CS 5665 Homework #2
Karun Joseph, A02240287

All code, plots reside here:
https://github.com/karunmj/usu-coursework/tree/master/cs5660datasc/hw/hw2

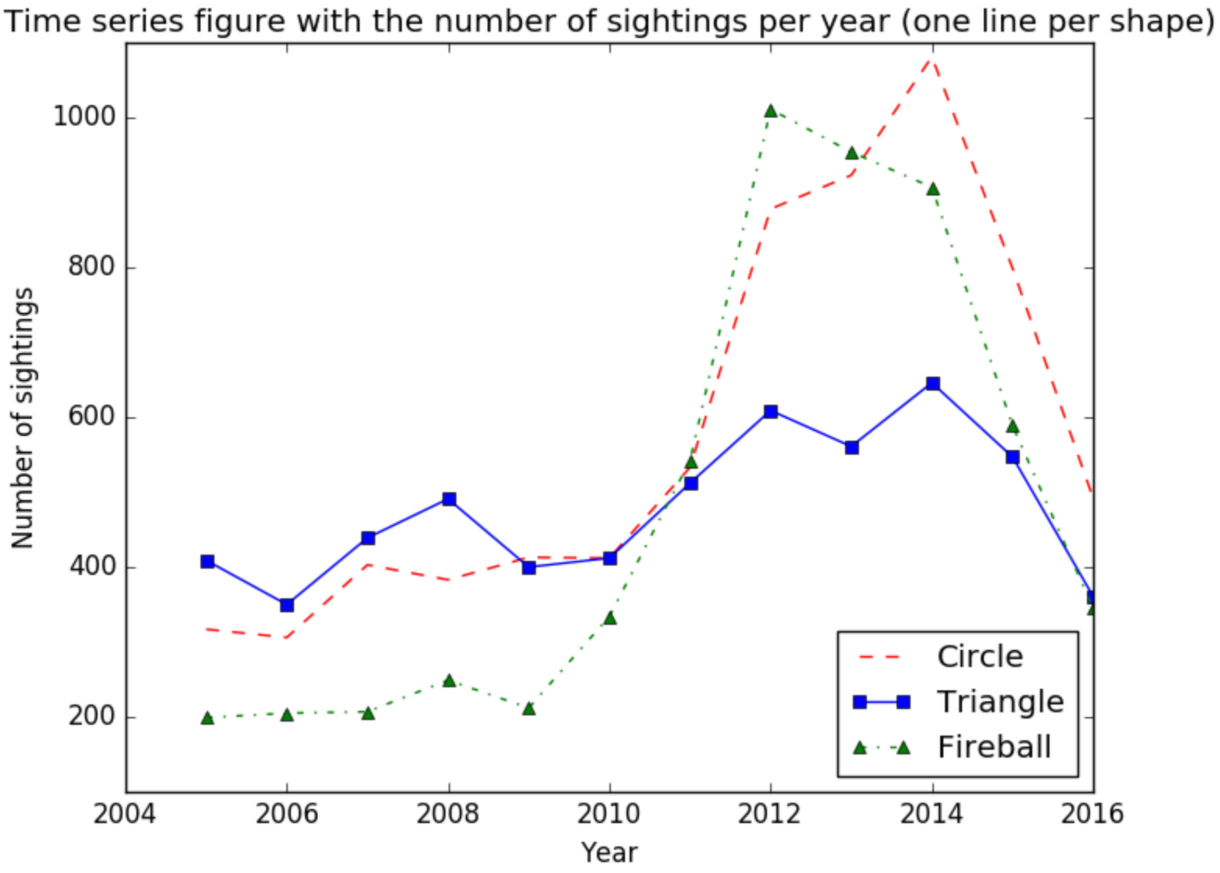
Data collection involved using request and beautiful soup libraries to get UFO data for circles, triangles and fireball shape sightings in the form of a Pandas data frame object in Python. The column labels include date of sighting, time of sighting, city, state, shape, duration, summary and posted date.

Data preprocessing involved including sighting between 1/1/2005 and 9/22/2016, representing duration of sighting in seconds and sightings in either of the 50 (+1 by including DC) states of US.

1. Box plot of duration of UFO sightings of each shape

(not done)

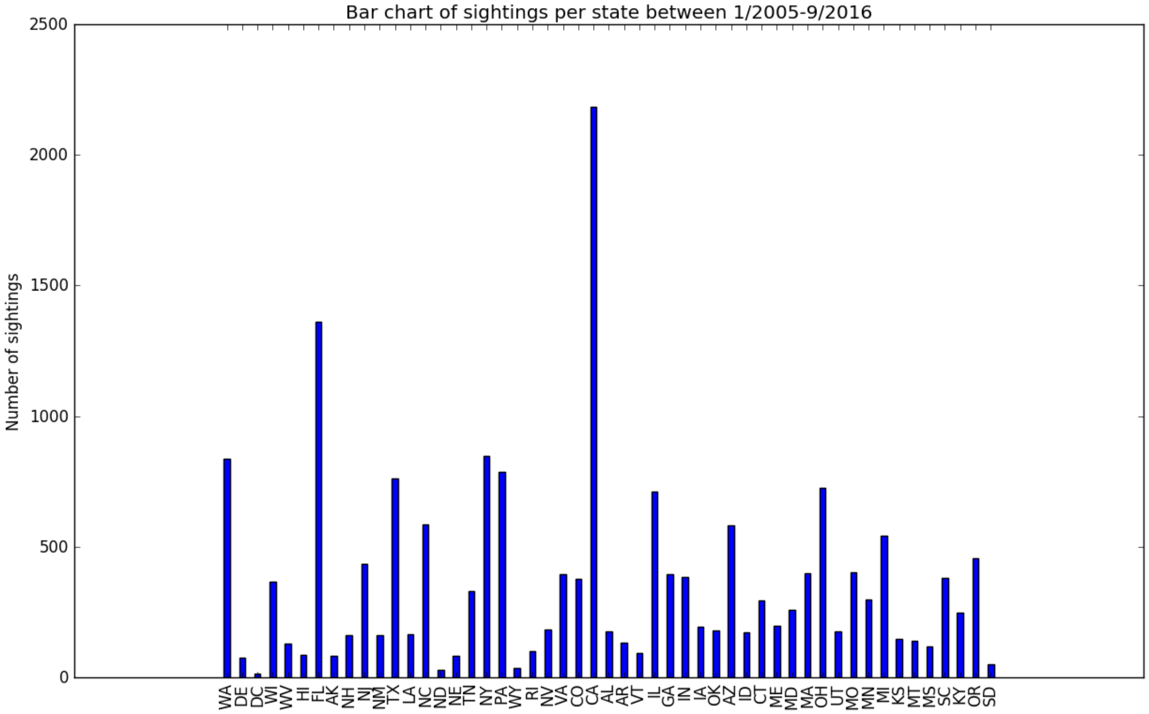
2. Time series figure with the number of sightings per year (one line per shape)



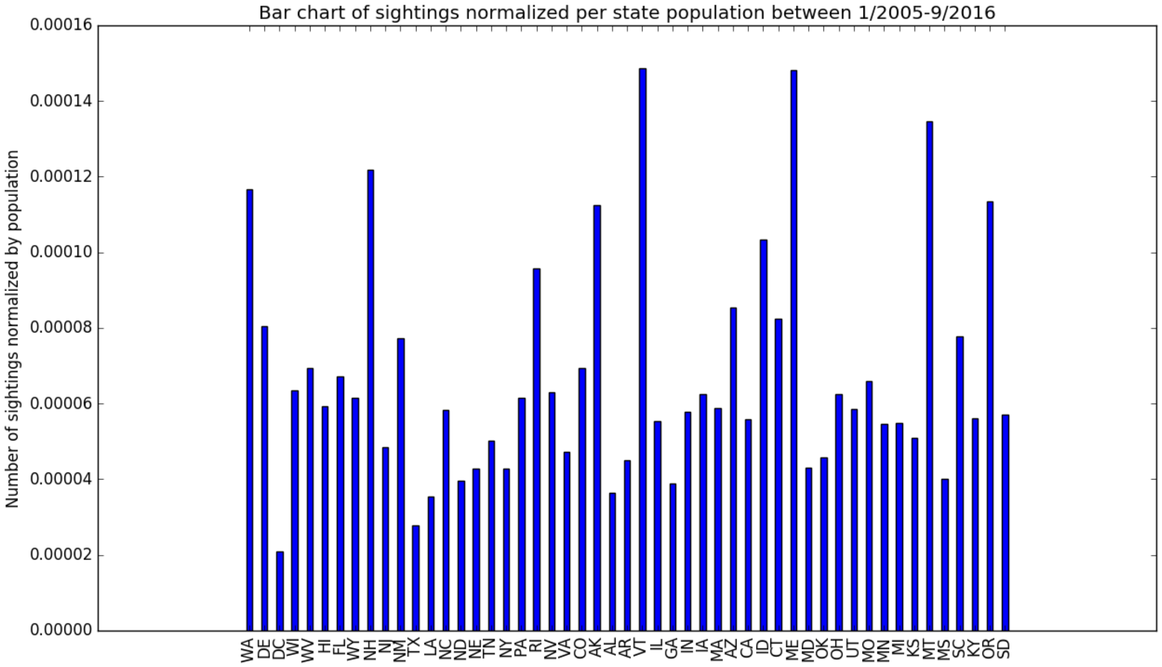
There seems to be higher number of sightings during 2012 to 2014 than the rest of years.

CS 5665 Homework #2
Karun Joseph, A02240287

3. Bar chart for sightings of:
   a. Absolute numbers by state



   b. Normalized numbers, based on state population, by state

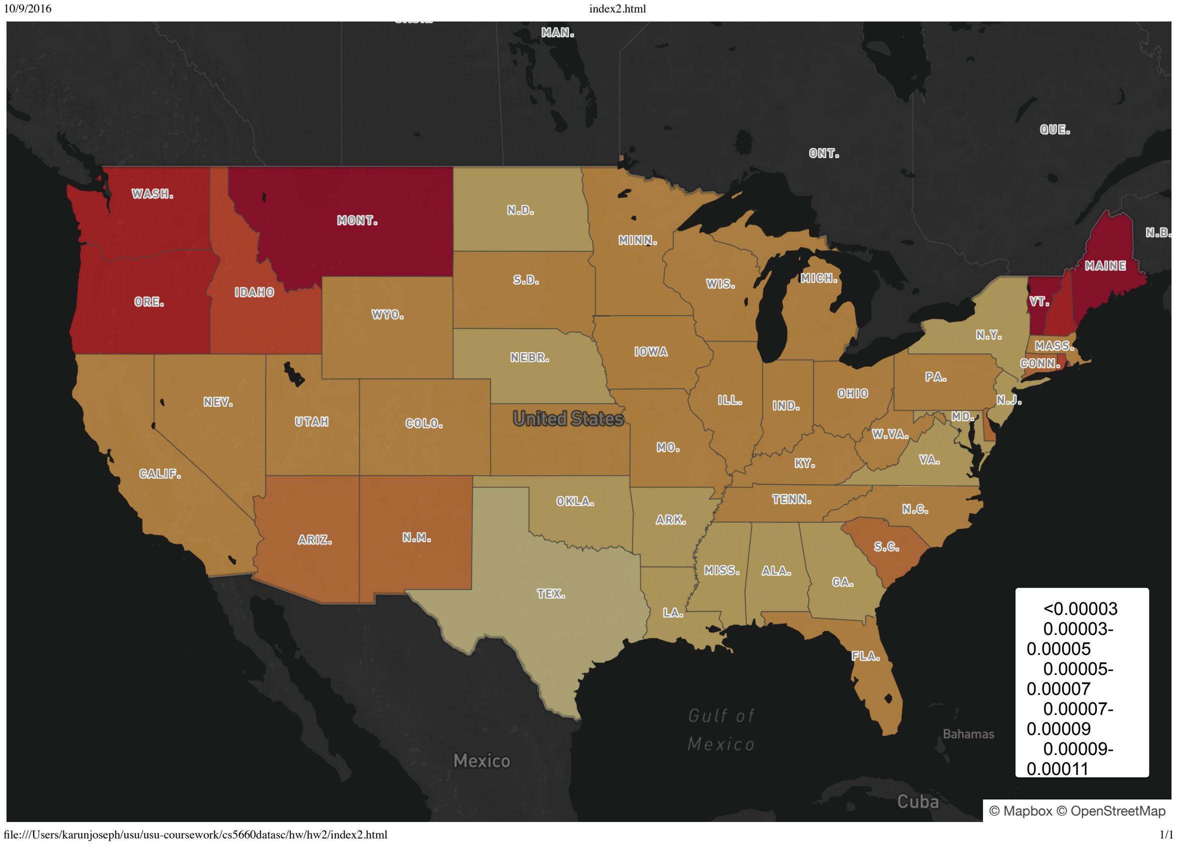


CS 5665 Homework #2
Karun Joseph, A02240287

State population numbers were obtained from US Census API service for

4. Normalized sightings by state population visualized on a map

Mapbox Studio had been used to generate a static HTML page with the normalized sightings.


10/9/2016 index2.html

file:///Users/karunjoseph/usu/usu-coursework/cs5660datasc/hw/hw2/index2.html 1/1

It looks like states like Montana, Vermont and Maine had the highest number of sightings. These states also have very low population, hence higher sightings per person. Also, these states have fewer cities that result in lower light pollution levels. Residents are more likely to see moving planetary objects like comets. meteors which could be mistaken for UFOs.

For classification, Python's sklearn tree package had been used to build the decision tree classifier. Since this package doesn't deal with categorical type of attribute directly [which is in our case], they had to be vectorized. For example, given an observation's attributes in training data set

time_of_day	region
night	s

it is converted to:

region=mw	region=ne	region=s	region=w	time_of_day=afternoon	time_of_day=evening	time_of_day=morning	time_of_day=night
0	0	1	0	0	0	0	1

Similarly, the class labels ‘Circle’, Fireball’ and ‘Triangle’ were mapped to 0, 1 and 2 respectively. Gini criterion had been used to build the classifier.

5. Classification accuracy of decision tree using test set data

Python’s sklearn metrics’ precision score had been used to calculate the ratio $\frac{t_p}{t_p + f_p}$, where t_p is the number of true positives and f_p the number of false positives, for each class

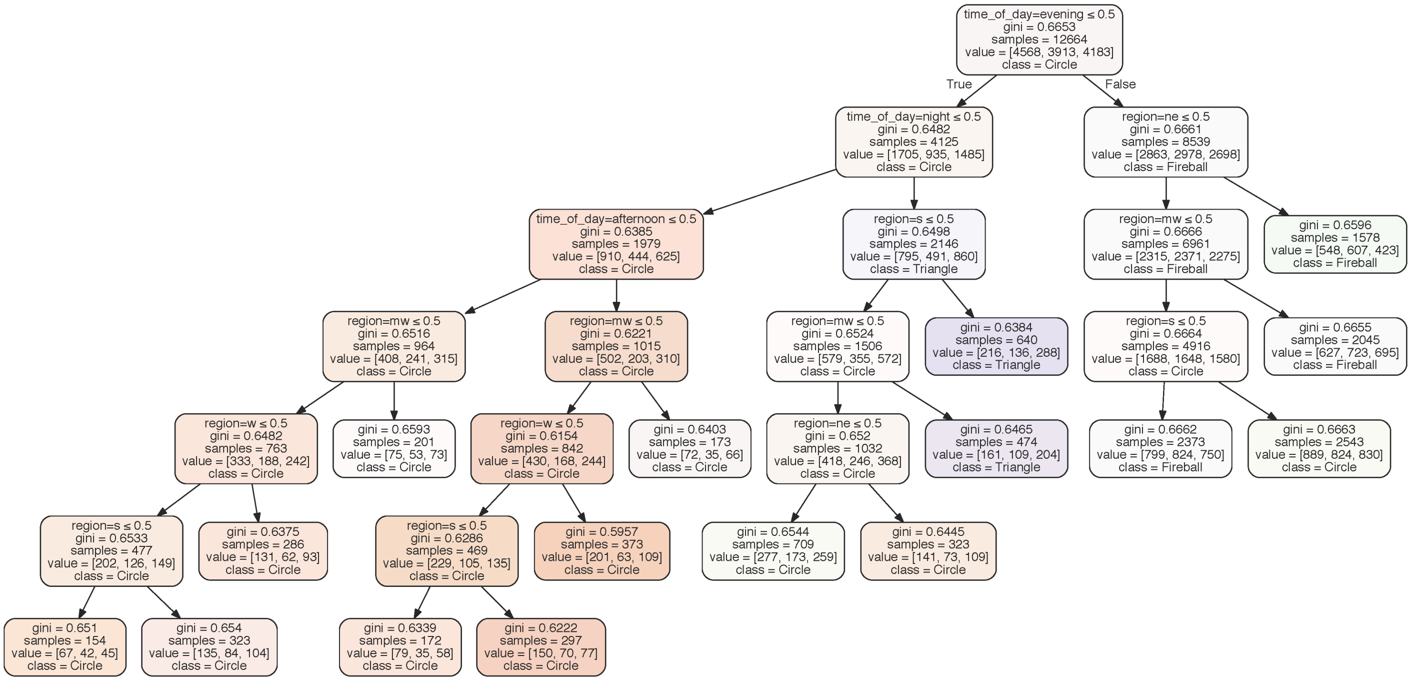
Circle	Fireball	Triangle
0.43863816	0.35789094	0.33474576

The confusion matrix of predicted vs true class labels of test set data were as follows

True label	Circle	1108	1067	197
	Fireball	733	991	117
	Triangle	685	711	158
		Circle	Fireball	Triangle
		Predicted label		

Overall, the decision tree classifier had done a poor job.

6. Illustration of built decision tree


time_of_day=evening ≤ 0.5
gini = 0.6653
samples = 12664
value = [4568, 3913, 4183]
class = Circle
True
False
time_of_day=night ≤ 0.5
gini = 0.6482
samples = 4125
value = [1705, 935, 1485]
class = Circle
region=ne ≤ 0.5
gini = 0.6661
samples = 8539
value = [2863, 2978, 2698]
class = Fireball
time_of_day=afternoon ≤ 0.5
gini = 0.6385
samples = 1979
value = [910, 444, 625]
class = Circle
region=s ≤ 0.5
gini = 0.6498
samples = 2146
value = [795, 491, 860]
class = Triangle
region=mw ≤ 0.5
gini = 0.6666
samples = 6961
value = [2315, 2371, 2275]
class = Fireball
gini = 0.6596
samples = 1578
value = [548, 607, 423]
class = Fireball
region=mw ≤ 0.5
gini = 0.6516
samples = 964
value = [408, 241, 315]
class = Circle
region=mw ≤ 0.5
gini = 0.6221
samples = 1015
value = [502, 203, 310]
class = Circle
region=mw ≤ 0.5
gini = 0.6524
samples = 1506
value = [579, 355, 572]
class = Circle
gini = 0.6384
samples = 640
value = [216, 136, 288]
class = Triangle
region=s ≤ 0.5
gini = 0.6664
samples = 4916
value = [1688, 1648, 1580]
class = Circle
gini = 0.6655
samples = 2045
value = [627, 723, 695]
class = Fireball
region=w ≤ 0.5
gini = 0.6482
samples = 763
value = [333, 188, 242]
class = Circle
gini = 0.6593
samples = 201
value = [75, 53, 73]
class = Circle
region=w ≤ 0.5
gini = 0.6154
samples = 842
value = [430, 168, 244]
class = Circle
gini = 0.6403
samples = 173
value = [72, 35, 66]
class = Circle
region=ne ≤ 0.5
gini = 0.652
samples = 1032
value = [418, 246, 368]
class = Circle
gini = 0.6465
samples = 474
value = [161, 109, 204]
class = Triangle
gini = 0.6662
samples = 2373
value = [799, 824, 750]
class = Fireball
gini = 0.6663
samples = 2543
value = [889, 824, 830]
class = Circle
region=s ≤ 0.5
gini = 0.6533
samples = 477
value = [202, 126, 149]
class = Circle
gini = 0.6375
samples = 286
value = [131, 62, 93]
class = Circle
region=s ≤ 0.5
gini = 0.6286
samples = 469
value = [229, 105, 135]
class = Circle
gini = 0.5957
samples = 373
value = [201, 63, 109]
class = Circle
gini = 0.6544
samples = 709
value = [277, 173, 259]
class = Circle
gini = 0.6445
samples = 323
value = [141, 73, 109]
class = Circle
gini = 0.651
samples = 154
value = [67, 42, 45]
class = Circle
gini = 0.654
samples = 323
value = [135, 84, 104]
class = Circle
gini = 0.6339
samples = 172
value = [79, 35, 58]
class = Circle
gini = 0.6222
samples = 297
value = [150, 70, 77]
class = Circle