

Healthcare Data Analysis: Uncovering Patient Patterns and Medical Trends

Using NumPy, Pandas, and Matplotlib for Insights and Visualization

Subham Choudhary

Healthcare Data Analysis Project

I recently completed an extensive data analysis project using Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn. This project aimed to uncover patterns and trends within a healthcare dataset, providing valuable insights into medical conditions, billing amounts, and hospital admissions. Utilizing these powerful frameworks, I was able to efficiently clean, preprocess, and analyze the data, and create impactful visualizations to communicate the findings.

Data Set = [Link](#)

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

Load the dataset

Load

```
[3] file_path = '/content/healthcare_dataset.xlsx'
data = pd.read_excel(file_path)
```

```
[4] print(data.head(5))
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	\
0	Bobby Jackson	30	Male	B-	Cancer	2024-01-31	
1	Leslie Terry	62	Male	A+	Obesity	2019-08-20	
2	Danny Smith	76	Female	A-	Obesity	2022-09-22	
3	Andrew Watts	28	Female	O+	Diabetes	2020-11-18	
4	Adrienne Bell	43	Female	AB+	Cancer	2022-09-19	

	Doctor	Hospital	Insurance Provider	\
0	Matthew Smith	Sons and Miller	Blue Cross	
1	Samantha Davies	Kim Inc	Medicare	
2	Tiffany Mitchell	Cook PLC	Aetna	
3	Kevin Wells	Hernandez Rogers and Vang,	Medicare	
4	Kathleen Hanna	White-White	Aetna	

Clean the data

Clean the data - Handle missing values, incorrect data types

```
[5] data['Date of Admission'] = pd.to_datetime(data['Date of Admission'], errors='coerce')
data['Discharge Date'] = pd.to_datetime(data['Discharge Date'], errors='coerce')
data['Billing Amount'] = pd.to_numeric(data['Billing Amount'], errors='coerce')
```

Fill missing values with appropriate methods

missing values with appropriate methods

```
data.fillna({
    'Date of Admission': pd.Timestamp('today'),
    'Discharge Date': pd.Timestamp('today'),
    'Billing Amount': 0
}, inplace=True)
```

Average Billing

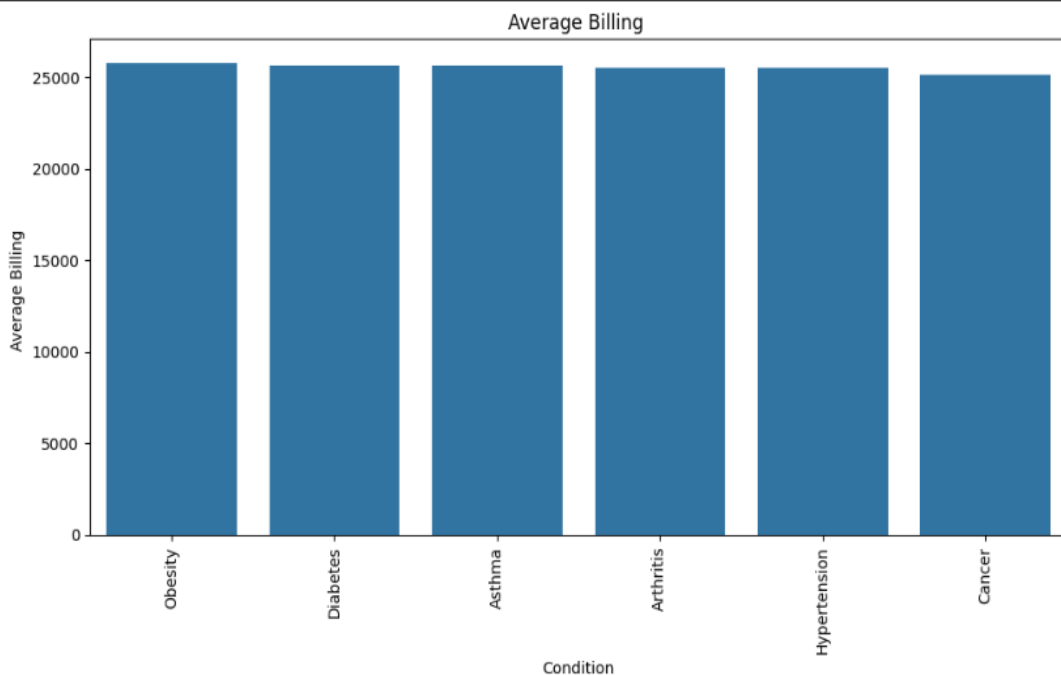
Analysis

Average billing amount

```
avg_bil = data.groupby('Medical Condition')['Billing Amount'].mean().sort_values(ascending=False)
avg_bil
```

```
Medical Condition
Obesity          25805.971259
Diabetes          25638.405577
Asthma           25635.249359
Arthritis        25497.327056
Hypertension     25497.095761
Cancer           25161.792707
Name: Billing Amount, dtype: float64
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=avg_billing_by_condition.index, y=avg_billing_by_condition.values)
plt.title('Average Billing ')
plt.xlabel('Condition')
plt.ylabel('Average Billing ')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



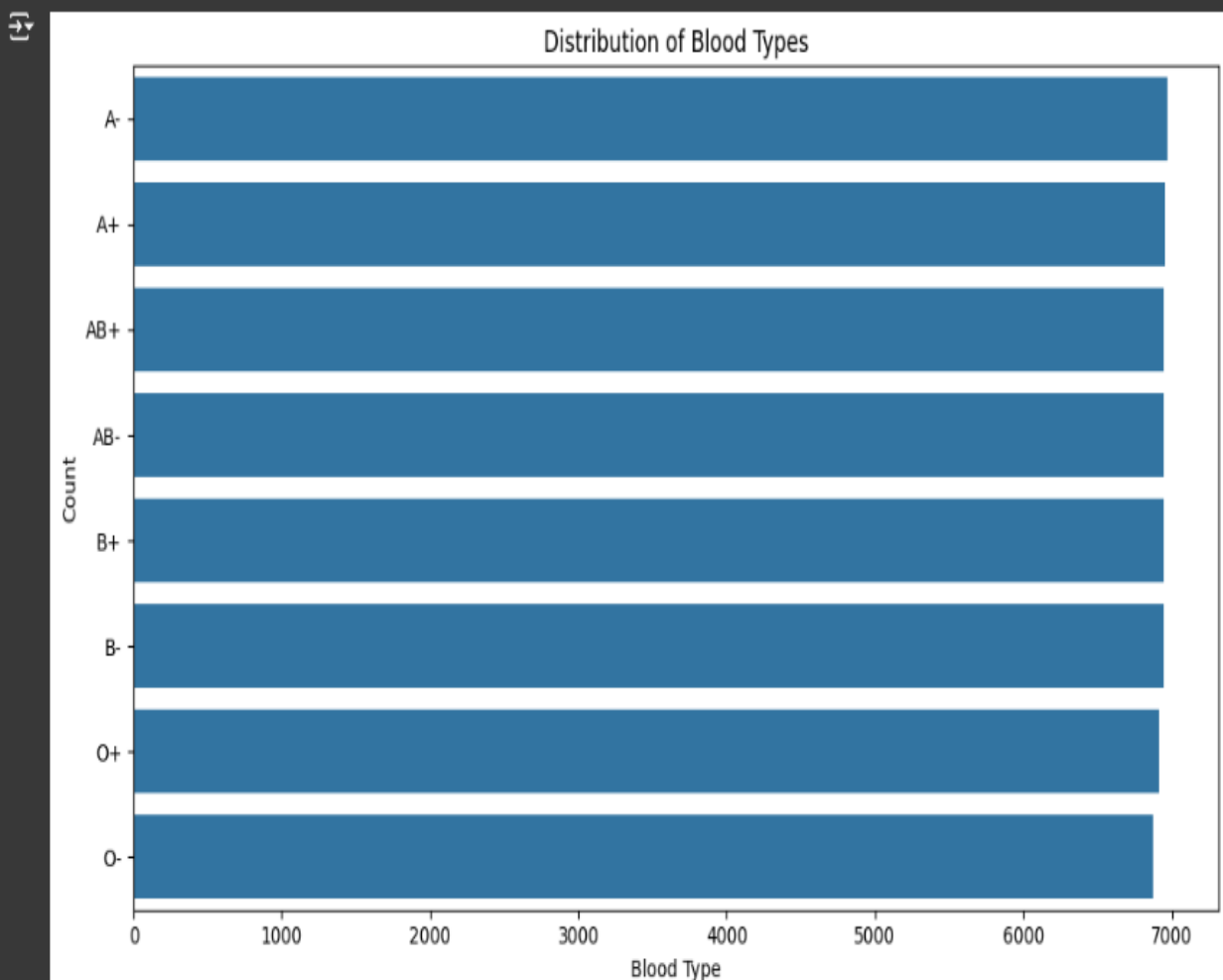
Blood Type

blood types

```
▶ blood_ty= data['Blood Type'].value_counts()  
blood_ty
```

```
⇄ Blood Type  
A-      6969  
A+      6956  
AB+     6947  
AB-     6945  
B+      6945  
B-      6944  
O+      6917  
O-      6877  
Name: count, dtype: int64
```

```
▶ plt.figure(figsize=(10, 6))  
sns.countplot(data['Blood Type'], order=blood_ty.index)  
plt.title('Distribution of Blood Types')  
plt.xlabel('Blood Type')  
plt.ylabel('Count')  
plt.tight_layout()  
plt.show()
```



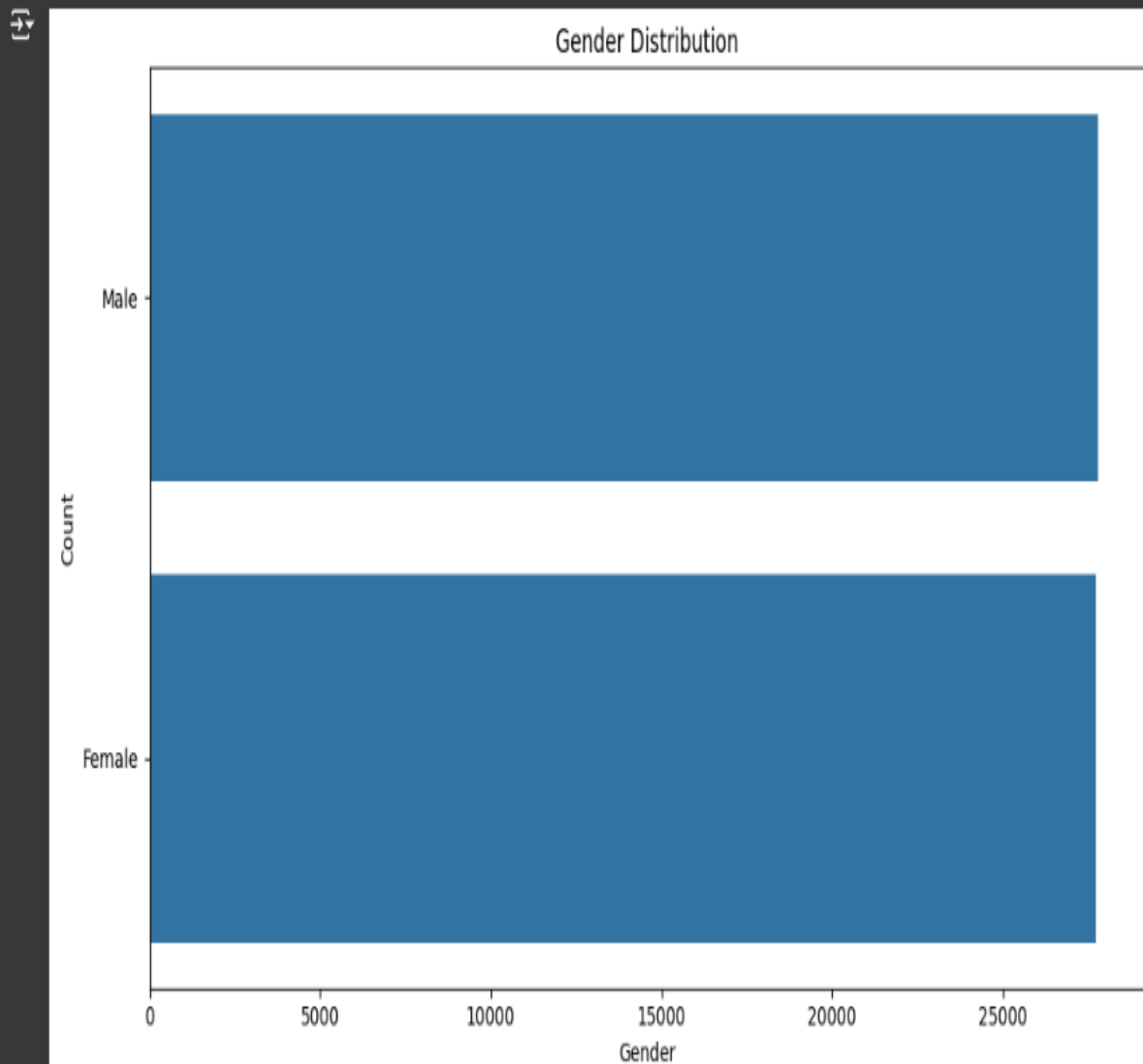
Gender Distribution

Gender distribution

```
[10] gender_di = data['Gender'].value_counts()  
gender_di
```

```
Gender  
Male      27774  
Female    27726  
Name: count, dtype: int64
```

```
plt.figure(figsize=(10, 6))  
sns.countplot(data['Gender'], order=gender_di.index)  
plt.title('Gender Distribution')  
plt.xlabel('Gender')  
plt.ylabel('Count')  
plt.tight_layout()  
plt.show()
```



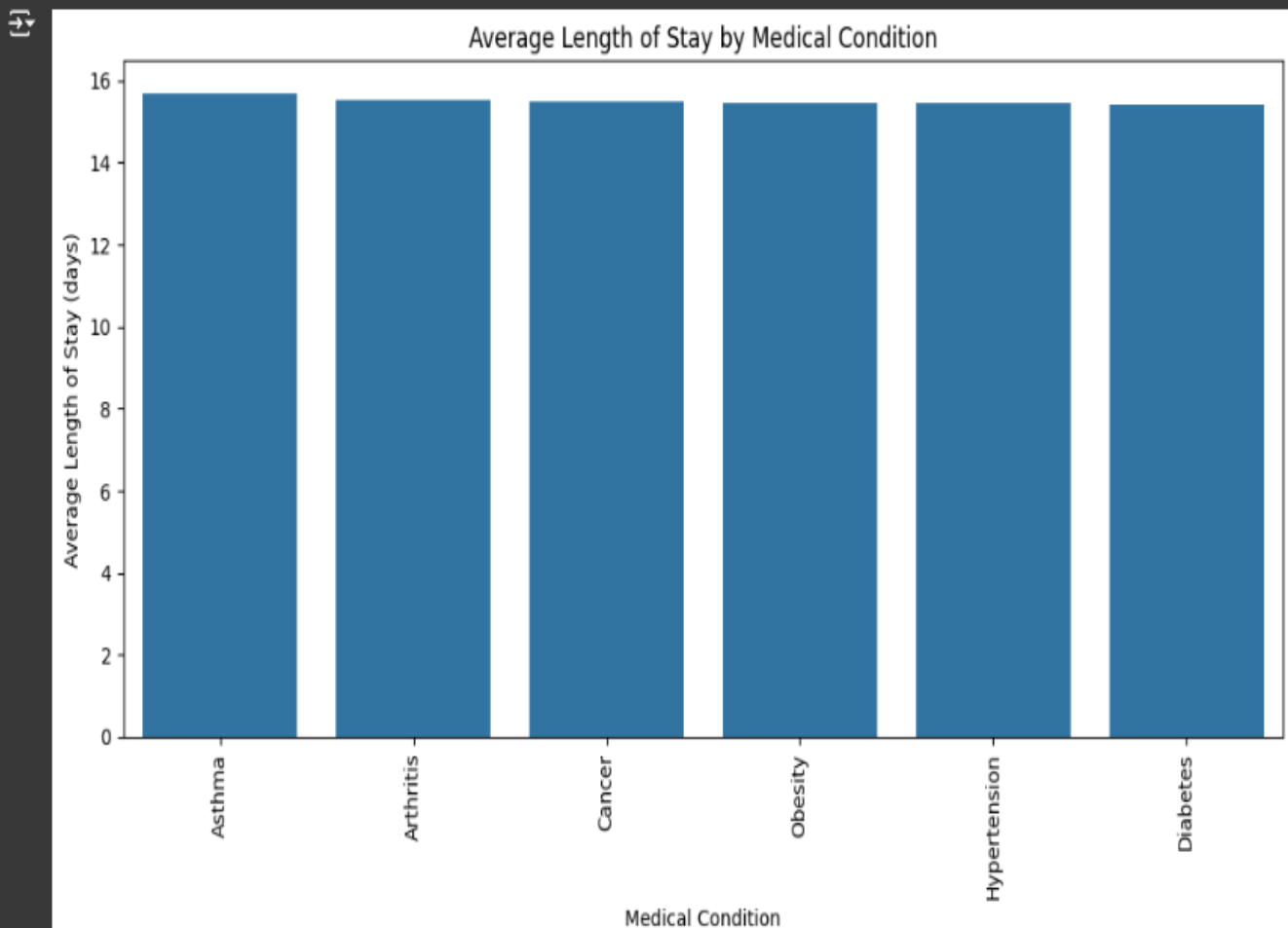
Medical Condition

Average stay by medical condition

```
[13] data['Length of Stay'] = (data['Discharge Date'] - data['Date of Admission']).dt.days
     avg_le = data.groupby('Medical Condition')['Length of Stay'].mean().sort_values(ascending=False)
     avg_le
```

```
Medical Condition
Asthma          15.696570
Arthritis       15.517404
Cancer          15.495827
Obesity         15.464305
Hypertension    15.458626
Diabetes        15.422936
Name: Length of Stay, dtype: float64
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=avg_le.index, y=avg_le.values)
plt.title('Average Length of Stay by Medical Condition')
plt.xlabel('Medical Condition')
plt.ylabel('Average Length of Stay (days)')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



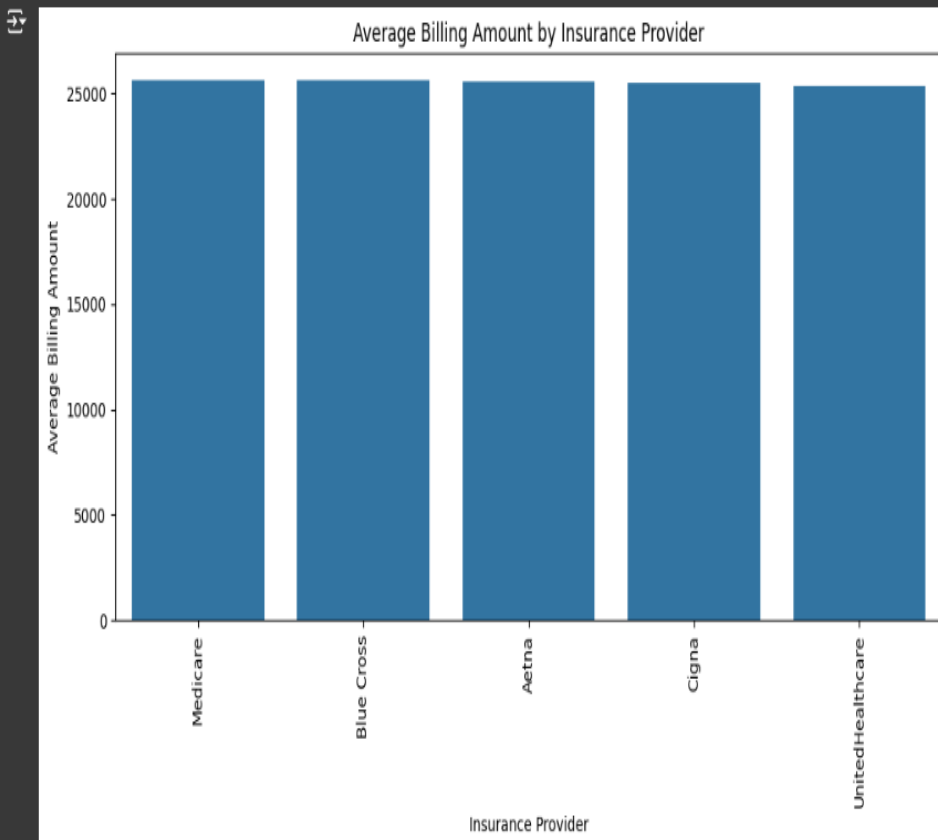
Average amount Provided by Insurance

Average billing amount by insurance provider

```
[22] avg_insurance = data.groupby('Insurance Provider')['Billing Amount'].mean().sort_values(ascending=False)
      avg_insurance
```

```
Insurance Provider
Medicare          25615.990508
Blue Cross        25613.011503
Aetna              25553.294506
Cigna              25525.766314
UnitedHealthcare  25389.172390
Name: Billing Amount, dtype: float64
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=avg_billing_by_insurance.index, y=avg_billing_by_insurance.values)
plt.title('Average Billing Amount by Insurance Provider')
plt.xlabel('Insurance Provider')
plt.ylabel('Average Billing Amount')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



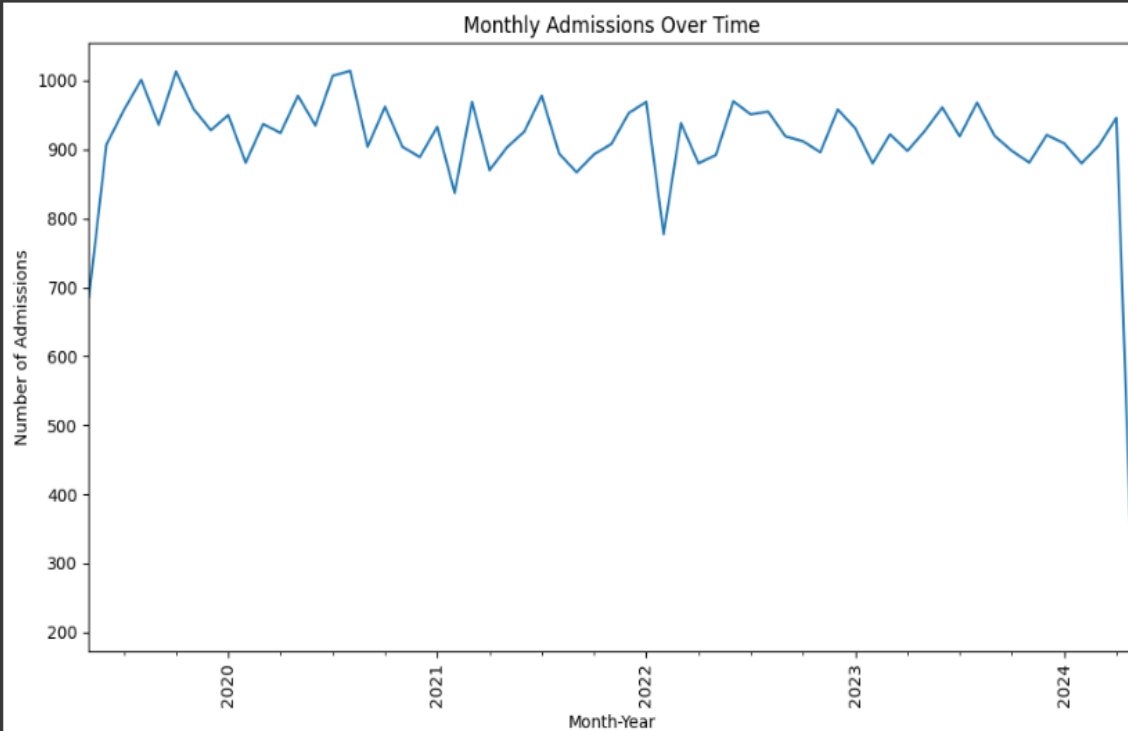
Monthly Admission

Monthly admissions over time

```
[23] data['Month-Year'] = data['Date of Admission'].dt.to_period('M')
monthly_admissions = data['Month-Year'].value_counts().sort_index()
monthly_admissions
```

```
Month-Year
2019-05      686
2019-06      907
2019-07      957
2019-08     1001
2019-09      936
...
2024-01      909
2024-02      880
2024-03      906
2024-04      946
2024-05      213
Freq: M, Name: count, Length: 61, dtype: int64
```

```
plt.figure(figsize=(10, 6))
monthly_admissions.plot(kind='line')
plt.title('Monthly Admissions Over Time')
plt.xlabel('Month-Year')
plt.ylabel('Number of Admissions')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



Medical Condition – By Gender

Medical condition distribution by gender

```
[25] condition_ge = data.groupby(['Medical Condition', 'Gender']).size().unstack().fillna(0)  
condition_ge
```

	Gender	Female	Male
Medical Condition			
Arthritis		4686	4622
Asthma		4553	4632
Cancer		4602	4625
Diabetes		4651	4653
Hypertension		4612	4633
Obesity		4622	4609

```
plt.figure(figsize=(10, 6))  
condition_ge.plot(kind='bar', stacked=True)  
plt.title("Medical Condition Distribution by Gender")  
plt.xlabel('Medical Condition')  
plt.ylabel('Count')  
plt.xticks(rotation=90)  
plt.tight_layout()  
plt.show()
```

<Figure size 1000x600 with 0 Axes>

