EXPLORATORY DATA ANALYSIS OF HEALTHCARE DATASET

KEY INDICATORS

The Dataset includes patient demographics, medical issues, admission details, and billing info.

Key fields: age, gender, blood type, medical condition, admission/discharge dates, doctor, hospital, insurance, and billing amount.

Records admission type (emergency, urgent, elective), prescribed medications, and test results.

The data provides insights on healthcare resource use, treatment patterns, and billing trends by demographics.

KEY DATA ANALYSIS:

Data Blood Group by Amount

medical condition by Billing Amount

Age by Billing Amount

Billing by Gender

Count of Medical Condition

Average Billing Amount by Blood Type

Average Billing Amount by Admission Type

Average Billing Amount by Medical Condition

Average Billing by Gender

Distribution of Billing Amounts

In [35]:
```
# packages
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import datetime
from matplotlib import pyplot as plt
```

In [11]:
```
df = pd.read_csv('healthcare_dataset.csv')
```

In [ ]:
```
#DATA EXPLORATION
```

In [27]:
```
#First 4 rows of the data
df.head(5)
```

Out[27]:

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurar Provic
0	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller	B Cr
1	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc	Medic
2	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC	Aet
3	andrEw waTtS	28	Female	O+	Diabetes	2020-11-18	Kevin Wells	Hernandez Rogers and Vang,	Medic
4	adrIENNE bEll	43	Female	AB+	Cancer	2022-09-19	Kathleen Hanna	White-White	Aet

In [29]:
```
#data description
df.describe()
```

Out[29]:

	Age	Billing Amount	Room Number
count	55500.000000	55500.000000	55500.000000
mean	51.539459	25539.316097	301.134829
std	19.602454	14211.454431	115.243069
min	13.000000	-2008.492140	101.000000
25%	35.000000	13241.224652	202.000000
50%	52.000000	25538.069376	302.000000
75%	68.000000	37820.508436	401.000000
max	89.000000	52764.276736	500.000000

In [31]:
```
#data info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55500 entries, 0 to 55499
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Name                55500 non-null  object 
 1   Age                 55500 non-null  int64  
 2   Gender              55500 non-null  object 
 3   Blood Type          55500 non-null  object 
 4   Medical Condition   55500 non-null  object 
 5   Date of Admission   55500 non-null  object 
 6   Doctor              55500 non-null  object 
 7   Hospital            55500 non-null  object 
 8   Insurance Provider  55500 non-null  object 
 9   Billing Amount      55500 non-null  float64
 10  Room Number         55500 non-null  int64  
 11  Admission Type      55500 non-null  object 
 12  Discharge Date      55500 non-null  object 
 13  Medication          55500 non-null  object 
 14  Test Results        55500 non-null  object 
dtypes: float64(1), int64(2), object(12)
memory usage: 6.4+ MB
```

In [35]:
```
#data size
df.size
```

Out[35]:
```
832500
```

In [39]:
```
#data number of rows and columns
df.shape
```

Out[39]:
```
(55500, 15)
```

In [45]:
```
#column name
df.columns
```

Out[45]:
```
Index(['Name', 'Age', 'Gender', 'Blood Type', 'Medical Condition',
       'Date of Admission', 'Doctor', 'Hospital', 'Insurance Provider',
       'Billing Amount', 'Room Number', 'Admission Type', 'Discharge Date',
       'Medication', 'Test Results'],
      dtype='object')
```

In [314…
```
df['Age'].mean()
```

Out[314…
```
51.53945945945946
```

In [316…
```
df['Billing Amount'].mean()
```

Out[316…
```
25539.316097211795
```

In [13]:
```
# Data NA
df.isna()
```

Out[13]:

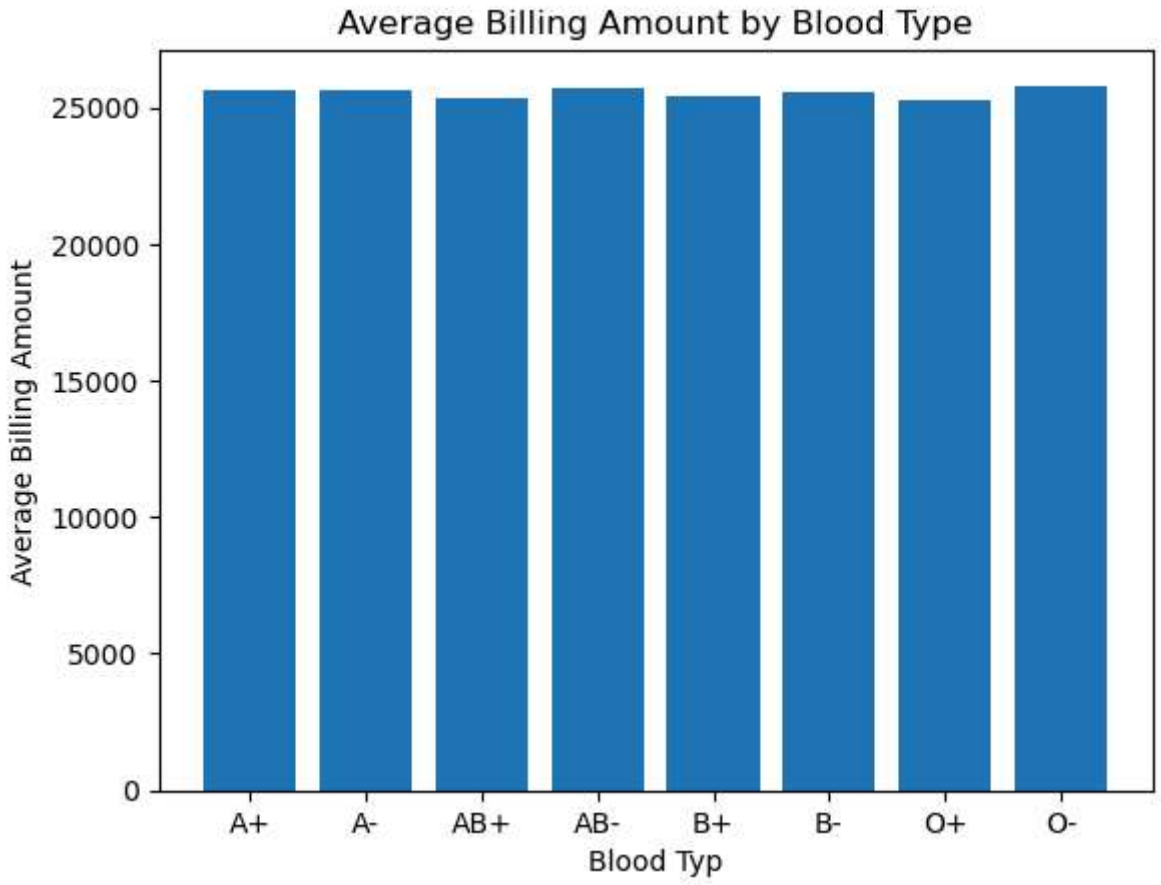
	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insuranc Provide
0	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	.
55495	False	False	False	False	False	False	False	False	Fals
55496	False	False	False	False	False	False	False	False	Fals
55497	False	False	False	False	False	False	False	False	Fals
55498	False	False	False	False	False	False	False	False	Fals
55499	False	False	False	False	False	False	False	False	Fals

55500 rows × 15 columns

In [256…

```
#Data Blood Group by Billing Amount
group_data = df.groupby('Blood Type')['Billing Amount'].mean().reset_index()
print(group_data)

#Bar plot
plt.bar(group_data['Blood Type'], group_data['Billing Amount'])
plt.xlabel('Blood Typ')
plt.ylabel('Average Billing Amount')
plt.title('Average Billing Amount by Blood Type')
plt.show()
```

```
  Blood Type  Billing Amount
0         A+    25664.566404
1         A-    25595.024701
2        AB+    25361.458784
3        AB-    25694.933091
4         B+    25429.723237
5         B-    25524.424636
6         O+    25249.740696
7         O-    25795.657833
```



In [130…
```
# Data by medical condition
groupby_condition = df.groupby('Medical Condition')['Billing Amount'].mean().res
print(groupby_condition)
```

	Medical Condition	Billing Amount
0	Arthritis	25497.327056
1	Asthma	25635.249359
2	Cancer	25161.792707
3	Diabetes	25638.405577
4	Hypertension	25497.095761
5	Obesity	25805.971259

In [ ]:
```
#Viusal by medical condition
```

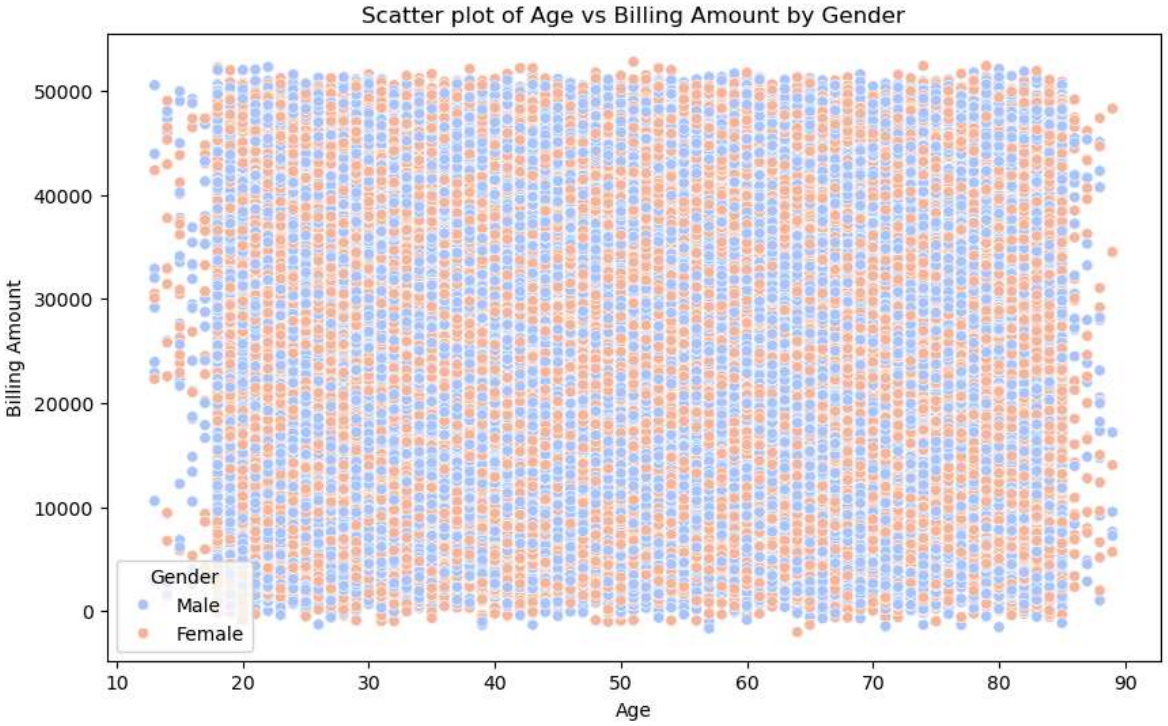
In [168…
```
# Extract of 'Age' 'Room number' and 'Billing_Amount' columns
x = df[['Age', 'Room Number', 'Billing Amount']]
print(x.head(5))
```

	Age	Room Number	Billing Amount
0	30	328	18856.281306
1	62	265	33643.327287
2	76	205	27955.096079
3	28	450	37909.782410
4	43	458	14238.317814

In [31]:
```
# Extract of 'Age' 'Room number' and 'Billing_Amount' columns
x = df[['Age', "Gender", 'Billing Amount']]
print(x.head(5))

# BILLING PER AGE
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Age', y='Billing Amount', hue='Gender', palette='coo
plt.title('Scatter plot of Age vs Billing Amount by Gender')
```

```
plt.xlabel('Age')
plt.ylabel('Billing Amount')
plt.show()
```

```
   Age  Gender  Billing Amount
0   30    Male    18856.281306
1   62    Male    33643.327287
2   76  Female    27955.096079
3   28  Female    37909.782410
4   43  Female    14238.317814
```



In [180…

```
#BILLING BY GENDER
groupby_gender = df.groupby('Gender')['Billing Amount'].mean().reset_index()
print(groupby_gender)
```

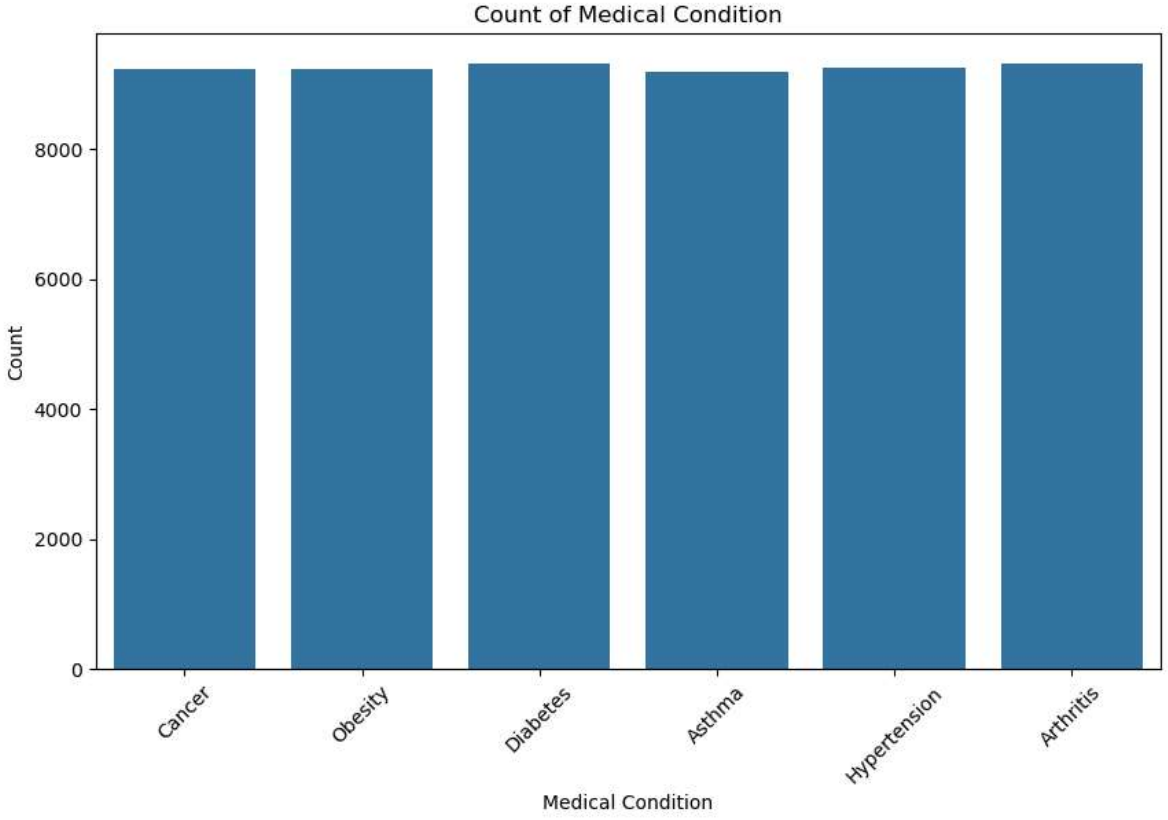
```
   Gender  Billing Amount
0  Female    25470.652958
1    Male    25607.860571
```

In [182…

```
# DISTRIBUTION BY HOSPITAL
groupby_hospital = df.groupby('Hospital')['Billing Amount'].mean().reset_index()
print(groupby_hospital)
```

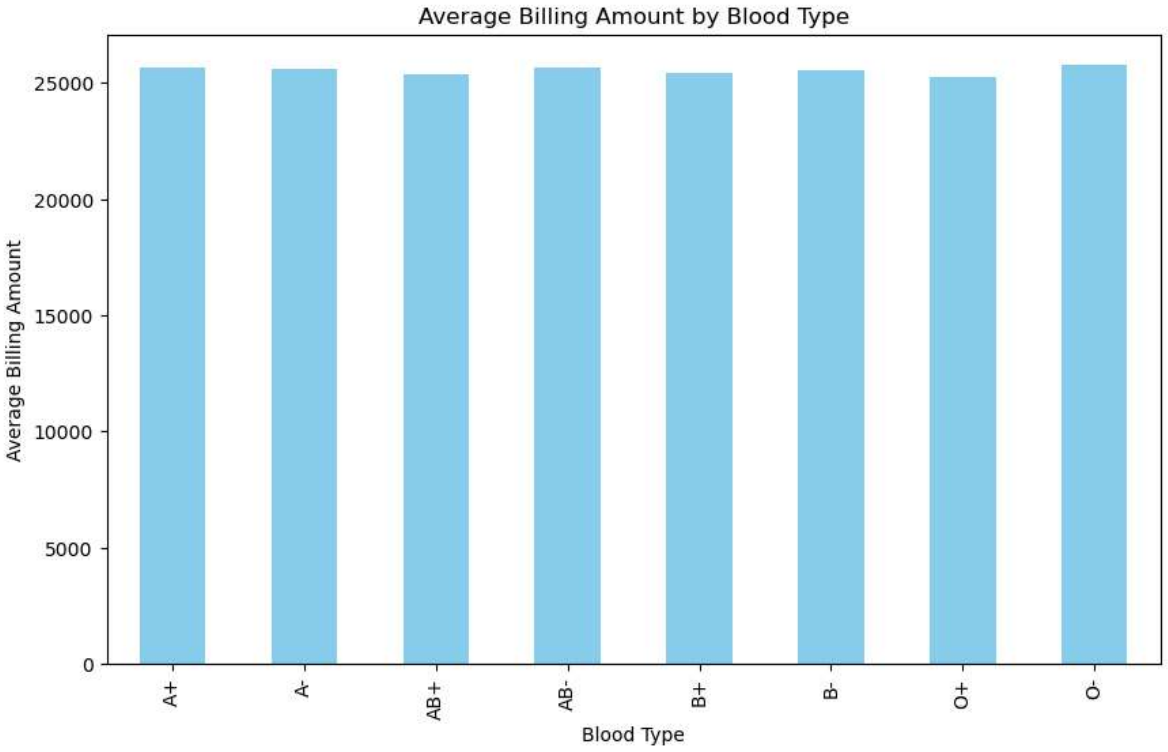
```
                          Hospital  Billing Amount
0                       Abbott Inc    38052.041917
1                       Abbott Ltd    29877.586483
2        Abbott Moore and Williams,    24799.596339
3     Abbott and Thompson, Sullivan    16738.569765
4        Abbott, Peters and Hoffman    18842.396863
...                             ...             ...
39871            and Zimmerman Sons    32706.652625
39872     and Zuniga Davis Carlson,    42867.041298
39873  and Zuniga Francis Peterson,    33689.630726
39874               and Zuniga Sons    33950.170483
39875    and Zuniga Thompson, Blake    22067.428763

[39876 rows x 2 columns]
```

In [214…
```
# COUNT OF MEDICAL CONDITION
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Medical Condition')
plt.title('Count of Medical Condition')
plt.xlabel('Medical Condition')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



In [306…
```
# 2. Average Billing Amount by Blood Type
avg_billing_by_blood_type = df.groupby('Blood Type')['Billing Amount'].mean()
print(avg_billing_by_blood_type)

plt.figure(figsize=(10, 6))
avg_billing_by_blood_type.plot(kind='bar', color='skyblue')
plt.title('Average Billing Amount by Blood Type')
plt.xlabel('Blood Type')
plt.ylabel('Average Billing Amount')
plt.show()
```

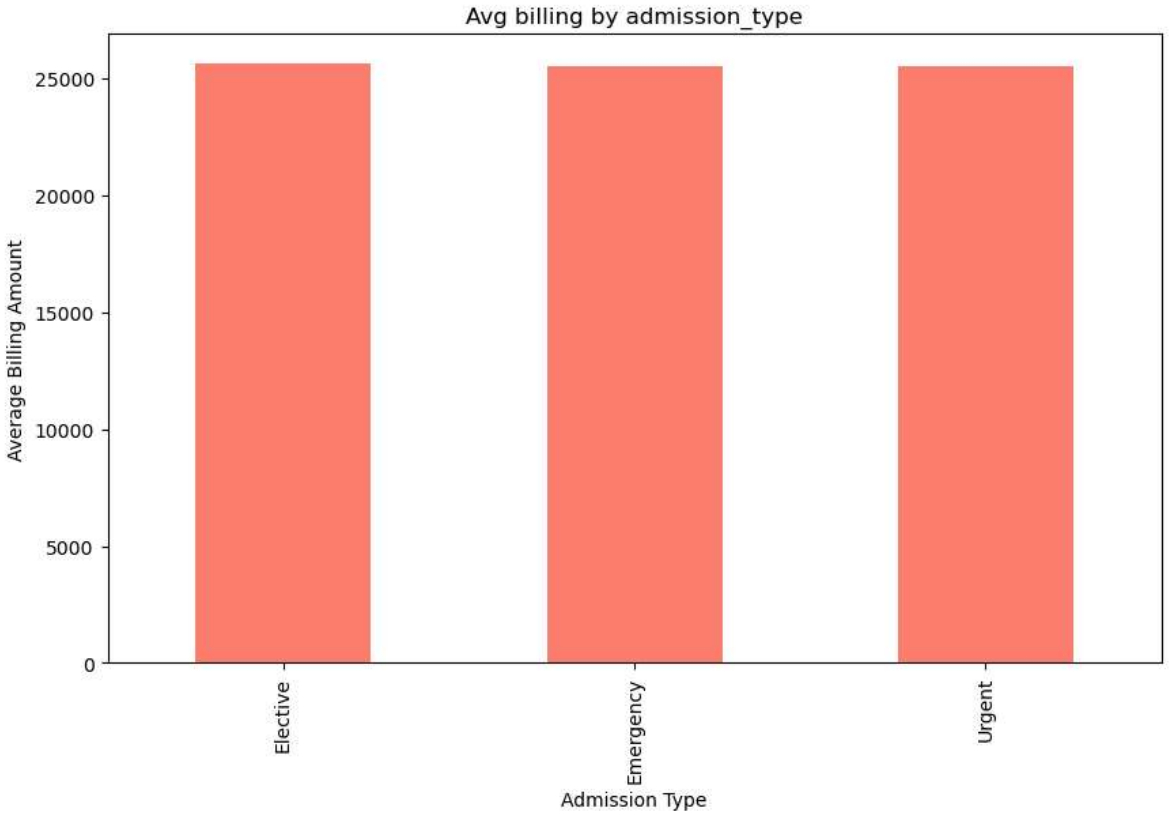
```
Average Billing Amount by Blood Type:
Blood Type
A+     25664.566404
A-     25595.024701
AB+    25361.458784
AB-    25694.933091
B+     25429.723237
B-     25524.424636
O+     25249.740696
O-     25795.657833
Name: Billing Amount, dtype: float64
```



In [338…

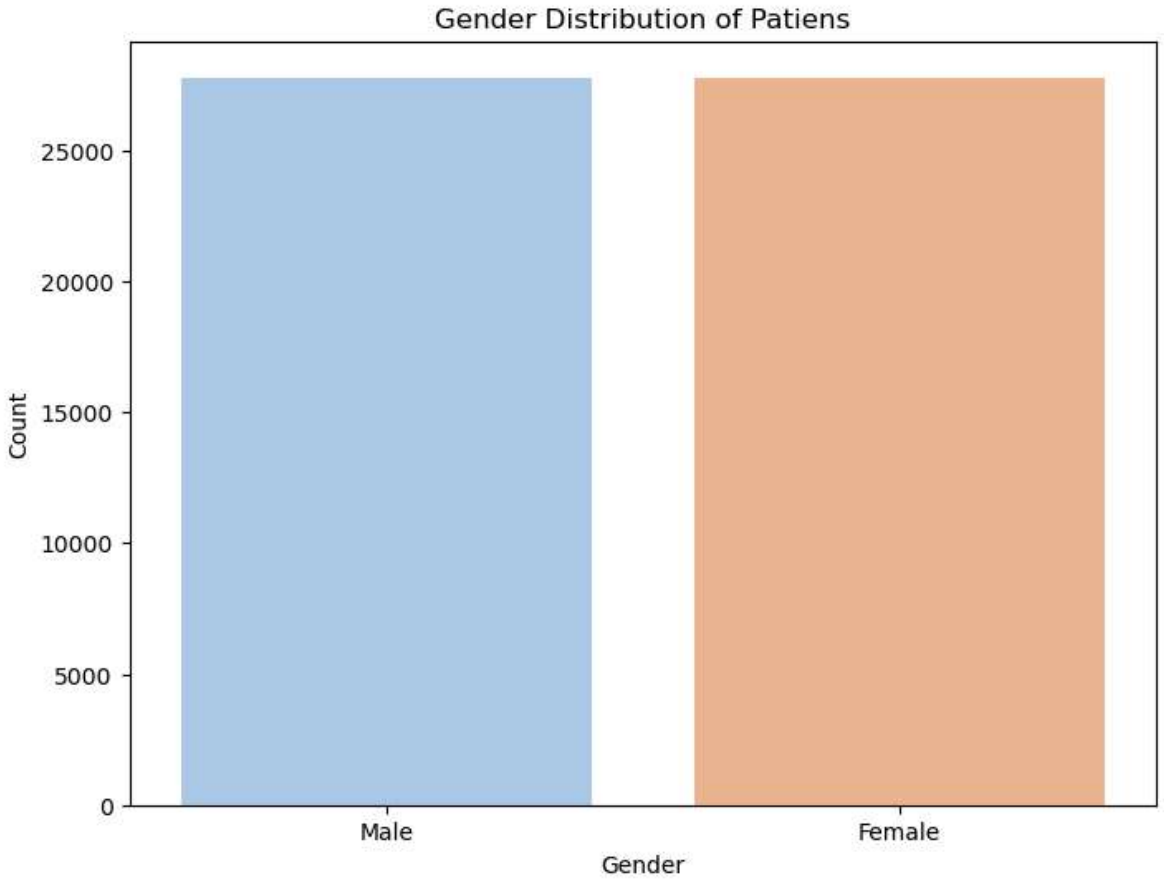
```
# AVERAGE BILLING AMOUNT BY ADMISSION TYPE
avg_billing_by_admission_type =df.groupby('Admission Type')['Billing Amount'].me
print(avg_billing_by_admission_type)

#ploting
plt.figure(figsize=(10, 6))
avg_billing_by_admission_type.plot(kind='bar', color="salmon")
plt.title('Avg billing by admission_type')
plt.xlabel('Admission Type')
plt.ylabel('Average Billing Amount')
plt.show()
```

```
Admission Type
Elective     25602.226311
Emergency    25497.397157
Urgent       25517.364497
Name: Billing Amount, dtype: float64
```



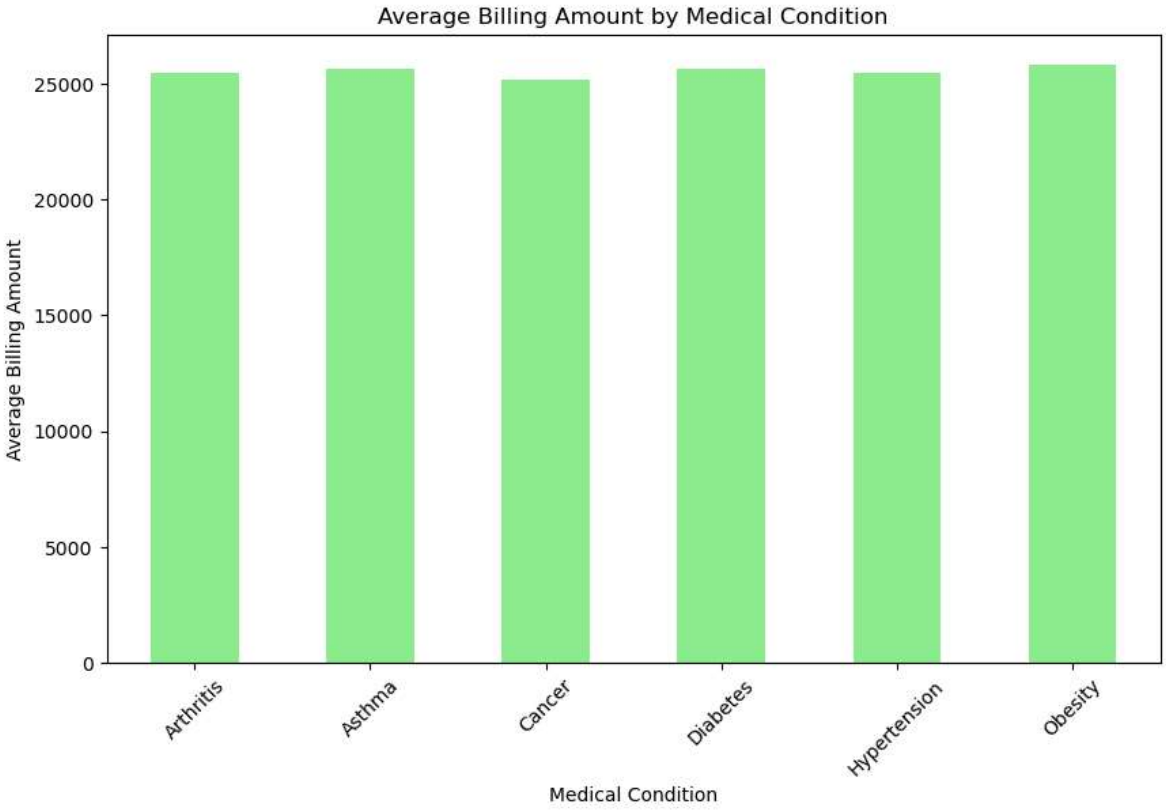
In [23]:
```
# AVERAGE BILLING BY GENDER
groupby_gender = df.groupby('Gender')['Billing Amount'].mean().reset_index()
print(groupby_gender)

#GENDER DISTRIBUTION OF PATIENTS
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Gender', hue='Gender', palette='pastel')
plt.title('Gender Distribution of Patiens')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

```
   Gender  Billing Amount
0  Female    25470.652958
1    Male    25607.860571
```



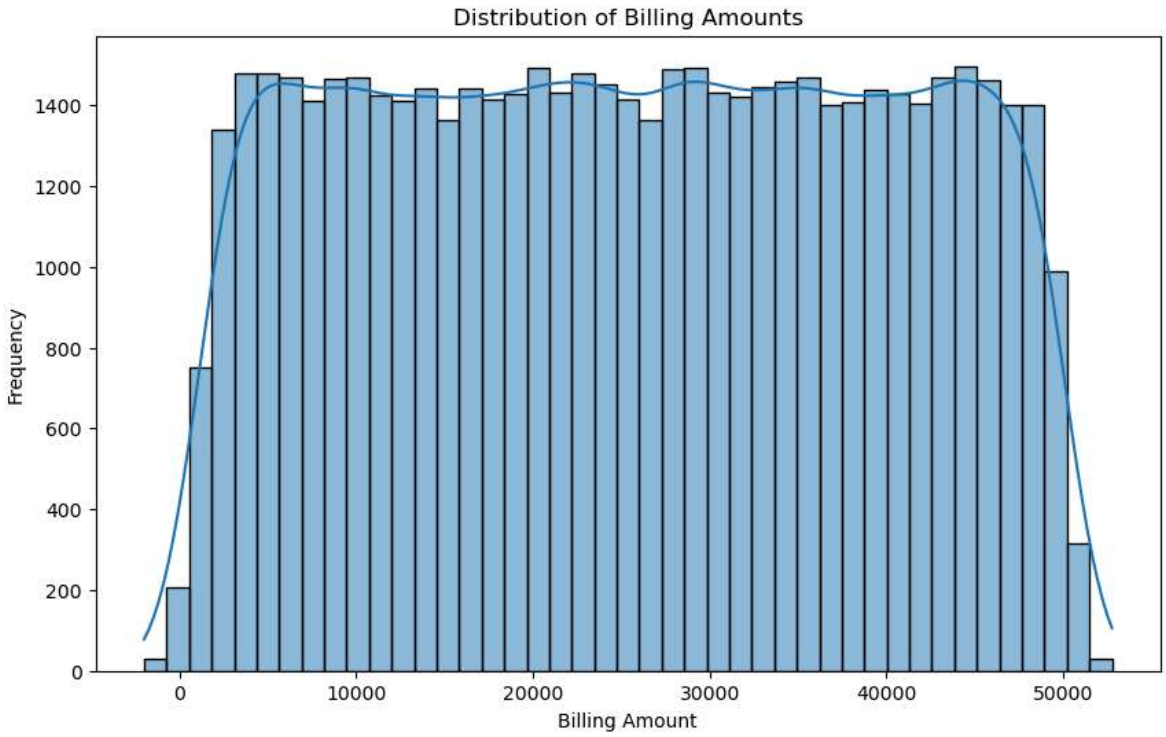
In [25]:
```
# 7. Average Billing Amount by Medical Condition
avg_billing_by_medical_condition = df.groupby('Medical Condition')['Billing Amou
print(avg_billing_by_medical_condition)

plt.figure(figsize=(10, 6))
avg_billing_by_medical_condition.plot(kind='bar', color='lightgreen')
plt.title('Average Billing Amount by Medical Condition')
plt.xlabel('Medical Condition')
plt.ylabel('Average Billing Amount')
plt.xticks(rotation=45)
plt.show()
```

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



In [312…

```
# Distribution of Billing Amounts
plt.figure(figsize=(10, 6))
sns.histplot(df['Billing Amount'], kde=True)
plt.title('Distribution of Billing Amounts')
plt.xlabel('Billing Amount')
plt.ylabel('Frequency')
plt.show()
```



Key Insights, Take away and Implications

Resource Alocation: High rates of cancer, diabetes, and emergency cases need specialized staff and resources.

Cost Optimization: Billing trends by condition and insurer can highlight cost-saving areas and standardize treatments.

Insurance Collaboration: Insights on insurance and admissions can aid in better patient distribution and insurer partnerships.

Better Diagnostics: High inconclusive tests and emergency admissions suggest a need for improved diagnostics and preventive care.

Personalized Care: Variations by age, gender, and blood type support developing targeted and effective patient care.

Data source: https://www.kaggle.com/datasets/prasad22/healthcdataset