In [22]:

```
import pandas as pd
import math
import matplotlib.pyplot as plt
a=pd.read_csv("Poverty.csv",index_col=1)
a
```

Out[22]:

State	Sr.no	Number Below Poverty Line in % (2004-05)	Number Below Poverty Line in % (2009-10)	Number Below Poverty Line in % (2011-12)	Number Below Poverty Line in Lakhs (2004-05)	Number Below Poverty Line in Lakhs (2009-10)	Number Below Poverty Line in Lakhs (2011-12)
Andhra Pradesh	1	29.9	21.1	9.2	238.8	176.6	78.8
Arunachal Pradesh	2	31.1	25.9	34.7	3.6	4.5	4.9
Assam	3	34.4	37.9	32.0	97.3	116.4	101.3
Bihar	4	54.4	53.5	33.7	485.6	543.5	358.2
Chhattisgarh	5	49.4	48.7	39.9	109.9	121.9	104.1
Goa	6	25.0	14.2	9.9	3.6	23.3	17
Gujarat	7	31.8	8.7	5.1	172.2	1.3	0.8
Haryana	8	24.1	23.0	16.6	55.1	136.2	102.2
Himachal Pradesh	9	22.9	20.1	11.2	14.6	50	28.8
Jammu & Kashmir	10	13.2	9.5	8.1	14.2	6.4	5.6
Jharkhand	11	45.3	9.4	10.3	130.7	11.5	13.3
Karnataka	12	33.4	39.1	37.0	185.7	126.2	124.3
Kerala	13	19.7	23.6	20.9	65	142.3	129.8
Madhya Pradesh	14	48.6	12.0	7.1	316.9	39.6	23.9
Maharashtra	15	38.1	36.7	31.6	393.3	261.8	234.1
Manipur	16	38.0	24.5	17.4	8.7	270.8	197.9
Meghalaya	17	16.1	47.1	36.9	3.9	12.5	10.2
Mizoram	18	15.3	17.1	11.9	1.4	4.9	3.6
Nagaland	19	9.0	21.1	20.4	1.9	2.3	2.3
Orissa	20	57.2	20.9	18.9	220.2	4.1	3.8
Punjab	21	20.9	37.0	32.6	53.8	153.2	138.5
Rajasthan	22	34.4	1.2	8.3	210.3	0.1	23.2
Sikkim	23	31.1	15.9	14.7	1.8	43.5	102.9
Tamil Nadu	24	28.9	24.8	8.2	186.8	167	0.5
Tripura	25	40.6	13.1	11.3	13.7	0.8	82.6

State	Sr.no	Number Below Poverty Line in % (2004-05)	Number Below Poverty Line in % (2009-10)	Number Below Poverty Line in % (2011-12)	Number Below Poverty Line in Lakhs (2004-05)	Number Below Poverty Line in Lakhs (2009-10)	Number Below Poverty Line in Lakhs (2011-12)
Uttar Pradesh	26	40.9	17.1	14.0	735.5	121.8	5.2
Uttarkhand	27	32.7	17.4	29.4	29.7	6.3	598.2
West Bengal	28	34.3	37.7	11.3	289.1	737.9	11.6
A & N Islands	29	3.0	18.0	20.0	0.1	17.9	185
Chandigarh	30	11.6	26.7	9.7	1.1	240.3	1.2
Daman & Diu	31	8.8	9.2	21.8	0.2	1	2.3
Delhi	32	13.1	39.1	39.3	20.4	1.3	1.4
India	33	37.2	29.8	21.9	4,076.10	3,546.80	2,697.80

In [23]:

```
b=a[0:32]
b
```

Out[23]:

State	Sr.no	Number Below Poverty Line in % (2004-05)	Number Below Poverty Line in % (2009-10)	Number Below Poverty Line in % (2011-12)	Number Below Poverty Line in Lakhs (2004-05)	Number Below Poverty Line in Lakhs (2009-10)	Number Below Poverty Line in Lakhs (2011-12)
Andhra Pradesh	1	29.9	21.1	9.2	238.8	176.6	78.8
Arunachal Pradesh	2	31.1	25.9	34.7	3.6	4.5	4.9
Assam	3	34.4	37.9	32.0	97.3	116.4	101.3
Bihar	4	54.4	53.5	33.7	485.6	543.5	358.2
Chhattisgarh	5	49.4	48.7	39.9	109.9	121.9	104.1
Goa	6	25.0	14.2	9.9	3.6	23.3	17
Gujarat	7	31.8	8.7	5.1	172.2	1.3	0.8
Haryana	8	24.1	23.0	16.6	55.1	136.2	102.2
Himachal Pradesh	9	22.9	20.1	11.2	14.6	50	28.8
Jammu & Kashmir	10	13.2	9.5	8.1	14.2	6.4	5.6
Jharkhand	11	45.3	9.4	10.3	130.7	11.5	13.3
Karnataka	12	33.4	39.1	37.0	185.7	126.2	124.3
Kerala	13	19.7	23.6	20.9	65	142.3	129.8
Madhya Pradesh	14	48.6	12.0	7.1	316.9	39.6	23.9
Maharashtra	15	38.1	36.7	31.6	393.3	261.8	234.1
Manipur	16	38.0	24.5	17.4	8.7	270.8	197.9
Meghalaya	17	16.1	47.1	36.9	3.9	12.5	10.2
Mizoram	18	15.3	17.1	11.9	1.4	4.9	3.6
Nagaland	19	9.0	21.1	20.4	1.9	2.3	2.3
Orissa	20	57.2	20.9	18.9	220.2	4.1	3.8
Punjab	21	20.9	37.0	32.6	53.8	153.2	138.5
Rajasthan	22	34.4	1.2	8.3	210.3	0.1	23.2
Sikkim	23	31.1	15.9	14.7	1.8	43.5	102.9
Tamil Nadu	24	28.9	24.8	8.2	186.8	167	0.5
Tripura	25	40.6	13.1	11.3	13.7	0.8	82.6
Uttar Pradesh	26	40.9	17.1	14.0	735.5	121.8	5.2

State	Sr.no	Number Below Poverty Line in % (2004-05)	Number Below Poverty Line in % (2009-10)	Number Below Poverty Line in % (2011-12)	Number Below Poverty Line in Lakhs (2004-05)	Number Below Poverty Line in Lakhs (2009-10)	Number Below Poverty Line in Lakhs (2011-12)
Uttarkhand	27	32.7	17.4	29.4	29.7	6.3	598.2
West Bengal	28	34.3	37.7	11.3	289.1	737.9	11.6
A & N Islands	29	3.0	18.0	20.0	0.1	17.9	185
Chandigarh	30	11.6	26.7	9.7	1.1	240.3	1.2
Daman & Diu	31	8.8	9.2	21.8	0.2	1	2.3
Delhi	32	13.1	39.1	39.3	20.4	1.3	1.4

In [24]:

```
c=round(b["Number Below Poverty Line in % (2004-05)"].mean(),2)
print("mean of Number Below Poverty Line in % (2004-05) is : ", c)     #c=mean of Number Be
d=round(b["Number Below Poverty Line in % (2009-10)"].mean(),2)
print("mean of Number Below Poverty Line in % (2009-10) is : ", d)     #d=mean of Number Be
```

```
mean of Number Below Poverty Line in % (2004-05) is :  29.29
mean of Number Below Poverty Line in % (2009-10) is :  24.1
```

In [25]:

```
b['Number Below Poverty Line in Lakhs (2004-05)'] = b['Number Below Poverty Line in Lakhs (
b['Number Below Poverty Line in Lakhs (2009-10)'] = b['Number Below Poverty Line in Lakhs (
```

```
<ipython-input-25-9ec78652db7c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  b['Number Below Poverty Line in Lakhs (2004-05)'] = b['Number Below Povert
y Line in Lakhs (2004-05)'].astype(float)
<ipython-input-25-9ec78652db7c>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  b['Number Below Poverty Line in Lakhs (2009-10)'] = b['Number Below Povert
y Line in Lakhs (2009-10)'].astype(float)
```

In [26]:

```
e=round(b["Number Below Poverty Line in Lakhs (2004-05)"].mean(),2)
print("mean of Number Below Poverty Line in Lakhs (2004-05) is : ", e)          #c=mean of
                                                                                 #Lakhs (200
f=round(b["Number Below Poverty Line in Lakhs (2009-10)"].mean(),2)
print("mean of Number Below Poverty Line in Lakhs (2009-10) is : ", f) #d=mean of Number Be
                                                                       #Lakhs (2009-10)
```

```
mean of Number Below Poverty Line in Lakhs (2004-05) is :  127.03
mean of Number Below Poverty Line in Lakhs (2009-10) is :  110.85
```

In [27]:

```
g=round(b["Number Below Poverty Line in Lakhs (2004-05)"].std(),2)
print("s.d. of Number Below Poverty Line in Lakhs (2004-05) is : ", g) #c=standard deviatio
                                                                        #Lakhs (2004-05)
h=round(b["Number Below Poverty Line in Lakhs (2009-10)"].std(),2)
print("mean of Number Below Poverty Line in Lakhs (2009-10) is : ", h) #d=standard deviatio
                                                                       #Lakhs (2009-10)
```

```
s.d. of Number Below Poverty Line in Lakhs (2004-05) is :  170.26
mean of Number Below Poverty Line in Lakhs (2009-10) is :  164.03
```

In [28]:

```
k=pd.read_csv("t test table.csv")
k
```

Out[28]:

	v	0.1	0.05	0.025	0.01	0.00833	0.00625	0.005
0	1	3.078	6.314	12.706	31.821	38.204	50.923	63.657
1	2	1.886	2.920	4.303	6.965	7.650	8.860	9.925
2	3	1.638	2.353	3.182	4.541	4.857	5.392	5.841
3	4	1.533	2.132	2.776	3.747	3.961	4.315	4.604
4	5	1.476	2.015	2.571	3.365	3.534	3.810	4.032
5	6	1.440	1.943	2.447	3.143	3.288	3.521	3.707
6	7	1.415	1.895	2.365	2.998	3.128	3.335	3.499
7	8	1.397	1.860	2.306	2.896	3.016	3.206	3.355
8	9	1.383	1.833	2.262	2.821	2.934	3.111	3.250
9	10	1.372	1.812	2.228	2.764	2.870	3.038	3.169
10	11	1.363	1.796	2.201	2.718	2.820	2.891	3.106
11	12	1.356	1.782	2.179	2.681	2.780	2.934	3.055
12	13	1.350	1.771	2.160	2.650	2.746	2.896	3.012
13	14	1.345	1.761	2.145	2.624	2.718	2.864	2.977
14	15	1.341	1.753	2.131	2.602	2.694	2.837	2.947
15	16	1.337	1.746	2.120	2.583	2.673	2.813	2.921
16	17	1.333	1.740	2.110	2.567	2.655	2.793	2.898
17	18	1.330	1.734	2.101	2.552	2.639	2.775	2.878
18	19	1.328	1.729	2.093	2.539	2.625	2.759	2.861
19	20	1.325	1.725	2.086	2.528	2.613	2.744	2.845
20	21	1.323	1.721	2.080	2.518	2.602	2.732	2.831
21	22	1.321	1.717	2.074	2.508	2.591	2.720	2.819
22	23	1.319	1.714	2.069	2.500	2.582	2.710	2.807
23	24	1.318	1.711	2.064	2.492	2.574	2.700	2.797
24	25	1.316	1.708	2.060	2.485	2.566	2.692	2.787
25	26	1.315	1.706	2.056	2.479	2.559	2.684	2.779
26	27	1.314	1.703	2.052	2.473	2.553	2.676	2.771
27	28	1.313	1.701	2.048	2.467	2.547	2.669	2.763
28	29	1.311	1.699	2.045	2.462	2.541	2.663	2.756
29	30	1.282	1.645	1.960	2.326	2.394	2.498	2.576

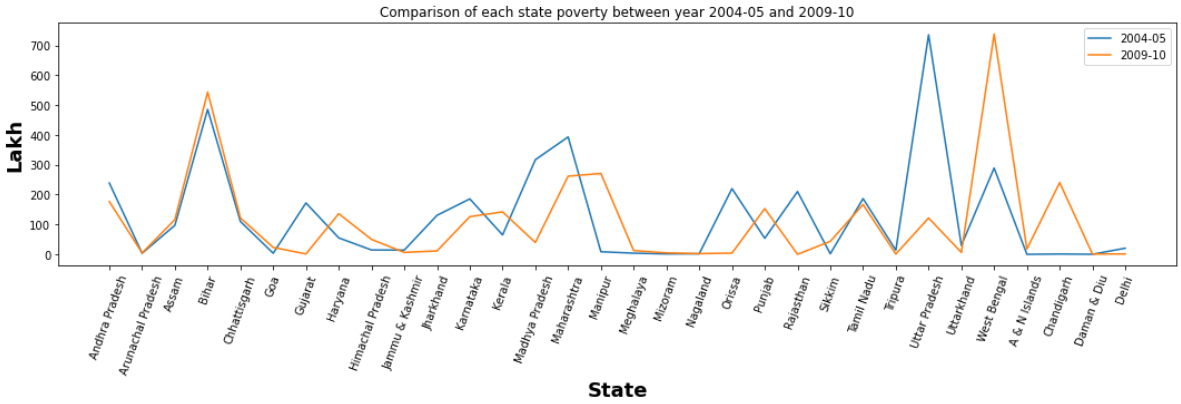
In [29]:

```
#Hypothesis concerning difference of means (Two Populations)

delta=0
n1=33            #sample 1 size
n2=33
z=(e-f-delta)/(math.sqrt((g*g)/n1+(h*h)/n2))     #g=sample1 standard deviation  # h=sample2
print("z :",z)



dof=int(input('Enter the degree of freedom:'))
lof=float(input('Enter the level of significance:'))                    #lof=level of signifi
lof=str(lof/2)     # for two tailed test


l=k['v']
counter=0
for i in range(0,dof):
    if dof>=30:
        for i in k['v']:
            if dof>=i:
                y=k.at[29,lof]
        print('critical value of z is :', y)
        break

    else:
        for i in k['v']:
            if dof==i:
                x=k.at[dof-1,lof]
        print('critical value of z is :', x)
        break

if x or y>abs(z):
    print("Null hypothesis is accepted")
else:
    print("Null hypothesis is rejectted")
```

```
z : 0.39314403745352305
Enter the degree of freedom:32
Enter the level of significance:0.05
critical value of z is : 1.96
Null hypothesis is accepted
```

In [34]:

```
plt.figure(figsize=(18,4))
p1=plt.plot(b['Number Below Poverty Line in Lakhs (2004-05)'])
p2=plt.plot(b['Number Below Poverty Line in Lakhs (2009-10)'])
plt.legend((p1[0],p2[0]),('2004-05','2009-10'))
plt.xlabel('State',fontweight='bold',fontsize=18)
plt.ylabel('Lakh',fontweight='bold',fontsize=18)
plt.xticks(rotation=70)
plt.title('Comparison of each state poverty between year 2004-05 and 2009-10')
plt.show()
```



In [32]:

```
j=pd.read_csv("Poverty1.csv")
j
```

Out[32]:

	State	Number Below Poverty Line in Lakhs (2004-05)	Number Below Poverty Line in Lakhs (2009-10)
0	Andhra Pradesh	238.8	176.6
1	Arunachal Pradesh	3.6	4.5
2	Assam	97.3	116.4
3	Bihar	485.6	543.5
4	Chhattisgarh	109.9	121.9
5	Goa	3.6	23.3
6	Gujarat	172.2	1.3
7	Haryana	55.1	136.2
8	Himachal Pradesh	14.6	50.0
9	Jammu & Kashmir	14.2	6.4
10	Jharkhand	130.7	11.5
11	Karnataka	185.7	126.2
12	Kerala	65.0	142.3
13	Madhya Pradesh	316.9	39.6
14	Maharashtra	393.3	261.8
15	Manipur	8.7	270.8
16	Meghalaya	3.9	12.5
17	Mizoram	1.4	4.9
18	Nagaland	1.9	2.3
19	Orissa	220.2	4.1
20	Punjab	53.8	153.2
21	Rajasthan	210.3	0.1
22	Sikkim	1.8	43.5
23	Tamil Nadu	186.8	167.0
24	Tripura	13.7	0.8
25	Uttar Pradesh	735.5	121.8
26	Uttarkhand	29.7	6.3
27	West Bengal	289.1	737.9
28	A & N Islands	0.1	17.9
29	Chandigarh	1.1	240.3

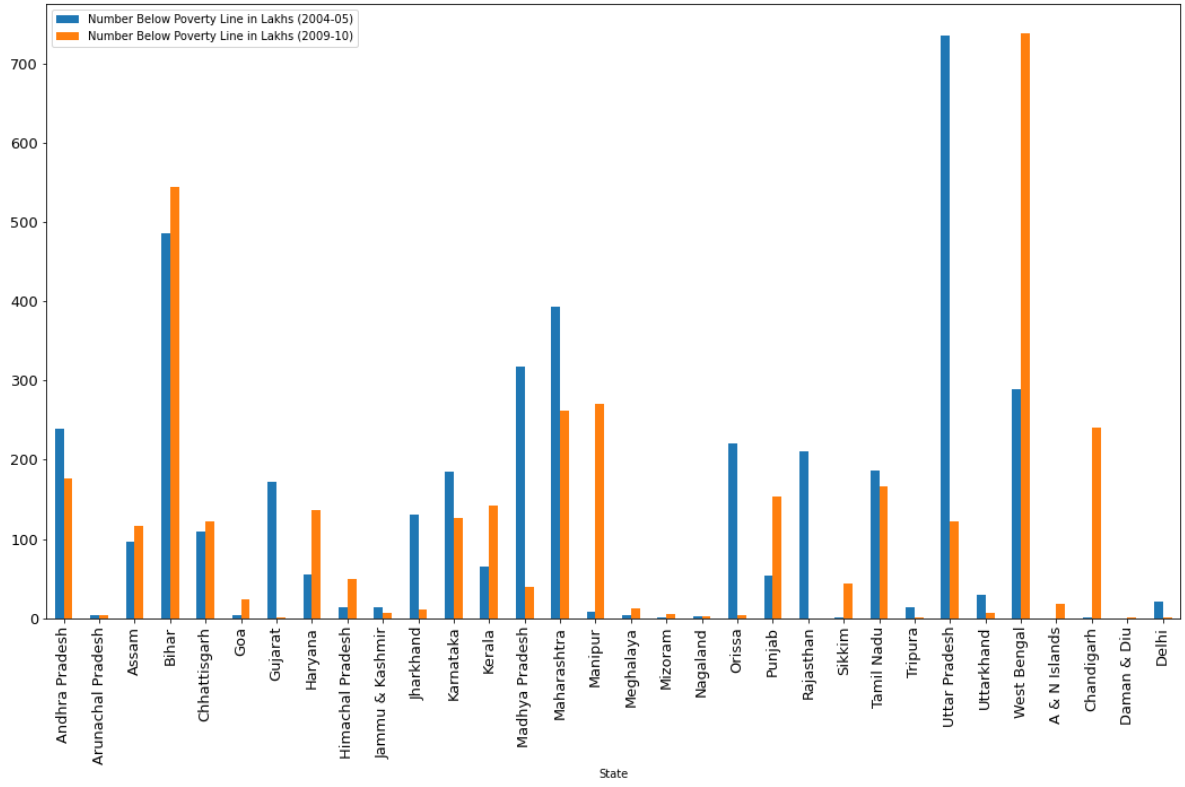
	State	Number Below Poverty Line in Lakhs (2004-05)	Number Below Poverty Line in Lakhs (2009-10)
30	Daman & Diu	0.2	1.0
31	Delhi	20.4	1.3

In [33]:

```
j.plot.bar(x="State",figsize=(18,10),fontsize=13)
```

Out[33]:

```
<AxesSubplot:xlabel='State'>
```



In [ ]:

In [ ]:

In [3]:

```
import pandas as pd
a=pd.read_csv("Corrilation of observation 1.csv")
a
```

Out[3]:

	State	Number Below Poverty Line in Lakhs (2009-10)	GDP per capita income(2009-2010)	x^2	y^2	xy
0	Andhra Pradesh	176.6	52814	31187.56	2789318596	9326952.4
1	Arunachal Pradesh	4.5	48662	20.25	2367990244	218979.0
2	Assam	116.4	27464	13548.96	754271296	3196809.6
3	Bihar	543.5	17064	295392.25	291180096	9274284.0
4	Chhattisgarh	121.9	35121	14859.61	1233484641	4281249.9
5	Goa	23.3	148136	542.89	21944274496	3451568.8
6	Gujarat	1.3	63549	1.69	4038475401	82613.7
7	Haryana	136.2	80759	18550.44	6522016081	10999375.8
8	Himachal Pradesh	50.0	56706	2500.00	3215570436	2835300.0
9	Jammu & Kashmir	6.4	33665	40.96	1133332225	215456.0
10	Jharkhand	11.5	27132	132.25	736145424	312018.0
11	Karnataka	126.2	52191	15926.44	2723900481	6586504.2
12	Kerala	142.3	60264	20249.29	3631749696	8575567.2
13	Madhya Pradesh	39.6	28571	1568.16	816302041	1131411.6
14	Maharashtra	261.8	74027	68539.24	5479996729	19380268.6
15	Manipur	270.8	27332	73332.64	747038224	7401505.6
16	Meghalaya	12.5	45006	156.25	2025540036	562575.0
17	Mizoram	4.9	43467	24.01	1889380089	212988.3
18	Nagaland	2.3	49465	5.29	2446786225	113769.5
19	Orissa	4.1	34361	16.81	1180678321	140880.1
20	Punjab	153.2	61894	23470.24	3830867236	9482160.8
21	Rajasthan	0.1	34042	0.01	1158857764	3404.2
22	Sikkim	43.5	68731	1892.25	4723950361	2989798.5
23	Tamil Nadu	167.0	63547	27889.00	4038221209	10612349.0
24	Tripura	0.8	39949	0.64	1595922601	31959.2
25	Uttar Pradesh	121.8	23392	14835.24	547185664	2849145.6
26	Uttarkhand	6.3	59316	39.69	3518387856	373690.8
27	West Bengal	737.9	41837	544496.41	1750334569	30871522.3

	State	Number Below Poverty Line in Lakhs (2009-10)	GDP per capita income(2009-2010)	x^2	y^2	xy
28	A & N Islands	17.9	75836	320.41	5751098896	1357464.4
29	Chandigarh	240.3	118136	57744.09	13956114496	28388080.8
30	Delhi	1.3	129746	1.69	16834024516	168669.8

In [4]:

```
import math
n=31
b=a["Number Below Poverty Line in Lakhs (2009-10)"].sum()
print("Sum of x column is :",b)
c=a["GDP per capita income(2009-2010)"].sum()
print("Sum of y column is :",c)
d=a["x^2"].sum()
e=a["y^2"].sum()
f=a["xy"].sum()
Sxy=(f-((b*c)/n))
print("Sxy",Sxy)
Syy=(e-((c*c)/n))
print("Syy",Syy)
Sxx=(d-((b*b)/n))
print("Sxx",Sxx)
r=Sxy/(math.sqrt(Sxx*Syy))
print("r:",r)
```

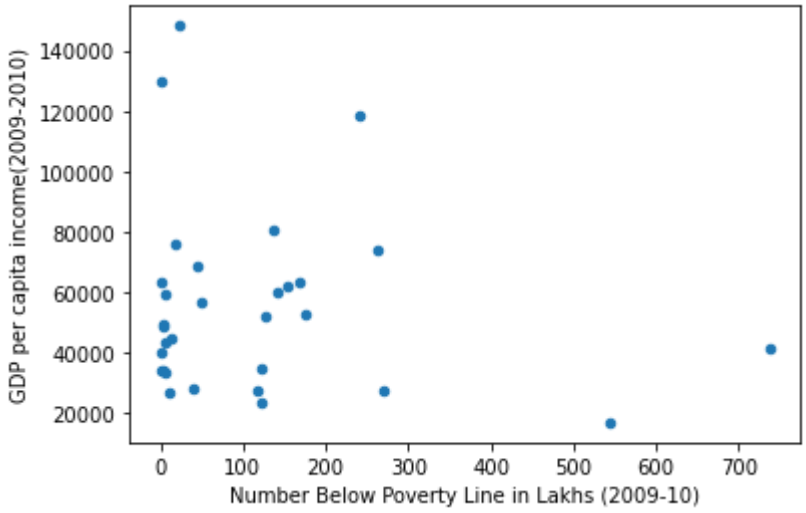
```
Sum of x column is : 3546.200000000001
Sum of y column is : 1722182
Sxy -21578187.248387128
Syy 27997852683.935486
Sxx 821622.258709677
r: -0.14227103706555655
```

In [5]:

```
a.plot.scatter(x='Number Below Poverty Line in Lakhs (2009-10)',y='GDP per capita income(20
```

Out[5]:

```
<AxesSubplot:xlabel='Number Below Poverty Line in Lakhs (2009-10)', ylabel
='GDP per capita income(2009-2010)'>
```



In [6]:

```
#y on x
#y=alpha+B(y/x)x
g=a["GDP per capita income(2009-2010)"].mean()
h=a["Number Below Poverty Line in Lakhs (2009-10)"].mean()
i=Sxy/Sxx
alpha=g-(i*h)
print("y =",alpha,"+",i,"x")
```

```
y = 58558.56494756082 + -26.262904905077374 x
```

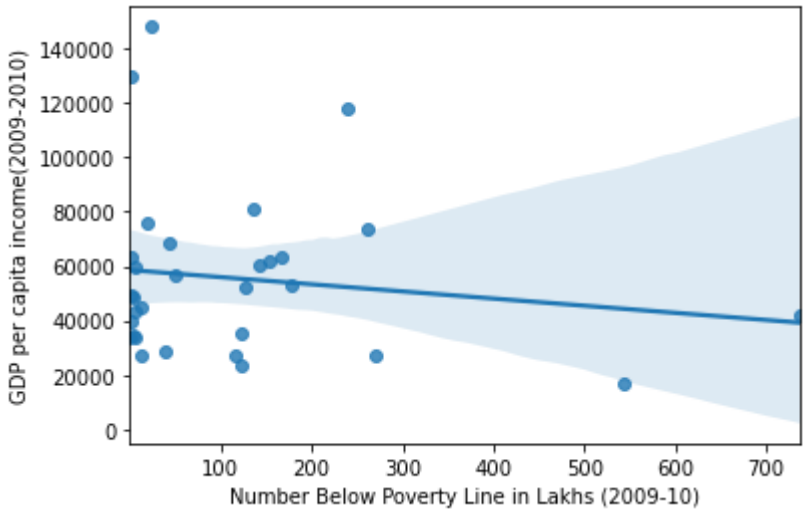
In [7]:

```
import seaborn as sns
x=a['Number Below Poverty Line in Lakhs (2009-10)']
y=a['GDP per capita income(2009-2010)']
sns.regplot(x,y)
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[7]:

```
<AxesSubplot:xlabel='Number Below Poverty Line in Lakhs (2009-10)', ylabel
='GDP per capita income(2009-2010)'>
```



In [ ]:

In [2]:

```
import pandas as pd
import math
a=pd.read_csv("Poverty2.csv",index_col=0)
a
```

Out[2]:

State	% of indebtedness among households
Andhra Pradesh	66.6
Assam	59.3
Bihar	34.6
Gujarat	56.0
Haryana	51.1
Karnataka	35.4
Kerala	64.3
Madhya Pradesh	34.2
Maharashtra	42.0
Odisha	29.3
Punjab	46.9
Rajasthan	38.8
Tamil Nadu	53.0
Uttar Pradesh	42.9
West Bengal	61.5

In [3]:

```
b=pd.read_csv("t test table.csv")
b
```

Out[3]:

	v	0.1	0.05	0.025	0.01	0.00833	0.00625	0.005
0	1	3.078	6.314	12.706	31.821	38.204	50.923	63.657
1	2	1.886	2.920	4.303	6.965	7.650	8.860	9.925
2	3	1.638	2.353	3.182	4.541	4.857	5.392	5.841
3	4	1.533	2.132	2.776	3.747	3.961	4.315	4.604
4	5	1.476	2.015	2.571	3.365	3.534	3.810	4.032
5	6	1.440	1.943	2.447	3.143	3.288	3.521	3.707
6	7	1.415	1.895	2.365	2.998	3.128	3.335	3.499
7	8	1.397	1.860	2.306	2.896	3.016	3.206	3.355
8	9	1.383	1.833	2.262	2.821	2.934	3.111	3.250
9	10	1.372	1.812	2.228	2.764	2.870	3.038	3.169
10	11	1.363	1.796	2.201	2.718	2.820	2.891	3.106
11	12	1.356	1.782	2.179	2.681	2.780	2.934	3.055
12	13	1.350	1.771	2.160	2.650	2.746	2.896	3.012
13	14	1.345	1.761	2.145	2.624	2.718	2.864	2.977
14	15	1.341	1.753	2.131	2.602	2.694	2.837	2.947
15	16	1.337	1.746	2.120	2.583	2.673	2.813	2.921
16	17	1.333	1.740	2.110	2.567	2.655	2.793	2.898
17	18	1.330	1.734	2.101	2.552	2.639	2.775	2.878
18	19	1.328	1.729	2.093	2.539	2.625	2.759	2.861
19	20	1.325	1.725	2.086	2.528	2.613	2.744	2.845
20	21	1.323	1.721	2.080	2.518	2.602	2.732	2.831
21	22	1.321	1.717	2.074	2.508	2.591	2.720	2.819
22	23	1.319	1.714	2.069	2.500	2.582	2.710	2.807
23	24	1.318	1.711	2.064	2.492	2.574	2.700	2.797
24	25	1.316	1.708	2.060	2.485	2.566	2.692	2.787
25	26	1.315	1.706	2.056	2.479	2.559	2.684	2.779
26	27	1.314	1.703	2.052	2.473	2.553	2.676	2.771
27	28	1.313	1.701	2.048	2.467	2.547	2.669	2.763
28	29	1.311	1.699	2.045	2.462	2.541	2.663	2.756
29	30	1.282	1.645	1.960	2.326	2.394	2.498	2.576

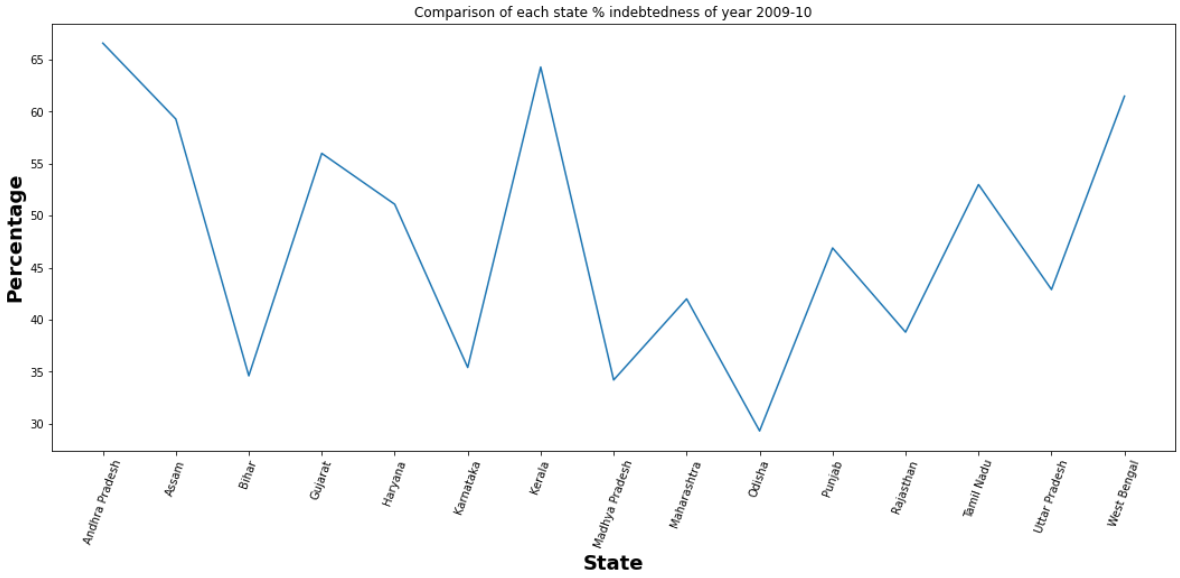
In [4]:

```
#Test of Hypothesis concerning Proportion (Single Population)
#Null Hypothesis H0: p = p0
#Alternative Hypothesis: p > p0 (Right tailed); p < p0 (Left tailed); p ≠ p0 (Two tailed)
#Level of Significance: α
#Sample Size: n
n=15
p=0.477267
p0=0.4
q0=1-p0
z=(p-p0)/(math.sqrt((p0*q0)/n))
print("z :",z)

dof=int(input('Enter the degree of freedom:'))
lof=float(input('Enter the level of significance:'))                #lof=level of signifi
lof=str(lof/2)     # for two tailed test


l=b['v']
counter=0
for i in range(0,dof):
    if dof>=30:
        for i in b['v']:
            if dof>=i:
                y=b.at[29,lof]
        print('critical value of z is :', y)
        break

    else:
        for i in b['v']:
            if dof==i:
                x=b.at[dof-1,lof]
        print('critical value of z is :', x)
        break

if x or y>abs(z):
    print("Null hypothesis is accepted")
else:
    print("Null hypothesis is rejectted")
```

```
z : 0.6108492699205752
Enter the degree of freedom:14
Enter the level of significance:0.05
critical value of z is : 2.145
Null hypothesis is accepted
```

In [5]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,7))
plt.plot(a['% of indebtedness among households'])
plt.xlabel('State',fontweight='bold',fontsize=18)
plt.ylabel('Percentage',fontweight='bold',fontsize=18)
plt.xticks(rotation=70)
plt.title('Comparison of each state % indebtedness of year 2009-10')
plt.show()
```



In [ ]:

```
import numpy as np
import sys
L=[]
n = int(input('Enter order of matrix: '))

a = np.zeros((n,n))

print('Enter Matrix Coefficients:')
for i in range(n):
    for j in range(n):
        a[i][j] = float(input( 'a['+str(i)+']['+ str(j)+']='))
print("The Matrix A is\n",a)
x = np.zeros((n))

print('Enter initial guess vector: ')
for i in range(n):
    x[i] = float(input( 'x['+str(i)+']='))

tolerable_error = float(input('Enter tolerable error: '))

max_iteration = int(input('Enter maximum number of steps: '))

lambda_old = 1.0
condition =  True
step = 1
while condition:
    x = np.matmul(a,x)
    #max Valur from asis i.e. lambda
    lambda_new = max(abs(x))

    x = x/lambda_new
    L.append(lambda_new)
    print('\nSTEP %d' %(step))
    print('----------')
    print('Eigen Value = %0.4f' %(lambda_new))
    print('Eigen Vector: ')
    for i in range(n):
        print('%0.3f\t' % (x[i]))

    step = step + 1
    if step > max_iteration:
        print('Not convergent in given maximum iteration!')
        break

    error = abs(lambda_new - lambda_old)
    print('errror='+ str(error))
    lambda_old = lambda_new
    condition = error > tolerable_error
print('This is the list of eigenvalues',L)
print()
print('The range of the eigenvalues is between',min(L),'to',max(L))
```

In [ ]:

```
sid=pd.read_csv("correlation of observation 1.csv")
sid
```

In [ ]: