The Use of Machine Learning in Earthquake Prediction

Austin Clark, Timothy Lee, Andy Schleicher, Mingcan Tang, Aiden Yoon, Benson Wu

March 25th, 2019

Introduction

The overall goal of this analysis is to study the methods and procedures of predicting earthquakes. There are three meaningful aspects of earthquake prediction: when, where and how large the next earthquake will be. Specifically, we will focus on predicting when an earthquake will take place. The significance of predicting when an earthquake will occur is protecting infrastructure and civilians, while trying to better understand the earthquake process. The emphasis of this prediction will be on the time remaining before a simulated earthquake (laboratory earthquake) will take place from "real-time" earthquake data[1]. The data that we will use is taken from a prominent earthquake experiment. The data has two variables: acoustic data and time to failure. The acoustic data is the seismic signal or the amplitude of the earthquake. Time to failure data is the time (seconds) until the next simulated earthquake. The relationship between both sets is that acoustic data will be used to predict time to failure data[1]. Our aim is that statistical testing on this data will help improve the efficiency of predicting earthquakes.

A simulated earthquake is a laboratory experiment that attempts to recreate the earthquake cycle. The data is drawn from an intricate, physical model known as the classic laboratory earthquake model, which is used to replicate the earthquake process with the overarching goal of studying the physics of earthquakes[1]. It is important to note that this experiment does not exactly replicate the natural phenomena of an earthquake, but it does provide key information that is valuable to the study of earthquakes. According to the United States Geological Survey (USGS), the lab reproduces small earthquakes to better understand the physical nature of earthquakes. There are a few questions that are of interest: how does an earthquake begin? What happens when an earthquake initiates? Where does the energy from an earthquake go? Why does an earthquake stop? The optimal experiment would help to answer all of these questions[3]. This is another key reason why we are testing this data in order to improve the accuracy of predicting earthquakes.

There are multiple experiments that can be used when recreating an earthquake. The basic idea of these experiments is to confine a piece of rock that has been cut diagonally into two pieces, most likely granite, to a metal apparatus that will apply pressure in various ways. In some experiments, there are controlled conditions that are applied to the simulated fault that include high pressure, high water pressure, and high temperatures. Usually, there is material from the earth's surface that is spread out in between the diagonal gash of the rock. Then pressure is applied to the top and bottom of the rock where the fault is being observed while strain is applied. The diagonal cut is assembled with electronic sensors that measure slips, acoustic readings, and various other reactions from this experiment. This process is the recreation of the stick-slip failure along a fault. The stick-slip fault is one of many behaviors in the earthquake phenomena[3].

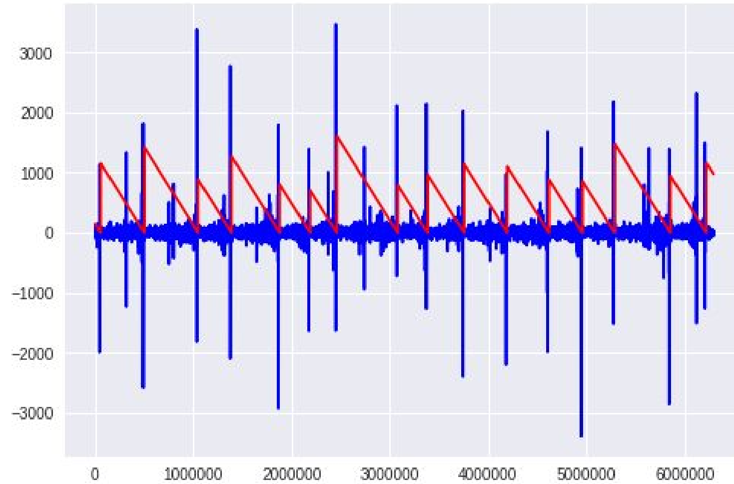
Many of these various earthquake behaviors are also studied in laboratory experiments, such as slow-slip behavior, which is related to low-frequency earthquakes. The low-frequency earthquake is associated with the slow earthquake. The slow earthquake refers to a fault slip with acceleration but not enough movement to create significant seismic energy. A regular earthquake differs from a slow earthquake by regular stick-slip failure properties[2]. Numerous experiments and data collection from regular earthquakes have provided a comprehensive understanding of how regular earthquakes operate. The same cannot be said of the slow earthquake, which can provoke detrimental regular earthquakes by transmitting strain[2]. If we can gain a better understanding of slow earthquakes, then we can become more efficient at predicting regular earthquakes. This falls in line with our goal of testing earthquake data to better our accuracy of earthquake prediction. In this report we hope to find useful statistical methods to optimize the way earthquakes are predicted.

Data Description

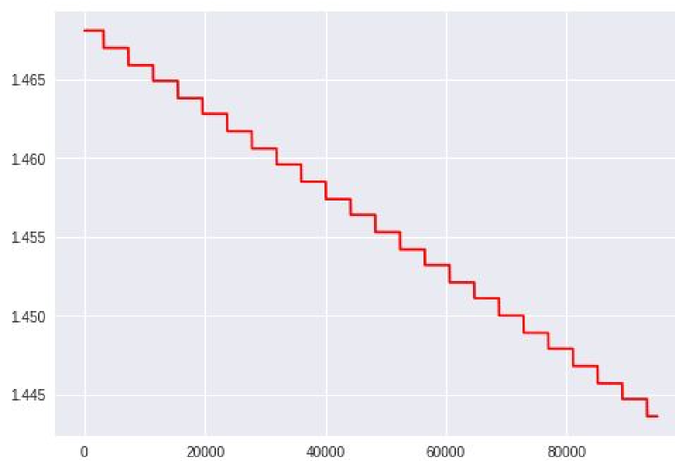
The data set used for this analysis is provided by a Kaggle challenge, hosted by Los Alamitos National Laboratory, in an attempt to find methods to accurately predict when laboratory simulated earthquakes would occur. We were given a training data set that contains 629,145,480 observations of two variables: 1) input acoustic signal 2) Time remaining until a failure (time until a lab simulated earthquake occurs). Due to memory limitations on our machines, we were not able to use the entire dataset for analysis and thus performed our analyses on small sets of the data. R and Python were used for this analysis. In the following sections, we perform basic distribution analysis to help us understand the data better.

Basic Analysis

To understand the data, we begin by visualizing the data in full scale. In the figure below, the blue lines represent acoustic signals, and the red lines represent the times to failure corresponding to the occurrence of each acoustic signal. This plot shows that there are in total 16 simulated earthquakes contained in the dataset, as represented by the 16 times that the red line hits 0. It is also noticeable that in all cases, earthquakes happen shortly after a significant acoustic signal (a spike) is recorded.

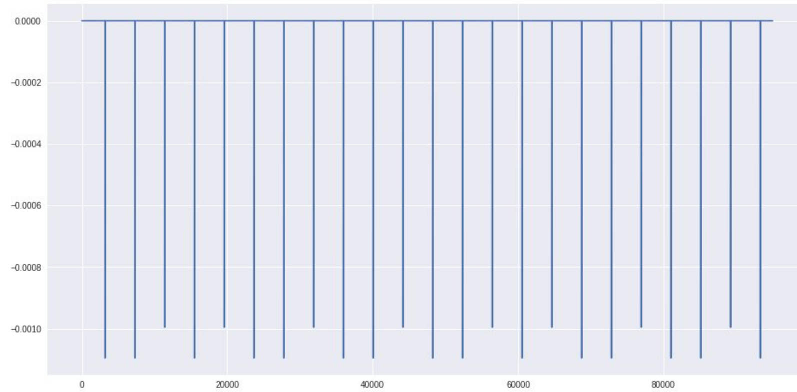


Having noticed that the acoustic data seems to lead to each occurrence of earthquakes linearly in time, we then proceed to understand the data on smaller scales. When we zoom in on the time to failure data on a decreasing time segment and investigate these values, we arrive at the following plot, where the 5000th to 100,000th time-to-failure values are plotted. The fact that the plot is, in fact, not smooth intrigues us. We are interested in understanding why this ladder-shaped descent occurs instead of a smooth and linear one. We observe that in this descent pattern, there always exists a regular, perhaps periodic, smooth stationary period (SP) followed by a radical descent (RD). Thus, we raise a series of questions concerning this pattern here and attempt to answer them below, which will help us better understand the organization of this data set.



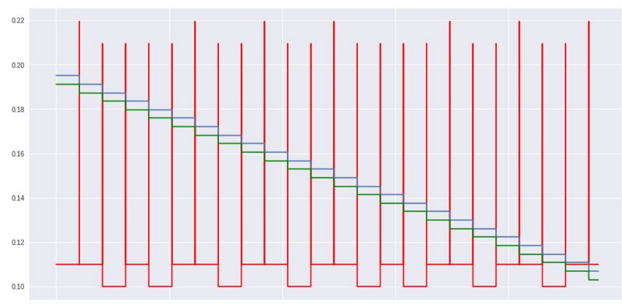
Q: Are the radical descents in time-to-failure values random?

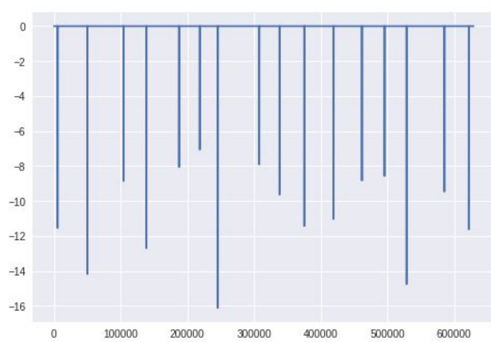
To attempt to answer this question, we plot out only the values of the radical descents as follows. Here we observe that the descent values seem to take on only two values, namely around -0.0010 and around -0.0011. We do not know whether this occurs simply due to the locality of the samples we choose to plot here, or due to other reasons. Thus, to answer this question, we visualize 500,004,100th to 500,104,100th data in the same manner, which turns out to follow the same identical pattern with descents having the same values. We thus conclude that the same radical descent pattern in time-to-failure values is present throughout this data set, and is thus likely not random. We proceed with further analysis on the stationary period.



Q: Does the lengths of the stationary periods follow a certain distribution?

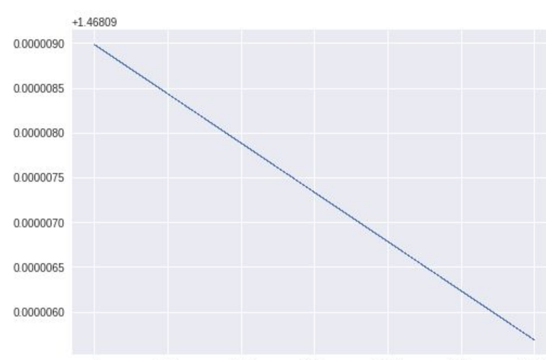
With some experimentation, we come to the conclusion that the stationary periods are 4100 data long throughout the entire dataset. This is locally demonstrated in the plot on the right, which represents the difference between the original data and the data with a 4100 lag, and globally illustrated in the second plot, which represents the same difference throughout the entire data set. The spikes that we observe in the second plot correspond to the beginning of the time-remaining countdown for each of the 16 simulated earthquakes.





With this knowledge, we understand that the length of the stationary period is not random, and was likely intentional when this data set was created. We continue our analysis by further zooming in on the stationary periods.

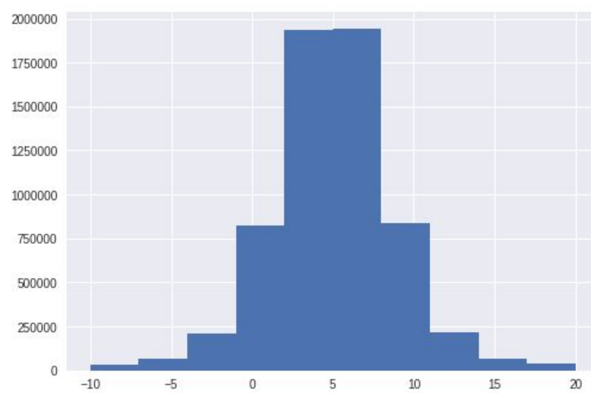
So far, the plots have informed us that, during the stationary periods, the time-to-failure values stay constant for 4100 data points until a radical descent occurs. To further understand this, we zoom in to one specific stationary



period, the 5000th to 8000th data, and plot the values on the right. Contrary to our original understanding, we see that the values within one stationary period is in fact not constant. Instead, they decrease slowly and linearly.

With the analysis above, we conclude that the original raw data from the experiments consist of a continuous decrease of time-to-failure values that lead to 0 each time a simulated earthquake occurs. The raw experiment data was then truncated so that a window size of every 4100 data points represent a continuous experiment segment, segments are then concatenated together to form the data set that we now have. Now, after some extensive investigation on the time-to-failure variable, we turn our focus to the acoustic signal variable.

To begin our analysis on the acoustic data, we plot the histogram of the acoustic data on the entire dataset. The result visually resembles a normal distribution. We thus raise and attempt to answer a series of questions regarding the normality of the acoustic data. For brevity of this section, all theories concerning the normality tests used are summarized in the *Other Tests for Normality section*.

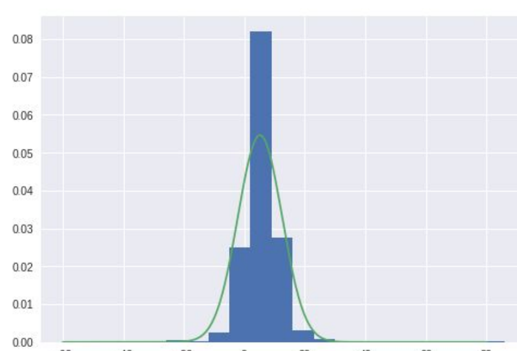


Q: Is the acoustic data normal?

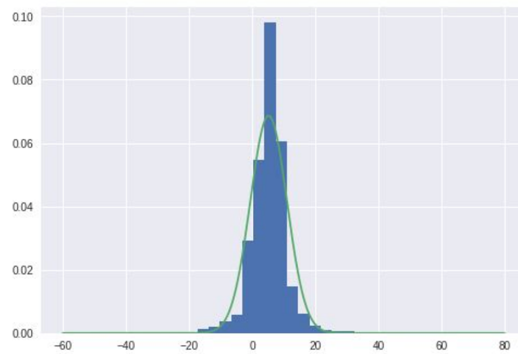
A K-S test between all acoustic data and the normal distribution with MLE-estimated means and variances yields a test statistic of 0.248 and a p-value of 0, prompting our rejection of the normality assumption. Further, D'Agostino-Pearson normality test yields a test statistic of 10,390,876 and a p-value of 0 as well. Thus, we conclude that the acoustic data of the entire earthquake dataset combined does not follow a normal distribution.

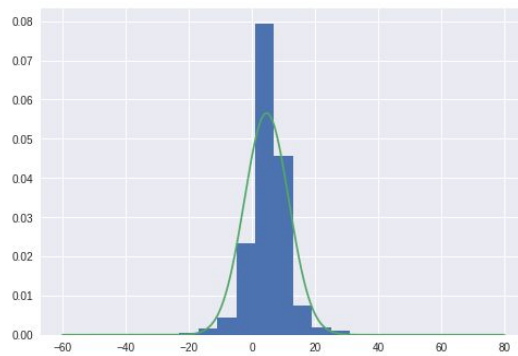
Q: Does the acoustic data in each continuous window follow a normal distribution?

Now that we have concluded that the entire acoustic data in the dataset does not follow a normal distribution, we further ask whether the acoustic data in each continuous 4100 window (of individual SP) follow a normal distribution. To that end, we plot the histograms of the first 2560,



4100th to 8200th, and 410,000,000th to 410,004,100th data, arranged from top to bottom. The green curves represent the theoretical pdf of a normal distribution with the MLE-estimated means and variances. Again, the visual resemblance to a normal distribution triggers us to hypothesize that each of these might be normal.





We thus perform a D'Agostino-Pearson normality test and a K-S test on each sample to test for normality. The test statistics and the p-values obtained in each case is summarized below (the p-values are all less than 1.0 * 10^{-10}).

sample	D-A test stats	D-A test p-value	K-S test stats	K-S test p-value
0-2560	1052	0	0.138	0
4100	425	0	0.089	0
410,00 4,100	901	0	0.120	0

When we plot the initial 2650 acoustic data, and based on the plot of acoustic data of the full dataset, we notice that there exist extreme values on the acoustic data. We thus hypothesize that the data might be normal without these extreme values. The D'Agostino-Pearson Normality Test again shows p-values of 0 for the above three samples. (Other tests used for normality here are included in the *Other Tests for Normality Section*.) Thus, we can safely conclude that the acoustic samples are not normally distributed.

Q: Does the distribution of the acoustic data stay invariant over time?

Now that we have understood that the acoustic data does not follow normal distribution, we observe that the distribution of the above three sets of samples visually resemble each other. We thus inquire into whether the distribution are identical over time. The following table summarizes the K-S two sample test results of samples taken from different times in the original data.

Samples		K-S test statistic	K-S test p-value
1-2,650	4,101-8,200	0.0165	0.768
4,101-8,200	41,000-45,100	0.0317	0.032
4,101-8,200	41,000,000 - 41,004,100	0.1298	1.443e-30
4,101-8,200	410,000,000 - 410,004,100	0.0544	1.0113e-5
4,101-8,200	615,000,000 - 615,004,100	0.0970	2.699e-17

Based on the above table, the p-values are very small for data that are relatively far away. However, we further analyze the distribution of the acoustic when normalized. The above analysis is only performed on the original acoustic data. We hypothesized that it is possible that certain damping or amplifying effects might scale up or down the acoustic data over time, which might affect its distribution over time. Thus, for the same samples as above, we normalized the samples and again performed K-S tests. Due to the fact the resulting p-values are all less than 1.0×10^{-25}, for brevity we choose to not include the test summaries here. We thus conclude that the acoustic data distribution changes over time.

Other Tests for Normality

To explain the analysis above, D'Agostino-Pearson Normality Test is a goodness-of-fit test that tests if the given sample comes from a normally distributed population. The test is based on transformations of the sample kurtosis and skewness. The sample skewness is the following equation:

$$\frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^3}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^{3/2}}$$

And the sample kurtosis is the following:

$$\frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^4}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^{2}}-3$$

To explain the Kolmogorov-Smirnov Test used above, the Kolmogorov-Smirnov Test is a nonparametric test that compares a sample with a reference probability distribution. It quantifies a distance between the empirical CDF of the sample and the CDF of the reference distribution. The test statistic is the following:

$$D_n = \sup_x |F_n(x) - F(x)|$$

We performed other normality tests to further verify that the acoustic signals did not follow a normal distribution. We use a Chi-Square Goodness of Fit test to compare an empirical distribution with a target distribution and divide our data into 10 intervals. The test statistic for Goodness of Fit is given in the following equation:

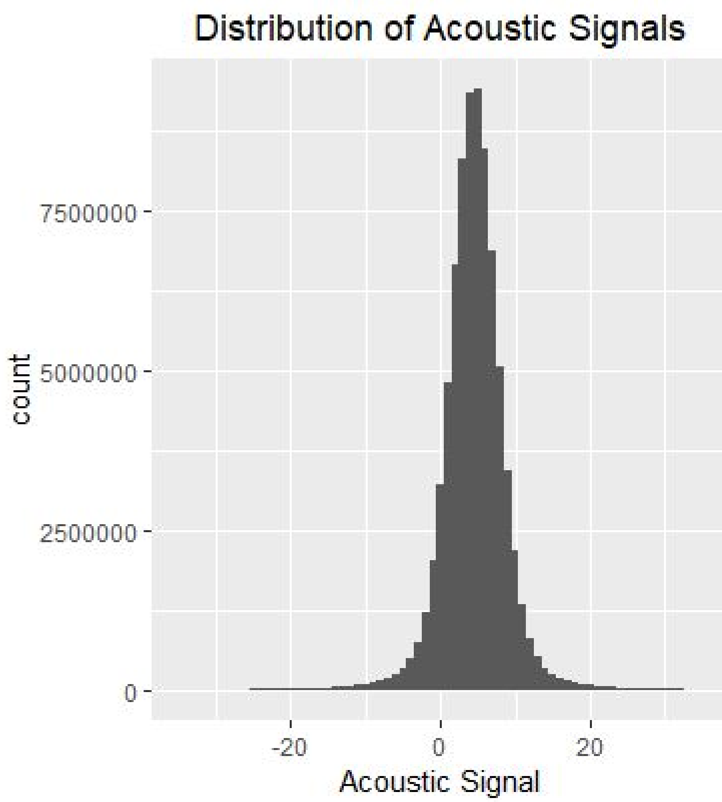
$$\chi^2_{m-k-1}$$

Such that *m* is the number of categories and *k* is the number of parameters being estimated to obtain the expected counts (2 in this case; one for the mean and one for the standard deviation). We obtain a test statistic of 5238800 and a p-value of 2.2e-16, thus concluding that the acoustic data is not normally distributed.

Another test that we used to test for normality was the Shapiro-Wilks test. We obtained a test statistic of W = 0.2649 and a p-value of 2.2e-16, so yet again we conclude that the acoustic data is not normally distributed. The test statistic is the following for the Shapiro-Wilks test:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

We decided to test the distribution of the acoustic signals without “spikes.” We defined spikes to be anything outside of 3 standard deviations.

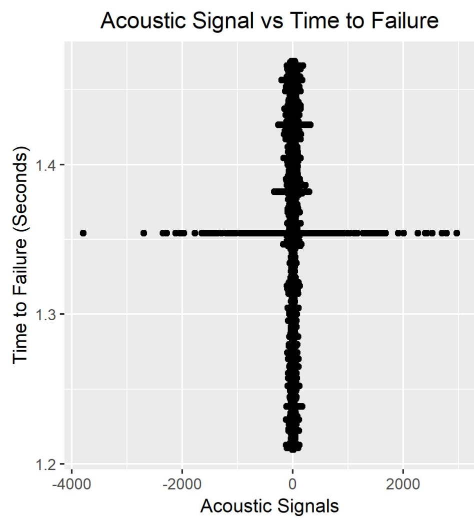


At first glance, it seems like these data follow a normal distribution. However, we perform a K-S test, Chi Square Goodness of Fit test, and Shapiro Wilks test on this subset of the acoustic signals and find that is is still not distributed normally. Now, we move on attempting to predict time to failure with various algorithms.

Prediction

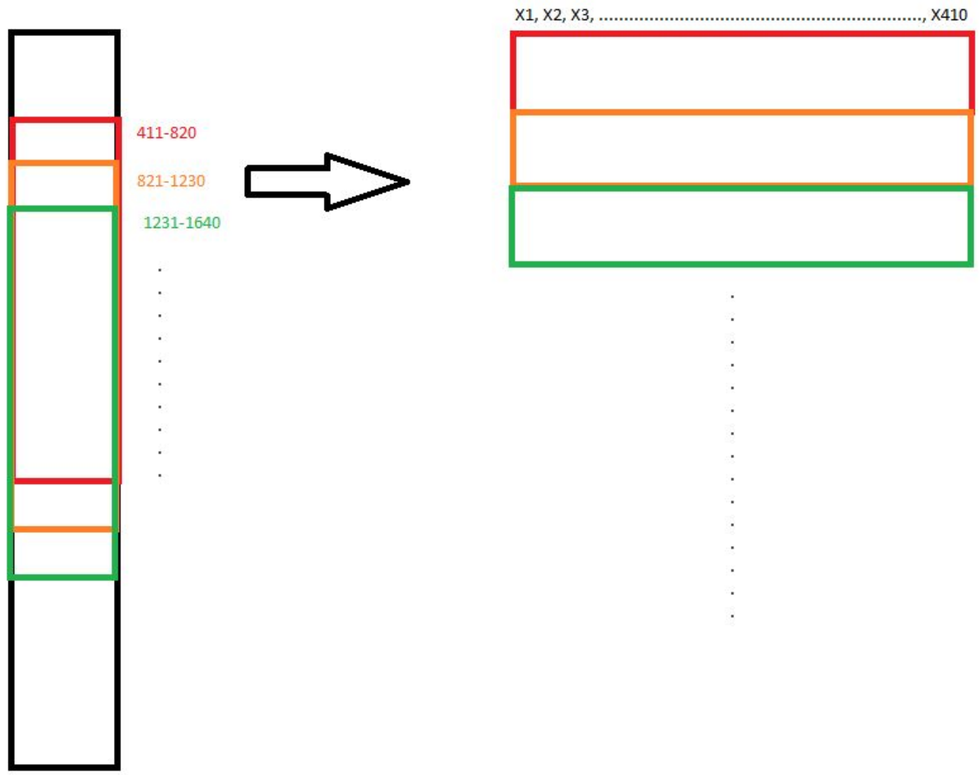
Multiple Linear Regression

In our first attempt to predict the time until a failure, we created a multiple linear regression algorithm and trained our data on a portion of the dataset. The following is a scatter plot of the subset that is used for this algorithm. As we can see, most of the acoustic signal values tend to be smaller values around 0.



The algorithm design is as follows:

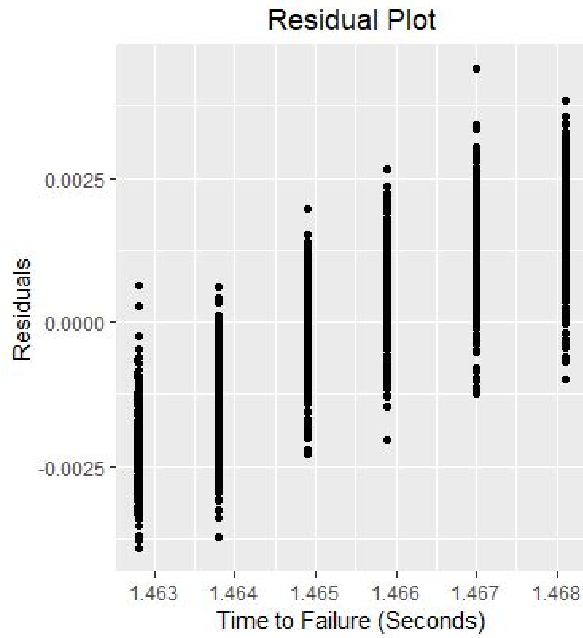
1) Subset original data to have 9,999,900 observations of acoustic signals in a single vector.
2) Sample every 10th value of our data in order to optimize memory and have a new data set containing 1,000,000 observations of acoustic signals.
3) Starting from the 411th observation of the subsetted data, we store every set of 410 data in a row of a matrix, so that we end up with 2,420 rows. The reasoning for choosing 410 is due to the length of a stationary period as mentioned in the *Basic Analysis* section, but since we are sampling every 10th observation, we use 410. Thus, we create 410 variables such that each variable contains the leading values up to the end of each stationary period.



Now, we use a multiple linear regression to predict time to failure. Our multiple linear regression equation is the following:

$$\widehat{Time\ to\ Failure} = \hat{\beta}_0 + \hat{\beta}_1 * X_1 + \hat{\beta}_2 * X_2 + ... + \hat{\beta}_{410} * X_{410}$$

It is not apparent that the Residual plot follows any kind of distribution.



In order to test the accuracy of our model, we test it on the following 836,400 data of the original data (observations 9,999,901 to 10,836,300), sampled at every 10 points and stored in a matrix the same way that we did for our regression algorithm to predict the time until failure for those observations. We obtained a mean squared error of 79.9945 with the following formula.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Our time to failure observations take on values from 0 to about 15, so this seemingly high mean squared error is not necessarily indicative of a poor algorithm to estimate time to failure. Next, we move to more advanced prediction algorithms.

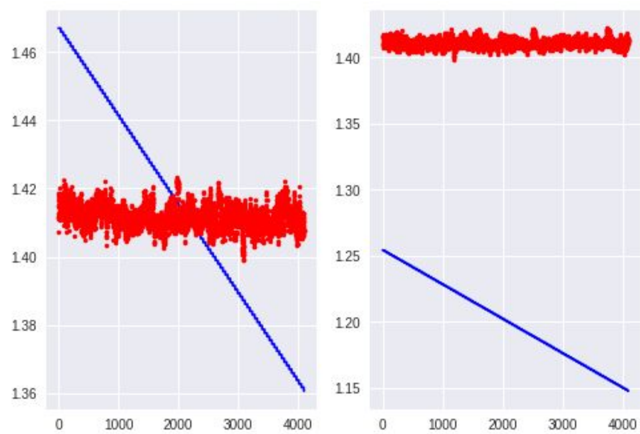
K Nearest Neighbors Regression (KNN Regression)

In considering the fact that similarities in acoustic data might lead to similarities in the experimental conditions and thus similarities in time-to-failure values, we attempt to utilize KNN Regression on our dataset to explore the possibility of predicting earthquake times. KNN operates on multidimensional training data and a metric. When predicting new data, KNN Regression finds the K nearest data points in the training data based on the specified metric, and then perform a regression on this subsetted dataset to arrive at a prediction value for the inquiry vector.

In order to preserve the temporal locality of the data, we decide to segment the acoustic signals into windows of 4100 data, with each window the data represent a continuous time frame. Our first job is to look for an optimal K value. To that end, we compare the results of 3 different K values - 300, 60, and 12. The scoring metric used for comparison here is the R^2, the formula of which is given below.

$$R^2 = 1 - \sum \frac{(TrueY - PredY)^2}{(TrueY - \frac{1}{n}\sum TrueY)^2}$$

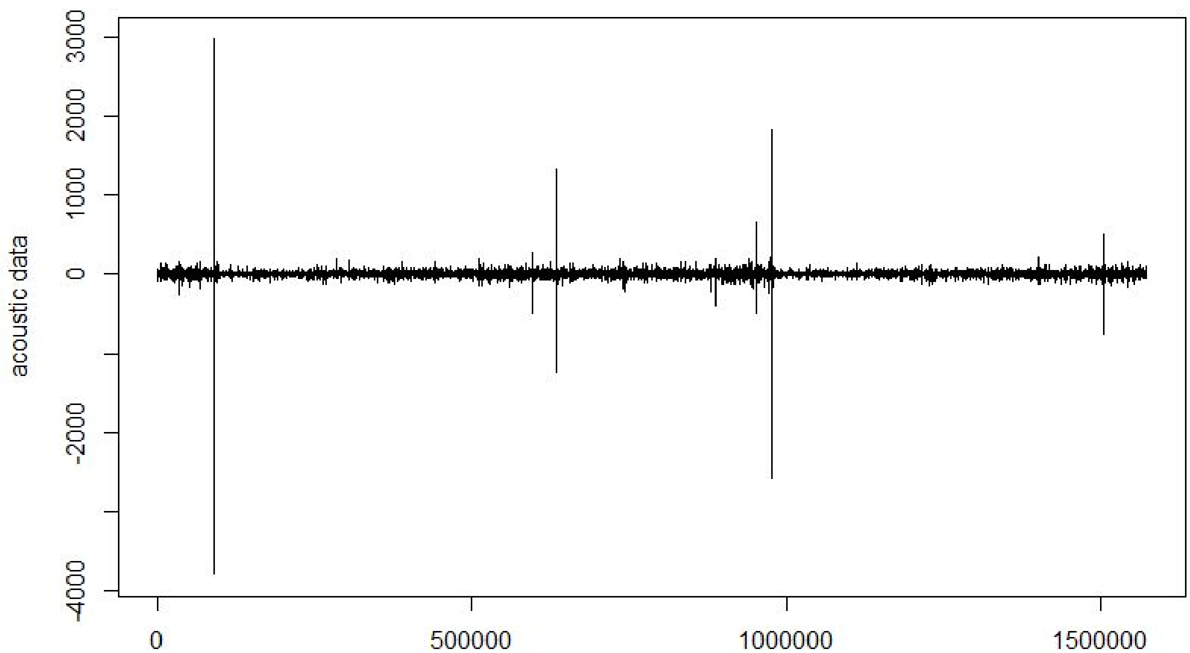
The R^2 of the three models are very similar to each other, with K=300 yields slightly better value than the other two. The three R^2 values are, respectively, -0.00087, -0.0244, and -0.0387. We plot the training and prediction values of the K=300 model here. The blue line represents real time-to-failure values, and the red dots represent the values predicted by the KNN Regression algorithm. From the graph, it is clear why KNN Regression does not perform well. The similarities of acoustic data along in fact does not predict the time to failure. Thus, the algorithm merely attempt to match the most similar data points and predict the means. As we increase the value of K, the algorithm simply take into account more training values in the calculation of mean.



Thus, we conclude that KNN Regression is not a very useful model in this case.

Time Series Analysis

From the analysis above we see that there appears to be patterns in the data which look like a time series so we attempt to fit a model to this series and apply forecasting to get a prediction for the spikes which have predictive values to the occurrence of earthquakes.



The above graph shows the first ⅛ of the data which we will be analyzing seeing as we are limited in computing power. We hope that the period between the spikes can be represented as a time series. Note that in order to run the code on our computers we had to further chunk our data by selecting every 100th data point in our sample.

First we test to determine if our series is stationary which would imply that there are no overall trends in our data which is what we expect given the description of the data. We use the Dickey Fuller Test.

H_0: The data is not stationary
H_1: The data is stationary

We get a p-value of less than .01 so we reject the null hypothesis in favor of the alternative and proceed with the assumption the time series is stationary.

This test also estimates the Auto Regressive Moving Average model we can use to forecast this data:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

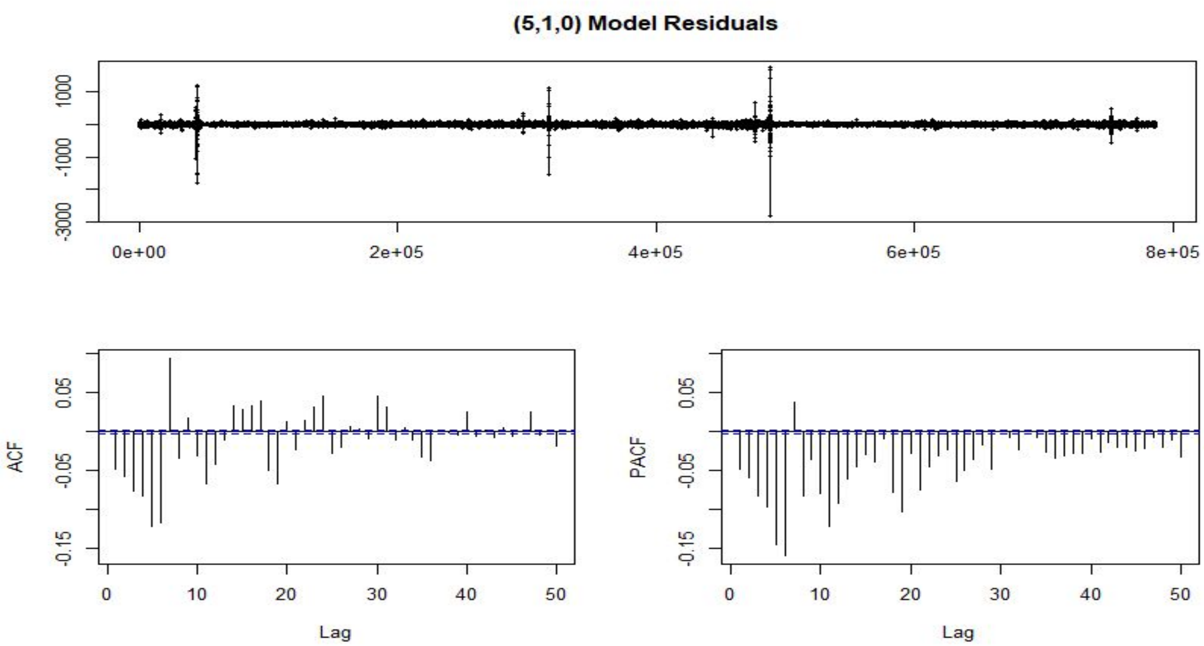
The equation above represents Xt as a sum of a constant c, and error ε, φ_t represents the coefficient of the Auto Regressive(AR) model, and ε_t is the coefficient of the Moving Average(MA) model.

Using Auto Arima in R we estimate the AR to be of order 5 with the following coefficients:

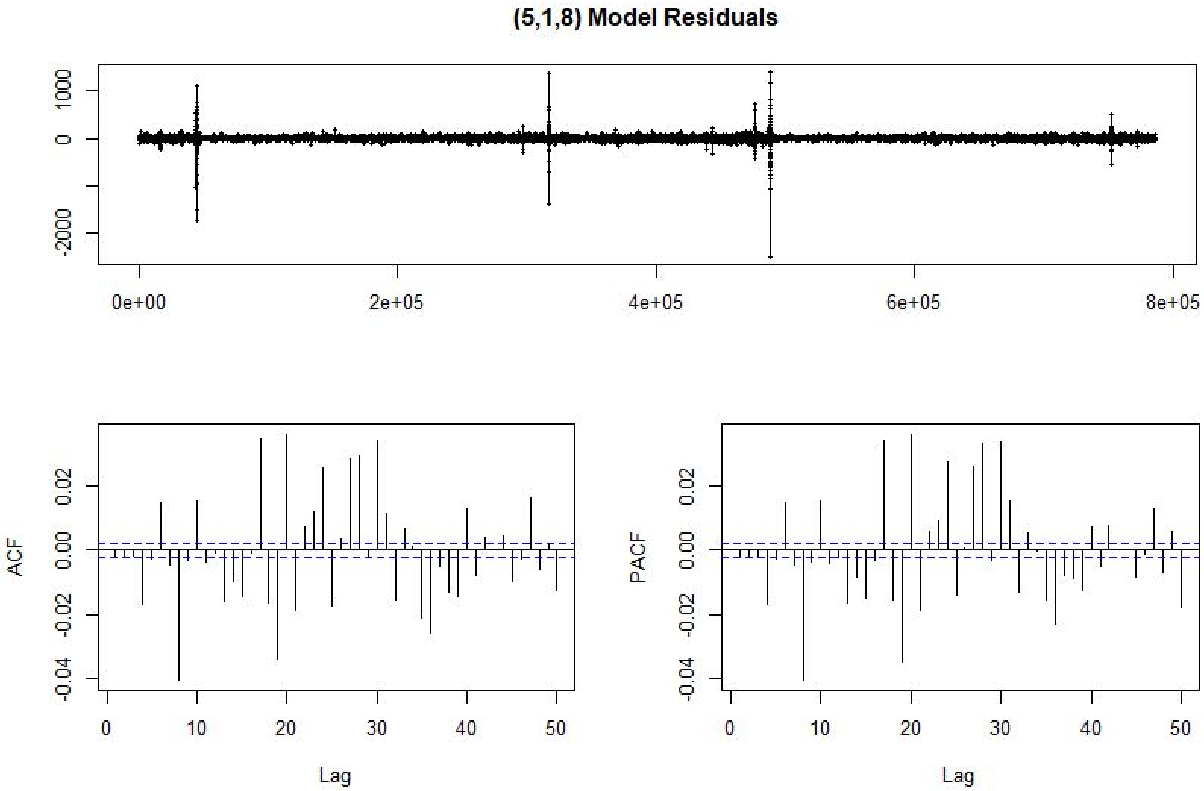
ar1	ar2	ar3	ar4	ar5
-0.8895	-0.6239	-0.5201	-0.3942	-0.2284

The σ^2 is estimated to be 108.2.

We plot this AR(5) MA(0) process and show the results below.



Ideally we would like the lag spikes in the ACF graph shown to be concentrated within the blue lines which would indicate that our model accurately represents the lag component of the time series. Note that the lag is the size of the interval for which data will be dependent and outside the lag data are independent of one another. We see a pattern about every 8 units in the lag so we apply a MA(8) process to our data. The results are shown below.



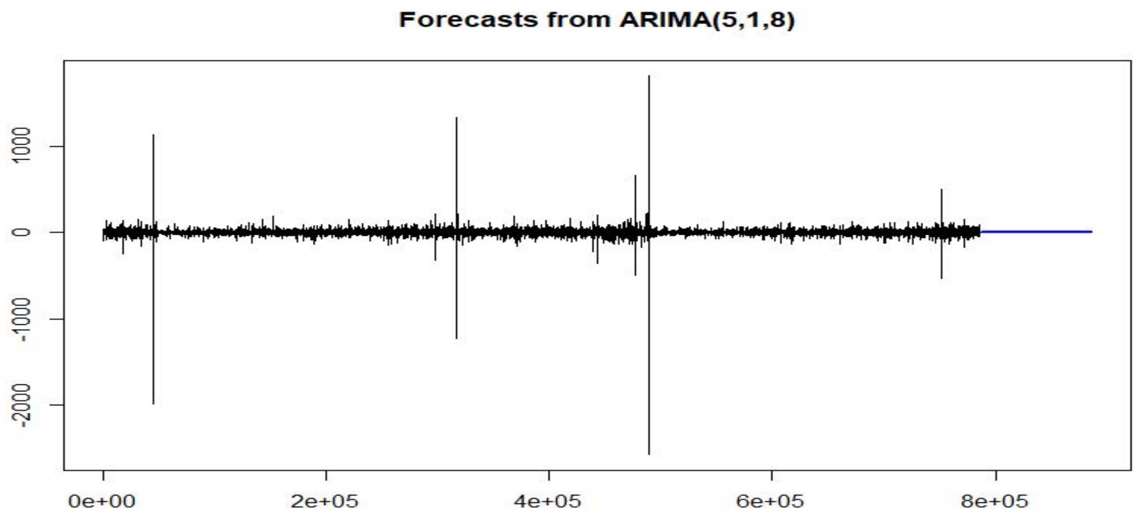
As we can see the data looks better with an AR(5) MA(8) representation but it is still not ideal. Unfortunately I am limited by computing power in my ability to try to fit an ARMA model with larger coefficients therefore I will proceed with forecasting the AR(5) MA(8) model as that is the best fitting one that I can create.

Here are the coefficients for the AR(5) MA(8) process used:

ar1	ar2	ar3	ar4	ar5	ma1	ma2	ma3	ma4	ma5	ma6	ma7	ma8
-.888	.152	-.256	-.574	.094	.981	-.087	.278	.338	-.483	.015	-.027	-.053

σ^2 is estimated to be 91.65.

The model below is our forecast for the next 100000 data.



It is clear that this model does not give any useful information in trying to predict the next earthquake which leads me to believe that our model does not take into account a large enough lag and unfortunately our technology is not able to simulate higher order ARMA processes on this data but this may be worth studying on a transformation of the data or on a more capable computer.

Kernel Ridge Regression

Other forms of prediction methods we used were Ridge Regression and Kernel Regression, both of which can be imported from sci-kit learn. In order to use regression models, our first step to this problem was to conduct feature engineering for each segment in the test set. A segment was defined as every 150,000 rows in the training set, so we divided up the entire data set by 150,000 which gave us 4194 data points. Since our data set was a time series, we decided to calculate certain summary statistics for each segment to act as our features. These included the mean, standard deviation, maximum, minimum, the average of the rolling standard mean, the average of the rolling standard deviation, the standard deviation of the rolling mean, and the standard deviation of the rolling standard deviation. We including rolling values in the feature set because of how our data set is a time series. When we did some exploratory analysis and plotting the acoustic data along with the time to failure we noticed that there would be a spike in acoustic values followed up small values of acoustic data before time to failure reached 0. Because of this, it made sense to use rolling values to help us potentially capture this trend. For the time to fail, we decided to take the last value of the segment (150,000th observation) and use this as the value to predict on. Our resulting training x data frame was below:

	mean	standard_dev	max	min	kurtosis	avg_rolling_mean	avg_rolling_std	std_rolling_mean	std_rolling_std
0	4.884113	5.101106	104.0	-98.0	33.662481	4.883418	4.288590	0.295716	2.769782
1	4.725767	6.588824	181.0	-154.0	98.758517	4.724876	4.843486	0.231587	4.492920
2	4.906393	6.967397	140.0	-106.0	33.555211	4.905840	5.423013	0.267013	4.402155
3	4.902240	6.922305	197.0	-199.0	116.548172	4.901486	4.939280	0.266701	4.873539
4	4.908720	7.301110	145.0	-126.0	52.977905	4.910196	5.121868	0.228006	5.213382
5	4.913513	5.434111	142.0	-144.0	50.215147	4.914758	4.538771	0.293942	2.988892
6	4.855660	5.687823	120.0	-78.0	23.173004	4.856829	4.799256	0.219473	3.076807
7	4.505427	5.854512	139.0	-134.0	52.388738	4.504153	4.503100	0.218515	3.763111
8	4.717833	7.789643	168.0	-156.0	65.360261	4.718397	5.695230	0.240917	5.328307
9	4.730960	6.890459	152.0	-126.0	53.760207	4.731377	5.209570	0.258463	4.536904
10	4.582873	6.157272	245.0	-115.0	117.464170	4.583277	4.664996	0.270993	4.038939
11	4.329933	15.254000	410.0	-478.0	196.821674	4.330214	7.614385	0.264042	13.273237
12	4.464040	8.660502	224.0	-169.0	67.314625	4.465509	6.011158	0.264724	6.269617
13	4.680813	6.033346	139.0	-129.0	42.453016	4.681638	4.887262	0.277479	3.557937
14	4.596720	5.402047	98.0	-125.0	41.255940	4.594519	4.361161	0.211897	3.173082
15	4.597953	6.328509	126.0	-117.0	38.297358	4.597498	5.103126	0.257716	3.761440

After creating the features, we created a RidgeCV model and tested it on various alphas and chose a 5 fold cross-validation. After fitting the model, we scored it using the original training data and received a score of around .36. We tried around with other cross validation values and received a score of .36 for each one. This model wasn't very successful, so we were eager to try using the Kernel Regression model.

For our Kernel Regression we decided to use the Kernel Ridge model with the same list of alphas as before. We experimented with both rbf and linear for the kernels. The rbf yielded much better results compared to linear which yielded results with less than .30. Since rbf requires us to use a gamma value, we also experimented with several gammas in order to find one that does not overfit our model. A gamma greater than .05 yielded scores of .99 and greater. After playing around with the gamma, we found that .005 yielded a score of .85 which we concluded maintained high prediction accuracy while not overfitting the model.

Conclusion

There is clearly a lot of potential to use mathematics and machine learning to better understand how earthquakes can be better predicted. Though we had limitations from not having computational power, we were still able to develop methods to attempt to predict occurrence of laboratory simulated earthquakes.

Unfortunately, neither multivariate regression or KNN regression yielded a result with significant accuracy. Also, the data when interpreted as a time series appeared to have too large a lag for us to accurately fit an ARMA process to it and forecast the data. Kernel Ridge Regression with rbf kernel yielded the best fitting results. By computing the rolling max, min, means, and standard deviations, we were able to predict the time to failure corresponding to the last acoustic signal for each set of computed features and obtain a training R^2 of 0.9976. However, the testing result is only comparable with KNN regression with an R^2 of -1.59.

This analysis showed how having limited information, i.e. having too little variables, can present challenges for data analysis. However, this raises very important questions about how analysts can construct data with what they are given, and to really notice trends and patterns in the limited data.

Works Cited

1. "LANL Earthquake Prediction." *Kaggle*, Jan. 2019, www.kaggle.com/c/LANL-Earthquake-Prediction.
2. Leeman, J. R., et al. "Laboratory Observations of Slow Earthquakes and the Spectrum of Tectonic Fault Slip Modes." *Nature News*, Nature Publishing Group, 31 Mar. 2016, www.nature.com/articles/ncomms11104#introduction.
3. World, Physics. "Earthquake in a Lab." *YouTube*, YouTube, 12 July 2017, www.youtube.com/watch?time_continue=73&v=m_dBwwDJ4uo.