- In this document the purpose of the scripts included in this folder is explained along with comments on any pre-requisite steps if any.

List of scripts

1. youtubedataextraction.py
2. Twitter-analysis.py
3. Visualization-FanRating.Rmd
4. Visualization-twitterSentiments.Rmd
5. Visualization-youtube.Rmd

Youtubedataextraction.py - The purpose of this script is to extract comments of a defined list of youtube videos and extract sentiments of the comments extracted and aggregate all the sentiments of comments for a particular video to create an aggregated number called "Excitement Index". The list of youtube videos is compiled manually. We also extract the view count, like count, dislike count and comment count of the youtube videos in addition to the comments. All these extracted youtube statistics and sentiments for races are compiled into the youtube_sentiments.csv file with data about one race consisting in one row. The data about races is extracted from ergast api which provides data about the f1 races. http://ergast.com/mrd/

Prerequisites to run this script -

1. This is a python script which requires python installations to run. Python version 3.7.9 is used while developing this script. It uses some of the libraries which can be easily installed using pip (package manager – which often comes with your python installation)

2. You have to create an account in google cloud platform and enable the youtube data V3 api for use and also create necessary key to call the api from the script. You can find the documentation about the usage of the api and the necessary steps to be performed to call it from your script at https://developers.google.com/youtube/v3/docs/

3. This script uses stanford coreNLP to extract the sentiments of the comments which are extracted using the youtube api. Stanford CoreNLP is an NLP engine built using JAVA. In order to be able to use this, we have to first download the dump and all the additional model files for the language support. https://stanfordnlp.github.io/CoreNLP/download.html . You can run this by typing the following command in the command prompt (java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -annotators "tokenize,ssplit,pos,lemma,parse,sentiment" -port 9000 -timeout 100000). This starts a server to run at port 9000 and we send requests to this port in the python script and get the sentiments for the comments. This step needs to be performed before you run the python script.

Twitter-analysis.py - The purpose of this script is to read the extracted tweets and get the sentiments of the tweets using the stanford CoreNLP engine. The sentiments for the tweets for all the races collected are compiled into twitter_sentiments.csv file with data about one race consisting in one row. The data about races is extracted from ergast api which provides data about the f1 races. http://ergast.com/mrd/

1. This is a python script which requires python installations to run. Python version 3.7.9 is used while developing this script. It uses some of the libraries which can be easily installed using pip (package manager – which often comes with your python installation)
2. This script uses stanford coreNLP to extract the sentiments of the comments which are extracted using the youtube api. Stanford CoreNLP is an NLP engine built using JAVA. In order to be able to use this, we have to first download the dump and all the additional model files for the language support. https://stanfordnlp.github.io/CoreNLP/download.html . You can run this by typing the following command in the command prompt (java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -annotators "tokenize,ssplit,pos,lemma,parse,sentiment" -port 9000 -timeout 100000). This starts a server to run at port 9000 and we send requests to this port in the python script and get the sentiments for the comments. This step needs to be performed before you run the python script.

Visualization-FanRating.Rmd - This R markdown file is used to create the visualizations where we analyze the effect of number of overtakes occurred in a race (novertakes), number of drivers who did not finish the race (unfin) and the weather condition of the race if it is declared as a wet weather race or not (wetweather) on the rating of the race. The rating of the race is the rating given by the audience to a particular race.

Visualization-twitterSentiments.Rmd - This R markdown file is used to create the visualizations where we analyze the effect of number of overtakes occurred in a race (novertakes), number of drivers who did not finish the race (unfin) and the weather condition of the race, "if it is declared as a wet weather race or not" (wetweather) on the Excitement index of the race. The Excitement index of the race is calculated using the sentiments of the tweets which are related to individual races. Refer to the main function in twitter-analysis.py for details along with comments about the calculation of excitement index from sentiments.

Visualization-youtube.Rmd - This R markdown file is used to create the visualizations where we analyze the effect of number of overtakes occurred in a race (novertakes), number of drivers who did not finish the race (unfin) and the weather condition of the race, "if it is declared as a wet weather race or not" (wetweather) on the Excitement index of the race. The Excitement index of the race is calculated using the sentiments of the comments from youtube videos which are related to individual races. Refer to the main function in youtubedataextraction.py for details along with comments about the calculation of excitement index from sentiments.

twitterdataextraction.R - This R script is used to download the Tweets from the Twitter API. Further information on how to run the script is provided in the comments.

Note - Please ignore all the directory paths which are relative and replace them with paths that makes sense. The api-key which is used to call the api in youtubedataextraction.py is also removed from the code and requires to be replaced with a personal key.