

Maps and Twitter data

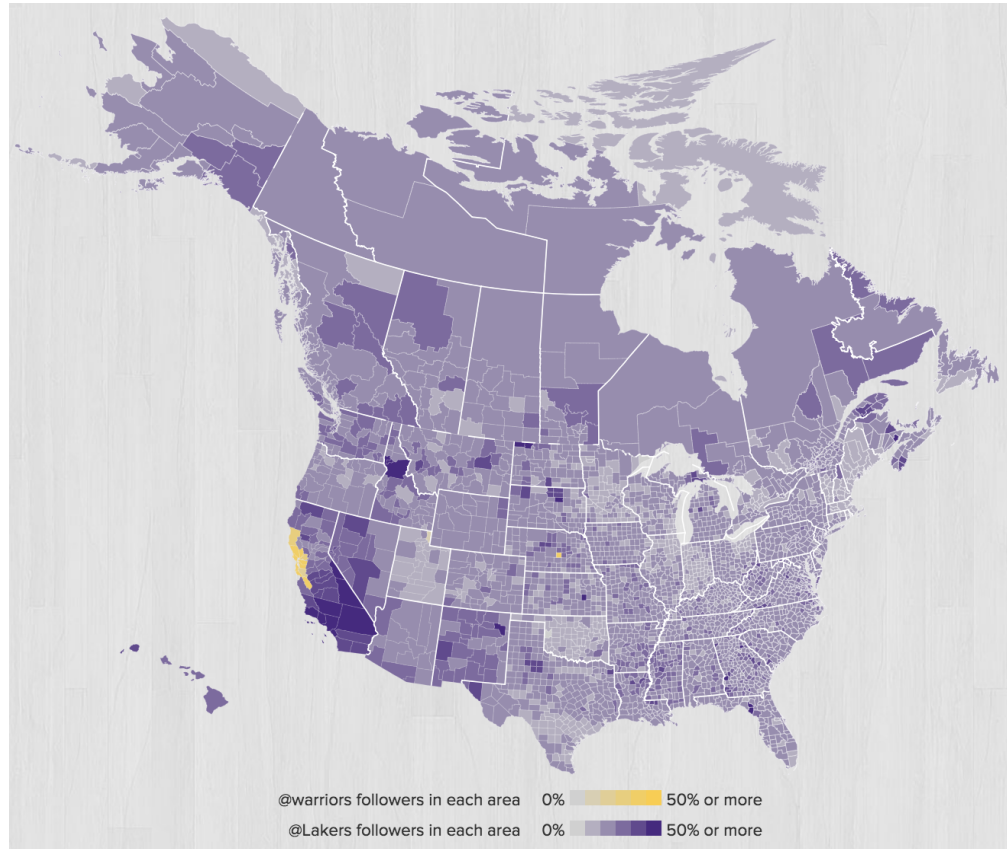
ANALYZING SOCIAL MEDIA DATA IN PYTHON



Alex Hanna

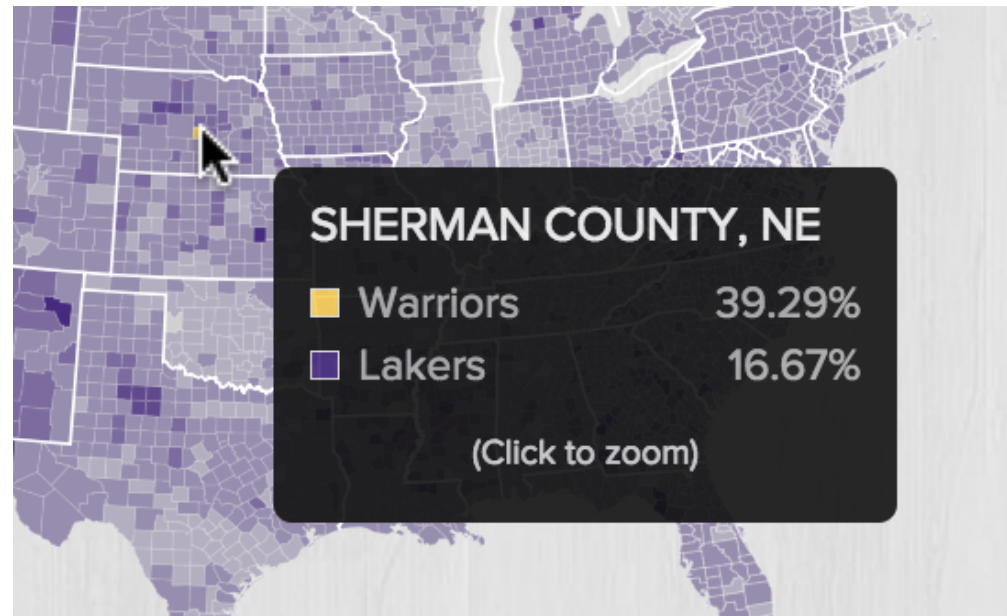
Computational Social Scientist

Why maps?



- Geographical scope
 - Participants or observers?
- Differentiating tweets
 - For or against?

How Twitter gets location data



- Location is device-dependent
- In practice, aggregate geographical to county, state-level

Beware selection biases!

- *Warning:* only 1-3% of Twitter data have geographical data
- Limits the generalizability of inference

Types of geographical data available in Twitter

- Twitter text (most imprecise)
- User location
- Bounding boxes
- Coordinates and points (most precise)

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Geographical data in Twitter JSON

ANALYZING SOCIAL MEDIA DATA IN PYTHON



Alex Hanna

Computational Social Scientist

Locations in Twitter text



Dr. Alex Hanna, Skatin Data Witch

@alexhanna



In Zurich! It's lovely and about as hot as Toronto.

12:32 PM - 4 Jul 2018

User-defined location

**Dr. Alex Hanna, Skatin
Data Witch**

@alexhanna

Tech curriculum dev @GCPcloud.
Sociology PhD. Computational social
scientist. Trans. Roller derby athlete (Kate
Silver #538). She/her.

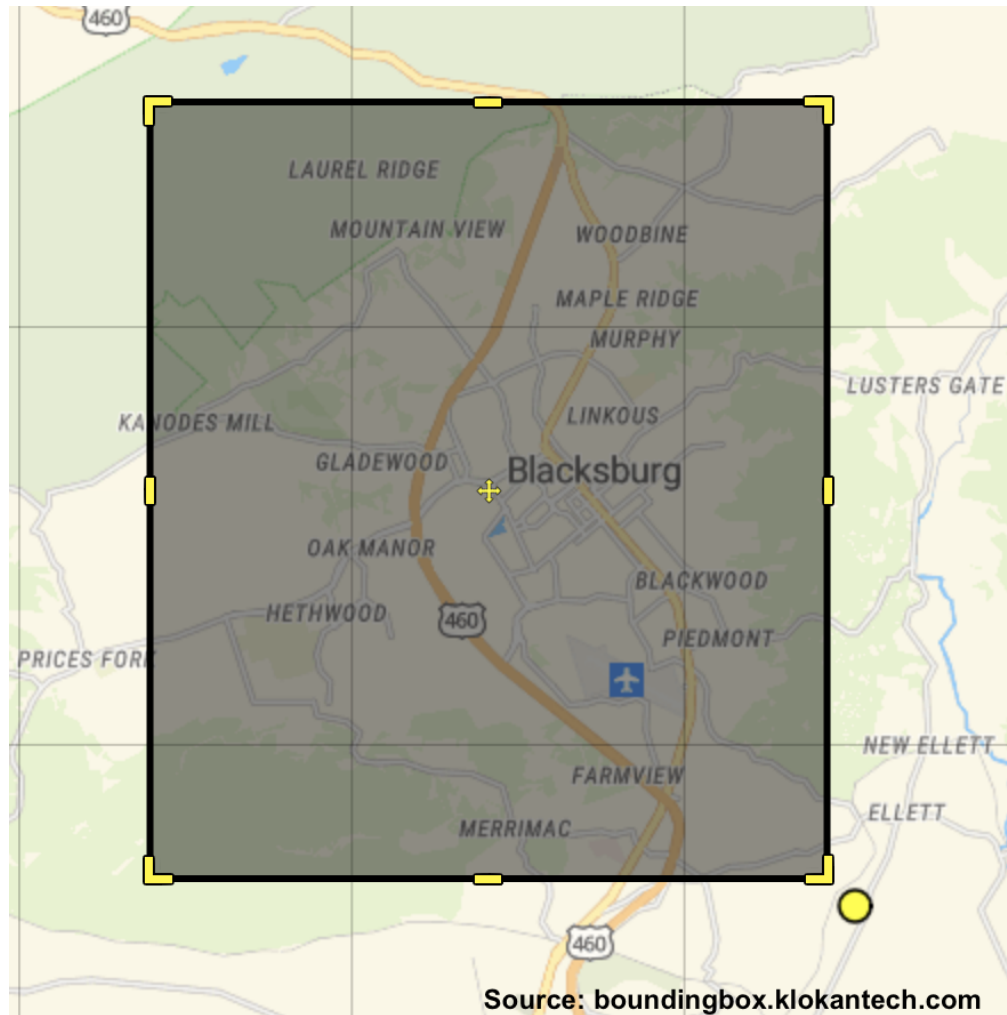
📍 Bay Area

🔗 alex-hanna.com

```
print(tweet['user']['location'])
```

Bay Area

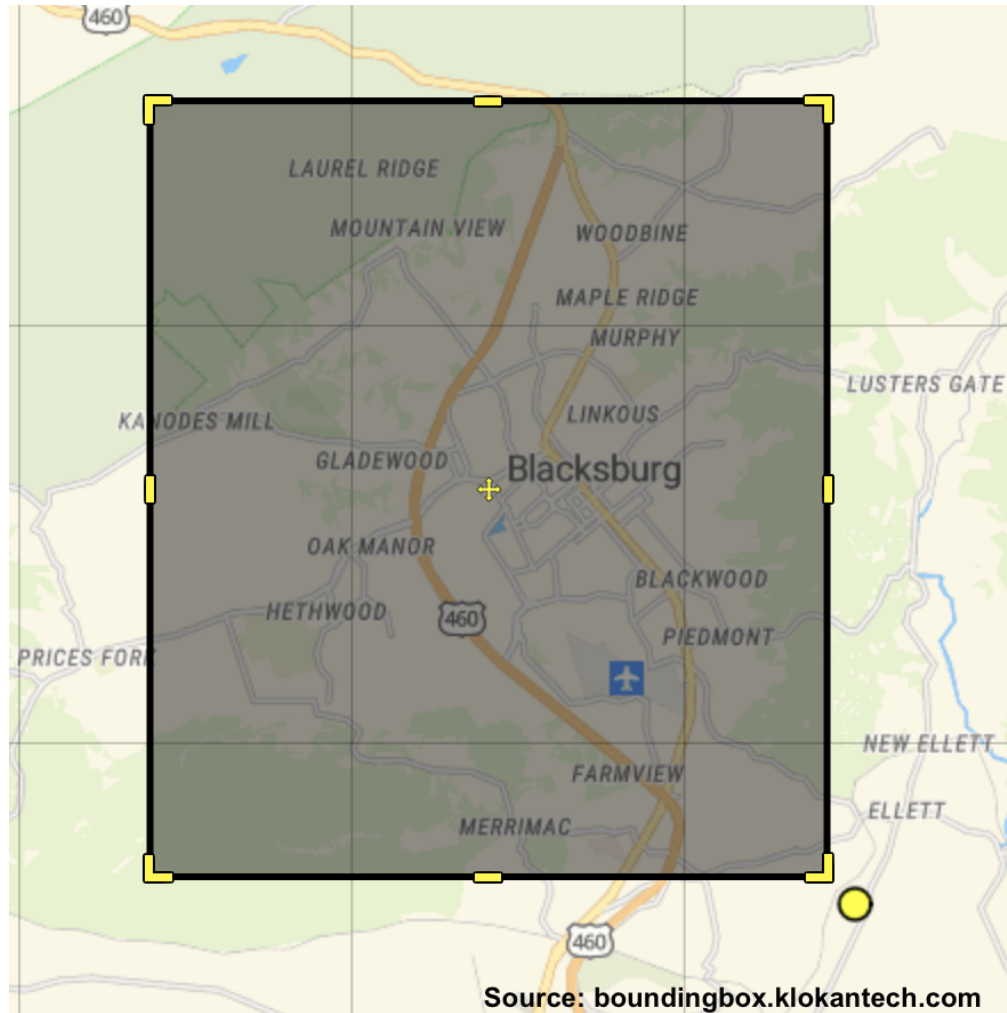
place JSON



```
print(tweet['place'])
```

```
{'attributes': {},  
 'bounding_box': {'coordinates':  
   [[[-80.47611, 37.185195],  
     [-80.47611, 37.273387],  
     [-80.381618, 37.273387],  
     [-80.381618, 37.185195]]],  
   'type': 'Polygon'},  
 'country': 'United States',  
 'country_code': 'US',  
 'full_name': 'Blacksburg, VA',  
 'name': 'Blacksburg',  
 'place_type': 'city',  
 ...}
```

Calculating the centroid



```
coordinates = [  
    [-80.47611, 37.185195],  
    [-80.47611, 37.273387],  
    [-80.381618, 37.273387],  
    [-80.381618, 37.185195]]  
  
longs = np.unique( [x[0] for x  
    in coordinates] )  
lats = np.unique( [x[1] for x  
    in coordinates] )  
  
central_long = np.sum(longs) / 2  
central_lat = np.sum(lats) / 2
```

coordinates JSON



```
print(tweet['coordinates'])
```

```
{'type': 'Point',  
 'coordinates': [-72.2833,  
                 21.7833]}
```

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Creating Twitter maps

ANALYZING SOCIAL MEDIA DATA IN PYTHON

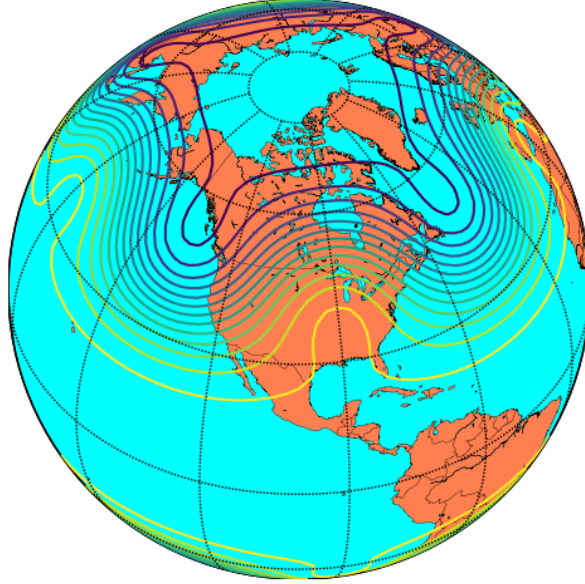


Alex Hanna

Computational Social Scientist

Introducing Basemap

contour lines over filled continent background



Source: <https://matplotlib.org/basemap/users/examples.html>

- Library for plotting two-dimensional maps
- Built on top of matplotlib
- Converts coordinates into map projections

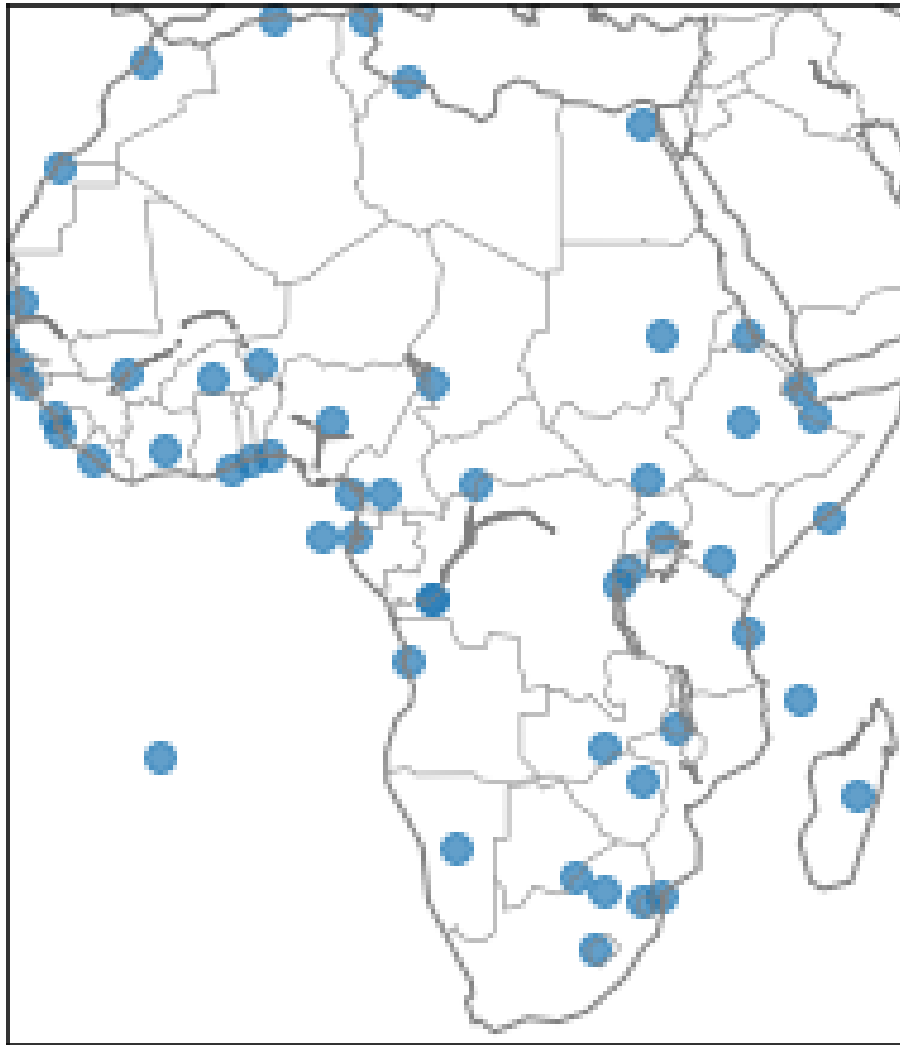
Beginning with Basemap

```
from mpl_toolkits.basemap
import Basemap
m = Basemap(projection='merc',
            llcrnrlat = -35.62,
            llcrnrlon = -17.29,
            urcrnrlat = 37.73,
            urcrnrlon = 51.39)
m.fillcontinents(color='white')
m.drawcoastlines(color='gray')
m.drawcountries(color='gray')
```



Plotting points

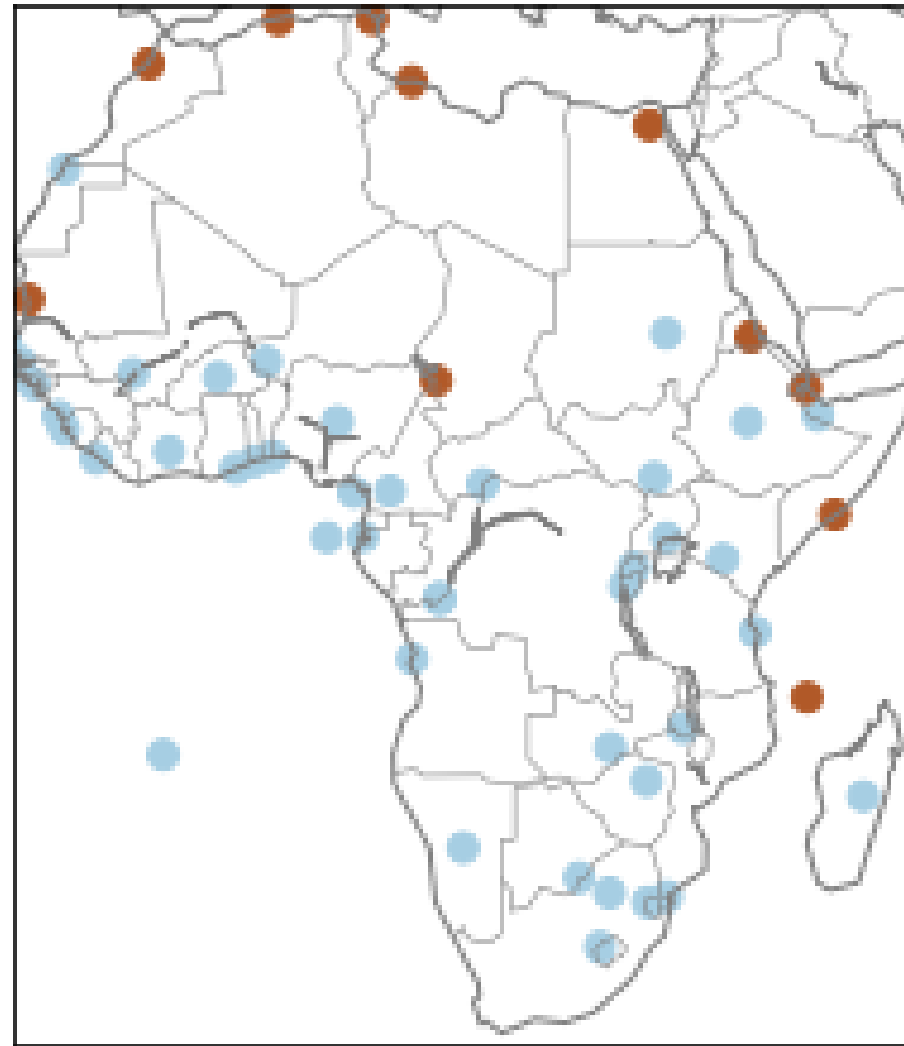
```
africa = pd.read_csv('africa.csv')
longs  = africa['CapitalLongitude']
lats   = africa['CapitalLatitude']
m = Basemap(...)
m.fillcontinents(color='white',
                 zorder = 0)
m.drawcoastlines(color='gray')
m.drawcountries(color='gray')
m.scatter(longs.values,
          lats.values,
          latlon = True,
          alpha = 0.7)
```



Using color

```
africa = pd.read_csv('africa.csv')
longs  = africa['CapitalLongitude']
lats   = africa['CapitalLatitude']
arabic = africa['Arabic']

m = Basemap(...)
m.fillcontinents(color='white',
                 zorder = 0)
m.drawcoastlines(color='gray')
m.drawcountries(color='gray')
m.scatter(longs.values,
          lats.values,
          latlon = True,
          c = arabic.values,
          cmap = 'Paired',
          alpha = 1)
```



Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Congratulations!

ANALYZING SOCIAL MEDIA DATA IN PYTHON



Alex Hanna

Computational Social Scientist

Next steps

ANALYZING SOCIAL MEDIA DATA IN PYTHON