Name: Arjun Singh Baghel

Roll Number - DDS1910090

Email ID: arjusingh89baghel@gmail.com

IIITB Email ID: arjunsinghbaghel.dds10@iiitb.net

GDP Analysis – Assignmment

Part I – B:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import glob

# Data I-B

# Reading data from all .csv files once.

path = r'/Users/ArjunSinghBaghel/Documents/Data_Science/Assignment/ Part_I-
B_and_Part_II/DataSet_Part_I-B ' all_files = glob.glob(path + "/*.csv")

dataList = []

for filename in all_files:

   df = pd.read_csv(filename, index_col=None, header=0, encoding = 'unicode_escape')

   dataList.append(df)

mainDf = pd.concat(dataList, axis=0, ignore_index=True, sort=True)

mainDf['State']=sorted(['Andhra Pradesh','Arunachal
Pradesh','Assam','Bihar','Chattisgarh','Goa','Gujarat','Haryana','Himachal
Pradesh','Jharkhand','Karnataka','Kerala','Madhyapradesh','Maharastra','Manipur','Meghalaya','
Mizoram','Nagaland','Odisha','Punjab','Rajasthan','Sikkim','Tamil
Nadu','Telangana','Tripura','Uttarpradesh','Uttarakhand']*33)

MainDf
```

	2011-12	2012-13	2013-14	2014-15	2015-16	2016-17	Item	S.No.	State
0	11910650.0	13572561.0	14418664.0	15044394.0	NaN	NaN	Agriculture, forestry and fishing	1	Andhra Pradesh
1	7346942.0	8389344.0	8675414.0	7673441.0	NaN	NaN	Crops	1.1	Andhra Pradesh
2	3113617.0	3523365.0	3962174.0	5356257.0	NaN	NaN	Livestock	1.2	Andhra Pradesh
3	1416986.0	1622443.0	1738606.0	1956660.0	NaN	NaN	Forestry and logging	1.3	Andhra Pradesh
4	33105.0	37408.0	42470.0	58036.0	NaN	NaN	Fishing and aquaculture	1.4	Andhra Pradesh
5	1841530.0	3395874.0	3716758.0	4069385.0	NaN	NaN	Mining and quarrying	2	Andhra Pradesh
6	13752181.0	16968435.0	18135422.0	19113780.0	NaN	NaN	Primary	Total	Andhra Pradesh
7	6666606.0	5627437.0	6015398.0	6552580.0	NaN	NaN	Manufacturing	3	Andhra Pradesh
8	763271.0	1059921.0	979452.0	1122888.0	NaN	NaN	Electricity, gas, water supply & other utility...	4	Andhra Pradesh
9	4359171.0	4568112.0	4965979.0	5353326.0	NaN	NaN	Construction	5	Andhra Pradesh
10	11789049.0	11255470.0	11960829.0	13028794.0	NaN	NaN	Secondary	Total	Andhra Pradesh
11	4374625.0	5241569.0	6201013.0	7297290.0	NaN	NaN	Trade, repair, hotels and restaurants	6	Andhra Pradesh
12	4097894.0	4943212.0	5878264.0	6942748.0	NaN	NaN	Trade & repair services	6.1	Andhra Pradesh
13	276731.0	298357.0	322749.0	354543.0	NaN	NaN	Hotels & restaurants	6.2	Andhra Pradesh
14	2312809.0	2748068.0	3300711.0	3814461.0	NaN	NaN	Transport, storage, communication & services r...	7	Andhra Pradesh
15	240962.0	336900.0	401763.0	464638.0	NaN	NaN	Railways	7.1	Andhra Pradesh
16	1410655.0	1636644.0	1906004.0	2121206.0	NaN	NaN	Road transport	7.2	Andhra Pradesh
17	0.0	0.0	0.0	0.0	NaN	NaN	Water transport	7.3	Andhra Pradesh
18	6698.0	12715.0	10561.0	13469.0	NaN	NaN	Air transport	7.4	Andhra Pradesh
19	31956.0	37192.0	43300.0	47609.0	NaN	NaN	Services incidental to transport	7.5	Andhra Pradesh

Main Data Frame

 # Get data for 2014-15

```
columns = ['2014-15','Item', 'S.No.','State']
subMainDf = pd.DataFrame(mainDf, columns=columns)
SubMainDf
```

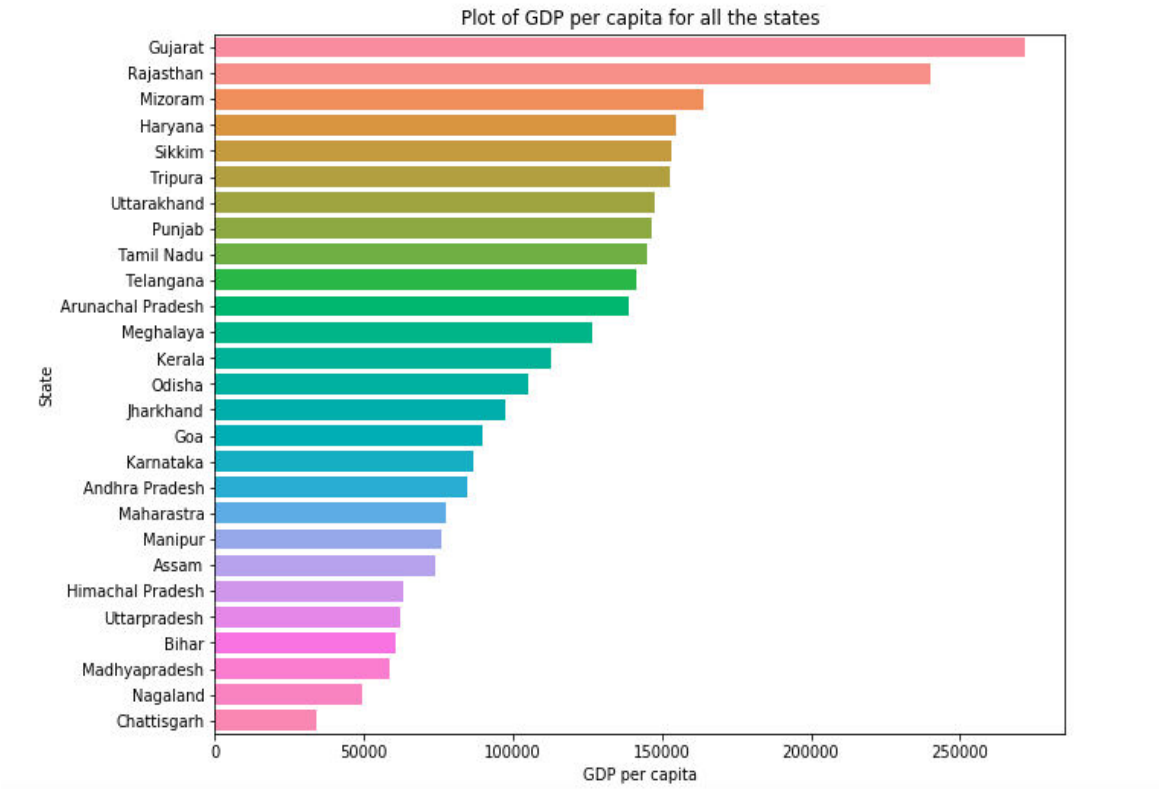
]:

	2014-15	Item	S.No.	State
0	15044394.0	Agriculture, forestry and fishing	1	Andhra Pradesh
1	7673441.0	Crops	1.1	Andhra Pradesh
2	5356257.0	Livestock	1.2	Andhra Pradesh
3	1956660.0	Forestry and logging	1.3	Andhra Pradesh
4	58036.0	Fishing and aquaculture	1.4	Andhra Pradesh
5	4069385.0	Mining and quarrying	2	Andhra Pradesh
6	19113780.0	Primary	Total	Andhra Pradesh
7	6552580.0	Manufacturing	3	Andhra Pradesh
8	1122888.0	Electricity, gas, water supply & other utility...	4	Andhra Pradesh
9	5353326.0	Construction	5	Andhra Pradesh
10	13028794.0	Secondary	Total	Andhra Pradesh
11	7297290.0	Trade, repair, hotels and restaurants	6	Andhra Pradesh
12	6942748.0	Trade & repair services	6.1	Andhra Pradesh
13	354543.0	Hotels & restaurants	6.2	Andhra Pradesh
14	3814461.0	Transport, storage, communication & services r...	7	Andhra Pradesh
15	464638.0	Railways	7.1	Andhra Pradesh
16	2121206.0	Road transport	7.2	Andhra Pradesh
17	0.0	Water transport	7.3	Andhra Pradesh
18	13469.0	Air transport	7.4	Andhra Pradesh
19	47609.0	Services incidental to transport	7.5	Andhra Pradesh

2014-15 Data Frame

 # Plot the GDP per capita for all the states.

```
gdpPerCapitalAllStates = subMainDf['Item'] == "Per Capita GSDP (Rs.)"

gdpPerCapitalAllStatesDf =subMainDf[gdpPerCapitalAllStates] # Getting Per Capita GSDP (Rs.)

gdpPerCapitalAllStatesDf = gdpPerCapitalAllStatesDf.sort_values(by='2014-15',
ascending=False) #Sorting

plt.figure(figsize=(10,8))

sns.barplot(x=gdpPerCapitalAllStatesDf['2014-15'], y=gdpPerCapitalAllStatesDf.State)

plt.xlabel('GDP per capita')

plt.title('Plot of GDP per capita for all the states')

plt.show()
```

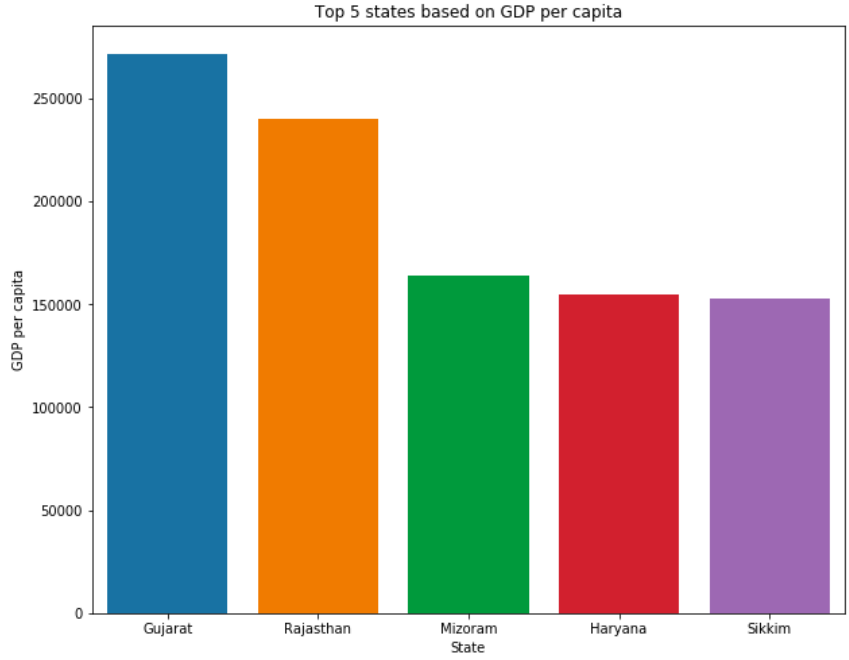


Plot of GDP per capital for all the states

 # Identify the top-5 states based on GDP per capita.

```
top5 = gdpPerCapitalAllStatesDf.head(5)
print(top5)
plt.figure(figsize=(10,8))
sns.barplot(x=top5.State, y=top5['2014-15'])
plt.title('Top 5 states based on GDP per capita')
plt.ylabel('GDP per capita')
plt.show()
```

	2014-15	Item	S.No.	State
230	271793.0	Per Capita GSDP (Rs.)	17	Gujarat
692	240274.0	Per Capita GSDP (Rs.)	17	Rajasthan
560	164077.0	Per Capita GSDP (Rs.)	17	Mizoram
263	154778.0	Per Capita GSDP (Rs.)	17	Haryana
725	153076.0	Per Capita GSDP (Rs.)	17	Sikkim

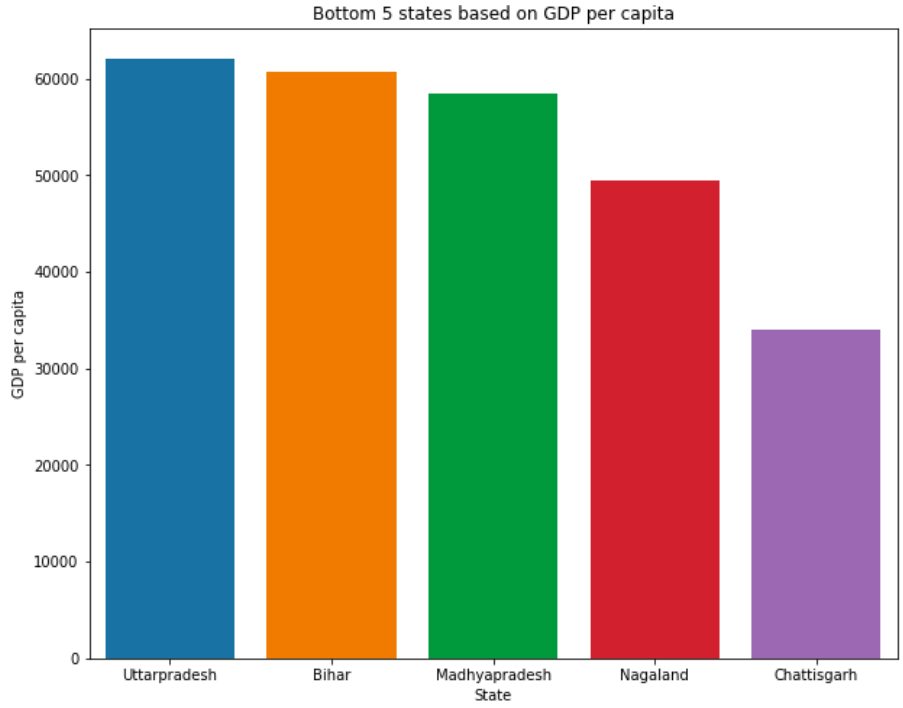


Plot of top-5 states based on GDP per capital

 # Identify the bottom-5 states based on GDP per capita.

```
bottom5 = gdpPerCapitalAllStatesDf.tail(5)
print(bottom5)
plt.figure(figsize=(10,8))
sns.barplot(x=bottom5.State, y=bottom5['2014-15'])
plt.title('Bottom 5 states based on GDP per capita')
plt.ylabel('GDP per capita')
plt.show()
```

	2014-15	Item	S.No.	State
890	62091.0	Per Capita GSDP (Rs.)	17	Uttarpradesh
131	60621.0	Per Capita GSDP (Rs.)	17	Bihar
428	58442.0	Per Capita GSDP (Rs.)	17	Madhyapradesh
593	49450.0	Per Capita GSDP (Rs.)	17	Nagaland
164	33954.0	Per Capita GSDP (Rs.)	17	Chattisgarh



Plot of bottom-5 states based on GDP per capital

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

```
#  'subMainDf' is only contain 2014-15 data.
primary = subMainDf['Item'] == "Primary"
secondary = subMainDf['Item'] == "Secondary"
tertiary = subMainDf['Item'] == "Tertiary"
gdp = subMainDf['Item'] == "Gross State Domestic Product"
perCapitaGSDP = subMainDf['Item'] == 'Per Capita GSDP (Rs.)'
new_Df =subMainDf[primary | secondary | tertiary | gdp | perCapitaGSDP]
new_Df
```

	2014-15	Item	S.No.	State
6	19113780.0	Primary	Total	Andhra Pradesh
10	13028794.0	Secondary	Total	Andhra Pradesh
26	26015812.0	Tertiary	Total	Andhra Pradesh
30	61219447.0	Gross State Domestic Product	15	Andhra Pradesh
32	84837.0	Per Capita GSDP (Rs.)	17	Andhra Pradesh
39	9133354.0	Primary	Total	Arunachal Pradesh
43	9924001.0	Secondary	Total	Arunachal Pradesh
59	28471410.0	Tertiary	Total	Arunachal Pradesh
63	51117765.0	Gross State Domestic Product	15	Arunachal Pradesh
65	139035.0	Per Capita GSDP (Rs.)	17	Arunachal Pradesh
72	9009306.0	Primary	Total	Assam
76	8989693.0	Secondary	Total	Assam
92	12256258.0	Tertiary	Total	Assam
96	32197092.0	Gross State Domestic Product	15	Assam
98	73979.0	Per Capita GSDP (Rs.)	17	Assam
105	5326697.0	Primary	Total	Bihar
109	4033091.0	Secondary	Total	Bihar
125	9307109.0	Tertiary	Total	Bihar
129	19809800.0	Gross State Domestic Product	15	Bihar
131	60621.0	Per Capita GSDP (Rs.)	17	Bihar

Data Frame of Primary Secondary Tertiary GDP PerCapitaGSDP for 2014-15

 # Set 'State' as a index and drop 'S.No' column

```
new_Df = new_Df.set_index('State')
new_Df.drop('S.No.',axis=1)
```

State	2014-15	Item
Andhra Pradesh	19113780.0	Primary
Andhra Pradesh	13028794.0	Secondary
Andhra Pradesh	26015812.0	Tertiary
Andhra Pradesh	61219447.0	Gross State Domestic Product
Andhra Pradesh	84837.0	Per Capita GSDP (Rs.)
Arunachal Pradesh	9133354.0	Primary
Arunachal Pradesh	9924001.0	Secondary
Arunachal Pradesh	28471410.0	Tertiary
Arunachal Pradesh	51117765.0	Gross State Domestic Product
Arunachal Pradesh	139035.0	Per Capita GSDP (Rs.)
Assam	9009306.0	Primary
Assam	8989693.0	Secondary
Assam	12256258.0	Tertiary
Assam	32197092.0	Gross State Domestic Product
Assam	73979.0	Per Capita GSDP (Rs.)

 # Reshape the dataFrame

```
new_Df=pd.DataFrame(new_Df['2014-15'].values.reshape(27,5),columns=['Primary','Secondary','Tertiary','Gross State Domestic Product', 'Per Capita GSDP (Rs.)'],index=new_Df.index.unique())
```

```
new_Df
```

State	Primary	Secondary	Tertiary	Gross State Domestic Product	Per Capita GSDP (Rs.)
Andhra Pradesh	19113780.0	13028794.0	26015812.0	61219447.0	84837.0
Arunachal Pradesh	9133354.0	9924001.0	28471410.0	51117765.0	139035.0
Assam	9009306.0	8989693.0	12256258.0	32197092.0	73979.0
Bihar	5326697.0	4033091.0	9307109.0	19809800.0	60621.0
Chattisgarh	8019997.0	5984896.0	22179969.0	37391988.0	33954.0
Goa	616178.0	212361.0	992956.0	1841424.0	89607.0

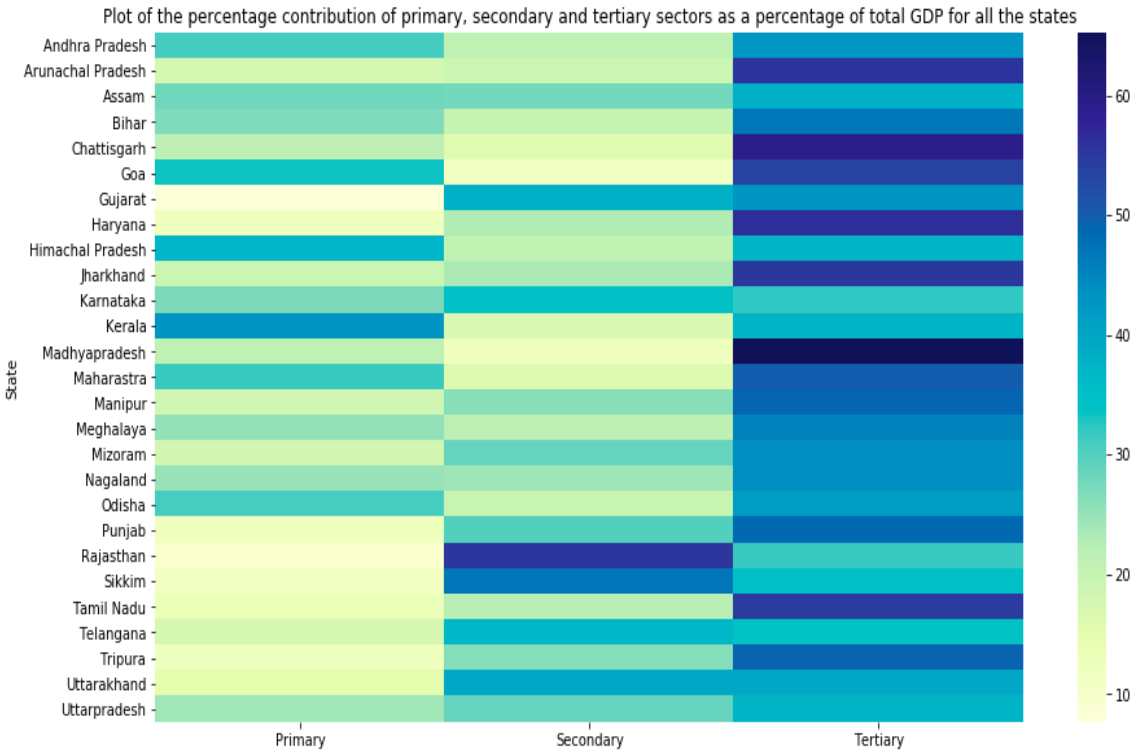
 # Calculate percentage contribution of primary, secondary and tertiary sectors

```
pstDf = new_Df # Cipping new_Df for future use
pstDf['Primary'] = (pstDf['Primary'] / pstDf['Gross State Domestic Product']) * 100
pstDf['Secondary'] = (pstDf['Secondary'] / pstDf['Gross State Domestic Product']) * 100
pstDf['Tertiary'] = (new_Df['Tertiary'] / pstDf['Gross State Domestic Product']) * 100
pstDf = pstDf.drop(['Gross State Domestic Product','Per Capita GSDP (Rs.)'], axis=1)
PstDf
```

State	Primary	Secondary	Tertiary
Andhra Pradesh	31.221746	21.282116	42.495993
Arunachal Pradesh	17.867280	19.413996	55.697682
Assam	27.981738	27.920823	38.066351
Bihar	26.889201	20.359070	46.982347
Chattisgarh	21.448437	16.005825	59.317437
Goa	33.462038	11.532434	53.923268
Gujarat	7.681649	38.085628	42.778382
Haryana	12.337288	22.946744	56.413780
Himachal Pradesh	37.042942	20.840810	37.589312
Jharkhand	19.516529	23.363984	55.160550
Karnataka	27.239629	35.061805	32.295173
Kerala	42.774946	17.152064	37.696846

 # Plot heatmap

```
plt.figure(figsize=(15,8))

sns.heatmap(pstDf, cmap="YlGnBu")

plt.title('Plot of the percentage contribution of primary, secondary and tertiary sectors as a percentage of total GDP for all the states')

plt.show()
```



Plot of the percentage contribution of Primary, secondary and tertiary sectors as a percentage of total GDP for all the states

 # Categorise the states into four categories based on GDP per capita (C1, C2, C3, C4 - C1 would have the highest per capita GDP, C4 the lowest). The quantile values are (0.20,0.5, 0.85, 1), i.e. the states lying between the 85th and the 100th percentile are in C1, those between 50th and 85th percentile are in C2 and so on.

```
categoryDF = pd.DataFrame(new_Df.loc[:,'Per Capita GSDP (Rs.)'])
CategoryDF
```

State	Per Capita GSDP (Rs.)
Andhra Pradesh	84837.0
Arunachal Pradesh	139035.0
Assam	73979.0
Bihar	60621.0
Chattisgarh	33954.0
Goa	89607.0
Gujarat	271793.0
Haryana	154778.0
Himachal Pradesh	62989.0
Jharkhand	97687.0
Karnataka	86860.0
Kerala	112718.0
Madhyapradesh	58442.0

 # Calculating quantile (0.20,0.5, 0.85, 1) on 'Per Capita GSDP (Rs.)'

```
categoryDF['Per Capita GSDP (Rs.)'].quantile([0.20,0.5, 0.85, 1])
```

```
0.20     65187.0
0.50    104977.0
0.85    153246.2
1.00    271793.0
Name: Per Capita GSDP (Rs.), dtype: float64
```

 # Adding 'Categorise' coloumn based on quantile

```
categoryGroupList = []
for perCapitaGSDPValue in categoryDF['Per Capita GSDP (Rs.)']:
    if perCapitaGSDPValue > 153246.2 and perCapitaGSDPValue <= 271793.0:
        categoryGroupList.append('c1')
    if perCapitaGSDPValue > 104977.0 and perCapitaGSDPValue <= 153246.2:
        categoryGroupList.append('c2')
    if perCapitaGSDPValue > 65187.0 and perCapitaGSDPValue <= 104977.0:
        categoryGroupList.append('c3')
    if perCapitaGSDPValue <=65187.0:
        categoryGroupList.append('c4')
categoryDF['Category'] = categoryGroupList
CategoryDF
```

State	Per Capita GSDP (Rs.)	Category
Andhra Pradesh	84837.0	c3
Arunachal Pradesh	139035.0	c2
Assam	73979.0	c3
Bihar	60621.0	c4
Chattisgarh	33954.0	c4
Goa	89607.0	c3
Gujarat	271793.0	c1
Haryana	154778.0	c1
Himachal Pradesh	62989.0	c4
Jharkhand	97687.0	c3
Karnataka	86860.0	c3
Kerala	112718.0	c2

 # Mapping Category to the main dataframe

```
subSectorsDF  = mainDf

subSectorsDF = subSectorsDF.set_index('State')

subSectorsDF['Category']=
subSectorsDF.index.map(dict(zip(categoryDF.index,categoryDF['Category'])))

 subSectorsDF = subSectorsDF.drop(['2011-12','2012-13','2013-14','2015-16','2016-
17','S.No.'],axis=1)

subSectorsDF = subSectorsDF[subSectorsDF.Item != 'Primary']

subSectorsDF = subSectorsDF[subSectorsDF.Item != 'Secondary']

subSectorsDF = subSectorsDF[subSectorsDF.Item != 'Tertiary']

subSectorsDF
```

State	2014-15	Item	Category
Andhra Pradesh	15044394.0	Agriculture, forestry and fishing	c3
Andhra Pradesh	7673441.0	Crops	c3
Andhra Pradesh	5356257.0	Livestock	c3
Andhra Pradesh	1956660.0	Forestry and logging	c3
Andhra Pradesh	58036.0	Fishing and aquaculture	c3
Andhra Pradesh	4069385.0	Mining and quarrying	c3
Andhra Pradesh	6552580.0	Manufacturing	c3
Andhra Pradesh	1122888.0	Electricity, gas, water supply & other utility...	c3
Andhra Pradesh	5353326.0	Construction	c3
Andhra Pradesh	7297290.0	Trade, repair, hotels and restaurants	c3
Andhra Pradesh	6942748.0	Trade & repair services	c3
Andhra Pradesh	354543.0	Hotels & restaurants	c3

 # Calculate group wise sum

new_subSectorsDF = subSectorsDF[subSectorsDF.loc[:,'Item']=='Gross State Domestic Product'].groupby('Category').sum()

new_subSectorsDF

	2014-15
Category	
c1	101930677.0
c2	586380452.0
c3	177966387.0
c4	233252066.0

 # Map the data according to category

new_subSectorsDF = subSectorsDF['Category'].map(dict(zip(['c1','c2','c3','c4'],new_subSectorsDF['2014-15'].values)))

new_subSectorsDF

```
State
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
Andhra Pradesh    177966387.0
```

 # Add 'Contribution %' column

subSectorsDF['Contribution %'] = (subSectorsDF['2014-15'] / new_subSectorsDF) *100

SubSectorsDF

State	2014-15	Item	Category	Contribution %
Andhra Pradesh	15044394.0	Agriculture, forestry and fishing	c3	8.453503
Andhra Pradesh	7673441.0	Crops	c3	4.311736
Andhra Pradesh	5356257.0	Livestock	c3	3.009701
Andhra Pradesh	1956660.0	Forestry and logging	c3	1.099455
Andhra Pradesh	58036.0	Fishing and aquaculture	c3	0.032611
Andhra Pradesh	4069385.0	Mining and quarrying	c3	2.286603
Andhra Pradesh	6552580.0	Manufacturing	c3	3.681920
Andhra Pradesh	1122888.0	Electricity, gas, water supply & other utility...	c3	0.630955
Andhra Pradesh	5353326.0	Construction	c3	3.008055
Andhra Pradesh	7297290.0	Trade, repair, hotels and restaurants	c3	4.100375
Andhra Pradesh	6942748.0	Trade & repair services	c3	3.901157
Andhra Pradesh	354543.0	Hotels & restaurants	c3	0.199219

 # Filter only ‘S.No.’ is 'Int' value

```
item1 = subSectorsDF['Item'] == "Agriculture, forestry and fishing"

item2 = subSectorsDF['Item'] == "Mining and quarrying"

item3 = subSectorsDF['Item'] == "Manufacturing"

item4 = subSectorsDF['Item'] == "Electricity, gas, water supply & other utility services"

item5 = subSectorsDF['Item'] == "Construction"

item6 = subSectorsDF['Item'] == "Trade, repair, hotels and restaurants"

item7 = subSectorsDF['Item'] == "Transport, storage, communication & services related to broadcasting"

item8 = subSectorsDF['Item'] == "Financial services"

item9 = subSectorsDF['Item'] == "Real estate, ownership of dwelling & professional services"

item10 = subSectorsDF['Item'] == "Public administration"

item11 = subSectorsDF['Item'] == "Other services"

subSectorsDF = subSectorsDF[item1 | item2 | item3 | item4 | item5 | item6 | item7 | item8 | item9 | item10 | item11]

subSectorsDF
```

 # Set 'Category' and 'Item' both as an index

subSectorsDF = subSectorsDF.set_index(['Category', 'Item'])

SubSectorsDF

]:

Category	Item	2014-15	Contribution %
c3	Agriculture, forestry and fishing	15044394.0	8.453503
	Mining and quarrying	4069385.0	2.286603
	Manufacturing	6552580.0	3.681920
	Electricity, gas, water supply & other utility services	1122888.0	0.630955
	Construction	5353326.0	3.008055
	Trade, repair, hotels and restaurants	7297290.0	4.100375
	Transport, storage, communication & services related to broadcasting	3814461.0	2.143360
	Financial services	1827413.0	1.026830
	Real estate, ownership of dwelling & professional services	6451997.0	3.625402
	Public administration	2460364.0	1.382488
	Other services	4164287.0	2.339929
c2	Agriculture, forestry and fishing	7591501.0	1.294637
	Mining and quarrying	1541853.0	0.262944
	Manufacturing	6353711.0	1.083548
	Electricity, gas, water supply & other utility services	716266.0	0.122150
	Construction	2854024.0	0.486719
	Trade, repair, hotels and restaurants	6494607.0	1.107576
	Transport, storage, communication & services related to broadcasting	3604741.0	0.614744
	Financial services	3023729.0	0.515660
	Real estate, ownership of dwelling & professional services	9478839.0	1.616500
	Public administration	1711265.0	0.291835
	Other services	4158229.0	0.709135
c3	Agriculture, forestry and fishing	6422978.0	3.609096

 # Caculate sum groupby both 'Category' &'Item'

new_subSectorsDF = subSectorsDF

new_subSectorsDF = new_subSectorsDF.groupby(['Category', 'Item']).sum()

new_subSectorsDF

Category	Item	2014-15	Contribution %
c1	Agriculture, forestry and fishing	14391809.0	14.119213
	Construction	11264451.0	11.051090
	Electricity, gas, water supply & other utility services	2000998.0	1.963097
	Financial services	3936489.0	3.861928
	Manufacturing	13758793.0	13.498187
	Mining and quarrying	588961.0	0.577805
	Other services	8059922.0	7.907258
	Public administration	3571292.0	3.503648
	Real estate, ownership of dwelling & professional services	14740245.0	14.461049
	Trade, repair, hotels and restaurants	13995159.0	13.730076
	Transport, storage, communication & services related to broadcasting	6818560.0	6.689409
c2	Agriculture, forestry and fishing	75209022.0	12.825977
	Construction	40653562.0	6.932967
	Electricity, gas, water supply & other utility services	13823378.0	2.357408
	Financial services	36296642.0	6.189947
	Manufacturing	109196530.0	18.622130

 # sort values according to both 'Category' & 'Contribution %'

```
new_subSectorsDF.sort_values(['Category','Contribution %'],ascending=[True,False],inplace=True)
```

```
new_subSectorsDF
```

Category	Item	2014-15	Contribution %
c1	Real estate, ownership of dwelling & professional services	14740245.0	14.461049
	Agriculture, forestry and fishing	14391809.0	14.119213
	Trade, repair, hotels and restaurants	13995159.0	13.730076
	Manufacturing	13758793.0	13.498187
	Construction	11264451.0	11.051090
	Other services	8059922.0	7.907258
	Transport, storage, communication & services related to broadcasting	6818560.0	6.689409
	Financial services	3936489.0	3.861928
	Public administration	3571292.0	3.503648
	Electricity, gas, water supply & other utility services	2000998.0	1.963097
	Mining and quarrying	588961.0	0.577805
c2	Manufacturing	109196530.0	18.622130
	Real estate, ownership of dwelling & professional services	92121446.0	15.710184
	Agriculture, forestry and fishing	75209022.0	12.825977
	Trade, repair, hotels and restaurants	61238862.0	10.443537

 # sub-sectors contribute to approx. 80% of the GDP'

Category C1

```
top_C1_index = sum(new_subSectorsDF.loc['c1']['Contribution %'].cumsum(axis = 0) < 80)+1
top_C1 =new_subSectorsDF.loc['c1'].iloc[:top_C1_index,1]
top_C1.name ='C1'
top_C1
```

Category C2

```
top_C2_index = sum(new_subSectorsDF.loc['c2']['Contribution %'].cumsum(axis = 0) < 80)+1
top_C2 =new_subSectorsDF.loc['c2'].iloc[:top_C2_index,1]
top_C2.name ='C2'
top_C2
```

Category C3

```
top_C3_index = sum(new_subSectorsDF.loc['c3']['Contribution %'].cumsum(axis = 0) < 80)+1
top_C3 =new_subSectorsDF.loc['c3'].iloc[:top_C3_index,1]
top_C3.name ='C3'
top_C3
```

Category C4

```
top_C4_index = sum(new_subSectorsDF.loc['c4']['Contribution %'].cumsum(axis = 0) < 80)+1
top_C4 =new_subSectorsDF.loc['c4'].iloc[:top_C4_index,1]
top_C4.name ='C4'
top_C4
```

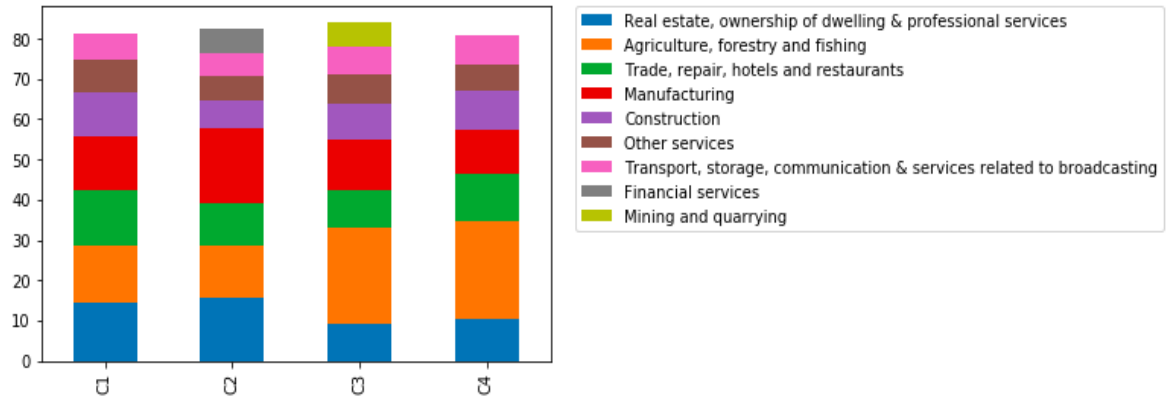
 # Concat all categories C1, C2, C3, C4

```
topResultDf = pd.concat([top_C1,top_C2,top_C3,top_C4],axis=1,sort=False)
TopResultDf
```

	C1	C2	C3	C4
Real estate, ownership of dwelling & professional services	14.461049	15.710184	9.116395	10.365410
Agriculture, forestry and fishing	14.119213	12.825977	23.727451	24.323490
Trade, repair, hotels and restaurants	13.730076	10.443537	9.690206	11.783216
Manufacturing	13.498187	18.622130	12.383423	10.712459
Construction	11.051090	6.932967	8.826341	9.764521
Other services	7.907258	6.182617	7.201032	6.799089
Transport, storage, communication & services related to broadcasting	6.689409	5.755616	6.872918	6.941761
Financial services	NaN	6.189947	NaN	NaN
Mining and quarrying	NaN	NaN	6.088662	NaN

 # Plot the contribution of the sub-sectors as a percentage of the GSDP of each category.

```
plt.figure(num=None,figsize=(100,2000),dpi=60,facecolor='w',edgecolor='k')
ax=topResultDf.T.plot(kind='bar',stacked=True)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```



 # Data II

```
drop_out_rate_df = pd.read_csv("/Users/ArjunSinghBaghel/Documents/Data_Science/Assignment/Part_I-B_and_Part_II/DataSet_Part_II/rs_session243_au570_1.1.csv ")
drop_out_rate_df
```

	Sl. No.	Level of Education - State	Primary - 2012-2013	Primary - 2014-2015	Primary - 2014-2015.1	Upper Primary - 2012-2013	Upper Primary - 2013-2014	Upper Primary - 2014-2015	Secondary - 2012-2013	Secondary - 2013-2014	Secondary - 2014-2015	Senior Secondary - 2012-2013	Senior Secondary - 2013-2014	Senior Secondary - 2014-2015
0	1	A & N Islands	0.68	1.21	0.51	1.23	0.51	1.69	5.56	7.20	9.870	14.14	15.87	16.93
1	2	Andhra Pradesh	3.18	4.35	6.72	3.36	3.78	5.20	12.72	12.65	15.710	0.35	11.79	NaN
2	3	Arunachal Pradesh	15.16	10.89	10.82	7.47	5.59	6.71	12.93	14.49	17.110	5.11	17.07	18.42
3	4	Assam	6.24	7.44	15.36	7.20	7.05	10.51	26.77	30.43	27.060	4.69	7.24	NaN
4	5	Bihar	NaN	2.09	NaN	NaN	2.98	4.08	30.14	25.33	25.900	NaN	NaN	NaN
5	6	Chandigarh	NaN	NaN	NaN	0.10	1.08	0.44	NaN	NaN	NaN	13.65	11.28	10.55
6	7	Chhatisgarh	4.14	1.42	2.91	5.42	3.80	5.85	14.86	23.41	21.260	NaN	NaN	2.76
7	8	Dadra & Nagar Haveli	NaN	1.57	1.47	3.38	4.13	4.02	24.41	28.59	16.770	9.65	6.24	9.47

 # Set 'Level of Education – State' as index

```
dropIndexs = drop_out_rate_df.index[[0,-1]]

drop_out_rate_df = drop_out_rate_df.drop(dropIndexs)

drop_out_rate_df = drop_out_rate_df.set_index('Level of Education - State')

drop_out_rate_df
```

 # Maped drop_out_rate_df & categoryDF

```
drop_out_rate_df['per capita GDP']=
drop_out_rate_df.index.map(dict(zip(categoryDF.index,categoryDF['Per Capita GSDP (Rs.)'])))

drop_out_rate_df
```

 # Select Primary - 2014-2015, Upper Primary - 2014-2015, Secondary - 2014-2015 & per capita GDP

```
drop_out_rate_df = drop_out_rate_df[['Primary - 2014-2015','Upper Primary - 2014-2015',
'Secondary - 2014-2015', 'per capita GDP']]

drop_out_rate_df
```

Level of Education - State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	per capita GDP
Andhra Pradesh	4.35	5.20	15.710	84837.0
Arunachal Pradesh	10.89	6.71	17.110	139035.0
Assam	7.44	10.51	27.060	73979.0
Bihar	2.09	4.08	25.900	60621.0
Chandigarh	NaN	0.44	NaN	NaN
Chhatisgarh	1.42	5.85	21.260	NaN
Dadra & Nagar Haveli	1.57	4.02	16.770	NaN
Daman & Diu	0.85	3.11	32.270	NaN
Delhi	NaN	0.76	11.810	NaN
Goa	0.20	0.07	11.150	89607.0
Gujarat	0.76	6.41	25.040	271793.0

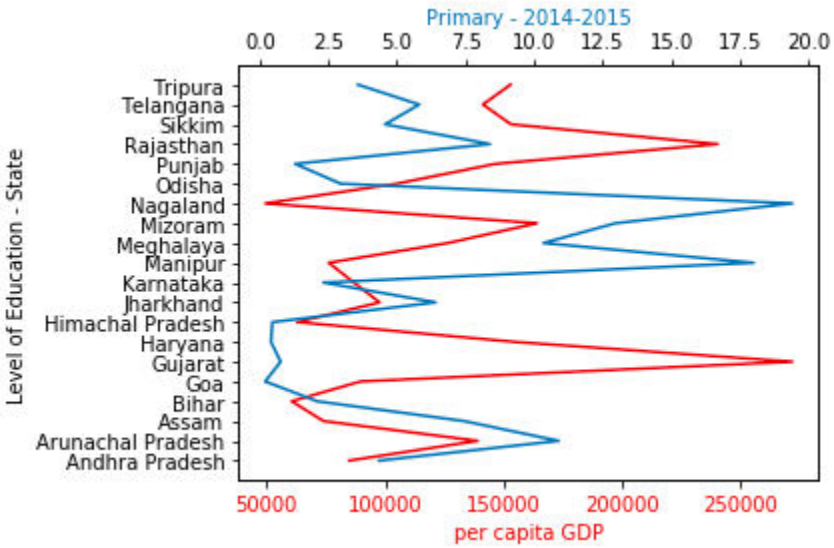
 # drop all null values

```
drop_out_rate_df = drop_out_rate_df.dropna()
drop_out_rate_df
```

 # Plot for primary, upper primary and secondary education system with Per capita GDP of different states

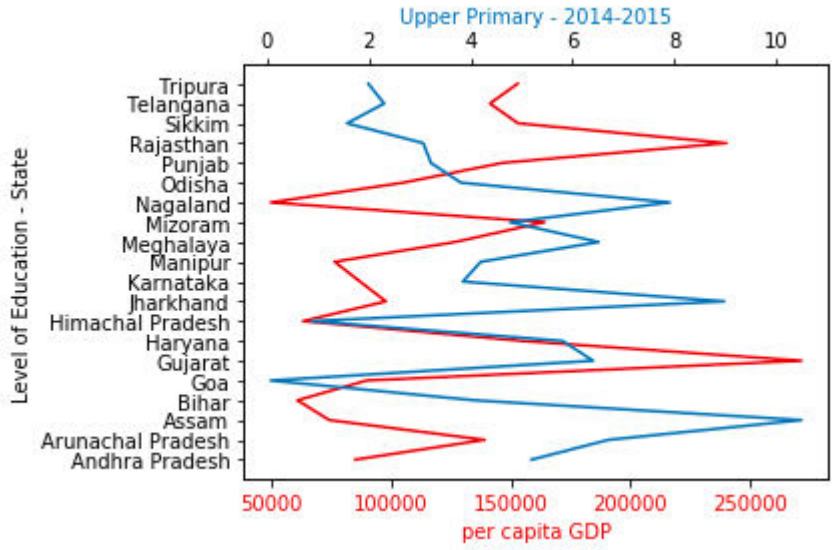
 # Primary - 2014-2015

```
plt.figure(num=None,figsize=(50,10),dpi=60,facecolor='w',edgecolor='k')
fig, ax1 = plt.subplots()
color = 'tab:red'
ax1.set_xlabel('per capita GDP',color=color)
ax1.set_ylabel('Level of Education - State')
ax1.plot(drop_out_rate_df['per capita GDP'],drop_out_rate_df.index, color=color)
ax1.tick_params(axis='x', labelcolor=color)
ax2 = ax1.twiny()  # instantiate a second axes that shares the same x-axis
 color = 'tab:blue'
ax2.set_xlabel('Primary - 2014-2015', color=color)  # we already handled the x-label with ax1
ax2.plot(drop_out_rate_df['Primary - 2014-2015'],drop_out_rate_df.index, color=color)
ax2.tick_params(axis='y', labelcolor=color)
 fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()
```



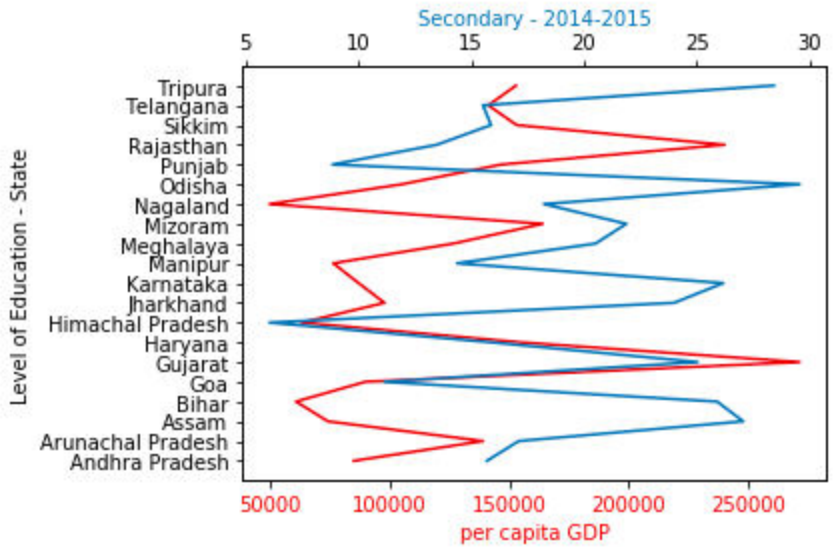
 # Upper Primary - 2014-2015

```
plt.figure(num=None,figsize=(50,10),dpi=60,facecolor='w',edgecolor='k')
fig, ax1 = plt.subplots()
color = 'tab:red'
ax1.set_xlabel('per capita GDP',color=color)
ax1.set_ylabel('Level of Education - State')
ax1.plot(drop_out_rate_df['per capita GDP'],drop_out_rate_df.index, color=color)
ax1.tick_params(axis='x', labelcolor=color)
ax2 = ax1.twiny()  # instantiate a second axes that shares the same x-axis
color = 'tab:blue'
ax2.set_xlabel('Upper Primary - 2014-2015', color=color)  # we already handled the x-label with ax1
ax2.plot(drop_out_rate_df['Upper Primary - 2014-2015'],drop_out_rate_df.index, color=color)
ax2.tick_params(axis='y', labelcolor=color)
fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()
```



 # Secondary - 2014-2015

```
plt.figure(num=None,figsize=(50,10),dpi=60,facecolor='w',edgecolor='k')
fig, ax1 = plt.subplots()
color = 'tab:red'
ax1.set_xlabel('per capita GDP',color=color)
ax1.set_ylabel('Level of Education - State')
ax1.plot(drop_out_rate_df['per capita GDP'],drop_out_rate_df.index, color=color)
ax1.tick_params(axis='x', labelcolor=color)
 ax2 = ax1.twiny()  # instantiate a second axes that shares the same x-axis
color = 'tab:blue'
ax2.set_xlabel('Secondary - 2014-2015', color=color)  # we already handled the x-label with ax1
ax2.plot(drop_out_rate_df['Secondary - 2014-2015'],drop_out_rate_df.index, color=color)
ax2.tick_params(axis='y', labelcolor=color)
 fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()
```

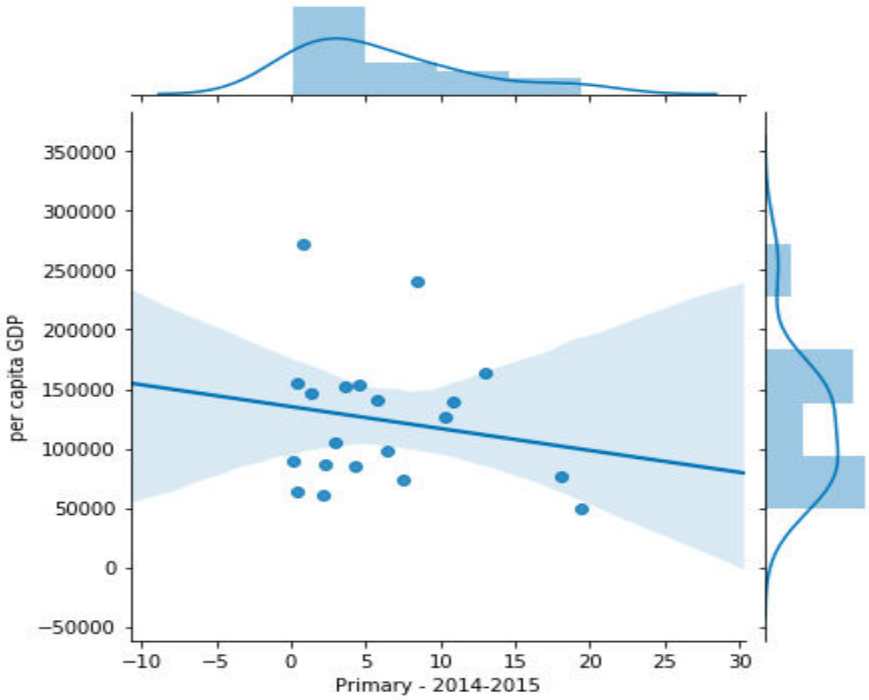


 # Other way to plot for more visualization

<u>1- Primary - 2014-2015</u>

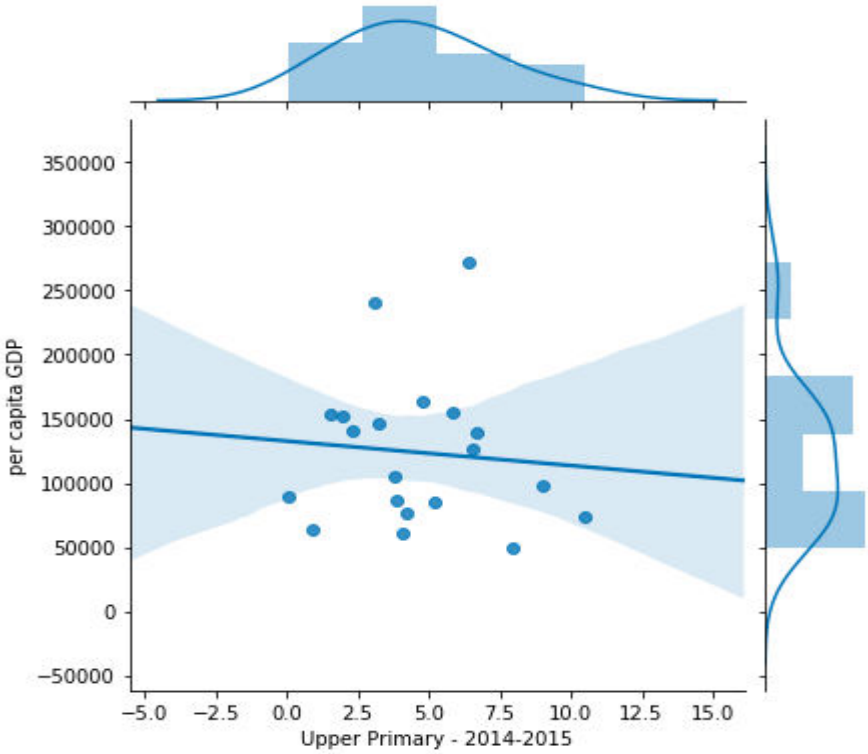
sns.jointplot(x="Primary - 2014-2015", y="per capita GDP", data=drop_out_rate_df, kind='reg');

plt.show()



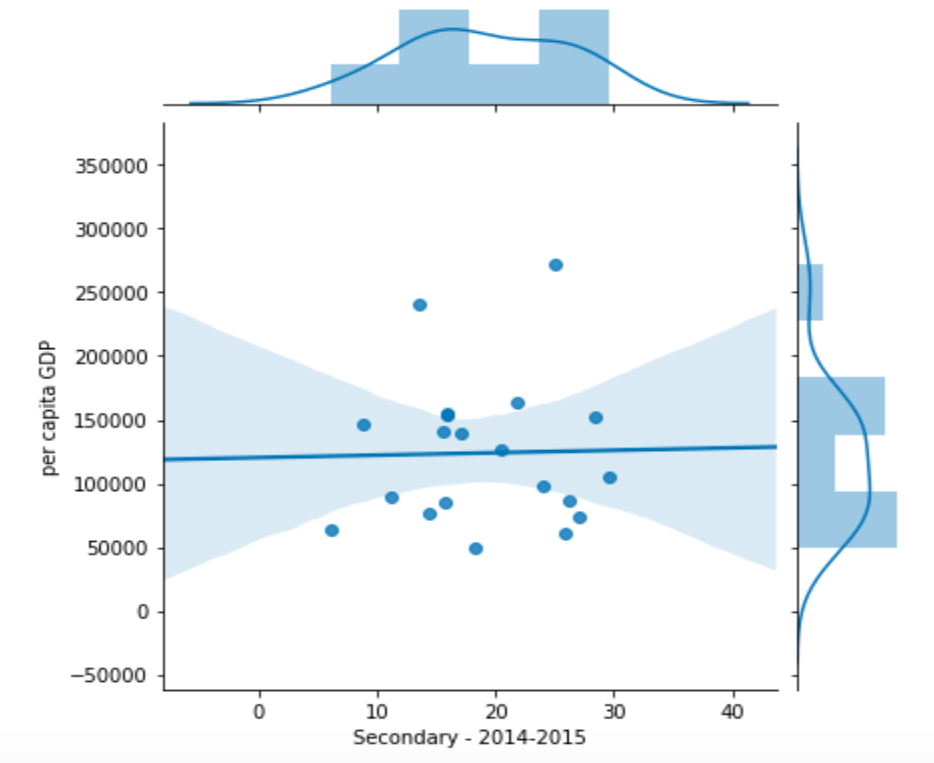
<u>2- Upper Primary - 2014-2015</u>

```
sns.jointplot(x="Upper Primary - 2014-2015", y="per capita GDP", data=drop_out_rate_df,
kind='reg');

plt.show()
```



3- Secondary - 2014-2015

```
sns.jointplot(x="Secondary - 2014-2015", y="per capita GDP", data=drop_out_rate_df,
kind='reg');

plt.show()
```



Question: Form at least one reasonable hypothesis for the observations from the data?

Answer: In Primary - 2014-2015 "Level of Education " state wise is very up and down and also not good as compare to "Upper Primary - 2014-2015" and "Secondary - 2014-2015", but in level of education in "Upper Primary - 2014-2015" growing batter than "Primary - 2014-2015" but not as much "Secondary - 2014-2015".

In Secondary - 2014-2015 Level of Education is good in almost all states and also "Secondary - 2014-2015" is the best among all three.

GDP for each section is same in all states but level of education is differ in each section and also we can see in some states if the GDP growing fast comparatively education level growing not much or just opposite.

Education level of Punjab, Himanchal Pradesh, Haryana, Gujarat, Goa, in 'Primary' section, Himanchal Pradesh, Goa in 'Upper Primary' section and Himanchal Pradesh, Punjab in 'Secondary' section is very low.

We need to focus more on education level in each section of Himanchal Pradesh.

In jointPlot we can see that there is positive or negative co-relation between per capital GDP and sub sectors . In Primary - 2014-2015 and Upper Primary - 2014-2015 points are more closer to the line it means highly corelated, but in Secondary - 2014-2015 points are not more closer to the line its shows per capital GDP and Secondary - 2014-2015 sub sector not much corelated.