

# Twitter Data Analysis

David Wilson & Grant Haataja

April 30, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Goals of Analysis</b>	<b>3</b>
<b>3</b>	<b>Methodology</b>	<b>3</b>
<b>4</b>	<b>Twitter API</b>	<b>3</b>
<b>5</b>	<b>Code to parse .json files</b>	<b>3</b>
<b>6</b>	<b>Tweet Examples</b>	<b>4</b>
<b>7</b>	<b>Code to Create Graphs</b>	<b>5</b>
<b>8</b>	<b>The Data</b>	<b>6</b>
<b>9</b>	<b>Conclusions</b>	<b>7</b>
<b>10</b>	<b>Further Research</b>	<b>7</b>

# Introduction

A growing concern of social media is the amount of profanity and inappropriate content appearing to the public. This paper will address this issue by outlining data analysis for profanity on Twitter.

## Goals of Analysis

The goals for this analysis were to test the occurrence of profanity from a sample of tweets, to determine whether Twitter was more than 50% profane or not. But more interestingly, the goal was to find the proportion of tweets that contain profanity.

## Methodology

\*\*\*David Write your stuff here\*\*\*

## Twitter API

\*\*\*David Write your stuff here\*\*\*

## Code to parse .json files

```
1 import json
2
3 jsonNum = 100
4 tweets = []
5 profanity = []
6
7 # Load profanity list
8 fprofanity = open('Profanity.txt', 'r')
9 for text in fprofanity:
10     profanity += [text.strip().lower()]
11
12 # Load tweets
13 for i in range(1, jsonNum+1):
14     temp = open('word' + str(i) + '.json', 'r')
15     temp = json.load(temp)
16     for tweet in temp['statuses']:
17         tweets += [tweet['text'].lower()]
18
19 profaneTweets = []
20 uses = {}
21 for prof in profanity:
22     uses.update({prof:0})
```

```

23
24 for tweet in tweets:
25     for word in profanity:
26         if ' ' + word + ' ' in tweet:
27             uses.update({word: uses[word]+1})
28             if tweet not in profaneTweets:
29                 profaneTweets += [tweet]
30
31 print('\tProfane Tweets:')
32 print(str(len(profaneTweets)) + '/' + str(len(tweets)))
33 print('\n\tUses:')
34 for use in uses:
35     print(use + ': ' + str(uses[use]))

```

The following picture shows the output of the above code when ran on the 100 json files that were created from the queries.

```

grant@DarkWolf: ~/undClasses/csci260/wordCountingApp/Twurl Requests
File Edit View Search Terminal Help
grant@DarkWolf:~/undClasses/csci260/wordCountingApp/Twurl Requests$ py jsonreader.py
Profane Tweets:
37/1136

Uses:
ass: 2
asses: 1
asshole: 0
bitches: 7
damn: 3
fuck: 7
fucker: 0
fucks: 2
fucking: 9
cunt: 0
cock: 0
cocksucker: 0
shit: 13
shits: 1
shitty: 0
grant@DarkWolf:~/undClasses/csci260/wordCountingApp/Twurl Requests$

```

## Tweet Examples

In this section, several of the tweets that were searched are shown. The program did not output the entirety of some of the longer tweets. Note that many of them have multiple occurrences of different profane words, so some of them may be counted twice.

---

rt @quackityhq: fuck koalas why can those little shits sleep 22 hours a day and be seen as cute but when i do it im seen as a lazy piece of...

---

rt @braindumtweets: trying to figure out how sex works but it doesn't seem plausible, how am i supposed to fit my entire ass in a woman's...

---

adding arya to my muted words because you bitches can't be trusted.

---

rt @iamvee.: i haven't been this happy in about 3 fucking years boy

---

he appears healthy but he seems just to be very damn tired. i'll keep an eye on him and call an animal sanctuary later

---

if i could get my tubes tied today i fucking would. i'd sell every single one of my eggs. fuck dem kids.

---

waw i already miss them ;(( my spoiled ass need to stop i keep missing them even if they feed me so much

---

## Code to Create Graphs

Shown below is the Python code used to generate the pie chart used to display the data results.

```
1 from matplotlib import pyplot
2 import numpy
3
4 #define the pie chart
5 labels = ['Ass', 'Asses', 'Bitches', 'Damn', 'Fuck', 'Fucks', 'Fucking', 'Shit', 'Shits']
6 sizes = numpy.array([2, 1, 7, 3, 7, 2, 9, 13, 1])
7 colors = ['yellow', 'orange', 'aquamarine', 'purple', 'dodgerblue', 'gray', 'red', 'green', '
           mediumvioletred']
8
9 #do the plot
10 pyplot.pie(sizes, labels=labels, colors=colors,
11 autopct='%1.1f%%', startangle=350)
12
13 pyplot.axis('equal')
14 pyplot.show()
```

The following is the code to generate the bar chart containing the actual number of tweets with the queried profanities.

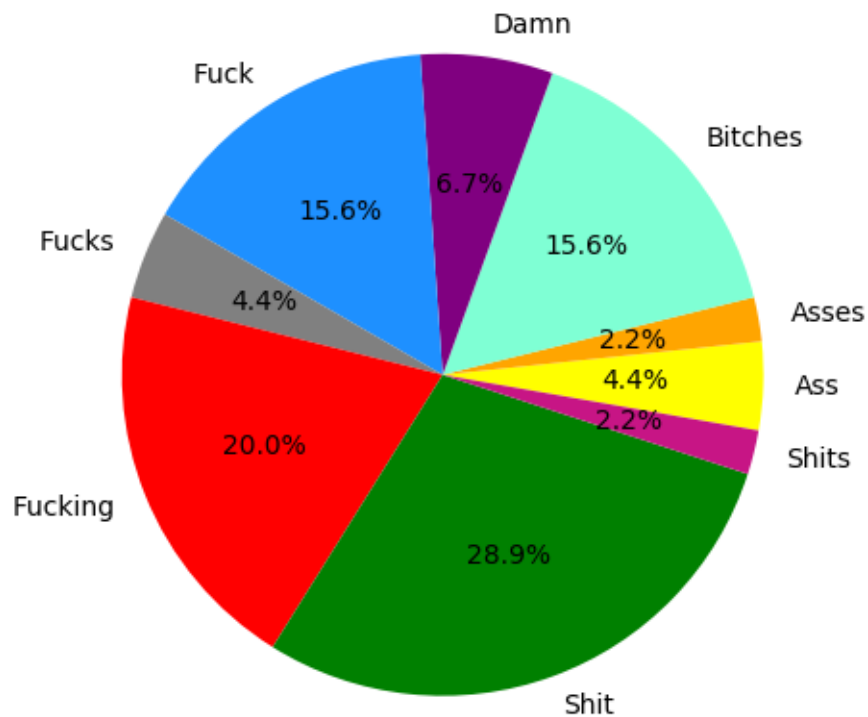
```

1 from matplotlib import pyplot
2 import numpy
3
4 #define the bar chart
5 labels = [ 'Shit', 'Fucking', 'Fuck', 'Bitches', 'Damn', 'Fucks', 'Ass', 'Shits', 'Asses' ]
6 y_pos = numpy.arange(len(labels))
7 sizes = [13, 9, 7, 7, 3, 2, 2, 1, 1]
8
9 #do the plot
10 pyplot.bar(y_pos, sizes, align='center', alpha=0.5)
11 pyplot.xticks(y_pos, labels)
12 pyplot.ylabel('Occurrence')
13 pyplot.title('Profanity in Tweets')
14 pyplot.show()

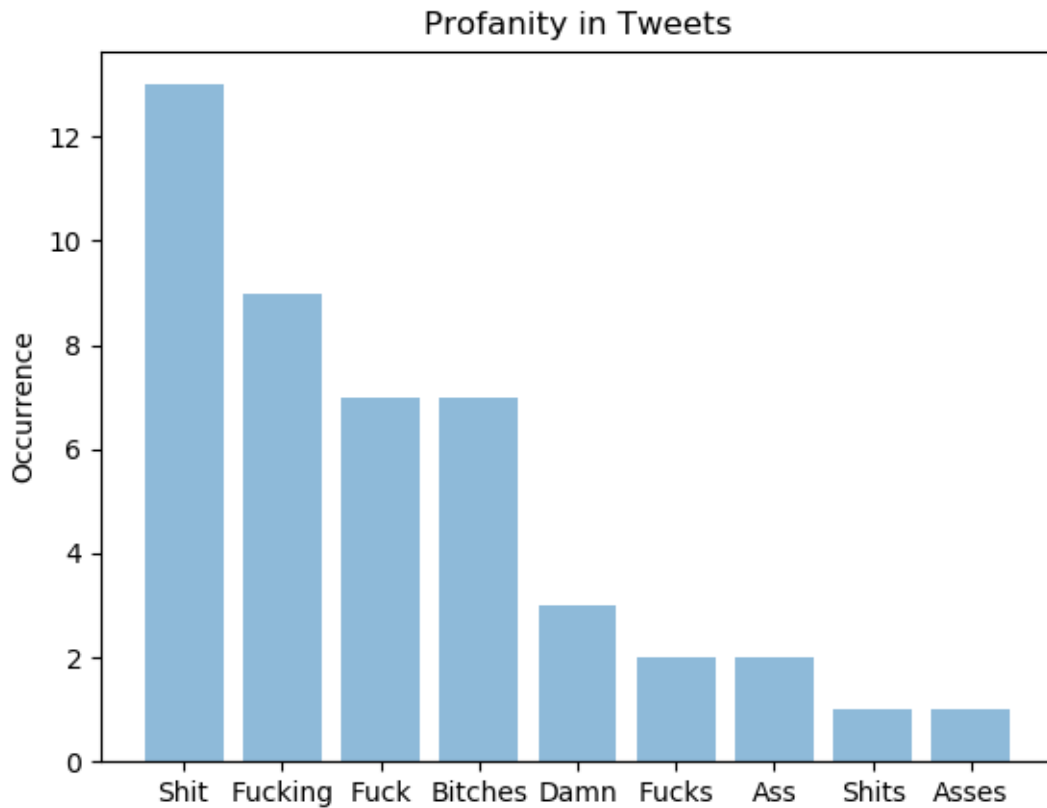
```

## The Data

Below are the recorded percentages of various profanity from the 1136 tweets in the sample size. A total of 37 tweets were found to be profane.



Shown in the next chart are the actual numbers of occurrences of the various profanity in the 37 profane tweets found from the sample.



## Conclusions

The data collected show that Twitter is much less profane than 50%. In fact, only about 3.3% of tweets collected were found to be profane. However, those that did contain profanity often had multiple occurrences of it. The question of the safety of a platform such as Twitter must ultimately be answered by the individual.

## Further Research

This small data study could be expanded into a much larger research project quite easily by collecting profanity proportions from other social media platforms, and analyzing possible correlations between variables. It would be interesting indeed to see which of the major social media platforms had the most occurrences of profanity, which had the largest total number of profane words, or how the amount of profanity of a platform was correlated to users' opinion of the platform.

## References

- [1] I don't think we need any references, do we?