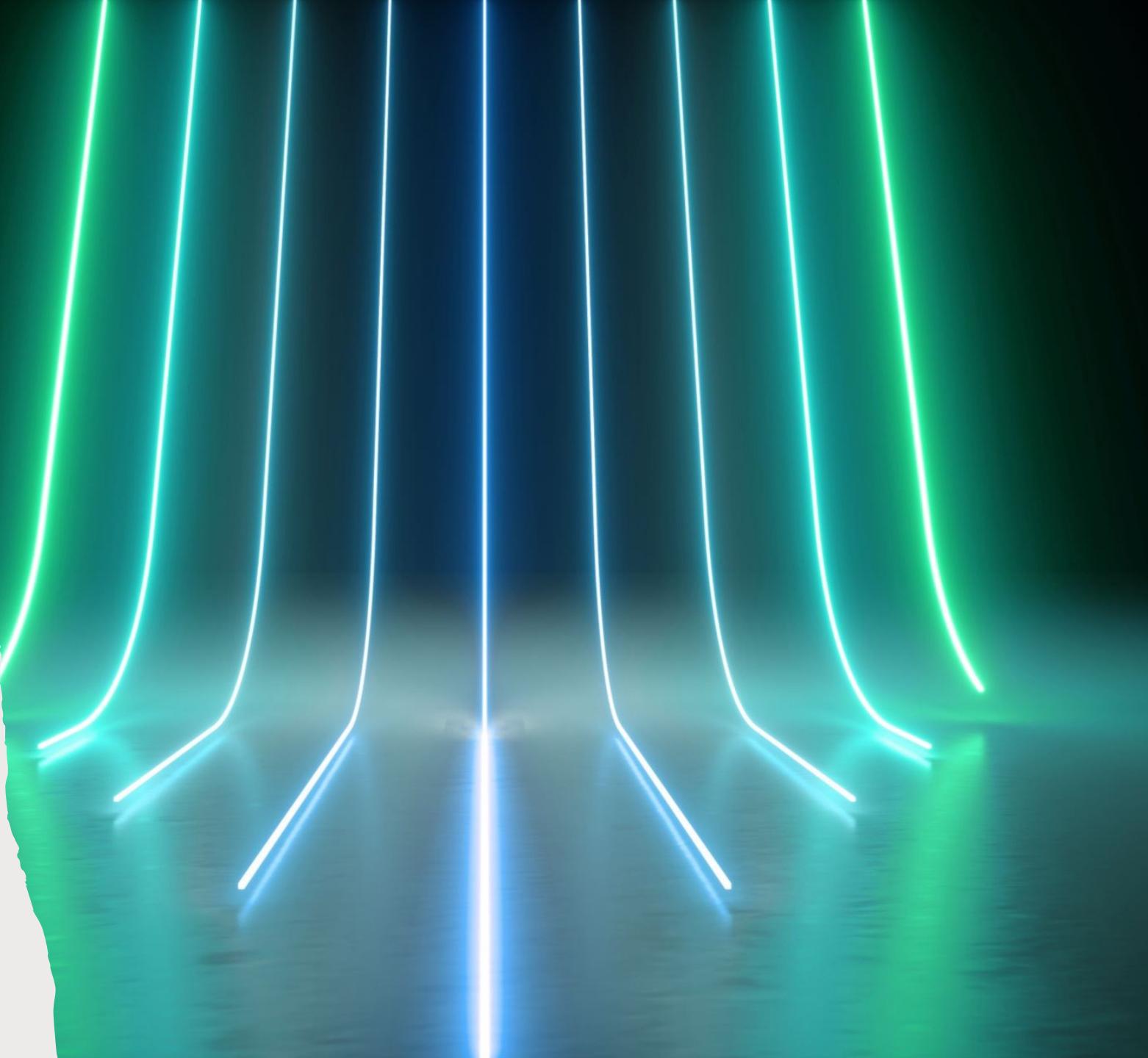


# AD SPACE PURCHASE STRATEGY

Timothy Tsung





# INTRODUCTION

- **Objective**

To identify top 10 NYC MTA stations with the highest volume of foot traffic

- **Who?**

New York Marketing Firm for their client Feel

- **Why?**

For ad space purchase to boost sales

# METHODOLOGY

## Assumptions

- Highest foot traffic takes place during rush hour
- Recent data is representative of population behavior
- COVID's impact is here to last

## Data

- Examined 6 months of MTA turnstile data from **7/31/21-1/29/22**
- Focused on 4-hour time blocks during the rush hours periods (6am – 10am & 3pm – 7pm)
- Analyzed AVG station activity, entries, and exits for the specified time blocks

## METHODOLOGY

Cleaned 5,458,607 rows of data into

- 6am – 10am total data points: 378
- 3pm – 7pm total data points: 379

Used Python Pandas for data manipulation and cleaning

- `.indexer_between_time()`
- `.groupby()`
- Masking
- Basic arithmetic functions i.e. `.sum()` & `.mean()`

Used Matplotlib for plotting top 15 stations with highest foot traffic during rush hour periods based on **activity**, **entries**, and **exits**

## E X A M P L E 1

```
index = pd.DatetimeIndex(MTAClean['Date_Time'])  
target_time_morning = MTAClean.iloc[index.indexer_between_time('06:00:00','10:00:00')]
```



Convert date and time to  
DateTimeIndex format



Filter for the time period I want using  
.indexer\_between\_time( ) function

## EXAMPLE 2

```
per_scp_morn = activity_morn.reset_index().groupby(['STATION','SCP'])\n                [['Daily_Entries','Daily_Exits']].sum()\nper_scp_morn['Tot_Activity_Morn'] = per_scp_morn['Tot_Entries_Morn']+\\ \n                                per_scp_morn['Tot_Exits_Morn']\ntot_mean_morn = per_scp_morn.reset_index().groupby(['STATION'])\n[['Tot_Entries_Morn','Tot_Exits_Morn','Tot_Activity_Morn']].mean()
```

- Grouping of data using the .groupby( ) function
- Create new column
- Calculating AVG activity level using python .mean( ) function

# RESULTS

## AVG daily activity

- St. George, 103 St, 125 St, 116 St, York St, 103 St-Corona, 167 St, 7<sup>th</sup> Ave, Journal Square, 145 St

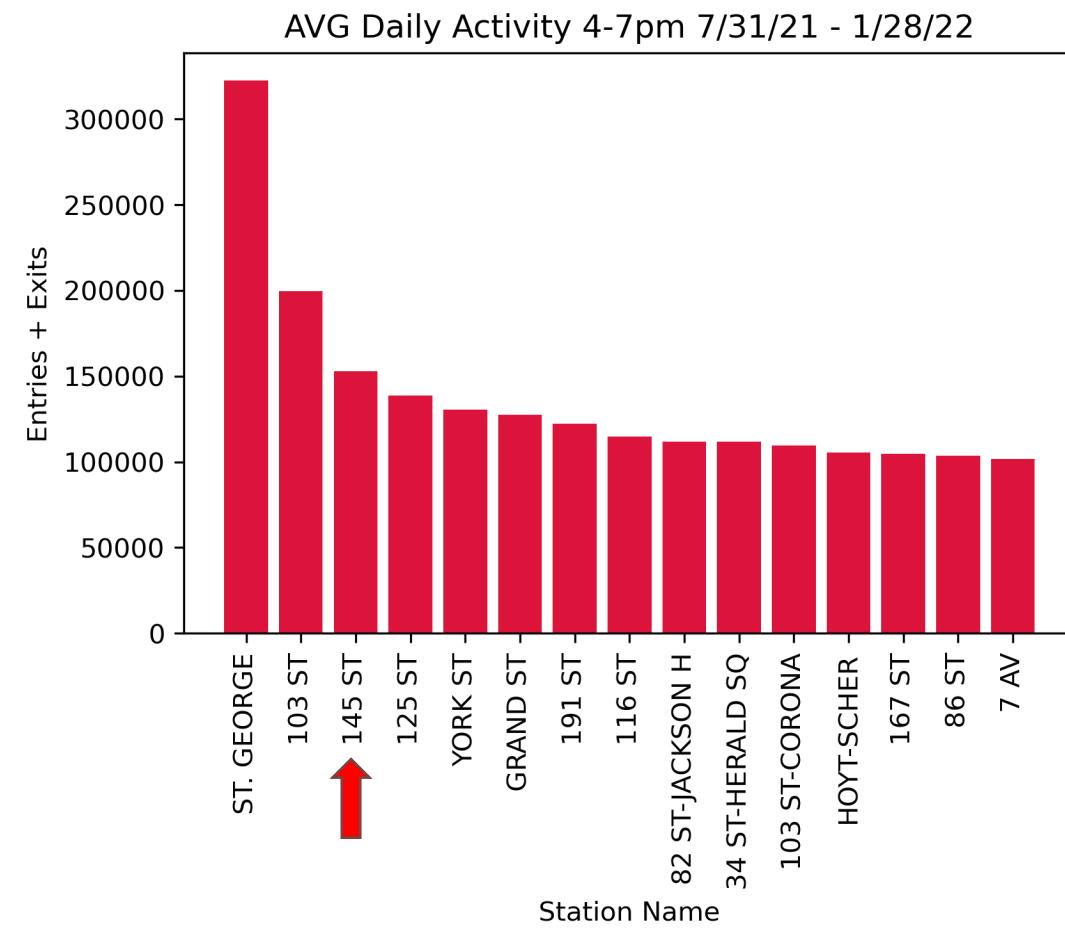
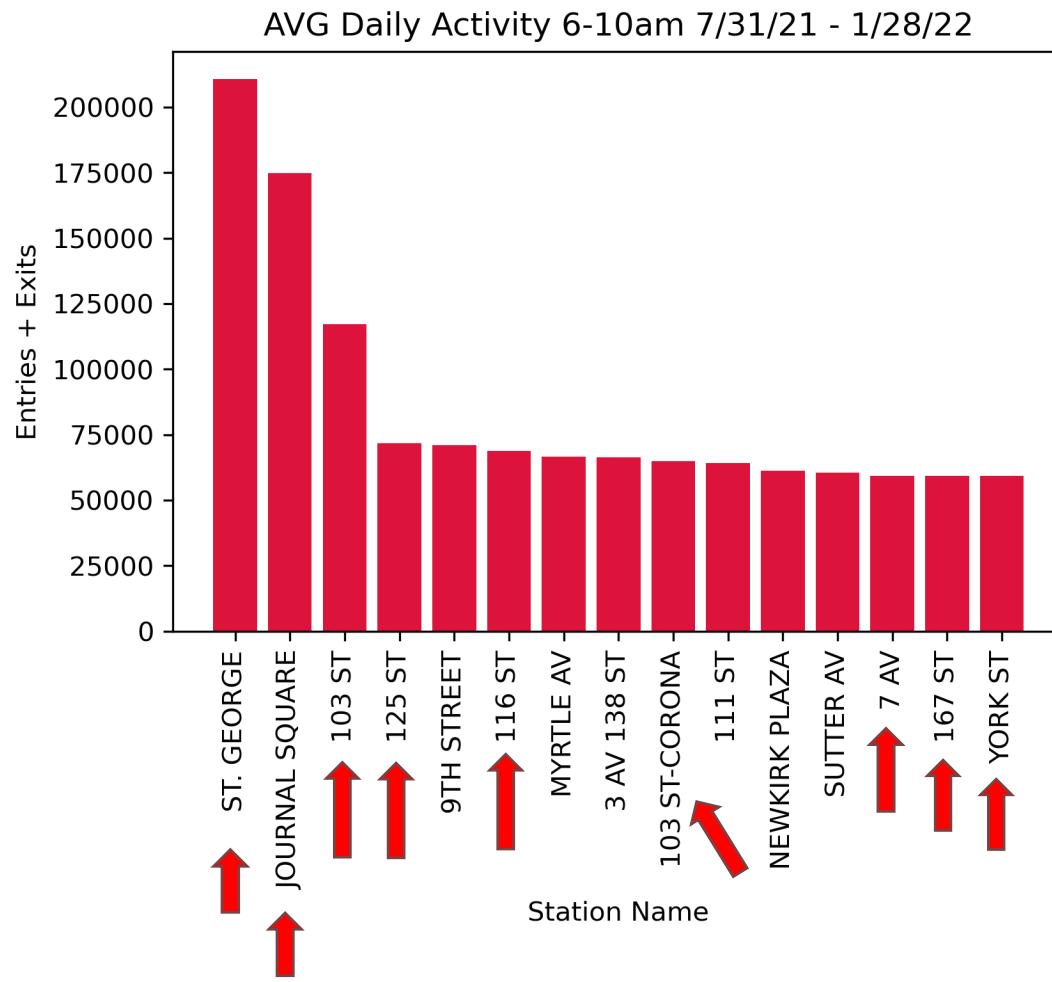
## AVG daily entries

- St. George, 103 St, Journal Square, 145 St, Grand St, York St, 125 St, 116 St, Cathedral Pkwy, 111 St

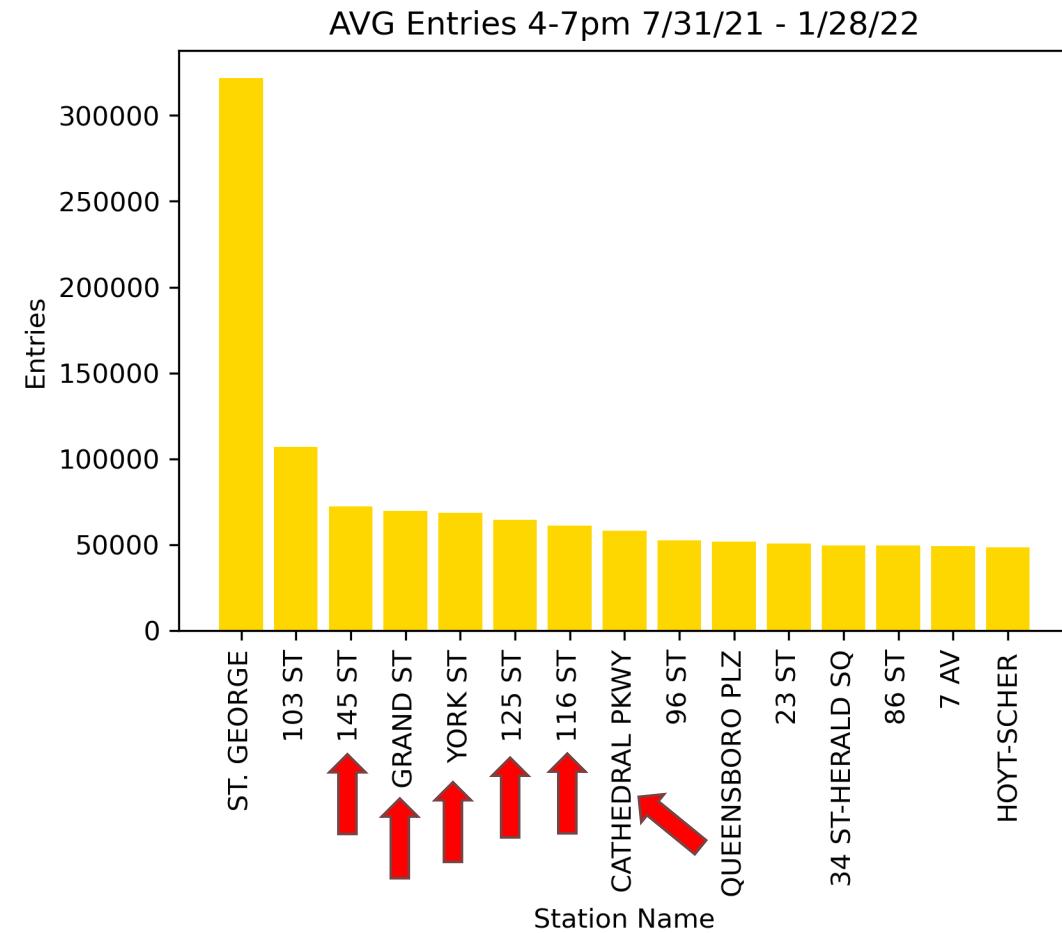
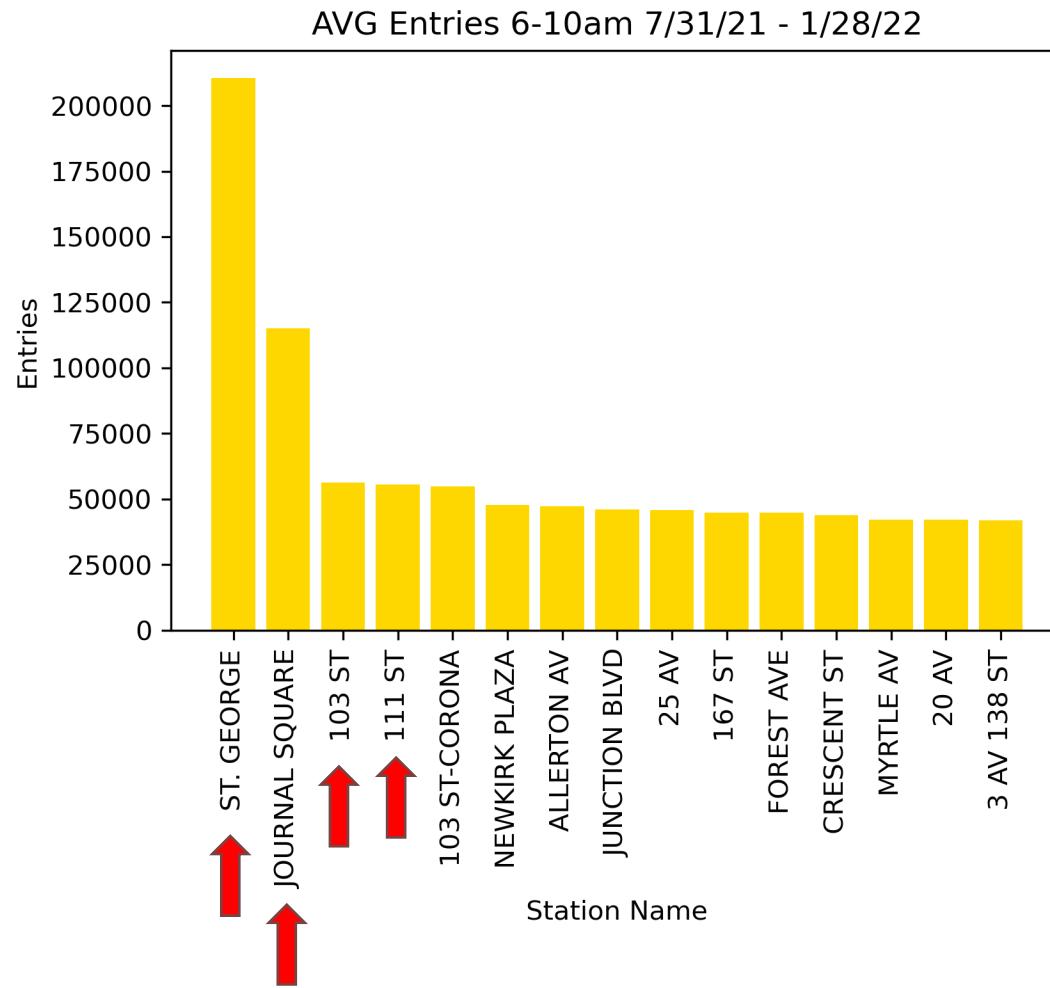
## AVG daily exits

- 103 St, 125 St, York St, 145 St, 191 St, 82 St-Jackson H, 103 St-Corona, 34 St-Penn Station, 34 St-Herald Sq, Junction Blvd

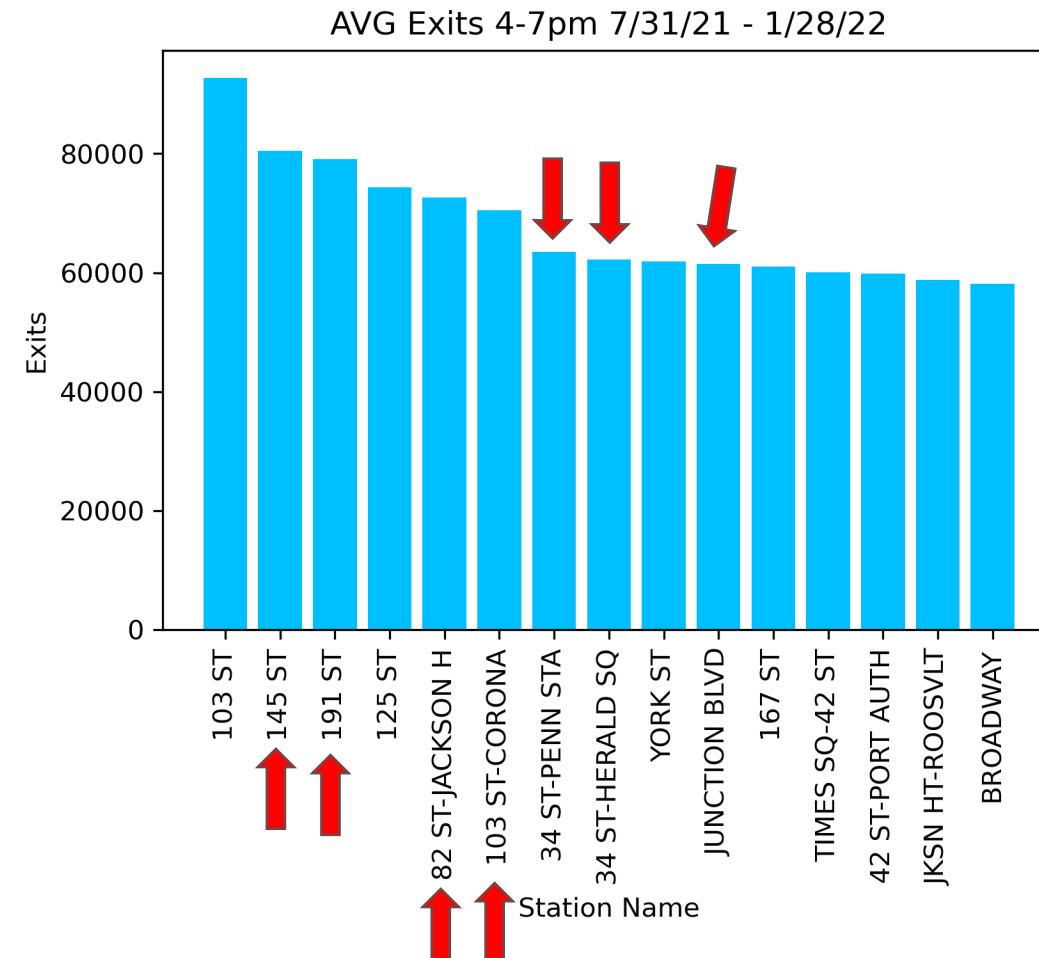
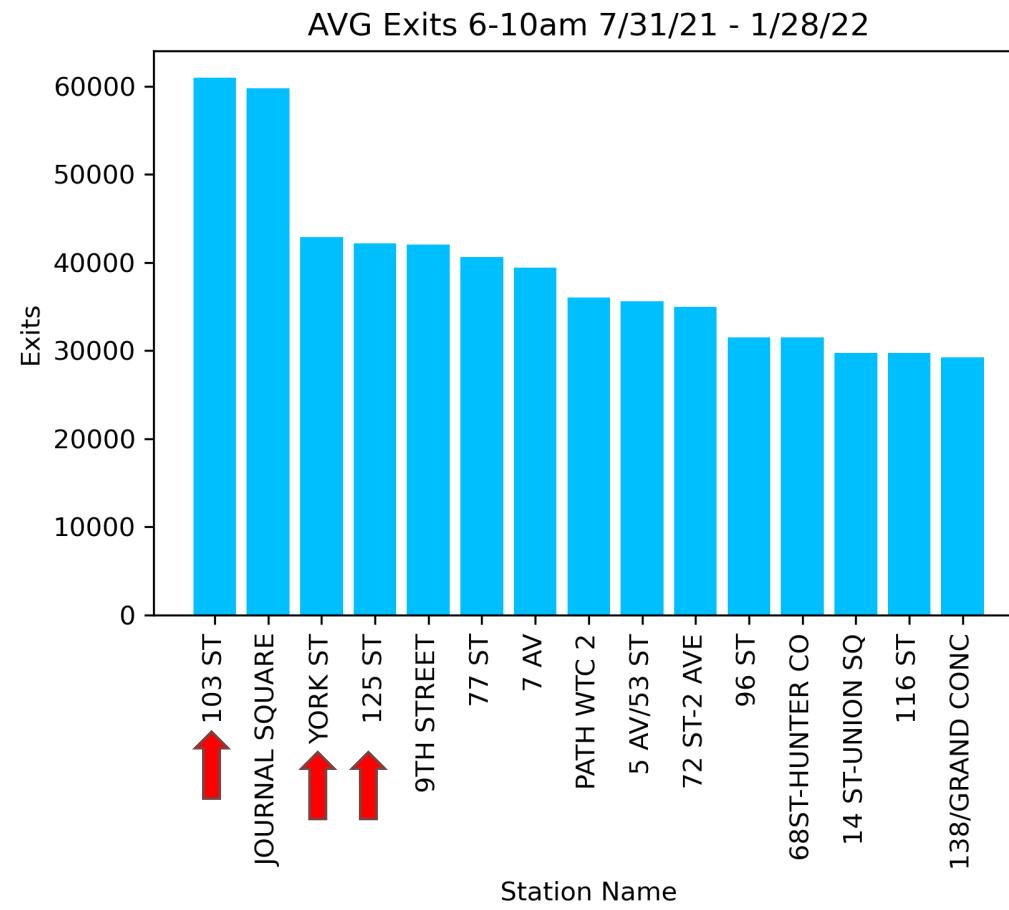
# DAILY ACTIVITY



# DAILY ENTRIES

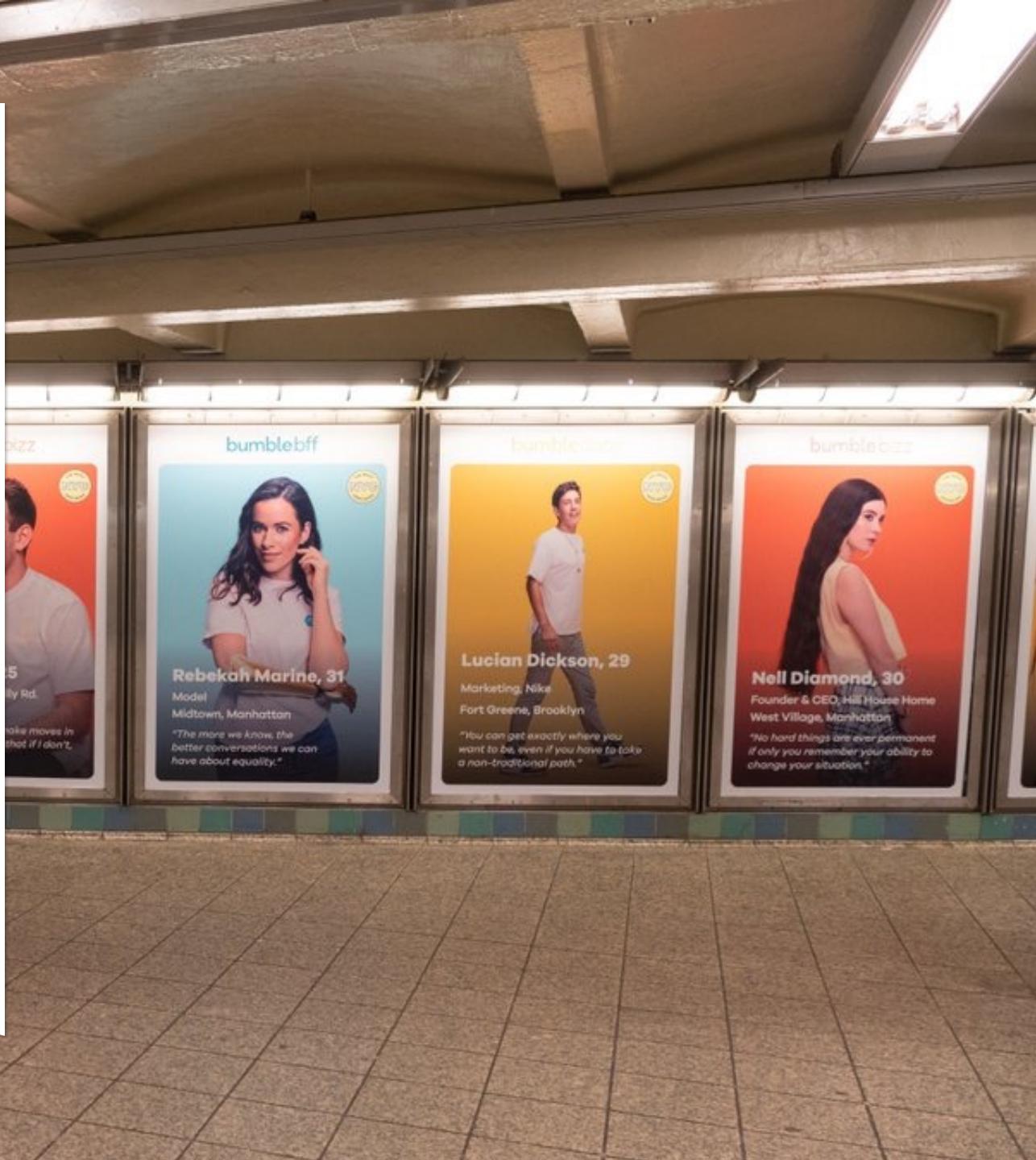


# DAILY EXITS



# CONCLUSION

- **Strategy 1** – base purchase on total activity of a station and buy platform ads
- Cost per ad per Subway platform: \$40,000 to cover 200 stations (cheaper per unit station, but more competition and smaller space)



## CONCLUSION

- **Strategy 2** – base purchase on entries/exits and buy entry/exit ads
- Cost per ad per static subway entrance/exit: \$3,000



# FUTURE WORK

## Insight

- Noticed some stations are not located in Manhattan i.e. St. George (Staten Island), 103 St-Corona (Queens) and Journal Square (Jersey City)
  - Could be a good thing because travelers are most likely locals/commuters
- Some streets also have multiple stations on the same street with the same name i.e. 103 St, 145 St and 7<sup>th</sup> Ave
  - Could have had some data representing a group of stations instead of one

## If I had more time...

- Filter same street stations into individual stations for a more accurate traffic count
- Analyze top station layout for optimal ad space purchase

# APPENDIX

- To view the full project including code, write-up, and presentation slide deck please visit my [Github](#)
- Main source of data is from the [MTA Turnstile Data](#)
- Source of ad space prices found [here](#)

QUESTIONS?

#MadeAtMetis