JUSTIN PATE

IST-652

CRIME DATA--MAP

```
#DEFINE WORKING DIRECTORY LINK TO GOOGLE DRIVE
import os
import plotly.express as px #plotly express for plotting
os.getcwd()
from google.colab import drive
drive.mount('/content/drive/', force_remount=True)
pathlocation = '/content/drive/My Drive/Colab Notebooks/IST_652/PROJECT/'
os.chdir(pathlocation)
os.getcwd()
```

```
Mounted at /content/drive/
'/content/drive/My Drive/Colab Notebooks/IST_652/PROJECT'
```

```
#import crime file.  File is in zip format so needs to have compression option listed
import pandas as pd
import zipfile

df = pd.read_csv('crime_open_database_core_2018.csv.gz', compression='gzip', sep=',')
```

```
/usr/local/lib/python3.6/dist-packages/IPython/core/interactiveshell.py:2718: DtypeWarning: Columns (9,10,12,13) have mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

```
#show a sample of the data frame
df
```

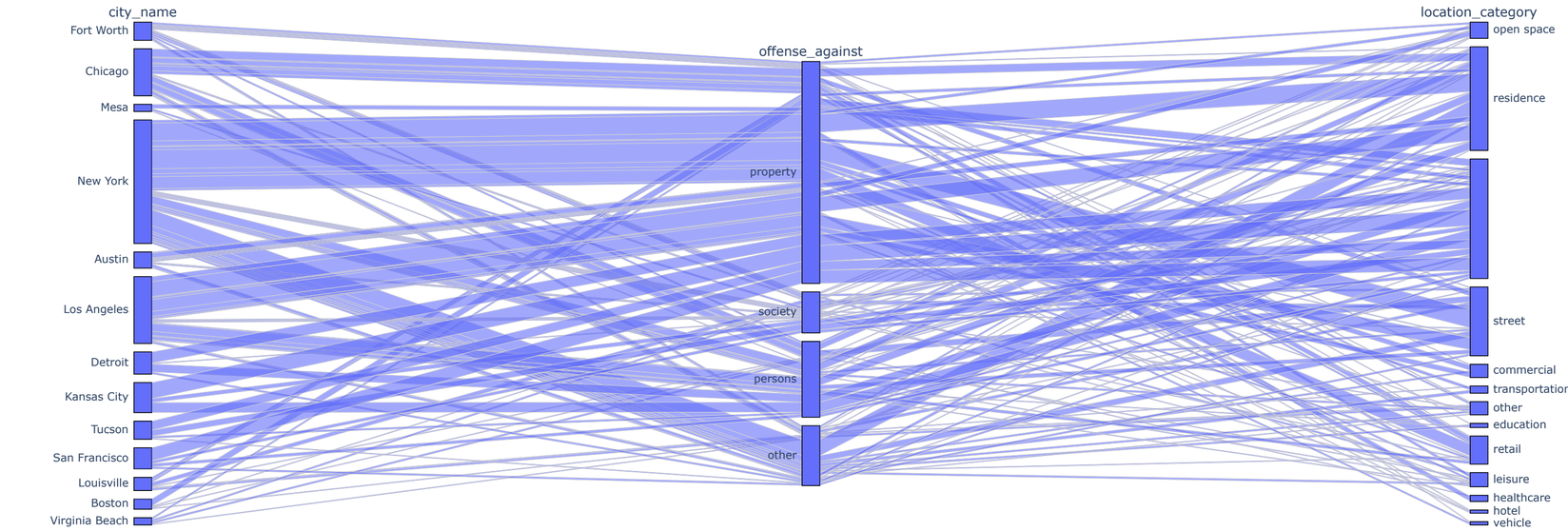
	uid	city_name	offense_code	offense_type	offense_group	offense_against	date_single	longitude	latitude	location_type	location_category	census_block	date_start	date_end
0	1187951	Austin	22U	other burglary/breaking & entering	burglary/breaking & entering	property	2018-01-01 00:00	-97.710191	30.349433	other	other	484530018043000	NaN	NaN
1	1187952	Austin	520	weapon law violations	weapon law violations	society	2018-01-01 00:00	-97.741558	30.411489	residence	residence	484530017541002	NaN	NaN
2	1187953	Austin	23H	all other larceny	larceny/theft offenses	property	2018-01-01 00:00	-97.741119	30.305911	vehicle parking	open space	484530002031007	NaN	NaN
3	1187954	Austin	290	destruction/damage/vandalism of property (exce...	destruction/damage/vandalism of property (exce...	property	2018-01-01 00:00	-97.672452	30.363035	residence	residence	484530018333009	NaN	NaN
4	1187955	Austin	290	destruction/damage/vandalism of property (exce...	destruction/damage/vandalism of property (exce...	property	2018-01-01 00:00	-97.699980	30.258932	other	other	484530009021000	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1608784	18882009	Virginia Beach	35A	drug/narcotic violations	drug/narcotic offenses	society	2018-12-31 23:10	-76.065879	36.759361	NaN	NaN	518100454221000	NaN	NaN
1608785	18882010	Virginia Beach	520	weapon law violations	weapon law violations	society	2018-12-31 23:10	-76.065879	36.759361	NaN	NaN	518100454221000	NaN	NaN
1608786	18882011	Virginia Beach	35A	drug/narcotic violations	drug/narcotic offenses	society	2018-12-31 23:23	-76.159443	36.830235	NaN	NaN	518100460051035	NaN	NaN
1608787	18882012	Virginia Beach	12U	other robbery	robbery	property	2018-12-31 23:30	-75.971098	36.833776	NaN	NaN	518100440031000	NaN	NaN
1608788	18882013	Virginia Beach	23F	theft from motor vehicle (except theft of moto...	larceny/theft offenses	property	2018-12-31 23:59	-76.089190	36.800369	NaN	NaN	518100454053001	NaN	NaN

1608789 rows × 14 columns

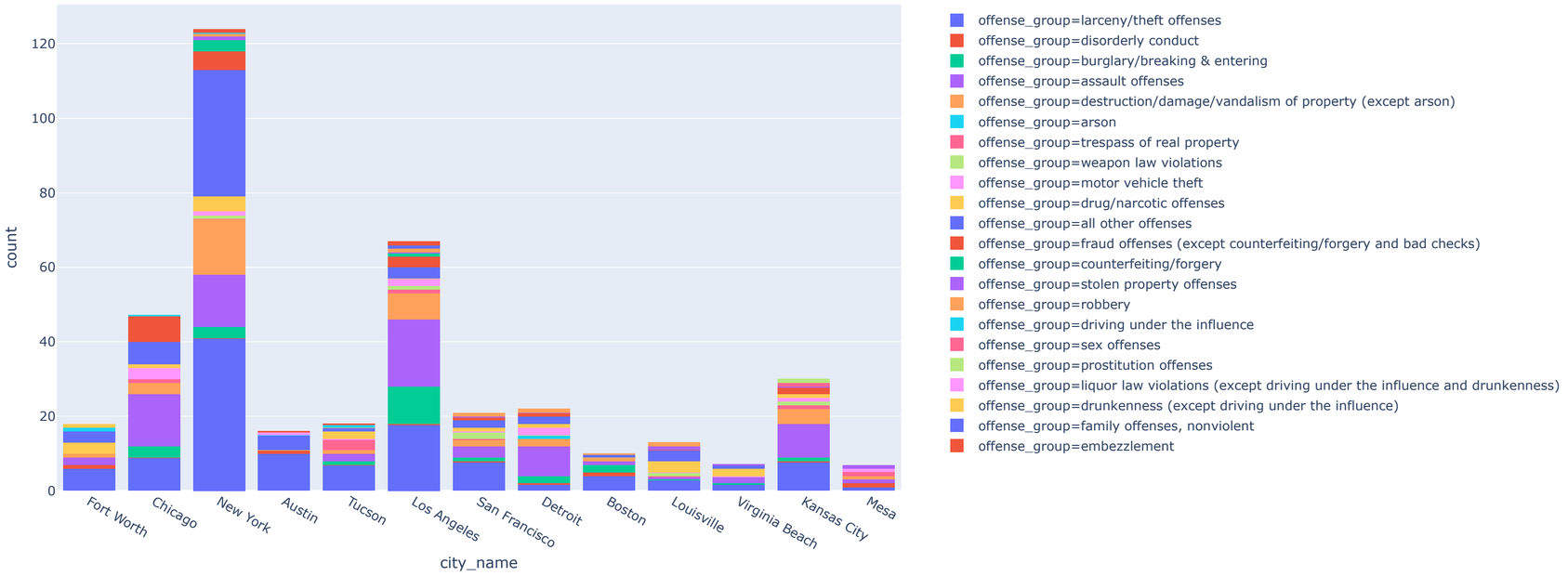
```
df.ftypes
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning:

DataFrame.ftypes is deprecated and will be removed in a future version. Use DataFrame.dtypes instead.

uid                  int64:dense
city_name           object:dense
offense_code        object:dense
offense_type        object:dense
offense_group       object:dense
offense_against     object:dense
date_single         object:dense
longitude          float64:dense
latitude           float64:dense
location_type       object:dense
location_category   object:dense
census_block         int64:dense
date_start          object:dense
date_end            object:dense
dtype: object
```

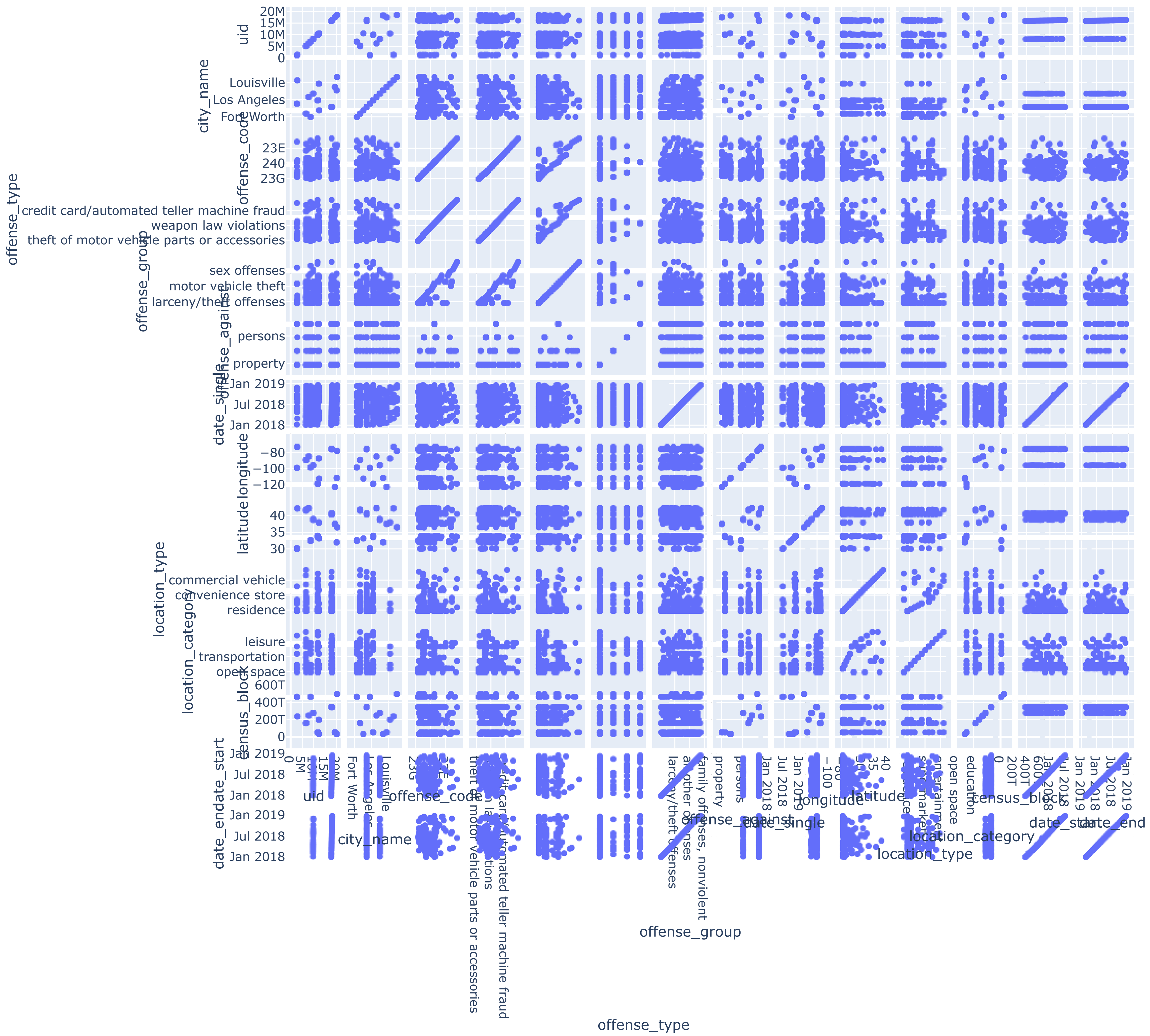
```
dfchart = df.sample(n=400)

px.parallel_categories(dfchart, dimensions=[])
```



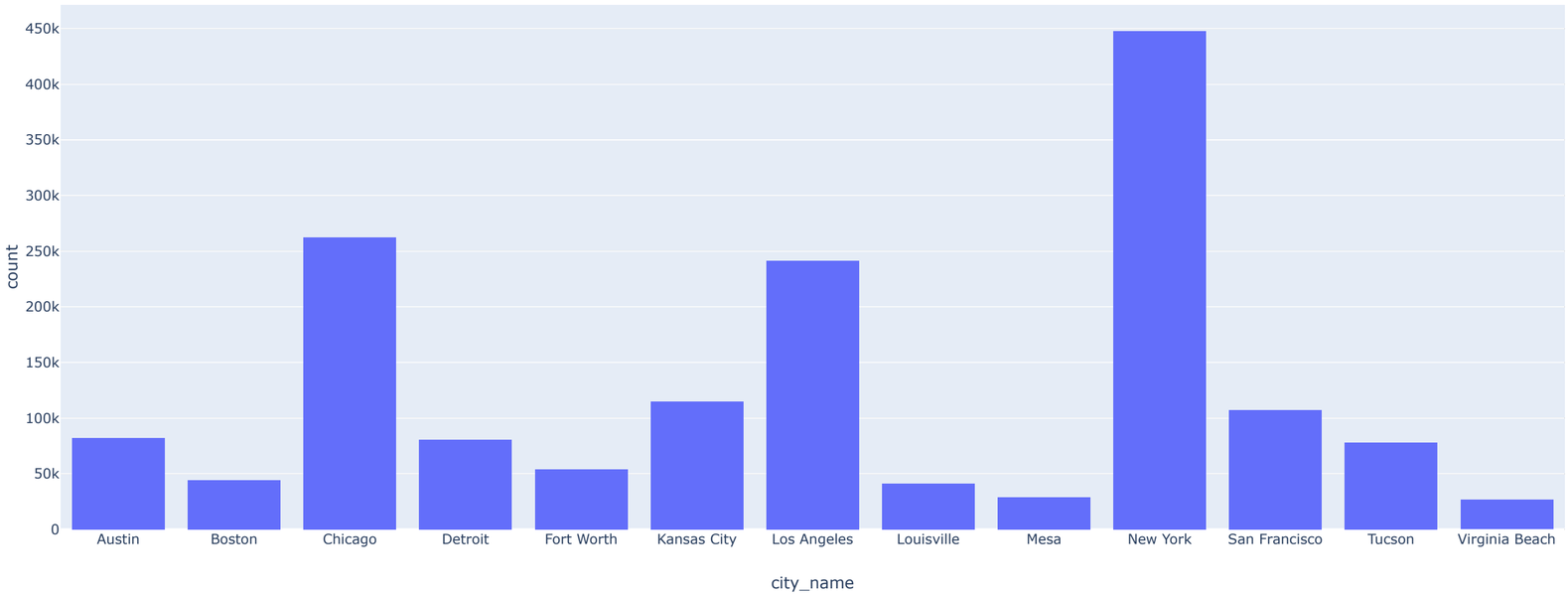

```
px.histogram(dfchart, x='city_name',color="offense_group")
```




```
px.scatter_matrix(dfchart, height=1200,width=1200)
```




```
px.histogram(df, x='city_name')
```




```
#look at the city options
dfcities = df.groupby('city_name').count()
dfcities = dfcities.reset_index()
dfcities[['city_name']]
```

	city_name
0	Austin
1	Boston
2	Chicago
3	Detroit
4	Fort Worth
5	Kansas City
6	Los Angeles
7	Louisville
8	Mesa
9	New York
10	San Francisco
11	Tucson
12	Virginia Beach

```
#define a function that will filter to the specific city
def citycrime(citystr):
  dfmap=df.loc[df['city_name'] == citystr]
  dfmap = dfmap[['latitude','longitude', 'offense_group', 'offense_code']]
  return dfmap.sample(n=400)
```

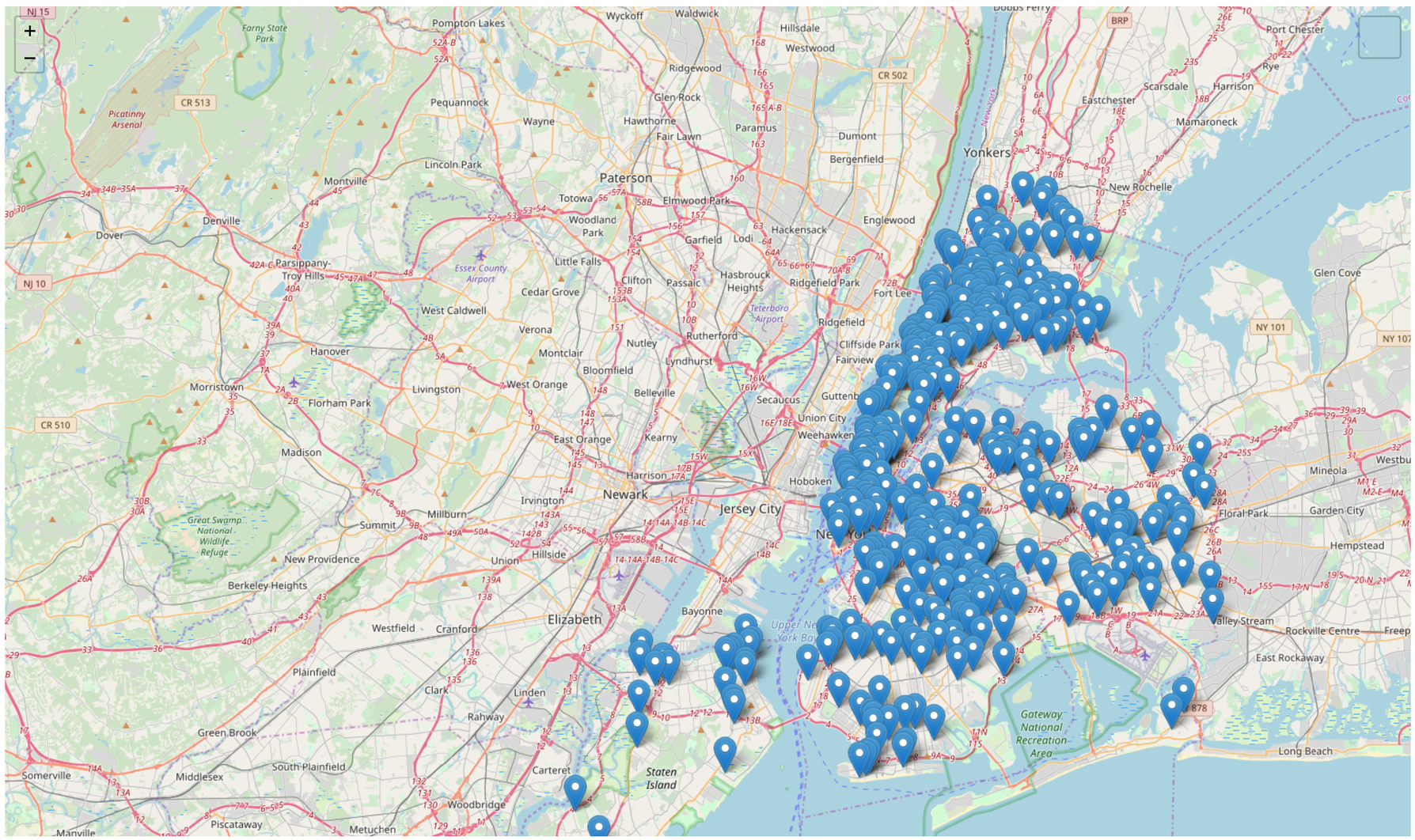
Interactive map

The below map will plot based on the city selection

```
# Import the Folium library.
def displaymap(z):
  import folium
```

```
#sample tile styles below
    #Stamen Terrain
    #Stamen Toner
    #OpenStreetMap
#https://earthengine.googleapis.com/map/"+mapID['mapid']+"/{z}/{x}/{y}?token="+mapID['token']
# Define a method for displaying Earth Engine image tiles to folium map.
def add_ee_layer(self, eeImageObject, visParams, name):
  mapID = ee.Image(eeImageObject).getMapId(visParams)
  folium.raster_layers.TileLayer(tiles = "Stamen Terrain",
                                 attr = "Map Data &copy; <a href='https://earthengine.google.com/'>Google Earth Engine</a>",
                                 name = name,overlay = True,control = True).add_to(self)

# Add EE drawing method to folium.
folium.Map.add_ee_layer = add_ee_layer

# Set visualization parameters.
visParams = {'min':0, 'max':500, 'height':500, 'palette':['#1B1B1B','#1B1B1B','#1B1B1B','#1B1B1B']}
#visParams = {'min':0, 'max':500, 'height':500, 'palette':['225ea8','41b6c4','a1dab4','ffffcc']}

# Create a folium map object.
#also get average lat and long of dataframe

myMap = folium.Map(location=[dfformap["latitude"].mean(), dfformap["longitude"].mean()], zoom_start=z, tiles = "OpenStreetMap")

# Add the elevation model to the map object.
#myMap.add_ee_layer(dem, visParams, 'DEM')

# Add a layer control panel to the map.
myMap.add_child(folium.LayerControl())

#add crime data
crimedata = dfformap
for index, row in crimedata.iterrows():
  folium.Marker(location=[row["latitude"], row["longitude"]],tooltip=row["offense_group"]).add_to(myMap)

# Display the map.
display(myMap)
#print(type(myMap))
```

```
dfformap=citycrime('New York')
#dfformap = df.sample(n=300)
displaymap(10)
```




```
#dfformap=citycrime('Kansas City')
dfformap = df.sample(n=300)
displaymap(4.3)
```

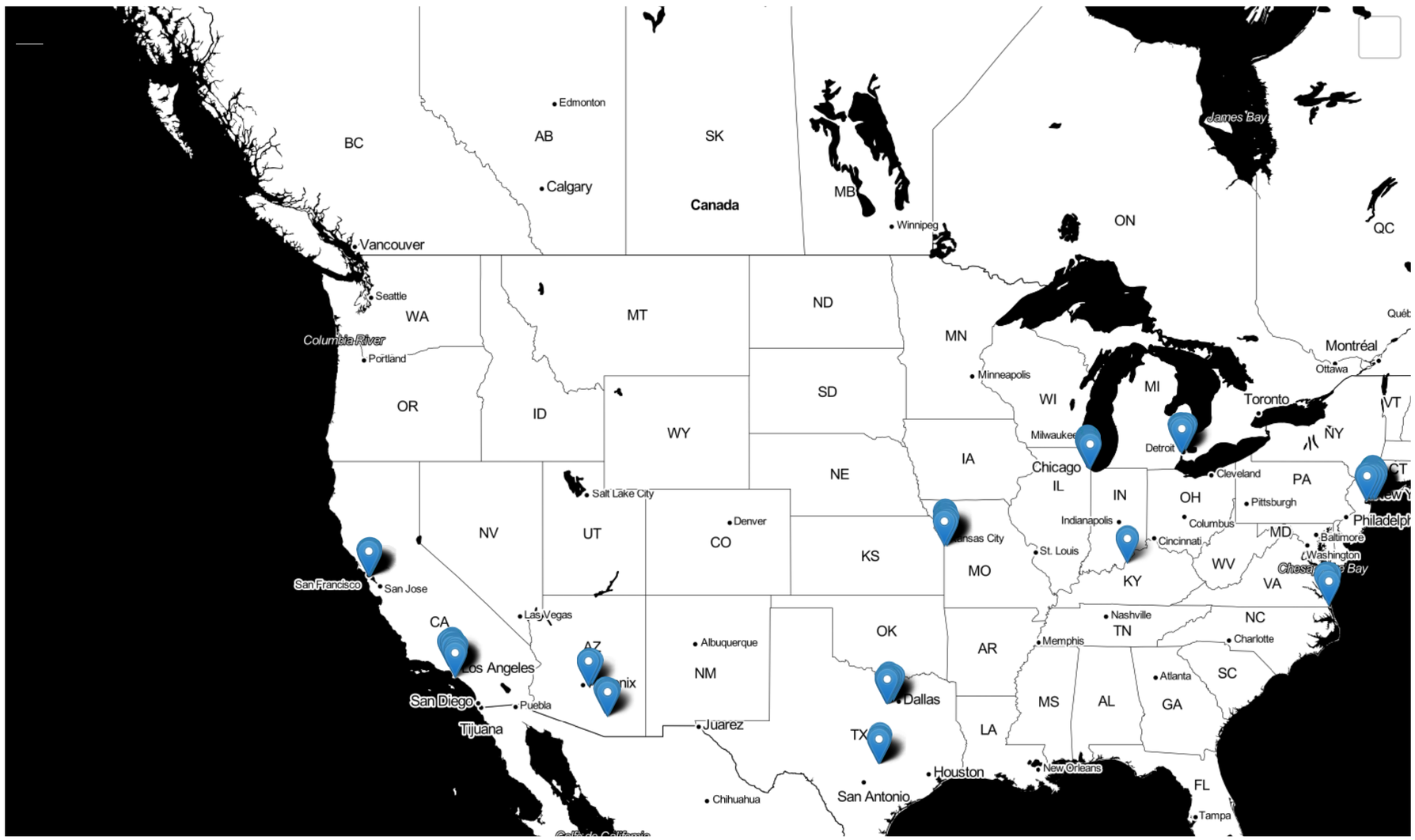


Chart visualization

Some Earth Engine functions produce tabular data that can be plotted by data visualization packages such as `matplotlib`. The following example demonstrates the display of tabular data from Earth Engine as a scatter plot. See Charting in Colaboratory for more information.

```
#some final aggregations by city with full dataset
df.groupby('city_name').count()
```

city_name	uid	offense_code	offense_type	offense_group	offense_against	date_single	longitude	latitude	location_type	location_category	census_block	date_start	date_end
Austin	82353	82353	82353	82353	82353	82353	82353	82353	81788	81788	82353	0	0
Boston	44165	44165	44165	44165	44165	44165	44165	44165	0	0	44165	0	0
Chicago	262258	262258	262258	262258	262258	262258	262258	262258	261601	261601	262258	0	0
Detroit	80618	80618	80618	80618	80618	80618	80618	80618	0	0	80618	0	0
Fort Worth	53819	53819	53819	53819	53819	53819	53819	53819	0	53818	53819	0	0
Kansas City	114889	114889	114889	114889	114889	114889	114889	114889	0	0	114889	114889	114889
Los Angeles	241220	241220	241220	241220	241220	241220	241220	241220	241059	241059	241220	0	0
Louisville	41008	41008	41008	41008	41008	41008	41008	41008	40974	40974	41008	0	0
Mesa	28947	28947	28947	28947	28947	28947	28947	28947	0	0	28947	0	0
New York	447766	447766	447766	447766	447766	447766	447766	447766	442802	442802	447766	447766	385601
San Francisco	107095	107095	107095	107095	107095	107095	107095	107095	0	0	107095	0	0
Tucson	78033	78033	78033	78033	78033	78033	78033	78033	0	0	78033	0	0
Virginia Beach	26618	26618	26618	26618	26618	26618	26618	26618	0	0	26618	0	0

```
dfbycity = df.loc[df['city_name'] == "New York"]
dfbycity

dfbycitychart = dfbycity.groupby('offense_type').count()
dfbycitychart = dfbycitychart.sort_values('uid', ascending = False)
dfbycitychart=dfbycitychart.reset_index()
dfbycitychart
px.bar(dfbycitychart, x="offense_type", y = "uid")
```