```python
#import all lib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import scipy.stats as stats


# Load the NYC Wi-Fi Hotspot Dataset
df = pd.read_csv(r"C:\Users\LENOVO\Downloads\NYC_Wi-
Fi_Hotspot_Locations (1).csv")

# Preview the dataset
print("First 5 entries:\n", df.head())
```

```
First 5 entries:
    OBJECTID  Borough          Type            Provider
Name  \
0     10604        4  Limited Free            SPECTRUM       Baisley
Pond Park
1     10555        4  Limited Free            SPECTRUM
Kissena Park
2     12370        3          Free   Transit Wireless             Grand
St (L)
3      9893        3          Free  Downtown Brooklyn
NaN
4     10169        1          Free   Transit Wireless  Lexington Av-63
St (F)

                  Location   Latitude  Longitude             X
Y  \
0          Park Perimeter  40.674860 -73.784120  1.044132e+06
185219.892077
1          Park Perimeter  40.747560 -73.818150  1.034638e+06
211685.217755
2            Grand St (L)  40.711926 -73.940670  1.000698e+06
198655.908840
3           125 Court St.  40.689985 -73.991995  9.864700e+05
190656.680416
4  Lexington Av-63 St (F)  40.764630 -73.966115  9.936366e+05
217853.888161

   ... Neighborhood Tabulation Area (NTA) Council Distrcit Postcode
BoroCD  \
0  ...          Springfield Gardens North               28    11434
412
1  ...                           Flushing               20    11355
407
```

```
2  ...                  East Williamsburg                34    11206
301
3  ...       Brooklyn Heights-Cobble Hill                33    11201
302
4  ...      Upper East Side-Carnegie Hill                 4    10065
108

  Census Tract BCTCB2010      BIN         BBL DOITT_ID  \
0          294       294        0           0     1408
1          845       845        0           0     1359
2          495       495        0           0     1699
3            9         9  3388736  3002777501      298
4          120       120        0           0      599

               Location (Lat, Long)
0  (40.6748599999, -73.7841200005)
1  (40.7475599996, -73.8181499997)
2  (40.7119259997, -73.9406699994)
3  (40.6899850001, -73.9919950004)
4  (40.7646300002, -73.9661150001)

[5 rows x 29 columns]
```

```python
# Print complete information about the dataset
print("=== Dataset Info ===")
print(df.info())

# Print top 5 rows
print("\n=== Top 5 Rows ===")
print(df.head())

# Print bottom 5 rows
print("\n=== Bottom 5 Rows ===")
print(df.tail())

# Print complete statistical summary (only for numerical columns)
print("\n=== Statistical Summary ===")
print(df.describe())

# Count of null values in each column
print("\n=== Null Values in Each Column ===")
print(df.isnull().sum())
```

```
=== Dataset Info ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3319 entries, 0 to 3318
Data columns (total 29 columns):
 #   Column                                        Non-Null Count
Dtype
---  ------                                        --------------
```

```
-----
 0   OBJECTID                                          3319 non-null
int64
 1   Borough                                           3319 non-null
int64
 2   Type                                              3319 non-null
object
 3   Provider                                          3319 non-null
object
 4   Name                                              3089 non-null
object
 5   Location                                          3319 non-null
object
 6   Latitude                                          3319 non-null
float64
 7   Longitude                                         3319 non-null
float64
 8   X                                                 3319 non-null
float64
 9   Y                                                 3319 non-null
float64
 10  Location_T                                        3319 non-null
object
 11  Remarks                                           2771 non-null
object
 12  City                                              3319 non-null
object
 13  SSID                                              3319 non-null
object
 14  SourceID                                          2257 non-null
object
 15  Activated                                         3319 non-null
object
 16  BoroCode                                          3319 non-null
int64
 17  Borough Name                                      3319 non-null
object
 18  Neighborhood Tabulation Area Code (NTACODE)  3319 non-null
object
 19  Neighborhood Tabulation Area (NTA)           3319 non-null
object
 20  Council Distrcit                                  3319 non-null
int64
 21  Postcode                                          3319 non-null
int64
 22  BoroCD                                            3319 non-null
int64
 23  Census Tract                                      3319 non-null
int64
```

```
 24  BCTCB2010                                     3319 non-null
int64
 25  BIN                                           3319 non-null
int64
 26  BBL                                           3319 non-null
int64
 27  DOITT_ID                                      3319 non-null
int64
 28  Location (Lat, Long)                          3319 non-null
object
dtypes: float64(4), int64(11), object(14)
memory usage: 752.1+ KB
None

=== Top 5 Rows ===
   OBJECTID  Borough          Type           Provider
Name  \
0     10604        4  Limited Free           SPECTRUM       Baisley
Pond Park
1     10555        4  Limited Free           SPECTRUM
Kissena Park
2     12370        3          Free   Transit Wireless            Grand
St (L)
3      9893        3          Free  Downtown Brooklyn
NaN
4     10169        1          Free   Transit Wireless  Lexington Av-63
St (F)

                 Location   Latitude  Longitude             X
Y  \
0          Park Perimeter  40.674860 -73.784120  1.044132e+06
185219.892077
1          Park Perimeter  40.747560 -73.818150  1.034638e+06
211685.217755
2            Grand St (L)  40.711926 -73.940670  1.000698e+06
198655.908840
3           125 Court St.  40.689985 -73.991995  9.864700e+05
190656.680416
4  Lexington Av-63 St (F)  40.764630 -73.966115  9.936366e+05
217853.888161

   ... Neighborhood Tabulation Area (NTA) Council Distrcit Postcode
BoroCD  \
0  ...          Springfield Gardens North               28    11434
412
1  ...                            Flushing               20    11355
407
2  ...                  East Williamsburg               34    11206
301
3  ...       Brooklyn Heights-Cobble Hill               33    11201
```

```
302
4  ...      Upper East Side-Carnegie Hill                4    10065
108

  Census Tract BCTCB2010      BIN         BBL DOITT_ID  \
0          294       294        0           0     1408
1          845       845        0           0     1359
2          495       495        0           0     1699
3            9         9  3388736  3002777501      298
4          120       120        0           0      599

               Location (Lat, Long)
0  (40.6748599999, -73.7841200005)
1  (40.7475599996, -73.8181499997)
2  (40.7119259997, -73.9406699994)
3  (40.6899850001, -73.9919950004)
4  (40.7646300002, -73.9661150001)

[5 rows x 29 columns]

=== Bottom 5 Rows ===
      OBJECTID  Borough          Type               Provider
Name  \
3314     10872        3  Limited Free               SPECTRUM  Carroll
Park
3315     12026        2          Free  LinkNYC - Citybridge  bx-01-
138789
3316     12063        3          Free  LinkNYC - Citybridge  bk-01-
143982
3317     12066        3          Free  LinkNYC - Citybridge  bk-17-
126527
3318     12083        3          Free  LinkNYC - Citybridge  bk-02-
108355

                                                 Location   Latitude
Longitude  \
3314  Court off Smith St between Carrol St and 1st P...  40.680630 -
73.995382
3315                                  312 WILLIS AVENUE  40.810896 -
73.921434
3316                                   32 GRAHAM AVENUE  40.701930 -
73.942239
3317                                1339 FLATBUSH AVENUE  40.638560 -
73.953603
3318                                 402 Atlantic Avenue  40.686812 -
73.984970

                  X              Y  ...  \
3314  9.855309e+05  187248.314202  ...
3315  1.005999e+06  234718.294065  ...
```

```
3316  1.000265e+06  195013.901033  ...
3317  9.971268e+05  171924.271359  ...
3318  9.884185e+05  189500.763109  ...

                    Neighborhood Tabulation Area (NTA) Council
Distrcit  \
3314          Carroll Gardens-Columbia Street-Red Hook
39
3315                              Mott Haven-Port Morris
8
3316                                   East Williamsburg
34
3317                                             Erasmus
45
3318  DUMBO-Vinegar Hill-Downtown Brooklyn-Boerum Hill
33

     Postcode BoroCD Census Tract BCTCB2010      BIN         BBL
DOITT_ID  \
3314    11231    306           77        77  3007547  3004490015
1338
3315    10454    201           39        39  2000335  2022850010
4113
3316    11206    301          491       491  3071609  3031200000
3018
3317    11226    317          790       790  3120360  3052110060
3021
3318    11217    302           41        41  3255607  3001837500
3038

                  Location (Lat, Long)
3314  (40.6806299998, -73.9953819995)
3315  (40.8108964904, -73.9214341701)
3316  (40.7019303441, -73.9422392381)
3317  (40.6385596088, -73.9536032378)
3318  (40.6868115694, -73.9849696135)

[5 rows x 29 columns]

=== Statistical Summary ===
           OBJECTID      Borough     Latitude    Longitude
X  \
count   3319.000000  3319.000000  3319.000000  3319.000000
3.319000e+03
mean   11279.197349     2.117505    40.741789   -73.950112
9.980701e+05
std      967.314772     1.269830     0.066799     0.062109
1.722035e+04
min     9601.000000     1.000000    40.509531   -74.244107
9.163706e+05
```

```
25%    10445.500000     1.000000    40.695480   -73.985788
9.881911e+05
50%    11286.000000     1.000000    40.745906   -73.962685
9.945985e+05
75%    12116.500000     3.000000    40.791719   -73.926875
1.004498e+06
max    12946.000000     5.000000    40.903723   -73.714838
1.063266e+06

                  Y     BoroCode  Council Distrcit      Postcode  \
count    3319.000000  3319.000000       3319.000000   3319.000000
mean   209540.439778     2.117505         17.210003  10535.245556
std     24334.633844     1.269830         14.845808    583.464313
min    125007.163094     1.000000          1.000000  10001.000000
25%    192679.650451     1.000000          4.000000  10022.000000
50%    211036.902451     1.000000          9.000000  10075.000000
75%    227723.332824     3.000000         32.000000  11211.000000
max    268543.647017     5.000000         51.000000  11694.000000

            BoroCD   Census Tract      BCTCB2010           BIN
BBL  \
count  3319.000000    3319.000000    3319.000000  3.319000e+03
3.319000e+03
mean    218.880988    2353.715276    2353.715276  1.581973e+06
1.751089e+09
std     128.098784   12476.228592   12476.228592  1.453220e+06
1.415620e+09
min     101.000000       1.000000       1.000000  0.000000e+00
0.000000e+00
25%     106.000000      76.000000      76.000000  0.000000e+00
1.007220e+09
50%     112.000000     163.000000     163.000000  1.056626e+06
1.017260e+09
75%     309.000000     271.000000     271.000000  3.018582e+06
3.013530e+09
max     595.000000  157902.000000  157902.000000  5.162377e+06
5.078990e+09

          DOITT_ID
count  3319.000000
mean   2679.822537
std    1598.918823
min       1.000000
25%    1099.500000
50%    3265.000000
75%    4094.500000
max    4950.000000

=== Null Values in Each Column ===
OBJECTID                                          0
```

```
Borough                                              0
Type                                                 0
Provider                                             0
Name                                               230
Location                                             0
Latitude                                             0
Longitude                                            0
X                                                    0
Y                                                    0
Location_T                                           0
Remarks                                            548
City                                                 0
SSID                                                 0
SourceID                                          1062
Activated                                            0
BoroCode                                             0
Borough Name                                         0
Neighborhood Tabulation Area Code (NTACODE)          0
Neighborhood Tabulation Area (NTA)                   0
Council Distrcit                                     0
Postcode                                             0
BoroCD                                               0
Census Tract                                         0
BCTCB2010                                            0
BIN                                                  0
BBL                                                  0
DOITT_ID                                             0
Location (Lat, Long)                                 0
dtype: int64
```

```
# Count of duplicated rows
print("\n=== Number of Duplicate Rows ===")
print(df.duplicated().sum())

# Shape of the dataset
print("\n=== Shape of DataFrame ===")
print(df.shape)
```

```
=== Number of Duplicate Rows ===
0

=== Shape of DataFrame ===
(3319, 29)
```

```
# Column names
print("\n=== Column Names ===")
print(df.columns)

# Drop irrelevant columns (example: 'Provider', 'X')
```

```
df1 = df.drop(['Provider', 'X'], axis=1)  # You can adjust this based
on your analysis
print("\n=== After Dropping Irrelevant Columns ===")
print(df1.head())

# Count of non-null values column-wise
print("\n=== Non-null Count in Each Column ===")
print(df.count())

# Show duplicate rows
print("\n=== Duplicate Rows ===")
print(df[df.duplicated()])


=== Column Names ===
Index(['OBJECTID', 'Borough', 'Type', 'Provider', 'Name', 'Location',
       'Latitude', 'Longitude', 'X', 'Y', 'Location_T', 'Remarks',
'City',
       'SSID', 'SourceID', 'Activated', 'BoroCode', 'Borough Name',
       'Neighborhood Tabulation Area Code (NTACODE)',
       'Neighborhood Tabulation Area (NTA)', 'Council Distrcit',
'Postcode',
       'BoroCD', 'Census Tract', 'BCTCB2010', 'BIN', 'BBL',
'DOITT_ID',
       'Location (Lat, Long)'],
      dtype='object')

=== After Dropping Irrelevant Columns ===
   OBJECTID  Borough          Type                      Name  \
0     10604        4  Limited Free       Baisley Pond Park
1     10555        4  Limited Free            Kissena Park
2     12370        3          Free            Grand St (L)
3      9893        3          Free                     NaN
4     10169        1          Free  Lexington Av-63 St (F)

                 Location   Latitude  Longitude             Y  \
0          Park Perimeter  40.674860 -73.784120  185219.892077
1          Park Perimeter  40.747560 -73.818150  211685.217755
2            Grand St (L)  40.711926 -73.940670  198655.908840
3           125 Court St.  40.689985 -73.991995  190656.680416
4  Lexington Av-63 St (F)  40.764630 -73.966115  217853.888161

           Location_T                 Remarks  ...  \
0  Outdoor TWC Aerial  3 free 10 min sessions  ...
1  Outdoor TWC Aerial  3 free 10 min sessions  ...
2      Subway Station                  SN 123  ...
3             Outdoor                     NaN  ...
4      Subway Station                  SN 223  ...

  Neighborhood Tabulation Area (NTA) Council Distrcit Postcode BoroCD
```

```
\
0          Springfield Gardens North                28    11434    412

1                           Flushing                20    11355    407

2                  East Williamsburg                34    11206    301

3       Brooklyn Heights-Cobble Hill                33    11201    302

4      Upper East Side-Carnegie Hill                 4    10065    108


   Census Tract BCTCB2010      BIN         BBL  DOITT_ID  \
0           294       294        0           0      1408
1           845       845        0           0      1359
2           495       495        0           0      1699
3             9         9  3388736  3002777501       298
4           120       120        0           0       599

              Location (Lat, Long)
0  (40.6748599999, -73.7841200005)
1  (40.7475599996, -73.8181499997)
2  (40.7119259997, -73.9406699994)
3  (40.6899850001, -73.9919950004)
4  (40.7646300002, -73.9661150001)

[5 rows x 27 columns]

=== Non-null Count in Each Column ===
OBJECTID                                          3319
Borough                                           3319
Type                                              3319
Provider                                          3319
Name                                              3089
Location                                          3319
Latitude                                          3319
Longitude                                         3319
X                                                 3319
Y                                                 3319
Location_T                                        3319
Remarks                                           2771
City                                              3319
SSID                                              3319
SourceID                                          2257
Activated                                         3319
BoroCode                                          3319
Borough Name                                      3319
Neighborhood Tabulation Area Code (NTACODE)       3319
Neighborhood Tabulation Area (NTA)                3319
Council Distrcit                                  3319
```

```
Postcode                                    3319
BoroCD                                      3319
Census Tract                                3319
BCTCB2010                                   3319
BIN                                         3319
BBL                                         3319
DOITT_ID                                    3319
Location (Lat, Long)                        3319
dtype: int64

=== Duplicate Rows ===
Empty DataFrame
Columns: [OBJECTID, Borough, Type, Provider, Name, Location, Latitude,
Longitude, X, Y, Location_T, Remarks, City, SSID, SourceID, Activated,
BoroCode, Borough Name, Neighborhood Tabulation Area Code (NTACODE),
Neighborhood Tabulation Area (NTA), Council Distrcit, Postcode,
BoroCD, Census Tract, BCTCB2010, BIN, BBL, DOITT_ID, Location (Lat,
Long)]
Index: []

[0 rows x 29 columns]
```

```
# 1.  Find hotspots provided by 'SPECTRUM' in the 'Bronx' with
'Limited Free' service
spectrum_bronx = df[
    (df["Provider"] == "SPECTRUM") &
    ((df["Borough"] == 2) | (df["Borough Name"].str.upper() ==
"BRONX")) &
    (df["Type"].str.contains("Limited Free", case=False, na=False))
]
print("\n SPECTRUM Limited Free hotspots in Bronx:\n",
spectrum_bronx[["Name", "Location", "Type", "Provider"]])
```

```
 SPECTRUM Limited Free hotspots in Bronx:
 Empty DataFrame
Columns: [Name, Location, Type, Provider]
Index: []
```

```
# 2.  Find hotspots located in parks (based on 'Location' field)
parks_wifi = df[df["Location"].str.contains("Park", case=False,
na=False)]
print("\n Wi-Fi hotspots in Parks:\n", parks_wifi[["Name",
"Location", "Borough Name"]])

# 3.  Identify hotspots without a 'Name' (missing info)
missing_names = df[df["Name"].isna()]
print("\n Hotspots without a name:\n", missing_names[["Location",
"Provider", "Type"]])
```

```
 Wi-Fi hotspots in Parks:
                                      Name  \
0                         Baisley Pond Park
1                              Kissena Park
5                              Kissena Park
10                             Crotona Park
18                           MARIA HERNANDEZ
...                                     ...
3155                      Baisley Pond Park
3231                           Crotona Park
3256                                    NaN
3271  NYC - Detective Keith Williams Park
3273                           Crotona Park

                                              Location Borough Name
0                                     Park Perimeter       Queens
1                                     Park Perimeter       Queens
5                                     Park Perimeter       Queens
10        CROTONA PARK C/O EAST 173RD ST AND FULTON AVE        Bronx
18             IN PARK BELOW PLAYGROUND AREA SOUTH SIDE     Brooklyn
...                                                 ...          ...
3155                                  Park Perimeter       Queens
3231   SOUTH OF PLAYGROUND INSIDE PARK, CROTONA PK EAST        Bronx
3256                            Boro Hall Park 2 (Pole)     Brooklyn
3271                                Outdoor - Park Area       Queens
3273  Crotona Park-CROTONA PARK SOUTH 1/P/E/O CROTON...        Bronx

[176 rows x 3 columns]

 Hotspots without a name:
                                Location           Provider  Type
3                         125 Court St.  Downtown Brooklyn  Free
6                   Pole 94 - LenWS1N133             Harlem  Free
12                  pole 51n - 131NWC5th             Harlem  Free
52                    110 Livingston St.  Downtown Brooklyn  Free
53    254 Flatbush Ave. Extension (pole)  Downtown Brooklyn  Free
...                                  ...                ...   ...
3149                  110 Livingston St.  Downtown Brooklyn  Free
3160         10th between, 17th and 18th            Chelsea  Free
3192               pole 55n - 136NS2EACP             Harlem  Free
3193                Pole 87 - LenWS1N134             Harlem  Free
3256             Boro Hall Park 2 (Pole)  Downtown Brooklyn  Free

[230 rows x 3 columns]
```

```
# We'll simulate "signal strength" based on Longitude (just for practice)
np.random.seed(42)  # For consistent results
signal_1 = np.random.uniform(50, 100, len(df))  # Simulated signal
```

```
strength in %
signal_2 = np.random.uniform(50, 100, len(df))
signal_3 = np.random.uniform(50, 100, len(df))
signal_4 = np.random.uniform(50, 100, len(df))

# Convert to NumPy Array
signals = np.array([signal_1, signal_2, signal_3, signal_4])

# 1.  Mean, Median, Std Dev for each simulated signal array
mean_signal = np.mean(signals, axis=1)
median_signal = np.median(signals, axis=1)
std_signal = np.std(signals, axis=1)

print("\n Simulated Wi-Fi Signal Statistics:")
print("Mean Signal Strength:", mean_signal)
print("Median Signal Strength:", median_signal)
print("Standard Deviation:", std_signal)

# 2.  Highest simulated signal and which 'sensor' it came from
max_signal = np.max(signals)
sensor_max = np.argmax(np.max(signals, axis=1))

print("\n Highest Simulated Signal Strength:", max_signal)
print("Captured by Sensor:", sensor_max + 1)
```

```
 Simulated Wi-Fi Signal Statistics:
Mean Signal Strength: [74.88814182 74.7870873  74.42548506
75.10287756]
Median Signal Strength: [75.13185466 74.62676904 74.09500836
75.03776374]
Standard Deviation: [14.53371954 14.40305819 14.202006   14.54899694]

 Highest Simulated Signal Strength: 99.98588366430653
Captured by Sensor: 1
```

```
# Show all Boroughs and count of hotspots per borough
borough_counts = df['Borough'].value_counts()
print("\nWi-Fi Hotspot Count by Borough:")
print(borough_counts)
```

```
Wi-Fi Hotspot Count by Borough:
Borough
1    1672
3     700
4     531
2     316
5     100
Name: count, dtype: int64
```

```python
# Filter: Hotspots in Manhattan only
manhattan_data = df[df['Borough'] == 'Manhattan']
print("\nManhattan Wi-Fi Hotspots (first 5):")
print(manhattan_data[['Location', 'Provider', 'SSID']].head())

# Filter: Hotspots provided by 'LinkNYC - Citybridge'
linknyc = df[df['Provider'] == 'LinkNYC - Citybridge']
print("\nHotspots provided by LinkNYC - Citybridge:")
print(linknyc[['Borough', 'Location', 'Type', 'SSID']].head())
```

```
Manhattan Wi-Fi Hotspots (first 5):
Empty DataFrame
Columns: [Location, Provider, SSID]
Index: []

Hotspots provided by LinkNYC - Citybridge:
    Borough                  Location  Type                SSID
13        3          181 Court Street  Free  LinkNYC Free Wi-Fi
14        3       620 ATLANTIC AVENUE  Free  LinkNYC Free Wi-Fi
15        5         19 SEAVIEW AVENUE  Free  LinkNYC Free Wi-Fi
17        4  43-40 NORTHERN BOULEVARD  Free  LinkNYC Free Wi-Fi
19        1              237 1 AVENUE  Free  LinkNYC Free Wi-Fi
```

```python
# Find all outdoor hotspots
outdoor_hotspots = df[df['Type'] == 'Outdoor']
print("\nOutdoor Wi-Fi Hotspots:")
print(outdoor_hotspots[['Borough', 'Location', 'Provider']].head())

# Find hotspots with a certain SSID
ssid_query = df[df['SSID'].str.contains("Free", na=False)]
print("\nHotspots with 'Free' in SSID:")
print(ssid_query[['Location', 'SSID']].head())
```

```
Outdoor Wi-Fi Hotspots:
Empty DataFrame
Columns: [Borough, Location, Provider]
Index: []

Hotspots with 'Free' in SSID:
                    Location                SSID
13          181 Court Street  LinkNYC Free Wi-Fi
14       620 ATLANTIC AVENUE  LinkNYC Free Wi-Fi
15         19 SEAVIEW AVENUE  LinkNYC Free Wi-Fi
17  43-40 NORTHERN BOULEVARD  LinkNYC Free Wi-Fi
19              237 1 AVENUE  LinkNYC Free Wi-Fi
```

```python
# Group by Provider and count
provider_group =
df.groupby('Provider').size().reset_index(name='Hotspot Count')
```

```python
print("\nHotspot Count by Provider:")
print(provider_group.sort_values(by='Hotspot Count', ascending=False))

# Show missing data per column
print("\nMissing Values in Dataset:")
print(df.isna().sum())

# Save a filtered dataset (example: Manhattan outdoor hotspots) to a new CSV
filtered = manhattan_data[manhattan_data['Type'] == 'Outdoor']
filtered.to_csv("manhattan_outdoor_wifi.csv", index=False)
print("\nFiltered data saved to 'manhattan_outdoor_wifi.csv'")
```

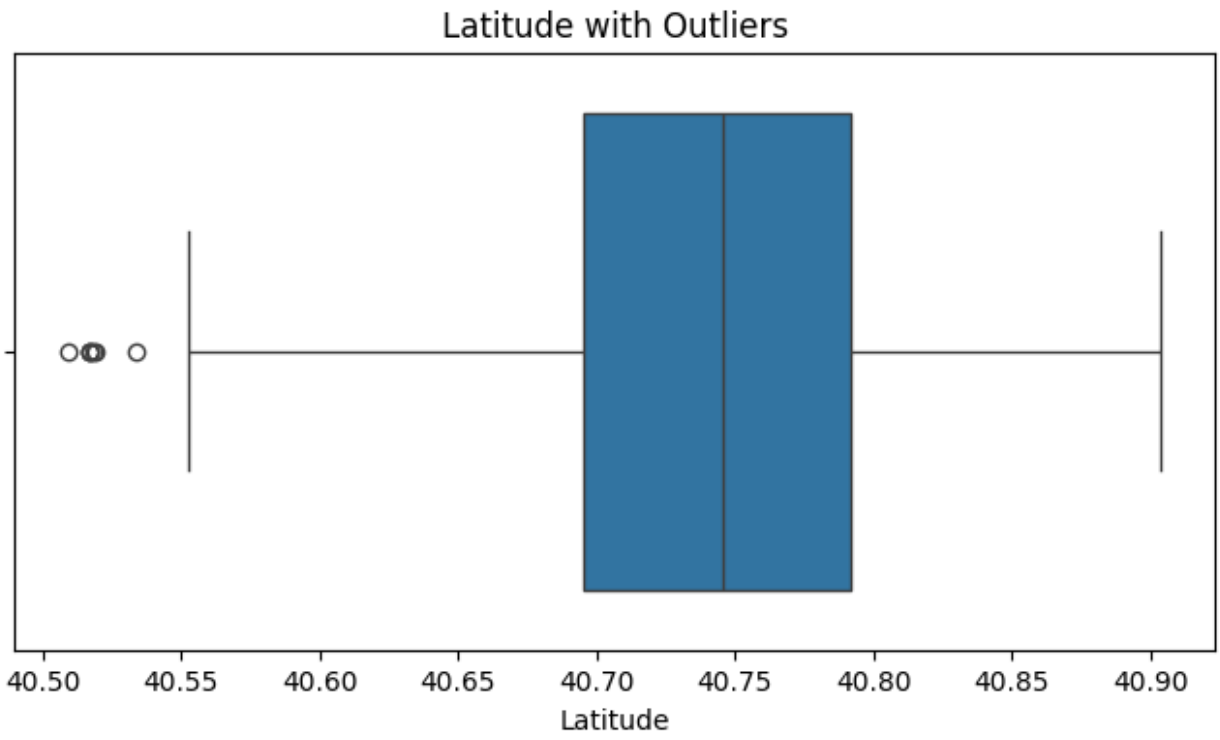
```
Hotspot Count by Provider:
                    Provider  Hotspot Count
8      LinkNYC - Citybridge           1868
14                 SPECTRUM            343
16         Transit Wireless            276
0                 ALTICEUSA            237
7                    Harlem            101
5          Downtown Brooklyn            100
11                      NYPL             90
13                       QPL             65
2                        BPL             59
9   Manhattan Down Alliance             36
3                    Chelsea             30
6                  Fiberless             30
10                     NYCHA             28
1                        AT&T             27
15         Spot On Networks             16
4                  City Tech             11
12                   Partner              2

Missing Values in Dataset:
OBJECTID                0
Borough                 0
Type                    0
Provider                0
Name                  230
Location                0
Latitude                0
Longitude               0
X                       0
Y                       0
Location_T              0
Remarks               548
City                    0
SSID                    0
SourceID             1062
```

```
Activated                                          0
BoroCode                                           0
Borough Name                                       0
Neighborhood Tabulation Area Code (NTACODE)        0
Neighborhood Tabulation Area (NTA)                 0
Council Distrcit                                   0
Postcode                                           0
BoroCD                                             0
Census Tract                                       0
BCTCB2010                                          0
BIN                                                0
BBL                                                0
DOITT_ID                                           0
Location (Lat, Long)                               0
dtype: int64

Filtered data saved to 'manhattan_outdoor_wifi.csv'
```

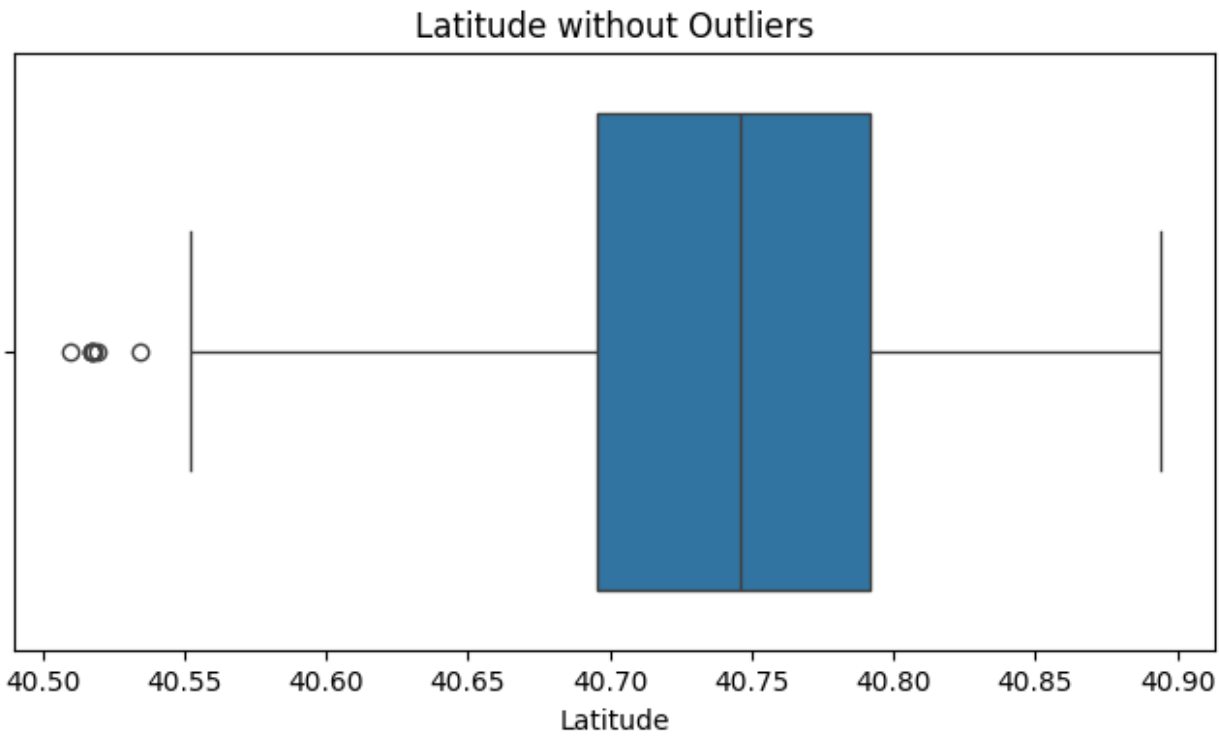
```
# Load NYC Wi-Fi Hotspot dataset
df = pd.read_csv(r"C:\Users\LENOVO\Downloads\NYC_Wi-
Fi_Hotspot_Locations (1).csv")

# Display initial dataset info
print("Initial DataFrame:")
print(df.head())
print(df.columns)

# Let's check for outliers in Latitude using a boxplot
plt.figure(figsize=(8, 4))
sns.boxplot(x=df['Latitude'])
plt.title('Latitude with Outliers')
plt.show()
```

```
Initial DataFrame:
   OBJECTID  Borough          Type           Provider
Name  \
0     10604        4  Limited Free           SPECTRUM       Baisley
Pond Park
1     10555        4  Limited Free           SPECTRUM
Kissena Park
2     12370        3          Free   Transit Wireless            Grand
St (L)
3      9893        3          Free  Downtown Brooklyn
NaN
4     10169        1          Free   Transit Wireless  Lexington Av-63
St (F)

                  Location   Latitude  Longitude             X
Y  \
0          Park Perimeter  40.674860 -73.784120  1.044132e+06
```

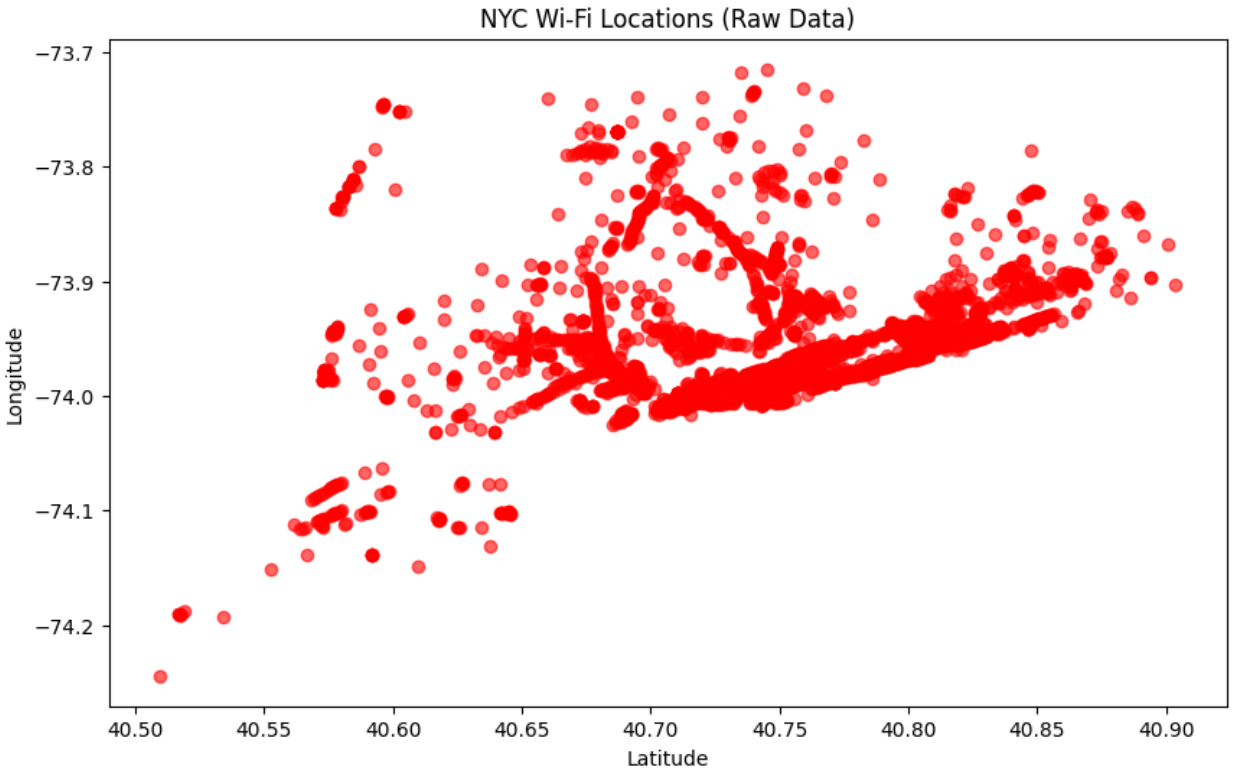
```
185219.892077
1          Park Perimeter  40.747560 -73.818150  1.034638e+06
211685.217755
2            Grand St (L)  40.711926 -73.940670  1.000698e+06
198655.908840
3           125 Court St.  40.689985 -73.991995  9.864700e+05
190656.680416
4  Lexington Av-63 St (F)  40.764630 -73.966115  9.936366e+05
217853.888161

   ... Neighborhood Tabulation Area (NTA) Council Distrcit Postcode
BoroCD  \
0  ...          Springfield Gardens North                28    11434
412
1  ...                            Flushing                20    11355
407
2  ...                   East Williamsburg                34    11206
301
3  ...        Brooklyn Heights-Cobble Hill                33    11201
302
4  ...      Upper East Side-Carnegie Hill                 4    10065
108

  Census Tract BCTCB2010      BIN         BBL DOITT_ID  \
0          294       294        0           0     1408
1          845       845        0           0     1359
2          495       495        0           0     1699
3            9         9  3388736  3002777501      298
4          120       120        0           0      599

               Location (Lat, Long)
0  (40.6748599999, -73.7841200005)
1  (40.7475599996, -73.8181499997)
2  (40.7119259997, -73.9406699994)
3  (40.6899850001, -73.9919950004)
4  (40.7646300002, -73.9661150001)

[5 rows x 29 columns]
Index(['OBJECTID', 'Borough', 'Type', 'Provider', 'Name', 'Location',
       'Latitude', 'Longitude', 'X', 'Y', 'Location_T', 'Remarks',
'City',
       'SSID', 'SourceID', 'Activated', 'BoroCode', 'Borough Name',
       'Neighborhood Tabulation Area Code (NTACODE)',
       'Neighborhood Tabulation Area (NTA)', 'Council Distrcit',
'Postcode',
       'BoroCD', 'Census Tract', 'BCTCB2010', 'BIN', 'BBL',
'DOITT_ID',
       'Location (Lat, Long)'],
      dtype='object')
```



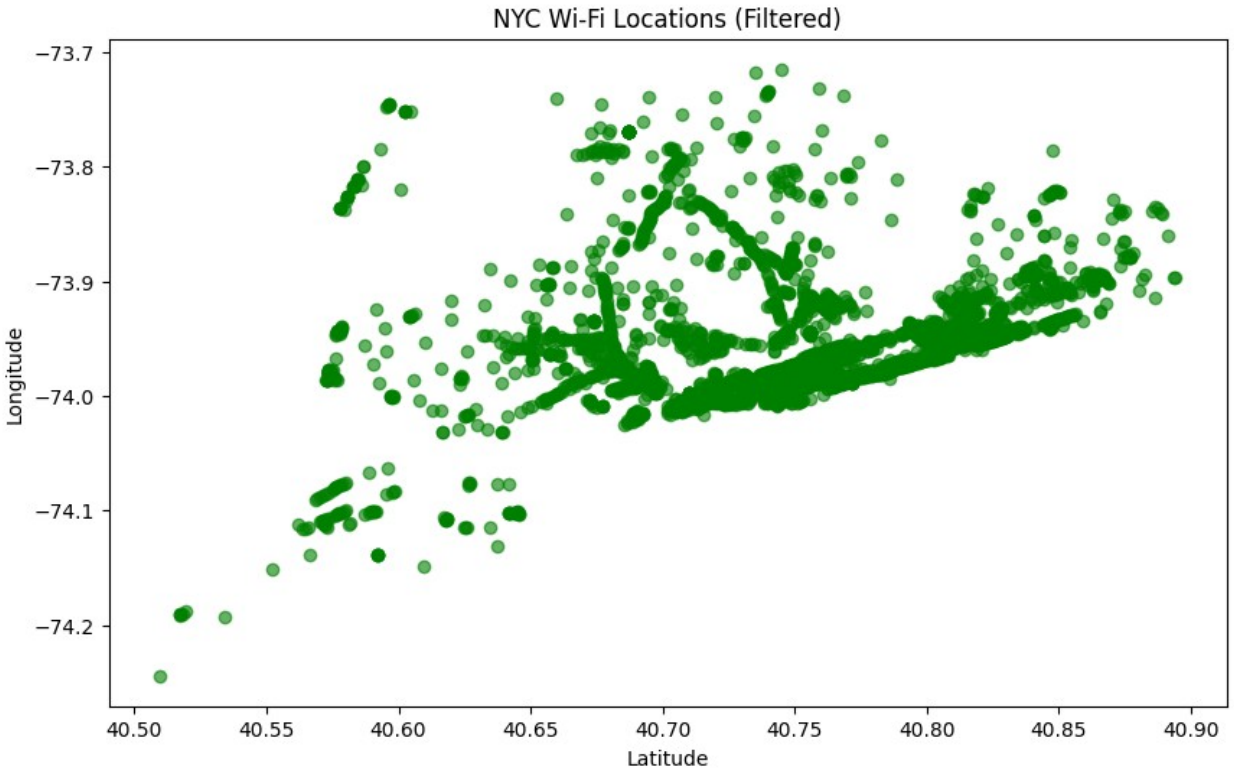

```
# Remove outliers: Latitude values above 40.9 (adjustable threshold)
lat_threshold = 40.9
df_no_lat_outliers = df[df['Latitude'] <= lat_threshold]

plt.figure(figsize=(8, 4))
sns.boxplot(x=df_no_lat_outliers['Latitude'])
plt.title('Latitude without Outliers')
plt.show()
```



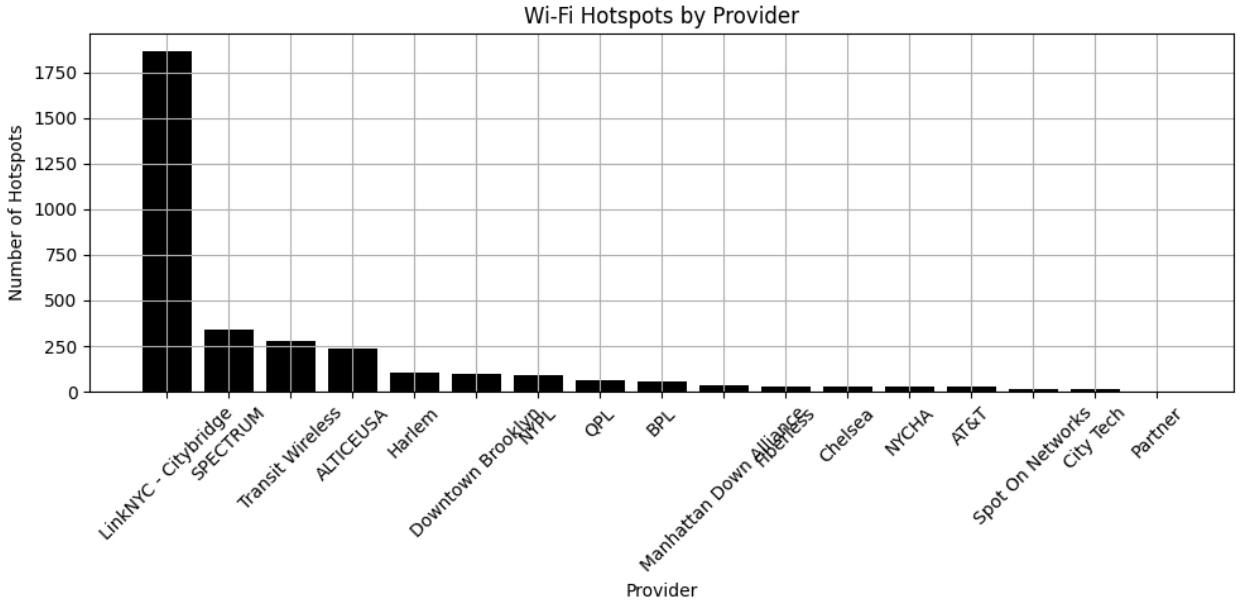

```
# Scatter plot to explore relation between Latitude and Longitude
plt.figure(figsize=(10, 6))
plt.scatter(df['Latitude'], df['Longitude'], color='red', alpha=0.6)
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.title("NYC Wi-Fi Locations (Raw Data)")
plt.show()
```



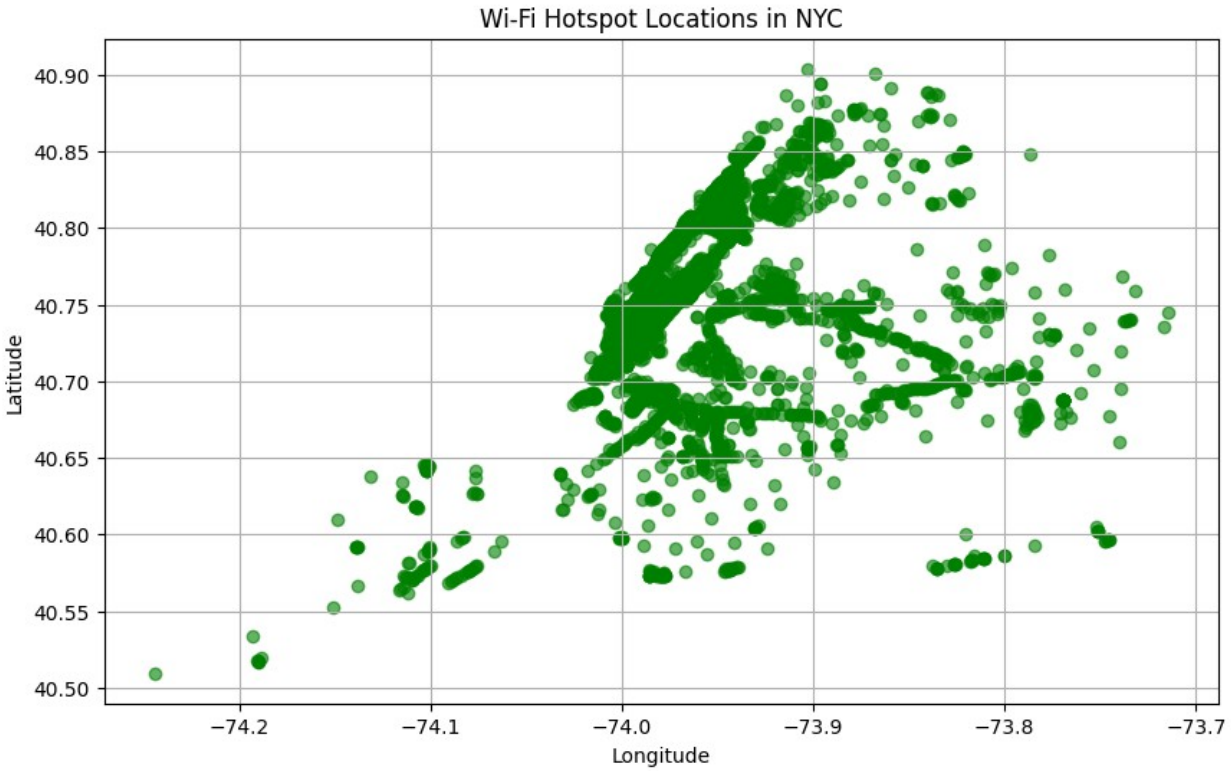

```
# Advanced filter: remove extreme locations that are outside expected
NYC bounds
filtered_df = df[
    (df['Latitude'].between(40.4, 40.9)) &
    (df['Longitude'].between(-74.3, -73.7))
]

# Scatter plot after removing geographic outliers
plt.figure(figsize=(10, 6))
plt.scatter(filtered_df['Latitude'], filtered_df['Longitude'],
color='green', alpha=0.6)
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.title("NYC Wi-Fi Locations (Filtered)")
plt.show()
```



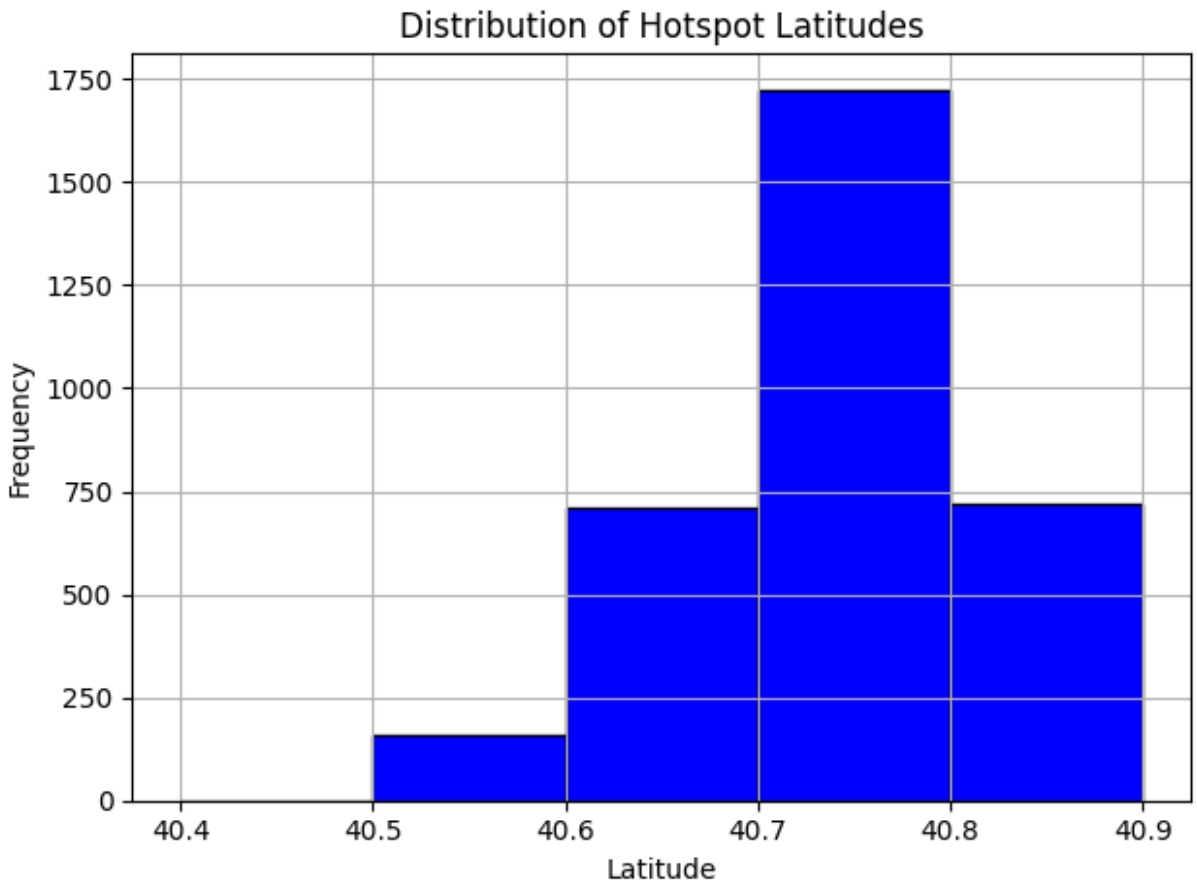

```
# ----------------- BAR CHART ----------------------
# Count of Hotspots by Provider
provider_counts = df['Provider'].value_counts()
x = provider_counts.index
y = provider_counts.values

plt.figure(figsize=(10, 5))
plt.bar(x, y, color="black")
plt.xlabel("Provider")
plt.ylabel("Number of Hotspots")
plt.title("Wi-Fi Hotspots by Provider")
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```



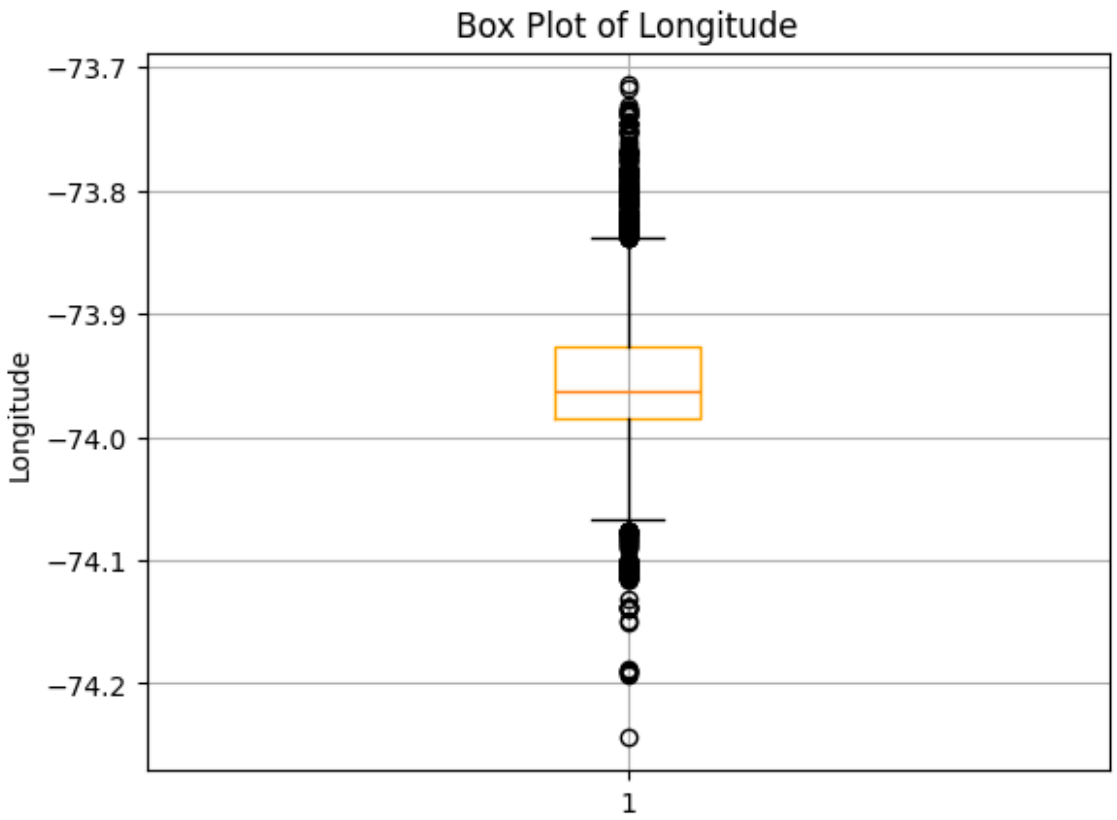

```
# ----------------- SCATTER PLOT ----------------------
# Latitude vs Longitude to visualize hotspot distribution
x = df['Longitude'].dropna()
y = df['Latitude'].dropna()

plt.figure(figsize=(10, 6))
plt.scatter(x, y, color="green", alpha=0.6)
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.title("Wi-Fi Hotspot Locations in NYC")
plt.grid(True)
plt.show()
```



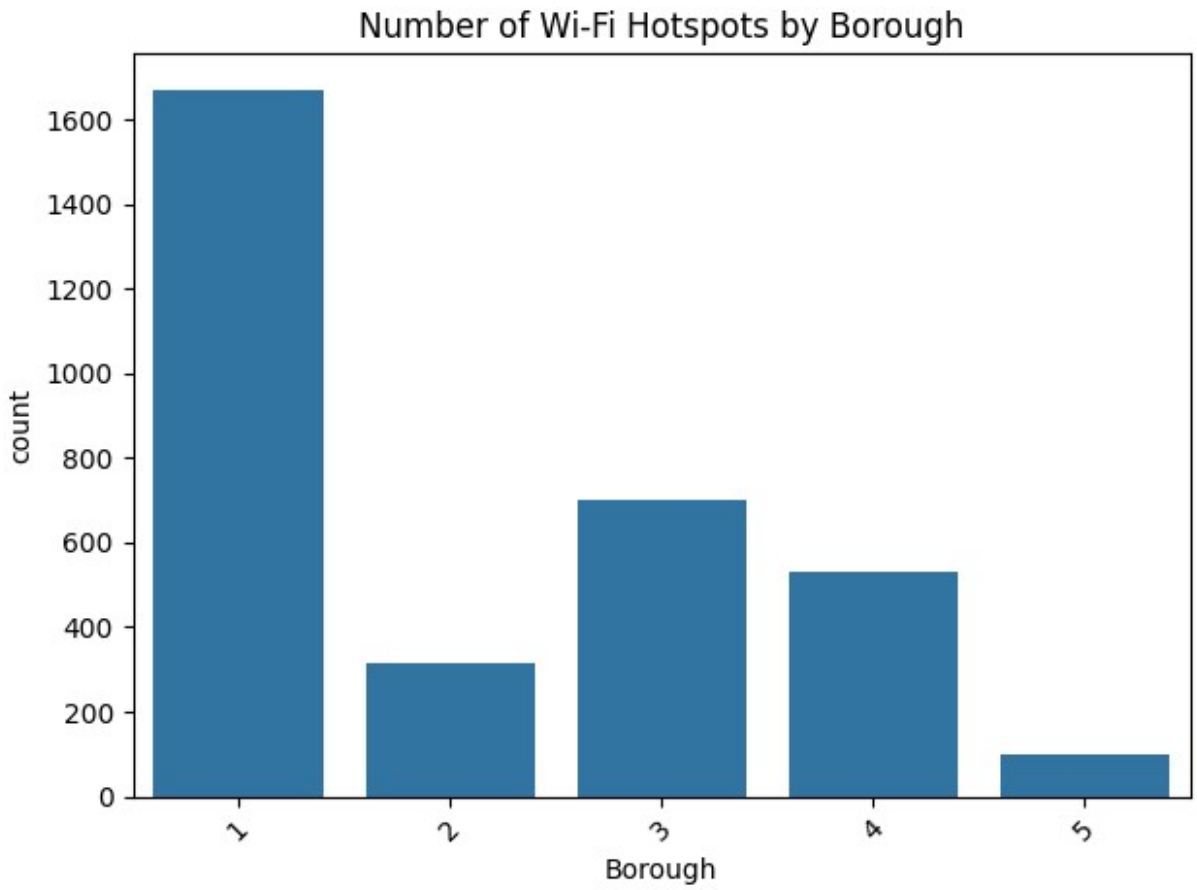

```
# ----------------- HISTOGRAM ----------------------
# Histogram of Latitude
latitudes = df['Latitude'].dropna()

plt.hist(latitudes, bins=[40.4, 40.5, 40.6, 40.7, 40.8, 40.9],
color="blue", edgecolor="black")
plt.xlabel("Latitude")
plt.ylabel("Frequency")
plt.title("Distribution of Hotspot Latitudes")
plt.grid(True)
plt.xticks([40.4, 40.5, 40.6, 40.7, 40.8, 40.9])
plt.tight_layout()
plt.show()
```




```
# ----------------- BOX PLOT ----------------------
# Box plot of Longitude values
longitudes = df['Longitude'].dropna()

plt.boxplot(longitudes, boxprops=dict(color="orange"))
plt.title("Box Plot of Longitude")
plt.ylabel("Longitude")
plt.grid(True)
plt.show()
```




```
# Barplot: Average number of hotspots per Borough
# We'll count the number of records per borough and plot that
sns.countplot(x="Borough", data=df)
plt.title("Number of Wi-Fi Hotspots by Borough")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# Let's say we want to analyze Borough-wise distribution of hotspots
print("Number of Hotspots by Borough:\n",
df['Borough'].value_counts())

# Example: Average Latitude and Longitude per Borough
print(df.groupby('Borough')[['Latitude', 'Longitude']].mean())

# General Statistics (for numeric fields)
print(df.describe())
```

```
Number of Hotspots by Borough:
 Borough
1    1672
3     700
4     531
2     316
5     100
Name: count, dtype: int64
          Latitude  Longitude
Borough
1        40.768318 -73.974071
2        40.840512 -73.895770
```

```
3        40.670915 -73.965008
4        40.721014 -73.857519
5        40.592685 -74.108638
            OBJECTID      Borough     Latitude    Longitude
X  \
count   3319.000000  3319.000000  3319.000000  3319.000000
3.319000e+03
mean   11279.197349     2.117505    40.741789   -73.950112
9.980701e+05
std      967.314772     1.269830     0.066799     0.062109
1.722035e+04
min     9601.000000     1.000000    40.509531   -74.244107
9.163706e+05
25%    10445.500000     1.000000    40.695480   -73.985788
9.881911e+05
50%    11286.000000     1.000000    40.745906   -73.962685
9.945985e+05
75%    12116.500000     3.000000    40.791719   -73.926875
1.004498e+06
max    12946.000000     5.000000    40.903723   -73.714838
1.063266e+06

                     Y     BoroCode  Council Distrcit      Postcode  \
count    3319.000000  3319.000000       3319.000000   3319.000000
mean   209540.439778     2.117505         17.210003  10535.245556
std     24334.633844     1.269830         14.845808    583.464313
min    125007.163094     1.000000          1.000000  10001.000000
25%    192679.650451     1.000000          4.000000  10022.000000
50%    211036.902451     1.000000          9.000000  10075.000000
75%    227723.332824     3.000000         32.000000  11211.000000
max    268543.647017     5.000000         51.000000  11694.000000

             BoroCD   Census Tract      BCTCB2010           BIN
BBL  \
count  3319.000000    3319.000000    3319.000000  3.319000e+03
3.319000e+03
mean    218.880988    2353.715276    2353.715276  1.581973e+06
1.751089e+09
std     128.098784   12476.228592   12476.228592  1.453220e+06
1.415620e+09
min     101.000000       1.000000       1.000000  0.000000e+00
0.000000e+00
25%     106.000000      76.000000      76.000000  0.000000e+00
1.007220e+09
50%     112.000000     163.000000     163.000000  1.056626e+06
1.017260e+09
75%     309.000000     271.000000     271.000000  3.018582e+06
3.013530e+09
max     595.000000  157902.000000  157902.000000  5.162377e+06
```

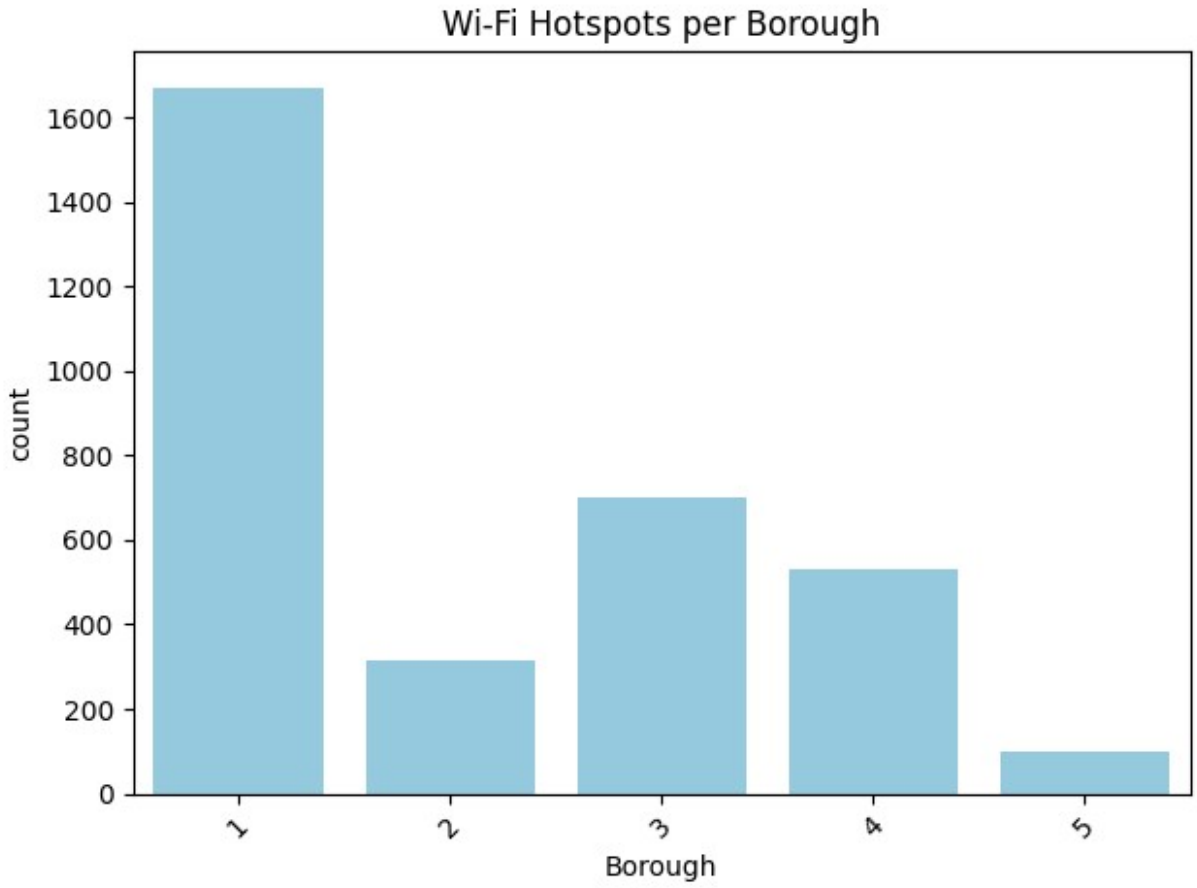
```
5.078990e+09
```

```
          DOITT_ID
count  3319.000000
mean   2679.822537
std    1598.918823
min       1.000000
25%    1099.500000
50%    3265.000000
75%    4094.500000
max    4950.000000
```

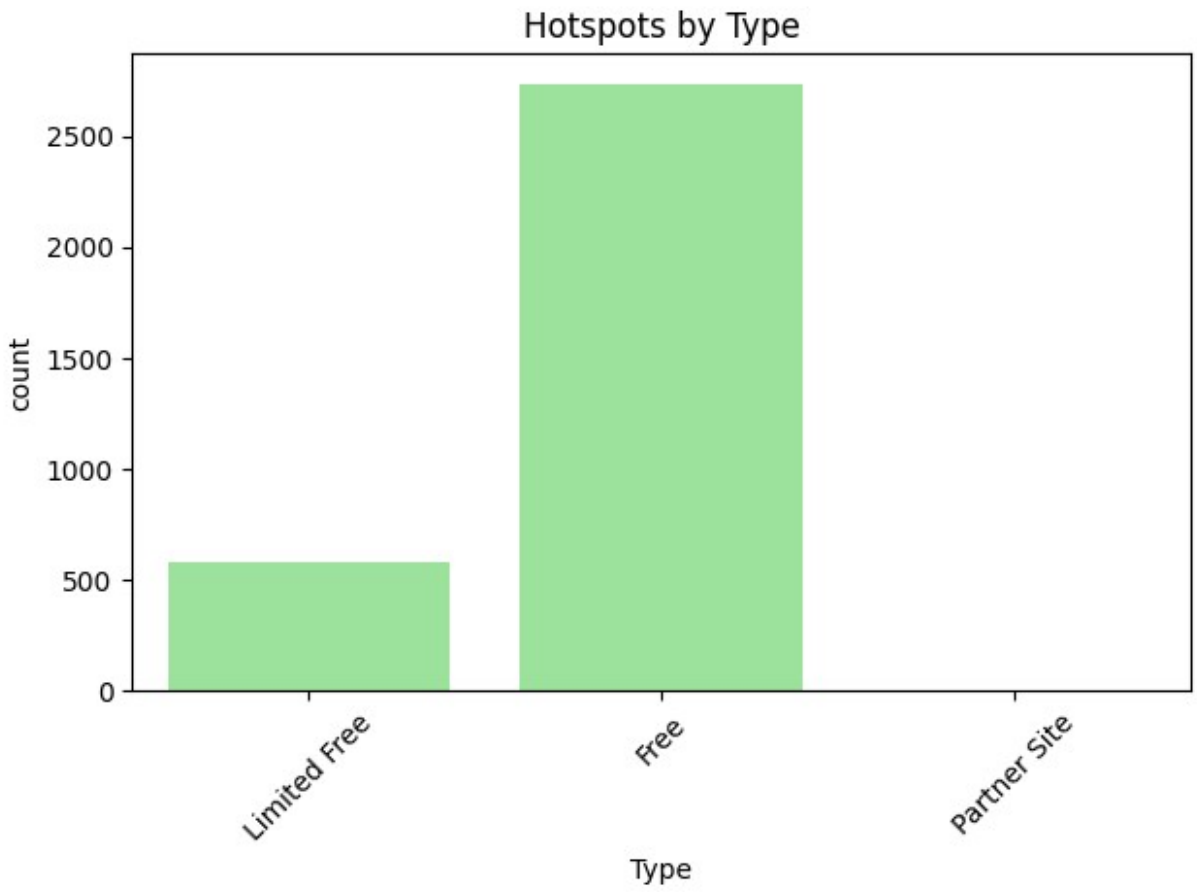
```
# Aggregate Functions for Latitude and Longitude (as sample numerics)
print(df.agg({"Latitude": ['max', 'min', 'mean', 'median'],
              "Longitude": ['max', 'min', 'mean', 'median']}))

# Filtering: Example - filter hotspots in Manhattan
df_manhattan = df[df['Borough'] == 'Manhattan']
print("Total hotspots in Manhattan:", len(df_manhattan))

# Grouping by Provider
print(df.groupby("Provider")['SSID'].count())
```

```
          Latitude  Longitude
max     40.903723 -73.714838
min     40.509531 -74.244107
mean    40.741789 -73.950112
median  40.745906 -73.962685
Total hotspots in Manhattan: 0
Provider
ALTICEUSA                   237
AT&T                         27
BPL                          59
Chelsea                      30
City Tech                    11
Downtown Brooklyn           100
Fiberless                    30
Harlem                      101
LinkNYC - Citybridge       1868
Manhattan Down Alliance      36
NYCHA                        28
NYPL                         90
Partner                       2
QPL                          65
SPECTRUM                    343
Spot On Networks             16
Transit Wireless            276
Name: SSID, dtype: int64
```

```
# Visualization: Count of hotspots per Borough
sns.countplot(data=df, x='Borough', color='skyblue')
plt.title('Wi-Fi Hotspots per Borough')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# Visualization: Distribution of Hotspots by Type
sns.countplot(data=df, x='Type', color='lightgreen')
plt.title('Hotspots by Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

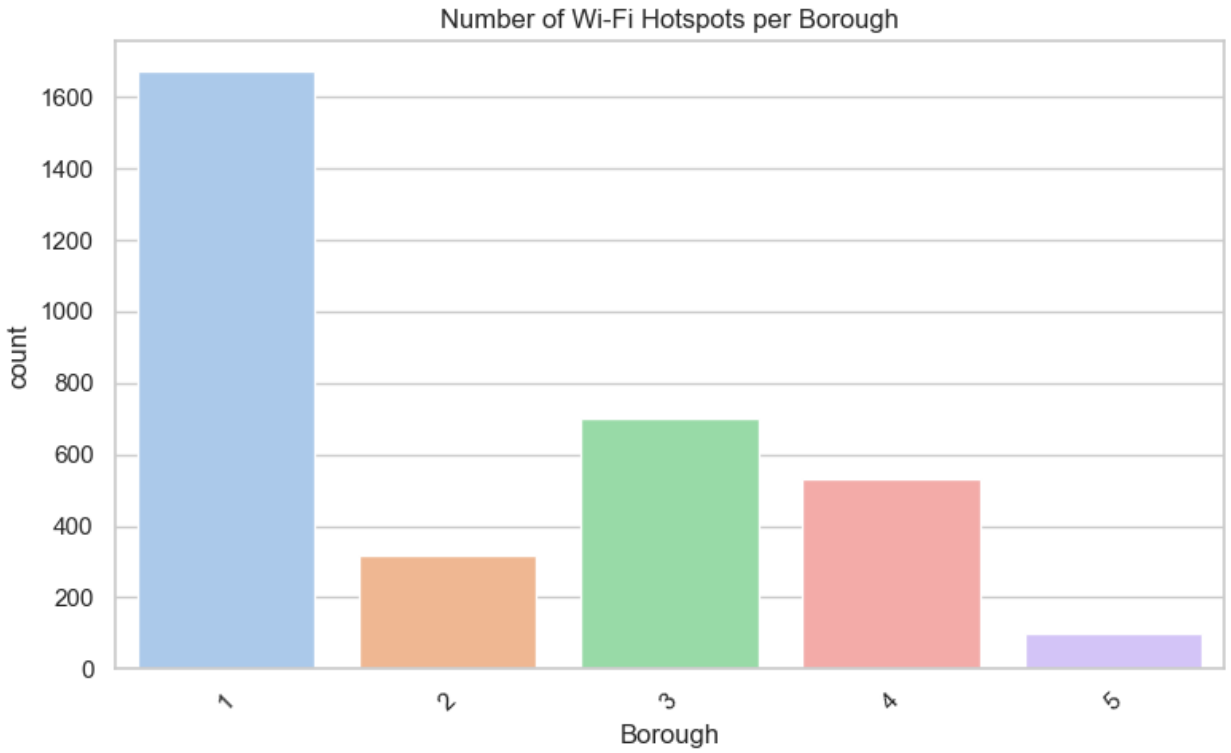



```
import scipy.stats as stats
# Drop rows with missing Latitude values
latitudes = df['Latitude'].dropna()

# ------------------------------------------------
# One-Sample T-Test
# Hypotheses:
# H0 (Null Hypothesis): The average latitude of all hotspots is equal
to 40.75
# H1 (Alternative Hypothesis): The average latitude of all hotspots is
not equal to 40.75
# ------------------------------------------------
test_statistic, p_value = stats.ttest_1samp(latitudes, 40.75)
print("One-Sample T-Test:")
print("Test Statistic:", test_statistic)
print("P-Value:", p_value)

# Sample mean
SM = np.mean(latitudes)
print("Sample Mean Latitude:", SM)

# Decision based on p-value
```

```
if p_value < 0.05:
    print("Null Hypothesis Rejected: The mean latitude is
significantly different from 40.75")
else:
    print("Null Hypothesis Accepted: No significant difference from
40.75")


One-Sample T-Test:
Test Statistic: -7.081780107723254
P-Value: 1.729662863190324e-12
Sample Mean Latitude: 40.741788815192045
Null Hypothesis Rejected: The mean latitude is significantly different
from 40.75

#****************************************************************
**
# ------------------------------------------------
# Two-Sample T-Test: Compare Manhattan vs Brooklyn
# Hypotheses:
# H0 (Null Hypothesis): The average latitude of hotspots in Manhattan
is equal to that in Brooklyn
# H1 (Alternative Hypothesis): The average latitude of hotspots in
Manhattan is not equal to that in Brooklyn
# ------------------------------------------------
lat_manhattan = df[df['Borough'] == 'Manhattan']['Latitude'].dropna()
lat_brooklyn = df[df['Borough'] == 'Brooklyn']['Latitude'].dropna()

SM1 = np.mean(lat_manhattan)
SM2 = np.mean(lat_brooklyn)

test_stat, p_val = stats.ttest_ind(lat_manhattan, lat_brooklyn)
print("\nTwo-Sample T-Test (Manhattan vs Brooklyn Latitudes):")
print("Test Statistic:", test_stat)
print("P-Value:", p_val)
print("Mean Latitude (Manhattan):", SM1)
print("Mean Latitude (Brooklyn):", SM2)

if p_val < 0.05:
    print("Null Hypothesis Rejected: Latitude differs significantly
between Manhattan and Brooklyn")
else:
    print("Null Hypothesis Accepted: No significant difference in
latitude between Manhattan and Brooklyn")


Two-Sample T-Test (Manhattan vs Brooklyn Latitudes):
Test Statistic: nan
P-Value: nan
Mean Latitude (Manhattan): nan
```

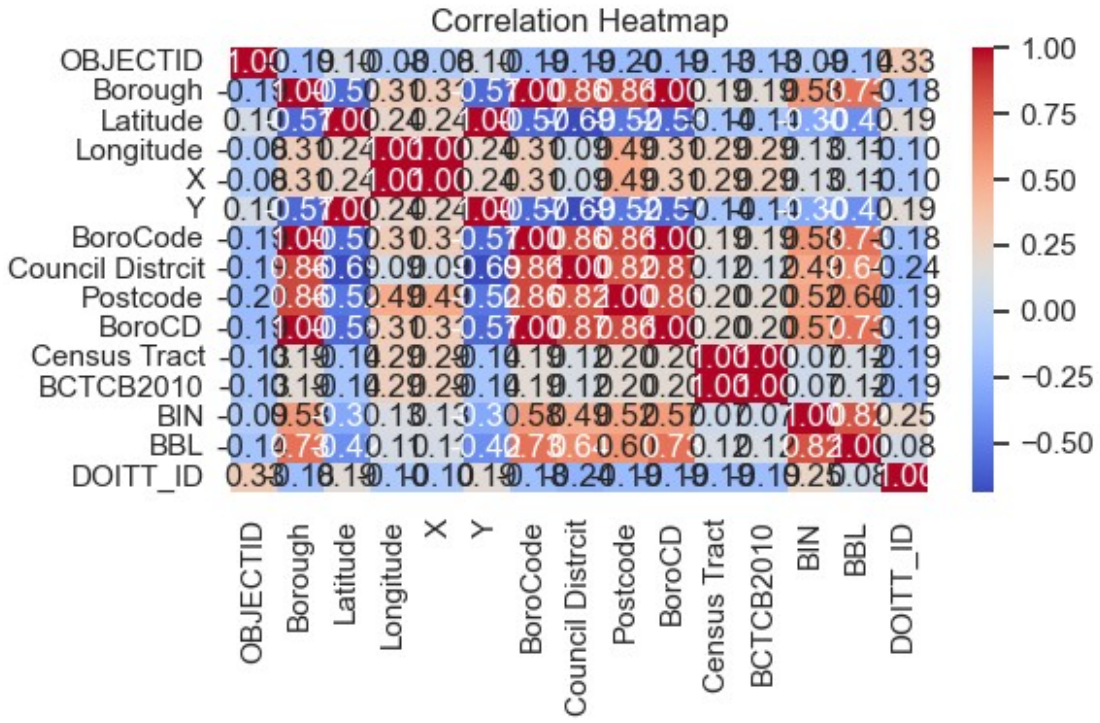
```
Mean Latitude (Brooklyn): nan
Null Hypothesis Accepted: No significant difference in latitude
between Manhattan and Brooklyn
```

```
# Set style
sns.set(style="whitegrid")

# ------------------------
# 1. Count Plot - Hotspots per Borough
# ------------------------
plt.figure(figsize=(8, 5))
sns.countplot(x='Borough', data=df, palette='pastel')
plt.title('Number of Wi-Fi Hotspots per Borough')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

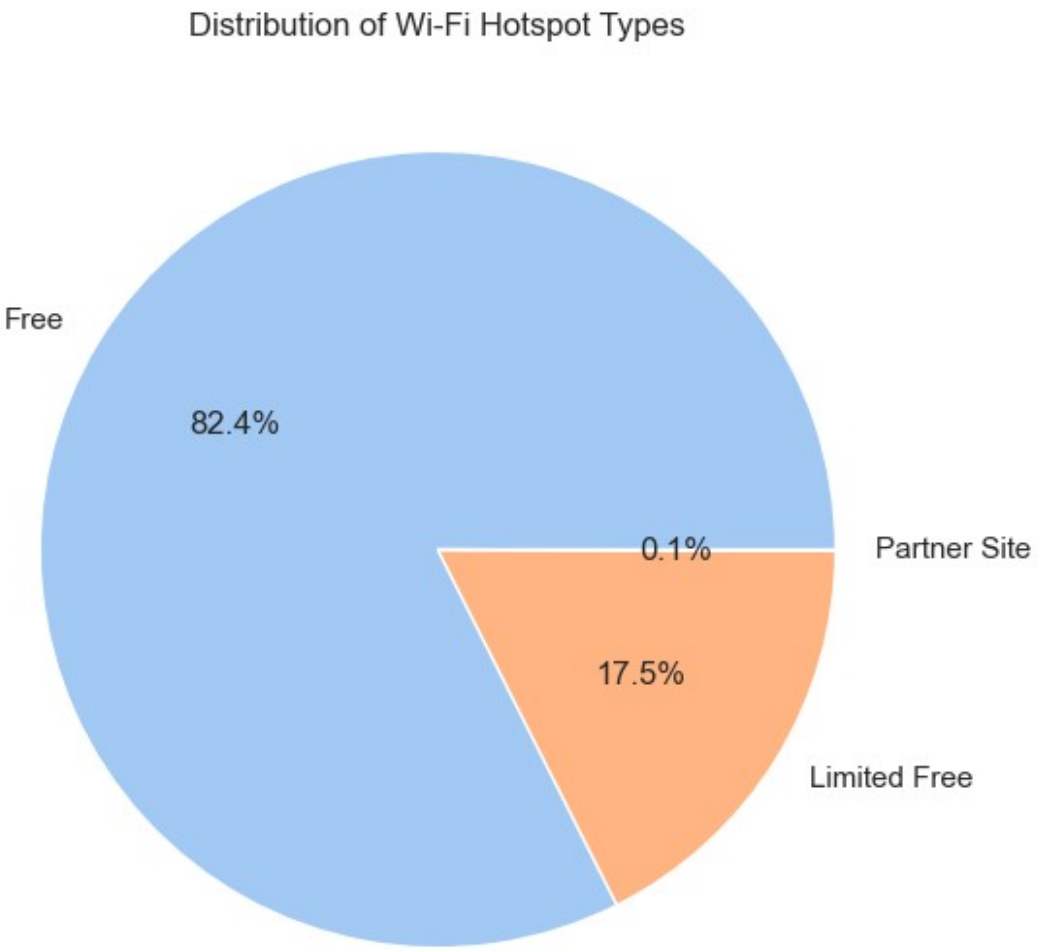



```
# ------------------------
# 2. Heatmap - Correlation Matrix
# ------------------------
# Select numeric columns only
numeric_df = df.select_dtypes(include=['float64', 'int64'])

plt.figure(figsize=(6, 4))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
```

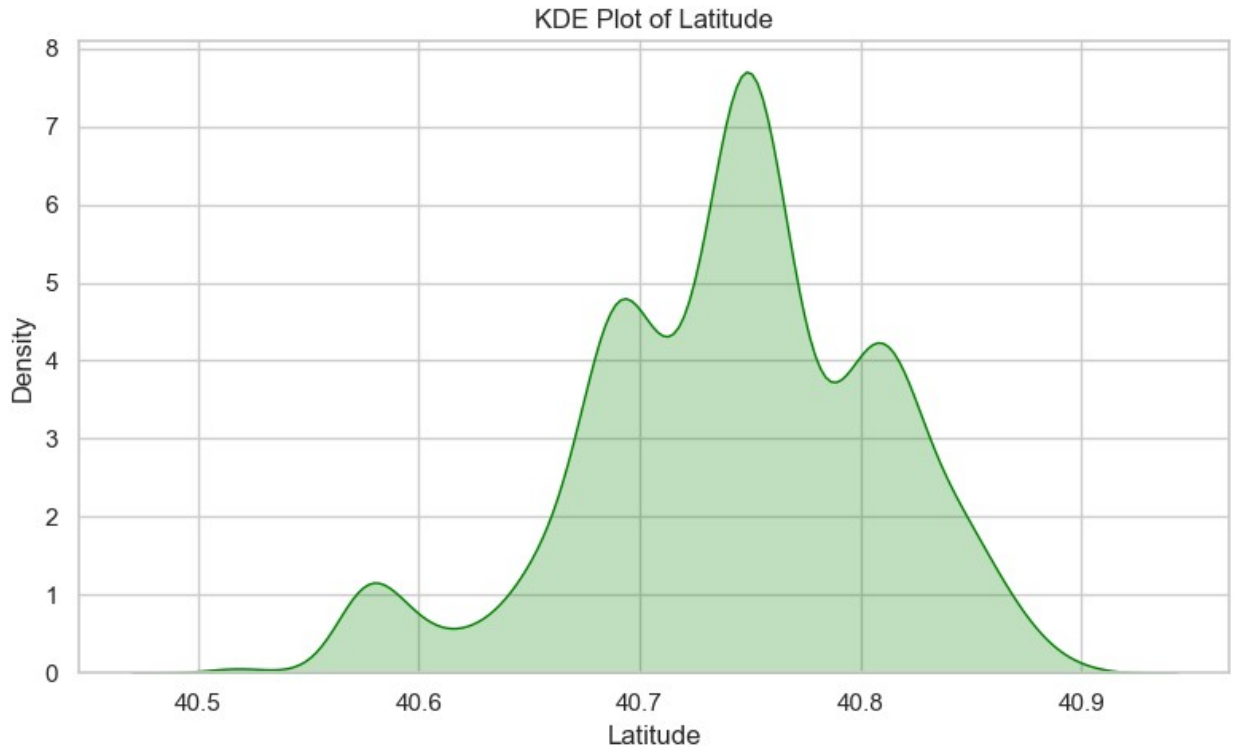
```
plt.tight_layout()
plt.show()
```




```
# 3. Pie Chart - Hotspot Type Distribution
# ------------------------
plt.figure(figsize=(6, 6))
type_counts = df['Type'].value_counts()
plt.pie(type_counts, labels=type_counts.index, autopct='%1.1f%%',
colors=sns.color_palette('pastel'))
plt.title("Distribution of Wi-Fi Hotspot Types")
plt.tight_layout()
plt.show()
```




```
# 4. KDE Plot - Latitude Distribution
# ------------------------
plt.figure(figsize=(8, 5))
sns.kdeplot(df['Latitude'].dropna(), shade=True, color='green')
plt.title("KDE Plot of Latitude")
plt.xlabel("Latitude")
plt.tight_layout()
plt.show()
```


KDE Plot of Latitude
Density
Latitude
0
1
2
3
4
5
6
7
8
40.5
40.6
40.7
40.8
40.9