Importing enrolment file1

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        df1 = pd.read_csv('/Users/karansingh/Desktop/DAtaHackathon/api_data_aadhar_e
```

```
In [2]: 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 [3]: df1.shape
```

```
Out[3]: (500000, 7)
```

```
In [4]: df1.columns
```

```
Out[4]: Index(['date', 'state', 'district', 'pincode', 'age_0_5', 'age_5_17',
               'age_18_greater'],
              dtype='object')
```

```
In [5]: df1['state'].unique()
```

```
Out[5]: 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'], dtype=object)
```

In [6]:
```
df1['new_date'] = pd.to_datetime(df1['date'], format='%d-%m-%Y').dt.strftime
df1
```

Out[6]:

	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
...	...	...	...	...	...	...	...
499995	26-10-2025	Andhra Pradesh	Mahbubnagar	509207	1	0	0
499996	26-10-2025	Andhra Pradesh	Medak	502220	1	0	0
499997	26-10-2025	Andhra Pradesh	Medak	502256	0	1	0
499998	26-10-2025	Andhra Pradesh	Medak	502286	1	0	0
499999	26-10-2025	Andhra Pradesh	N. T. R	521402	1	0	0

500000 rows × 8 columns

In [7]:
```
df1['new_date'].isnull().sum()
```

Out[7]:
```
np.int64(0)
```

In [8]:
```
print(df1['new_date'].min())
print(df1['new_date'].max())
```

```
20250302
20251026
```

```
In [9]: ## check null value
        df1['state'].isnull().sum()
```

```
Out[9]: np.int64(0)
```

```
In [10]: ## check state column
         df1['state'].nunique()
```

```
Out[10]: 54
```

```
In [11]: df1['state'].unique()
```

```
Out[11]: 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'], dtype=object)
```

Import enrolment file2

```
In [12]: df2 = pd.read_csv('/Users/karansingh/Desktop/DAtaHackathon/api_data_aadhar_e
```

```
In [13]: 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 [14]: df2.shape
```

```
Out[14]: (500000, 7)
```

In [15]:

```
df2.columns
```

Out[15]:

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

In [16]:

```
df2['state'].unique()
```

Out[16]:

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

In [17]:

```
df2['new_date']=pd.to_datetime(df2['date']).dt.strftime('%Y%m%d')
df2
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_24203/548265193.p
y:1: UserWarning: Parsing dates in %d-%m-%Y format when dayfirst=False (the
default) was specified. Pass `dayfirst=True` or specify a format to silence
this warning.
  df2['new_date']=pd.to_datetime(df2['date']).dt.strftime('%Y%m%d')
```

Out[17]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	n
0	26-10-2025	Andhra Pradesh	Nalgonda	508004	0	1	0	2
1	26-10-2025	Andhra Pradesh	Nalgonda	508238	1	0	0	2
2	26-10-2025	Andhra Pradesh	Nalgonda	508278	1	0	0	2
3	26-10-2025	Andhra Pradesh	Nandyal	518432	0	1	0	2
4	26-10-2025	Andhra Pradesh	Nandyal	518543	1	0	0	2
...	...	...	...	...	...	...	...	
499995	31-12-2025	Telangana	Hyderabad	500045	4	5	1	2
499996	31-12-2025	Telangana	Hyderabad	500057	0	2	0	2
499997	31-12-2025	Telangana	Hyderabad	500061	4	2	0	2
499998	31-12-2025	Telangana	Hyderabad	500062	1	4	0	2
499999	31-12-2025	Telangana	Hyderabad	500095	0	1	0	2

500000 rows × 8 columns

In [18]:

```
df2['new_date'].isnull().sum()
```

Out[18]:

```
np.int64(0)
```

In [19]:

```
print(df2['new_date'].min())
print(df2['new_date'].max())
```

```
20251026
20251231
```

In [20]:

```
## check null value
df2['state'].isnull().sum()
```

```
Out[20]:  np.int64(0)
```

```
In [21]:  df2['state'].nunique()
```

```
Out[21]:  52
```

```
In [22]:  df2['state'].unique()
```

```
Out[22]:  array(['Andhra Pradesh', 'Arunachal Pradesh', 'Assam', 'West Bengal',
                 'Chhattisgarh', 'Delhi', 'Goa', 'Gujarat', 'Haryana',
                 'Himachal Pradesh', 'Jammu and Kashmir', 'Jharkhand', 'Karnataka',
                 'Kerala', 'Ladakh', 'Lakshadweep', 'Madhya Pradesh', 'Maharashtra',
                 'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Orissa',
                 'Pondicherry', 'Puducherry', 'Punjab', 'Rajasthan', 'Tamil Nadu',
                 'Telangana', 'Tripura', 'Uttar Pradesh', 'Uttarakhand',
                 'Andaman & Nicobar Islands', 'Andaman and Nicobar Islands',
                 'Bihar', 'Chandigarh', 'Sikkim', 'West Bangal',
                 'Dadra and Nagar Haveli', 'Daman and Diu',
                 'Dadra and Nagar Haveli and Daman and Diu', 'Jammu & Kashmir',
                 'andhra pradesh', 'Dadra & Nagar Haveli', 'Westbengal',
                 'Daman & Diu', 'WESTBENGAL', 'West bengal', 'West  Bengal',
                 'WEST BENGAL', '100000'], dtype=object)
```

Import enrolment file 3

```
In [23]:  df3 = pd.read_csv('/Users/karansingh/Desktop/DAtaHackathon/api_data_aadhar_e
          df3
```

Out[23]:

	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
...	...	...	...	...	...	...	...
6024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0
6025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0
6026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0
6027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0
6028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0

6029 rows × 7 columns

In [24]:

```
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 [25]:

```
df3.head()
```

Out[25]:

	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 [26]:

```
df3.shape
```

Out[26]:

```
(6029, 7)
```

In [27]:

```
df3['state'].unique()
```

Out[27]:

```
array(['Karnataka', 'Kerala', 'Ladakh', 'Lakshadweep', 'Madhya Pradesh',
       'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland',
       'Odisha', 'Orissa', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim',
       'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh',
       'Uttarakhand', 'West Bengal', 'Andhra Pradesh',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
       'Chhattisgarh', 'Daman and Diu', 'Delhi', 'Goa', 'Gujarat',
       'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir', 'Jharkhand',
       'Pondicherry'], dtype=object)
```

In [28]:

```
df3['new_date']=pd.to_datetime(df3['date']).dt.strftime('%Y%m%d')
df3
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_24203/880774394.p
y:1: UserWarning: Parsing dates in %d-%m-%Y format when dayfirst=False (the
default) was specified. Pass `dayfirst=True` or specify a format to silence
this warning.
  df3['new_date']=pd.to_datetime(df3['date']).dt.strftime('%Y%m%d')
```

Out[28]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_
0	31-12-2025	Karnataka	Bidar	585330	2	3	0	202
1	31-12-2025	Karnataka	Bidar	585402	6	0	0	202
2	31-12-2025	Karnataka	Bidar	585413	1	0	0	202
3	31-12-2025	Karnataka	Bidar	585418	1	2	0	202
4	31-12-2025	Karnataka	Bidar	585421	4	3	0	202
...	...	...	...	...	...	...	...	
6024	31-12-2025	West Bengal	West Midnapore	721149	2	0	0	202
6025	31-12-2025	West Bengal	West Midnapore	721150	2	2	0	202
6026	31-12-2025	West Bengal	West Midnapore	721305	0	1	0	202
6027	31-12-2025	West Bengal	West Midnapore	721504	1	0	0	202
6028	31-12-2025	West Bengal	West Midnapore	721517	2	1	0	202

6029 rows × 8 columns

In [29]:

```
df3['new_date'].isnull().sum()
```

Out[29]:

```
np.int64(0)
```

In [30]:

```
print(df3['new_date'].min())
print(df3['new_date'].max())
```

```
20251231
20251231
```

In [31]:

```
## check null value
df3['state'].isnull().sum()
```

```
Out[31]:  np.int64(0)
```

```
In [32]:  ## check state column
          df3['state'].nunique()
```

```
Out[32]:  37
```

```
In [33]:  df3['state'].unique()
```

```
Out[33]:  array(['Karnataka', 'Kerala', 'Ladakh', 'Lakshadweep', 'Madhya Pradesh',
                 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland',
                 'Odisha', 'Orissa', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim',
                 'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh',
                 'Uttarakhand', 'West Bengal', 'Andhra Pradesh',
                 'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
                 'Chhattisgarh', 'Daman and Diu', 'Delhi', 'Goa', 'Gujarat',
                 'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir', 'Jharkhand',
                 'Pondicherry'], dtype=object)
```

Merging three datasets

```
In [34]:  df = pd.concat([df1,df2,df3],ignore_index=True)
          df.shape
```

```
Out[34]:  (1006029, 8)
```

```
In [35]:  df.head()
```

Out[35]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_dat
0	02-03-2025	Meghalaya	East Khasi Hills	793121	11	61	37	2025030
1	09-03-2025	Karnataka	Bengaluru Urban	560043	14	33	39	2025030
2	09-03-2025	Uttar Pradesh	Kanpur Nagar	208001	29	82	12	2025030
3	09-03-2025	Uttar Pradesh	Aligarh	202133	62	29	15	2025030
4	09-03-2025	Karnataka	Bengaluru Urban	560016	14	16	21	2025030

```
In [36]:  df['new_date'].isnull().sum()
```

```
Out[36]:  np.int64(0)
```

In [37]:

```
df['state'].value_counts()
```

```
Out[37]:  state
          Uttar Pradesh                                   110369
          Tamil Nadu                                       92552
          Maharashtra                                      77191
          West Bengal                                      76519
          Karnataka                                        70198
          Andhra Pradesh                                   65658
          Bihar                                            60567
          Rajasthan                                        56159
          Madhya Pradesh                                   50225
          Gujarat                                          46624
          Odisha                                           43691
          Telangana                                        42774
          Kerala                                           39145
          Assam                                            31827
          Jharkhand                                        23218
          Punjab                                           20439
          Chhattisgarh                                     18550
          Haryana                                          15997
          Jammu and Kashmir                                11314
          Himachal Pradesh                                 10346
          Uttarakhand                                      10007
          Delhi                                             6804
          Meghalaya                                         3771
          Tripura                                           3729
          Orissa                                            3319
          Manipur                                           3218
          Nagaland                                          1999
          Arunachal Pradesh                                 1601
          Goa                                               1527
          Mizoram                                           1481
          Puducherry                                        1042
          Sikkim                                            1010
          Chandigarh                                         859
          Pondicherry                                        817
          Ladakh                                             304
          Andaman and Nicobar Islands                        289
          Dadra and Nagar Haveli                             162
          Lakshadweep                                        159
          Jammu & Kashmir                                    139
          Dadra and Nagar Haveli and Daman and Diu           116
          Andaman & Nicobar Islands                          103
          Daman and Diu                                       92
          Dadra & Nagar Haveli                                24
          100000                                              22
          Daman & Diu                                         20
          West  Bengal                                        15
          West Bangal                                          9
          West bengal                                          7
          Westbengal                                           6
          andhra pradesh                                       5
          WEST BENGAL                                          4
          Jammu And Kashmir                                    2
          The Dadra And Nagar Haveli And Daman And Diu         2
          ODISHA                                               1
```

```
WESTBENGAL                                         1
Name: count, dtype: int64
```

In [38]:

```
df['state'].nunique()
```

Out[38]:

```
55
```

In [39]:

```
df['state'].unique()
```

Out[39]:

```
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)
```

In [ ]:

```

```

In [40]:

```python
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 [41]:

```python
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 [42]:

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

In [43]:

```
# 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 [44]: df['state_clean'].nunique()
```

```
Out[44]: 36
```

```
In [45]: df['state_clean'].unique()
```

```
Out[45]: 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',
                'Dadra and Nagar Haveli and Daman and Diu', 'Arunachal Pradesh',
                'Himachal Pradesh', 'Goa', 'Ladakh', 'Andaman and Nicobar Islands',
                'Puducherry', 'Lakshadweep'], dtype=object)
```

```
In [46]: df
```

Out[46]:

	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
...	...	...	...	...	...	...	...
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

1006007 rows × 9 columns

In [47]:

```
df.dtypes
```

Out[47]:

```
date              object
state             object
district          object
pincode            int64
age_0_5            int64
age_5_17           int64
age_18_greater     int64
new_date          object
state_clean       object
dtype: object
```

In [48]:
```python
df_bihar= df[df['state_clean']=='Bihar']
df_bihar
```

Out[48]:

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

60567 rows × 9 columns

Bihar ke liye

In [ ]:
```python

```

In [49]:
```python
df_bihar['district'].unique()
## yha pr district ki bhi mapping krni padegi
## Aurangabad(bh)', 'Purnea', 'Pashchim Champaran', 'Sheikpura',
    #    'Bhabua', 'Aurangabad(BH)'], dtype=object)) issko dekho
```

Out[49]:

```
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)
```

In [50]:

```
df_bihar['district'].nunique()
```

Out[50]:

```
48
```

In [51]:

```
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 [52]:

```
## 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 [53]:

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

)
```

In [54]:

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

```
# count check
df['district_clean'].nunique()
```

Out[54]: 39

In [55]:

```
dff_bihar = df[df['state_clean']=='Bihar']
dff_bihar
```

Out[55]:

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

60567 rows × 10 columns

In [56]:

```
# 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()
```

```
---------------------------------------------------------------------------
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: 'district_clean'

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

KeyError                                  Traceback (most recent call last)
Cell In[56], line 2
      1 # Check Bihar-specific unmapped districts
----> 2 df_bihar_unmapped = df_bihar[df_bihar[                ].isna()]
      3 print(f"Unmapped Bihar districts count: {len(df_bihar_unmapped)}")
      4 df_bihar_unmapped['district'].unique()

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: 'district_clean'
```

```
In [ ]: df_bihar['district_clean'].unique()
```

Out[ ]:

```
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 [ ]:

```

```

In [ ]:

```python
df_bihar['new_date'].isnull().sum()
```

Out[ ]:

```
np.int64(0)
```

In [ ]:

```python
# unique pincodes in bihar
df_bihar['pincode'].nunique()
```

Out[ ]:

```
906
```

In [ ]:

```python
pincode_check = df_bihar.groupby('district_clean')['pincode'].nunique().rese
pincode_check
```

Out[ ]:

	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 [ ]:

```
df_bihar[df_bihar['district_clean']=='Kaimur']['pincode'].unique()
```

Out[ ]:

```
array([821106, 821108, 821109, 821105, 821110, 802132, 821101, 821102,
       821104, 821103, 821112, 821311])
```

In [ ]:

```
pin_district_count = (
    df_bihar.groupby('pincode')['district_clean']
    .nunique()
    .reset_index(name='district_count')
)
```

In [ ]:

```
pin_district_count
```

Out[ ]:

	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 [ ]:

```
problem_pins = pin_district_count[
    pin_district_count['district_count'] > 1
]
```

In [ ]:

```
problem_pins
## ek pin code 2 district se belong kr skta hai theek ye govt ki website pr
```

Out[ ]:

	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 [ ]:

```
df_flagged = df_bihar.merge(
    problem_pins[['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
```

Out[ ]:

	date	state	district	pincode	age_0_5	age_5_17	age_18_greater	new_d
0	09-03-2025	Bihar	Purbi Champaran	845418	30	48	10	202503
1	09-03-2025	Bihar	Purbi Champaran	845304	18	72	12	202503
2	15-03-2025	Bihar	Purbi Champaran	845303	12	121	13	20250
3	01-04-2025	Bihar	Sitamarhi	843315	102	125	18	20250
4	01-04-2025	Bihar	Munger	811213	191	278	22	20250
...	...	...	...	...	...	...	...	
16964	31-12-2025	Bihar	Sheohar	843325	4	2	0	20251
16965	31-12-2025	Bihar	Sitamarhi	843325	11	6	0	20251
16966	31-12-2025	Bihar	Siwan	841243	2	11	0	20251
16967	31-12-2025	Bihar	Supaul	852108	0	9	0	20251
16968	31-12-2025	Bihar	Supaul	852131	15	19	0	20251

16969 rows × 10 columns

In [ ]:

```
flagged_pincode=df_flagged.groupby(['district_clean','pincode'])[['age_0_5',
#flagged_pincode.to_excel('flagged_pincode_domain.xlsx')
```

In [ ]:

```
flagged_pincode['total_enrollment']=flagged_pincode['age_0_5']+flagged_pinco
flagged_pincode.sort_values('pincode')
```

Out[ ]:

	district_clean	pincode	age_0_5	age_5_17	age_18_greater	total_enrollment
241	Patna	801304	17	36	1	54
223	Nalanda	801304	58	114	0	172
224	Nalanda	801305	39	62	1	102
242	Patna	801305	12	33	2	47
63	Buxar	802112	117	224	1	342
...	...	...	...	...	...	...
282	Purnia	855105	61	18	2	81
283	Purnia	855107	235	70	0	305
166	Kishanganj	855107	1685	446	1	2132
167	Kishanganj	855115	774	208	4	986
284	Purnia	855115	222	60	0	282

365 rows × 6 columns

In [ ]:

```
## isse pta chlega hai kiss pincode me jda aadhar enrolment ho rha hai is pi
## baad me agr jarurat pde too hum iska flag bna skte hai ki jo bhi district
## same pin code se hai usse true ur jo ek hi pincode se hai usse false
```

In [ ]:

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

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_bihar['pin_multi_district_flag']=(
    df_bihar.groupby('pincode')['district_clean']
    .transform('nunique')>1
)
```

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_23469/378759105.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_bihar['pin_multi_district_flag']=(
```

In [57]:

```
pin_district_map= (
    df_bihar[df_bihar['pin_multi_district_flag']]
    .groupby('pincode')['district_clean']  # noqa: SC100
    .unique()
    .reset_index()
)
```

```
---------------------------------------------------------------------------
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: 'pin_multi_district_flag'

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

KeyError                                  Traceback (most recent call last)
Cell In[57], line 2
      1 pin_district_map= (
----> 2     df_bihar[df_bihar[                         ]]
      3     .groupby('pincode')['district_clean']  # noqa: SC100
      4     .unique()
      5     .reset_index()
      6 )

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: 'pin_multi_district_flag'
```

In [ ]:

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

```
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_23469/2055184884.
py: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['new_date'].astype(str).str[4:6]
/var/folders/bf/c5g8t8hs08j4x62fvvx5j4100000gn/T/ipykernel_23469/2055184884.
py: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['new_date'].astype(str).str[4:6]
```

Out[ ]:

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

60567 rows × 11 columns

In [ ]:

```
df_bihar_cleaned=df_bihar.drop(columns=['date','district','state'], axis=1)
df_bihar_cleaned
```

Out[ ]:

	pincode	age_0_5	age_5_17	age_18_greater	new_date	state_clean	distri
5	843331	20	49	12	20250309	Bihar	S
6	843330	23	24	42	20250309	Bihar	S
9	845418	30	48	10	20250309	Bihar	Ch
11	843317	35	94	16	20250309	Bihar	S
13	843324	49	186	34	20250309	Bihar	S
...	...	...	...	...	...	...	
1002992	844134	2	0	0	20251231	Bihar	
1002993	844504	15	26	1	20251231	Bihar	
1002994	844509	1	2	0	20251231	Bihar	
1002995	845404	13	17	1	20251231	Bihar	Ch
1002996	845449	9	45	0	20251231	Bihar	Ch

60567 rows × 7 columns

In [ ]: