Sunny Futures STEM

Anterra Kennedy
Sasha Prokhorova
Nick Horton

Table of Contents

01. Mission

02. Plan

03. Research

04. Findings

05. Results

06. Conclusion

The Goal...

To create long-term improvement in the lives of underprivileged kids by holding fun science demonstrations at outreach events to get them excited about and engaged with STEM.

Sunny Futures STEM

How can we reach the greatest total number of at-risk kids?

Methodology:

01. Identify underperforming schools

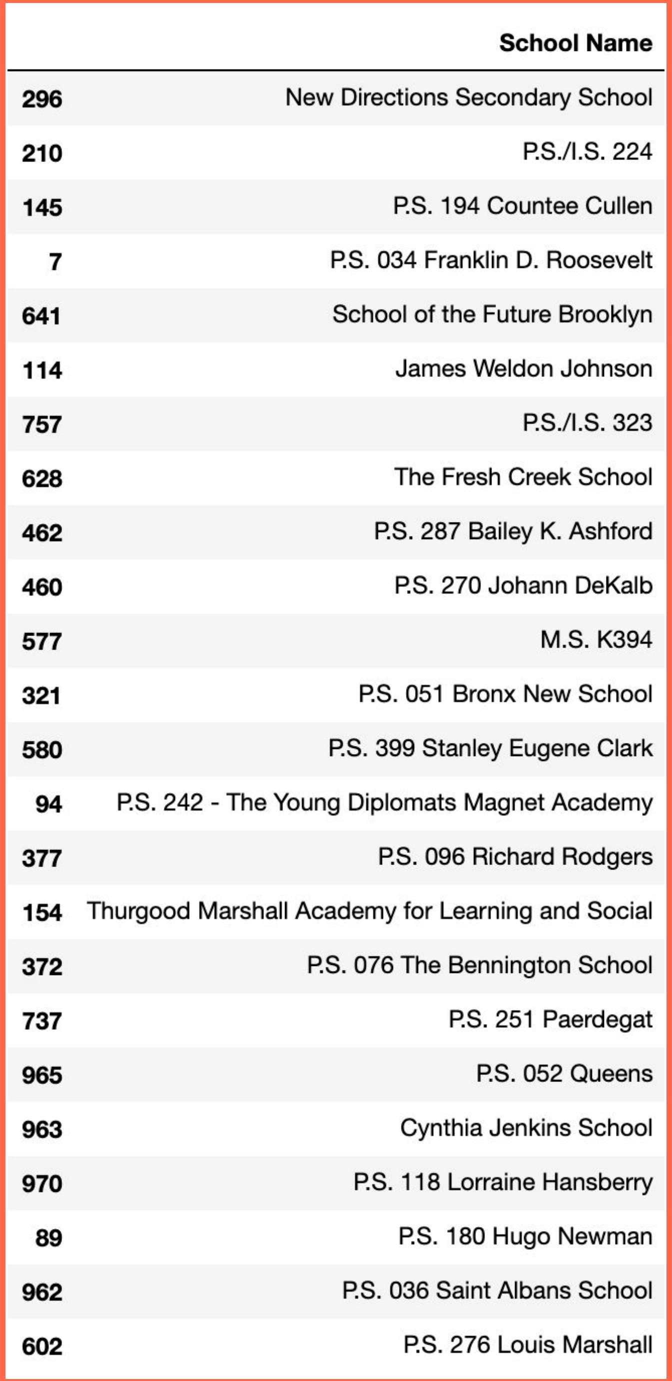
02. Find nearest subway stations

03. Analyze which stations have the highest weekday afternoon traffic

Where are at-risk kids?

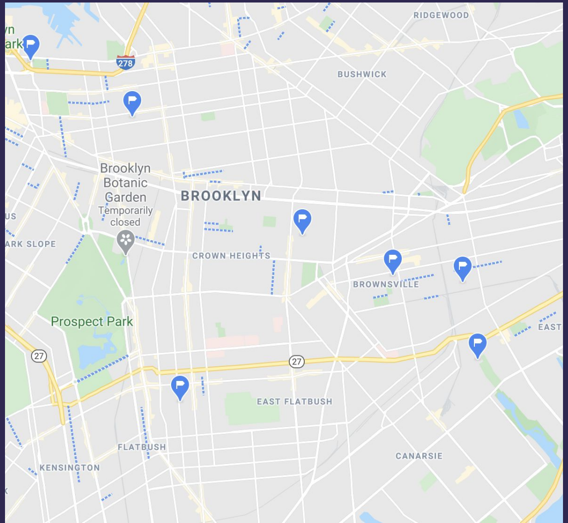
We used NYC Dept of Education's School Performance Data...

... to find the top 20 neediest and worst-performing schools on the metrics of student achievement and economic need index.

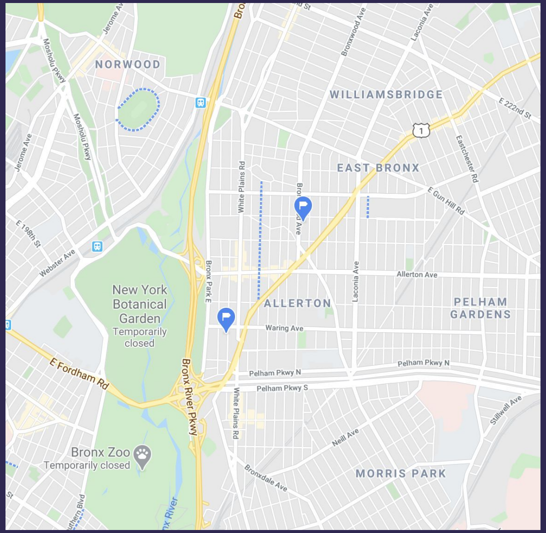


	School Name
296	New Directions Secondary School
210	P.S./I.S. 224
145	P.S. 194 Countee Cullen
7	P.S. 034 Franklin D. Roosevelt
641	School of the Future Brooklyn
114	James Weldon Johnson
757	P.S./I.S. 323
628	The Fresh Creek School
462	P.S. 287 Bailey K. Ashford
460	P.S. 270 Johann DeKalb
577	M.S. K394
321	P.S. 051 Bronx New School
580	P.S. 399 Stanley Eugene Clark
94	P.S. 242 - The Young Diplomats Magnet Academy
377	P.S. 096 Richard Rodgers
154	Thurgood Marshall Academy for Learning and Social
372	P.S. 076 The Bennington School
737	P.S. 251 Paerdegat
965	P.S. 052 Queens
963	Cynthia Jenkins School
970	P.S. 118 Lorraine Hansberry
89	P.S. 180 Hugo Newman
962	P.S. 036 Saint Albans School
602	P.S. 276 Louis Marshall

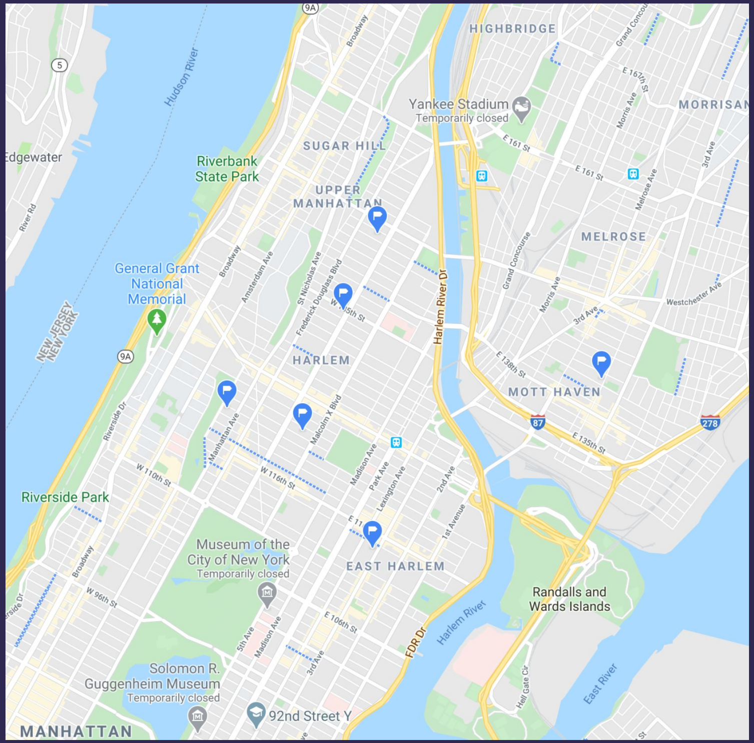
Many under-performing schools are nearby…



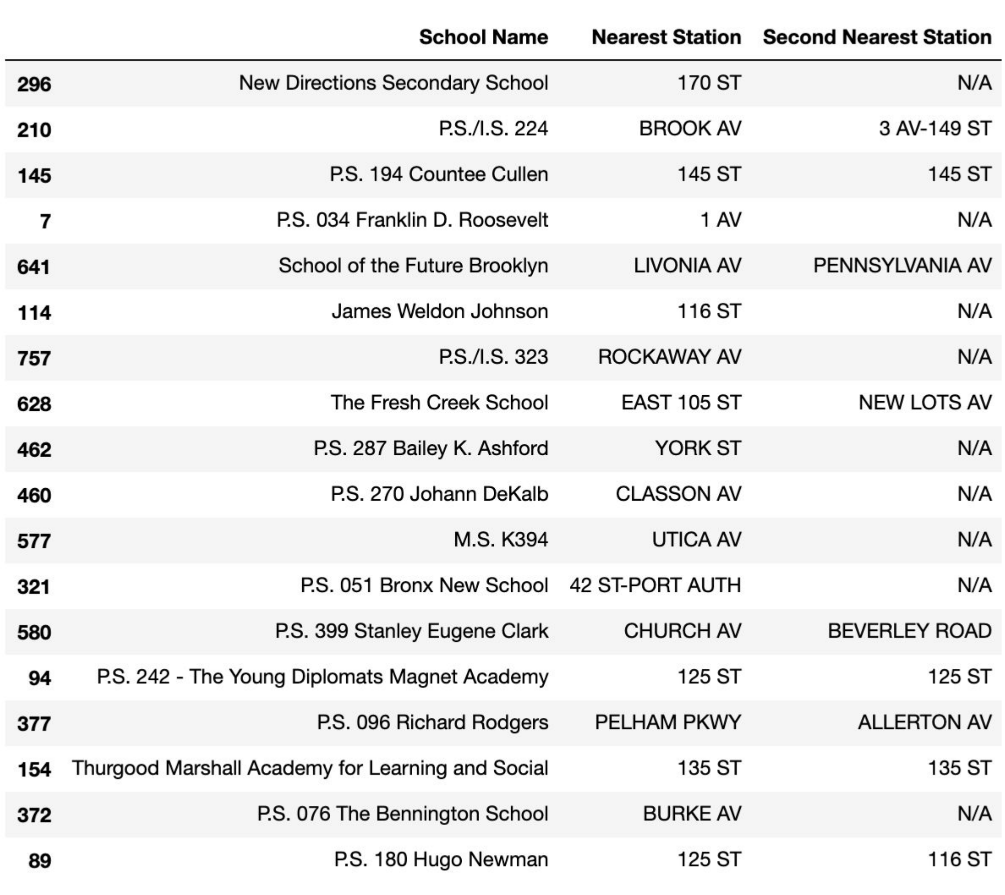
East Brooklyn



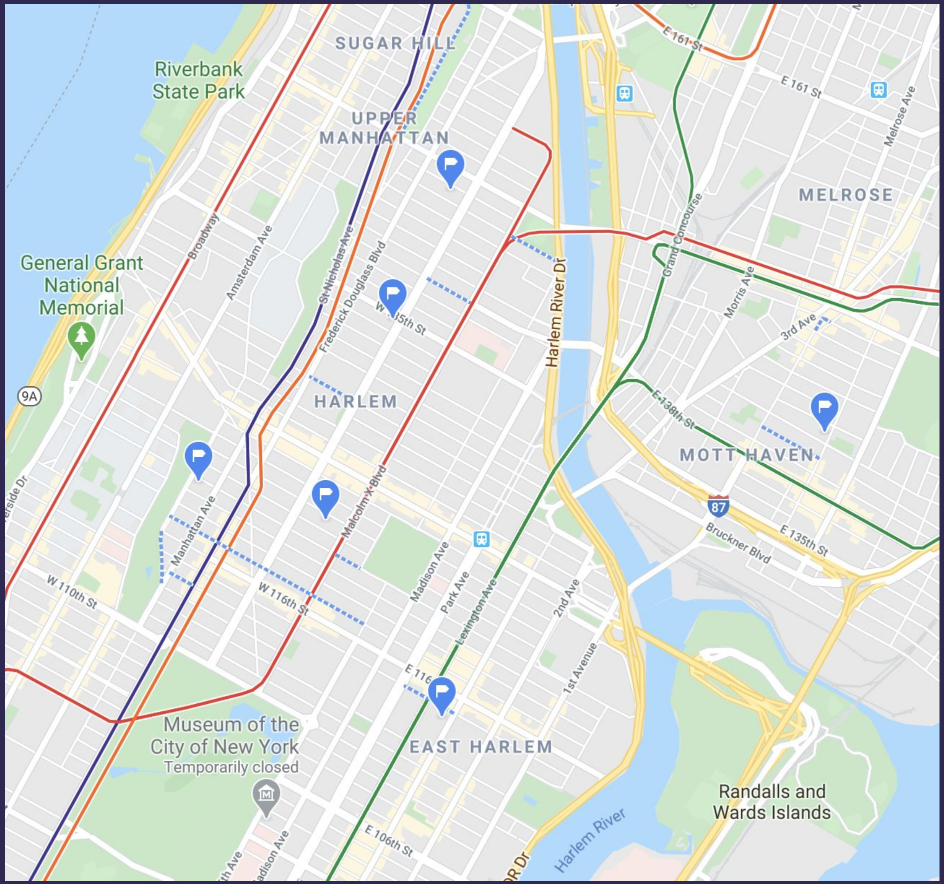
East Bronx



Harlem



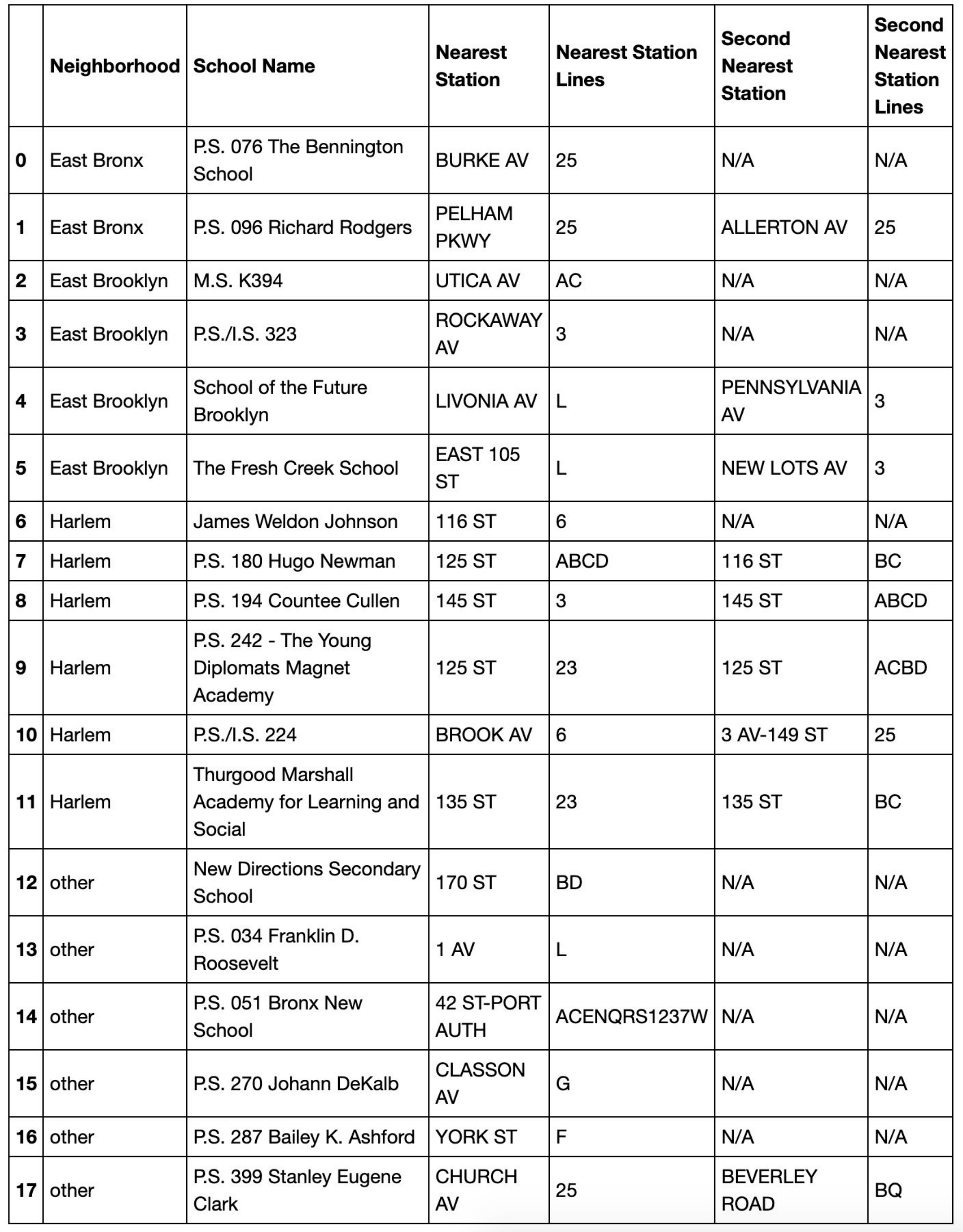
	School Name	Nearest Station	Second Nearest Station
296	New Directions Secondary School	170 ST	N/A
210	P.S./I.S. 224	BROOK AV	3 AV-149 ST
145	P.S. 194 Countee Cullen	145 ST	145 ST
7	P.S. 034 Franklin D. Roosevelt	1 AV	N/A
641	School of the Future Brooklyn	LIVONIA AV	PENNSYLVANIA AV
114	James Weldon Johnson	116 ST	N/A
757	P.S./I.S. 323	ROCKAWAY AV	N/A
628	The Fresh Creek School	EAST 105 ST	NEW LOTS AV
462	P.S. 287 Bailey K. Ashford	YORK ST	N/A
460	P.S. 270 Johann DeKalb	CLASSON AV	N/A
577	M.S. K394	UTICA AV	N/A
321	P.S. 051 Bronx New School	42 ST-PORT AUTH	N/A
580	P.S. 399 Stanley Eugene Clark	CHURCH AV	BEVERLEY ROAD
94	P.S. 242 - The Young Diplomats Magnet Academy	125 ST	125 ST
377	P.S. 096 Richard Rodgers	PELHAM PKWY	ALLERTON AV
154	Thurgood Marshall Academy for Learning and Social	135 ST	135 ST
372	P.S. 076 The Bennington School	BURKE AV	N/A
89	P.S. 180 Hugo Newman	125 ST	116 ST



... and close to subway stations.

p / 5

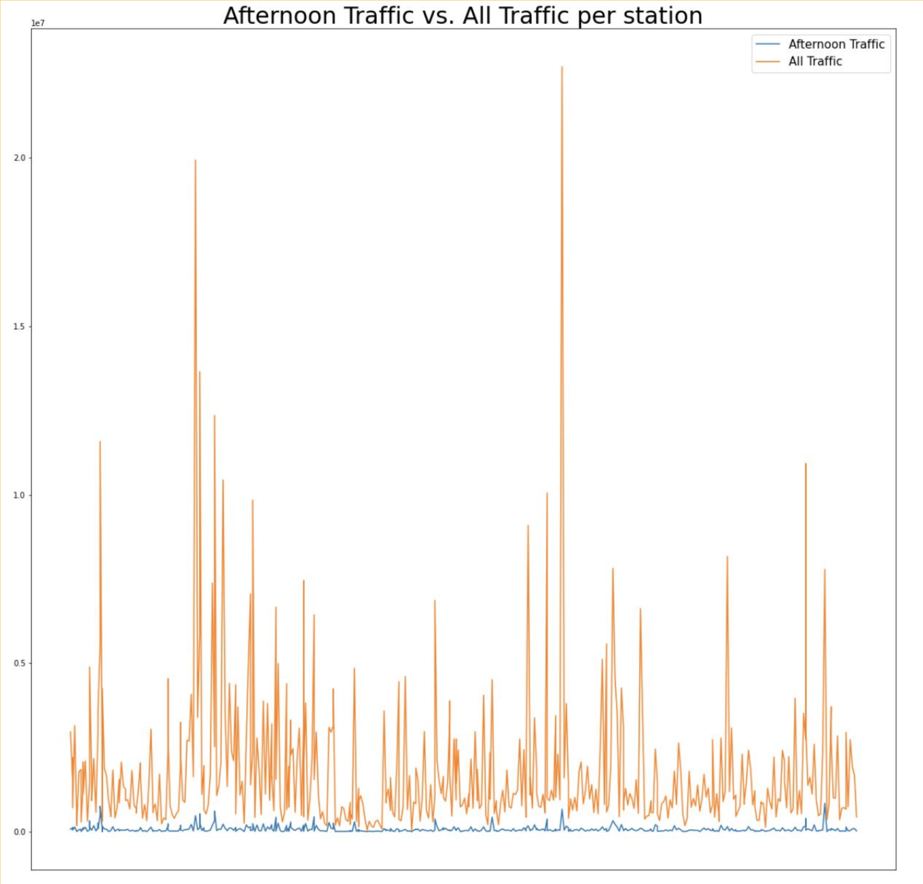
Stations with duplicate names service different subway lines.

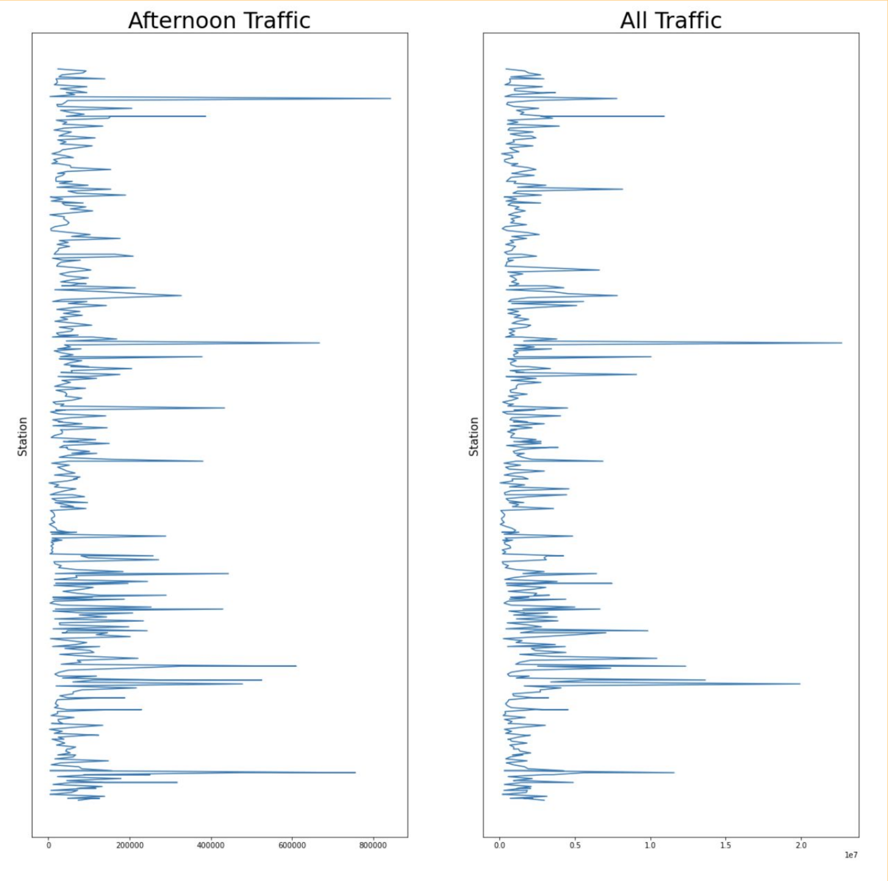


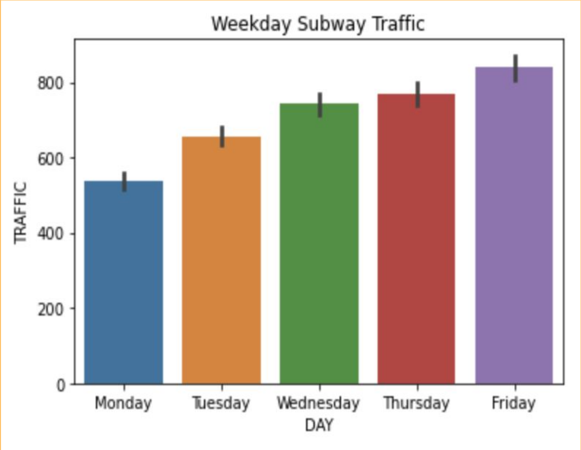
	Neighborhood	School Name	Nearest Station	Nearest Station Lines	Second Nearest Station	Second Nearest Station Lines
0	East Bronx	P.S. 076 The Bennington School	BURKE AV	25	N/A	N/A
1	East Bronx	P.S. 096 Richard Rodgers	PELHAM PKWY	25	ALLERTON AV	25
2	East Brooklyn	M.S. K394	UTICA AV	AC	N/A	N/A
3	East Brooklyn	P.S./I.S. 323	ROCKAWAY AV	3	N/A	N/A
4	East Brooklyn	School of the Future Brooklyn	LIVONIA AV	L	PENNSYLVANIA AV	3
5	East Brooklyn	The Fresh Creek School	EAST 105 ST	L	NEW LOTS AV	3
6	Harlem	James Weldon Johnson	116 ST	6	N/A	N/A
7	Harlem	P.S. 180 Hugo Newman	125 ST	ABCD	116 ST	BC
8	Harlem	P.S. 194 Countee Cullen	145 ST	3	145 ST	ABCD
9	Harlem	P.S. 242 - The Young Diplomats Magnet Academy	125 ST	23	125 ST	ACBD
10	Harlem	P.S./I.S. 224	BROOK AV	6	3 AV-149 ST	25
11	Harlem	Thurgood Marshall Academy for Learning and Social	135 ST	23	135 ST	BC
12	other	New Directions Secondary School	170 ST	BD	N/A	N/A
13	other	P.S. 034 Franklin D. Roosevelt	1 AV	L	N/A	N/A
14	other	P.S. 051 Bronx New School	42 ST-PORT AUTH	ACENQRS1237W	N/A	N/A
15	other	P.S. 270 Johann DeKalb	CLASSON AV	G	N/A	N/A
16	other	P.S. 287 Bailey K. Ashford	YORK ST	F	N/A	N/A
17	other	P.S. 399 Stanley Eugene Clark	CHURCH AV	25	BEVERLEY ROAD	BQ

MTA Turnstile Data

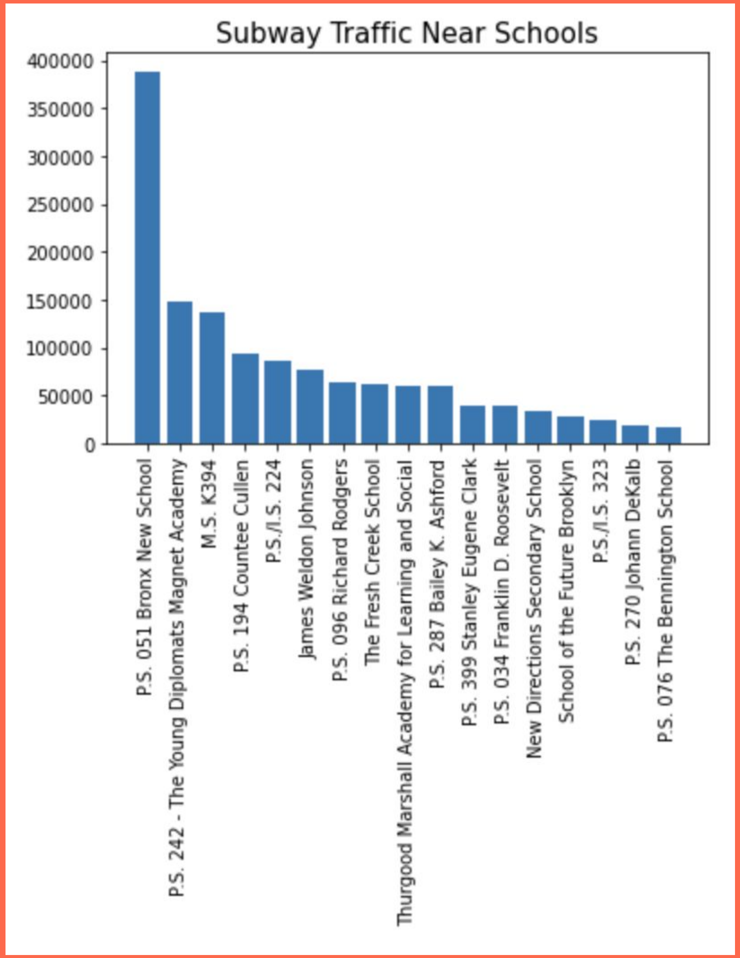
We filtered the data to isolate **weekday afternoons...**





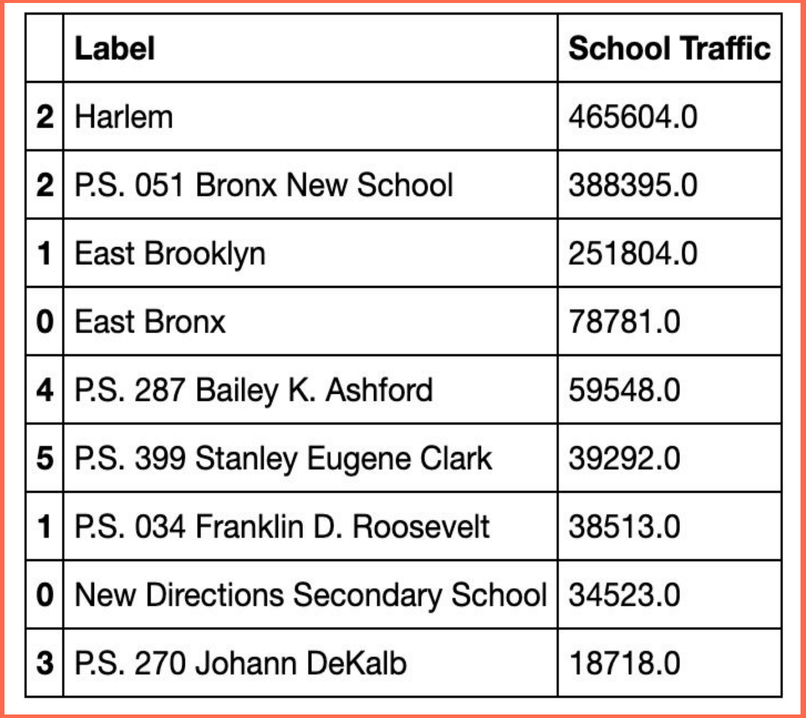


... which reflects students **after-school commute.**

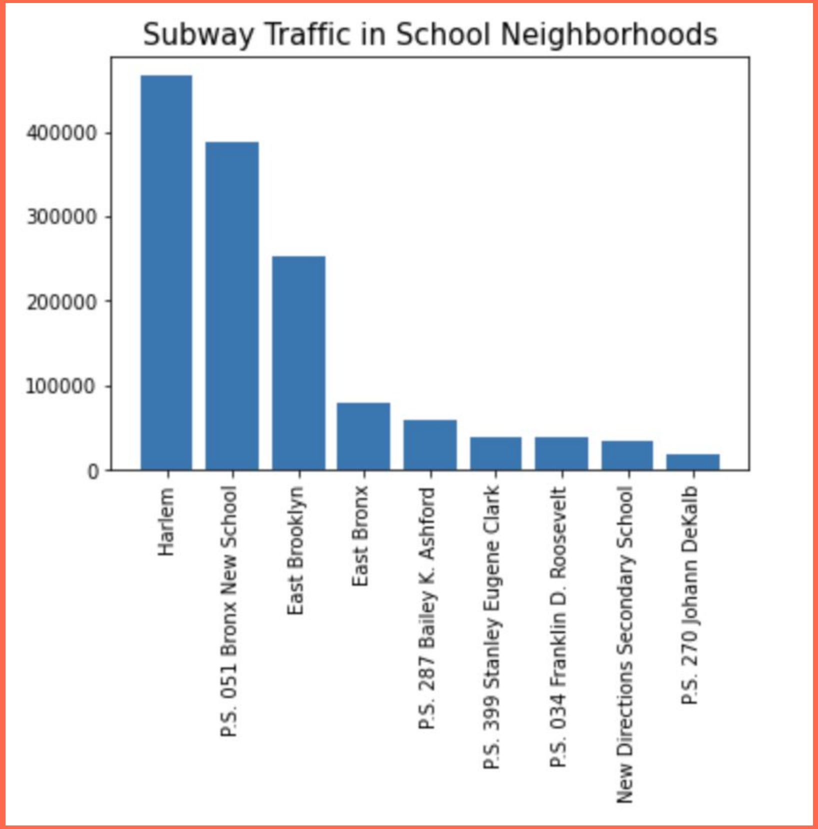


Result: Harlem

has the most subway traffic



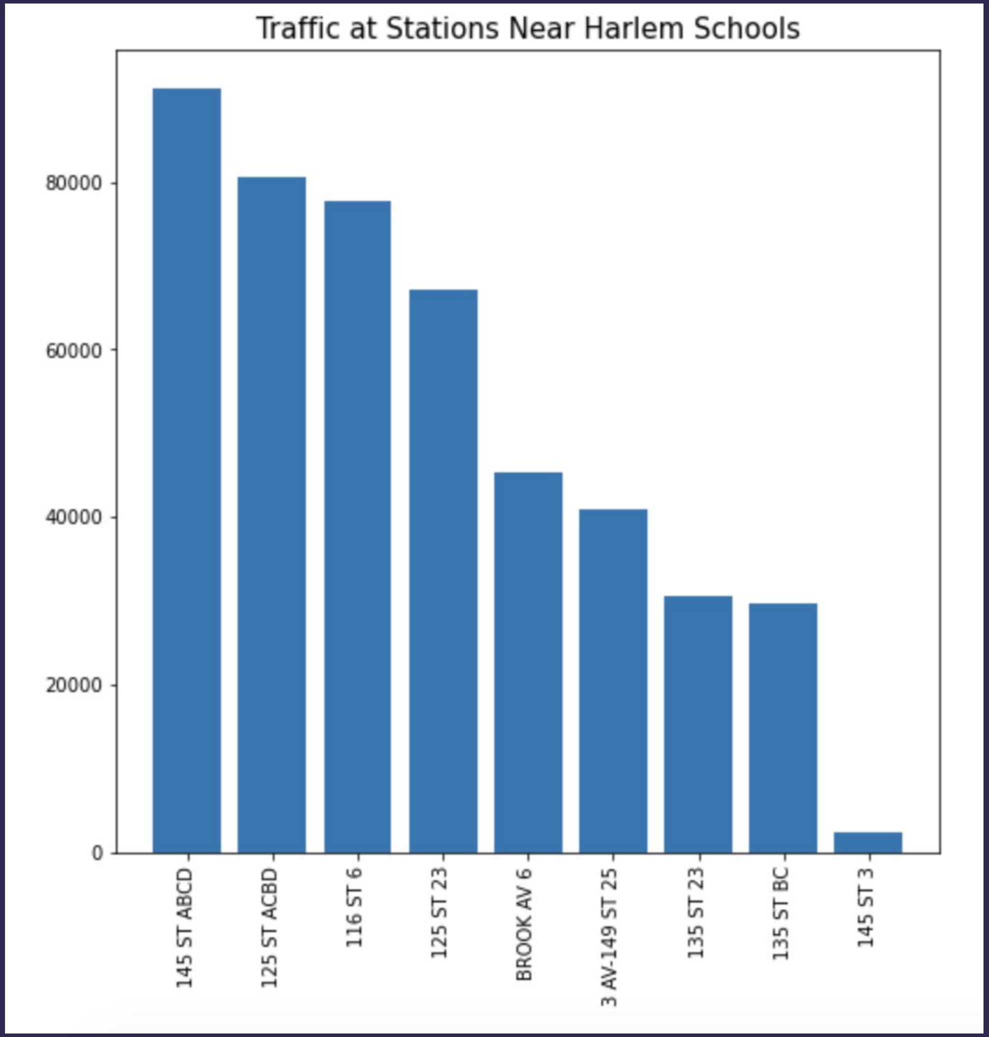
	Label	School Traffic
2	Harlem	465604.0
2	P.S. 051 Bronx New School	388395.0
1	East Brooklyn	251804.0
0	East Bronx	78781.0
4	P.S. 287 Bailey K. Ashford	59548.0
5	P.S. 399 Stanley Eugene Clark	39292.0
1	P.S. 034 Franklin D. Roosevelt	38513.0
0	New Directions Secondary School	34523.0
3	P.S. 270 Johann DeKalb	18718.0

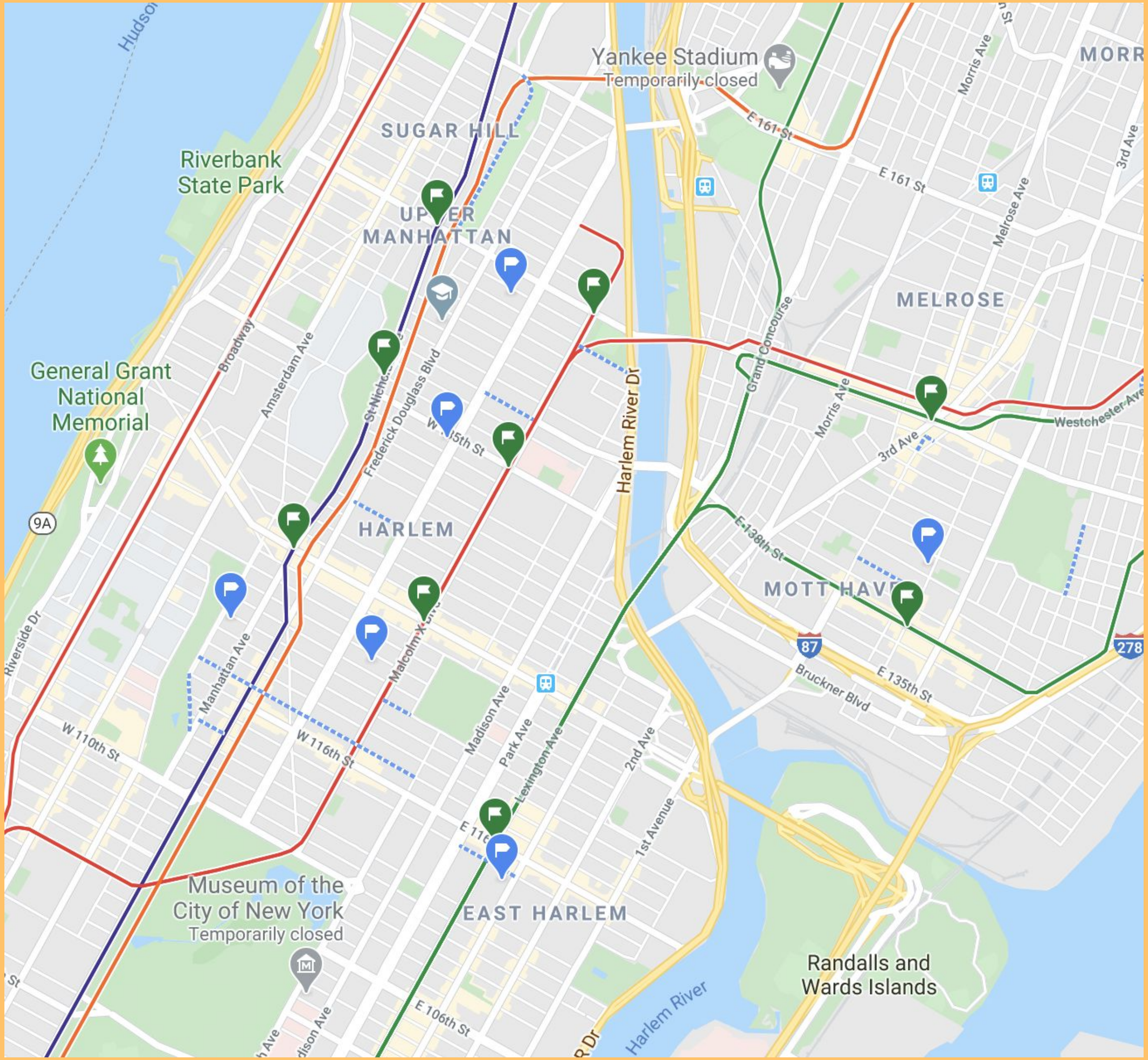


Where is ideal?

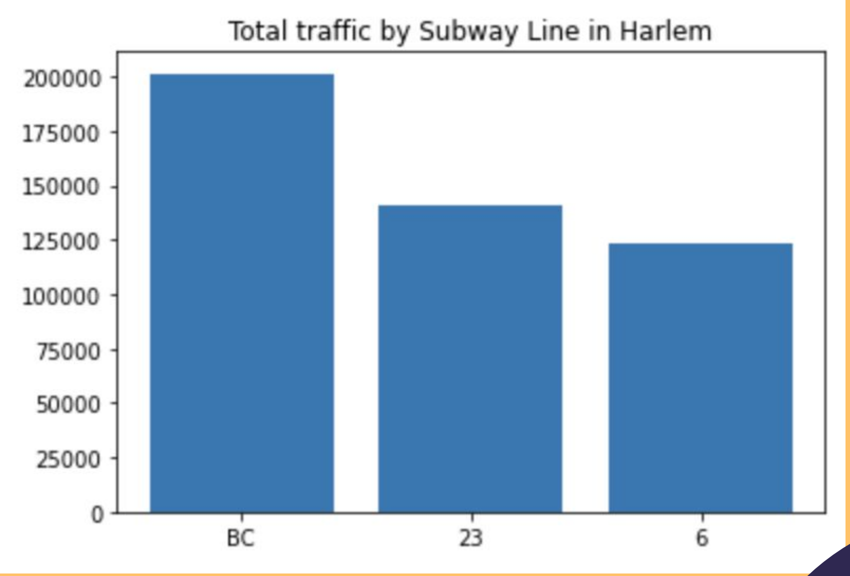
We want a station with high individual traffic...

...that can also reach students from more than just the one nearest school.





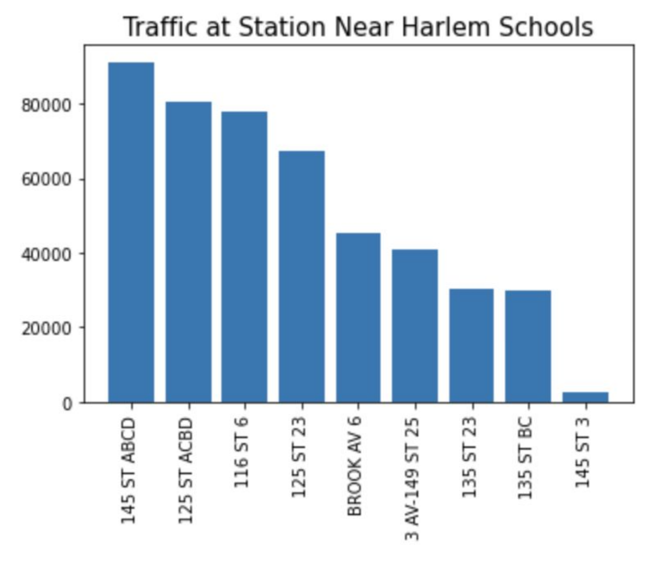
All stations near the schools are servicing *the same 3 subway lines.*


Total traffic by Subway Line in Harlem

Students are getting on the same subways at consecutive stops.

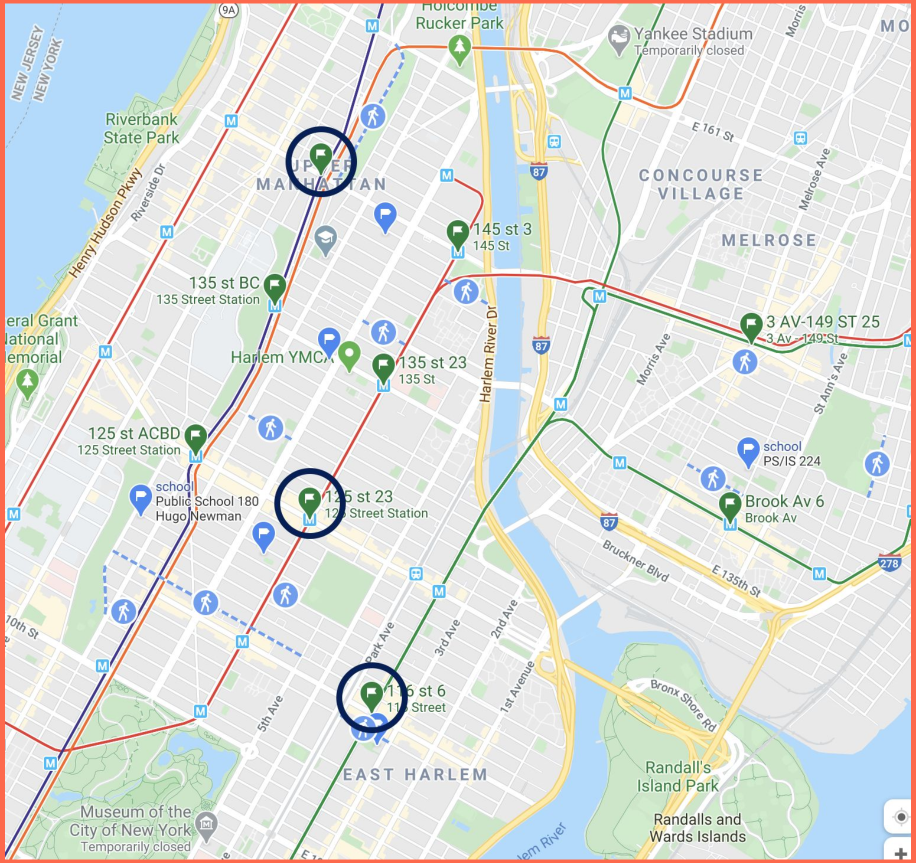
p/10

Ideal event placement:



One event at the busiest station on each line:

- 145th St (ABCD)
- 125 St (23)
- 116 St (6)



Conclusion:

3 events, flyers in the week preceding each, and free pizza

This way, we can draw kids from all stops along each line. The pizza will incentivize them to get off their subway 1 or 2 stops up.

Appendix

future work

01

Future Work Areas of Interest

- Expanding into other cities
- Use geo-spatial data to automate

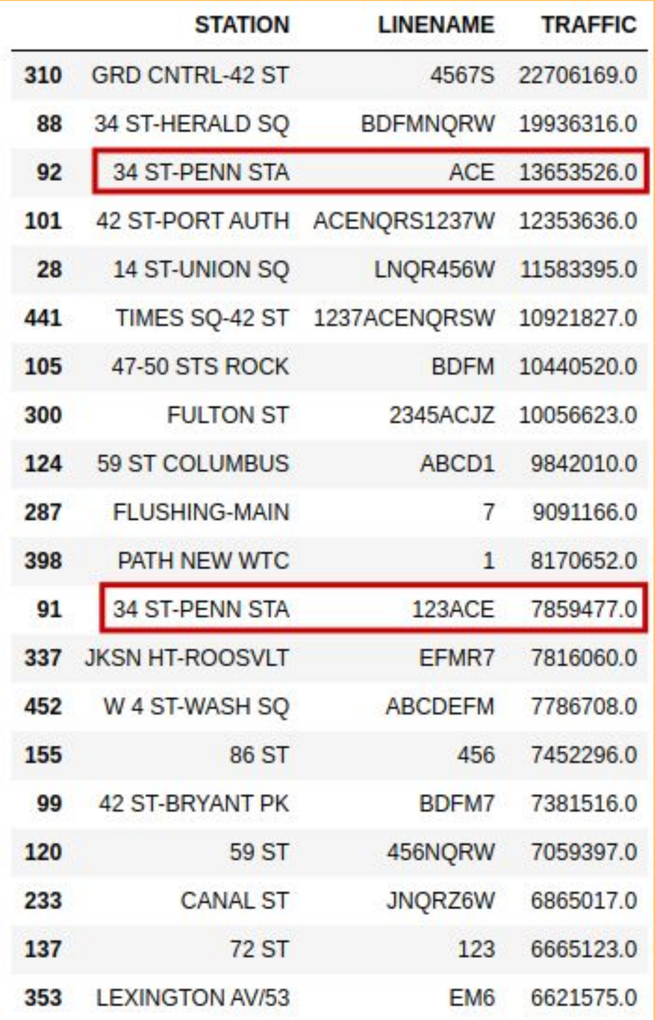
Advanced Methods

For the purposes of this project, we stuck to simple EDA (exploratory data analysis) and data cleaning. We wanted to analyze clusters of schools based on multiple factors like distance from nearby schools/stations and economic neediness of the school vs its proximity to stations, possibly using regression. There were plenty of unexpected quirks with the data and trial/error with pandas syntax., but these methods could potentially be implemented later on.

02

Navigating Errors...

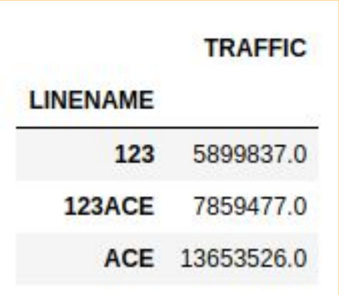
Total Station Ridership by Line



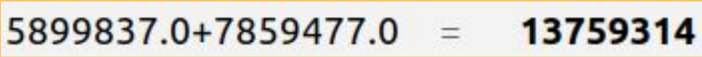
	STATION	LINENAME	TRAFFIC
310	GRD CNTRL-42 ST	4567S	22706169.0
88	34 ST-HERALD SQ	BDFMNQRW	19936316.0
92	34 ST-PENN STA	ACE	13653526.0
101	42 ST-PORT AUTH	ACENQRS1237W	12353636.0
28	14 ST-UNION SQ	LNQR456W	11583395.0
441	TIMES SQ-42 ST	1237ACENQRSW	10921827.0
105	47-50 STS ROCK	BDFM	10440520.0
300	FULTON ST	2345ACJZ	10056623.0
124	59 ST COLUMBUS	ABCD1	9842010.0
287	FLUSHING-MAIN	7	9091166.0
398	PATH NEW WTC	1	8170652.0
91	34 ST-PENN STA	123ACE	7859477.0
337	JKSN HT-ROOSVLT	EFMR7	7816060.0
452	W 4 ST-WASH SQ	ABCDEFM	7786708.0
155	86 ST	456	7452296.0
99	42 ST-BRYANT PK	BDFM7	7381516.0
120	59 ST	456NQRW	7059397.0
233	CANAL ST	JNQRZ6W	6865017.0
137	72 ST	123	6665123.0
353	LEXINGTON AV/53	EM6	6621575.0

Later on during analysis, something strange became evident... Ridership at Penn Station for lines 123 and ACE together was lower than ridership for line ACE alone, by about 2x.

I went back and checked the data, it turns out someone made an error when compiling the data - I'm guessing the entry for Line 123 + ACE was switched with the entry for Line ACE. If we swapped those, the numbers make sense.



	TRAFFIC
LINENAME	
123	5899837.0
123ACE	7859477.0
ACE	13653526.0



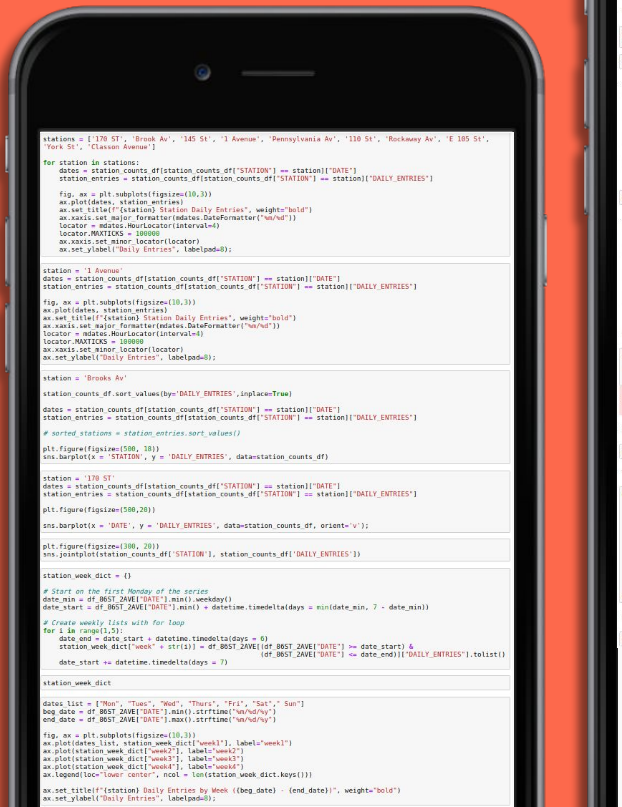
5899837.0+7859477.0 = **13759314**

Earlier Iterations

Early On

These were some of our original forays into the data.

We were only using one week of data, compared with the 3 months of data we used in the final project. So we had the right idea with our analysis, but the incomplete data skewed our results.



```
stations = ['170 ST', 'Brook Av', '145 St', '1 Avenue', 'Pennsylvania Av', '110 St', 'Rockaway Av', 'E 105 St',
'York St', 'Classon Avenue']

for station in stations:
    dates = station_counts_df[station_counts_df["STATION"] == station]["DATE"]
    station_entries = station_counts_df[station_counts_df["STATION"] == station]["DAILY_ENTRIES"]

    fig, ax = plt.subplots(figsize=(10,3))
    ax.plot(dates, station_entries)
    ax.set_title(f"{station} Station Daily Entries", weight="bold")
    ax.xaxis.set_major_formatter(mdates.DateFormatter("%m/%d"))
    locator = mdates.HourLocator(interval=4)
    locator.MAXTICKS = 100000
    ax.xaxis.set_minor_locator(locator)
    ax.set_ylabel("Daily Entries", labelpad=8);
```

```
station = '1 Avenue'
dates = station_counts_df[station_counts_df["STATION"] == station]["DATE"]
station_entries = station_counts_df[station_counts_df["STATION"] == station]["DAILY_ENTRIES"]

fig, ax = plt.subplots(figsize=(10,3))
ax.plot(dates, station_entries)
ax.set_title(f"{station} Station Daily Entries", weight="bold")
ax.xaxis.set_major_formatter(mdates.DateFormatter("%m/%d"))
locator = mdates.HourLocator(interval=4)
locator.MAXTICKS = 100000
ax.xaxis.set_minor_locator(locator)
ax.set_ylabel("Daily Entries", labelpad=8);
```

```
station = 'Brooks Av'

station_counts_df.sort_values(by='DAILY_ENTRIES',inplace=True)

dates = station_counts_df[station_counts_df["STATION"] == station]["DATE"]
station_entries = station_counts_df[station_counts_df["STATION"] == station]["DAILY_ENTRIES"]

# sorted_stations = station_entries.sort_values()

plt.figure(figsize=(500, 18))
sns.barplot(x = 'STATION', y = 'DAILY_ENTRIES', data=station_counts_df)
```

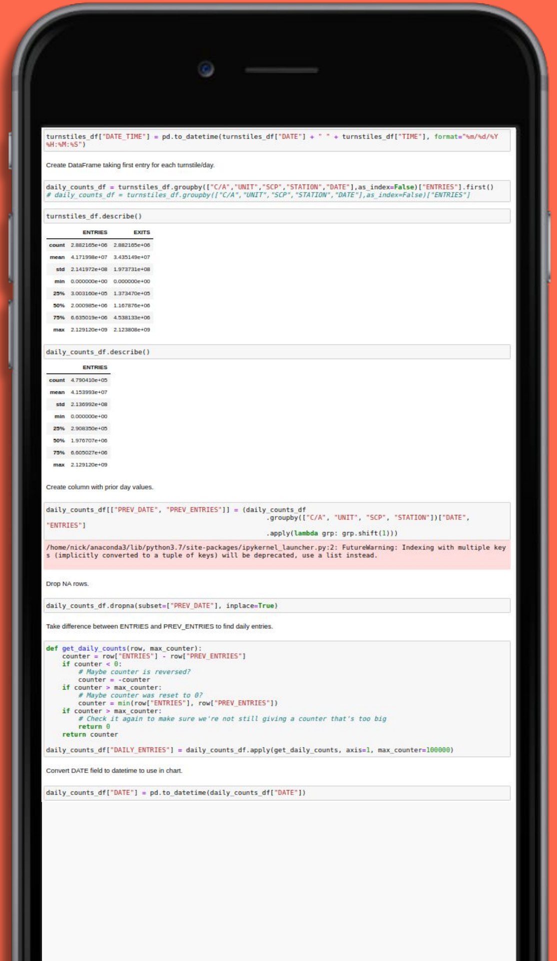
```
station = '170 ST'
dates = station_counts_df[station_counts_df["STATION"] == station]["DATE"]
station_entries = station_counts_df[station_counts_df["STATION"] == station]["DAILY_ENTRIES"]

plt.figure(figsize=(500,20))

sns.barplot(x = 'DATE', y = 'DAILY_ENTRIES', data=station_counts_df, orient='v');
```

```
plt.figure(figsize=(300, 20))
sns.jointplot(station_counts_df['STATION'], station_counts_df['DAILY_ENTRIES'])
```

```
station_week_dict = {}

# Start on the first Monday of the series
date_min = df_86ST_2AVE["DATE"].min().weekday()
date_start = df_86ST_2AVE["DATE"].min() + datetime.timedelta(days = min(date_min, 7 - date_min))

# Create weekly lists with for loop
for i in range(1,5):
    date_end = date_start + datetime.timedelta(days = 6)
    station_week_dict["week" + str(i)] = df_86ST_2AVE[(df_86ST_2AVE["DATE"] >= date_start) &
                                              (df_86ST_2AVE["DATE"] <= date_end)]["DAILY_ENTRIES"].tolist()
    date_start += datetime.timedelta(days = 7)
```

```
station_week_dict
```

```
dates_list = ["Mon", "Tues", "Wed", "Thurs", "Fri", "Sat"," Sun"]
beg_date = df_86ST_2AVE["DATE"].min().strftime("%m/%d/%y")
end_date = df_86ST_2AVE["DATE"].max().strftime("%m/%d/%y")

fig, ax = plt.subplots(figsize=(10,3))
ax.plot(dates_list, station_week_dict["week1"], label="week1")
ax.plot(station_week_dict["week2"], label="week2")
ax.plot(station_week_dict["week3"], label="week3")
ax.plot(station_week_dict["week4"], label="week4")
ax.legend(loc="lower center", ncol = len(station_week_dict.keys()))

ax.set_title(f"{station} Daily Entries by Week ({beg_date} - {end_date})", weight="bold")
ax.set_ylabel("Daily Entries", labelpad=8);
```



```
turnstiles_df["DATE_TIME"] = pd.to_datetime(turnstiles_df["DATE"] + " " + turnstiles_df["TIME"], format="%m/%d/%Y
%H:%M:%S")
```

Create DataFrame taking first entry for each turnstile/day.

```
daily_counts_df = turnstiles_df.groupby(["C/A","UNIT","SCP","STATION","DATE"],as_index=False)["ENTRIES"].first()
# daily_counts_df = turnstiles_df.groupby(["C/A","UNIT","SCP","STATION","DATE"],as_index=False)["ENTRIES"]
```

```
turnstiles_df.describe()
```

	ENTRIES	EXITS
count	2.882165e+06	2.882165e+06
mean	4.171998e+07	3.435149e+07
std	2.141972e+08	1.973731e+08
min	0.000000e+00	0.000000e+00
25%	3.003160e+05	1.373470e+05
50%	2.000985e+06	1.167876e+06
75%	6.635019e+06	4.538133e+06
max	2.129120e+09	2.123808e+09

```
daily_counts_df.describe()
```

	ENTRIES
count	4.790410e+05
mean	4.153993e+07
std	2.136992e+08
min	0.000000e+00
25%	2.908350e+05
50%	1.976707e+06
75%	6.605027e+06
max	2.129120e+09

Create column with prior day values.

```
daily_counts_df[["PREV_DATE", "PREV_ENTRIES"]] = (daily_counts_df
                                                   .groupby(["C/A", "UNIT", "SCP", "STATION"])["DATE",
"ENTRIES"]
                                                   .apply(lambda grp: grp.shift(1)))
```

```
/home/nick/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: FutureWarning: Indexing with multiple key
s (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```

Drop NA rows.

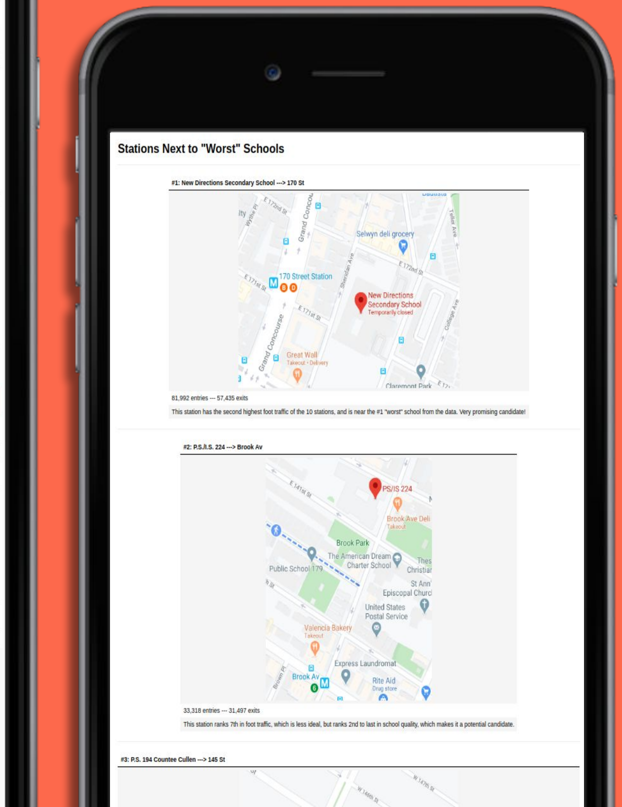
```
daily_counts_df.dropna(subset=["PREV_DATE"], inplace=True)
```

Take difference between ENTRIES and PREV_ENTRIES to find daily entries.

```
def get_daily_counts(row, max_counter):
    counter = row["ENTRIES"] - row["PREV_ENTRIES"]
    if counter < 0:
        # Maybe counter is reversed?
        counter = -counter
    if counter > max_counter:
        # Maybe counter was reset to 0?
        counter = min(row["ENTRIES"], row["PREV_ENTRIES"])
    if counter > max_counter:
        # Check it again to make sure we're not still giving a counter that's too big
        return 0
    return counter

daily_counts_df["DAILY_ENTRIES"] = daily_counts_df.apply(get_daily_counts, axis=1, max_counter=100000)
```

Convert DATE field to datetime to use in chart.

```
daily_counts_df["DATE"] = pd.to_datetime(daily_counts_df["DATE"])
```



Stations Next to "Worst" Schools

#1: New Directions Secondary School ---> 170 St

81,992 entries --- 57,435 exits

This station has the second highest foot traffic of the 10 stations, and is near the #1 "worst" school from the data. Very promising candidate!

#2: P.S./I.S. 224 ---> Brook Av

33,318 entries --- 31,497 exits

This station ranks 7th in foot traffic, which is less ideal, but ranks 2nd to last in school quality, which makes it a potential candidate.

#3: P.S. 194 Countee Cullen ---> 145 St

Ak Sp Nh :)

References

- https://infohub.nyced.org/reports/school-quality/school-quality-reports-and-resources
- https://infohub.nyced.org/docs/default-source/default-document-library/201819_ems_sqr_results.xlsx
- http://web.mta.info/developers/turnstile.html
- https://towardsdatascience.com/mta-turstile-data-my-first-taste-of-a-data-science-project-493b03f1708a