Train Wreck Analysis

Long Nguyen
SJSU ID: 010806471

1. Project abstract

Use train wreck datasets http://www.trainwreckdb.com/ with spark service in bluemix to figure out what are the 10 most dangerous places for accidents and why.

2. Project Scope

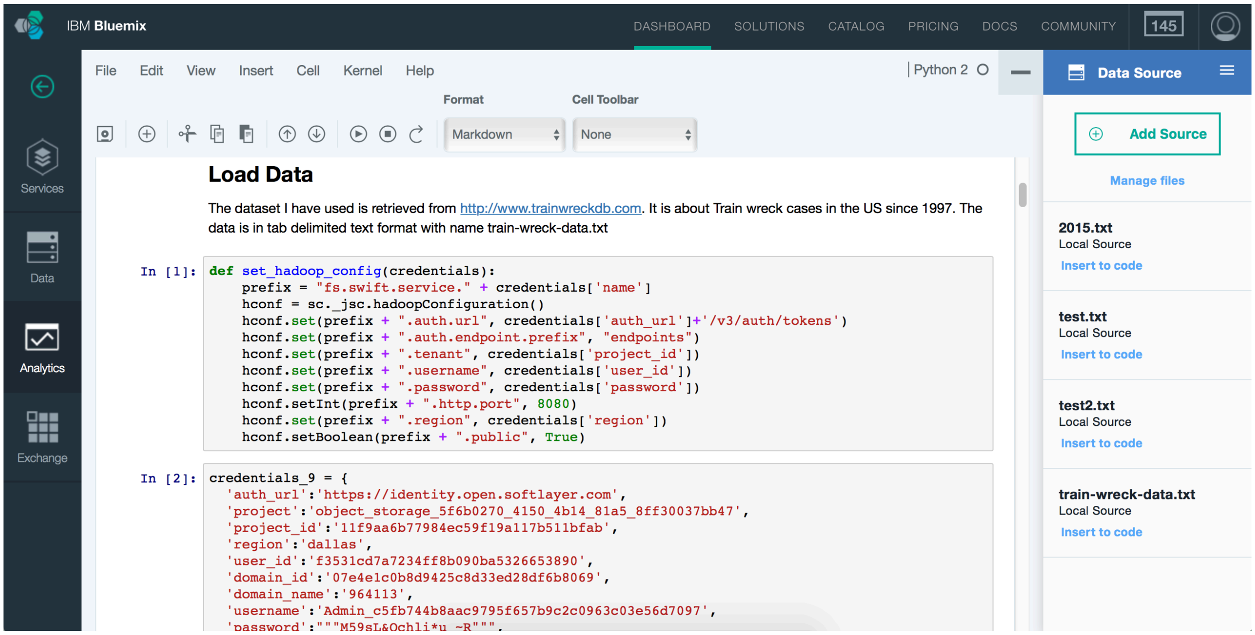
In this project, I will do analysis to point out the 10 most dangerous Cities and State for train accidents.

The dataset also contains information about the Street where accidents occurred. However, after do experimental analysis on the Street, the number of wreck cases mostly lay on the the streets with common name like Private or Private Rd (see Github link for more details). This result is not able to come to conclusion.

3. Github Link

https://github.com/longnguyen0708/TrainWreckAnalysis

4. Load Dataset

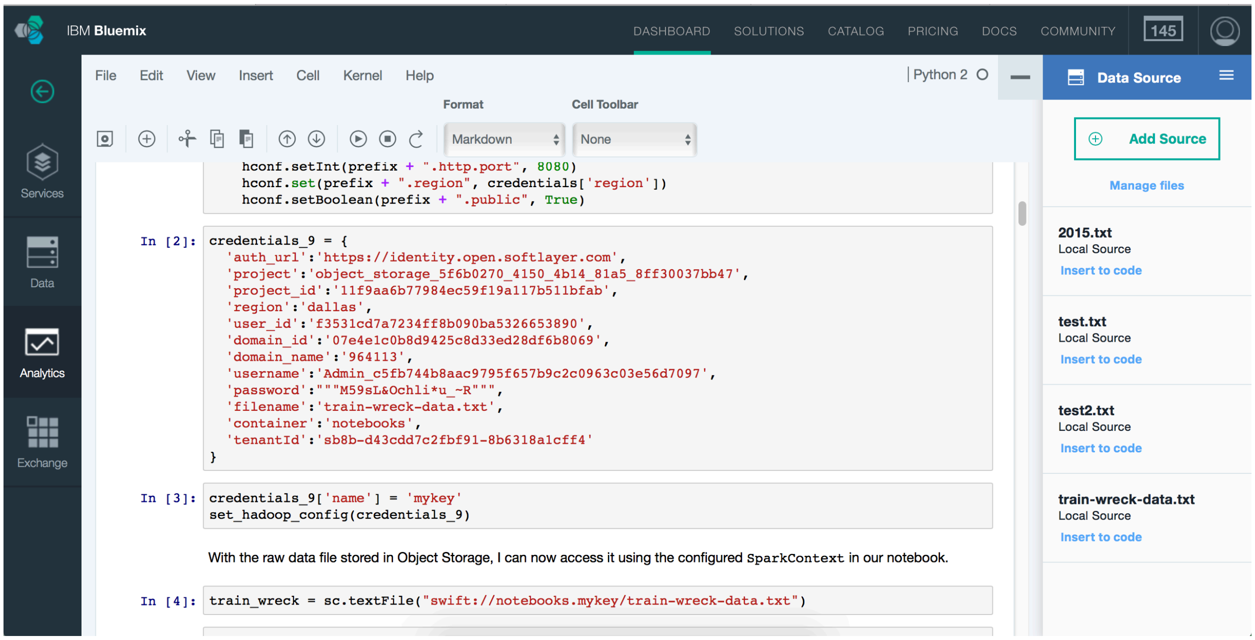


Load Data

The dataset I have used is retrieved from http://www.trainwreckdb.com. It is about Train wreck cases in the US since 1997. The data is in tab delimited text format with name train-wreck-data.txt

```
In [1]: def set_hadoop_config(credentials):
            prefix = "fs.swift.service." + credentials['name']
            hconf = sc._jsc.hadoopConfiguration()
            hconf.set(prefix + ".auth.url", credentials['auth_url']+'/v3/auth/tokens')
            hconf.set(prefix + ".auth.endpoint.prefix", "endpoints")
            hconf.set(prefix + ".tenant", credentials['project_id'])
            hconf.set(prefix + ".username", credentials['user_id'])
            hconf.set(prefix + ".password", credentials['password'])
            hconf.setInt(prefix + ".http.port", 8080)
            hconf.set(prefix + ".region", credentials['region'])
            hconf.setBoolean(prefix + ".public", True)
```

```
In [2]: credentials_9 = {
          'auth_url':'https://identity.open.softlayer.com',
          'project':'object_storage_5f6b0270_4150_4b14_81a5_8ff30037bb47',
          'project_id':'11f9aa6b77984ec59f19a117b511bfab',
          'region':'dallas',
          'user_id':'f3531cd7a7234ff8b090ba5326653890',
          'domain_id':'07e4e1c0b8d9425c8d33ed28df6b8069',
          'domain_name':'964113',
          'username':'Admin_c5fb744b8aac9795f657b9c2c0963c03e56d7097',
          'password':"""M59sL&Ochli*u ~R""",
```



```
        hconf.setInt(prefix + ".http.port", 8080)
        hconf.set(prefix + ".region", credentials['region'])
        hconf.setBoolean(prefix + ".public", True)
```

```
In [2]: credentials_9 = {
          'auth_url':'https://identity.open.softlayer.com',
          'project':'object_storage_5f6b0270_4150_4b14_81a5_8ff30037bb47',
          'project_id':'11f9aa6b77984ec59f19a117b511bfab',
          'region':'dallas',
          'user_id':'f3531cd7a7234ff8b090ba5326653890',
          'domain_id':'07e4e1c0b8d9425c8d33ed28df6b8069',
          'domain_name':'964113',
          'username':'Admin_c5fb744b8aac9795f657b9c2c0963c03e56d7097',
          'password':"""M59sL&Ochli*u_~R""",
          'filename':'train-wreck-data.txt',
          'container':'notebooks',
          'tenantId':'sb8b-d43cdd7c2fbf91-8b6318a1cff4'
        }
```

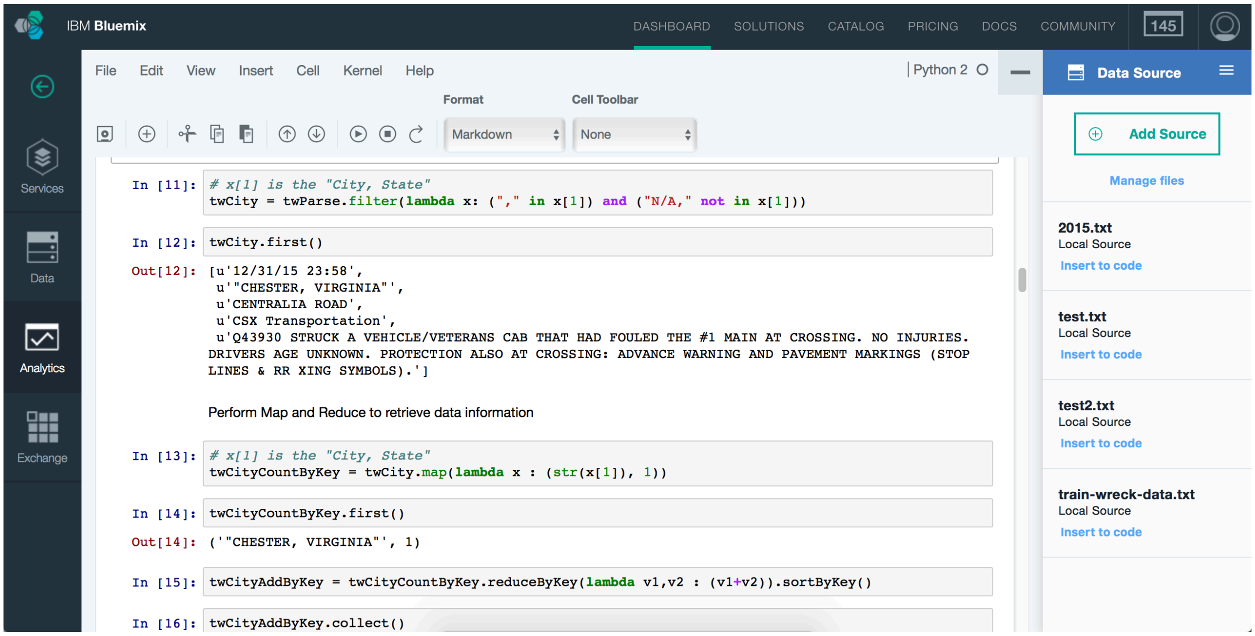
```
In [3]: credentials_9['name'] = 'mykey'
        set_hadoop_config(credentials_9)
```

With the raw data file stored in Object Storage, I can now access it using the configured `SparkContext` in our notebook.

```
In [4]: train_wreck = sc.textFile("swift://notebooks.mykey/train-wreck-data.txt")
```

5. Point out 10 most dangerous Cities for train accidents

Perform Map and Reduce on data

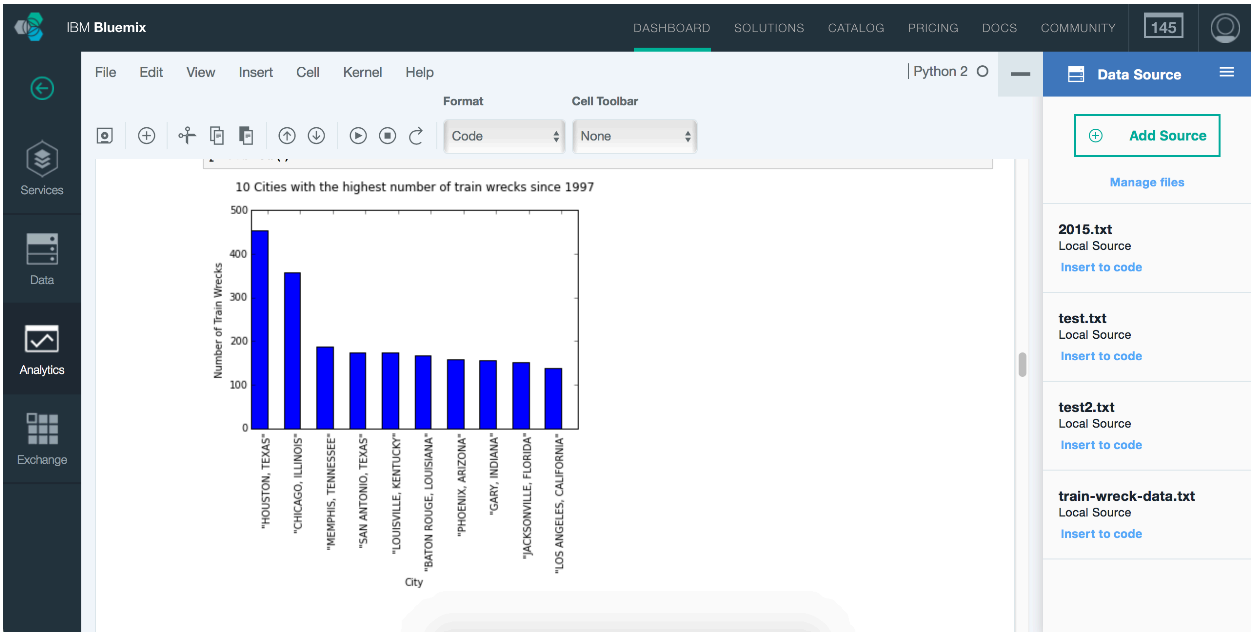


```
In [11]: # x[1] is the "City, State"
         twCity = twParse.filter(lambda x: ("," in x[1]) and ("N/A," not in x[1]))

In [12]: twCity.first()

Out[12]: [u'12/31/15 23:58',
          u'"CHESTER, VIRGINIA"',
          u'CENTRALIA ROAD',
          u'CSX Transportation',
          u'Q43930 STRUCK A VEHICLE/VETERANS CAB THAT HAD FOULED THE #1 MAIN AT CROSSING. NO INJURIES.
         DRIVERS AGE UNKNOWN. PROTECTION ALSO AT CROSSING: ADVANCE WARNING AND PAVEMENT MARKINGS (STOP
         LINES & RR XING SYMBOLS).']
```

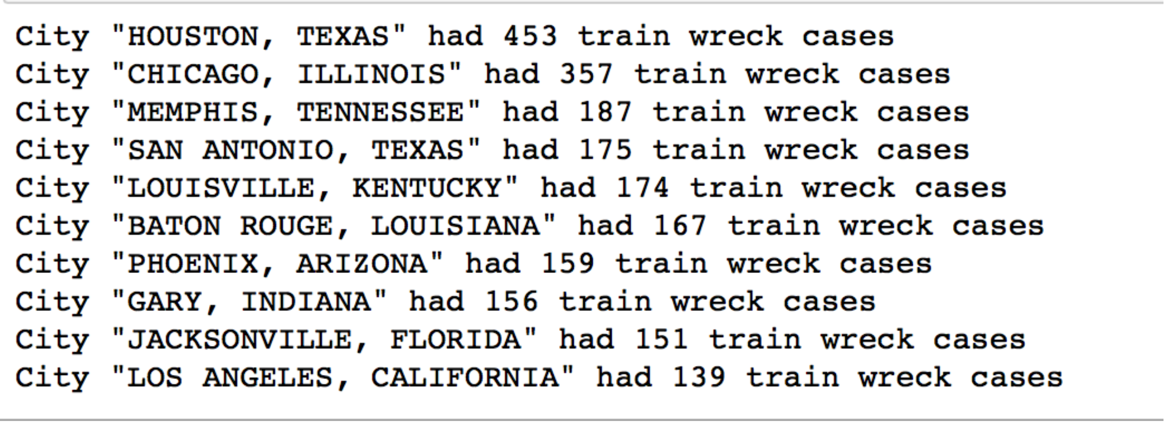
Perform Map and Reduce to retrieve data information

```
In [13]: # x[1] is the "City, State"
         twCityCountByKey = twCity.map(lambda x : (str(x[1]), 1))

In [14]: twCityCountByKey.first()

Out[14]: ('"CHESTER, VIRGINIA"', 1)

In [15]: twCityAddByKey = twCityCountByKey.reduceByKey(lambda v1,v2 : (v1+v2)).sortByKey()

In [16]: twCityAddByKey.collect()
```

Visualize the 10 Cities with the highest number of train accidents since 1997



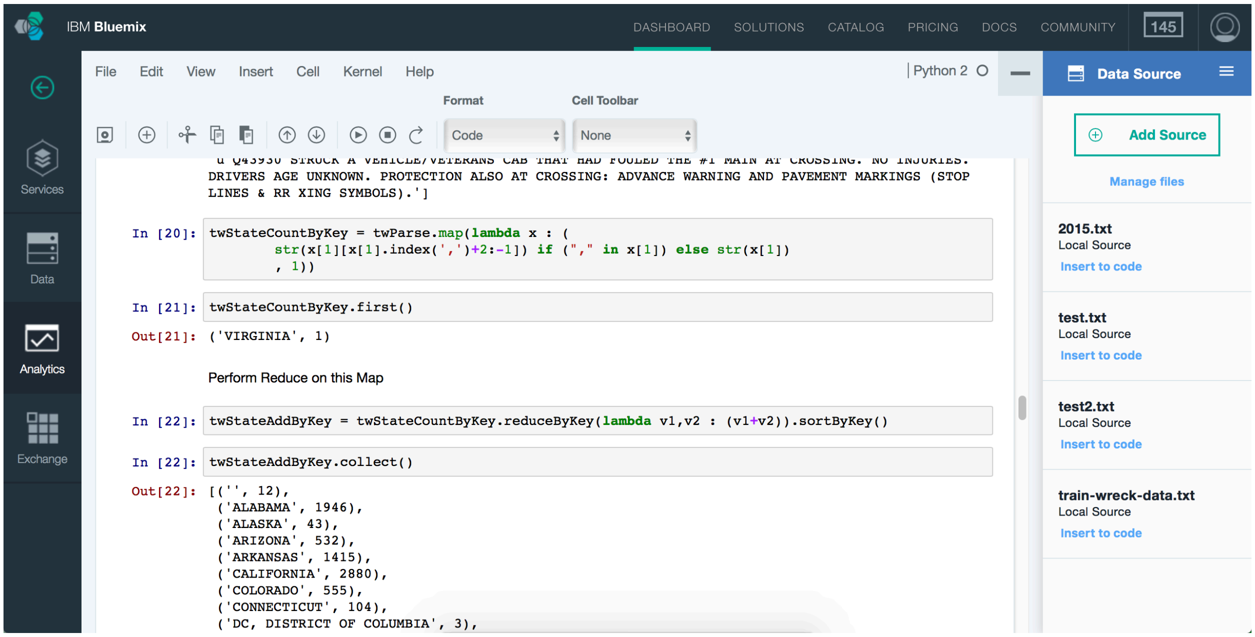
From the Graph above, we can conclude that top 10 dangerous cities for train accidents are



```
City "HOUSTON, TEXAS" had 453 train wreck cases
City "CHICAGO, ILLINOIS" had 357 train wreck cases
City "MEMPHIS, TENNESSEE" had 187 train wreck cases
City "SAN ANTONIO, TEXAS" had 175 train wreck cases
City "LOUISVILLE, KENTUCKY" had 174 train wreck cases
City "BATON ROUGE, LOUISIANA" had 167 train wreck cases
City "PHOENIX, ARIZONA" had 159 train wreck cases
City "GARY, INDIANA" had 156 train wreck cases
City "JACKSONVILLE, FLORIDA" had 151 train wreck cases
City "LOS ANGELES, CALIFORNIA" had 139 train wreck cases
```

6. Point out 10 most dangerous States for train accidents

Perform Map and Reduce on data

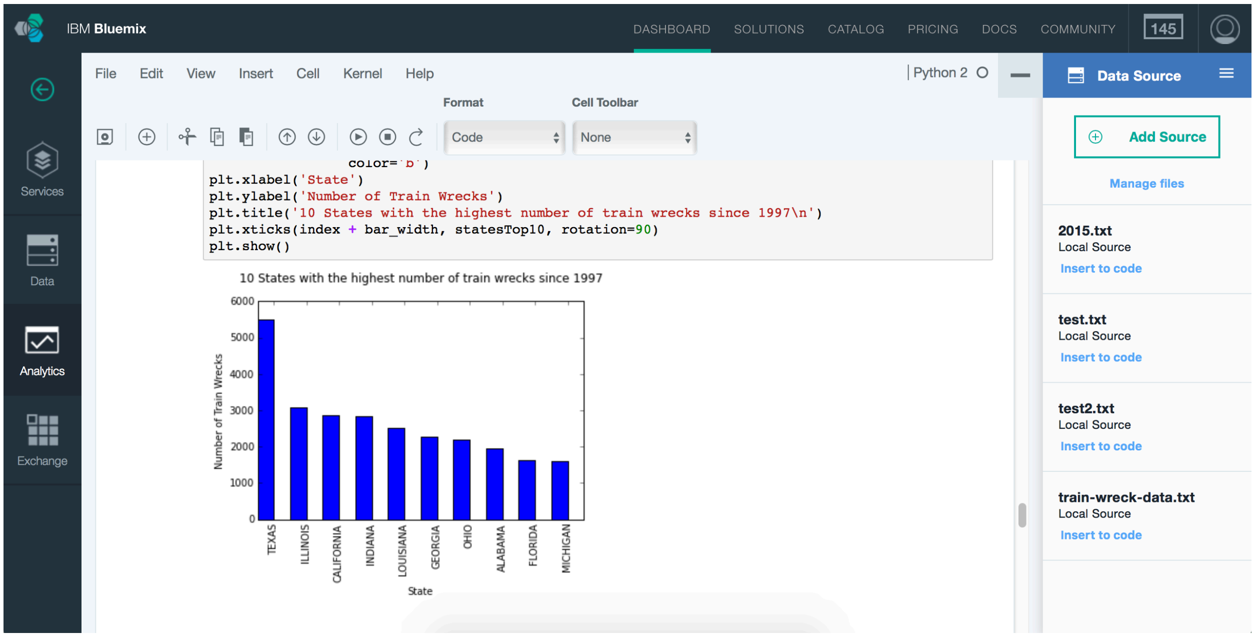


```
 u Q43930 STRUCK A VEHICLE/VETERANS CAB THAT HAD FOULED THE #1 MAIN AT CROSSING. NO INJURIES.
DRIVERS AGE UNKNOWN. PROTECTION ALSO AT CROSSING: ADVANCE WARNING AND PAVEMENT MARKINGS (STOP
LINES & RR XING SYMBOLS).']

In [20]: twStateCountByKey = twParse.map(lambda x : (
                 str(x[1][x[1].index(',')+2:-1]) if ("," in x[1]) else str(x[1])
                 , 1))

In [21]: twStateCountByKey.first()

Out[21]: ('VIRGINIA', 1)
```

Perform Reduce on this Map

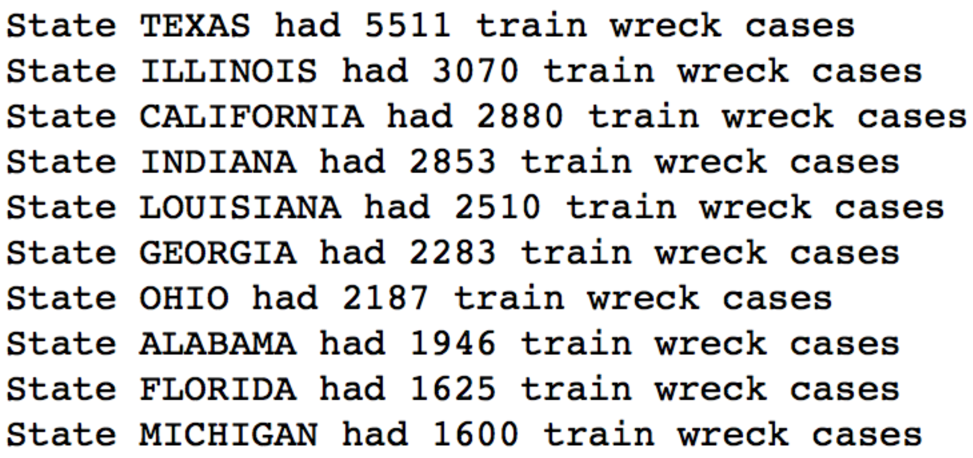
```
In [22]: twStateAddByKey = twStateCountByKey.reduceByKey(lambda v1,v2 : (v1+v2)).sortByKey()

In [22]: twStateAddByKey.collect()

Out[22]: [('', 12),
          ('ALABAMA', 1946),
          ('ALASKA', 43),
          ('ARIZONA', 532),
          ('ARKANSAS', 1415),
          ('CALIFORNIA', 2880),
          ('COLORADO', 555),
          ('CONNECTICUT', 104),
          ('DC, DISTRICT OF COLUMBIA', 3),
```

Visualize the 10 States with the highest number of train accidents since 1997



```
                  color='b')
plt.xlabel('State')
plt.ylabel('Number of Train Wrecks')
plt.title('10 States with the highest number of train wrecks since 1997\n')
plt.xticks(index + bar_width, statesTop10, rotation=90)
plt.show()
```

From the Graph above, we can conclude that top 10 dangerous cities for train accidents are



```
State TEXAS had 5511 train wreck cases
State ILLINOIS had 3070 train wreck cases
State CALIFORNIA had 2880 train wreck cases
State INDIANA had 2853 train wreck cases
State LOUISIANA had 2510 train wreck cases
State GEORGIA had 2283 train wreck cases
State OHIO had 2187 train wreck cases
State ALABAMA had 1946 train wreck cases
State FLORIDA had 1625 train wreck cases
State MICHIGAN had 1600 train wreck cases
```