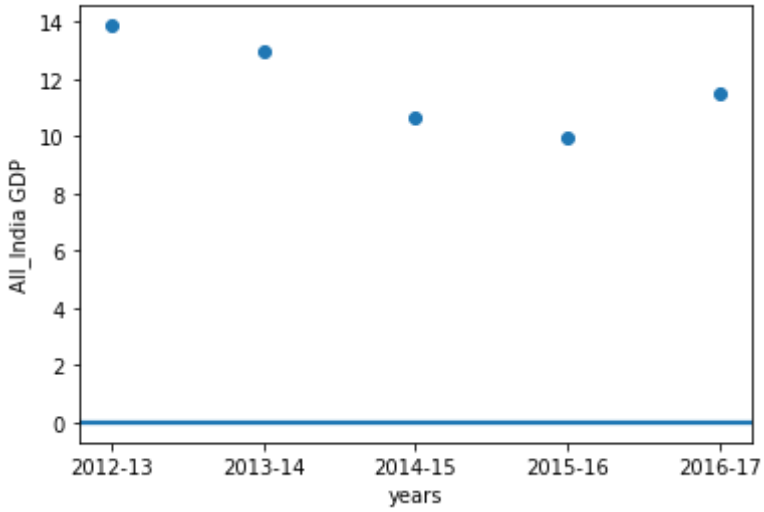
In [65]:

```
#PART 1 A -DATA 1 A :
# Q1 : a) Downloading the data and treatment of Missing values : The da
ta for Data 1A is downloaded from the link provided .
    #   b) West Bengal state does not contain any values for any of the
 items , but still i will take the data as it is so that phython code i
s used to address the missing values
     #  c) for the rest of the states  , missing values are treated as
 it is .
    # Plot a graph for rows , " % Growth over previous year" for all st
ates (not UTs) by using best fit line :

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\DATA 1A\df_year.c
sv')
df_year.head(10)
#df_year=df_year.fillna(0)


df_year=df_year.drop(['Items  Description'], axis=1)
df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Isla
nds','Chandigarh','Puducherry','West Bengal1'], axis=1)
to_del = [0,1,2,3,4,5]
df_year=df_year.drop(df_year.index[to_del])


plt.scatter(x='Duration',y='All_India GDP',data=df_year)
df_year=df_year[['Duration','All_India GDP']]
df_year=df_year[6:10]
df_year.insert(1,"years",[2012,2013,2014,2015],True)
sns.regplot(x='years',y='All_India GDP',data=df_year)
plt.show()

#inference: There was a slight decrease in the GDP growth rate of the n
```

```
ation from 2012-13 till 2014-15 and from 2015-16 onwards there was an i
ncrease in the Overall GDP of the nation
```

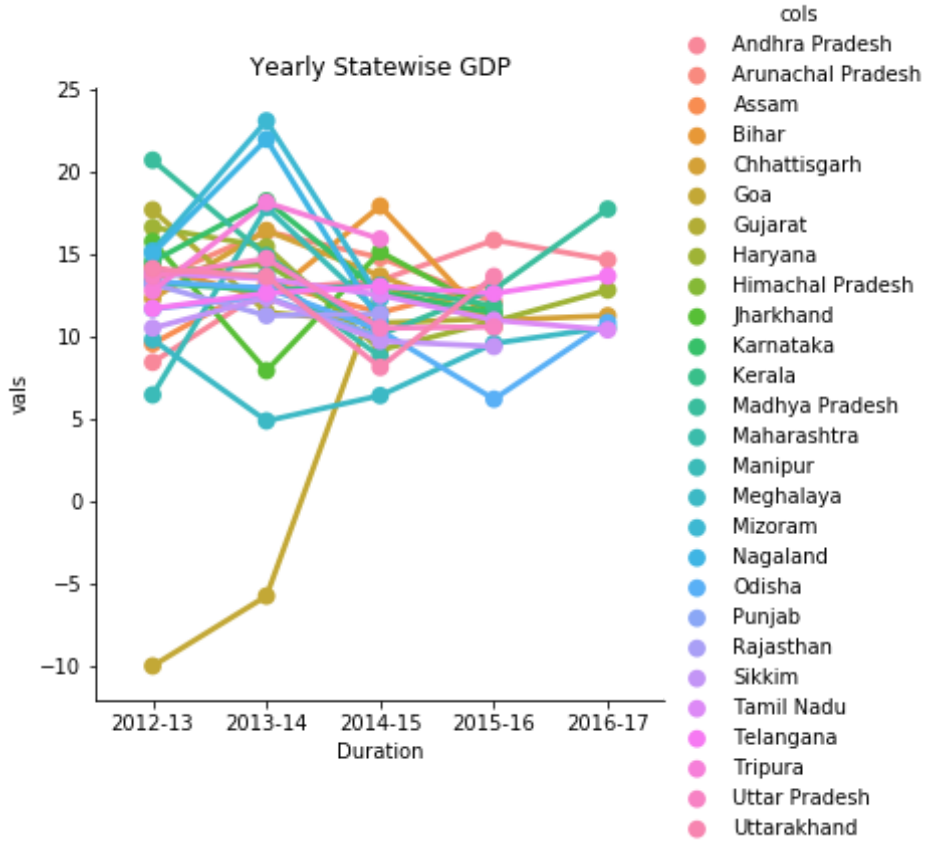


In [64]:
```
#DATA 1 A
# Yearly GDP Statewise for all years -Line Graph :
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\df_year.csv')
#df_year.head(10)
#df_year=df_year.fillna(0)
df_year=df_year.drop(['Items  Description','All_India GDP'], axis=1)

df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Isla
nds','Chandigarh','Puducherry','West Bengal1'], axis=1)
to_del = [0,1,2,3,4,5]
df_year=df_year.drop(df_year.index[to_del])

df_year = df_year.melt('Duration', var_name='cols',  value_name='vals')
g = sns.factorplot(x="Duration", y="vals", hue='cols', data=df_year,fon
```

```
tsize=10)

plt.title(" Yearly Statewise GDP ")

# inference : # inference : a)  As it is seen , Mizoram , Nagaland and
 Tripura , are the states that are performing better than other states
 in respect of the % growth of  GDP across the years
# b) From the data we can see that though there was slight decrease in
 the % GDP growth rates from 2014-15 to 2015-16 across all the states n
ation wide , but still due to a good amount of increase is shown from 2
012-15by these states , they are performing better
#c) On the other hand , states :sikkim , meghalaya and Goa fall under t
he bottom three states and has to work in most of the areas to imporve
 better.
#c1: Meghalaya and Goa showed a steep decline in the growth rate from 2
014-16 .
```

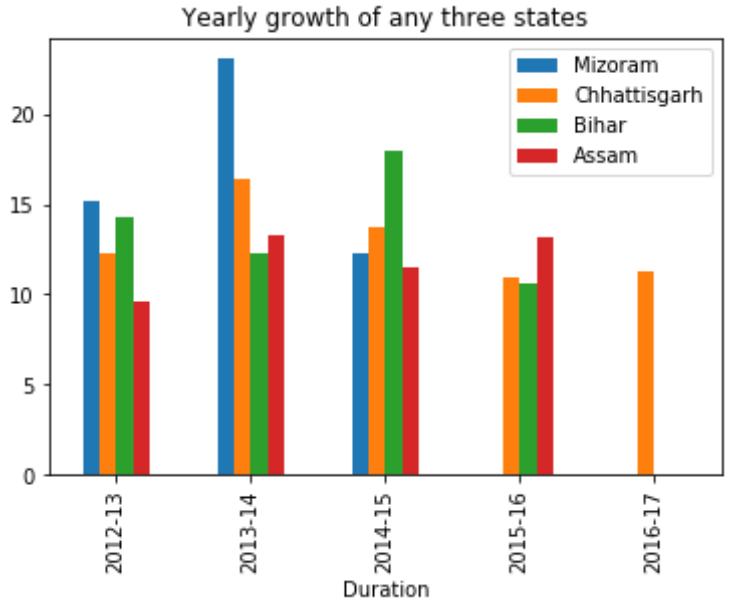
```
C:\Users\amuly\Anaconda3\lib\site-packages\seaborn\categorical.py:3666:
UserWarning: The `factorplot` function has been renamed to `catplot`. T
he original name will be removed in a future release. Please update you
r code. Note that the default `kind` in `factorplot` (`'point'`) has ch
anged `'strip'` in `catplot`.
  warnings.warn(msg)
```

Out[64]: Text(0.5, 1, ' Yearly Statewise GDP ')



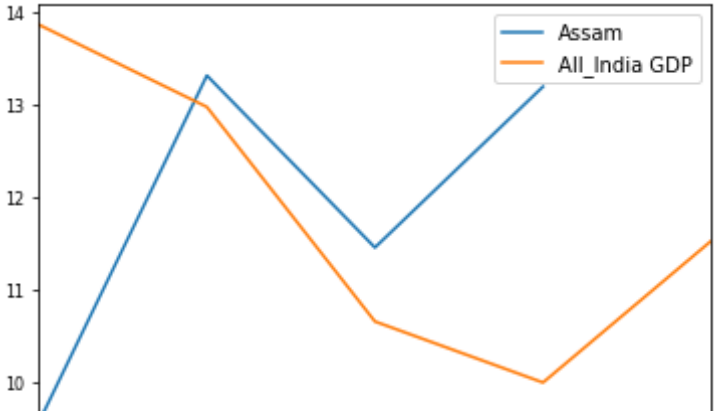
In [66]:
```
#PART 1 A -DATA 1 A :
#Q2 : e)How will you compare the growth rate of GDP of anyn two states
 :
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

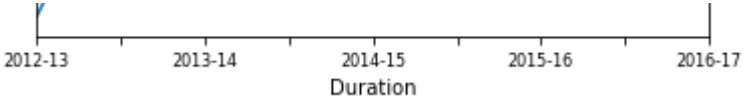
```
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\DATA 1A\df_year.csv')
#df_year.head(10)
#df_year=df_year.fillna(0)
df_year=df_year.drop(['Items  Description','All_India GDP'], axis=1)
df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Islands','Chandigarh','Puducherry','West Bengal1'], axis=1)
to_del = [0,1,2,3,4,5]
df_year=df_year.drop(df_year.index[to_del])
df_year = df_year.plot(x="Duration",y=["Mizoram","Chhattisgarh","Bihar","Assam"], kind="bar")
plt.title("Yearly growth of any three states")
plt.show()

#inference1 : as you can see Mizoram is performing well among the states chosen for 2012-2015 . There seems to be a slight increase in growth
#infeernce 2: in bihar from 2013-14 to 14-15
```



In [15]:
```
#PART 1 A -DATA 1 A :
#Q3 : what is the grwoth rate of your homestate (GDP)vs the All India G
```

```
DP:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\DATA 1A\df_year.csv')
#df_year.head(10)
#df_year=df_year.fillna(0)
df_year=df_year.drop(['Items  Description'], axis=1)
df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Islands','Chandigarh','Puducherry','West Bengal1'], axis=1)
to_del = [0,1,2,3,4,5]
df_year=df_year.drop(df_year.index[to_del])
df_year=df_year.plot(x="Duration",y=['Assam','All_India GDP'],fontsize=8)
plt.show()

#inference : a) we can see that the All India GDP was increased from the years between 2012-13 & till 2014 , porbably due to some economic changes
#made for certain sectors across the nation , and again there was some
 slight decrease in the GDP from years 2014-15 and again it followed an
 increasing trend thereafter.
#b) Assam on the other hand showed a very steep decrease in the GDP grwoth grate when compared to Nation GDP till 2016 and afterthat showed an
 increasing trend due to some focus on key areas of the state.
```



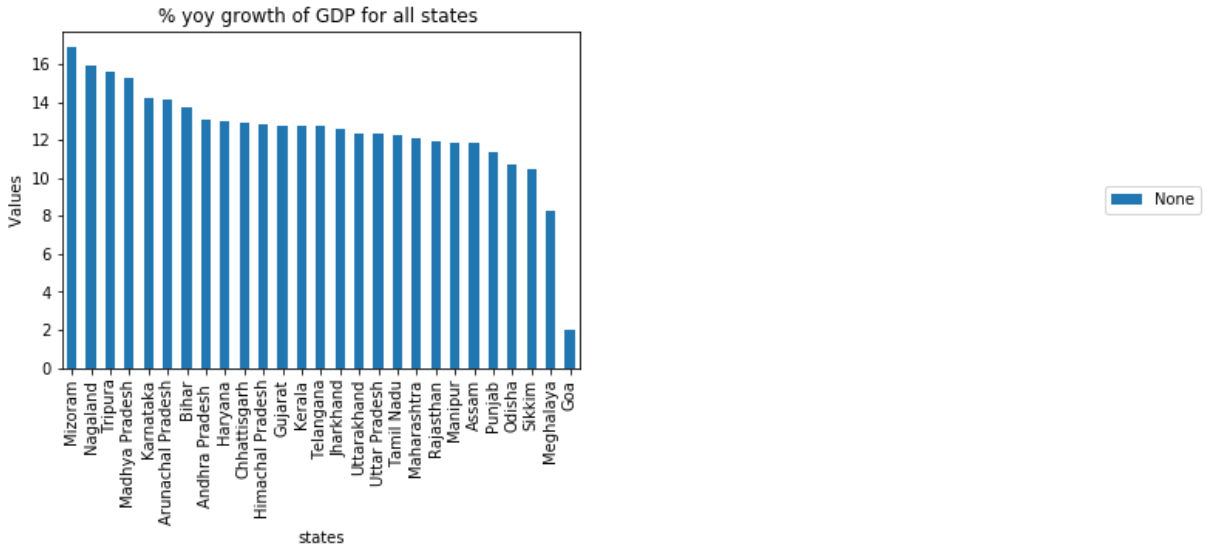


In [16]:
```
#PART 1 A -DATA 1 A :
#Q4 : which states have been consistently grwoing faster and which ones
 were strugglig . Rank Top 3 and bottom 3:

# Note : for this exercise , the average percentage Growth of GDP is ta
ken for all the years of the states individually and plotted , to find
 the  top 3 and bottom 3

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\DATA 1A\df_year.c
sv')
#df_year=df_year.fillna(0)
df_year=df_year.drop(['Items  Description'], axis=1)
df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Isla
nds','Chandigarh','Puducherry','West Bengal1','All_India GDP'], axis=1)
to_del = [0,1,2,3,4,5]
df_year=df_year.drop(df_year.index[to_del])
#df_year.head(10)
#g=[df_year.loc[6:11]]
#print(g)
ap_av=df_year.mean(axis=0)
#print(ap_av)
sort_a=ap_av.sort_values(ascending=False)
sort_b=round(sort_a,2)
print("Sorted list is ")
print(" ")
print(sort_b)
print(sort_b.plot(x=["values"],y=["states"],kind="bar"))
plt.xlabel("states")
plt.ylabel(" Values ")
plt.title("% yoy growth of GDP for all states ")
plt.legend(loc='center left', bbox_to_anchor=(2, 0.5))
```

```
plt.show()

# inference : a)  As it is seen , Mizoram , Nagaland and Tripura , are
 the states that are performing better than other states in respect of
 the % growth of  GDP across the years
# b) From the data we can see that though there was slight decrease in
 the % GDP growth rates from 2014-15 to 2015-16 across all the states n
ation wide , but still due to a good amount of increase is shown from 2
012-15by these states , they are performing better
#c) On the other hand , states :sikkim , meghalaya and Goa fall under t
he bottom three states and has to work in most of the areas to imporve
 better.
#c1: Meghalaya and Goa showed a steep decline in the growth rate from 2
014-16 .
```

```
Sorted list is

Mizoram              16.87
Nagaland             15.95
Tripura              15.61
Madhya Pradesh       15.27
Karnataka            14.23
Arunachal Pradesh    14.16
Bihar                13.76
Andhra Pradesh       13.04
Haryana              13.00
Chhattisgarh         12.93
Himachal Pradesh     12.82
Gujarat              12.77
Kerala               12.75
Telangana            12.73
Jharkhand            12.57
Uttarakhand          12.38
Uttar Pradesh        12.37
Tamil Nadu           12.25
Maharashtra          12.10
Rajasthan            11.94
Manipur              11.89
Assam                11.88
Punjab               11.35
```

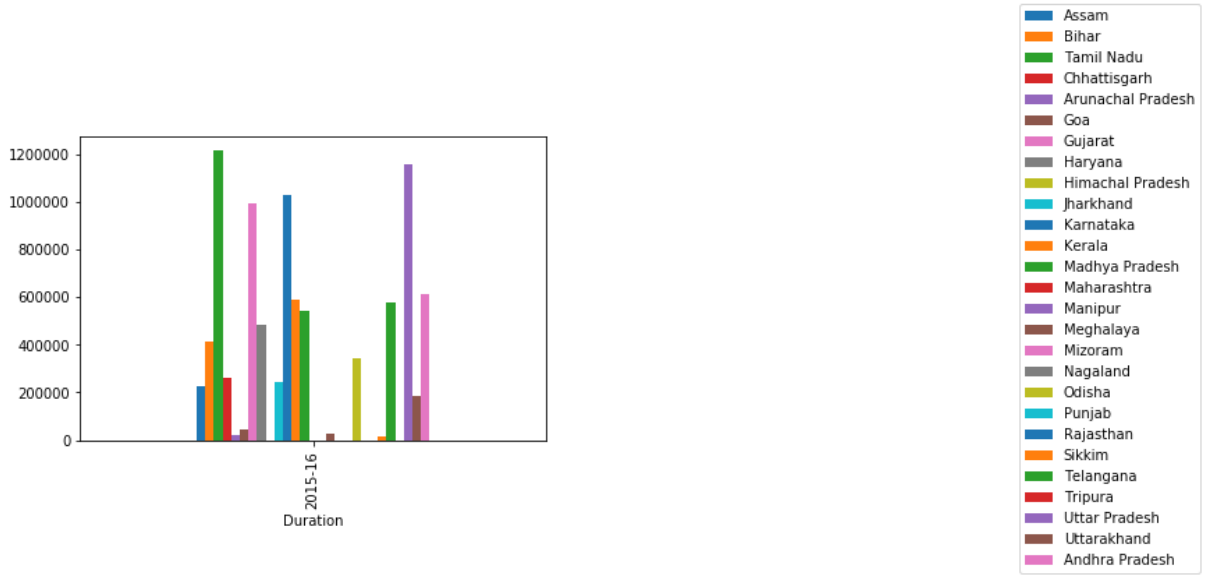
```
Odisha                10.74
Sikkim                10.49
Meghalaya              8.24
Goa                    2.02


dtype: float64
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [17]:
```
#PART 2 -DATA 1 A :
# Q5:GDP of the states for 2015-16"
#Which Plot will you use for this? Why? (Remeber to plot the graph in a
 way such as it is easier to read and compare):
#1) We use Bar Plot for this exercise as it is easier to compare the GD
P in a year for different states . Also bar chart is easier to compare
 values for a particualr category when compared to other chart types:
    #eg : Histogram is used to identify the distribution of GDP of a pa
rticular state over the years and we also cannot use trend plot as it i
s not to find the trend of the GDP over a number of years.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\DATA 1A\df_year.csv')
df_year.head(10)
#df_year=df_year.fillna(0)
df_year=df_year.drop(['Items  Description','All_India GDP'], axis=1)
df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Islands','Chandigarh','Puducherry','West Bengal1'], axis=1)
to_del = [0,1,2,3,5,6,7,8,9,10]
df_year=df_year.drop(df_year.index[to_del])
df_year = df_year.plot(x="Duration",y=["Assam","Bihar","Tamil Nadu","Chhattisgarh","Arunachal Pradesh","Goa","Gujarat","Haryana","Himachal Pradesh","Jharkhand","Karnataka","Kerala","Madhya Pradesh","Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punjab","Rajasthan","Sikkim","Telangana","Tripura","Uttar Pradesh","Uttarakhand","Andhra Pradesh "],kind="bar")
plt.legend(loc='center left', bbox_to_anchor=(2, 0.5))

#inference: this is the GDP of all states for 2015-16 , to find the top and bottom states performance , we need to take the mean values and compare.
```

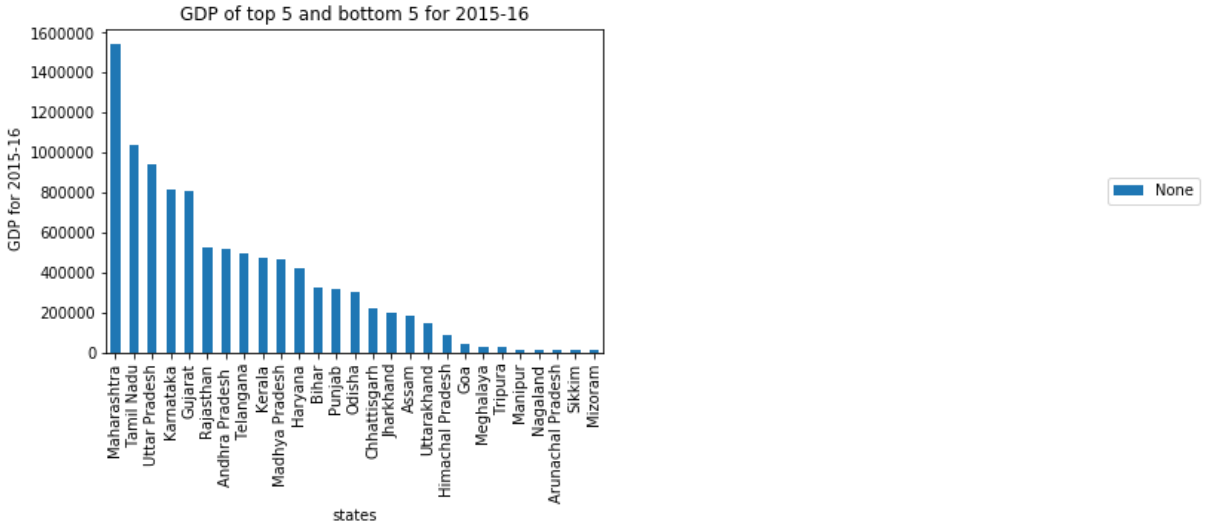
Out[17]: <matplotlib.legend.Legend at 0x1d23fede608>



In [18]:
```
#Part 2- Data -1A
#Identify the top 5 and the bottom 5 states based on total GDP:
#What insights can you draw from this graph? What states are performing
 poorly? (Remember: this will not be solely based on total GDP)


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df_year=pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\DATA 1A\df_year.c
sv')
#df_year=df_year.fillna(0)
df_year=df_year.drop(['Items  Description',"All_India GDP"], axis=1)
df_year=df_year.drop(['Delhi','Jammu & Kashmir','Andaman & Nicobar Isla
nds','Chandigarh','Puducherry','West Bengal1'], axis=1)
to_del = [6,7,8,9,10]
df_year=df_year.drop(df_year.index[to_del])
#df_year.head(10)
ap_av=df_year.mean(axis=0)
sort_a=ap_av.sort_values(ascending=False)
sort_b=round(sort_a,2)
print("Sorted list is ")
print(" ")
print(sort_b)
print(sort_b.plot(x=["values"],y=["states"],kind="bar"))
plt.xlabel("states")
plt.ylabel(" GDP for 2015-16")
plt.title("GDP of top 5 and bottom 5 for 2015-16")
plt.legend(loc='center left', bbox_to_anchor=(2, 0.5))
plt.show()
print("Mean GDP of the states in 2015-16")
print("  ")
print(sort_b.mean())
#inference:Mean GDP of the states in 2015-16 is 368884.9255 . So states
 which are above mean perform better when compared to states below mean
```

```
Sorted list is
```

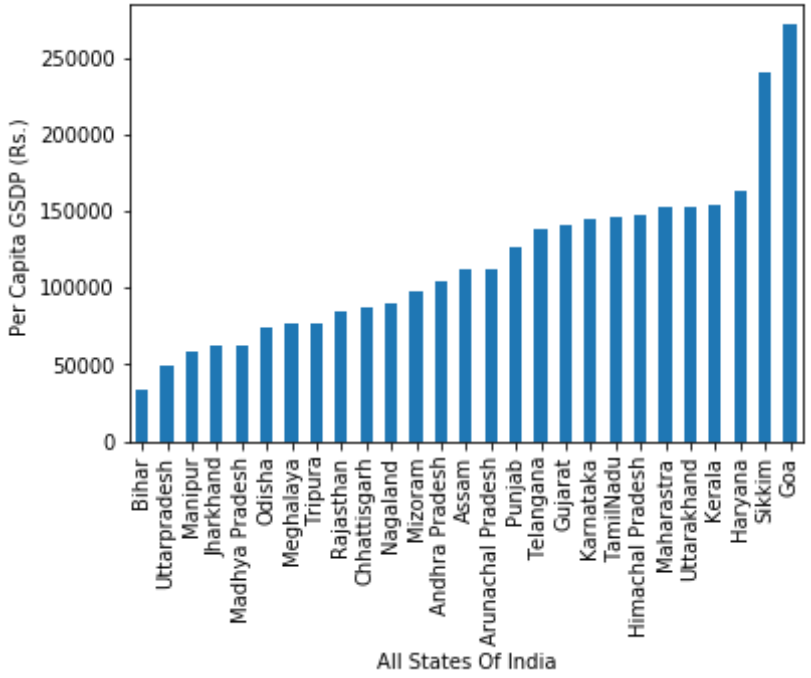
```
Maharashtra          1540265.25
Tamil Nadu           1037009.00
Uttar Pradesh         937652.80
Karnataka             812444.00
Gujarat               807413.40
Rajasthan             523091.00
Andhra Pradesh        515131.17
Telangana             492369.17
Kerala                471148.20
Madhya Pradesh        466777.17
Haryana               419212.50
Bihar                 326807.20
Punjab                316771.75
Odisha                303450.83
Chhattisgarh          221362.17
Jharkhand             194654.20
Assam                 180023.20
Uttarakhand           148650.20
Himachal Pradesh       88668.25
Goa                    40408.60
Meghalaya              24241.33
Tripura                24032.75
Manipur                15226.00
Nagaland               15121.00
Arunachal Pradesh      14751.40
Sikkim                 13842.20
Mizoram                 9368.25
dtype: float64
AxesSubplot(0.125,0.125;0.775x0.755)
```




```
Mean GDP of the states in 2015-16

368884.92555555544
```

In [29]:
```
# PART 1-B- Q1) : Find the percapita GDP of all the states for 2014-15:

import numpy as np
import pandas as pd
import glob
import os
path = (r'C:\Users\amuly\Desktop\GDP JSON\New All states')
all_files = glob.glob(path + "/*.csv")

req_columns = ['S.No.','Item','2014-15']
union_terr=['Delhi','Jammu & Kashmir','Andaman & Nicobar Islands','Chan
digarh','Puducherry','West Bengal1']
df_all_states = pd.concat([pd.read_csv(i, encoding = 'ISO8859', usecols
=req_columns).assign(State = i.split('-')[1].replace('_',' '))
                   for i in all_files if i.split('-')[2].replace('_',' ')
not in union_terr])
```

```
#plotting the GDP Per capita of all states :

df_all_states.loc[(df_all_states.Item == "Per Capita GSDP (Rs.)")].iloc
[:,2:].sort_values(by = '2014-15').set_index('State').iloc[:,-1].plot(k
ind='bar')

plt.ylabel('Per Capita GSDP (Rs.) ')

plt.xlabel('All States Of India')

plt.show()

#inference: we can find that in the year 2014-15 , Goa , Sikkim , Harya
na , Kerala & Uttarakhand are the top 5 states that performed well with
 respect to the Per apita in 2014-15 , states like Bihar , UP , Manipu
r,Jharkhand , MP are the bottom 5 states for the year2014-15
```



In [25]:
```
# workings of percapita 2014-15
df_all_states.head(50)
```

Out[25]:

	S.No.	Item	2014-15	State
0	1	Agriculture, forestry and fishing	14819416.0	Andhra Pradesh
1	1.1	Crops	7893514.0	Andhra Pradesh
2	1.2	Livestock	4309078.0	Andhra Pradesh
3	1.3	Forestry and logging	346160.0	Andhra Pradesh
4	1.4	Fishing and aquaculture	2270664.0	Andhra Pradesh
5	2	Mining and quarrying	1484300.0	Andhra Pradesh
6	Total	Primary	16303716.0	Andhra Pradesh
7	3	Manufacturing	4672266.0	Andhra Pradesh
8	4	Electricity, gas, water supply & other utility...	1151729.0	Andhra Pradesh
9	5	Construction	4664889.0	Andhra Pradesh
10	Total	Secondary	10488884.0	Andhra Pradesh
11	6	Trade, repair, hotels and restaurants	4233400.0	Andhra Pradesh
12	6.1	Trade & repair services	3716000.0	Andhra Pradesh
13	6.2	Hotels & restaurants	517400.0	Andhra Pradesh
14	7	Transport, storage, communication & services r...	5076984.0	Andhra Pradesh
15	7.1	Railways	424228.0	Andhra Pradesh
16	7.2	Road transport	2816000.0	Andhra Pradesh
17	7.3	Water transport	94200.0	Andhra Pradesh
18	7.4	Air transport	14900.0	Andhra Pradesh
19	7.5	Services incidental to transport	780200.0	Andhra Pradesh
20	7.6	Storage	18700.0	Andhra Pradesh
21	7.7	Communication & services related to broadcasting	928756.0	Andhra Pradesh

	S.No.	Item	2014-15	State
22	8	Financial services	1900863.0	Andhra Pradesh
23	9	Real estate, ownership of dwelling & professio...	4405409.0	Andhra Pradesh
24	10	Public administration	2200897.0	Andhra Pradesh
25	11	Other services	4215389.0	Andhra Pradesh
26	Total	Tertiary	22032942.0	Andhra Pradesh
27	12	TOTAL GSVA at basic prices	48825542.0	Andhra Pradesh
28	13	Taxes on Products	5512100.0	Andhra Pradesh
29	14	Subsidies on products	1690800.0	Andhra Pradesh
30	15	Gross State Domestic Product	52646842.0	Andhra Pradesh
31	16	Population ('00)	501510.0	Andhra Pradesh
32	17	Per Capita GSDP (Rs.)	104977.0	Andhra Pradesh
0	1	Agriculture, forestry and fishing	686117.0	Arunachal Pradesh
1	1.1	Crops	415520.0	Arunachal Pradesh
2	1.2	Livestock	38387.0	Arunachal Pradesh
3	1.3	Forestry and logging	224017.0	Arunachal Pradesh
4	1.4	Fishing and aquaculture	8193.0	Arunachal Pradesh
5	2	Mining and quarrying	30842.0	Arunachal Pradesh
6	Total	Primary	716959.0	Arunachal Pradesh
7	3	Manufacturing	26120.0	Arunachal Pradesh
8	4	Electricity, gas, water supply & other utility...	113527.0	Arunachal Pradesh
9	5	Construction	147842.0	Arunachal Pradesh
10	Total	Secondary	287489.0	Arunachal Pradesh
11	6	Trade, repair, hotels and restaurants	60421.0	Arunachal Pradesh
12	6.1	Trade & repair services	56796.0	Arunachal Pradesh
13	6.2	Hotels & restaurants	3625.0	Arunachal Pradesh

	S.No.	Item	2014-15	State
14	7	Transport, storage, communication & services r...	35203.0	Arunachal Pradesh
15	7.1	Railways	59.0	Arunachal Pradesh
16	7.2	Road transport	15467.0	Arunachal Pradesh

In [40]:
```
# Part 1- B -Q2) Find the ratio of Highest percapita GDP to lowest Per
 Capita GDP : 2014-15:

#Highest Percapita state based on above : Goa
#Lowest percapita state : Bihar

r=df_all_states.loc[(df_all_states.State == "Goa")&(df_all_states.Item
== "Per Capita GSDP (Rs.)")].iloc[:,-2].T[32] / df_all_states.loc[(df_a
ll_states.State == "Bihar")&(df_all_states.Item == "Per Capita GSDP (R
s.)")].iloc[:,-2].T[32]
print(r)
print(round(r,3))
```

```
8.004741709371503
8.005
```

In [54]:
```
#Part 1- B:Q3) Find the percentage contribution of the primary , second
ary , territory sectors as a percentage contribution of GDP:



df_total_GDP = df_all_states.loc[(df_all_states.Item == "Gross State Do
mestic Product")][['2014-15','State']].rename(columns={'2014-15':'GSDP'
})

df_total_GDP.head(70)

#getting the GSDP of all states for 2014-15 for comparing within the se
ctors:
```

Out[54]:

	GSDP	State

30	52646842.0	Andhra Pradesh

	GSDP	State
30	1676119.0	Arunachal Pradesh
30	1676119.0	Assam
30	37391988.0	Bihar
30	23498180.0	Chhattisgarh
30	4063307.0	Goa
30	89502727.0	Gujarat
30	43746207.0	Haryana
30	10436879.0	Himachal Pradesh
30	21710718.0	Jharkhand
30	92178806.0	Karnataka
30	52600230.0	Kerala
30	48198169.0	Madhya Pradesh
30	179212165.0	Maharastra
30	1804276.0	Manipur
30	2440807.0	Meghalaya
30	1155933.0	Mizoram
30	1841424.0	Nagaland
30	32197092.0	Odisha
30	36801089.0	Punjab
30	61219447.0	Rajasthan
30	1520933.0	Sikkim
30	109256373.0	TamilNadu
30	51117765.0	Telangana
30	2966662.0	Tripura

30	16198529.0	Uttarakhand

	GSDP	State
30	104337115.0	Uttarpradesh

In [4]:
```
#Part 1- B:Q3) Find the percentage contribution of the primary , second
ary , territory sectors as a percentage contribution of GDP- finding ou
t the total of these three sectors vs State-workings
df_primary_sector = df_all_states.loc[(df_all_states.Item == "Primary"
)][['2014-15','State']].rename(columns={'2014-15':'Primary_GSVA'})
df_primary_sector.head(50)
df_primary_sector = pd.merge(df_primary_sector, df_all_states.loc[(df_a
ll_states.Item == "Secondary")][['2014-15','State']], how = 'inner', on
 = 'State').rename(columns={'2014-15':'Secondary_GSVA'})
df_primary_sector = pd.merge(df_primary_sector, df_all_states.loc[(df_a
ll_states.Item == "Tertiary")][['2014-15','State']], how = 'inner', on
= 'State').rename(columns={'2014-15':'Tertiary_GSVA'})

df_primary_sector.head(40)
```

Out[4]:

	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA
0	16303716.0	Andhra Pradesh	10488884.0	22032942.0
1	716959.0	Arunachal Pradesh	287489.0	631844.0
2	716959.0	Assam	287489.0	631844.0
3	8019997.0	Bihar	5984896.0	22179969.0
4	6400817.0	Chhattisgarh	8238886.0	7588778.0
5	312129.0	Goa	1547536.0	1738217.0
6	15887187.0	Gujarat	33023538.0	30220377.0
7	8040424.0	Haryana	12561411.0	19226568.0
8	1548366.0	Himachal Pradesh	4119162.0	4133326.0
9	5248354.0	Jharkhand	6241471.0	8133341.0

	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA
10	12066304.0	Karnataka	20484404.0	50490630.0
11	6489442.0	Kerala	12070040.0	29673778.0
12	17854020.0	Madhya Pradesh	10044889.0	18117360.0
13	21758383.0	Maharastra	47445207.0	88631076.0
14	383140.0	Manipur	220173.0	1177334.0
15	451050.0	Meghalaya	637942.0	1200655.0
16	225598.0	Mizoram	270072.0	637619.0
17	616178.0	Nagaland	212361.0	992956.0
18	9009306.0	Odisha	8989693.0	12256258.0
19	9296070.0	Punjab	7904914.0	16717805.0
20	19113780.0	Rajasthan	13028794.0	26015812.0
21	138776.0	Sikkim	845253.0	483103.0
22	13329774.0	TamilNadu	32841892.0	53343788.0
23	9133354.0	Telangana	9924001.0	28471410.0
24	942216.0	Tripura	484393.0	1484709.0
25	1845972.0	Uttarakhand	7642865.0	5587975.0
26	25999255.0	Uttarpradesh	25548724.0	45968959.0

In [5]:
```
# Merging the dataframes: primary sector ,total_GDP to get the result a
s shown in below table - workings
df_total_GDP = df_all_states.loc[(df_all_states.Item == "Gross State Do
mestic Product")][['2014-15','State']].rename(columns={'2014-15':'GSDP'
})
df_primary_sector = pd.merge(df_primary_sector, df_total_GDP, how = 'in
ner', on = 'State')

df_primary_sector.head()
```

Out[5]:

	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA	GSDP

	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA	GSDP
0	16303716.0	Andhra Pradesh	10488884.0	22032942.0	52646842.0
1	716959.0	Arunachal Pradesh	287489.0	631844.0	1676119.0
2	716959.0	Assam	287489.0	631844.0	1676119.0
3	8019997.0	Bihar	5984896.0	22179969.0	37391988.0
4	6400817.0	Chhattisgarh	8238886.0	7588778.0	23498180.0

In [17]:
```
#Part 1- B:Q3) Find the percentage contribution of the primary , second
ary , territory sectors as a percentage contribution of GDP:



df_primary_sector['%_Primary_Contribution'] = (df_primary_sector['Prima
ry_GSVA']/df_primary_sector['GSDP'])*100
df_primary_sector['%_Secondary_Contribution'] = (df_primary_sector['Sec
ondary_GSVA']/df_primary_sector['GSDP'])*100
df_primary_sector['%_Tertiary_Contribution'] = (df_primary_sector['Tert
iary_GSVA']/df_primary_sector['GSDP'])*100
df_primary_sector['TotalPrimarysecondaryteritary%'] = df_primary_sector
['%_Primary_Contribution']+df_primary_sector['%_Secondary_Contribution'
]+df_primary_sector['%_Tertiary_Contribution']
df_primary_sector = df_primary_sector.sort_values(by='TotalPrimarysecon
daryteritary%',ascending=False)

df_primary_sector.head(100)
```

Out[17]:

	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA	GSDP	%_Primary_Contril
17	616178.0	Nagaland	212361.0	992956.0	1841424.0	33.4
14	383140.0	Manipur	220173.0	1177334.0	1804276.0	21.2
24	942216.0	Tripura	484393.0	1484709.0	2966662.0	31.7
16	225598.0	Mizoram	270072.0	637619.0	1155933.0	19.5

	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA	GSDP	%_Primary_Contril
1	716959.0	Arunachal Pradesh	287489.0	631844.0	1676119.0	42.7
2	716959.0	Assam	287489.0	631844.0	1676119.0	42.7
3	8019997.0	Bihar	5984896.0	22179969.0	37391988.0	21.4
21	138776.0	Sikkim	845253.0	483103.0	1520933.0	9.1
12	17854020.0	Madhya Pradesh	10044889.0	18117360.0	48198169.0	37.0
20	19113780.0	Rajasthan	13028794.0	26015812.0	61219447.0	31.2
4	6400817.0	Chhattisgarh	8238886.0	7588778.0	23498180.0	27.2
18	9009306.0	Odisha	8989693.0	12256258.0	32197092.0	27.9
8	1548366.0	Himachal Pradesh	4119162.0	4133326.0	10436879.0	14.8
15	451050.0	Meghalaya	637942.0	1200655.0	2440807.0	18.4
26	25999255.0	Uttarpradesh	25548724.0	45968959.0	104337115.0	24.9
25	1845972.0	Uttarakhand	7642865.0	5587975.0	16198529.0	11.3
23	9133354.0	Telangana	9924001.0	28471410.0	51117765.0	17.8
0	16303716.0	Andhra Pradesh	10488884.0	22032942.0	52646842.0	30.9
19	9296070.0	Punjab	7904914.0	16717805.0	36801089.0	25.2
11	6489442.0	Kerala	12070040.0	29673778.0	52600230.0	12.3
22	13329774.0	TamilNadu	32841892.0	53343788.0	109256373.0	12.2
7	8040424.0	Haryana	12561411.0	19226568.0	43746207.0	18.3
9	5248354.0	Jharkhand	6241471.0	8133341.0	21710718.0	24.1
10	12066304.0	Karnataka	20484404.0	50490630.0	92178806.0	13.0
5	312129.0	Goa	1547536.0	1738217.0	4063307.0	7.6
6	15887187.0	Gujarat	33023538.0	30220377.0	89502727.0	17.7

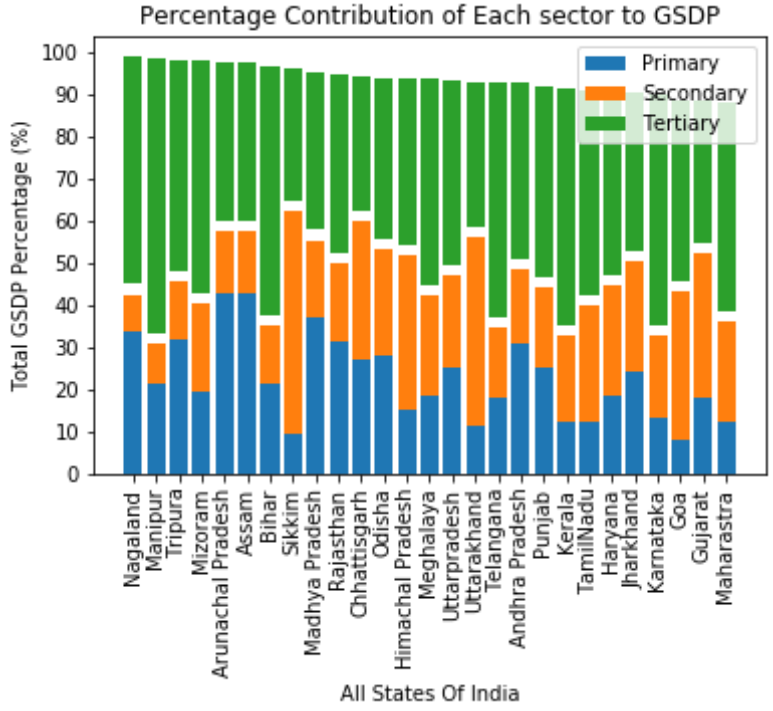
	Primary_GSVA	State	Secondary_GSVA	Tertiary_GSVA	GSDP	%_Primary_Contril
13	21758383.0	Maharastra	47445207.0	88631076.0	179212165.0	12.1

In [20]:

```
# Plotting a Stacke bar-chart to represent the percentage contribution
 of primary, secondary and tertiary sectors as a percentage of total GD
P for all the states.
#Part 1- B:Q3) Find the percentage contribution of the primary , second
ary , territory sectors as a percentage contribution of GDP:
# here we are using the stacked bar chart as this shows 100%  clarity o
n the sector wise contribution across all states for a particular year.
#unlike other techniques , this will show how much contribution each se
ctor is  made in a stack /bunch of sectors , it will also help us to re
ad the
#huge number of data points at one glance .
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns


Primary = df_primary_sector['%_Primary_Contribution']
Secondary =df_primary_sector['%_Secondary_Contribution']
Tertiary = df_primary_sector['%_Tertiary_Contribution']
States = df_primary_sector['State']

p1 = plt.bar(States, Primary)

p2 = plt.bar(States, Secondary, bottom=Primary)

p3 = plt.bar(States, Tertiary, bottom=np.array(Primary)+np.array(Second
ary))

plt.ylabel('Total GSDP Percentage (%)')

plt.title('Percentage Contribution of Each sector to GSDP')
```

```
plt.xticks(States,rotation=90)

plt.yticks(np.arange(0, 110, 10)); plt.xlabel('All States Of India')

plt.legend((p1[0], p2[0], p3[0]), ('Primary', 'Secondary', 'Tertiary'))

plt.show()

#inference of percentage contribution of the sectors ->Primary , Second
ary , Territory to the Overall GSDP per state for 2014-15:

# 1) AS we can see that out of the three sectors , Teritary sector cont
ributes more to the GDP income across all states for 2014-15, the contr
ibution is close to 45% when compared to the other sectors
#2) Out ot the teritary sectors , Trade and transportation services and
 communication services contribute mostly to the growth
#3)Agricultural sector seems to be growing good in some states like MP,
Arunachal Pradesh and Assam , Nagaland when compared to other states .
#4) a)Key focus areas : we can see that though teritary sector is contr
ibuting more to the GDP , however we are lagging behind in other
 #  b)Scope of imporvement in Primary Sector : there can be comsiderabl
e imporvement in states of Sikkim, Himachal Pradesh , uttarakhand , Goa
 and Maharastra where focus on Agriculture and allied products , and fo
restry is needed.on the other hand , for such states which lag behind t
his sector, if we can improve the production of agricultural products ,
 increase in exports , the percapita income of the families that earn t
heit livelihood in this sector can be improved.
 # c-1) Scope of Imporvement in Secondary Sector:  in the secondary cat
egory we can see that states like Sikkim , Uttarakhad , Himachal prades
h are contributing more to GDP ,where as Nagaland aand Maniput still ha
ve to improve in this sector that are the least .
 # c-2) as you can on an average almost all the states are doing equall
y better in terms of secondary sector(nearly 13 states are above the av
erage %) , with the exception of Sikkim bagging nearly 55% of the total
 contribution of the secondary sector within.
 # c-4) Utility services production like Gas , Electricity units have t
o be established more and still the manufaturing sector has to be impor
ved with many industries getting established , thereby increasing the p
```

```
roduction , creation of empoloyment opportunities which on the other ha
nd will increase the percapita of the nation.
 # d-1)Scope of imporvement in Teritary Sector :unlike the Primary sect
or , the Teritary sector is performing better and contribution to the o
ver GDP is nearly 50% of the total GDP percentage growth.We can see sta
tes like Manipur , Bihar , Kerala are performing better and it is also
 seen that nearly 50% of the states are above average with the  percent
age growth of this sector. This is basically due to the reason that the
re is good performance seen in the Road ,Transport and communications s
ector.
 # e) on the whole states like  Sikkim ,Chattisgarh , Gujarat, Uttarakh
and , Jahrkhand are the five states that lie in the bottom five with re
spect to the average percentage growth as well as the GDP growth rate f
or the year 2014-15. So more focus should be kept onthese states
 # e contd:) to improve in all the sectors .
```



In [37]:
```
#Part  1-B : Q4) workings on the quartiles based on 27 states :
```

```
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
states_per_capita_sorted.head(30)
```

Out[37]:

	per_capita_GSDP	State
32	33954.0	Bihar
32	49450.0	Uttarpradesh
32	58442.0	Manipur
32	62091.0	Jharkhand
32	62989.0	Madhya Pradesh
32	73979.0	Odisha
32	76228.0	Meghalaya
32	77358.0	Tripura
32	84837.0	Rajasthan
32	86860.0	Chhattisgarh
32	89607.0	Nagaland
32	97687.0	Mizoram
32	104977.0	Andhra Pradesh
32	112718.0	Assam
32	112718.0	Arunachal Pradesh
32	126606.0	Punjab
32	139035.0	Telangana

	per_capita_GSDP	State
32	141263.0	Gujarat
32	145141.0	Karnataka
32	146503.0	TamilNadu
32	147330.0	Himachal Pradesh
32	152853.0	Maharastra
32	153076.0	Uttarakhand
32	154778.0	Kerala
32	164077.0	Haryana
32	240274.0	Sikkim
32	271793.0	Goa

In [48]:
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories :
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)

q1=round(27*0.20)# First quartlle
q2=round(27*0.50)# Second quartile
q3=round(27*0.85)# Third quartile
q4=round(27*1)# Fourth quartile



c4 = states_per_capita_sorted.iloc[:q1,:] # 0-20 percentile states
```

```
print(c4)

c3 = states_per_capita_sorted.iloc[q1:q2,:] # 20-50 percentile states
print(c3)
c2 = states_per_capita_sorted.iloc[q2:q3,:] # 50-85 percentile states
print(c2)
c1 = states_per_capita_sorted.iloc[q3:q4,:] # 85-100 percentile states
print(c1)
# inference : we have identified the categories of the states that fall
 under the percentiles C1,C2,C3&c4 respectively.
```

```
    per_capita_GDP           State
32         33954.0           Bihar
32         49450.0    Uttarpradesh
32         58442.0         Manipur
32         62091.0       Jharkhand
32         62989.0  Madhya Pradesh
    per_capita_GDP           State
32         73979.0          Odisha
32         76228.0       Meghalaya
32         77358.0         Tripura
32         84837.0       Rajasthan
32         86860.0    Chhattisgarh
32         89607.0        Nagaland
32         97687.0         Mizoram
32        104977.0  Andhra Pradesh
32        112718.0           Assam
    per_capita_GDP              State
32        112718.0  Arunachal Pradesh
32        126606.0             Punjab
32        139035.0          Telangana
32        141263.0            Gujarat
32        145141.0          Karnataka
32        146503.0          TamilNadu
32        147330.0   Himachal Pradesh
32        152853.0         Maharastra
32        153076.0        Uttarakhand
    per_capita_GDP    State
32        154778.0   Kerala
32        164077.0  Haryana
```

```
32          240274.0   Sikkim
32          271793.0      Goa
```

In [49]:

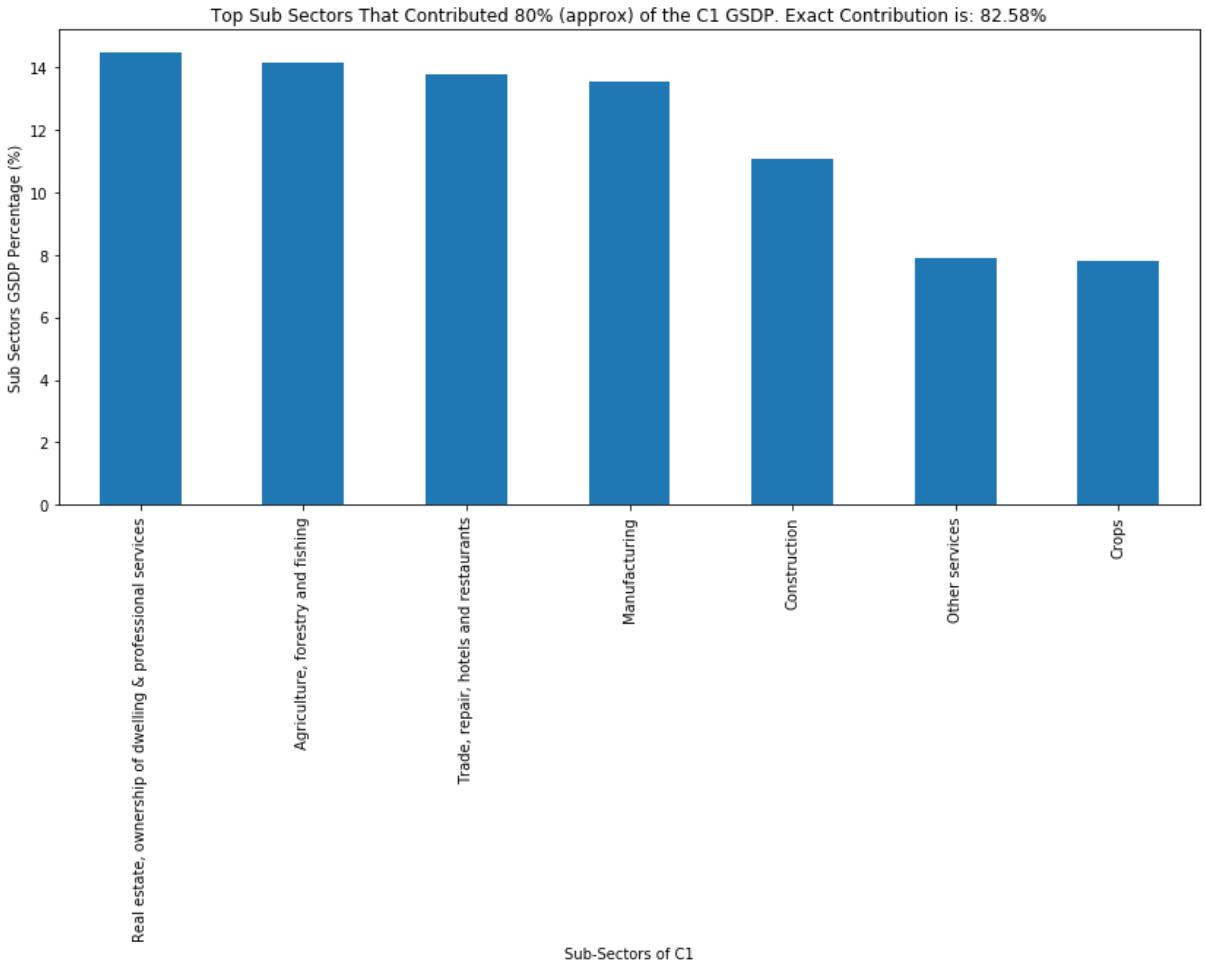
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories :c1->85 to 100 percent
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)

q1=round(27*0.20)# First quartlle
q2=round(27*0.50)# Second quartile
q3=round(27*0.85)# Third quartile
q4=round(27*1)# Fourth quartile

c1 = states_per_capita_sorted.iloc[q3:q4,:] # 85-100 percentile states
#get the subsectors for C1:
df_C1 = df_all_states.loc[df_all_states.State.isin(c1.State)&(df_all_st
ates['S.No.']!='Total')&
        (~df_all_states['Item'].isin(['TOTAL GSVA at basic prices','Tax
es on Products','Subsidies on products',"Population ('00)",'Per Capita
 GSDP (Rs.)']))]
#Retaining the necessary fields and sorting them :
df_C1 = df_C1[['Item','2014-15']].groupby(by='Item').sum().sort_values(
by='2014-15',ascending=False).reset_index()
df_C1['%_of_GSDP_Contribution'] = df_C1['2014-15']/(df_C1['2014-15'][0
])*100 # here index index 0 has GSDP since we have sorted in descending
 order
#getting 80% contribution sub sectors :
start =1; End = 4 # Taking first top 3 sectors initially to check wheth
er it contributes approximately 80%. Starting with 1 to avoid first row
 which is GSDP
```

```
while df_C1.iloc[start:End ,-1].sum() <= 78: #considering anything less
 than or equal to 78% does not contribute 80% approximately, only equal
 to greater than 79% does.
    End = End+1
# Contribution of subsectors approximately 80% For category C1 to the t
otal GSDP are as follows

C1_Sub_Sectors_contributes_80_percent_apprx = df_C1[['Item','%_of_GSDP_
Contribution']].iloc[start:End].append({'Item':'ABOVE C1 SUB-SECTORS EX
ACT CONTRIBUTION =','%_of_GSDP_Contribution':round(df_C1.iloc[start:End
 ,-1].sum(),2)},ignore_index=True).rename(columns={'Item':'C1_Sub_Secto
rs_that_contributes_80%_approximately_to_GSDP_in_Total'})
print(C1_Sub_Sectors_contributes_80_percent_apprx)
#inference : we can see that Real estate , Agriculture , Trading , Repa
irs , Manfacturing , Construction , Otherservices and crops contribute
 to nearly 80% of the value
# we can also find the similar trend while working with the sector wise
 .
```

```
  C1_Sub_Sectors_that_contributes_80%_approximately_to_GSDP_in_Total  \
0  Real estate, ownership of dwelling & professio...
1                   Agriculture, forestry and fishing
2               Trade, repair, hotels and restaurants
3                                       Manufacturing
4                                        Construction
5                                      Other services
6                                               Crops
7           ABOVE C1 SUB-SECTORS EXACT CONTRIBUTION =

   %_of_GSDP_Contribution
0               14.461049
1               14.119213
2               13.730076
3               13.498187
4               11.051090
5                7.907258
6                7.811695
7               82.580000
```

In [50]:

```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C1-85to100 percent
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)

plt.figure(figsize=(14,6))

C1_Sub_Sectors_contributes_80_percent_apprx.set_index("C1_Sub_Sectors_t
hat_contributes_80%_approximately_to_GSDP_in_Total").iloc[:-1,:]['%_of_
GSDP_Contribution'].plot(kind='bar')

plt.ylabel('Sub Sectors GSDP Percentage (%)'); plt.xlabel('Sub-Sectors
 of C1')

plt.title('Top Sub Sectors That Contributed 80% (approx) of the C1 GSD
P. Exact Contribution is: {0}%'.format(C1_Sub_Sectors_contributes_80_pe
rcent_apprx.iloc[-1:,-1:].values[0][0]))

plt.show()

#inference : we can see that Real estate , Agriculture , Trading , Repa
irs , Manfacturing , Construction , Otherservices and crops contribute
 to nearly 80% of the value
# we can also find the similar trend while working with the sector wise
 .
```



In [52]:
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C2:-50 to 85 percent
# Similarly we write for C2:
import numpy as np
import pandas as pd
```

```
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)
c2 = states_per_capita_sorted.iloc[q2:q3,:] # 50-85 percentile states

df_C2 = df_all_states.loc[df_all_states.State.isin(c2.State)&(df_all_st
ates['S.No.']!='Total')&
        (~df_all_states['Item'].isin(['TOTAL GSVA at basic prices','Tax
es on Products','Subsidies on products',"Population ('00)",'Per Capita
 GSDP (Rs.)']))]
df_C2 = df_C2[['Item','2014-15']].groupby(by='Item').sum().sort_values(
by='2014-15',ascending=False).reset_index()
df_C2['%_of_GSDP_Contribution'] = df_C2['2014-15']/(df_C2['2014-15'][0
])*100
start =1; End = 4
while df_C2.iloc[start:End ,-1].sum() <= 78:
    End = End+1
C2_Sub_Sectors_contributes_80_percent_apprx = df_C2[['Item','%_of_GSDP_
Contribution']].iloc[start:End].append({'Item':'ABOVE C2 SUB-SECTORS EX
ACT CONTRIBUTION =','%_of_GSDP_Contribution':round(df_C2.iloc[start:End
 ,-1].sum(),2)},ignore_index=True).rename(columns={'Item':'C2_Sub_Secto
rs_that_contributes_80%_approximately_to_GSDP_in_Total'})

C2_Sub_Sectors_contributes_80_percent_apprx
```

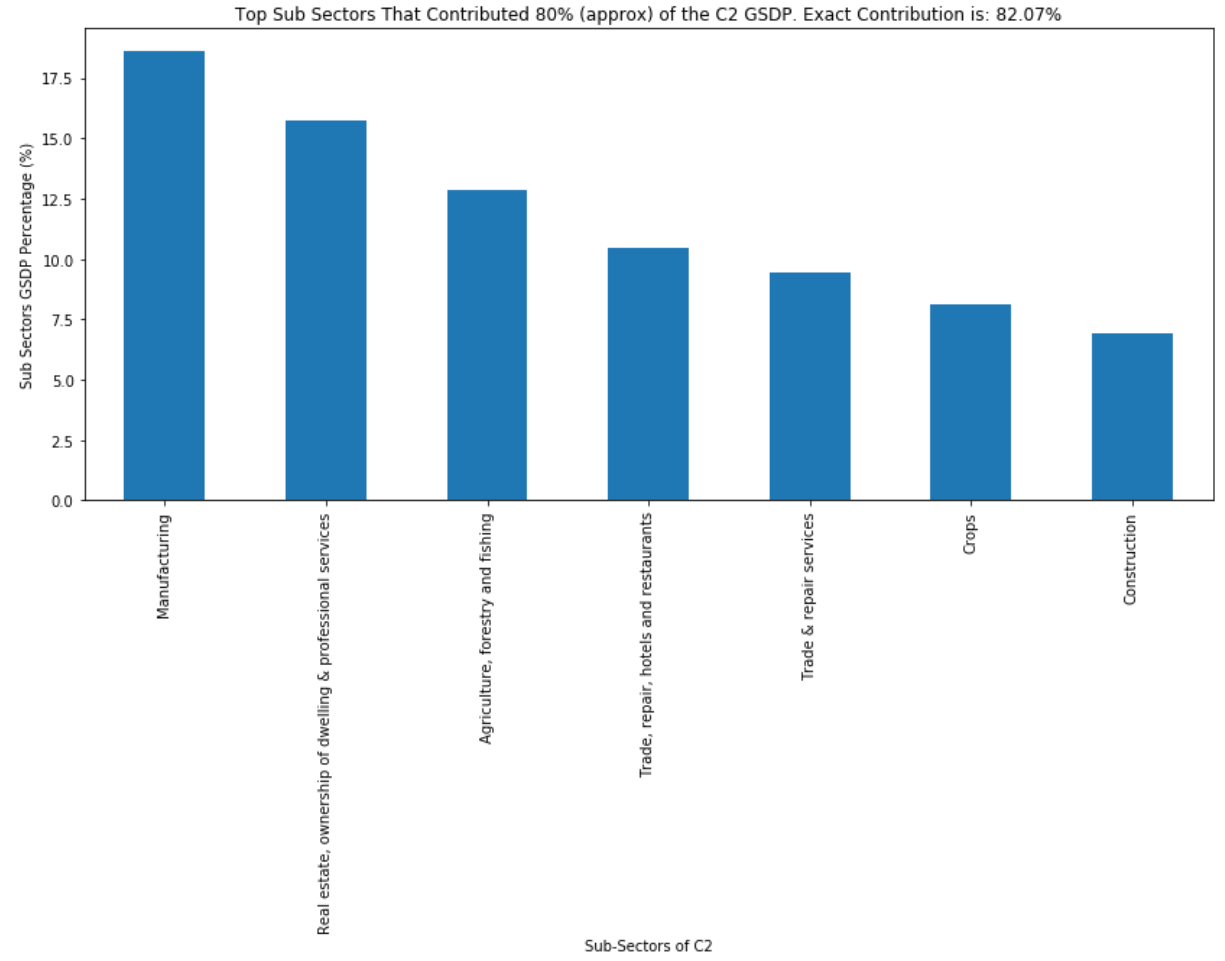
Out[52]:

	C2_Sub_Sectors_that_contributes_80%_approximately_to_GSDP_in_Total	%_of_GSDP_Contribution
0	Manufacturing	18.622130
1	Real estate, ownership of dwelling & professio...	15.710184
2	Agriculture, forestry and fishing	12.825977
3	Trade, repair, hotels and restaurants	10.443537
4	Trade & repair services	9.422608

	C2_Sub_Sectors_that_contributes_80%_approximately_to_GSDP_in_Total	%_of_GSDP_Contributior
5	Crops	8.109086
6	Construction	6.932967
7	ABOVE C2 SUB-SECTORS EXACT CONTRIBUTION =	82.070000

In [53]:
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C2:-50 to 85 percent
# Similarly we write for C2:
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)
c2 = states_per_capita_sorted.iloc[q2:q3,:] # 50-85 percentile states
plt.figure(figsize=(14,6))

C2_Sub_Sectors_contributes_80_percent_apprx.set_index("C2_Sub_Sectors_t
hat_contributes_80%_approximately_to_GSDP_in_Total").iloc[:-1,:]['%_of_
GSDP_Contribution'].plot(kind='bar')

plt.ylabel('Sub Sectors GSDP Percentage (%)');plt.xlabel('Sub-Sectors o
f C2')

plt.title('Top Sub Sectors That Contributed 80% (approx) of the C2 GSD
P. Exact Contribution is: {0}%'.format(C2_Sub_Sectors_contributes_80_pe
rcent_apprx.iloc[-1:,-1:].values[0][0]))

plt.show()

#inference : we can see that Manufacture , Real estate  agriculture and
```

```
 forestry  , Trade , repaairs , hotels , crops & Construction contribut
e to nearly 80% of the value
# we can also find the similar trend while working with the sector wise
 , also to note that Real estate , agriculture along with trade play a
 major role to the contribution even in this  sector as well.
# here hotel and restaraunts play more role along with construction . s
o we need to work towards imporving the scope of tourism and related ac
tiviies that will help us in improving business of restaurants and also
 focus more on the manfacturing sector as well to gain more income thro
ugh external and internal business
```



In [54]:
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C3:-20-50 percent
# Similarly we write for C3:

import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)
df_C3 = df_all_states.loc[df_all_states.State.isin(c3.State)&(df_all_st
ates['S.No.']!='Total')&
        (~df_all_states['Item'].isin(['TOTAL GSVA at basic prices','Tax
es on Products','Subsidies on products',"Population ('00)",'Per Capita
 GSDP (Rs.)']))]
df_C3 = df_C3[['Item','2014-15']].groupby(by='Item').sum().sort_values(
by='2014-15',ascending=False).reset_index()
df_C3['%_of_GSDP_Contribution'] = df_C3['2014-15']/(df_C3['2014-15'][0
])*100
start =1; End = 4
while df_C3.iloc[start:End ,-1].sum() <= 78:
    End = End+1
    C3_Sub_Sectors_contributes_80_percent_apprx = df_C3[['Item','%_of_G
SDP_Contribution']].iloc[start:End].append({'Item':'ABOVE C3 SUB-SECTOR
S EXACT CONTRIBUTION =','%_of_GSDP_Contribution':round(df_C3.iloc[start
:End ,-1].sum(),2)},ignore_index=True).rename(columns={'Item':'C3_Sub_S
ectors_that_contributes_80%_approximately_to_GSDP_in_Total'})

C3_Sub_Sectors_contributes_80_percent_apprx
```

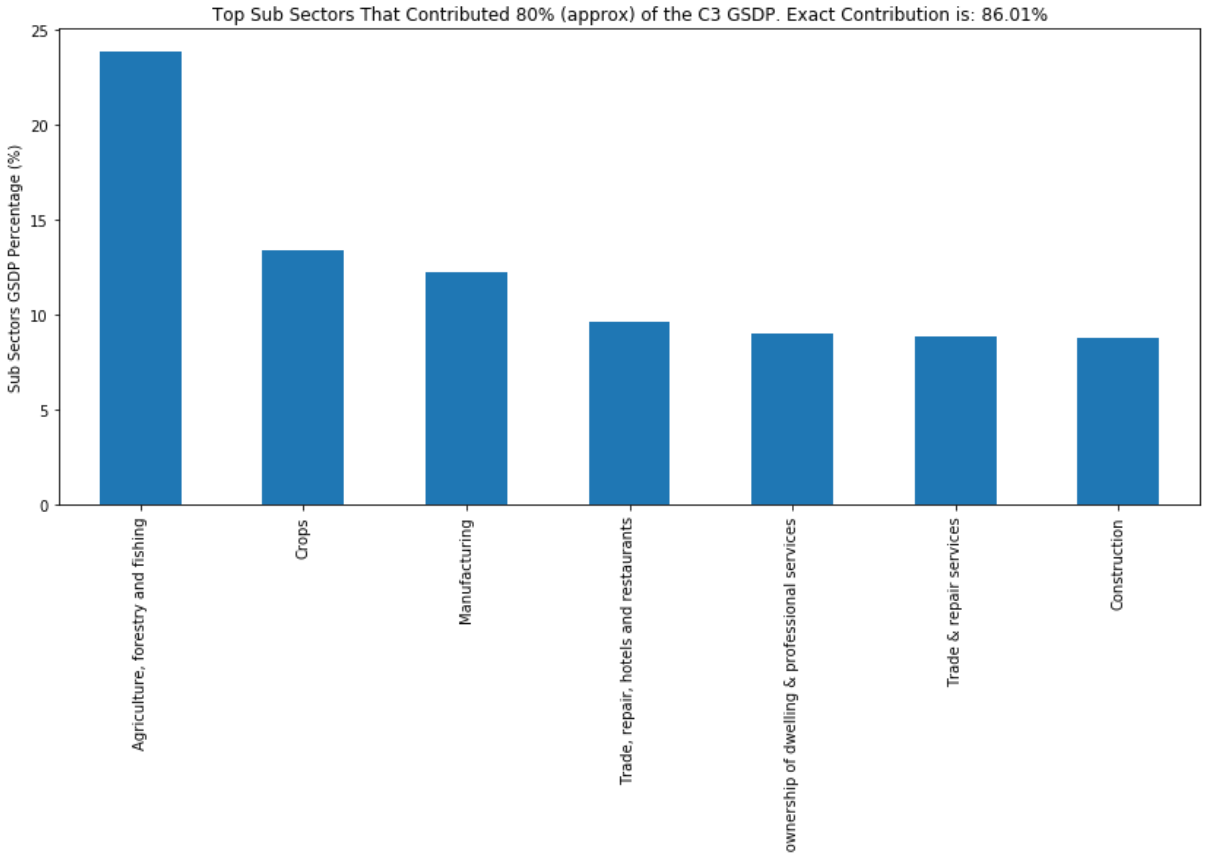
Out[54]:

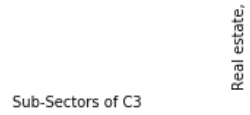
	C3_Sub_Sectors_that_contributes_80%_approximately_to_GSDP_in_Total	%_of_GSDP_Contributior
0	Agriculture, forestry and fishing	23.888002
1	Crops	13.410989

	C3_Sub_Sectors_that_contributes_80%_approximately_to_GSDP_in_Total	%_of_GSDP_Contributior
2	Manufacturing	12.282422
3	Trade, repair, hotels and restaurants	9.633427
4	Real estate, ownership of dwelling & professio...	9.058288
5	Trade & repair services	8.912225
6	Construction	8.826287
7	ABOVE C3 SUB-SECTORS EXACT CONTRIBUTION =	86.010000

In [55]:
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C3:-20-50 percent
# Similarly we write for C3:

import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)

plt.figure(figsize=(14,6))
C3_Sub_Sectors_contributes_80_percent_apprx.set_index("C3_Sub_Sectors_t
hat_contributes_80%_approximately_to_GSDP_in_Total").iloc[:-1,:]['%_of_
GSDP_Contribution'].plot(kind='bar')

plt.ylabel('Sub Sectors GSDP Percentage (%)'); plt.xlabel('Sub-Sectors
 of C3')

plt.title('Top Sub Sectors That Contributed 80% (approx) of the C3 GSD
P. Exact Contribution is: {0}%'.format(C3_Sub_Sectors_contributes_80_pe
rcent_apprx.iloc[-1:,-1:].values[0][0]))
```

```
plt.show()
#inference : we can see that  agriculture and forestry  crops ,manfactu
ring , Hotels & Restaurants , Real estate and professional services con
tribute to nearly 80% of the value
#  also to note that  agriculture & forestry play a   play a major role
 to the contribution even in this sector as well.
# also to note that Real estate along with professional services contri
bute to a considerable extent to the growth of this sector. Ideally , w
e should
# continue to focus on Agricultural and crops , also putting extra effo
rts on education here in such states where the chances of gaining more
 income from professional services like doctors , engineers etc can be
 significantly imporved.
```





In [57]:
```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C4:-0-20 percent
# Similarly we write for C4:

import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)

df_C4 = df_all_states.loc[df_all_states.State.isin(c4.State)&(df_all_st
ates['S.No.']!='Total')&
        (~df_all_states['Item'].isin(['TOTAL GSVA at basic prices','Tax
es on Products','Subsidies on products',"Population ('00)",'Per Capita
 GSDP (Rs.)']))]
df_C4 = df_C4[['Item','2014-15']].groupby(by='Item').sum().sort_values(
by='2014-15',ascending=False).reset_index()
df_C4['%_of_GSDP_Contribution'] = df_C4['2014-15']/(df_C4['2014-15'][0
])*100
start =1; End = 4
while df_C4.iloc[start:End ,-1].sum() <= 78:
    End = End+1
    C4_Sub_Sectors_contributes_80_percent_apprx = df_C4[['Item','%_of_G
SDP_Contribution']].iloc[start:End].append({'Item':'ABOVE C4 SUB-SECTOR
S EXACT CONTRIBUTION =','%_of_GSDP_Contribution':round(df_C4.iloc[start
:End ,-1].sum(),2)},ignore_index=True).rename(columns={'Item':'C4_Sub_S
ectors_that_contributes_80%_approximately_to_GSDP_in_Total'})

C4_Sub_Sectors_contributes_80_percent_apprx
```

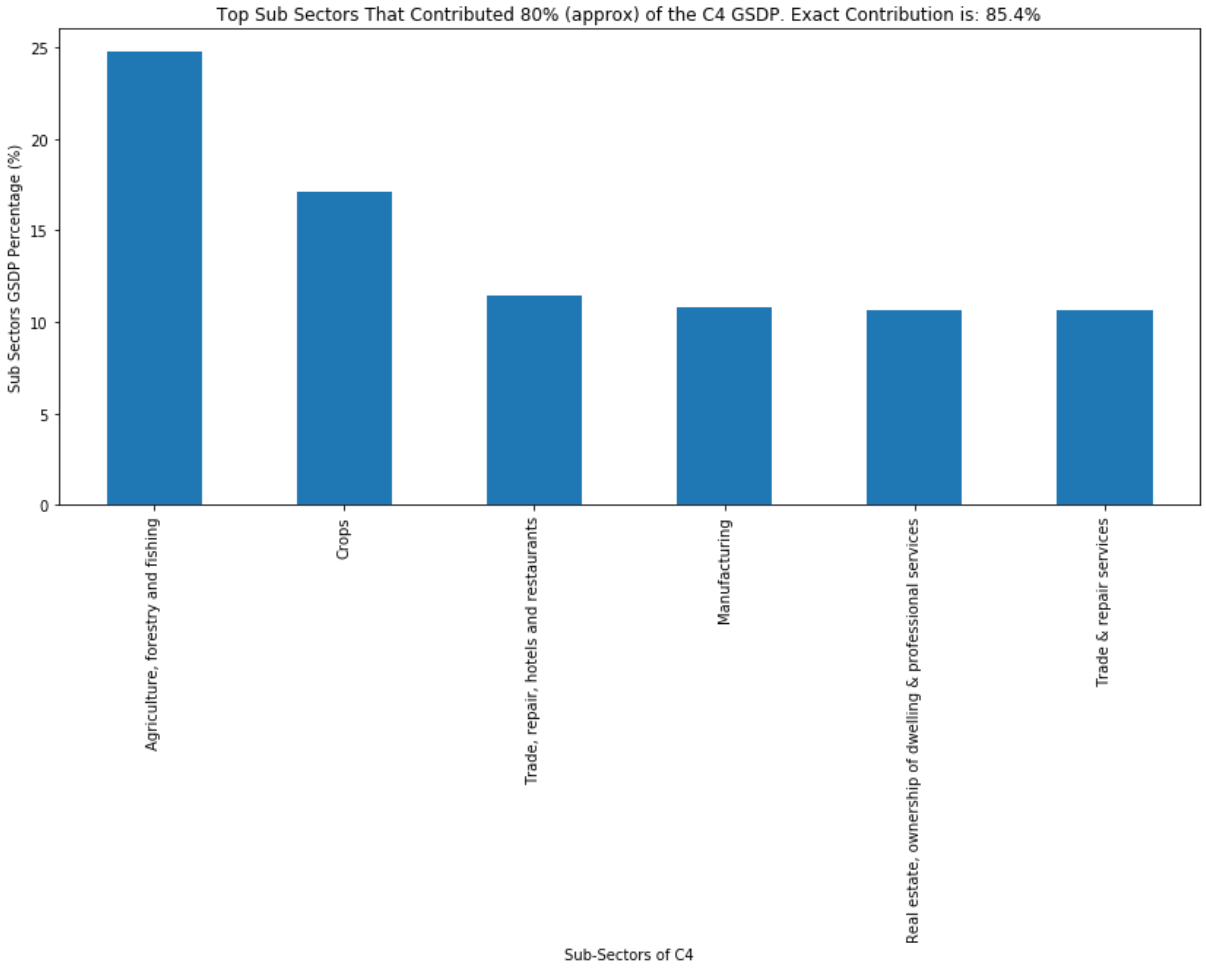
Out[57]:

	C4_Sub_Sectors_that_contributes_80%_approximately_to_GSDP_in_Total	%_of_GSDP_Contribution
0	Agriculture, forestry and fishing	24.774613
1	Crops	17.072772
2	Trade, repair, hotels and restaurants	11.477314
3	Manufacturing	10.768296
4	Real estate, ownership of dwelling & professio...	10.665679
5	Trade & repair services	10.637531
6	ABOVE C4 SUB-SECTORS EXACT CONTRIBUTION =	85.400000

In [58]:

```
#Part  1-B : Q5) Identifying the states that based on the four percenti
le categories -> C4:-0-20 percent
# Similarly we write for C4:

import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns
states_per_capita_sorted = df_all_states.loc[df_all_states.Item=='Per C
apita GSDP (Rs.)'].sort_values(by='2014-15')[['2014-15','State']].renam
e(columns = {'2014-15':'per_capita_GSDP'})
#states_per_capita_sorted.head(30)

plt.figure(figsize=(14,6))

# Selecting the required rows and columns using: iloc[:-1,:]['%_of_GSDP
_Contribution'] and plotting the graph using: plot(kind='bar')

C4_Sub_Sectors_contributes_80_percent_apprx.set_index("C4_Sub_Sectors_t
hat_contributes_80%_approximately_to_GSDP_in_Total").iloc[:-1,:]['%_of_
```

```
GSDP_Contribution'].plot(kind='bar')

plt.ylabel('Sub Sectors GSDP Percentage (%)'); plt.xlabel('Sub-Sectors
 of C4')

plt.title('Top Sub Sectors That Contributed 80% (approx) of the C4 GSD
P. Exact Contribution is: {0}%'.format(C4_Sub_Sectors_contributes_80_pe
rcent_apprx.iloc[-1:,-1:].values[0][0]))

plt.show()
#inference : we can see that  agriculture and forestry  crops ,manfactu
ring , Hotels & Restaurants , Real estate and professional services con
tribute to nearly 80% of the value
#  also to note that  agriculture & forestry play a   play a major role
 to the contribution even in this sector as well.
# also to note that Real estate along with professional services contri
bute to a considerable extent to the growth of this sector. Ideally , w
e should
# continue to focus on Agricultural and crops , also putting extra effo
rts on education here in such states where the chances of gaining more
 income from professional services like doctors , engineers etc can be
 significantly imporved.
```



In [ ]:

In [62]:
```
# Final conclusion :
```

```
#1) we can see that the sectors Agriculture , forestry & fishing , Manu
facturing , real estate , trade and repair are mostly correlaated with
 each other
#2) Also we can see that though transportation , being one of the bigge
st sector is contributing less to the GDP . We need to undertand if the
# expenses in this area is more when compared to the revenue generated
 .
#3) we need to focus on these primary sector areas where there are chan
ces of getting more income , but still contributing less to GDP , by un
derstanding
# the expenses to income gained etc.we need to find ways to cut the cos
ts thereby avoiding getting losses
#4)on the other hand, Teritary sector that contributes more to the GDP
 can be imporved alot especially in some states where the average % of
 GSDP is below the avergae .
# hence CM should focus on improving the primary areas where there is f
ocus for imporvement by understanding the losses that occur on par with
 the income gained and also focus on such areas like
# manfacturing , professional services where we can generate more emplo
yment opportunities in future to increase the GDP/Percapita
```

In [69]:

```
#PART 1 C: Analyse if there is any correlation of GDP per capita with d
ropout rates in education (primary, upper primary and secondary) for th
e year 2014-2015 for the states. Choose an appropriate plot to conduct
 this analysis
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns

df_drop_out = pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\State-wise
 Average Annual Drop-Out Rate from 2012-13 to 2014-15.csv')
df_drop_out.head()

df_drop_out = df_drop_out.rename(columns = {'Primary - 2014-2015':'Prim
ary - 2013-2014','Primary - 2014-2015.1':'Primary - 2014-2015','Level o
```

```
f Education - State':'State'})
df_drop_out.head()
```

Out[69]:

	Sl. No.	State	Primary - 2012-2013	Primary - 2013-2014	Primary - 2014-2015	Upper Primary - 2012-2013	Upper Primary - 2013-2014	Upper Primary - 2014-2015	Secondary - 2012-2013	Secondary - 2013-2014
0	1	A & N Islands	0.68	1.21	0.51	1.23	0.51	1.69	5.56	7.20
1	2	Andhra Pradesh	3.18	4.35	6.72	3.36	3.78	5.20	12.72	12.65
2	3	Arunachal Pradesh	15.16	10.89	10.82	7.47	5.59	6.71	12.93	14.49
3	4	Assam	6.24	7.44	15.36	7.20	7.05	10.51	26.77	30.43
4	5	Bihar	NaN	2.09	NaN	NaN	2.98	4.08	30.14	25.33

In [74]:

```
#workings for 2014-15 :
df_drop_out = df_drop_out[['State','Primary - 2014-2015','Upper Primary
 - 2014-2015','Secondary - 2014-2015']]



df_drop_out = df_drop_out.dropna(how='any') # working on the dropping o
f null values in dataframe
df_drop_out = df_drop_out.replace(['Chhatisgarh','Uttrakhand'],['Chhatt
isgarh','Uttarakhand']) # removing UTs



df_drop_out.head(40)
```

Out[74]:

	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015
0	A & N Islands	0.51	1.69	9.87

	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015
1	Andhra Pradesh	6.72	5.20	15.71
2	Arunachal Pradesh	10.82	6.71	17.11
3	Assam	15.36	10.51	27.06
6	Chhattisgarh	2.91	5.85	21.26
7	Dadra & Nagar Haveli	1.47	4.02	16.77
8	Daman & Diu	1.11	3.11	32.27
10	Goa	0.73	0.07	11.15
11	Gujarat	0.89	6.41	25.04
12	Haryana	5.61	5.81	15.89
13	Himachal Pradesh	0.64	0.87	6.07
14	Jammu and Kashmir	6.79	5.44	17.28
15	Jharkhand	5.48	8.99	24.00
16	Karnataka	2.02	3.85	26.18
19	Madhya Pradesh	6.59	9.20	24.77
20	Maharashtra	1.26	1.79	12.87
21	Manipur	9.66	4.20	14.38
22	Meghalaya	9.46	6.52	20.52
23	Mizoram	10.10	4.78	21.88
24	Nagaland	5.61	7.92	18.23
25	Odisha	2.86	3.81	29.56
26	Puducherry	0.37	0.56	12.19
27	Punjab	3.05	3.22	8.86
28	Rajasthan	5.02	3.07	13.48
29	Sikkim	2.27	1.57	15.89
31	Telangana	2.08	2.30	15.53

	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015
32	Tripura	1.28	1.99	28.42
33	Uttar Pradesh	8.58	2.70	10.22
34	Uttarakhand	4.04	1.19	10.40
35	West Bengal	1.47	4.30	17.80
36	All India	4.13	4.03	17.06

In [77]:
```
df_drop_out = df_drop_out.dropna(how='any') # working on the dropping of null values in dataframe
df_drop_out = df_drop_out.replace(['Chhatisgarh','Uttrakhand'],['Chhattisgarh','Uttarakhand']) # removing UTs



df_drop_out.head(40)
# now mwerging two dataframes to get the percapita and drop out rate together :
df_dropout_percap = pd.merge(df_all_states[df_all_states.Item=='Per Capita GSDP (Rs.)'], df_drop_out, how = 'inner', on = 'State')

df_dropout_percap.head(40)
```

Out[77]:

	S.No.	Item	2014-15	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015
0	17	Per Capita GSDP (Rs.)	104977.0	Andhra Pradesh	6.72	5.20	15.71
1	17	Per Capita GSDP (Rs.)	112718.0	Arunachal Pradesh	10.82	6.71	17.11
2	17	Per Capita GSDP (Rs.)	112718.0	Assam	15.36	10.51	27.06
3	17	Per Capita GSDP (Rs.)	86860.0	Chhattisgarh	2.91	5.85	21.26
4	17	Per Capita GSDP (Rs.)	271793.0	Goa	0.73	0.07	11.15

	S.No.	Item	2014-15	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015
5	17	Per Capita GSDP (Rs.)	141263.0	Gujarat	0.89	6.41	25.04
6	17	Per Capita GSDP (Rs.)	164077.0	Haryana	5.61	5.81	15.89
7	17	Per Capita GSDP (Rs.)	147330.0	Himachal Pradesh	0.64	0.87	6.07
8	17	Per Capita GSDP (Rs.)	62091.0	Jharkhand	5.48	8.99	24.00
9	17	Per Capita GSDP (Rs.)	145141.0	Karnataka	2.02	3.85	26.18
10	17	Per Capita GSDP (Rs.)	62989.0	Madhya Pradesh	6.59	9.20	24.77
11	17	Per Capita GSDP (Rs.)	58442.0	Manipur	9.66	4.20	14.38
12	17	Per Capita GSDP (Rs.)	76228.0	Meghalaya	9.46	6.52	20.52
13	17	Per Capita GSDP (Rs.)	97687.0	Mizoram	10.10	4.78	21.88
14	17	Per Capita GSDP (Rs.)	89607.0	Nagaland	5.61	7.92	18.23
15	17	Per Capita GSDP (Rs.)	73979.0	Odisha	2.86	3.81	29.56
16	17	Per Capita GSDP (Rs.)	126606.0	Punjab	3.05	3.22	8.86
17	17	Per Capita GSDP (Rs.)	84837.0	Rajasthan	5.02	3.07	13.48
18	17	Per Capita GSDP (Rs.)	240274.0	Sikkim	2.27	1.57	15.89
19	17	Per Capita GSDP (Rs.)	139035.0	Telangana	2.08	2.30	15.53

	S.No.	Item	2014-15	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015
20	17	Per Capita GSDP (Rs.)	77358.0	Tripura	1.28	1.99	28.42
21	17	Per Capita GSDP (Rs.)	153076.0	Uttarakhand	4.04	1.19	10.40

In [78]:
```
df_dropout_percap['Total_dropout_in_2014-15'] = df_dropout_percap.iloc
[:,-3:].sum(axis = 1) # data for 2014-15

df_dropout_percap.head(10)
```

Out[78]:

	S.No.	Item	2014-15	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Total_dropout_in_2014-15
0	17	Per Capita GSDP (Rs.)	104977.0	Andhra Pradesh	6.72	5.20	15.71	27.63
1	17	Per Capita GSDP (Rs.)	112718.0	Arunachal Pradesh	10.82	6.71	17.11	34.64
2	17	Per Capita GSDP (Rs.)	112718.0	Assam	15.36	10.51	27.06	52.93
3	17	Per Capita GSDP (Rs.)	86860.0	Chhattisgarh	2.91	5.85	21.26	30.02
4	17	Per Capita GSDP (Rs.)	271793.0	Goa	0.73	0.07	11.15	11.95

	S.No.	Item	2014-15	State	Primary - 2014-2015	Upper Primary - 2014-2015	Secondary - 2014-2015	Total_dropout_in_2014-15
5	17	Per Capita GSDP (Rs.)	141263.0	Gujarat	0.89	6.41	25.04	32.34
6	17	Per Capita GSDP (Rs.)	164077.0	Haryana	5.61	5.81	15.89	27.31
7	17	Per Capita GSDP (Rs.)	147330.0	Himachal Pradesh	0.64	0.87	6.07	7.58
8	17	Per Capita GSDP (Rs.)	62091.0	Jharkhand	5.48	8.99	24.00	38.47
9	17	Per Capita GSDP (Rs.)	145141.0	Karnataka	2.02	3.85	26.18	32.05

In [2]:
```
import numpy as np
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
import seaborn as sns

path = (r'C:\Users\amuly\Desktop\GDP JSON\New All states')

all_files = glob.glob(path + "/*.csv")
req_columns = ['S.No.','Item','2014-15']
union_terr=['Delhi','Jammu & Kashmir','Andaman & Nicobar Islands','Chan
digarh','Puducherry','West Bengal1']
df_all_states = pd.concat([pd.read_csv(i, encoding = 'ISO8859', usecols
```

```
=req_columns).assign(State = i.split('-')[1].replace('_',' '))
                for i in all_files if i.split('-')[2].replace('_',' ')
not in union_terr])
df_drop_out = pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\State-wise
 Average Annual Drop-Out Rate from 2012-13 to 2014-15.csv')

df_dropout_percap = pd.merge(df_all_states[df_all_states.Item=='Per Cap
ita GSDP (Rs.)'], df_drop_out,how='inner',on ='State')
x = df_dropout_percap['2014-15'].values # per-capita GSDP
y1 = df_dropout_percap['Primary - 2014-2015'].values
y2 = df_dropout_percap['Upper Primary - 2014-2015'].values
y3 = df_dropout_percap['Secondary - 2014-2015'].values
y4 = df_dropout_percap['Total_dropout_in_2014-15'].values

#df_dropout_percap = pd.merge(df_all_states[df_all_states.Item=='Per Ca
pita GSDP (Rs.)'], df_drop_out,how='inner',on ='State')
plt.figure(figsize=(14,12))

plt.subplot(221)

plt.title('GSDP vs Dropout Rate in Primary During 2014-2015')

plt.xlabel('GSDP in Crores (Rs.)')

plt.ylabel('Dropout rate in Percentage')

plt.scatter(x,y1)

plt.subplot(222)

plt.title('GSDP vs Dropout Rate in Upper-Primary During 2014-2015')

plt.xlabel('GSDP in Crores (Rs.)')

plt.ylabel('Dropout rate in Percentage')

plt.scatter(x,y2)

plt.subplot(223)
```

```
plt.title('GSDP vs Dropout Rate in Secondary During 2014-2015')

plt.xlabel('GSDP in Crores (Rs.)')

plt.ylabel('Dropout rate in Percentage')

plt.scatter(x,y3)

plt.subplot(224)

plt.title('GSDP vs Total Dropout Rate During 2014-2015')

plt.xlabel('GSDP in Crores (Rs.)')

plt.ylabel('Dropout rate in Percentage')

plt.scatter(x,y4)

plt.show()

# inference :   there is slight increase in GSDP with decrease in drop
 out , means that as the GSDP increases the drop out decreses and
# vice versa . so as the educational level increses , there will be dem
aand and increase in income for manufacturing , professional , type of
 services and units
# thereby creating more job opportunities , increasing income and impor
ving the standard of living of people.
```

```
---------------------------------------------------------------------
----
KeyError                                  Traceback (most recent call l
ast)
<ipython-input-2-1737aa11f0e8> in <module>
     15 df_drop_out = pd.read_csv(r'C:\Users\amuly\Desktop\GDP JSON\Sta
te-wise Average Annual Drop-Out Rate from 2012-13 to 2014-15.csv')
     16
---> 17 df_dropout_percap = pd.merge(df_all_states[df_all_states.Item==
'Per Capita GSDP (Rs.)'], df_drop_out,how='inner',on ='State')
     18 x = df_dropout_percap['2014-15'].values # per-capita GSDP
```

```
     19 y1 = df_dropout_percap['Primary - 2014-2015'].values


~\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py in merge(lef
t, right, how, on, left_on, right_on, left_index, right_index, sort, su
ffixes, copy, indicator, validate)
     79         copy=copy,
     80         indicator=indicator,
---> 81         validate=validate,
     82     )
     83     return op.get_result()

~\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py in __init__
(self, left, right, how, on, left_on, right_on, axis, left_index, right
_index, sort, suffixes, copy, indicator, validate)
    624             self.right_join_keys,
    625             self.join_names,
--> 626         ) = self._get_merge_keys()
    627
    628         # validate the merge keys dtypes. We may need to coerce

~\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py in _get_merg
e_keys(self)
    973                     if not is_rkey(rk):
    974                         if rk is not None:
--> 975                             right_keys.append(right._get_label_
or_level_values(rk))
    976                         else:
    977                             # work-around for merge_asof(right_
index=True)

~\Anaconda3\lib\site-packages\pandas\core\generic.py in _get_label_or_l
evel_values(self, key, axis)
   1772             values = self.axes[axis].get_level_values(key)._val
ues
   1773         else:
-> 1774             raise KeyError(key)
   1775
   1776         # Check for duplicates

KeyError: 'State'
```

KeyError: 'state'

In [ ]: