Task 2: Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data.

Import Libraries & Load Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_csv("healthcare_dataset.csv")
# Display first 5 rows
data.head(5)
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Admission Type	Discharge Date
0	Bobby JacksOn	30	Male	B-	Cancer	1/31/2024	Matthew Smith	Sons and Miller	Blue Cross	18856.28131	328	Urgent	2/2/2024
1	LesLie TErRy	62	Male	A+	Obesity	8/20/2019	Samantha Davies	Kim Inc	Medicare	33643.32729	265	Emergency	8/26/2019
2	DaNnY sMitH	76	Female	A-	Obesity	9/22/2022	Tiffany Mitchell	Cook PLC	Aetna	27955.09608	205	Emergency	10/7/2022
3	andrEw waTtS	28	Female	O+	Diabetes	11/18/2020	Kevin Wells	Hernandez Rogers and Vang,	Medicare	37909.78241	450	Elective	12/18/2020
4	adrIENNE bEll	43	Female	AB+	Cancer	9/19/2022	Kathleen Hanna	White-White	Aetna	14238.31781	458	Urgent	10/9/2022

Dataset Information

```
data.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
```

```
data.shape
```

```
(55500, 15)
```

```
data.describe()
```

	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	13241.224655	202.000000
50%	52.000000	25538.069380	302.000000
75%	68.000000	37820.508432	401.000000
max	89.000000	52764.276740	500.000000

Data Cleaning

Check Missing Values

```
data.isnull().sum()
```

	0
Name	0
Age	0
Gender	0
Blood Type	0
Medical Condition	0
Date of Admission	0
Doctor	0
Hospital	0
Insurance Provider	0
Billing Amount	0
Room Number	0
Admission Type	0
Discharge Date	0
Medication	0
Test Results	0

dtype: int64

Fill Missing Values

```
# Fill Age missing values with mean
data['Age'].fillna(data['Age'].mean(), inplace=True)

# Fill categorical columns with mode
data['Gender'].fillna(data['Gender'].mode()[0], inplace=True)
data['Blood Type'].fillna(data['Blood Type'].mode()[0], inplace=True)
```

Remove Duplicate Records

```
data.drop_duplicates(inplace=True)
```

Convert Date Columns

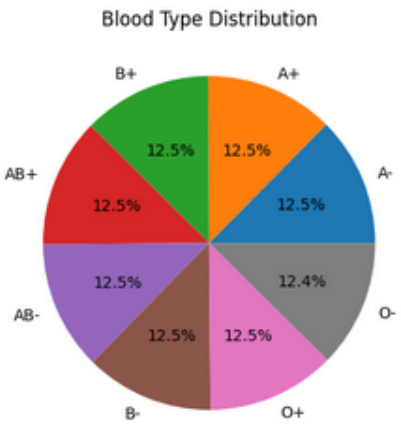
```
data['Date of Admission'] = pd.to_datetime(data['Date of Admission'])
data['Discharge Date'] = pd.to_datetime(data['Discharge Date'])
```

Create New Feature (Length of Stay)

```
data['Stay_Days'] = (data['Discharge Date'] - data['Date of Admission']).dt.days
```

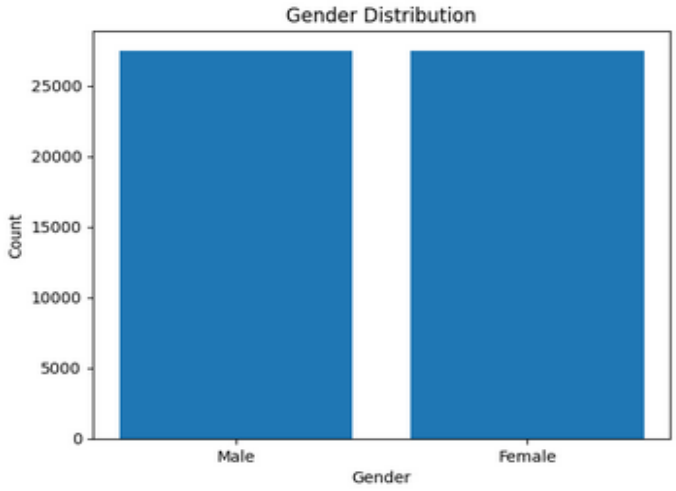
Pie Chart – Blood Type Distribution

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

plt.figure()
plt.pie(blood_count.values, labels=blood_count.index, autopct='%1.1f%%')
plt.title("Blood Type Distribution")
plt.show()
```



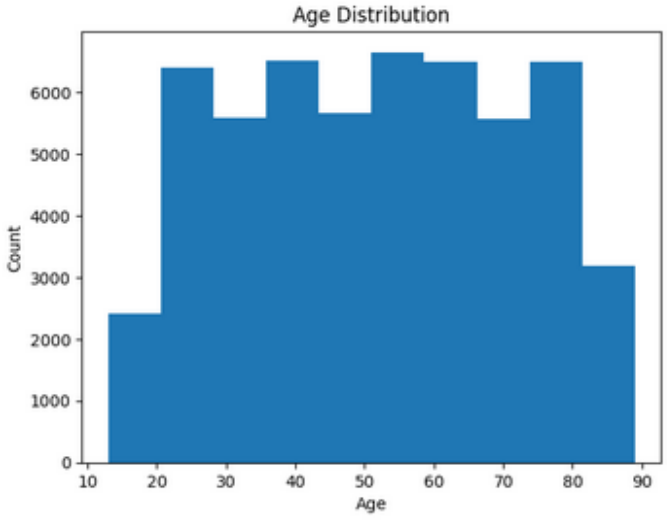
Bar Chart – Gender Distribution

```
gender_count = data['Gender'].value_counts()

plt.figure()
plt.bar(gender_count.index, gender_count.values)
plt.xlabel("Gender")
plt.ylabel("Count")
plt.title("Gender Distribution")
plt.show()
```



Histogram – Age Distribution

```
plt.figure()
plt.hist(data['Age'], bins=10)
plt.xlabel("Age")
plt.ylabel("Count")
plt.title("Age Distribution")
plt.show()
```



Heat Plot

```
import seaborn as sns
numeric_data = data.select_dtypes(include=['int64','float64'])
plt.figure(figsize=(10,6))
sns.heatmap(numeric_data.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

Correlation Analysis (Numeric Columns)

```
numeric_data = data.select_dtypes(include=['int64','float64'])
numeric_data.corr()
```

	Age	Billing Amount	Room Number	Stay_Days
Age	1.000000	-0.003427	-0.000352	0.007890
Billing Amount	-0.003427	1.000000	-0.003930	-0.004891
Room Number	-0.000352	-0.003930	1.000000	-0.004540
Stay_Days	0.007890	-0.004891	-0.004540	1.000000