Indian Crime Data Visualization Portal: An Interactive Exploration Platform

Group Number - 5
Akshay Toshniwal - 241110005
Ansh Makwe - 241110010
Devang Agarwal - 241110019
Divyansh Chaurasia - 241110022
Kunal Anand - 241110040
Parjanya Aditya Shukla - 241110046
Prakhar Mandloi - 241110051
Roshan Kumar - 241110058

The GitHub Repo Link for the project is - Watch-Tower

April 25, 2025

Contents

Chapter 1

Executive Summary

This report details the "Indian Crime Data Visualization Portal," an interactive web application developed using Python, Dash, and Plotly. The portal provides a comprehensive platform for exploring diverse datasets related to crime in India, sourced primarily from kaggle - datacard , spanning various years (largely 2001-2013, with some variations per dataset). The application integrates multiple datasets covering Indian Penal Code (IPC) crimes, crimes against specific groups (SC, ST, Women, Children), juvenile justice statistics, property crimes, custodial deaths, and more. Through a multi-tabbed interface, users can analyze crime data geographically (state and district levels), temporally (year-wise trends), comparatively (across areas or crime types), and thematically (e.g., victim-offender relationships, place of occurrence, kidnapping motives). Key features include interactive maps, dynamic charts (bar, line, pie, sunburst, Sankey, treemap, heatmap), data filtering controls, and integrated machine learning components for clustering (K-Means) and forecasting (ARIMA). This report outlines the project's objectives, data handling procedures, application architecture, and provides a detailed walkthrough of each functional tab, explaining its purpose and showcasing its capabilities.

The GitHub Repo Link for the project is - Watch-Tower.

Chapter 2

Introduction

2.1 Project Overview and Goals

Crime is a significant societal issue with far-reaching consequences. Understanding its patterns, trends, and underlying factors is crucial for effective prevention, law enforcement, and policy-making. India, with its vast population and diverse socio-economic landscape, presents a complex picture of crime. While government agencies collect extensive data, accessing, integrating, and visualizing this information in a meaningful way can be challenging.

The "Indian Crime Data Visualization Portal" project was undertaken to address this challenge. The primary goal is to develop a user-friendly, interactive web application. The portal aims to:

- Provide a centralized platform for accessing diverse crime statistics.
- Enable exploration of crime data across different dimensions: geographic (state/district), temporal (yearly trends), demographic (victim/offender characteristics), and situational (place of occurrence, motive).
- Facilitate comparative analysis between regions, time periods, and crime types.
- Employ data visualization techniques to reveal patterns, hotspots, and correlations that might be obscured in raw data tables.
- Incorporate basic machine learning techniques (clustering, forecasting) to uncover deeper insights and potential future trends.

2.2 Scope

The scope of this project encompasses:

- **Data Integration:** Loading, cleaning, and standardizing multiple crime-related datasets primarily from the 2001-2013 period, covering various crime categories (IPC, SC/ST, Women, Children, etc.) and related statistics (juvenile arrests, property theft, custodial deaths, fraud, unidentified bodies, victim/offender details).
- **Visualization Development:** Creating a range of interactive visualizations including choropleth maps (state and district), line charts, bar charts, pie charts, radar charts, sunburst diagrams, Sankey diagrams, treemaps, and heatmaps.
- **Web Application Interface:** Building a multi-tabbed Dash web application to host the visualizations and provide user controls for filtering and interaction.
- **Interactivity:** Implementing callbacks to allow users to dynamically filter data by category, year, specific crime type, state, district, etc., and update visualizations accordingly.
- **Machine Learning Features:** Integrating K-Means clustering for district crime profiles, ARIMA forecasting for cluster trends.

2.3 Technologies Used

The portal is built entirely using the Python ecosystem:

- **Dash:** A framework by Plotly for building analytical web applications with Python. Used for the overall structure, UI components, and interactivity.
- **Plotly (Express and Graph Objects):** Libraries for creating interactive, publication-quality visualizations (maps, charts, etc.). Plotly Express is used for rapid chart generation, while Graph Objects offer finer control.
- **Pandas:** The cornerstone library for data manipulation, cleaning, aggregation, and analysis. Used extensively for preparing data for visualization.
- **Numpy:** Fundamental package for numerical computation in Python. Used for array operations and calculations.
- **Scikit-learn:** Machine learning library used for:
  - `StandardScaler`: Scaling features before clustering.
  - `KMeans`: Implementing K-Means clustering algorithm.
  - `silhouette_score`, `silhouette_samples`: Evaluating cluster quality.
- **Statsmodels:** Library for statistical models, used here for `ARIMA` time series forecasting.
- **Rapidfuzz:** Library for fuzzy string matching, used to reconcile differences between GeoJSON names and CSV data names.
- **JSON:** Standard library for handling JSON data, used for loading GeoJSON files.
- **OS:** Used for environment variable settings (e.g., `LOKY_MAX_CPU_COUNT`).

Chapter 3

Data Sources and Preprocessing

A robust visualization portal relies heavily on well-prepared data. The dataset was sourced from kaggle whose source has already been provided above. Significant effort, was dedicated to loading, cleaning, standardizing, and structuring this data for analysis and visualization.

3.1 Overview of Datasets

The application utilizes a wide array of CSV files, categorized broadly as:

1. **Core Crime Statistics (District-wise):**
   - IPC Crimes (`01_District_wise_crimes_committed_IPC_final.csv`)
   - Crimes against Scheduled Castes (SC) (`02_01_District_wise_crimes_committed_against_SC_final.csv`)
   - Crimes against Scheduled Tribes (ST) (`02_District_wise_crimes_committed_against_ST_final.csv`)
   - Crimes against Children (`03_District_wise_crimes_committed_against_children_final.csv`)
   - Crimes against Women (`42_District_wise_crimes_committed_against_women_2001_2013.csv`)
2. **Juvenile Justice Statistics:**
   - Education Level (`18_01_Juveniles_arrested_Education.csv`)
   - Economic Setup (`18_02_Juveniles_arrested_Economic_setup.csv`)
   - Family Background (`18_03_Juveniles_arrested_Family_background.csv`)
   - Recidivism (`18_04_Juveniles_arrested_Recidivism.csv`)
3. **Property Crime:**
   - Property Stolen and Recovered (`property_stolen.csv`)
4. **Specific Crime Types / Circumstances:**
   - Unidentified Dead Bodies (`38_Unidentified_dead_bodies_recovered_and_inquest_conducted.csv`)
   - Serious Fraud (`31_Serious_fraud.csv`)
   - Custodial Deaths (`40_05_Custodial_death_others.csv`)
   - Murder Victim Age/Sex (`32_Murder_victim_age_sex.csv`)
   - Firearm Use in Murder (`34_Use_of_fire_arms_in_murder_cases.csv`)
   - Place of Occurrence (`17_Crime_by_place_of_occurrence_2001_2012.csv`)

- Offender-Victim Relationship (21_Offenders_known_to_the_victim.csv)
- Rape Victims by Subgroup (20_Victims_of_rape.csv)
- Kidnapping Purpose (39_Specific_purpose_of_kidnapping_and_abduction.csv)

5. **Geographic Data:**
    - India State Boundaries (india_state.geojson)
    - India District Boundaries (india_district.geojson)

3.2 Data Loading and Initial Cleaning (load_and_clean_csv, load_and_clean_csv1)

Two primary functions handle the loading of CSV data: load_and_clean_csv and load_and_clean_csv1.

- load_and_clean_csv: Designed for the district-wise core crime datasets (IPC, SC, ST, Children, Women). It reads the CSV, standardizes columns (see below), removes rows where the DISTRICT is "TOTAL" (to avoid double-counting aggregates), calculates a TOTAL_CRIMES column specific to the category, and ensures the YEAR column is treated as a string.
- load_and_clean_csv1: A simpler version used for other datasets (like Juvenile stats) that might have slightly different structures (e.g., AREA_NAME instead of STATE/DISTRICT, no district-level totals to remove). It performs column standardization and basic type conversion.

Specific datasets like property_stolen.csv, 38_Unidentified_dead_bodies..., 31_Serious_fraud.csv, 40_05_Custodial_death_others.csv, etc., undergo tailored loading and cleaning steps within the main script body, often involving renaming columns, handling 'NULL' values, melting data into a long format for easier plotting, and ensuring correct data types. For instance, custodial death data replaces "NULL" with NaN and melts multiple cause columns into a 'Cause' and 'Count' format. Fraud data is melted to handle different loss bands.

3.3 Column Standardization (standardize_columns, standardize_columns1)

Consistency across datasets is crucial for merging and analysis. The standardize_columns and standardize_columns1 functions play a vital role here. Their key actions include:

- **Case Conversion:** Converting all column names to uppercase.
- **Whitespace Removal:** Stripping leading/trailing whitespace from column names.
- **Space Replacement:** Replacing spaces in column names with underscores (e.g., "Area Name" becomes AREA_NAME).
- **Standard Naming:** Renaming common variations like "STATE/UT" or Area_Name to a consistent STATE. Specific variations like KIDNAPPING_ABDUCTION or KIDNAPPING_&_ABDUCTION are standardized to KIDNAPPING_AND_ABDUCTION. "Year" is standardized to YEAR.
- **Numeric Conversion:** Attempting to convert columns likely to be numeric (excluding identifiers like STATE, DISTRICT, YEAR, and known text categories like PURPOSE, RELATIONSHIP etc.) to numeric types using pd.to_numeric(errors='coerce'). This gracefully handles non-numeric entries by converting them to NaN (Not a Number).
- **Identifier Typing:** Ensuring key identifiers like YEAR are consistently treated as strings (or integers where appropriate for sliders/calculations like in the Kidnapping or Rape tabs) after other operations.

This standardization ensures that columns representing the same information have the same name and comparable data types across different input files, simplifying subsequent data merging and plotting operations.

3.4 Total Crime Calculation (calculate_total_crimes)

For the core district-wise datasets (IPC, SC, ST, Children, Women), understanding the overall crime burden is important. The calculate_total_crimes function generates a consistent TOTAL_CRIMES column for each category:

- **IPC:** Renames the existing TOTAL_IPC_CRIMES column if present; otherwise, sums the individual IPC crime columns defined in crime_options['ipc'].
- **SC/ST/Women:** Sums the specific crime columns listed for that category in the crime_options dictionary. It handles potential missing columns gracefully.
- **Children:** Renames an existing TOTAL column if present, otherwise sums the relevant columns from crime_options['children'].

This function ensures each primary dataset has a comparable total crime metric used in various visualizations, particularly the initial state and district maps.

3.5 GeoJSON Handling and Name Normalization (normalize_geojson, fix_geojson_names)

Choropleth maps require geographic boundary data (GeoJSON) and a way to link it to the crime statistics.

- **Loading and Normalization (normalize_geojson):** The india_state.geojson and india_district.geojson files are loaded. The normalize_geojson function standardizes the relevant property key within the GeoJSON (e.g., "NAME_1" for states, "NAME_2" for districts) to a consistent uppercase key (STATE_UPPER, DISTRICT_UPPER) for easier matching with the cleaned CSV data.
- **Fuzzy Matching (fix_geojson_names, best_match_fuzzy):** Geographic names in GeoJSON files often have slight variations (spelling, abbreviations) compared to administrative names in CSV data. To bridge this gap, the fix_geojson_names function iterates through the GeoJSON features. If a name (e.g., DISTRICT_UPPER) doesn't exactly match any district name found in the loaded CSV dataframes, it uses the rapidfuzz library's process.extractOne with the fuzz.WRatio scorer to find the best fuzzy match above a certain threshold (75) within the list of known CSV district names. If a good match is found, the GeoJSON property is updated to match the CSV name, significantly improving the linkage between geographic boundaries and statistical data for map plotting. This step is crucial for ensuring districts appear correctly on the choropleth maps. Diagnostics are printed to show the number of mismatches and successful updates.

3.6 Specific Dataset Notes

Beyond the general cleaning, several datasets required specific handling:

- **Kidnapping Purpose (load_and_clean_kidnapping_purpose_csv):** This function handles loading the kidnapping data, standardizing columns, renaming key fields (AREA_NAME to STATE, SUB_GROUP_NAME to PURPOSE, K_A_GRAND_TOTAL to COUNT), converting types, and crucially, cleaning the PURPOSE strings (e.g., removing leading numbers like "01. ") to create a clean PURPOSE_CLEAN column for user-friendly dropdown labels. It explicitly handles 'NULL' values during loading.
- **Murder Victims Flow (df_murder):** While basic loading occurs, the Sankey diagram callback performs significant reshaping. It aggregates different age columns (e.g., Victims_Above_50_Yrs, Victims_Upto_10_15_18_30_50_Yrs) into broader categories like 'Adult' and 'Non-Adult' (though

the exact logic might need refinement based on column names) and filters for specific gender subgroups ('Male Victims', 'Female Victims') to structure the data for the Sankey flow visualization.

- **Offender Relationships (`df_relative`, `df_rel_long`)**: The data is melted from a wide format (with columns like `No_of_Cases_in_which_offenders_were_Relatives`) into a long format (`df_rel_long`) with 'Relationship' and 'Count' columns. The relationship labels are cleaned by removing prefixes and underscores.
- **Place Occurrence (`df_place`, `df_place_long`)**: Similar to the relationship data, the numerous columns representing `PLACE - CRIME` combinations are melted into a long format (`df_place_long`) with `PLACE_CRIME` and `COUNT` columns. The `PLACE_CRIME` column is then split into separate `PLACE` and `CRIME` columns for use in the sunburst and small multiple visualizations.
- **Firearm Use (`df_raw`, `df_aggregated`)**: This data focuses on comparing victims based on whether registered or unregistered arms were used. The code aggregates data across all years by `Area_Name` (standardized to `STATE`) and sums the relevant numeric columns (`Victims`, `By Registered Arms`, `By Unregistered Arms` - after standardization) into `df_aggregated` for use in the Area Comparison radar chart.
- **Custodial Deaths (`df_cust`, `melted`)**: Handles 'NULL' values, converts count columns to numeric, and melts the data so each row represents a specific `Area_Name`, `Year`, `Cause` (e.g., `CD_Accidents`), and `Count`. A national aggregate (`nat_melt`) is also created.

This extensive preprocessing ensures that the diverse datasets are harmonized and structured appropriately for the various visualizations and analyses presented in the portal.

Chapter 4

Application Architecture

The Indian Crime Data Visualization Portal is developed as a web application using the Dash framework, leveraging several key components for its structure and interactivity.

4.1 Framework: Dash by Plotly

Dash is the core framework upon which the application is built. Dash is ideal for building data-driven applications purely in Python, eliminating the need for frontend web development languages like JavaScript for basic interactions. Key advantages relevant to this project include:

- **Python Native:** Allows developers to use familiar Python libraries (Pandas, Scikit-learn, etc.) directly within the application logic.
- **Component-Based UI:** Dash applications are built using components (like graphs, dropdowns, sliders, tabs) arranged in a layout.
- **Reactive Callbacks:** The interactivity is powered by callbacks – Python functions that are triggered by changes in UI component properties (e.g., selecting a value in a dropdown) and update other component properties (e.g., changing the data in a graph).
- **Plotly Integration:** Seamless integration with Plotly.py for creating rich, interactive visualizations.
- **Web Deployment:** Dash applications are web servers that can be deployed easily. The code includes `server = app.server` which is standard practice for deployment platforms like Heroku or Render.

The application initializes a Dash app instance. The `suppress_callback_exceptions=True` argument is necessary because components (like sliders and dropdowns within tabs) are not present in the initial layout but are generated dynamically by other callbacks; this setting prevents Dash from raising errors about missing components during startup.

4.2 Core Components: Layout and Callbacks

The application's structure and behavior are defined by two main parts:

- **Layout (`app.layout`):** This defines the visual structure of the application using Dash HTML Components (`html.Div`, `html.H1`, `html.P`, etc.) and Dash Core Components (`dcc.Tabs`, `dcc.Tab`, `dcc.Graph`, `dcc.Dropdown`, `dcc.Slider`, `dcc.RadioItems`, `dcc.Store`, `dcc.Loading`, etc.).
  - The main structure is a `html.Div` containing a title (`html.H1`) and a `dcc.Tabs` component (`id="main-tabs"`).
  - Each `dcc.Tab` within `main-tabs` represents a distinct section/page of the application (e.g., State Wise, District Wise, Year Wise, etc.).

  - Inside each tab, HTML and DCC components are nested to create sections for controls (dropdowns, sliders, radio buttons) and visualizations (`dcc.Graph`).
  - `dcc.Loading` components wrap around `dcc.Graph` elements to provide visual feedback to the user while data is being processed and plots are being generated.
  - `dcc.Store` components (`id="selected-state-store"`, `id="selected-district-store"`) are used to hold intermediate data or user selections (like the clicked state/district) that need to be shared between callbacks without being directly visible in the UI.

- **Callbacks (`@app.callback`)**: These Python functions define the application's dynamic behavior. They link inputs (changes in component properties like the `value` of a dropdown) to outputs (updates to other component properties like the `figure` of a graph or the `options` of another dropdown).

  - **Structure:** Each callback is defined by the `@app.callback` decorator, which specifies the `Output` component(s) and property(ies) to be updated, and the `Input` and optional `State` component(s) and property(ies) that trigger the function and provide data to it.
  - **Functionality:** Callbacks handle tasks such as:
    - Updating dropdown options based on selections in other controls (e.g., updating the crime dropdown based on the selected category).
    - Filtering Pandas DataFrames based on user inputs (years, states, districts, crime types).
    - Performing aggregations and calculations on the filtered data.
    - Generating or updating Plotly figures based on the processed data.
    - Running machine learning models (KMeans, ARIMA, LinearRegression) and processing their results for visualization.
    - Updating text displays or storing intermediate selections.
  - **Dynamic Updates:** The extensive use of callbacks makes the application highly interactive, allowing users to explore the data dynamically without page reloads. For example, clicking a state on the map triggers a callback to show state-specific analysis options, and changing the year slider in that section triggers another callback to update the state-specific graphs.

4.3 Styling and UI Elements

While the core logic is Python, the application incorporates basic styling for a professional look and feel:

- **CSS Styling:** Inline style dictionaries are used extensively within the `html` and `dcc` components to control layout (width, display, padding, margin), colors, borders, and fonts. Consistent styles are defined for elements like dropdowns (`dropdown_style`), radio items (`radio_style`, `radio_label_style`), and plot containers (`card_style`) to maintain visual uniformity.

- **Bootstrap Inspiration:** Although not explicitly importing Bootstrap, the use of card-like containers with shadows (`card_style`) and the arrangement of controls suggest inspiration from modern web design frameworks.

- **Plotly Visual Theming:** Plotly figures themselves are styled within the callbacks, setting background colors (`plot_bgcolor`, `paper_bgcolor`), gridline colors, font properties, and color scales to ensure clarity and visual appeal. Specific color palettes (`px.colors.qualitative.Plotly`, `CLUSTER_COLORS`) are used for consistency in categorical data representation.

The combination of Dash's component system, reactive callbacks, and Plotly's visualization capabilities, along with careful data preprocessing, forms the architectural foundation of this interactive crime data portal.

Chapter 5

Overall Application Story: Unveiling Crime Patterns in India

Imagine trying to understand the complex issue of crime across a nation as vast and diverse as India. Where are different types of crimes most prevalent? How have patterns changed over time? Are certain demographic groups more affected? Are there hidden similarities between crime profiles in different districts? Answering these questions requires sifting through massive amounts of data, often fragmented across different reports and formats. Raw numbers in tables can be overwhelming and may obscure critical trends or geographical hotspots.

This **Indian Crime Data Visualization Portal** was created to cut through that complexity. It serves as a unified, interactive lens, bringing together diverse crime statistics from the early 2000s into a single, explorable platform. The story of this application is one of **empowerment through visualization**.

Instead of static reports, the portal offers a dynamic journey. Users can start with a broad overview, visualizing total crime counts across states on an interactive map. Noticing a high crime rate in a particular state? They can instantly drill down to the district level, examining the distribution of specific crime types within that state for selected years. Curious about how a particular crime, like robbery or crimes against women, has evolved nationally or within a specific state over the years? The "Year Wise" and state/district detail sections provide trend analyses.

The portal goes beyond simple counts. It allows users to investigate specific facets of crime: Where do certain crimes typically occur? What is the relationship between victims and known offenders? What are the primary motives behind kidnappings? How do demographics like age and gender factor into murder victimization? It even delves into systemic issues like custodial deaths and the backgrounds of juveniles involved in the justice system.

Furthermore, the application incorporates analytical tools. It uses clustering techniques to group districts with similar IPC crime patterns, potentially revealing underlying socio-economic or policing similarities that aren't immediately obvious. It employs forecasting models to offer tentative glimpses into potential near-future trends for specific crimes or cluster behaviors.

Ultimately, the story is about transforming raw data into actionable insights. It's a tool designed not just to present data, but to facilitate discovery.

Chapter 6

Tab-Specific Analysis and Features

The portal's functionality is organized into distinct tabs, each designed to answer specific questions about crime in India.

6.1 State Wise Tab

6.1.1 Functionality Overview

This tab provides a high-level, state-centric overview of crime patterns across India. It serves as an initial entry point for understanding the geographic distribution of various crime categories aggregated over the available years or for drilling down into a specific state's profile.

6.1.2 Core Visualization: Choropleth Map

The central element is an interactive choropleth map of India at the state level (`id="state-map"`).

- **Data Display:** The map visually encodes the total number of crimes for the selected category (IPC, SC, ST, Women, Children) aggregated across all available years in the respective dataset. States are colored based on the crime volume, typically using a sequential color scale (e.g., Viridis) where darker shades indicate higher crime counts.
- **Interactivity:** Hovering over a state reveals a tooltip displaying the state's name and the total aggregated crime count for the selected category. Clicking on a state selects it for further analysis within the tab.
- **GeoJSON Integration:** The map utilizes the `india_state.geojson` file for state boundaries, matched with the aggregated crime data using standardized state names (`featureidkey="properties.STATE_UPPER"`). Fuzzy name matching performed during data loading helps ensure maximum coverage.
- **Dynamic Zoom:** The map automatically adjusts its zoom level and center (`fitbounds="locations"`) to best fit the states with available data.

6.1.3 Interactive Elements

- **Crime Category Selection (`id="state-category-radio"`)**: A set of radio buttons allows the user to choose the primary crime category (IPC, SC, ST, Women, Children) to display on the map and analyze within the tab. Selecting a category updates the map's data and color scheme, as well as the options available for further analysis.
- **Map Click Interaction**: Clicking a state on the map triggers updates to other components within the tab. It stores the selected state's name (`id="selected-state-store"`) and displays it (`id="selected-state-display"`). Crucially, it populates the "State Analysis Options" section below the map.

- **State Analysis Options (`id="state-analysis-options"`)**: This container appears *after* a state is clicked. It dynamically generates:
  - **Year Range Slider (`id="state-year-slider"`)**: Allows selection of a specific year range relevant *only* to the data available for the *selected state* and *selected category.* Marks are dynamically generated based on available years for that state.
  - **Crime Type Dropdown (`id="state-crime-type-dropdown"`)**: Populated with specific crime types relevant to the *selected category* (e.g., "MURDER", "RAPE" for IPC), including a "TOTAL CRIMES" option. Allows the user to focus the subsequent visualizations on a particular crime within the chosen state and year range.
- **State-Specific Visualizations (`id="state-visualizations-container"`)**: This container displays graphs generated based on the selections made in the "State Analysis Options" section. Typically includes:
  - A bar chart showing the breakdown of the selected crime type by district within the chosen state and year range.
  - A line chart showing the trend of the selected crime type over the selected years for the chosen state.

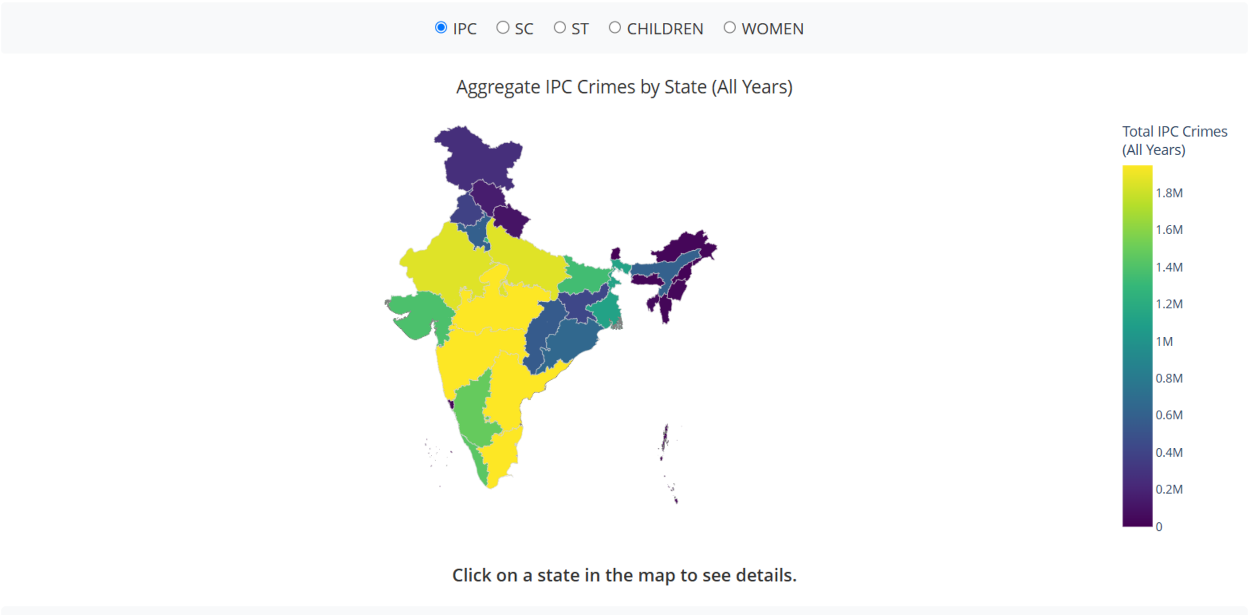
6.1.4 Underlying Callbacks

- `update_state_map`: Triggered by the `state-category-radio`. Aggregates data for the chosen category across all years by state and updates the main choropleth map (`state-map.figure`).
- `state_selected`: Triggered by clicking the `state-map` or changing the `state-category-radio`. Stores the selected state name (`selected-state-store.data`), updates the display text (`selected-state-display.children`), and dynamically creates the Year Range Slider and Crime Type Dropdown within the `state-analysis-options` container.
- `update_state_visualizations`: Triggered by changes in the dynamically generated `state-year-slider` or `state-crime-type-dropdown`. Filters the data for the selected state, category, years, and specific crime type, then generates the district-level bar chart and yearly trend line chart displayed in `state-visualizations-container.children`.

6.1.5 The Story: Gaining a High-Level Geographic Perspective

- Why start with a state-level view? Before diving into granular details, it's often essential to understand the bigger picture. This tab addresses the fundamental question: **"Where are different categories of crime most concentrated geographically across India?"**
- Imagine a national-level police analyst or policymaker. They need to quickly identify which states face the highest burden of IPC crimes overall, or which states report the most crimes against women or children. The initial map provides this immediate visual assessment. Seeing a state colored darkly prompts further investigation.
- The story then unfolds interactively. Clicking on that high-crime state allows the user to pivot their focus. "Okay, State X has high overall IPC crime. *Within* State X, which districts are the primary contributors? And has this been consistent over time, or are recent years different?" The dynamically appearing controls and subsequent charts allow the user to seamlessly transition from a national overview to a targeted state-level investigation, exploring both spatial distribution (which districts?) and temporal trends (how has it changed over the selected years?) for specific crimes or the total within that state. It's about moving from broad geographic patterns to focused state-level insights.

6.1.6 Dash-board Screenshots

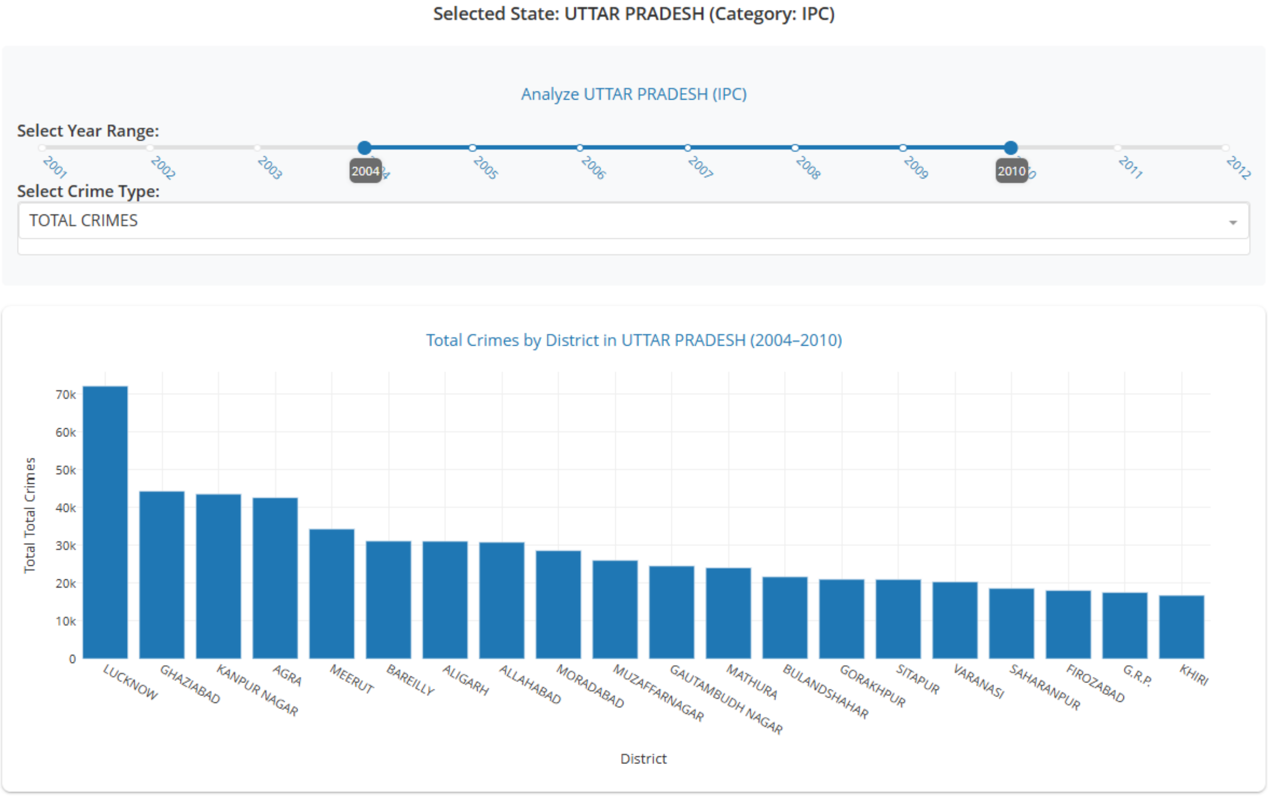


● IPC ○ SC ○ ST ○ CHILDREN ○ WOMEN

Aggregate IPC Crimes by State (All Years)

Click on a state in the map to see details.

Figure 6.1: Initial State Wise Tab view showing the IPC crime map



Selected State: UTTAR PRADESH (Category: IPC)

Analyze UTTAR PRADESH (IPC)

Select Year Range: 2004 – 2010

Select Crime Type: TOTAL CRIMES

Total Crimes by District in UTTAR PRADESH (2004–2010)

Figure 6.2: State Wise Tab after clicking on a state (e.g., Uttar Pradesh), showing the map, selected state display, and the dynamically generated Year Slider and Crime Type Dropdown

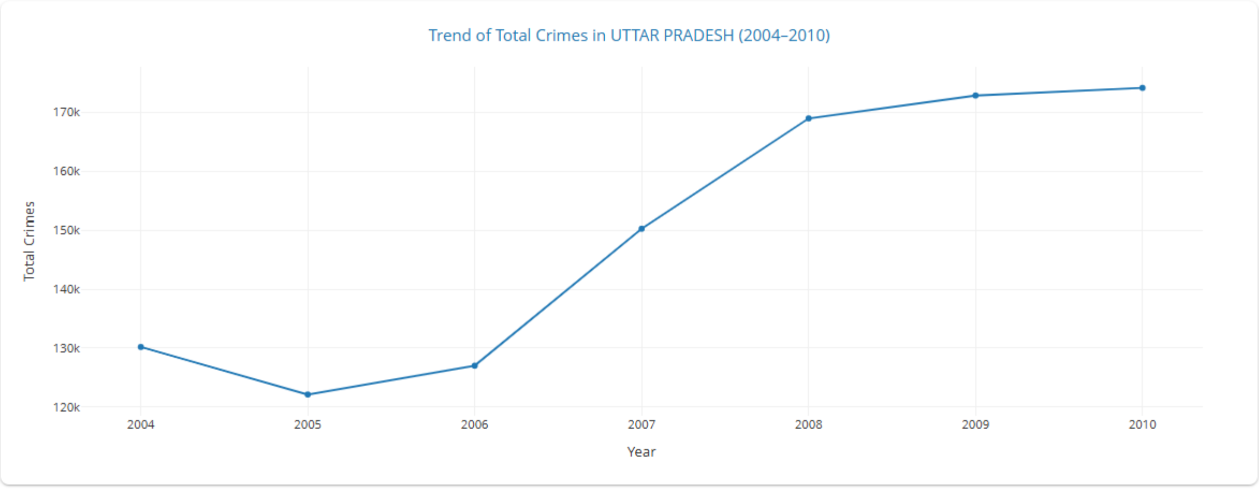


Figure 6.3: State Wise Tab showing the district-level bar chart and yearly trend line generated below the map after selecting a state, year range, and crime type

6.1.7 Our Key Observations from Sample Plots

- The map shows that northern states like UP and MP have way more crime reports (shown in yellow) than northeastern states (in dark purple). This pattern jumps out right away when looking at the national picture.
- In UP's district breakdown, Lucknow has about 70,000 crimes, which is way higher than other places. The biggest cities (Lucknow, Ghaziabad, Kanpur) all show up at the top of the chart, suggesting crime concentrates in urban areas.
- The line graph shows crime in UP actually dropped slightly from 2004 to 2005, but then shot up dramatically between 2006-2008. By 2010, crime levels were almost 50,000 cases higher than in 2005 - that's roughly a 40% increase in just five years.
- Having radio buttons to switch between IPC, crimes against SC/ST, children, and women lets users look for places where vulnerable groups might be specifically targeted.
- The bar chart shows huge differences between districts in the same state. The lowest reporting districts have about 1/4 the cases of the middle-ranked ones, and barely 1/5 of what Lucknow reports.

6.2 District Wise Tab

6.2.1 Functionality Overview

This tab facilitates a more granular geographic analysis, allowing users to explore crime patterns at the district level across India. It enables detailed investigation of specific crime types within selected years and supports comparison between districts.

6.2.2 Core Visualization: District Choropleth Map

The primary visualization is an interactive choropleth map of India showing district boundaries (`id="district-map"`).

- **Data Display:** Unlike the State Wise map which aggregates across all years by default, this map displays crime data aggregated *only* for the year range selected by the user via the `district-year-slider`. It shows data for the selected crime category (IPC, SC, etc.) and can display either the "TOTAL CRIMES" for that category or a specific crime type chosen from the `district-crime-dropdown`. Districts are colored based on the crime count for the selection.

- **Interactivity:** Hovering over a district reveals its name and the corresponding crime count for the selection. Clicking a district selects it for detailed analysis in a dedicated section below the map (`id="district-detail-graphs"`).
- **GeoJSON Integration:** Uses `india_district.geojson`. The `update_district_map_detailed` callback merges the aggregated crime data (for the selected years/category/crime) with a list of all districts derived from the GeoJSON. Districts present in the GeoJSON but having no matching crime data for the selection are displayed with a distinct color (e.g., light grey) or zero value, ensuring a complete map representation. Linkage relies on the standardized `DISTRICT_UPPER` key.
- **Dynamic Updates:** The map redraws whenever the category, year range, or specific crime type selection changes.

6.2.3 Interactive Elements

This tab features several control sections:

- **Main Controls (Row 1):**
  - **Crime Category Selection (`id="district-category-radio"`)**: Selects the overall crime category (IPC, SC, ST, Women, Children). This determines which dataset is used and updates the options in the year slider and crime dropdowns.
  - **Year Range Slider (`id="district-year-slider"`)**: Selects the time period for analysis. Min/max values and marks are dynamically updated based on the selected category's data availability.
  - **Specific Crime Dropdown (`id="district-crime-dropdown"`)**: Allows selection of a specific crime type within the chosen category (e.g., "BURGLARY", "THEFT" for IPC) or "TOTAL CRIMES". Options are updated based on the category selection. The map visualizes data for this specific selection.
- **District Map Interaction (Row 2):** Clicking the map selects a district.
- **Detailed District Analysis (Row 3):** Activated upon map click.
  - **Selected District Display (`id="selected-district-display"`)**: Shows the name of the clicked district.
  - **District Detail Graphs (`id="district-detail-graphs"`)**: Displays visualizations specific to the *selected district* across *all available years* for the chosen category. Typically includes:
    - A time series line chart showing the trend of "TOTAL CRIMES" for that district.
    - A time series line chart showing the trend of the "selected crime" for that district.
    - A pie chart showing the breakdown of specific crime types (within the selected category) aggregated over all years for that district.
- **District Comparison (Row 4):**
  - **Multi-Select Dropdown (`id="compare-districts-multi"`)**: Allows the user to select two or more districts for side-by-side comparison. District options are populated based on the selected category.
  - **Comparison Graphs (`id="district-comparison-graphs"`)**: Displays visualizations comparing the selected districts based on the crime type chosen in the *main* `district-crime-dropdown` and the selected year range. Typically includes:
    - A multi-line chart showing the trend for each selected district.
    - A grouped bar chart comparing the total count for the period across the selected districts.

- **Crime Hotspot Analysis (Row 5):**
  - **Crime Hotspot Dropdown (`id="crime-hotspot-dropdown"`)**: Allows selection of a *specific* crime type (cannot be "TOTAL CRIMES") for which to identify top districts. Options are based on the selected category.
  - **Top N Slider (`id="crime-hotspot-top-n-slider"`)**: Selects how many top districts to display (e.g., Top 5, Top 10).
  - **Hotspot Graph (`id="crime-comparison-graphs"`)**: Displays a bar chart showing the top N districts with the highest count for the selected *specific* crime type over the selected year range.

6.2.4 Underlying Callbacks

- `update_district_controls`: Triggered by `district-category-radio`. Updates the min / max / value / marks of the year slider, and the options/value for the main crime dropdown, the district comparison multi-select dropdown (options only), and the crime hotspot dropdown.

- `update_district_map_detailed`: Triggered by changes in category, year range, or main crime dropdown. Filters/aggregates data, merges with GeoJSON districts, and updates the main district map (`district-map.figure`).

- `district_map_click_handler`: Triggered by clicking the `district-map`. Stores the selected district name and category (`selected-district-store.data`) and updates the display text (`selected-district-display.children`).

- `update_district_detail_visualizations`: Triggered by changes in the `selected-district-store`. Filters data for the stored district/category across all years and generates the time series and crime breakdown pie chart in `district-detail-graphs.children`.

- `update_district_comparison`: Triggered by changes in the `compare-districts-multi` dropdown. Filters data for the selected districts, category, year range, and main crime type, then generates the comparative line and bar charts in `district-comparison-graphs.children`.

- `update_crime_hotspots`: Triggered by changes in the `crime-hotspot-dropdown` or `crime-hotspot-top-n-slider`. Filters data for the category/years, aggregates for the specific hotspot crime, finds the top N districts, and generates the hotspot bar chart in `crime-comparison-graphs.children`. (Note: The code seems to output this to the same div as the district comparison; this might be intentional or a potential area for refinement to have separate output divs).

6.2.5 The Story: Drilling Down to Local Crime Dynamics

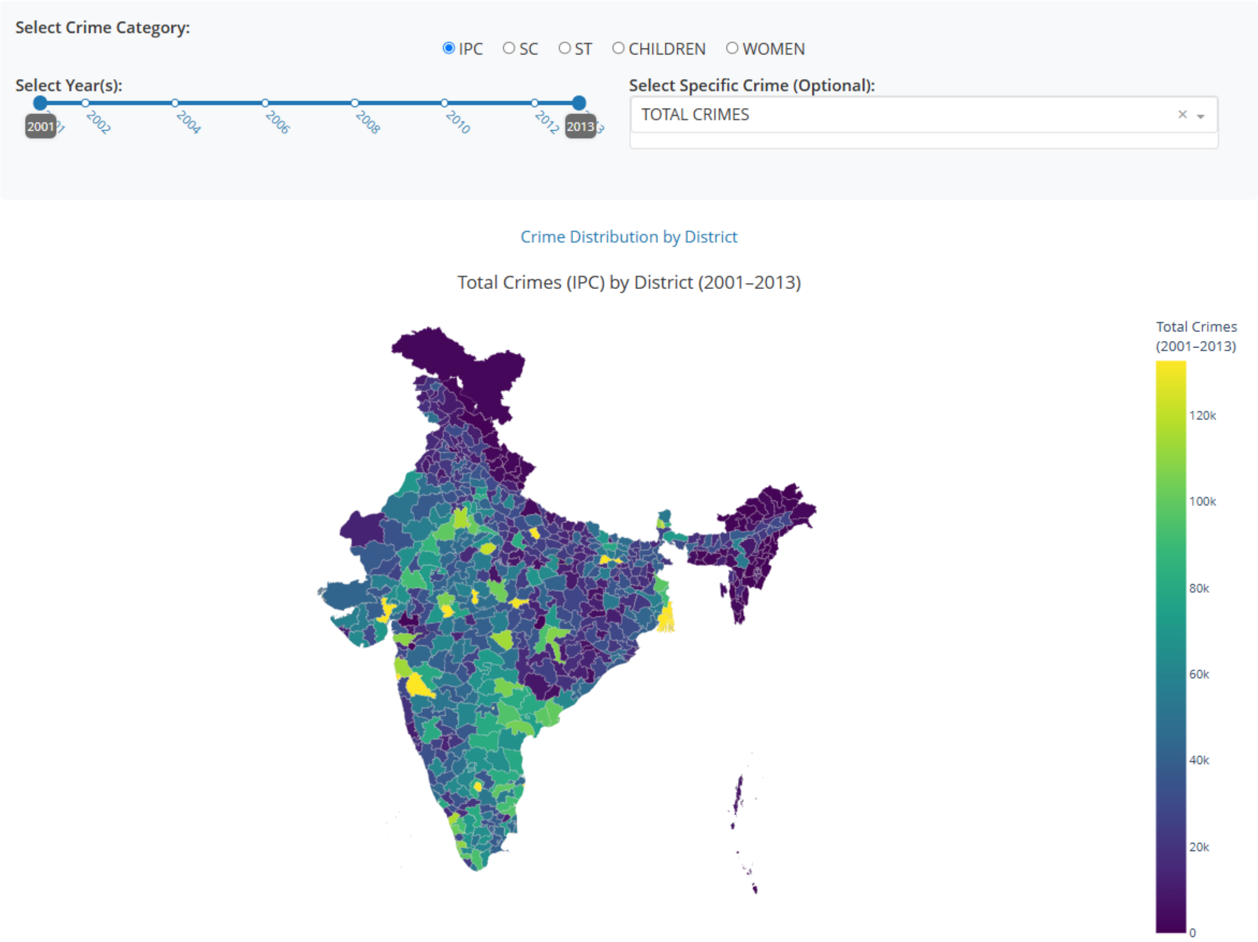
- While state-level analysis provides context, crime is often a localized phenomenon. This tab addresses the need to **"understand crime patterns at the district level, compare specific districts, and identify high-incidence areas (hotspots)."**

- Imagine an analyst investigating theft in a particular state identified from the previous tab. Using the District Wise tab, they can select the "IPC" category, filter for "THEFT", and choose a relevant year range. The map immediately shows which districts had the highest reported theft cases during that period. They might notice neighboring districts with vastly different theft rates. Clicking on these districts allows for detailed exploration: How did theft trend over time in each district individually? What other crimes are prevalent there?

- Furthermore, the comparison feature allows direct juxtaposition. The analyst can select both districts in the multi-select dropdown to see their theft trends plotted on the same axes and their total theft counts compared in a bar chart for the chosen period. This facilitates identifying divergent trends or confirming similarities.

- Finally, the hotspot analysis answers the question: "For a *specific* crime like 'BURGLARY' in this category and timeframe, which districts consistently rank highest across the region/country?" This helps pinpoint areas requiring targeted attention or further investigation, moving beyond anecdotal evidence to data-driven identification of critical zones. This tab, therefore, supports granular investigation, comparative analysis, and hotspot identification at the local level.

6.2.6 Dash-board Screenshots



Select Crime Category:

● IPC ○ SC ○ ST ○ CHILDREN ○ WOMEN

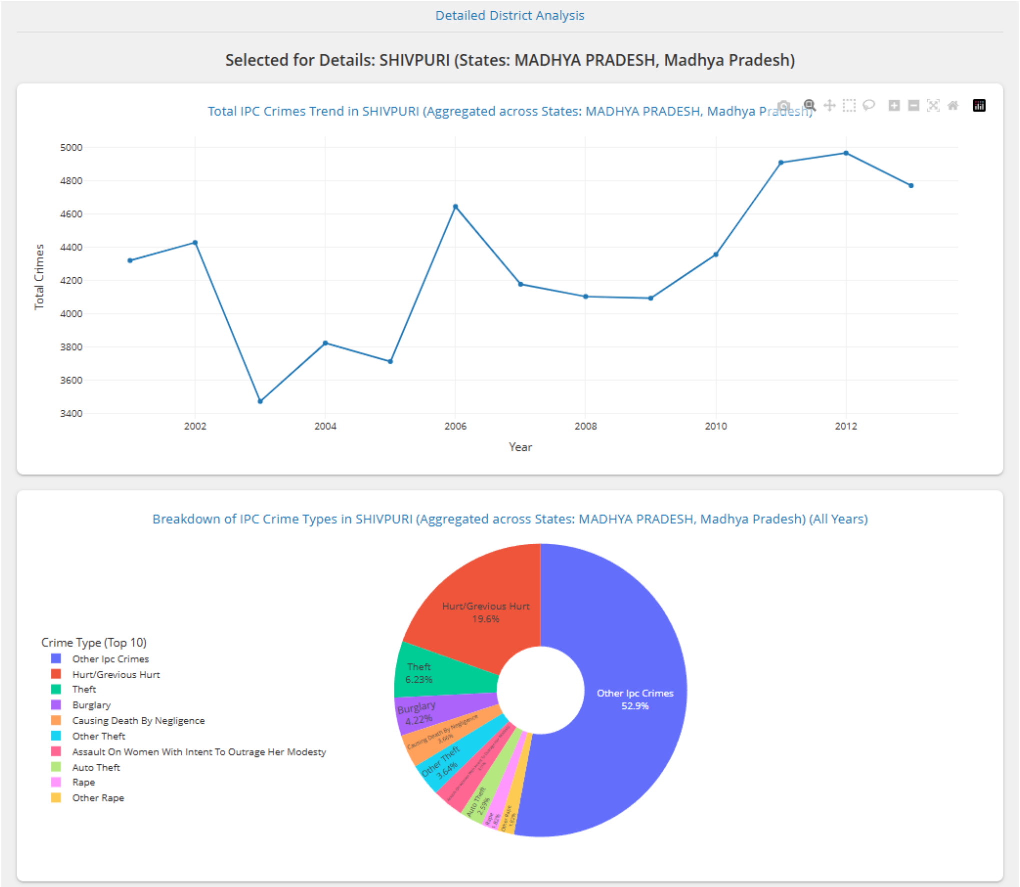
Select Year(s): 2001 – 2013

Select Specific Crime (Optional): TOTAL CRIMES

Crime Distribution by District

Total Crimes (IPC) by District (2001–2013)

Figure 6.4: District Wise Tab view showing the district map for IPC/Total Crimes for a selected year range

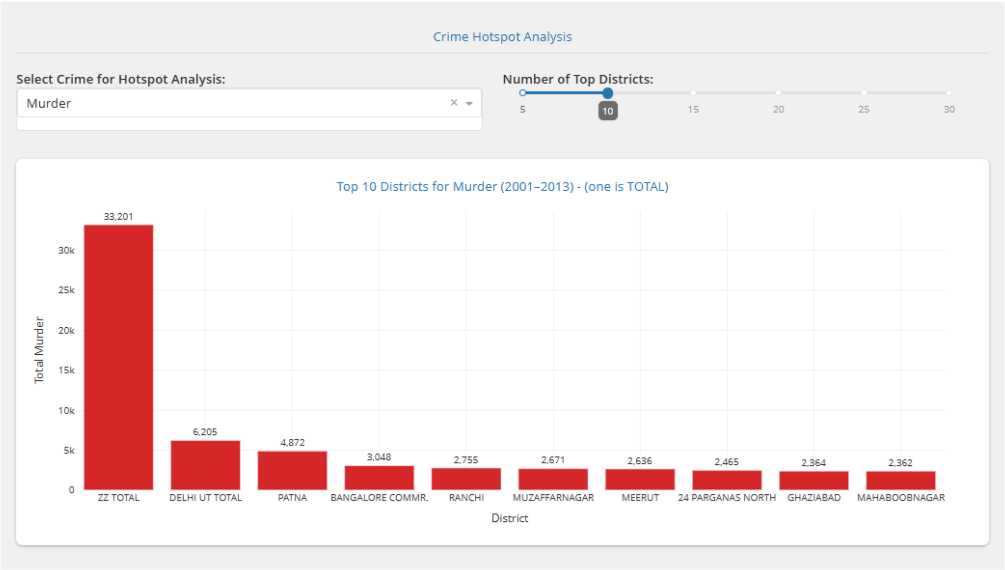


Detailed District Analysis

Selected for Details: SHIVPURI (States: MADHYA PRADESH, Madhya Pradesh)

Total IPC Crimes Trend in SHIVPURI (Aggregated across States: MADHYA PRADESH, Madhya Pradesh)
Total Crimes
5000
4800
4600
4400
4200
4000
3800
3600
3400
2002
2004
2006
2008
2010
2012
Year
Breakdown of IPC Crime Types in SHIVPURI (Aggregated across States: MADHYA PRADESH, Madhya Pradesh) (All Years)
Crime Type (Top 10)
Other Ipc Crimes
Hurt/Grevious Hurt
Theft
Burglary
Causing Death By Negligence
Other Theft
Assault On Women With Intent To Outrage Her Modesty
Auto Theft
Rape
Other Rape
Hurt/Grevious Hurt 19.6%
Theft 6.23%
Burglary 4.22%
Other Theft 3.64%
Other Ipc Crimes 52.9%

Figure 6.5: District Wise Tab after clicking a district, showing the map and the 'Detailed District Analysis' section with time series and pie chart



Crime Hotspot Analysis

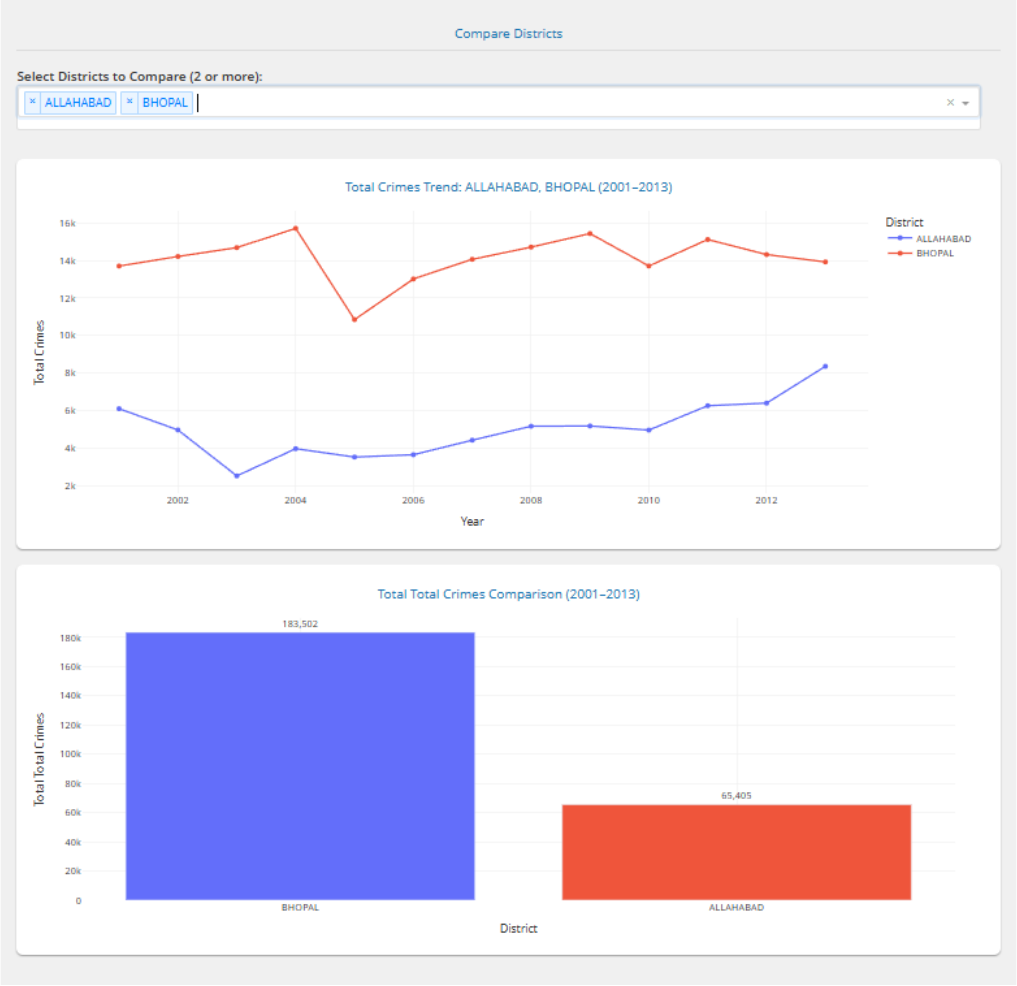
Select Crime for Hotspot Analysis: Murder

Number of Top Districts: 10

Figure 6.7: District Wise Tab showing the 'Crime Hotspot Analysis' section with controls and the resulting top N districts bar chart

6.2.7 Our Key Observations from Sample Plots

- The time series for Shivpuri district shows several peaks and valleys between 2001-2013, with a significant drop around 2003 (to about 3,450 cases) followed by recovery and another major peak around 2011-2012 (nearly 5,000 cases). This uneven pattern suggests local factors affecting crime rates.



Compare Districts

Select Districts to Compare (2 or more):

× ALLAHABAD × BHOPAL

Figure 6.6: District Wise Tab showing the 'Compare Districts' section with the multi-select dropdown and the resulting comparison line/bar charts

pie chart for Shivpuri shows that while "Other IPC Crimes" account for about 53% of cases, "Hurt/Grievous Hurt" makes up nearly 34% - a specific crime category that local authorities might want to address with targeted interventions.

- The comparison feature reveals Bhopal consistently maintained much higher crime levels (around 14,000 cases annually) than Allahabad (around 4,000-6,000 cases), with both districts showing a dip around 2004 that might warrant investigation into policy changes during that period.
- The aggregate comparison shows Bhopal reporting almost three times more total crimes than Allahabad (183,502 vs 65,405), highlighting how crime reporting or actual crime rates can differ dramatically between major urban centers.
- The murder hotspot chart shows extremely uneven distribution, with one area reporting over 33,000 cases while even major cities like Delhi (6,205) and Patna (4,872) show much lower figures. This concentration suggests where homicide prevention efforts might be prioritized.
- The ability to identify top districts for specific crimes like murder creates clear priorities for resource allocation, with the chart showing dramatic differences even among the top 10 districts (Delhi has more than twice the murders of Bangalore).

6.3 Year Wise Tab

6.3.1 Functionality Overview

This tab focuses specifically on temporal analysis, allowing users to explore how crime trends have evolved over time across India or within specific categories and crime types. It provides different lenses (trend, comparison, breakdown) to view yearly changes.

6.3.2 Core Visualizations: Trend Lines, Bar Charts, Pie Charts

Unlike the geographically focused tabs, this tab presents national-level summaries visualized through standard time-series and comparison charts within the `year-visualizations-container`. The specific charts depend on the "Visualization Type" selected:

- **Trend Analysis:**
  - **Line Chart:** Shows the national total count of the selected crime type/category plotted against the years in the selected range. Highlights the overall upward or downward trend.
  - **Bar Chart:** Displays the year-over-year percentage change for the selected crime/category, indicating the rate of increase or decrease annually. Bars are typically colored green for increase and red for decrease.
- **State Comparison:**
  - **Horizontal Bar Chart:** Ranks the top N (e.g., 15) states based on their total count for the selected crime/category aggregated over the chosen year range. Helps identify states with the highest cumulative burden during the period.
  - **Pie Chart:** Shows the percentage contribution of the top 5 states (and an "Other States" category) to the national total for the selected crime/category over the chosen years. Illustrates the concentration of crime.
- **Crime Type Breakdown:**
  - **Pie Chart:** Displays the proportional breakdown of all specific crime types *within* the selected *category* aggregated nationally over the selected year range. Shows which specific crimes dominate the category.
  - **Horizontal Bar Chart:** Ranks the top N (e.g., 10) specific crime types *within* the selected category by their total national count over the chosen years.

6.3.3 Interactive Elements

- **Year Range Slider (`id="year-slider"`):** Selects the time period for analysis. The range covers all available years across all loaded datasets. Marks help orient the user.
- **Crime Category Selection (`id="year-category-radio"`):** Chooses the primary crime category (IPC, SC, ST, Women, Children). Updates the crime type dropdown options.
- **Crime Type Dropdown (`id="year-crime-type-dropdown"`):** Selects either "TOTAL CRIMES" for the chosen category or a specific crime type within that category (e.g., "DACOITY", "RIOTS" for IPC). Options are dynamically populated based on the category selection.
- **Visualization Type Radio (`id="year-viz-type-radio"`):** Allows the user to switch between the three modes of temporal analysis: "Trend Analysis", "State Comparison", and "Crime Type Breakdown". This selection determines which set of charts (as described above) is generated.

6.3.4 Underlying Callbacks

- `update_year_crime_types`: Triggered by `year-category-radio`. Updates the `options` of the `year-crime-type-dropdown` based on the available crimes for the selected category, always including "TOTAL CRIMES".

- `set_default_year_crime_type`: Triggered when the `options` of the `year-crime-type-dropdown` change. Sets the default `value` to "TOTAL CRIMES" if available, otherwise to the first option.

- `update_year_visualizations`: The main callback for this tab. Triggered by changes in the year slider, category radio, crime type dropdown, or visualization type radio. Based on the `viz_type` selected, it filters the appropriate DataFrame, performs the necessary aggregations (national trends, state totals, or crime type totals), and generates the corresponding set of Plotly figures (line/bar for trend, bar/pie for state comparison, pie/bar for breakdown) displayed in `year-visualizations-container.children`.

6.3.5 The Story: Understanding Temporal Crime Evolution

- While geographic analysis shows *where* crime occurs, understanding *when* and *how* it changes over time is equally critical. This tab answers questions like: **"How has the national count of a specific crime (or category) changed year over year? Which states contributed most significantly during a particular period? Within a crime category, which specific offenses are most dominant?"**

- Consider a researcher studying trends in theft (IPC). They can select "IPC", "THEFT", and the desired year range. Choosing "Trend Analysis" reveals the national theft trend line and the year-over-year percentage changes, instantly showing periods of increase or decrease. Switching to "State Comparison" shows which states had the most reported thefts cumulatively during those years and the contribution of the top states to the national total. If they were interested in the IPC category overall, selecting "TOTAL CRIMES" and "Crime Type Breakdown" would show which specific IPC crimes (like theft, robbery, hurt, etc.) made up the largest portions of the total reported IPC incidents nationally during the selected period.

- This tab allows users to shift focus from geography to chronology, analyzing national-level temporal dynamics, comparing state contributions over time, and dissecting the composition of broader crime categories year by year. It provides the temporal context often missing from purely spatial analyses.

6.3.6 Dash-board Screenshots

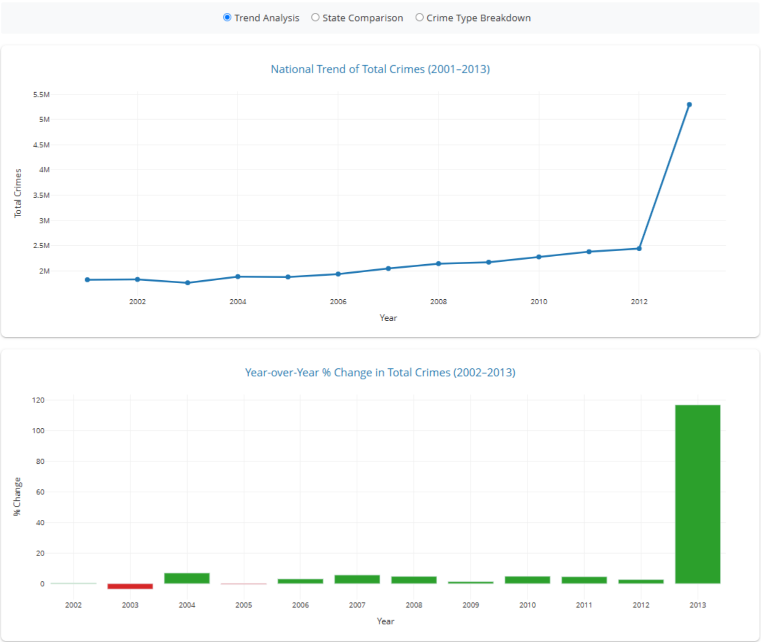


Figure 6.8: Year Wise Tab showing Trend Analysis view - National Line Chart and YoY Bar Chart for a selected crime/category/year range

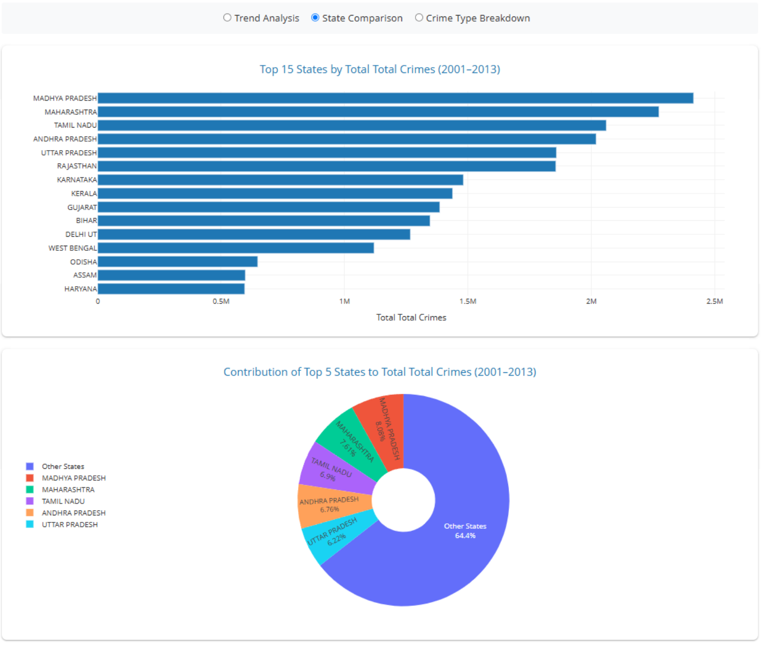
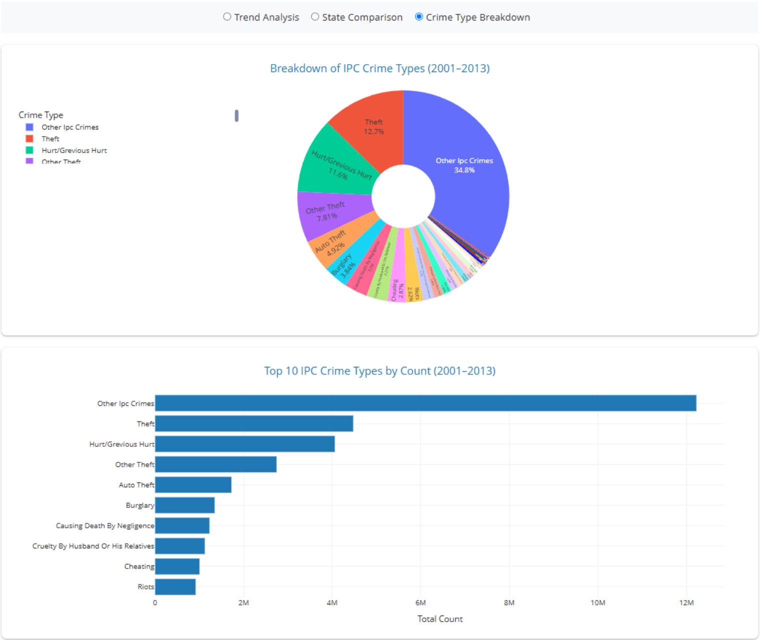


Figure 6.9: Year Wise Tab showing State Comparison view - Top States Bar Chart and Contribution Pie Chart



○ Trend Analysis ○ State Comparison ● Crime Type Breakdown

Figure 6.10: Year Wise Tab showing Crime Type Breakdown view - Pie Chart and Top Crime Types Bar Chart for a selected category/year range

6.3.7 Our Key Observations from Sample Plots

- The trend line shows crime stayed relatively steady from 2001-2012 (between 2.5-3 million cases), but then jumped dramatically in 2013 to nearly 5 million cases. This massive increase of about 145% in a single year stands out immediately and raises questions about what happened.
- Looking at the bar chart of yearly changes, most years show small positive growth (green bars) of roughly 5-10%, with only one slight decrease around 2003 (red bar). This pattern suggests a gradual worsening situation until the 2013 explosion.
- Madhya Pradesh tops the state ranking with approximately 750,000 total crimes during 2001-2013, followed closely by Maharashtra. The gap between the top states and those lower on the list (like Bihar and Assam) is quite large, showing uneven distribution.
- The pie chart reveals that despite having high individual numbers, the top 5 states together account for only about 35% of all crimes nationwide. The "Other States" slice (about 65%) shows that crime is actually spread widely across India rather than concentrated in a few hotspots.
- The crime breakdown shows "Other IPC Crimes" make up the largest segment (around 35-40%), followed by theft (approximately 18%) and hurt/grievous hurt (about 15%). This suggests many crimes don't fit neatly into major categories.

6.4 Area Comparison Tab

6.4.1 Functionality Overview

This tab provides a specific comparative visualization focused on firearm involvement in violent crime (presumably murder, based on the likely source dataset `34_Use_of_fire_arms_in_murder_cases.csv`). It allows a direct visual comparison between two selected areas (States/UTs) across different metrics related to firearm use.

6.4.2 Core Visualization: Radar Chart

The sole visualization in this tab is a Radar Chart (also known as a spider chart) (`id="area-comparison-radar-plot"`).

- **Data Display:** The chart compares two selected areas based on aggregated values for predefined metrics. Based on the data loading code for `df_aggregated` and the callback `update_area_comparison_radar`, these metrics are likely:
  - Total Victims (presumably murder victims)
  - Victims by Registered Arms
  - Victims by Unregistered Arms
- **Comparison:** Each selected area is represented by a different colored shape on the radar. The distance of a point from the center along each axis indicates the magnitude of that metric for the area. This allows for a quick visual assessment of which area scores higher on each dimension and the overall "shape" of firearm involvement profile for each area.
- **Aggregation:** The data displayed (`df_aggregated`) represents totals summed across all available years in the source dataset (`34_Use_of_fire_arms_in_murder_cases.csv`).

6.4.3 Interactive Elements

- **Area Selection Dropdowns (`id="compare-area-dropdown-1"`, `id="compare-area-dropdown-2"`):** Two separate dropdowns allow the user to select the two areas (States/UTs) they wish to compare. The options are populated from the unique `Area_Name` values (standardized during loading) present in the `df_aggregated` DataFrame.

6.4.4 Underlying Data: Firearm Use (`df_aggregated`)

The data for this tab comes from preprocessing the `34_Use_of_fire_arms_in_murder_cases.csv` file. The script aggregates this data by `Area_Name` (standardized), summing up the counts for `Victims`, `By Registered Arms`, and `By Unregistered Arms` across all years into the `df_aggregated` DataFrame. Standardized column names are used in the aggregation and plotting process.

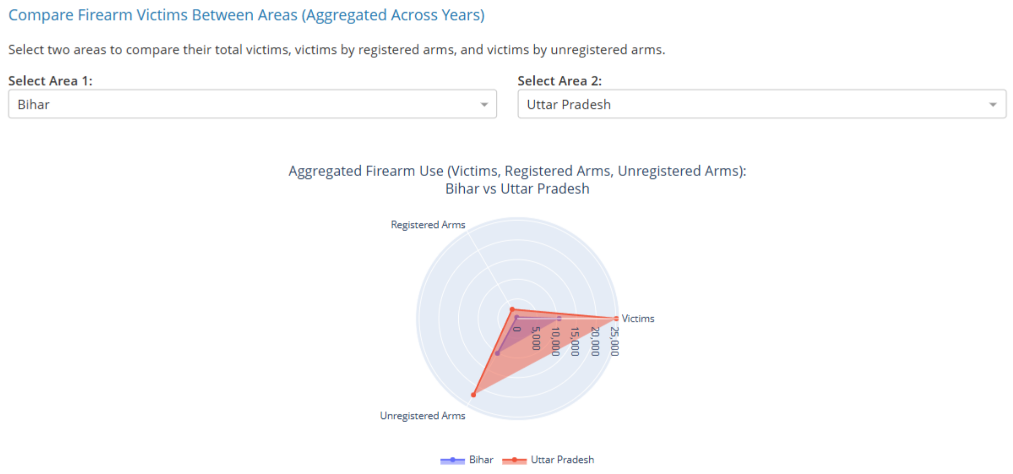
6.4.5 Underlying Callback (`update_area_comparison_radar`)

This callback is triggered when the value in either of the area selection dropdowns changes. It retrieves the aggregated data for the two selected areas from `df_aggregated`, extracts the values for the comparison metrics, formats the category labels, and generates the `go.Scatterpolar` traces for the radar chart, updating `area-comparison-radar-plot.figure`.

6.4.6 The Story: Comparative Analysis of Firearm Involvement

- Firearm usage is a critical aspect of violent crime analysis. This tab addresses the specific comparative question: "How do two specific states/UTs compare in terms of total murder victims and the involvement of registered vs. unregistered firearms in those incidents?"
- Imagine a researcher studying gun control policies. They might want to compare two states with different regulations. By selecting the two states in the dropdowns, they can immediately see on the radar chart: Does State A have significantly more total victims than State B? In State A, are most firearms used unregistered, while in State B registered arms are more common?
- The radar chart provides a concise visual summary of these comparisons across the three dimensions simultaneously. It helps identify distinct profiles of firearm involvement, prompting further questions about the reasons for these differences (e.g., effectiveness of licensing, prevalence of illegal arms). It's a focused tool for a specific type of comparative analysis relevant to violence and weapon control discussions.

6.4.7 Dash-board Screenshots



Compare Firearm Victims Between Areas (Aggregated Across Years)

Select two areas to compare their total victims, victims by registered arms, and victims by unregistered arms.

Select Area 1: Bihar

Select Area 2: Uttar Pradesh

Aggregated Firearm Use (Victims, Registered Arms, Unregistered Arms): Bihar vs Uttar Pradesh

Figure 6.11: Area Comparison Tab showing the two dropdowns and the resulting Radar Chart comparing two selected states/UTs

6.4.8 Our Key Observations from Sample Plots

- The radar chart effectively compares Bihar and Uttar Pradesh across three key dimensions simultaneously (total victims, registered firearm victims, and unregistered firearm victims).
- Uttar Pradesh shows significantly higher numbers across all three dimensions compared to Bihar, with the total victims axis reaching what appears to be around 6000-7000 victims for UP versus a much lower figure for Bihar.
- Both states show a pattern where unregistered firearms account for substantially more victims than registered firearms, though this disparity appears more pronounced in Uttar Pradesh.
- The difference in registered versus unregistered firearm victims in both states suggests potential issues with firearm regulation effectiveness.
- While UP has higher absolute numbers, the similar triangular shapes suggest both states face comparable challenges with unregistered weapons.

6.5 Place Occurrence Tab

6.5.1 Functionality Overview

This tab explores the spatial context *within* which crimes occur, moving beyond administrative boundaries (state/district) to specific types of locations.

6.5.2 Core Visualizations: Sunburst Chart, Small Multiples Bar Chart

- **Sunburst Chart (`id="place-sunburst"`)**: This hierarchical visualization shows the breakdown of crime counts based on the selected year range.
  - **Hierarchy:** The levels likely represent Year → Place Type → Crime Type (based on `path=['YEAR','PLACE','CRIME']`). Inner rings represent years, the next ring shows different places (e.g., "Residential Premises", "Highway", "Railway"), and the outer ring breaks down crimes occurring in those places. The size of each segment corresponds to the aggregated crime count.

  - **Interaction:** Users can click on segments to zoom in and explore specific branches of the hierarchy. Hovering reveals the count for the specific level.

- **Small Multiples Bar Chart (`id="place-small-multiples"`):** This visualization provides a faceted view, showing trends over time for different places.
  - **Structure:** The chart creates separate bar charts (facets) for each 'PLACE' category. Within each facet, the x-axis represents the 'YEAR' (within the selected range), the y-axis represents the 'COUNT', and different colored bars represent different 'CRIME' types occurring at that place.
  - **Insight:** This allows users to compare the temporal trends of various crimes across different types of locations simultaneously.

6.5.3 Interactive Elements

- **Year Range Slider (`id="place-year-slider"`):** Allows the user to select the start and end year (between 2001 and 2012, based on the likely dataset range) for filtering the place of occurrence data used in both visualizations.

6.5.4 Underlying Data: Place of Occurrence (`df_place_long`)

The data originates from `17_Crime_by_place_of_occurrence_2001_2012.csv`. The code preprocesses this by melting the numerous `PLACE - CRIME` columns into a long format DataFrame (`df_place_long`) containing `STATE`, `YEAR`, `PLACE`, `CRIME`, and `COUNT` columns. This structure is suitable for generating the hierarchical sunburst and the faceted bar chart.

6.5.5 Underlying Callback (`update_place_viz`)

Triggered by changes in the `place-year-slider`. It filters `df_place_long` based on the selected year range, ensures the 'COUNT' is numeric, and then generates both the Sunburst figure (`place-sunburst.figure`) and the Small Multiples Bar Chart figure (`place-small-multiples.figure`).

6.5.6 The Story: Identifying Where Crimes Occur

- Understanding *where* crimes happen provides crucial context for prevention and policing strategies. This tab helps answer: **"What types of locations are most common for different crimes, and how has this changed over the selected years?"**

- A crime prevention unit might use this tab to analyze patterns within their jurisdiction (though the provided data seems aggregated at the state level in `df_place_long`, the visualization concepts apply).

- The sunburst chart gives a quick overview: are residential premises the dominant location for crimes overall in the selected period? Clicking deeper might reveal that while residential areas see many crimes, specific types like robbery might be more prevalent on highways.

- The small multiples view complements this by showing trends: Has crime on railways increased or decreased over the years compared to crime in markets? Are certain crimes becoming more common in specific locations recently?

- This information can guide resource deployment (e.g., increased patrols in specific location types) and inform situational crime prevention strategies tailored to the environments where crimes are most likely to occur.

6.5.7 Dash-board Screenshots

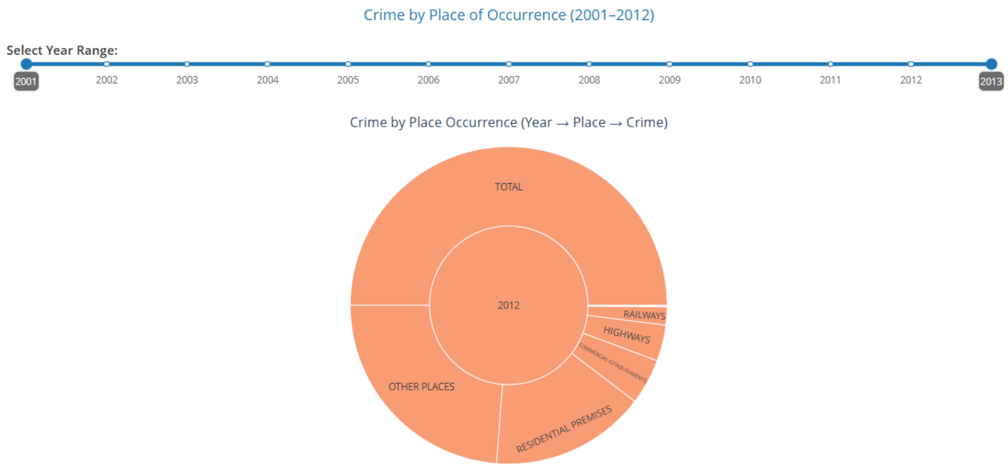


Figure 6.12: Place Occurrence Tab showing the Sunburst Chart for a selected year range

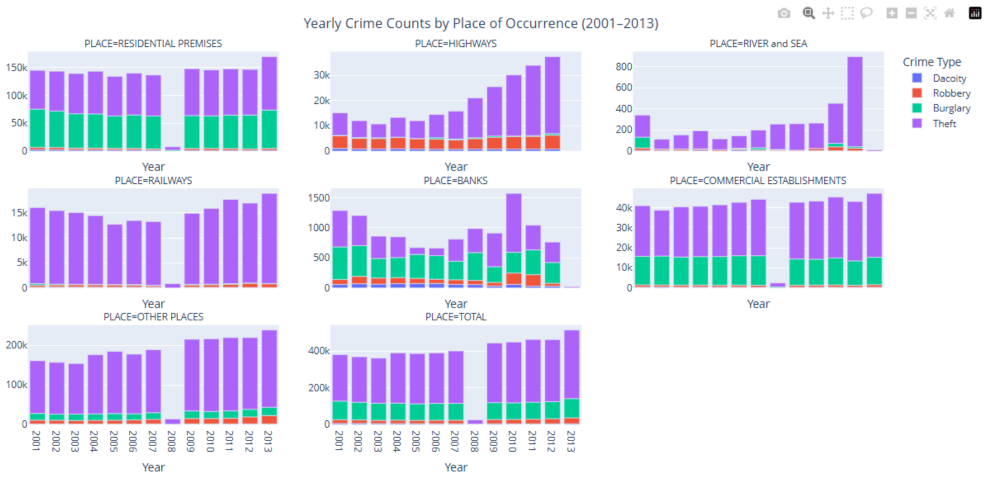


Figure 6.13: Place Occurrence Tab showing the Small Multiples Bar Chart visualization

6.5.8 Our Key Observations from Sample Plots

- The small multiples chart shows highway crimes nearly doubled over the period, rising from about 10,000 cases to almost 20,000, suggesting a growing need for highway patrol resources.

- The purple segments (theft) appear as the largest crime type in almost every location category, especially in residential premises and commercial establishments, helping focus prevention strategies.

- After relatively low crime numbers for years, river and sea locations show a dramatic jump in 2013 to around 800 cases, mostly thefts. This unexpected pattern demands investigation into what changed.

- Crime on railways declined during 2005-2008 but then increased again, returning to earlier levels by 2013. This pattern might reflect changes in railway security measures that initially worked but later lost effectiveness.

- Unlike other locations, crimes in commercial establishments maintain relatively steady levels throughout the period, suggesting existing security measures may be providing consistent protection.

6.6 Murder Victims Flow Tab

6.6.1 Functionality Overview

This tab visualizes the flow of murder victims through different demographic categories (State → Gender → Age Group) for a selected year using a Sankey diagram. It helps understand the composition of victims within top affected states.

6.6.2 Core Visualization: Sankey Diagram

The primary element is a Sankey diagram (`id="sankey-diagram"`).

- **Data Flow:** Sankey diagrams are used to depict flows between nodes. In this context, it shows the flow of murder victims:
  - From the selected **Top N States** (source nodes on the left).
  - To **Gender** categories (intermediate nodes, e.g., 'Male Victims', 'Female Victims').
  - To broad **Age Groups** (destination nodes on the right, e.g., 'Adult', 'Non-Adult' - based on aggregation logic in the callback).
- **Link Thickness:** The width of the links (flows) between nodes is proportional to the number of victims represented in that flow (e.g., a thicker link from State A to 'Female Victims' indicates more female victims from State A compared to a state with a thinner link).
- **Insight:** Allows users to visually trace and compare the demographic pathways of murder victims originating from the most affected states in a given year.

6.6.3 Interactive Elements

- **Year Selection Dropdown (`id="year-dropdown"`):** Allows the user to select a specific year for which to view the murder victim flow. Options are populated from the years available in the `df_murder` dataset.
- **Top States Slider (`id="states-slider"`):** Allows the user to select the number (N) of top states (ranked by total victims in the selected year) to include as source nodes in the Sankey diagram.

6.6.4 Underlying Data: Murder Victim Age/Sex (`df_murder`)

The data comes from `32_Murder_victim_age_sex.csv`. The `update_sankey` callback processes this data for the selected year:

- It identifies the top N states based on total victims.
- It filters the data for these top states and specific gender subgroups (e.g., '1. Male Victims', '2. Female Victims').
- It aggregates counts across more detailed age categories into broader groups like 'Adult' and 'Non-Adult'. This aggregation logic is crucial for the final Sankey structure.
- It then constructs the source, target, and value lists required by Plotly's `go.Sankey` trace.

6.6.5 Underlying Callback (`update_sankey`)

Triggered by changes in the `year-dropdown` or `states-slider`. It performs the data filtering, aggregation, and Sankey data structure preparation described above, then generates the Plotly Sankey figure (`sankey-diagram.figure`). It includes logic to handle cases with missing data or subgroups and assigns colors to nodes based on their category (State, Gender, Age).

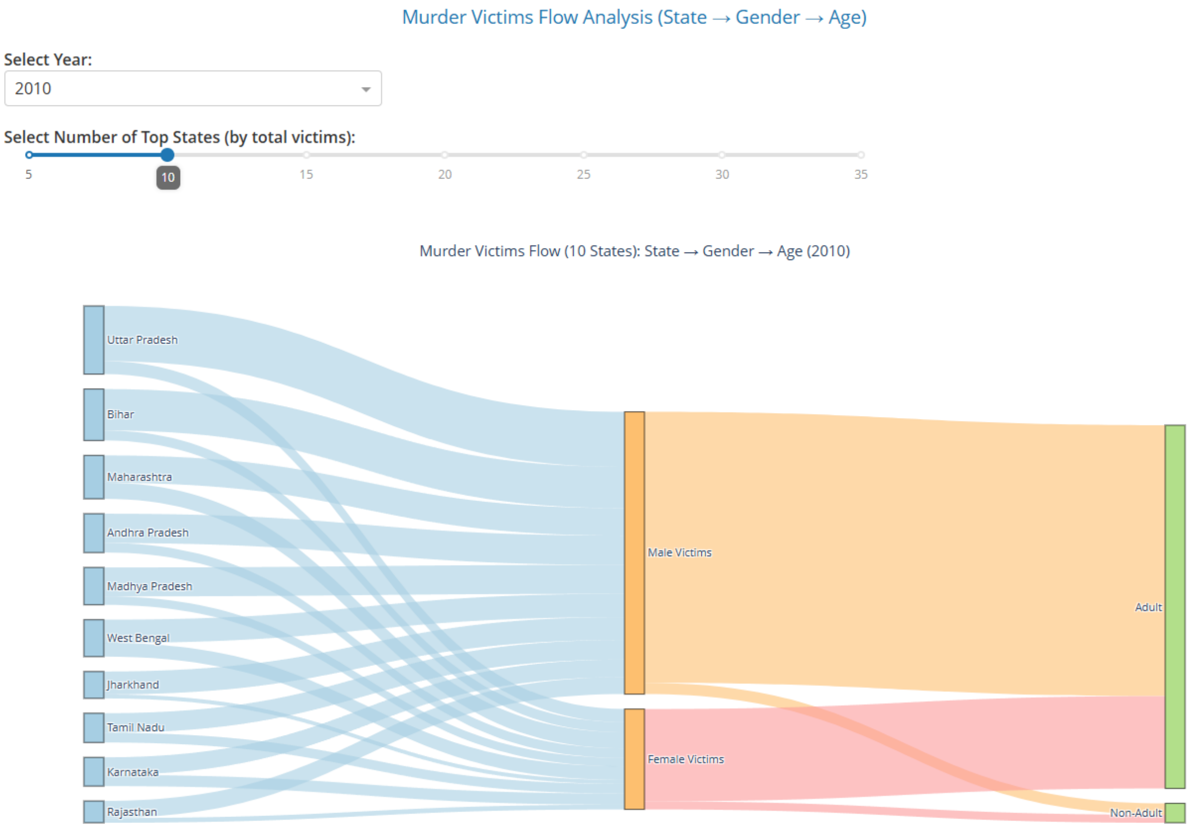
6.6.6 The Story: Visualizing the Flow of Murder Victim Demographics

- Aggregate murder statistics only tell part of the story. To understand the human cost and potential risk factors, analyzing victim demographics is essential. This tab asks: **"For a given year, considering the states with the most murders, what is the flow of victims through gender and broad age categories?"**

- A sociologist or public health official might use this to investigate patterns. For instance, in State X (one of the top states), does the flow predominantly go through 'Male Victims' and then to the 'Adult' age group? Does State Y show a different pattern, perhaps with a larger flow through 'Female Victims'?

- The Sankey diagram makes these complex multi-stage breakdowns visually intuitive. Seeing a thick band flowing from a state to a specific gender and age group highlights a potential area of concern or a demographic profile at higher risk within that state for that year. It transforms tables of numbers into a visual narrative of victim pathways.

6.6.7 Dash-board Screenshots



Murder Victims Flow Analysis (State → Gender → Age)

Select Year: 2010

Select Number of Top States (by total victims): 10

Figure 6.14: Murder Victims Flow Tab showing the Sankey Diagram for a selected year and number of top states

6.6.8 Our Key Observations from Sample Plots

- The diagram clearly shows Uttar Pradesh having the widest flow band, indicating it has significantly more murder victims than other states in 2010. Bihar appears to be second, with Maharashtra following closely behind.

- The final stage shows most victims are adults, with relatively thin bands flowing to the Non-Adult category. This age pattern appears consistent across both gender categories.

- Northern and central states (UP, Bihar, MP) appear to have higher victim counts than southern states like Karnataka and Tamil Nadu, suggesting potential regional factors worth investigating.

6.7 Offender Relationships Tab

6.7.1 Functionality Overview

This tab explores the relationship between victims and known offenders, specifically focusing on cases where the offender was known to the victim prior to the crime. It uses data likely from `21_Offenders_known_to_the_victim.csv`.

6.7.2 Core Visualization: Treemap

The main visualization is a Treemap (`id="relative-treemap"`).

- **Hierarchy:** Treemaps are used to display hierarchical data as nested rectangles. Here, the hierarchy is likely State → Relationship Type.
  - The top level represents the selected Top N states (based on the total number of cases where offenders were known).
  - Within each state's rectangle, smaller rectangles represent different relationship categories (e.g., 'Relatives', 'Neighbors', 'Other Known Persons').
- **Size and Color:** The area of each rectangle is proportional to the number of cases ('Count') it represents. Rectangles might be colored based on the relationship type, allowing quick identification of prevalent offender categories within and across states.
- **Insight:** Provides a compact view of how the profile of known offenders varies across the top affected states.

6.7.3 Interactive Elements

- **Top N States Slider (`id="rel-top-n"`)**: Allows the user to select how many states (ranked by the total count of cases with known offenders) should be included in the treemap visualization.

6.7.4 Underlying Data: Offender-Victim Relationship (`df_rel_long`)

Data comes from `21_Offenders_known_to_the_victim.csv`. The script preprocesses this by melting the wide-format data into a long format (`df_rel_long`) with columns `Area_Name` (State), `Relationship`, and `Count`. Relationship labels are cleaned for readability
(e.g., "No_of_Cases_in_which_offenders_were_Relatives" becomes "Relatives").

6.7.5 Underlying Callback (`update_relative_treemap`)

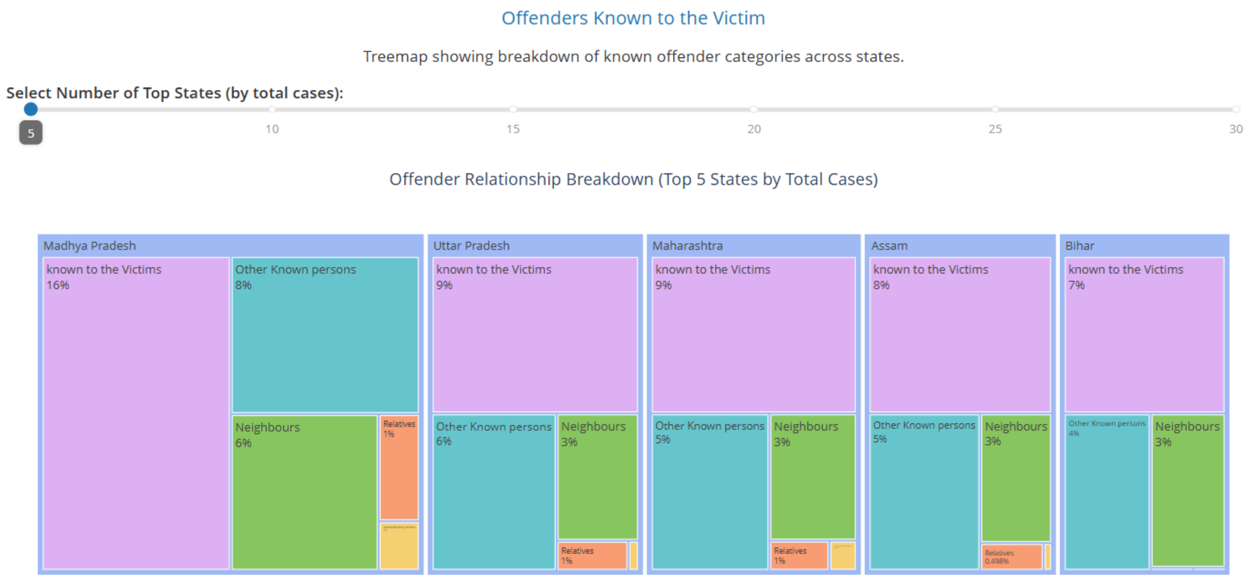
Triggered by the `rel-top-n` slider. It calls the helper function `build_offender_treemap`, passing the selected value of N. The helper function filters `df_rel_long` to include only the top N states (based on total count), ensures 'Count' is numeric, and generates the Plotly Treemap figure using `px.treemap` with the path `['Area_Name', 'Relationship']`.

6.7.6 The Story: Exploring the Relationship Between Victims and Known Offenders

- The relationship between victim and offender is a crucial factor in understanding crime dynamics, particularly for violent crimes often included in such datasets. This tab addresses the question: **"In cases where the offender was known to the victim, what types of relationships (family, neighbor, other) are most common, and how does this vary across the states with the most such cases?"**

- This visualization can be insightful for social workers, domestic violence prevention groups, or law enforcement focusing on interpersonal crime.
- The treemap quickly reveals if, for example, relatives constitute the largest group of known offenders in most top states, or if neighbors are more prominent in certain regions.
- A large rectangle for "Other Known Persons" might prompt further investigation into the specific nature of those relationships if more granular data were available. It helps quantify and visualize the often-hidden aspect of crime occurring within existing social networks, highlighting potential areas for targeted intervention or awareness campaigns.

6.7.7 Dash-board Screenshots



Offenders Known to the Victim

Treemap showing breakdown of known offender categories across states.

Select Number of Top States (by total cases): 5

Offender Relationship Breakdown (Top 5 States by Total Cases)

Figure 6.15: Offender Relationships Tab showing the Treemap for a selected number of top states

6.7.8 Our Key Observations from Sample Plots

- Madhya Pradesh stands out with 16% of offenders being "Known to Victims" - a notably higher percentage than other states which range from 7-9%, suggesting regional differences in crime relationship dynamics.
- The "Neighbours" category (green boxes) shows significant variation, with Madhya Pradesh having the highest percentage (9%) compared to just 3% in Uttar Pradesh and Bihar, pointing to different community dynamics across states.
- Madhya Pradesh shows a uniquely large teal section (8%) for "Other Known Persons" compared to smaller percentages in other states, suggesting different social relationship factors in crimes there.

6.8 Crime Clusters (IPC) Tab

6.8.1 Functionality Overview

This sophisticated tab moves beyond simple aggregation and applies machine learning (K-Means clustering) to group districts based on their IPC crime profiles. It aims to uncover underlying similarities and differences in crime patterns across districts that might not be apparent from simple geographic proximity or total counts. It also includes time series forecasting for the identified clusters.

6.8.2 Core Visualizations: Cluster Map, Centroid Bar Chart, Silhouette Plot, Trend/Forecast Lines

This tab presents a suite of interconnected visualizations:

- **Cluster Map (`id="cluster-map"`)**: A choropleth map showing districts colored according to their assigned cluster membership based on the K-Means algorithm run on selected IPC crime features (averaged over all years). Allows visual identification of geographic concentrations of specific cluster types. Users can toggle the visibility of clusters using the checklist. Hovering shows the district, its assigned cluster, and the average value of the selected features used for clustering.

- **Centroid Bar Chart (`id="cluster-centroid-bar"`)**: A grouped bar chart comparing the average values (centroids) of the selected IPC crime features for each cluster. This chart shows the *profile* of each cluster – e.g., Cluster 0 might be high in theft but low in murder, while Cluster 1 might be the opposite. Values are shown in the original data scale (after inverse transformation from scaled data used in clustering).

- **Silhouette Plot (`id="cluster-centroid-silhouette"`)**: A diagnostic plot used to help evaluate the quality of the clustering. It visualizes how similar each district is to its own cluster compared to other clusters. Bars represent individual districts, grouped by cluster. Wider, more uniform bars indicate better clustering. A vertical red line shows the average silhouette score across all districts. Scores range from -1 to 1, with higher values indicating better-defined clusters.

- **Cluster Trend/Forecast Lines (`id="cluster-trends-total"`, `id="cluster-trends-avg"`)**: Two separate line charts (potentially faceted) showing the time series trends for each cluster:
  - **Total IPC Trend:** Shows the average `TOTAL_IPC_CRIMES` per district for each cluster over the years.
  - **Average Selected Feature Trend:** Shows the average value of the *features selected for clustering* per district for each cluster over the years.
  - **Forecast:** Both trend charts include a one-year forecast generated using an ARIMA(1,1,1) model for each cluster's trend line, shown as a dashed line extension.

6.8.3 Interactive Elements

- **Feature Selection Dropdown (`id="cluster-features"`)**: A multi-select dropdown allowing the user to choose which IPC crime types (columns from `df_ipc`) will be used as features for the K-Means clustering algorithm. The data for these features is averaged across all years for each district before clustering.

- **Cluster Count Slider (`id="cluster-count"`)**: Allows the user to specify the number of clusters (K) for the K-Means algorithm (typically ranging from 2 to 10). Changing K reruns the clustering and updates all related visualizations.

- **Cluster Visibility Checklist (`id="cluster-visibility-checklist"`)**: Dynamically populated checkboxes, one for each generated cluster. Allows the user to selectively hide or show clusters on the map (`cluster-map`) for focused analysis. Unchecking a cluster typically renders districts belonging to it in a neutral color (e.g., white) on the map.

6.8.4 Machine Learning: K-Means Clustering, ARIMA Forecasting

- **Data Prep for Clustering:** The code selects the IPC dataset (`df_ipc`), filters out 'TOTAL' districts, calculates the mean value for each selected feature (`cluster-features`) across all years for each district, handles potential missing values (`fillna(0)`), and scales the features using `StandardScaler` to ensure all features contribute equally to the distance calculations in K-Means. Districts with no variance might be removed.

- **K-Means (KMeans)**: The KMeans algorithm from Scikit-learn is applied to the scaled, aggregated district data with the specified number of clusters (n_clusters) and standard parameters (init='k-means++', n_init=10). It assigns each district to a cluster.

- **ARIMA (ARIMA)**: For the trend plots, an ARIMA(1,1,1) model from statsmodels.tsa.arima.model is fitted to the historical time series (average crime count per year) for each cluster. A one-step forecast is generated using fit.forecast(steps=1). Error handling is included for cases where ARIMA fitting fails (e.g., insufficient data, constant series).

6.8.5 Underlying Callback and Helpers

- update_clusters: The main callback triggered by the feature dropdown, cluster slider, or visibility checklist. It performs data preparation, runs K-Means, generates cluster assignments (labels), calculates silhouette scores, determines checklist options/values, and then calls helper functions to create the figures for the map (cluster-map.figure), centroid bar chart (cluster-centroid-bar.figure), silhouette plot (cluster-centroid-silhouette.figure), and the two trend/forecast plots (cluster-trends-total.figure, cluster-trends-avg.figure). It uses dash.no_update for plots not needing regeneration if only the visibility checklist changed.

- create_choropleth_map: Helper function to generate the cluster map figure.

- create_centroid_bar_chart: Helper function to generate the centroid profile bar chart.

- create_silhouette_plot: Helper function to generate the silhouette plot figure.

- create_trend_forecast_plot: Helper function (used twice) to generate the historical trend and ARIMA forecast line charts for each cluster, adaptable for total crime or average of selected features.

6.8.6 The Story: Uncovering Hidden Patterns and Similarities in District Crime Profiles

- Simple geographic maps show crime volume, but they don't reveal if districts with similar crime *levels* also have similar crime *mixes*.

- This tab employs unsupervised machine learning to address a deeper question: **"Based on the prevalence of various IPC crimes, can we identify groups of districts that share similar crime profiles, irrespective of their absolute crime volume or location, and how do these groups trend over time?"**

- Imagine discovering through clustering that certain districts, perhaps geographically distant, form a distinct cluster characterized by high rates of property crime but low violent crime, while another cluster shows the reverse profile.

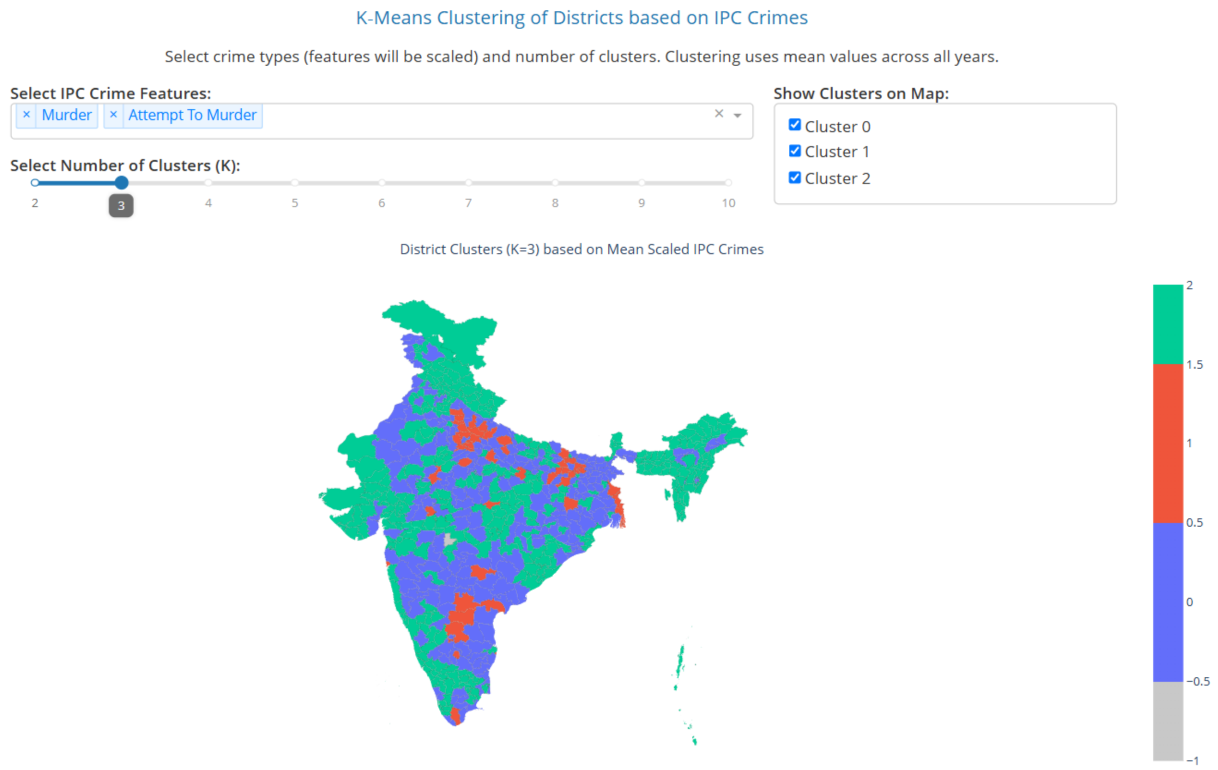
- This insight, derived from the centroid bar chart, is powerful. It suggests potentially similar underlying socio-economic factors or policing challenges driving these specific crime patterns within the districts of a cluster.

- The cluster map then shows *where* these different profiles manifest geographically. The silhouette plot provides a measure of confidence in these groupings.

- Finally, the trend charts with forecasts allow analysts to see if these distinct crime profiles are worsening or improving over time within each cluster type, offering a potential glimpse into future challenges or successes related to specific crime patterns.

- This tab moves beyond descriptive statistics into pattern discovery and exploratory forecasting.

6.8.7 Dash-board Screenshots



K-Means Clustering of Districts based on IPC Crimes

Select crime types (features will be scaled) and number of clusters. Clustering uses mean values across all years.

Select IPC Crime Features: × Murder × Attempt To Murder

Select Number of Clusters (K): 3

Show Clusters on Map:

☑ Cluster 0
☑ Cluster 1
☑ Cluster 2

Figure 6.16: Crime Clusters Tab showing the Cluster Map, Centroid Bar Chart, and Silhouette Plot for a given set of features and K

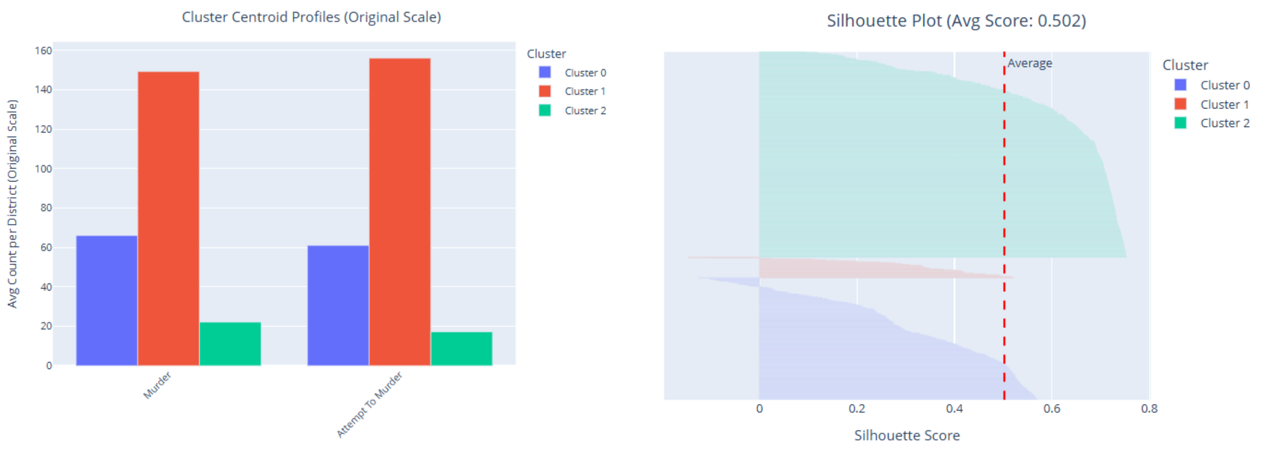


Figure 6.17: Crime Clusters Tab showing the Trend/Forecast plot for Total IPC Crimes by cluster

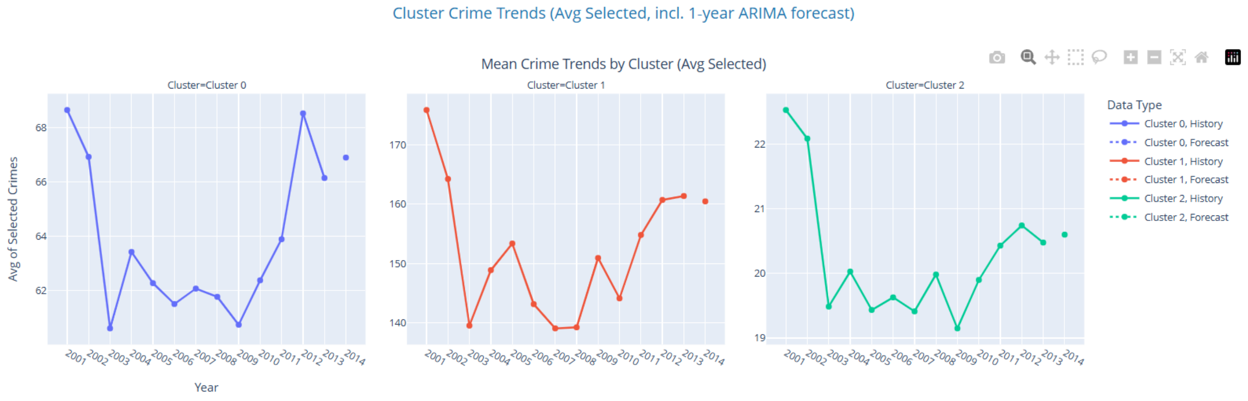


Figure 6.18: Crime Clusters Tab showing the Trend/Forecast plot for the Average of Selected Features by cluster

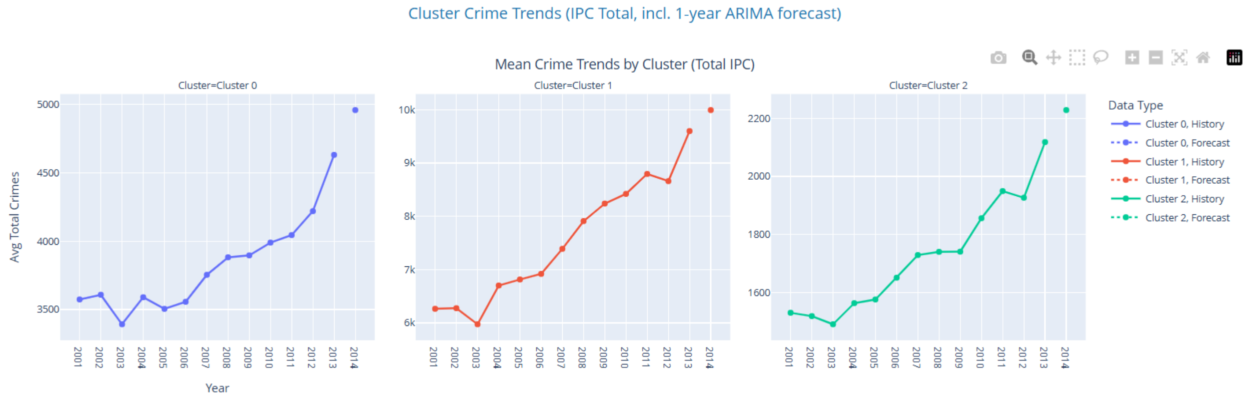


Figure 6.19: Crime Clusters Tab showing the Cluster Visibility Checklist and the map with some clusters hidden

6.8.8 Our Key Observations from Sample Plots

- The K-means clustering (with K=3) has grouped districts into three clusters with markedly different crime patterns, showing how similar crime mixes can appear in geographically separated areas.

- The red areas on the map (Cluster 1) show districts with extremely high murder and attempted murder rates, with centroid values around 150 cases - more than twice the values in Cluster 0 (blue) and seven times higher than Cluster 2 (green).

- The cluster map reveals that high-violence districts (red) appear scattered across central and eastern India rather than concentrated in one region, contradicting assumptions that crime patterns follow strict geographic boundaries.

- The silhouette plot shows an average score of 0.502, suggesting the three clusters are reasonably distinct, though not perfectly separated. Most districts fall above the average line, indicating generally good cluster assignment.

- The trend show all three clusters experiencing growth in total IPC crimes, but at dramatically different rates in Cluster 1 (highest violence) shows the steepest projected increase, suggesting a widening gap between district types.

- Cluster 2 (green) shows the most stable trend line for murder cases, staying consistently around 20 cases with minor fluctuations, indicating these districts maintain relatively steady low violence levels despite changes elsewhere.

6.9 Custodial Deaths Plots Tab

6.9.1 Functionality Overview

This tab focuses on the sensitive issue of deaths occurring in police or judicial custody. It allows users to visualize trends in different causes of custodial deaths, either nationally or by comparing selected states.

6.9.2 Core Visualizations: Stacked Area/Bar Charts (Small Multiples or National)

The primary visualization (`id="timeline-graph"`) dynamically changes based on user selections:

- **Chart Type (Area/Bar):** Users can choose between:
  - **Stacked Area Chart:** Shows the trend of total custodial deaths over time, with different colored areas representing the contribution of each cause (e.g., Suicide, Illness, Accidents). Useful for seeing the total trend and the changing proportions of causes.
  - **Stacked Bar Chart:** Shows the total deaths per year as stacked bars, where segments within each bar represent different causes. Also shows total and proportions but may be clearer for discrete year comparisons.
- **Mode (National/States):** Users can choose:
  - **National Aggregate (`mode='nat'`):** Displays a single chart (area or bar) showing the total counts for each cause summed across all states/UTs for each year. Uses the `nat_melt` DataFrame.
  - **Small Multiples (`mode='states'`):** Displays multiple charts (area or bar), one for each selected state (up to 6). Each chart shows the trend/breakdown of causes *within* that specific state. Uses the `melted` DataFrame filtered by `selected_states`. Faceting (`facet_col='Area_Name'`) arranges these charts for comparison.

6.9.3 Interactive Elements

- **Chart Type Radio (`id="chart-type"`):** Radio buttons to select 'Stacked Area' or 'Stacked Bar'.
- **Mode Radio (`id="mode"`):** Radio buttons to select 'Few States (small multiples)' or 'National aggregate'.
- **State Multi-Select Dropdown (`id="state-select"`):** A dropdown allowing the user to select up to 6 states for comparison when 'Few States' mode is active. A callback (`disable_extra_states`) dynamically disables options once 6 states are selected to enforce the limit.

6.9.4 Underlying Data: Custodial Deaths (`df_cust`, `melted`, `nat_melt`)

The data originates from `40_05_Custodial_death_others.csv`. The script preprocesses it by:

- Replacing 'NULL' with NaN and converting relevant columns to numeric.
- Melting the multiple cause columns (e.g., `CD_Accidents`, `CD_By_Suicide`) into a long format (`melted`) with `Area_Name`, `Year`, `Cause`, and `Count` columns.
- Creating a national aggregate by grouping the original `df_cust` by `Year`, summing the cause columns, and then melting this aggregated data into `nat_melt`.

6.9.5 Underlying Callbacks

- `disable_extra_states`: Triggered by changes in the `state-select` value. If 6 states are selected, it adds the `disabled=True` attribute to all other state options in the dropdown.

- `update_chart`: Triggered by changes in chart type, mode, or selected states. It selects the appropriate DataFrame (`melted` or `nat_melt`) based on the mode, filters by selected states if necessary, and generates the corresponding Plotly figure (either `px.area` or `px.bar`, with or without faceting) for `timeline-graph.figure`.

6.9.6 The Story: Analyzing Trends and Causes of Deaths in Custody

- Deaths in custody are a matter of significant public concern and scrutiny, raising questions about conditions, procedures, and potential foul play.

- This tab provides a tool to ask: **"What are the reported causes of custodial deaths, how have their numbers trended over time nationally, and how do these trends and causes compare across different states?"**

- Human rights organizations, oversight committees, or researchers could use this tab to track national trends in custodial suicides versus deaths attributed to illness or accidents.

- Selecting the 'Few States' mode allows for direct comparison: does State A report significantly more deaths due to 'Illness/Natural Death' compared to State B, which might report more 'Suicides'?

- Visualizing this data, particularly comparing states side-by-side, can highlight potential discrepancies or areas requiring investigation regarding custodial conditions and protocols.

- It helps move the discussion from individual incidents to broader, data-informed patterns of reported causes over time and geography.

6.9.7 Dash-board Screenshots

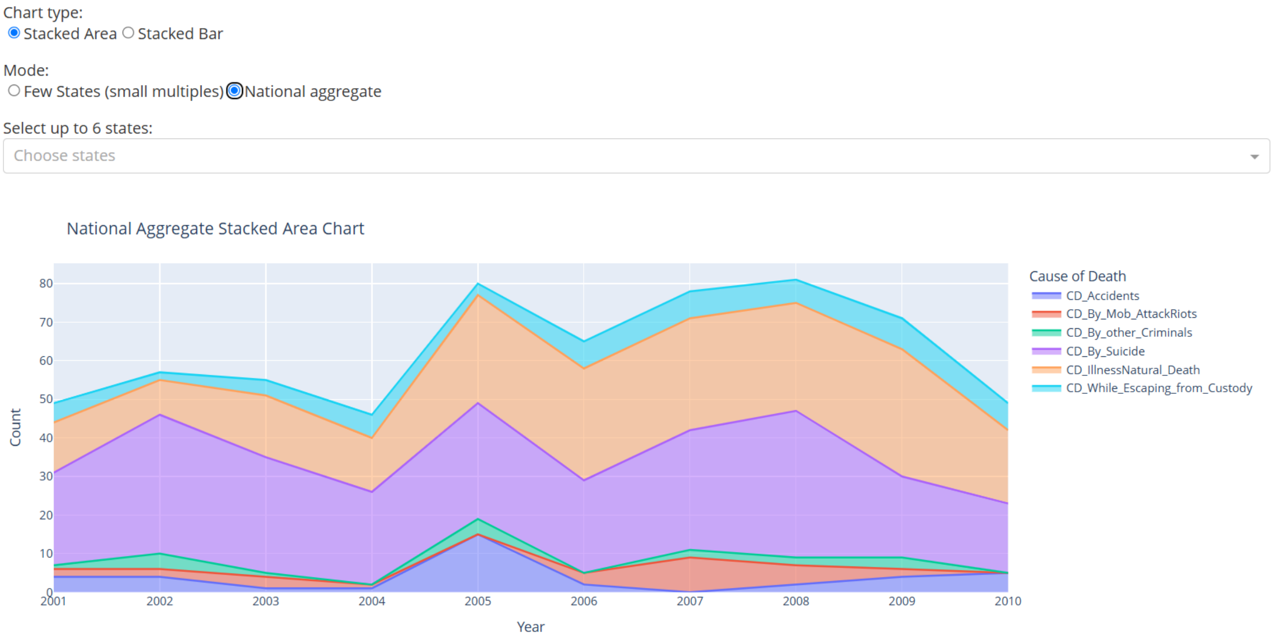


Chart type:
● Stacked Area ○ Stacked Bar

Mode:
○ Few States (small multiples) ● National aggregate

Select up to 6 states:
Choose states

Figure 6.20: Custodial Deaths Tab showing National Aggregate view (e.g., Stacked Area Chart)

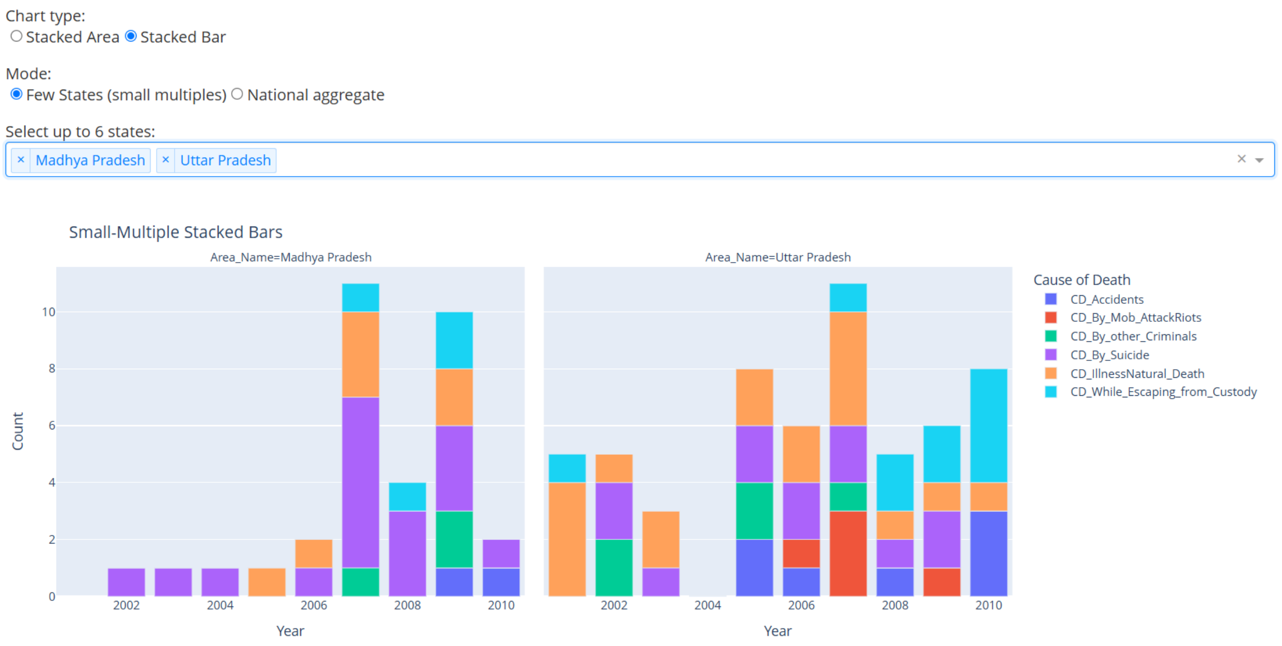


Chart type:
○ Stacked Area ● Stacked Bar

Mode:
● Few States (small multiples) ○ National aggregate

Select up to 6 states:
× Madhya Pradesh × Uttar Pradesh

Figure 6.21: Custodial Deaths Tab showing Small Multiples view comparing several selected states (e.g., Stacked Bar Charts)

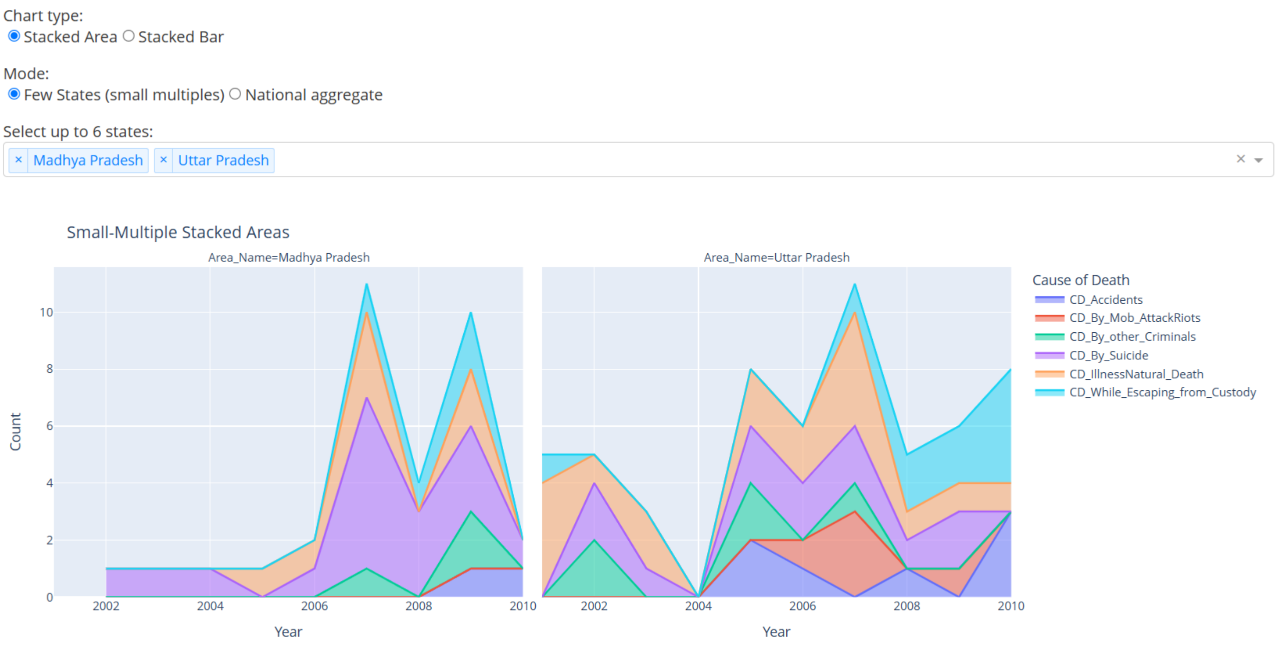


Chart type:
● Stacked Area ○ Stacked Bar

Mode:
● Few States (small multiples) ○ National aggregate

Select up to 6 states:
× Madhya Pradesh × Uttar Pradesh

Figure 6.22: Custodial Deaths Tab showing the State Selection dropdown with some options disabled after selecting 6 states

6.9.8 Our Key Observations from Sample Plots

- The stacked area chart shows "Illness/Natural Death" (orange area) consistently forms the largest portion of custodial deaths across 2001-2010, suggesting healthcare in custody may deserve significant attention from oversight committees.
- The purple area representing "Suicide" maintains a substantial presence throughout the decade, forming the second largest cause category nationally, pointing to mental health and supervision concerns in detention facilities.

- Total custodial deaths show dramatic variation between years, with peaks around 2005 (approximately 80 deaths) and 2007-2008, followed by notable declines, suggesting improvements.
- The MP-UP comparison shows Madhya Pradesh experienced major spikes in 2007 and 2009 while Uttar Pradesh shows a different pattern with a major peak in 2008.
- Madhya Pradesh shows higher proportions of suicides in custody, while Uttar Pradesh reports more deaths from accidents and while escaping.
- The light blue segments representing "While Escaping from Custody" appear more prominent in later years, have a look into Uttar Pradesh after 2008.

6.10 Juvenile Plots Tab

6.10.1 Functionality Overview

This tab provides insights into the backgrounds of juveniles arrested, based on datasets covering their education, economic status, family background, and recidivism. It visualizes the distribution across different categories within these domains.

6.10.2 Core Visualizations: Pie Charts

The tab dynamically generates multiple pie charts within the `juv-plots-container`, one for each category selected by the user.

- **Data Display:** Each pie chart shows the proportional breakdown of arrested juveniles based on the selected category's sub-groups (e.g., for 'Education', slices might represent 'Illiterate', 'Primary', 'Middle', 'Matric', etc.). The size of each slice corresponds to the total count aggregated across all states and years present in the respective dataset.
- **Clarity:** Labels showing the percentage and category name are displayed, potentially outside the slices for better readability, and hover tooltips provide exact counts.

6.10.3 Interactive Elements

- **Category Multi-Select Dropdown (`id="juv-category-dropdown"`):** Allows the user to select one or more background categories (Education, Economic Setup, Family Background, Recidivism) to visualize. For each selected category, a corresponding pie chart is generated.

6.10.4 Underlying Data: Juvenile Arrest Demographics

Four separate datasets are used: `18_01_Juveniles_arrested_Education.csv`, `18_02_Juveniles_arrested_Economic_setup.csv`, `18_03_Juveniles_arrested_Family_background.csv`, `18_04_Juveniles_arrested_Recidivism.csv`. Each is loaded and cleaned using `load_and_clean_csv1`. The callback then processes the relevant DataFrame based on the user's selection.

6.10.5 Underlying Callback (`update_juv_plots`)

Triggered by changes in the `juv-category-dropdown`. For each selected category:

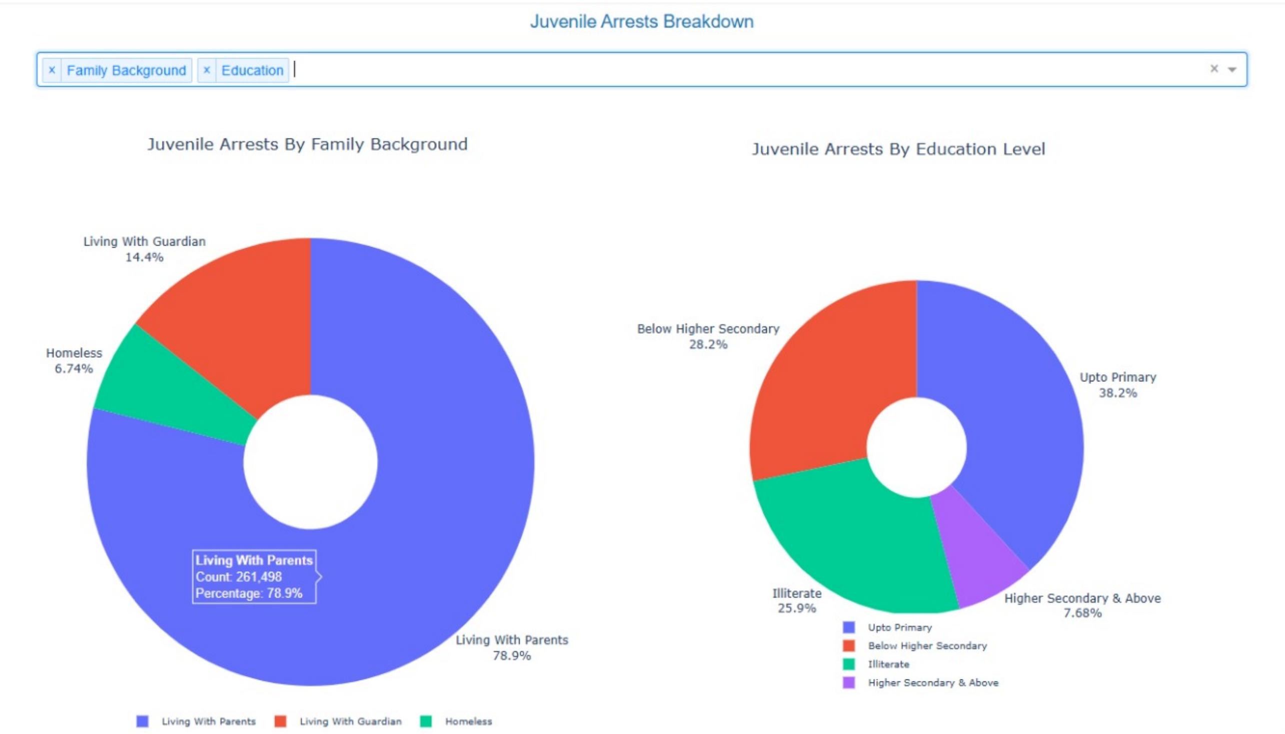
- It identifies the corresponding DataFrame and the prefix for relevant columns (e.g., `df_juv_edu` and `"EDUCATION_"`).
- It sums the values across all rows for columns matching the prefix (excluding totals).
- It cleans the column names to create meaningful labels (e.g., `EDUCATION_ILLITERATE` becomes "Illiterate").

- It generates a `go.Pie` figure for that category.
- It appends the generated pie chart (wrapped in a `html.Div` for layout) to a list of figures displayed in `juv-plots-container.children`.

6.10.6 The Story: Understanding the Backgrounds of Arrested Juveniles

- Juvenile delinquency is often linked to socio-economic and environmental factors. This tab helps explore these connections by asking: **"What are the aggregated profiles of arrested juveniles in terms of their education levels, economic backgrounds, family situations, and history of re-offending?"**
- Social researchers, policymakers working on juvenile justice reform, or NGOs focused on youth development could use this tab.
- The pie charts provide a quick visual summary of the national picture (aggregated across the dataset's scope).
- For instance, seeing a large slice for 'Illiterate' or 'Primary' education might highlight a link between lack of education and juvenile arrests. Similarly, observing the proportions related to family background (e.g., 'Living with Parents' vs. 'Homeless') or economic setup (e.g., income brackets) can inform targeted intervention programs for at-risk youth.
- The recidivism chart sheds light on the extent to which arrested juveniles are repeat offenders. While aggregated, these visualizations offer valuable context about the circumstances surrounding juvenile arrests.

6.10.7 Dash-board Screenshots



Juvenile Arrests Breakdown

Figure 6.23: Juvenile Plots Tab showing pie charts for selected categories like Education and Family Background

6.10.8 Our Key Observations from Sample Plots

- The left pie chart shows that, 78.9% of arrested juveniles (263,498 cases) were actually living with parents at the time of arrest, suggesting family presence alone doesn't prevent juvenile offending.
- About 14.4% of arrested juveniles lived with guardians rather than parents.
- Though only 6.74% of arrested juveniles were homeless.
- The right chart reveals that 64.1% of arrested juveniles had primary education or less (38.2% primary and 25.9% illiterate), showing a strong connection between limited education and criminal activities involvement.
- Only 7.68% of arrested juveniles had higher secondary education or above, suggesting that staying in school longer significantly reduces risk of juvenile's involvement.

6.11 Unidentified Bodies Heatmap Tab

6.11.1 Functionality Overview

This tab presents a static heatmap visualizing the number of unidentified dead bodies recovered and inquests conducted across different states/UTs over the years 2001-2010.

6.11.2 Core Visualization: Heatmap

The single visualization is a heatmap (`id="heatmap"`) generated using `px.imshow`.

- **Data Display:** The x-axis represents the 'Year' (2001-2010), and the y-axis represents the 'State/UT' (`Area_Name`). The color intensity of each cell corresponds to the number of `Unidentified_Dead_bodies_Recovered_Inquest_Conducted` for that specific state/UT and year. A color scale (e.g., "YlOrRd") indicates the magnitude, with darker/redder colors typically representing higher counts.
- **Ordering:** States/UTs on the y-axis are ordered by their total count across the years (`update_yaxes(categoryorder="total ascending")`), bringing areas with consistently higher numbers towards the top.

6.11.3 Interactive Elements

This tab is primarily static; there are no user controls to filter or change the data displayed in the heatmap. Users can hover over cells to see the exact count for a specific state/year.

6.11.4 Underlying Data: Unidentified Dead Bodies (`df_heat`, `heatmap_data`)

The data comes from `38_Unidentified_dead_bodies_recovered_and_inquest_conducted.csv`. It is loaded into `df_heat` and then pivoted using `df_heat.pivot` to create `heatmap_data`, a matrix where rows are `Area_Name`, columns are `Year`, and values are the counts of unidentified bodies/inquests. This matrix format is directly used by `px.imshow`.

6.11.5 The Story: Visualizing Temporal and Geographic Patterns of Unidentified Body Recoveries

- The recovery of unidentified bodies is a grim indicator potentially linked to various factors, including crime, accidents, migration, or issues in identification processes.
- This heatmap addresses the question: **"Where and when were the highest numbers of unidentified bodies recovered between 2001 and 2010?"**

- Law enforcement agencies tracking missing persons or researchers studying mortality patterns could use this visualization.
- The heatmap allows for rapid identification of:
  - **Spatial Hotspots:** States/UTs consistently showing darker colors across the years.
  - **Temporal Peaks:** Specific years where colors are generally darker across many states, potentially indicating a large-scale event or reporting change.
  - **Anomalies:** Specific cells (state/year combinations) with unusually high counts compared to their neighbors.
  - By visualizing the data in this matrix format, patterns that might be hidden in tables become immediately apparent, prompting further investigation into the reasons behind high counts in particular regions or years.

6.11.6 Dash-board Screenshots

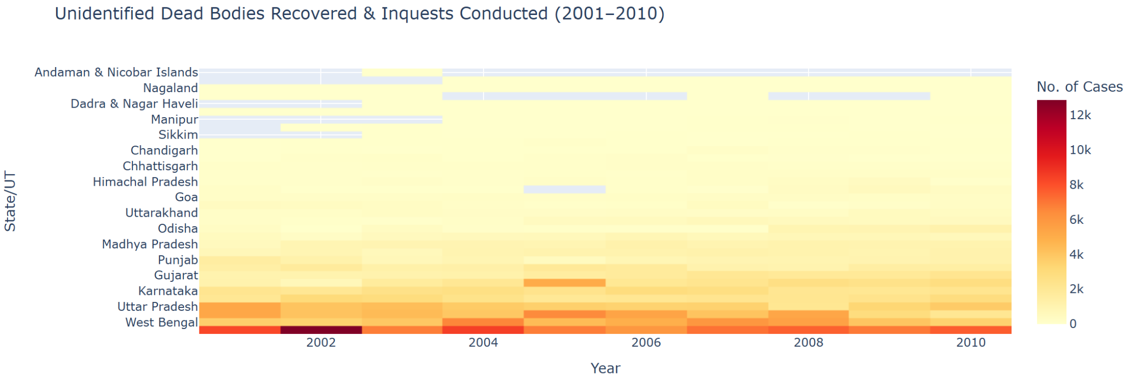


Figure 6.24: Unidentified Bodies Heatmap Tab showing the heatmap visualization

6.11.7 Our Key Observations from Sample Plots

- West Bengal (bottom row) shows consistently dark red coloring throughout the entire decade, indicating extraordinarily high numbers of unidentified bodies (appearing to exceed 10,000 cases in some years) compared to most other states.
- Uttar Pradesh consistently shows orange-red coloring, establishing it as having the second highest recovery rate of unidentified bodies across India.
- Northern and northeastern states (Sikkim, Manipur, Nagaland, and Andaman and Nicobar Islands) consistently show the lightest colors, indicating dramatically fewer unidentified body recoveries throughout the decade.
- States like Gujarat, Karnataka and Maharashtra display moderate yellow-orange coloring, placing them in a middle tier that's significantly higher than most states but well below the extreme cases of West Bengal and UP.

6.12 Serious Fraud Losses Tab

6.12.1 Functionality Overview

This tab focuses on serious financial fraud, visualizing the reported losses across different states/UTs, categorized by the magnitude of the loss (e.g., 1-25 Crores, 25-100 Crores, >100 Crores).

6.12.2 Core Visualization: Stacked Bar Chart

The main visualization is a horizontal stacked bar chart (`id="bar-chart"`).

- **Data Display:** Each bar represents a State/UT (`Area_Name`). The total length of the bar indicates the total number of serious fraud cases reported for that state (matching the selected filters). The bar is segmented by color, with each segment representing a different `Loss_Band` (e.g., '1–25 Cr', '25–100 Cr', '>100 Cr'). The length of each segment corresponds to the number of cases falling into that loss band.
- **Ordering:** States are typically ordered on the y-axis based on the total number of fraud cases (highest at the top) (`yaxis={'categoryorder':'array','categoryarray':order.tolist()}`).

6.12.3 Interactive Elements

- **Year Dropdown (`id="year-dropdown"`)**: Filters the data by year (with an 'All' option).
- **Group Dropdown (`id="group-dropdown"`)**: Filters the data by the type of fraud group (e.g., 'Serious Fraud', 'Cheating', etc., based on `Group_Name` column - with an 'All' option).
- **State Multi-Select Dropdown (`id="state-dropdown"`)**: Allows filtering for specific states/UTs (with an 'All' option implicitly when none are selected or if 'All' were an explicit option, though the code uses a default list).

6.12.4 Underlying Data: Serious Fraud (`df_long`)

Data comes from `31_Serious_fraud.csv`. It's preprocessed by melting the different loss columns (e.g., `Loss_of_Property_1_25_Crores`) into a long format (`df_long`) with columns `Area_Name`, `Year`, `Group_Name`, `Loss_Band` (cleaned labels), and `Loss` (representing the *count* of cases in that band, not the value). Rows with `Loss = NULL` are replced with zero.

6.12.5 Underlying Callback (`update_bar_chart`)

Triggered by changes in the year, group, or state dropdowns. It filters the `df_long` DataFrame based on the selections, aggregates the number of cases ('Loss' column) by `Area_Name` and `Loss_Band`, determines the state order based on total cases, and generates the `px.bar` stacked horizontal bar chart figure (`bar-chart.figure`).

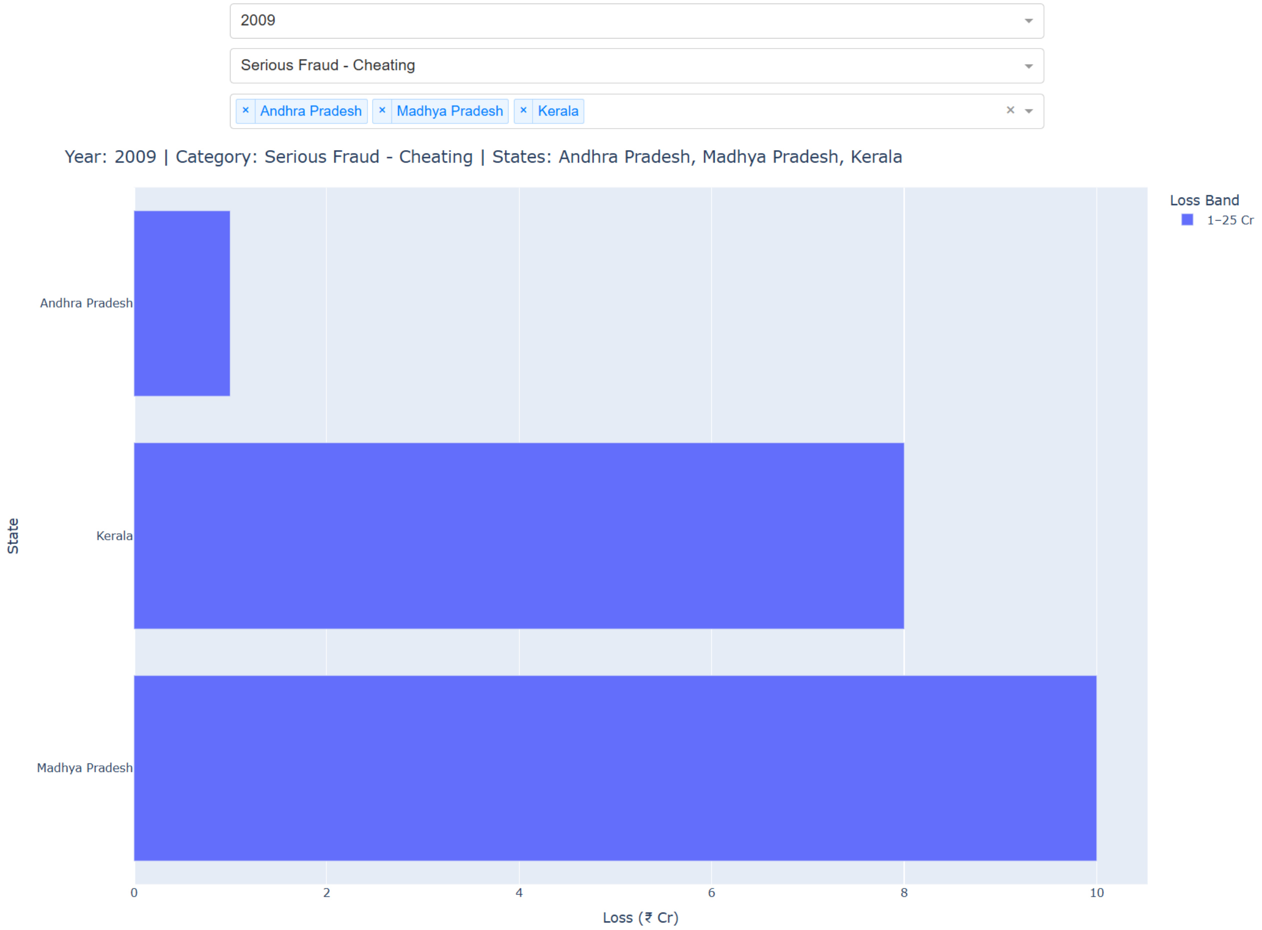
6.12.6 The Story: Assessing the Scale and Distribution of High-Value Fraud

Serious financial fraud can have significant economic impacts. This tab helps visualize:

- **"Where do most cases of serious fraud occur, what is the typical magnitude of these frauds (in terms of loss bands), and how does this vary by year or fraud type?"**
- The stacked bar chart helps police and researchers see which states have the most serious fraud cases.
- You can quickly spot which states have the highest number of fraud cases during specific time periods or for certain types of fraud.
- The chart breaks down fraud cases by money amounts: smaller cases (1-25 Crore), medium cases, and very large cases (more than 100 Crore).
- This breakdown reveals important patterns - some states might have many fraud cases but mostly smaller ones, while others might have fewer cases but they're mostly very large frauds
- Understanding these differences helps create better strategies to fight fraud in different regions.

- Users can filter the data by year or fraud type to see trends over time or focus on specific kinds of fraud.
- This graphs makes it easier to decide where to put resources and how to prevent future fraud based on where the biggest problems are.
- The visual format makes complex fraud data easier to understand and compare across different states.

6.12.7 Dash-board Screenshots



2009

Serious Fraud - Cheating

× Andhra Pradesh × Madhya Pradesh × Kerala

Year: 2009 | Category: Serious Fraud - Cheating | States: Andhra Pradesh, Madhya Pradesh, Kerala
Loss Band
■ 1–25 Cr
State
Andhra Pradesh
Kerala
Madhya Pradesh
0
2
4
6
8
10
Loss (₹ Cr)

Figure 6.25: Serious Fraud Losses Tab showing the stacked horizontal bar chart for selected filters

6.12.8 Our Key Observations from Sample Plots

- The horizontal bar chart shows dramatic differences in financial losses from serious fraud - cheating across the three selected states in 2009, with Madhya Pradesh facing nearly 10 crore rupees in losses compared to just 1-2 crore in Andhra Pradesh.
- All three states show fraud losses within the same band (1-25 crore rupees) as indicated by the blue color, suggesting that while the amounts differ, the magnitude category remains similar across regions.
- The dropdown selectors for year, fraud category, and states (with multiple state selection capability) allow for precise targeting of analysis, supporting the story's focus on exploring fraud patterns across different dimensions.
- The side-by-side display of different states using the same scale makes it easy to make direct comparisons, helping identify states that might require different intervention approaches.

6.13 Property Stolen Analysis Tab

6.13.1 Functionality Overview

This tab focuses on property crime, specifically comparing the value and number of cases of property stolen versus property recovered. It allows analysis for a single state or a comparison between multiple states.

6.13.2 Core Visualizations: Line Charts, Grouped Bar Charts

The visualizations (`id="stolen-plots-div"`) change based on the selected mode:

- **Single State Mode (`mode='single'`)**:
  - **Line Chart (Value):** Shows two lines over the selected year range for the chosen state: 'Value of Property Stolen' and 'Value of Property Recovered'. Allows comparison of value trends.
  - **Line Chart (Cases):** Shows two lines over the selected year range for the chosen state: 'Cases Property Stolen' and 'Cases Property Recovered'. Allows comparison of case volume trends.
- **Compare States Mode (`mode='multi'`)**:
  - **Grouped Bar Chart (Cases):** Compares the total 'Cases Property Stolen' and 'Cases Property Recovered' (aggregated over the selected years) across the chosen states. Each state has two bars (stolen/recovered) side-by-side.
  - **Grouped Bar Chart (Value):** Compares the total 'Value of Property Stolen' and 'Value of Property Recovered' (aggregated over the selected years) across the chosen states. Similar grouping as the cases chart.

6.13.3 Interactive Elements

- **Year Range Slider (`id="stolen-year-slider"`)**: Filters the data by year.
- **Mode Radio (`id="stolen-mode"`)**: Selects between 'Single State' and 'Compare States' analysis modes. This controls which set of plots is displayed and which state dropdown is active.
- **Single State Dropdown (`id="stolen-single-dd"`)**: Appears only in 'Single State' mode. Allows selecting the state for trend analysis.
- **Multi-State Dropdown (`id="stolen-multi-dd"`)**: Appears only in 'Compare States' mode. Allows selecting 2-7 states for comparison.

6.13.4 Underlying Data: Property Stolen/Recovered (`df_stolen`)

Data is loaded from `property_stolen.csv` into `df_stolen`. Basic type conversion for 'Year' is performed. The callbacks then filter and aggregate this DataFrame based on user selections.

6.13.5 Underlying Callbacks

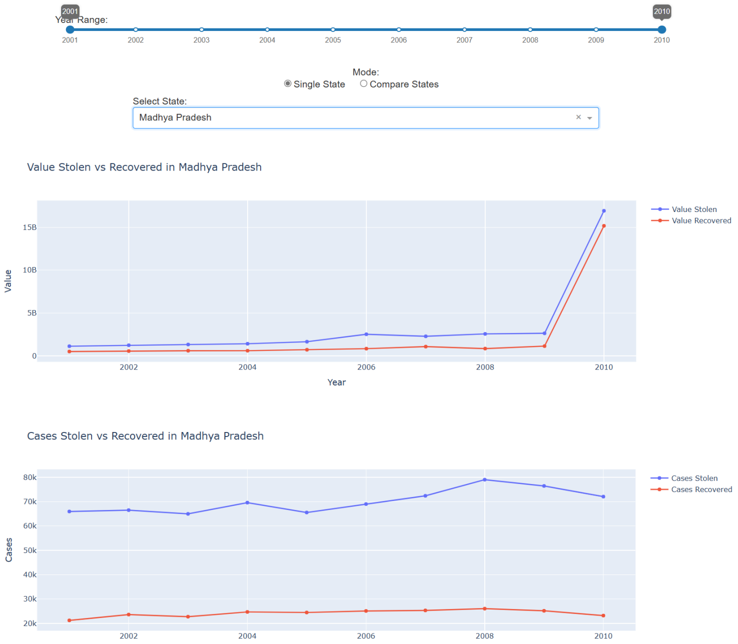
- `toggle_stolen`: Triggered by the `stolen-mode` radio button. Shows/hides the single or multi-state dropdown container based on the selected mode.
- `update_stolen`: Triggered by changes in the year slider or either of the state dropdowns (depending on the mode). It filters `df_stolen` by year and selected state(s), performs aggregations (yearly for single mode, total for multi mode), and generates the appropriate Plotly line or bar charts (`go.Scatter`, `go.Bar`) displayed in `stolen-plots-div.children`.

6.13.6 The Story: Comparing Property Theft Value/Cases and Recovery Rates

This tab helps answer: **"For a given state, how have the value and number of property theft cases trended compared to recoveries? How do different states compare in their total stolen/recovered values and case numbers over a period?"**

- Property crime affects many citizens, and these graphs helps track how well police recover stolen items.
- The tab answers important questions about theft trends and recovery performance in different states.
- In 'Single State' mode, users can see if a state is getting better or worse at recovering stolen property over time.
- The tool shows if the gap between stolen and recovered property is growing smaller (improvement) or larger (worsening).
- In 'Compare States' mode, users can directly compare several states against each other.
- Users can easily spot which states have the highest total value of stolen property.
- The tool reveals which states have better recovery rates by comparing the height of the 'Recovered' bar to the 'Stolen' bar.
- This comparison helps identify which states have more effective property crime investigation methods.
- Law enforcement agencies can use this data to learn from better-performing states.
- The visual format makes it easy to spot trends and differences that might not be obvious in raw data.

6.13.7 Dash-board Screenshots

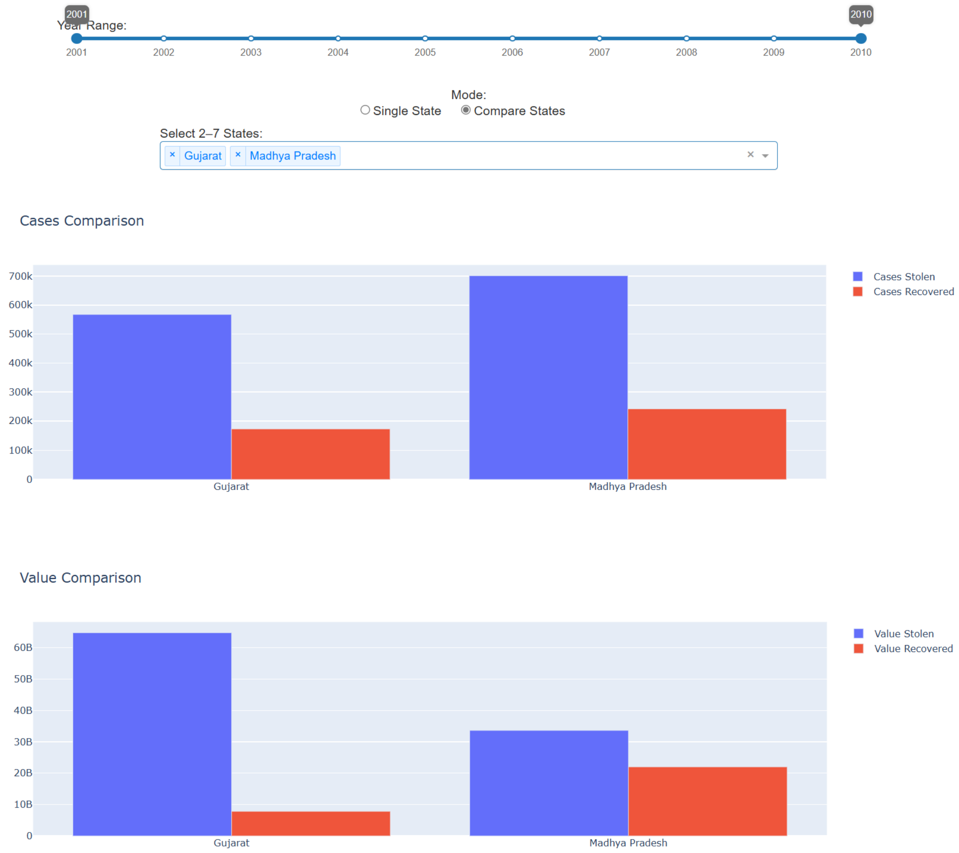


Year Range: 2001 – 2010

Mode: ● Single State ○ Compare States

Select State: Madhya Pradesh

Figure 6.26: Property Stolen Tab showing Single State mode with Value and Cases line charts



Year Range: 2001 – 2010

Mode:
○ Single State ● Compare States

Select 2–7 States:
× Gujarat × Madhya Pradesh

Figure 6.27: Property Stolen Tab showing Compare States mode with grouped bar charts for Cases and Value

6.13.8 Our Key Observations from Sample Plots

- Madhya Pradesh shows a startling jump in both stolen and recovered property values in 2010, increasing from about 30 units to nearly 175 units.
- Throughout 2001-2010, Madhya Pradesh consistently recovers only about one-third of stolen property cases (red line hovering around 20-25K compared to blue line at 60-80K), showing challenges faced in solving property crimes.
- Gujarat has fewer total theft cases than Madhya Pradesh (550K vs 700K) but significantly higher total stolen value (620 units vs 350 units), indicating Gujarat experiences fewer but much higher-value thefts.
- The graph's ability to toggle between single state trends and direct state comparisons perfectly supports the story's purpose of analyzing both temporal patterns and geographic differences in property crime recovery performance.

6.14 Rape Victims Trend & Prediction Tab

6.14.1 Functionality Overview

This tab focuses specifically on reported rape cases, allowing users to explore historical trends for different states and victim subgroups, and providing a simple one-year prediction based on linear regression.

6.14.2 Core Visualization: Time Series Line Chart with Prediction Point

The main visualization is a line chart (`id="test-time-series"`) displaying the number of rape victims over time.

- **Historical Data:** Shows lines representing the number of `Victims_of_Rape_Total` for each selected state/subgroup combination over the historical years chosen by the slider. Multiple lines can be overlaid for comparison.

- **Prediction:** If the selected year range includes the year *after* the maximum year in the dataset, a distinct marker (diamond symbol) is plotted representing the predicted victim count for that future year. The prediction is generated using a separate `LinearRegression` model trained on the historical data for that specific state/subgroup combination.

6.14.3 Interactive Elements

- **State Multi-Select Dropdown (`id="test-state-dd"`):** Allows selecting one or more states, including an 'All States' option for the national aggregate.

- **Subgroup Multi-Select Dropdown (`id="test-sub-dd"`):** Allows selecting one or more victim subgroups (likely age groups based on typical NCRB data), including an 'All Subgroups' option.

- **Year Range Slider (`id="test-year-slider"`):** Selects the time period for displaying historical trends. The slider includes the prediction year as its maximum value. Marks indicate historical vs. predicted years.

6.14.4 Underlying Data: Rape Victims (`df_rape`, `pred_df`)

- `df_rape`: Loaded from `20_Victims_of_rape.csv`, containing historical data by state, year, and subgroup.

- `pred_df`: Generated during the initial script execution. For each state/subgroup combination (and the 'All' aggregates), a `LinearRegression` model is trained on the historical `Victims_of_Rape_Total` vs. `Year`. This model is used to predict the value for the `next_year` (max_year + 1), and these predictions are stored in `pred_df`.

6.14.5 Underlying Callback (`update_test`)

Triggered by changes in the state dropdown, subgroup dropdown, or year slider. It filters `df_rape` for historical data based on selections and filters `pred_df` for relevant predictions. It then constructs the `go.Scatter` traces for the historical lines and the prediction markers, combining them into a single figure (`test-time-series.figure`). It handles the logic for aggregating data when 'All States' or 'All Subgroups' is selected.

6.14.6 The Story: Examining Historical Trends and Predicting Future Rape Victim Counts

Understanding trends in reported rape cases is crucial for resource allocation, victim support services, and prevention strategies. This tab addresses: **"How have reported rape victim counts changed over time for specific states or victim subgroups, and what is a simple projection for the upcoming year based on historical linear trends?"**

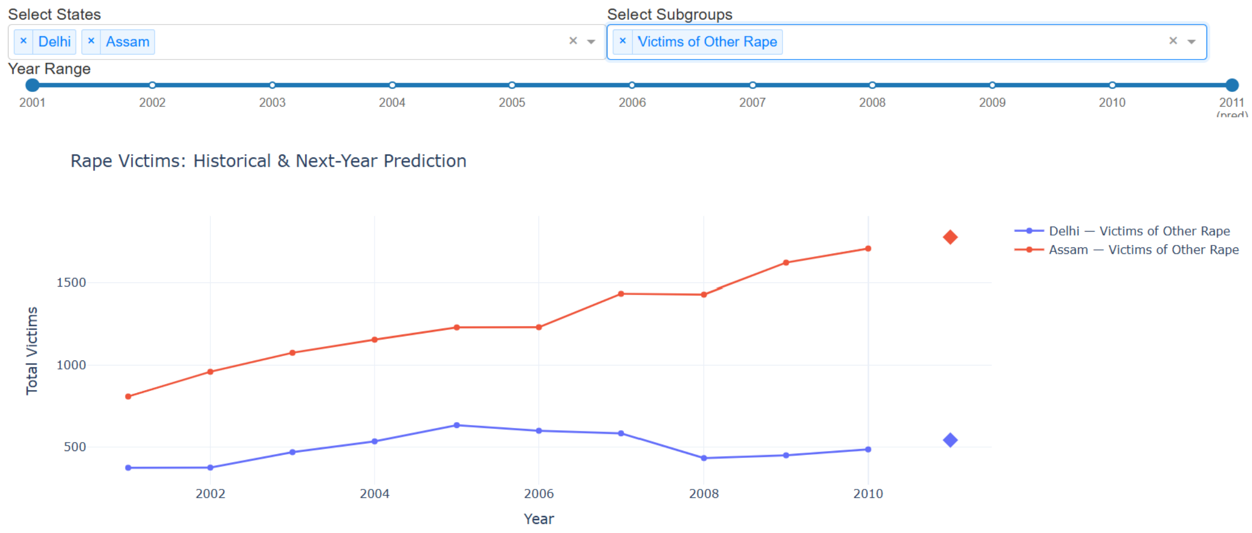
- This tab helps understand how rape case numbers have changed over time, which is important for planning police resources and victim support services.

- The graphs answers questions about how reported rape victim counts have changed across different states or victim groups, and provides a simple forecast for the upcoming year.

- Users can compare trends between different states to see where cases are increasing, decreasing, or remaining stable.

- The graph also allows users to examine if certain age groups show different patterns in reported cases across the country.
- A basic prediction point shows what next year's numbers might be if current trends continue in a straight line.
- While simple, this prediction gives officials a starting point for planning resources and preparing for future cases.
- The visualization combines historical data with a forward-looking estimate to support better planning.
- Police departments and support organizations can use this information to prepare for expected changes in case numbers.
- These graphs also helps in identify states or groups that may need additional attention or resources based on their trends.
- The visual format makes it easier to spot important patterns in this sensitive crime category.

6.14.7 Dash-board Screenshots



Select States: Delhi, Assam

Select Subgroups: Victims of Other Rape

Year Range

Figure 6.28: Rape Victims Trend Tab showing historical lines for selected states/subgroups and the diamond prediction marker

6.14.8 Our Key Observations from Sample Plots

- Assam consistently reports much higher numbers of "Victims of Other Rape" than Delhi throughout the decade, with Assam showing nearly triple the cases by 2010 (approximately 1,600 versus 500 cases).
- The red line for Assam shows a steady upward trend from about 800 cases in 2001 to approximately 1,700 by 2010, with the steepest increases occurring between 2005-2007 and again after 2008.
- The blue line for Delhi shows a rise from about 400 cases in 2001 to a peak of roughly 650 cases around 2005, followed by a significant drop to about 450 cases in 2008, before a slight recovery by 2010.

- The diamond markers suggest that if current trends continue, Assam will see another increase to potentially 1,800 cases in 2011, while Delhi might experience a moderate increase to about 550 cases.
- The visualization highlights that Assam would need substantially more victim support resources than Delhi if these trends continue, directly supporting the resource allocation purpose mentioned in the story.

6.15 Kidnappings & Abductions Tab

6.15.1 Functionality Overview

This comprehensive tab provides a multi-faceted analysis of kidnapping and abduction cases, focusing on the purpose or motive behind the crime, temporal trends, state comparisons, and victim demographics.

6.15.2 Core Visualizations: Trend Lines, Bar Charts, Pie Charts, Heatmap

Similar to the Year Wise tab, this tab dynamically generates various visualizations based on the selected "Visualization Type" within the `kidnap-visualizations-container`.

- **Trend Analysis:** Shows the national trend line and YoY percentage change bar chart for kidnappings, either for `TOTAL_KIDNAPPINGS` or for a specific purpose selected by the user.
- **State Comparison (Counts):** Displays the top N states bar chart and top 5 contribution pie chart, based on the total count of kidnappings (either 'TOTAL' or for a specific purpose) aggregated over the selected years.
- **Purpose Profile Comparison (%):** Shows a heatmap where rows are selected states, columns are different kidnapping purposes (excluding 'Total'), and cell colors represent the percentage of that state's total kidnappings attributed to that specific purpose over the selected years. Tooltips show percentages and absolute counts. Includes a text display of total cases per selected state.
- **Victim Demographics Breakdown:** Presents a bar chart showing the distribution of victims across different age-gender groups (e.g., Female 0-10, Male 18-30) for the selected purpose ('TOTAL' or specific) and location (a specific state or 'All India') over the chosen years.

6.15.3 Interactive Elements

- **Year Range Slider (`id="kidnap-year-slider"`):** Selects the time period for analysis.
- **Purpose Dropdown (`id="kidnap-purpose-dropdown"`):** Allows selecting `TOTAL_KIDNAPPINGS` or a specific purpose (e.g., 'For Ransom', 'For Marriage', 'For Illicit Intercourse'). Options are dynamically populated and cleaned from the data. Used for Trend, State Comparison (Counts), and Demographics views.
- **State Multi-Select Dropdown (`id="kidnap-state-multiselect"`):** Allows selecting multiple states for the 'Purpose Profile Comparison' heatmap.
- **State Demographics Dropdown (`id="kidnap-state-demographics-dropdown"`):** Allows selecting a single state or 'All India' for the 'Victim Demographics Breakdown' view.
- **Visualization Type Radio (`id="kidnap-viz-type-radio"`):** Switches between the four analysis views: "Trend Analysis", "State Comparison (Counts)", "Purpose Profile Comparison (%)", and "Victim Demographics Breakdown".

6.15.4 Underlying Data: Kidnapping Purpose and Demographics (`df_kidnap`)

Data is loaded from `39_Specific_purpose_of_kidnapping_and_abduction_roshan.csv` using the dedicated `load_and_clean_kidnapping_purpose_csv` function. This function standardizes columns, renames key fields, cleans the purpose labels (`PURPOSE_CLEAN`), and handles numeric conversions. This single DataFrame contains counts broken down by state, year, specific purpose, and detailed victim age/gender groups (`K_A_FEMALE_UPTO_10_YEARS`, etc.).

6.15.5 Underlying Callbacks

- `update_kidnap_purpose_options`: Triggered when the main tab value changes. Populates the `kidnap-purpose-dropdown` options with cleaned purpose labels and `TOTAL_KIDNAPPINGS`.
- `update_kidnap_visualizations`: The main callback for this tab. Triggered by changes in any of the controls (slider, dropdowns, radio). Based on the selected `viz_type`, it filters `df_kidnap` by year, purpose (if applicable), and state(s) (if applicable), performs the necessary aggregations (national totals, state totals, purpose percentages, demographic sums), and generates the corresponding Plotly figures (line, bar, pie, heatmap) displayed in `kidnap-visualizations-container.children`.

6.15.6 The Story: Deep Dive into the Motives and Victimology of Kidnappings

Kidnapping is a serious crime. Understanding these motives and who the victims are is essential for targeted prevention and effective investigation. This tab facilitates this deep dive by addressing questions like: **"What are the most common reasons for kidnapping nationally or in specific states? How do states differ in their kidnapping profiles based on motive? What are the age and gender characteristics of victims for different types of kidnappings?"**

- These graphs helps police and researchers understand why kidnappings happen and who the victims are, which is crucial for preventing these crimes.
- The tab shows the most common reasons for kidnappings across the country or in specific states.
- Users can see how different states compare in terms of kidnapping motives and patterns.
- The graph reveals age and gender patterns of victims for different types of kidnappings.
- With 'Trend Analysis,' users can track if certain types of kidnappings are increasing or decreasing over time.
- The 'State Comparison' feature shows which states have the highest numbers of specific kidnapping types, for example **"kidnappings for marriage"**.
- The 'Purpose Profile Comparison' heatmap visually shows how kidnapping motives differ between states - for example, one state might have mostly ransom cases while another has mostly marriage-related cases.
- The 'Victim Demographics' view breaks down victim age and gender for specific kidnapping types in selected states.
- This detailed approach goes beyond simple counting of cases to reveal the underlying patterns and purposes.
- Understanding these patterns helps authorities create targeted prevention strategies and improve investigation techniques.

6.15.7 Dash-board Screenshots

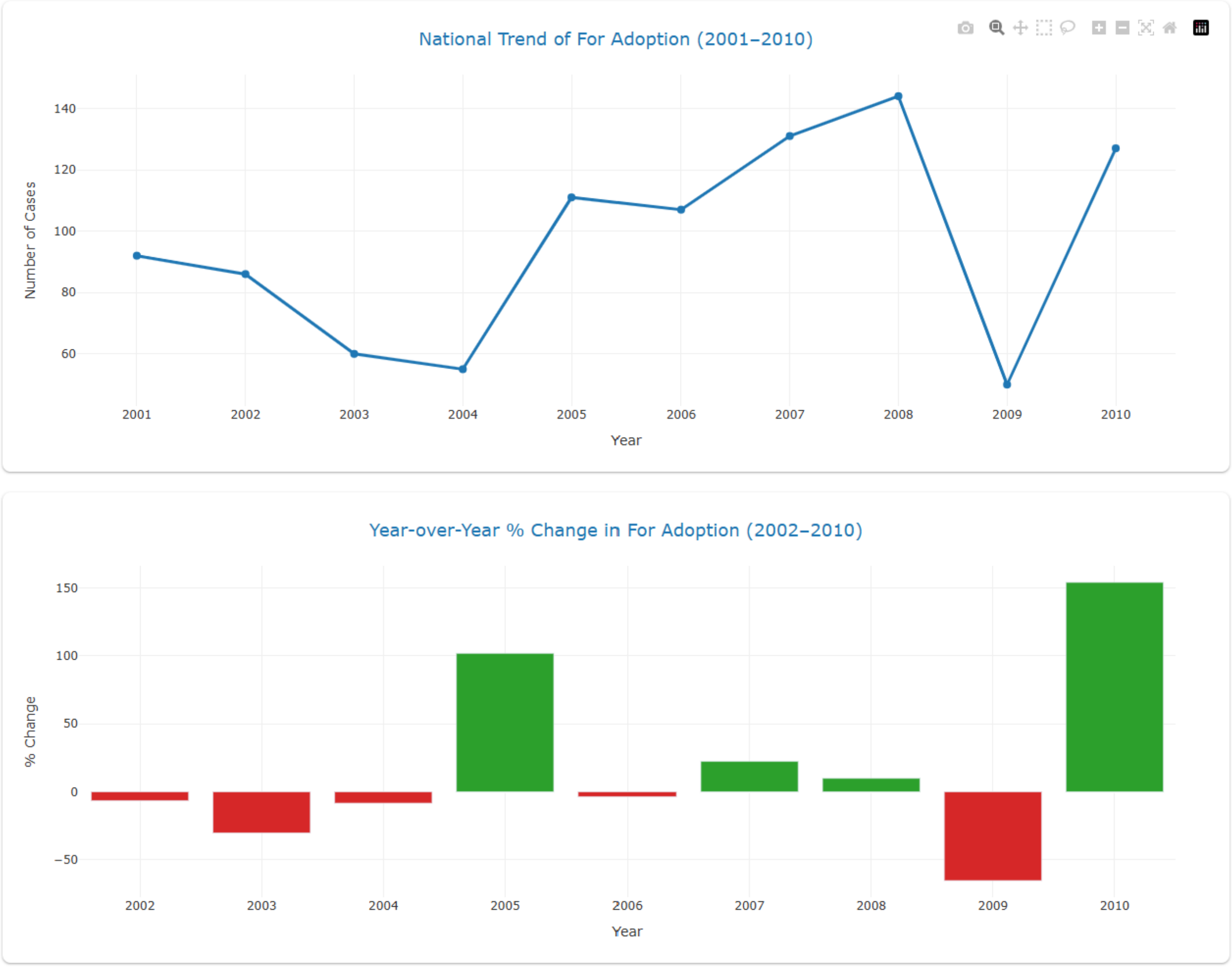


Figure 6.29: Kidnapping Tab showing Trend Analysis view for a specific purpose

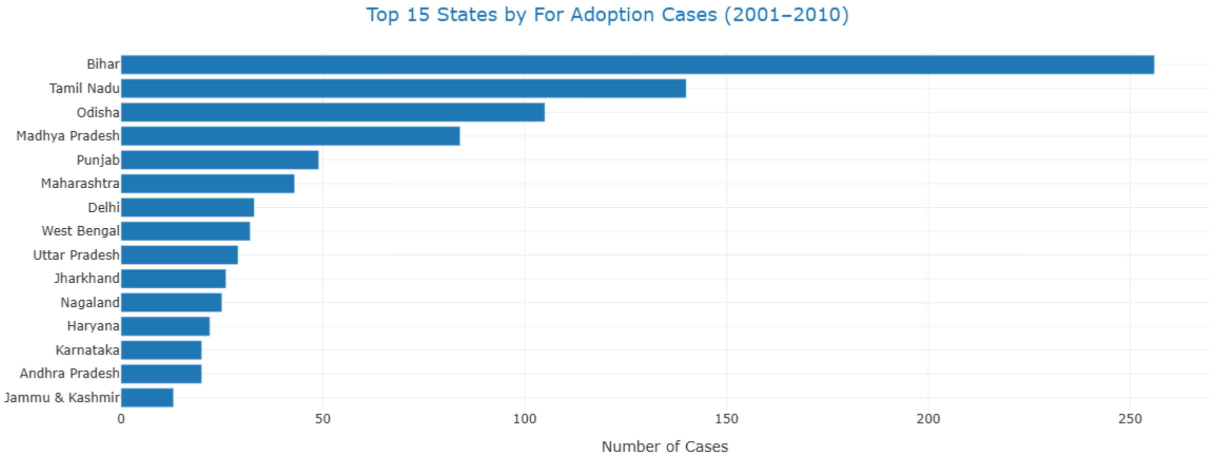


Figure 6.30: Kidnapping Tab showing State Comparison (Counts) view

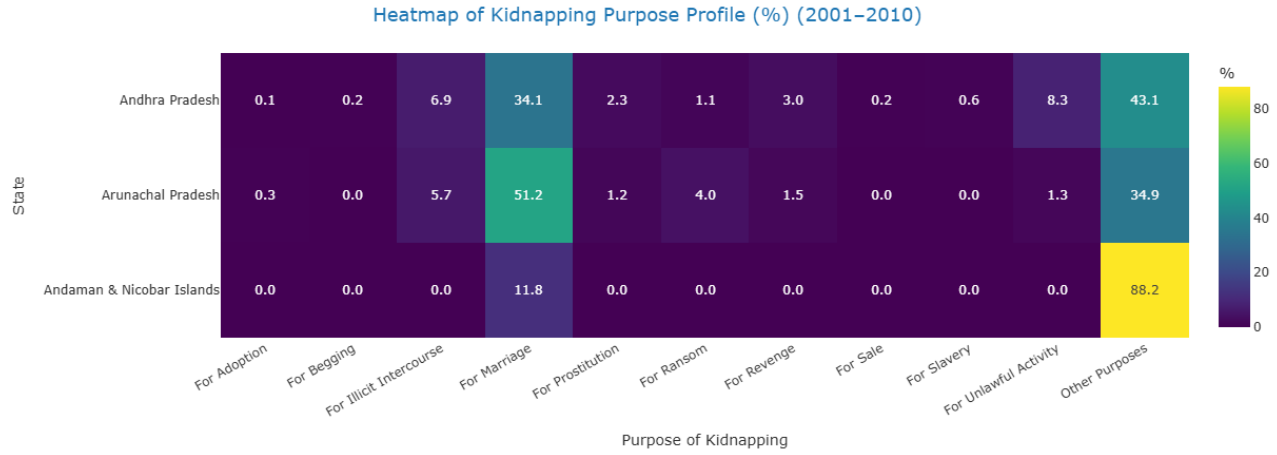


Figure 6.31: Kidnapping Tab showing Purpose Profile Comparison heatmap for selected states

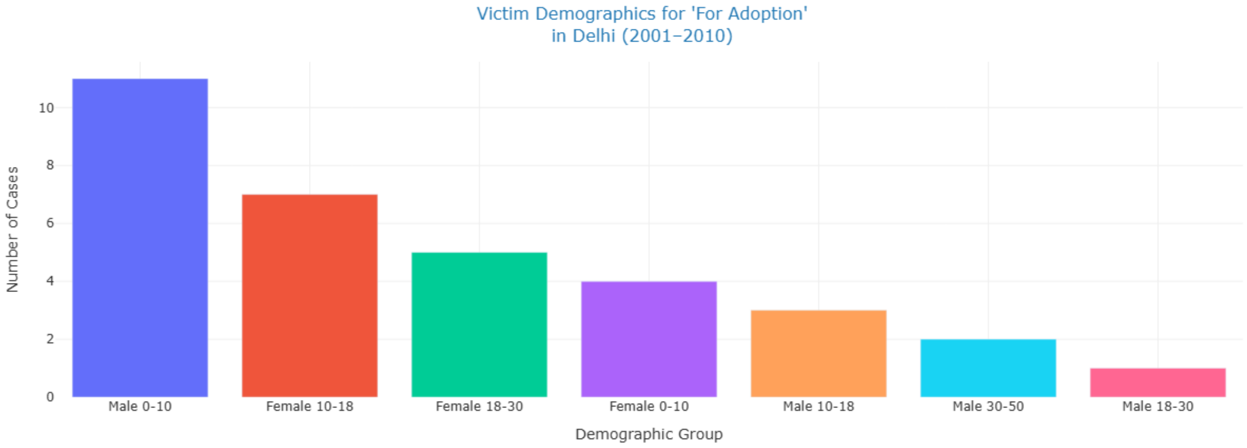


Figure 6.32: Kidnapping Tab showing Victim Demographics Breakdown bar chart for a selected purpose/location

6.15.8 Our Key Observations from Sample Plots

- The trend line shows dramatic fluctuations in kidnappings for adoption from 2001-2010, with cases nearly tripling between 2004-2005 (from about 55 to 110 cases) and sharply declining in 2009 before rising again in 2010.

- The year-over-year changes show kidnappings for adoption increased by approximately 100% in 2005 and over 150% in 2010, while dropping by nearly 60% in 2009, revealing the highly unstable nature of this crime category.

- Bihar stands out dramatically in the state comparison, reporting roughly 250 adoption-related kidnapping cases over the decade - nearly twice as many as second-ranked Tamil Nadu (approximately 140 cases) and significantly more than any other state.

- The purpose profile heatmap reveals "For Marriage" as the primary kidnapping motive in both Andhra Pradesh (34.1%) and Arunachal Pradesh (51.2%), exactly matching the story's example of identifying states with marriage-related kidnappings.

- The victim demographics chart confirms children are primary targets for adoption-related kidnappings in Delhi, with male children 0-10 years being the largest group (about 10 cases), followed by female adolescents 10-18 years (about 7 cases).

- The demographics breakdown shows a clear gender distribution pattern where males dominate the youngest victim category (0-10), while females are more prevalent as victims in the adolescent and young adult categories.

Chapter 7

Machine Learning Integration

The portal integrates several machine learning techniques to provide deeper analytical insights beyond basic descriptive statistics.

7.1 K-Means Clustering (Crime Clusters Tab)

- **Purpose:** To automatically group districts into clusters based on similarities in their reported IPC crime profiles (using user-selected crime features averaged over time). This helps identify distinct crime patterns that might not be obvious geographically.
- **Implementation:** Uses `sklearn.cluster.KMeans`. Data for selected features is aggregated (mean over years) per district, scaled using `StandardScaler`, and then fed into the KMeans algorithm with a user-specified number of clusters (K).
- **Visualization:** Results are shown via the Cluster Map (geographic distribution), Centroid Bar Chart (cluster profiles), and Silhouette Plot (cluster quality assessment).

7.2 ARIMA Forecasting (Crime Clusters Tab)

- **Purpose:** To provide a simple, short-term (one-year) forecast of the crime trend for each identified district cluster. This offers an exploratory glimpse into potential future developments based on past trends within each cluster profile.
- **Implementation:** Uses `statsmodels.tsa.arima.model.ARIMA`. For each cluster, the historical time series of average crime counts (either Total IPC or average of selected features) is extracted. An ARIMA model is fitted to this series, and a one-step forecast is generated. Error handling is included.
- **Visualization:** The forecast is displayed as a dashed line extension on the Cluster Trend line charts.

Chapter 8

Contributions

Student Name	**Specific Contributions**
Akshay Toshniwal	State-wise crimes + Juvenile Background Analysis
Ansh Makwe	Rape victims trend + Place of occurrence
Devang Agarwal	Firearms Victims Statewise + Serious Fraud Losses
Divyansh Chaurasia	District Wise crimes + Clustering of district from IPC crimes
Kunal Anand	Property Stolen + Year wise crimes
Parjanya Aditya Shukla	Custodial Death Plots + Murder Victims by age and sex
Prakhar Mandloi	Offenders Known to victim + Unidentified Dead Bodies Recovered
Roshan Kumar	Kidnapping and Abduction + Clustering of district from IPC crime

Table 8.1: CONTRIBUTIONS MADE BY EACH TEAM MEMBER

*** One major issue that we faced was non availability of complete geo-json of Indian geography. To resolve this issue we integrated multiple geo-json file to achieve complete India's Map. Another thing that we did was correcting names of different district if not matched with our data and geo-json file.**
* Every Team member was also responsible for applying required data preprocessing on their respective file and had complete freedom to come up with innovative representation for their data.
* Also all team members have combined effort for integrating their part of code and mentioning their part and observation in the final report.

Chapter 9

Challenges and Limitations

While the Indian Crime Data Visualization Portal offers a powerful tool for exploration, it's important to acknowledge its challenges and limitations:

- **Data Timeliness:** The core datasets used primarily cover the period up to 2012 or 2013. Crime patterns can change significantly over time, so the insights derived may not fully reflect the current situation. Accessing and integrating more recent, comparable data would be a major enhancement.
- **Data Quality and Consistency:** Publicly available administrative data can sometimes suffer from inconsistencies, under-reporting, definitional changes over time, or missing values.
- **Aggregation Level:** Most analyses are performed at the state or district level. This aggregation can mask significant variations within these administrative units. More granular data (e.g., police station level), if available and processable, could offer deeper insights but also presents greater data handling challenges.
- **Fuzzy Matching Limitations:** While `rapidfuzz` helps reconcile geographic names, it's not perfect. Incorrect matches are possible, potentially leading to data being assigned to the wrong geographic unit on maps. Manual verification might be needed for critical applications. We indeed did manual verification for a lot of districts' names.
- **Computational Load:** Processing large datasets and running ML models within callbacks can sometimes lead to slower response times, especially for complex operations like clustering or detailed map updates. The code attempts to mitigate this slightly by setting `LOKY_MAX_CPU_COUNT`.

Chapter 10

Future Work

The current portal provides a solid foundation, but numerous enhancements could further increase its value and utility:

- **Data Updates:** The most critical enhancement would be incorporating more recent crime data (post-2013) as it becomes available in a usable format, allowing for analysis of current trends.
- **Expanded Datasets:** Integrating additional relevant datasets, such as socio-economic indicators (poverty rates, education levels, unemployment) at the district level, could enable richer correlational analyses. Court conviction data could add another dimension to the justice system perspective.
- **Advanced ML Models:** Implementing more sophisticated machine learning models could provide deeper insights. Examples include:
  - Advanced time series models (e.g., SARIMA, Prophet) incorporating seasonality or external regressors for forecasting.
  - Different clustering algorithms (e.g., DBSCAN, hierarchical clustering) to compare results with K-Means.
  - Anomaly detection algorithms to automatically flag unusual spikes or deviations in crime rates.
- **Granular Geography:** If feasible based on data availability, adding analysis at sub-district levels (e.g., police station boundaries) would significantly enhance localized insights. This would require corresponding GeoJSON files and careful data mapping.
- **User Data Upload:** Allowing users to upload their own relevant datasets (e.g., local survey data, specific intervention results) for comparison or integration with the existing data could broaden the application's scope.

Chapter 11

Conclusion

The Indian Crime Data Visualization Portal, developed using Dash and Plotly, successfully integrates a diverse range of crime-related datasets spanning primarily 2001-2013 into a single, interactive platform. Through its multi-tabbed interface, the application empowers users to explore complex crime patterns across geographic, temporal, and thematic dimensions. Interactive choropleth maps, dynamic charts, and filtering controls enable users to move seamlessly from high-level state overviews to granular district-level analysis and specific crime investigations.

The portal demonstrates the power of data visualization in making large datasets accessible and interpretable. Visualizations like Sankey diagrams, treemaps, and heatmaps effectively communicate complex relationships and distributions related to victim demographics, offender relationships, and crime circumstances. Furthermore, the integration of machine learning techniques like K-Means clustering and ARIMA forecasting adds an analytical layer, enabling the discovery of hidden patterns in district crime profiles and offering exploratory insights into potential future trends.

While acknowledging limitations related to data timeliness and the inherent complexities of crime data, this portal serves as a valuable tool for users. By transforming raw statistics into interactive visual narratives, it fosters a more nuanced, data-informed understanding of crime in India and provides a robust foundation for future development and analysis.