WTWY

OPTIMISE THE PLACEMENT OF STREET TEAMS

#WomenTechWomenYes

01

- Introduction
- Methodology
- Progress
- Results
- Summary
- Future Work

01

So how can we optimise this in the best way?

I **Identify the most used Turnstiles**

So we can find the stations where people are most active.

II Gender distribution by boroughs

The boroughs with the highest ♀ population can help us.

III Boroughs with the highest donation potential

We would like to know how inclined the people of the district are to DONATE

IV Work efficiency of the teams according to the  report

We would like to know which days the street teams will go out according to the weather conditions. For example, days when the temperature is above 20 degrees and lack of precipitation.

Data Sources

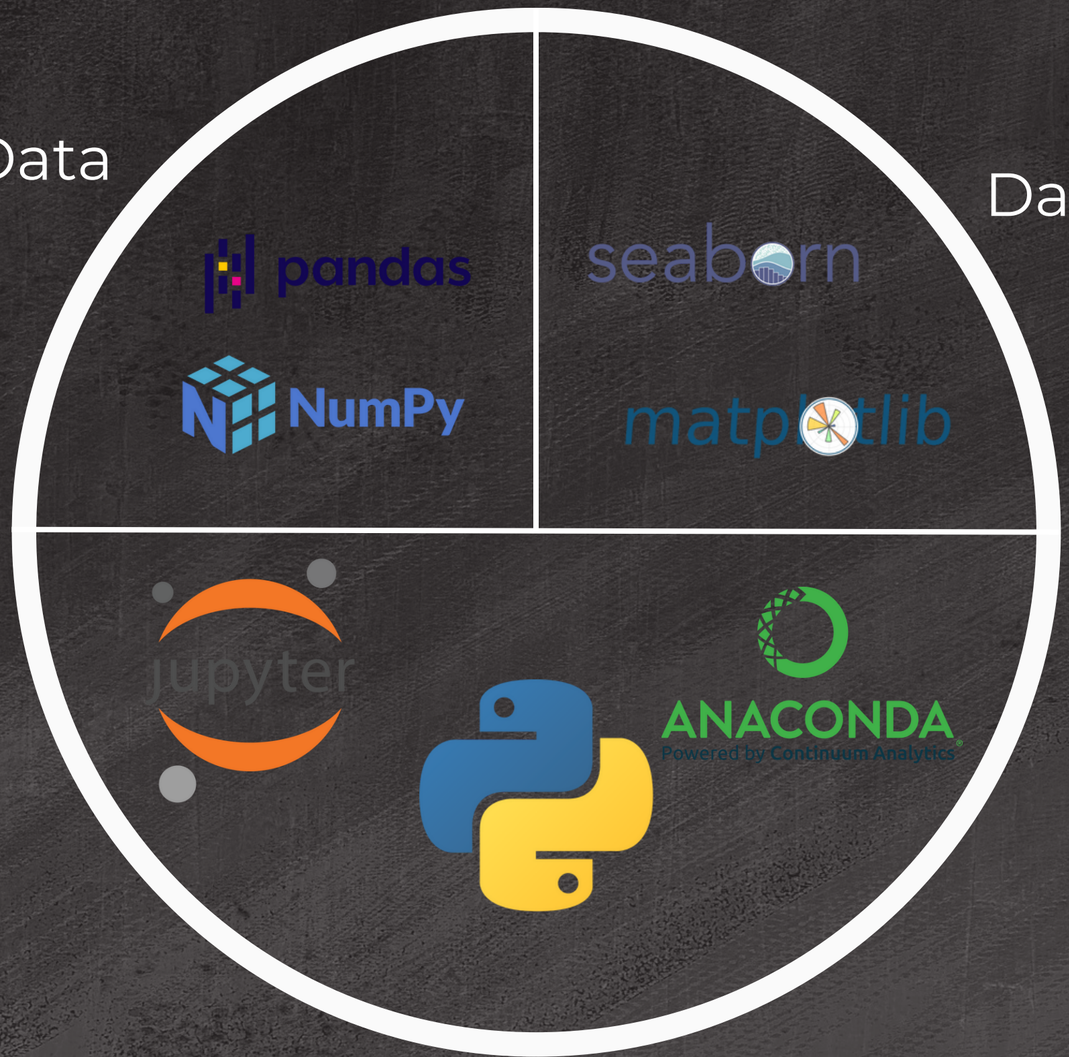
Turnstile Usage Data: 2022

DYCD Participant Demographics by Borough

NYC Weather - 2016 to 2022
(2022 data filtered)

DSNY DonateNYC Directory

Tools


Data
pandas
NumPy
Data Visualization
seaborn
matplotlib
jupyter
ANACONDA
Powered by Continuum Analytics

TURNSTILE USAGE DATA: 2022 FILTERING PROCESSES

We add the libraries we will use in our Jupyter Notebook (this step is common to all)

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

Then we select the data path we will use.
MTA Turnstile Usage Data .

```python
# Veri dosyasının yolu
file_path = r"C:\Users\halha\Downloads\Turnstile_Usage_Data__2022_20240124.csv"

# Veriyi oku
df1 = pd.read_csv(file_path)

# Veriyi yazdır
print(df1)
```

As can be seen here, we have successfully extracted the data.

	C/A	Unit	SCP	Station	Line Name	Division	Date	Time	Description	Entries	Exits
0	A002	R051	02-00-00	59 ST	NQR456W	BMT	12/30/2022	03:00:00	REGULAR	7811029	2770909
1	A002	R051	02-00-00	59 ST	NQR456W	BMT	12/30/2022	07:00:00	REGULAR	7811032	2770930
2	A002	R051	02-00-00	59 ST	NQR456W	BMT	12/30/2022	11:00:00	REGULAR	7811067	2771080
3	A002	R051	02-00-00	59 ST	NQR456W	BMT	12/30/2022	15:00:00	REGULAR	7811217	2771126
4	A002	R051	02-00-00	59 ST	NQR456W	BMT	12/30/2022	19:00:00	REGULAR	7811477	2771174
...	...	...	...	...	...	...	...	...	...	...	...
10963246	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	01/01/2022	00:00:00	REGULAR	5562	1001
10963247	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	01/01/2022	04:00:00	REGULAR	5562	1001
10963248	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	01/01/2022	08:00:00	REGULAR	5562	1001
10963249	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	01/01/2022	16:00:00	REGULAR	5562	1006
10963250	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	01/01/2022	20:00:00	REGULAR	5562	1007

10963251 rows × 11 columns

We check whether it is ⊘ data.

```
In [6]: df1.isnull().sum()

Out[6]: C/A            0
        Unit           0
        SCP            0
        Station        0
        Line Name      0
        Division       0
        Date           0
        Time           0
        Description    0
        Entries        0
        Exits          0
        dtype: int64
```

```
df1.duplicated()

0           False
1           False
2           False
3           False
4           False
            ...
10963246    False
10963247    False
10963248    False
10963249    False
10963250    False
Length: 10963251, dtype: bool
```

We check for duplicate rows.

NYC WEATHER FILTERING PROCESSES

Here we add NYC Weather - 2016 to 2022 as a supplement to our MTA Turnstile Usage Dataset.

As we go through similar steps, our NYC Weather dataset takes the form we want.

	Date	temperature_2m (°C)	precipitation (mm)	rain (mm)	cloudcover (%)	cloudcover_low (%)	cloudcover_mid (%)	cloudcover_high (%)	windspeed_10m (km/h)	winddirection_10m (°)
0	01/01/2016	7.6	0.0	0.0	69.0	53.0	0.0	72.0	10.0	296.0
1	01/01/2016	7.5	0.0	0.0	20.0	4.0	0.0	56.0	9.8	287.0
2	01/01/2016	7.1	0.0	0.0	32.0	3.0	0.0	99.0	9.7	285.0
3	01/01/2016	6.6	0.0	0.0	35.0	5.0	0.0	100.0	9.2	281.0
4	01/01/2016	6.3	0.0	0.0	34.0	4.0	0.0	100.0	9.1	279.0
...	...	...	...	...	...	...	...	...	...	...
59582	10/19/2022	12.2	0.0	0.0	68.0	9.0	100.0	0.0	21.2	212.0
59583	10/19/2022	12.2	0.0	0.0	61.0	8.0	90.0	0.0	20.3	207.0
59584	10/19/2022	12.2	0.0	0.0	62.0	21.0	72.0	0.0	16.6	207.0
59585	10/19/2022	11.1	0.0	0.0	4.0	0.0	6.0	0.0	10.8	206.0
59586	10/19/2022	10.4	0.0	0.0	0.0	0.0	0.0	0.0	7.7	217.0

59587 rows × 10 columns

```
In [9]: file_path = r"C:\Users\halha\OneDrive\Masaüstü\archive\NYC_Weather_2016_2022.csv"

        df2 = pd.read_csv(file_path)
        print(df2)
```

```
                    Date  temperature_2m (°C)  precipitation (mm)  \
0      2016-01-01 00:00:00                  7.6                 0.0
1      2016-01-01 01:00:00                  7.5                 0.0
2      2016-01-01 02:00:00                  7.1                 0.0
3      2016-01-01 03:00:00                  6.6                 0.0
4      2016-01-01 04:00:00                  6.3                 0.0
...                    ...                  ...                 ...
59582  2022-10-19 19:00:00                 12.2                 0.0
59583  2022-10-19 20:00:00                 12.2                 0.0
59584  2022-10-19 21:00:00                 12.2                 0.0
59585  2022-10-19 22:00:00                 11.1                 0.0
59586  2022-10-19 23:00:00                 10.4                 0.0

       rain (mm)  cloudcover (%)  cloudcover_low (%)  cloudcover_mid (%)  \
0            0.0            69.0                53.0                 0.0
1            0.0            20.0                 4.0                 0.0
2            0.0            32.0                 3.0                 0.0
3            0.0            35.0                 5.0                 0.0
4            0.0            34.0                 4.0                 0.0
...          ...             ...                 ...                 ...
59582        0.0            68.0                 9.0               100.0
59583        0.0            61.0                 8.0                90.0
59584        0.0            62.0                21.0                72.0
59585        0.0             4.0                 0.0                 6.0
59586        0.0             0.0                 0.0                 0.0

       cloudcover_high (%)  windspeed_10m (km/h)  winddirection_10m (°)
0                     72.0                  10.0                  296.0
1                     56.0                   9.8                  287.0
2                     99.0                   9.7                  285.0
3                    100.0                   9.2                  281.0
4                    100.0                   9.1                  279.0
...                    ...                   ...                    ...
59582                  0.0                  21.2                  212.0
59583                  0.0                  20.3                  207.0
59584                  0.0                  16.6                  207.0
59585                  0.0                  10.8                  206.0
59586                  0.0                   7.7                  217.0

[59587 rows x 10 columns]
```

What we want to do now is to merge the two data sets. We will do this via the “**Date**“ column.

However, there is a problem we encounter here. In the NYC Weather data set, the date is written as DD/MM/YYYYY. In order to merge it with our main data set, we need to edit it as MM/DD/YYYYYY.

Line Name	Division	Date
NQR456W	BMT	12/30/2022
NQR456W	BMT	12/30/2022
NQR456W	BMT	12/30/2022
NQR456W	BMT	12/30/2022
NQR456W	BMT	12/30/2022
...	...	...
R	RIT	01/01/2022
R	RIT	01/01/2022
R	RIT	01/01/2022
R	RIT	01/01/2022

MTA Turnstile Usage Dataset

	Date	tem
0	01-01-2016	
1	01-01-2016	
2	01-01-2016	
3	01-01-2016	
4	01-01-2016	
...	...	
59582	19-10-2022	
59583	19-10-2022	

NYC Weather dataset

```python
# Tarih sütununu MM/DD/YYYY formatına çevir
df2['Date'] = pd.to_datetime(df2['Date'], format='%d-%m-%Y').dt.strftime('%m/%d/%Y')

# Sonucu yazdır
print(df2)
```

Edited as MM/DD/YYYYYY.

	Date	temperature_2m (°C)	precipitation (mm)	rain (mm)	cloudcover (%)	cloudcover_low (%)	cloudcover_mid (%)	cloudcover_high (%)	windspeed_10m (km/h)	winddirection_10m (°)
0	01/01/2016	7.6	0.0	0.0	69.0	53.0	0.0	72.0	10.0	296.0
1	01/01/2016	7.5	0.0	0.0	20.0	4.0	0.0	56.0	9.8	287.0
2	01/01/2016	7.1	0.0	0.0	32.0	3.0	0.0	99.0	9.7	285.0
3	01/01/2016	6.6	0.0	0.0	35.0	5.0	0.0	100.0	9.2	281.0
4	01/01/2016	6.3	0.0	0.0	34.0	4.0	0.0	100.0	9.1	279.0
...	...	...	...	...	...	...	...	...	...	...
59582	10/19/2022	12.2	0.0	0.0	68.0	9.0	100.0	0.0	21.2	212.0
59583	10/19/2022	12.2	0.0	0.0	61.0	8.0	90.0	0.0	20.3	207.0
59584	10/19/2022	12.2	0.0	0.0	62.0	21.0	72.0	0.0	16.6	207.0
59585	10/19/2022	11.1	0.0	0.0	4.0	0.0	6.0	0.0	10.8	206.0
59586	10/19/2022	10.4	0.0	0.0	0.0	0.0	0.0	0.0	7.7	217.0

59587 rows × 10 columns

We can go back to the **merge step**.

What we want to do now is to merge the two data sets. We will do this through the **“Date“** column.

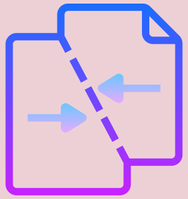
And of course we need to do some filtering for a more accurate result.
We want street teams to work when it is **20 degrees and above lack of precipitation**.

Because we think there will be more people using the metro at these periods.

Start time: 03-15-2022
End time: 06-01-2022

```python
# Tarih sütunlarını datetime formatına çevir
df1['Date'] = pd.to_datetime(df1['Date'], format='%m/%d/%Y')
df2['Date'] = pd.to_datetime(df2['Date'], format='%m/%d/%Y')

# Belirli bir tarih aralığı belirleme
start_date = '03-15-2022'
end_date = '06-01-2022'

# Tarih aralığına göre filtreleme
filtered_df1 = df1[(df1['Date'] >= start_date) & (df1['Date'] <= end_date)]
filtered_df2 = df2[(df2['Date'] >= start_date) & (df2['Date'] <= end_date)]

# İki filtrelenmiş veri setini tarih sütunlarına göre birleştir
merged_df = pd.merge(filtered_df1, filtered_df2, on='Date')

# Sonucu yazdır
print(merged_df)
```

So, our data was merged.



from IPython.display import display

pd.options.display.max_columns = None
display(merged_df)

	C/A	Unit	SCP	Station	Line Name	Division	Date	Time	Description	Entries	Exits	temperature_2m (°C)	
0	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-06-01	00:00:00	REGULAR	7719083	2710086	4.8	
1	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-06-01	00:00:00	REGULAR	7719083	2710086	4.2	
2	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-06-01	00:00:00	REGULAR	7719083	2710086	3.1	
3	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-06-01	00:00:00	REGULAR	7719083	2710086	2.8	
4	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-06-01	00:00:00	REGULAR	7719083	2710086	2.4	
...	...	...	...	...	...	...	...	...	...	...	...	...	
54252961	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-03-15	21:00:00	REGULAR	0	23	13.9	
54252962	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-03-15	21:00:00	REGULAR	0	23	14.5	
54252963	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-03-15	21:00:00	REGULAR	0	23	14.2	
54252964	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-03-15	21:00:00	REGULAR	0	23	13.7	
54252965	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-03-15	21:00:00	REGULAR	0	23	13.5	

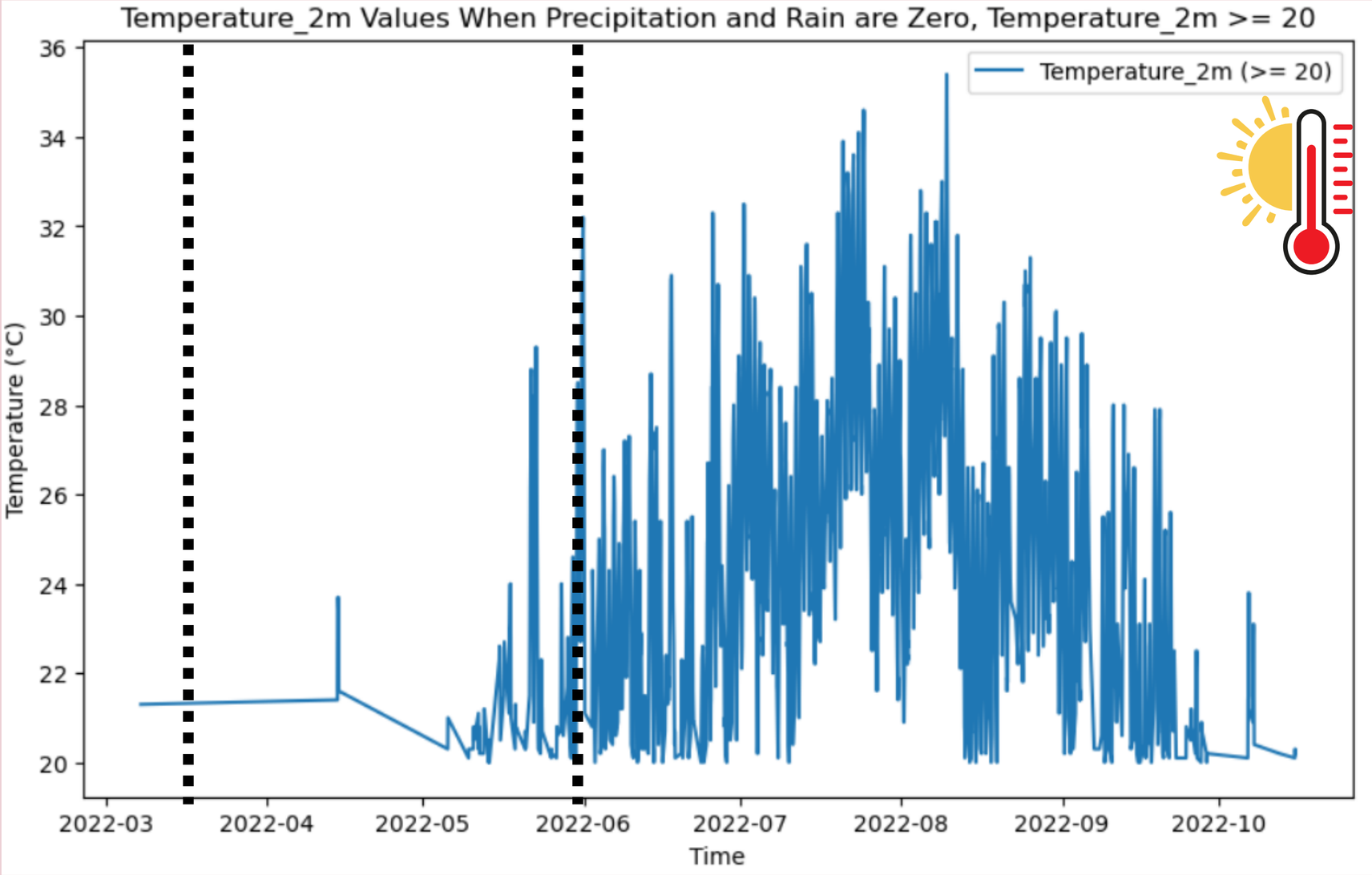
54252966 rows × 20 columns

```python
# precipitation ve rain değerleri sıfır olan ve temperature_2m değeri 20 veya daha büyük olan durumları seç
filtrelenmis_durumlar = merged_df[(merged_df['precipitation (mm)'] == 0)
                                  & (merged_df['rain (mm)'] == 0) & (merged_df['temperature_2m (°C)'] >= 20)]
print(filtrelenmis_durumlar)
```

We filter the range where the temperature is higher than 20 degrees and without precipitation.

	C/A	Unit	SCP	Station	Line Name	Division	Date	Time	Description	Entries	Exits	temperature_2m (°C)	precipitation (mm)	rain (mm)	clou
724248	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-05-31	00:00:00	REGULAR	7718566	2709606	26.9	0.0	0.0	
724249	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-05-31	00:00:00	REGULAR	7718566	2709606	26.3	0.0	0.0	
724250	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-05-31	00:00:00	REGULAR	7718566	2709606	25.6	0.0	0.0	
724251	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-05-31	00:00:00	REGULAR	7718566	2709606	24.5	0.0	0.0	
724252	A002	R051	02-00-00	59 ST	NQR456W	BMT	2022-05-31	00:00:00	REGULAR	7718566	2709606	24.1	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
38352241	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-04-06	21:00:00	REGULAR	0	68	26.8	0.0	0.0	
38352242	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-04-06	21:00:00	REGULAR	0	68	27.0	0.0	0.0	
38352243	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-04-06	21:00:00	REGULAR	0	68	26.8	0.0	0.0	
38352244	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-04-06	21:00:00	REGULAR	0	68	25.8	0.0	0.0	
38352245	TRAM2	R469	00-05-01	RIT-ROOSEVELT	R	RIT	2022-04-06	21:00:00	REGULAR	0	68	25.4	0.0	0.0	

9452797 rows × 20 columns



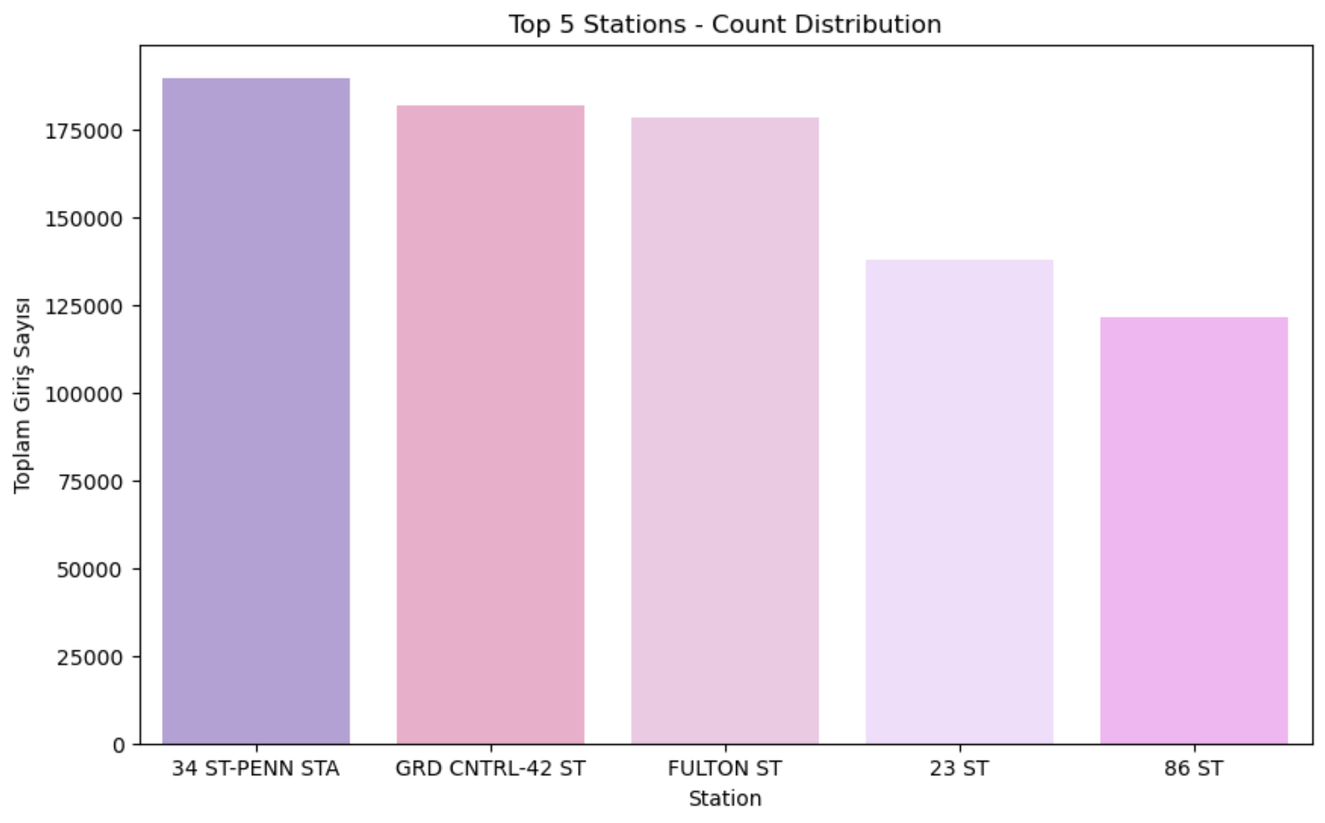
The best time to deploy street teams at metro stations is **03.15.2022-06.01.2022** as shown in the chart.

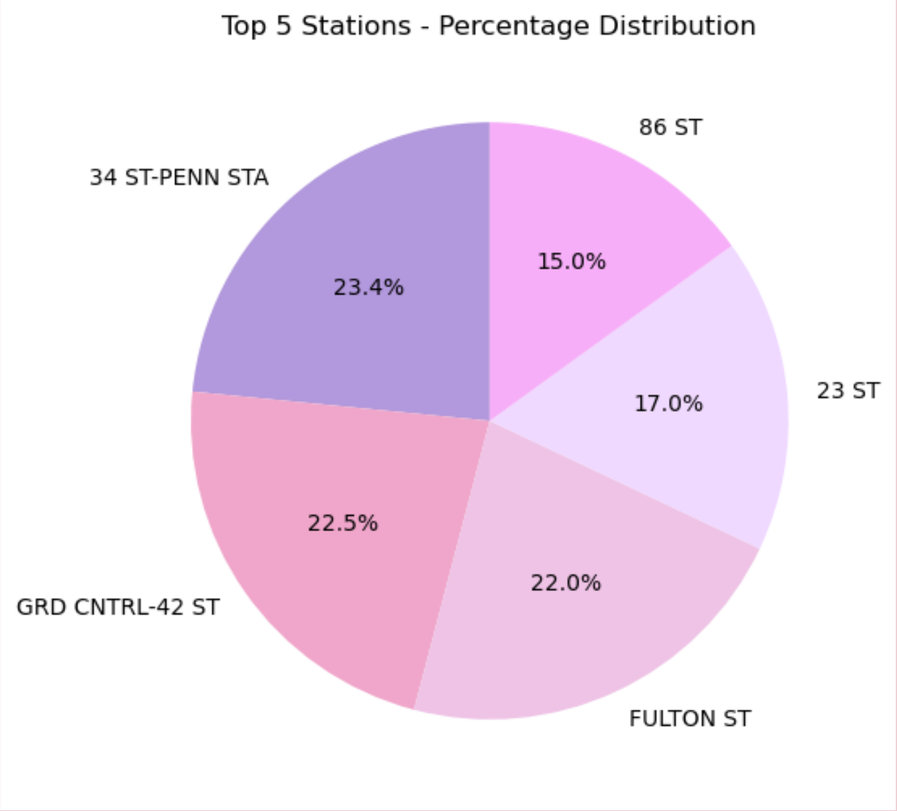
Now we can see the most used stations.

The 5 most used stations are as shown in the picture.

```
In [54]: filtrelenmis_durumlar.Station.value_counts() #kaç tane olduğunu

Out[54]: 34 ST-PENN STA     189824
         GRD CNTRL-42 ST    182120
         FULTON ST          178544
         23 ST              137830
         86 ST              121708
                             ...
         CORTELYOU RD         5538
         CHAUNCEY ST          5312
         SUTTER AV            3756
         CLEVELAND ST         3756
         ORCHARD BEACH        3254
         Name: Station, Length: 379, dtype: int64
```

```
In [55]: #en çok giriş-çıkış yapılan ilk beş durak
         top_stations_new = filtrelenmis_durumlar['Station'].value_counts().head(5)
         print(top_stations_new)

         34 ST-PENN STA     189824
         GRD CNTRL-42 ST    182120
         FULTON ST          178544
         23 ST              137830
         86 ST              121708
         Name: Station, dtype: int64
```


Top 5 Stations - Count Distribution
Toplam Giriş Sayısı
175000
150000
125000
100000
75000
50000
25000
0
34 ST-PENN STA
GRD CNTRL-42 ST
FULTON ST
23 ST
86 ST
Station


Top 5 Stations - Percentage Distribution
86 ST
34 ST-PENN STA
15.0%
23.4%
17.0%
23 ST
22.5%
22.0%
GRD CNTRL-42 ST
FULTON ST

DYCD PARTICIPANT DEMOGRAPHICS BY BOROUGH FILTERING PROCESSES

We will examine one of our additional data sources, DYCD Participant Demographics by Borough.

What we will investigate here is the **distribution of females by borough.**

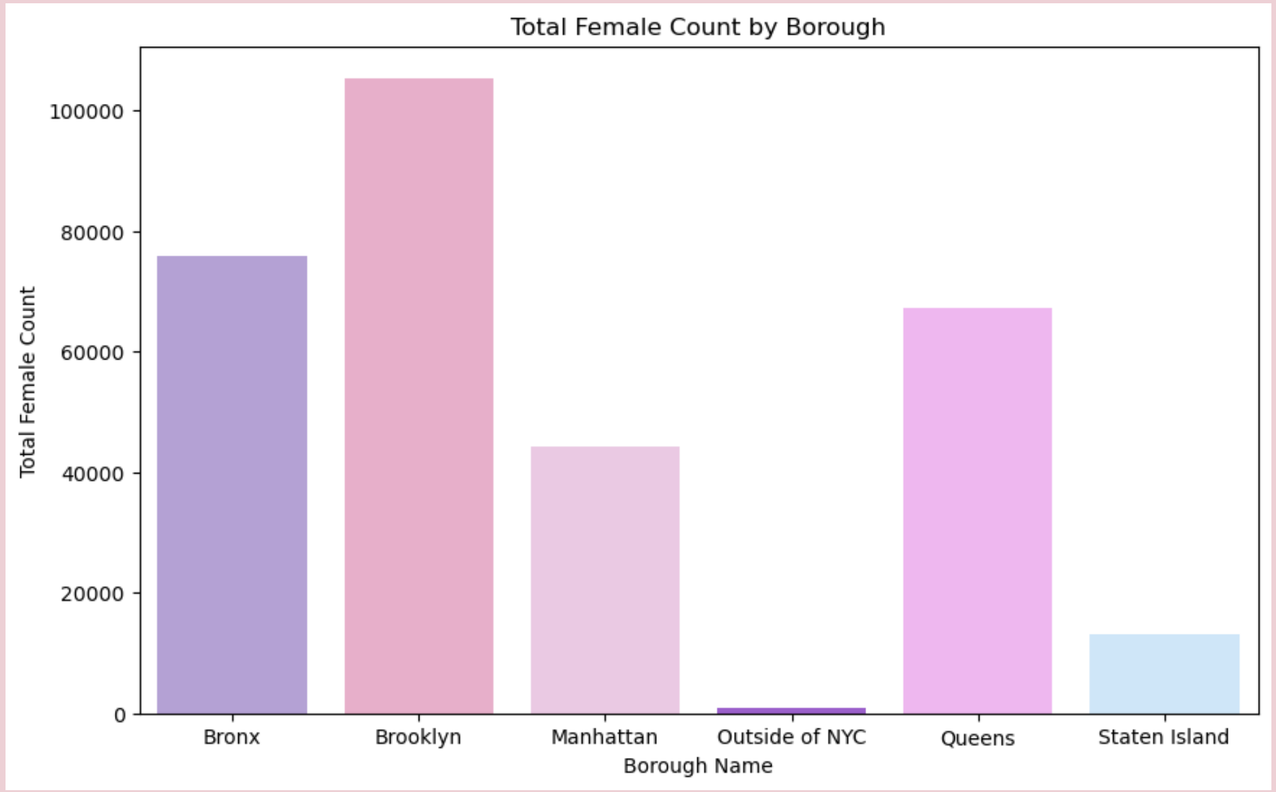
	data_os_of_date	boroughname	programtypename	female_count	female_percentage	male_count	m
0	2020-06-26T00:00:00.000	Brooklyn	Adult Literacy - ABE/HSE (Discretionary)	94	3.08	47	
1	2020-06-26T00:00:00.000	Brooklyn	Adult Literacy - BENL/ESOL (Discretionary)	708	23.25	250	
2	2020-06-26T00:00:00.000	Brooklyn	Adolescent Literacy	28	0.91	25	
3	2020-06-26T00:00:00.000	Brooklyn	Adult Literacy - NDA (ABE/HSE/Spanish HSE)	323	10.60	158	
4	2020-06-26T00:00:00.000	Brooklyn	Adult Literacy - BENL/ESOL	1810	59.45	646	
...	...	...	...	...	...	...	

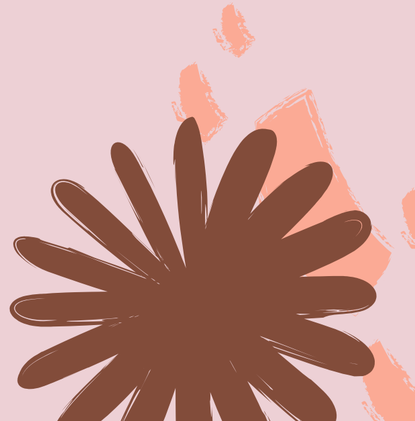
```
In [36]: toplam_female_count_by_borough = df.groupby('boroughname')['female_count'].sum()
         print(toplam_female_count_by_borough)

boroughname
Bronx              75933
Brooklyn          105393
Manhattan          44155
Outside of NYC       984
Queens             67270
Staten Island      13090
Name: female_count, dtype: int64
```

Number of females by boroughs.

Brooklyn>Bronx>Queens>Manhattan>Staten Island>Outside of NYC

```
In [34]: subset_df = df[['boroughname', 'female_count']]

         # boroughname'e göre female_count'ları topla
         toplam_female_count_by_borough = subset_df.groupby('boroughname')['female_count'].sum().reset_index()

         # Çubuk grafiğini çiz
         plt.figure(figsize=(10, 6))
         sns.barplot(x='boroughname', y='female_count', data=toplam_female_count_by_borough, palette=["#b298dc", "#F0A6CA","#EFC3E6", "#
         plt.title('Total Female Count by Borough')  # "Toplam" yerine "Total"
         plt.xlabel('Borough Name')
         plt.ylabel('Total Female Count')  # "Toplam" yerine "Total"
         plt.show()
```





DSNY DONATENYC DIRECTORY FILTERING PROCESSES

We will examine one of our additional data sources, DSNY DonateNYC Directory.

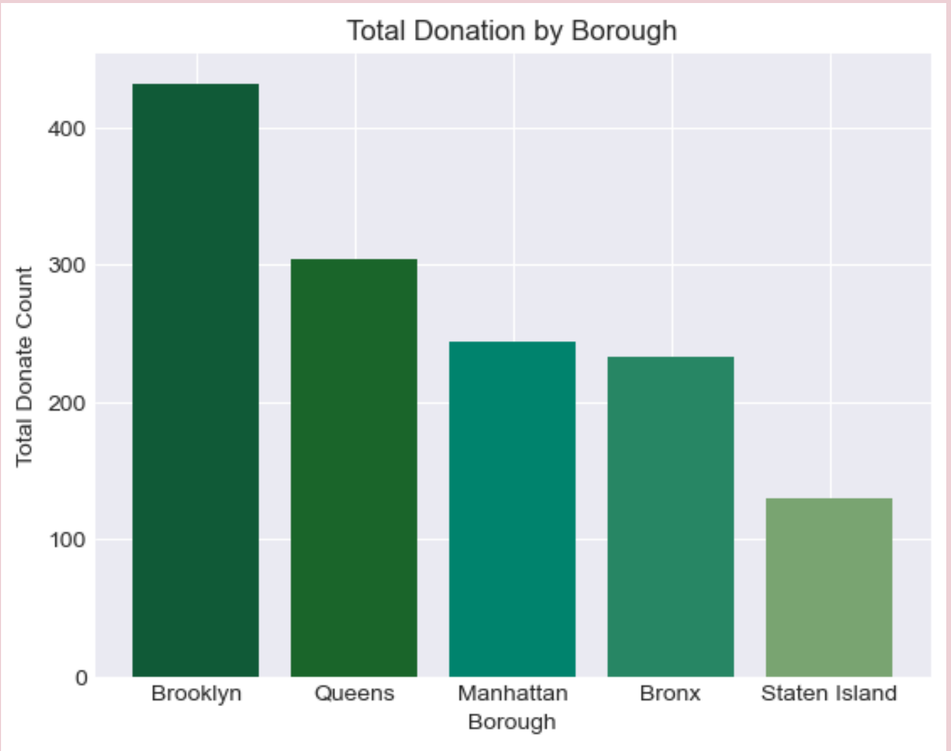
What we will investigate here is the **distribution of donation rates by borough.**

	Borough	NTAName	SiteName	SiteAddress	CategoriesAccepted	CategoriesAvailable	MoreInfo	PickupStatus	Hours
0	Queens	Jackson Heights	Parents in Action	9551 Roosevelt Avenue, Flushing, NY 11372, USA	Food/Beverage	NaN	NaN	NaN	Monday 8:00 AM-9:00 PM, Tuesday 8:00 AM-9:00 P...
1	Brooklyn	East New York (North)	Maribel's Clothing & Accessories	2301 Pitkin Avenue, Brooklyn, NY 11207, USA	NaN	Clothing/Personal Accessories, Footwear	NaN	NaN	Monday 10:30 AM-06:00 PM, Tuesday 10:30 AM-06:...
2	Manhattan	Washington Heights (South)	Sister's Uptown Bookstore	1942 Amsterdam Avenue, New York, NY 10032, USA	Books/Media	Books/Media	NaN	NaN	Monday 01:00 PM-09:00 PM, Tuesday 01:00 PM-09:...
3	Manhattan	Hamilton Heights-Sugar Hill	Harlem Collective	1850 Amsterdam Ave, New York, NY, USA	Musical Instruments, Personal Care/Hygiene	NaN	We accept guitars in good working condition an...	NaN	Monday 9:00 AM-5:00 PM, Tuesday 9:00 AM-5:00 P...
4	Bronx	Soundview-Bruckner-Bronx River	El Amanecer de la Esperanza	1769 East 172nd Street, Bronx, NY 10472, USA	Food/Beverage	Food/Beverage	Accepts donations of canned and packaged food ...	NaN	Monday Closed, Tuesday Closed, Wednesday 09:00...
...	...	...	...	...	...	...	...	...	...

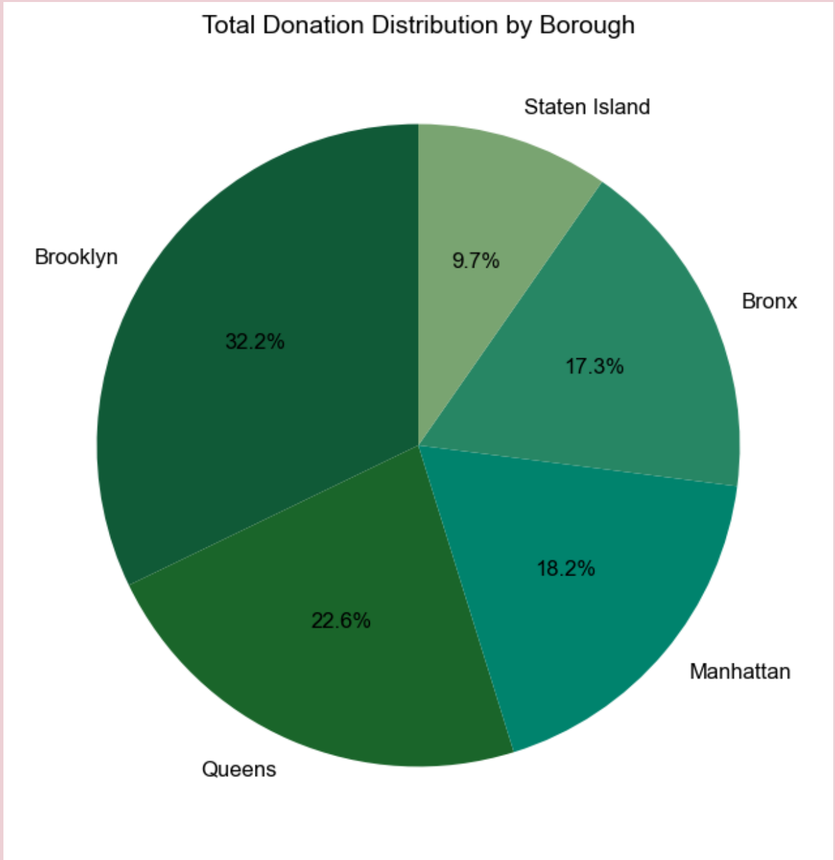
```
In [18]: df.Borough.value_counts()

Out[18]: Brooklyn         432
         Queens           304
         Manhattan        244
         Bronx            233
         Staten Island    130
         Name: Borough, dtype: int64
```

Brooklyn > Queens > Manhattan > Bronx > Staten Island

In [36]:
```python
# Renk paletini belirle
custom_palette = ["#105a37", "#1a652a", "#00836d", "#278664", "#79a471"]

# Borough sütunundaki değerleri grupla ve toplam sayıyı hesapla
borough_counts = df['Borough'].value_counts()

# Çubuk grafiğini çiz
plt.bar(borough_counts.index, borough_counts.values, color=custom_palette)
plt.xlabel('Borough')
plt.ylabel('Total Donate Count')
plt.title('Total Donation by Borough')
plt.show()
```



In [34]:
```
# Renk paletini belirle
custom_palette = ["#105a37", "#1a652a", "#00836d", "#278664", "#79a471"]

# Borough sütunundaki değerleri grupla ve toplam sayıyı hesapla
borough_counts = df['Borough'].value_counts()

# Pasta grafiğini çiz
plt.figure(figsize=(8, 8))
plt.pie(borough_counts, labels=borough_counts.index, autopct='%1.1f%%', startangle=90, colors=custom_palette, textprops={'color':
plt.title('Total Donation Distribution by Borough', fontdict={'fontsize': 14, 'color': 'black'})  # Başlık boyutu ve rengi
plt.show()
```



```
In [58]: top_stations = merged_df['Station'].value_counts().head(25)
         print(top_stations)

34 ST-PENN STA     1096410
GRD CNTRL-42 ST    1039332
FULTON ST          1034650
23 ST               795813
86 ST               700200
CANAL ST            626988
TIMES SQ-42 ST      617982
59 ST               604752
CHAMBERS ST         594606
34 ST-HERALD SQ     587076
ATL AV-BARCLAY      571014
14 ST               550184
42 ST-PORT AUTH     548838
WALL ST             534924
125 ST              462846
PATH NEW WTC        449562
14 ST-UNION SQ      445266
59 ST COLUMBUS      442650
CORTLANDT ST        422358
28 ST               420678
JAY ST-METROTEC     416772
CHURCH AV           402882
161/YANKEE STAD     400098
96 ST               399786
47-50 STS ROCK      391608
Name: Station, dtype: int64
```

These are the 2 most used metro stations in Brooklyn.

The reason why we filtered Brooklyn specifically: **Both the female population of the boroughs and the rate of donations are highest**.

We had previously identified the 5 most used metro stations from turnstile and weather data.

Thus, we position our street teams at **7 metro stations.** So, we think we will get the highest efficiency.

These are:

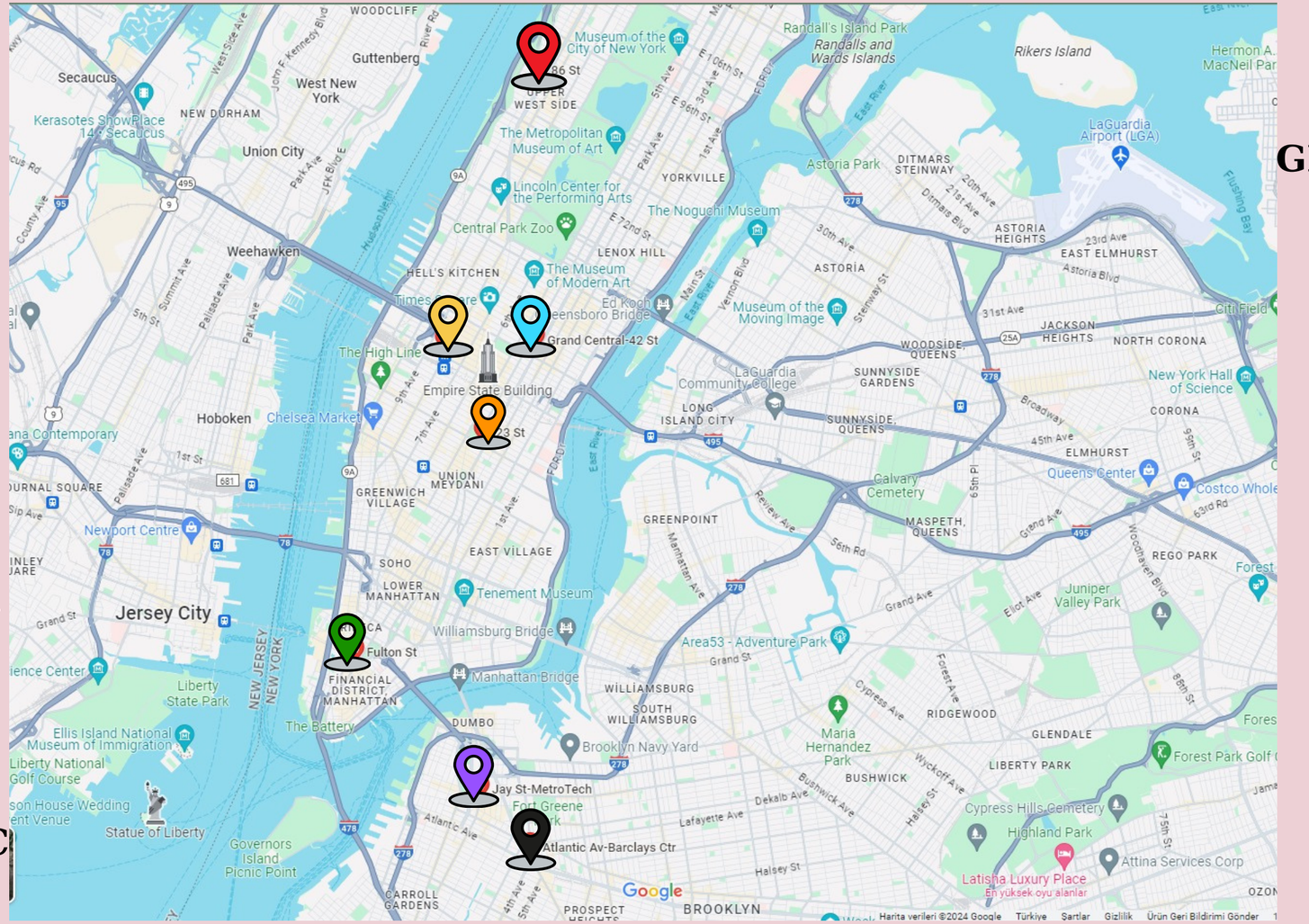
- 34 ST-PENN STA
- GRD CNTRL-42 ST
- FULTON ST
- 23 ST
- 86 ST
- ATL AV-BARCLAY
- JAY ST-METROTEC

34 ST-PENN STA

FULTON ST

ATL AV-BARCLAY

JAY ST-METROTEC



GRD CNTRL-42 ST

23 ST

86 ST

- [x] Metro data,
- [x] Gender in the district,
- [x] County donation,
- [x] Weather conditions

We have analysed and finally, the most optimal solution would be to assign street teams to the 7 stations on the map.

This project,

Problem: WTWY targets the maximum number of people at the Gala it organises every year.

Purpose: Optimise the placement of street teams

Solution Proposal: To identify the metro stations where people interact most intensively. Utilising appropriate data for this

Sources Used: We used Turnstile Usage Data, DYCD Participant Demographics by Borough, NYC Weather, DSNY DonateNYC Directory datasets.

Tools Used: Pandas, Numpy, Seaborn, Matplotlib, Jupyter Notebook

The result: Using this datas, we identified the most used metro stations.

Additional data can be added.

For example, Boroughs can be investigated according to technology use..

Visualisation can be improved by using more technical tools.

For example, map visualisation could be improved.

We have prepared this data based on the year 2022. For a more thorough and precise result, they can take a wider time interval.

? If there were more common columns in our data, we could evaluate them all together.

? We couldn't do enough research about this issue because we didn't know the location of the gala.

PRESENTED BY

AZAD HALHALLI
BERKE ILBAY
SONER KOCOGLU

THANK YOU VERY MUCH!

#WomenTechWomenYes