In [118…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from IPython.core.pylabtools import figsize
```

In [119…

```
pd.set_option("display.max_columns",60)
```

In [120…

```
data = pd.read_csv("F:/Dhrumil/r/Tableau/2016ny.csv")
data.head()
```

Out[120…

	Order	Property Id	Property Name	Parent Property Id	Parent Property Name	BBL - 10 digits	NYC Borough, Block and Lot (BBL) self-reported	NYC Building Identification Number (BIN)	Address 1 (self-reported)	Address 2	Postal Code	Street Number	Street Na
0	1	13286	201/205	13286	201/205	1013160001	1013160001	1037549	201/205 East 42nd st.	Not Available	10017	675	3 AVEN
1	2	28400	NYP Columbia (West Campus)	28400	NYP Columbia (West Campus)	1021380040	1-02138-0040	1084198; 1084387;1084385; 1084386; 1084388; 10...	622 168th Street	Not Available	10032	180	WASHINGT AVEN
2	3	4778226	MSCHoNY North	28400	NYP Columbia (West Campus)	1021380030	1-02138-0030	1063380	3975 Broadway	Not Available	10032	3975	BROADW

	Order	Property Id	Property Name	Parent Property Id	Parent Property Name	BBL - 10 digits	NYC Borough, Block and Lot (BBL) self-reported	NYC Building Identification Number (BIN)	Address 1 (self-reported)	Address 2	Postal Code	Street Number	Street Na
3	4	4778267	Herbert Irving Pavilion & Millstein Hospital	28400	NYP Columbia (West Campus)	1021390001	1-02139-0001	1087281; 1076746	161 Fort Washington Ave	177 Fort Washington Ave	10032	161	WASHINGT AVEN
4	5	4778288	Neuro Institute	28400	NYP Columbia (West Campus)	1021390085	1-02139-0085	1063403	710 West 168th Street	Not Available	10032	193	WASHINGT AVEN

In [121…

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11746 entries, 0 to 11745
Data columns (total 60 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Order                                           11746 non-null  int64
 1   Property Id                                     11746 non-null  int64
 2   Property Name                                   11746 non-null  object
 3   Parent Property Id                              11746 non-null  object
 4   Parent Property Name                            11746 non-null  object
 5   BBL - 10 digits                                 11735 non-null  object
 6   NYC Borough, Block and Lot (BBL) self-reported  11746 non-null  object
 7   NYC Building Identification Number (BIN)        11746 non-null  object
 8   Address 1 (self-reported)                       11746 non-null  object
 9   Address 2                                       11746 non-null  object
 10  Postal Code                                     11746 non-null  object
 11  Street Number                                   11622 non-null  object
 12  Street Name                                     11624 non-null  object
 13  Borough                                         11628 non-null  object
 14  DOF Gross Floor Area                            11628 non-null  float64
```

```
15  Primary Property Type - Self Selected                          11746 non-null  object
16  List of All Property Use Types at Property                     11746 non-null  object
17  Largest Property Use Type                                      11746 non-null  object
18  Largest Property Use Type - Gross Floor Area (ft²)             11746 non-null  object
19  2nd Largest Property Use Type                                  11746 non-null  object
20  2nd Largest Property Use - Gross Floor Area (ft²)              11746 non-null  object
21  3rd Largest Property Use Type                                  11746 non-null  object
22  3rd Largest Property Use Type - Gross Floor Area (ft²)         11746 non-null  object
23  Year Built                                                     11746 non-null  int64
24  Number of Buildings - Self-reported                            11746 non-null  int64
25  Occupancy                                                      11746 non-null  int64
26  Metered Areas (Energy)                                         11746 non-null  object
27  Metered Areas  (Water)                                         11746 non-null  object
28  ENERGY STAR Score                                              11746 non-null  object
29  Site EUI (kBtu/ft²)                                            11746 non-null  object
30  Weather Normalized Site EUI (kBtu/ft²)                         11746 non-null  object
31  Weather Normalized Site Electricity Intensity (kWh/ft²)        11746 non-null  object
32  Weather Normalized Site Natural Gas Intensity (therms/ft²)     11746 non-null  object
33  Weather Normalized Source EUI (kBtu/ft²)                       11746 non-null  object
34  Fuel Oil #1 Use (kBtu)                                         11746 non-null  object
35  Fuel Oil #2 Use (kBtu)                                         11746 non-null  object
36  Fuel Oil #4 Use (kBtu)                                         11746 non-null  object
37  Fuel Oil #5 & 6 Use (kBtu)                                     11746 non-null  object
38  Diesel #2 Use (kBtu)                                           11746 non-null  object
39  District Steam Use (kBtu)                                      11746 non-null  object
40  Natural Gas Use (kBtu)                                         11746 non-null  object
41  Weather Normalized Site Natural Gas Use (therms)               11746 non-null  object
42  Electricity Use - Grid Purchase (kBtu)                         11746 non-null  object
43  Weather Normalized Site Electricity (kWh)                      11746 non-null  object
44  Total GHG Emissions (Metric Tons CO2e)                         11746 non-null  object
45  Direct GHG Emissions (Metric Tons CO2e)                        11746 non-null  object
46  Indirect GHG Emissions (Metric Tons CO2e)                      11746 non-null  object
47  Property GFA - Self-Reported (ft²)                             11746 non-null  int64
48  Water Use (All Water Sources) (kgal)                           11746 non-null  object
49  Water Intensity (All Water Sources) (gal/ft²)                  11746 non-null  object
50  Source EUI (kBtu/ft²)                                          11746 non-null  object
51  Release Date                                                   11746 non-null  object
52  Water Required?                                                11628 non-null  object
53  DOF Benchmarking Submission Status                             11716 non-null  object
54  Latitude                                                       9483 non-null   float64
55  Longitude                                                      9483 non-null   float64
56  Community Board                                                9483 non-null   float64
57  Council District                                               9483 non-null   float64
58  Census Tract                                                   9483 non-null   float64
59  NTA                                                            9483 non-null   object
```

```
dtypes: float64(6), int64(6), object(48)
memory usage: 5.4+ MB
```

In [122…]
```
data = data.replace({"Not Available":np.nan})
```

In [123…]
```
data.head()
```

Out[123…]

	Order	Property Id	Property Name	Parent Property Id	Parent Property Name	BBL - 10 digits	NYC Borough, Block and Lot (BBL) self-reported	NYC Building Identification Number (BIN)	Address 1 (self-reported)	Address 2	Postal Code	Street Number	Street Na
0	1	13286	201/205	13286	201/205	1013160001	1013160001	1037549	201/205 East 42nd st.	NaN	10017	675	3 AVEN
1	2	28400	NYP Columbia (West Campus)	28400	NYP Columbia (West Campus)	1021380040	1-02138-0040	1084198; 1084387;1084385; 1084386; 1084388; 10...	622 168th Street	NaN	10032	180	WASHINGT AVEN
2	3	4778226	MSCHoNY North	28400	NYP Columbia (West Campus)	1021380030	1-02138-0030	1063380	3975 Broadway	NaN	10032	3975	BROADW
3	4	4778267	Herbert Irving Pavilion & Millstein Hospital	28400	NYP Columbia (West Campus)	1021390001	1-02139-0001	1087281; 1076746	161 Fort Washington Ave	177 Fort Washington Ave	10032	161	WASHINGT AVEN
4	5	4778288	Neuro Institute	28400	NYP Columbia (West Campus)	1021390085	1-02139-0085	1063403	710 West 168th Street	NaN	10032	193	WASHINGT AVEN

In [124…

```
data.isnull().sum()
```

Out[124…

```
Order                                                    0
Property Id                                              0
Property Name                                            0
Parent Property Id                                       0
Parent Property Name                                     0
BBL - 10 digits                                         11
NYC Borough, Block and Lot (BBL) self-reported          11
NYC Building Identification Number (BIN)               162
Address 1 (self-reported)                                0
Address 2                                            11539
Postal Code                                              0
Street Number                                          124
Street Name                                            122
Borough                                                118
DOF Gross Floor Area                                   118
Primary Property Type - Self Selected                    0
List of All Property Use Types at Property               0
Largest Property Use Type                                2
Largest Property Use Type - Gross Floor Area (ft²)       2
2nd Largest Property Use Type                         8005
2nd Largest Property Use - Gross Floor Area (ft²)     8005
3rd Largest Property Use Type                        10262
3rd Largest Property Use Type - Gross Floor Area (ft²)10262
Year Built                                               0
Number of Buildings - Self-reported                      0
Occupancy                                                0
Metered Areas (Energy)                                  57
Metered Areas  (Water)                                4609
ENERGY STAR Score                                     2104
Site EUI (kBtu/ft²)                                    163
Weather Normalized Site EUI (kBtu/ft²)                1465
Weather Normalized Site Electricity Intensity (kWh/ft²)787
Weather Normalized Site Natural Gas Intensity (therms/ft²)1963
Weather Normalized Source EUI (kBtu/ft²)              1465
Fuel Oil #1 Use (kBtu)                               11737
Fuel Oil #2 Use (kBtu)                                9165
Fuel Oil #4 Use (kBtu)                               10425
Fuel Oil #5 & 6 Use (kBtu)                           11152
Diesel #2 Use (kBtu)                                 11730
District Steam Use (kBtu)                            10810
Natural Gas Use (kBtu)                                1442
```

```
Weather Normalized Site Natural Gas Use (therms)              1962
Electricity Use - Grid Purchase (kBtu)                         244
Weather Normalized Site Electricity (kWh)                      786
Total GHG Emissions (Metric Tons CO2e)                          74
Direct GHG Emissions (Metric Tons CO2e)                         83
Indirect GHG Emissions (Metric Tons CO2e)                       65
Property GFA - Self-Reported (ft²)                               0
Water Use (All Water Sources) (kgal)                          3984
Water Intensity (All Water Sources) (gal/ft²)                 3984
Source EUI (kBtu/ft²)                                          163
Release Date                                                     0
Water Required?                                                118
DOF Benchmarking Submission Status                              30
Latitude                                                      2263
Longitude                                                     2263
Community Board                                               2263
Council District                                              2263
Census Tract                                                  2263
NTA                                                           2263
dtype: int64
```

In [125…

```
for col in list(data.columns):
    if("ft²" in col or "kbtu" in col or "kwh" in col or "therms" in col or
       "gal" in col or "Metric Tons CO2e" in col or "Score" in col):

        data[col] = data[col].astype(float)
```

In [126…

```
data.describe().round()
```

Out[126…

	Order	Property Id	DOF Gross Floor Area	Largest Property Use Type - Gross Floor Area (ft²)	2nd Largest Property Use - Gross Floor Area (ft²)	3rd Largest Property Use Type - Gross Floor Area (ft²)	Year Built	Number of Buildings - Self-reported	Occupancy	ENERGY STAR Score	Site EUI (kBtu/ft²)	Weather Normalized Site EUI (kBtu/ft²)	Weather Normalized Site Electricity Intensity (kWh/ft²)
count	11746.0	11746.0	11628.0	11744.0	3741.0	1484.0	11746.0	11746.0	11746.0	9642.0	11583.0	10281.0	10959.0
mean	7186.0	3642958.0	173269.0	160552.0	22779.0	12017.0	1949.0	1.0	99.0	60.0	280.0	310.0	11.0
std	4324.0	1049070.0	336705.0	309575.0	55094.0	27960.0	31.0	4.0	8.0	30.0	8607.0	9785.0	128.0

	Order	Property Id	DOF Gross Floor Area	Largest Property Use Type - Gross Floor Area (ft²)	2nd Largest Property Use - Gross Floor Area (ft²)	3rd Largest Property Use Type - Gross Floor Area (ft²)	Year Built	Number of Buildings - Self-reported	Occupancy	ENERGY STAR Score	Site EUI (kBtu/ft²)	Weather Normalized Site EUI (kBtu/ft²)	Weather Normalized Site Electricity Intensity (kWh/ft²)
min	1.0	7365.0	50028.0	54.0	0.0	0.0	1600.0	0.0	0.0	1.0	0.0	0.0	0.0
25%	3428.0	2747222.0	65240.0	65201.0	4000.0	1721.0	1927.0	1.0	100.0	37.0	62.0	65.0	4.0
50%	6986.0	3236404.0	93138.0	91324.0	8654.0	5000.0	1941.0	1.0	100.0	65.0	78.0	82.0	5.0
75%	11054.0	4409092.0	159614.0	153255.0	20000.0	12000.0	1966.0	1.0	100.0	85.0	98.0	102.0	9.0
max	14993.0	5991312.0	13540113.0	14217119.0	962428.0	591640.0	2019.0	161.0	100.0	100.0	869265.0	939329.0	6259.0

In [127…

```
def miss_val_tab(df):
    miss_val = df.isnull().sum()
    miss_val_per = 100*df.isnull().sum()/len(df)
    miss_val_tab = pd.concat([miss_val,miss_val_per], axis = 1)
    miss_val_ren_tab = miss_val_tab.rename(columns = {0:"missing value",1:"% of value"})
    miss_val_ren_tab = miss_val_ren_tab[miss_val_ren_tab.iloc[:,1] !=0].sort_values("% of value", ascending = False).
    return miss_val_ren_tab
```

In [128…

```
miss_val_tab(data)
```

Out[128…

	missing value	% of value
Fuel Oil #1 Use (kBtu)	11737	99.9
Diesel #2 Use (kBtu)	11730	99.9
Address 2	11539	98.2
Fuel Oil #5 & 6 Use (kBtu)	11152	94.9
District Steam Use (kBtu)	10810	92.0
Fuel Oil #4 Use (kBtu)	10425	88.8

	missing value	% of value
3rd Largest Property Use Type - Gross Floor Area (ft²)	10262	87.4
3rd Largest Property Use Type	10262	87.4
Fuel Oil #2 Use (kBtu)	9165	78.0
2nd Largest Property Use Type	8005	68.2
2nd Largest Property Use - Gross Floor Area (ft²)	8005	68.2
Metered Areas (Water)	4609	39.2
Water Intensity (All Water Sources) (gal/ft²)	3984	33.9
Water Use (All Water Sources) (kgal)	3984	33.9
Latitude	2263	19.3
Longitude	2263	19.3
Community Board	2263	19.3
Council District	2263	19.3
Census Tract	2263	19.3
NTA	2263	19.3
ENERGY STAR Score	2104	17.9
Weather Normalized Site Natural Gas Intensity (therms/ft²)	1963	16.7
Weather Normalized Site Natural Gas Use (therms)	1962	16.7
Weather Normalized Source EUI (kBtu/ft²)	1465	12.5
Weather Normalized Site EUI (kBtu/ft²)	1465	12.5
Natural Gas Use (kBtu)	1442	12.3
Weather Normalized Site Electricity Intensity (kWh/ft²)	787	6.7
Weather Normalized Site Electricity (kWh)	786	6.7
Electricity Use - Grid Purchase (kBtu)	244	2.1
Site EUI (kBtu/ft²)	163	1.4
Source EUI (kBtu/ft²)	163	1.4

	missing value	% of value
NYC Building Identification Number (BIN)	162	1.4
Street Number	124	1.1
Street Name	122	1.0
Borough	118	1.0
DOF Gross Floor Area	118	1.0
Water Required?	118	1.0
Direct GHG Emissions (Metric Tons CO2e)	83	0.7
Total GHG Emissions (Metric Tons CO2e)	74	0.6
Indirect GHG Emissions (Metric Tons CO2e)	65	0.6
Metered Areas (Energy)	57	0.5
DOF Benchmarking Submission Status	30	0.3
NYC Borough, Block and Lot (BBL) self-reported	11	0.1
BBL - 10 digits	11	0.1
Largest Property Use Type	2	0.0
Largest Property Use Type - Gross Floor Area (ft²)	2	0.0

In [129…

```
missing_df = miss_val_tab(data)
missing_cols = missing_df[missing_df["% of value"]>50].index
data = data.drop(columns = list(missing_cols))
```

In [130…

```
miss_val_tab(data)
```

Out[130…

	missing value	% of value
Metered Areas (Water)	4609	39.2
Water Intensity (All Water Sources) (gal/ft²)	3984	33.9
Water Use (All Water Sources) (kgal)	3984	33.9

	missing value	% of value
NTA	2263	19.3
Census Tract	2263	19.3
Council District	2263	19.3
Community Board	2263	19.3
Longitude	2263	19.3
Latitude	2263	19.3
ENERGY STAR Score	2104	17.9
Weather Normalized Site Natural Gas Intensity (therms/ft²)	1963	16.7
Weather Normalized Site Natural Gas Use (therms)	1962	16.7
Weather Normalized Source EUI (kBtu/ft²)	1465	12.5
Weather Normalized Site EUI (kBtu/ft²)	1465	12.5
Natural Gas Use (kBtu)	1442	12.3
Weather Normalized Site Electricity Intensity (kWh/ft²)	787	6.7
Weather Normalized Site Electricity (kWh)	786	6.7
Electricity Use - Grid Purchase (kBtu)	244	2.1
Site EUI (kBtu/ft²)	163	1.4
Source EUI (kBtu/ft²)	163	1.4
NYC Building Identification Number (BIN)	162	1.4
Street Number	124	1.1
Street Name	122	1.0
Water Required?	118	1.0
DOF Gross Floor Area	118	1.0
Borough	118	1.0
Direct GHG Emissions (Metric Tons CO2e)	83	0.7
Total GHG Emissions (Metric Tons CO2e)	74	0.6

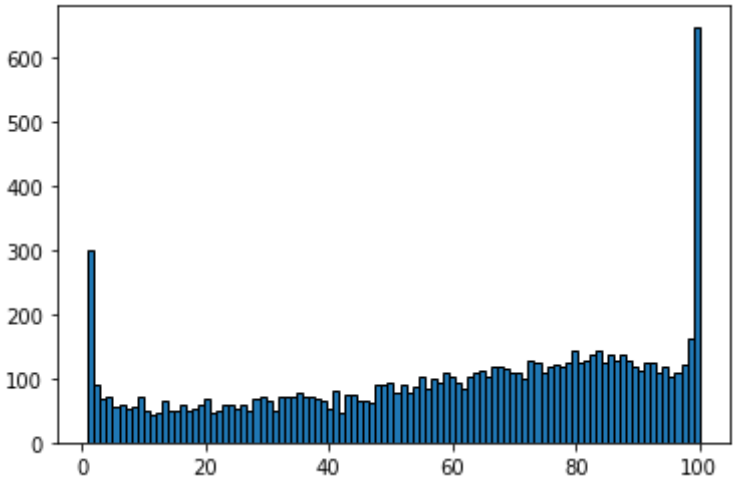
	missing value	% of value
Indirect GHG Emissions (Metric Tons CO2e)	65	0.6
Metered Areas (Energy)	57	0.5
DOF Benchmarking Submission Status	30	0.3
BBL - 10 digits	11	0.1
NYC Borough, Block and Lot (BBL) self-reported	11	0.1
Largest Property Use Type - Gross Floor Area (ft²)	2	0.0
Largest Property Use Type	2	0.0

In [131…
```
data.rename(columns= {"Number of Buildings - Self-reported":"number"}, inplace= True)
```

In [132…
```
data['ENERGY STAR Score'].value_counts()
```

Out[132…
```
100.0    649
1.0      299
99.0     162
80.0     144
84.0     142
        ...
15.0      48
12.0      47
42.0      47
21.0      46
11.0      44
Name: ENERGY STAR Score, Length: 100, dtype: int64
```

In [133…
```
data.rename(columns= {"ENERGY STAR Score":"score"}, inplace= True)
plt.hist(data.score.dropna(),bins=100, edgecolor= "k");
```



In [134…

```
data['Site EUI (kBtu/ft²)'].describe()
```

Out[134…

```
count     11583.000000
mean        280.071484
std        8607.178877
min           0.000000
25%          61.800000
50%          78.500000
75%          97.600000
max      869265.000000
Name: Site EUI (kBtu/ft²), dtype: float64
```

In [135…

```
iqr= data['Site EUI (kBtu/ft²)'].quantile(0.75)-data['Site EUI (kBtu/ft²)'].quantile(0.25)
iqr
```

Out[135…

```
35.8
```

In [136…

```
llmt= data['Site EUI (kBtu/ft²)'].quantile(0.25)-1.5*iqr
ulmt= data['Site EUI (kBtu/ft²)'].quantile(0.75)+1.5*iqr

llmt, ulmt
```

Out[136… (8.100000000000001, 151.29999999999998)

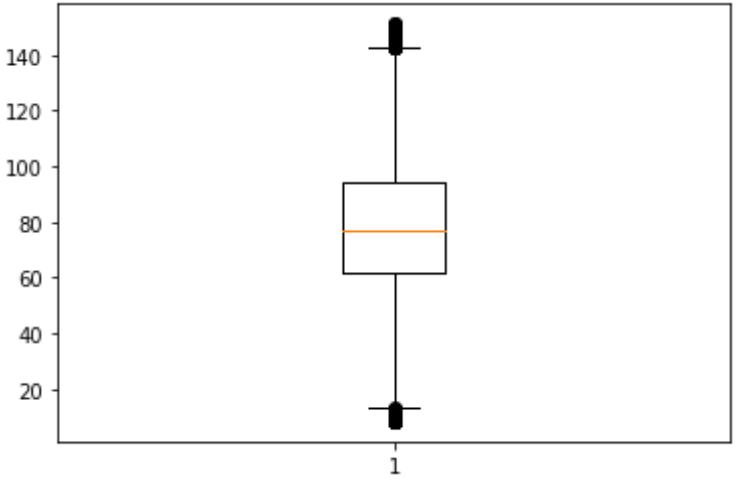
In [137…
```
data = data[(data['Site EUI (kBtu/ft²)']>llmt) & (data['Site EUI (kBtu/ft²)']<ulmt)]
data.shape
```

Out[137… (10766, 49)

In [138…
```
data['Site EUI (kBtu/ft²)'].describe()
```

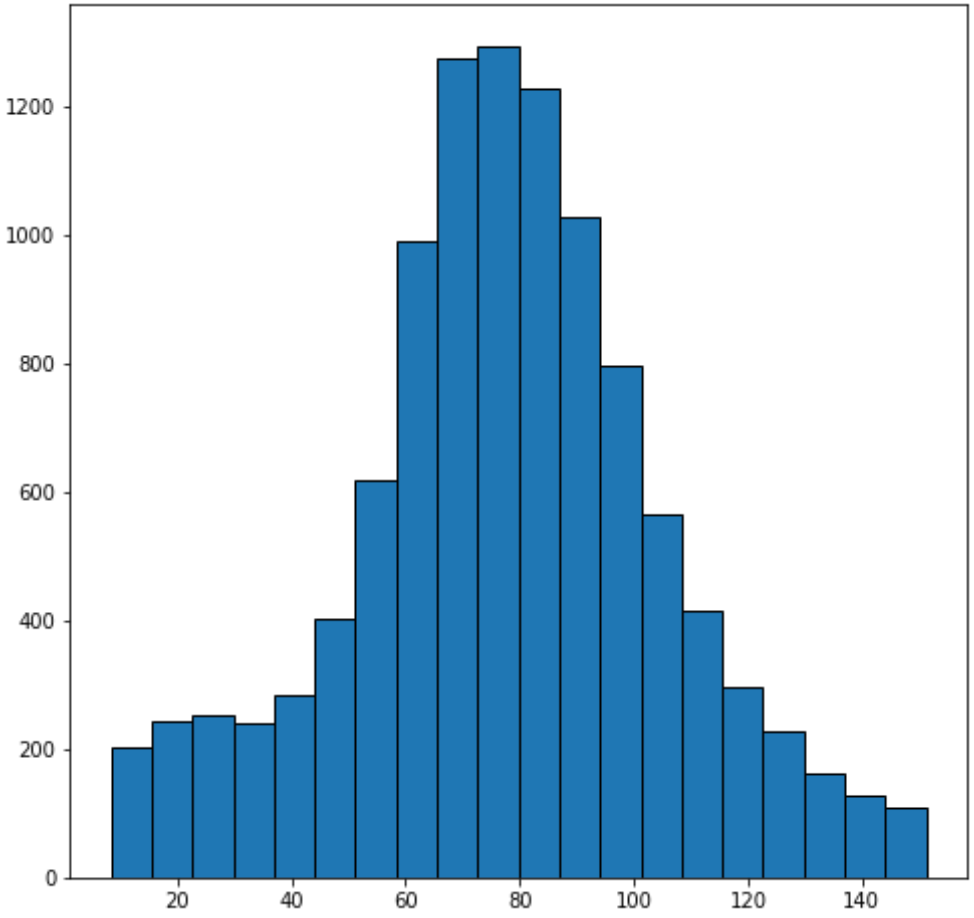
Out[138…
```
count    10766.000000
mean        77.354719
std         27.708067
min          8.400000
25%         61.700000
50%         77.300000
75%         94.100000
max        151.200000
Name: Site EUI (kBtu/ft²), dtype: float64
```

In [139…
```
plt.boxplot(data['Site EUI (kBtu/ft²)']);
```



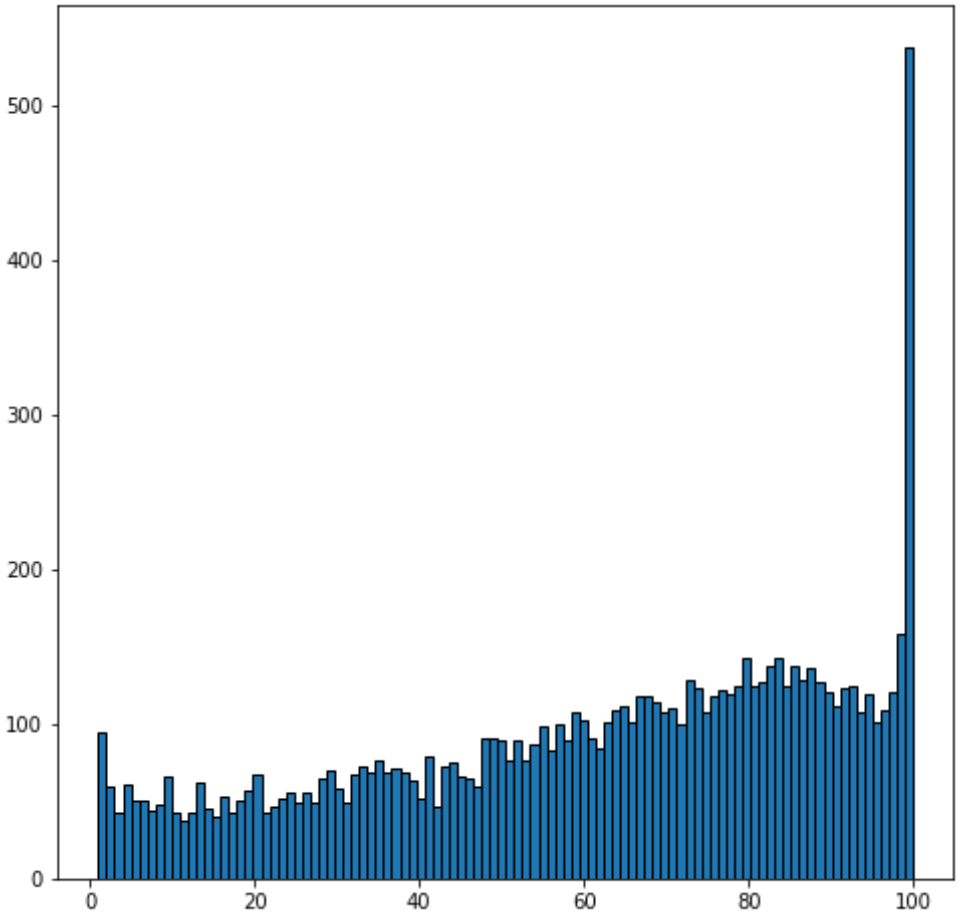
In [140…
```
figsize(8,8)
```

```
plt.hist(data['Site EUI (kBtu/ft²)'].dropna(), bins=20, edgecolor = "black");
```



In [141…

```
plt.hist(data.score, edgecolor="k", bins = 100);
```



In [142…

```
data['score'].value_counts()
```

Out[142…

```
100.0    538
99.0     158
80.0     143
84.0     142
86.0     138
        ...
3.0       43
17.0      43
21.0      42
```

```
15.0      40
11.0      38
Name: score, Length: 100, dtype: int64
```

In [143…

```
data['Largest Property Use Type'].value_counts()
```

Out[143…

```
Multifamily Housing                                8267
Office                                             1227
Non-Refrigerated Warehouse                          178
Hotel                                               177
Other                                               120
K-12 School                                          98
Residence Hall/Dormitory                             86
Retail Store                                         79
Self-Storage Facility                                75
College/University                                   64
Senior Care Community                                62
Distribution Center                                  62
Manufacturing/Industrial Plant                       41
Parking                                              29
Medical Office                                       26
Other - Entertainment/Public Assembly                16
Worship Facility                                     15
Financial Office                                     15
Other - Mall                                         11
Hospital (General Medical & Surgical)                10
Other - Education                                    10
Supermarket/Grocery Store                            10
Refrigerated Warehouse                                8
Strip Mall                                            8
Performing Arts                                       8
Other - Lodging/Residential                           6
Urgent Care/Clinic/Other Outpatient                   5
Enclosed Mall                                         5
Social/Meeting Hall                                   4
Automobile Dealership                                 4
Repair Services (Vehicle, Shoe, Locksmith, etc.)      4
Other - Specialty Hospital                            4
Movie Theater                                         4
Wholesale Club/Supercenter                            4
Residential Care Facility                             3
Museum                                                3
Adult Education                                       3
Other - Public Services                               2
```

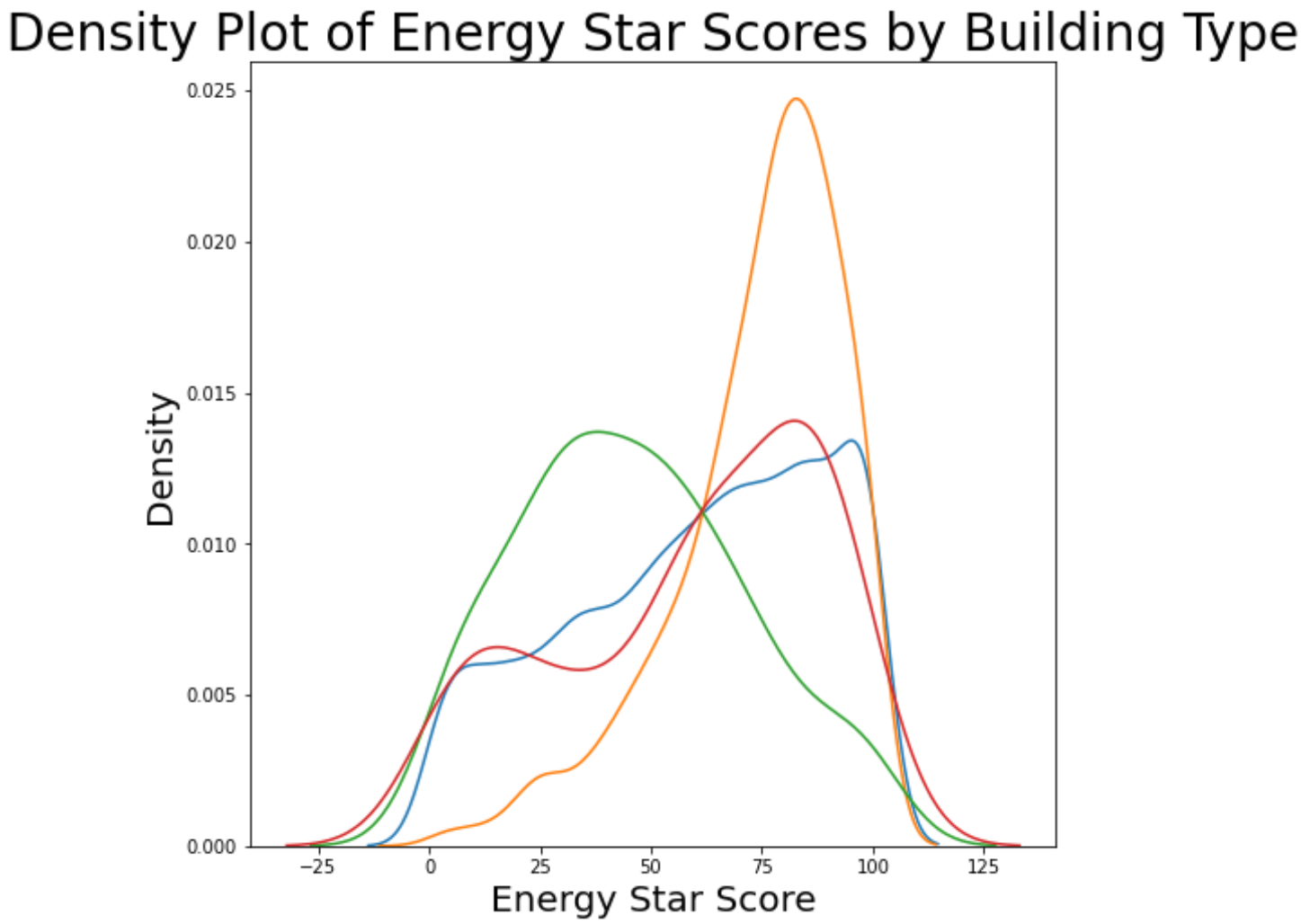
```
Courthouse                                                  2
Outpatient Rehabilitation/Physical Therapy                  2
Other - Services                                            2
Other - Recreation                                          2
Pre-school/Daycare                                          1
Mailing Center/Post Office                                  1
Bank Branch                                                 1
Library                                                     1
Convenience Store without Gas Station                       1
Name: Largest Property Use Type, dtype: int64
```

In [144…

```
types = data.dropna(subset = ['score'])
types = types['Largest Property Use Type'].value_counts()
types = list(types[types.values>100].index)
```

In [145…

```
for b_type in types:
    subset = data[data['Largest Property Use Type'] == b_type]
    sns.kdeplot(subset['score'])
plt.xlabel('Energy Star Score', size = 20); plt.ylabel('Density', size = 20);
plt.title('Density Plot of Energy Star Scores by Building Type', size = 28);
```

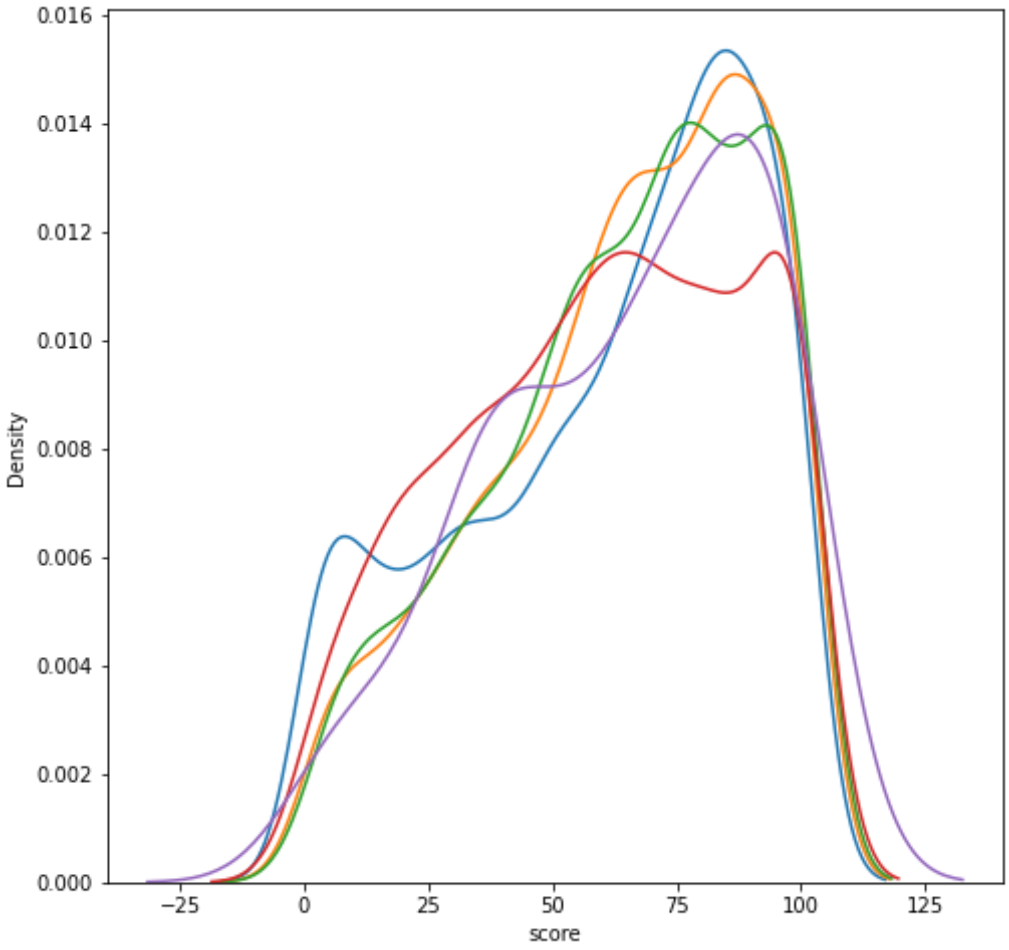


In [146…

```
b_ind = list(data.Borough.value_counts().index)
```

In [147…

```
for b in b_ind:
    subset= data[data['Borough'] == b]
    sns.kdeplot(subset['score'].dropna())
```



In [148…

```
num_var = data.select_dtypes('number')

for i in num_var.columns:
    if i == 'score':
        next
    else:
        num_var['sqrt_'+i] = np.sqrt(num_var[i])
        num_var['log_'+i] = np.log(num_var[i])

cat_var = data[['Borough','Largest Property Use Type']]
```

```
cat_var = pd.get_dummies(cat_var)
features= pd.concat([num_var,cat_var], axis=1)
features =features.dropna(subset = ['score'])
corr = features.corr()['score'].dropna().sort_values()
```

```
<ipython-input-148-f33e199f34e6>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  num_var['sqrt_'+i] = np.sqrt(num_var[i])
<ipython-input-148-f33e199f34e6>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  num_var['log_'+i] = np.log(num_var[i])
c:\users\hp\appdata\local\programs\python\python39\lib\site-packages\pandas\core\arraylike.py:358: RuntimeWarning: di
vide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
c:\users\hp\appdata\local\programs\python\python39\lib\site-packages\pandas\core\arraylike.py:358: RuntimeWarning: in
valid value encountered in sqrt
  result = getattr(ufunc, method)(*inputs, **kwargs)
c:\users\hp\appdata\local\programs\python\python39\lib\site-packages\pandas\core\arraylike.py:358: RuntimeWarning: in
valid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

In [149…

```
corr.head()
```

Out[149…

```
Site EUI (kBtu/ft²)                          -0.707467
Weather Normalized Site EUI (kBtu/ft²)       -0.692744
sqrt_Site EUI (kBtu/ft²)                     -0.682425
sqrt_Weather Normalized Site EUI (kBtu/ft²)  -0.667331
sqrt_Weather Normalized Source EUI (kBtu/ft²)  -0.660359
Name: score, dtype: float64
```

In [150…

```
corr.tail()
```

Out[150…

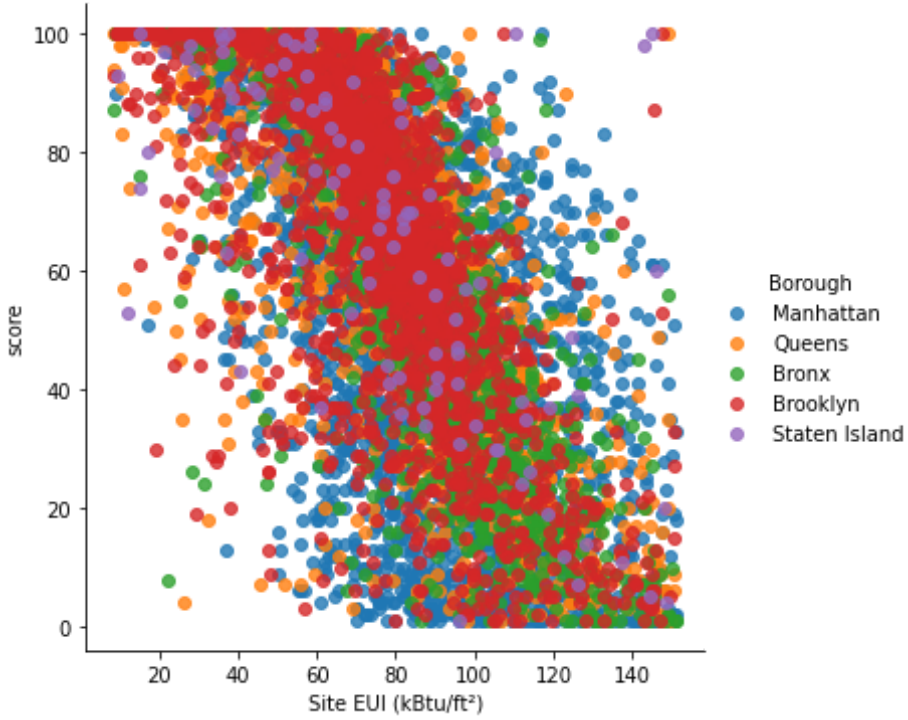
```
Community Board                    0.063283
```

```
log_Community Board                  0.065167
sqrt_Community Board                 0.066266
Largest Property Use Type_Office     0.168125
score                                1.000000
Name: score, dtype: float64
```

In [152…

```
features['Borough'] = data.dropna(subset = ['score'])['Borough']
features = features[features['Borough'].isin(b_ind)]

sns.lmplot('Site EUI (kBtu/ft²)','score',hue= 'Borough',data = features, fit_reg = False);
```

```
c:\users\hp\appdata\local\programs\python\python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, a
nd passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

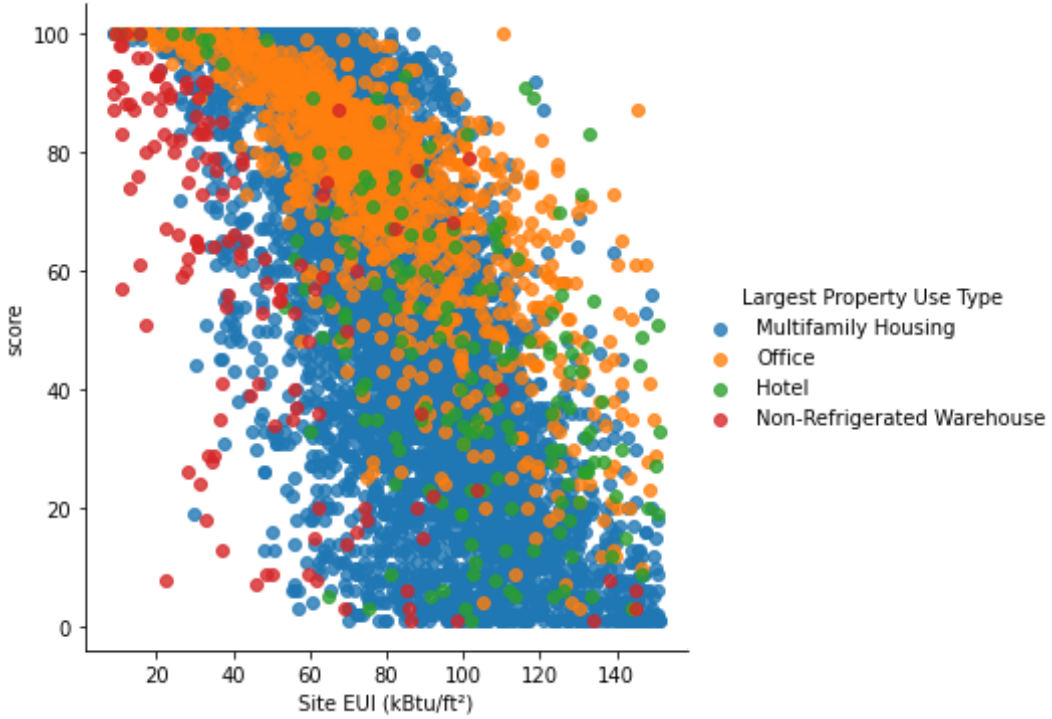


In [154…

```
features['Largest Property Use Type'] = data.dropna(subset= ['score'])['Largest Property Use Type']
featuress= features[features['Largest Property Use Type'].isin(types)]
```

```
sns.lmplot('Site EUI (kBtu/ft²)','score',hue= 'Largest Property Use Type', data= featuress,fit_reg= False);
```

```
c:\users\hp\appdata\local\programs\python\python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, a
nd passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [155…

```
corr.index
```

Out[155…

```
Index(['Site EUI (kBtu/ft²)', 'Weather Normalized Site EUI (kBtu/ft²)',
       'sqrt_Site EUI (kBtu/ft²)',
       'sqrt_Weather Normalized Site EUI (kBtu/ft²)',
       'sqrt_Weather Normalized Source EUI (kBtu/ft²)',
       'sqrt_Source EUI (kBtu/ft²)', 'log_Source EUI (kBtu/ft²)',
       'log_Weather Normalized Source EUI (kBtu/ft²)',
       'Weather Normalized Source EUI (kBtu/ft²)', 'Source EUI (kBtu/ft²)',
       'log_Site EUI (kBtu/ft²)', 'log_Weather Normalized Site EUI (kBtu/ft²)',
       'log_Weather Normalized Site Electricity Intensity (kWh/ft²)',
```

```
'sqrt_Weather Normalized Site Electricity Intensity (kWh/ft²)',
'Weather Normalized Site Electricity Intensity (kWh/ft²)',
'log_Total GHG Emissions (Metric Tons CO2e)',
'Weather Normalized Site Natural Gas Intensity (therms/ft²)',
'sqrt_Direct GHG Emissions (Metric Tons CO2e)',
'sqrt_Weather Normalized Site Natural Gas Intensity (therms/ft²)',
'log_Direct GHG Emissions (Metric Tons CO2e)',
'log_Indirect GHG Emissions (Metric Tons CO2e)',
'sqrt_Weather Normalized Site Natural Gas Use (therms)',
'log_Weather Normalized Site Natural Gas Intensity (therms/ft²)',
'log_Weather Normalized Site Natural Gas Use (therms)',
'sqrt_Total GHG Emissions (Metric Tons CO2e)',
'Direct GHG Emissions (Metric Tons CO2e)',
'Weather Normalized Site Natural Gas Use (therms)',
'log_Water Intensity (All Water Sources) (gal/ft²)', 'Year Built',
'sqrt_Year Built', 'log_Year Built',
'sqrt_Indirect GHG Emissions (Metric Tons CO2e)',
'log_Water Use (All Water Sources) (kgal)',
'Largest Property Use Type_Multifamily Housing',
'Total GHG Emissions (Metric Tons CO2e)',
'Largest Property Use Type_Hotel',
'sqrt_Water Intensity (All Water Sources) (gal/ft²)',
'sqrt_Water Use (All Water Sources) (kgal)', 'log_Property Id',
'sqrt_Property Id', 'Property Id', 'log_Latitude', 'sqrt_Latitude',
'Latitude', 'Longitude', 'Borough_Bronx',
'Indirect GHG Emissions (Metric Tons CO2e)', 'Borough_Manhattan',
'Occupancy', 'sqrt_Occupancy',
'Largest Property Use Type_Senior Care Community', 'log_Occupancy',
'Largest Property Use Type_Distribution Center', 'sqrt_number',
'number', 'log_number',
'Largest Property Use Type_Wholesale Club/Supercenter',
'Water Intensity (All Water Sources) (gal/ft²)',
'Largest Property Use Type_Non-Refrigerated Warehouse',
'Water Use (All Water Sources) (kgal)',
'Largest Property Use Type_K-12 School', 'Census Tract',
'Largest Property Use Type_Bank Branch', 'sqrt_Census Tract',
'log_DOF Gross Floor Area',
'Largest Property Use Type_Worship Facility', 'log_Census Tract',
'Largest Property Use Type_Refrigerated Warehouse',
'Largest Property Use Type_Medical Office', 'sqrt_DOF Gross Floor Area',
'Borough_Staten Island', 'Largest Property Use Type_Parking',
'Largest Property Use Type_Courthouse',
'Largest Property Use Type_Residence Hall/Dormitory',
'Largest Property Use Type_Financial Office', 'DOF Gross Floor Area',
'log_Order', 'sqrt_Property GFA - Self-Reported (ft²)',
```

```
       'log_Property GFA - Self-Reported (ft²)',
       'sqrt_Largest Property Use Type - Gross Floor Area (ft²)',
       'Property GFA - Self-Reported (ft²)',
       'Largest Property Use Type_Retail Store',
       'Largest Property Use Type - Gross Floor Area (ft²)',
       'log_Largest Property Use Type - Gross Floor Area (ft²)',
       'Largest Property Use Type_Supermarket/Grocery Store', 'sqrt_Order',
       'Borough_Queens', 'Borough_Brooklyn', 'Order',
       'Largest Property Use Type_Hospital (General Medical & Surgical)',
       'sqrt_Council District', 'log_Council District', 'Council District',
       'Community Board', 'log_Community Board', 'sqrt_Community Board',
       'Largest Property Use Type_Office', 'score'],
      dtype='object')
```

In [156…
```
features.shape
```

Out[156…
```
(9032, 130)
```

In [157…
```
featuress.shape
```

Out[157…
```
(8625, 130)
```

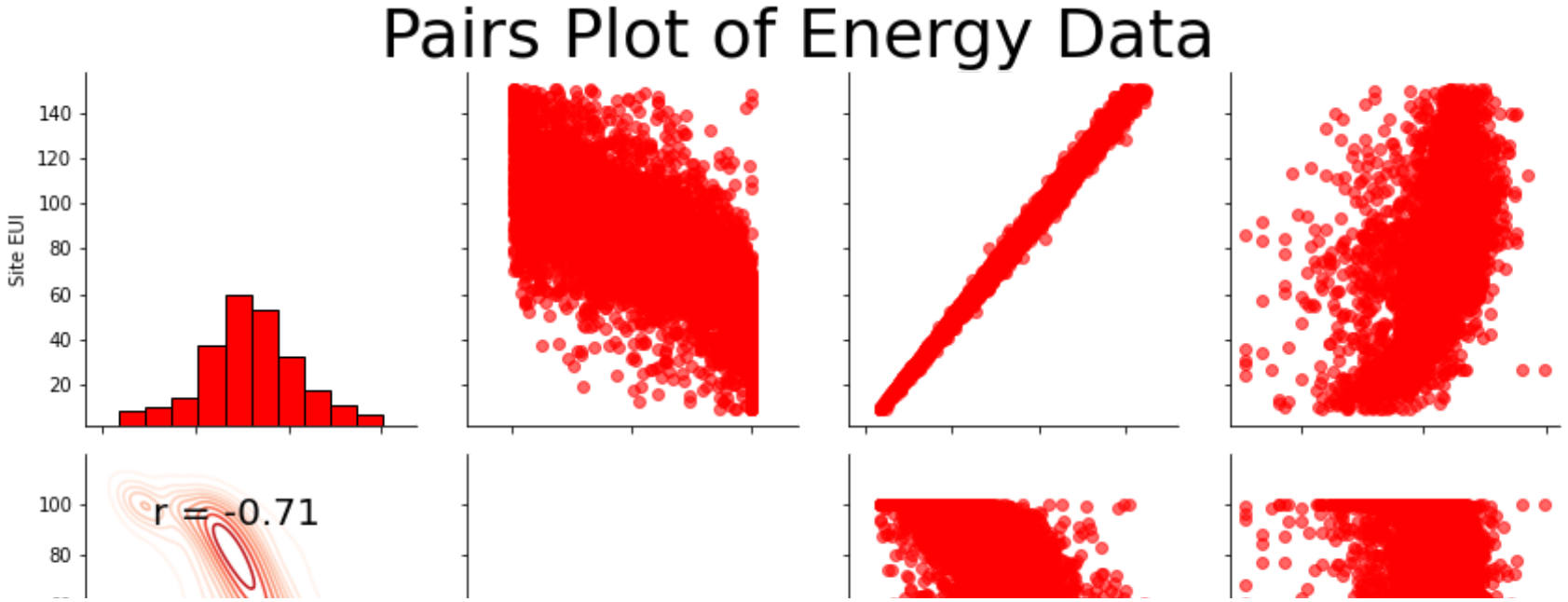
In [158…
```
corr.shape
```

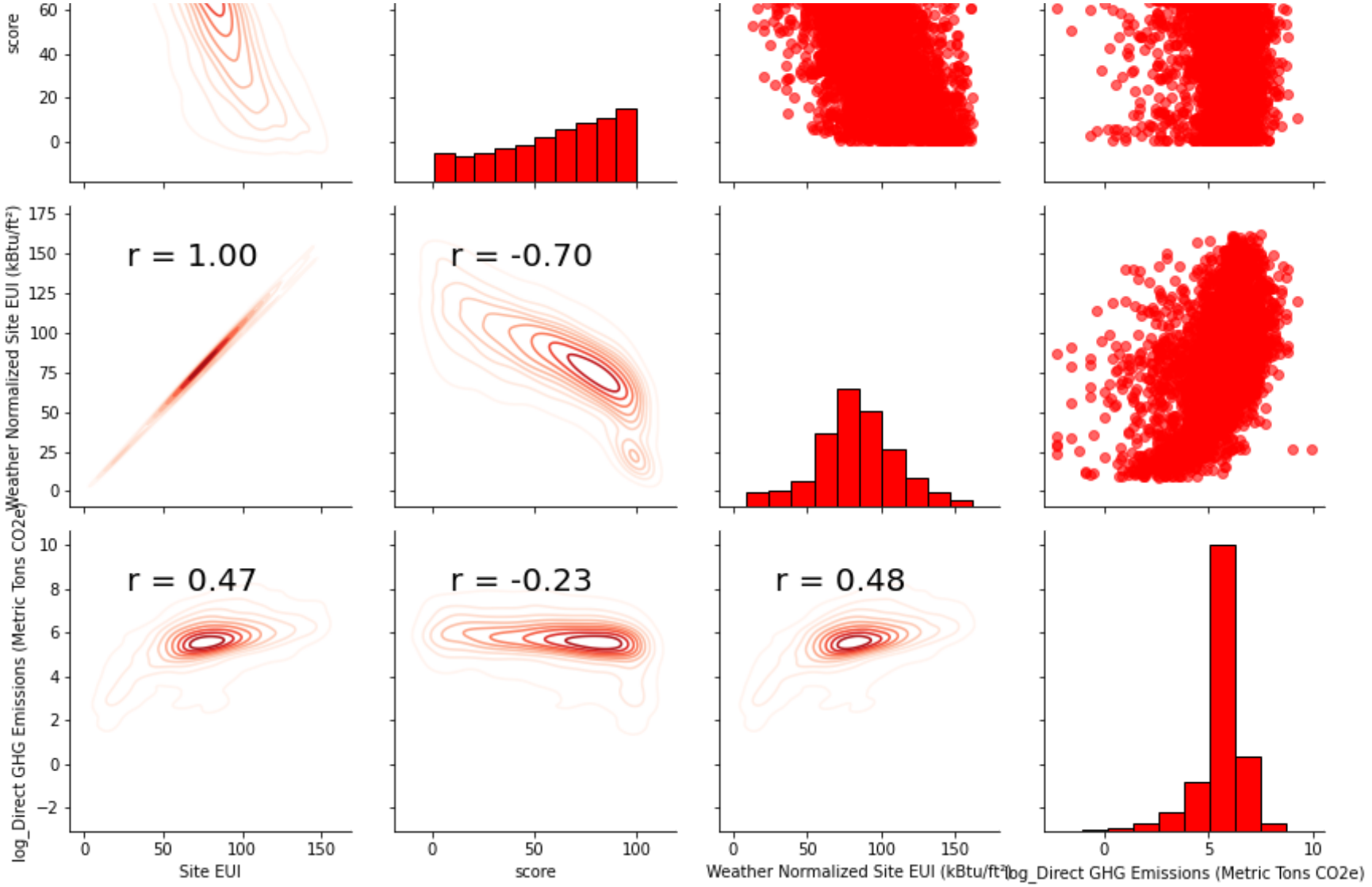
Out[158…
```
(98,)
```

In [159…
```
plot_data = features[['Site EUI (kBtu/ft²)','score','Weather Normalized Site EUI (kBtu/ft²)',
                      'log_Direct GHG Emissions (Metric Tons CO2e)']]

plot_data = plot_data.replace({np.inf: np.nan, -np.inf: np.nan})
plot_data = plot_data.rename(columns = {'Site EUI (kBtu/ft²)': 'Site EUI',
                                        'Weather Normalized Source EUI (kBtu/ft²)': 'Weather Norm EUI',
                                        'log_Total GHG Emissions (Metric Tons CO2e)': 'log GHG Emissions'})


plot_data = plot_data.dropna()

def corr_func(x, y, **kwargs):
    r = np.corrcoef(x, y)[0][1]
```

```
    ax = plt.gca()
    ax.annotate("r = {:.2f}".format(r),
                xy=(.2, .8), xycoords=ax.transAxes,
                size = 20)
grid = sns.PairGrid(data = plot_data, size = 3)

# Upper is a scatter plot
grid.map_upper(plt.scatter, color = 'red', alpha = 0.6)

# Diagonal is a histogram
grid.map_diag(plt.hist, color = 'red', edgecolor = 'black')

# Bottom is correlation and density plot
grid.map_lower(corr_func);
grid.map_lower(sns.kdeplot, cmap = plt.cm.Reds)

# Title for entire plot
plt.suptitle('Pairs Plot of Energy Data', size = 36, y = 1.02);
```

```
c:\users\hp\appdata\local\programs\python\python39\lib\site-packages\seaborn\axisgrid.py:1152: UserWarning: The `size
` parameter has been renamed to `height`; please update your code.
  warnings.warn(UserWarning(msg))
```





In [160…

```
def remove_collinear_features(x, threshold):
    '''

    Objective:
        Remove collinear features in a dataframe with a correlation coefficient
        greater than the threshold. Removing collinear features can help a model
```

```
        to generalize and improves the interpretability of the model.

    Inputs:
        threshold: any features with correlations greater than this value are removed

    Output:
        dataframe that contains only the non-highly-collinear features
    '''

    # Dont want to remove correlations between Energy Star Score
    y = x['score']
    x = x.drop(columns = ['score'])

    # Calculate the correlation matrix
    corr_matrix = x.corr()
    iters = range(len(corr_matrix.columns) - 1)
    drop_cols = []

    # Iterate through the correlation matrix and compare correlations
    for i in iters:
        for j in range(i):
            item = corr_matrix.iloc[j:(j+1), (i+1):(i+2)]
            col = item.columns
            row = item.index
            val = abs(item.values)

            # If correlation exceeds the threshold
            if val >= threshold:
                # Print the correlated features and the correlation value
                # print(col.values[0], "|", row.values[0], "|", round(val[0][0], 2))
                drop_cols.append(col.values[0])

    # Drop one of each pair of correlated columns
    drops = set(drop_cols)
    x = x.drop(columns = drops)
    x = x.drop(columns = ['Weather Normalized Site EUI (kBtu/ft²)',
                          'Water Use (All Water Sources) (kgal)',
                          'log_Water Use (All Water Sources) (kgal)',
                          'Largest Property Use Type - Gross Floor Area (ft²)'])

    # Add the score back in to the data
    x['score'] = y
```

```
    return x
```

In [161…
```
features.shape
```

Out[161…
```
(9032, 130)
```

In [162…
```
features = remove_collinear_features(features, 0.6);
```

In [163…
```
features.shape
```

Out[163…
```
(9032, 68)
```

In [164…
```
features  = features.dropna(axis=1, how = 'all')
features.shape
```

Out[164…
```
(9032, 66)
```

In [168…
```
no_score = features[features['score'].isna()]
score = features[features['score'].notnull()]

print(no_score.shape)
print(score.shape)
```

```
(0, 66)
(9032, 66)
```

In [176…
```
features = score.drop(columns= 'score')
targets = pd.DataFrame(score['score'])

features = features.replace({np.inf:np.nan,-np.inf:np.nan})

x, x_test, y, y_test = train_test_split(features, targets, test_size=0.3,random_state = 42)
```

```
print(x.shape)
print(x_test.shape)
print(y.shape)
print(y_test.shape)
```

```
(6322, 65)
(2710, 65)
(6322, 1)
(2710, 1)
```

In [172…

```
def mae(y_true,y_pred):
    return np.mean(abs(y_true - y_pred))
```

In [175…

```
baseline_guess = np.median(y)
print(baseline_guess)
print(mae(y_test,baseline_guess))
```

```
67.0
score    23.687085
dtype: float64
```

In [177…

```
x.to_csv('F:/Dhrumil/r/Tableau/training_features.csv', index = False)
x_test.to_csv('F:/Dhrumil/r/Tableau/testing_features.csv', index = False)
y.to_csv('F:/Dhrumil/r/Tableau/training_labels.csv', index = False)
y_test.to_csv('F:/Dhrumil/r/Tableau/testing_labels.csv', index = False)
```

TO BE CONTINUED....