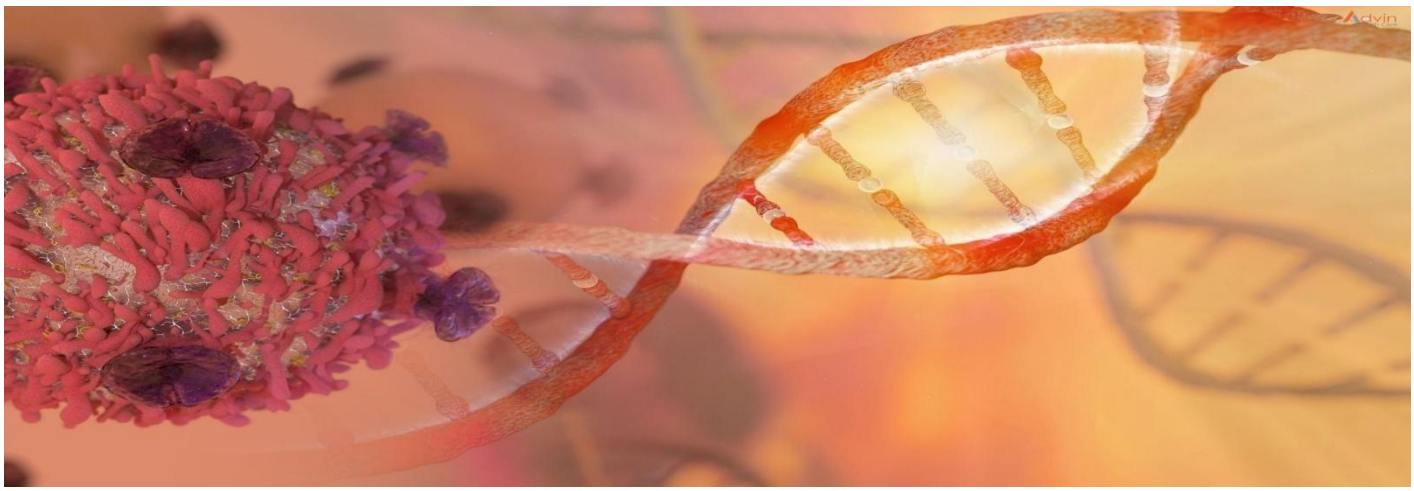
INTRODUCTION

This dataset provides a detailed view of global cancer trends across the 50 most populated countries. With 160,000 records, it encompasses a wide range of variables including cancer types, risk factors, healthcare expenditure, and environmental factors. The data is designed to assist researchers, healthcare policymakers, and data scientists in identifying patterns, predicting future trends, and crafting effective cancer control strategies.

Kaggle- https://www.kaggle.com/datasets/ankushpanday1/cancer-datasettop-50-populated-countries

Key Features of the Dataset

1. **Scope**:
    - Represents global cancer trends, focusing on the 50 most populated countries.
    - Covers a variety of metrics that are useful for analysing cancer prevalence, progression, and control strategies.
2. **Volume**:
    - A large dataset with **160,000 records**, providing a robust sample size for analysis.
3. **Variables Included**:
    - **Cancer Types**: Different types of cancer (e.g., lung, breast, prostate, etc.).
    - **Risk Factors**: Data on factors contributing to cancer risk (e.g., smoking rates, obesity, pollution).
    - **Healthcare Expenditure**: Spending on healthcare per country or per capita.
    - **Environmental Factors**: Variables like pollution levels, UV exposure, or industrial presence.



IMPORTING OF DATA AND IMAGE-

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

```
from PIL import Image
import matplotlib.pyplot as plt
image = Image.open('/content/cancer.jpg')
plt.figure(figsize=(20, 8))
plt.imshow(image)
plt.axis('off')
plt.show()
```

Libraries Imported:

- **pandas and numpy for data manipulation and analysis.**
- **matplotlib and seaborn for visualizations.**
- **PIL for image handling.**

Image Display:

- **A visual introduction to the analysis by loading and displaying an image (likely related to cancer awareness or trends).**

Data Loading

- **The dataset (global_cancer_predictions[1].csv) is loaded using pd.read_csv().**
- **The first five rows are displayed using ss.head() to inspect the structure and contents of the dataset.**

```
ss=pd.read_csv('/content/global_cancer_predictions[1].csv')
ss.head()
```

	Country	Age_Group	Cancer_Type	Risk_Factors	Incidence	Mortality	Prevalence	Urban_Population	Health_Expenditure_%GDP	Tobacco_Use_%	...	Air_Quality_Index
0	Turkey	15-24	Lung	Obesity	44	457	955	32.906758	11.834005	12.578421	...	96
1	Canada	0-14	Prostate	Genetic	643	278	150	40.207750	6.412955	25.120870	...	69
2	China	15-24	Breast	Pollution	565	161	1428	62.225708	7.066045	33.662102	...	10
3	India	15-24	Leukemia	Inactivity	509	117	1996	84.119599	12.102488	29.599358	...	179
4	Nigeria	15-24	Prostate	Pollution	288	170	383	37.403640	14.487316	15.348235	...	151

5 rows × 23 columns

```
ss.describe()
```

	Incidence	Mortality	Prevalence	Urban_Population	Health_Expenditure_%GDP	Tobacco_Use_%	Alcohol_Consumption_Liters	Physical_Activity_%
count	13604.000000	13604.000000	13604.000000	13604.000000	13604.000000	13604.000000	13604.000000	13604.000000
mean	501.197883	248.590194	1048.312041	59.840123	9.052365	24.978641	7.978528	50.032686
std	287.656378	144.045660	547.077595	17.336396	3.451148	8.713032	4.068451	17.286236
min	1.000000	0.000000	100.000000	30.005392	3.000290	10.000124	1.001268	20.004040
25%	251.000000	123.000000	575.000000	44.818202	6.098859	17.354440	4.429249	35.050151
50%	502.000000	248.000000	1048.000000	59.978924	9.050518	25.023015	7.989202	50.037754
75%	750.000000	373.000000	1522.000000	74.716272	12.035808	32.644990	11.536078	65.122639
max	999.000000	499.000000	1999.000000	89.997004	14.999171	39.998479	14.998450	79.997859

- **Handle missing data by imputation or removal.**

```
# Check for missing values
print("Missing values per column:")
print(ss.isnull().sum())

# Fill missing values (example: filling missing numerical data with mean)
ss['Incidence'] = ss['Incidence'].fillna(ss['Incidence'].mean()) # fillna is a method replaces all NaN (missing) values in the specified column with a given value.

# Alternatively, drop rows with missing values
ss = ss.dropna() #dropna is a method removes all rows in the DataFrame that have any missing (NaN) values.
print("After handling missing values:")
print(ss.isnull().sum())
```

 Checking for Missing Values:

- Using ss.isnull().sum() to count the missing values (NaN) in each column.
- Prints the count of missing values for all columns.

 Fill Missing Values in "Incidence" Column:

- ss['Incidence'].fillna(ss['Incidence'].mean()) replaces missing values in the "Incidence" column with the column's mean (average value).
- This method ensures the missing values are handled without deleting rows.

 Remove Rows with Missing Values:

- ss.dropna() removes any rows from the dataset that contain missing values.
- Ensures that no NaN values are left in the dataset.

 Recheck Missing Values:

- Prints the count of missing values again to confirm they've been handled properly.

Output-

```
Missing values per column:
Country                         0
Age_Group                       0
Cancer_Type                     0
Risk_Factors                    0
Incidence                       0
Mortality                       0
Prevalence                      0
Urban_Population                0
Health_Expenditure_%GDP         0
Tobacco_Use_%                   0
Alcohol_Consumption_Liters      0
Physical_Activity_%             0
Obesity_%                       0
Air_Quality_Index               0
UV_Radiation                    0
Family_History_%                1
Genetic_Mutation_%              1
Treatment_Coverage_%            1
GDP_per_Capita                  1
Life_Expectancy                 1
Health_Infrastructure_Index     1
Education_Index                 1
Population_Density              1
dtype: int64
...
Health_Infrastructure_Index     0
Education_Index                 0
Population_Density              0
dtype: int64
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

- **Remove duplicate records and inconsistent data formats.**

```python
# Checking for duplicate rows
print("Number of duplicate rows:", ss.duplicated().sum())

# Remove duplicates
ss = ss.drop_duplicates()
print("Data shape after removing duplicates:", ss.shape)
```

```
Number of duplicate rows: 0
Data shape after removing duplicates: (13603, 23)
```

- **Check for Duplicates**: Counts duplicate rows using `ss.duplicated().sum()`.
- **Remove Duplicates**: Deletes duplicate rows with `ss.drop_duplicates()`.
- **Check Data Shape**: Displays the new dataset dimensions after cleanup using `ss.shape`.
- **Purpose**: Ensures data is clean and free of redundant rows for accurate analysis.

- **Convert data types where necessary for analysis.**

```
# Checking data types
print("Data types before conversion:")
print(ss.dtypes)

# Example: Convert year column to datetime if applicable
# Example: Ensure numerical data types are correct
ss['Mortality'] = pd.to_numeric(ss['Mortality'], errors='coerce')

print("Data types after conversion:")
print(ss.dtypes)
```

Output-

```
Data types before conversion:
Country                          object
Age_Group                        object
Cancer_Type                      object
Risk_Factors                     object
Incidence                         int64
Mortality                         int64
Prevalence                        int64
Urban_Population                float64
Health_Expenditure_%GDP         float64
Tobacco_Use_%                   float64
Alcohol_Consumption_Liters      float64
Physical_Activity_%             float64
Obesity_%                       float64
Air_Quality_Index                 int64
UV_Radiation                    float64
Family_History_%                float64
Genetic_Mutation_%              float64
Treatment_Coverage_%            float64
GDP_per_Capita                  float64
Life_Expectancy                 float64
Health_Infrastructure_Index     float64
Education_Index                 float64
Population_Density              float64
dtype: object
...
Health_Infrastructure_Index     float64
Education_Index                 float64
Population_Density              float64
dtype: object
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

#pd.to_numeric():

*A Pandas function that converts data to a numeric type (e.g., int64 or float64).

*If the data is already numeric, it remains unchanged.

*If the data contains non-numeric values (like strings or symbols), they will be handled based on the errors parameter.

#errors='coerce':

Instructs Pandas on how to handle non-numeric values:

*'coerce': Converts invalid values (e.g., strings, symbols) to NaN (Not a Number).

*'raise': Raises an error if invalid values are encountered.

*'ignore': Leaves invalid values unchanged.

#Why 'coerce'?

**Using 'coerce' ensures that non-numeric data won't crash the program. It replaces invalid entries with NaN, which can be handled later (e.g., by filling or dropping missing values).

- **<u>Check for outliers and handle them appropriately.</u>**

```
# Identify and handle outliers using IQR
Q1 = ss['Incidence'].quantile(0.25) #Calculates the 25th percentile (Q1) of the Incidence column.
Q3 = ss['Incidence'].quantile(0.75) #Calculates the 75th percentile (Q1) of the Incidence column.
IQR = Q3 - Q1

# Define bounds
lower_bound = Q1 - 1.5 * IQR #The 1.5 multiplier in the IQR method is an empirical, statistically grounded standard for detecting outliers.
upper_bound = Q3 + 1.5 * IQR

# Remove outliers
data = ss[(ss['Incidence'] >= lower_bound) & (ss['Incidence'] <= upper_bound)]
print("Data shape after removing outliers:", data.shape)
```

```
Data shape after removing outliers: (13603, 23)
```

#IQR (Interquartile Range) measures the spread of the middle 50% of the data.

Formula: IQR = Q3 - Q1, where:

Q1 (25th percentile): The value below which 25% of the data falls.

Q3 (75th percentile): The value below which 75% of the data falls.

- **Summary Statistics**

```
ss=df['Incidence'].mean()
print(ss)
```

```
501.18672351687127
```

```
ss=df['Incidence'].median()
print(ss)
```

```
502.0
```

```
ss=df['Incidence'].mode()
print(ss)
```

```
0    63
Name: Incidence, dtype: int64
```

```
ss=df['Incidence'].std()
print(ss)
```

```
287.664007000459
```

```
ss=df['Incidence'].var()
print(ss)
```

```
82750.58092356012
```

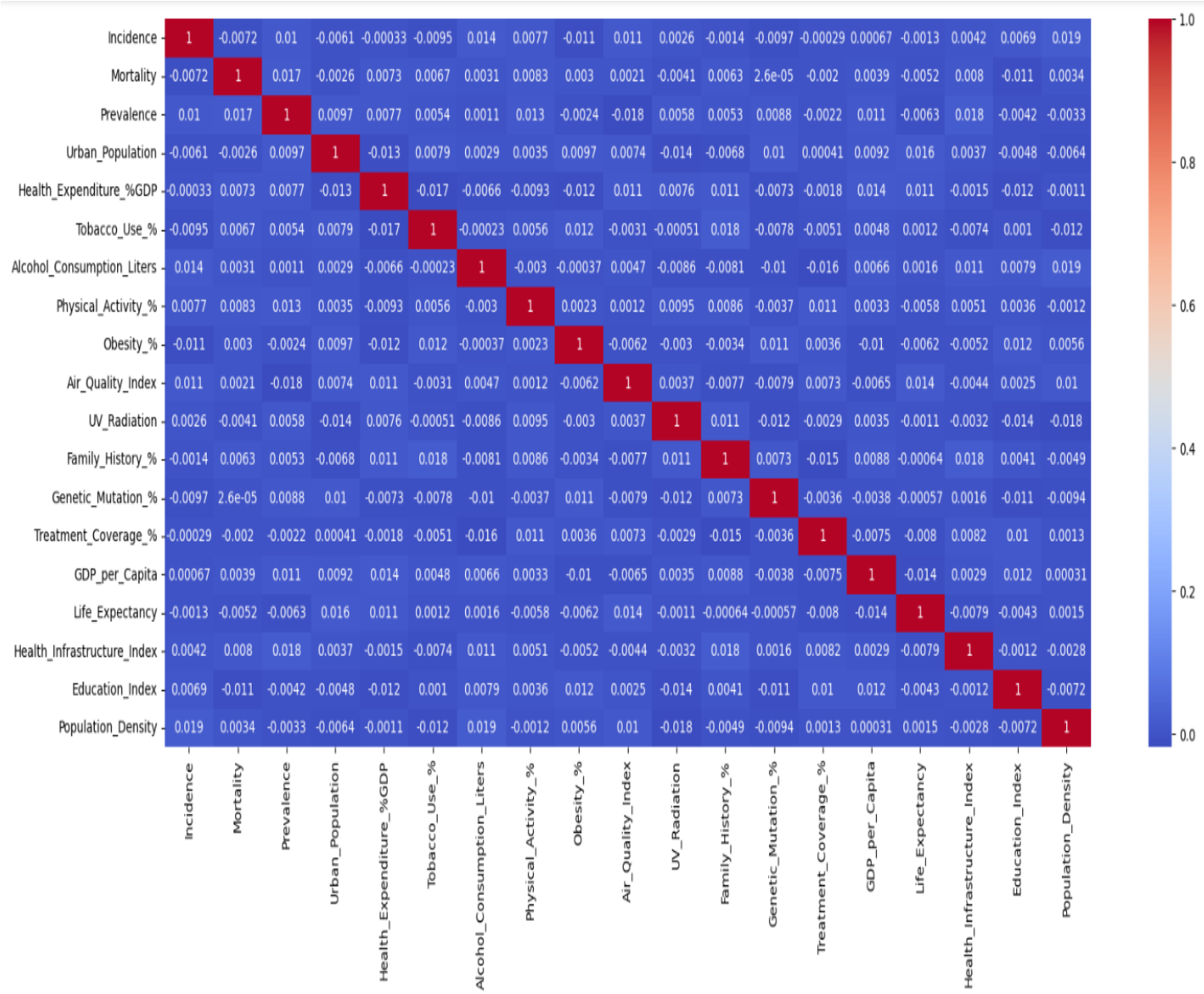
```
#To find the range we need to find the maximum and minimum values
ss=df['Incidence'].max() - df['Incidence'].min()
print(ss)
```

```
998
```

- **Identify relationships between numerical variables using correlation analysis.**

```
# Correlation analysis
numerical_ss = ss.select_dtypes(include=np.number)
plt.figure(figsize=(20, 8))
correlation_matrix = numerical_ss.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.show()
```

Output-



 Selects numerical columns: Filters only numeric data from the dataset using select_dtypes(include=np.number).

 Calculates correlations: Computes the correlation matrix of these columns with .corr().

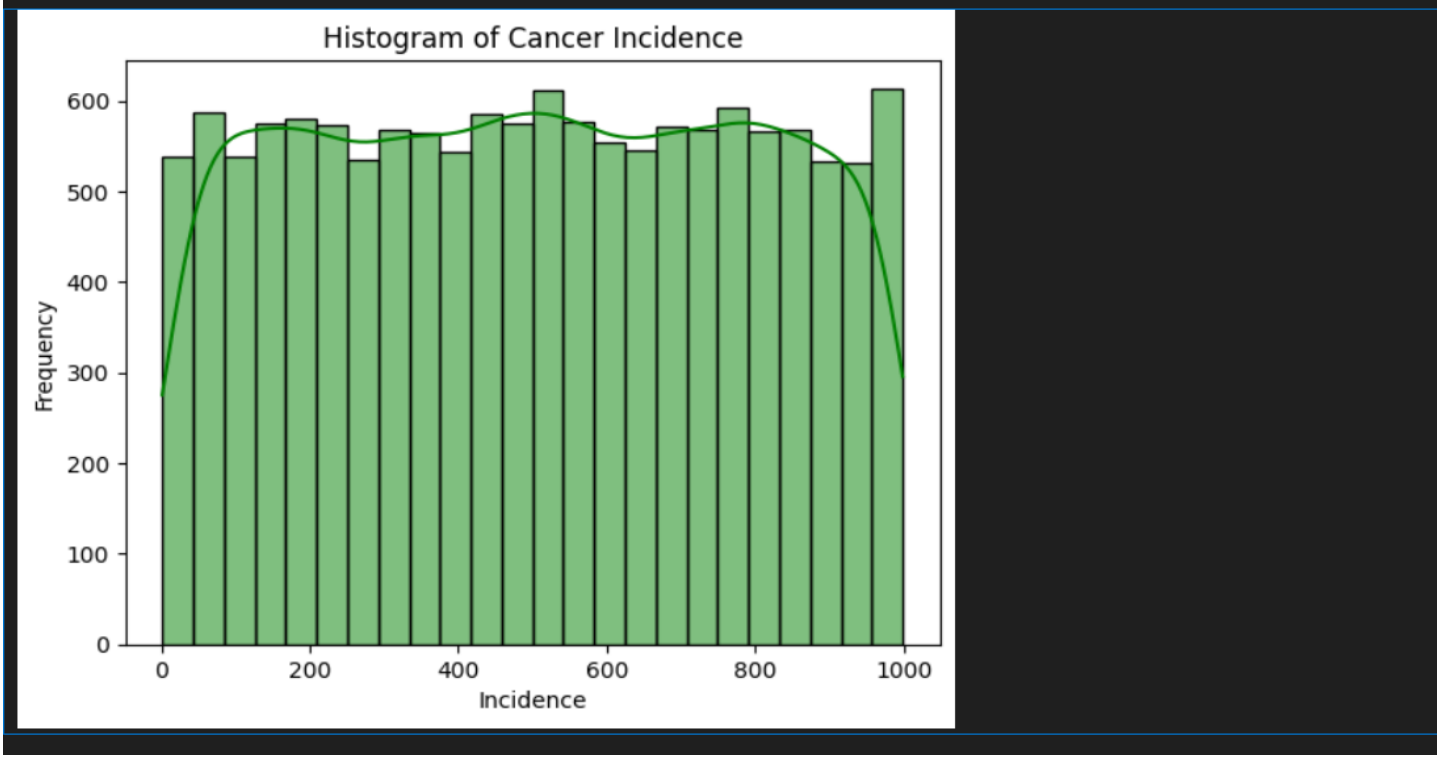
 Plots heatmap: Displays the correlation matrix as a color-coded heatmap using sns.heatmap():

- annot=True shows the correlation values.
- cmap='coolwarm' uses a color scale to visualize correlations.

➢ Data Visualization

I. Histograms for frequency distribution

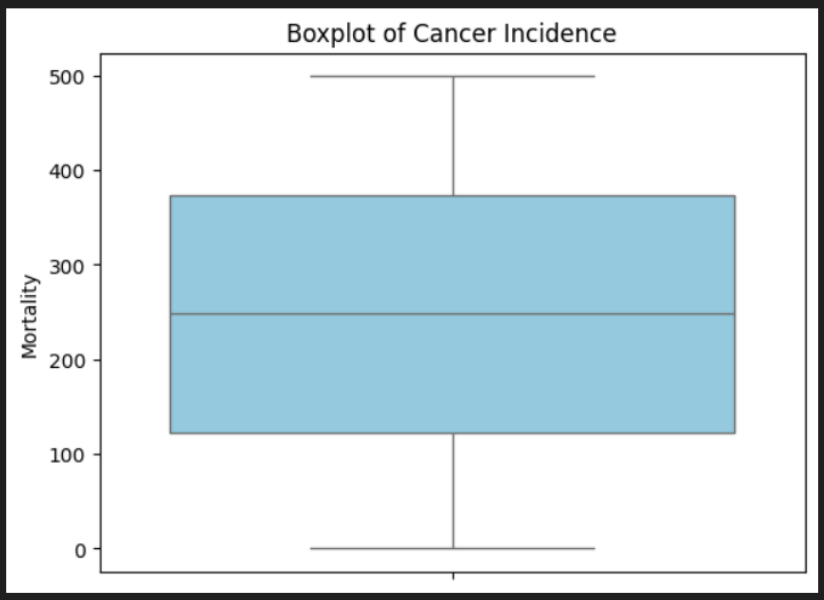
```
# 1. Histogram for frequency distribution
sns.histplot(ss['Incidence'], kde=True, color='green')
plt.title('Histogram of Cancer Incidence')
plt.xlabel('Incidence')
plt.ylabel('Frequency')
plt.show()
```



The histogram represents the frequency distribution of cancer incidence values, with green bars showing how many data points fall within specific ranges. The relatively consistent height of the bars suggests a uniform distribution of the data. Overlaid on the histogram is a green KDE (Kernel Density Estimate) line, which provides a smooth representation of the data's probability density, further confirming the even spread of values. The x-axis displays the range of cancer incidence values (0 to 1000), while the y-axis indicates their frequency. Overall, the chart highlights that the cancer incidence values are uniformly distributed, with no noticeable peaks or clustering.

II. Boxplots to identify outliers

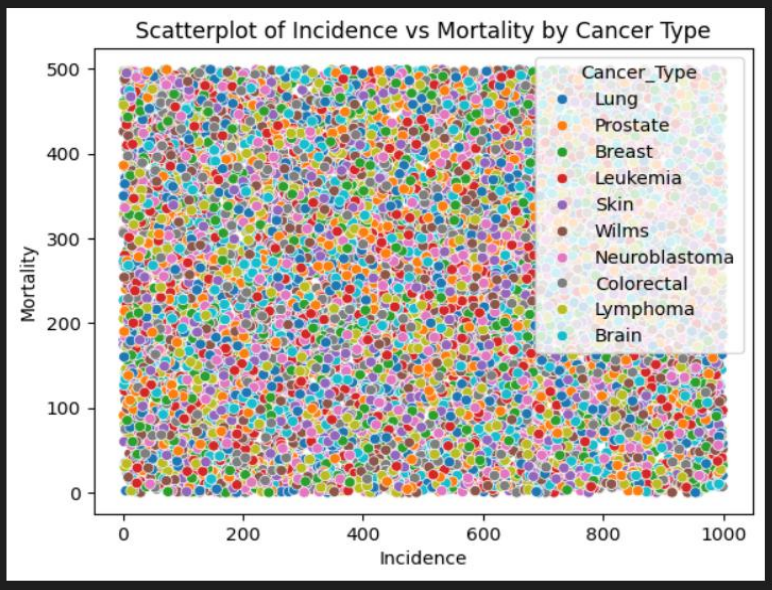
```
# 2. Boxplots to identify outliers
sns.boxplot(ss['Mortality'], color='skyblue')
plt.title('Boxplot of Cancer Incidence')
plt.show()
```



The boxplot visualizes the distribution of cancer mortality rates and helps identify outliers. The blue box represents the interquartile range (IQR), which contains the middle 50% of the data. The line inside the box indicates the median mortality rate. The "whiskers" extend to the minimum and maximum values within 1.5 times the IQR from the box. Data points outside the whiskers would be considered outliers, but none are visible in this plot. The chart shows that the mortality data is symmetric, with most values concentrated within the range depicted by the box and whiskers.

III. Scatter plots for variable relationships

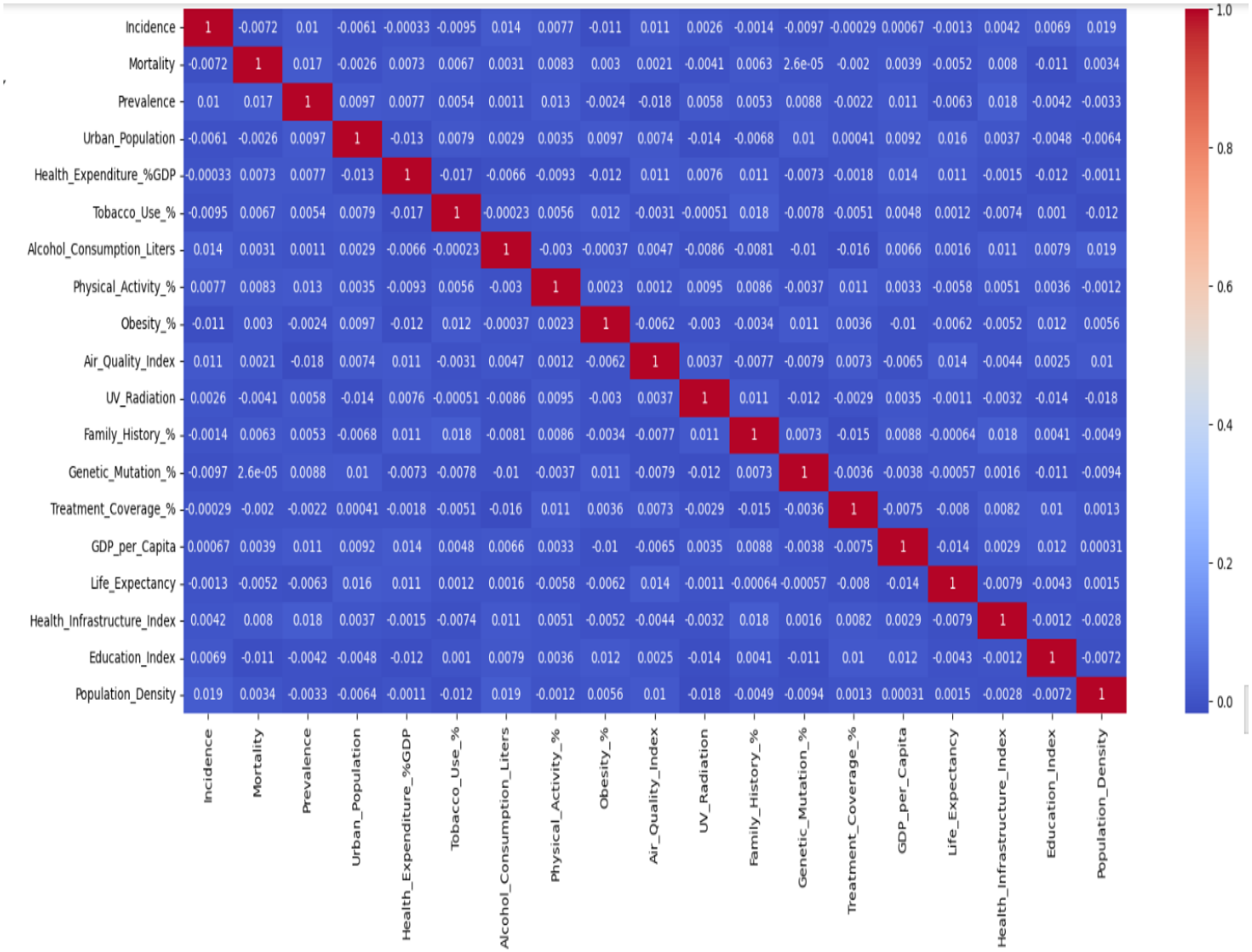
```
# 3. Scatter plots for variable relationships
sns.scatterplot(x='Incidence', y='Mortality', data=data, hue='Cancer_Type')
plt.title('Scatterplot of Incidence vs Mortality by Cancer Type')
plt.show()
```



The scatterplot illustrates the relationship between cancer incidence and mortality, categorized by cancer type. Each point represents a data entry, with its horizontal position (x-axis) indicating incidence rates and its vertical position (y-axis) representing mortality rates. The x-axis represents the number of cancer cases (incidence), ranging from 0 to 1000, while the y-axis represents the number of deaths (mortality), ranging from 0 to 500. The dense clustering of points suggests a large volume of data, with incidence and mortality distributed over the entire range. Different cancer types are differentiated by color, as shown in the legend. This plot helps identify patterns, clusters, or correlations across cancer types. The distribution appears dense, suggesting a wide range of incidence and mortality values for each cancer type without a clear trend visible on this scale.

IV. Correlation Matrix Heatmap

```
# 4. Correlation Matrix Heatmap
numerical_ss = ss.select_dtypes(include=np.number)
plt.figure(figsize=(20, 8))
correlation_matrix = numerical_ss.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
```

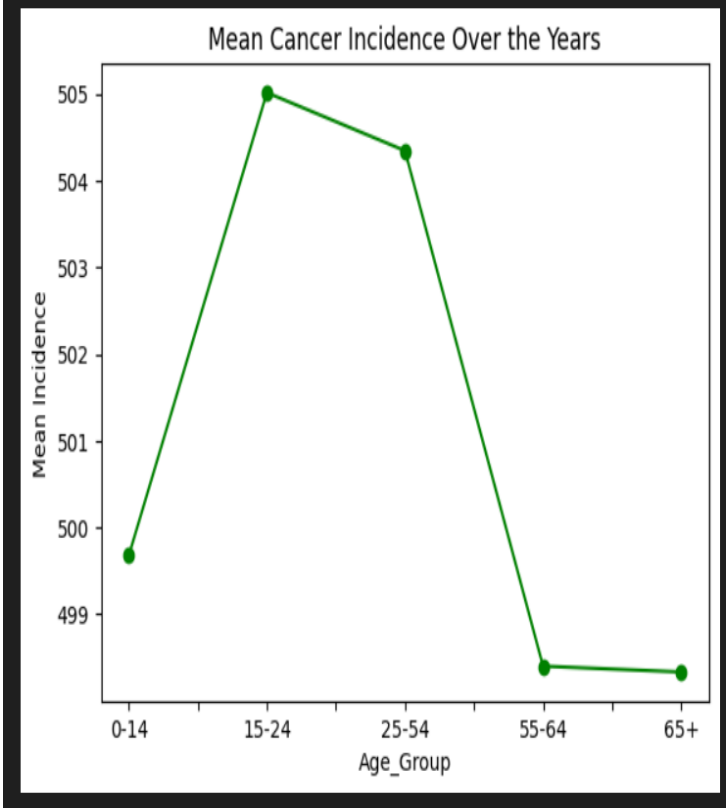


The heatmap shows the correlation between health-related factors and cancer metrics. Red indicates strong positive correlations, blue shows negative or weak correlations, and values along the diagonal are always 1 (perfect self-correlation). Most variables show weak or negligible correlations, but factors like "Health Expenditure" and "Urban Population" have slightly stronger positive correlations with others, hinting at broader influences. This helps identify key relationships for further analysis.

V. Line charts and bar charts for trend analysis

- LINE CHART

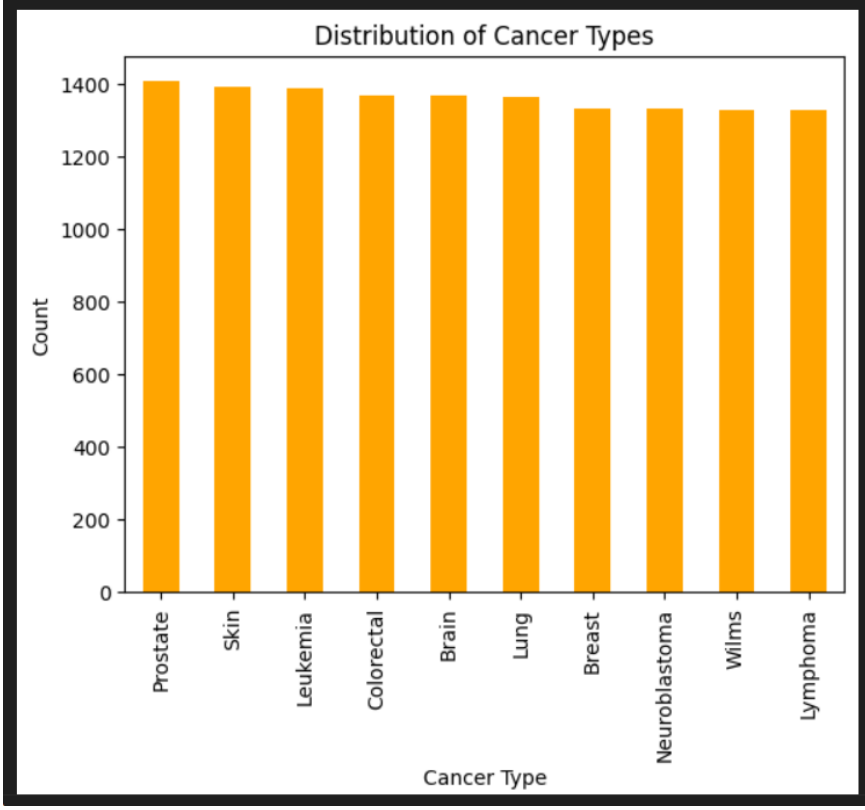
```
# 5. Line charts and bar charts for trend analysis
# Example: Line chart for Incidence over time
ss.groupby('Age_Group')['Incidence'].mean().plot(kind='line', color='green', marker='o')
plt.title('Mean Cancer Incidence Over the Years')
plt.xlabel('Age_Group')
plt.ylabel('Mean Incidence')
plt.show()
```



The line chart illustrates the mean cancer incidence across different age groups. The x-axis represents age groups, while the y-axis indicates the mean incidence. The chart shows a sharp increase in cancer incidence for the 15–24 age group, peaking in the 25–54 age group, followed by a significant decline in the 55–64 and 65+ age groups. This trend highlights varying cancer incidence rates across age demographics.

- **BAR CHART**

```
ss['Cancer_Type'].value_counts().plot(kind='bar', color='orange')
plt.title('Distribution of Cancer Types')
plt.xlabel('Cancer Type')
plt.ylabel('Count')
plt.show()
```



The bar chart displays the distribution of various cancer types. The x-axis represents the different types of cancer, and the y-axis indicates their respective counts. All cancer types appear to have nearly equal counts, suggesting a uniform distribution in the dataset.

SUMMARY:

The analysed covers the global cancer data to identify trends, correlations, and patterns across different variables such as incidence, mortality, healthcare factors, and demographics. A range of visualizations and statistical analyses were used to extract meaningful insights, focusing on cancer prevalence and its association with contributing factors.

KEY FINDINGS:

1. **Cancer Incidence and Age Groups:**
    - The age group 15-24 has the highest average cancer incidence, followed by a sharp decline in older age groups, challenging traditional expectations of cancer trends increasing with age.
2. **Variable Correlation:**
    - The correlation heatmap showed weak relationships between most variables. Key factors like healthcare infrastructure and GDP per capita exhibited limited impact on cancer metrics, indicating the influence of other unmeasured variables.
3. **Cancer Type Distribution:**
    - The distribution of cancer types (e.g., Prostate, Lung, Breast, etc.) appears uniform, suggesting the dataset has a balanced representation across all types.
4. **Mortality vs. Incidence:**
    - There is no strong correlation between cancer incidence and mortality, implying that survival rates and outcomes are likely influenced by treatment quality, access to healthcare, and other region-specific factors.
5. **Healthcare and Lifestyle Indicators:**
    - Factors like physical activity, alcohol consumption, and health expenditure showed negligible correlations with cancer outcomes, hinting at the need for exploring other biological and environmental influences.

CONCLUSION:

The analysis of cancer statistics globally exhibits complicated patterns of cancer trends and the varied factors contributing to the incidence and mortality rates of cancer worldwide. Age, healthcare system, and economics are primary contributors to cancer outcomes, but their relationships are weaker than expected, indicating that other elements are possibly influencing the trends of cancer. The pattern of distribution of various types of cancer is presumed to be the same, but the other variables which are not so well correlated with each other indicate that there is a need for a different approach towards understanding the causative factors of cancer and its consequences. Moreover, the quality and veracity of treatment offered on a regional level greatly affects the survival rates.

SUGGESTIONS:

1. **Focus on Research**:
    - Conduct further research to identify underlying factors influencing cancer prevalence and mortality, such as genetic, environmental, and socio-cultural elements.
2. **Improve Healthcare Accessibility**:
    - Enhance access to healthcare in low-income regions to reduce disparities in cancer outcomes, particularly for early diagnosis and treatment availability.
3. **Targeted Awareness Campaigns**:
    - Create region-specific awareness programs focusing on high-risk groups, such as the younger population, where unexpected spikes in incidence were observed.
4. **Promote Preventive Measures**:
    - Encourage lifestyle changes such as regular physical activity, tobacco and alcohol reduction, and balanced nutrition to mitigate potential risk factors.
5. **Data Enrichment**:
    - Expand data collection to include factors like environmental exposure, diet, and genetic predisposition to improve predictive accuracy and insights.
6. **Policy Implementation**:
    - Governments should invest in healthcare infrastructure, screening programs, and advanced treatment technologies to reduce the cancer burden.