Capstone 1 Data Wrangling: Valery Lynn

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

Acquiring Data

The data were acquired from The County Health Rankings dataset, a collaboration between the Robert Wood Johnson Foundation and the University of Wisconsin Population Health Institute, for the year 2017. The database for the rankings is available for downloading as an Excel spreadsheet at:

http://www.countyhealthrankings.org/explore-health-rankings/rankings-data-documentation

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, 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. means) to give more predictive power to the model by increasing the variability of the feature variables.

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 all of the distributions were normal. After performing a principal component analysis (PCA), many of these feature variables may be eliminated before fitting a model. It is reasonable to wait before addressing these outliers until a preliminary attempt has been made to fit a

model. If there is difficulty fitting a model, the first thing I will attend to is removing any data points that appear to belong to a different distribution (severe outliers).

The dataframe was now complete and I sent it to a csv file for EDA and preprocessing.