****

**DBMS Final Project**

**Twitter Sentiment Analysis and Text GeoAnalysis**

**05/11/2015**

Pushkar Dharmatti N19909968 (pad354)

Aditya Varma Konda N17564723 (avk287)

Yashwin Sai Gokavarapu N10764622 (ss8163)

**What is Sentiment Analysis and why is it important?**

Sentiment Analysis is analyzing a given piece of text and providing a behavioral analysis with useful positive, negative or neutral values assigned to the text.

In technology driven world revolving around social marketing and media reception analytics, sentiment analysis helps entities and enterprises to have an aggregated view on a particular topic of interest or review on their enterprise.

e.g: People express their opinions on social media regarding a restaurant they visited and bombard it with comments about food, service quality, ambience etc. A centralized sentiment analysis will help the restaurant get an analysis on each domain and check for faults and areas of improvement.

Is it always accurate?

No. Sentiment analysis has some loop holes. For e.g, a person supporting Dallas Mavericks can tweet negative comments about the Rockets or LA Lakers, but this doesn’t help providing useful information in Mavericks support rating sentiment analysis as the negative tweet towards other teams were indirectly related to the persons support for a certain team.

There is a fine line between Sentiment analysis and Stance analysis- (favorable, unfavorable)

In this project we have attempted to achieve text based sentiment analysis and in tweet location based GeoAnalysis.

**Software Setup:**

Python

Tweepy API for tweet capture

Semantria API with added functionality for Sentiment Analysis and score generation and extract location (if available) from tweet data

Google Geocode API to extract full address, Latitude, Longitude corresponding to location

SQL module to capture results and load into normalized DB tables

Matplotlib to provide statistical analysis of location based average sentiment score and tweets

**Flow:**



Retrieve tweet details:

User ID

Tweet data

Source

Timestamp

Language

If location

Tweets sent to **Semantria** API to retrieve:

Sentiment score

Sentiment Value

Location

|  |
| --- |
| Tweets captured using **Tweepy API** |

If no

Store data into **Normalized Database** containing 3 tables:

Tweets Data

Sentiment Analysis

Location Analysis

If yes



Graph based representation of:

Location based sentiment analysis

Forward location to **Google Geolocation API** to fetch:

Latitude

Longitude

**Database Schema:**

|  |
| --- |
| **Twitter** |
| ID |
| Text |
| Source |
| Language |
| Location |

|  |
| --- |
| **Sentiment** |
| ID |
| Sentiment type |
| Sentiment score |

|  |
| --- |
| **Location** |
| Location |
| Latitude |
| Longitude |

**Tables created:**

Twitter (ID, Text, Source, Language, Location)

Sentiment (ID, sentiment, sentiment\_score)

Location (location, Latitude, Longitude)

**Code:**

1. **Tweet Capture:**

Tweets are captured using Tweepy into data1.txt which will be then fed into Semantria API to get the sentiment analysis and location for tweet text.



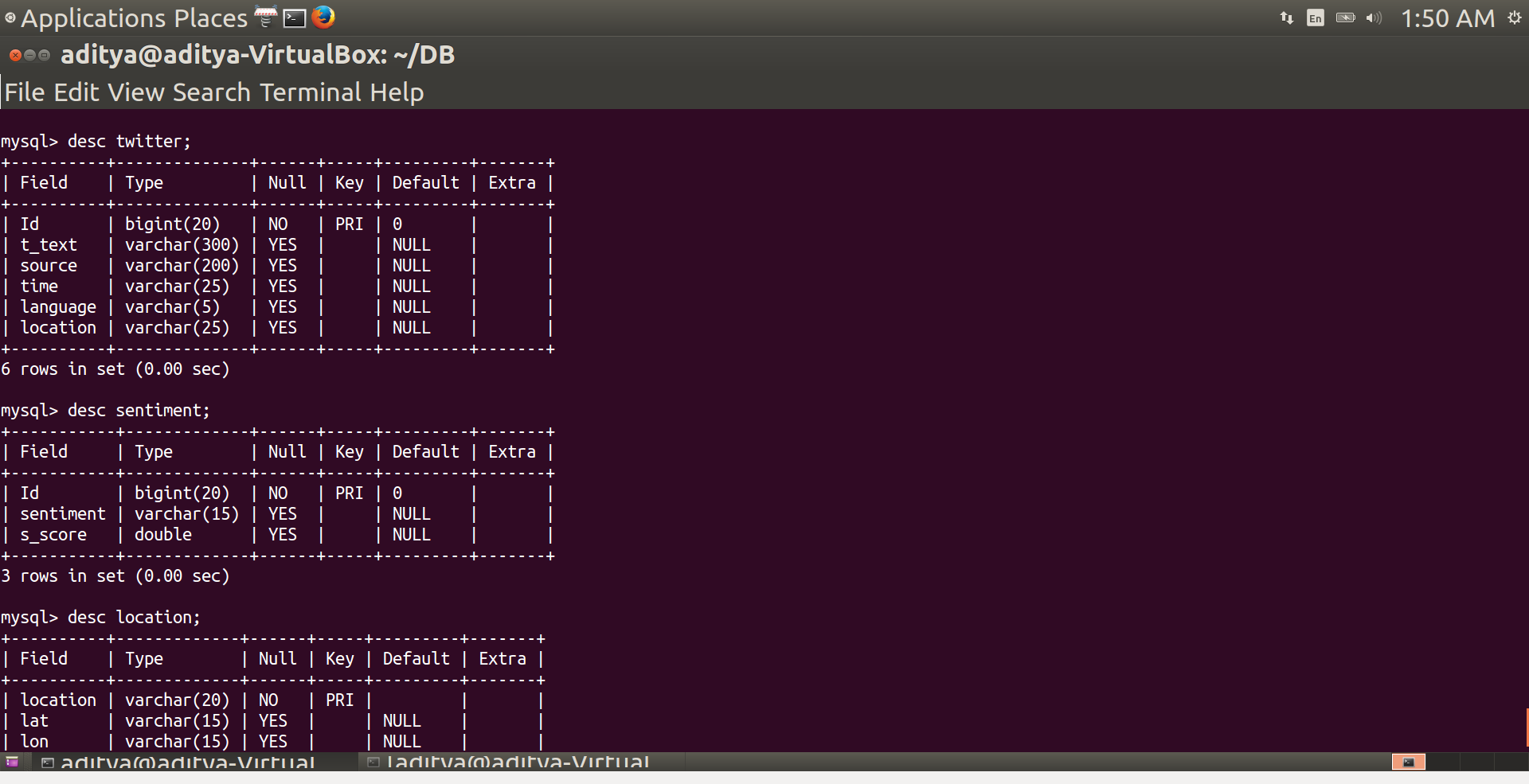
1. **Sentiment Analysis and Geotagging for location based analysis:**
2. Semantria provides sentiment score per tweet and places in form of entities by analyzing tweet text. The API was updated to get the sentiment type along with score to give an aggregate based analysis on the keywords of tweet which will then be used to provide a graphical analysis on general sentiment
3. For the locations fetched from tweet text, the location is forwarded to the google

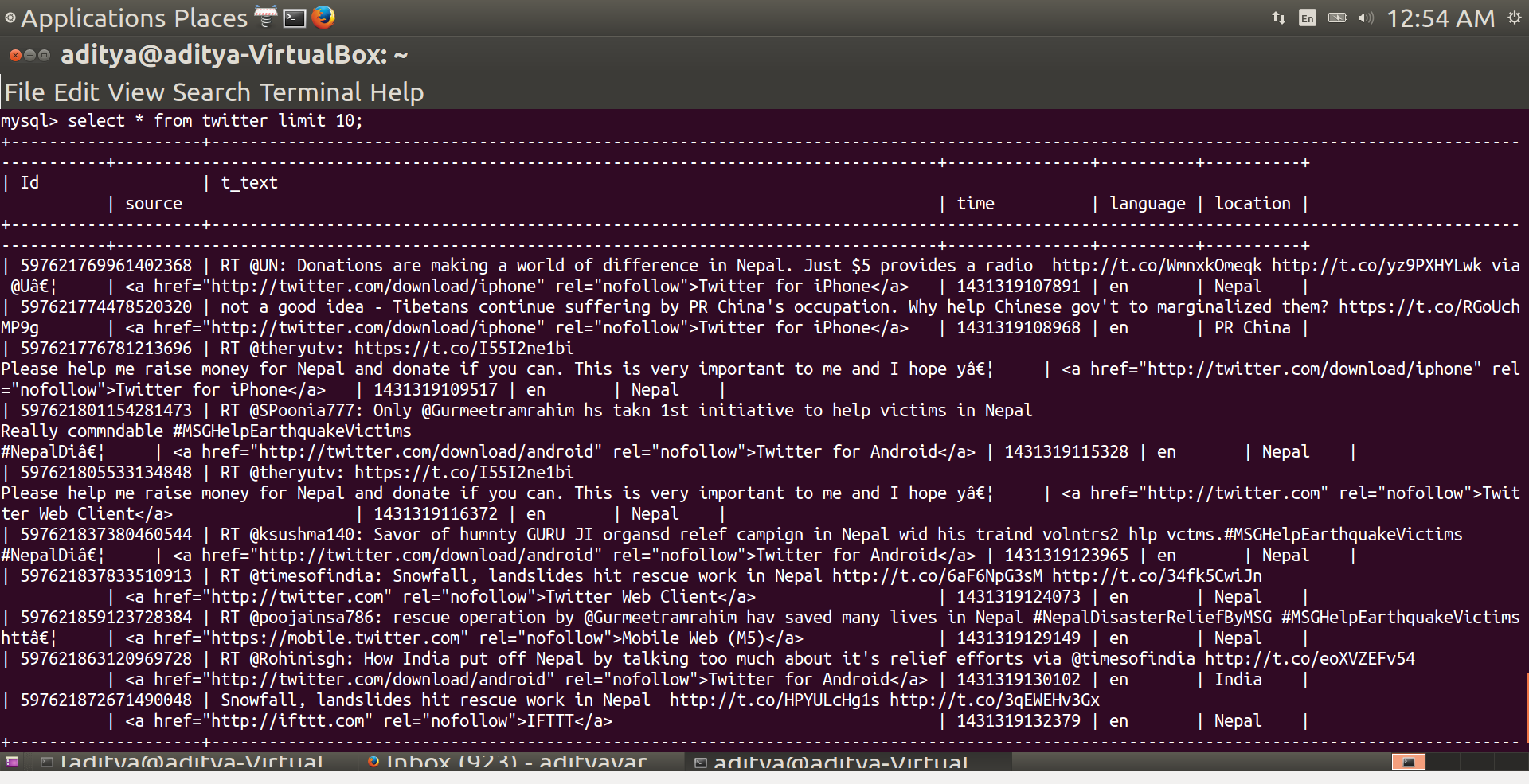
Geocoder API which provides the location latitude and longitude corresponding to its address parsed

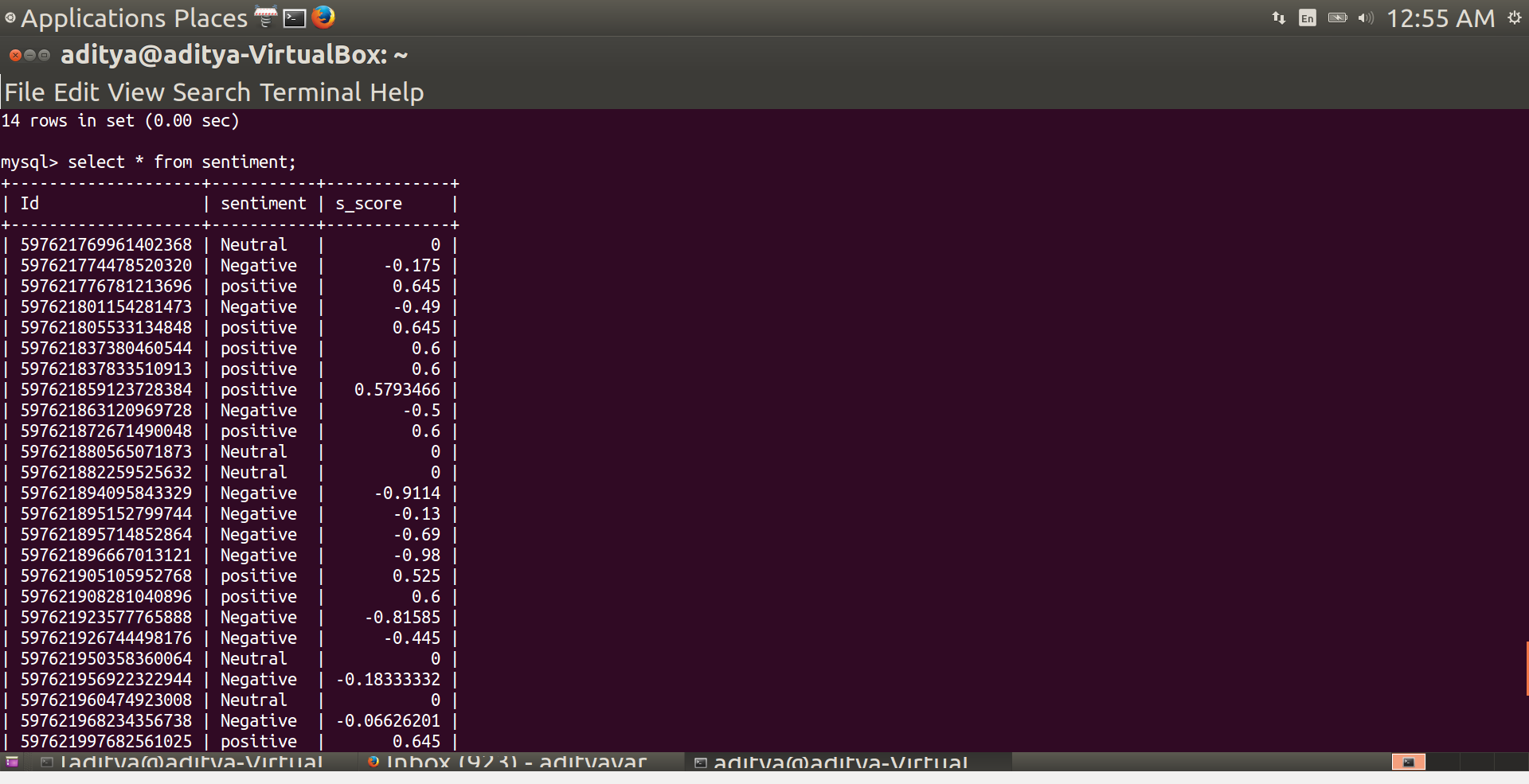
1. The resultant data from the above two codes is parsed through a SQLDB lib via localhost into Database schema described above for further analysis
2. Using Queries to get aggregated view of sentiment analysis, we used Matplotlib to finally graphically represent the data.

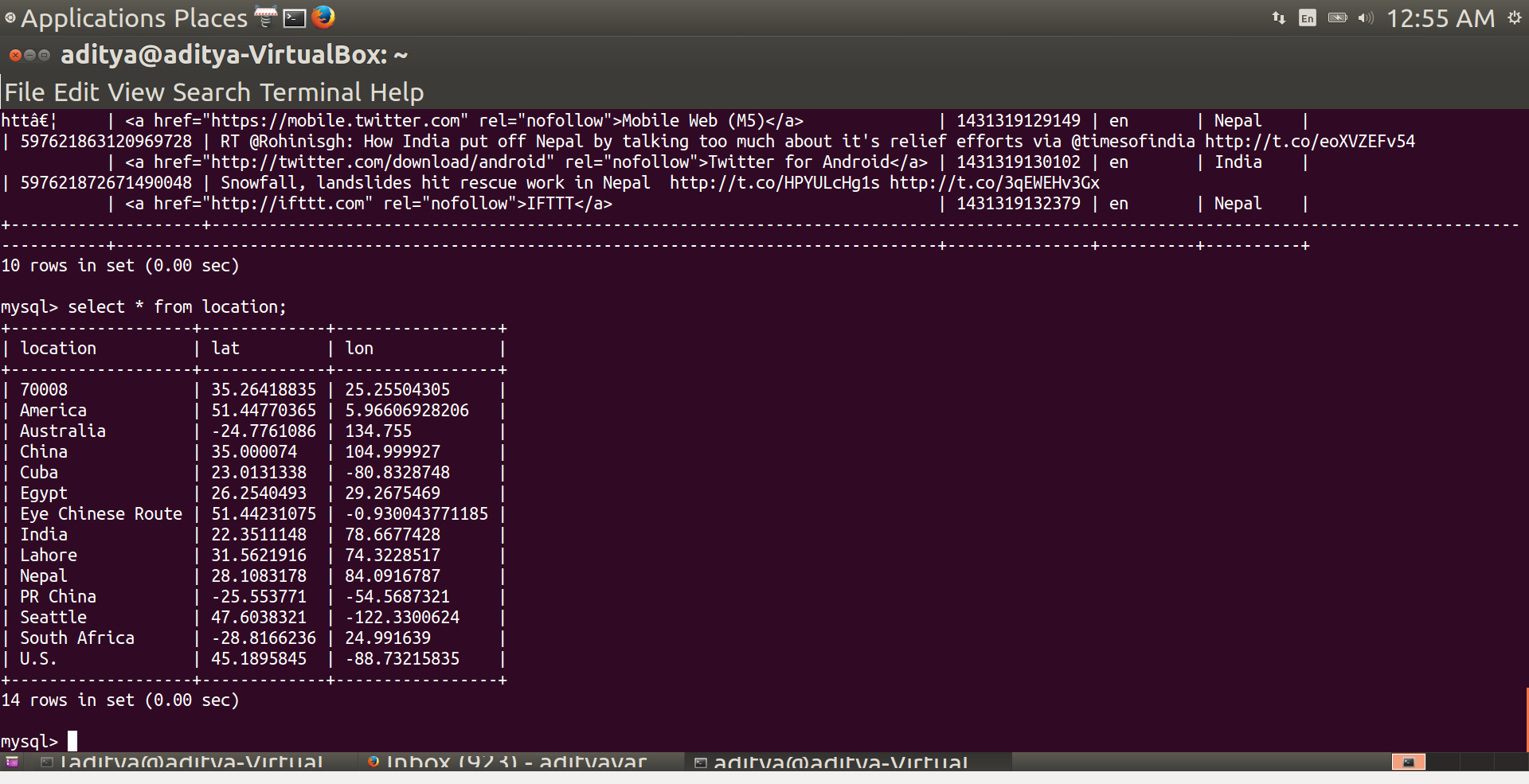


**Database:**









**Analysis Examples: Sample collected for 150-200 tweets:**

Location based Average Sentiment Analysis:

Q1 = cursor.execute(‘’’select Avg(s.s\_score) , count(t.Id) , t.location from twitter t , sentiment s where t.Id = s.Id group by t.location; ‘’’)

*Note: (cursor pointer on the graph peak will provide exact value of average sentiment score as observed in table results in MySQL)*

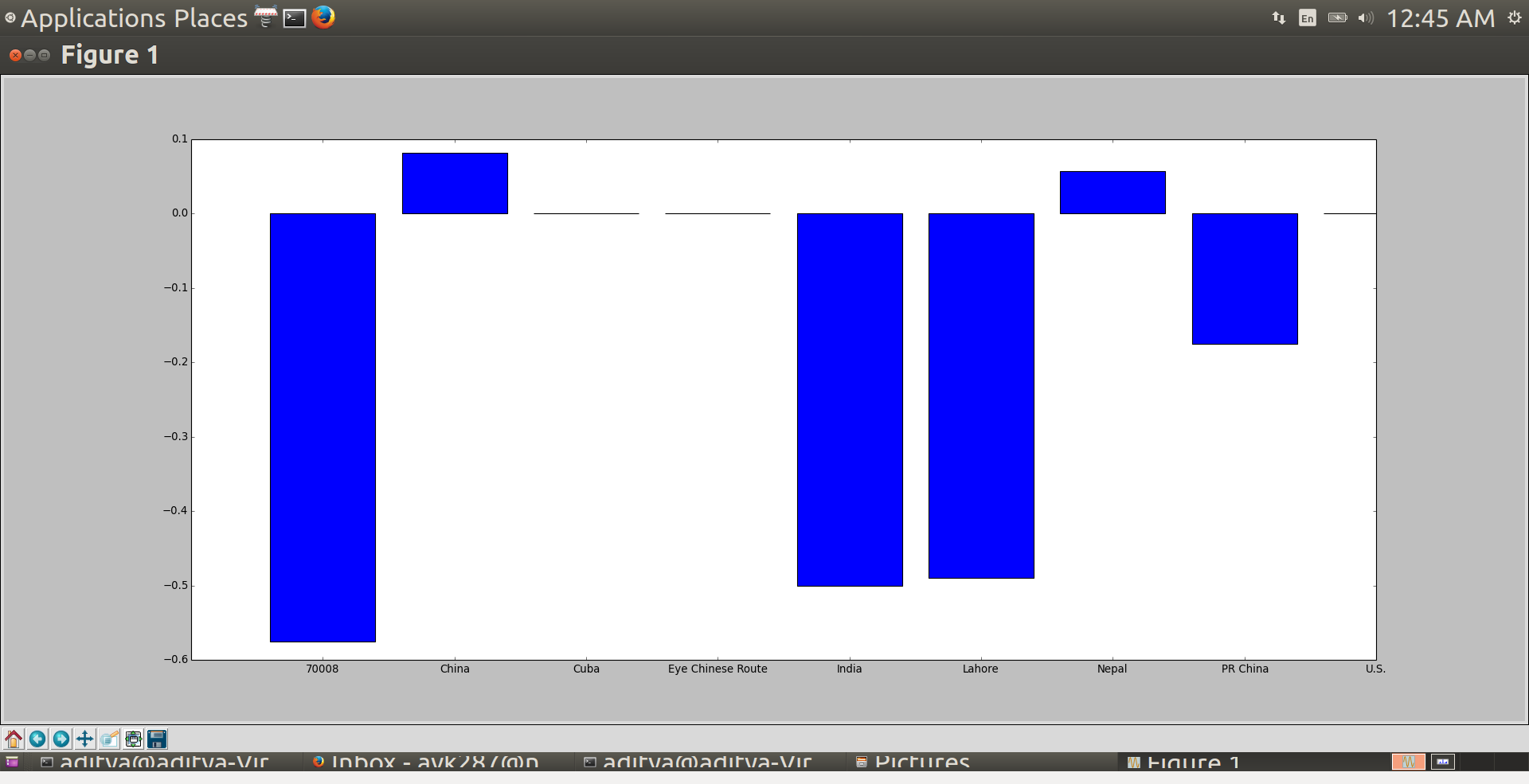
Tweets search keywords:

1.

Key1: Nepal

Key2: Earthquake

Key3: relief Funds

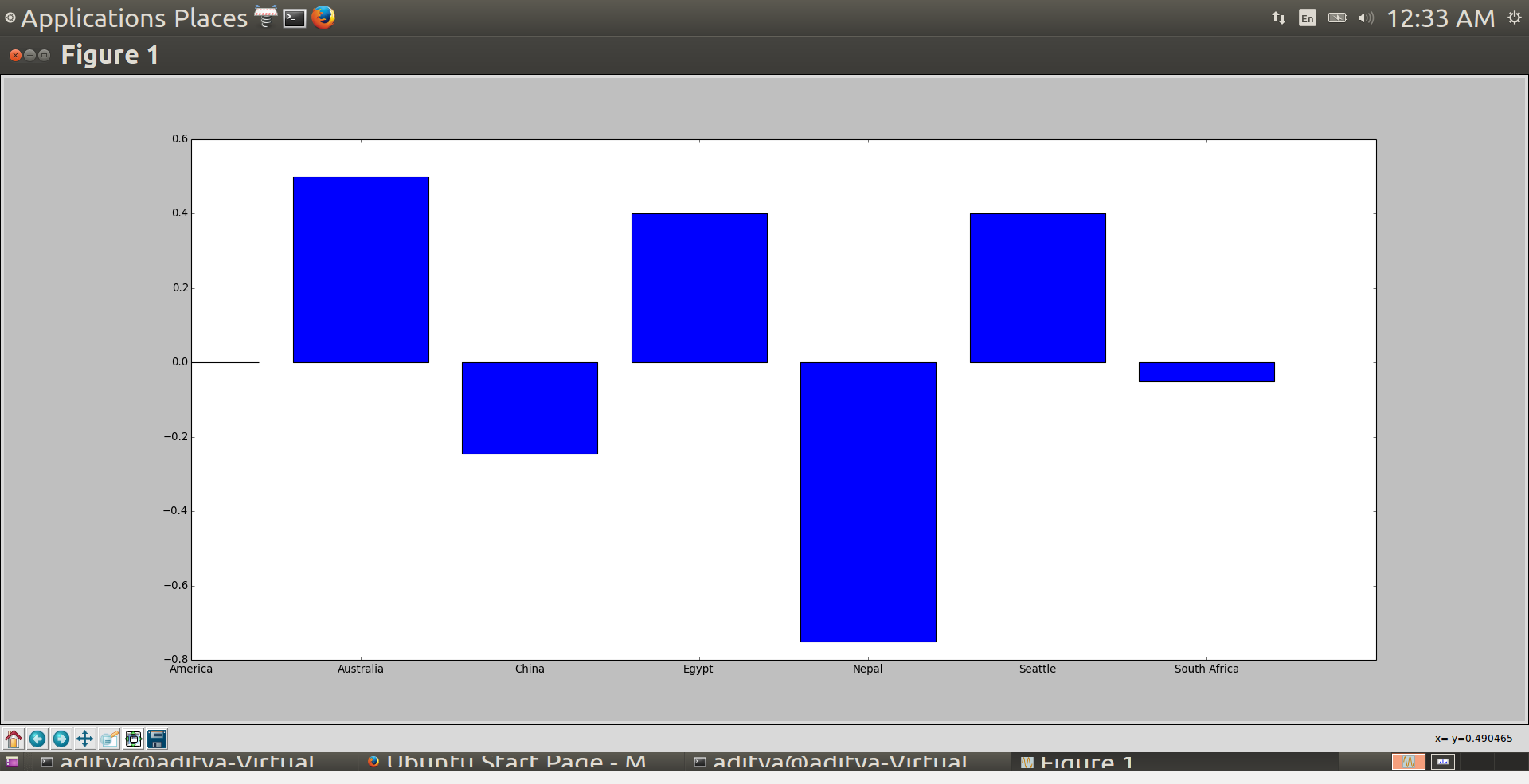


2.

Key1: Economy

Key2: Finance

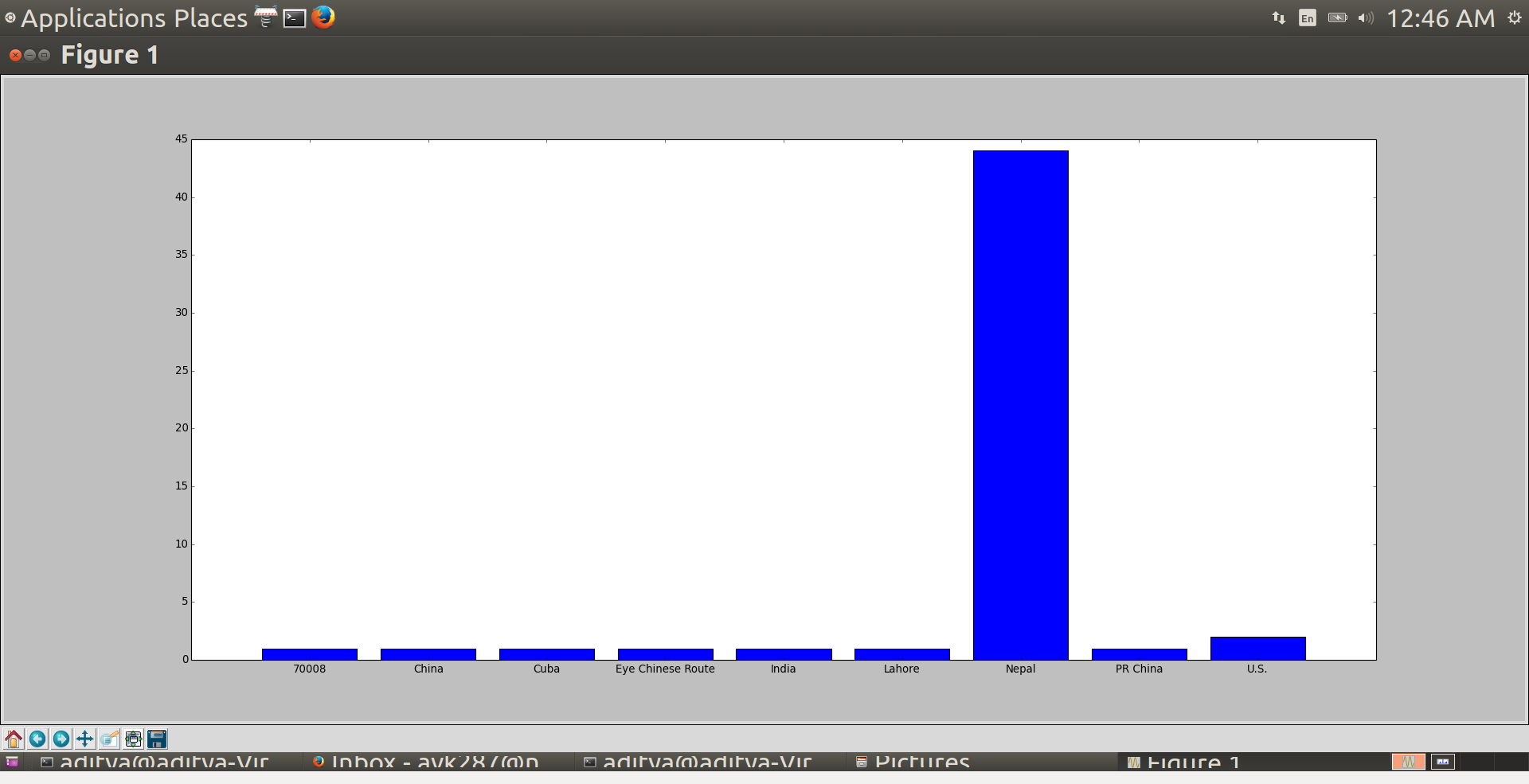
Key3: World



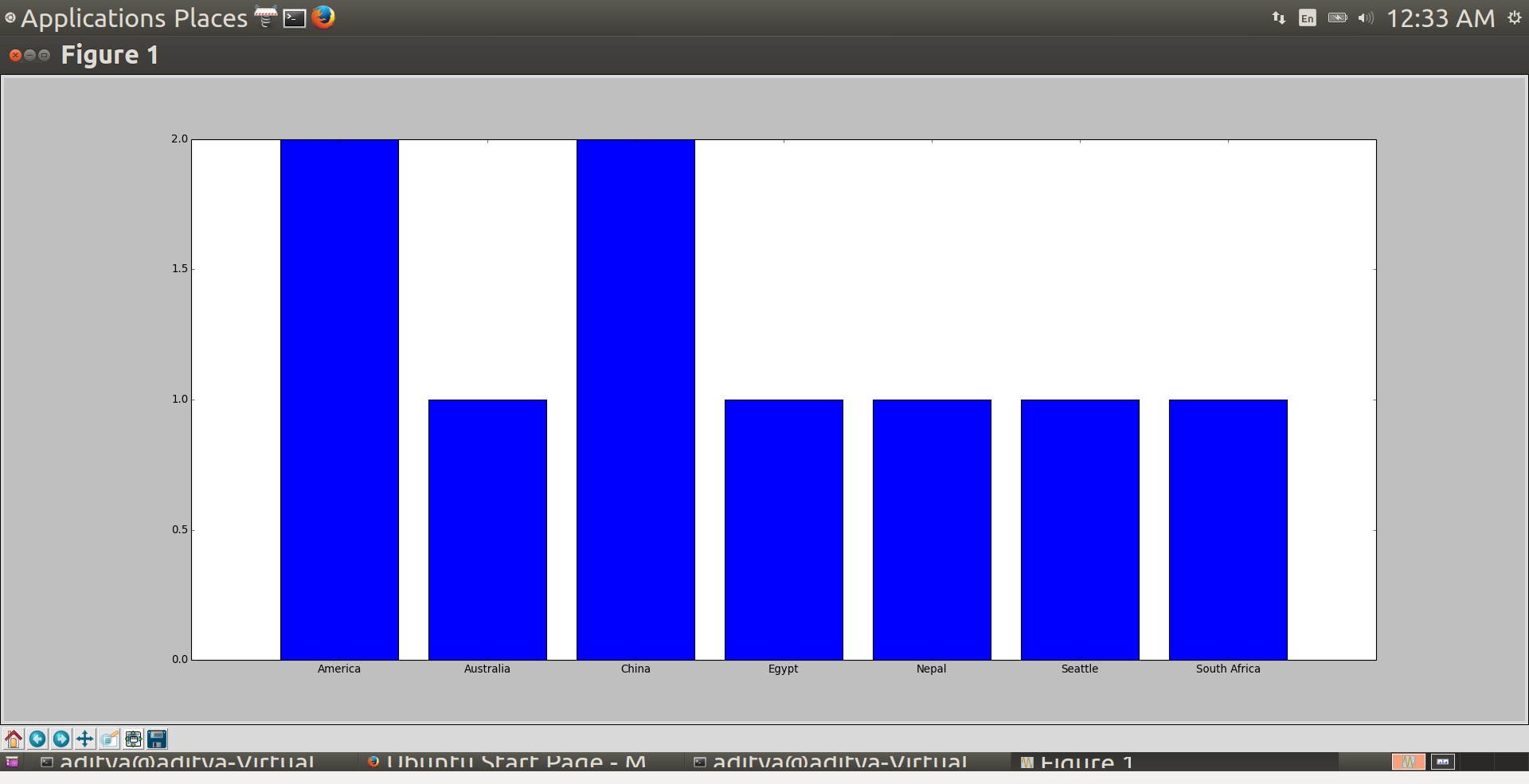
**Location based average tweets ratio:**

Q2 = cursor.execute(""" select Avg(s.s\_score) , count(t.Id) , t.location from twitter t , sentiment s where t.Id = s.Id group by t.location;""")

For 1:



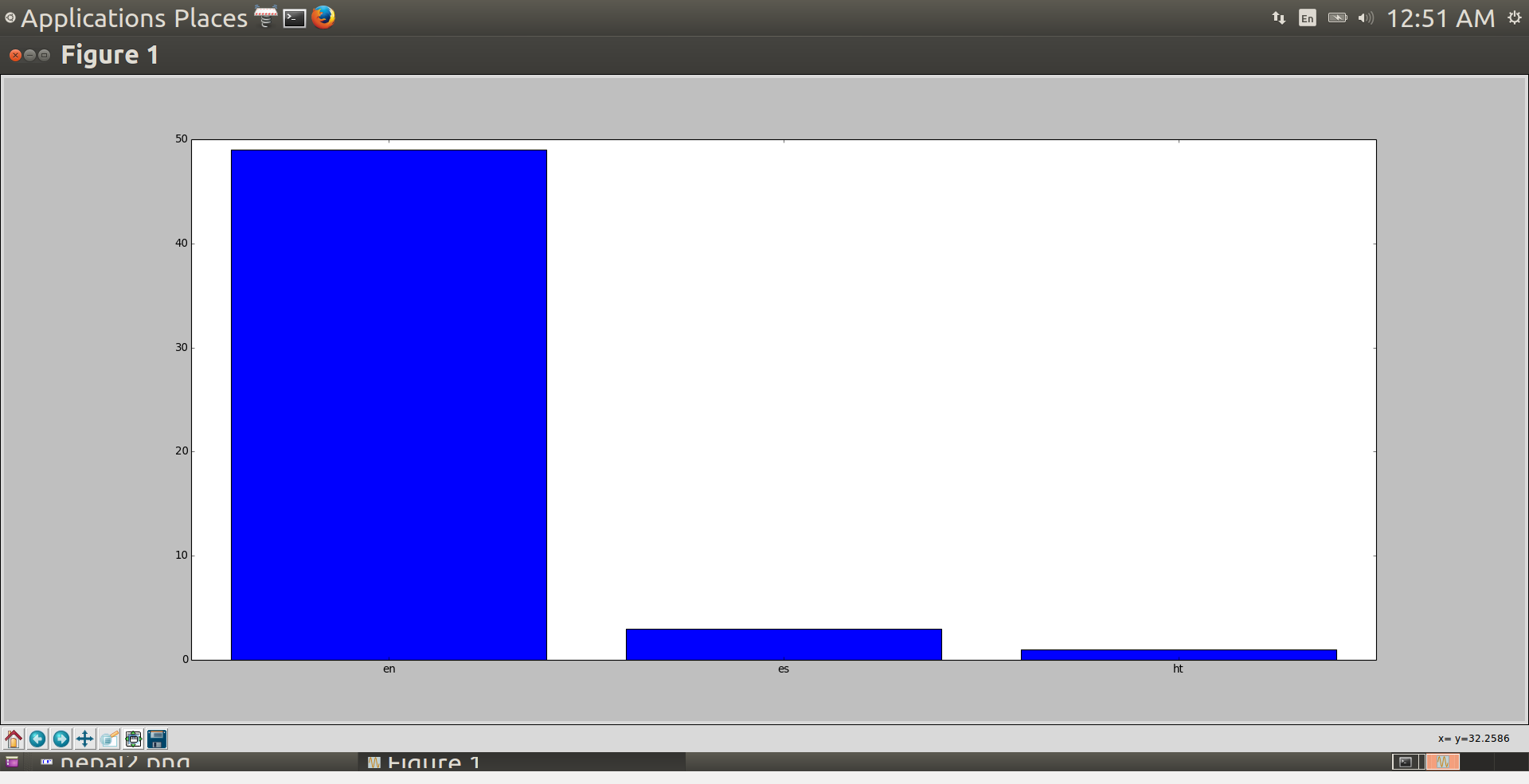
For 2:



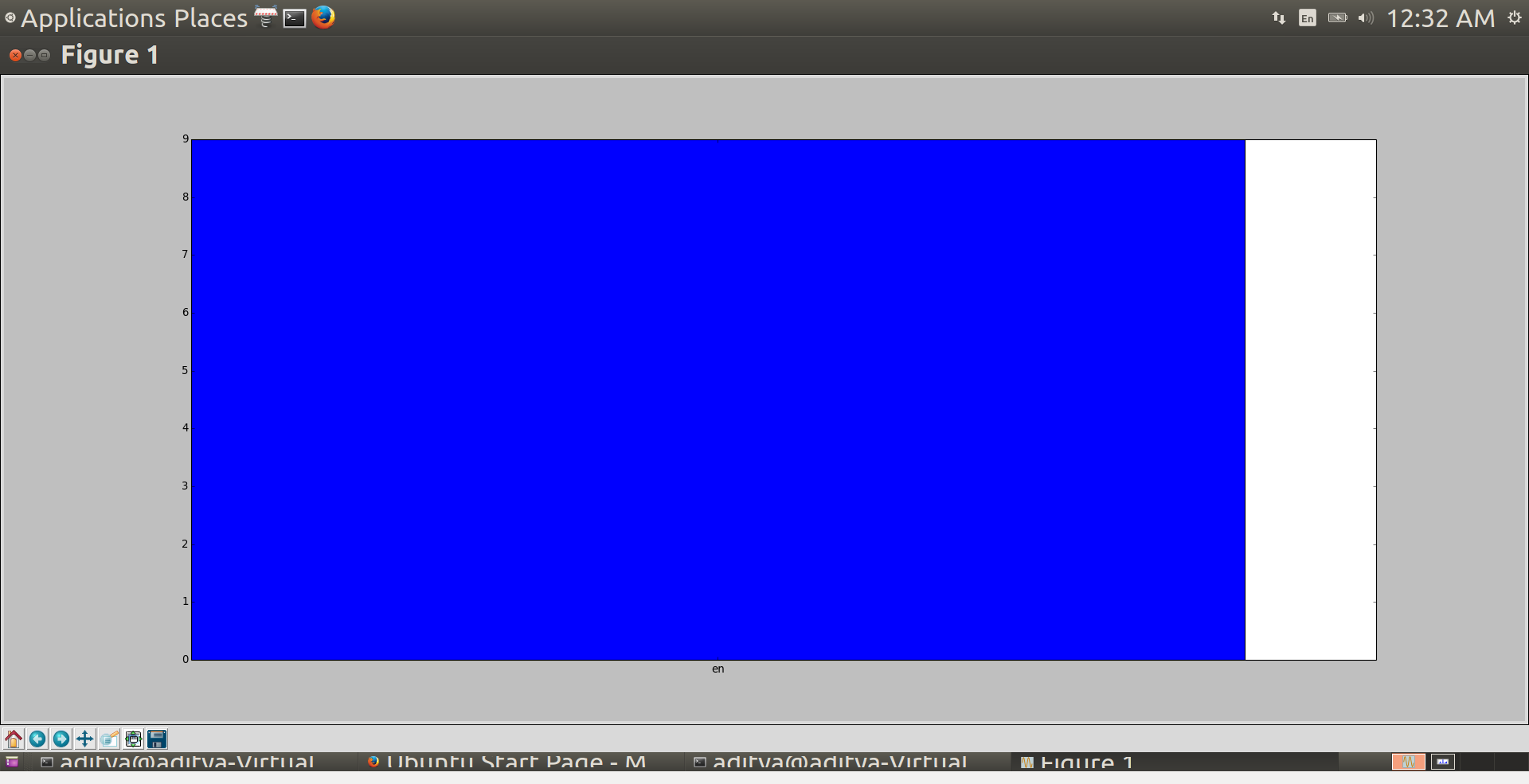
**Language based analysis of tweets:**

Q3 = Cursor.execute(""" select language , count(\*) from twitter group by language; """)

For 1:



For 2:



3.

Sample data for irregular tweets: 50 tweets

Key1: Travel USA

Key2: Brooklyn

Key 3: Europe

