In [ ]:

```
#Project milestone 1
```

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statsmodels.formula.api as smf
import seaborn as sns
```

In [3]:

```
NYC = pd.read_csv("DSNY_Monthly_Tonnage_Data.csv")
```

In [4]:

```
NYC['PAPERTONSCOLLECTED'] = NYC['PAPERTONSCOLLECTED'].replace(np.nan, 0)
NYC['MGPTONSCOLLECTED'] = NYC['MGPTONSCOLLECTED'].replace(np.nan, 0)
NYC['RESORGANICSTONS'] = NYC['RESORGANICSTONS'].replace(np.nan, 0)
NYC['SCHOOLORGANICTONS'] = NYC['SCHOOLORGANICTONS'].replace(np.nan, 0)
NYC['LEAVESORGANICTONS'] = NYC['LEAVESORGANICTONS'].replace(np.nan, 0)
NYC['XMASTREETONS'] = NYC['XMASTREETONS'].replace(np.nan, 0)
```

In [6]:

```
NYC.head()
```

Out[6]:

	MONTH	BOROUGH	COMMUNITYDISTRICT	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORGAI
0	1993 / 11	Manhattan	01	625.2	119.4	34.4	
1	1992 / 03	Manhattan	01	726.0	0.0	0.0	
2	1995 / 08	Manhattan	01	627.5	82.4	35.6	
3	1991 / 08	Manhattan	01	695.1	0.0	0.0	
4	1995 / 09	Manhattan	01	633.1	117.2	45.8	

In [7]:

```
Month_2020 = NYC["MONTH"] > "2020"
Month_2020_3 = NYC["MONTH"] < "2020 / 03"
```

In [8]:

```
NYC2 = NYC[Month_2020 & Month_2020_3]
NYC2.head()
```

Out[8]:

	MONTH	BOROUGH	COMMUNITYDISTRICT	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORG
511	2020 / 01	Queens	01	3935.7	662.1	696.1	
627	2020 / 02	Brooklyn	05	3871.4	212.0	343.6	
815	2020 / 01	Staten Island	02	3946.9	586.0	534.8	

	MONTH	BOROUGH	COMMUNITYDISTRICT	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORG
906	2020 / 02	Brooklyn	14	4062.5	429.5	360.5	
983	2020 / 01	Queens	02	2927.0	461.1	438.2	

In [10]:

```
NYC2.dtypes
```

Out[10]:

```
MONTH                  object
BOROUGH                object
COMMUNITYDISTRICT      object
REFUSETONSCOLLECTED   float64
PAPERTONSCOLLECTED    float64
MGPTONSCOLLECTED      float64
RESORGANICSTONS       float64
SCHOOLORGANICTONS     float64
LEAVESORGANICTONS     float64
XMASTREETONS          float64
BOROUGH_ID            float64
dtype: object
```

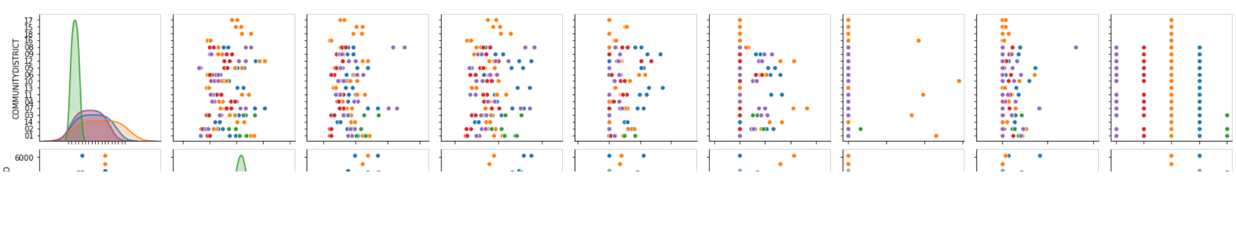
In [11]:

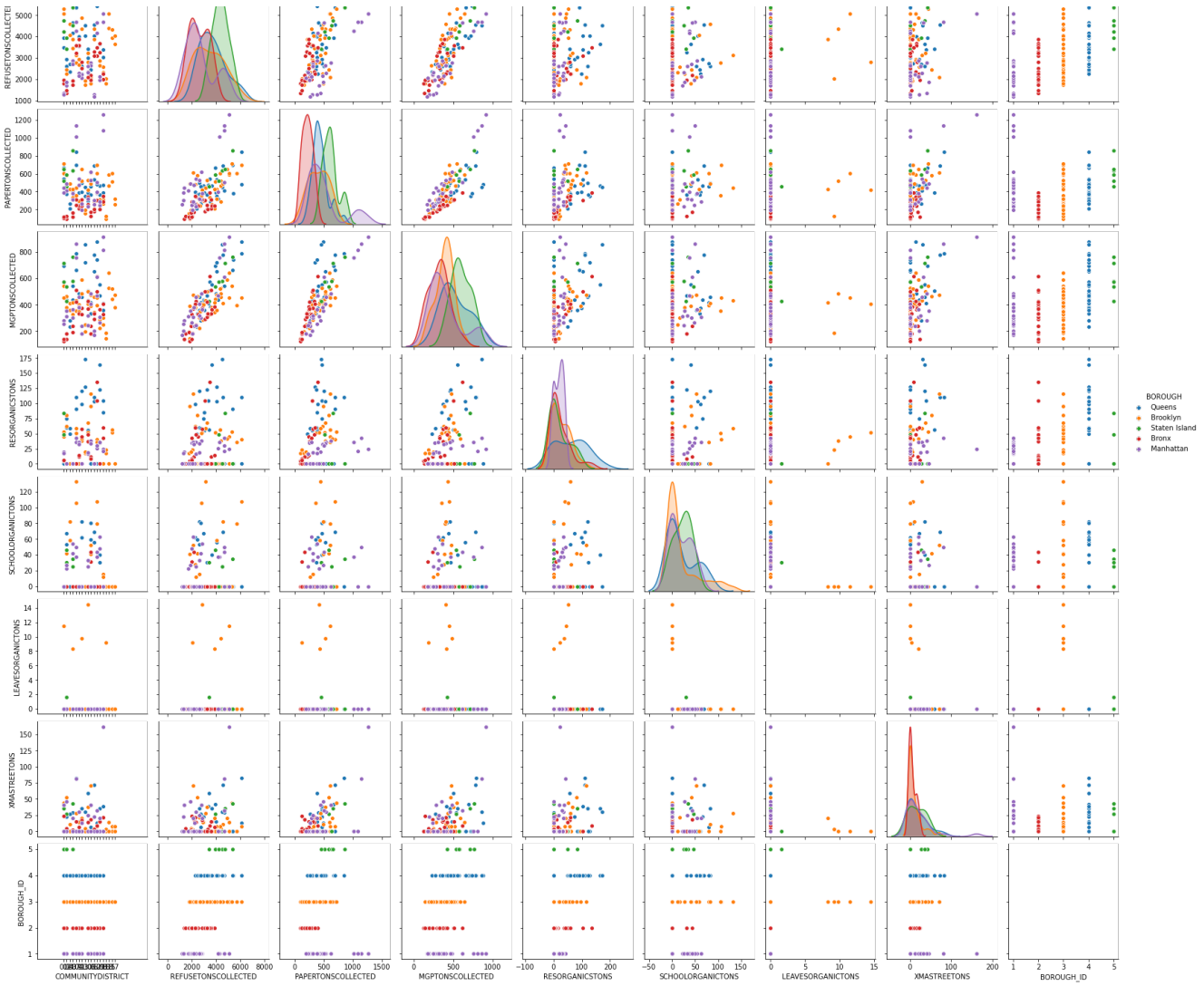
```
sns.pairplot(data = NYC2, hue = "BOROUGH")
```

```
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
```

Out[11]:

```
<seaborn.axisgrid.PairGrid at 0x16b3d105be0>
```



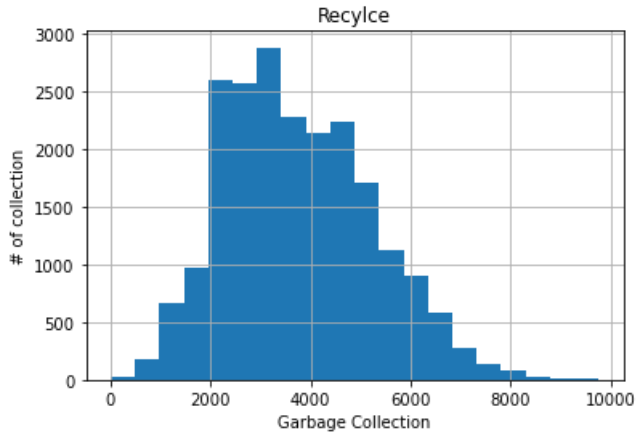


In [9]:

```
NYC["REFUSETONSCOLLECTED"].hist(bins = 20)
plt.title("Recylce")
plt.xlabel("Garbage Collection")
plt.ylabel("# of collection")
```

Out[9]:

```
Text(0, 0.5, '# of collection')
```



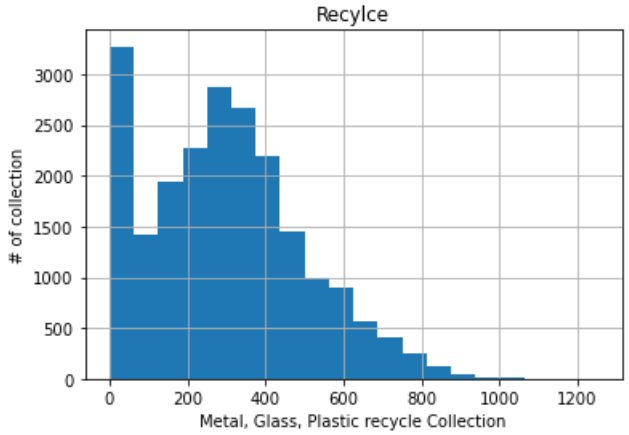
In [10]:

```
NYC["MGPTONSCOLLECTED"].hist(bins = 20)
plt.title("Recylce")
plt.xlabel("Metal, Glass, Plastic recycle Collection")
```

```
plt.ylabel("# of collection")
```

Out[10]:

```
Text(0, 0.5, '# of collection')
```

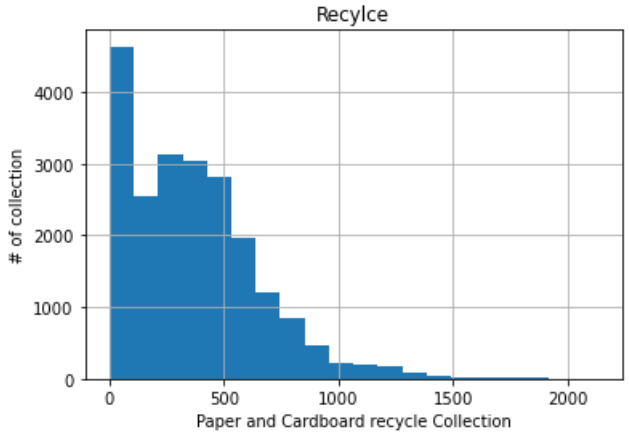


In [11]:

```
NYC["PAPERTONSCOLLECTED"].hist(bins = 20)
plt.title("Recylce")
plt.xlabel("Paper and Cardboard recycle Collection")
plt.ylabel("# of collection")
```

Out[11]:

```
Text(0, 0.5, '# of collection')
```

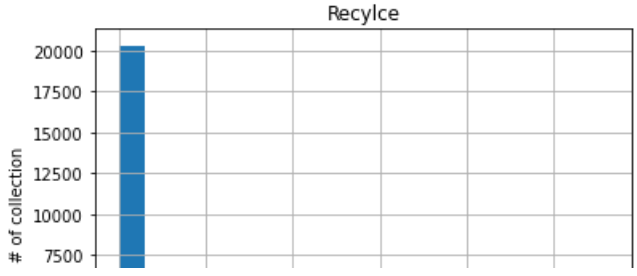


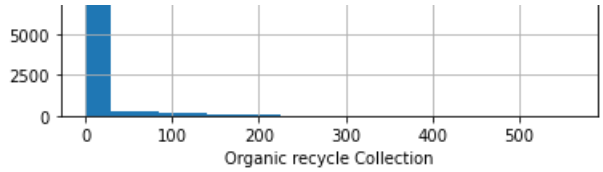
In [12]:

```
NYC["RESORGANICSTONS"].hist(bins = 20)
plt.title("Recylce")
plt.xlabel("Organic recycle Collection")
plt.ylabel("# of collection")
```

Out[12]:

```
Text(0, 0.5, '# of collection')
```



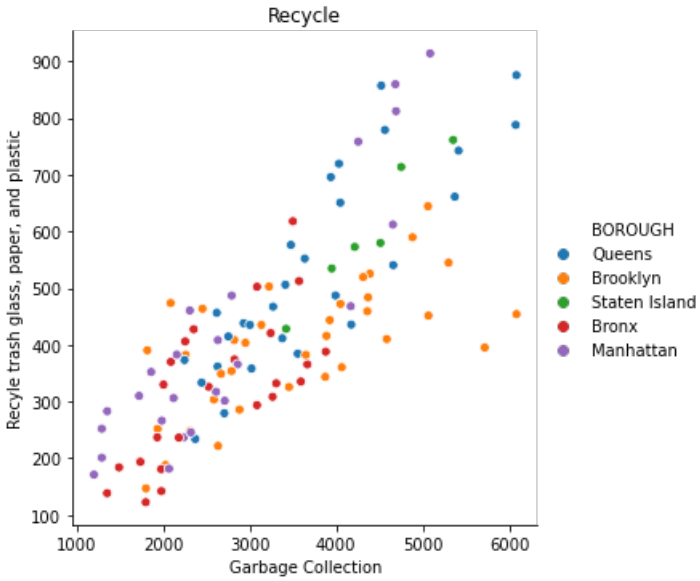


In [20]:

```
sns.relplot(x = "REFUSETONSCOLLECTED", y = "MGPTONSCOLLECTED", hue = "BOROUGH", data = NYC2)
plt.title("Recycle")
plt.xlabel("Garbage Collection")
plt.ylabel("Recyle trash glass, paper, and plastic")
```

Out[20]:

```
Text(35.17570312500001, 0.5, 'Recyle trash glass, paper, and plastic')
```

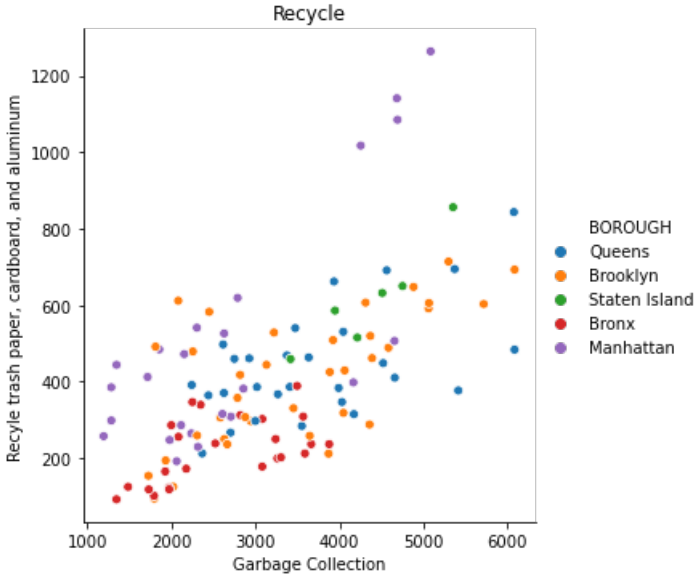


In [14]:

```
sns.relplot(x = "REFUSETONSCOLLECTED", y = "PAPERTONSCOLLECTED", hue = "BOROUGH", data = NYC2)
plt.title("Recycle")
plt.xlabel("Garbage Collection")
plt.ylabel("Recyle trash paper, cardboard, and aluminum")
```

Out[14]:

```
Text(36.9753125, 0.5, 'Recyle trash paper, cardboard, and aluminum')
```



In [41]:

```
#Project  Milesstone 2
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import mean_squared_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import confusion_matrix
import matplotlib
%matplotlib inline
```

In [16]:

```
x = NYC2[["PAPERTONSCOLLECTED", "MGPTONSCOLLECTED","RESORGANICSTONS", "SCHOOLORGANICTONS",
"LEAVESORGANICTONS","XMASTREETONS"]]
y = NYC2["REFUSETONSCOLLECTED"]
```

In [29]:

```
#Decision Tree
```

In [18]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [37]:

```
reg2 = DecisionTreeRegressor(max_depth = 2)
reg2 = reg2.fit(x_train, y_train)
```

In [38]:

```
y_test_pred_2 = reg2.predict(x_test)
```

In [39]:

```
mean_squared_error(y_test_pred_2, y_test)
```

Out[39]:

```
450188.80874662125
```

In [23]:

```
reg2 = DecisionTreeRegressor(max_depth = 2)
reg2 = reg2.fit(x_train, y_train)
```

In [24]:

```
pred_2 = reg2.predict(x_test)
```

In [25]:

```
mean_squared_error(pred_2, y_test)
```

Out[25]:

```
450188.80874662125
```

In [40]:

```
#Linear Regression
```

```
#Linear Regression
```

In [42]:

```
linear_model = LinearRegression()
linear_model.fit(x_train,y_train)
```

Out[42]:

```
LinearRegression()
```

In [43]:

```
y_test_preds_linear = linear_model.predict(x_test)
```

In [44]:

```
mean_squared_error(y_test_preds_linear,y_test)
```

Out[44]:

```
397428.2837027192
```

In [57]:

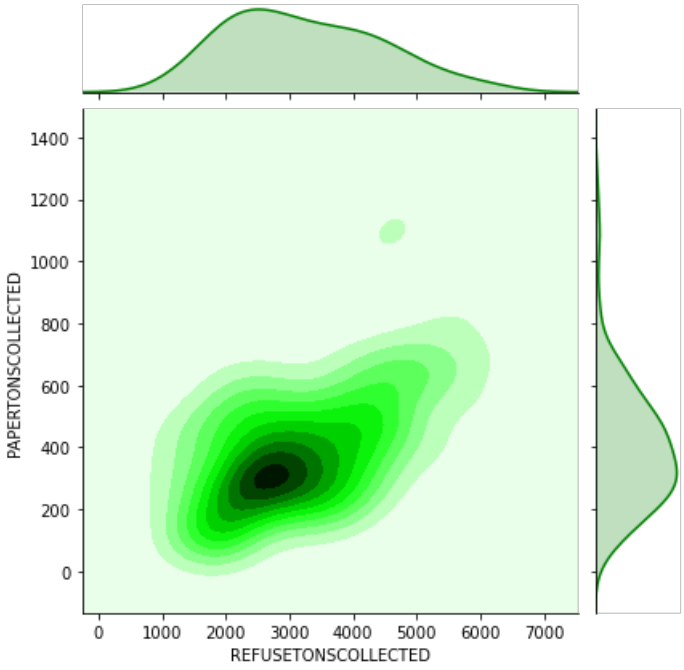
```
# Project Milestone 3
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import confusion_matrix
```

In [58]:

```
sns.jointplot(x = "REFUSETONSCOLLECTED", y = "PAPERTONSCOLLECTED", kind = "kde", color = "Green",da
ta = NYC2)
```

Out[58]:

```
<seaborn.axisgrid.JointGrid at 0x27b83e69f40>
```

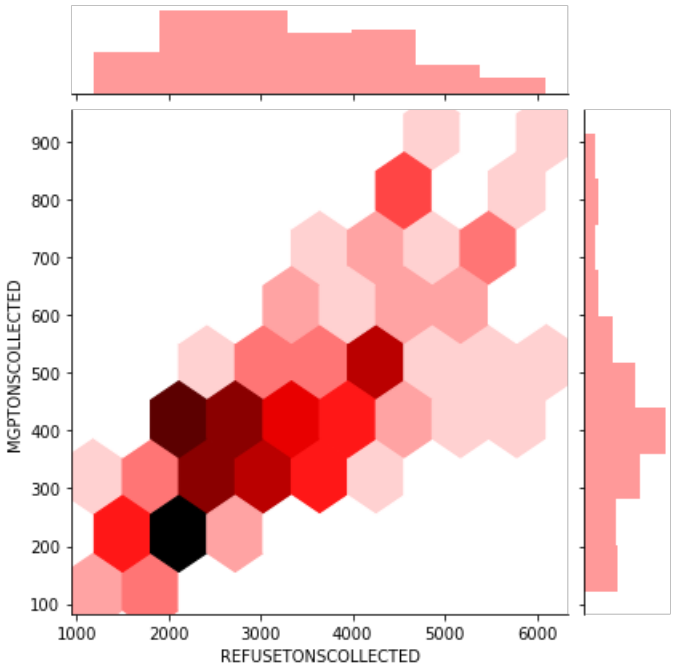


In [59]:

```
sns.jointplot(x = "REFUSETONSCOLLECTED", y = "MGPTONSCOLLECTED", kind = "hex", color = "red",data =
NYC2)
```

Out[59]:

```
<seaborn.axisgrid.JointGrid at 0x27b842111f0>
```



In [60]:

```
!pip install --user folium
```

```
Requirement already satisfied: folium in c:\users\steve\appdata\roaming\python\python38\site-
packages (0.12.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\steve\anaconda3\lib\site-packages (from
folium) (2.11.2)
Requirement already satisfied: numpy in c:\users\steve\anaconda3\lib\site-packages (from folium)
(1.18.5)
Requirement already satisfied: branca>=0.3.0 in
c:\users\steve\appdata\roaming\python\python38\site-packages (from folium) (0.4.2)
Requirement already satisfied: requests in c:\users\steve\anaconda3\lib\site-packages (from
folium) (2.24.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\steve\anaconda3\lib\site-packages
(from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\steve\anaconda3\lib\site-packages (from requests->folium) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in c:\users\steve\anaconda3\lib\site-packages (from
requests->folium) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\steve\anaconda3\lib\site-packages
(from requests->folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\steve\anaconda3\lib\site-packages
(from requests->folium) (2020.6.20)
```

In [122]:

```
import folium
```

In [126]:

```
map = folium.Map(location=[40.8747, -73.8951])
```

In [130]:

```
folium.Marker([40.75, -73.86667], popup="Queens", tooltip = "Click for information",
              icon = folium.Icon(color = 'orange')).add_to(map)
```

Out[130]:

```
<folium.map.Marker at 0x27b9a444e20>
```

In [131]:

```
folium.Marker([40.826153, -73.920265], popup="Bronx", tooltip = "Click for information",
              icon = folium.Icon(color = "red")).add_to(map)
```

Out[131]:

```
<folium.map.Marker at 0x27b9a444e80>
```

In [132]:

```
folium.Marker([40.75325, -74.00381], popup="Manhattan", tooltip = "Click for information",
              icon = folium.Icon(color = "green")).add_to(map)
```

Out[132]:

```
<folium.map.Marker at 0x27b9a5c8790>
```

In [136]:

```
folium.Marker([40.692528, -73.991], popup="Brooklyn", tooltip = "Click for information",
              icon = folium.Icon(color = "blue")).add_to(map)
```

Out[136]:

```
<folium.map.Marker at 0x27b9a4299d0>
```

In [134]:

```
folium.Marker([40.580753, -74.152794], popup="Staten Island", tooltip = "Click for information",
              icon = folium.Icon(color = "purple")).add_to(map)
```

Out[134]:

```
<folium.map.Marker at 0x27b9a50de50>
```

In [137]:

```
map
```

Out[137]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [112]:

```
kmeans = KMeans(n_clusters = 5)
```

In [113]:

```
kmeans.fit(x)
```

Out[113]:

```
KMeans(n_clusters=5)
```

In [114]:

```
clusters = kmeans.predict(x)
```

In [115]:

```
NYC2["clusters"] = clusters
NYC2.head()
```

```
<ipython-input-115-3c21e78c1c31>:1: 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
  NYC2["clusters"] = clusters
```

Out[115]:

	MONTH	BOROUGH	COMMUNITYDISTRICT	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORG
511	2020 / 01	Queens	01	3935.7	662.1	696.1	
627	2020 / 02	Brooklyn	05	3871.4	212.0	343.6	
815	2020 / 01	Staten Island	02	3946.9	586.0	534.8	
906	2020 / 02	Brooklyn	14	4062.5	429.5	360.5	
983	2020 / 01	Queens	02	2927.0	461.1	438.2	

In [96]:

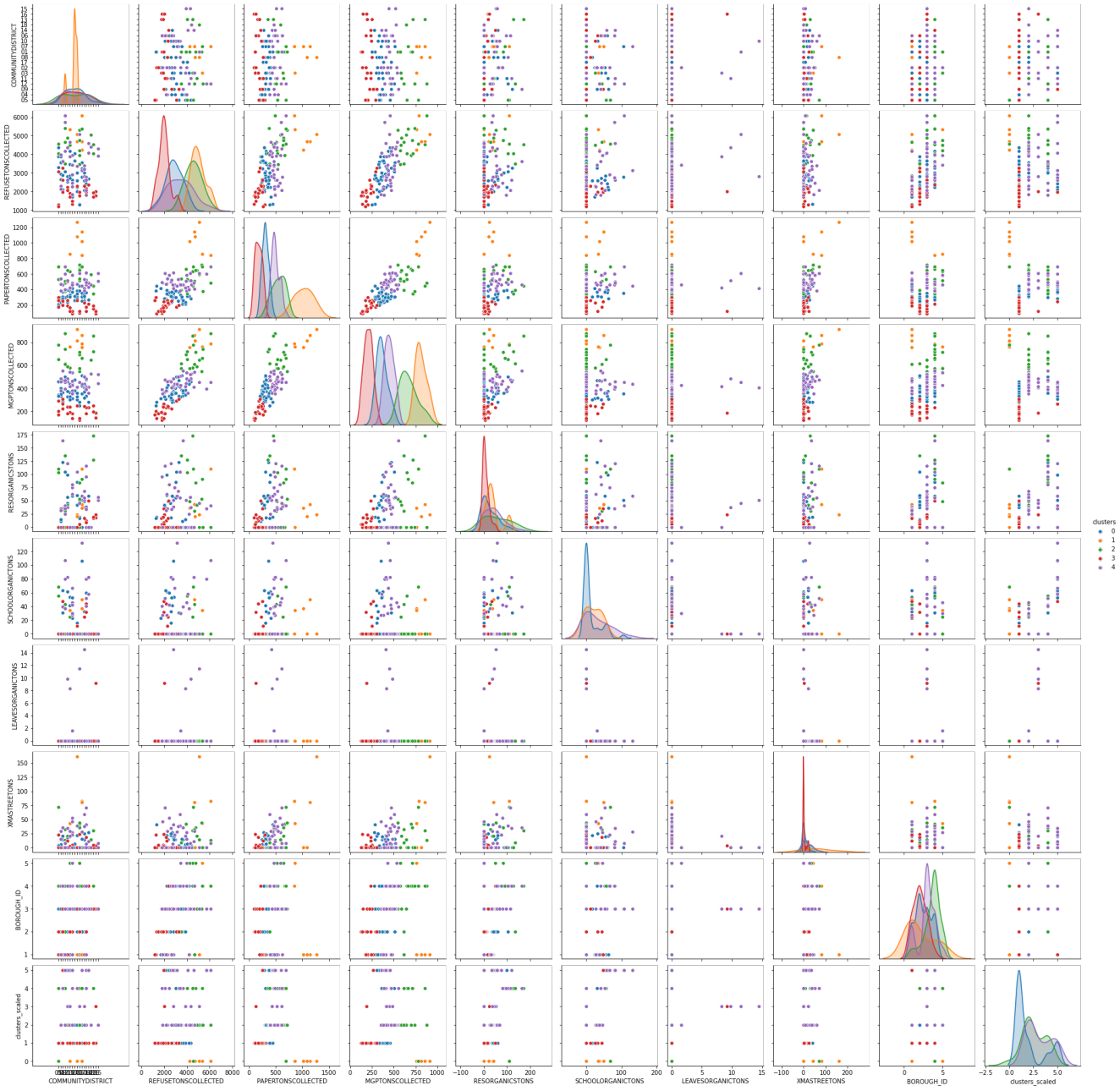
```
sns.pairplot(NYC2, hue = "clusters")
```

```
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
```

```
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
```

Out[96]:

```
<seaborn.axisgrid.PairGrid at 0x27b93c8adf0>
```



In [116]:

```
scaler = MinMaxScaler()
```

In [117]:

```
x_scaled = scaler.fit_transform(x)
x_scaled
```

Out[117]:

```
array([[0.4863388 , 0.72455165, 0.        , 0.        , 0.        ,
        0.259259261,
```

```
 0.25925926],
[0.1020321 , 0.27936348, 0.        , 0.        , 0.        ,
 0.        ],
[0.4213627 , 0.5208386 , 0.        , 0.34867925, 0.        ,
 0.16481481],
[0.28773907, 0.30070725, 0.        , 0.44075472, 0.        ,
 0.        ],
[0.31471995, 0.39883809, 0.43453071, 0.50415094, 0.        ,
 0.14074074],
[0.10262978, 0.14094468, 0.28852839, 0.        , 0.        ,
 0.        ],
[0.41854508, 0.43116949, 0.46755504, 0.31773585, 0.        ,
 0.32407407],
[0.3300888 , 0.32836575, 0.35921205, 0.20830189, 0.        ,
 0.        ],
[0.00187842, 0.03081586, 0.10428737, 0.        , 0.        ,
 0.        ],
[0.02817623, 0.0828492 , 0.1373117 , 0.        , 0.63448276,
 0.02345679],
[0.10228825, 0.26888103, 0.06720742, 0.        , 0.        ,
 0.        ],
[0.30037568, 0.39504925, 0.33893395, 1.        , 0.        ,
 0.17407407],
[0.46063866, 0.57779742, 0.        , 0.18867925, 0.        ,
 0.        ],
[0.42631489, 0.65950998, 0.        , 0.        , 0.        ,
 0.08518519],
[0.12337773, 0.30714827, 0.        , 0.        , 0.        ,
 0.09074074],
[0.36116803, 0.56895681, 0.28331402, 0.        , 0.        ,
 0.        ],
[0.16709358, 0.42548623, 0.        , 0.        , 0.        ,
 0.02839506],
[0.44339139, 0.44379894, 0.66917729, 0.39320755, 0.        ,
 0.43580247],
[0.27160178, 0.52828997, 0.        , 0.        , 0.        ,
 0.18209877],
[0.17930328, 0.48004547, 0.60428737, 0.        , 0.        ,
 0.        ],
[0.6411373 , 0.84074261, 0.63673233, 0.        , 0.        ,
 0.51111111],
[0.24863388, 0.46059611, 0.        , 0.        , 0.        ,
 0.09382716],
[0.18809768, 0.31813589, 0.33140209, 0.        , 0.        ,
 0.10617284],
[0.3457138 , 0.42207628, 0.698146  , 0.61962264, 0.        ,
 0.21604938],
[0.27792008, 0.36145491, 0.29837775, 0.        , 1.        ,
 0.        ],
[0.        , 0.02020712, 0.        , 0.        , 0.        ,
 0.        ],
[0.78961749, 0.80348573, 0.20973349, 0.28      , 0.        ,
 0.        ],
[0.24709699, 0.30714827, 0.        , 0.40679245, 0.        ,
 0.12345679],
[0.27313866, 0.23692852, 0.        , 0.17283019, 0.        ,
 0.        ],
[0.12320697, 0.33543824, 0.08458864, 0.        , 0.        ,
 0.0382716 ],
[0.43826844, 0.41576156, 0.26013905, 0.        , 0.79310345,
 0.        ],
[0.17605874, 0.09926749, 0.        , 0.        , 0.        ,
 0.1382716 ],
[0.21721311, 0.75423087, 0.73696408, 0.        , 0.        ,
 0.        ],
[0.35553279, 0.40578429, 0.29895713, 0.        , 0.        ,
 0.        ],
[0.19006148, 0.39580702, 0.        , 0.        , 0.        ,
 0.        ],
[0.16376366, 0.33101793, 0.        , 0.        , 0.        ,
 0.        ],
[0.51280738, 0.41904521, 0.23522596, 0.8090566 , 0.        ,
 0.04506173],
[0.25093921, 0.48459207, 0.32502897, 0.        , 0.        ,
 0.36419753],
[0.12482923, 0.25713564, 0.26419467, 0.        , 0.        ,
 0.        ],
[0.14182036, 0.32836575, 0.        , 0.        , 0.
```

```
[0.14102050, 0.52050575, 0.        , 0.        , 0.        ,
 0.        ],
[0.31360997, 0.36928517, 0.        , 0.60603774, 0.        ,
 0.03950617],
[0.18280396, 0.22871937, 0.        , 0.11849057, 0.        ,
 0.13271605],
[0.33478484, 0.2902248 , 0.        , 0.2045283 , 0.        ,
 0.28395062],
[0.5307377 , 0.53372064, 0.30880649, 0.        , 0.        ,
 0.27469136],
[0.20346653, 0.25675676, 0.        , 0.        , 0.        ,
 0.        ],
[1.        , 1.        , 0.13904983, 0.        , 0.        ,
 1.        ],
[0.07317281, 0.21634251, 0.        , 0.        , 0.        ,
 0.        ],
[0.4497097 , 0.46046982, 0.22247972, 0.        , 0.        ,
 0.25308642],
[0.14711407, 0.1451124 , 0.        , 0.        , 0.        ,
 0.07716049],
[0.38311134, 0.42725436, 0.20625724, 0.        , 0.        ,
 0.18333333],
[0.30020492, 0.202829  , 0.11297798, 0.        , 0.        ,
 0.25123457],
[0.17460724, 0.35514019, 0.        , 0.        , 0.        ,
 0.08950617],
[0.06839139, 0.14397575, 0.06836616, 0.3290566 , 0.        ,
 0.11358025],
[0.02621243, 0.073756  , 0.05214368, 0.23773585, 0.        ,
 0.        ],
[0.22651981, 0.29237181, 0.297219  , 0.79849057, 0.        ,
 0.        ],
[0.65206626, 0.80727456, 0.        , 0.25962264, 0.        ,
 0.26296296],
[0.36483948, 0.45655469, 0.2184241 , 0.        , 0.67586207,
 0.        ],
[0.51127049, 0.82937611, 0.63673233, 0.51773585, 0.        ,
 0.4462963 ],
[0.13968579, 0.31295782, 0.21436848, 0.        , 0.        ,
 0.        ],
[0.21115096, 0.38545087, 0.28331402, 0.        , 0.        ,
 0.08641975],
[0.24299863, 0.7834049 , 0.        , 0.        , 0.        ,
 0.        ],
[0.3823429 , 0.57325082, 0.        , 0.        , 0.        ,
 0.        ],
[0.31685451, 0.54281384, 0.95075319, 0.30188679, 0.        ,
 0.22222222],
[0.43596311, 0.34465774, 0.20509849, 0.6       , 0.        ,
 0.        ],
[0.25529372, 0.31687295, 0.57126304, 0.46490566, 0.        ,
 0.        ],
[0.32137978, 0.36524375, 0.60544612, 0.        , 0.        ,
 0.23703704],
[0.02792008, 0.07767113, 0.        , 0.        , 0.        ,
 0.02777778],
[0.1835724 , 0.20636524, 0.        , 0.        , 0.        ,
 0.08518519],
[0.4741291 , 0.59030058, 0.2433372 , 0.        , 0.        ,
 0.12901235],
[0.37406079, 0.66746653, 0.59849363, 0.42037736, 0.        ,
 0.        ],
[0.31523224, 0.50909321, 0.        , 0.        , 0.        ,
 0.        ],
[0.33418716, 0.9516292 , 0.        , 0.        , 0.        ,
 0.08333333],
[0.05268101, 0.09156353, 0.10081112, 0.        , 0.        ,
 0.        ],
[0.08683402, 0.1635514 , 0.11239861, 0.        , 0.        ,
 0.02962963],
[0.17503415, 0.39504925, 0.31923523, 0.        , 0.        ,
 0.        ],
[0.25076844, 0.29792877, 0.49188876, 0.        , 0.        ,
 0.        ],
[0.23215505, 0.26648143, 0.        , 0.45056604, 0.        ,
 0.        ],
[0.1425888 , 0.1588785 , 0.        , 0.08754717, 0.        ,
 0.        ],
```

```
 0.        ],
[0.00802596, 0.        , 0.03012746, 0.        , 0.        ,
 0.        ],
[0.1411373 , 0.06137914, 0.        , 0.        , 0.        ,
 0.        ],
[0.13251366, 0.18173781, 0.20336037, 0.36150943, 0.        ,
 0.        ],
[0.16555669, 0.23212933, 0.20683662, 0.47622642, 0.        ,
 0.12654321],
[0.18476776, 0.49280121, 0.        , 0.        , 0.        ,
 0.0962963 ],
[0.02211407, 0.0897954 , 0.        , 0.        , 0.        ,
 0.        ],
[0.09153005, 0.23490781, 0.        , 0.        , 0.        ,
 0.        ],
[0.4760929 , 0.74703208, 0.48435689, 0.        , 0.        ,
 0.22037037],
[0.23463115, 0.43571609, 0.70973349, 0.23245283, 0.        ,
 0.        ],
[0.25315915, 0.62604193, 0.78215527, 0.        , 0.        ,
 0.05185185],
[0.31318306, 0.38633493, 0.        , 0.22792453, 0.11034483,
 0.        ],
[0.23744877, 0.30234908, 0.34530707, 0.40075472, 0.        ,
 0.        ],
[0.14899249, 0.19828239, 0.34125145, 0.        , 0.        ,
 0.16666667],
[0.89506489, 0.93142208, 0.24739282, 0.37584906, 0.        ,
 0.5       ],
[0.13473361, 0.12553675, 0.        , 0.        , 0.        ,
 0.        ],
[0.30404713, 0.92826471, 1.        , 0.        , 0.        ,
 0.18518519],
[0.33845628, 0.36360192, 0.        , 0.62037736, 0.        ,
 0.06666667],
[0.0224556 , 0.02475373, 0.03360371, 0.        , 0.        ,
 0.14691358],
[0.18468238, 0.22606719, 0.        , 0.31471698, 0.        ,
 0.        ],
[0.11731557, 0.15559485, 0.09559676, 0.18867925, 0.        ,
 0.        ],
[0.26101434, 0.43672645, 0.14716107, 0.28      , 0.        ,
 0.        ],
[0.37030396, 0.36120232, 0.19524913, 0.        , 0.        ,
 0.        ],
[0.08512637, 0.07464006, 0.        , 0.        , 0.        ,
 0.        ],
[0.32428279, 0.32887093, 0.18597914, 0.        , 0.        ,
 0.        ],
[0.25008538, 0.16380399, 0.10023175, 0.        , 0.        ,
 0.        ],
[0.12295082, 0.28618338, 0.        , 0.        , 0.        ,
 0.        ],
[0.06224385, 0.14460722, 0.        , 0.        , 0.        ,
 0.01296296],
[0.43912227, 0.50164183, 0.32734647, 0.        , 0.        ,
 0.0462963 ],
[0.34084699, 0.33846931, 0.55330243, 0.30867925, 0.        ,
 0.        ],
[0.51408811, 0.68072746, 0.52838934, 0.        , 0.        ,
 0.        ],
[0.37226776, 0.48055064, 0.39339513, 0.        , 0.        ,
 0.2345679 ],
[0.19304986, 0.44190452, 0.        , 0.        , 0.        ,
 0.04382716],
[0.16555669, 0.26206113, 0.24623407, 0.        , 0.        ,
 0.        ],
[0.21712773, 0.35817126, 0.34994206, 0.        , 0.        ,
 0.13333333],
[0.19057377, 0.24602172, 0.10660487, 0.24679245, 0.        ,
 0.15555556],
[0.28423839, 0.37067441, 0.        , 0.        , 0.57241379,
 0.12901235],
[0.35365437, 0.61871685, 0.17381228, 0.32830189, 0.        ,
 0.19753086],
[0.84725068, 0.87130589, 0.1193511 , 0.        , 0.        ,
 0.        ],
[0.09375   , 0.26483961, 0.        , 0.        , 0.        ,
```

```
        [0.09375   , 0.26483961, 0.        , 0.        , 0.        ,
         0.02345679],
        [0.13464822, 0.37686284, 0.        , 0.        , 0.        ,
         0.        ]])
```

In [118]:

```
kmeans_scaled = KMeans(n_clusters = 5)
kmeans_scaled.fit(x_scaled)
```

Out[118]:

```
KMeans(n_clusters=5)
```

In [119]:

```
clusters_scaled = kmeans_scaled.predict(x_scaled)
clusters_scaled
```

Out[119]:

```
array([0, 1, 0, 2, 2, 1, 3, 1, 1, 4, 1, 2, 0, 0, 1, 0, 1, 3, 0, 3, 3, 1,
       1, 3, 4, 1, 0, 2, 1, 1, 4, 1, 3, 1, 1, 1, 2, 3, 1, 1, 2, 1, 1, 0,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 2, 0, 4, 3, 1, 1, 0, 0, 3, 2, 2, 3,
       1, 1, 0, 3, 1, 0, 1, 1, 1, 3, 2, 1, 1, 1, 2, 2, 1, 1, 1, 3, 3, 3,
       1, 2, 1, 0, 1, 3, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 3, 3, 3, 1,
       1, 1, 1, 4, 0, 0, 1, 1])
```

In [120]:

```
NYC2["clusters_scaled"] = clusters_scaled
NYC2.head()
```

```
<ipython-input-120-52af0efa935c>:1: 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
  NYC2["clusters_scaled"] = clusters_scaled
```

Out[120]:

	MONTH	BOROUGH	COMMUNITYDISTRICT	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORG
511	2020 / 01	Queens	01	3935.7	662.1	696.1	
627	2020 / 02	Brooklyn	05	3871.4	212.0	343.6	
815	2020 / 01	Staten Island	02	3946.9	586.0	534.8	
906	2020 / 02	Brooklyn	14	4062.5	429.5	360.5	
983	2020 / 01	Queens	02	2927.0	461.1	438.2	

In [102]:

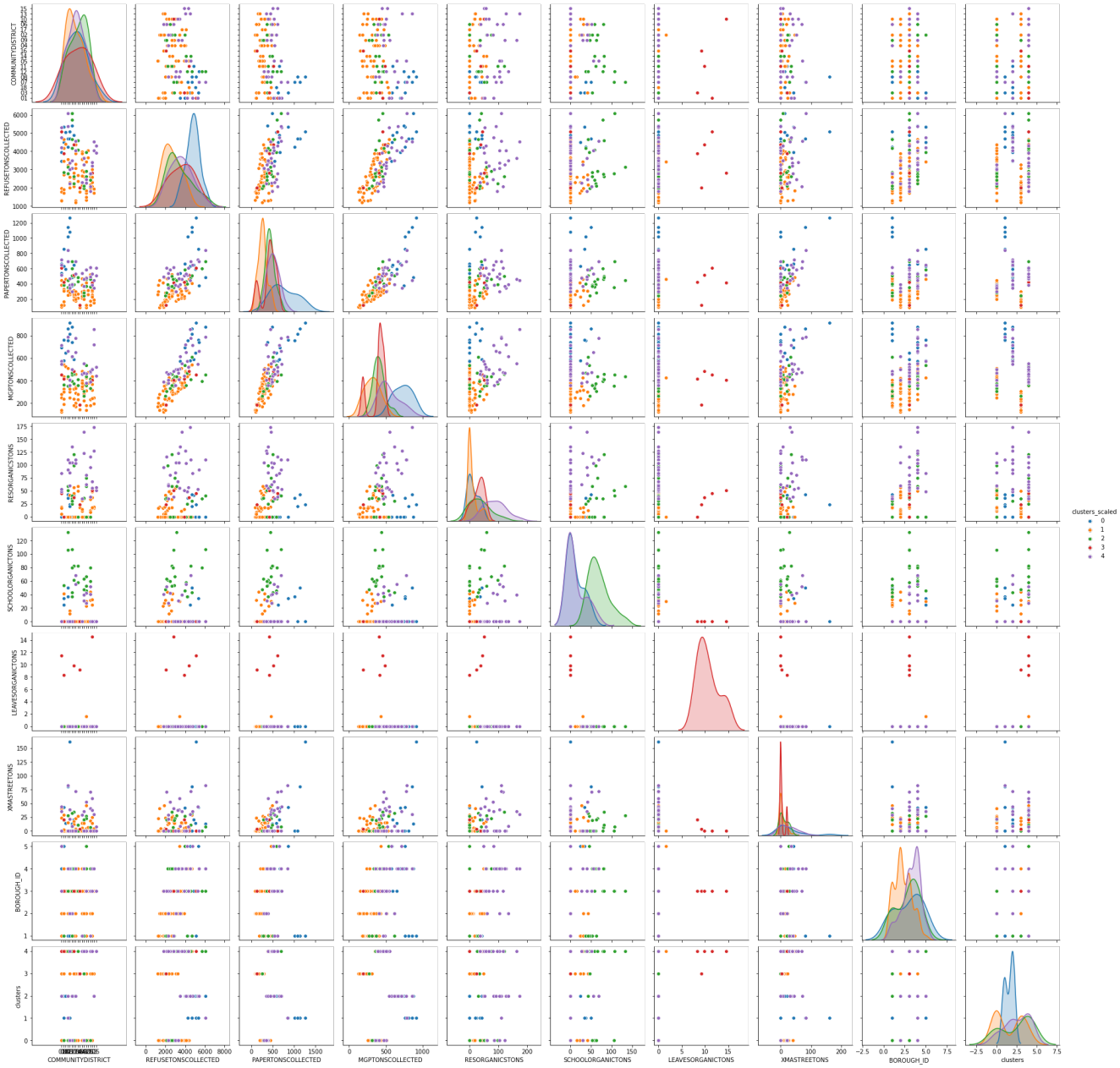
```
sns.pairplot(NYC2, hue = "clusters_scaled")
```

```
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
```

```
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must ha
ve variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
C:\Users\steve\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default
bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
```

Out[102]:

```
<seaborn.axisgrid.PairGrid at 0x27b9aa8a160>
```



In [111]:

```
cluster_scaled_map = {"0":"Queens", "1":"Brooklyn", "2":"Staten Island", "3":"Bronx", "4":"Manhatta
n"}
NYC2["mapped_clusters_scaled"] = NYC2["clusters_scaled"].apply(str).map(cluster_scaled_map)
NYC2.head()
```

```
<ipython-input-111-e75cdb2d6f1c>:2: 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
  NYC2["mapped_clusters_scaled"] = NYC2["clusters_scaled"].apply(str).map(cluster_scaled_map)
```

Out[111]:

	MONTH	BOROUGH	COMMUNITYDISTRICT	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORG
511	2020 / 01	Queens	01	3935.7	662.1	696.1	
627	2020 / 02	Brooklyn	05	3871.4	212.0	343.6	
815	2020 / 01	Staten Island	02	3946.9	586.0	534.8	
906	2020 / 02	Brooklyn	14	4062.5	429.5	360.5	
983	2020 / 01	Queens	02	2927.0	461.1	438.2	

In [ ]:

```
# Correlation, causation, and heat maps
```

In [152]:

```
NYC2.corr()
```

Out[152]:

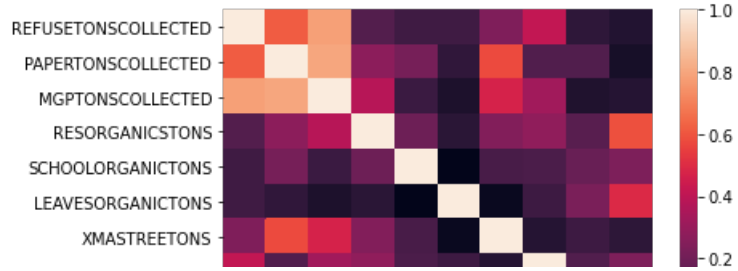
	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORGANICSTONS	SCHOOL
REFUSETONSCOLLECTED	1.000000	0.616804	0.784573	0.122240	
PAPERTONSCOLLECTED	0.616804	1.000000	0.802113	0.274381	
MGPTONSCOLLECTED	0.784573	0.802113	1.000000	0.382142	
RESORGANICSTONS	0.122240	0.274381	0.382142	1.000000	
SCHOOLORGANICTONS	0.064230	0.217337	0.051459	0.189335	
LEAVESORGANICTONS	0.065371	0.019029	-0.037174	0.000350	
XMASTREETONS	0.237746	0.576346	0.470813	0.248555	
BOROUGH_ID	0.414455	0.116527	0.327866	0.283357	
clusters	0.014794	0.115484	-0.029883	0.133505	
clusters_scaled	-0.018963	-0.053365	-0.010390	0.591980	

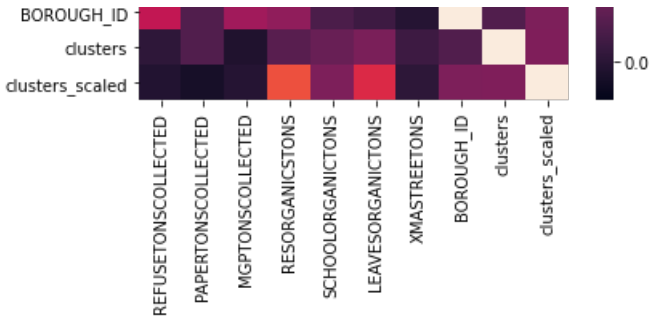
In [150]:

```
corr_matrix = NYC2.corr()
sns.heatmap(corr_matrix)
```

Out[150]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x27b9a51c670>
```





In [153]:

```
NYC.corr()
```

Out[153]:

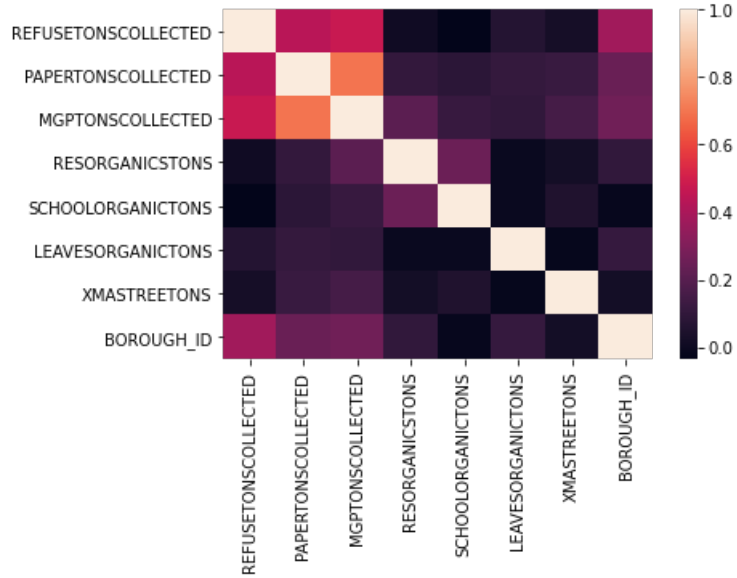
	REFUSETONSCOLLECTED	PAPERTONSCOLLECTED	MGPTONSCOLLECTED	RESORGANICSTONS	SCHOOL
REFUSETONSCOLLECTED	1.000000	0.436293	0.477313	0.009548	
PAPERTONSCOLLECTED	0.436293	1.000000	0.700245	0.105411	
MGPTONSCOLLECTED	0.477313	0.700245	1.000000	0.211602	
RESORGANICSTONS	0.009548	0.105411	0.211602	1.000000	
SCHOOLORGANICTONS	-0.034091	0.083622	0.119135	0.251629	
LEAVESORGANICTONS	0.065504	0.107765	0.101338	-0.008753	
XMASTREETONS	0.024222	0.120995	0.151805	0.018633	
BOROUGH_ID	0.378589	0.246063	0.262113	0.100164	

In [154]:

```
corr_matrix2 = NYC.corr()
sns.heatmap(corr_matrix2)
```

Out[154]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x27b9a676400>
```



In [ ]: