AADHAAR

UIDAI DATA HACKATHON

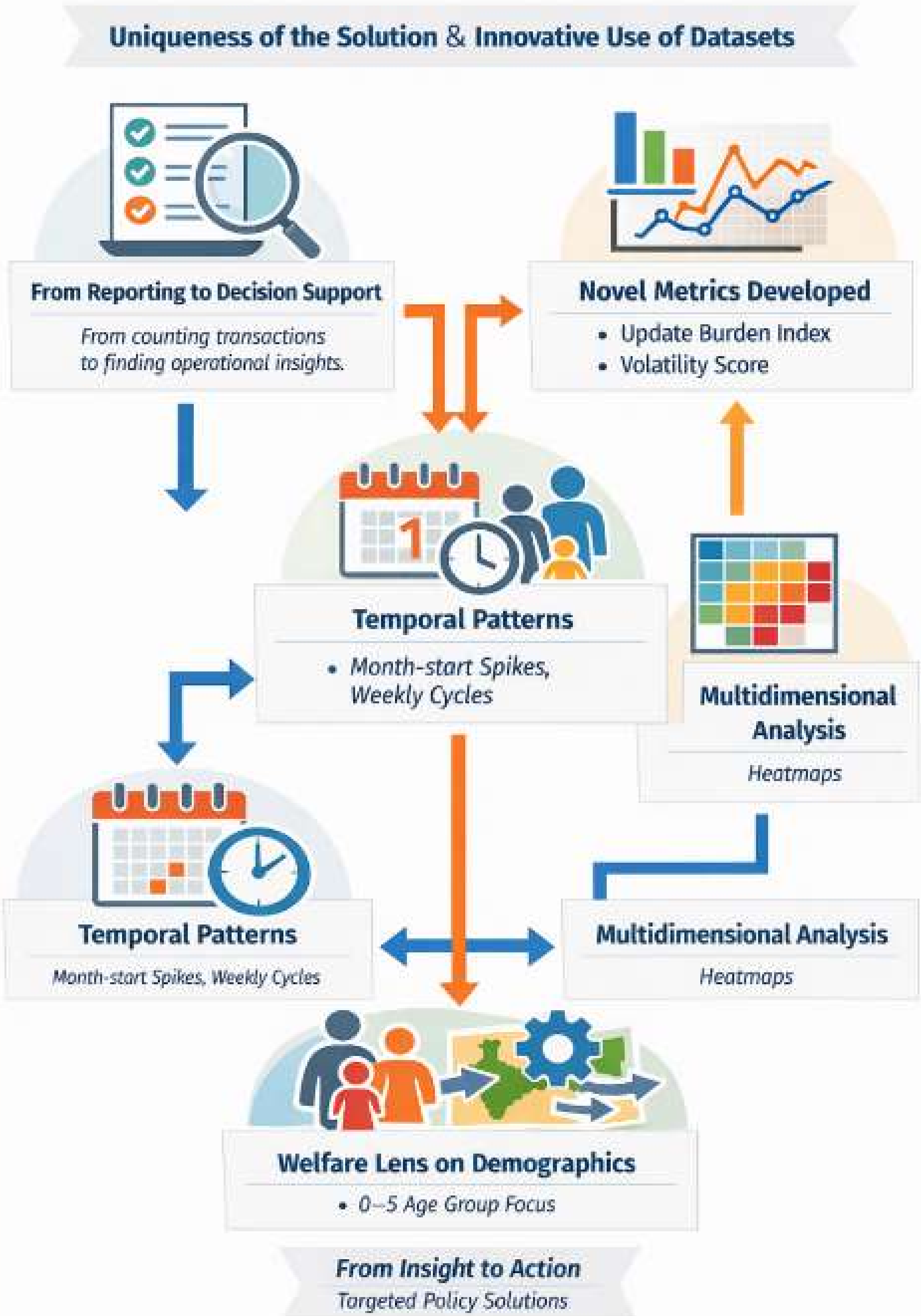
Unlocking Societal Trends in Aadhaar Enrolment & Updates

- Temporal Analysis
- Demographic Insights
- Operational Metrics

Submitted By:

Nikhil Singh Bhadoriya

nb4047051@gmail.com


Uniqueness of the Solution & Innovative Use of Datasets
From Reporting to Decision Support
From counting transactions to finding operational insights.
Novel Metrics Developed
• Update Burden Index
• Volatility Score
1
Temporal Patterns
• Month-start Spikes, Weekly Cycles
Multidimensional Analysis
Heatmaps
Temporal Patterns
Month-start Spikes, Weekly Cycles
Multidimensional Analysis
Heatmaps
Welfare Lens on Demographics
• 0–5 Age Group Focus
From Insight to Action
Targeted Policy Solutions

Problem Statement Approach

.Unique Problem Framing (Beyond Generic Analysis)

Most Aadhaar analyses focus on descriptive reporting, such as total enrolments by state or yearly trends

.

This work reframes the problem from reporting to decision support:
Instead of asking:
"How many Aadhaar enrolments and updates occurred?"
We ask:
"What does Aadhaar demand behaviour reveal about governance capacity, operational stress, and welfare delivery?"

This reframing itself is a key element of originality.

Problem Statement Approach

Transforming Aadhaar Data into Operational Intelligence

UIDAI datasets provide raw transaction counts.
This study innovatively converts those raw counts into governance-ready indicators, enabling actionable insights.

A. Update Burden Index (Original Metric)
Why it is innovative:
1. Not provided by UIDAI datasets
2. Converts raw activity logs into an operational stress signal
3. Distinguishes enrolment-heavy regions from maintenance-heavy regions

Why it matters:
1. Supports differentiated staffing and infrastructure strategies

B. Volatility Score (Original Metric)
Why it is innovative:
1. Captures demand instability, not just volume
2. Introduces risk-aware planning into Aadhaar analytics

Why it matters:
1. Identifies regions needing flexible or surge staffing

Problem Statement Approach

Innovative Temporal Interpretation

Rather than plotting time-series alone, the analysis extracts behavioural patterns, including:

1. Consistent 1st-of-the-month enrolment spikes
2. Weekday demand cycles

Why this is unique:

1. Links Aadhaar demand to administrative and policy cycles
2. Demonstrates that demand is systematic, not random

Problem Statement Approach

Innovative Social Interpretation of Demographic Data

Age-wise enrolment analysis is interpreted through a welfare and inclusion lens, not just demographics:

1. Dominance of 0–5 age group linked to child welfare schemes
2. District-level enrolments interpreted as rural inclusion signals

Why this is unique:

1. Moves beyond numbers to societal meaning
2. Connects Aadhaar analytics with policy outcomes

Problem Statement Approach

From Data to Action (Solution-Level Originality)

The solution does not stop at insights.
Each finding is mapped to a feasible administrative action, such as:

1. Month-start staffing optimisation
2. Update-only counters in high-burden regions
3. Flexible deployment in volatile states

Why this is innovative:

1. Converts analytics into implementable governance strategies
2. Treats Aadhaar data as a planning tool, not just an archive

Dataset Description

The analysis uses official Aadhaar datasets published by the Unique Identification Authority of India (UIDAI), ensuring authenticity and reliability of the data.

Three datasets were utilised: Aadhaar Enrolment, Aadhaar Demographic Update, and Aadhaar Biometric Update datasets

Each dataset contains date-wise and location-wise records, with geographic identifiers such as state, district, and pincode enabling multi-level regional analysis

The Aadhaar Enrolment dataset includes age-group-wise enrolment counts, specifically for 0–5 years, 5–17 years, and 18 years and above, allowing demographic segmentation.

Dataset Description

The Demographic Update dataset captures age-wise corrections to personal information of Aadhaar holders, while the Biometric Update dataset records biometric maintenance transactions such as fingerprint and iris updates

Core analytical columns used include date, state, district, age-wise enrolment/update fields, from which derived measures such as total enrolments and total updates were computed.

Temporal features (month, weekday, and month-start indicators) were derived from the date column to support time-series, seasonal, and behavioural analysis.

Methodology

Data Ingestion and Integration

The analysis utilised three official UIDAI datasets covering Aadhaar enrolments, demographic updates, and biometric updates. All datasets were ingested in their raw CSV format and processed using a single, unified analytical pipeline to ensure consistency across sources. Records were aligned temporally using the date field to enable combined time-series analysis.

Data Cleaning

To ensure data quality and analytical reliability, the following cleaning steps were applied:

1. Standardised column names and trimmed textual fields to remove formatting inconsistencies.
2. Converted date fields into a uniform datetime format to enable temporal analysis.
3. Coerced age-wise and count-based fields into numeric types, handling non-numeric values gracefully.
4. Removed duplicate records at the date–location level where applicable.
5. Checked for and validated the absence of negative values in transaction counts.

Problem Statement Approach

Creative Multidimensional Analysis

The study goes beyond basic univariate and bivariate plots by applying:

1. Trivariate heatmaps (State × Month × Update/Enrolment Ratio)
2. State × Age Group × Enrolment analysis
3. Multivariate PCA clustering to identify operational archetypes

Why this is innovative:

1. Reveals interaction effects not visible in simple summaries
2. Rarely applied in student-level Aadhaar analysis

Methodology

Preprocessing and Feature Engineering

Several preprocessing steps and derived features were introduced to make the data analytically meaningful:

1. Total enrolments were computed by aggregating age-wise enrolmen fields (0–5, 5–17, 18+).
2. Total updates were calculated by combining demographic and biometric update counts.
3. Total Aadhaar activity was defined as the sum of enrolments and updates.
4. Temporal features such as year, month, weekday, and month-start indicators were derived from the date field to capture seasonal and administrative patterns.
5. Rolling averages (7-day and 14-day) were applied to smooth short-term noise and isolate underlying trends.

Transformations and Normalisation

To enable fair comparison across regions with different population sizes and activity levels, ratio-based transformations were applied:

1. Update Burden Index was computed as the ratio of total updates to total enrolments, highlighting maintenance-heavy regions.
2. Volatility Score was calculated as the standard deviation of daily activity divided by the mean daily activity, capturing demand instability.
3. Zero-denominator safeguards were implemented to prevent misleading ratio values in low-activity regions.

Methodology

Aggregation Strategy

The cleaned and transformed data were aggregated at multiple levels to support layered analysis:

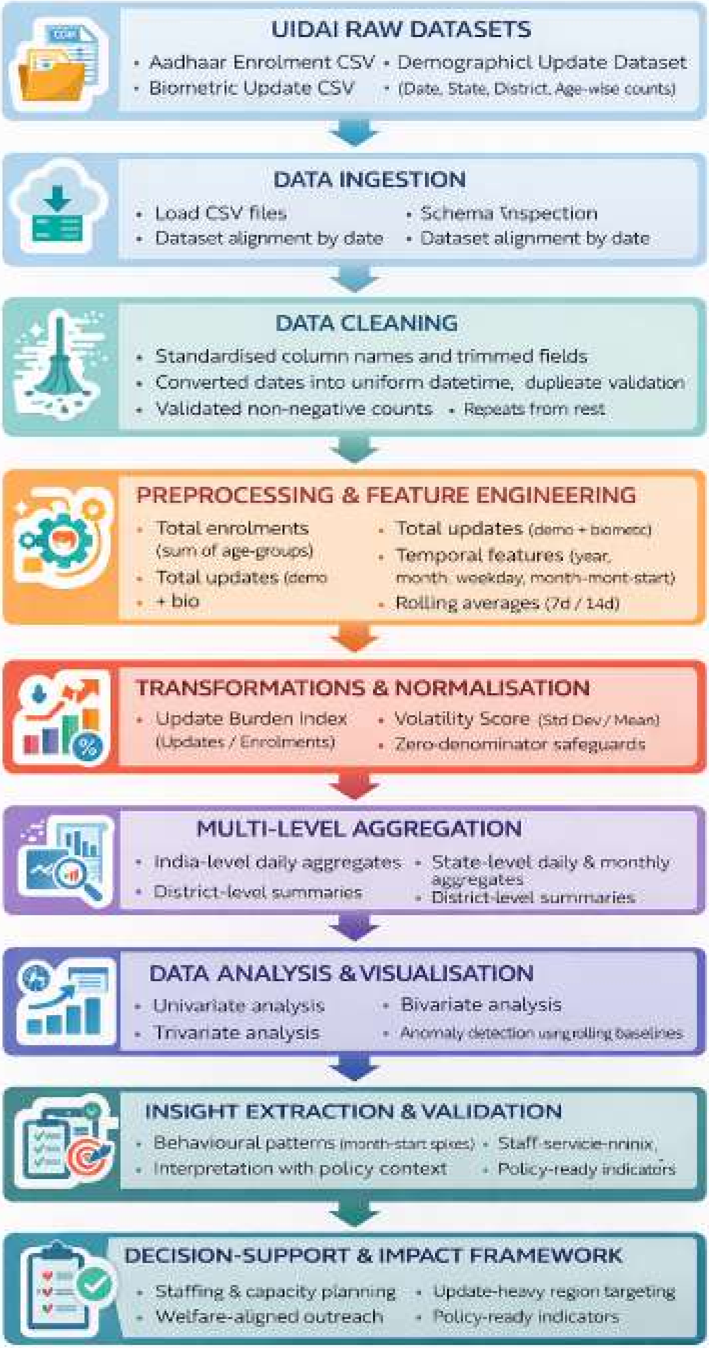
- India-level daily aggregates for national trends.
- State-level daily and monthly aggregates for regional comparison and seasonality.
- District-level summaries to assess localised inclusion and outreach patterns.

Analytical Techniques

The study employed a combination of univariate, bivariate, and trivariate techniques:

- Univariate analysis to establish baseline demand distributions.
- Bivariate analysis to examine relationships between enrolments and updates across geography and time.
- Trivariate analysis using heatmaps and stacked charts to uncover interaction effects between state, time, and demographics.
- Anomaly detection using statistical thresholds and rolling baselines to identify unusual demand patterns.

Aadhaar Analytics Methodology Pipeline



Data Analysis and Visualisation

Analytical Approach

The analysis followed a structured progression from univariate to multivariate techniques in order to extract reliable and meaningful insights from the UIDAI Aadhaar datasets. National-, state-, and district-level views were combined with temporal and demographic dimensions to understand enrolment and update behaviour comprehensively.

Key Findings and Insights

adhaar demand is update-dominant rather than enrolment-dominant, indicating that Aadhaar functions primarily as a lifecycle identity system requiring continuous maintenance.

Strong temporal patterns were observed, including consistent enrolment spikes on the first day of each month, revealing administratively timed demand cycles.

Significant regional variation exists across states, with some regions exhibiting high enrolment volumes while others are dominated by update activity.

Age-wise analysis shows that the 0–5 years group contributes the largest share of enrolments, highlighting Aadhaar's alignment with child-centric welfare and inclusion programs.

Update Burden Index and Volatility Score revealed maintenance-heavy and unstable-demand regions that require differentiated operational strategies rather than uniform policy treatment.

District-level analysis demonstrated strong rural and semi-urban participation, reinforcing Aadhaar's role in inclusive governance.

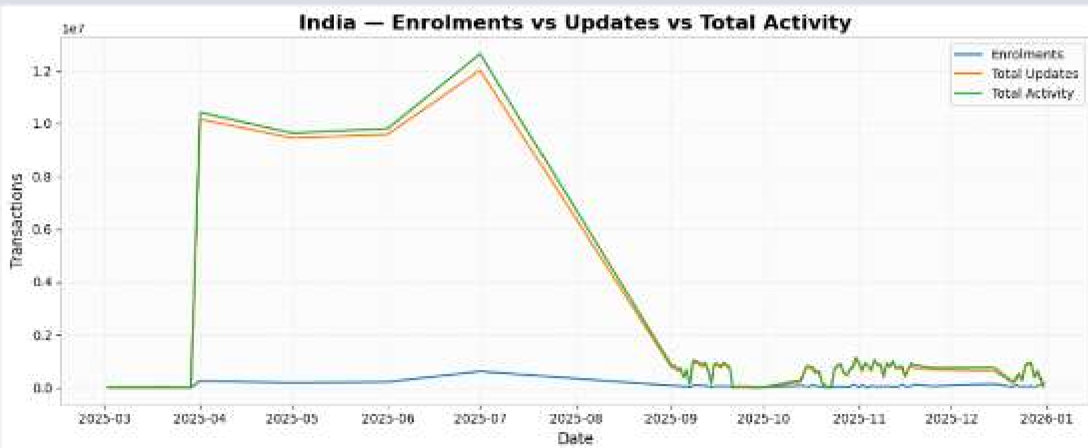
Data Analysis and Visualisation

Daily Aadhaar Enrolments and Updates

Purpose:
This graph compares daily Aadhaar enrolments with demographic and biometric updates to understand the composition of Aadhaar demand over time.

What It shows

Total activity closely tracks update volumes.
Enrolments form a smaller but stable component.



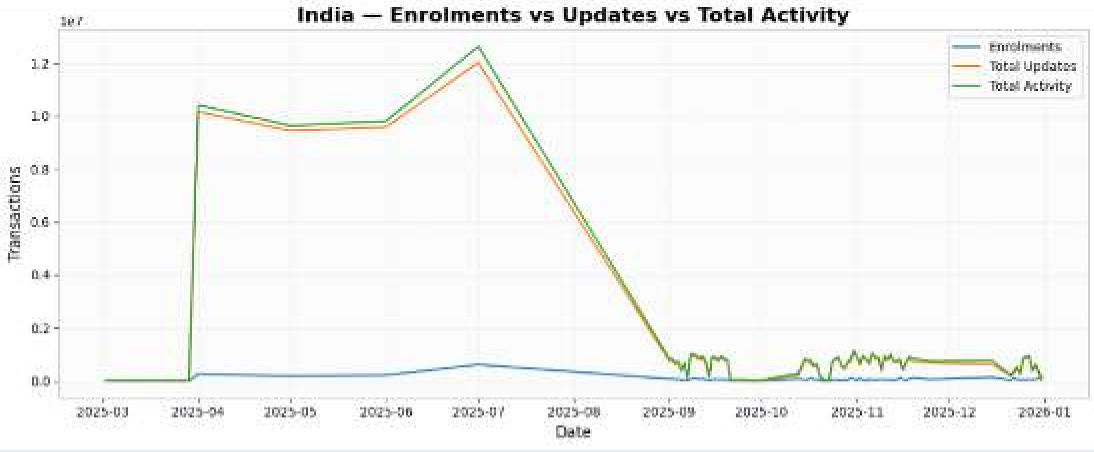
Data Analysis and Visualisation

Daily Aadhaar Enrolments and Updates

Purpose:
This graph compares daily Aadhaar enrolments with demographic and biometric updates to understand the composition of Aadhaar demand over time.

What It shows

Total activity closely tracks update volumes.
Enrolments form a smaller but stable component.



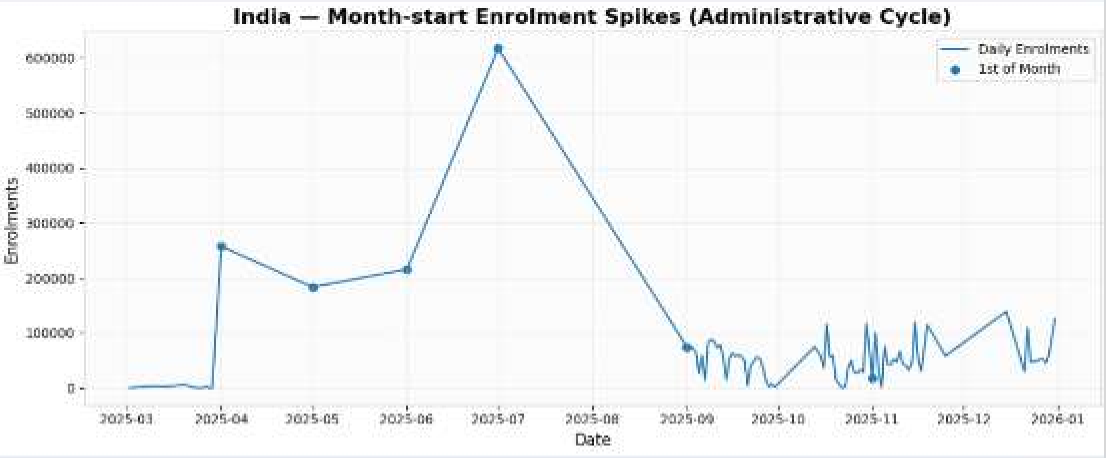
Data Analysis and Visualisation

Month-Start Enrolment Spikes

Purpose:
To detect administratively driven demand cycles

What It shows

Clear and recurring enrolment spikes on the first day of each month



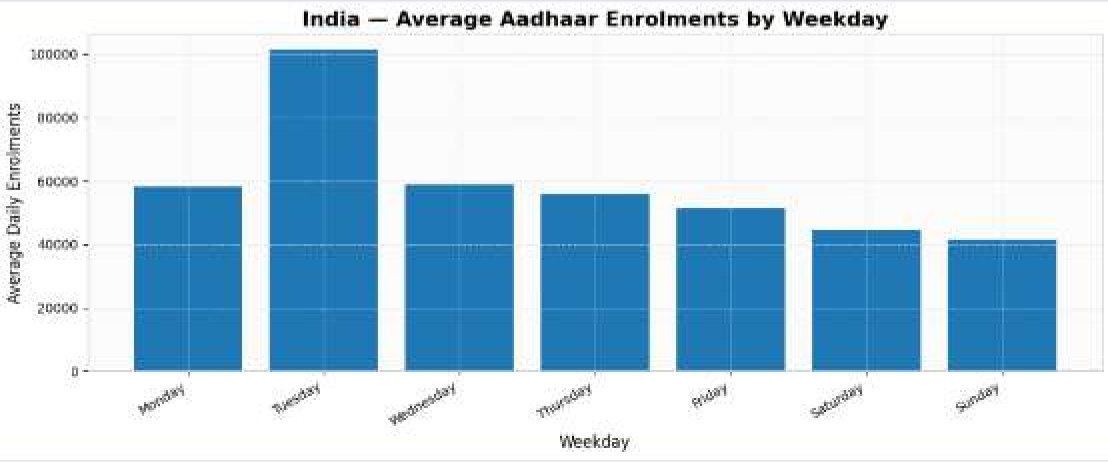
Data Analysis and Visualisation

Average Aadhaar Enrolments by Weekd

Purpose:
To analyse weekday-wise behavioural patterns in enrolment demand

What It shows

Mid-week days typically have higher average enrolments.
Weekends show comparatively lower activity.



Data Analysis and Visualisation

Enrolment Anomalies

Purpose:
To identify exceptional demand days beyond normal variability.

What It shows

Extreme spikes and drops that deviate significantly from historical norms.

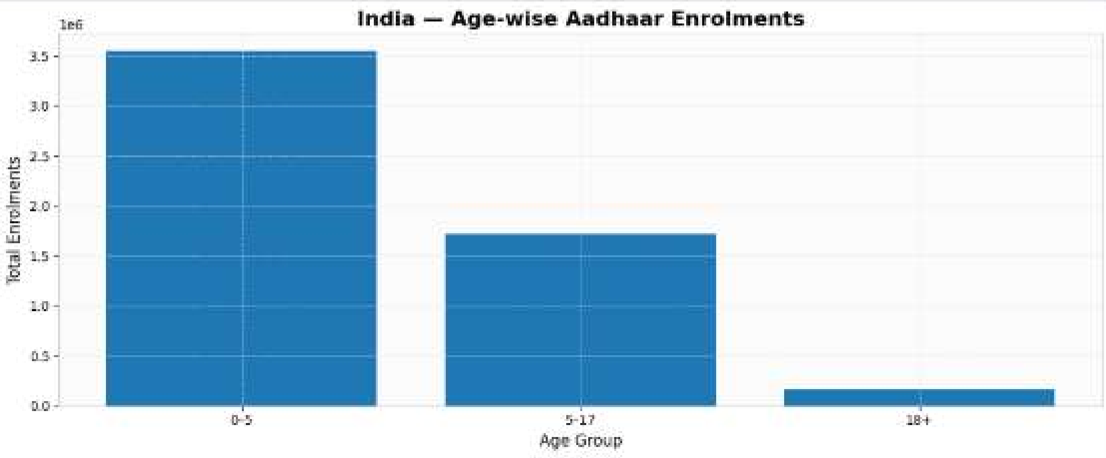
Data Analysis and Visualisation

Age-wise Aadhaar Enrolments

Purpose:
To understand who is enrolling in Aadhaar

What It shows

The 0–5 age group dominates enrolments



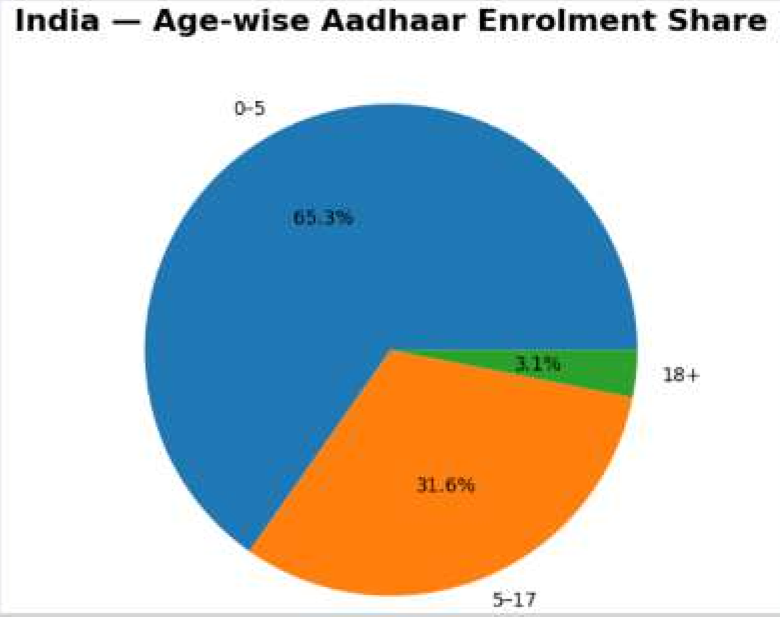
Data Analysis and Visualisation

Age-wise Aadhaar Enrolment Share

Purpose:
To present relative contribution of age groups clearly.

What It shows

Visual dominance of child enrolments over adult first-time registrations.



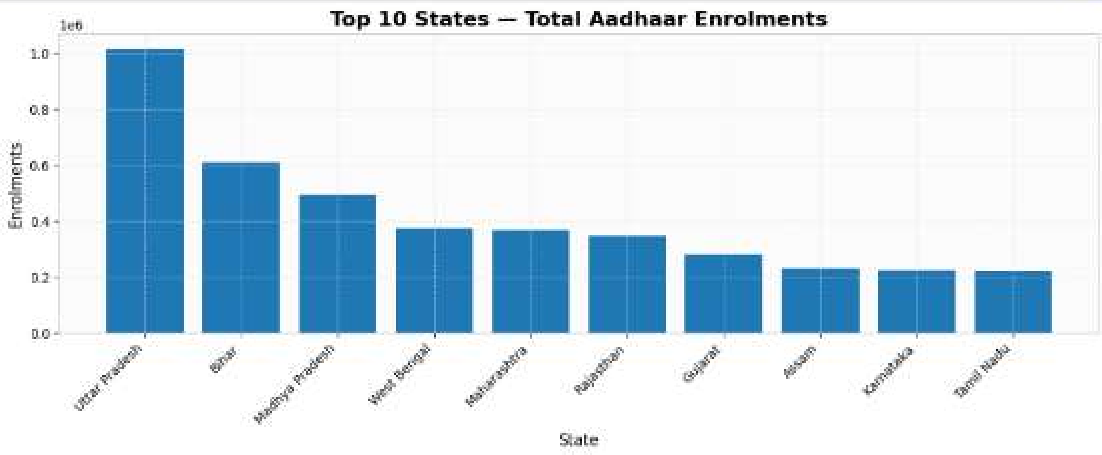
Data Analysis and Visualisation

Top 10 States — Total Aadhaar Enrolments

Purpose:
To identify high-impact states by enrolment volume

What It shows

Aadhaar enrolments are concentrated in a few populous states.



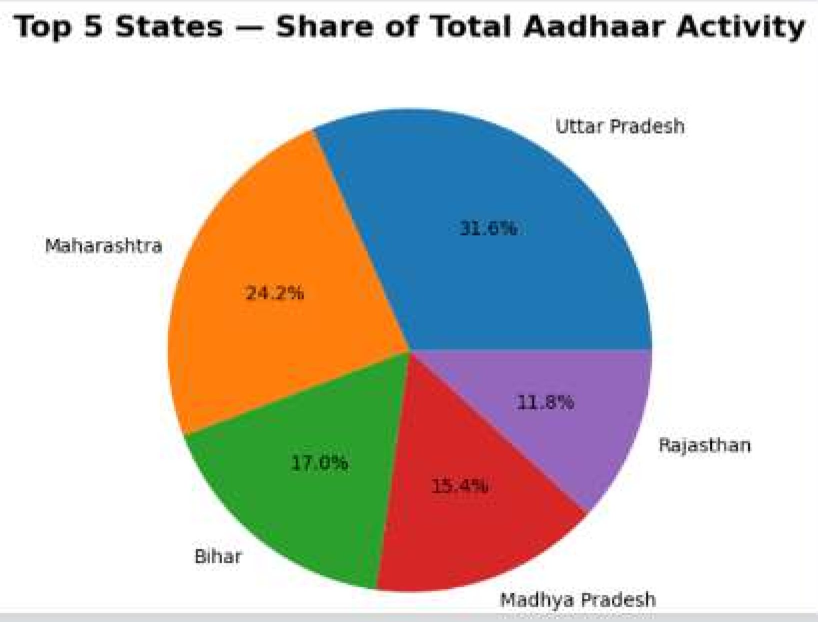
Data Analysis and Visualisation

Top 5 States — Share of Total Aadhaar Activity

Purpose:
To show concentration of total Aadhaar workload.

What It shows

A small number of states account for a majority of activity.



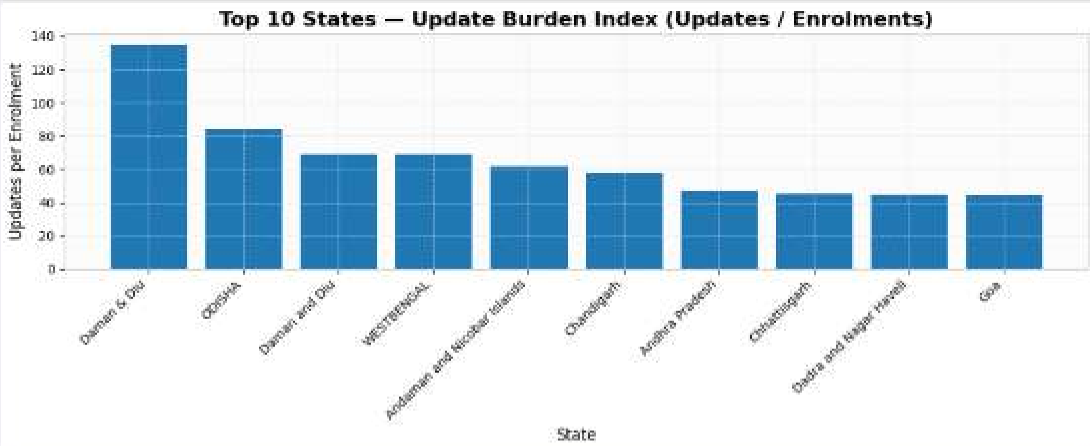
Data Analysis and Visualisation

Top 10 States — Update Burden Index

Purpose:
To quantify how update-heavy each state is relative to enrolments.

What It shows

Some states process far more updates than enrolments.



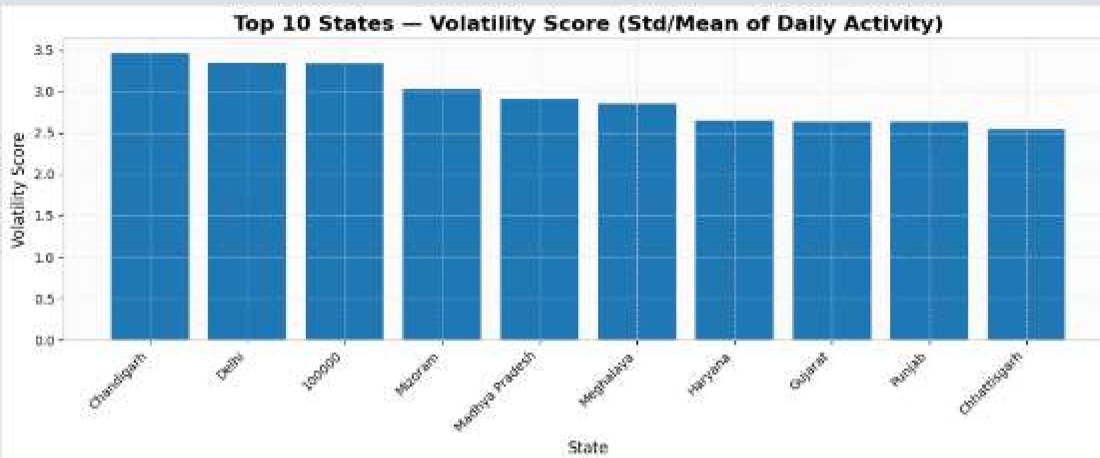
Data Analysis and Visualisation

Top 10 States — Volatility Score

Purpose:
To measure demand stability vs instability.

What It shows

Some states experience highly fluctuating daily demand.



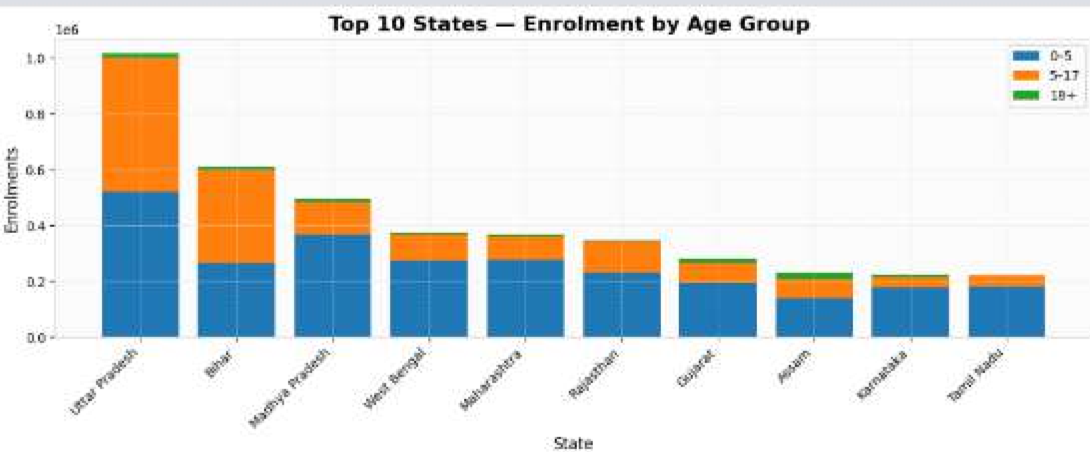
Data Analysis and Visualisation

Top 10 States — Enrolment by Age Group

Purpose:
To analyse state × age group × enrolment volume simultaneously.

What It shows

Child enrolments dominate even in high-volume states.



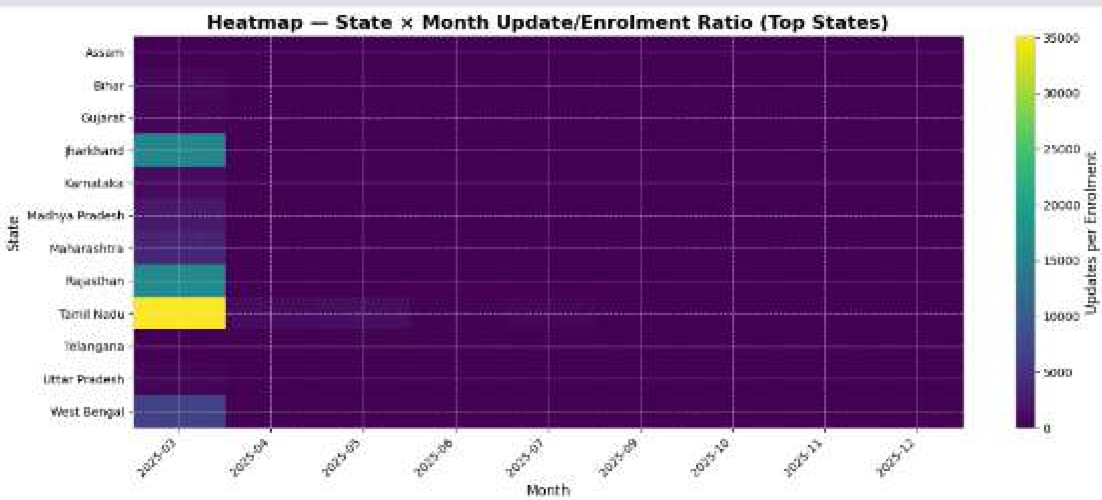
Data Analysis and Visualisation

Heatmap — State × Month Update/Enrolment Ratio

Purpose:
To capture seasonal and regional interaction effects.

What It shows

Update pressure varies across both states and months



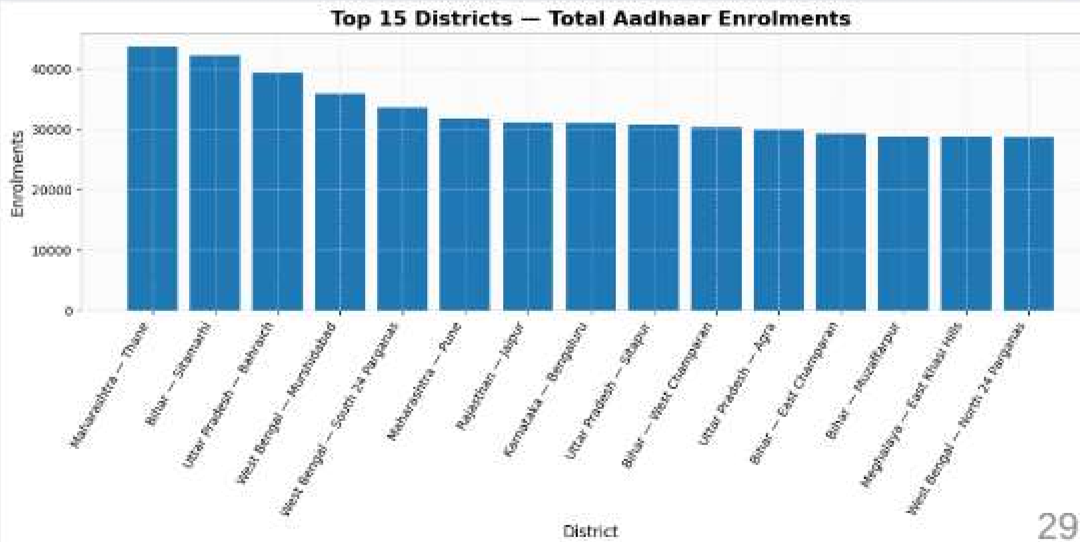
Data Analysis and Visualisation

Top 15 Districts — Total Aadhaar Enrolments

Purpose:
To assess grassroots participation.

What It shows

Strong enrolment activity in rural and semi-urban districts.



Data Analysis Impact & Applicability

Administrative Impact: Smarter Capacity Planning

Finding:
Aadhaar demand is update-dominant and follows predictable temporal patterns (month-start and weekday peaks).
Actionable Application:
1. Allocate additional operators and biometric kits during:
2. First week of each month
3. High-demand weekdays (mid-week)
4. Reduce staffing during consistently low-demand periods.

Impact:
1. Shorter queues and faster service delivery
2. Improved utilisation of existing infrastructure
3. No additional capital expenditure required

Feasibility:
✓ Can be implemented immediately using historical demand data
✓ Requires only scheduling changes, not system redesign

Data Analysis Impact & Applicability

Service Mix Optimization Using Update Burden Index

Finding:
Several states process significantly more updates than new enrolments, indicating mature Aadhaar regions.

Actionable Application:
1. Increase update-only counters in high-burden states
2. Prioritise biometric infrastructure where update load is high

Impact:
1. Reduced processing delays for updates
2. Improved Aadhaar lifecycle maintenance
3. Better compliance with service-level agreements (SLAs)

Feasibility:
✓ Uses existing UIDAI data and infrastructure
✓ Scalable across states without new datasets

Data Analysis Impact & Applicability

Risk-Aware Planning Through Volatility Analysis

Finding:
Certain states exhibit high volatility in daily Aadhaar demand.

Actionable Application:

1. Deploy flexible staffing models in volatile regions
2. Introduce surge-response playbooks during campaigns or policy changes

Impact:

1. Improved system resilience
2. Reduced service disruptions during demand surges

Feasibility:

✓ Volatility Score can be auto-computed monthly

✓ Integrates easily into existing monitoring dashboards

Data Analysis Impact & Applicability

District-Level Insights for Last-Mile Governance

Finding:
High enrolment volumes in rural and semi-urban districts.

Actionable Application:

1. Deploy mobile enrolment units in high-participation districts
2. Schedule district-level drives during peak demand windows

Impact:

1. Enhanced last-mile service delivery
2. Reduced urban-rural access gaps

Feasibility:

✓ Based on already available district data

✓ Supports decentralised planning

Data Analysis Impact & Applicability

Governance-Ready Indicators for Long-Term Use

What was created:

1. Update Burden Index
2. Volatility Score

Why it matters:

These indicators:

1. Convert raw UIDAI data into governance signals
2. Can be tracked monthly or quarterly
3. Enable evidence-based administrative decisions

Feasibility:

✓ No new data collection required
✓ Easily automated and scalable nationally

Data Analysis Impact & Applicability

Governance-Ready Indicators for Long-Term Use

What was created:

- Update Burden Index
- Volatility Score

Why it matters:

These indicators:

- Convert raw UIDAI data into governance signals
- Can be tracked monthly or quarterly
- Enable evidence-based administrative decisions

Feasibility:

✓ No new data collection required
✓ Easily automated and scalable nationally

Conclusion

This study reframes Aadhaar enrolment and update data as operational intelligence, applying multi-dimensional analytics to support evidence-based governance and capacity planning.

Official UIDAI Aadhaar enrolment and update datasets were systematically utilised to derive demographic, temporal, and regional insights relevant to administrative decision-making.

A reproducible, stepwise methodology involving data cleaning, feature engineering, and statistical transformations ensured analytical rigour and reliability of results.

Multi-level analysis and purpose-driven visualisations translated complex Aadhaar transaction data into clear, actionable insights.

-------Thank You------

sol.py

```python
# ============================================================
# Aadhaar Analytics — Full Graph Generation Code (Attractive)
# Covers: univariate, bivariate, trivariate, multivariate visuals
# Uses: pandas, numpy, matplotlib, scikit-learn
# NOTE: No seaborn, and no manual color setting.
# ============================================================

import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# -------------------------
# CONFIG
# -------------------------
ENROL_PATH = "api_data_aadhar_enrolment_0_1006029.csv"
DEMO_PATH  = "api_data_aadhar_demographic_0_2071700.csv"
BIO_PATH   = "api_data_aadhar_biometric_0_1861108.csv"

SAVE_FIGS = False
OUTDIR = "/mnt/data/aadhaar_figs"
os.makedirs(OUTDIR, exist_ok=True)

plt.rcParams["figure.figsize"] = (12, 5)
plt.rcParams["axes.grid"] = True

def savefig(name: str):
    if SAVE_FIGS:
        fp = os.path.join(OUTDIR, name)
        plt.savefig(fp, dpi=220, bbox_inches="tight")
        print("Saved:", fp)

# -------------------------
# LOAD + CLEAN
# -------------------------
def load_uidai_csv(path: str) -> pd.DataFrame:
    df = pd.read_csv(path)
    if "Unnamed: 0" in df.columns:
        df = df.drop(columns=["Unnamed: 0"])
    df.columns = [c.strip() for c in df.columns]
    if "date" not in df.columns:
        raise ValueError(f"Missing 'date' column in {path}")
    df["date"] = pd.to_datetime(df["date"], dayfirst=True, errors="coerce")
    # strip common geo fields if present
    for c in ["state", "district", "pincode"]:
        if c in df.columns:
            df[c] = df[c].astype(str).str.strip()
```

```
    return df

enrol = load_uidai_csv(ENROL_PATH)
demo  = load_uidai_csv(DEMO_PATH)
bio   = load_uidai_csv(BIO_PATH)

# -------------------------
# FEATURE ENGINEERING
# -------------------------
# totals
enrol["total_enrol"] = enrol[["age_0_5", "age_5_17",
"age_18_greater"]].apply(pd.to_numeric, errors="coerce").fillna(0).sum(axis=1)

# demo/bio columns vary a bit; sum all columns that start with demo_ / bio_
demo_cols = [c for c in demo.columns if c.startswith("demo_")]
bio_cols  = [c for c in bio.columns  if c.startswith("bio_")]

demo["total_demo"] = demo[demo_cols].apply(pd.to_numeric,
errors="coerce").fillna(0).sum(axis=1)
bio["total_bio"]   = bio[bio_cols].apply(pd.to_numeric,
errors="coerce").fillna(0).sum(axis=1)

eps = 1e-9

# -------------------------
# INDIA DAILY TABLE
# -------------------------
daily = (
    enrol.groupby("date")["total_enrol"].sum().to_frame("Enrolments")
    .join(demo.groupby("date")["total_demo"].sum())
    .join(bio.groupby("date")["total_bio"].sum())
    .fillna(0)
)
daily.rename(columns={"total_demo": "Demographic Updates", "total_bio": "Biometric
Updates"}, inplace=True)
daily["Total Updates"] = daily["Demographic Updates"] + daily["Biometric Updates"]
daily["Total Activity"] = daily["Enrolments"] + daily["Total Updates"]
daily["Enrol_7d_avg"] = daily["Enrolments"].rolling(7, min_periods=3).mean()
daily["Upd_7d_avg"]   = daily["Total Updates"].rolling(7, min_periods=3).mean()
daily["Act_7d_avg"]   = daily["Total Activity"].rolling(7, min_periods=3).mean()
daily["is_month_start"] = daily.index.is_month_start

# ============================================================
# 1) Time-series: Daily enrolments + demo + bio updates
# (supports: update-dominant lifecycle insight)
# ============================================================
plt.figure()
plt.plot(daily.index, daily["Enrolments"], label="Enrolments")
plt.plot(daily.index, daily["Demographic Updates"], label="Demographic Updates")
plt.plot(daily.index, daily["Biometric Updates"], label="Biometric Updates")
plt.title("India — Daily Aadhaar Enrolments and Updates")
plt.xlabel("Date")
plt.ylabel("Transactions")
plt.legend()
```

```python
plt.tight_layout()
savefig("01_india_daily_enrol_demo_bio.png")
plt.show()

# ============================================================
# 2) Time-series: Enrolments vs Total Updates vs Total Activity
# ============================================================
plt.figure()
plt.plot(daily.index, daily["Enrolments"], label="Enrolments")
plt.plot(daily.index, daily["Total Updates"], label="Total Updates")
plt.plot(daily.index, daily["Total Activity"], label="Total Activity")
plt.title("India — Enrolments vs Updates vs Total Activity")
plt.xlabel("Date")
plt.ylabel("Transactions")
plt.legend()
plt.tight_layout()
savefig("02_india_enrol_vs_updates_vs_activity.png")
plt.show()

# ============================================================
# 3) Rolling averages (trend vs noise)
# ============================================================
plt.figure()
plt.plot(daily.index, daily["Enrol_7d_avg"], label="Enrolments (7d avg)")
plt.plot(daily.index, daily["Upd_7d_avg"], label="Updates (7d avg)")
plt.plot(daily.index, daily["Act_7d_avg"], label="Activity (7d avg)")
plt.title("India — 7-Day Rolling Averages (Trend)")
plt.xlabel("Date")
plt.ylabel("Transactions")
plt.legend()
plt.tight_layout()
savefig("03_india_rolling_7d.png")
plt.show()

# ============================================================
# 4) Annotated: Month-start enrolment spikes
# ============================================================
plt.figure()
plt.plot(daily.index, daily["Enrolments"], label="Daily Enrolments")
ms = daily[daily["is_month_start"]]
plt.scatter(ms.index, ms["Enrolments"], label="1st of Month")
plt.title("India — Month-start Enrolment Spikes (Administrative Cycle)")
plt.xlabel("Date")
plt.ylabel("Enrolments")
plt.legend()
plt.tight_layout()
savefig("04_month_start_spikes.png")
plt.show()

# ============================================================
# 5) Anomaly plot (global z-score)
# (supports: anomaly periods for operational interpretation)
# ============================================================
s = daily["Enrolments"].astype(float)
```

```
mu, sigma = s.mean(), s.std(ddof=0)
sigma = sigma if (sigma and not np.isnan(sigma)) else 1.0
z = (s - mu) / sigma
anom = z.abs() >= 3.0

plt.figure()
plt.plot(daily.index, s, label="Daily Enrolments")
plt.scatter(daily.index[anom], s[anom], label="Anomaly (|z|≥3)")
plt.title("India — Enrolment Anomalies ")
plt.xlabel("Date")
plt.ylabel("Enrolments")
plt.legend()
plt.tight_layout()
savefig("05_anomaly_global_z.png")
plt.show()

# ============================================================
# 6) Age-wise enrolments (bar)
# ============================================================
age_totals = enrol[["age_0_5", "age_5_17", "age_18_greater"]].apply(pd.to_numeric,
errors="coerce").fillna(0).sum()
plt.figure()
plt.bar(["0-5", "5-17", "18+"], age_totals.values)
plt.title("India — Age-wise Aadhaar Enrolments")
plt.xlabel("Age Group")
plt.ylabel("Total Enrolments")
plt.tight_layout()
savefig("06_agewise_enrol_bar.png")
plt.show()

# ============================================================
# 7) Age-wise share (pie)
# ============================================================
plt.figure()
plt.pie(age_totals.values, labels=["0-5", "5-17", "18+"], autopct="%1.1f%%")
plt.title("India — Age-wise Aadhaar Enrolment Share")
plt.tight_layout()
savefig("07_agewise_enrol_pie.png")
plt.show()

# ============================================================
# STATE-LEVEL TABLES (for regional variation, burden, volatility)
# ============================================================
state = (
    enrol.groupby("state")["total_enrol"].sum().to_frame("Enrol")
    .join(demo.groupby("state")["total_demo"].sum())
    .join(bio.groupby("state")["total_bio"].sum())
    .fillna(0)
)
state.rename(columns={"total_demo": "Demo", "total_bio": "Bio"}, inplace=True)
state["Updates"] = state["Demo"] + state["Bio"]
state["Total Activity"] = state["Enrol"] + state["Updates"]
state["Update Burden Index"] = state["Updates"] / (state["Enrol"] + eps)

```

```python
# ============================================================
# 8) Top states by total enrolments (bar ranking)
# ============================================================
top_enrol_states = state.sort_values("Enrol", ascending=False).head(10)
plt.figure()
plt.bar(top_enrol_states.index, top_enrol_states["Enrol"])
plt.title("Top 10 States — Total Aadhaar Enrolments")
plt.xlabel("State")
plt.ylabel("Enrolments")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
savefig("08_top10_states_enrol.png")
plt.show()

# ============================================================
# 9) Top states: share of total Aadhaar activity (pie)
# ============================================================
top5_activity = state.sort_values("Total Activity", ascending=False).head(5)
plt.figure()
plt.pie(top5_activity["Total Activity"].values, labels=top5_activity.index,
autopct="%1.1f%%")
plt.title("Top 5 States — Share of Total Aadhaar Activity")
plt.tight_layout()
savefig("09_top5_state_activity_share.png")
plt.show()

# ============================================================
# 10) Update Burden Index (bar ranking)
# ============================================================
top_burden = state.sort_values("Update Burden Index", ascending=False).head(10)
plt.figure()
plt.bar(top_burden.index, top_burden["Update Burden Index"])
plt.title("Top 10 States — Update Burden Index (Updates / Enrolments)")
plt.xlabel("State")
plt.ylabel("Updates per Enrolment")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
savefig("10_top10_update_burden.png")
plt.show()

# ============================================================
# 11) Volatility Score (std/mean of DAILY activity at state level)
# ============================================================
# build state-day activity (enrol + updates) to compute volatility
state_day_enrol = enrol.groupby(["state", "date"])["total_enrol"].sum().reset_index()
state_day_demo  = demo.groupby(["state", "date"])["total_demo"].sum().reset_index()
state_day_bio   = bio.groupby(["state", "date"])["total_bio"].sum().reset_index()

state_day = (
    state_day_enrol.merge(state_day_demo, on=["state","date"], how="left")
                  .merge(state_day_bio,  on=["state","date"], how="left")
                  .fillna(0)
)
```

```python
state_day["activity"] = state_day["total_enrol"] + state_day["total_demo"] + state_day["total_bio"]

vol = state_day.groupby("state")["activity"].agg(["mean","std"]).fillna(0)
vol["Volatility Score"] = vol["std"] / (vol["mean"] + eps)

top_vol = vol.sort_values("Volatility Score", ascending=False).head(10)
plt.figure()
plt.bar(top_vol.index, top_vol["Volatility Score"])
plt.title("Top 10 States – Volatility Score (Std/Mean of Daily Activity)")
plt.xlabel("State")
plt.ylabel("Volatility Score")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
savefig("11_top10_volatility.png")
plt.show()

# ============================================================
# 12) Stacked bars: Top states × age group (trivariate)
# ============================================================
state_age = enrol.groupby("state")[["age_0_5","age_5_17","age_18_greater"]].apply(
    lambda x: x.apply(pd.to_numeric, errors="coerce").fillna(0).sum()
)
state_age["total"] = state_age.sum(axis=1)
state_age = state_age.sort_values("total", ascending=False).head(10)

x = np.arange(len(state_age.index))
plt.figure(figsize=(12,5))
plt.bar(x, state_age["age_0_5"].values, label="0-5")
plt.bar(x, state_age["age_5_17"].values, bottom=state_age["age_0_5"].values, label="5-17")
plt.bar(x, state_age["age_18_greater"].values, bottom=(state_age["age_0_5"]+state_age["age_5_17"]).values, label="18+")
plt.title("Top 10 States – Enrolment by Age Group ")
plt.xlabel("State")
plt.ylabel("Enrolments")
plt.xticks(x, state_age.index, rotation=45, ha="right")
plt.legend()
plt.tight_layout()
savefig("12_stacked_age_top10.png")
plt.show()

# ============================================================
# 13) Trivariate heatmap: State × Month × Update/Enrol ratio
# (use Total Updates (demo+bio) / Enrolments)
# ============================================================
enrol_m = enrol.copy()
demo_m  = demo.copy()
bio_m   = bio.copy()

enrol_m["month"] = enrol_m["date"].dt.to_period("M").astype(str)
demo_m["month"]  = demo_m["date"].dt.to_period("M").astype(str)
bio_m["month"]   = bio_m["date"].dt.to_period("M").astype(str)

em = enrol_m.groupby(["state","month"])["total_enrol"].sum().reset_index()
```

```python
dm = demo_m.groupby(["state","month"])["total_demo"].sum().reset_index()
bm = bio_m.groupby(["state","month"])["total_bio"].sum().reset_index()

sm = em.merge(dm, on=["state","month"], how="left").merge(bm, on=["state","month"], how="left").fillna(0)
sm["updates"] = sm["total_demo"] + sm["total_bio"]
sm["ratio"] = sm["updates"] / (sm["total_enrol"] + eps)

top_states = sm.groupby("state")["total_enrol"].sum().sort_values(ascending=False).head(12).index
pivot = sm[sm["state"].isin(top_states)].pivot(index="state", columns="month", values="ratio").fillna(0)

plt.figure(figsize=(14,6))
plt.imshow(pivot.values, aspect="auto")
plt.title("Heatmap — State × Month Update/Enrolment Ratio (Top States)")
plt.xlabel("Month")
plt.ylabel("State")
plt.xticks(range(len(pivot.columns)), pivot.columns, rotation=45, ha="right")
plt.yticks(range(len(pivot.index)), pivot.index)
plt.colorbar(label="Updates per Enrolment")
plt.tight_layout()
savefig("13_heatmap_state_month_ratio.png")
plt.show()

# ============================================================
# 14) District-level analysis: Top districts by enrolments (bar)
# (supports: rural/semi-urban participation)
# NOTE: uses districts available in dataset; interpret with context.
# ============================================================
if "district" in enrol.columns:
    dist = enrol.groupby(["state","district"])["total_enrol"].sum().reset_index()
    # take top 15 overall districts
    dist_top = dist.sort_values("total_enrol", ascending=False).head(15)
    labels = dist_top["state"] + " — " + dist_top["district"]

    plt.figure(figsize=(12,6))
    plt.bar(labels, dist_top["total_enrol"].values)
    plt.title("Top 15 Districts — Total Aadhaar Enrolments")
    plt.xlabel("District")
    plt.ylabel("Enrolments")
    plt.xticks(rotation=60, ha="right")
    plt.tight_layout()
    savefig("14_top15_districts_enrol.png")
    plt.show()
```