Tutorials

Tweetinvi Twitter Wrapper: https://github.com/linvi/tweetinvi/wiki

Google Maps Geocoding API Documentation: https://developers.google.com/maps/documentation/

Mongo DB General Tutorials: https://docs.mongodb.com/manual/tutorial/

Mongo DB C# Driver: https://docs.mongodb.com/ecosystem/drivers/csharp/

Scala Language Tutorials: http://www.tutorialspoint.com/scala/

Risks & Assumptions

Risks:

Location data is not valid, mitigate via google maps

Fault in data sourcing, mitigate by prioritising ASAP and providing two points of sourcing

Incongruent Data formats, mitigate via dialogue and proper testing

Raspberry PIs overheating, mitigate via cooling or distributing work among more PIs if needed

Front End charts being too intense for PI Processing, mitigate by minimising work visualisation app has to do

Assumptions:

We assume a rudimentary algorithm for sentiment assessment is "Good Enough", forgoing complex natural language inference

We assume that any incidental tweets sourced (such as EFF = Electronic Freedom Foundation) are minimal and acceptable

We assume that Twitter is providing us with either the full extent of our search criteria, or if this is not the case, then the sample we receive is a representational slice

We assume information that users provide on twitter (name of their location in particular) is correct and relevant, and that those who do not are in the minority

Failing the above, we assume that of those who do provide location based information, their distribution is representative of the norm

We acknowledge that our demographics are skewed (younger people more likely to use twitter, poorer South Africans less likely to use twitter)

We acknowledge that application software is not flawlessly optimised

Design Decisions: Data Collection

Reasons in ascending order of importance

1) Easiest task to fragment at the early stage, we can have standalone data sourcing, but we couldn’t have standalone transforming and visualising elements.

2) Provided a failsafe in the event that one source was not suitable or not reliable

3) Permitted a two streams to aggressively source data samples as soon as possible to see what data was available and how reliable each data piece was.

4) Provided alternative use cases: stream was for monitoring, has potential to grow into an automated feed into processing. Batch has "time window" focus for analytical purposes (eg looking at election primary / around big debate dates)

5) Emphasised an important premise of the project, that all sorts of technology stacks can be used to leverage the PIs: if we had more PIs we may have even made the data sourcing components run on them.