GDP ANALYSIS USING DATA MINING TECHNIQUES

Submitted in fulfilment for the Component of
Data Mining Techniques (ITE2006)

CAL Course
in
B.Tech. – Information Technology

by
Rutul Ramesh Jadhav (20BIT0037)

Under the guidance of
Professor Ramkumar T

School of Information technology and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

April 2023

TABLE OF CONTENTS

1. Introduction –

1.1 Objective –

- GDP analysis is important because it gives information about the size of the economy and how an economy is performing.
- For Chief Ministers (CMs) of various states, it will help them to prioritise areas of development for their respective states so that development will be at faster rate.
- Since different states are in different phases of development, the recommendations should be specific to the states and areas where it is required.
- The overall goal of this project is to help the CMs focus on areas that will help to encourage the economic development for their respective states.
- Since the most common measure of economic development is the GDP, so in this project we will analyse the GDP of the various states of India and suggest ways to improve it.

2. Dataset Description –

- The data is taken from Indian Government website which is Open Government Data (OCD) platform of India: https://data.gov.in/
- The dataset is divided into Two parts:

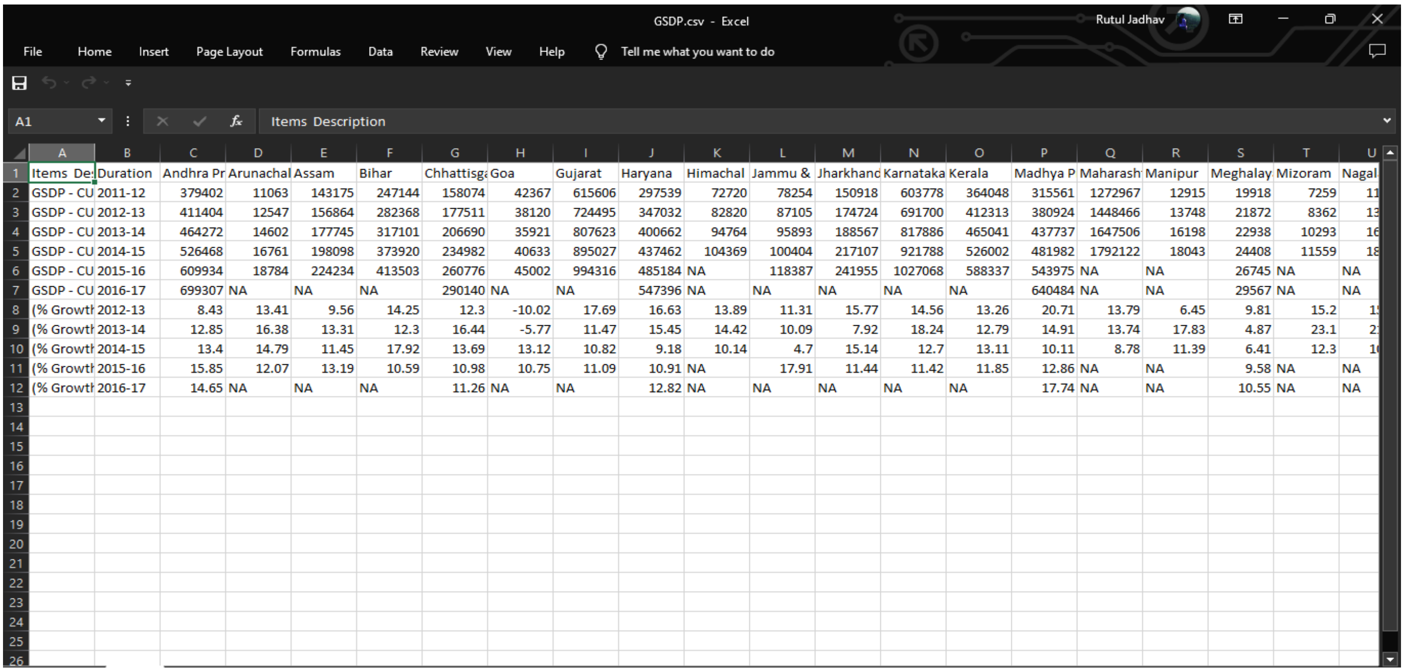
- Data 1: This dataset consists of Gross State Domestic Product data for the respective states and union territories
- Data 2: This includes distribution of GSDP into three sectors: -

a. Primary sector (which include Agriculture)
b. Secondary sector (which include Industry)
c. Tertiary sector (which include services along with taxes and subsidies).

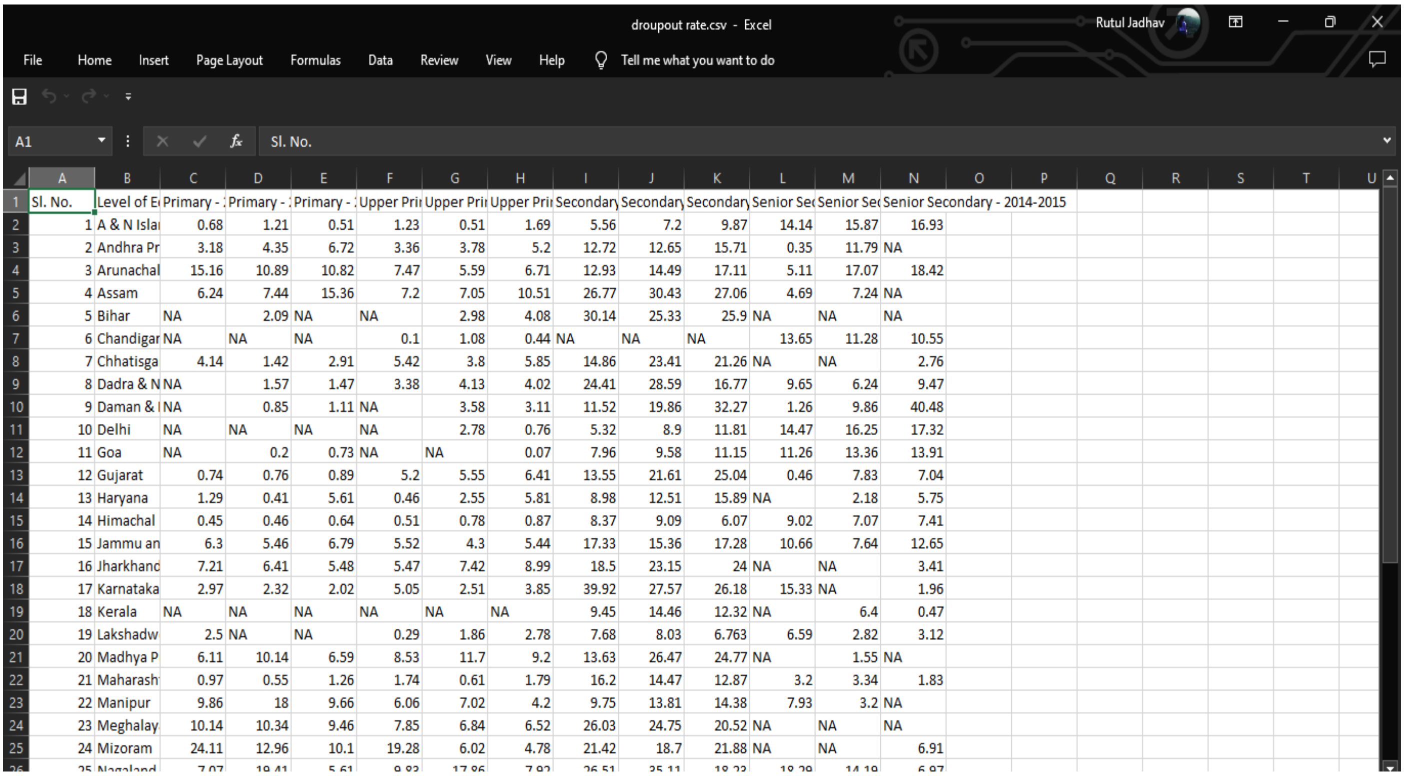
- In this we will analyse and compare the GDPs of various states like Andhra Pradesh, Tamil Nadu, Maharashtra, Bihar, Telangana etc. along with total and per capita.
- The GDP of state is mostly referred to as the GSDP which is Gross State Domestic Product. Then we will divide the states into four categories based on their GDP per capita and for each of these four categories we will analyse the sectors that contributes the most to the GDP (such as real estate, manufacturing etc.).
- Then we will analyse whether GDP per capita is related to dropout rates in schools and colleges.

  We will be importing python libraries like: -

  - ➢ import numpy as np
  - ➢ import pandas as pd
  - ➢ import matplotlib.pyplot as plt
  - ➢ import seaborn as sns



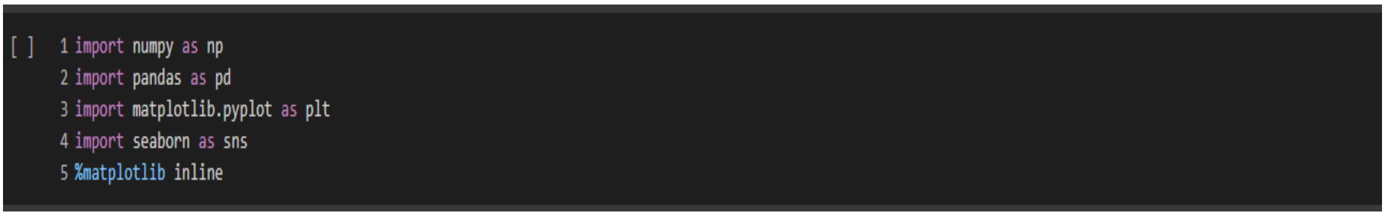
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Items De:	Duration	Andhra Pr	Arunachal	Assam	Bihar	Chhattisga	Goa	Gujarat	Haryana	Himachal	Jammu &	Jharkhand	Karnataka	Kerala	Madhya P	Maharash	Manipur	Meghalay	Mizoram	Nagal
2	GSDP - CU	2011-12	379402	11063	143175	247144	158074	42367	615606	297539	72720	78254	150918	603778	364048	315561	1272967	12915	19918	7259	11
3	GSDP - CU	2012-13	411404	12547	156864	282368	177511	38120	724495	347032	82820	87105	174724	691700	412313	380924	1448466	13748	21872	8362	13
4	GSDP - CU	2013-14	464272	14602	177745	317101	206690	35921	807623	400662	94764	95893	188567	817886	465041	437737	1647506	16198	22938	10293	16
5	GSDP - CU	2014-15	526468	16761	198098	373920	234982	40633	895027	437462	104369	100404	217107	921788	526002	481982	1792122	18043	24408	11559	18
6	GSDP - CU	2015-16	609934	18784	224234	413503	260776	45002	994316	485184	NA	118387	241955	1027068	588337	543975	NA	NA	26745	NA	NA
7	GSDP - CU	2016-17	699307	NA	NA	NA	290140	NA	NA	547396	NA	NA	NA	NA	NA	640484	NA	NA	29567	NA	NA
8	(% Growth	2012-13	8.43	13.41	9.56	14.25	12.3	-10.02	17.69	16.63	13.89	11.31	15.77	14.56	13.26	20.71	13.79	6.45	9.81	15.2	15
9	(% Growth	2013-14	12.85	16.38	13.31	12.3	16.44	-5.77	11.47	15.45	14.42	10.09	7.92	18.24	12.79	14.91	13.74	17.83	4.87	23.1	2
10	(% Growth	2014-15	13.4	14.79	11.45	17.92	13.69	13.12	10.82	9.18	10.14	4.7	15.14	12.7	13.11	10.11	8.78	11.39	6.41	12.3	1
11	(% Growth	2015-16	15.85	12.07	13.19	10.59	10.98	10.75	11.09	10.91	NA	17.91	11.44	11.42	11.85	12.86	NA	NA	9.58	NA	NA
12	(% Growth	2016-17	14.65	NA	NA	NA	11.26	NA	NA	12.82	NA	NA	NA	NA	NA	17.74	NA	NA	10.55	NA	NA



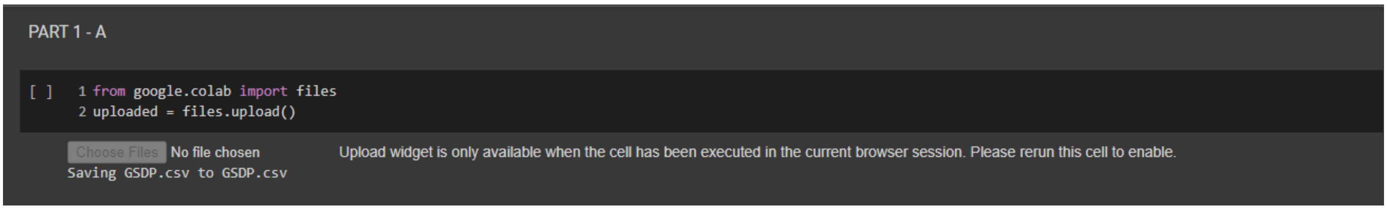
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Sl. No.	Level of E	Primary -	Primary -	Primary -	Upper Pri	Upper Pri	Upper Pri	Secondary	Secondary	Secondary	Senior Se	Senior Se	Senior Secondary - 2014-2015		
2	1	A & N Isla	0.68	1.21	0.51	1.23	0.51	1.69	5.56	7.2	9.87	14.14	15.87	16.93		
3	2	Andhra Pr	3.18	4.35	6.72	3.36	3.78	5.2	12.72	12.65	15.71	0.35	11.79	NA		
4	3	Arunachal	15.16	10.89	10.82	7.47	5.59	6.71	12.93	14.49	17.11	5.11	17.07	18.42		
5	4	Assam	6.24	7.44	15.36	7.2	7.05	10.51	26.77	30.43	27.06	4.69	7.24	NA		
6	5	Bihar	NA	2.09	NA	NA	2.98	4.08	30.14	25.33	25.9	NA	NA	NA		
7	6	Chandigar	NA	NA	NA	0.1	1.08	0.44	NA	NA	NA	13.65	11.28	10.55		
8	7	Chhatisga	4.14	1.42	2.91	5.42	3.8	5.85	14.86	23.41	21.26	NA	NA	2.76		
9	8	Dadra & N	NA	1.57	1.47	3.38	4.13	4.02	24.41	28.59	16.77	9.65	6.24	9.47		
10	9	Daman & I	NA	0.85	1.11	NA	3.58	3.11	11.52	19.86	32.27	1.26	9.86	40.48		
11	10	Delhi	NA	NA	NA	NA	2.78	0.76	5.32	8.9	11.81	14.47	16.25	17.32		
12	11	Goa	NA	0.2	0.73	NA	NA	0.07	7.96	9.58	11.15	11.26	13.36	13.91		
13	12	Gujarat	0.74	0.76	0.89	5.2	5.55	6.41	13.55	21.61	25.04	0.46	7.83	7.04		
14	13	Haryana	1.29	0.41	5.61	0.46	2.55	5.81	8.98	12.51	15.89	NA	2.18	5.75		
15	14	Himachal	0.45	0.46	0.64	0.51	0.78	0.87	8.37	9.09	6.07	9.02	7.07	7.41		
16	15	Jammu an	6.3	5.46	6.79	5.52	4.3	5.44	17.33	15.36	17.28	10.66	7.64	12.65		
17	16	Jharkhand	7.21	6.41	5.48	5.47	7.42	8.99	18.5	23.15	24	NA	NA	3.41		
18	17	Karnataka	2.97	2.32	2.02	5.05	2.51	3.85	39.92	27.57	26.18	15.33	NA	1.96		
19	18	Kerala	NA	NA	NA	NA	NA	NA	9.45	14.46	12.32	NA	6.4	0.47		
20	19	Lakshadw	2.5	NA	NA	0.29	1.86	2.78	7.68	8.03	6.763	6.59	2.82	3.12		
21	20	Madhya P	6.11	10.14	6.59	8.53	11.7	9.2	13.63	26.47	24.77	NA	1.55	NA		
22	21	Maharash	0.97	0.55	1.26	1.74	0.61	1.79	16.2	14.47	12.87	3.2	3.34	1.83		
23	22	Manipur	9.86	18	9.66	6.06	7.02	4.2	9.75	13.81	14.38	7.93	3.2	NA		
24	23	Meghalay	10.14	10.34	9.46	7.85	6.84	6.52	26.03	24.75	20.52	NA	NA	NA		
25	24	Mizoram	24.11	12.96	10.1	19.28	6.02	4.78	21.42	18.7	21.88	NA	NA	6.91		
26	25	Nagaland	7.07	19.41	5.61	9.83	17.86	7.92	26.51	35.11	18.23	18.29	14.19	6.97		

3. Data Pre-Processing Techniques –

3.1 Codes -



```
[ ]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline
```

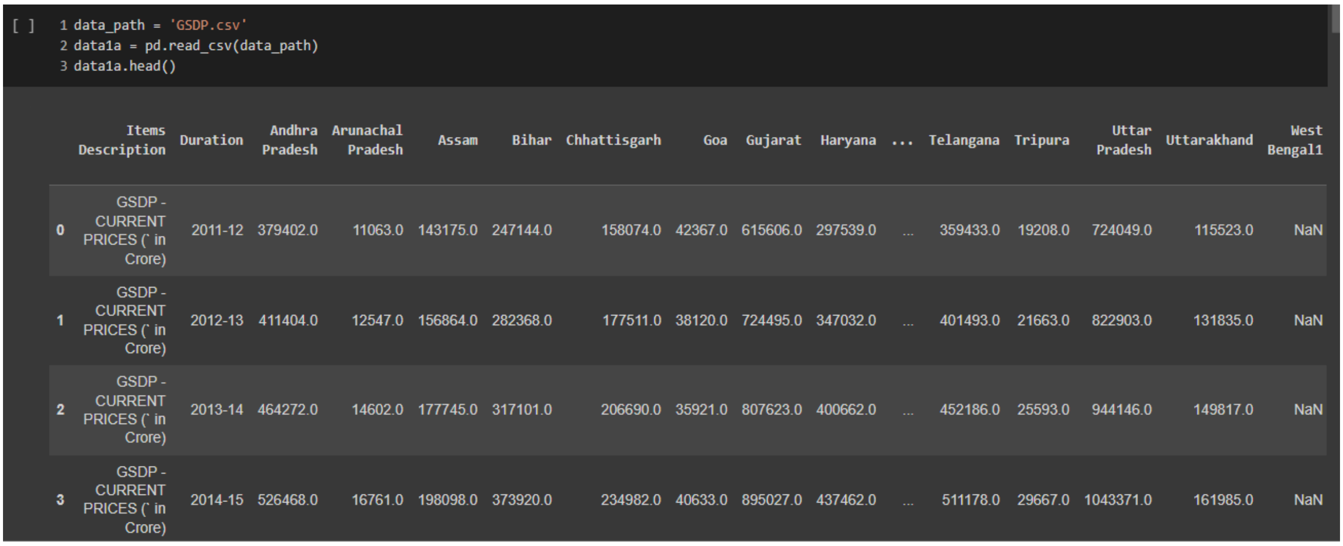


PART 1 - A

```
[ ]  from google.colab import files
     uploaded = files.upload()
```

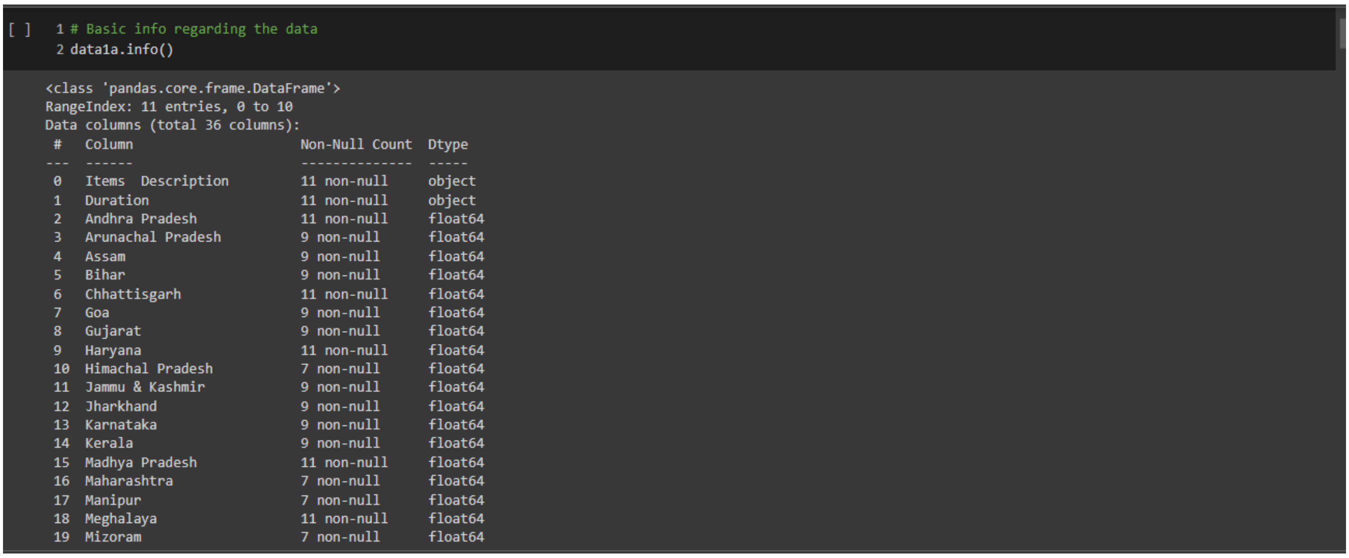
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving GSDP.csv to GSDP.csv
```



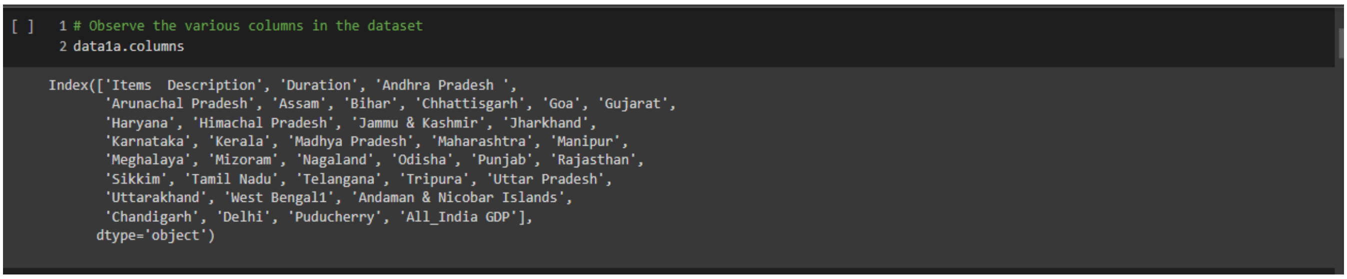
```
[ ]  data_path = 'GSDP.csv'
     data1a = pd.read_csv(data_path)
     data1a.head()
```

	Items Description	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	...	Telangana	Tripura	Uttar Pradesh	Uttarakhand	West Bengal1
0	GSDP - CURRENT PRICES (` in Crore)	2011-12	379402.0	11063.0	143175.0	247144.0	158074.0	42367.0	615606.0	297539.0	...	359433.0	19208.0	724049.0	115523.0	NaN
1	GSDP - CURRENT PRICES (` in Crore)	2012-13	411404.0	12547.0	156864.0	282368.0	177511.0	38120.0	724495.0	347032.0	...	401493.0	21663.0	822903.0	131835.0	NaN
2	GSDP - CURRENT PRICES (` in Crore)	2013-14	464272.0	14602.0	177745.0	317101.0	206690.0	35921.0	807623.0	400662.0	...	452186.0	25593.0	944146.0	149817.0	NaN
3	GSDP - CURRENT PRICES (` in Crore)	2014-15	526468.0	16761.0	198098.0	373920.0	234982.0	40633.0	895027.0	437462.0	...	511178.0	29667.0	1043371.0	161985.0	NaN



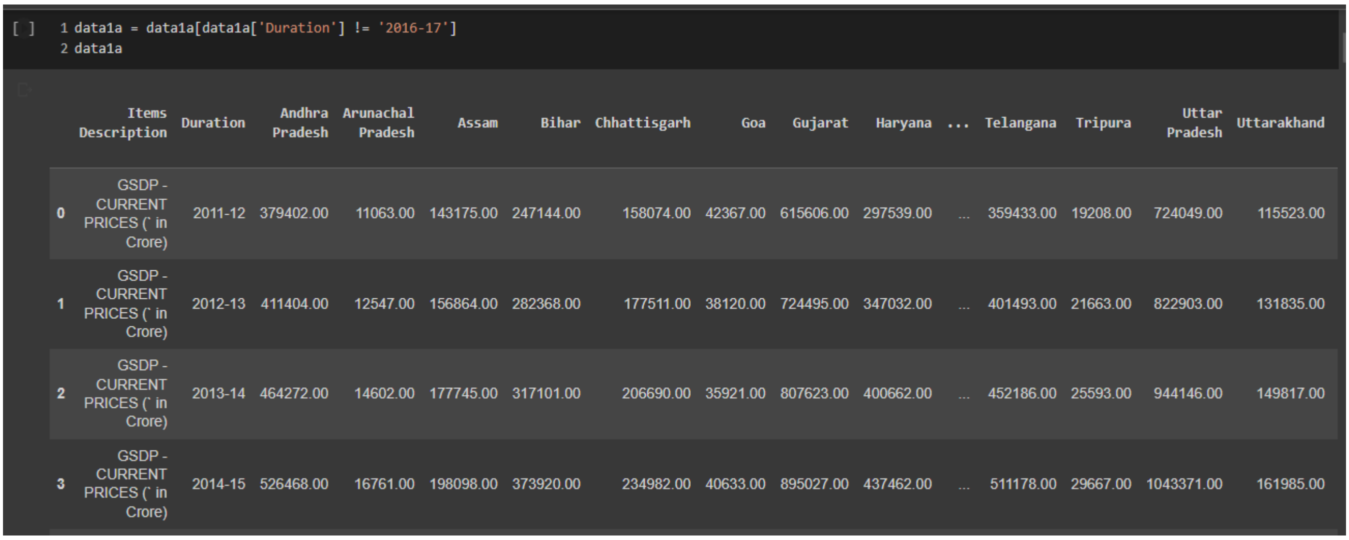
```
# Basic info regarding the data
data1a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 36 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Items  Description     11 non-null     object 
 1   Duration               11 non-null     object 
 2   Andhra Pradesh         11 non-null     float64
 3   Arunachal Pradesh      9 non-null      float64
 4   Assam                  9 non-null      float64
 5   Bihar                  9 non-null      float64
 6   Chhattisgarh           11 non-null     float64
 7   Goa                    9 non-null      float64
 8   Gujarat                9 non-null      float64
 9   Haryana                11 non-null     float64
 10  Himachal Pradesh       7 non-null      float64
 11  Jammu & Kashmir        9 non-null      float64
 12  Jharkhand              9 non-null      float64
 13  Karnataka              9 non-null      float64
 14  Kerala                 9 non-null      float64
 15  Madhya Pradesh         11 non-null     float64
 16  Maharashtra            7 non-null      float64
 17  Manipur                7 non-null      float64
 18  Meghalaya              11 non-null     float64
 19  Mizoram                7 non-null      float64
```



```
# Observe the various columns in the dataset
data1a.columns
```

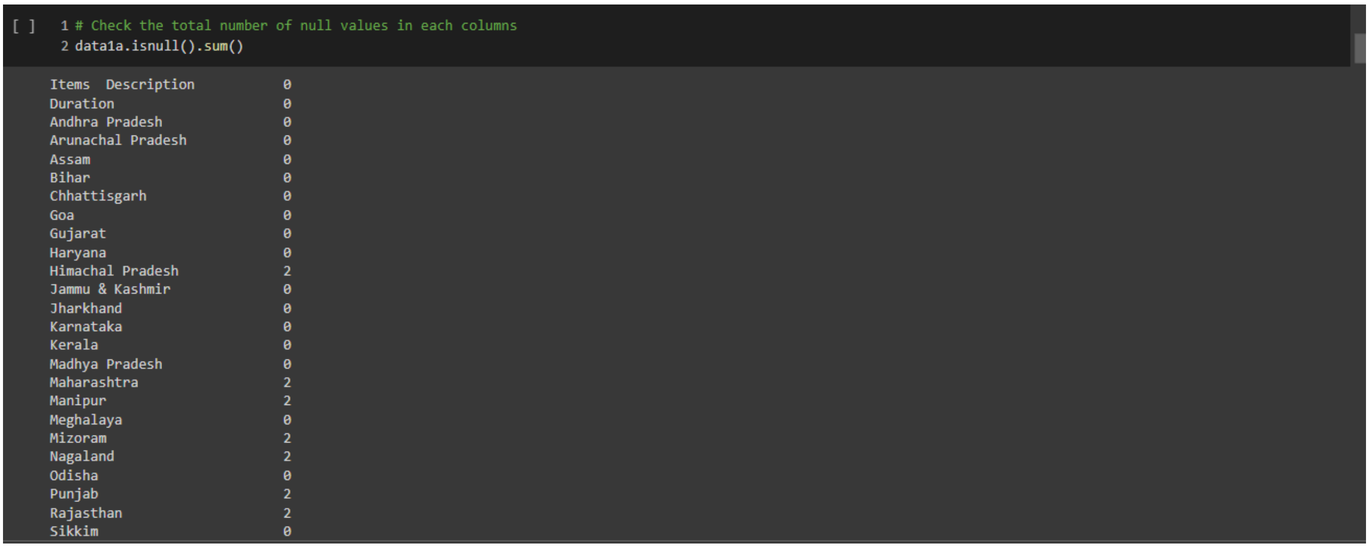
```
Index(['Items  Description', 'Duration', 'Andhra Pradesh ',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chhattisgarh', 'Goa', 'Gujarat',
       'Haryana', 'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand',
       'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
       'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Punjab', 'Rajasthan',
       'Sikkim', 'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh',
       'Uttarakhand', 'West Bengal1', 'Andaman & Nicobar Islands',
       'Chandigarh', 'Delhi', 'Puducherry', 'All_India GDP'],
      dtype='object')
```



```
data1a = data1a[data1a['Duration'] != '2016-17']
data1a
```

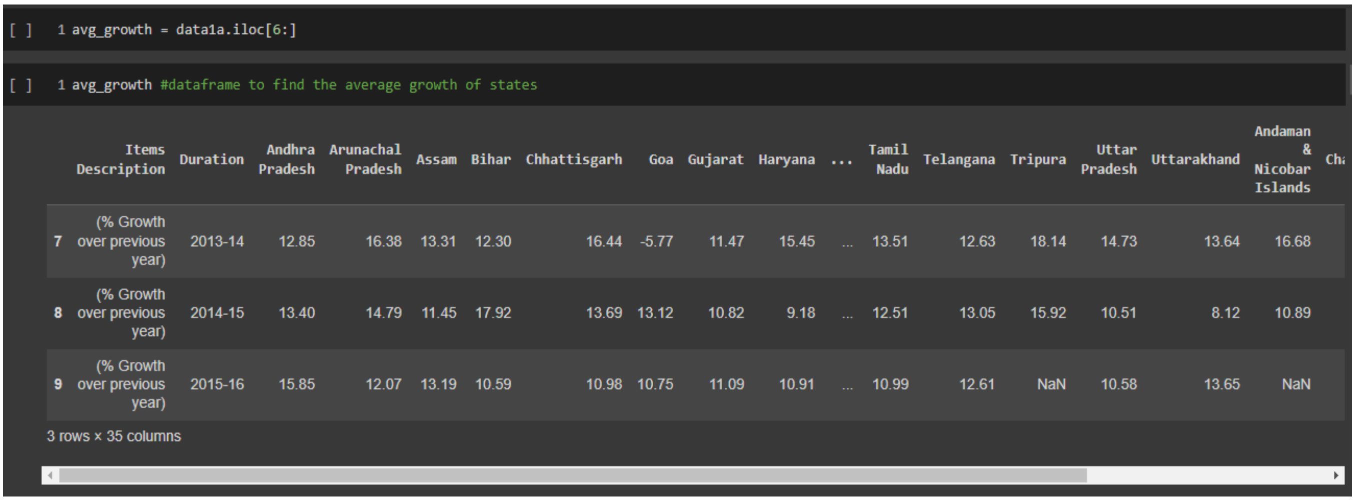
	Items Description	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	...	Telangana	Tripura	Uttar Pradesh	Uttarakhand
0	GSDP - CURRENT PRICES (` in Crore)	2011-12	379402.00	11063.00	143175.00	247144.00	158074.00	42367.00	615606.00	297539.00	...	359433.00	19208.00	724049.00	115523.00
1	GSDP - CURRENT PRICES (` in Crore)	2012-13	411404.00	12547.00	156864.00	282368.00	177511.00	38120.00	724495.00	347032.00	...	401493.00	21663.00	822903.00	131835.00
2	GSDP - CURRENT PRICES (` in Crore)	2013-14	464272.00	14602.00	177745.00	317101.00	206690.00	35921.00	807623.00	400662.00	...	452186.00	25593.00	944146.00	149817.00
3	GSDP - CURRENT PRICES (` in Crore)	2014-15	526468.00	16761.00	198098.00	373920.00	234982.00	40633.00	895027.00	437462.00	...	511178.00	29667.00	1043371.00	161985.00

Calculating the average growth of states for the duration 2013-14, 2014-15 and 2015-16 by taking the mean of the row '(% Growth over previous year)'.



```
[ ]  # Check the total number of null values in each columns
     data1a.isnull().sum()
```

```
Items  Description           0
Duration                     0
Andhra Pradesh               0
Arunachal Pradesh            0
Assam                        0
Bihar                        0
Chhattisgarh                 0
Goa                          0
Gujarat                      0
Haryana                      0
Himachal Pradesh             2
Jammu & Kashmir              0
Jharkhand                    0
Karnataka                    0
Kerala                       0
Madhya Pradesh               0
Maharashtra                  2
Manipur                      2
Meghalaya                    0
Mizoram                      2
Nagaland                     2
Odisha                       0
Punjab                       2
Rajasthan                    2
Sikkim                       0
```

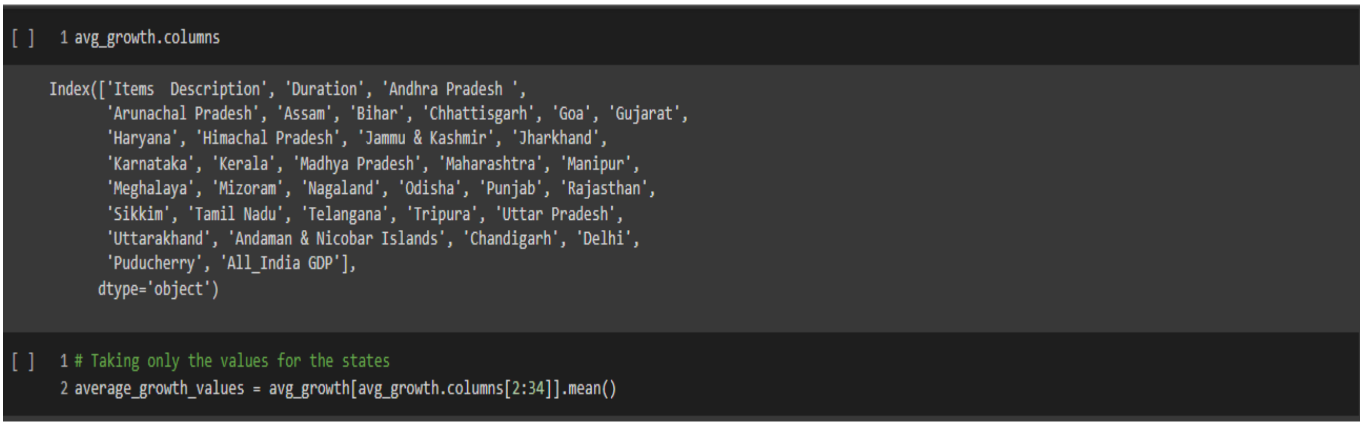


```
[ ]  avg_growth = data1a.iloc[6:]
```

```
[ ]  avg_growth #dataframe to find the average growth of states
```

	Items Description	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	...	Tamil Nadu	Telangana	Tripura	Uttar Pradesh	Uttarakhand	Andaman & Nicobar Islands	Cha
7	(% Growth over previous year)	2013-14	12.85	16.38	13.31	12.30	16.44	-5.77	11.47	15.45	...	13.51	12.63	18.14	14.73	13.64	16.68	
8	(% Growth over previous year)	2014-15	13.40	14.79	11.45	17.92	13.69	13.12	10.82	9.18	...	12.51	13.05	15.92	10.51	8.12	10.89	
9	(% Growth over previous year)	2015-16	15.85	12.07	13.19	10.59	10.98	10.75	11.09	10.91	...	10.99	12.61	NaN	10.58	13.65	NaN	

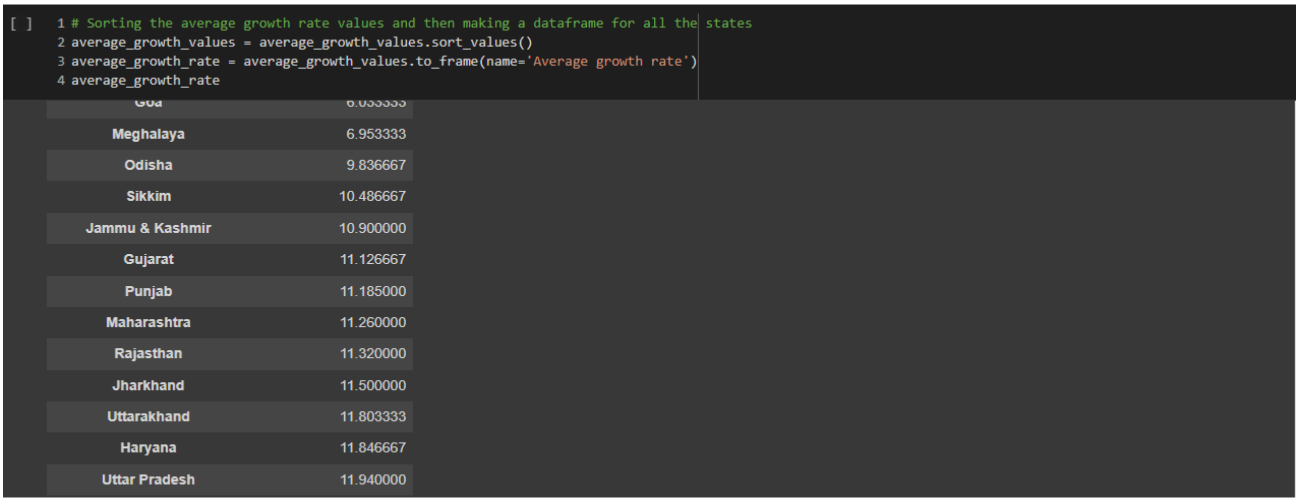
3 rows × 35 columns



```
[ ]  avg_growth.columns
```

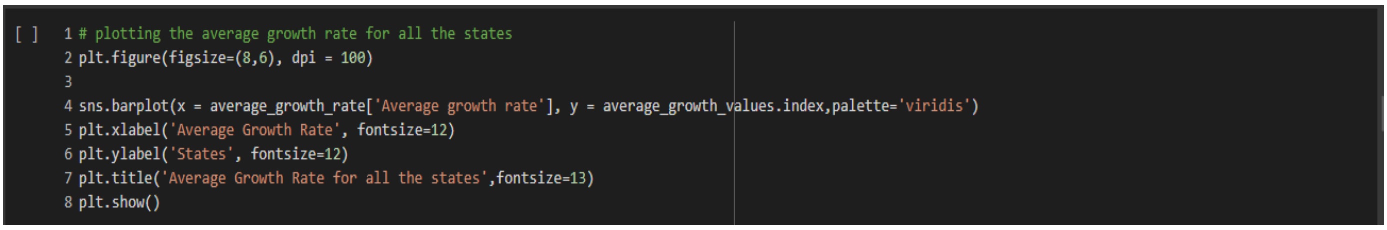
```
Index(['Items  Description', 'Duration', 'Andhra Pradesh ',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chhattisgarh', 'Goa', 'Gujarat',
       'Haryana', 'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand',
       'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
       'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Punjab', 'Rajasthan',
       'Sikkim', 'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh',
       'Uttarakhand', 'Andaman & Nicobar Islands', 'Chandigarh', 'Delhi',
       'Puducherry', 'All_India GDP'],
      dtype='object')
```

```
[ ]  # Taking only the values for the states
     average_growth_values = avg_growth[avg_growth.columns[2:34]].mean()
```

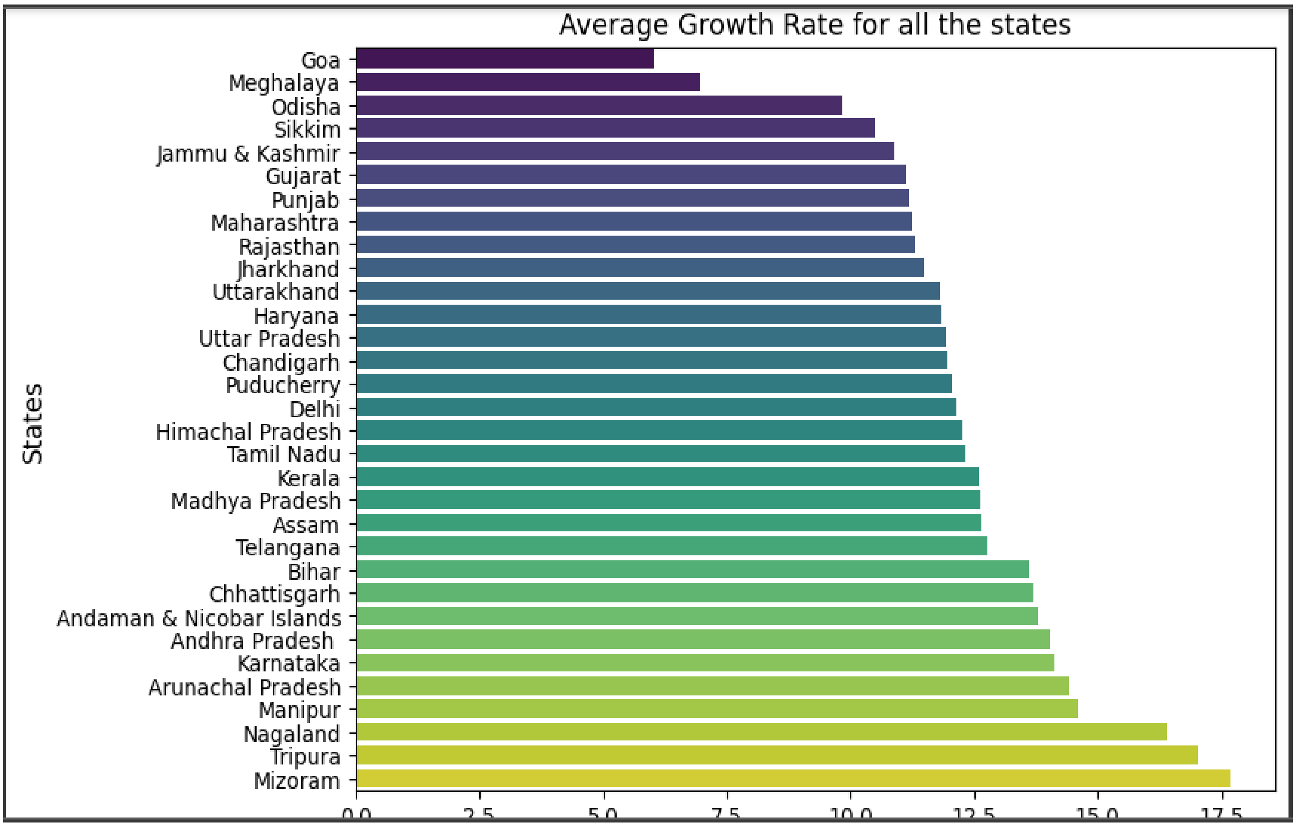


```
[ ]  # Sorting the average growth rate values and then making a dataframe for all the states
     average_growth_values = average_growth_values.sort_values()
     average_growth_rate = average_growth_values.to_frame(name='Average growth rate')
     average_growth_rate
```

Goa	6.033333
Meghalaya	6.953333
Odisha	9.836667
Sikkim	10.486667
Jammu & Kashmir	10.900000
Gujarat	11.126667
Punjab	11.185000
Maharashtra	11.260000
Rajasthan	11.320000
Jharkhand	11.500000
Uttarakhand	11.803333
Haryana	11.846667
Uttar Pradesh	11.940000

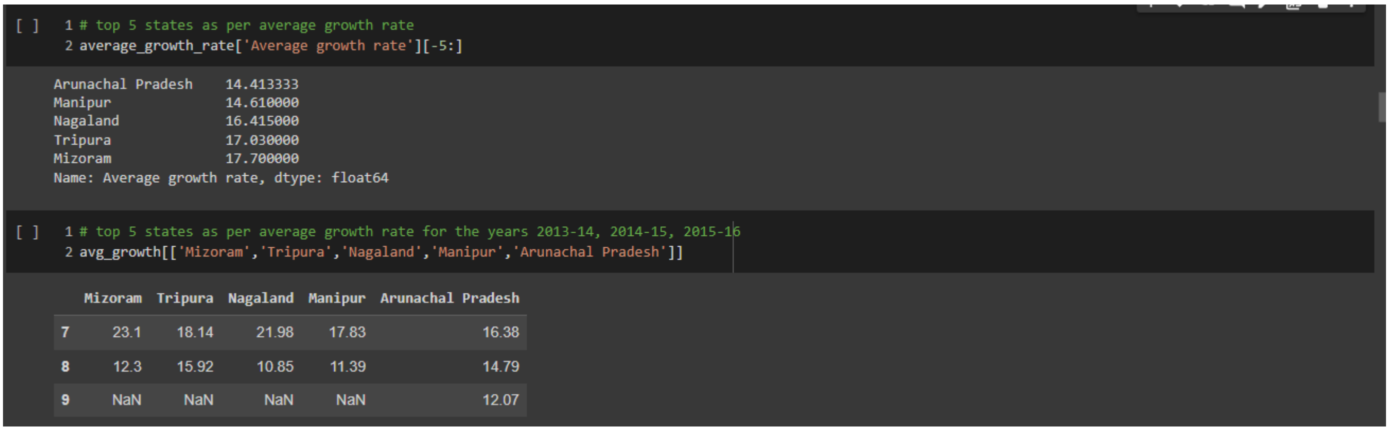


```
[ ]
# plotting the average growth rate for all the states
plt.figure(figsize=(8,6), dpi = 100)

sns.barplot(x = average_growth_rate['Average growth rate'], y = average_growth_values.index,palette='viridis')
plt.xlabel('Average Growth Rate', fontsize=12)
plt.ylabel('States', fontsize=12)
plt.title('Average Growth Rate for all the states',fontsize=13)
plt.show()
```



Observations: The average growth rate has been the maximum for the North East states except for Assam and Meghalaya which is not what we generally expect so we should take a further look at these states.

The average growth rate has been least for states like Goa, Odisha, Meghalaya, Sikkim, Jammu & Kashmir etc.



```
[ ]
# top 5 states as per average growth rate
average_growth_rate['Average growth rate'][-5:]
```

```
Arunachal Pradesh    14.413333
Manipur              14.610000
Nagaland             16.415000
Tripura              17.030000
Mizoram              17.700000
Name: Average growth rate, dtype: float64
```

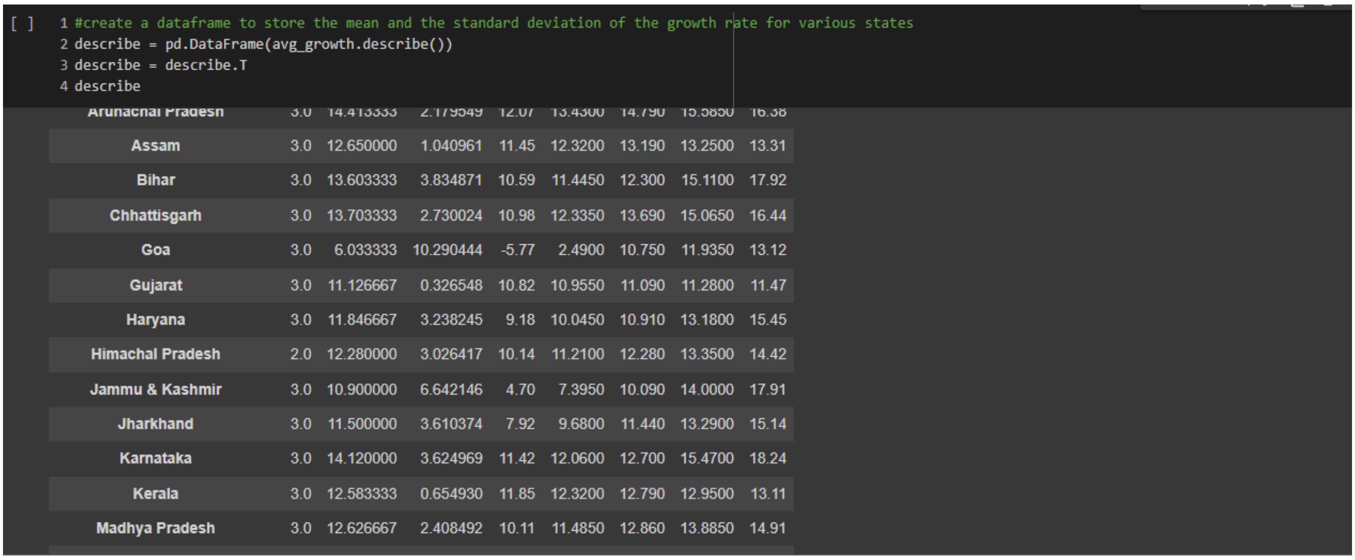
```
[ ]
# top 5 states as per average growth rate for the years 2013-14, 2014-15, 2015-16
avg_growth[['Mizoram','Tripura','Nagaland','Manipur','Arunachal Pradesh']]
```

	Mizoram	Tripura	Nagaland	Manipur	Arunachal Pradesh
7	23.1	18.14	21.98	17.83	16.38
8	12.3	15.92	10.85	11.39	14.79
9	NaN	NaN	NaN	NaN	12.07

The growth rate for the above states decreased substantially for the year 14-15 in comparison to the year 13-14 but as the growth rate was very high for the year 13-14 so the average is higher for these states. In the absence of data for the year 2015-

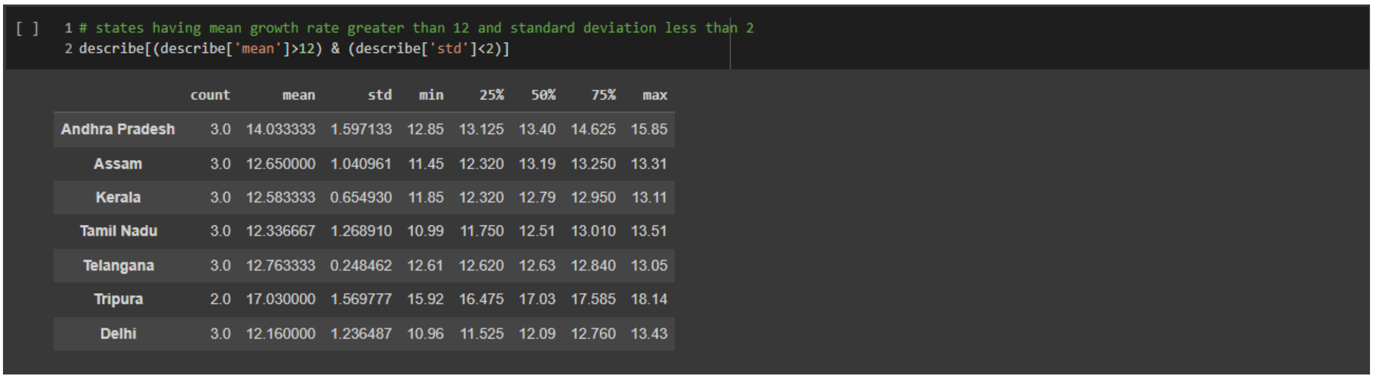
16 we cannot say definitively that these are high performing states as their growth rate decreased for the year 2014-15

To find out the states that have been growing continuously fast we need to look at the Standard Deviation and the Mean growth rate for the states.



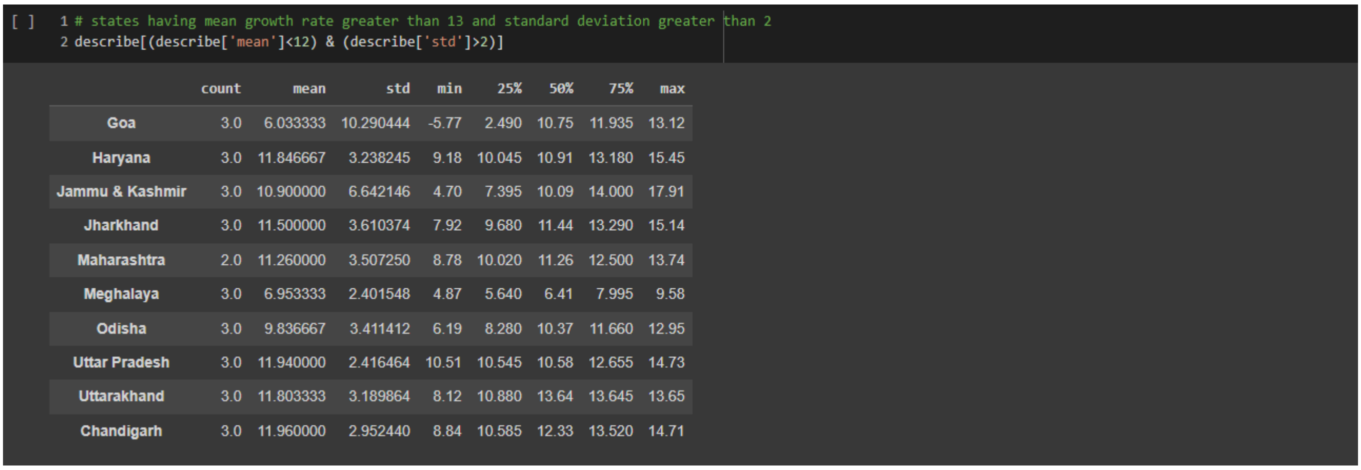
```
[ ]
#create a dataframe to store the mean and the standard deviation of the growth rate for various states
describe = pd.DataFrame(avg_growth.describe())
describe = describe.T
describe
```

Arunachal Pradesh	3.0	14.413333	2.179549	12.07	13.4300	14.790	15.5850	16.38
Assam	3.0	12.650000	1.040961	11.45	12.3200	13.190	13.2500	13.31
Bihar	3.0	13.603333	3.834871	10.59	11.4450	12.300	15.1100	17.92
Chhattisgarh	3.0	13.703333	2.730024	10.98	12.3350	13.690	15.0650	16.44
Goa	3.0	6.033333	10.290444	-5.77	2.4900	10.750	11.9350	13.12
Gujarat	3.0	11.126667	0.326548	10.82	10.9550	11.090	11.2800	11.47
Haryana	3.0	11.846667	3.238245	9.18	10.0450	10.910	13.1800	15.45
Himachal Pradesh	2.0	12.280000	3.026417	10.14	11.2100	12.280	13.3500	14.42
Jammu & Kashmir	3.0	10.900000	6.642146	4.70	7.3950	10.090	14.0000	17.91
Jharkhand	3.0	11.500000	3.610374	7.92	9.6800	11.440	13.2900	15.14
Karnataka	3.0	14.120000	3.624969	11.42	12.0600	12.700	15.4700	18.24
Kerala	3.0	12.583333	0.654930	11.85	12.3200	12.790	12.9500	13.11
Madhya Pradesh	3.0	12.626667	2.408492	10.11	11.4850	12.860	13.8850	14.91



```
[ ]
# states having mean growth rate greater than 12 and standard deviation less than 2
describe[(describe['mean']>12) & (describe['std']<2)]
```

	count	mean	std	min	25%	50%	75%	max
Andhra Pradesh	3.0	14.033333	1.597133	12.85	13.125	13.40	14.625	15.85
Assam	3.0	12.650000	1.040961	11.45	12.320	13.19	13.250	13.31
Kerala	3.0	12.583333	0.654930	11.85	12.320	12.79	12.950	13.11
Tamil Nadu	3.0	12.336667	1.268910	10.99	11.750	12.51	13.010	13.51
Telangana	3.0	12.763333	0.248462	12.61	12.620	12.63	12.840	13.05
Tripura	2.0	17.030000	1.569777	15.92	16.475	17.03	17.585	18.14
Delhi	3.0	12.160000	1.236487	10.96	11.525	12.09	12.760	13.43



```
[ ]
# states having mean growth rate greater than 13 and standard deviation greater than 2
describe[(describe['mean']<12) & (describe['std']>2)]
```

	count	mean	std	min	25%	50%	75%	max
Goa	3.0	6.033333	10.290444	-5.77	2.490	10.75	11.935	13.12
Haryana	3.0	11.846667	3.238245	9.18	10.045	10.91	13.180	15.45
Jammu & Kashmir	3.0	10.900000	6.642146	4.70	7.395	10.09	14.000	17.91
Jharkhand	3.0	11.500000	3.610374	7.92	9.680	11.44	13.290	15.14
Maharashtra	2.0	11.260000	3.507250	8.78	10.020	11.26	12.500	13.74
Meghalaya	3.0	6.953333	2.401548	4.87	5.640	6.41	7.995	9.58
Odisha	3.0	9.836667	3.411412	6.19	8.280	10.37	11.660	12.95
Uttar Pradesh	3.0	11.940000	2.416464	10.51	10.545	10.58	12.655	14.73
Uttarakhand	3.0	11.803333	3.189864	8.12	10.880	13.64	13.645	13.65
Chandigarh	3.0	11.960000	2.952440	8.84	10.585	12.33	13.520	14.71

Comparing the average growth rate for the year 2013-14, 2014-15, 2015-16 and the standard deviation.

States that are growing consistently fast are:

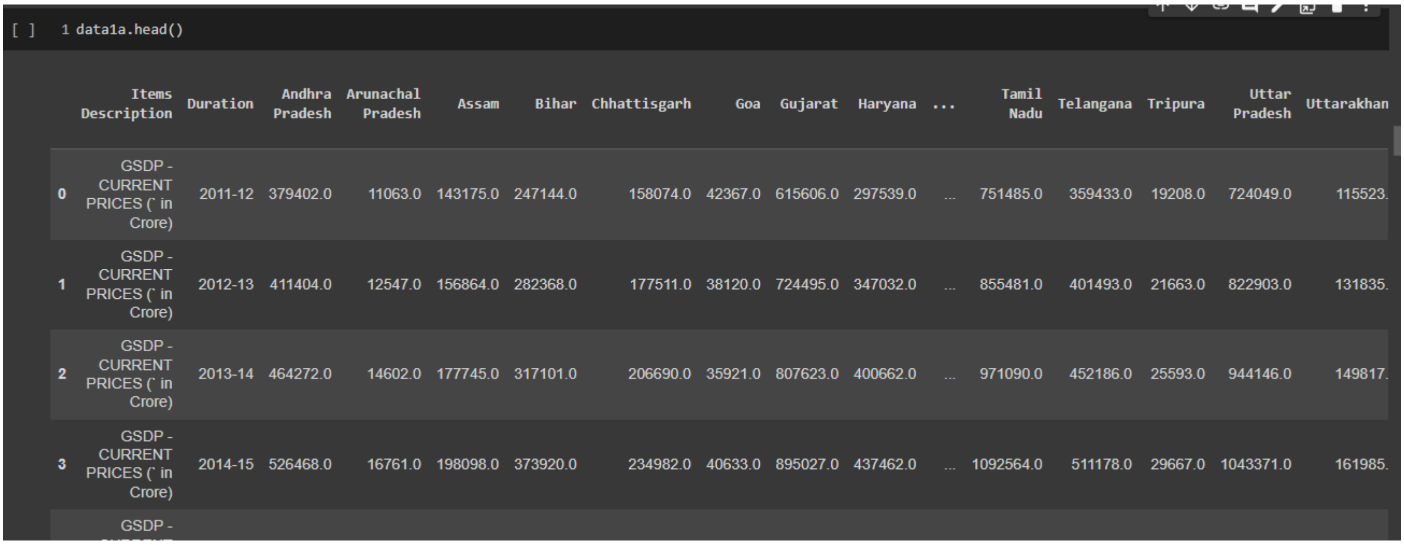
- Andhra Pradesh
- Assam

- Kerala
- Tamil Nadu
- Telangana

States that are struggling are:

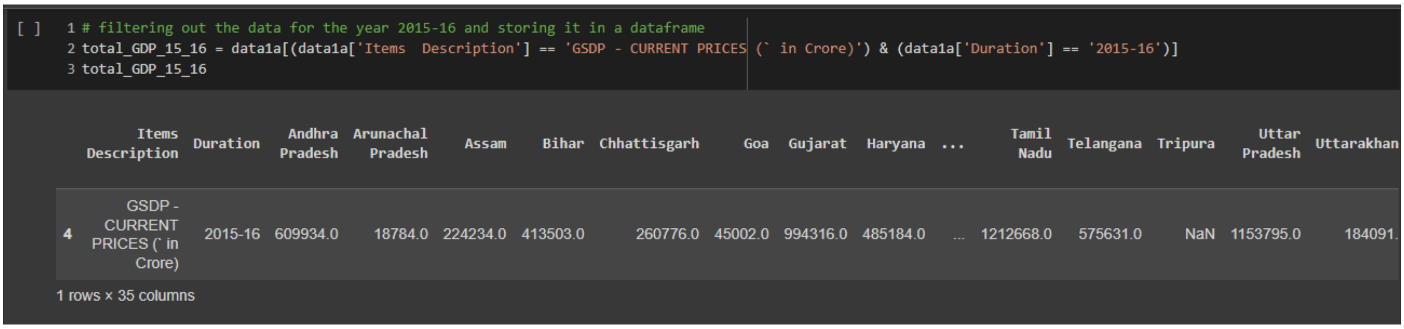
- Goa
- Meghalaya
- Odisha
- Jammu & Kashmir
- Jharkhand

Plotting the total GDP of the states for the year 2015-16: Identifying the top 5 and the bottom 5 states based on total GDP.



```
data1a.head()
```

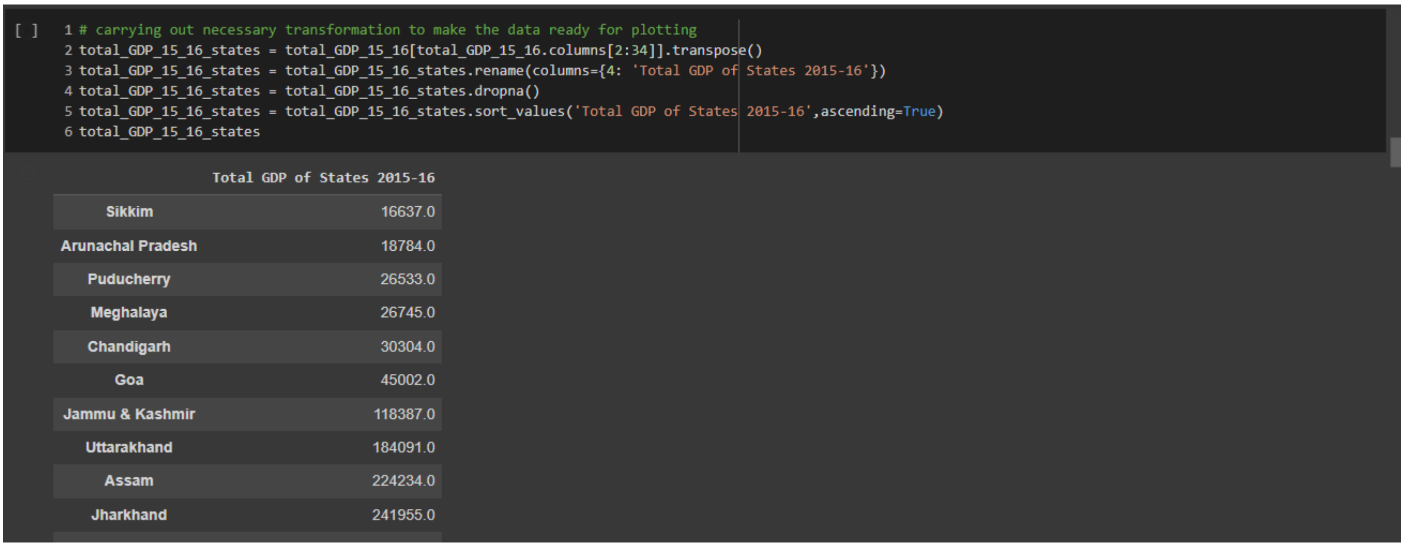
	Items Description	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	...	Tamil Nadu	Telangana	Tripura	Uttar Pradesh	Uttarakhan
0	GSDP - CURRENT PRICES (` in Crore)	2011-12	379402.0	11063.0	143175.0	247144.0	158074.0	42367.0	615606.0	297539.0	...	751485.0	359433.0	19208.0	724049.0	115523.
1	GSDP - CURRENT PRICES (` in Crore)	2012-13	411404.0	12547.0	156864.0	282368.0	177511.0	38120.0	724495.0	347032.0	...	855481.0	401493.0	21663.0	822903.0	131835.
2	GSDP - CURRENT PRICES (` in Crore)	2013-14	464272.0	14602.0	177745.0	317101.0	206690.0	35921.0	807623.0	400662.0	...	971090.0	452186.0	25593.0	944146.0	149817.
3	GSDP - CURRENT PRICES (` in Crore)	2014-15	526468.0	16761.0	198098.0	373920.0	234982.0	40633.0	895027.0	437462.0	...	1092564.0	511178.0	29667.0	1043371.0	161985.
	GSDP -															



```
# filtering out the data for the year 2015-16 and storing it in a dataframe
total_GDP_15_16 = data1a[(data1a['Items  Description'] == 'GSDP - CURRENT PRICES (` in Crore)') & (data1a['Duration'] == '2015-16')]
total_GDP_15_16
```

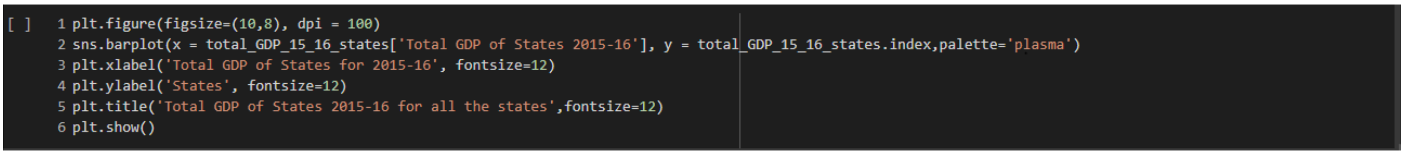
	Items Description	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	...	Tamil Nadu	Telangana	Tripura	Uttar Pradesh	Uttarakhan
4	GSDP - CURRENT PRICES (` in Crore)	2015-16	609934.0	18784.0	224234.0	413503.0	260776.0	45002.0	994316.0	485184.0	...	1212668.0	575631.0	NaN	1153795.0	184091.

1 rows × 35 columns

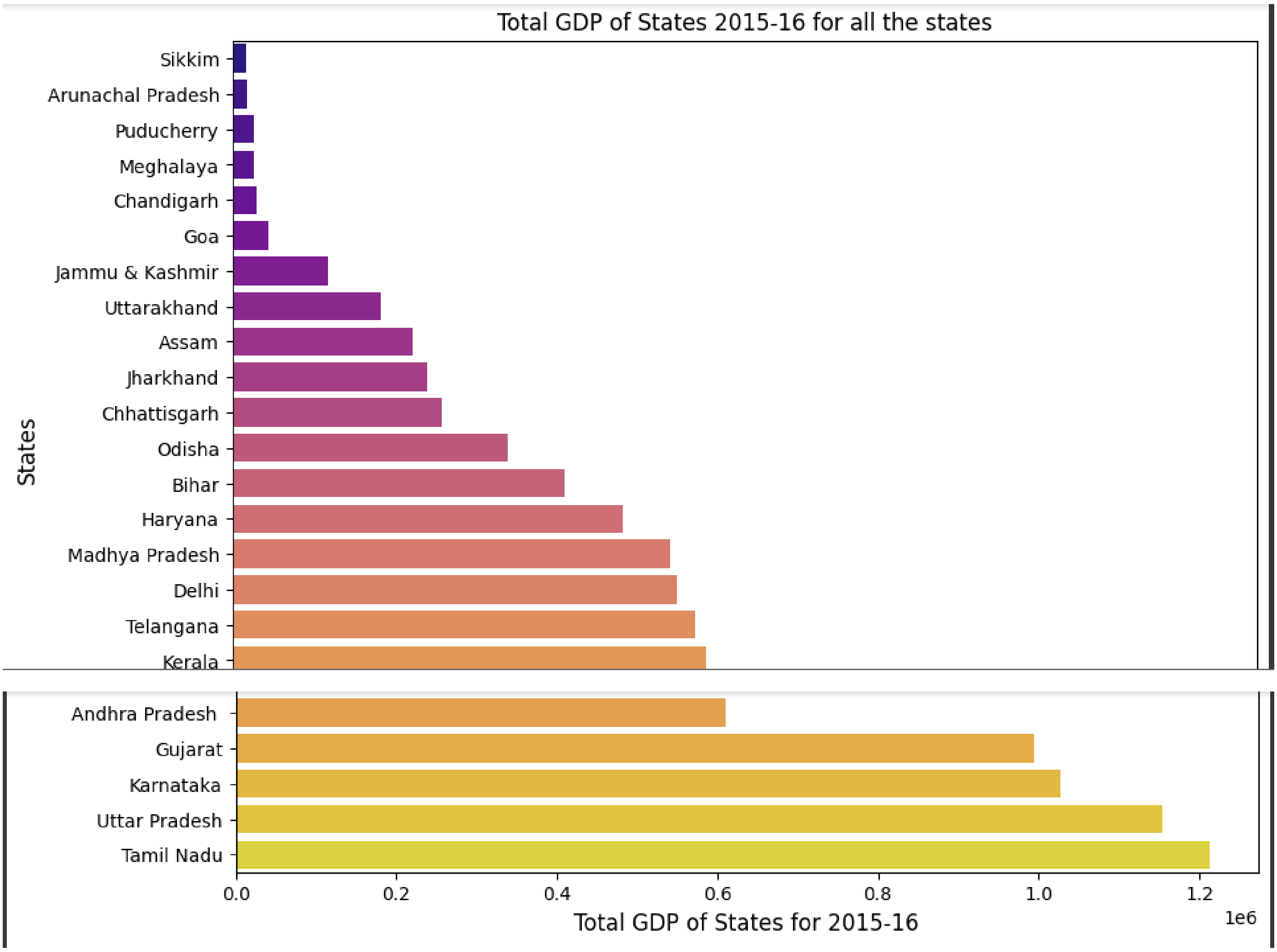


```
[ ]
# carrying out necessary transformation to make the data ready for plotting
total_GDP_15_16_states = total_GDP_15_16[total_GDP_15_16.columns[2:34]].transpose()
total_GDP_15_16_states = total_GDP_15_16_states.rename(columns={4: 'Total GDP of States 2015-16'})
total_GDP_15_16_states = total_GDP_15_16_states.dropna()
total_GDP_15_16_states = total_GDP_15_16_states.sort_values('Total GDP of States 2015-16',ascending=True)
total_GDP_15_16_states
```

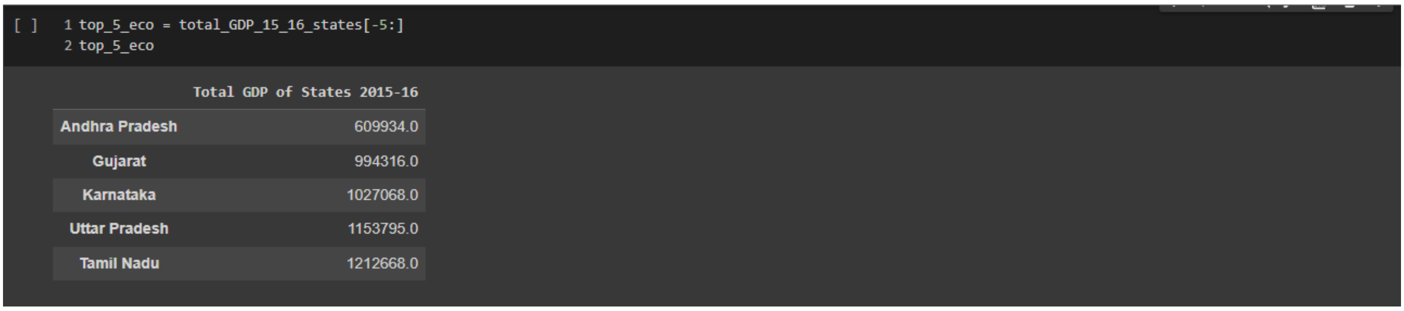
	Total GDP of States 2015-16
Sikkim	16637.0
Arunachal Pradesh	18784.0
Puducherry	26533.0
Meghalaya	26745.0
Chandigarh	30304.0
Goa	45002.0
Jammu & Kashmir	118387.0
Uttarakhand	184091.0
Assam	224234.0
Jharkhand	241955.0



```
[ ]
plt.figure(figsize=(10,8), dpi = 100)
sns.barplot(x = total_GDP_15_16_states['Total GDP of States 2015-16'], y = total_GDP_15_16_states.index,palette='plasma')
plt.xlabel('Total GDP of States for 2015-16', fontsize=12)
plt.ylabel('States', fontsize=12)
plt.title('Total GDP of States 2015-16 for all the states',fontsize=12)
plt.show()
```



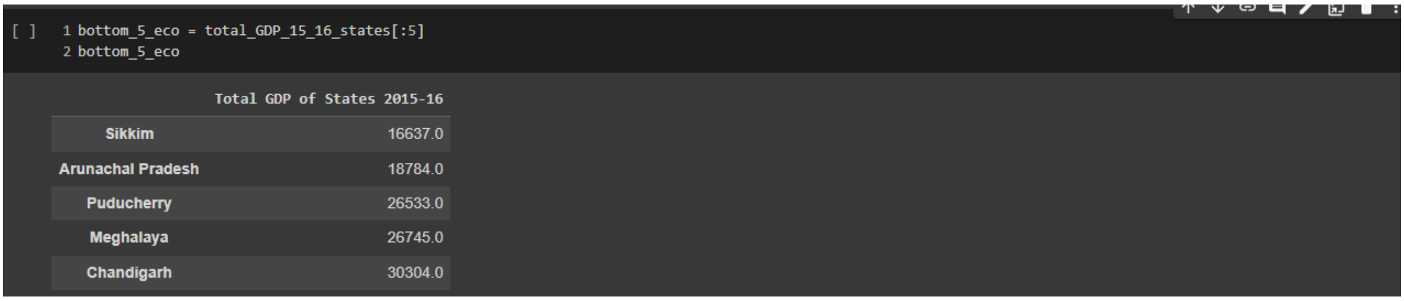
Top 5 states in terms of total GDP for the year 2015-16



```
top_5_eco = total_GDP_15_16_states[-5:]
top_5_eco
```

	Total GDP of States 2015-16
Andhra Pradesh	609934.0
Gujarat	994316.0
Karnataka	1027068.0
Uttar Pradesh	1153795.0
Tamil Nadu	1212668.0

Bottom 5 states in terms of total GDP for the year 2015-16



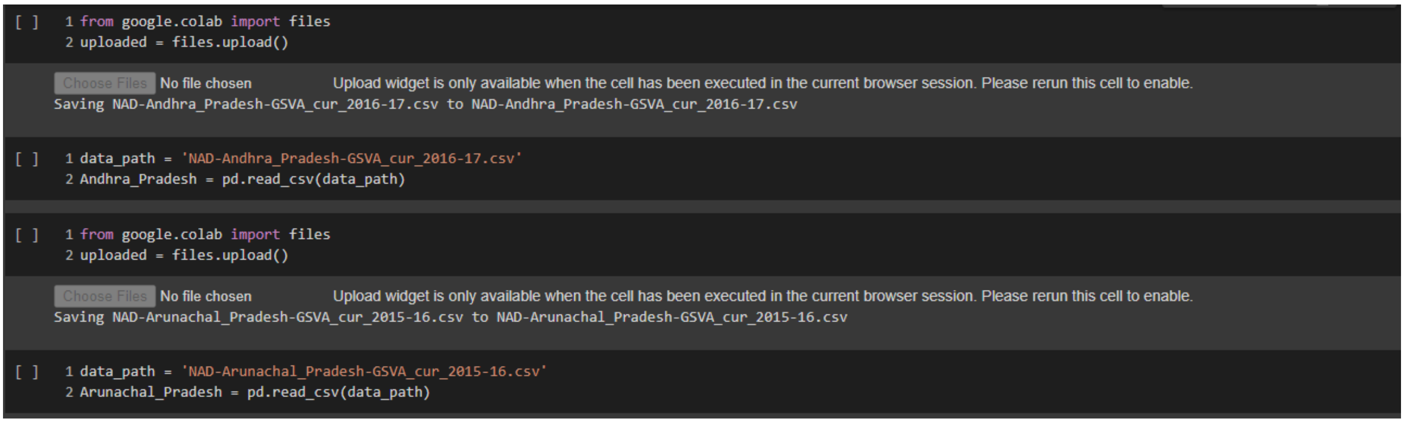
```
bottom_5_eco = total_GDP_15_16_states[:5]
bottom_5_eco
```

	Total GDP of States 2015-16
Sikkim	16637.0
Arunachal Pradesh	18784.0
Puducherry	26533.0
Meghalaya	26745.0
Chandigarh	30304.0

4. Data Mining Techniques Applied –
   - Data Visualization - It is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.
   - Exploratory Data Analysis - It is an approach to analysing datasets to summarize their main characteristics, often with visual methods. EDA is used for seeing what the data can tell us before the modelling task.
   - Standard deviation – We will compare the average growth rate for the year 2013-14, 2014-15, 2015-16 and the standard deviation.
   - Data cleaning - It is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

4.1Codes –

Reading the CSV files for all the states



```
from google.colab import files
uploaded = files.upload()
```

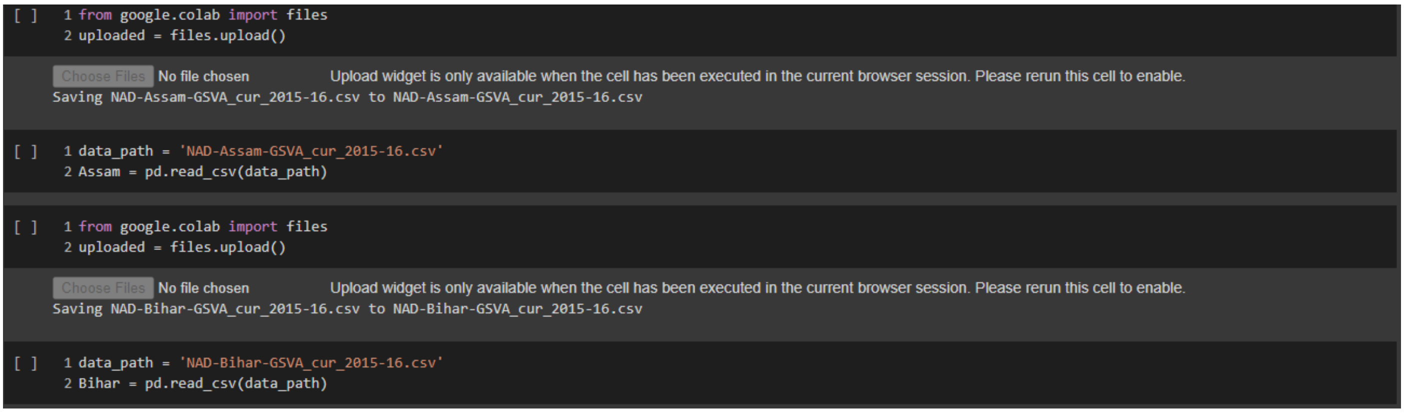
```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Andhra_Pradesh-GSVA_cur_2016-17.csv to NAD-Andhra_Pradesh-GSVA_cur_2016-17.csv
```

```
data_path = 'NAD-Andhra_Pradesh-GSVA_cur_2016-17.csv'
Andhra_Pradesh = pd.read_csv(data_path)
```

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Arunachal_Pradesh-GSVA_cur_2015-16.csv to NAD-Arunachal_Pradesh-GSVA_cur_2015-16.csv
```

```
data_path = 'NAD-Arunachal_Pradesh-GSVA_cur_2015-16.csv'
Arunachal_Pradesh = pd.read_csv(data_path)
```



```
[ ]  from google.colab import files
     uploaded = files.upload()
```

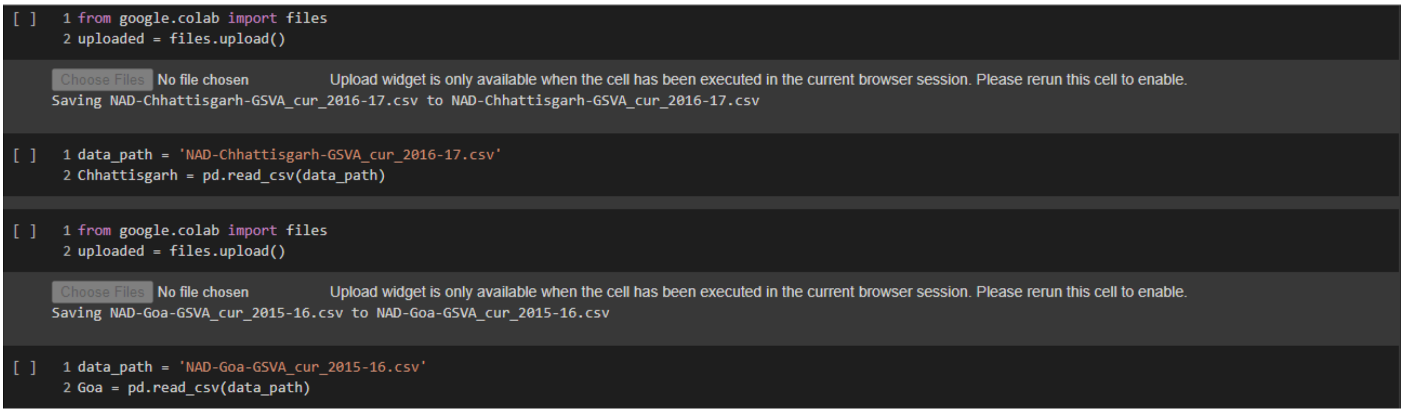
```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Assam-GSVA_cur_2015-16.csv to NAD-Assam-GSVA_cur_2015-16.csv
```

```
[ ]  data_path = 'NAD-Assam-GSVA_cur_2015-16.csv'
     Assam = pd.read_csv(data_path)
```

```
[ ]  from google.colab import files
     uploaded = files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Bihar-GSVA_cur_2015-16.csv to NAD-Bihar-GSVA_cur_2015-16.csv
```

```
[ ]  data_path = 'NAD-Bihar-GSVA_cur_2015-16.csv'
     Bihar = pd.read_csv(data_path)
```



```
[ ]  from google.colab import files
     uploaded = files.upload()
```

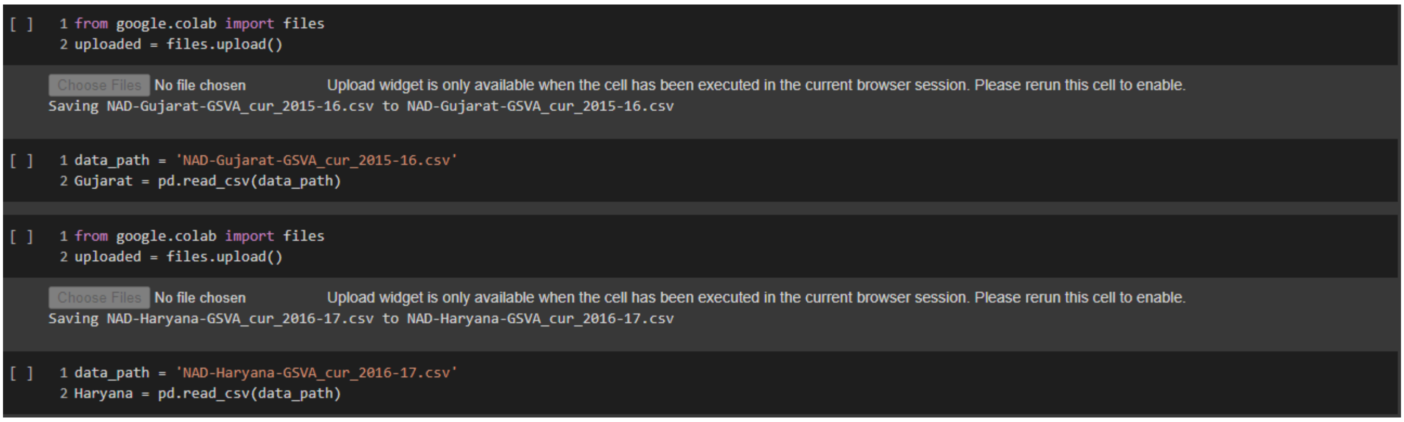
```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Chhattisgarh-GSVA_cur_2016-17.csv to NAD-Chhattisgarh-GSVA_cur_2016-17.csv
```

```
[ ]  data_path = 'NAD-Chhattisgarh-GSVA_cur_2016-17.csv'
     Chhattisgarh = pd.read_csv(data_path)
```

```
[ ]  from google.colab import files
     uploaded = files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Goa-GSVA_cur_2015-16.csv to NAD-Goa-GSVA_cur_2015-16.csv
```

```
[ ]  data_path = 'NAD-Goa-GSVA_cur_2015-16.csv'
     Goa = pd.read_csv(data_path)
```



```
[ ]  from google.colab import files
     uploaded = files.upload()
```

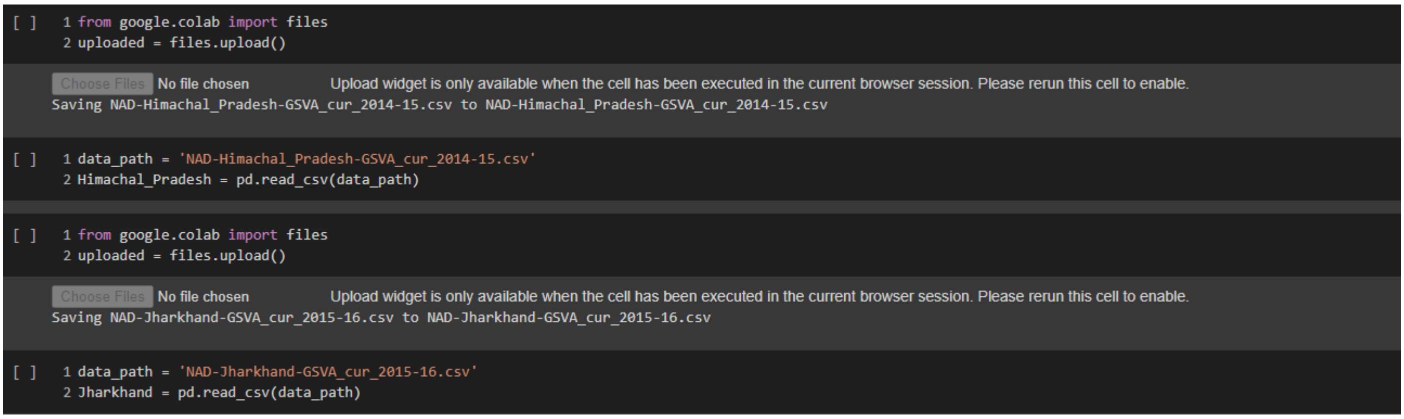
```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Gujarat-GSVA_cur_2015-16.csv to NAD-Gujarat-GSVA_cur_2015-16.csv
```

```
[ ]  data_path = 'NAD-Gujarat-GSVA_cur_2015-16.csv'
     Gujarat = pd.read_csv(data_path)
```

```
[ ]  from google.colab import files
     uploaded = files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Haryana-GSVA_cur_2016-17.csv to NAD-Haryana-GSVA_cur_2016-17.csv
```

```
[ ]  data_path = 'NAD-Haryana-GSVA_cur_2016-17.csv'
     Haryana = pd.read_csv(data_path)
```



```
[ ]  from google.colab import files
     uploaded = files.upload()
```

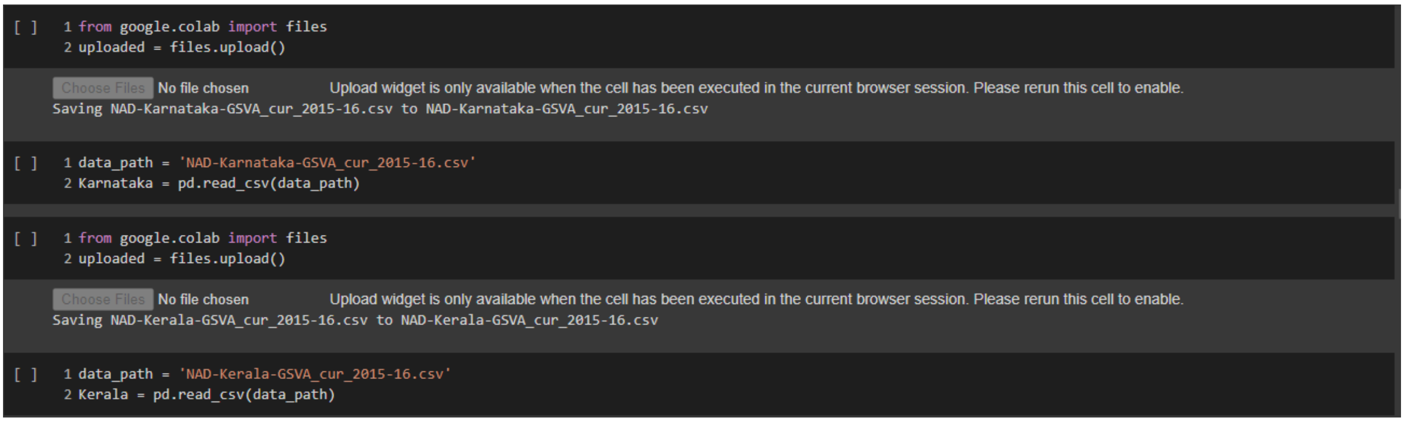
```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Himachal_Pradesh-GSVA_cur_2014-15.csv to NAD-Himachal_Pradesh-GSVA_cur_2014-15.csv
```

```
[ ]  data_path = 'NAD-Himachal_Pradesh-GSVA_cur_2014-15.csv'
     Himachal_Pradesh = pd.read_csv(data_path)
```

```
[ ]  from google.colab import files
     uploaded = files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Jharkhand-GSVA_cur_2015-16.csv to NAD-Jharkhand-GSVA_cur_2015-16.csv
```

```
[ ]  data_path = 'NAD-Jharkhand-GSVA_cur_2015-16.csv'
     Jharkhand = pd.read_csv(data_path)
```



```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Karnataka-GSVA_cur_2015-16.csv to NAD-Karnataka-GSVA_cur_2015-16.csv
```

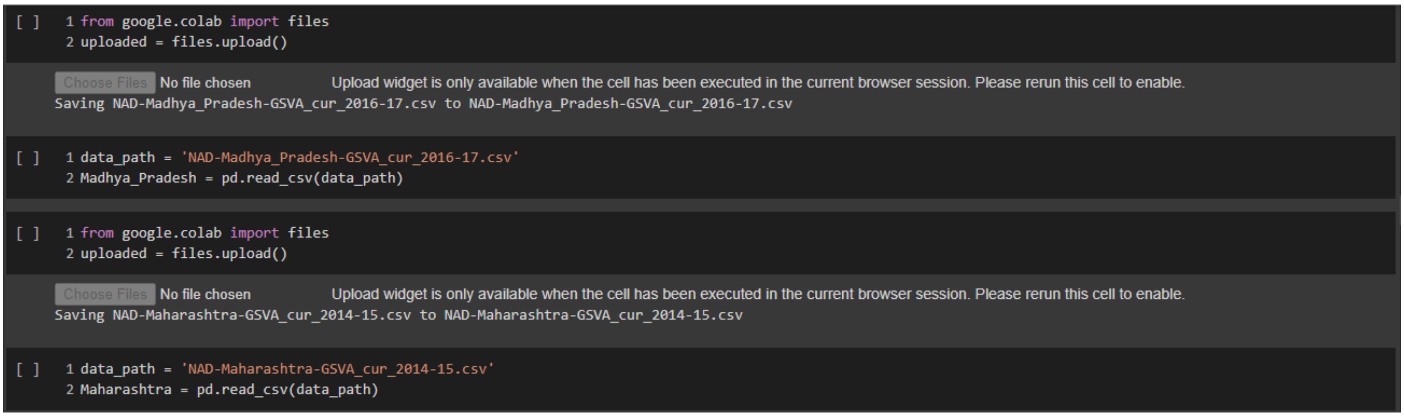
```
data_path = 'NAD-Karnataka-GSVA_cur_2015-16.csv'
Karnataka = pd.read_csv(data_path)
```

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Kerala-GSVA_cur_2015-16.csv to NAD-Kerala-GSVA_cur_2015-16.csv
```

```
data_path = 'NAD-Kerala-GSVA_cur_2015-16.csv'
Kerala = pd.read_csv(data_path)
```



```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Madhya_Pradesh-GSVA_cur_2016-17.csv to NAD-Madhya_Pradesh-GSVA_cur_2016-17.csv
```

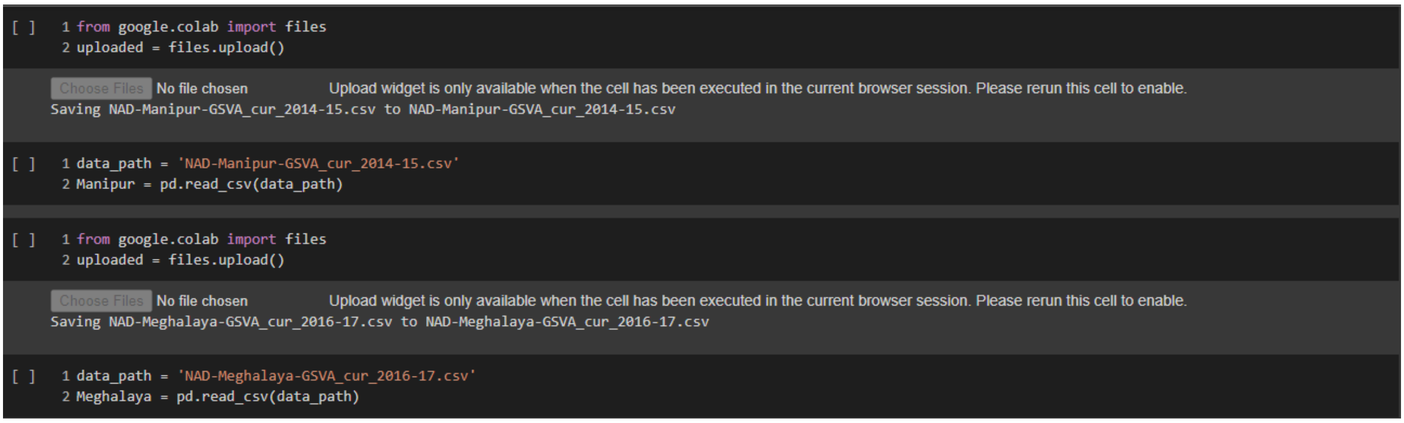
```
data_path = 'NAD-Madhya_Pradesh-GSVA_cur_2016-17.csv'
Madhya_Pradesh = pd.read_csv(data_path)
```

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Maharashtra-GSVA_cur_2014-15.csv to NAD-Maharashtra-GSVA_cur_2014-15.csv
```

```
data_path = 'NAD-Maharashtra-GSVA_cur_2014-15.csv'
Maharashtra = pd.read_csv(data_path)
```



```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Manipur-GSVA_cur_2014-15.csv to NAD-Manipur-GSVA_cur_2014-15.csv
```

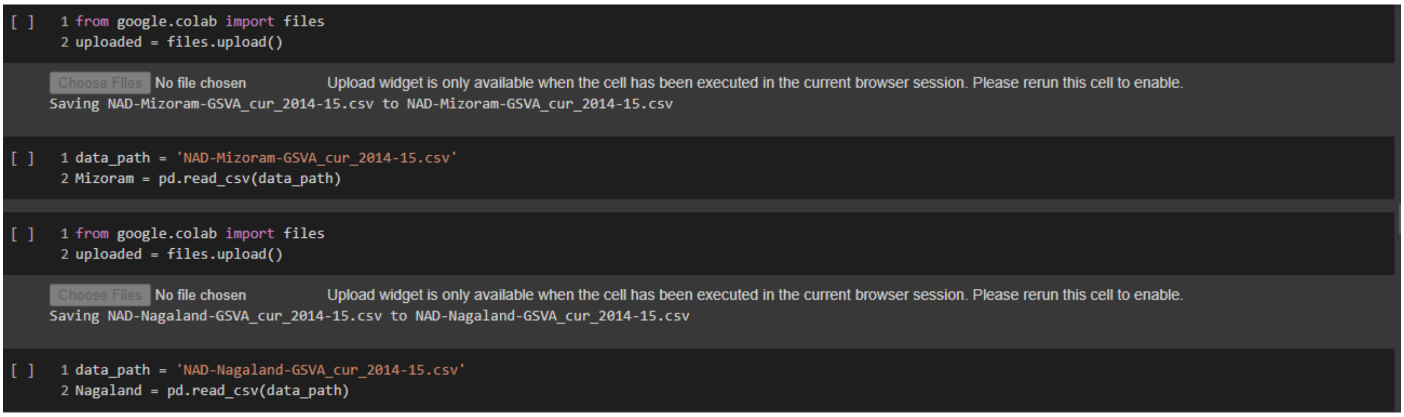
```
data_path = 'NAD-Manipur-GSVA_cur_2014-15.csv'
Manipur = pd.read_csv(data_path)
```

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Meghalaya-GSVA_cur_2016-17.csv to NAD-Meghalaya-GSVA_cur_2016-17.csv
```

```
data_path = 'NAD-Meghalaya-GSVA_cur_2016-17.csv'
Meghalaya = pd.read_csv(data_path)
```



```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Mizoram-GSVA_cur_2014-15.csv to NAD-Mizoram-GSVA_cur_2014-15.csv
```

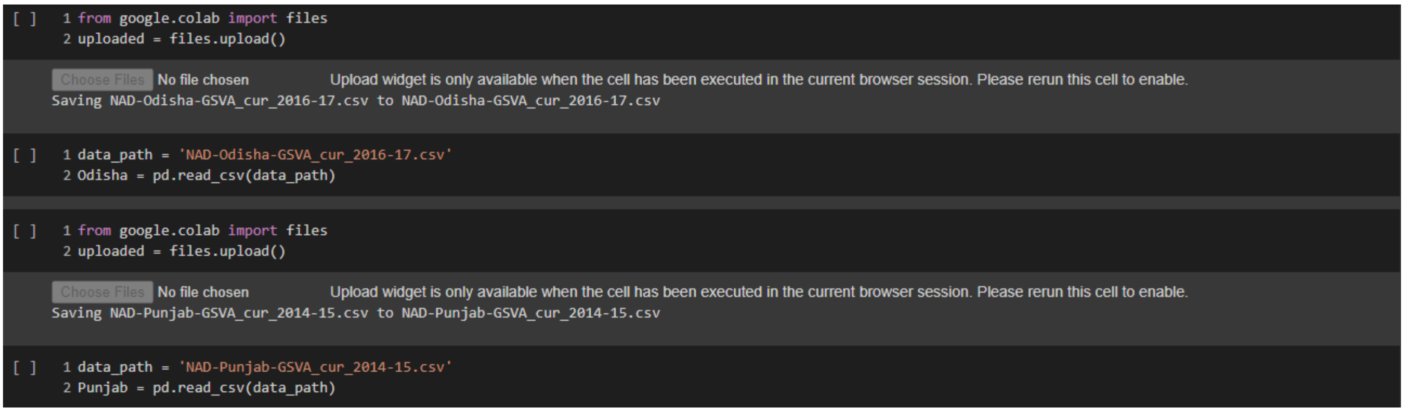
```
data_path = 'NAD-Mizoram-GSVA_cur_2014-15.csv'
Mizoram = pd.read_csv(data_path)
```

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving NAD-Nagaland-GSVA_cur_2014-15.csv to NAD-Nagaland-GSVA_cur_2014-15.csv
```

```
data_path = 'NAD-Nagaland-GSVA_cur_2014-15.csv'
Nagaland = pd.read_csv(data_path)
```



```python
from google.colab import files
uploaded = files.upload()
```

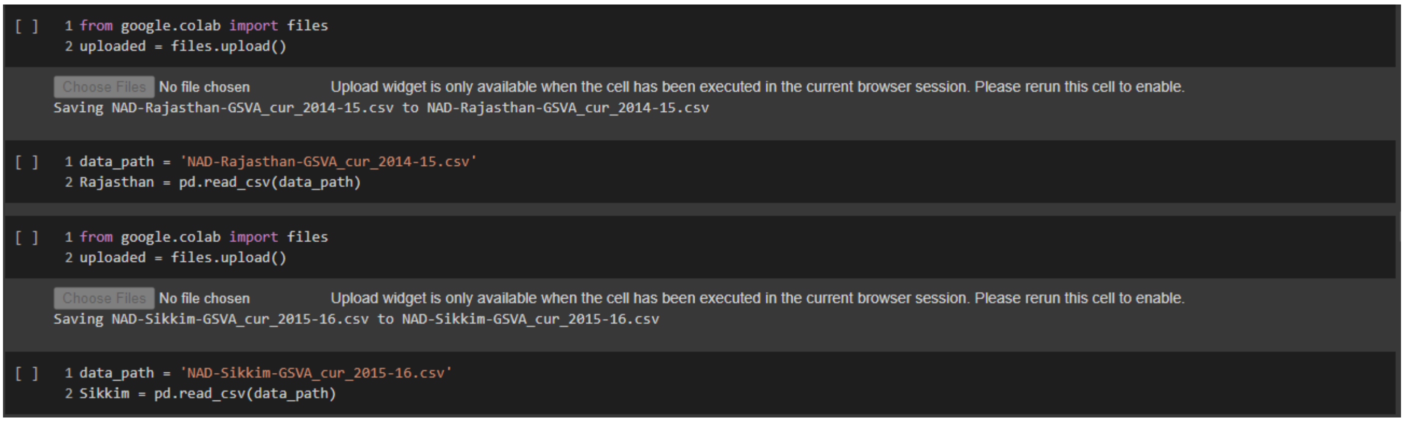
```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Odisha-GSVA_cur_2016-17.csv to NAD-Odisha-GSVA_cur_2016-17.csv
```

```python
data_path = 'NAD-Odisha-GSVA_cur_2016-17.csv'
Odisha = pd.read_csv(data_path)
```

```python
from google.colab import files
uploaded = files.upload()
```

```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Punjab-GSVA_cur_2014-15.csv to NAD-Punjab-GSVA_cur_2014-15.csv
```

```python
data_path = 'NAD-Punjab-GSVA_cur_2014-15.csv'
Punjab = pd.read_csv(data_path)
```



```python
from google.colab import files
uploaded = files.upload()
```

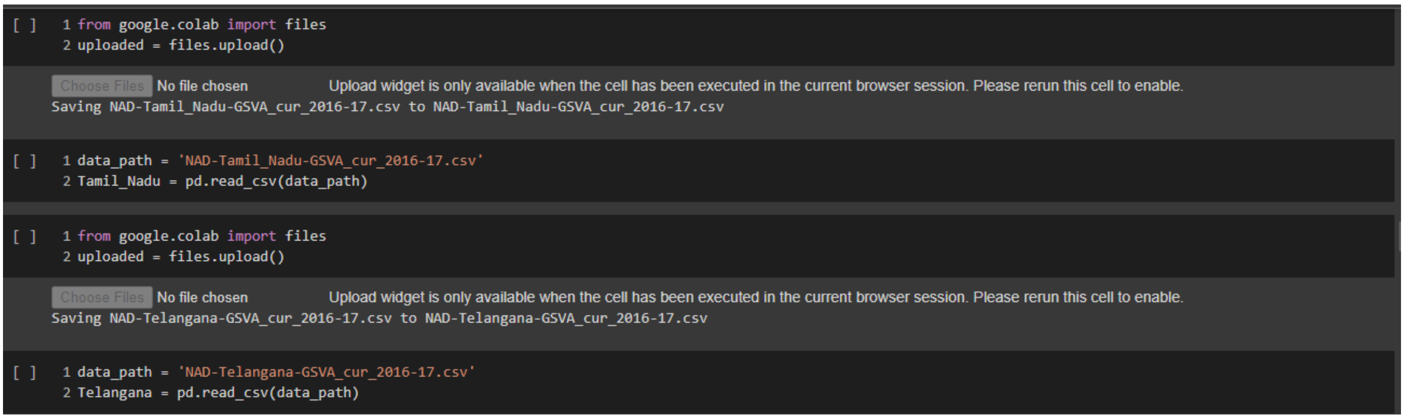
```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Rajasthan-GSVA_cur_2014-15.csv to NAD-Rajasthan-GSVA_cur_2014-15.csv
```

```python
data_path = 'NAD-Rajasthan-GSVA_cur_2014-15.csv'
Rajasthan = pd.read_csv(data_path)
```

```python
from google.colab import files
uploaded = files.upload()
```

```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Sikkim-GSVA_cur_2015-16.csv to NAD-Sikkim-GSVA_cur_2015-16.csv
```

```python
data_path = 'NAD-Sikkim-GSVA_cur_2015-16.csv'
Sikkim = pd.read_csv(data_path)
```



```python
from google.colab import files
uploaded = files.upload()
```

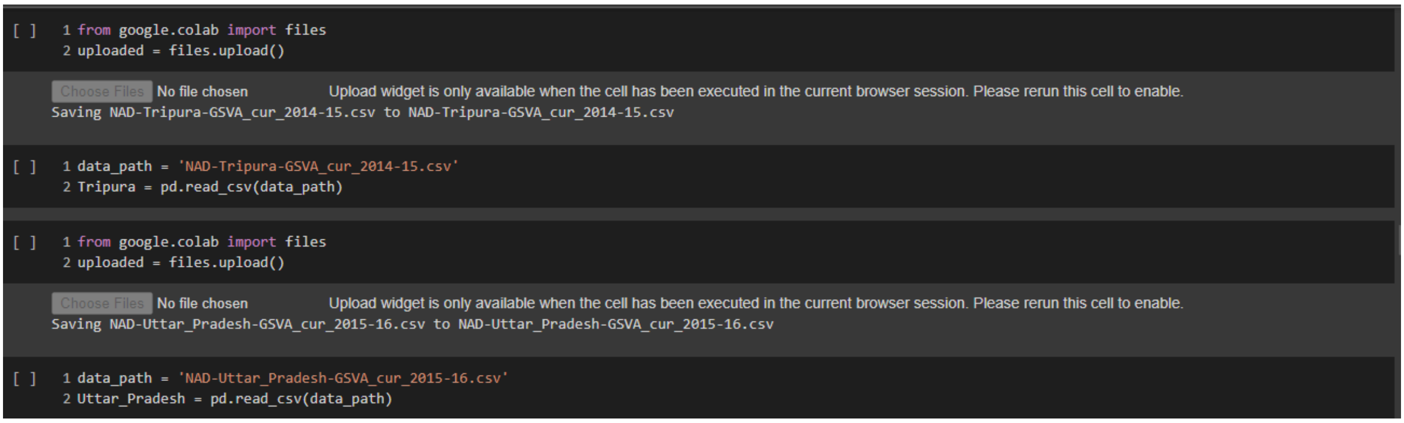
```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Tamil_Nadu-GSVA_cur_2016-17.csv to NAD-Tamil_Nadu-GSVA_cur_2016-17.csv
```

```python
data_path = 'NAD-Tamil_Nadu-GSVA_cur_2016-17.csv'
Tamil_Nadu = pd.read_csv(data_path)
```

```python
from google.colab import files
uploaded = files.upload()
```

```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Telangana-GSVA_cur_2016-17.csv to NAD-Telangana-GSVA_cur_2016-17.csv
```

```python
data_path = 'NAD-Telangana-GSVA_cur_2016-17.csv'
Telangana = pd.read_csv(data_path)
```



```python
from google.colab import files
uploaded = files.upload()
```

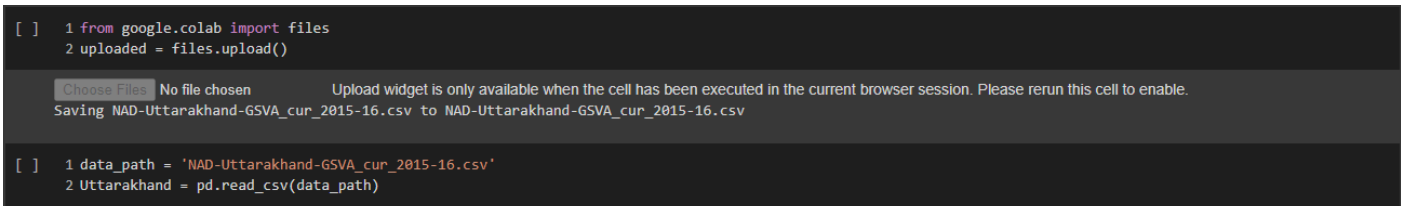
```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Tripura-GSVA_cur_2014-15.csv to NAD-Tripura-GSVA_cur_2014-15.csv
```

```python
data_path = 'NAD-Tripura-GSVA_cur_2014-15.csv'
Tripura = pd.read_csv(data_path)
```

```python
from google.colab import files
uploaded = files.upload()
```

```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Uttar_Pradesh-GSVA_cur_2015-16.csv to NAD-Uttar_Pradesh-GSVA_cur_2015-16.csv
```

```python
data_path = 'NAD-Uttar_Pradesh-GSVA_cur_2015-16.csv'
Uttar_Pradesh = pd.read_csv(data_path)
```

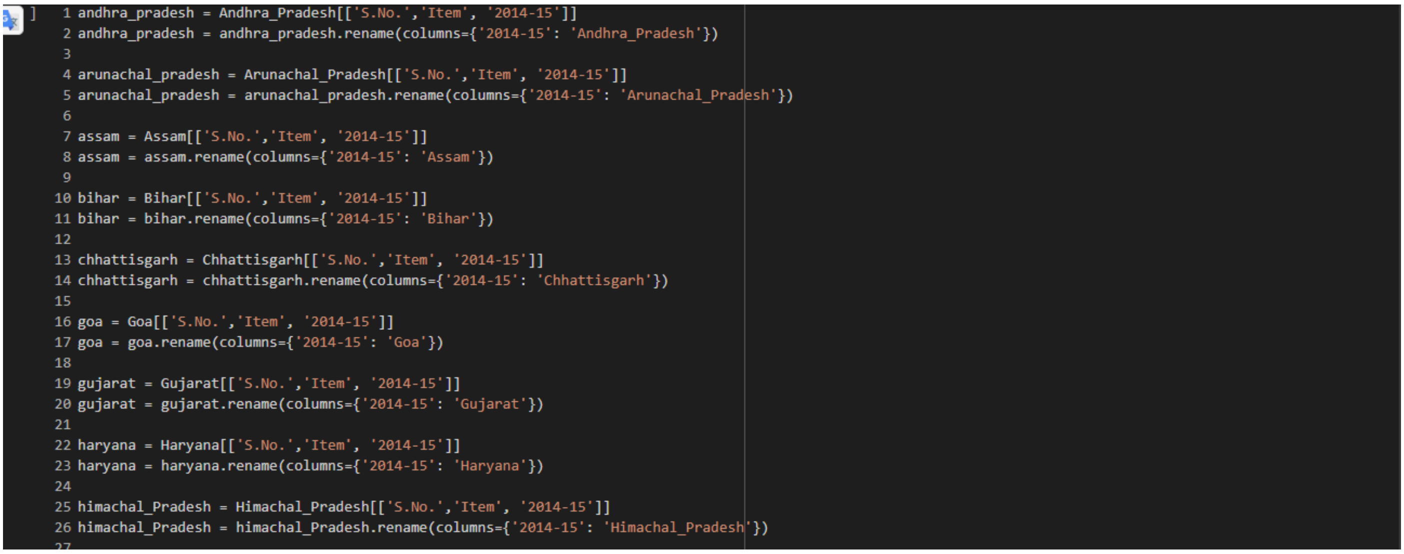


```python
from google.colab import files
uploaded = files.upload()
```

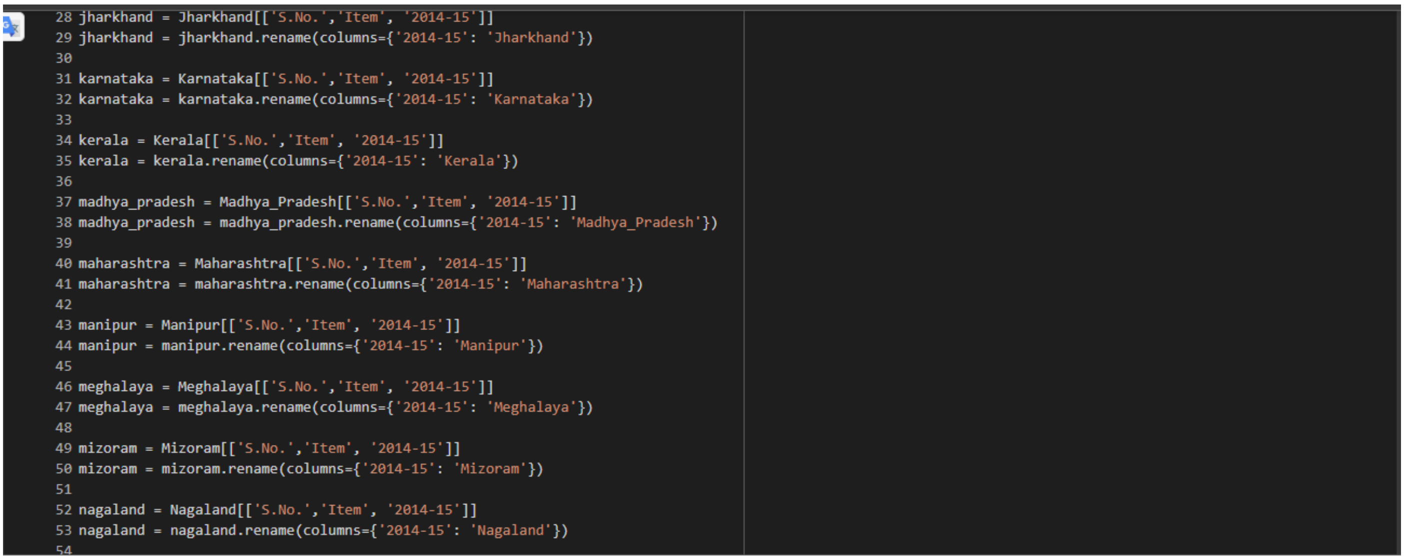
```
No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving NAD-Uttarakhand-GSVA_cur_2015-16.csv to NAD-Uttarakhand-GSVA_cur_2015-16.csv
```

```python
data_path = 'NAD-Uttarakhand-GSVA_cur_2015-16.csv'
Uttarakhand = pd.read_csv(data_path)
```

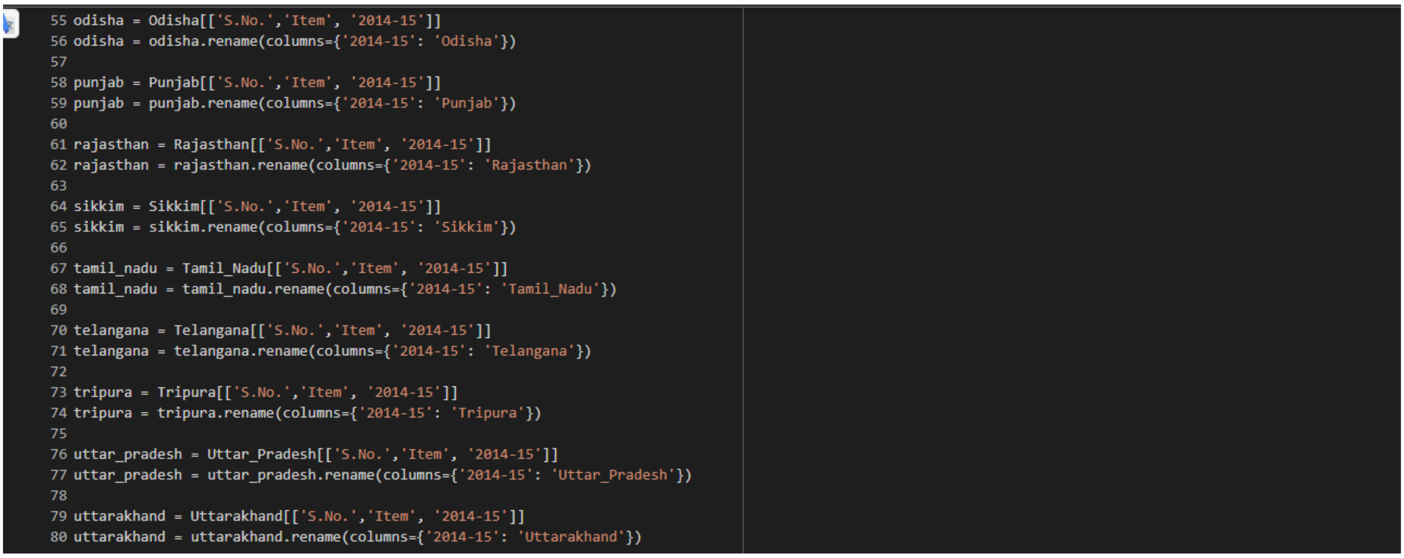
Taking data only for year 2014-15



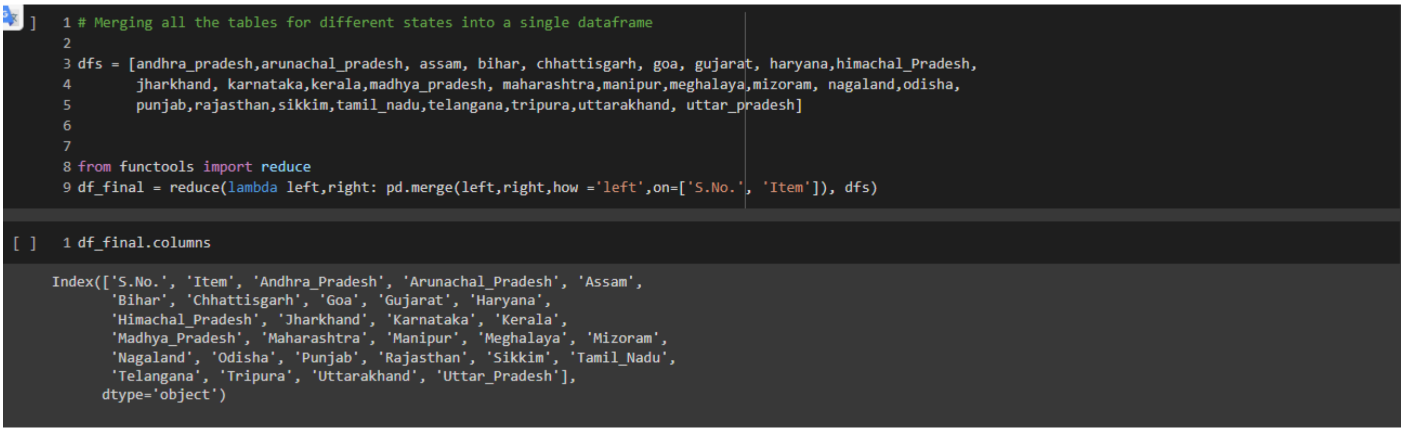
```
andhra_pradesh = Andhra_Pradesh[['S.No.','Item', '2014-15']]
andhra_pradesh = andhra_pradesh.rename(columns={'2014-15': 'Andhra_Pradesh'})

arunachal_pradesh = Arunachal_Pradesh[['S.No.','Item', '2014-15']]
arunachal_pradesh = arunachal_pradesh.rename(columns={'2014-15': 'Arunachal_Pradesh'})

assam = Assam[['S.No.','Item', '2014-15']]
assam = assam.rename(columns={'2014-15': 'Assam'})

bihar = Bihar[['S.No.','Item', '2014-15']]
bihar = bihar.rename(columns={'2014-15': 'Bihar'})

chhattisgarh = Chhattisgarh[['S.No.','Item', '2014-15']]
chhattisgarh = chhattisgarh.rename(columns={'2014-15': 'Chhattisgarh'})

goa = Goa[['S.No.','Item', '2014-15']]
goa = goa.rename(columns={'2014-15': 'Goa'})

gujarat = Gujarat[['S.No.','Item', '2014-15']]
gujarat = gujarat.rename(columns={'2014-15': 'Gujarat'})

haryana = Haryana[['S.No.','Item', '2014-15']]
haryana = haryana.rename(columns={'2014-15': 'Haryana'})

himachal_Pradesh = Himachal_Pradesh[['S.No.','Item', '2014-15']]
himachal_Pradesh = himachal_Pradesh.rename(columns={'2014-15': 'Himachal_Pradesh'})

```



```
jharkhand = Jharkhand[['S.No.','Item', '2014-15']]
jharkhand = jharkhand.rename(columns={'2014-15': 'Jharkhand'})

karnataka = Karnataka[['S.No.','Item', '2014-15']]
karnataka = karnataka.rename(columns={'2014-15': 'Karnataka'})

kerala = Kerala[['S.No.','Item', '2014-15']]
kerala = kerala.rename(columns={'2014-15': 'Kerala'})

madhya_pradesh = Madhya_Pradesh[['S.No.','Item', '2014-15']]
madhya_pradesh = madhya_pradesh.rename(columns={'2014-15': 'Madhya_Pradesh'})

maharashtra = Maharashtra[['S.No.','Item', '2014-15']]
maharashtra = maharashtra.rename(columns={'2014-15': 'Maharashtra'})

manipur = Manipur[['S.No.','Item', '2014-15']]
manipur = manipur.rename(columns={'2014-15': 'Manipur'})

meghalaya = Meghalaya[['S.No.','Item', '2014-15']]
meghalaya = meghalaya.rename(columns={'2014-15': 'Meghalaya'})

mizoram = Mizoram[['S.No.','Item', '2014-15']]
mizoram = mizoram.rename(columns={'2014-15': 'Mizoram'})

nagaland = Nagaland[['S.No.','Item', '2014-15']]
nagaland = nagaland.rename(columns={'2014-15': 'Nagaland'})

```



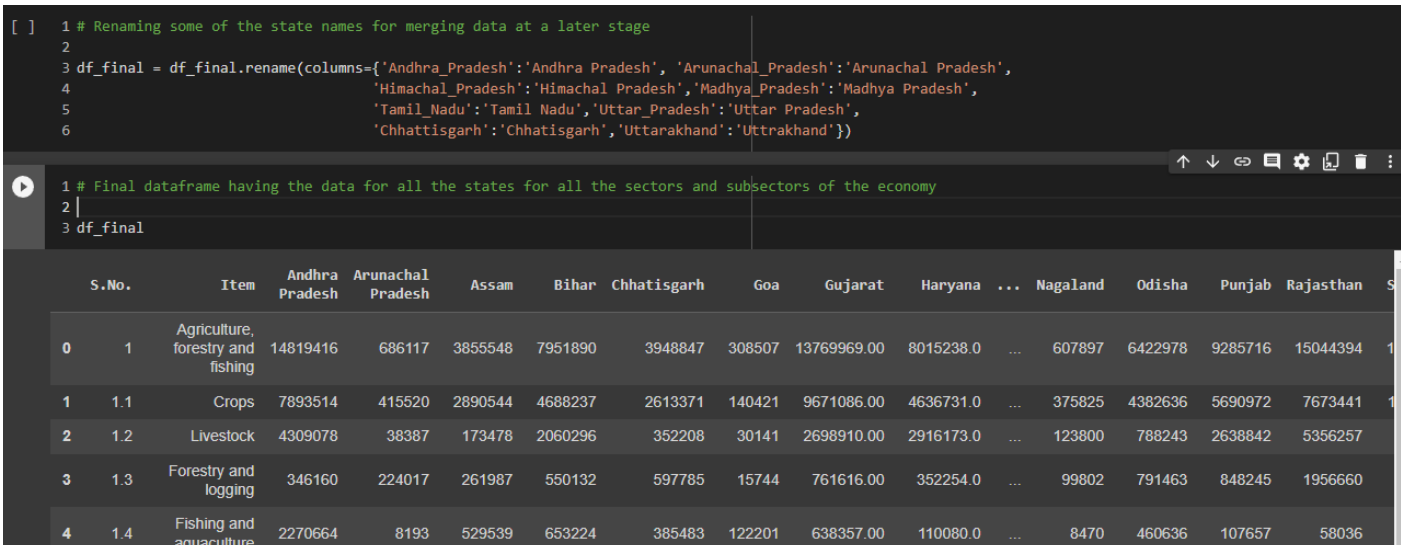
```
odisha = Odisha[['S.No.','Item', '2014-15']]
odisha = odisha.rename(columns={'2014-15': 'Odisha'})

punjab = Punjab[['S.No.','Item', '2014-15']]
punjab = punjab.rename(columns={'2014-15': 'Punjab'})

rajasthan = Rajasthan[['S.No.','Item', '2014-15']]
rajasthan = rajasthan.rename(columns={'2014-15': 'Rajasthan'})

sikkim = Sikkim[['S.No.','Item', '2014-15']]
sikkim = sikkim.rename(columns={'2014-15': 'Sikkim'})

tamil_nadu = Tamil_Nadu[['S.No.','Item', '2014-15']]
tamil_nadu = tamil_nadu.rename(columns={'2014-15': 'Tamil_Nadu'})

telangana = Telangana[['S.No.','Item', '2014-15']]
telangana = telangana.rename(columns={'2014-15': 'Telangana'})

tripura = Tripura[['S.No.','Item', '2014-15']]
tripura = tripura.rename(columns={'2014-15': 'Tripura'})

uttar_pradesh = Uttar_Pradesh[['S.No.','Item', '2014-15']]
uttar_pradesh = uttar_pradesh.rename(columns={'2014-15': 'Uttar_Pradesh'})

uttarakhand = Uttarakhand[['S.No.','Item', '2014-15']]
uttarakhand = uttarakhand.rename(columns={'2014-15': 'Uttarakhand'})
```



```python
# Merging all the tables for different states into a single dataframe

dfs = [andhra_pradesh,arunachal_pradesh, assam, bihar, chhattisgarh, goa, gujarat, haryana,himachal_Pradesh,
       jharkhand, karnataka,kerala,madhya_pradesh, maharashtra,manipur,meghalaya,mizoram, nagaland,odisha,
       punjab,rajasthan,sikkim,tamil_nadu,telangana,tripura,uttarakhand, uttar_pradesh]


from functools import reduce
df_final = reduce(lambda left,right: pd.merge(left,right,how ='left',on=['S.No.', 'Item']), dfs)
```

```python
df_final.columns
```

```
Index(['S.No.', 'Item', 'Andhra_Pradesh', 'Arunachal_Pradesh', 'Assam',
       'Bihar', 'Chhattisgarh', 'Goa', 'Gujarat', 'Haryana',
       'Himachal_Pradesh', 'Jharkhand', 'Karnataka', 'Kerala',
       'Madhya_Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',
       'Nagaland', 'Odisha', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil_Nadu',
       'Telangana', 'Tripura', 'Uttarakhand', 'Uttar_Pradesh'],
      dtype='object')
```

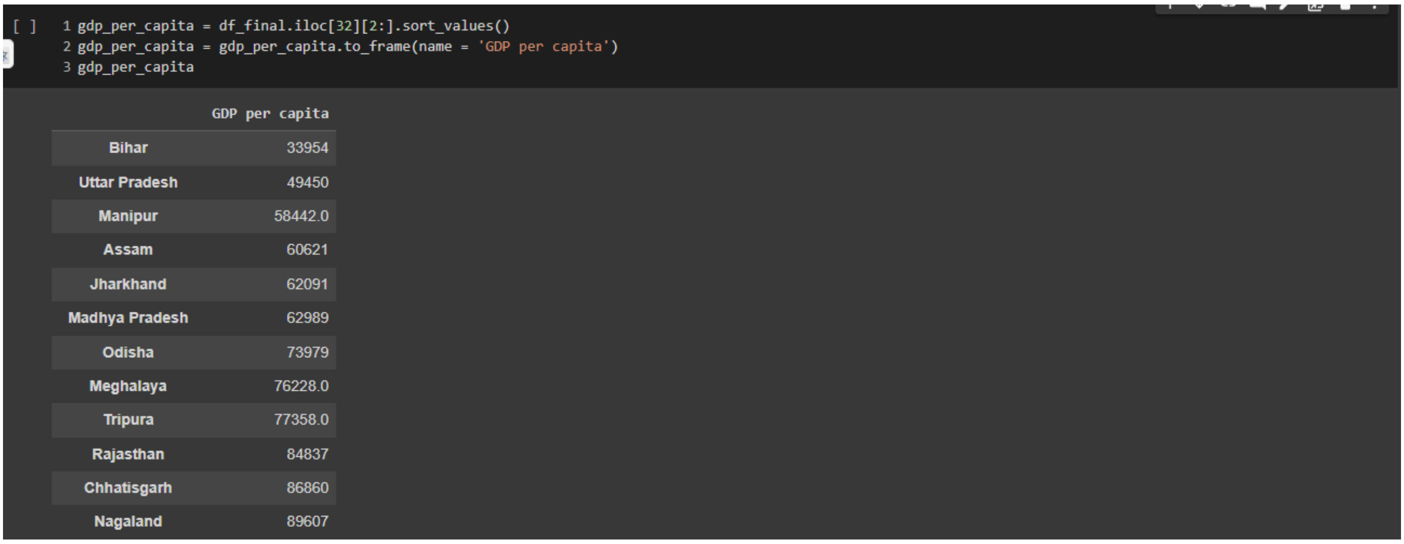


```python
# Renaming some of the state names for merging data at a later stage

df_final = df_final.rename(columns={'Andhra_Pradesh':'Andhra Pradesh', 'Arunachal_Pradesh':'Arunachal Pradesh',
                                    'Himachal_Pradesh':'Himachal Pradesh','Madhya_Pradesh':'Madhya Pradesh',
                                    'Tamil_Nadu':'Tamil Nadu','Uttar_Pradesh':'Uttar Pradesh',
                                    'Chhattisgarh':'Chhatisgarh','Uttarakhand':'Uttrakhand'})
```

```python
# Final dataframe having the data for all the states for all the sectors and subsectors of the economy

df_final
```

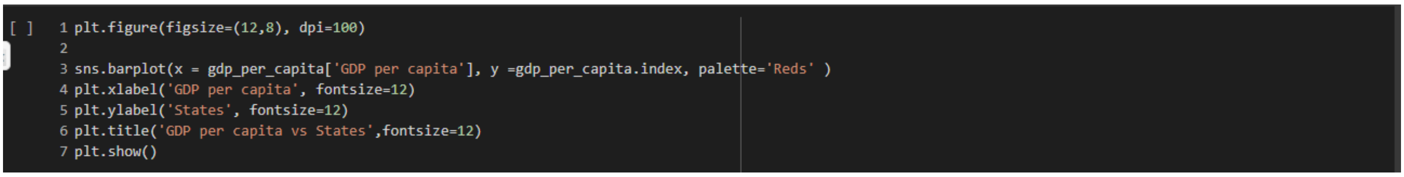
	S.No.	Item	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhatisgarh	Goa	Gujarat	Haryana	...	Nagaland	Odisha	Punjab	Rajasthan	S
0	1	Agriculture, forestry and fishing	14819416	686117	3855548	7951890	3948847	308507	13769969.00	8015238.0	...	607897	6422978	9285716	15044394	1
1	1.1	Crops	7893514	415520	2890544	4688237	2613371	140421	9671086.00	4636731.0	...	375825	4382636	5690972	7673441	1
2	1.2	Livestock	4309078	38387	173478	2060296	352208	30141	2698910.00	2916173.0	...	123800	788243	2638842	5356257	
3	1.3	Forestry and logging	346160	224017	261987	550132	597785	15744	761616.00	352254.0	...	99802	791463	848245	1956660	
4	1.4	Fishing and aquaculture	2270664	8193	529539	653224	385483	122201	638357.00	110080.0	...	8470	460636	107657	58036	

Creating the GDP per capita Data Frame

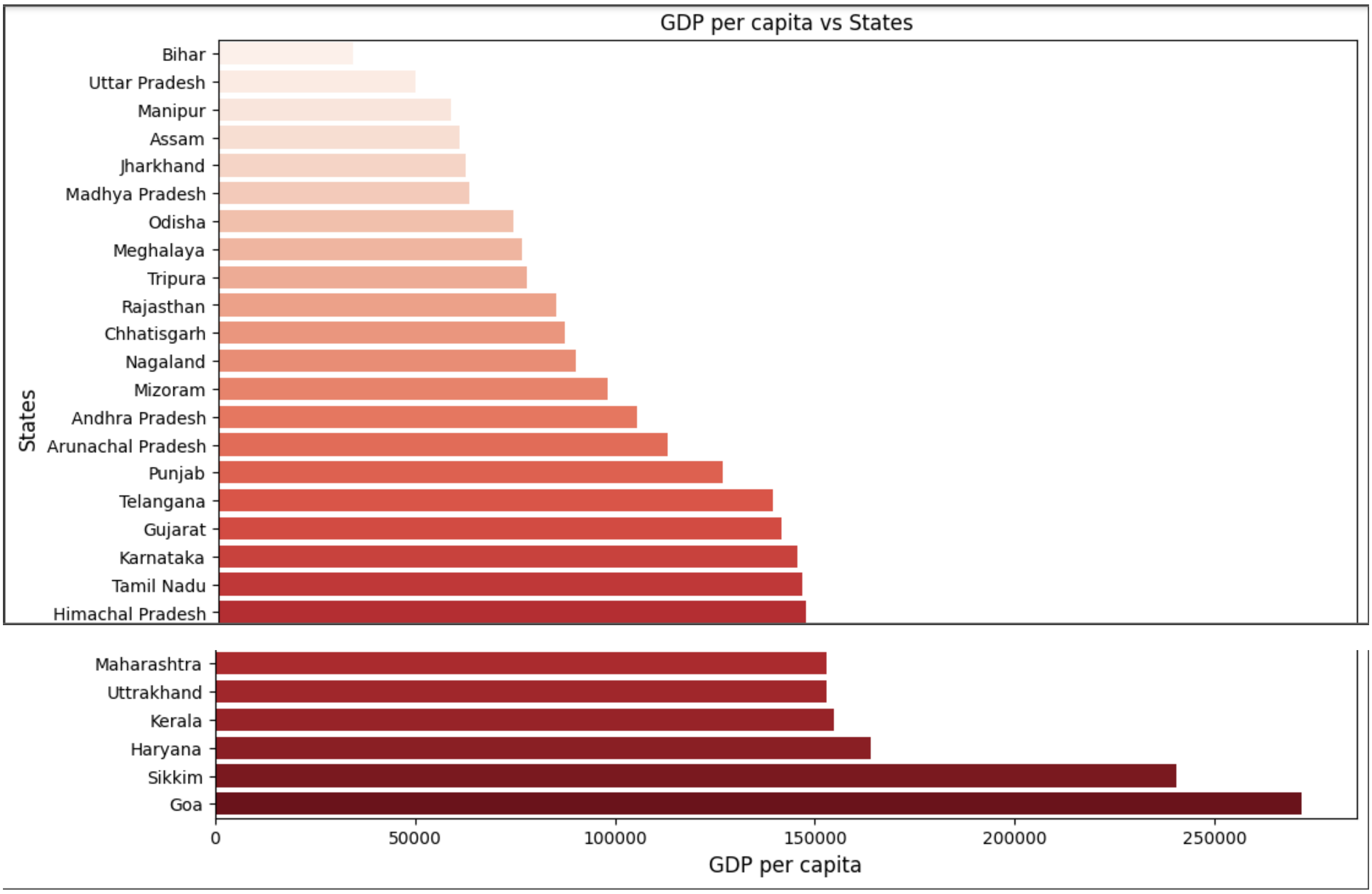


```python
gdp_per_capita = df_final.iloc[32][2:].sort_values()
gdp_per_capita = gdp_per_capita.to_frame(name = 'GDP per capita')
gdp_per_capita
```

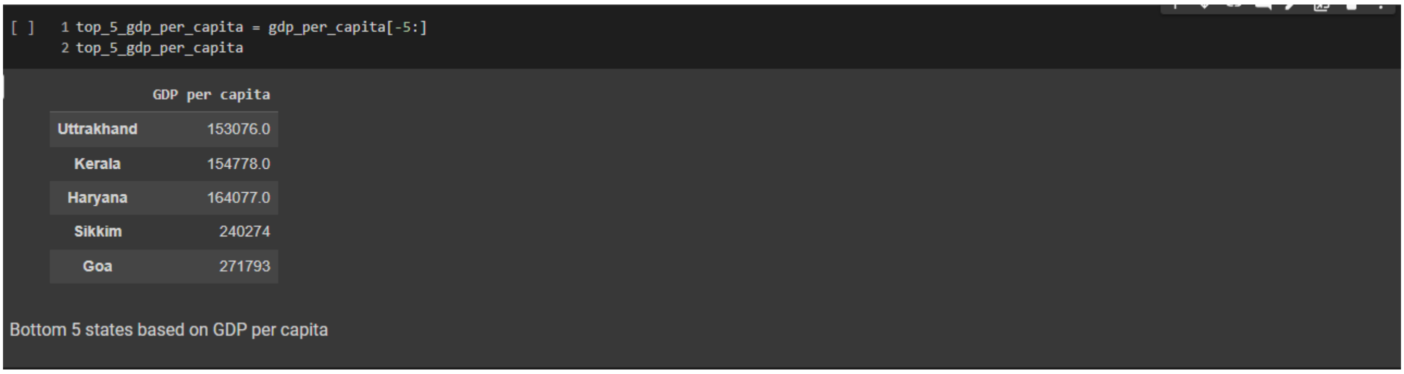
	GDP per capita
Bihar	33954
Uttar Pradesh	49450
Manipur	58442.0
Assam	60621
Jharkhand	62091
Madhya Pradesh	62989
Odisha	73979
Meghalaya	76228.0
Tripura	77358.0
Rajasthan	84837
Chhatisgarh	86860
Nagaland	89607



```python
plt.figure(figsize=(12,8), dpi=100)

sns.barplot(x = gdp_per_capita['GDP per capita'], y =gdp_per_capita.index, palette='Reds' )
plt.xlabel('GDP per capita', fontsize=12)
plt.ylabel('States', fontsize=12)
plt.title('GDP per capita vs States',fontsize=12)
plt.show()
```



Top 5 states based on GDP per capita

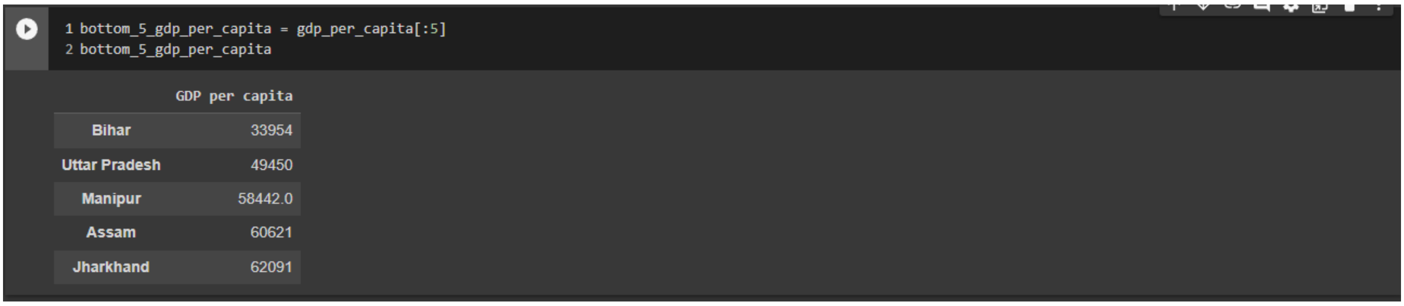


```
top_5_gdp_per_capita = gdp_per_capita[-5:]
top_5_gdp_per_capita
```

	GDP per capita
Uttrakhand	153076.0
Kerala	154778.0
Haryana	164077.0
Sikkim	240274
Goa	271793

Bottom 5 states based on GDP per capita

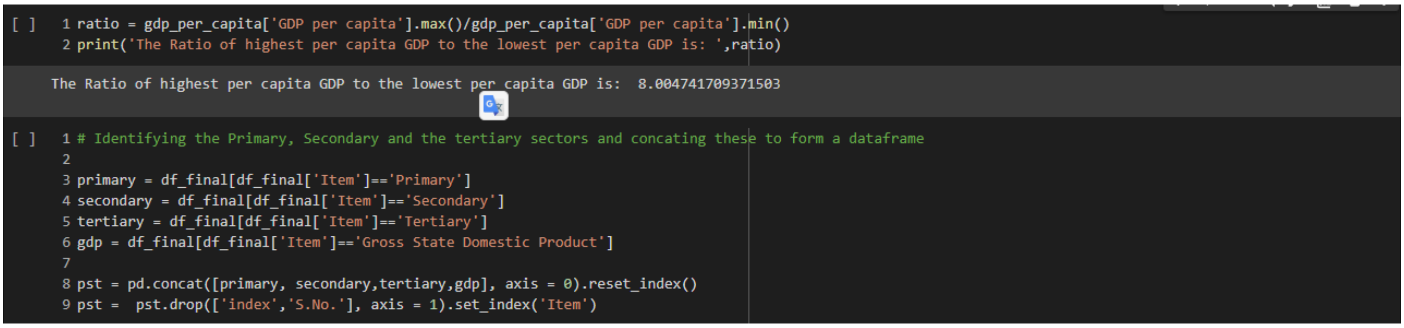
Bottom 5 states based on GDP per capita



```
bottom_5_gdp_per_capita = gdp_per_capita[:5]
bottom_5_gdp_per_capita
```

	GDP per capita
Bihar	33954
Uttar Pradesh	49450
Manipur	58442.0
Assam	60621
Jharkhand	62091

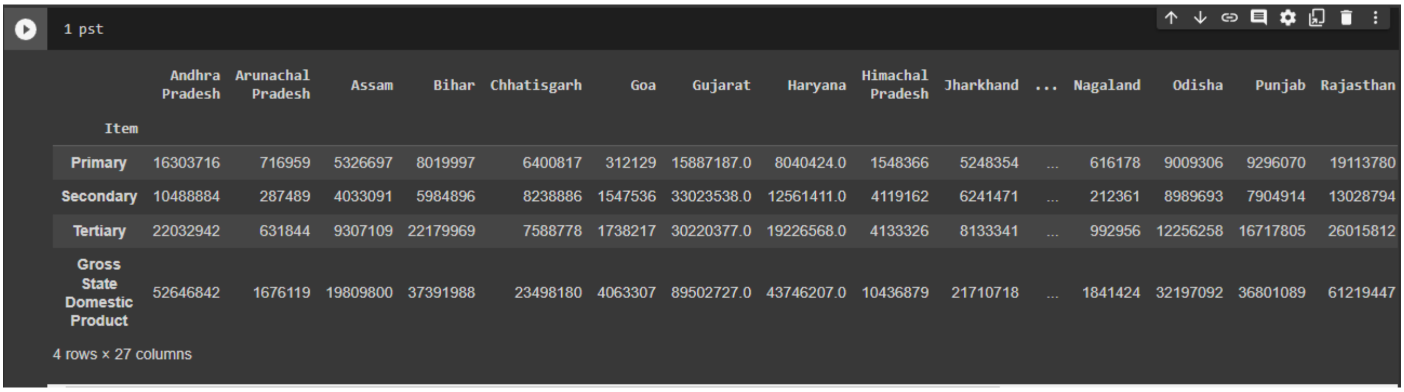
Ratio of highest per capita GDP tp the lowest per capita GDP



```
ratio = gdp_per_capita['GDP per capita'].max()/gdp_per_capita['GDP per capita'].min()
print('The Ratio of highest per capita GDP to the lowest per capita GDP is: ',ratio)
```

```
The Ratio of highest per capita GDP to the lowest per capita GDP is:  8.004741709371503
```

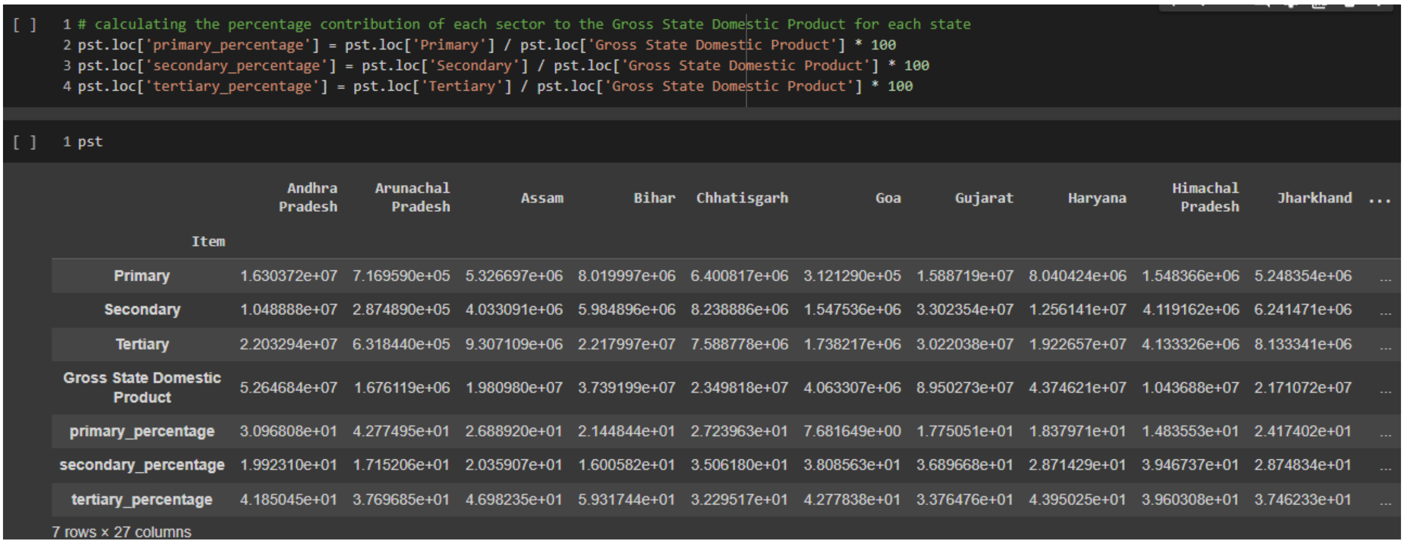
```
# Identifying the Primary, Secondary and the tertiary sectors and concating these to form a dataframe

primary = df_final[df_final['Item']=='Primary']
secondary = df_final[df_final['Item']=='Secondary']
tertiary = df_final[df_final['Item']=='Tertiary']
gdp = df_final[df_final['Item']=='Gross State Domestic Product']

pst = pd.concat([primary, secondary,tertiary,gdp], axis = 0).reset_index()
pst =  pst.drop(['index','S.No.'], axis = 1).set_index('Item')
```



```
pst
```

Item	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhatisgarh	Goa	Gujarat	Haryana	Himachal Pradesh	Jharkhand	...	Nagaland	Odisha	Punjab	Rajasthan
Primary	16303716	716959	5326697	8019997	6400817	312129	15887187.0	8040424.0	1548366	5248354	...	616178	9009306	9296070	19113780
Secondary	10488884	287489	4033091	5984896	8238886	1547536	33023538.0	12561411.0	4119162	6241471	...	212361	8989693	7904914	13028794
Tertiary	22032942	631844	9307109	22179969	7588778	1738217	30220377.0	19226568.0	4133326	8133341	...	992956	12256258	16717805	26015812
Gross State Domestic Product	52646842	1676119	19809800	37391988	23498180	4063307	89502727.0	43746207.0	10436879	21710718	...	1841424	32197092	36801089	61219447

4 rows × 27 columns

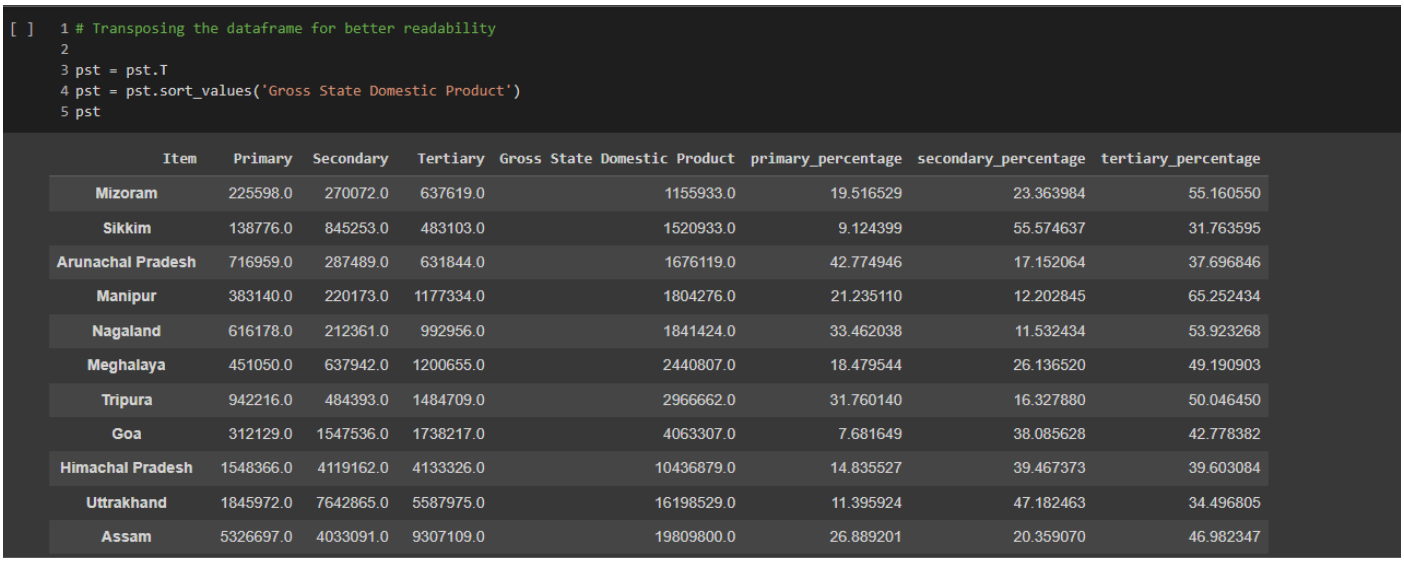


```
# calculating the percentage contribution of each sector to the Gross State Domestic Product for each state
pst.loc['primary_percentage'] = pst.loc['Primary'] / pst.loc['Gross State Domestic Product'] * 100
pst.loc['secondary_percentage'] = pst.loc['Secondary'] / pst.loc['Gross State Domestic Product'] * 100
pst.loc['tertiary_percentage'] = pst.loc['Tertiary'] / pst.loc['Gross State Domestic Product'] * 100
```

```
pst
```

Item	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhatisgarh	Goa	Gujarat	Haryana	Himachal Pradesh	Jharkhand	...
Primary	1.630372e+07	7.169590e+05	5.326697e+06	8.019997e+06	6.400817e+06	3.121290e+05	1.588719e+07	8.040424e+06	1.548366e+06	5.248354e+06	...
Secondary	1.048888e+07	2.874890e+05	4.033091e+06	5.984896e+06	8.238886e+06	1.547536e+06	3.302354e+07	1.256141e+07	4.119162e+06	6.241471e+06	...
Tertiary	2.203294e+07	6.318440e+05	9.307109e+06	2.217997e+07	7.588778e+06	1.738217e+06	3.022038e+07	1.922657e+07	4.133326e+06	8.133341e+06	...
Gross State Domestic Product	5.264684e+07	1.676119e+06	1.980980e+07	3.739199e+07	2.349818e+07	4.063307e+06	8.950273e+07	4.374621e+07	1.043688e+07	2.171072e+07	...
primary_percentage	3.096808e+01	4.277495e+01	2.688920e+01	2.144844e+01	2.723963e+01	7.681649e+00	1.775051e+01	1.837971e+01	1.483553e+01	2.417402e+01	...
secondary_percentage	1.992310e+01	1.715206e+01	2.035907e+01	1.600582e+01	3.506180e+01	3.808563e+01	3.689668e+01	2.871429e+01	3.946737e+01	2.874834e+01	...
tertiary_percentage	4.185045e+01	3.769685e+01	4.698235e+01	5.931744e+01	3.229517e+01	4.277838e+01	3.376476e+01	4.395025e+01	3.960308e+01	3.746233e+01	...

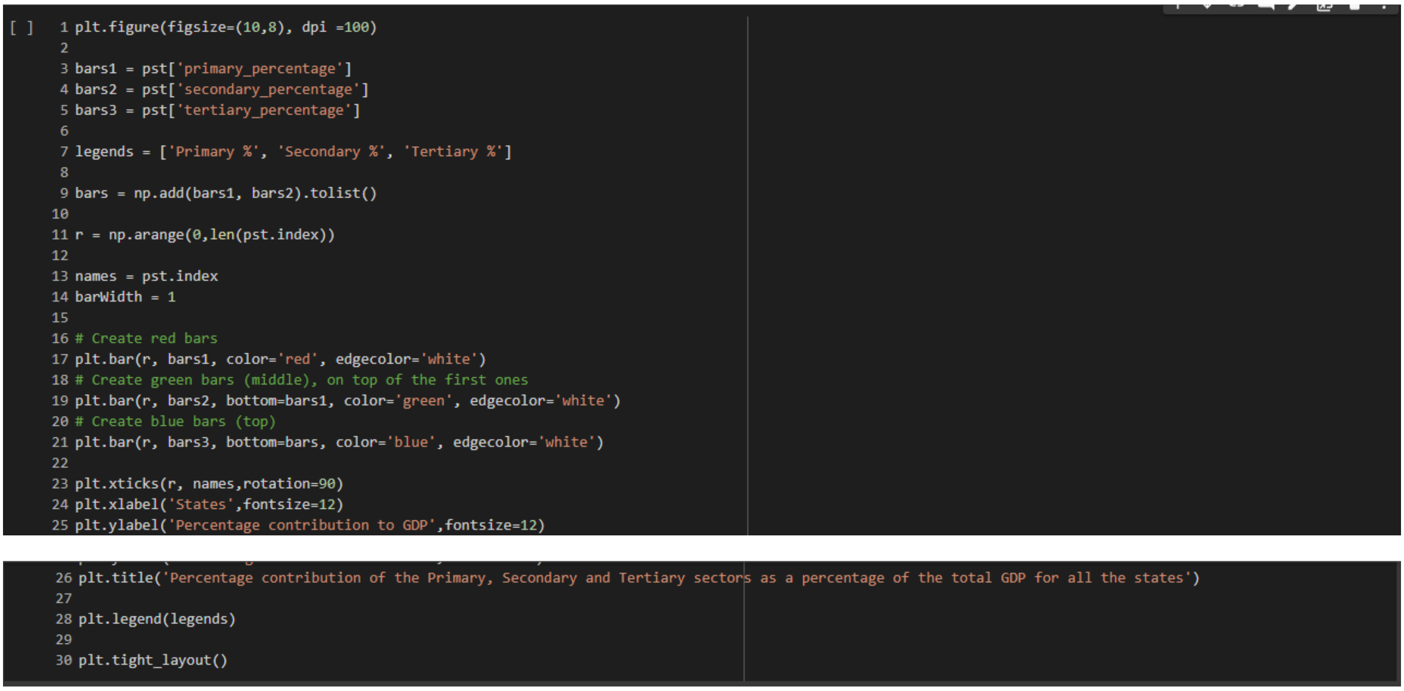
7 rows × 27 columns



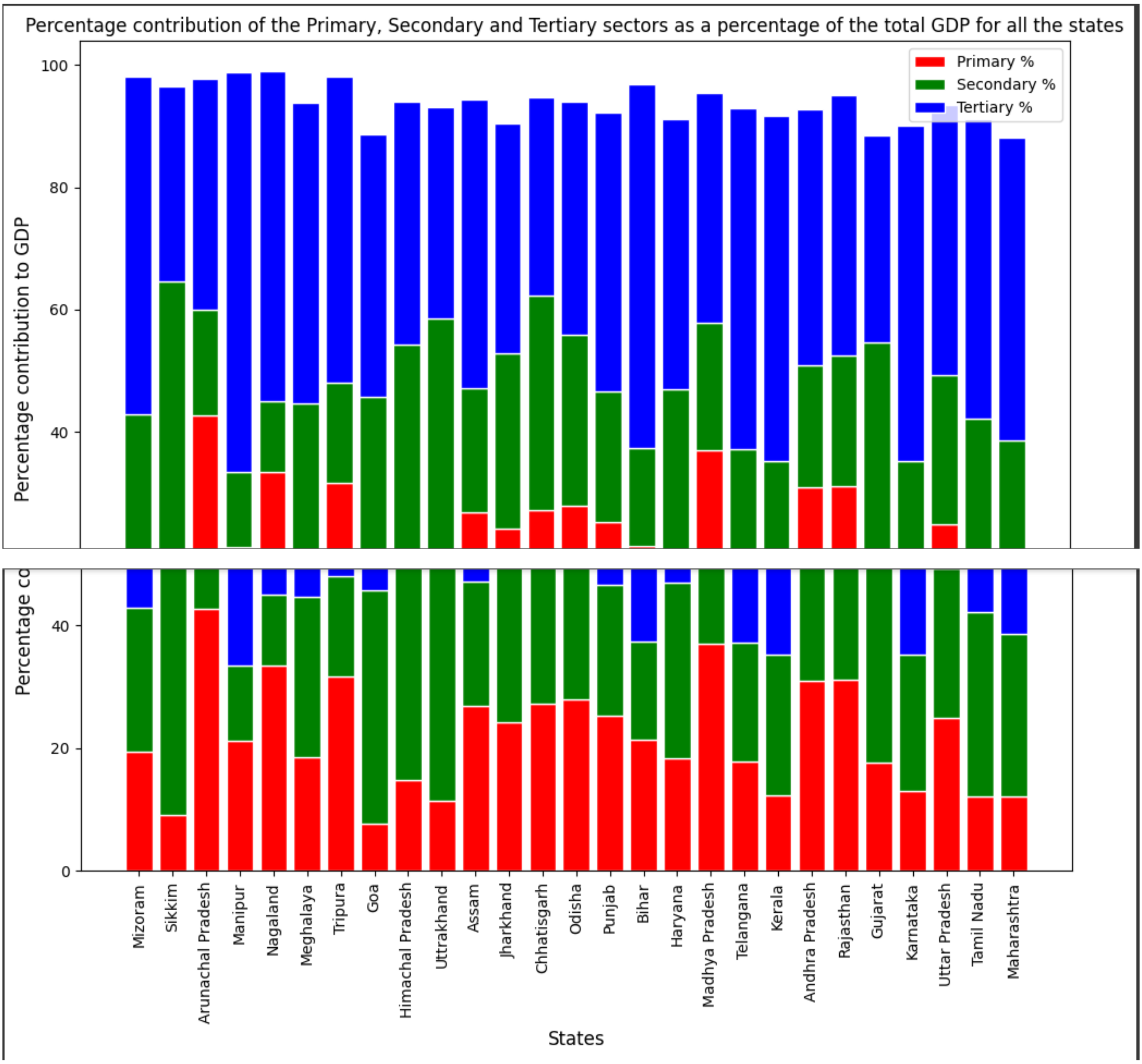
```
# Transposing the dataframe for better readability

pst = pst.T
pst = pst.sort_values('Gross State Domestic Product')
pst
```

Item	Primary	Secondary	Tertiary	Gross State Domestic Product	primary_percentage	secondary_percentage	tertiary_percentage
Mizoram	225598.0	270072.0	637619.0	1155933.0	19.516529	23.363984	55.160550
Sikkim	138776.0	845253.0	483103.0	1520933.0	9.124399	55.574637	31.763595
Arunachal Pradesh	716959.0	287489.0	631844.0	1676119.0	42.774946	17.152064	37.696846
Manipur	383140.0	220173.0	1177334.0	1804276.0	21.235110	12.202845	65.252434
Nagaland	616178.0	212361.0	992956.0	1841424.0	33.462038	11.532434	53.923268
Meghalaya	451050.0	637942.0	1200655.0	2440807.0	18.479544	26.136520	49.190903
Tripura	942216.0	484393.0	1484709.0	2966662.0	31.760140	16.327880	50.046450
Goa	312129.0	1547536.0	1738217.0	4063307.0	7.681649	38.085628	42.778382
Himachal Pradesh	1548366.0	4119162.0	4133326.0	10436879.0	14.835527	39.467373	39.603084
Uttrakhand	1845972.0	7642865.0	5587975.0	16198529.0	11.395924	47.182463	34.496805
Assam	5326697.0	4033091.0	9307109.0	19809800.0	26.889201	20.359070	46.982347

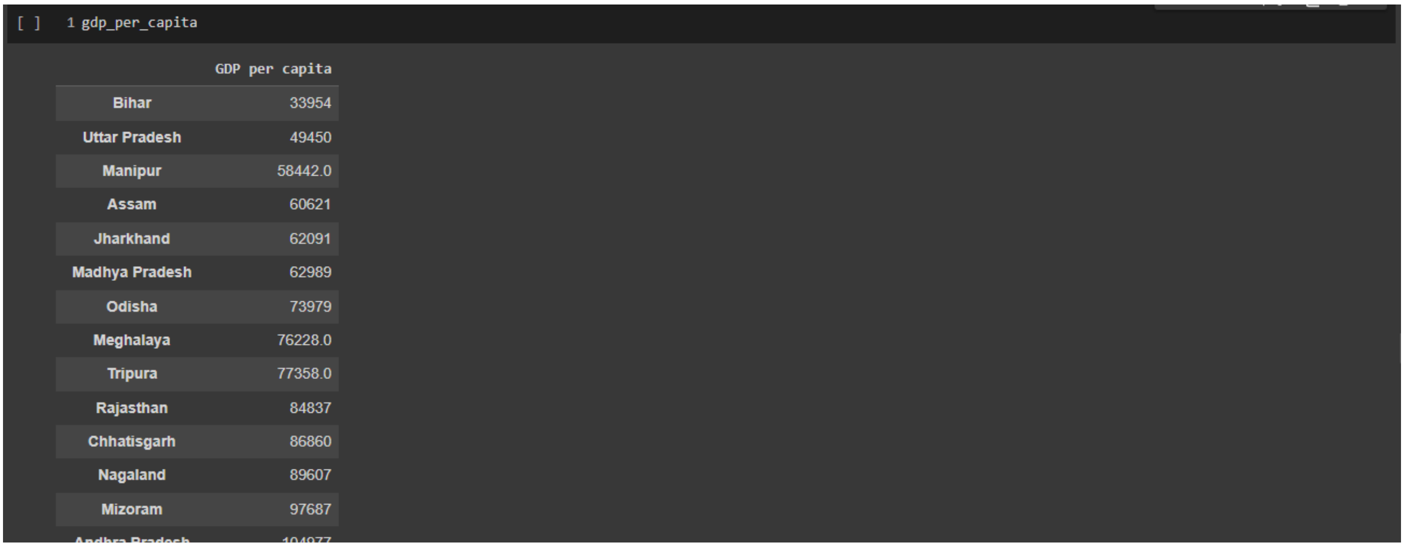
Plotting the percentage contribution of the primary, secondary and tertiary sectors as a percentage of the total GDP for all the states.



```
plt.figure(figsize=(10,8), dpi =100)

bars1 = pst['primary_percentage']
bars2 = pst['secondary_percentage']
bars3 = pst['tertiary_percentage']

legends = ['Primary %', 'Secondary %', 'Tertiary %']

bars = np.add(bars1, bars2).tolist()

r = np.arange(0,len(pst.index))

names = pst.index
barWidth = 1

# Create red bars
plt.bar(r, bars1, color='red', edgecolor='white')
# Create green bars (middle), on top of the first ones
plt.bar(r, bars2, bottom=bars1, color='green', edgecolor='white')
# Create blue bars (top)
plt.bar(r, bars3, bottom=bars, color='blue', edgecolor='white')

plt.xticks(r, names,rotation=90)
plt.xlabel('States',fontsize=12)
plt.ylabel('Percentage contribution to GDP',fontsize=12)
plt.title('Percentage contribution of the Primary, Secondary and Tertiary sectors as a percentage of the total GDP for all the states')

plt.legend(legends)

plt.tight_layout()
```

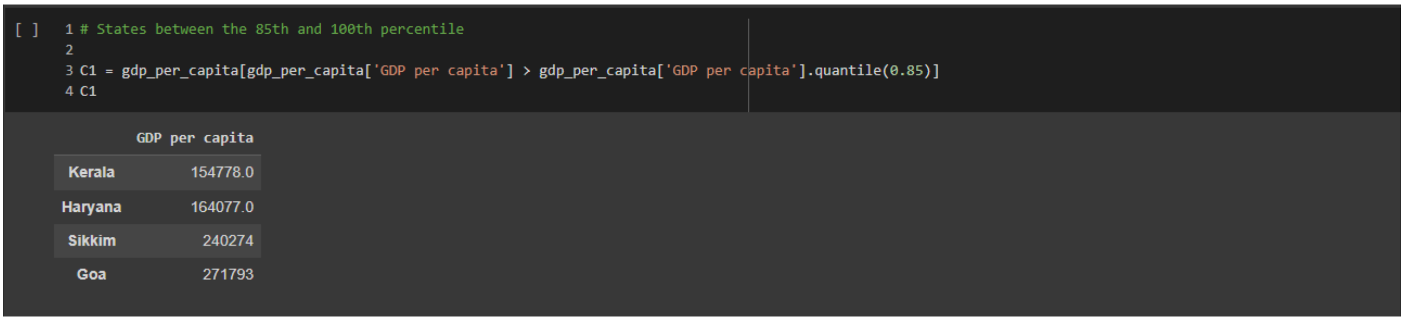


Dividing the states in to group based on GDP per capita for the 20th, 50th, 85th and 100th percentile values



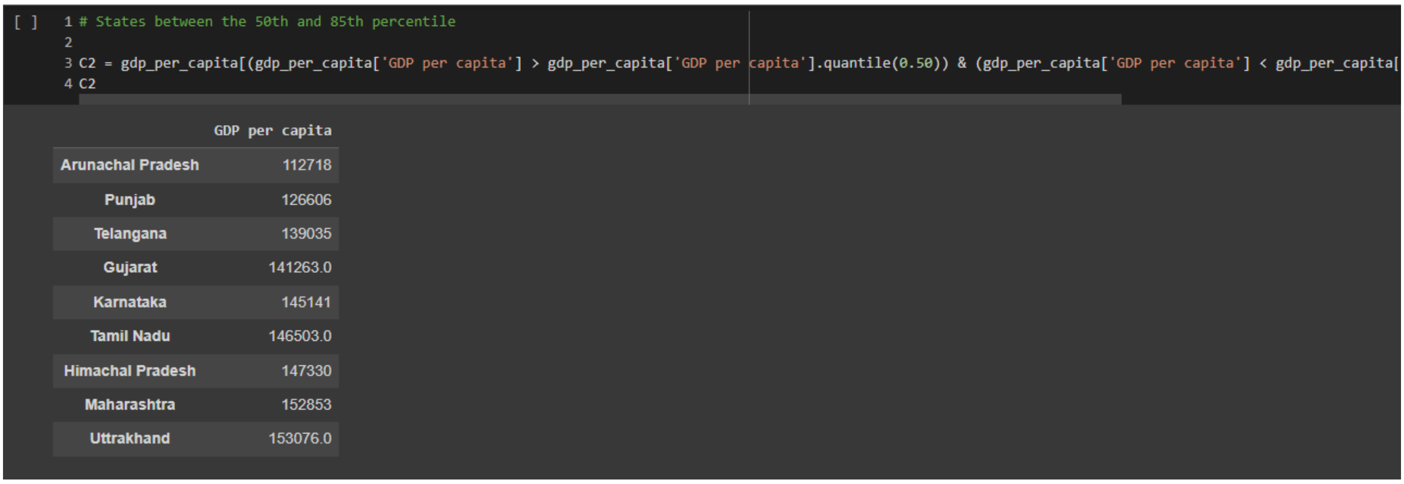
```
[ ]  gdp_per_capita
```

	GDP per capita
Bihar	33954
Uttar Pradesh	49450
Manipur	58442.0
Assam	60621
Jharkhand	62091
Madhya Pradesh	62989
Odisha	73979
Meghalaya	76228.0
Tripura	77358.0
Rajasthan	84837
Chhatisgarh	86860
Nagaland	89607
Mizoram	97687
Andhra Pradesh	104977



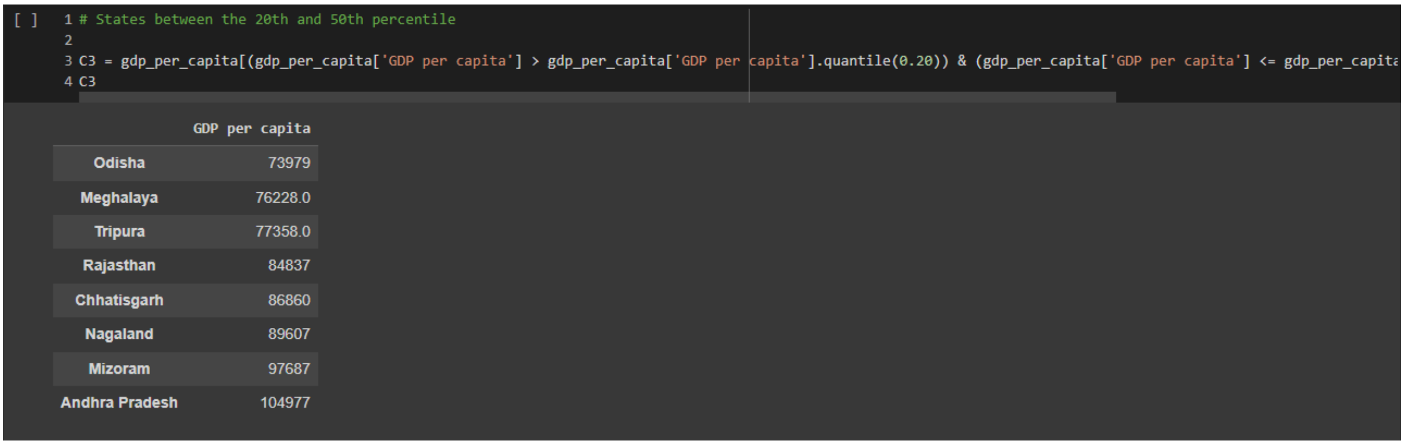
```
[ ]  # States between the 85th and 100th percentile

     C1 = gdp_per_capita[gdp_per_capita['GDP per capita'] > gdp_per_capita['GDP per capita'].quantile(0.85)]
     C1
```

	GDP per capita
Kerala	154778.0
Haryana	164077.0
Sikkim	240274
Goa	271793



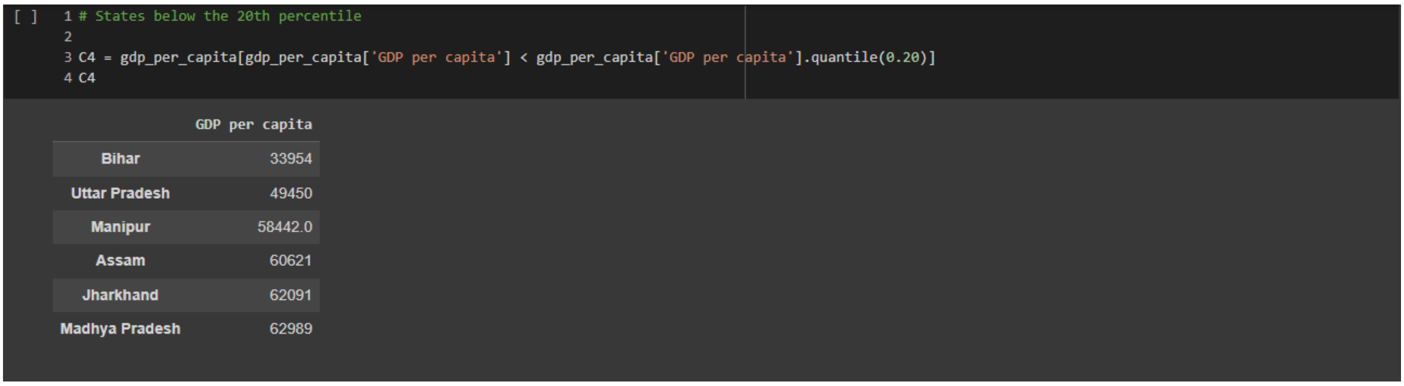
```
[ ]  # States between the 50th and 85th percentile

     C2 = gdp_per_capita[(gdp_per_capita['GDP per capita'] > gdp_per_capita['GDP per capita'].quantile(0.50)) & (gdp_per_capita['GDP per capita'] < gdp_per_capita[
     C2
```

	GDP per capita
Arunachal Pradesh	112718
Punjab	126606
Telangana	139035
Gujarat	141263.0
Karnataka	145141
Tamil Nadu	146503.0
Himachal Pradesh	147330
Maharashtra	152853
Uttrakhand	153076.0



```
[ ]  # States between the 20th and 50th percentile

     C3 = gdp_per_capita[(gdp_per_capita['GDP per capita'] > gdp_per_capita['GDP per capita'].quantile(0.20)) & (gdp_per_capita['GDP per capita'] <= gdp_per_capita
     C3
```

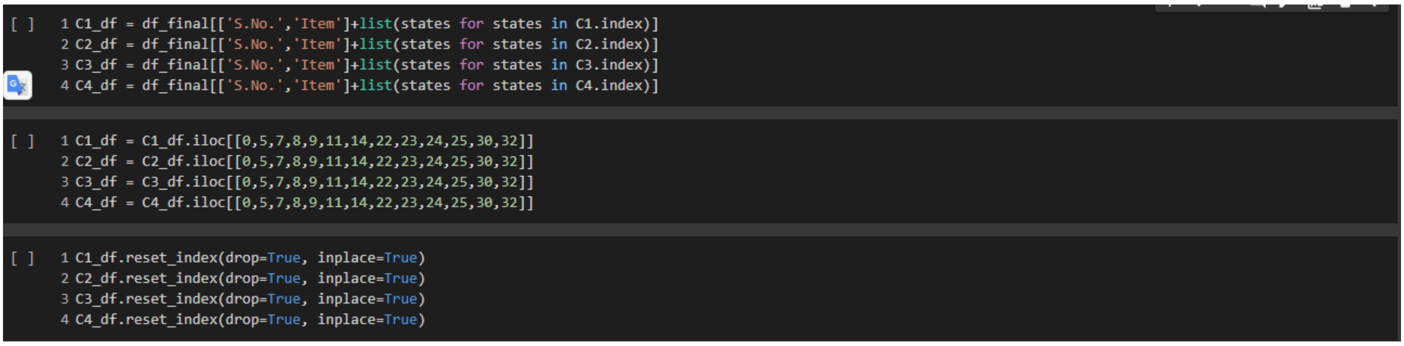
	GDP per capita
Odisha	73979
Meghalaya	76228.0
Tripura	77358.0
Rajasthan	84837
Chhatisgarh	86860
Nagaland	89607
Mizoram	97687
Andhra Pradesh	104977



```
[ ]
# States below the 20th percentile

C4 = gdp_per_capita[gdp_per_capita['GDP per capita'] < gdp_per_capita['GDP per capita'].quantile(0.20)]
C4
```

	GDP per capita
Bihar	33954
Uttar Pradesh	49450
Manipur	58442.0
Assam	60621
Jharkhand	62091
Madhya Pradesh	62989

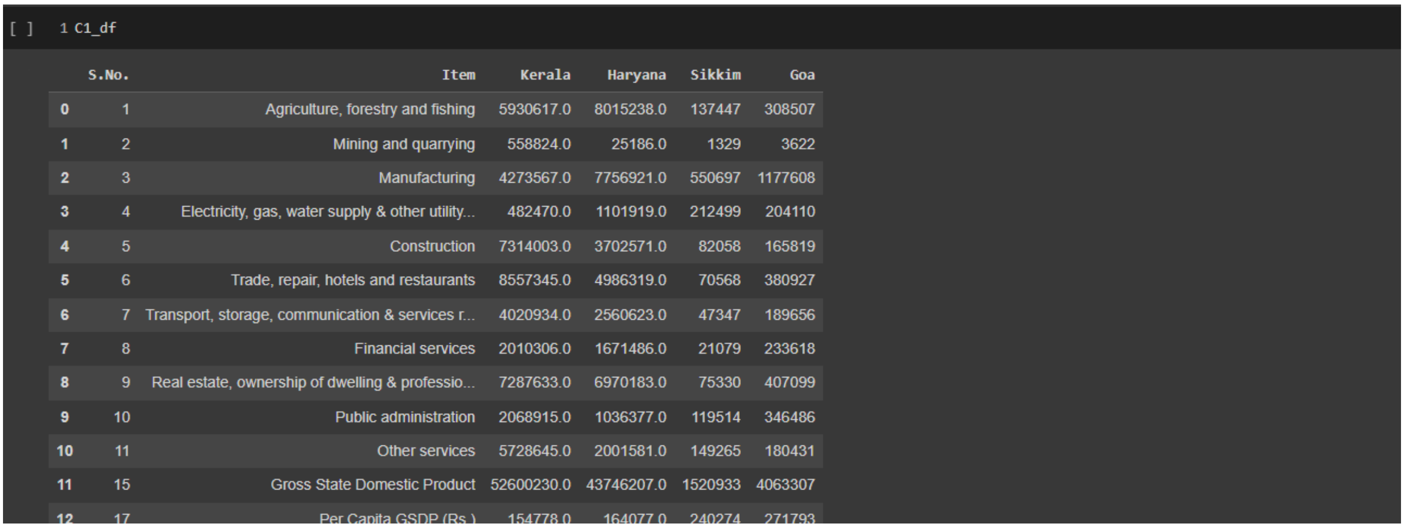
Creating data frame for C1, C2, C3 and C4 states



```
[ ]
C1_df = df_final[['S.No.','Item']+list(states for states in C1.index)]
C2_df = df_final[['S.No.','Item']+list(states for states in C2.index)]
C3_df = df_final[['S.No.','Item']+list(states for states in C3.index)]
C4_df = df_final[['S.No.','Item']+list(states for states in C4.index)]
```

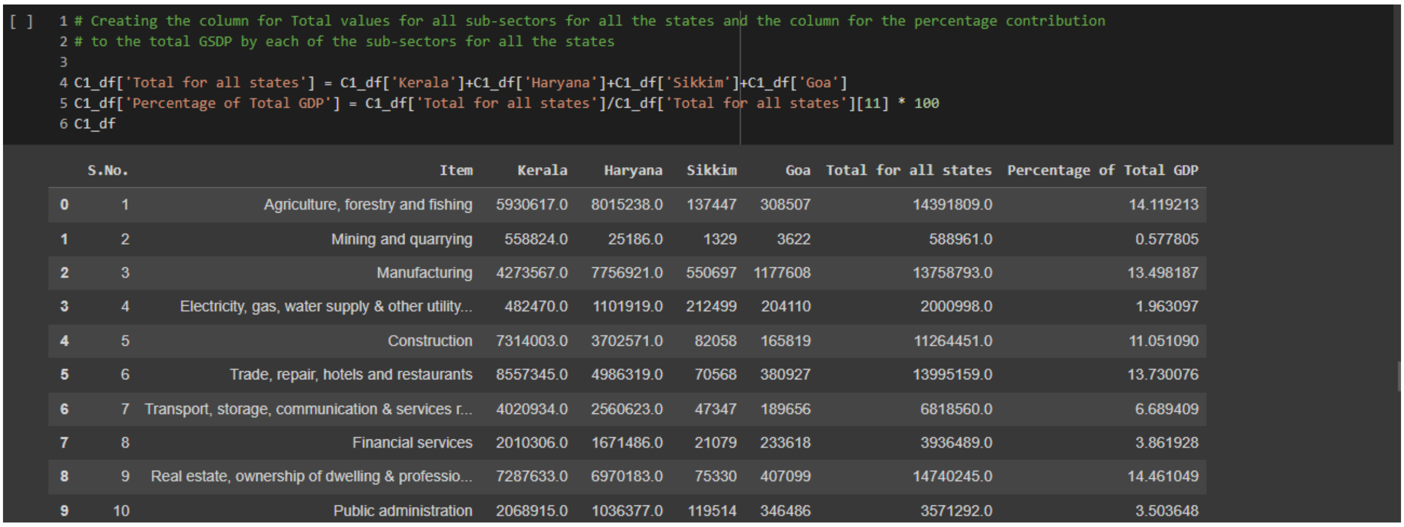
```
[ ]
C1_df = C1_df.iloc[[0,5,7,8,9,11,14,22,23,24,25,30,32]]
C2_df = C2_df.iloc[[0,5,7,8,9,11,14,22,23,24,25,30,32]]
C3_df = C3_df.iloc[[0,5,7,8,9,11,14,22,23,24,25,30,32]]
C4_df = C4_df.iloc[[0,5,7,8,9,11,14,22,23,24,25,30,32]]
```

```
[ ]
C1_df.reset_index(drop=True, inplace=True)
C2_df.reset_index(drop=True, inplace=True)
C3_df.reset_index(drop=True, inplace=True)
C4_df.reset_index(drop=True, inplace=True)
```



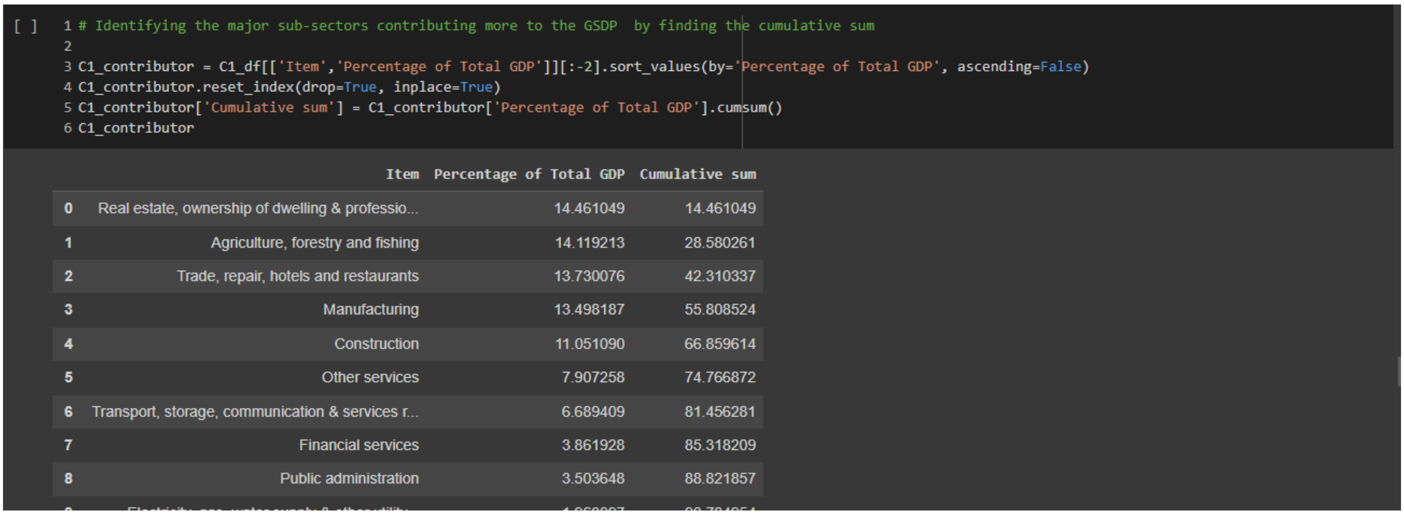
```
[ ]
C1_df
```

	S.No.	Item	Kerala	Haryana	Sikkim	Goa
0	1	Agriculture, forestry and fishing	5930617.0	8015238.0	137447	308507
1	2	Mining and quarrying	558824.0	25186.0	1329	3622
2	3	Manufacturing	4273567.0	7756921.0	550697	1177608
3	4	Electricity, gas, water supply & other utility...	482470.0	1101919.0	212499	204110
4	5	Construction	7314003.0	3702571.0	82058	165819
5	6	Trade, repair, hotels and restaurants	8557345.0	4986319.0	70568	380927
6	7	Transport, storage, communication & services r...	4020934.0	2560623.0	47347	189656
7	8	Financial services	2010306.0	1671486.0	21079	233618
8	9	Real estate, ownership of dwelling & professio...	7287633.0	6970183.0	75330	407099
9	10	Public administration	2068915.0	1036377.0	119514	346486
10	11	Other services	5728645.0	2001581.0	149265	180431
11	15	Gross State Domestic Product	52600230.0	43746207.0	1520933	4063307
12	17	Per Capita GSDP (Rs.)	154778.0	164077.0	240274	271793



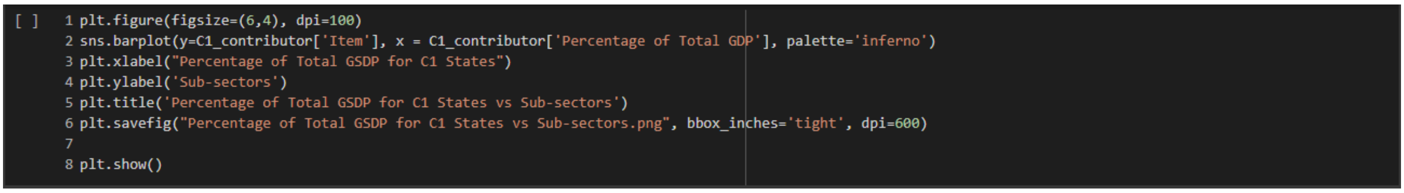
```
[ ]
# Creating the column for Total values for all sub-sectors for all the states and the column for the percentage contribution
# to the total GSDP by each of the sub-sectors for all the states

C1_df['Total for all states'] = C1_df['Kerala']+C1_df['Haryana']+C1_df['Sikkim']+C1_df['Goa']
C1_df['Percentage of Total GDP'] = C1_df['Total for all states']/C1_df['Total for all states'][11] * 100
C1_df
```

	S.No.	Item	Kerala	Haryana	Sikkim	Goa	Total for all states	Percentage of Total GDP
0	1	Agriculture, forestry and fishing	5930617.0	8015238.0	137447	308507	14391809.0	14.119213
1	2	Mining and quarrying	558824.0	25186.0	1329	3622	588961.0	0.577805
2	3	Manufacturing	4273567.0	7756921.0	550697	1177608	13758793.0	13.498187
3	4	Electricity, gas, water supply & other utility...	482470.0	1101919.0	212499	204110	2000998.0	1.963097
4	5	Construction	7314003.0	3702571.0	82058	165819	11264451.0	11.051090
5	6	Trade, repair, hotels and restaurants	8557345.0	4986319.0	70568	380927	13995159.0	13.730076
6	7	Transport, storage, communication & services r...	4020934.0	2560623.0	47347	189656	6818560.0	6.689409
7	8	Financial services	2010306.0	1671486.0	21079	233618	3936489.0	3.861928
8	9	Real estate, ownership of dwelling & professio...	7287633.0	6970183.0	75330	407099	14740245.0	14.461049
9	10	Public administration	2068915.0	1036377.0	119514	346486	3571292.0	3.503648

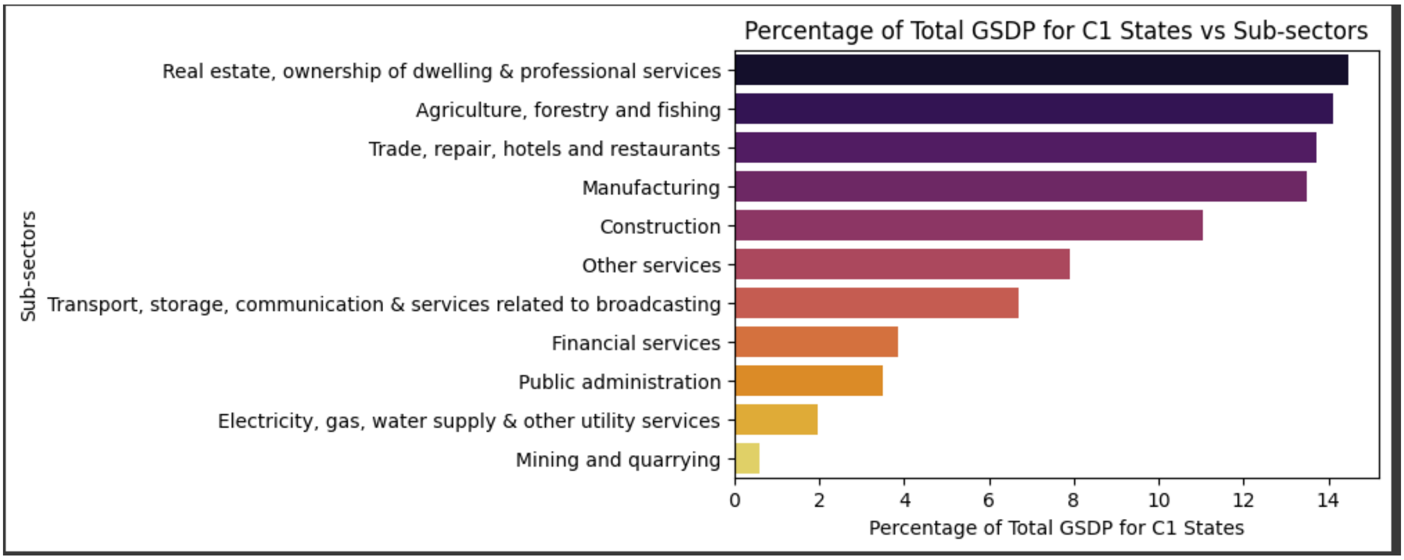


```
[ ]  # Identifying the major sub-sectors contributing more to the GSDP  by finding the cumulative sum

     C1_contributor = C1_df[['Item','Percentage of Total GDP']][:-2].sort_values(by='Percentage of Total GDP', ascending=False)
     C1_contributor.reset_index(drop=True, inplace=True)
     C1_contributor['Cumulative sum'] = C1_contributor['Percentage of Total GDP'].cumsum()
     C1_contributor
```

```
                                                Item  Percentage of Total GDP  Cumulative sum
0  Real estate, ownership of dwelling & professio...                14.461049       14.461049
1                  Agriculture, forestry and fishing                14.119213       28.580261
2                Trade, repair, hotels and restaurants                13.730076       42.310337
3                                      Manufacturing                13.498187       55.808524
4                                       Construction                11.051090       66.859614
5                                     Other services                 7.907258       74.766872
6  Transport, storage, communication & services r...                 6.689409       81.456281
7                                 Financial services                 3.861928       85.318209
8                              Public administration                 3.503648       88.821857
```

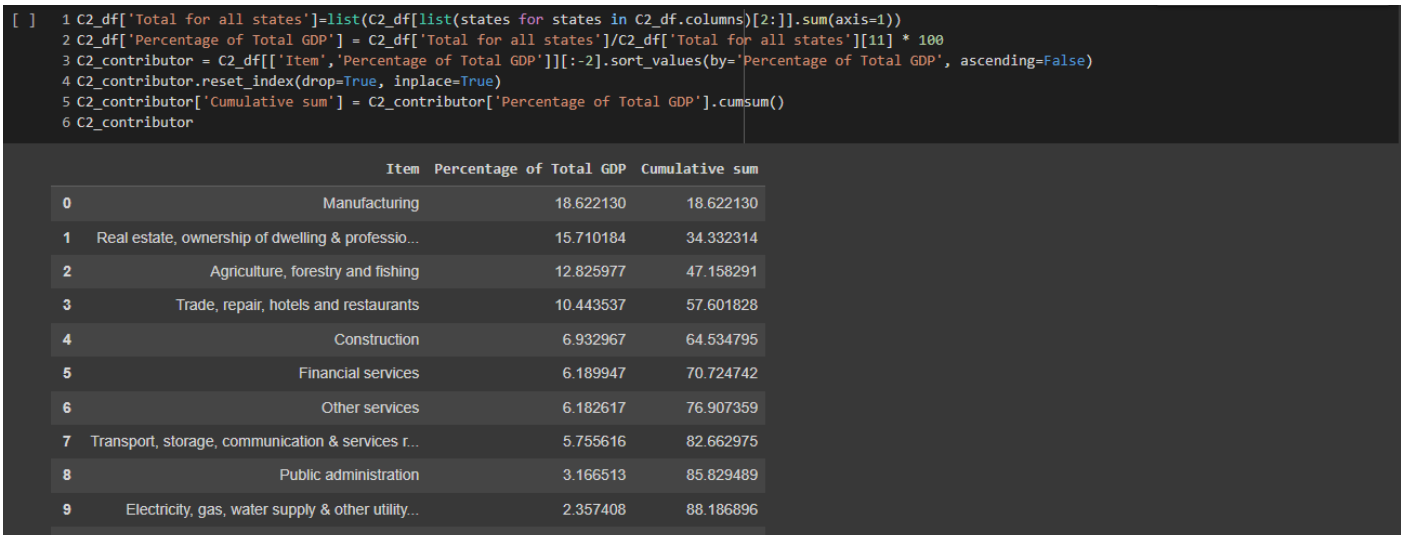


```
[ ]  plt.figure(figsize=(6,4), dpi=100)
     sns.barplot(y=C1_contributor['Item'], x = C1_contributor['Percentage of Total GDP'], palette='inferno')
     plt.xlabel("Percentage of Total GSDP for C1 States")
     plt.ylabel('Sub-sectors')
     plt.title('Percentage of Total GSDP for C1 States vs Sub-sectors')
     plt.savefig("Percentage of Total GSDP for C1 States vs Sub-sectors.png", bbox_inches='tight', dpi=600)

     plt.show()
```



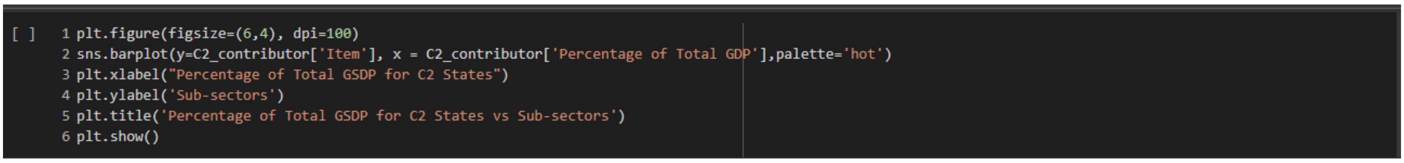
C1 states subsectors like Real Estate, Agriculture, Trade and Hotels, Manufacturing contributes evenly with very high contribution for each category which leads to the overall increase in the GDP for C1 States.

Construction also contributes substantially to the total GDP for C1 states as these states have rapid urbanization taking place which leads to increase in overall GDP.

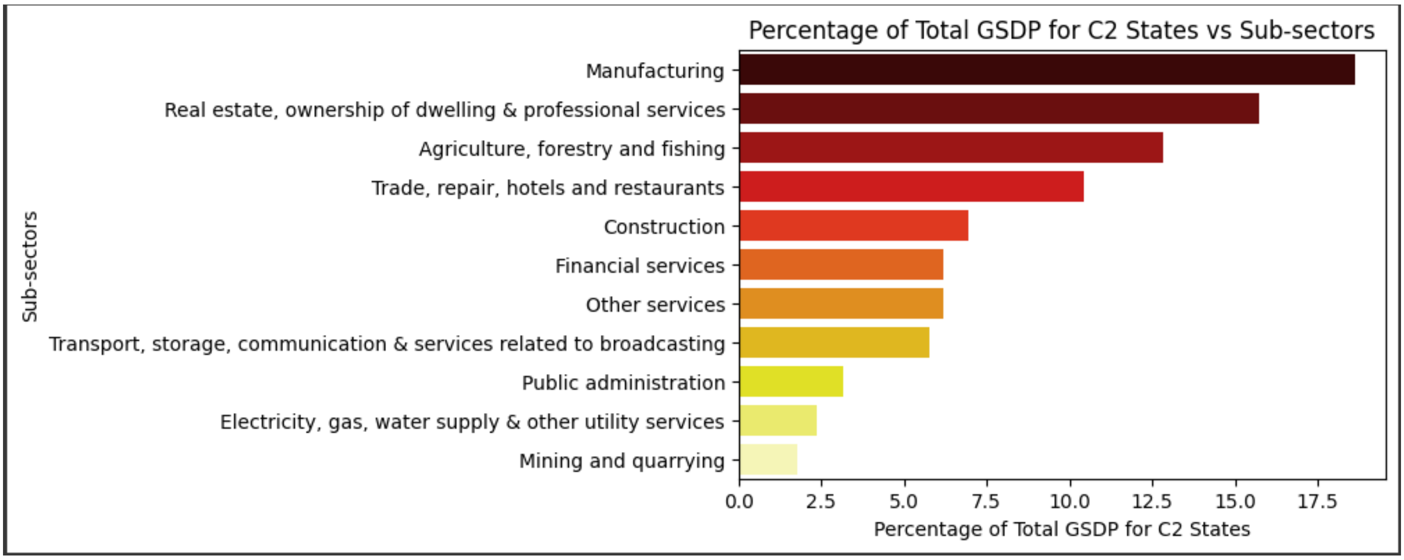


```
[ ]
C2_df['Total for all states']=list(C2_df[list(states for states in C2_df.columns)[2:]].sum(axis=1))
C2_df['Percentage of Total GDP'] = C2_df['Total for all states']/C2_df['Total for all states'][11] * 100
C2_contributor = C2_df[['Item','Percentage of Total GDP']][:-2].sort_values(by='Percentage of Total GDP', ascending=False)
C2_contributor.reset_index(drop=True, inplace=True)
C2_contributor['Cumulative sum'] = C2_contributor['Percentage of Total GDP'].cumsum()
C2_contributor
```

	Item	Percentage of Total GDP	Cumulative sum
0	Manufacturing	18.622130	18.622130
1	Real estate, ownership of dwelling & professio...	15.710184	34.332314
2	Agriculture, forestry and fishing	12.825977	47.158291
3	Trade, repair, hotels and restaurants	10.443537	57.601828
4	Construction	6.932967	64.534795
5	Financial services	6.189947	70.724742
6	Other services	6.182617	76.907359
7	Transport, storage, communication & services r...	5.755616	82.662975
8	Public administration	3.166513	85.829489
9	Electricity, gas, water supply & other utility...	2.357408	88.186896



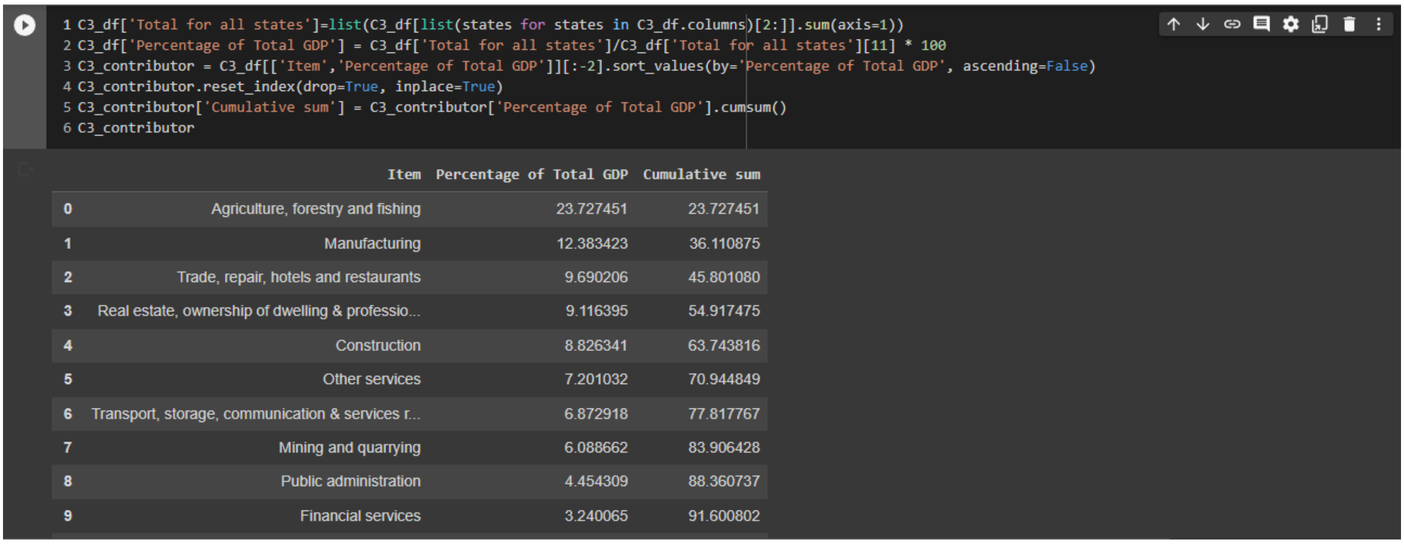
```
[ ]
plt.figure(figsize=(6,4), dpi=100)
sns.barplot(y=C2_contributor['Item'], x = C2_contributor['Percentage of Total GDP'],palette='hot')
plt.xlabel("Percentage of Total GSDP for C2 States")
plt.ylabel('Sub-sectors')
plt.title('Percentage of Total GSDP for C2 States vs Sub-sectors')
plt.show()
```



C2 states Manufacturing leads in terms of overall contribution to GDP which comes a no surprise as states like Gujarat, Karnataka, Tamil Nadu and Maharashtra are considered to be manufacturing hubs of India which huge investments in hte field of Automobiles and other tech industries are taking place in these states.

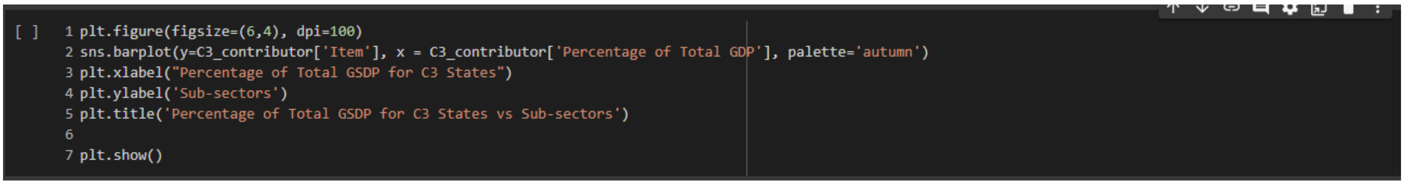
Real Estate and Professional services also contribute substantially to the total GDP for C2 states as these states have rapid urbanization taking place and people are moving to these states from villages in search of jobs and better livelihood.

Agriculture forms the backbone of India's GDP so it is obvious that it finds a place in the top 3 sub-sectors for C2 states as well but since rapid urbanization may be leading to less land available for agricultural purpose it contributes fairly less when compared to the top 2 sub-sectors.

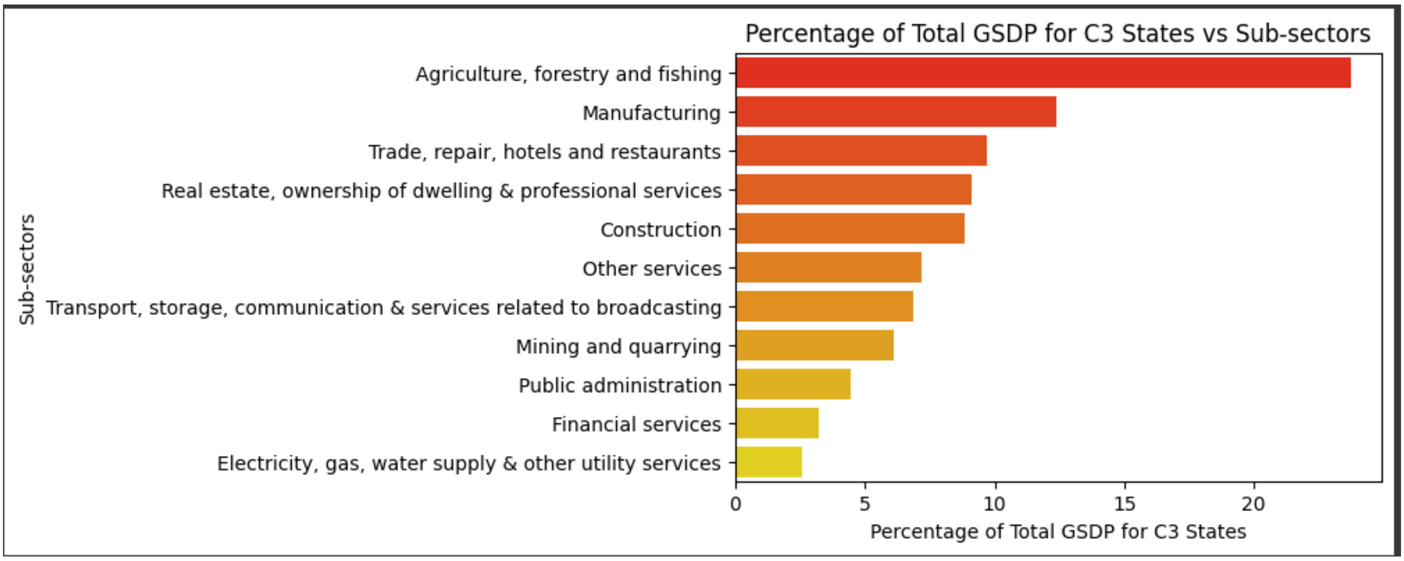


```python
C3_df['Total for all states']=list(C3_df[list(states for states in C3_df.columns)[2:]].sum(axis=1))
C3_df['Percentage of Total GDP'] = C3_df['Total for all states']/C3_df['Total for all states'][11] * 100
C3_contributor = C3_df[['Item','Percentage of Total GDP']][:-2].sort_values(by='Percentage of Total GDP', ascending=False)
C3_contributor.reset_index(drop=True, inplace=True)
C3_contributor['Cumulative sum'] = C3_contributor['Percentage of Total GDP'].cumsum()
C3_contributor
```

	Item	Percentage of Total GDP	Cumulative sum
0	Agriculture, forestry and fishing	23.727451	23.727451
1	Manufacturing	12.383423	36.110875
2	Trade, repair, hotels and restaurants	9.690206	45.801080
3	Real estate, ownership of dwelling & professio...	9.116395	54.917475
4	Construction	8.826341	63.743816
5	Other services	7.201032	70.944849
6	Transport, storage, communication & services r...	6.872918	77.817767
7	Mining and quarrying	6.088662	83.906428
8	Public administration	4.454309	88.360737
9	Financial services	3.240065	91.600802



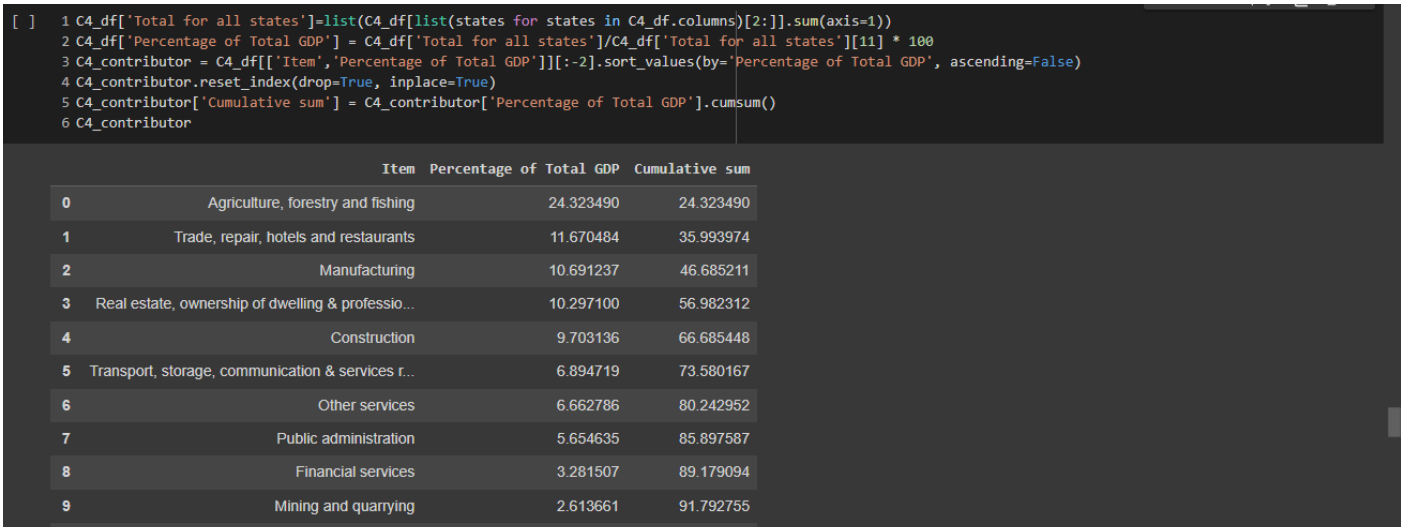
```python
[ ]
plt.figure(figsize=(6,4), dpi=100)
sns.barplot(y=C3_contributor['Item'], x = C3_contributor['Percentage of Total GDP'], palette='autumn')
plt.xlabel("Percentage of Total GSDP for C3 States")
plt.ylabel('Sub-sectors')
plt.title('Percentage of Total GSDP for C3 States vs Sub-sectors')

plt.show()
```



C3 states like Andhra Pradesh, Odisha, Meghalaya, Chattisgarh, Mizoram have highly arable land and receive good amount of rain every year during the monsoon so it is obvious that Agriculture is the sub-sector that contributes more than 23% to these states.

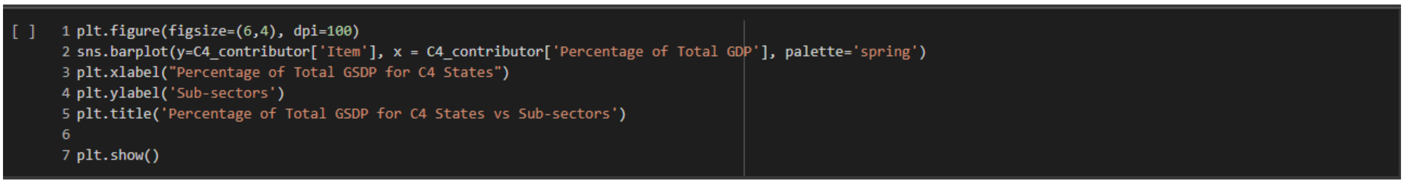
Manufacturing is at a distant second place contributing about 12% to the overall GDP followed by Trade, Hotels and restraunts as these states are home to some of the top tourist attractions in India.

Slowly but steadily these states are experiencing increase in urbanization and hence Real Estate and Construction feature in the top 5 contributors as well.

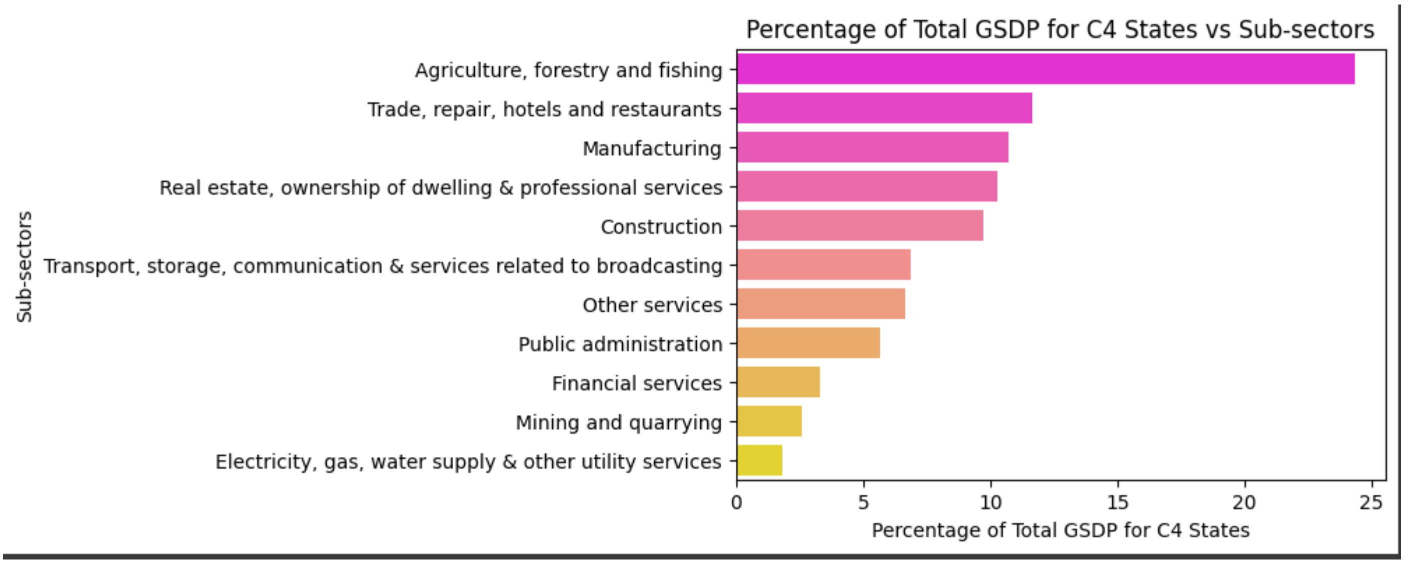


```
[ ]  C4_df['Total for all states']=list(C4_df[list(states for states in C4_df.columns)[2:]].sum(axis=1))
     C4_df['Percentage of Total GDP'] = C4_df['Total for all states']/C4_df['Total for all states'][11] * 100
     C4_contributor = C4_df[['Item','Percentage of Total GDP']][:-2].sort_values(by='Percentage of Total GDP', ascending=False)
     C4_contributor.reset_index(drop=True, inplace=True)
     C4_contributor['Cumulative sum'] = C4_contributor['Percentage of Total GDP'].cumsum()
     C4_contributor
```

	Item	Percentage of Total GDP	Cumulative sum
0	Agriculture, forestry and fishing	24.323490	24.323490
1	Trade, repair, hotels and restaurants	11.670484	35.993974
2	Manufacturing	10.691237	46.685211
3	Real estate, ownership of dwelling & professio...	10.297100	56.982312
4	Construction	9.703136	66.685448
5	Transport, storage, communication & services r...	6.894719	73.580167
6	Other services	6.662786	80.242952
7	Public administration	5.654635	85.897587
8	Financial services	3.281507	89.179094
9	Mining and quarrying	2.613661	91.792755



```
[ ]  plt.figure(figsize=(6,4), dpi=100)
     sns.barplot(y=C4_contributor['Item'], x = C4_contributor['Percentage of Total GDP'], palette='spring')
     plt.xlabel("Percentage of Total GSDP for C4 States")
     plt.ylabel('Sub-sectors')
     plt.title('Percentage of Total GSDP for C4 States vs Sub-sectors')

     plt.show()
```



C4 states like Bihar, Jharkhand, Uttar Pradesh have low literacy rate and huge population(U.P. is the most populous state in India) and thus agriculture features at the top again.

UP is one of the top tourist attracting states as it is home to some of the most amazing places like Agra which has the Taj Mahal, Varansi (Regarded as the spiritual capital of India, the city draws Hindu pilgrims who bathe in the Ganges River’s sacred waters and perform funeral rites. Along the city's winding streets are some 2,000 temples, including Kashi Vishwanath, the “Golden Temple,” dedicated to the Hindu god Shiva.), Jim Corbett National Park, India's oldest national park, opened in 1936, with a Bengal tiger reserve, visitor centre & safaris.

C1, C2, C3 and C4 states:

1.The major sub sectors contributing to the economy of the states are:

Agriculture, Real Estate, Manufacturing, Trade Hotels and restraunts and Construction. One key observation is that for C1,C2 states the major contribution comes from Real Estate which is reasonable as these states have a big real estate and housing industry due to people migrating from villages to these states for employment. Agriculture forms the back-bone of the Indian economy and hence it features in the top 3 spot for all the category of states India is home to some of the top hotels, restraunts and tourist destinations and hence these contribute significantly to the economy as well.

Slowly but steadily India is working to increase its manufacturing capabilities and new companies are opening their factories in India which is the reason for manufacting appearing as a top contributor as well. The 'Make in India' initiative by PM. Modi is also helping to increase manufacturing activities in India. For any country to improve the standard of living of its people, it requires good quality infrastructure. India is experincing rapid urbanization with several new roads, bridges, ports etc. being constructed to aid in increasing the GDP of India which is growing very fast and people need good quality jobs. Construction provides jobs to several people and leads to people having better livelihood. Sub-sectors for which states should pay invest more or pay greater attention: Improving Road, Railways, Air transpotation services will not only help in easier access for people to each and every nook and corner of India but also aid in transportation of goods and materials required for construction purposes. So all the catergories of states should improve Transportation seervices.

C1 States:

All the 4 states in C1 categories are top tourist destination so they should invest more in:

Trade, repair, hotels and restaurants.

Transport, storage, communication & services for easier access to tourist destination.

Being high GDP per capita states, they should focus on improving the Financial Services and also invest more in public administration.

C2 States:

C2 states comprises of some of the powerhouse states in India which contribute immensely to India's overall GDP:

States like Karnataka, Maharashtra, Tamil Nadu and Telengana are manufacturing hubs of India and thus should invest even more in the Manufacturing sector.

Same could also be said for Real estate indudtry as well.

Construction, Transportation and other services should also be looked at to improve the overall GDP.

C3 States:

C3 states should focus on:

Manufacturing Sector provinding easier access to lands to industries.

Mining and quarrying should also be considered as these states have large deposits of natural resources.

States in the C3 categories like Odisha, Mizoram, Nagaland, Tripura and Meghalaya are big tourist destinations so they should invest more in Trade, repair, hotels and restaurants and Transport, storage, communication & services.

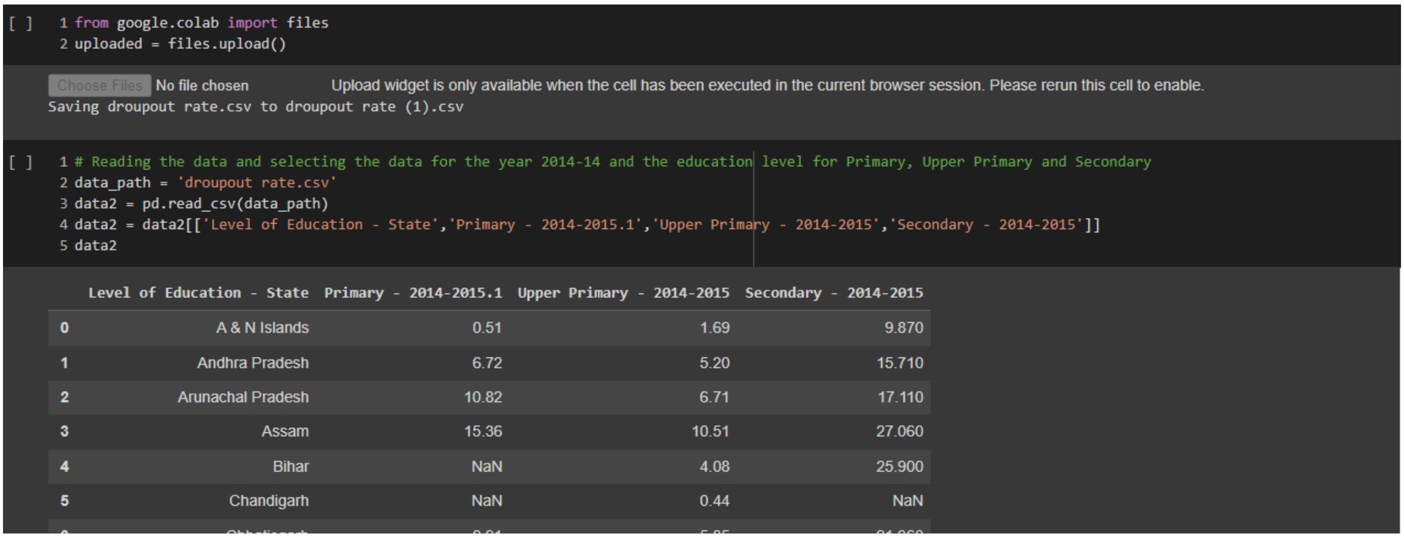
C4 States:

C4 states should focus on:

States like UP and Bihar should focus on investing in Public administration,Transport, storage, communication & services as they have a very large population and some people still live in remote places with mo direct access to the major cities .

Invest more in Trade, repair, hotels and restaurants to boost tourism. Construction is also area where more investment is needed and not to forget Transport, storage, communication & services.

5.Results –



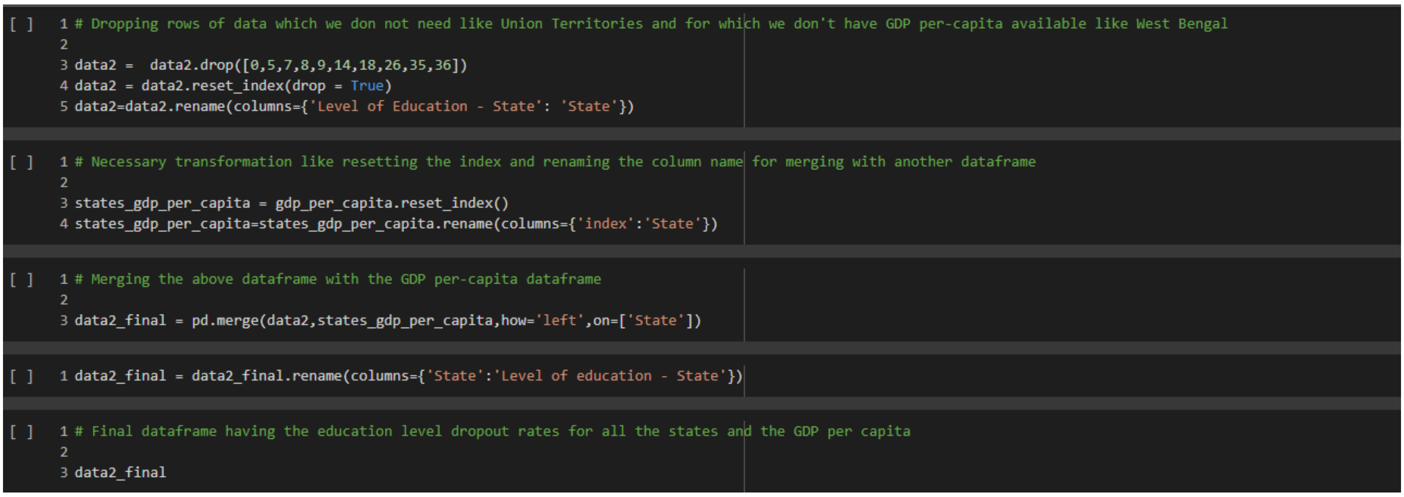
```
from google.colab import files
uploaded = files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving droupout rate.csv to droupout rate (1).csv
```

```
# Reading the data and selecting the data for the year 2014-14 and the education level for Primary, Upper Primary and Secondary
data_path = 'droupout rate.csv'
data2 = pd.read_csv(data_path)
data2 = data2[['Level of Education - State','Primary - 2014-2015.1','Upper Primary - 2014-2015','Secondary - 2014-2015']]
data2
```

	Level of Education - State	Primary - 2014-2015.1	Upper Primary - 2014-2015	Secondary - 2014-2015
0	A & N Islands	0.51	1.69	9.870
1	Andhra Pradesh	6.72	5.20	15.710
2	Arunachal Pradesh	10.82	6.71	17.110
3	Assam	15.36	10.51	27.060
4	Bihar	NaN	4.08	25.900
5	Chandigarh	NaN	0.44	NaN



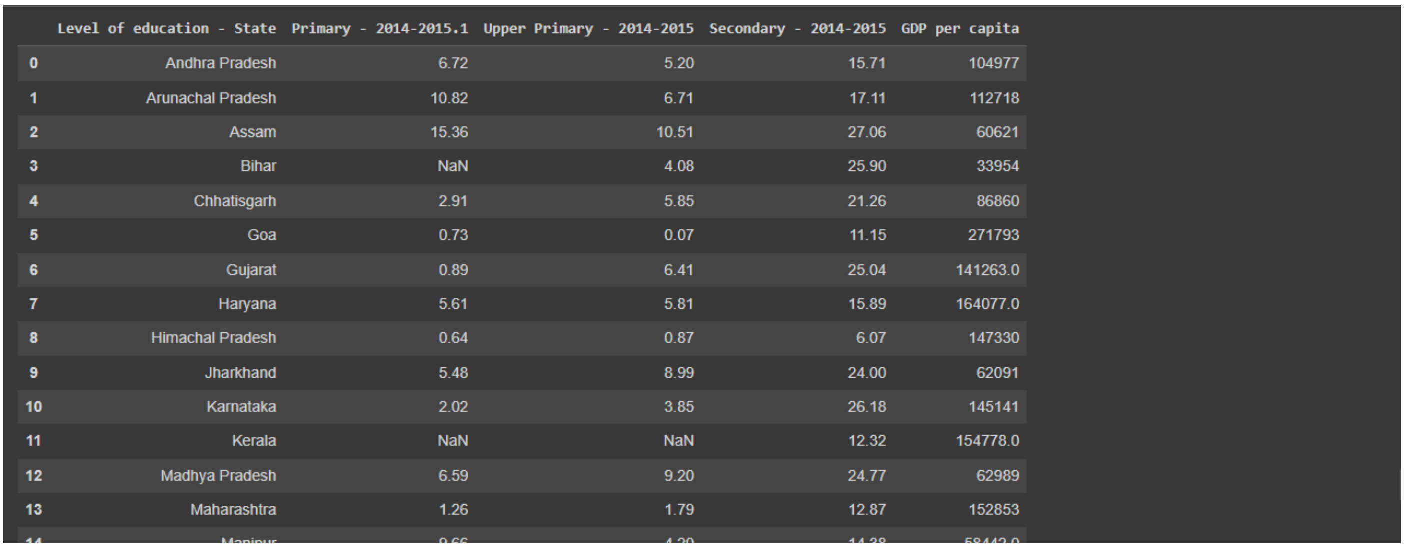
```
[ ]  # Dropping rows of data which we don not need like Union Territories and for which we don't have GDP per-capita available like West Bengal

     data2 =  data2.drop([0,5,7,8,9,14,18,26,35,36])
     data2 = data2.reset_index(drop = True)
     data2=data2.rename(columns={'Level of Education - State': 'State'})
```

```
[ ]  # Necessary transformation like resetting the index and renaming the column name for merging with another dataframe

     states_gdp_per_capita = gdp_per_capita.reset_index()
     states_gdp_per_capita=states_gdp_per_capita.rename(columns={'index':'State'})
```

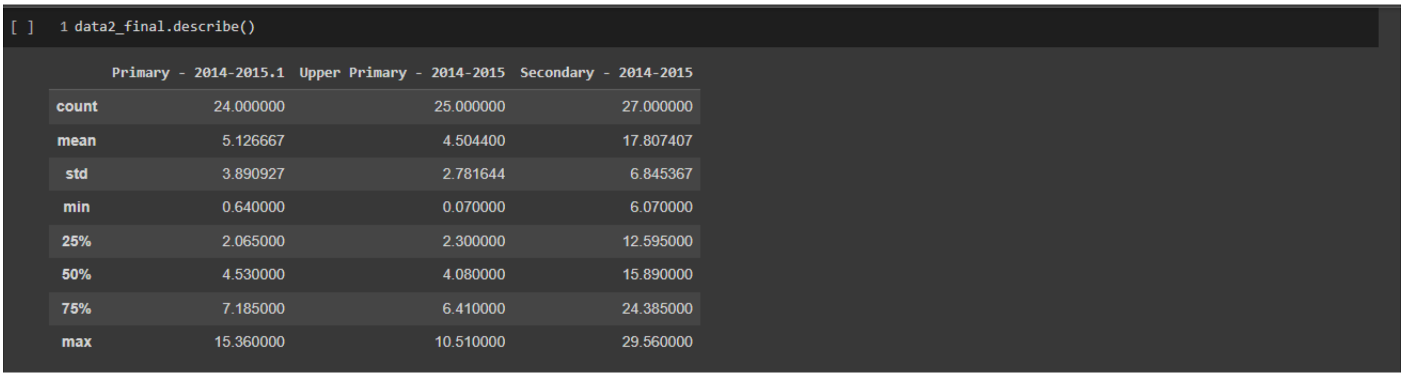
```
[ ]  # Merging the above dataframe with the GDP per-capita dataframe

     data2_final = pd.merge(data2,states_gdp_per_capita,how='left',on=['State'])
```

```
[ ]  data2_final = data2_final.rename(columns={'State':'Level of education - State'})
```

```
[ ]  # Final dataframe having the education level dropout rates for all the states and the GDP per capita

     data2_final
```



	Level of education - State	Primary - 2014-2015.1	Upper Primary - 2014-2015	Secondary - 2014-2015	GDP per capita
0	Andhra Pradesh	6.72	5.20	15.71	104977
1	Arunachal Pradesh	10.82	6.71	17.11	112718
2	Assam	15.36	10.51	27.06	60621
3	Bihar	NaN	4.08	25.90	33954
4	Chhatisgarh	2.91	5.85	21.26	86860
5	Goa	0.73	0.07	11.15	271793
6	Gujarat	0.89	6.41	25.04	141263.0
7	Haryana	5.61	5.81	15.89	164077.0
8	Himachal Pradesh	0.64	0.87	6.07	147330
9	Jharkhand	5.48	8.99	24.00	62091
10	Karnataka	2.02	3.85	26.18	145141
11	Kerala	NaN	NaN	12.32	154778.0
12	Madhya Pradesh	6.59	9.20	24.77	62989
13	Maharashtra	1.26	1.79	12.87	152853
14	Manipur	0.66	4.20	14.38	58442.0



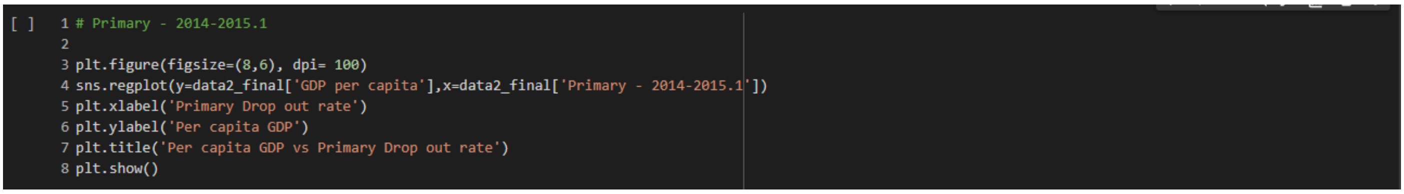
```
[ ]  data2_final.describe()
```

	Primary - 2014-2015.1	Upper Primary - 2014-2015	Secondary - 2014-2015
count	24.000000	25.000000	27.000000
mean	5.126667	4.504400	17.807407
std	3.890927	2.781644	6.845367
min	0.640000	0.070000	6.070000
25%	2.065000	2.300000	12.595000
50%	4.530000	4.080000	15.890000
75%	7.185000	6.410000	24.385000
max	15.360000	10.510000	29.560000

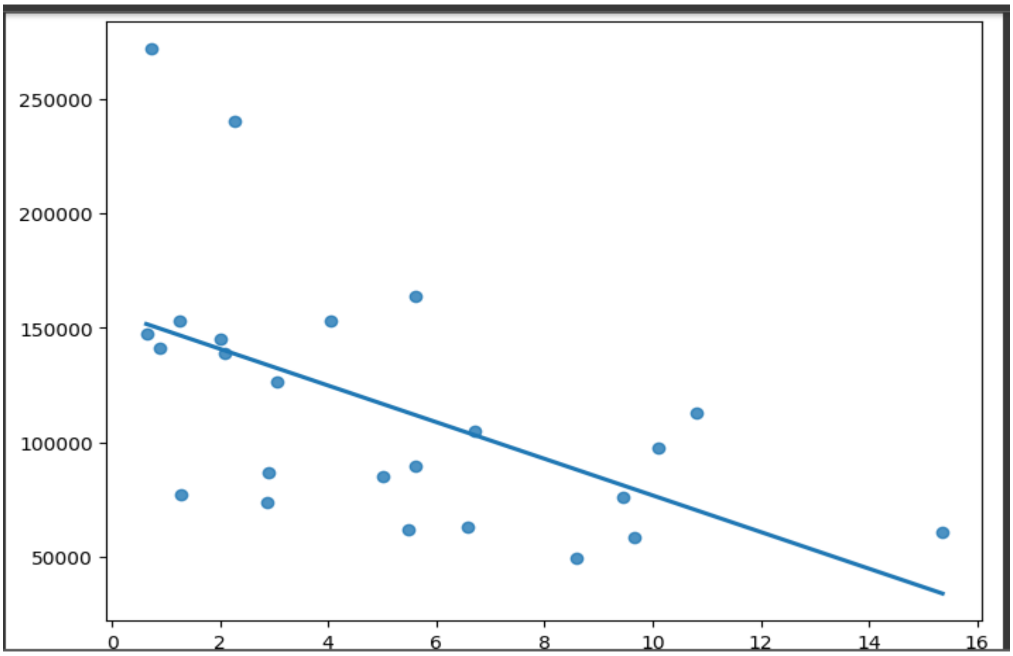
Drop out rate for Primary and Upper primary are comparable at approximately 5 and 4.5 % whereas the mean Drop out rate for Secondary is extremely large at 17.8%.

The minimum Drop out rate for Secondary is also high at 6%.

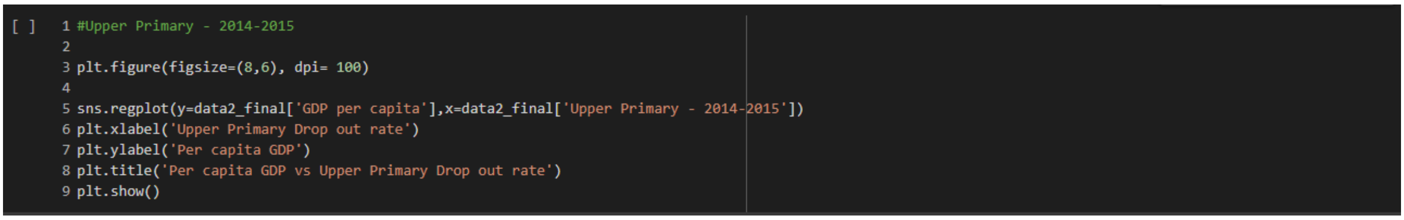
This means greater number of students are more likely to continue their Primary and Upper primary education but not Secondary education.



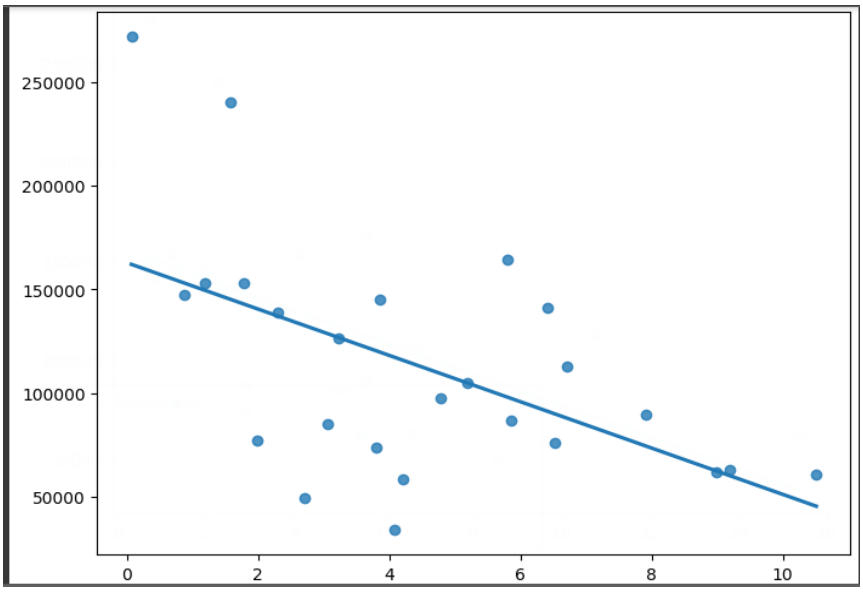
```
[ ]  # Primary - 2014-2015.1

     plt.figure(figsize=(8,6), dpi= 100)
     sns.regplot(y=data2_final['GDP per capita'],x=data2_final['Primary - 2014-2015.1'])
     plt.xlabel('Primary Drop out rate')
     plt.ylabel('Per capita GDP')
     plt.title('Per capita GDP vs Primary Drop out rate')
     plt.show()
```

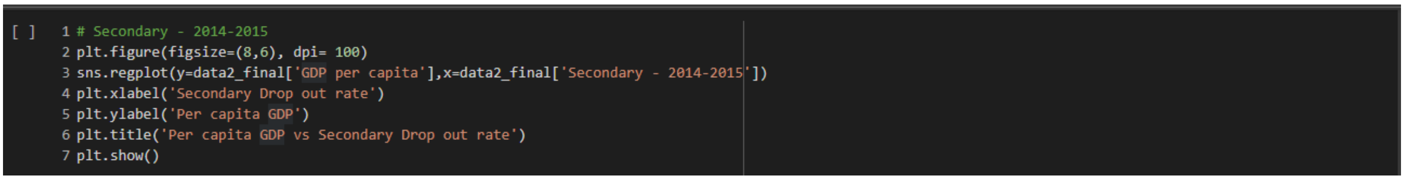


An almost linear relationship between GDP percapita and Primary Dropout rate for the year 2014-15. a linear relationship between GDP percapita and Upper Primary Dropout rate for the year 2014-15.

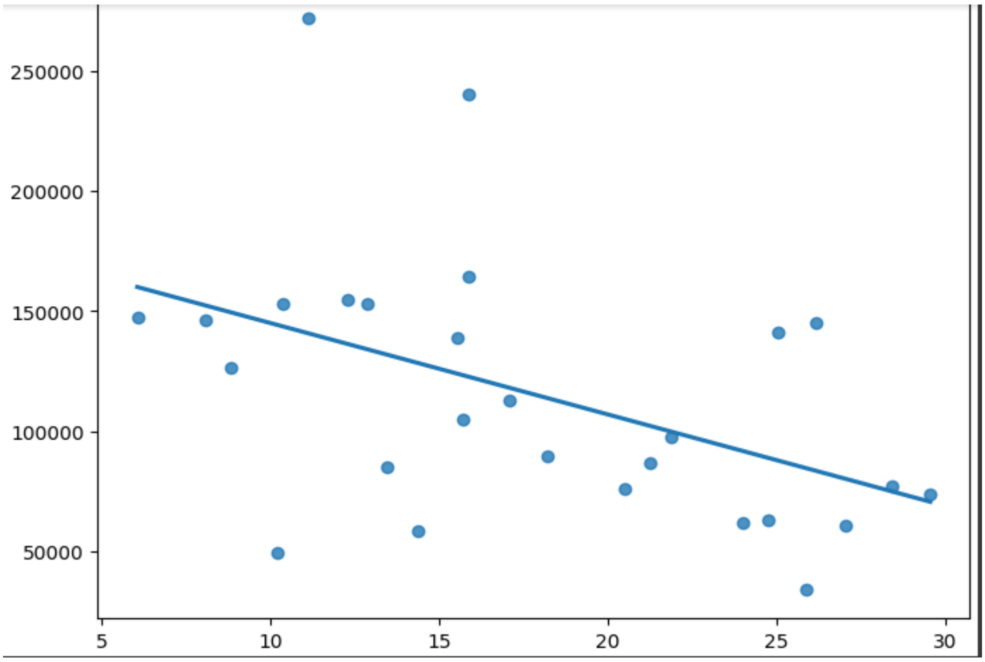


```
#Upper Primary - 2014-2015

plt.figure(figsize=(8,6), dpi= 100)

sns.regplot(y=data2_final['GDP per capita'],x=data2_final['Upper Primary - 2014-2015'])
plt.xlabel('Upper Primary Drop out rate')
plt.ylabel('Per capita GDP')
plt.title('Per capita GDP vs Upper Primary Drop out rate')
plt.show()
```





```
# Secondary - 2014-2015
plt.figure(figsize=(8,6), dpi= 100)
sns.regplot(y=data2_final['GDP per capita'],x=data2_final['Secondary - 2014-2015'])
plt.xlabel('Secondary Drop out rate')
plt.ylabel('Per capita GDP')
plt.title('Per capita GDP vs Secondary Drop out rate')
plt.show()
```



6.Conclusion –

It is evident that education level dropout rate has a direct correlation with GDP per capita. This is obvious as there are less number of skilled worker the quality of jobs available to them is less and hence they earn less when compared to their graduate counterparts. The states should investigate why the Secondary education dropout level is high and find a solution to this problem. Normally there are a lot of programs which focus on Primary and Upper education in India so students are less likely ot drop out from these levels.

7.References –

1. https://www.researchgate.net/publication/328631143_Analysis_of_Value_Added_Services_on_GDP_Growth_Rate_Using_Data_Mining_Techniques

2. https://www.researchgate.net/publication/369946578_SECTORAL_GDP_AND_TAX_REVENUE_AN_EMPIRICAL_EVIDENCE_WITH_PANEL_DATA_ANALYSIS

3. https://www.researchgate.net/publication/365377277_Impact_of_Urban_Mining_on_Energy_Efficiency_Evidence_from_China

4. https://www.researchgate.net/publication/369397667_A_causality-weighted_approach_for_prioritizing_mining_40_strategies_integrating_reliability-based_fuzzy_cognitive_map_and_hybrid_decision-making_methods_A_case_study_of_Nigerian_Mining_Sector

5.
https://www.researchgate.net/publication/355757924_Predicting_the_Contribution_of_Mining_Sector_to_the_Gross_Domestic_Product_GDP_Index_Utilizing_Heuristic_Approaches

6.
https://www.researchgate.net/publication/369876736_The_Economic_and_Environmental_Effects_of_Coal_Mining_South_Africa

7.
https://www.researchgate.net/publication/361538081_Comprehensive_Accounting_of_Resources_Environment_and_Economy_Integrating_Machine_Learning_and_Establishment_of_Green_GDP

8.
https://www.researchgate.net/publication/364667868_The_asymmetric_effects_of_fossil_fuel_dependency_on_the_carbon_intensity_of_well-being_A_US_state-level_analysis_1999-2017

9.
https://www.researchgate.net/publication/369226716_A_new_pattern_to_quantitatively_evaluate_the_value_of_ecosystem_services_in_the_large-scale_open-pit_coal_mining_area

10.
https://www.researchgate.net/publication/363732072_A_sustainability_model_for_globalized_mining_supply_chain