**Project Deliverable 02**

Sentiment Analysis of Twitter data

**Name: Dean D’souza**

**H.U. ID: 168424**

# The Data

The data was obtained from the Sentiment140 project, which is a tool for the sentiment analysis of twitter data, and is in the form of a Microsoft Excel Comma Separated file.

It was built by the authors using a variety of technologies including the Twitter API, Amazon EC2, Google Gadgets, etc. The Twitter Search API was used to collect all the tweets in the dataset using keyword search and was automatically created. The authors used emoticons as indicators of positive and negative sentiment, particularly ☺ to indicate a positive tweet and ☹ to indicate negative tweets.

However, the dataset which was free to download consisted of training and test data which had all emoticons stripped out of the tweets.

The data set consists of the following variables:

1. Tweet Polarity (0 = negative, 2 = neutral, 4 = positive)
2. Id of the tweet (2087)
3. Date of the tweet (Sat May 16 23:58:44 UTC 2009)
4. The Query (For example: lyx). If there is no query, then this value is NO\_QUERY.
5. The user that tweeted (For example: robotickilldozr)
6. The text of the tweet (For example: Lyx is cool)

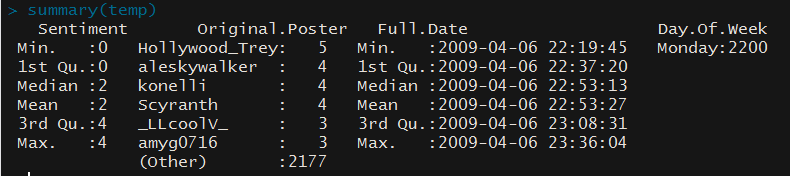
As the data consists of over 100,000 records (or rows), a sample of 2,200 records (1,100 of negative and 1,100 of positive) is taken for the purpose of this project. However, this number is subject to change as the project goes forward (especially during the final stages). The test data on the other hand consists of only 500 records and so can be left as is for the purpose of testing the final model.

The sample of the original training data is attached with the file. An intermediate form of the training data after removing unnecessary variables and creating new variables was also created and is also attached.

# Data Summarization

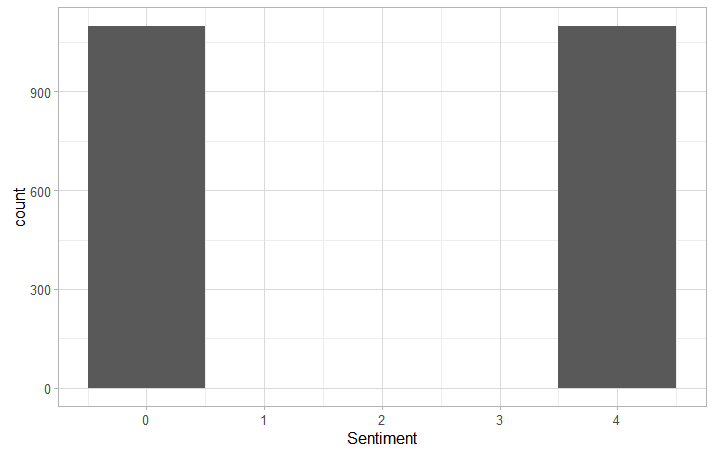
An important note to made at this point is that the following statistics is based off the sample data of 2,200 records and that most of the tasks of preparing the data and presenting the statistics was done in R (using R studio, for convenience). Additionally, a few more interesting aspects of the training data that came up during this deliverable were that the training set had no query observations and there were no neutral sentiment tweets.

The summary statistics from the R summary() is as follows:

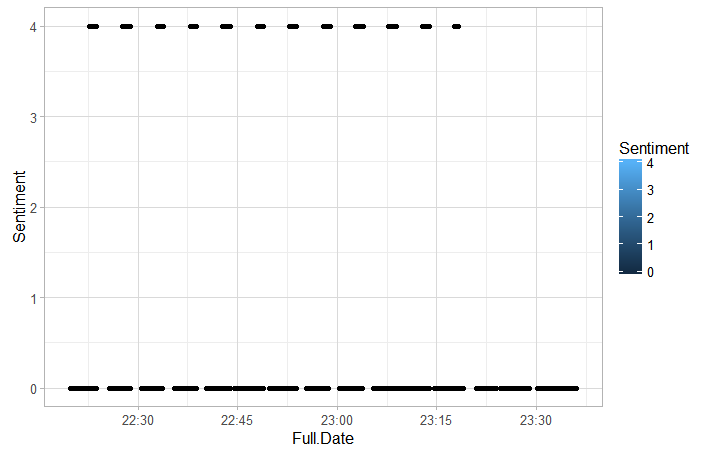


As we can see from the above statistics, the sample data chosen shows expected mean and median values for the Sentiment variable. We can also see that the sample data takes into account tweets from around 2183 twitter users and the data was collected over a period of a few hours (though this may not be true as the entire training data is not present). Additionally, the data was collected on a Monday so we can now choose to drop the Day.of.Week variable as well. These statistics naturally does not take into account the Text data which needs to be given to the model built in Python.

We also explore these facts through graphical means as follows:



As we can see the training set has an equal distribution of positive and negative cases. We can also see how the sentiments varies over the course of a few hours as follows:



Due to the nature of how ggplot package of R renders scatterplots, we tend to think that there are very few cases of positive sentiments over time. However, the actual case is that most of the positive tweets are clustered around a particular time while the negative tweets are more spread out over time.

We also try to find the Original Posters who have contributed the most number of positive and negative tweets (and the number of those tweets) in the training set as follow:

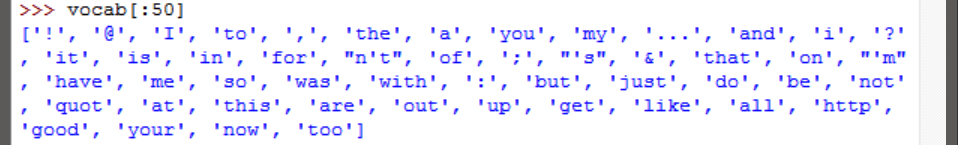
Positive



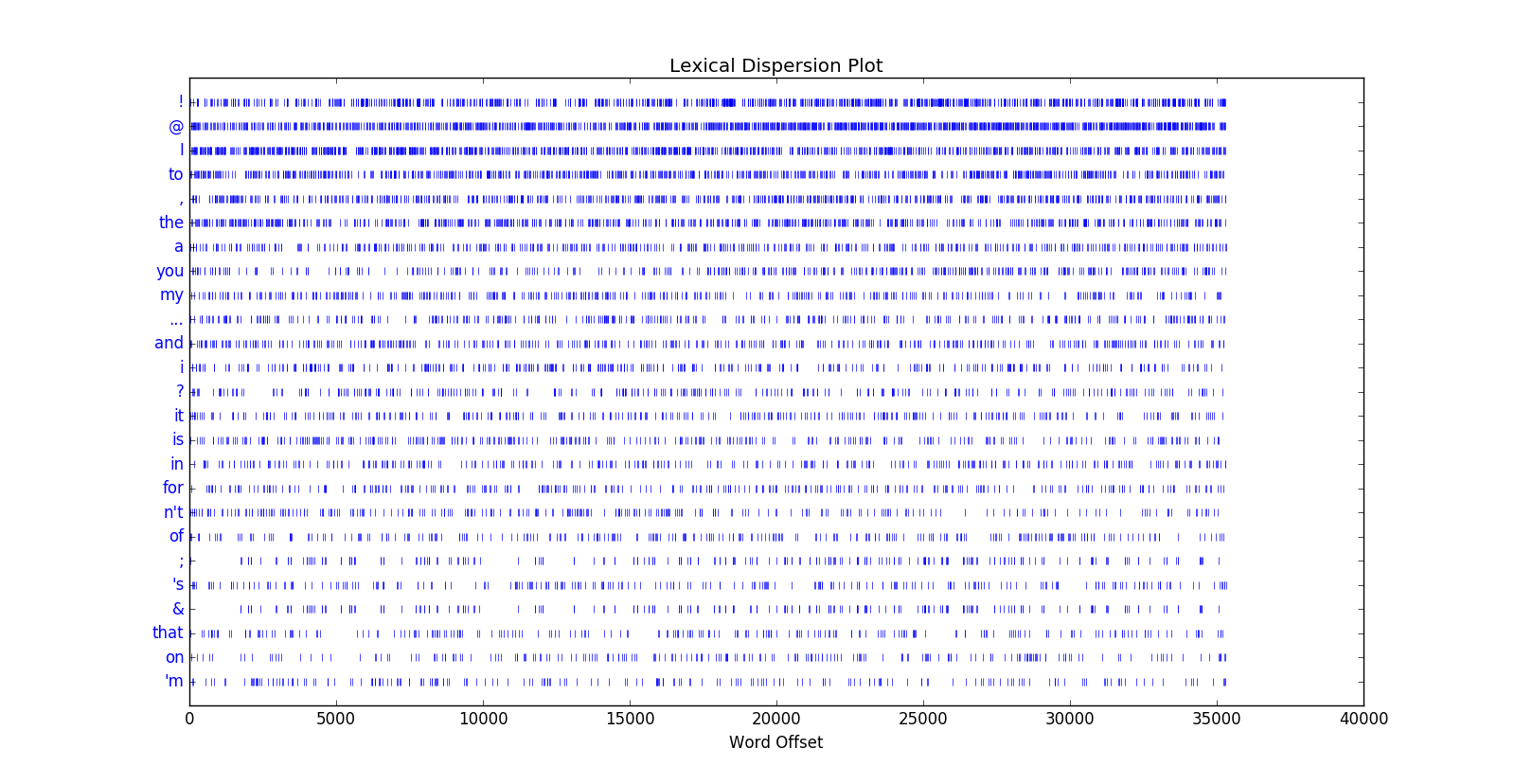
Negative



We also get a list of the 50 most common words types (or rather tokens) used in the tweets as follows:



We also create a dispersion plot for the most commonly used tokens as follows:



# Proposed models

As a number of minor details still need to be worked out, the only models under consideration are the Naïve Bayes Classifier and the Decision Tree Classifier models, for which the implementation is already present in NLTK.

However, a good corpora still needs to be made in order to support the process of sentiment analysis.

# References

[1] Sentiment140, (<http://help.sentiment140.com/for-students/>)

[2] “Text classification for Sentiment analysis -Naïve Bayes Classifier”, Jacob Perkins, (<http://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/>)

[3] “Twitter Sentiment analysis using python and NLTK”, Laurent Luce, (<http://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/>)

[4] “Natural Language Processing with Python”, Steven Bird, Ewan Klein, and Edward Loper, O'Reilly Media, 2009, (<http://www.nltk.org/book_1ed/>)