CareMetrics

HEALTHCARE & DATA ANALYSIS

Project name:

Cara Metrics.

Project description:

The Healthcare Data Insights project aims to extract valuable insights from the comprehensive healthcare dataset. By analyzing patient data, we seek to uncover trends, patterns, and associations that can inform healthcare decision-making and improve patient outcomes.

Project objectives:

1. Identify Healthcare Trends: Analyze data to uncover trends in healthcare utilization, disease prevalence, and treatment outcomes.

2. Optimize Resource Allocation: Provide insights for better management of staffing, bed capacity, and equipment.

3. Enhance Patient Care: Identify areas to improve patient care, reduce readmissions, and boost patient satisfaction.

4. Support Evidence-Based Decisions: Offer data-driven insights for healthcare professionals and policymakers to make informed decisions.

5. Improve Hospital Efficiency: Recommend ways to enhance operational efficiency and reduce healthcare costs.

Methodology:

- Data Overview:

The dataset contains 55,501 rows and 15 columns, covering patient demographics, medical conditions, admission details, medications, and financial information.

- Data Collection:

The data was sourced from Kaggle and contained structured healthcare data involving patients' medical and administrative records.

- Data Cleaning:

Identify and handle missing or inconsistent values (e.g., incomplete patient records or missing discharge dates). Standardize categorical variables such as gender and blood type. Correct formatting inconsistencies in numerical fields like billing amounts

- Data Exploration:

Use descriptive statistics to summarize patient demographics (age, gender) and health conditions. Explore correlations between variables (e.g., length of stay and medical conditions). Visualize key metrics such as common medical conditions, medication usage, and admission types.

- Advanced Analysis:

Apply predictive models to analyze factors influencing patient outcomes (e.g., readmission rates, cost of care). Clustering will be conducted to identify patient groups based on their medical and treatment history.

- Data Visualization:

Generate charts and graphs to present trends (e.g., distribution of admission types, and test results).

Summary:

- Similar prevalence across genders:

There is no significant difference between the number of females and males for each medical condition. This suggests that the prevalence of these medical conditions is relatively balanced between the sexes.

- There is no significant difference by blood type:

The counts of each medical condition (e.g., arthritis, asthma, cancer, diabetes, hypertension, and obesity) are quite similar across different blood types. There are no very large differences that are prominent, suggesting that blood type may not play a major role in determining susceptibility to these conditions. Most common conditions: The conditions with the highest counts among all blood types are diabetes and hypertension, suggesting that these conditions are generally more prevalent in the sample population. Highest and lowest counts: Diabetes: Blood type B+ has the highest count (1196), while blood type O- has the lowest count (1122). High blood pressure: Blood type AB+ has the highest number (1215), while blood type O- has the lowest number (1145). Asthma and obesity: These conditions are relatively evenly distributed among all blood types.

- The analysis identifies the hospitals with the highest and lowest average billing amounts.

The results show a clear distinction between the most and least expensive hospitals based on their average billing charges. These hospitals have the highest billing amounts, with Hernandez-Morton being the most expensive. The average charges for these hospitals are all above $52,000. Least Expensive Hospitals: Rowe, Stone, and Patterson: $49,450.12

- Summary of Infection Trends by Age and Medical Condition:

1. Cancer: Highest infection rates are observed in individuals in their mid-fifties. A significant concentration of infections occurs from the early to late fifties. Early fifties infections show a correlation with obesity.

2. Diabetes: Most infections are concentrated in individuals in their late thirties and late forties.

3. Asthma: Infection rates peak in the mid-twenties and early fifties.

4. High Blood Pressure: Infection rates rise in the early twenties and again in the early seventies, with noticeable increases also in the late thirties.

5. Joint Disease: Infection rates are relatively consistent across most ages, but there is a noticeable increase in the late thirties and late fifties.

- The analysis shows how different medical conditions are distributed across three types of hospital admissions:

Elective Admissions: For elective admissions, conditions such as cancer and diabetes have relatively high occurrence rates. This indicates that many patients are likely planning their hospital stays for treatments or management of chronic conditions.

Emergency Admissions: Emergency admissions see high occurrences for conditions like asthma and hypertension. This suggests that these medical conditions often require urgent intervention, potentially due to sudden complications or exacerbations.

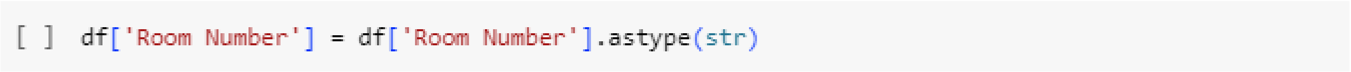
Urgent Admissions: Urgent admissions also have a notable number of cases for conditions such as obesity and diabetes, indicating that these conditions may lead to health emergencies that necessitate immediate care but are less sudden than typical emergencies.

From this data, we can observe that the billing amounts are quite similar across different insurance providers, with only slight variations. Medicare seems to have the highest average billing amount, while UnitedHealthcare has the lowest.

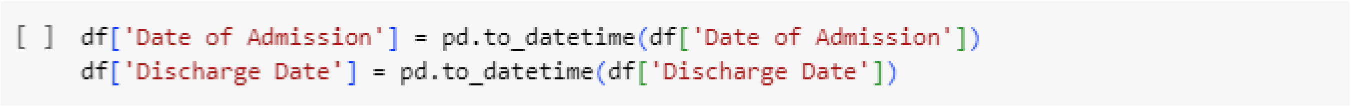
Conclusion:

In conclusion, this analysis provides valuable insights into the relationships between medical conditions, demographics, hospital billing, and admission types. The findings reveal that gender and blood type have little impact on the prevalence of common medical conditions, while hospital billing varies significantly between institutions. Additionally, the distribution of medical conditions across admission types emphasizes the need for both planned and urgent care in managing chronic diseases. These results can help healthcare providers and policymakers better understand patterns in medical care, ultimately improving patient outcomes and resource allocation.

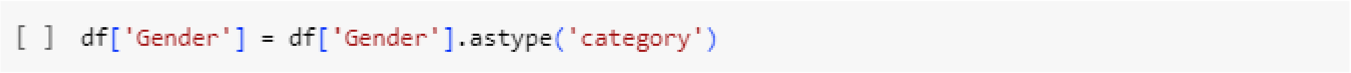
The python cods:



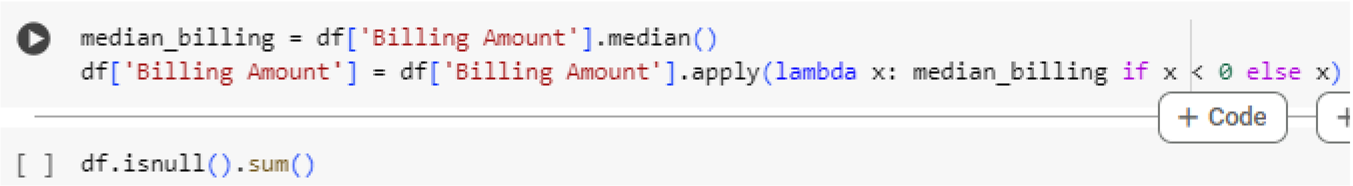
```
[ ] df['Room Number'] = df['Room Number'].astype(str)
```



```
[ ] df['Date of Admission'] = pd.to_datetime(df['Date of Admission'])
    df['Discharge Date'] = pd.to_datetime(df['Discharge Date'])
```

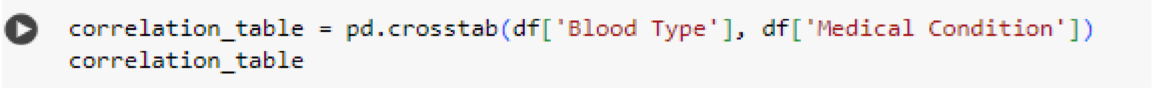


```
[ ] df['Gender'] = df['Gender'].astype('category')
```

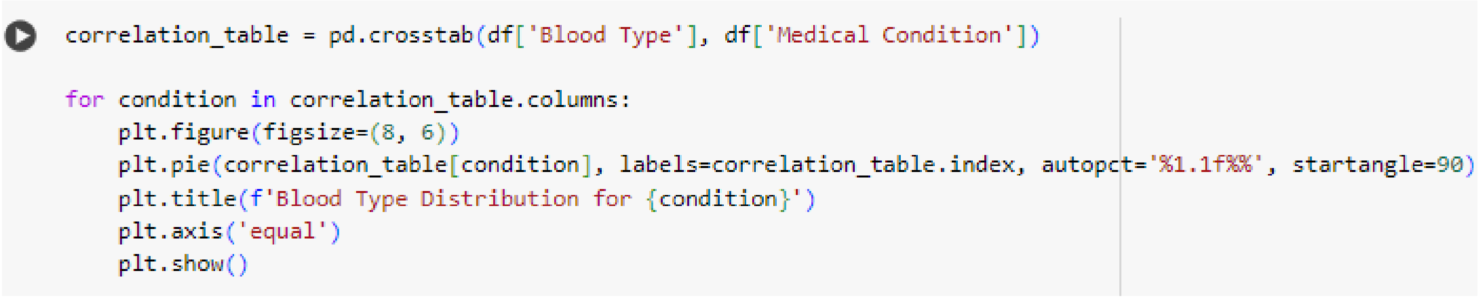


```
median_billing = df['Billing Amount'].median()
df['Billing Amount'] = df['Billing Amount'].apply(lambda x: median_billing if x < 0 else x)
```

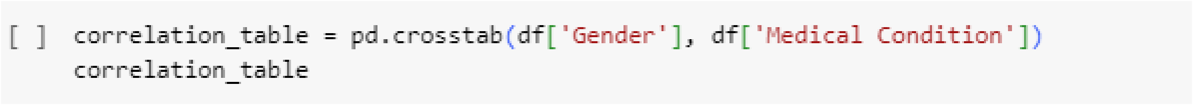
```
[ ] df.isnull().sum()
```



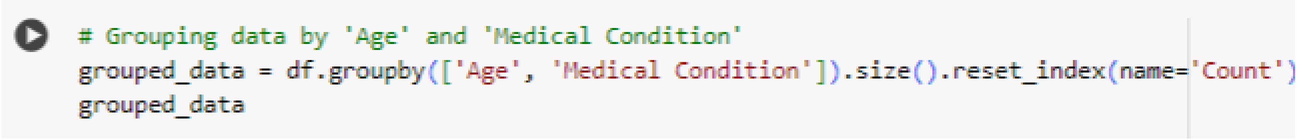
```
correlation_table = pd.crosstab(df['Blood Type'], df['Medical Condition'])
correlation_table
```



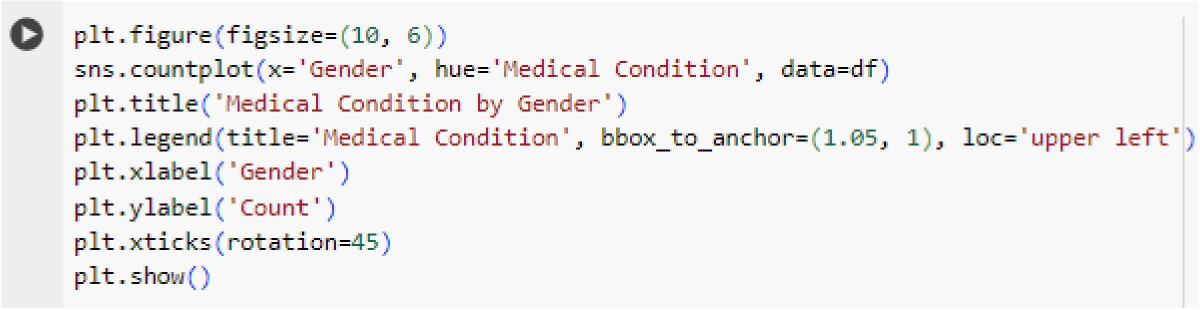
```
correlation_table = pd.crosstab(df['Blood Type'], df['Medical Condition'])

for condition in correlation_table.columns:
    plt.figure(figsize=(8, 6))
    plt.pie(correlation_table[condition], labels=correlation_table.index, autopct='%1.1f%%', startangle=90)
    plt.title(f'Blood Type Distribution for {condition}')
    plt.axis('equal')
    plt.show()
```



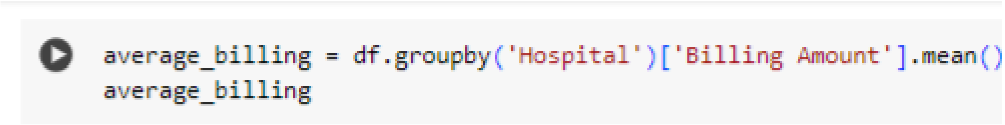
```
[ ] correlation_table = pd.crosstab(df['Gender'], df['Medical Condition'])
    correlation_table
```



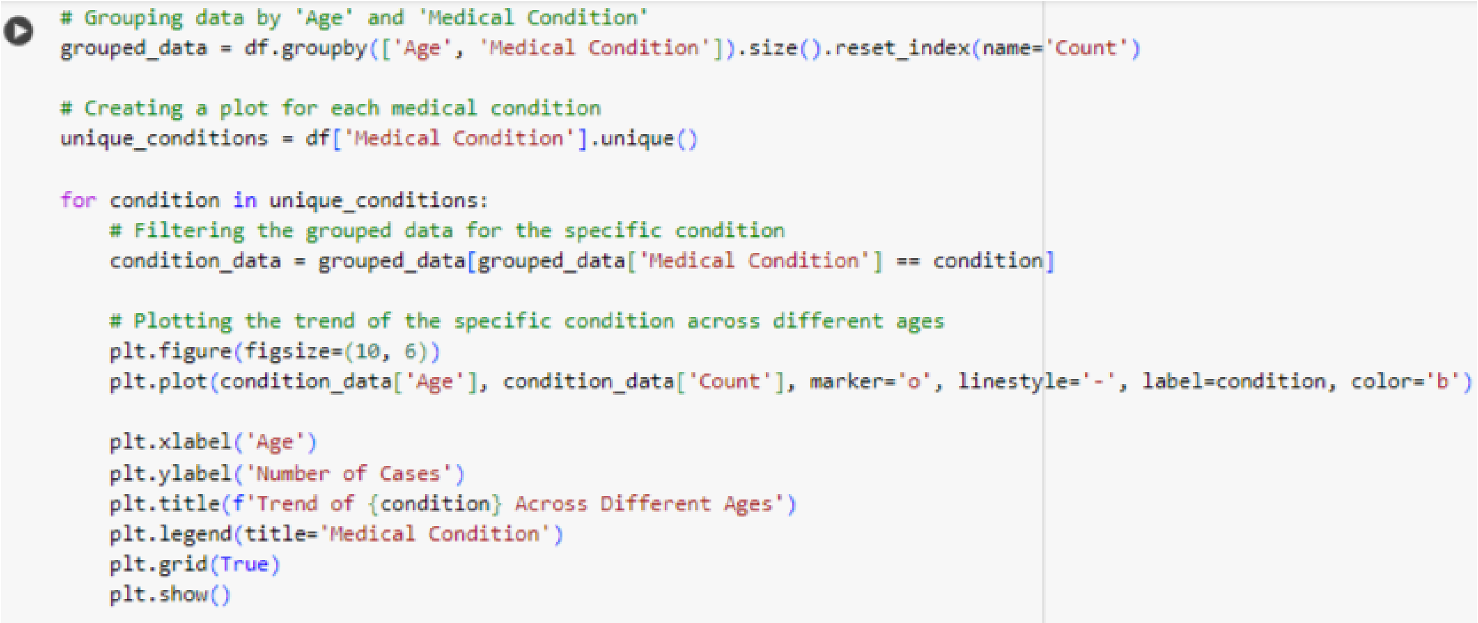
```
# Grouping data by 'Age' and 'Medical Condition'
grouped_data = df.groupby(['Age', 'Medical Condition']).size().reset_index(name='Count')
grouped_data
```



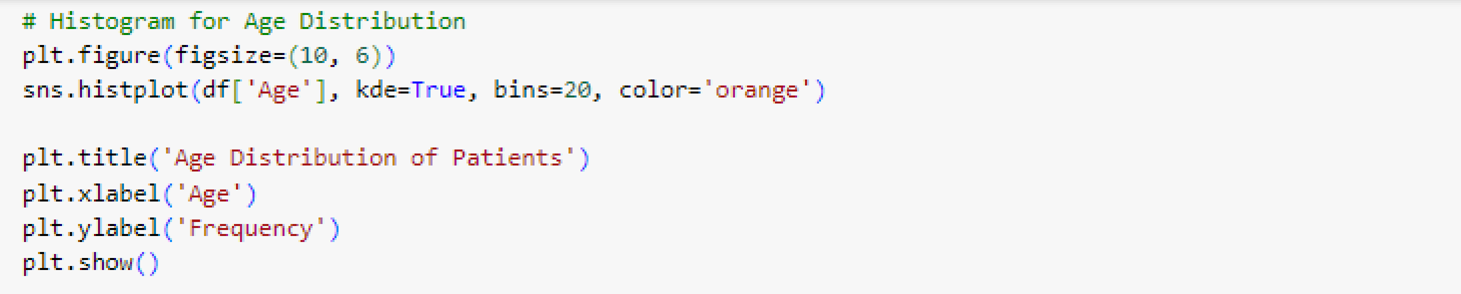
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Gender', hue='Medical Condition', data=df)
plt.title('Medical Condition by Gender')
plt.legend(title='Medical Condition', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



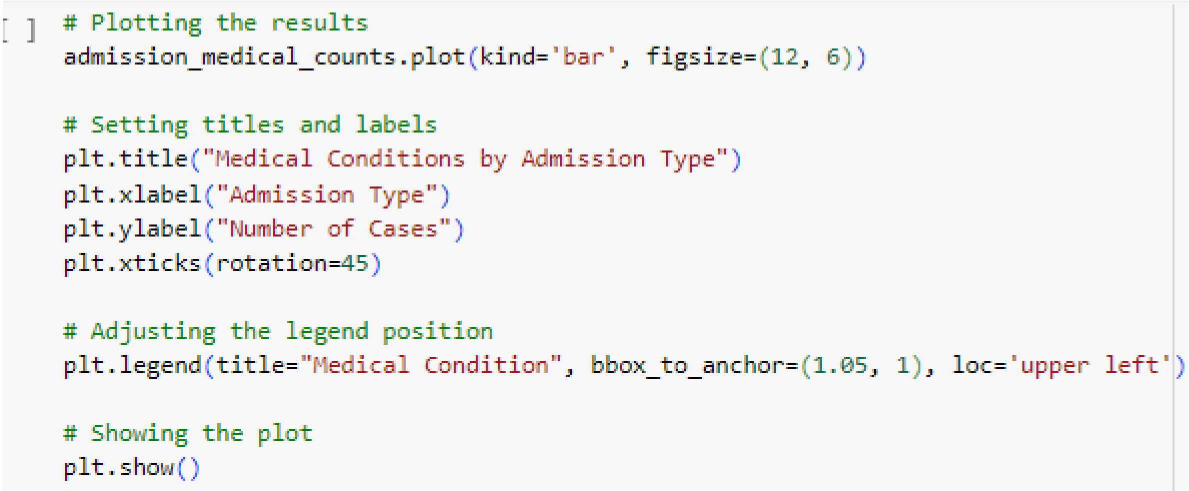
```
average_billing = df.groupby('Hospital')['Billing Amount'].mean()
average_billing
```



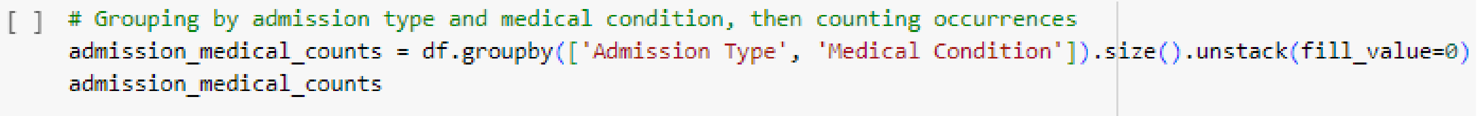
```
# Grouping data by 'Age' and 'Medical Condition'
grouped_data = df.groupby(['Age', 'Medical Condition']).size().reset_index(name='Count')

# Creating a plot for each medical condition
unique_conditions = df['Medical Condition'].unique()

for condition in unique_conditions:
    # Filtering the grouped data for the specific condition
    condition_data = grouped_data[grouped_data['Medical Condition'] == condition]

    # Plotting the trend of the specific condition across different ages
    plt.figure(figsize=(10, 6))
    plt.plot(condition_data['Age'], condition_data['Count'], marker='o', linestyle='-', label=condition, color='b')

    plt.xlabel('Age')
    plt.ylabel('Number of Cases')
    plt.title(f'Trend of {condition} Across Different Ages')
    plt.legend(title='Medical Condition')
    plt.grid(True)
    plt.show()
```



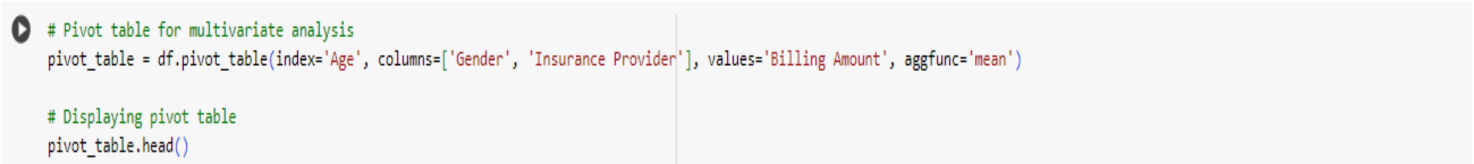
```
# Histogram for Age Distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['Age'], kde=True, bins=20, color='orange')

plt.title('Age Distribution of Patients')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



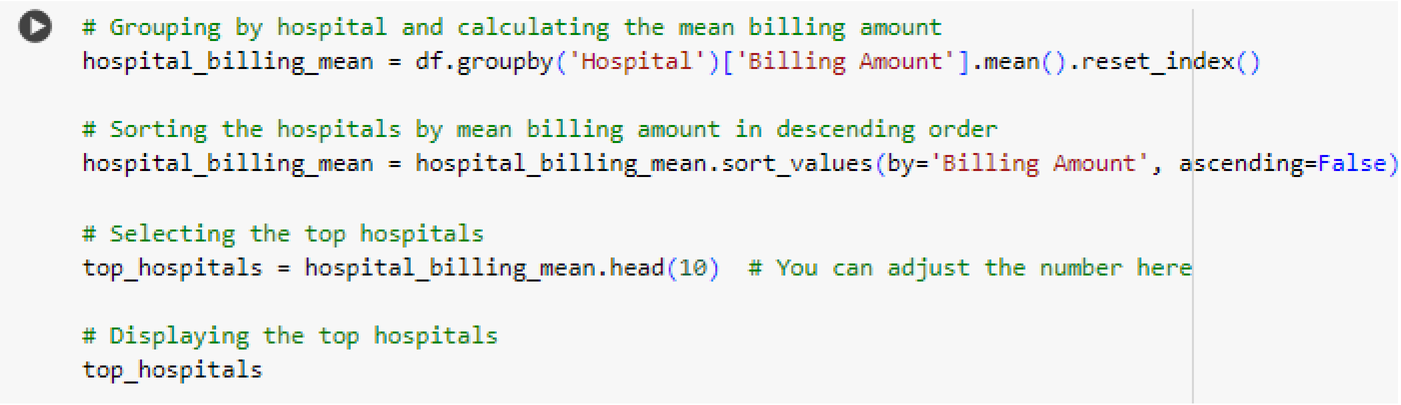
```
[ ] # Plotting the results
    admission_medical_counts.plot(kind='bar', figsize=(12, 6))

    # Setting titles and labels
    plt.title("Medical Conditions by Admission Type")
    plt.xlabel("Admission Type")
    plt.ylabel("Number of Cases")
    plt.xticks(rotation=45)

    # Adjusting the legend position
    plt.legend(title="Medical Condition", bbox_to_anchor=(1.05, 1), loc='upper left')

    # Showing the plot
    plt.show()
```



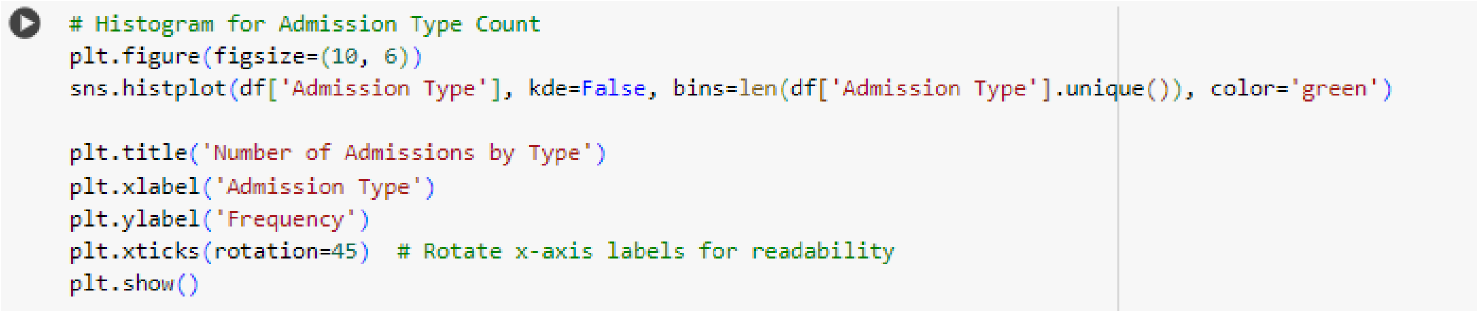
```
[ ] # Grouping by admission type and medical condition, then counting occurrences
    admission_medical_counts = df.groupby(['Admission Type', 'Medical Condition']).size().unstack(fill_value=0)
    admission_medical_counts
```



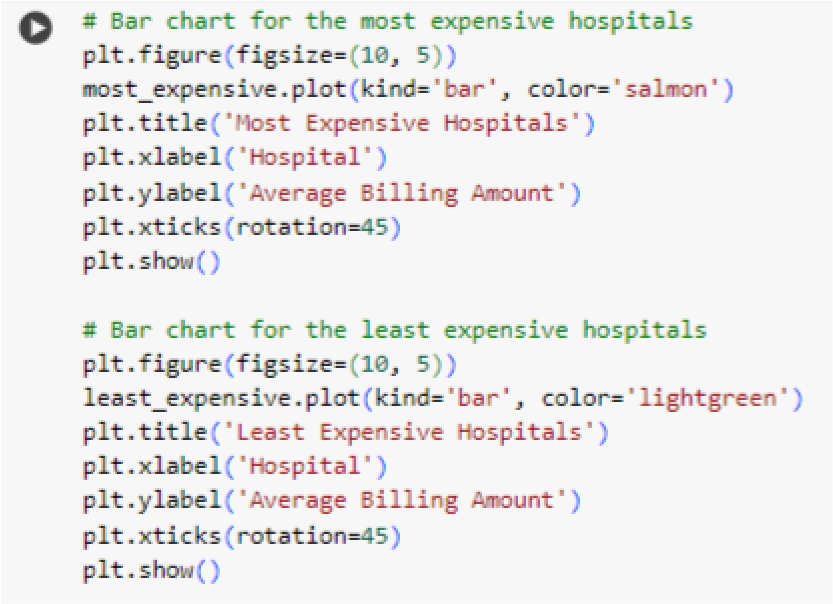
```
# Pivot table for multivariate analysis
pivot_table = df.pivot_table(index='Age', columns=['Gender', 'Insurance Provider'], values='Billing Amount', aggfunc='mean')

# Displaying pivot table
pivot_table.head()
```



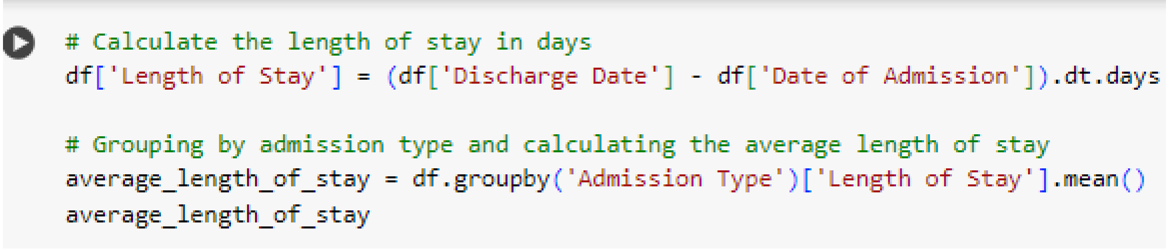
```
# Grouping by hospital and calculating the mean billing amount
hospital_billing_mean = df.groupby('Hospital')['Billing Amount'].mean().reset_index()

# Sorting the hospitals by mean billing amount in descending order
hospital_billing_mean = hospital_billing_mean.sort_values(by='Billing Amount', ascending=False)

# Selecting the top hospitals
top_hospitals = hospital_billing_mean.head(10)  # You can adjust the number here

# Displaying the top hospitals
top_hospitals
```



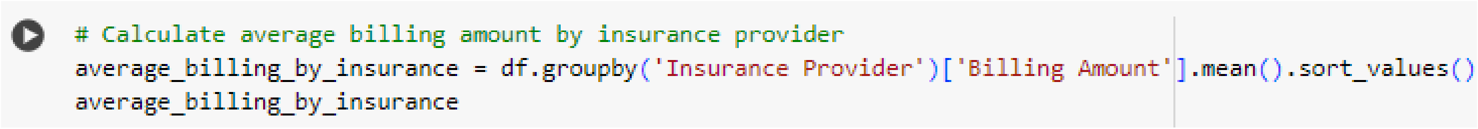
```
# Histogram for Admission Type Count
plt.figure(figsize=(10, 6))
sns.histplot(df['Admission Type'], kde=False, bins=len(df['Admission Type'].unique()), color='green')

plt.title('Number of Admissions by Type')
plt.xlabel('Admission Type')
plt.ylabel('Frequency')
plt.xticks(rotation=45)  # Rotate x-axis labels for readability
plt.show()
```



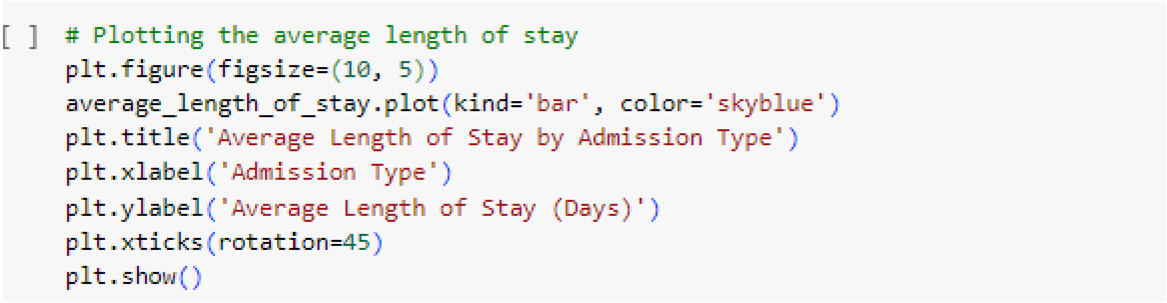
```
# Bar chart for the most expensive hospitals
plt.figure(figsize=(10, 5))
most_expensive.plot(kind='bar', color='salmon')
plt.title('Most Expensive Hospitals')
plt.xlabel('Hospital')
plt.ylabel('Average Billing Amount')
plt.xticks(rotation=45)
plt.show()

# Bar chart for the least expensive hospitals
plt.figure(figsize=(10, 5))
least_expensive.plot(kind='bar', color='lightgreen')
plt.title('Least Expensive Hospitals')
plt.xlabel('Hospital')
plt.ylabel('Average Billing Amount')
plt.xticks(rotation=45)
plt.show()
```



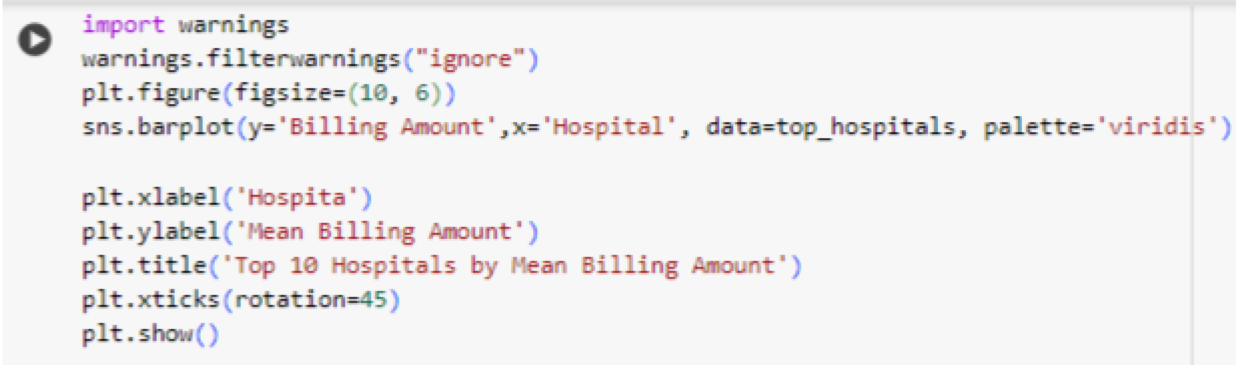
```
# Calculate the length of stay in days
df['Length of Stay'] = (df['Discharge Date'] - df['Date of Admission']).dt.days

# Grouping by admission type and calculating the average length of stay
average_length_of_stay = df.groupby('Admission Type')['Length of Stay'].mean()
average_length_of_stay
```



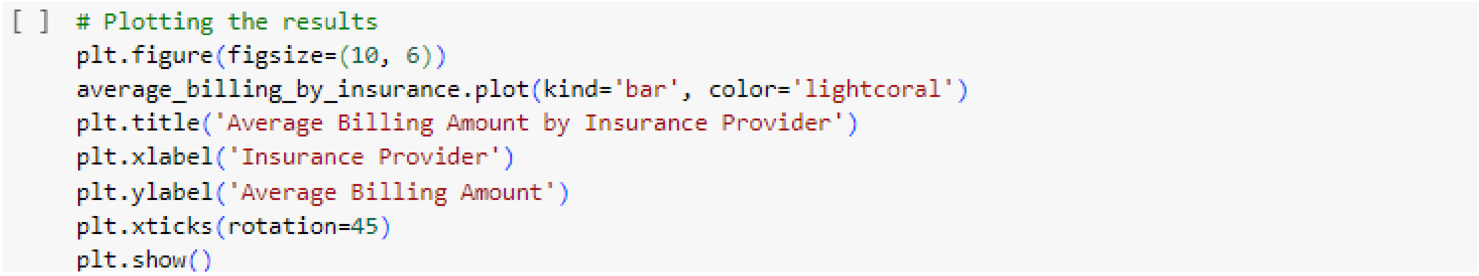
```
# Calculate average billing amount by insurance provider
average_billing_by_insurance = df.groupby('Insurance Provider')['Billing Amount'].mean().sort_values()
average_billing_by_insurance
```



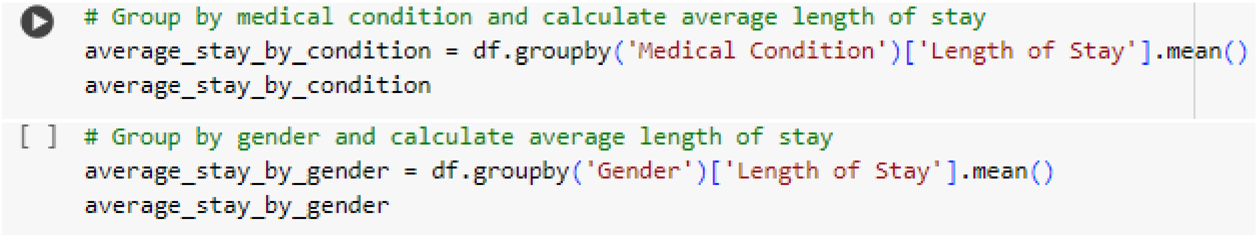
```
[ ]  # Plotting the average length of stay
     plt.figure(figsize=(10, 5))
     average_length_of_stay.plot(kind='bar', color='skyblue')
     plt.title('Average Length of Stay by Admission Type')
     plt.xlabel('Admission Type')
     plt.ylabel('Average Length of Stay (Days)')
     plt.xticks(rotation=45)
     plt.show()
```



```
import warnings
warnings.filterwarnings("ignore")
plt.figure(figsize=(10, 6))
sns.barplot(y='Billing Amount',x='Hospital', data=top_hospitals, palette='viridis')

plt.xlabel('Hospita')
plt.ylabel('Mean Billing Amount')
plt.title('Top 10 Hospitals by Mean Billing Amount')
plt.xticks(rotation=45)
plt.show()
```

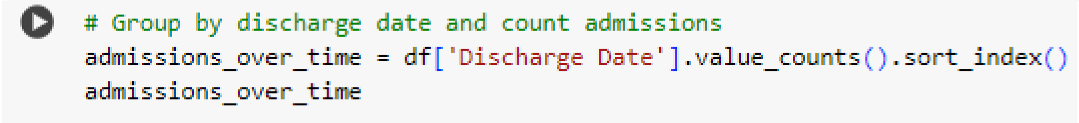


```
[ ]  # Plotting the results
     plt.figure(figsize=(10, 6))
     average_billing_by_insurance.plot(kind='bar', color='lightcoral')
     plt.title('Average Billing Amount by Insurance Provider')
     plt.xlabel('Insurance Provider')
     plt.ylabel('Average Billing Amount')
     plt.xticks(rotation=45)
     plt.show()
```

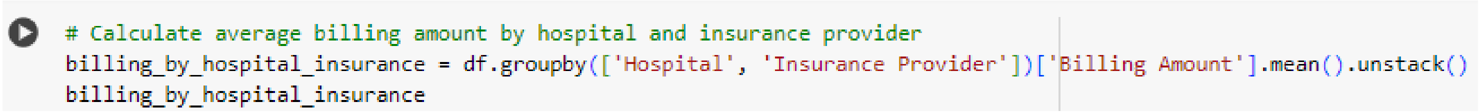


```
# Group by medical condition and calculate average length of stay
average_stay_by_condition = df.groupby('Medical Condition')['Length of Stay'].mean()
average_stay_by_condition
```

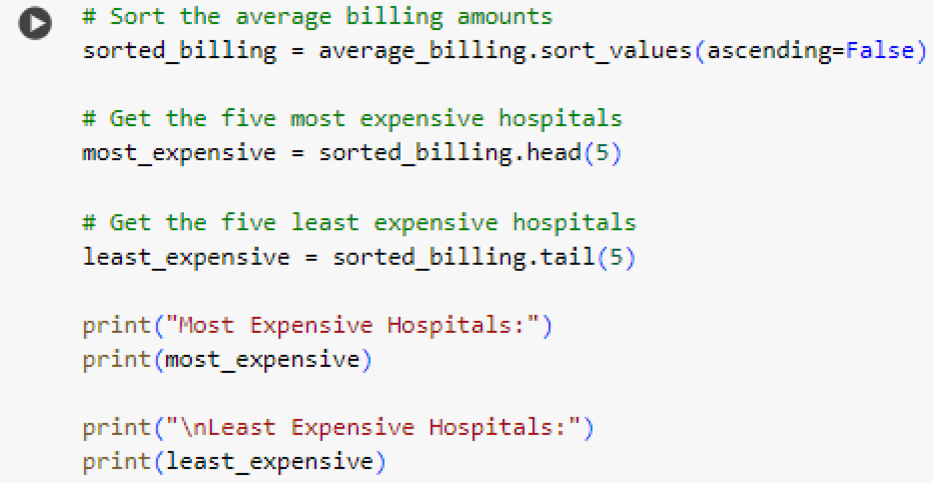
```
[ ]  # Group by gender and calculate average length of stay
     average_stay_by_gender = df.groupby('Gender')['Length of Stay'].mean()
     average_stay_by_gender
```



```
# Group by discharge date and count admissions
admissions_over_time = df['Discharge Date'].value_counts().sort_index()
admissions_over_time
```



```
# Calculate average billing amount by hospital and insurance provider
billing_by_hospital_insurance = df.groupby(['Hospital', 'Insurance Provider'])['Billing Amount'].mean().unstack()
billing_by_hospital_insurance
```



```
# Sort the average billing amounts
sorted_billing = average_billing.sort_values(ascending=False)

# Get the five most expensive hospitals
most_expensive = sorted_billing.head(5)

# Get the five least expensive hospitals
least_expensive = sorted_billing.tail(5)

print("Most Expensive Hospitals:")
print(most_expensive)

print("\nLeast Expensive Hospitals:")
print(least_expensive)
```

Tools used:

1.Excel: analyze data and create graphs and reports, used to analyze patient data and create visual graphs and reports that highlight patient admission patterns, treatment types, and distribution across departments.

(Pivot tables, Charts, Filter, Conditional formatting).

2.Python: Python was used in this project to analyze healthcare data through the use of Pandas for data manipulation and various functions to perform calculations and extract insights from the datasets.

3.SQL: was used to query, clean, and analyze the healthcare data. A custom schema was created to relate patient, doctor, hospital, and billing tables. Key tasks included filtering data, calculating hospital bills, and summarizing patient outcomes.

4.Tableau: is used to visualize healthcare data through interactive dashboards. The visualizations provided insights into patient demographics, disease distribution, hospital billing trends, doctor performance, and insurance provider statistics. Key metrics were represented using bar charts, line graphs, and heat maps, with filters and sliders to explore data across different time frames, age groups, and hospital departments.