Programming in R: Final Project

Laura Ye

September 26, 2017

Motivation

There were a few key points to the motivation for this project.

The first key point was that I wanted to utilize R with machine learning applications. This was my first time using R and my only previous engagement with machine learning was with Python. I wanted to use R to apply machine learning techniques so that I can master the R language and also to evaluate which language was better with machine learning applications.

The second and final key point was that I wanted to develop something that had real-world applications. This is not just purely data visualization. My hope was that this would eventually become a tool that others can use repeatedly and reliably.

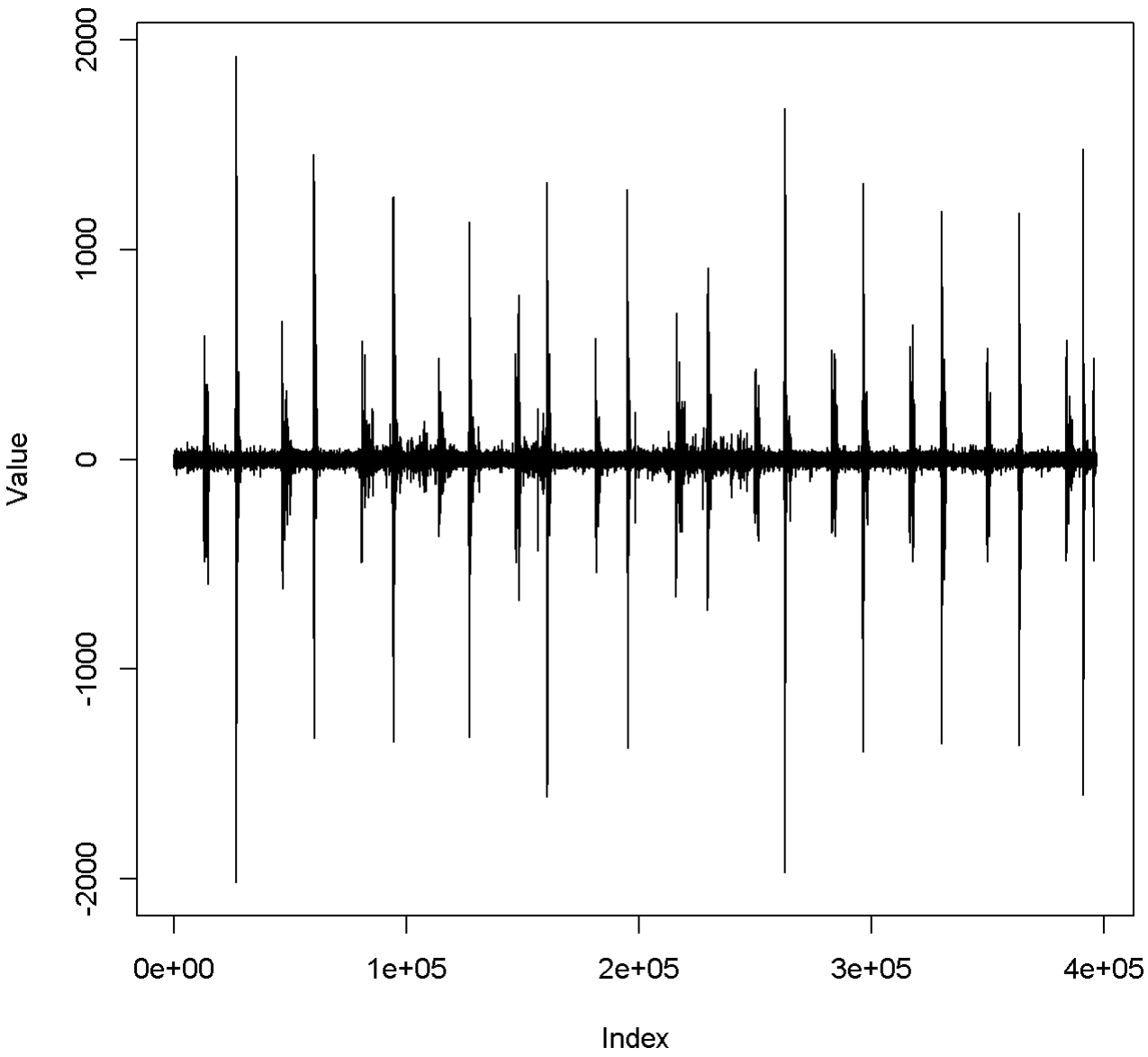
Data Sources

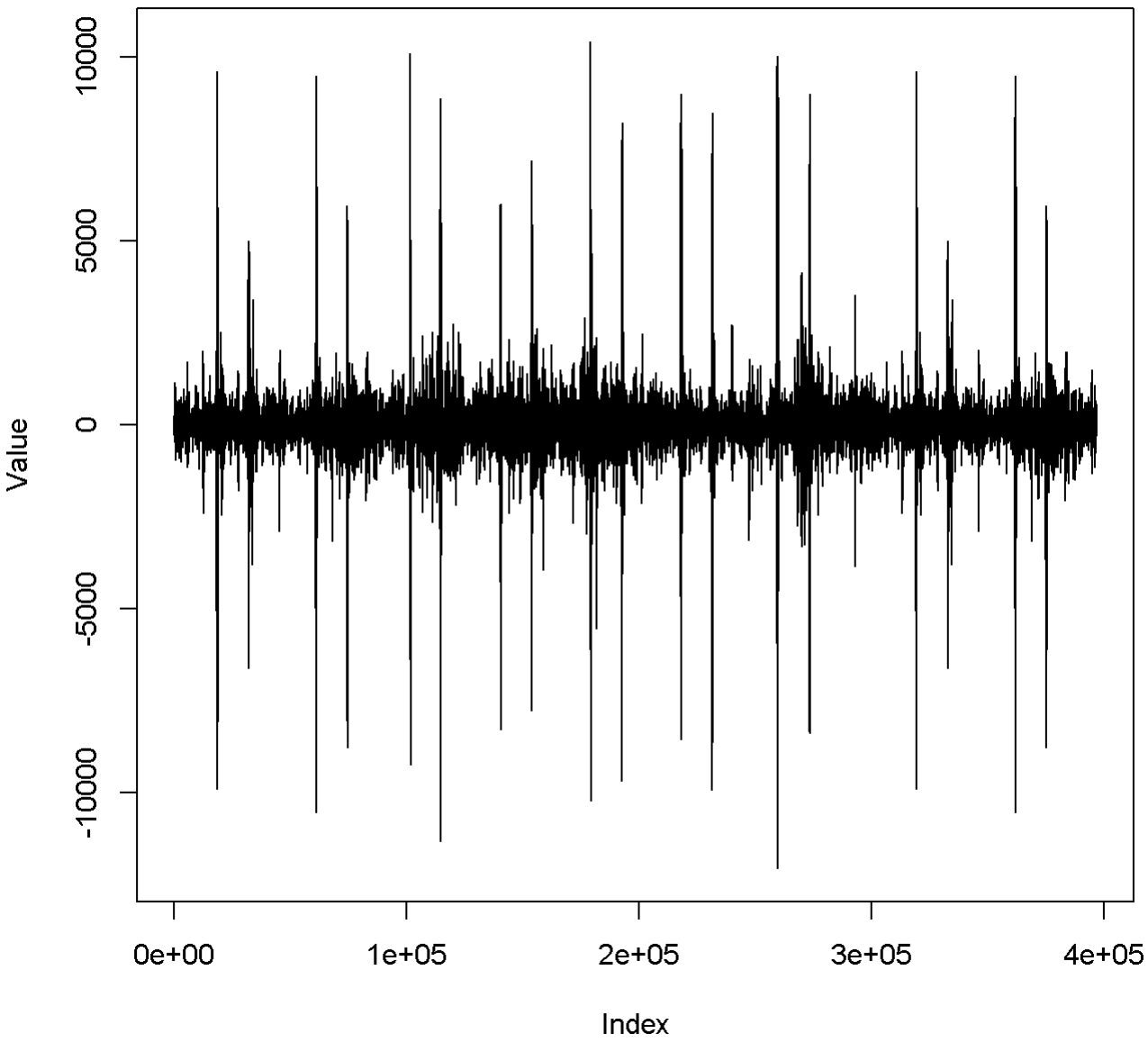
The source of this data came from a Kaggle Dataset called, “Heartbeat Sounds”. Sound waves of four different types of heartbeat sounds (labelled) were analyzed.

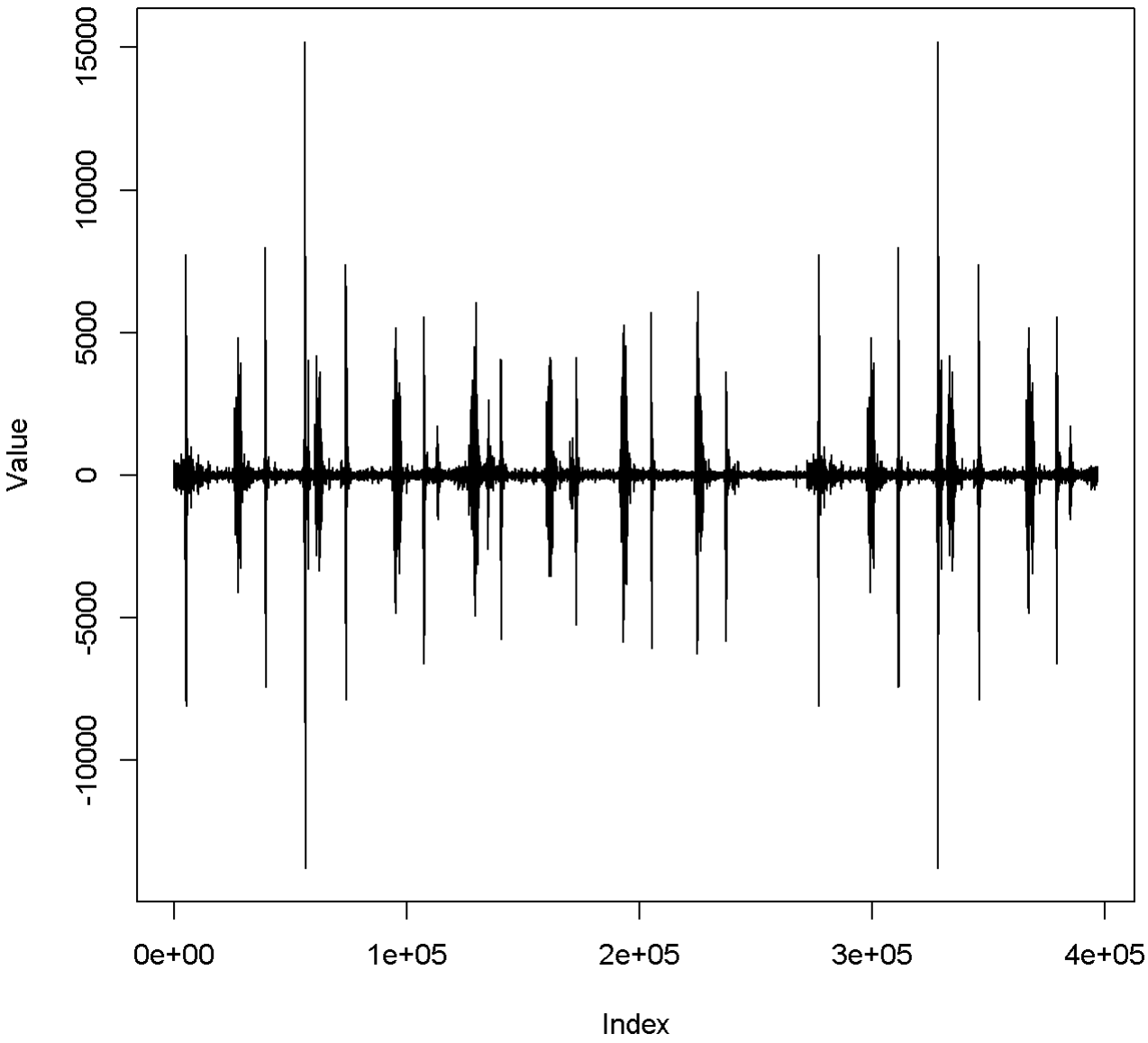
Download the data set here: Heartbeat Sounds (http://www.kaggle.com/kinguistics/heartbeat-sounds)

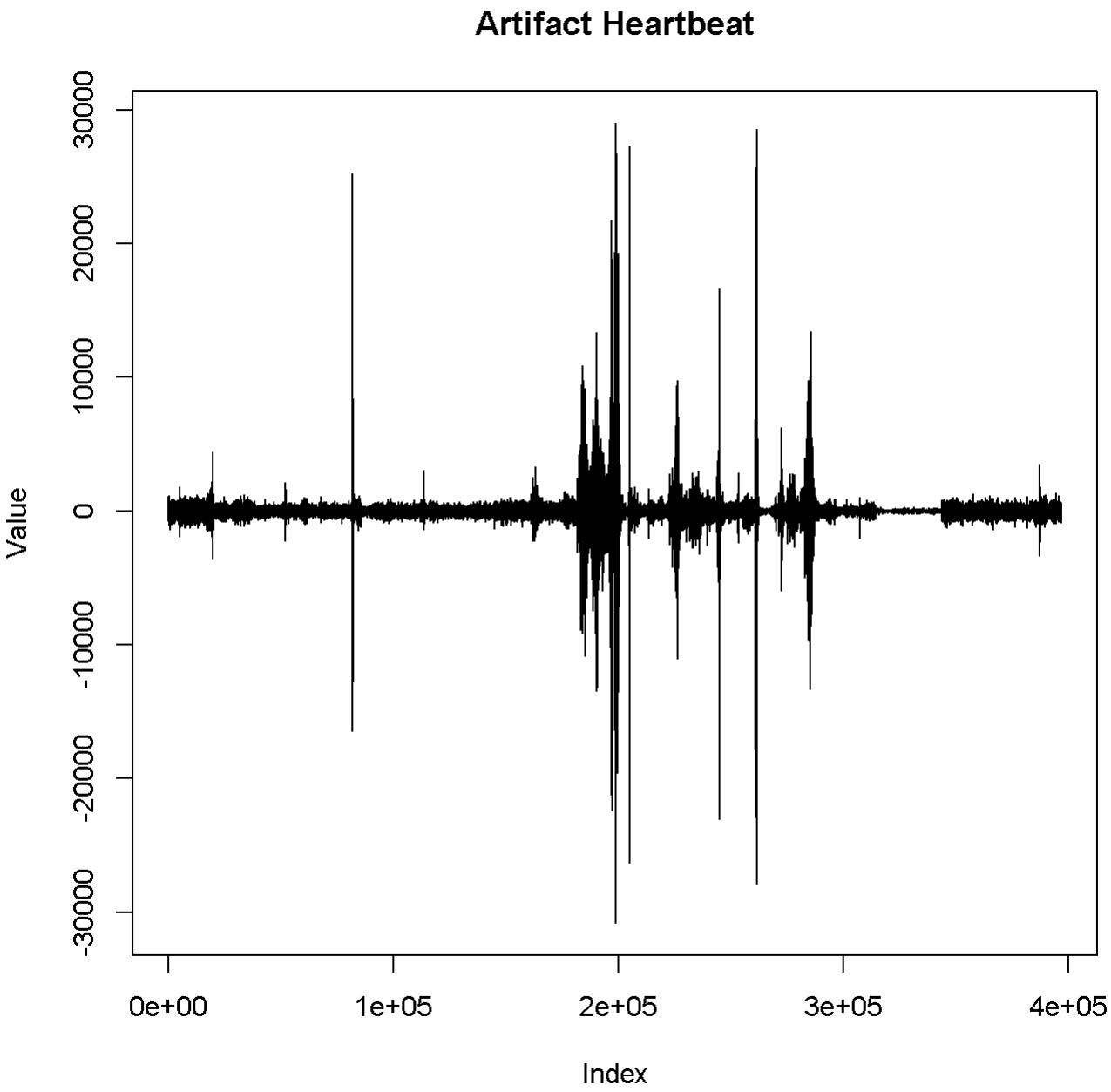
The data set consists of heartbeats in audio .wav files. These files were imported as numeric vectors using tuneR::readwave, and then manipulated as data frames.

Here is a sample of each type of heartbeat:


Normal Heartbeat
Value
2000
1000
0
-1000
-2000
0e+00
1e+05
2e+05
3e+05
4e+05
Index


Murmur Heartbeat
Value
10000
5000
0
-5000
-10000
0e+00
1e+05
2e+05
3e+05
4e+05
Index


Extrahl Heartbeat
Value
15000
10000
5000
0
-5000
-10000
0e+00
1e+05
2e+05
3e+05
4e+05
Index

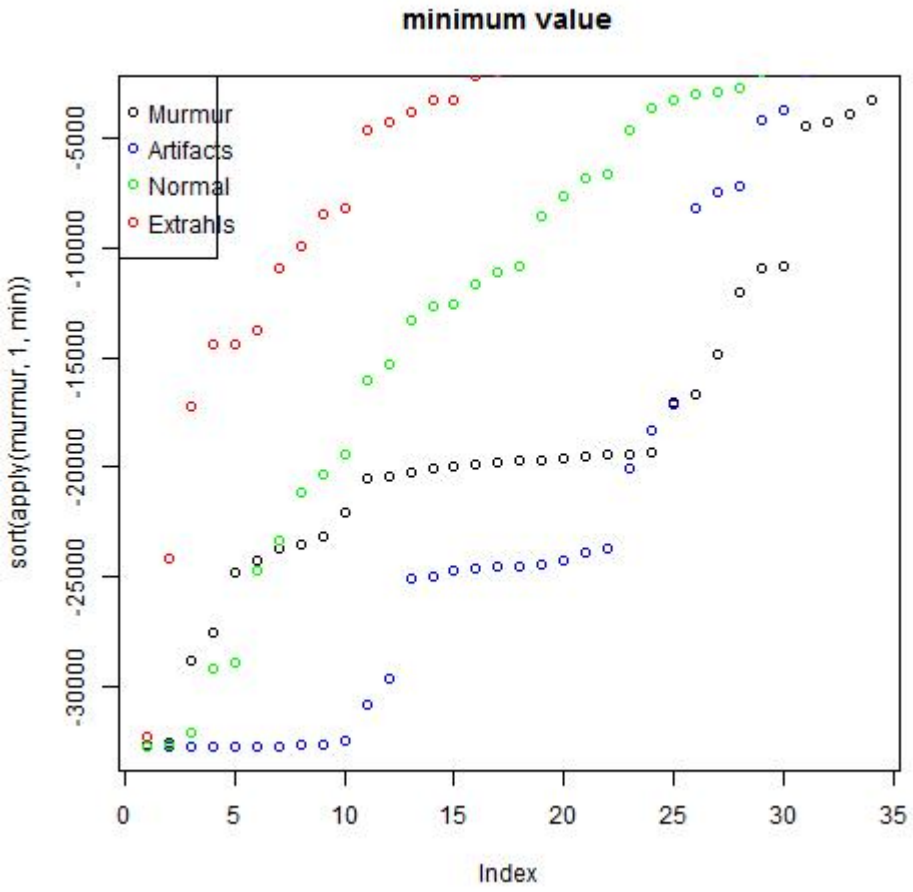


Effective Analysis Methods Applied

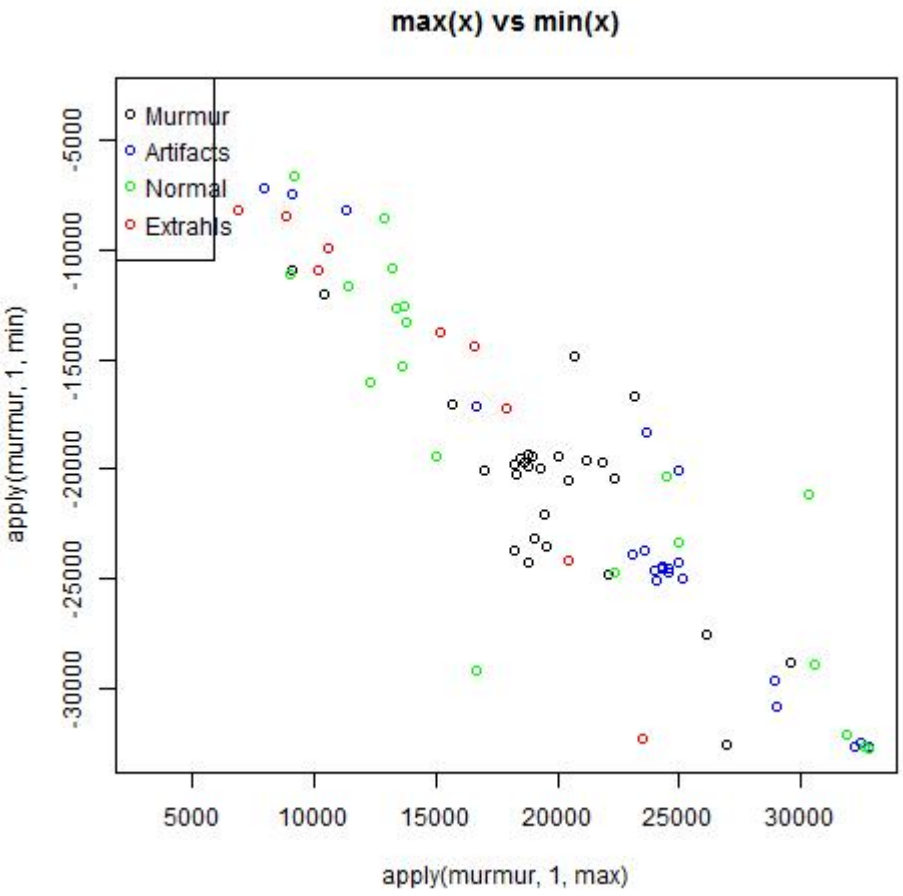
Exploratory Data Analysis

First, exploratory data analysis was conducted to visually inspect differences between vectors. Since each vector consisted of nearly 400,000 elements, they were simplified to one characteristic to simplify the visualization.

Of the various characterstics chosen, the minimum value of each vector seems to be well separated. image:



Characteristics were also graphed against each other and the plot for minimum value vs maximum value shows a linear correlation. This is expected since these are unprocessed audio signals and for each minimum value there is a maximum. image:



Other plots did not show strong correlations.

Time Series Conversion

Since each of these vectors were audio signals, one way to normalize the data is to convert them into a time series using `ts` , a built-in R function:

```
ts_model <- apply(labelled,1,ts)
```

These time series models were then used for the next subsequent steps.

Principal Component Analysis

Since there were close to 400,000 elements in each of the 124 vectors, the most practical way is to reduce dimensionality using Principal Component Analysis.

> Note that time series conversion does not reduce dimensionality and we still have close to 400,000 elements in each vector after the previous operation.

I used prcomp to compute the principal components and then selected a subset to use for subsequent machine learning applications.

Example below shows Principal Component Analysis using 2 components:

```
PCA <- prcomp(ts_model)
newdat <- PCA$x[,1:2]
```

Machine Learning

After time series conversion and principal component analysis, I tried to classify the data using these methods from the `caret` package: LDA, CART, Random Forest (rf), K-nearest neighbors (knn), SVM (svmRadial), Naive Bayesian (naive_bayes) and also neural network (nnet). These methods were diverse, yet applicable to this data set and I wanted to see which type of classifier would be the most accurate.

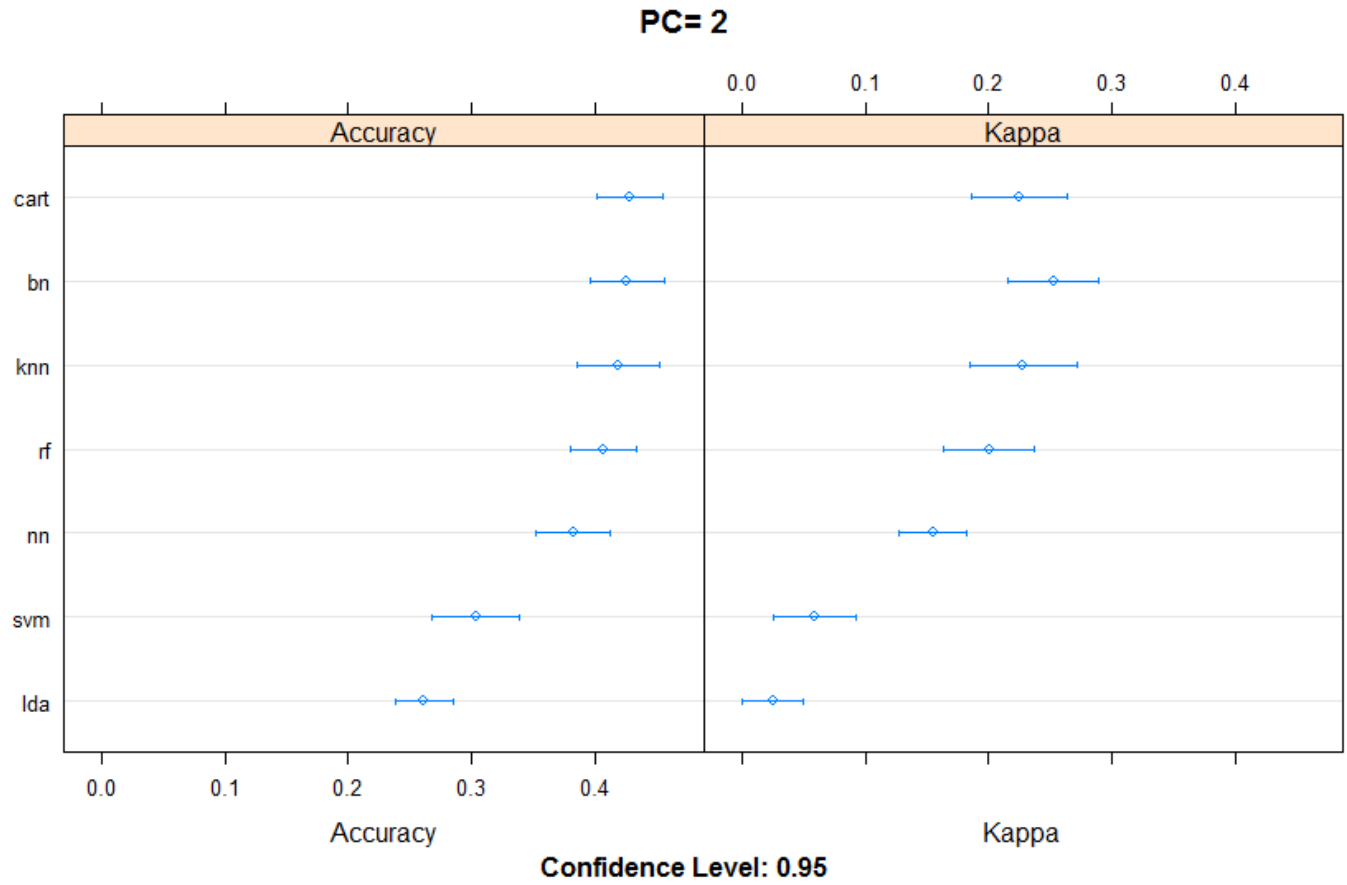
I also attempted k-means clustering for the data set for four clusters and visually inspected the results.

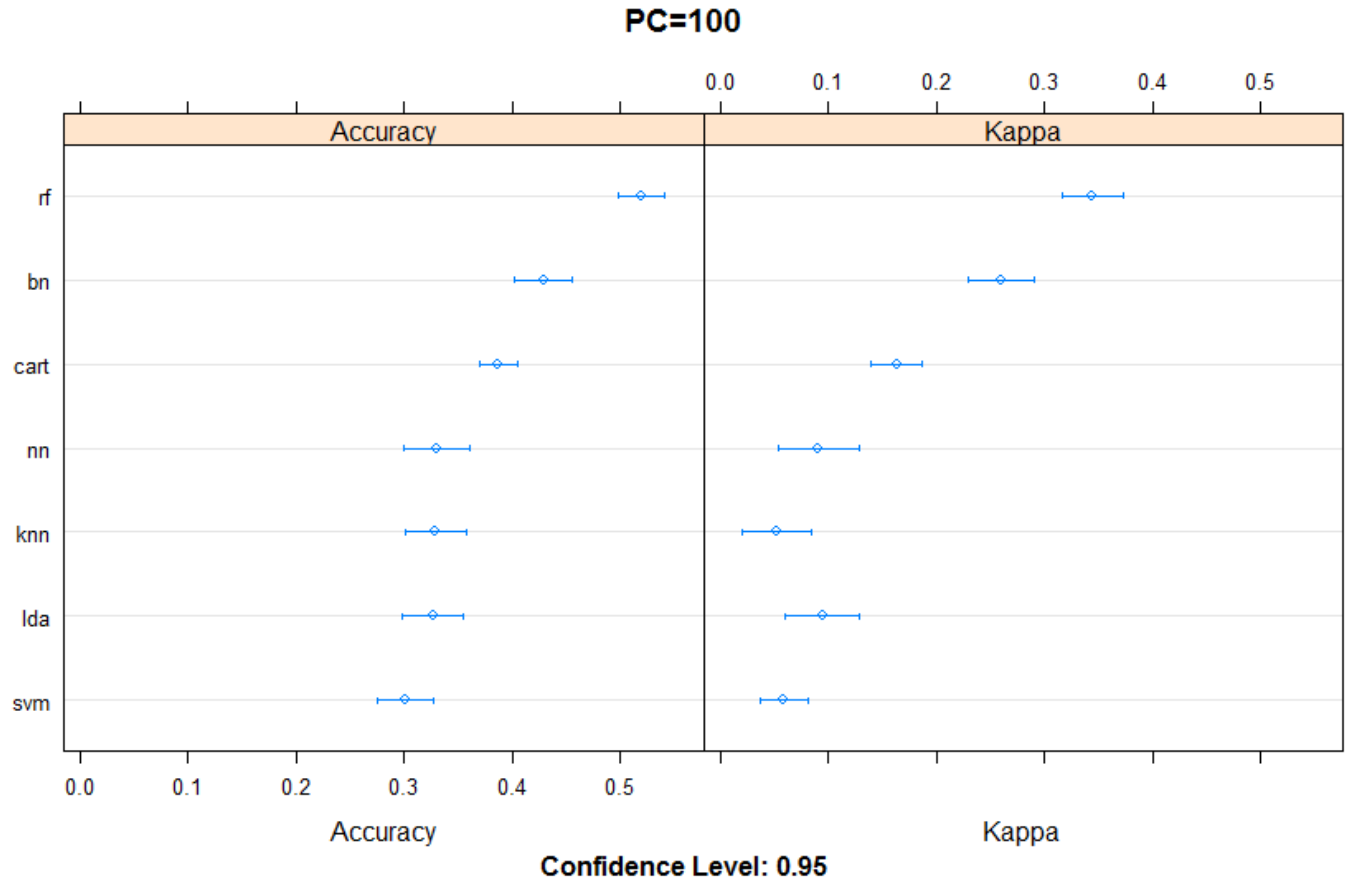
Results and Interpretation

The machine learning methods that I used were not very accurate for this application. The best classifier had an accuracy of ~50%. This is likely due to the nature of the data. There may be other methods that would produce more accurate predictions for this data set.

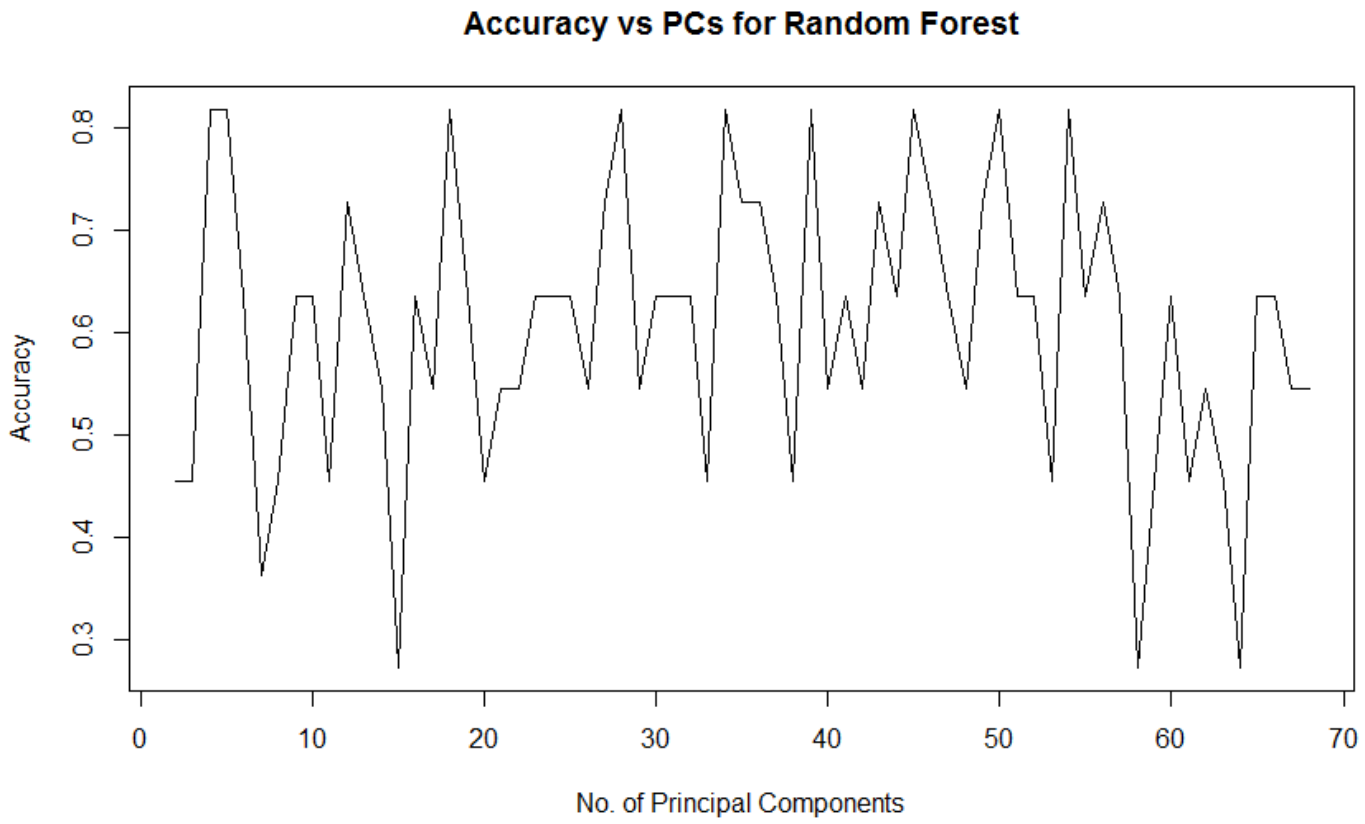
Best results out of `caret` package

Random Forest returned the best results in accuracy whether I used 2 principal components or 100. However, overall results for these methods were in the same general vicinity.





Let us see the changes in accuracy for random forest as number of principal components increased.



We can see the fluctuation is pronounced for number of principal components from 2 to 68.

K-means Clustering

Measuring the accuracy of k-means clustering was more subjective. Visually inspecting the results, for each iteration using different number of principal components, a majority of results (97%) would fall on one category and this category was not consistently the same.

No. of PCs	1	2	3	4
2	121	1	1	1
3	1	2	120	1
4	1	1	1	121
5	1	121	1	1
10	1	1	1	121
15	1	121	1	1
20	1	1	1	121
25	121	1	1	1
50	1	1	1	121
100	1	1	1	121

Conclusions

What did I learn?

From this project, I learned that each step in data analysis do not exist separately. For example, when importing and organizing the data set, it was important to think about something as essential as class type and also data distribution. Because each step builds on top of each other, when a change is required in an earlier step, some time is wasted in regenerating results from the intermediate steps. Therefore, it is important to understand what your data consists of, and plan out the implementation of data analysis for each data set.

I also learned more about handling big data. I was only dealing with 125 labelled data vectors, but since each vector consisted of almost 400,000 elements, analyzing these vectors took a long time without simplifying them using time series conversion and principal component analysis. It was unrealistic to wait for an extremely long time for the script to run. Once I implemented principal component analysis, even with 50 or 100 principal components, the time to run the script for machine learning part of the analysis was significantly reduced.

Future Work

How can this work be expanded?

Additional classification methods, especially those built into the `caret` package, may be more accurate in classifying this data. Of the few that I modelled in this project, random forest was the most accurate. Further work can be done to determine the best and most accurate caret method for classification.

Dimension reduction and decomposition of the audio signals using another method besides may also help in reducing time needed to run these classification methods.