```
In [1]: import sqlite3
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [ ]:
```

```
In [2]: con = sqlite3.connect('dallas-ois.sqlite')
```

```
In [ ]:
```

```
In [3]: incidents = pd.read_sql('SELECT * FROM incidents', con)
        subjects = pd.read_sql('SELECT * FROM subjects', con)
        officers = pd.read_sql('SELECT * FROM officers', con)
```

```
In [ ]:
```

```
In [4]: len(incidents), len(subjects), len(officers)

Out[4]: (219, 223, 370)
```

```
In [ ]:
```

```
In [5]: subject_incidents = pd.read_sql('SELECT * FROM subjects JOIN incidents ON subjects. ca
```

```
In [ ]:
```

```
In [6]: len(subject_incidents)

Out[6]: 223
```

```
In [ ]:
```

```
In [7]: subject_incidents.dtypes
```

Out[7]:

```
case_number                    object
race                           object
gender                         object
last_name                      object
first_name                     object
full_name                      object
case_number                    object
date                           object
location                       object
subject_statuses               object
subject_weapon                 object
subjects                       object
subject_count                   int64
officers                       object
officer_count                   int64
grand_jury_disposition         object
attorney_general_forms_url     object
summary_url                    object
summary_text                   object
latitude                      float64
longitude                     float64
dtype: object
```

In [ ]:

```
```

In [8]:

```
subject_incidents.isnull().sum()
```

Out[8]:

```
case_number                     0
race                            0
gender                          0
last_name                       0
first_name                     18
full_name                       0
case_number                     0
date                            0
location                        0
subject_statuses                0
subject_weapon                  0
subjects                        0
subject_count                   0
officers                        0
officer_count                   0
grand_jury_disposition         88
attorney_general_forms_url    221
summary_url                     3
summary_text                    3
latitude                        9
longitude                       9
dtype: int64
```

In [ ]:

```
```

In [9]:

```
subject_incidents.groupby('subject_statuses').count()
```

Out[9]:

subject_statuses	case_number	race	gender	last_name	first_name	full_name	case_number	date	lo
1 Deceased 1 Injured	2	2	2	2	2	2	2	2	
2 Injured	1	1	1	1	1	1	1	1	
Deceased	69	69	69	69	67	69	69	69	
Deceased Injured	2	2	2	2	2	2	2	2	
Injured	60	60	60	60	60	60	60	60	
Other	2	2	2	2	2	2	2	2	
Shoot and Miss	87	87	87	87	71	87	87	87	

In [ ]:

In [10]:

```
subject_incidents.groupby('race').count()
```

Out[10]:

race	case_number	gender	last_name	first_name	full_name	case_number	date	location	subject_st
A	2	2	2	2	2	2	2	2	
B	111	111	111	104	111	111	111	111	
L	72	72	72	61	72	72	72	72	
W	38	38	38	38	38	38	38	38	

In [ ]:

In [11]:

```
dataset = subject_incidents[['race', 'subject_statuses', 'latitude', 'longitude']].drc
dataset.head()
```

Out[11]:

	race	subject_statuses	latitude	longitude
9	L	Deceased	32.68642	-96.908674
10	B	Deceased	32.86400	-96.898998
11	W	Shoot and Miss	32.81482	-96.826787
12	B	Injured	32.77540	-96.767489
13	B	Shoot and Miss	32.74417	-96.828470

In [ ]:

In [12]:

```
len(dataset)
```

Out[12]: 214

In [ ]:

In [13]:

```
mapping = {'B':'blue', 'W':'yellow', 'A':'red', 'L':'green'}
dataset['race'] = dataset['race'].apply(lambda X:mapping[X])
dataset.head()
```

Out[13]:

	race	subject_statuses	latitude	longitude
9	green	Deceased	32.68642	-96.908674
10	blue	Deceased	32.86400	-96.898998
11	yellow	Shoot and Miss	32.81482	-96.826787
12	blue	Injured	32.77540	-96.767489
13	blue	Shoot and Miss	32.74417	-96.828470

In [ ]:

In [14]:

```
mapping = {'Deceased':1000, 'Injured':500, 'Shoot and Miss':250}
dataset['subject_statuses'] = dataset['subject_statuses'].apply(lambda x:mapping.get(x
dataset.head()
```

Out[14]:

	race	subject_statuses	latitude	longitude
9	green	1000	32.68642	-96.908674
10	blue	1000	32.86400	-96.898998
11	yellow	250	32.81482	-96.826787
12	blue	500	32.77540	-96.767489
13	blue	250	32.74417	-96.828470

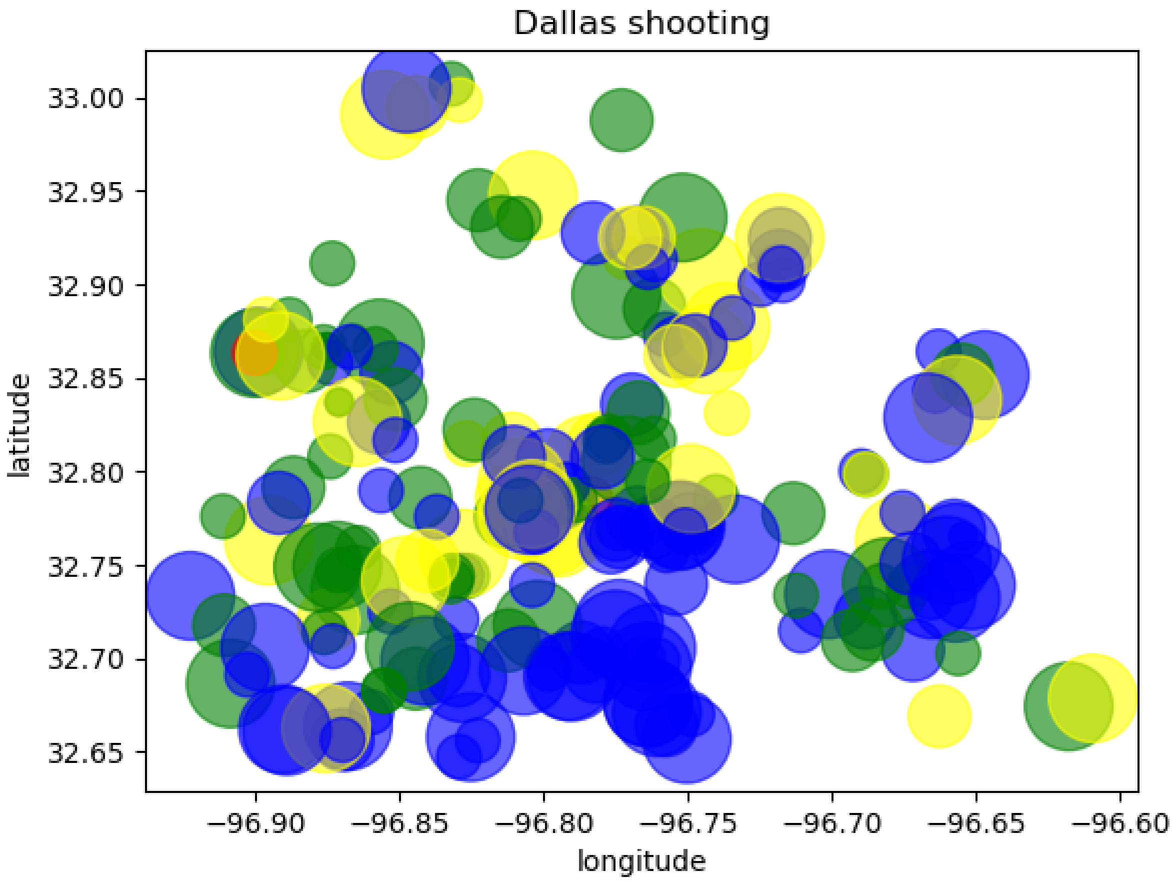
In [ ]:

In [15]:

```
dataset.plot.scatter(x='longitude', y='latitude', s='subject_statuses', c='race', titl
```

Out[15]:

```
<AxesSubplot:title={'center':'Dallas shooting'}, xlabel='longitude', ylabel='latitud
e'>
```



In [ ]:

In [16]:

```
import folium
m = folium.Map(location=[32.8, - 96.8])
```

In [17]:

```
m
```

Out[17]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [18]:

```
for _, row in dataset.iterrows():

    folium.CircleMarker(
        location=[row['latitude'], row['longitude']],
        radius=row['subject_statuses']//100,
        color=row['race'],
        fill=True,
        fill_color=row['race']
    ).add_to(m)
```

In [20]:

```
m
```

Out[20]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [ ]:

```
```