

```

import pymongo
import geocoder
import csv
from datetime import datetime, timedelta
from email.utils import parsedate_tz

# Function to convert the Twitter date/time String

def to_datetime(time):
    time_tuple = parsedate_tz(time.strip())
    dt = datetime(*time_tuple[:6])
    return str(dt - timedelta(seconds=time_tuple[-1]))

def euclidean_distance(xCoordinate1, xCoordinate2, yCoordinate1, yCoordinate2):
    differenceX = xCoordinate1 - xCoordinate2
    differenceY = yCoordinate1 - yCoordinate2
    return pow(pow(differenceX, 2) + pow(differenceY, 2), 0.5)

# Function to search collection based on a query

def load_from_mongo(mongo_db, mongo_db_coll, return_cursor=False, criteria=None, projection=None, **mongo_conn_kw):

    # Can use critera and projection to limit the data that is returned
    # look at http://docs.mongodb.org/manual/reference/method/db.collection.find/

    # Can use MongoDB's aggregations framework for more detailed queries

    client = pymongo.MongoClient(**mongo_conn_kw)
    db = client[mongo_db]
    coll = db[mongo_db_coll]

    tweet_iterator = coll.find()
    print tweet_iterator
    with open('outputs/geoWorld.csv', 'wb') as geoworldFile:
        my_file_writer = csv.writer(geoworldFile)
        my_file_writer.writerow(['latitude', 'longitude', 'userlat', 'userlng', 'difference', 'user'])
        for tweet in tweet_iterator:
            date_created = tweet['date_tweeted']
            user = tweet['user']['name'].encode('ascii', 'ignore').decode('ascii')
            text = tweet['text'].encode('ascii', 'ignore').decode('ascii')
            userlocation = tweet['user']['location'].encode('ascii', 'ignore').decode('ascii')
            tweetLatitude = tweet['coordinates']['coordinates'][1]
            tweetLongitude = tweet['coordinates']['coordinates'][0]
            g = geocoder.google(userlocation)
            if len(g.latlng) != 0:
                userLatitude = g.latlng['lat']
                userLongitude = g.latlng['lng']
                differenceOfLocation = euclidean_distance(userLatitude, tweetLatitude, userLongitude, tweetLongitude)
            else:
                userLatitude = 0
                userLongitude = 0
                differenceOfLocation = euclidean_distance(userLatitude, tweetLatitude, userLongitude, tweetLongitude)

            my_file_writer.writerow([tweetLatitude, tweetLongitude, userLatitude, userLongitude, differenceOfLocation, user])

load_from_mongo('geoWorldStreamDatabase', 'geolocated')

```