Exploratory Data Analysis of Aadhaar Enrolment Data

Problem Statement

Aadhaar enrolment data captures Enrolment patterns across different age groups over time. Understanding these patterns is crucial for identifying enrolment trends, enrolment participation, and potential accessibility gaps.

This study performs a structured Exploratory Data Analysis (EDA) on Aadhaar enrolment data to:

- Understand enrolment distribution across age groups
- Identify temporal trends
- Explore relationships between Enrolment groups
- Derive insights with potential administrative and social impact

Importing api_data_aadhar_enrolment_0_500000.csv

In [222…

```
#import warnings
#warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df1 = pd.read_csv('/Users/karansingh/Desktop/Hackathon/DAtaHackathon/api_dat
```

Dataset Overview

Understanding dataset structure, size, and column information.

In [223…

```
df1.head()
```

Out[223…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
0	02-03-2025	Meghalaya	East Khasi Hills	793121	11	61	37
1	09-03-2025	Karnataka	Bengaluru Urban	560043	14	33	39
2	09-03-2025	Uttar Pradesh	Kanpur Nagar	208001	29	82	12
3	09-03-2025	Uttar Pradesh	Aligarh	202133	62	29	15
4	09-03-2025	Karnataka	Bengaluru Urban	560016	14	16	21

In [224…
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500000 entries, 0 to 499999
Data columns (total 7 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   date            500000 non-null  object
 1   state           500000 non-null  object
 2   district        500000 non-null  object
 3   pincode         500000 non-null  int64 
 4   age_0_5         500000 non-null  int64 
 5   age_5_17        500000 non-null  int64 
 6   age_18_greater  500000 non-null  int64 
dtypes: int64(4), object(3)
memory usage: 26.7+ MB
```

In [225…
```
df1.shape
```

Out[225… `(500000, 7)`

In [226…
```
df1.columns
```

Out[226…
```
Index(['date', 'state', 'district', 'pincode', 'age_0_5', 'age_5_17',
       'age_18_greater'],
      dtype='object')
```

In [227…
```
df1.dtypes
```

Out[227…
```
date              object
state             object
district          object
pincode            int64
age_0_5            int64
age_5_17           int64
age_18_greater     int64
dtype: object
```

In [228…
```
df1.describe()
```

Out[228…

	pincode	age_0_5	age_5_17	age_18_greater
count	500000.000000	500000.000000	500000.000000	500000.000000
mean	519204.051054	4.040812	2.315682	0.245558
std	206793.322085	24.417921	20.191196	4.438557
min	100000.000000	0.000000	0.000000	0.000000
25%	362229.000000	1.000000	0.000000	0.000000
50%	517131.000000	2.000000	0.000000	0.000000
75%	712139.000000	3.000000	1.000000	0.000000
max	855456.000000	2688.000000	1812.000000	855.000000

In [232…
```
## check null value
df1['state'].isnull().sum()
```

Out[232…
```
np.int64(0)
```

In [233…
```
## check state column
df1['state'].nunique()
```

Out[233…
```
54
```

Importing api_data_aadhar_enrolment_500000_10000

In [235…
```
df2 = pd.read_csv('/Users/karansingh/Desktop/Hackathon/DAtaHackathon/api_dat
```

Dataset Overview

Understanding dataset structure, size, and column information.

In [236…
```
df2.head()
```

Out[236…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
0	26-10-2025	Andhra Pradesh	Nalgonda	508004	0	1	0
1	26-10-2025	Andhra Pradesh	Nalgonda	508238	1	0	0
2	26-10-2025	Andhra Pradesh	Nalgonda	508278	1	0	0
3	26-10-2025	Andhra Pradesh	Nandyal	518432	0	1	0
4	26-10-2025	Andhra Pradesh	Nandyal	518543	1	0	0

In [237…
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500000 entries, 0 to 499999
Data columns (total 7 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   date            500000 non-null  object
 1   state           500000 non-null  object
 2   district        500000 non-null  object
 3   pincode         500000 non-null  int64 
 4   age_0_5         500000 non-null  int64 
 5   age_5_17        500000 non-null  int64 
 6   age_18_greater  500000 non-null  int64 
dtypes: int64(4), object(3)
memory usage: 26.7+ MB
```

In [238…
```
# check shape
df2.shape
```

Out[238… `(500000, 7)`

In [239…
```
# check columns
df2.columns
```

Out[239…
```
Index(['date', 'state', 'district', 'pincode', 'age_0_5', 'age_5_17',
       'age_18_greater'],
      dtype='object')
```

In [241…
```
df2.describe()
```

Out[241…

	pincode	age_0_5	age_5_17	age_18_greater
count	500000.000000	500000.000000	500000.000000	500000.000000
mean	518077.362504	3.009628	1.082962	0.089984
std	204615.861812	4.658289	2.537596	1.073745
min	100000.000000	0.000000	0.000000	0.000000
25%	365541.000000	1.000000	0.000000	0.000000
50%	517583.000000	2.000000	0.000000	0.000000
75%	695023.000000	3.000000	1.000000	0.000000
max	855456.000000	210.000000	97.000000	318.000000

Importing api_data_aadhar_enrolment_1000000_1006(

In [248…

```
df3 = pd.read_csv('/Users/karansingh/Desktop/Hackathon/DAtaHackathon/api_dat
```

Dataset Overview

Understanding dataset structure, size, and column information.

In [249…

```
df3.head()
```

Out[249…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
0	31-12-2025	Karnataka	Bidar	585330	2	3	0
1	31-12-2025	Karnataka	Bidar	585402	6	0	0
2	31-12-2025	Karnataka	Bidar	585413	1	0	0
3	31-12-2025	Karnataka	Bidar	585418	1	2	0
4	31-12-2025	Karnataka	Bidar	585421	4	3	0

In [250…

```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6029 entries, 0 to 6028
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   date            6029 non-null   object
 1   state           6029 non-null   object
 2   district        6029 non-null   object
 3   pincode         6029 non-null   int64 
 4   age_0_5         6029 non-null   int64 
 5   age_5_17        6029 non-null   int64 
 6   age_18_greater  6029 non-null   int64 
dtypes: int64(4), object(3)
memory usage: 329.8+ KB
```

In [251…
```
df3.shape
```

Out[251… (6029, 7)

In [252…
```
df3.describe()
```

Out[252…

	pincode	age_0_5	age_5_17	age_18_greater
count	6029.000000	6029.000000	6029.000000	6029.000000
mean	518765.547023	3.606734	3.493448	0.096533
std	193308.752319	6.055847	6.694502	0.479475
min	110003.000000	0.000000	0.000000	0.000000
25%	380022.000000	1.000000	0.000000	0.000000
50%	518005.000000	2.000000	1.000000	0.000000
75%	685595.000000	4.000000	3.000000	0.000000
max	855116.000000	102.000000	89.000000	9.000000

Merging all three Enrolment datasets

In [260…
```
df = pd.concat([df1,df2,df3],ignore_index=True)
```

In [261…
```
#Datatype of column and number of values in each column are
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1006029 entries, 0 to 1006028
Data columns (total 7 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   date            1006029 non-null  object
 1   state           1006029 non-null  object
 2   district        1006029 non-null  object
 3   pincode         1006029 non-null  int64
 4   age_0_5         1006029 non-null  int64
 5   age_5_17        1006029 non-null  int64
 6   age_18_greater  1006029 non-null  int64
dtypes: int64(4), object(3)
memory usage: 53.7+ MB
```

In [262…

```
# Top rows of dataset are :
df.head()
```

Out[262…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
0	02-03-2025	Meghalaya	East Khasi Hills	793121	11	61	37
1	09-03-2025	Karnataka	Bengaluru Urban	560043	14	33	39
2	09-03-2025	Uttar Pradesh	Kanpur Nagar	208001	29	82	12
3	09-03-2025	Uttar Pradesh	Aligarh	202133	62	29	15
4	09-03-2025	Karnataka	Bengaluru Urban	560016	14	16	21

In [263…

```
# Bottom rows of dataset are :
df.tail()
```

Out[263…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
1006024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0
1006025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0
1006026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0
1006027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0
1006028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0

In [264…
```
#Total Columns Present in Dataset
df.columns
```

Out[264…
```
Index(['date', 'state', 'district', 'pincode', 'age_0_5', 'age_5_17',
       'age_18_greater'],
      dtype='object')
```

In [265…
```
# check shape
print("Shape of df is :",df.shape)
```

```
Shape of df is : (1006029, 7)
```

In [266…
```
#Describing total count , mean, std deviation, min value , max value,25%ile,
df.describe()
```

Out[266…

	pincode	age_0_5	age_5_17	age_18_greater
count	1.006029e+06	1.006029e+06	1.006029e+06	1.006029e+06
mean	5.186415e+05	3.525709e+00	1.710074e+00	1.673441e-01
std	2.056360e+05	1.753851e+01	1.436963e+01	3.220525e+00
min	1.000000e+05	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.636410e+05	1.000000e+00	0.000000e+00	0.000000e+00
50%	5.174170e+05	2.000000e+00	0.000000e+00	0.000000e+00
75%	7.001040e+05	3.000000e+00	1.000000e+00	0.000000e+00
max	8.554560e+05	2.688000e+03	1.812000e+03	8.550000e+02

Data Cleaning & Preprocessing

This step ensures consistency, correctness, and readiness for analysis.

In [270…
```
# Convert date column
df['date'] = pd.to_datetime(df['date'], dayfirst=True)

# Extract temporal features
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
df['month_name'] = df['date'].dt.strftime('%b')
```

In [271…
```
#Checking Count of Unique value present in sate
print("There are ",df['state'].nunique() ,"Values present in state column.")
```

```
There are  55 Values present in state column.
```

In [272…
```
#Checking for unique value present in state
df['state'].unique()
```

Out[272…
```
array(['Meghalaya', 'Karnataka', 'Uttar Pradesh', 'Bihar', 'Maharashtra',
       'Haryana', 'Rajasthan', 'Punjab', 'Delhi', 'Madhya Pradesh',
       'West Bengal', 'Assam', 'Uttarakhand', 'Gujarat', 'Andhra Pradesh',
       'Tamil Nadu', 'Chhattisgarh', 'Jharkhand', 'Nagaland', 'Manipur',
       'Telangana', 'Tripura', 'Mizoram', 'Jammu and Kashmir',
       'Chandigarh', 'Sikkim', 'Odisha', 'Kerala',
       'The Dadra And Nagar Haveli And Daman And Diu',
       'Arunachal Pradesh', 'Himachal Pradesh', 'Goa',
       'Jammu And Kashmir', 'Dadra and Nagar Haveli and Daman and Diu',
       'Ladakh', 'Andaman and Nicobar Islands', 'Orissa', 'Pondicherry',
       'Puducherry', 'Lakshadweep', 'Andaman & Nicobar Islands',
       'Dadra & Nagar Haveli', 'Dadra and Nagar Haveli', 'Daman and Diu',
       'WEST BENGAL', 'Jammu & Kashmir', 'West  Bengal', '100000',
       'Daman & Diu', 'West Bangal', 'Westbengal', 'West bengal',
       'andhra pradesh', 'ODISHA', 'WESTBENGAL'], dtype=object)
```

There are some state with different spelling We are going to map them to a single state name .

In [275…
```
import pandas as pd
import re

def clean_state_name(x):
    if pd.isna(x):
        return x
    x = str(x).lower()
    x = re.sub(r'[^a-z]', '', x) # remove symbols like &,
    x = re.sub(r'\s+',' ', x).strip() # remove extra spaces
    return x
```

In [276…
```
state_mapping = {
    # Andhra Pradesh
    "andhrapradesh": "Andhra Pradesh",

    # Arunachal Pradesh
    "arunachalpradesh": "Arunachal Pradesh",
```

```
    # Assam
    "assam": "Assam",

    # Bihar
    "bihar": "Bihar",

    # Chhattisgarh
    "chhattisgarh": "Chhattisgarh",

    # Delhi
    "delhi": "Delhi",

    # Goa
    "goa": "Goa",

    # Gujarat
    "gujarat": "Gujarat",

    # Haryana
    "haryana": "Haryana",

    # Himachal Pradesh
    "himachalpradesh": "Himachal Pradesh",

    # Jammu & Kashmir / Ladakh
    "jammuandkashmir": "Jammu and Kashmir",
    "jammukashmir": "Jammu and Kashmir",

    "ladakh": "Ladakh",

    # Jharkhand
    "jharkhand": "Jharkhand",

    # Karnataka
    "karnataka": "Karnataka",

    # Kerala
    "kerala": "Kerala",

    # Madhya Pradesh
    "madhyapradesh": "Madhya Pradesh",

    # Maharashtra
    "maharashtra": "Maharashtra",

    # Manipur
    "manipur": "Manipur",

    # Meghalaya
    "meghalaya": "Meghalaya",

    # Mizoram
    "mizoram": "Mizoram",

    # Nagaland
    "nagaland": "Nagaland",
```

```
    # Odisha (Orissa old name)
    "odisha": "Odisha",
    "orissa": "Odisha",

    # Punjab
    "punjab": "Punjab",

    # Rajasthan
    "rajasthan": "Rajasthan",

    # Sikkim
    "sikkim": "Sikkim",

    # Tamil Nadu
    "tamilnadu": "Tamil Nadu",

    # Telangana
    "telangana": "Telangana",

    # Tripura
    "tripura": "Tripura",

    # Uttar Pradesh
    "uttarpradesh": "Uttar Pradesh",

    # Uttarakhand
    "uttarakhand": "Uttarakhand",

    # West Bengal (ALL variations including typo "Bangal")
    "westbengal": "West Bengal",
    "westbangal": "West Bengal",

    # Andaman & Nicobar Islands
    "andamannicobarislands": "Andaman and Nicobar Islands",
    "andamanandnicobarislands": "Andaman and Nicobar Islands",

    # Chandigarh
    "chandigarh": "Chandigarh",

    # Dadra & Nagar Haveli / Daman & Diu (merged UT)
    "dadraandnagarhaveli": "Dadra and Nagar Haveli and Daman and Diu",
    "damananddiu": "Dadra and Nagar Haveli and Daman and Diu",
    "dadranagarhaveli": "Dadra and Nagar Haveli and Daman and Diu",
    "damandiu": "Dadra and Nagar Haveli and Daman and Diu",
    "dadraandnagarhavelianddamananddiu": "Dadra and Nagar Haveli and Daman a
    "thedadraandnagarhavelianddamananddiu": "Dadra and Nagar Haveli and Dama

    # Lakshadweep
    "lakshadweep": "Lakshadweep",

    # Puducherry
    "pondicherry": "Puducherry",
    "puducherry": "Puducherry",
}
```

In [277…
```
df['state_clean'] = (
    df['state']
    .apply(clean_state_name)
    .map(state_mapping)
)
```

In [278…
```
# Drop invalid entries
df = df[~df['state'].astype(str).str.isnumeric()]

# check unmapped states
unmapped_states = df[df['state_clean'].isnull()]['state'].unique()
print("Unmapped States:", unmapped_states)
```

```
Unmapped States: []
```

In [279…
```
# Counting state unique values after mapping
print("There are ",df['state_clean'].nunique(),"Unique value present in stat
```

```
There are  36 Unique value present in state.
```

In [280…
```
df['state_clean'].value_counts()
```

Out[280…

```
state_clean
Uttar Pradesh                                    110369
Tamil Nadu                                        92552
Maharashtra                                       77191
West Bengal                                       76561
Karnataka                                         70198
Andhra Pradesh                                    65663
Bihar                                             60567
Rajasthan                                         56159
Madhya Pradesh                                    50225
Odisha                                            47011
Gujarat                                           46624
Telangana                                         42774
Kerala                                            39145
Assam                                             31827
Jharkhand                                         23218
Punjab                                            20439
Chhattisgarh                                      18550
Haryana                                           15997
Jammu and Kashmir                                 11455
Himachal Pradesh                                  10346
Uttarakhand                                       10007
Delhi                                              6804
Meghalaya                                          3771
Tripura                                            3729
Manipur                                            3218
Nagaland                                           1999
Puducherry                                         1859
Arunachal Pradesh                                  1601
Goa                                                1527
Mizoram                                            1481
Sikkim                                             1010
Chandigarh                                          859
Dadra and Nagar Haveli and Daman and Diu            416
Andaman and Nicobar Islands                         392
Ladakh                                              304
Lakshadweep                                         159
Name: count, dtype: int64
```

In [285…

```
# checking null values in each column
df.isnull().sum()
```

Out[285…

```
date              0
state             0
district          0
pincode           0
age_0_5           0
age_5_17          0
age_18_greater    0
year              0
month             0
month_name        0
state_clean       0
dtype: int64
```

In [286…
```
# Checking for duplicate rows
df.duplicated().sum()
```

Out[286… np.int64(22956)

In [287…
```
# dropping duplicates rows
df = df.drop_duplicates()
```

In [288…
```
## total enrolment by state
state_summary = df.groupby('state_clean')[[
    'age_0_5', 'age_5_17', 'age_18_greater'
]].sum()

state_summary['total_enrollment'] = (
    state_summary['age_0_5'] +
    state_summary['age_5_17'] +
    state_summary['age_18_greater']
)

top10_states = state_summary.sort_values(
    by='total_enrollment', ascending=False
).head(10)
```

In [289…
```
state_summary
```

Out [289...

state_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Andaman and Nicobar Islands	469	32	0	501
Andhra Pradesh	109394	13414	1465	124273
Arunachal Pradesh	1914	2176	150	4240
Assam	137970	64834	22555	225359
Bihar	254911	327043	11799	593753
Chandigarh	2377	210	33	2620
Chhattisgarh	79653	18158	1962	99773
Dadra and Nagar Haveli and Daman and Diu	1484	248	50	1782
Delhi	67844	21971	3023	92838
Goa	1871	253	156	2280
Gujarat	188709	70270	16063	275042
Haryana	85112	8897	1076	95085
Himachal Pradesh	16081	650	178	16909
Jammu and Kashmir	39314	7802	522	47638
Jharkhand	96048	56152	1412	153612
Karnataka	176178	33402	10038	219618
Kerala	52950	18360	2640	73950
Ladakh	466	133	18	617
Lakshadweep	188	10	1	199
Madhya Pradesh	363244	115172	9476	487892
Maharashtra	274274	81069	8103	363446
Manipur	5044	7895	260	13199
Meghalaya	21072	53089	35078	109239
Mizoram	4044	1259	471	5774
Nagaland	4453	9856	1120	15429
Odisha	97500	22228	726	120454
Puducherry	2746	193	44	2983
Punjab	60481	12175	3117	75773
Rajasthan	224977	110131	5483	340591

state_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Sikkim	1040	1030	105	2175
Tamil Nadu	178294	36214	1202	215710
Telangana	103768	24035	1145	128948
Tripura	7165	3597	246	11008
Uttar Pradesh	511727	473205	17699	1002631
Uttarakhand	31208	5410	338	36956
West Bengal	270419	90335	8495	369249

Data Quality & Consistency Checks

Logical consistency

Output Interpretation

True ✅ → No negative values in that column

False ❌ → At least one negative value exists (data error)

In [290…
```
(df['age_0_5'] >= 0).all()
(df['age_5_17'] >= 0).all()
(df['age_18_greater'] >= 0).all()
```

Out[290… `np.True_`

In [291…
```
(df['age_0_5'] + df['age_5_17'] + df['age_18_greater'] > 0).all()
```

Out[291… `np.True_`

In [292…
```
df['total_enrollment'] = df['age_0_5'] + df['age_5_17'] + df["age_18_greater
```

Distribution Shape Analysis

Skewness measures the asymmetry of a data distribution, while kurtosis measures the tailedness (presence of outliers) in the distribution.

- Skewness = 0 → Distribution is perfectly symmetric
- Skewness > 0 → Distribution is right-skewed (long right tail)
- Skewness < 0 → Distribution is left-skewed (long left tail)
- Kurtosis = 0 → Distribution is normal (mesokurtic)

- Kurtosis > 0 → Distribution has heavy tails (more extreme outliers)
- Kurtosis < 0 → Distribution has light tails (fewer outliers)

In [293…
```
from scipy.stats import skew, kurtosis

skew(df['total_enrollment']), kurtosis(df['total_enrollment'])
```

Out[293…
```
(np.float64(38.72179961381051), np.float64(2273.9610853382146))
```

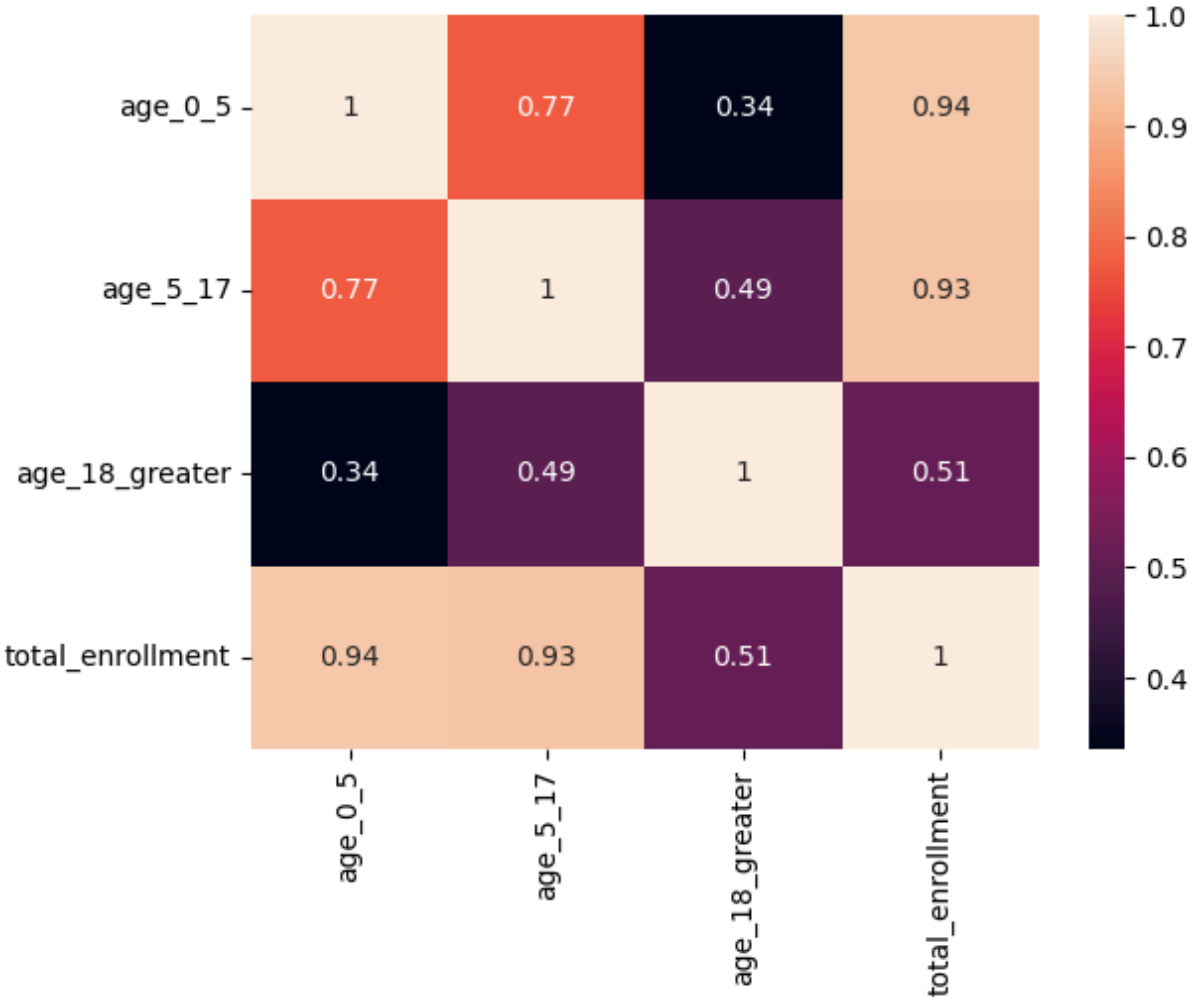
Our data is highly non-normal, with most values clustered at the lower end and a few extreme high values dominating the distribution.

Correlation Analysis

Total enrollment shows a strong positive correlation with all age-group variables, indicating that higher population in any age group contributes directly to higher total enrollment.

In [294…
```
import seaborn as sns
corr = df[['age_0_5','age_5_17','age_18_greater','total_enrollment']].corr()

sns.heatmap(corr, annot=True)
```

Out[294…
```
<Axes: >
```



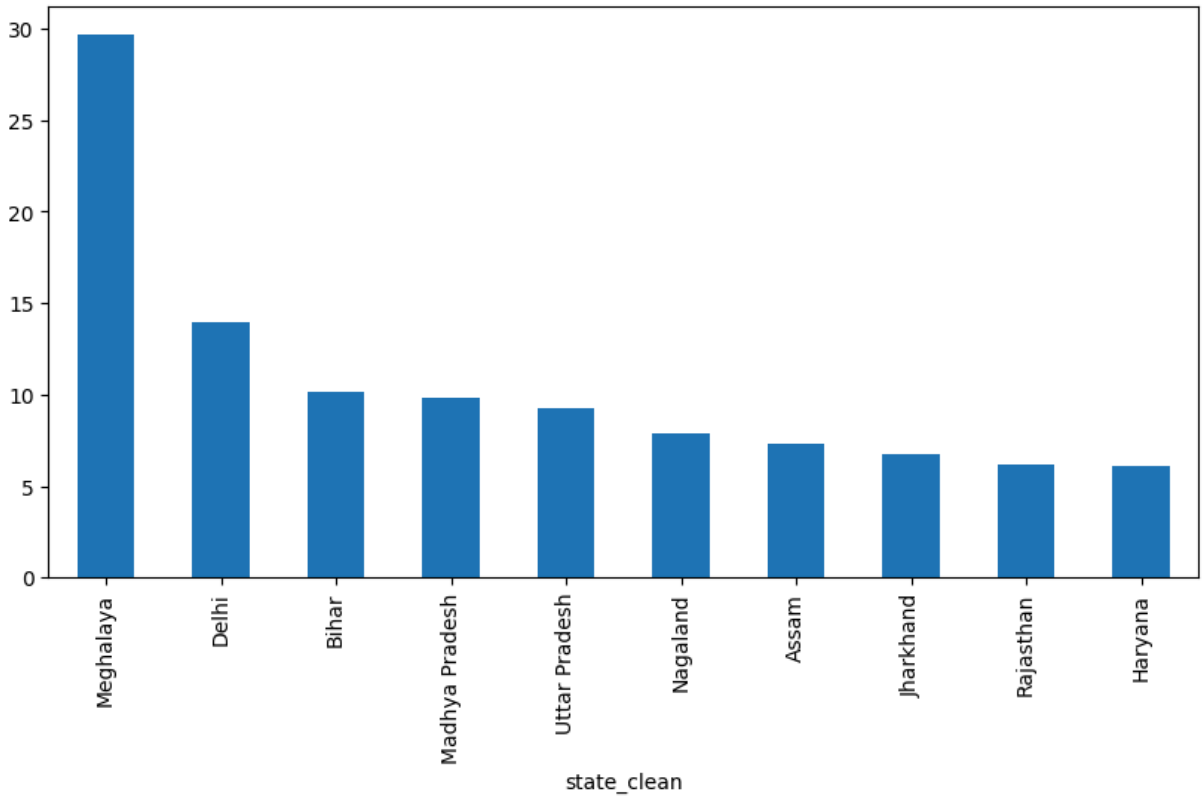
Normalized Comparisons

The top-ranked states show higher average enrollment per institution, suggesting the presence of larger schools or denser student populations.

In [295…

```
# Per-capita style normalization (proxy)
state_avg = df.groupby('state_clean')['total_enrollment'].mean()

state_avg.sort_values(ascending=False).head(10).plot(
    kind='bar', figsize=(10,5)
)
```

Out[295… <Axes: xlabel='state_clean'>



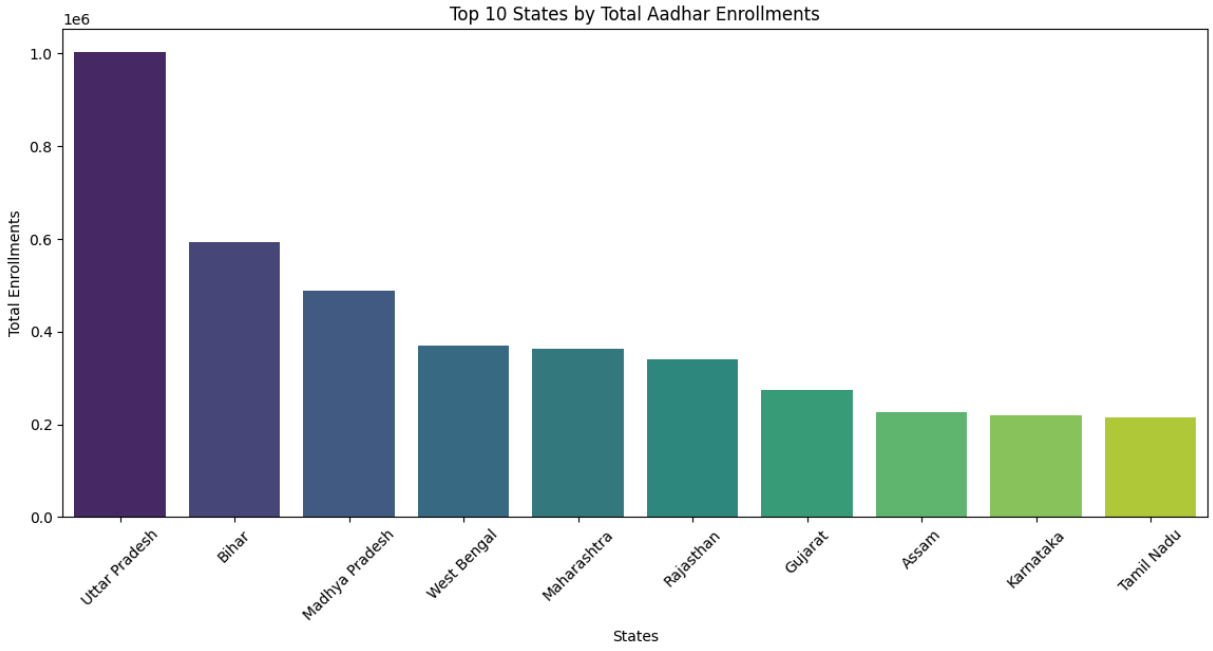
Bar chart → Top 10 states by total enrollment

Top 10 states account for a significant share of total enrollments, indicating population concentration and higher enrollment activity

In [296…

```
## bar plot for top 10 states
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.barplot(
    x=top10_states.index,
    y=top10_states['total_enrollment'],
    palette='viridis'
)
plt.title('Top 10 States by Total Aadhar Enrollments')
plt.xlabel('States')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.show()
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/3755885565.p
y:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(
```



Rank Analysis

In [297…
```
state_rank = state_summary['total_enrollment'].rank(ascending=False)

state_rank.sort_values().head(10)
```

Out[297…
```
state_clean
Uttar Pradesh      1.0
Bihar              2.0
Madhya Pradesh     3.0
West Bengal        4.0
Maharashtra        5.0
Rajasthan          6.0
Gujarat            7.0
Assam              8.0
Karnataka          9.0
Tamil Nadu        10.0
Name: total_enrollment, dtype: float64
```

Variability Analysis

🧠 Meaning:

- High std → inconsistent enrollment
- Low std → stable system

In [298…
```
state_std = df.groupby('state_clean')['total_enrollment'].std()

state_std.sort_values(ascending=False).head(10)
```

Out[298…
```
state_clean
Meghalaya         135.104214
Delhi              77.593888
Uttar Pradesh      56.025312
Bihar              45.115614
Madhya Pradesh     42.377277
Nagaland           41.984775
Haryana            36.049266
Gujarat            34.775802
Assam              33.639722
Manipur            33.200808
Name: total_enrollment, dtype: float64
```

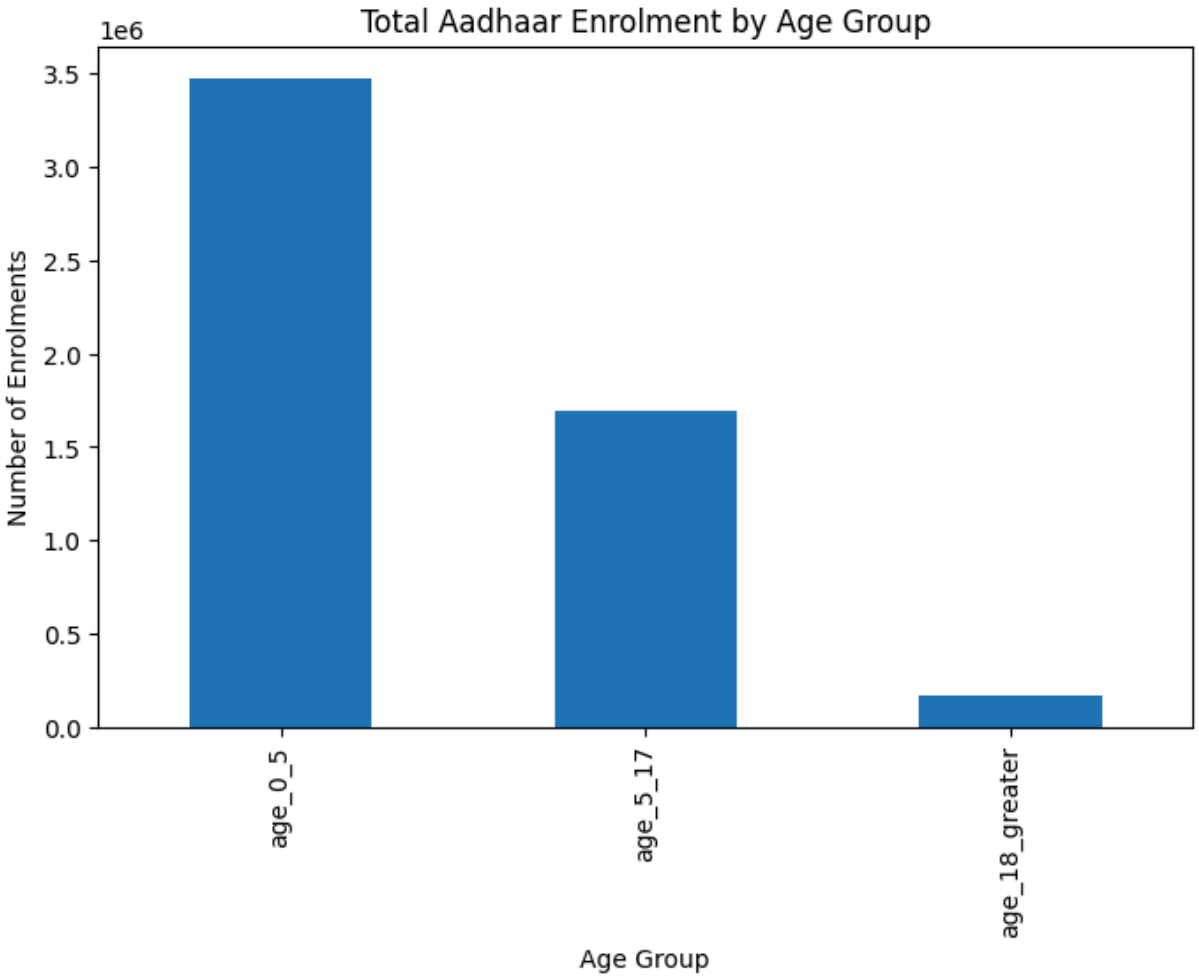
Age Group Distribution

This analysis examines enrolment contribution of individual age groups.

In [299…
```
# Identify age group columns dynamically
age_cols = [c for c in df.columns if c.startswith('age_')]
age_cols
```

Out[299…
```
['age_0_5', 'age_5_17', 'age_18_greater']
```

In [300…
```
age_totals = df[age_cols].sum().sort_values(ascending=False)

plt.figure(figsize=(8,5))
age_totals.plot(kind='bar')
plt.title("Total Aadhaar Enrolment by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Number of Enrolments")
plt.show()
```



Children vs Adult Enrollment (Top 10 States)

In [301…

```
top10_states['children_enrollment'] = (
    top10_states['age_0_5'] +
    top10_states['age_5_17']
)
```

📊 Age Group Enrollment Comparison (Top 10 States)

The bar chart compares enrollment counts between **children** and **adults (18+ age group)** across the top 10 states.

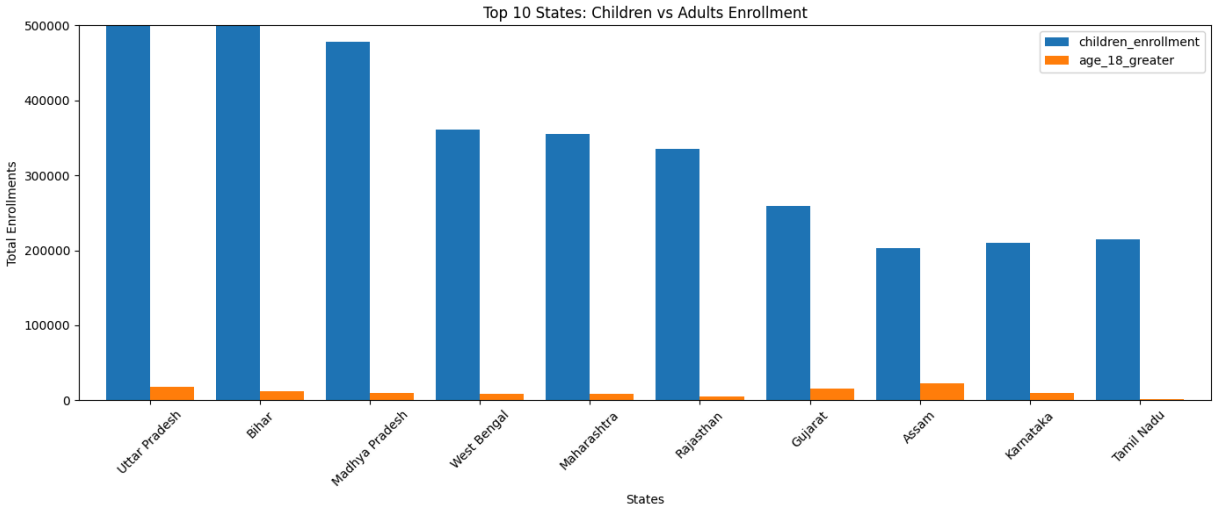
Key Observations:

- Adult (18+) enrollment is consistently higher than children enrollment in most states.
- The gap between adult and children enrollment varies significantly by state.
- States with higher total enrollments show a stronger dominance of the adult age group.

In [303…

```
import matplotlib.pyplot as plt
```

```
ax = top10_states[['children_enrollment', 'age_18_greater']].plot(
    kind='bar',
    figsize=(14,6),
    width=0.8
)

ax.set_title('Top 10 States: Children vs Adults Enrollment')
ax.set_xlabel('States')
ax.set_ylabel('Total Enrollments')

ax.set_ylim(0, 500000)   # 👈 adjust this value as needed

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Age Composition Percentage

States with higher child enrollment percentages may indicate stronger newborn and school-level enrollment outreach.

In [304…

```
top10_states['children_percent'] = (
    top10_states['children_enrollment'] /
    top10_states['total_enrollment']
) * 100

top10_states['adult_percent'] = (
    top10_states['age_18_greater'] /
    top10_states['total_enrollment']
) * 100
```

In [305…

```
top10_states[['children_percent', 'adult_percent']]
```

Out [305…

state_clean	children_percent	adult_percent
Uttar Pradesh	98.234744	1.765256
Bihar	98.012810	1.987190
Madhya Pradesh	98.057767	1.942233
West Bengal	97.699384	2.300616
Maharashtra	97.770508	2.229492
Rajasthan	98.390151	1.609849
Gujarat	94.159801	5.840199
Assam	89.991525	10.008475
Karnataka	95.429336	4.570664
Tamil Nadu	99.442770	0.557230

Plotting Monthly Enrollment trend by age group

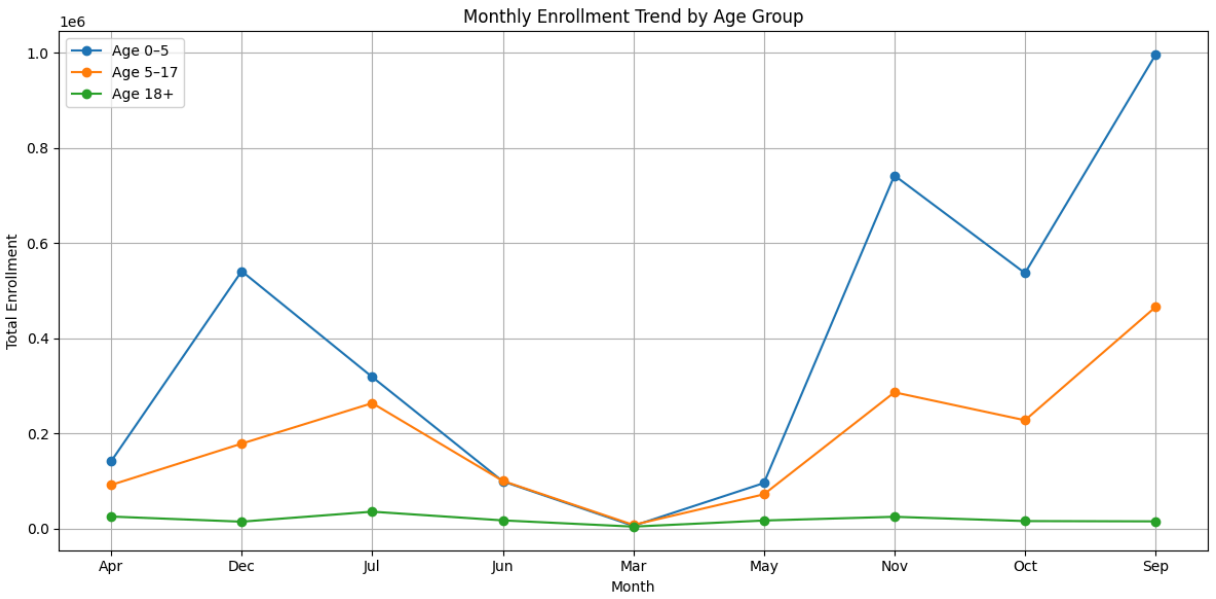
In [306…

```
# # Monthly enrollment trend
monthly_enrollment = df.groupby('month_name')[[
    'age_0_5', 'age_5_17', 'age_18_greater'
]].sum()
```

In [307…

```
plt.figure(figsize=(12,6))

plt.plot(monthly_enrollment.index,
         monthly_enrollment['age_0_5'],
         marker='o',
         label='Age 0–5')

plt.plot(monthly_enrollment.index,
         monthly_enrollment['age_5_17'],
         marker='o',
         label='Age 5–17')

plt.plot(monthly_enrollment.index,
         monthly_enrollment['age_18_greater'],
         marker='o',
         label='Age 18+')

plt.title('Monthly Enrollment Trend by Age Group')
plt.xlabel('Month')
plt.ylabel('Total Enrollment')
plt.legend()
plt.grid(True)
```

```
plt.tight_layout()
plt.show()
```



In [308…
```
df.head()
```

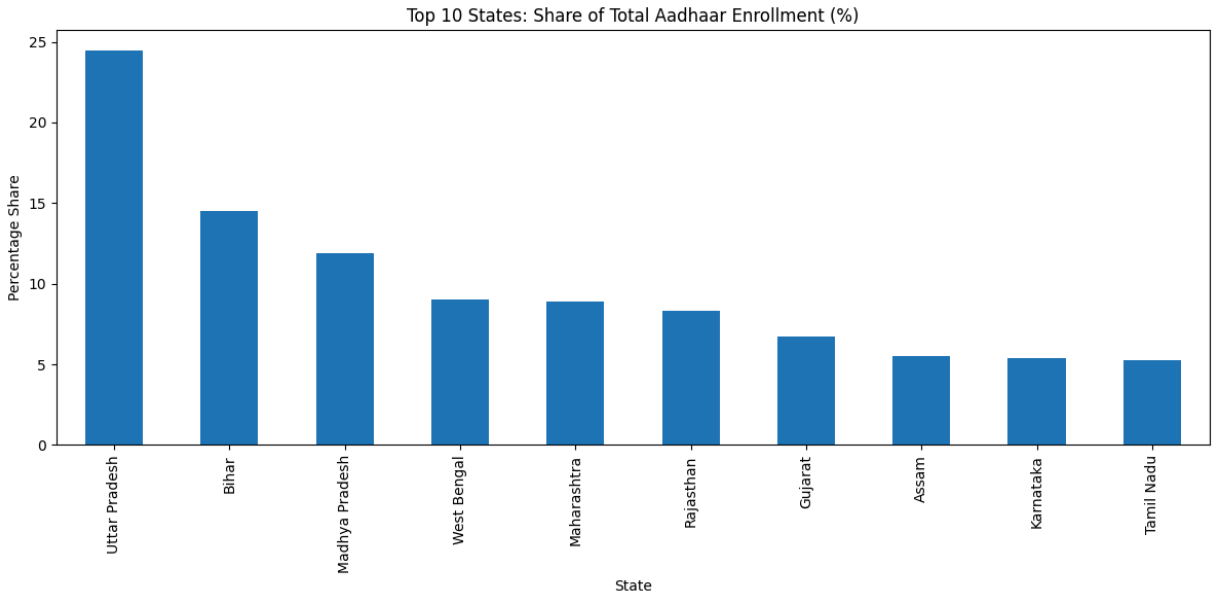
Out[308…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	year
0	2025-03-02	Meghalaya	East Khasi Hills	793121	11	61	37	2025
1	2025-03-09	Karnataka	Bengaluru Urban	560043	14	33	39	2025
2	2025-03-09	Uttar Pradesh	Kanpur Nagar	208001	29	82	12	2025
3	2025-03-09	Uttar Pradesh	Aligarh	202133	62	29	15	2025
4	2025-03-09	Karnataka	Bengaluru Urban	560016	14	16	21	2025

State-wise Enrollment Share (%)

In [309…
```
state_total = df.groupby('state_clean')[['age_0_5','age_5_17','age_18_greate
state_total['total'] = state_total.sum(axis=1)

top10 = state_total.sort_values('total', ascending=False).head(10)

(top10['total'] / top10['total'].sum() * 100).plot(
    kind='bar',
```

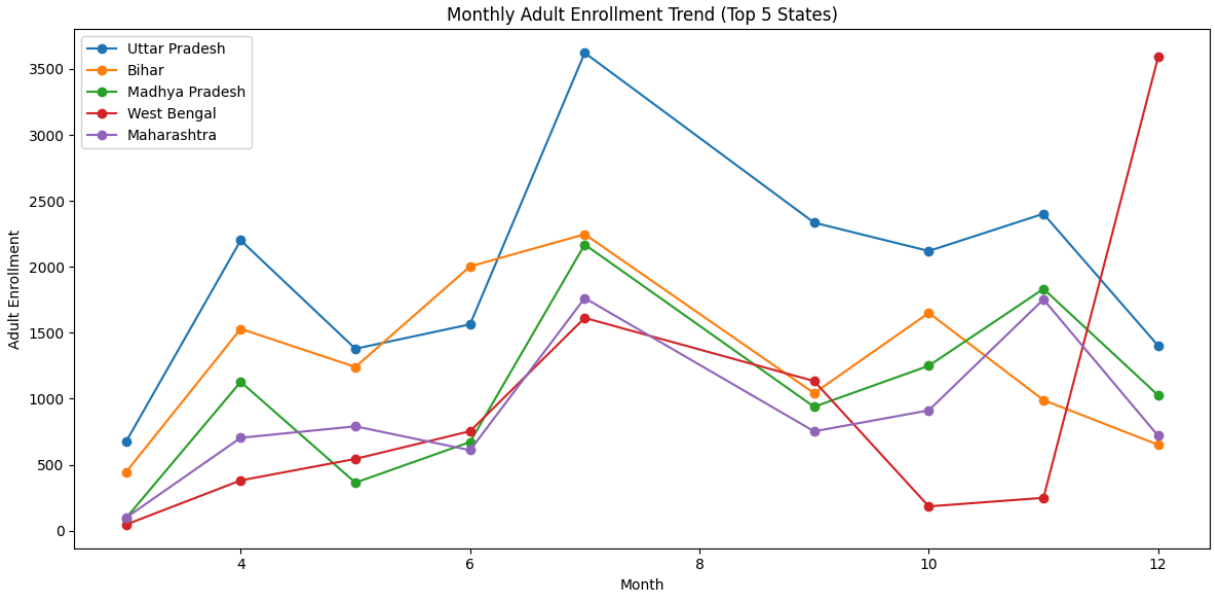
```
    figsize=(12,6)
)

plt.title('Top 10 States: Share of Total Aadhaar Enrollment (%)')
plt.xlabel('State')
plt.ylabel('Percentage Share')
plt.tight_layout()
plt.show()
```



State-wise Monthly Trend (Top 5 States)

In [310…

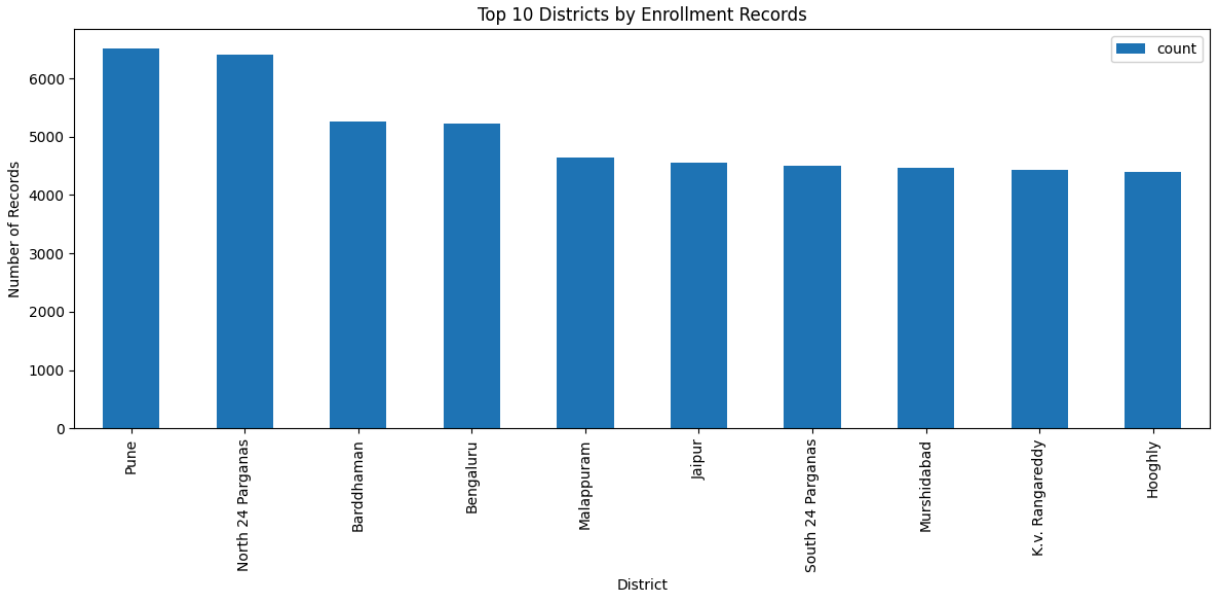
```
top5_states = top10.index[:5]

monthly_state = df[df['state_clean'].isin(top5_states)].groupby(
    ['state_clean','month']
)['age_18_greater'].sum().reset_index()

plt.figure(figsize=(12,6))

for state in top5_states:
    data = monthly_state[monthly_state['state_clean'] == state]
    plt.plot(data['month'], data['age_18_greater'], marker='o', label=state)

plt.title('Monthly Adult Enrollment Trend (Top 5 States)')
plt.xlabel('Month')
plt.ylabel('Adult Enrollment')
plt.legend()
plt.tight_layout()
plt.show()
```



District Concentration (Top Districts in Each State)

In [311…

```
district_count = df.groupby(['state_clean','district']).size().reset_index(r

top_districts = district_count.sort_values('count', ascending=False).head(10

top_districts.plot(
    x='district',
    y='count',
    kind='bar',
    figsize=(12,6)
)

plt.title('Top 10 Districts by Enrollment Records')
plt.xlabel('District')
plt.ylabel('Number of Records')
plt.tight_layout()
plt.show()
```



Performing Eda on Top Five States with heights Enrolment

Bihar

In [312…

```
# Extracting rows where state is Bihar
df_bihar= df[df['state_clean']=='Bihar']
df_bihar
```

Out[312…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	ye
5	2025-03-09	Bihar	Sitamarhi	843331	20	49	12	202
6	2025-03-09	Bihar	Sitamarhi	843330	23	24	42	202
9	2025-03-09	Bihar	Purbi Champaran	845418	30	48	10	202
11	2025-03-09	Bihar	Sitamarhi	843317	35	94	16	202
13	2025-03-09	Bihar	Sitamarhi	843324	49	186	34	202
...	...	...	...	...	...	...	...	
1002992	2025-12-31	Bihar	Vaishali	844134	2	0	0	202
1002993	2025-12-31	Bihar	Vaishali	844504	15	26	1	202
1002994	2025-12-31	Bihar	Vaishali	844509	1	2	0	202
1002995	2025-12-31	Bihar	West Champaran	845404	13	17	1	202
1002996	2025-12-31	Bihar	West Champaran	845449	9	45	0	202

58542 rows × 12 columns

In [313…
```
# Total unique districts in Bihar
df_bihar['district'].nunique()
```

Out[313… 48

In [314…
```
# Same District have more than one spelling so we have to map them
df_bihar['district'].unique()
```

Out[314…
```
array(['Sitamarhi', 'Purbi Champaran', 'Madhubani', 'Bhagalpur', 'Patna',
       'Pashchim Champaran', 'Muzaffarpur', 'Munger', 'Gaya',
       'Kaimur (Bhabua)', 'West Champaran', 'Purnia', 'Saran',
       'East Champaran', 'Vaishali', 'Jehanabad', 'Jamui', 'Gopalganj',
       'Saharsa', 'Arwal', 'Katihar', 'Siwan', 'Lakhisarai', 'Banka',
       'Nalanda', 'Araria', 'Darbhanga', 'Nawada', 'Samastipur',
       'Begusarai', 'Bhojpur', 'Aurangabad', 'Buxar', 'Khagaria',
       'Kishanganj', 'Madhepura', 'Rohtas', 'Sheohar', 'Supaul',
       'Aurangabad(bh)', 'Purba Champaran', 'Purnea', 'Sheikhpura',
       'Sheikpura', 'Bhabua', 'Monghyr', 'Samstipur', 'Aurangabad(BH)'],
      dtype=object)
```

Mapping District

In [315…
```python
import pandas as pd
import re

def clean_name(x):
    if pd.isna(x):
        return x
    x = str(x).lower()
    x = re.sub(r'[^a-z]', '', x)
    x = re.sub(r'\s+',' ', x).strip()
    return x
```

In [316…
```python
## District mapping bihar
bihar_district_mapping = {

    # Arwal
    "arwal": "Arwal",

    # Aurangabad
    "aurangabad": "Aurangabad",
    "aurangabadbh": "Aurangabad",

    # Araria
    "araria": "Araria",

    # Banka
    "banka": "Banka",

    # Begusarai
    "begusarai": "Begusarai",

    # Bhagalpur
    "bhagalpur": "Bhagalpur",

    # Bhojpur
    "bhojpur": "Bhojpur",

    # Buxar
    "buxar": "Buxar",
```

```
    # Darbhanga
    "darbhanga": "Darbhanga",

    # East Champaran
    "eastchamparan": "East Champaran",
    "purbachamparan": "East Champaran",

    # West Champaran
    "westchamparan": "West Champaran",
    "pashchimchamparan": "West Champaran",

    # Gaya
    "gaya": "Gaya",

    # Gopalganj
    "gopalganj": "Gopalganj",

    # Jamui
    "jamui": "Jamui",

    # Jehanabad
    "jehanabad": "Jehanabad",

    # Kaimur
    "kaimurbhabua": "Kaimur",
    "bhabua": "Kaimur",

    # Katihar
    "katihar": "Katihar",

    # Khagaria
    "khagaria": "Khagaria",

    # Kishanganj
    "kishanganj": "Kishanganj",

    # Lakhisarai
    "lakhisarai": "Lakhisarai",

    # Madhepura
    "madhepura": "Madhepura",

    # Madhubani
    "madhubani": "Madhubani",

    # Munger
    "munger": "Munger",
    "monghyr": "Munger",

    # Muzaffarpur
    "muzaffarpur": "Muzaffarpur",

    # Nalanda
    "nalanda": "Nalanda",

    # Nawada
```

```
    "nawada": "Nawada",

    # Patna
    "patna": "Patna",

    # Purnia
    "purnia": "Purnia",
    "purnea": "Purnia",

    # Rohtas
    "rohtas": "Rohtas",

    # Saharsa
    "saharsa": "Saharsa",

    # Samastipur
    "samastipur": "Samastipur",
    "samstipur": "Samastipur",

    # Saran
    "saran": "Saran",

    # Sheikhpura
    "sheikhpura": "Sheikhpura",
    "sheikpura": "Sheikhpura",

    # Sheohar
    "sheohar": "Sheohar",

    # Sitamarhi
    "sitamarhi": "Sitamarhi",

    # Siwan
    "siwan": "Siwan",

    # Supaul
    "supaul": "Supaul",

    # Vaishali
    "vaishali": "Vaishali",
}
```

In [317…

```
df['district_clean'] = (
    df['district']
    .apply(clean_name)
    .map(bihar_district_mapping)
    .fillna(df_bihar['district'])

)
```

In [318…

```
## Remaining unmapped
df[df['district_clean'].isna()]['district'].unique()
# count check
df['district_clean'].nunique()
```

Out[318… 39

In [319…
```
df_bihar = df[df['state_clean']=='Bihar']
df_bihar
```

Out[319…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	yea
5	2025-03-09	Bihar	Sitamarhi	843331	20	49	12	202
6	2025-03-09	Bihar	Sitamarhi	843330	23	24	42	202
9	2025-03-09	Bihar	Purbi Champaran	845418	30	48	10	202
11	2025-03-09	Bihar	Sitamarhi	843317	35	94	16	202
13	2025-03-09	Bihar	Sitamarhi	843324	49	186	34	202
...	...	...	...	...	...	...	...	
1002992	2025-12-31	Bihar	Vaishali	844134	2	0	0	202
1002993	2025-12-31	Bihar	Vaishali	844504	15	26	1	202
1002994	2025-12-31	Bihar	Vaishali	844509	1	2	0	202
1002995	2025-12-31	Bihar	West Champaran	845404	13	17	1	202
1002996	2025-12-31	Bihar	West Champaran	845449	9	45	0	202

58542 rows × 13 columns

In [320…
```
# Check Bihar-specific unmapped districts
df_bihar_unmapped = df_bihar[df_bihar['district_clean'].isna()]
print(f"Unmapped Bihar districts count: {len(df_bihar_unmapped)}")
df_bihar_unmapped['district'].unique()
```

```
Unmapped Bihar districts count: 0
```

Out[320… `array([], dtype=object)`

In [321…
```
# Final unique districts in Bihar after cleaning
df_bihar['district_clean'].unique()
```

Out[321…
```
array(['Sitamarhi', 'Purbi Champaran', 'Madhubani', 'Bhagalpur', 'Patna',
       'West Champaran', 'Muzaffarpur', 'Munger', 'Gaya', 'Kaimur',
       'Purnia', 'Saran', 'East Champaran', 'Vaishali', 'Jehanabad',
       'Jamui', 'Gopalganj', 'Saharsa', 'Arwal', 'Katihar', 'Siwan',
       'Lakhisarai', 'Banka', 'Nalanda', 'Araria', 'Darbhanga', 'Nawada',
       'Samastipur', 'Begusarai', 'Bhojpur', 'Aurangabad', 'Buxar',
       'Khagaria', 'Kishanganj', 'Madhepura', 'Rohtas', 'Sheohar',
       'Supaul', 'Sheikhpura'], dtype=object)
```

In [322…
```
# Total unique districts in Bihar after cleaning
df_bihar['district_clean'].nunique()
```

Out[322…
```
39
```

In [323…
```
# unique pincodes in bihar
df_bihar['pincode'].nunique()
```

Out[323…
```
906
```

In [324…
```
# Check unique pincodes per district in Bihar
pincode_check = df_bihar.groupby('district_clean')['pincode'].nunique().rese
pincode_check
```

Out[324...

	district_clean	unique_pincodes
0	Araria	19
1	Arwal	19
2	Aurangabad	29
3	Banka	32
4	Begusarai	33
5	Bhagalpur	34
6	Bhojpur	41
7	Buxar	27
8	Darbhanga	46
9	East Champaran	39
10	Gaya	39
11	Gopalganj	23
12	Jamui	14
13	Jehanabad	21
14	Kaimur	12
15	Katihar	23
16	Khagaria	15
17	Kishanganj	9
18	Lakhisarai	13
19	Madhepura	21
20	Madhubani	44
21	Munger	12
22	Muzaffarpur	53
23	Nalanda	31
24	Nawada	24
25	Patna	69
26	Purbi Champaran	12
27	Purnia	30
28	Rohtas	33
29	Saharsa	18
30	Samastipur	42
31	Saran	51

	district_clean	unique_pincodes
32	Sheikhpura	8
33	Sheohar	7
34	Sitamarhi	25
35	Siwan	46
36	Supaul	23
37	Vaishali	38
38	West Champaran	19

In [326…
```
# Pincode associated with multiple districts in Bihar
pin_district_count = (
    df_bihar.groupby('pincode')['district_clean']
    .nunique()
    .reset_index(name='district_count')
)
```

In [327…
```
pin_district_count
```

Out[327…

	pincode	district_count
0	800001	1
1	800002	1
2	800003	1
3	800004	1
4	800005	1
…	…	…
901	855114	1
902	855115	2
903	855116	1
904	855117	1
905	855456	1

906 rows × 2 columns

In [328…
```
# Extract problematic pincodes (associated with >1 district)
problem_pins = pin_district_count[
    pin_district_count['district_count'] > 1
]
```

In [329…
```
problem_pins
```

Out[329…

	pincode	district_count
40	801304	2
41	801305	2
53	802112	2
73	802134	2
83	802160	2
…	…	…
890	854337	2
894	855101	3
896	855105	2
898	855107	2
902	855115	2

177 rows × 2 columns

In [330…

```
# Get all records with problematic pincodes(flagged records)
df_flagged = df_bihar.merge(
    problem_pins[['pincode']],
    on='pincode',
    how='inner'
)
```

In [331…

```
df_flagged
```

Out [331…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	year
0	2025-03-09	Bihar	Purbi Champaran	845418	30	48	10	2025
1	2025-03-09	Bihar	Purbi Champaran	845304	18	72	12	2025
2	2025-03-15	Bihar	Purbi Champaran	845303	12	121	13	2025
3	2025-04-01	Bihar	Sitamarhi	843315	102	125	18	2025
4	2025-04-01	Bihar	Munger	811213	191	278	22	2025
...	...	...	...	...	...	...	...	...
16363	2025-12-31	Bihar	Sheohar	843325	4	2	0	2025
16364	2025-12-31	Bihar	Sitamarhi	843325	11	6	0	2025
16365	2025-12-31	Bihar	Siwan	841243	2	11	0	2025
16366	2025-12-31	Bihar	Supaul	852108	0	9	0	2025
16367	2025-12-31	Bihar	Supaul	852131	15	19	0	2025

16368 rows × 13 columns

In [332…

```
# Summary of flagged records by district and pincode(for review)
flagged_pincode=df_flagged.groupby(['district_clean','pincode'])[['age_0_5',
# flagged_pincode.to_excel('flagged_pincode_domain.xlsx')
```

In [333…

```
# Add total enrollment column to flagged_pincode
flagged_pincode['total_enrollment']=flagged_pincode['age_0_5']+flagged_pincc
flagged_pincode
```

Out[333…

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Araria	854102	127	84	1	212
1	Araria	854201	28	30	0	58
2	Araria	854202	57	73	0	130
3	Araria	854304	410	412	14	836
4	Araria	854312	831	540	6	1377
...	...	...	...	...	...	...
360	Vaishali	843104	60	34	0	94
361	Vaishali	843105	7	2	0	9
362	Vaishali	844111	143	183	0	326
363	Vaishali	844112	177	185	1	363
364	Vaishali	844120	22	30	0	52

365 rows × 6 columns

In [334…

```
idx = flagged_pincode.groupby('pincode')['total_enrollment'].idxmax()
df_filtered = flagged_pincode.loc[idx]
df_filtered
```

Out[334…

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
223	Nalanda	801304	57	114	0	171
224	Nalanda	801305	38	62	1	101
63	Buxar	802112	114	212	1	327
64	Buxar	802134	117	357	1	475
59	Bhojpur	802160	66	268	2	336
...	...	...	...	...	...	...
281	Purnia	854337	323	127	0	450
165	Kishanganj	855101	1745	334	5	2084
158	Katihar	855105	361	120	6	487
166	Kishanganj	855107	1576	406	1	1983
167	Kishanganj	855115	735	198	4	937

177 rows × 6 columns

In [335…

```
# Creating a flag for pincodes associated with multiple districts
df_bihar['pin_multi_district_flag']=(
    df_bihar.groupby('pincode')['district_clean']
```

```
    .transform('nunique')>1
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/2091407170.p
y:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_bihar['pin_multi_district_flag']=(
```

In [336…
```
# Creating a flag for pincodes associated with multiple districts
pin_district_map= (
    df_bihar[df_bihar['pin_multi_district_flag']]
    .groupby('pincode')['district_clean']  # noqa: SC100
    .unique()
    .reset_index()
)
```

In [337…
```
## monthly enrolment check
df_bihar['month'] = df_bihar['date'].dt.month.astype(str).str.zfill(2)
df_bihar
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/3313290577.p
y:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_bihar['month'] = df_bihar['date'].dt.month.astype(str).str.zfill(2)
```

Out [337…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	ye
5	2025-03-09	Bihar	Sitamarhi	843331	20	49	12	202
6	2025-03-09	Bihar	Sitamarhi	843330	23	24	42	202
9	2025-03-09	Bihar	Purbi Champaran	845418	30	48	10	202
11	2025-03-09	Bihar	Sitamarhi	843317	35	94	16	202
13	2025-03-09	Bihar	Sitamarhi	843324	49	186	34	202
...	...	...	...	...	...	...	...	
1002992	2025-12-31	Bihar	Vaishali	844134	2	0	0	202
1002993	2025-12-31	Bihar	Vaishali	844504	15	26	1	202
1002994	2025-12-31	Bihar	Vaishali	844509	1	2	0	202
1002995	2025-12-31	Bihar	West Champaran	845404	13	17	1	202
1002996	2025-12-31	Bihar	West Champaran	845449	9	45	0	202

58542 rows × 14 columns

In [338…

```
# Dropping Date District and state as we have District clean State clean
df_bihar_cleaned=df_bihar.drop(columns=['date','district','state'], axis=1)
df_bihar_cleaned
```

Out [338…

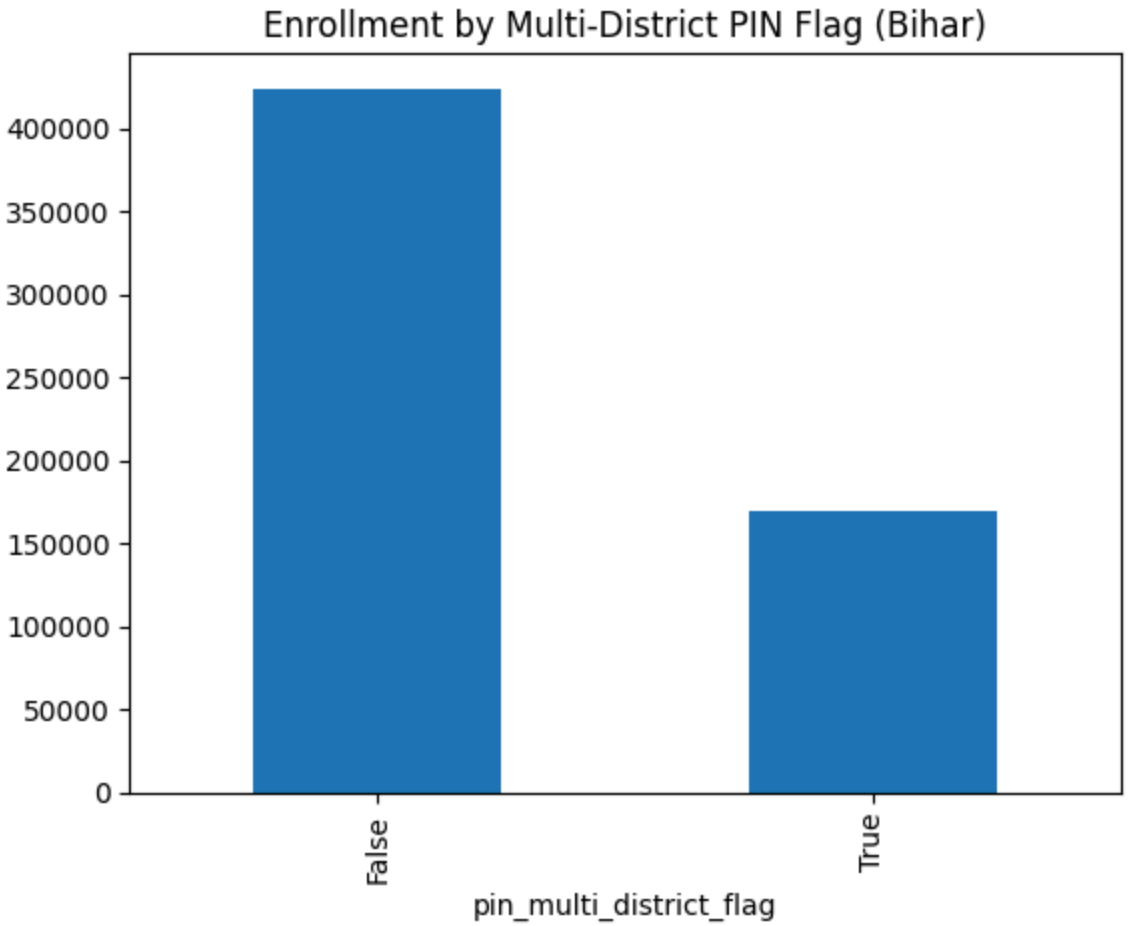
	pincode	age_0_5	age_5_17	age_18_greater	year	month	month_name	s
5	843331	20	49	12	2025	03	Mar	
6	843330	23	24	42	2025	03	Mar	
9	845418	30	48	10	2025	03	Mar	
11	843317	35	94	16	2025	03	Mar	
13	843324	49	186	34	2025	03	Mar	
...	...	...	...	...	...	...	...	
1002992	844134	2	0	0	2025	12	Dec	
1002993	844504	15	26	1	2025	12	Dec	
1002994	844509	1	2	0	2025	12	Dec	
1002995	845404	13	17	1	2025	12	Dec	
1002996	845449	9	45	0	2025	12	Dec	

58542 rows × 11 columns

Flagged vs unflagged pincodes

In [380…

```
df_bihar.groupby('pin_multi_district_flag')['total_enrollment'].sum().plot(
    kind='bar'
)
plt.title('Enrollment by Multi-District PIN Flag (Bihar)')
plt.show()
```



In [339…
```
# Aggregate Bihar enrollment data at the district level by summing enrollmen
# across different age groups (0–5, 5–17, and 18+).
df_bihar_dist_level = df_bihar_cleaned.groupby('district_clean')[
    ['age_0_5', 'age_5_17', 'age_18_greater']
].sum()

# Compute total enrollment for each district by adding all age-group enrollm
df_bihar_dist_level['total_enrollment'] = (
    df_bihar_dist_level['age_0_5'] +
    df_bihar_dist_level['age_5_17'] +
    df_bihar_dist_level['age_18_greater']
)
```

In [340…
```
df_bihar_dist_level.shape
```

Out[340… (39, 4)

In [341…
```
df_bihar_dist_level
```

Out [341...

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Araria	9646	5357	124	15127
Arwal	777	2939	33	3749
Aurangabad	2922	6927	72	9921
Banka	5945	4669	79	10693
Begusarai	7719	4851	55	12625
Bhagalpur	10196	10025	375	20596
Bhojpur	3043	9832	104	12979
Buxar	2407	4453	60	6920
Darbhanga	9633	5466	68	15167
East Champaran	10003	18105	792	28900
Gaya	6540	19786	434	26760
Gopalganj	5195	9461	192	14848
Jamui	5042	4897	159	10098
Jehanabad	888	3683	139	4710
Kaimur	2938	4631	84	7653
Katihar	10877	4757	113	15747
Khagaria	4167	3653	67	7887
Kishanganj	6527	1680	23	8230
Lakhisarai	2599	3364	68	6031
Madhepura	4371	4225	38	8634
Madhubani	12324	12275	791	25390
Munger	2397	3466	102	5965
Muzaffarpur	13787	13854	657	28298
Nalanda	3527	10452	134	14113
Nawada	2661	12099	288	15048
Patna	6559	16758	744	24061
Purbi Champaran	4000	10071	800	14871
Purnia	11978	7687	183	19848
Rohtas	2623	4974	76	7673
Saharsa	6054	6335	201	12590
Samastipur	10877	7571	131	18579

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Saran	7401	16040	217	23658
Sheikhpura	1191	1863	19	3073
Sheohar	1900	1119	11	3030
Sitamarhi	20358	18600	2694	41652
Siwan	5961	8778	135	14874
Supaul	5473	4408	155	10036
Vaishali	7972	9424	52	17448
West Champaran	16433	28508	1330	46271

In [342…

```
# Sorting District with total enrollment
df_bihar_dist_level.sort_values("total_enrollment",ascending=False).reset_ir
# df_bihar_dist_level.to_excel('bihar_district_level_enrolment.xlsx')
```

Out [342...

	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
0	West Champaran	16433	28508	1330	46271
1	Sitamarhi	20358	18600	2694	41652
2	East Champaran	10003	18105	792	28900
3	Muzaffarpur	13787	13854	657	28298
4	Gaya	6540	19786	434	26760
5	Madhubani	12324	12275	791	25390
6	Patna	6559	16758	744	24061
7	Saran	7401	16040	217	23658
8	Bhagalpur	10196	10025	375	20596
9	Purnia	11978	7687	183	19848
10	Samastipur	10877	7571	131	18579
11	Vaishali	7972	9424	52	17448
12	Katihar	10877	4757	113	15747
13	Darbhanga	9633	5466	68	15167
14	Araria	9646	5357	124	15127
15	Nawada	2661	12099	288	15048
16	Siwan	5961	8778	135	14874
17	Purbi Champaran	4000	10071	800	14871
18	Gopalganj	5195	9461	192	14848
19	Nalanda	3527	10452	134	14113
20	Bhojpur	3043	9832	104	12979
21	Begusarai	7719	4851	55	12625
22	Saharsa	6054	6335	201	12590
23	Banka	5945	4669	79	10693
24	Jamui	5042	4897	159	10098
25	Supaul	5473	4408	155	10036
26	Aurangabad	2922	6927	72	9921
27	Madhepura	4371	4225	38	8634
28	Kishanganj	6527	1680	23	8230
29	Khagaria	4167	3653	67	7887
30	Rohtas	2623	4974	76	7673
31	Kaimur	2938	4631	84	7653

	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
32	Buxar	2407	4453	60	6920
33	Lakhisarai	2599	3364	68	6031
34	Munger	2397	3466	102	5965
35	Jehanabad	888	3683	139	4710
36	Arwal	777	2939	33	3749
37	Sheikhpura	1191	1863	19	3073
38	Sheohar	1900	1119	11	3030

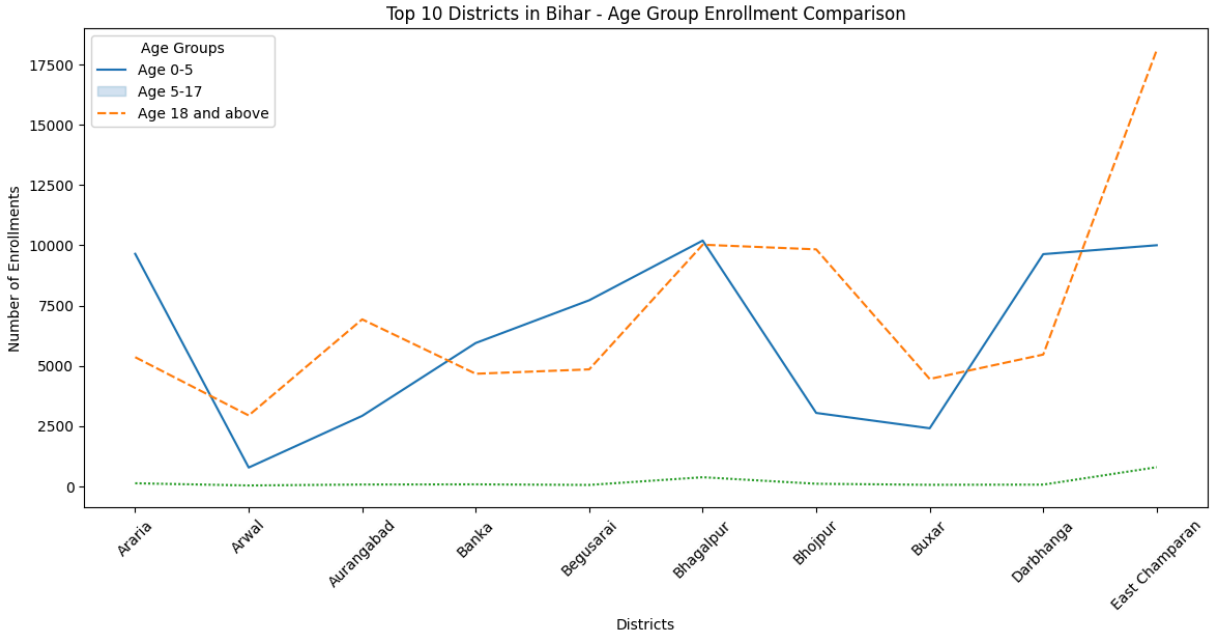
Plotting Top 10 Districts by number of Enrollment

Visualize age-wise enrollment distribution across top 10 Bihar districts using a line plot

In [343…
```
df_bihar_dist_level1 = df_bihar_dist_level.head(10)
```

In [344…
```
# Import required visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Set the figure size for better readability
plt.figure(figsize=(14,6))

# Plot line chart to compare enrollment trends across age groups
# for the top 10 districts in Bihar
sns.lineplot(
    data=df_bihar_dist_level1[['age_0_5', 'age_5_17', 'age_18_greater']]
)

# Add title and axis labels
plt.title('Top 10 Districts in Bihar - Age Group Enrollment Comparison')
plt.xlabel('Districts')
plt.ylabel('Number of Enrollments')

# Customize legend to clearly represent age groups
plt.legend(
    title='Age Groups',
    labels=['Age 0-5', 'Age 5-17', 'Age 18 and above']
)

# Set district names on x-axis and rotate labels for clarity
plt.xticks(
    ticks=range(len(df_bihar_dist_level1.index)),
    labels=df_bihar_dist_level1.index,
    rotation=45
```

```
)

# Display the plot
plt.show()
```



Plotting month vs total enrollment

In [345…
```
## monthly enrolment trend in bihar
df_bihar_monthly = df_bihar_cleaned.groupby('month')[['age_0_5','age_5_17','
df_bihar_monthly['total_enrollment'] = df_bihar_monthly['age_0_5'] + df_biha
```

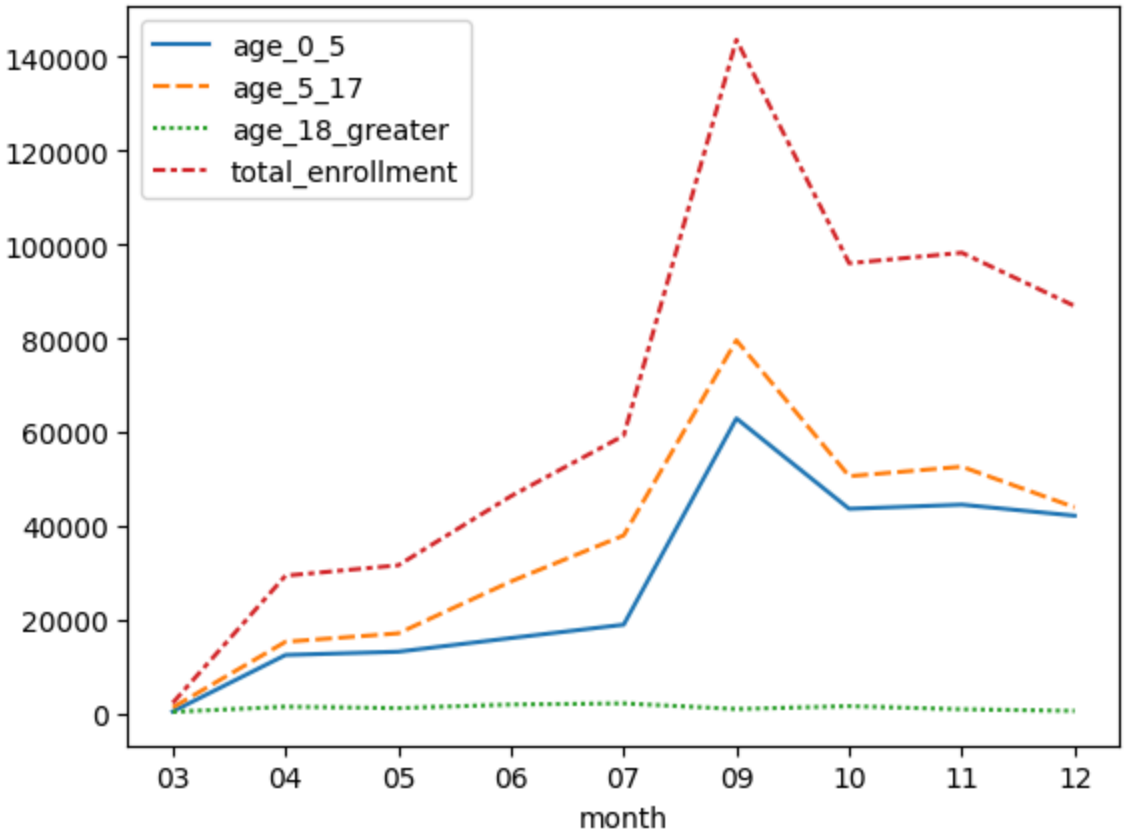
In [346…
```
df_bihar_monthly.sort_values("total_enrollment",ascending=False).reset_index
```

Out[346…

	month	age_0_5	age_5_17	age_18_greater	total_enrollment
0	09	62940	79583	1042	143565
1	11	44572	52661	990	98223
2	10	43690	50636	1651	95977
3	12	42223	44009	651	86883
4	07	19008	38019	2247	59274
5	06	16160	28223	2003	46386
6	05	13251	17159	1241	31651
7	04	12551	15361	1530	29442
8	03	516	1392	444	2352

In [347…
```
sns.lineplot(data=df_bihar_monthly)
```

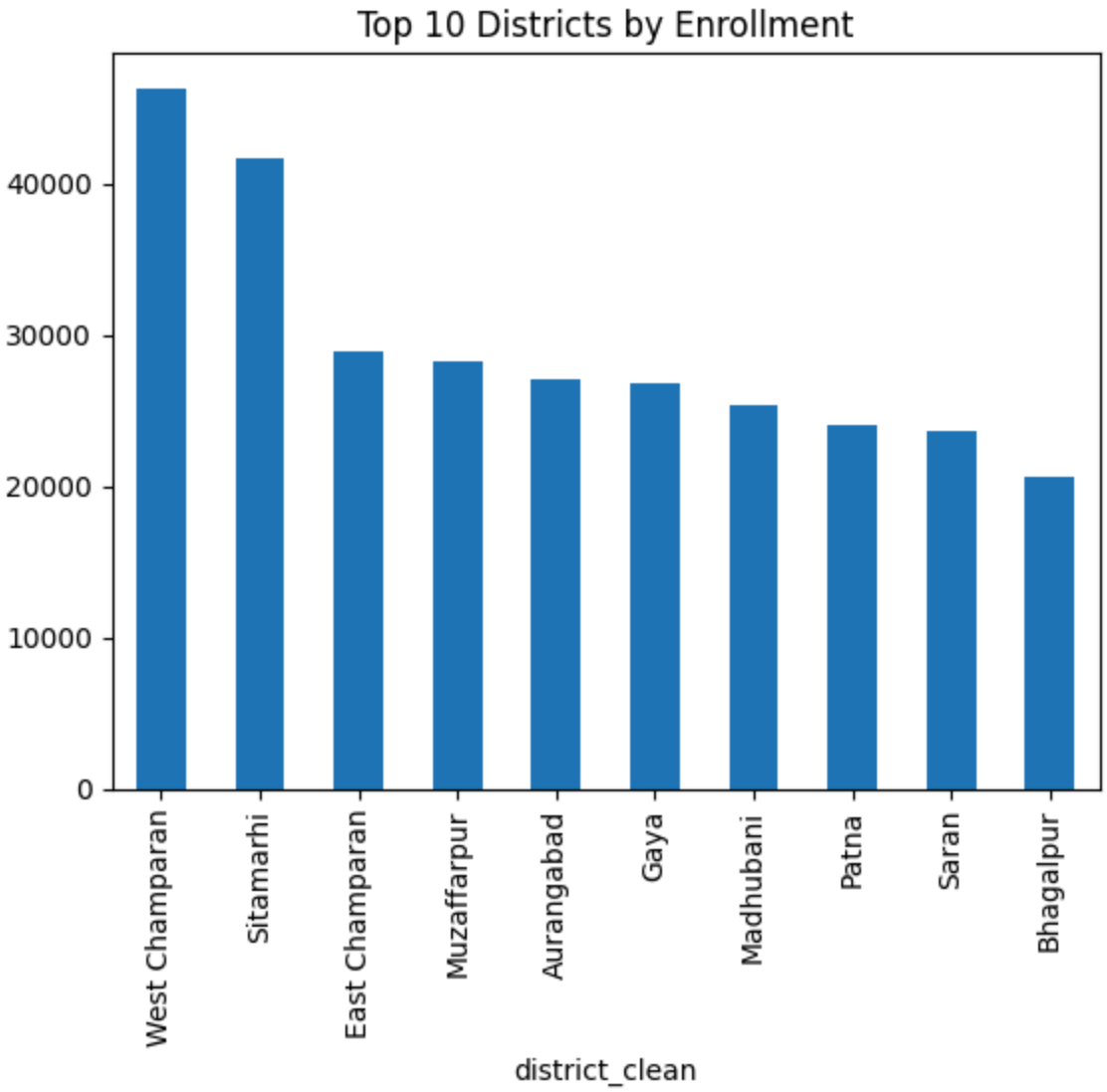
Out[347… <Axes: xlabel='month'>



Top 10 Districts by Enrollment

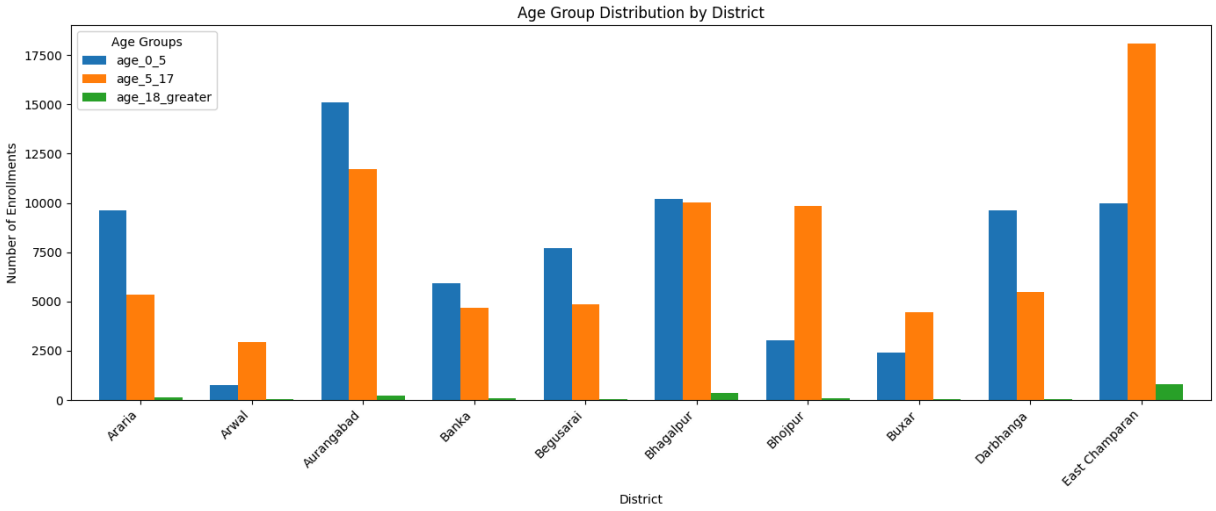
In [374…

```
top_districts = df.groupby('district_clean')['total_enrollment'] \
                  .sum().sort_values(ascending=False).head(10)

top_districts.plot(kind='bar')
plt.title('Top 10 Districts by Enrollment')
plt.show()
```



In [377…

```python
import matplotlib.pyplot as plt

district_age = df.groupby('district_clean')[
    ['age_0_5','age_5_17','age_18_greater']
].sum().head(10)

ax = district_age.plot(
    kind='bar',
    stacked=False,      # important → no overlap
    figsize=(14,6),
    width=0.75
)

plt.title('Age Group Distribution by District')
plt.xlabel('District')
plt.ylabel('Number of Enrollments')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Age Groups')
plt.tight_layout()
plt.show()
```

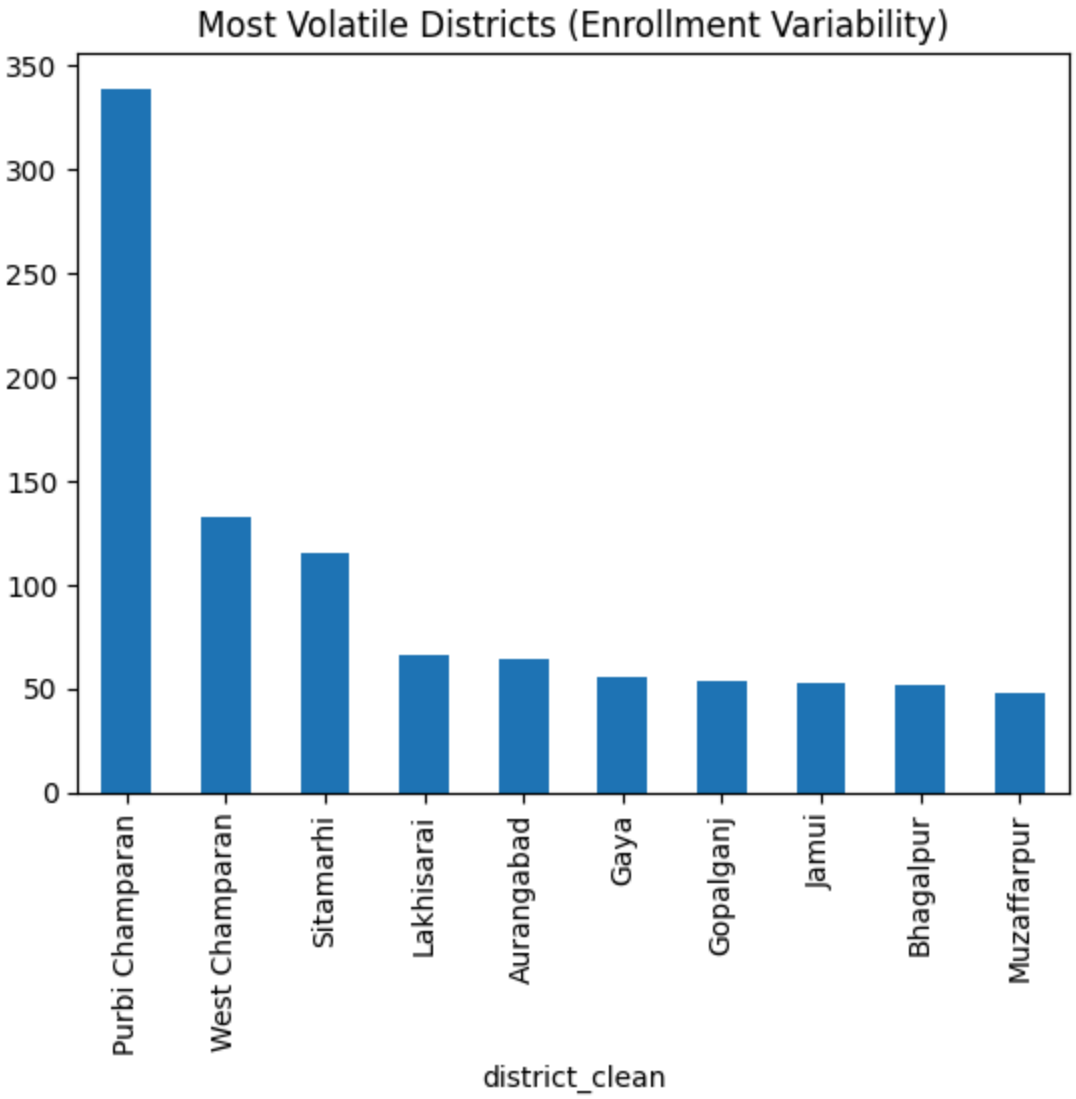


📊 Enrollment Variability Across Districts

This bar chart highlights the top 10 districts with the highest standard deviation in total enrollment, indicating significant variability over time. High volatility suggests inconsistent enrollment patterns, possibly due to seasonal drives, migration, or administrative factors.

In [383…

```
district_std = df.groupby('district_clean')['total_enrollment'].std()

district_std.sort_values(ascending=False).head(10).plot(kind='bar')
plt.title('Most Volatile Districts (Enrollment Variability)')
plt.show()
```



📌 District-wise Enrollment Ranking Analysis

This analysis ranks districts based on their **total Aadhaar enrollment** over the entire study period.

In [384…

```
district_rank = df.groupby('district_clean')['total_enrollment'].sum() \
                  .rank(ascending=False)

district_rank.sort_values().head(10)
```

Out[384…
```
district_clean
West Champaran     1.0
Sitamarhi          2.0
East Champaran     3.0
Muzaffarpur        4.0
Aurangabad         5.0
Gaya               6.0
Madhubani          7.0
Patna              8.0
Saran              9.0
Bhagalpur         10.0
Name: total_enrollment, dtype: float64
```

We performed univariate, bivariate, and multivariate analysis to study

enrollment distribution across states, districts, and age groups.

Advanced analysis included district-level volatility, age composition,

time-based trends, and flag-based segmentation to uncover hidden patterns.

Uttar Pradesh ke liye

In [351…
```
df_uttarpradesh= df[df['state_clean']=='Uttar Pradesh']
df_uttarpradesh
```

Out [351…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
2	2025-03-09	Uttar Pradesh	Kanpur Nagar	208001	29	82	12
3	2025-03-09	Uttar Pradesh	Aligarh	202133	62	29	15
7	2025-03-09	Uttar Pradesh	Bahraich	271865	26	60	14
8	2025-03-09	Uttar Pradesh	Firozabad	283204	28	26	10
10	2025-03-09	Uttar Pradesh	Maharajganj	273164	31	70	13
...	...	...	...	...	...	...	...
1005743	2025-12-31	Uttar Pradesh	Varanasi	221002	10	18	0
1005744	2025-12-31	Uttar Pradesh	Varanasi	221104	4	11	0
1005745	2025-12-31	Uttar Pradesh	Varanasi	221107	1	15	0
1005746	2025-12-31	Uttar Pradesh	Varanasi	221207	1	9	0
1005747	2025-12-31	Uttar Pradesh	Varanasi	221313	1	0	0

108066 rows × 13 columns

In [352…

```
df_uttarpradesh['district'].unique()
```

Out[352…
```
array(['Kanpur Nagar', 'Aligarh', 'Bahraich', 'Firozabad', 'Maharajganj',
       'Ghaziabad', 'Gautam Buddha Nagar', 'Lucknow', 'Agra', 'Unnao',
       'Saharanpur', 'Jaunpur', 'Gorakhpur', 'Bulandshahr', 'Mathura',
       'Banda', 'Kheri', 'Budaun', 'Kanpur Dehat', 'Varanasi', 'Baghpat',
       'Fatehpur', 'Etawah', 'Shamli', 'Balrampur', 'Bara Banki',
       'Shahjahanpur', 'Gonda', 'Bareilly', 'Sitapur', 'Sultanpur',
       'Shrawasti', 'Chandauli', 'Mainpuri', 'Muzaffarnagar',
       'Siddharthnagar', 'Ambedkar Nagar', 'Pilibhit', 'Kaushambi',
       'Jalaun', 'Etah', 'Meerut', 'Basti', 'Azamgarh', 'Rampur',
       'Moradabad', 'Amroha', 'Bijnor', 'Deoria', 'Prayagraj', 'Lalitpur',
       'Hapur', 'Hathras', 'Kushinagar', 'Shravasti', 'Farrukhabad',
       'Hardoi', 'Ayodhya', 'Siddharth Nagar', 'Barabanki', 'Sambhal',
       'Jhansi', 'Kasganj', 'Kannauj', 'Kushi Nagar', 'Allahabad',
       'Amethi', 'Auraiya', 'Ballia', 'Bhadohi', 'Chitrakoot', 'Faizabad',
       'Ghazipur', 'Mahoba', 'Mau', 'Mirzapur', 'Pratapgarh',
       'Rae Bareli', 'Sant Kabir Nagar', 'Sant Ravidas Nagar',
       'Sonbhadra', 'Bulandshahar', 'Hamirpur', 'Jyotiba Phule Nagar',
       'Sant Ravidas Nagar Bhadohi', 'Raebareli', 'Mahrajganj',
       'Kushinagar *', 'Bagpat'], dtype=object)
```

In [353…
```
df_uttarpradesh['district'].nunique()
```

Out[353… 89

In [354…
```
df_uttarpradesh['district_clean'] = (
    df_uttarpradesh['district']
    .str.lower()
    .str.strip()
    .str.replace(r'\*', '', regex=True)
    .str.replace(r'\(.*?\)', '', regex=True)
    .str.replace(r'[^a-z\s]', '', regex=True)
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/1172066687.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_uttarpradesh['district_clean'] = (
```

District names contained spelling variations and legacy names. A canonical district mapping was applied to standardize district identities before analysis, preventing double counting.

In [355…
```
district_standard_map = {
    "bara banki": "Barabanki",
    "barabanki": "Barabanki",

    "bulandshahar": "Bulandshahr",
    "bulandshahr": "Bulandshahr",

    "siddharth nagar": "Siddharthnagar",
    "siddharthnagar": "Siddharthnagar",
```

```
    "kushi nagar": "Kushinagar",
    "kushinagar *": "Kushinagar",
    "kushinagar": "Kushinagar",

    "shrawasti": "Shravasti",
    "shravasti": "Shravasti",

    "mahrajganj": "Maharajganj",
    "maharajganj": "Maharajganj",

    "sant ravidas nagar bhadohi": "Bhadohi",
    "sant ravidas nagar": "Bhadohi",
    "bhadohi": "Bhadohi",

    "jyotiba phule nagar": "Amroha",
    "amroha": "Amroha",

    "allahabad": "Prayagraj",
    "prayagraj": "Prayagraj",

    "faizabad": "Ayodhya",
    "ayodhya": "Ayodhya",

    "rae bareli": "Raebareli",
    "raebareli": "Raebareli",

    "bagpat": "Baghpat",
    "baghpat": "Baghpat"
}
```

In [356…
```
df_uttarpradesh['district_clean'] = (
    df['district']
    .apply(clean_name)
    .map(district_standard_map)
    .fillna(df_uttarpradesh['district'])

)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/973221572.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_uttarpradesh['district_clean'] = (
```

In [357…
```
## Remaining unmapped
df_uttarpradesh[df_uttarpradesh['district_clean'].isna()]['district'].unique
# count check
df_uttarpradesh['district_clean'].nunique()
```

Out[357… 78

In [358…
```
df_uttarpradesh.isnull().sum()
```

Out[358…
```
date                0
state               0
district            0
pincode             0
age_0_5             0
age_5_17            0
age_18_greater      0
year                0
month               0
month_name          0
state_clean         0
total_enrollment    0
district_clean      0
dtype: int64
```

In [359…
```
df_uttarpradesh
```

Out [359…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
2	2025-03-09	Uttar Pradesh	Kanpur Nagar	208001	29	82	12
3	2025-03-09	Uttar Pradesh	Aligarh	202133	62	29	15
7	2025-03-09	Uttar Pradesh	Bahraich	271865	26	60	14
8	2025-03-09	Uttar Pradesh	Firozabad	283204	28	26	10
10	2025-03-09	Uttar Pradesh	Maharajganj	273164	31	70	13
...	...	...	...	...	...	...	...
1005743	2025-12-31	Uttar Pradesh	Varanasi	221002	10	18	0
1005744	2025-12-31	Uttar Pradesh	Varanasi	221104	4	11	0
1005745	2025-12-31	Uttar Pradesh	Varanasi	221107	1	15	0
1005746	2025-12-31	Uttar Pradesh	Varanasi	221207	1	9	0
1005747	2025-12-31	Uttar Pradesh	Varanasi	221313	1	0	0

108066 rows × 13 columns

In [360…

```
## total enrolment by district in uttar pradesh
df_uttarpradesh_dist_level = df_uttarpradesh.groupby('district_clean')[['age
```

In [361…

```
df_uttarpradesh['total_enrollment'] = (
    df_uttarpradesh['age_0_5'] +
    df_uttarpradesh['age_5_17'] +
    df_uttarpradesh['age_18_greater']
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/1685002294.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_uttarpradesh['total_enrollment'] = (
```

In [362…
```
df_uttarpradesh
```

Out[362…

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
2	2025-03-09	Uttar Pradesh	Kanpur Nagar	208001	29	82	12
3	2025-03-09	Uttar Pradesh	Aligarh	202133	62	29	15
7	2025-03-09	Uttar Pradesh	Bahraich	271865	26	60	14
8	2025-03-09	Uttar Pradesh	Firozabad	283204	28	26	10
10	2025-03-09	Uttar Pradesh	Maharajganj	273164	31	70	13
...	...	...	...	...	...	...	...
1005743	2025-12-31	Uttar Pradesh	Varanasi	221002	10	18	0
1005744	2025-12-31	Uttar Pradesh	Varanasi	221104	4	11	0
1005745	2025-12-31	Uttar Pradesh	Varanasi	221107	1	15	0
1005746	2025-12-31	Uttar Pradesh	Varanasi	221207	1	9	0
1005747	2025-12-31	Uttar Pradesh	Varanasi	221313	1	0	0

108066 rows × 13 columns

In [363…
```
# enrolment top 10 district
df_district_level = df_uttarpradesh["district_clean"].head(10)
```

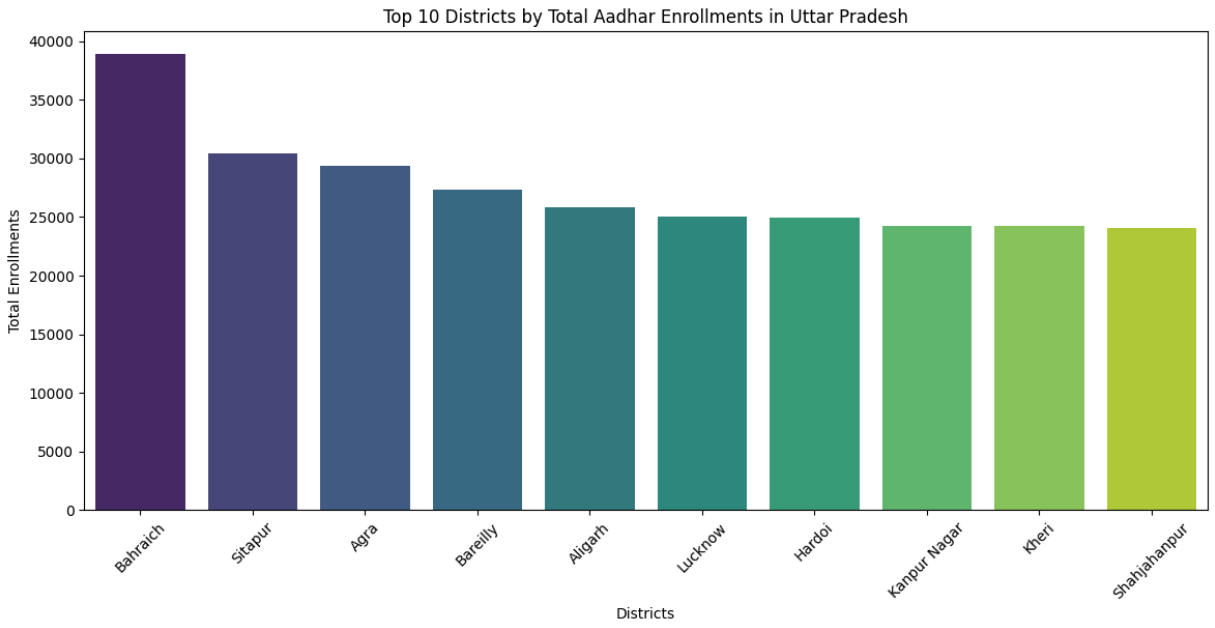
In [364…
```
## total enrolment by district in uttar pradesh
state_summary2 = df_uttarpradesh.groupby('district_clean')[[
    'age_0_5', 'age_5_17', 'age_18_greater'
]].sum() ## means of all age group by district

state_summary2['total_enrollment'] = (
    state_summary2['age_0_5'] +
    state_summary2['age_5_17'] +
    state_summary2['age_18_greater'] ## total enrolment by district
)

top10_districts = state_summary2.sort_values(
```

```
    by='total_enrollment', ascending=False
).head(10)
```

In [365…

```
## bar plot for top 10 states
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.barplot(
    x=top10_districts.index,
    y=top10_districts['total_enrollment'],
    palette='viridis'
)
plt.title('Top 10 Districts by Total Aadhar Enrollments in Uttar Pradesh')
plt.xlabel('Districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.show()
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_4285/3726560018.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



In [366…

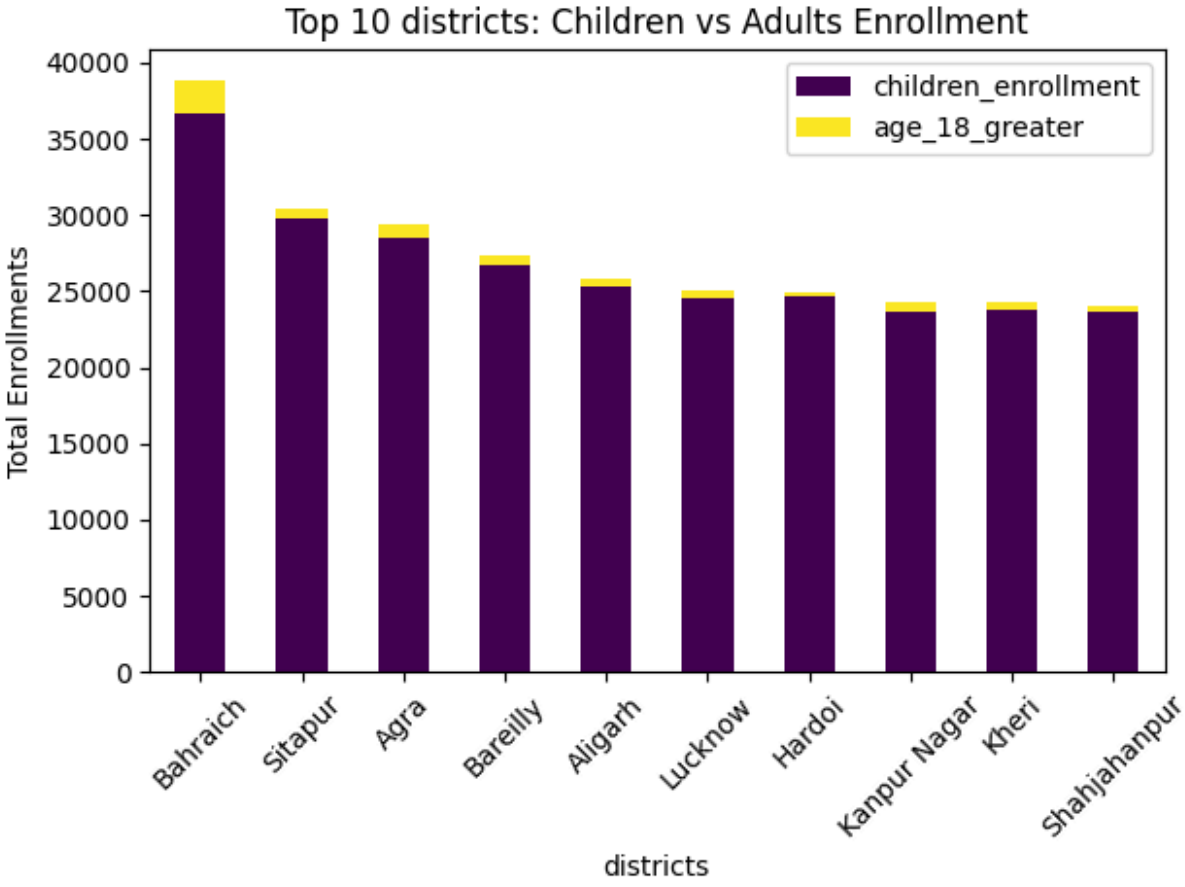
```
top10_districts['children_enrollment'] = (
    top10_districts['age_0_5'] +
    top10_districts['age_5_17']
)
```

In [367…

```
# Stacked bar plot for age group comparison
plt.figure(figsize=(14,6))
top10_districts[['children_enrollment', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
```

```
    colormap='viridis'
)
plt.title('Top 10 districts: Children vs Adults Enrollment')
plt.xlabel('districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<Figure size 1400x600 with 0 Axes>
```



In [368…

```
df_uttarpradesh['new_date'].isnull().sum()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File /Library/Frameworks/Python.framework/Versions/3.14/lib/python3.14/site-
packages/pandas/core/indexes/base.py:3812, in Index.get_loc(self, key)
   3811 try:
-> 3812     return self._engine.get_loc(casted_key)
   3813 except KeyError as err:

File pandas/_libs/index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:7088, in pandas._libs.hashtabl
e.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:7096, in pandas._libs.hashtabl
e.PyObjectHashTable.get_item()

KeyError: 'new_date'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[368], line 1
----> 1 df_uttarpradesh[          ].isnull().sum()

File /Library/Frameworks/Python.framework/Versions/3.14/lib/python3.14/site-
packages/pandas/core/frame.py:4113, in DataFrame.__getitem__(self, key)
   4111 if self.columns.nlevels > 1:
   4112     return self._getitem_multilevel(key)
-> 4113 indexer = self.columns.get_loc(key)
   4114 if is_integer(indexer):
   4115     indexer = [indexer]

File /Library/Frameworks/Python.framework/Versions/3.14/lib/python3.14/site-
packages/pandas/core/indexes/base.py:3819, in Index.get_loc(self, key)
   3814     if isinstance(casted_key, slice) or (
   3815         isinstance(casted_key, abc.Iterable)
   3816         and any(isinstance(x, slice) for x in casted_key)
   3817     ):
   3818         raise InvalidIndexError(key)
-> 3819     raise KeyError(key) from err
   3820 except TypeError:
   3821     # If we have a listlike key, _check_indexing_error will raise
   3822     #  InvalidIndexError. Otherwise we fall through and re-raise
   3823     #  the TypeError.
   3824     self._check_indexing_error(key)

KeyError: 'new_date'
```

In [ ]:
```
pincode_check_UP = df_uttarpradesh.groupby('district_clean')['pincode'].nuni
pincode_check_UP
```

Out[ ]:

	district_clean	unique_pincodes
0	Agra	29
1	Aligarh	39
2	Ambedkar Nagar	34
3	Amethi	39
4	Amroha	14
...	...	...
73	Sitapur	25
74	Sonbhadra	20
75	Sultanpur	35
76	Unnao	31
77	Varanasi	42

78 rows × 2 columns

In [ ]:
```
pin_district_count_UP = (
    df_uttarpradesh.groupby('pincode')['district_clean']
    .nunique()
    .reset_index(name='district_count')
)
```

In [ ]:
```
pin_district_count_UP
```

Out[ ]:

	pincode	district_count
0	121705	1
1	201001	2
2	201002	2
3	201003	1
4	201004	1
...	...	...
1732	285203	1
1733	285204	1
1734	285205	2
1735	285206	1
1736	285223	2

1737 rows × 2 columns

In [ ]:
```
problem_pins_UP = pin_district_count_UP[
    pin_district_count_UP['district_count'] > 1
]
```

In [ ]:
```
problem_pins_UP
## ek pin code 2 district se belong kr skta hai theek ye govt ki website pr
```

Out[ ]:

	pincode	district_count
1	201001	2
2	201002	2
5	201005	2
6	201006	2
7	201007	2
...	...	...
1713	284403	2
1724	285125	2
1725	285126	2
1734	285205	2
1736	285223	2

276 rows × 2 columns

In [ ]:
```
df_flagged_UP = df_uttarpradesh.merge(
    problem_pins_UP[['pincode']],
    on='pincode',
    how='inner'
)
```

In [ ]:
```
## ye sab o hai jissme ek district ke 2 pincode hai
## yha se hum pta kr skte hai kiss district me jda use ho rha hai
df_flagged_UP
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
0	09-03-2025	Uttar Pradesh	Kanpur Nagar	208001	29	82	12
1	09-03-2025	Uttar Pradesh	Bahraich	271865	26	60	14
2	09-03-2025	Uttar Pradesh	Ghaziabad	201102	50	113	11
3	15-03-2025	Uttar Pradesh	Ghaziabad	201001	33	125	16
4	15-03-2025	Uttar Pradesh	Ghaziabad	201102	19	146	30
...	...	...	...	...	...	...	...
23723	31-12-2025	Uttar Pradesh	Siddharthnagar	272152	3	31	1
23724	31-12-2025	Uttar Pradesh	Sultanpur	227806	3	2	0
23725	31-12-2025	Uttar Pradesh	Sultanpur	227808	9	11	0
23726	31-12-2025	Uttar Pradesh	Unnao	209801	25	65	0
23727	31-12-2025	Uttar Pradesh	Unnao	241504	26	32	0

23728 rows × 11 columns

In [ ]:
```
flagged_pincode_UP=df_flagged.groupby(['district_clean','pincode'])[['age_0_
#flagged_pincode.to_excel('flagged_pincode_domain.xlsx')
```

In [ ]:
```
flagged_pincode_UP['total_enrollment']=flagged_pincode_UP['age_0_5']+flagged
flagged_pincode_UP
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Araria	854102	131	86	1	218
1	Araria	854201	31	33	0	64
2	Araria	854202	58	74	0	132
3	Araria	854304	410	412	14	836
4	Araria	854312	837	546	6	1389
...	...	...	...	...	...	...
360	Vaishali	843104	66	37	0	103
361	Vaishali	843105	7	2	0	9
362	Vaishali	844111	153	188	0	341
363	Vaishali	844112	181	195	1	377
364	Vaishali	844120	22	30	0	52

365 rows × 6 columns

In [ ]:
```
idx = flagged_pincode_UP.groupby('pincode')['total_enrollment'].idxmax()
df_filtered_UP = flagged_pincode_UP.loc[idx]
df_filtered_UP
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
223	Nalanda	801304	58	114	0	172
224	Nalanda	801305	39	62	1	102
63	Buxar	802112	117	224	1	342
64	Buxar	802134	125	370	1	496
59	Bhojpur	802160	66	273	2	341
...	...	...	...	...	...	...
281	Purnia	854337	336	131	0	467
165	Kishanganj	855101	1906	357	7	2270
158	Katihar	855105	386	125	6	517
166	Kishanganj	855107	1685	446	1	2132
167	Kishanganj	855115	774	208	4	986

177 rows × 6 columns

In [ ]:
```
df_uttarpradesh['pin_multi_district_flag']=(
    df_uttarpradesh.groupby('pincode')['district_clean']
```

```
    .transform('nunique')>1
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/2731928923.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_uttarpradesh['pin_multi_district_flag']=(
```

In [ ]:
```
pin_district_map_UP = (
    df_uttarpradesh[df_uttarpradesh['pin_multi_district_flag']]
    .groupby('pincode')['district_clean']  # noqa: SC100
    .unique()
    .reset_index()
)
```

In [ ]:
```
## monthly enrolment check
df_uttarpradesh['month'] = df_uttarpradesh['new_date'].astype(str).str[4:6]
df_uttarpradesh
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/3194271396.p
y:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_uttarpradesh['month'] = df_uttarpradesh['new_date'].astype(str).str[4:
6]
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	r
2	09-03-2025	Uttar Pradesh	Kanpur Nagar	208001	29	82	12	2
3	09-03-2025	Uttar Pradesh	Aligarh	202133	62	29	15	2
7	09-03-2025	Uttar Pradesh	Bahraich	271865	26	60	14	2
8	09-03-2025	Uttar Pradesh	Firozabad	283204	28	26	10	2
10	09-03-2025	Uttar Pradesh	Maharajganj	273164	31	70	13	2
...	...	...	...	...	...	...	...	
1005743	31-12-2025	Uttar Pradesh	Varanasi	221002	10	18	0	1
1005744	31-12-2025	Uttar Pradesh	Varanasi	221104	4	11	0	1
1005745	31-12-2025	Uttar Pradesh	Varanasi	221107	1	15	0	1
1005746	31-12-2025	Uttar Pradesh	Varanasi	221207	1	9	0	1
1005747	31-12-2025	Uttar Pradesh	Varanasi	221313	1	0	0	1

110369 rows × 13 columns

In [ ]:
```
df_uttarpradesh_cleaned_UP=df_uttarpradesh.drop(columns=['date','district','
df_uttarpradesh_cleaned_UP
```

Out[ ]:

	pincode	age_0_5	age_5_17	age_18_greater	new_date	state_clean	distri
2	208001	29	82	12	20250309	Uttar Pradesh	Kanp
3	202133	62	29	15	20250309	Uttar Pradesh	
7	271865	26	60	14	20250309	Uttar Pradesh	
8	283204	28	26	10	20250309	Uttar Pradesh	F
10	273164	31	70	13	20250309	Uttar Pradesh	Mal
...	...	...	...	...	...	...	
1005743	221002	10	18	0	20251231	Uttar Pradesh	
1005744	221104	4	11	0	20251231	Uttar Pradesh	
1005745	221107	1	15	0	20251231	Uttar Pradesh	
1005746	221207	1	9	0	20251231	Uttar Pradesh	
1005747	221313	1	0	0	20251231	Uttar Pradesh	

110369 rows × 10 columns

In [ ]:
```
df_uttarpradesh_dist_level_UP = df_uttarpradesh_cleaned_UP.groupby('district
df_uttarpradesh_dist_level_UP['total_enrollment'] = df_uttarpradesh_dist_lev
```

In [ ]:
```
df_uttarpradesh_dist_level_UP.shape
```

Out[ ]:
```
(78, 4)
```

In [ ]:
```
df_uttarpradesh_dist_level_UP
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Agra	16314	12691	905	29910
Aligarh	13830	11776	586	26192
Ambedkar Nagar	3856	4096	50	8002
Amethi	3890	3251	56	7197
Amroha	5605	2476	61	8142
...	...	...	...	...
Sitapur	16237	13869	748	30854
Sonbhadra	2998	2023	13	5034
Sultanpur	5465	5837	132	11434
Unnao	9838	8943	367	19148
Varanasi	9479	13404	451	23334

78 rows × 4 columns

In [ ]:
```
df_uttarpradesh_dist_level_UP.sort_values("total_enrollment",ascending=False
# df_uttarpradesh_dist_level_UP.to_excel('uttarpradesh_district_level_enrolm
```

Out[ ]:

	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Bahraich	14674	22360	2304	39338
1	Sitapur	16237	13869	748	30854
2	Agra	16314	12691	905	29910
3	Bareilly	17187	10017	607	27811
4	Aligarh	13830	11776	586	26192
...	...	...	...	...	...
73	Hamirpur	2040	1398	27	3465
74	Mahoba	1979	1029	12	3020
75	Bhadohi	265	277	4	546
76	Sant Ravidas Nagar Bhadohi	37	47	0	84
77	Jyotiba Phule Nagar	61	23	0	84

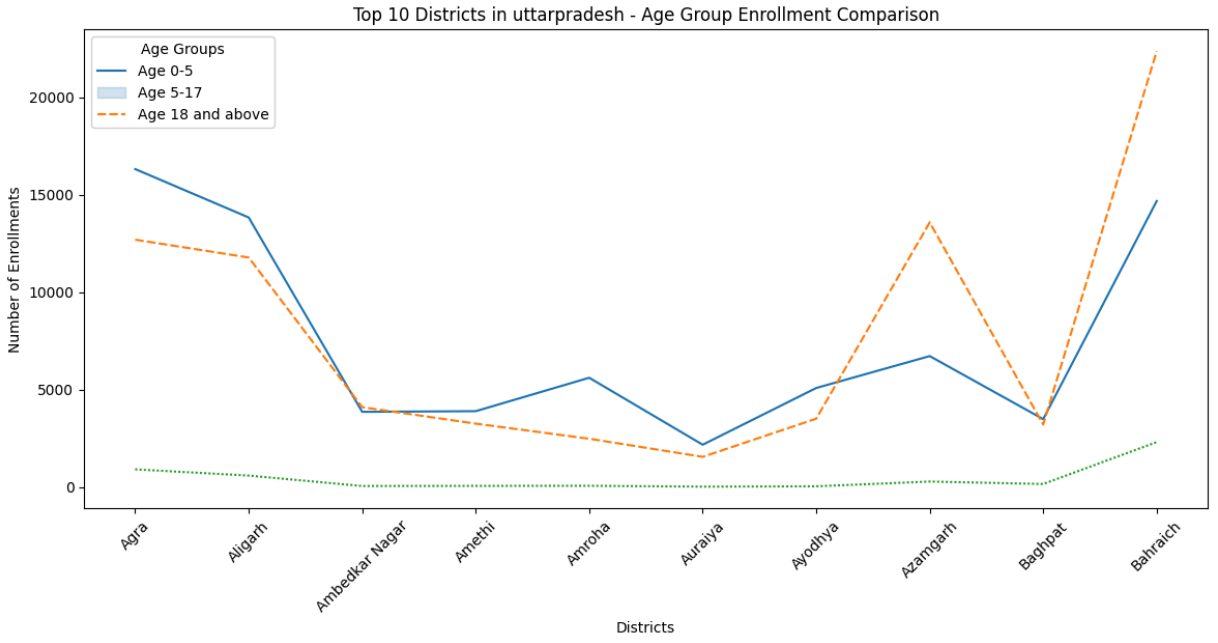
78 rows × 5 columns

In [ ]:
```
df_uttarpradesh_dist_level1 = df_uttarpradesh_dist_level_UP.head(10)
df_uttarpradesh_dist_level1
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Agra	16314	12691	905	29910
Aligarh	13830	11776	586	26192
Ambedkar Nagar	3856	4096	50	8002
Amethi	3890	3251	56	7197
Amroha	5605	2476	61	8142
Auraiya	2169	1547	17	3733
Ayodhya	5079	3505	35	8619
Azamgarh	6717	13577	279	20573
Baghpat	3482	3204	154	6840
Bahraich	14674	22360	2304	39338

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.lineplot(data=df_uttarpradesh_dist_level1[['age_0_5','age_5_17','age_18_
plt.title('Top 10 Districts in uttarpradesh - Age Group Enrollment Comparisc
plt.xlabel('Districts')
plt.ylabel('Number of Enrollments')
plt.legend(title='Age Groups', labels=['Age 0-5', 'Age 5-17', 'Age 18 and ab
plt.xticks(ticks=range(len(df_uttarpradesh_dist_level1.index)), labels=df_ut
plt.show()
```



In [ ]:
```
df_uttarpradesh_pincode_level = df_uttarpradesh_cleaned_UP.groupby('pincode'
df_uttarpradesh_pincode_level['total_enrollment'] = df_uttarpradesh_pincode_
df_uttarpradesh_pincode_level.shape
```

Out[ ]: (1737, 4)

In [ ]:
```
df_uttarpradesh_pincode_level.sort_values("total_enrollment",ascending=False
# df_uttarpradesh_pincode_level.to_excel('uttarpradesh_pincode_level_enrolme

df_uttarpradesh_pincode_level
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
121705	78	119	0	197
201001	2483	1993	73	4549
201002	556	548	41	1145
201003	161	208	6	375
201004	1	2	1	4
...	...	...	...	...
285203	171	209	10	390
285204	140	137	0	277
285205	428	231	31	690
285206	50	39	3	92
285223	38	37	0	75

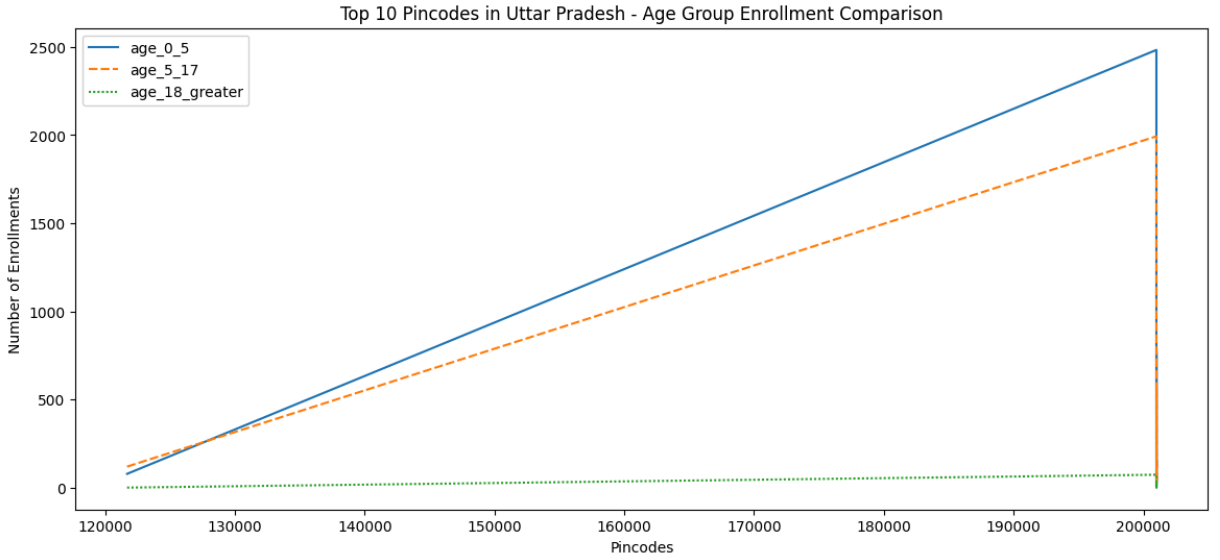
1737 rows × 4 columns

In [ ]:
```
df_uttarpradesh_pincode_level1 = df_uttarpradesh_pincode_level.head(10)
df_uttarpradesh_pincode_level1
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
121705	78	119	0	197
201001	2483	1993	73	4549
201002	556	548	41	1145
201003	161	208	6	375
201004	1	2	1	4
201005	769	562	31	1362
201006	124	37	4	165
201007	156	134	3	293
201008	39	25	0	64
201009	790	555	35	1380

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=df_uttarpradesh_pincode_level1[['age_0_5','age_5_17','age_
plt.title('Top 10 Pincodes in Uttar Pradesh - Age Group Enrollment Comparisc
plt.xlabel('Pincodes')
plt.ylabel('Number of Enrollments')
plt.show()
```



In [ ]:
```
## monthly enrolment trend in bihar
df_uttarpradesh_monthly = df_uttarpradesh_cleaned_UP.groupby('month')[['age_
df_uttarpradesh_monthly['total_enrollment'] = df_uttarpradesh_monthly['age_0
df_uttarpradesh_monthly.shape
```

Out[ ]:
```
(9, 4)
```

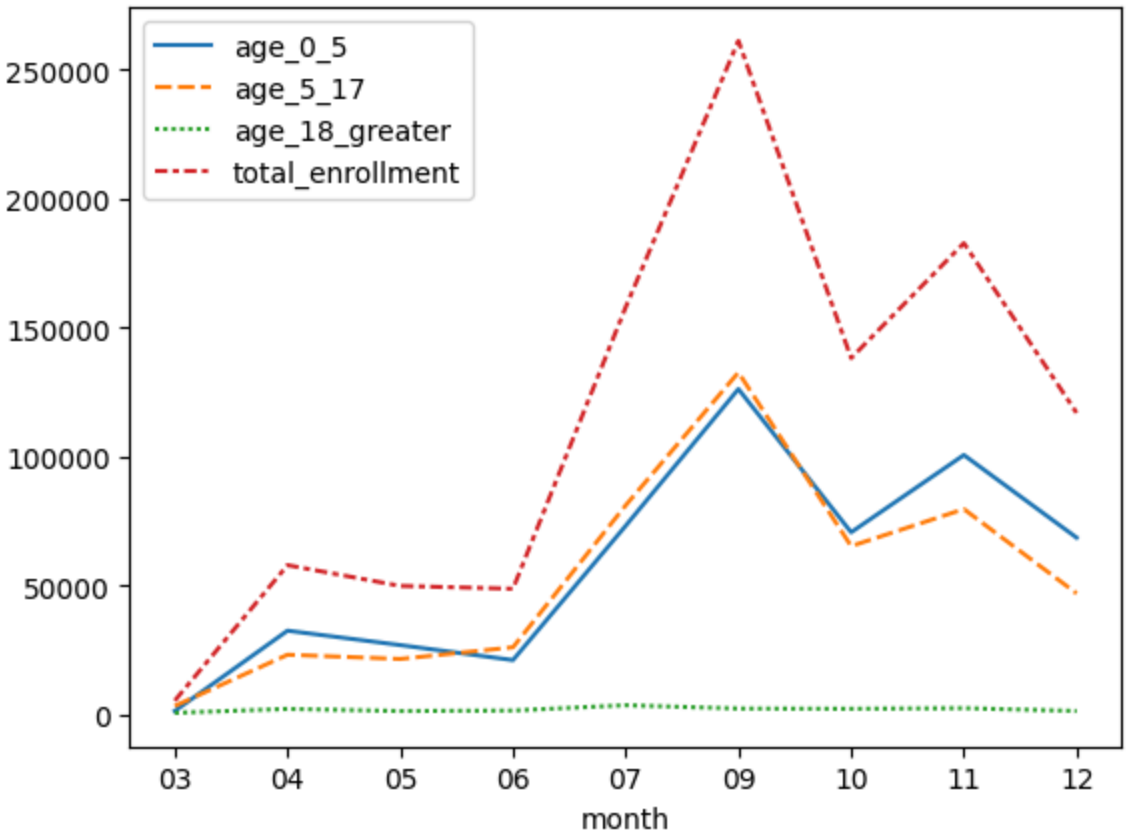
In [ ]:
```
df_uttarpradesh_monthly
df_uttarpradesh_monthly.sort_values("total_enrollment",ascending=False).rese
```

Out[ ]:

	month	age_0_5	age_5_17	age_18_greater	total_enrollment
0	09	126150	132594	2335	261079
1	11	100584	79629	2482	182695
2	07	73166	81078	3622	157866
3	10	70638	65271	2243	138152
4	12	68564	46961	1402	116927
5	04	32513	23189	2201	57903
6	05	26928	21554	1379	49861
7	06	21109	26080	1564	48753
8	03	1393	3326	674	5393

In [ ]:
```
sns.lineplot(data=df_uttarpradesh_monthly)
```

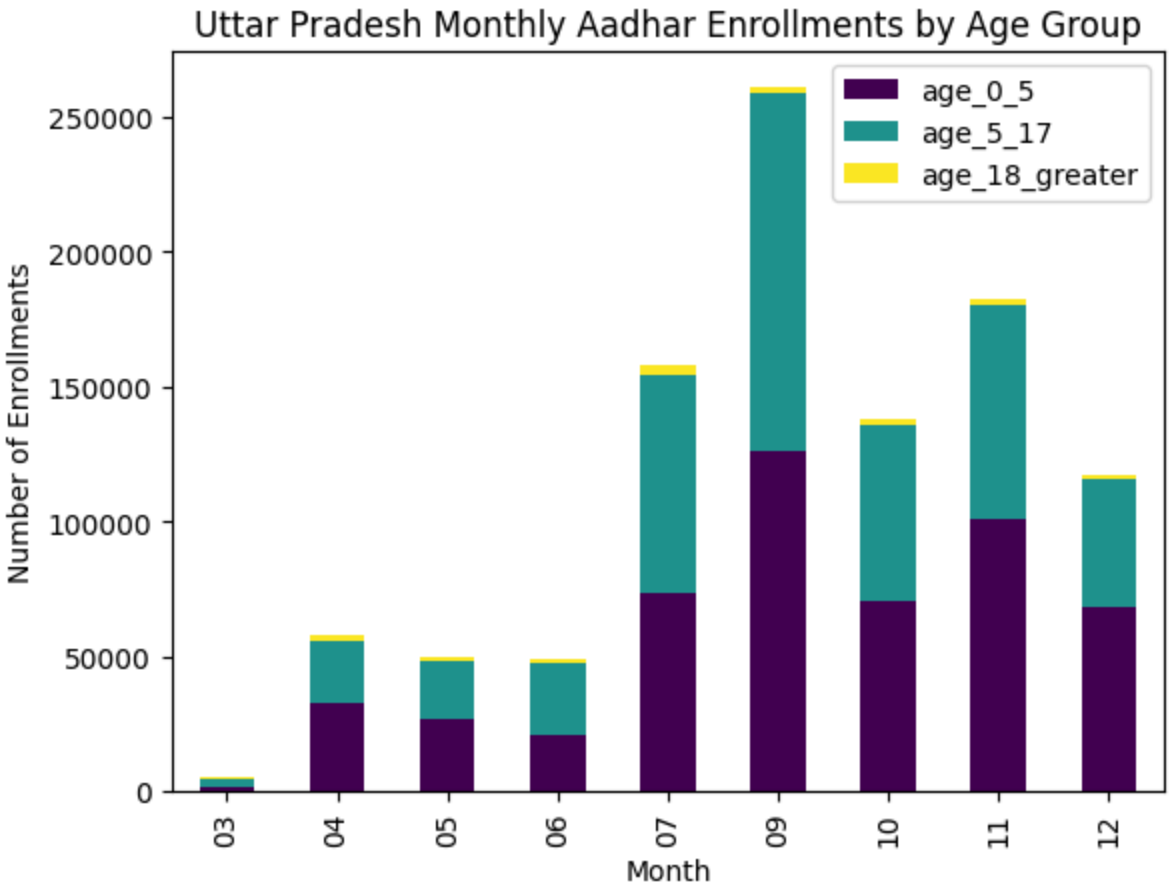
Out[ ]:
```
<Axes: xlabel='month'>
```



In [ ]:
```
## Stacked bar plot for age group comparison
plt.figure(figsize=(10,6))
df_uttarpradesh_monthly[['age_0_5', 'age_5_17', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
```

```
plt.title('Uttar Pradesh Monthly Aadhar Enrollments by Age Group')
plt.xlabel('Month')
plt.ylabel('Number of Enrollments')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



In [ ]:

Madhya Pradesh ke liye

In [ ]:
```
df_madhyapradesh= df[df['state_clean']=='Madhya Pradesh']
df_madhyapradesh
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_
25	09-03-2025	Madhya Pradesh	Bhind	477116	37	18	11	2025(
26	09-03-2025	Madhya Pradesh	Gwalior	475110	42	14	14	2025(
27	09-03-2025	Madhya Pradesh	Katni	483501	53	42	11	2025(
76	20-03-2025	Madhya Pradesh	Gwalior	474001	73	25	18	2025(
125	23-03-2025	Madhya Pradesh	Gwalior	474001	10	21	16	2025(
...	...	...	...	...	...	...	...	
1004141	31-12-2025	Madhya Pradesh	Umaria	484661	66	23	0	2025
1004142	31-12-2025	Madhya Pradesh	Umaria	486661	2	0	0	2025
1004143	31-12-2025	Madhya Pradesh	Vidisha	464001	32	6	0	2025
1004144	31-12-2025	Madhya Pradesh	Vidisha	464220	7	3	0	2025
1004145	31-12-2025	Madhya Pradesh	West Nimar	451111	2	1	0	2025

50225 rows × 10 columns

In [ ]:
```
df_madhyapradesh['district'].unique()
```

Out[ ]:
```
array(['Bhind', 'Gwalior', 'Katni', 'Ashok Nagar', 'Betul', 'Guna',
       'Khargone', 'Chhatarpur', 'Morena', 'Dhar', 'Jabalpur', 'Barwani',
       'Shivpuri', 'Raisen', 'Jhabua', 'Indore', 'Bhopal', 'Datia',
       'Sagar', 'Sheopur', 'Vidisha', 'Burhanpur', 'Panna', 'Khandwa',
       'Satna', 'Alirajpur', 'Dewas', 'Ashoknagar', 'Balaghat',
       'East Nimar', 'Sehore', 'Narsinghpur', 'Hoshangabad', 'Agar Malwa',
       'Anuppur', 'Chhindwara', 'Damoh', 'Dindori', 'Harda', 'Maihar',
       'Mandla', 'Mandsaur', 'Narmadapuram', 'Narsimhapur', 'Neemuch',
       'Niwari', 'Rajgarh', 'Ratlam', 'Rewa', 'Seoni', 'Shahdol',
       'Shajapur', 'Sidhi', 'Singrauli', 'Tikamgarh', 'Ujjain',
       'West Nimar', 'Harda *', 'Mauganj', 'Umaria', 'Pandhurna'],
      dtype=object)
```

In [ ]:
```
df_madhyapradesh['district'].nunique()
```

Out[ ]:
```
61
```

In [ ]:
```
df_madhyapradesh['district_clean'] = (
    df_madhyapradesh['district']
    .str.lower()
    .str.strip()
    .str.replace(r'\*', '', regex=True)
    .str.replace(r'\(.*?\)', '', regex=True)
    .str.replace(r'[^a-z\s]', '', regex=True)
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1754192845.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_madhyapradesh['district_clean'] = (
```

In [ ]:
```
mp_district_standard_map = {
    "ashok nagar": "Ashoknagar",
    "ashoknagar": "Ashoknagar",

    "east nimar": "Khandwa",
    "west nimar": "Khargone",

    "hoshangabad": "Narmadapuram",
    "narmadapuram": "Narmadapuram",

    "narsimhapur": "Narsinghpur",
    "narsinghpur": "Narsinghpur",

    "harda *": "Harda",
    "harda": "Harda",

    "mauganj": "Mauganj",
    "pandhurna": "Pandhurna"
}
```

In [ ]:
```
df_madhyapradesh['district_clean'] = (
    df['district']
    .apply(clean_name)
    .map(mp_district_standard_map)
    .fillna(df_madhyapradesh['district'])

)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/2376381053.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_madhyapradesh['district_clean'] = (
```

In [ ]:
```
## Remaining unmapped
df_madhyapradesh[df_madhyapradesh['district_clean'].isna()]['district'].unic
# count check
df_madhyapradesh['district_clean'].nunique()
```

Out[ ]:
```
57
```

In [ ]:
```
df_madhyapradesh.isnull().sum()
```

Out[ ]:
```
date              0
state             0
district          0
pincode           0
age_0_5           0
age_5_17          0
age_18_greater    0
new_date          0
state_clean       0
district_clean    0
dtype: int64
```

In [ ]:
```
df_madhyapradesh
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_
25	09-03-2025	Madhya Pradesh	Bhind	477116	37	18	11	2025(
26	09-03-2025	Madhya Pradesh	Gwalior	475110	42	14	14	2025(
27	09-03-2025	Madhya Pradesh	Katni	483501	53	42	11	2025(
76	20-03-2025	Madhya Pradesh	Gwalior	474001	73	25	18	2025(
125	23-03-2025	Madhya Pradesh	Gwalior	474001	10	21	16	2025(
...	...	...	...	...	...	...	...	
1004141	31-12-2025	Madhya Pradesh	Umaria	484661	66	23	0	2025
1004142	31-12-2025	Madhya Pradesh	Umaria	486661	2	0	0	2025
1004143	31-12-2025	Madhya Pradesh	Vidisha	464001	32	6	0	2025
1004144	31-12-2025	Madhya Pradesh	Vidisha	464220	7	3	0	2025
1004145	31-12-2025	Madhya Pradesh	West Nimar	451111	2	1	0	2025

50225 rows × 10 columns

In [ ]:
```
## total enrolment by district in uttar pradesh
df_madhyapradesh_dist_level = df_madhyapradesh.groupby('district_clean')[['a
```

In [ ]:
```
df_madhyapradesh['total_enrollment'] = (
    df_madhyapradesh['age_0_5'] +
    df_madhyapradesh['age_5_17'] +
    df_madhyapradesh['age_18_greater']
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/2512688501.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_madhyapradesh['total_enrollment'] = (
```

In [ ]:
```
df_madhyapradesh
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_
25	09-03-2025	Madhya Pradesh	Bhind	477116	37	18	11	2025(
26	09-03-2025	Madhya Pradesh	Gwalior	475110	42	14	14	2025(
27	09-03-2025	Madhya Pradesh	Katni	483501	53	42	11	2025(
76	20-03-2025	Madhya Pradesh	Gwalior	474001	73	25	18	2025(
125	23-03-2025	Madhya Pradesh	Gwalior	474001	10	21	16	2025(
...	...	...	...	...	...	...	...	
1004141	31-12-2025	Madhya Pradesh	Umaria	484661	66	23	0	2025
1004142	31-12-2025	Madhya Pradesh	Umaria	486661	2	0	0	2025
1004143	31-12-2025	Madhya Pradesh	Vidisha	464001	32	6	0	2025
1004144	31-12-2025	Madhya Pradesh	Vidisha	464220	7	3	0	2025
1004145	31-12-2025	Madhya Pradesh	West Nimar	451111	2	1	0	2025

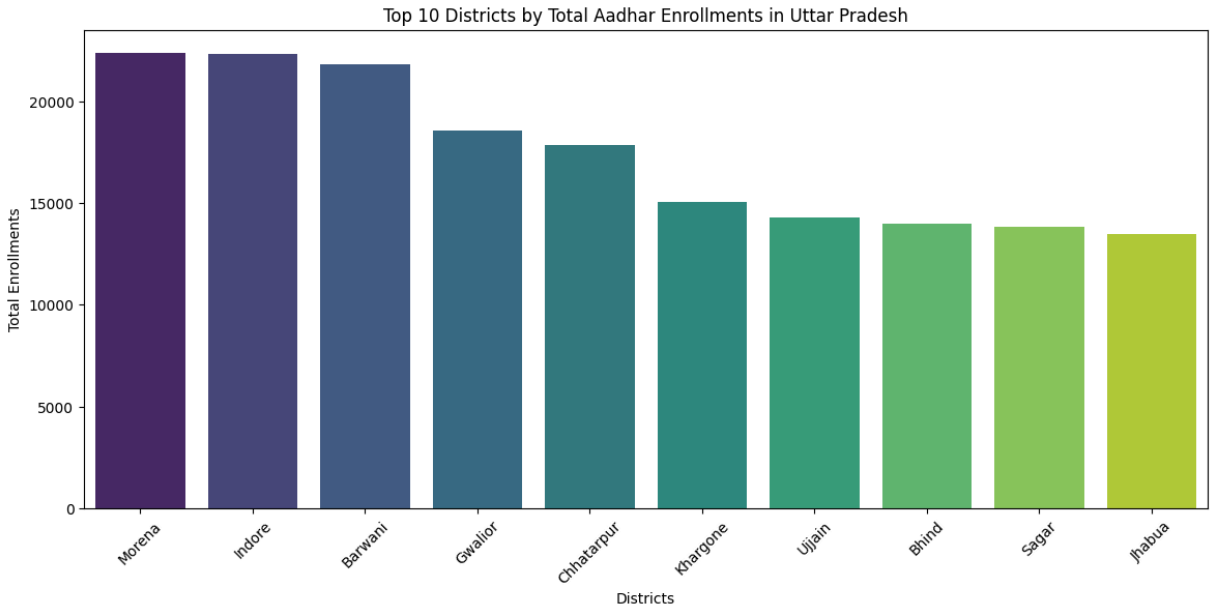
50225 rows × 11 columns

In [ ]:
```
# enrolment top 10 district
df_MP_district_level =df_madhyapradesh["district_clean"].head(10)
```

In [ ]:
```
## total enrolment by district in uttar pradesh
state_summary_MP = df_madhyapradesh.groupby('district_clean')[[
    'age_0_5', 'age_5_17', 'age_18_greater'
]].sum() ## means of all age group by district

state_summary_MP['total_enrollment'] = (
    state_summary_MP['age_0_5'] +
    state_summary_MP['age_5_17'] +
    state_summary_MP['age_18_greater'] ## total enrolment by district
)

top10_districts_MP = state_summary_MP.sort_values(
    by='total_enrollment', ascending=False
).head(10)
```

In [ ]:
```
## bar plot for top 10 states
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.barplot(
    x=top10_districts_MP.index,
    y=top10_districts_MP['total_enrollment'],
    palette='viridis'
)
plt.title('Top 10 Districts by Total Aadhar Enrollments in Uttar Pradesh')
plt.xlabel('Districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.show()
```

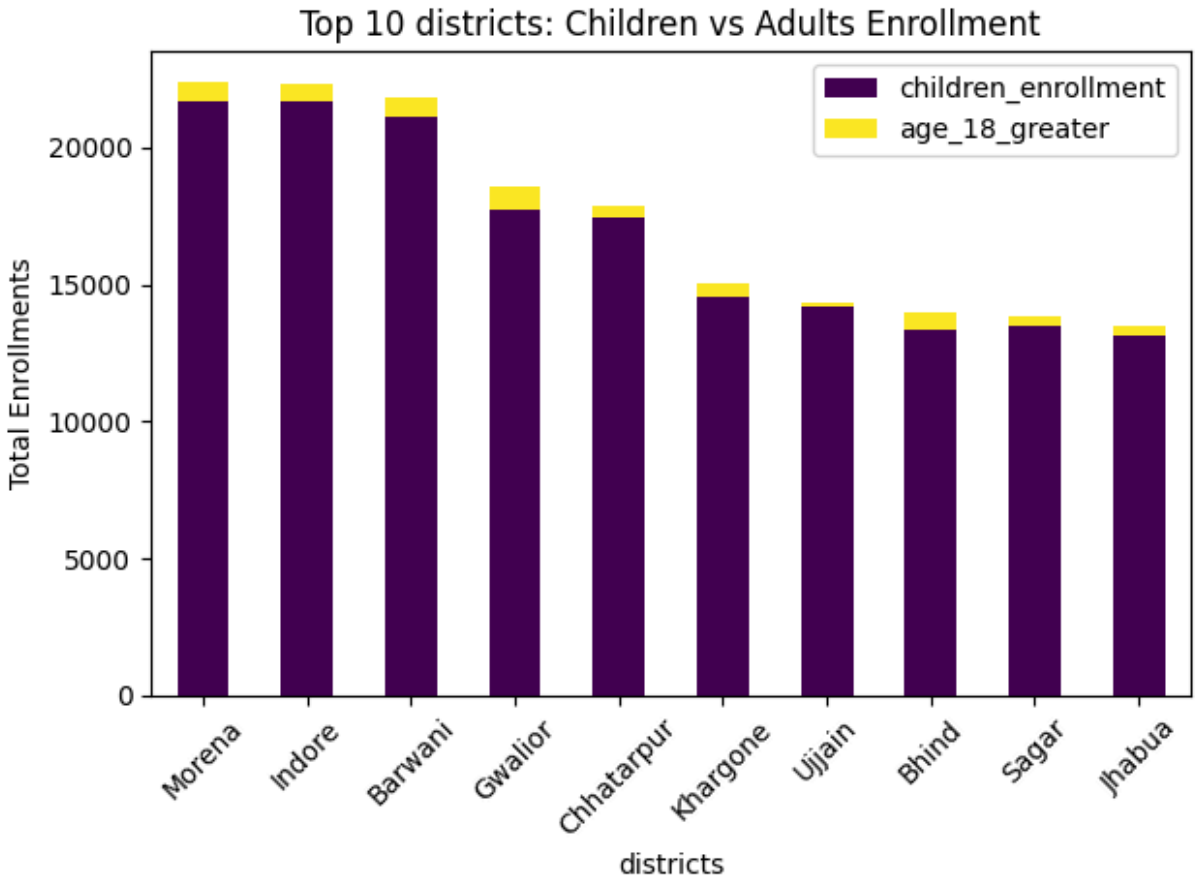
```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/4119891440.p
y:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(
```



In [ ]:
```
top10_districts_MP['children_enrollment'] = (
    top10_districts_MP['age_0_5'] +
    top10_districts_MP['age_5_17']
)
```

In [ ]:
```
# Stacked bar plot for age group comparison
plt.figure(figsize=(14,6))
top10_districts_MP[['children_enrollment', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
plt.title('Top 10 districts: Children vs Adults Enrollment')
plt.xlabel('districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<Figure size 1400x600 with 0 Axes>
```



In [ ]:
```
df_madhyapradesh['new_date'].isnull().sum()
```

Out[ ]:
```
np.int64(0)
```

In [ ]:
```
pincode_check_MP = df_madhyapradesh.groupby('district_clean')['pincode'].nur
pincode_check_MP
```

Out[ ]:

	district_clean	unique_pincodes
0	Agar Malwa	6
1	Alirajpur	10
2	Anuppur	20
3	Ashoknagar	10
4	Balaghat	24
5	Barwani	11
6	Betul	21
7	Bhind	20
8	Bhopal	38
9	Burhanpur	5
10	Chhatarpur	20
11	Chhindwara	27
12	Damoh	14
13	Datia	9
14	Dewas	18
15	Dhar	23
16	Dindori	9
17	East Nimar	20
18	Guna	20
19	Gwalior	21
20	Harda	6
21	Indore	31
22	Jabalpur	27
23	Jhabua	14
24	Katni	17
25	Khandwa	16
26	Khargone	20
27	Maihar	8
28	Mandla	17
29	Mandsaur	20
30	Mauganj	9
31	Morena	12

	district_clean	unique_pincodes
32	Narmadapuram	20
33	Narsinghpur	16
34	Neemuch	15
35	Niwari	9
36	Pandhurna	8
37	Panna	15
38	Raisen	20
39	Rajgarh	15
40	Ratlam	14
41	Rewa	30
42	Sagar	28
43	Satna	26
44	Sehore	18
45	Seoni	17
46	Shahdol	23
47	Shajapur	23
48	Sheopur	5
49	Shivpuri	21
50	Sidhi	12
51	Singrauli	12
52	Tikamgarh	18
53	Ujjain	23
54	Umaria	10
55	Vidisha	13
56	West Nimar	20

In [ ]:
```
pin_district_count_MP = (
    df_madhyapradesh.groupby('pincode')['district_clean']
    .nunique()
    .reset_index(name='district_count')
)
```

In [ ]:
```
pin_district_count_MP
```

Out[ ]:

	pincode	district_count
0	450001	2
1	450051	2
2	450110	2
3	450112	2
4	450114	2
...	...	...
782	488441	1
783	488442	1
784	488443	1
785	488446	1
786	488448	1

787 rows × 2 columns

In [ ]:
```
problem_pins_MP = pin_district_count_MP[
    pin_district_count_MP['district_count'] > 1
]
```

In [ ]:
```
problem_pins_MP
## ek pin code 2 district se belong kr skta hai theek ye govt ki website pr
```

Out[ ]:

	pincode	district_count
0	450001	2
1	450051	2
2	450110	2
3	450112	2
4	450114	2
...	...	...
748	486882	2
749	486884	2
751	486886	4
754	486889	2
755	486890	2

163 rows × 2 columns

In [ ]:
```
df_flagged_MP = df_madhyapradesh.merge(
    problem_pins_MP[['pincode']],
    on='pincode',
    how='inner'
)
```

In [ ]:
```
## ye sab o hai jissme ek district ke 2 pincode hai
## yha se hum pta kr skte hai kiss district me jda use ho rha hai
df_flagged_MP
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_
0	09-03-2025	Madhya Pradesh	Gwalior	475110	42	14	14	2025
1	27-03-2025	Madhya Pradesh	Gwalior	475110	54	26	14	2025
2	01-04-2025	Madhya Pradesh	Ashok Nagar	473335	125	29	22	2025
3	01-04-2025	Madhya Pradesh	Ashok Nagar	473443	141	57	15	2025
4	01-04-2025	Madhya Pradesh	Guna	473101	110	32	11	2025
...	...	...	...	...	...	...	...	
13797	31-12-2025	Madhya Pradesh	Tikamgarh	472246	6	5	0	2025
13798	31-12-2025	Madhya Pradesh	Umaria	484661	66	23	0	2025
13799	31-12-2025	Madhya Pradesh	Umaria	486661	2	0	0	2025
13800	31-12-2025	Madhya Pradesh	Vidisha	464001	32	6	0	2025
13801	31-12-2025	Madhya Pradesh	West Nimar	451111	2	1	0	2025

13802 rows × 11 columns

In [ ]:
```
flagged_pincode_MP=df_flagged_MP.groupby(['district_clean','pincode'])[['age
#flagged_pincode_MP.to_excel('flagged_pincode_domain.xlsx')
```

In [ ]:
```
flagged_pincode_MP['total_enrollment']=flagged_pincode_MP['age_0_5']+flagged
flagged_pincode_MP
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Agar Malwa	465230	257	56	3	316
1	Agar Malwa	465441	618	136	16	770
2	Agar Malwa	465445	371	66	0	437
3	Agar Malwa	465447	389	81	2	472
4	Agar Malwa	465550	205	47	7	259
...	...	...	...	...	...	...
345	West Nimar	451335	18	8	0	26
346	West Nimar	451440	45	6	0	51
347	West Nimar	451441	76	33	0	109
348	West Nimar	451442	27	9	1	37
349	West Nimar	451660	13	1	0	14

350 rows × 6 columns

In [ ]:
```
idx = flagged_pincode_MP.groupby('pincode')['total_enrollment'].idxmax()
df_filtered_MP = flagged_pincode_MP.loc[idx]
df_filtered_MP
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
141	Khandwa	450001	1091	302	17	1410
142	Khandwa	450051	47	17	1	65
143	Khandwa	450110	54	23	0	77
144	Khandwa	450112	107	61	5	173
84	East Nimar	450114	144	103	2	249
...	...	...	...	...	...	...
303	Singrauli	486882	1006	131	5	1142
297	Sidhi	486884	47	6	0	53
305	Singrauli	486886	2250	364	3	2617
306	Singrauli	486889	228	69	7	304
300	Sidhi	486890	69	24	0	93

163 rows × 6 columns

In [ ]:
```
df_madhyapradesh['pin_multi_district_flag']=(
    df_madhyapradesh.groupby('pincode')['district_clean']
    .transform('nunique')>1
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1963035757.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_madhyapradesh['pin_multi_district_flag']=(
```

In [ ]:
```
pin_district_map_MP = (
    df_madhyapradesh[df_madhyapradesh['pin_multi_district_flag']]
    .groupby('pincode')['district_clean']  # noqa: SC100
    .unique()
    .reset_index()
)
```

In [ ]:
```
## monthly enrolment check
df_madhyapradesh['month'] = df_madhyapradesh['new_date'].astype(str).str[4:6
df_madhyapradesh
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1113108687.p
y:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_madhyapradesh['month'] = df_madhyapradesh['new_date'].astype(str).str
[4:6]
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_
25	09-03-2025	Madhya Pradesh	Bhind	477116	37	18	11	2025(
26	09-03-2025	Madhya Pradesh	Gwalior	475110	42	14	14	2025(
27	09-03-2025	Madhya Pradesh	Katni	483501	53	42	11	2025(
76	20-03-2025	Madhya Pradesh	Gwalior	474001	73	25	18	2025(
125	23-03-2025	Madhya Pradesh	Gwalior	474001	10	21	16	2025(
...	...	...	...	...	...	...	...	
1004141	31-12-2025	Madhya Pradesh	Umaria	484661	66	23	0	2025
1004142	31-12-2025	Madhya Pradesh	Umaria	486661	2	0	0	2025
1004143	31-12-2025	Madhya Pradesh	Vidisha	464001	32	6	0	2025
1004144	31-12-2025	Madhya Pradesh	Vidisha	464220	7	3	0	2025
1004145	31-12-2025	Madhya Pradesh	West Nimar	451111	2	1	0	2025

50225 rows × 13 columns

In [ ]:
```
df_madhyapradesh_cleaned_MP=df_madhyapradesh.drop(columns=['date','district'
df_madhyapradesh_cleaned_MP
```

Out[ ]:

	pincode	age_0_5	age_5_17	age_18_greater	new_date	state_clean	distri
25	477116	37	18	11	20250309	Madhya Pradesh	
26	475110	42	14	14	20250309	Madhya Pradesh	
27	483501	53	42	11	20250309	Madhya Pradesh	
76	474001	73	25	18	20250320	Madhya Pradesh	
125	474001	10	21	16	20250323	Madhya Pradesh	
...	...	...	...	...	...	...	
1004141	484661	66	23	0	20251231	Madhya Pradesh	
1004142	486661	2	0	0	20251231	Madhya Pradesh	
1004143	464001	32	6	0	20251231	Madhya Pradesh	
1004144	464220	7	3	0	20251231	Madhya Pradesh	
1004145	451111	2	1	0	20251231	Madhya Pradesh	We

50225 rows × 10 columns

In [ ]:
```
df_madhyapradesh_dist_level_MP = df_madhyapradesh_cleaned_MP.groupby('distri
df_madhyapradesh_dist_level_MP['total_enrollment'] = df_madhyapradesh_dist_l
```

In [ ]:
```
df_madhyapradesh_dist_level_MP.shape
```

Out[ ]:
```
(57, 4)
```

In [ ]:
```
df_madhyapradesh_dist_level_MP
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Agar Malwa	2087	421	29	2537
Alirajpur	3251	3390	369	7010
Anuppur	2864	349	13	3226
Ashoknagar	5320	2608	261	8189
Balaghat	6237	540	28	6805
Barwani	12801	8355	667	21823
Betul	6829	1207	130	8166
Bhind	8483	4845	639	13967
Bhopal	11256	1452	154	12862
Burhanpur	6542	2759	388	9689
Chhatarpur	11504	5965	367	17836
Chhindwara	8308	625	48	8981
Damoh	5081	938	47	6066
Datia	3995	1805	43	5843
Dewas	8393	1958	91	10442
Dhar	8734	3553	425	12712
Dindori	2409	249	1	2659
East Nimar	2338	1500	21	3859
Guna	7496	2249	66	9811
Gwalior	12110	5602	830	18542
Harda	2236	490	18	2744
Indore	16685	5009	609	22303
Jabalpur	10653	1713	101	12467
Jhabua	6586	6570	332	13488
Katni	8413	2410	118	10941
Khandwa	3207	1232	82	4521
Khargone	9870	4663	506	15039
Maihar	378	20	20	418
Mandla	4645	261	5	4911
Mandsaur	7444	808	24	8276
Mauganj	379	58	1	438

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Morena	14064	7647	670	22381
Narmadapuram	5766	912	41	6719
Narsinghpur	4826	1445	92	6363
Neemuch	4183	764	17	4964
Niwari	117	16	9	142
Pandhurna	66	4	28	98
Panna	6807	2404	128	9339
Raisen	6215	1230	122	7567
Rajgarh	6542	1594	35	8171
Ratlam	6427	1478	77	7982
Rewa	9346	1463	29	10838
Sagar	11572	1882	387	13841
Satna	11198	1946	132	13276
Sehore	5155	1308	55	6518
Seoni	5558	513	10	6081
Shahdol	4410	828	59	5297
Shajapur	5937	1070	18	7025
Sheopur	3765	3326	139	7230
Shivpuri	7127	4867	747	12741
Sidhi	5606	671	15	6292
Singrauli	5669	859	22	6550
Tikamgarh	5674	1539	29	7242
Ujjain	12450	1763	95	14308
Umaria	4360	906	34	5300
Vidisha	7998	2206	170	10374
West Nimar	618	136	6	760

In [ ]:
```
df_madhyapradesh_dist_level_MP.sort_values("total_enrollment",ascending=Fals
# df_madhyapradesh_dist_level_MP.to_excel('madhyapradesh_district_level_enro
```

Out[ ]:

	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Morena	14064	7647	670	22381
1	Indore	16685	5009	609	22303
2	Barwani	12801	8355	667	21823
3	Gwalior	12110	5602	830	18542
4	Chhatarpur	11504	5965	367	17836
5	Khargone	9870	4663	506	15039
6	Ujjain	12450	1763	95	14308
7	Bhind	8483	4845	639	13967
8	Sagar	11572	1882	387	13841
9	Jhabua	6586	6570	332	13488
10	Satna	11198	1946	132	13276
11	Bhopal	11256	1452	154	12862
12	Shivpuri	7127	4867	747	12741
13	Dhar	8734	3553	425	12712
14	Jabalpur	10653	1713	101	12467
15	Katni	8413	2410	118	10941
16	Rewa	9346	1463	29	10838
17	Dewas	8393	1958	91	10442
18	Vidisha	7998	2206	170	10374
19	Guna	7496	2249	66	9811
20	Burhanpur	6542	2759	388	9689
21	Panna	6807	2404	128	9339
22	Chhindwara	8308	625	48	8981
23	Mandsaur	7444	808	24	8276
24	Ashoknagar	5320	2608	261	8189
25	Rajgarh	6542	1594	35	8171
26	Betul	6829	1207	130	8166
27	Ratlam	6427	1478	77	7982
28	Raisen	6215	1230	122	7567
29	Tikamgarh	5674	1539	29	7242
30	Sheopur	3765	3326	139	7230
31	Shajapur	5937	1070	18	7025

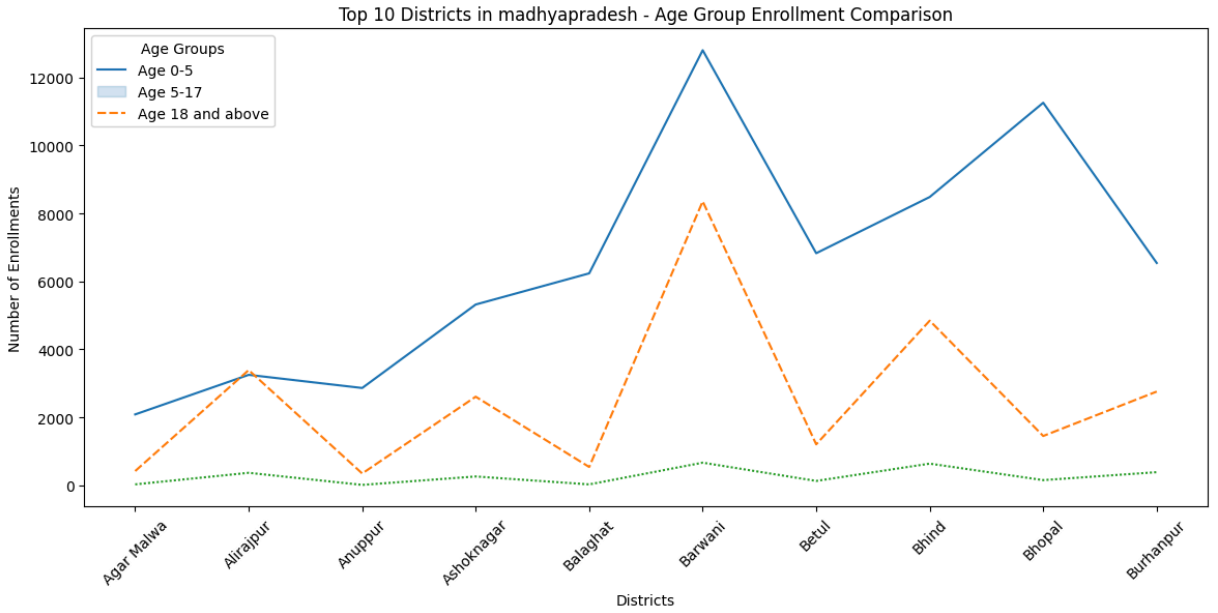
	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
32	Alirajpur	3251	3390	369	7010
33	Balaghat	6237	540	28	6805
34	Narmadapuram	5766	912	41	6719
35	Singrauli	5669	859	22	6550
36	Sehore	5155	1308	55	6518
37	Narsinghpur	4826	1445	92	6363
38	Sidhi	5606	671	15	6292
39	Seoni	5558	513	10	6081
40	Damoh	5081	938	47	6066
41	Datia	3995	1805	43	5843
42	Umaria	4360	906	34	5300
43	Shahdol	4410	828	59	5297
44	Neemuch	4183	764	17	4964
45	Mandla	4645	261	5	4911
46	Khandwa	3207	1232	82	4521
47	East Nimar	2338	1500	21	3859
48	Anuppur	2864	349	13	3226
49	Harda	2236	490	18	2744
50	Dindori	2409	249	1	2659
51	Agar Malwa	2087	421	29	2537
52	West Nimar	618	136	6	760
53	Mauganj	379	58	1	438
54	Maihar	378	20	20	418
55	Niwari	117	16	9	142
56	Pandhurna	66	4	28	98

In [ ]:
```
df_madhyapradesh_dist_level1 = df_madhyapradesh_dist_level_MP.head(10)
df_madhyapradesh_dist_level1
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Agar Malwa	2087	421	29	2537
Alirajpur	3251	3390	369	7010
Anuppur	2864	349	13	3226
Ashoknagar	5320	2608	261	8189
Balaghat	6237	540	28	6805
Barwani	12801	8355	667	21823
Betul	6829	1207	130	8166
Bhind	8483	4845	639	13967
Bhopal	11256	1452	154	12862
Burhanpur	6542	2759	388	9689

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.lineplot(data=df_madhyapradesh_dist_level1[['age_0_5','age_5_17','age_18
plt.title('Top 10 Districts in madhyapradesh - Age Group Enrollment Comparis
plt.xlabel('Districts')
plt.ylabel('Number of Enrollments')
plt.legend(title='Age Groups', labels=['Age 0-5', 'Age 5-17', 'Age 18 and ab
plt.xticks(ticks=range(len(df_madhyapradesh_dist_level1.index)), labels=df_m
plt.show()
```



In [ ]:
```
df_madhyapradesh_pincode_level = df_madhyapradesh_cleaned_MP.groupby('pincod
df_madhyapradesh_pincode_level['total_enrollment'] = df_madhyapradesh_pincod
df_madhyapradesh_pincode_level.shape
```

Out[ ]: (787, 4)

In [ ]:
```
df_madhyapradesh_pincode_level.sort_values("total_enrollment",ascending=Fals
# df_madhyapradesh_pincode_level.to_excel('madhyapradesh_pincode_level_enrol

df_madhyapradesh_pincode_level
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
450001	1535	420	17	1972
450051	76	44	1	121
450110	77	33	0	110
450112	194	113	6	313
450114	224	155	2	381
...	...	...	...	...
488441	763	222	21	1006
488442	903	280	8	1191
488443	5	0	0	5
488446	734	256	3	993
488448	485	130	4	619

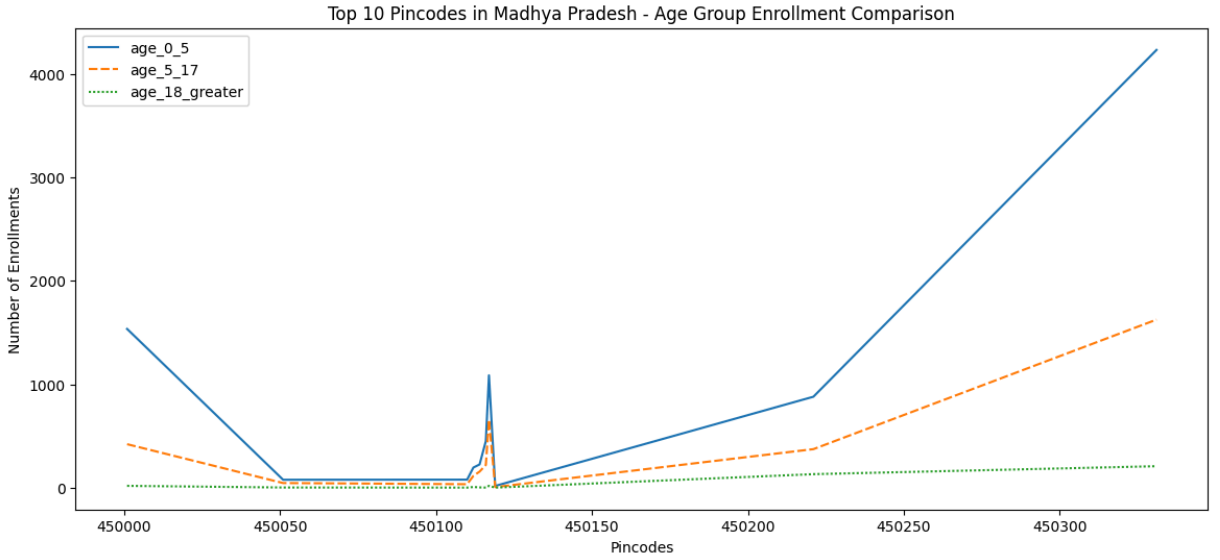
787 rows × 4 columns

In [ ]:
```
df_madhyapradesh_pincode_level1 = df_madhyapradesh_pincode_level.head(10)
df_madhyapradesh_pincode_level1
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
450001	1535	420	17	1972
450051	76	44	1	121
450110	77	33	0	110
450112	194	113	6	313
450114	224	155	2	381
450116	450	219	0	669
450117	1086	656	17	1759
450119	16	5	0	21
450221	879	372	130	1381
450331	4234	1625	207	6066

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=df_madhyapradesh_pincode_level1[['age_0_5','age_5_17','age
plt.title('Top 10 Pincodes in Madhya Pradesh - Age Group Enrollment Comparis
plt.xlabel('Pincodes')
plt.ylabel('Number of Enrollments')
plt.show()
```



In [ ]:
```
## monthly enrolment trend in bihar
df_madhyapradesh_monthly = df_madhyapradesh_cleaned_MP.groupby('month')[['ag
df_madhyapradesh_monthly['total_enrollment'] = df_madhyapradesh_monthly['age
df_madhyapradesh_monthly.shape
```

Out[ ]: (9, 4)

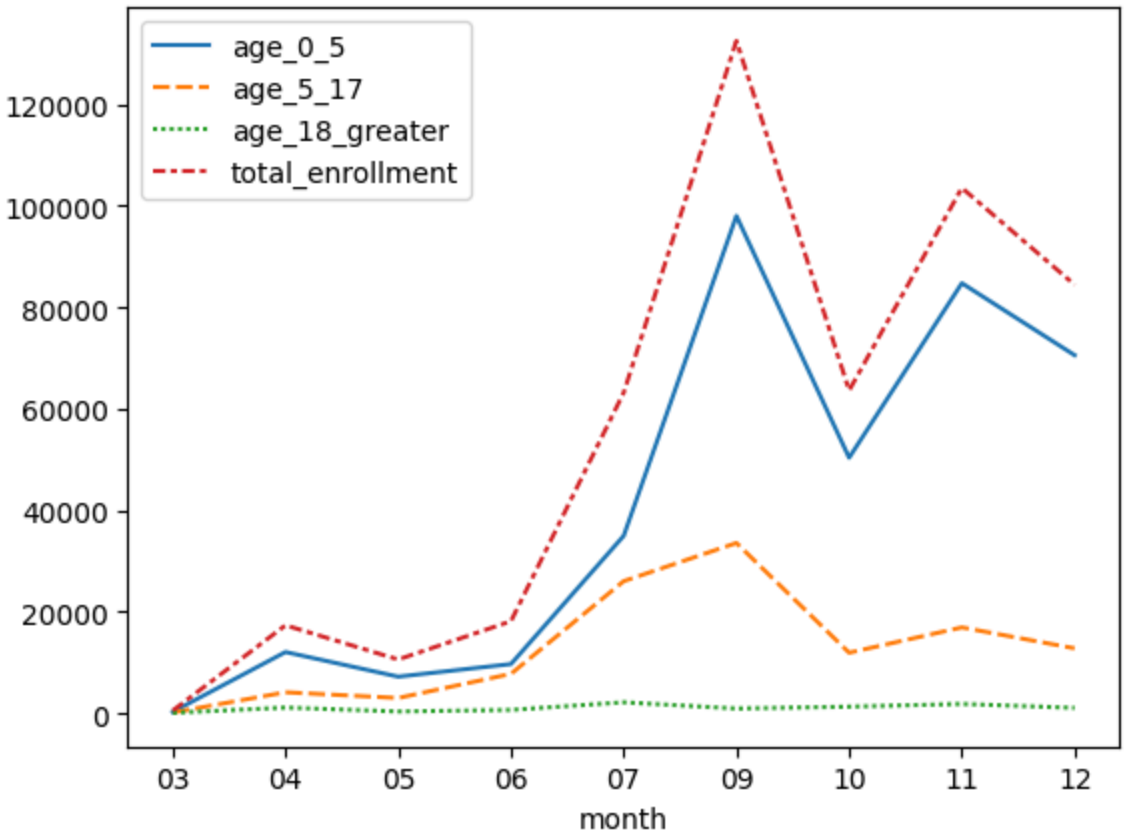
In [ ]:
```
df_madhyapradesh_monthly
df_madhyapradesh_monthly.sort_values("total_enrollment",ascending=False).res
```

Out[ ]:

	month	age_0_5	age_5_17	age_18_greater	total_enrollment
0	09	98016	33562	939	132517
1	11	84762	16930	1847	103539
2	12	70570	12807	1074	84451
3	10	50371	11943	1311	63625
4	07	34979	26040	2169	63188
5	06	9708	7774	672	18154
6	04	12077	4146	1128	17351
7	05	7221	3018	364	10603
8	03	286	161	95	542

In [ ]:
```
sns.lineplot(data=df_madhyapradesh_monthly)
```

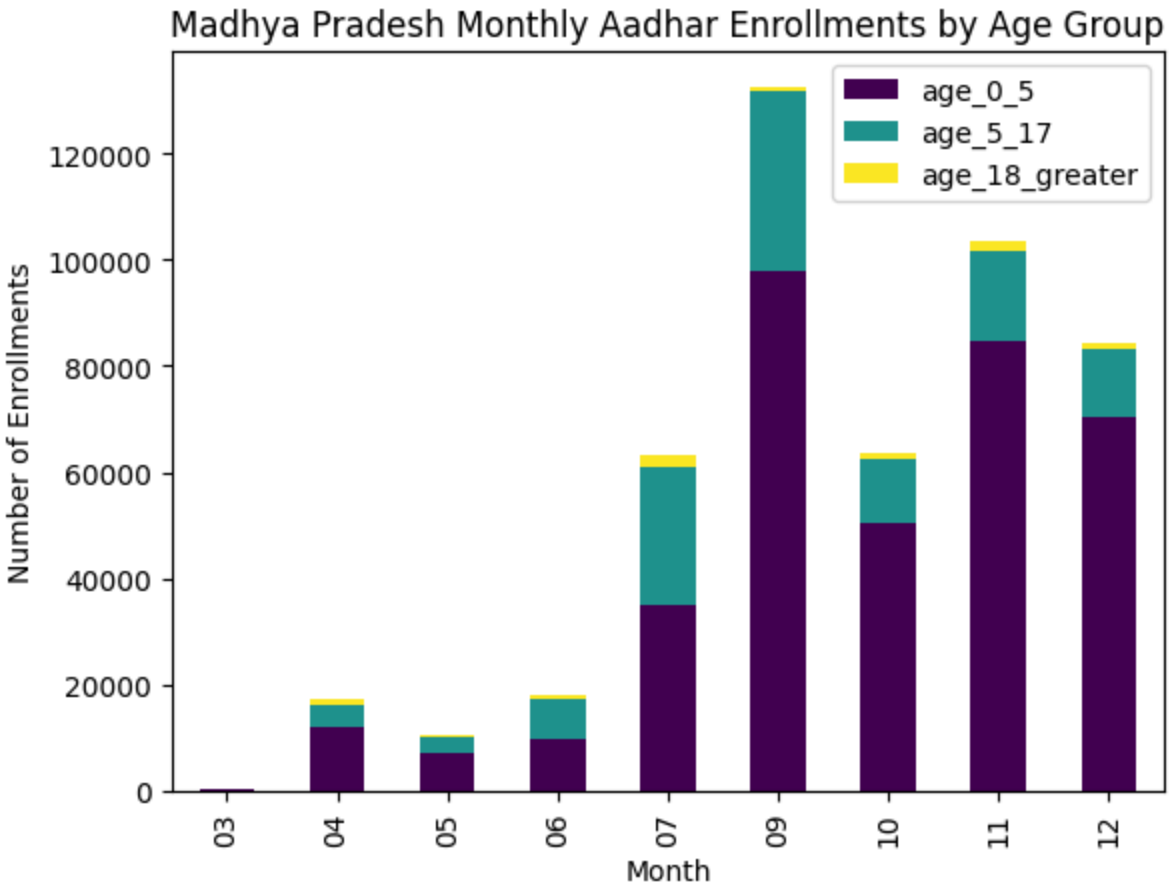
Out[ ]:
```
<Axes: xlabel='month'>
```



In [ ]:
```
## Stacked bar plot for age group comparison
plt.figure(figsize=(10,6))
df_madhyapradesh_monthly[['age_0_5', 'age_5_17', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
```

```
plt.title('Madhya Pradesh Monthly Aadhar Enrollments by Age Group')
plt.xlabel('Month')
plt.ylabel('Number of Enrollments')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



West Bengal

In [ ]:
```
df_West_Bengal= df[df['state_clean']=='West Bengal']
df_West_Bengal
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	ne
30	09-03-2025	West Bengal	Coochbehar	736135	19	12	19	20
32	09-03-2025	West Bengal	Dinajpur Uttar	733129	26	18	27	20
173	01-04-2025	West Bengal	Darjeeling	734010	81	49	17	20
199	01-04-2025	West Bengal	Cooch Behar	736135	243	127	20	20
208	01-04-2025	West Bengal	North 24 Parganas	700159	35	28	14	20
...	...	...	...	...	...	...	...	
1006024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0	2
1006025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0	2
1006026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0	2
1006027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0	2
1006028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0	2

76561 rows × 10 columns

In [ ]:
```
df_West_Bengal['district'].unique()
```

Out[ ]:
```
array(['Coochbehar', 'Dinajpur Uttar', 'Darjeeling', 'Cooch Behar',
       'North 24 Parganas', 'Uttar Dinajpur', 'Jhargram', 'Nadia',
       'Jalpaiguri', 'Alipurduar', 'Malda', 'Kolkata', 'Dakshin Dinajpur',
       'Kalimpong', 'Birbhum', '24 Paraganas North', 'Medinipur West',
       'Purba Bardhaman', 'Hooghly', '24 Paraganas South', 'Howrah',
       'Dinajpur Dakshin', 'Bankura', 'Barddhaman', 'Bardhaman',
       'Darjiling', 'East Midnapore', 'Haora', 'Koch Bihar', 'Maldah',
       'Murshidabad', 'North Twenty Four Parganas', 'Paschim Bardhaman',
       'Paschim Medinipur', 'Purba Medinipur', 'Purulia', 'Puruliya',
       'South 24 Parganas', 'South Dinajpur',
       'South Twenty Four Parganas', 'West Midnapore', 'Hugli',
       'North Dinajpur', 'HOOGHLY', 'NADIA', 'HOWRAH', 'Hawrah', 'MALDA',
       'hooghly', 'Medinipur', 'East Midnapur', 'nadia', 'Hooghiy',
       'KOLKATA', 'West Medinipur', 'Burdwan', 'South 24 parganas',
       'South 24 Pargana'], dtype=object)
```

In [ ]:
```
df_West_Bengal['district'].nunique()
```

Out[ ]:
```
58
```

In [ ]:
```
df_West_Bengal['district_clean'] = (
    df_West_Bengal['district']
    .str.lower()
    .str.strip()
    .str.replace(r'\*', '', regex=True)
    .str.replace(r'\(.*?\)', '', regex=True)
    .str.replace(r'[^a-z\s]', '', regex=True)
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/2946288043.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_West_Bengal['district_clean'] = (
```

In [ ]:
```
wb_district_standard_map = {
    "coochbehar": "Cooch Behar",
    "cooch behar": "Cooch Behar",
    "koch bihar": "Cooch Behar",

    "dinajpur uttar": "Uttar Dinajpur",
    "uttar dinajpur": "Uttar Dinajpur",
    "north dinajpur": "Uttar Dinajpur",

    "dinajpur dakshin": "Dakshin Dinajpur",
    "dakshin dinajpur": "Dakshin Dinajpur",
    "south dinajpur": "Dakshin Dinajpur",

    "darjeeling": "Darjeeling",
    "darjiling": "Darjeeling",

    "kalimpong": "Kalimpong",
```

```
    "north 24 parganas": "North 24 Parganas",
    "24 paraganas north": "North 24 Parganas",
    "north twenty four parganas": "North 24 Parganas",

    "south 24 parganas": "South 24 Parganas",
    "south 24 pargana": "South 24 Parganas",
    "south 24 parganas": "South 24 Parganas",
    "24 paraganas south": "South 24 Parganas",
    "south twenty four parganas": "South 24 Parganas",

    "nadia": "Nadia",

    "jalpaiguri": "Jalpaiguri",
    "alipurduar": "Alipurduar",

    "malda": "Malda",
    "maldah": "Malda",

    "kolkata": "Kolkata",

    "jhargram": "Jhargram",
    "birbhum": "Birbhum",

    "medinipur west": "Paschim Medinipur",
    "west medinipur": "Paschim Medinipur",
    "west midnapore": "Paschim Medinipur",
    "paschim medinipur": "Paschim Medinipur",
    "medinipur": "Paschim Medinipur",

    "purba medinipur": "Purba Medinipur",
    "east midnapore": "Purba Medinipur",
    "east midnapur": "Purba Medinipur",

    "barddhaman": "Bardhaman",
    "bardhaman": "Bardhaman",
    "burdwan": "Bardhaman",

    "purba bardhaman": "Purba Bardhaman",
    "paschim bardhaman": "Paschim Bardhaman",

    "hooghly": "Hooghly",
    "hugli": "Hooghly",
    "hooghiy": "Hooghly",

    "howrah": "Howrah",
    "haora": "Howrah",
    "hawrah": "Howrah",

    "murshidabad": "Murshidabad",
    "bankura": "Bankura",

    "purulia": "Purulia",
    "puruliya": "Purulia"
}
```

In [ ]:
```
df_West_Bengal['district_clean'] = (
    df['district']
    .apply(clean_name)
    .map(wb_district_standard_map)
    .fillna(df_West_Bengal['district'])

)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/3916602445.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_West_Bengal['district_clean'] = (
```

In [ ]:
```
## Remaining unmapped
df_West_Bengal[df_West_Bengal['district_clean'].isna()]['district'].unique()
# count check
df_West_Bengal['district_clean'].nunique()
```

Out[ ]:
```
40
```

In [ ]:
```
df_West_Bengal.isnull().sum()
```

Out[ ]:
```
date              0
state             0
district          0
pincode           0
age_0_5           0
age_5_17          0
age_18_greater    0
new_date          0
state_clean       0
district_clean    0
dtype: int64
```

In [ ]:
```
df_West_Bengal
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	ne
30	09-03-2025	West Bengal	Coochbehar	736135	19	12	19	20
32	09-03-2025	West Bengal	Dinajpur Uttar	733129	26	18	27	20
173	01-04-2025	West Bengal	Darjeeling	734010	81	49	17	20
199	01-04-2025	West Bengal	Cooch Behar	736135	243	127	20	20
208	01-04-2025	West Bengal	North 24 Parganas	700159	35	28	14	20
...	...	...	...	...	...	...	...	
1006024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0	2
1006025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0	2
1006026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0	2
1006027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0	2
1006028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0	2

76561 rows × 10 columns

In [ ]:
```
# total enrolment by district in uttar pradesh
df_West_Bengal_dist_level = df_West_Bengal.groupby('district_clean')[['age_0
```

In [ ]:
```
df_West_Bengal['total_enrollment'] = (
    df_West_Bengal['age_0_5'] +
    df_West_Bengal['age_5_17'] +
    df_West_Bengal['age_18_greater']
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1577518966.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_West_Bengal['total_enrollment'] = (
```

In [ ]:
```
df_West_Bengal
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	ne
30	09-03-2025	West Bengal	Coochbehar	736135	19	12	19	20
32	09-03-2025	West Bengal	Dinajpur Uttar	733129	26	18	27	20
173	01-04-2025	West Bengal	Darjeeling	734010	81	49	17	20
199	01-04-2025	West Bengal	Cooch Behar	736135	243	127	20	20
208	01-04-2025	West Bengal	North 24 Parganas	700159	35	28	14	20
...	...	...	...	...	...	...	...	
1006024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0	2
1006025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0	2
1006026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0	2
1006027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0	2
1006028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0	2

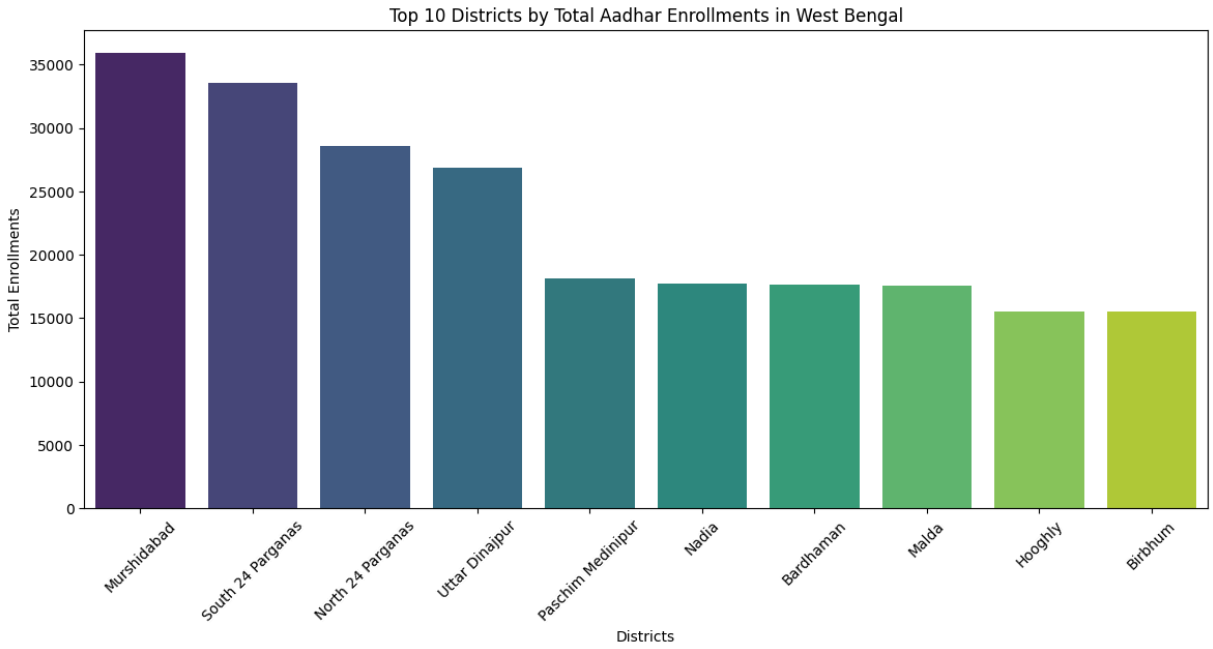
76561 rows × 11 columns

In [ ]:
```
# enrolment top 10 district
df_West_Bengal_level = df_West_Bengal["district_clean"].head(10)
```

In [ ]:
```
## total enrolment by district in uttar pradesh
state_summary_wb = df_West_Bengal.groupby('district_clean')[[
    'age_0_5', 'age_5_17', 'age_18_greater'
]].sum() ## means of all age group by district

state_summary_wb['total_enrollment'] = (
    state_summary_wb['age_0_5'] +
    state_summary_wb['age_5_17'] +
    state_summary_wb['age_18_greater'] ## total enrolment by district
)

top10_districts_wb = state_summary_wb.sort_values(
    by='total_enrollment', ascending=False
).head(10)
```

In [ ]:
```
## bar plot for top 10 states
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.barplot(
    x=top10_districts_wb.index,
    y=top10_districts_wb['total_enrollment'],
    palette='viridis'
)
plt.title('Top 10 Districts by Total Aadhar Enrollments in West Bengal')
plt.xlabel('Districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.show()
```

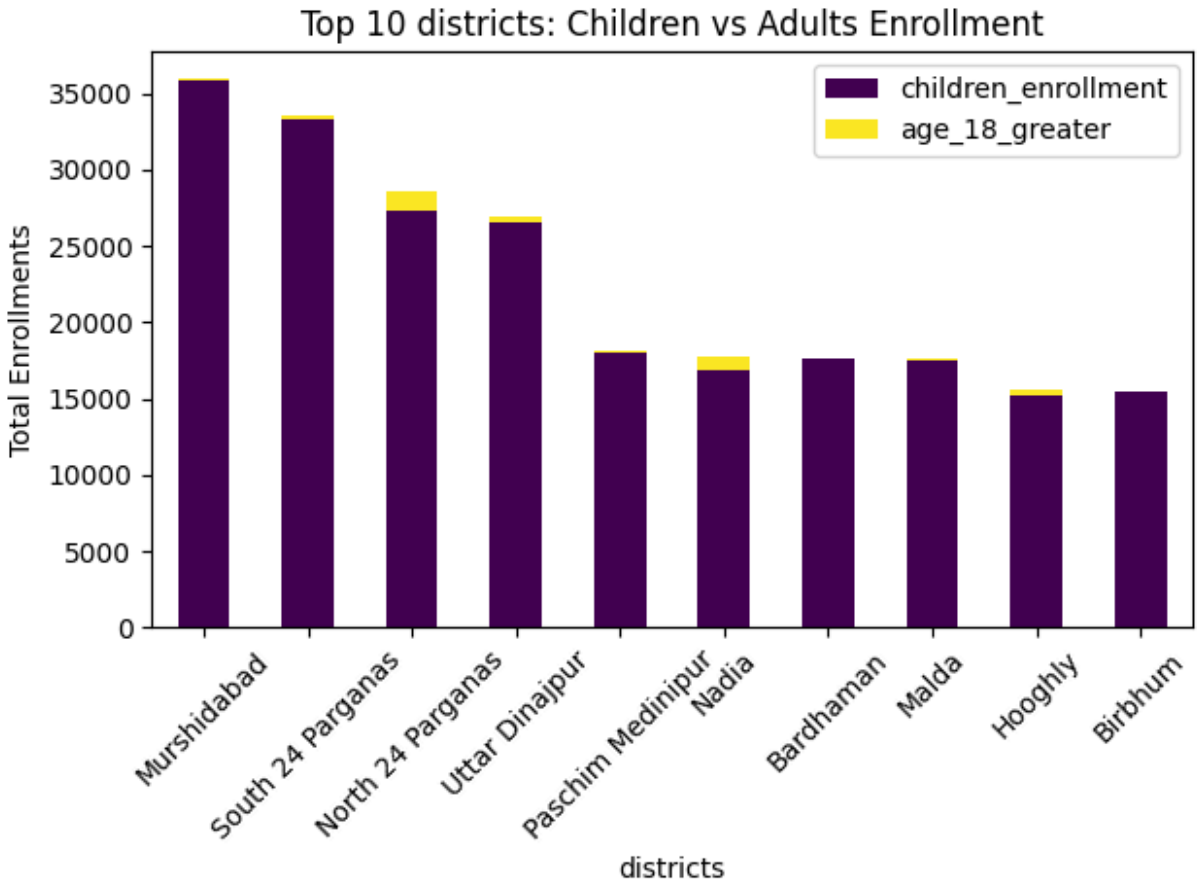
```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1106510956.p
y:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(
```



In [ ]:
```
top10_districts_wb['children_enrollment'] = (
    top10_districts_wb['age_0_5'] +
    top10_districts_wb['age_5_17']
)
```

In [ ]:
```
# Stacked bar plot for age group comparison
plt.figure(figsize=(14,6))
top10_districts_wb[['children_enrollment', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
plt.title('Top 10 districts: Children vs Adults Enrollment')
plt.xlabel('districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<Figure size 1400x600 with 0 Axes>
```



In [ ]:
```
df_West_Bengal['new_date'].isnull().sum()
```

Out[ ]:
```
np.int64(0)
```

In [ ]:
```
pincode_check_wb = df_West_Bengal.groupby('district_clean')['pincode'].nunic
pincode_check_wb
```

Out[ ]:

	district_clean	unique_pincodes
0	24 Paraganas North	23
1	24 Paraganas South	1
2	Alipurduar	25
3	Bankura	74
4	Bardhaman	167
5	Birbhum	49
6	Cooch Behar	45
7	Dakshin Dinajpur	27
8	Darjeeling	64
9	Dinajpur Dakshin	2
10	Dinajpur Uttar	6
11	East Midnapore	77
12	East Midnapur	1
13	Hooghly	106
14	Howrah	59
15	Jalpaiguri	71
16	Jhargram	46
17	Kalimpong	17
18	Koch Bihar	39
19	Kolkata	94
20	Malda	39
21	Medinipur West	2
22	Murshidabad	98
23	Nadia	79
24	North 24 Parganas	150
25	North Dinajpur	15
26	North Twenty Four Parganas	96
27	Paschim Bardhaman	83
28	Paschim Medinipur	95
29	Purba Bardhaman	91
30	Purba Medinipur	103
31	Purulia	43

	district_clean	unique_pincodes
32	South 24 Pargana	1
33	South 24 Parganas	92
34	South 24 parganas	1
35	South Dinajpur	20
36	South Twenty Four Parganas	64
37	Uttar Dinajpur	25
38	West Medinipur	1
39	West Midnapore	61

In [ ]:
```
pin_district_count_wb = (
    df_West_Bengal.groupby('pincode')['district_clean']
    .nunique()
    .reset_index(name='district_count')
)
```

In [ ]:
```
pin_district_count_wb
```

Out[ ]:

	pincode	district_count
0	700001	1
1	700002	1
2	700003	1
3	700004	1
4	700005	1
...	...	...
1331	743702	1
1332	743704	2
1333	743710	2
1334	743711	3
1335	756084	1

1336 rows × 2 columns

In [ ]:
```
problem_pins_wb = pin_district_count_wb[
    pin_district_count_wb['district_count'] > 1
]
```

In [ ]:
```
problem_pins_wb
## ek pin code 2 district se belong kr skta hai theek ye govt ki website pr
```

Out[ ]:

	pincode	district_count
7	700008	3
17	700018	2
23	700024	2
27	700028	2
29	700030	3
...	...	...
1328	743613	2
1330	743701	2
1332	743704	2
1333	743710	2
1334	743711	3

634 rows × 2 columns

In [ ]:
```
df_flagged_wb = df_West_Bengal.merge(
    problem_pins_wb[['pincode']],
    on='pincode',
    how='inner'
)
```

In [ ]:
```
## ye sab o hai jissme ek district ke 2 pincode hai
## yha se hum pta kr skte hai kiss district me jda use ho rha hai
df_flagged_wb
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new
0	09-03-2025	West Bengal	Coochbehar	736135	19	12	19	2025
1	09-03-2025	West Bengal	Dinajpur Uttar	733129	26	18	27	2025
2	01-04-2025	West Bengal	Cooch Behar	736135	243	127	20	2025
3	01-04-2025	West Bengal	North 24 Parganas	700159	35	28	14	2025
4	01-04-2025	West Bengal	Uttar Dinajpur	733134	484	109	27	2025
...	...	...	...	...	...	...	...	
44684	31-12-2025	West Bengal	West Midnapore	721149	2	0	0	202
44685	31-12-2025	West Bengal	West Midnapore	721150	2	2	0	202
44686	31-12-2025	West Bengal	West Midnapore	721305	0	1	0	202
44687	31-12-2025	West Bengal	West Midnapore	721504	1	0	0	202
44688	31-12-2025	West Bengal	West Midnapore	721517	2	1	0	202

44689 rows × 11 columns

In [ ]:
```
flagged_pincode_wb=df_flagged_wb.groupby(['district_clean','pincode'])[['age
#flagged_pincode.to_excel('flagged_pincode_domain.xlsx')
```

In [ ]:
```
flagged_pincode_wb['total_enrollment']=flagged_pincode_wb['age_0_5']+flagged
flagged_pincode_wb
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
0	24 Paraganas North	700049	55	67	12	134
1	24 Paraganas North	700056	34	53	11	98
2	24 Paraganas North	700102	76	67	14	157
3	24 Paraganas North	700110	114	127	25	266
4	24 Paraganas North	700119	116	132	13	261
...	...	...	...	...	...	...
1445	West Midnapore	721513	20	8	1	29
1446	West Midnapore	721515	1	0	0	1
1447	West Midnapore	721516	3	0	0	3
1448	West Midnapore	721517	52	11	0	63
1449	West Midnapore	721641	3	0	0	3

1450 rows × 6 columns

In [ ]:
```
idx = flagged_pincode_wb.groupby('pincode')['total_enrollment'].idxmax()
df_filtered_wb = flagged_pincode_wb.loc[idx]
df_filtered_wb
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
1211	South 24 Parganas	700008	30	29	0	59
573	Kolkata	700018	174	263	9	446
574	Kolkata	700024	303	289	9	601
639	North 24 Parganas	700028	86	38	7	131
640	North 24 Parganas	700030	22	21	2	45
...	...	...	...	...	...	...
1286	South 24 Parganas	743613	37	13	0	50
753	North 24 Parganas	743701	100	15	2	117
754	North 24 Parganas	743704	183	40	12	235
755	North 24 Parganas	743710	192	36	14	242
756	North 24 Parganas	743711	298	53	3	354

634 rows × 6 columns

In [ ]:
```
df_West_Bengal['pin_multi_district_flag']=(
    df_West_Bengal.groupby('pincode')['district_clean']
    .transform('nunique')>1
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/3703270715.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_West_Bengal['pin_multi_district_flag']=(
```

In [ ]:
```
pin_district_map_wb = (
    df_West_Bengal[df_West_Bengal['pin_multi_district_flag']]
    .groupby('pincode')['district_clean']  # noqa: SC100
    .unique()
    .reset_index()
)
```

In [ ]:
```
## monthly enrolment check
df_West_Bengal['month'] = df_West_Bengal['new_date'].astype(str).str[4:6]
df_West_Bengal
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/3600340522.p
y:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_West_Bengal['month'] = df_West_Bengal['new_date'].astype(str).str[4:6]
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	ne
30	09-03-2025	West Bengal	Coochbehar	736135	19	12	19	20
32	09-03-2025	West Bengal	Dinajpur Uttar	733129	26	18	27	20
173	01-04-2025	West Bengal	Darjeeling	734010	81	49	17	20
199	01-04-2025	West Bengal	Cooch Behar	736135	243	127	20	20
208	01-04-2025	West Bengal	North 24 Parganas	700159	35	28	14	20
...	...	...	...	...	...	...	...	
1006024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0	2
1006025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0	2
1006026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0	2
1006027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0	2
1006028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0	2

76561 rows × 13 columns

In [ ]:
```
West_Bengal_cleaned_wb=df_West_Bengal.drop(columns=['date','district','state
West_Bengal_cleaned_wb
```

Out[ ]:

	pincode	age_0_5	age_5_17	age_18_greater	new_date	state_clean	distri
30	736135	19	12	19	20250309	West Bengal	Coo
32	733129	26	18	27	20250309	West Bengal	Dinaj
173	734010	81	49	17	20250401	West Bengal	D
199	736135	243	127	20	20250401	West Bengal	Coo
208	700159	35	28	14	20250401	West Bengal	
...	...	...	...	...	...	...	
1006024	721149	2	0	0	20251231	West Bengal	M
1006025	721150	2	2	0	20251231	West Bengal	M
1006026	721305	0	1	0	20251231	West Bengal	M
1006027	721504	1	0	0	20251231	West Bengal	M
1006028	721517	2	1	0	20251231	West Bengal	M

76561 rows × 10 columns

In [ ]:
```
West_Bengal_dist_level_wb = West_Bengal_cleaned_wb.groupby('district_clean')
West_Bengal_dist_level_wb['total_enrollment'] = West_Bengal_dist_level_wb['a
```

In [ ]:
```
West_Bengal_dist_level_wb.shape
```

Out[ ]:
```
(40, 4)
```

In [ ]:
```
West_Bengal_dist_level_wb
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
24 Paraganas North	3177	2458	512	6147
24 Paraganas South	364	104	22	490
Alipurduar	1521	1714	438	3673
Bankura	9083	2122	150	11355
Bardhaman	14163	3434	17	17614
Birbhum	13361	2041	107	15509
Cooch Behar	9402	5307	429	15138
Dakshin Dinajpur	5358	1069	99	6526
Darjeeling	4502	2768	775	8045
Dinajpur Dakshin	707	256	34	997
Dinajpur Uttar	6478	4859	334	11671
East Midnapore	1796	318	4	2118
East Midnapur	1	0	0	1
Hooghly	11312	3912	342	15566
Howrah	10087	4047	148	14282
Jalpaiguri	5674	4442	538	10654
Jhargram	741	171	50	962
Kalimpong	176	184	153	513
Koch Bihar	2227	1111	10	3348
Kolkata	5028	4611	723	10362
Malda	12979	4523	84	17586
Medinipur West	325	300	20	645
Murshidabad	31442	4383	86	35911
Nadia	13249	3624	843	17716
North 24 Parganas	21517	5741	1348	28606
North Dinajpur	263	79	0	342
North Twenty Four Parganas	1226	399	2	1627
Paschim Bardhaman	865	318	93	1276
Paschim Medinipur	14266	3698	150	18114
Purba Bardhaman	2523	390	128	3041
Purba Medinipur	12954	1310	58	14322

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Purulia	10213	3655	134	14002
South 24 Pargana	1	1	0	2
South 24 Parganas	25039	8197	304	33540
South 24 parganas	2	0	0	2
South Dinajpur	375	71	0	446
South Twenty Four Parganas	2978	1125	4	4107
Uttar Dinajpur	18397	8127	368	26892
West Medinipur	1	1	0	2
West Midnapore	1647	540	3	2190

In [ ]:
```
West_Bengal_dist_level_wb.sort_values("total_enrollment",ascending=False).re
# West_Bengal_dist_level_wb.to_excel('west_bengal_district_level_enrolment.x
```

Out[ ]:

	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Murshidabad	31442	4383	86	35911
1	South 24 Parganas	25039	8197	304	33540
2	North 24 Parganas	21517	5741	1348	28606
3	Uttar Dinajpur	18397	8127	368	26892
4	Paschim Medinipur	14266	3698	150	18114
5	Nadia	13249	3624	843	17716
6	Bardhaman	14163	3434	17	17614
7	Malda	12979	4523	84	17586
8	Hooghly	11312	3912	342	15566
9	Birbhum	13361	2041	107	15509
10	Cooch Behar	9402	5307	429	15138
11	Purba Medinipur	12954	1310	58	14322
12	Howrah	10087	4047	148	14282
13	Purulia	10213	3655	134	14002
14	Dinajpur Uttar	6478	4859	334	11671
15	Bankura	9083	2122	150	11355
16	Jalpaiguri	5674	4442	538	10654
17	Kolkata	5028	4611	723	10362
18	Darjeeling	4502	2768	775	8045
19	Dakshin Dinajpur	5358	1069	99	6526
20	24 Paraganas North	3177	2458	512	6147
21	South Twenty Four Parganas	2978	1125	4	4107
22	Alipurduar	1521	1714	438	3673
23	Koch Bihar	2227	1111	10	3348
24	Purba Bardhaman	2523	390	128	3041
25	West Midnapore	1647	540	3	2190
26	East Midnapore	1796	318	4	2118
27	North Twenty Four Parganas	1226	399	2	1627
28	Paschim Bardhaman	865	318	93	1276
29	Dinajpur Dakshin	707	256	34	997

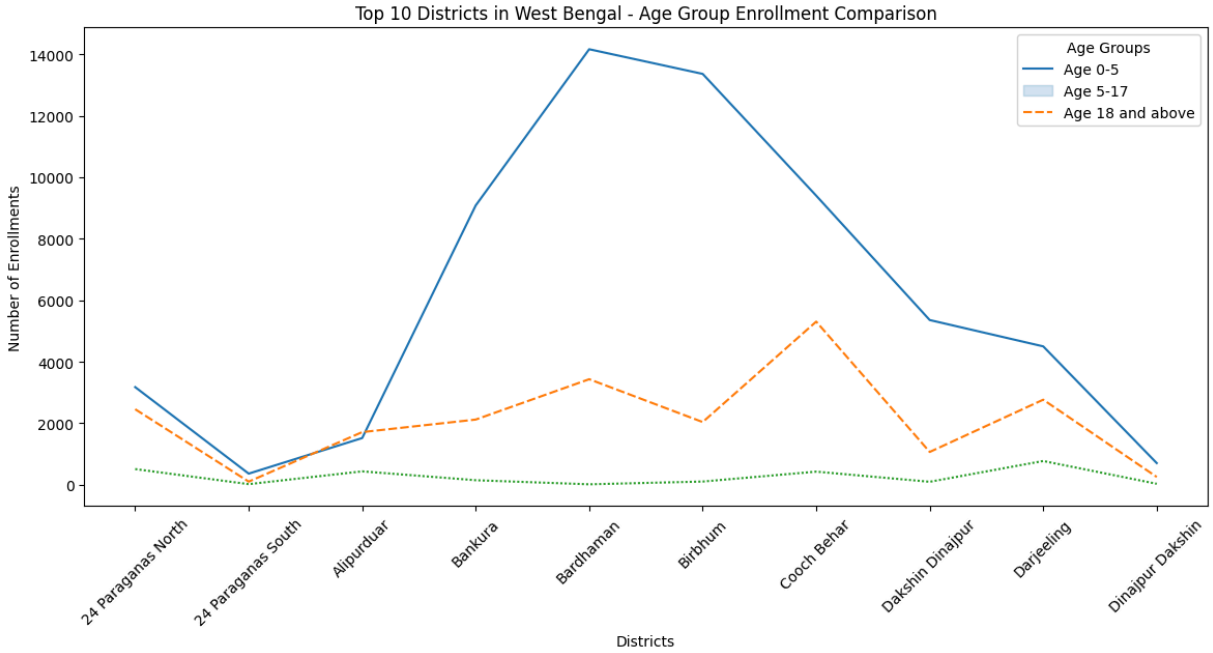
	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
30	Jhargram	741	171	50	962
31	Medinipur West	325	300	20	645
32	Kalimpong	176	184	153	513
33	24 Paraganas South	364	104	22	490
34	South Dinajpur	375	71	0	446
35	North Dinajpur	263	79	0	342
36	South 24 Pargana	1	1	0	2
37	South 24 parganas	2	0	0	2
38	West Medinipur	1	1	0	2
39	East Midnapur	1	0	0	1

In [ ]:
```
West_Bengal_dist_level_wb = West_Bengal_dist_level_wb.head(10)
West_Bengal_dist_level_wb
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
24 Paraganas North	3177	2458	512	6147
24 Paraganas South	364	104	22	490
Alipurduar	1521	1714	438	3673
Bankura	9083	2122	150	11355
Bardhaman	14163	3434	17	17614
Birbhum	13361	2041	107	15509
Cooch Behar	9402	5307	429	15138
Dakshin Dinajpur	5358	1069	99	6526
Darjeeling	4502	2768	775	8045
Dinajpur Dakshin	707	256	34	997

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.lineplot(data=West_Bengal_dist_level_wb[['age_0_5','age_5_17','age_18_gr
plt.title('Top 10 Districts in West Bengal - Age Group Enrollment Comparisor
plt.xlabel('Districts')
plt.ylabel('Number of Enrollments')
plt.legend(title='Age Groups', labels=['Age 0-5', 'Age 5-17', 'Age 18 and ab
plt.xticks(ticks=range(len(West_Bengal_dist_level_wb.index)), labels=West_Be
plt.show()
```



In [ ]:
```
West_Bengal_pincode_level = West_Bengal_cleaned_wb.groupby('pincode')[['age_
West_Bengal_pincode_level['total_enrollment'] = West_Bengal_pincode_level['a
West_Bengal_pincode_level.shape
```

Out[ ]: (1336, 4)

In [ ]:
```
West_Bengal_pincode_level.sort_values("total_enrollment",ascending=False).re
# West_Bengal_pincode_level.to_excel('west_bengal_pincode_level_enrolment.xl

West_Bengal_pincode_level
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
700001	10	18	0	28
700002	126	74	6	206
700003	18	13	3	34
700004	99	81	173	353
700005	29	16	2	47
...	...	...	...	...
743702	74	20	1	95
743704	184	40	12	236
743710	193	36	14	243
743711	342	73	3	418
756084	2	2	1	5

1336 rows × 4 columns

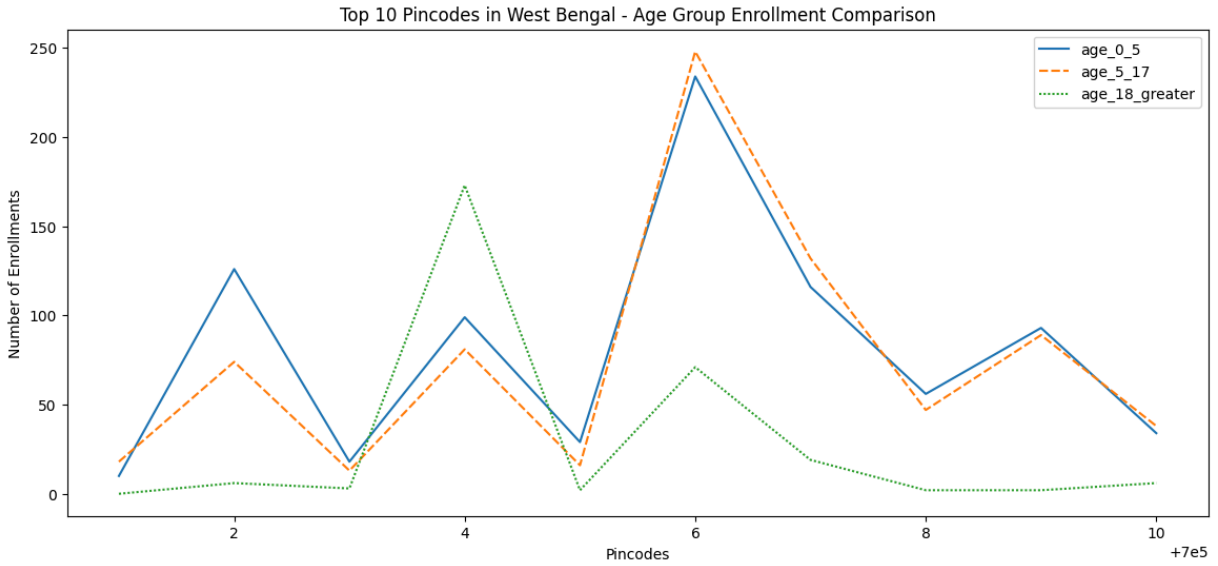
In [ ]:
```
West_Bengal_pincode_level_wb = West_Bengal_pincode_level.head(10)
West_Bengal_pincode_level_wb
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
700001	10	18	0	28
700002	126	74	6	206
700003	18	13	3	34
700004	99	81	173	353
700005	29	16	2	47
700006	234	248	71	553
700007	116	132	19	267
700008	56	47	2	105
700009	93	89	2	184
700010	34	38	6	78

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=West_Bengal_pincode_level_wb[['age_0_5','age_5_17','age_18
plt.title('Top 10 Pincodes in West Bengal – Age Group Enrollment Comparison'
plt.xlabel('Pincodes')
```

```
plt.ylabel('Number of Enrollments')
plt.show()
```



In [ ]:
```
## monthly enrolment trend in west bengal
West_Bengal_monthly =West_Bengal_cleaned_wb.groupby('month')[['age_0_5','age
West_Bengal_monthly['total_enrollment'] = West_Bengal_monthly['age_0_5'] + W
West_Bengal_monthly.shape
```

Out[ ]: (9, 4)

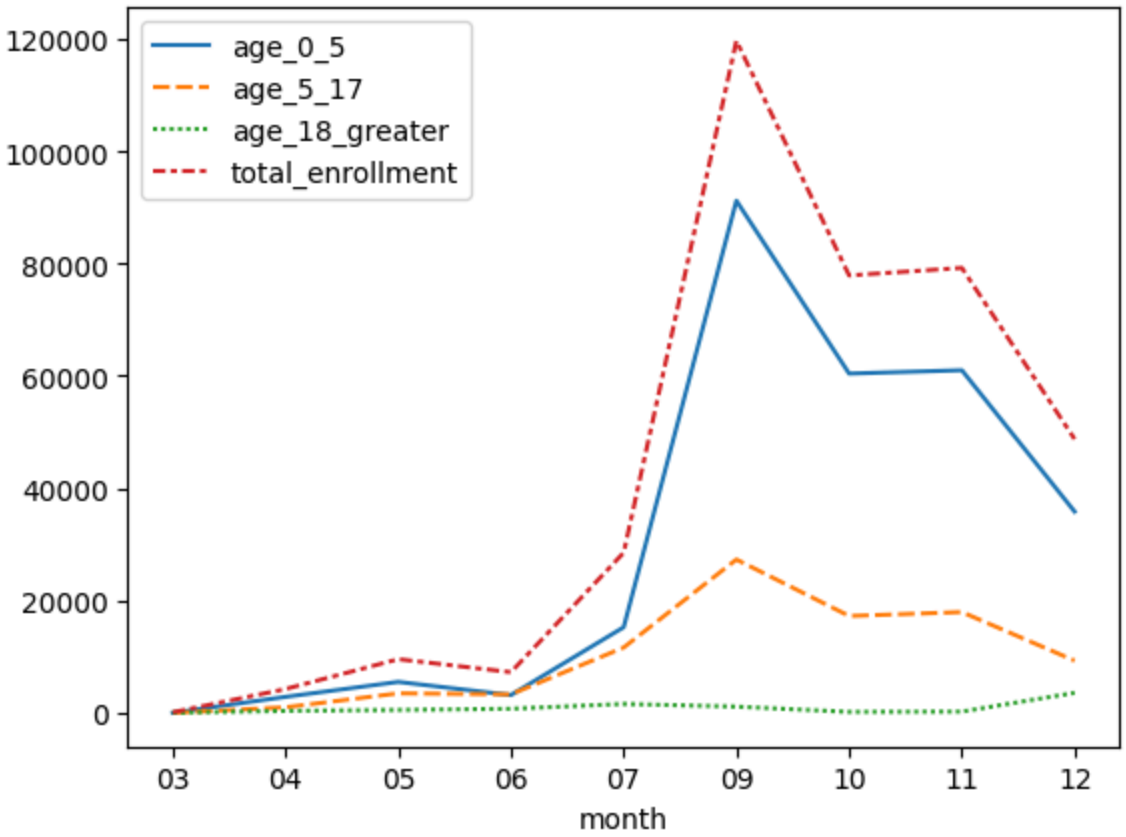
In [ ]:
```
West_Bengal_monthly
West_Bengal_monthly.sort_values("total_enrollment",ascending=False).reset_ir
```

Out[ ]:

	month	age_0_5	age_5_17	age_18_greater	total_enrollment
0	09	91195	27328	1134	119657
1	11	61020	17967	249	79236
2	10	60423	17270	199	77892
3	12	35858	9321	3590	48769
4	07	15274	11657	1613	28544
5	05	5538	3500	544	9582
6	06	3194	3323	754	7271
7	04	2873	1014	381	4268
8	03	45	30	46	121

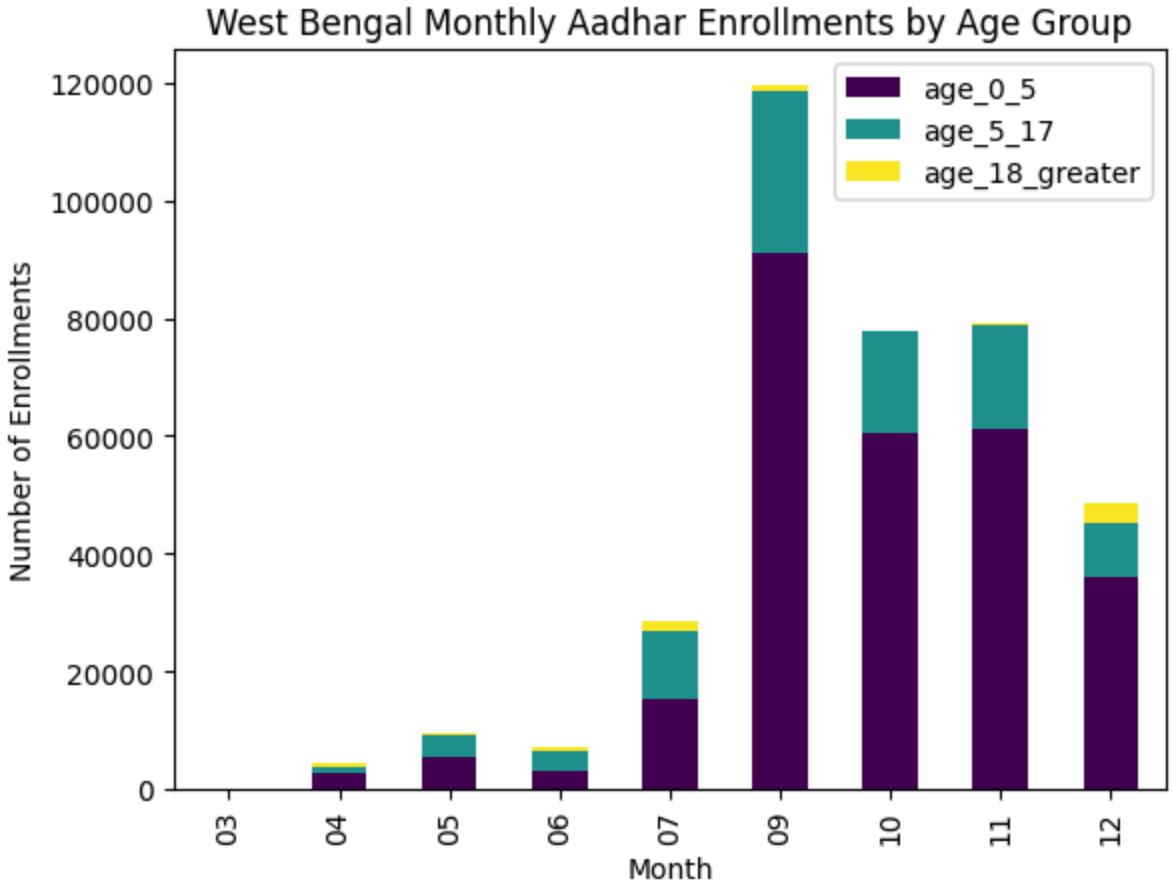
In [ ]:
```
sns.lineplot(data=West_Bengal_monthly)
```

Out[ ]: <Axes: xlabel='month'>



In [ ]:
```
## Stacked bar plot for age group comparison
plt.figure(figsize=(10,6))
West_Bengal_monthly[['age_0_5', 'age_5_17', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
plt.title('West Bengal Monthly Aadhar Enrollments by Age Group')
plt.xlabel('Month')
plt.ylabel('Number of Enrollments')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



Maharashtra ke liye

In [ ]:
```
df_Maharashtra= df[df['state_clean']=='Maharashtra']
df_Maharashtra
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greate
12	09-03-2025	Maharashtra	Aurangabad	431001	42	46	1
43	15-03-2025	Maharashtra	Parbhani	431401	17	14	3
67	20-03-2025	Maharashtra	Thane	421503	19	13	1
78	20-03-2025	Maharashtra	Aurangabad	431001	134	38	1
151	27-03-2025	Maharashtra	Aurangabad	431001	20	19	1
...	...	...	...	...	...	...	.
1004434	31-12-2025	Maharashtra	Yavatmal	445105	2	1	
1004435	31-12-2025	Maharashtra	Yavatmal	445109	6	1	
1004436	31-12-2025	Maharashtra	Yavatmal	445205	4	2	
1004437	31-12-2025	Maharashtra	Yavatmal	445211	1	0	
1004438	31-12-2025	Maharashtra	Yavatmal	445401	2	1	

77191 rows × 10 columns

In [ ]:
```
df_Maharashtra['district'].unique()
```

Out[ ]:
```
array(['Aurangabad', 'Parbhani', 'Thane', 'Nanded', 'Nagpur', 'Jalgaon',
       'Hingoli', 'Ahmadnagar', 'Palghar', 'Satara', 'Raigad',
       'Mumbai Suburban', 'Nandurbar', 'Beed', 'Chandrapur', 'Solapur',
       'Pune', 'Latur', 'Nashik', 'Yavatmal', 'Dhule', 'Washim', 'Sangli',
       'Buldhana', 'Amravati', 'Ahmednagar', 'Mumbai', 'Akola',
       'Osmanabad', 'Ahmed Nagar', 'Bhandara', 'Buldana',
       'Chhatrapati Sambhajinagar', 'Gadchiroli', 'Gondiya', 'Jalna',
       'Kolhapur', 'Mumbai City', 'Raigarh', 'Ratnagiri', 'Sindhudurg',
       'Wardha', 'Chatrapati Sambhaji Nagar', 'Mumbai( Sub Urban )',
       'Nandurbar *', 'Bid', 'Gondiya *', 'Dharashiv', 'Gondia',
       'Washim *', 'Raigarh(MH)', 'Hingoli *', 'Ahilyanagar'],
      dtype=object)
```

In [ ]:
```
df_Maharashtra['district'].nunique()
```

Out[ ]:
```
53
```

In [ ]:
```
df_Maharashtra['district_clean'] = (
    df_Maharashtra['district']
    .str.lower()
    .str.strip()
    .str.replace(r'\*', '', regex=True)
    .str.replace(r'\(.*?\)', '', regex=True)
    .str.replace(r'[^a-z\s]', '', regex=True)
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/4045077458.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_Maharashtra['district_clean'] = (
```

In [ ]:
```
mh_district_standard_map = {
    "aurangabad": "Chhatrapati Sambhajinagar",
    "chhatrapati sambhajinagar": "Chhatrapati Sambhajinagar",
    "chatrapati sambhaji nagar": "Chhatrapati Sambhajinagar",

    "ahilyanagar": "Ahmednagar",
    "ahmadnagar": "Ahmednagar",
    "ahmednagar": "Ahmednagar",
    "ahmed nagar": "Ahmednagar",

    "bid": "Beed",
    "beed": "Beed",

    "buldana": "Buldhana",
    "buldhana": "Buldhana",

    "osmanabad": "Dharashiv",
    "dharashiv": "Dharashiv",

    "gondiya": "Gondia",
    "gondiya *": "Gondia",
```

```
    "gondia": "Gondia",

    "raigarh(mh)": "Raigad",
    "raigad": "Raigad",

    "nandurbar *": "Nandurbar",
    "nandurbar": "Nandurbar",

    "hingoli *": "Hingoli",
    "hingoli": "Hingoli",

    "washim *": "Washim",
    "washim": "Washim",

    "mumbai suburban": "Mumbai Suburban",
    "mumbai( sub urban )": "Mumbai Suburban",

    "mumbai": "Mumbai City",
    "mumbai city": "Mumbai City"
}
```

In [ ]:
```
df_Maharashtra['district_clean'] = (
    df_Maharashtra['district']
    .apply(clean_name)
    .map(mh_district_standard_map)
    .fillna(df_Maharashtra['district'])
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/2727463810.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_Maharashtra['district_clean'] = (
```

In [ ]:
```
## Remaining unmapped
df_Maharashtra[df_Maharashtra['district_clean'].isna()]['district'].unique()
# count check
df_Maharashtra['district_clean'].nunique()
```

Out[ ]: 40

In [ ]:
```
df_Maharashtra.isnull().sum()
```

Out[ ]:
```
date              0
state             0
district          0
pincode           0
age_0_5           0
age_5_17          0
age_18_greater    0
new_date          0
state_clean       0
district_clean    0
dtype: int64
```

In [ ]:
```
df_Maharashtra
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greate
12	09-03-2025	Maharashtra	Aurangabad	431001	42	46	1
43	15-03-2025	Maharashtra	Parbhani	431401	17	14	3
67	20-03-2025	Maharashtra	Thane	421503	19	13	1
78	20-03-2025	Maharashtra	Aurangabad	431001	134	38	1
151	27-03-2025	Maharashtra	Aurangabad	431001	20	19	1
...	...	...	...	...	...	...	.
1004434	31-12-2025	Maharashtra	Yavatmal	445105	2	1	
1004435	31-12-2025	Maharashtra	Yavatmal	445109	6	1	
1004436	31-12-2025	Maharashtra	Yavatmal	445205	4	2	
1004437	31-12-2025	Maharashtra	Yavatmal	445211	1	0	
1004438	31-12-2025	Maharashtra	Yavatmal	445401	2	1	

77191 rows × 10 columns

In [ ]:
```
## total enrolment by district in Maharashtra
df_Maharashtra_dist_level = df_Maharashtra.groupby('district_clean')[['age_0
```

In [ ]:
```
df_Maharashtra['total_enrollment'] = (
    df_Maharashtra['age_0_5'] +
    df_Maharashtra['age_5_17'] +
    df_Maharashtra['age_18_greater']
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/3053212460.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_Maharashtra['total_enrollment'] = (
```

In [ ]:
```
df_Maharashtra
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greate
12	09-03-2025	Maharashtra	Aurangabad	431001	42	46	1
43	15-03-2025	Maharashtra	Parbhani	431401	17	14	3
67	20-03-2025	Maharashtra	Thane	421503	19	13	1
78	20-03-2025	Maharashtra	Aurangabad	431001	134	38	1
151	27-03-2025	Maharashtra	Aurangabad	431001	20	19	1
...	...	...	...	...	...	...	.
1004434	31-12-2025	Maharashtra	Yavatmal	445105	2	1	
1004435	31-12-2025	Maharashtra	Yavatmal	445109	6	1	
1004436	31-12-2025	Maharashtra	Yavatmal	445205	4	2	
1004437	31-12-2025	Maharashtra	Yavatmal	445211	1	0	
1004438	31-12-2025	Maharashtra	Yavatmal	445401	2	1	

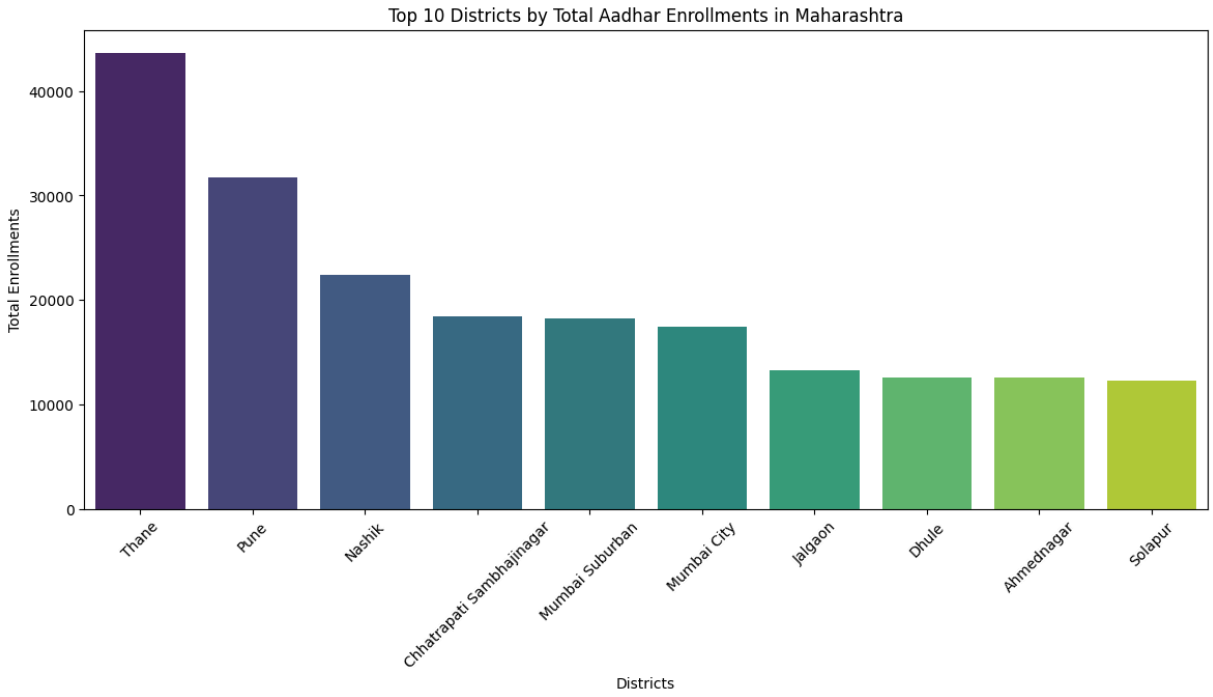
77191 rows × 11 columns

In [ ]:
```
# enrolment top 10 district
df_district_level_mh = df_Maharashtra["district_clean"].head(10)
```

In [ ]:
```
## total enrolment by district in uttar pradesh
state_summary_mh = df_Maharashtra.groupby('district_clean')[[
    'age_0_5', 'age_5_17', 'age_18_greater'
]].sum() ## means of all age group by district

state_summary_mh['total_enrollment'] = (
    state_summary_mh['age_0_5'] +
    state_summary_mh['age_5_17'] +
    state_summary_mh['age_18_greater'] ## total enrolment by district
)

top10_districts_mh = state_summary_mh.sort_values(
    by='total_enrollment', ascending=False
).head(10)
```

In [ ]:
```
## bar plot for top 10 states
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.barplot(
    x=top10_districts_mh.index,
    y=top10_districts_mh['total_enrollment'],
    palette='viridis'
)
plt.title('Top 10 Districts by Total Aadhar Enrollments in Maharashtra')
plt.xlabel('Districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.show()
```

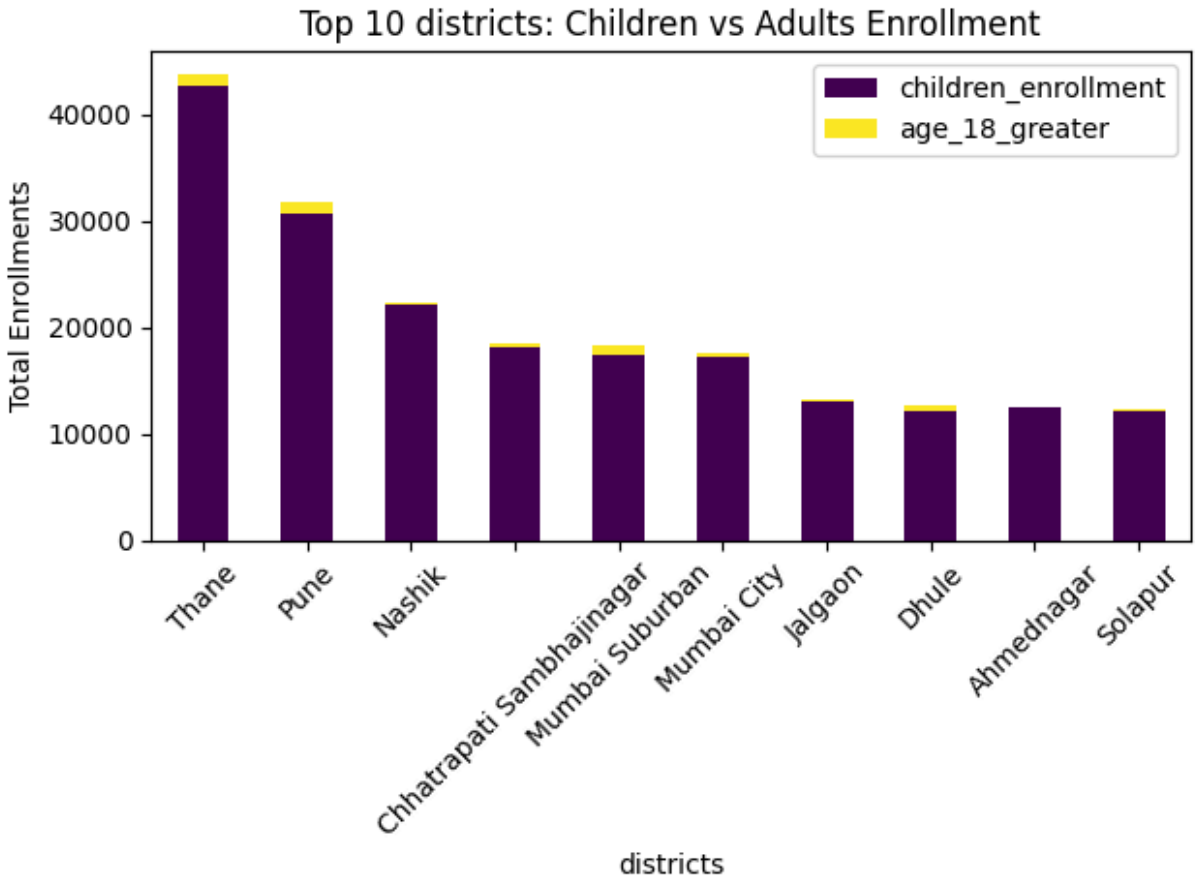
```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1953306520.p
y:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(
```



In [ ]:
```
top10_districts_mh['children_enrollment'] = (
    top10_districts_mh['age_0_5'] +
    top10_districts_mh['age_5_17']
)
```

In [ ]:
```
# Stacked bar plot for age group comparison
plt.figure(figsize=(14,6))
top10_districts_mh[['children_enrollment', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
plt.title('Top 10 districts: Children vs Adults Enrollment')
plt.xlabel('districts')
plt.ylabel('Total Enrollments')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<Figure size 1400x600 with 0 Axes>
```



In [ ]:
```
df_Maharashtra['new_date'].isnull().sum()
```

Out[ ]:
```
np.int64(0)
```

In [ ]:
```
pincode_check_mh = df_Maharashtra.groupby('district_clean')['pincode'].nunic
pincode_check_mh
```

Out[ ]:

	district_clean	unique_pincodes
0	Ahmednagar	90
1	Akola	29
2	Amravati	42
3	Beed	28
4	Bhandara	21
5	Buldhana	32
6	Chandrapur	33
7	Chatrapati Sambhaji Nagar	3
8	Chhatrapati Sambhajinagar	44
9	Dharashiv	28
10	Dhule	29
11	Gadchiroli	15
12	Gondia	14
13	Hingoli	14
14	Jalgaon	62
15	Jalna	26
16	Kolhapur	78
17	Latur	24
18	Mumbai City	109
19	Mumbai Suburban	55
20	Mumbai(Sub Urban)	47
21	Nagpur	67
22	Nanded	44
23	Nandurbar	20
24	Nashik	76
25	Palghar	37
26	Parbhani	19
27	Pune	147
28	Raigad	58
29	Raigarh	62
30	Raigarh(MH)	14
31	Ratnagiri	72

	district_clean	unique_pincodes
32	Sangli	66
33	Satara	79
34	Sindhudurg	52
35	Solapur	71
36	Thane	93
37	Wardha	17
38	Washim	14
39	Yavatmal	33

In [ ]:
```
pin_district_count_mh = (
    df_Maharashtra.groupby('pincode')['district_clean']
    .nunique()
    .reset_index(name='district_count')
)
```

In [ ]:
```
pin_district_count_mh
```

Out[ ]:

	pincode	district_count
0	400001	2
1	400002	1
2	400003	1
3	400004	1
4	400005	1
...	...	...
1575	445307	1
1576	445308	1
1577	445323	1
1578	445401	1
1579	445402	1

1580 rows × 2 columns

In [ ]:
```
problem_pins_mh = pin_district_count_mh[
    pin_district_count_mh['district_count'] > 1
]
```

In [ ]:
```
problem_pins_mh
## ek pin code 2 district se belong kr skta hai theek ye govt ki website pr
```

Out[ ]:

	pincode	district_count
0	400001	2
11	400012	2
22	400024	3
27	400029	2
31	400033	2
...	...	...
1472	444108	2
1474	444110	2
1494	444405	2
1497	444501	2
1499	444503	2

213 rows × 2 columns

In [ ]:
```
df_flagged_mh = df_Maharashtra.merge(
    problem_pins_mh[['pincode']],
    on='pincode',
    how='inner'
)
```

In [ ]:
```
## ye sab o hai jissme ek district ke 2 pincode hai
## yha se hum pta kr skte hai kiss district me jda use ho rha hai
df_flagged_mh
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater
0	09-03-2025	Maharashtra	Aurangabad	431001	42	46	12
1	20-03-2025	Maharashtra	Aurangabad	431001	134	38	18
2	27-03-2025	Maharashtra	Aurangabad	431001	20	19	16
3	01-04-2025	Maharashtra	Hingoli	431512	243	22	33
4	01-04-2025	Maharashtra	Palghar	401209	174	90	16
...	...	...	...	...	...	...	...
16616	31-12-2025	Maharashtra	Thane	401209	4	5	0
16617	31-12-2025	Maharashtra	Thane	401302	0	1	0
16618	31-12-2025	Maharashtra	Thane	401404	1	0	0
16619	31-12-2025	Maharashtra	Thane	421303	2	0	0
16620	31-12-2025	Maharashtra	Wardha	442302	3	0	0

16621 rows × 11 columns

In [ ]:
```
flagged_pincode_mh=df_flagged_mh.groupby(['district_clean','pincode'])[['age
#flagged_pincode_mh.to_excel('flagged_pincode_domain.xlsx')
```

In [ ]:
```
flagged_pincode_mh['total_enrollment']=flagged_pincode_mh['age_0_5']+flagged
flagged_pincode_mh
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Ahmednagar	400037	2	2	0	4
1	Ahmednagar	412210	32	5	0	37
2	Akola	444105	3	0	0	3
3	Akola	444107	152	20	0	172
4	Akola	444108	177	32	1	210
...	...	...	...	...	...	...
492	Washim	444105	352	28	5	385
493	Washim	444107	32	4	0	36
494	Washim	444110	80	5	4	89
495	Washim	444405	3	1	0	4
496	Washim	444503	273	21	4	298

497 rows × 6 columns

In [ ]:
```
idx = flagged_pincode_mh.groupby('pincode')['total_enrollment'].idxmax()
df_filtered_mh = flagged_pincode_mh.loc[idx]
df_filtered_mh
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
77	Mumbai City	400001	44	13	6	63
78	Mumbai City	400012	251	21	9	281
154	Mumbai Suburban	400024	51	12	2	65
80	Mumbai City	400029	39	9	0	48
81	Mumbai City	400033	129	28	2	159
...	...	...	...	...	...	...
4	Akola	444108	177	32	1	210
494	Washim	444110	80	5	4	89
6	Akola	444405	56	2	0	58
7	Akola	444501	252	23	3	278
496	Washim	444503	273	21	4	298

213 rows × 6 columns

In [ ]:
```
df_Maharashtra['pin_multi_district_flag']=(
    df_Maharashtra.groupby('pincode')['district_clean']
```

```
    .transform('nunique')>1
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/1346061413.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_Maharashtra['pin_multi_district_flag']=(
```

In [ ]:
```
pin_district_map_mh = (
    df_Maharashtra[df_Maharashtra['pin_multi_district_flag']]
    .groupby('pincode')['district_clean']  # noqa: SC100
    .unique()
    .reset_index()
)
```

In [ ]:
```
## monthly enrolment check
df_Maharashtra['month'] = df_Maharashtra['new_date'].astype(str).str[4:6]
df_Maharashtra
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_2081/4211111142.p
y:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_Maharashtra['month'] = df_Maharashtra['new_date'].astype(str).str[4:6]
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greate
12	09-03-2025	Maharashtra	Aurangabad	431001	42	46	1
43	15-03-2025	Maharashtra	Parbhani	431401	17	14	3
67	20-03-2025	Maharashtra	Thane	421503	19	13	1
78	20-03-2025	Maharashtra	Aurangabad	431001	134	38	1
151	27-03-2025	Maharashtra	Aurangabad	431001	20	19	1
...	...	...	...	...	...	...	.
1004434	31-12-2025	Maharashtra	Yavatmal	445105	2	1	
1004435	31-12-2025	Maharashtra	Yavatmal	445109	6	1	
1004436	31-12-2025	Maharashtra	Yavatmal	445205	4	2	
1004437	31-12-2025	Maharashtra	Yavatmal	445211	1	0	
1004438	31-12-2025	Maharashtra	Yavatmal	445401	2	1	

77191 rows × 13 columns

In [ ]:
```
df_Maharashtra_cleaned_mh=df_Maharashtra.drop(columns=['date','district','st
df_Maharashtra_cleaned_mh
```

Out[ ]:

	pincode	age_0_5	age_5_17	age_18_greater	new_date	state_clean	distri
12	431001	42	46	12	20250309	Maharashtra	Ch Samb
43	431401	17	14	37	20250315	Maharashtra	
67	421503	19	13	15	20250320	Maharashtra	
78	431001	134	38	18	20250320	Maharashtra	Ch Samb
151	431001	20	19	16	20250327	Maharashtra	Ch Samb
...	...	...	...	...	...	...	
1004434	445105	2	1	0	20251231	Maharashtra	
1004435	445109	6	1	0	20251231	Maharashtra	
1004436	445205	4	2	0	20251231	Maharashtra	
1004437	445211	1	0	0	20251231	Maharashtra	
1004438	445401	2	1	0	20251231	Maharashtra	

77191 rows × 10 columns

In [ ]:
```
df_Maharashtra_dist_level_mh = df_Maharashtra_cleaned_mh.groupby('district_c
df_Maharashtra_dist_level_mh['total_enrollment'] = df_Maharashtra_dist_level
```

In [ ]:
```
df_Maharashtra_dist_level_mh.shape
```

Out[ ]:
```
(40, 4)
```

In [ ]:
```
df_Maharashtra_dist_level_mh
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Ahmednagar	9840	2560	157	12557
Akola	3966	693	49	4708
Amravati	5783	1025	107	6915
Beed	7893	2590	270	10753
Bhandara	1842	55	6	1903
Buldhana	6812	1205	142	8159
Chandrapur	4014	375	86	4475
Chatrapati Sambhaji Nagar	114	81	16	211
Chhatrapati Sambhajinagar	12785	5239	379	18403
Dharashiv	3392	741	37	4170
Dhule	7244	4842	519	12605
Gadchiroli	2084	268	22	2374
Gondia	2858	112	6	2976
Hingoli	6360	1457	373	8190
Jalgaon	10454	2579	227	13260
Jalna	4627	944	19	5590
Kolhapur	6423	985	72	7480
Latur	5828	1631	148	7607
Mumbai City	12908	4330	243	17481
Mumbai Suburban	12480	4946	790	18216
Mumbai(Sub Urban)	340	79	0	419
Nagpur	10093	1502	233	11828
Nanded	9164	2447	336	11947
Nandurbar	7755	2663	238	10656
Nashik	16262	5802	304	22368
Palghar	7343	3191	165	10699
Parbhani	5062	1011	199	6272
Pune	24088	6536	1139	31763
Raigad	2585	743	105	3433
Raigarh	4414	815	1	5230
Raigarh(MH)	72	16	0	88

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Ratnagiri	2352	219	32	2603
Sangli	5697	1802	133	7632
Satara	5148	717	156	6021
Sindhudurg	1121	59	22	1202
Solapur	9537	2495	260	12292
Thane	29092	13629	967	43688
Wardha	1859	89	5	1953
Washim	3069	304	45	3418
Yavatmal	6054	1339	201	7594

In [ ]:
```
df_Maharashtra_dist_level_mh.sort_values("total_enrollment",ascending=False)
# df_Maharashtra_dist_level_mh.to_excel('maharashtra_district_level_enrolmer
```

Out[ ]:

	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
0	Thane	29092	13629	967	43688
1	Pune	24088	6536	1139	31763
2	Nashik	16262	5802	304	22368
3	Chhatrapati Sambhajinagar	12785	5239	379	18403
4	Mumbai Suburban	12480	4946	790	18216
5	Mumbai City	12908	4330	243	17481
6	Jalgaon	10454	2579	227	13260
7	Dhule	7244	4842	519	12605
8	Ahmednagar	9840	2560	157	12557
9	Solapur	9537	2495	260	12292
10	Nanded	9164	2447	336	11947
11	Nagpur	10093	1502	233	11828
12	Beed	7893	2590	270	10753
13	Palghar	7343	3191	165	10699
14	Nandurbar	7755	2663	238	10656
15	Hingoli	6360	1457	373	8190
16	Buldhana	6812	1205	142	8159
17	Sangli	5697	1802	133	7632
18	Latur	5828	1631	148	7607
19	Yavatmal	6054	1339	201	7594
20	Kolhapur	6423	985	72	7480
21	Amravati	5783	1025	107	6915
22	Parbhani	5062	1011	199	6272
23	Satara	5148	717	156	6021
24	Jalna	4627	944	19	5590
25	Raigarh	4414	815	1	5230
26	Akola	3966	693	49	4708
27	Chandrapur	4014	375	86	4475
28	Dharashiv	3392	741	37	4170
29	Raigad	2585	743	105	3433
30	Washim	3069	304	45	3418

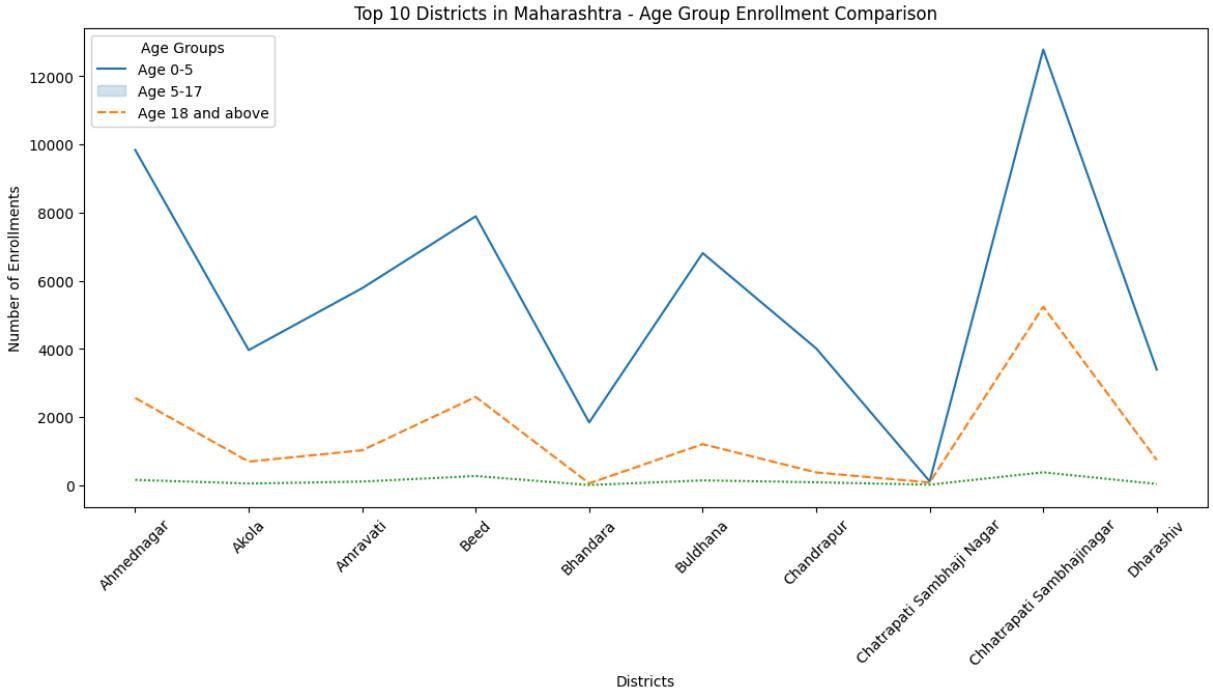
	district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
31	Gondia	2858	112	6	2976
32	Ratnagiri	2352	219	32	2603
33	Gadchiroli	2084	268	22	2374
34	Wardha	1859	89	5	1953
35	Bhandara	1842	55	6	1903
36	Sindhudurg	1121	59	22	1202
37	Mumbai(Sub Urban)	340	79	0	419
38	Chatrapati Sambhaji Nagar	114	81	16	211
39	Raigarh(MH)	72	16	0	88

In [ ]:
```
df_Maharashtra_dist_level_mh = df_Maharashtra_dist_level_mh.head(10)
df_Maharashtra_dist_level_mh
```

Out[ ]:

district_clean	age_0_5	age_5_17	age_18_greater	total_enrollment
Ahmednagar	9840	2560	157	12557
Akola	3966	693	49	4708
Amravati	5783	1025	107	6915
Beed	7893	2590	270	10753
Bhandara	1842	55	6	1903
Buldhana	6812	1205	142	8159
Chandrapur	4014	375	86	4475
Chatrapati Sambhaji Nagar	114	81	16	211
Chhatrapati Sambhajinagar	12785	5239	379	18403
Dharashiv	3392	741	37	4170

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(14,6))
sns.lineplot(data=df_Maharashtra_dist_level_mh[['age_0_5','age_5_17','age_18
plt.title('Top 10 Districts in Maharashtra - Age Group Enrollment Comparisor
plt.xlabel('Districts')
plt.ylabel('Number of Enrollments')
plt.legend(title='Age Groups', labels=['Age 0-5', 'Age 5-17', 'Age 18 and ab
plt.xticks(ticks=range(len(df_Maharashtra_dist_level_mh.index)), labels=df_M
plt.show()
```



In [ ]:
```
df_Maharashtra_pincode_level = df_Maharashtra_cleaned_mh.groupby('pincode')[
df_Maharashtra_pincode_level['total_enrollment'] = df_Maharashtra_pincode_le
df_Maharashtra_pincode_level.shape
```

Out[ ]: (1580, 4)

In [ ]:
```
df_Maharashtra_pincode_level.sort_values("total_enrollment",ascending=False)
# df_Maharashtra_pincode_level.to_excel('Maharashtra_pincode_level_enrolment

df_Maharashtra_pincode_level
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
400001	44	14	6	64
400002	95	21	7	123
400003	165	18	6	189
400004	180	29	4	213
400005	143	32	7	182
...	...	...	...	...
445307	28	2	0	30
445308	31	6	1	38
445323	68	23	0	91
445401	65	16	2	83
445402	94	13	4	111

1580 rows × 4 columns

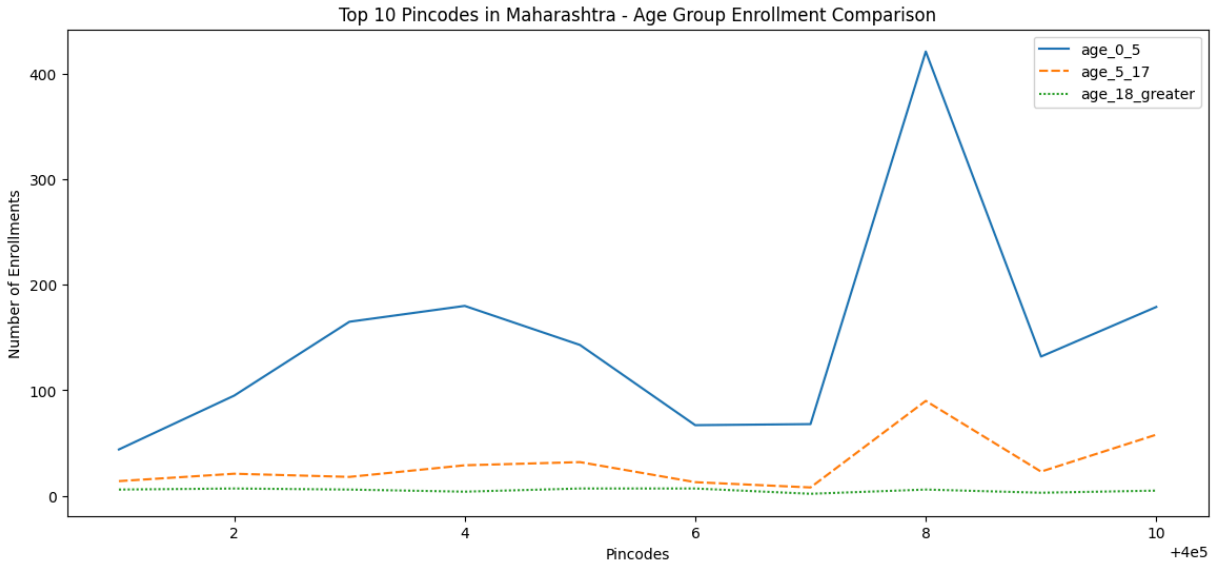
In [ ]:
```
df_Maharashtra_pincode_level1 = df_Maharashtra_pincode_level.head(10)
df_Maharashtra_pincode_level1
```

Out[ ]:

pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
400001	44	14	6	64
400002	95	21	7	123
400003	165	18	6	189
400004	180	29	4	213
400005	143	32	7	182
400006	67	13	7	87
400007	68	8	2	78
400008	421	90	6	517
400009	132	23	3	158
400010	179	58	5	242

In [ ]:
```
plt.figure(figsize=(14,6))
sns.lineplot(data=df_Maharashtra_pincode_level1[['age_0_5','age_5_17','age_1
plt.title('Top 10 Pincodes in Maharashtra - Age Group Enrollment Comparison'
plt.xlabel('Pincodes')
```

```
plt.ylabel('Number of Enrollments')
plt.show()
```



In [ ]:
```
## monthly enrolment trend in bihar
df_Maharashtra_monthly = df_Maharashtra_cleaned_mh.groupby('month')[['age_0_
df_Maharashtra_monthly['total_enrollment'] = df_Maharashtra_monthly['age_0_5
df_Maharashtra_monthly.shape
```

Out[ ]: (9, 4)

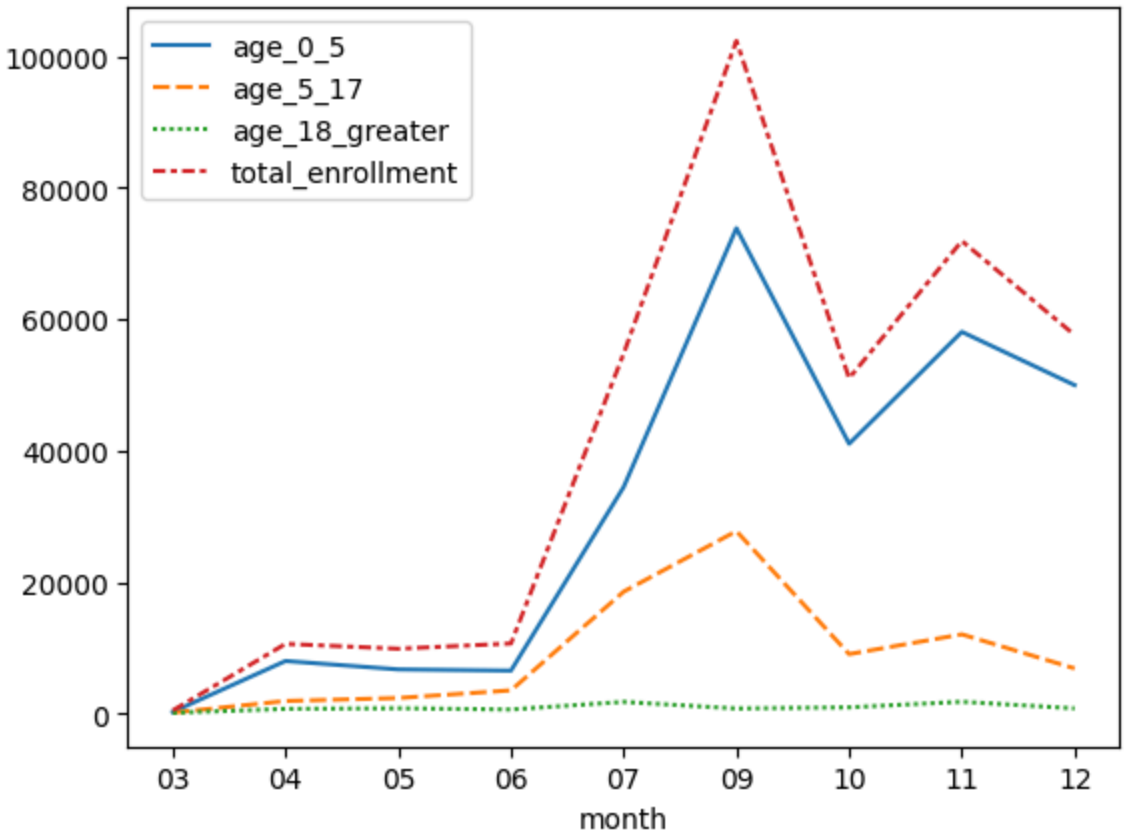
In [ ]:
```
df_Maharashtra_monthly
df_Maharashtra_monthly.sort_values("total_enrollment",ascending=False).reset
```

Out[ ]:

	month	age_0_5	age_5_17	age_18_greater	total_enrollment
0	09	73837	27751	753	102341
1	11	58064	12039	1783	71886
2	12	49960	6857	774	57591
3	07	34468	18536	1763	54767
4	10	41040	9032	933	51005
5	06	6521	3518	610	10649
6	04	7986	1901	704	10591
7	05	6706	2352	791	9849
8	03	232	130	98	460

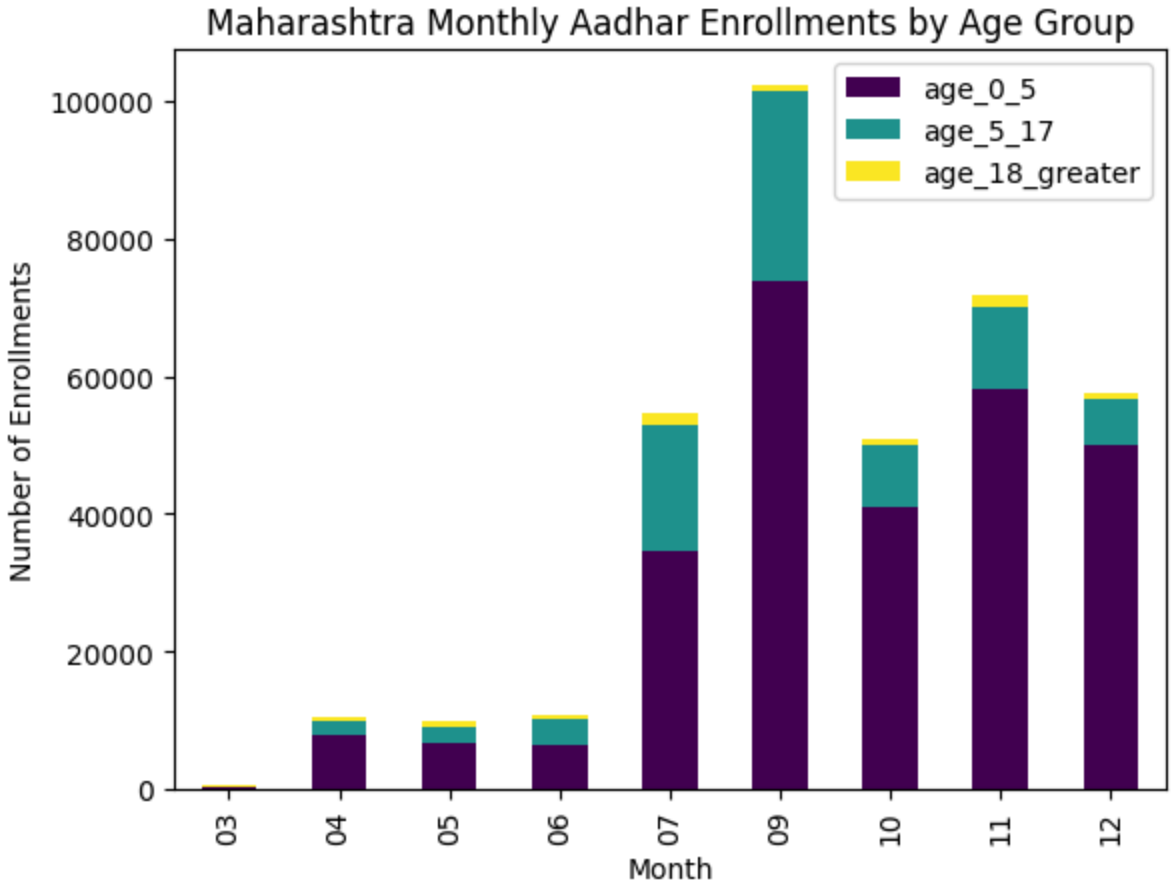
In [ ]:
```
sns.lineplot(data=df_Maharashtra_monthly)
```

Out[ ]: <Axes: xlabel='month'>



In [ ]:
```
## Stacked bar plot for age group comparison
plt.figure(figsize=(10,6))
df_Maharashtra_monthly[['age_0_5', 'age_5_17', 'age_18_greater']].plot(
    kind='bar',
    stacked=True,
    colormap='viridis'
)
plt.title('Maharashtra Monthly Aadhar Enrollments by Age Group')
plt.xlabel('Month')
plt.ylabel('Number of Enrollments')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```


Maharashtra Monthly Aadhar Enrollments by Age Group
Number of Enrollments
Month
age_0_5
age_5_17
age_18_greater
0
20000
40000
60000
80000
100000
03
04
05
06
07
09
10
11
12

In [ ]:
```
## # #
```