Data wrangling - Project Report

The relationship between twitter usage and the occurence of world events.

Group Members:

James Zoryk	Sanne Petersen	Nicolina
jzk340	spn239	id3

1 Research Question

In this paper we shall investigate to see if there is any correlation between the density of social media posts per a given time interval against discrete events in the news. We hypothesis that a significant increase in the density for a giving time period shall indicate either a notable news or world event that has occurred in that time period. Furthermore, we believe that observing specific languages can lead to more localise events. However, we note that many languages spread across different countries, thus the phenomenon must be limited to localised languages.

The scope of work for this paper is to choose a significant social news event, and then analysis the density of tweets created to see if we can observe a significant trend in the data. For the purpose of this paper while shall observe a localise language, namely Dutch.

For this paper the social media posts that we shall consider will be 'tweets' from the internet server called twitter. The reason for this is that all the tweets are in public domain, and that twitter serves as an online discussion platform, which is indicate by their slogan *"What's happening"*. Also from a data point of view, each tweet contains a lot of potential meta data that can be used.

2 Data Source.

A non-profit organisation called *Archive Team* (**link**) runs a web-scrapper that collects data from all tweets created on the social media platform called twitter. This data is collected into packets which can be accessed by using the following website, from which the raw tweet data of a specific month can be downloaded via a torrent, **here**.

Since the data set mentioned above is rather dense as it contains more meta data than we shall need for this investigation, a selection of reduce data sets have been produced and can be accessed via our **GitHub** page, along with all other files related to this paper, including all figures.

3 Data Wrangling methods

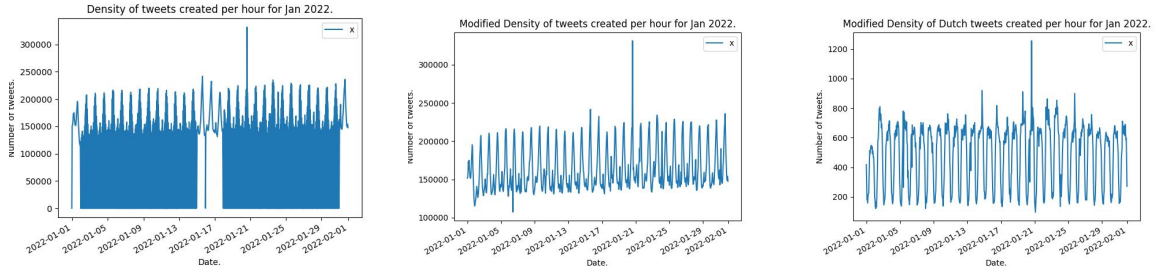
3.1 Getting the Data.

Using the link provided in the Data Source part, the raw twitter data for the given can be downloaded. This download file contains multiply `tar` files, which inside each then contains approximately 1440 `gz` compressed `json` files. The `json` files held more information than we required for this project, hence we need to extract only the two following fields *'created_at'* and *'lang'*, the first giving us the time the tweet was posted to the website and the second gives us information about the user language it was written in. Note that twitter supports 34 languages, with more details can be found via **twitter website**. In order to process all this downloaded information the python script `TwitterZip2DataFrame.py` was created, which can be found via our **GitHub/TwitterZip2DataFrame**. Due to the sheer size of the files being process extra care is needed to process them. Hence the python script unzips one of the tar files into a temporary file, from which a multiprocess function is called to process the gz jsons files. This is to speed up the process. The json are convert in pandas data frames then the data frames are stripped for the desired information, which are then concatenated with the other json files and saved to the disk as csv files. Once all json files are proccessed in the .tar file, the temporary file is cleared to allow space for the next .tar file, until all are done. The result is that approximately 100Gb of

raw twitter data can be filtered to under 7Gb of data, which is more user friendly. This processed data can be found at **GitHub/Jan22All**.

3.2 Processing and visualising the data.

The first task here is to get a better understanding of the data, in order to achieve this we wish to plot all the data into a line plot, with time along the x-axis and the density of tweets for the y-axis. Therefore we had to group all the tweet data into 1 hour bin of posting their time, this was achieved by using pandas to convert the timeformat into timestamp data types for the 'created_at' column and then using the pandas groupby function with specific options. The result of which is we have a density of tweets per the hour. This data was plot by using pandas line plot function, this can be seen in Figure [1*a*]. It become clear from the plot that the data contained outliers, namely data points that contained less than the expected number of tweets for that time. There are a number of reason for these outliers, such as the web-scrapper may not have been working correctly for that time period or that the website twitter was down itself. Noting that this would remove outliers below the line, while outliers above will be harder to distinguish from real data. The outliers in the data were removed, which resulted in a clear plot of the tweets, which can be seen in Figure [1*b*]. Furthermore, using the pandas groupby function we can split the data by the twitter supported language, in our case we shall be looking at the Dutch tweets 'nl'. Again removing outliers we can generate a good plot for all Dutch tweets, see Figure [1*c*].



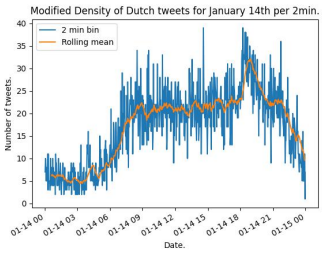
(a) Raw data. (b) Outliers removed. (c) Dutch data, outliers removed.

Figure 1: Line plots of density of tweets per 60min intervals.

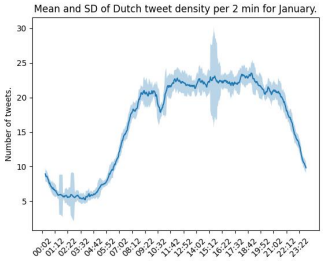
Since we are looking at discrete time events, we would like to have a better resolution of the density of tweets per a time period. We achieve this by selecting again only the Dutch tweets and then choosing a smaller time interval, in this case '2min', while doing so we also choose a smaller range of data, as we have indexed the dataframe by a Timestamp data type, hence we can focus on the day 14-01-2022. Plotting this we see large fluctuations in the data points, making the plot hard to read. The readability can be improved by using the pandas rolling function in alignment with the mean function, the result is a smooth rolling mean plot of the data, the results can be observed in Figure [2*a*].

As we want to test whether or not the behaviour of a specific day is in line with the expected tweet density in general. Hence, we take 2min interval over a 24 hour period again, which gives us 719 data points, then each day in the month is a column, thus each cell indicates the density for that time interval. Using this data frame we can then compute the mean and standard deviation of each time interval for the whole month. Noting again we have large fluctuations, hence we then take a rolling mean of both the mean and standard deviation in order to produce a smoother and more readable graph, see Figure [2*b*]. We use the standard deviation in association with the plot function fillbetween, which gives us a more visual indication of the confidence level of the data points.

4 Conclusion



(a) 2min intervals, with rolling mean.



(b) Rolling mean of the mean of each 2 min interval, with confidence itervals.