Part-I: GDP Analysis of Indian States

Part I-A:

In [468]:

```
import numpy as np
import pandas as pd
import os
```

Importing Data and looking at Head

In [15]:

```
df= pd.read_csv(os.getcwd()+'\\data.csv',index_col= "Items  Description")
df.head()
```

Out[15]:

	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	Himachal Pradesh	...	Te
Items Description												
GSDP - CURRENT PRICES (` in Crore)	2011-12	379402.0	11063.0	143175.0	247144.0	158074.0	42367.0	615606.0	297539.0	72720.0	...	35
GSDP - CURRENT PRICES (` in Crore)	2012-13	411404.0	12547.0	156864.0	282368.0	177511.0	38120.0	724495.0	347032.0	82820.0	...	40
GSDP - CURRENT PRICES (` in Crore)	2013-14	464272.0	14602.0	177745.0	317101.0	206690.0	35921.0	807623.0	400662.0	94764.0	...	45
GSDP - CURRENT PRICES (` in Crore)	2014-15	526468.0	16761.0	198098.0	373920.0	234982.0	40633.0	895027.0	437462.0	104369.0	...	51
GSDP - CURRENT PRICES (` in Crore)	2015-16	609934.0	18784.0	224234.0	413503.0	260776.0	45002.0	994316.0	485184.0	NaN	...	57

5 rows × 35 columns

Removing the following Rows

'(% Growth over the previous year)' and 'GSDP - CURRENT PRICES (` in Crore)' for the year 2016-17

In [35]:

```
df1 = df[ df['Duration'] != '2016-17' ]
df1
```

Out[35]:

	Duration	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	Himacha Pradesh
Items Description										
GSDP - CURRENT PRICES (` in Crore)	2011-12	379402.00	11063.00	143175.00	247144.00	158074.00	42367.00	615606.00	297539.00	72720.00
GSDP - CURRENT PRICES (` in Crore)	2012-13	411404.00	12547.00	156864.00	282368.00	177511.00	38120.00	724495.00	347032.00	82820.00
GSDP - CURRENT PRICES (` in Crore)	2013-14	464272.00	14602.00	177745.00	317101.00	206690.00	35921.00	807623.00	400662.00	94764.00
GSDP - CURRENT PRICES (` in Crore)	2014-15	526468.00	16761.00	198098.00	373920.00	234982.00	40633.00	895027.00	437462.00	104369.0(
GSDP - CURRENT PRICES (` in Crore)	2015-16	609934.00	18784.00	224234.00	413503.00	260776.00	45002.00	994316.00	485184.00	NaN
(% Growth over previous year)	2012-13	8.43	13.41	9.56	14.25	12.30	-10.02	17.69	16.63	13.89
(% Growth over previous year)	2013-14	12.85	16.38	13.31	12.30	16.44	-5.77	11.47	15.45	14.42
(% Growth over previous year)	2014-15	13.40	14.79	11.45	17.92	13.69	13.12	10.82	9.18	10.14
(% Growth over previous year)	2015-16	15.85	12.07	13.19	10.59	10.98	10.75	11.09	10.91	NaN

9 rows × 35 columns

In [41]:

```
yaxis1 = df1.loc["(% Growth over previous year)",:].mean()
```

In [43]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

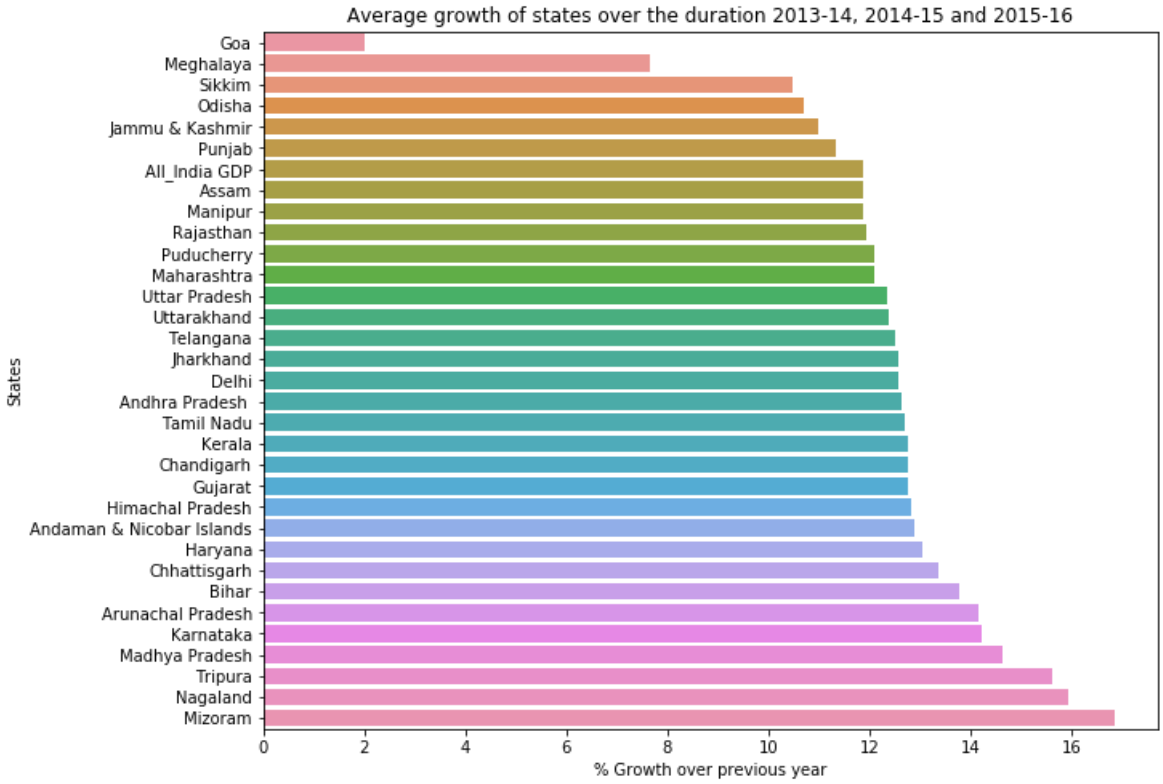
In [96]:

```
dfx=pd.DataFrame(yaxis1)
dfx.sort_values(by=[0],inplace=True)
dfx.columns = ["% Growth over previous year"]
dfx.index.name = 'States'
dfx.dropna(inplace=True)
```

Plotting a Graph below to identify Average growth of states over the duration 2013-14, 2014-15 and 2015-16

In [97]:

```
plt.figure(figsize=(10, 8))
sns.barplot(x=dfx["% Growth over previous year"], y=dfx.index)
plt.title('Average growth of states over the duration 2013-14, 2014-15 and 2015-16')
plt.show()
```



The above Graph states that
Mizoram is the state which is growing consistantly fast and Goa is struggling
National Average is 11.8675 and UP(State where i live in) is performing better at 12.3675

In [119]:

```
dfydata = df1.iloc[4,1:]
```

In [120]:

```
dfy=pd.DataFrame(dfydata)
```

In [121]:

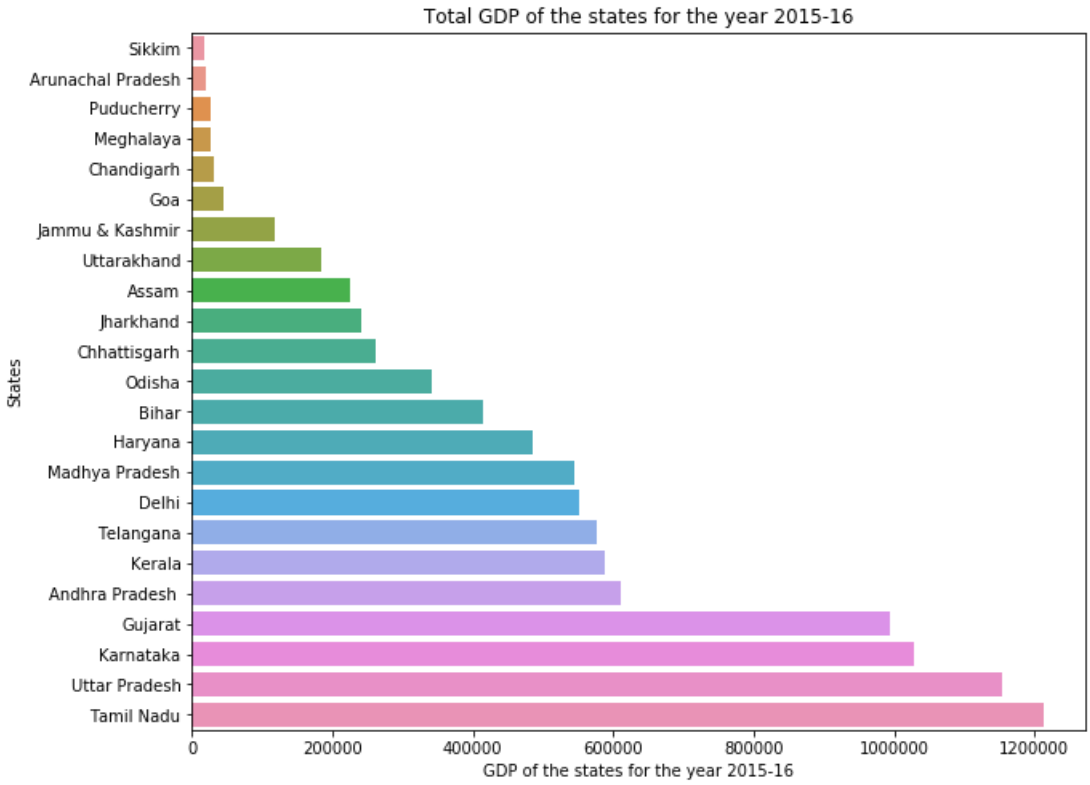
```
dfy.columns = ["GDP of the states for the year 2015-16"]
dfy.index.name = 'States'
```

In [122]:

```
dfy.sort_values(by=["GDP of the states for the year 2015-16"],inplace=True)
dfy.dropna(inplace=True)
dfy.drop(['All_India GDP'],axis=0,inplace=True)
```

In [123]:

```
plt.figure(figsize=(10, 8))
sns.barplot(x=dfy["GDP of the states for the year 2015-16"], y=dfy.index)
plt.title('Total GDP of the states for the year 2015-16')
plt.show()
```



Top 5 states are Tamil Nadu, Uttar Pradesh, Karnataka, Gujarat, Andhra Pradesh

Bottom 5 states are Sikkim,Arunuchal Pradesh, Pudducherry, Meghalaya, Chandigarh

Part I-B:

In [236]:

```
filelist = os.listdir(os.getcwd()+'\\GSVA')
```

In [237]:

```
dfb = pd.read_csv(os.getcwd()+'\\GSVA\\'+filelist[0],encoding = "ISO-8859-1")
m = dfb.iloc[:,1].values
```

In [238]:

```
df = pd.DataFrame(index=m)
```

In [239]:

```
for x in filelist:
    dfb = pd.read_csv(os.getcwd()+'\\GSVA\\'+x,encoding = "ISO-8859-1")
    m = dfb.iloc[:,5].values
    df[x[4:-21]] = m
```

In [466]:

```
df.tail()
```

Out[466]:

	Bihar	Uttar_Pradesh	Manipur	Assam	Jharkhand	Madhya_Pradesh	Odisha	Meghalaya	Tripura
Taxes on Products	3213546	10107396	92766.0	1725309	2499171	4284700	3151184	213272.0	149345.0
Subsidies on products	2006421	3287219	69137.0	582406	411619	2102800	1209349	62112.0	94002.0
Gross State Domestic Product	37391988	104337115	1804276.0	19809800	21710718	48198169	32197092	2440807.0	2966662.0
Population ('00)	1101240	2109940	30873.0	326780	349660	765180	435220	32020.0	38350.0
Per Capita GSDP (Rs.)	33954	49450	58442.0	60621	62091	62989	73979	76228.0	77358.0

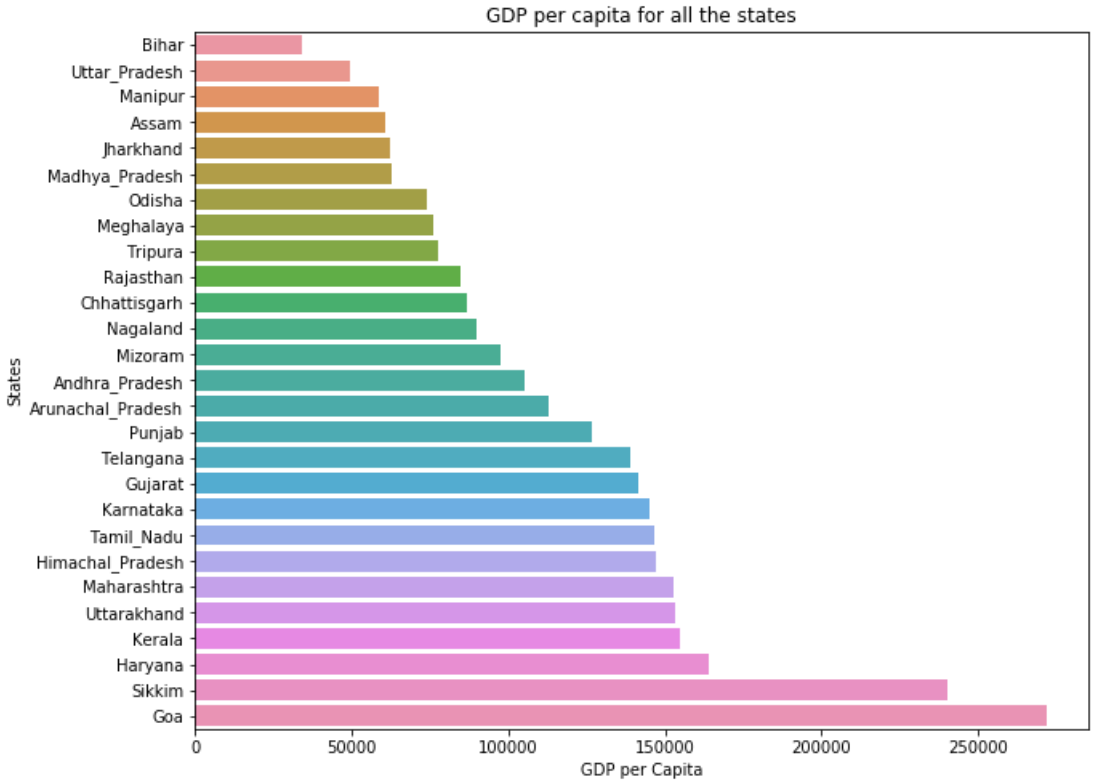
5 rows × 27 columns

In [248]:

```
df.sort_values(by=['Per Capita GSDP (Rs.)'],axis=1,inplace=True)
```

In [261]:

```
plt.figure(figsize=(10, 8))
sns.barplot(x=df.iloc[32,:].values, y=df.columns.values)
plt.xlabel('GDP per Capita')
plt.ylabel('States')
plt.title('GDP per capita for all the states')
plt.show()
```



Based on GDP per Capita

Top 5 States Are Goa, Sikkim, Haryana, Kerela, Uttarakhand
Bottom 5 States are Bihar, Uttar Pradesh, Manipur, Assam , Jharkhand

Ratio of Higest vs Lowest GDP is approximately 8:1

Highest belong to Goa that is 271793 and Lowest belong to Bihar 33954

In [288]:

```
df2 = pd.DataFrame(index=df.columns.values)
df2['Primary'] = df.loc['Primary'].values
df2['Secondary'] = df.loc['Secondary'].values
df2['Tertiary'] = df.loc['Tertiary'].values
df2['TOTAL GSVA at basic prices'] = df.loc['TOTAL GSVA at basic prices'].values
```

In [289]:

```
df2.index.name = 'States'
```

In [290]:

```
df2.sort_values(by=['TOTAL GSVA at basic prices'],axis=0,inplace=True)
```

In [302]:

```
df3= df2.iloc[:,0:3]
```

Creating a new dataframe to calculate percentage of each category

In [533]:

```
primarypercent = [(x/(x+y+z))*100 for x,y,z in zip(df3['Primary'].values,df3['Secondary'].values,df3
['Tertiary'].values) ]
Secondarypercent = [(y/(x+y+z))*100 for x,y,z in zip(df3['Primary'].values,df3['Secondary'].values,d
f3['Tertiary'].values) ]
Tertiarypercent = [(z/(x+y+z))*100 for x,y,z in
zip(df3['Primary'].values,df3['Secondary'].values,df3['Tertiary'].values) ]
```

In [539]:

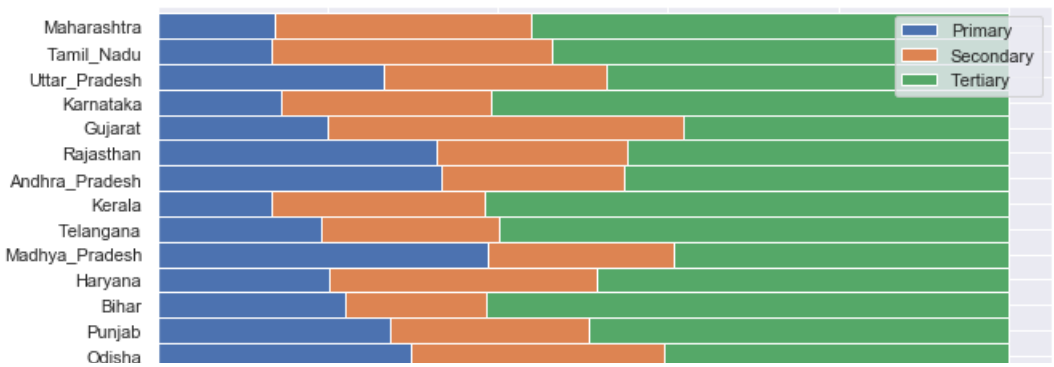
```
df8 = pd.DataFrame(index=df3.index.values)
df8['Primary'] = primarypercent
df8['Secondary'] = Secondarypercent
df8['Tertiary'] = Tertiarypercent
```

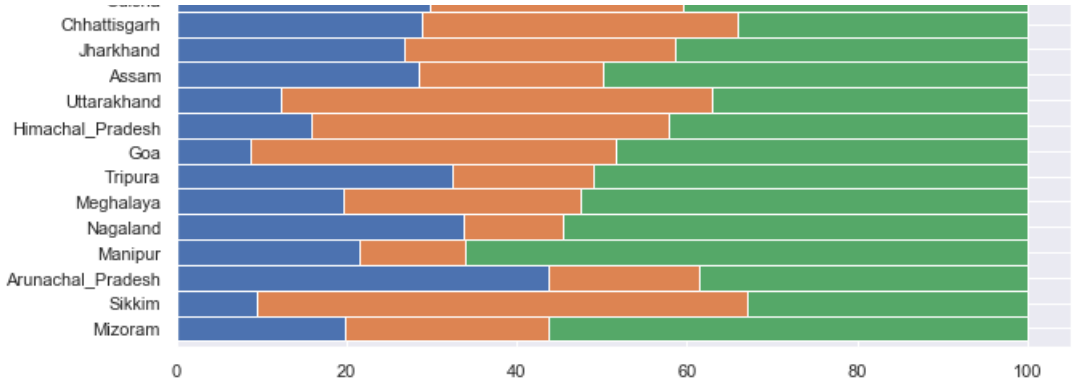
In [541]:

```
df8.plot(stacked=True, width=1, kind="barh", lw=1, figsize=(10, 8))
```

Out[541]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xb3b3d70>
```





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

In [465]:

```
df.head()
```

Out[465]:

	Bihar	Uttar_Pradesh	Manipur	Assam	Jharkhand	Madhya_Pradesh	Odisha	Meghalaya	Tripura	Raja
Agriculture, forestry and fishing	7951890	25097754	383140.0	3855548	3211065	16235647	6422978	366521.0	799825.0	1504
Crops	4688237	16215212	226583.0	2890544	2094466	13216013	4382636	223330.0	397591.0	7673
Livestock	2060296	7096876	75483.0	173478	562503	1932769	788243	72442.0	88176.0	5356
Forestry and logging	550132	1404936	45831.0	261987	445149	965588	791463	61166.0	145096.0	1956
Fishing and aquaculture	653224	380730	35243.0	529539	108947	121277	460636	9583.0	168961.0	5803

5 rows × 27 columns

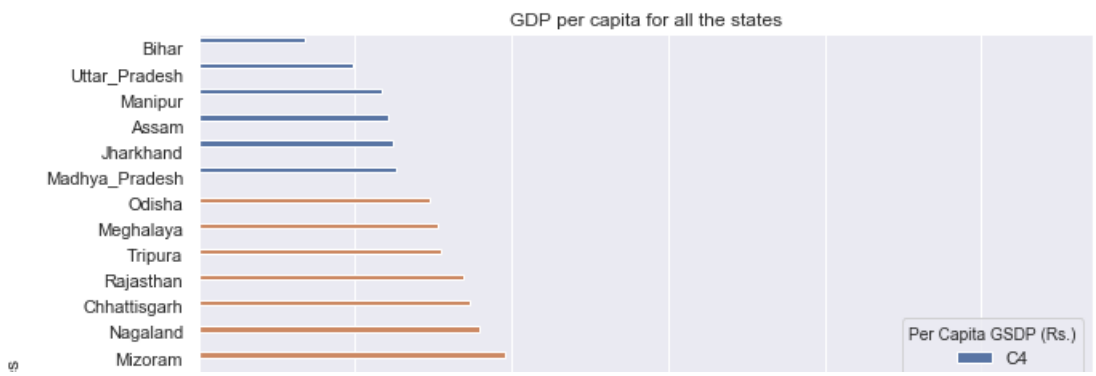
In [373]:

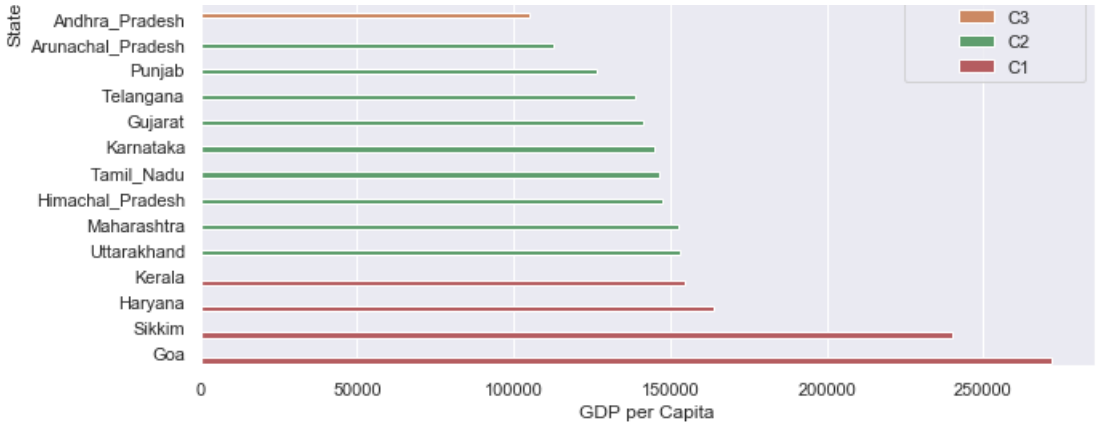
```
quantiles = pd.qcut(df.iloc[32,:], q= [0,0.20,0.5, 0.85,1],labels=['C4','C3','C2','C1'])
```

Below we are categorizing states based on Quantiles

In [345]:

```
plt.figure(figsize=(10, 8))
sns.barplot(x=df.iloc[32,:].values, y=df.columns.values,hue=quantiles)
plt.xlabel('GDP per Capita')
plt.ylabel('States')
plt.title('GDP per capita for all the states')
plt.show()
```





The above graph shows quantiles containing C1,C2,C3,C4 categories based on (0.20,0.5, 0.85, 1) quantiles where 85th and the 100th percentile are in C1, those between 50th and 85th percentile are in C2 and so on

In [350]:

```
quantiles = pd.DataFrame(quantiles)
```

In [391]:

```
dftranspose = df.transpose()
```

In [409]:

```
df5 = dftranspose.groupby(quantiles.values,axis=0)[dftranspose.columns.values].sum()
```

In [410]:

```
df5
```

Out[410]:

	Agriculture, forestry and fishing	Crops	Livestock	Forestry and logging	Fishing and aquaculture	Mining and quarrying	Primary	Manufacturing	Electricity, gas, water supply & other utility services	Cons
C4	56735044.0	39331055.0	11901405.0	3673623.0	1828960.0	6096419.0	62831463.0	24987032.0	4310264.0	2277
C3	42226888.0	23676317.0	11135794.0	4045895.0	3368881.0	10835771.0	53062661.0	22038331.0	4604774.0	1570
C2	75209022.0	47550095.0	19363891.0	5603213.0	2691822.0	10373346.0	85582369.0	109196530.0	13823378.0	4065
C1	14391809.0	7962514.0	4619756.0	872335.0	937204.0	588961.0	14980771.0	13758793.0	2000998.0	1126

4 rows × 33 columns

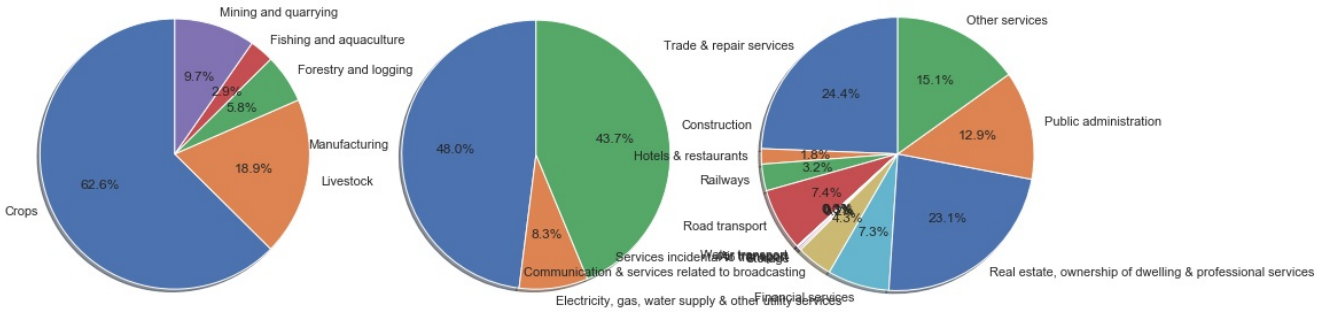
The above Dataframe is a grouped into Four Categories namely C1, C2, C3 and C4

For each category C1, C2, C3, C4 we are creating Pie Charts below

In [461]:

```
labels1 = 'Crops', 'Livestock','Forestry and logging', 'Fishing and aquaculture','Mining and quarr
ying'
sizes1 = df5.iloc[0,1:6].values
labels2 = 'Manufacturing','Electricity, gas, water supply & other utility services','Construction'
sizes2 = df5.iloc[0,7:10].values
labels3 =  'Trade & repair services','Hotels & restaurants','Railways', 'Road transport', 'Water
```

```
transport', 'Air transport','Services incidental to transport', 'Storage','Communication &
services related to broadcasting','Financial services','Real estate, ownership of dwelling & profe
ssional services','Public administration', 'Other services'
sizes3 = df5.iloc[0,np.r_[12, 13, 15:26]]

fig1, (ax1,ax2,ax3) = plt.subplots(1,3,figsize=(16, 8))

ax1.pie(sizes1,  labels=labels1, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax2.pie(sizes2,  labels=labels2, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax2.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax3.pie(sizes3,  labels=labels3, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax3.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.


plt.show()
```



The above Pie chart shows percentage composition of states belonging to C4 category

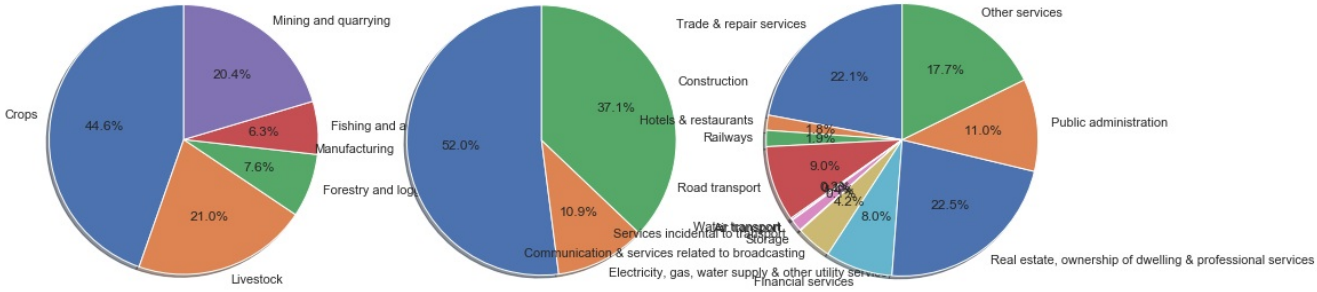
> With First represents Primary, Second Represents Secondary And Third Represent Tertiary However an assumption is taken here
>
> > Primary -- Since "Agriculture, forestry and fishing" includes "Crops","Livestock","Forestry and logging" and "Fishing and aquaculture". Therefore only Sub- Categories of "Agriculture, forestry and fishing" are taken into consideration
> > Tertiary -- Since "Trade, repair, hotels and restaurants" and "Transport, storage, communication & services related to broadcasting" also contains sub categories therefore those subcategories are taken into consideration

> C4 category
>
> > Primary -- Crops and Livstock covers approximately 80%
> > Secondary-- Manufacturing and Construction covers approximately 80%
> > Tertiary -- 'Real estate, ownership of dwelling & professional services', 'Trade and Repair service', 'Public Administration' and 'Other Services' covers approximately 80%

In [462]:

```
labels1 = 'Crops', 'Livestock','Forestry and logging', 'Fishing and aquaculture','Mining and quarr
```

```
ying'
sizes1 = df5.iloc[1,1:6].values
labels2 = 'Manufacturing','Electricity, gas, water supply & other utility services','Construction'
sizes2 = df5.iloc[1,7:10].values
labels3 =  'Trade & repair services','Hotels & restaurants','Railways', 'Road transport', 'Water
transport', 'Air transport','Services incidental to transport', 'Storage','Communication &
services related to broadcasting','Financial services','Real estate, ownership of dwelling & profe
ssional services','Public administration', 'Other services'
sizes3 = df5.iloc[1,np.r_[12, 13, 15:26]]

fig1, (ax1,ax2,ax3) = plt.subplots(1,3,figsize=(16, 8))

ax1.pie(sizes1,  labels=labels1, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax2.pie(sizes2,  labels=labels2, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax2.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax3.pie(sizes3,  labels=labels3, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax3.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.


plt.show()
```



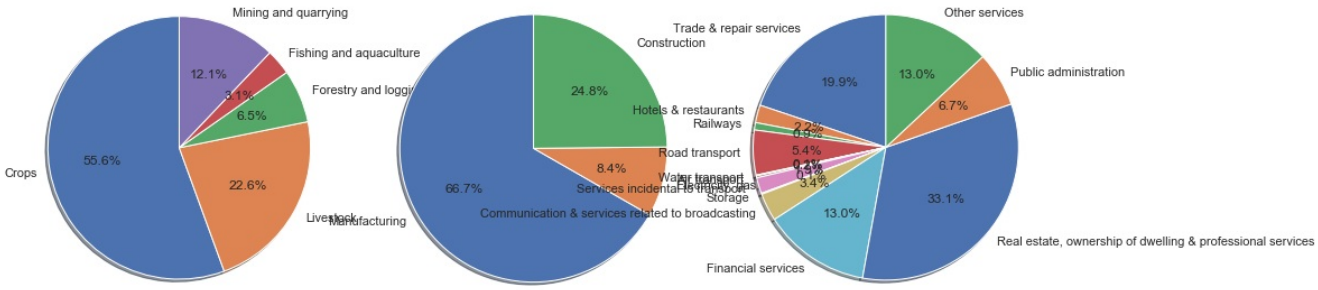
The above Pie chart shows percentage composition of states belonging to C3 category

> With First represents Primary, Second Represents Secondary And Third Represent Tertiary However an assumption is taken here
>
> > Primary -- Since "Agriculture, forestry and fishing" includes "Crops","Livestock","Forestry and logging" and "Fishing and aquaculture". Therefore only Sub- Categories of "Agriculture, forestry and fishing" are taken into consideration
> > Tertiary -- Since "Trade, repair, hotels and restaurants" and "Transport, storage, communication & services related to broadcasting" also contains sub categories therefore those subcategories are taken into consideration

> C3 category
>
> > Primary -- Crops,Livstock, Mining and Quarrying covers approximately 80%
> > Secondary-- Manufacturing and Construction covers approximately 80%
> > Tertiary -- 'Real estate, ownership of dwelling & professional services', 'Trade and Repair service', 'Public Administration' 'Other Services' and 'Road Transport' covers approximately 80%

In [463]:

```
labels1 = 'Crops', 'Livestock','Forestry and logging', 'Fishing and aquaculture','Mining and quarr
ying'
sizes1 = df5.iloc[2,1:6].values
labels2 = 'Manufacturing','Electricity, gas, water supply & other utility services','Construction'
sizes2 = df5.iloc[2,7:10].values
labels3 =  'Trade & repair services','Hotels & restaurants','Railways', 'Road transport', 'Water
transport', 'Air transport','Services incidental to transport', 'Storage','Communication &
services related to broadcasting','Financial services','Real estate, ownership of dwelling & profe
ssional services','Public administration', 'Other services'
sizes3 = df5.iloc[2,np.r_[12, 13, 15:26]]

fig1, (ax1,ax2,ax3) = plt.subplots(1,3,figsize=(16, 8))

ax1.pie(sizes1,  labels=labels1, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax2.pie(sizes2,  labels=labels2, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax2.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax3.pie(sizes3,  labels=labels3, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax3.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.


plt.show()
```



The above Pie chart shows percentage composition of states belonging to C2 category

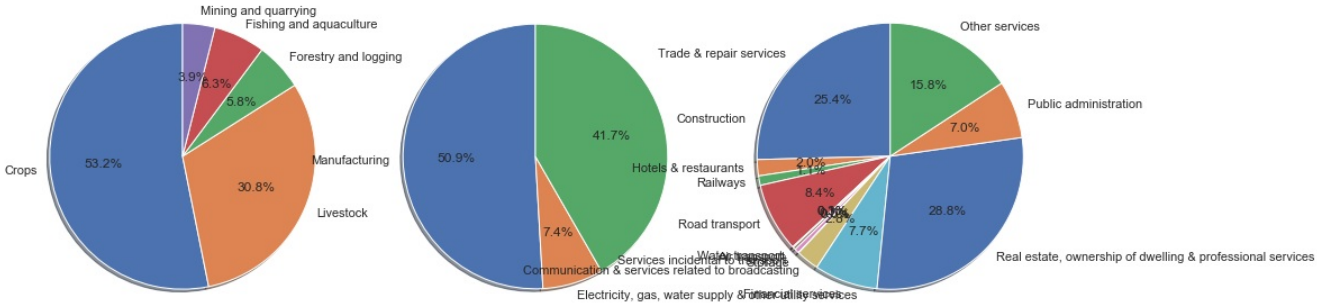
> With First represents Primary, Second Represents Secondary And Third Represent Tertiary However an assumption is taken here
>
> > Primary -- Since "Agriculture, forestry and fishing" includes "Crops","Livestock","Forestry and logging" and "Fishing and aquaculture". Therefore only Sub- Categories of "Agriculture, forestry and fishing" are taken into consideration
> > Tertiary -- Since "Trade, repair, hotels and restaurants" and "Transport, storage, communication & services related to broadcasting" also contains sub categories therefore those subcategories are taken into consideration

> C2 category
>
> > Primary -- Crops,Livstock covers approximately 80%

Secondary-- Manufacturing and Construction covers approximately 80%
Tertiary -- 'Real estate, ownership of dwelling & professional services', 'Trade and Repair service', 'Public Administration' 'Other Services' and 'Finincial Services' covers approximately 80%

In [464]:

```
labels1 = 'Crops', 'Livestock','Forestry and logging', 'Fishing and aquaculture','Mining and quarr
ying'
sizes1 = df5.iloc[3,1:6].values
labels2 = 'Manufacturing','Electricity, gas, water supply & other utility services','Construction'
sizes2 = df5.iloc[3,7:10].values
labels3 =  'Trade & repair services','Hotels & restaurants','Railways', 'Road transport', 'Water
transport', 'Air transport','Services incidental to transport', 'Storage','Communication &
services related to broadcasting','Financial services','Real estate, ownership of dwelling & profe
ssional services','Public administration', 'Other services'
sizes3 = df5.iloc[3,np.r_[12, 13, 15:26]]

fig1, (ax1,ax2,ax3) = plt.subplots(1,3,figsize=(16, 8))

ax1.pie(sizes1,  labels=labels1, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax2.pie(sizes2,  labels=labels2, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax2.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

ax3.pie(sizes3,  labels=labels3, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax3.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.


plt.show()
```



The above Pie chart shows percentage composition of states belonging to C1 category

With First represents Primary, Second Represents Secondary And Third Represent Tertiary However an assumption is taken here

Primary -- Since "Agriculture, forestry and fishing" includes "Crops","Livestock","Forestry and logging" and "Fishing and aquaculture". Therefore only Sub- Categories of "Agriculture, forestry and fishing" are taken into consideration
Tertiary -- Since "Trade, repair, hotels and restaurants" and "Transport, storage, communication & services related to broadcasting" also contains sub categories therefore those subcategories are taken into consideration

C1 category

> Primary -- Crops,Livstock covers approximately 80%
> Secondary-- Manufacturing and Construction covers approximately 80%
> Tertiary -- 'Real estate, ownership of dwelling & professional services', 'Trade and Repair service', 'Public Administration' 'Other Services' and 'Road Transport' covers approximately 80%

Improvements can be as follows

> Primary sector of all the Categories is mostly thrived with Crops and Livstock. Even a small improvement in the internal process can improve the GDP by a large amount
> Manufacturing sector is another sector which provides a lot of Jobs and also contribute Largely to GDP in all the categories. Even a small improvement in the internal process can improve the GDP by a large amount

Part-II: GDP and Education Drop-out Rates

Part-II: GDP and Education

In [471]:

```
df.tail()
```

Out[471]:

	Bihar	Uttar_Pradesh	Manipur	Assam	Jharkhand	Madhya_Pradesh	Odisha	Meghalaya	Tripura
Taxes on Products	3213546	10107396	92766.0	1725309	2499171	4284700	3151184	213272.0	149345.0
Subsidies on products	2006421	3287219	69137.0	582406	411619	2102800	1209349	62112.0	94002.0
Gross State Domestic Product	37391988	104337115	1804276.0	19809800	21710718	48198169	32197092	2440807.0	2966662.0
Population ('00)	1101240	2109940	30873.0	326780	349660	765180	435220	32020.0	38350.0
Per Capita GSDP (Rs.)	33954	49450	58442.0	60621	62091	62989	73979	76228.0	77358.0

5 rows × 27 columns

In [474]:

```
dfread = pd.read_csv(os.getcwd() +'\\Dropout rates\\data.csv')
dfread.head()
```

Out[474]:

	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	Se
0	1	A & N	0.68	1.21	0.51	1.23	0.51	1.69	5.56	7.20	9.87	14.14	15

		Islands											
1	2	Andhra Pradesh	3.18	4.35	6.79	3.36	3.78	5.20	12.72	12.65	15.71	0.35	1
2	3	Arunachal Pradesh	15.16	10.89	10.82	7.47	5.59	6.71	12.93	14.49	17.11	5.11	17
3	4	Assam	6.24	7.44	15.36	7.20	7.05	10.51	26.77	30.43	27.06	4.69	7.
4	5	Bihar	NaN	2.09	NaN	NaN	2.98	4.08	30.14	25.33	25.90	NaN	Na

Clening and taking out the required data

In [488]:

```
df2 = dfread.iloc[np.r_[1:5,6,10:14,15:18,19:26,27:35],np.r_[1, 3,7,10,13]]
```

In [499]:

```
states = [word.replace(' ','_') for word in df2['Level of Education - State']]
```

In [501]:

```
df2['Level of Education - State'] = states
```

In [515]:

```
df6 = df.transpose()
df6 = df6.iloc[:,[32]]
```

In [517]:

```
df2
```

Out[517]:

	Level of Education - State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Senior Secondary - 2014-2015
1	Andhra_Pradesh	4.35	5.20	15.71	NaN
2	Arunachal_Pradesh	10.89	6.71	17.11	18.42
3	Assam	7.44	10.51	27.06	NaN
4	Bihar	2.09	4.08	25.90	NaN
6	Chhatisgarh	1.42	5.85	21.26	2.76
10	Goa	0.20	0.07	11.15	13.91
11	Gujarat	0.76	6.41	25.04	7.04
12	Haryana	0.41	5.81	15.89	5.75
13	Himachal_Pradesh	0.46	0.87	6.07	7.41
15	Jharkhand	6.41	8.99	24.00	3.41
16	Karnataka	2.32	3.85	26.18	1.96
17	Kerala	NaN	NaN	12.32	0.47
19	Madhya_Pradesh	10.14	9.20	24.77	NaN
20	Maharashtra	0.55	1.79	12.87	1.83
21	Manipur	18.00	4.20	14.38	NaN
22	Meghalaya	10.34	6.52	20.52	NaN
23	Mizoram	12.96	4.78	21.88	6.91
24	Nagaland	19.41	7.92	18.23	6.97

	Level of Education - State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Senior Secondary - 2014-2015
25	Odisha	2.94	3.81	29.96	NaN
27	Punjab	1.29	3.22	8.86	5.83
28	Rajasthan	8.39	3.07	13.48	NaN
29	Sikkim	4.57	1.57	15.89	11.76
30	Tamil_Nadu	0.46	NaN	8.10	3.41
31	Telangana	5.81	2.30	15.53	0.77
32	Tripura	3.58	1.99	28.42	8.93
33	Uttar_Pradesh	7.08	2.70	10.22	2.10
34	Uttrakhand	3.07	1.19	10.40	3.01

We will be working with df2 now

In [520]:

```
corr = pd.merge(df2, df6, how="inner", left_on="Level of Education - State", right_on=df6.index)

corr.head()
```

Out[520]:

	Level of Education - State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Senior Secondary - 2014-2015	Per Capita GSDP (Rs.)
0	Andhra_Pradesh	4.35	5.20	15.71	NaN	104977.0
1	Arunachal_Pradesh	10.89	6.71	17.11	18.42	112718.0
2	Assam	7.44	10.51	27.06	NaN	60621.0
3	Bihar	2.09	4.08	25.90	NaN	33954.0
4	Goa	0.20	0.07	11.15	13.91	271793.0

The above dataset will be used to find the correlaation between dropouts and GDP per capita

In [523]:

```
corr2= corr.iloc[:,1:]
corr2.head()
```

Out[523]:

	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Senior Secondary - 2014-2015	Per Capita GSDP (Rs.)
0	4.35	5.20	15.71	NaN	104977.0
1	10.89	6.71	17.11	18.42	112718.0
2	7.44	10.51	27.06	NaN	60621.0
3	2.09	4.08	25.90	NaN	33954.0
4	0.20	0.07	11.15	13.91	271793.0

In [525]:

```
cor = corr2.corr()
round(cor, 3)
```

Out[525]:

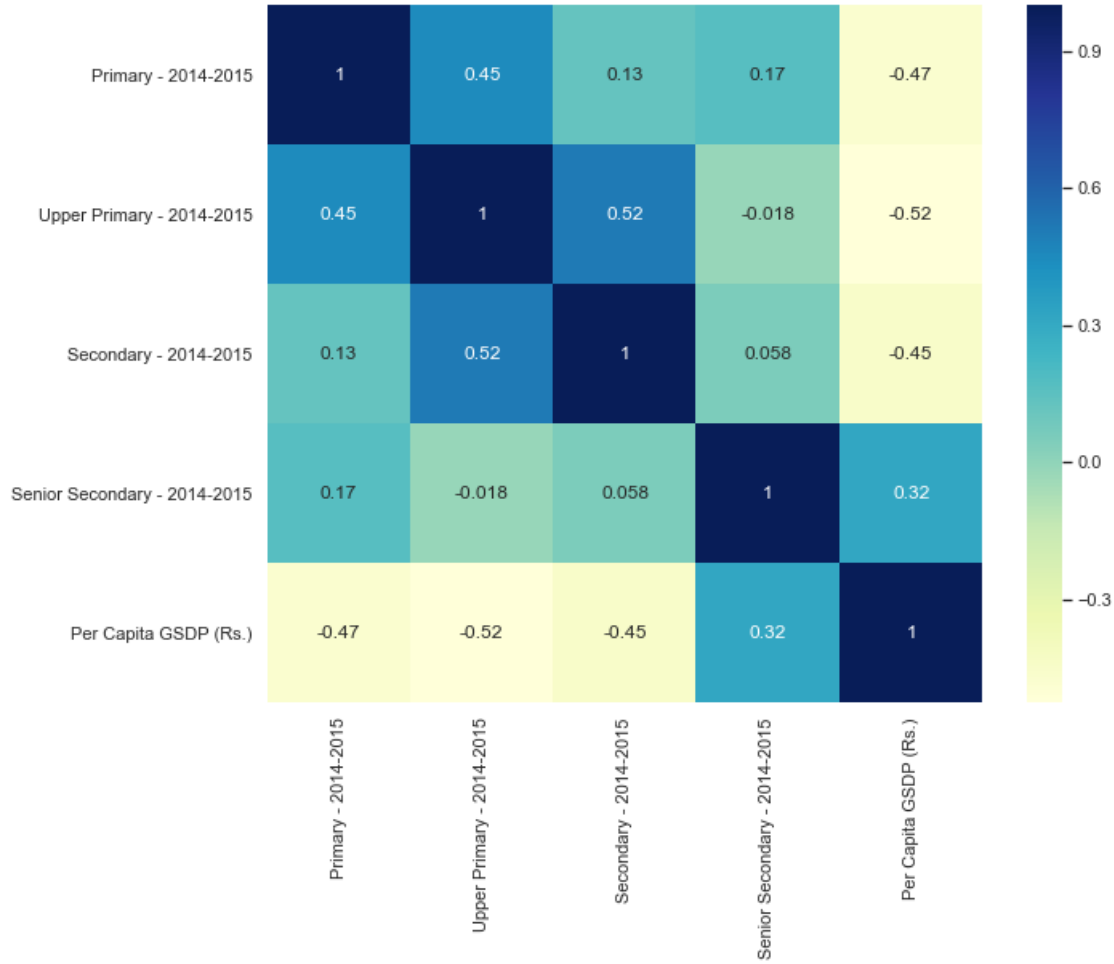
	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Senior Secondary - 2014-2015	Per Capita GSDP (Rs.)
Primary - 2014-2015	1.000	0.452	0.125	0.170	-0.472
Upper Primary - 2014-2015	0.452	1.000	0.516	-0.018	-0.524
Secondary - 2014-2015	0.125	0.516	1.000	0.058	-0.446
Senior Secondary - 2014-2015	0.170	-0.018	0.058	1.000	0.320
Per Capita GSDP (Rs.)	-0.472	-0.524	-0.446	0.320	1.000

The above table represents the Correletion Table between Dropouts and GDP per capita

Representing the above table in the form of Heatmap below

In [526]:

```
plt.figure(figsize=(10,8))
# heatmap
sns.heatmap(cor, cmap="YlGnBu", annot=True)
plt.show()
```



This Heatmap represents the Correlation between Dropout rates compared to GDP per capita

Negative relation between Primary, Upper Primary and Secondry shows that GDP is inversely propotional to the dropout rates in the same