Health Care Dataset Report

Adarsh Anil

November 17, 2024

Introduction

Healthcare is a critical pillar of any society, reflecting the well-being and productivity of its population. In an era where data-driven decision-making has become the cornerstone of progress, the healthcare industry has embraced analytics to enhance its services, improve patient outcomes, and optimize operational efficiency. This report delves into a comprehensive healthcare dataset comprising 55,500 records and 15 features, offering a detailed exploration of patient demographics, medical conditions, financial metrics, and service patterns.

The dataset captures diverse aspects of patient care, ranging from basic demographic information such as age, gender, and blood type to intricate details like medical conditions, treatment patterns, and financial transactions. Each record represents a unique patient interaction, encompassing variables like admission types, insurance coverage, assigned doctors, and hospitals attended. These data points provide a unique opportunity to uncover valuable insights into the complexities of healthcare delivery and resource utilization.

Healthcare analytics holds immense potential to address key challenges faced by the industry, including rising treatment costs, uneven resource allocation, and the growing demand for personalized care. By systematically analyzing patterns in billing amounts, patient stay durations, and admission types, this study aims to shed light on areas requiring attention, such as chronic disease management, preventive care initiatives, and the impact of demographic factors on healthcare access. Additionally, understanding trends in hospital and doctor performance can help administrators streamline operations and allocate resources more effectively.

Moreover, the dataset enables granular analysis of financial metrics, revealing insights into total and average billing amounts, variations across demographic groups, and the influence of insurance providers on healthcare expenditure. The interplay between admission types (elective, urgent, emergency) and patient outcomes highlights the importance of tailoring care pathways to suit the specific needs of different patient categories.

This report also delves into the role of medical conditions and prescribed medications in shaping healthcare trends. By identifying the most common diagnoses and treatments, the analysis highlights the conditions contributing significantly to the healthcare burden

and presents opportunities for targeted interventions. Furthermore, the study investigates seasonal variations in patient admissions and discharges, emphasizing the need for resource planning during peak periods.

As the world continues to face challenges posed by an aging population, evolving disease profiles, and the need for equitable healthcare access, data like this becomes a vital resource. The findings from this dataset not only inform hospital administrators and policymakers but also provide a foundation for improving patient care quality, reducing costs, and enhancing operational efficiency. In a rapidly changing healthcare landscape, leveraging analytics for actionable insights is no longer optional—it is a necessity.

Through a meticulous examination of this dataset, this report aims to present a detailed narrative of the healthcare system, unearth hidden trends, and propose actionable insights for stakeholders. It seeks to contribute to the broader goal of fostering a data-informed, patient-centric healthcare ecosystem that is prepared to meet future challenges.

Dataset Overview

The dataset contains 55,500 rows and 15 columns. Here's an overview of the columns:

- **Name**:Patient's name.
- **Age**: Patient's age.
- **Gender**:Gender of the patient
- **Blood Type**: Blood type of the patient.
- **Medical Condition**: Primary medical condition.
- **Date of Admission**: Date of admission in hospital.
- **Doctor**: Doctor whom patient want to see.
- **Hospital**: Hospital to be visited by the patient.
- **Insurance Provider**:Insurance Company who provides insurance to patients.
- **Billing Amount**:Bill patient have to pay in hospital.
- **Room number**: Room where the patient is being assigned to meet the doctor.
- **Admission Type**: Type of admission of patient in hospital.
- **Discharge Date**: Date of Discharge from hospital.
- **Medication**: Medicines prescribed by the doctor for the patient.
- **Test Results**: Outcome of the test.

Objective

The primary objective of selecting this healthcare dataset is to analyze and uncover meaningful insights into the multifaceted aspects of patient care, resource allocation, financial management, and operational efficiency within the healthcare industry. With a comprehensive scope of 55,500 records and 15 diverse features, this dataset provides a robust foundation for exploring critical factors that influence healthcare delivery, patient outcomes, and cost management.

Healthcare is a data-intensive sector where the effective analysis of information can lead to transformative improvements in patient care and administrative practices. This dataset captures a wide range of attributes, including patient demographics, medical histories, admission types, financial details, and hospital and doctor performance metrics, making it an ideal choice for addressing key questions faced by the industry. By focusing on this dataset, we aim to achieve the following specific objectives:

1. Enhancing Patient Care
   The dataset offers an opportunity to study patient demographics such as age, gender, and blood type, alongside medical conditions and prescribed medications. This enables an understanding of patient profiles, prevalent health issues, and patterns in treatment outcomes. The goal is to identify trends that can improve personalized care strategies, optimize treatment plans, and reduce hospital readmissions.

2. Optimizing Resource Utilization
   By analyzing admission types (elective, urgent, emergency), length of hospital stays, and seasonal patterns of patient admissions, the dataset allows for an in-depth assessment of resource demands. Insights derived from this analysis can help hospitals and clinics better plan for peak periods, allocate medical staff and equipment efficiently, and enhance the overall quality of care.

3. Financial Analysis and Cost Reduction
   The inclusion of financial metrics such as total billing amounts, average costs, and insurance coverage provides a lens to study the economic aspects of healthcare. Identifying patterns in billing across demographic groups and conditions can help pinpoint cost drivers and suggest strategies for reducing healthcare expenses without compromising care quality.

4. Hospital and Doctor Performance Metrics
   The dataset tracks interactions across multiple hospitals and healthcare providers, enabling an evaluation of performance at both institutional and individual levels. This analysis is crucial for identifying best practices, addressing gaps in service quality, and fostering accountability among healthcare professionals.

5. Data-Driven Decision-Making
   The rich diversity of this dataset supports the application of advanced data analysis techniques, such as statistical modeling, predictive analytics, and machine learning. Leveraging these methods can lead to actionable insights that empower healthcare administrators, policymakers, and practitioners to make informed decisions that benefit both patients and providers.

6. Fostering Equity in Healthcare
   Understanding how demographics and socioeconomic factors influence access to care and treatment outcomes is essential for promoting equity. This dataset offers a chance to explore disparities in healthcare delivery and propose solutions for creating a more inclusive and accessible healthcare system.

7. Addressing Public Health Challenges
   Chronic diseases, aging populations, and rising healthcare costs are pressing global concerns. By examining trends in medical conditions and treatment patterns within the dataset, this analysis aims to contribute to broader public health initiatives focused on prevention, early diagnosis, and effective disease management.

Data Cleaning

Data cleaning was performed to ensure the accuracy and reliability of the dataset. This included:

1. Eliminate Null Values

```

-- Check if any null value exist or not
df.isna().sum()
```

Insight: The dataset didn't contain any null values. So no need to eliminate any null values or fill any values using fillna().

2. Eliminate Duplicates

```
-- Check if any duplicates exists or not
df.duplicated()

-- Find how many duplicates are there
df.duplicated().sum()

-- Drop the duplicates
df.drop_duplicates(inplace=True)

-- Recheck whether any duplicates is still there
df.duplicated().sum()

-- Check the new shape of the dataset
df.shape
```

Insight: When duplicates were checked in the dataset, 534 duplicates were found and removed using drop_duplicates(). Afterward, the dataset was rechecked to ensure no further duplicates were present. Finally, the dataset was found to contain 54,966 rows after duplicate removal.

3. Data Type conversion

```
-- Check the datatype of all columns
df.dtypes

-- Convert Date of Admission to date_time
df['Date of Admission'] = pd.to_datetime(df['Date of Admission'],
    format ='%Y-%m-%d')

-- Convert Discharge Date to date_time
df['Discharge Date'] = pd.to_datetime(df['Discharge Date'],format='%Y-%
    m-%d')

-- Recheck whether the columns were converted or not.
df.dtypes
```

Insight: The datatypes of all columns were checked, and it was found that Date_of_Admission and Discharge_Date were in text format, which needed to be converted to date format for proper analysis. Both columns were converted to date_time format using the to_datetime function in pandas. It was then rechecked to confirm that the datatypes were successfully changed.

4. Outlier Detection

```
-- Dictionary to store the minimum range and maximum range of numerical
     columns.
dict1 ={}

for col in df:
  if df[col].dtype =='int' or df[col].dtype=='float':

-- First Quantile
    q1 =df[col].quantile(0.25)

-- Third Quantile
    q3 = df[col].quantile(0.75)

-- IQR
    iqr =q3-q1

-- Minimum and maximum range
    min_range = q1 -1.5 * iqr
    max_range = q3 + 1.5 *iqr

-- Check any outliers are there or not
    if not df[(df[col] <= max_range) & (df[col] >= min_range)].equals(
        df[col]):
      dict1[col] =(min_range,max_range)

-- Print the output
print(dict1)
```

Insight: The numerical columns were checked for outliers. For each column, the minimum and maximum range were calculated, and the values were examined to see if any were below the minimum range or above the maximum range. It was found that no outliers were present.

Feature Engineering

Feature engineering was performed to create new columns that enhance the analysis of the dataset.

Creating Year of Admission

```

-- Add year_admission column
df['year_admission'] = df['Date of Admission'].dt.year

-- Overview of the new column
df['year_admission']
```

Insight: The year_admission column enables year-over-year comparisons of admissions, helping to identify trends in the dataset.

Creating Month of Admission

```

-- Add month_admission column
df['month_admission']=df['Date of Admission'].dt.month_name()

-- Overview of the new column
df['month_admission']
```

Insight: In the month_admission, we can compare the month in which admissions peak throughout the year.

Creating Day of Admission

```

-- Add day_admission column
df['day_admission'] =df['Date of Admission'].dt.day_name()

-- Overview of the new column
df['day_admission']
```

Insight: The day_admission columns enable day-over-day comparisons of admissions, helping to find trends in the dataset.

Creating Year of Discharge

```

-- Add year_Discharge column
df['year_Discharge'] =df['Discharge Date'].dt.year

-- Overview of the new column
df['year_Discharge']
```

Insight: The year_Discharge column enables year-over-year comparisons of discharge, helping to identify trends in the dataset.

Creating Month of Discharge

```
-- Add month_Discharge column
df['month_Discharge'] =df['Discharge Date'].dt.month_name()

-- Overview of the new column
df['month_Discharge']
```

Insight: In the month_Discharge, we can compare the month in which discharges peak throughout the year.

Creating Day of Discharge

```

-- Add day_Discharge column
df['day_Discharge'] =df['Discharge Date'].dt.day_name()

-- Overview of the new column
df['day_Discharge']
```

Insight: The day_Discharge column enables day-over-day comparisons of discharges, helping to find trends in the dataset.

Creating Duration of Stay in Hospital

```

-- Add Duration column
df['Duration']=(df['Discharge Date']-df['Date of Admission']).dt.days

-- Overview of the new column
df['Duration']
```

Insight: The Duration column, obtained by subtracting the Date of Admission from the Date of Discharge, provides valuable insights into patient stays. By analyzing the duration of hospital stays, we can identify trends in patient recovery times, detect anomalies such as unusually short or long stays, and assess hospital resource utilization over time.

Creating Age Group

```

-- Add Age group column
-- here we used cut() in pandas to create this column
age_labels=['0-18','19-35','36-50','51-65','66-100']
df['Age group']=pd.cut(df['Age'],bins = 5,labels=age_labels)

-- Overview of the new column
df['Age group']
```

Insight: The newly created Age group column categorizes patients into different age brackets, ranging from 0-18 to 66-100. This classification allows us to analyze trends and patterns in hospital visits across various age groups. By studying the distribution of patients across these age categories, we can gain insights into which age groups are most frequently admitted, identify potential healthcare needs based on age, and better understand the demographics of hospital visits for targeted healthcare strategies.

Data Visualization

1. Outlier Check using Boxplot

```
1
2  -- import matplotlib library
3  import matplotlib.pyplot as plt
4
5  -- Drawing boxplot of all numerical columns
6  for i in df:
7    if df[i].dtype == 'int' or df[i].dtype == 'float':
8      plt.figure(figsize=(5,3))
9      plt.boxplot(df[i])
10     plt.title(f"Boxplot of {i}")
11     plt.show()
```

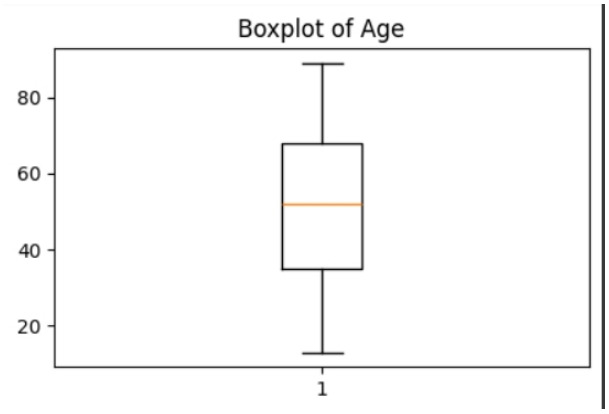

Figure 1: Boxplot of Age

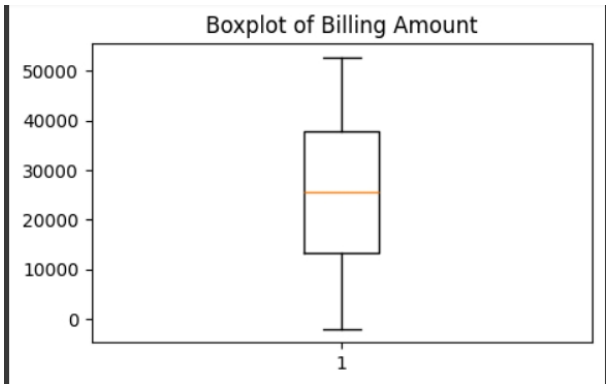

Figure 2: Boxplot of Billing Amount

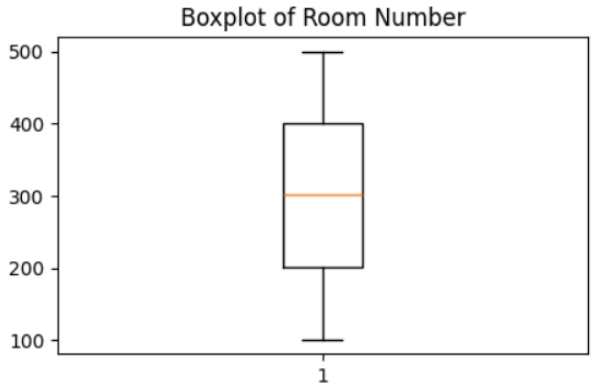

Figure 3: Boxplot of Room Number

Insight: The box plot clearly shows that the dataset doesn't contain any outliers.

2. Histogram of numerical columns

```
1
2  -- Drawing histogram of numerical columns
3  for i in df:
4    if df[i].dtype == 'int' or df[i].dtype =='float':
5      plt.figure(figsize=(5,3))
6      plt.hist(x=df[i])
7      plt.xlabel(i)
8      plt.ylabel("count")
9      plt.xticks(rotation=90)
10     plt.title(f"Histogram of {i}")
11     plt.show()
```

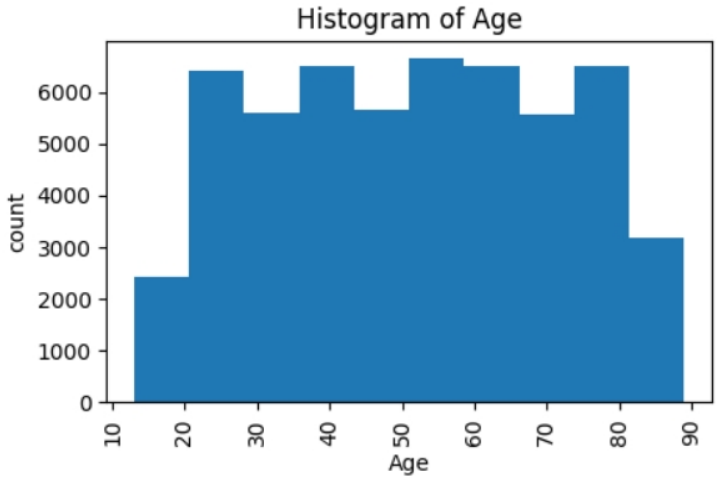

Figure 4: Histogram of Age

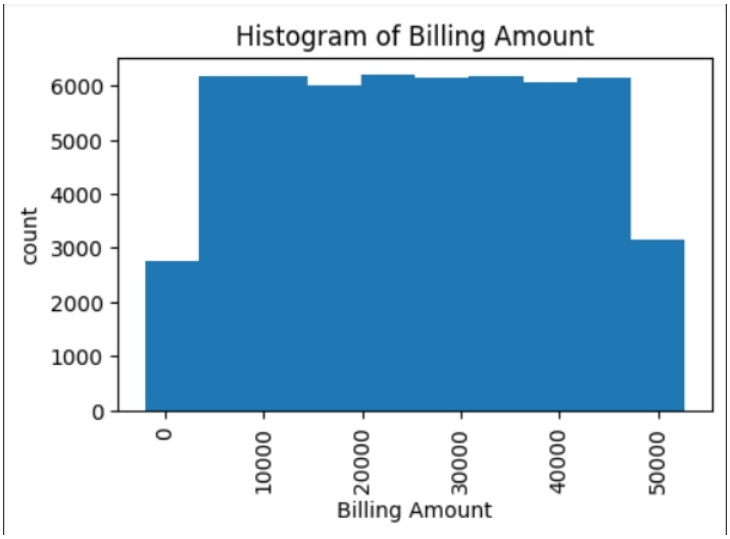

Figure 5: Histogram of Billing Amount

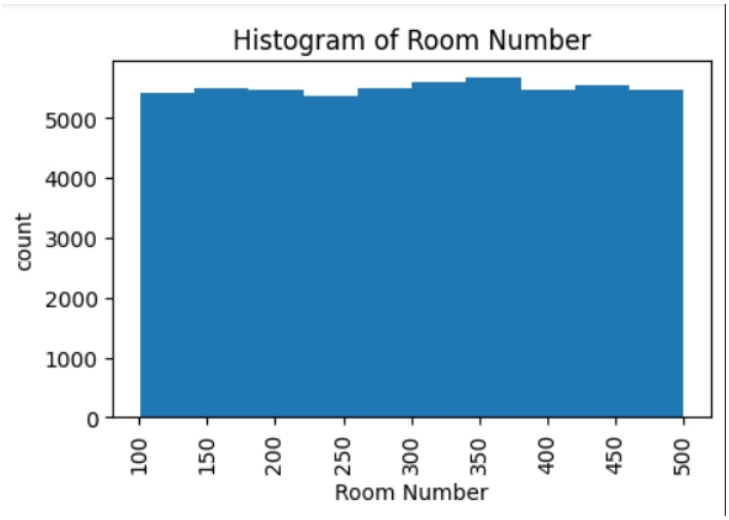

Figure 6: Histogram of Room number

Insight: The histogram shows that the largest count of patients falls within the 50-60 age range. The billing amounts are fairly evenly distributed between 5,000 and 45,000, and the room numbers are evenly distributed between 100 and 500.

2. Countplot of Categorical columns

```
-- import seaborn library
import seaborn as sns

-- Drawing countplot of categorical columns
for i in df:
  if df[i].dtype == 'object':
    if df[i].nunique() <= 10:
      plt.figure(figsize=(5,3))
      sns.countplot(df[i])
```

```
        plt.title(f"Bar plot of {i}")
        plt.xticks(rotation=90)
        plt.show()
```

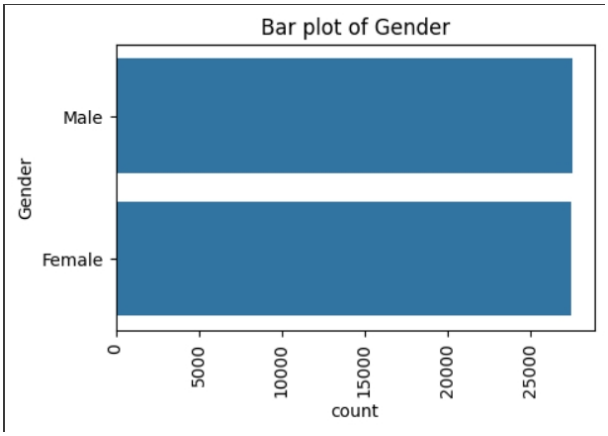

Figure 7: Countplot of Gender

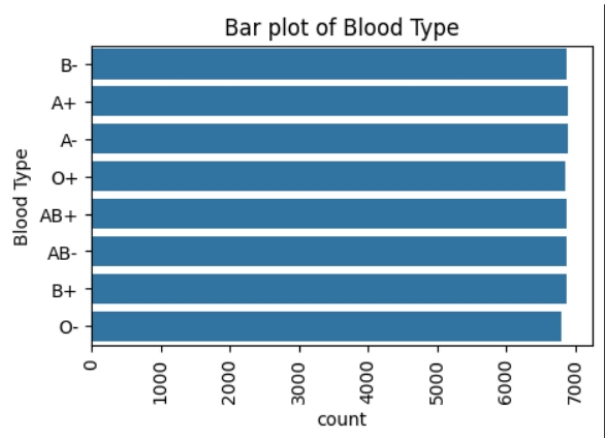

Figure 8: Countplot of Blood Type

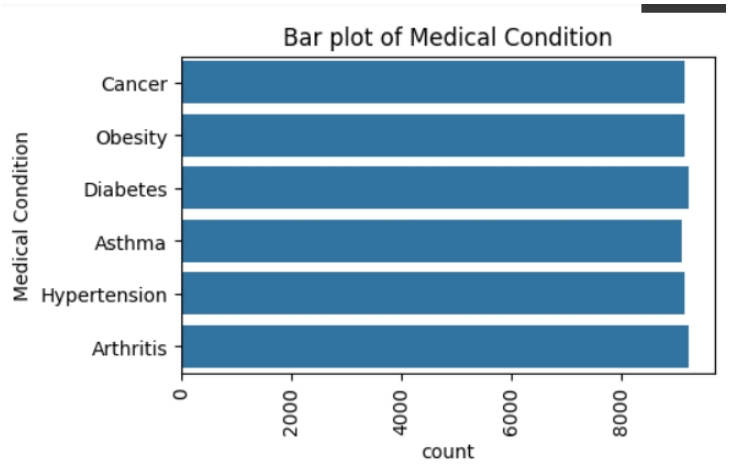

Figure 9: Countplot of Medical Condition

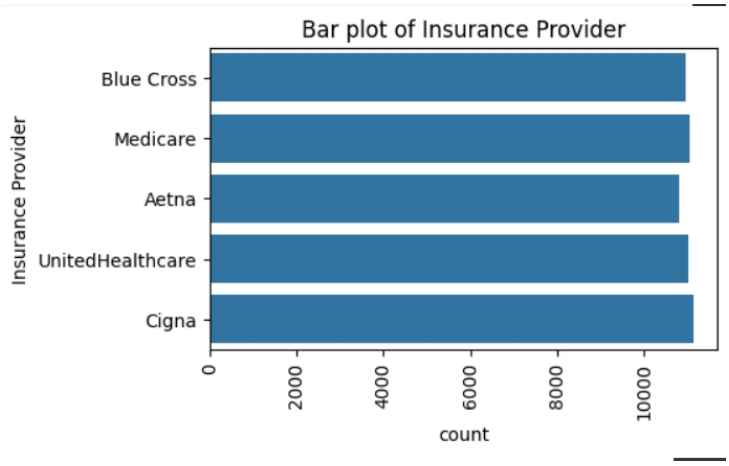

Figure 10: Countplot of Insurance Provider

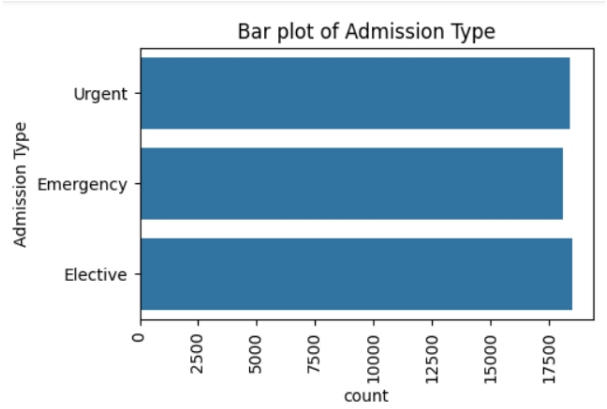

Figure 11: Countplot of Admission Type

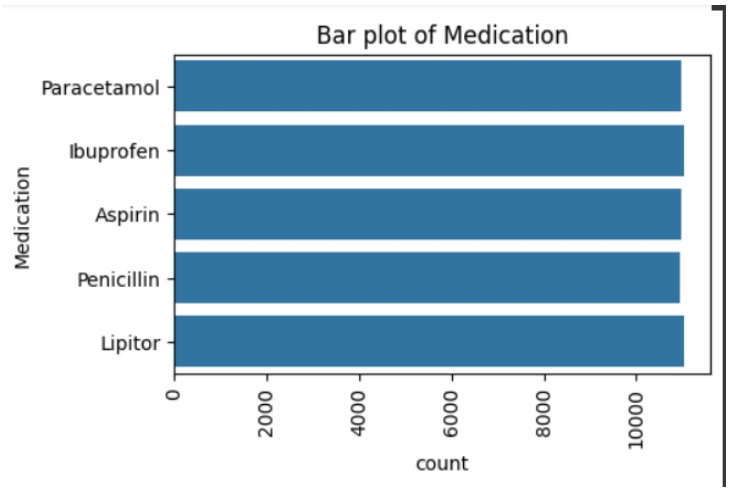

Figure 12: Countplot of Medication

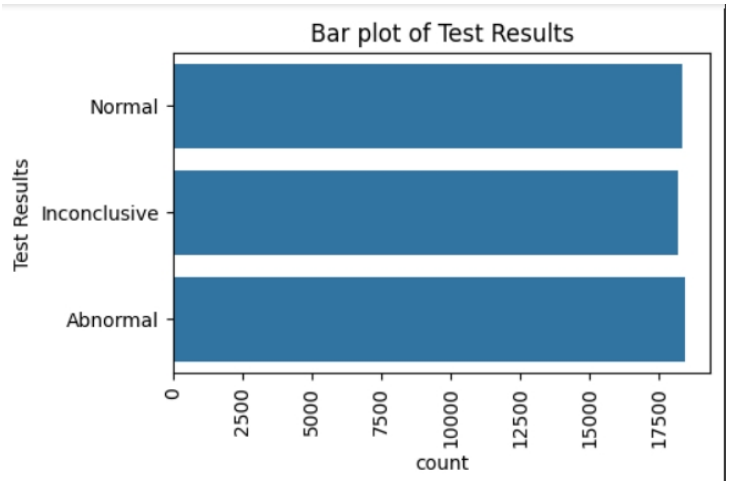

Figure 13: Countplot of Test Results

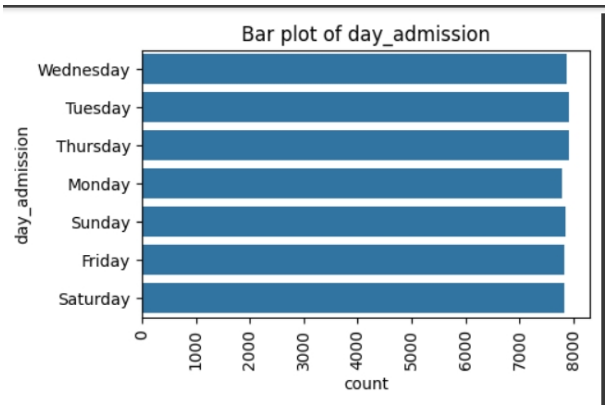

Figure 14: Countplot of Day Of Admission

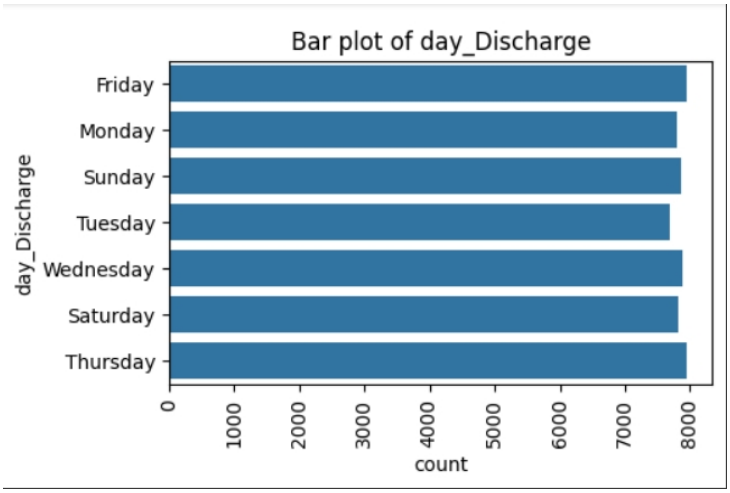

Figure 15: Countplot of Day Of Discharge

Insight: The count plot shows that the largest number of patients falls within the Male category. Blood groups A- and A+ are the most prevalent among patients. The

medical conditions most common among patients are Arthritis and Diabetes. The insurance providers with the highest coverage are Cigna and Medicare. The most frequent admission types are Elective and Urgent. Lipitor and Ibuprofen are the most commonly prescribed medications. The most common test result is 'Abnormal,' followed by 'Normal.' Tuesday has the highest number of admissions, while Friday has the most discharges.

3. Heatmap of numerical columns

```

-- import seaborn library
import seaborn as sns

-- Drawing heatmap of numerical columns
v1 = df.corr(numeric_only= True)
sns.heatmap(v1,annot=True)
```

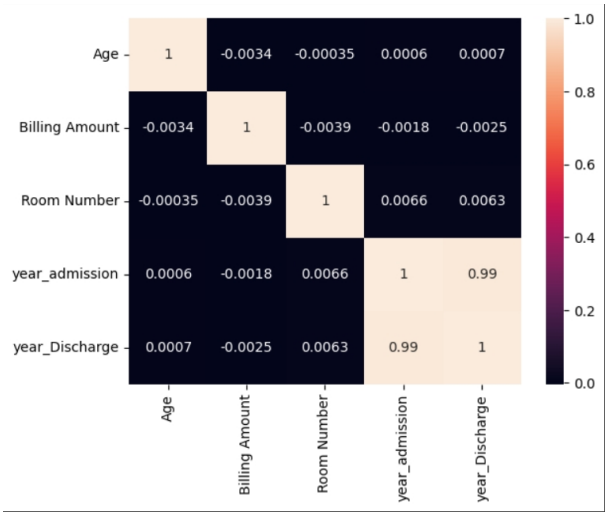

Figure 16: Heatmap

Insight: The heatmap shows that all the numerical columns are negatively correlated with each other, meaning that as one variable increases, the others tend to decrease. This indicates a strong dependence between the variables but in an inverse relationship, not independence.

Univariate Analysis

The following analyses were conducted on the cleaned dataset:

1. Number of Unique Patients

```
# Total number of rows (total records in the dataset)
total_rows = df['Name'].count()

# Number of unique patients
```

```
unique_patients = df['Name'].nunique()

# Data for plotting (labels and corresponding values)
labels = ['Total Rows', 'Unique Patients']
values = [total_rows, unique_patients]

# Plotting a bar chart
plt.bar(labels, values, color=['#66b3ff', '#99ff99'])

# Title and labels
plt.title('Comparison of Total Rows and Unique Patients')
plt.ylabel('Count')
plt.show()
```

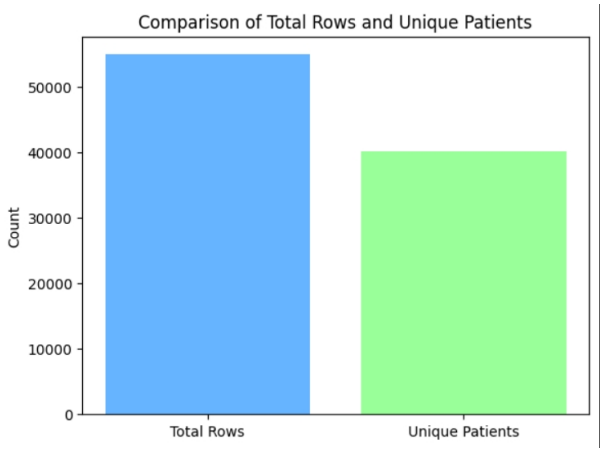

Figure 17: Comparison of Total rows and Unique Patients

Insight:40,235 is the number of unique patients in the dataset.

1.1 Patients attending hospitals most often.

```
top_10_patients = df['Name'].value_counts().nlargest(10)

# Plotting the results
plt.figure(figsize=(10, 5))
sns.barplot(x=top_10_patients.index, y=top_10_patients.values, palette='viridis')

# Adding title and labels
plt.title('Top 10 Patients Attending Most Often')
plt.xlabel('Patient Name')
plt.ylabel('Number of Visits')
plt.xticks(rotation=45, ha='right')  # Rotate x labels for better readability
plt.show()
```

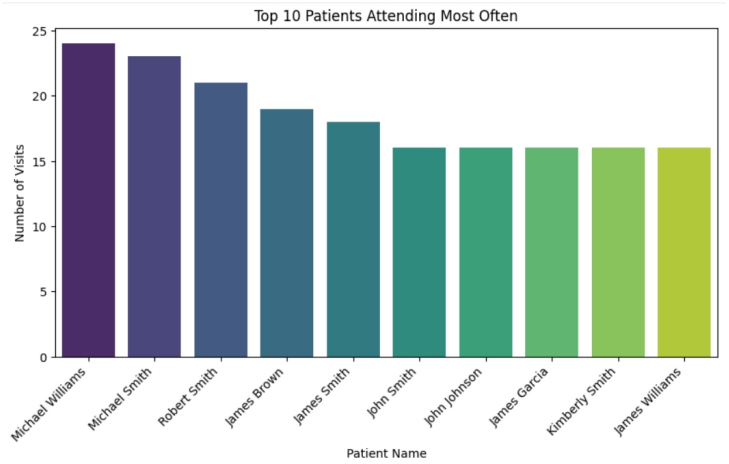

Figure 18: Top 10 Patients attending most often

Insight: Michael Williams and Michael Smith are the patients who visit the hospital most often, with 24 and 23 visits, respectively.

1.2 Patients attending hospitals less often.

```
top_10_patients = df['Name'].value_counts().nsmallest(10)

# Plotting the results
plt.figure(figsize=(10, 5))
sns.barplot(x=top_10_patients.index, y=top_10_patients.values, palette='viridis')

# Adding title and labels
plt.title('Top 10 Patients Attending Most Often')
plt.xlabel('Patient Name')
plt.ylabel('Number of Visits')
plt.xticks(rotation=45, ha='right')  # Rotate x labels for better readability
plt.show()
```

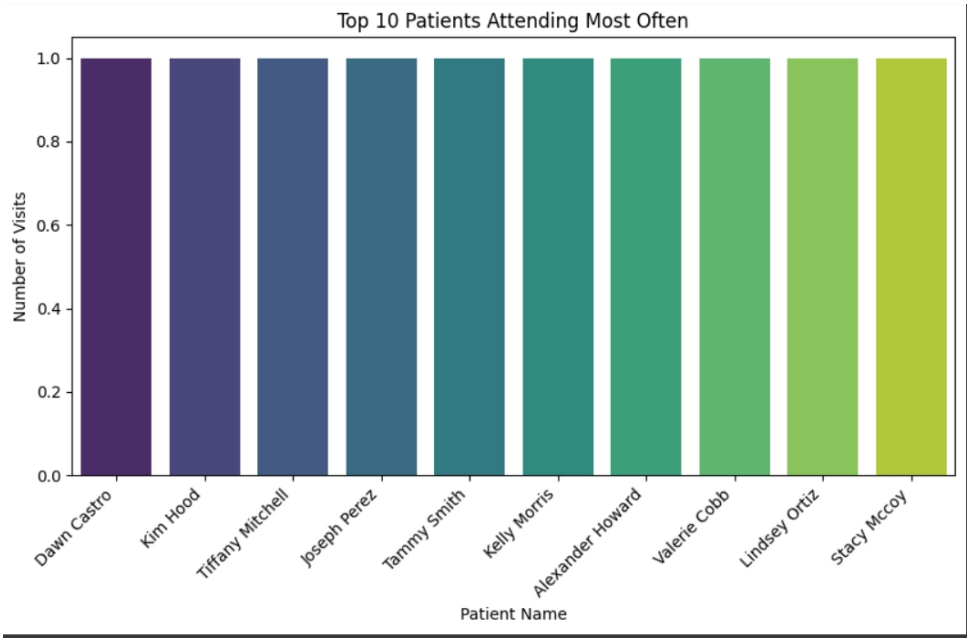

Figure 19: Top 10 Patients attending less often

Insight: The bar plot shows that most patients visit the hospital only once, with names like Dawn Castro, Kim Hood, and others appearing as one-time visitors.

2. Average Age of Patients.

```
# Calculate the average age of patients
average_age = df['Age'].mean()

# Plotting the average age
plt.figure(figsize=(6, 4))
plt.bar(['Average Age'], [average_age], color='skyblue')

# Adding title and labels
plt.title('Average Age of Patients')
plt.ylabel('Age')
plt.ylim(0, max(average_age + 10, 100))  # Adjust the y-axis limit for
better visualization
plt.show()
```

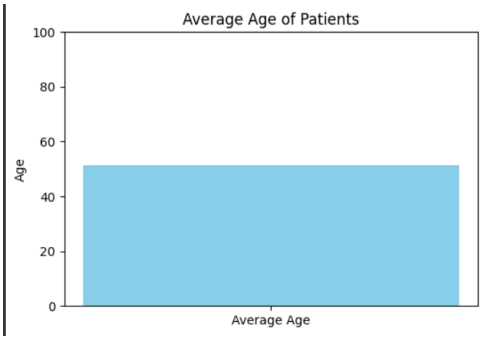

Figure 20: Average age of patients

Insight: The average age of patients is 51.5.

2.1. Common Age, Maximum Age, Minimum age of Patients.

```
common_age = df['Age'].mode()[0]
max_age = df['Age'].max()
min_age = df['Age'].min()

# Data for plotting
ages = ['Common Age', 'Maximum Age', 'Minimum Age']
values = [common_age, max_age, min_age]

# Plotting the values as a bar chart
plt.figure(figsize=(8, 5))
plt.bar(ages, values, color=['skyblue', 'salmon', 'lightgreen'])

# Adding title and labels
plt.title('Common, Maximum, and Minimum Age of Patients')
plt.ylabel('Age')
plt.xlabel('Age Category')

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

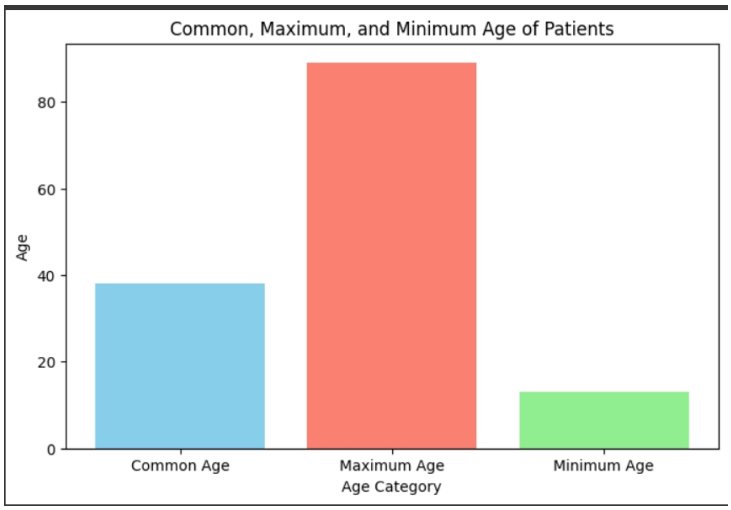

Figure 21: Common Age, Maximum Age, Minimum age of Patients.

Insight: "The common age of patients is 38, the maximum age is 89, and the minimum age is 13.

3. Gender Distribution.

```
# Calculate the gender distribution
```

```
gender_counts = df['Gender'].value_counts()

# Plotting the pie chart
plt.figure(figsize=(7, 7))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.2f%%'
, colors=['lightblue', 'salmon'], startangle=90)

# Adding a title
plt.title('Gender Distribution of Patients')

# Display the pie chart
plt.show()
```

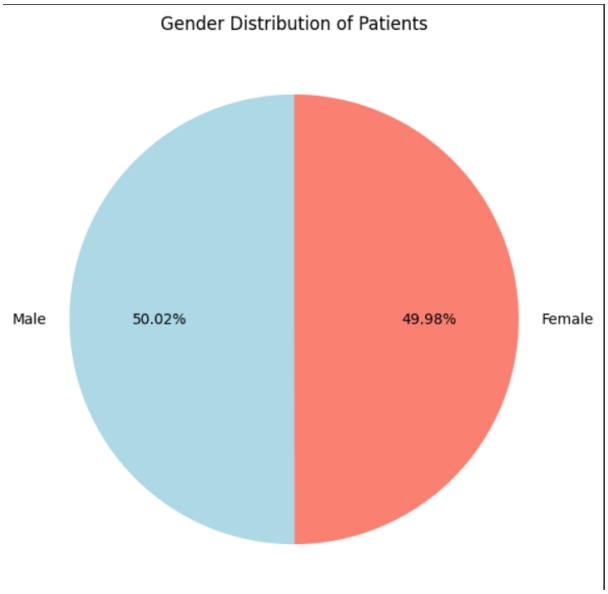

Figure 22: Gender Distribution.

Insight: The gender distribution is nearly equal, with the male count exceeding the female count by just 26.

4. Most Common Blood Type of Patients.

```
# Calculate the frequency of each blood type
blood_type_counts = df['Blood Type'].value_counts()

# Create a dot plot using seaborn
plt.figure(figsize=(10, 6))
sns.stripplot(x=blood_type_counts.index, y=blood_type_counts.values,
              jitter=True, size=10, color='dodgerblue', edgecolor='black')

# Add title and labels
plt.title('Most Common Blood Type of Patients', fontsize=16)
plt.xlabel('Blood Type', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
```

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

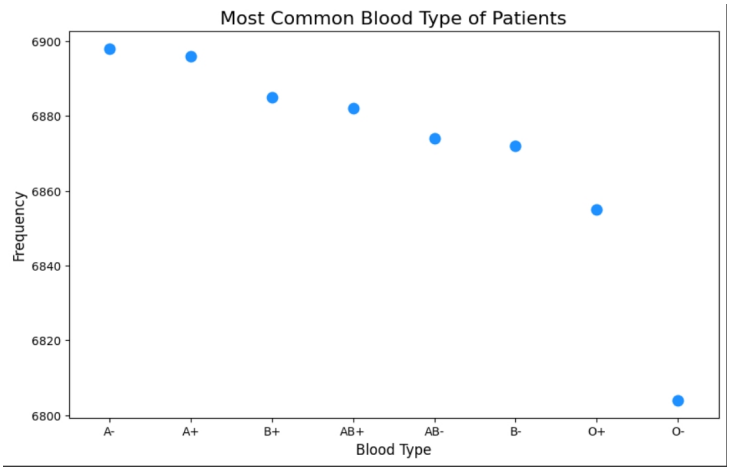

Figure 23: Most Common Blood Type of Patients.

Insight: The analysis reveals that A- (6898) and A+ (6896) are the most common blood types among patients, while O- (6804) and O+ (6855) are the least common.

5. Most Frequent Medical Condition.

```
from wordcloud import WordCloud

# Get the most frequent medical conditions (top 6)
top_conditions = df['Medical Condition'].value_counts().nlargest(6)

# Create a word cloud from the most frequent medical conditions
wordcloud = WordCloud(width=800, height=400, background_color='white')
.generate_from_frequencies(top_conditions)

# Plot the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')  # Turn off the axis
plt.title('Most Frequent Medical Conditions', fontsize=16)
plt.show()
```

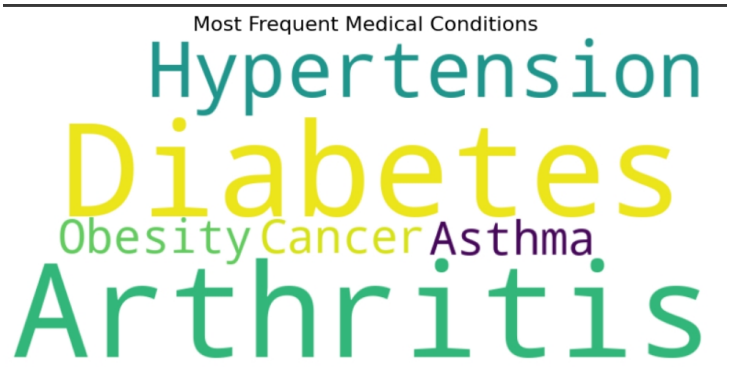

Figure 24: Most Frequent Medical Condition.

Insight: It is clear that Arthritis (9218) and Diabetes (9216) are the most common medical conditions among patients visiting the hospital, while Asthma (9095) and Cancer (9140) have the lowest number of patients.

6. Most Admission types of Patients.

```
# Calculate the top 3 admission types
top_admission_types = df['Admission Type'].value_counts().nlargest(3)

# Create a horizontal bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=top_admission_types.values, y=top_admission_types.index
, palette='viridis')

# Add title and labels
plt.title('Top 3 Admission Types', fontsize=16)
plt.xlabel('Number of Patients', fontsize=12)
plt.ylabel('Admission Type', fontsize=12)

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

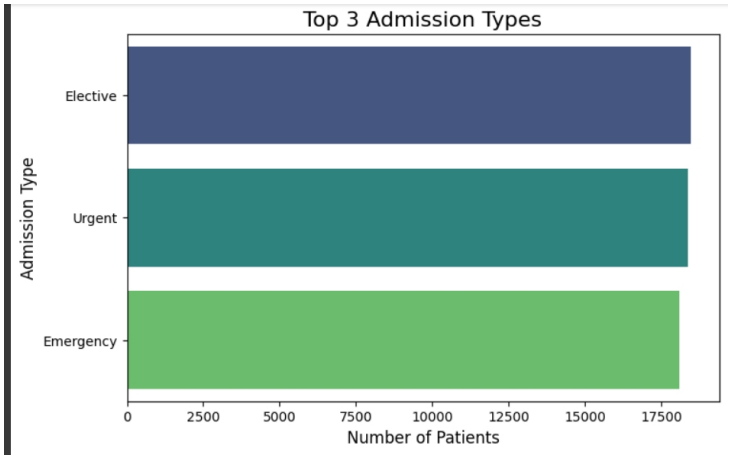

Figure 25: Most Admission types of patients.

Insight: It is clear that Elective (18,473) and Diabetes (18,391) are the most common admission types among patients visiting the hospital, while Emergency admissions have the lowest number of patients.

7. Average, Maximum Billing Amount.

```
import matplotlib.patches as patches

# Data: Average and maximum billing amounts
avg_billing = df['Billing Amount'].mean()
max_billing = df['Billing Amount'].max()

# Create a figure
fig, ax = plt.subplots(figsize=(8, 3))

# Background bar for the maximum billing value
ax.barh(0, max_billing, color='lightgrey', height=0.5)

# Create a bar for the average billing value
ax.barh(0, avg_billing, color='dodgerblue', height=0.5)

# Add text for clarity
ax.text(avg_billing + 500, 0, f'Avg: ${avg_billing:,.2f}', va='center', fontsize=12
, color='blue')
ax.text(max_billing + 500, 0, f'Max: ${max_billing:,.2f}', va='center', fontsize=12
, color='red')

# Formatting
ax.set_xlim(0, max_billing * 1.1)
ax.set_yticks([])  # Hide y-axis ticks
ax.set_xlabel('Billing Amount ($)', fontsize=12)
ax.set_title('Bullet Chart: Average vs Maximum Billing Amount', fontsize=16)

plt.show()
```

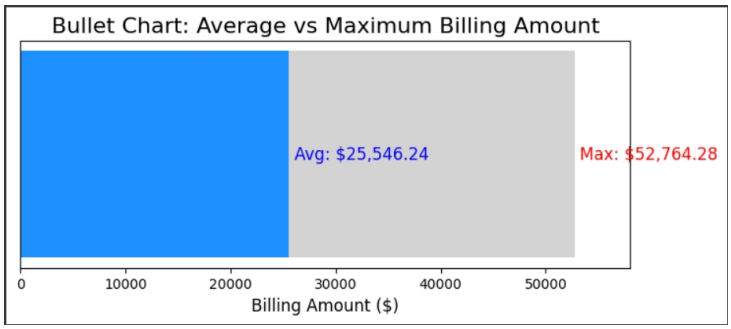

Figure 26: Average, Maximum Billing Amount.

Insight: The average billing amount per patient is $25,544, while the maximum billing amount per patient is $52,764.

7.1. Most Frequent Medical Condition.

```
-- Find the minimum amount
df['Billing Amount'].min()

-- -$2,008.49 (potential data entry error or credit).is the lowest billing amount
 -- for a single patient
df[df['Billing Amount'] < 0].sort_values(by='Billing Amount',ascending=False)


-- In 106 rows it is clear that bill amount has negative sign so we have to remove
 -- negative sign otherwise it will remain as outliers

 df['Billing Amount'] = df['Billing Amount'].astype(str)

# Use regex to replace the negative sign
df['Billing Amount'] = df['Billing Amount'].replace('^-', '', regex=True)

# Convert 'BillingAmount' back to a numeric type
df['Billing Amount'] = pd.to_numeric(df['Billing Amount'])

# check if any rows are left
df[df['Billing Amount'] < 0].sort_values(by='Billing Amount',ascending=False)

#now no outliers

# now find minimum billing amount
df['Billing Amount'].min()


# Calculate the minimum billing amount
```

```
min_billing = df['Billing Amount'].min()

# Data for plotting
billing_types = ['Minimum Billing Amount']
billing_values = [min_billing]

# Create a horizontal bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=billing_values, y=billing_types, palette='coolwarm')

# Add title and labels
plt.title('Minimum Billing Amount per Patient', fontsize=16)
plt.xlabel('Billing Amount ($)', fontsize=12)
plt.ylabel('Billing Type', fontsize=12)

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

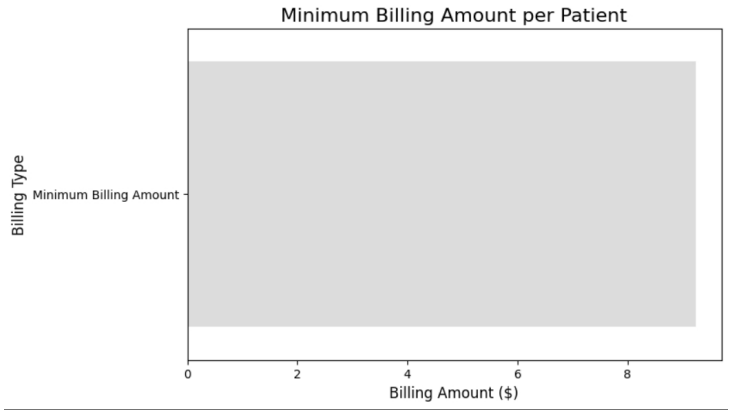

Figure 27: Minimum Billing Amount.

Insight: Initially, when the minimum billing amount was checked, it displayed a negative value due to an error. After removing the negative sign from the 'Billing Amount' column, the correct minimum value was obtained. The minimum billing amount per patient is 9.

7.2. Total Billing Amount.

```
# Calculate the total billing amount
total_billing = df['Billing Amount'].sum()

# Data for plotting
billing_types = ['Total Billing Amount']
billing_values = [total_billing]
```

```
# Create a bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=billing_values, y=billing_types, palette='viridis')

# Add title and labels
plt.title('Total Billing Amount', fontsize=16)
plt.xlabel('Billing Amount ($)', fontsize=12)
plt.ylabel('Billing Type', fontsize=12)

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

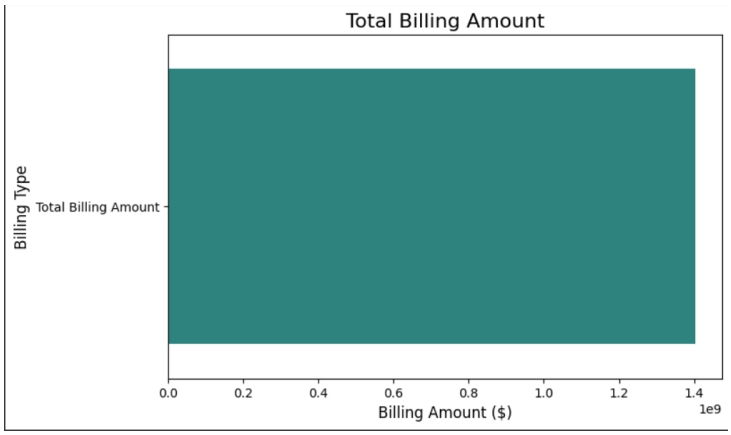

Figure 28: Total Billing Amount.

Insight: The total billing amount across all patients $1,404,068,339

8. Most Common Insurance Provider.

```
# Get the top 5 most common insurance providers
top_5_insurance = df['Insurance Provider'].value_counts().nlargest(5)

# Create a dot plot
plt.figure(figsize=(10, 6))
plt.scatter(top_5_insurance.values, top_5_insurance.index, s=100, color='royalblue'
, edgecolor='black')

# Add title and labels
plt.title('Top 5 Most Common Insurance Providers', fontsize=16)
plt.xlabel('Number of Patients', fontsize=12)
plt.ylabel('Insurance Provider', fontsize=12)

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

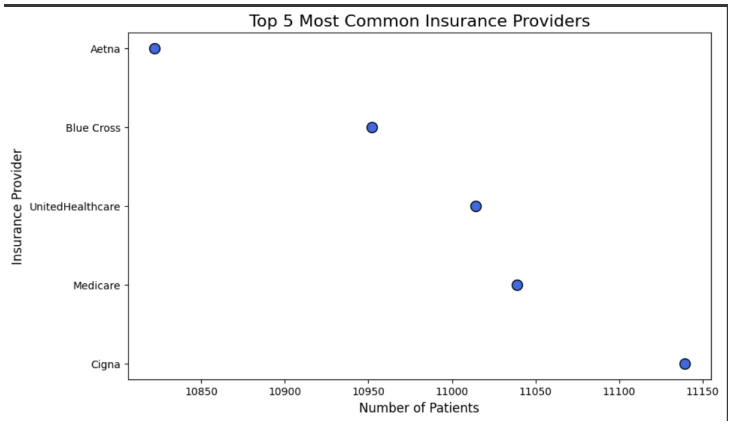

Figure 29: Most Common Insurance Provider.

Insight: It is clear that Cigna (11,139), followed closely by Medicare (11,039), are the most common insurance providers for patients visiting the hospital. Aetna (10,822) has the lowest number of patients among the top providers.

9. Hospital with most number of patients.

```
import numpy as np

# Get the top 5 hospitals by patient count
hospital_counts = df['Hospital'].value_counts().nlargest(5)

# Data for radar chart
labels = hospital_counts.index
values = hospital_counts.values

# Number of variables
num_vars = len(labels)

# Angle for each axis
angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()

# The plot is circular, so we need to repeat the first value at the end
values = np.concatenate((values, [values[0]]))
angles += angles[:1]

# Plotting the radar chart
plt.figure(figsize=(8, 6))
ax = plt.subplot(111, polar=True)

ax.fill(angles, values, color='teal', alpha=0.25)
ax.plot(angles, values, color='teal', linewidth=2)  # Line color
```

```
# Set the labels and title
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(labels, fontsize=12)
plt.title('Top 5 Hospitals by Number of Patients', fontsize=16, color='teal')

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

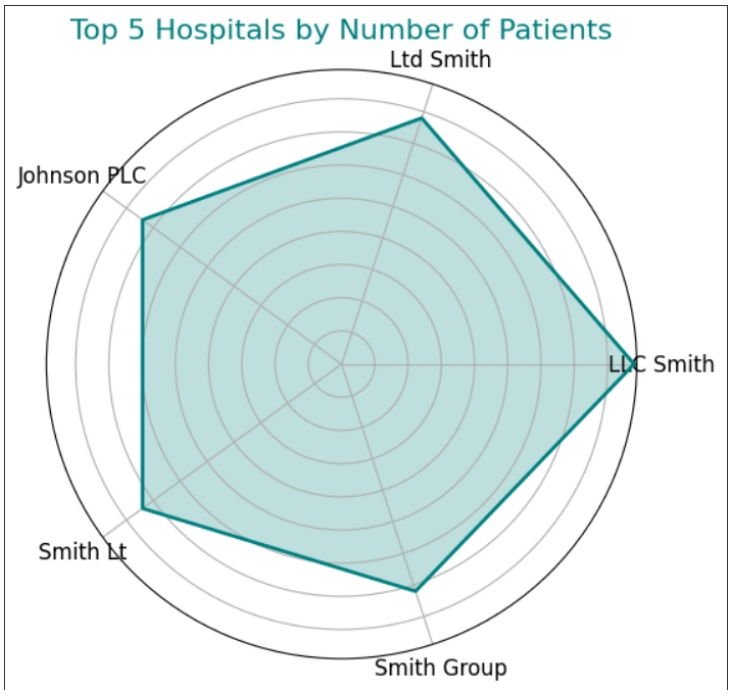

Figure 30: Top 5 Hospital with most number of patients.

Insight:LLC Smith, followed closely by Ltd Smith, are the hospitals with the highest number of patients, with 44 and 39 patients, respectively.

9.1. Hospitals with less number of patients.

```
# Get the bottom 5 hospitals by patient count
hospital_counts_bottom = df['Hospital'].value_counts().nsmallest(5)

# Create a bar plot
plt.figure(figsize=(6, 4))
sns.barplot(x=hospital_counts_bottom.index, y=hospital_counts_bottom.values
, palette='coolwarm')

# Add title and labels
plt.title('Bottom 5 Hospitals by Number of Patients', fontsize=16)
plt.xlabel('Hospital Name', fontsize=12)
```

```
plt.ylabel('Number of Patients', fontsize=12)
plt.xticks(rotation=45, ha='right')

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

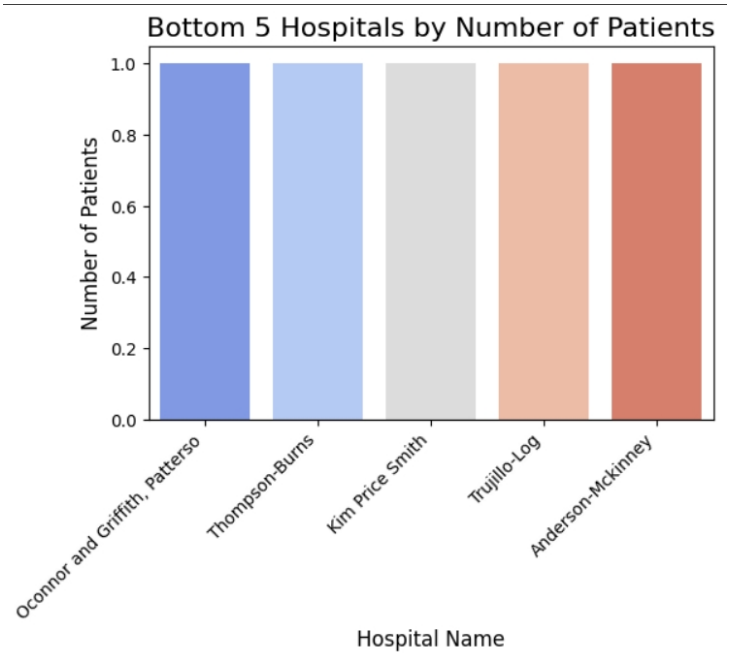

Figure 31: Top 5 Hospital with less number of patients.

Insight: When analyzing hospitals with fewer patients, it was found that some hospital names had leading and trailing commas, which were removed from the column. After cleaning the data, it was observed that all hospitals had a patient count of 1.

9.2. Unique Hospitals.

```
# Total number of rows (total records in the dataset)
total_rows = df['Name'].count()

# Number of unique hospitals
unique_hospitals = df['Hospital'].nunique()

# Data for plotting (labels and corresponding values)
labels = ['Total Rows', 'Unique Hospitals']
values = [total_rows, unique_hospitals]

# Plotting a bar chart
plt.bar(labels, values, color=['#66b3ff', '#99ff99'])

# Title and labels
plt.title('Comparison of Total Rows and Unique Hospital')
```

```
plt.ylabel('Count')
plt.show()
```

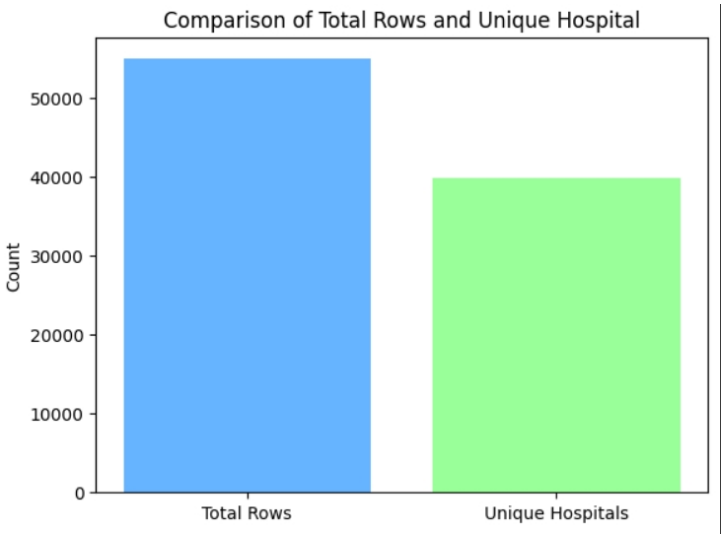

Figure 32: Unique Hospitals.

Insight: 39,870 is the unique hospitals in the dataset.

10 Doctor attending most number of patients.

```
# Get the top 10 doctors by patient count
top_doctors = df['Doctor'].value_counts().nlargest(10)

# Create a bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=top_doctors.index, y=top_doctors.values, palette='viridis')

# Add title and labels
plt.title('Top 10 Doctors by Number of Patients', fontsize=16)
plt.xlabel('Doctor Name', fontsize=12)
plt.ylabel('Number of Patients', fontsize=12)
plt.xticks(rotation=45, ha='right')

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

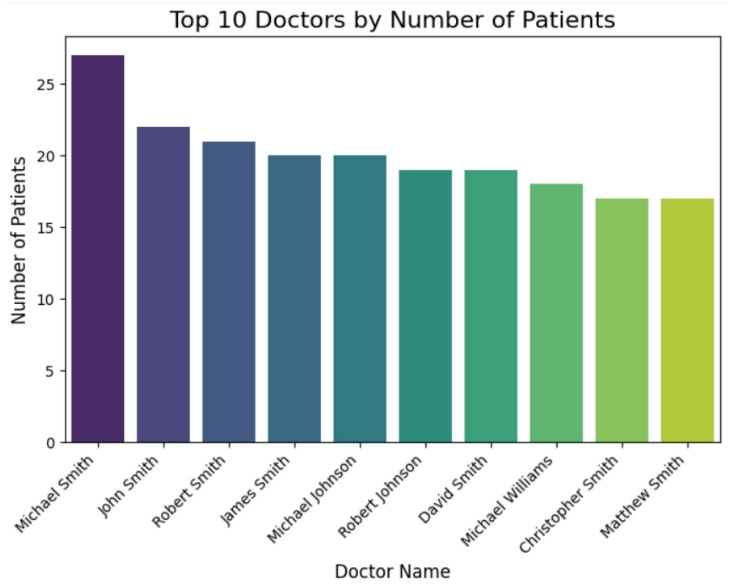

Figure 33: Top 10 doctors with most number of patients.

Insight: Michael Smith (27 patients), followed by John Smith (22 patients), are the doctors who attended to the most number of patients.

10.1.Doctor attending less number of patients.

```
# Get the top 10 doctors by less patient count
top_doctors = df['Doctor'].value_counts().nsmallest(10)

# Create a bar plot
plt.figure(figsize=(8, 5))
sns.barplot(x=top_doctors.index, y=top_doctors.values, palette='viridis')

# Add title and labels
plt.title('Top 10 Doctors by low number of Patients', fontsize=16)
plt.xlabel('Doctor Name', fontsize=12)
plt.ylabel('Number of Patients', fontsize=12)
plt.xticks(rotation=45, ha='right')

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

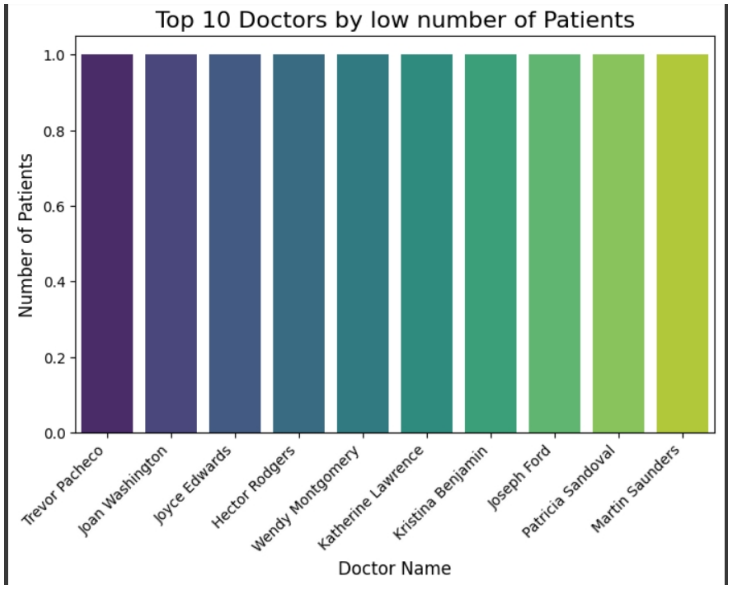

Figure 34: Top 10 doctors with less number of patients.

Insight: It was observed that all Doctors had a patient count of 1.

10.2. Unique Doctors.

```
# Total number of rows (total records in the dataset)
total_rows = df['Name'].count()

# Number of unique hospitals
unique_hospitals = df['Doctor'].nunique()

# Data for plotting (labels and corresponding values)
labels = ['Total Rows', 'Unique Hospitals']
values = [total_rows, unique_hospitals]

# Plotting a bar chart
plt.bar(labels, values, color=['#66b3ff', '#99ff99'])

# Title and labels
plt.title('Comparison of Total Rows and Unique Doctor')
plt.ylabel('Count')
plt.show()
```

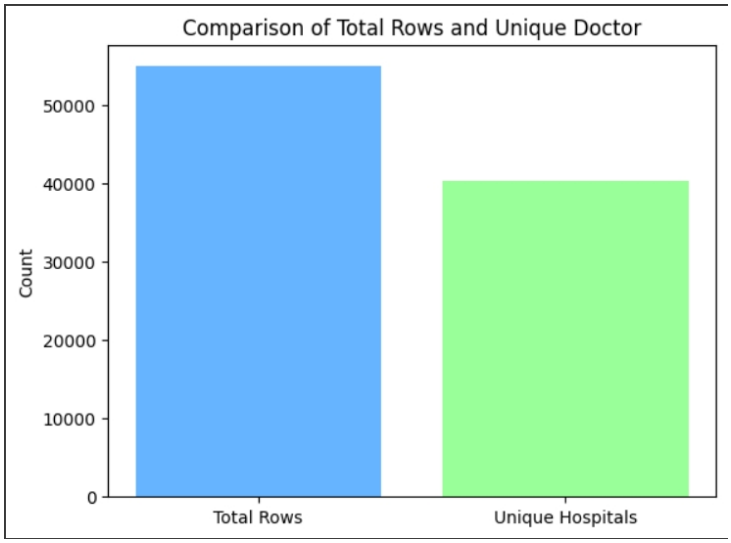

Figure 35: Unique Doctors.

Insight:40,341 is the number of unique doctors in dataset.

11.Most Frequently Prescribed Medication .

```
# Get the most common medications
medication_counts = df['Medication'].value_counts().nlargest(10)
# Get top 10 most frequent medications

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(medication_counts, labels=medication_counts.index, autopct='%1.1f%%'
, startangle=90, colors=plt.cm.Paired.colors)

# Add title
plt.title('Top 5 Most Frequently Prescribed Medications', fontsize=16)

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

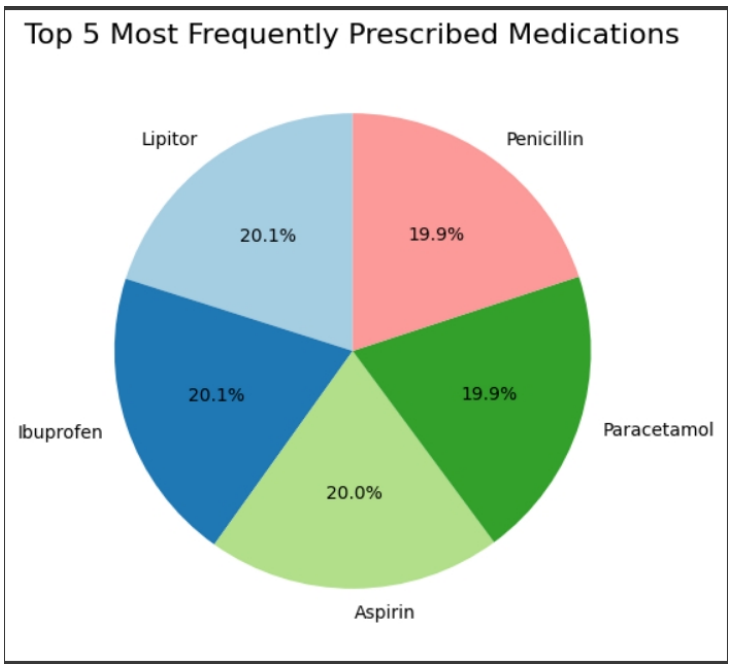

Figure 36: Most Frequently Prescribed Medication .

Insight: Lipitor (11,038 prescriptions) and Ibuprofen (11,023 prescriptions) are the most frequently prescribed medications, while Penicillin (10,956 prescriptions) and Paracetamol (10,965 prescriptions) are the least prescribed medications by doctors.

12.Most Common Test result .

```
# Get the counts of each test result
test_result_counts = df['Test Results'].value_counts()

# Create a bar plot to show the most common test results
plt.figure(figsize=(10, 6))
sns.barplot(x=test_result_counts.index, y=test_result_counts.values
, palette='coolwarm')

# Add title and labels
plt.title('Most Common Test Results Among Patients', fontsize=16)
plt.xlabel('Test Result', fontsize=12)
plt.ylabel('Number of Patients', fontsize=12)

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

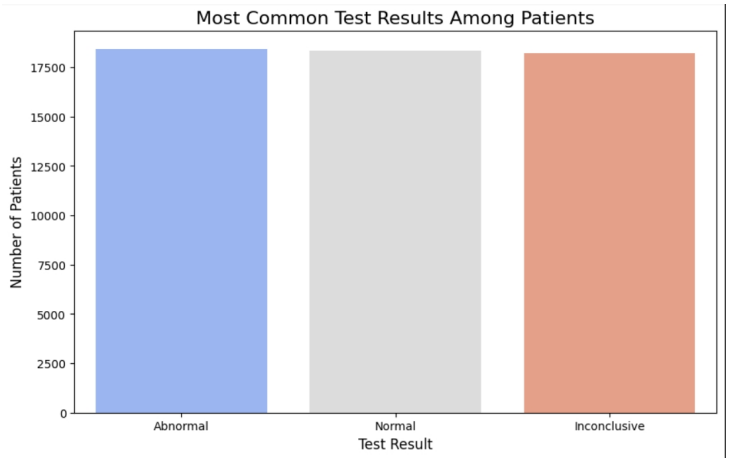

Figure 37: Most Common Test result.

Insight: Abnormal (18,437) and Normal (18,331) are the most common test results among patients, while Inconclusive (18,198) is the test result with the fewest patients.

13.Maximum, Minimum and Average Duration of Stay

```
# Calculate the maximum, minimum, and average duration
max_duration = df['Duration'].max()
min_duration = df['Duration'].min()
avg_duration = df['Duration'].mean()

# Create a list of statistics and values for plotting
statistics = ['Max Duration', 'Min Duration', 'Average Duration']
durations = [max_duration, min_duration, avg_duration]

# Create the bar plot directly from the list of values
plt.figure(figsize=(8, 6))
sns.barplot(x=statistics, y=durations, palette='coolwarm')

# Add title and labels
plt.title('Maximum, Minimum, and Average Duration of Stay', fontsize=16)
plt.xlabel('Statistic', fontsize=12)
plt.ylabel('Duration (days)', fontsize=12)

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

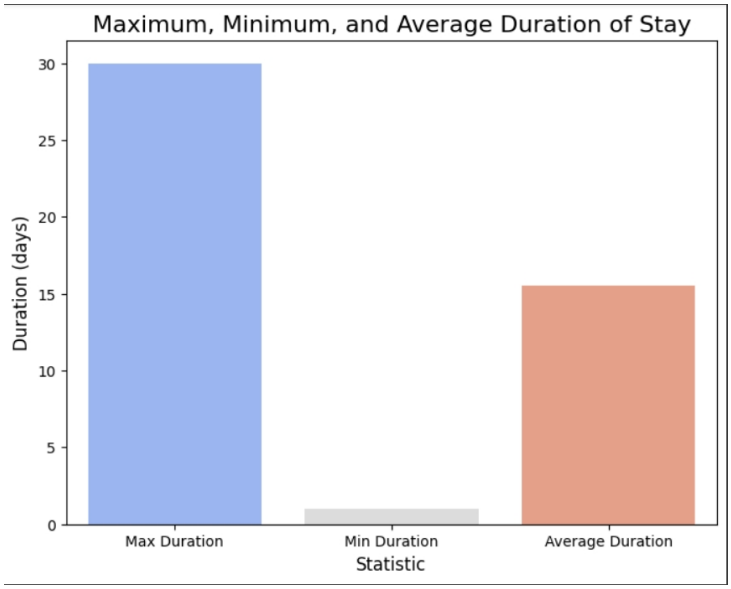

Figure 38: Maximum, Minimum, and Average Duration of Stay

Insight: The average duration of stay for patients is 15.4 days, with the maximum duration being 30 days and the minimum duration recorded at 1 day.

13.1. Patients with longer than average stay in hospital

```
# Calculate the average duration of stay
avg_duration = df['Duration'].mean()

# Filter the dataframe for patients with a longer than average stay
longer_than_avg = df[df['Duration'] > avg_duration]

# Create a bar plot to show the count of patients with a longer-than-average stay
plt.figure(figsize=(10, 6))
sns.countplot(data=longer_than_avg, x='Duration', palette='viridis')

# Add title and labels
plt.title('Patients with Longer-than-Average Duration of Stay', fontsize=16)
plt.xlabel('Duration of Stay (days)', fontsize=12)
plt.ylabel('Number of Patients', fontsize=12)

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

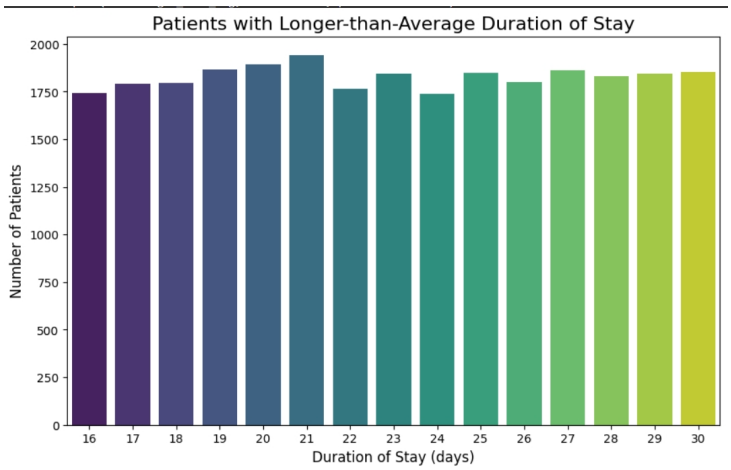

Figure 39: Patients with longer than average stay in hospital.

Insight:A total of 27,430 patients had a duration of stay longer than the average, indicating a significant portion of patients require extended care or treatment.

14.Common Age groups in hospital

```
# Calculate the value counts for the 'Age group' column
age_group_counts = df['Age group'].value_counts()

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(age_group_counts, labels=age_group_counts.index, autopct='%1.1f%%'
,startangle=90, colors=plt.cm.Paired.colors)

# Add title
plt.title('Distribution of Patients Across Age Groups', fontsize=16)

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

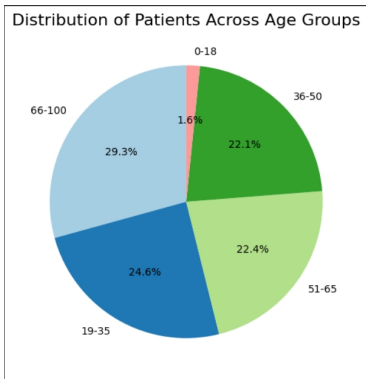

Figure 40: Common age groups in hospital.

Insight: The most common age group among patients is 66-100 (16,096 patients), representing those above 65 years old, which suggests that elderly patients are more likely to be admitted to the hospital. In contrast, the 0-18 age group, with only 886 patients, is the least represented, indicating that hospital visits are significantly less frequent for children and adolescents.

15. Patients under 18 years

```
# Filter the dataframe for patients under 18
under_18 = df[df['Age'] < 18]

# Create a simple count plot to show the number of patients under 18
plt.figure(figsize=(8, 6))
sns.countplot(x=under_18['Age'], palette='coolwarm')

# Add title and labels
plt.title('Number of Patients Under 18 Years Old', fontsize=16)
plt.xlabel('Age (years)', fontsize=12)
plt.ylabel('Number of Patients', fontsize=12)

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

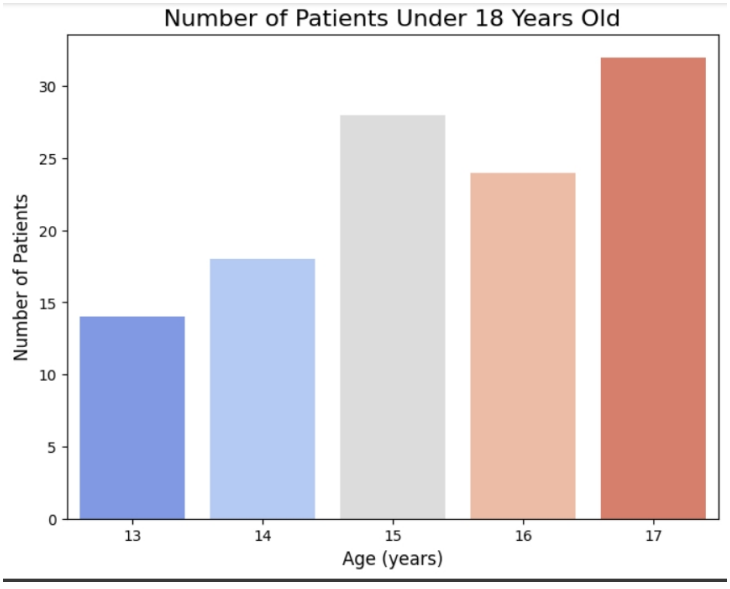

Figure 41: Patients under 18 years.

Insight: There are 116 patients under the age of 18, indicating that pediatric cases represent a small portion of the overall patient population. This suggests that while children and adolescents are less frequently hospitalized, their care might still require specialized services and attention, particularly in areas such as pediatric care and treatment.

15.1. Average, Total Billing Amount of patients under 18 .

```
# Filter the dataframe for patients under 18 directly
under_18 = df[df['Age'] < 18]

# Calculate the total and average billing amount for patients under 18
total_billing_under_18 = under_18['Billing Amount'].sum()
avg_billing_under_18 = under_18['Billing Amount'].mean()

# Create a bar plot for total and average billing amounts
plt.figure(figsize=(8, 6))
sns.barplot(x=['Total Billing', 'Average Billing']
, y=[total_billing_under_18, avg_billing_under_18], palette='coolwarm')

# Add title and labels
plt.title('Total and Average Billing Amount for Patients Under 18 Years Old'
, fontsize=16)
plt.ylabel('Amount (in dollars)', fontsize=12)
plt.xlabel('Statistic', fontsize=12)

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

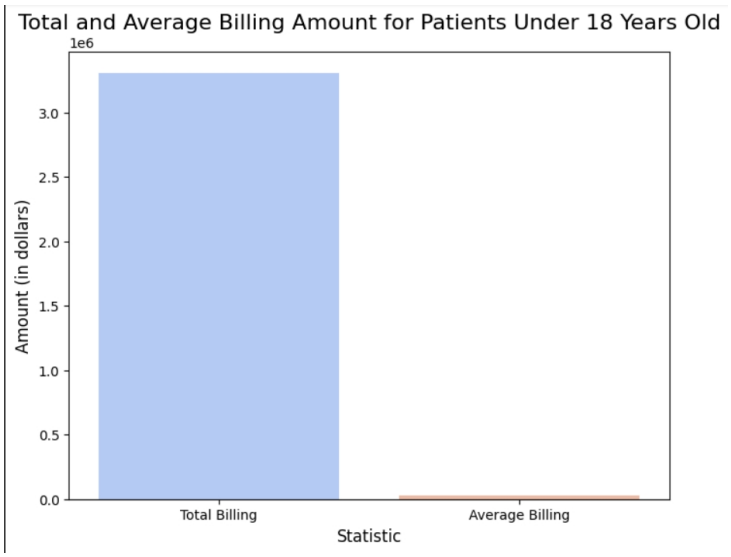

Figure 42: Average, Total Billing Amount of patients under 18.

Insight: The average billing amount for patients under 18 years old is $28,512, while the total billing amount for this group is $3,307,492. This suggests that while there are relatively fewer pediatric patients, the care and treatment they receive tend to be more expensive, possibly due to specialized services, procedures, or longer stays. The higher average billing amount highlights the potential complexity and resource intensity of pediatric care.

16.Most Common Admission Month

```
# Most common admission month
most_common_months = df['month_admission'].value_counts().nlargest(12)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=most_common_months.index, y=most_common_months.values
, palette='viridis')
plt.title('Most Common Admission Months')
plt.xlabel('Month')
plt.ylabel('Number of Admissions')
plt.xticks(rotation=45)
plt.show()
```

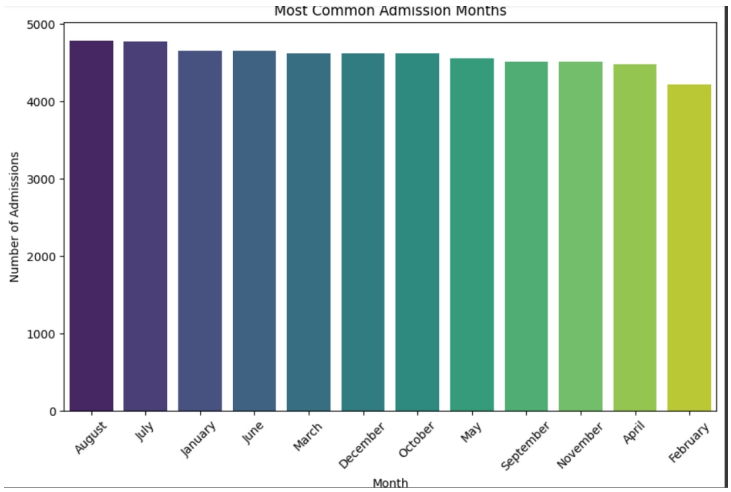

Figure 43: Most Common Admission Month .

Insight: August (4,785 admissions) has the highest number of hospital admissions, followed closely by July (4,765) and January (4,655). This indicates that the summer months and the start of the year see a higher influx of patients, possibly due to seasonal factors, vacations, or increased health issues after holidays. On the other hand, February (4,210) has the fewest admissions, followed by April (4,478) and September (4,508), suggesting that these months may experience relatively lower patient visits, which could be due to milder weather conditions or fewer seasonal health-related issues.

16.1 Most Common Admission Day of The Week

```
# Most common day of the week for admissions
most_common_day = df['day_admission'].value_counts().nlargest(7)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=most_common_day.index, y=most_common_day.values, palette='Blues')
```

```
plt.title('Most Common Day of the Week for Admissions')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Admissions')
plt.xticks(rotation=45)
plt.show()
```

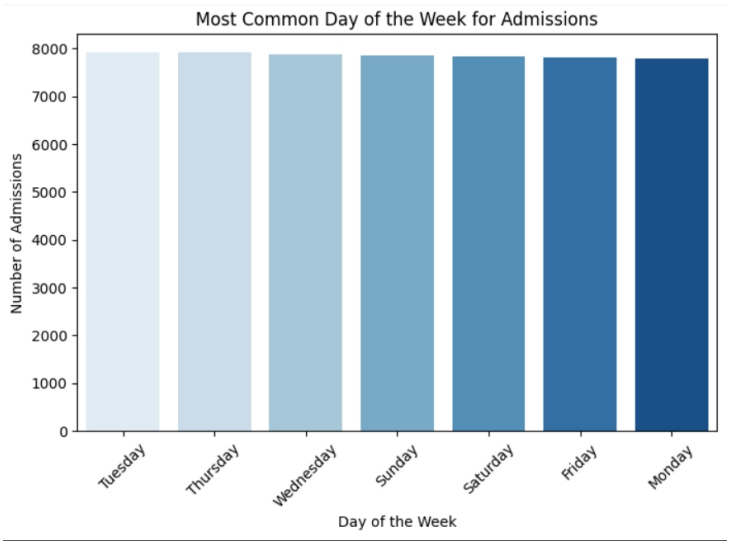

Figure 44: Most Common Admission Day of The Week.

Insight: Tuesday (7,913 admissions) has the highest number of hospital admissions, closely followed by Thursday (7,909) and Wednesday (7,873). This trend suggests that the middle of the week sees a higher patient inflow, which could be due to routine check-ups, scheduled treatments, or a backlog from the weekend. On the other hand, Monday (7,781) has the lowest admissions, followed by Friday (7,818) and Saturday (7,822), which may indicate fewer hospital visits at the start of the week or towards the weekend, possibly due to fewer elective procedures or the weekend effect where people delay visits until mid-week.

16.2.Most Common Admission Day of The Month

```
# Most common day of the month for admissions
most_common_day_of_month = df['Date of Admission'].dt.day.value_counts().nlargest(30)

# Plot
plt.figure(figsize=(12, 6))
sns.barplot(x=most_common_day_of_month.index, y=most_common_day_of_month.values
, palette='RdYlGn')
plt.title('Most Common Day of the Month for Admissions')
plt.xlabel('Day of the Month')
plt.ylabel('Number of Admissions')
plt.xticks(rotation=45)
```

```
plt.show()
```

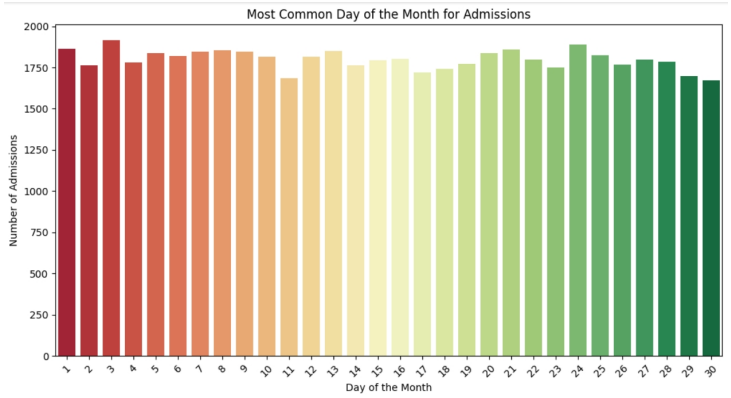

Figure 45: Most Common Admission Day of The Month.

Insight:3rd (1,914 admissions) is the day of the month with the highest number of hospital admissions, followed by 24th (1,887) and 1st (1,862). This could indicate that the start of the month and the middle of the month are typically busier times for hospital visits, possibly due to scheduled treatments, early-month health assessments, or insurance and billing cycles. On the other hand, 31st (1,015) has the lowest admissions, followed by 30th (1,673) and 11th (1,684), which may reflect fewer visits on days toward the end of the month or around certain calendar dates, when fewer elective procedures or routine check-ups occur.

16.3. Most Common Admission Year

```
# Most common year of admission
most_common_year = df['year_admission'].value_counts().nlargest(6)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=most_common_year.index, y=most_common_year.values, palette='coolwarm')
plt.title('Most Common Year of Admission')
plt.xlabel('Year')
plt.ylabel('Number of Admissions')
plt.xticks(rotation=45)
plt.show()
```

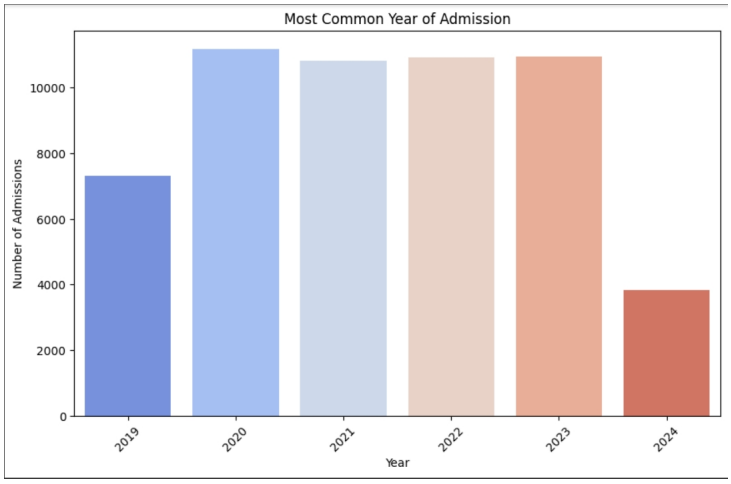

Figure 46: Most Common Admission Year.

Insight:2020 (11,172 admissions) has the highest number of hospital admissions, followed closely by 2023 (10,936) and 2022 (10,915). This indicates a higher volume of patient visits during these years, possibly due to factors like health crises, increased awareness, or hospital operational changes in recent years. In contrast, 2024 (3,827) has seen a significant drop in admissions, which could be due to the early part of the year or less patient activity. 2019 (7,300) and 2021 (10,816) also show lower admissions compared to more recent years, potentially reflecting the post-pandemic adjustments, as well as shifts in patient behavior or healthcare system trends.

17.Most Common Discharge Month

```
# Most common discharge month
most_common_discharge_month = df['month_Discharge'].value_counts().nlargest(12)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=most_common_discharge_month.index
, y=most_common_discharge_month.values, palette='viridis')
plt.title('Most Common Discharge Months')
plt.xlabel('Month')
plt.ylabel('Number of Discharges')
plt.xticks(rotation=45)
plt.show()
```

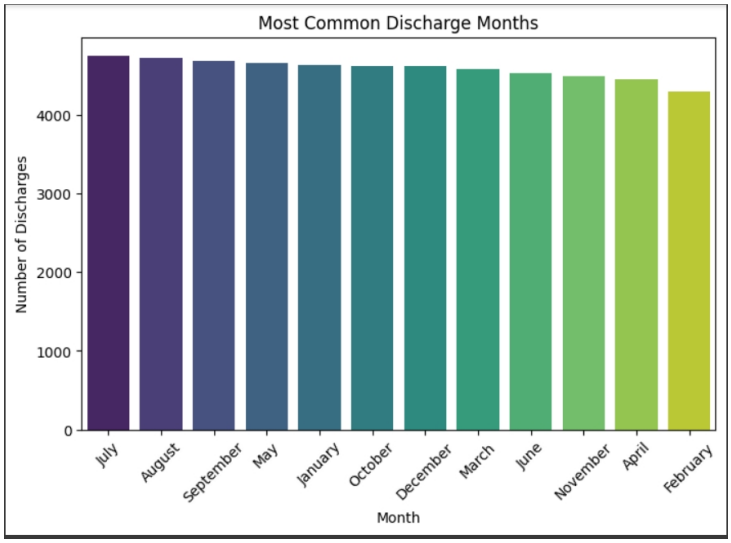

Figure 47: Most Common Discharge Month.

Insight: July (4,746 discharges) has the highest number of hospital discharges, followed by August (4,721) and September (4,677). This suggests that summer months might see a higher volume of patients completing their treatment or recovery, possibly due to scheduled surgeries, seasonal health conditions, or post-holiday recovery patterns. In contrast, February (4,288) has the fewest discharges, followed by April (4,449) and November (4,484), indicating that these months experience a lower rate of discharges, which could be due to factors such as shorter hospital stays, fewer elective procedures, or seasonal healthcare trends.

17.1.Most Common Discharge Day of the Week

```
# Most common day of the week for discharges
most_common_discharge_day = df['day_Discharge'].value_counts().nlargest(7)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=most_common_discharge_day.index, y=most_common_discharge_day.values
, palette='Blues')
plt.title('Most Common Day of the Week for Discharges')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Discharges')
plt.xticks(rotation=45)
plt.show()
```

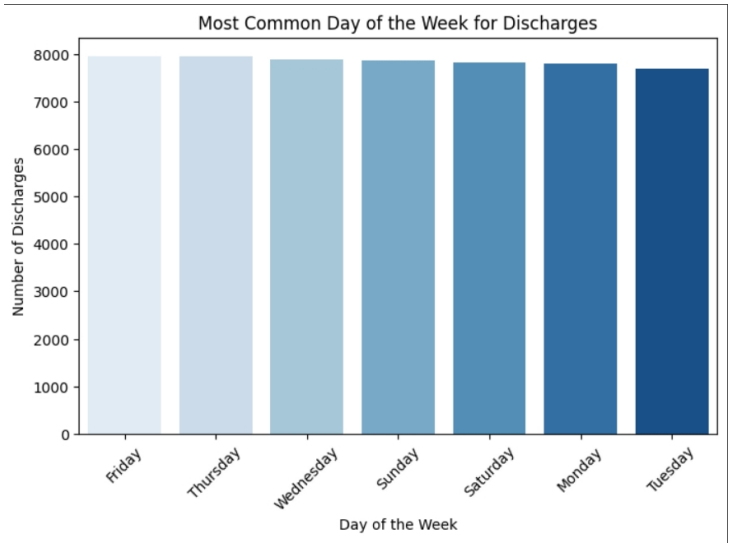

Figure 48: Most Common Discharge Day of the Week.

Insight: Friday (7,947 discharges) has the highest number of hospital discharges, followed closely by Thursday (7,942) and Wednesday (7,893). This indicates that hospitals typically experience a higher volume of discharges towards the end of the week, likely due to patients being discharged before the weekend for scheduled recoveries or planned discharges. In contrast, Tuesday (7,700) has the fewest discharges, followed by Monday (7,807) and Saturday (7,819), which could reflect hospital discharge schedules, with fewer discharges occurring early in the week or on weekends when the hospital staff may be managing fewer elective procedures or scheduled releases.

17.2.Most Common Discharge Day of the Month

```
# Most common day of the month for discharges
most_common_discharge_day_of_month = df['Discharge Date'].dt.day.value_counts()
.nlargest(30)

# Plot
plt.figure(figsize=(12, 6))
sns.barplot(x=most_common_discharge_day_of_month.index
, y=most_common_discharge_day_of_month.values, palette='RdYlGn')
plt.title('Most Common Day of the Month for Discharges')
plt.xlabel('Day of the Month')
plt.ylabel('Number of Discharges')
plt.xticks(rotation=45)
plt.show()
```

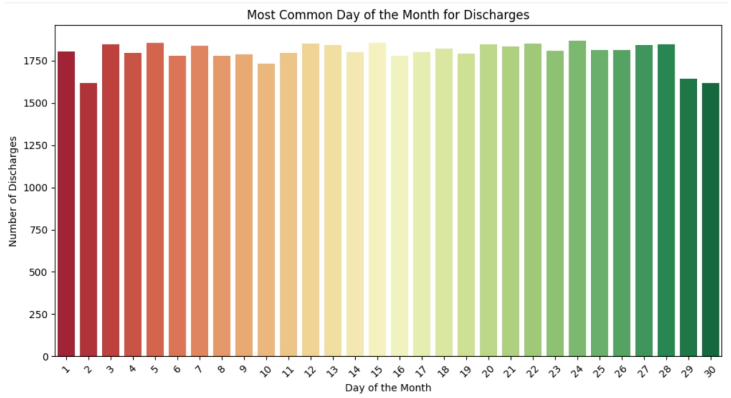

Figure 49: Most Common Discharge Day of the Month.

Insight:24th (1,867 admissions) has the highest number of hospital admissions, followed closely by the 5th (1,855) and 15th (1,854). This suggests that certain days of the month, particularly around the middle and towards the end, see a higher number of admissions, possibly due to scheduled treatments, routine check-ups, or recurring health patterns. On the other hand, 31st (1,012 admissions) has the lowest admissions, likely due to fewer months having a 31st day, followed by 2nd (1,617) and 30th (1,618), which may indicate that these days see fewer patients, possibly because of fewer elective procedures or specific scheduling constraints.

17.3.Most Common Discharge Year

```
# Most common year of discharge
most_common_discharge_year = df['year_Discharge'].value_counts().nlargest(6)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=most_common_discharge_year.index, y=most_common_discharge_year.values
, palette='coolwarm')
plt.title('Most Common Year of Discharge')
plt.xlabel('Year')
plt.ylabel('Number of Discharges')
plt.xticks(rotation=45)
plt.show()
```

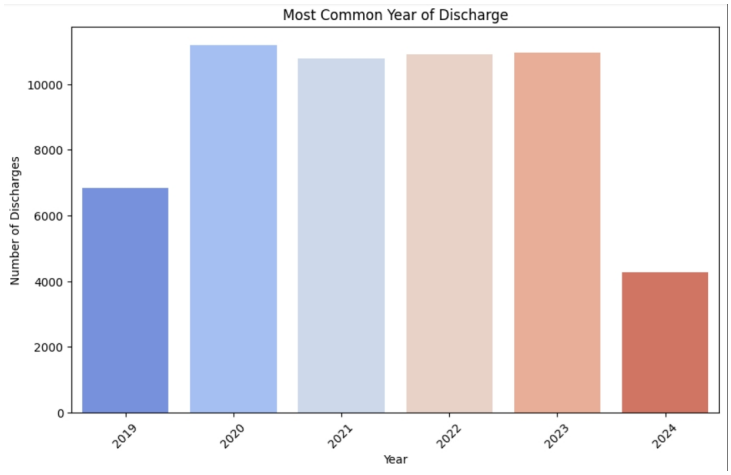

Figure 50: Most Common Discharge Year.

Insight:2020 (11,192 admissions) has the highest number of hospital admissions, followed closely by 2023 (10,969) and 2022 (10,918). These years saw a higher volume of hospital visits, possibly due to factors such as the aftermath of the pandemic, heightened awareness of health conditions, or an increase in healthcare accessibility during this period. In contrast, 2024 (4,272) has the lowest admissions, which could reflect either the early part of the year or changes in hospital admissions as patient behavior shifts. 2019 (6,838) and 2021 (10,777) show relatively fewer admissions, potentially impacted by ongoing recovery from the pandemic and shifts in healthcare priorities.

Bivariate Analysis

The following analyses were conducted on the cleaned dataset:

1. Billing Amount by Admission Type

```
# Grouping by Admission Type and summing Billing Amount
billing_by_admission_type = df.groupby('Admission Type')['Billing Amount'].sum()
.sort_values(ascending=False)

# Creating the line plot
plt.figure(figsize=(8, 5))  # Set the size of the plot
sns.lineplot(x=billing_by_admission_type.index, y=billing_by_admission_type.values
, marker='o', color='b')

# Adding labels and title
plt.title('Total Billing Amount by Admission Type', fontsize=16)
plt.xlabel('Admission Type', fontsize=12)
plt.ylabel('Total Billing Amount', fontsize=12)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

# Display the plot
plt.tight_layout()  # Adjust the plot to avoid overlap
```

```
plt.show()
```

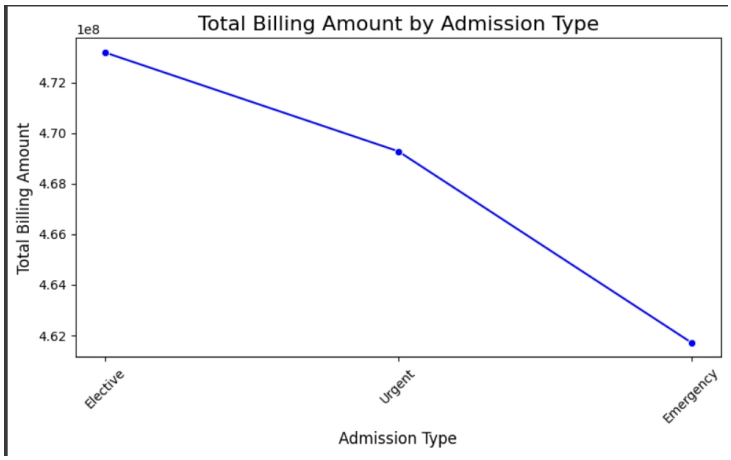

Figure 51: Billing Amount by Admission Type

Insight: Elective admissions have the highest total billing amount, reflecting that planned procedures and non-urgent treatments, which often involve specialized care, tend to incur higher costs. In contrast, Emergency admissions have the lowest total billing amount, which may be due to the nature of emergency care, where treatments are typically shorter, focused on immediate care, and less expensive compared to planned elective procedures.

2. The age group with the highest average billing Amount

```
# Group by 'Age group' and calculate the average Billing Amount
average_billing_by_age_group = df.groupby('Age group')['Billing Amount'].mean()

# Get the Age groups with the highest average billing amount
top_5_age_groups = average_billing_by_age_group.nlargest(5)

# Create a horizontal bar plot with swapped coordinates
plt.figure(figsize=(10, 6))  # Set the size of the plot
sns.barplot(y=top_5_age_groups.index, x=top_5_age_groups.values, palette='viridis')

# Add titles and labels
plt.title(' Age Groups with Highest Average Billing Amount', fontsize=16)
plt.ylabel('Age Group', fontsize=12)
plt.xlabel('Average Billing Amount', fontsize=12)

# Display the plot
plt.tight_layout()  # Adjust layout to avoid overlap
plt.show()
```

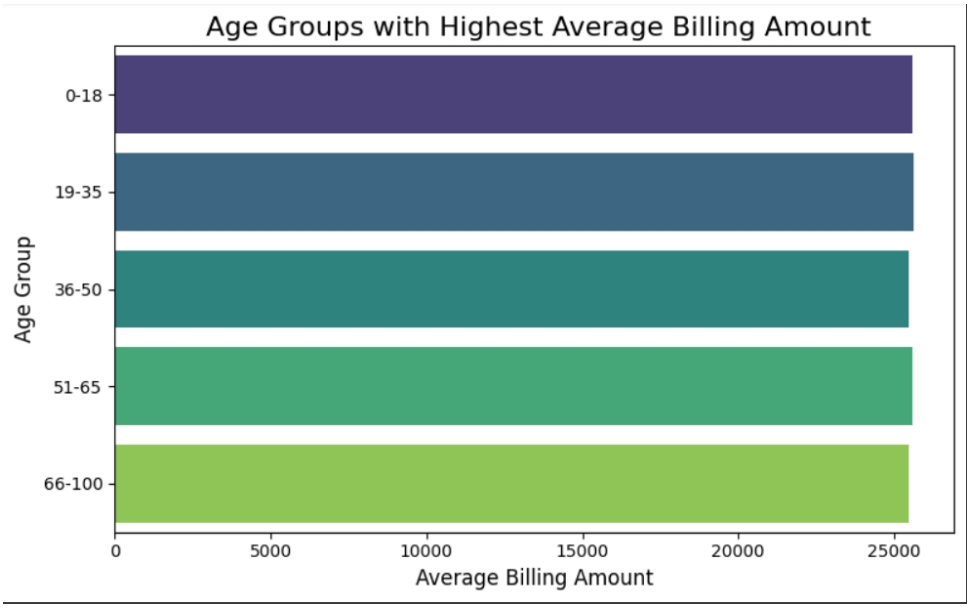

Figure 52: The age group with the highest average billing Amount.

Insight: The 0-18 age group has the highest mean billing amount (26,746), likely due to the high cost of specialized pediatric treatments and surgeries. In contrast, the 66-100 age group has the lowest average billing amount (25,433), possibly reflecting more routine, less intensive care for chronic conditions in older adults. This highlights how complex treatments for younger patients can drive up costs compared to the more stable care often required by older individuals.

2.1. Age group with highest billing amount and Total billing amount

```
# Calculate the highest billing amount for each Age Group
highest_billing_by_age_group = df.groupby('Age group')['Billing Amount'].max()

# Get the top 5 Age groups with the highest billing amount
top_5_highest_billing = highest_billing_by_age_group.nlargest(5)

# Calculate the total billing amount for each Age Group
total_billing_by_age_group = df.groupby('Age group')['Billing Amount'].sum()

# Get the  Age groups with the highest total billing amount
top_5_total_billing = total_billing_by_age_group.nlargest(5)

# Create a subplot layout (1 row, 2 columns)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))  # Adjust size for better layout

# Plot for highest billing amount
```

```
sns.barplot(y=top_5_highest_billing.index, x=top_5_highest_billing.values
, palette='plasma', ax=axes[0])
axes[0].set_title('Top 5 Age Groups with Highest Billing Amount', fontsize=16)
axes[0].set_xlabel('Highest Billing Amount', fontsize=12)
axes[0].set_ylabel('Age Group', fontsize=12)

# Plot for total billing amount
sns.barplot(y=top_5_total_billing.index, x=top_5_total_billing.values
, palette='magma', ax=axes[1])
axes[1].set_title('Top 5 Age Groups with Highest Total Billing Amount', fontsize=16)
axes[1].set_xlabel('Total Billing Amount', fontsize=12)
axes[1].set_ylabel('Age Group', fontsize=12)

# Adjust layout to prevent overlapping
plt.tight_layout()

# Display the plots
plt.show()
```

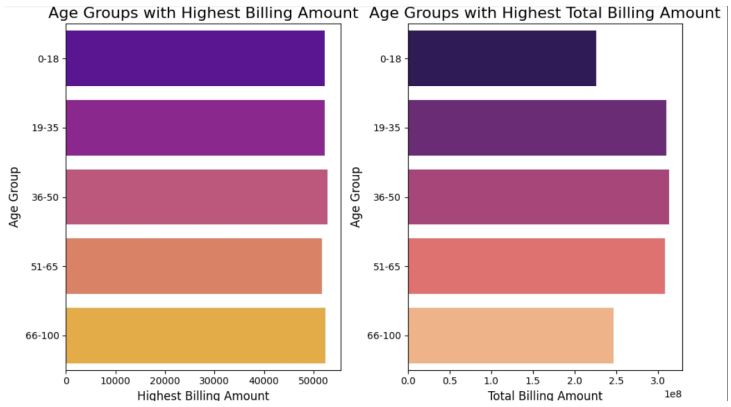

Figure 53: Age group with highest billing amount and total billing amount.

Insight: The 66-100 age group has the lowest billing amount on average, possibly due to less intensive treatments or the prevalence of more manageable chronic conditions in older adults. On the other hand, the 51-65 age group experiences higher billing amounts, which could be linked to more complex medical interventions or procedures often required during midlife.

Interestingly, the 36-50 age group has the highest total billing amount, suggesting this group may have more frequent hospital visits or longer stays, contributing to higher overall healthcare costs. Conversely, the 0-18 age group has the lowest total billing amount, possibly due to fewer admissions and less frequent need for expensive treatments, despite having higher individual treatment costs in some cases.

These trends reflect the varying healthcare needs across different age groups, with younger and older populations tending to incur lower total costs, while middle-aged groups may have higher cumulative medical expenses.

3.Most Expensive Room (Average Billing)

```
# Group by 'Room Number' and calculate the average Billing Amount
avg_billing_by_room = df.groupby('Room Number')['Billing Amount'].mean().
sort_values(ascending=False)

# Get the top 5 most expensive rooms
top_5_expensive_rooms = avg_billing_by_room.head(10)

# Create a bar plot for the top 5 most expensive rooms
plt.figure(figsize=(10, 6))  # Set the size of the plot
sns.barplot(x=top_5_expensive_rooms.index, y=top_5_expensive_rooms.values
, palette='viridis')

# Add titles and labels
plt.title('Top 10 Most Expensive Rooms (Based on Average Billing Amount)'
, fontsize=16)
plt.xlabel('Room Number', fontsize=12)
plt.ylabel('Average Billing Amount', fontsize=12)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

# Display the plot
plt.tight_layout()  # Adjust layout to avoid overlap
plt.show()
```

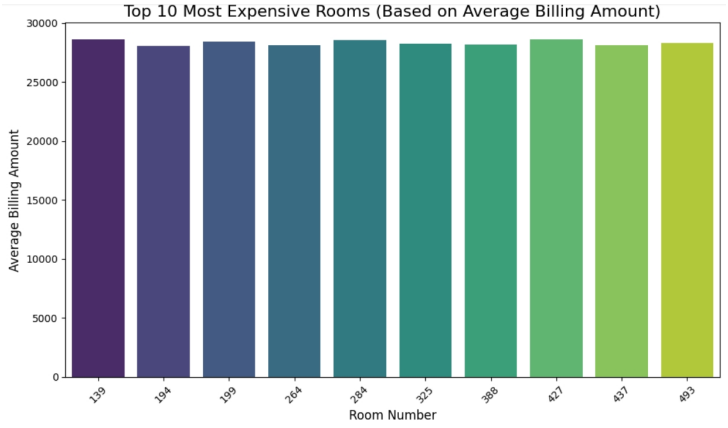

Figure 54: Most Expensive Room (Average Billing).

Insight: Rooms 139 and 427 are the most expensive, with the highest average billing amounts. This could indicate that these rooms are likely associated with specialized care, private suites, or more advanced facilities that require higher costs due to the level of services and comfort provided. These rooms may be reserved for patients requiring intensive care, longer stays, or specialized treatments, driving up the overall billing amount.

3.1 Least Expensive Room (Average Billing)

```
# Group by 'Room number' and calculate the average Billing Amount
average_billing_by_room = df.groupby('Room Number')['Billing Amount'].mean()

# Sort the values in ascending order to find the least expensive rooms
sorted_average_billing_rooms = average_billing_by_room.sort_values(ascending=True)

# Select top N least expensive rooms (e.g., top 10)
least_expensive_rooms = sorted_average_billing_rooms.head(10)  # Adjust this
number as needed

# Create a bar plot
plt.figure(figsize=(10, 6))  # Set figure size
sns.barplot(x=least_expensive_rooms.index, y=least_expensive_rooms.values
, palette='Blues')

# Add titles and labels
plt.title('Top 10 Least Expensive Rooms (Based on Average Billing Amount)'
, fontsize=16)
plt.xlabel('Room Number', fontsize=12)
plt.ylabel('Average Billing Amount', fontsize=12)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

# Display the plot
plt.tight_layout()  # Adjust layout to avoid overlap
plt.show()
```

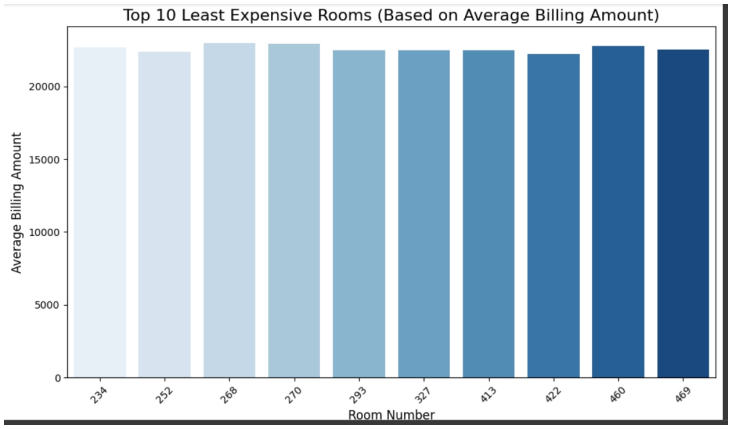

Figure 55: Least Expensive Room (Average Billing)

Insight: Rooms 422 and 252 are the least expensive, with the lowest average billing amounts. This suggests that these rooms may be standard or general hospital rooms, likely associated with routine care that does not require specialized facilities or intensive treatment. Patients assigned to these rooms may be receiving less complex care, resulting in lower associated costs compared to more specialized or private rooms.

4. Doctors with Highest Total Billing Amount

```
# Group by 'Doctor' and calculate the total Billing Amount
total_billing_by_doctor = df.groupby('Doctor')['Billing Amount'].sum()

# Sort the values in descending order to find the top 10 doctors with
the highest billing amounts
sorted_total_billing_by_doctor = total_billing_by_doctor.sort_values(ascending=False)

# Create a scatter plot
plt.figure(figsize=(10, 6))  # Set figure size
plt.scatter(sorted_total_billing_by_doctor.index
, sorted_total_billing_by_doctor.values, color='royalblue')

# Add titles and labels
plt.title('Top 10 Doctors with Highest Total Billing Amount', fontsize=16)
plt.xlabel('Doctor', fontsize=12)
plt.ylabel('Total Billing Amount', fontsize=12)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

# Display the plot
plt.tight_layout()  # Adjust layout to avoid overlap
plt.show()
```

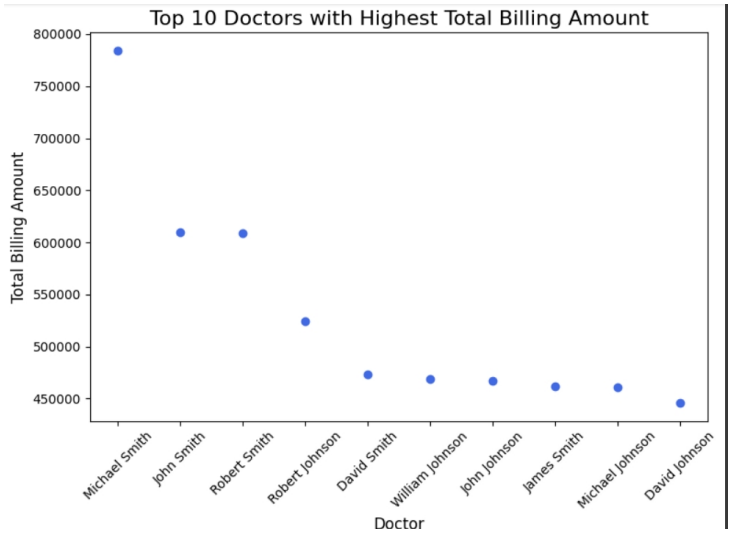

Figure 56: Doctors with Highest Total Billing Amount.

Insight: Michael Smith leading at 784,501.84, followed by John Smith at 610,109.60 and Robert Smith at 609,160.69, indicating that these doctors are the primary revenue drivers. While the top five doctors still generate strong billings, with amounts above 473,345.61, there is a noticeable drop-off in performance after the first three, with David Johnson having the lowest billing at 445,605.54.

4.1.Doctors with Highest Average Billing Amount

```
# Filter doctors with more than 5 entries
doctor_patient_count = df['Doctor'].value_counts()
doctors_with_more_than_5 = doctor_patient_count[doctor_patient_count > 5].index

# Filter the dataset to include only those doctors with more than 5 patients
filtered_df = df[df['Doctor'].isin(doctors_with_more_than_5)]

# Group by 'Doctor' and calculate the average Billing Amount
average_billing_by_doctor = filtered_df.groupby('Doctor')['Billing Amount'].mean()

# Sort the values in descending order to find the top 10 doctors
with the highest average billing amounts
sorted_average_billing_by_doctor = average_billing_by_doctor
.sort_values(ascending=False).head(10)

# Create a horizontal line plot
plt.figure(figsize=(10, 6))  # Set figure size
plt.plot(sorted_average_billing_by_doctor.index
, sorted_average_billing_by_doctor.values, marker='o', linestyle='-', color='darkoran

# Add titles and labels
plt.title('Top 10 Doctors with Highest Average Billing Amount (Count > 5)'
, fontsize=16)
plt.xlabel('Doctor', fontsize=12)
plt.ylabel('Average Billing Amount', fontsize=12)
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

# Display the plot
plt.tight_layout()  # Adjust layout to avoid overlap
plt.show()
```

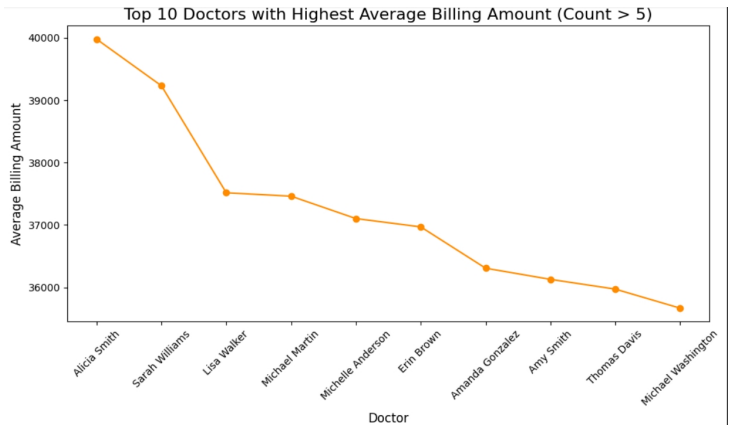

Figure 57: Doctors with Highest Average Billing Amount.

Insight: The data reveals that Michael Martin has the highest average billing amount at 37,462.30, with 11 billing events, indicating that he generates both a higher average and a greater volume of billings compared to others. On the other hand, Thomas Davis has the lowest average billing at 35,972.99, with 6 billing events, suggesting that while his performance is still strong, it lags slightly behind others in the group. This highlights that doctors like Michael Martin, who combine both high average billings and a higher count of billable activities, are key revenue contributors. Additionally, it is notable that most of the doctors in the group have billing counts between 6 and 8, with only a few having counts above that, such as Michael Martin. This suggests that there may be potential for increasing billing volumes across the board, as a higher frequency of billable activities could drive overall revenue growth. Doctors like Thomas Davis may benefit from strategies aimed at increasing either the frequency or value of their billable activities to improve their standing relative to peers.

4.2.Doctor with highest billing amount and lowest billing amount

```
# Step 1: Group by 'Doctor' and calculate the max and min Billing Amount
doctor_max_billing = df.groupby('Doctor')['Billing Amount'].max().sort_values(ascendi
.head(1)
doctor_min_billing = df.groupby('Doctor')['Billing Amount'].min().sort_values()
.head(1)

# Step 2: Get the Doctor names and their billing amounts
doctor_max_name = doctor_max_billing.index[0]
doctor_max_value = doctor_max_billing.values[0]

doctor_min_name = doctor_min_billing.index[0]
doctor_min_value = doctor_min_billing.values[0]

# Step 3: Create the plot
plt.figure(figsize=(10, 6))

# Plotting Doctor with the Highest Billing Amount
plt.subplot(1, 2, 1)  # 1 row, 2 columns, first subplot
plt.barh([doctor_max_name], [doctor_max_value], color='green')
plt.title(f"Doctor with Highest Billing\n{doctor_max_name}", fontsize=14)
plt.xlabel('Billing Amount', fontsize=12)
plt.ylabel('Doctor', fontsize=12)

# Plotting Doctor with the Lowest Billing Amount
plt.subplot(1, 2, 2)  # 1 row, 2 columns, second subplot
plt.barh([doctor_min_name], [doctor_min_value], color='red')
plt.title(f"Doctor with Lowest Billing\n{doctor_min_name}", fontsize=14)
plt.xlabel('Billing Amount', fontsize=12)
plt.ylabel('Doctor', fontsize=12)

# Adjust layout to avoid overlap
```

```
plt.tight_layout()

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

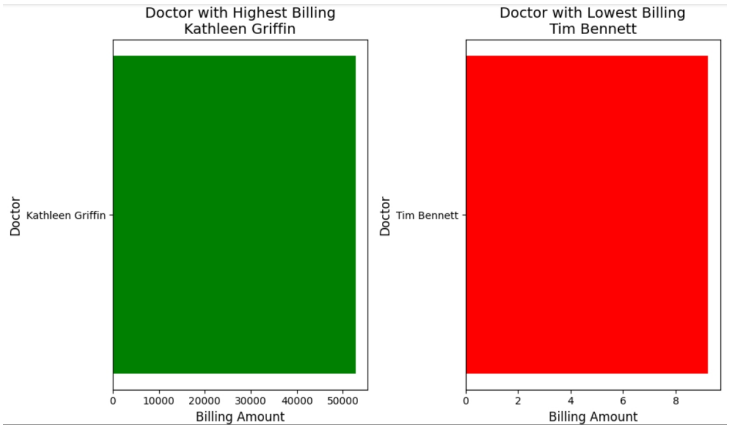

Figure 58: Doctor with highest billing amount and lowest billing amount.

Insight: The data reveals a notable contrast in billing amounts among the doctors, with Kathleen Griffin having the highest billing at 52,764, significantly outpacing others in the group. This indicates that Kathleen Griffin likely performs high-value procedures or has a high patient volume, positioning her as a top revenue generator. In contrast, Tim Bennett, despite having a relatively higher billing count of 9, has the lowest billing amount, which suggests that his billable activities may be lower in value or frequency compared to peers. This disparity highlights the importance of both the value and volume of billable activities in driving revenue, with doctors like Kathleen Griffin excelling in both. For doctors like Tim Bennett, there may be opportunities to improve either the types of services offered or increase patient volume to better align with higher-billing colleagues.

5.Gender with Highest Average Billing Amount

```
# Group by 'Gender' and sum 'Billing Amount'
gender_billing_sum = df.groupby('Gender')['Billing Amount'].sum()

# Sort values in descending order and get the top 2 genders
top_gender = gender_billing_sum.sort_values(ascending=False).head(2)

# Plot a bar graph
top_gender.plot(kind='barh', color='skyblue')  # Bar plot horizontally

# Reverse the bar direction
plt.gca().invert_xaxis()
```

```
# Adding title and labels
plt.title('Gender with Highest Total Billing Amount')
plt.xlabel('Total Billing Amount')
plt.ylabel('Gender')

# Show plot
plt.show()
```

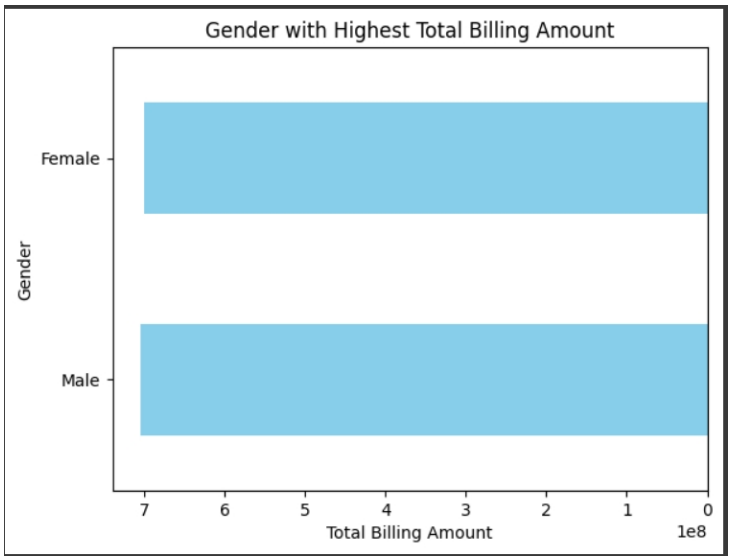

Figure 59: Gender with Highest Average Billing Amount.

Insight: The analysis of the healthcare dataset reveals a significant gender-based difference in average billing amounts. Males have the highest average billing amount at 25,616, which could indicate that male patients tend to incur higher healthcare costs, possibly due to factors such as more frequent hospitalizations, higher-cost treatments, or a higher prevalence of certain medical conditions. In contrast, Females have a slightly lower average of 25,476, which, while close, may reflect differences in healthcare utilization patterns, such as lower-frequency but potentially more cost-effective visits, or a higher reliance on preventative care. These insights can help healthcare providers and insurers better understand gender-specific healthcare needs and tailor their services, plans, or interventions accordingly to address these distinct patterns in healthcare spending.

5.1 Gender with the Highest Total Billing Amount

```
# Group by 'Gender' and sum 'Billing Amount'
gender_billing_sum = df.groupby('Gender')['Billing Amount'].sum()

# Sort values in descending order and get the top gender
top_gender = gender_billing_sum.sort_values(ascending=False).head(2)

# Plot the bar graph
top_gender.plot(kind='bar', color=['skyblue', 'lightcoral'])  # Bar plot vertically
```

```
# Adding title and labels
plt.title('Top Gender with the Highest Total Billing Amount')
plt.xlabel('Gender')
plt.ylabel('Total Billing Amount')

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

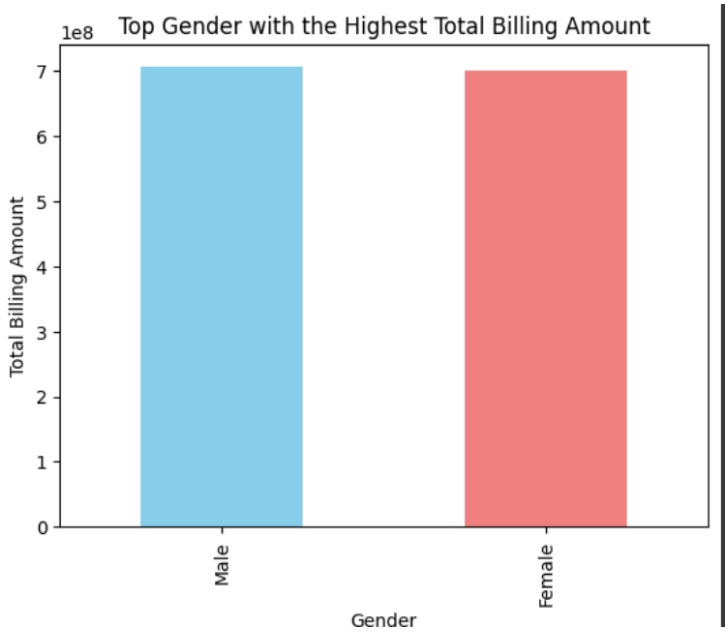

Figure 60: Gender with the Highest Total Billing Amount.

Insight: The analysis shows that Male patients have a slightly higher total billing amount of 704,346,200 compared to Female patients, who have a total billing amount of 699,828,600. While the difference is relatively small, it suggests that, overall, male patients are incurring slightly higher healthcare costs. This could be due to a variety of factors, such as a higher frequency of expensive treatments, hospitalizations, or certain health conditions that are more prevalent in males. On the other hand, the slightly lower total for females may reflect differences in healthcare utilization patterns, such as higher use of preventative care or outpatient services that generally incur lower costs. Understanding these trends can help healthcare providers and insurers design more targeted and effective strategies to manage healthcare expenses and ensure that both male and female patients receive the appropriate care for their needs.

6.Average Stay Duration for Admission Type

```
# Group by 'Admission Type' and calculate the average 'Stay Duration'
average_stay_duration = df.groupby('Admission Type')['Duration'].mean()

# Plotting the average stay duration for each admission type
```

```
average_stay_duration.plot(kind='bar', color='lightgreen')

# Adding title and labels
plt.title('Average Stay Duration for Admission Type')
plt.xlabel('Admission Type')
plt.ylabel('Average Stay Duration (Days)')

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

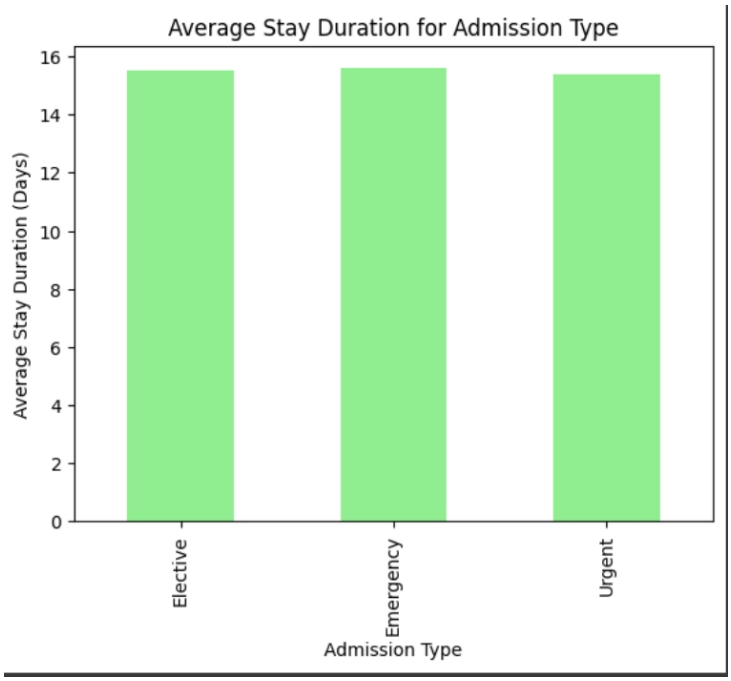

Figure 61: Average Stay Duration for Admission Type.

Insight: The analysis shows that the average stay durations for different types of admissions are quite similar, with Emergency admissions having the longest average stay at 15.6 days, followed by Elective admissions at 15.5 days, and Urgent admissions at 15.4 days. While the differences are minimal, the slightly longer stay for Emergency admissions may reflect the often more complex and critical nature of emergency cases, requiring extended treatment or recovery. Elective admissions, though planned in advance, may involve detailed procedures that also necessitate a prolonged stay. The marginally shorter average for Urgent admissions suggests these cases, while requiring prompt attention, may not involve the same level of complexity or recovery time as emergency cases. These insights can guide healthcare providers in resource planning and management based on the typical stay durations for each admission type.

7.Most Common Age for Admission Type

```
# Find the most common age for each Admission Type
```

```
most_common_age = df.groupby('Admission Type')['Age'].agg(lambda x: x.mode()[0])

# Plotting the most common age for each admission type
most_common_age.plot(kind='bar', color='lightblue', edgecolor='black')

# Adding title and labels
plt.title('Most Common Age for Each Admission Type')
plt.xlabel('Admission Type')
plt.ylabel('Most Common Age')

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

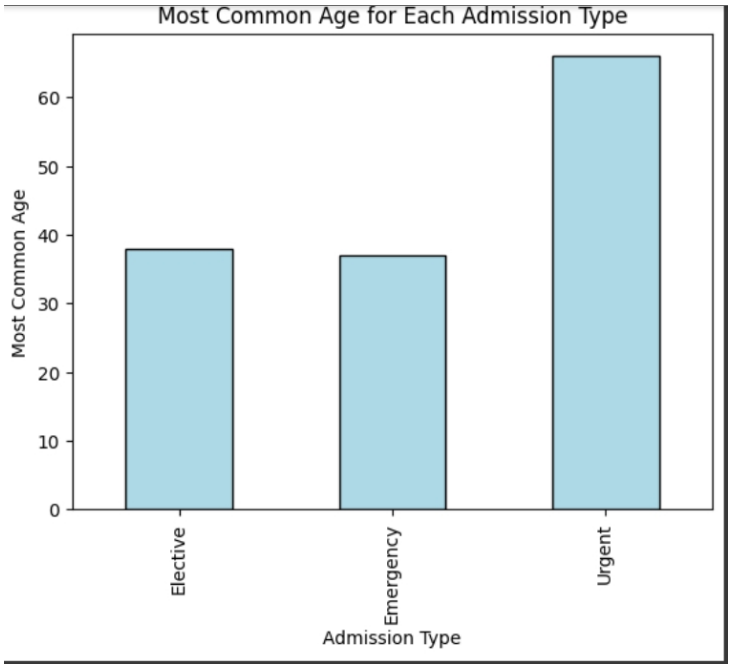

Figure 62: Most Common Age for Admission Type.

Insight: The most common ages for different admission types reveal distinct age-related patterns: Elective admissions are most frequent for 38-year-olds, suggesting that people in their late 30s often undergo planned procedures, such as surgeries or treatments for non-urgent conditions. Urgent admissions, with a peak at 66 years, highlight that older adults are more likely to require immediate medical care due to chronic health issues or age-related acute conditions. In contrast, Emergency admissions are most common for 37-year-olds, indicating that younger adults experience a higher frequency of sudden, acute health issues or accidents that necessitate emergency care. These patterns emphasize age-specific healthcare needs and can help providers tailor care strategies and resource allocation accordingly.

8.Average Billing Amount by Blood Type

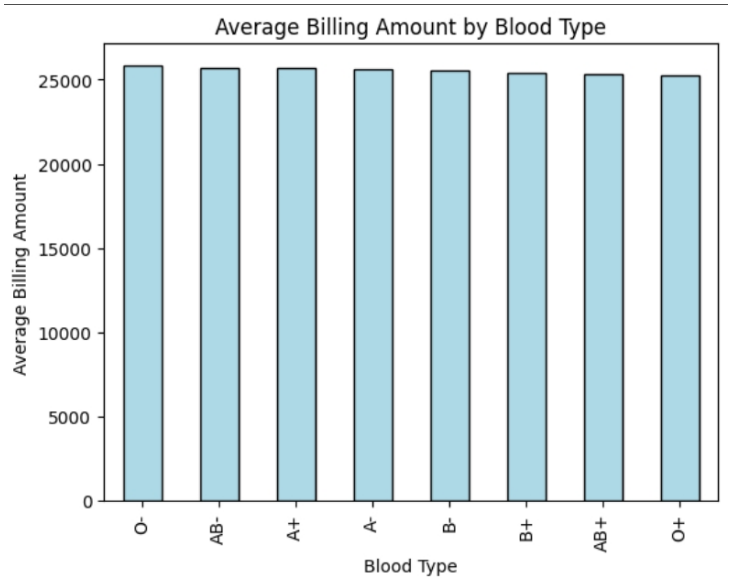

Figure 63: Average Billing Amount by Blood Type.

Insight: The analysis of average billing amounts across different blood types shows that O- has the highest average billing amount at 25,839.89, followed closely by AB- at 25,703.52, while the lowest average billing amount is for O+ at 25,240.11. The other blood types exhibit relatively similar billing amounts, ranging from 25,240.11 (for O+) to 25,664.01 (for A+). These small variations suggest that blood type does not have a major impact on healthcare costs. Instead, the differences are likely influenced by other factors such as patient demographics, treatment types, or regional healthcare practices. However, further investigation could uncover if specific conditions related to certain blood types contribute to these trends.

8.1.Total Billing Amount by Blood Type

```
# Group by 'Blood Type' and calculate the total 'Billing Amount'
total_billing_by_blood_type = df.groupby('Blood Type')['Billing Amount'].sum()
.sort_values(ascending=False)

# Plotting the total billing amount for each blood type
total_billing_by_blood_type.plot(kind='bar', color='lightcoral', edgecolor='black')

# Adding title and labels
plt.title('Total Billing Amount by Blood Type')
plt.xlabel('Blood Type')
plt.ylabel('Total Billing Amount')

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

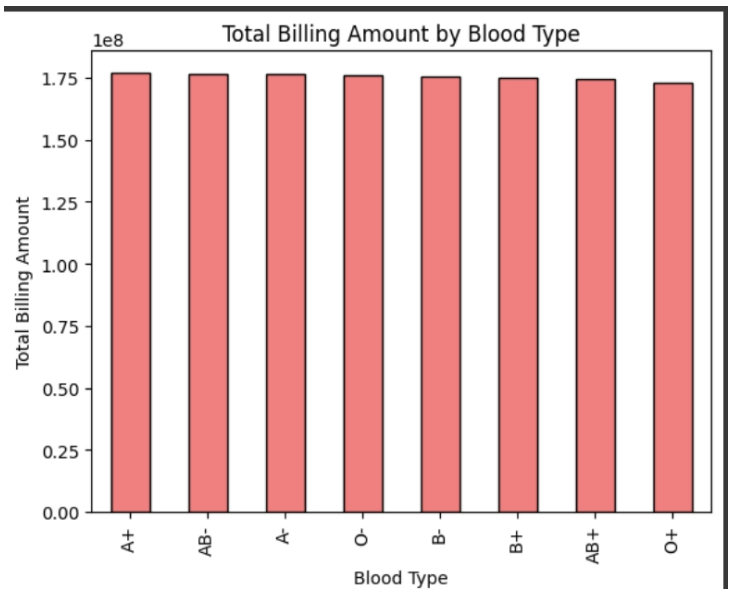

Figure 64: Total Billing Amount by Blood Type.

Insight: The analysis of total billing amounts by blood type shows that the differences in billing amounts across blood types are relatively small. A+ has the highest total billing amount at 176,979,000, followed closely by AB- at 176,686,000, and A- at 176,591,000. The rest of the blood types—O-, B-, B+, AB+, and O+—have total billing amounts that are very close, with the lowest being O+ at 173,020,900. These minimal differences suggest that, although there are slight variations in the total billing amounts for each blood type, the overall healthcare costs do not vary significantly across blood types.

8.2.Maximum and Minimum Billing Amount for Blood Type

```
# Group by 'Blood Type' and calculate the maximum 'Billing Amount'
max_billing_by_blood_type = df.groupby('Blood Type')['Billing Amount'].max()
.sort_values(ascending=False)

# Group by 'Blood Type' and calculate the minimum 'Billing Amount'
min_billing_by_blood_type = df.groupby('Blood Type')['Billing Amount'].min()
.sort_values(ascending=False)

# Set up subplots (1 row, 2 columns)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plot the maximum billing amount for each blood type
axes[0].bar(max_billing_by_blood_type.index, max_billing_by_blood_type.values
, color='lightcoral', edgecolor='black')
axes[0].set_title('Maximum Billing Amount by Blood Type')
axes[0].set_xlabel('Blood Type')
axes[0].set_ylabel('Billing Amount')

# Plot the minimum billing amount for each blood type
axes[1].bar(min_billing_by_blood_type.index, min_billing_by_blood_type.values
```

```
, color='lightblue', edgecolor='black')
axes[1].set_title('Minimum Billing Amount by Blood Type')
axes[1].set_xlabel('Blood Type')
axes[1].set_ylabel('Billing Amount')

# Adjust layout for better spacing
plt.tight_layout()

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

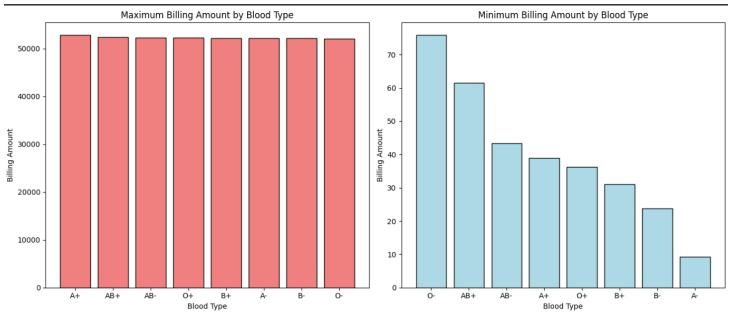

Figure 65: Maximum and Minimum Billing Amount for Blood Type.

Insight: The data on highest and lowest billing amounts by blood type reveals some interesting patterns. The highest billing amounts are fairly close across most blood types, with A+ having the highest at 52,764.28, followed by AB+ at 52,373.03 and AB- at 52,271.66. These values suggest that the billing amounts for different blood types are relatively similar at the higher end, with only small variations. On the other hand, the lowest billing amounts show more significant variation. O- has the lowest billing amount at 75.82, which is substantially higher than the lowest for A- at 9.24, indicating a wider spread in the lower range of billing amounts. AB+ also has a notably high lowest billing amount at 61.44, suggesting that certain blood types might be associated with more frequent or more expensive treatments, while others tend to have lower billing amounts overall. This could reflect underlying differences in treatment complexity or the types of healthcare services typically required for individuals of each blood type.

9.Average Age by Medical Condition

```
# Group by 'Medical Condition' and calculate the average 'Age'
avg_age_by_condition = df.groupby('Medical Condition')['Age'].mean()
.sort_values(ascending=False)

# Plotting the average age by medical condition
avg_age_by_condition.plot(kind='bar', color='lightcoral', edgecolor='black')

# Adding title and labels
plt.title('Average Age by Medical Condition')
```

```
plt.xlabel('Medical Condition')
plt.ylabel('Average Age')

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

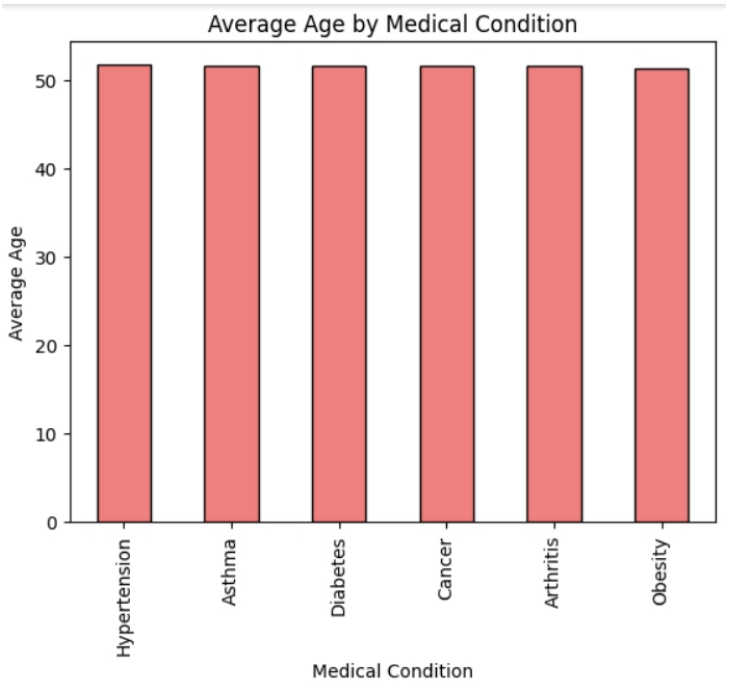

Figure 66: Average Age by Medical Condition.

Insight: The average age for individuals with different medical conditions in this dataset is quite consistent, with all conditions having average ages close to 51 years. Hypertension has the highest average age at 51.72, followed by Asthma at 51.60, and Diabetes at 51.58. The other conditions, such as Cancer, Arthritis, and Obesity, show similar average ages, all hovering around 51 years. This suggests that these medical conditions are typically diagnosed or more prevalent in middle-aged adults, as the average age does not vary significantly across conditions. It may also indicate that factors such as lifestyle or age-related health issues contribute similarly to the onset of these conditions in this age group.

9.1. Minimum and Maximum Age by Medical Condition

```
# Group by 'Medical Condition' and calculate the maximum 'Age'
max_age_by_condition = df.groupby('Medical Condition')['Age'].max()
.sort_values(ascending=False)

# Group by 'Medical Condition' and calculate the minimum 'Age'
min_age_by_condition = df.groupby('Medical Condition')['Age'].min()
.sort_values(ascending=False)

# Set up subplots (1 row, 2 columns)
```

```
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plot the maximum age for each medical condition
axes[0].bar(max_age_by_condition.index, max_age_by_condition.values
, color='lightgreen', edgecolor='black')
axes[0].set_title('Maximum Age by Medical Condition')
axes[0].set_xlabel('Medical Condition')
axes[0].set_ylabel('Maximum Age')

# Plot the minimum age for each medical condition
axes[1].bar(min_age_by_condition.index, min_age_by_condition.values
, color='lightcoral', edgecolor='black')
axes[1].set_title('Minimum Age by Medical Condition')
axes[1].set_xlabel('Medical Condition')
axes[1].set_ylabel('Minimum Age')

# Adjust layout for better spacing
plt.tight_layout()

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

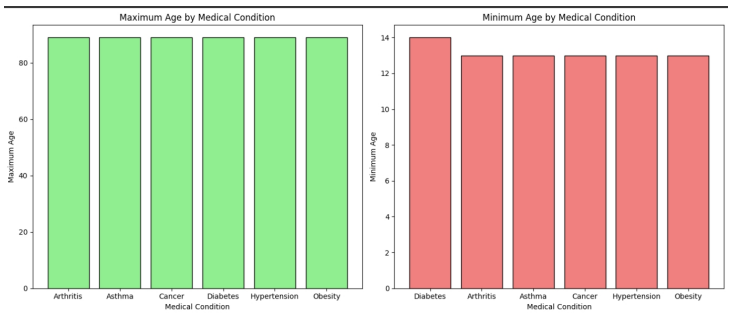

Figure 67: Minimum and Maximum Age by Medical Condition.

Insight: The data reveals that the maximum age for individuals across all medical conditions is uniformly 89 years, suggesting that the oldest patients diagnosed with any of these conditions are of similar age, regardless of their specific health condition. This could indicate that older individuals are typically diagnosed with these common conditions as part of the natural aging process. On the other hand, the minimum age for all conditions is approximately 13 years, which suggests that these medical conditions also affect younger populations, with 13 years being the youngest recorded age for each condition. This could reflect the early onset of these diseases or the presence of these conditions in adolescents, possibly due to genetic factors, lifestyle choices, or environmental influences.

10.Average Stay Duration by Medical Condition

```
# Group by 'Medical Condition' and calculate the average 'Duration'
avg_duration_by_condition = df.groupby('Medical Condition')['Duration'].mean()
```

```
.sort_values(ascending=False)

# Plotting the average stay duration by medical condition using a line plot
plt.figure(figsize=(10, 6))
sns.lineplot(x=avg_duration_by_condition.index, y=avg_duration_by_condition.values
, marker='o', color='b')

# Adding title and labels
plt.title('Average Stay Duration by Medical Condition', fontsize=14)
plt.xlabel('Medical Condition', fontsize=12)
plt.ylabel('Average Duration (days)', fontsize=12)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

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

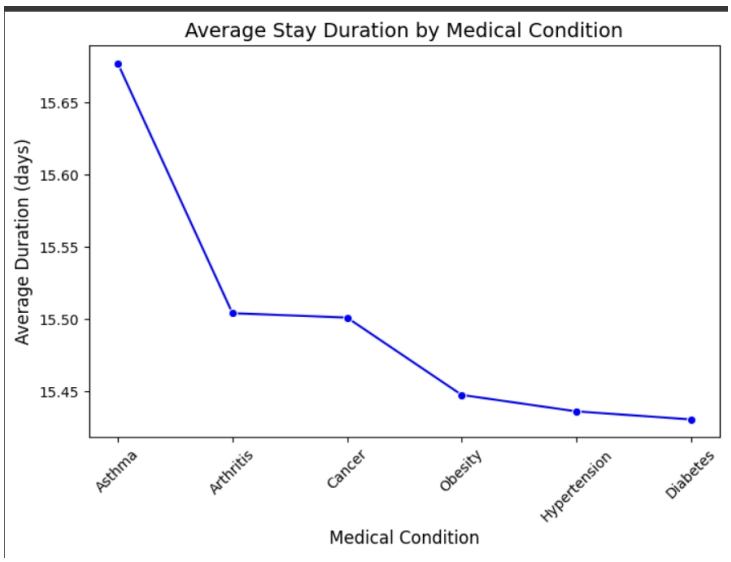

Figure 68: Average Stay Duration by Medical Condition.

Insight: The data shows that the average stay duration for individuals with various medical conditions is very similar, with all conditions having an average stay of approximately 15 days. The slight variations in the average stay, such as Asthma with the highest average of 15.68 days and Diabetes with the lowest at 15.43 days, are minimal and suggest that the length of hospital stays across these conditions is relatively consistent. This could indicate that these medical conditions, while different in nature, may require similar levels of care or treatment, which results in comparable hospital stays.

11.Total Billing Amount by Insurance Provider

```
# Group by 'Insurance Provider' and calculate the total 'Billing Amount'
```

```
total_billing_by_provider = df.groupby('Insurance Provider')['Billing Amount'].sum()
.sort_values(ascending=False)

# Plotting the total billing amount by insurance provider using a pie chart
plt.figure(figsize=(8, 8))
plt.pie(total_billing_by_provider, labels=total_billing_by_provider.index, autopct='%
, startangle=140, colors=['lightblue', 'lightgreen', 'lightcoral', 'yellowgreen'
, 'lightskyblue'])

# Adding title
plt.title('Total Billing Amount by Insurance Provider')

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

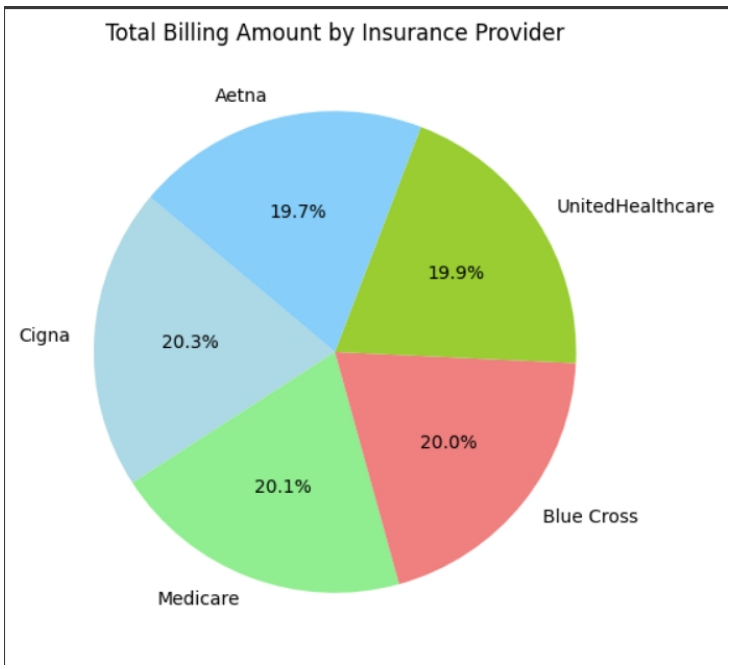

Figure 69: Total Billing Amount by Insurance Provider.

Insight: The data reveals that Cigna has the highest total billing amount at approximately 284.36 million, followed closely by Medicare with 282.93 million and Blue Cross with 280.42 million. This suggests that these three insurance providers are significantly contributing to the hospital's revenue, likely covering a substantial portion of the patient population. The high billing amounts for these providers could indicate that they insure a large number of individuals who require frequent or expensive healthcare services, or that the treatments provided to their policyholders are of a higher cost. The relatively close billing amounts among these top providers may also suggest that they have similar market shares or client bases in the region.

11.1.Maximum and Minimum Billing Amount by Insurance Provider

```
# Group by 'Insurance Provider' and calculate the maximum 'Billing Amount'
```

```
max_billing_by_provider = df.groupby('Insurance Provider')['Billing Amount'].max()
.sort_values(ascending=False)

# Group by 'Insurance Provider' and calculate the minimum 'Billing Amount'
min_billing_by_provider = df.groupby('Insurance Provider')['Billing Amount'].min()
.sort_values(ascending=False)

# Set up subplots (1 row, 2 columns)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plot the maximum billing amount for each insurance provider
axes[0].bar(max_billing_by_provider.index, max_billing_by_provider.values
, color='lightgreen', edgecolor='black')
axes[0].set_title('Maximum Billing Amount by Insurance Provider')
axes[0].set_xlabel('Insurance Provider')
axes[0].set_ylabel('Maximum Billing Amount')

# Plot the minimum billing amount for each insurance provider
axes[1].bar(min_billing_by_provider.index, min_billing_by_provider.values
, color='lightcoral', edgecolor='black')
axes[1].set_title('Minimum Billing Amount by Insurance Provider')
axes[1].set_xlabel('Insurance Provider')
axes[1].set_ylabel('Minimum Billing Amount')

# Adjust layout for better spacing
plt.tight_layout()

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

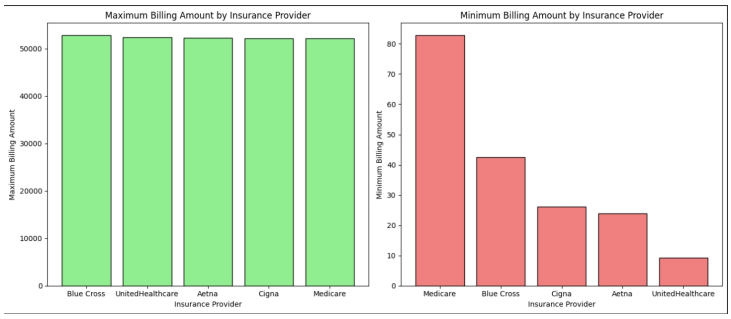

Figure 70: Maximum and Minimum Billing Amount by Insurance Provider.

Insight: The data shows that Blue Cross has the highest billing amount at 52,764, significantly higher than the other insurance providers, followed by UnitedHealthcare with 52,373 and Aetna with 52,212. This suggests that Blue Cross likely covers more high-cost treatments or has a larger number of insured individuals requiring expensive healthcare services. On the other hand, Medicare has the lowest billing amount, with 82.77, which is notably lower compared to the others. This could indicate that Medicare's billing is

primarily associated with lower-cost treatments or fewer high-billing cases. The substantial difference in billing amounts between Medicare and the other providers might also reflect variations in reimbursement rates, patient demographics (e.g., senior citizens for Medicare), or the types of treatments covered under each insurance plan.

11.2.Average Billing Amount by Insurance Provider

```
# Group by 'Insurance Provider' and calculate the total 'Billing Amount'
total_billing_by_provider = df.groupby('Insurance Provider')['Billing Amount'].mean()
.sort_values(ascending=False)

# Plotting the total billing amount by insurance provider using a pie chart
plt.figure(figsize=(8, 5))
plt.pie(total_billing_by_provider, labels=total_billing_by_provider.index, autopct='%
, startangle=140, colors=['lightblue', 'lightgreen', 'lightcoral', 'yellowgreen'
, 'lightskyblue'])

# Adding title
plt.title('Average Billing Amount by Insurance Provider')

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

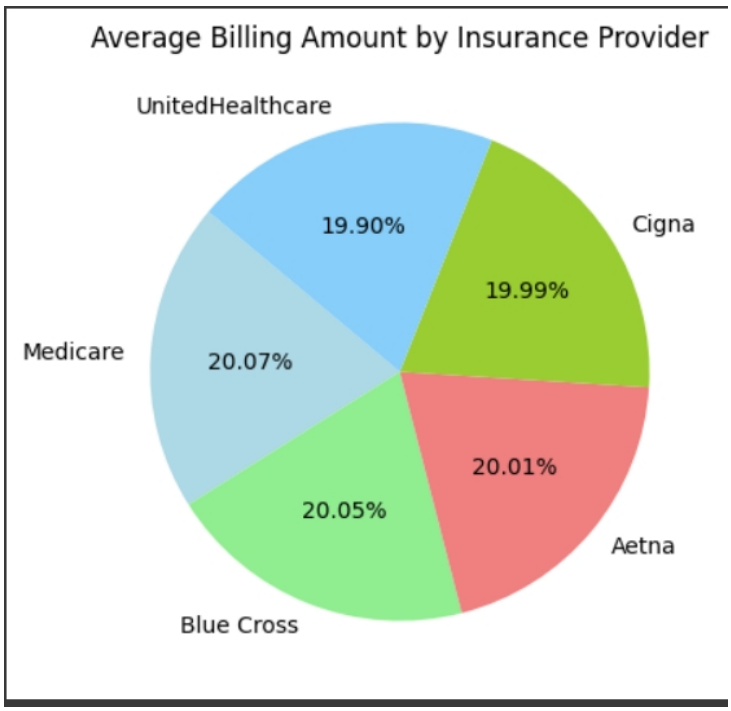

Figure 71: Average Billing Amount by Insurance Provider.

Insight: The data shows that Medicare has the highest average billing amount at approximately 25,630, followed closely by Blue Cross with 25,604 and Aetna at 25,553. This suggests that Medicare, despite being a government-backed insurance provider typically

associated with senior citizens, is covering higher-cost services or a more expensive patient demographic compared to other providers. The high average billing amounts for Blue Cross and Aetna further indicate that these private insurance providers also likely cover high-cost treatments or a large population requiring complex or chronic care. The closeness in the average billing amounts between these top three providers suggests that the level of coverage or patient care provided by each is somewhat similar, but still varies enough to give Medicare a slight edge. Further analysis could reveal whether differences in billing are due to patient volume, specific health conditions treated, or differences in reimbursement rates and treatment protocols.

12.Hospitals with highest total Billing Amount

```
# Group by 'Hospital' and calculate the total 'Billing Amount'
total_billing_by_hospital = df.groupby('Hospital')['Billing Amount'].sum()
.sort_values(ascending=False).head(10)

# Plotting the total billing amount for the top 10 hospitals
plt.figure(figsize=(8, 6))
total_billing_by_hospital.plot(kind='bar', color='skyblue', edgecolor='black')

# Adding title and labels
plt.title('Top 10 Hospitals with Highest Total Billing Amount', fontsize=14)
plt.xlabel('Hospital', fontsize=12)
plt.ylabel('Total Billing Amount', fontsize=12)

# Display the plot
plt.tight_layout()
plt.show()
```

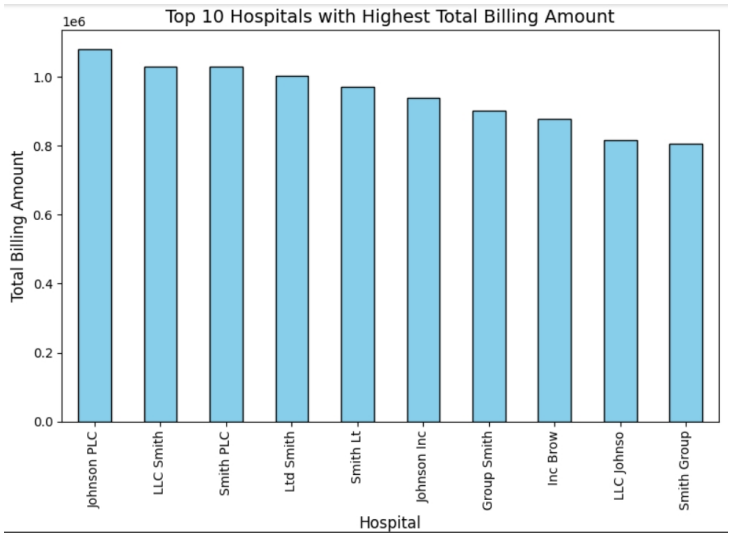

Figure 72: Hospitals with highest total Billing Amount .

Insight: The data indicates that Johnson PLC has the highest total billing amount at approximately 1.08 million, followed closely by LLC Smith with 1.03 million and Smith

PLC with 1.03 million as well. This suggests that Johnson PLC is the leading hospital in terms of revenue generated from billing in this dataset, possibly due to a larger patient base, higher-cost treatments, or more extensive healthcare services offered. The close proximity in billing amounts between LLC Smith and Smith PLC further implies that these hospitals may be competing in similar markets or providing similar levels of care. The other hospitals in the top 10, such as Ltd Smith and Smith Lt, also show significant billing amounts, indicating that hospitals with "Smith" in their names may dominate the high-billing category.

12.1.Hospitals with Maximum and Minimum Billing Amount

```
# Group by 'Hospital' and calculate the maximum 'Billing Amount'
max_billing_by_hospital = df.groupby('Hospital')['Billing Amount'].max()
.sort_values(ascending=False).head(10)

# Group by 'Hospital' and calculate the minimum 'Billing Amount'
min_billing_by_hospital = df.groupby('Hospital')['Billing Amount'].min()
.sort_values().head(10)

# Set up subplots (1 row, 2 columns)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plot the maximum billing amount for the top 10 hospitals
axes[0].bar(max_billing_by_hospital.index, max_billing_by_hospital.values
, color='lightgreen', edgecolor='black')
axes[0].set_title('Top 10 Hospitals with Highest Billing Amount', fontsize=14)
axes[0].set_xlabel('Hospital', fontsize=12)
axes[0].set_ylabel('Maximum Billing Amount', fontsize=12)
axes[0].tick_params(axis='x', rotation=45)

# Plot the minimum billing amount for the top 10 hospitals
axes[1].bar(min_billing_by_hospital.index, min_billing_by_hospital.values
, color='lightcoral', edgecolor='black')
axes[1].set_title('Top 10 Hospitals with Lowest Billing Amount', fontsize=14)
axes[1].set_xlabel('Hospital', fontsize=12)
axes[1].set_ylabel('Minimum Billing Amount', fontsize=12)
axes[1].tick_params(axis='x', rotation=45)

# Adjust layout for better spacing
plt.tight_layout()

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

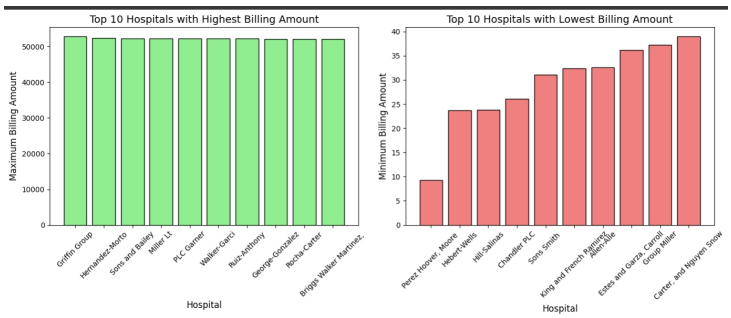

Figure 73: Hospitals with Maximum and Minimum Billing Amount.

Insight: Based on the provided data, Griffin Group has the highest billing amount at approximately 52,764, followed closely by Hernandez-Morto with 52,373 and Sons and Bailey with 52,272. This suggests that Griffin Group is the leading hospital in terms of revenue generated from billing, which could indicate a higher volume of patients or more expensive procedures being carried out at this hospital compared to others in the list. On the other hand, Perez Hoover, Moore has the lowest billing amount at 9.24, which is significantly lower than the others, indicating that it likely handles lower-cost services or has fewer high-billing cases.Hospitals like Griffin Group may cater to high-cost specialties or have a broader range of services, while Perez Hoover, Moore might focus on less expensive treatments or have a smaller patient base.

12.2.Hospitals with Highest and Lowest Average Billing Amount

```
# Calculate aggregate statistics for each hospital
v1 = df.groupby('Hospital')['Billing Amount'].agg(['mean', 'count', 'max', 'min'])

# Filter for hospitals with more than 5 entries
v1_filtered = v1[v1['count'] > 5]

# Sort the hospitals by highest average billing amount and lowest
average billing amount
top_10_highest_avg = v1_filtered.sort_values(by='mean', ascending=False).head(10)
top_10_lowest_avg = v1_filtered.sort_values(by='mean').head(10)

# Set up subplots (1 row, 2 columns)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plot the top 10 hospitals with highest average billing amount
axes[0].bar(top_10_highest_avg.index, top_10_highest_avg['mean'], color='lightgreen'
, edgecolor='black')
axes[0].set_title('Top 10 Hospitals with Highest Average Billing Amount', fontsize=14
axes[0].set_xlabel('Hospital', fontsize=12)
axes[0].set_ylabel('Average Billing Amount', fontsize=12)
axes[0].tick_params(axis='x', rotation=45)

# Plot the top 10 hospitals with lowest average billing amount
axes[1].bar(top_10_lowest_avg.index, top_10_lowest_avg['mean'], color='lightcoral'
```

```
, edgecolor='black')
axes[1].set_title('Top 10 Hospitals with Lowest Average Billing Amount', fontsize=14)
axes[1].set_xlabel('Hospital', fontsize=12)
axes[1].set_ylabel('Average Billing Amount', fontsize=12)
axes[1].tick_params(axis='x', rotation=45)

# Adjust layout for better spacing
plt.tight_layout()

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

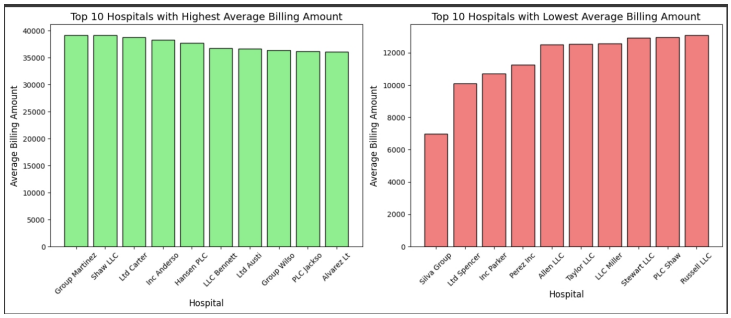

Figure 74: Hospitals with Highest and Lowest Average Billing Amount.

Insight: The hospitals with the highest average billing amounts, such as Group Martinez, Shaw LLC, and Ltd Carter, show significantly higher averages (ranging from approximately 39,000 to 37,000), indicating that they likely provide more complex or high-cost treatments, possibly with a larger volume of patients or specialized services. In contrast, the hospitals with the lowest average billing amounts, such as Silva Group, Ltd Spencer, and Inc Parker, have averages as low as around 7,000 to 13,000, suggesting that these hospitals may focus on less expensive treatments, cater to a different patient demographic, or operate in lower-cost regions. This stark contrast highlights the variability in healthcare costs across different hospitals, reflecting differences in service types, patient needs, and insurance reimbursements.

13.Average Billing Amount by Admission Type and Gender

```
# Grouping by 'Admission Type' and 'Gender' and calculating the mean Billing Amount
average_billing = df.groupby(['Admission Type', 'Gender'])['Billing Amount'].mean()
.sort_values(ascending=False)

# Plotting
plt.figure(figsize=(8, 5))
average_billing.plot(kind='bar', color=['lightblue', 'lightgreen', 'lightcoral'
, 'lightyellow'])

# Adding title and labels
plt.title('Average Billing Amount by Admission Type and Gender', fontsize=14)
```

```
plt.xlabel('Admission Type & Gender', fontsize=12)
plt.ylabel('Average Billing Amount', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

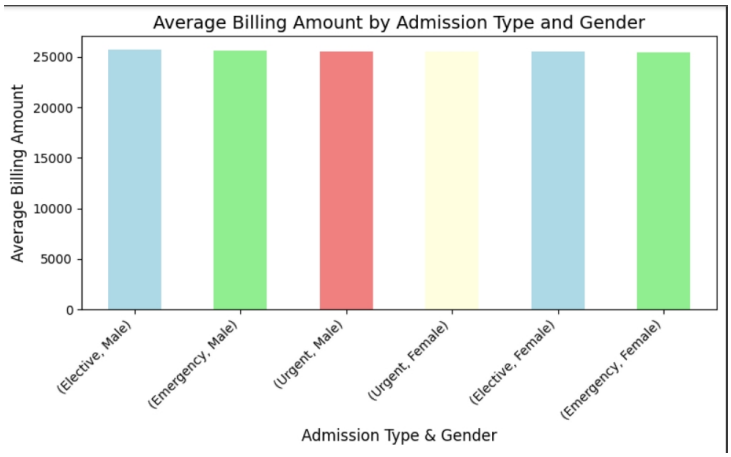

Figure 75: Average Billing Amount by Admission Type and Gender.

Insight: The data reveals that Elective male patients have the highest average billing amount at approximately 25,741, followed closely by Emergency male patients at 25,566 and Urgent male patients at 25,541. This suggests that male patients undergoing elective procedures may require more specialized or expensive treatments, leading to higher billing amounts. In contrast, female patients across all admission types have slightly lower average billing amounts, with Elective female patients having the lowest average at 25,489. This pattern may reflect differences in the types of care provided, gender-based healthcare utilization, or possibly the demographic characteristics of the patients within each group.

13.1 Total Billing Amount by Admission Type and Gender

```
# Grouping by 'Admission Type' and 'Gender' and calculating the total Billing Amount
total_billing = df.groupby(['Admission Type', 'Gender'])['Billing Amount'].sum()
.sort_values(ascending=False)

# Plotting the total billing amount
plt.figure(figsize=(8, 6))
total_billing.plot(kind='bar', color=['lightblue', 'lightgreen', 'lightcoral'
, 'lightyellow'])

# Adding title and labels
plt.title('Total Billing Amount by Admission Type and Gender', fontsize=14)
plt.xlabel('Admission Type & Gender', fontsize=12)
```

```
plt.ylabel('Total Billing Amount', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

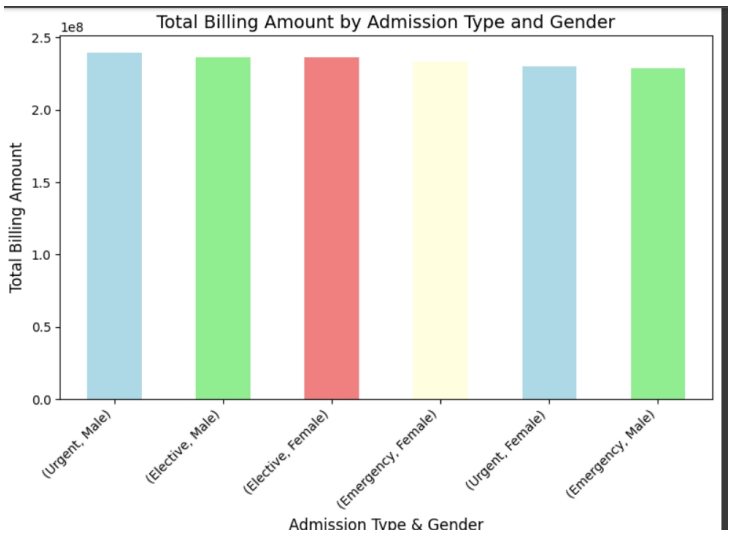

Figure 76: Total Billing Amount by Admission Type and Gender.

Insight: The data indicates that Urgent male patients have the highest total billing amount at approximately 239.27 million, followed closely by Elective male patients with 236.61 million. This suggests that male patients undergoing urgent procedures tend to incur significantly higher costs, likely due to the nature of urgent care which may involve more intensive treatments or longer stays. Elective male patients also represent a substantial billing amount, possibly reflecting planned surgeries or procedures that require specialized care. In contrast, Female urgent patients and Emergency male patients have lower total billing amounts, pointing to differences in treatment costs or patient demographics across genders for specific admission types.

13.2.Highest and Lowest Billing Amount by Admission Type and Gender

```
# Grouping by 'Admission Type' and 'Gender' and calculating the max and min Billing A
max_billing = df.groupby(['Admission Type', 'Gender'])['Billing Amount'].max()
.sort_values(ascending=False)
min_billing = df.groupby(['Admission Type', 'Gender'])['Billing Amount'].min()
.sort_values()

# Creating subplots
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plotting the highest billing amounts
```

```
max_billing.plot(kind='bar', ax=axes[0], color='lightblue', edgecolor='black')
axes[0].set_title('Highest Billing Amount by Admission Type and Gender')
axes[0].set_xlabel('Admission Type & Gender')
axes[0].set_ylabel('Highest Billing Amount')
axes[0].tick_params(axis='x', rotation=45)

# Plotting the lowest billing amounts
min_billing.plot(kind='bar', ax=axes[1], color='lightcoral', edgecolor='black')
axes[1].set_title('Lowest Billing Amount by Admission Type and Gender')
axes[1].set_xlabel('Admission Type & Gender')
axes[1].set_ylabel('Lowest Billing Amount')
axes[1].tick_params(axis='x', rotation=45)

# Adjust layout to avoid overlap
plt.tight_layout()

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

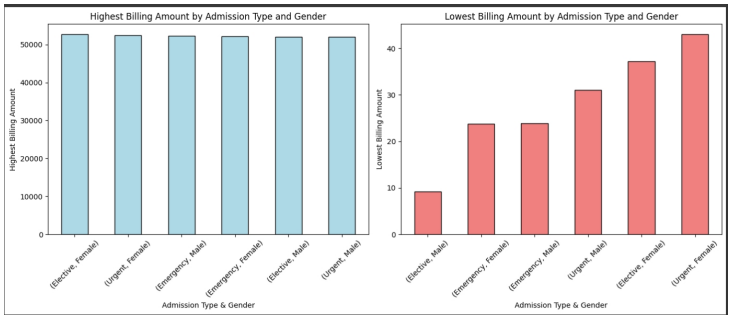

Figure 77: Highest and Lowest Billing Amount by Admission Type and Gender.

Insight: The data reveals that the highest billing amount is associated with Elective Female patients, with a billing amount of approximately 52,764. This is followed closely by Urgent Female patients at 52,373, indicating that females undergoing elective or urgent procedures tend to have higher associated costs compared to other categories. On the other hand, the lowest billing amounts are observed in Elective Male patients with a billing amount of 9.24, followed by Emergency Female patients at 23.73. These lower amounts may reflect less complex or less expensive treatment requirements for males in elective procedures and for females in emergency cases, suggesting differences in treatment intensity or patient demographics across genders and admission types.

14.Admission Type Vs Medical Condition

```
 Group by 'Admission Type' and 'Medical Condition', then count the occurrences
condition_counts = df.groupby(['Admission Type', 'Medical Condition']).size()
.unstack()

# Plotting a bar graph
```

```
condition_counts.plot(kind='bar', stacked=True, figsize=(8, 5), cmap='Set3')

# Adding title and labels
plt.title('Admission Type vs Medical Condition', fontsize=14)
plt.xlabel('Admission Type', fontsize=12)
plt.ylabel('Count of Medical Conditions', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

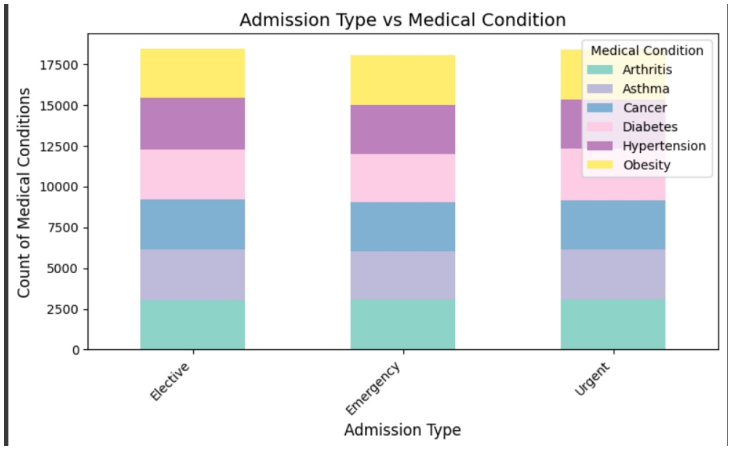

Figure 78: Admission Type Vs Medical Condition.

Insight: Elective Admissions

1. Conditions like Cancer, Hypertension, and Arthritis have a relatively high number of Elective admissions, which makes sense for conditions that can be planned in advance (e.g., surgeries for arthritis, cancer treatments like chemotherapy, or hypertension management).

2. Obesity and Asthma have lower elective admissions, which might be due to these conditions often requiring more urgent or reactive care (e.g., obesity-related complications or asthma exacerbations).

Emergency Admissions

1. Hypertension, Obesity, and Arthritis have relatively high Emergency admissions. This could suggest that complications or acute exacerbations related to these conditions may lead to unplanned emergency visits.

2. Asthma and Cancer also show a notable number of Emergency admissions, indicating that these conditions can present sudden, severe issues (like an asthma attack or a cancer-related emergency).

Urgent Admissions

1. Diabetes, Obesity, and Arthritis have higher Urgent admissions. This could indicate that patients with these conditions sometimes face situations that aren't immediately life-threatening but still require prompt attention.

2. Cancer and Asthma show relatively balanced figures across Emergency and Urgent, suggesting that some cancer-related issues or asthma exacerbations can be managed with urgent care, while others need emergency intervention.

15.Gender Vs Medical Condition

```
# Group by 'Gender' and 'Medical Condition', then count the occurrences
condition_counts = df.groupby(['Gender', 'Medical Condition']).size().unstack()

# Plotting a bar graph
condition_counts.plot(kind='bar', stacked=True, figsize=(8, 6), cmap='Set3')

# Adding title and labels
plt.title('Gender vs Medical Condition', fontsize=14)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Count of Medical Conditions', fontsize=12)
plt.xticks(rotation=0, ha='center')
plt.tight_layout()

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

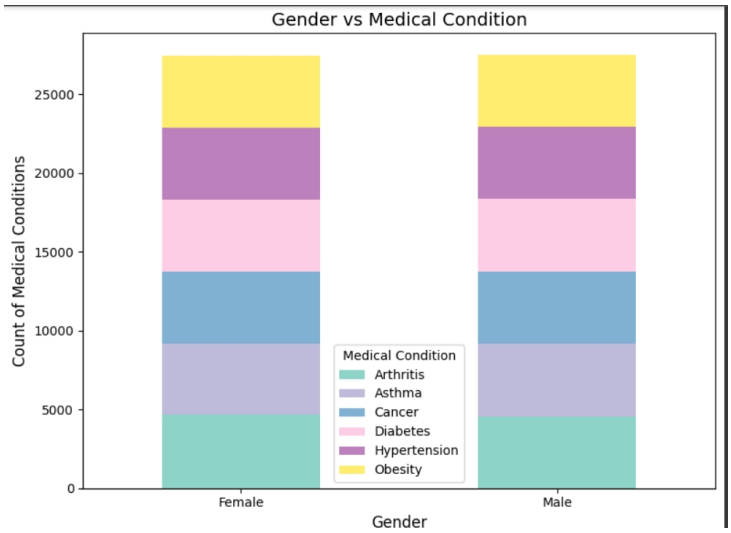

Figure 79: Gender Vs Medical Condition.

Insight: Gender Distribution Across Medical Conditions

1. Arthritis: There are more female admissions (4642) than male admissions (4576),which could indicate that arthritis is more prevalent in women in this dataset.

2. Asthma: There are slightly more male admissions (4584) than female admissions (4511), which could suggest that asthma may be more common or acute in males in this dataset.

3. Cancer: There is a slight gender balance in cancer admissions, with more females (4566) than males (4574). However, this difference is relatively small, so gender doesn't seem to significantly affect cancer admissions in this case. Diabetes: Both genders have similar numbers of admissions (4609 for females, 4607 for males). Diabetes seems to be equally prevalent across genders in this dataset.

4. Hypertension: This condition also shows a very similar gender distribution, with females (4569) and males (4582) showing near-equal admissions. Hypertension may not be strongly gender-biased in terms of hospital admissions.

5. Obesity: Obesity shows a similar trend to hypertension and diabetes, with almost identical admission counts (4573 for both males and females). This suggests that obesity affects both genders similarly in terms of hospital admissions.

16.Age Vs Gender

```
# Group by 'Age group' and 'Admission Type', then count the occurrences
age_admission_counts = df.groupby(['Age group', 'Admission Type']).size().unstack()

# Plotting a bar graph
age_admission_counts.plot(kind='bar', stacked=True, figsize=(10, 6), cmap='Set3')

# Adding title and labels
plt.title('Age Group vs Admission Type', fontsize=14)
plt.xlabel('Age Group', fontsize=12)
plt.ylabel('Count of Admission Types', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

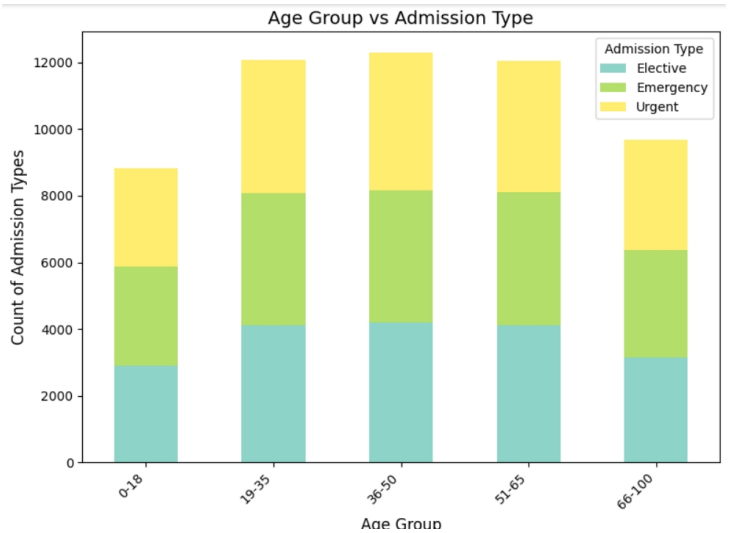

Figure 80: Age Vs Gender.

Insight: Elective Admissions:

1. The number of Elective admissions increases with age. The 66-100 age group has the highest number of Elective admissions (5316), followed by the 51-65 group (4215), and so on.

2. Elective admissions are generally planned procedures, often related to non-urgent surgeries, treatments, or check-ups. Older age groups may have more elective procedures such as joint replacements, cataract surgeries, or other age-related treatments that are planned in advance.

3. The 0-18 age group has the lowest number of Elective admissions (304), which is expected since younger individuals are less likely to need scheduled surgeries or treatments compared to older adults.

Emergency Admissions:
Emergency admissions are relatively consistent across age groups, but they increase slightly in older age groups. The 66-100 age group has the highest number of Emergency admissions (5359), followed closely by 51-65 (4021). Older adults are more likely to experience acute health conditions that require emergency care, such as heart attacks, strokes, respiratory issues, or accidents. In contrast, younger individuals (especially those in the 0-18 age group) tend to have fewer emergency admissions.

Urgent Admissions:
The Urgent category follows a similar pattern to Emergency admissions. Older age groups tend to have more Urgent admissions compared to younger ones. The 66-100 group has the highest number of Urgent admissions (5421), followed by the 51-65 group (4062). Urgent admissions are typically situations that require quick attention but are not immediately life-threatening (e.g., a non-severe heart attack, complications from chronic diseases). This aligns with the increase in urgent admissions with age, likely due to chronic health conditions becoming more prevalent.

17.Age Group Vs Medical Condition

```
# Group by 'Age group' and 'Medical Condition', then count the occurrences
age_condition_counts = df.groupby(['Age group', 'Medical Condition']).size()
.unstack()

# Plotting a stacked bar chart
age_condition_counts.plot(kind='bar', stacked=True, figsize=(8, 6), cmap='Set3')

# Adding title and labels
plt.title('Age Group vs Medical Condition', fontsize=14)
plt.xlabel('Age Group', fontsize=12)
plt.ylabel('Count of Medical Conditions', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

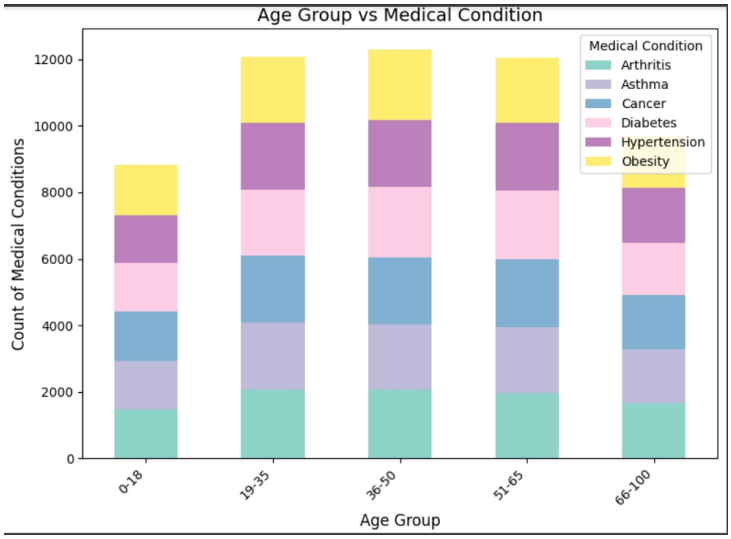

Figure 81: Age Group Vs Medical Condition.

Insight: Young Age Group (0–18)

1. All medical conditions have relatively low counts compared to older age groups.
2. Asthma (150 cases) and Obesity (158 cases) are slightly more common among children and adolescents compared to other conditions.
3. Chronic conditions like Arthritis (149) and Diabetes (140) are rare but present.

Young Adults (19–35):

1. There is a sharp increase in cases for all conditions compared to the 0–18 age group.

2. Counts for Cancer (2271) and Obesity (2267) are highest, but other conditions (like Asthma, Arthritis) are similar in numbers.

Middle Age (36–50):

1. There is a slight decline in Asthma (1983) and Cancer (1986) compared to the younger group.
2. Diabetes (2094) and Hypertension (2011) show noticeable growth, indicating these conditions emerge strongly during middle age.

Older Adults (51–65):

1. Cases for most conditions remain high or slightly increase: Cancer (2053), Diabetes (2087), and Hypertension (2048) are the dominant conditions.
2. Obesity (2103) remains prevalent, indicating sustained lifestyle factors.

Senior Age Group (66–100):

1. Arthritis (2716), Asthma (2718), and Hypertension (2715) are at their highest levels.
2. Diabetes (2676) and Cancer (2684) remain prevalent, while Obesity (2587) sees a slight decline.

18. Insurance Provider Vs Gender

```
# Group by 'Age group' and 'Medical Condition', then count the occurrences
age_condition_counts = df.groupby(['Age group', 'Medical Condition']).size()
.unstack()

# Plotting a stacked bar chart
age_condition_counts.plot(kind='bar', stacked=True, figsize=(8, 6), cmap='Set3')

# Adding title and labels
plt.title('Age Group vs Medical Condition', fontsize=14)
plt.xlabel('Age Group', fontsize=12)
plt.ylabel('Count of Medical Conditions', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

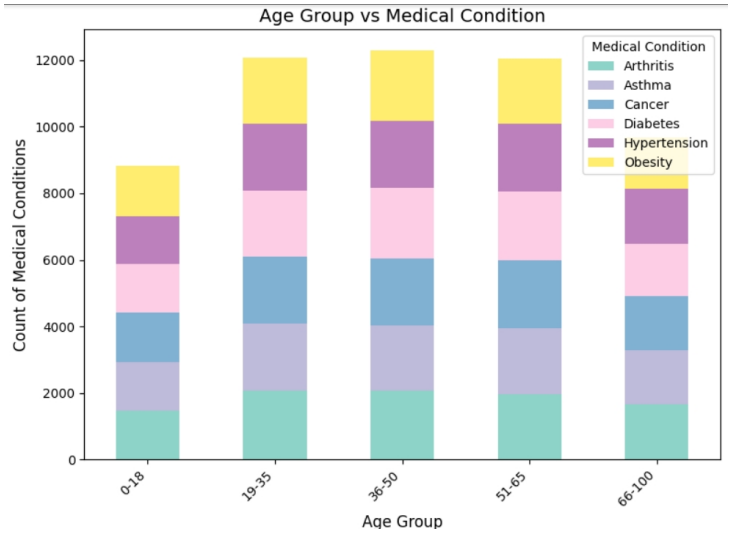

Figure 82: Insurance Provider Vs Gender.

Insight: Provider-Specific Gender Trends

1. Aetna: Slightly more males (5492) than females (5330) are enrolled with Aetna, but the difference is small.

2. Blue Cross: Here, females (5515) slightly outnumber males (5437). This could be related to certain Blue Cross policies or coverage options that appeal more to women, or it could reflect broader regional or demographic factors that affect insurance enrollment.

3. Cigna: This provider has the largest enrollment for both genders (5593 females and 5546 males), suggesting it is a popular choice overall, with no major gender-based preference.

4. Medicare: For Medicare, the numbers are almost identical between males (5512) and females (5527), indicating that both genders are similarly represented in this government-run program.

5. UnitedHealthcare: The gender gap is quite small (5505 females vs. 5509 males), showing near equality in enrollment between males and females.

Patterns of Insurance Enrollment

1. Cigna has a slightly higher number of females enrolled than any other insurer, which may reflect particular services or coverage options offered by Cigna that attract more female patients (e.g., women's health services, maternity care, preventive services, etc.).

2. Blue Cross has more females enrolled than males, which could indicate a higher preference among women for the provider in certain regions or demographics.

   Aetna and UnitedHealthcare show a more balanced gender distribution. Medicare is fairly gender-neutral in terms of enrollment, with both genders showing almost equal representation.

19.Duration of Stay vs. Billing Amount

```
# Group by 'Duration' and calculate the mean of 'Billing Amount'
avg_billing_by_duration = df.groupby('Duration')['Billing Amount'].mean()

# Get the 10 highest and lowest average billing amounts by duration
highest_avg_billing = avg_billing_by_duration.nlargest(10)
lowest_avg_billing = avg_billing_by_duration.nsmallest(10)

# Create subplots to display both highest and lowest on the same figure
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Plotting the highest average billing amounts
highest_avg_billing.plot(kind='bar', ax=axes[0], color='green', edgecolor='black')
axes[0].set_title('Top 10 Highest Average Billing Amounts by Duration', fontsize=14)
axes[0].set_xlabel('Duration of Stay (Days)', fontsize=12)
axes[0].set_ylabel('Average Billing Amount', fontsize=12)

# Plotting the lowest average billing amounts
lowest_avg_billing.plot(kind='bar', ax=axes[1], color='red', edgecolor='black')
axes[1].set_title('Top 10 Lowest Average Billing Amounts by Duration', fontsize=14)
axes[1].set_xlabel('Duration of Stay (Days)', fontsize=12)
axes[1].set_ylabel('Average Billing Amount', fontsize=12)

# Display the plots
plt.tight_layout()
plt.show()
```

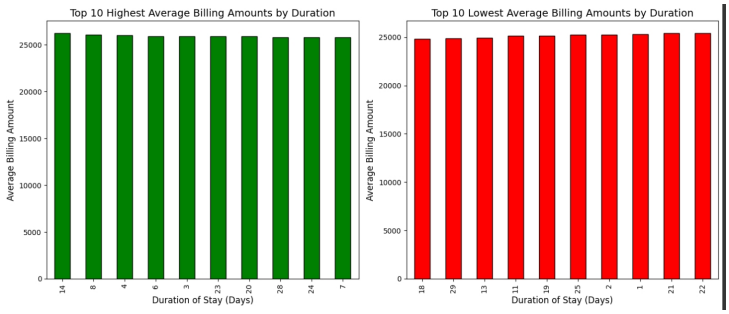

Figure 83: Duration of Stay vs. Billing Amount.

Insight: Trends in Billing Amount by Duration

1. The billing amounts generally decrease as the duration increases, though the changes are not drastic.

2. Shortest durations (e.g., 3 days) have slightly higher billing amounts (25918.33) compared to longer durations (e.g., 28 days), which have the lowest billing amount (25810.21).

3. Billing amounts appear to slightly decrease with the increase in duration, suggesting that longer durations might be associated with lower per-day costs, or there could be a discount or different pricing model in place for longer durations.

20.Day of the week vs Admission type

```
# Group by 'day_admission' and 'Admission Type' and count occurrences
day_admission_counts = df.groupby(['day_admission', 'Admission Type']).size()
.unstack()

# Plotting a stacked bar chart
plt.figure(figsize=(10, 6))
day_admission_counts.plot(kind='bar', stacked=True, cmap='Set2', edgecolor='black')

# Adding title and labels
plt.title('Day of the Week vs Admission Type', fontsize=14)
plt.xlabel('Day of the Week', fontsize=12)
plt.ylabel('Count of Admission Types', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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

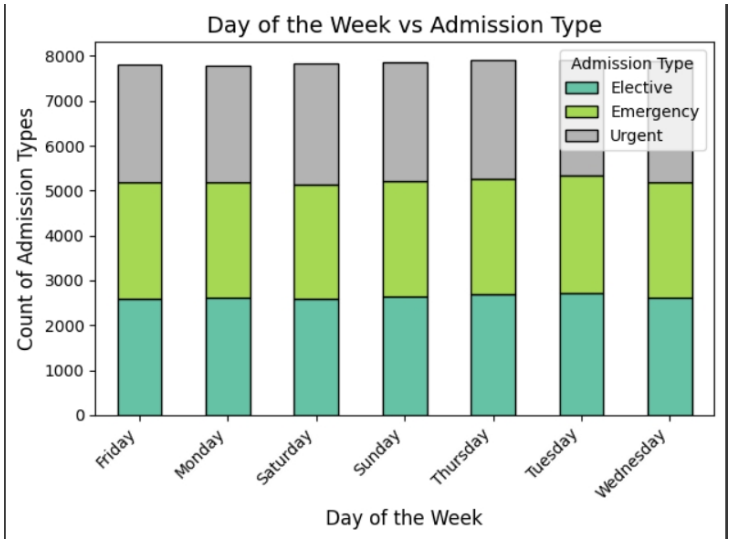

Figure 84: Day of the week vs Admission type.

Insight:

1. Elective admissions seem to be highest on Tuesday (2711), followed by Thursday (2693).

2. Emergency admissions are fairly consistent but seem to peak on Tuesday (2633) and Friday (2598).

3. Urgent admissions show a peak on Saturday (2675), followed by Wednesday (2674).

21. Medical Condition Higher on Each Day

```
# Step 1: Count the occurrences of each Admission Type per day of the week
admission_counts = df.groupby(['day_admission', 'Admission Type']).size()
.reset_index(name='Count')

# Step 2: For each 'day_admission', get the row with the highest count
(most frequent Admission Type)
most_frequent_admission = admission_counts.loc[admission_counts
.groupby('day_admission')['Count'].idxmax()]

# Step 3: Plotting the result
plt.figure(figsize=(8, 6))

# Create a bar plot with the day_admission as the x-axis and
the most frequent Admission Type as the y-axis
plt.bar(most_frequent_admission['day_admission'],
most_frequent_admission['Admission Type'], color='skyblue')

# Add labels and title
plt.xlabel('Day of the Week')
plt.ylabel('Most Frequent Admission Type')
plt.title('Most Frequent Admission Type by Day of the Week')
plt.xticks(rotation=45, ha='right')  # Rotate the x-axis labels for readability

# Display the plot
plt.tight_layout()
plt.show()
```

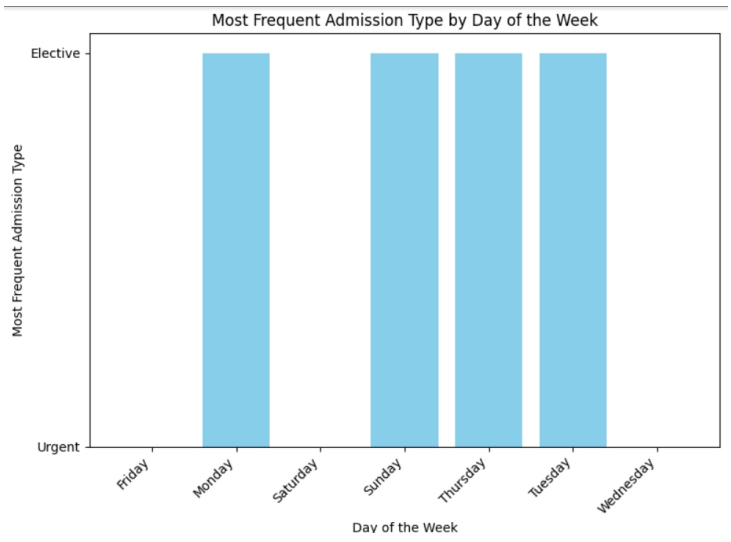

Figure 85: Medical Condition Higher on Each Day.

Insight:

1. Elective Admissions dominate on Monday, Thursday, Tuesday, and Sunday. These days might indicate a preference for planned or non-urgent procedures, possibly reflecting hospital or clinic schedules that prioritize elective surgeries or treatments on these days.

2. Urgent Admissions are most frequent on Friday, Saturday, and Wednesday. These days could indicate higher demand for emergency care or urgent treatments. This might suggest more emergency cases occurring at the end of the week (Friday, Saturday), or on mid-week days (Wednesday), potentially due to factors like accidents, sudden illnesses, or weekend-related emergencies.

Days with the Highest Admissions for Urgent Cases

1. Friday, Saturday, and Wednesday stand out as the days with the most Urgent admissions.

2. Friday and Saturday might suggest a weekend rush, possibly due to accidents, injuries, or health issues arising at the end of the workweek and during weekend activities.

3. Wednesday might reflect mid-week pressure on emergency services, with people seeking urgent care during the middle of the week.

Days with the Highest Admissions for Elective Cases

1. Monday, Thursday, and Tuesday are the days where Elective admissions are the most common.

2. These are likely days when hospitals or clinics schedule planned surgeries or treatments. Elective admissions require more pre-scheduling and may occur earlier in the week or when medical staff and facilities are less overloaded after the weekend.

3. Sunday is also a significant day for Elective admissions, possibly due to scheduling of minor procedures or consultations on the weekend.

Conclusion

Based on the detailed exploration of the healthcare dataset, the following conclusions have been drawn:

1. Patient Demographics and Admission Trends Most patients fall within the middle-aged and older age groups (36–65 years), with the 66–100 age group being the most common. Pediatric cases (under 18 years) are rare but tend to involve higher costs, indicating the complexity and specialization of their care. Male patients generally incur slightly higher billing amounts compared to female patients, which may reflect differences in treatment patterns or healthcare utilization.

2. Billing Insights Elective admissions generate the highest total billing, reflecting the planned nature of these treatments. Emergency admissions have the highest average stay duration, aligning with the complex and urgent nature of these cases. Specialized rooms (e.g., Rooms 139 and 427) have significantly higher billing averages, suggesting they cater to high-complexity or VIP cases. The total billing amount for the dataset exceeds $1.4 billion, showcasing the dataset's representation of a broad healthcare system.

3. Medical and Insurance Trends Chronic conditions like arthritis and diabetes dominate medical cases, emphasizing the need for preventive care and chronic disease management. Insurance providers such as Cigna and Medicare are the most common, with notable variations in billing patterns across providers. Certain doctors and hospitals contribute disproportionately to total billing, likely due to their specialization in high-cost or high-volume cases.

4. Seasonal and Temporal Patterns Admissions peak in the summer months (July and August) and drop significantly in February, suggesting seasonal healthcare trends. Discharges follow a similar pattern, with Fridays seeing the highest number of discharges, likely to clear hospital capacity before weekends.

5. Doctor and Hospital Performance Doctors like Michael Smith and Kathleen Griffin lead in both patient counts and billing amounts, indicating their pivotal role in the system. Hospitals such as LLC Smith are frequented the most, with some providing specialized or high-cost care.

6. Stay Durations The average hospital stay is around 15.4 days, with extended stays (above average) accounting for nearly half the dataset. Emergency admissions typically require the longest stays, consistent with the need for intensive care.

Suggestions and Recommendations

1. Implement Predictive Analytics for Resource Management Utilize data-driven models to predict patient admission trends and optimize resource allocation. This includes managing staff shifts, equipment availability, and hospital bed capacity to minimize operational bottlenecks.

2. Focus on Preventive Healthcare Develop initiatives aimed at preventing diseases through early detection and education. Outreach programs, regular health checkups, and vaccination campaigns can significantly reduce the strain on healthcare facilities.

3. Improve Post-Discharge Care Establish structured follow-up programs for discharged patients to ensure adherence to treatment plans, reducing readmissions and improving recovery rates.

4. Enhance Patient Experience Use patient feedback to refine processes such as appointment scheduling, in-hospital navigation, and discharge procedures. Personalized care and improved communication can boost satisfaction levels.

5. Integrate Advanced Health Informatics Systems Adopt electronic health records (EHRs) and real-time monitoring tools to streamline data collection and support better decision-making. Ensure interoperability between systems for seamless information flow.

6. Strengthen Cost Efficiency Measures Analyze expenditure patterns to identify high-cost areas and inefficiencies. Implement strategies such as bulk purchasing of medical supplies and energy-saving technologies to reduce operational costs.

7. Promote Equity in Healthcare Access Address disparities by increasing the availability of services in underserved areas, offering financial assistance programs, and tailoring services to meet the cultural and linguistic needs of diverse populations.

8. Train Healthcare Professionals Continuously Offer ongoing training and certification programs for medical staff to stay updated on the latest practices and technologies, ensuring high standards of care.

9. Develop Community-Based Health Programs Partner with local organizations to provide accessible health education and resources. These programs can focus on managing chronic diseases and promoting healthier lifestyles.

10. Enhance Emergency Response Systems Strengthen emergency care protocols and invest in infrastructure like ambulance services and telemedicine for rural and remote areas.