Name: Supriya Kumari
Email: supriya20122002@gmail.com
Phone: 9319210309
Experience: No

Screenshots

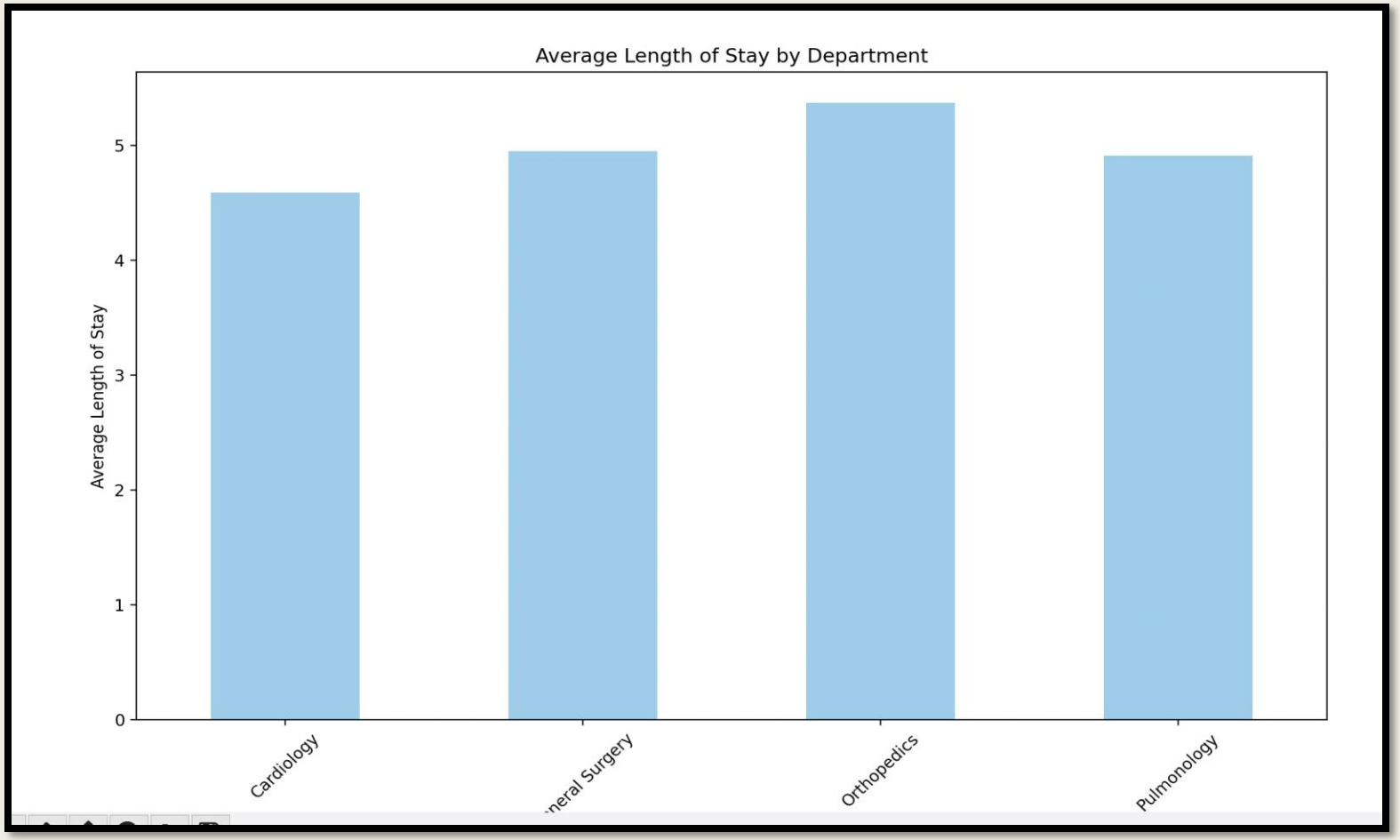
Initial Dataset:

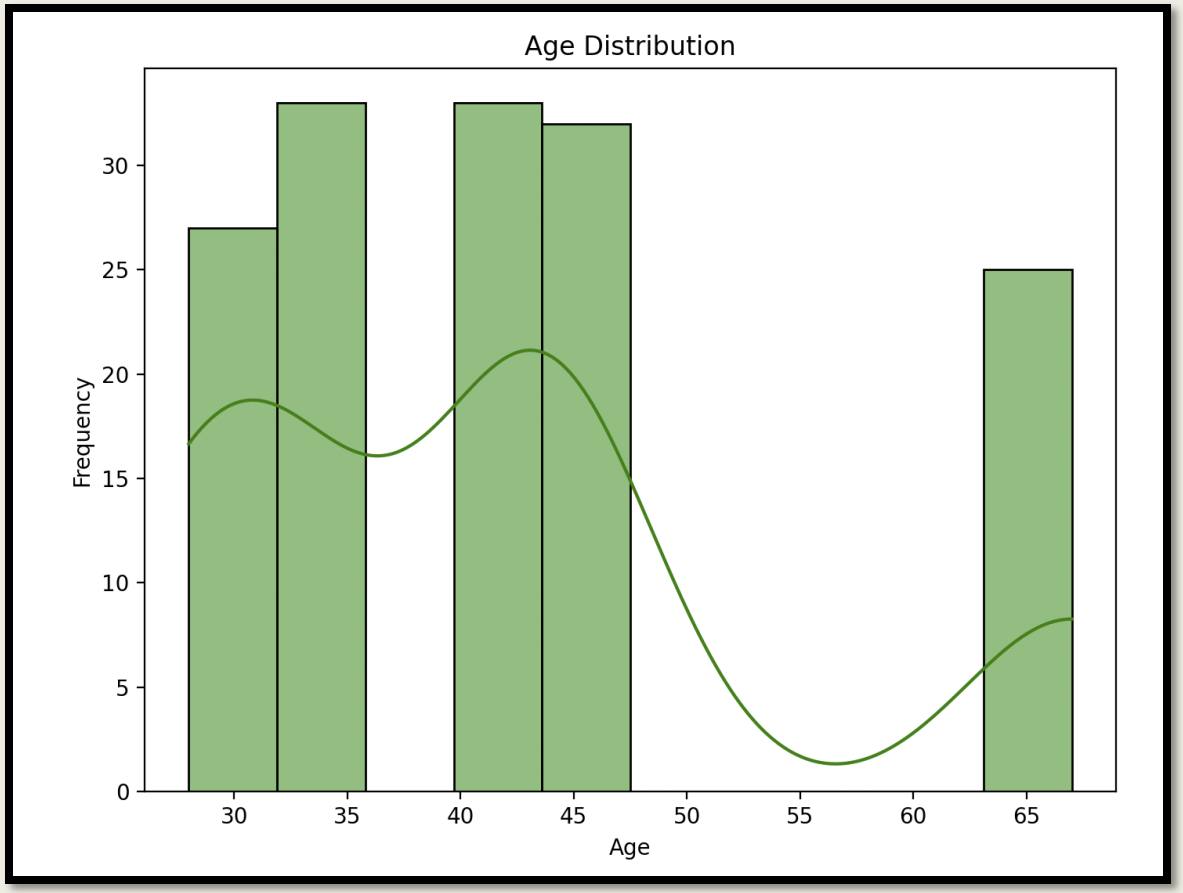
	Patient ID	Age	Diagnosis	Length of Stay	Hospital Department
0	1001	NaN	Appendicitis	7	Pulmonology
1	1002	45.0	Fracture	8	Orthopedics
2	1003	45.0	Fracture	9	Orthopedics
3	1004	NaN	Appendicitis	2	General Surgery
4	1005	28.0	Appendicitis	4	Cardiology
5	1006	NaN	Hypertension	4	Orthopedics
6	1007	28.0	Hypertension	7	Cardiology
7	1008	32.0	Appendicitis	4	Cardiology
8	1009	32.0	Hypertension	5	Cardiology
9	1010	NaN	Fracture	6	Cardiology
10	1011	45.0	Fracture	6	Pulmonology
11	1012	45.0	Hypertension	3	Pulmonology
12	1013	67.0	Pneumonia	8	Orthopedics
13	1014	45.0	Hypertension	7	Cardiology
14	1015	32.0	Appendicitis	7	Cardiology
15	1016	45.0	Fracture	4	Cardiology

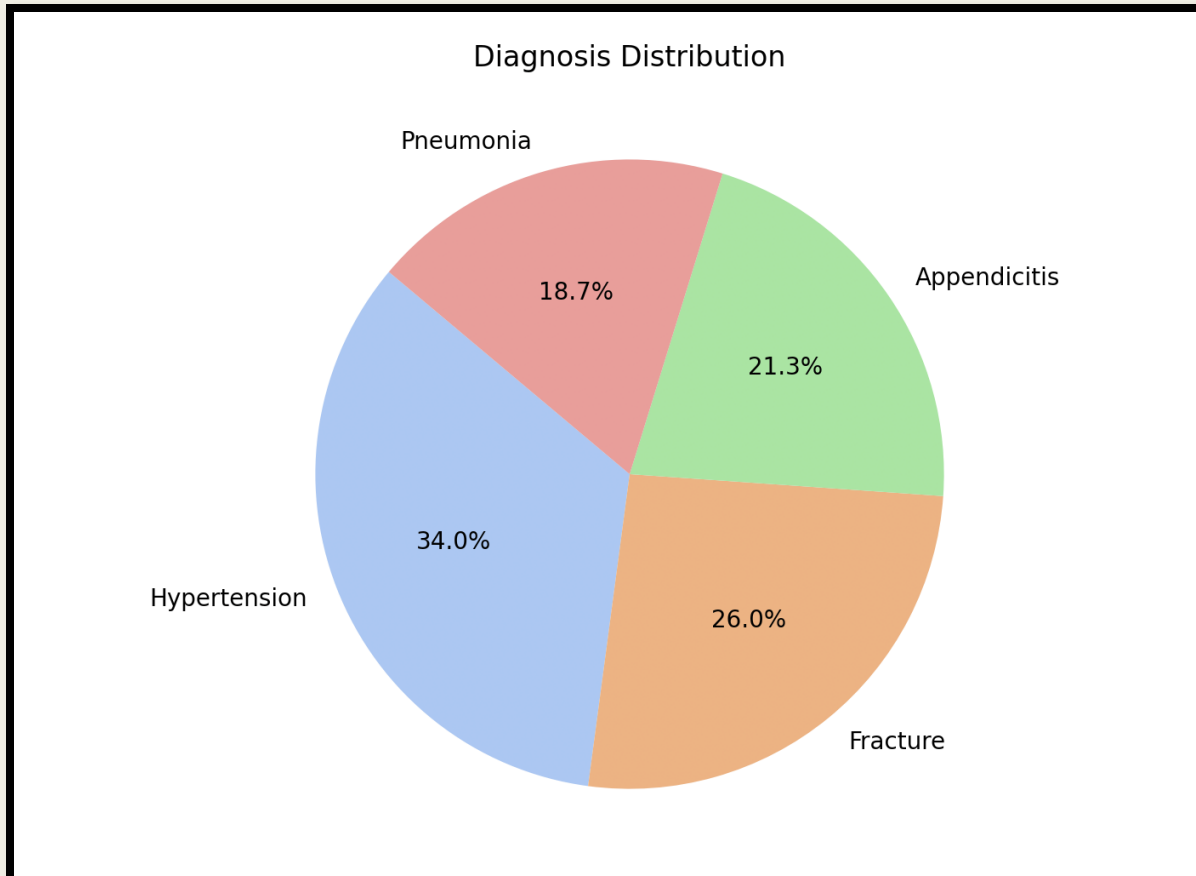
Cleaned Dataset:

	Patient ID	Age	Diagnosis	Length of Stay	Hospital Department
0	1001	42.111111	Appendicitis	7	Pulmonology
1	1002	45.000000	Fracture	8	Orthopedics
2	1003	45.000000	Fracture	9	Orthopedics
3	1004	42.111111	Appendicitis	2	General Surgery
4	1005	28.000000	Appendicitis	4	Cardiology
5	1006	42.111111	Hypertension	4	Orthopedics
6	1007	28.000000	Hypertension	7	Cardiology
7	1008	32.000000	Appendicitis	4	Cardiology
8	1009	32.000000	Hypertension	5	Cardiology
9	1010	42.111111	Fracture	6	Cardiology
10	1011	45.000000	Fracture	6	Pulmonology
11	1012	45.000000	Hypertension	3	Pulmonology
12	1013	67.000000	Pneumonia	8	Orthopedics
13	1014	45.000000	Hypertension	7	Cardiology
14	1015	32.000000	Appendicitis	7	Cardiology
15	1016	45.000000	Fracture	4	Cardiology

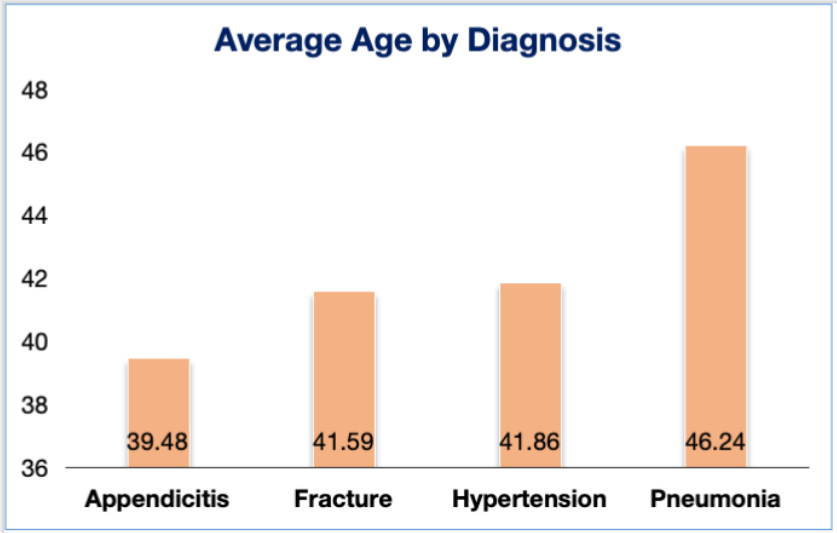
```
Statistical Analysis:
          Metric      Mean    Median  Standard Deviation
0            Age  42.111111  42.111111          12.823938
1  Length of Stay   4.980000   5.000000           2.550089
2025-01-27 10:32:10.671 Python[27128:3012120] +[IMKClient subclass]: chose IMKClient_Modern
2025-01-27 10:32:10.671 Python[27128:3012120] +[IMKInputSession subclass]: chose IMKInputSession_Modern
ritikseni@MacBook Air 4 Big Data Analytics %
```

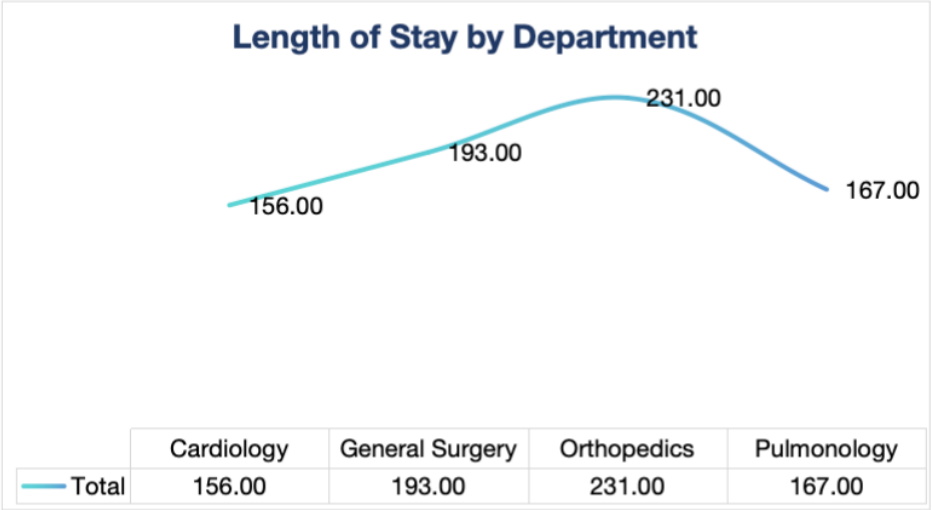


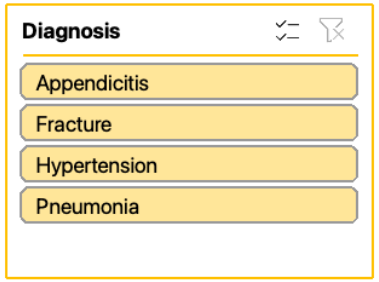

Age Distribution
Frequency
0
5
10
15
20
25
30
30
35
40
45
50
55
60
65
Age

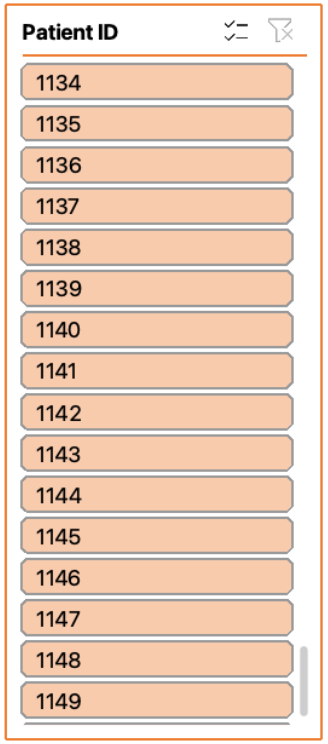

Diagnosis Distribution
Pneumonia
18.7%
Appendicitis
21.3%
34.0%
Hypertension
26.0%
Fracture

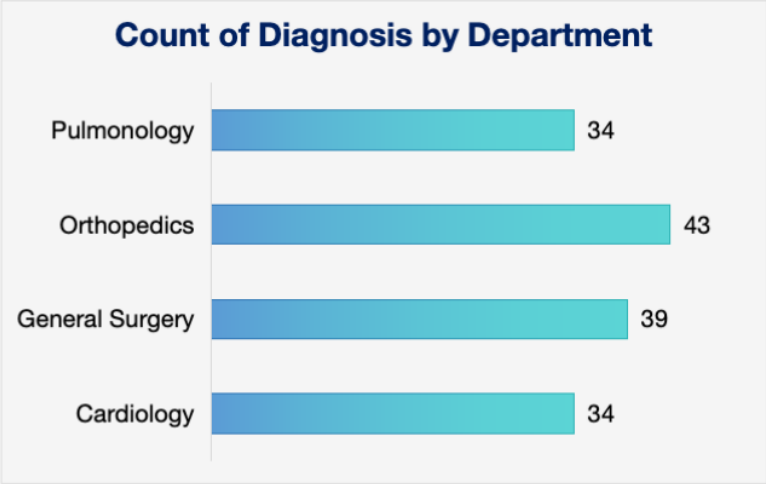
Dashboard

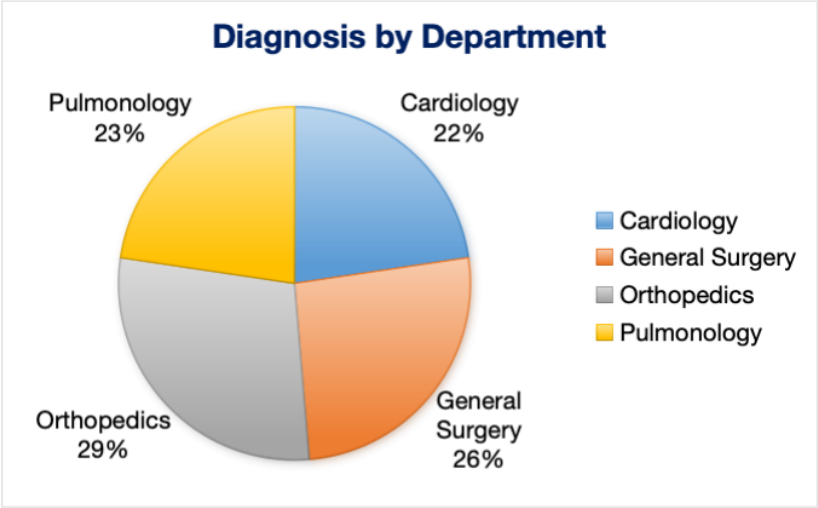

Average Age by Diagnosis
48
46
44
42
40
38
36
39.48
41.59
41.86
46.24
Appendicitis
Fracture
Hypertension
Pneumonia


Length of Stay by Department
156.00
193.00
231.00
167.00
Cardiology
General Surgery
Orthopedics
Pulmonology
Total
156.00
193.00
231.00
167.00


Diagnosis
Appendicitis
Fracture
Hypertension
Pneumonia


Patient ID
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149


Count of Diagnosis by Department
Pulmonology
34
Orthopedics
43
General Surgery
39
Cardiology
34


Diagnosis by Department
Pulmonology
23%
Cardiology
22%
Orthopedics
29%
General
Surgery
26%
Cardiology
General Surgery
Orthopedics
Pulmonology

Codebase Environment

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path

file_path = Path.home() / 'Downloads' / 'Chrome Downloads' / 'hospital_patient_data (1).csv'
df = pd.read_csv(file_path)

print("Initial Dataset:")
print(df.to_string())

df = df.drop_duplicates()
df['Age'].fillna(df['Age'].mean(), inplace=True)

if 'Hospital Department' in df.columns:
    df['Hospital Department'] = df['Hospital Department'].str.title()

df.to_csv("cleaned_hospital_data.csv", index=False)
print("Cleaned Dataset:")
print(df.to_string())

stats = {
    "Metric": ["Age", "Length of Stay"],
    "Mean": [df["Age"].mean(), df["Length of Stay"].mean()],
    "Median": [df["Age"].median(), df["Length of Stay"].median()],
    "Standard Deviation": [df["Age"].std(), df["Length of Stay"].std()]
}

stats_df = pd.DataFrame(stats)
print("Statistical Analysis:")
print(stats_df)

avg_stay = df.groupby("Hospital Department")["Length of Stay"].mean()

plt.figure(figsize=(8, 6))
avg_stay.plot(kind="bar", color="skyblue")
plt.title("Average Length of Stay by Department")
plt.xlabel("Hospital Department")
plt.ylabel("Average Length of Stay")
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(8, 6))
sns.histplot(df["Age"], bins=10, kde=True, color="green")
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()

diag_counts = df["Diagnosis"].value_counts()

plt.figure(figsize=(8, 6))
diag_counts.plot(kind="pie", autopct='%1.1f%%', startangle=140, colors=sns.color_palette("pastel"))
plt.title("Diagnosis Distribution")
plt.ylabel("")
plt.show()
```

Big Data Analytics

EXPLORER:...

- archive (1).zip
- archive.zip
- cardekho.csv
- cleaned_hospital_data.csv
- cleaned_transformed_data.csv
- da1.py
- Data_Analytics.py
- Exp_1_DV
- Exp_1_DV.py
- Exp_1.py
- Exp_2_DV.py
- Exp_2.java
- Exp_3_DV.py
- Exp1_dataset.csv
- hospital_data.csv
- vgsales new.csv

da1.py > ...

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path

file_path = Path.home() / 'Downloads' / 'Chrome Downloads' / 'hospital_pa
df = pd.read_csv(file_path)

print("Initial Dataset:")
print(df.to_string())

df = df.drop_duplicates()
df['Age'].fillna(df['Age'].mean(), inplace=True)

if 'Hospital Department' in df.columns:
    df['Hospital Department'] = df['Hospital Department'].str.title()

df.to_csv("cleaned_hospital_data.csv", index=False)
print("Cleaned Dataset:")
print(df.to_string())

stats = {
    "Metric": ["Age", "Length of Stay"],
    "Mean": [df["Age"].mean(), df["Length of Stay"].mean()],
    "Median": [df["Age"].median(), df["Length of Stay"].median()],
```

Challenges:

- **Handling Missing Data**: Missing values can distort results, requiring decisions on filling or removing them. Excel and VS Code both offer methods, but filling in large datasets manually in Excel is time-consuming.

- **Data Cleaning and Duplicate Removal**: Identifying duplicates, especially in large datasets, is challenging. Excel provides manual tools, but Python's pandas in VS Code automates this efficiently.

- **Data Transformation and Standardization**: Inconsistent formats (e.g., department names) require standardization. Excel offers manual solutions, while Python handles this faster for large datasets.

- **Creating Interactive Dashboards**: Excel's basic charts can be limiting for interactive dashboards. Power BI or Tableau might be more effective, but Excel offers the necessary functionality with learning.

- **Data Visualization Limitations**: Excel struggles with large datasets, while Python in VS Code offers more customization. Both tools have their strengths depending on the visualization complexity.