Midterm Part 1

November 16, 2020

1 Research Question

For our project, we will be researching crime data in Los Angeles and how that data is affected by various variables such as COVID-19, educational attainment, and household income. Due to the global pandemic, crime rates have fluctuated substantially due to the lockdown and the reopening of the county.

2 Data Sources

- Crime Data from 2020 to present, https://data.lacity.org/A-Safe-City/Crime-Data-from-2020-to-Present/2nrs-mtv8
- COVID-19 Data from 2020 to present, https://github.com/datadesk/california-coronavirus-data/blob/master/latimes-place-totals.csv
- Educational Attainment for LA County (2014-2018), Social Explorer
- Household Income for LA County (2018), Social Explorer
- White vs. Non-White Homeowners (2018), Social Explorer
- Mapping Inequality/ Home Owners Loan Corporation (HOLC) LA Redlining Map (1939), clsl.richmond.edu

3 Data Exploration and Analysis

Now we want to explore our data sources and provide an analysis of our datasets.

3.1 COVID-19 Rates in California

We will begin our data exploration by importing the current COVID-19 data from the LA times.

[1]:
```
import plotly.express as px
import pandas as pd
```

[2]:
```
latimes = pd.read_csv(
    "https://raw.githubusercontent.com/datadesk/california-coronavirus-data/
↪master/latimes-place-totals.csv")
```

[3]:
```
# Now we want to get some basic statistics from the dataset. How many rows and␣
↪columns?
latimes.shape
```

[3]:
```
(193110, 8)
```

[4]:
```
#What are the first 5 rows?
latimes.head()
```

[4]:
```
         date   county  fips           place  confirmed_cases note  \
0  2020-11-14  Alameda   1.0  94501: Alameda              468  NaN
1  2020-11-14  Alameda   1.0  94502: Alameda               69  NaN
2  2020-11-14  Alameda   1.0  94536: Fremont              715  NaN
3  2020-11-14  Alameda   1.0  94538: Fremont              718  NaN
4  2020-11-14  Alameda   1.0  94539: Fremont              223  NaN

            x          y
0 -122.274583  37.774606
1 -122.241149  37.736988
2 -121.987951  37.570977
3 -121.977924  37.499148
4 -121.912764  37.526588
```

[5]:
```
# dataframe info?
latimes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193110 entries, 0 to 193109
Data columns (total 8 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   date             193110 non-null  object
 1   county           193110 non-null  object
 2   fips             191334 non-null  float64
 3   place            193110 non-null  object
 4   confirmed_cases  193110 non-null  int64
 5   note             6828 non-null    object
 6   x                190846 non-null  float64
 7   y                190846 non-null  float64
dtypes: float64(3), int64(1), object(4)
memory usage: 11.8+ MB
```

[6]:
```
# Next, we want to clean up the data. This includes empty coordinates, empty␣
 ↪confirmed cases, and incorrect coordinates (Note: positive longitudes do not␣
 ↪exist in California)
# We do this by using the .query() method that allows us to query and filter␣
 ↪the dataset using SQL syntax.
latimes.query("confirmed_cases == 'NaN'")
```

[6]:
```
Empty DataFrame
Columns: [date, county, fips, place, confirmed_cases, note, x, y]
```

```
Index: []
```

[7]:
```
# NaN values for 'x'?
latimes.query("x == 'NaN'")
```

[7]:
```
              date        county  fips       place  confirmed_cases note   x  \
62      2020-11-14  Contra Costa  13.0  Kensington               14  NaN NaN
68      2020-11-14  Contra Costa  13.0       Other              206  NaN NaN
84      2020-11-14         Kings  31.0       Other               84  NaN NaN
85      2020-11-14         Kings  31.0     Prisons             4478  NaN NaN
517     2020-11-14     Riverside  65.0       Jails              489  NaN NaN
…              …           …    …   …         …                …   …   ..
190888  2020-03-28        Orange  59.0       Other               39  NaN NaN
191259  2020-03-27        Orange  59.0       Other               27  NaN NaN
192884  2020-03-18   Los Angeles  37.0       Other               62  NaN NaN
192961  2020-03-17   Los Angeles  37.0       Other               27  NaN NaN
193026  2020-03-16   Los Angeles  37.0       Other               11  NaN NaN

          y
62      NaN
68      NaN
84      NaN
85      NaN
517     NaN
…      ..
190888  NaN
191259  NaN
192884  NaN
192961  NaN
193026  NaN

[2264 rows x 8 columns]
```

[8]:
```
#NaN values for 'y'?
latimes.query("y == 'NaN'")
```

[8]:
```
              date        county  fips       place  confirmed_cases note   x  \
62      2020-11-14  Contra Costa  13.0  Kensington               14  NaN NaN
68      2020-11-14  Contra Costa  13.0       Other              206  NaN NaN
84      2020-11-14         Kings  31.0       Other               84  NaN NaN
85      2020-11-14         Kings  31.0     Prisons             4478  NaN NaN
517     2020-11-14     Riverside  65.0       Jails              489  NaN NaN
…              …           …    …   …         …                …   …   ..
190888  2020-03-28        Orange  59.0       Other               39  NaN NaN
191259  2020-03-27        Orange  59.0       Other               27  NaN NaN
192884  2020-03-18   Los Angeles  37.0       Other               62  NaN NaN
192961  2020-03-17   Los Angeles  37.0       Other               27  NaN NaN
```

```
193026  2020-03-16   Los Angeles  37.0       Other                11  NaN NaN

          y
62      NaN
68      NaN
84      NaN
85      NaN
517     NaN
…       ..
190888  NaN
191259  NaN
192884  NaN
192961  NaN
193026  NaN

[2264 rows x 8 columns]
```

[9]:
```
# Positive longitude coordinates?
latimes.query("x > 0")
```

[9]:
```
              date     county  fips     place  confirmed_cases note  \
1599    2020-11-13  San Mateo  81.0  Pacifica              262  NaN
7893    2020-11-06  San Mateo  81.0  Pacifica              245  NaN
15437   2020-10-29  San Mateo  81.0  Pacifica              237  NaN
20825   2020-10-23  San Mateo  81.0  Pacifica              228  NaN
27182   2020-10-16  San Mateo  81.0  Pacifica              223  NaN
33451   2020-10-09  San Mateo  81.0  Pacifica              219  NaN
39760   2020-10-02  San Mateo  81.0  Pacifica              215  NaN
46208   2020-09-25  San Mateo  81.0  Pacifica              212  NaN
52577   2020-09-18  San Mateo  81.0  Pacifica              206  NaN
59082   2020-09-11  San Mateo  81.0  Pacifica              199  NaN
65192   2020-09-04  San Mateo  81.0  Pacifica              190  NaN
70794   2020-08-29  San Mateo  81.0  Pacifica              176  NaN
76664   2020-08-22  San Mateo  81.0  Pacifica              158  NaN
84065   2020-08-14  San Mateo  81.0  Pacifica              146  NaN
90168   2020-08-07  San Mateo  81.0  Pacifica              126  NaN
96123   2020-07-31  San Mateo  81.0  Pacifica              109  NaN
102239  2020-07-24  San Mateo  81.0  Pacifica              100  NaN
109321  2020-07-16  San Mateo  81.0  Pacifica               93  NaN
114326  2020-07-10  San Mateo  81.0  Pacifica               81  NaN
119186  2020-07-03  San Mateo  81.0  Pacifica               75  NaN
125869  2020-06-25  San Mateo  81.0  Pacifica               59  NaN
130837  2020-06-19  San Mateo  81.0  Pacifica               51  NaN
136554  2020-06-12  San Mateo  81.0  Pacifica               47  NaN
142307  2020-06-05  San Mateo  81.0  Pacifica               45  NaN
144015  2020-06-03  San Mateo  81.0  Pacifica               43  NaN
146410  2020-05-31  San Mateo  81.0  Pacifica               43  NaN
```

```
147952  2020-05-29  San Mateo  81.0  Pacifica              43  NaN
149681  2020-05-27  San Mateo  81.0  Pacifica              39  NaN
153274  2020-05-22  San Mateo  81.0  Pacifica              39  NaN
154142  2020-05-21  San Mateo  81.0  Pacifica              39  NaN
159561  2020-05-14  San Mateo  81.0  Pacifica              38  NaN
160424  2020-05-13  San Mateo  81.0  Pacifica              38  NaN
162088  2020-05-11  San Mateo  81.0  Pacifica              38  NaN
164464  2020-05-08  San Mateo  81.0  Pacifica              38  NaN

                x          y
1599    122.480689  37.610177
7893    122.480689  37.610177
15437   122.480689  37.610177
20825   122.480689  37.610177
27182   122.480689  37.610177
33451   122.480689  37.610177
39760   122.480689  37.610177
46208   122.480689  37.610177
52577   122.480689  37.610177
59082   122.480689  37.610177
65192   122.480689  37.610177
70794   122.480689  37.610177
76664   122.480689  37.610177
84065   122.480689  37.610177
90168   122.480689  37.610177
96123   122.480689  37.610177
102239  122.480689  37.610177
109321  122.480689  37.610177
114326  122.480689  37.610177
119186  122.480689  37.610177
125869  122.480689  37.610177
130837  122.480689  37.610177
136554  122.480689  37.610177
142307  122.480689  37.610177
144015  122.480689  37.610177
146410  122.480689  37.610177
147952  122.480689  37.610177
149681  122.480689  37.610177
153274  122.480689  37.610177
154142  122.480689  37.610177
159561  122.480689  37.610177
160424  122.480689  37.610177
162088  122.480689  37.610177
164464  122.480689  37.610177
```

[10]:
```python
# Do we have any null dates?
latimes.query("date.isnull()", engine='python')
```

[10]:
```
Empty DataFrame
Columns: [date, county, fips, place, confirmed_cases, note, x, y]
Index: []
```

[11]:
```
# Now we will combine our arguments and clean the data:
latimes = latimes.query("confirmed_cases != 'NaN' & x < 0 & x != 'NaN' & date.
 ↪notnull()", engine='python')
latimes.head()
```

[11]:

	date	county	fips	place	confirmed_cases	note	\
0	2020-11-14	Alameda	1.0	94501: Alameda	468	NaN	
1	2020-11-14	Alameda	1.0	94502: Alameda	69	NaN	
2	2020-11-14	Alameda	1.0	94536: Fremont	715	NaN	
3	2020-11-14	Alameda	1.0	94538: Fremont	718	NaN	
4	2020-11-14	Alameda	1.0	94539: Fremont	223	NaN	

	x	y
0	-122.274583	37.774606
1	-122.241149	37.736988
2	-121.987951	37.570977
3	-121.977924	37.499148
4	-121.912764	37.526588

[12]:
```
# How many records do we have now?
latimes.shape
# Less columns than before
```

[12]:
```
(190812, 8)
```

[13]:
```
# Now we want to look at more statistics in our dataset. Let's look at␣
 ↪confirmed cases.
latimes.confirmed_cases.describe()
```

[13]:
```
count    190812.000000
mean        442.665985
std        1098.762492
min           1.000000
25%          18.000000
50%          88.000000
75%         392.000000
max       27311.000000
Name: confirmed_cases, dtype: float64
```

[14]:
```
# Let's see which counties in California have the most confirmed cases.
latimes.groupby("county").confirmed_cases.describe().sort_values(by=["max"],␣
 ↪ascending=False)
```

[14]:

county	count	mean	std	min	25%	50%	\
San Diego	8642.0	573.523374	2234.294469	1.0	11.00	47.5	
Fresno	2473.0	437.694298	1756.307493	1.0	17.00	56.0	
Santa Clara	3036.0	746.642292	2427.580461	1.0	35.00	117.0	
Sacramento	1329.0	1194.107600	2908.513814	1.0	49.00	246.0	
Los Angeles	73937.0	476.442160	930.242849	1.0	22.00	111.0	
Orange	9166.0	678.481671	1542.729321	1.0	44.00	172.0	
San Bernardino	10999.0	566.338122	1359.518986	1.0	7.00	39.0	
San Joaquin	707.0	1073.121641	2245.680873	1.0	53.00	171.0	
Riverside	10657.0	442.500047	1022.028026	1.0	21.00	93.0	
Stanislaus	1466.0	537.008868	1158.550871	1.0	16.00	122.0	
Monterey	784.0	1382.892857	1701.334298	8.0	178.25	735.0	
Santa Barbara	1895.0	411.397361	771.196473	1.0	27.00	135.0	
Kern	7471.0	454.190871	799.470624	1.0	6.00	55.0	
Sonoma	1207.0	264.661972	617.675528	1.0	17.00	50.0	
Placer	552.0	638.402174	1023.312509	11.0	72.00	188.5	
Imperial	2305.0	528.576573	926.639541	1.0	13.00	123.0	
Contra Costa	5933.0	317.030339	606.048301	1.0	27.00	82.0	
Merced	1327.0	475.657121	724.314229	6.0	40.00	211.0	
Long Beach	1776.0	666.904842	681.602606	11.0	191.00	444.5	
Tulare	2935.0	476.795571	694.648820	1.0	40.00	173.0	
Solano	1045.0	502.331100	696.693481	1.0	36.00	172.0	
Ventura	4042.0	256.779565	395.698354	1.0	24.00	89.5	
Alameda	11448.0	225.487945	366.321257	1.0	20.00	80.0	
Madera	308.0	886.448052	687.410943	58.0	318.75	610.0	
Kings	971.0	444.768280	537.139362	1.0	65.00	217.0	
Butte	548.0	336.237226	560.322477	3.0	26.00	70.0	
Marin	3879.0	105.949729	318.441575	1.0	1.00	15.0	
San Mateo	1083.0	172.687904	370.865721	1.0	1.00	25.0	
San Francisco	5877.0	242.657819	359.702356	1.0	38.00	113.0	
Santa Cruz	1059.0	186.904627	347.830931	5.0	25.00	58.0	
Sutter	104.0	558.846154	496.414460	68.0	148.00	305.0	
Yolo	843.0	331.823250	399.559518	4.0	51.00	125.0	
Napa	2207.0	60.373811	180.402387	1.0	2.00	7.0	
Shasta	60.0	303.300000	339.550765	35.0	105.00	187.5	
San Luis Obispo	2712.0	108.513274	180.000192	5.0	11.00	34.0	
Mendocino	699.0	122.383405	225.347173	1.0	11.00	39.0	
Yuba	175.0	207.360000	166.044522	22.0	63.00	158.0	
El Dorado	1164.0	80.721649	127.850877	1.0	7.00	25.0	
Humboldt	183.0	105.644809	110.180669	3.0	22.00	75.0	
Nevada	1446.0	28.482711	49.351413	1.0	1.00	1.0	
Lake	129.0	50.170543	70.424841	1.0	4.00	28.0	
Inyo	174.0	67.034483	71.469295	1.0	2.00	28.0	
Mono	324.0	45.404321	57.558048	1.0	4.00	14.0	
Calaveras	260.0	38.311538	41.963294	1.0	7.75	28.0	
Del Norte	126.0	48.079365	45.383980	1.0	9.00	32.0	

Siskiyou	44.0	67.340909	33.699040	17.0	40.00	67.0
Amador	967.0	23.589452	29.556242	1.0	4.00	14.0
Plumas	279.0	11.473118	10.023632	1.0	3.00	10.0
Trinity	59.0	9.254237	10.887502	1.0	1.00	4.0

county	75%	max
San Diego	315.00	27311.0
Fresno	255.00	18956.0
Santa Clara	431.50	18531.0
Sacramento	751.00	17102.0
Los Angeles	502.00	14200.0
Orange	617.00	12329.0
San Bernardino	336.00	11482.0
San Joaquin	1165.00	10826.0
Riverside	344.00	10772.0
Stanislaus	378.00	7090.0
Monterey	1911.50	6507.0
Santa Barbara	369.00	5226.0
Kern	557.00	4902.0
Sonoma	253.00	4853.0
Placer	423.50	4297.0
Imperial	466.00	4289.0
Contra Costa	281.00	3895.0
Merced	613.00	3786.0
Long Beach	931.50	3500.0
Tulare	565.00	3223.0
Solano	586.00	2804.0
Ventura	356.00	2709.0
Alameda	267.00	2437.0
Madera	1376.00	2384.0
Kings	607.50	2377.0
Butte	380.25	2298.0
Marin	54.00	2236.0
San Mateo	131.00	2163.0
San Francisco	237.00	2015.0
Santa Cruz	128.00	1890.0
Sutter	947.75	1643.0
Yolo	468.50	1576.0
Napa	38.50	1490.0
Shasta	294.00	1315.0
San Luis Obispo	125.00	1223.0
Mendocino	82.50	977.0
Yuba	339.00	646.0
El Dorado	93.00	627.0
Humboldt	127.00	411.0
Nevada	31.00	279.0

```
Lake              39.00    236.0
Inyo             154.00    207.0
Mono              71.25    186.0
Calaveras         46.50    176.0
Del Norte         88.75    149.0
Siskiyou          94.00    139.0
Amador            29.00    139.0
Plumas            16.00     51.0
Trinity           15.50     40.0
```

[15]:
```
# Since our research question is focused on Los Angeles County, let's look at which cities in LA County have the highest confirmed cases.
latimes_LA = latimes.query("county=='Los Angeles'")
```

[16]:
```
latimes_LA.groupby("place").confirmed_cases.describe().sort_values(by=["max"],ascending=False).head(50)
```

[16]:

place	count	mean	std	min	\
Long Beach	240.0	6323.141667	4993.659325	5.0	
East Los Angeles	238.0	3506.025210	2684.810990	1.0	
Pomona	232.0	2943.056034	2426.656164	1.0	
Palmdale	234.0	2327.346154	1828.881392	1.0	
South Gate	235.0	2501.829787	1895.593048	1.0	
El Monte	227.0	2490.127753	1872.596417	1.0	
North Hollywood	240.0	2101.541667	1669.723565	1.0	
Boyle Heights	241.0	2376.278008	1830.011127	5.0	
Glendale	241.0	2120.165975	1539.791633	2.0	
Lancaster	238.0	1893.004202	1524.231229	1.0	
Downey	233.0	2348.553648	1736.735758	1.0	
Santa Clarita	240.0	1940.950000	1506.136810	2.0	
Compton	234.0	2239.341880	1707.960321	1.0	
Pacoima	231.0	2010.450216	1463.987332	1.0	
Sylmar	239.0	1890.087866	1379.801894	1.0	
Norwalk	233.0	1850.429185	1412.091798	1.0	
Unincorporated - Florence-Firestone	225.0	2027.720000	1417.270417	19.0	
Van Nuys	238.0	1716.075630	1259.557485	1.0	
Lynwood	240.0	1828.591667	1386.202431	1.0	
Panorama City	232.0	1733.237069	1201.820239	1.0	
Baldwin Park	226.0	1654.840708	1285.759335	2.0	
West Covina	236.0	1581.805085	1255.961578	1.0	
Inglewood	241.0	1599.481328	1187.875997	1.0	
Vernon Central	236.0	1722.351695	1240.140136	1.0	
Huntington Park	229.0	1672.371179	1205.807689	2.0	
Pasadena	241.0	1549.095436	988.904256	2.0	
Reseda	238.0	1293.012605	920.563611	1.0	
Pico Rivera	233.0	1406.103004	1005.855909	1.0	

Whittier	238.0	1229.172269	981.177446	2.0
Bellflower	235.0	1333.982979	1008.896444	1.0
Paramount	235.0	1333.957447	1022.741250	1.0
Montebello	232.0	1315.767241	967.294982	1.0
West Vernon	241.0	1299.763485	965.865121	1.0
Florence-Firestone	230.0	1336.421739	957.019281	3.0
Westlake	236.0	1467.364407	934.258105	1.0
Wholesale District	233.0	1459.755365	961.061763	1.0
Canoga Park	235.0	1128.914894	780.248904	1.0
Central	230.0	1266.982609	879.036938	2.0
North Hills	231.0	1089.038961	756.263822	1.0
Bell Gardens	232.0	1102.775862	838.176046	1.0
Melrose	241.0	1144.543568	741.241098	2.0
Hawthorne	238.0	1098.462185	800.637571	1.0
South Park	234.0	1169.064103	843.152316	1.0
Sun Valley	236.0	913.792373	721.231072	1.0
South Whittier	237.0	951.345992	804.119646	1.0
Watts	232.0	1083.831897	812.883155	1.0
San Pedro	241.0	1245.780083	730.242458	1.0
Castaic	238.0	1337.420168	863.980309	1.0
Carson	241.0	1034.767635	744.236222	1.0
Vermont Vista	236.0	1075.898305	794.292420	1.0

	25%	50%	75%	max
place				
Long Beach	1199.75	6188.0	11217.00	14200.0
East Los Angeles	559.50	3661.0	6045.00	7700.0
Pomona	335.50	2880.5	5275.00	6963.0
Palmdale	588.50	2010.0	3927.50	6068.0
South Gate	386.00	2797.0	4266.50	5385.0
El Monte	441.00	2730.0	4252.00	5322.0
North Hollywood	548.75	1804.0	3616.25	5310.0
Boyle Heights	360.00	2538.0	4130.00	5211.0
Glendale	802.00	1897.0	3409.00	5192.0
Lancaster	471.75	1671.5	3169.75	5128.0
Downey	483.00	2685.0	3937.00	5082.0
Santa Clarita	569.25	1752.0	3217.75	5073.0
Compton	395.00	2357.5	3914.50	4845.0
Pacoima	601.00	1926.0	3334.50	4725.0
Sylmar	587.00	1816.0	3127.50	4421.0
Norwalk	327.00	2016.0	3137.00	4186.0
Unincorporated - Florence-Firestone	553.00	2226.0	3354.00	4127.0
Van Nuys	545.25	1582.0	2832.00	4055.0
Lynwood	317.25	1960.0	3140.75	3953.0
Panorama City	615.00	1653.5	2794.75	3878.0
Baldwin Park	288.00	1771.0	2874.00	3679.0
West Covina	217.75	1747.5	2758.50	3578.0

Inglewood	453.00	1526.0	2766.00	3512.0
Vernon Central	404.25	1870.5	2907.25	3508.0
Huntington Park	367.00	1815.0	2817.00	3472.0
Pasadena	662.00	1643.0	2479.00	3073.0
Reseda	394.50	1290.5	2073.25	3050.0
Pico Rivera	378.00	1547.0	2343.00	3031.0
Whittier	214.00	1264.5	2154.75	2933.0
Bellflower	260.00	1439.0	2299.00	2902.0
Paramount	210.00	1467.0	2319.00	2887.0
Montebello	306.25	1423.5	2242.50	2884.0
West Vernon	305.00	1317.0	2257.00	2800.0
Florence-Firestone	306.75	1483.5	2233.25	2754.0
Westlake	523.00	1673.0	2330.25	2753.0
Wholesale District	533.00	1775.0	2330.00	2743.0
Canoga Park	431.00	1100.0	1802.00	2559.0
Central	319.00	1454.5	2117.25	2508.0
North Hills	371.00	1084.0	1763.50	2449.0
Bell Gardens	201.50	1178.5	1863.25	2431.0
Melrose	443.00	1168.0	1829.00	2397.0
Hawthorne	306.25	1108.5	1887.50	2375.0
South Park	244.25	1285.5	1998.75	2374.0
Sun Valley	239.75	813.0	1546.25	2370.0
South Whittier	105.00	978.0	1696.00	2306.0
Watts	211.75	1142.0	1889.75	2277.0
San Pedro	772.00	1394.0	1901.00	2262.0
Castaic	276.00	1834.5	1923.00	2249.0
Carson	327.00	971.0	1776.00	2248.0
Vermont Vista	236.75	1141.5	1872.50	2230.0

[17]:
```
# Let's create a bar chart representing the confirmed cases in LA County␣
↪overtime.
LACounty = latimes.query("county == ['Los Angeles']")
px.bar(LACounty,
       x='date',
       y='confirmed_cases')
```

[18]:
```
# Let's be more specific. Let's create a bar chart of the top three cities in␣
↪LA County with the highest confirmed cases: Long Beach, East Los Angeles,␣
↪and Pomona.
TopLA = latimes.query("place == ['Long Beach','East Los Angeles','Pomona']")
px.bar(TopLA,
       x='date',
       y='confirmed_cases',
       color = 'place')
```

Now that we've looked at the top three cities with the highest confirmed COVID-19 cases, let's represent our dataset in a different visualization format. Let's create an animated scatter plot to

represent the change overtime of confirmed cases in cities across LA County.

[19]:
```
# What is the mean of  confirmed cases in LA County?
latimes_LA_mean = latimes_LA.confirmed_cases.mean()
latimes_LA_mean
```

[19]:
```
476.44216021748247
```

[20]:
```
px.scatter(latimes_LA,
           x='x',
           y='y',
           color='confirmed_cases',
           size='confirmed_cases',
           size_max=40,
           hover_name='place',
           animation_frame='date', # this creates a frame by frame animation by␣
 ↪day
           color_continuous_scale = 'RdYlGn_r',
           range_color = (0,latimes_LA_mean*2))
```

[46]:
```
# Now, let's put this information on a map.
fig = px.scatter_geo(latimes_LA,
           lon='x',
           lat='y',
           color='confirmed_cases',
           size='confirmed_cases',
           size_max=40,
           hover_name='place',
           scope='usa',
           animation_frame='date',
           color_continuous_scale = 'RdYlGn_r',
           range_color = (0,latimes_LA_mean*2))

fig.update_geos(fitbounds="locations")
```

An issue we had with these two animated scatterplots was that the animation began from the present to March, instead of vice versa. This is something we will have to address in the future.

3.2 Crime Rates in the City of Los Angeles

Let's look at crime rates in LA County from 2020 to present. We will begin by importing the data.

[22]:
```
import pandas as pd
import plotly.express as px
from sodapy import Socrata
```

3.2.1 Creating a Socrata Client

Next, we acquire the data using the socrata API. - https://dev.socrata.com/foundry/data.lacity.org/2nrs-mtv8

[23]:
```
# connect to the data portal
client = Socrata("data.lacity.org", None)

# First 2000 results, returned as JSON from API / converted to Python list of
# dictionaries by sodapy.
results = client.get("2nrs-mtv8", limit=2000)

# Convert to pandas DataFrame
df = pd.DataFrame.from_records(results)

# print it with .sample, which gives you random rows
df.sample(2)
```

```
WARNING:root:Requests made without an app_token will be subject to strict
throttling limits.
```

[23]:

```
          dr_no                 date_rptd                 date_occ time_occ  \
1756  200204784  2020-01-16T00:00:00.000  2020-01-15T00:00:00.000     1500
829   200105671  2020-01-24T00:00:00.000  2020-01-24T00:00:00.000     0001

     area area_name rpt_dist_no part_1_2 crm_cd  \
1756   02   Rampart        0235        1    420
829    01   Central        0138        1    821

                                          crm_cd_desc  … weapon_used_cd  \
1756    THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)  …            NaN
829   SODOMY/SEXUAL CONTACT B/W PENIS OF ONE PERS TO…  …            400

                                     weapon_desc status  status_desc  \
1756                                         NaN     IC  Invest Cont
829   STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)     IC  Invest Cont

     crm_cd_1                                location      lat        lon  \
1756      420  100    ROSELAKE                     AV  34.0675  -118.2722
829       821  300 S  ALAMEDA                     ST  34.0468  -118.2415

     crm_cd_2 cross_street
1756      NaN          NaN
829       NaN          NaN

[2 rows x 26 columns]
```

[24]:
```
# Now, we want to add a "where" statement to look at the data from March 1,␣
↪2020 to April 30, 2020, limited to 30,000.
results = client.get("2nrs-mtv8",
                     limit = 30000,
                     where = "date_rptd between '2020-03-01T00:00:00' and␣
↪'2020-04-30T00:00:00'"
                    )
```

[25]:
```
# Convert to pandas DataFrame
df = pd.DataFrame.from_records(results)
```

3.2.2 Data Exploration and Analysis of Crime Data

[26]:
```
# how many rows and columns?
df.shape
```

[26]:
```
(30000, 28)
```

[27]:
```
# what fields and datatypes?
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   dr_no           30000 non-null  object
 1   date_rptd       30000 non-null  object
 2   date_occ        30000 non-null  object
 3   time_occ        30000 non-null  object
 4   area            30000 non-null  object
 5   area_name       30000 non-null  object
 6   rpt_dist_no     30000 non-null  object
 7   part_1_2        30000 non-null  object
 8   crm_cd          30000 non-null  object
 9   crm_cd_desc     30000 non-null  object
 10  mocodes         25994 non-null  object
 11  vict_age        30000 non-null  object
 12  vict_sex        26189 non-null  object
 13  vict_descent    26189 non-null  object
 14  premis_cd       29999 non-null  object
 15  premis_desc     29987 non-null  object
 16  weapon_used_cd  10827 non-null  object
 17  weapon_desc     10827 non-null  object
 18  status          30000 non-null  object
 19  status_desc     30000 non-null  object
 20  crm_cd_1        30000 non-null  object
```

```
 21  location        30000 non-null  object
 22  cross_street    5195 non-null   object
 23  lat             30000 non-null  object
 24  lon             30000 non-null  object
 25  crm_cd_2        2518 non-null   object
 26  crm_cd_3        71 non-null     object
 27  crm_cd_4        3 non-null      object
dtypes: object(28)
memory usage: 6.4+ MB
```

[28]:
```
# First 5 rows?
df.head()
```

[28]:

	dr_no	date_rptd	date_occ	time_occ	area	area_name	rpt_dist_no	part_1_2	crm_cd	crm_cd_desc	…	status	status_desc	crm_cd_1	location	cross_street	lat	lon	crm_cd_2	crm_cd_3	crm_cd_4
0	200607206	2020-03-01T00:00:00.000	2020-02-29T00:00:00.000	1900	06	Hollywood	0645	1	815	SEXUAL PENETRATION W/FOREIGN OBJECT	…	IC	Invest Cont	815	HOLLYWOOD	HIGHLAND	34.1016	-118.3387	NaN	NaN	NaN
1	201407485	2020-03-01T00:00:00.000	2020-03-01T00:00:00.000	1350	14	Pacific	1406	1	761	BRANDISH WEAPON	…	AO	Adult Other	624	3800 KEYSTONE	AV	NaN	34.0189	-118.4056	761	998	NaN
2	202006971	2020-03-01T00:00:00.000	2020-02-02T00:00:00.000	1000	20	Olympic	2062	2	860	BATTERY WITH SEXUAL CONTACT	…	AO	Adult Other	860	3200 W PICO	BL	NaN	34.0506	-118.3127	NaN	NaN	NaN
3	200307647	2020-03-01T00:00:00.000	2020-03-01T00:00:00.000	0935	03	Southwest	0317	2	740	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA…	…	AO	Adult Other	740	2400 S CATALINA	ST	NaN	34.0337	-118.2942	NaN	NaN	NaN
4	201807506	2020-03-01T00:00:00.000	2020-03-01T00:00:00.000	1450	18	Southeast	1846	2	626	INTIMATE PARTNER - SIMPLE ASSAULT	…	AA	Adult Arrest	626	ZAMORA	114TH	33.931	-118.2511	NaN	NaN	NaN

```
[5 rows x 28 columns]
```

[30]:
```
px.bar(df,
       x='date_rptd',
       title='Crime Rates in Los Angeles, March to April 2020'
      )
```

[29]:
```
# Let's clean up the labels.
px.bar(df,
       x='date_rptd',
       title='Crime Rates in Los Angeles, March to April 2020',
       labels={'date_rptd':'Date of Crimes','count':'Number of Crimes'}
      )
```

[31]:
```
# Let's look at the distinct value of charges
df.crm_cd_desc.unique()
```

[31]:
```
array(['SEXUAL PENETRATION W/FOREIGN OBJECT', 'BRANDISH WEAPON',
       'BATTERY WITH SEXUAL CONTACT',
       'VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)',
       'INTIMATE PARTNER - SIMPLE ASSAULT', 'VEHICLE - STOLEN',
       'INTIMATE PARTNER - AGGRAVATED ASSAULT', 'CRIMINAL HOMICIDE',
       'THEFT PLAIN - PETTY ($950 & UNDER)', 'VIOLATION OF COURT ORDER',
       'BURGLARY FROM VEHICLE',
       'VIOLATION OF TEMPORARY RESTRAINING ORDER',
       'VIOLATION OF RESTRAINING ORDER',
       'ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT',
       'ATTEMPTED ROBBERY', 'BATTERY - SIMPLE ASSAULT',
       'CRIMINAL THREATS - NO WEAPON DISPLAYED', 'ROBBERY',
       'LETTERS, LEWD  -  TELEPHONE CALLS, LEWD',
       'SHOPLIFTING-GRAND THEFT ($950.01 & OVER)',
       'CHILD NEGLECT (SEE 300 W.I.C.)',
       'SHOPLIFTING - PETTY THEFT ($950 & UNDER)',
       'OTHER MISCELLANEOUS CRIME',
       'VANDALISM - MISDEAMEANOR ($399 OR UNDER)', 'RESISTING ARREST',
       'RAPE, FORCIBLE', 'CHILD ABUSE (PHYSICAL) - AGGRAVATED ASSAULT',
       'THEFT OF IDENTITY', 'THROWING OBJECT AT MOVING VEHICLE',
       'DOCUMENT FORGERY / STOLEN FELONY', 'TRESPASSING', 'OTHER ASSAULT',
       'BURGLARY', 'BATTERY POLICE (SIMPLE)',
       'THEFT FROM MOTOR VEHICLE - GRAND ($400 AND OVER)',
       'THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)',
       'THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LIVESTK,PROD',
       'THEFT, PERSON', 'BURGLARY, ATTEMPTED', 'BIKE - STOLEN',
       'ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER', 'PICKPOCKET',
       'FAILURE TO YIELD', 'BUNCO, GRAND THEFT', 'BUNCO, PETTY THEFT',
       'UNAUTHORIZED COMPUTER ACCESS', 'INDECENT EXPOSURE',
```

```
'VEHICLE - ATTEMPT STOLEN', 'THEFT FROM PERSON - ATTEMPT',
'SODOMY/SEXUAL CONTACT B/W PENIS OF ONE PERS TO ANUS OTH',
'CHILD ABUSE (PHYSICAL) - SIMPLE ASSAULT',
'SEX OFFENDER REGISTRANT OUT OF COMPLIANCE', 'CONTEMPT OF COURT',
'DEFRAUDING INNKEEPER/THEFT OF SERVICES, OVER $400',
'DEFRAUDING INNKEEPER/THEFT OF SERVICES, $400 & UNDER',
'ILLEGAL DUMPING', 'EMBEZZLEMENT, PETTY THEFT ($950 & UNDER)',
'PURSE SNATCHING', 'THEFT FROM MOTOR VEHICLE - ATTEMPT', 'ARSON',
'CHILD ANNOYING (17YRS & UNDER)', 'EXTORTION',
'CREDIT CARDS, FRAUD USE ($950 & UNDER',
'THREATENING PHONE CALLS/LETTERS', 'CHILD STEALING',
'EMBEZZLEMENT, GRAND THEFT ($950.01 & OVER)',
'SHOTS FIRED AT INHABITED DWELLING',
'SEX,UNLAWFUL(INC MUTUAL CONSENT, PENETRATION W/ FRGN OBJ',
'HUMAN TRAFFICKING - COMMERCIAL SEX ACTS',
'DISCHARGE FIREARMS/SHOTS FIRED',
'BURGLARY FROM VEHICLE, ATTEMPTED', 'RAPE, ATTEMPTED',
'BATTERY ON A FIREFIGHTER', 'KIDNAPPING', 'STALKING',
'ORAL COPULATION', 'CHILD PORNOGRAPHY', 'BUNCO, ATTEMPT',
'DISTURBING THE PEACE',
'CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER)',
'PEEPING TOM', 'CREDIT CARDS, FRAUD USE ($950.01 & OVER)',
'PANDERING', 'FALSE IMPRISONMENT', 'PROWLER', 'BOMB SCARE',
'KIDNAPPING - GRAND ATTEMPT', 'CONTRIBUTING',
'VEHICLE - MOTORIZED SCOOTERS, BICYCLES, AND WHEELCHAIRS',
'SHOPLIFTING - ATTEMPT', 'THEFT PLAIN - ATTEMPT', 'COUNTERFEIT',
'LEWD CONDUCT', 'PIMPING',
'HUMAN TRAFFICKING - INVOLUNTARY SERVITUDE',
'DRIVING WITHOUT OWNER CONSENT (DWOC)',
'TILL TAP - PETTY ($950 & UNDER)', 'CRUELTY TO ANIMALS',
'THEFT, COIN MACHINE - PETTY ($950 & UNDER)',
'DISHONEST EMPLOYEE - GRAND THEFT',
'LEWD/LASCIVIOUS ACTS WITH CHILD', 'CONSPIRACY',
'DISHONEST EMPLOYEE - PETTY THEFT', 'DRUGS, TO A MINOR',
'FIREARMS RESTRAINING ORDER (FIREARMS RO)',
'REPLICA FIREARMS(SALE,DISPLAY,MANUFACTURE OR DISTRIBUTE)',
'SHOTS FIRED AT MOVING VEHICLE, TRAIN OR AIRCRAFT',
'RECKLESS DRIVING', 'FALSE POLICE REPORT',
'WEAPONS POSSESSION/BOMBING', 'BOAT - STOLEN',
'BIKE - ATTEMPTED STOLEN', 'DOCUMENT WORTHLESS ($200.01 & OVER)',
'GRAND THEFT / AUTO REPAIR'], dtype=object)
```

[32]:
```
# Let's look at  the top 25 distinct value of charges
crime_by_type = df.crm_cd_desc.value_counts().reset_index()
crime_by_type.head(25)
```

[32]:

	index	crm_cd_desc
0	VEHICLE - STOLEN	3146
1	BATTERY - SIMPLE ASSAULT	2488
2	BURGLARY FROM VEHICLE	2152
3	BURGLARY	1999
4	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA…	1952
5	INTIMATE PARTNER - SIMPLE ASSAULT	1731
6	THEFT PLAIN - PETTY ($950 & UNDER)	1669
7	THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)	1632
8	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	1606
9	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	1089
10	ROBBERY	1009
11	THEFT OF IDENTITY	983
12	THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LI…	840
13	CRIMINAL THREATS - NO WEAPON DISPLAYED	670
14	THEFT FROM MOTOR VEHICLE - GRAND ($400 AND OVER)	665
15	SHOPLIFTING - PETTY THEFT ($950 & UNDER)	571
16	INTIMATE PARTNER - AGGRAVATED ASSAULT	462
17	VIOLATION OF RESTRAINING ORDER	444
18	BRANDISH WEAPON	434
19	TRESPASSING	416
20	OTHER MISCELLANEOUS CRIME	284
21	BIKE - STOLEN	281
22	VIOLATION OF COURT ORDER	212
23	LETTERS, LEWD - TELEPHONE CALLS, LEWD	210
24	ATTEMPTED ROBBERY	203

[33]:
```
# Rename our columns
crime_by_type.columns=['crime','count']
crime_by_type.head(25)
```

[33]:

	crime	count
0	VEHICLE - STOLEN	3146
1	BATTERY - SIMPLE ASSAULT	2488
2	BURGLARY FROM VEHICLE	2152
3	BURGLARY	1999
4	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA…	1952
5	INTIMATE PARTNER - SIMPLE ASSAULT	1731
6	THEFT PLAIN - PETTY ($950 & UNDER)	1669
7	THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)	1632
8	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	1606
9	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	1089
10	ROBBERY	1009
11	THEFT OF IDENTITY	983
12	THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LI…	840
13	CRIMINAL THREATS - NO WEAPON DISPLAYED	670
14	THEFT FROM MOTOR VEHICLE - GRAND ($400 AND OVER)	665

```
15           SHOPLIFTING - PETTY THEFT ($950 & UNDER)    571
16              INTIMATE PARTNER - AGGRAVATED ASSAULT    462
17                    VIOLATION OF RESTRAINING ORDER    444
18                                   BRANDISH WEAPON    434
19                                      TRESPASSING    416
20                          OTHER MISCELLANEOUS CRIME    284
21                                       BIKE - STOLEN    281
22                           VIOLATION OF COURT ORDER    212
23            LETTERS, LEWD  -  TELEPHONE CALLS, LEWD    210
24                                ATTEMPTED ROBBERY    203
```

[34]:
```
px.bar(crime_by_type.head(25),
       x='crime',
       y='count',
       title='Crime Rates in Los Angeles, March to April 2020')
```

[35]:
```
# Let's see if creating a horizontal chart will help the overlapping text issue.
px.bar(crime_by_type.head(25).sort_values(by='count',ascending=True),
       y='crime',
       x='count',
       orientation= 'h',
       title='Crime Rates in Los Angeles, March to April 2020')
```

[36]:
```
# Now, let's subset our data and begin mapping the dataset.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   dr_no           30000 non-null  object
 1   date_rptd       30000 non-null  object
 2   date_occ        30000 non-null  object
 3   time_occ        30000 non-null  object
 4   area            30000 non-null  object
 5   area_name       30000 non-null  object
 6   rpt_dist_no     30000 non-null  object
 7   part_1_2        30000 non-null  object
 8   crm_cd          30000 non-null  object
 9   crm_cd_desc     30000 non-null  object
 10  mocodes         25994 non-null  object
 11  vict_age        30000 non-null  object
 12  vict_sex        26189 non-null  object
 13  vict_descent    26189 non-null  object
 14  premis_cd       29999 non-null  object
 15  premis_desc     29987 non-null  object
```

```
 16  weapon_used_cd  10827 non-null  object
 17  weapon_desc     10827 non-null  object
 18  status          30000 non-null  object
 19  status_desc     30000 non-null  object
 20  crm_cd_1        30000 non-null  object
 21  location        30000 non-null  object
 22  cross_street    5195 non-null   object
 23  lat             30000 non-null  object
 24  lon             30000 non-null  object
 25  crm_cd_2        2518 non-null   object
 26  crm_cd_3        71 non-null     object
 27  crm_cd_4        3 non-null      object
dtypes: object(28)
memory usage: 6.4+ MB
```

Let's eliminate the unnecessary fields and create a subset of the data with just the following fields:

- date_rptd
- crm_cd
- crm_cd_desc
- lat
- lon

[38]:
```
# subset the data
df_mini = df[['date_rptd','crm_cd','crm_cd_desc','lat','lon']].copy()
df_mini.head()
```

[38]:

	date_rptd	crm_cd	crm_cd_desc	lat	lon
0	2020-03-01T00:00:00.000	815	SEXUAL PENETRATION W/FOREIGN OBJECT	34.1016	-118.3387
1	2020-03-01T00:00:00.000	761	BRANDISH WEAPON	34.0189	-118.4056
2	2020-03-01T00:00:00.000	860	BATTERY WITH SEXUAL CONTACT	34.0506	-118.3127
3	2020-03-01T00:00:00.000	740	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA…	34.0337	-118.2942
4	2020-03-01T00:00:00.000	626	INTIMATE PARTNER - SIMPLE ASSAULT	33.931	-118.2511

[40]:
```
# Check the info for our subset data
df_mini.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
```

```
 0   date_rptd    30000 non-null  object
 1   crm_cd       30000 non-null  object
 2   crm_cd_desc  30000 non-null  object
 3   lat          30000 non-null  object
 4   lon          30000 non-null  object
dtypes: object(5)
memory usage: 1.1+ MB
```

[41]:
```
# Now we want to convert latitude and longitude to floats
df_mini['lat'] = df_mini['lat'].astype(float)
df_mini['lon'] = df_mini['lon'].astype(float)
df_mini.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   date_rptd    30000 non-null  object
 1   crm_cd       30000 non-null  object
 2   crm_cd_desc  30000 non-null  object
 3   lat          30000 non-null  float64
 4   lon          30000 non-null  float64
dtypes: float64(2), object(3)
memory usage: 1.1+ MB
```

[51]:
```
# Now, let's create a scatter plot.
px.scatter(df_mini,
           x='lon',
           y='lat'
          )
```

This scatter plot does not look correct. This will be a problem that we will have to correct in the upcoming future.

[52]:
```
# What if we try to map it with plotly?
fig = px.scatter_mapbox(df_mini,
                        lat='lat',
                        lon='lon',
                        mapbox_style="stamen-terrain")
fig.show()
```

[45]:
```
# Let's try color-coding the crimes and creating an animation.
fig = px.scatter_mapbox(df_mini,
                        lat="lat",
                        lon="lon",
                        color="crm_cd",
                        animation_frame = 'date_rptd',
```

```
                        )
fig.update_layout(mapbox_style="carto-darkmatter")

fig.show()
```

4 Group Contributions

1. Donna Heydar (Donna contributed to breaking down the educational attainment data in Los Angeles County. Both members discussed which datasets to use and discussed similarities between the two after breaking them down. Donna also contributed to the data exploration and analysis of Crime data in LA as well as COVID-19 data,)
2. Daniel Ruiz (Daniel contributed to breaking down the household income data in Los Angeles County. Both members discussed which datasets to use and discussed similarities between the two after breaking them down. Daniel also contributed to the data exploration and analysis of HOLC Redlining. He also contributed to the comparison between education income, and homeownership in Los Angeles.)