Crime rate analysis across United States

Data Science question: 1. Which state is currently having highest crime rate and which is safe to live in with lowest crime rate ? 2. Predicting which state will have highest and lowest crime rate in nearby future ?

Problem: Scrap web site for acquiring the data and storing the data in MongoDB.

Solution:

```
from lxml import html
import requests
import re
import json
from pymongo import MongoClient
import time
connection = MongoClient()
db=connection.crime
collection=db.crimestats
states=['ca','ky','ma','md','me','mi','mn','oh','or','pa','ri','sc','sd','tn','tx','ut','va','vt','dc','wa','wi','wv','wy','nv','ak','
for s in range(len(states)):
    print states[s]
    link1="http://www.disastercenter.com/crime/"+states[s]+"crime.htm"
    startTime = time.time()
    page = requests.get("http://www.disastercenter.com/crime/uscrime.htm")

    tree = html.fromstring(page.text)


    tables = [tree.xpath('//table/tbody/tr[2]/td/center/center/font/table/tbody')]

    tabs = []

    for table in tables:
        tab = []
        for row in table:
            for col in row:
                var = col.text_content()
                var = var.strip().replace(" ", "")
                var = var.split('\n')
                if re.match('^\d{4}$', var[0].strip()):
                    db.crimestats.insert({"State":states[s],
                    "Year":re.sub("[^0-9]", "",var[0].strip()),
                    "Population": re.sub("[^0-9]", "",var[1].strip()),
                    "Total":re.sub("[^0-9]", "", var[2].strip()),
                    "Violent":re.sub("[^0-9]", "",var[3].strip()),
                    "Property": re.sub("[^0-9]", "",var[4].strip()),
                    "Murder":re.sub("[^0-9]", "", var[5].strip()),
                    "Forcible_Rape": re.sub("[^0-9]", "",var[6].strip()),
                    "Robbery":re.sub("[^0-9]", "",var[7].strip()),
                    "Aggravated_Assault": re.sub("[^0-9]", "",var[8].strip()),
                    "Burglary": re.sub("[^0-9]", "",var[9].strip()),
                    "Larceny_Theft": re.sub("[^0-9]", "",var[10].strip()),
                    "Vehicle_Theft": re.sub("[^0-9]", "",var[11].strip())})
print "DATA DUMP1 SUCCESS"
```

Problem: Retrieving the data from the database and analysing the data using pymongo , statistics library. Analysing the data.

Solution:

```
import csv
import pymongo
import statistics
import operator
import json
from pprint import pprint
from pymongo import MongoClient
connection = MongoClient()
db=connection.crime
collection=db.crimestats

item=collection.find()
violent=[]
prop=[]
total=[]
pop=[]
Percent={}
data1={}

[ak,al,ar,az,co,ct,de,fl,ga,hi,ia,,id,il,in,kn,la,mo,ms,mt,nc,nd,ne,nh,nj,nm,ny]
print db.crimestats.find()
states=['ca','ky','ma','md','me','mi','mn','oh','or','pa','ri','sc','sd','tn','tx','ut','va','vt','dc','wa','wi','wv','wy','nv','ak','
with open('data1.csv', 'w') as csvfile:
    fieldnames = ['State','Violent', 'Property','Total','Population']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
```

```
    writer.writeheader()
    for s in range(len(states)):
        j=states.index(states[s])
        for doc in collection.find({"State":states[s]}):
            print doc['Murder']
            violent1=str(doc['Violent']).strip()
            prop1=str(doc['Property']).strip()
            pop1=str(doc['Population']).strip()
            total1=str(doc['Total']).strip()
            if(violent1!=''):
             violent2=int(violent1)
             violent.append(violent2)
            if(prop1!=''):
             prop2=int(prop1)
             prop.append(prop2)
            if(pop1!=''):
             pop2=int(pop1)
             pop.append(pop2)
            if(total1!=''):
             total2=int(total1)
             total.append(total2)
        print "RES=",statistics.mean(res)
        percent=statistics.mean(total)/statistics.mean(pop)*100
        data1[j]={"State":states[s],"Violence":statistics.mean(violent),"Property":statistics.mean(prop),"Percent":percent}
        Percentage=(statistics.mean(total)/statistics.mean(pop))*100
        Percent[states[s]]=Percentage
        print states[s],"Population MEAN=",statistics.mean(pop)
        print states[s],"Total Crime MEAN=",statistics.mean(total)
        print states[s],"violent MEAN=",statistics.mean(violent)
        print states[s],"violent MEDIAN=",statistics.median(violent)
        print states[s],"violent Standard Deviation=",statistics.stdev(violent)
        print states[s],"Property MEAN=",statistics.mean(prop)
        print states[s],"Property  MEDIAN=",statistics.median(prop)
        print states[s],"Property  Standard Deviation=",statistics.stdev(prop)
        print states[s],"Population MEAN=",statistics.mean(violent)
        print "***-----------------------------------***"
        print states[s],"PERCENTAGE=",Percentage
        print "***-----------------------------------***"
        print "========================="
        writer.writerow({'State':states[s],'Violent': statistics.mean(violent), 'Property': statistics.mean(prop), 'Total': statistics
        percent=statistics.mean(total)/statistics.mean(pop)*100
        data1[j]={"State":states[s],"Violence":statistics.mean(violent),"Property":statistics.mean(prop),"Percent":percent}
print Percent
print "sorted=",sorted(Percent.iteritems(),key=operator.itemgetter(1))
gg=sorted(Percent.iteritems(),key=operator.itemgetter(1))
it = iter(sorted(Percent.iteritems()))
print it.next()
gg=sorted(Percent.values())
j=1
for k in range(len(gg)):

    print "****========================*****"
    print gg[k]
print "state with highest crimerate(Not SAFE)==>",max(Percent.iteritems(), key=operator.itemgetter(1))[0],"=" ,max(Percent.iteritems()
print "state with lowest crimerate(SAFE)==>",min(Percent.iteritems(), key=operator.itemgetter(1))[0],"=" ,min(Percent.iteritems(), key
with open('data.json', 'w') as outfile:
    data2="data=",data1
    print data1
    json.dump(data1, outfile)
    print "JSON CREATED"
```