Advanced Data Science I Project: Hurricane Harvey Severity by Tweets

Ben Barrett

October 15, 2017

Introduction

ADD A FULL INTRODUCTION - ANY EVIDENCE OF TWEETS AND DISASTERS; USING TWEETS IN ANALYSES (OR OTHER SOCIAL MEDIA), WITH REFERENCES

These methods are designed to pull Twitter data to identify times and areas hardest hit by Hurricane Harvey. Geocoding the tweets primarily relies on the user location (entered once when a user first starts Twitter), rather than the location associated with an individual tweet, as most users turn the tweet location data off. In this case, however, these methods should be appropriate - as people who fled the impacted area before the storm hit will still have their user location linked to the impacted area. This mode of analysis relies on the assumption that the number of tweets about Hurricane Harvey coming out of an area correlates with the level of destruction in that area.

Data and Methods

Data

The Python library 'Tweepy' was used to connect to the Twitter Streaming API and download relevant tweets. The Python program, twitter_streaming.py (reproduced below), was adapted from code provided by Mikael Brunila (1), and used to live stream tweets. The stream was set to search for the hashtags 'HurricaneHarvey' and 'HurricaneHarveyRelief', and was started at 9:10AM on 9/1/2017. The live tweet stream was stopped at 9:56AM on 9/4/2017, which resulted in a program run time of 36 hours, 46 minutes and a total of 3,289,336 KB (3.290 GB) of Twitter data collected. This corresponds to 1,491.086 KB of data per minute, on average. All of the output was saved as a JSON file, twitter_data.json, available in my Dropbox.

Methods

Because of the large data file size, the twitter_data.json file was first streamed into R using a handler to randomly sample 25% of each page of JSON lines (total run time = 1.5 hours, code reproduced below). This code found 420,294 records, each corresponding to a tweet collected during the livestream. The random sample yielded a study dataset of 22,689 tweets. The reduced file serves as the basis for reproducibility.

Tweets were restricted to only those that had a user location recorded, which gave a sample of 15,535 tweets. Then, tweets were further restricted to those either with geographic coordinates already saved, those made in Texas or Louisiana (tweet location), those with a user location set within Texas or Louisiana, or those with a user location of geographic coordinates. This left 1,842 tweets, 21 of which had geographic coordinates already saved, and 80 of which had a location of tweet creation recorded (state and city). Next, user locations set as geographic coordinates were extracted as user geographic coordinates. Tweet locations and user locations that were assigned a state (Texas or Louisiana), but were not assigned a city, were removed from the dataset if a corresponding tweet or user geographic coordinate did not exist. Then, tweet locations that were assigned a city and state had their geographic coordinates imputed by assigning a random latitude and random longitude, bound between the respective city's most extreme border points, as identified by Google Maps. These imputed tweet location geographic coordinates were assigned as a tweet's coordinates if coordinates did not already exist from when the tweet was originally made. Following this, Google Maps was used to assess whether user geographic coordinates that had originally been set as the user location fell within Texas or Louisiana, and if not, these observations were deleted if a tweet geographic coordinate was not already assigned. User locations that were assigned a state and city then had their geographic coordinates

imputed using the same method as tweet locations, and the user geographic coordinates from the original user location, as well as the imputed user location geographic coordinates, were assigned as a tweet's coordinates if coordinates were not already assigned. This process left a final sample size of 1256, with each tweet having geographic coordinates and a city and state associated with it.

Bandwidth selected from the Berman and Diggle (1989) method, designed to minimize the mean squared error of the kernel smoothing estimator.

Results

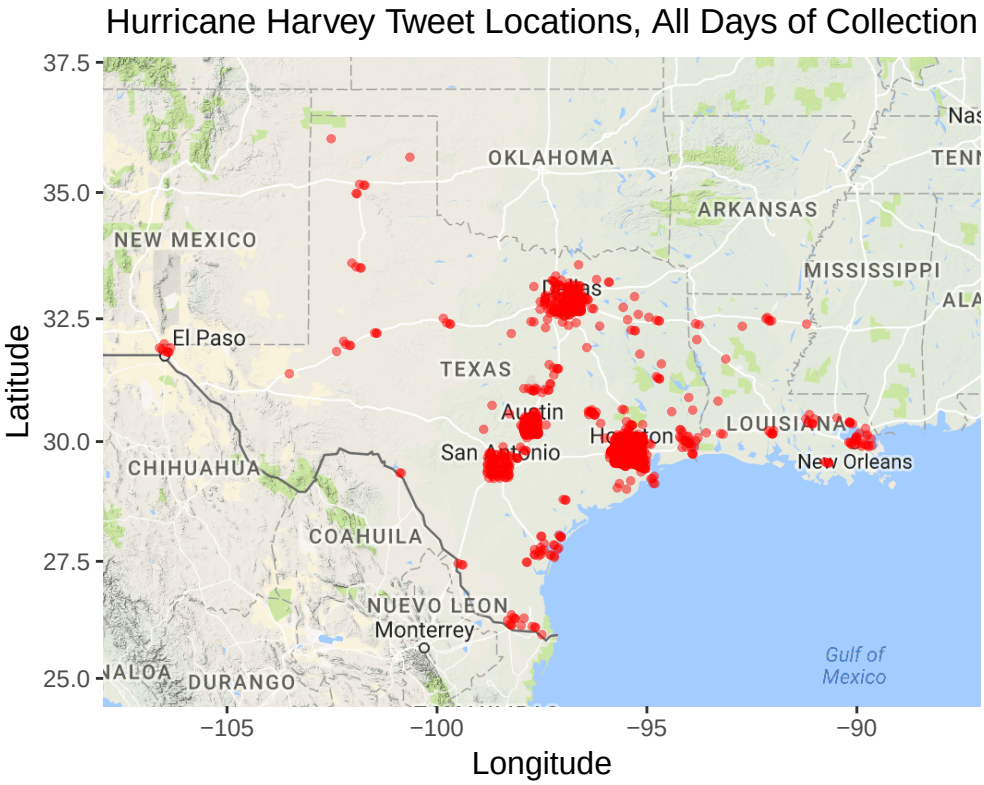


Figure 1: Locations of tweets made with #HurricaneHarvey or #HurricaneHarveyRelief, collected between 9:10AM on 9/1/2017 and 9:56AM on 9/4/2017.

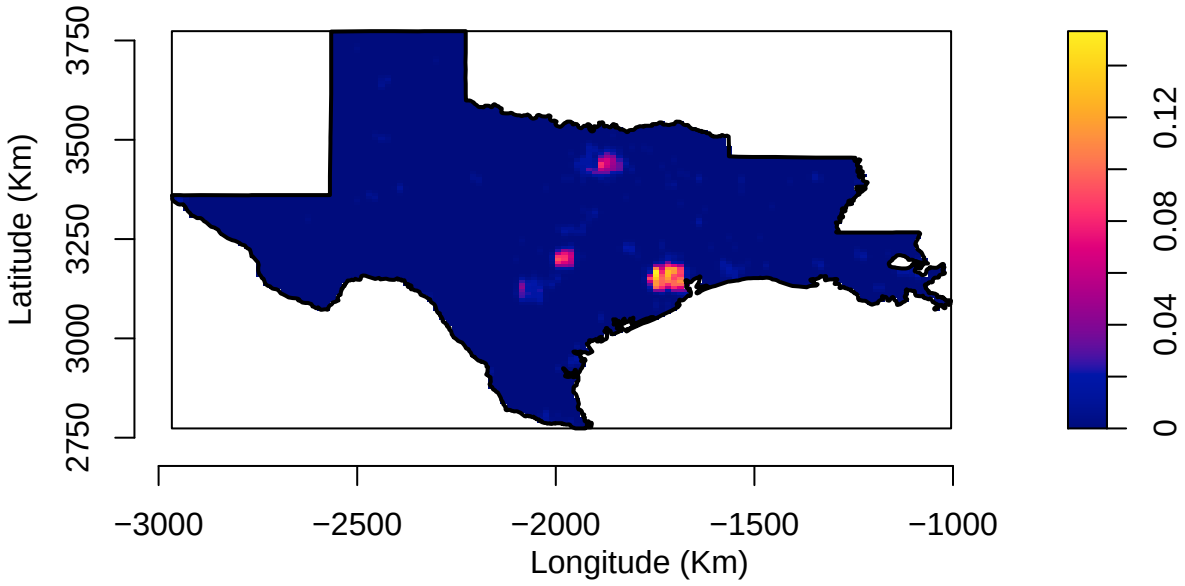


Figure 2: Spatial intensity of Hurricane Harvey tweets, calculated using kernel estimation with a bandwidth of 6.606245 kilometers.

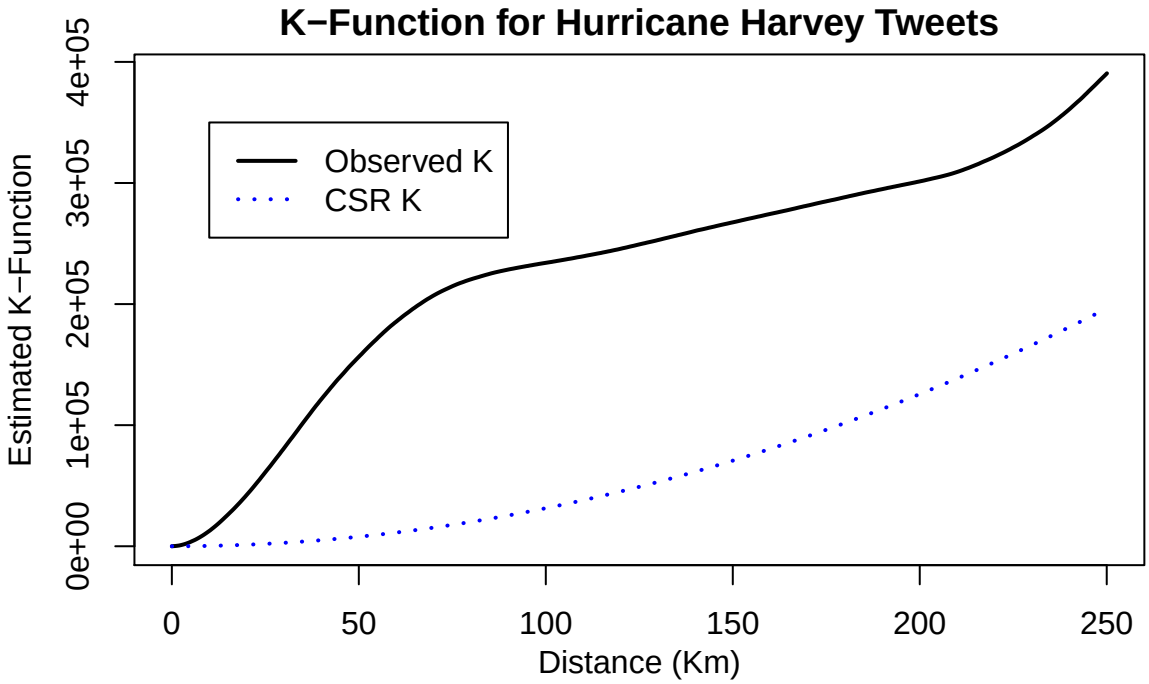


Figure 3: Estimated K−function for Hurricane Harvey tweets, comparing the observed clustering of tweets (black) to the clustering that would be expected under complete spatial randomness (CSR; blue).

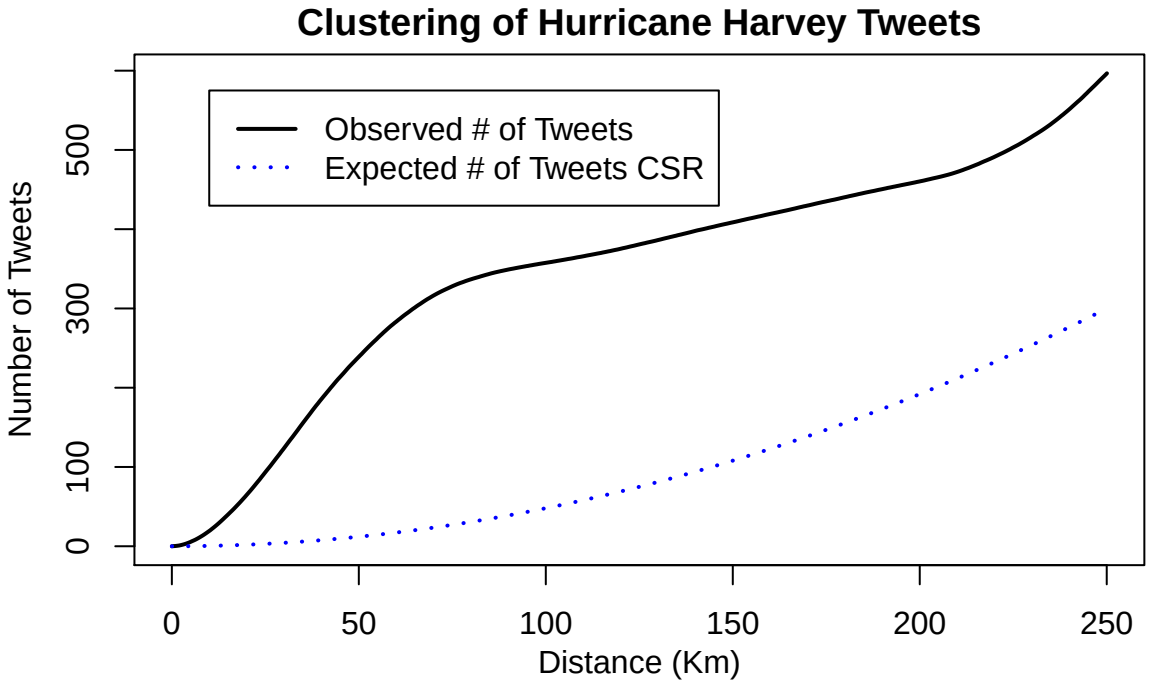


Figure 4: Observed clustering of Hurricane Harvey tweets (black), as compared to the number of tweets as would be expected under complete spatial randomness (CSR; blue).

CITE R PACKAGES AS WELL.

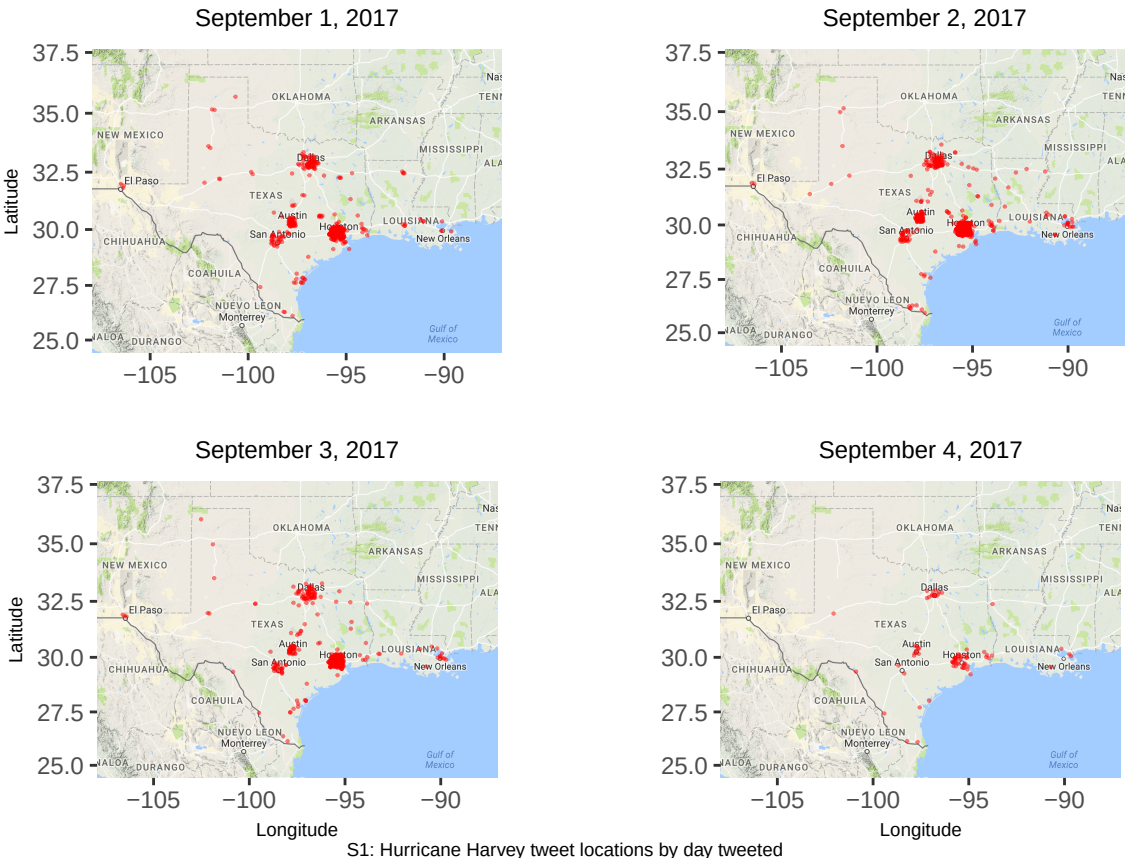
Discussion

Limitation: Cities imputed as rectangles - in future could trace city borders in Google Earth and import as shapefiles.

In the future: Sensitivity analysis - Assign new random seeds to the coordinate imputation and see if the same clusters still show up.

This analysis took place with tweets made at the tail-end of Hurricane Harvey. Would be interesting to "live track" a storm by collecting tweets from before it hit all the way through to the end to see if the tweet clusters follow the path of the storm.

Supplementary Material



S1: Hurricane Harvey tweet locations by day tweeted

References

1: Brunila, M. (2017). Scraping, extracting and mapping geodata from Twitter. http://www.mikaelbrunila.fi/2017/03/27/scraping-extracting-mapping-geodata-twitter/. [accessed September 1, 2017].