healthcare-data-analysis

September 21, 2024

Data Source: https://www.kaggle.com/datasets/prasad22/healthcare-dataset

About Dataset Context: This synthetic healthcare dataset has been created to serve as a valuable resource for data science, machine learning, and data analysis enthusiasts. It is designed to mimic real-world healthcare data, enabling users to practice, develop, and showcase their data manipulation and analysis skills in the context of the healthcare industry.

Dataset Information: Each column provides specific information about the patient, their admission, and the healthcare services provided, making this dataset suitable for various data analysis and modeling tasks in the healthcare domain. Here's a brief explanation of each column in the dataset -

- **Name**: This column represents the name of the patient associated with the healthcare record.
- **Age**: The age of the patient at the time of admission, expressed in years.
- **Gender**: Indicates the gender of the patient, either "Male" or "Female."
- **Blood Type**: The patient's blood type, which can be one of the common blood types (e.g., "A+", "O-", etc.).
- **Medical Condition**: This column specifies the primary medical condition or diagnosis associated with the patient, such as "Diabetes," "Hypertension," "Asthma," and more.
- **Date of Admission**: The date on which the patient was admitted to the healthcare facility.
- **Doctor**: The name of the doctor responsible for the patient's care during their admission.
- **Hospital**: Identifies the healthcare facility or hospital where the patient was admitted.
- **Insurance Provider**: This column indicates the patient's insurance provider, which can be one of several options, including "Aetna," "Blue Cross," "Cigna," "UnitedHealthcare," and "Medicare."
- **Billing Amount**: The amount of money billed for the patient's healthcare services during their admission. This is expressed as a floating-point number.
- **Room Number**: The room number where the patient was accommodated during their admission.
- **Admission Type**: Specifies the type of admission, which can be "Emergency," "Elective," or "Urgent," reflecting the circumstances of the admission.
- **Discharge Date**: The date on which the patient was discharged from the healthcare facility, based on the admission date and a random number of days within a realistic range.
- **Medication**: Identifies a medication prescribed or administered to the patient during their admission. Examples include "Aspirin," "Ibuprofen," "Penicillin," "Paracetamol," and "Lipitor."
- **Test Results**: Describes the results of a medical test conducted during the patient's admission. Possible values include "Normal," "Abnormal," or "Inconclusive," indicating the outcome of the test.

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline
```

Reding the dataset and showing first 10 rows

```
[2]: data=pd.read_csv("healthcare_dataset.csv")
     data.head(10)
```

```
[2]:                  Name  Age  Gender Blood Type Medical Condition  \
     0       Bobby JacksOn   30    Male         B-            Cancer
     1        LesLie TErRy   62    Male         A+           Obesity
     2         DaNnY sMitH   76  Female         A-           Obesity
     3        andrEw waTtS   28  Female         O+          Diabetes
     4       adrIENNE bEll   43  Female        AB+            Cancer
     5       EMILY JOHNSOn   36    Male         A+            Asthma
     6      edwArD EDWaRDs   21  Female        AB-          Diabetes
     7  CHrisTInA MARtinez   20  Female         A+            Cancer
     8     JASmINe aGuIlaR   82    Male        AB+            Asthma
     9    ChRISTopher BerG   58  Female        AB-            Cancer

       Date of Admission            Doctor                     Hospital  \
     0        2024-01-31     Matthew Smith              Sons and Miller
     1        2019-08-20   Samantha Davies                      Kim Inc
     2        2022-09-22  Tiffany Mitchell                     Cook PLC
     3        2020-11-18       Kevin Wells   Hernandez Rogers and Vang,
     4        2022-09-19    Kathleen Hanna                  White-White
     5        2023-12-20     Taylor Newton               Nunez-Humphrey
     6        2020-11-03       Kelly Olson              Group Middleton
     7        2021-12-28    Suzanne Thomas  Powell Robinson and Valdez,
     8        2020-07-01   Daniel Ferguson                Sons Rich and
     9        2021-05-23       Heather Day               Padilla-Walker

       Insurance Provider  Billing Amount  Room Number Admission Type  \
     0         Blue Cross    18856.281306          328         Urgent
     1           Medicare    33643.327287          265      Emergency
     2              Aetna    27955.096079          205      Emergency
     3           Medicare    37909.782410          450       Elective
     4              Aetna    14238.317814          458         Urgent
     5   UnitedHealthcare    48145.110951          389         Urgent
     6           Medicare    19580.872345          389      Emergency
     7              Cigna    45820.462722          277      Emergency
     8              Cigna    50119.222792          316       Elective
     9   UnitedHealthcare    19784.631062          249       Elective
```

```
  Discharge Date   Medication  Test Results
0     2024-02-02  Paracetamol        Normal
1     2019-08-26    Ibuprofen  Inconclusive
2     2022-10-07      Aspirin        Normal
3     2020-12-18    Ibuprofen      Abnormal
4     2022-10-09   Penicillin      Abnormal
5     2023-12-24    Ibuprofen        Normal
6     2020-11-15  Paracetamol  Inconclusive
7     2022-01-07  Paracetamol  Inconclusive
8     2020-07-14      Aspirin      Abnormal
9     2021-06-22  Paracetamol  Inconclusive
```

Size of the dataset

[3]:

```
data.shape
```

[3]:

```
(55500, 15)
```

Types of the data

[4]:

```
data.dtypes
```

[4]:

```
Name                   object
Age                     int64
Gender                 object
Blood Type             object
Medical Condition      object
Date of Admission      object
Doctor                 object
Hospital               object
Insurance Provider     object
Billing Amount        float64
Room Number             int64
Admission Type         object
Discharge Date         object
Medication             object
Test Results           object
dtype: object
```

Frequency distribution

[5]:

```
## counting the frequency of test result
freq_count=data['Test Results'].value_counts()
freq_count
```

[5]:

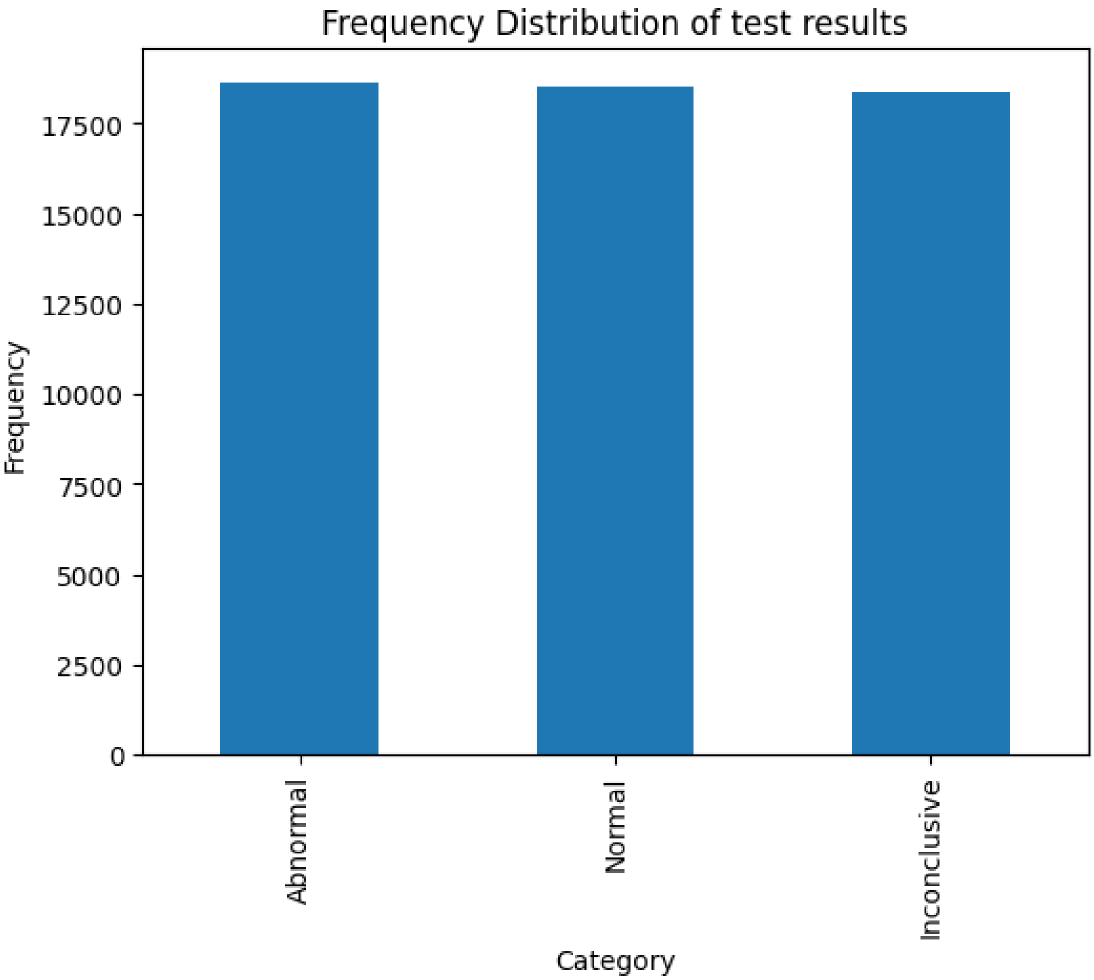
```
Test Results
Abnormal        18627
Normal          18517
```

```
Inconclusive    18356
Name: count, dtype: int64
```

Bar diagram of the frequency distribution of the following

[6]:

```
freq_count.plot(kind='bar')
plt.title('Frequency Distribution of test results')
plt.xlabel('Category')
plt.ylabel('Frequency')
plt.show()
```



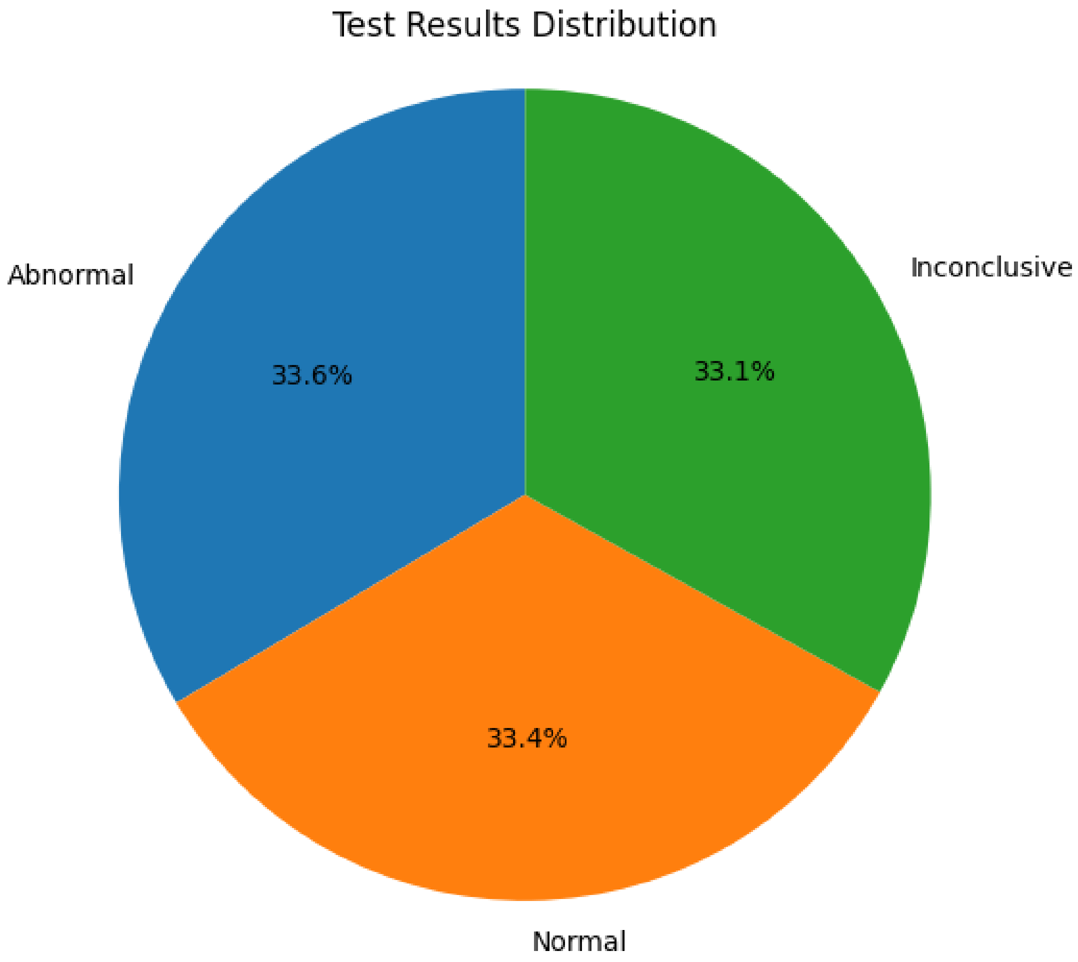
Pie chart of the frequency distribution of the following

[7]:

```
# Get the frequency of each unique value in 'Test Results'
test_results_counts =data['Test Results'].value_counts()

# Create a pie chart
```

```
plt.figure(figsize=(6,6))
plt.pie(test_results_counts, labels=test_results_counts.index, autopct='%1.
  ↪1f%%', startangle=90)
plt.title('Test Results Distribution')
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is circular.
plt.show()
```



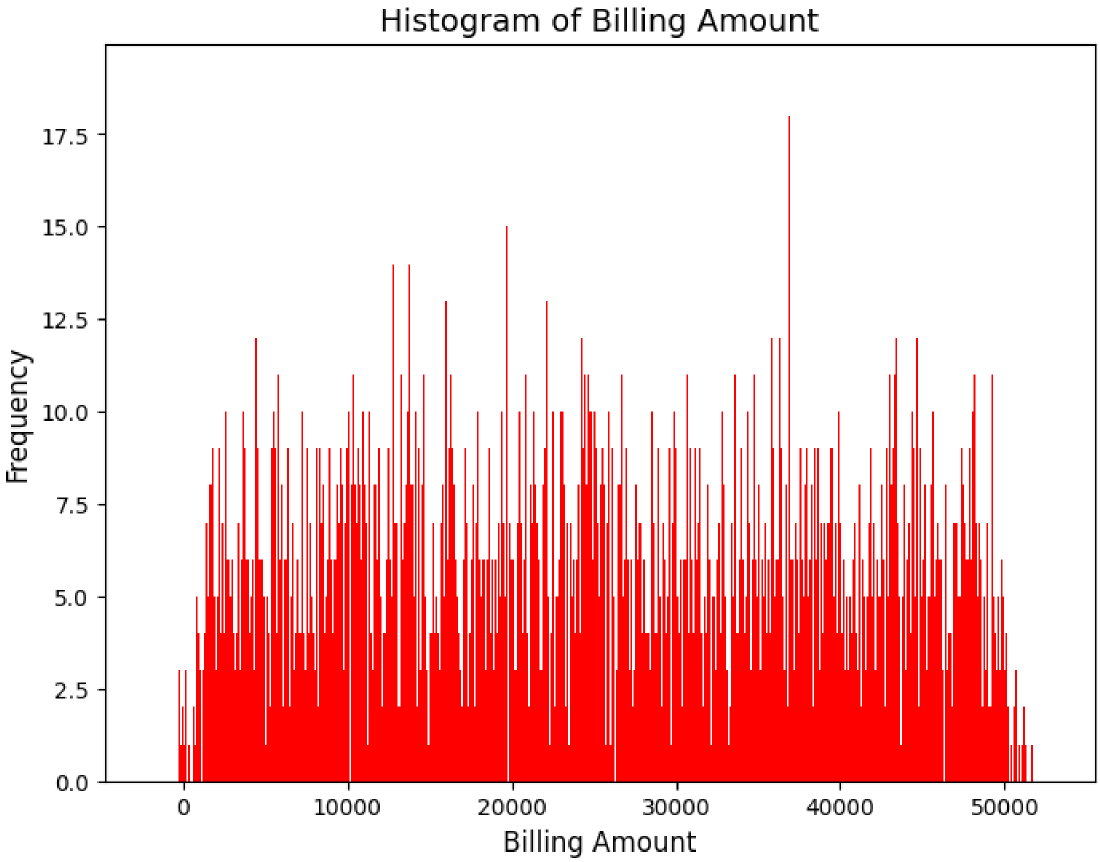
Histogram of the frequency distribution of the following

[8]:

```
plt.figure(figsize=(8, 6))
plt.hist(data['Billing Amount'], bins=10000, color='red')

# Add titles and labels
plt.title('Histogram of Billing Amount', fontsize=14)
plt.xlabel('Billing Amount', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
```

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

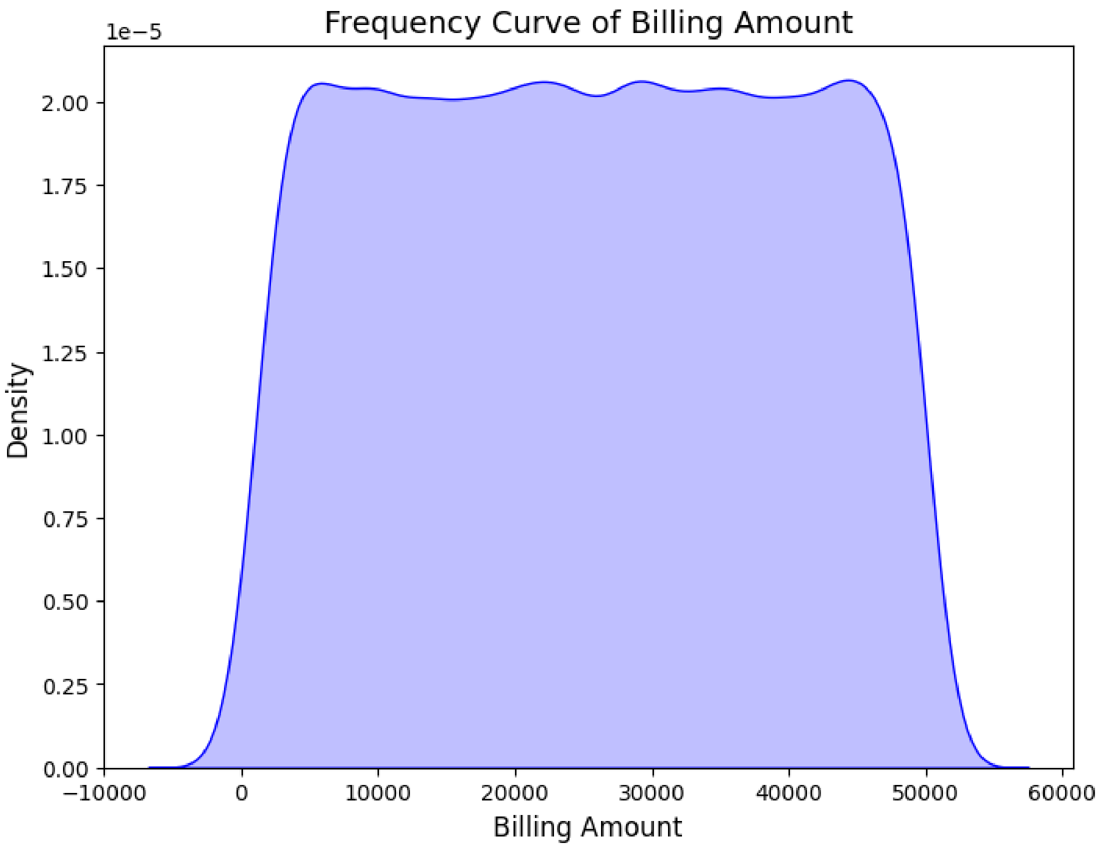


frequency curve of the frequency distribution of the following

```
[9]: plt.figure(figsize=(8, 6))
sns.kdeplot(data['Billing Amount'], color='blue', fill=True)

# Add titles and labels
plt.title('Frequency Curve of Billing Amount', fontsize=14)
plt.xlabel('Billing Amount', fontsize=12)
plt.ylabel('Density', fontsize=12)

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



```
[10]: data.columns
```

```
[10]: 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')
```

Scatter Plot between Age and Billing Amount

```
[11]: # Convert the 'Date of Admission' and 'Discharge Date' columns to datetime
      data['Date of Admission'] = pd.to_datetime(data['Date of Admission'])
      data['Discharge Date'] = pd.to_datetime(data['Discharge Date'])
      # Calculate the difference in days between the two dates
      data['Admission Duration'] = (data['Discharge Date'] - data['Date of␣
        ↪Admission']).dt.days
      # Display the DataFrame with the new column
```

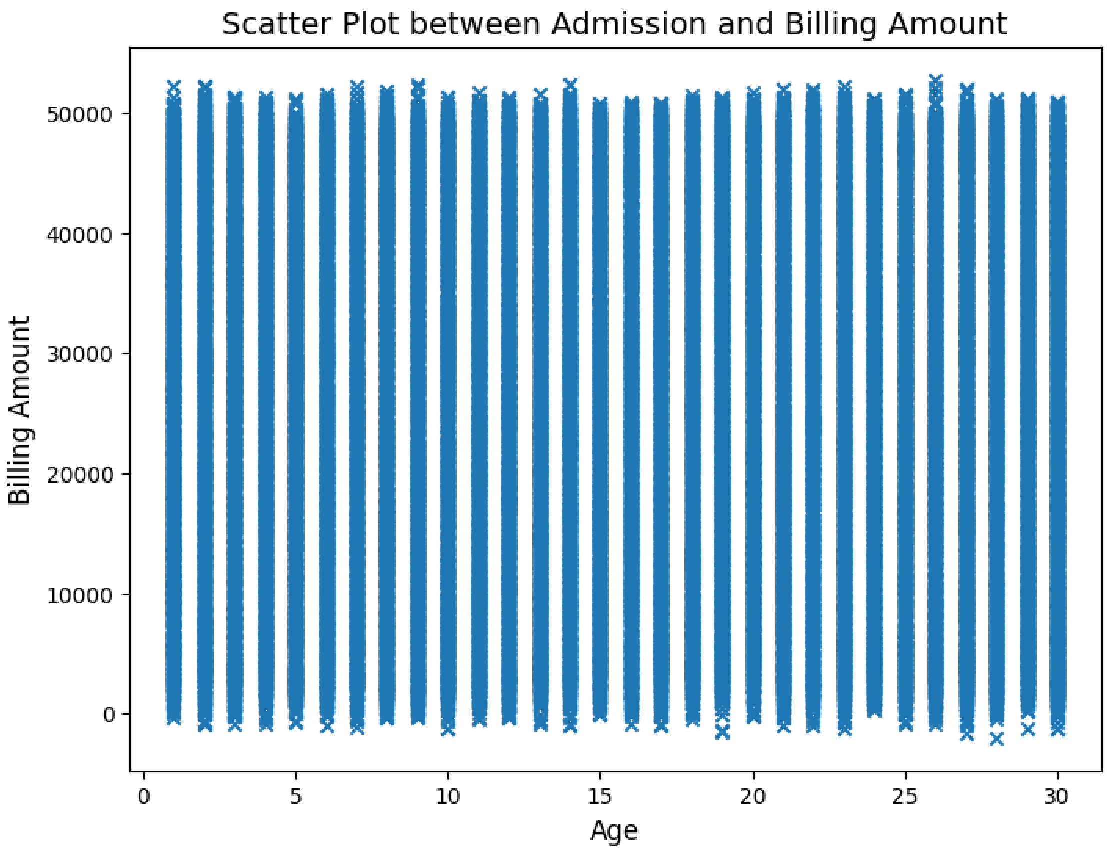
Scatter Plot between Admission and Billing Amount

[12]:

```
plt.figure(figsize=(8, 6))
plt.scatter(data['Admission Duration'], data['Billing Amount'],marker='x')

# Add titles and labels
plt.title('Scatter Plot between Admission and Billing Amount', fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Billing Amount', fontsize=12)

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



Box plot of age column

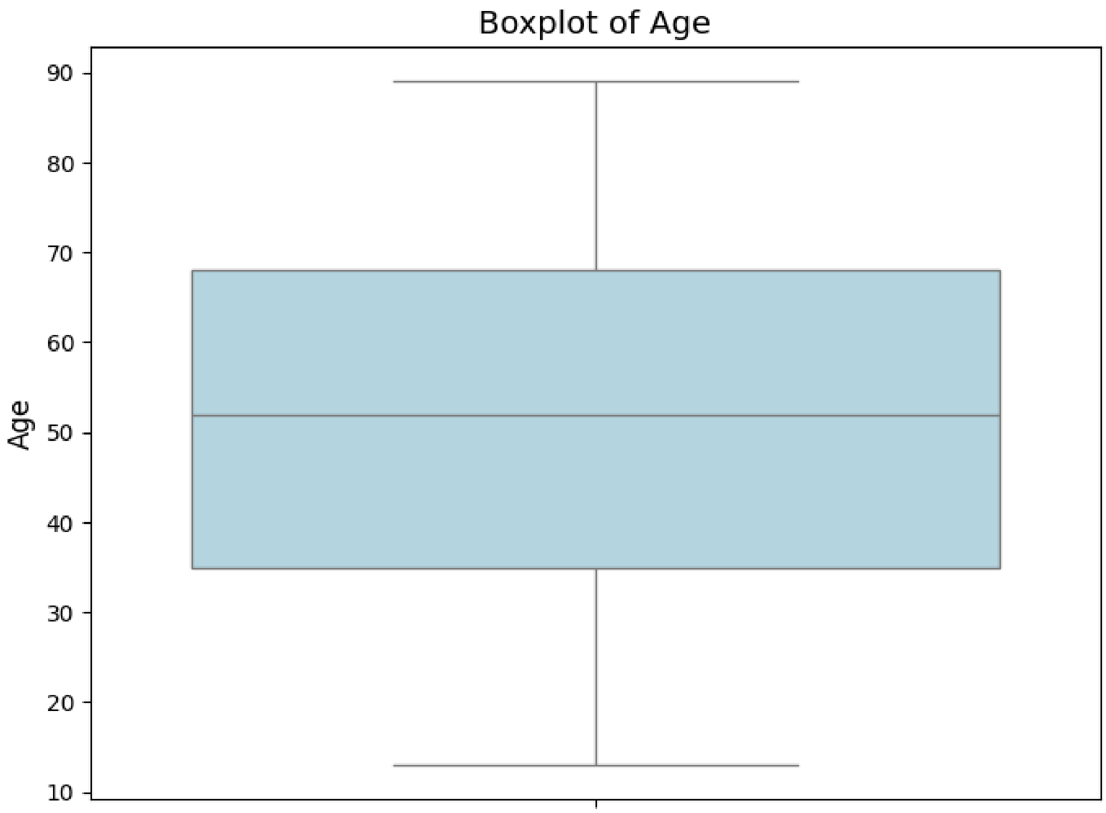
[13]:

```
plt.figure(figsize=(8, 6))
sns.boxplot(y=data['Age'], color='lightblue')

# Add titles and labels
plt.title('Boxplot of Age', fontsize=14)
plt.ylabel('Age', fontsize=12)
```

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

```
c:\Users\Acer\AppData\Local\Programs\Python\Python312\Lib\site-
packages\seaborn\categorical.py:640: FutureWarning: SeriesGroupBy.grouper is
deprecated and will be removed in a future version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```



calculating the quartiles of the age column

[14]:
```
Q1 = data['Age'].quantile(0.25)  # First quartile (25th percentile)
Q2 = data['Age'].quantile(0.50)  # Second quartile (50th percentile or median)
Q3 = data['Age'].quantile(0.75)  # Third quartile (75th percentile)

# Display the quartiles
print(f"First Quartile (Q1): {Q1}")
print(f"Second Quartile (Q2): {Q2}")
print(f"Third Quartile (Q3): {Q3}")
```

```
First Quartile (Q1): 35.0
Second Quartile (Q2): 52.0
Third Quartile (Q3): 68.0
```

MEAN, MEDIAN, SKEWNESS, RANGE, MEAN DEVIATION, VARIANCE, STANDARD DEVIATION and coefficient of variation (CV) for the "Age" and "Billing Amount" columnsT

[15]:

```python
from scipy.stats import skew

# Sample data for 'Age' and 'Billing Amount'
copy_data = {'Age': data['Age'],
        'Billing Amount': data['Billing Amount']}
df = pd.DataFrame(copy_data)

# Function to calculate mean deviation
def mean_deviation(series):
    mean_value = series.mean()
    return (series - mean_value).abs().mean()
# Calculate statistics for 'Age' and 'Billing Amount'
for column in ['Age', 'Billing Amount']:
    print(f"Statistics for {column}:")
    print(f"Mean: {df[column].mean()}")
    print(f"Median: {df[column].median()}")
    print(f"Skewness: {skew(df[column])}")
    print(f"Range: {df[column].max() - df[column].min()}")
    print(f"Mean Deviation: {mean_deviation(df[column])}")
    print(f"Variance: {df[column].var()}")
    print(f"Standard Deviation: {df[column].std()}")
    print(f"Coefficient of Variation (CV): {df[column].std() / df[column].mean()}")
    print()
```

```
Statistics for Age:
Mean: 51.53945945945946
Median: 52.0
Skewness: -0.005735115665703919
Range: 76
Mean Deviation: 16.948137092768444
Variance: 384.2561953149387
Standard Deviation: 19.602453808514348
Coefficient of Variation (CV): 0.38033875430791986

Statistics for Billing Amount:
Mean: 25539.316097211795
Median: 25538.069375965664
Skewness: -0.0009777690118698563
Range: 54772.76887632831
Mean Deviation: 12297.475837537435
Variance: 201965437.04053578
Standard Deviation: 14211.454430864414
Coefficient of Variation (CV): 0.5564539933947535
```

Correlation and correlation coefficient between the “Admission Duration” and “Billing Amount” columns

[16]:

```python
import pandas as pd

# Sample data for 'Age' and 'Billing Amount'
copy_data = {'Admission Duration': data['Admission Duration'],
        'Billing Amount': data['Billing Amount']}
df = pd.DataFrame(copy_data)

# Calculate the correlation matrix
correlation_matrix = df.corr()

# Extract the correlation coefficient between 'Age' and 'Billing Amount'
corr_coefficient = df['Admission Duration'].corr(df['Billing Amount'])

# Display the correlation matrix and the specific correlation coefficient
print("Correlation Matrix:")
print(correlation_matrix)

print(f"\nCorrelation Coefficient between Age and Billing Amount:␣
 ↪{corr_coefficient:.4f}")
```

```
Correlation Matrix:
                    Admission Duration  Billing Amount
Admission Duration            1.000000       -0.005602
Billing Amount               -0.005602        1.000000

Correlation Coefficient between Age and Billing Amount: -0.0056
```

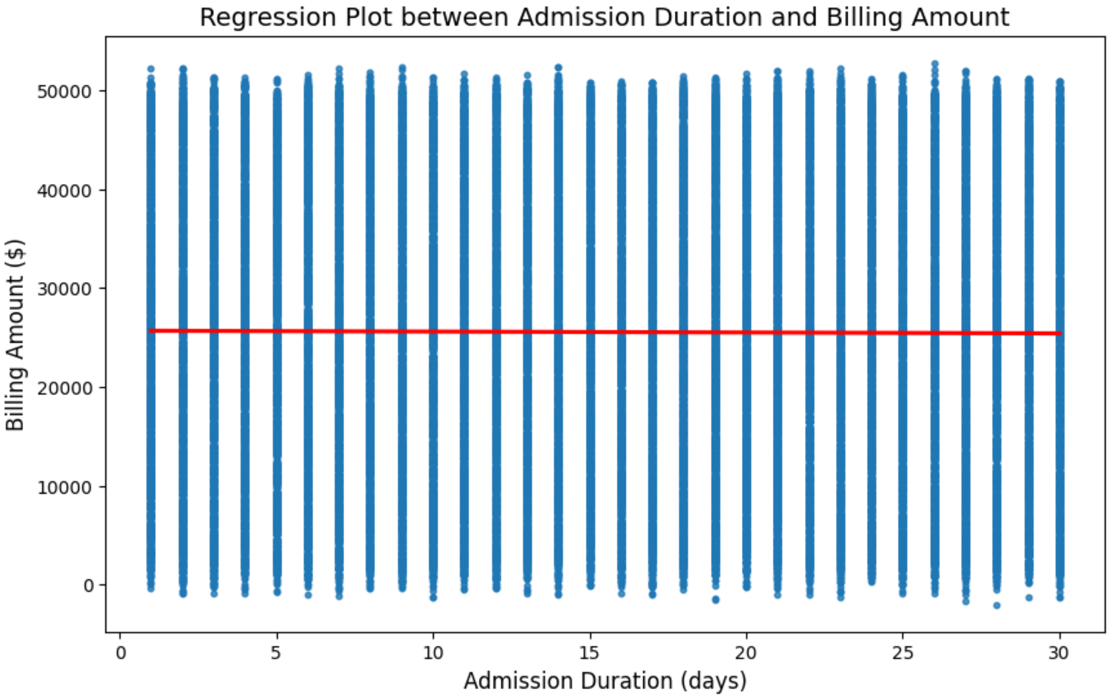
Regression Plot between Admission Duration and Billing Amount

[17]:

```python
plt.figure(figsize=(10, 6))
sns.regplot(x='Admission Duration', y='Billing Amount', data=data,␣
 ↪scatter_kws={'s':10}, line_kws={'color':'red'})

# Add titles and labels
plt.title('Regression Plot between Admission Duration and Billing Amount',␣
 ↪fontsize=14)
plt.xlabel('Admission Duration (days)', fontsize=12)
plt.ylabel('Billing Amount ($)', fontsize=12)

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


Regression Plot between Admission Duration and Billing Amount
Billing Amount ($)
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
30
Admission Duration (days)