Geocode

Mapping the H2A visa work sites

In [73]:
```
import os
import csv
import time
import random
import calculate
import numpy as np
import pandas as pd
import timeout_decorator
from geopy import Location
from geopy.geocoders import Bing
```

In [74]:
```
import warnings
warnings.filterwarnings("ignore")
```

Read in all the visas

In [75]:
```
df = pd.concat([
    pd.read_csv("./output/transformed_master_cases.csv"),
    pd.read_csv("./output/transformed_sub_cases.csv"),
])
```

Extract the distinct locations

In [76]:
```
locations = df.groupby(['city', 'state']).size().reset_index().rename(columns=
{0: "count"})
```

Read in previously geocoded locations

In [77]:
```
geocoded = pd.read_csv("./output/geocoded.csv")
```

In [78]:
```
geocode_cache = dict(
    (d['key'], d) for i, d in geocoded.iterrows()
)
```

Identify how many remain unmapped

In [79]:
```
df['key'] = df.apply(lambda x: "{}, {}".format(x.city, x.state), axis=1)
```

In [80]:
```
not_geocoded = df[~df.key.isin(geocoded.key)]
```

In [81]:
```
print "{:,} of {:,} geocoded ({}%)".format(
    len(df) - len(not_geocoded),
    len(df),
    calculate.percentage(len(df) - len(not_geocoded), len(df))
)
```

```
83,087 of 83,088 geocoded (99.9987964568%)
```

Extract the unmapped locations

In [82]:
```
unmapped = not_geocoded.groupby(['key']).size().reset_index().rename(columns={
0: "count"})
```

In [83]:
```
df_list = list(unmapped.iterrows())
```

In [84]:
```
random.shuffle(df_list)
```

Try to geocode them

In [85]:
```
@timeout_decorator.timeout(10)
def bingit(key):
    bing = Bing(os.getenv("BING_API_KEY"), timeout=10)
    address = "{}, United States".format(key)
    print "Geocoding {}".format(address)
    try:
        geocode_cache[key]
        print "Already mapped"
        return
    except KeyError:
        pass

    result = bing.geocode(address, exactly_one=False)
    if not result:
        return
    first_result = result[0]

    print "Mapped to {}".format(first_result)
    geocode_cache[key] = first_result
    time.sleep(0.5)
```

In [86]:
```
for i, row in df_list:
    try:
        bingit(row.key)
    except:
        print "TIMEOUT"
        continue
```

```
Geocoding Juniata, NE, United States
Mapped to Juniata, NE, United States
```

Merged the newly geocoded locations with the old ones

In [87]:
```
def transform_geocode(key, value):
    if isinstance(value, pd.Series):
        return [key, value['geocoder_address'], value['lat'], value['lng'], va
lue['geocoder_type']]
    return [key, value.address, value.latitude, value.longitude, "bing"]
```

In [88]:
```
rows = [transform_geocode(k, v) for k, v in geocode_cache.items()]
```

In [89]:
```
rows.sort(key=lambda x:x[0])
```

Save the geocoded locations

In [90]:
```
with open("./output/geocoded.csv", 'w') as f:
    w = csv.writer(f)
    w.writerow(["key", "geocoder_address", "lat", "lng", "geocoder_type"])
    w.writerows(rows)
```

Merge geocoded points onto cases

In [91]:
```
mapped = pd.read_csv("./output/geocoded.csv")
```

In [92]:
```
def create_key(row):
    # Skip any nulls
    if row.city in [np.NaN, 'nan', '']:
        return ''
    elif row.state in [np.NaN, 'nan', '']:
        return ''
    else:
        return "{}, {}".format(row.city, row.state)
```

In [93]:
```
def add_points(name):
    df = pd.read_csv("./output/transformed_{}.csv".format(name))
    df['key'] = df.apply(create_key, axis=1)
    mapped_df = df.merge(mapped, on=["key"], how="left")
    mapped_df.drop('key', axis=1, inplace=True)
    mapped_df.to_csv("./output/geocoded_{}.csv".format(name), index=False, enc
oding="utf-8")
```

In [94]:
```
add_points("master_cases")
```

In [95]:
```
add_points("sub_cases")
```

In [96]:
```
add_points("all_cases")
```