My Equation:

y = -3.95(x1) -2.29(x2) -0.62(x3) + 1.65(x4) + 40

Where x1 = legor\_so; x2 = latent\_modernization; x3 = latent\_earlydev, and x4 = sd\_index\_ext

How I got it:

The first step I took was deleting every empty row in the dataset; this left me with around 200 countries left to work with. At this point, rather than imputing, I used the dropna function to deal with any missing data.

The next step I took was to find the coefficient of each continuous independent variable with gini score by running a respective regression for each variable. I took note of each variables whose coefficient was above 0.01 (which is quite low, but I wanted to further narrow down the coefficients with other methods). I repeated the same process for each discrete variable.

Next, I made a correlation matrix containing the measure of correlation of each continuous independent variable using the correlation matrix function in pandas. I took this step to test for possible multicollinearity among the variables. I received the following matrix: A screen shot of a chart

Description automatically generated

I checked each variable with a, previously measured, high coefficient that showed high correlation with other variables to identify possible candidates of variables to predict gini scores, disregarding the variables which are colinear to it.

Next, I made a similar correlation matrix using a point bi-serial correlation measure to include the binary variables and the continuous variables chosen from the previous step (previously I ran the same code except with a phi correlation with only binary variables and found that ‘federalism\_GT’ could be removed): A screenshot of a computer screen

Description automatically generated

This matrix prompted me to drop all of the variables corresponding to continents because it seemed as though they are mostly colinear with the other independent variables chosen. Thus, I was left with 5 variables in my final regression, as seen below:

A graph with blue dots and red line

Description automatically generated

After my results, I decided to use an imputation method to account for previously dropping rows with the dropna function. Specifically, I used MissForest, which is based on RandomForest, that creates classification trees from subsets and uses a mean value estimate to predict an imputed value. This resulted in the following regression:

After seeing the relatively low coefficient for ‘BatDeath\_pc\_cum\_ln’, I decided to drop it and I received my final regression; it turns out that this change improved the R-square value:

Coefficients:

Variable Coefficient

0 legor\_so -3.950488

1 latent\_modernization -2.285144

2 latent\_earlydev -0.623263

3 sd\_index\_ext 1.649470

Intercept 39.99837041579912

Mean Squared Error: 29.27162497096421

R-squared: 0.6141273519327579

A graph with blue dots and red line

Description automatically generated