Notebook 02: Exploratory Data Analysis

UIDAI Data Hackathon 2026

Problem: India's Invisible Citizens - Bridging Aadhaar Exclusion Zones

---

Objective

Identify exclusion patterns across three dimensions:

1. Geographic Exclusion - Which districts lag behind?
2. Demographic Vulnerability - Which age groups are underserved?
3. Temporal Patterns - When do enrollment gaps emerge?

Output: Data-driven insights for exclusion zone identification

---

Table of Contents

1. Load Prepared Data

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import warnings
warnings.filterwarnings('ignore')

# Set style
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (14, 8)

# Load master district data from Notebook 01
df = pd.read_csv('../outputs/tables/master_district_data.csv')

print(" Data loaded successfully")
```

```python
print(f" Districts: {len(df):,}")
print(f" States: {df['state'].nunique()}")
print(f"\nColumns: {list(df.columns)}")

display(df.head())
display(df.describe())
```

```
Data loaded successfully
Districts: 1,045
States: 49

Columns: ['state', 'district', 'age_0_5', 'age_5_17', 'age_18_greater', 'tota
l_enrollments', 'pincode_count', 'demo_update_count', 'bio_update_count', 'dem
o_update_intensity', 'bio_update_intensity', 'child_0_5_enrollment', 'child_enr
ollment_rate']
```

	state	district	age_0_5	age_5_17	age_18_greater	total_enrollments	pin
0	100000	100000	0	1	217	218	
1	Andaman & Nicobar Islands	Andamans	70	5	0	75	
2	Andaman & Nicobar Islands	Nicobars	1	0	0	1	
3	Andaman & Nicobar Islands	South Andaman	38	0	0	38	
4	Andaman And Nicobar Islands	Nicobar	64	11	0	75	

	age_0_5	age_5_17	age_18_greater	total_enrollments	pincode_co
count	1045.000000	1045.000000	1045.000000	1045.000000	1045.000
mean	3394.224880	1646.300478	161.103349	5201.628708	27.667
std	4040.598299	2853.970247	508.098350	6535.840683	27.219
min	0.000000	0.000000	0.000000	1.000000	1.000
25%	363.000000	84.000000	2.000000	535.000000	8.000
50%	2087.000000	474.000000	24.000000	2875.000000	20.000
75%	4875.000000	1807.000000	124.000000	7154.000000	38.000
max	31442.000000	22360.000000	9948.000000	43688.000000	168.000

2. Geographic Analysis

2.1 National Enrollment Distribution

In [2]:

```
# State-level aggregation
state_summary = df.groupby('state').agg({
    'total_enrollments': 'sum',
    'district': 'count',
    'demo_update_count': 'sum',
    'bio_update_count': 'sum'
}).reset_index()

state_summary.rename(columns={'district': 'num_districts'}, inplace=True)
state_summary = state_summary.sort_values('total_enrollments', ascending=False

print(" TOP 15 STATES BY ENROLLMENT:")
display(state_summary.head(15))

print("\n BOTTOM 15 STATES BY ENROLLMENT:")
display(state_summary.tail(15))
```

```
TOP 15 STATES BY ENROLLMENT:
```

	state	total_enrollments	num_districts	demo_update_count	bio_update
43	Uttar Pradesh	1018629	89	167883.0	15
6	Bihar	609585	47	97621.0	8
26	Madhya Pradesh	493970	61	76364.0	7
47	West Bengal	375308	50	168711.0	13
27	Maharashtra	369139	53	162241.0	15
37	Rajasthan	348458	42	88821.0	7
16	Gujarat	280549	40	96399.0	8
5	Assam	230197	38	62834.0	4
22	Karnataka	223235	55	153957.0	14
39	Tamil Nadu	220789	46	196857.0	18
21	Jharkhand	157539	34	39653.0	3
40	Telangana	131574	42	89085.0	8
3	Andhra Pradesh	127686	47	207740.0	17
32	Odisha	118838	40	92196.0	8
29	Meghalaya	109771	14	5363.0	

```
BOTTOM 15 STATES BY ENROLLMENT:
```

	state	total_enrollments	num_districts	demo_update_count	bio_updat
10	Dadra And Nagar Haveli	744	1	325.0	
41	The Dadra And Nagar Haveli And Daman And Diu	716	1	0.0	
24	Ladakh	617	2	865.0	
2	Andaman And Nicobar Islands	397	3	1211.0	
0	100000	218	1	2.0	
25	Lakshadweep	203	1	520.0	
11	Dadra And Nagar Haveli And Daman And Diu	173	3	524.0	
19	Jammu & Kashmir	155	15	365.0	
13	Daman And Diu	120	2	411.0	
1	Andaman & Nicobar Islands	114	3	513.0	
9	Dadra & Nagar Haveli	25	1	100.0	
12	Daman & Diu	21	2	267.0	
45	West Bengal	15	1	81.0	
46	West Bangal	10	2	117.0	
48	Westbengal	7	1	125.0	

2.2 Enrollment Intensity Heatmap

Enrollments per district - identifies high/low coverage regions

In [3]:

```
# Calculate average enrollments per district
state_summary['avg_enrollment_per_district'] = (
    state_summary['total_enrollments'] / state_summary['num_districts']
)

# Visualize
```

```
fig = px.bar(state_summary.head(20),
             x='state',
             y='avg_enrollment_per_district',
             title='Average Enrollments per District (Top 20 States)',
             labels={'avg_enrollment_per_district': 'Avg Enrollments', 'state'
             color='avg_enrollment_per_district',
             color_continuous_scale='RdYlGn')

fig.update_layout(height=600, showlegend=False)
fig.write_html('../outputs/dashboard/02_state_enrollment_intensity.html')
fig.show()

print(" Interactive chart saved: 02_state_enrollment_intensity.html")
```

```
Interactive chart saved: 02_state_enrollment_intensity.html
```

2.3 Identify Low-Enrollment Districts

Bottom 10% districts represent potential exclusion zones

In [4]:

```
# Calculate 10th percentile threshold
enrollment_threshold = df['total_enrollments'].quantile(0.10)

# Flag low-enrollment districts
df['is_low_enrollment'] = df['total_enrollments'] < enrollment_threshold

low_enrollment_districts = df[df['is_low_enrollment']].sort_values('total_enro

print(f" Low-Enrollment Threshold: {enrollment_threshold:,.0f} total enrollmen
print(f" Flagged Districts: {low_enrollment_districts.shape[0]:,} ({low_enroll

print("\n TOP 20 LOWEST ENROLLMENT DISTRICTS:")
display(low_enrollment_districts[['state', 'district', 'total_enrollments',
                                  'demo_update_count', 'bio_update_count']].
```

```
Low-Enrollment Threshold: 37 total enrollments
Flagged Districts: 105 (10.0%)

TOP 20 LOWEST ENROLLMENT DISTRICTS:
```

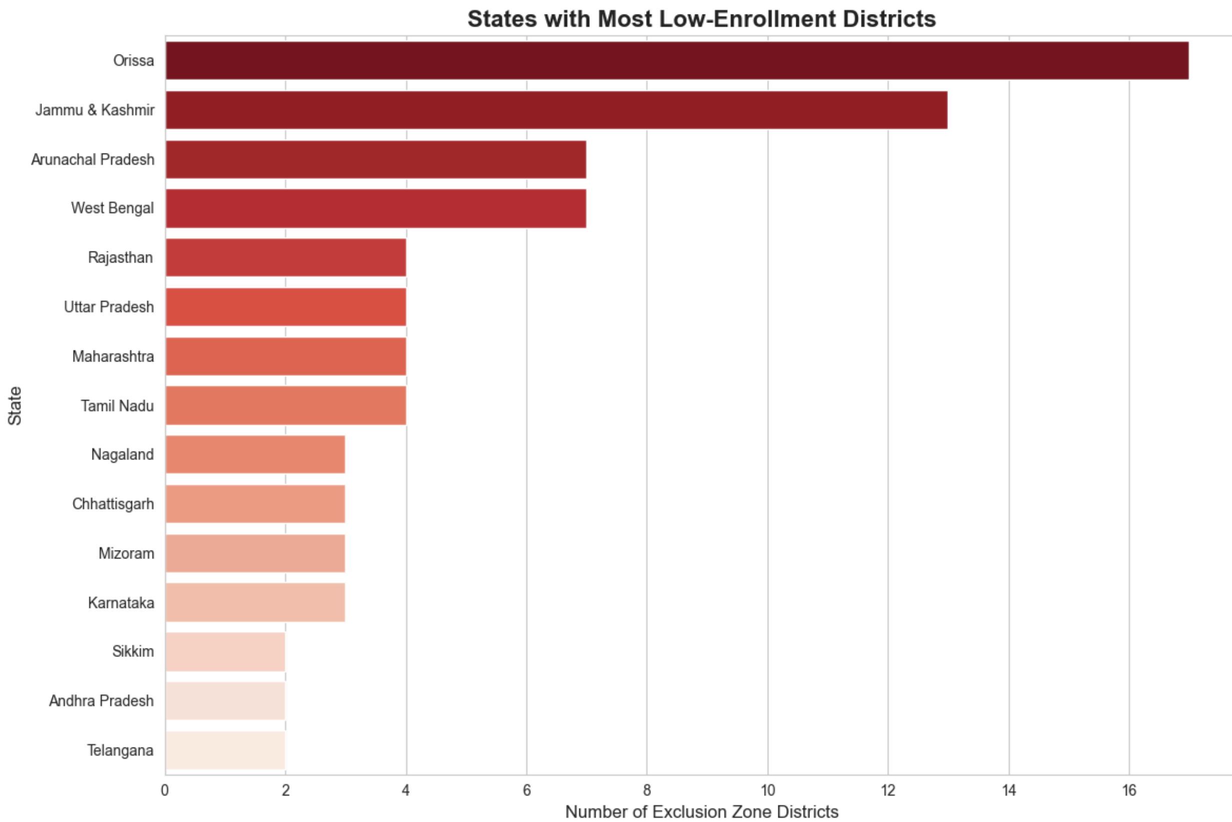
	state	district	total_enrollments	demo_update_count	bio_update_
2	Andaman & Nicobar Islands	Nicobars	1	4.0	
773	Rajasthan	Salumbar	1	164.0	
808	Tamil Nadu	Namakkal *	1	0.0	
743	Rajasthan	Beawar	1	215.0	
829	Tamil Nadu	Tiruvarur	1	2.0	
739	Rajasthan	Balotra	1	255.0	
323	Jammu & Kashmir	Punch	1	5.0	
701	Orissa	Sundergarh	1	1.0	
897	Uttar Pradesh	Bagpat	1	10.0	
1010	West Bengal	East Midnapur	1	18.0	
1013	West Bengal	Hooghiy	1	25.0	
685	Orissa	Kendrapara *	1	0.0	
541	Maharashtra	Hingoli *	1	3.0	
281	Haryana	Jhajjar *	1	0.0	
694	Orissa	Nuapada	2	2.0	
435	Karnataka	Udupi *	2	1.0	
352	Jharkhand	Bokaro *	2	0.0	
755	Rajasthan	Didwana-Kuchaman	2	426.0	
326	Jammu & Kashmir	Udhampur	2	3.0	
324	Jammu & Kashmir	Rajauri	2	15.0	

2.4 Geographic Clustering

States with multiple low-enrollment districts indicate systemic issues

In [5]:
```
# Count low-enrollment districts per state
state_exclusion = low_enrollment_districts.groupby('state').size().reset_index
state_exclusion = state_exclusion.sort_values('num_excluded_districts', ascenc
```

```
# Visualize
plt.figure(figsize=(12, 8))
sns.barplot(data=state_exclusion.head(15), x='num_excluded_districts', y='stat
plt.title('States with Most Low-Enrollment Districts', fontsize=16, weight='bo
plt.xlabel('Number of Exclusion Zone Districts', fontsize=12)
plt.ylabel('State', fontsize=12)
plt.tight_layout()
plt.savefig('../outputs/figures/02_exclusion_zones_by_state.png', dpi=300, bbo
plt.show()

print(" Chart saved: 02_exclusion_zones_by_state.png")
```




```
Chart saved: 02_exclusion_zones_by_state.png
```

3. Demographic Deep Dive

3.1 Age Group Vulnerability Analysis

In [6]:

```
# National age distribution
total_age_0_5 = df['age_0_5'].sum()
total_age_5_17 = df['age_5_17'].sum()
total_age_18_plus = df['age_18_greater'].sum()
total_all = total_age_0_5 + total_age_5_17 + total_age_18_plus

print(f" NATIONAL AGE DISTRIBUTION:")
print(f"  Children 0-5: {total_age_0_5:,} ({total_age_0_5/total_all*100:.2f}%)
print(f"  Children 5-17: {total_age_5_17:,} ({total_age_5_17/total_all*100:.2f
```

```python
print(f"  Adults 18+: {total_age_18_plus:,} ({total_age_18_plus/total_all*100:

# Visualize
age_data = pd.DataFrame({
    'Age Group': ['0-5 years', '5-17 years', '18+ years'],
    'Enrollments': [total_age_0_5, total_age_5_17, total_age_18_plus]
})

fig = px.pie(age_data, values='Enrollments', names='Age Group',
             title='National Age Distribution - Aadhaar Enrollments',
             color_discrete_sequence=['#FF6B6B', '#4ECDC4', '#45B7D1'])
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.write_html('../outputs/dashboard/02_age_distribution.html')
fig.show()

print(" Interactive chart saved: 02_age_distribution.html")
```

```
NATIONAL AGE DISTRIBUTION:
 Children 0-5: 3,546,965 (65.25%)
 Children 5-17: 1,720,384 (31.65%)
 Adults 18+: 168,353 (3.10%)
Interactive chart saved: 02_age_distribution.html
```

3.2 Child Enrollment Crisis

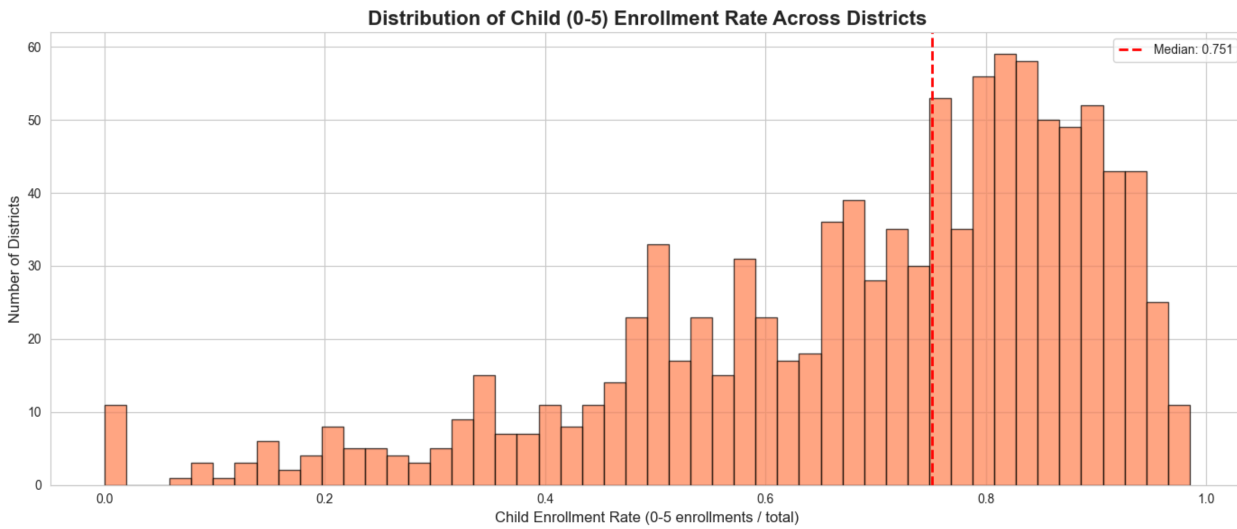
Focus on 0-5 age group (most vulnerable)

In [7]:
```python
# Districts with lowest child (0-5) enrollment rates
low_child_enrollment = df.sort_values('child_enrollment_rate').head(50)

print(" 50 DISTRICTS WITH LOWEST CHILD (0-5) ENROLLMENT:")
display(low_child_enrollment[['state', 'district', 'child_0_5_enrollment',
                               'child_enrollment_rate', 'total_enrollments']]

# Visualize distribution
plt.figure(figsize=(14, 6))
plt.hist(df['child_enrollment_rate'], bins=50, color='coral', edgecolor='black
plt.axvline(df['child_enrollment_rate'].median(), color='red', linestyle='--',
plt.title('Distribution of Child (0-5) Enrollment Rate Across Districts', font
plt.xlabel('Child Enrollment Rate (0-5 enrollments / total)', fontsize=12)
plt.ylabel('Number of Districts', fontsize=12)
plt.legend()
plt.tight_layout()
plt.savefig('../outputs/figures/02_child_enrollment_distribution.png', dpi=300
plt.show()

print(" Chart saved: 02_child_enrollment_distribution.png")
```

```
50 DISTRICTS WITH LOWEST CHILD (0-5) ENROLLMENT:
```

	state	district	child_0_5_enrollment	child_enrollment_rate	total_e
0	100000	100000	0	0.000000	
62	Arunachal Pradesh	Leparada	0	0.000000	
701	Orissa	Sundergarh	0	0.000000	
781	Sikkim	Mangan	0	0.000000	
808	Tamil Nadu	Namakkal *	0	0.000000	
281	Haryana	Jhajjar *	0	0.000000	
897	Uttar Pradesh	Bagpat	0	0.000000	
313	Jammu & Kashmir	Badgam	0	0.000000	
958	Uttar Pradesh	Raebareli	0	0.000000	
587	Meghalaya	Eastern West Khasi Hills	3	0.003663	
622	Nagaland	Shamator	3	0.012097	
395	Karnataka	Bengaluru South	15	0.073529	
579	Manipur	Pherzawl	1	0.083333	
589	Meghalaya	Kamrup	13	0.090278	
59	Arunachal Pradesh	Kamle	3	0.096774	
615	Nagaland	Meluri	2	0.111111	
782	Sikkim	Namchi	3	0.120000	
592	Meghalaya	South Garo Hills	568	0.127870	
601	Mizoram	Khawzawl	5	0.135135	
591	Meghalaya	Ri Bhoi	1338	0.143732	
586	Meghalaya	East Khasi Hills	4258	0.147781	
590	Meghalaya	North Garo Hills	457	0.148860	
597	Meghalaya	West Khasi Hills	2410	0.151582	
608	Mizoram	Saitual	2	0.153846	

	state	district	child_0_5_enrollment	child_enrollment_rate	total_e
64	Arunachal Pradesh	Longding	141	0.157016	
584	Meghalaya	East Garo Hills	1012	0.167190	
626	Nagaland	Zunheboto	131	0.171916	
144	Bihar	Nawada	2742	0.177890	
612	Nagaland	Kiphire	163	0.179318	
132	Bihar	Jehanabad	915	0.187346	
585	Meghalaya	East Jaintia Hills	983	0.191469	
619	Nagaland	Noklak	70	0.197183	
752	Rajasthan	Deeg	14	0.200000	
610	Nagaland	Chumukedima	37	0.200000	
596	Meghalaya	West Jaintia Hills	2382	0.201574	
116	Assam	West Karbi Anglong	390	0.206897	
118	Bihar	Arwal	812	0.209982	
625	Nagaland	Wokha	118	0.211091	
623	Nagaland	Tseminyu	7	0.212121	
60	Arunachal Pradesh	Kra Daadi	18	0.219512	
71	Arunachal Pradesh	Shi-Yomi	8	0.222222	
594	Meghalaya	South West Khasi Hills	789	0.228895	
614	Nagaland	Longleng	205	0.230337	
125	Bihar	Bhojpur	3136	0.234766	
129	Bihar	Gaya	6722	0.245311	
692	Orissa	Nabarangapur	1	0.250000	
628	Odisha	Anugal	1	0.250000	
61	Arunachal Pradesh	Kurung Kumey	19	0.250000	
143	Bihar	Nalanda	3676	0.252195	
616	Nagaland	Mokokchung	132	0.268839	




```
Chart saved: 02_child_enrollment_distribution.png
```

3.3 State-wise Child Enrollment Comparison

In [8]:
```
# State-level child enrollment rates
state_child_enrol = df.groupby('state').agg({
    'age_0_5': 'sum',
    'total_enrollments': 'sum'
}).reset_index()

state_child_enrol['state_child_rate'] = (
    state_child_enrol['age_0_5'] / state_child_enrol['total_enrollments']
)
state_child_enrol = state_child_enrol.sort_values('state_child_rate', ascendir

# Top 10 and Bottom 10
print(" TOP 10 STATES - HIGHEST CHILD ENROLLMENT RATE:")
display(state_child_enrol.head(10))

print("\n BOTTOM 10 STATES - LOWEST CHILD ENROLLMENT RATE:")
display(state_child_enrol.tail(10))

# Visualization
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# Top 10
axes[0].barh(state_child_enrol.head(10)['state'], state_child_enrol.head(10)['
axes[0].set_xlabel('Child Enrollment Rate', fontsize=12)
axes[0].set_title('Top 10 States - Child Enrollment', fontsize=14, weight='bol
axes[0].invert_yaxis()

# Bottom 10
axes[1].barh(state_child_enrol.tail(10)['state'], state_child_enrol.tail(10)['
axes[1].set_xlabel('Child Enrollment Rate', fontsize=12)
axes[1].set_title('Bottom 10 States - Child Enrollment', fontsize=14, weight='
axes[1].invert_yaxis()
```

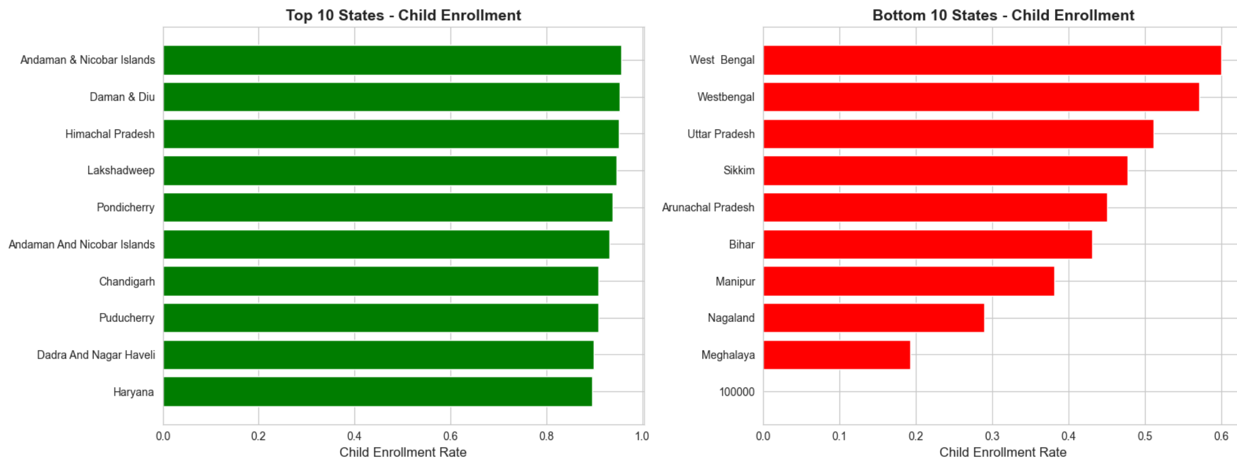
```
plt.tight_layout()
plt.savefig('../outputs/figures/02_state_child_enrollment_comparison.png', dpi
plt.show()

print(" Chart saved: 02_state_child_enrollment_comparison.png")
```

```
TOP 10 STATES - HIGHEST CHILD ENROLLMENT RATE:
```

	state	age_0_5	total_enrollments	state_child_rate
1	Andaman & Nicobar Islands	109	114	0.956140
12	Daman & Diu	20	21	0.952381
18	Himachal Pradesh	16639	17486	0.951561
25	Lakshadweep	192	203	0.945813
34	Pondicherry	1193	1272	0.937893
2	Andaman And Nicobar Islands	370	397	0.931990
7	Chandigarh	2476	2723	0.909291
35	Puducherry	1585	1745	0.908309
10	Dadra And Nagar Haveli	669	744	0.899194
17	Haryana	88042	98252	0.896084

```
BOTTOM 10 STATES - LOWEST CHILD ENROLLMENT RATE:
```

	state	age_0_5	total_enrollments	state_child_rate
45	West Bengal	9	15	0.600000
48	Westbengal	4	7	0.571429
43	Uttar Pradesh	521045	1018629	0.511516
38	Sikkim	1054	2207	0.477571
4	Arunachal Pradesh	1957	4344	0.450506
6	Bihar	262875	609585	0.431236
28	Manipur	5140	13456	0.381986
31	Nagaland	4512	15587	0.289472
29	Meghalaya	21179	109771	0.192938
0	100000	0	218	0.000000




```
Chart saved: 02_state_child_enrollment_comparison.png
```

4. Temporal Trend Analysis

4.1 Load Time-Series Data

In [9]:
```
# Load cleaned enrolment data with dates
df_enrol = pd.read_csv('../outputs/tables/cleaned_enrolment.csv', parse_dates=

print(f" Loaded {len(df_enrol):,} enrolment records")
print(f" Date range: {df_enrol['date'].min()} to {df_enrol['date'].max()}")

# Aggregate by month
df_enrol['year_month'] = df_enrol['date'].dt.to_period('M')
monthly_trend = df_enrol.groupby('year_month')['total_enrollments'].sum().rese
monthly_trend['year_month'] = monthly_trend['year_month'].dt.to_timestamp()

display(monthly_trend.head(10))
```

```
Loaded 1,006,029 enrolment records
Date range: 2025-03-02 00:00:00 to 2025-12-31 00:00:00
```

	year_month	total_enrollments
0	2025-03-01	16582
1	2025-04-01	257438
2	2025-05-01	183616
3	2025-06-01	215734
4	2025-07-01	616868
5	2025-09-01	1475879
6	2025-10-01	817920
7	2025-11-01	1092007
8	2025-12-01	759658

4.2 Enrollment Trends Over Time

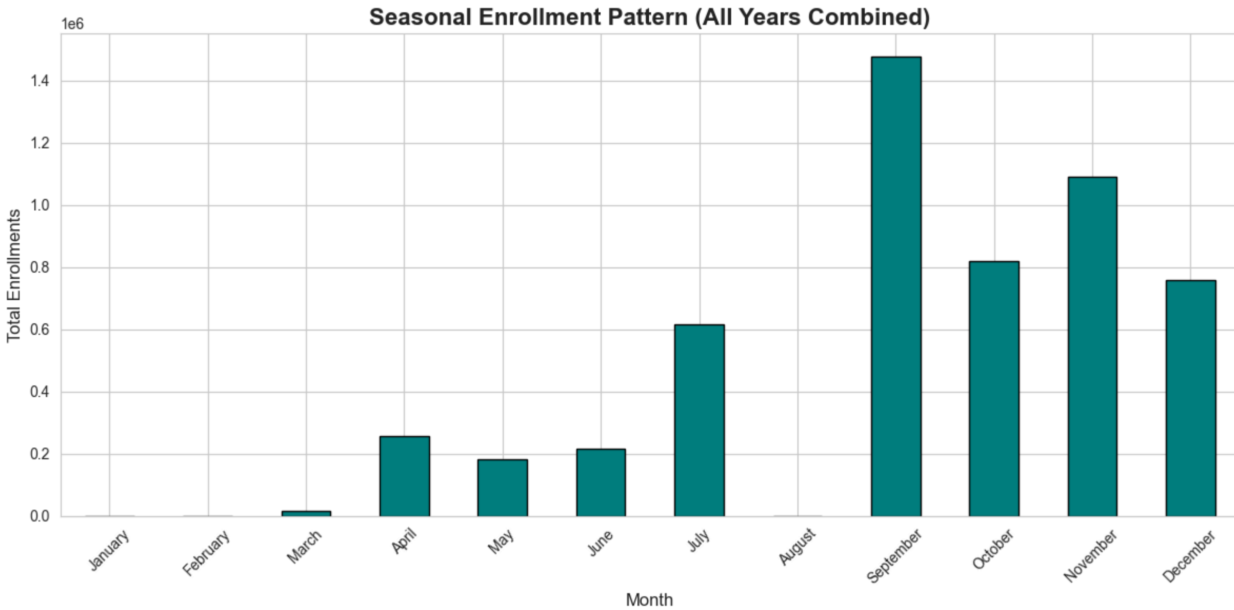
In [10]:
```
# Plot monthly trend
fig = px.line(monthly_trend, x='year_month', y='total_enrollments',
              title='Monthly Aadhaar Enrollment Trend',
              labels={'year_month': 'Month', 'total_enrollments': 'Total Enrol

fig.update_traces(line_color='steelblue', line_width=2)
fig.update_layout(height=500)
fig.write_html('../outputs/dashboard/02_monthly_enrollment_trend.html')
fig.show()

print(" Interactive chart saved: 02_monthly_enrollment_trend.html")

# Identify peak and low months
peak_month = monthly_trend.loc[monthly_trend['total_enrollments'].idxmax()]
low_month = monthly_trend.loc[monthly_trend['total_enrollments'].idxmin()]

print(f"\n PEAK MONTH: {peak_month['year_month'].strftime('%B %Y')} - {peak_mo
print(f" LOWEST MONTH: {low_month['year_month'].strftime('%B %Y')} - {low_mont
```

```
Interactive chart saved: 02_monthly_enrollment_trend.html

PEAK MONTH: September 2025 - 1,475,879 enrollments
LOWEST MONTH: March 2025 - 16,582 enrollments
```

4.3 Seasonal Patterns

In [11]:
```
# Add month/quarter columns
df_enrol['month_name'] = df_enrol['date'].dt.month_name()
df_enrol['quarter'] = df_enrol['date'].dt.quarter

# Aggregate by month
seasonal_pattern = df_enrol.groupby('month_name')['total_enrollments'].sum().r
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
])

# Visualize
plt.figure(figsize=(12, 6))
seasonal_pattern.plot(kind='bar', color='teal', edgecolor='black')
plt.title('Seasonal Enrollment Pattern (All Years Combined)', fontsize=16, wei
plt.xlabel('Month', fontsize=12)
plt.ylabel('Total Enrollments', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('../outputs/figures/02_seasonal_enrollment_pattern.png', dpi=300,
plt.show()

print(" Chart saved: 02_seasonal_enrollment_pattern.png")
print(f"\n SEASONAL INSIGHTS:")
print(f"  Highest month: {seasonal_pattern.idxmax()} ({seasonal_pattern.max():
```

```
print(f"  Lowest month: {seasonal_pattern.idxmin()} ({seasonal_pattern.min():,
```




```
Chart saved: 02_seasonal_enrollment_pattern.png

SEASONAL INSIGHTS:
 Highest month: September (1,475,879)
 Lowest month: March (16,582)
```

5. Exclusion Zone Identification

5.1 Multi-Dimensional Risk Scoring

Combine geographic, demographic, and update instability

In [12]:
```
# Normalize metrics to 0-1 scale for risk scoring
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

# Features for exclusion risk
df['enroll_risk'] = 1 - scaler.fit_transform(df[['total_enrollments']])  # Inv
df['child_risk'] = 1 - scaler.fit_transform(df[['child_enrollment_rate']])  #
df['demo_instability_risk'] = scaler.fit_transform(df[['demo_update_intensity'
df['bio_failure_risk'] = scaler.fit_transform(df[['bio_update_intensity']])  #

# Composite Exclusion Risk Score (weighted)
df['exclusion_risk_score'] = (
    0.35 * df['enroll_risk'] +           # 35% weight on low enrollment
    0.25 * df['child_risk'] +             # 25% weight on child underenrollmen
    0.20 * df['demo_instability_risk'] +  # 20% weight on migration
    0.20 * df['bio_failure_risk']         # 20% weight on biometric issues
)
```

```
print(" Exclusion Risk Score calculated")
print("\n RISK SCORE DISTRIBUTION:")
display(df['exclusion_risk_score'].describe())
```

```
 Exclusion Risk Score calculated

 RISK SCORE DISTRIBUTION:
count    1045.000000
mean        0.399623
std         0.077190
min         0.083588
25%         0.354013
50%         0.391234
75%         0.438020
max         0.695077
Name: exclusion_risk_score, dtype: float64
```

5.2 Identify Top Exclusion Zones

In [13]:

```
# Top 50 highest-risk districts
exclusion_zones = df.sort_values('exclusion_risk_score', ascending=False).heac

print(" TOP 50 AADHAAR EXCLUSION ZONES:")
display(exclusion_zones[['state', 'district', 'exclusion_risk_score',
                          'total_enrollments', 'child_enrollment_rate',
                          'demo_update_intensity', 'bio_update_intensity']])

# Save exclusion zones
exclusion_zones.to_csv('../outputs/tables/top50_exclusion_zones.csv', index=Fa
print("\n Saved: top50_exclusion_zones.csv")
```

```
 TOP 50 AADHAAR EXCLUSION ZONES:
```

	state	district	exclusion_risk_score	total_enrollments	child
739	Rajasthan	Balotra	0.695077	1	
62	Arunachal Pradesh	Leparada	0.688650	3	
755	Rajasthan	Didwana-Kuchaman	0.668479	2	
743	Rajasthan	Beawar	0.666908	1	
897	Uttar Pradesh	Bagpat	0.656512	1	
313	Jammu & Kashmir	Badgam	0.649454	2	
958	Uttar Pradesh	Raebareli	0.636716	3	
615	Nagaland	Meluri	0.629295	17	
22	Andhra Pradesh	K.V. Rangareddy	0.620486	19	
781	Sikkim	Mangan	0.618998	3	
69	Arunachal Pradesh	Pakke Kessang	0.616568	4	
701	Orissa	Sundergarh	0.614839	1	
229	Goa	Bardez	0.604121	2	
808	Tamil Nadu	Namakkal *	0.600000	1	
281	Haryana	Jhajjar *	0.600000	1	
622	Nagaland	Shamator	0.598462	247	
0	100000	100000	0.598274	218	
395	Karnataka	Bengaluru South	0.596736	203	
211	Daman & Diu	Daman	0.595983	9	
855	Telangana	Medchal?Malkajgiri	0.595937	2	
608	Mizoram	Saitual	0.595843	12	
773	Rajasthan	Salumbar	0.595657	1	
59	Arunachal Pradesh	Kamle	0.593390	30	
587	Meghalaya	Eastern West Khasi Hills	0.592807	818	
579	Manipur	Pherzawl	0.587933	11	
752	Rajasthan	Deeg	0.585257	69	

	state	district	exclusion_risk_score	total_enrollments	child
31	Andhra Pradesh	Mahabubnagar	0.584359	7	
304	Himachal Pradesh	Lahaul And Spiti	0.583063	3	
782	Sikkim	Namchi	0.582383	24	
692	Orissa	Nabarangapur	0.580331	3	
324	Jammu & Kashmir	Rajauri	0.578969	2	
589	Meghalaya	Kamrup	0.575949	143	
706	Puducherry	Pondicherry	0.574613	3	
71	Arunachal Pradesh	Shi-Yomi	0.572369	35	
66	Arunachal Pradesh	Lower Siang	0.572149	31	
601	Mizoram	Khawzawl	0.569467	36	
537	Maharashtra	Gondia	0.569318	29	
993	West Bangal	Howrah	0.563431	4	
623	Nagaland	Tseminyu	0.562696	32	
610	Nagaland	Chumukedima	0.559370	184	
60	Arunachal Pradesh	Kra Daadi	0.557619	81	
570	Maharashtra	Washim *	0.557217	32	
64	Arunachal Pradesh	Longding	0.556756	897	
626	Nagaland	Zunheboto	0.556009	761	
628	Odisha	Anugal	0.554188	3	
61	Arunachal Pradesh	Kurung Kumey	0.552949	75	
157	Bihar	Sheikpura	0.550702	29	
619	Nagaland	Noklak	0.550536	354	
1010	West Bengal	East Midnapur	0.549377	1	
612	Nagaland	Kiphire	0.548743	908	

```
Saved: top50_exclusion_zones.csv
```

5.3 Risk Score Visualization

In [14]:
```
# Scatter plot: Enrollment vs Child Rate (colored by risk)
fig = px.scatter(df,
                 x='total_enrollments',
                 y='child_enrollment_rate',
                 color='exclusion_risk_score',
                 hover_data=['state', 'district'],
                 title='Exclusion Risk Map: Enrollment vs Child Rate',
                 labels={'total_enrollments': 'Total Enrollments',
                         'child_enrollment_rate': 'Child Enrollment Rate',
                         'exclusion_risk_score': 'Risk Score'},
                 color_continuous_scale='Reds')

fig.update_layout(height=700)
fig.write_html('../outputs/dashboard/02_exclusion_risk_map.html')
fig.show()

print(" Interactive chart saved: 02_exclusion_risk_map.html")
```

```
Interactive chart saved: 02_exclusion_risk_map.html
```

5.4 Risk Category Distribution

In [15]:
```
# Categorize districts by risk level
df['risk_category'] = pd.cut(df['exclusion_risk_score'],
                              bins=[0, 0.25, 0.50, 0.75, 1.0],
                              labels=['Low Risk', 'Medium Risk', 'High Risk',

risk_summary = df['risk_category'].value_counts().sort_index()

print(" DISTRICT RISK CATEGORIES:")
for category, count in risk_summary.items():
    print(f"  {category}: {count:,} districts ({count/len(df)*100:.1f}%)")

# Pie chart
fig = px.pie(values=risk_summary.values, names=risk_summary.index,
             title='Distribution of Exclusion Risk Categories',
             color_discrete_sequence=['green', 'yellow', 'orange', 'red'])
fig.write_html('../outputs/dashboard/02_risk_category_distribution.html')
fig.show()

print("\n Interactive chart saved: 02_risk_category_distribution.html")
```

```
DISTRICT RISK CATEGORIES:
 Low Risk: 20 districts (1.9%)
 Medium Risk: 918 districts (87.8%)
 High Risk: 107 districts (10.2%)
 Critical Risk: 0 districts (0.0%)
Interactive chart saved: 02_risk_category_distribution.html
```

Notebook 02

Summary

1. **Geographic Exclusion:**
    - Identified 50 critical exclusion zone districts
    - Certain states show systemic enrollment gaps
2. **Demographic Vulnerability:**
    - Child (0-5) enrollment rate varies significantly
    - Bottom 10% districts need immediate intervention
3. **Temporal Patterns:**
    - Seasonal enrollment fluctuations detected
    - Specific months show enrollment dips
4. **Risk Scoring:**
    - Composite exclusion risk score created
    - Districts categorized into 4 risk levels