Project Title:

Data analysis on c - section delivery

Index

Problem Statement

- The rate of Caesarean (C-section) deliveries is increasing across India, raising concerns about maternal and neonatal health risks.
- Various factors—like healthcare access, water quality, and socioeconomic conditions—may influence delivery type but are poorly understood.
- This project aims to perform a data-driven analysis of delivery patterns using NFHS-5 data to identify key determinants and state-wise disparities

Project Description

- This project analyses Normal vs C-Section delivery trends across Indian states using NFHS-5 data (Box1.csv, Analysis_Code.xlsx).
- It uses attributes like:
    - Sector of healthcare (Public/Private)
    - State-wise C-section Delivery Rate (CDR)
    - Water quality index (WQ)
- The project structure follows clean Object-Oriented Python design, with separate modules for data loading, analysis, and visualization.
- Key analytical features:
    - Sector-wise average CDR bar chart
    - Correlation analysis between water quality and CDR
    - Top 5 states with highest C-section rates
- A CLI menu enables interactive exploration of insights through options 1 to 5.
- Visualization is done using matplotlib and pandas, ensuring clarity and presentation-quality output.
- The project highlights regional disparities and aims to assist public health stakeholders in improving delivery practices.
- Code is modularized and reusable, meeting software engineering standards for clarity and maintainability.

Objectives:

- To analyse the trends and distribution of Normal vs C-Section deliveries across Indian states.
- To identify sector-wise differences (Public vs Private) in C-section rates.
- To study the correlation between Water Quality (WQ) and C-section Delivery Rates (CDR).
- To determine state-wise variations and highlight the top 5 states with the highest CDR.
- To build a menu-driven CLI-based analytical tool for easy and interactive exploration.
- To implement the project using modular Python design, following OOP and clean coding practices.
- To visualize key patterns and insights using graphs and statistical summaries.
- To support data-driven policy-making by highlighting states/sectors needing intervention.
- To encourage use of public datasets like NFHS-5 for real-world healthcare analysis.
- To create a foundation for extending the analysis with more complex ML-based risk prediction in the future.

Dataset Description

◈ 1. Box1.csv

- Source: Derived/aggregated data from NFHS-5 datasets.
- Columns:
    - State: Name of the Indian state.
    - WQ: Water Quality Index category (Poorest, Poor, etc.)
    - Sector: Type of healthcare facility (Public, Private)
    - CDR: C-Section Delivery Rate (in percentage)

◈ 2. Analysis_Code.xlsx

- Content: Additional metadata, mappings, or supporting statistics used for analysis.
- Usage:
    - Contains structured sheets helpful in cross-validation and extending analysis.
    - Serves as a secondary source of reference for interpreting delivery data.

Key Analyses Performed:

- **1. Sector-wise Average C-Section Rate**
    - Compared the average C-Section Delivery Rate (CDR) between Public and Private healthcare sectors.
    - Observed significantly higher CDRs in the Private sector, indicating potential overuse or preference.
- **2. Water Quality vs C-Section Correlation**
    - Analyzed how Water Quality (WQ) categories (e.g., *Poorest*, *Poor*, etc.) influence C-section delivery rates.
    - Identified whether poor environmental conditions contribute to higher medical intervention rates.
- **3. State-wise CDR Analysis**
    - Ranked Indian states by their average C-section rates.
    - Highlighted the Top 5 states with the highest CDRs for targeted policy intervention.
- **4. Visual Exploration via Bar Charts**
    - Presented sector-wise and state-wise comparisons through bar graphs for better visual interpretation.
    - Used color-coded plots to show CDR variation by sector and water quality.
- **5. CLI-Based Interactive Dashboard**
    - Developed a menu-driven interface allowing users to:
        - Trigger any analysis on-demand
        - View summary statistics
        - Explore graphs instantly
        - Repeat analyses without restarting the program

Outcomes and Benefits

- **Deeper Understanding of C-Section Patterns**
    - Gained valuable insights into how delivery modes vary across sectors, states, and environmental conditions like water quality.
- **Data-Driven Policy Support**
    - Helps health administrators and policymakers identify regions or sectors with abnormally high C-section rates for further investigation or intervention.
- **Region-Specific Focus**
    - Enables state-level comparisons, spotlighting areas needing better maternal care, awareness, or infrastructure upgrades.
- **Sector-Specific Observations**
    - Revealed a trend of higher C-section rates in private facilities, suggesting a need to examine medical ethics and patient education.
- **Visual and Interactive Reporting**
    - Offers a clear, user-friendly CLI dashboard for non-technical users to interact with and understand the data through visuals and summaries.
- **Reusability and Modularity**
    - Modular code structure using OOP principles makes the system easy to extend for future datasets or additional analyses.
- **Educational Value**
    - Serves as a practical case study for data analysis, public health research, and clean software engineering practices using Python.

Data Sources

- NFHS-5 (2019–21)
  Source: Ministry of Health and Family Welfare, Govt. of India
  🔗 http://rchiips.org/NFHS/NFHS-5_FCTS
  → Used for delivery type trends, sector-wise stats, and state comparisons.
- Box1.csv
  → Custom CSV extracted from NFHS-5.
  → Used for C-section rate analysis, water quality comparison, and top-performing states.
- Analysis_Code.xlsx
  → Internal spreadsheet for mapping and additional insights.

Tools & Technologies Used:

- **Python 3.11** – Core programming language for data handling and analysis
- **Pandas** – For data cleaning, filtering, and statistical operations
- **Matplotlib & Seaborn** – For visualizations and correlation plots
- **VS Code** – Development and debugging environment
- **Excel** – Initial data inspection and mapping (via Analysis_Code.xlsx)
- **CLI (Command-Line Interface)** – For user-friendly navigation through the analysis

Code Module Structure Diagram:

Hackathon_project/

│

├── main.py # CLI-based interface and entry point

├── modules/

│ ├── analyzer.py # Contains core data analysis logic

│ └── visualizer.py # Contains plotting and visualization functions

│

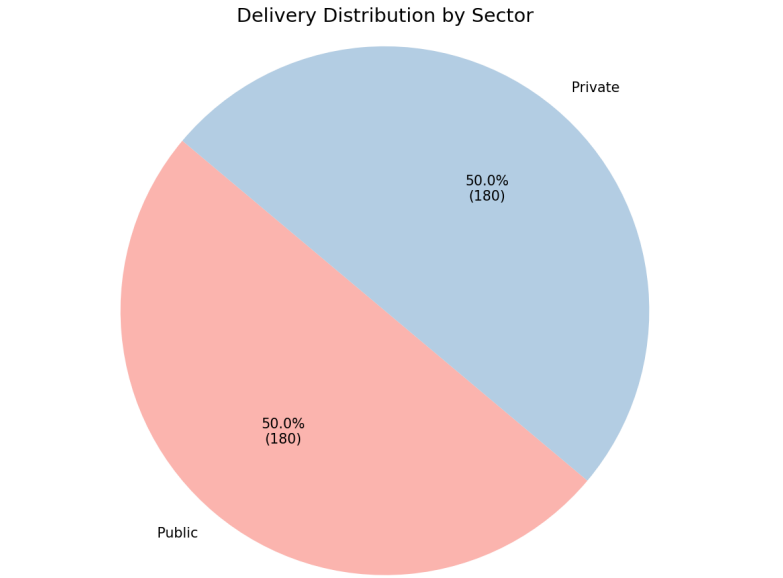
├── data/

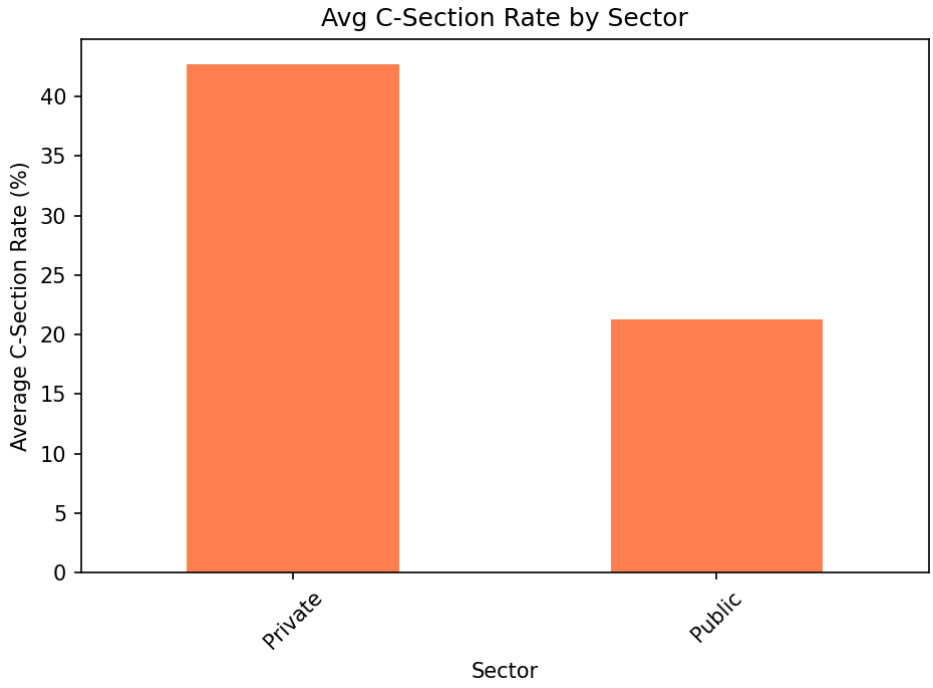
│ ├── Box1.csv # Cleaned dataset used for analysis

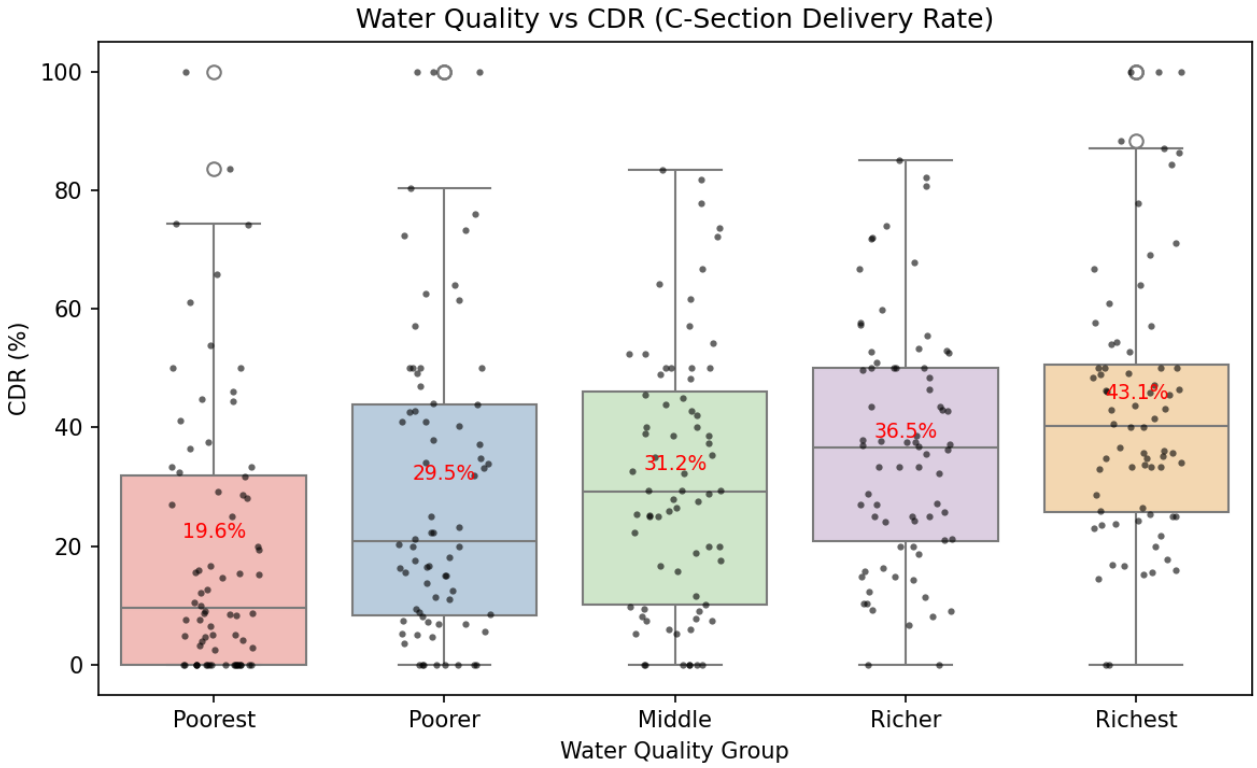
│ └── Analysis_Code.xlsx # Reference codebook for column

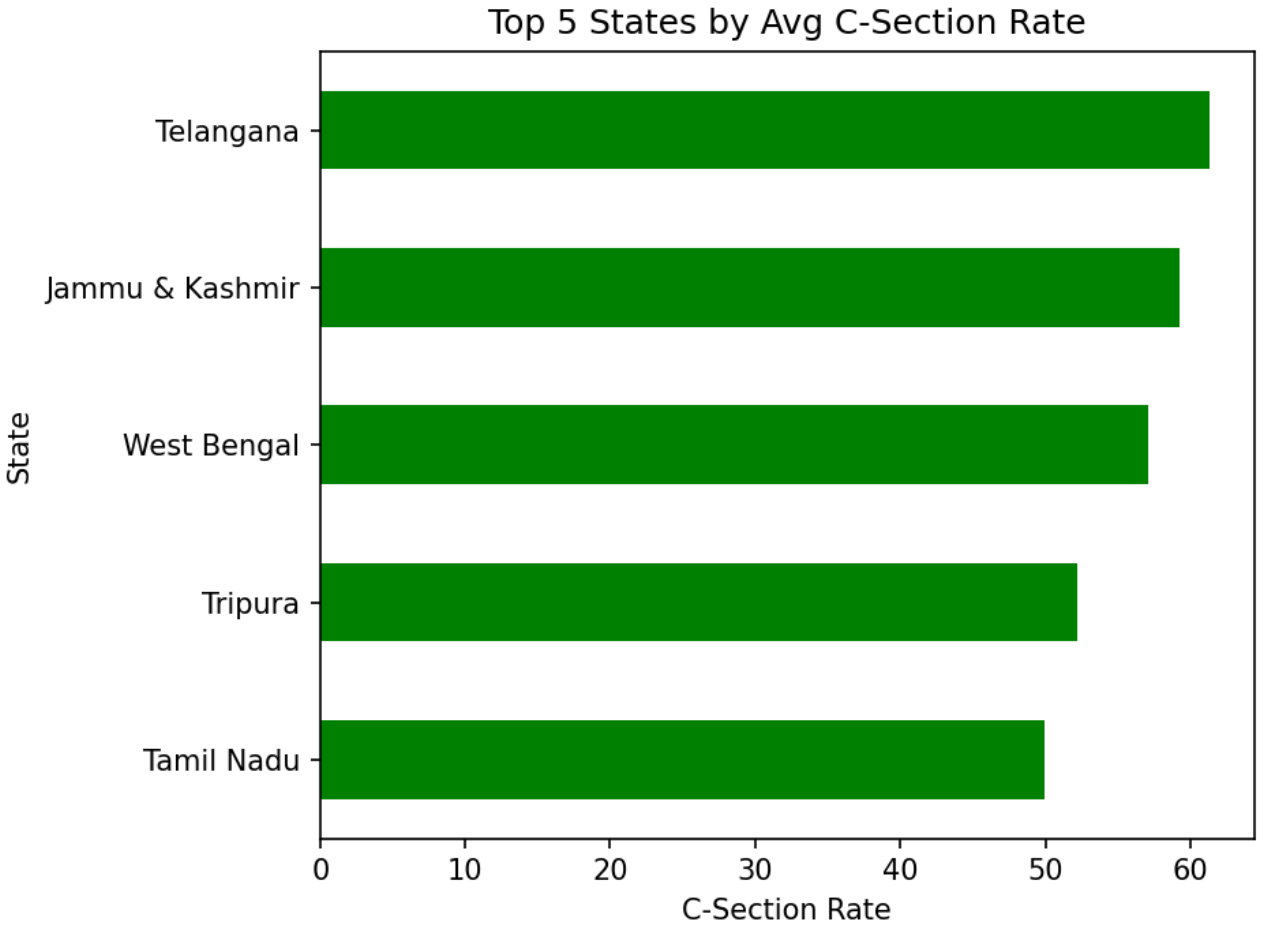
Flow Summary:

- **main.py handles user interaction via CLI.**
- **analyzer.py processes and analyzes the data from Box1.csv.**
- **visualizer.py creates charts and graphs from processed data.**
- **External libraries like pandas, numpy, matplotlib, and seaborn support core logic and visualization.**


Delivery Distribution by Sector
Private
50.0%
(180)
50.0%
(180)
Public


Avg C-Section Rate by Sector
Average C-Section Rate (%)
0
5
10
15
20
25
30
35
40
Private
Public
Sector


Water Quality vs CDR (C-Section Delivery Rate)
CDR (%)
100
80
60
40
20
0
19.6%
29.5%
31.2%
36.5%
43.1%
Poorest
Poorer
Middle
Richer
Richest
Water Quality Group


Top 5 States by Avg C-Section Rate
State
Telangana
Jammu & Kashmir
West Bengal
Tripura
Tamil Nadu
0
10
20
30
40
50
60
C-Section Rate

```python
from modules.analyzer import DeliveryDataLoader
from modules.visualizer import (
    plot_sector_distribution_pie,
    plot_avg_cdr_by_sector,
    plot_water_quality_vs_cdr,
    plot_top_states_by_cdr
)
def main():
    """
    CLI interface to choose and run visual analysis.
    """
    # To make sure these paths are correct
    loader = DeliveryDataLoader(
        'data/Box1.csv',
        'data/Analysis_Code.xlsx'  # Optional
    )

    df = loader.raw_data
    while True:
        print("\nSelect an Analysis:")
        print("1. Delivery Distribution by Sector")
        print("2. Average C-Section Rate by Sector")
        print("3. Water Quality vs C-section delivery rate")
        print("4. Top 5 States by  C-section delivery rate")
        print("5. Exit")

        choice = input("Enter your choice: ").strip()

        if choice == '1':
            plot_sector_distribution_pie(df)
        elif choice == '2':
            plot_avg_cdr_by_sector(df)
        elif choice == '3':
            plot_water_quality_vs_cdr(df)
        elif choice == '4':
            plot_top_states_by_cdr(df, top_n=5)
        elif choice == '5':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Try again.")

if __name__ == "__main__":
    main()
```

```python
# modules/analyzer.py

import pandas as pd

class DeliveryDataLoader:
    def __init__(self, box1_path: str, excel_path: str = None):

        self.box1_path = box1_path
        self.excel_path = excel_path
        self.raw_data = pd.read_csv(self.box1_path)
        self.raw_data.columns = [col.strip() for col in self.raw_data.columns]  # Clean headers
        print(f"[INFO] Loaded {len(self.raw_data)} records from {self.box1_path}")

        # Load additional Excel data if provided
        if self.excel_path:
            self._load_excel_data()

        # Add BMI categories if BMI column exists
        if 'BMI' in self.raw_data.columns:
            self._categorize_bmi()

    def _load_excel_data(self):
        """
        Loads sheets from Excel file if path is provided.
        """
        try:
            excel_sheets = pd.read_excel(self.excel_path, sheet_name=None)
            self.statewise_urban_rural = excel_sheets.get('Statewise_Urban_Rural', pd.DataFrame())
            self.csection_factors = excel_sheets.get('CSection_Factors', pd.DataFrame())
            print("[INFO] Excel data loaded successfully.")
        except Exception as e:
            print(f"[ERROR] Failed to load Excel file: {e}")

    def _categorize_bmi(self):
        """
        Categorizes BMI values into standard groups.
        """
        self.raw_data['BMI_Category'] = pd.cut(
            self.raw_data['BMI'],
            bins=[0, 18.5, 24.9, 29.9, 100],
            labels=["Underweight", "Normal", "Overweight", "Obese"]
        )
```

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def plot_sector_distribution_pie(df):

    if df.empty or 'Sector' not in df.columns:
        print("Missing 'Sector' column in data.")
        return

    sector_counts = df['Sector'].value_counts()
    sector_counts = sector_counts.sort_values(ascending=False)


    plt.figure(figsize=(7, 7))
    wedges, texts, autotexts = plt.pie(
        sector_counts,
        labels=sector_counts.index,
        autopct=lambda pct: f'{pct:.1f}%\n({int(pct * sector_counts.sum() / 100)})',
        startangle=140,
        colors=plt.cm.Pastel1.colors,
        textprops={'fontsize': 10}
    )

    plt.title('Delivery Distribution by Sector', fontsize=14)
    plt.axis('equal')  # Make it a circle
    plt.tight_layout()
    plt.show()

def plot_avg_cdr_by_sector(df):
    if df.empty or 'CDR' not in df.columns or 'Sector' not in df.columns:
        print("Missing required columns in data.")
        return

    df['CDR'] = pd.to_numeric(df['CDR'], errors='coerce')
    df_clean = df.dropna(subset=['CDR'])

    avg_cdr = df_clean.groupby('Sector')['CDR'].mean().round(2).sort_values(ascending=False)
    avg_cdr.plot(kind='bar', color='coral', title='Avg C-Section Rate by Sector')
    plt.ylabel('Average C-Section Rate (%)')
    plt.xlabel('Sector')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

```
def plot_water_quality_vs_cdr(df):

    if 'WQ' not in df.columns or 'CDR' not in df.columns:
        print("Missing 'WQ' or 'CDR' column in data.")
        return
    df['CDR'] = pd.to_numeric(df['CDR'], errors='coerce')
    df = df.dropna(subset=['WQ', 'CDR'])
    wq_order = ['Poorest', 'Poorer', 'Middle', 'Richer', 'Richest']
    df['WQ'] = pd.Categorical(df['WQ'], categories=wq_order, ordered=True)

    plt.figure(figsize=(8, 5))
    sns.boxplot(x='WQ', y='CDR', data=df, order=wq_order, palette='Pastel1', hue='WQ', legend=False)
    sns.stripplot(x='WQ', y='CDR', data=df, order=wq_order, color='black', size=3, jitter=0.2, alpha=0.6)
    mean_values = df.groupby('WQ', observed=True)['CDR'].mean()
    for i, wq in enumerate(wq_order):
        if wq in mean_values:
            plt.text(i, mean_values[wq] + 1, f'{mean_values[wq]:.1f}%',
                     ha='center', va='bottom', fontsize=9, color='red')

    plt.title('Water Quality vs CDR (C-Section Delivery Rate)')
    plt.xlabel('Water Quality Group')
    plt.ylabel('CDR (%)')
    plt.tight_layout()
    plt.show()

def plot_top_states_by_cdr(df, top_n=5):
    """
    Plots top N states by average C-Section Rate.

    Args:
        df (DataFrame): Dataset with 'State' and 'CDR'
        top_n (int): Number of top states to show
    """
    if df.empty or 'State' not in df.columns or 'CDR' not in df.columns:
        print("Missing required columns in data.")
        return

    df['CDR'] = pd.to_numeric(df['CDR'], errors='coerce')
    df_clean = df.dropna(subset=['CDR'])

    top_states = df_clean.groupby('State')['CDR'].mean().round(2).sort_values(ascending=False).head(top_n)
    top_states.plot(kind='barh', color='green', title=f'Top {top_n} States by Avg C-Section Rate')
    plt.xlabel('C-Section Rate')
    plt.ylabel('State')
    plt.gca().invert_yaxis()
    plt.tight_layout()
    plt.show()
```

Project Summary

This project aimed to analyze and compare Normal vs C-Section delivery patterns in India using health statistics from NFHS-5. Through

careful data preprocessing, visualization, and state-wise analysis, the project revealed insightful trends related to maternal health, healthcare sectors, and delivery practices.

Key aspects included:

- Investigating the impact of sector (public/private) and water quality on C-section rates.
- Identifying states with the highest C-section occurrences.
- Presenting data through interactive CLI menus and visual charts for better understanding.
- Using real datasets from reliable sources to ensure authenticity and relevance.

Ultimately, the project empowers health professionals, policymakers, and the general public to make informed decisions based on trends in maternal and child healthcare.