simon.zhang.rapidtask

March 6, 2018

0.1 Simon Zhang | | Feb '18 Data Analyst Performance Task

simonczhang@gmail.com | | 832-857-3826

0.1.1 Intro

Come with me as I dive deep into the amazing world of the school and district datasets, and along the way, answer the 4 main questions posed for this particular task. This 'first-cut' analysis will be mainly based on my stream-of-conscious and my curiosities as I discover the many new and exciting facets of the effects of State X's school intervention after a one year long peer-coaching program!

```
In [1]: from IPython.core.interactiveshell import InteractiveShell
        InteractiveShell.ast_node_interactivity = "all"
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        from ggplot import *
        from pprint import pprint
        from patsy import dmatrices
        import statsmodels.api as sm
        import statsmodels.formula.api as smf
        from statsmodels.stats.outliers_influence import variance_inflation_factor
        from scipy.stats import binom_test
```

```
//anaconda/envs/py35/lib/python3.5/site-packages/ggplot/utils.py:81: FutureWarning: pandas.tsli
You can access Timestamp as pandas.Timestamp
  pd.tslib.Timestamp,
//anaconda/envs/py35/lib/python3.5/site-packages/ggplot/stats/smoothers.py:4: FutureWarning: Th
  from pandas.lib import Timestamp
//anaconda/envs/py35/lib/python3.5/site-packages/statsmodels/compat/pandas.py:56: FutureWarning
  from pandas.core import datetools
```

```
In [2]: #load data
        ddf = pd.read_csv('district_data.csv')
        sdf = pd.read_csv('school_data.csv')
```

```
In [3]: ddf.head()

Out[3]:    corp1  treatment
        0     29          1
        1    129          1
        2    239          0
        3    259          1
        4    369          1

In [4]: ddf.shape

Out[4]: (156, 2)

In [5]: ddf['treatment'].value_counts()

Out[5]: 1    88
        0    68
        Name: treatment, dtype: int64

In [6]: ddf['corp1'].value_counts()

Out[6]: 5374    1
        5459    1
        2654    1
        3164    1
        3419    1
        5284    1
        1624    1
        1879    1
        5629    1
        4219    1
        5619    1
        3409    1
        3029    1
        3149    1
        2124    1
        1604    1
        3139    1
        2114    1
        2399    1
        1129    1
        4349    1
        4684    1
        4734    1
        5344    1
        2819    1
        1659    1
        1914    1
        3449    1
```

```
5624    1
4419    1
       ..
4149    1
399     1
3009    1
1164    1
5259    1
3464    1
2439    1
3714    1
4229    1
3459    1
3484    1
3999    1
1184    1
4594    1
5489    1
4539    1
5524    1
5304    1
1974    1
2739    1
4529    1
944     1
6059    1
4744    1
2984    1
4519    1
3494    1
4004    1
674     1
5639    1
Name: corp1, Length: 156, dtype: int64
```

The district dataset has 156 rows and 2 columns. There is 88 districts that have been treated and 68 that have not, out of 156 districts. Also, I wanted to make sure all the 156 district keys were indeed uniqe in the district dataset.

```
In [7]: ddf.isnull().sum()
```

```
Out[7]: corp1        0
        treatment    0
        dtype: int64
```

There doesn't seem to be any values explicitly missing from the district dataset.

```
In [8]: sdf.head()
```

```
Out[8]:    district  schl1  enrollment  asian_pct  black_pct  hispanic_pct  white_pct  \
        0      5914   6320         241   0.006920   0.000000      0.000000   0.989619
        1       239    155         514   0.017143   0.179048      0.032381   0.735238
        2      4319   3514         258   0.000000   0.000000      0.019084   0.938931
        3      3449   2842         320   0.000000   0.015823      0.000000   0.958861
        4      4209   3428         472   0.030426   0.000000      0.010142   0.959432

            pct_frl  ed_lesshs  positive_env  mathscore_gain_std
        0  0.161512  18.500000             0           -1.045121
        1  0.349462  16.000000             1           -0.846501
        2  0.653199  20.400000             0           -0.146986
        3  0.226837  23.299999             0           -0.064126
        4  0.000000   7.700000             0            2.243462
```

```
In [9]: sdf.shape
```

```
Out[9]: (520, 11)
```

```
In [10]: sdf['schl1'].value_counts()
```

```
Out[10]: 3068    1
         2796    1
         5420    1
         302     1
         303     1
         1328    1
         6450    1
         5954    1
         2450    1
         1334    1
         6368    1
         2362    1
         1344    1
         321     1
         3394    1
         171     1
         1350    1
         1352    1
         2898    1
         5454    1
         1359    1
         2386    1
         4440    1
         4442    1
         4444    1
         210     1
         4446    1
         2858    1
         4448    1
```

```
1378    1
        ..
3738    1
5344    1
2262    1
642     1
590     1
3298    1
1702    1
3724    1
2704    1
1054    1
1026    1
1398    1
3734    1
760     1
2718    1
3766    1
1690    1
3746    1
2579    1
2726    1
2727    1
2728    1
2670    1
2732    1
4862    1
688     1
3762    1
129     1
3134    1
5362    1
Name: schl1, Length: 520, dtype: int64
```

The schools dataset has 520 rows and 11 columns. Here I also wanted to make sure there are indeed 520 unique schools in the dataset.

```
In [11]: sdf['district'].value_counts()
```

```
Out[11]: 239     31
         1014    12
         4714    12
         5744    12
         1974    12
         5364    10
         5334    10
         5279    10
         259      9
```

5079	9
5344	9
5354	8
5929	8
5374	8
369	8
2399	8
4419	7
4929	7
1129	7
3009	7
2869	7
3629	7
3949	7
4619	6
4664	6
3999	6
4209	6
3449	6
5314	5
4229	5
	..
2459	1
3819	1
8539	1
3484	1
5524	1
3494	1
2444	1
2439	1
3459	1
3059	1
519	1
4584	1
3439	1
4459	1
5624	1
5459	1
4329	1
3409	1
5999	1
3644	1
5384	1
5709	1
1624	1
3329	1
2654	1
3714	1

```
4774    1
3309    1
5529    1
2739    1
Name: district, Length: 152, dtype: int64
```

Interesting...In the district dataset there were 156 unique district keys. In the school dataset there are only 152 unique districts. The below districts are not present in the school dataset but exist in the district dataset that I'll deal with later when I combine the datasets.

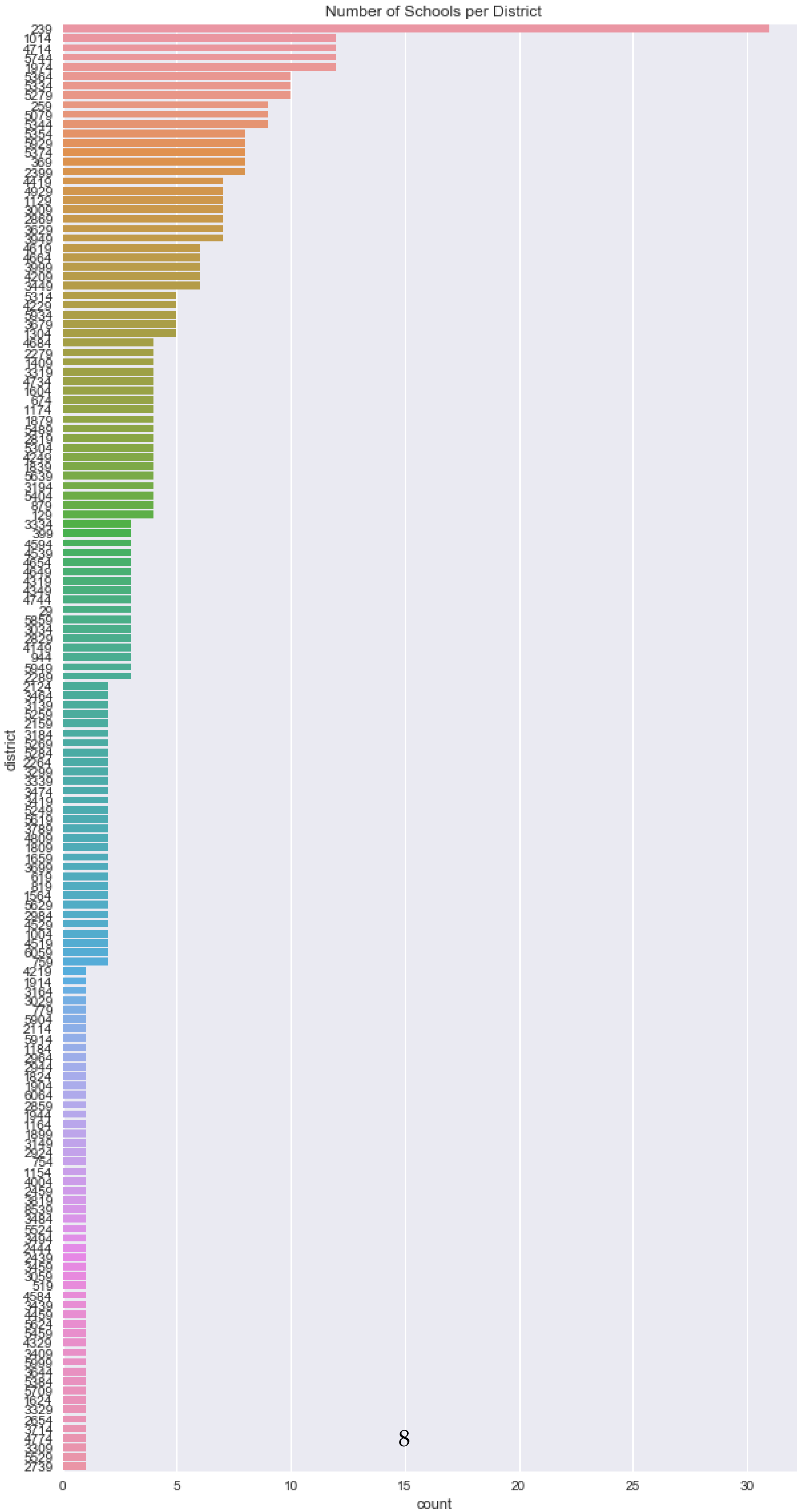
```
In [12]: d_s = sdf['district'].tolist()
         missing_districts = []
         for i, s in ddf.iterrows():
             district = s['corp1']
             if district not in d_s:
                 missing_districts.append(district)
         print(missing_districts)
```

```
[2044, 2769, 4019, 5839]
```

```
In [13]: district_count = sdf['district'].value_counts()
         print(len(district_count[district_count<2]))
```

```
53
```

```
In [14]: sns.set(rc={'figure.figsize':(10, 20)})
         p = sns.countplot(y = 'district', data = sdf, order = sdf['district'].value_counts().i
```


Number of Schools per District
district
count
0
5
10
15
20
25
30

There seems to be a disproportionate number of schools per district represented in the dataset. District 239 has 31 schools while there are 53 other districts with only 1 school represented. This seems rather odd. I cannot tell from the information provided if this is normal, but from my general knowledge from going through public school, I feel like there should be more than 1 school in a district and at least where I grew up, and also not 31 schools in a district. However, the information provided said that all schools in each treated district received treatment. Since the districts were the ones to implement this, maybe the districts did something wrong... In any case, I think this is very important to keep in mind as I continue on my journey.

In [15]:
```
sdf.isnull().sum()
```

Out[15]:
```
district              0
schl1                 0
enrollment            0
asian_pct            27
black_pct            27
hispanic_pct         27
white_pct            27
pct_frl              27
ed_lesshs             1
positive_env          0
mathscore_gain_std    0
dtype: int64
```

In [16]:
```
null_data = sdf[sdf.isnull().any(axis=1)]
null_data.shape
null_data
```

Out[16]: (28, 11)

Out[16]:

	district	schl1	enrollment	asian_pct	black_pct	hispanic_pct	\
39	2159	1642	358	NaN	NaN	NaN	
67	4539	3738	268	NaN	NaN	NaN	
68	5859	6286	358	NaN	NaN	NaN	
81	1304	1054	200	NaN	NaN	NaN	
85	5334	5289	702	NaN	NaN	NaN	
88	1974	1462	300	NaN	NaN	NaN	
103	3999	3294	140	NaN	NaN	NaN	
144	4664	3766	355	NaN	NaN	NaN	
145	4249	3474	347	NaN	NaN	NaN	
148	4619	3830	675	NaN	NaN	NaN	
183	1659	1258	522	NaN	NaN	NaN	
187	2944	2426	725	NaN	NaN	NaN	
195	239	186	286	NaN	NaN	NaN	
274	4744	4334	397	NaN	NaN	NaN	
285	754	622	610	NaN	NaN	NaN	
293	3949	3250	122	NaN	NaN	NaN	

329	5354	5353	552	NaN	NaN	NaN
371	1839	1334	314	NaN	NaN	NaN
386	3184	2626	456	NaN	NaN	NaN
394	3149	2579	496	0.0	0.002053	0.0
398	4714	4440	355	NaN	NaN	NaN
410	879	698	397	NaN	NaN	NaN
418	1974	1478	243	NaN	NaN	NaN
421	2984	2450	335	NaN	NaN	NaN
474	3999	3298	138	NaN	NaN	NaN
483	1304	1058	229	NaN	NaN	NaN
488	1129	916	232	NaN	NaN	NaN
512	4419	3654	308	NaN	NaN	NaN

	white_pct	pct_frl	ed_lesshs	positive_env	mathscore_gain_std
39	NaN	NaN	31.900000	0	1.112068
67	NaN	NaN	24.900000	1	0.407426
68	NaN	NaN	18.799999	0	1.354968
81	NaN	NaN	29.400000	0	-1.035049
85	NaN	NaN	9.400000	0	0.867408
88	NaN	NaN	23.799999	1	-1.633518
103	NaN	NaN	18.299999	0	0.411487
144	NaN	NaN	12.400000	0	1.212134
145	NaN	NaN	12.800000	0	0.298029
148	NaN	NaN	8.600000	0	1.738986
183	NaN	NaN	21.200001	1	-0.032557
187	NaN	NaN	21.200001	0	-0.816038
195	NaN	NaN	16.000000	0	-0.057523
274	NaN	NaN	7.000000	0	0.870401
285	NaN	NaN	14.900000	1	1.002799
293	NaN	NaN	21.500000	0	0.457790
329	NaN	NaN	7.900000	0	1.552696
371	NaN	NaN	15.200000	0	-0.555412
386	NaN	NaN	18.900000	1	-0.294817
394	0.98152	0.15587	NaN	1	-0.032405
398	NaN	NaN	24.400000	1	-2.896326
410	NaN	NaN	21.900000	0	0.145097
418	NaN	NaN	23.799999	0	-2.096035
421	NaN	NaN	19.700001	0	-0.369864
474	NaN	NaN	18.299999	0	-0.880281
483	NaN	NaN	29.400000	0	-0.040749
488	NaN	NaN	17.799999	0	-0.012751
512	NaN	NaN	17.500000	0	-1.601093

In the school dataset there are 28 rows with at least 1 missing value. I printed all of them out above. I'll deal with this once I merge the two datasets.

In [17]:
```
#merge two datasets to one
df = pd.merge(ddf, sdf, left_on='corp1', right_on='district', how = 'outer')
df.head(16)
```

Out[17]:

	corp1	treatment	district	schl1	enrollment	asian_pct	black_pct
0	29	1	29.0	30.0	329.0	0.003021	0.009063
1	29	1	29.0	34.0	335.0	0.003135	0.003135
2	29	1	29.0	2.0	233.0	0.000000	0.004237
3	129	1	129.0	38.0	499.0	0.038710	0.019355
4	129	1	129.0	64.0	507.0	0.054230	0.013015
5	129	1	129.0	61.0	383.0	0.020089	0.042411
6	129	1	129.0	39.0	531.0	0.032653	0.073469
7	239	0	239.0	155.0	514.0	0.017143	0.179048
8	239	0	239.0	263.0	621.0	0.045662	0.242009
9	239	0	239.0	146.0	452.0	0.047930	0.250545
10	239	0	239.0	198.0	400.0	0.021220	0.320955
11	239	0	239.0	214.0	511.0	0.012195	0.101626
12	239	0	239.0	210.0	489.0	0.088745	0.270563
13	239	0	239.0	182.0	389.0	0.010101	0.219697
14	239	0	239.0	246.0	392.0	0.007895	0.339474
15	239	0	239.0	186.0	286.0	NaN	NaN

\

	hispanic_pct	white_pct	pct_frl	ed_lesshs	positive_env
0	0.096677	0.851964	0.379603	14.5	0.0
1	0.065831	0.815047	0.318885	14.5	0.0
2	0.042373	0.927966	0.300000	14.5	0.0
3	0.019355	0.901075	0.018382	3.8	1.0
4	0.019523	0.902386	0.003846	3.8	1.0
5	0.022321	0.892857	0.031008	3.8	0.0
6	0.030612	0.840816	0.028926	3.8	0.0
7	0.032381	0.735238	0.349462	16.0	1.0
8	0.070015	0.563166	0.392366	16.0	0.0
9	0.032680	0.603486	0.423767	16.0	1.0
10	0.143236	0.496021	0.582310	16.0	1.0
11	0.038618	0.788618	0.450677	16.0	1.0
12	0.145022	0.450216	0.556660	16.0	0.0
13	0.204545	0.489899	0.559078	16.0	0.0
14	0.231579	0.328947	0.784810	16.0	0.0
15	NaN	NaN	NaN	16.0	0.0

\

	mathscore_gain_std
0	-0.457215
1	-0.537477
2	0.300080
3	0.955901
4	1.343666
5	1.630452
6	1.739707
7	-0.846501
8	-0.608421
9	1.103983
10	-0.786966

```
11          -1.221598
12          -2.542357
13          -0.622134
14          -2.927336
15          -0.057523
```

In [18]:
```
df.shape
```

Out[18]:
```
(524, 13)
```

After doing an outer join to merge both datasets there seems to be the the right expected number of data rows and columns. There were 4 districts that were in the districts dataset that were added on top of the 520 rows in the schools dataset.

In [19]:
```
null_data = df[df.isnull().any(axis=1)]
null_data.shape
null_data
```

Out[19]:
```
(32, 13)
```

Out[19]:

	corp1	treatment	district	schl1	enrollment	asian_pct	black_pct	\
15	239	0	239.0	186.0	286.0	NaN	NaN	
65	754	1	754.0	622.0	610.0	NaN	NaN	
73	879	0	879.0	698.0	397.0	NaN	NaN	
98	1129	1	1129.0	916.0	232.0	NaN	NaN	
106	1304	1	1304.0	1054.0	200.0	NaN	NaN	
110	1304	1	1304.0	1058.0	229.0	NaN	NaN	
123	1659	1	1659.0	1258.0	522.0	NaN	NaN	
129	1839	0	1839.0	1334.0	314.0	NaN	NaN	
139	1974	0	1974.0	1462.0	300.0	NaN	NaN	
149	1974	0	1974.0	1478.0	243.0	NaN	NaN	
151	2044	1	NaN	NaN	NaN	NaN	NaN	
155	2159	1	2159.0	1642.0	358.0	NaN	NaN	
179	2769	1	NaN	NaN	NaN	NaN	NaN	
196	2944	0	2944.0	2426.0	725.0	NaN	NaN	
199	2984	1	2984.0	2450.0	335.0	NaN	NaN	
214	3149	0	3149.0	2579.0	496.0	0.0	0.002053	
217	3184	0	3184.0	2626.0	456.0	NaN	NaN	
275	3949	1	3949.0	3250.0	122.0	NaN	NaN	
278	3999	1	3999.0	3294.0	140.0	NaN	NaN	
282	3999	1	3999.0	3298.0	138.0	NaN	NaN	
285	4019	0	NaN	NaN	NaN	NaN	NaN	
302	4249	1	4249.0	3474.0	347.0	NaN	NaN	
318	4419	1	4419.0	3654.0	308.0	NaN	NaN	
324	4539	1	4539.0	3738.0	268.0	NaN	NaN	
331	4619	1	4619.0	3830.0	675.0	NaN	NaN	
345	4664	1	4664.0	3766.0	355.0	NaN	NaN	
362	4714	0	4714.0	4440.0	355.0	NaN	NaN	
370	4744	1	4744.0	4334.0	397.0	NaN	NaN	

418	5334	1	5334.0	5289.0	702.0	NaN	NaN
441	5354	1	5354.0	5353.0	552.0	NaN	NaN
497	5839	1	NaN	NaN	NaN	NaN	NaN
498	5859	0	5859.0	6286.0	358.0	NaN	NaN

	hispanic_pct	white_pct	pct_frl	ed_lesshs	positive_env	\
15	NaN	NaN	NaN	16.000000	0.0	
65	NaN	NaN	NaN	14.900000	1.0	
73	NaN	NaN	NaN	21.900000	0.0	
98	NaN	NaN	NaN	17.799999	0.0	
106	NaN	NaN	NaN	29.400000	0.0	
110	NaN	NaN	NaN	29.400000	0.0	
123	NaN	NaN	NaN	21.200001	1.0	
129	NaN	NaN	NaN	15.200000	0.0	
139	NaN	NaN	NaN	23.799999	1.0	
149	NaN	NaN	NaN	23.799999	0.0	
151	NaN	NaN	NaN	NaN	NaN	
155	NaN	NaN	NaN	31.900000	0.0	
179	NaN	NaN	NaN	NaN	NaN	
196	NaN	NaN	NaN	21.200001	0.0	
199	NaN	NaN	NaN	19.700001	0.0	
214	0.0	0.98152	0.15587	NaN	1.0	
217	NaN	NaN	NaN	18.900000	1.0	
275	NaN	NaN	NaN	21.500000	0.0	
278	NaN	NaN	NaN	18.299999	0.0	
282	NaN	NaN	NaN	18.299999	0.0	
285	NaN	NaN	NaN	NaN	NaN	
302	NaN	NaN	NaN	12.800000	0.0	
318	NaN	NaN	NaN	17.500000	0.0	
324	NaN	NaN	NaN	24.900000	1.0	
331	NaN	NaN	NaN	8.600000	0.0	
345	NaN	NaN	NaN	12.400000	0.0	
362	NaN	NaN	NaN	24.400000	1.0	
370	NaN	NaN	NaN	7.000000	0.0	
418	NaN	NaN	NaN	9.400000	0.0	
441	NaN	NaN	NaN	7.900000	0.0	
497	NaN	NaN	NaN	NaN	NaN	
498	NaN	NaN	NaN	18.799999	0.0	

	mathscore_gain_std
15	-0.057523
65	1.002799
73	0.145097
98	-0.012751
106	-1.035049
110	-0.040749
123	-0.032557
129	-0.555412

```
139           -1.633518
149           -2.096035
151                 NaN
155            1.112068
179                 NaN
196           -0.816038
199           -0.369864
214           -0.032405
217           -0.294817
275            0.457790
278            0.411487
282           -0.880281
285                 NaN
302            0.298029
318           -1.601093
324            0.407426
331            1.738986
345            1.212134
362           -2.896326
370            0.870401
418            0.867408
441            1.552696
497                 NaN
498            1.354968
```

In [20]:
```
32/524
```

Out[20]:
```
0.061068702290076333
```

Summary of Missing Values in the Joined Dataframe Now you can see that there are 32 rows with missing values because I used an outer join and included all values in both datasets. I wanted to make sure my joined dataset looked right. I previewed the 32 rows that had missing values in the new dataframe and everything seems correct. Rows with missing values account for about 6.11% of the total data. 4 of the rows only have data from the district dataset. There are 27 rows with only missing Race data (black, white, Asian, Hispanic) and percent free lunch. There is 1 row that only has missing local-area education level data.

Before I try to find any anomolies in the data, I'm going to have to deal with the missing values first. I'm going to use two different methods:

In [21]:
```
df = pd.merge(ddf, sdf, left_on='corp1', right_on='district')
df.drop(['district'], axis=1, inplace=True)
df.shape
```

Out[21]:
```
(520, 12)
```

I decided to get rid of the 4 data points that didn't have any school data because there was really no use for them in analyzing math performance when there is no data for it at all and because there was just 4 of them so I wasn't losing too much data. I also dropped the `district` column as it is the same as the `corp1` column.

For the rest of the missing values, I am going to replace them with a randomly generated number, who's mean is set as the mean of the given column values and who's standard deviation is set to the std of the given column values. This way I can generate values that are close approximations of the data that is reasonably likely to appear in each point.

```
In [22]: #create dictionary for mu,sigma of all columns that contain missing values
         column_stat_table = {}

         list_col = df.columns[df.isna().any()].tolist()
         print(list_col)
         for i in list_col:
             mean = np.nanmean(df[i])
             std = np.nanstd(df[i])
             name = i
             column_stat_table[name] = (mean, std)

         pprint(column_stat_table)

         #replace all missing values with a random number in the column's distribution
         np.random.seed(1)
         df.fillna(99999, inplace = True)
         for column in df:
             if column in list_col:
                 for tup in df[column].iteritems():
                     val = tup[1]
                     if val == 99999:
                         mu = column_stat_table[column][0]
                         std = column_stat_table[column][1]
                         random_val = abs(np.random.normal(mu, std))
                         df[column].replace(val, random_val, inplace=True)
```

```
['asian_pct', 'black_pct', 'hispanic_pct', 'white_pct', 'pct_frl', 'ed_lesshs']
{'asian_pct': (0.010756779699797l6, 0.020236213079422721),
 'black_pct': (0.081781314880324557, 0.14718297638478267),
 'ed_lesshs': (17.528130864973026, 6.4023793195403895),
 'hispanic_pct': (0.055240283257606493, 0.089825756735112949),
 'pct_frl': (0.33870936964097359, 0.18899462583584203),
 'white_pct': (0.81393108290872207, 0.21220774675443163)}
```

```
In [23]: null_data = df[df.isnull().any(axis=1)]
         print(null_data.shape)
         print(null_data)
         print(df.iloc[15])
```

```
(0, 12)
Empty DataFrame
Columns: [corp1, treatment, schl1, enrollment, asian_pct, black_pct, hispanic_pct, white_pct, p
Index: []
```

```
corp1                    239.000000
treatment                  0.000000
schl1                    186.000000
enrollment               286.000000
asian_pct                  0.043627
black_pct                  0.055948
hispanic_pct               0.036476
white_pct                  0.771329
pct_frl                    0.288396
ed_lesshs                 16.000000
positive_env               0.000000
mathscore_gain_std        -0.057523
Name: 15, dtype: float64
```

After replacing all the missing values, I wanted to preview one of the rows that used to have missing values to make sure they are filled correctly. I knew row 15 had missing values so I previewed it above to check my work. Now that I've finished dealing with all the missing values, I will begin doing some exploration!

1 Exploring the Data

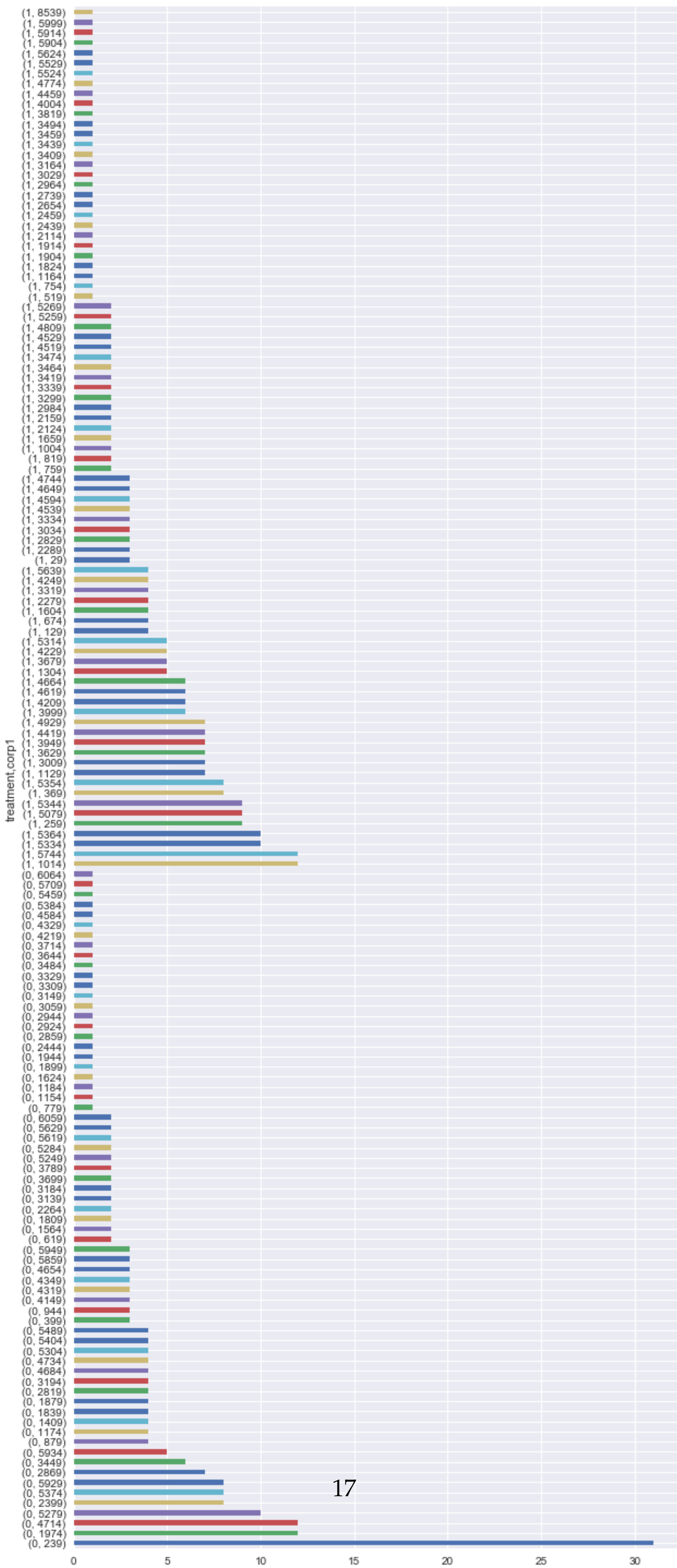
1.0.1 Single Variable Analysis

I think that separating the data into two groups, treatment 1 and treatment 0 groups, will be the most helpful in evaluating the effectiveness of the program. First, I'll do some single variable analyses looking at distribution of variables. Then, I'll move into some multivariable analyses of the relationship between variables.

I'll begin by retouching on the weird phenomenon I found earlier regarding schools per district in the dataset.

Schools

```
In [24]: #Distribution of schools per district, per treatment group
         a = df.groupby(['treatment'])['corp1'].value_counts()
         a.plot(kind='barh', figsize= (10,26))

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1592a668>
```



treatment,corp1
(1, 8539)
(1, 5999)
(1, 5914)
(1, 5904)
(1, 5624)
(1, 5529)
(1, 5524)
(1, 4774)
(1, 4459)
(1, 4004)
(1, 3819)
(1, 3494)
(1, 3459)
(1, 3439)
(1, 3409)
(1, 3164)
(1, 3029)
(1, 2964)
(1, 2739)
(1, 2654)
(1, 2459)
(1, 2439)
(1, 2114)
(1, 1914)
(1, 1904)
(1, 1824)
(1, 1164)
(1, 754)
(1, 519)
(1, 5269)
(1, 5259)
(1, 4809)
(1, 4529)
(1, 4519)
(1, 3474)
(1, 3464)
(1, 3419)
(1, 3339)
(1, 3299)
(1, 2984)
(1, 2159)
(1, 2124)
(1, 1659)
(1, 1004)
(1, 819)
(1, 759)
(1, 4744)
(1, 4649)
(1, 4594)
(1, 4539)
(1, 3334)
(1, 3034)
(1, 2829)
(1, 2289)
(1, 29)
(1, 5639)
(1, 4249)
(1, 3319)
(1, 2279)
(1, 1604)
(1, 674)
(1, 129)
(1, 5314)
(1, 4229)
(1, 3679)
(1, 1304)
(1, 4664)
(1, 4619)
(1, 4209)
(1, 3999)
(1, 4929)
(1, 4419)
(1, 3949)
(1, 3629)
(1, 3009)
(1, 1129)
(1, 5354)
(1, 369)
(1, 5344)
(1, 5079)
(1, 259)
(1, 5364)
(1, 5334)
(1, 5744)
(1, 1014)
(0, 6064)
(0, 5709)
(0, 5459)
(0, 5384)
(0, 4584)
(0, 4329)
(0, 4219)
(0, 3714)
(0, 3644)
(0, 3484)
(0, 3329)
(0, 3309)
(0, 3149)
(0, 3059)
(0, 2944)
(0, 2924)
(0, 2859)
(0, 2444)
(0, 1944)
(0, 1899)
(0, 1624)
(0, 1184)
(0, 1154)
(0, 779)
(0, 6059)
(0, 5629)
(0, 5619)
(0, 5284)
(0, 5249)
(0, 3789)
(0, 3699)
(0, 3184)
(0, 3139)
(0, 2264)
(0, 1809)
(0, 1564)
(0, 619)
(0, 5949)
(0, 5859)
(0, 4654)
(0, 4349)
(0, 4319)
(0, 4149)
(0, 944)
(0, 399)
(0, 5489)
(0, 5404)
(0, 5304)
(0, 4734)
(0, 4684)
(0, 3194)
(0, 2819)
(0, 1879)
(0, 1839)
(0, 1409)
(0, 1174)
(0, 879)
(0, 5934)
(0, 3449)
(0, 2869)
(0, 5929)
(0, 5374)
(0, 2399)
(0, 5279)
(0, 4714)
(0, 1974)
(0, 239)
0
5
10
15
20
25
30

Here I separated the two treatment groups and looked at the number of schools in each district. The distribution of schools per district still seems odd but at least both the treatment groups show a similar trend here. This helps support the notion that the treatment assignments by the researchers was random. Before I say anything conclusion of the random assignment, I want to look at the distribution of other variables between the two groups as well.

```
In [25]: #find number of schools per treatment group
         a = df.groupby(['treatment'])['schl1'].count()
         a
         #a.plot(kind='bar')
         b = ddf['treatment'].value_counts()
         b
```

```
Out[25]: treatment
         0    229
         1    291
         Name: schl1, dtype: int64
```

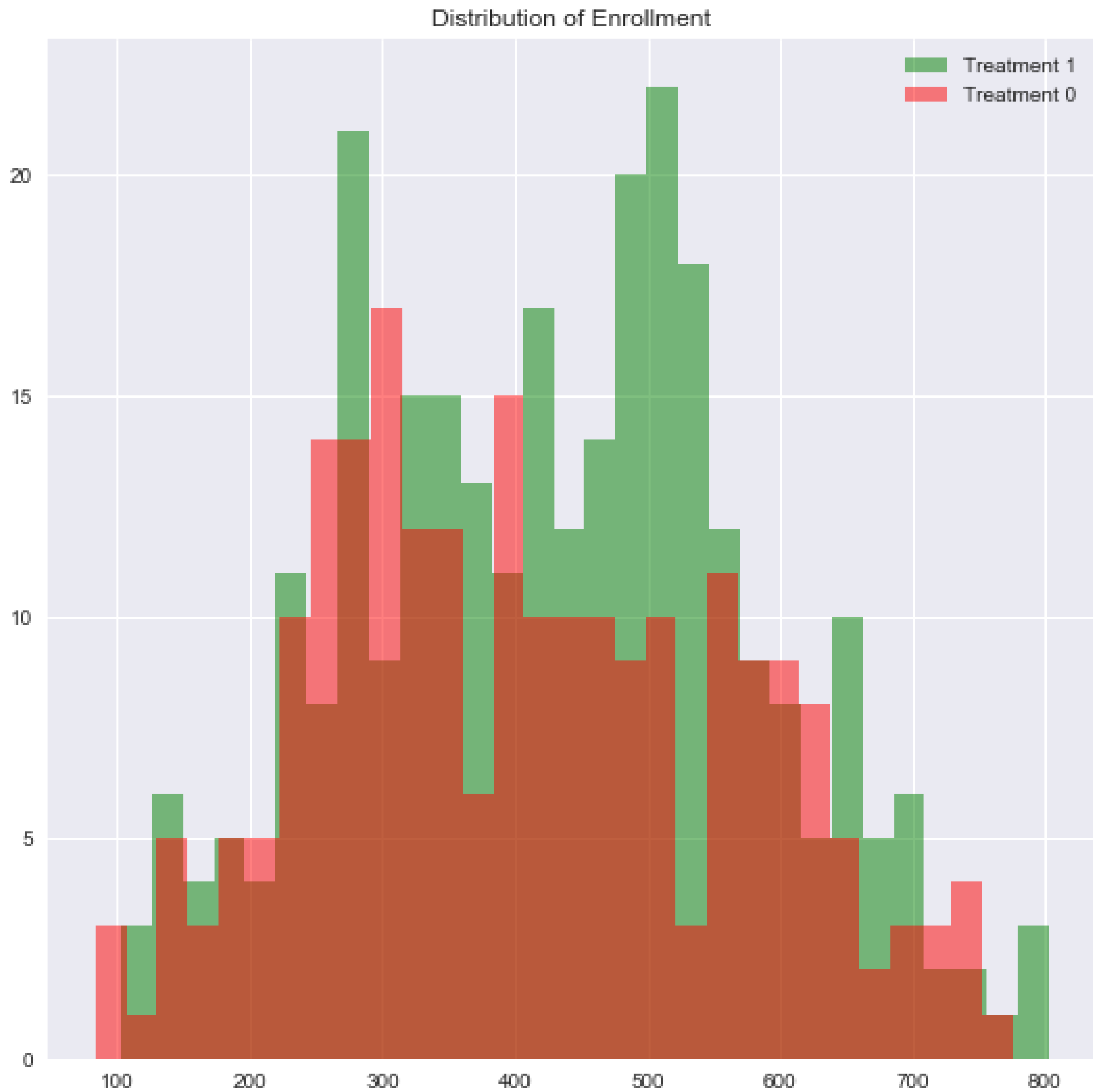
```
Out[25]: 1    88
         0    68
         Name: treatment, dtype: int64
```

Now that I've seen the distribution of schools per district in the two groups, I wanted to look at total number of schools in each treatment group. There are 291 schools in the treatment 1 group and 229 schools in treatment 0 group. This seems to be normal given there's 20 more treatment 1 districts than treatment 0 districts.

Next, I'll make some histograms and look at the distribution of all the other variables separated by treatment group and also some basic statistics of the variables.

Enrollment

```
In [26]: a = df[df['treatment'] == 1]['enrollment']
         b = df[df['treatment'] == 0]['enrollment']

         bins = np.linspace(-1, 1, 10)
         q = plt.figure(figsize=(9,9))
         q = plt.hist(a, 30, alpha=0.5, label='Treatment 1', color='green', density=False)
         q = plt.hist(b, 30, alpha=0.5, label='Treatment 0', color='red', density=False)
         q = plt.legend(loc='upper right')
         q = plt.title('Distribution of {}'.format('Enrollment'))
         plt.show()

         df.groupby(['treatment'])['enrollment'].describe()
```



Out[26]:

treatment	count	mean	std	min	25%	50%	75%	max
0	229.0	408.615721	155.350173	84.0	288.0	397.0	524.0	775.0
1	291.0	431.395189	151.222781	103.0	317.0	439.0	530.0	802.0

Looks like there is slightly more kids enrolled in treatment 1 group schools on average. The variance is pretty big though so I don't think there's a significant difference.

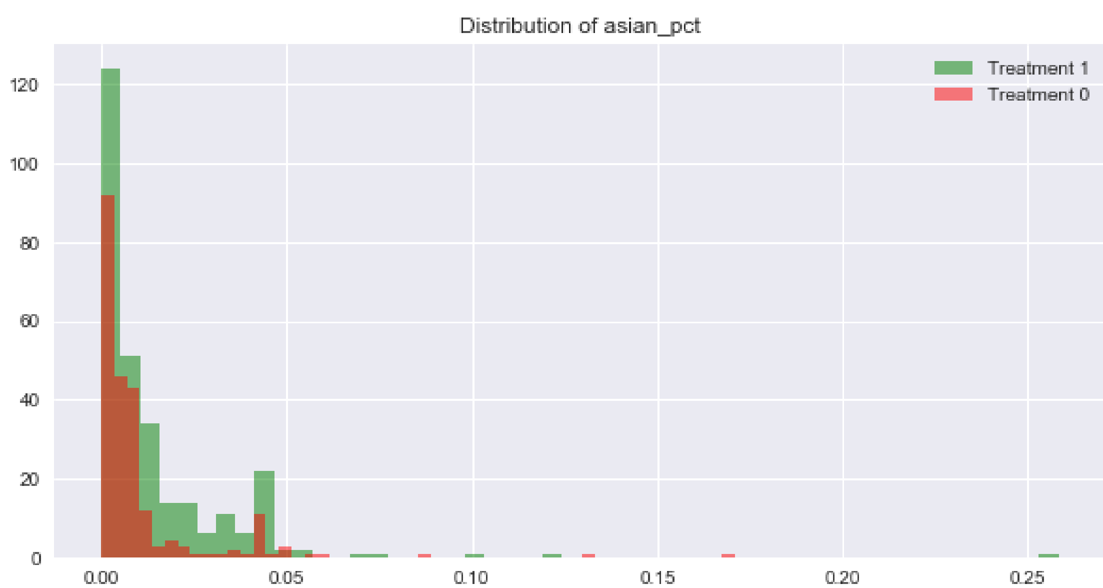
```
In [27]: df[df['corp1']==239]['enrollment'].describe()
```

```
Out[27]: count     31.000000
         mean     459.870968
         std      129.393390
```

```
min      257.000000
25%      354.000000
50%      464.000000
75%      572.000000
max      711.000000
Name: enrollment, dtype: float64
```

I wanted to see if the school in treatment 0 that had 31 schools could be pulling the average enrollment for treatment 0 group down but it doesn't seem so. I'll go ahead and look at the distribution of the rest of the variables by the two groups. I'll write a function that will automatically output all the graphs and statistics for the rest of the variables for me. YAY!

```
In [28]: plot_list = ['asian_pct', 'black_pct', 'hispanic_pct', 'white_pct', 'pct_frl', 'ed_les
         for i in plot_list:
             a = df[df['treatment'] == 1][i]
             b = df[df['treatment'] == 0][i]

             bins = np.linspace(-1, 1, 10)
             q = plt.figure(figsize=(10,5))
             q = plt.hist(a, 50, alpha=0.5, label='Treatment 1', color='green')
             q = plt.hist(b, 50, alpha=0.5, label='Treatment 0', color='red')
             q = plt.legend(loc='upper right')
             q = plt.title('Distribution of {}'.format(i))
             plt.show()

             print('Statistics for {} below:'.format(i))
             df.groupby(['treatment'])[i].describe()
```

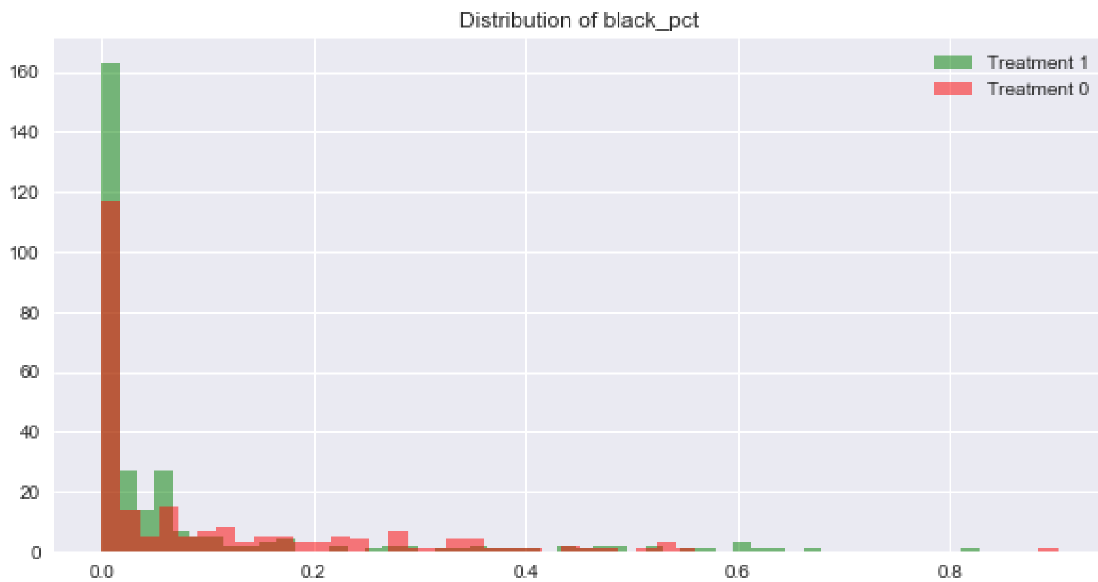


Statistics for asian_pct below:

Out[28]:

	count	mean	std	min	25%	50%	75%	\
treatment								
0	229.0	0.010264	0.019066	0.0	0.000000	0.004950	0.009288	
1	291.0	0.014195	0.022336	0.0	0.000653	0.007042	0.018387	

	max
treatment	
0	0.170732
1	0.258352

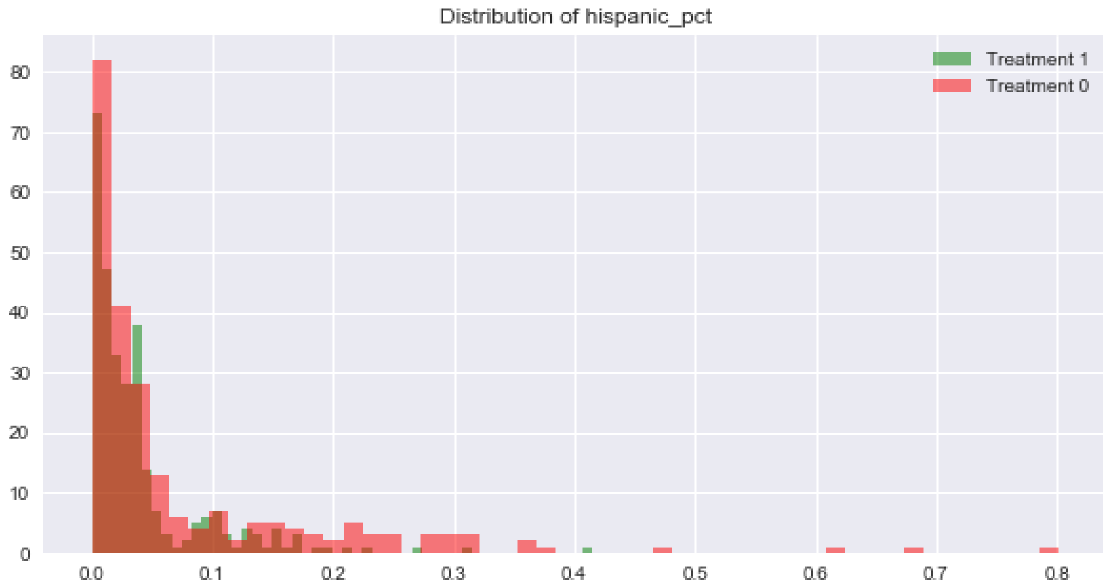


Statistics for black_pct below:

Out[28]:

	count	mean	std	min	25%	50%	75%	\
treatment								
0	229.0	0.091761	0.141057	0.0	0.001377	0.017143	0.129496	
1	291.0	0.071531	0.145127	0.0	0.000800	0.011129	0.055948	

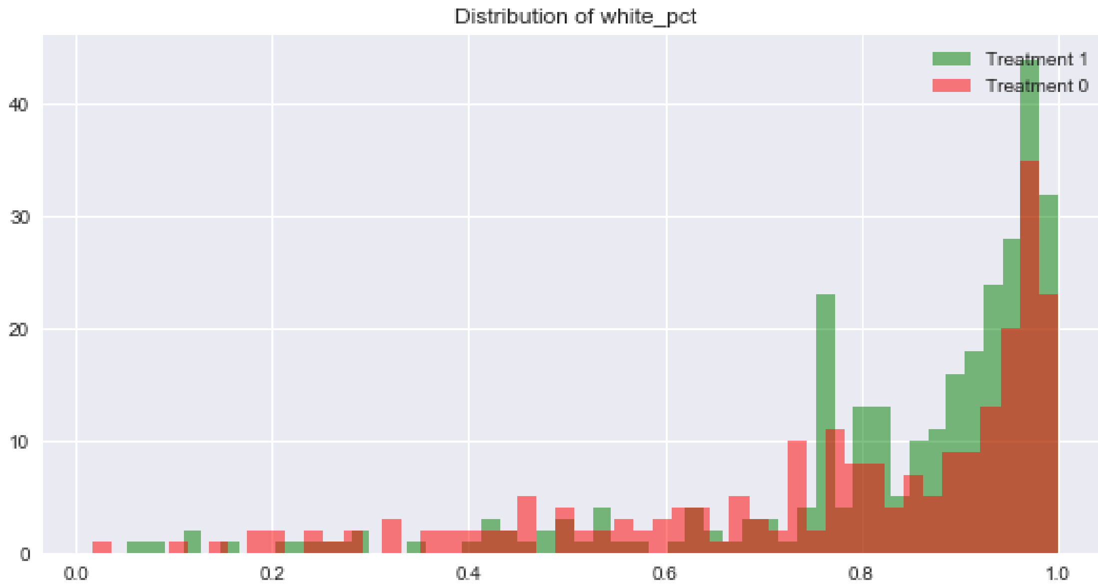
	max
treatment	
0	0.901674
1	0.827068



Statistics for hispanic_pct below:

Out[28]:

```
           count      mean       std  min       25%       50%       75%  \
treatment
0          229.0  0.073642  0.115330  0.0  0.008547  0.027864  0.084942
1          291.0  0.039018  0.052543  0.0  0.008227  0.022321  0.040603

                max
treatment
0          0.801075
1          0.414634
```



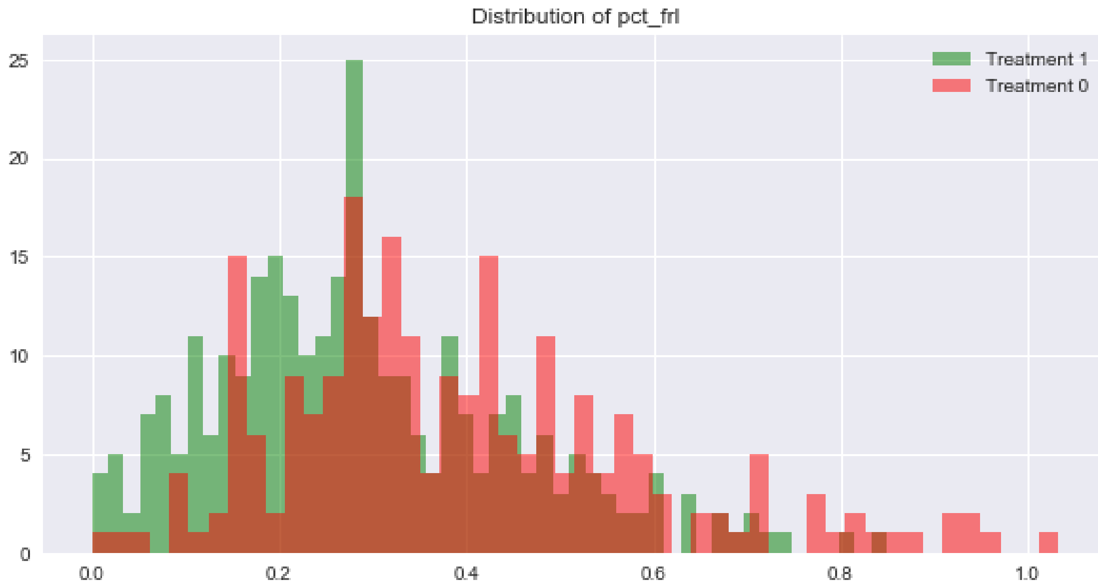
Statistics for white_pct below:

Out[28]:

treatment	count	mean	std	min	25%	50%	75%
0	229.0	0.781051	0.225351	0.016736	0.678392	0.861742	0.962687
1	291.0	0.835853	0.188312	0.052632	0.773742	0.905336	0.963189

\

treatment	max
0	1.0
1	1.0

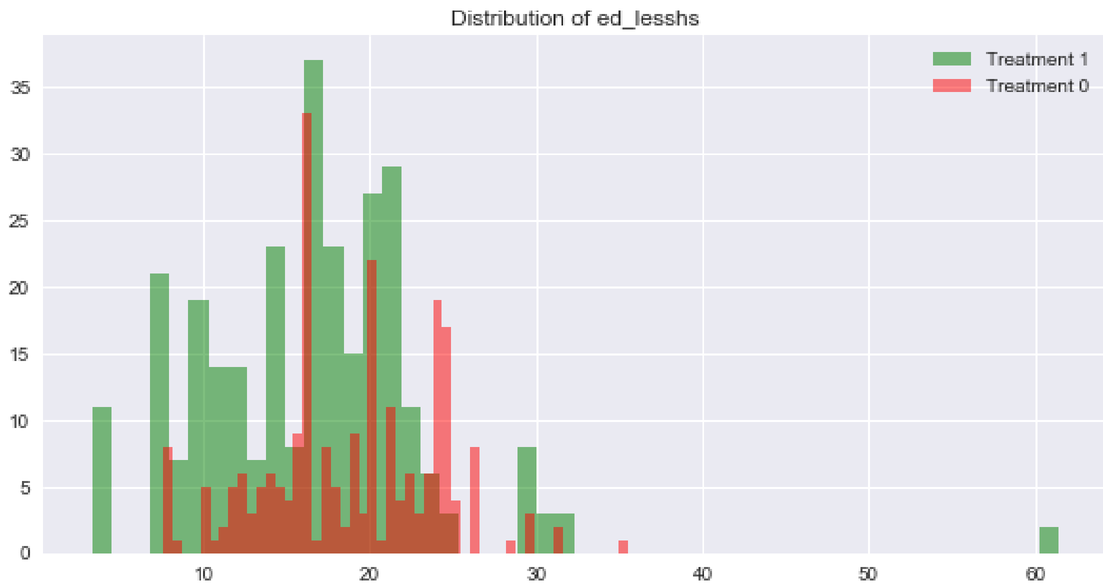


Statistics for pct_frl below:

Out[28]:

treatment	count	mean	std	min	25%	50%	75%
0	229.0	0.393854	0.193718	0.0	0.267218	0.349462	0.494662
1	291.0	0.290645	0.163507	0.0	0.177597	0.274924	0.387736

treatment	max
0	1.032051
1	0.849315

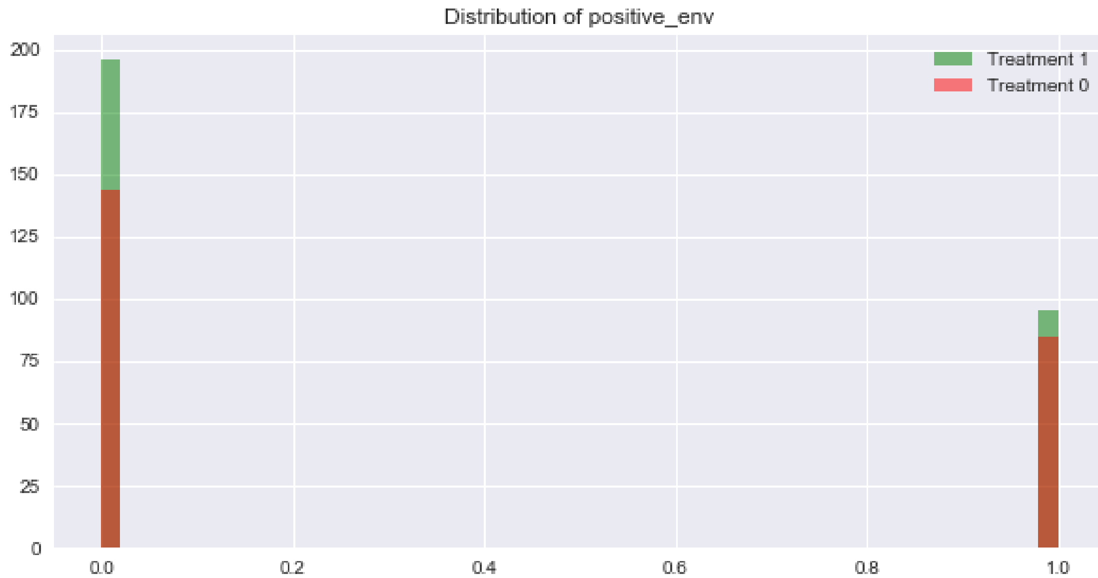


Statistics for ed_lesshs below:

Out[28]:

treatment	count	mean	std	min	25%	50%	75%	\
0	229.0	18.880943	5.179399	7.6	15.9	19.299999	23.299999	
1	291.0	16.449828	7.050597	3.3	11.8	16.400000	20.500000	

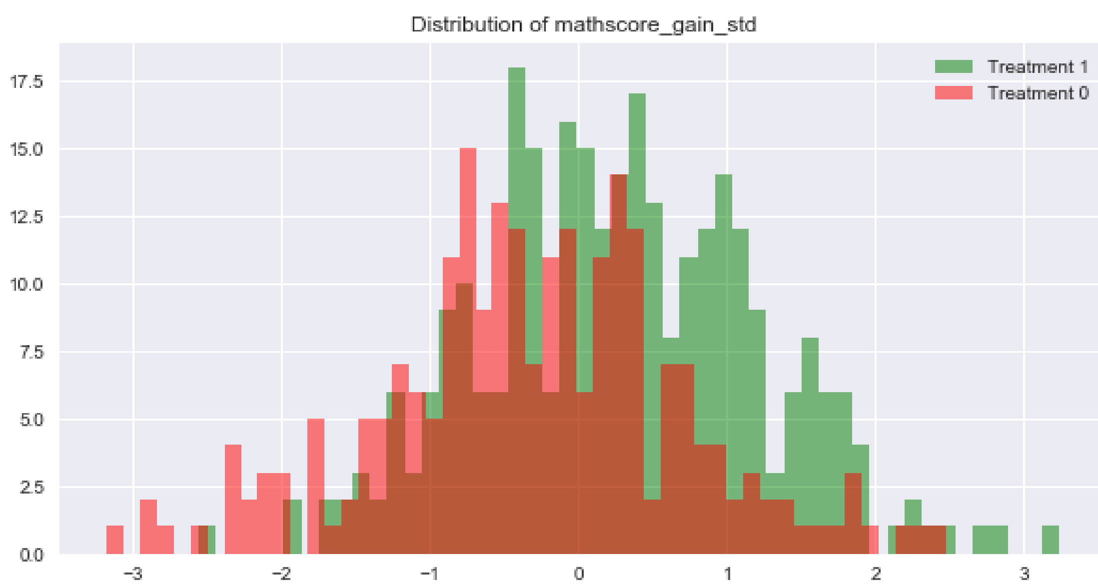
treatment	max
0	35.500000
1	61.400002




```
Statistics for positive_env below:
```

Out[28]:

treatment	count	mean	std	min	25%	50%	75%	max
0	229.0	0.371179	0.484179	0.0	0.0	0.0	1.0	1.0
1	291.0	0.326460	0.469726	0.0	0.0	0.0	1.0	1.0



```
Statistics for mathscore_gain_std below:
```

```
Out[28]:            count      mean       std       min       25%       50%       75%  \
treatment
0           229.0 -0.314813  1.013710 -3.177708 -0.878541 -0.329303  0.308151
1           291.0  0.275373  0.943731 -2.564522 -0.371354  0.237052  0.939742

                max
treatment
0           2.470063
1           3.238389
```

Since I wrote a function, I'm not able to comment on each plot in between outputs. I'll briefly go through each plot below:

Asian Percent Graphically, it looks like the Asian percentage is mainly under 1%. The distribution is heavily skewed to the right and the median for both treatment groups is less than 1%. It's much better to use median instead of mean when looking for the center when the data is heavily skewed like so.

```
In [29]: #conditional explore asian pct
         a = df[df['asian_pct'] > .1]
         b = pd.pivot_table(index = ['treatment', 'corp1'], data = a)
         b
```

```
Out[29]:                  asian_pct  black_pct  ed_lesshs  enrollment  hispanic_pct  \
treatment corp1
0         5404    0.151059   0.255415       12.0       275.5      0.128909
1         4744    0.122830   0.042724        7.0       715.0      0.048064
          5744    0.179269   0.038046       11.4       496.5      0.041752

                  mathscore_gain_std   pct_frl  positive_env   schl1  white_pct
treatment corp1
0         5404              1.318291  0.474622           0.0  5892.0   0.390496
1         4744              1.644482  0.038700           1.0  4336.0   0.765020
          5744              1.882334  0.221811           0.0  6199.5   0.671903
```

In districts where the Asian percentage is greater than 10% for both treatment groups, the math score gain is positive. There is a slightly higher gain in treatment 1 group.

Black Percent The black percentage looks similarly skewed to the right but far less sparse than the Asian distribution. It ranges from 0% all the way up to 90%. The difference between the two group's median percentage is less than 1% and all else seems rather equal as well. The standard deviations are pretty large for both groups.

```
In [30]: a = df[df['black_pct'] > .5]
         b = pd.pivot_table(index=['treatment', 'schl1', 'black_pct'], data = a)
         b
```

Out[30]:

treatment	schl1	black_pct	asian_pct	corp1	ed_lesshs	enrollment
0	4454	0.539683	0.000000	4714	24.400000	640
	4458	0.901674	0.004184	4714	24.400000	470
	5399	0.523810	0.004329	5374	7.600000	458
	5411	0.545817	0.013944	5374	7.600000	501
	5414	0.528139	0.034632	5374	7.600000	473
	5429	0.505725	0.009542	5374	7.600000	506
1	303	0.663934	0.045082	259	16.900000	400
	310	0.827068	0.002506	259	16.900000	425
	4804	0.601286	0.000000	4929	20.799999	369
	4822	0.525469	0.000000	4929	20.799999	311
	5345	0.610294	0.017647	5354	7.900000	651
	5347	0.627451	0.015251	5354	7.900000	448
	5350	0.629243	0.022193	5354	7.900000	741
	5352	0.521385	0.024440	5354	7.900000	511
	5364	0.599057	0.009434	5364	16.400000	407
	5373	0.567073	0.029268	5354	7.900000	797
	5382	0.547216	0.026634	5364	16.400000	318

treatment	schl1	black_pct	hispanic_pct	mathscore_gain_std	pct_frl
0	4454	0.539683	0.285714	-1.950318	1.032051
	4458	0.901674	0.046025	-2.230451	0.914405
	5399	0.523810	0.021645	0.683436	0.329193
	5411	0.545817	0.105578	2.349952	0.495202
	5414	0.528139	0.168831	0.946275	0.492308
	5429	0.505725	0.192748	0.597386	0.328386
1	303	0.663934	0.108607	-0.536247	0.564706
	310	0.827068	0.037594	-1.227202	0.747036
	4804	0.601286	0.032154	-1.947255	0.668539
	4822	0.525469	0.016086	-2.564522	0.663551
	5345	0.610294	0.213235	0.902085	0.546314
	5347	0.627451	0.163399	0.414198	0.336484
	5350	0.629243	0.121410	-0.097336	0.241245
	5352	0.521385	0.136456	0.130693	0.317708
	5364	0.599057	0.018868	-0.844053	0.512195
	5373	0.567073	0.051220	0.619762	0.171795
	5382	0.547216	0.169492	1.411931	0.462222

treatment	schl1	black_pct	positive_env	white_pct
0	4454	0.539683	0	0.107937
	4458	0.901674	1	0.016736
	5399	0.523810	1	0.361472
	5411	0.545817	1	0.260956
	5414	0.528139	1	0.177489
	5429	0.505725	1	0.234733

```
1         303   0.663934            0   0.120902
          310   0.827068            0   0.052632
          4804  0.601286            1   0.241158
          4822  0.525469            0   0.404826
          5345  0.610294            0   0.077941
          5347  0.627451            0   0.124183
          5350  0.629243            0   0.155352
          5352  0.521385            1   0.242363
          5364  0.599057            0   0.285377
          5373  0.567073            0   0.280488
          5382  0.547216            1   0.210654
```

In schools that have 50% or more black students, it seems here that treatment doesn't have a direct relationship with math scores, but if you look at the `ed_lesshs` column, you can see there's a really good inverse relationship between it and math scores. This relationship isn't quite as apparently when looking at the Asian percent possibly due to the fact that there's much less variation in the `ed_lesshs` as well as much less data. I'll keep this in mind and continue.

Hispanic Percent The hispanic percent distribution is similar to the black one but the median for both treatment groups is about 1% higher. It does look like there are more schools in the treatment 0 group with hispanic percentage greater than .1 though.

```
In [31]: a = df[df['hispanic_pct'] > .4]
         b = pd.pivot_table(index=['treatment', 'schl1'], data = a)
         b
```

```
Out[31]:                  asian_pct  black_pct  corp1  ed_lesshs  enrollment  \
         treatment schl1
         0         994     0.000000   0.000000   1174       24.9         409
                   4444    0.000000   0.014793   4714       24.4         337
                   4448    0.002203   0.090308   4714       24.4         414
                   4456    0.003155   0.064669   4714       24.4         661
         1         3790    0.006098   0.048780   4594       29.1         339

                          hispanic_pct  mathscore_gain_std   pct_frl  positive_env  \
         treatment schl1
         0         994        0.801075           -1.283995  0.913151             1
                   4444       0.473373           -0.499026  0.496855             1
                   4448       0.687225           -1.021157  0.966292             0
                   4456       0.616719           -1.501899  0.718153             0
         1         3790       0.414634            0.552072  0.696165             0

                          white_pct
         treatment schl1
         0         994     0.193548
                   4444    0.500000
                   4448    0.193833
                   4456    0.272871
         1         3790    0.500000
```

Here the `ed_lesshs` column is more stable like in the Asian percent table but much higher values. With such high values, it makes sense that the gain in math score is much lower as per my previous hypothesis. I think this table gives a good case for the positive influence of the treatment on math score. It does show a slight positive increase in the treatment 1 group as opposed to a negative gain in ALL the treatment 0 schools in schools with hispanic dominated schools. There is only 1 school in treatment 1 group though so I don't have much data to work off of.

White Percent The white distribution is the first race distribution where the graph is heavily skewed left. The median for treatment group 0 and group 1 are .86 and .91, respectively. This is the completely opposite distribution as the other three races that I looked at. There's some schools with just 1% white students and some with 100%.

```
In [32]: a = df[df['white_pct'] > .9]
         b = pd.pivot_table(index=['treatment'], data = a)
         b

         a = df[df['white_pct'] > .9]
         b = pd.pivot_table(index=['treatment', 'corp1'], data = a)
         b
```

```
Out[32]:            asian_pct  black_pct        corp1  ed_lesshs  enrollment  \
         treatment
         0           0.002964   0.004053  3540.050000  18.264360  395.420000
         1           0.005911   0.005328  3294.433333  17.848667  389.186667

                    hispanic_pct  mathscore_gain_std   pct_frl  positive_env    schl1  \
         treatment
         0              0.012150           -0.145132  0.283402      0.330000  3252.57
         1              0.012558            0.250329  0.255561      0.346667  2882.86

                    white_pct
         treatment
         0           0.961312
         1           0.957380
```

```
Out[32]:                  asian_pct  black_pct  ed_lesshs  enrollment  hispanic_pct  \
         treatment corp1
         0         399     0.001548   0.006192  12.300000  244.500000      0.024360
                   619     0.001149   0.002857  14.500000  486.000000      0.008046
                   779     0.000000   0.000000  12.200000  537.000000      0.007394
                   944     0.001238   0.003713  17.799999  515.666667      0.014495
                   1154    0.000000   0.003339  12.700000  585.000000      0.005008
                   1184    0.006993   0.004662  14.000000  415.000000      0.000000
                   1409    0.004202   0.002801  22.400000  267.000000      0.015406
                   1564    0.005228   0.001743  14.000000  554.000000      0.002568
                   1624    0.009288   0.021672  21.100000  676.000000      0.003096
                   1809    0.002747   0.000000  18.400000  424.000000      0.011905
                   1839    0.004993   0.000583  15.200000  540.000000      0.017527
```

	1879	0.002467	0.005863	10.300000	318.250000	0.002002
	1899	0.000000	0.003676	17.000000	261.000000	0.011029
	1944	0.011080	0.005540	12.400000	402.000000	0.000000
	2264	0.007605	0.022814	22.600000	271.000000	0.011407
	2399	0.001319	0.009539	26.200001	259.714286	0.001309
	2444	0.000000	0.000000	13.900000	423.000000	0.009592
	2819	0.000831	0.003967	13.700000	223.750000	0.005967
	2859	0.003263	0.000000	21.500000	630.000000	0.009788
	2924	0.003431	0.000000	17.500000	610.000000	0.005146
	3059	0.006250	0.000000	14.000000	336.000000	0.016667
	3139	0.007800	0.014041	11.500000	618.000000	0.024961
	3149	0.000000	0.002053	13.536051	496.000000	0.000000
	3184	0.000000	0.000000	18.900000	364.000000	0.005540
	3194	0.003544	0.003428	21.400000	312.500000	0.014606
	3309	0.008646	0.017291	8.200000	683.000000	0.005764
	3329	0.002755	0.001377	10.600000	731.000000	0.012397
	3449	0.002195	0.006593	23.299999	323.250000	0.002660
	3484	0.008584	0.005722	10.300000	691.000000	0.008584
	3644	0.000000	0.000000	28.799999	142.000000	0.019481
...		...	...	...	...	...
1	3629	0.002878	0.000269	15.000000	386.857143	0.006886
	3679	0.007541	0.006407	19.299999	294.666667	0.024699
	3819	0.000000	0.000000	18.500000	527.000000	0.040541
	3949	0.006806	0.003340	21.500000	319.166667	0.022393
	3999	0.000000	0.001626	18.299999	201.000000	0.004167
	4004	0.003125	0.007812	20.900000	649.000000	0.004687
	4209	0.028594	0.005905	7.700000	532.833333	0.014621
	4229	0.003552	0.007057	19.700001	402.000000	0.013243
	4249	0.012550	0.009946	12.800000	426.500000	0.019374
	4419	0.017488	0.011129	17.500000	532.000000	0.044515
	4459	0.002183	0.006550	20.000000	474.000000	0.039301
	4519	0.001106	0.002212	20.000000	336.000000	0.026765
	4529	0.001458	0.000000	61.400002	336.000000	0.015881
	4539	0.007752	0.003876	24.900000	241.000000	0.027132
	4649	0.000000	0.002137	11.800000	472.333333	0.024352
	4664	0.008967	0.007348	12.400000	485.333333	0.039530
	4774	0.003937	0.007874	14.400000	243.000000	0.003937
	4809	0.007317	0.004878	16.500000	414.000000	0.009756
	5079	0.002189	0.002274	22.200001	253.777778	0.004312
	5259	0.001406	0.000846	14.500000	652.000000	0.002956
	5269	0.000000	0.000000	21.400000	440.000000	0.026846
	5314	0.013630	0.044842	9.800000	519.500000	0.011848
	5524	0.000000	0.000000	31.500000	428.000000	0.000000
	5529	0.000000	0.000000	21.500000	329.000000	0.002941
	5624	0.003226	0.003226	15.100000	585.000000	0.001613
	5639	0.006655	0.024009	21.600000	369.000000	0.001704
	5744	0.006349	0.013760	11.400000	393.000000	0.013150
	5904	0.000000	0.000000	20.000000	342.000000	0.000000

```
          5914    0.006920   0.000000  18.500000  241.000000      0.000000
          8539    0.000000   0.000000  12.200000  174.000000      0.006024

                 mathscore_gain_std   pct_frl  positive_env        schl1  \
treatment corp1
0         399             -0.013240  0.355320      0.000000   426.000000
          619              0.500160  0.188083      0.000000   522.000000
          779             -0.437973  0.241316      1.000000   642.000000
          944              0.106775  0.208573      0.666667   783.333333
          1154             0.312879  0.306914      0.000000   954.000000
          1184             0.622525  0.170792      0.000000  1026.000000
          1409            -0.289881  0.251728      0.000000  1110.000000
          1564             0.072249  0.118709      0.500000  1184.000000
          1624            -0.506010  0.320937      1.000000  1210.000000
          1809             0.974572  0.219663      0.500000  1321.000000
          1839            -0.234049  0.248755      1.000000  1348.666667
          1879             0.208705  0.267732      0.750000  1408.000000
          1899            -0.108519  0.301115      0.000000  1374.000000
          1944            -1.181111  0.254011      0.000000  1398.000000
          2264            -0.535092  0.225962      0.000000  1698.000000
          2399            -1.236177  0.408151      0.428571  1888.285714
          2444            -1.168462  0.319481      0.000000  2002.000000
          2819            -0.170270  0.263646      0.000000  2295.000000
          2859            -0.037224  0.299373      0.000000  2339.000000
          2924            -1.074776  0.432000      0.000000  2410.000000
          3059            -0.719352  0.267218      0.000000  2458.000000
          3139             0.552604  0.154762      0.000000  2551.000000
          3149            -0.032405  0.155870      1.000000  2579.000000
          3184             0.591808  0.344388      1.000000  2614.000000
          3194            -0.416747  0.328916      0.250000  2646.500000
          3309             0.830550  0.056543      0.000000  2704.000000
          3329             0.164062  0.113737      0.000000  2714.000000
          3449             0.486371  0.292702      0.250000  2845.500000
          3484             0.678973  0.084892      0.000000  2902.000000
          3644             0.247893  0.458599      1.000000  3088.000000
...                             ...       ...           ...          ...
1         3629            -0.018884  0.299195      0.571429  3043.142857
          3679             0.357374  0.245997      0.000000  3127.333333
          3819            -0.416112  0.377649      0.000000  3198.000000
          3949             1.090026  0.392292      0.333333  3262.000000
          3999             0.003706  0.363920      0.666667  3314.666667
          4004            -0.291998  0.403429      1.000000  3334.000000
          4209             1.024638  0.079869      0.166667  3432.000000
          4229            -0.133124  0.234402      0.500000  3437.500000
          4249             1.305121  0.200899      0.000000  3471.000000
          4419             0.223383  0.123314      0.000000  3601.000000
          4459            -0.344054  0.379374      0.000000  3635.000000
          4519            -1.166867  0.248933      0.500000  3678.500000
```

4529	0.188597	0.153418	0.500000	3699.000000
4539	0.920374	0.271375	0.000000	3734.000000
4649	0.204220	0.163299	0.666667	3808.333333
4664	0.220028	0.121761	0.333333	3883.000000
4774	1.161966	0.199153	0.000000	4362.000000
4809	-0.400506	0.208633	1.000000	4672.000000
5079	0.809345	0.330650	0.555556	4880.222222
5259	-0.235837	0.103236	0.000000	4991.000000
5269	-1.165664	0.298544	1.000000	5062.000000
5314	1.458297	0.124148	0.500000	5193.500000
5524	-1.403217	0.551724	0.000000	5982.000000
5529	0.720727	0.295597	0.000000	5990.000000
5624	-0.008663	0.232787	1.000000	6044.000000
5639	0.019098	0.443587	0.000000	6092.666667
5744	0.376481	0.250169	0.400000	6156.200000
5904	-0.280807	0.147139	0.000000	6316.000000
5914	-1.045121	0.161512	0.000000	6320.000000
8539	0.878349	0.238889	1.000000	3163.000000

		white_pct
treatment	corp1	
0	399	0.957268
	619	0.981839
	779	0.983364
	944	0.967877
	1154	0.974958
	1184	0.981352
	1409	0.953003
	1564	0.977255
	1624	0.931889
	1809	0.958015
	1839	0.958536
	1879	0.975282
	1899	0.970588
	1944	0.969529
	2264	0.908745
	2399	0.977547
	2444	0.976019
	2819	0.977696
	2859	0.970636
	2924	0.969125
	3059	0.972917
	3139	0.931357
	3149	0.981520
	3184	0.972299
	3194	0.957101
	3309	0.938040
	3329	0.954545

```
               3449    0.957774
               3484    0.945637
               3644    0.980519
...                         ...
1              3629    0.956045
               3679    0.936704
               3819    0.926641
               3949    0.953704
               3999    0.968893
               4004    0.970312
               4209    0.945988
               4229    0.947104
               4249    0.921008
               4419    0.918919
               4459    0.947598
               4519    0.950234
               4529    0.971878
               4539    0.934109
               4649    0.957992
               4664    0.916957
               4774    0.972441
               4809    0.963415
               5079    0.981644
               5259    0.983529
               5269    0.959732
               5314    0.909373
               5524    0.972973
               5529    0.976471
               5624    0.988710
               5639    0.929081
               5744    0.937601
               5904    0.997245
               5914    0.989619
               8539    0.975904

[124 rows x 10 columns]
```

Theres 250 schools that have a white percentage that is greater than 90%. That's almost half of all the schools in the dataset. This accounts for 124 districts out of the 156 total districts. The first table above looks like there's another good case for the treatment working. With all other variables being close to equal, there is about a .395 positive change in math gain in schools where the white percentage is greater than 90%.

Free Lunch Eligibility Percentage Wow! The weird thing is that there exists a school in which the percentage of students that are eligible for lunch subsidies is greater than 100% or 1.0! Do the faculty and staff get to eat for free too?!

```
In [33]: a = df[df['pct_frl'] > 1]
         a
         b = sdf[sdf['pct_frl'] > 1]
         b

Out[33]:      corp1  treatment  schl1  enrollment  asian_pct  black_pct  hispanic_pct  \
         354   4714          0   4454         640        0.0   0.539683      0.285714

              white_pct   pct_frl  ed_lesshs  positive_env  mathscore_gain_std
         354   0.107937  1.032051       24.4             0           -1.950318

Out[33]:      district  schl1  enrollment  asian_pct  black_pct  hispanic_pct  \
         152      4714   4454         640        0.0   0.539683      0.285714

              white_pct   pct_frl  ed_lesshs  positive_env  mathscore_gain_std
         152   0.107937  1.032051       24.4             0           -1.950318

In [34]: df.at[354, 'pct_frl'] = 1.0
         df.iloc[354]

Out[34]: corp1                 4714.000000
         treatment                0.000000
         schl1                 4454.000000
         enrollment             640.000000
         asian_pct                0.000000
         black_pct                0.539683
         hispanic_pct             0.285714
         white_pct                0.107937
         pct_frl                  1.000000
         ed_lesshs               24.400000
         positive_env             0.000000
         mathscore_gain_std      -1.950318
         Name: 354, dtype: float64
```

I thought this might have been a mistake when I added random variables to fill in the missing values but I checked with the original school dataset and it was there all along. I went ahead and change this number to 1.0 so it makes sense. I'm glad I detected an anomoly in the data and was able to fix it.

Looking at the stats, the difference in means between the two groups is is more than we've seen in the past at 10%, but the variance is still too large to determine any kind of significance; however, graphically, you can see a clear shift in the distribution.

Education Less than HS

Of the many, there are two outliers with values of 61.4% in the `ed_lesshs` variable! Both schools from the same district (4529) that are almost completely white students, both in treatment 1, almost identical variables other than the positive environment school had a math loss of .43 while the non-positive environment school had a math gain of .81. How odd. The treatment in one school produced a postive environment but didn't produce good math score gains while another similar school produced the exact opposite results.

```
In [35]: a = df[df['ed_lesshs'] > 60]
         a
```

```
Out[35]:      corp1  treatment  schl1  enrollment  asian_pct  black_pct  hispanic_pct  \
         319   4529          1   3707         337   0.002915        0.0      0.002915
         320   4529          1   3691         335   0.000000        0.0      0.028846

              white_pct   pct_frl  ed_lesshs  positive_env  mathscore_gain_std
         319   0.985423  0.141618  61.400002             1           -0.436910
         320   0.958333  0.165217  61.400002             0            0.814104
```

Positive Environment Only about 1/3 of the schools have positive environments in both treatment groups. Oddly, treatment 0 group actually has a 5% higher average positive environment but the variability in both groups is really high. The treatment doesn't seem to promote a positive environment it seems.

Std Math Score Gain The mean math gain seems appears that the program does have a little affect on. Graphically, the green distribution is clearly shifted toward the positive side of the graph, but the variance is still too high to see a significant difference in the two means.

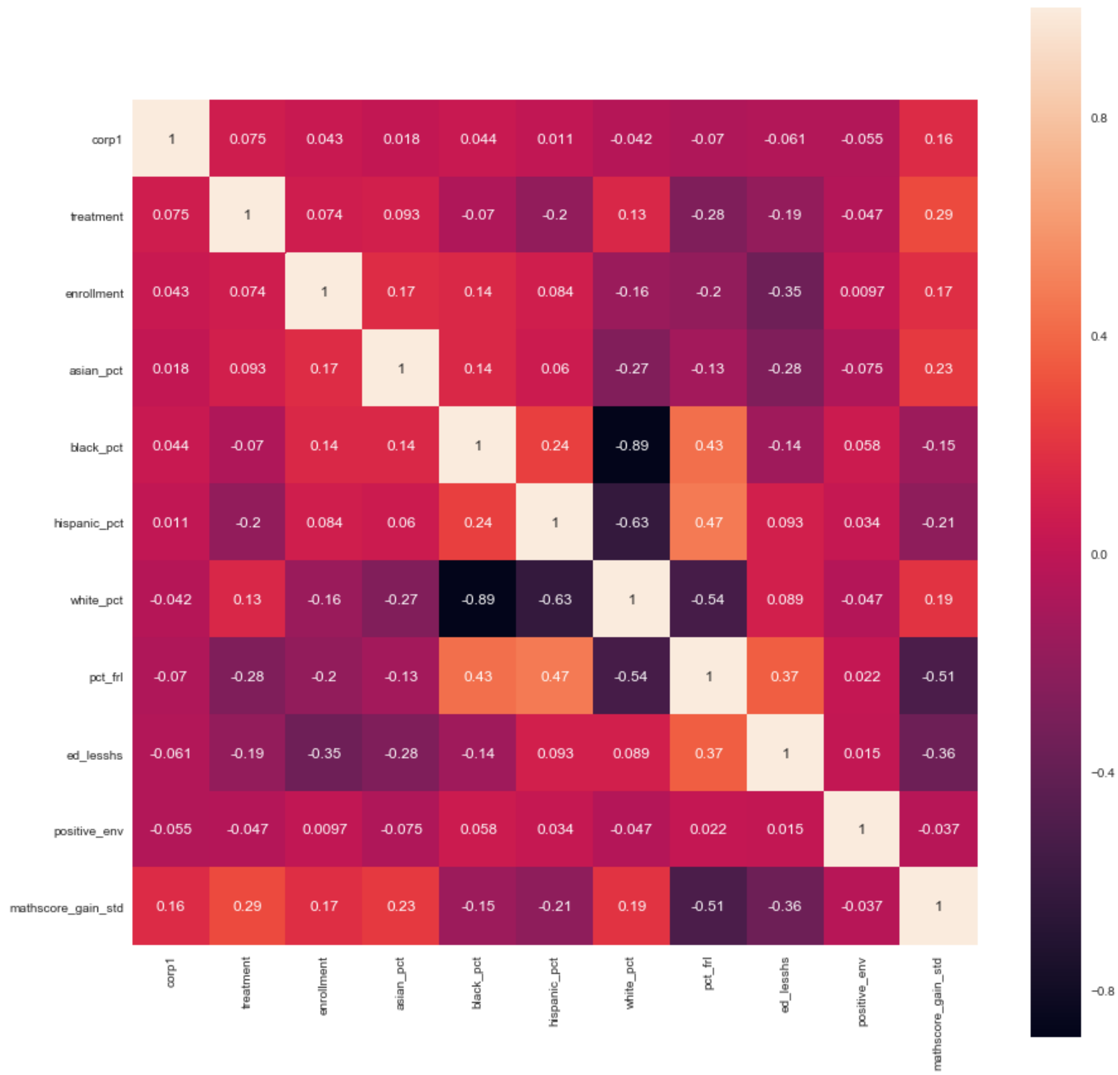
Now that I've taken a look in 2D space, I want to look at the data in higher dimensions as well.

1.0.2 Multivariable Analysis

I'll start by making a heatmap correlation matrix to see the relationships between all the variables in the dataset.

```
In [36]: cor_list = list(df.columns)
         cor_list.remove('schl1') # remove school id from matrices
         cor_df = df[cor_list]

         #correlation matrix
         corr = cor_df.corr()
         sns.set(rc={'figure.figsize':(15,15)})
         sns.heatmap(corr,
                     xticklabels=corr.columns.values,
                     yticklabels=corr.columns.values, square=True, annot=True)
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x1c16b97cc0>
```



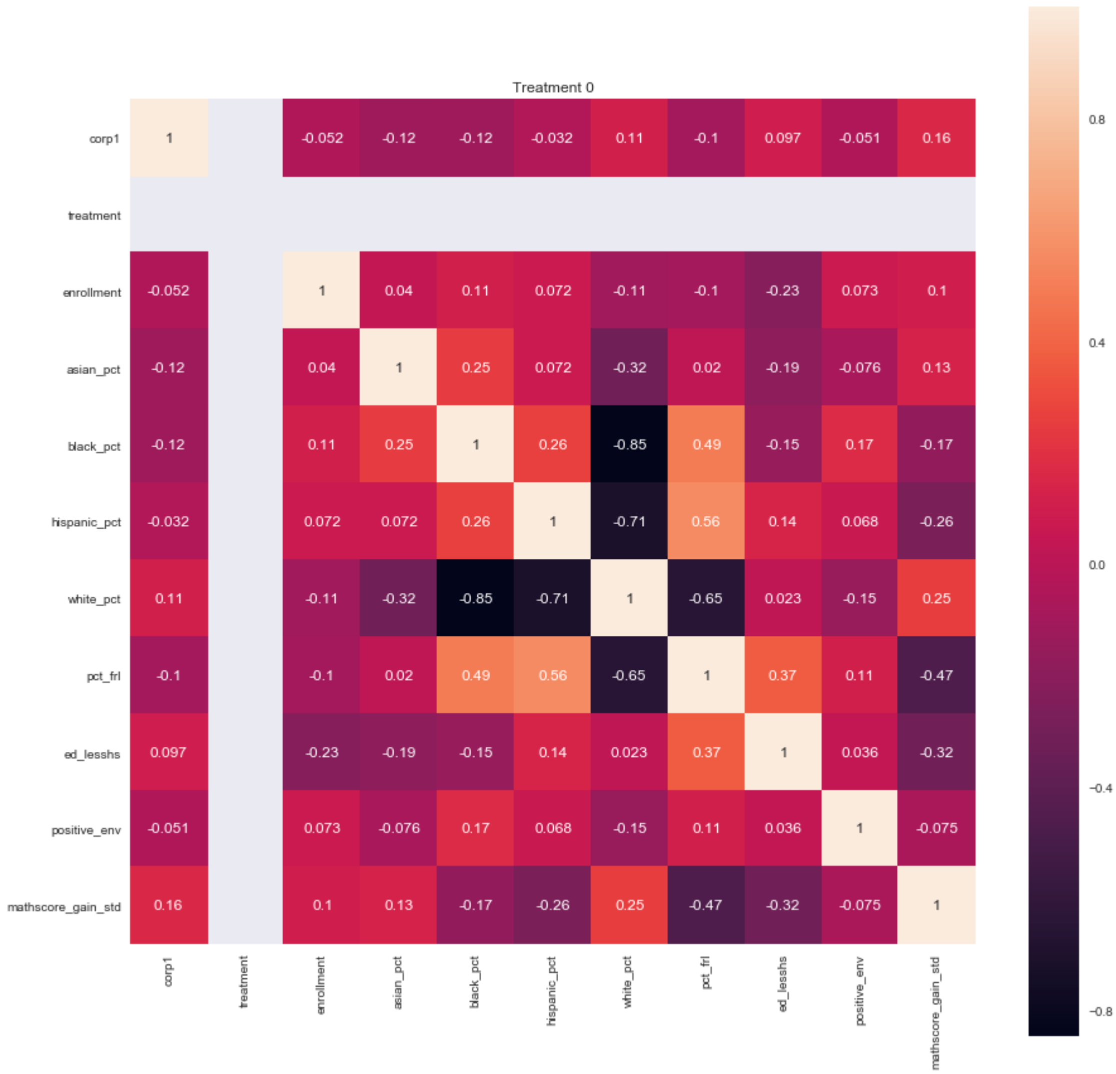
Here we can see that treatment does have a moderate positive relationship with math score gain, however, it has almost no relationship with a positive work environment. In fact, a positive work environment doesn't seem to correlated strongly with any variables in this dataset. This could have to do with the way SEA X translated the survey data.

There are also some relationships between student's race and their math scores. The strongest correlations are free lunch eligibility at -.51 and `ed_lesshs` at -.36. This means that as eligibility for lunch subsidies increases, math score decreases, and as local area population with less than a high school diploma increases, math scores decrease as well.

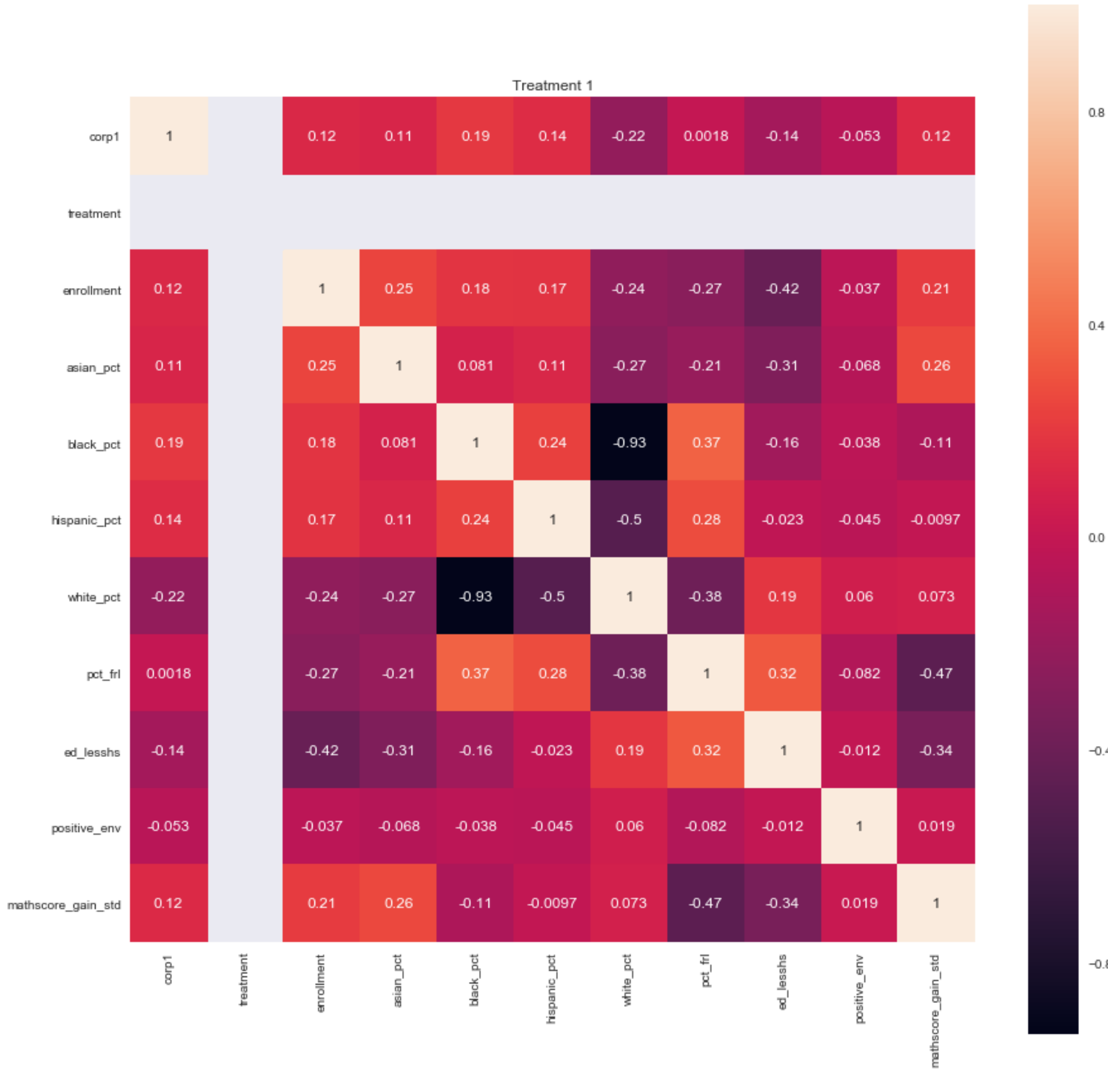
These two variables have a correlation with each other of .37. This makes sense. If we assume that more education has a correlation with more income, then we could say that areas with high percentage of population with less than a high school diploma also probably has lower income, and kids coming from those families are more likely to be eligible for free lunch. Eligibility for free lunch has a moderate positive relationship with black and hispanic percentage.

I'll also take a look at the change in correlations between variables between the two treatment groups.

```
In [37]: #correlation matrix for treatment 0 group
         corr = cor_df[cor_df['treatment']==0].corr()
         a = sns.set(rc={'figure.figsize':(15,15)})
         a = plt.title('Treatment 0')
         a = sns.heatmap(corr,
                     xticklabels=corr.columns.values,
                     yticklabels=corr.columns.values, square=True, annot=True)
```




```
In [38]: #correlation matrix for treatment 1 group
         corr = cor_df[cor_df['treatment']==1].corr()
         a = sns.set(rc={'figure.figsize':(15,15)})
         a = plt.title('Treatment 1')
         a = sns.heatmap(corr,
                     xticklabels=corr.columns.values,
                     yticklabels=corr.columns.values, square=True, annot=True)
```



Here we can see that race becomes less of a factor in treatment 1 group, except for Asian percent which actually increases. In both maps, the free lunch eligibility and `ed_lesshs` correlations stay almost the same. I think this shows just how much of a factor that external environment plays in a student's academic success. The negative correlation between white percent and black percent is the strong out of any of the variables.

Looking at correlations numerically is a good way to explore the relationship but I'd also like it see it graphically.

```
In [39]: #scatter matrix
         g = sns.set(style="white")

         g = sns.PairGrid(cor_df, diag_sharey=False)
         g.map_lower(sns.kdeplot, cmap="Blues_d")
         g.map_upper(plt.scatter)
         g.map_diag(sns.kdeplot, lw=3)
```

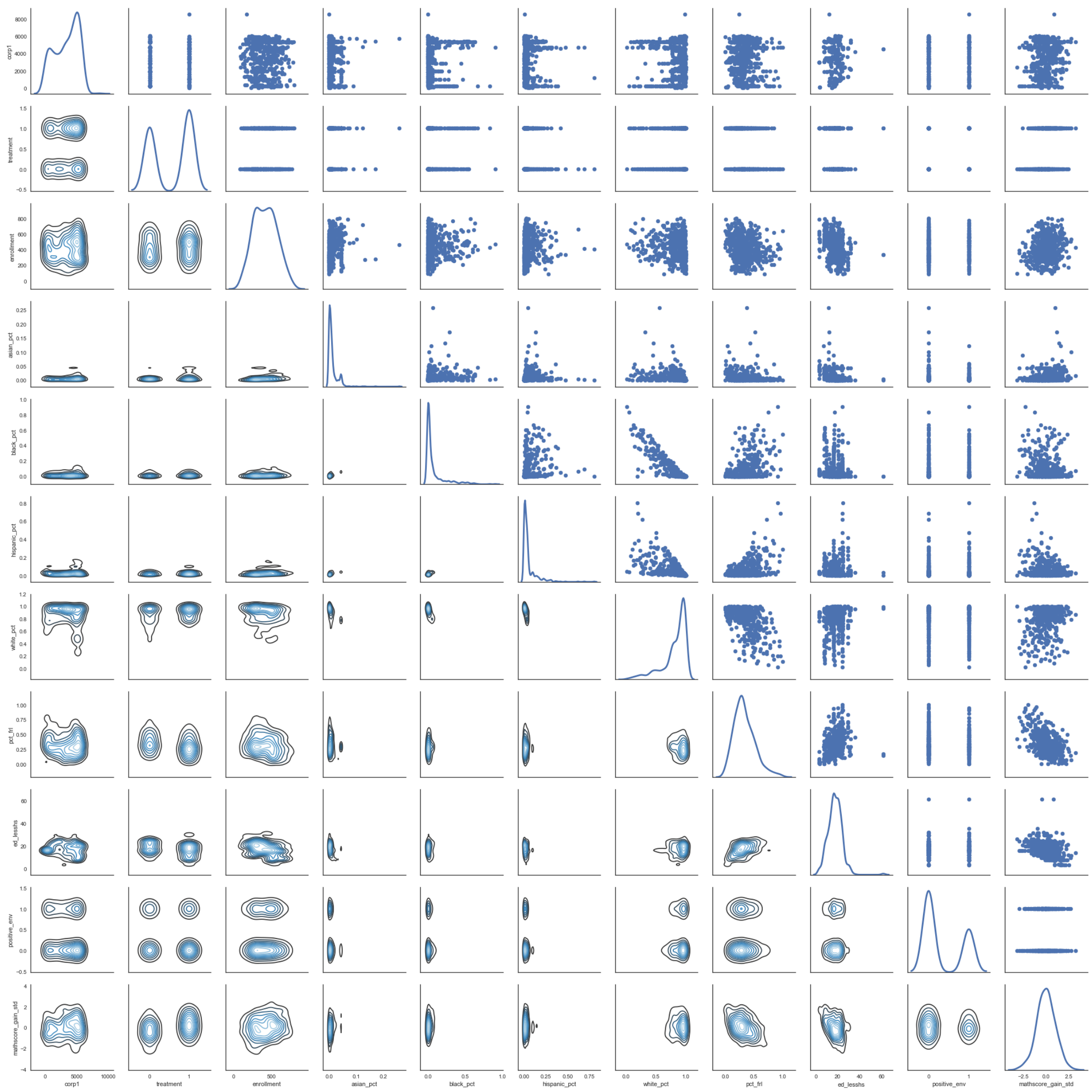
```
//anaconda/envs/py35/lib/python3.5/site-packages/matplotlib/contour.py:967: UserWarning: The fo
  s)
```

Out[39]: <seaborn.axisgrid.PairGrid at 0x1c16755438>

Out[39]: <seaborn.axisgrid.PairGrid at 0x1c16755438>

```
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
```

Out[39]: <seaborn.axisgrid.PairGrid at 0x1c16755438>

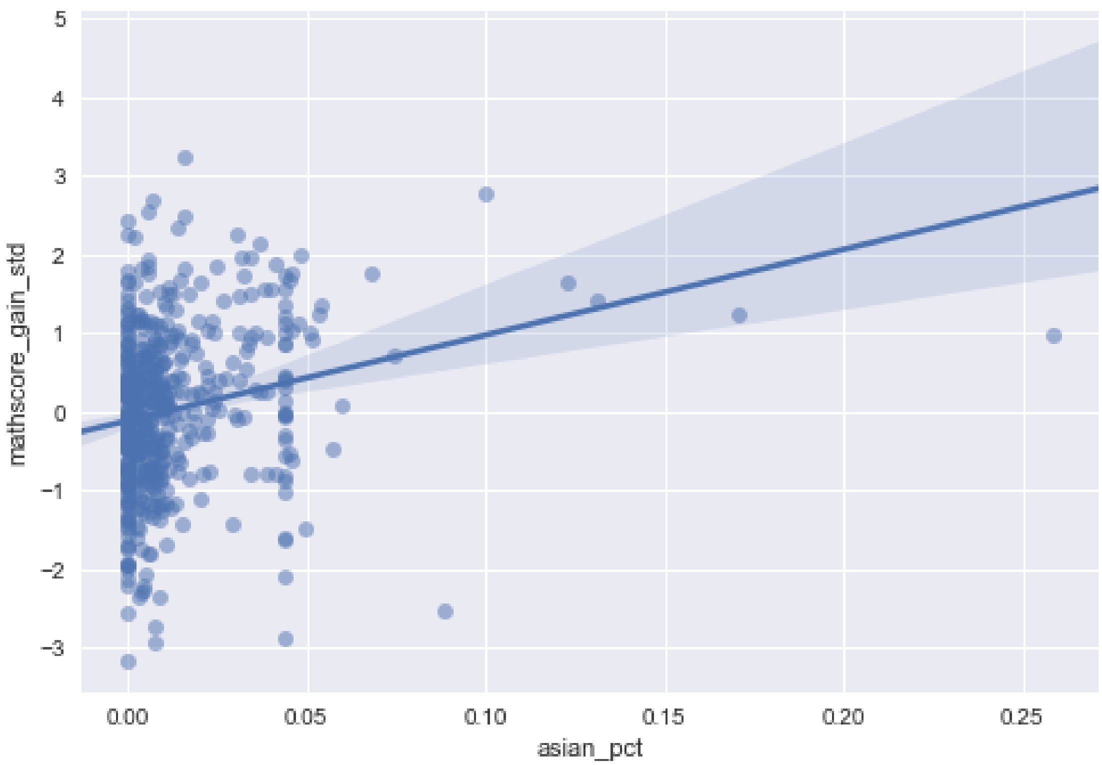


Above is a scatter matrix of the relationship between the variables in the dataset. The upper portion are scatter plots and the lower portions are bivariate density plots.

From these plots we can see the relationship between white pct and black pct more clearly. There are definitely downward trends when you look at `pct_frl` and `ed_lesshs`, each, with `mathscore_gain_std`. There doesn't seem to be too many really strong trends except for school and district ID. This makes sense but shouldn't be a real concern because school ID is just an indexer.

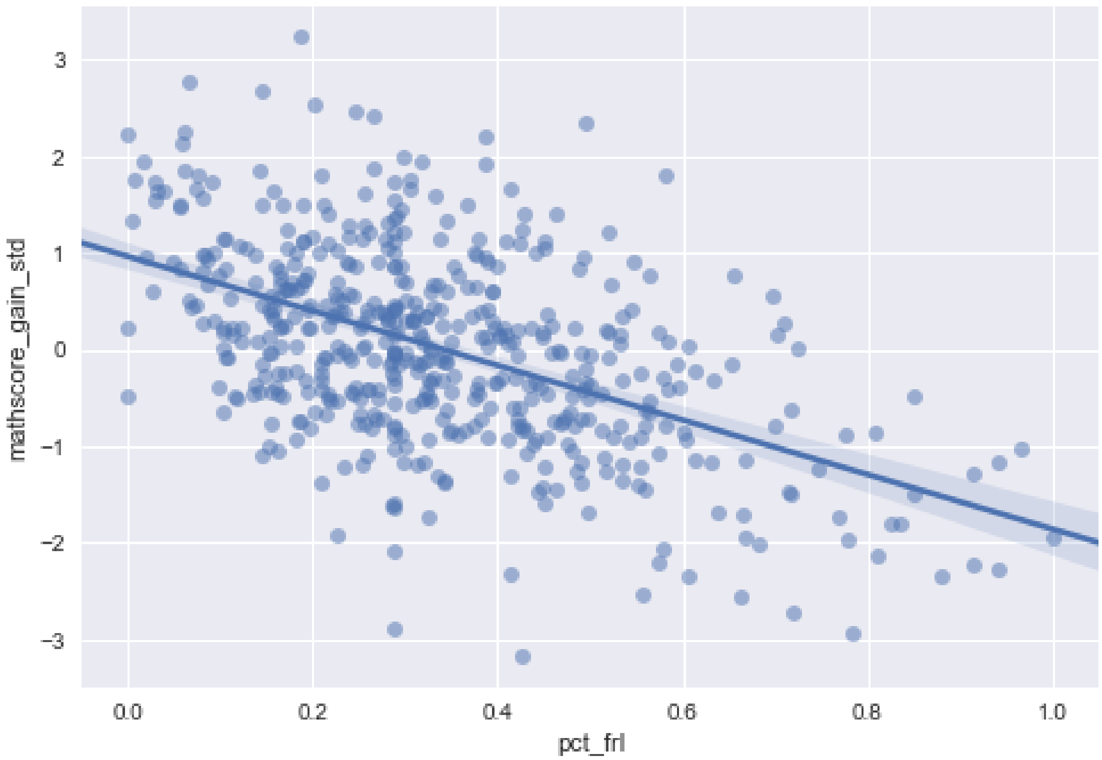
I'll take a closer look at a few of the interesting scatterplots.

```
In [69]: plt.style.use('seaborn')
         p = sns.regplot(x=df['asian_pct'], y=df['mathscore_gain_std'], scatter_kws={'alpha':0
```



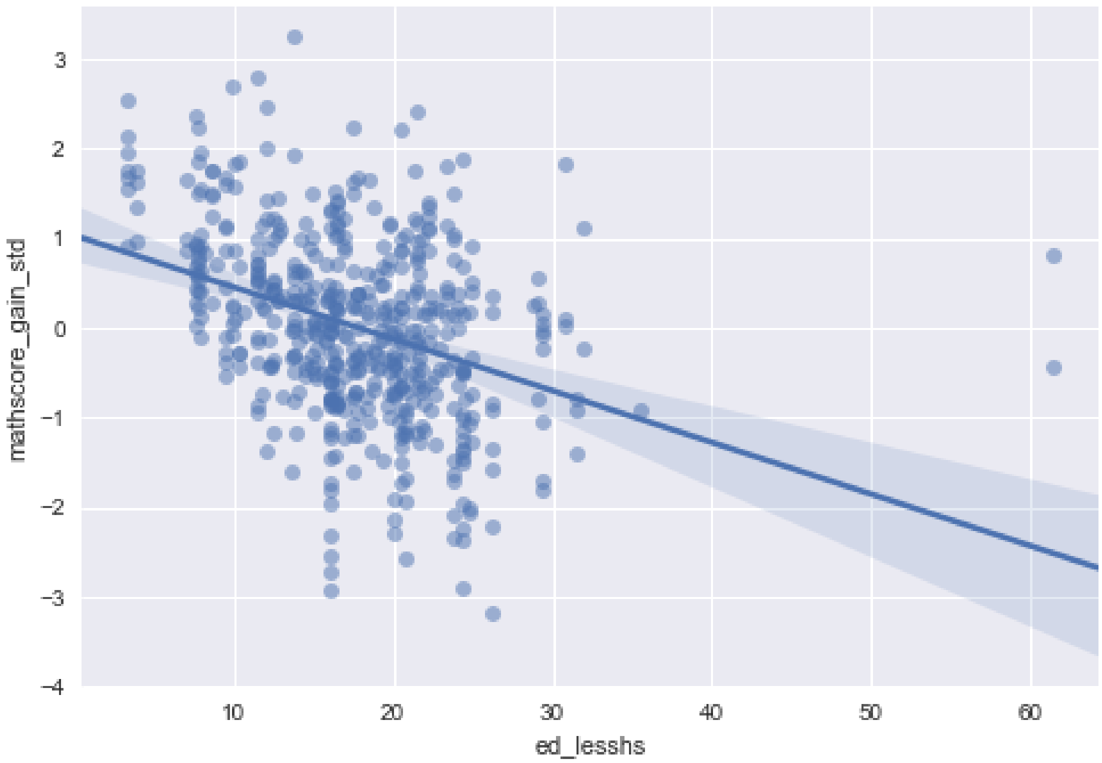
Here we can see that the correlation between asian percentage and mathscore gain. I discovered a moderate positive relationship between the two variables, however, the trend moves up due to a few outlier data points.

```
In [70]: plt.style.use('seaborn')
         p = sns.regplot(x=df['pct_frl'], y=df['mathscore_gain_std'], scatter_kws={'alpha':0.5
```



The trend here is much more obvious. There's a pretty clear negative relationship between pct_frl and math score gain.

```
In [71]: plt.style.use('seaborn')
         p = sns.regplot(x=df['ed_lesshs'], y=df['mathscore_gain_std'], scatter_kws={'alpha':0
```




```
In [73]: plt.style.use('seaborn')
         sub=df[df['ed_lesshs'] < 60]

         p = sns.regplot(x=sub['ed_lesshs'], y=sub['mathscore_gain_std'], scatter_kws={'alpha'
```


```
In [78]: print('correlation with outliers')
         test = df[['mathscore_gain_std', 'ed_lesshs']]
         r = test.corr()
         r
         print('correlation without outliers')
         sub = sub[['mathscore_gain_std', 'ed_lesshs']]
         r = sub.corr()
         r
```

```
correlation with outliers
```

Out[78]:

	mathscore_gain_std	ed_lesshs
mathscore_gain_std	1.000000	-0.362498
ed_lesshs	-0.362498	1.000000

```
correlation without outliers
```

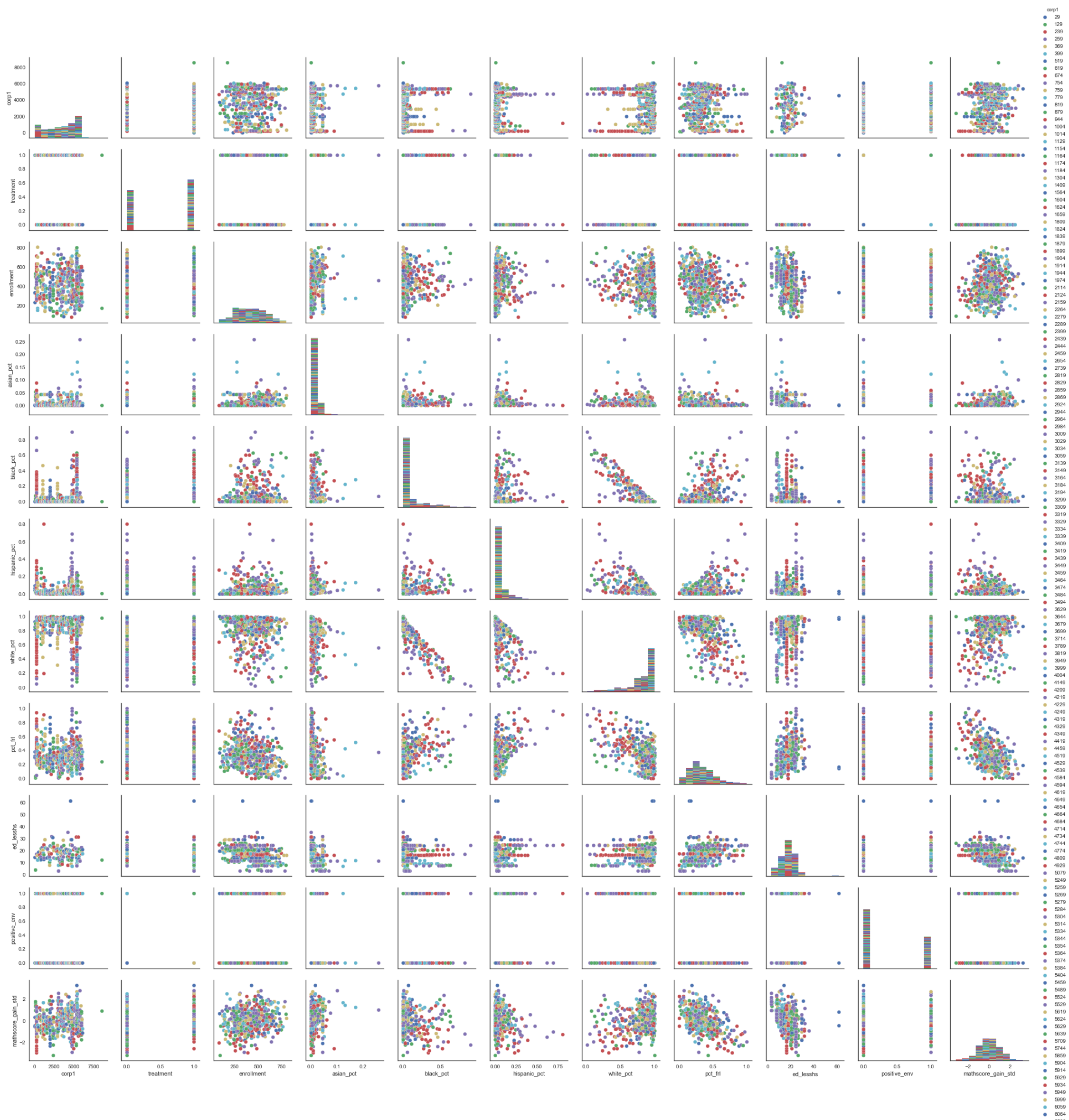
Out[78]:

	mathscore_gain_std	ed_lesshs
mathscore_gain_std	1.000000	-0.406001
ed_lesshs	-0.406001	1.000000

Here the two outlier data points that I mentioned earlier in my analysis play a role in decreasing the correlation between the `ed_lesshs` and math score. The first plot is the original, the second is without the outliers. You can see in the tables above that the correlation increases without the two outlier points.

```
In [40]: #correlogram
         sns.pairplot(cor_df, kind="scatter", hue='corp1', palette=sns.color_palette())
         plt.show()
```

```
Out[40]: <seaborn.axisgrid.PairGrid at 0x1c16bd9780>
```

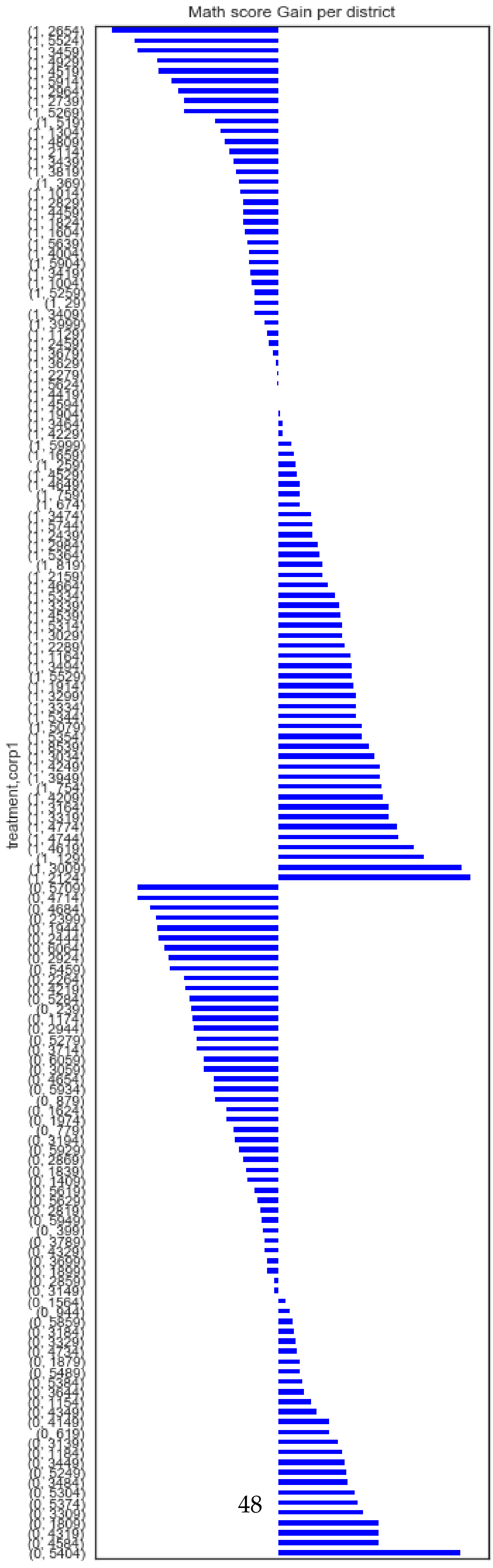


I tried to make the scatter matrix filter by district, but there are way too many districts to be able to decipher anything meaningful here. Because there are so many districts, the scatterplots are extremely overplotted. I'll try to use bar graphs instead to see the differences in the districts.

In [41]: #average math score gain by district for both treatment groups

```
table = pd.pivot_table(df, index =['treatment','corp1'], values = ['mathscore_gain_std
g = table['mathscore_gain_std'].groupby(level=0, group_keys=False)
g = g.nlargest(100000000)
g.plot(kind='barh', figsize=(5,20), colormap='winter', title='Math score Gain per dis
```

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x1c205fff98>


Math score Gain per district
treatment,corp1
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0

This plot shows the amount of math gain per district. The top half of the graph are all the districts in treatment 1 and bottom are the districts in treatment 0. Just from skimming, you can see that there are more districts and more average gain in districts in treatment 1, and there are more negative districts and more average negative gain in treatment 0.

Just out of curiosity though, I want to compare the schools with the highest positive gain and negative gain in both treatment groups.

```
In [42]: a = df[(df['corp1'] == 5709) | (df['corp1'] == 4714) | (df['corp1'] == 2654)]
         table = pd.pivot_table(index=['treatment', 'corp1'], data =a)
         table

Out[42]:                  asian_pct  black_pct  ed_lesshs  enrollment  hispanic_pct  \
         treatment corp1
         0         4714    0.008009   0.239278       24.4  455.166667      0.323919
                   5709    0.000000   0.000000       12.0  201.000000      0.000000
         1         2654    0.002257   0.000000       13.6  429.000000      0.013544

                          mathscore_gain_std   pct_frl  positive_env        schl1  \
         treatment corp1
         0         4714            -1.366431  0.610732      0.666667  4449.833333
                   5709            -1.375356  0.343348      0.000000  6110.000000
         1         2654            -1.615582  0.286041      0.000000  2150.000000

                          white_pct
         treatment corp1
         0         4714    0.374083
                   5709    0.972067
         1         2654    0.966140

In [43]: a = df[(df['corp1'] == 5404) | (df['corp1'] == 3009) | (df['corp1'] == 2124)]
         table = pd.pivot_table(index=['treatment', 'corp1'], data =a)
         table

Out[43]:                  asian_pct  black_pct  ed_lesshs  enrollment  hispanic_pct  \
         treatment corp1
         0         5404    0.091634   0.149552       12.0  230.000000      0.085522
         1         2124    0.005556   0.003333       17.5  435.500000      0.050000
                   3009    0.039835   0.047168        3.3  554.142857      0.028894

                          mathscore_gain_std   pct_frl  positive_env        schl1  \
         treatment corp1
         0         5404             1.775480  0.373259      0.000000  5892.000000
         1         2124             1.872473  0.137258      0.000000  1578.000000
                   3009             1.786763  0.068124      0.285714  2469.142857

                          white_pct
         treatment corp1
```

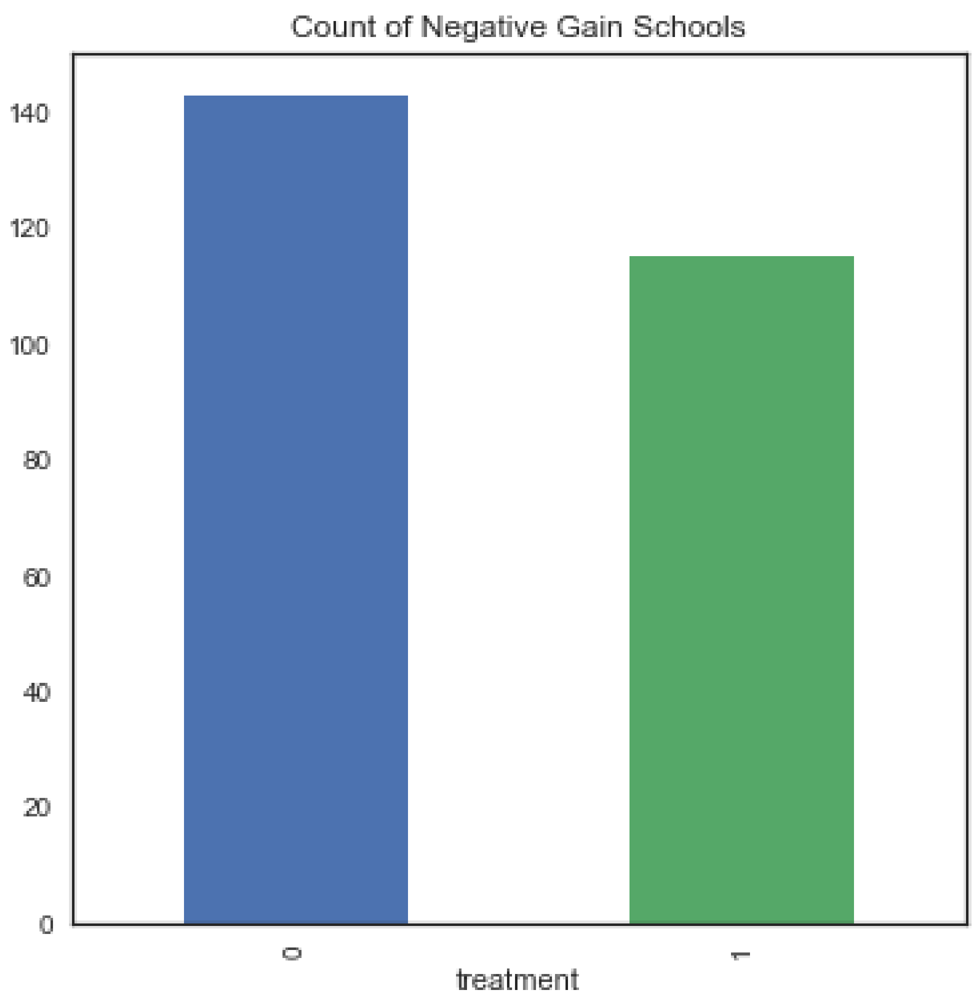
```
0          5404    0.608659
1          2124    0.936486
           3009    0.842013
```

```
In [44]: print('Count of schools where mathscore gain is less than or equal to 0 - NEGATIVE')
         a = df[df['mathscore_gain_std'] <= 0].groupby(['treatment'])['schl1'].count()
         a
         a.plot(kind = 'bar', title='Count of Negative Gain Schools', figsize=(6,6))
```

```
Count of schools where mathscore gain is less than or equal to 0 - NEGATIVE
```

```
Out[44]: treatment
         0    143
         1    115
         Name: schl1, dtype: int64
```

```
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x1c2079bac8>
```

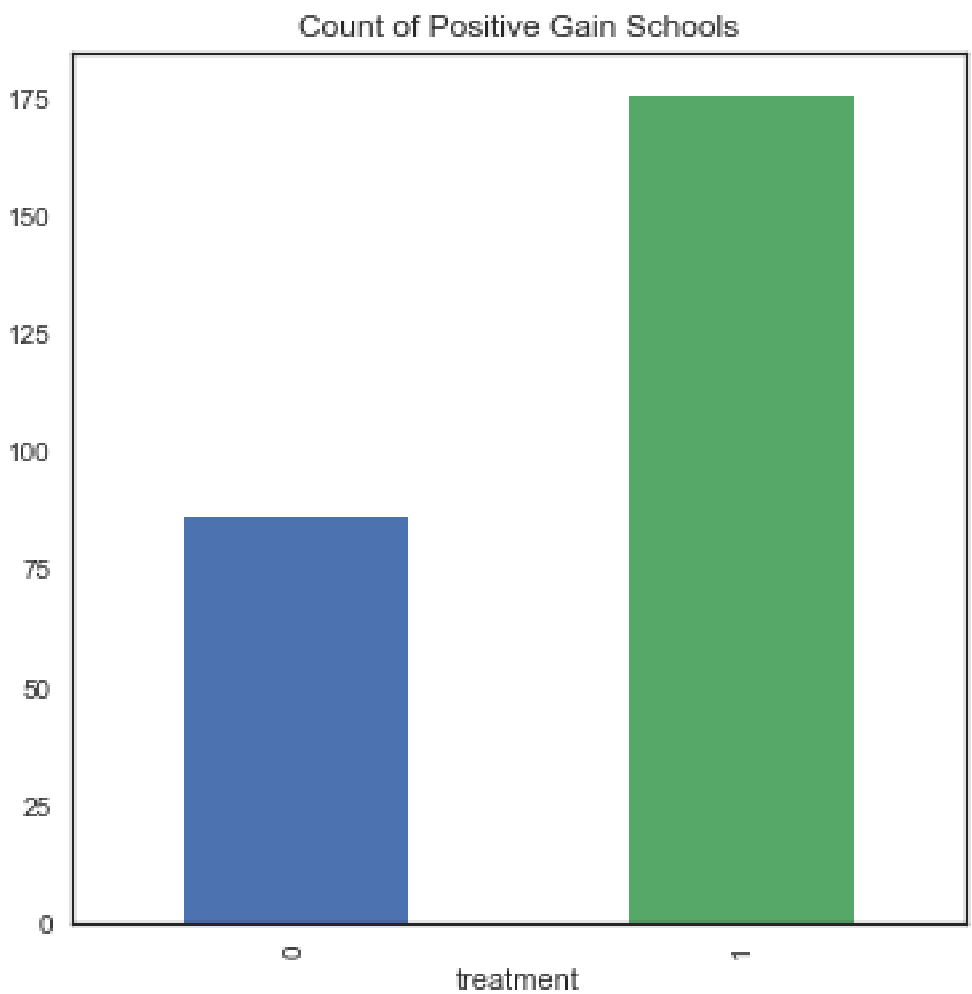


```
In [45]: print('Count of schools where mathscore gain greater than 0 - POSTIIVE')
         b = df[df['mathscore_gain_std'] > 0].groupby(['treatment'])['schl1'].count()
         b
         b.plot(kind = 'bar', title='Count of Positive Gain Schools', figsize=(6,6))
```

```
Count of schools where mathscore gain greater than 0 - POSTIIVE
```

```
Out[45]: treatment
         0     86
         1    176
         Name: schl1, dtype: int64
```

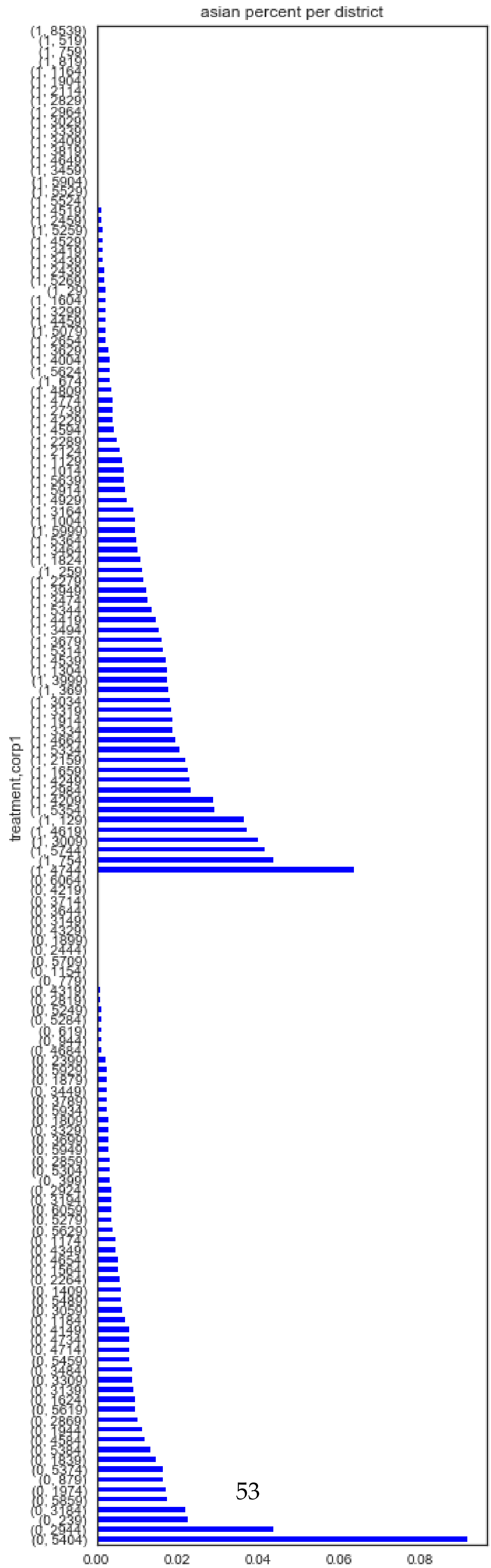
```
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x1c276b2978>
```



Above we can see that schools that did not have treatment have 143 negative gains and only 86 positive gain. On the other hand there, schools that were treated had 115 negative gain and 176 positive gain. A lot more schools math scores improved with treatment.

I'm going to continue and make a few plots to explore the districts of each treatment group as I did with the math score gain a few plots ago.

```
In [46]: table = pd.pivot_table(df, index =['treatment','corp1'], values = ['asian_pct'], aggfu
         g = table['asian_pct'].groupby(level=0, group_keys=False)
         g = g.nlargest(100000000)
         g.plot(kind='barh', figsize=(5,20), colormap='winter', title='asian percent per distri

Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x1c27e33a90>
```

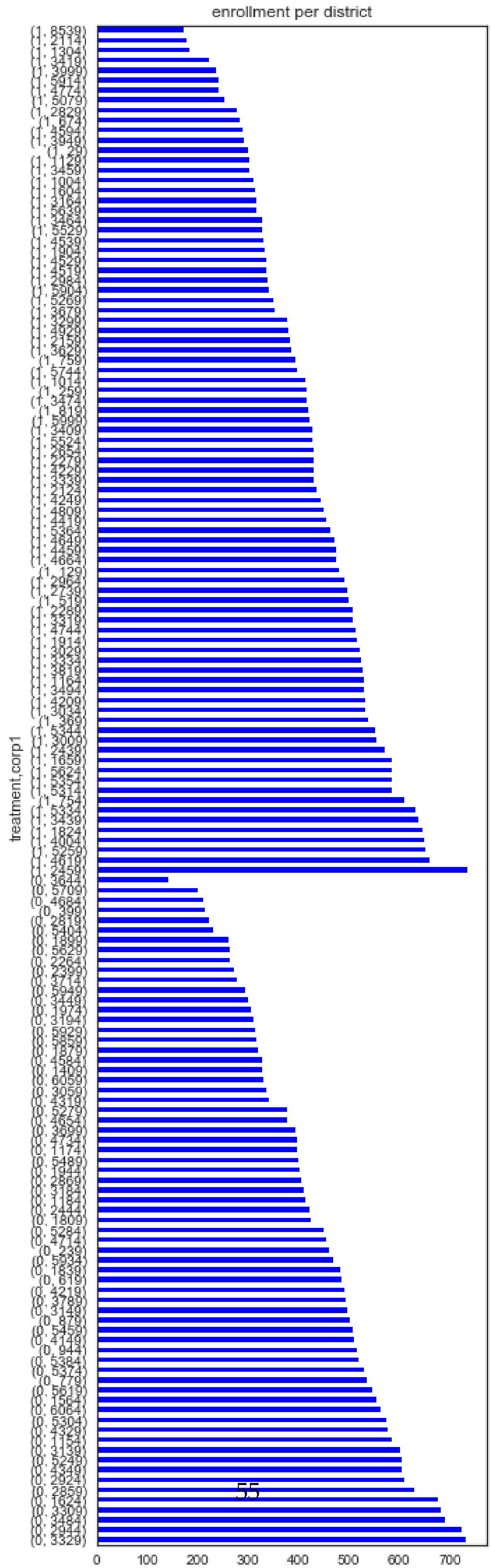

asian percent per district
treatment,corp1
0.00
0.02
0.04
0.06
0.08

Treatment 1 districts seem to have higher Asian percentage. Just want to check out the one distict that averages about 9% Asian students.

```
In [47]: df[df['corp1'] == 5404]

Out[47]:      corp1  treatment  schl1  enrollment  asian_pct  black_pct  hispanic_pct  \
         461   5404          0   5894         200   0.048544   0.087379      0.063107
         462   5404          0   5898         274   0.131387   0.226277      0.127737
         463   5404          0   5890         169   0.015873   0.000000      0.021164
         464   5404          0   5886         277   0.170732   0.284553      0.130081

              white_pct   pct_frl  ed_lesshs  positive_env  mathscore_gain_std
         461   0.733010  0.298077       12.0             0            1.995273
         462   0.459854  0.429530       12.0             0            1.415690
         463   0.920635  0.245714       12.0             0            2.470063
         464   0.321138  0.519713       12.0             0            1.220893
```

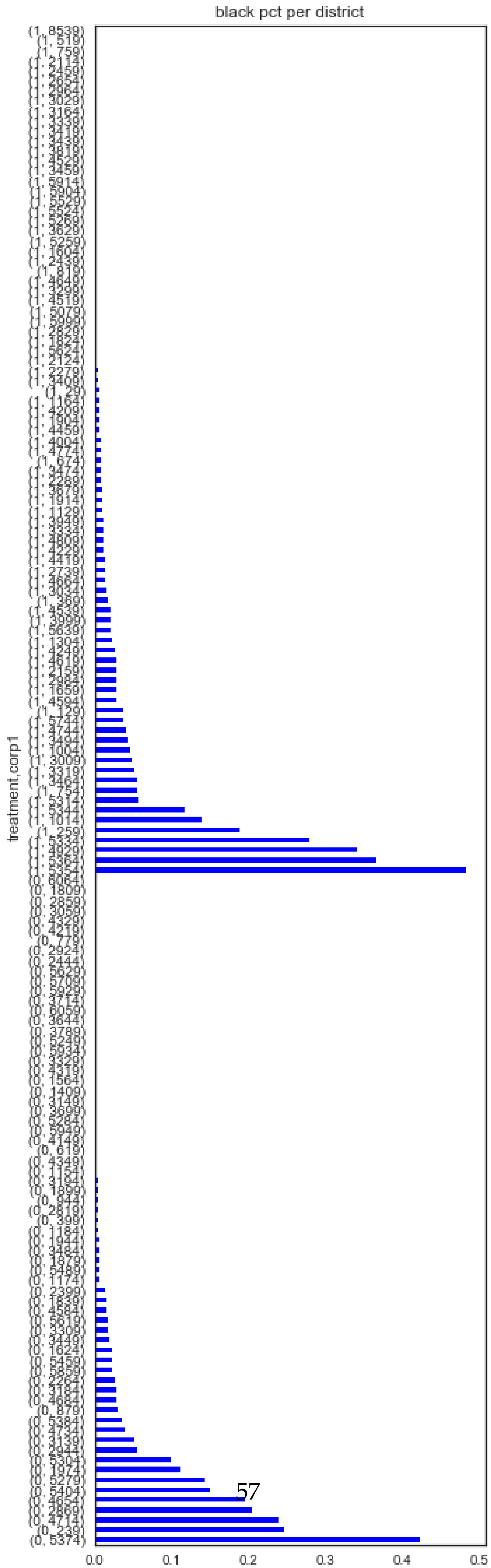
This school has all positive math scores. It looks like free lunch eligibility has an inverse relationship with math score still.

```
In [48]: table = pd.pivot_table(df, index =['treatment','corp1'], values = ['enrollment'], aggr
         g = table['enrollment'].groupby(level=0, group_keys=False)
         g = g.nlargest(100000000)
         g.plot(kind='barh', figsize=(5,20), colormap='winter', title='enrollment per district

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x1c35493080>
```


enrollment per district
treatment,corp1
0
100
200
300
400
500
600
700

Enrollment doesn't seem too different for the districts in each group.

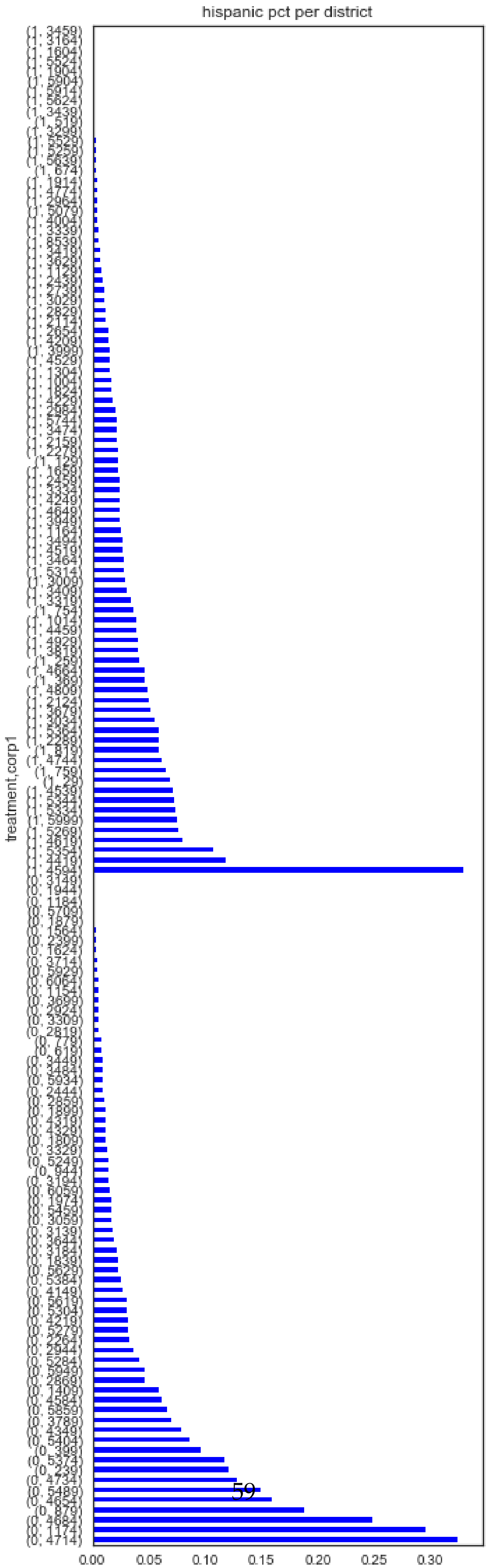
```
In [49]: table = pd.pivot_table(df, index =['treatment','corp1'], values = ['black_pct'], aggfu
         g = table['black_pct'].groupby(level=0, group_keys=False)
         g = g.nlargest(100000000)
         g.plot(kind='barh', figsize=(5,20), colormap='winter', title='black pct per district')

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x1c2aac3400>
```


black pct per district
treatment,corp1
0.0
0.1
0.2
0.3
0.4
0.5

Treatment 1 districts seem to have either very low or very high relative black percentages. Treatment 0 seems to be more in the middle of the range of black percentages.

In [50]:
```
table = pd.pivot_table(df, index =['treatment','corp1'], values = ['hispanic_pct'], ag
g = table['hispanic_pct'].groupby(level=0, group_keys=False)
g = g.nlargest(100000000)
g.plot(kind='barh', figsize=(5,20), colormap='winter', title='hispanic pct per distric
```

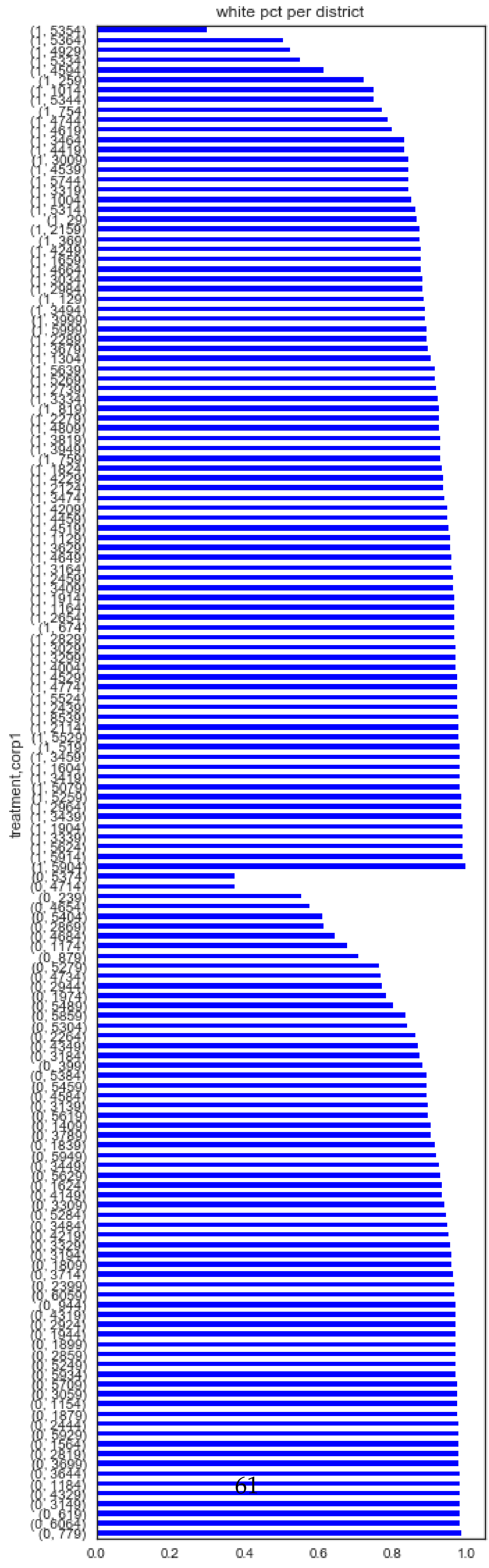
Out[50]: <matplotlib.axes._subplots.AxesSubplot at 0x1c2d093b00>


hispanic pct per district
treatment,corp1
0.00
0.05
0.10
0.15
0.20
0.25
0.30

Treatment 0 seems to have many more districts with higher hispanic percentages than treatment 1 districts.

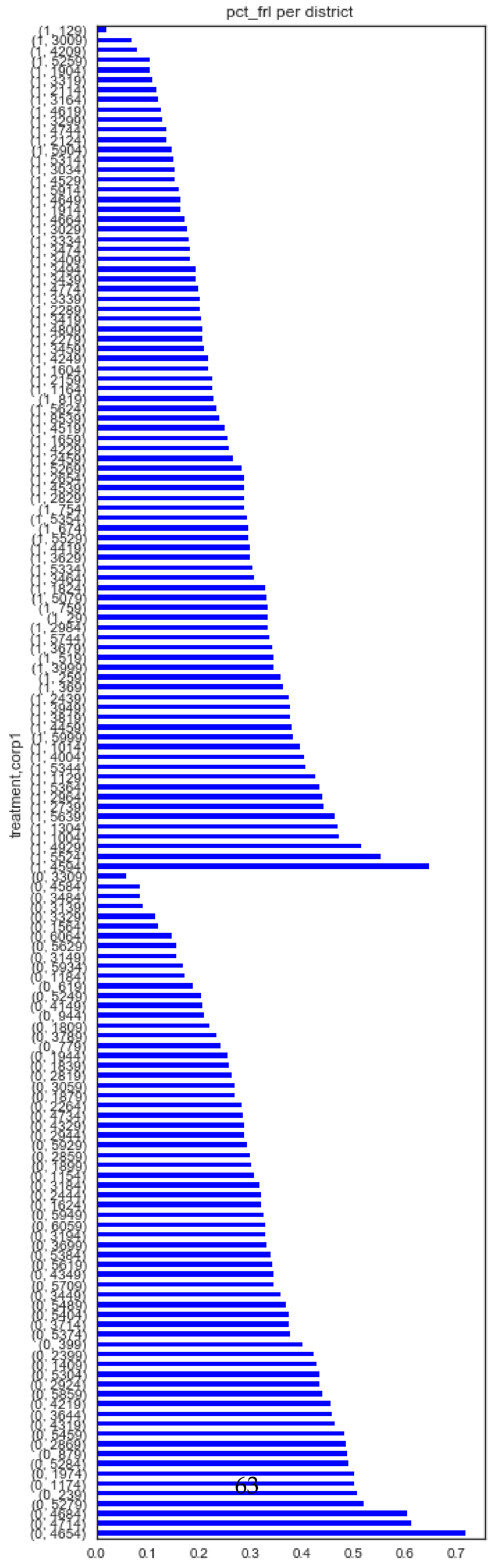
```
In [51]: table = pd.pivot_table(df, index =['treatment','corp1'], values = ['white_pct'], aggfu
         g = table['white_pct'].groupby(level=0, group_keys=False)
         g = g.nlargest(100000000)
         g.plot(kind='barh', figsize=(5,20), colormap='winter', title='white pct per district')
```

```
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x1c208f7d30>
```


white pct per district
treatment,corp1
0.0
0.2
0.4
0.6
0.8
1.0

It looks like most districts have a really high white percentage in both treatment groups.

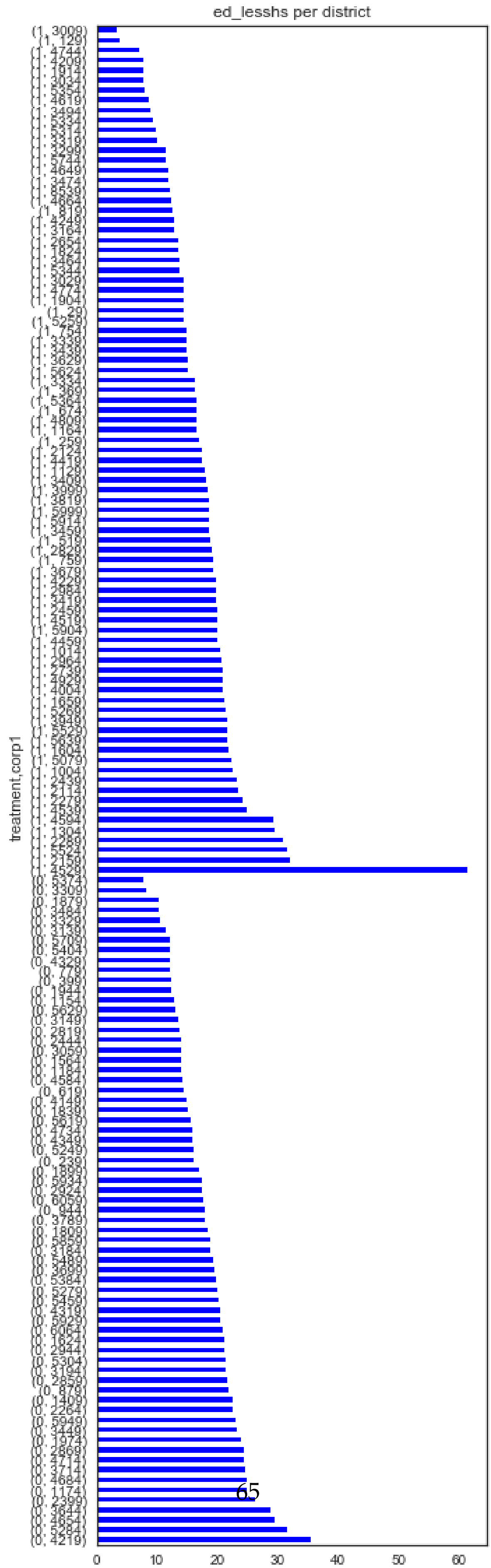
```
In [52]: table = pd.pivot_table(df, index =['treatment','corp1'], values = ['pct_frl'], aggfunc
         g = table['pct_frl'].groupby(level=0, group_keys=False)
         g = g.nlargest(100000000)
         g.plot(kind='barh', figsize=(5,20), colormap='winter', title='pct_frl per district')

Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x1c206d1978>
```


pct_frl per district
treatment,corp1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

It is a bit difficult to see any apparent differences regarding free lunch eligibility percentage in the two treatment groups based on district.

```
In [53]: table = pd.pivot_table(df, index =['treatment','corp1'], values = ['ed_lesshs'], aggfu
         g = table['ed_lesshs'].groupby(level=0, group_keys=False)
         g = g.nlargest(100000000)
         g.plot(kind='barh', figsize=(5,20), colormap='winter', title='ed_lesshs per district')
```

```
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x1c352c0b70>
```



ed_lesshs per district
treatment,corp1
0
10
20
30
40
50
60

There's one district that stands out here in the treatment 1 group. This is the same district that we saw earlier in the analysis when looking at the histogram distribution of education percentage.

Overall there doesn't seem to be too dramatic of differences between the districts within each treatment group.

1.0.3 Regression Modeling

I will do a standard multivariate regression to assess the predictive importance of the available school and district characteristics for average math test score gain in state X.

```
In [54]: reg = smf.ols('mathscore_gain_std ~ treatment + corp1 + \
                       enrollment + \
                       asian_pct + \
                       black_pct + \
                       hispanic_pct + \
                       white_pct + \
                       pct_frl + \
                       ed_lesshs', data=df).fit()

         reg.summary()
```

```
Out[54]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                     OLS Regression Results
         ==============================================================================
         Dep. Variable:     mathscore_gain_std   R-squared:                       0.343
         Model:                            OLS   Adj. R-squared:                  0.331
         Method:                 Least Squares   F-statistic:                     29.56
         Date:                Tue, 06 Mar 2018   Prob (F-statistic):           1.85e-41
         Time:                        12:08:12   Log-Likelihood:                -637.18
         No. Observations:                 520   AIC:                             1294.
         Df Residuals:                     510   BIC:                             1337.
         Df Model:                           9
         Covariance Type:            nonrobust
         ================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
         --------------------------------------------------------------------------------
         Intercept       -1.3258      1.419     -0.934      0.351      -4.114       1.462
         treatment        0.2680      0.077      3.462      0.001       0.116       0.420
         corp1         6.029e-05   1.97e-05      3.059      0.002    2.16e-05     9.9e-05
         enrollment     7.67e-05      0.000      0.291      0.771      -0.000       0.001
         asian_pct        8.7542      2.753      3.179      0.002       3.345      14.164
         black_pct        2.0826      1.565      1.331      0.184      -0.992       5.157
         hispanic_pct     2.1394      1.479      1.447      0.149      -0.765       5.044
         white_pct        2.0716      1.418      1.461      0.145      -0.714       4.858
         pct_frl         -2.0586      0.291     -7.064      0.000      -2.631      -1.486
```

```
ed_lesshs       -0.0245      0.007     -3.563      0.000      -0.038      -0.011
==============================================================================
Omnibus:                        6.577   Durbin-Watson:                   1.758
Prob(Omnibus):                  0.037   Jarque-Bera (JB):                7.951
Skew:                           0.141   Prob(JB):                       0.0188
Kurtosis:                       3.536   Cond. No.                     3.97e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly speci
[2] The condition number is large, 3.97e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

After running the regression, it looks like some of my hypotheses from my earlier analysis were confirmed. The main factors that influence math score gain are `treatment`, `corp1`, `asian_pct`, `pct_frl`, and `ed_lesshs`. That is because these variables' p-value is less than alpha of .05, which is the level of significance, so we can reject the null hypothesis that the variable has no effect on the math score gain. I had a feeling that treatment, free lunch eligibility and education less than HS would play a big role. I did not include positive environment in this regression because it is an endogenous variable. I also did not include school id because I thought of it more as a unique key indexer. The r^2 value is .343 which means that about 34.3% of the variability in math score gain is accounted for with our model's inputs. In order to correctly assess the valid use of the regression as a good model, I would look at the residual plot to make sure there's no trends there and that the distribution of residual points is random.

One very important side note that I want to discuss is that the way I used `corp1` in this regression is actually incorrect. Because district is encoded with numerical identifiers, it can be easy to mistake it as an actual numerical variable, but in really it is a categorical variable! In this particular case, the district variable is actually quite messy to deal with because as I've shown earlier in my analysis, it is extremely imbalanced as about 1/3 of the districts only have 1 school and 1 district has 31 schools. In this particular regression, `corp1` turned out to have an almost negligible coefficient even though it as a p-value of .002 showing significance. Because this project is just an initial exploration of the data and for my general curiosity, I put `corp1` in as a numerical variable just to see what would happen. In most cases to deal with categorical values I would create dummy variables for the district column in order to encode each district as numeric variables. The problem with that is the imbalance as I mentioned before; I would be left with 53 variables that only had 1 data point. If I had to do a deeper analysis of the dataset, this would be a point of curiosity for me and I would definitely research ways of handling this imbalance but I would also ask the question why this imbalance even exists in the districts in the first place.

Another problem with this model is the likely multicollinearity of the data. There are also some really strong relationships between race percentages that can be seen from the correlation matrices above. This has the potential to change the standard error of the coefficients and thus change the decision of the null hypothesis. I address this issue below by looking at the Variance Inflation Factor of the exogenous variables.

```
In [55]: sub_df = df[['treatment', 'corp1', 'enrollment', 'asian_pct', 'black_pct', 'hispanic_
                      'ed_lesshs']]
         sub_df = sm.add_constant(sub_df)
```

In [56]:
```
#Variance Inflation Factor (VIF) for multicollinearity detection
vif = pd.DataFrame()
vif["Features"] = sub_df.columns
vif["VIF Factor"] = [variance_inflation_factor(sub_df.values, i) for i in range(sub_d
vif
```

Out[56]:

	Features	VIF Factor
0	const	1512.977094
1	treatment	1.109075
2	corp1	1.018962
3	enrollment	1.223453
4	asian_pct	2.513782
5	black_pct	37.832637
6	hispanic_pct	12.590201
7	pct_frl	2.163147
8	white_pct	64.619304
9	ed_lesshs	1.449521

In [57]:
```
sub_df = df[['treatment', 'corp1', 'enrollment', 'asian_pct', 'black_pct', 'hispanic_
             'ed_lesshs']]
sub_df = sm.add_constant(sub_df)
#sub_df.head()
```

In [58]:
```
#Variance Inflation Factor (VIF) for multicollinearity detection
vif = pd.DataFrame()
vif["Features"] = sub_df.columns
vif["VIF Factor"] = [variance_inflation_factor(sub_df.values, i) for i in range(sub_d
vif
```

Out[58]:

	Features	VIF Factor
0	const	35.055448
1	treatment	1.109047
2	corp1	1.017540
3	enrollment	1.223037
4	asian_pct	1.126182
5	black_pct	1.464788
6	hispanic_pct	1.378843
7	pct_frl	2.081777
8	ed_lesshs	1.448140

The first table shows the initial VIF results with all the input variables present. I iteratively removed the feature with the largest value greater than 5. Luckily, I only had to do it once for the white percentage variable as it had collinearity with the black percentage variable. This could have been at least suspected from looking at the heatmap.

In [59]:
```
reg = smf.ols('mathscore_gain_std ~ treatment + corp1 + \
                enrollment + \
                asian_pct + \
                black_pct + \
```

```
                  hispanic_pct + \
                  pct_frl + \
                  ed_lesshs', data=df).fit()

reg.summary()
```

Out[59]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:     mathscore_gain_std   R-squared:                       0.340
Model:                            OLS   Adj. R-squared:                  0.330
Method:                 Least Squares   F-statistic:                     32.91
Date:                Tue, 06 Mar 2018   Prob (F-statistic):           8.80e-42
Time:                        12:08:13   Log-Likelihood:                -638.26
No. Observations:                 520   AIC:                             1295.
Df Residuals:                     511   BIC:                             1333.
Df Model:                           8
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        0.7234      0.216      3.345      0.001       0.299       1.148
treatment        0.2685      0.077      3.466      0.001       0.116       0.421
corp1         5.922e-05   1.97e-05      3.003      0.003    2.05e-05     9.8e-05
enrollment     6.96e-05      0.000      0.264      0.792      -0.000       0.001
asian_pct        5.7655      1.845      3.125      0.002       2.141       9.390
black_pct       -0.1588      0.308     -0.515      0.607      -0.764       0.447
hispanic_pct     0.1010      0.490      0.206      0.837      -0.861       1.063
pct_frl         -2.1412      0.286     -7.481      0.000      -2.703      -1.579
ed_lesshs       -0.0248      0.007     -3.606      0.000      -0.038      -0.011
==============================================================================
Omnibus:                        7.161   Durbin-Watson:                   1.765
Prob(Omnibus):                  0.028   Jarque-Bera (JB):                8.673
Skew:                           0.157   Prob(JB):                       0.0131
Kurtosis:                       3.550   Cond. No.                     1.97e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
[2] The condition number is large, 1.97e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

After running the regression again, I can see that the r^2 value didn't change too much with the white percentage variable removed. The coefficient for Asian percentage decreased, and all other variables' significance and coefficients stayed relatively the same.

1.0.4 Test for Random Assignment

```
In [60]: ddf['treatment'].value_counts()
         ddf.count()

Out[60]: 1    88
         0    68
         Name: treatment, dtype: int64

Out[60]: corp1        156
         treatment    156
         dtype: int64

In [61]: #question 4
         p = binom_test(88, n=156, p=0.5, alternative='two-sided')
         p

Out[61]: 0.12793767570364564
```

The researchers claimed they assigned the intervention to districts randomly. Since there are 2 outcomes we can assume that the probability of being assigned treatment 1 or 0 is .5 if the process was fair. The methodology in which they did the random assignment matters here. If we assumed they used a random number generator with probability of 50% to be 0 or 1 to generate 156 values and assigned each district based on the given random number generated, we can use a binomial test to see how likely these particular results would occur. There turned out to be 88 assignments to treatment 1 out of 156 total districts. I performed a simple two-sided binomial test to see their claim was true or not. The probability of getting these results for each treatment group turned out to be .128 or 12.8% meaning the assignment was not significantly biased towards treatment 1 or treatment 0. I conclude that indeed the assignments were random in the intervention.

1.0.5 Conclusion

In the end, I did find the intervention to be effective at least moderately. The percentage of Asian students also seemed to be a significant factor. There were definitely other important factors that affected students math score gains such as the eligibility for lunch subsidies as well as percent of local area population with less than a high school diploma. Higher education is definitely an important factor in predicting income. Thus, these factors I believe are tied into socioeconomic class of the students and the overall perception of the importance of education by parents of the students in lower socioeconomic classes. These out-of-classroom factors play just as big of a role in student's academic success as the intervention did, if not more. Even at the elementary school level, we can see the effects of these influences and cannot help but believe that they will continue to play a huge role as the students get older and go on to middle and high school.

Overall, our model was only able to predict about 34% of the variability of math score gain. I think one way to improve our model would to use a robust linear model because of the outliers that I discussed in the multivariate analysis portion. The main challenge in making the model was figuring out how to properly deal with the district variable due to its imbalanced nature in the dataset. In order to get a more accurate model I think that we would need to add more variables to discover other factors that might come into play to determine the validity of the intervention in improving students' math scores. Another thing to note is the fact that the districts implemented the intervention and not the researchers. It's highly possible that the any given district did not implement the intervention the way the researchers had envisioned. We would need to continue to ask more questions and get more relevant data.