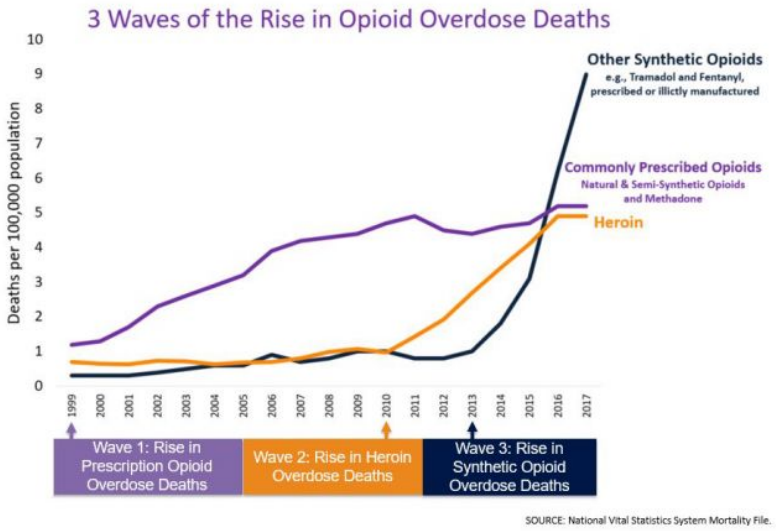
Capstone 1 Final Report
Predicting Overdose Risk from Opioid Prescriptions

1. Problem Statement

The problems of this project are threefold:

a. What is an individual's risk of overdose based on background information for their county, including counts of opioid prescriptions and demographic data?
b. Which factors are best used to predict death rate from opioid overdose?
c. The CDC has identified 3 separate "waves" in the ongoing opioid crisis, the second being the rise in heroin overdoses around 2010 (https://www.cdc.gov/drugoverdose/epidemic/index.html). Will we see any evidence of this in our data? For reference:



This project would be most useful to the Department of Health and Human Services, or other more localized health services, who could use the information to better target their addiction and recovery programs. Additionally, law enforcement agencies such as the DEA could use the information to better track illicit drug distributors based on top consumers. The American Medical Association or private law firms could also pursue medical malpractice suits against doctors whose overprescription of opioids has damaged their communities.

2. Data Collection and Wrangling

The data for this project come from three sources and were cleaned separately:

a. The data for opioid prescriptions was compiled by The Washington Post and held on Kaggle.com as part of a competition. It includes virtually every purchase of opioids by a legal distributor across the United States from 2006 to 2012. Features include state, county, city, zip code, quantity, and other buyer information.

   The size of this datafile was enormous - over 80 GB zipped and 1.5 trillion lines. Because of this, directly loading the original dataset was simply not possible. Instead, I used a random number generator to skip 95% of the file so that I could work with 5% of the total data. Assuming the transaction records were stored randomly, even this 5% is sufficiently large to assume it is a representative sample from the original data.

   One of the main challenges of cleaning this data was that the zip codes and transaction dates were stored as integers, which resulted in front zeros being dropped. My solution was to convert this data to strings and add front zeros with a loop. I also converted the dates to a datetime object and extracted the year, which was more important for this project than the specific date.

   I used Pandas to examine records with missing location data and found 111 rows out of nearly 9 million total without county data. Because this was such a miniscule amount in comparison to the total, I simply dropped these records. Next, I grouped by location and year, then summed records, in order to get individual rows for each county in each year 2006-2012.

   Finally, I converted the abbreviations to full state names using a dictionary and the Python map function, then removed any potential extra white spaces using the built-in strip function.

   Source: https://www.kaggle.com/paultimothymooney/pain-pills-in-the-usa

b. Statistics on opioid overdose deaths can be found by making a request to the Multiple Cause of Death Data found on the CDC Wonder website. From the website: “Data are based on death certificates for U.S. residents. Each death certificate contains a single underlying cause of death, up to twenty additional multiple causes”. While there are literally hundreds of possible combinations for multiple causes of death, the database does allow for requests to be limited to overdose deaths resulting specifically from opioid use. The resulting dataset contained death counts for every American county over the period 2006 to 2012.

   As is common for government data, queries to the Wonder database had a low limit on file size. This meant I had to make several queries and merge the results together before taking further data cleaning steps. First, I converted the year data to a datetime object to keep it consistent with the prescription data.

In many cases, when data on opioid deaths was not properly recorded or authorized for release, it was recorded as “Suppressed”; to ensure ease of manipulation I replace these values with Pandas’ null data type ‘NaN’ (not a number).

Unlike the prescription dataset, which simply had county names, this dataset included the title “County” and the state abbreviation as part of the name. I stripped this extra information, removed white space, and merged it with the prescription data.

Source: https://wonder.cdc.gov/mcd.html

c. Demographic data was obtained from the 2010 U.S. Census. From the database found online I requested data for each American county, divided by age, sex, and race. There are 5 racial groups - Asian, Black, White, those of Pacific Islander descent, and those of Native American or Indigenous descent. The age groups are divided 5 years at a time (10-14, 15-19, etc.) from ages 0 to 84, then 85 and up.

   The first step in cleaning this dataset was to change the age group column from reference numbers (1 = ages 0-4, 2 = ages 5-9, etc.) to descriptive names. The demographics for each group were recorded as raw population numbers, but I wanted to normalize them as percentages of the population, so I simply divided them by the population total. To avoid redundancy I dropped the original population totals, as well as columns which accounted for mixed-race people, who had previously been recorded in other racial groups.

   Data for each age group was recorded as separate rows. However, since I wanted each row to include all data for a given county, I used Pandas’ pivot table function to move these rows to columns.

   As with the previous dataset, I stripped white spaces and merged it with the previous datasets.

   Source: https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml

3. Machine Learning Methods

Typically, one would start building machine learning models after preliminary data analysis. However, the data I found on opioid overdoses was incredibly sparse; I suspect they are often underreported, perhaps due to lack of infrastructure in rural areas or due to political interests. As a result, it was virtually impossible to create visualizations without first using a model to predict the missing data.

I decided that this was a supervised learning problem, considering there are clear labels for both features and targets. Further, because this project involves mathematically computing rates from previous data instead of labelling/classifying data, I believed I should use a model with regression instead of classification. I started with the most fundamental regression model: linear regression.

On its face, I had over 250 potential features to use for this model, including year, state, county, prescription rate, population. Among these features were the demographic data, which consisted of population percentages for different groups divided by age, race, and sex. My aim in building these machine learning models was to discover which features would produce the best model, in terms of both performance and interpretability. Each model roughly follows the same construction plan - define target and features, drop rows with missing data since they cannot be used in the model, split the remaining data into training and test data, then fit the model and measure and accuracy.

I hypothesized that there was a linear relationship between prescription rate and death rate. Consequently, I built my first linear model using Sklearn with just prescription rate as a feature. Its accuracy score was around 20%, which told me that prescription rate was indeed important, but would not suffice on its own. Adding year as a feature to prescription rate actually produced a much worse score, which told me year should probably be excluded from the model.

Next, I wanted to test county and state as features, but Sklearn does not allow categorical data to be used for linear models as is. This led me to use Pandas' get_dummies function to generate 1521 columns for each county and state, which hold a 1 if they represent a given location and a 0 otherwise. I also set the function parameter drop_first to True, in order to avoid redundant data which can negatively impact Sklearn's models. The score for this model was quite low - ~0.3% - but I thought it might be because of the large number of features. The negative impact of an overabundance of features, called the "curse of dimensionality", can be overcome with dimensionality reduction. For this I used Sklearn's Principal Component Analysis (PCA), which is able to convert these features to lower-dimensional space, in this case from 1522 features to 100 components. This model with reduced features was much improved, with a score of 35%. I also examined the model's explained variance ratio for its components, which showed that every component was making a significant contribution to the model.

For my next model I tested demographic data as the features, which had an accuracy of 43%. Combining demographic data with prescription rate improved the model even more, with it predicting at 48% accuracy. It seemed both features should be used in the final model.

To my disappointment, adding prescription rate to the county and state data with PCA only marginally improved the accuracy (35%). Running a similar model without PCA resulted in a clear case of overfitting, with a training accuracy of 72% and a test accuracy around 0%. My linear model with county and state data plus demographic data improved somewhat to 39%, which is about how well the next model performed when I added prescription rate again. Location data seemed to have some predictive power, but performed poorly with other data.

To this point, the best model I had was linear regression with demographic data and prescription rate as features. I decided to test other algorithms to see if I could get further improvement. I implemented Sklearn's Random Forest Regressor, which builds many decision tree estimators and takes the mean of them for predictive power. Instead of running this algorithm a number of times with different numbers of estimators, I instead used Sklearn's GridSearchCV, which allows you to cross-validate data and optimize model hyperparameters at the cost of runtime. With GridSearch I also utilized Lasso and Ridge regressions; these algorithms are similar to linear regressions, but they penalize some coefficients to reduce model complexity and overfitting, the difference being that Lasso can reduce some coefficients to 0 (which also allows for features selection; more on that later). GridSearch is useful for choosing the regularization parameter of these models, which determines how much the variables are reduced. Compared to the previous accuracy of 48% for linear regression, these models scored -.09%, 37%, and 42% for Random Forest, Lasso, and Ridge respectively. Rerunning the Random Forest model with Randomized Search CV (which runs more quickly than GridSearch by not running every single model) did not change the accuracy.

I was seeing only incremental improvements in my model accuracy, yet they were all still far from good models. After much consideration, I shifted the target variable from death rate to number of deaths, which allowed me to use population as another feature. I reran my basic linear regression model with population, prescription rate, and demographic data as features and immediately saw drastic results: a 91% accuracy score. When I constructed another model adding county and state data, I saw a slight improvement in test accuracy, but because it was higher than training accuracy, I consider this result to be unreliable. Furthermore, there is good reason to not use a model with PCA in this project: because my aim is to see how individual features contribute to the model, dimensionality reduction actually makes interpreting the final result quite difficult.

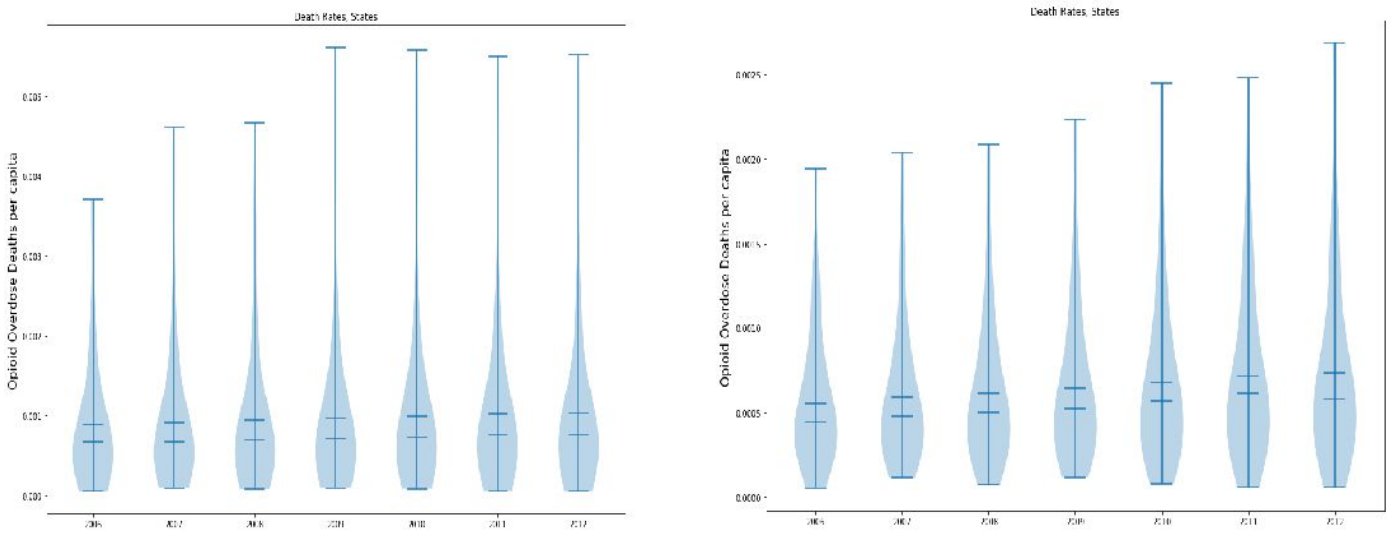
The hyperparameter table for all my models is below. Note that models which used a search function to find the best parameters have unknown training accuracy.

model	target	n_features	features	test accuracy	training accuracy
Linear Regression	death rate	1	prescription rate	0.205	0.149
Linear Regression	death rate	2	prescription rate, year	0.007	0.0099
Linear Regression	death rate	1521	county, state (dummy variables)	0.003	0.005
Linear Regression with PCA	death rate	1521	county, state (dummy variables)	0.346	0.341
Linear Regression	death rate	249	demographic data	0.432	0.525
Linear Regression	death rate	250	demo data, prescription rate	0.484	0.553
Linear Regression with PCA	death rate	1522	county, state, prescription rate	0.351	0.343
Linear Regression, no PCA	death rate	1522	county, state, prescription rate	-3.68E+23	0.724
Linear Regression, no PCA	death rate	1770	county, state, demo data	0.031	0.038
Linear Regression with PCA	death rate	1770	county, state, demo data	0.392	0.409
Linear Regression with PCA	death rate	1771	county, state, prescription rate, demo data	0.391	0.411
Random Forest	death rate	250	demo data, prescription rate	-0.0009	-
Lasso Linear	death rate	250	demo data, prescription rate	0.368	-
Ridge	death rate	250	demo data, prescription rate	0.422	-
Random Forest with Randomized Search CV	death rate	250	demo data, prescription rate	-0.0009	-
Linear Regression	deaths	251	demo data, prescription rate, population	0.908	0.914
Lasso Linear	deaths	251	demo data, prescription rate, population	0.864	-
Ridge	deaths	251	demo data, prescription rate, population	0.871	-

Linear Regression with PCA	deaths	1772	county, state, prescription rate, demo data, population	0.912	0.889

Despite its impressive performance, there were problems with using the linear regression model above. Because a linear regression is a simple combination of features which can have negative coefficients, the model was actually outputting negative values for death totals. One possible solution was to use Numpy's clip function to hold the lower limit of values to 0, but this led to a decrease in predictive accuracy.

I began to look to Lasso as a good alternative to basic linear regression. Lasso has several advantages: it can be programmed to output only positive values (at the cost of some accuracy) and it makes feature selection quite easy, as it sets the coefficients of unimportant features to 0. I also compared the spread of predictions for each model visually, using violin plots (linear regression on the left, lasso on the right):

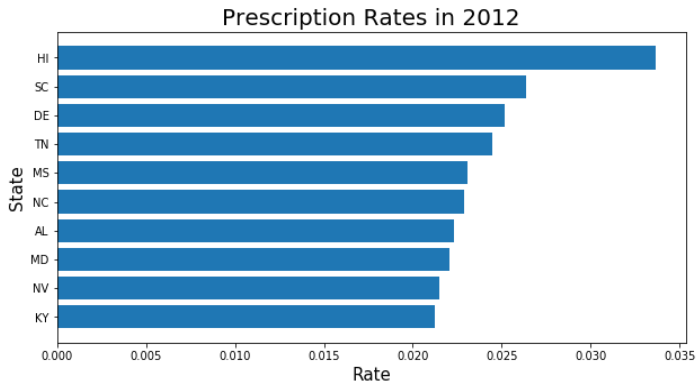


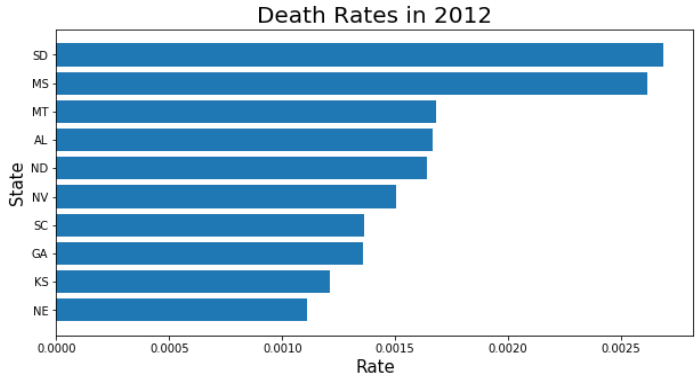
These visualizations tell me that a) the spread of values is more reasonable for the Lasso model, considering death rates should be much smaller than prescription rates, and b) we see a steady increase in death rate over time, which matches the CDC's own analysis. Though the predictive accuracy of the Lasso model is lower (~80%), it seemed to be more useful for this project.

4. Exploratory Data Analysis

a) Visual Analysis

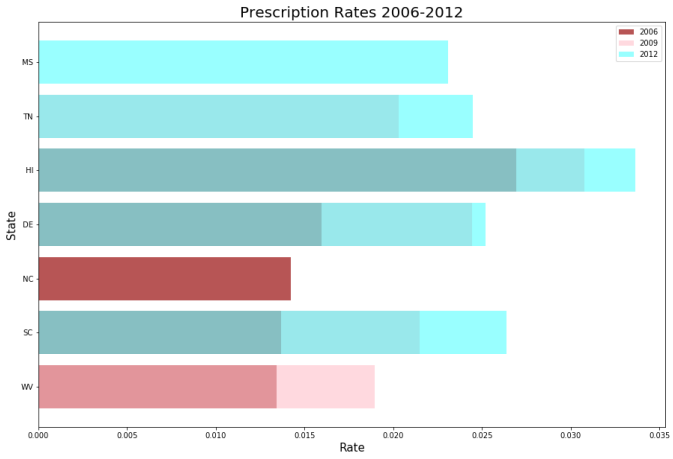
First, I wanted to see who the top states are for prescriptions and opioid overdoses, taking 2012 as an example year:

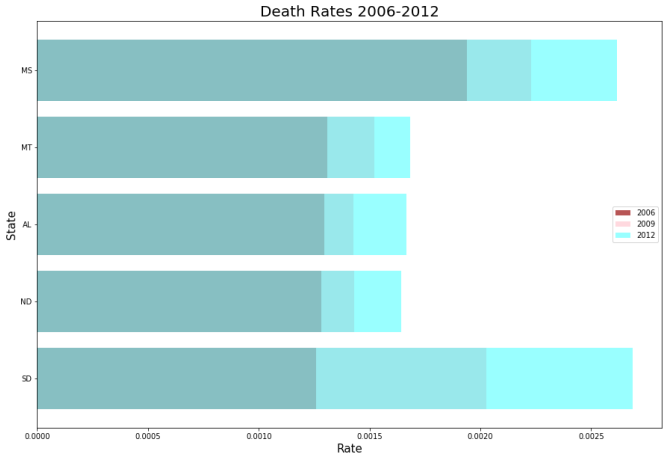




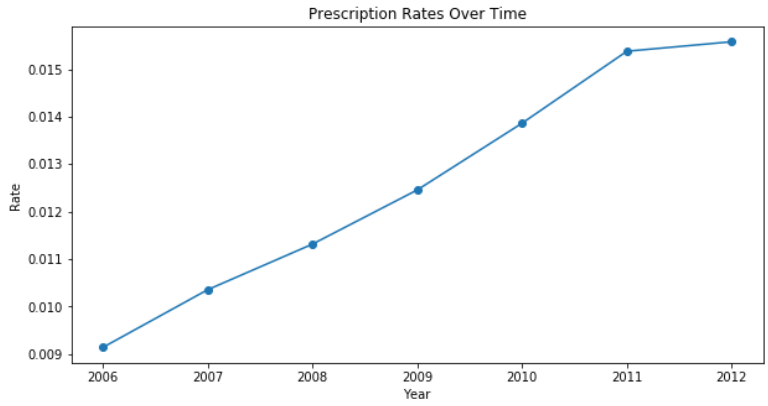
From these graphs we see a general tendency: rural and Southern states tend to have high rates of prescriptions and overdose. One potential explanation is that because these states tend to have lower economic development, people are more likely to abuse opioids as a means of escape.

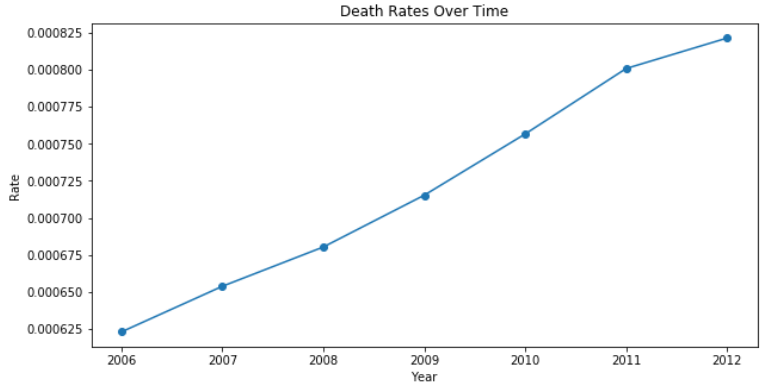
I created stacked bar graphs to show how the top states changed over time:



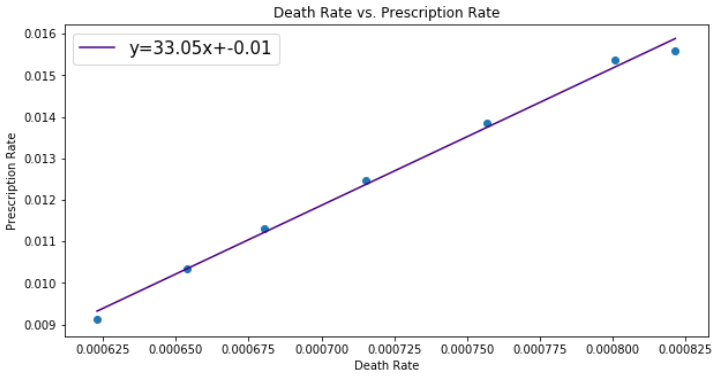


Both of these graphs (the second more clearly) demonstrate the rates increasing over time. Next, I wanted to see if this trend held true across the entire US, so I averaged the rates by year and created line graphs:

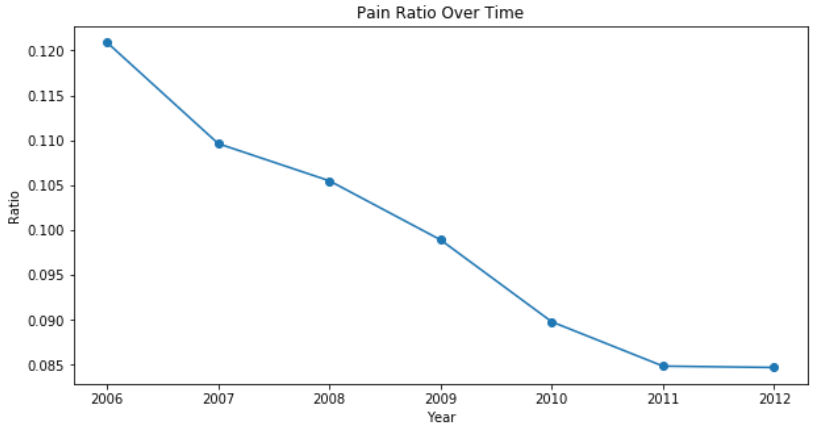




Having seen positive trends for both rates over time, I examined the relationship between death rate and prescription rate. I created a scatter plot of the aggregate values, then added a fit line:



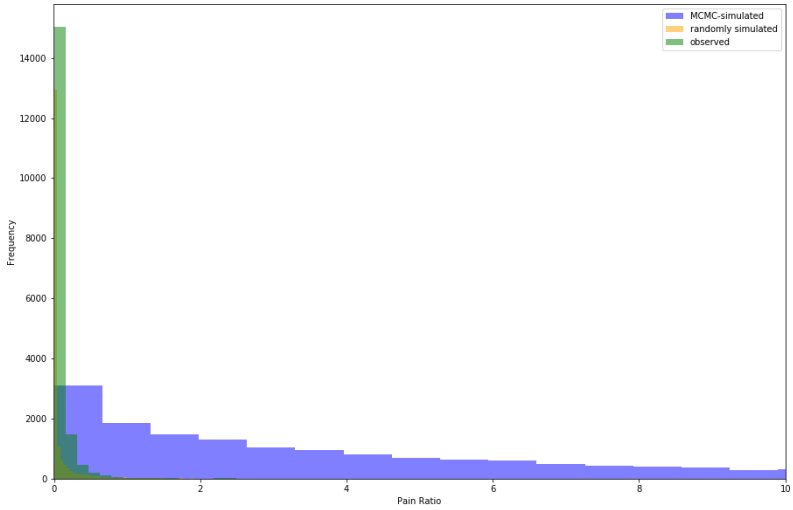
Here we can clearly see a linear relationship between the two rates. I also looked at how this ratio of rates, which I called the “pain ratio”, changed over time:



This shows that the pain ratio has actually decreased in this period, which I take to mean that even though prescription rates have increased, they have become less deadly over time. For me, this could be explained by the “Second Wave” the CDC identified - even though prescription pills are becoming less deadly, opioid overdoses are still rising because some became addicted to legal pills and subsequently made the switch to heroin.

b) Inferential Analysis

Based on the previous analysis, I began to think that pain pills may actually be less deadly than expected. To test this idea I compared the observed pain ratios to two simulations. The first was a basic simulation using random continuous variables from the same interval as the original observations. The second simulation was a Bayesian model which used the Markov Chain Monte Carlo method to create a distribution of likely values. Within the PyMC3 library, this method randomly generates a large number of samples from the original distribution, which tend towards values with higher probability. We can compare these models visually as well:



Plotting this Bayesian simulation shows a model with a much wider distribution than the observed data - in fact, less than 5% (4.7%) of the means of simulated pain ratios were as small as the observed mean. If the null hypothesis is that the mean of observed pain ratios are equal to the simulated mean of ratios, then the small p-value from the Bayesian model allows us to reject the null hypothesis: opioid prescriptions are not likely to be more deadly than observed.

c) Model Analysis

As mentioned before, finding the key features of the Lasso model we used for predictions is quite easy - simply find the features with non-zero coefficients. Unsurprisingly, population and prescription rate were among the most important features. We already saw the linear relationship between prescription rate and death rate, and places with urban concentrations may be more likely to have prescriptions or other illegal opioids. Other features included the black male population from ages 0 to 4, male Pacific Islanders ages 15 to 19, indiginous females ages 45 to 49, white males 85 and up, and the total population ages 80 to 84. Curiously, when examining the most important features of the linear model, the only common feature was male Pacific Islanders 15-19. In my opinion, this result does not indicate that these groups have the greatest risk of opioid overdose, but rather the model may be using these specific features to understand the overall demographics of the given county. The fact that our best models use demographic data and not county or state data tell me that opioid use and abuse is determined more by demographics than physical location. Rural and Southern states show similar prescription and overdose rates not because of their geographic similarities, but because of their demographic similarities.

5. Conclusion

The model we constructed was able to predict death rates (using death counts) at about 80% accuracy. Analysis of trends for prescription and overdose rates showed at least some evidence of the “Second Wave” the CDC identified, in which those with opioid addictions began to switch from legal prescriptions to illegal drugs like heroin. Prescription rates and population were both important features for predicting overdose risk, and demographic data certainly played a part as well.