Health Status Indicators

In [11]:
```
#importing all the libraries
import os
from os import listdir
from os.path import isfile, join
import struct
import numpy as np
import random
import operator
import matplotlib.pyplot as plt
import gzip
import pandas as pd
import seaborn as sns
```

In [12]:
```
mypath=r'D:\Drive\Coursework\IDS\Project_IDS\Dataset'
os.chdir(mypath)
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]
onlyfiles
```

Out[12]:
```
['CHSI DataSet.xls',
 'CSV File Index.txt',
 'DATAELEMENTDESCRIPTION.csv',
 'DEFINEDDATAVALUE.csv',
 'DEMOGRAPHICS.csv',
 'HEALTHYPEOPLE2010.csv',
 'LEADINGCAUSESOFDEATH.csv',
 'MEASURESOFBIRTHANDDEATH.csv',
 'PREVENTIVESERVICESUSE.csv',
 'RELATIVEHEALTHIMPORTANCE.csv',
 'RISKFACTORSANDACCESSTOCARE.csv',
 'SUMMARYMEASURESOFHEALTH.csv',
 'VUNERABLEPOPSANDENVHEALTH.csv']
```

Demographics

In [13]:
```
df_Demog=pd.read_csv('DEMOGRAPHICS.csv')
#df_Demog=df_Demog.loc[df_Demog['CHSI_State_Name']=='Illinois']
#df_Demog.columns
df_Demog=df_Demog[['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name','CHSI_State_Name', 'CHSI_State_Abbr', 'S
ListofNans=[-9999,-2222,-2222.2,-2,-1111.1,-1111,-1,-9998.9]
df_Demog=df_Demog.replace([i for i in ListofNans], np.NAN)#replacing odd values with nan
df_Demog.head()
```

Out[13]:

	State_FIPS_Code	County_FIPS_Code	CHSI_County_Name	CHSI_State_Name	CHSI_State_Abbr	Strata_ID_Number	Population_Size	Populatio
0	1	1	Autauga	Alabama	AL	29	48612	
1	1	3	Baldwin	Alabama	AL	16	162586	
2	1	5	Barbour	Alabama	AL	51	28414	
3	1	7	Bibb	Alabama	AL	42	21516	
4	1	9	Blount	Alabama	AL	28	55725	

Granularity : Every record in the dataframe is a record of one county in the US

In [14]:
```
PovertyStats=df_Demog['Poverty'].describe()
print("Poverty Across Counties Stats\n\n ",PovertyStats)
```

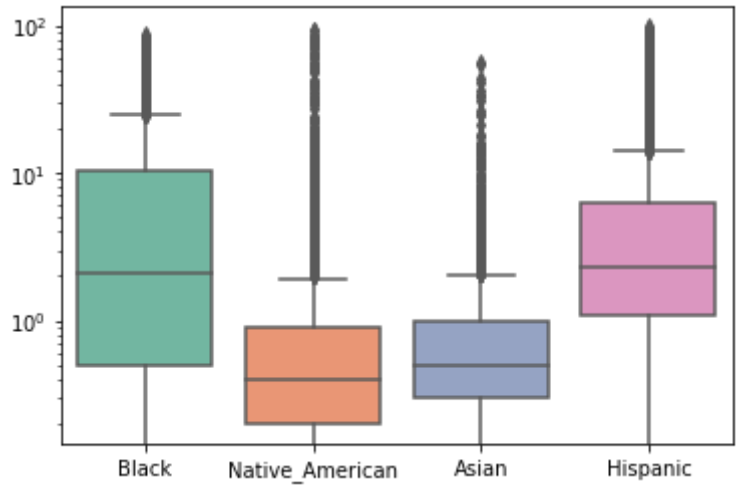
```
Poverty Across Counties Stats

  count    3140.000000
mean       13.350159
std         4.883308
min         2.200000
25%         9.800000
50%        12.600000
75%        16.200000
max        36.200000
Name: Poverty, dtype: float64
```

In [15]:
```
Races_df=df_Demog[['White','Black','Asian', 'Hispanic']]
print("Races \n\n",Races_df.describe())
```

Races

	White	Black	Asian	Hispanic
count	3141.000000	3141.000000	3141.000000	3141.000000
mean	87.017892	8.986692	1.123050	7.017988
std	16.150479	14.545659	2.757237	12.464727
min	4.700000	0.000000	0.000000	0.000000
25%	82.800000	0.500000	0.300000	1.100000
50%	94.100000	2.100000	0.500000	2.300000
75%	97.600000	10.300000	1.000000	6.300000
max	100.000000	86.000000	55.900000	97.500000

In [16]:
```
ax = sns.boxplot(data=df_Demog[['Black', 'Native_American','Asian', 'Hispanic']], palette="Set2")
ax.set_yscale('log')
```



In [17]:
```
Ages_df=df_Demog[['Age_19_Under','Age_19_64', 'Age_65_84','Age_85_and_Over']]
print("Age Groups \n\n",Ages_df.describe())
```

```
Age Groups

        Age_19_Under    Age_19_64    Age_65_84  Age_85_and_Over
count   3141.000000  3141.000000  3141.000000      3141.000000
mean      24.806527    60.289398    12.789430         2.115409
std        3.281777     3.356056     3.334035         0.949119
min        1.400000    47.600000     2.100000         0.100000
25%       22.700000    58.300000    10.700000         1.500000
50%       24.600000    60.300000    12.500000         1.900000
75%       26.400000    62.300000    14.700000         2.600000
max       47.200000    83.300000    29.200000         7.600000
```

In [18]:
```
Race_Age=df_Demog[['White','Black','Asian', 'Hispanic','Age_19_Under','Age_19_64', 'Age_65_84','Age_85_and_Over']]
Race_AgeCorr=pd.DataFrame(Race_Age.corr())
Race_AgeCorr=Race_AgeCorr[Race_AgeCorr.index.isin(['White','Black','Asian', 'Hispanic'])]
Race_AgeCorr=Race_AgeCorr[['Age_19_Under','Age_19_64', 'Age_65_84','Age_85_and_Over']]
print(Race_AgeCorr)
```

```
          Age_19_Under  Age_19_64  Age_65_84  Age_85_and_Over
White        -0.369316  -0.042609   0.319331         0.305791
Black         0.211044   0.064710  -0.207554        -0.228924
Asian         0.033618   0.254596  -0.251417        -0.132692
Hispanic      0.377215  -0.166735  -0.155450        -0.167781
```

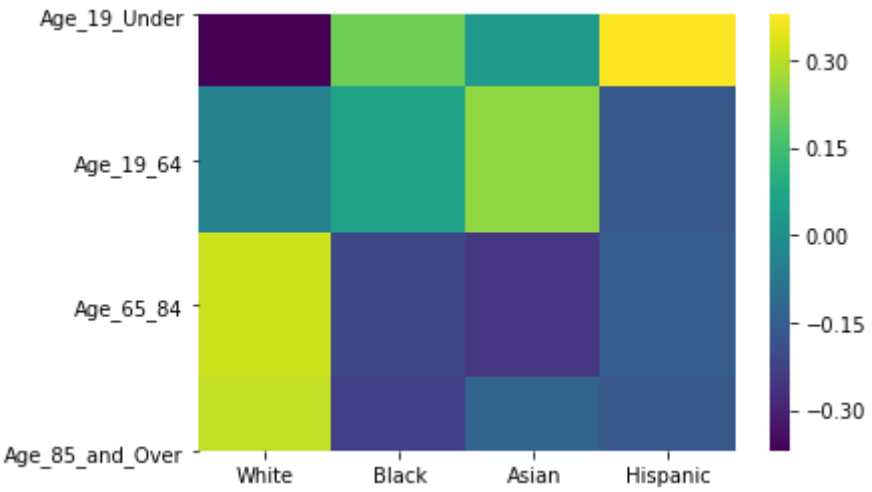
The table represents the correlation values among the features. We See races except "White" to be negatively correlated with the age groups of 19 and above.

Counties in which older age populations are there in more percentages have lower number of blacks, asians and hispanics, i.e. they are negatively correlated to the higher ages

1. Can we infer that these races tend to live in the counties which have much younger populations?
2. Or these races might have populations of young people which make it seem so?

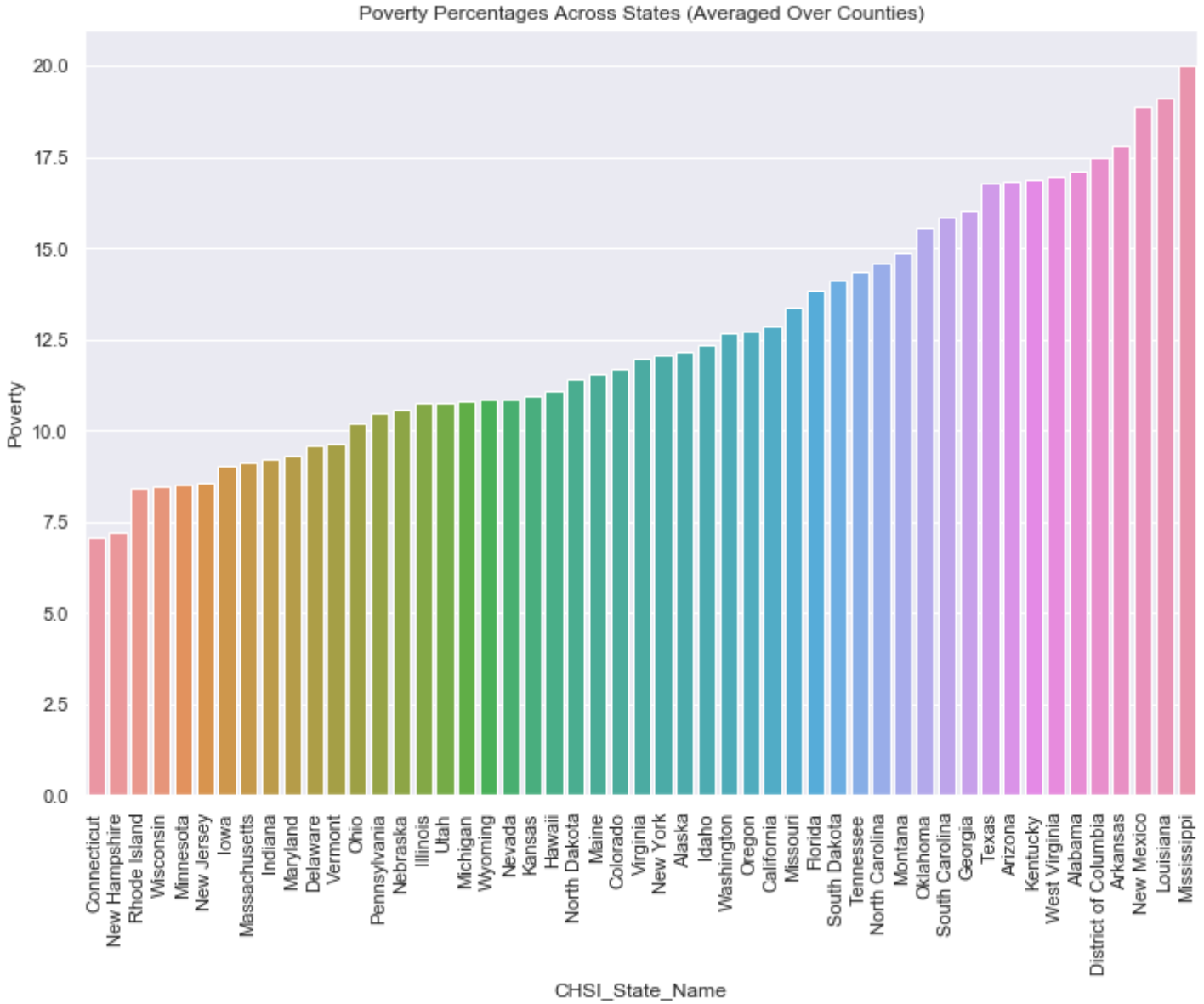
Any how the correlation values are not that strong.

In [19]:
```
ax = sns.heatmap(Race_AgeCorr.transpose(),cmap="viridis" )
```



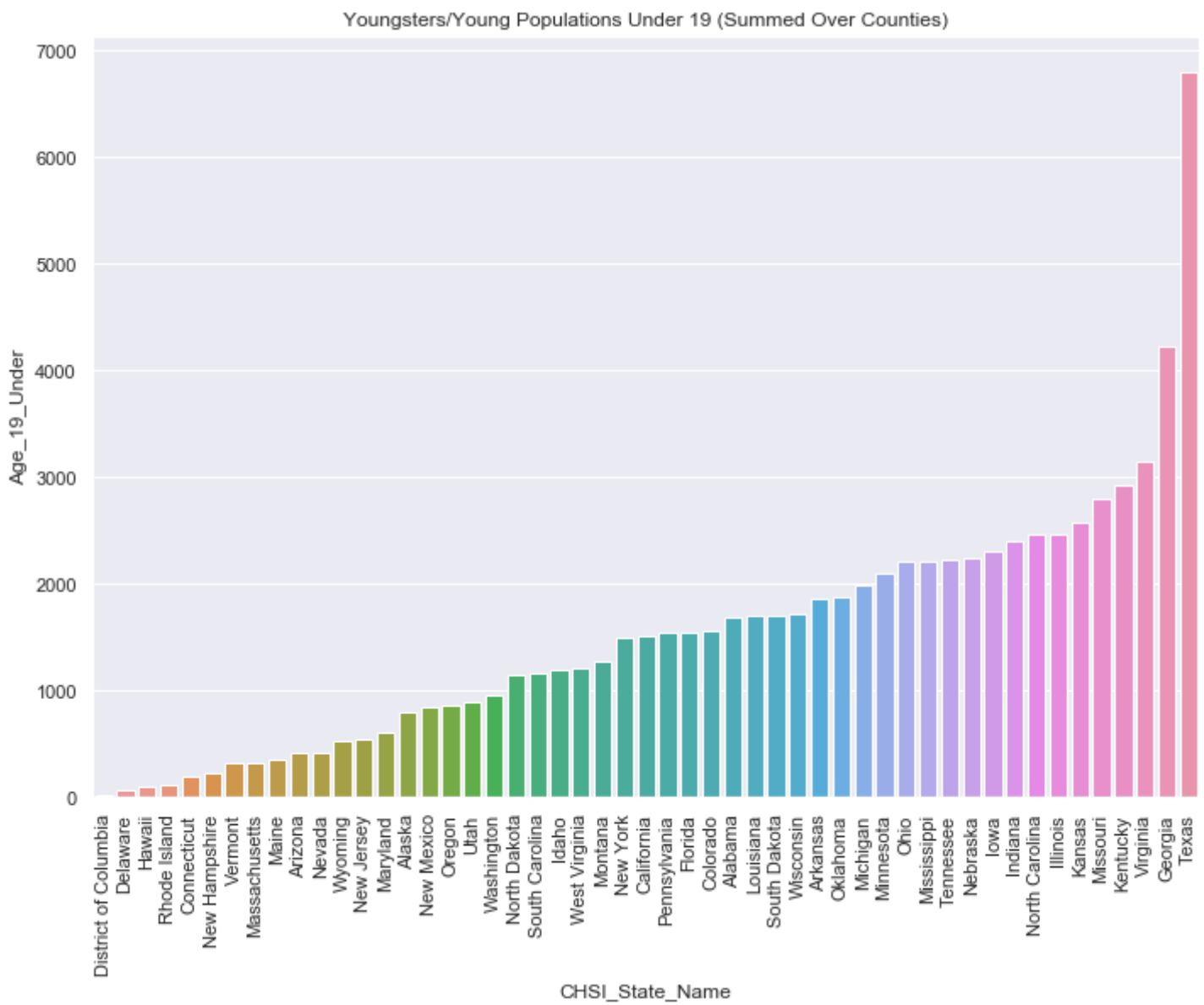
In [20]:
```
PovertyDf=df_Demog[['Poverty']].groupby(df_Demog['CHSI_State_Name']).mean().sort_values(by=['Poverty'])
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=PovertyDf.index, y="Poverty", data=PovertyDf)
plt.xticks(rotation=90)
plt.title('Poverty Percentages Across States (Averaged Over Counties)')
```

Out[20]:
```
Text(0.5, 1.0, 'Poverty Percentages Across States (Averaged Over Counties)')
```


Poverty Percentages Across States (Averaged Over Counties)
Poverty
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Connecticut
New Hampshire
Rhode Island
Wisconsin
Minnesota
New Jersey
Iowa
Massachusetts
Indiana
Maryland
Delaware
Vermont
Ohio
Pennsylvania
Nebraska
Illinois
Utah
Michigan
Wyoming
Nevada
Kansas
Hawaii
North Dakota
Maine
Colorado
Virginia
New York
Alaska
Idaho
Washington
Oregon
California
Missouri
Florida
South Dakota
Tennessee
North Carolina
Montana
Oklahoma
South Carolina
Georgia
Texas
Arizona
Kentucky
West Virginia
Alabama
District of Columbia
Arkansas
New Mexico
Louisiana
Mississippi
CHSI_State_Name

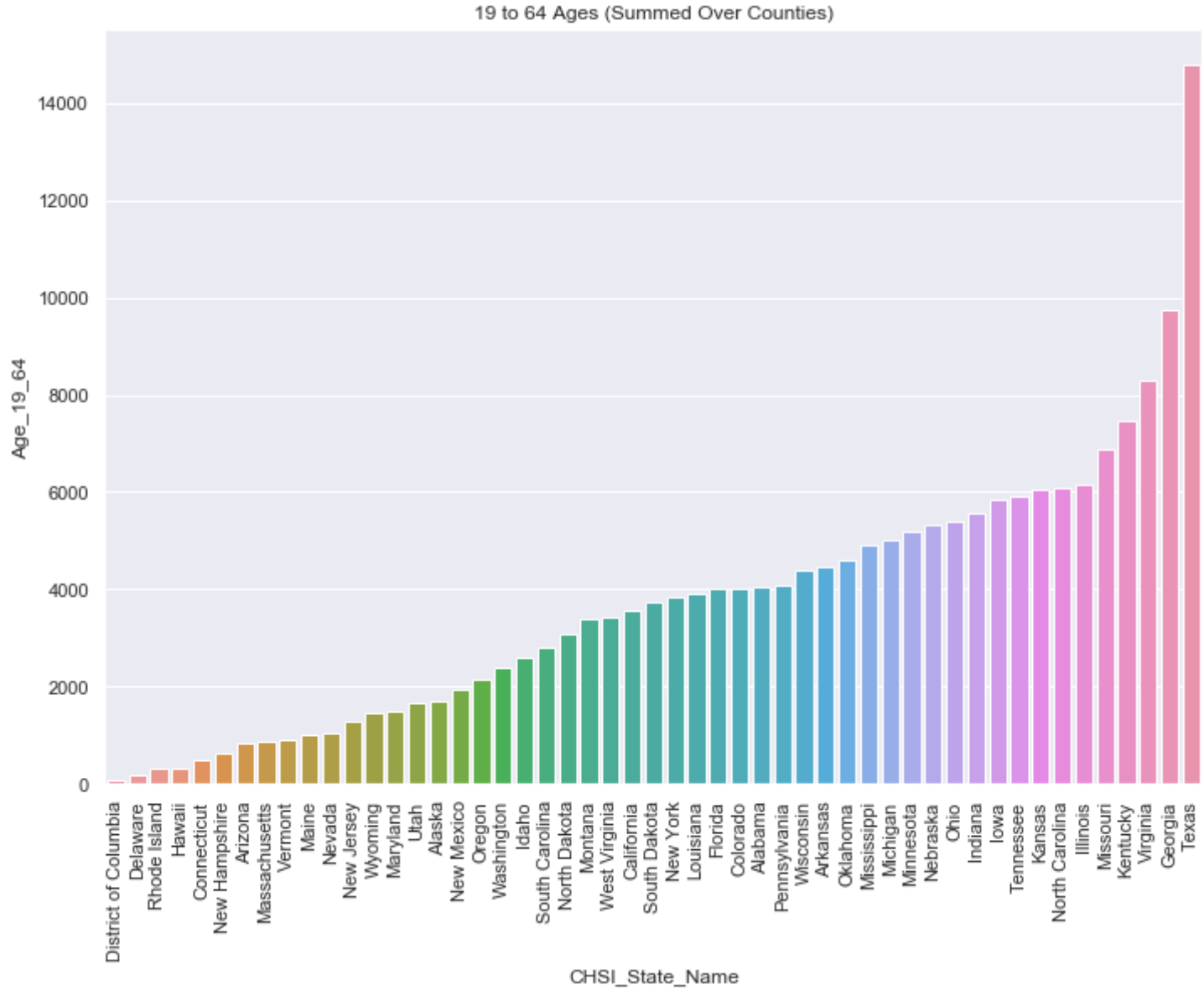
In [21]:
```
Age_19_Under_df=df_Demog[['Age_19_Under']].groupby(df_Demog['CHSI_State_Name']).sum().sort_values(by=['Age_19_Under']
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=Age_19_Under_df.index, y="Age_19_Under", data=Age_19_Under_df)
plt.xticks(rotation=90)
plt.title('Youngsters/Young Populations Under 19 (Summed Over Counties)')
```

Out[21]:
```
Text(0.5, 1.0, 'Youngsters/Young Populations Under 19 (Summed Over Counties)')
```


Youngsters/Young Populations Under 19 (Summed Over Counties)
Age_19_Under
0
1000
2000
3000
4000
5000
6000
7000
District of Columbia
Delaware
Hawaii
Rhode Island
Connecticut
New Hampshire
Vermont
Massachusetts
Maine
Arizona
Nevada
Wyoming
New Jersey
Maryland
Alaska
New Mexico
Oregon
Utah
Washington
North Dakota
South Carolina
Idaho
West Virginia
Montana
New York
California
Pennsylvania
Florida
Colorado
Alabama
Louisiana
South Dakota
Wisconsin
Arkansas
Oklahoma
Michigan
Minnesota
Ohio
Mississippi
Tennessee
Nebraska
Iowa
Indiana
North Carolina
Illinois
Kansas
Missouri
Kentucky
Virginia
Georgia
Texas
CHSI_State_Name

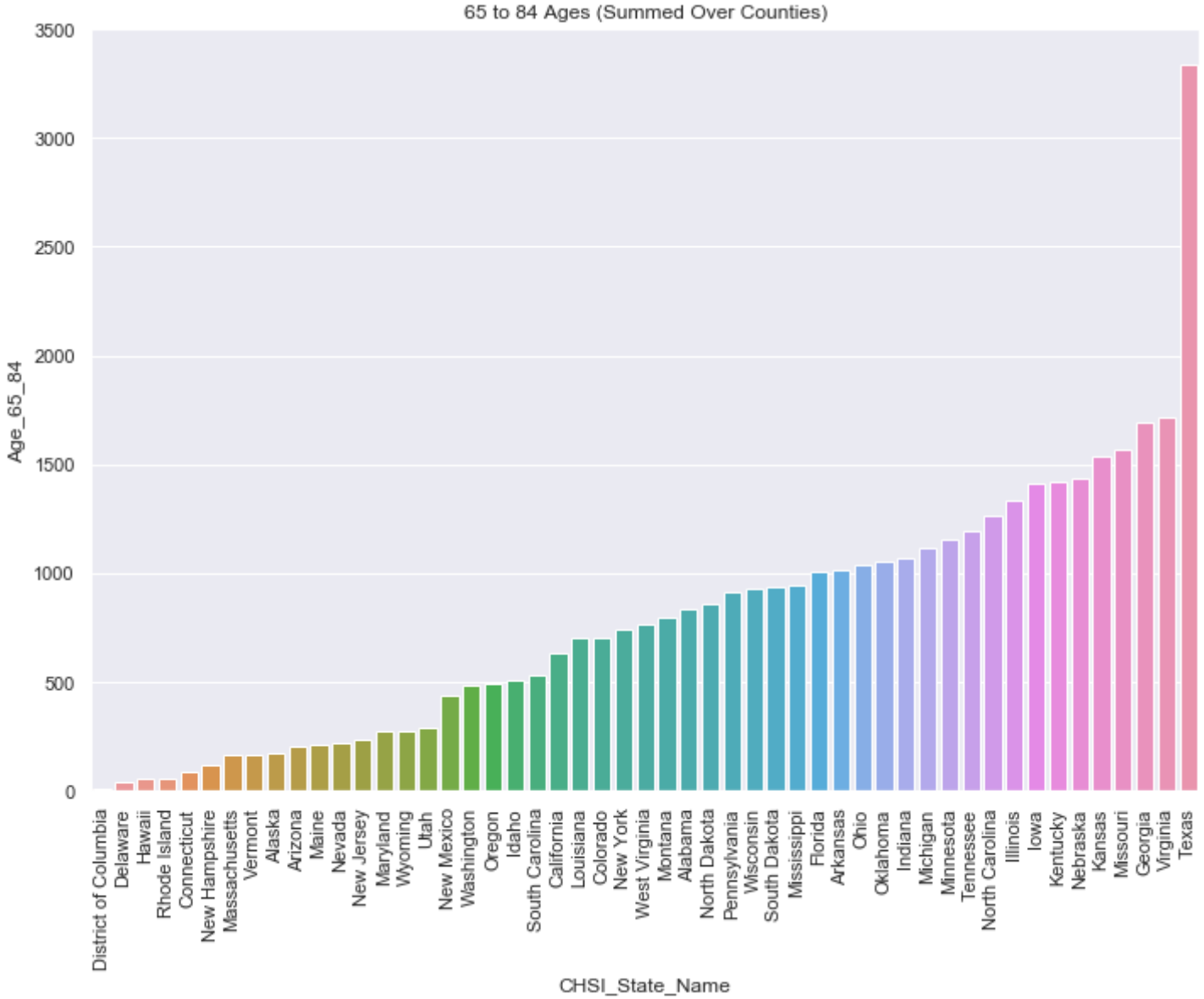
In [22]:
```
Age_19_64_df=df_Demog[['Age_19_64']].groupby(df_Demog['CHSI_State_Name']).sum().sort_values(by=['Age_19_64'])
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=Age_19_64_df.index, y="Age_19_64", data=Age_19_64_df)
plt.xticks(rotation=90)
plt.title('19 to 64 Ages (Summed Over Counties)')
```

Out[22]:
```
Text(0.5, 1.0, '19 to 64 Ages (Summed Over Counties)')
```


19 to 64 Ages (Summed Over Counties)
Age_19_64
0
2000
4000
6000
8000
10000
12000
14000
CHSI_State_Name
District of Columbia
Delaware
Rhode Island
Hawaii
Connecticut
New Hampshire
Arizona
Massachusetts
Vermont
Maine
Nevada
New Jersey
Wyoming
Maryland
Utah
Alaska
New Mexico
Oregon
Washington
Idaho
South Carolina
North Dakota
Montana
West Virginia
California
South Dakota
New York
Louisiana
Florida
Colorado
Alabama
Pennsylvania
Wisconsin
Arkansas
Oklahoma
Mississippi
Michigan
Minnesota
Nebraska
Ohio
Indiana
Iowa
Tennessee
Kansas
North Carolina
Illinois
Missouri
Kentucky
Virginia
Georgia
Texas

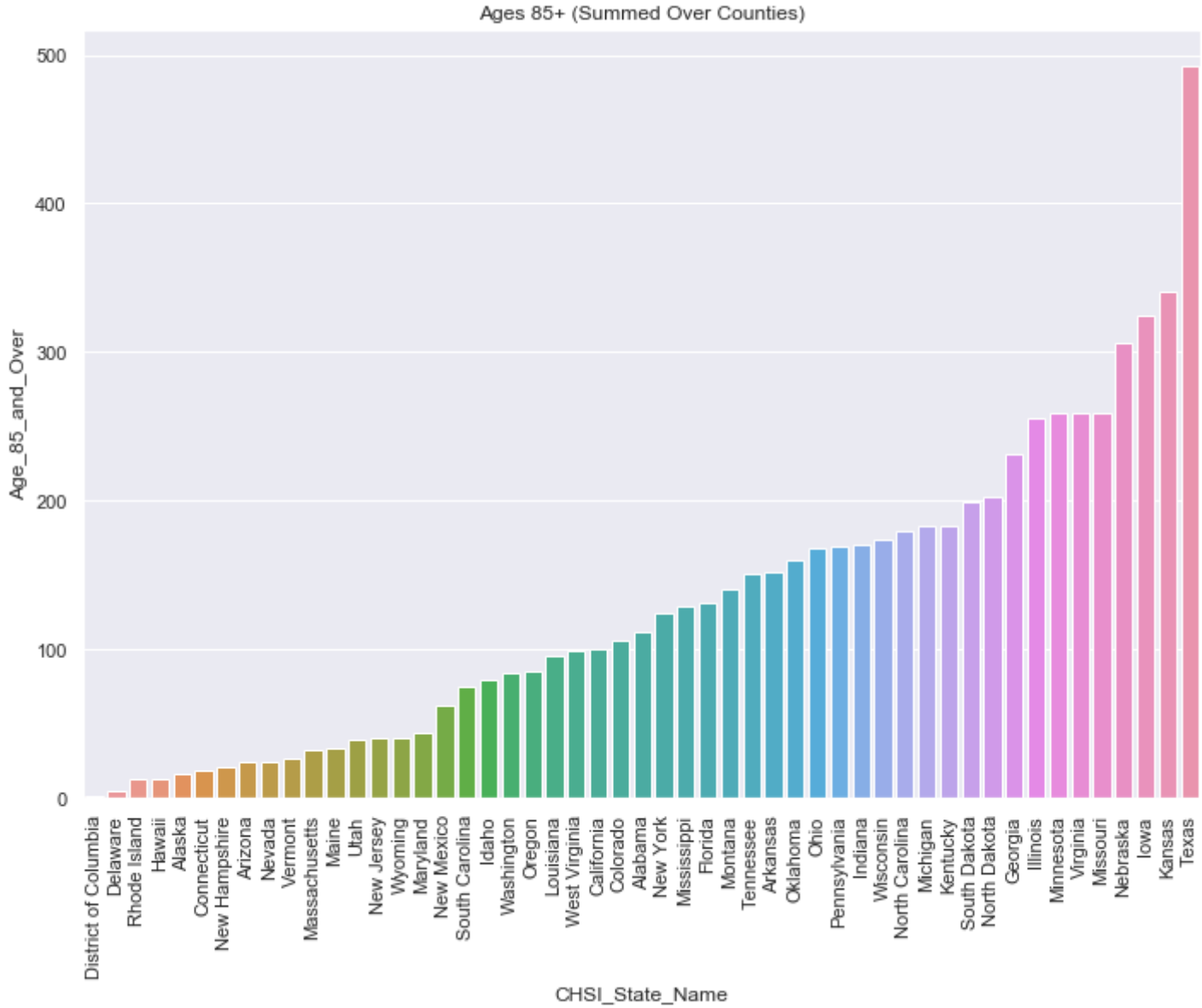
In [23]:
```
Age_65_84_df=df_Demog[['Age_65_84']].groupby(df_Demog['CHSI_State_Name']).sum().sort_values(by=['Age_65_84'])
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=Age_65_84_df.index, y="Age_65_84", data=Age_65_84_df)
plt.xticks(rotation=90)
plt.title('65 to 84 Ages (Summed Over Counties)')
```

Out[23]:
```
Text(0.5, 1.0, '65 to 84 Ages (Summed Over Counties)')
```


65 to 84 Ages (Summed Over Counties)
Age_65_84
0
500
1000
1500
2000
2500
3000
3500
District of Columbia
Delaware
Hawaii
Rhode Island
Connecticut
New Hampshire
Massachusetts
Vermont
Alaska
Arizona
Maine
Nevada
New Jersey
Maryland
Wyoming
Utah
New Mexico
Washington
Oregon
Idaho
South Carolina
California
Louisiana
Colorado
New York
West Virginia
Montana
Alabama
North Dakota
Pennsylvania
Wisconsin
South Dakota
Mississippi
Florida
Arkansas
Ohio
Oklahoma
Indiana
Michigan
Minnesota
Tennessee
North Carolina
Illinois
Iowa
Kentucky
Nebraska
Kansas
Missouri
Georgia
Virginia
Texas
CHSI_State_Name

In [24]:
```
Age_85_and_Over_df=df_Demog[['Age_85_and_Over']].groupby(df_Demog['CHSI_State_Name']).sum().sort_values(by=['Age_85_a
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=Age_85_and_Over_df.index, y="Age_85_and_Over", data=Age_85_and_Over_df)
plt.xticks(rotation=90)
plt.title('Ages 85+ (Summed Over Counties)')
```

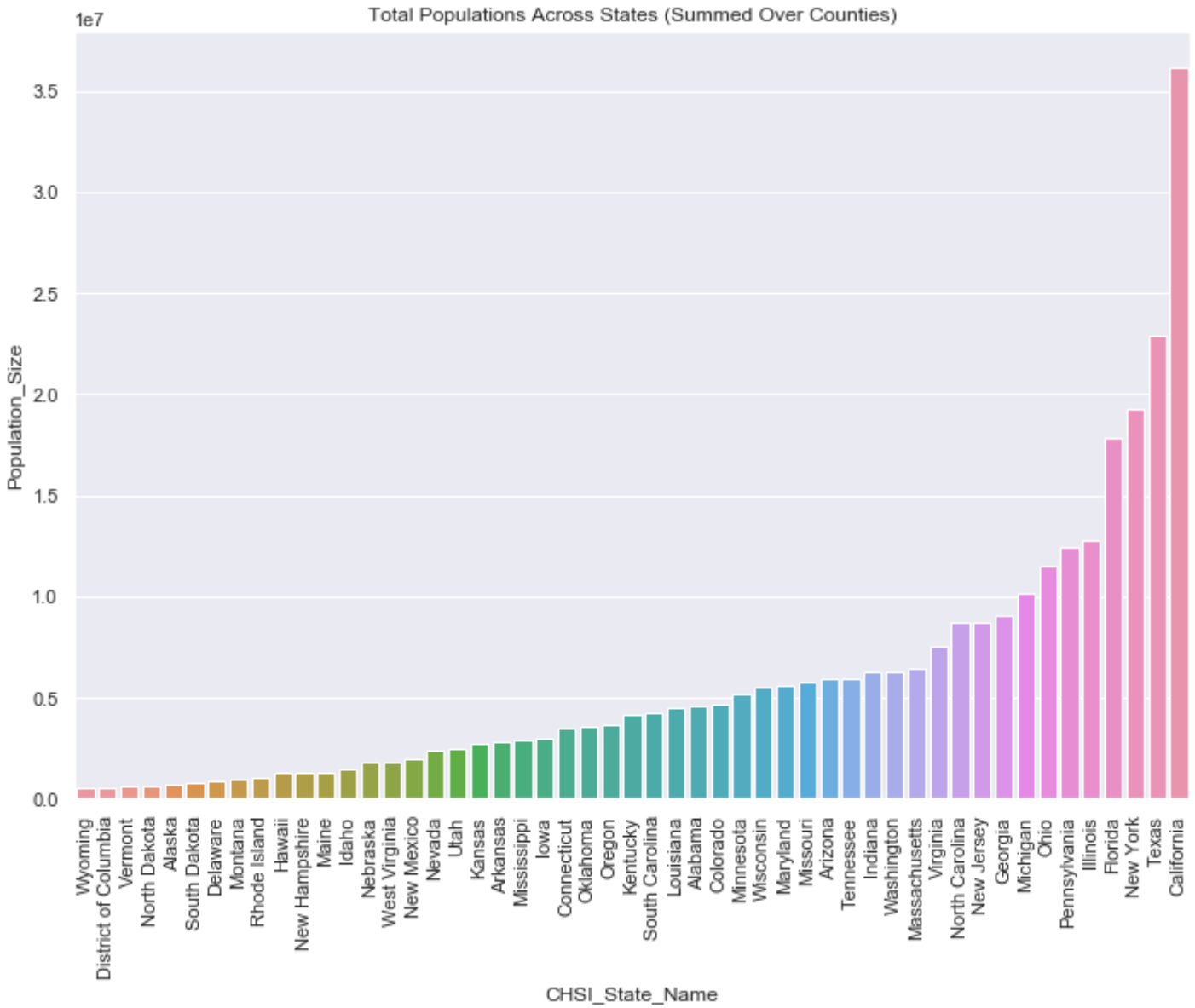
Out[24]:
```
Text(0.5, 1.0, 'Ages 85+ (Summed Over Counties)')
```


Ages 85+ (Summed Over Counties)
Age_85_and_Over
0
100
200
300
400
500
District of Columbia
Delaware
Rhode Island
Hawaii
Alaska
Connecticut
New Hampshire
Arizona
Nevada
Vermont
Massachusetts
Maine
Utah
New Jersey
Wyoming
Maryland
New Mexico
South Carolina
Idaho
Washington
Oregon
Louisiana
West Virginia
California
Colorado
Alabama
New York
Mississippi
Florida
Montana
Tennessee
Arkansas
Oklahoma
Ohio
Pennsylvania
Indiana
Wisconsin
North Carolina
Michigan
Kentucky
South Dakota
North Dakota
Georgia
Illinois
Minnesota
Virginia
Missouri
Nebraska
Iowa
Kansas
Texas
CHSI_State_Name

In [25]:
```
PopSize=df_Demog[['Population_Size']].groupby(df_Demog['CHSI_State_Name']).sum().sort_values(by=['Population_Size'])
chart = sns.barplot(x=PopSize.index, y="Population_Size", data=PopSize)
plt.xticks(rotation=90)
plt.title('Total Populations Across States (Summed Over Counties)')
PopSize.tail()
```

Out[25]:

CHSI_State_Name	Population_Size
Illinois	12763371
Florida	17789864
New York	19254630
Texas	22859968
California	36132147


Total Populations Across States (Summed Over Counties)
1e7
Population_Size
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Wyoming
District of Columbia
Vermont
North Dakota
Alaska
South Dakota
Delaware
Montana
Rhode Island
Hawaii
New Hampshire
Maine
Idaho
Nebraska
West Virginia
New Mexico
Nevada
Utah
Kansas
Arkansas
Mississippi
Iowa
Connecticut
Oklahoma
Oregon
Kentucky
South Carolina
Louisiana
Alabama
Colorado
Minnesota
Wisconsin
Maryland
Missouri
Arizona
Tennessee
Indiana
Washington
Massachusetts
Virginia
North Carolina
New Jersey
Georgia
Michigan
Ohio
Pennsylvania
Illinois
Florida
New York
Texas
California
CHSI_State_Name

In the below two tables we see the distribution of the three majority races across different states

In [26]:
```
Races_df_grp=df_Demog[['White','Black','Asian']].groupby(df_Demog['CHSI_State_Name']).mean().sort_values(by=['White',
print(Races_df_grp.head(10))
print("\n\n\n")
print(Races_df_grp.tail(10))
```

CHSI_State_Name	White	Black	Asian
Hawaii	34.480000	0.940000	46.260000
District of Columbia	38.000000	57.000000	3.200000
Alaska	54.485185	1.174074	4.851852
Mississippi	58.269512	40.413415	0.437805
South Carolina	61.213043	37.119565	0.691304
Louisiana	66.328125	31.707813	0.740625
Alabama	69.795522	28.510448	0.455224
Georgia	70.218868	27.981761	0.928302
North Carolina	75.408000	21.386000	0.934000
Delaware	76.500000	19.533333	2.133333

CHSI_State_Name	White	Black	Asian
Kansas	95.712381	1.885714	0.694286
Indiana	95.984783	2.428261	0.677174
Wyoming	96.369565	0.460870	0.517391
Idaho	96.493182	0.406818	0.645455
West Virginia	96.907273	1.941818	0.389091
New Hampshire	96.960000	0.680000	1.230000
Nebraska	97.272043	0.492473	0.504301
Vermont	97.307143	0.507143	0.728571
Maine	97.331250	0.550000	0.600000
Iowa	97.671717	0.844444	0.730303

Vulnerable Populations & Environmental Health

In [27]:
```
df_VPEH=pd.read_csv('VUNERABLEPOPSANDENVHEALTH.csv')
Demo_VPEH_df=df_Demog.merge(df_VPEH, on=['State_FIPS_Code', 'County_FIPS_Code'], how='left', indicator=True)
Demo_VPEH_df=Demo_VPEH_df.replace([i for i in ListofNans], np.NAN)
imp_cols=['No_HS_Diploma', 'Unemployed', 'Sev_Work_Disabled', 'Major_Depression','Recent_Drug_Use']
Demo_VPEH_df[imp_cols[0]+str('%')]=Demo_VPEH_df[imp_cols[0]]/Demo_VPEH_df['Population_Size']
Demo_VPEH_df[imp_cols[1]+str('%')]=Demo_VPEH_df[imp_cols[1]]/Demo_VPEH_df['Population_Size']
Demo_VPEH_df[imp_cols[2]+str('%')]=Demo_VPEH_df[imp_cols[2]]/Demo_VPEH_df['Population_Size']
Demo_VPEH_df[imp_cols[3]+str('%')]=Demo_VPEH_df[imp_cols[3]]/Demo_VPEH_df['Population_Size']
Demo_VPEH_df[imp_cols[4]+str('%')]=Demo_VPEH_df[imp_cols[4]]/Demo_VPEH_df['Population_Size']
Demo_VPEH_df.head()
```

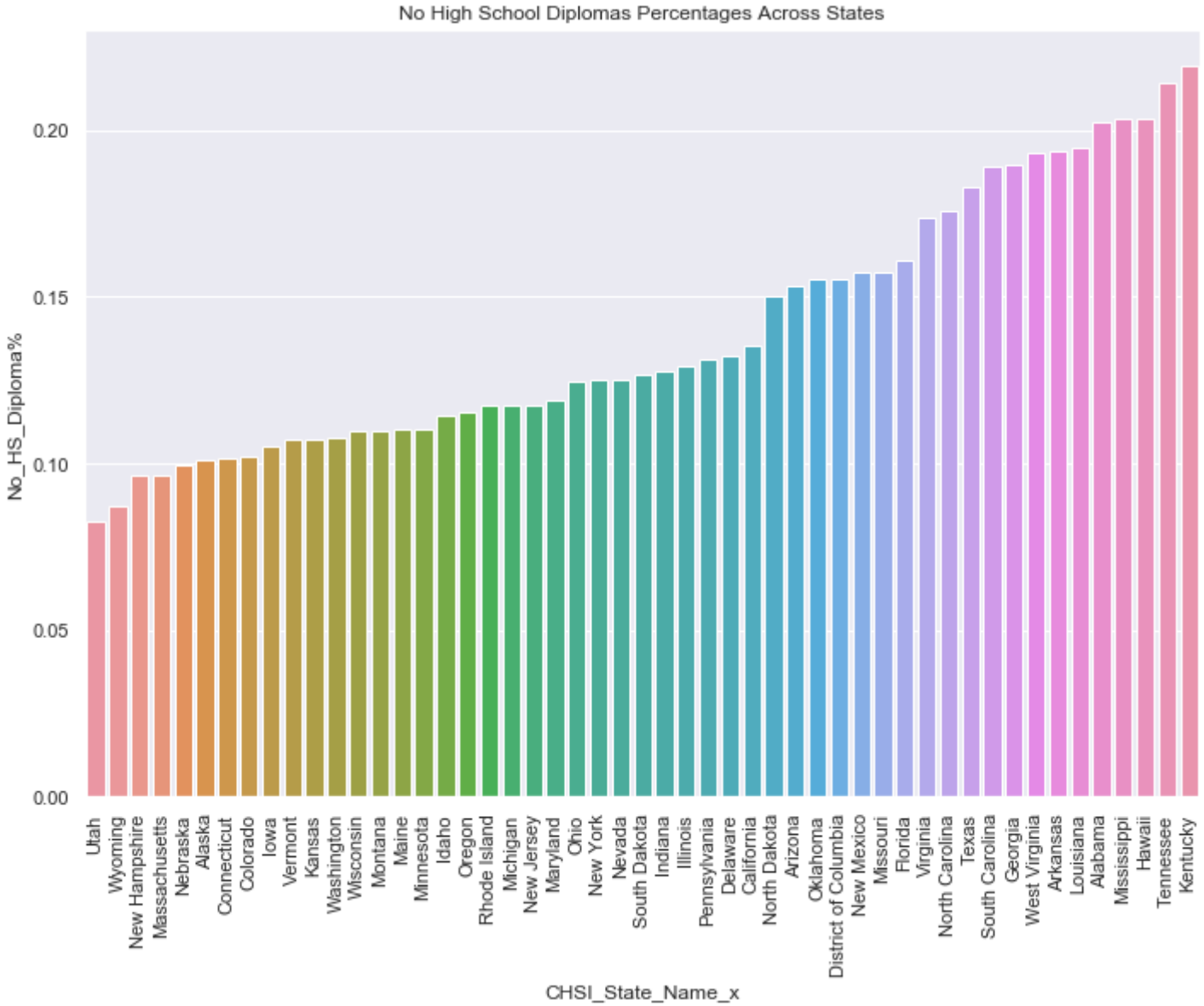
Out[27]:

	State_FIPS_Code	County_FIPS_Code	CHSI_County_Name_x	CHSI_State_Name_x	CHSI_State_Abbr_x	Strata_ID_Number_x	Population_Size	I
0	1	1	Autauga	Alabama	AL	29	48612	
1	1	3	Baldwin	Alabama	AL	16	162586	
2	1	5	Barbour	Alabama	AL	51	28414	
3	1	7	Bibb	Alabama	AL	42	21516	
4	1	9	Blount	Alabama	AL	28	55725	

5 rows × 50 columns

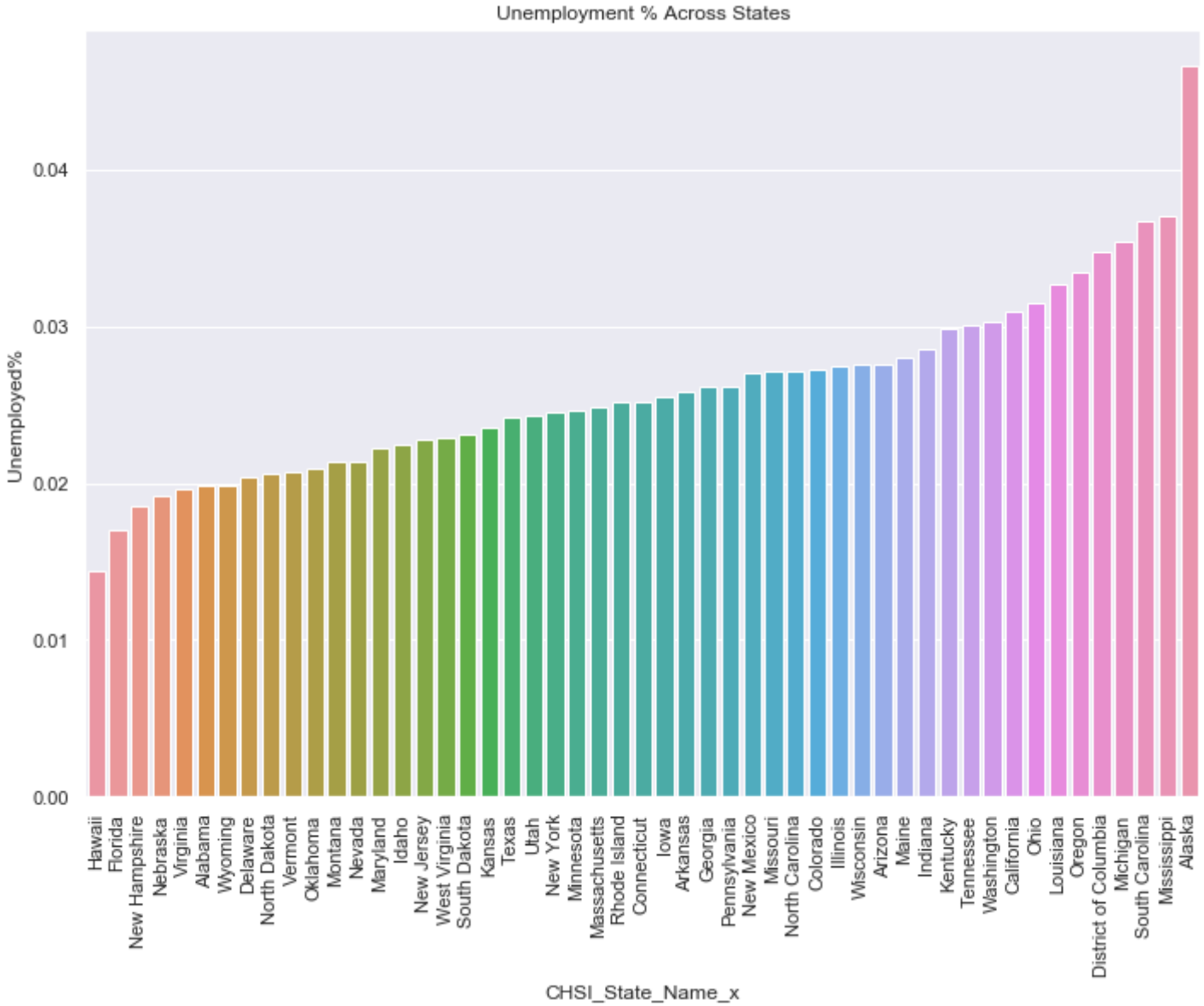
In [28]:
```
plot_cols=['No_HS_Diploma%', 'Unemployed%','Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%']
i=0
plot1=Demo_VPEH_df[plot_cols].groupby(Demo_VPEH_df['CHSI_State_Name_x']).mean().sort_values(by=plot_cols[i])
sns.set(rc={'figure.figsize':(11.7,8.27)})
name='chart'+ str(i)
name = sns.barplot(x=plot1.index, y=plot1[plot_cols[i]], data=plot1)
plt.xticks(rotation=90)
plt.title('No High School Diplomas Percentages Across States')
```

Out[28]:
```
Text(0.5, 1.0, 'No High School Diplomas Percentages Across States')
```


No High School Diplomas Percentages Across States
No_HS_Diploma%
0.00
0.05
0.10
0.15
0.20
Utah
Wyoming
New Hampshire
Massachusetts
Nebraska
Alaska
Connecticut
Colorado
Iowa
Vermont
Kansas
Washington
Wisconsin
Montana
Maine
Minnesota
Idaho
Oregon
Rhode Island
Michigan
New Jersey
Maryland
Ohio
New York
Nevada
South Dakota
Indiana
Illinois
Pennsylvania
Delaware
California
North Dakota
Arizona
Oklahoma
District of Columbia
New Mexico
Missouri
Florida
Virginia
North Carolina
Texas
South Carolina
Georgia
West Virginia
Arkansas
Louisiana
Alabama
Mississippi
Hawaii
Tennessee
Kentucky
CHSI_State_Name_x

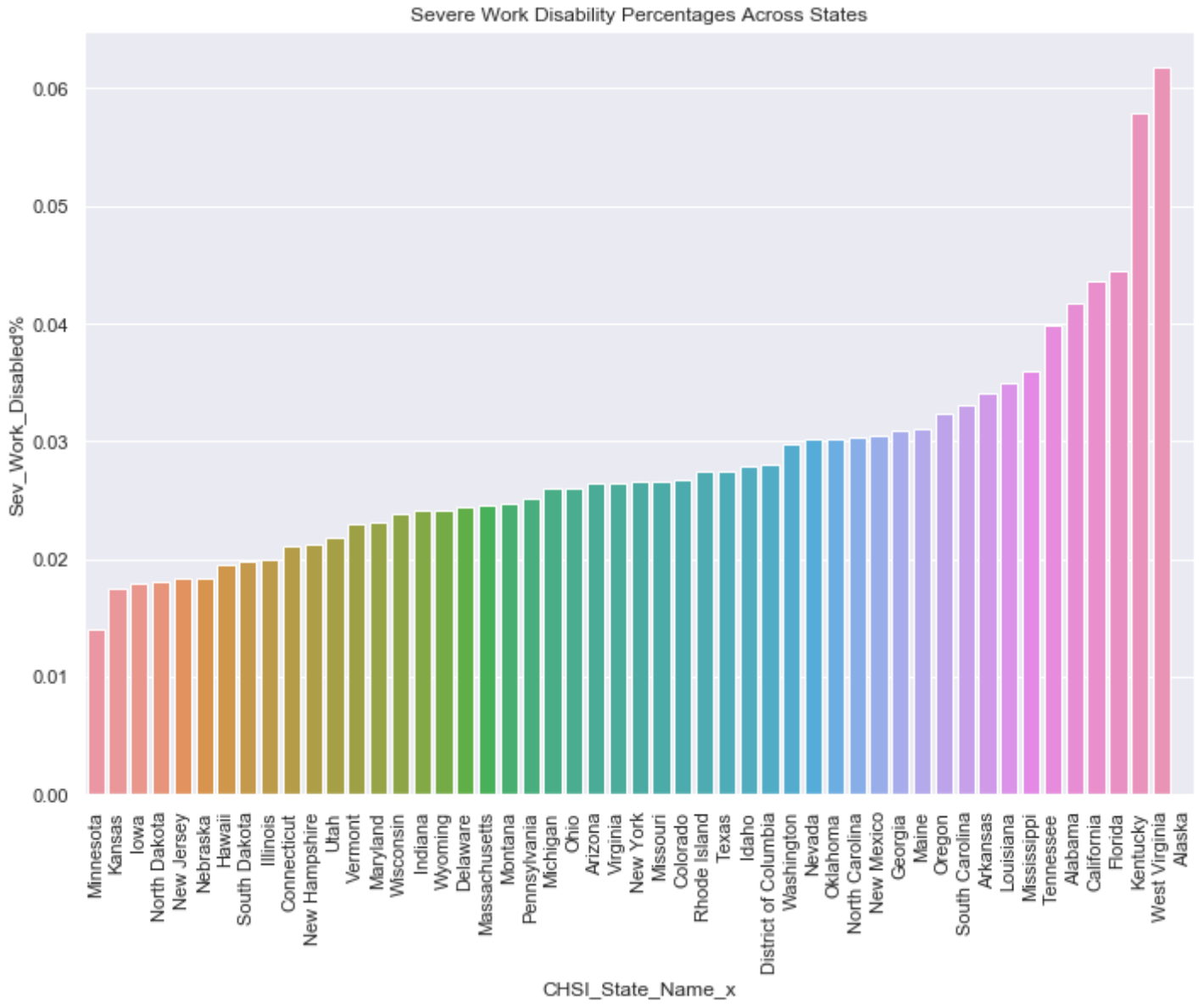
In [29]:
```
plot_cols=['No_HS_Diploma%', 'Unemployed%','Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%']
i=1
plot1=Demo_VPEH_df[plot_cols].groupby(Demo_VPEH_df['CHSI_State_Name_x']).mean().sort_values(by=plot_cols[i])
sns.set(rc={'figure.figsize':(11.7,8.27)})
name='chart'+ str(i)
name = sns.barplot(x=plot1.index, y=plot1[plot_cols[i]], data=plot1)
plt.xticks(rotation=90)
plt.title('Unemployment % Across States')
```

Out[29]:
```
Text(0.5, 1.0, 'Unemployment % Across States')
```


Unemployment % Across States
Unemployed%
0.00
0.01
0.02
0.03
0.04
Hawaii
Florida
New Hampshire
Nebraska
Virginia
Alabama
Wyoming
Delaware
North Dakota
Vermont
Oklahoma
Montana
Nevada
Maryland
Idaho
New Jersey
West Virginia
South Dakota
Kansas
Texas
Utah
New York
Minnesota
Massachusetts
Rhode Island
Connecticut
Iowa
Arkansas
Georgia
Pennsylvania
New Mexico
Missouri
North Carolina
Colorado
Illinois
Wisconsin
Arizona
Maine
Indiana
Kentucky
Tennessee
Washington
California
Ohio
Louisiana
Oregon
District of Columbia
Michigan
South Carolina
Mississippi
Alaska
CHSI_State_Name_x

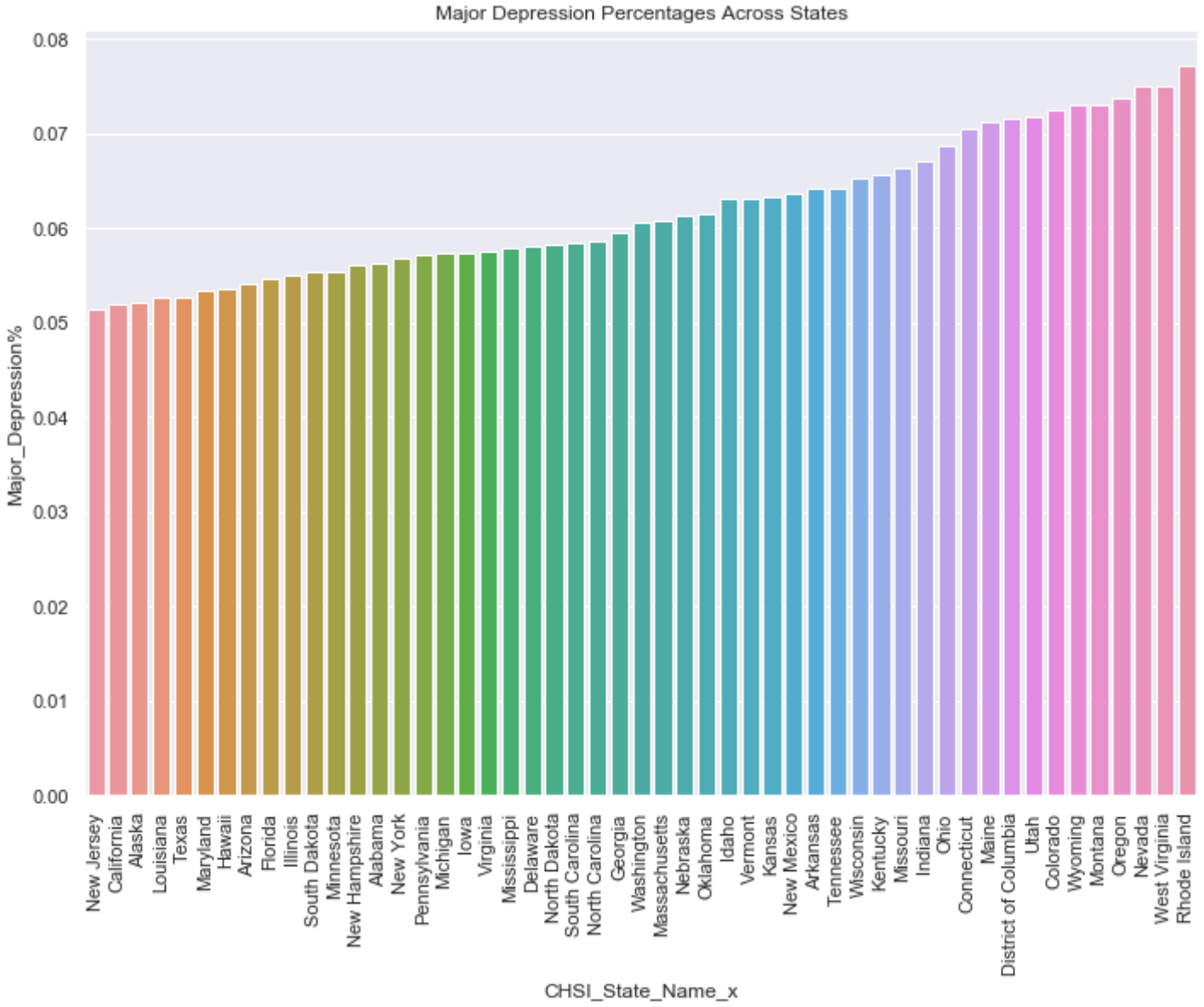
In [30]:
```
plot_cols=['No_HS_Diploma%', 'Unemployed%','Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%']
i=2
plot1=Demo_VPEH_df[plot_cols].groupby(Demo_VPEH_df['CHSI_State_Name_x']).mean().sort_values(by=plot_cols[i])
sns.set(rc={'figure.figsize':(11.7,8.27)})
name='chart'+ str(i)
name = sns.barplot(x=plot1.index, y=plot1[plot_cols[i]], data=plot1)
plt.xticks(rotation=90)
plt.title('Severe Work Disability Percentages Across States')
```

Out[30]:
```
Text(0.5, 1.0, 'Severe Work Disability Percentages Across States')
```


Severe Work Disability Percentages Across States
Sev_Work_Disabled%
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Minnesota
Kansas
Iowa
North Dakota
New Jersey
Nebraska
Hawaii
South Dakota
Illinois
Connecticut
New Hampshire
Utah
Vermont
Maryland
Wisconsin
Indiana
Wyoming
Delaware
Massachusetts
Montana
Pennsylvania
Michigan
Ohio
Arizona
Virginia
New York
Missouri
Colorado
Rhode Island
Texas
Idaho
District of Columbia
Washington
Nevada
Oklahoma
North Carolina
New Mexico
Georgia
Maine
Oregon
South Carolina
Arkansas
Louisiana
Mississippi
Tennessee
Alabama
California
Florida
Kentucky
West Virginia
Alaska
CHSI_State_Name_x

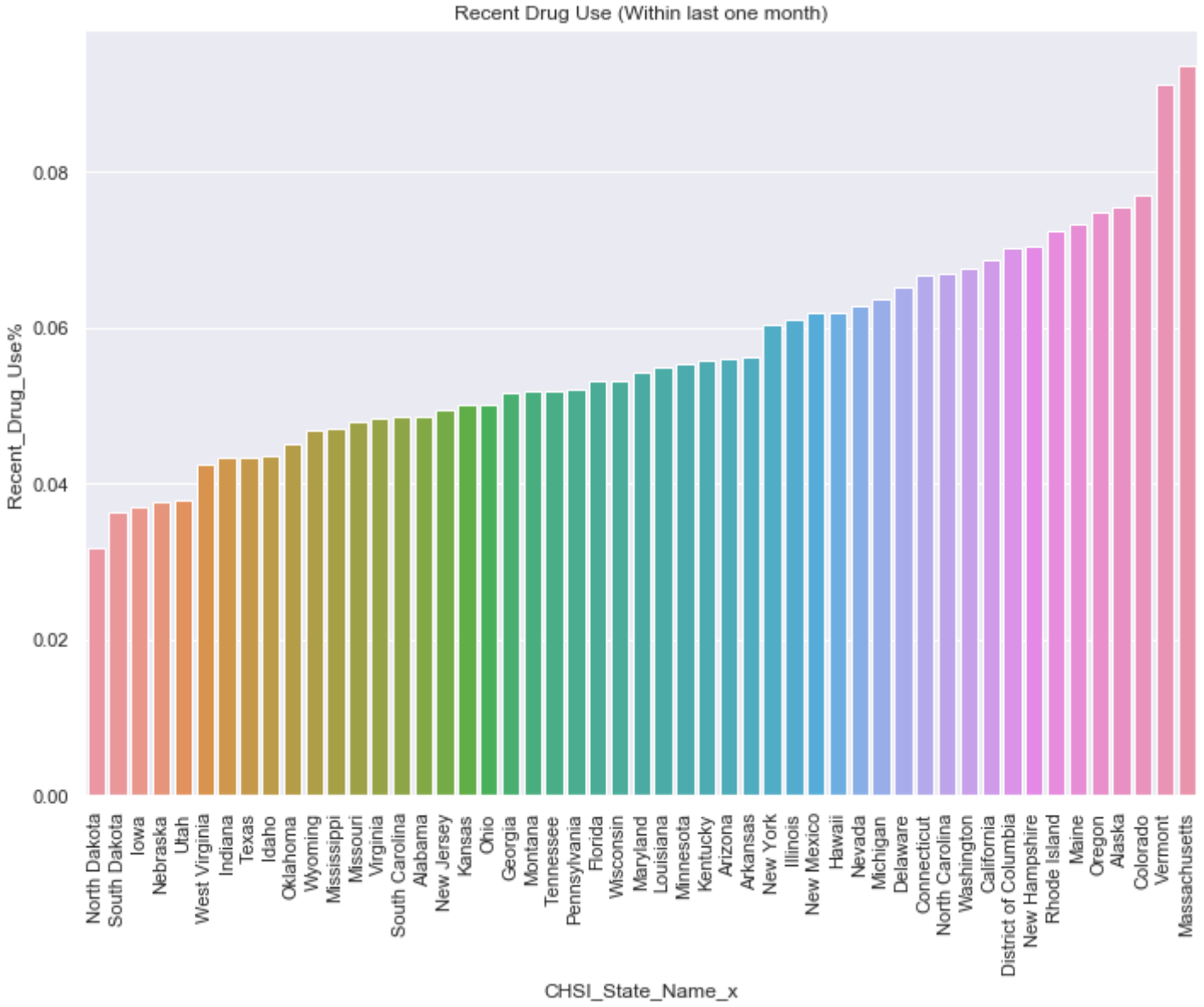
In [31]:
```
plot_cols=['No_HS_Diploma%', 'Unemployed%','Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%']
i=3
plot1=Demo_VPEH_df[plot_cols].groupby(Demo_VPEH_df['CHSI_State_Name_x']).mean().sort_values(by=plot_cols[i])
sns.set(rc={'figure.figsize':(11.7,8.27)})
name='chart'+ str(i)
name = sns.barplot(x=plot1.index, y=plot1[plot_cols[i]], data=plot1)
plt.xticks(rotation=90)
plt.title('Major Depression Percentages Across States')
```

Out[31]:
```
Text(0.5, 1.0, 'Major Depression Percentages Across States')
```


Major Depression Percentages Across States
Major_Depression%
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
New Jersey
California
Alaska
Louisiana
Texas
Maryland
Hawaii
Arizona
Florida
Illinois
South Dakota
Minnesota
New Hampshire
Alabama
New York
Pennsylvania
Michigan
Iowa
Virginia
Mississippi
Delaware
North Dakota
South Carolina
North Carolina
Georgia
Washington
Massachusetts
Nebraska
Oklahoma
Idaho
Vermont
Kansas
New Mexico
Arkansas
Tennessee
Wisconsin
Kentucky
Missouri
Indiana
Ohio
Connecticut
Maine
District of Columbia
Utah
Colorado
Wyoming
Montana
Oregon
Nevada
West Virginia
Rhode Island
CHSI_State_Name_x

In [32]:
```
plot_cols=['No_HS_Diploma%', 'Unemployed%','Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%']
i=4
plot1=Demo_VPEH_df[plot_cols].groupby(Demo_VPEH_df['CHSI_State_Name_x']).mean().sort_values(by=plot_cols[i])
sns.set(rc={'figure.figsize':(11.7,8.27)})
name='chart'+ str(i)
name = sns.barplot(x=plot1.index, y=plot1[plot_cols[i]], data=plot1)
plt.xticks(rotation=90)
plt.title('Recent Drug Use (Within last one month)')
```

Out[32]:
```
Text(0.5, 1.0, 'Recent Drug Use (Within last one month)')
```


Recent Drug Use (Within last one month)
Recent_Drug_Use%
0.00
0.02
0.04
0.06
0.08
North Dakota
South Dakota
Iowa
Nebraska
Utah
West Virginia
Indiana
Texas
Idaho
Oklahoma
Wyoming
Mississippi
Missouri
Virginia
South Carolina
Alabama
New Jersey
Kansas
Ohio
Georgia
Montana
Tennessee
Pennsylvania
Florida
Wisconsin
Maryland
Louisiana
Minnesota
Kentucky
Arizona
Arkansas
New York
Illinois
New Mexico
Hawaii
Nevada
Michigan
Delaware
Connecticut
North Carolina
Washington
California
District of Columbia
New Hampshire
Rhode Island
Maine
Oregon
Alaska
Colorado
Vermont
Massachusetts
CHSI_State_Name_x

In [33]:
```
##some data transformations for better visualisations
Demo_VPEH_df['Poverty_log']=np.log(Demo_VPEH_df['Poverty'])
Demo_VPEH_df['Unemployed%_log']=np.log(Demo_VPEH_df['Unemployed%'])
Demo_VPEH_df['Major_Depression%_log']=np.log(Demo_VPEH_df['Major_Depression%'])
Demo_VPEH_df['Major_Depression_log']=np.log(Demo_VPEH_df['Major_Depression'])
Demo_VPEH_df['Population_Density_log']=np.log(Demo_VPEH_df['Population_Density'])
Demo_VPEH_df['Toxic_Chem_log']=np.log(Demo_VPEH_df['Toxic_Chem'])
Demo_VPEH_df['Population_Size_log']=np.log(Demo_VPEH_df['Population_Size'])
Demo_VPEH_df['Ecol_Rpt_log']=np.log(Demo_VPEH_df['Ecol_Rpt'])
Demo_VPEH_df['Ecol_Salm_Shig']=Demo_VPEH_df['Ecol_Rpt']+Demo_VPEH_df['Salm_Rpt']+Demo_VPEH_df['Shig_Rpt']
```

```
C:\ProgramData\Anaconda3\envs\cs418env\lib\site-packages\pandas\core\series.py:856: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

Observations:

1. There is a positive correlation of poverty and unemployment.
2. Negative correlation of poverty and population density.
3. No relationship between poverty and depression observed.
4. Strong positive correlation between population density and depression.
5. Positive correlation between poverty and No High School Diploma Percentages.
6. Population size and E.Colli, Salmonella and Shigella Correlated (Hygiene Related Diseases)
7. Races Vs Poverty Trends

In [34]:

```
##Poverty & Unemployment
g =sns.FacetGrid(Demo_VPEH_df, col='CHSI_State_Name_x',col_wrap=4)
g =(g.map(plt.scatter, "Poverty", "Unemployed%_log", edgecolor="w").add_legend())
print(" There is a positive correlation of poverty and Unemployment (as expected) in most states")
```

```
There is a positive correlation of poverty and Unemployment (as expected) in most states
```

In [35]:

```
##Poverty & PopulationDensity
g =sns.FacetGrid(Demo_VPEH_df, col='CHSI_State_Name_x',col_wrap=4)
g =(g.map(plt.scatter, "Poverty", 'Population_Density_log').add_legend())
print("There is a negative Correlation between Poverty and Population Density. \n\n More dense areas seem to have les
```

```
There is a negative Correlation between Poverty and Population Density.

 More dense areas seem to have less poverty -- maybe because more job opportunities?
```

In [36]:
```
##Poverty & Major Depression
g =sns.FacetGrid(Demo_VPEH_df, col='CHSI_State_Name_x',col_wrap=4)
g =(g.map(plt.scatter, "Major_Depression_log", 'Poverty').add_legend())
print("There's no relationship between poverty and depression")
```

```
There's no relationship between poverty and depression
```

In [37]:
```
##Major Depression & Population Density
g =sns.FacetGrid(Demo_VPEH_df, col='CHSI_State_Name_x',col_wrap=4)
g =(g.map(plt.scatter, 'Major_Depression_log', 'Population_Density_log').add_legend())
print("strong correlation of major depression and Population density")
```

```
strong correlation of major depression and Population density
```

In [38]:

```
###Poverty & No High School Diploma Numbers
g =sns.FacetGrid(Demo_VPEH_df, col='CHSI_State_Name_x',col_wrap=5)
g =(g.map(plt.scatter, 'Poverty', 'No_HS_Diploma%').add_legend())
print("Positive correlation between poverty and no hs diploma")
```

```
Positive correlation between poverty and no hs diploma
```

In [39]:
```
ToCorr=Demo_VPEH_df[['Population_Size', 'Population_Density', 'Poverty', 'Age_19_Under',
       'Age_19_64', 'Age_65_84', 'Age_85_and_Over', 'White', 'Black',
       'Native_American', 'Asian', 'Hispanic','No_HS_Diploma%', 'Unemployed%',
       'Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%',
       'Ecol_Rpt', 'Ecol_Rpt_Ind', 'Ecol_Exp', 'Salm_Rpt',
       'Salm_Rpt_Ind', 'Salm_Exp', 'Shig_Rpt', 'Shig_Rpt_Ind', 'Shig_Exp',
       'Toxic_Chem']]

CorrelationTable=ToCorr.corr()
CorrelationTable=CorrelationTable[(CorrelationTable>0.5) | (CorrelationTable<-0.5)]
CorrelationTable=CorrelationTable.reset_index()
CorrelationTable=pd.melt(CorrelationTable, id_vars=['index'])
CorrelationTable=CorrelationTable.dropna()
CorrelationTable=CorrelationTable[CorrelationTable['value']!=1]
print("Correlated Attributes, Print more to see")
CorrelationTable[:5]
```

```
Correlated Attributes, Print more to see
```

Out[39]:

	index	variable	value
19	Ecol_Exp	Population_Size	0.955662
20	Salm_Rpt	Population_Size	0.660316
22	Salm_Exp	Population_Size	0.976794
23	Shig_Rpt	Population_Size	0.506793
25	Shig_Exp	Population_Size	0.897634

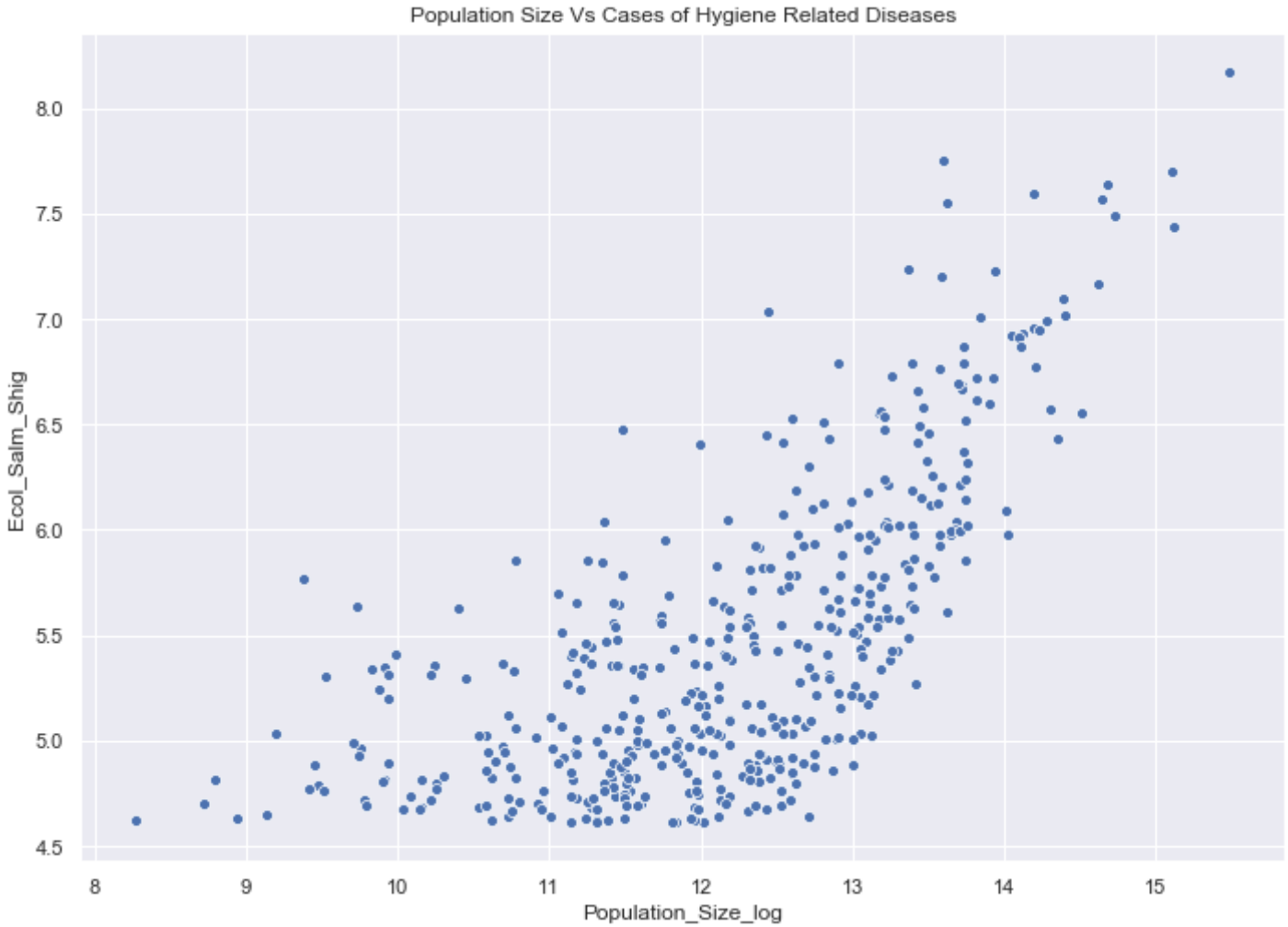
In [40]:
```
Demo_VPEH_dfTemp=Demo_VPEH_df[Demo_VPEH_df['Ecol_Salm_Shig']>100]
Demo_VPEH_dfTemp['Ecol_Salm_Shig']=np.log(Demo_VPEH_dfTemp['Ecol_Salm_Shig'])
Plot1=sns.scatterplot(x='No_HS_Diploma%', y='Poverty', data=Demo_VPEH_dfTemp)
plt.title('No High School Diploma Percentages & Poverty')
```

```
C:\ProgramData\Anaconda3\envs\cs418env\lib\site-packages\ipykernel_launcher.py: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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

Out[40]:
```
Text(0.5, 1.0, 'No High School Diploma Percentages & Poverty')
```


No High School Diploma Percentages & Poverty
Poverty
No_HS_Diploma%
35
30
25
20
15
10
5
0.05
0.10
0.15
0.20
0.25

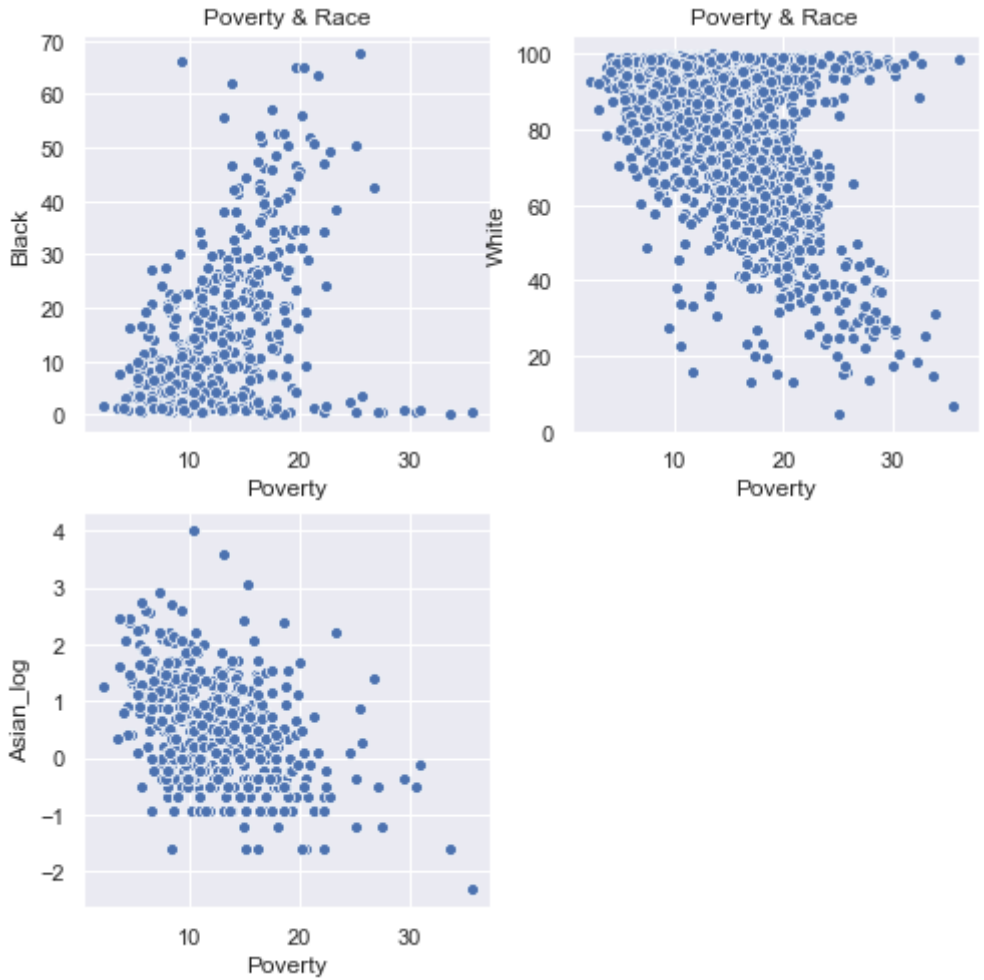
In [41]:
```
Plot2=sns.scatterplot(x='Population_Size_log', y='Ecol_Salm_Shig', data=Demo_VPEH_dfTemp)
plt.title('Population Size Vs Cases of Hygiene Related Diseases')
plt.rcParams["figure.figsize"] = (5,5)
```



In [42]:
```
plt.rcParams["figure.figsize"] = (8,8)
fig = plt.figure()
gs = fig.add_gridspec(2, 2)
ax1 = fig.add_subplot(gs[0, 0])
ax1=sns.scatterplot(x='Poverty', y='Black', data=Demo_VPEH_dfTemp)
plt.title('Poverty & Race')

ax2 = fig.add_subplot(gs[1, 0])
Demo_VPEH_dfTemp['Asian_log']=np.log(Demo_VPEH_dfTemp['Asian'])
ax2=sns.scatterplot(x='Poverty', y='Asian_log', data=Demo_VPEH_dfTemp)
#plt.title('Poverty & Race: Asian')

ax3 = fig.add_subplot(gs[0, 1])
ax3=sns.scatterplot(x='Poverty', y='White', data=Demo_VPEH_df)
plt.title('Poverty & Race')
```

```
C:\ProgramData\Anaconda3\envs\cs418env\lib\site-packages\ipykernel_launcher.py:9: 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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  if __name__ == '__main__':
```

Out[42]:
```
Text(0.5, 1.0, 'Poverty & Race')
```



Preventive Services Use

In [43]:
```
Demo_VPEH_df_tojoin=Demo_VPEH_df[['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name_x',
       'CHSI_State_Name_x', 'CHSI_State_Abbr_x', 'Strata_ID_Number_x',
       'Population_Size', 'Population_Density', 'Poverty', 'Age_19_Under',
       'Age_19_64', 'Age_65_84', 'Age_85_and_Over', 'White', 'Black',
       'Native_American', 'Asian', 'Hispanic','No_HS_Diploma', 'Unemployed',
       'Sev_Work_Disabled', 'Major_Depression',
       'Recent_Drug_Use', 'Ecol_Rpt',  'Salm_Rpt', 'Shig_Rpt','Toxic_Chem', 'No_HS_Diploma%', 'Unemployed%',
       'Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%',
       'Poverty_log']]
df_PSU=pd.read_csv('PREVENTIVESERVICESUSE.csv')
Useful=['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name',
       'CHSI_State_Name', 'CHSI_State_Abbr', 'Strata_ID_Number', 'FluB_Rpt', 'HepA_Rpt','HepB_Rpt','Meas_Rpt',  'Pert
       'Pap_Smear','Mammogram', 'Proctoscopy',  'Pneumo_Vax', 'Flu_Vac']
df_PSU=df_PSU[Useful]
HandleNanCols=['FluB_Rpt',
       'HepA_Rpt', 'HepB_Rpt', 'Meas_Rpt', 'Pert_Rpt', 'CRS_Rpt',
       'Syphilis_Rpt', 'Pap_Smear', 'Mammogram', 'Proctoscopy', 'Pneumo_Vax',
       'Flu_Vac']
df_PSU[df_PSU[HandleNanCols]<0]=np.nan
PSU_Demo_VPEH_df=df_PSU.merge(Demo_VPEH_df_tojoin, on=['State_FIPS_Code', 'County_FIPS_Code'], how='left', indicator=
PSU_Demo_VPEH_df
```

Out[43]:

	State_FIPS_Code	County_FIPS_Code	CHSI_County_Name	CHSI_State_Name	CHSI_State_Abbr	Strata_ID_Number	FluB_Rpt	HepA_Rp
0	1	1	Autauga	Alabama	AL	29	0.0	1.(
1	1	3	Baldwin	Alabama	AL	16	0.0	2.(
2	1	5	Barbour	Alabama	AL	51	0.0	2.(
3	1	7	Bibb	Alabama	AL	42	0.0	2.(
4	1	9	Blount	Alabama	AL	28	2.0	3.(
...	...	...	...	...	...	...	...	...
3136	56	37	Sweetwater	Wyoming	WY	77	0.0	1.(
3137	56	39	Teton	Wyoming	WY	78	0.0	9.(
3138	56	41	Uinta	Wyoming	WY	38	0.0	23.(
3139	56	43	Washakie	Wyoming	WY	82	0.0	1.(

	State_FIPS_Code	County_FIPS_Code	CHSI_County_Name	CHSI_State_Name	CHSI_State_Abbr	Strata_ID_Number	FluB_Rpt	HepA_Rp
3140	56	45	Weston	Wyoming	WY	78	0.0	0.

3141 rows × 50 columns

In [44]:
```
for i in range(0,len(HandleNanCols)):
    stringname=HandleNanCols[i]
    stringnameo=stringname+'%'
    PSU_Demo_VPEH_df[stringnameo]=PSU_Demo_VPEH_df[stringname]/PSU_Demo_VPEH_df['Population_Size']

def CorrelationTable(ToCorr):
    CorrelationTable=ToCorr.corr()
    CorrelationTable=CorrelationTable[(CorrelationTable>0.5) | (CorrelationTable<-0.5)]
    CorrelationTable=CorrelationTable.reset_index()
    CorrelationTable=pd.melt(CorrelationTable, id_vars=['index'])
    CorrelationTable=CorrelationTable.dropna()
    CorrelationTable=CorrelationTable[CorrelationTable['value']!=1]
    return(CorrelationTable)

Table=CorrelationTable(PSU_Demo_VPEH_df)
Table
```

Out[44]:

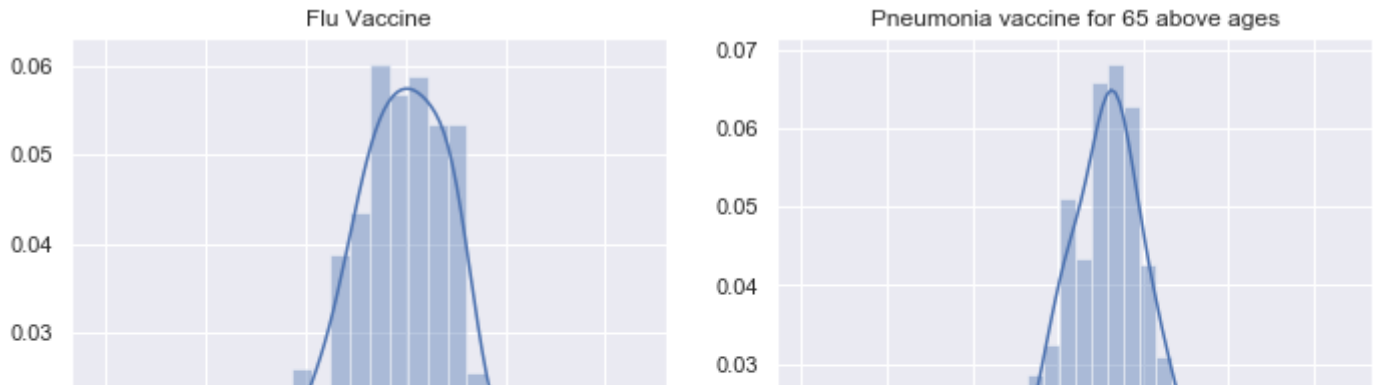
	index	variable	value
122	Proctoscopy	Strata_ID_Number	-0.506233
160	Pap_Smear%	Strata_ID_Number	0.661957
161	Mammogram%	Strata_ID_Number	0.657951
162	Proctoscopy%	Strata_ID_Number	0.728391
163	Pneumo_Vax%	Strata_ID_Number	0.699381
...	...	...	...
2985	Strata_ID_Number_x	Flu_Vac%	0.686189
3020	Pap_Smear%	Flu_Vac%	0.994975
3021	Mammogram%	Flu_Vac%	0.994759
3022	Proctoscopy%	Flu_Vac%	0.969355
3023	Pneumo_Vax%	Flu_Vac%	0.992826

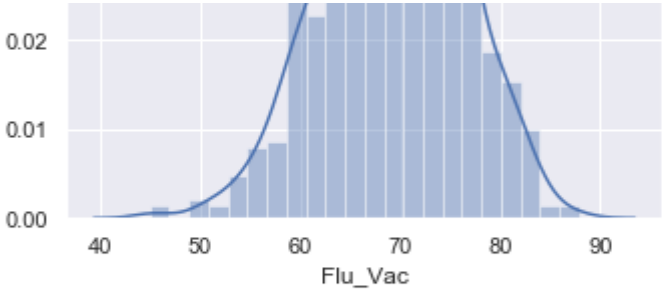
192 rows × 3 columns

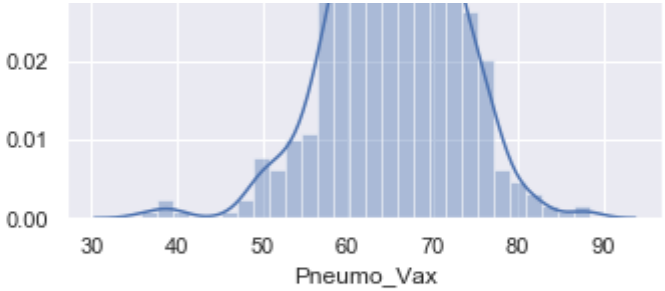
By this table we find the most correlated columns and study those

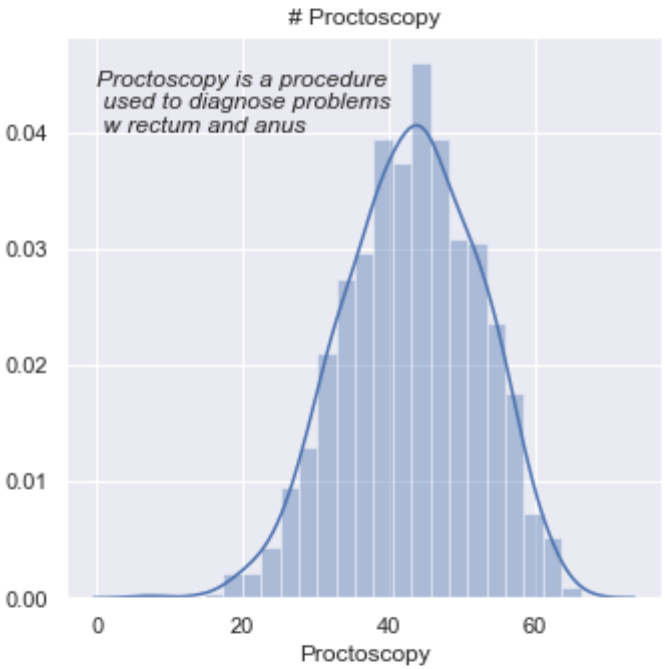
In [47]:

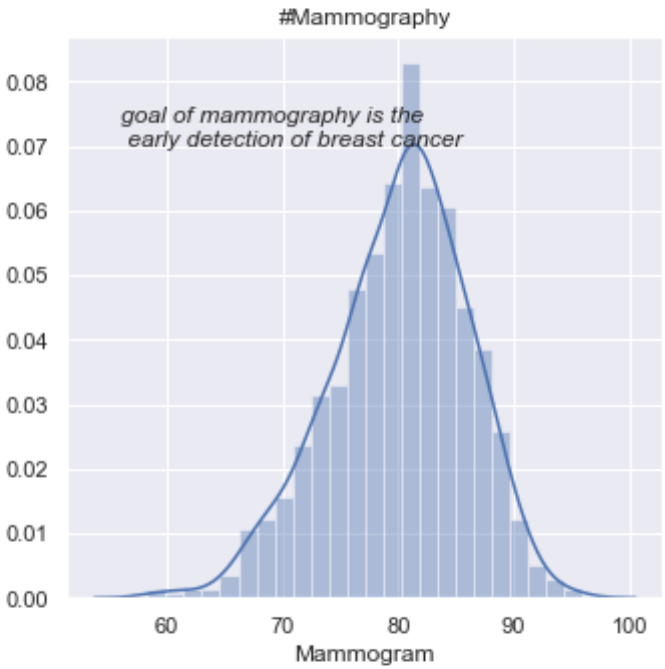
```
Histogramsdf=PSU_Demo_VPEH_df.copy()
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

ax1 = plt.subplot2grid((3, 3), (0, 0))
ax1=sns.distplot((Histogramsdf['Flu_Vac'].dropna()))
ax1.set_title("Flu Vaccine")

ax2 = plt.subplot2grid((3, 3), (0, 1))
ax2=sns.distplot((Histogramsdf['Pneumo_Vax'].dropna()))
ax2.set_title("Pneumonia vaccine for 65 above ages")

ax3 = plt.subplot2grid((3, 3), (1, 0))
ax3=sns.distplot((Histogramsdf['Proctoscopy'].dropna()))
ax3.set_title("# Proctoscopy")
ax3.text(0, 0.04, 'Proctoscopy is a procedure \n used to diagnose problems\n w rectum and anus', style='italic')

ax4 = plt.subplot2grid((3, 3), (1, 1))
ax4=sns.distplot((Histogramsdf['Mammogram'].dropna()))
ax4.set_title("#Mammography")
ax4.text(56, 0.07, 'goal of mammography is the \n early detection of breast cancer', style='italic')

ax5 = plt.subplot2grid((3, 3), (2, 0))
ax5=sns.distplot((Histogramsdf['Pap_Smear'].dropna()))
ax5.set_title("# Pap Smear")
ax5.text(56, 0.07, 'cervical screening used to detect \npotentially precancerous and cancerous \n processes in the ce
plt.tight_layout(pad=2, w_pad=2, h_pad=2.0)
```

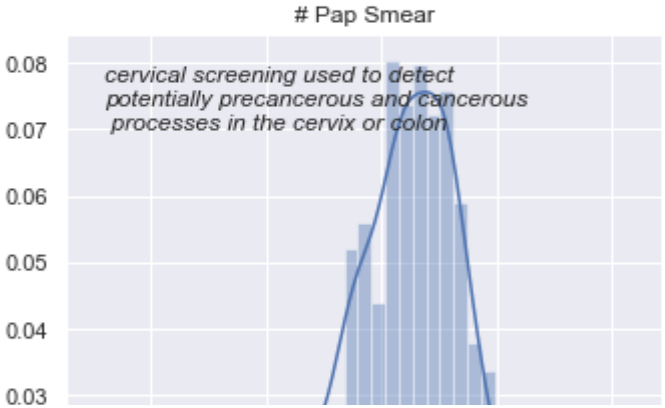


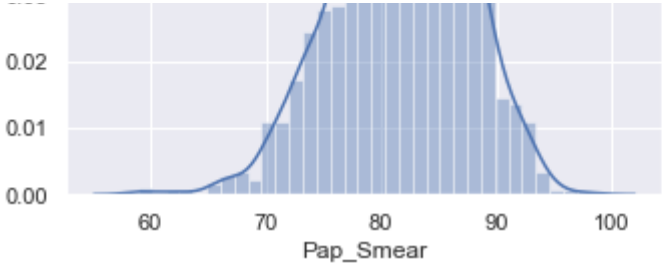

0.02
0.01
0.00
40
50
60
70
80
90
Flu_Vac


0.02
0.01
0.00
30
40
50
60
70
80
90
Pneumo_Vax


Proctoscopy
Proctoscopy is a procedure used to diagnose problems w rectum and anus
0.04
0.03
0.02
0.01
0.00
0
20
40
60
Proctoscopy

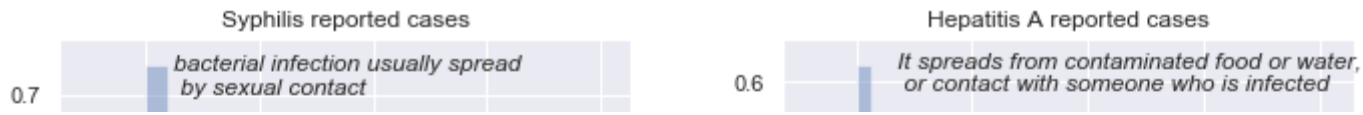

#Mammography
goal of mammography is the early detection of breast cancer
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
60
70
80
90
100
Mammogram

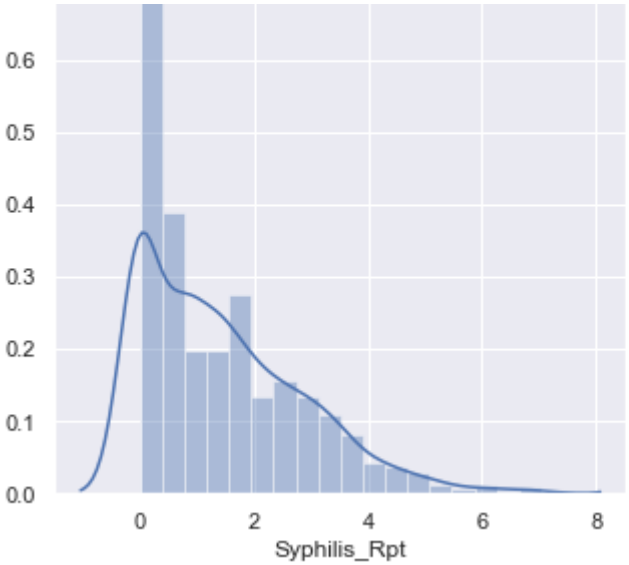

Pap Smear
cervical screening used to detect potentially precancerous and cancerous processes in the cervix or colon
0.08
0.07
0.06
0.05
0.04
0.03

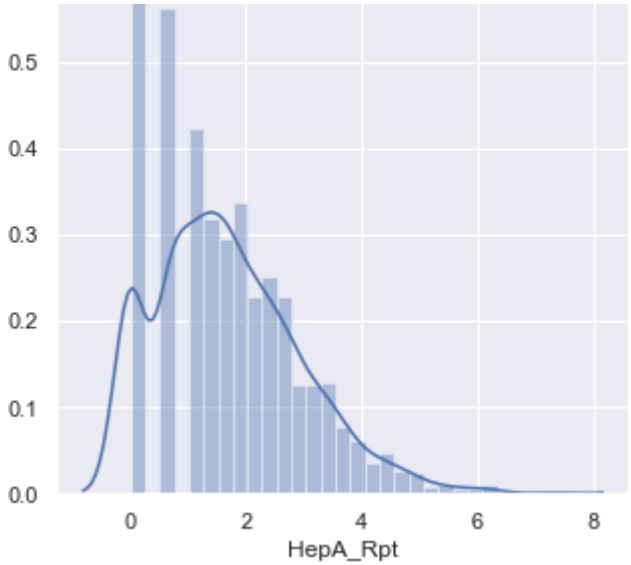

0.02
0.01
0.00
60
70
80
90
100
Pap_Smear

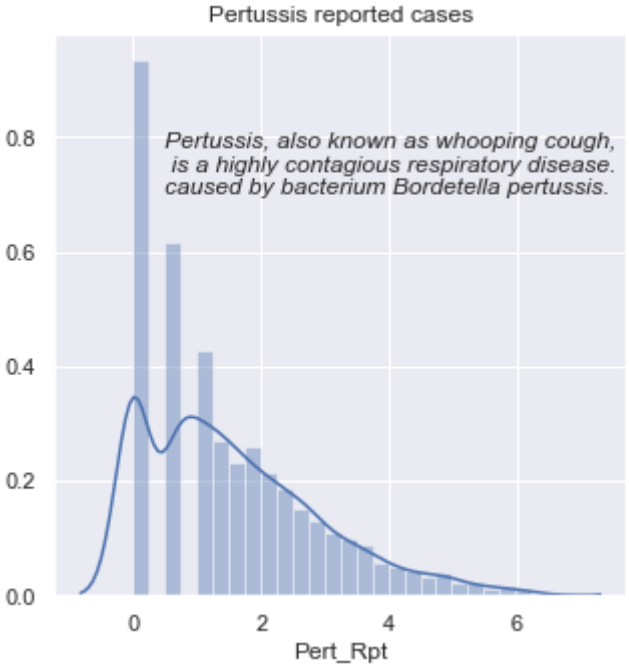
In [48]:

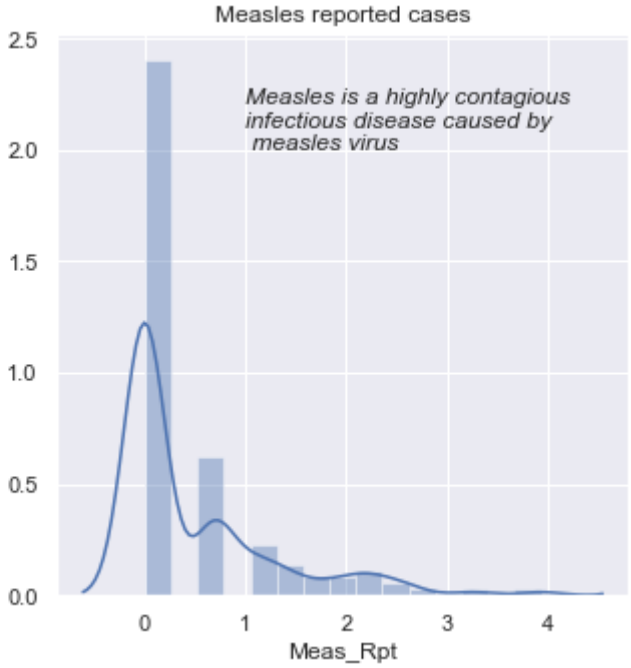
```
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

ax6 = plt.subplot2grid((3, 3), (0, 0))
ax6=sns.distplot(np.log(Histogramsdf[Histogramsdf['Syphilis_Rpt']>0]['Syphilis_Rpt']))
ax6.set_title("Syphilis reported cases")
ax6.text(0.5, 0.7, 'bacterial infection usually spread\n by sexual contact', style='italic')

ax7 = plt.subplot2grid((3, 3), (0, 1), colspan=1)
ax7=sns.distplot(np.log(Histogramsdf[Histogramsdf['HepA_Rpt']>0]['HepA_Rpt']))
ax7.set_title("Hepatitis A reported cases")
ax7.text(0.7, 0.59, 'It spreads from contaminated food or water,\n or contact with someone who is infected', style='i

ax8 = plt.subplot2grid((3, 3), (1, 0), colspan=1)
ax8=sns.distplot(np.log(Histogramsdf[Histogramsdf['Pert_Rpt']>0]['Pert_Rpt']))
ax8.set_title("Pertussis reported cases")
ax8.text(0.5, 0.7, 'Pertussis, also known as whooping cough,\n is a highly contagious respiratory disease.\ncaused by

ax9 = plt.subplot2grid((3, 3), (1, 1), colspan=1)
ax9=sns.distplot(np.log(Histogramsdf[Histogramsdf['Meas_Rpt']>0]['Meas_Rpt']))
ax9.set_title("Measles reported cases")
ax9.text(1, 2, 'Measles is a highly contagious \ninfectious disease caused by \n measles virus', style='italic')

ax10 = plt.subplot2grid((3, 3), (2, 0), colspan=1)
ax10=sns.distplot(np.log(Histogramsdf[Histogramsdf['HepB_Rpt']>0]['HepB_Rpt']))
ax10.set_title("Hepatitis B reported cases")
ax10.text(0.4, 0.9, 'This disease is most commonly spread \nby exposure to infected body fluids.', style='italic')


ax11 = plt.subplot2grid((3, 3), (2, 1), colspan=1)
ax11=sns.distplot(np.log(Histogramsdf[Histogramsdf['FluB_Rpt']>0]['FluB_Rpt']))
ax11.set_title("Haemophilus Influenzae B  reported cases")
ax11.text(0.8, 1.5, 'infection caused by bacteria', style='italic')

plt.tight_layout(pad=2, w_pad=2, h_pad=2.0)
```

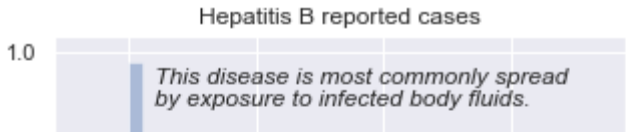


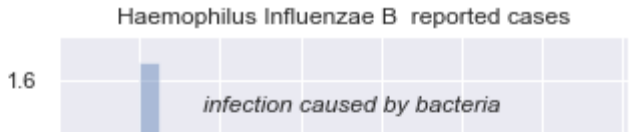

0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
Syphilis_Rpt

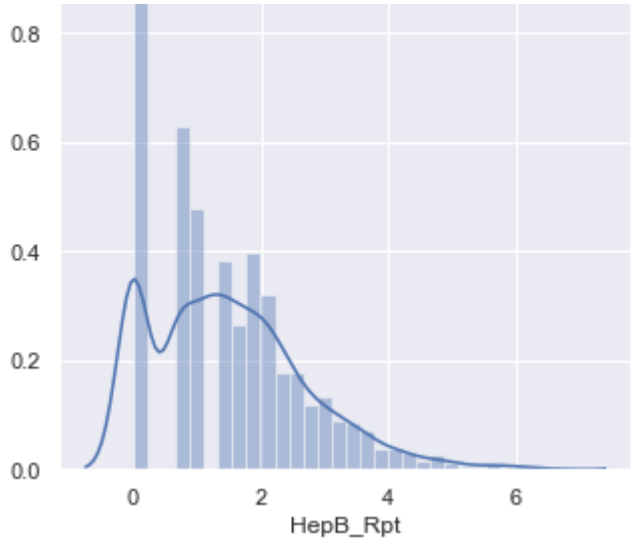

0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
HepA_Rpt

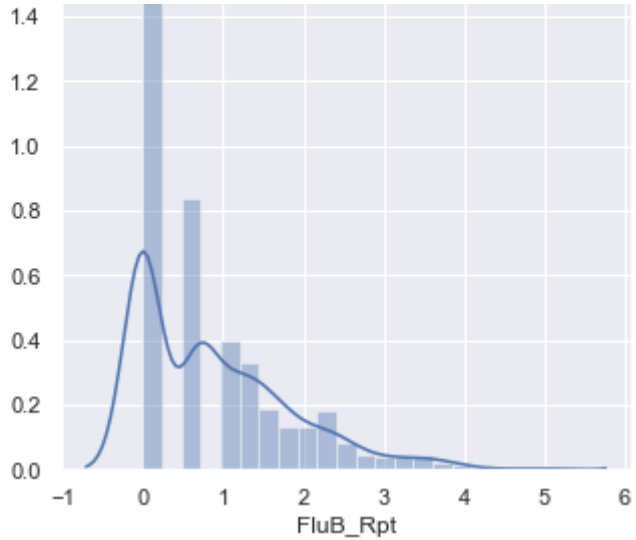

Pertussis reported cases
Pertussis, also known as whooping cough,
is a highly contagious respiratory disease.
caused by bacterium Bordetella pertussis.
0.8
0.6
0.4
0.2
0.0
0
2
4
6
Pert_Rpt


Measles reported cases
Measles is a highly contagious
infectious disease caused by
measles virus
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
Meas_Rpt


Hepatitis B reported cases
1.0
This disease is most commonly spread
by exposure to infected body fluids.

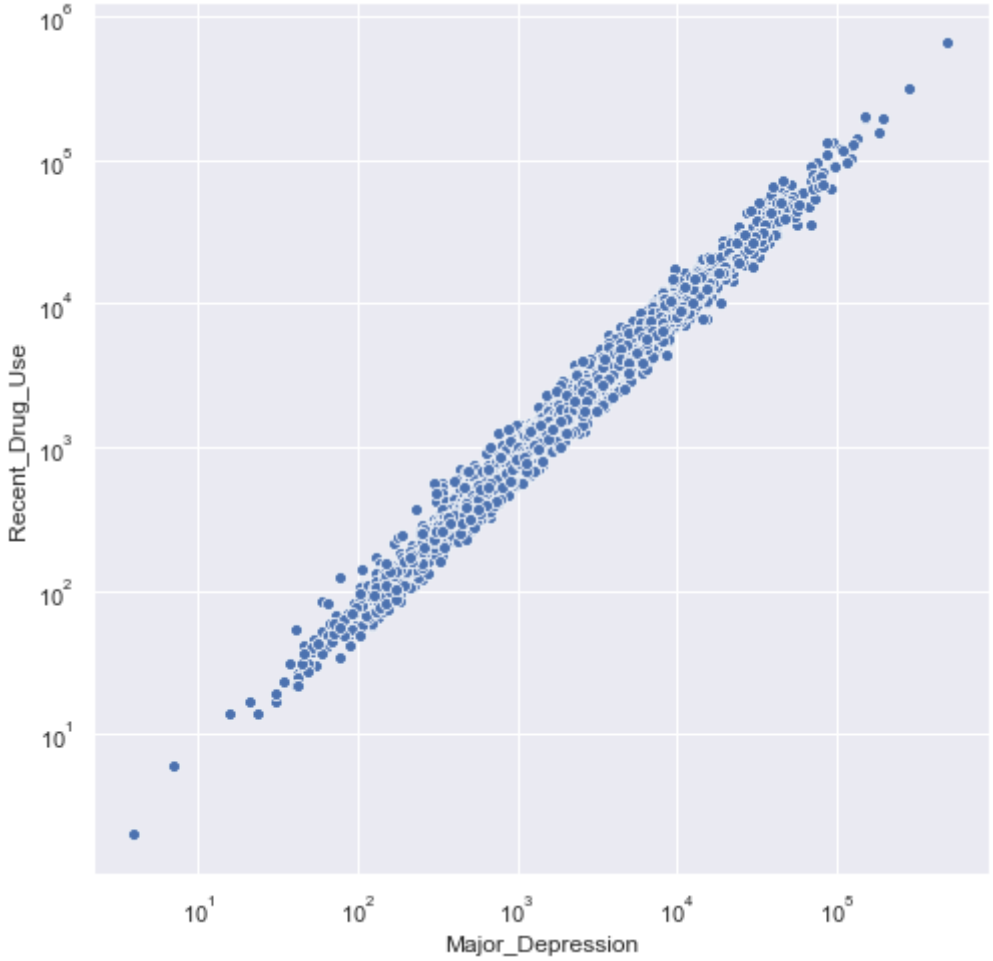

Haemophilus Influenzae B reported cases
1.6
infection caused by bacteria


0.8
0.6
0.4
0.2
0.0
0
2
4
6
HepB_Rpt


1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1
0
1
2
3
4
5
6
FluB_Rpt

In [49]:
```
Plot1=sns.scatterplot(x='Major_Depression', y='Recent_Drug_Use', data=PSU_Demo_VPEH_df)
Plot1.set_yscale('log')
Plot1.set_xscale('log')
print("Strong Correlation of Depression & Drug Usage Causation or Correlation?")
```

```
Strong Correlation of Depression & Drug Usage Causation or Correlation?
```



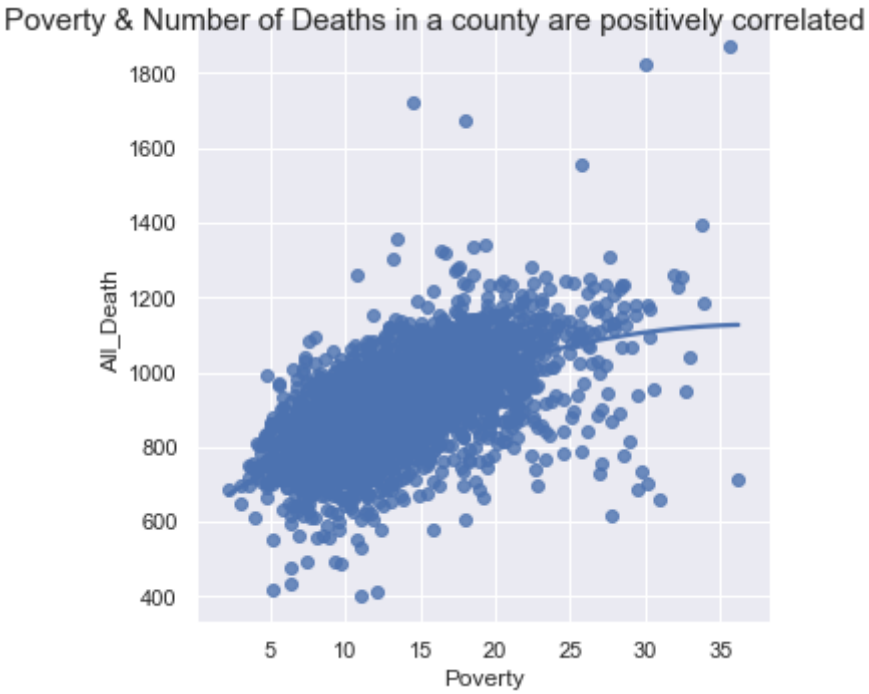
Summary Measures of Health

In [52]:
```
df_SMOH=pd.read_csv('SUMMARYMEASURESOFHEALTH.csv')
ForUse=['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name','CHSI_State_Name', 'CHSI_State_Abbr', 'Strata_ID_Nu
UsefulCols=['ALE', 'All_Death', 'Health_Status', 'Unhealthy_Days']
df_SMOH[df_SMOH[UsefulCols]<0]=np.nan
df_SMOH=df_SMOH[ForUse+UsefulCols]
PSU_Demo_VPEH_df=PSU_Demo_VPEH_df[['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name', 'CHSI_State_Name',
 'CHSI_State_Abbr', 'Strata_ID_Number', 'FluB_Rpt', 'HepA_Rpt', 'HepB_Rpt',
 'Meas_Rpt', 'Pert_Rpt', 'CRS_Rpt',
 'Syphilis_Rpt', 'Pap_Smear', 'Mammogram', 'Proctoscopy', 'Pneumo_Vax',
 'Flu_Vac', 'Population_Size',
 'Population_Density', 'Poverty', 'Age_19_Under', 'Age_19_64',
 'Age_65_84', 'Age_85_and_Over', 'White', 'Black', 'Native_American',
 'Asian', 'Hispanic', 'No_HS_Diploma', 'Unemployed', 'Sev_Work_Disabled',
 'Major_Depression', 'Recent_Drug_Use', 'Ecol_Rpt', 'Salm_Rpt',
 'Shig_Rpt', 'Toxic_Chem', 'No_HS_Diploma%', 'Unemployed%',
 'Sev_Work_Disabled%', 'Major_Depression%', 'Recent_Drug_Use%',
 'FluB_Rpt%', 'HepA_Rpt%', 'HepB_Rpt%',
 'Meas_Rpt%', 'Pert_Rpt%', 'CRS_Rpt%', 'Syphilis_Rpt%', 'Pap_Smear%',
 'Mammogram%', 'Proctoscopy%', 'Pneumo_Vax%', 'Flu_Vac%']]
df_SMOH.columns
PSU_Demo_VPEH_SMOH_df=PSU_Demo_VPEH_df.merge(df_SMOH, on=['State_FIPS_Code', 'County_FIPS_Code'], how='left', indicat
PSU_Demo_VPEH_SMOH_df.columns
```

Out[52]:
```
Index(['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name_x',
       'CHSI_State_Name_x', 'CHSI_State_Abbr_x', 'Strata_ID_Number_x',
       'FluB_Rpt', 'HepA_Rpt', 'HepB_Rpt', 'Meas_Rpt', 'Pert_Rpt', 'CRS_Rpt',
       'Syphilis_Rpt', 'Pap_Smear', 'Mammogram', 'Proctoscopy', 'Pneumo_Vax',
       'Flu_Vac', 'Population_Size', 'Population_Density', 'Poverty',
       'Age_19_Under', 'Age_19_64', 'Age_65_84', 'Age_85_and_Over', 'White',
       'Black', 'Native_American', 'Asian', 'Hispanic', 'No_HS_Diploma',
       'Unemployed', 'Sev_Work_Disabled', 'Major_Depression',
       'Recent_Drug_Use', 'Ecol_Rpt', 'Salm_Rpt', 'Shig_Rpt', 'Toxic_Chem',
       'No_HS_Diploma%', 'Unemployed%', 'Sev_Work_Disabled%',
       'Major_Depression%', 'Recent_Drug_Use%', 'FluB_Rpt%', 'HepA_Rpt%',
       'HepB_Rpt%', 'Meas_Rpt%', 'Pert_Rpt%', 'CRS_Rpt%', 'Syphilis_Rpt%',
       'Pap_Smear%', 'Mammogram%', 'Proctoscopy%', 'Pneumo_Vax%', 'Flu_Vac%',
       'CHSI_County_Name_y', 'CHSI_State_Name_y', 'CHSI_State_Abbr_y',
       'Strata_ID_Number_y', 'ALE', 'All_Death', 'Health_Status',
       'Unhealthy_Days', '_merge'],
      dtype='object')
```

In [64]:
```
ax1=sns.lmplot('Poverty', 'All_Death', data=PSU_Demo_VPEH_SMOH_df, ci=None, order=2, truncate=True, palette="Set1")
fig = ax1.fig
fig.suptitle("Poverty & Number of Deaths in a county are positively correlated", fontsize=15)
```

Out[64]:
```
Text(0.5, 0.98, 'Poverty & Number of Deaths in a county are positively correlated')
```



In [66]:
```
ax2=sns.lmplot('Poverty', 'ALE', data=PSU_Demo_VPEH_SMOH_df, ci=None, order=2, truncate=True, palette="Set1")
fig = ax2.fig
fig.suptitle("Poverty & Average Life Expectancy in a county are negatively correlated", fontsize=15)
```

Out[66]:
```
Text(0.5, 0.98, 'Poverty & Average Life Expectancy in a county are negatively correlated')
```