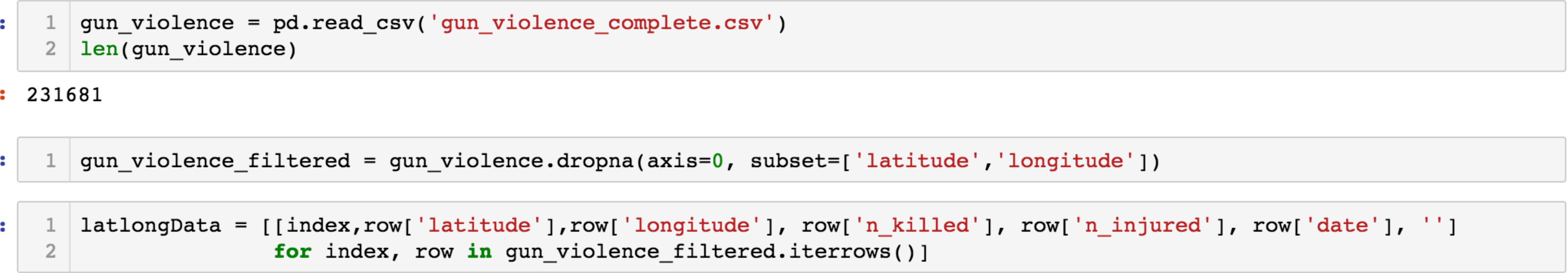
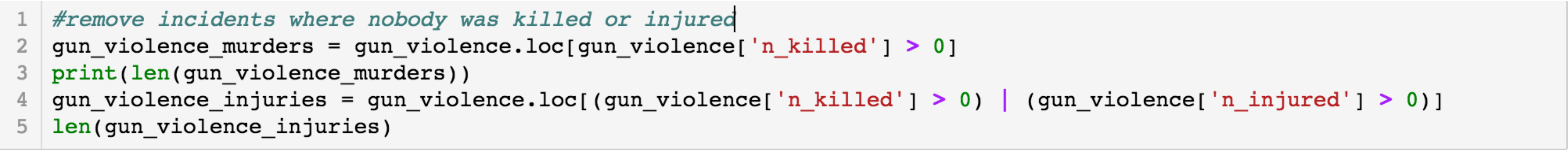


```python
gun_violence = pd.read_csv('gun_violence_complete.csv')
len(gun_violence)
```

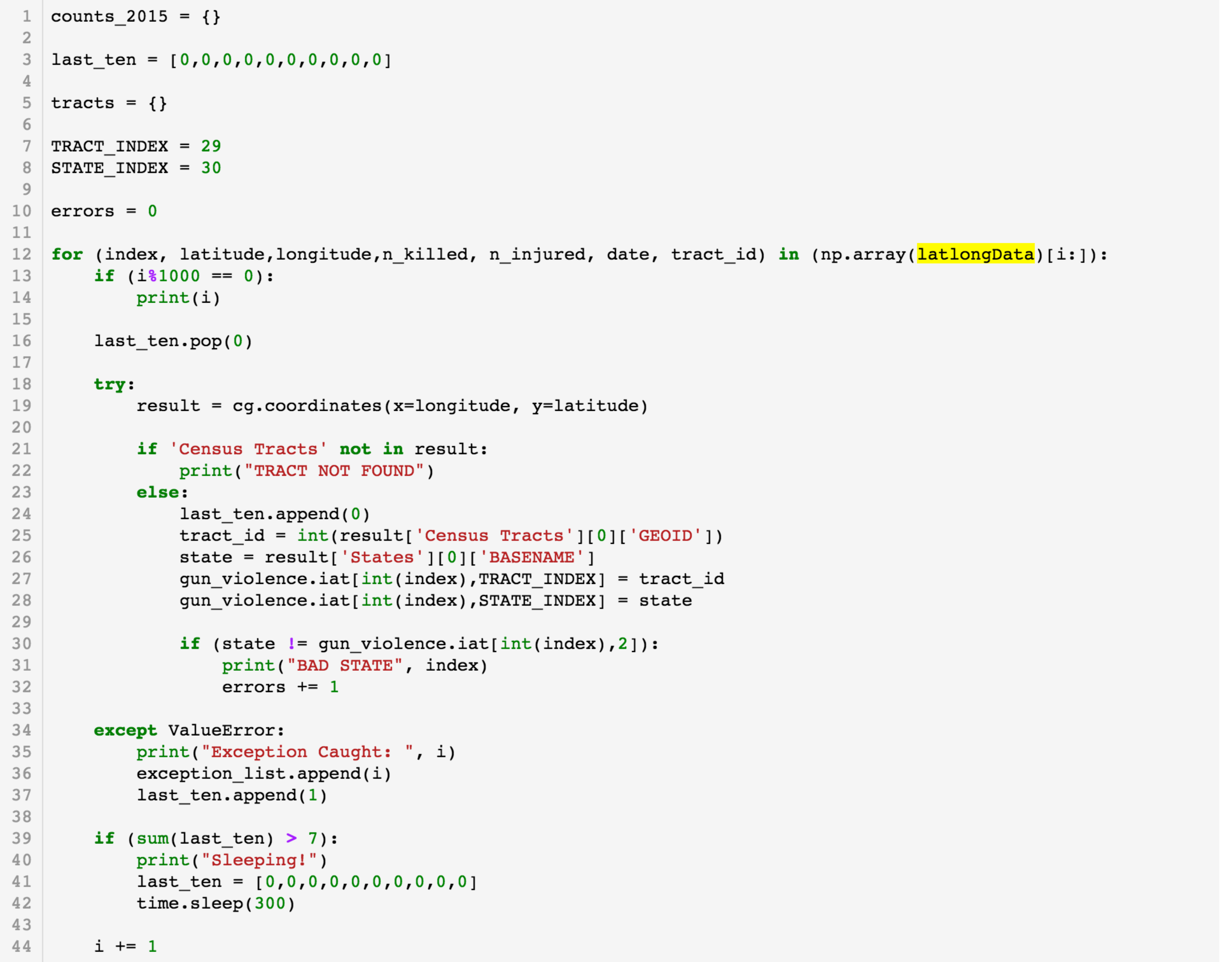
```
231681
```

```python
gun_violence_filtered = gun_violence.dropna(axis=0, subset=['latitude','longitude'])
```

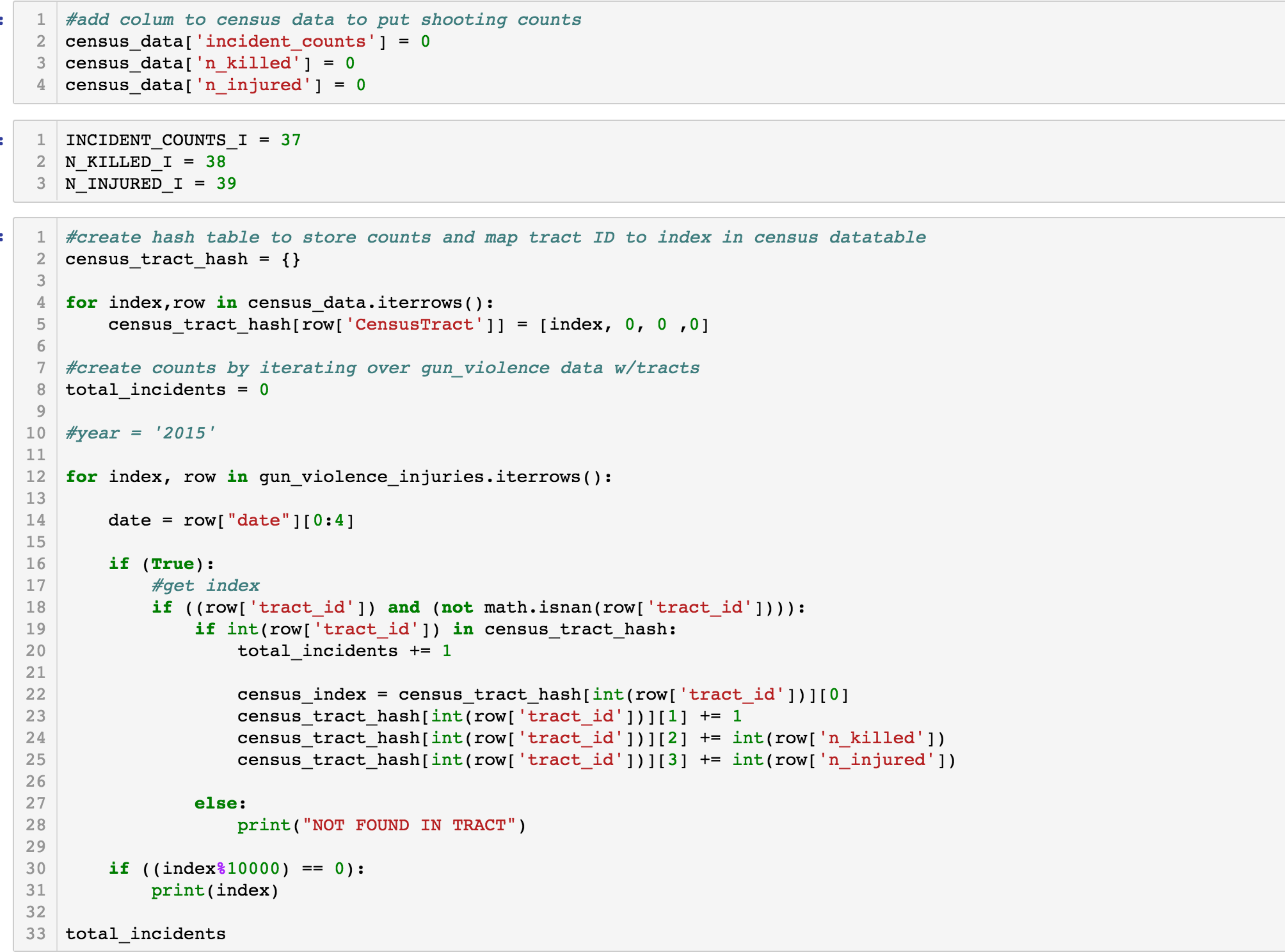
```python
latlongData = [[index,row['latitude'],row['longitude'], row['n_killed'], row['n_injured'], row['date'], '']
               for index, row in gun_violence_filtered.iterrows()]
```



```python
#remove incidents where nobody was killed or injured
gun_violence_murders = gun_violence.loc[gun_violence['n_killed'] > 0]
print(len(gun_violence_murders))
gun_violence_injuries = gun_violence.loc[(gun_violence['n_killed'] > 0) | (gun_violence['n_injured'] > 0)]
len(gun_violence_injuries)
```



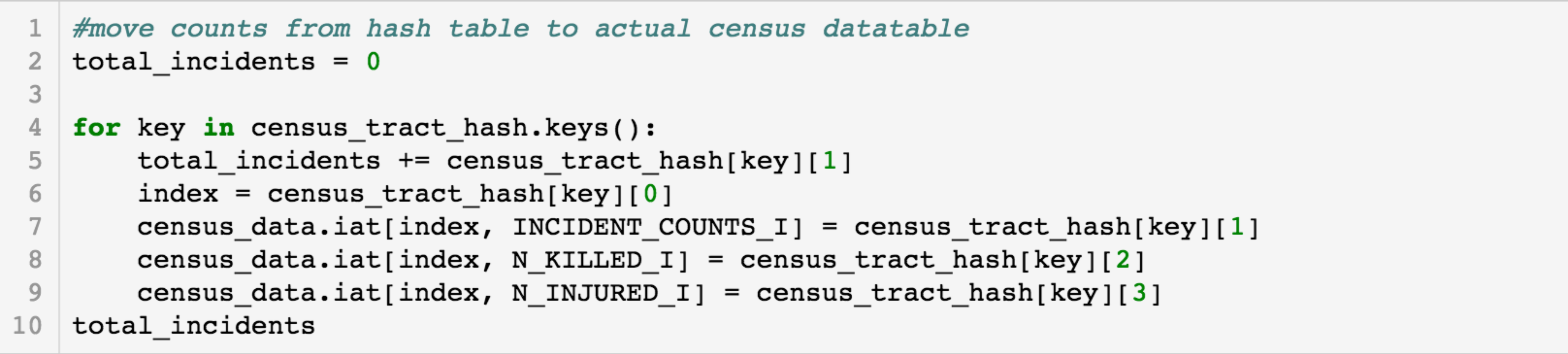
```python
counts_2015 = {}

last_ten = [0,0,0,0,0,0,0,0,0,0]

tracts = {}

TRACT_INDEX = 29
STATE_INDEX = 30

errors = 0

for (index, latitude,longitude,n_killed, n_injured, date, tract_id) in (np.array(latlongData)[i:]):
    if (i%1000 == 0):
        print(i)

    last_ten.pop(0)

    try:
        result = cg.coordinates(x=longitude, y=latitude)

        if 'Census Tracts' not in result:
            print("TRACT NOT FOUND")
        else:
            last_ten.append(0)
            tract_id = int(result['Census Tracts'][0]['GEOID'])
            state = result['States'][0]['BASENAME']
            gun_violence.iat[int(index),TRACT_INDEX] = tract_id
            gun_violence.iat[int(index),STATE_INDEX] = state

            if (state != gun_violence.iat[int(index),2]):
                print("BAD STATE", index)
                errors += 1

    except ValueError:
        print("Exception Caught: ", i)
        exception_list.append(i)
        last_ten.append(1)

    if (sum(last_ten) > 7):
        print("Sleeping!")
        last_ten = [0,0,0,0,0,0,0,0,0,0]
        time.sleep(300)

    i += 1
```



```
#add colum to census data to put shooting counts
census_data['incident_counts'] = 0
census_data['n_killed'] = 0
census_data['n_injured'] = 0
```

```
INCIDENT_COUNTS_I = 37
N_KILLED_I = 38
N_INJURED_I = 39
```

```
#create hash table to store counts and map tract ID to index in census datatable
census_tract_hash = {}

for index,row in census_data.iterrows():
    census_tract_hash[row['CensusTract']] = [index, 0, 0 ,0]

#create counts by iterating over gun_violence data w/tracts
total_incidents = 0

#year = '2015'

for index, row in gun_violence_injuries.iterrows():

    date = row["date"][0:4]

    if (True):
        #get index
        if ((row['tract_id']) and (not math.isnan(row['tract_id']))):
            if int(row['tract_id']) in census_tract_hash:
                total_incidents += 1

                census_index = census_tract_hash[int(row['tract_id'])][0]
                census_tract_hash[int(row['tract_id'])][1] += 1
                census_tract_hash[int(row['tract_id'])][2] += int(row['n_killed'])
                census_tract_hash[int(row['tract_id'])][3] += int(row['n_injured'])

            else:
                print("NOT FOUND IN TRACT")

    if ((index%10000) == 0):
        print(index)

total_incidents
```



```python
#move counts from hash table to actual census datatable
total_incidents = 0

for key in census_tract_hash.keys():
    total_incidents += census_tract_hash[key][1]
    index = census_tract_hash[key][0]
    census_data.iat[index, INCIDENT_COUNTS_I] = census_tract_hash[key][1]
    census_data.iat[index, N_KILLED_I] = census_tract_hash[key][2]
    census_data.iat[index, N_INJURED_I] = census_tract_hash[key][3]
total_incidents
```