Capstone Project 1: Final Report - Valery Lynn

Title: Predicting drug overdose mortality rates by county level in the U.S.

Problem:
According to data collected by the Centers for Disease Control and Prevention (CDC) there were more than 63,000 drug overdose deaths in 2016. More than 66% of those involved an opioid. On October 26, 2017 the opioid crisis was officially declared a national Public Health Emergency under federal law. The economic burden of prescription opioid misuse is estimated by the CDC to be more than $78 billion a year.

County level services such as hospitals, crisis centers, and local planning boards are in need of predictive models to inform planning, preparation, and resource allocation. This model can be used to better estimate the needs of the county to address this crisis. Examples include how many overdose kits hospitals need to have on stock, how many full-time crisis prevention professionals to employ, how many drug rehabilitation centers need to be established or funded, etc. Local employers are also stakeholders in this crisis. Reports suggest that companies in regions of high opioid usage are unable to maintain employees that can pass prerequisite drug tests for employment. Results from this study can be used as a trends indicator for able local workers.

Knowing what key variables play a role in predicting drug overdoses, and how those may vary across counties having different rates of overdose, can help counties invest wisely in therapeutic resources. Examples would be that a set of counties having particular predicted conditions would require a different set of responses than a set of counties with another set of predicted conditions, and a model could optimize (or customize) those responses. An example of this would be that counties with high overdose rates would see better results by increasing the number of mental health doctors while counties with moderate overdose rates would see better results by encouraging more job growth. Or, that the coefficients of the same set of variables would differ, requiring differing intensities of responses. For example, a new group home vs a new mental hospital.

The above accounts for responsive planning and action, but a model like this can be useful for preventative action as well. Policy decisions can be informed by models such as this where key predictors can point to strategic areas regarding investment in infrastructure for education, social services, and economic growth, should those sectors prove to have a positive outcome on reducing drug overdose mortality.

Finally, nearly half of the counties in the U.S. have missing data for drug overdose mortality in the publicly available databases. This gives an incomplete picture of the crisis at a national level, and hinders counties not reporting this data from taking preventative and curative actions. This study focuses on estimating those missing values for a more accurate (complete) visualization. Good models for prediction don't necessarily have to provide good variable explanations. In this study I put more emphasis on prediction. However, I discuss how much faith we can place in explanation (the effect of individual variables on the response), and offer suggestions for improving the model for more refined explanatory power.

Acquiring Data:
County level data of drug overdose mortality rates and social, economic, and demographic indicators are available for this model from the The County Health Rankings dataset, a collaboration between the Robert Wood Johnson Foundation and the University of Wisconsin Population Health Institute.This database was built predominantly from the following: The Behavioral Risk Factor Surveillance System (BRFSS), the National Center for Health Statistics, and the CDC WONDER mortality data.

The database for the rankings is available for downloading as an Excel spreadsheet at:
http://www.countyhealthrankings.org/explore-health-rankings/rankings-data-documentation

For a full account of how data for each variable was attained see:
http://www.countyhealthrankings.org/sites/default/files/resources/2017_Measures_DataSourcesYears.pdf

Variables are all reported as percentages, rates per 100,000, ratios, or similar population adjusted measures. After data cleaning, they are all of numerical type float or integer. Output data will be in the form of a float for regression analyses, and character for classification and clustering algorithms.

The following is a table with sample input and output variables and their selected summary statistics.

Table 1: Sample input and output variables with summary statistics.

Variable Name	**Description**	**Min Value**	**Max Value**	**Mean Value**
Output: Drug Overdose Mortality	Deaths per 100,000 population.	3	93	18.16
Input: Mentally Unhealthy Days	**Health Outcomes:** average number of mentally unhealthy days reported in the past 30 days.	2.3	5.8	3.78
Input: High School Graduation Rate	**Economic Environment:** percentage of ninth-grade cohort that graduates in four years.	30	100	86.28
Input: Housing Problems	**Physical Environment:** percentage of households with at least 1 of 4 housing problems: overcrowding, high housing costs, or lack of kitchen or plumbing facilities.	1	62	14.47

This spreadsheet contained both raw and ranked data. I am interested in raw data for the analysis so I saved two tabs (Excel worksheets) of data as *.csv files: ‘Ranked Measure Data’ and ‘Additional Measure Data’. These needed to be joined using Pandas merge function. Before joining the dataframes, I removed columns from the csv files that contained counts (rather than rates), confidence intervals, and any other columns containing calculated measures.

The data were contained in two separate csv files. These were merged with an 'inner join' method. While I could perform the join on one column, 'FIPS' (the Federal Information Processing Standards code (FIPS) which uniquely identifies counties and county equivalents in the United States), I also joined on 'State' and 'County' names in the event that the county code was erroneously entered. This assured that the join would be correct. I chose an inner join because I wanted to avoid missing data as much as possible and I didn’t want any duplication in the ‘State’ and ‘County’ columns.

The dataframe with additional measures initially had 3136 rows and 38 columns. Each row is a county in the U.S. The dataframe with rank measures initially had 3136 rows and 39 columns. After the inner join, the dataframe had 3134 rows and 74 columns. The reduction of 2 rows was due to discrepancies between the dataframes (so they were dropped in the join) and the reduction of 3 columns took into account that the join was executed on 3 columns. This left values reported as rates, percentages or ratios as functions of county population. The predicted variable (y-variable) is drug overdose mortality as a rate (per 100,000). Approximately half of the counties did not report this rate. I removed all the rows (counties) that had a missing values for drug mortality rate, the y-variable. These will be estimated by the model. After dropping all missing values for the y-variable, there were 1623 rows and 74 columns. Finally, I converted the ‘FIPS’ column to strings because the codes are categorical and cannot be treated as numeric.

There were four columns ('PCP Ratio', 'MHP Ratio', 'Other PCP Ratio', ‘Dentist Ratio’) that were reported as ratios in the format ####:##:## or #####:#. I did a string split on the first ':', then converted the columns from strings to numeric (float64).

I chose to impute missing values using state medians after an analysis of the data. More than ⅓ of the variables had differences between the mean and median greater than 1 standard deviation, with some differences that were quite large. This suggests that for some variables, the distributions are skewed, or not normal, and therefore the median is a more robust measure of center. I chose to impute with state medians rather than column medians (U.S. medians) to give more predictive power to the model by preserving as much of the variability of the feature variables as possible.

To do this, I created a new dataframe grouping by state and calculating medians for each variable. This dataframe had 51 rows, one for each state. I created a dataframe with just the column of state names from the original dataframe (1623 rows, 1 column). I did an outer join of these two dataframes, creating a new dataframe with 1623 rows and 73 columns called df_meds. When I created the dataframe with state medians it ignored the 'FIPS' and 'County' columns as they were not numeric. These were later inserted back into the joined medians (df_meds) dataframe. I checked for missing values after this join with the following command:

```
df_filled[df_filled.isnull().any(axis=1)]
```

It returned four rows that still had missing values. This was because South Dakota (with only 3 counties included in the dataframe after the missing y-variable rows were dropped) had no values reported for one or more of the feature variables and therefore had no state medians for that variable(s). Likewise, the District of Columbia (1 row) had missing values. For those, I used column medians to fill missing data with U.S. means across all feature variables.

To impute the missing values in the original dataframe, I combined the original dataframe with the means dataframe with the following command:

```
df = df.combine_first(df_means)
```

I checked for missing values and there were none.

Finally, I plotted boxplots and probability plots to check for outliers. Boxplots provide good visuals for detecting outliers and probability plots check if outliers belong to the distribution. Many of the variables had outliers that were shown by the boxplots and the probability plots. In addition, the plots showed that not many of the distributions were normal. The following section describes how I attended to this.

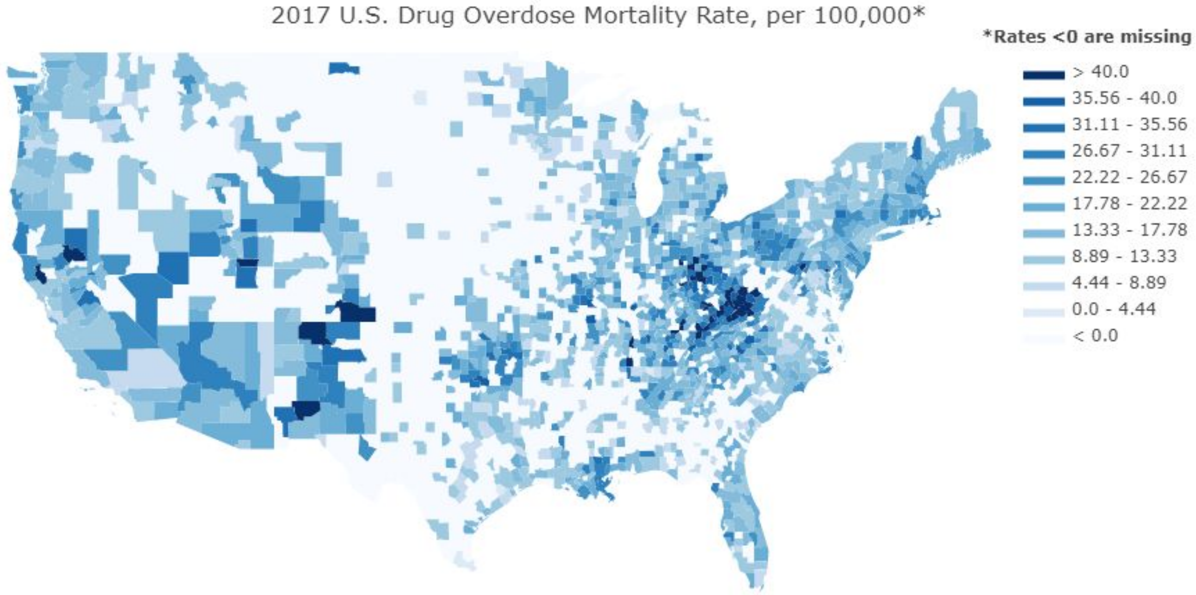
The Data Story:

My first task was to visualize the incidence of drug overdose mortality by creating an interactive cloropleth (heat) map of the U.S. with counties shaded to represent drug overdose mortality rates.

It was obvious that a large portion of the map was unshaded due to missing values for drug overdose mortality rates. Upon inspection there was more than half of all counties with missing values. This study will answer two framing questions:

1. How can we best estimate missing drug overdose mortality rates using supervised machine learning algorithms?
2. What are the principal predictors for drug overdose mortality rates?

Figure 1. Heatmap of drug overdose mortality rates in the U.S.



The best way to estimate a predicted quantity from a set of quantitative continuous data is with linear regression analysis. I started by considering a linear model because linear models can capture a great deal of data relationships. The goal of linear modeling is to use the simplest model that captures the most variation. My next task was to begin shrinking my data by finding only those predictors that have an influence on the response variable. I began by looking at how each predictor correlated to the response variable. The Pearson-r correlation test assumes that variables are normally distributed. To get the most accurate results I needed to check the shapes of the distributions for each variable and use transformations to get them to a normal distribution if possible.

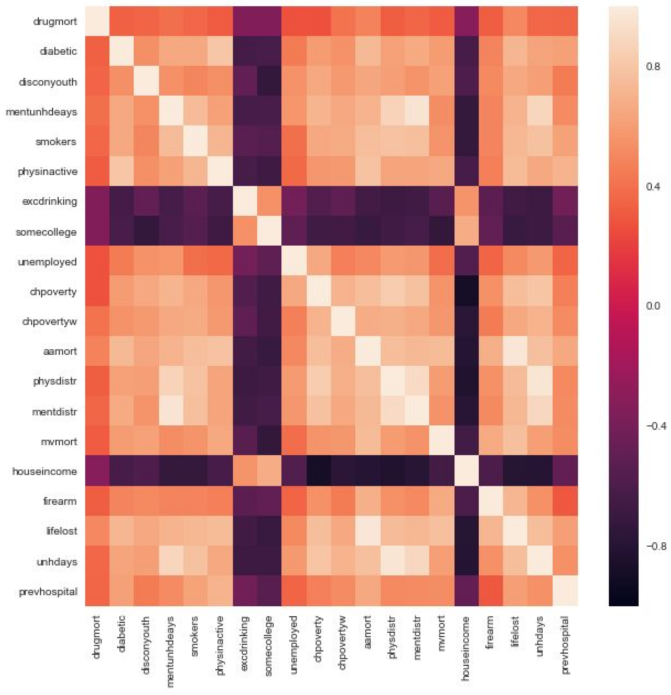
I started by examining the distributions of all the variables by plotting their kernel density curves and probability plots to check for normality. It appeared that many of them were normal or log-normal, including the response variable. Taking the log of a log-normal distribution leaves a normal distribution. I did this and ran the probability plots again to check for normality. Not all of the variables could be transformed this way, but there are other methods later that can be used such as a ridge regression with built-in regularization and shrinkage.

I ran pearson-r correlation tests on the response vs all other variables and selected any that were +/- 0.3 or higher. This left 20 variables that demonstrated considerable relationships with the response variable. A description of these variables, along with their Pearson-r coefficients and p-values can be found in Table 1 of Appendix 1. These were not the final variables used in the model, but this was a way to determine if linear modeling would be a good place to start with this data. Linear modeling works well with variables that are correlated to the response. Having at least 20 variables with decent correlations is good justification for starting with a linear model.

It is important to note that it is possible to transform a response variable for linear modeling but it is usually better not to. It greatly reduces the ability to make predictions because there is no straightforward way to scale back to the original (back-transformed) values that can be interpreted on the original scale. You cannot compare regression coefficients in models that have transformations performed on the response variable(Faraway, 2014). The exception to this is with log-normal response variables. In this study, it was determined using the Box-Cox method (done in R) that the response variable required a log transformation (see Appendix 2 for the results of the Box-Cox method), also Faraway, 2014 p. 134.

Major Findings:

The 19 variables correlated with drug overdose mortality begin to tell a story about counties as drug mortality rates rise. However, there are a few things to note before drawing conclusions. Several of these variables are likely collinear. Meaning that predictors are correlated to each other. For example, poverty rates will decrease as median income rises. It may not be necessary to include both of these as they are similar measures. Several variables are measures of physical health (diabetic, smokers, physinactive, physdistr) and are likely to be correlated. Ridge regression is a good model to use if there are many variables that are thought to contribute to the response, and where there may be collinearity between covariates (Faraway, 2014). Below is heatmap plot of the correlation matrix:



There are three predictors that are negatively correlated with drug overdose mortality: excessive drinking, household income, and some college education. Income is usually positively correlated to college attainment so it is not surprising that they would both have the same direction in the effect.

Regression analysis and prediction

Choice of algorithms:
This is a fairly large data set that has a great deal of collinearity (features that correlate to each other). Ridge regression is well-suited for dealing with collinearity. It also performs well when there are many features that are believed to have an effect on the response, as is the case in this study.

Although Lasso regression is also used for regularization, it is best when the effects are sparse. Meaning, when it is believed that only a small number of features have an effect on the response and the rest do not. The EDA found that at least 19 features had moderate to high correlations with the response so in this case ridge regression is better suited.

For ridge regression, all features and the response were centered by their means and scaled by their standard deviations. Sklearn does this with the "scale" function.

The standard score of a sample x is calculated as:

$$z = (x - u) / s$$

where u is the mean of the training samples or zero if with_mean=False, and s is the standard deviation of the training samples or one if with_std=False (from Sklearn documentation).

The ridge regression chooses the *β* that minimizes:

$$(y-X\beta)^T(y-X\beta)+\lambda\beta^T\beta$$

λ is a (>= 0) user-set value determined in this study below by plotting the cross-validation score (CV) against several alpha values and choosing the one that optimizes the cv score (this equation uses λ , scikit-learn uses α) . β is a vector of weights, sometimes referred to as the coefficients of regression, βi .

The ridge regression estimates of βi are given by the function:

$$\beta\hat{} = (X^TX+\lambda I)^{-1}X^Ty$$

I also tested an ordinary least squares model (OLS). It is possible that the degree of collinearity won't be so high that it negatively impacts the model. The major goal is prediction so it will be easy to test the ability of each model to predict using cross-validation. Maximizing explanation, knowing the degree to which each feature contributes to the response, requires careful feature selection. This is beyond the scope of this study. However, I will discuss further refinements of the models that could lead to better explanations. In general, the OLS looks like this:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots \beta_n X_n + \epsilon$$

That being said, the EDA demonstrated that some variables were not highly correlated to the response. I will test both models on data where log-normal features have been transformed, and I will create a dataset with only those variables that are shown to have an influence on the response.

The model then becomes:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 \log X_2 \ldots \beta_n X_n + \epsilon$$

Finally, I used the "step" function in R to determine the optimal size of the model for ordinary least squares (ridge regression has its own method of shrinkage built in). The step function in R sequentially drops features that are not significant, and measures the Akaike Information Criterion (AIC). The best choice is the model that minimizes the AIC. The “step model” that resulted shrank 74 features to 34 features, leaving a model where all features were significant.

The following table shows the six different models to be tested:

Table 2. Six Regression Models for Prediction Testing

#	Regression Type	Transformation	Number of Features
1)	Linear Regression (OLS)	None	Full Model
2)	Linear Regression (OLS)	None	35
3)	Ridge Regression	None	Full Model
4)	Linear Regression (OLS)	Log*	Full Model
5)	Linear Regression (OLS)	Log*	35
6)	Ridge Regression	Log*	Full Model

*Log is taken only on log-normal variables.

The Box-Cox function (in R) determined that it is appropriate to take the log of the response variable (see Appendix 2). In most cases, transforming the response variable is discouraged because it creates problems with interpretability. However, the one exception is with a log transformation (Faraway, 2014).

The final OLS model becomes:

$$\log Y = \beta_0 + \beta_1 X_1 + \beta_2 \log X_2 \ldots \beta_n X_n + \epsilon$$

Results:

Many methods exist that are used to assess the quality of a model. Most common, and the ones I use here, are the R-Squared value (or Adjusted R-Squared for models with more than one feature), the Root Mean Square Error (RMSE), and the CV-score. The R-Squared value measures what proportion of the variation in the response variable can be explained by the features (predictor variables). In a model with many predictors, an R-Square value higher than .50 is often considered to be a well-fitting model. RMSE is the standard deviation of the residuals. RMSE is a measure of how spread out are the residuals. On scaled data, the RMSE is aligned with the correlation. The higher the correlation, the lower the RMSE value. This makes sense because a well-fitting model should have less spread around the predicted line. Finally, the CV-score is a measure of how accurate the model performs in prediction. It is percentage of observations that were correctly predicted.

Table 3. Regression results for six models

#	Regression Type	Adjusted R^2 and Root Mean Square Error (RMSE)	Average 10-fold Cross-Validation Score (CV)
1)	Linear Regression (OLS) - Full, untransformed	**Adjusted R^2**: 0.5923868474405766 **RMSE**: 0.7256769706179685	**CV:** 0.3582203997916027
2)	Linear Regression (OLS) - Step reduced, untransformed	**Adjusted R^2**: 0.5739796408675106 **RMSE**: 0.7100274761313532	**CV:** 0.4239415366434971
3)	Ridge Regression - untransformed	**Adjusted R^2**: 0.5736075788576231 **RMSE**: 0.7210173693911791	**CV:** 0.3837487217984189
4)	Linear Regression (OLS) - Full, log-transformed	**Adjusted R^2**: 0.5809871211269844 **RMSE**: 0.6727687906026559	**CV:** 0.391952652638861
5)	Linear Regression (OLS) - Step reduced, log-transformed	**Adjusted R^2**: 0.56778520526356 **RMSE**: 0.6490841781141685	**CV:** 0.4530938467232424
6)	Ridge Regression - full, log-transformed	**Adjusted R^2**: 0.5685524777888835 **RMSE**: 0.6680830771598685	**CV:** 0.414709583385758

Conclusions:

The best R-Square score was with the linear regression model on full, untransformed data, but the cross-validation was low, predicting with only a 35.8% accuracy. The best RMSE score and the best CV-score were obtained with the linear regression model with the step-transformed data (34 features). Because the purpose of this study is to predict missing values, the linear regression model with the step-transformed data is the best choice. This model will be used to predict the missing values and produce a completed, estimated map. But first we should check that the assumptions for linear regression are met. The final check of this model is to look at the residuals and Q-Q plots (in R) to check for constant variance and a normal distribution.

Figure 2. Fitted values against residuals showing constant variance.

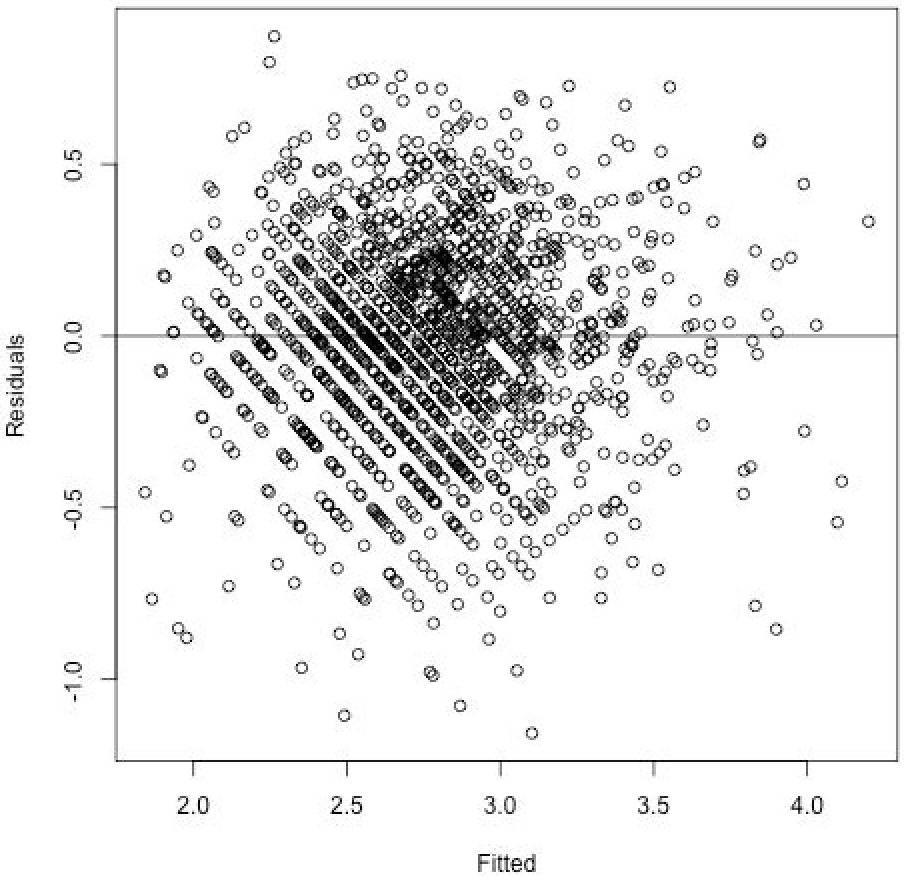
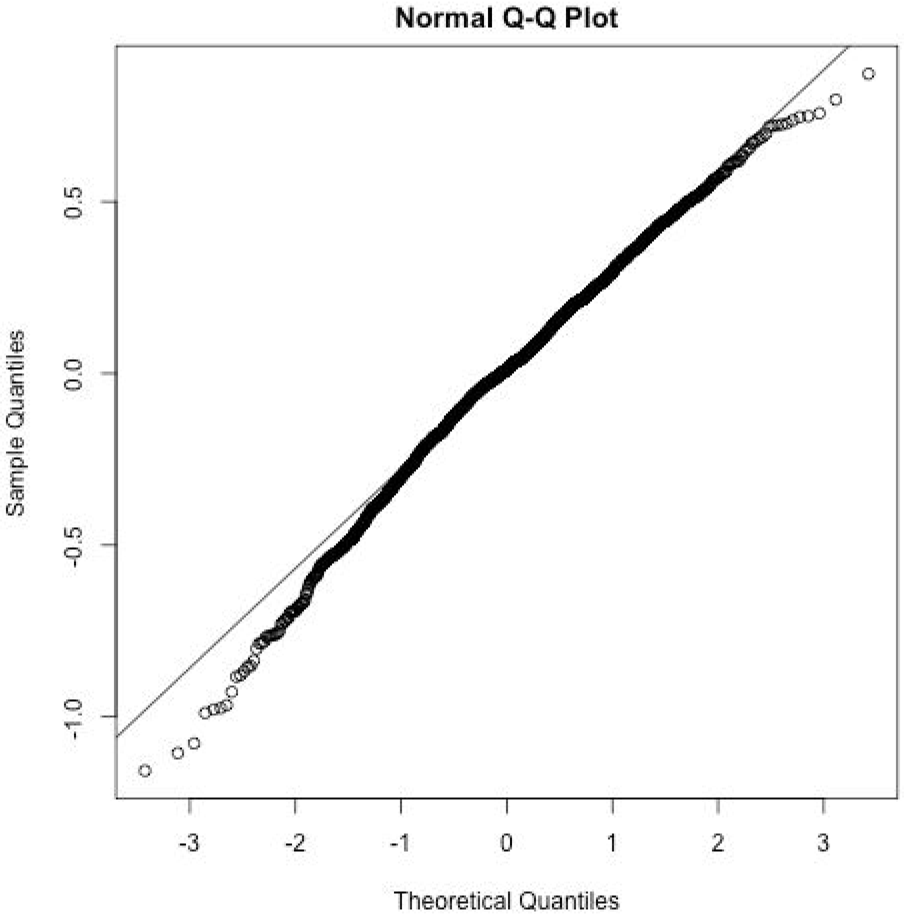


Figure 3. Q-Q plot showing that the residuals are distributed normally.



Prediction:
This model now becomes the final model (called ‘model_step’) used to predict the missing values. I used the following code to train the final model on the step-transformed data set with 34 features (Xst = matrix of features, Yst = the response array):

```
#Train the final model on testing data.
model = LinearRegression()
model_step = model.fit(Xst, yst)

# Predict on the test data: Ypred
Ypred = model_step.predict(Xpred)
```

Ypred is an array of predicted values.
The predicted values were then unscaled and un-transformed by using the following code:

```
#Unscale the predicted values
```

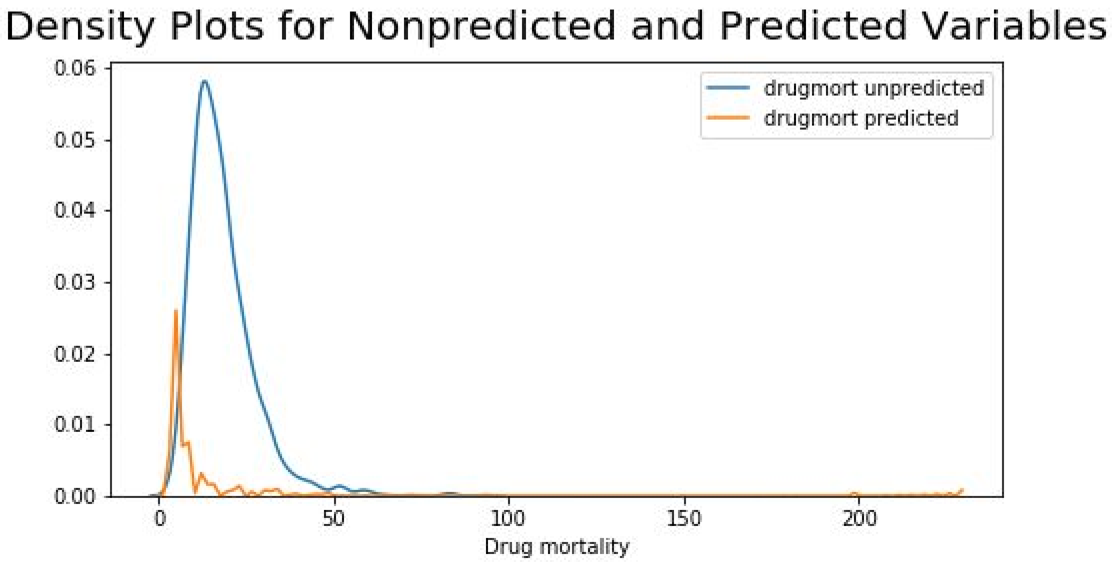
```
mean_of_array = Ypred.mean(axis=0)
std_of_array = Ypred.std(axis=0)

Yunscaled = (Ypred * std_of_array) + mean_of_array

#Unlog the predicted unscaled values
predictedY = np.exp(Yunscaled)
```

I plotted the distribution of these values, along with the distribution of the non-missing values for drug overdose mortality. My hope is that they would be similar, considering that both datasets were approximately the same size. Below are the kernel density plots for the training and the predicted values.

Figure 4. Kernel density plots for the predicted and unpredicted values for drug overdose mortality rates.

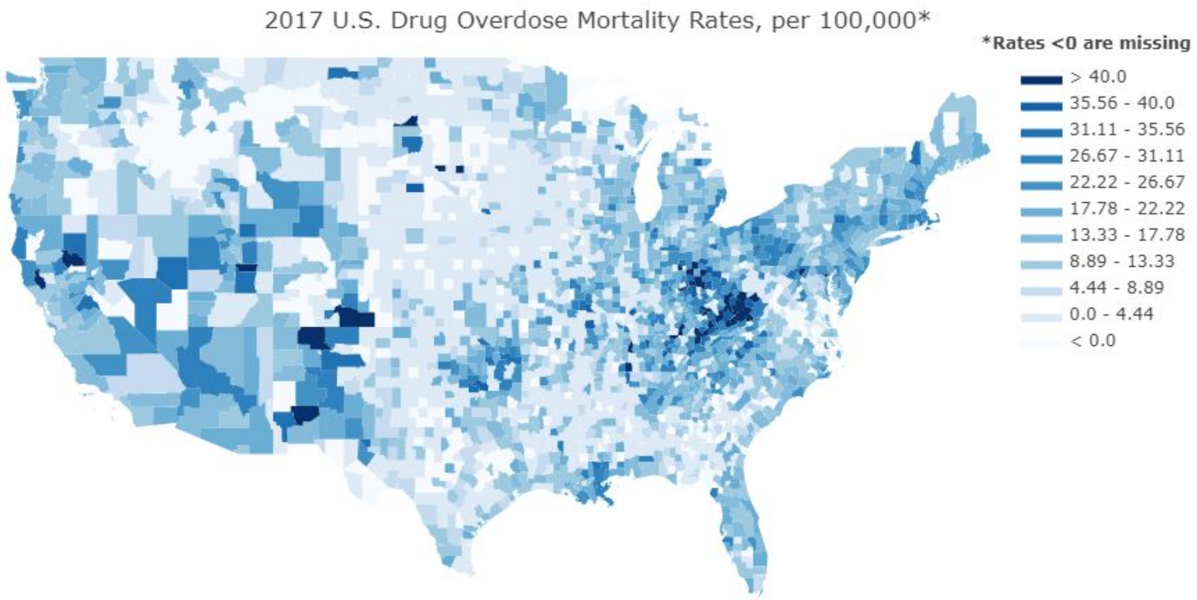


You can see from this that the shapes of the plots are similar, indicating that model performed adequately in predicting the variability in the response variable. The shape of the distributions were both right skewed with most of the data near the lower end of the values. However, the model made predictions with more values at the extremes. This is not helped by having 377 predicted values that are zero. The standard deviations are similar but the predicted values have a lower mean and much higher maximum values. There are more outliers in the predicted values. This means that most likely the predicted values will be either lower or higher than the real values would be had they been reported.

There are several things that could be done to improve this model. The first would be to include interaction terms, which were not tested in this study. The second thing would be to consider generalized linear regression models that may better handle this data, as it included variables that were not normally distributed. The last would be to include the Elastic Net in the group of tested models, which uses methods from both Ridge and Lasso regressions.

Going forward, the predicted values were joined with the predictor values to recreate the test data set. This was joined with the training set and finally, the heatmap was created using both reported and predicted values.

Figure 5. Heatmap of drug overdose mortality rates in the U.S. with predicted values.



There are still some values showing as zero. But it is an improvement! Care must be taken in interpreting these results. But the process for predicting missing data this way is promising.

References

FARAWAY, J. J. (2014). *Linear Models With R, Second Edition*. Taylor & Francis.

Appendix 1

Table 4. Variable names and descriptions with Pearson-r correlation coefficients and p-values.

Name	Description	Pearson-r	p-value
drugmort	Drug overdose mortality rate	1.0	0.0
diabetic	Percentage of population that is diabetic	0.3249	3.1650e-41
disconyouth	Percentage of teenagers and young adults between the ages of 16 and 24 who are neither working nor in school.	0.3662	1.1146e-52
mentunhdeays	Average number of reported mentally unhealthy days per month	0.4029	2.1010e-64
smokers	Percentage of adults that reported currently smoking	0.3915	1.2912e-60
physinactive	Percentage of adults that report no leisure-time physical activity	0.3168	3.4893e-39
excdrinking	Percentage of adults that report excessive drinking	-0.3687	1.8666e-53
somecollege	Percentage of adults age 25-44 with some post-secondary education	-0.3381	1.0696e-44
unemployed	Percentage of population ages 16+ unemployed and looking for work	0.3077	5.9276e-37
chpoverty	Percentage of children (under age 18) living in poverty	0.3030	7.8189e-36
chpovertyw	Percentage of white children (under age 18) living in poverty	0.4322	7.2575e-75
aamort	Premature age-adjusted mortality	0.5067	1.3829e-106

physdistr	Frequent physical distress (measured through the Behavioral Risk Factor Surveillance System)	0.3659	1.3131e-52
mentdistr	Frequent mental distress (measured through the Behavioral Risk Factor Surveillance System)	0.3711	3.6176e-54
mvmort	Motor vehicle crash mortality rate	0.3289	2.9953e-42
houseincome	Median household income	-0.3037	5.5081e-36
firearm	Firearm Fatalities Rate	0.3429	5.2499e-46
lifelost	Years of potential life lost rate	0.5277	4.4739e-117
unhdays	Average number of reported physically unhealthy days per month	0.3996	2.7472e-63
prevhospital	Preventable hospital stay rate (Discharges for ambulatory care sensitive conditions/Medicare enrollees * 1,000)	0.4291	1.0011e-73

Appendix 2

Results from the Box-Cox method in R:
The following is code in R for a linear regression on all the features, untransformed, in the cleaned training set. It is used with the Box-Cox method to determine if transformation of the response variable is warranted.

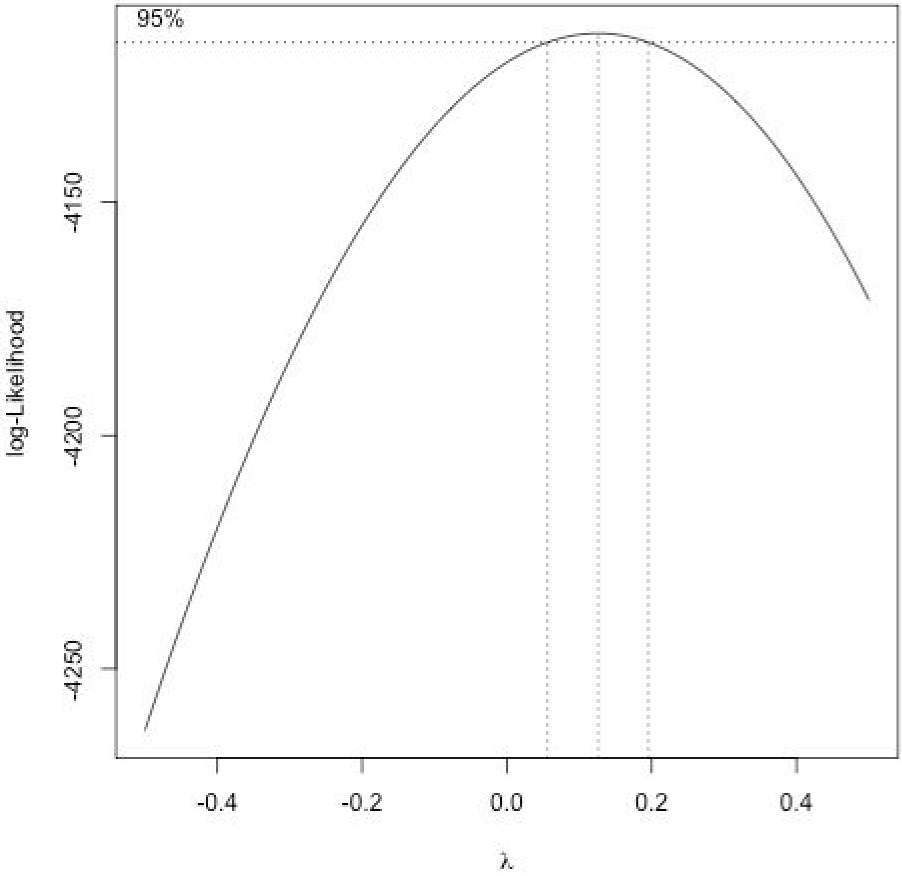
```
#Load the file from the csv file.
df_cl <- read.csv(file="VLynn_DrugOverdose_cleaned.csv", header=TRUE, sep=",")

#Take out non-numeric columns
df_cl <- df_cl[,-c(1:5,39)]

#Run a linear regression model with all features against the response variable, drugmort.
lmod <- lm(drugmort ~ ., df_cl)
```

```
#Run the Box-Cox method
require(MASS)
boxcox(lmod, plotit=T, lambda=seq(-.5, .5, by=0.1))
```

Figure 6. Box-Cox plot showing maximization of the log-Likelihood around 0.1.



The boxcox test shows a lambda around 0.1. This is very close to zero which would support taking the log of the response variable (See Table 2 below).

Table 5: Box-Cox Lambda values and transformations of response variable.

Lambda	Standard Transformation
-3	Inverse Cube
-2	Inverse Square
-1	Inverse

-0.5	Inverse Square Root
0	Logarithmic
0.5	Square Root
1	No Transformation
2	Square
3	Cube