Predicting Community Health to Combat the Leading Causes of Death in the USA; Heart Disease and Cancer

Shiva Safaei –Final Report

• Problem Statement

The community health status depends on many direct and indirect factors. While it is difficult to measure how healthy a community truly is, it is straightforward to check how unhealthy a community is. A substantial indicator of an unhealthy community is its higher death rate. A high death rate could be a consequence of a wide range of parameters from obesity, lack of physical activity, poor nutrition to poverty, education, etc. However the impact of these variables are not always the same since –in another level– they depend on the population characteristics or demographics such as weather, racial diversity, religion etc. Thus it is not the best practice to find a solution for the health problems of a sample community and expand it to others.
The purpose of this project is to better understand the complexity of these influences and identify any causal relationships in order to combat major components of community health for every county in the USA.
The dataset contains crucial information and key local health indicators and encourages dialogue about actions that can be taken to improve community health. Also nationwide analysis provides links between health risk factor and urban indicators such as finance, education, weather, etc. This analysis can serve as baselines for setting goals and targets for the future of community health.
Here, I performed a through study which spans through all the counties of the fifty-two states in the USA, in order to find the most important contributing factors to death. Based on the data (and other reliable sources) I have found that heart disease and cancer are the leading causes of death in the country. Henceforth the focus of this study is these two disease.
With the acquired insight in data analysis part we can create ML model to predict future of communities health.

- Description of Data

Our dataset (here) has 12 different .csv files whose descriptions are listed in an additional .txt file. Among the .csv files 11 of them are distinct –CHSI_DataSet and DATA_ELEMENT_DESCRIPTION are identical. Three of the .csv files incorporate definitions of the acronyms and parameters, where they can be found, explanation of the values and missing values and health indicators in 2010 (DATA_ELEMENT_DESCRIPTION, DEFINED_DATA_VALUE, HEALTHY_PEOPLE_2010).
The 8 remaining files contain the data. The structure is based on demographics where each row represents one county. The total number of rows is 3141, that is equal to the total number of counties in the USA. The columns are thorough categories of information, e.g., type of diseases, the leading cause of death for different ethnicities within different age groups, preventative care, environmental and social conditions etc.
In order to have a meaningful dataset and define the correlation between these parameters they should be normalized and concatenated.

```
LEADING_CAUSES_OF_DEATH.head()
```

	State_FIPS_Code	County_FIPS_Code	CHSI_County_Name	CHSI_State_Name	CHSI_State_Abbr	Strata_ID_Number	A_Wh_Comp	CI_Max_F_Hi_Cancer	LCD_Time_Span
0	1	1	Autauga	Alabama	AL	29	-1111	-1111	1999-2003
1	1	3	Baldwin	Alabama	AL	16	57	-1111	2001-2003
2	1	5	Barbour	Alabama	AL	51	-1111	-1111	1999-2003
3	1	7	Bibb	Alabama	AL	42	-1111	-1111	1994-2003
4	1	9	Blount	Alabama	AL	28	34	-1111	1999-2003

5 rows × 235 columns

Figure 1: An overview of one of the .csv files

* Normalization:

The data for each county is collected in different time spans. For example, Fig. 1 (‘LCD_Time_Span’ column) shows the leading cause of death data for Autauga is collected in 1999-2003 while it is collected in 2001-2003 for Baldwin. Using my keyword search function, I first identified the *time-dependent* columns, then I divided their values by the number of years in which data was taken.

* Neutral parameters:

There are some columns whose number of unique values are equal to 1, which means they are neutral and they make no impact on the dataset. I dropped them.

* Redundant columns:

Columns whose descriptions include: `Favorable indicator`, `Confidence interval lower limit`, `Confidence interval upper limit` and `percentile` are not necessarily needed

for the analysis; they can be calculated whenever they are required. I dropped them.

* Not available or not reported data cleaning:

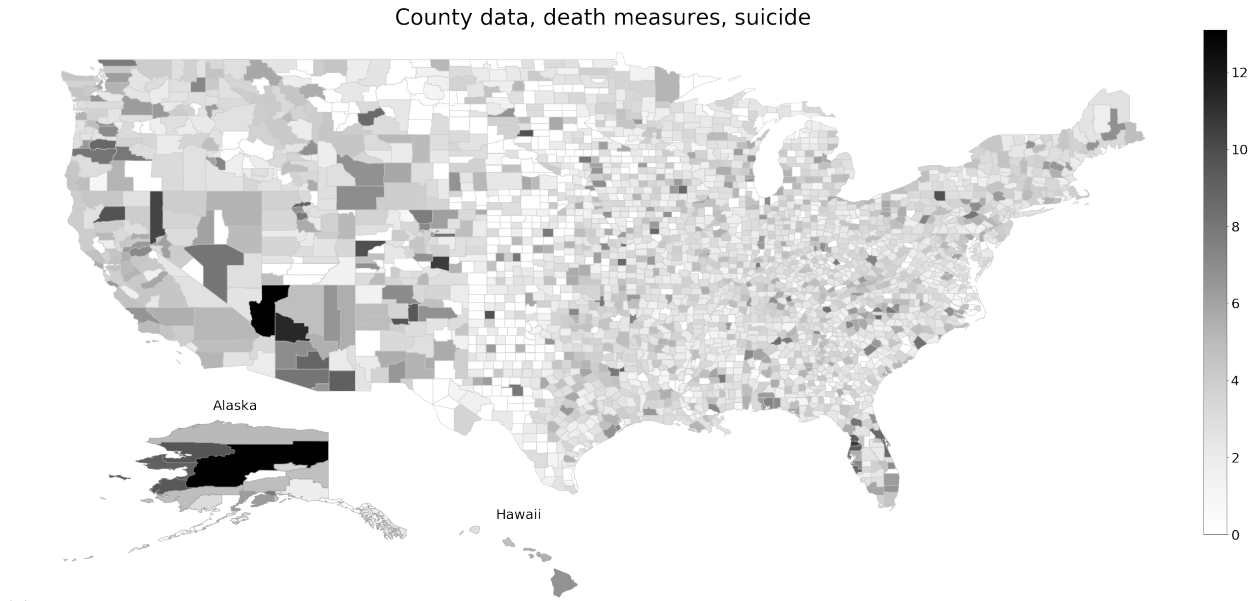
It is shown in DEFINED_DATA_VALUE file that the not-available/reported values are coded as: `nan_values = [-1111, -1111.1, -1, -9999, -2222, -2222.2, -2]`; I replaced them with 0's.
It should emphasized that removing these negative values does not hurt the integrity of the dataset, because all the numerical values in this dataset are either linear functions of time or they are indicators of varieties of magnitudes, which means they are all positive.
The normalization step has to be done before this step, otherwise `nan_values` will be divided by time spans and creates more numbers to be cleaned.

* Merging the data:

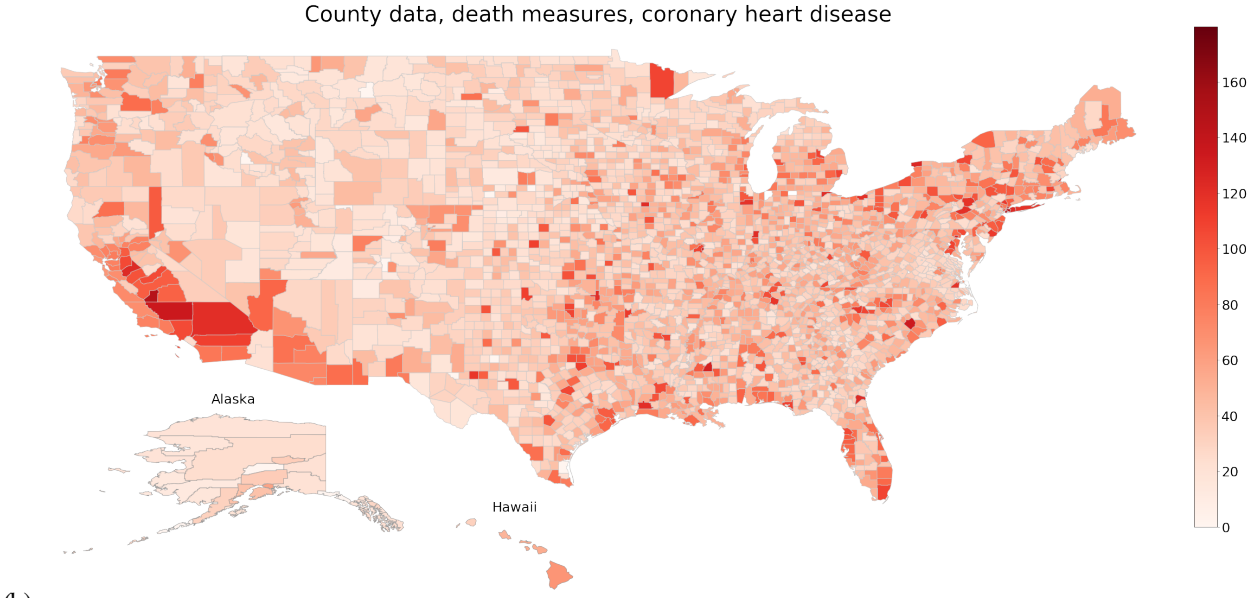
One more cleaning step: A couple of column names were not capitalized and they would not be shown in proper alphabetic order; they are taken care of!
Finally after all of the constituent .csv files are wrangled I concatenated them based on the list of `county_info = ['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name', 'CHSI_State_Name', 'CHSI_State_Abbr', 'Strata_ID_Number'`, which is the common list of columns in all the .csv files.
The final dataframe has 3141 rows corresponding each county and 198 columns of information. Without cleaning the merged dataframe would have 500> columns!

• Exploratory Data Analysis

Our dataframe contains all the counties in the country (rows) with each having various properties (columns). While plotting the numerical properties versus the name of counties is a useful way of illustrating the trends and correlations, it could be overwhelming given the number of counties is over 3000.
Instead representing the features of counties on a geographical map is visually easier to comprehend. In this case we can color code the magnitude of any parameter associated to each county by a different color intensity. This will provide an easy to understand qualitative analysis.
Here I defined two sets of demographic maps, one county based and another one state based. To do so, I merged the map shapefile and cleaned dataframe to make `us_merge` dataframes. Because Alaska and Hawaii are further away and out of proportion I merged them separately. `US_plot` function creates the US map with any numeric variables in the data –default color is blue. The examples of such a representation are Fig. 2a and Fig. 2b which show respectively the suicide and heart disease rate for each counties.



(a)



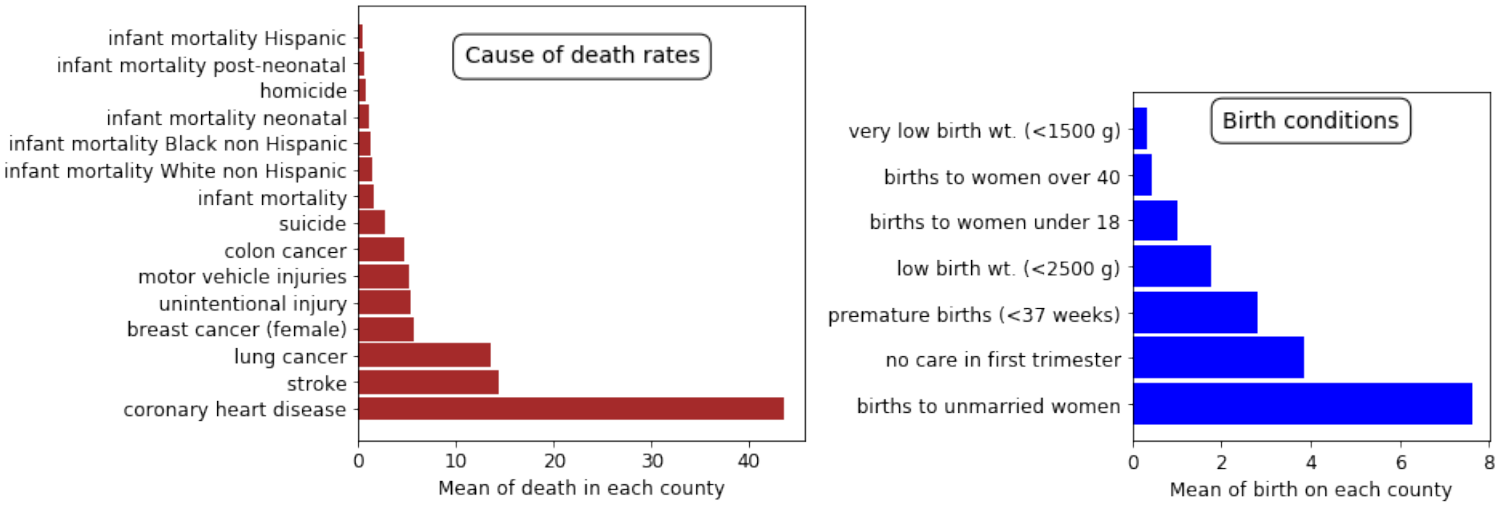
(b)

Figure 2: The representation of the data on the map

* Measures of Birth and Death:

The first question that should be address is: *what is the major cause of death in the country?* Our dataframe provides information about the number of death for different causes per year (since we normalized the data). What we need to do is to find the list of cause of death, and for each one of these causes we need to find the average.
I used a keyword filter to detect columns whose 'description' indicate a cause of death

–column's names alone could not be helpful because they are abbreviations and could not be used directly for the keyword filtering.



(a) (b)

Figure 3: The average cause of death (a) and the average condition of birth (b)

As it is shown in Fig. 3a, the leading cause of the in the nation is coronary heart disease, and (unfortunately) after stroke and lung cancer, breast cancer is the number 4th. Taking to account that only women can have breast cancer and they are roughly half of the population, this ranking is quite astonishing.

* Death measure among different races:

We can further investigate the cause of death among different races and ages. The dataframe contains categorical information about these subgroups. Using the same description search strategy, I created lists of column names corresponding to cause of death for each race and the two age classes of below 25 and above 25 years old. Then we can find the average for each of these columns.
The results are shown in Fig.4 and Fig. 5.

* State-based analysis:

To cover larger scale trends and other interesting aspects of our data, we can switch to the state-based analysis. This could be done by `groupby` state names (or state abbreviations), in which all the counties belong to a state are grouped together and their values are aggregated –here I used sum of their columns values.
As an example Fig. 6 shows the total homicides in each state (sum of homicides of all the counties in the associated state). Similar to the county-based analysis, here I also defined a plotting function which shows the data on the state-based map. For the sake

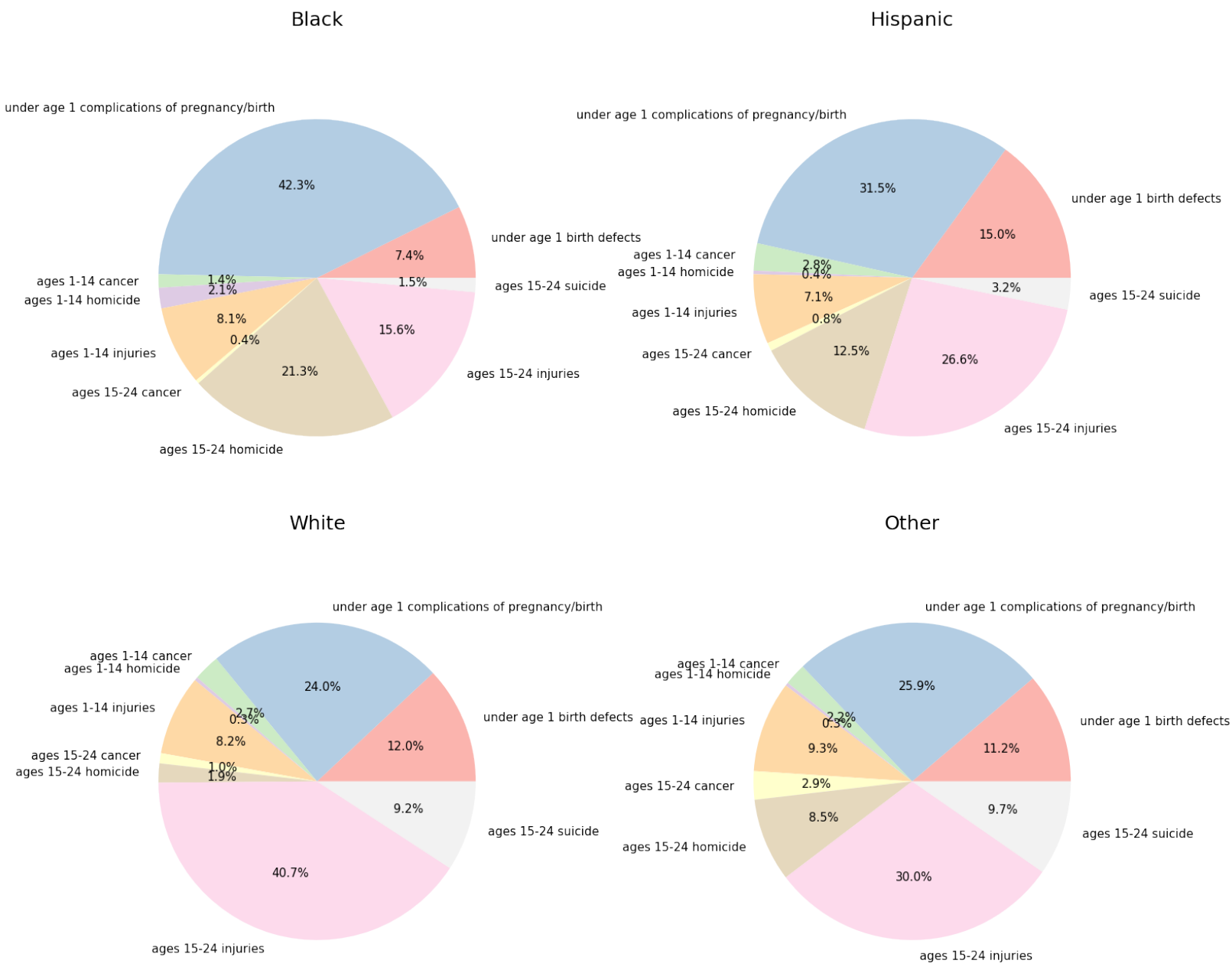

Cause of Death, age under 25: Different Races
Black
under age 1 complications of pregnancy/birth
42.3%
under age 1 birth defects
7.4%
ages 1-14 cancer
1.4%
ages 1-14 homicide
2.1%
ages 15-24 suicide
1.5%
ages 1-14 injuries
8.1%
0.4%
ages 15-24 injuries
15.6%
ages 15-24 cancer
21.3%
ages 15-24 homicide
Hispanic
under age 1 complications of pregnancy/birth
31.5%
under age 1 birth defects
15.0%
ages 1-14 cancer
2.8%
ages 1-14 homicide
0.4%
ages 15-24 suicide
3.2%
ages 1-14 injuries
7.1%
0.8%
ages 15-24 cancer
12.5%
26.6%
ages 15-24 homicide
ages 15-24 injuries
White
under age 1 complications of pregnancy/birth
ages 1-14 cancer
ages 1-14 homicide
24.0%
ages 1-14 injuries
2.7%
0.3%
under age 1 birth defects
12.0%
8.2%
ages 15-24 cancer
1.0%
ages 15-24 homicide
1.9%
9.2%
ages 15-24 suicide
40.7%
ages 15-24 injuries
Other
under age 1 complications of pregnancy/birth
ages 1-14 cancer
ages 1-14 homicide
25.9%
ages 1-14 injuries
2.2%
0.3%
under age 1 birth defects
11.2%
9.3%
ages 15-24 cancer
2.9%
9.7%
ages 15-24 suicide
8.5%
ages 15-24 homicide
30.0%
ages 15-24 injuries

Figure 4:

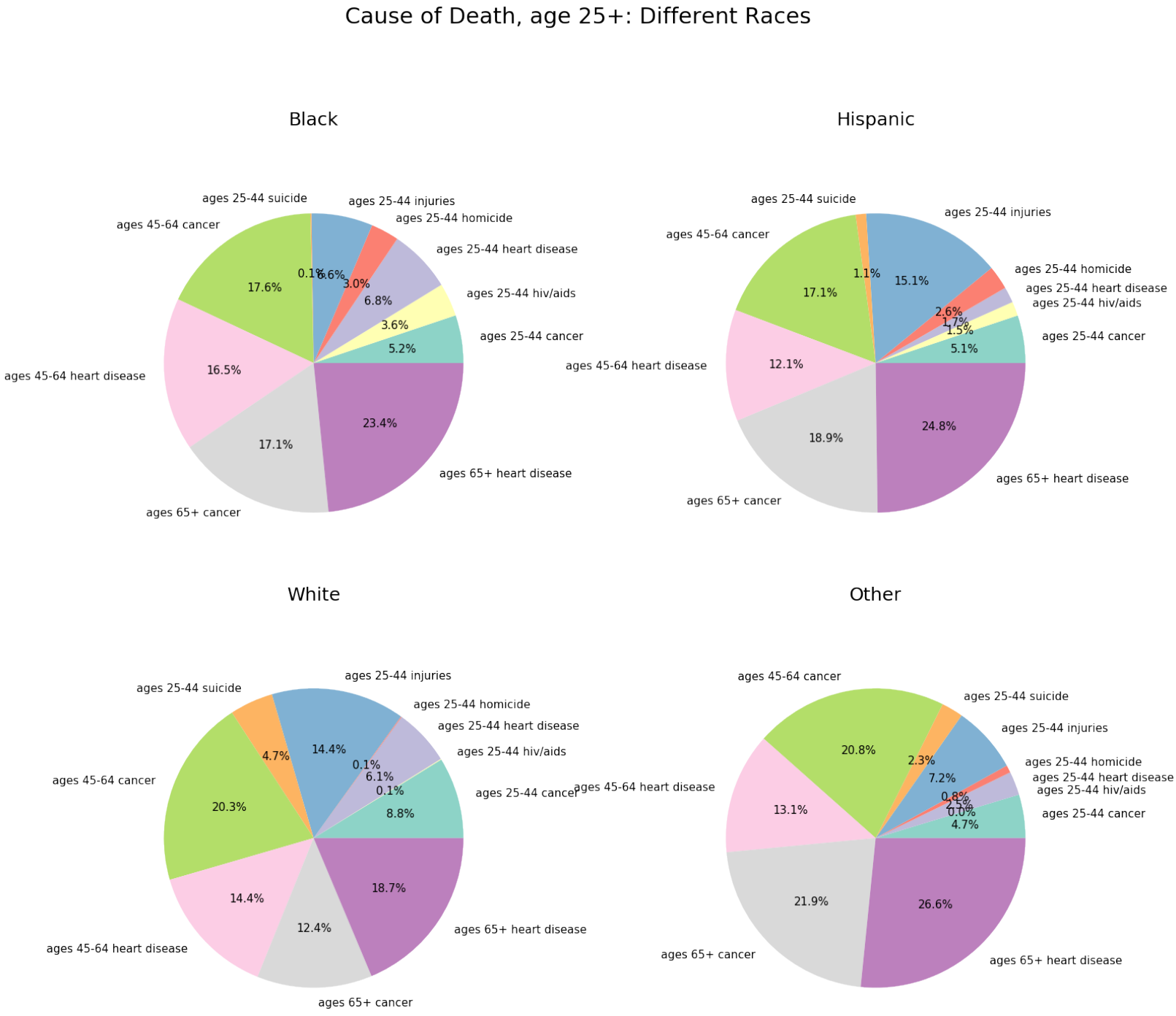

Cause of Death, age 25+: Different Races
Black
ages 25-44 suicide
ages 25-44 injuries
ages 25-44 homicide
ages 45-64 cancer
ages 25-44 heart disease
ages 25-44 hiv/aids
ages 25-44 cancer
ages 45-64 heart disease
ages 65+ heart disease
ages 65+ cancer
0.1%
6.6%
3.0%
6.8%
3.6%
5.2%
17.6%
16.5%
17.1%
23.4%
Hispanic
ages 25-44 suicide
ages 25-44 injuries
ages 45-64 cancer
ages 25-44 homicide
ages 25-44 heart disease
ages 25-44 hiv/aids
ages 25-44 cancer
ages 45-64 heart disease
ages 65+ heart disease
ages 65+ cancer
1.1%
15.1%
2.6%
1.7%
1.5%
5.1%
17.1%
12.1%
18.9%
24.8%
White
ages 25-44 injuries
ages 25-44 suicide
ages 25-44 homicide
ages 25-44 heart disease
ages 25-44 hiv/aids
ages 45-64 cancer
ages 25-44 cancer
ages 45-64 heart disease
ages 65+ heart disease
ages 65+ cancer
4.7%
14.4%
0.1%
6.1%
0.1%
8.8%
20.3%
14.4%
12.4%
18.7%
Other
ages 45-64 cancer
ages 25-44 suicide
ages 25-44 injuries
ages 25-44 homicide
ages 25-44 heart disease
ages 45-64 heart disease
ages 25-44 hiv/aids
ages 25-44 cancer
ages 65+ cancer
ages 65+ heart disease
20.8%
2.3%
7.2%
0.8%
2.1%
0.0%
4.7%
13.1%
21.9%
26.6%

Figure 5:

of comparison with Fig. 2a, Fig. 7 shows the suicide rate per 100.000 people in each state.

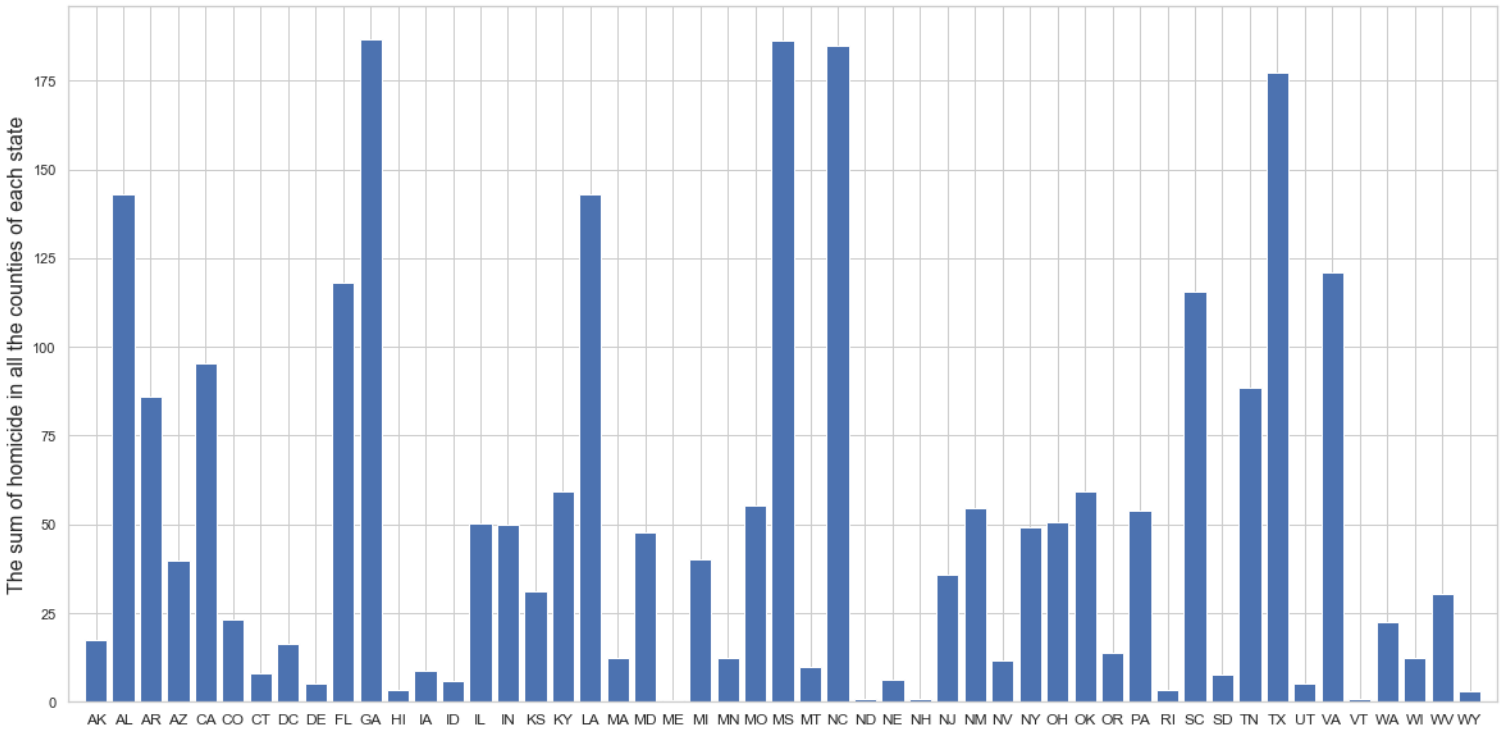


Figure 6:

* Correlations between variables:

Our goal is to identify the features that are directly (positive or negative) contributing to the health or death of people. One simple way is to use Pearson correlation function, e.g., `df.corr(method ='pearson')`. This method renders a 192×192 matrix of the correlation between every single element of our dataframe. However, this is not helpful; there are too many features and it is difficult to analyze which ones are highly correlated. Also, we should get rid of diagonal elements –they mean every feature is 100% correlated with itself.
Hence, in the next step I made a smaller correlation-dataframe with 3 columns:

- column one and two → correlated pairs
- column 3 → the absolute value magnitude of their correlations

To make it easier to see the strongly correlated parameters I sort them. I also defined a threshold 0.7 to cut not very important ones. Fig. 8a shows the 10 highly correlated features in the dataframe. Their linear plots (Fig. 8b) suggest that these parameters are completely correlated. However, they are not informative for the purpose of our study.

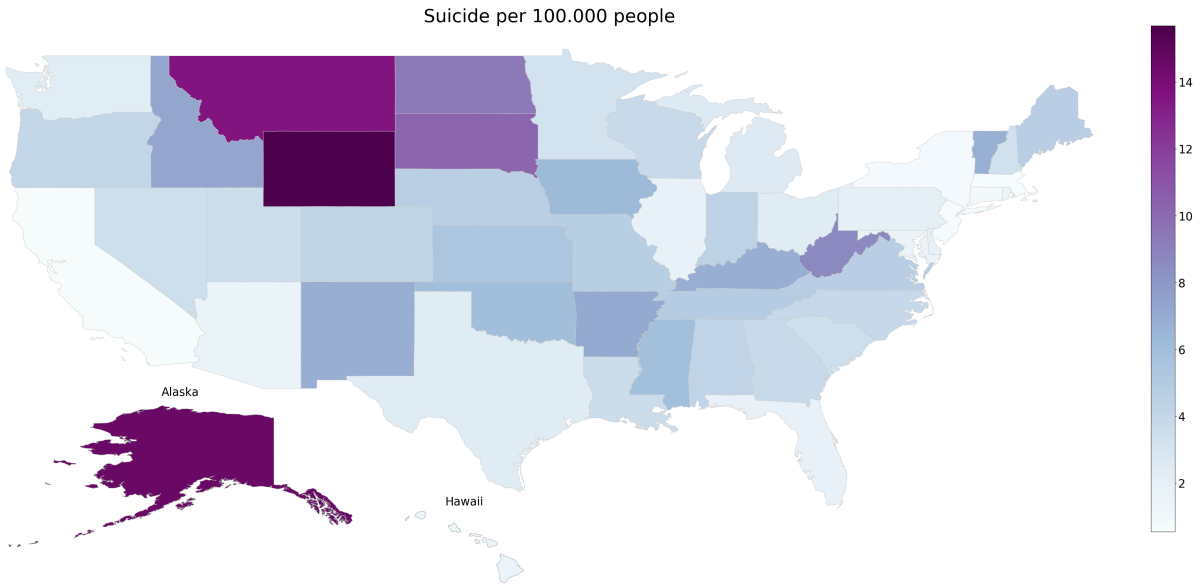
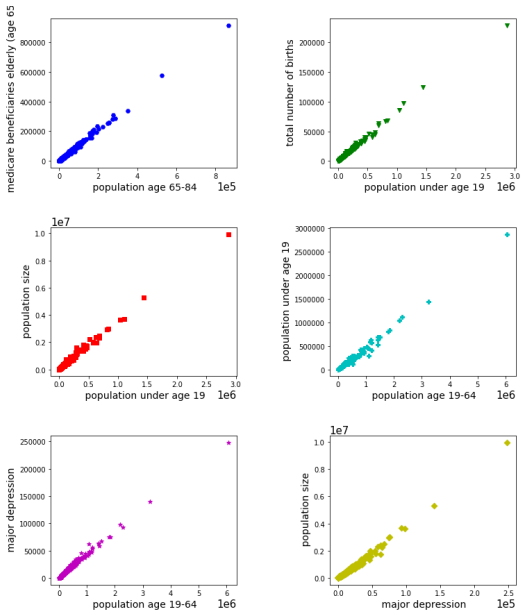


Figure 7:

	attribute_1	attribute_2	correlation
0	Age_19_64	Population_Size	0.999554
1	Age_65_84	Elderly_Medicare	0.997419
2	Age_19_Under	Total_Births	0.997241
3	Age_19_Under	Population_Size	0.996372
4	Age_19_64	Age_19_Under	0.994867
5	Age_19_64	Major_Depression	0.994330
6	Major_Depression	Population_Size	0.993877
7	Population_Size	Total_Births	0.993762
8	Pert_Exp	Population_Size	0.993672
9	Age_19_64	Pert_Exp	0.993333
10	Elderly_Medicare	Total_Deaths	0.993192

(a)

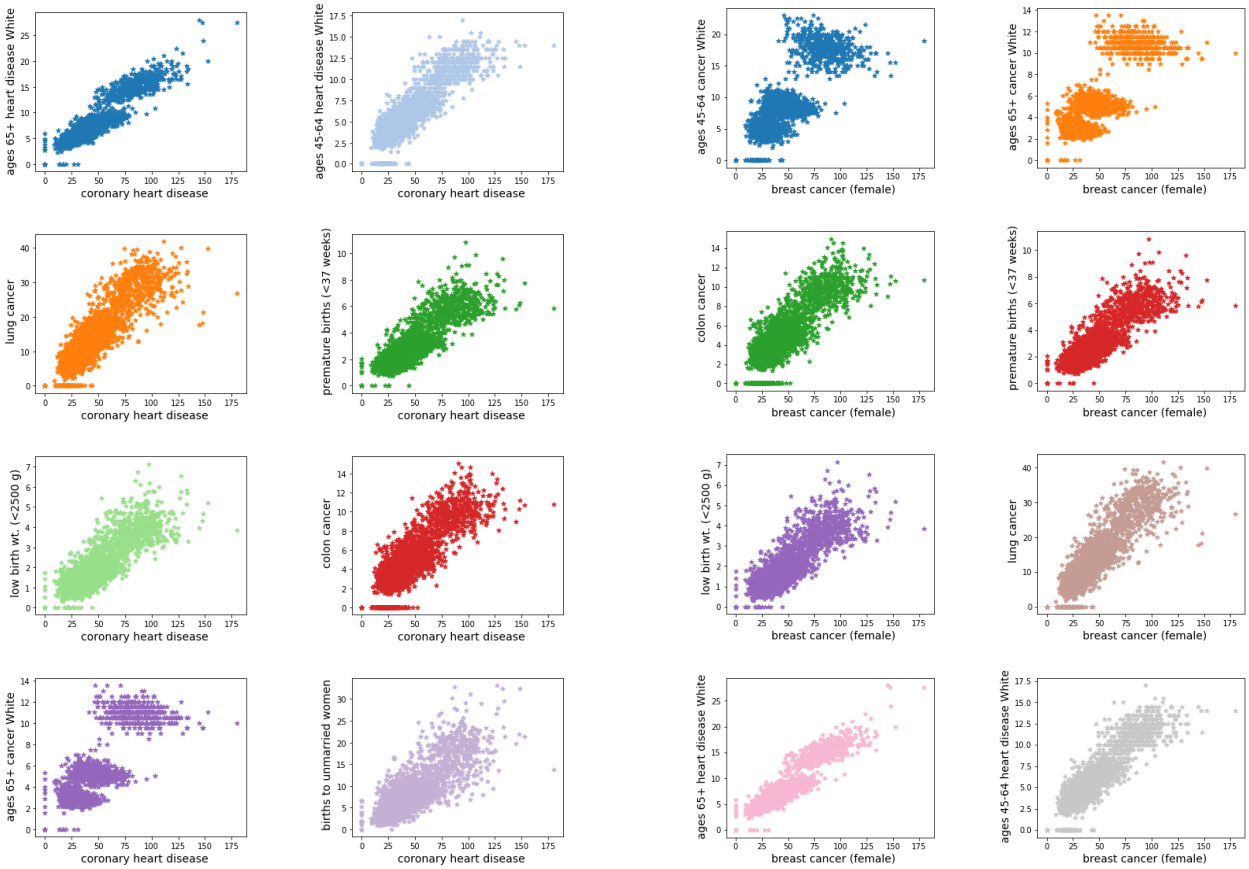


(b)

Figure 8: The table of highly correlated parameters (a) and plots of highly correlated parameters (b)

There are different groups of contributing parameters to communities death; social such as poverty, marriage rate, race, environmental such as exposure to polluted air, weather, and others like access to health care, education, etc. For the purpose of this study, we now pivot our attention to the parameters that impact only on heart disease and breast

cancer.
Such parameters can be extracted by using a filter in the `correlation-dataframe`. The results are shown in Fig. 9a and Fig. 9b. The list of parameters are not exactly causal but they are linearly proportional.



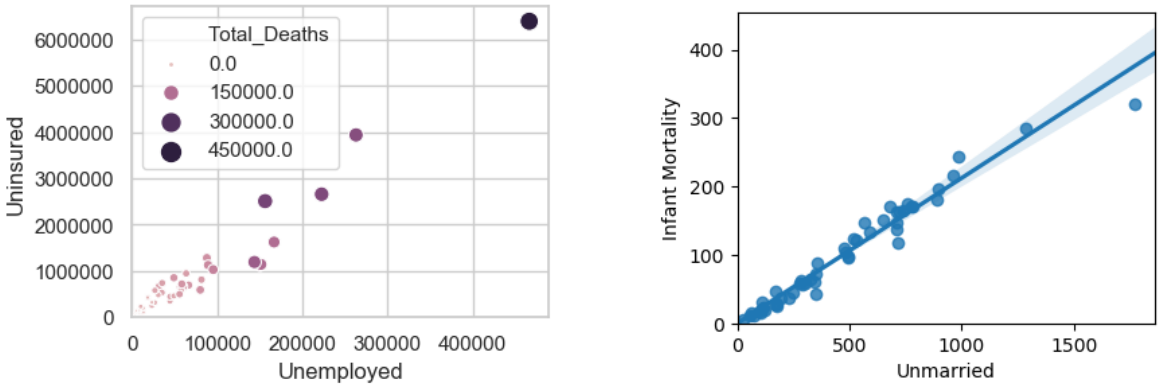
(a) Plots of highly correlated parameters with heart disease

(b) Plots of highly correlated parameters with breast cancer

Figure 9:

- The final dataframe

After preliminary exploratory data analysis and statistical analysis, and having enough insight about the data, important parameters and how they are related to each other, It is time to make a final dataframe (`df_final`) for our modeling part.
We perform a final cleaning and wrangling to remove any duplicates and wrap up the county and state info –we probably need only their names not codes. Ultimately we have a dataframe with only numerical values except the
`county_info_2 = ['CHSI_County_Name', 'CHSI_State_Abbr', 'CHSI_State_Name']`.



(a) (b)

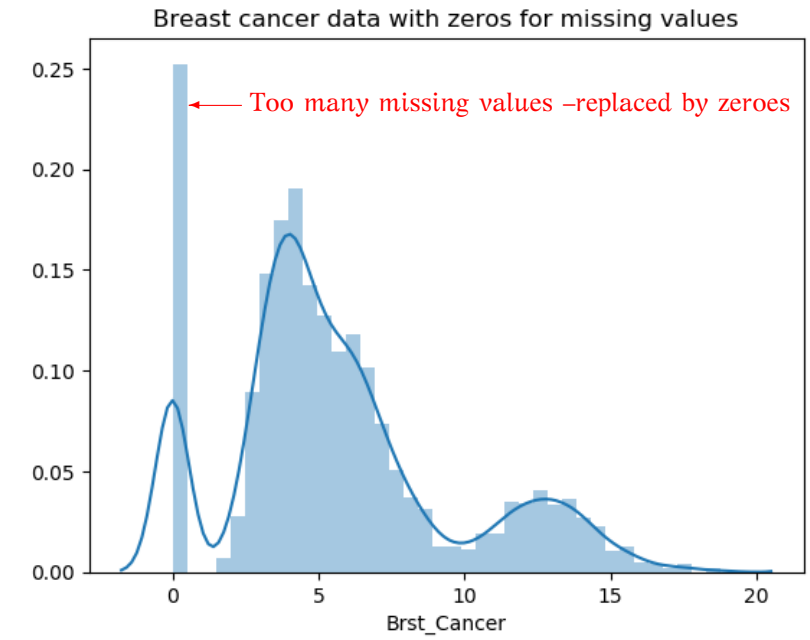
Figure 10: Other highly correlated variables

One last crucial step; If more than 2/3 of the values in a column is zero (or not reported) it would not be helpful and even could hurt our model. I got rid of those columns too!

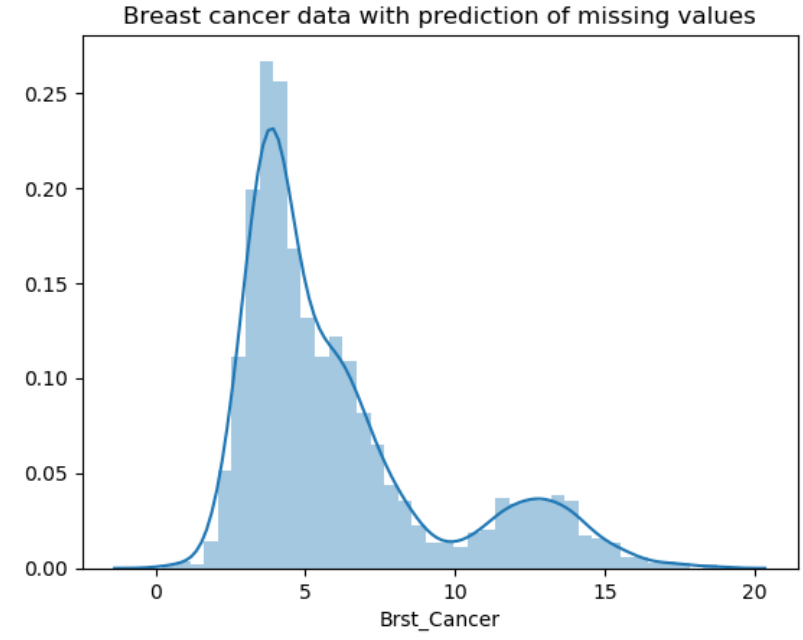
• Modeling –Breast Cancer

∗ Linear Regression to predict missing values:

As it is shown in the Fig. 11a, breast cancer data has many zeroes. These zeroes are not actual data; they are the replacement of the missing and not reporting data as I did in the wrangling section. The problem is these zeroes could affect the outcome of the predictions. If we substituted the missing values by other statistical parameter such as average or mean, it would not be helpful either, since in that case we would have many average or min values.
Fortunately, we can perform linear regression and predict these values.
First, I create a test and a train dataframe which are representing dataframes with breast cancer values are missing and not missing respectively. I use the train dataset and fit it to our linear regression model. Then I predict the missing breast cancer values by applying the `predict` on `y_test`.
Finally, I concatenate the train and predicted dataframes to have a new complete dataset. Fig. 11b shows the new values of breast cancer. It is quite clear that the zeroes are no longer there, instead we have predicted values for 390 counties which they did not have breast cancer information.
Furthermore, with our linear regression, we can find a list of important contributing parameters. The list of important features and their corresponding coefficients are shown in Fig. 12.


Breast cancer data with zeros for missing values
Too many missing values –replaced by zeroes
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
Brst_Cancer

(a)


Breast cancer data with prediction of missing values
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
Brst_Cancer

(b)

Figure 11:

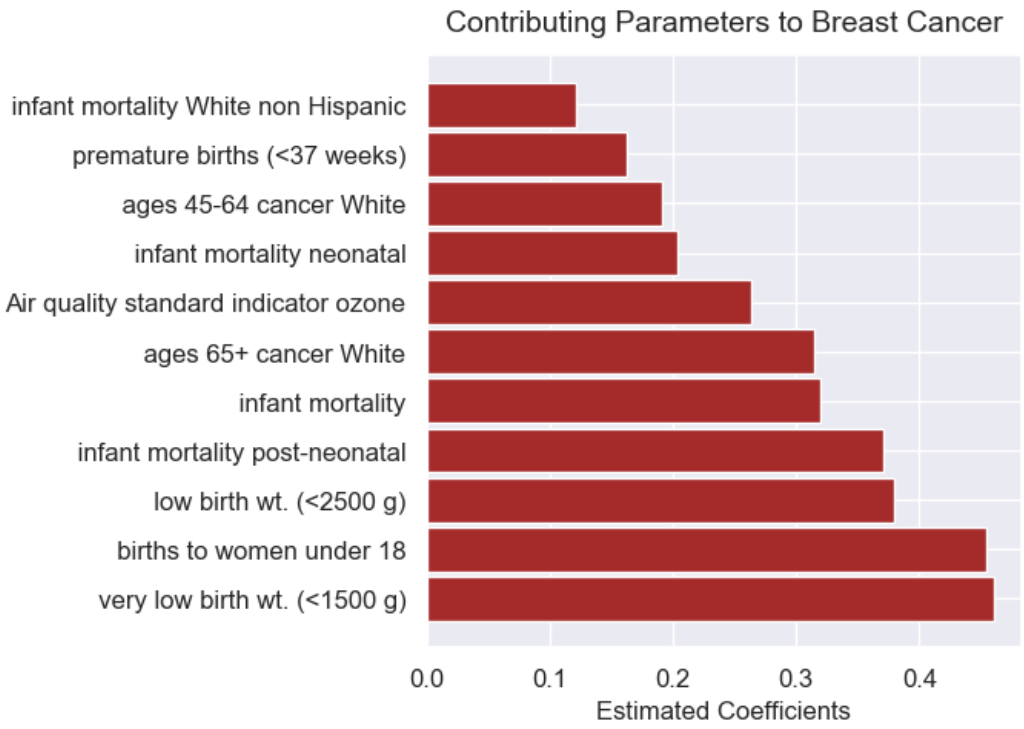
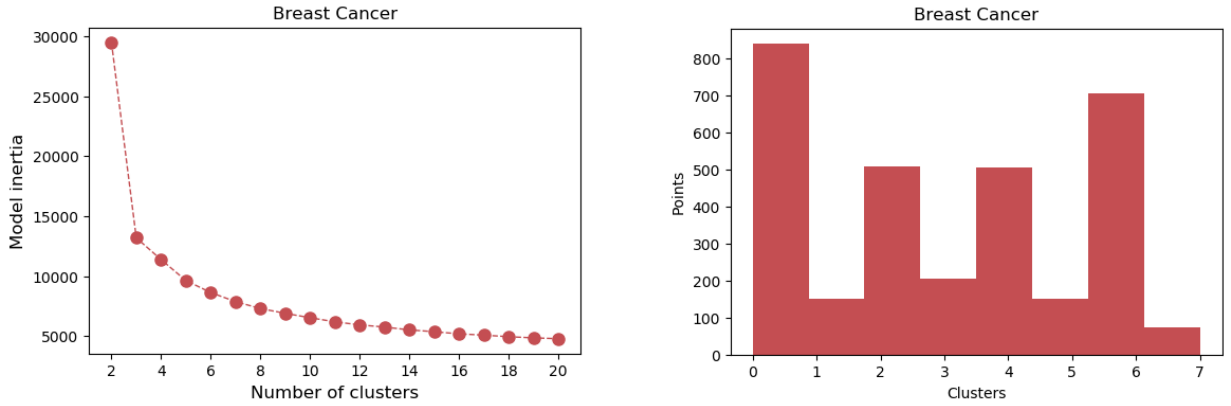


Figure 12: Top contributing parameters to breast cancer predictions –Linear Regression model

* Unsupervised Clustering:

In this part we to classify the counties (or states) based on their similarities in their breast cancer rate.
In order to find patterns of breast cancer, we can perform KMeans clustering method. First, we should know how many clusters are required to fit our data into. The Elbow Sum-of-Squares Method does this (Fig. 13a). If the line chart looks like an arm, then the 'elbow' on the arm is the value of `k` that is the best. The goal is to choose a small value of `k` that still has a low sum of squared error, and the elbow usually represents where we start to have diminishing returns by increasing `k`. I picked `k=8`; Fig. 13b shows the representation of these clusters.
However, choosing the right `k` is not always easy; one could argue that based on the Fig. 13a the optimal `k` is 5 or 9! To clarify this ambiguity we can check it by Silhouette method. The silhouette value measures how similar a point is to its own cluster compared to other clusters. (Silhouette figures could be found in this link –they are too big to be shown here!)
The result of labeled data is depicted in the Fig. 14. It is apparent (from the side bar) that there are 8 groups on the map, marked by 0-7.



(a) (b)

Figure 13: Finding the optimal `k` (a) and optimal `k=8` for KMean clustering method

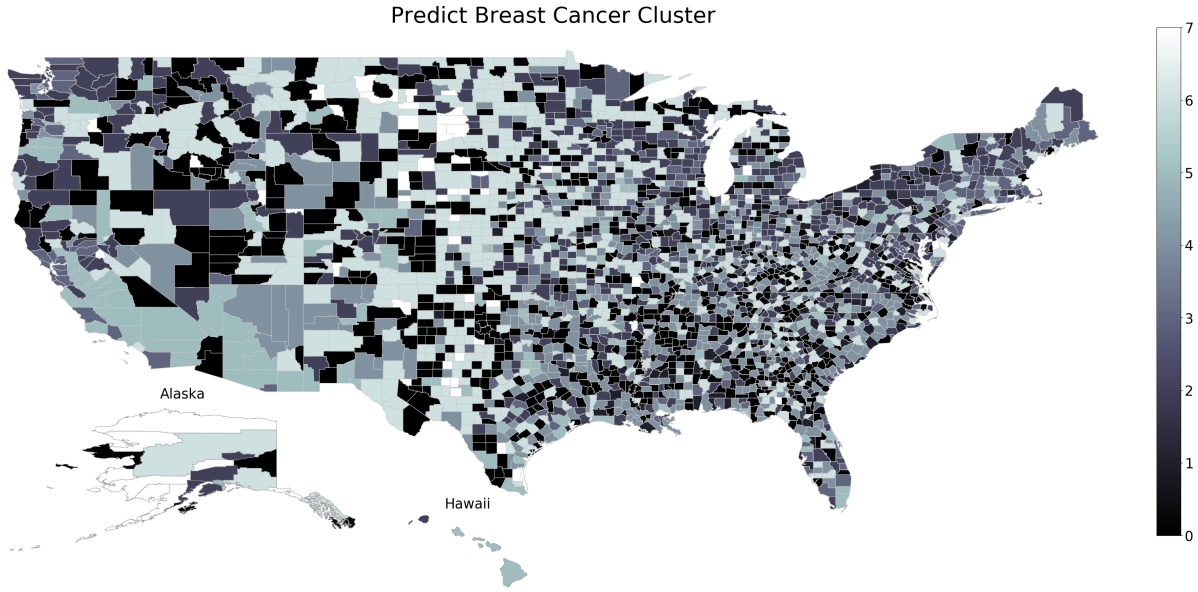


Figure 14: Predicted breast cancer groups using KMean method

• Modeling –Coronary Heart Disease

At the end, we can apply the same strategy we performed to find the missing values of breast cancer data, based on which we classified the counties. The first step is to diagnose the missing values and replace them by predictions of the linear regression model. The predicted data is merged to the original dataframe and we obtained the complete dataset (Fig. 15).
Since the number of missing values for heart disease is just 19 << 3141, much less as compared to the total number of rows, fixing these missing values does not dramatically

change the diagram.
Next, I produce the list of top contributing parameters to heart disease (Fig. 17). Eventually, similar to the breast cancer case I create a clustering model (KMean method) to group the counties. I use both Elbow Sum-of-Squares and Silhouette methods to find the best `k=10`. The results of this analysis are shown in Fig. 16a, Fig. 16b and Fig. 18.

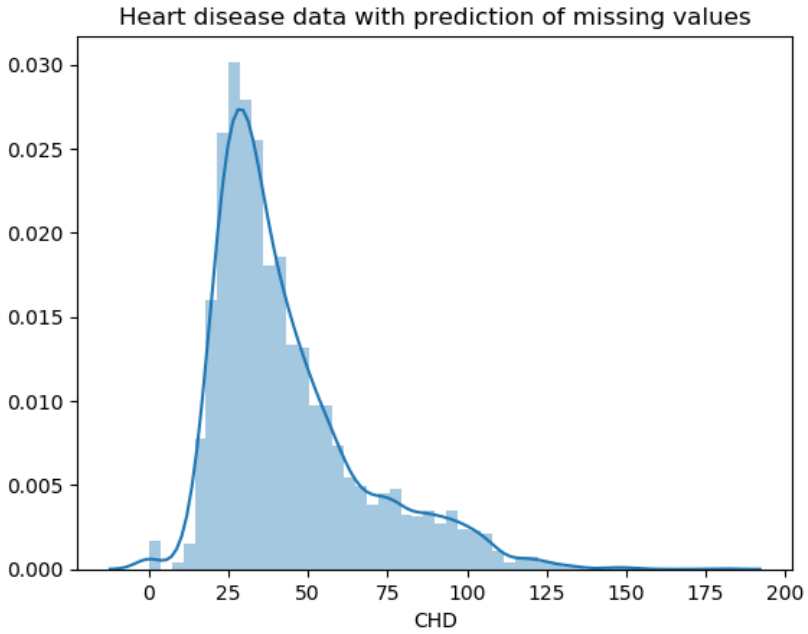
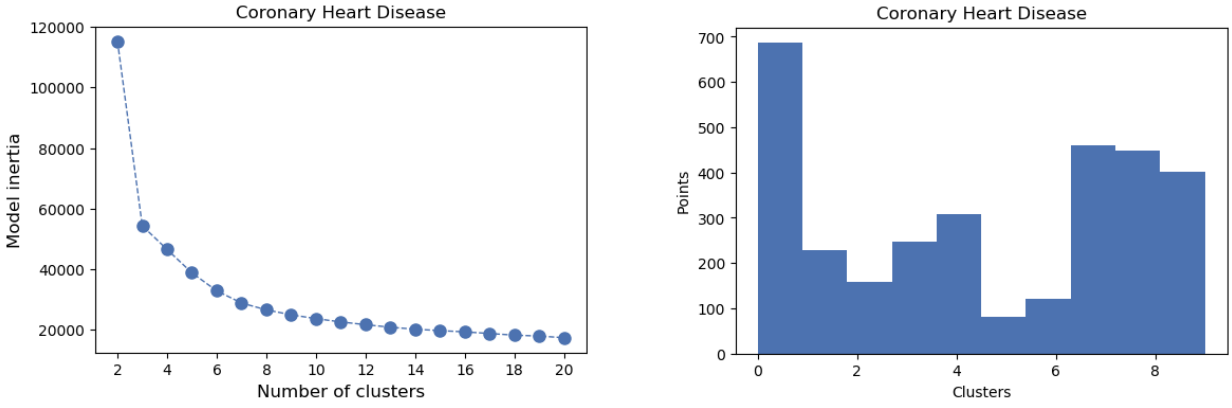


Figure 15:



(a) (b)

Figure 16: Finding the optimal `k` (a) and optimal `k=10` for KMean clustering method

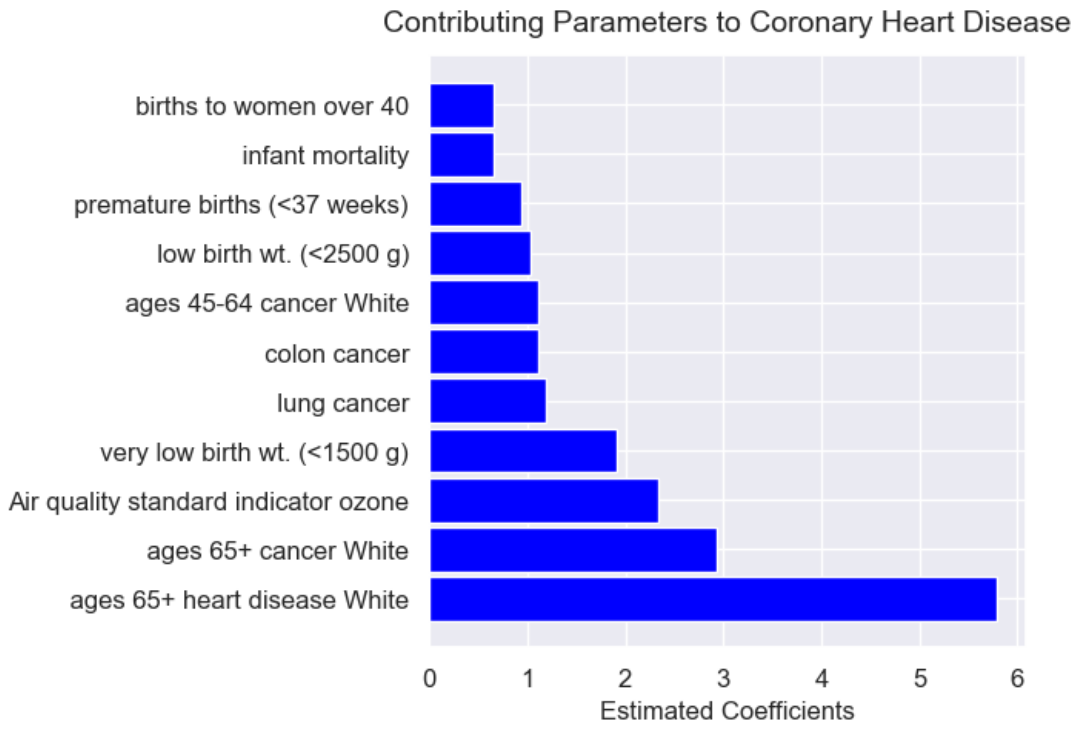


Figure 17: Top contributing parameters to hear disease predictions –Linear Regression model

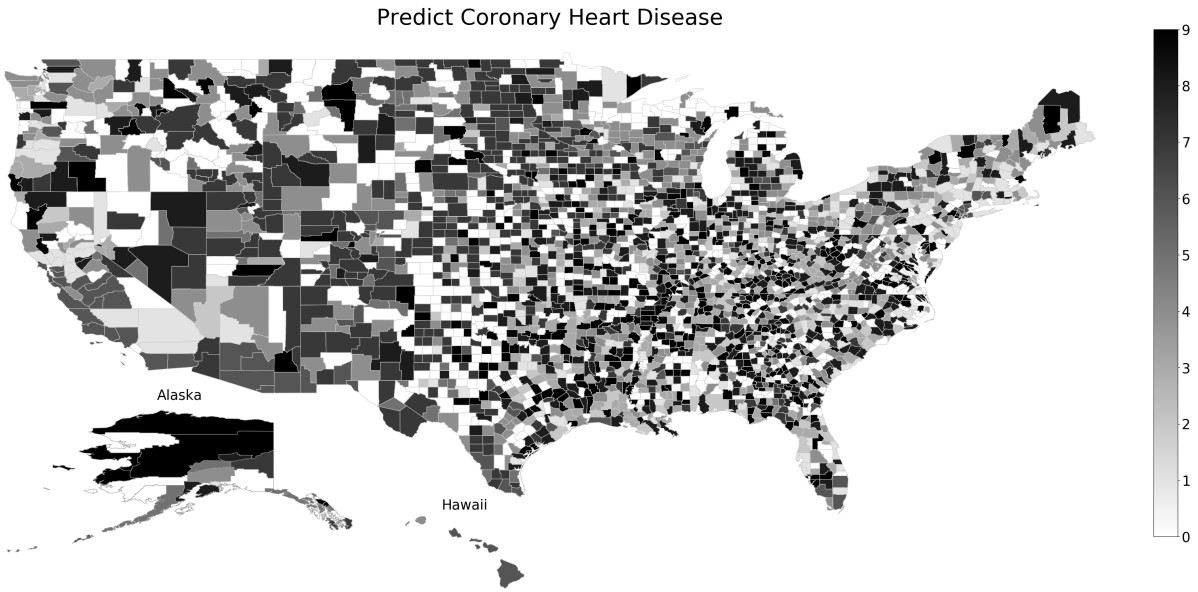


Figure 18: Predicted heart disease groups using KMean method

• Appendix –File Descriptions

DATA_ELEMENT_DESCRIPTION.csv	defines each data element and indicates where its description is found in Data Sources, Definitions, and Notes.
DEFINED_DATA_VALUE.csv	defines the meaning of specific values (such as missing or suppressed data).
DEMOGRAPHICS.csv	identifies the data elements and values in the Demographics indicator domain.
HEALTHY_PEOPLE_2010.csv	identifies the Healthy People 2010 Targets and the U.S. Percentages or Rates.
LEADING_CAUSES_OF_DEATH.csv	identifies the data elements and values in the Leading Causes of Death indicator domain.
MEASURES_OF_BIRTH_AND_DEATH.csv	identifies the data elements and values in the Measures of Birth and Death indicator domain.
PREVENTIVE_SERVICES_USE.csv	identifies the data elements and values in the Preventive Services indicator domain.
RELATIVE_HEALTH_IMPORTANCE.csv	identifies the data elements and values in the Relative Health Importance indicator domain.
RISK_FACTORS_AND_ACCESS_TO_CARE.csv	identifies the data elements and values in the Risk Factors and Access to Care indicator domain.
SUMMARY_MEASURES_OF_HEALTH.csv	identifies the data elements and values in the Summary Measures of Health indicator domain.
VULNERABLE_POPS_AND_ENV_HEALTH.csv	identifies the data elements and values in the Vulnerable Populations and Environmental Health indicator domain.