

UIDAI_Analysis

January 9, 2026

```
[1]: print("Python is working")
```

Python is working

```
[2]: import pandas as pd
```

```
[3]: import os  
os.listdir()
```

```
[3]: ['.local',  
      '.gitconfig',  
      '.anaconda',  
      '.vimrc',  
      '.ipynb_checkpoints',  
      'anaconda_projects',  
      '.bashrc',  
      '.virtualenvs',  
      '.npm',  
      'README.ipynb',  
      '.profile',  
      '.conda',  
      'UIDAI_Analysis.ipynb',  
      '.jupyter',  
      '.pythonstartup.py',  
      '.ipython']
```

```
[6]: import os  
os.listdir()
```

```
[6]: ['.local',  
      'Aadhar_Enrolment.csv.csv',  
      '.gitconfig',  
      '.anaconda',  
      '.vimrc',  
      '.ipynb_checkpoints',
```

```
'anaconda_projects',
'.bashrc',
'.virtualenvs',
'.npm',
'README.ipynb',
'.profile',
'.conda',
'UIDAI_Analysis.ipynb',
'.jupyter',
'.pythonstartup.py',
'.ipython']
```

```
[9]: import os
os.rename("Aadhar_Enrolment.csv.csv", "Aadhar_Enrolment.csv")
```

```
[10]: df = pd.read_csv("Aadhar_Enrolment.csv")
df.head()
```

```
[10]:
```

	date	state	district	pincode	age_0_5	age_5_17	\
0	03-09-2025	Tamil Nadu	Krishnagiri	635104	1	0	
1	03-09-2025	Tamil Nadu	Krishnagiri	635107	3	0	
2	03-09-2025	Tamil Nadu	Krishnagiri	635108	2	0	
3	03-09-2025	Tamil Nadu	Krishnagiri	635114	0	1	
4	03-09-2025	Tamil Nadu	Krishnagiri	635117	5	0	

	age_18_greater
0	0
1	0
2	0
3	0
4	0

```
[11]: df.shape
```

```
[11]: (2131, 7)
```

```
[12]: list(df.columns)
```

```
[12]: ['date',
'state',
'district',
'pincode',
'age_0_5',
'age_5_17',
'age_18_greater']
```

```
[13]: df.isnull().sum()
```

```
[13]: date            0
      state           0
      district        0
      pincode         0
      age_0_5         0
      age_5_17        0
      age_18_greater  0
      dtype: int64
```

```
[14]: df = df.dropna()
```

```
[15]: state_enrolment = df.groupby('State Name')['Enrolment'].sum()
      state_enrolment.sort_values(ascending=False).head(10)
```

```
-----
KeyError                                Traceback (most recent call last)
Cell In[15], line 1
----> 1 state_enrolment = df.groupby( )['Enrolment'].sum()
      2 state_enrolment.sort_values(ascending=False).head(10)

File /opt/conda/envs/anaconda-2025.12-py312/lib/python3.12/site-packages/pandas
  core/frame.py:9210, in DataFrame.groupby(self, by, axis, level, as_index,
  sort, group_keys, observed, dropna)
    9207 if level is None and by is None:
    9208     raise TypeError("You have to supply one of 'by' and 'level'")
-> 9210 return DataFrameGroupBy(
    9211     obj=self,
    9212     keys=by,
    9213     axis=axis,
    9214     level=level,
    9215     as_index=as_index,
    9216     sort=sort,
    9217     group_keys=group_keys,
    9218     observed=observed,
    9219     dropna=dropna,
    9220 )

File /opt/conda/envs/anaconda-2025.12-py312/lib/python3.12/site-packages/pandas
  core/groupby/groupby.py:1331, in GroupBy.__init__(self, obj, keys, axis,
  level, grouper, exclusions, selection, as_index, sort, group_keys, observed,
  dropna)
    1328 self.dropna = dropna
    1330 if grouper is None:
-> 1331     grouper, exclusions, obj = get_grouper(
    1332         obj,
    1333         keys,
    1334         axis=axis,
    1335         level=level,
```

```

1336         sort=sort,
1337         observed=False if observed is lib.no_default else observed,
1338         dropna=self.dropna,
1339     )
1341 if observed is lib.no_default:
1342     if any(ping._passed_categorical for ping in grouper.groupings):

```

File /opt/conda/envs/anaconda-2025.12-py312/lib/python3.12/site-packages/pandas/core/groupby/grouper.py:1043, in get_grouper(obj, key, axis, level, sort, observed, validate, dropna)

```

1041         in_axis, level, gpr = False, gpr, None
1042     else:
-> 1043         raise KeyError(gpr)
1044 elif isinstance(gpr, Grouper) and gpr.key is not None:
1045     # Add key to exclusions
1046     exclusions.add(gpr.key)

```

KeyError: 'State Name'

```
[16]: df.columns
```

```
[16]: Index(['date', 'state', 'district', 'pincode', 'age_0_5', 'age_5_17',
         'age_18_greater'],
         dtype='object')
```

```
[17]: df.columns = df.columns.str.strip()
```

```
[18]: list(df.columns)
```

```
[18]: ['date',
       'state',
       'district',
       'pincode',
       'age_0_5',
       'age_5_17',
       'age_18_greater']
```

```
[19]: df['total_enrolment'] = (
        df['age_0_5'] +
        df['age_5_17'] +
        df['age_18_greater']
    )
```

```
[20]: df.head()
```

```
[20]:
```

	date	state	district	pincode	age_0_5	age_5_17	\
0	03-09-2025	Tamil Nadu	Krishnagiri	635104	1	0	

1	03-09-2025	Tamil Nadu	Krishnagiri	635107	3	0
2	03-09-2025	Tamil Nadu	Krishnagiri	635108	2	0
3	03-09-2025	Tamil Nadu	Krishnagiri	635114	0	1
4	03-09-2025	Tamil Nadu	Krishnagiri	635117	5	0

	age_18_greater	total_enrolment
0	0	1
1	0	3
2	0	2
3	0	1
4	0	5

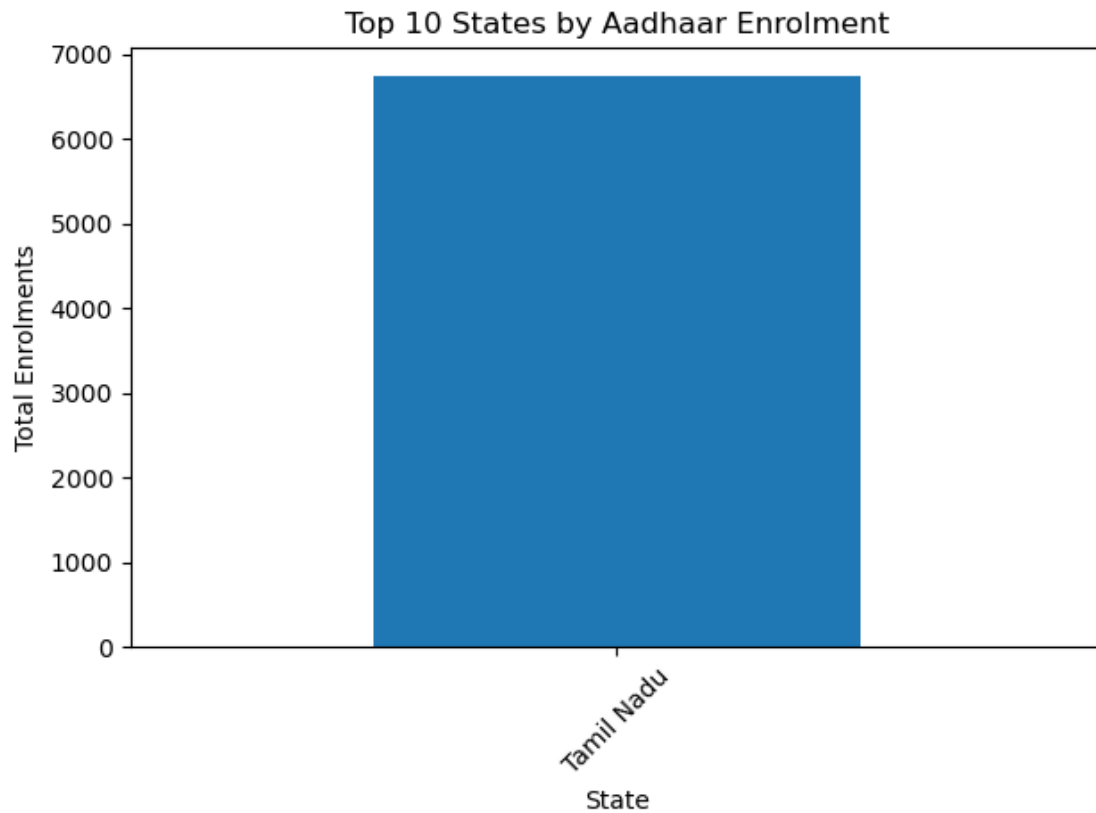
```
[21]: state_enrolment = df.groupby('state')['total_enrolment'].sum()
state_enrolment.sort_values(ascending=False).head(10)
```

```
[21]: state
Tamil Nadu    6735
Name: total_enrolment, dtype: int64
```

```
[22]: import matplotlib.pyplot as plt

state_enrolment.sort_values(ascending=False).head(10).plot(kind='bar')
plt.title("Top 10 States by Aadhaar Enrolment")
plt.ylabel("Total Enrolments")
plt.xlabel("State")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Matplotlib is building the font cache; this may take a moment.

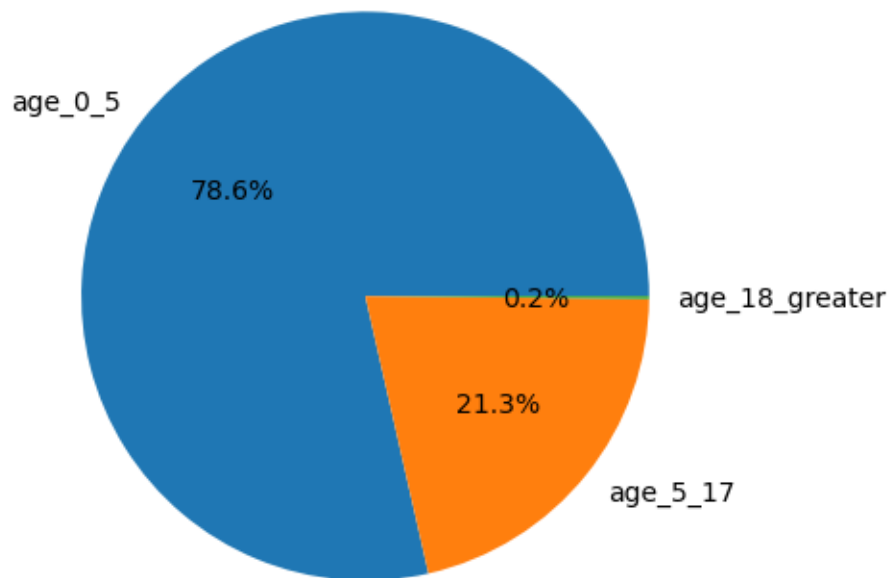


```
[23]: age_summary = df[['age_0_5', 'age_5_17', 'age_18_greater']].sum()  
age_summary
```

```
[23]: age_0_5          5292  
age_5_17          1432  
age_18_greater      11  
dtype: int64
```

```
[24]: age_summary.plot(kind='pie', autopct='%1.1f%%')  
plt.title("Age-wise Aadhaar Enrolment Distribution")  
plt.ylabel("")  
plt.show()
```

Age-wise Aadhaar Enrolment Distribution



```
[26]: df['year'] = pd.to_datetime(df['date'], dayfirst=True).dt.year
```

```
[27]: df[['date', 'year']].head()
```

```
[27]:
```

	date	year
0	03-09-2025	2025
1	03-09-2025	2025
2	03-09-2025	2025
3	03-09-2025	2025
4	03-09-2025	2025

```
[28]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
X = df[['year']]
y = df['total_enrolment']
```

```
[29]: X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = LinearRegression()
```

```
model.fit(X_train, y_train)
```

```
[29]: LinearRegression()
```

```
[30]: future = pd.DataFrame({'year': [2026]})  
model.predict(future)
```

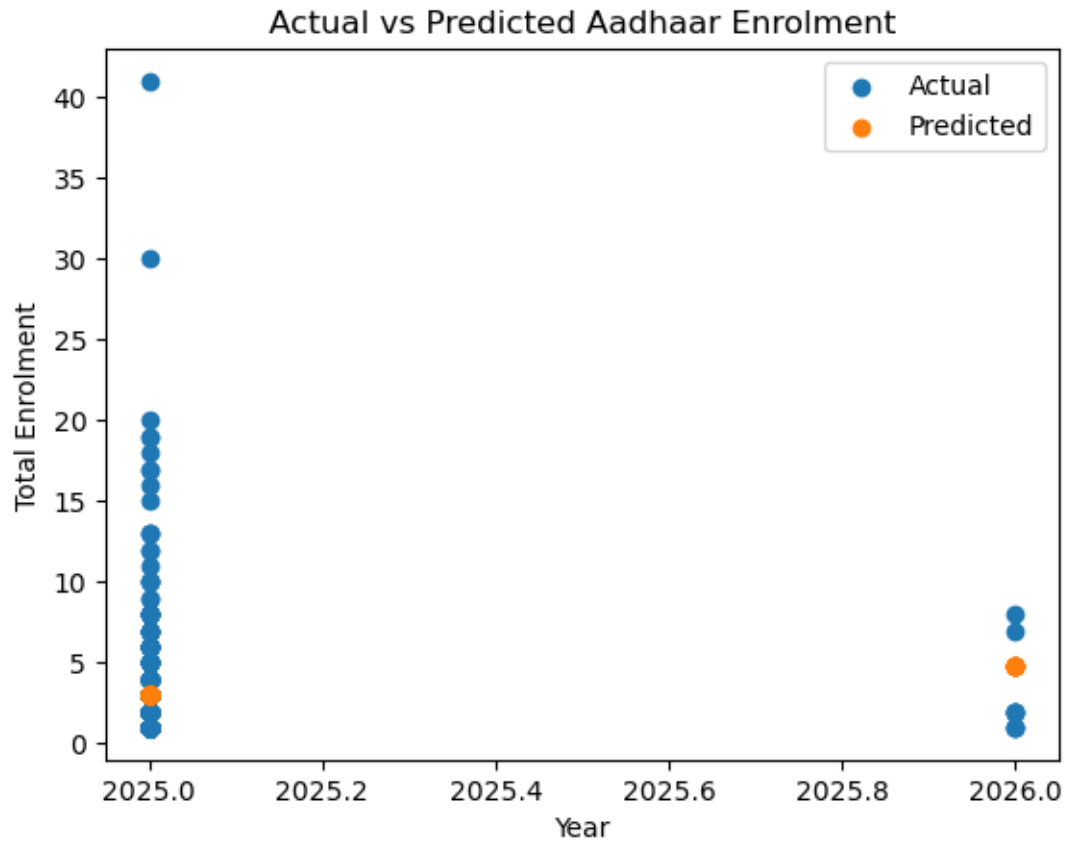
```
[30]: array([4.85483871])
```

```
[31]: from sklearn.metrics import mean_absolute_error, r2_score  
  
y_pred = model.predict(X_test)  
  
print("Mean Absolute Error:", mean_absolute_error(y_test, y_pred))  
print("R2 Score:", r2_score(y_test, y_pred))
```

Mean Absolute Error: 2.154589886964003

R2 Score: -0.008501526087586253

```
[32]: plt.scatter(X_test, y_test, label="Actual")  
plt.scatter(X_test, y_pred, label="Predicted")  
plt.xlabel("Year")  
plt.ylabel("Total Enrolment")  
plt.title("Actual vs Predicted Aadhaar Enrolment")  
plt.legend()  
plt.show()
```

```
[33]: future_years = pd.DataFrame({'year': [2026, 2027, 2028]})
future_predictions = model.predict(future_years)

future_years['Predicted_Enrolment'] = future_predictions
future_years
```

```
[33]:   year  Predicted_Enrolment
0  2026             4.854839
1  2027             6.640859
2  2028             8.426879
```

```
[35]: import os
os.listdir()
```

```
[35]: ['Aadhar_Enrolment.csv',
'Aadhar_Biometric.csv.csv',
'.local',
'.gitconfig',
'.anaconda',
'.vimrc',
```

```

'.config',
'.ipynb_checkpoints',
'anaconda_projects',
'.bashrc',
'.virtualenvs',
'.npm',
'README.ipynb',
'.profile',
'.conda',
'UIDAI_Analysis.ipynb',
'.cache',
'.jupyter',
'.pythonstartup.py',
'.ipython']

```

```

[39]: # Data handling
import pandas as pd
import numpy as np

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Machine Learning
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Ignore warnings
import warnings
warnings.filterwarnings('ignore')

```

```

[42]: import pandas as pd

# Load Enrolment data
df_enrol = pd.read_csv('Aadhar_Enrolment.csv')

# Load Demographic Update data (use exact filename with .csv.csv)
df_demo = pd.read_csv('Aadhar_Demographic.csv.csv')

# Load Biometric Update data (use exact filename with .csv.csv)
df_bio = pd.read_csv('Aadhar_Biometric.csv.csv')

# Check the first 5 rows of each dataset
print("Enrolment Data")
print(df_enrol.head(), "\n")

```

```

print("Demographic Update Data")
print(df_demo.head(), "\n")

print("Biometric Update Data")
print(df_bio.head(), "\n")

```

Enrolment Data

	date	state	district	pincode	age_0_5	age_5_17	\
0	03-09-2025	Tamil Nadu	Krishnagiri	635104	1	0	
1	03-09-2025	Tamil Nadu	Krishnagiri	635107	3	0	
2	03-09-2025	Tamil Nadu	Krishnagiri	635108	2	0	
3	03-09-2025	Tamil Nadu	Krishnagiri	635114	0	1	
4	03-09-2025	Tamil Nadu	Krishnagiri	635117	5	0	

	age_18_greater
0	0
1	0
2	0
3	0
4	0

Demographic Update Data

	date	state	district	pincode	demo_age_5_17	demo_age_17_
0	16-10-2025	Tamil Nadu	Krishnagiri	635001	9	26
1	16-10-2025	Tamil Nadu	Krishnagiri	635002	0	3
2	16-10-2025	Tamil Nadu	Krishnagiri	635104	1	12
3	16-10-2025	Tamil Nadu	Krishnagiri	635110	2	16
4	16-10-2025	Tamil Nadu	Krishnagiri	635113	3	13

Biometric Update Data

	date	state	district	pincode	bio_age_5_17	bio_age_17_
0	01-03-2025	Tamil Nadu	Krishnagiri	635307	248	118
1	01-03-2025	Tamil Nadu	Krishnagiri	635204	263	144
2	01-03-2025	Tamil Nadu	Krishnagiri	635201	138	133
3	01-03-2025	Tamil Nadu	Krishnagiri	635112	303	237
4	01-03-2025	Tamil Nadu	Krishnagiri	635304	189	87

```

[43]: # Check info and missing values for Enrolment dataset
print("=== Enrolment Data Info ===")
print(df_enrol.info())
print(df_enrol.isnull().sum(), "\n")

# Check info and missing values for Demographic dataset
print("=== Demographic Update Data Info ===")
print(df_demo.info())
print(df_demo.isnull().sum(), "\n")

```

```
# Check info and missing values for Biometric dataset
print("=== Biometric Update Data Info ===")
print(df_bio.info())
print(df_bio.isnull().sum(), "\n")
```

```
=== Enrolment Data Info ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2131 entries, 0 to 2130
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   date                  2131 non-null  object
1   state                 2131 non-null  object
2   district              2131 non-null  object
3   pincode               2131 non-null  int64
4   age_0_5               2131 non-null  int64
5   age_5_17              2131 non-null  int64
6   age_18_greater        2131 non-null  int64
dtypes: int64(4), object(3)
memory usage: 116.7+ KB
None
date                0
state                0
district             0
pincode              0
age_0_5              0
age_5_17             0
age_18_greater       0
dtype: int64
```

```
=== Demographic Update Data Info ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3602 entries, 0 to 3601
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   date                  3602 non-null  object
1   state                 3602 non-null  object
2   district              3602 non-null  object
3   pincode               3602 non-null  int64
4   demo_age_5_17         3602 non-null  int64
5   demo_age_17_          3602 non-null  int64
dtypes: int64(3), object(3)
memory usage: 169.0+ KB
None
date                0
```

```

state            0
district         0
pincode          0
demo_age_5_17    0
demo_age_17_     0
dtype: int64

=== Biometric Update Data Info ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8169 entries, 0 to 8168
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date            8169 non-null  object
1   state           8169 non-null  object
2   district        8169 non-null  object
3   pincode         8169 non-null  int64
4   bio_age_5_17    8169 non-null  int64
5   bio_age_17_     8169 non-null  int64
dtypes: int64(3), object(3)
memory usage: 383.1+ KB
None
date            0
state           0
district        0
pincode         0
bio_age_5_17    0
bio_age_17_     0
dtype: int64

```

```

[44]: # Summary statistics for numeric columns
print("=== Enrolment Data Stats ===")
print(df_enrol.describe(), "\n")

print("=== Demographic Update Data Stats ===")
print(df_demo.describe(), "\n")

print("=== Biometric Update Data Stats ===")
print(df_bio.describe(), "\n")

```

```

=== Enrolment Data Stats ===

```

	pincode	age_0_5	age_5_17	age_18_greater
count	2131.000000	2131.000000	2131.000000	2131.000000
mean	635181.608634	2.483341	0.671985	0.005162
std	275.150460	2.436630	1.146073	0.089188
min	635001.000000	0.000000	0.000000	0.000000
25%	635108.000000	1.000000	0.000000	0.000000

50%	635117.000000	2.000000	0.000000	0.000000
75%	635201.000000	3.000000	1.000000	0.000000
max	636902.000000	33.000000	13.000000	3.000000

=== Demographic Update Data Stats ===

	pincode	demo_age_5_17	demo_age_17_
count	3602.000000	3602.000000	3602.000000
mean	635230.080788	2.753470	17.361743
std	390.486927	12.279774	81.034158
min	635001.000000	0.000000	0.000000
25%	635108.000000	0.000000	3.000000
50%	635117.000000	1.000000	8.000000
75%	635203.000000	3.000000	15.000000
max	636902.000000	375.000000	2259.000000

=== Biometric Update Data Stats ===

	pincode	bio_age_5_17	bio_age_17_
count	8169.000000	8169.000000	8169.000000
mean	635245.165993	12.437508	9.129881
std	417.775448	37.283611	23.747110
min	635001.000000	0.000000	0.000000
25%	635108.000000	1.000000	2.000000
50%	635117.000000	5.000000	5.000000
75%	635203.000000	12.000000	10.000000
max	636906.000000	1270.000000	627.000000

```
[45]: # Rename columns for easier access
df_enrol.rename(columns={
    'age_0_5': 'Age_0_5',
    'age_5_17': 'Age_5_17',
    'age_18_greater': 'Age_18_plus'
}, inplace=True)

# Convert date column to datetime
df_enrol['date'] = pd.to_datetime(df_enrol['date'], dayfirst=True)

# Fill missing values with 0
df_enrol.fillna(0, inplace=True)

# Check cleaned data
print(df_enrol.head())
```

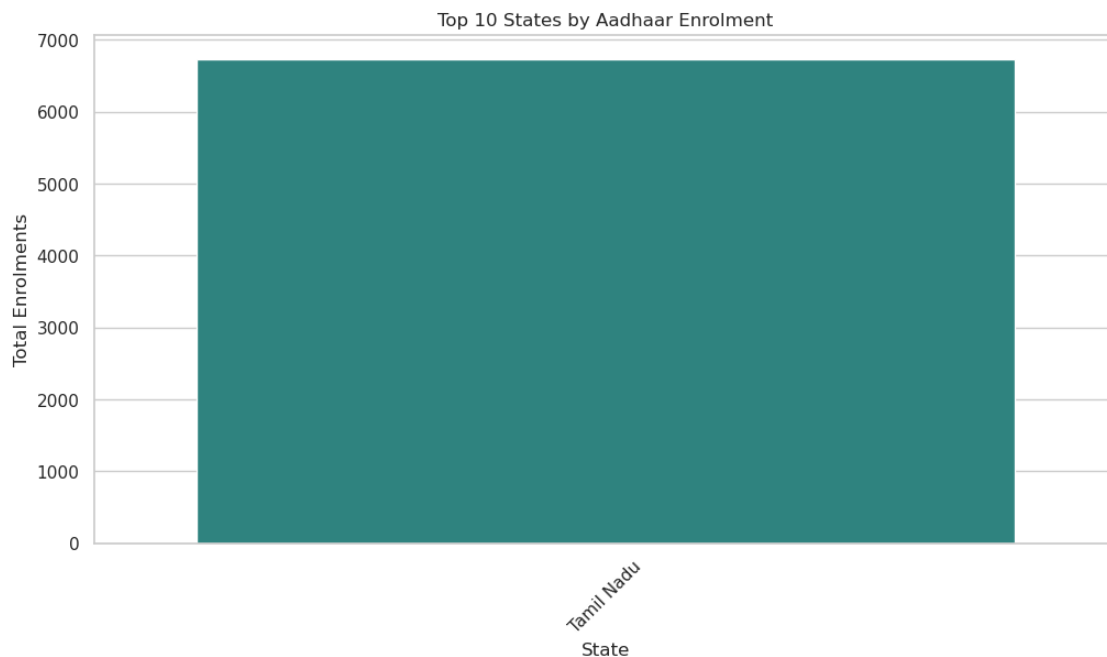
	date	state	district	pincode	Age_0_5	Age_5_17	Age_18_plus
0	2025-09-03	Tamil Nadu	Krishnagiri	635104	1	0	0
1	2025-09-03	Tamil Nadu	Krishnagiri	635107	3	0	0
2	2025-09-03	Tamil Nadu	Krishnagiri	635108	2	0	0
3	2025-09-03	Tamil Nadu	Krishnagiri	635114	0	1	0

```
[46]: import matplotlib.pyplot as plt
import seaborn as sns

# For nicer plots
sns.set(style="whitegrid")
%matplotlib inline

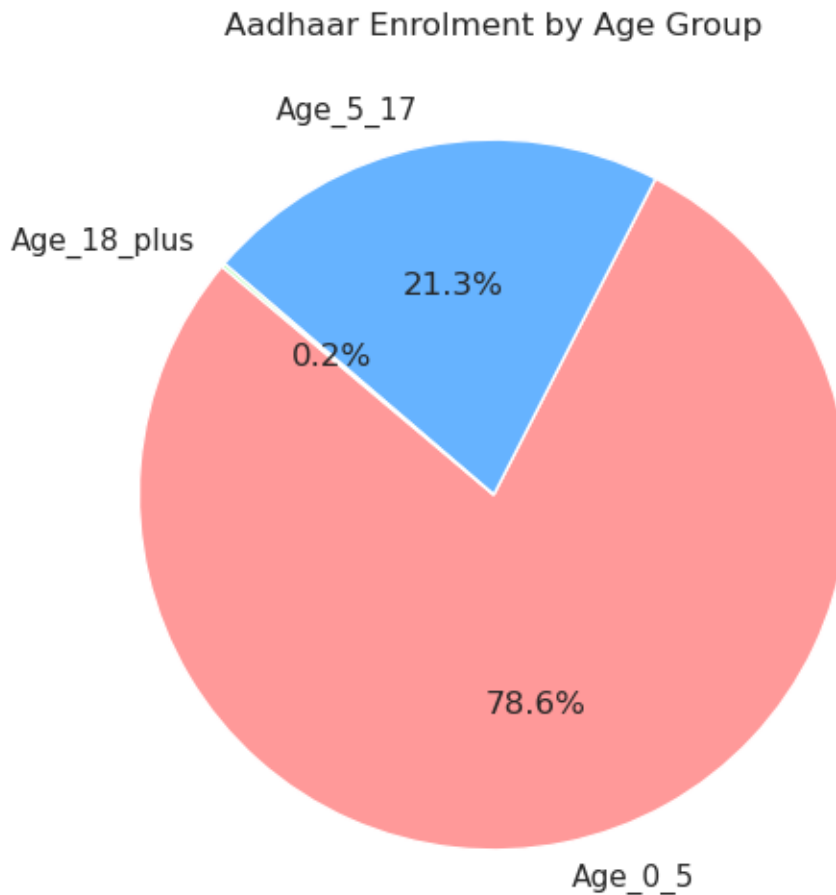
[47]: # Calculate total enrolments per state (sum of all age groups)
df_enrol['Total_Enrolment'] = df_enrol['Age_0_5'] + df_enrol['Age_5_17'] +
    ↪df_enrol['Age_18_plus']
state_enrol = df_enrol.groupby('state')['Total_Enrolment'].sum().
    ↪sort_values(ascending=False)

# Plot top 10 states
plt.figure(figsize=(12,6))
sns.barplot(x=state_enrol.head(10).index, y=state_enrol.head(10).values,
    ↪palette="viridis")
plt.title("Top 10 States by Aadhaar Enrolment")
plt.ylabel("Total Enrolments")
plt.xlabel("State")
plt.xticks(rotation=45)
plt.show()
```



```
[48]: # Sum of each age group
age_group_totals = df_enrol[['Age_0_5', 'Age_5_17', 'Age_18_plus']].sum()

# Pie chart
plt.figure(figsize=(6,6))
plt.pie(age_group_totals, labels=age_group_totals.index, autopct='%1.1f%%',
        ↪startangle=140, colors=['#ff9999', '#66b3ff', '#99ff99'])
plt.title("Aadhaar Enrolment by Age Group")
plt.show()
```

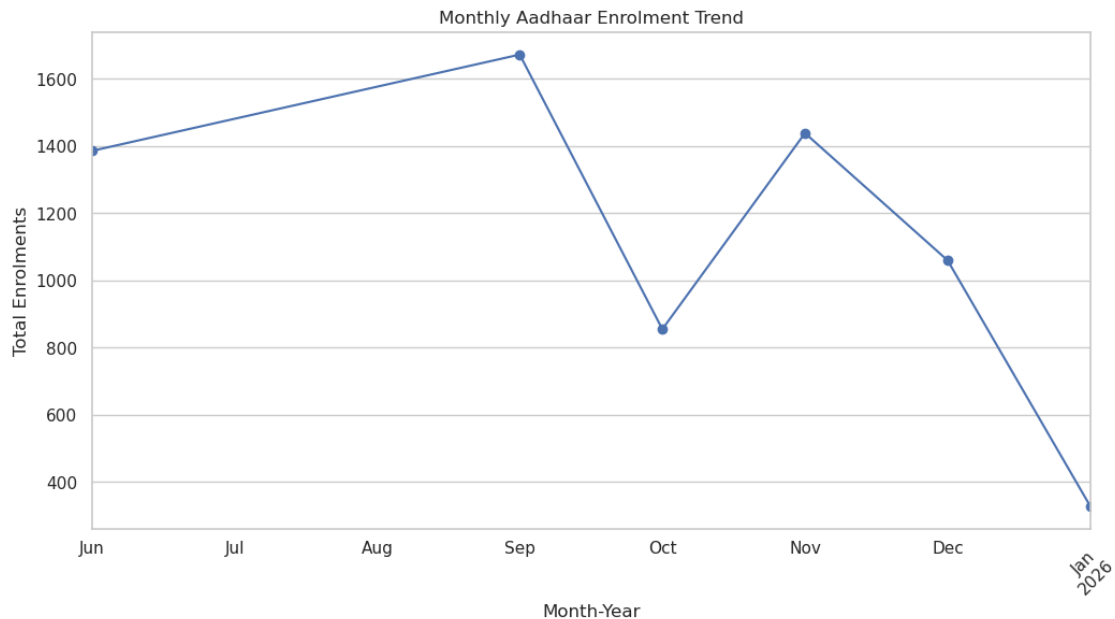


```
[49]: # Monthly enrolments
df_enrol['month_year'] = df_enrol['date'].dt.to_period('M')
monthly_enrol = df_enrol.groupby('month_year')['Total_Enrolment'].sum()

# Line plot
plt.figure(figsize=(12,6))
monthly_enrol.plot(kind='line', marker='o')
```



```
plt.title("Monthly Aadhaar Enrolment Trend")
plt.ylabel("Total Enrolments")
plt.xlabel("Month-Year")
plt.xticks(rotation=45)
plt.show()
```



```
[51]: # Check first few rows to see column types
df_demo.head()
df_demo.dtypes
```

```
[51]: date                object
state                object
district            object
pincode              int64
demo_age_5_17        int64
demo_age_17_         int64
dtype: object
```

```
[52]: # Convert all columns except 'date' and 'state' to numeric
cols_to_convert = df_demo.columns.difference(['date', 'state'])
df_demo[cols_to_convert] = df_demo[cols_to_convert].apply(pd.to_numeric,
    ↪errors='coerce')

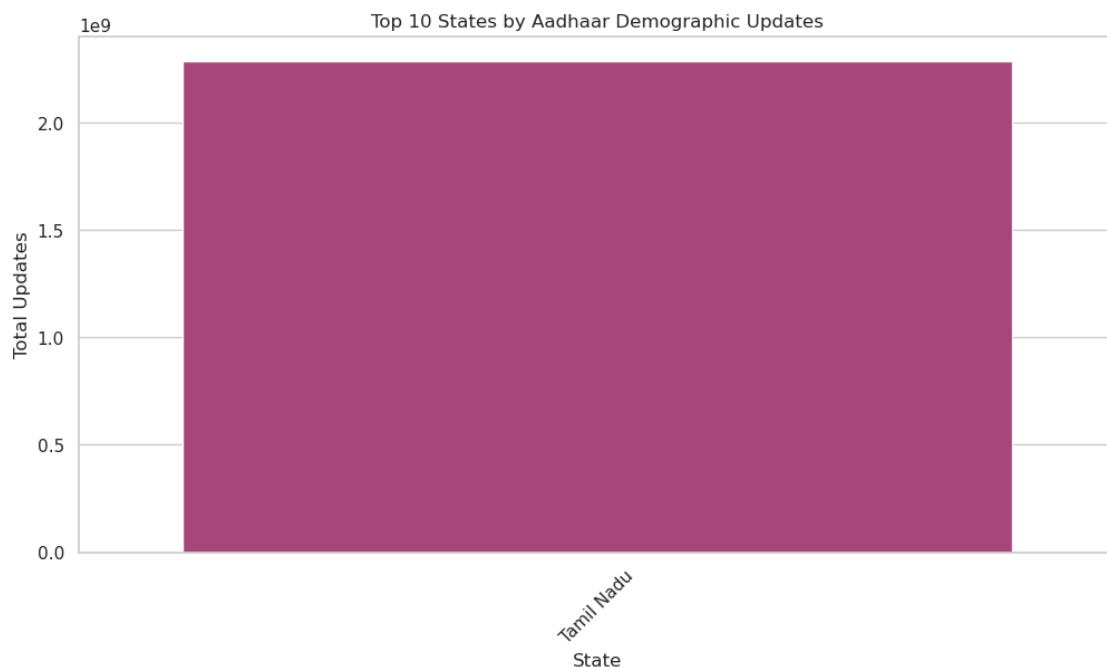
# Now sum updates per row
df_demo['Total_Updates'] = df_demo[cols_to_convert].sum(axis=1)
```

```

# Total updates per state
state_demo = df_demo.groupby('state')['Total_Updates'].sum().
    ↪sort_values(ascending=False)

# Plot top 10 states
plt.figure(figsize=(12,6))
sns.barplot(x=state_demo.head(10).index, y=state_demo.head(10).values,
    ↪palette="magma")
plt.title("Top 10 States by Aadhaar Demographic Updates")
plt.ylabel("Total Updates")
plt.xlabel("State")
plt.xticks(rotation=45)
plt.show()

```



```

[53]: # Load Biometric Update data
df_bio = pd.read_csv('Aadhar_Biometric.csv.csv') # make sure the filename
    ↪matches exactly

# Check first few rows and column types
df_bio.head()
df_bio.dtypes

```

```

[53]: date          object
      state         object
      district      object

```

```

pincode          int64
bio_age_5_17     int64
bio_age_17_      int64
dtype: object

```

```

[54]: # Assume columns like 'fingerprint', 'iris', 'face' exist (check df_bio.columns)
cols_to_convert_bio = df_bio.columns.difference(['date', 'state'])
df_bio[cols_to_convert_bio] = df_bio[cols_to_convert_bio].apply(pd.to_numeric,
    ↪errors='coerce')

```

```

# Sum all updates per row
df_bio['Total_Biometric_Updates'] = df_bio[cols_to_convert_bio].sum(axis=1)

```

```

[55]: state_bio = df_bio.groupby('state')['Total_Biometric_Updates'].sum().
    ↪sort_values(ascending=False)

```

```

[57]: # Aggregate numeric columns only per state per year
enrol_yearly = df_enrol.groupby(['state', 'year']).sum(numeric_only=True).
    ↪reset_index()
demo_yearly = df_demo.groupby(['state', 'year']).sum(numeric_only=True).
    ↪reset_index()
bio_yearly = df_bio.groupby(['state', 'year']).sum(numeric_only=True).
    ↪reset_index()

# Merge all three datasets
df_combined = enrol_yearly.merge(demo_yearly, on=['state', 'year'], how='outer',
    ↪suffixes=('_enrol', '_demo'))
df_combined = df_combined.merge(bio_yearly, on=['state', 'year'], how='outer',
    ↪suffixes=('', '_bio'))

# Fill missing values with 0
df_combined.fillna(0, inplace=True)

# Check the combined data
df_combined.head()

```

```

[57]:
   state  year  pincode_enrol  Age_0_5  Age_5_17  Age_18_plus  \
0  Tamil Nadu  2025      1308470224      5047      1350         11
1  Tamil Nadu  2026      45101784       245        82          0

   Total_Enrolment  district  pincode_demo  demo_age_5_17  demo_age_17_  \
0              6408         0.0  2223307917          9746        61743
1              327         0.0   64790834          172         794

   Total_Updates  district_bio  pincode  bio_age_5_17  bio_age_17_  \
0  2.223379e+09         0.0  5121351255          100288        73955

```

1	6.479180e+07	0.0	67966506	1314	627
---	--------------	-----	----------	------	-----

	Total_Biometric_Updates
0	5.121525e+09
1	6.796845e+07

```
[58]: # Check column names
print(df_combined.columns)

# Check data types
print(df_combined.dtypes)

# Check first few rows
df_combined.head()
```

```
Index(['state', 'year', 'pincode_enrol', 'Age_0_5', 'Age_5_17', 'Age_18_plus',
      'Total_Enrolment', 'district', 'pincode_demo', 'demo_age_5_17',
      'demo_age_17_', 'Total_Updates', 'district_bio', 'pincode',
      'bio_age_5_17', 'bio_age_17_', 'Total_Biometric_Updates'],
      dtype='object')
```

```
state          object
year           int32
pincode_enrol  int64
Age_0_5        int64
Age_5_17       int64
Age_18_plus    int64
Total_Enrolment int64
district       float64
pincode_demo   int64
demo_age_5_17  int64
demo_age_17_   int64
Total_Updates  float64
district_bio   float64
pincode        int64
bio_age_5_17   int64
bio_age_17_    int64
Total_Biometric_Updates float64
dtype: object
```

```
[58]:      state  year  pincode_enrol  Age_0_5  Age_5_17  Age_18_plus  \
0  Tamil Nadu  2025    1308470224    5047    1350         11
1  Tamil Nadu  2026     45101784     245     82          0

      Total_Enrolment  district  pincode_demo  demo_age_5_17  demo_age_17_  \
0                6408        0.0    2223307917        9746        61743
1                327        0.0     64790834         172         794
```

	Total_Updates	district_bio	pincode	bio_age_5_17	bio_age_17_	\
0	2.223379e+09	0.0	5121351255	100288	73955	
1	6.479180e+07	0.0	67966506	1314	627	

	Total_Biometric_Updates
0	5.121525e+09
1	6.796845e+07

```
[66]: # Get summary statistics
df_combined.describe()
```

```
[66]:
```

	year	pincode_enrol	Age_0_5	Age_5_17	Age_18_plus	\
count	2.000000	2.000000e+00	2.000000	2.000000	2.000000	
mean	2025.500000	6.767860e+08	2646.000000	716.000000	5.500000	
std	0.707107	8.933364e+08	3395.526763	896.611399	7.778175	
min	2025.000000	4.510178e+07	245.000000	82.000000	0.000000	
25%	2025.250000	3.609439e+08	1445.500000	399.000000	2.750000	
50%	2025.500000	6.767860e+08	2646.000000	716.000000	5.500000	
75%	2025.750000	9.926281e+08	3846.500000	1033.000000	8.250000	
max	2026.000000	1.308470e+09	5047.000000	1350.000000	11.000000	

	Total_Enrolment	district	pincode_demo	demo_age_5_17	demo_age_17_	\
count	2.000000	2.0	2.000000e+00	2.000000	2.000000	
mean	3367.500000	0.0	1.144049e+09	4959.000000	31268.500000	
std	4299.916336	0.0	1.526302e+09	6769.840323	43097.451207	
min	327.000000	0.0	6.479083e+07	172.000000	794.000000	
25%	1847.250000	0.0	6.044201e+08	2565.500000	16031.250000	
50%	3367.500000	0.0	1.144049e+09	4959.000000	31268.500000	
75%	4887.750000	0.0	1.683679e+09	7352.500000	46505.750000	
max	6408.000000	0.0	2.223308e+09	9746.000000	61743.000000	

	Total_Updates	district_bio	pincode	bio_age_5_17	bio_age_17_	\
count	2.000000e+00	2.0	2.000000e+00	2.000000	2.000000	
mean	1.144086e+09	0.0	2.594659e+09	50801.000000	37291.000000	
std	1.526352e+09	0.0	3.573283e+09	69985.186561	51850.726051	
min	6.479180e+07	0.0	6.796651e+07	1314.000000	627.000000	
25%	6.044387e+08	0.0	1.331313e+09	26057.500000	18959.000000	
50%	1.144086e+09	0.0	2.594659e+09	50801.000000	37291.000000	
75%	1.683733e+09	0.0	3.858005e+09	75544.500000	55623.000000	
max	2.223379e+09	0.0	5.121351e+09	100288.000000	73955.000000	

	Total_Biometric_Updates
count	2.000000e+00
mean	2.594747e+09
std	3.573404e+09
min	6.796845e+07
25%	1.331358e+09

```
50%                2.594747e+09
75%                3.858136e+09
max                 5.121525e+09
```

```
[62]: # Check column names in df_combined
print(df_combined.columns)
```

```
Index(['state', 'year', 'pincode_enrol', 'Age_0_5', 'Age_5_17', 'Age_18_plus',
      'Total_Enrolment', 'district', 'pincode_demo', 'demo_age_5_17',
      'demo_age_17_', 'Total_Updates', 'district_bio', 'pincode',
      'bio_age_5_17', 'bio_age_17_', 'Total_Biometric_Updates'],
      dtype='object')
```

```
[65]: ['date', 'state', 'district', 'pincode', 'age_0_5', 'age_5_17',
      ↪ 'age_18_greater']
```

```
[65]: ['date',
      'state',
      'district',
      'pincode',
      'age_0_5',
      'age_5_17',
      'age_18_greater']
```

```
[67]: # Check all columns in the combined dataframe
print(df_combined.columns)
```

```
Index(['state', 'year', 'pincode_enrol', 'Age_0_5', 'Age_5_17', 'Age_18_plus',
      'Total_Enrolment', 'district', 'pincode_demo', 'demo_age_5_17',
      'demo_age_17_', 'Total_Updates', 'district_bio', 'pincode',
      'bio_age_5_17', 'bio_age_17_', 'Total_Biometric_Updates'],
      dtype='object')
```

```
[ ]:
```