Q2

November 4, 2016

0.1 Question 2

Group 19 Abhisek Mohanty Abhishek Nigam

```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        import numpy as np

        %matplotlib inline
```

```
In [2]: q2_data = pd.read_csv('datasets/events_train_holdout.tsv', delimiter='\t', error_bad_lines=False
        clean_TS = q2_data[q2_data["created_tstamp"].isnull() == False]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "NaN"]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "0"]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "1"]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "2"]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "3"]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "4"]
        clean_TS = clean_TS[clean_TS["created_tstamp"] != "5"]
```

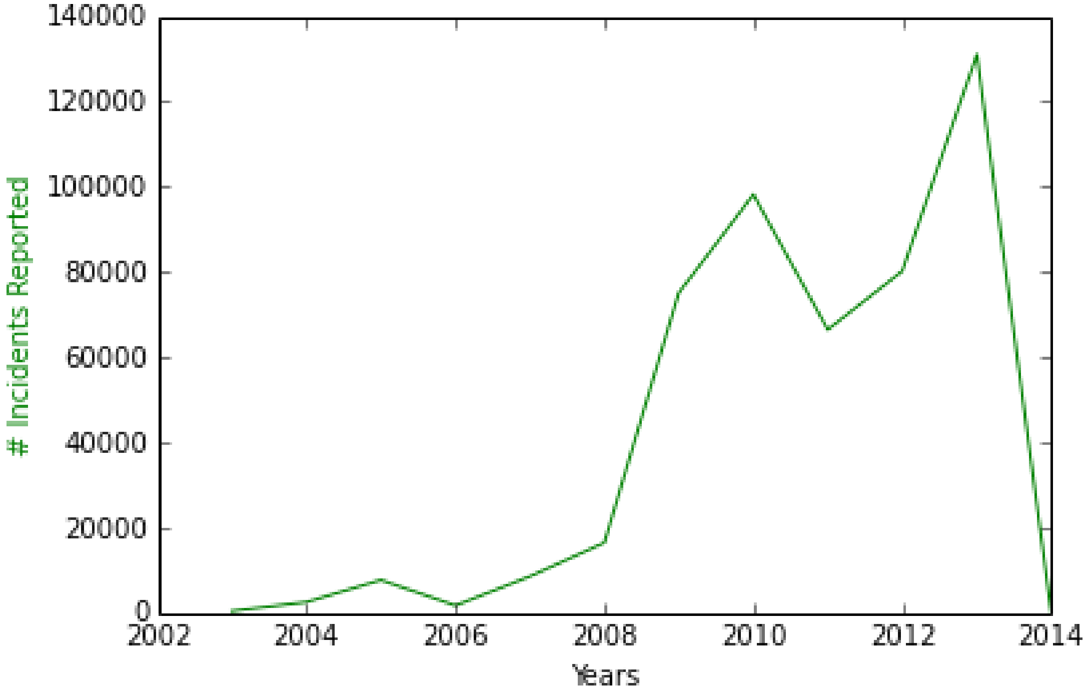
```
Skipping line 45149: expected 13 fields, saw 15
```

```
In [4]: def split_year(row):
            val = str(row['created_tstamp'])
            val1 = str(row['start_tstamp'])
            val2 = str(row['confirmed_tstamp'])
            if "-" in val:
                return val.split('T')[0].split('-')[0]
            if "-" in val1:
                return val1.split('T')[0].split('-')[0]
            if "-" in val2:
                return val2.split('T')[0].split('-')[0]


        clean_TS['new_year'] = clean_TS.apply(lambda row: split_year(row), axis=1)
        year_grouped = clean_TS.groupby('new_year')
```

```
In [5]: fig, ax1 = plt.subplots()
        x = year_grouped.size().index
        ax1.plot(x, year_grouped.size(), 'g-')
        ax1.set_xlabel('Years')
        ax1.set_ylabel('# Incidents Reported', color='g')
```

```
Out[5]: <matplotlib.text.Text at 0x7f0cd41ef150>
```


140000
120000
100000
80000
60000
40000
20000
0
# Incidents Reported
2002
2004
2006
2008
2010
2012
2014
Years

```
In []:
```