CMPE 272

Student Name: Mandar Gade

Student ID: 011168300

Semester: Fall,2016

Assignment: Extra Credit Assignment

Extra Credit Assignment Report

The assignment has been implemented in IPython Jupyter notebook in IBM Bluemix. IBM Bluemix provided me with the Python 2, Spark 1.6 and various visualization services to represent the analytics results.

The assignment is regarding the analysis of top ten dangerous places for train accidents in USA. The data set I used for the assignment is taken from http://www.trainwreckdb.com.

Table of Content: 1) Data gathering 2) Data loading 3) Data analysis 4) Visualization

Data Gathering:

Data is provided by http://www.trainwreckdb.com. It is provided as year wise data from 1997. The data for this assignment is gathered from the website and converted into a csv file for further use.

Data Loading:

Loading the data into spark is easy in jupyter notebook. You can use add source option provided in the notebook. Create your Object Storage instance for the Apache Spark associated with your analytics instance in IBM Bluemix. To use the data from data source you need to use the 'insert to code' option. One of the important step is to set the Hadoop configuration for your Object Storage instance.

In [318]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from pyspark.sql import SQLContext, Row

credentials_1 = {
  'auth_url':'https://identity.open.softlayer.com',
  'project':'object_storage_c516a28b_2872_43df_b03c_69cf06d29a87',
  'project_id':'def159f2228840568e646c8538e356c5',
  'region':'dallas',
  'user_id':'73445096049c44d6bf2274030f99b35f',
  'domain_id':'a499d11b021f48868434a19bd3bef871',
  'domain_name':'1165655',
  'username':'admin_ea361f279abdc7f8ec33b331b9e2114d3834eb7b',
  'password':"""XsQ?[/q{84gKLiAc""",
  'filename':'train_wreck_2.csv',
  'container':'notebooks',
  'tenantId':'s55b-5734a6344ac010-5331a58899ab'
}

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)
```

Data Analysis:

Next step is to configure Hadoop with object storage instance. The csv data that is in our object storage, we are now connecting the object storage instance to hadoop. This allows us to use the data.

Here I have connected my object storage instance to Hadoop. I am parsing the data to create a rdd dataset that I will use to create a spark dataframe.

```
In [16]: credentials_1['name'] = 'keystone'
         set_hadoop_config(credentials_1)

         data2 = sc.textFile("swift://notebooks.keystone/train_wreck_2.csv")

In [17]: data2.count()
Out[17]: 53085

In [48]: data_Parse = data2.map(lambda line : line.split(","))

In [104]: data_Parse.first()
          data_Parse.take(5)
Out[104]: [[u'"City', u' State"', u'Street', u'Railroad', u'Description'],
           [u'OKLAHOMA', u'171ST', u'BNSF Railway Company', u''],
           [u'MINNESOTA', u'COUNTY ROAD', u'BNSF Railway Company', u''],
           [u'"PERRY', u' KANSAS"', u'FRONT', u'Union Pacific Railroad Company', u''],
           [u'"LA GRANGE',
            u' TENNESSEE"',
            u'CHESTNUT ST',
            u'Norfolk Southern Corporation',
            u'']]
```

Create a spark dataframe by paring the data_Parse rdd to get the column with the accident places. Use sqlContext to query the dataframe table and to get the top places with most accidents.

```
In [108]: dataPrepare = data_Parse.map(lambda x: Row(City=(x[0],x[1]), State=x[1], Street=x[2]))

In [109]: dataSchema = sqlContext.createDataFrame(dataPrepare)
          dataSchema.registerTempTable("list")
          sqlContext.cacheTable("list")

In [120]: DataFrame = sqlContext.sql("Select City, State, Street from list")

          DataFrame.count()
Out[120]: 53085

In [121]: place = sqlContext.sql("SELECT City FROM list GROUP BY City ORDER BY COUNT(*) DESC LIMIT 10")

In [122]: place.show()
          +--------------------+
          |                City|
          +--------------------+
          |  ["HOUSTON, TEXAS"]|
          |["CHICAGO, ILLINO...|
          |["MEMPHIS, TENNES...|
          |["LOUISVILLE, KEN...|
          |["SAN ANTONIO, TE...|
          |["BATON ROUGE, LO...|
          |["PHOENIX, ARIZONA"]|
          |   ["GARY, INDIANA"]|
          |["JACKSONVILLE, F...|
          |["LOS ANGELES, CA...|
          +--------------------+
```

Visualization:

I used google map and amcharts map to visualize the results using HTML.

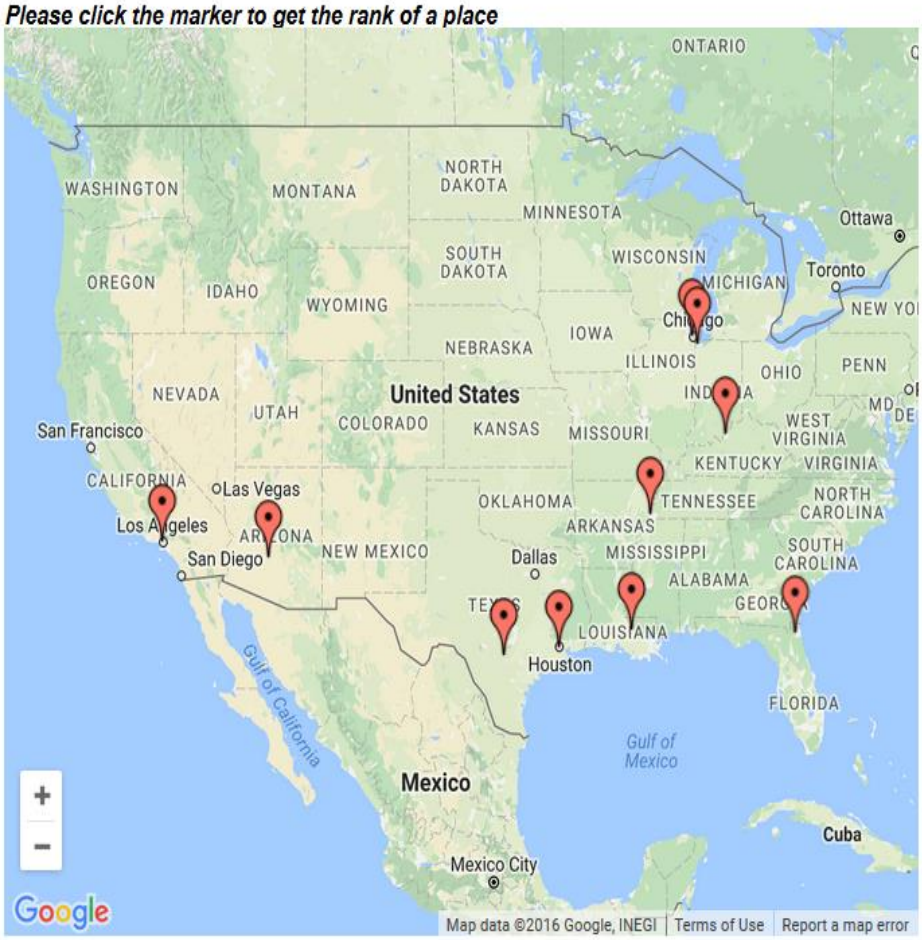
In [316]:
```
from IPython.core.display import HTML

def putHTML():
    source = """
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <title>Google Maps Multiple Markers</title>
  <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA_x7AiIUf6PazB_sieNg_DjSrq69Z7
nhM"></script>
</head>
<body>
  <h5>Top 10 most dangerous places in USA for train accidents</h5>
  <h6>Please click the marker to get the rank of a place</h6>
  <div id="map" style="width: 600px; height: 500px;"></div>

  <script>
    // Define your locations: HTML content for the info window, latitude, longitude
    var locations = [
      ['<h4>1-HOUSTON TEXAS</h4>', 29.7604, -95.3698],
      ['<h4>2-CHICAGO ILLINOIS</h4>', 41.8781, -87.6298],
      ['<h4>3-MEMPHIS TENNESSEE</h4>', 35.1495, -90.0490],
      ['<h4>4-LOUISVILLE KENTUCKY</h4>', 38.2527, -85.7585],
      ['<h4>5-SAN ANTONIO TEXAS</h4>', 29.4241, -98.4936],
      ['<h4>6-BATON ROUGE LOUISIANA</h4>', 30.4583, -91.1403],
      ['<h4>7-PHOENIX ARIZONA</h4>', 33.4484, -112.0740],
      ['<h4>8-GARY INDIANA</h4>', 41.5934, -87.3464],
      ['<h4>9-JACKSONVILLE FLORIDA</h4>', 30.3322, -81.6557],
      ['<h4>10-LOS ANGELES CALIFORNIA</h4>', 34.0522, -118.2437]
    ];
```

```
    // Setup the different icons and shadows
    var iconURLPrefix = 'http://maps.google.com/mapfiles/ms/icons/';

    var icons = [
      iconURLPrefix + 'red-dot.png'
    ]
    var iconsLength = icons.length;

    var map = new google.maps.Map(document.getElementById('map'), {
      zoom: 10,
      center: new google.maps.LatLng(-37.92, 151.25),
      mapTypeId: google.maps.MapTypeId.ROADMAP,
      mapTypeControl: false,
      streetViewControl: false,
      panControl: false,
      zoomControlOptions: {
         position: google.maps.ControlPosition.LEFT_BOTTOM
      }
    });

    var infowindow = new google.maps.InfoWindow({
      maxWidth: 160
    });

    var markers = new Array();

    var iconCounter = 0;

    // Add the markers and infowindows to the map
    for (var i = 0; i < locations.length; i++) {
      var marker = new google.maps.Marker({
        position: new google.maps.LatLng(locations[i][1], locations[i][2]),
        map: map,
        icon: icons[iconCounter]
      });

      markers.push(marker);

      google.maps.event.addListener(marker, 'click', (function(marker, i) {
        return function() {
          infowindow.setContent(locations[i][0]);
          infowindow.open(map, marker);
        }
```

```
      iconCounter++;
      // We only have a limited number of possible icon colors, so we may have to restart the counter
      if(iconCounter >= iconsLength) {
        iconCounter = 0;
      }
    }

    function autoCenter() {
      //  Create a new viewpoint bound
      var bounds = new google.maps.LatLngBounds();
      //  Go through each...
      for (var i = 0; i < markers.length; i++) {
                                bounds.extend(markers[i].position);
      }
      //  Fit these bounds to the map
      map.fitBounds(bounds);
    }
    autoCenter();
  </script>
</body>
    """
    return HTML(source)


putHTML()
```

Top 10 most dangerous places in USA for train accidents

Please click the marker to get the rank of a place



Visualization of top 10 states with most train accidents:

In [312]:
```
place_3 = sqlContext.sql("SELECT State FROM list GROUP BY State ORDER BY COUNT(*) DESC LIMIT 11")
place_3.filter(place_3['State'] !='PRIVATE').show()
```

```
+------------+
|       State|
+------------+
|      TEXAS"|
|   ILLINOIS"|
|    INDIANA"|
| CALIFORNIA"|
|  LOUISIANA"|
|    GEORGIA"|
|       OHIO"|
|    ALABAMA"|
|    FLORIDA"|
|   MICHIGAN"|
+------------+
```

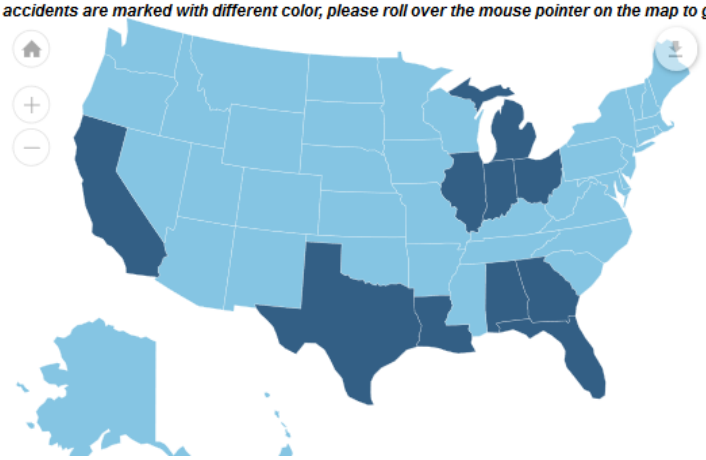
In [300]:
```
from IPython.core.display import HTML

def putHTML():
    source = """
<head>
    <title></title>
    <script src="https://www.amcharts.com/lib/3/ammap.js"></script>
    <script src="https://www.amcharts.com/lib/3/maps/js/usaLow.js"></script>
    <script src="https://www.amcharts.com/lib/3/plugins/export/export.min.js"></script>
    <link rel="stylesheet" href="https://www.amcharts.com/lib/3/plugins/export/export.css" type="text/css" media="all"
 />
    <script src="https://www.amcharts.com/lib/3/themes/light.js"></script>

</head>
```

```
   <script>
var map = AmCharts.makeChart( "chartdiv", {
  "type": "map",
  "theme": "light",
  "colorSteps": 10,

  "dataProvider": {
    "map": "usaLow",
    "areas": [ {
      "id": "US-AL",
      "value": 20851820
    }, {
      "id": "US-AK",
      "value": 0
    }, {
      "id": "US-AZ",
      "value": 0
    }, {
      "id": "US-AR",
      "value": 0
    }, {
      "id": "US-CA",
      "value": 20851820
    }, {
      "id": "US-CO",
      "value": 0
    }, {
      "id": "US-CT",
      "value": 0
    }, {
      "id": "US-DE",
      "value": 0
    }, {
      "id": "US-FL",
      "value": 20851820
    }, {
      "id": "US-GA",
      "value": 20851820
    }, {
      "id": "US-HI",
      "value": 0
    }, {
      "id": "US-ID",
      "value": 0
    }, {
      "id": "US-IL",
      "value": 20851820
    }, {
      "id": "US-IN",
      "value": 20851820
    }, {
      "id": "US-IA",
      "value": 0
    }, {
      "id": "US-KS",
      "value": 0
    }, {
      "id": "US-KY",
```

```
      "value": 0
    }, {
      "id": "US-WI",
      "value": 0
    }, {
      "id": "US-WY",
      "value": 0
    } ]
  },

  "areasSettings": {
    "autoZoom": true
  },



  "export": {
    "enabled": true
  }

} );
    </script>
<h5>Top 10 states with most train accidents are marked with different color, please roll over the mouse pointer on the map t
o get the state name
<div id="chartdiv" style="min-width: 310px; max-width: 600px; height: 400px; margin: 0 auto"></div>

</body>
    """
    return HTML(source)


putHTML()
```

Out[300]:

Top 10 states with most train accidents are marked with different color, please roll over the mouse pointer on the map to get the state name



Jupyter Notebook Link: Please use the link provided below to access the Jupyter notebook.

https://cdsx.ng.bluemix.net/data/notebooks/1e26e28a-e1de-479d-890f-57902b07d7fa/view?access_token=d91858ac0695eea2d1f07296b5d761068e661af1eabdcff899f7bcee95bb60dd

Short URL: https://goo.gl/OmOSiG

GitHub Link: https://github.com/MandarGade/272-Extra_Credit_Assignment