LastFMTagsExploratory

April 28, 2015

In [ ]:
```
#How about a month metric?
# Load and display details of the packaged datasets
import urllib2
import json
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

response_artists = urllib2.urlopen('http://ws.audioscrobbler.com/2.0/?method=chart.
                                    gettopartists&api_key=3fefe94153850a5c6b7ac1ca3a1a9096&forma
data_artists = json.loads(response_artists) # "loads UTF decoded text above
  # into Python
raw_artists= data_artists['artists']['artist']

#creates three arrays of 1 dimension, all with the indexes lined up. Do analysis off
    #this
name = range(1000)
listeners = range(1000)
playcount = range(1000)
counter = 0
for artist in raw_artists:
    name[counter] = artist['name'] #array index 0 is value, in list index 0 is key&value
    listeners[counter] = artist['listeners'] # [] signifies the index position
    playcount[counter] = artist['playcount']
   # print(artist["name"]+" "+artist["listeners"] ) #with artist list open, prints name within
    counter+= 1 #counter= counter + 1

#Different x-y correlation metrics
N=1000
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2 # 0 to 15 point radiuses
plt.scatter(playcount, listeners, s=area, c=colors, alpha=0.1)
plt.xlabel('listeners')
plt.ylabel('playcount')
plt.show()

N=1000
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2
plt.scatter(playcount, listeners, s=area, c=colors, alpha=0.1)
plt.xlabel('name')
plt.ylabel('listeners')
plt.show()
```

```
N=1000
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2 #
plt.scatter(playcount, listeners, s=area, c=colors, alpha=0.1)
plt.xlabel('name')
plt.ylabel('listeners')
plt.show()

N=1000
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2
plt.scatter(playcount, listeners, s=area, c=colors, alpha=0.1)
plt.xlabel('name')
plt.ylabel('playcount')
plt.show()
```