In [1]:

```
import pandas as pd
```

Exploratory Data Analysis(EDA)

In [9]:

```
# Load the dataset
df = pd.read_csv("C:/Users/admin/Desktop/Healthcare_visit/Dataset/Healthcare_Visits_Report.csv")

# Preview the dataset
print(df.head())
```

```
   VisitID   VisitDate PatientID         Hospital  Department     Diagnosis  \
0  VIS1000   8/13/2024   PAT5000       St. Mary’s  Cardiology        Cancer
1  VIS1001    2/2/2025   PAT5001  Oakwood Medical  Pediatrics  Hypertension
2  VIS1002   11/2/2024   PAT5002  Oakwood Medical   Neurology        Cancer
3  VIS1003    5/6/2024   PAT5003    Hope General    Oncology        Asthma
4  VIS1004  11/20/2023   PAT5004       St. Mary’s  Pediatrics  Hypertension

  Region  WaitTimeMin  TreatmentCost  SatisfactionScore  Readmitted
0   East           67        2309.79                  1           0
1  North           28        2264.52                  5           0
2   West           61        4547.17                  3           0
3   East           76        1639.25                  3           0
4   East           93        6076.89                  1           1
```

In [10]:

```
# Check for missing values
print(df.isnull().sum())
```

```
VisitID              0
VisitDate            0
PatientID            0
Hospital             0
Department           0
Diagnosis            0
Region               0
WaitTimeMin          0
TreatmentCost        0
SatisfactionScore    0
Readmitted           0
dtype: int64
```

In [11]:

```
# Data types and basic info
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   VisitID            1000 non-null   object
 1   VisitDate          1000 non-null   object
 2   PatientID          1000 non-null   object
 3   Hospital           1000 non-null   object
 4   Department         1000 non-null   object
 5   Diagnosis          1000 non-null   object
 6   Region             1000 non-null   object
 7   WaitTimeMin        1000 non-null   int64
 8   TreatmentCost      1000 non-null   float64
 9   SatisfactionScore  1000 non-null   int64
 10  Readmitted         1000 non-null   int64
dtypes: float64(1), int64(3), object(7)
memory usage: 86.1+ KB
None
```

In [12]:

```
# Summary statistics
print(df.describe())
```

```
       WaitTimeMin  TreatmentCost  SatisfactionScore   Readmitted
count  1000.000000    1000.000000        1000.000000  1000.000000
mean     93.613000    5028.892690           2.986000     0.486000
std      51.390156    2846.208306           1.431032     0.500054
min       5.000000     132.130000           1.000000     0.000000
25%      49.000000    2488.272500           2.000000     0.000000
50%      92.000000    5027.930000           3.000000     0.000000
75%     139.000000    7395.610000           4.000000     1.000000
max     180.000000    9996.100000           5.000000     1.000000
```

In [13]:

```
# Distribution of visits across hospitals
print(df['Hospital'].value_counts())
```

```
Hospital
Sunrise Hospital       221
Oakwood Medical        202
Green Valley Clinic    197
St. Mary’s             191
Hope General           189
Name: count, dtype: int64
```

In [14]:

```python
# Check unique values in key columns
print(df['Department'].unique())
print(df['Diagnosis'].unique())
```

```
['Cardiology' 'Pediatrics' 'Neurology' 'Oncology' 'Orthopedics'
 'Emergency']
['Cancer' 'Hypertension' 'Asthma' 'Diabetes' 'Fracture' 'Migraine']
```

In [15]:

```python
# Grouped analysis: Average wait time and satisfaction by hospital
hospital_summary = df.groupby('Hospital').agg({
    'WaitTimeMin': 'mean',
    'SatisfactionScore': 'mean',
    'TreatmentCost': 'sum'
}).sort_values(by='WaitTimeMin', ascending=False)

print(hospital_summary)
```

```
                     WaitTimeMin  SatisfactionScore  TreatmentCost
Hospital
Hope General           97.063492           2.835979      891270.91
Green Valley Clinic    95.989848           3.000000     1043585.91
Oakwood Medical        94.242574           2.985149     1024370.41
St. Mary’s             92.706806           2.958115      997989.58
Sunrise Hospital       88.751131           3.126697     1071675.88
```

Data Cleaning

In [16]:

```python
# Remove duplicates
df.drop_duplicates(inplace=True)
```

In [ ]:

```python

```