1 Finding Trends

1.1

In [24]:

```
# Read txt
with open("test_set_tweets.txt","r", encoding='utf-8') as file:
    lines = [next(file) for x in range(500000)]
```

In [9]:

```
import re

def extractHashtags(string):
    pattern = re.compile(r"#(\S+)")
    strs = re.findall(pattern, string)

    pattern = re.compile('[^a-zA-Z]')
    output = []
    for i in strs:
        output.append(pattern.sub('', i.lower()))
    return output

# Example:
extractHashtags("22077441       10470781081     #confession.    I can't live with my mama!!! Especially if I don't have my own car!      2010-03-14 09:21:58")
```

Out[9]:

```
['confession']
```

In [10]:

```
def mapper_hashtags_line(line):
    words = extractHashtags(line)
    output = []
    for word in words:
        if word:
            output.append((word,1))
    return output

# Example:
mapper_hashtags_line("22077441  10470781081     #confession.    I can't live with my mama!!! Especially if I don't have my own car!      2010-03-14 09:21:58")
```

Out[10]:

```
[('confession', 1)]
```

In [15]:

```
def mapper_hashtags(lines):
    output = []
    for line in lines:
        list = mapper_hashtags_line(line)
        if list:
            output += list
    return output

#Example:
test = [″#John. 2010″, ″#Jerry 2011″, ″#Tom 2012″, ″#Jerry 2013″]
mapper_hashtags(test)
```

Out[15]:

```
[(’john’, 1), (’jerry’, 1), (’tom’, 1), (’jerry’, 1)]
```

In [16]:

```
def combiner_heshtags(mapper_output):
    groups = {} # group by key values
    for item in mapper_output:
        k = item[0]
        v = item[1]
        if k not in groups:
            groups[k] = [v]
        else:
            groups[k].append(v)
    return groups

#Example:
combiner_heshtags(mapper_hashtags(test))
```

Out[16]:

```
{’john’: [1], ’jerry’: [1, 1], ’tom’: [1]}
```

In [17]:

```
def reducer_heshtags(keyWord, counts):
    return (keyWord, sum(counts))

reducer_heshtags( ’jerry’,[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

Out[17]:

```
(’jerry’, 14)
```

In [25]:

```
def execute_heshtags(lines):
    groups = combiner_heshtags(mapper_hashtags(lines))
    output = [reducer_heshtags(k,v) for k,v in groups.items()]
    output.sort()
    return output

hashtags_freq = execute_heshtags(lines)
```

In [33]:

```
def Sort(orig):

    orig.sort(key = lambda x: x[1], reverse = True)
    return orig

print(Sort(hashtags_freq)[:10])
```

```
[('ff', 3581), ('nowplaying', 1809), ('fb', 1402), ('mm', 1029), ('fail', 686),
('random', 622), ('haiti', 591), ('shoutout', 529), ('followfriday', 457), ('music
monday', 452)]
```

In [35]:

```
import timeit

start = timeit.default_timer()
hashtags_freq = execute_heshtags(lines)
print(Sort(hashtags_freq)[:10])
stop = timeit.default_timer()
print('Time: ', stop - start)
```

```
[('ff', 3581), ('nowplaying', 1809), ('fb', 1402), ('mm', 1029), ('fail', 686),
('random', 622), ('haiti', 591), ('shoutout', 529), ('followfriday', 457), ('music
monday', 452)]
Time:  1.6265050999999175
```