# HC7 Data Exploration an Cleaning

YOUR TURN!

Now that you know how to explore the data, clean the data, obtain statistics about the data, visualize the data and select a subset of the data based on the value in a particular column (e.g. neighbourhood_group == 'Staten Island"), think about how you want to explore the data for your analysis:

1. As a group, think about an overall data-driven discussion of your borough and how it compares to the others.
2. Individually, analyze the data in your borough and compare to the data for other boroughs.

As you explore your data, keep in mind your analysis and findings from HC2 and HC3 and see if you can make any connections, or if you find that the data supports those findings.

# These are the three libraries that we need to import in order to properly convey our data:

---

"pandas" is used for data sets.

"matplotlib.pylot" is for plotting & arrays.

Finally, "gdown" is imported for the user to download a file from Google Drive to Python.

```
import pandas as pd
import matplotlib.pyplot as plt
import gdown
```

```
# download the file from our drive
!wget https://huntercsci127.github.io/files/clean_heat_dataset.csv
```

```
--2023-10-26 00:01:48--  https://huntercsci127.github.io/files/clean_heat_dataset.csv
Resolving huntercsci127.github.io (huntercsci127.github.io)... 185.199.108.153, 185.199.109.153, 185.199.110.153, ...
Connecting to huntercsci127.github.io (huntercsci127.github.io)|185.199.108.153|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1903535 (1.8M) [text/csv]
Saving to: ‘clean_heat_dataset.csv.2’

clean_heat_dataset. 100%[===================>]   1.81M  --.-KB/s    in 0.01s

2023-10-26 00:01:48 (150 MB/s) - ‘clean_heat_dataset.csv.2’ saved [1903535/1903535]
```

```
#list the files in the current directory to confirm the file is there
!ls
```

```
clean_heat_dataset.csv  clean_heat_dataset.csv.1  clean_heat_dataset.csv.2  sample_data
```

```
#We're having the code read in the csv into a data frame.
clean_heat = pd.read_csv("clean_heat_dataset.csv")

print("The dimension of the table is: ", clean_heat.shape)
```

```
The dimension of the table is:  (4789, 42)
```

```
print("Number of datapoints with null entry for each column:\n",clean_heat.isnull().sum())
```

```
Number of datapoints with null entry for each column:
 Borough, Block, Lot #              0
Street Address                    0
Postcode                          4
Borough                           0
Utility                           0
Building Manager                  1
Owner                             1
Owner Address                     1
Owner Telephone                 448
DEP Boiler Application #          0
#6 Deadline                       0
Boiler Model                      6
```

```
# of Identical Boilers            1
Boiler Capacity (Gross  BTU)      0
Boiler Installation Date          1
Boiler Age Range                  1
Est. Retirement Year              1
Burner Model                      6
Primary Fuel                      0
Total Gallons (High)              0
Total Gallons (Low)               0
Total MMBTU (High)                0
Total MMBTU (low)                 0
Greener Greater Buildings         1
GGB Deadline                   1979
Building Type                     1
Council District                  1
Community Board                   0
Bldg Sqft                         1
# of Bldgs                        1
# of Floors                       1
# of Res. Units                   1
Total Units                       1
Year Built                        1
Condo?                         4603
Coop?                          3979
Latitude                         26
Longitude                        26
Census Tract                     26
BIN                              33
BBL                              33
NTA                              26
dtype: int64
```

```
clean_heat['BIN'].value_counts()
```

```
4455390.0    10
1000000.0    10
4455441.0     9
4442362.0     7
4433296.0     6
             ..
1055061.0     1
1055117.0     1
1055119.0     1
1055157.0     1
5015146.0     1
Name: BIN, Length: 4392, dtype: int64
```

```
columns= ['Borough, Block, Lot #', 'Street Address', 'Postcode', 'Borough', 'Utility', 'Building Manager', 'Owner', 'Owner Addres
cat_columns= ['Street Address', 'Borough', 'Utility', 'Building Manager', 'Owner','Owner Address','Owner Telephone', 'DEP Boiler 
num_columns=['Borough, Block, Lot #','Postcode','#6 Deadline','# of Identical Boilers','Boiler Capacity (Gross  BTU)', 'Boiler In
```

```
clean_heat[num_columns]=clean_heat[num_columns].fillna(value=0)
```

```
clean_heat[cat_columns]=clean_heat[cat_columns].fillna(value="")
```

## Now, we will dive deeper with our data!

We are going to further analyze the data and manipulate it in order to convey results pertaining to the borough of the Bronx.

```
# General data exploration
print(clean_heat.head())  # Showing the first few rows to understand the data structure
print(clean_heat.describe())  # Summary statistics for the dataset
print(clean_heat.isnull().sum())  # Checking for missing values
```

```
   Borough, Block, Lot #       Street Address  Postcode    Borough  \
0             1008120001   155 WEST 36 STREET   10018.0  Manhattan
1             1008340048         330 5 AVENUE   10001.0  Manhattan
2             1008390009    49 WEST 37 STREET   10018.0  Manhattan
3             1008670001         411 5 AVENUE   10016.0  Manhattan
4             1022420029  639 WEST 207 STREET   10034.0  Manhattan

      Utility                           Building Manager  \
0  Con Edison                 485 7 AVE.ASSOC./COLLIERS
1  Con Edison  SKYLER 330 LLCC/O SHULSKY PROPERTIES INC.
2  Con Edison                        49 W 37 ST REALTY CO
```

```
3  Con Edison                      ADMS& CO. REAL ESTATE
4  Con Edison                              WEINER REALTY

                            Owner                              Owner Address  \
0                     485 SHUR LLC       485 7 AVENUE#777, MANHATTAN NY 10018
1          SHULSKY PROPERTIES INC.                  307 FITH AVE, NY NY 10016
2          49 W 37TH ST REALTY CO  440 PARK AVENUE SOUTH, MANHATTAN NY 10016
3     ADAMS & CO. LLC/FRED LIGUORI           411 5 AVENUE, MANHATTAN NY 10016
4  PINNACLE WASHINGTON HEIGHTS LLC          P.O.BO. 1920, NEW YORK NY 10116

  Owner Telephone DEP Boiler Application #  ...  Total Units Year Built  \
0    212-971-4000                CA160181H  ...         70.0     1906.0
1    212 984-8370                CA323565K  ...         62.0     1926.0
2    212 685-6400                CA145582N  ...         24.0     1925.0
3    212-679-5500                CA417870Y  ...          1.0     1915.0
4                                CA068682Y  ...         58.0     1925.0

   Condo?  Coop?   Latitude  Longitude  Census Tract        BIN           BBL  \
0                 40.751828 -73.988595         109.0  1015235.0  1.008128e+09
1                 40.747521 -73.985239          76.0  1015853.0  1.008340e+09
2                 40.751018 -73.984758          84.0  1015958.0  1.008390e+09
3                 40.750430 -73.983098          82.0  1017191.0  1.008670e+09
4                 40.868598 -73.921780         303.0  1064990.0  1.022420e+09

                   NTA
0  Midtown-Midtown South
1  Midtown-Midtown South
2  Midtown-Midtown South
3   Murray Hill-Kips Bay
4     Marble Hill-Inwood

[5 rows x 42 columns]
       Borough, Block, Lot #     Postcode  #6 Deadline  \
count           4.789000e+03   4789.00000  4789.000000
mean            1.746012e+09  10332.17227   529.380455
std             9.941284e+08    539.63210   886.518517
min             1.000160e+09      0.00000     0.000000
25%             1.012590e+09  10023.00000     0.000000
50%             1.021790e+09  10040.00000     0.000000
75%             2.033480e+09  10467.00000  2012.000000
max             5.005900e+09  11435.00000  2015.000000

       # of Identical Boilers  Boiler Capacity (Gross  BTU)  \
count             4789.000000                  4.789000e+03
mean                 1.101065                  1.054811e+03
std                  0.330528                  7.255502e+04
```

```
# Analyzing the Bronx
bronx_data = clean_heat[clean_heat['Borough'] == 'Bronx']
print(bronx_data.describe())  # Summary statistics for the Bronx
```

```
std             8.665177e+06      6.316834   826.520302
min             2.022650e+09  10451.000000     0.000000
25%             2.029478e+09  10457.000000     0.000000
50%             2.032765e+09  10462.000000     0.000000
75%             2.039143e+09  10467.000000     0.000000
max             2.059580e+09  10474.000000  2015.000000

       # of Identical Boilers  Boiler Capacity (Gross  BTU)  \
count             1498.000000                   1498.000000
mean                 1.042724                      4.614559
std                  0.211976                      3.790257
min                  1.000000                      0.000000
25%                  1.000000                      2.945000
50%                  1.000000                      4.100000
75%                  1.000000                      5.250000
max                  3.000000                    105.000000

       Boiler Installation Date  Est. Retirement Year  Total Gallons (High)  \
count               1498.000000           1498.000000          1.498000e+03
mean                1988.685581           2023.997997          1.220819e+05
std                    8.875179              8.219638          1.282715e+06
min                 1955.000000           2010.000000          0.000000e+00
25%                 1984.000000           2019.000000          2.752100e+04
50%                 1987.000000           2022.000000          3.865350e+04
75%                 1994.000000           2029.000000          5.469150e+04
max                 2009.000000           2044.000000          3.477172e+07

       Total Gallons (Low)  Total MMBTU (High)  ...   # of Bldgs  # of Floors  \
count         1.498000e+03         1498.000000  ...  1498.000000  1498.000000
```

```
25%          1.926500e+04        2812.565000  ...     1.000000     5.000000
50%          2.705750e+04        3906.875000  ...     1.000000     6.000000
75%          3.828450e+04        5183.457500  ...     1.000000     6.000000
max          2.434021e+07       75196.800000  ...    14.000000    30.000000

       # of Res. Units  Total Units   Year Built     Latitude    Longitude  \
count      1498.000000  1498.000000  1498.000000  1498.000000  1498.000000
mean         60.360481    61.293057  1929.297063    40.612645   -73.446620
std          49.078654    49.049580    87.658324     3.158566     5.712077
min           0.000000     0.000000     0.000000     0.000000   -73.931257
25%          39.000000    39.000000  1925.000000    40.841243   -73.906553
50%          52.000000    53.000000  1928.000000    40.858436   -73.895353
75%          67.000000    67.000000  1939.000000    40.872923   -73.874754
max         462.000000   462.000000  2007.000000    40.912869     0.000000

       Census Tract           BIN           BBL
count   1498.000000  1.498000e+03  1.498000e+03
mean    9727.348465  2.013303e+06  2.020252e+09
std    14535.237012  1.753610e+05  1.740315e+08
min        0.000000  0.000000e+00  0.000000e+00
25%      251.000000  2.010811e+06  2.029268e+09
50%      394.000000  2.016588e+06  2.032710e+09
75%    22402.500000  2.045594e+06  2.039048e+09
max    45102.000000  2.128625e+06  2.059580e+09

[8 rows x 24 columns]
```

```
# Average building age in the Bronx
average_age_bronx = bronx_data['Year Built'].mean()
print("Average building age in the Bronx:", average_age_bronx.round())
```

```
Average building age in the Bronx: 1929.0
```

We notice that the "**average building age**" of most buildings in the Bronx is **1929**. This is a huge contribution to the health of the children and elderly living in certain parts like Melrose or Mott Haven, since they are at increased chances of getting asthma. The materials used in the buildings are also extremely old, and most likely spread cancer causing or gaseous air throughout the borough.

```
# Building Type distribution in the Bronx
bronx_building_type = bronx_data['Building Type'].value_counts(normalize=True)
print("Building Type distribution in the Bronx:")
print(bronx_building_type)
```

```
Building Type distribution in the Bronx:
Elevator Apartments               0.530040
Walk-Up Apartments                0.425234
Educational Structures            0.012016
Churches, Synagogues, etc.        0.007343
Factory & Industrial Buildings    0.005340
Store Buildings                   0.004005
Condominiums                      0.004005
Office Buildings                  0.003338
Hospitals & Health                0.003338
Warehouses                        0.002670
Vacant Land                       0.001335
Hotels                            0.000668
Asylums & Homes                   0.000668
Name: Building Type, dtype: float64
```

Concerningly, the reader could notice that *"Hospitals and Health"* are towards the ***bottom*** of the list. This may explain why so many people have health issue within the Bronx that go untreated, since there are no nearby hospitals. There are also an **alarming number of Factory & Industrial buildings**, which we can note that there is an increased risk of c02 emissions since there is a large amount of them.

```
# Average boiler installation date in the Bronx
average_boiler_age_bronx = bronx_data['Boiler Installation Date'].mean()
print("Average boiler installation date in the Bronx:", average_boiler_age_bronx.round())
```

```
Average boiler installation date in the Bronx: 1989.0
```

```
# Average MMBTU totals in the Bronx
average_mmbtu_bronx = bronx_data['Total MMBTU (low)'].mean()
print("Average MMBTU totals in the Bronx:", average_mmbtu_bronx)
```

```
Average MMBTU totals in the Bronx: 6939.6464753004
```

```
# Average boiler capacity in the Bronx
average_boiler_capacity_bronx = bronx_data['Boiler Capacity (Gross  BTU)'].mean()
print("Average boiler capacity in the Bronx:", average_boiler_capacity_bronx)
```

```
Average boiler capacity in the Bronx: 4.614559412550067
```

```
# Distribution of primary fuel used in the Bronx
primary_fuel = bronx_data['Primary Fuel'].value_counts(normalize=True)
print("Primary fuel type distribution in the Bronx:")
print(primary_fuel)
```

```
Primary fuel type distribution in the Bronx:
#4    0.783044
#6    0.216956
Name: Primary Fuel, dtype: float64
```

Number 4 and 6 fuels are derived from petroleum, and are used all throughout heating system engines in the borough. With number 4 being the oil of higher usage, it greatly contributes to the ongoing chemical and air pollution within the Bronx. As shown in our groups HC3, "In NYC, these oils were identified as significant contributors to pollution, being responsible for 86% of soot pollution despite being used in only 1% of buildings." This thereby elucidates the fact that oils 4 & 6 are heavy contributers of pollution, and they are both used.

```
# Distribution of burner models in the Bronx
burner_model = bronx_data['Burner Model'].value_counts(normalize=True)
print("Burner model distribution in the Bronx:")
print(burner_model)
```

```
Burner model distribution in the Bronx:
ICI DEG 42 P                           0.027370
ICI DEG 42P                            0.020694
ICI DEG 54 P                           0.014019
ICI MMG 42 P                           0.012684
ICI DEG42P                             0.011348
                                         ...
ICI ME-63 P                            0.000668
IND. COMB. DE-63 (P)                   0.000668
I.C. DEG-54 (P)                        0.000668
INDUSTRIAL COMBUSTION MODEL MEG-42-P   0.000668
HEVE MMG 42 (P)                        0.000668
Name: Burner Model, Length: 910, dtype: float64
```

## Now that we have given general information, we are going to manipulate it in order to show graphs based on both the borough & individual neighborhoods!

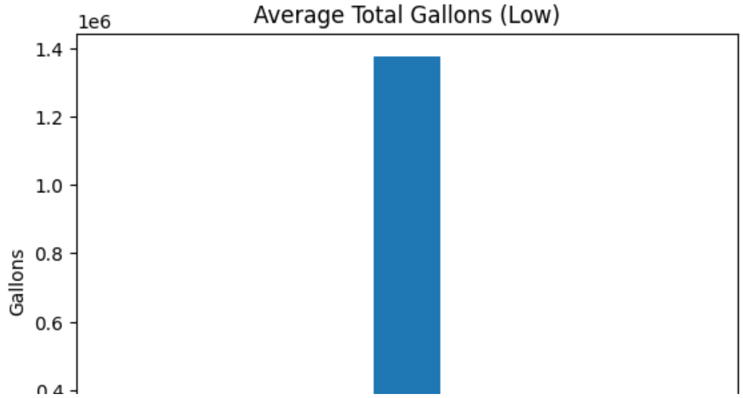
```
#Now, we will only be selecting rows pertaining to the Bronx itself and graphing it.
st = clean_heat[clean_heat['Borough'].isin(['Bronx'])]
print("Number of entries in the Bronx: ", len(st))
```

```
Number of entries in the Bronx:  1498
```

```
boro_group = clean_heat.groupby(['Borough'])
```

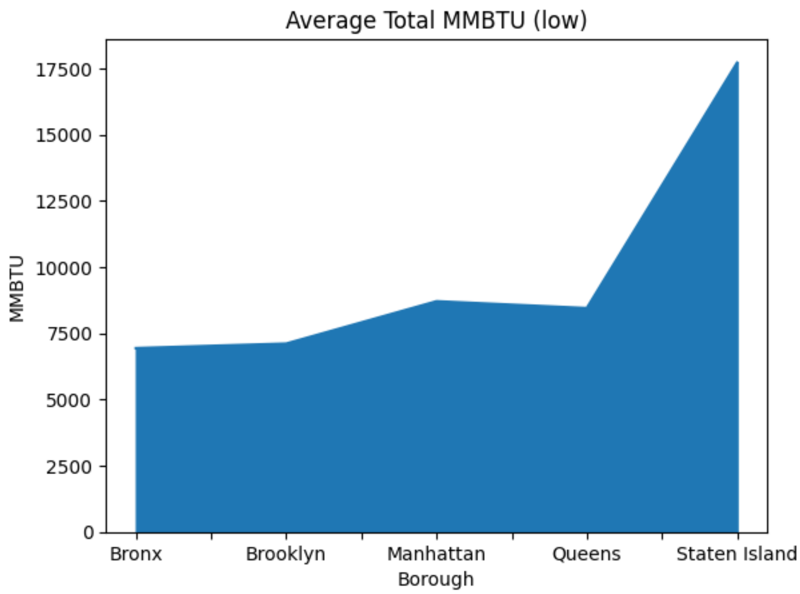
```
boro_group['Total Gallons (Low)'].mean().plot.bar()
plt.title('Average Total Gallons (Low)')
plt.xlabel('Borough')
plt.ylabel('Gallons')
```

```
Text(0, 0.5, 'Gallons')
```



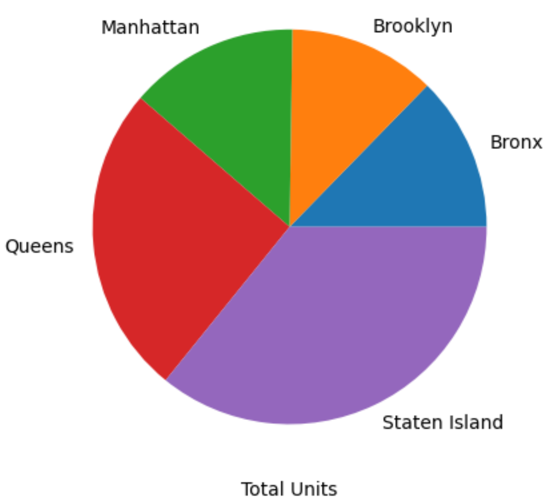

```
boro_group['Total MMBTU (low)'].mean().plot.area()
plt.title('Average Total MMBTU (low)')
plt.xlabel('Borough')
plt.ylabel('MMBTU')
```

```
Text(0, 0.5, 'MMBTU')
```



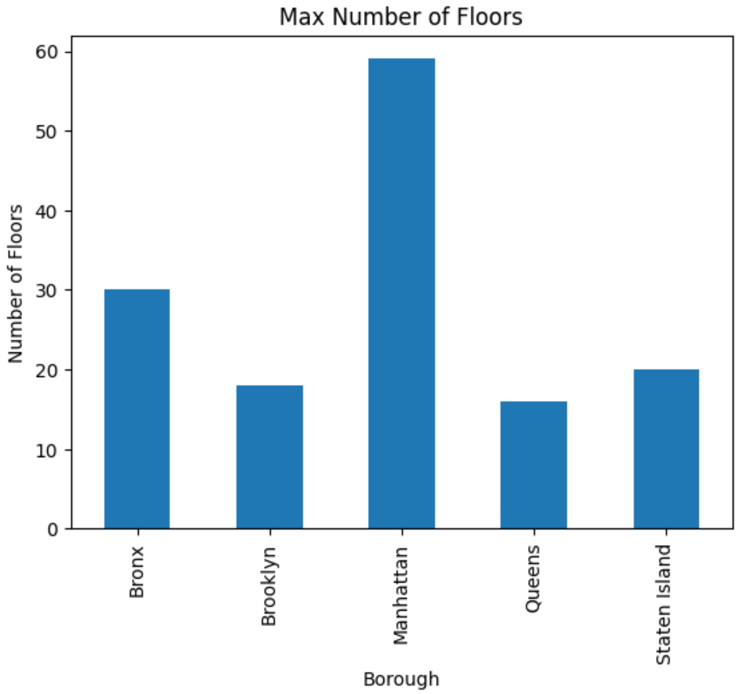

```
boro_group['Total Units'].mean().plot.pie()
plt.ylabel('')
plt.xlabel('Total Units')
```

```
Text(0.5, 0, 'Total Units')
```



```
boro_group['# of Floors'].max().plot.bar()
plt.title('Max Number of Floors')
plt.xlabel('Borough')
plt.ylabel('Number of Floors')
```

```
Text(0, 0.5, 'Number of Floors')
```




From this bar graph, we can see that Manhattan has the highest number of floors for buildings in the borough. Interestingly enough, the Bronx is second, with the max number of floors being around 30. From this, we may be able to make the assumption that both Manhattan and Bronx suffer from heavy air pollution, and this is because the buildings can contribute to pollutants being spread amongst the environment. This, combined with other emissions, according to our groups HC4 Emissions Report, contriubtes to "approximately 11% of the local fine particulate matter and 28% of the nitrogen oxide emissions.”

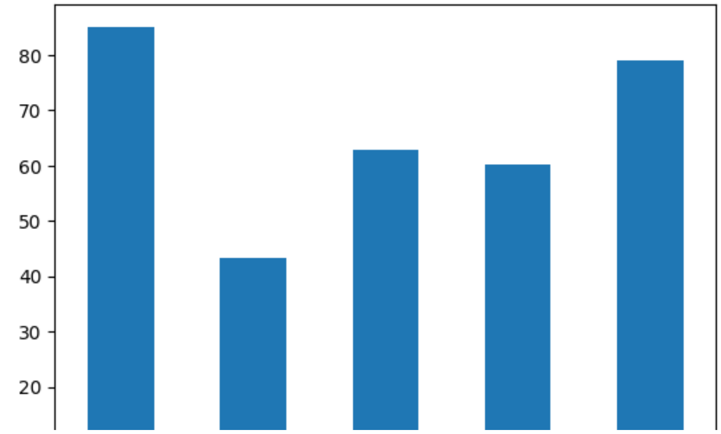
DATA FOR BRONX NEIGHBORHOODS

```
neighborhoods = ['Fordham South', 'West Farms-Bronx River', 'Bronxdale', 'Pelham Bay-Country Club-City Island', 'Westchester-Unio

# Filter the DataFrame to include only the specific neighborhoods
NTA_data = clean_heat[clean_heat['NTA'].isin(neighborhoods)]
NTA_group = NTA_data.groupby(['NTA'])
```

```
NTA_group['# of Res. Units'].mean().plot.bar()
```

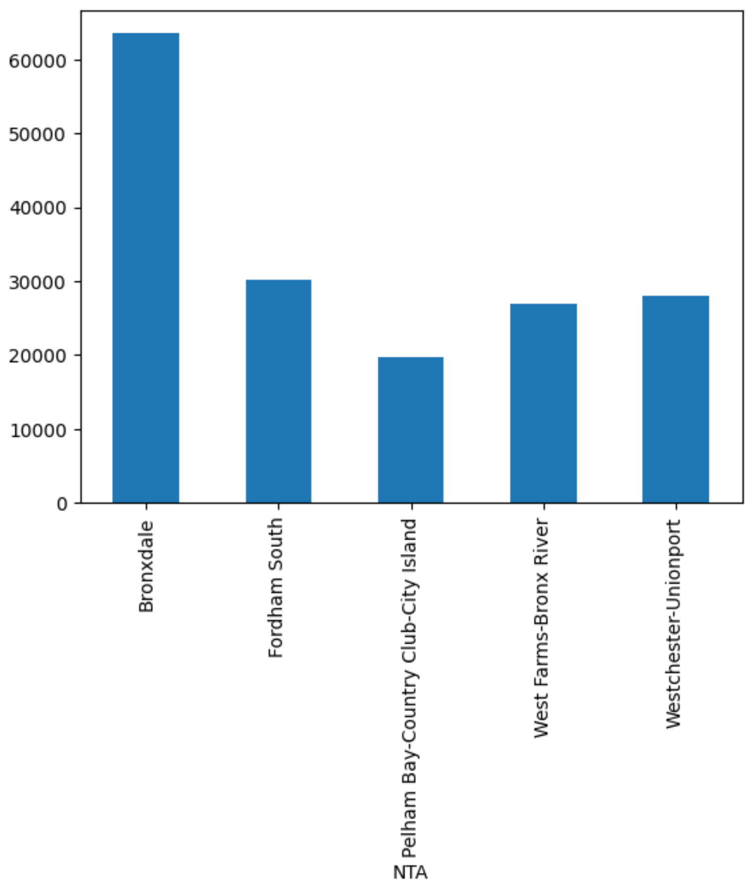
```
<Axes: xlabel='NTA'>
```




From the following bar graph, we are able to determine that Bronxdale has the most amount of housing for residents, with the number being well over 80. South Fordham, on the other hand, exhibts the lowest amount of available housing. Since South Fordham is located in the South Bronx, when compared to our findings in HC4, the South Bronx often has increasingly concerning traffic congestion. This may elucidate the idea that there are too much highways and roads, but not enough space for residents. They also contribute heavily to the c02 emissions - affecting around 17% of young children in the South Bronx neighborhoods.

```
NTA_group['Total MMBTU (low)'].max().plot.bar()
```

```
<Axes: xlabel='NTA'>
```




With Bronxdale having an alarmingly high MMBTU, it goes to show that there is most likely a high demand for things like heat and a high volume of citizens living there. However, high MMBTU contributes to varioius types of pollution such as Greenhouse Gas Emissions, land pollution (waste disposal), indoor air pollution, and chemical pollution since there are constant chemicals being released into the air. This may also contribute to the idea we discussed in HC4, where the Bronx is lagging behind the other boroughs in terms of its emissions goal. Last time we researched, it was only determined that the Bronx reached a mere 7% dec

```
NTA_group['Boiler Capacity (Gross  BTU)'].mean().plot.bar()
```

```
<Axes: xlabel='NTA'>
```