covid

May 19, 2020

0.1 Community Health Equity Lab: New York COVID19 Response Analysis

[1]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')

pd.options.display.max_columns = 300
pd.set_option('display.max_rows', 500)
```

0.2 Data

Data comes from the 2018 NY Census

The Definitive Healthcare Hospital Bed DB

Google Location Mobility Report (Feb 15 - Apr 11)

[2]:
```
#reading zip code data and joining borough name
ny = pd.read_csv('data/ny18_cleaned.csv')
ny.head()
```

[2]:

	GEOID_ZIP	ALAND10	AWATER10	tot_pop	white	per_white	black	per_black
0	10065	984654	0	28109	24285	86.4	619	2.2
1	10069	249050	0	5085	3155	62.0	148	2.9
2	10075	477137	0	21556	18396	85.3	677	3.1
3	10128	1206191	0	59256	47167	79.6	2182	3.7
4	10280	297253	38409	9384	7360	78.4	184	2.0

	native	per_native	asian	per_asian	nativehawaiian	per_hawaiian	other
0	37	0.1	2666	9.5	0	0.0	154
1	0	0.0	1558	30.6	0	0.0	0
2	225	1.0	1047	4.9	0	0.0	902
3	0	0.0	5844	9.9	0	0.0	1666
4	0	0.0	1474	15.7	0	0.0	77

	per_other	two_or_mor	per_two_or	male	female	under_5	per_under_5
0	0.5	348	1.2	12248	15861	1650	5.9
1	0.0	224	4.4	2354	2731	368	7.2

```
2        4.2         309         1.4  10096   11460     1041          4.8
3        2.8        2397         4.0  25338   33918     3204          5.4
4        0.8         289         3.1   4477    4907      711          7.6

   5_to_9  per_5_to_9  10_to_14  per_10_to_14  15_to_19  per_15_to_19  \
0    1000         3.6       916           3.3       356           1.3
1     416         8.2       274           5.4        95           1.9
2    1252         5.8       736           3.4       763           3.5
3    2138         3.6      1911           3.2      1691           2.9
4     628         6.7       454           4.8       192           2.0

   20_to_24  per_20_to_24  25_to_29  per_25_to_29  30_to_34  per_30_to_34  \
0      1785           6.4      2671           9.5      3176          11.3
1       235           4.6       427           8.4       594          11.7
2       778           3.6      1400           6.5      2022           9.4
3      2644           4.5      6825          11.5      7216          12.2
4       239           2.5       523           5.6      1374          14.6

   35_to_39  per_35_to_39  40_to_44  per_40_to_44  45_to_49  per_45_to_49  \
0      2558           9.1      1607           5.7      1570           5.6
1       639          12.6       508          10.0       374           7.4
2      1643           7.6      1148           5.3      1453           6.7
3      4349           7.3      3849           6.5      4103           6.9
4       945          10.1      1021          10.9       771           8.2

   50_to_54  per_50_to_54  55_to_59  per_55_to_59  60_to_64  per_60_to_64  \
0      1796           6.4      1460           5.2      1522           5.4
1       378           7.4       235           4.6       197           3.9
2      1244           5.8      1118           5.2      1131           5.2
3      3845           6.5      2989           5.0      3494           5.9
4       763           8.1       530           5.6       375           4.0

   65_to_69  per_65_to_69  70_to_74  per_70_to_74  75_to_79  per_75_to_79  \
0      1388           4.9      1486           5.3      1159           4.1
1        61           1.2        87           1.7       109           2.1
2      1561           7.2      1276           5.9      1169           5.4
3      3461           5.8      2656           4.5      1980           3.3
4       370           3.9       254           2.7        79           0.8

   80_to_84  per_80_to_89  80_and_over  per_80_and_over  disabled  \
0       954           3.4         1055              3.8      1894
1         0           0.0           88              1.7       182
2       570           2.6         1251              5.8      1689
3      1557           2.6         1344              2.3      4825
4       115           1.2           40              0.4       326

   per_disabled  unemployment_rate  tot_households_snap  households_snap  \
```

```
0           6.8              70.2               14844             414
1           3.6              74.7                2552              73
2           8.0              68.1               11035             217
3           8.1              72.2               30691            1506
4           3.5              83.7                4560              97

   per_households_snap  tot_pop_mobility  same_house  moved_within_1yr  \
0                  2.8             27635       22612              2247
1                  2.9              5025        3790               859
2                  2.0             21381       18455              1508
3                  4.9             58419       48263              4921
4                  2.1              9234        7638               779

   less_10k  10k_15k  15k_25k  25k_35k  35k_50k  50k_75k  75k_100k  100k_150k  \
0       5.0      2.7      4.0      2.7      6.2     10.6       8.8       14.4
1       8.7      3.2      0.0      9.8      5.7      4.2      14.7       11.4
2       1.9      0.8      4.8      6.4      5.0     12.8      10.6       17.9
3       4.5      2.2      6.5      3.6      5.6     12.4      11.2       14.6
4       3.4      1.1      0.5      1.8      3.3      8.0       6.8       18.6

   150k_200k  more_200k  med_income  mean_incom  speaks_only_english  \
0        9.8       35.8      127375      242978                19681
1       11.6       30.9      110625      225183                 2922
2        9.6       30.3      137146      233358                13727
3       10.0       29.4      114010      196844                39143
4       16.1       40.5      169844      224631                 5618

   naturalized  non_citizen  pop_pov  pop_below_  pcnt_pov  \
0         3045         3306    27963        1922       3.0
1          331         1207     5085         522       7.3
2         3315         2372    21155        1075       4.5
3         7054         6535    58980        3686       3.4
4         1372         1539     9384         336       1.1

   families_on_suplimental_income  families_on_social_security  2_ppl_fam  \
0                             109                         1683       3797
1                              25                           88        517
2                             129                         1401       3133
3                             500                         3513       8171
4                               0                          216       1046

   3_to_4_ppl  5_to_6_ppl  GT_7_ppl_f  insured  per_insured  uninsured  \
0        2267         275          13    27299         97.3        744
1         662          27           0     4992         98.2         93
2        1765         236           0    20715         97.9        440
3        5192         530           3    57424         96.9       1820
4        1009         130          16     9027         96.2        357
```

	per_uninsured	owner	rent	4_16_2020_positive	4_16_2020_tests
0	2.7	5365	9479	233	682
1	1.8	675	1877	40	105
2	2.1	4933	6102	294	631
3	3.1	10129	20562	460	1207
4	3.8	1233	3327	32	92

	4_16_2020_positive_rate	4_2_2020_positive	4_2_2020_tests
0	34.16	121	385
1	38.10	24	57
2	46.59	160	371
3	38.11	212	596
4	34.78	17	50

	4_2_2020_positive_rate	4_3_2020_positive	4_3_2020_tests
0	0.314286	121	385
1	0.421053	24	57
2	0.431267	160	371
3	0.355705	212	596
4	0.340000	17	50

	4_3_2020_positive_rate	4_7_2020_positive	4_7_2020_tests
0	0.314286	171	492
1	0.421053	29	72
2	0.431267	204	452
3	0.355705	281	737
4	0.340000	20	61

	4_7_2020_positive_rate	4_8_2020_positive	4_8_2020_tests
0	34.76	187	544
1	40.28	32	81
2	45.13	232	501
3	38.13	317	834
4	32.79	23	73

	4_8_2020_positive_rate	hospital_count	3_26_bb_beds	4_2_bb_beds
0	34.38	14	7625	7993
1	39.51	9	4922	4922
2	46.31	15	7573	7729
3	38.01	13	5936	6233
4	31.51	4	679	688

	4_7_bb_beds	licensed_beds	staffed_beds	ICU_beds	adult_icu_beds
0	7993	6493	5519	745	745
1	4922	4093	3765	410	410
2	7729	6277	5678	788	788

```
3         6233           4713          4177       514              514
4          688            713           420        35               35

   pediatrics_icu_beds  bed_utilization_rate  potential_increase_bed_capacity  \
0                  402              6.647990                              974
1                  273              4.741397                              328
2                  412              7.253486                              599
3                  337              6.464822                              536
4                   28              1.790607                              293

   avg_ventilator_use
0                 181
1                 138
2                 198
3                 147
4                  12
```

[3]:
```
boroughs_zip = pd.read_csv('data/bor_zip_codes.csv')
ny = pd.merge(ny,boroughs_zip[['GEOID10', 'borough']], left_on='GEOID_ZIP',␣
 ↪right_on='GEOID10')
del(boroughs_zip)
ny.head()
```

[3]:
```
   GEOID_ZIP  ALAND10  AWATER10  tot_pop  white  per_white  black  per_black  \
0      10065   984654         0    28109  24285       86.4    619        2.2
1      10069   249050         0     5085   3155       62.0    148        2.9
2      10075   477137         0    21556  18396       85.3    677        3.1
3      10128  1206191         0    59256  47167       79.6   2182        3.7
4      10280   297253     38409     9384   7360       78.4    184        2.0

   native  per_native  asian  per_asian  nativehawaiian  per_hawaiian  other  \
0      37         0.1   2666        9.5               0           0.0    154
1       0         0.0   1558       30.6               0           0.0      0
2     225         1.0   1047        4.9               0           0.0    902
3       0         0.0   5844        9.9               0           0.0   1666
4       0         0.0   1474       15.7               0           0.0     77

   per_other  two_or_mor  per_two_or   male  female  under_5  per_under_5  \
0        0.5         348         1.2  12248   15861     1650          5.9
1        0.0         224         4.4   2354    2731      368          7.2
2        4.2         309         1.4  10096   11460     1041          4.8
3        2.8        2397         4.0  25338   33918     3204          5.4
4        0.8         289         3.1   4477    4907      711          7.6

   5_to_9  per_5_to_9  10_to_14  per_10_to_14  15_to_19  per_15_to_19  \
0    1000         3.6       916           3.3       356           1.3
1     416         8.2       274           5.4        95           1.9
```

```
2    1252         5.8       736           3.4       763           3.5
3    2138         3.6      1911           3.2      1691           2.9
4     628         6.7       454           4.8       192           2.0

   20_to_24  per_20_to_24  25_to_29  per_25_to_29  30_to_34  per_30_to_34  \
0      1785           6.4      2671           9.5      3176          11.3
1       235           4.6       427           8.4       594          11.7
2       778           3.6      1400           6.5      2022           9.4
3      2644           4.5      6825          11.5      7216          12.2
4       239           2.5       523           5.6      1374          14.6

   35_to_39  per_35_to_39  40_to_44  per_40_to_44  45_to_49  per_45_to_49  \
0      2558           9.1      1607           5.7      1570           5.6
1       639          12.6       508          10.0       374           7.4
2      1643           7.6      1148           5.3      1453           6.7
3      4349           7.3      3849           6.5      4103           6.9
4       945          10.1      1021          10.9       771           8.2

   50_to_54  per_50_to_54  55_to_59  per_55_to_59  60_to_64  per_60_to_64  \
0      1796           6.4      1460           5.2      1522           5.4
1       378           7.4       235           4.6       197           3.9
2      1244           5.8      1118           5.2      1131           5.2
3      3845           6.5      2989           5.0      3494           5.9
4       763           8.1       530           5.6       375           4.0

   65_to_69  per_65_to_69  70_to_74  per_70_to_74  75_to_79  per_75_to_79  \
0      1388           4.9      1486           5.3      1159           4.1
1        61           1.2        87           1.7       109           2.1
2      1561           7.2      1276           5.9      1169           5.4
3      3461           5.8      2656           4.5      1980           3.3
4       370           3.9       254           2.7        79           0.8

   80_to_84  per_80_to_89  80_and_over  per_80_and_over  disabled  \
0       954           3.4         1055              3.8      1894
1         0           0.0           88              1.7       182
2       570           2.6         1251              5.8      1689
3      1557           2.6         1344              2.3      4825
4       115           1.2           40              0.4       326

   per_disabled  unemployment_rate  tot_households_snap  households_snap  \
0           6.8               70.2                14844              414
1           3.6               74.7                 2552               73
2           8.0               68.1                11035              217
3           8.1               72.2                30691             1506
4           3.5               83.7                 4560               97

   per_households_snap  tot_pop_mobility  same_house  moved_within_1yr  \
```

0	2.8	27635	22612	2247
1	2.9	5025	3790	859
2	2.0	21381	18455	1508
3	4.9	58419	48263	4921
4	2.1	9234	7638	779

	less_10k	10k_15k	15k_25k	25k_35k	35k_50k	50k_75k	75k_100k	100k_150k	\
0	5.0	2.7	4.0	2.7	6.2	10.6	8.8	14.4	
1	8.7	3.2	0.0	9.8	5.7	4.2	14.7	11.4	
2	1.9	0.8	4.8	6.4	5.0	12.8	10.6	17.9	
3	4.5	2.2	6.5	3.6	5.6	12.4	11.2	14.6	
4	3.4	1.1	0.5	1.8	3.3	8.0	6.8	18.6	

	150k_200k	more_200k	med_income	mean_incom	speaks_only_english	\
0	9.8	35.8	127375	242978	19681	
1	11.6	30.9	110625	225183	2922	
2	9.6	30.3	137146	233358	13727	
3	10.0	29.4	114010	196844	39143	
4	16.1	40.5	169844	224631	5618	

	naturalized	non_citizen	pop_pov	pop_below_	pcnt_pov	\
0	3045	3306	27963	1922	3.0	
1	331	1207	5085	522	7.3	
2	3315	2372	21155	1075	4.5	
3	7054	6535	58980	3686	3.4	
4	1372	1539	9384	336	1.1	

	families_on_suplimental_income	families_on_social_security	2_ppl_fam	\
0	109	1683	3797	
1	25	88	517	
2	129	1401	3133	
3	500	3513	8171	
4	0	216	1046	

	3_to_4_ppl	5_to_6_ppl	GT_7_ppl_f	insured	per_insured	uninsured	\
0	2267	275	13	27299	97.3	744	
1	662	27	0	4992	98.2	93	
2	1765	236	0	20715	97.9	440	
3	5192	530	3	57424	96.9	1820	
4	1009	130	16	9027	96.2	357	

	per_uninsured	owner	rent	4_16_2020_positive	4_16_2020_tests	\
0	2.7	5365	9479	233	682	
1	1.8	675	1877	40	105	
2	2.1	4933	6102	294	631	
3	3.1	10129	20562	460	1207	
4	3.8	1233	3327	32	92	

	4_16_2020_positive_rate	4_2_2020_positive	4_2_2020_tests
0	34.16	121	385
1	38.10	24	57
2	46.59	160	371
3	38.11	212	596
4	34.78	17	50

	4_2_2020_positive_rate	4_3_2020_positive	4_3_2020_tests
0	0.314286	121	385
1	0.421053	24	57
2	0.431267	160	371
3	0.355705	212	596
4	0.340000	17	50

	4_3_2020_positive_rate	4_7_2020_positive	4_7_2020_tests
0	0.314286	171	492
1	0.421053	29	72
2	0.431267	204	452
3	0.355705	281	737
4	0.340000	20	61

	4_7_2020_positive_rate	4_8_2020_positive	4_8_2020_tests
0	34.76	187	544
1	40.28	32	81
2	45.13	232	501
3	38.13	317	834
4	32.79	23	73

	4_8_2020_positive_rate	hospital_count	3_26_bb_beds	4_2_bb_beds
0	34.38	14	7625	7993
1	39.51	9	4922	4922
2	46.31	15	7573	7729
3	38.01	13	5936	6233
4	31.51	4	679	688

	4_7_bb_beds	licensed_beds	staffed_beds	ICU_beds	adult_icu_beds
0	7993	6493	5519	745	745
1	4922	4093	3765	410	410
2	7729	6277	5678	788	788
3	6233	4713	4177	514	514
4	688	713	420	35	35

	pediatrics_icu_beds	bed_utilization_rate	potential_increase_bed_capacity
0	402	6.647990	974
1	273	4.741397	328
2	412	7.253486	599

```
3                  337              6.464822                                536
4                   28              1.790607                                293

   avg_ventilator_use  GEOID10    borough
0                 181    10065  Manhattan
1                 138    10069  Manhattan
2                 198    10075  Manhattan
3                 147    10128  Manhattan
4                  12    10280  Manhattan
```

[4]:
```
# ny.drop(['GEOID10_x', 'GEOID10_y', 'borough_y'], axis=1, inplace=True)
# ny.rename(columns={"borough_x": "borough"})
# ny.head()
```

[5]:
```
gm = pd.read_csv('data/Google_Mobility_Report_Filtered.csv')
gm.head()
```

[5]:
```
  country_region_code country_region sub_region_1 sub_region_2       date  \
0                  US  United States     New York        Bronx  2/15/2020
1                  US  United States     New York        Bronx  2/16/2020
2                  US  United States     New York        Bronx  2/17/2020
3                  US  United States     New York        Bronx  2/18/2020
4                  US  United States     New York        Bronx  2/19/2020

   retail_and_recreation_percent_change_from_baseline  \
0                                                   0
1                                                  -1
2                                                   3
3                                                  -2
4                                                   2

   grocery_and_pharmacy_percent_change_from_baseline  \
0                                                  -8
1                                                  -4
2                                                  -8
3                                                  -6
4                                                  -6

   parks_percent_change_from_baseline  \
0                                  -5
1                                   5
2                                 -11
3                                 -11
4                                   0

   transit_stations_percent_change_from_baseline  \
0                                             -3
```

```
1                                                -2
2                                               -22
3                                                -7
4                                                -6

   workplaces_percent_change_from_baseline  \
0                                        0
1                                       -2
2                                      -38
3                                      -10
4                                       -9

   residential_percent_change_from_baseline
0                                         1
1                                         0
2                                        10
3                                         3
4                                         2
```

[6]:
```
# using an aggregate of drop in baseline from "normal" starting from Feb 15 -␣
 ↪Apr 11
# figured an aggregate would be best way to deal with the abundance of data
gm_agg = gm.groupby(['sub_region_2']).agg({col: ['sum'] for col in␣
 ↪['retail_and_recreation_percent_change_from_baseline',␣
 ↪'grocery_and_pharmacy_percent_change_from_baseline',␣
 ↪'parks_percent_change_from_baseline',␣
 ↪'transit_stations_percent_change_from_baseline',␣
 ↪'workplaces_percent_change_from_baseline',␣
 ↪'residential_percent_change_from_baseline']})
gm_agg.columns = ['_'.join(multi_index) for multi_index in gm_agg.columns.
 ↪ravel()]
gm_agg = gm_agg.reset_index()
gm_agg
```

[6]:
```
     sub_region_2  retail_and_recreation_percent_change_from_baseline_sum  \
0           Bronx                                                   -1155
1        Brooklyn                                                   -1407
2       Manhattan                                                   -2296
3          Queens                                                   -1470
4  Staten Island                                                   -1239

   grocery_and_pharmacy_percent_change_from_baseline_sum  \
0                                                   -211
1                                                   -249
2                                                  -1006
3                                                   -315
4                                                   -254
```

```
   parks_percent_change_from_baseline_sum  \
0                                    -758
1                                     -45
2                                   -1544
3                                     500
4                                    -229

   transit_stations_percent_change_from_baseline_sum  \
0                                              -1366
1                                              -1677
2                                              -2199
3                                              -1986
4                                              -1643

   workplaces_percent_change_from_baseline_sum  \
0                                        -1437
1                                        -1601
2                                        -1926
3                                        -1650
4                                        -1407

   residential_percent_change_from_baseline_sum
0                                           611
1                                           715
2                                           777
3                                           778
4                                           660
```

[7]:
```
ny = pd.merge(ny, gm_agg, left_on='borough', right_on='sub_region_2')
ny.sample(10).head(10)
```

[7]:
```
     GEOID_ZIP  ALAND10  AWATER10  tot_pop  white  per_white  black  \
89       11229  5592659    143662    83615  54043       64.6   4985
116      11221  3582803         0    83835  23203       27.7  41079
72       10467  6049162         0   103732  23469       22.6  36599
166      11417  2898418         0    31927  11426       35.8   2362
169      11420  5380501         0    48489   7399       15.3  11883
119      11224  4107702     78749    46707  27137       58.1  10495
173      11426  3494228         0    20801   8083       38.9   1379
129      11363  2227891     34217     6952   4124       59.3    124
66       10461  6201816         0    50348  24270       48.2   4714
61       10456  2635671         0    94218  10146       10.8  40069

     per_black  native  per_native  asian  per_asian  nativehawaiian   \
89         6.0     208         0.2  18540       22.2               76
116       49.0     558         0.7   2815        3.4               16
```

```
72       35.3     666         0.6   6000        5.8              51
166       7.4      75         0.2   9273       29.0               5
169      24.5     449         0.9  15052       31.0              76
119      22.5      68         0.1   2769        5.9               4
173       6.6      98         0.5   9202       44.2               0
129       1.8       0         0.0   2264       32.6               0
66        9.4     250         0.5   5600       11.1              43
61       42.5    1311         1.4    809        0.9             130

     per_hawaiian  other  per_other  two_or_mor  per_two_or   male  female  \
89            0.1   3437        4.1        2326         2.8  40272   43343
116           0.0  12889       15.4        3275         3.9  40147   43688
72            0.0  32759       31.6        4188         4.0  49745   53987
166           0.0   7092       22.2        1694         5.3  15311   16616
169           0.2  10448       21.5        3182         6.6  23765   24724
119           0.0   4934       10.6        1300         2.8  20929   25778
173           0.0   1395        6.7         644         3.1  10388   10413
129           0.0    301        4.3         139         2.0   3341    3611
66            0.1  13312       26.4        2159         4.3  24497   25851
61            0.1  38727       41.1        3026         3.2  43844   50374

     under_5  per_under_5  5_to_9  per_5_to_9  10_to_14  per_10_to_14  \
89      5306          6.3    5095         6.1      4895           5.9
116     5506          6.6    4236         5.1      4278           5.1
72      8525          8.2    7106         6.9      7207           6.9
166     1893          5.9    1740         5.4      2080           6.5
169     2433          5.0    2417         5.0      2858           5.9
119     2692          5.8    2620         5.6      2490           5.3
173     1613          7.8    1285         6.2      1067           5.1
129      339          4.9     399         5.7       439           6.3
66      3177          6.3    2744         5.5      3038           6.0
61      7236          7.7    7322         7.8      7756           8.2

     15_to_19  per_15_to_19  20_to_24  per_20_to_24  25_to_29  per_25_to_29  \
89       4007           4.8      5232           6.3      5891           7.0
116      4993           6.0      8461          10.1     11608          13.8
72       6131           5.9      7343           7.1      9373           9.0
166      2122           6.6      2494           7.8      2491           7.8
169      3079           6.3      3651           7.5      3990           8.2
119      2537           5.4      2493           5.3      3133           6.7
173       880           4.2      1387           6.7      1697           8.2
129       219           3.2       275           4.0       471           6.8
66       2169           4.3      3059           6.1      4354           8.6
61       7503           8.0      8149           8.6      7858           8.3

     30_to_34  per_30_to_34  35_to_39  per_35_to_39  40_to_44  per_40_to_44  \
89       5086           6.1      5305           6.3      4959           5.9
```

```
116      7254           8.7      6217           7.4      4867           5.8
72       8179           7.9      7532           7.3      6330           6.1
166      1893           5.9      1961           6.1      2174           6.8
169      3472           7.2      2838           5.9      3210           6.6
119      2068           4.4      2664           5.7      2338           5.0
173      1568           7.5      1243           6.0      1192           5.7
129       235           3.4       411           5.9       528           7.6
66       4335           8.6      3774           7.5      3052           6.1
61       6603           7.0      5963           6.3      5360           5.7

     45_to_49  per_45_to_49  50_to_54  per_50_to_54  55_to_59  per_55_to_59  \
89       5285           6.3      5280           6.3      6145           7.3
116      4859           5.8      4912           5.9      4604           5.5
72       6403           6.2      6678           6.4      5618           5.4
166      2443           7.7      2559           8.0      2354           7.4
169      3325           6.9      4028           8.3      4199           8.7
119      2634           5.6      2921           6.3      2874           6.2
173      1233           5.9      1543           7.4      1565           7.5
129       356           5.1       701          10.1       681           9.8
66       3464           6.9      3620           7.2      2966           5.9
61       5661           6.0      5991           6.4      5857           6.2

     60_to_64  per_60_to_64  65_to_69  per_65_to_69  70_to_74  per_70_to_74  \
89       6203           7.4      5067           6.1      3421           4.1
116      3364           4.0      3119           3.7      2093           2.5
72       5947           5.7      3541           3.4      2946           2.8
166      1746           5.5      1781           5.6       796           2.5
169      3298           6.8      1758           3.6      1434           3.0
119      3173           6.8      3599           7.7      2263           4.8
173      1325           6.4      1314           6.3       942           4.5
129       399           5.7       393           5.7       404           5.8
66       2758           5.5      2284           4.5      1635           3.2
61       4457           4.7      3138           3.3      2205           2.3

     75_to_79  per_75_to_79  80_to_84  per_80_to_89  80_and_over  \
89       2593           3.1      1753           2.1         2092
116      1486           1.8      1100           1.3          878
72       2162           2.1      1288           1.2         1423
166       524           1.6       435           1.4          441
169      1420           2.9       471           1.0          608
119      2658           5.7      1574           3.4         1976
173       421           2.0       139           0.7          387
129       355           5.1       126           1.8          221
66       1234           2.5      1389           2.8         1296
61       1297           1.4      1044           1.1          818

     per_80_and_over  disabled  per_disabled  unemployment_rate  \
```

```
89               2.5      8497          10.2               59.6
116              1.0      7865           9.4               66.7
72               1.4     15955          15.6               62.6
166              1.4      3330          10.4               62.9
169              1.3      5885          12.1               63.9
119              4.2      9752          21.2               48.4
173              1.9      1512           7.3               65.5
129              3.2       581           8.4               63.3
66               2.6      6522          13.2               63.0
61               0.9     15956          17.1               56.8

     tot_households_snap  households_snap  per_households_snap  \
89                 30946             6207                 20.1
116                30871             8964                 29.0
72                 36732            13256                 36.1
166                 9060             1353                 14.9
169                13159             2081                 15.8
119                18725             7717                 41.2
173                 6356              429                  6.7
129                 2661              144                  5.4
66                 18927             3088                 16.3
61                 30774            15294                 49.7

     tot_pop_mobility  same_house  moved_within_1yr  less_10k  10k_15k  \
89              82637       76079              4347       7.1      6.9
116             82645       72633              5393      12.9      7.4
72             102147       90397              7873      12.4     10.3
166             31685       28823              1979       5.6      3.1
169             48082       45594              1334       4.6      3.1
119             46259       42489              2869      11.8     15.0
173             20600       18953              1370       3.0      2.9
129              6856        6539               150       4.0      0.9
66              49872       43199              4131       7.3      6.3
61              93006       85005              4564      18.1     12.7

     15k_25k  25k_35k  35k_50k  50k_75k  75k_100k  100k_150k  150k_200k  \
89       9.8      8.2     12.0     13.9      11.4       15.9        6.7
116      9.0      8.9     11.7     14.4      11.4       14.0        5.5
72      13.2     11.8     14.6     17.7       7.9        8.0        2.6
166      8.7      6.6     10.1     17.5      13.3       18.9        9.4
169      8.3      8.2     13.0     16.1      13.0       16.5        9.1
119     15.0     10.3     11.6     13.1       8.2        8.8        3.3
173      4.3      4.1      9.7     16.3      15.4       21.5       12.1
129      6.6      6.0      9.7     13.7       7.7       22.4       10.7
66       9.7      9.2     11.3     18.2      13.1       14.1        6.8
61      16.2     11.1     12.8     14.4       6.7        5.7        1.2
```

	more_200k	med_income	mean_incom	speaks_only_english	naturalized	\
89	7.9	60873	83189	33659	27089	
116	4.9	50153	72291	50139	9743	
72	1.4	37015	51062	38389	18436	
166	7.0	73424	88365	13572	9296	
169	8.2	67932	91191	31649	17572	
119	2.9	33131	54541	17700	15947	
173	10.7	88803	107612	8309	6087	
129	18.4	104792	131100	3372	1638	
66	4.1	56612	73335	21513	7609	
61	1.1	27106	43257	28544	14872	

	non_citizen	pop_pov	pop_below_	pcnt_pov	\
89	12278	83400	12184	11.7	
116	10012	83193	19276	19.9	
72	20402	102221	28506	25.8	
166	5025	31856	4521	11.8	
169	7386	48303	5825	10.4	
119	4843	45892	12792	22.2	
173	2684	20722	1564	6.4	
129	630	6929	409	5.8	
66	6065	49411	7332	14.0	
61	18382	92563	35856	35.4	

	families_on_suplimental_income	families_on_social_security	2_ppl_fam	\
89	2249	5735	8673	
116	2345	3547	6530	
72	4100	4601	9027	
166	514	1893	2005	
169	1098	2550	2966	
119	2992	3819	5765	
173	239	1146	1500	
129	84	622	885	
66	1257	2961	4981	
61	5437	3569	7218	

	3_to_4_ppl	5_to_6_ppl	GT_7_ppl_f	insured	per_insured	uninsured	\
89	9125	3082	701	77857	93.2	5639	
116	6928	2483	549	74102	88.7	9450	
72	10060	3552	912	92373	90.1	10097	
166	3646	1555	248	29372	92.0	2552	
169	5464	2182	537	44260	91.3	4198	
119	4889	1198	153	43055	93.7	2878	
173	2243	779	253	19535	94.1	1218	
129	891	178	9	6705	96.4	247	
66	5416	1492	211	45733	92.4	3784	
61	9706	3317	946	83525	89.7	9558	

```
     per_uninsured  owner   rent  4_16_2020_positive  4_16_2020_tests  \
89             6.8  14601  16345                 971             1699
116           11.3   6646  24225                 740             1261
72             9.9   5011  31721                2087             3449
166            8.0   5451   3609                 490              807
169            8.7   8799   4360                 789             1256
119            6.3   5203  13522                 588             1004
173            5.9   4832   1524                 328              588
129            3.6   2008    653                  73              161
66             7.6   6162  12765                1173             2220
61            10.3   1756  29018                1376             2252

     4_16_2020_positive_rate  4_2_2020_positive  4_2_2020_tests  \
89                     57.15                316             640
116                    58.68                260             455
72                     60.51                638            1134
166                    60.72                173             291
169                    62.82                223             390
119                    58.57                133             304
173                    55.78                101             202
129                    45.34                 27              65
66                     52.84                376             714
61                     61.10                355             693

     4_2_2020_positive_rate  4_3_2020_positive  4_3_2020_tests  \
89                 0.493750                316             640
116                0.571429                260             455
72                 0.562610                638            1134
166                0.594502                173             291
169                0.571795                223             390
119                0.437500                133             304
173                0.500000                101             202
129                0.415385                 27              65
66                 0.526611                376             714
61                 0.512266                355             693

     4_3_2020_positive_rate  4_7_2020_positive  4_7_2020_tests  \
89                 0.493750                470             890
116                0.571429                384             644
72                 0.562610                941            1625
166                0.594502                254             418
169                0.571795                398             656
119                0.437500                235             446
173                0.500000                163             313
129                0.415385                 42              93
66                 0.526611                536             993
```

```
61                 0.512266                619            1054

     4_7_2020_positive_rate  4_8_2020_positive  4_8_2020_tests  \
89                    52.81                603            1117
116                   59.63                477             809
72                    57.91               1209            2091
166                   60.77                323             522
169                   60.67                506             816
119                   52.69                314             569
173                   52.08                225             404
129                   45.16                 55             116
66                    53.98                687            1311
61                    58.73                802            1353

     4_8_2020_positive_rate  hospital_count  3_26_bb_beds  4_2_bb_beds  \
89                    53.98               3           717          827
116                   58.96               4          1260         1291
72                    57.82               6          2532         2609
166                   61.88               1           408          408
169                   62.01               1           408          408
119                   55.18               1           371          481
173                   55.69               1          1025         1025
129                   47.41               1          1025         1025
66                    52.40               3          1133         1133
61                    59.28               6          2673         3339

     4_7_bb_beds  licensed_beds  staffed_beds  ICU_beds  adult_icu_beds  \
89           827            505           426        29              29
116         1291           1278           809        63              63
72          2609           1908          2197       172             172
166          408            402           285        19              19
169          408            402           285        19              19
119          481            371           292        22              22
173         1025              0             0         0               0
129         1025              0             0         0               0
66          1133            657           551        71              71
61          3339           1236          1261       115             115

     pediatrics_icu_beds  bed_utilization_rate  \
89                    10              1.525469
116                   20              2.582287
72                   102              2.700614
166                   19              0.694420
169                   19              0.694420
119                   10              0.798808
173                    0              0.000000
129                    0              0.000000
```

```
66                    25             1.708695
61                    90             2.906461

     potential_increase_bed_capacity  avg_ventilator_use  GEOID10   borough  \
89                                79                  22    11229  Brooklyn
116                              469                  25    11221  Brooklyn
72                              -289                  60    10467     Bronx
166                              117                  12    11417    Queens
169                              117                  12    11420    Queens
119                               79                  16    11224  Brooklyn
173                                0                   0    11426    Queens
129                                0                   0    11363    Queens
66                               106                   9    10461     Bronx
61                               -25                  33    10456     Bronx

    sub_region_2  retail_and_recreation_percent_change_from_baseline_sum  \
89      Brooklyn                                                  -1407
116     Brooklyn                                                  -1407
72         Bronx                                                  -1155
166       Queens                                                  -1470
169       Queens                                                  -1470
119     Brooklyn                                                  -1407
173       Queens                                                  -1470
129       Queens                                                  -1470
66         Bronx                                                  -1155
61         Bronx                                                  -1155

     grocery_and_pharmacy_percent_change_from_baseline_sum  \
89                                                    -249
116                                                   -249
72                                                    -211
166                                                   -315
169                                                   -315
119                                                   -249
173                                                   -315
129                                                   -315
66                                                    -211
61                                                    -211

     parks_percent_change_from_baseline_sum  \
89                                      -45
116                                     -45
72                                     -758
166                                     500
169                                     500
119                                     -45
173                                     500
```

```
129                                    500
66                                    -758
61                                    -758

     transit_stations_percent_change_from_baseline_sum  \
89                                                -1677
116                                               -1677
72                                                -1366
166                                               -1986
169                                               -1986
119                                               -1677
173                                               -1986
129                                               -1986
66                                                -1366
61                                                -1366

     workplaces_percent_change_from_baseline_sum  \
89                                         -1601
116                                        -1601
72                                         -1437
166                                        -1650
169                                        -1650
119                                        -1601
173                                        -1650
129                                        -1650
66                                         -1437
61                                         -1437

     residential_percent_change_from_baseline_sum
89                                            715
116                                           715
72                                            611
166                                           778
169                                           778
119                                           715
173                                           778
129                                           778
66                                            611
61                                            611
```

0.3 Data Transformations

[8]:
```
ny['per_minority'] =  ny['per_black'] + ny['per_native'] + ny['per_asian'] +␣
↪ny['per_hawaiian'] + ny['per_other'] + ny['per_two_or']
```

```
␣
↪---------------------------------------------------------------------------

       ModuleNotFoundError                       Traceback (most recent call␣
↪last)

       ModuleNotFoundError: No module named 'numpy.core._multiarray_umath'
```

[9]:
```
conditions = [(ny['per_minority'] <= 50.0),
              (ny['per_minority'] > 50.0)]
choices = [0,1]

ny['minority-majority-50'] = np.select(conditions, choices, default=2)
```

[10]:
```
conditions = [(ny['per_minority'] <= 60.0),
              (ny['per_minority'] > 60.0)]
choices = [0,1]

ny['minority-majority-60'] = np.select(conditions, choices, default=2)
```

[11]:
```
conditions = [(ny['per_minority'] <= 70.0),
              (ny['per_minority'] > 70.0)]
choices = [0,1]

ny['minority-majority-70'] = np.select(conditions, choices, default=2)
```

[12]:
```
ny["pop_density"] = ny['tot_pop']/ny['ALAND10']
ny['per_infected'] = ny['4_16_2020_positive']/ny['tot_pop']
ny['change_in_bed'] = ny['4_7_bb_beds']-ny['3_26_bb_beds']
ny['per_male'] = ny['male']/ny['tot_pop']
ny['per_female'] = ny['female']/ny['tot_pop']

ny['per_youth'] =␣
 ↪ny['per_under_5']+ny['per_5_to_9']+ny['per_10_to_14']+ny['per_15_to_19']
ny['per_young_adult'] = ny['per_20_to_24']+ny['per_25_to_29']+ny['per_30_to_34']
ny['per_late_adult'] = ny['per_35_to_39']+ny['per_40_to_44'] +␣
 ↪ny['per_45_to_49']+ny['per_50_to_54']+ny['per_55_to_59']
ny['per_elderly'] =␣
 ↪ny['per_60_to_64']+ny['per_65_to_69']+ny['per_70_to_74']+ny['per_75_to_79']+ny['per_80_to_8
```

[13]:
```
old_columns = [x for x in ny.columns if x not in ['per_minority',␣
 ↪'minority-majority-50', 'minority-majority-60',␣
 ↪'minority-majority-70',"pop_density",'per_infected','change_in_bed',␣
 ↪'per_female', 'per_male','per_youth','per_young_adult','per_late_adult',␣
 ↪'per_elderly']]
insert_index = old_columns.index('4_16_2020_positive')
```

```
ny = ny[old_columns[:insert_index] + ['per_minority', 'minority-majority-50',␣
↪'minority-majority-60',␣
↪'minority-majority-70',"pop_density",'per_infected','change_in_bed','per_female',␣
↪'per_male', 'per_youth','per_young_adult','per_late_adult', 'per_elderly'] +␣
↪old_columns[insert_index:]]
```

[14]:
```
ny.head()
```

[14]:

	GEOID_ZIP	ALAND10	AWATER10	tot_pop	white	per_white	black	per_black
0	10065	984654	0	28109	24285	86.4	619	2.2
1	10069	249050	0	5085	3155	62.0	148	2.9
2	10075	477137	0	21556	18396	85.3	677	3.1
3	10128	1206191	0	59256	47167	79.6	2182	3.7
4	10280	297253	38409	9384	7360	78.4	184	2.0

	native	per_native	asian	per_asian	nativehawaiian	per_hawaiian	other
0	37	0.1	2666	9.5	0	0.0	154
1	0	0.0	1558	30.6	0	0.0	0
2	225	1.0	1047	4.9	0	0.0	902
3	0	0.0	5844	9.9	0	0.0	1666
4	0	0.0	1474	15.7	0	0.0	77

	per_other	two_or_mor	per_two_or	male	female	under_5	per_under_5
0	0.5	348	1.2	12248	15861	1650	5.9
1	0.0	224	4.4	2354	2731	368	7.2
2	4.2	309	1.4	10096	11460	1041	4.8
3	2.8	2397	4.0	25338	33918	3204	5.4
4	0.8	289	3.1	4477	4907	711	7.6

	5_to_9	per_5_to_9	10_to_14	per_10_to_14	15_to_19	per_15_to_19
0	1000	3.6	916	3.3	356	1.3
1	416	8.2	274	5.4	95	1.9
2	1252	5.8	736	3.4	763	3.5
3	2138	3.6	1911	3.2	1691	2.9
4	628	6.7	454	4.8	192	2.0

	20_to_24	per_20_to_24	25_to_29	per_25_to_29	30_to_34	per_30_to_34
0	1785	6.4	2671	9.5	3176	11.3
1	235	4.6	427	8.4	594	11.7
2	778	3.6	1400	6.5	2022	9.4
3	2644	4.5	6825	11.5	7216	12.2
4	239	2.5	523	5.6	1374	14.6

	35_to_39	per_35_to_39	40_to_44	per_40_to_44	45_to_49	per_45_to_49
0	2558	9.1	1607	5.7	1570	5.6
1	639	12.6	508	10.0	374	7.4
2	1643	7.6	1148	5.3	1453	6.7

```
3      4349          7.3      3849          6.5      4103          6.9
4       945         10.1      1021         10.9       771          8.2

   50_to_54  per_50_to_54  55_to_59  per_55_to_59  60_to_64  per_60_to_64  \
0      1796           6.4      1460           5.2      1522           5.4
1       378           7.4       235           4.6       197           3.9
2      1244           5.8      1118           5.2      1131           5.2
3      3845           6.5      2989           5.0      3494           5.9
4       763           8.1       530           5.6       375           4.0

   65_to_69  per_65_to_69  70_to_74  per_70_to_74  75_to_79  per_75_to_79  \
0      1388           4.9      1486           5.3      1159           4.1
1        61           1.2        87           1.7       109           2.1
2      1561           7.2      1276           5.9      1169           5.4
3      3461           5.8      2656           4.5      1980           3.3
4       370           3.9       254           2.7        79           0.8

   80_to_84  per_80_to_89  80_and_over  per_80_and_over  disabled  \
0       954           3.4         1055              3.8      1894
1         0           0.0           88              1.7       182
2       570           2.6         1251              5.8      1689
3      1557           2.6         1344              2.3      4825
4       115           1.2           40              0.4       326

   per_disabled  unemployment_rate  tot_households_snap  households_snap  \
0           6.8               70.2                14844              414
1           3.6               74.7                 2552               73
2           8.0               68.1                11035              217
3           8.1               72.2                30691             1506
4           3.5               83.7                 4560               97

   per_households_snap  tot_pop_mobility  same_house  moved_within_1yr  \
0                  2.8             27635       22612              2247
1                  2.9              5025        3790               859
2                  2.0             21381       18455              1508
3                  4.9             58419       48263              4921
4                  2.1              9234        7638               779

   less_10k  10k_15k  15k_25k  25k_35k  35k_50k  50k_75k  75k_100k  100k_150k  \
0       5.0      2.7      4.0      2.7      6.2     10.6       8.8       14.4
1       8.7      3.2      0.0      9.8      5.7      4.2      14.7       11.4
2       1.9      0.8      4.8      6.4      5.0     12.8      10.6       17.9
3       4.5      2.2      6.5      3.6      5.6     12.4      11.2       14.6
4       3.4      1.1      0.5      1.8      3.3      8.0       6.8       18.6

   150k_200k  more_200k  med_income  mean_incom  speaks_only_english  \
0        9.8       35.8      127375      242978                19681
```

```
1       11.6       30.9      110625      225183                2922
2        9.6       30.3      137146      233358               13727
3       10.0       29.4      114010      196844               39143
4       16.1       40.5      169844      224631                5618

   naturalized  non_citizen  pop_pov  pop_below_  pcnt_pov  \
0         3045         3306    27963        1922       3.0
1          331         1207     5085         522       7.3
2         3315         2372    21155        1075       4.5
3         7054         6535    58980        3686       3.4
4         1372         1539     9384         336       1.1

   families_on_suplimental_income  families_on_social_security  2_ppl_fam  \
0                             109                         1683       3797
1                              25                           88        517
2                             129                         1401       3133
3                             500                         3513       8171
4                               0                          216       1046

   3_to_4_ppl  5_to_6_ppl  GT_7_ppl_f  insured  per_insured  uninsured  \
0        2267         275          13    27299         97.3        744
1         662          27           0     4992         98.2         93
2        1765         236           0    20715         97.9        440
3        5192         530           3    57424         96.9       1820
4        1009         130          16     9027         96.2        357

   per_uninsured  owner   rent  per_minority  minority-majority-50  \
0            2.7   5365   9479          13.5                     0
1            1.8    675   1877          37.9                     0
2            2.1   4933   6102          14.6                     0
3            3.1  10129  20562          20.4                     0
4            3.8   1233   3327          21.6                     0

   minority-majority-60  minority-majority-70  pop_density  per_infected  \
0                     0                     0     0.028547      0.008289
1                     0                     0     0.020418      0.007866
2                     0                     0     0.045178      0.013639
3                     0                     0     0.049127      0.007763
4                     0                     0     0.031569      0.003410

   change_in_bed  per_female  per_male  per_youth  per_young_adult  \
0            368    0.564268  0.435732       14.1             27.2
1              0    0.537070  0.462930       22.7             24.7
2            156    0.531639  0.468361       17.5             19.5
3            297    0.572398  0.427602       15.1             28.2
4              9    0.522911  0.477089       21.1             22.7
```

```
   per_late_adult  per_elderly  4_16_2020_positive  4_16_2020_tests  \
0            32.0         26.9                 233              682
1            42.0         10.6                  40              105
2            30.6         32.1                 294              631
3            32.2         24.4                 460             1207
4            42.9         13.0                  32               92

   4_16_2020_positive_rate  4_2_2020_positive  4_2_2020_tests  \
0                    34.16                121             385
1                    38.10                 24              57
2                    46.59                160             371
3                    38.11                212             596
4                    34.78                 17              50

   4_2_2020_positive_rate  4_3_2020_positive  4_3_2020_tests  \
0                0.314286                121             385
1                0.421053                 24              57
2                0.431267                160             371
3                0.355705                212             596
4                0.340000                 17              50

   4_3_2020_positive_rate  4_7_2020_positive  4_7_2020_tests  \
0                0.314286                171             492
1                0.421053                 29              72
2                0.431267                204             452
3                0.355705                281             737
4                0.340000                 20              61

   4_7_2020_positive_rate  4_8_2020_positive  4_8_2020_tests  \
0                   34.76                187             544
1                   40.28                 32              81
2                   45.13                232             501
3                   38.13                317             834
4                   32.79                 23              73

   4_8_2020_positive_rate  hospital_count  3_26_bb_beds  4_2_bb_beds  \
0                   34.38              14          7625         7993
1                   39.51               9          4922         4922
2                   46.31              15          7573         7729
3                   38.01              13          5936         6233
4                   31.51               4           679          688

   4_7_bb_beds  licensed_beds  staffed_beds  ICU_beds  adult_icu_beds  \
0         7993           6493          5519       745             745
1         4922           4093          3765       410             410
2         7729           6277          5678       788             788
3         6233           4713          4177       514             514
```

```
4          688            713           420        35              35

   pediatrics_icu_beds  bed_utilization_rate  potential_increase_bed_capacity  \
0                  402              6.647990                              974
1                  273              4.741397                              328
2                  412              7.253486                              599
3                  337              6.464822                              536
4                   28              1.790607                              293

   avg_ventilator_use  GEOID10    borough sub_region_2  \
0                 181    10065  Manhattan    Manhattan
1                 138    10069  Manhattan    Manhattan
2                 198    10075  Manhattan    Manhattan
3                 147    10128  Manhattan    Manhattan
4                  12    10280  Manhattan    Manhattan

   retail_and_recreation_percent_change_from_baseline_sum  \
0                                                 -2296
1                                                 -2296
2                                                 -2296
3                                                 -2296
4                                                 -2296

   grocery_and_pharmacy_percent_change_from_baseline_sum  \
0                                                 -1006
1                                                 -1006
2                                                 -1006
3                                                 -1006
4                                                 -1006

   parks_percent_change_from_baseline_sum  \
0                                   -1544
1                                   -1544
2                                   -1544
3                                   -1544
4                                   -1544

   transit_stations_percent_change_from_baseline_sum  \
0                                                 -2199
1                                                 -2199
2                                                 -2199
3                                                 -2199
4                                                 -2199

   workplaces_percent_change_from_baseline_sum  \
0                                        -1926
1                                        -1926
```

```
2                                          -1926
3                                          -1926
4                                          -1926

   residential_percent_change_from_baseline_sum
0                                           777
1                                           777
2                                           777
3                                           777
4                                           777
```

[15]:
```
ny.to_csv('output/ny_final.csv', index=False)
```

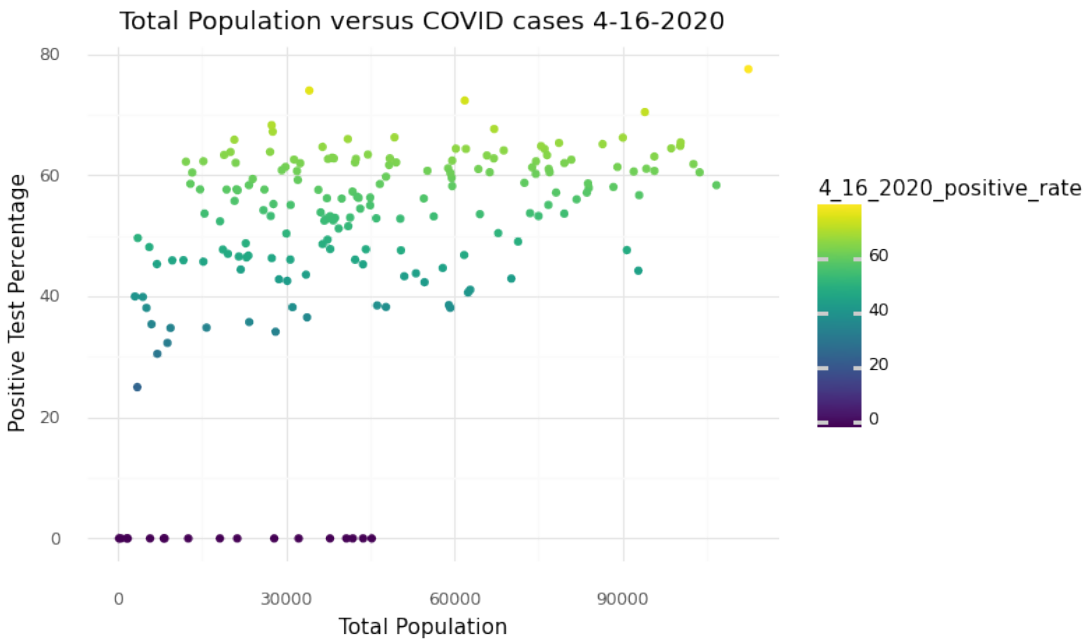
0.4 Question: Do the socio-economic variables of a zip-code predict COVID cases?

0.5 1 - Which variables go together using factor analysis?

0.6 2 - Which of these factors predict COVID cases?

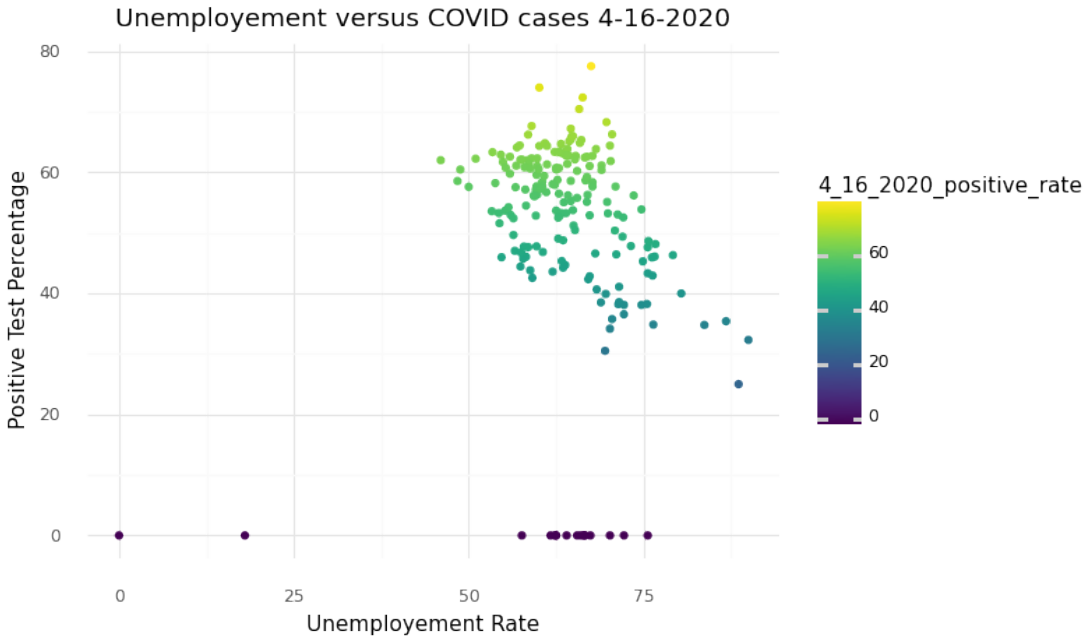
[16]:
```
from plotnine import *
import statsmodels.api as sm

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score

from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from factor_analyzer import FactorAnalyzer
import matplotlib.pyplot as plt

%matplotlib inline
```

[17]:
```
(ggplot(ny, aes(x='tot_pop', y='4_16_2020_positive_rate', color=␣
↪'4_16_2020_positive_rate'))+ geom_point()+theme_minimal()+ ggtitle("Total␣
↪Population versus COVID cases 4-16-2020")+ xlab("Total␣
↪Population")+ylab("Positive Test Percentage"))
```



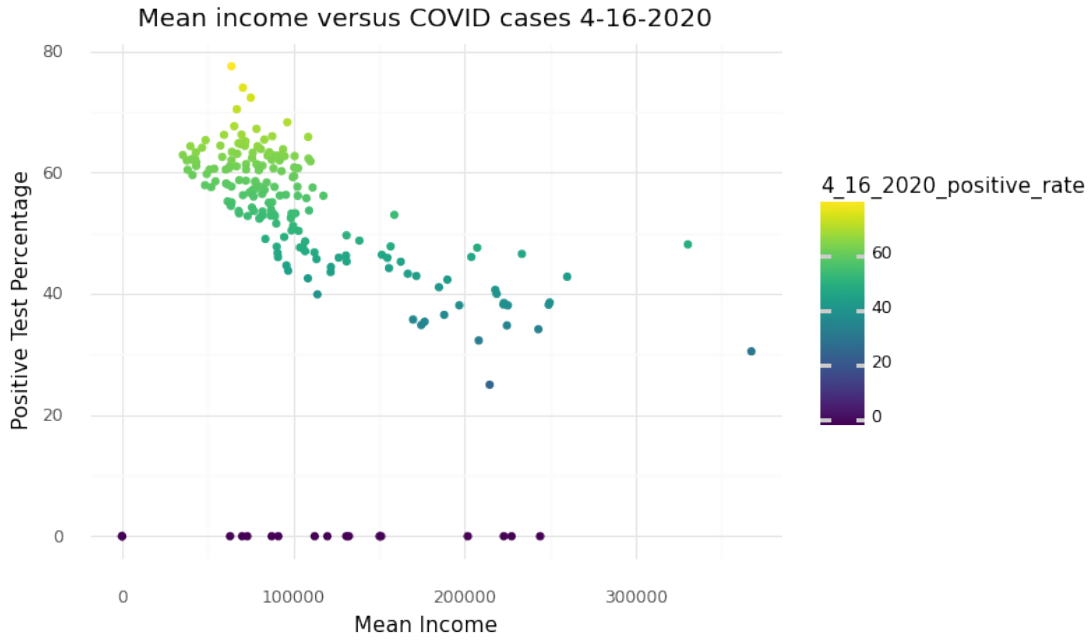
[17]: <ggplot: (-9223371901821560884)>

[18]:
```
(ggplot(ny, aes(x='unemployment_rate', y='4_16_2020_positive_rate', color=␣
↪'4_16_2020_positive_rate'))+ geom_point()+theme_minimal()+␣
↪ggtitle("Unemployement versus COVID cases 4-16-2020")+ xlab("Unemployement␣
↪Rate")+ylab("Positive Test Percentage"))
```



[18]:
```
<ggplot: (-9223371901819367808)>
```

[19]:
```
(ggplot(ny, aes(x='mean_incom', y='4_16_2020_positive_rate', color=␣
↪'4_16_2020_positive_rate'))+ geom_point()+theme_minimal()+ ggtitle("Mean␣
↪income versus COVID cases 4-16-2020")+ xlab("Mean Income")+ylab("Positive␣
↪Test Percentage"))
```



[19]:
```
<ggplot: (-9223371901819323140)>
```

[20]:
```
#creating a subset of the ny dataframe containing socio-economic variables
df = ny[['tot_pop','disabled', 'per_disabled',␣
↪'unemployment_rate','tot_households_snap','households_snap','per_households_snap','tot_pop_
↪'4_16_2020_positive_rate']].copy()

features = ['tot_pop','disabled', 'per_disabled',␣
↪'unemployment_rate','tot_households_snap','households_snap','per_households_snap','tot_pop_
```

[21]:
```
#standardize variables
zScore = StandardScaler() #standardize variables makes them easier with the math
zScore.fit(df[features])
df[features] = zScore.transform(df[features])
```

[22]:
```
# Create factor analysis object and perform factor analysis
fa = FactorAnalyzer(n_factors = 5, rotation=None)
```

```
fa.fit(df[features])

loadings = fa.loadings_

# Check Eigenvalues
ev, v =  fa.get_eigenvalues()

#create dataframe with relations between factors and variables
meaning = pd.DataFrame(loadings, columns = ['factor1',␣
 ↪'factor2','factor3','factor4','factor5'], index=features)
meaning
```

[22]:

	factor1	factor2	factor3	factor4	\
tot_pop	0.836267	0.543765	0.031111	-0.023744	
disabled	0.898024	0.256774	0.132348	0.159859	
per_disabled	0.374822	-0.369962	-0.051127	0.396301	
unemployment_rate	-0.319211	0.396844	0.212329	-0.059144	
tot_households_snap	0.732409	0.588941	0.184262	0.044988	
households_snap	0.912059	-0.014669	0.280308	0.059714	
per_households_snap	0.830969	-0.394368	0.150628	0.123576	
tot_pop_mobility	0.835847	0.544910	0.029210	-0.023882	
less_10k	0.680071	-0.368770	0.312668	0.136480	
10k_15k	0.740718	-0.421340	0.200697	0.284955	
15k_25k	0.807153	-0.350088	-0.038995	0.128516	
25k_35k	0.770754	-0.241363	-0.319964	0.065335	
35k_50k	0.711773	-0.224412	-0.441800	0.061116	
50k_75k	0.397687	0.051170	-0.721685	0.128649	
75k_100k	-0.130069	0.283265	-0.729151	0.183699	
100k_150k	-0.607033	0.461195	-0.375901	0.120247	
150k_200k	-0.729648	0.445769	-0.074266	0.149161	
more_200k	-0.692630	0.308830	0.569008	-0.058026	
med_income	-0.745729	0.427163	0.217099	0.028780	
mean_incom	-0.696819	0.372863	0.455279	-0.009971	
non_citizen	0.769302	0.285566	-0.076679	-0.398411	
pop_pov	0.835882	0.545098	0.025804	-0.025025	
pop_below_	0.919901	0.041718	0.265728	-0.013178	
pcnt_pov	0.816379	-0.415057	0.196487	0.086333	
families_on_suplimental_income	0.916874	0.046863	0.214837	0.099376	
families_on_social_security	0.641541	0.611993	-0.180285	0.158315	
insured	0.810734	0.571761	0.056286	0.035468	
per_insured	-0.246880	0.269769	0.101500	0.711325	
uninsured	0.801791	0.238953	-0.126373	-0.406268	
per_uninsured	0.615897	-0.163287	-0.310800	-0.445576	
owner	0.128210	0.758173	-0.264553	0.183667	
rent	0.794852	0.306834	0.344115	-0.042772	

	factor5

```
tot_pop                          -0.044915
disabled                         -0.089742
per_disabled                     -0.167317
unemployment_rate                 0.589986
tot_households_snap              -0.071236
households_snap                   0.045686
per_households_snap               0.133156
tot_pop_mobility                 -0.045705
less_10k                          0.094291
10k_15k                           0.005535
15k_25k                           0.141552
25k_35k                           0.172766
35k_50k                           0.184467
50k_75k                           0.196068
75k_100k                          0.186859
100k_150k                         0.134768
150k_200k                         0.068437
more_200k                         0.147563
med_income                        0.199084
mean_incom                        0.175744
non_citizen                       0.185982
pop_pov                          -0.041208
pop_below_                        0.062886
pcnt_pov                          0.148051
families_on_suplimental_income   -0.013240
families_on_social_security      -0.248735
insured                          -0.076096
per_insured                       0.492857
uninsured                         0.189029
per_uninsured                     0.333330
owner                            -0.308109
rent                              0.077882
```

[23]:
```
# get the column name of max values in every row
maxValueIndexObj = meaning.idxmax(axis=1)

print("Max values of row are at following columns :")
print(maxValueIndexObj)
```

```
Max values of row are at following columns :
tot_pop                          factor1
disabled                         factor1
per_disabled                     factor4
unemployment_rate                factor5
tot_households_snap              factor1
households_snap                  factor1
per_households_snap              factor1
```

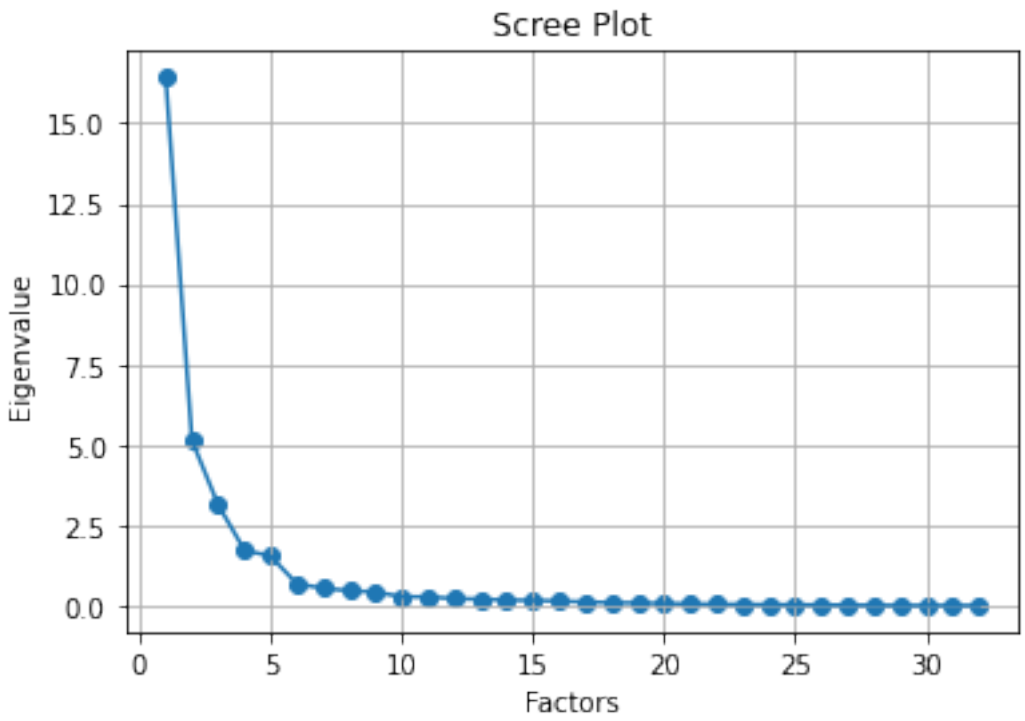
```
tot_pop_mobility                  factor1
less_10k                          factor1
10k_15k                           factor1
15k_25k                           factor1
25k_35k                           factor1
35k_50k                           factor1
50k_75k                           factor1
75k_100k                          factor2
100k_150k                         factor2
150k_200k                         factor2
more_200k                         factor3
med_income                        factor2
mean_incom                        factor3
non_citizen                       factor1
pop_pov                           factor1
pop_below_                        factor1
pcnt_pov                          factor1
families_on_suplimental_income    factor1
families_on_social_security       factor1
insured                           factor1
per_insured                       factor4
uninsured                         factor1
per_uninsured                     factor1
owner                             factor2
rent                              factor1
dtype: object
```

0.7 1 - Which variables go together using factor analysis? - RESPONSE

From this analysis we can see the breakdown of each of the 5 factors and what the variables are that correspond to each. Factor 1 encompasses total population, disabled, total households snap, total population mobility, non_citizen, population in poverty, families on social security, insured, owner, and rent. Factor 2 encompasses primarily percent disabled, households snap, percent households snap, less than 10k, between 10k and 15k, between 12k and 25k, between 25k and 305k. Factor 3 is smaller than the others being primarily comprised of number of people between 35k and 50k, 50k and 75k, and 75k and 100k. Factor 4 is comprised of the higher incomes and mean and median incomes. Strangely, factor 5 is the highest value for only one variable and that is per_uninsured.

[24]:
```
xvals = range(1,df[features].shape[1]+1)
```

[25]:
```
# Create scree plot using matplotlib
plt.scatter(xvals,ev)
plt.plot(xvals,ev)
plt.title('Scree Plot')
plt.xlabel('Factors')
plt.ylabel('Eigenvalue')
plt.grid()
plt.show()
```



[26]:
```
variance = fa.get_factor_variance()

vDF = pd.DataFrame(variance, columns = ['factor1',␣
↪'factor2','factor3','factor4','factor5'], index= ["SS Loadings", "Proportion␣
↪Var", "Cumulative Var"])
vDF
```

[26]:

	factor1	factor2	factor3	factor4	factor5
SS Loadings	16.310404	4.993170	2.925724	1.532068	1.315206
Proportion Var	0.509700	0.156037	0.091429	0.047877	0.041100
Cumulative Var	0.509700	0.665737	0.757166	0.805043	0.846143

[27]:
```
X = df[features]
y = df["4_16_2020_positive_rate"]
```

[28]:
```
model1 = LinearRegression()
model1.fit(X, y)
print("all data without train/test: ", model1.score(X, y))
```

```
all data without train/test:  0.5319451640160224
```

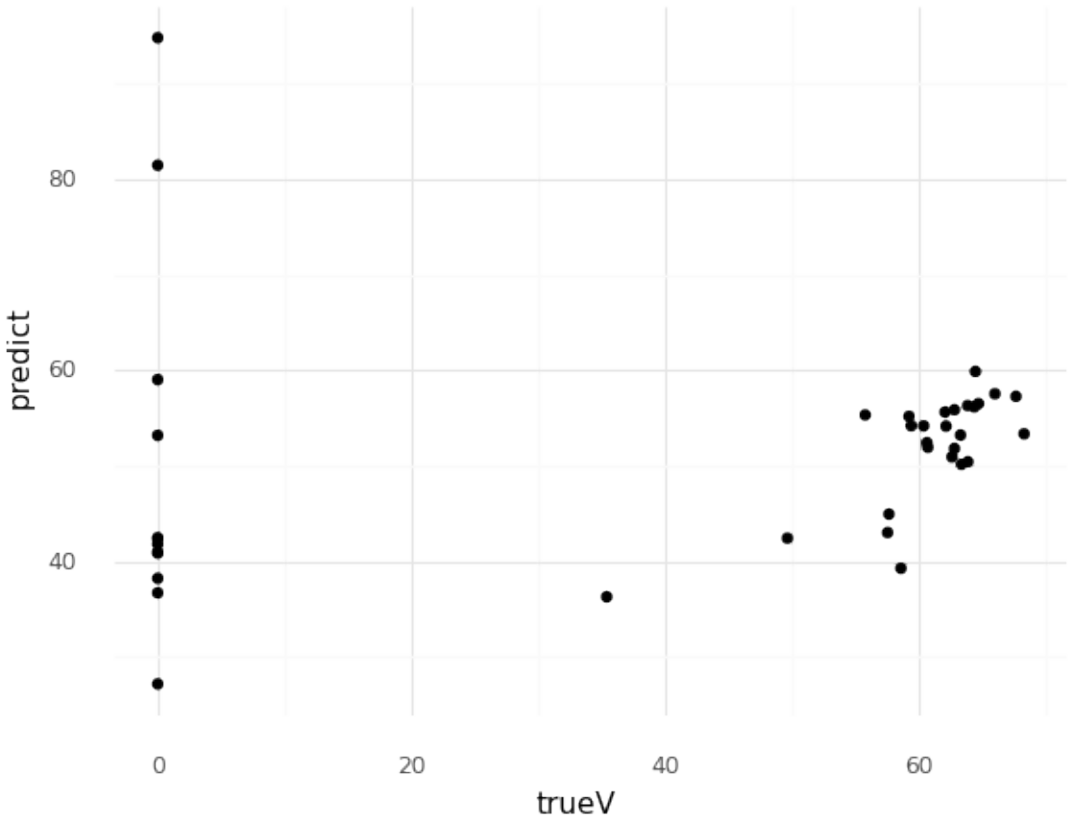
[29]:
```
#kfold split
kf = KFold(n_splits = 5)
```

```
model2 = LinearRegression()
scores = []

for train, test in kf.split(X,y):
    X_train = X.iloc[train]
    X_test = X.iloc[test]
    y_train = y[train]
    y_test = y[test]

    model2.fit(X_train,y_train)
    scores.append(model2.score(X_test,y_test))
    covid_pred = model2.predict(X_test)

print(scores)
print("mean all data score: ", np.mean(scores))
```

```
[-1.1439417235238447, 0.5767110159412059, -0.05024878858004245,
-2.864607537089714, -0.17382609197526921]
mean all data score:  -0.7311826250455329
```

[30]:
```
true_v_pred= pd.DataFrame({"predict": covid_pred, "trueV": y_test})
```

[31]:
```
ggplot(true_v_pred, aes(x = "trueV", y = "predict")) +geom_point()␣
↪+theme_minimal()
```



[31]:
```
<ggplot: (-9223371901819174608)>
```

[32]:
```
#features = ['tot_pop','disabled', 'per_disabled',␣
↪'unemployment_rate','tot_households_snap','households_snap','per_households_snap','tot_pop_

fact = fa.transform(df[features])
fact = pd.DataFrame(fact)

model3 = LinearRegression()
model3.fit(fact, df["4_16_2020_positive_rate"])
print("5 factors: ", model3.score(fact, df["4_16_2020_positive_rate"]))
```

```
5 factors:  0.3939553988823895
```

[33]:
```
ny.reset_index(inplace = True)
fact.reset_index(inplace = True)
ny_fact = pd.merge(ny,fact, on='index')
```

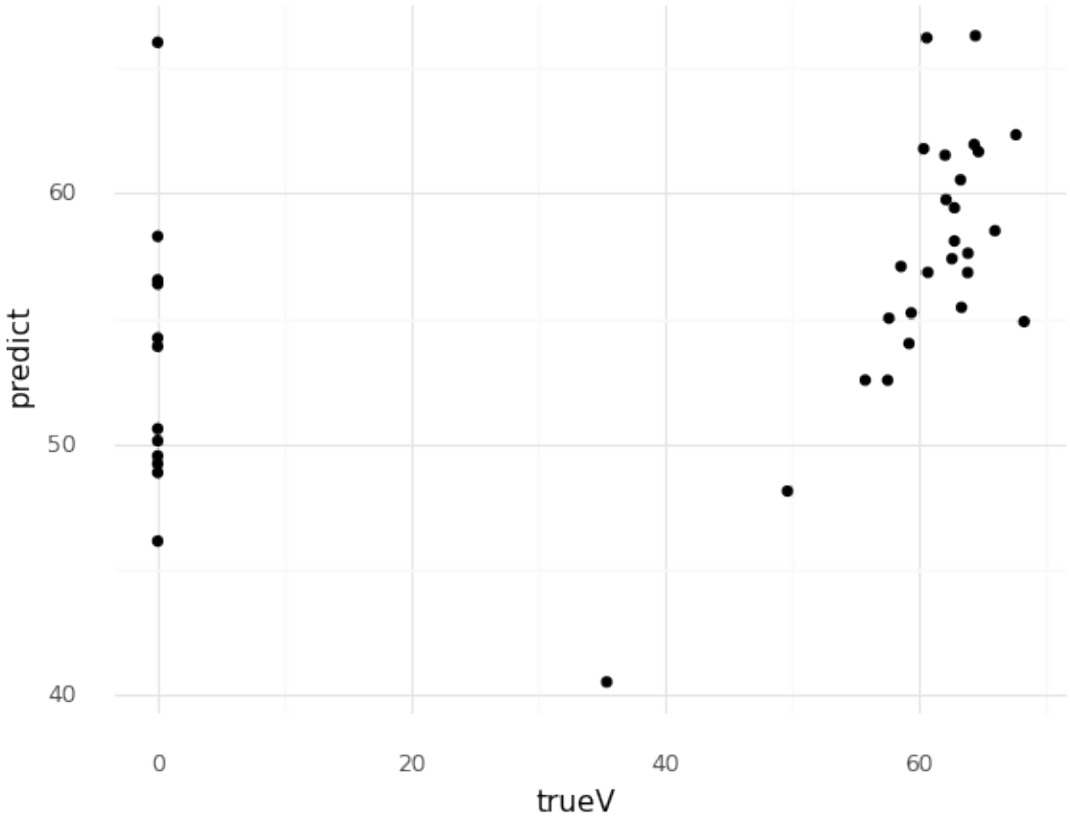
[34]:
```
X = fact
y = df["4_16_2020_positive_rate"]
```

```
#kfold split
kf = KFold(n_splits = 5)
model4 = LinearRegression()
scores = []

for train, test in kf.split(X,y):
    X_train = X.iloc[train]
    X_test = X.iloc[test]
    y_train = y[train]
    y_test = y[test]

    model4.fit(X_train,y_train)
    scores.append(model4.score(X_test,y_test))
    covid_pred = model4.predict(X_test)

print(scores)
print("mean all data score: ", np.mean(scores))
```

```
[-0.19505030628811995, 0.6648558744605146, 0.036738120075155756,
-1.6071637398472278, -0.12351987981802126]
mean all data score:  -0.24482798628353972
```

[35]:
```
true_v_pred = pd.DataFrame({"predict": covid_pred, "trueV": y_test})
```

[36]:
```
ggplot(true_v_pred, aes(x = "trueV", y = "predict")) +geom_point()␣
↪+theme_minimal()
```



[36]: <ggplot: (-9223371901819113992)>

0.8 2 - Which of these factors predict COVID cases? - RESPONSE

The model with all 33 socio-economic variables fit with a linear model did not predict the number of COVID cases on 4/16/2020 that well. Without a train - test split it had an r2 score of roughly 0.5. We can see that the model was fitting to noise however because when a fit was performed using a train test split the model had a negative r2 score. This can happen if the regression line that the model predicts is worse than simply using the mean value, which is why r squared values are usually non-negative.

The model with the 5 factored variables did not predict the COVID cases to any level os significance with an r2 score of 0.39 without using a kfold train test split. With a train test split the model had a score of -0.21.

From this analysis we know that there is some relationship between socio-economic variables and the number of COVID cases however, our model simply did not contain enough information to create an effective predictive model of COVID cases.

We can see this visually from looking at the graphs from the last test split of predicted values versus true values. We would hope to see a clear diagonal line but see a large scattering of data points instead.

0.9 Question: Does the strictness with which neighborhoods follow social distancing predict COVID cases?

0.10 3 - Did adherence to social distancing within boroughs affect covid rates

[37]:
```
import geopandas as gpd
import folium

from sklearn.cluster import AgglomerativeClustering, KMeans
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score
```

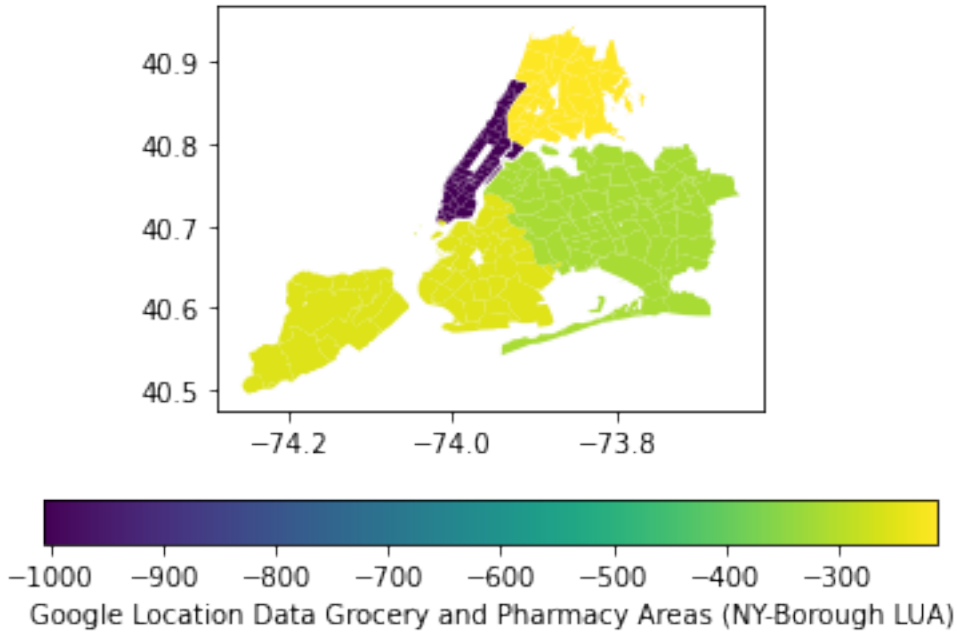
[38]:
```
ny_shp = gpd.read_file('data/spatial/bor_zip_codes.shp')
```

[39]:
```
ny = pd.merge(ny_shp, ny, how='inner', left_on='GEOID10', right_on='GEOID_ZIP')
ny.crs= {'init': 'epsg:4326'} #WGS84
```

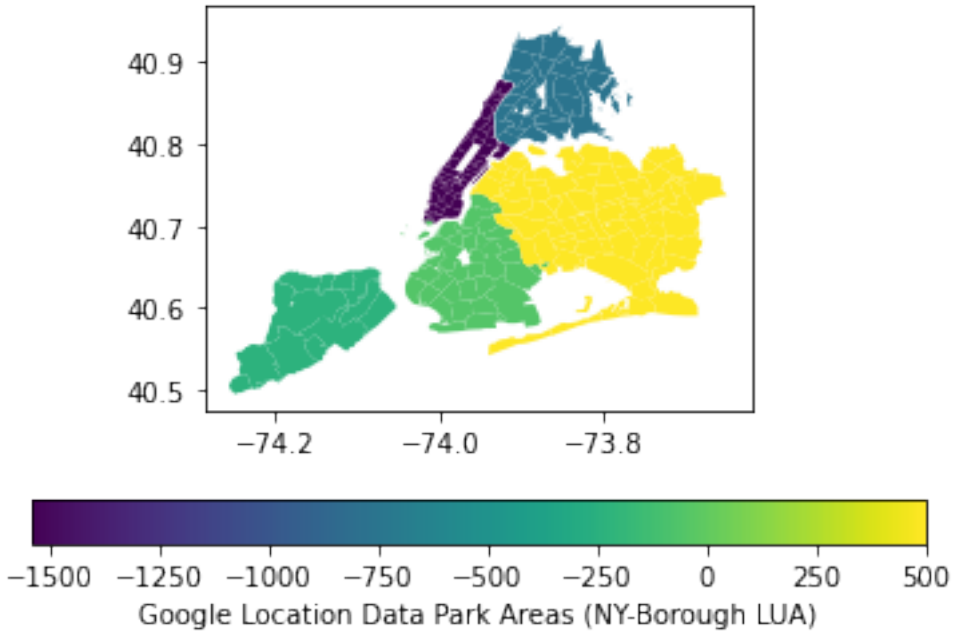
[40]:
```
map = folium.Map(location=[40.692416, -74.025393],zoom_start=10)
poly = ny.to_crs(epsg=4326).to_json()
geom = folium.features.GeoJson(poly)
map.add_child(geom)
map
```

[40]:
```
<folium.folium.Map at 0x1f70e9d3708>
```

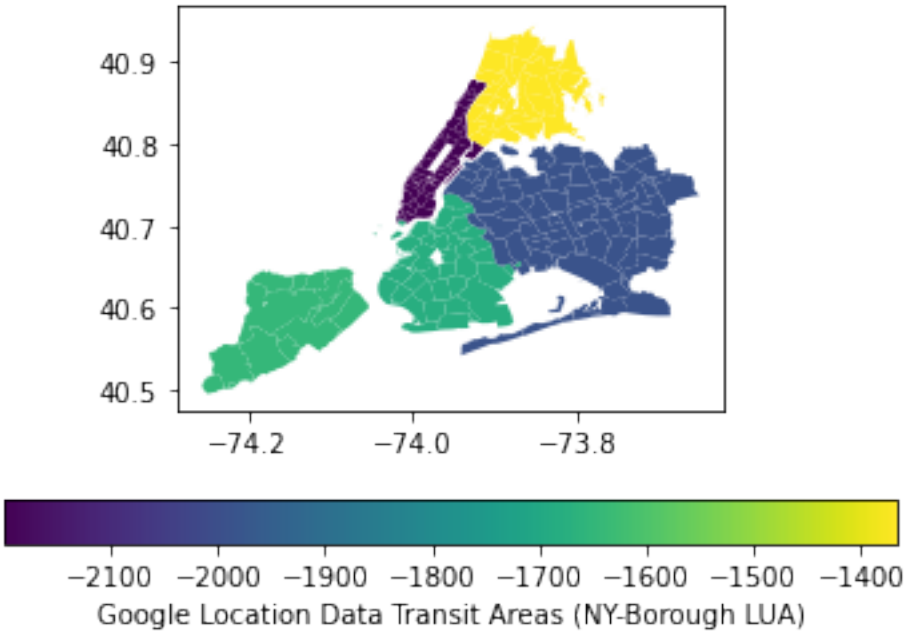
[41]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='grocery_and_pharmacy_percent_change_from_baseline_sum',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Google Location Data Grocery and Pharmacy Areas␣
 ↪(NY-Borough LUA)",
        'orientation': "horizontal"})
plt.show()
```



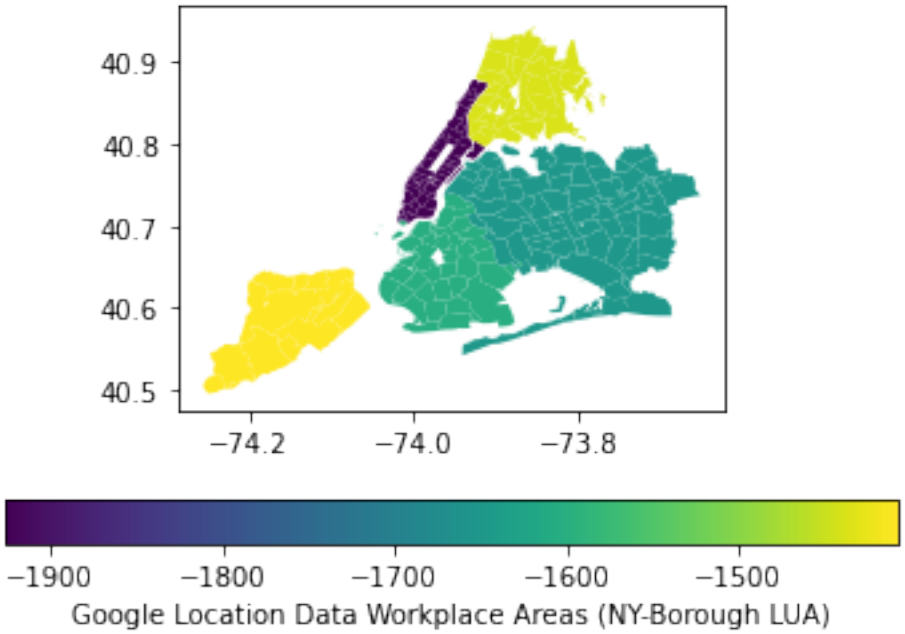
[42]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='parks_percent_change_from_baseline_sum',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Google Location Data Park Areas (NY-Borough␣
 ↪LUA)",
        'orientation': "horizontal"})
plt.show()
```



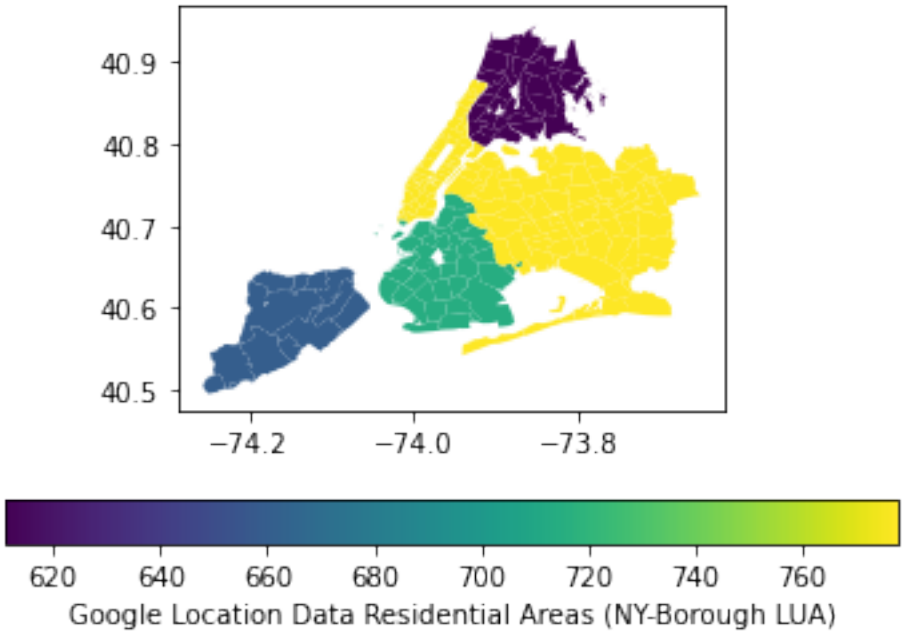
[43]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='transit_stations_percent_change_from_baseline_sum',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Google Location Data Transit Areas (NY-Borough␣
 ↪LUA)",
        'orientation': "horizontal"})
plt.show()
```



[44]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='workplaces_percent_change_from_baseline_sum',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Google Location Data Workplace Areas (NY-Borough␣
 ↪LUA)",
        'orientation': "horizontal"})
plt.show()
```



[45]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='residential_percent_change_from_baseline_sum',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Google Location Data Residential Areas␣
 ↪(NY-Borough LUA)",
        'orientation': "horizontal"})
plt.show()
```



[46]:
```
predictors = ["grocery_and_pharmacy_percent_change_from_baseline_sum",␣
 ↪"parks_percent_change_from_baseline_sum",␣
 ↪"transit_stations_percent_change_from_baseline_sum",␣
 ↪"workplaces_percent_change_from_baseline_sum",␣
 ↪"residential_percent_change_from_baseline_sum"]

# Not standardizing because vars are on the same scale
X_train, X_test, y_train, y_test = train_test_split(ny[predictors],␣
 ↪ny["4_16_2020_positive"], test_size=0.2,random_state=123)
```

[47]:
```
mod = sm.OLS(y_train,X_train)
fit = mod.fit()
```

[48]:
```
fit.summary()
```

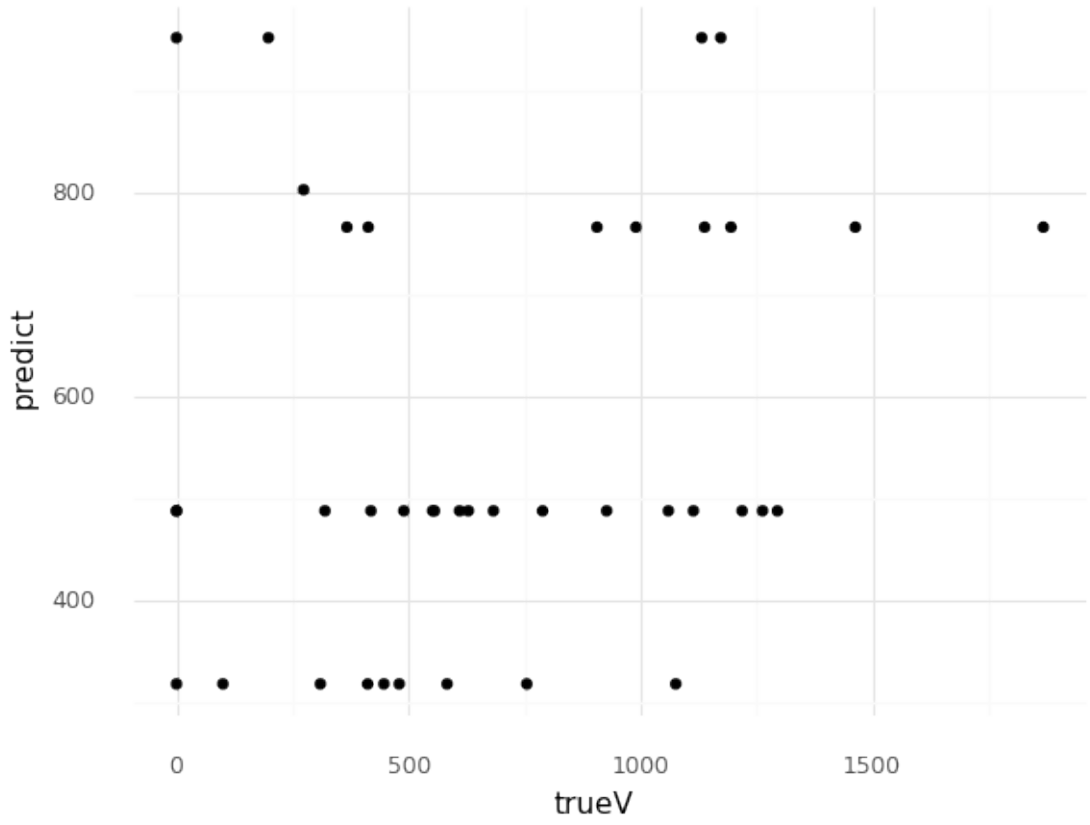
[48]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:     4_16_2020_positive   R-squared:                       0.208
Model:                            OLS   Adj. R-squared:                  0.187
Method:                 Least Squares   F-statistic:                     9.840
Date:                Tue, 19 May 2020   Prob (F-statistic):           4.26e-07
Time:                        09:18:43   Log-Likelihood:                -1162.7
No. Observations:                 155   AIC:                             2335.
Df Residuals:                     150   BIC:                             2351.
```

```
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
==========================================
                                                                        coef    std err
t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
------------------------------------------
grocery_and_pharmacy_percent_change_from_baseline_sum    -0.1878      3.876
-0.048      0.961      -7.847       7.471
parks_percent_change_from_baseline_sum                   -0.4909      0.224
-2.188      0.030      -0.934      -0.048
transit_stations_percent_change_from_baseline_sum         2.2182      4.842
0.458      0.648      -7.349      11.785
workplaces_percent_change_from_baseline_sum               2.9630      5.505
0.538      0.591      -7.914      13.840
residential_percent_change_from_baseline_sum             12.8128     22.155
0.578      0.564     -30.963      56.589
==============================================================================
Omnibus:                       45.945   Durbin-Watson:                   1.881
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              127.431
Skew:                           1.174   Prob(JB):                     2.13e-28
Kurtosis:                       6.771   Cond. No.                     1.77e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.77e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[49]:
```
test_preds = fit.predict(X_test)
```

[50]:
```
true_v_pred = pd.DataFrame({"predict": test_preds, "trueV": y_test})
```

[51]:
```
ggplot(true_v_pred, aes(x = "trueV", y = "predict")) +geom_point()␣
↪+theme_minimal()
```



[51]: <ggplot: (-9223371901815386612)>

[52]:
```
print('testing r2 is:', r2_score(test_preds, y_test))
```

```
testing r2 is: -4.2873808705790095
```

[53]:
```
#creating a subset of the ny dataframe containing socio-economic variables
df = ny[['tot_pop','disabled', 'per_disabled',␣
 ↪'unemployment_rate','tot_households_snap','households_snap','per_households_snap','tot_pop_
 ↪'4_16_2020_positive_rate',␣
 ↪"grocery_and_pharmacy_percent_change_from_baseline_sum",␣
 ↪"parks_percent_change_from_baseline_sum",␣
 ↪"transit_stations_percent_change_from_baseline_sum",␣
 ↪"workplaces_percent_change_from_baseline_sum",␣
 ↪"residential_percent_change_from_baseline_sum"]].copy()
```

```
features = ['tot_pop','disabled', 'per_disabled',␣
 ↪'unemployment_rate','tot_households_snap','households_snap','per_households_snap','tot_pop_
 ↪"parks_percent_change_from_baseline_sum",␣
 ↪"transit_stations_percent_change_from_baseline_sum",␣
 ↪"workplaces_percent_change_from_baseline_sum",␣
 ↪"residential_percent_change_from_baseline_sum"]
```

[54]:
```
#standardize variables
zScore = StandardScaler() #standardize variables makes them easier with the math
zScore.fit(df[features])
df[features] = zScore.transform(df[features])
```

[55]:
```
predictors = ['tot_pop','disabled', 'per_disabled',␣
 ↪'unemployment_rate','tot_households_snap','households_snap','per_households_snap','tot_pop_
 ↪"parks_percent_change_from_baseline_sum",␣
 ↪"transit_stations_percent_change_from_baseline_sum",␣
 ↪"workplaces_percent_change_from_baseline_sum",␣
 ↪"residential_percent_change_from_baseline_sum"]

# Not standardizing because vars are on the same scale
X_train, X_test, y_train, y_test = train_test_split(ny[predictors],␣
 ↪ny["4_16_2020_positive"], test_size=0.2,random_state=123)
```

[56]:
```
mod = sm.OLS(y_train,X_train)
fit = mod.fit()
```

[57]:
```
fit.summary()
```

[57]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:     4_16_2020_positive   R-squared:                       0.872
Model:                            OLS   Adj. R-squared:                  0.835
Method:                 Least Squares   F-statistic:                     23.94
Date:                Tue, 19 May 2020   Prob (F-statistic):           4.27e-39
Time:                        09:18:43   Log-Likelihood:                -1021.8
No. Observations:                 155   AIC:                             2114.
Df Residuals:                     120   BIC:                             2220.
Df Model:                          34
Covariance Type:            nonrobust
==============================================================================
                                                 coef    std err
t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
------------------------------------------
```

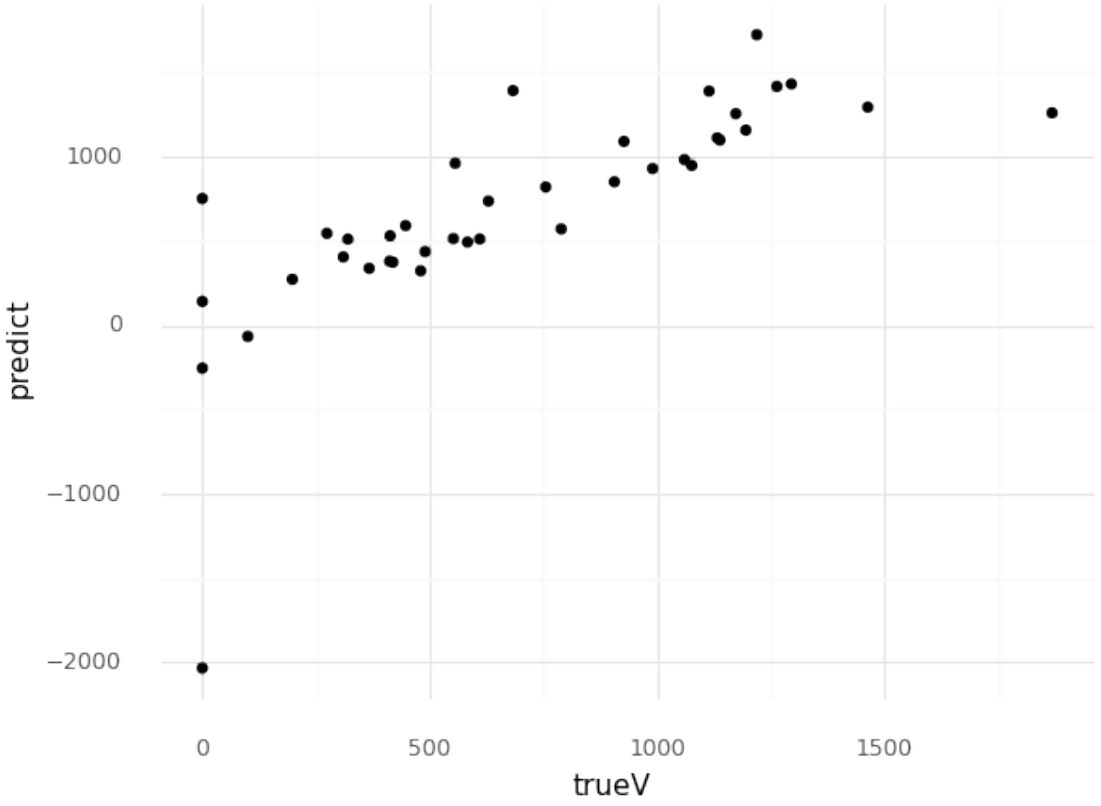
```
tot_pop                                                 0.1578      0.115
1.369      0.174      -0.071       0.386
disabled                                                0.0706      0.032
2.218      0.028       0.008       0.134
per_disabled                                           -3.2202     11.848
-0.272      0.786     -26.678      20.237
unemployment_rate                                      11.4882      5.528
2.078      0.040       0.543      22.434
tot_households_snap                                    -0.0210      0.010
-2.182      0.031      -0.040      -0.002
households_snap                                        -0.0254      0.047
-0.543      0.588      -0.118       0.067
per_households_snap                                    -5.0388      7.821
-0.644      0.521     -20.524      10.447
tot_pop_mobility                                       -0.0297      0.114
-0.260      0.795      -0.256       0.197
less_10k                                               14.8453     12.485
1.189      0.237      -9.873      39.564
10k_15k                                                15.0613     14.126
1.066      0.288     -12.908      43.030
15k_25k                                                 4.4207     12.975
0.341      0.734     -21.268      30.110
25k_35k                                                -0.3622     13.675
-0.026      0.979     -27.438      26.713
35k_50k                                                14.2725     11.519
1.239      0.218      -8.535      37.080
50k_75k                                                -3.2912     10.332
-0.319      0.751     -23.748      17.166
75k_100k                                               -3.7376     11.064
-0.338      0.736     -25.643      18.168
100k_150k                                              -6.8432      8.286
-0.826      0.411     -23.249       9.563
150k_200k                                             -24.2118     13.627
-1.777      0.078     -51.193       2.770
more_200k                                              -9.8542     10.093
-0.976      0.331     -29.838      10.129
med_income                                              0.0007      0.001
0.492      0.623      -0.002       0.003
mean_incom                                              0.0009      0.002
0.454      0.651      -0.003       0.005
non_citizen                                             0.0088      0.011
0.803      0.423      -0.013       0.030
pop_pov                                                 0.0254      0.020
1.256      0.212      -0.015       0.065
pop_below_                                             -0.0043      0.018
-0.237      0.813      -0.040       0.032
pcnt_pov                                               -7.4058     11.216
```

```
-0.660      0.510     -29.612      14.801
families_on_suplimental_income                          -0.0961      0.106
-0.903      0.368      -0.307       0.115
families_on_social_security                              0.0166      0.052
0.318      0.751      -0.087       0.120
insured                                                 -0.1363      0.052
-2.623      0.010      -0.239      -0.033
per_insured                                             18.1914      7.405
2.457      0.015       3.531      32.852
uninsured                                               -0.0944      0.054
-1.739      0.085      -0.202       0.013
per_uninsured                                          -18.1403      7.455
-2.433      0.016     -32.902      -3.379
owner                                                   -0.0117      0.011
-1.038      0.302      -0.034       0.011
rent                                                    -0.0093      0.007
-1.416      0.159      -0.022       0.004
grocery_and_pharmacy_percent_change_from_baseline_sum    0.5070      2.577
0.197      0.844      -4.595       5.609
parks_percent_change_from_baseline_sum                  -0.0888      0.256
-0.347      0.729      -0.596       0.418
transit_stations_percent_change_from_baseline_sum       -1.8153      3.138
-0.579      0.564      -8.028       4.397
workplaces_percent_change_from_baseline_sum              1.2615      3.496
0.361      0.719      -5.659       8.182
residential_percent_change_from_baseline_sum            -4.3965     14.247
-0.309      0.758     -32.604      23.811
==============================================================================
Omnibus:                       29.236   Durbin-Watson:                   1.965
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               66.173
Skew:                          -0.796   Prob(JB):                     4.27e-15
Kurtosis:                       5.776   Cond. No.                     1.25e+16
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 2.76e-20. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

[58]:
```
test_preds = fit.predict(X_test)
```

[59]:
```
true_v_pred = pd.DataFrame({"predict": test_preds, "trueV": y_test})
```

[60]:
```
ggplot(true_v_pred, aes(x = "trueV", y = "predict")) +geom_point()␣
↪+theme_minimal()
```



[60]: <ggplot: (-9223371901815331848)>

[61]:
```
print('testing r2 is:', r2_score(test_preds, y_test))
```

```
testing r2 is: 0.5706548247618044
```

0.11 3 - Did adherence to social distancing within boroughs affect covid rates - RESPONSE

Given the generalized google location tracking data we have, there is unclear results as to exactly how our social distancing metric affected our model. Some variables resulted in having a negative relationship with covid positive results as we had thought, we see this with the negative coefficient values in transit, parks, and residential areas. However positive relationships in grocery, pharmacy, and workplace areas. All in all though, we see p values $> .05$ signaling that although we have coeficient magnitude estimate, they are not statistically significant, this is likely due to the fact that the location data we have is set to the borough level unit of analysis and not the zipcode level.

Overall we do believe that social distancing has reduced covid transmission within communities, however we cannot confidently estimate figures detailing how much it has helped.

0.12 Question: What does a disadvantaged group look like in terms of COVID cases?

0.13 4 - Would a clustering model help us group zipcodes together

0.14 5 - How well do these clustered groups predict COVID

[62]:
```
from sklearn.cluster import AgglomerativeClustering, KMeans
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score

from sklearn import neighbors
from sklearn.model_selection import GridSearchCV
```

[63]:
```
features = ['per_female','per_male','less_10k', '10k_15k', '15k_25k',
       '25k_35k', '35k_50k', '50k_75k', '75k_100k', '100k_150k',
       '150k_200k', 'more_200k', 'med_income', 'mean_incom','pcnt_pov',␣
↪'families_on_suplimental_income',
       'families_on_social_security', 'per_white', 'per_black', 'per_native',
       'per_asian', 'per_hawaiian', 'per_other', 'per_two_or']
```

[64]:
```
X = ny[features]
```

[65]:
```
z = StandardScaler()
X[features] = z.fit_transform(X)
```

[66]:
```
n_components = [3,4,5,6,7,8,9,10]
Xdf = X
```

[67]:
```
sils = []

for n in n_components:
    gmm = GaussianMixture(n_components = n)
    gmm.fit(X)
    colName = str(n) + "GM_assign"
    clusters = gmm.predict(X)

    Xdf[colName] = clusters

    sils.append(silhouette_score(X, clusters))

print(sils)
```

```
[0.21229312078826607, 0.19844583682229602, 0.17308828746609303,
0.2606467832042456, 0.3010197108698031, 0.3833408484359511, 0.43187557096002077,
0.4874160344191789]
```

[68]:
```
sils = []

for n in n_components:
    hac = AgglomerativeClustering(n_clusters = n,
                                  affinity = "euclidean",
                                 linkage = "ward")
    hac.fit(X)

    colName = str(n) + "HAC_assign"
    clusters = hac.labels_

    Xdf[colName] = clusters

    sils.append(silhouette_score(X, clusters))

print(sils)
```

```
[0.32562313563951467, 0.39395047374346037, 0.44327594774593476,
0.4958388620437945, 0.5247086772079443, 0.5446422139531493, 0.5681473684623364,
0.5878053221454038]
```

[69]:
```
sils = []

for n in n_components:
    km = KMeans(n_clusters = n)
    km.fit(X)
    colName = str(n) + "KM_assign"
    clusters = km.predict(X)

    Xdf[colName] = clusters

    sils.append(silhouette_score(X, clusters))

print(sils)
```

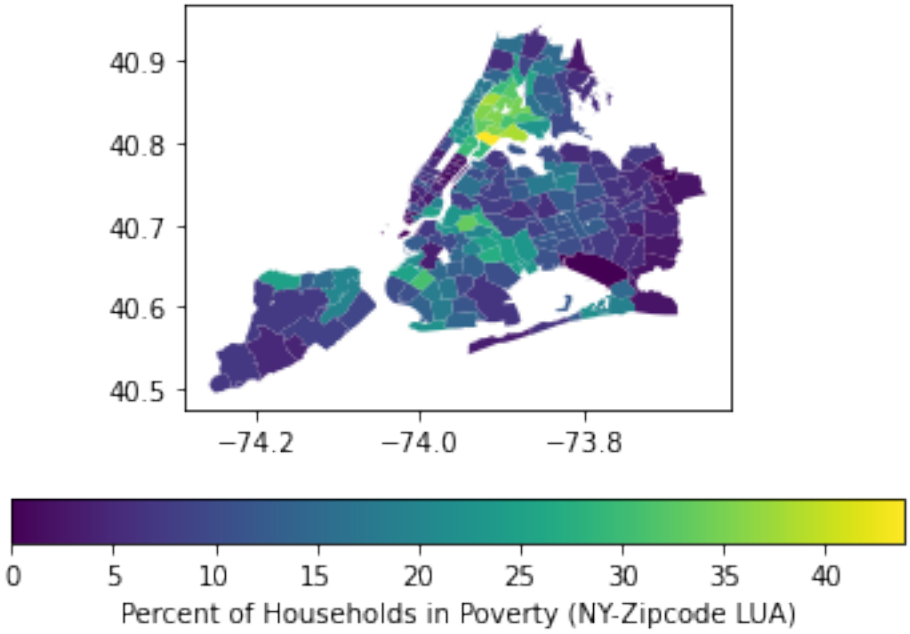
```
[0.3693770481561607, 0.4371774323771695, 0.48688366186949505,
0.5310619781099695, 0.5834174009529649, 0.6040701919449619, 0.6241074151297916,
0.6406526391947232]
```

[70]:
```
groups = ['5GM_assign', '5KM_assign', '5HAC_assign','9GM_assign', '9KM_assign',␣
↪'9HAC_assign', 'index']
```

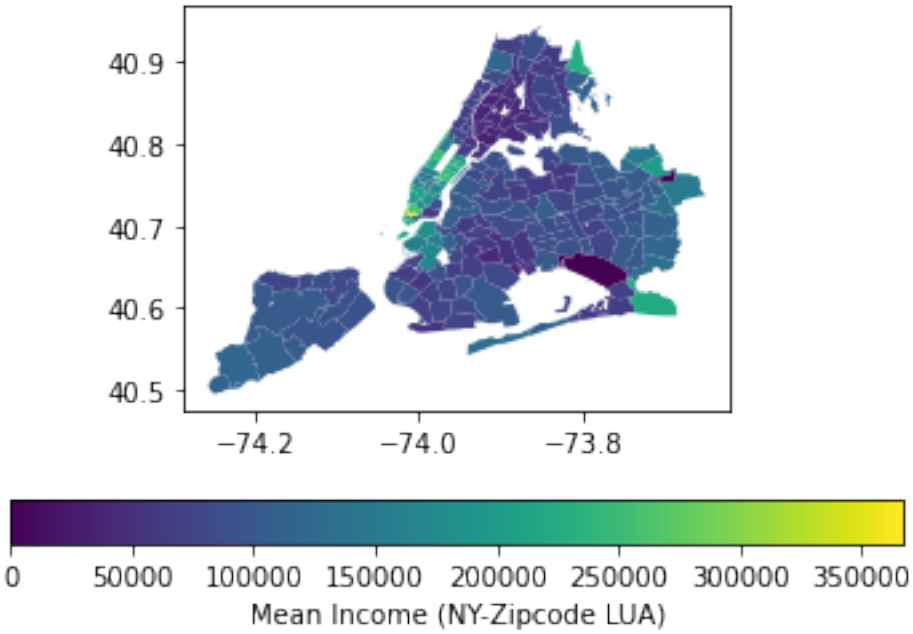
[71]:
```
Xdf.reset_index(inplace = True)
```

[72]:
```
ny_clusters = pd.merge(ny,Xdf[groups], on='index')
```

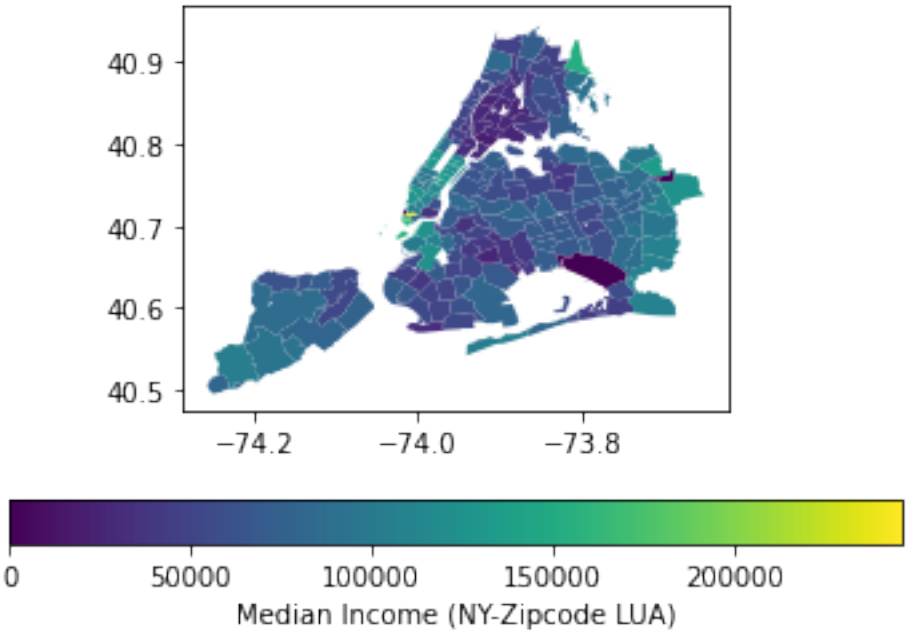
[73]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='pcnt_pov',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Percent of Households in Poverty (NY-Zipcode␣
 ↪LUA)",
        'orientation': "horizontal"})
plt.show()
```



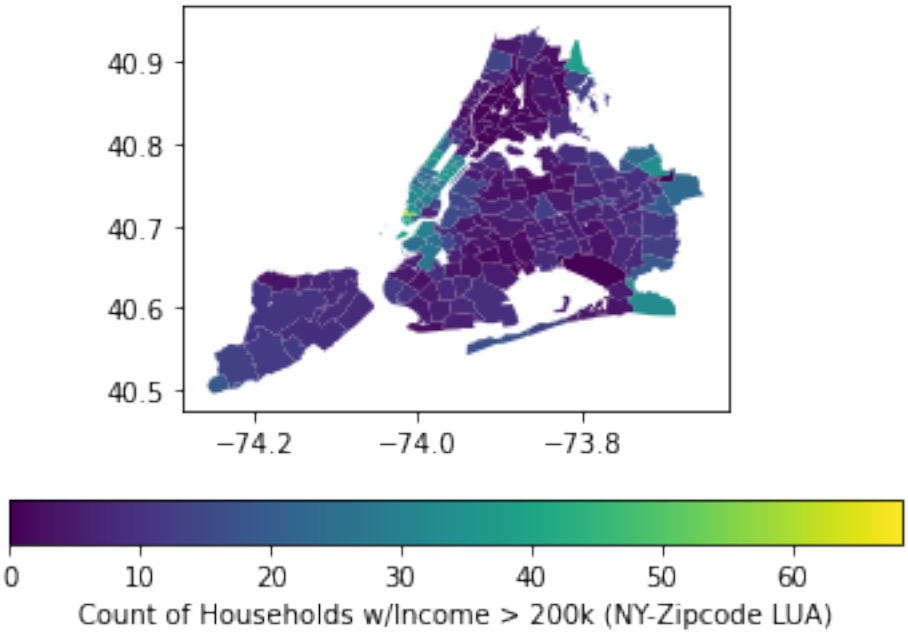
[74]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='mean_incom',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Mean Income (NY-Zipcode LUA)",
        'orientation': "horizontal"})
plt.show()
```



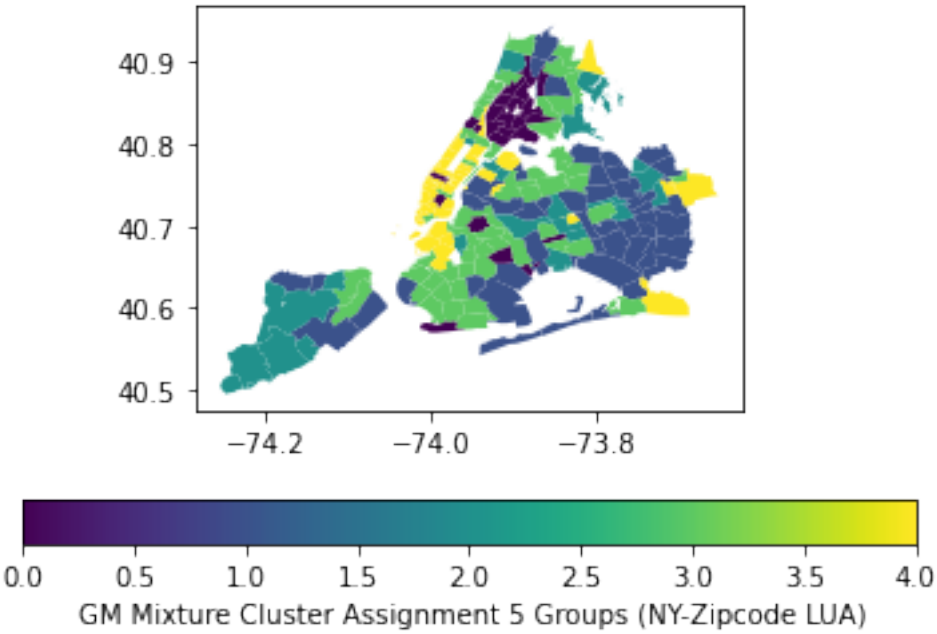
[75]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='med_income',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Median Income (NY-Zipcode LUA)",
        'orientation': "horizontal"})
plt.show()
```



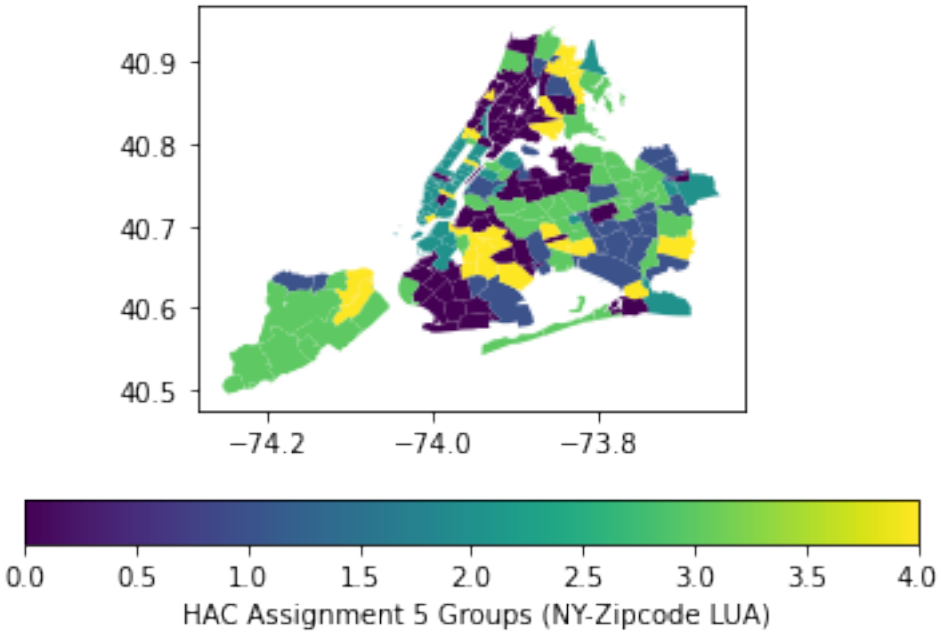
[76]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='more_200k',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Count of Households w/Income > 200k (NY-Zipcode␣
 ↪LUA)",
        'orientation': "horizontal"})
plt.show()
```



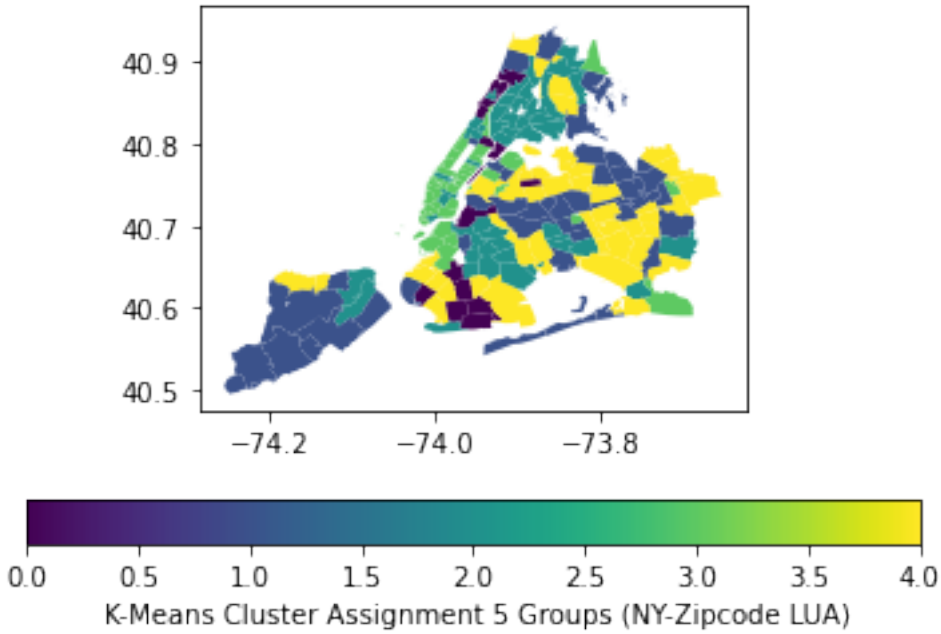
[77]:
```
fig, ax = plt.subplots(1, 1)

ny_clusters.plot(column='5GM_assign',
        ax=ax,
        legend=True,
        legend_kwds={'label': " GM Mixture Cluster Assignment 5 Groups␣
 ↪(NY-Zipcode LUA)",
        'orientation': "horizontal"})
plt.show()
```



[78]:
```
fig, ax = plt.subplots(1, 1)

ny_clusters.plot(column='5HAC_assign',
        ax=ax,
        legend=True,
        legend_kwds={'label': " HAC Assignment 5 Groups (NY-Zipcode LUA)",
        'orientation': "horizontal"})
plt.show()
```



[79]:
```
fig, ax = plt.subplots(1, 1)

ny_clusters.plot(column='5KM_assign',
        ax=ax,
        legend=True,
        legend_kwds={'label': " K-Means Cluster Assignment 5 Groups (NY-Zipcode␣
 ↪LUA)",
        'orientation': "horizontal"})
plt.show()
```



[80]:
```
features = ['per_female','per_male','less_10k', '10k_15k', '15k_25k',
       '25k_35k', '35k_50k', '50k_75k', '75k_100k', '100k_150k',
       '150k_200k', 'more_200k', 'med_income', 'mean_incom','pcnt_pov',␣
 ↪'families_on_suplimental_income',
       'families_on_social_security', 'per_white', 'per_black', 'per_native',
       'per_asian', 'per_hawaiian', 'per_other', 'per_two_or']

X = ny[features]
y = ny["per_infected"]

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3)

z = StandardScaler()
X_train = z.fit_transform(X_train)
X_test = z.transform(X_test)

params = {'n_neighbors': range(1,30)}

knn = neighbors.KNeighborsRegressor()

grid = GridSearchCV(knn, params, cv=5)

knnmod = grid.fit(X_train, y_train)
```

[81]:
```
knnmod.best_estimator_.get_params()["n_neighbors"]
```

[81]: 9

[82]:
```
knnmod.best_score_
```

[82]: 0.25079625698597496

[83]:
```
knnmod.score(X_test,y_test)
```

[83]: 0.24921388526964117

[84]:
```
test_pred = knnmod.predict(X_test)
```

[85]:
```
print('testing r2 is:', r2_score(test_pred, y_test))
```

```
testing r2 is: -2.0159723012188384
```

0.15 4 - Would a clustering model help us group zipcodes together - RESPONSE

We were unable to intuitively determine what unsupervised clustering method would perform the best and create the most distinct clusters determined by silhouette score. We performed 3 clustering models to compare which performed best. The models we chose were Hierarchical Agglomerative Clustering, Gaussian Mixture Modeling, and Kmeans. The clustering models were able to help us group zipcodes together based on gender, financial, and minority composition. In creating the models, we were cautious to avoid overfitting the data by using too many clusters to we capped the number of clusters that we tested for HAC, GMM, and Kmeans at 9. From 3-9 clusters, we selected 5 clusters and 9 clusters to analyze more in depth with each of the 3 models. We did this both graphically and analytically by looking at their silhouette score. The model that performed the best was Kmeans followed by HAC and GMM. Unfortunately, there is still a possibility that as we increased the number of clusters that the silhouette score would have also increased but this is perhaps not representative of our data. It simply means that we are creating more clusters so they will be more distict thereby creating a better silhouette score. We would need to spend more time analyzing the clusters to determine what number of clusters would most logically make sense. We will be passing along the cluster data created from the models to another reasearcher who is a part of the Community Health Equity Lab and is experienced at determining the relationships and commonalities in the large variety of variables that we have. We hope that this anaylsis is useful in determining what commonalities exist between certain zipcodes.

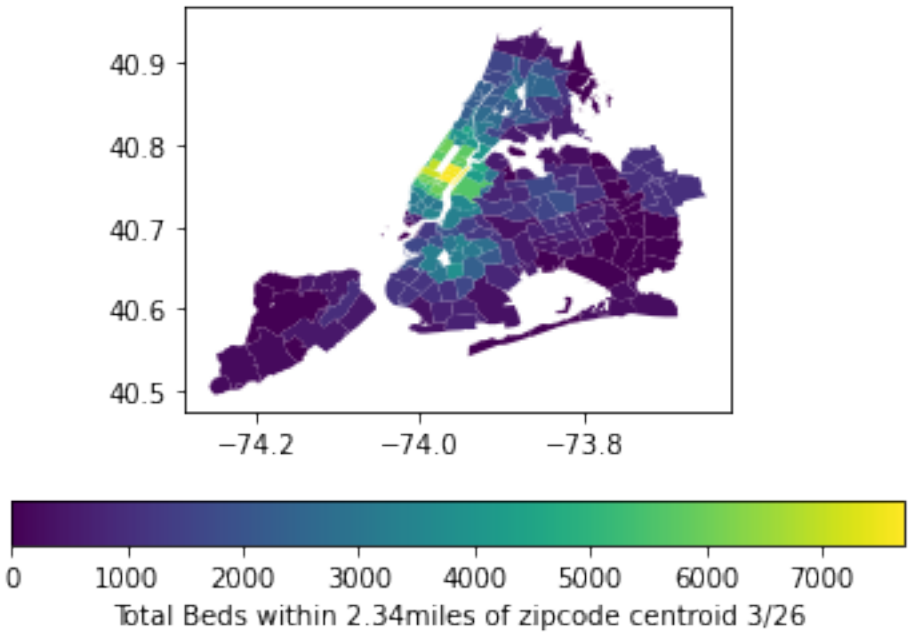
0.16 5 - How well do clustered groups predict COVID - RESPONSE

Using a KNN regressor model we used clustering to predict covid cases. The clustering method did not prove to predict covid cases well to any level of signigicance. A grid search selected 6 as the ideal number of neighbors. The model's best score was 0.36 and the score of the test set was 0.12. We do not know if the model itself was too simple to predict the number of covid cases or if the data that we have available is simply not significant enough to achieve a high prediction score.

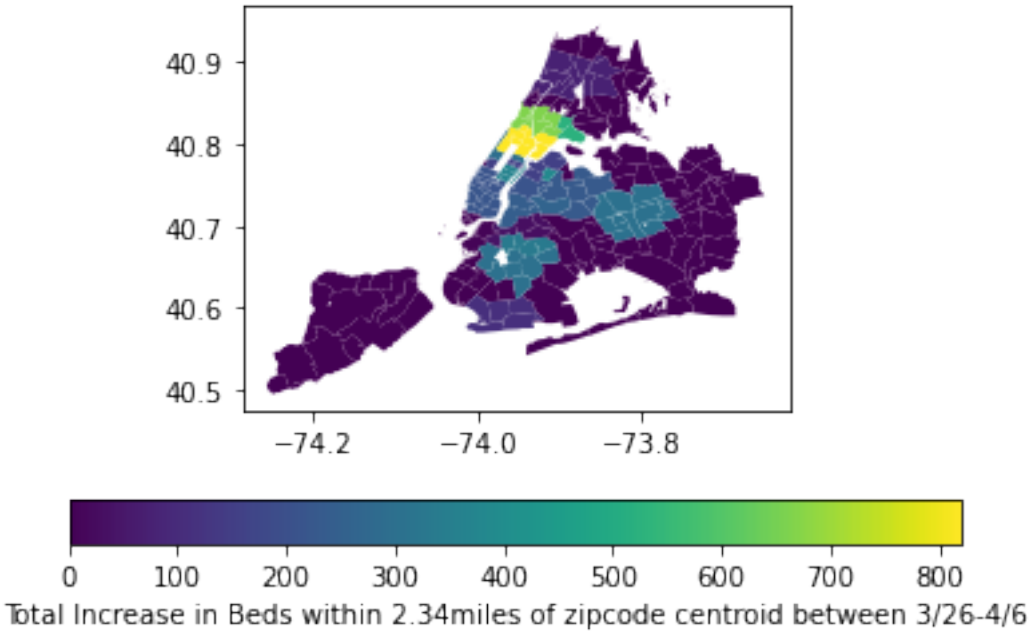
0.17 Question: Are beds being allocated to areas most affected by covid?

0.18 6 - What variables contribute to allocation of resources

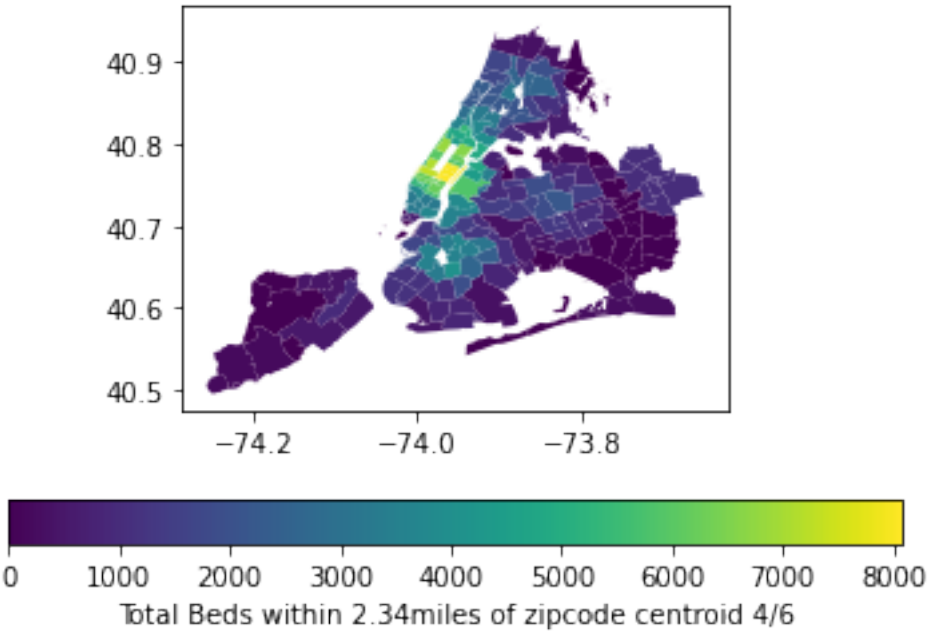
[86]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='3_26_bb_beds',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Total Beds within 2.34miles of zipcode centroid␣
 ↪3/26",
        'orientation': "horizontal"})
plt.show()
```



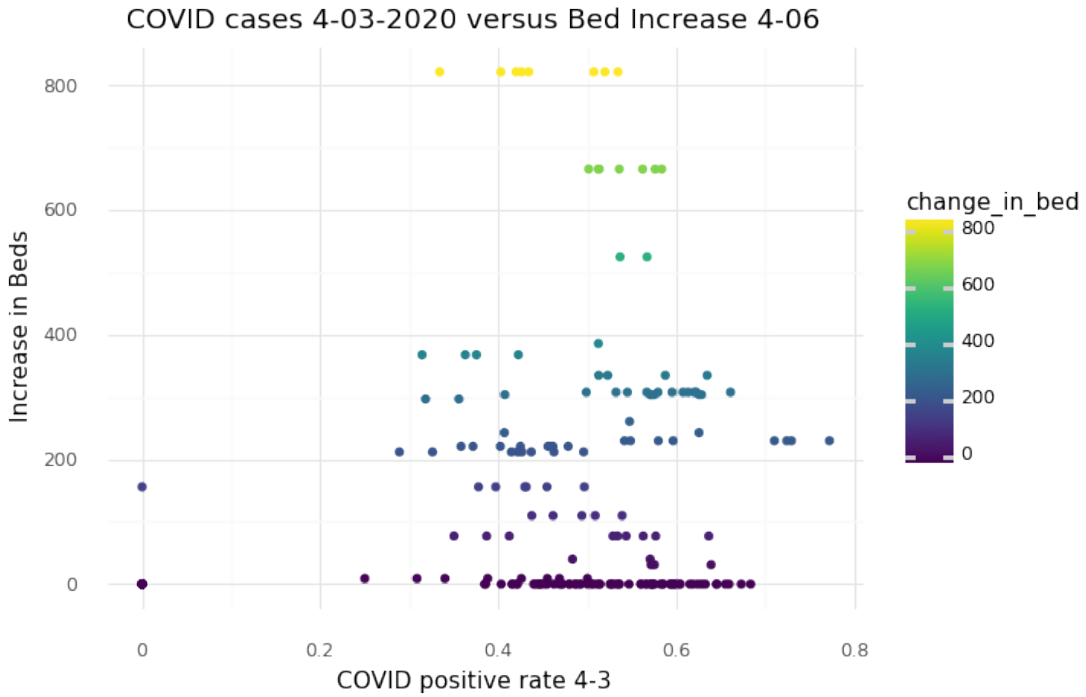
[87]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='change_in_bed',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Total Increase in Beds within 2.34miles of␣
 ↪zipcode centroid between 3/26-4/6",
        'orientation': "horizontal"})
plt.show()
```



[212]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='4_7_bb_beds',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Total Beds within 2.34miles of zipcode centroid␣
 ↪4/6",
        'orientation': "horizontal"})
plt.show()
```

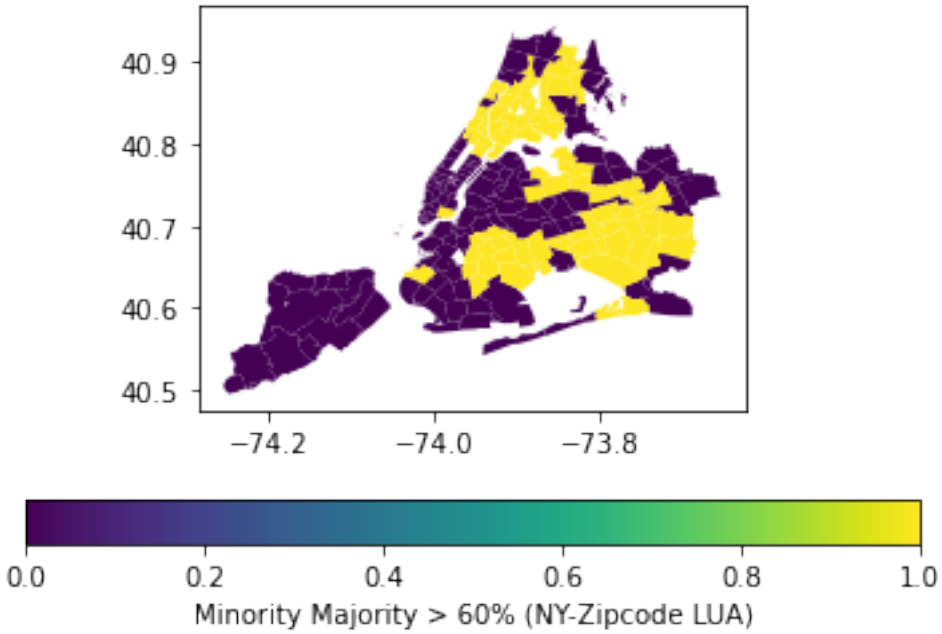


[89]:
```
(ggplot(ny, aes(x='4_3_2020_positive_rate', y='change_in_bed', color=␣
↪'change_in_bed'))+ geom_point()+theme_minimal()+ ggtitle("COVID cases␣
↪4-03-2020 versus Bed Increase 4-06")+ xlab("COVID positive rate␣
↪4-3")+ylab("Increase in Beds"))
```

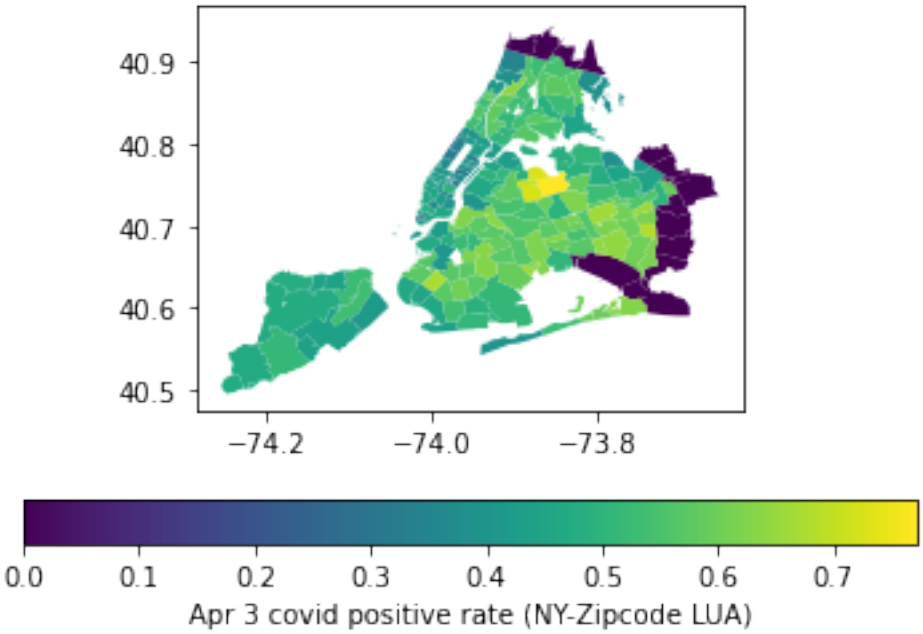


[89]: <ggplot: (-9223371901813721908)>

[90]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='minority-majority-60',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Minority Majority > 60% (NY-Zipcode LUA)",
        'orientation': "horizontal"})
plt.show()
```



[211]:
```
fig, ax = plt.subplots(1, 1)

ny.plot(column='4_3_2020_positive_rate',
        ax=ax,
        legend=True,
        legend_kwds={'label': "Apr 3 covid positive rate (NY-Zipcode LUA)",
        'orientation': "horizontal"})
plt.show()
```



[92]:
```
#creating a subset of the ny dataframe containing socio-economic variables
df = ny[['per_disabled',␣
 ↪'unemployment_rate','per_households_snap','less_10k','10k_15k','15k_25k','25k_35k','35k_50k
 ↪"4_3_2020_positive_rate",'change_in_bed']
].copy()

features = ['per_disabled',␣
 ↪'unemployment_rate','per_households_snap','less_10k','10k_15k','15k_25k','25k_35k','35k_50k
 ↪"4_3_2020_positive_rate"]
```

[93]:
```
#standardize variables

zScore = StandardScaler() #standardize variables makes them easier with the math
zScore.fit(df[features])
df[features] = zScore.transform(df[features])
```

[94]:
```
X_train, X_test, y_train, y_test = train_test_split(df[features],␣
 ↪df["change_in_bed"], test_size=0.2,random_state=123)
```

[95]:
```
mod = sm.OLS(y_train,X_train)
fit = mod.fit()
```

[96]:
```
fit.summary()
```

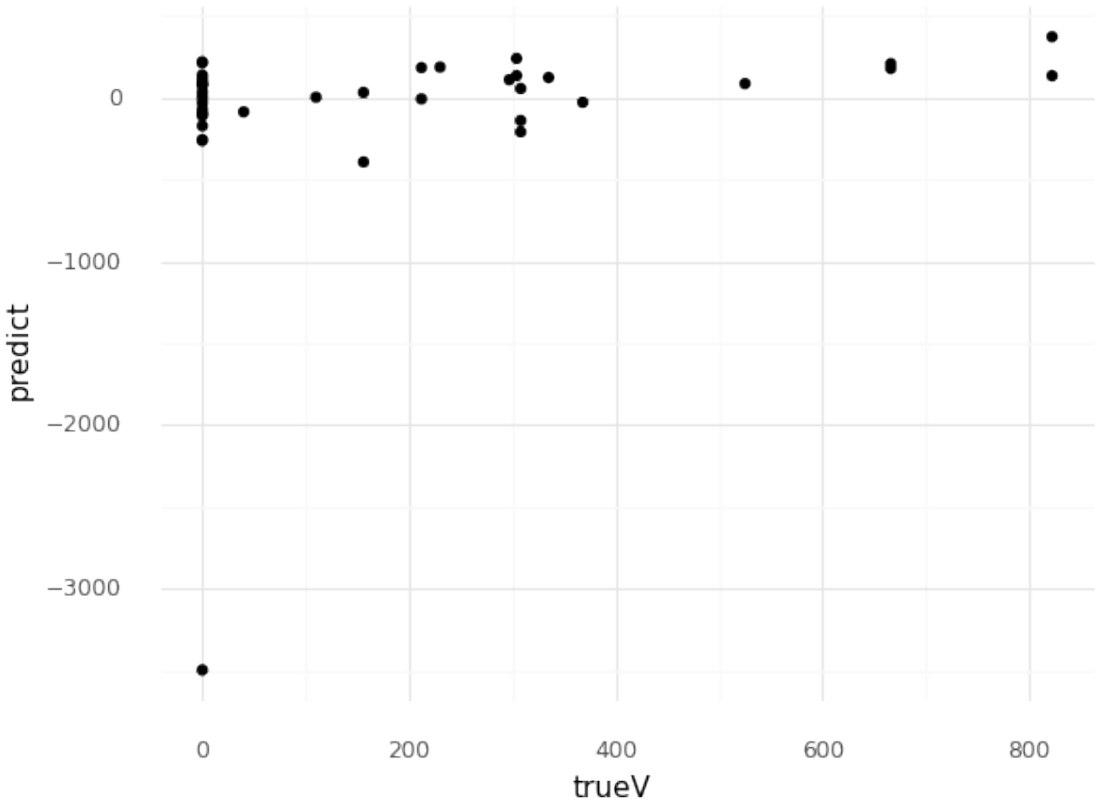
[96]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:          change_in_bed   R-squared (uncentered):
0.272
Model:                            OLS   Adj. R-squared (uncentered):
0.138
Method:                 Least Squares   F-statistic:
2.037
Date:                Tue, 19 May 2020   Prob (F-statistic):
0.00602
Time:                        09:18:47   Log-Likelihood:
-1055.1
No. Observations:                 155   AIC:
2158.
Df Residuals:                     131   BIC:
2231.
Df Model:                          24
Covariance Type:            nonrobust
==============================================================================
==================
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
------------------
per_disabled                     4.9207     45.996      0.107      0.915
-86.070      95.912
unemployment_rate              105.4346     47.518      2.219      0.028
11.432     199.437
per_households_snap           -144.1985     83.574     -1.725      0.087
-309.527      21.130
less_10k                       122.5094     57.244      2.140      0.034
9.267     235.751
10k_15k                        113.9009     55.860      2.039      0.043
3.397     224.405
15k_25k                          5.9978     56.487      0.106      0.916
-105.747     117.743
25k_35k                         17.5012     44.729      0.391      0.696
-70.984     105.986
35k_50k                          3.3643     46.247      0.073      0.942
-88.123      94.851
50k_75k                         -7.2284     44.643     -0.162      0.872
-95.542      81.085
75k_100k                       -45.7765     36.633     -1.250      0.214
-118.245      26.692
```

```
100k_150k                          51.2340     46.552      1.101      0.273
-40.858     143.326
150k_200k                         -41.3604     58.125     -0.712      0.478
-156.346      73.625
more_200k                        -262.7719    140.013     -1.877      0.063
-539.752      14.208
med_income                        -19.5878     55.588     -0.352      0.725
-129.555      90.379
mean_incom                        215.2056    126.268      1.704      0.091
-34.583     464.994
non_citizen                        97.5429     50.723      1.923      0.057
-2.799     197.884
pcnt_pov                           41.0190     90.440      0.454      0.651
-137.893     219.931
families_on_suplimental_income    -97.9964     72.027     -1.361      0.176
-240.483      44.490
families_on_social_security       -13.8095     74.311     -0.186      0.853
-160.814     133.195
per_insured                       205.1401     96.365      2.129      0.035
14.506     395.774
owner                              -4.7470     54.246     -0.088      0.930
-112.058     102.564
rent                               29.1182     48.853      0.596      0.552
-67.524     125.761
minority-majority-50               34.4424     27.626      1.247      0.215
-20.209      89.093
4_3_2020_positive_rate            -10.4909     25.982     -0.404      0.687
-61.890      40.908
==============================================================================
Omnibus:                       12.745   Durbin-Watson:                   1.379
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               14.072
Skew:                           0.602   Prob(JB):                     0.000879
Kurtosis:                       3.853   Cond. No.                         32.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

[97]:
```
test_preds = fit.predict(X_test)
```

[98]:
```
true_v_pred = pd.DataFrame({"predict": test_preds, "trueV": y_test})
```

[99]:
```
ggplot(true_v_pred, aes(x = "trueV", y = "predict")) +geom_point()␣
↪+theme_minimal()
```



[99]: <ggplot: (-9223371901813452040)>

[100]:
```
print('testing r2 is:', r2_score(test_preds, y_test))
```

```
testing r2 is: -0.15185170223281252
```

0.19 6 - What variables contribute to allocation of resources - RESPONSE

Although poor R2 and pvalues, variables that seem to have the strongest positive relationships with bed allocations are percentage of zipcodes with insurance, mean income, and count of households that make between 10k and 15k

variables that seem to have the strongest negative relationships with bed allocations are count of households that make between >200k, percentage of zipcodes with households on foodstamps, and percentage of zipcodes with households on supplemental income.

Given the poor results we conclude that a linear model is unlikely able to statistically describe allocation of beds throughout New York. However, from visually observing the graphs it is evident that manhattan received a large number of beds even though other boroughs (specifically queens and areas in brooklyn) had higher test-positive rates. Data related to number of patients at hospitals would be very useful in the situation, but without more info, other methods of examination are required

Instead a possible solution could be to use other testing methods such as a kruskall wallace h test between various communities.

0.20 Polynomial Regression because why not

[101]:
```
import seaborn as sns
import statsmodels.formula.api as smf
from scipy.optimize import curve_fit
```

[142]:
```
ny_fact = ny_fact.rename(columns={"4_16_2020_positive": "positive",
                                  0:"fact_0",
                                  1:"fact_1",
                                  2:"fact_2",
                                  3:"fact_3",
                                  4:"fact_4",
                                  '2_ppl_fam':'two',
                                  '3_to_4_ppl':'threefour',
                                  '5_to_6_ppl':'fivesix',
                                  'GT_7_ppl_f':'sevenmore'})
ny_fact.head()
```

[142]:

	index	GEOID_ZIP	ALAND10	AWATER10	tot_pop	white	per_white	black	\
0	0	10065	984654	0	28109	24285	86.4	619	
1	1	10069	249050	0	5085	3155	62.0	148	
2	2	10075	477137	0	21556	18396	85.3	677	
3	3	10128	1206191	0	59256	47167	79.6	2182	
4	4	10280	297253	38409	9384	7360	78.4	184	

	per_black	native	per_native	asian	per_asian	nativehawaiian	\
0	2.2	37	0.1	2666	9.5	0	
1	2.9	0	0.0	1558	30.6	0	
2	3.1	225	1.0	1047	4.9	0	
3	3.7	0	0.0	5844	9.9	0	
4	2.0	0	0.0	1474	15.7	0	

	per_hawaiian	other	per_other	two_or_mor	per_two_or	male	female	\
0	0.0	154	0.5	348	1.2	12248	15861	
1	0.0	0	0.0	224	4.4	2354	2731	
2	0.0	902	4.2	309	1.4	10096	11460	
3	0.0	1666	2.8	2397	4.0	25338	33918	
4	0.0	77	0.8	289	3.1	4477	4907	

	under_5	per_under_5	5_to_9	per_5_to_9	10_to_14	per_10_to_14	15_to_19	\
0	1650	5.9	1000	3.6	916	3.3	356	
1	368	7.2	416	8.2	274	5.4	95	
2	1041	4.8	1252	5.8	736	3.4	763	
3	3204	5.4	2138	3.6	1911	3.2	1691	

```
4      711          7.6     628          6.7       454           4.8       192

   per_15_to_19  20_to_24  per_20_to_24  25_to_29  per_25_to_29  30_to_34  \
0           1.3      1785           6.4      2671           9.5      3176
1           1.9       235           4.6       427           8.4       594
2           3.5       778           3.6      1400           6.5      2022
3           2.9      2644           4.5      6825          11.5      7216
4           2.0       239           2.5       523           5.6      1374

   per_30_to_34  35_to_39  per_35_to_39  40_to_44  per_40_to_44  45_to_49  \
0          11.3      2558           9.1      1607           5.7      1570
1          11.7       639          12.6       508          10.0       374
2           9.4      1643           7.6      1148           5.3      1453
3          12.2      4349           7.3      3849           6.5      4103
4          14.6       945          10.1      1021          10.9       771

   per_45_to_49  50_to_54  per_50_to_54  55_to_59  per_55_to_59  60_to_64  \
0           5.6      1796           6.4      1460           5.2      1522
1           7.4       378           7.4       235           4.6       197
2           6.7      1244           5.8      1118           5.2      1131
3           6.9      3845           6.5      2989           5.0      3494
4           8.2       763           8.1       530           5.6       375

   per_60_to_64  65_to_69  per_65_to_69  70_to_74  per_70_to_74  75_to_79  \
0           5.4      1388           4.9      1486           5.3      1159
1           3.9        61           1.2        87           1.7       109
2           5.2      1561           7.2      1276           5.9      1169
3           5.9      3461           5.8      2656           4.5      1980
4           4.0       370           3.9       254           2.7        79

   per_75_to_79  80_to_84  per_80_to_89  80_and_over  per_80_and_over  \
0           4.1       954           3.4         1055              3.8
1           2.1         0           0.0           88              1.7
2           5.4       570           2.6         1251              5.8
3           3.3      1557           2.6         1344              2.3
4           0.8       115           1.2           40              0.4

   disabled  per_disabled  unemployment_rate  tot_households_snap  \
0      1894           6.8               70.2                14844
1       182           3.6               74.7                 2552
2      1689           8.0               68.1                11035
3      4825           8.1               72.2                30691
4       326           3.5               83.7                 4560

   households_snap  per_households_snap  tot_pop_mobility  same_house  \
0              414                  2.8             27635       22612
1               73                  2.9              5025        3790
```

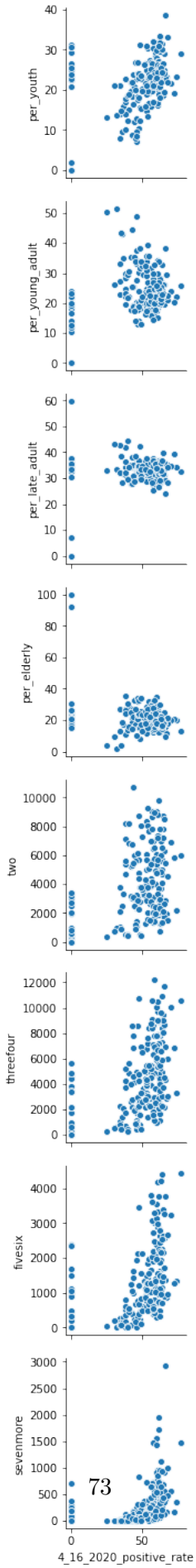
```
2              217                  2.0             21381       18455
3             1506                  4.9             58419       48263
4               97                  2.1              9234        7638

   moved_within_1yr  less_10k  10k_15k  15k_25k  25k_35k  35k_50k  50k_75k  \
0              2247       5.0      2.7      4.0      2.7      6.2     10.6
1               859       8.7      3.2      0.0      9.8      5.7      4.2
2              1508       1.9      0.8      4.8      6.4      5.0     12.8
3              4921       4.5      2.2      6.5      3.6      5.6     12.4
4               779       3.4      1.1      0.5      1.8      3.3      8.0

   75k_100k  100k_150k  150k_200k  more_200k  med_income  mean_incom  \
0       8.8       14.4        9.8       35.8      127375      242978
1      14.7       11.4       11.6       30.9      110625      225183
2      10.6       17.9        9.6       30.3      137146      233358
3      11.2       14.6       10.0       29.4      114010      196844
4       6.8       18.6       16.1       40.5      169844      224631

   speaks_only_english  naturalized  non_citizen  pop_pov  pop_below_  \
0                19681         3045         3306    27963        1922
1                 2922          331         1207     5085         522
2                13727         3315         2372    21155        1075
3                39143         7054         6535    58980        3686
4                 5618         1372         1539     9384         336

   pcnt_pov  families_on_suplimental_income  families_on_social_security  \
0       3.0                             109                         1683
1       7.3                              25                           88
2       4.5                             129                         1401
3       3.4                             500                         3513
4       1.1                               0                          216

    two  threefour  fivesix  sevenmore  insured  per_insured  uninsured  \
0  3797       2267      275         13    27299         97.3        744
1   517        662       27          0     4992         98.2         93
2  3133       1765      236          0    20715         97.9        440
3  8171       5192      530          3    57424         96.9       1820
4  1046       1009      130         16     9027         96.2        357

   per_uninsured  owner   rent  per_minority  minority-majority-50  \
0            2.7   5365   9479          13.5                     0
1            1.8    675   1877          37.9                     0
2            2.1   4933   6102          14.6                     0
3            3.1  10129  20562          20.4                     0
4            3.8   1233   3327          21.6                     0

   minority-majority-60  minority-majority-70  pop_density  per_infected  \
```

```
0                    0                    0     0.028547      0.008289
1                    0                    0     0.020418      0.007866
2                    0                    0     0.045178      0.013639
3                    0                    0     0.049127      0.007763
4                    0                    0     0.031569      0.003410

   change_in_bed  per_female  per_male  per_youth  per_young_adult  \
0            368    0.564268  0.435732       14.1             27.2
1              0    0.537070  0.462930       22.7             24.7
2            156    0.531639  0.468361       17.5             19.5
3            297    0.572398  0.427602       15.1             28.2
4              9    0.522911  0.477089       21.1             22.7

   per_late_adult  per_elderly  positive  4_16_2020_tests  \
0            32.0         26.9       233              682
1            42.0         10.6        40              105
2            30.6         32.1       294              631
3            32.2         24.4       460             1207
4            42.9         13.0        32               92

   4_16_2020_positive_rate  4_2_2020_positive  4_2_2020_tests  \
0                    34.16                121             385
1                    38.10                 24              57
2                    46.59                160             371
3                    38.11                212             596
4                    34.78                 17              50

   4_2_2020_positive_rate  4_3_2020_positive  4_3_2020_tests  \
0                0.314286                121             385
1                0.421053                 24              57
2                0.431267                160             371
3                0.355705                212             596
4                0.340000                 17              50

   4_3_2020_positive_rate  4_7_2020_positive  4_7_2020_tests  \
0                0.314286                171             492
1                0.421053                 29              72
2                0.431267                204             452
3                0.355705                281             737
4                0.340000                 20              61

   4_7_2020_positive_rate  4_8_2020_positive  4_8_2020_tests  \
0                   34.76                187             544
1                   40.28                 32              81
2                   45.13                232             501
3                   38.13                317             834
4                   32.79                 23              73
```

```
   4_8_2020_positive_rate  hospital_count  3_26_bb_beds  4_2_bb_beds  \
0                   34.38              14          7625         7993
1                   39.51               9          4922         4922
2                   46.31              15          7573         7729
3                   38.01              13          5936         6233
4                   31.51               4           679          688

   4_7_bb_beds  licensed_beds  staffed_beds  ICU_beds  adult_icu_beds  \
0         7993           6493          5519       745             745
1         4922           4093          3765       410             410
2         7729           6277          5678       788             788
3         6233           4713          4177       514             514
4          688            713           420        35              35

   pediatrics_icu_beds  bed_utilization_rate  potential_increase_bed_capacity  \
0                  402              6.647990                              974
1                  273              4.741397                              328
2                  412              7.253486                              599
3                  337              6.464822                              536
4                   28              1.790607                              293

   avg_ventilator_use  GEOID10    borough sub_region_2  \
0                 181    10065  Manhattan    Manhattan
1                 138    10069  Manhattan    Manhattan
2                 198    10075  Manhattan    Manhattan
3                 147    10128  Manhattan    Manhattan
4                  12    10280  Manhattan    Manhattan

   retail_and_recreation_percent_change_from_baseline_sum  \
0                                              -2296
1                                              -2296
2                                              -2296
3                                              -2296
4                                              -2296

   grocery_and_pharmacy_percent_change_from_baseline_sum  \
0                                              -1006
1                                              -1006
2                                              -1006
3                                              -1006
4                                              -1006

   parks_percent_change_from_baseline_sum  \
0                                   -1544
1                                   -1544
2                                   -1544
```

```
3                                    -1544
4                                    -1544

   transit_stations_percent_change_from_baseline_sum  \
0                                              -2199
1                                              -2199
2                                              -2199
3                                              -2199
4                                              -2199

   workplaces_percent_change_from_baseline_sum  \
0                                        -1926
1                                        -1926
2                                        -1926
3                                        -1926
4                                        -1926

   residential_percent_change_from_baseline_sum    fact_0    fact_1    fact_2  \
0                                           777 -2.653452 -4.713664  0.175708
1                                           777 -2.895952 -5.621049 -0.359911
2                                           777 -0.770275  1.813769  0.573820
3                                           777 -2.322450 -4.833800 -0.026389
4                                           777 -1.875722  0.288210  1.709647

     fact_3    fact_4
0 -2.235328  3.857765
1 -1.954594  4.112185
2  0.434803 -0.740402
3 -2.280653  4.690372
4 -1.122793  0.994076
```

[138]:
```
pp = sns.pairplot(data=ny_fact,
                  y_vars=['per_youth', 'per_young_adult', 'per_late_adult',␣
 ↪'per_elderly', 'two','threefour','fivesix', 'sevenmore'],
                  x_vars=['4_16_2020_positive_rate'])
```


per_youth
per_young_adult
per_late_adult
per_elderly
two
threefour
fivesix
sevenmore
4_16_2020_positive_rate

[104]:
```
#features = ['fact_0', 'fact_1', 'fact_2', 'fact_3', 'fact_4']
```

[167]:
```
train, test = train_test_split(ny_fact, test_size=0.2,random_state=123)
```

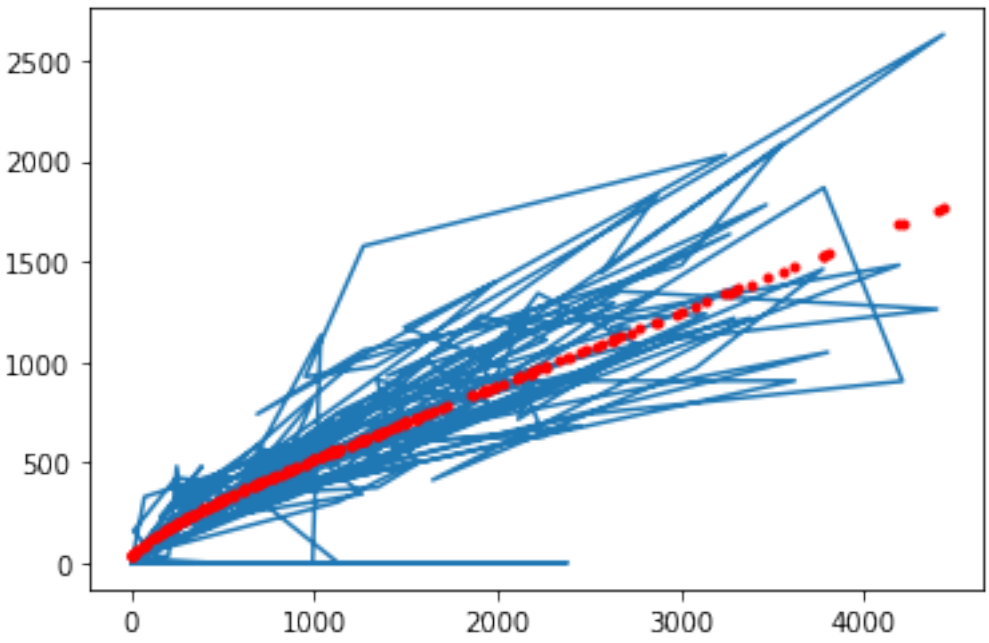
[168]:
```
def ferg(x,c0,c1,c2,c3):
    return c0+c1*x-c2*np.exp(-c3*x)
```

[173]:
```
guess = [0,.015,200,.01]
tp = ny_fact['fivesix'].values
pc = ny_fact['positive'].values

c,cov = curve_fit(ferg,tp,pc,guess)
print(c)
```

```
[1.54139731e+02 3.64875593e-01 1.14932250e+02 3.40684988e-03]
```

[174]:
```
n = len(ny_fact['fivesix'])
y = np.empty(n)

for i in range(n):
    y[i] = ferg(ny_fact['fivesix'][i],c[0],c[1],c[2],c[3])

plt.plot(ny_fact['fivesix'], ny_fact['positive'])
plt.plot(ny_fact['fivesix'], y, 'r.')
```

[174]:
```
[<matplotlib.lines.Line2D at 0x1f713058f88>]
```



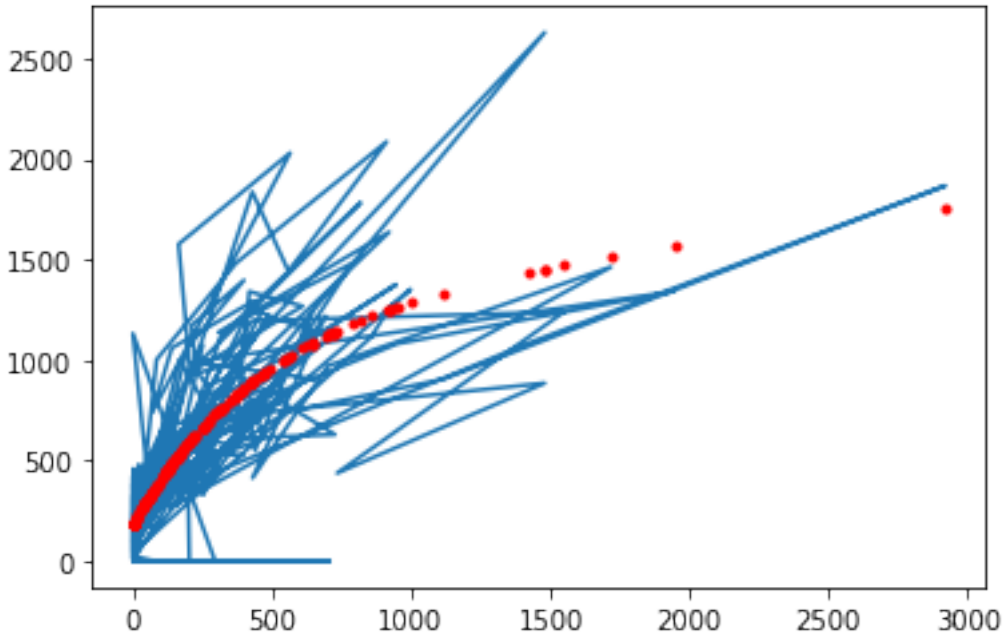
[176]:
```
def rocky(x,c0,c1,c2,c3):
    return c0+c1*x-c2*np.exp(-c3*x)
```

[177]:
```
guess = [0,.015,200,.01]
tp = ny_fact['sevenmore'].values
pc = ny_fact['positive'].values

c,cov = curve_fit(ferg,tp,pc,guess)
print(c)
```

```
[1.21909129e+03 1.85433417e-01 1.03267111e+03 2.17136626e-03]
```

[178]:
```
n = len(ny_fact['sevenmore'])
y = np.empty(n)

for i in range(n):
    y[i] = ferg(ny_fact['sevenmore'][i],c[0],c[1],c[2],c[3])

plt.plot(ny_fact['sevenmore'], ny_fact['positive'])
plt.plot(ny_fact['sevenmore'], y, 'r.')
```

[178]:
```
[<matplotlib.lines.Line2D at 0x1f712ebee88>]
```



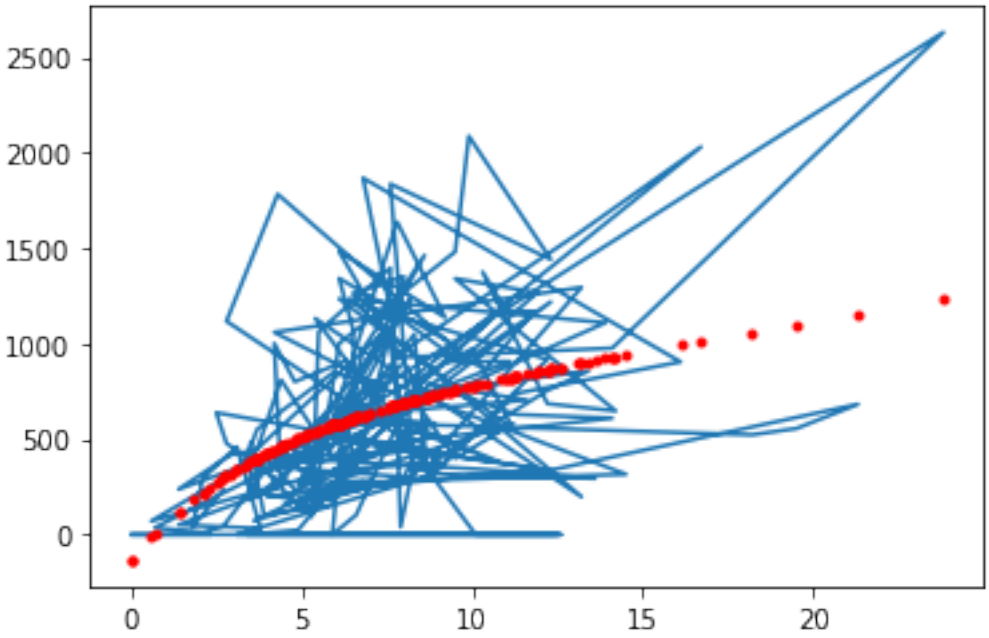
[201]:
```
def yams(x,c0,c1,c2,c3):
    return c0+c1*x-c2*np.exp(-c3*x)
```

[206]:
```
guess = [0,.015,200,.01]
tp = ny_fact['per_uninsured'].values
pc = ny_fact['positive'].values

c,cov = curve_fit(yams,tp,pc,guess)
print(c)
```

```
[5.19081513e+02 2.98961745e+01 6.52202263e+02 2.82789303e-01]
```

[207]:
```
n = len(ny_fact['per_uninsured'])
y = np.empty(n)

for i in range(n):
    y[i] = ferg(ny_fact['per_uninsured'][i],c[0],c[1],c[2],c[3])

plt.plot(ny_fact['per_uninsured'], ny_fact['positive'])
plt.plot(ny_fact['per_uninsured'], y, 'r.')
```

[207]:
```
[<matplotlib.lines.Line2D at 0x1f71782c7c8>]
```



Tried soooo many combinations of variables including using factors and polyfit curves, but still get really bad test R2 scores. oh well, goes to show the issue is very complex. Will keep messing around with this over the summer

[209]:
```
model = smf.ols(formula='positive~ pop_density+per_disabled+ unemployment_rate ␣
↪+per_households_snap + med_income + pcnt_pov +␣
↪families_on_suplimental_income +families_on_social_security␣
↪+yams(per_uninsured,c[0],c[1],c[2],c[3]) + per_insured+per_minority␣
↪+per_young_adult+per_youth+two +threefour +ferg(fivesix,c[0],c[1],c[2],c[3])␣
↪+rocky(sevenmore,c[0],c[1],c[2],c[3])', data=train)
res = model.fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                positive   R-squared:                       0.772
Model:                            OLS   Adj. R-squared:                  0.744
Method:                 Least Squares   F-statistic:                     27.29
Date:                Tue, 19 May 2020   Prob (F-statistic):           1.00e-35
Time:                        10:30:20   Log-Likelihood:                -1067.9
No. Observations:                 155   AIC:                             2172.
Df Residuals:                     137   BIC:                             2227.
Df Model:                          17
Covariance Type:            nonrobust
================================================================================
================================
                                                 coef    std err          t
P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
--------------------------------
Intercept                                    41.3584    250.985      0.165
0.869    -454.947     537.664
pop_density                                1715.5662   2189.236      0.784
0.435   -2613.498    6044.631
per_disabled                                 17.5507      8.492      2.067
0.041       0.758      34.344
unemployment_rate                            -3.1406      5.207     -0.603
0.547     -13.438       7.157
per_households_snap                           3.1743      5.849      0.543
0.588      -8.392      14.741
med_income                                   -0.0004      0.001     -0.386
0.700      -0.003       0.002
pcnt_pov                                     -6.9797      8.230     -0.848
0.398     -23.253       9.294
families_on_suplimental_income               -0.0622      0.061     -1.027
0.306      -0.182       0.058
families_on_social_security                   0.0023      0.041      0.056
0.956      -0.079       0.084
yams(per_uninsured, c[0], c[1], c[2], c[3])   0.0563      0.166      0.339
0.735      -0.273       0.385
per_insured                                  -5.0196      4.768     -1.053
```

```
0.294     -14.448       4.408
per_minority                                    0.6681      1.226      0.545
0.587      -1.756       3.092
per_young_adult                                 9.6890      6.373      1.520
0.131      -2.913      22.291
per_youth                                       8.9248      8.219      1.086
0.279      -7.327      25.176
two                                             0.0138      0.031      0.453
0.651      -0.047       0.074
threefour                                       0.1064      0.035      3.013
0.003       0.037       0.176
ferg(fivesix, c[0], c[1], c[2], c[3])           0.0018      0.002      0.706
0.481      -0.003       0.007
rocky(sevenmore, c[0], c[1], c[2], c[3])        0.0060      0.004      1.629
0.106      -0.001       0.013
==============================================================================
Omnibus:                       11.789   Durbin-Watson:                   2.112
Prob(Omnibus):                  0.003   Jarque-Bera (JB):               30.384
Skew:                           0.093   Prob(JB):                     2.53e-07
Kurtosis:                       5.161   Cond. No.                     9.73e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 9.73e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[210]:
```
test_preds = res.predict(test)
print('testing r2 is:', r2_score(test_preds, y_test))
```

```
testing r2 is: -1.8124989380943068
```

[ ]:

[ ]: