Data Analytics

PYTHON

Reem Hamraz | Mindshare Yuva | 29.06.24

Introduction

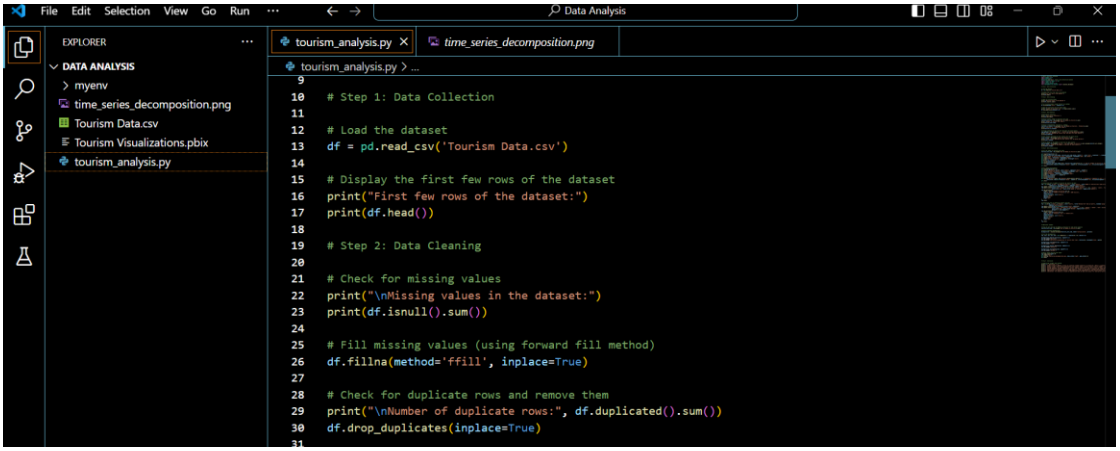
Hello!!

I have created this document to elaborate upon and present my analysis of a hypothetical tourism dataset that spans from 2015 to 2020. This dataset includes information on the number of tourists, revenue, and average spending across various countries. I've used Python for data analysis and visualization, as well as Power BI to enhance the presentation with interactive visuals.

1. DATA ANALYSIS AND VISUALIZATION IN PYTHON

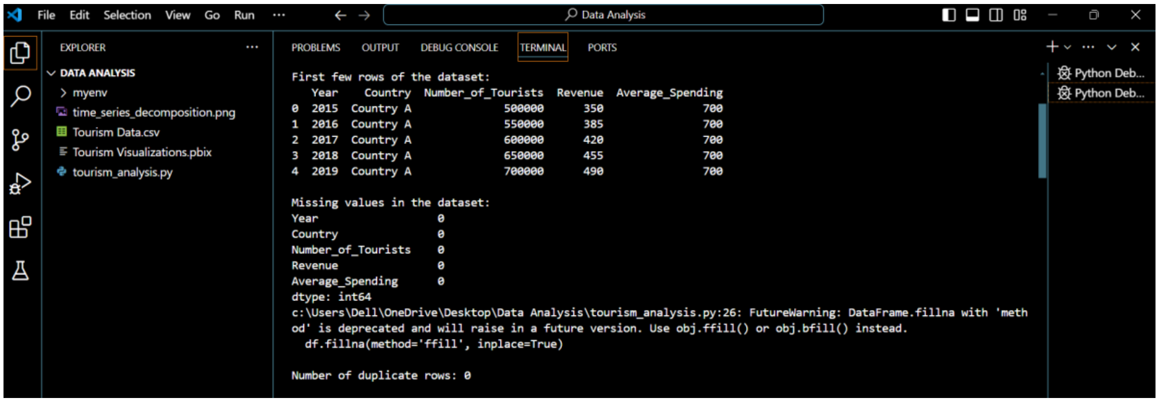
1. **Data Collection and Cleaning:** First, I imported the dataset using the Pandas library and performed initial exploration by displaying the first few rows. This helped me understand the structure and contents of the data. After that, I checked for and handled any missing values and duplicate rows to ensure data quality.

Code-



```python

# Step 1: Data Collection

# Load the dataset
df = pd.read_csv('Tourism Data.csv')

# Display the first few rows of the dataset
print("First few rows of the dataset:")
print(df.head())

# Step 2: Data Cleaning

# Check for missing values
print("\nMissing values in the dataset:")
print(df.isnull().sum())

# Fill missing values (using forward fill method)
df.fillna(method='ffill', inplace=True)

# Check for duplicate rows and remove them
print("\nNumber of duplicate rows:", df.duplicated().sum())
df.drop_duplicates(inplace=True)

```

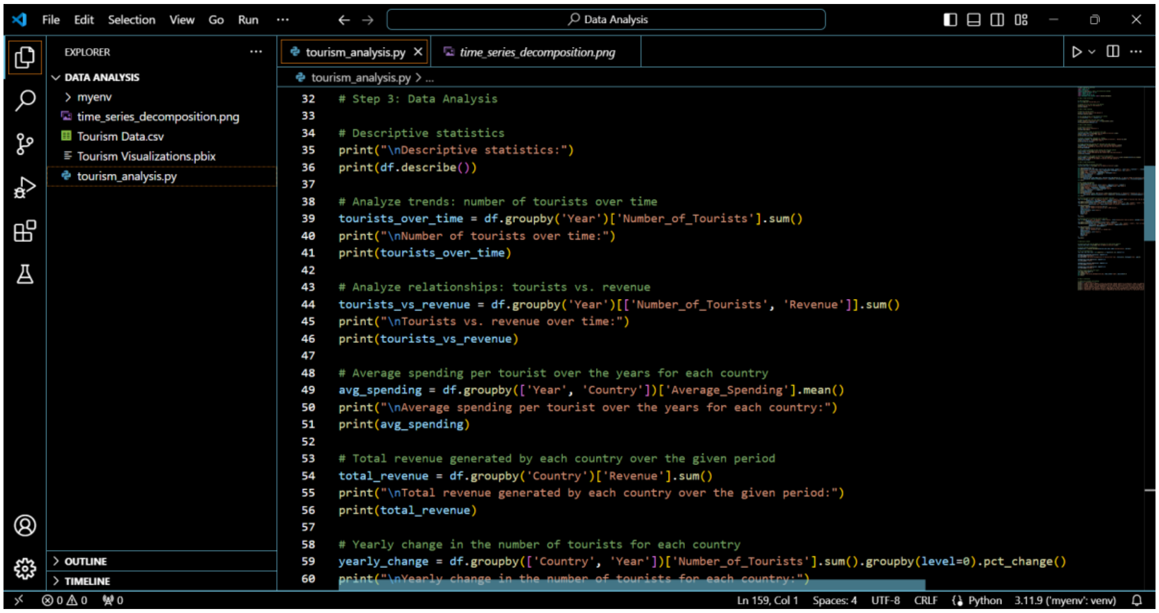
Output-



```
First few rows of the dataset:
   Year    Country  Number_of_Tourists  Revenue  Average_Spending
0  2015  Country A              500000      350               700
1  2016  Country A              550000      385               700
2  2017  Country A              600000      420               700
3  2018  Country A              650000      455               700
4  2019  Country A              700000      490               700

Missing values in the dataset:
Year                  0
Country               0
Number_of_Tourists    0
Revenue               0
Average_Spending      0
dtype: int64
c:\Users\Dell\OneDrive\Desktop\Data Analysis\tourism_analysis.py:26: FutureWarning: DataFrame.fillna with 'meth
od' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  df.fillna(method='ffill', inplace=True)

Number of duplicate rows: 0
```

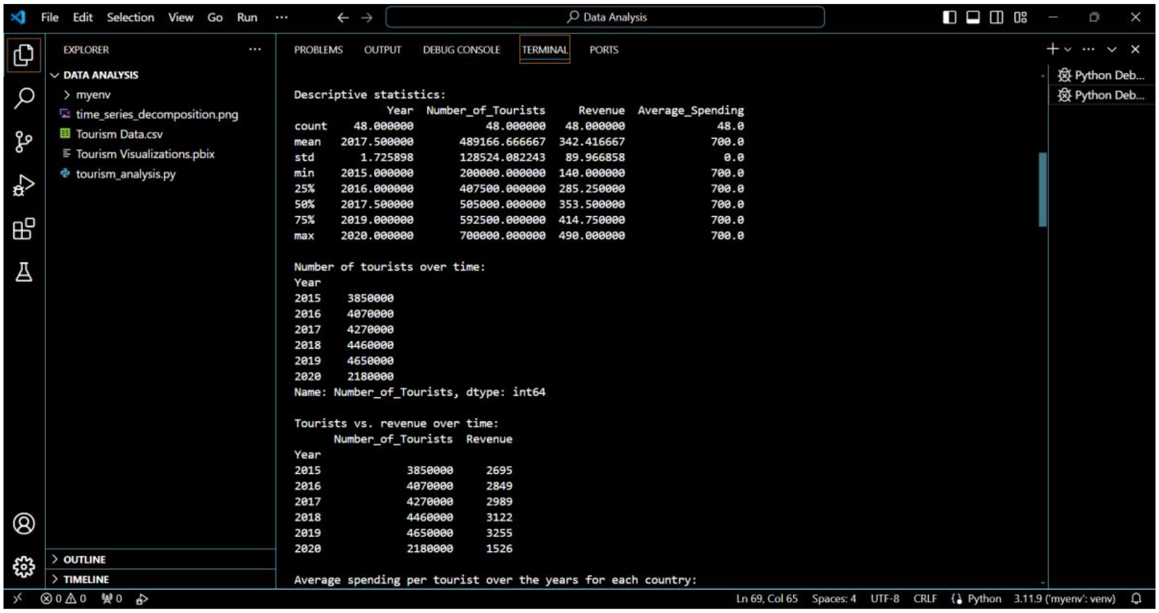
2. **Data Analysis:** I conducted a descriptive statistical analysis to summarize the data. This included calculating the total number of tourists, total revenue, and average spending per tourist over the years for each country. Additionally, I analyzed trends such as the yearly change in the number of tourists for each country and the relationship between the number of tourists and revenue.

Code-



```python
# Step 3: Data Analysis

# Descriptive statistics
print("\nDescriptive statistics:")
print(df.describe())

# Analyze trends: number of tourists over time
tourists_over_time = df.groupby('Year')['Number_of_Tourists'].sum()
print("\nNumber of tourists over time:")
print(tourists_over_time)

# Analyze relationships: tourists vs. revenue
tourists_vs_revenue = df.groupby('Year')[['Number_of_Tourists', 'Revenue']].sum()
print("\nTourists vs. revenue over time:")
print(tourists_vs_revenue)

# Average spending per tourist over the years for each country
avg_spending = df.groupby(['Year', 'Country'])['Average_Spending'].mean()
print("\nAverage spending per tourist over the years for each country:")
print(avg_spending)

# Total revenue generated by each country over the given period
total_revenue = df.groupby('Country')['Revenue'].sum()
print("\nTotal revenue generated by each country over the given period:")
print(total_revenue)

# Yearly change in the number of tourists for each country
yearly_change = df.groupby(['Country', 'Year'])['Number_of_Tourists'].sum().groupby(level=0).pct_change()
print("\nYearly change in the number of tourists for each country:")
```

Output-



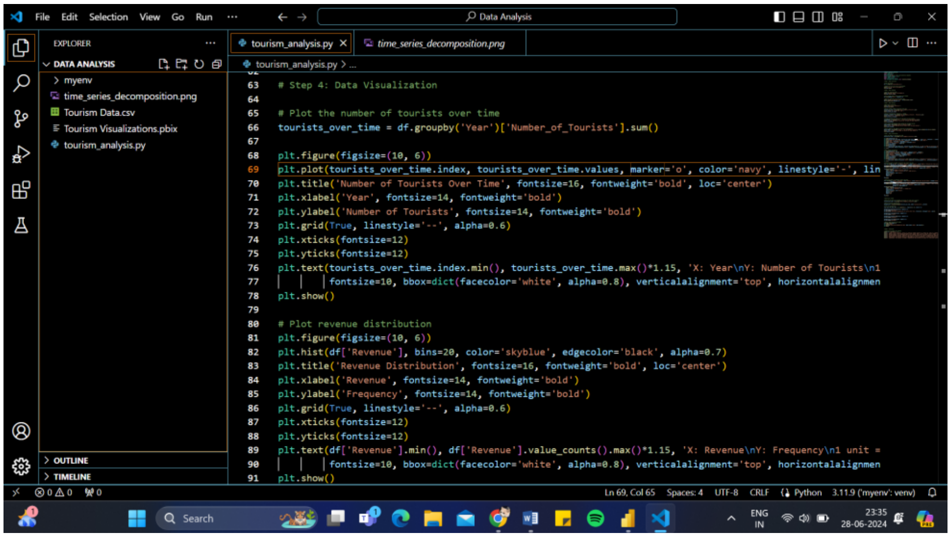
```
Descriptive statistics:
              Year  Number_of_Tourists     Revenue  Average_Spending
count    48.000000           48.000000   48.000000              48.0
mean   2017.500000       489166.666667  342.416667             700.0
std       1.725898       128524.082243   89.966858               0.0
min    2015.000000       200000.000000  140.000000             700.0
25%    2016.000000       407500.000000  285.250000             700.0
50%    2017.500000       505000.000000  353.500000             700.0
75%    2019.000000       592500.000000  414.750000             700.0
max    2020.000000       700000.000000  490.000000             700.0

Number of tourists over time:
Year
2015    3850000
2016    4070000
2017    4270000
2018    4460000
2019    4650000
2020    2180000
Name: Number_of_Tourists, dtype: int64

Tourists vs. revenue over time:
      Number_of_Tourists  Revenue
Year
2015             3850000     2695
2016             4070000     2849
2017             4270000     2989
2018             4460000     3122
2019             4650000     3255
2020             2180000     1526

Average spending per tourist over the years for each country:
```

3. **Data Visualization:** To visualize the data, I created several plots:

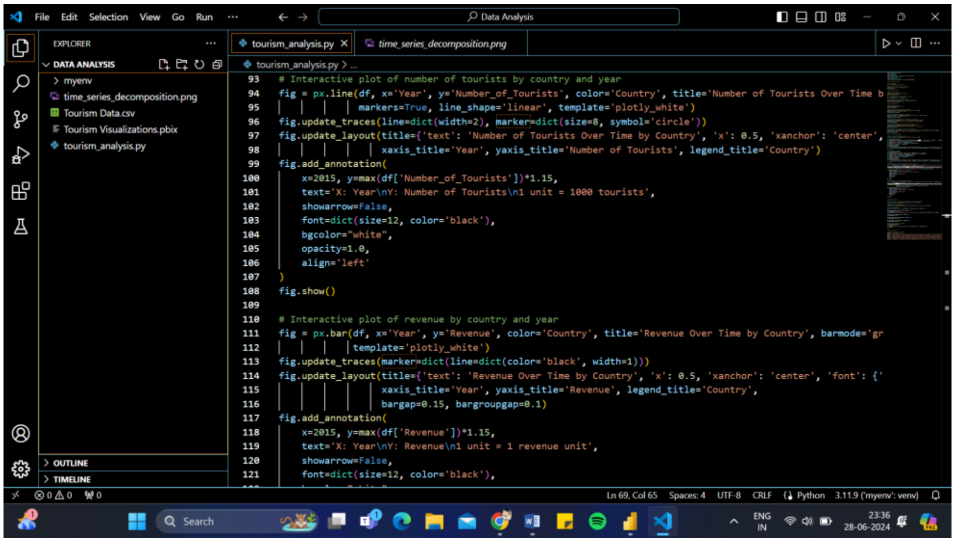
- **Line Chart:** This chart shows the number of tourists over time. It helps to visualize the trend and fluctuations in tourist numbers across the years.
- **Histogram:** The histogram displays the distribution of revenue, highlighting how revenue values are spread across the dataset.

Furthermore, I enhanced these visualizations by adding grid lines, borders, and customizing the aesthetics to make them more appealing and easier to interpret.

Code-

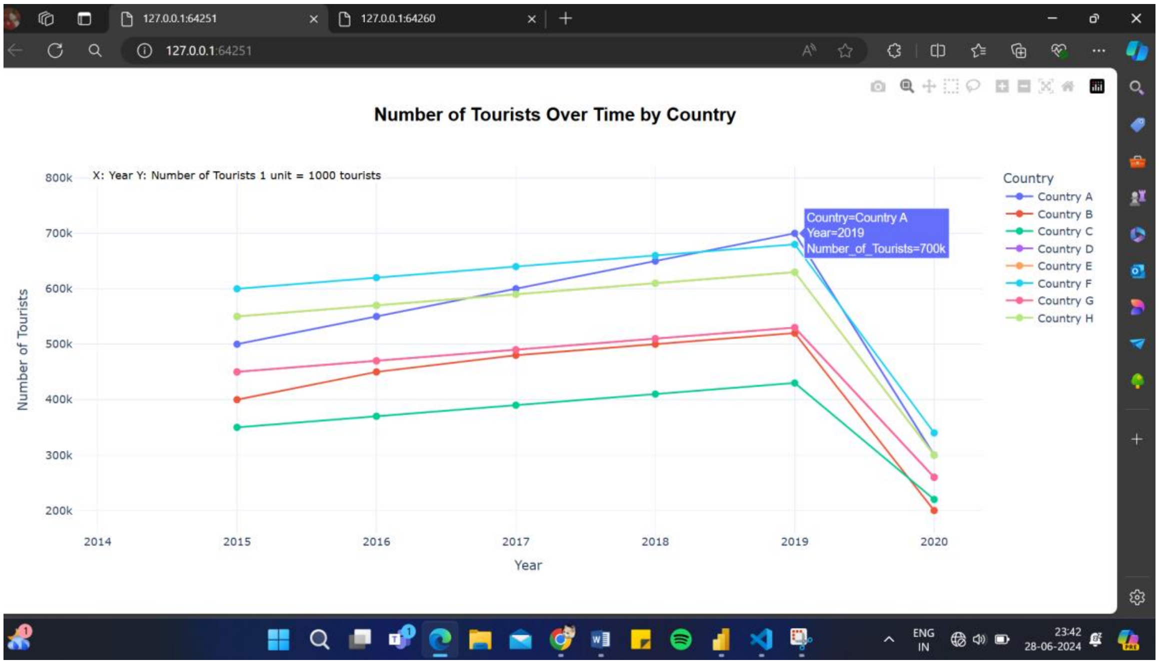

File Edit Selection View Go Run ···
Data Analysis
EXPLORER
DATA ANALYSIS
myenv
time_series_decomposition.png
Tourism Data.csv
Tourism Visualizations.pbix
tourism_analysis.py
tourism_analysis.py
time_series_decomposition.png
tourism_analysis.py > ...
OUTLINE
TIMELINE
Ln 69, Col 65 Spaces: 4 UTF-8 CRLF Python 3.11.9 ('myenv': venv)
Search
ENG IN
23:35
28-06-2024

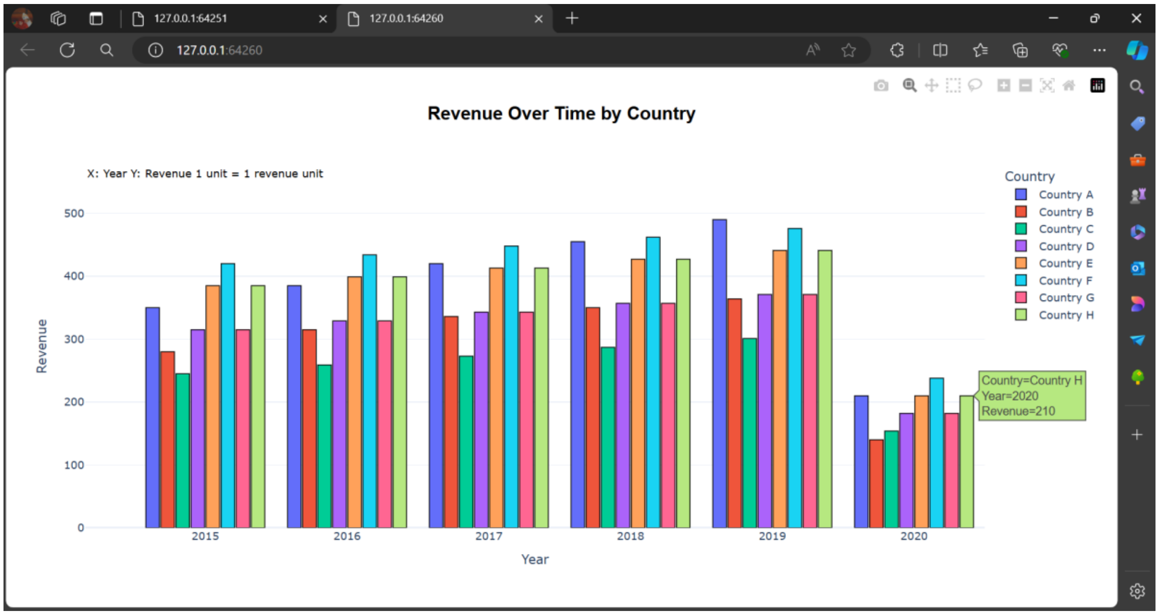
```python
# Step 4: Data Visualization

# Plot the number of tourists over time
tourists_over_time = df.groupby('Year')['Number_of_Tourists'].sum()

plt.figure(figsize=(10, 6))
plt.plot(tourists_over_time.index, tourists_over_time.values, marker='o', color='navy', linestyle='-', lin
plt.title('Number of Tourists Over Time', fontsize=16, fontweight='bold', loc='center')
plt.xlabel('Year', fontsize=14, fontweight='bold')
plt.ylabel('Number of Tourists', fontsize=14, fontweight='bold')
plt.grid(True, linestyle='--', alpha=0.6)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.text(tourists_over_time.index.min(), tourists_over_time.max()*1.15, 'X: Year\nY: Number of Tourists\n1
         fontsize=10, bbox=dict(facecolor='white', alpha=0.8), verticalalignment='top', horizontalalignmen
plt.show()

# Plot revenue distribution
plt.figure(figsize=(10, 6))
plt.hist(df['Revenue'], bins=20, color='skyblue', edgecolor='black', alpha=0.7)
plt.title('Revenue Distribution', fontsize=16, fontweight='bold', loc='center')
plt.xlabel('Revenue', fontsize=14, fontweight='bold')
plt.ylabel('Frequency', fontsize=14, fontweight='bold')
plt.grid(True, linestyle='--', alpha=0.6)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.text(df['Revenue'].min(), df['Revenue'].value_counts().max()*1.15, 'X: Revenue\nY: Frequency\n1 unit =
         fontsize=10, bbox=dict(facecolor='white', alpha=0.8), verticalalignment='top', horizontalalignmen
plt.show()
```


File Edit Selection View Go Run ···
Data Analysis
EXPLORER
DATA ANALYSIS
myenv
time_series_decomposition.png
Tourism Data.csv
Tourism Visualizations.pbix
tourism_analysis.py
tourism_analysis.py
time_series_decomposition.png
tourism_analysis.py > ...
OUTLINE
TIMELINE
Ln 69, Col 65 Spaces: 4 UTF-8 CRLF Python 3.11.9 ('myenv': venv)
Search
ENG IN
23:36
28-06-2024

```python
# Interactive plot of number of tourists by country and year
fig = px.line(df, x='Year', y='Number_of_Tourists', color='Country', title='Number of Tourists Over Time b
              markers=True, line_shape='linear', template='plotly_white')
fig.update_traces(line=dict(width=2), marker=dict(size=8, symbol='circle'))
fig.update_layout(title={'text': 'Number of Tourists Over Time by Country', 'x': 0.5, 'xanchor': 'center',
                  xaxis_title='Year', yaxis_title='Number of Tourists', legend_title='Country')
fig.add_annotation(
    x=2015, y=max(df['Number_of_Tourists'])*1.15,
    text='X: Year\nY: Number of Tourists\n1 unit = 1000 tourists',
    showarrow=False,
    font=dict(size=12, color='black'),
    bgcolor="white",
    opacity=1.0,
    align='left'
)
fig.show()

# Interactive plot of revenue by country and year
fig = px.bar(df, x='Year', y='Revenue', color='Country', title='Revenue Over Time by Country', barmode='gr
             template='plotly_white')
fig.update_traces(marker=dict(line=dict(color='black', width=1)))
fig.update_layout(title={'text': 'Revenue Over Time by Country', 'x': 0.5, 'xanchor': 'center', 'font': {'
                  xaxis_title='Year', yaxis_title='Revenue', legend_title='Country',
                  bargap=0.15, bargroupgap=0.1)
fig.add_annotation(
    x=2015, y=max(df['Revenue'])*1.15,
    text='X: Year\nY: Revenue\n1 unit = 1 revenue unit',
    showarrow=False,
    font=dict(size=12, color='black'),
```

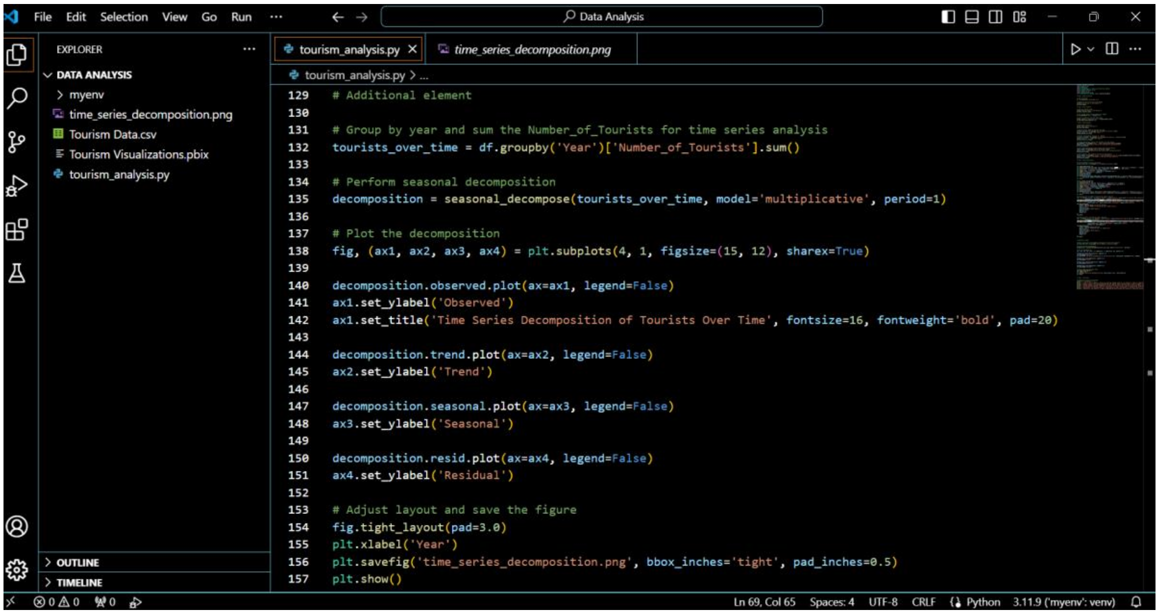
Output-


127.0.0.1:64251
127.0.0.1:64260
127.0.0.1:64251
Number of Tourists Over Time by Country
X: Year Y: Number of Tourists 1 unit = 1000 tourists
Number of Tourists
800k
700k
600k
500k
400k
300k
200k
2014
2015
2016
2017
2018
2019
2020
Year
Country=Country A
Year=2019
Number_of_Tourists=700k
Country
Country A
Country B
Country C
Country D
Country E
Country F
Country G
Country H
ENG
IN
23:42
28-06-2024


127.0.0.1:64251
127.0.0.1:64260
127.0.0.1:64260
Revenue Over Time by Country
X: Year Y: Revenue 1 unit = 1 revenue unit
Revenue
500
400
300
200
100
0
2015
2016
2017
2018
2019
2020
Year
Country=Country H
Year=2020
Revenue=210
Country
Country A
Country B
Country C
Country D
Country E
Country F
Country G
Country H

4. **Time Series Decomposition:** I performed a time series decomposition to break down the number of tourists into three components: trend, seasonality, and residuals. This decomposition helps in understanding the underlying patterns in the data. The trend component shows the long-term movement in the number of tourists, seasonality captures the periodic fluctuations, and residuals represent the random noise.

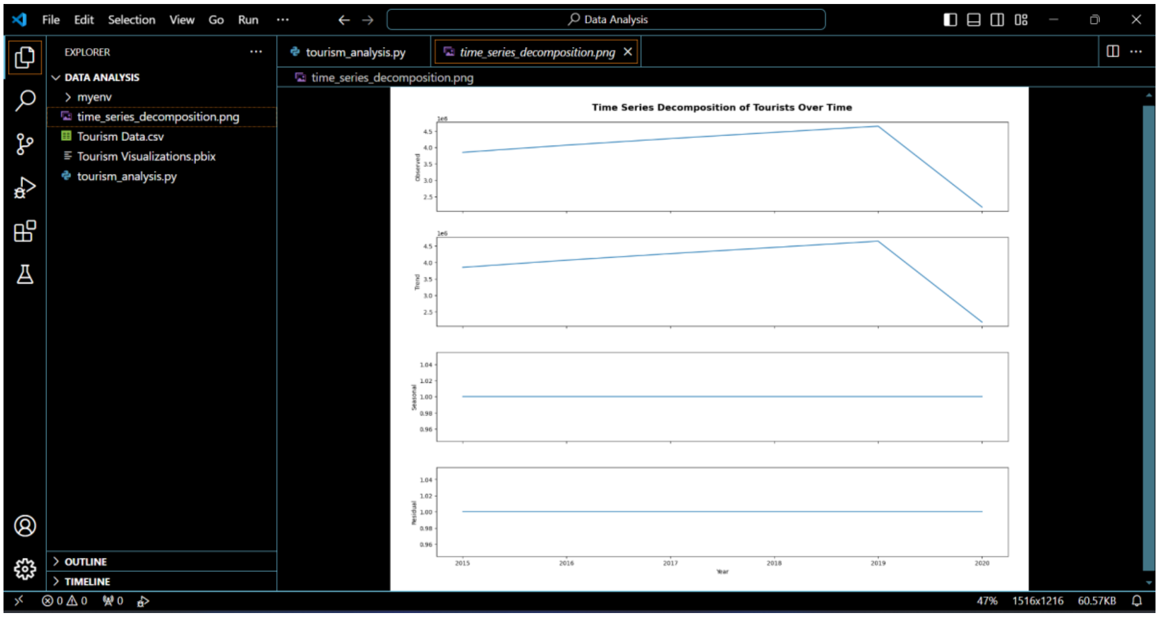
Code-



File Edit Selection View Go Run ···
Data Analysis
EXPLORER
DATA ANALYSIS
myenv
time_series_decomposition.png
Tourism Data.csv
Tourism Visualizations.pbix
tourism_analysis.py
OUTLINE
TIMELINE
tourism_analysis.py
time_series_decomposition.png
tourism_analysis.py > ...
Ln 69, Col 65 Spaces: 4 UTF-8 CRLF Python 3.11.9 ('myenv': venv)

```python
# Additional element

# Group by year and sum the Number_of_Tourists for time series analysis
tourists_over_time = df.groupby('Year')['Number_of_Tourists'].sum()

# Perform seasonal decomposition
decomposition = seasonal_decompose(tourists_over_time, model='multiplicative', period=1)

# Plot the decomposition
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(15, 12), sharex=True)

decomposition.observed.plot(ax=ax1, legend=False)
ax1.set_ylabel('Observed')
ax1.set_title('Time Series Decomposition of Tourists Over Time', fontsize=16, fontweight='bold', pad=20)

decomposition.trend.plot(ax=ax2, legend=False)
ax2.set_ylabel('Trend')

decomposition.seasonal.plot(ax=ax3, legend=False)
ax3.set_ylabel('Seasonal')

decomposition.resid.plot(ax=ax4, legend=False)
ax4.set_ylabel('Residual')

# Adjust layout and save the figure
fig.tight_layout(pad=3.0)
plt.xlabel('Year')
plt.savefig('time_series_decomposition.png', bbox_inches='tight', pad_inches=0.5)
plt.show()
```

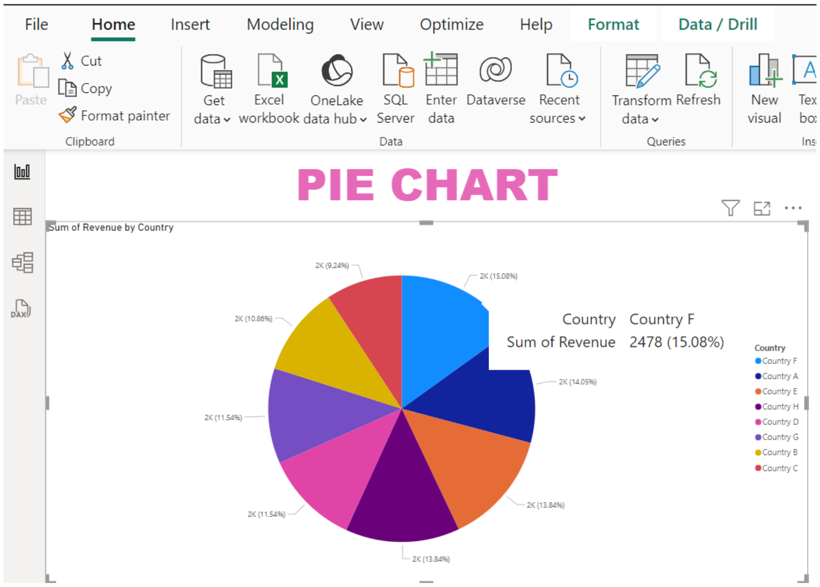
Output-



2. POWER BI VISUALIZATIONS

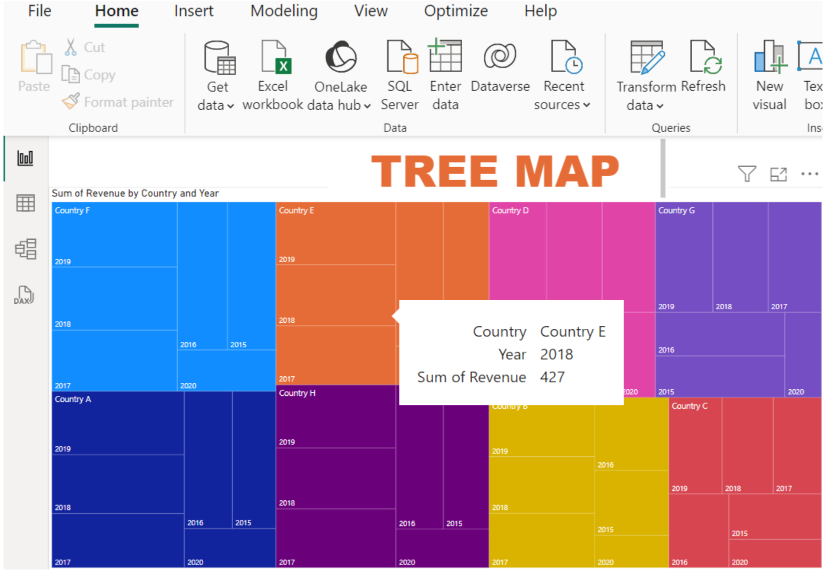
Furthermore, I used Power BI to create interactive visualizations:

1. **Pie Chart:** The pie chart illustrates the revenue share by country. Each slice represents a country's contribution to the total revenue, making it easy to compare the relative performance of different countries.



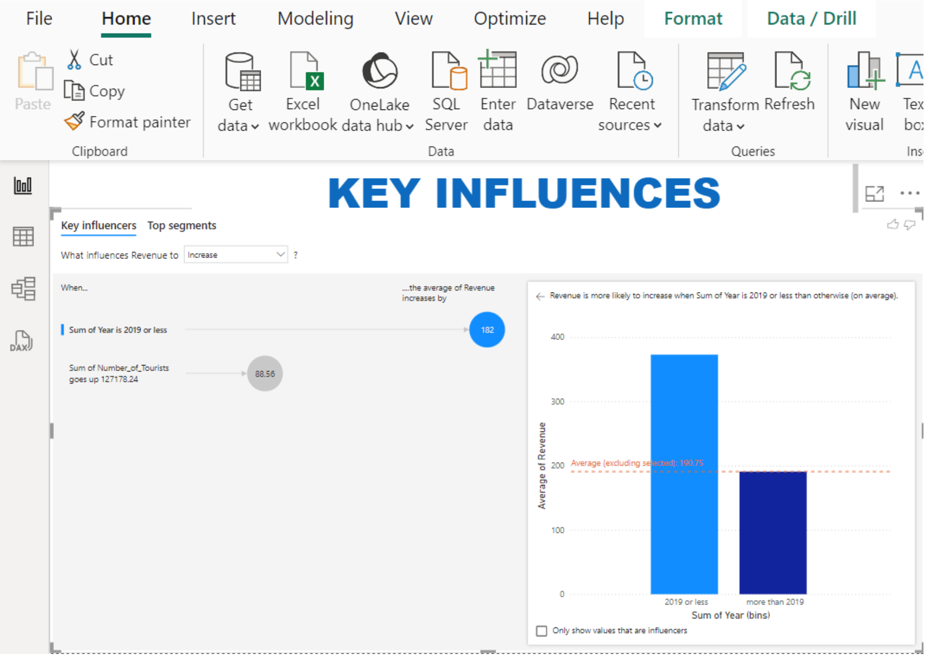
Country Country F
Sum of Revenue 2478 (15.08%)

2. **Treemap:** The treemap visualizes the contribution of each country to the total number of tourists. It uses a hierarchical layout to show the proportional size of each country's tourist numbers, providing a clear and immediate comparison.



Country Country E
Year 2018
Sum of Revenue 427

3. **Key Influences:** The Key Influences visual helps to identify the factors that have the most significant impact on the number of tourists. It highlights which variables, such as revenue or specific years, influence tourist numbers the most. This visual is particularly useful for uncovering insights and driving data-driven decisions.



What influences Revenue to Increase ?

When...

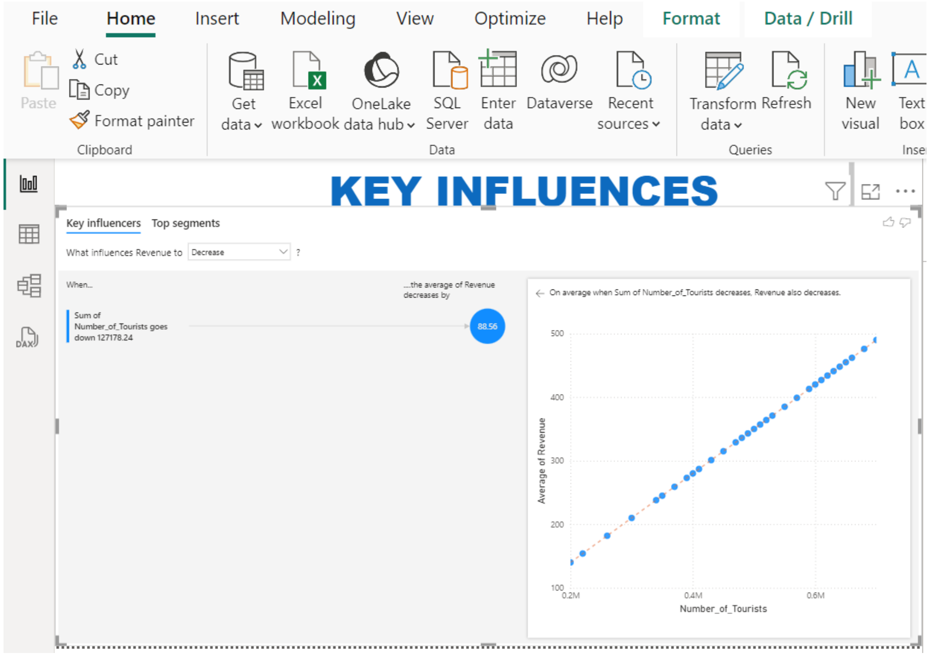
....the average of Revenue increases by

Sum of Year is 2019 or less — 182

Sum of Number_of_Tourists goes up 127178.24 — 88.56

Revenue is more likely to increase when Sum of Year is 2019 or less than otherwise (on average).

☐ Only show values that are influencers



What influences Revenue to Decrease ?

When...

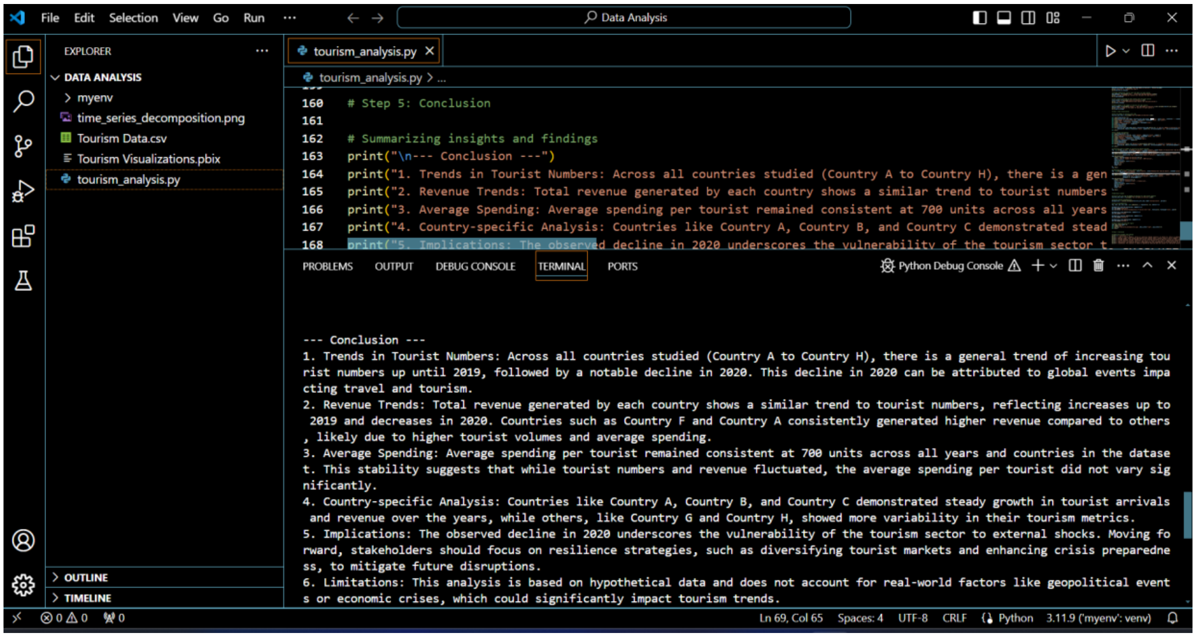
....the average of Revenue decreases by

Sum of Number_of_Tourists goes down 127178.24 — 88.56

On average when Sum of Number_of_Tourists decreases, Revenue also decreases.

Conclusion

Through this project, I gained valuable insights into the trends and patterns in the tourism world. The combination of Python for detailed analysis and Power BI for interactive visualizations has provided a comprehensive understanding of the data. I hope this document has demonstrated the power of combining these tools to analyze and visualize data effectively.



```
# Step 5: Conclusion

# Summarizing insights and findings
print("\n--- Conclusion ---")
print("1. Trends in Tourist Numbers: Across all countries studied (Country A to Country H), there is a gen
print("2. Revenue Trends: Total revenue generated by each country shows a similar trend to tourist numbers
print("3. Average Spending: Average spending per tourist remained consistent at 700 units across all years
print("4. Country-specific Analysis: Countries like Country A, Country B, and Country C demonstrated stead
print("5. Implications: The observed decline in 2020 underscores the vulnerability of the tourism sector t
```

```
--- Conclusion ---
1. Trends in Tourist Numbers: Across all countries studied (Country A to Country H), there is a general trend of increasing tou
rist numbers up until 2019, followed by a notable decline in 2020. This decline in 2020 can be attributed to global events impa
cting travel and tourism.
2. Revenue Trends: Total revenue generated by each country shows a similar trend to tourist numbers, reflecting increases up to
 2019 and decreases in 2020. Countries such as Country F and Country A consistently generated higher revenue compared to others
, likely due to higher tourist volumes and average spending.
3. Average Spending: Average spending per tourist remained consistent at 700 units across all years and countries in the datase
t. This stability suggests that while tourist numbers and revenue fluctuated, the average spending per tourist did not vary sig
nificantly.
4. Country-specific Analysis: Countries like Country A, Country B, and Country C demonstrated steady growth in tourist arrivals
 and revenue over the years, while others, like Country G and Country H, showed more variability in their tourism metrics.
5. Implications: The observed decline in 2020 underscores the vulnerability of the tourism sector to external shocks. Moving fo
rward, stakeholders should focus on resilience strategies, such as diversifying tourist markets and enhancing crisis preparedne
ss, to mitigate future disruptions.
6. Limitations: This analysis is based on hypothetical data and does not account for real-world factors like geopolitical event
s or economic crises, which could significantly impact tourism trends.
```

x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---