

1. Import Libraries and Load Dataset

```
import pandas as pd
import numpy as np

df = pd.read_excel("/content/patient_data.csv.xlsx")
```

```
print("🔴 First 5 rows:")
print(df.head())
```

```
print("\n🔴 Dataset Info:")
print(df.info())
```

```
print("\n🔴 Summary Statistics:")
print(df.describe(include='all'))
```

```
🔄 🔴 First 5 rows:
  patient_id  age  gender  department  length_of_stay
0         201   34   Male   Cardiology              5
1         202   56  Female   Neurology              7
2         203   45  Female   Oncology              3
3         204   67   Male   Cardiology              6
4         205   23  Female   Pediatrics             2
```

```
🔴 Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   patient_id      10 non-null    int64
1   age             10 non-null    int64
2   gender          10 non-null    object
3   department      10 non-null    object
4   length_of_stay  10 non-null    int64
dtypes: int64(3), object(2)
memory usage: 532.0+ bytes
None
```

```
🔴 Summary Statistics:
  patient_id      age  gender  department  length_of_stay
count    10.00000  10.00000    10         10    10.000000
unique      NaN      NaN      2         4         NaN
top         NaN      NaN   Male   Cardiology         NaN
freq         NaN      NaN    5         3         NaN
mean    205.50000  44.30000    NaN      NaN    5.300000
std      3.02765  14.047934   NaN      NaN    1.888562
min     201.00000  23.00000    NaN      NaN    2.000000
25%     203.25000  35.00000    NaN      NaN    4.250000
50%     205.50000  43.00000    NaN      NaN    5.500000
75%     207.75000  54.50000    NaN      NaN    6.750000
max     210.00000  67.00000    NaN      NaN    8.000000
```

2. Explore and Summarize Demographics

```
print("🔴 Age Distribution:")
print(df['age'].describe())
```

```
print("\n🔴 Gender Distribution:")
print(df['gender'].value_counts())
```

```
print("\n🔴 Admissions per Department:")
print(df['department'].value_counts())
```

```
🔄 🔴 Age Distribution:
count    10.000000
mean     44.300000
std      14.047934
min      23.000000
25%      35.000000
50%      43.000000
75%      54.500000
max      67.000000
Name: age, dtype: float64
```

```
🔴 Gender Distribution:
gender
Male      5
Female    5
Name: count, dtype: int64
```

```
🔴 Admissions per Department:
department
Cardiology    3
```

```
Neurology      3
Oncology       2
Pediatrics     2
Name: count, dtype: int64
```

3. Aggregate Key Statistics

```
dept_admissions = df.groupby('department').size().reset_index(name='admissions_count')
```

```
avg_stay = df['length_of_stay'].mean()
```

```
print("🔴 Department Admissions:")
print(dept_admissions)
```

```
print(f"\n🔴 Average Length of Stay: {avg_stay:.2f} days")
```

```
🔴 Department Admissions:
  department  admissions_count
0  Cardiology                3
1   Neurology                3
2    Oncology                2
3  Pediatrics                2

🔴 Average Length of Stay: 5.30 days
```

4. Data Cleaning

```
print("🔴 Missing Values:")
print(df.isnull().sum())
```

```
df['length_of_stay'].fillna(df['length_of_stay'].mean(), inplace=True)
```

```
df['gender'] = df['gender'].astype('category')
df['department'] = df['department'].astype('category')
```

```
print("\n🔴 Updated Data Types:")
print(df.dtypes)
```

```
🔴 Missing Values:
patient_id    0
age           0
gender        0
department    0
length_of_stay  0
dtype: int64

🔴 Updated Data Types:
patient_id    int64
age           int64
gender        category
department    category
length_of_stay  int64
dtype: object
<ipython-input-6-681093171>:10: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assign
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting
```

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col]

```
df['length_of_stay'].fillna(df['length_of_stay'].mean(), inplace=True)
```

5. Document Findings (Markdown)

```
from IPython.display import display, Markdown
```

```
display(Markdown("## 🏥 Hospital Data Summary"))
display(Markdown(f"- Total patients: **{len(df)}**"))
display(Markdown(f"- Unique departments: **{df['department'].nunique()}**"))
display(Markdown(f"- Average stay duration: **{df['length_of_stay'].mean():.2f}** days"))
display(Markdown("- Gender breakdown and age distribution analyzed.))
display(Markdown("- Missing values handled and data cleaned.))
```

```
🔴 Hospital Data Summary
```

- Total patients: **10**
- Unique departments: **4**
- Average stay duration: **5.30** days
- Gender breakdown and age distribution analyzed.
- Missing values handled and data cleaned.