HEALTHCARE PATIENT ADMISSION RECORDS DATA TRANSFORMATION

Pandas-based (no visualization) mini case study

SUBMITTED BY

BHAGYASHREE HOKRANI

4GW23CI005

bhagyashreehokrani2023@gmail.com

02/09/2025

INDEX

- PROJECT DESCRIPTION
- PROJECT OVERVIEW
- DETAILED EXPLANATION
- DATASET DESCRIPTION
- CODE
- FRONTEND INTERFACE DESIGN

PROJECT DESCRIPTION

The project focuses on transforming and cleaning patient admission records in a healthcare system. Using Pandas, you will perform data wrangling tasks like handling missing values, formatting dates, filtering based on conditions, grouping for analysis, and generating new derived columns. The goal is to prepare the dataset for insights such as patient stay duration, department utilization, and admission trends.

PROJECT OVERVIEW

This project ingests hospital patient admission records, cleans and standardizes the data with **Pandas**, derives useful fields (like patient stay

duration), filters and aggregates to produce operational insights (e.g., cardiology long-stays, admissions per department), and exports a clean CSV (transformed_admissions.csv). It's designed as a small ETL-style pipeline you can run standalone or call from a Flask endpoint after an upload

DEATAILED EXPLANATION

Step-by-step use-case explanations

Dataset Loading

```
df = pd.read_csv("patient_admissions.csv")
```

Reads the patient_admissions.csv file into a Pandas DataFrame.

This gives us the raw dataset to work with

Handling Missing Values

```
df["Diagnosis"] = df["Diagnosis"].fillna("Unknown")
df["DischargeDate"] = df["DischargeDate"].fillna(datetime.today().strftime("%d-%m-%Y"))
```

Missing values in **Diagnosis** → filled with "Unknown".

Missing **DischargeDate** → filled with today's date (simulating ongoing patients).

Data Type Conversion

```
df["AdmissionDate"] = pd.to_datetime(df["AdmissionDate"], dayfirst=True
errors="coerce")
df["DischargeDate"] = pd.to_datetime(df["DischargeDate"], dayfirst=True,
errors="coerce")
```

Converts admission and discharge dates into datetime objects.

This makes date arithmetic possible (e.g., calculating stay duration).

Derived Column – Stay Duration

```
df["StayDuration"] = (df["DischargeDate"] - df["AdmissionDate"]).dt.days
```

Creates a new column showing patient **length of stay** in days.

Essential for analysis like longest stays, averages, and trends.

Filter Specific Records

```
cardiology_patients = df[(df["Department"] == "Cardiology") & (df["StayDuration"] > 5)]
```

Filters **Cardiology patients** who stayed for more than 5 days.

Gives deeper insight into long-term admissions.

Grouping & Aggregation

```
admissions_per_dept = df.groupby("Department")["PatientID"].count().to_dict()
avg_stay_per_dept = df.groupby("Department")["StayDuration"].mean().round(2).to_dict()
```

Admissions per department → counts number of patients per department.

Average stay duration per department → calculates mean stay.

Sorting by Stay Duration

```
df = df.sort_values(by="StayDuration", ascending=False)
```

Sorts patients by longest hospital stay.

Helps easily spot patients with abnormal or prolonged admissions.

Removing Duplicates

```
df = df.drop_duplicates(subset=["PatientID", "AdmissionDate"], keep="first")
```

Ensures no duplicate entries exist for the same patient with the same admission date.

Saving Transformed Data

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

Saves the **cleaned & transformed dataset** into a new CSV file.

Ensures data is reusable for future analytics.

Flask Web Integration

```python
@app.route("/")
def index():
    df, cardiology_patients, admissions_per_dept, avg_stay_per_dept = transform_data()

    return render_template(
        "index.html",
        tables=df.to_html(classes="table table-bordered", index=False),
        cardiology=cardiology_patients.to_html(classes="table table-striped", index=False),
        admissions=admissions_per_dept,
        avg_stay=avg_stay_per_dept
    )
```

Runs the transformation pipeline each time / is accessed.

Passes the results to **index.html** for frontend display:

- Full transformed dataset (with stay duration, sorting, cleaning).
- Cardiology patients (stay > 5 days).
- Admissions per department.
- Average stay duration per department.

Dataset description

COLUMN NAME	DATA TYPE	DESCRIPTION
Patient ID	Integer/String	Unique identifier for each patient
Name	String	Patient full name
Age	integer	Patient age in years
Gender	String	Patient gender (Male/Female/Other)

Department	String	Cardiology, Neurology, orthopaedics
Diagnosis	String	Diagnosis description (can be missing)
Admission Date	Date	Date when patient was admitted (format: DD-MM-YYYY)
Discharge Date	Date	Date when patient was discharged (format: DD-MM-YYYY)
Stay Duration	Integer	Calculated: difference between discharge and admission dates in days

CODE

BACKEND CODE : app.py

```
from flask import Flask, render_template
import pandas as pd
from datetime import datetime

app = Flask(__name__)

def transform_data():
    # 1. Load Dataset
    df = pd.read_csv("patient_admissions.csv")

    # 2. Handle Missing Values
```

```
    df["Diagnosis"] = df["Diagnosis"].fillna("Unknown")
    df["DischargeDate"] = df["DischargeDate"].fillna(datetime.today().strftime("%d-%m-%Y"))

    # 3. Data Type Conversion
    df["AdmissionDate"] = pd.to_datetime(df["AdmissionDate"], dayfirst=True, errors="coerce")
    df["DischargeDate"] = pd.to_datetime(df["DischargeDate"], dayfirst=True, errors="coerce")

    # 4. Create Derived Column
    df["StayDuration"] = (df["DischargeDate"] - df["AdmissionDate"]).dt.days

    # 5. Filter Data
    cardiology_patients = df[(df["Department"] == "Cardiology") & (df["StayDuration"] > 5
   # 6. Group and Aggregate
    admissions_per_dept = df.groupby("Department")["PatientID"].count().to_dict()
    avg_stay_per_dept = df.groupby("Department")["StayDuration"].mean().round(2).to_dict()

    # 7. Sort Data
    df = df.sort_values(by="StayDuration", ascending=False)

    # 8. Remove Duplicates
    df = df.drop_duplicates(subset=["PatientID", "AdmissionDate"], keep="first")

    # 9. Save Transformed Data
    df.to_csv("transformed_admissions.csv", index=False)

    return df, cardiology_patients, admissions_per_dept, avg_stay_per_dept
```

```
@app.route("/")
def index():
    df, cardiology_patients, admissions_per_dept, avg_stay_per_dept = transform_data()

    return render_template(
        "index.html",
        tables=df.to_html(classes="table table-bordered", index=False),
        cardiology=cardiology_patients.to_html(classes="table table-striped", index=False),
        admissions=admissions_per_dept,
        avg_stay=avg_stay_per_dept
    )

if __name__ == "__main__":
    app.run(debug=True)
```

FRONTEND CODE : templates/index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Healthcare Admissions Dashboard</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="container mt-4">

    <h1 class="mb-4 text-center">🏥 Healthcare Admissions Dashboard</h1>

    <h3>📊 Full Transformed Dataset</h3>
    <div class="table-responsive">
```

```
    {{ tables | safe }}
  </div>

  <h3 class="mt-5">♡ Cardiology Patients (Stay > 5 Days)</h3>
  <div class="table-responsive">
    {{ cardiology | safe }}
  </div>

  <h3 class="mt-5">📌 Admissions per Department</h3>
  <ul>
    {% for dept, count in admissions.items() %}
      <li><b>{{ dept }}:</b> {{ count }}</li>
    {% endfor %}
  </ul>
  <h3 class="mt-5">⏱ Average Stay Duration per Department</h3>
  <ul>
    {% for dept, avg in avg_stay.items() %}
      <li><b>{{ dept }}:</b> {{ avg }} days</li>
    {% endfor %}
  </ul>

</body>
</html>
```

FRONTEND INTERFACE DESIGN

🏥 Healthcare Admissions Dashboard

📊 Full Transformed Dataset

PatientID	Name	Age	Gender	Department	Diagnosis	AdmissionDate	DischargeDate	StayDuration
P003	Michael Lee	29	M	Orthopedics	Fracture	2023-03-08	2025-09-03	910
P010	Anna Scott	40	F	Cardiology	Hypertension	2023-03-18	2025-09-03	900
P005	David Chen	61	M	Oncology	Cancer	2023-03-11	2023-03-25	14
P009	Daniel White	58	M	Oncology	Unknown	2023-03-16	2023-03-28	12
P001	John Doe	45	M	Cardiology	Heart Attack	2023-03-01	2023-03-10	9
P002	Jane Smith	34	F	Neurology	Stroke	2023-03-05	2023-03-12	7
P007	Robert Wilson	39	M	Cardiology	Arrhythmia	2023-03-14	2023-03-20	6
P004	Sarah Kim	52	F	Cardiology	Unknown	2023-03-10	2023-03-15	5
P008	Emily Davis	26	F	Orthopedics	Dislocation	2023-03-15	2023-03-18	3
P006	Linda Brown	47	F	Neurology	Epilepsy	2023-03-12	2023-03-14	2

💔 Cardiology Patients (Stay > 5 Days)

PatientID	Name	Age	Gender	Department	Diagnosis	AdmissionDate	DischargeDate	StayDuration
P001	John Doe	45	M	Cardiology	Heart Attack	2023-03-01	2023-03-10	9
P007	Robert Wilson	39	M	Cardiology	Arrhythmia	2023-03-14	2023-03-20	6
P010	Anna Scott	40	F	Cardiology	Hypertension	2023-03-18	2025-09-03	900

📌 Admissions per Department

- **Cardiology:** 4
- **Neurology:** 2
- **Oncology:** 2
- **Orthopedics:** 2

⏱️ Average Stay Duration per Department

- **Cardiology:** 230.0 days
- **Neurology:** 4.5 days
- **Oncology:** 13.0 days
- **Orthopedics:** 456.5 days