🏥 Hospital Data Cleaning Project

📌 Project Title:

Hospital Patient Admission Dataset Cleaning

🎯 Objective:

To clean and prepare a large real-world hospital dataset (2.1 million+ rows and 33 columns) by handling missing values, removing inconsistencies, and standardizing formats to make the data analysis-ready. This enables better analysis of patient demographics, diagnoses, admission types, and healthcare system performance.

📊 Dataset Overview:

- **Initial Rows**: 2,101,588
- **Initial Columns**: 33
- **Final Columns**: 27
- **Source**: Real-world hospital dataset (via Kaggle)
- **Focus**: Hospitalizations, patient demographics, diagnoses, insurance types, outcomes

🧹 Cleaning Workflow Summary

1️⃣ Load & Preview the Data

- Loaded the dataset using `pandas`
- Previewed the data using:
  - `df.shape` to see dimensions

- `df.head()` to preview rows
- `df.info()` for column types and null counts
- `df.describe()` for numeric summary

2️⃣ Handle Missing Values

🔻 Dropped Columns (Too Many Nulls):

Column	Null % (approx.)	Action
Birth Weight	~90.1%	Dropped
CCSR Procedure Code	~27%	Dropped
CCSR Procedure Description	~27%	Dropped
Total Charges	~35%	Dropped
Total Costs	~35%	Dropped
Ratio of Costs to Charges	~35%	Dropped

✅ Imputed or Cleaned Columns:

Column	Action Taken
Hospital Service Area	Filled with `'Unknown'`
Permanent Facility Id	Filled with `'MODE VALUE'`
APR Risk of Mortality	Filled with `'Unknown'`
APR Severity of Illness Description	Filled with `'Unknown'`
CSR Diagnosis Description	Dropped ~0.07% of rows having nulls
CCSR Diagnosis Code	Dropped ~0.07% of rows having nulls

3️⃣ Standardize Column Names

- Converted all column names to lowercase
- Replaced spaces and hyphens with underscores

```
python
CopyEdit
df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_").str.replace("-", "_")
```

4️⃣ Remove Duplicates

- Checked using `df.duplicated().sum()`
- Removed any duplicate rows using `df.drop_duplicates(inplace=True)`

5️⃣ Data Type Fixes

- Ensured `discharge_year` was an integer
- Converted `zipcode3` to string type to preserve formatting
- Converted categorical columns to `category` data type for memory efficiency

---

6️⃣ Categorical Value Normalization

- Standardized categorical values (e.g., `"M"` → `"Male"`, `"U"` → `"Unknown"`)
- Used `.replace()` and `.str.lower()` to normalize casing and values

Example:

```
python
CopyEdit
df['gender'] = df['gender'].replace({'M': 'Male', 'F': 'Female', 'U': 'Unknown'})
```

7️⃣ Final Dataset Shape

- **Final Rows**: ~2,099,954
- **Final Columns**: 27

- **Exported As:** `hospital_cleaned.csv`

```
python
CopyEdit
df.to_csv("hospital_cleaned.csv", index=False)
```

✅ Results

- Cleaned and standardized a massive dataset with over 2.1 million rows
- Removed 6 columns with excessive nulls
- Imputed or cleaned remaining critical columns
- Result: An analysis-ready hospital dataset for insights and visualization

---

📘 Key Learnings

- Data cleaning is **crucial** before any analysis or modeling
- Learned how to **strategically drop or impute columns** based on null thresholds
- Gained experience working with **large-scale datasets** using efficient Pandas operations
- Practiced handling **categorical normalization** and **data type optimization**

📁 Project Assets

- `hospital_raw.csv` – Original dataset
- `hospital_cleaned.csv` – Final cleaned dataset
- `Hospital cleaning Project.ipynb` – Full Jupyter Notebook
- `README.md` – GitHub documentation file