MTA Exploratory Data Analysis

Randy Grant (DSML)

Introduction

- Issue: As New York City Transit employees, it’s hard to predict when an extra train is needed for a station.

- Solution: With existing turnstile data, a data scientist can identify signs of that need prior to it becoming a problem.

- Goal: Demonstrate this concept to prove my value as a future employee of the New York City Transit.

Methods

<u>Data Used</u>

- MTA Turnstile Data - drilled down to:
  - Date
  - Station
  - Entries
- Tools Used:
  - VSCode -> Jupyter Notebook -> Python
  - Pandas
  - Scikit-learn
  - Seaborn
  - Plotly
- Metrics Created:
  - Spread of Data Across the Top 5 Stations
  - Top 5 Stations with its Daily Entries from 2020 to 2021
  - Total Daily Entries of the Top 5 Stations - Binned Monthly
  - Volume of Top 5 Stations with Linear Trend Line

Sample Data

Raw:

	C/A	UNIT	SCP	STATION	LINENAME	DIVISION	DATE	TIME	DESC	ENTRIES	EXITS
13257073	N003	R185	00-00-01	DYCKMAN ST	A	IND	12/01/2020	04:00:00	REGULAR	13458972	290097
23068229	N325A	R218	00-00-00	ELMHURST AV	MR	IND	01/06/2020	03:00:00	REGULAR	1016376	395958
14369626	N528	R257	01-06-01	EAST BROADWAY	F	IND	10/26/2020	08:00:00	REGULAR	3601084	2822378
10724979	H003	R163	01-00-02	6 AV	FLM123	BMT	02/24/2021	07:00:00	REGULAR	1208998199	1200551381
9140719	PTH05	R543	00-00-06	EXCHANGE PLACE	1	PTH	04/21/2021	11:15:39	REGULAR	198043	838570

Final:

	date	station	year	month	day	daily_entries	ordinal
2152	2021-11-21	fulton st	2021	11	21	12945.0	738115
1645	2020-07-02	fulton st	2020	07	02	14336.0	737608
1202	2021-04-16	34 st-herald sq	2021	04	16	23516.0	737896
2017	2021-07-09	fulton st	2021	07	09	24604.0	737980
2722	2021-06-13	grd cntrl-42 st	2021	06	13	12052.0	737954

Tool Details

My IDE that contains all of the rest of the awesomeness

pandas

I used this to bring in the data, filter it down, group it, count it, all to make it pretty

seaborn

This was used to graph out all of the daily entries for a 2 year period

plotly | Graphing Libraries

Used for the interactive graphs

scikit learn

Linear model was created and graphed out using this amazing library

Snapshot of The Process

Get the Data from a Local Database File

1 cell hidden...

```python
# open a connection to the local mta_data.db file
con = sqlite3.connect(db_path / "mta_data.db")

# make dataframe from the database query
# tried parse_dates option but did not work due to appears it's reading as object and not inferring datetime
#df = pd.read_sql("select * from mta_data", con, parse_dates=[['DATE','TIME']])
df = pd.read_sql("select * from mta_data", con)
```

Start Exploring the Data

4 cells hidden...

```python
df.shape
```

✓ 0.8s

```
(23397599, 11)
```

```python
df.columns
```

✓ 0.8s

```
Index(['C/A', 'UNIT', 'SCP', 'STATION', 'LINENAME', 'DIVISION', 'DATE', 'TIME',
       'DESC', 'ENTRIES', 'EXITS'],
      dtype='object')
```

Now that there's more info about what I have, I will start altering the data to make it more managable

2 cells hidden...

```python
# lowercase all
# ref: https://stackoverflow.com/a/50084009
df.rename(columns=str.lower, inplace=True)
df = df.applymap(lambda x: x.lower() if type(x) == str else x)

# make a timestamp column based on combined columns of DATE and TIME
df['timestamp'] = pd.to_datetime(df['date'] + df['time'], format='%m/%d/%Y%H:%M:%S')

# make the timestamp split out better
df['year'] = df['timestamp'].dt.year
df['month'] = df['timestamp'].dt.month.map("{:02}".format)
df['day'] = df['timestamp'].dt.day.map("{:02}".format)
df['hour'] = df['timestamp'].dt.hour.map("{:02}".format)
df['minute'] = df['timestamp'].dt.minute.map("{:02}".format)
df['second'] = df['timestamp'].dt.second.map("{:02}".format)

# sort and remove duplicates
df.sort_values(["c/a", "unit", "scp", "station", "timestamp"], inplace=True, ascending=False)
df.drop_duplicates(subset=["c/a", "unit", "scp", "station", "timestamp"], inplace=True)

# drop unused columns
df = df.drop(["exits", "desc", "date", "time"], axis=1, errors="ignore")

# check the data to ensure changes
print(df.columns)
print(df.shape)
```

```python
# most of this function was the instructor's as per "mta-pair-solution.ipynb" - modified a bit
def get_daily_counts(row, max_counter):
    counter = abs(row["daily_entries"])
    if counter > max_counter:
        while counter > max_counter:
            counter = counter / 3
        return counter
    return counter

# setting the max_counter to the median ridership per day
max_counter = int(np.max(rider_summary[['Average Weekday', 'Average Weekend']].sum(axis=1)))
ts_daily["daily_entries"] = ts_daily.apply(get_daily_counts, axis=1, max_counter=max_counter)
ts_daily.head()
```

Find the Top 5 Stations

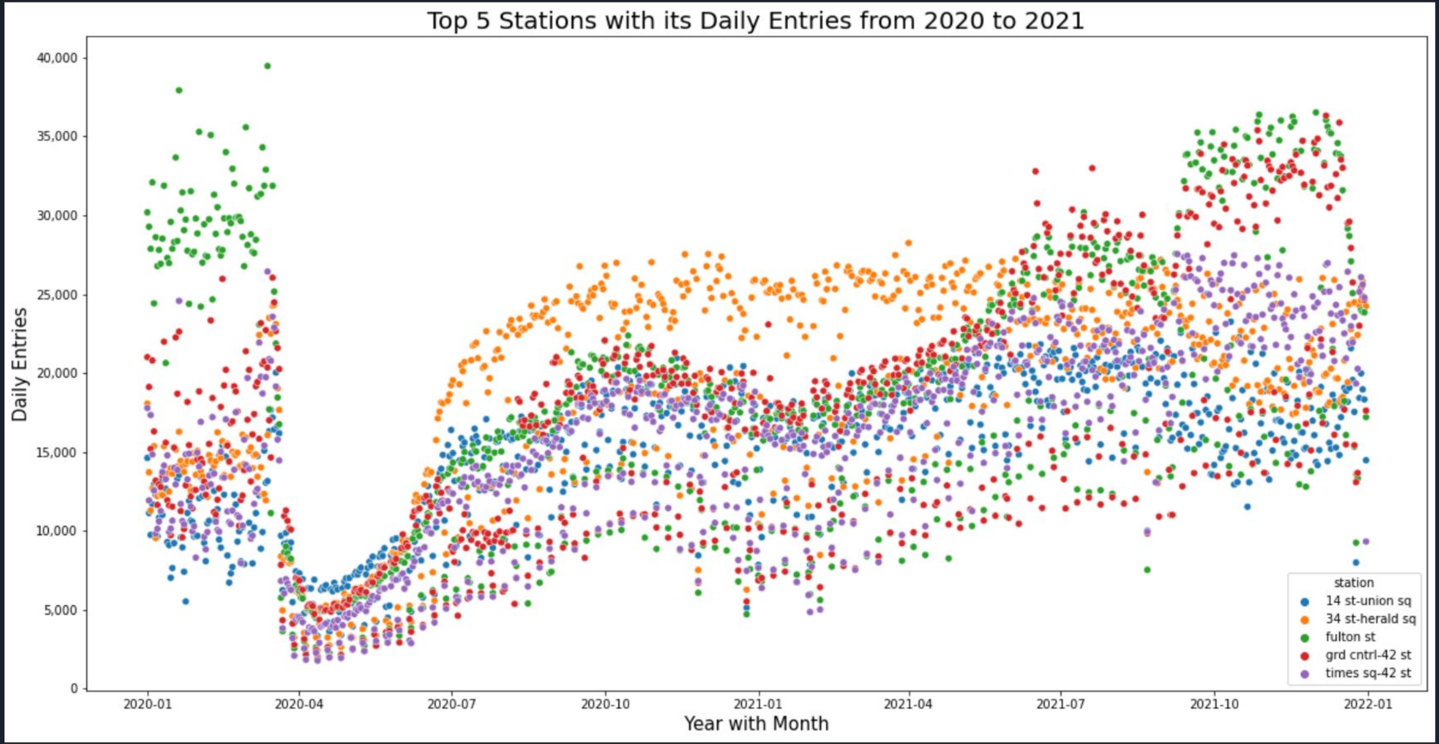
16 cells hidden...

Attempting a Linear Regression Trend Line

```python
# determine the lower and upper bounds in iqr to deal with outliers
# ref: https://androidkt.com/detect-and-remove-outliers-from-pandas-dataframe/

q1=ts_daily['daily_entries'].quantile(0.25)
q3=ts_daily['daily_entries'].quantile(0.75)
iqr=q3-q1
lower_bound=q1 - 1.5 * iqr
upper_bound=q3 + 1.5 * iqr

ts_daily = ts_daily[~((ts_daily['daily_entries'] < lower_bound) | (ts_daily['daily_entries'] > upper_bound))].sort_values(by='daily_entries', ascending=False)
```

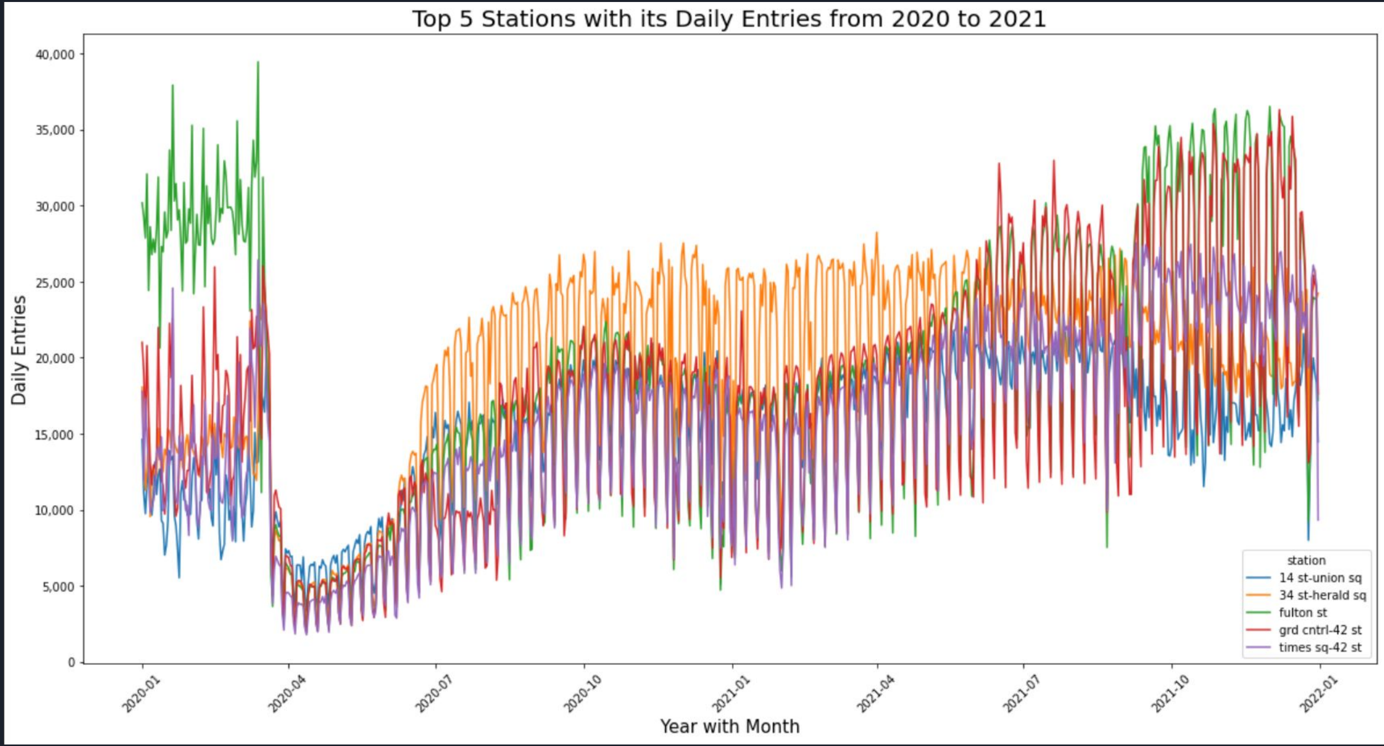
```python
stations_daily.set_index('date', inplace=True)
data = stations_daily

# add an ordinal column because sklearn doesn't work with datetimes
data['ordinal'] = data.index.map(pd.Timestamp.toordinal)

# create the model
model = LinearRegression()

# extract x and y from dataframe data
x = data[['ordinal']]
y = data[['daily_entries']]

# fit the mode
model.fit(x, y)

# print the slope and intercept if desired
print('intercept:', model.intercept_[0])
print('slope:', model.coef_[0][0])

# select x1 and x2 and get the corresponding date from the index
x1 = data.ordinal.min()
x1_date = data[data.ordinal.eq(x1)].index[0]
x2 = data.ordinal.max()
x2_date = data[data.ordinal.eq(x2)].index[0]

# calculate y1, given x1
y1 = model.predict(np.array([[x1]]))[0][0]
print('y1:', y1)

# calculate y2, given x2
y2 = model.predict(np.array([[x2]]))[0][0]
print('y2:', y2)

# plot it
ax1 = data.plot(y='daily_entries', c='black', figsize=(20, 10), grid=True, legend=False, label="Volume of Daily Entries", title="Volume of Top 5 Stations with Linear Trend Line")
locs, labels = plt.xticks()
plt.ticklabel_format(style='plain', axis='y')
plt.setp(labels, rotation=45)
ax1.set_yticklabels(['{:,}'.format(int(x)) for x in ax.get_yticks().tolist()])
ax1.plot([x1_date, x2_date], [y1, y2], label='Trend Line', c='red')
ax1.legend(fontsize=15)
ax1.set_xlabel("Year and Month", fontsize = 15)
ax1.set_ylabel("Daily Entries", fontsize = 15);

stations_daily.reset_index(inplace=True)
```

Metrics!!!

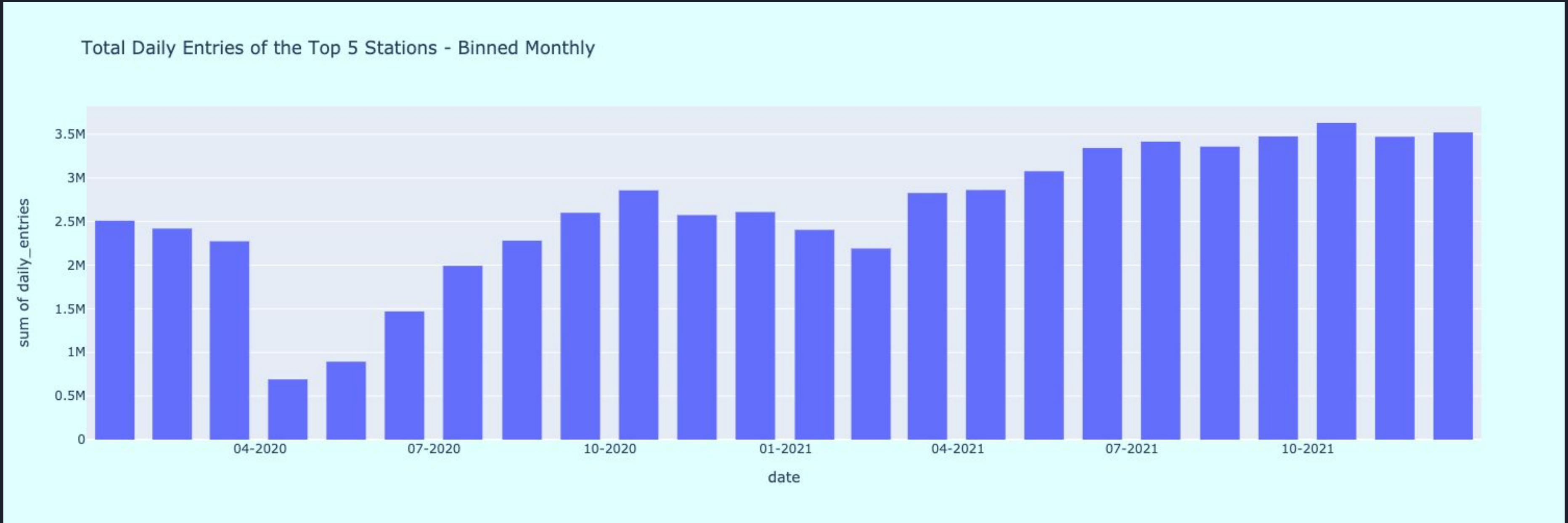


Top 5 Stations with its Daily Entries from 2020 to 2021
Daily Entries
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
2021-10
2022-01
Year with Month
station
14 st-union sq
34 st-herald sq
fulton st
grd cntrl-42 st
times sq-42 st

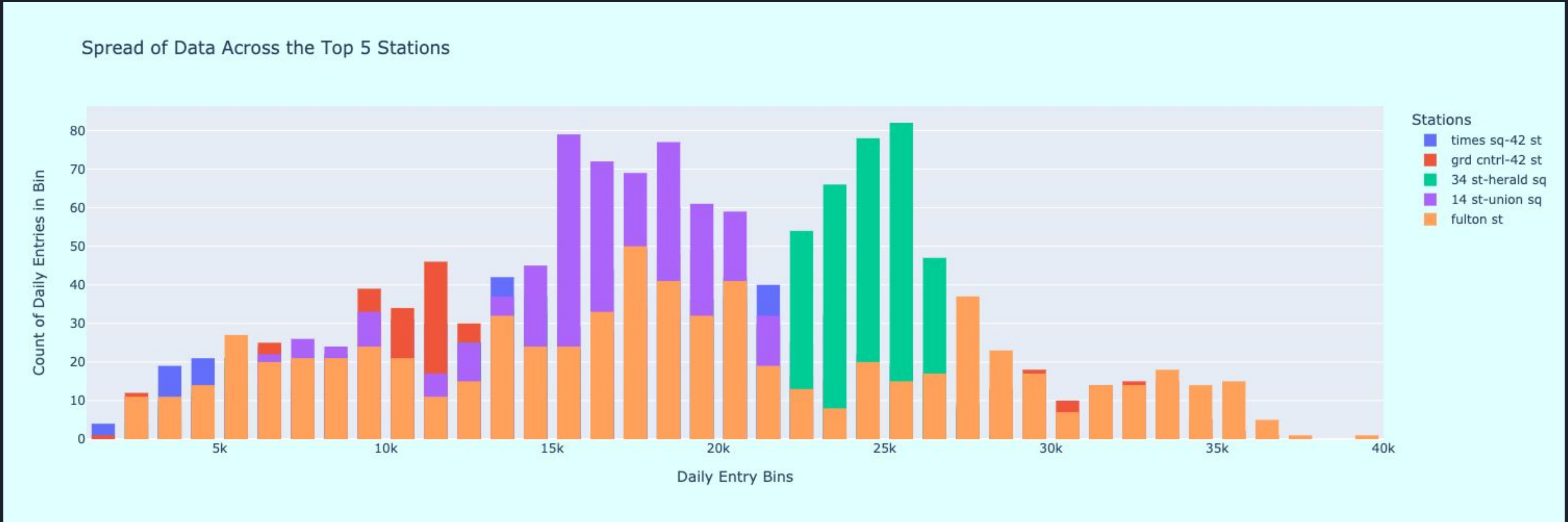
Metrics again!!!



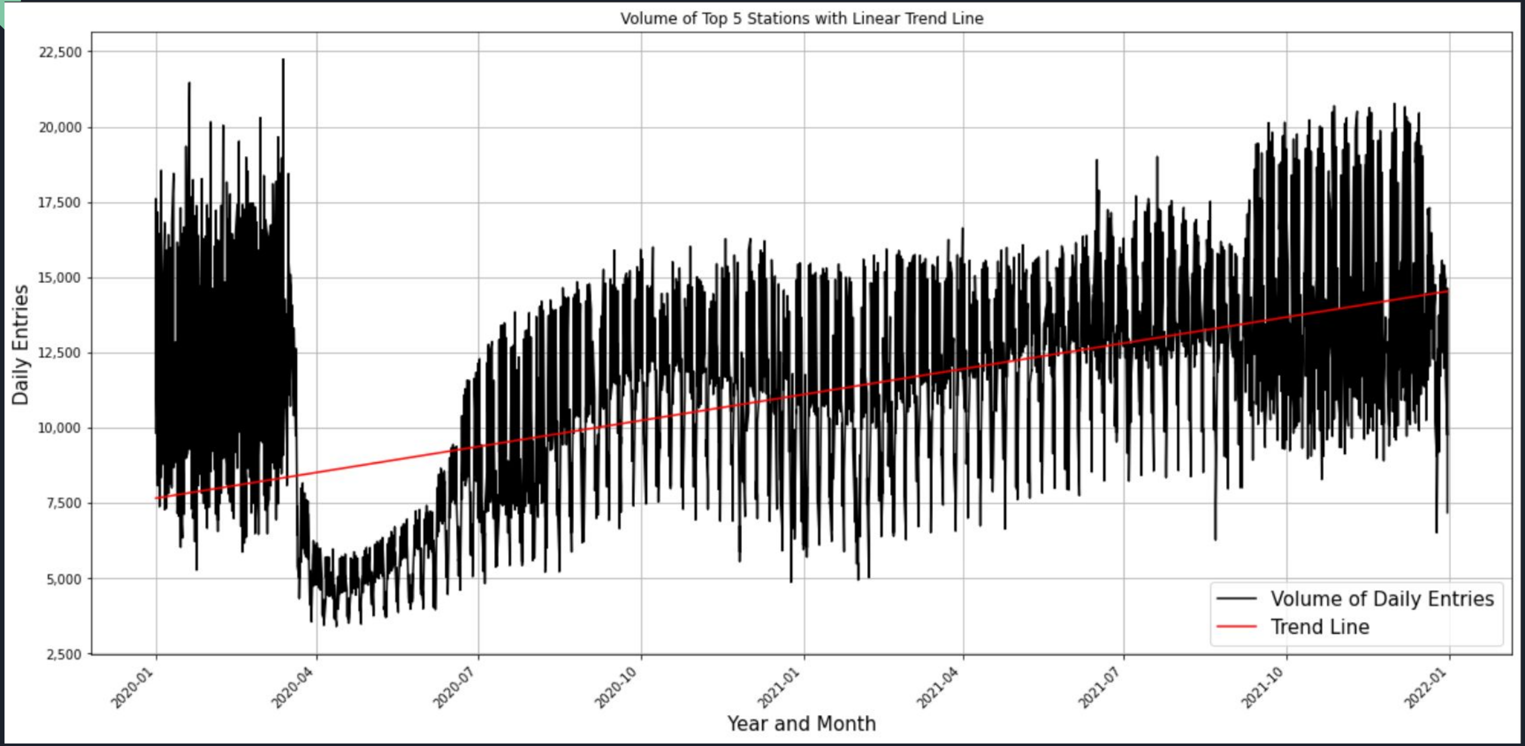
Metrics continued!!!!



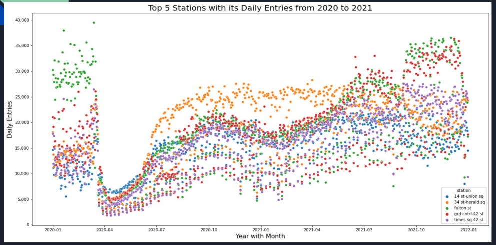
Metrics continued again!!!!

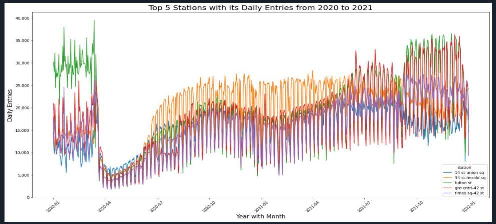


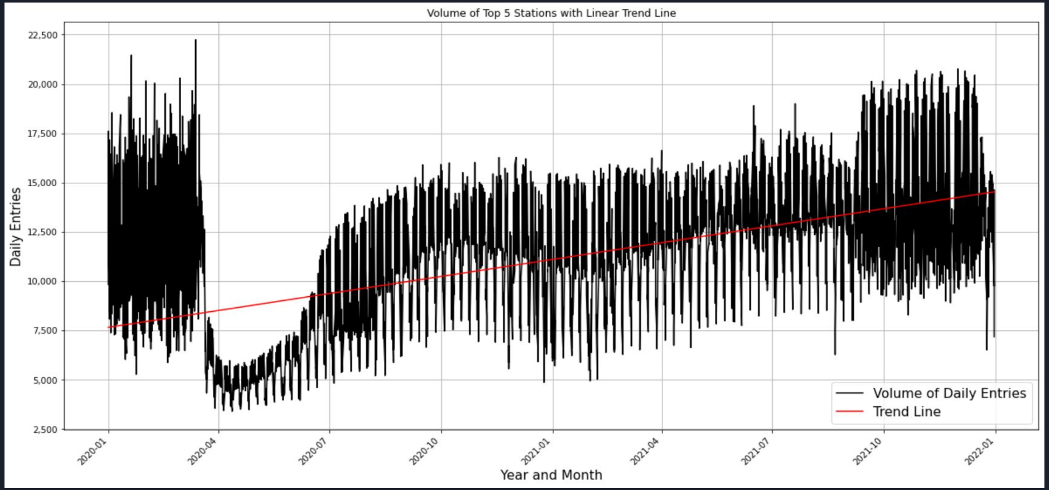
Metrics continued yet again!!!!



Findings and Recommendations







All charts indicated a big dip at the end of March 2020. This coincides with the height of COVID-19. At about May 2020, the daily entries starts going up again.

Even though the dip appears to have affected the trend line, there’s still evidence that more trains will be needed for the top 5 stations. This is due to the increased entry rate. Please plan accordingly as there will be a need soon.

Appendix

http://web.mta.info/developers/turnstile.html

http://web.mta.info/developers/resources/nyct/turnstile/ts_Field_Description.txt

https://new.mta.info/agency/new-york-city-transit/subway-bus-ridership-2020

https://stackoverflow.com/a/63755285

https://www.delftstack.com/howto/seaborn/rotate-tick-labels-seaborn/

https://stackoverflow.com/questions/25973581/how-do-i-format-axis-number-format-to-thousands-with-a-comma-in-matplotlib

Appendix continued

https://stackoverflow.com/a/50084009

https://androidkt.com/detect-and-remove-outliers-from-pandas-dataframe/

https://plotly.com/python/time-series/#summarizing-timeseries-data-with-histograms

https://plotly.com/python/histograms/

https://stackoverflow.com/a/69177333

https://en.wikipedia.org/wiki/COVID-19_pandemic_in_New_York_City

Metis solution notebooks "mta-pair-[1-3]-solution.ipynb"

get_mta.py

Thank you!

Questions?