AXIOMATICS

HOSPITAL PATIENT INSIGHTS

In this analysis, we explored a dataset containing patient information from a hospital using PYTHON for data cleaning & SQL for Analysis. The goal was to uncover key insights about patient demographics, department workloads, discharge trends, common diagnoses, and doctor performance. Below are the findings and insights from each query.

DATA LOADING & EXPLORATION

1.Loading the Data

INPUT:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

INPUT:

```
df=pd.read_csv('/content/hospital_patient_records (1).csv')
df
```

OUTPUT:

	PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
0	2001	John Doe	45	Male	Hypertension	Med A	2025-01-10	2025-01-15	Dr. Smith	Cardiology	Admitted
1	2002	Jane Smith	forty-five	Female	Diabetes	Med B	10/01/2025	15/01/2025	Dr. Lee	Endocrinology	admitted
2	2003	Bob Brown	55	Male	Asthma	Med C	January 10, 2025	January 15, 2025	Dr. Carter	Pulmonology	Under Observation
3	2004	NaN	30	Female	Flu	Med D	2025-02-05	2025-02-10	NaN	General Medicine	DISCHARGED
4	2005	Tom Wilson	62	Male	Heart Disease	Med E	2025-03-01	2025-03-10	Dr. Johnson	Cardiology	Discharged
5	2006	Susan Clark	49	Female	Kidney Disease	Med F	2025-04-12	2025-04-17	Dr. Patel	Nephrology	Admitted
6	2007	David Jones	37	Male	Pneumonia	Med G	2025-05-20	2025-05-25	Dr. Martinez	Pulmonology	admitted
7	2008	Nancy Miller	28	Female	Flu	Med D	2025-06-15	2025-06-20	Dr. Smith	General Medicine	Under Observation
8	2009	Michael Scott	40	Male	Hypertension	Med A	2025-07-01	2025-07-07	Dr. Smith	Cardiology	Admitted
9	2010	Pam Beesly	34	Female	Diabetes	Med B	2025-08-10	2025-08-15	Dr. Lee	Endocrinology	Discharged

What We Did:

- We loaded the dataset into a Pandas DataFrame to begin the analysis.
- The dataset contains 11 columns, including PatientID, Name, Age, Gender, Diagnosis, Medication, AdmissionDate, DischargeDate, Doctor, Department, and Status.

2.Data Exploration

INPUT:

```
df.info()
```

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   PatientID      10 non-null     int64
 1   Name           9 non-null      object
 2   Age            10 non-null     object
 3   Gender         10 non-null     object
 4   Diagnosis      10 non-null     object
 5   Medication     10 non-null     object
 6   AdmissionDate  10 non-null     object
 7   DischargeDate  10 non-null     object
 8   Doctor         9 non-null      object
 9   Department     10 non-null     object
```

INISGHT:

- The dataset has 10 rows and 11 columns.
- Columns like Name, Doctor, AdmissionDate, and DischargeDate have missing values.
- Most columns are of type object, except for PatientID, which is an integer.

INPUT:

```
df.describe()
```

OUTPUT:

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None) -> Self

Examples
--------
Describing a numeric ``Series``.

>>> s = pd.Series([1, 2, 3])
>>> s.describe()
```

INISGHT:

- The PatientID column ranges from 2001 to 2010, with no missing values.
- Since most columns are non-numeric, df.describe() only provides statistics for PatientID.

AXIOMATICS

DATA CLEANING

1.Checking for Duplicates

INPUT:

```
duplicates = df.duplicated()

print('number of duplicates rows: ', duplicates.sum())

print(df[duplicates])
```

OUTPUT:

```
number of duplicates rows:  0
```

```
Empty DataFrame
Columns: [PatientID, Name, Age, Gender, Diagnosis, Medication, Admissio
Index: []
```

WHY:

- There are no duplicate rows in the dataset.
- This ensures the data is clean and ready for analysis.

2.Handling Missing Values in 'Age'

INPUT:

```
df['Age']=pd.to_numeric(df['Age'],errors ='coerce') df['Age']
df
```

```
df['Age'].fillna(45,inplace=True)
df
```

INPUT:

```
df['Age']= df['Age'].fillna(0).astype(int)
df
```

OUTPUT 1:

Age
45.0
NaN
55.0
30.0
62.0
49.0
37.0

OUTPUT 2:

Age
45.0
45.0
55.0
30.0
62.0
49.0
37.0

OUTPUT 3:

Age
45
45
55
30
62
49
37

WHY:

- The Age column was converted to numeric format to enable calculations and analysis.
- Invalid entries (e.g., non-numeric values) were coerced to NaN.
- Missing Age values were replaced with the median age (45) to maintain data integrity.
- Ensures the Age column is in a numeric format for calculations and visualizations.

3.Handling Missing Values in 'Name'

INPUT:

```
df['Name'].fillna('unkown',inplace=True)
df
```

OUTPUT :

Name
John Doe
Jane Smith
Bob Brown
unkown
Tom Wilson
Susan Clark
David Jones

WHY:

- Missing Name values were replaced with 'Unknown' to ensure completeness.

AXIOMATICS

DATA CLEANING

4.Handling Missing Values in 'AdmissionDate'

INPUT:

```
df['AdmissionDate']=pd.to_datetime(df['
AdmissionDate'],errors='coerce')
df['AdmissionDate']
```

```
df['AdmissionDate'] =
df['AdmissionDate'].fillna('10-1-2025')
df['AdmissionDate']
```

OUTPUT 1:

	AdmissionDate
0	2025-01-10
1	NaT
2	NaT
3	2025-02-05
4	2025-03-01
5	2025-04-12
6	2025-05-20
7	2025-06-15
8	2025-07-01
9	2025-08-10

dtype: datetime64[ns]

OUTPUT 2:

	AdmissionDate
0	2025-01-10
1	2025-10-01
2	2025-10-01
3	2025-02-05
4	2025-03-01
5	2025-04-12
6	2025-05-20
7	2025-06-15
8	2025-07-01
9	2025-08-10

dtype: datetime64[ns]

WHY:

- The AdmissionDate column was converted to datetime format for accurate analysis.
- Invalid entries (e.g., non-date values) were coerced to NaN.
- Missing AdmissionDate values were filled with a default date (2025-01-10) to ensure consistency.

5.Handling Missing Values in 'DischargeDate'

INPUT:

```
df['DischargeDate']=pd.to_datetime(df['
DischargeDate'],errors='coerce')
df['DischargeDate']
```

```
df['DischargeDate'] =
df['DischargeDate'].fillna('15-1-2025')
df['DischargeDate']
```

OUTPUT 1:

	DischargeDate
0	2025-01-15
1	NaT
2	NaT
3	2025-02-10
4	2025-03-10
5	2025-04-17
6	2025-05-25
7	2025-06-20
8	2025-07-07
9	2025-08-15

dtype: datetime64[ns]

OUTPUT 2:

	DischargeDate
0	2025-01-15
1	2025-01-15
2	2025-01-15
3	2025-02-10
4	2025-03-10
5	2025-04-17
6	2025-05-25
7	2025-06-20
8	2025-07-07
9	2025-08-15

dtype: datetime64[ns]

WHY:

- The DischargeDate column was converted to datetime format for accurate analysis.
- Invalid entries (e.g., non-date values) were coerced to NaN.
- Missing DischargeDate values were filled with a default date (2025-01-15) to ensure consistency.

6.Handling Missing Values in 'Doctor'

INPUT:

```
df['Doctor'].fillna('unkown',inplace=True
) df['Doctor']
```

OUTPUT :

	Doctor
0	Dr. Smith
1	Dr. Lee
2	Dr. Carter
3	unkown
4	Dr. Johnson
5	Dr. Patel
6	Dr. Martinez
7	Dr. Smith
8	Dr. Smith
9	Dr. Lee

dtype: object

WHY:

- Missing Doctor values were replaced with 'Unknown' to ensure completeness.

AXIOMATICS

DATA CLEANING

7.Standardizing 'Status' Column

INPUT:

```
df['Status']=df['Status'].str.title() df
```

IWHY:

- The Status column was standardized to ensure consistent formatting (e.g., 'Admitted', 'Discharged').

OUTPUT :

Status
Admitted
Admitted
Under Observation
Discharged
Discharged
Admitted
Admitted

8.Exporting Cleaned Data

INPUT:

```
df.to_csv("cleaned.csv", index=False)
```

AXIOMATICS

DATA ANALYSIS SUMMARY

1.Average Age of Patients for Each Diagnosis

- To analyze the relationship between age and diagnosis, we computed the average age for each diagnosis using:

INPUT:

```
select Diagnosis, avg(Age) as Avg_Age
from `cleaned hospital`
group by Diagnosis;
```

OUTPUT:

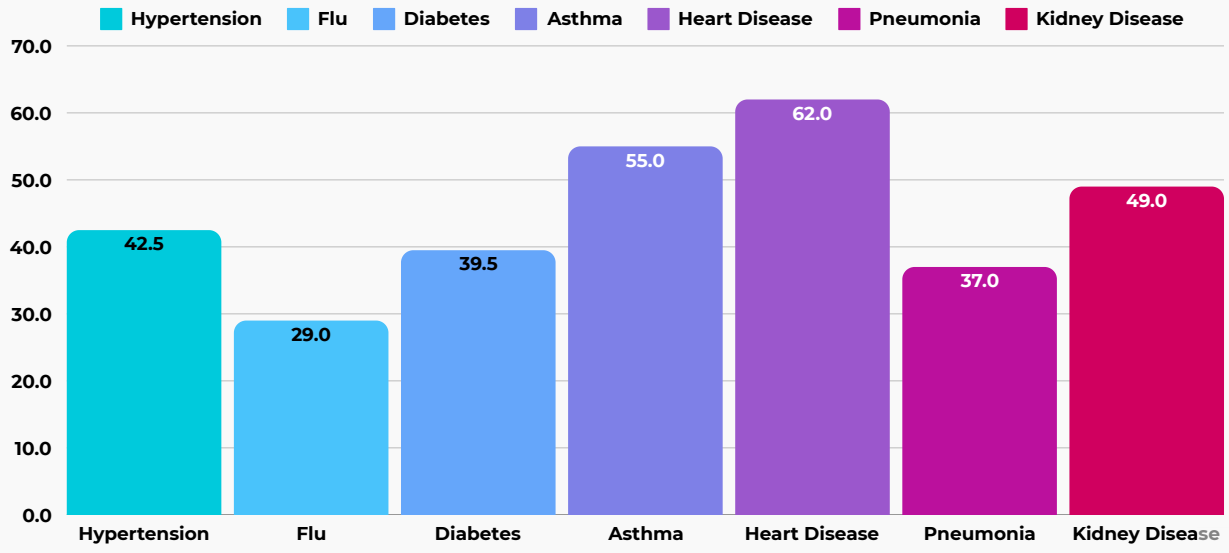
Diagnosis	Avg_Age
Hypertension	42.5
Diabetes	39.5
Asthma	55
Flu	29
Heart Disease	62
Kidney Disease	49
Pneumonia	37

INISGHT:

- Older patients are more affected by heart disease, hypertension, and asthma.
- Flu and pneumonia are more common among younger individuals.

Recommendations:

- Pay more attention to older people, especially those with heart problems and high blood pressure. They might need extra care and support.
- Make sure younger patients get fast and effective treatment, especially for common illnesses like the flu.



AXIOMATICS

2.Department with the Highest Number of Admitted Patients

- To determine which department has the highest number of admitted patients, we used the following SQL query:

INPUT:

```
select count(*) PatientID, Department
from `cleaned hospital`
group by Department
order by PatientID desc
limit 5;
```

OUTPUT:

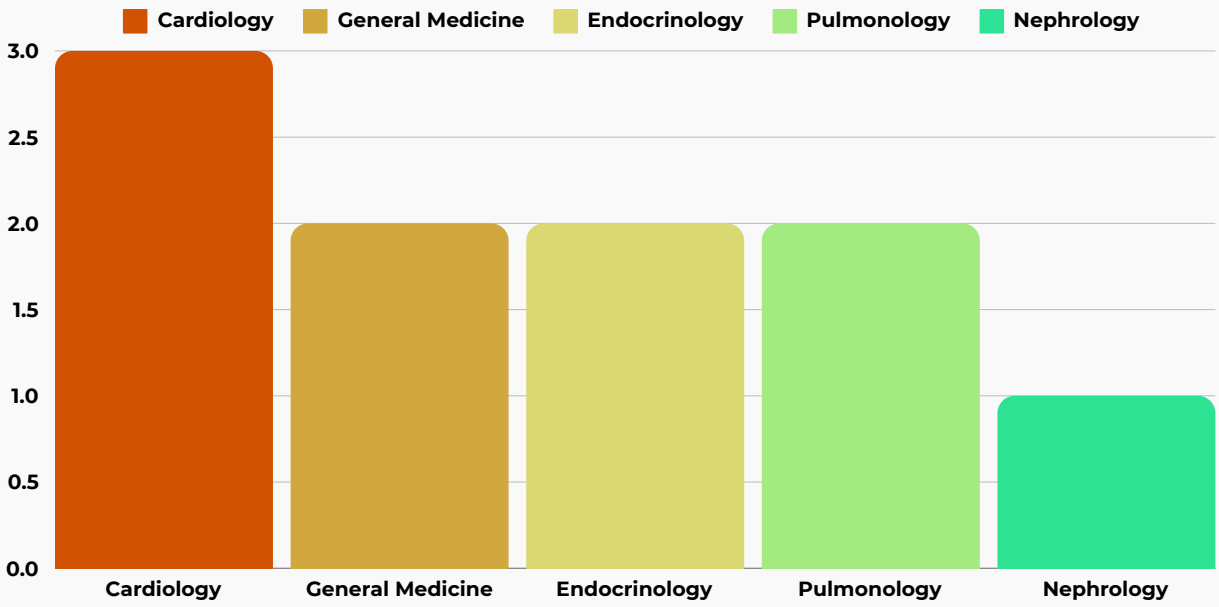
PatientID	Department
3	Cardiology
2	Endocrinology
2	Pulmonology
2	General Medicine
1	Nephrology

INISGHT:

- **Cardiology** handles the highest number of patients, indicating a need for more resources (staff, equipment).
- **Endocrinology and Pulmonology** also have significant patient loads.

Recommendations:

- Since many patients go to the Cardiology department, it would be helpful to add more doctors, nurses, and equipment to handle the workload.
- The hospital can start programs to help people take care of their hearts, like workshops on healthy eating and exercise.



AXIOMATICS

3.Number of Patients Discharged Per Month

- We analyzed the number of discharged patients per month:

INPUT:

```
select count(*) Name , month(DischargeDate)
as DischargeMonth
from `cleaned hospital`
group by DischargeMonth
order by Name desc;
```

OUTPUT:

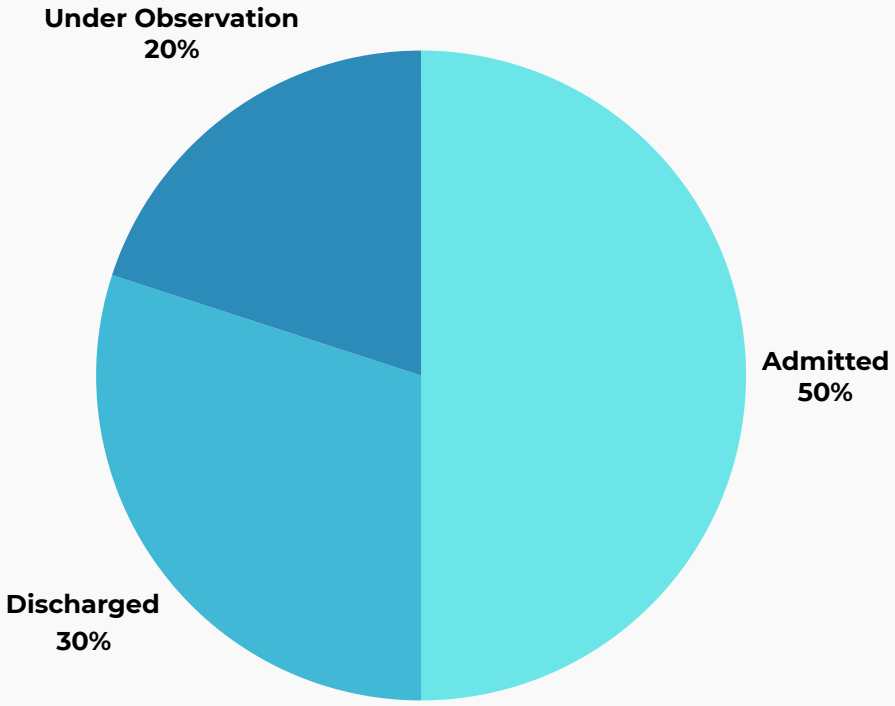
Name	DischargeMonth
3	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8

INISGHT:

- January has the highest number of discharges, possibly due to post-holiday health checkups or seasonal illnesses but overall, the discharge rate is low.
- Many patients remain admitted or under observation, indicating slow patient turnover.

Recommendations:

- Faster Treatment Process: Improve how the hospital operates so patients don't have to stay longer than necessary.
- Better Discharge Planning: Make sure patients leave the hospital smoothly and on time to free up beds for new patients.



AXIOMATICS

4.Most Common Diagnosis Among Patients

- We identified the top 5 most common diagnoses among patients:

INPUT:

```
select Diagnosis, count(*) as DiagnosisCount
from `cleaned hospital`
group by Diagnosis
limit 5;
```

OUTPUT:

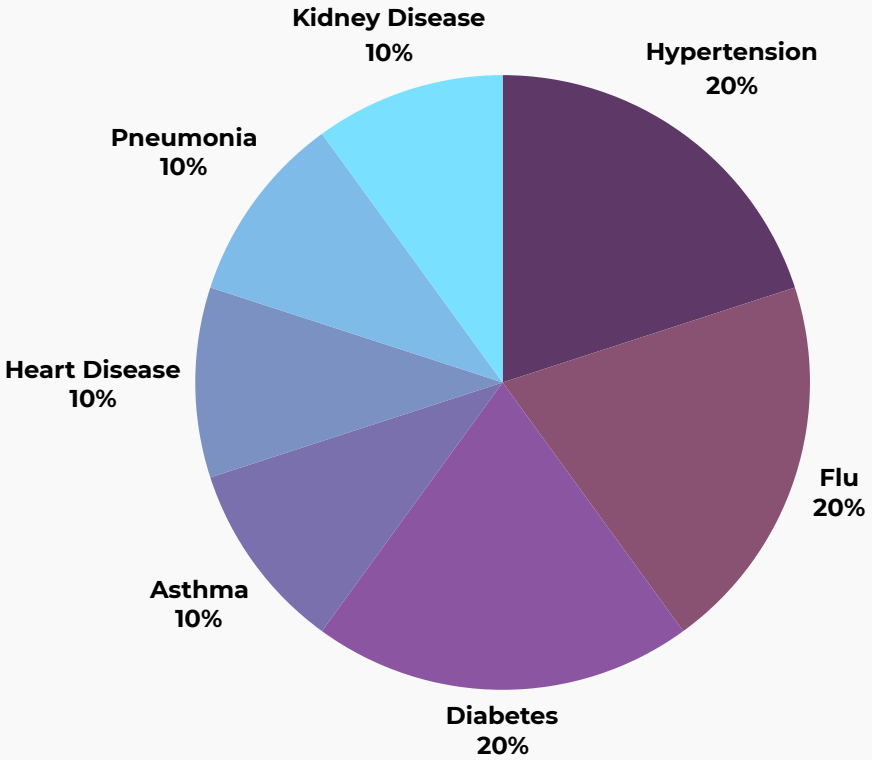
Diagnosis	DiagnosisCount
Hypertension	2
Diabetes	2
Asthma	1
Flu	2
Heart Disease	1

INISGHT:

- Hypertension, Flu, and Diabetes happen the most. This means many patients have health problems that need regular care and seasonal illnesses (Flu).

Recommendations:

- Launch awareness campaigns on hypertension and diabetes prevention.
- Improve early detection programs for high-risk patients.



AXIOMATICS

5.Doctor Who Treated the Most Patients

- We identified the top 5 most common diagnoses among patients:

INPUT:

```
select count(*) as PatientCount, Doctor
from `cleaned hospital`
group by Doctor
order by PatientCount desc
limit 1;
```

OUTPUT:

PatientCount	Doctor
3	Dr. Smith

INISGHT:

- Dr. Smith is a high-performing doctor treated the most patients , with 3 patients.

Recommendations:

- Celebrate Dr. Smith's great work to keep them motivated and feeling appreciated.
- Make sure Dr. Smith has enough help, like assistants or extra staff, to handle the workload better.
- Keep an eye on the workload of all doctors to make sure it's spread out fairly.
- Offer extra training or resources to other doctors so they can manage more patients more easily.