Health Care Data Exploration

A PROJECT REPORT
for
Introduction to AI (AI101B)
Session (2024-25)

Submitted by

Mayank Srivastava
202410116100118
Mohammad Daud
202410116100121

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr. Apoorva Jain
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(MARCH - 2025)

1. Introduction

In today's fast-paced healthcare environment, data-driven decision-making plays a crucial role in improving patient care, optimizing hospital resources, and enhancing overall efficiency. With the increasing availability of electronic health records (EHRs) and hospital management systems, healthcare institutions have access to vast amounts of patient data. However, making sense of this data requires systematic analysis and visualization to extract meaningful insights.

This project, "Healthcare Data Exploration," aims to analyze a hospital dataset to uncover key patterns related to patient demographics, medical conditions, treatment costs, and hospital utilization metrics. By performing exploratory data analysis (EDA) and visualization, we seek to answer essential questions such as:

- What are the most common medical conditions and procedures?
- How does age, gender, and medical history impact hospital stay duration and treatment costs?
- What are the trends in readmission rates and patient satisfaction?
- Are there any correlations between treatment costs, length of stay, and patient outcomes?

To achieve these objectives, this project follows a structured methodology, starting with data preprocessing to ensure quality, followed by EDA and visualization to derive actionable insights. The findings from this analysis can be valuable for hospital administrators, policymakers, and healthcare professionals to enhance patient care strategies and optimize hospital operations.

By leveraging Python and its powerful data science libraries (Pandas, Matplotlib, Seaborn, etc.), this project demonstrates how healthcare data analytics can help improve decision-making and patient outcomes in the medical field.

2. Methodology

The project follows a structured data exploration methodology, ensuring a systematic approach to analyzing the healthcare dataset. The methodology consists of the following steps:

Step 1: Data Collection

• The dataset is sourced from Kaggle's "Hospital Dataset for Practice", containing patient information, medical conditions, and treatment details.

• The dataset is in CSV format, making it suitable for direct processing using Pandas in Python.

Step 2: Data Preprocessing

To ensure data quality and reliability, preprocessing steps are applied:

1. Handling Missing Data:

• Numerical missing values (e.g., Cost, Length_of_Stay) are filled with the median value to maintain data consistency.

• Categorical missing values (e.g., Gender, Condition) are replaced with. "Unknown".

2. Data Type Conversion:

• Convert categorical variables (Gender, Readmission, Outcome) into the appropriate format for analysis.

• Ensure numerical columns are stored correctly to perform mathematical operations.

3.Feature Engineering:

• Create an Age Group column to categorize patients into predefined age ranges (e.g., 0-18, 19-35).

Step 3: Exploratory Data Analysis (EDA)

EDA helps uncover patterns, trends, and relationships in the data:

• Summary Statistics:

• Use describe() to compute mean, median, and standard deviation.

• Demographic Analysis:

• Explore the distribution of Age, Gender, and Conditions.

• Hospital Utilization:

• Analyze Length_of_Stay, Readmission rates, and Cost variations.

• Outcome Analysis: Study the relationship between conditions and patient outcomes.

Step 4: Data Visualization

Visualization helps in understanding key trends and correlations:

1. Bar Plots & Count Plots:
    • Used for Gender distribution, Condition frequency, and Readmission rates.

2.Histograms:
    • Show the distribution of Age and Length of Stay.

3. Scatter Plots:
    • Reveal relationships between Cost and Length of Stay.

4. Heatmaps:
    • Display correlations between numerical variables.

Step 5: Insights & Interpretation

    • Extract key patterns and observations from the analysis.
    • Identify high-risk conditions, costly treatments, and patient satisfaction trends.
    • Provide recommendations for improving hospital efficiency and patient care.

2. Project Implementation

```
import pandas as pd
df = pd.read_csv("hospital data analysis.csv")

# Display basic dataset information
print(df.info())
```

```
#print first 5 rows
print(df.head())

#find if there are any missing values
print(df.isnull().sum())

# Summary statistics
print(df.describe())

# Distribution of age groups
df["Age_Group"] = pd.cut(df["Age"], bins=[0, 18, 35, 50, 65, 100], labels=['0-18',
'19-35', '36-50', '51-65', '65+'])
print(df["Age_Group"].value_counts())

# Count patients by gender
print(df["Gender"].value_counts())

df.value_counts()

print(df["Condition"].value_counts())

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.countplot(y=df["Condition"], order=df["Condition"].value_counts().index,
palette='coolwarm')
plt.xlabel("Count")
plt.ylabel("Disease Type")
plt.title("Distribution of Diseases in Patients")
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x=df["Age_Group"], palette='viridis')
plt.xlabel("Age Group")
plt.ylabel("Count")
plt.title("Distribution of Patients Across Age Groups")
plt.show()

plt.figure(figsize=(10, 6))
sns.histplot(df["Length_of_Stay"], bins=20, kde=True, color='purple')
plt.xlabel("Hospital Stay Duration (Days)")
plt.ylabel("Frequency")
plt.title("Distribution of Hospital Stay Durations")
plt.show()

plt.figure(figsize=(10, 6))
```

```
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm',
linewidths=0.5)
plt.title("Correlation Matrix of Healthcare Data")
plt.show()
```

3. Output

```
# Display basic dataset information
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 984 entries, 0 to 983
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Patient_ID      984 non-null    int64
 1   Age             984 non-null    int64
 2   Gender          984 non-null    object
 3   Condition       984 non-null    object
 4   Procedure       984 non-null    object
 5   Cost            984 non-null    int64
 6   Length_of_Stay  984 non-null    int64
 7   Readmission     984 non-null    object
 8   Outcome         984 non-null    object
 9   Satisfaction    984 non-null    int64
dtypes: int64(5), object(5)
memory usage: 77.0+ KB
```

```
# Show the first few rows
df.head()
```

	Patient_ID	Age	Gender	Condition	Procedure	Cost	Length_of_Stay	Readmission	Outcome	Satisfaction	Age_Group
0	1	45	Female	Heart Disease	Angioplasty	15000	5	No	Recovered	4	36-50
1	2	60	Male	Diabetes	Insulin Therapy	2000	3	Yes	Stable	3	51-65
2	3	32	Female	Fractured Arm	X-Ray and Splint	500	1	No	Recovered	5	19-35
3	4	75	Male	Stroke	CT Scan and Medication	10000	7	Yes	Stable	2	65+
4	5	50	Female	Cancer	Surgery and Chemotherapy	25000	10	No	Recovered	4	36-50

```
df.value_counts()
```

Patient_ID	Age	Gender	Condition	Procedure	Cost	Length_of_Stay	Readmission	Outcome	Satisfaction	Age_Group	count
1000	25	Male	Allergic Reaction	Epinephrine Injection	100	68	No	Recovered	5	19-35	1
1	45	Female	Heart Disease	Angioplasty	15000	5	No	Recovered	4	36-50	1
2	60	Male	Diabetes	Insulin Therapy	2000	3	Yes	Stable	3	51-65	1
3	32	Female	Fractured Arm	X-Ray and Splint	500	1	No	Recovered	5	19-35	1
4	75	Male	Stroke	CT Scan and Medication	10000	7	Yes	Stable	2	65+	1
...	...	...	...	...	...	...	...	...	...	...	...
13	30	Female	Childbirth	Delivery and Postnatal Care	12000	3	No	Recovered	4	19-35	1
12	65	Male	Prostate Cancer	Radiation Therapy	20000	9	No	Recovered	3	51-65	1
11	48	Female	Respiratory Infection	Antibiotics and Rest	800	2	No	Stable	4	36-50	1
10	25	Male	Allergic Reaction	Epinephrine Injection	100	1	No	Recovered	5	19-35	1
9	70	Female	Heart Attack	Cardiac Catheterization	18000	8	Yes	Stable	2	65+	1

984 rows × 1 columns

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

# Distribution of age groups
df["Age_Group"] = pd.cut(df["Age"], bins=[0, 18, 35, 50, 65, 100], labels=['0-18', '19-35', '36-50', '51-65', '65+'])
print(df["Age_Group"].value_counts())
```

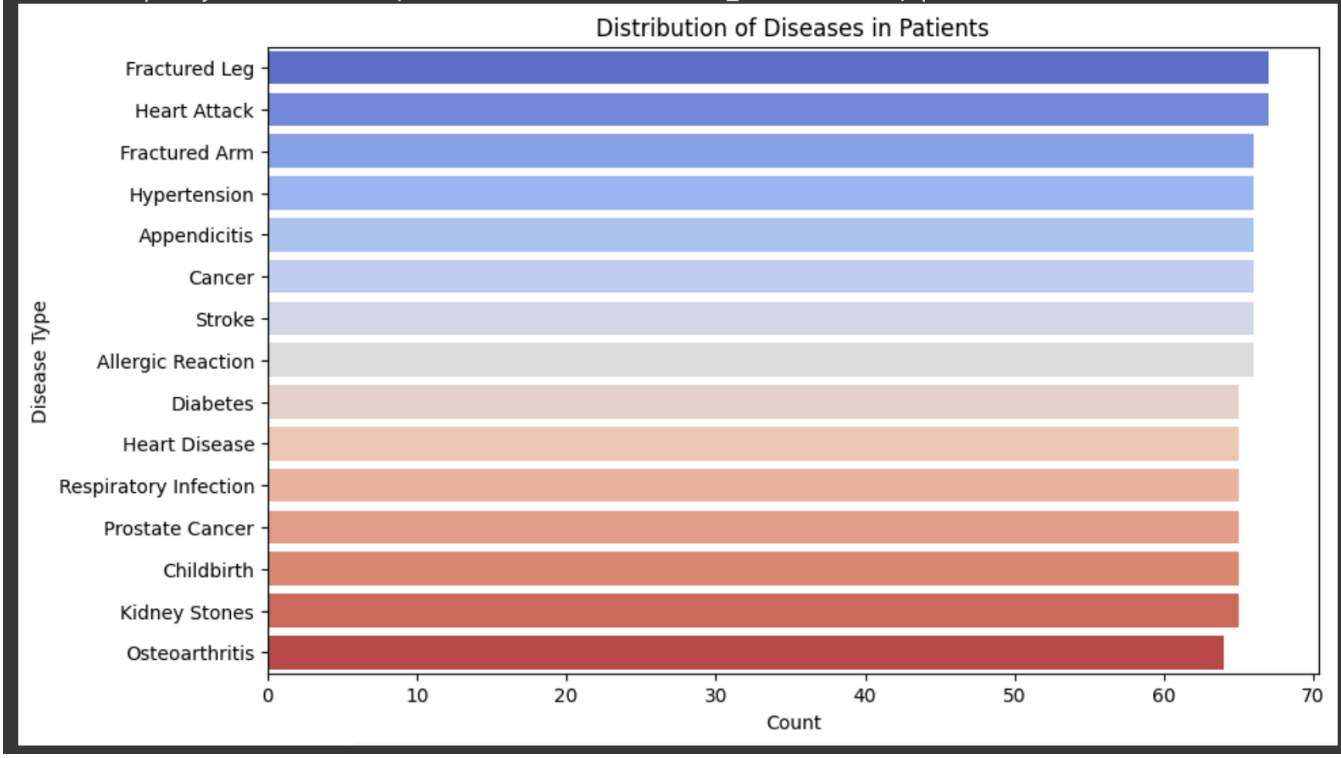
```
       Patient_ID         Age          Cost  Length_of_Stay  Satisfaction
count   984.000000  984.000000    984.000000      984.000000    984.000000
mean    500.329268   53.754065   8367.479675       37.663618      3.598577
std     288.979531   14.941135   7761.990976       19.595805      0.883002
min       1.000000   25.000000    100.000000        1.000000      2.000000
25%     250.750000   45.000000   1000.000000       21.000000      3.000000
50%     500.500000   55.000000   6000.000000       38.000000      4.000000
75%     750.250000   65.000000  15000.000000       54.250000      4.000000
max    1000.000000   78.000000  25000.000000       76.000000      5.000000
Age_Group
51-65    391
65+      231
19-35    197
36-50    165
0-18       0
Name: count, dtype: int64
```

```
print(df["Condition"].value_counts())
```

```
Condition
Fractured Leg            67
Heart Attack             67
Fractured Arm            66
Hypertension             66
Appendicitis             66
Cancer                   66
Stroke                   66
Allergic Reaction        66
Diabetes                 65
Heart Disease            65
Respiratory Infection    65
Prostate Cancer          65
Childbirth               65
Kidney Stones            65
Osteoarthritis           64
Name: count, dtype: int64
```

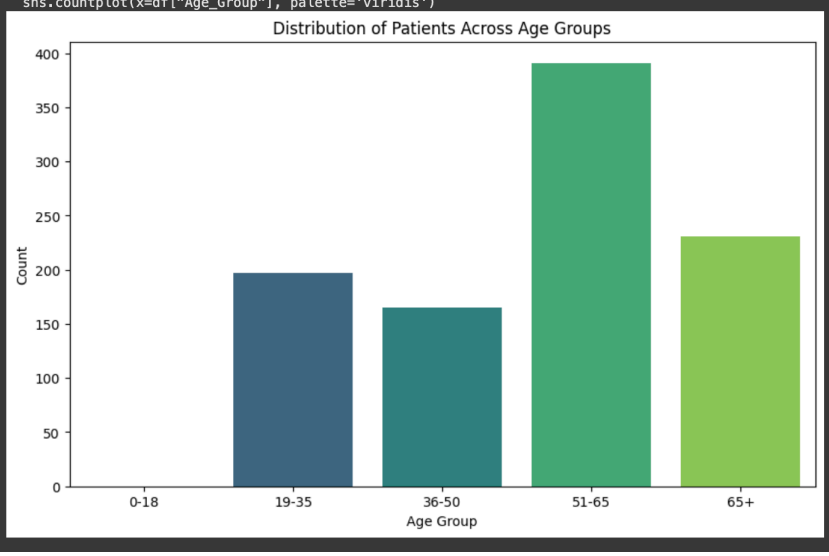
```
plt.figure(figsize=(10, 6))
sns.countplot(y=df["Condition"], order=df["Condition"].value_counts().index, palette='coolwarm')
plt.xlabel("Count")
plt.ylabel("Disease Type")
plt.title("Distribution of Diseases in Patients")
plt.show()
```

```
<ipython-input-17-41142e243604>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and

  sns.countplot(y=df["Condition"], order=df["Condition"].value_counts().index, palette='coolwarm')
```

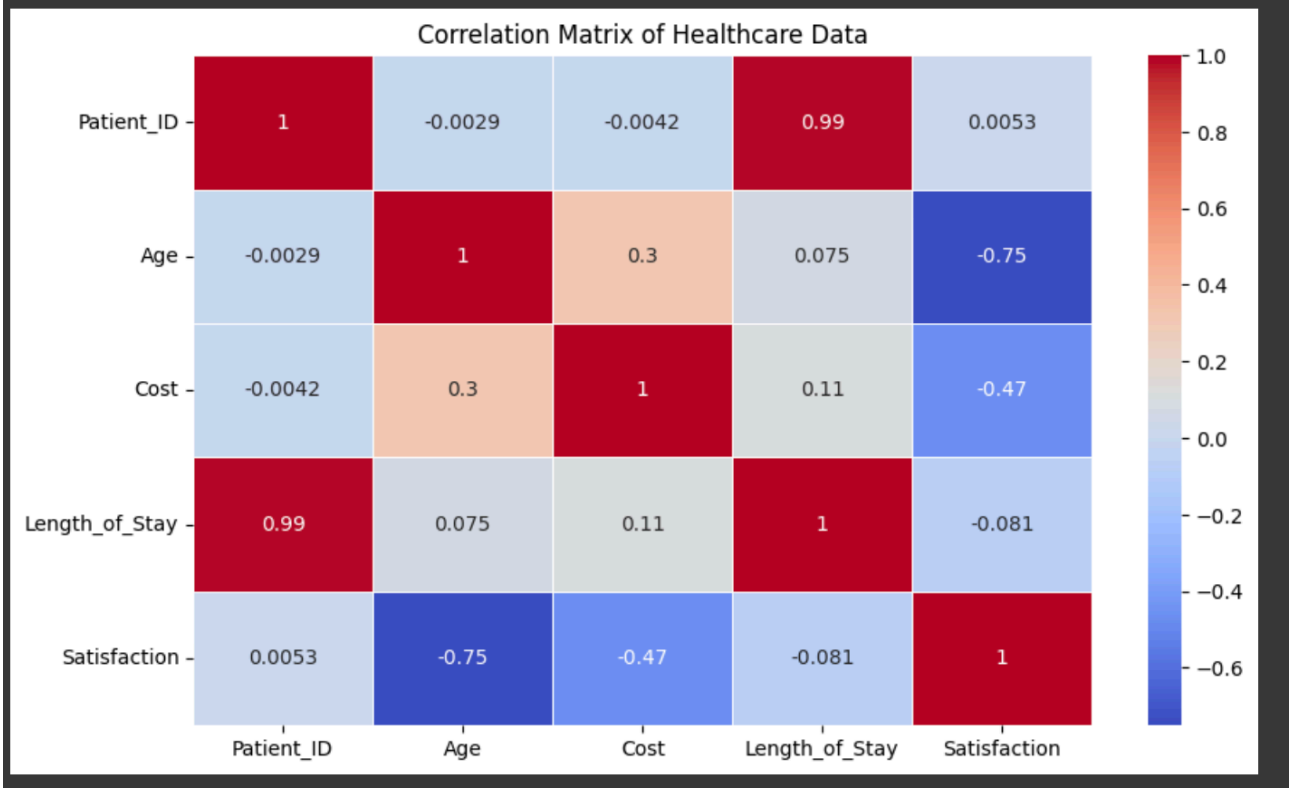



```
plt.figure(figsize=(10, 6))
sns.countplot(x=df["Age_Group"], palette='viridis')
plt.xlabel("Age Group")
plt.ylabel("Count")
plt.title("Distribution of Patients Across Age Groups")
plt.show()
```

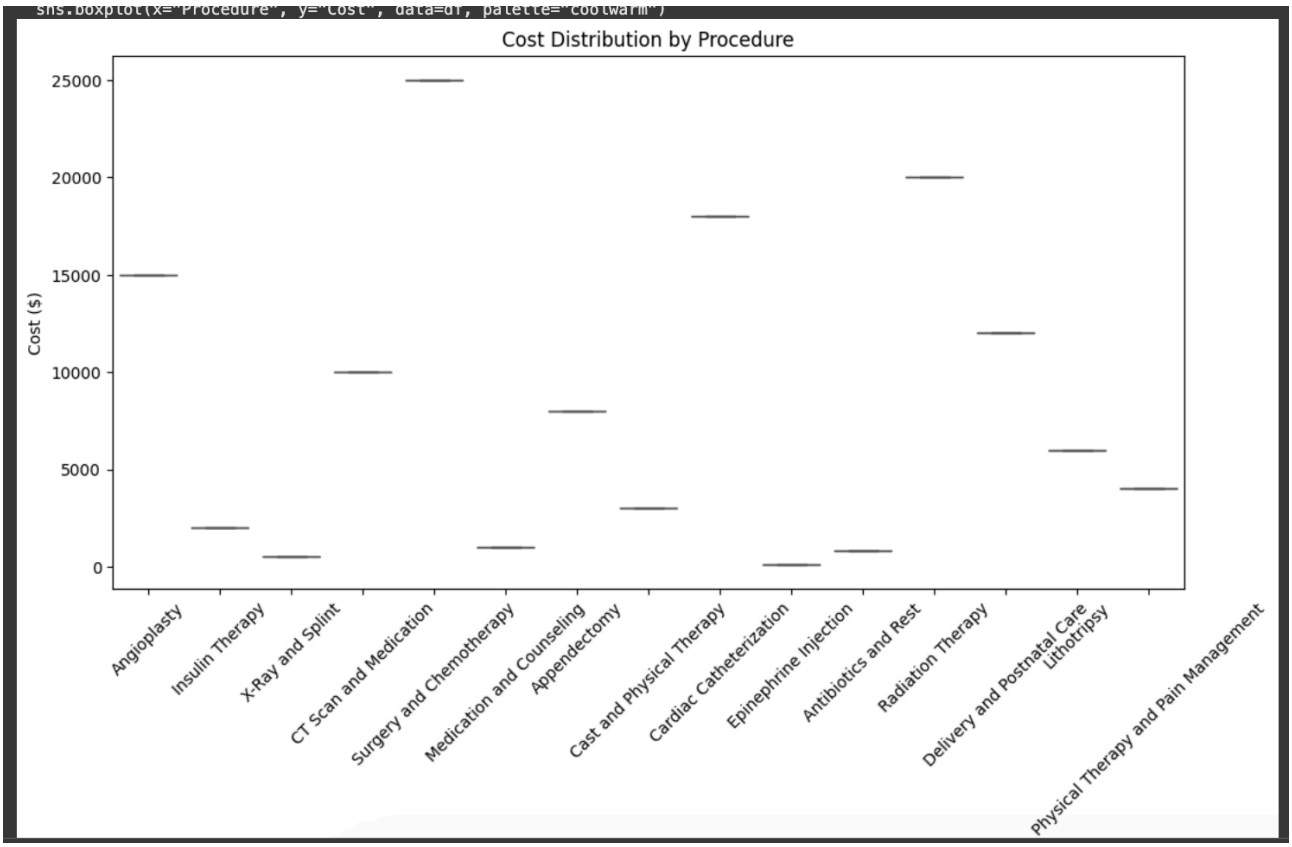
```
<ipython-input-13-669ae8b2fd6d>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=df["Age_Group"], palette='viridis')
```



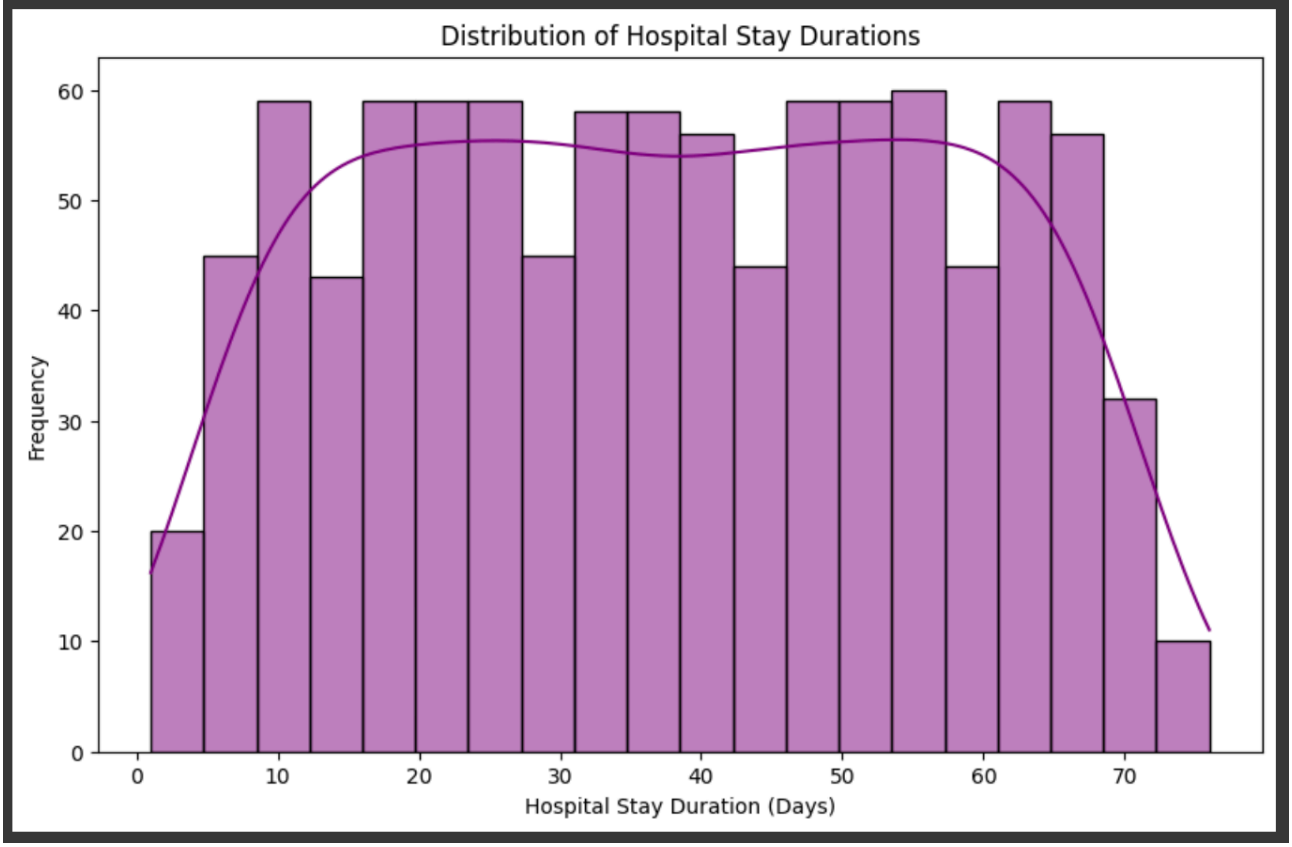
```
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Matrix of Healthcare Data")
plt.show()
```



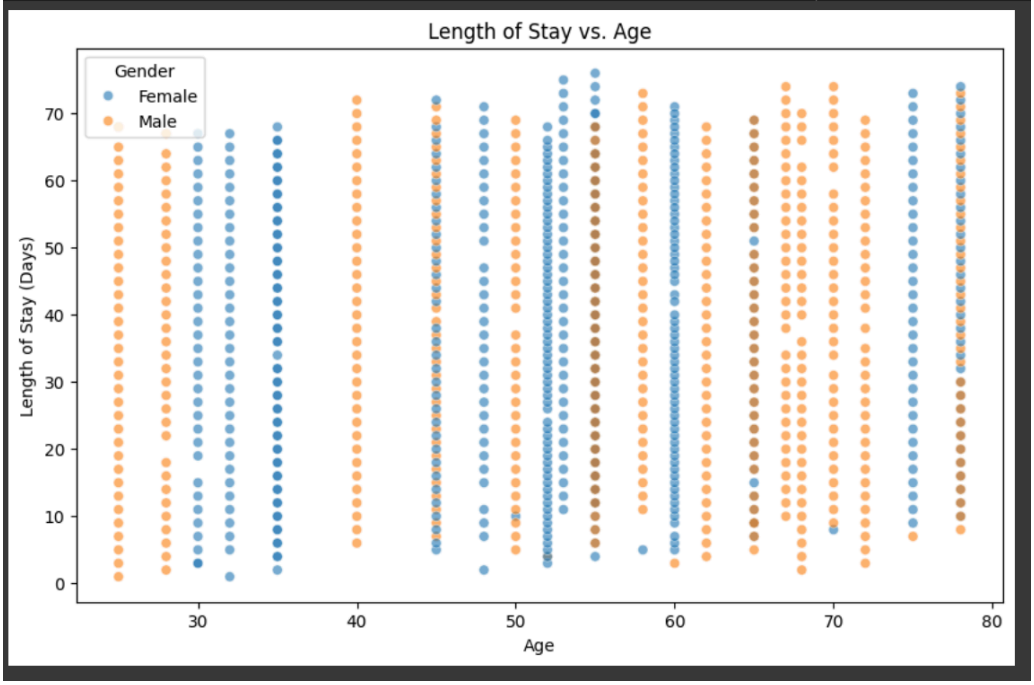

```
sns.boxplot(x="Procedure", y="Cost", data=df, palette="coolwarm")
```



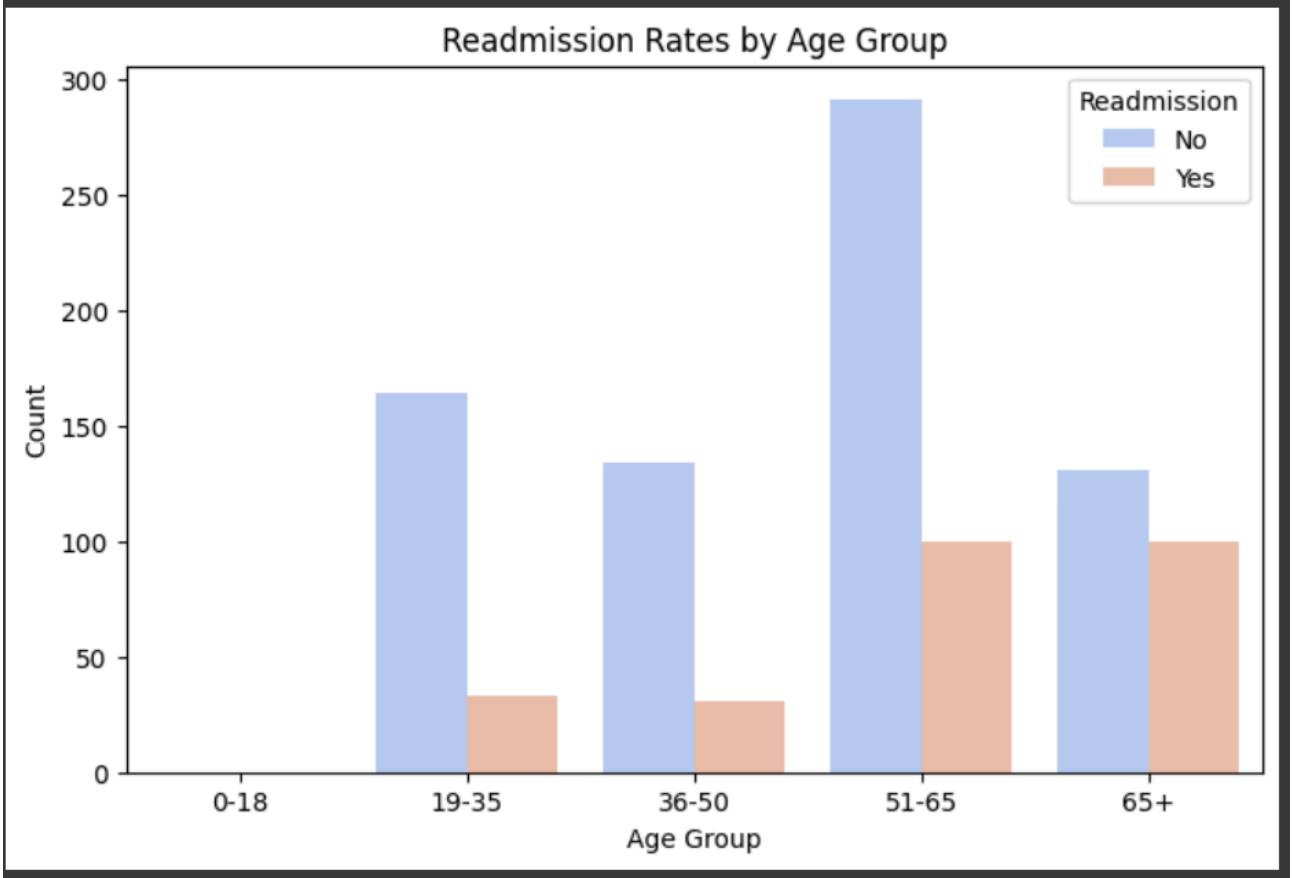
```
plt.figure(figsize=(10, 6))
sns.histplot(df["Length_of_Stay"], bins=20, kde=True, color='purple')
plt.xlabel("Hospital Stay Duration (Days)")
plt.ylabel("Frequency")
plt.title("Distribution of Hospital Stay Durations")
plt.show()
```




```
# Visualization 5: Length of Stay vs. Age
plt.figure(figsize=(10, 6))
sns.scatterplot(x="Age", y="Length_of_Stay", hue="Gender", data=df, alpha=0.6)
plt.title("Length of Stay vs. Age")
plt.xlabel("Age")
plt.ylabel("Length of Stay (Days)")
plt.legend(title="Gender")
plt.show()
```

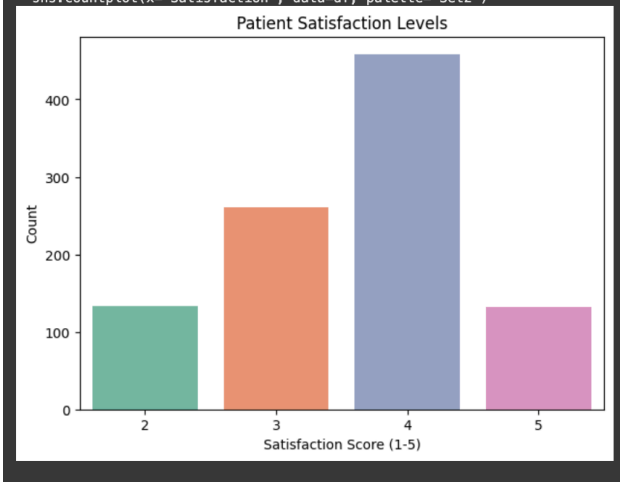


```python
# Visualization 6: Readmission Rates by Age Group
plt.figure(figsize=(8, 5))
sns.countplot(x="Age_Group", hue="Readmission", data=df, palette="coolwarm")
plt.title("Readmission Rates by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Count")
plt.show()
```



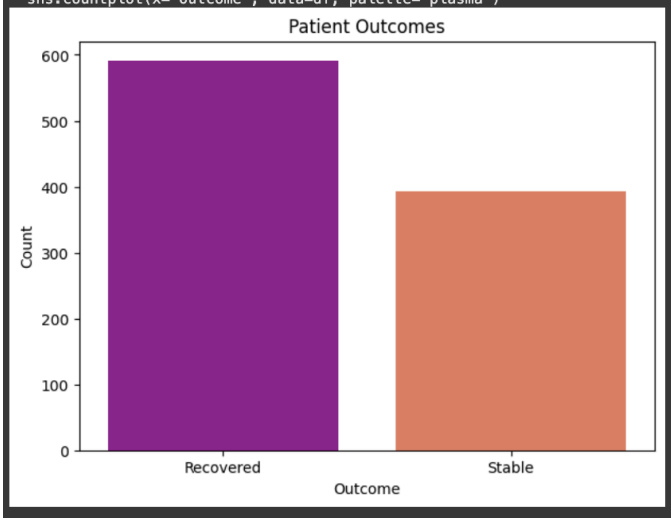

```python
# Visualization 7: Satisfaction Level Distribution
plt.figure(figsize=(7, 5))
sns.countplot(x="Satisfaction", data=df, palette="Set2")
plt.title("Patient Satisfaction Levels")
plt.xlabel("Satisfaction Score (1-5)")
plt.ylabel("Count")
plt.show()
```

```
<ipython-input-22-5af79c0a4fe2>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x="Satisfaction", data=df, palette="Set2")
```



```python
# Visualization 8: Outcome Distribution
plt.figure(figsize=(7, 5))
sns.countplot(x="Outcome", data=df, palette="plasma")
plt.title("Patient Outcomes")
plt.xlabel("Outcome")
plt.ylabel("Count")
plt.show()
```

```
<ipython-input-23-146143c2c988>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same

  sns.countplot(x="Outcome", data=df, palette="plasma")
```




```python
# Visualization 10: Cost vs. Length of Stay
plt.figure(figsize=(10, 6))
sns.scatterplot(x="Length_of_Stay", y="Cost", hue="Outcome", data=df, alpha=0.7)
plt.title("Cost vs. Length of Stay")
plt.xlabel("Length of Stay (Days)")
plt.ylabel("Cost ($)")
plt.legend(title="Outcome")
plt.show()
```