Chicago Schooling Viz Project

Douglas Williams

10/28/2022

```
plt.figure(figsize=(15,20))

regions = plt.bar(x_bar, y_bar, color=color_lst_bar, \
        label=labels_bar)

for pos,lab in zip(scat_positions, color_labels):
    plt.scatter(pos[0], pos[1], c=lab[0], label=lab[1], s=200)

plt.scatter(0, 300, s=1475, c='#DFE0DF', alpha=0.9,\
            label='\n  4,500 Students\n')

plt.scatter(0, 300, s=688, c='#DFE0DF', alpha=0.9,\
            label='\n  2,000 Students\n')

plt.scatter(0, 450, s=344, c='#DFE0DF', alpha=0.9,\
            label='\n  1,000 Students\n')

plt.scatter(0, 450, s=172, c='#DFE0DF', alpha=0.9,\
            label='\n  500 Students\n')

img = mpimg.imread('map_test.png')
imgplot = plt.imshow(img)

plt.axis('off')
plt.xlim([500,1950])
plt.legend(facecolor='#235A6B', labelcolor='white', loc='upper right', \
           fontsize=14, borderpad=1)
plt.title('Effects of COVID-19 on Chicago Public\nHigh School Graduation Rates\n', fontsiz
#plt.savefig('CPS_Covid_Effects', format='pdf', dpi=300)
plt.show()
```

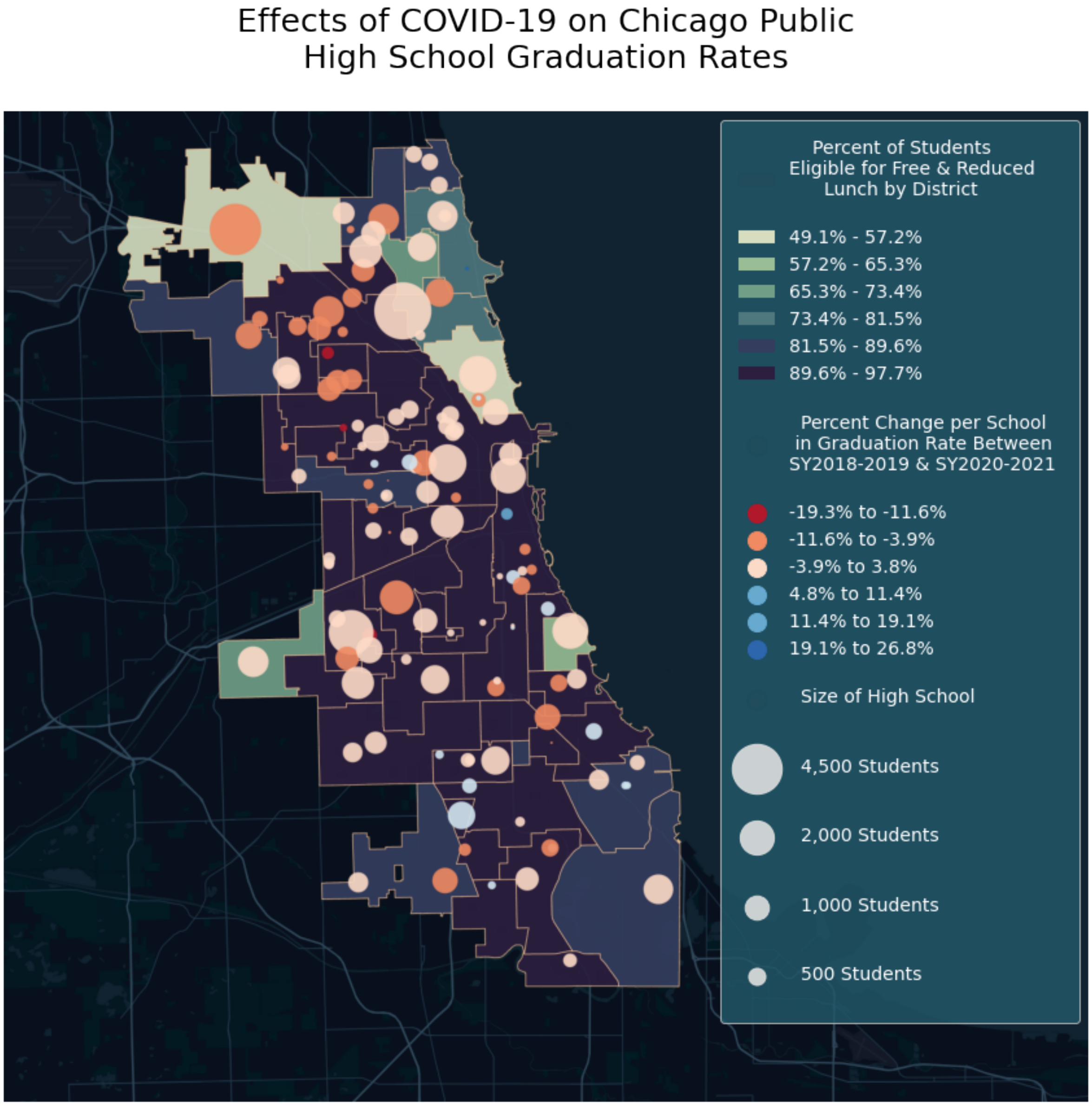

Figure 1: Data Sources: Chicago Public Schools - High School Attendance Boundaries SY1920 & SY1819. City of Chicago. https://catalog.data.gov/dataset/chicago-public-schools-high-school-attendance-boundaries-sy1920; https://catalog.data.gov/dataset/chicago-public-schools-school-locations-sy1819. Chicago Public Schools - School Locations SY1920. https://catalog.data.gov/dataset/chicago-public-schools-school-locations-sy1920. Cohort Graduation and Dropout Rates, 2020 Method. Chicago Public Schools. https://www.cps.edu/about/district-data/metrics/. Liited English Proficiency, Special Ed, Low Income, IEP Report & Racial/Ethnic Report 2018-2019, 2019-2020, 2020-2021. https://www.cps.edu/about/district-data/demographics/#a_racial-ethnic-report.

Caption

This visualization explores the effects of COVID-19 on Chicago public high schol graduation rates. Each region is a city zone assigned to a set of Chicago public high schools. The shade of these regions corresponds to the percentage of students that are considered economically disadvantaged by Chicago City Government standards (families whose income is within 18.5 percent of the federal poverty line) and are categorized as eligible for free and reduced lunch. Each dot is one of these high schools, with its size scaled proportional to the size of its student body and shade representing the change in graduation rate from the 2018-2019 to 2020-2021 school years. The neutral light pink indicates about a 4% increase of decrese over the period. This was chosen over a neutral color border at 0% change because there are always small fluxuations in graduation rates year to year. The goal is to highlight more extreme shifts over the most prevalent period of the pandemic.

Code and Notes from Production

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import matplotlib.image as mpimg
```

```
attendance_1819 = pd.read_csv("High_School_Attendance_1819.csv").rename({'School ID': 'SCH
attendance_1819 = attendance_1819[['SCHOOL_ID', 'the_geom']]
attendance_1920 = pd.read_csv("Chicago_Public_Schools_-_High_School_Attendance_Boundaries_
attendance_1920 = attendance_1920[['SCHOOL_ID', 'the_geom']]
```

```
school_loc_1819 = pd.read_csv("School_Locations_1819.csv").rename({'School_ID': 'SCHOOL_ID
school_loc_1819 = school_loc_1819[['SCHOOL_ID', 'the_geom']]
school_loc_1920 = pd.read_csv("Chicago_Public_Schools_-_School_Locations_SY1920.csv").rena
school_loc_1920 = school_loc_1920[['SCHOOL_ID', 'the_geom']]
```

```
cps_grad_data = pd.read_csv("reformatted_cps_grad_rates.csv").rename({'School ID': 'SCHOOL
cps_grad_data = cps_grad_data[cps_grad_data['SCHOOL_ID'].notna()]
cps_grad_data['SCHOOL_ID'] = np.int_(cps_grad_data['SCHOOL_ID'])
def clean_pct(array):
    cleaned_vals = []
    for val in array:
        if val == ' ':
            cleaned_vals.append(np.nan)
            continue
        cleaned_vals.append(float(val[:-1]))
    return np.array(cleaned_vals)

rates = ['dropout_rate_2017', 'dropout_rate_2018', \
         'dropout_rate_2019', 'dropout_rate_2020', \
         'dropout_rate_2021', 'dropout_rate_2022', \
         'grad_rate_2017', 'grad_rate_2018', \
         'grad_rate_2019', 'grad_rate_2020', \
         'grad_rate_2021', 'grad_rate_2022']
for col in rates:
    cps_grad_data[col] = clean_pct(cps_grad_data[col])

cps_grad_data['grad_diff_1920'] = np.subtract(cps_grad_data.grad_rate_2019, cps_grad_data.
cps_grad_data['grad_diff_2021'] = np.subtract(cps_grad_data.grad_rate_2020, cps_grad_data.
cps_grad_data['grad_diff_1921'] = np.subtract(cps_grad_data.grad_rate_2019, cps_grad_data.

cps_grad_data['drop_diff_1920'] = np.subtract(cps_grad_data.dropout_rate_2019, cps_grad_da
cps_grad_data['drop_diff_2021'] = np.subtract(cps_grad_data.dropout_rate_2020, cps_grad_da
cps_grad_data['drop_diff_1921'] = np.subtract(cps_grad_data.dropout_rate_2019, cps_grad_da

cps_grad_data = cps_grad_data.dropna(subset=['grad_diff_1920', 'grad_diff_2021','grad_diff
                                             'drop_diff_1920','drop_diff_2021', 'drop_diff

cps_grad_data = cps_grad_data[['SCHOOL_ID', 'grad_diff_1920', 'grad_diff_2021', 'grad_diff
       'drop_diff_2021', 'drop_diff_1921']]
```

1) Clean up cohort graduation and dropout rates in excel
2) Export as csv to notebook
3) Clean and merge on public high school names that we have shape files for
4) Calculate dropout changes over pandemic (2020 - 2019 rates or something)
5) Store as column in csv
6) Visualize shift in graduation rates in kepler.gl
7) *Potentially overlay demographic data as well*

Can use school ID number as a unique merger column for location, district shape, graduation rate, dropout rate, and cohort sizes across files.

```
school_loc_1819 = pd.merge(cps_grad_data, school_loc_1819, on='SCHOOL_ID', how='left').dro
school_loc_1920 = pd.merge(cps_grad_data, school_loc_1920, on='SCHOOL_ID', how='left').dro

attendance_1819 = pd.merge(cps_grad_data, attendance_1819, on='SCHOOL_ID', how='left').dro
attendance_1920 = pd.merge(cps_grad_data, attendance_1920, on='SCHOOL_ID', how='left').dro
```

WANT 2019-2019, 2019-2020, 2020-2021

Note: "Economically Disadvantaged Students" come from families whose income is within 18.5 percent of the federal poverty line. The District formerly referred to these students as "Free or Reduced Lunch Eligible Students," and adopted the new term after the federal government, under the Community Eligibility Provision, funded breakfasts and lunches for all students if more than 40 percent of students qualify.

```
def str_nums(array):
    float_nums = []
    for val in array:
        new_val = val.replace(',', '')
        new_val = new_val.strip()
        new_val = float(new_val)
        float_nums.append(new_val)
    return np.array(float_nums)
```

```
EDS_1819 = pd.read_csv("reduced_lunch_1819.csv").rename({'School ID': 'SCHOOL_ID'}, axis=1

EDS_1819['pct_eds_1819'] = clean_pct(EDS_1819['pct_eds_1819'])
EDS_1819['Total_1819'] = str_nums(EDS_1819['Total_1819'])
EDS_1819['SCHOOL_ID'] = np.int_(EDS_1819['SCHOOL_ID'])

EDS_1920 = pd.read_csv("reduced_lunch_1920.csv").rename({'School ID': 'SCHOOL_ID'}, axis=1

EDS_1920['pct_eds_1920'] = clean_pct(EDS_1920['pct_eds_1920'])
EDS_1920['Total_1920'] = str_nums(EDS_1920['Total_1920'])
EDS_1920['SCHOOL_ID'] = np.int_(EDS_1920['SCHOOL_ID'])

EDS_2021 = pd.read_csv("reduced_lunch_2021.csv").rename({'School ID': 'SCHOOL_ID'}, axis=1

EDS_2021['pct_eds_2021'] = clean_pct(EDS_2021['pct_eds_2021'])
EDS_2021['Total_2021'] = str_nums(EDS_2021['Total_2021'])
EDS_1920['SCHOOL_ID'] = np.int_(EDS_1920['SCHOOL_ID'])
```

```
racial_demos_1819 = pd.read_csv("racial_demos_1819.csv").rename({'School ID': 'SCHOOL_ID'}

racial_demos_1819['SCHOOL_ID'] = np.int_(racial_demos_1819['SCHOOL_ID'])

racial_demos_1920 = pd.read_csv("racial_demos_1920.csv").rename({'School ID': 'SCHOOL_ID'}

racial_demos_1920['SCHOOL_ID'] = np.int_(racial_demos_1920['SCHOOL_ID'])

racial_demos_2021 = pd.read_csv("racial_demos_2021.csv").rename({'School ID': 'SCHOOL_ID'}

racial_demos_2021['SCHOOL_ID'] = np.int_(racial_demos_2021['SCHOOL_ID'])
```

Want to get the average percentages and numbers of students on free and reduced lunch between 2018 and 2021

```
EDS_data = pd.merge(EDS_1819, EDS_1920, on='SCHOOL_ID', how='left')
EDS_data = pd.merge(EDS_data, EDS_2021, on='SCHOOL_ID', how='left')
EDS_data['avg_total_1821'] = np.round((EDS_data.Total_1819 + EDS_data.Total_1920 + EDS_dat
EDS_data['avg_pct_eds_1821'] = np.round((EDS_data.pct_eds_1819 + EDS_data.pct_eds_1920 + E
EDS_data = EDS_data[['SCHOOL_ID', 'avg_total_1821', 'avg_pct_eds_1821']]
```

```
def pct_non_white(white, unknown):
    non_white = 100 - (white + unknown)
    return non_white

racial_demos_1819['pct_nonwhite_1819'] = pct_non_white(racial_demos_1819.pct_white_1819, r
racial_demos_1920['pct_nonwhite_1920'] = pct_non_white(racial_demos_1920.pct_white_1920, r
racial_demos_2021['pct_nonwhite_2021'] = pct_non_white(racial_demos_2021.pct_white_2021, r
```

```
race_demo_data = pd.merge(racial_demos_1819, racial_demos_1920, on='SCHOOL_ID', how='left'
race_demo_data = pd.merge(race_demo_data, racial_demos_2021, on='SCHOOL_ID', how='left')
race_demo_data['avg_pct_nonwhite_1821'] = np.round(((race_demo_data.pct_nonwhite_1819 + ra
race_demo_data = race_demo_data[['SCHOOL_ID', 'avg_pct_nonwhite_1821']]
```

```
EDS_data.shape, race_demo_data.shape
```

```
((657, 3), (657, 2))
```

```
demo_eds_plot = pd.merge(EDS_data, race_demo_data, on='SCHOOL_ID')
```

```
attendance_1819 = pd.merge(attendance_1819, demo_eds_plot, on='SCHOOL_ID', how='left')
attendance_1920 = pd.merge(attendance_1920, demo_eds_plot, on='SCHOOL_ID', how='left')

school_loc_1819 = pd.merge(school_loc_1819, demo_eds_plot, on='SCHOOL_ID', how='left')
school_loc_1920 = pd.merge(school_loc_1920, demo_eds_plot, on='SCHOOL_ID', how='left')
```

```
school_loc_1819.to_csv('school_loc_1819_plot.csv', na_rep='NaN')
school_loc_1920.to_csv('school_loc_1920_plot.csv', na_rep='NaN')

attendance_1819.to_csv('attendance_1819_plot.csv', na_rep='NaN')
attendance_1920.to_csv('attendance_1920_plot.csv', na_rep='NaN')
```

I then used Kepler (https://kepler.gl) to visualize and layer the spatial data on top of a map of Chicago. Once I got that image, I exported it here and used Matplotlib to tune it, add labels, and create the legend.

```
class txt_style:
    ITALIC = "\x1B[3m"
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'
    END = '\033[0m'
```

```
x_bar = [400, 400, 401, 402, 403, 404, 405]
y_bar = [19, 20, 20, 20, 20, 20, 20]
color_lst_bar = ['#204c5c', '#d6debf', '#99be95', '#709e87',\
                '#4f777e', '#343d5e', '#2b1e3e']
labels_bar = ['    Percent of Students\nEligible for Free & Reduced \n      Lunch by Distr
             '73.4% - 81.5%', '81.5% - 89.6%', '89.6% - 97.7%']
```

```
for color in ['tab:blue', 'tab:orange', 'tab:green']:
    n = 750
    x, y = np.random.rand(2, n)
    scale = 200.0 * np.random.rand(n)
    ax.scatter(x, y, c=color, s=scale, label=color,
               alpha=0.3, edgecolors='none')
```

```
x_scat = [400, 400, 400, 400, 400, 400]
y_scat = [20, 21, 22, 23, 24, 25]
```

```
color_lst_scat = ['#b2182b', '#ef8a62', '#fbdbc7',\
                 '#d1e5f0', '#67a9cf', '#2d66ac']
labels_scat = ['-19.3% to -11.6%', '-11.6% to -3.9%',\
              '-3.9% to 3.8%', '4.8% to 11.4%',\
              '11.4% to 19.1%', '19.1% to 26.8%']

scat_positions = [(400,19), (400,20), (400,21), (400,22),\
                 (400,23), (400,24), (400,25), (400,26)]
color_labels = [('#204c5c', '\n  Percent Change per School\n in Graduation Rate Between\nS
                ('#b2182b','-19.3% to -11.6%'), ('#ef8a62','-11.6% to -3.9%'),\
               ('#fbdbc7','-3.9% to 3.8%'), ('#67a9cf','4.8% to 11.4%'),\
               ('#67a9cf','11.4% to 19.1%'), ('#2d66ac','19.1% to 26.8%'), \
               ('#204c5c', '\n  Size of High School \n')]
```

```
bubbles = school_loc_1819.avg_total_1821
bubbles.min(), bubbles.mean(), bubbles.max()
```

```
(52.33, 781.3224603174604, 4483.33)
```

Can add multiple legend titles with dummy plots of same background color as legend then bold title