Assignment Solution-2

```
import json
from pprint import pprint
import nltk
import numpy as np
import re
from datetime import datetime
from collections import OrderedDict
import pandas as pd
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt
import codecs

# For connecting MongoDB
from pymongo import MongoClient
client = MongoClient()
db = client['TwitterStream']

# Ques1. Show the collections in the TwitterStream database
print(db.collection_names())

# Ques2. Count the total number of tweets in the particular
collection(december4)
print("total no of tweets")
print(db.december4.count())

# Ques3. Print the top tweet
pprint(db.december4.find_one())

# Ques4. Print the fields of document
print(db.december4.find_one().keys())

# Ques5. Print the text of top 10 tweets with posting time
tweetsCol = db.december4.find().limit(10)
count=1
for tweet in tweetsCol:
        count+=1
        print(count)
        pprint(tweet['text'])
        print(tweet['created_at'])
```

 # Ques6. Print the most popular 50 hashtags and plot the scattar, bar and wordcloud diagram to represent most popular hashtags.

```
tweetsCol1 = db.december4.find()
hashtag_list={}
sorted_hashtags={}
for tweet in tweetsCol1:
        hts = tweet['entities']['hashtags']
        for hinfo in hts:
          h = hinfo['text']
          # add hashtag to list
          hashtag_list[h] = 1 + hashtag_list.get(h,0)

sorted_hashtags=OrderedDict(sorted(hashtag_list.items(), key=lambda x:x[1],
reverse=True))
names=[]
values=[]
c1=0
for ht in sorted_hashtags.items():
        if c1<50:
                c1+=1
                print("," + str(ht))
                names.append(ht[0])
                values.append(ht[1])

# Scattar plot
plt.scatter(list(range(50)), values, c='r', label='hashtags')
plt.savefig('scatter.png')
plt.show()

# Bar plot
plt.bar(range(50),values,tick_label=names)
plt.xticks(rotation=50)
plt.xlabel("hashtags")
plt.ylabel("frequency")
plt.savefig('bar.png')
plt.show()

# WordCloud plot
wordcloud=WordCloud(max_font_size=30, max_words=50,
background_color="white").generate_from_frequencies(hashtag_list)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.savefig('wordcloud.png')
plt.show()
```

 # Ques7. Print the top 50 retweeted tweets

```
tweetsCol1 = db.december4.find()
retweets = {}
count=0
for tweet in tweetsCol1:
        #print(tweet['text'])
        if 'retweeted_status' in tweet:
                if(count<50):
                        count+=1
                        rt = tweet['retweeted_status']
                        retweets[rt['id_str']] = rt
# convert to list
retweets = [retweets[w] for w in retweets.keys()]
   # sort by retweet count
retweets.sort(key=lambda x: -x['retweet_count'])
   # display top k retweets
for t in retweets:
        print(t['text'])
        print(t['retweet_count'])
```

 # Ques8. Print the top 10 users based on user frequency

```
user_list={}
sorted_users={}
c=1
for tweet in tweetsCol1:
        u = tweet['user']['screen_name']
        user_list[u] = 1 + user_list.get(u, 0)

sorted_users=OrderedDict(sorted(user_list.items(), key=lambda x:x[1],
reverse=True))
for ht in sorted_users.items():
        if c<10:
                c+=1
                print("," + str(ht))
```