Data Analytics Lab: Assignment - 1
A mathematical essay on linear regression

Nagappan N
Department of Metallurgical and Materials Engineering
IIT Madras
Chennai, India
mm19b040@smail.iitm.ac.in

***Abstract*—In this article, we aim to demonstrate whether or not cancer incidence and mortality are correlated with socioeconomic status. We also try to visually understand how the cancer incidence and mortality rates are dependent on various factors including the geography.**

I. Introduction

The issue is how to deal with cancer, which is one of the major killers in the globe, accounting for 1 in 6 fatalities. [1] In most nations, economic and living standards have increased over time, though frequently in conjunction with unfavourable changes in lifestyle and environment that are important cancer risk factors. [2]

With the help of the linear regression modelling technique, this study will profile and analyse the inter-correlated nature of socioeconomic status and cancer incidence, death rates in order to acquire some important insights.

Understanding whether or not cancer incidence and mortality are actually correlated with socioeconomic position, specifically income, their insurance status, and geographic location, is our main goal from the linear regression model.

In this paper, the section on datasets tells about the various features used in the predictions. It also includes several visual plots for better understanding the problem. The next section is a brief about the linear regression model which is followed by the section which describes how we have used the model for our analysis. Finally, conclusion includes some key inferences that we have made from our study.

II. Datasets

The dataset mainly comprises of three different data. One of them is income related data, one of them is cancer related and the other is insurance based data.

The final dataset given has the following columns:

1) All_poverty - The total number of people below the poverty line.
2) M_Poverty - The number of males who are below the poverty line.
3) F_Poverty - The number of females who are below the poverty line.
4) FIPS - Numbers which uniquely identify geographic areas in the United States.
5) Med_Income - The median income of all the people in that county.
6) M_With - The total number of males with health insurance in that county.
7) M_Without - The total number of males without health insurance in that county.
8) F_With - The total number of females with health insurance in that county.
9) F_With - The total number of females without health insurance in that county.
10) All_With - The total number of people with health insurance in that county.
11) All_With - The total number of people without health insurance in that county.
12) Incidence_Rate - It is the lung cancer incidence rate per 100,000 people.
13) Avg_Ann_Incidence - Average lung cancer incidence rate.
14) Recent Trend - It has been given as rising when 95% confidence interval of average annual percent change is above 0, stable when it includes 0 and falling when it is below zero. This data regarding confidence interva has been taken from incd.csv.
15) Mortality_Rate - It is the lung cancer mortality rate.
16) Avg_Ann_Deaths - It is the average lung cancer mortality.

A. Data Visualization

Firstly, we analyze how the various features are distributed geographically across the United States. For this, we use choropleth map which is a type of statistical map that uses colors to represent the aggregate/average of a feature over that county/state.

We see from figure 1 the distribution of income among the people in each of the states from here we can see how the geographical factors of the location influence the income levels. For example the median income is clearly higher in the urban area states significantly more than that of other counties.

From figures 2 and 4, we see that it is somewhat evident that the average annual deaths due to cancer is higher in the

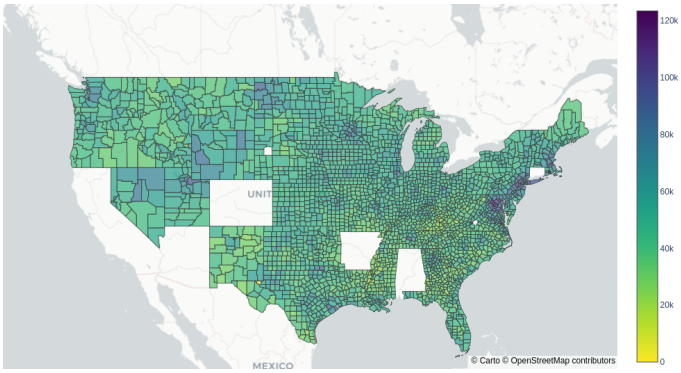


Fig. 1. Choropleth map of median income of people in each county.

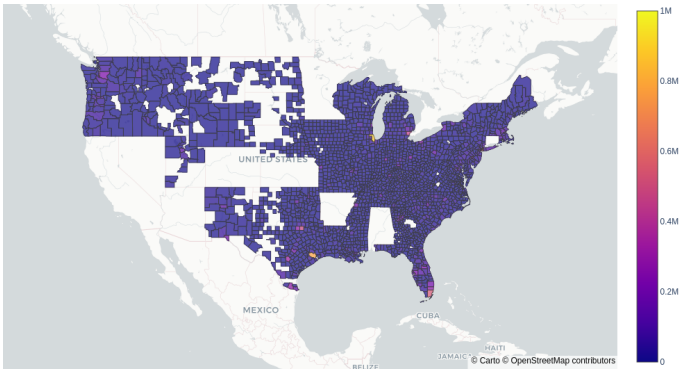


Fig. 2. Choropleth map of number of people below the poverty line in each county.

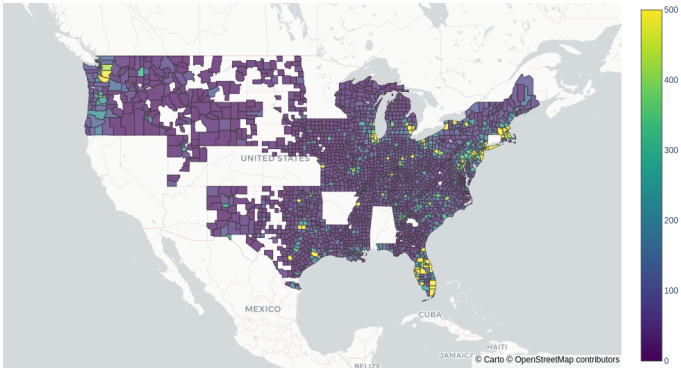


Fig. 3. Choropleth map of average annual incidence of lung cancer in each county.

regions where there are more people below the poverty line.

Next we make some scatter plots to visualize how different features are related to each other.

There are quite a few null values in the dataset. Within the Med_Income column, there is only one null value, which belongs to the area with FIPS 48301, which belongs to the state of Texas. That Med_Income of that particular row or area is filled with the mean value of Med_Income taken for the areas in the state of Texas, which turns out to be 46,745.77.

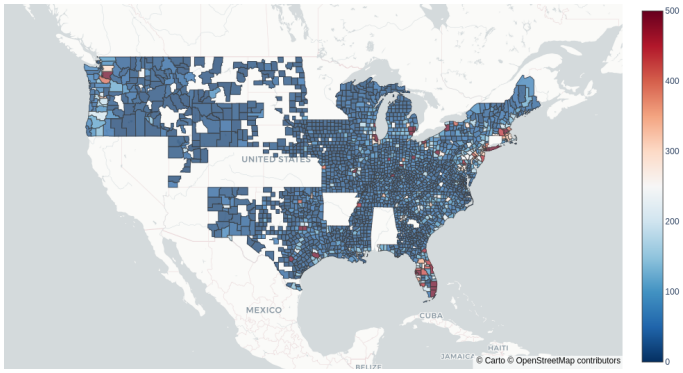


Fig. 4. Choropleth map of average annual death in each county.

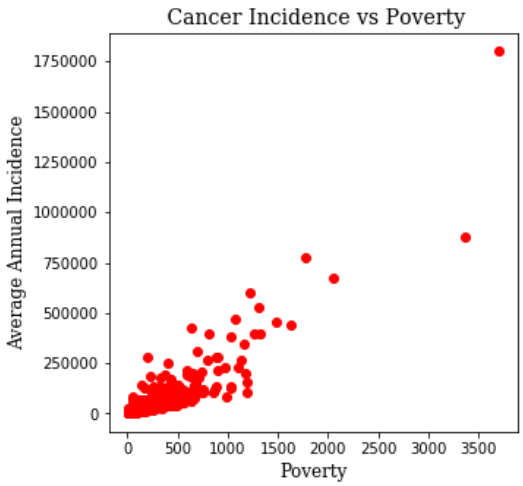


Fig. 5. Scatter plot between average annual cancer incidence and number of people below the poverty line.

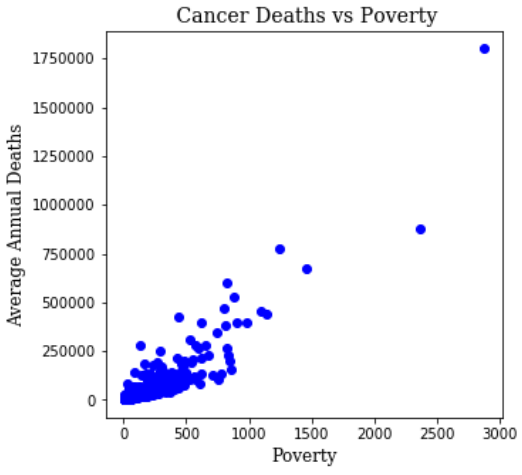


Fig. 6. Scatter plot between average annual deaths due to cancer and number of people below the poverty line.

Within the average annual deaths and average incidence, the '_' value indicates that data not available because of state legislation and regulations which prohibit the release of county

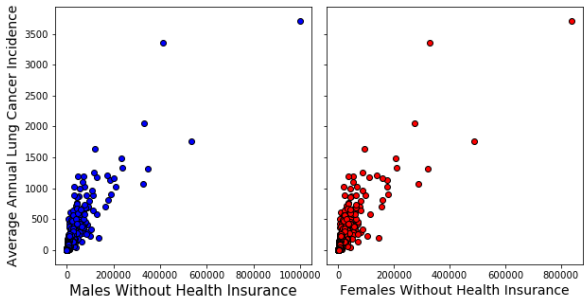


Fig. 7. Scatter plot showing average annual lung cancer incidence versus the number of people without health insurance shown separately for males and females.

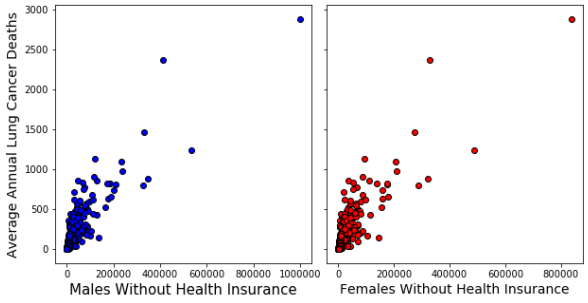


Fig. 8. Scatter plot showing average annual deaths due to cancer versus the number of people without health insurance shown separately for males and females.

level data to outside entities, '__' value indicates that data not available for Nevada and '*' value indicates that data has been suppressed to ensure confidentiality and stability of rate estimates. These values are converted into null or NaN values. The average annual incidence with "3 or fewer" entries are replaced with a mean value of 2. The dataset is then divided into datasets, one with average and annual incidence and the other with average annual deaths. The rows which have null values in these two columns of the respective datasets are dropped.

III. Model: Linear Regression

Linear regression is a modelling approach where we predict a variable called dependent variable based on other independent variables. It is termed as simple linear regression when there is only one independent variable whereas it is called multiple linear regression in case of multiple independent variables. [3] There is another form of linear regression which is called multivariate linear regression which is where there are multiple dependent variables that we are trying to predict. [4]

The dependent variable is assumed to have a linear relationship with all the independent variables. The model is of the form:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + ... + \beta_p x_{ip} + \epsilon_i \quad (1)$$

In the above equation, p is the number of independent variables, y is the dependent variable and ϵ accounts for the noise in the dataset. In vector notation, it is represented as:

$$y_i = {x_i}^T + \epsilon_i \quad (2)$$

Considering that there are n such equations (one for each row in the dataset), we can combine them and write it in the following matrix notation:

$$y = X\beta + \epsilon \quad (3)$$

IV. Modelling

From the given dataset, we have removed the columns related to ethnicity, notedly, Med_Income_White, Med_Income_Black, Med_Income_Nat_Am, Med_Income_Asian and Hispanic as they have very low correlation with the Average annual incidence and deaths, as we can see from the correlation heatmap (figure 9). Similarly, recent trend column is avoided due to low correlation. We have also removed the categorical columns such as FIPS and area name. In the case of categorical column State, we have preprocessed the data using one hot encoding. This type of encoding creates a new binary feature for each possible category and assigns a value of 1 to the feature of each sample that corresponds to its original category.

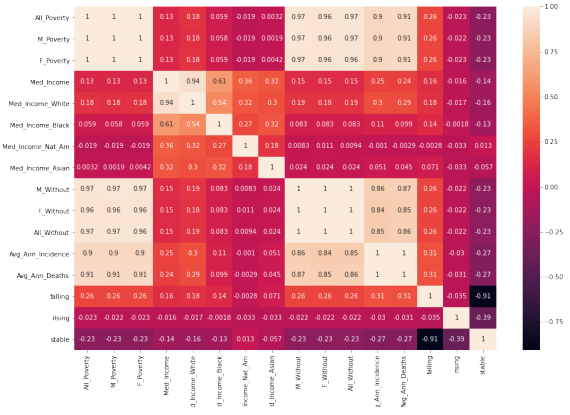


Fig. 9. Heatmap showing the correlation between various features in our dataset.

For the modeling purpose, the independent columns we choose are All_Poverty Med_Income and the categorical one hot encoded values of each state. The target column we have chosen are average annual incidence and average annual deaths, from which the null value rows are removed, which are then predicted separately using linear regression. The filtered dataset contains 2714 data points. Then, we create a linear regression model after importing from the sci-kit

learn[ref] library. The model is then fitted with the training data and the model is made to predict the test data for both incidence and deaths.

To gauge the accuracy and efficiency of the model, we calculate the R squared value for the test target column and predicted values. R-squared is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.

A. Linear regression model for average annual cancer incidence

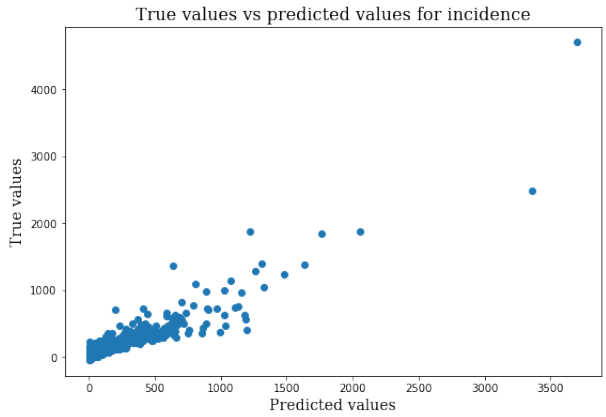


Fig. 10. Scatter plot showing the values of incidence rate predicted by our linear regression model and the original values.

Our model has a R square value of 0.8625 which is a pretty good score.

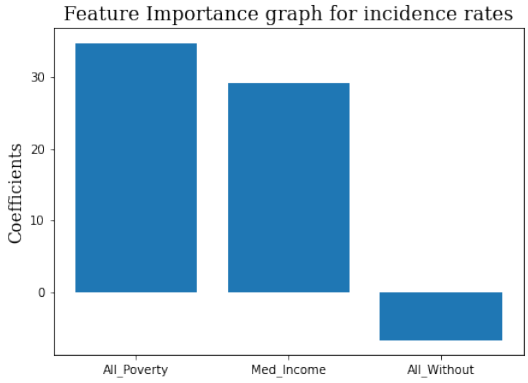


Fig. 11. Feature importance graph showing how much relevant each feature is with respect to prediction of incidence rate in our linear regression model.

We see that as the number of people below the poverty line increases, the incidence of cancer also increases. This might be due to the poor quality of living. Surprisingly, we also see that as median income increases, the incidence of cancer increases. This can be explained by the fact that only people with higher income would have the funds to get tested and screened for cancer. Whether or not a person has insurance should not and does not affect the incidence rate significantly.

B. Linear regression model for average annual deaths due to cancer

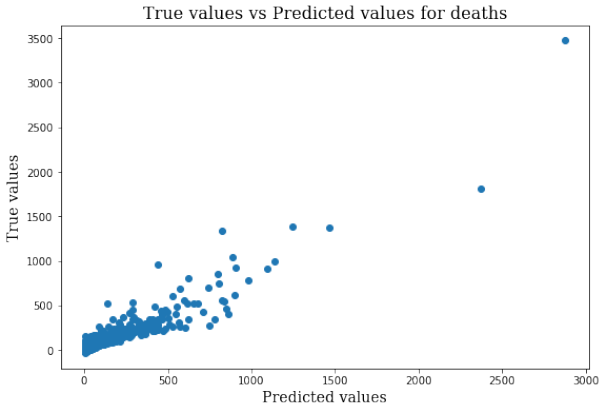


Fig. 12. Scatter plot showing the values of average annual deaths predicted by our linear regression model and the original values.

Our model has a R square value of 0.8741 which is again a good score.

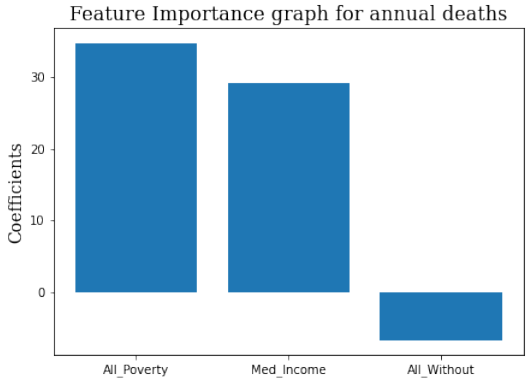


Fig. 13. Feature importance graph showing how much relevant each feature is with respect to prediction of average annual deaths in our linear regression model.

We see that as the number of people below the poverty line increases, the average annual deaths due to cancer also increases. This is because poor people cannot afford the expensive treatments that are there to tackle cancer.

V. Conclusions

We can make the following inferences from our analysis:

1) Firstly, from initial visualizations, we saw that the average annual deaths is higher in the regions where there are more number of people below the poverty line.
2) From the linear regression analysis for predicting the average annual incidence, we see that number of people

below the poverty line and median income are the number major regressors.

3) From the linear regression analysis for predicting the average annual deaths, we again see that number of people below the poverty line and median income are the number major regressors.

REFERENCES

[1] Lortet-Tieulent, J., Georges, D., Bray, F. and Vaccarella, S., 2020. Profiling global cancer incidence and mortality by socioeconomic development. International Journal of Cancer, 147(11), pp.3029-3036.

[2] Who.int. 2022. Cancer. [online] Available at: ¡https://www.who.int/news-room/fact-sheets/detail/cancer¿ [Accessed 2 September 2022].

[3] D. A. Freedman, Statistical Models: Theory and Practice. Cambridge University Press, 2009.

[4] A. C. Rencher and W. F. Christensen, Methods of Multivariate Analysis, 3rd ed. John Wiley Sons, 2012. Accessed: Sep. 02, 2022. [Online]. Available: https://books.google.com/books?id=0g-PAuKub3QCpg=PA19

Data Analytics Lab Assignment 1

Name: Nagappan N

Roll Number: MM19B040

First we import the necessary libraries.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import copy as cp
```

In [2]:
```
merged=pd.read_excel('merged_data.xlsx')
```

In [3]:
```
merged=merged.drop('Unnamed: 0',axis=1)
```

In [4]:
```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3134 entries, 0 to 3133
Data columns (total 25 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   State              3134 non-null   object 
 1   AreaName           3134 non-null   object 
 2   All_Poverty        3134 non-null   int64  
 3   M_Poverty          3134 non-null   int64  
 4   F_Poverty          3134 non-null   int64  
 5   FIPS               3134 non-null   int64  
 6   Med_Income         3133 non-null   float64
 7   Med_Income_White   3132 non-null   float64
 8   Med_Income_Black   1924 non-null   float64
 9   Med_Income_Nat_Am  1474 non-null   float64
 10  Med_Income_Asian   1377 non-null   float64
 11  Hispanic           2453 non-null   float64
 12  M_With             3134 non-null   int64  
 13  M_Without          3134 non-null   int64  
 14  F_With             3134 non-null   int64  
 15  F_Without          3134 non-null   int64  
 16  All_With           3134 non-null   int64  
 17  All_Without        3134 non-null   int64  
 18  fips_x             3134 non-null   int64  
 19  Incidence_Rate     3134 non-null   object 
 20  Avg_Ann_Incidence  3134 non-null   object 
 21  recent_trend       3134 non-null   object 
 22  fips_y             3134 non-null   int64  
 23  Mortality_Rate     3134 non-null   object 
 24  Avg_Ann_Deaths     3134 non-null   object 
dtypes: float64(6), int64(12), object(7)
memory usage: 612.2+ KB
```

In [5]:
```
merged.describe()
```

Out[5]:

	All_Poverty	M_Poverty	F_Poverty	FIPS	Med_Income	Med_Income_White	Med_Income_Black	Med_Income_Nat_Am	Med_Income_Asian	Hispanic	M_With	M_Without	F_With	F_Without
count	3.134000e+03	3134.000000	3134.000000	3134.000000	3133.000000	3132.000000	1924.000000	1474.000000	1377.000000	2453.000000	3.134000e+03	3134.000000	3.134000e+03	3134.000000
mean	1.522966e+04	6828.800893	8400.855775	30426.019145	46819.837855	49490.181992	34750.214137	43309.998643	65412.969499	41118.231553	4.158963e+04	6930.955329	4.487357e+04	5968.70102
std	5.457122e+04	24719.078097	29865.855831	15124.491165	12246.380184	12461.508031	18198.952565	23707.973354	34012.807537	16232.191608	1.293894e+05	28686.089548	1.406455e+05	24657.27699
min	1.000000e+01	5.000000	5.000000	1001.000000	19328.000000	19340.000000	2499.000000	2499.000000	2499.000000	2499.000000	3.200000e+01	4.000000	3.300000e+01	4.00000
25%	1.731250e+03	758.750000	957.000000	19001.500000	38826.000000	41393.500000	23747.250000	28895.750000	45974.000000	31563.000000	4.506750e+03	750.000000	4.657500e+03	633.00000
50%	4.294000e+03	1925.000000	2372.000000	29180.000000	45075.000000	47430.000000	30000.000000	39014.000000	60405.000000	38347.000000	1.040450e+04	1763.000000	1.110800e+04	1529.00000
75%	1.034550e+04	4697.500000	5812.500000	45080.500000	52224.000000	54534.500000	40570.250000	53199.250000	78504.000000	47500.000000	2.788775e+04	4407.250000	2.976475e+04	3834.00000
max	1.800265e+06	823612.000000	976653.000000	56045.000000	123453.000000	136311.000000	170195.000000	250001.000000	250001.000000	223750.000000	3.904322e+06	997326.000000	4.230137e+06	837175.00000

	Feature	Definition	Notes
0	State		
1	AreaName		
2	All_Poverty	Both male and female reported below poverty li...	
3	M_Poverty	Males below poverty (Raw)	
4	F_Poverty	Females below poverty (Raw)	
5	FIPS	State + County FIPS (Raw)	
6	Med_Income	Med_Income all enthnicities (Raw)	
7	M_With	Males with health insurance (Raw)	
8	M_Without	Males without health insurance (Raw)	
9	F_With	Females with health insurance (Raw)	
10	F_Without	Females without health insurance (Raw)	
11	All_With	Males and Femaes with health ins. (Raw)	
12	All_Without	Males an Females without health ins (Raw)	
13	Incidence_Rate	Lung cancer incidence rate (per 100,000)	'*' = fewer that 16 reported cases
14	Avg_Ann_Incidence	Average lung cancer incidence rate (Raw)	
15	Recent Trend	Recent trend (incidence)	
16	Mortality_Rate	Lung cancer mortality rate (per 100,000)	'*' = fewer that 16 reported cases
17	Avg_Ann_Deaths	Average lung cancer mortalities (Raw)	

In [6]:
```
merged['Incidence_Rate'].value_counts()
```

Out[6]:
```
*         211
_         192
65.2       19
__         17
68.9       12
         ...
92.8        1
39.7        1
29.5        1
73.2 #      1
123.7       1
Name: Incidence_Rate, Length: 813, dtype: int64
```

In [7]:
```
merged['Mortality_Rate'].value_counts()
```

Out[7]:
```
*        325
48.3      19
51.8      17
56.3      15
54.4      14
         ...
32.3       1
31.4       1
81.3       1
78.5       1
124.9      1
Name: Mortality_Rate, Length: 618, dtype: int64
```

An incidence rate is the number of new cases of a disease divided by the number of persons at risk for the disease.

In [8]:
```
merged[merged['Med_Income'].isnull()]
```

Out[8]:

	State	AreaName	All_Poverty	M_Poverty	F_Poverty	FIPS	Med_Income	Med_Income_White	Med_Income_Black	Med_Income_Nat_Am	...	F_Without	All_With	All_Without	fips_x	Incidence_Rate	Avg_Ann_Incidence	recent_trend
2666	TX	Loving County, Texas	33	24	9	48301	NaN	NaN	NaN	NaN	...	10	86	31	48301	*	3 or fewer	*

1 rows × 25 columns

In [9]:
```
merged.drop(['fips_y','fips_x','Hispanic'],axis=1,inplace=True)
```

We replace all the special characters with null values.

In [10]:
```
merged['Incidence_Rate']=merged['Incidence_Rate'].replace('*',value=np.nan)
merged['Incidence_Rate']=merged['Incidence_Rate'].replace('_',value=np.nan)
merged['Incidence_Rate']=merged['Incidence_Rate'].replace('__',value=np.nan)
merged['Incidence_Rate']=merged['Incidence_Rate'].replace('#','',regex=True)
```

In [11]:
```
merged['Avg_Ann_Incidence']=merged['Avg_Ann_Incidence'].replace('3 or fewer',np.nan)
merged['Avg_Ann_Incidence']=merged['Avg_Ann_Incidence'].replace('_',np.nan)
merged['Avg_Ann_Incidence']=merged['Avg_Ann_Incidence'].replace('__',np.nan)
```

In [12]:
```
merged['Avg_Ann_Deaths']=merged['Avg_Ann_Deaths'].replace('*',np.nan)
merged['Avg_Ann_Deaths']=merged['Avg_Ann_Deaths'].replace('_',np.nan)
merged['Avg_Ann_Deaths']=merged['Avg_Ann_Deaths'].replace('__',np.nan)
```

In [13]:
```
merged['Mortality_Rate']=merged['Mortality_Rate'].replace('*',np.nan)
merged['Mortality_Rate']=merged['Mortality_Rate'].replace('_',np.nan)
merged['Mortality_Rate']=merged['Mortality_Rate'].replace('__',np.nan)
```

In [14]:
```
merged['recent_trend']=merged['recent_trend'].replace('*',np.nan)
merged['recent_trend']=merged['recent_trend'].replace('_',np.nan)
merged['recent_trend']=merged['recent_trend'].replace('__',np.nan)
```

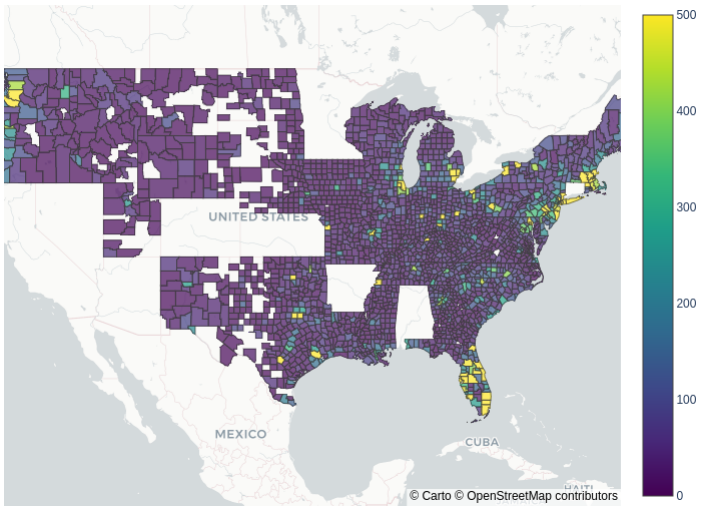
In [15]:
```
merged.info()
data=merged
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3134 entries, 0 to 3133
Data columns (total 22 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   State              3134 non-null   object 
 1   AreaName           3134 non-null   object 
 2   All_Poverty        3134 non-null   int64  
 3   M_Poverty          3134 non-null   int64  
 4   F_Poverty          3134 non-null   int64  
 5   FIPS               3134 non-null   int64  
 6   Med_Income         3133 non-null   float64
 7   Med_Income_White   3132 non-null   float64
 8   Med_Income_Black   1924 non-null   float64
 9   Med_Income_Nat_Am  1474 non-null   float64
 10  Med_Income_Asian   1377 non-null   float64
 11  M_With             3134 non-null   int64  
 12  M_Without          3134 non-null   int64  
 13  F_With             3134 non-null   int64  
 14  F_Without          3134 non-null   int64  
 15  All_With           3134 non-null   int64  
 16  All_Without        3134 non-null   int64  
 17  Incidence_Rate     2714 non-null   object 
 18  Avg_Ann_Incidence  2714 non-null   float64
 19  recent_trend       2667 non-null   object 
 20  Mortality_Rate     2809 non-null   float64
 21  Avg_Ann_Deaths     2809 non-null   float64
dtypes: float64(8), int64(10), object(4)
memory usage: 538.8+ KB
```

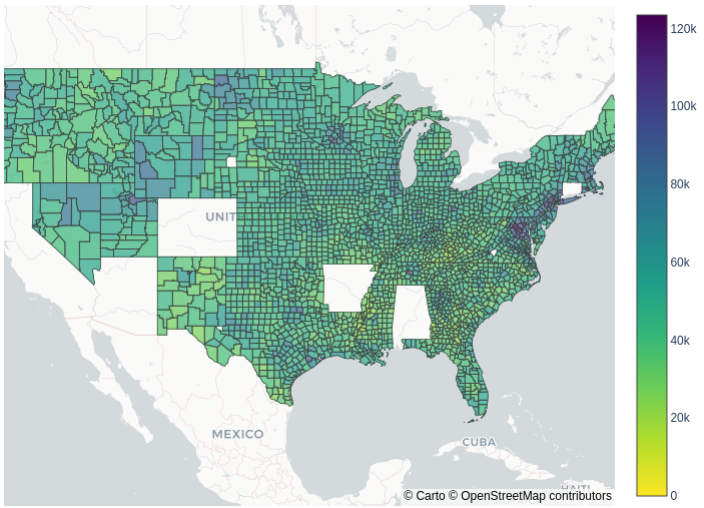
Data Visualization

In [16]:
```
from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)
df=merged
import plotly.io as pio
pio.renderers.default = "png"
import plotly.graph_objects as go

fig = go.Figure(go.Choroplethmapbox(geojson=counties, locations=df.FIPS, z=df.Avg_Ann_Incidence,
                                    colorscale="viridis", zmin=0,zmax=500,
                                    marker_opacity=0.7,marker_line_width=1))
fig.update_layout(mapbox_style="carto-positron",
                  mapbox_zoom=3, mapbox_center = {"lat": 37.0902, "lon": -95.7129})
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

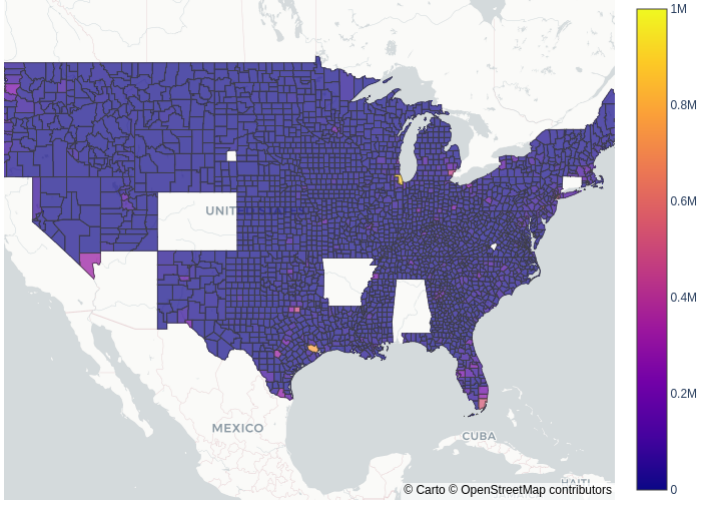
```
/home/rishaab/.local/lib/python3.8/site-packages/plotly/io/_renderers.py:396: DeprecationWarning:

distutils Version classes are deprecated. Use packaging.version instead.
```



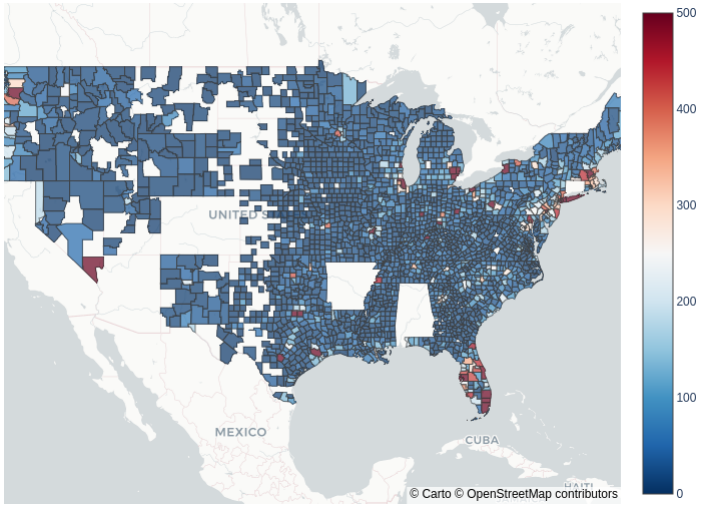
In [17]:
```
fig = go.Figure(go.Choroplethmapbox(geojson=counties, locations=df.FIPS, z=df.Med_Income,
                                    colorscale="viridis_r", zmin=0,zmax=max(df.Med_Income),
                                    marker_opacity=0.7, marker_line_width=1))
fig.update_layout(mapbox_style="carto-positron",
                  mapbox_zoom=3, mapbox_center = {"lat": 37.0902, "lon": -95.7129})
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



In [18]:
```
fig = go.Figure(go.Choroplethmapbox(geojson=counties, locations=df.FIPS, z=df.All_Poverty,
                                    autocolorscale=True, zmin=0,zmax=1000000,
                                    marker_opacity=0.7, marker_line_width=1))
fig.update_layout(mapbox_style="carto-positron",
                  mapbox_zoom=3, mapbox_center = {"lat": 37.0902, "lon": -95.7129})
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



In [19]:
```
fig = go.Figure(go.Choroplethmapbox(geojson=counties, locations=df.FIPS, z=df.Avg_Ann_Deaths,
                                    colorscale="RdBu_r", zmin=0,zmax=500,
                                    marker_opacity=0.7, marker_line_width=1))
fig.update_layout(mapbox_style="carto-positron",
                  mapbox_zoom=3, mapbox_center = {"lat": 37.0902, "lon": -95.7129})
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
```



Handling Missing values

In [20]:
```
med_income_TX=merged[merged['State']=='TX']['Med_Income'].mean()
med_income_TX
```

Out[20]:
```
46745.77865612648
```

In [21]:
```
merged['Med_Income']=merged['Med_Income'].replace(np.NaN,med_income_TX)
df2=merged[merged['Avg_Ann_Deaths'].notna()]
df2
```

Out[21]:

	State	AreaName	All_Poverty	M_Poverty	F_Poverty	FIPS	Med_Income	Med_Income_White	Med_Income_Black	Med_Income_Nat_Am	...	M_Without	F_With	F_Without	All_With	All_Without	Incidence_Rate	Avg_Ann_Incidence
2	AK	Anchorage Municipality, Alaska	23914	10698	13216	2020	78326.0	87235.0	50535.0	53935.0	...	23245	122426	21393	243173	44638	61.5	131.0
3	AK	Bethel Census Area, Alaska	4364	2199	2165	2050	51012.0	92647.0	73661.0	41594.0	...	2708	6627	1774	13023	4482	62.7	6.0
7	AK	Fairbanks North Star Borough, Alaska	7752	3523	4229	2090	71068.0	74242.0	56353.0	48333.0	...	6957	40210	5322	80815	12279	58.1	36.0
9	AK	Juneau City and Borough, Alaska	2110	1145	965	2110	85746.0	90553.0	106964.0	57821.0	...	2433	13582	2213	27321	4646	35.1	9.0
10	AK	Kenai Peninsula Borough, Alaska	5558	2596	2962	2122	63684.0	64663.0	122660.0	46458.0	...	6435	21668	5433	44059	11868	64.9	39.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3129	WY	Sweetwater County, Wyoming	5058	2177	2881	56037	69022.0	69333.0	23535.0	NaN	...	3318	18600	2683	38491	6001	39.9	14.0
3130	WY	Teton County, Wyoming	1638	1026	612	56039	75325.0	77651.0	NaN	NaN	...	2558	9555	1192	18503	3750	23.7	5.0
3131	WY	Uinta County, Wyoming	2845	1453	1392	56041	56569.0	56532.0	NaN	NaN	...	1413	8711	1503	17843	2916	31.7	6.0
3132	WY	Washakie County, Wyoming	1137	489	648	56043	47652.0	48110.0	NaN	NaN	...	691	3490	703	6839	1394	50	6.0
3133	WY	Weston County, Wyoming	958	354	604	56045	57738.0	57842.0	NaN	NaN	...	454	3087	314	6014	768	44.9	4.0

2809 rows × 22 columns

In [22]:
```
merged=merged[merged['Mortality_Rate'].notna()]
```

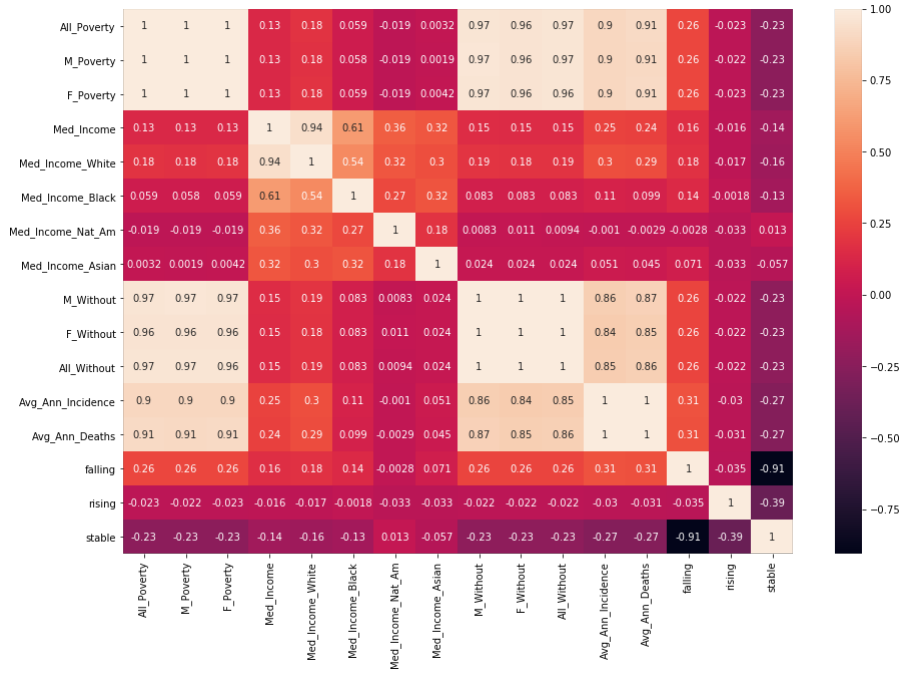
In [23]:
```
merged=merged[merged['recent_trend'].notna()]
```

Adding one-hot encoding for recent-trend

In [24]:
```
re=pd.get_dummies(merged['recent_trend'])
df=pd.concat([merged,re],axis=1)
```

Correlation

In [25]:
```
plt.figure(figsize = (15,10))
sns.heatmap(df.drop(['FIPS', 'M_With', 'F_With', 'All_With', 'Mortality_Rate'], axis=1).corr(), annot = True)
plt.savefig('corrheatmap.png')
```

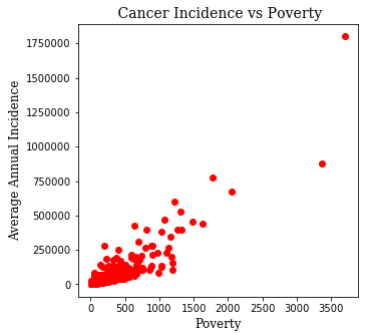


In [26]:
```
df_without_ethnicity = df.drop(['Med_Income_White', 'Med_Income_Black', 'Med_Income_Nat_Am', 'Med_Income_Asian'], axis =1 )
```

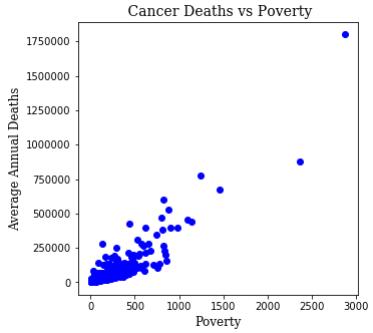
In [27]:
```
df_state = pd.get_dummies(df_without_ethnicity['State'])
df=pd.concat([df_without_ethnicity, df_state], axis=1)
```

In [28]:
```
plt.figure(figsize=[5,5])
font2 = {'family':'serif','color':'black','size':12}
plt.plot(df['Avg_Ann_Incidence'],df['All_Poverty'],'ro',linewidth=2)
font1 = {'family':'serif','color':'black','size':14}
plt.title(' Cancer Incidence vs Poverty',fontdict=font1)
plt.ylabel('Average Annual Incidence',fontdict=font2)
plt.xlabel('Poverty',fontdict=font2)
plt.savefig('Cancer_Incidence_vs_Poverty.png',bbox_inches='tight')
```

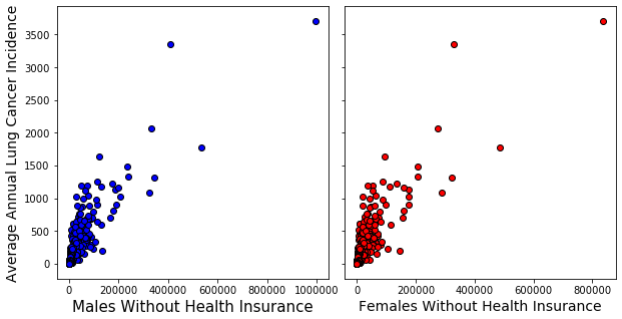
```
/usr/lib/python3/dist-packages/matplotlib/cbook/__init__.py:1402: FutureWarning:

Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy array before indexing instead.

/usr/lib/python3/dist-packages/matplotlib/axes/_base.py:276: FutureWarning:

Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy array before indexing instead.

/usr/lib/python3/dist-packages/matplotlib/axes/_base.py:278: FutureWarning:

Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy array before indexing instead.
```



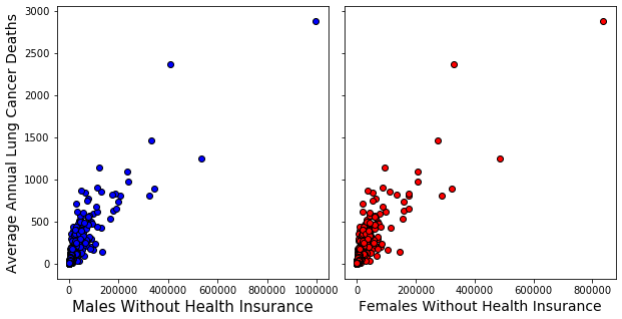
In [29]:
```
plt.figure(figsize=[5,5])
font2 = {'family':'serif','color':'black','size':12}
plt.plot(df['Avg_Ann_Deaths'],df['All_Poverty'],'bo',linewidth=2)
font1 = {'family':'serif','color':'black','size':14}
plt.title(' Cancer Deaths vs Poverty',fontdict=font1)
plt.ylabel('Average Annual Deaths',fontdict=font2)
plt.xlabel('Poverty',fontdict=font2)
plt.savefig('Cancer_Deaths_vs_Poverty.png',bbox_inches='tight')
```



In [30]:
```
fig, ax  = plt.subplots(1,2, figsize = (10,5), sharey=True) #Note that both images have same y-axis

ax[0].scatter(y = df['Avg_Ann_Incidence'].values.astype(float), x = df['M_Without'].values, c = 'blue', edgecolors='black')
ax[0].set_xlabel("Males Without Health Insurance", fontsize = 15)
ax[0].set_ylabel("Average Annual Lung Cancer Incidence", fontsize = 14)
ax[0].tick_params(axis='both', labelsize=10)

ax[1].scatter(y = df['Avg_Ann_Incidence'].values.astype(float), x = df['F_Without'].values, c = 'red', edgecolors='black')
ax[1].set_xlabel("Females Without Health Insurance", fontsize = 14)
ax[1].tick_params(axis='both', labelsize=10)

plt.subplots_adjust(wspace = 0.06) #to adjust the distance between two images
plt.savefig('WithoutInsurance.png')
```



In [31]:
```
fig, ax  = plt.subplots(1,2, figsize = (10,5), sharey=True) #Note that both images have same y-axis

ax[0].scatter(y = df['Avg_Ann_Deaths'].values.astype(float), x = df['M_Without'].values, c = 'blue', edgecolors='black')
ax[0].set_xlabel("Males Without Health Insurance", fontsize = 15)
ax[0].set_ylabel("Average Annual Lung Cancer Deaths", fontsize = 14)
ax[0].tick_params(axis='both', labelsize=10)

ax[1].scatter(y = df['Avg_Ann_Deaths'].values.astype(float), x = df['F_Without'].values, c = 'red', edgecolors='black')
ax[1].set_xlabel("Females Without Health Insurance", fontsize = 14)
ax[1].tick_params(axis='both', labelsize=10)

plt.subplots_adjust(wspace = 0.06) #to adjust the distance between two images
plt.savefig('deathWithoutInsurance.png')
```



Linear Regression

Deaths

In [32]:
```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.preprocessing import RobustScaler
from sklearn.preprocessing import StandardScaler
```

In [33]:
```
data1=data[data['Avg_Ann_Deaths'].notna()]
states=pd.get_dummies(data1['State'])
data_final1 = data1.drop(['M_Poverty','F_Poverty','Med_Income_White', 'Med_Income_Black', 'Med_Income_Nat_Am', 'Med_Income_Asian','State', 'AreaName', 'FIPS', 'M_With', 'F_Wi
scaler=RobustScaler()
X = data_final1.drop(['Avg_Ann_Deaths'], axis = 1)
scaler=scaler.fit(X)
X=scaler.transform(X)
X=np.concatenate([X,states],axis=1)
y = data_final1['Avg_Ann_Deaths']
```

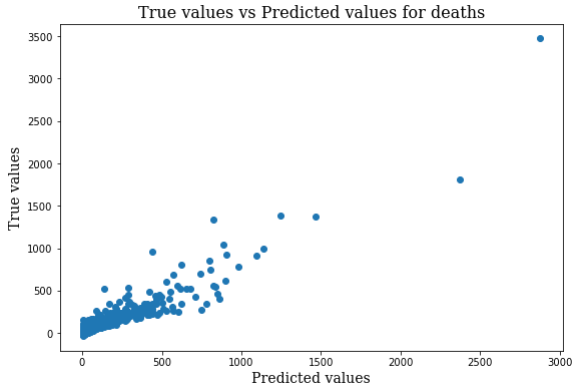
In [34]:
```
lm = LinearRegression()
```

In [35]:
```
lm.fit(X, y)
```

Out[35]:
```
LinearRegression()
```

In [36]:
```
predictions = lm.predict(X)
```

In [37]:
```
plt.figure(figsize=(9,6))
plt.scatter(y,predictions)
font2 = {'family':'serif','color':'black','size':14}
font1 = {'family':'serif','color':'black','size':16}
plt.title('True values vs Predicted values for deaths',fontdict=font1)
plt.ylabel('True values',fontdict=font2)
plt.xlabel('Predicted values',fontdict=font2)
plt.savefig('True values vs Predicted values-Deaths.png',bbox_inches='tight')
```



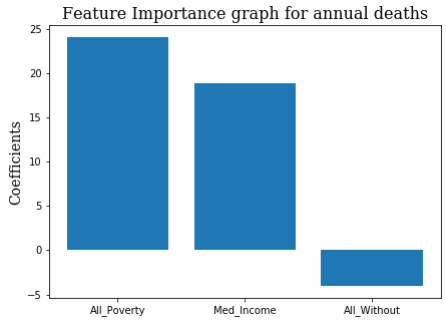
In [38]:
```
r2 = r2_score(y, predictions)
print('r2 score for the model is', r2)
```

```
r2 score for the model is 0.874146596957066
```

In [39]:
```
X=pd.DataFrame(X)
d=pd.concat([data_final1,states],axis=1)
X.columns=d.drop(['Avg_Ann_Deaths'], axis = 1).columns
```

In [40]:
```
importance = lm.coef_
for i,v in enumerate(importance[:3]):
    print('Feature: %s,  Score: %.5f' % (X.columns[i],v))
plt.figure(figsize=(7,5))
plt.bar([c for c in X.columns[:3]], importance[:3])
plt.title('Feature Importance graph for annual deaths',fontdict=font1)
plt.ylabel('Coefficients',fontdict=font2)
plt.savefig('FeatureImportance-deaths.png')
plt.show()
```

```
Feature: All_Poverty,  Score: 24.01030
Feature: Med_Income,  Score: 18.88855
Feature: All_Without,  Score: -3.98286
```



Incidence Rate

In [41]:
```
data2=data[data['Avg_Ann_Incidence'].notna()]
states=pd.get_dummies(data2['State'])
scaler=RobustScaler()
data_final2 = data2.drop(['M_Poverty','F_Poverty','Med_Income_White', 'Med_Income_Black', 'Med_Income_Nat_Am', 'Med_Income_Asian','State', 'AreaName', 'FIPS', 'M_With', 'F_Wi
X = data_final2.drop(['Avg_Ann_Incidence'], axis = 1)
scaler=scaler.fit(X)
X=scaler.transform(X)
X=np.concatenate([X,states],axis=1)
y = data_final2['Avg_Ann_Incidence']
```

In [42]:
```
X
```

Out[42]:
```
array([[ 1.89609085,  2.46848456,  5.32548124, ...,  0.        ,
         0.        ,  0.        ],
       [-0.06853583,  0.47701506,  0.08489396, ...,  0.        ,
         0.        ,  0.        ],
       [ 0.27193247,  1.93930225,  1.10244698, ...,  0.        ,
         0.        ,  0.        ],
       ...,
       [-0.2211838 ,  0.8821771 , -0.11947798, ...,  0.        ,
         0.        ,  1.        ],
       [-0.39282484,  0.23203675, -0.31810767, ...,  0.        ,
         0.        ,  1.        ],
       [-0.41081298,  0.96740914, -0.39980424, ...,  0.        ,
         0.        ,  1.        ]])
```

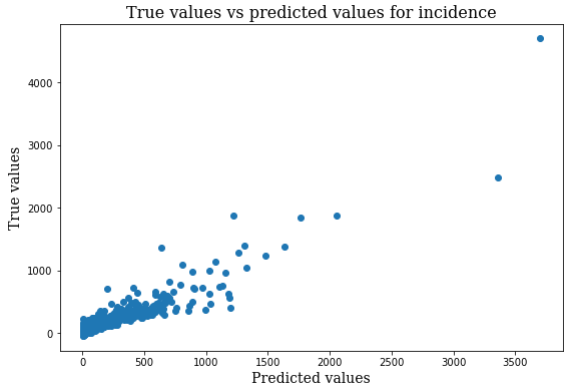
In [43]:
```
lm = LinearRegression()
```

In [44]:
```
lm.fit(X, y)
```

Out[44]:
```
LinearRegression()
```

In [45]:
```
predictions = lm.predict(X)
```

In [46]:
```
plt.figure(figsize=(9,6))
plt.scatter(y,predictions)
font2 = {'family':'serif','color':'black','size':14}
font1 = {'family':'serif','color':'black','size':16}
plt.title('True values vs predicted values for incidence',fontdict=font1)
plt.ylabel('True values',fontdict=font2)
plt.xlabel('Predicted values',fontdict=font2)
plt.savefig('True values vs Predicted values.png',bbox_inches='tight')
```



In [47]:
```
r2 = r2_score(y, predictions)
print('r2 score for the model is', r2)
```

```
r2 score for the model is 0.8625668573268775
```

In [48]:
```
X=pd.DataFrame(X)
d=pd.concat([data_final2,states],axis=1)
X.columns=d.drop(['Avg_Ann_Incidence'], axis = 1).columns
```

In [49]:
```
importance = lm.coef_
for i,v in enumerate(importance[:3]):
    print('Feature: %s,  Score: %.5f' % (X.columns[i],v))
plt.figure(figsize=(7,5))
plt.bar([c for c in X.columns[:3]], importance[:3])
plt.title('Feature Importance graph for incidence rates',font1)
plt.ylabel('Coefficients',fontdict=font2)
plt.savefig('FeatureImportance-incidence.png')
plt.show()
```

```
Feature: All_Poverty,  Score: 34.75648
Feature: Med_Income,  Score: 29.19633
Feature: All_Without,  Score: -6.68979
```