SS Cityscape properties

June 9, 2022

In this notebook, I examine data on all non tax-exempt properties in one of the community areas from Chicago Cityscape. To investigate how many properties are owned by owners living outside the community, I match street addresses of properties with the taxpayer addresses. This allows us to understand how many properties are owned by community members versus investors from outside the community.

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

[2]:
```
d1 = pd.read_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
 ↪drive-download-20220331T030638Z-001/Cityscape_ SS Nonexempt 1_5.csv')
d2 = pd.read_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
 ↪drive-download-20220331T030638Z-001/Cityscape_ SS Nonexempt 2_5.csv')
d3 = pd.read_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
 ↪drive-download-20220331T030638Z-001/Cityscape_ SS Nonexempt 3_5.csv')
d4 = pd.read_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
 ↪drive-download-20220331T030638Z-001/Cityscape_ SS Nonexempt 4_5.csv')
d5 = pd.read_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
 ↪drive-download-20220331T030638Z-001/Cityscape_ SS Nonexempt 5_5.csv')
```

Combine separate datasets to form a complete list of properties in South Shore.

[3]:
```
total = d1.append(d2).append(d3).append(d4).append(d5)
```

[89]:
```
total.columns
```

[89]:
```
Index(['PIN', 'PIN with dashes', 'Address', 'Address number', 'Address street',
       'ZIP Code', 'Chicago-owned property', 'Exempt', 'Property Class',
       'Property Description', 'Bill 2009', 'Bill 2010', 'Bill 2011',
       'Bill 2012', 'Bill 2013', 'Bill 2014', 'Bill 2015', 'Bill 2016',
       'Bill 2017', 'Bill 2018', 'Bill 2019', 'Bill 2020', 'Bill 2021',
       'Assessment 2009', 'Assessment 2010', 'Assessment 2011',
       'Assessment 2012', 'Assessment 2013', 'Assessment 2014',
       'Assessment 2015', 'Assessment 2016', 'Assessment 2017',
       'Assessment 2018', 'Assessment 2019', 'Assessment 2020',
```

```
       'Assessment 2021', 'Units', 'Area (sq. ft.)', 'Area (acres)',
       'Lot Size (s.f.)', 'Property Tax Year', 'Taxpayer Name',
       'Taxpayer Address', 'Taxpayer City, State ZIP',
       'Latest Assessment Year', 'Assessment Pass (First/Certified)',
       'Building Age', 'Zoning', 'Distance (feet)', 'QCT Tract', 'Photo URL',
       'Vacancy reason'],
      dtype='object')
```

[4]:
```
total['Address number']
```

[4]:
```
0       6720.0
1       6722.0
2       6724.0
3       6726.0
4       6728.0
         ...
1485    7819.0
1486    7819.0
1487    7819.0
1488    7819.0
1489       NaN
Name: Address number, Length: 11490, dtype: float64
```

To prepare the dataset for processing, I change the datatype of entries, eliminate parts of each string under a column, fill NaNs, and drop columns irrelevant to analysis.

[14]:
```
# change the street number to string, then eliminate decimals
total['Address number'] =total['Address number'].astype('str').map(lambda x:␣
 ↪str(x)[:-2])
# eliminate NANs
total.fillna('none', inplace = True)
total_w_add = total.loc[total['Address number']!='none']
# drop some columns that are irrelevant to analysis
total_w_add = total_w_add.drop(columns = ['PIN with dashes', 'Chicago-owned␣
 ↪property', 'Exempt','Zoning','Distance (feet)', 'QCT Tract', 'Photo URL',␣
 ↪'Assessment Pass (First/Certified)',
                                          'Latest Assessment Year'])
```

Next, I match the street addresses of the properties with the taxpayer addresses.

[18]:
```
# Add a new column indicating whether the taxpayer address matches the property␣
 ↪address
total_w_add['T/F'] = total_w_add.apply(lambda x: x['Address number'] in␣
 ↪x['Taxpayer Address']
                                        and x['ZIP Code'] in x['Taxpayer City,␣
 ↪State ZIP'], axis = 1)
```

[20]:
```
# matched 1 is the new dataset containing all properties where the taxpayer␣
 ↪lives at the property,
# with a new column indicating whether the street address matches the taxpayer␣
 ↪address.
matched1 = total_w_add.loc[total_w_add['T/F']==True]
# matched 2 is the new dataset containing all properties where the taxpayer does␣
 ↪not live at the property
unmatched1 = total_w_add.loc[total_w_add['T/F']==False]
```

[28]:
```
print(len(matched1), 'number of properties which the taxpayer lives at the␣
 ↪address.',
      len(unmatched1), 'number of properties which the taxpayer does not live at␣
 ↪the address.')
```

```
5623 number of properties which the taxpayer lives at the address. 5867 number
of properties which the taxpayer does not live at the address.
```

[44]:
```
# Check the outside owners who own the most properties (top 15 owners and the␣
 ↪number of properties they own)
owner2 = unmatched1.groupby(['Taxpayer Name']).count().nlargest(15,'PIN')
owner2['PIN']
```

[44]:
```
Taxpayer Name
none                        209
SHORELINE APARTMENTS R       49
MRC HOLDINGS LLC             35
NEXTCITY VETERANS DEV        32
THE MAYFAIR HABITAT GR       31
7020 7028 CREGIER RESI       28
JEAN RE2 AN ILLINOIS         28
KMART CORP PROP TAX          27
PARKWAYS PRESERVATION        25
VENTUS COLES 74 AN ILL       25
MIDDLETON REALTY GROUP       21
CHICAGO TITLE LAND TRU       19
TAXPAYER OF                  19
KALABICH MANAGEMENT          18
ARNOLD BANKS                 17
Name: PIN, dtype: int64
```

[119]:
```
# Save the two datasets (addressed matched and unmatched) to 2 separate csv
matched1.to_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
 ↪Taxpayer Addresses Matched.csv')
unmatched1.to_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape␣
 ↪data/Taxpayer Addresses Unmatched.csv')
```

[46]:
```
# From the original CityScape data, compile a list of condos
condos1 = total_w_add.loc[total_w_add['Property Description'] =='Residential␣
↪condominium']
condos2 = total_w_add.loc[total_w_add['Property Description'] =='Rental␣
↪condominium']
condos = condos1.append(condos2)
```

[50]:
```
# There are a total of 3387 condos
len(condos)
condos.to_csv('/Users/bling/Documents/21-22/Housing policy/SS Cityscape data/
↪Condos.csv')
```

[52]:
```
condo_owner = condos.groupby('Taxpayer Name').count()
# Find the top 10 condo owners in South Shore and how many condos they own
condo_owner.nlargest(10,'PIN')['PIN']
```

[52]:
```
Taxpayer Name
SHORELINE APARTMENTS R    49
TAXPAYER OF               37
MRC HOLDINGS LLC          36
NEXTCITY VETERANS DEV     32
THE MAYFAIR HABITAT GR    31
7020 7028 CREGIER RESI    28
JEAN RE2 AN ILLINOIS      28
VENTUS COLES 74 AN ILL    25
MIDDLETON REALTY GROUP    20
ARNOLD BANKS              17
Name: PIN, dtype: int64
```

[57]:
```
# How many owners living outside the community own the most condos?
# Find the top 10 owners who do not live in the community
condos_liv = condos.loc[condos['T/F']==False]
condo_owner2= condos_liv.groupby('Taxpayer Name').count()
condo_owner2.nlargest(10, 'PIN')['PIN']
```

[57]:
```
Taxpayer Name
SHORELINE APARTMENTS R    49
MRC HOLDINGS LLC          35
NEXTCITY VETERANS DEV     32
THE MAYFAIR HABITAT GR    31
7020 7028 CREGIER RESI    28
JEAN RE2 AN ILLINOIS      28
VENTUS COLES 74 AN ILL    25
MIDDLETON REALTY GROUP    20
ARNOLD BANKS              17
CLO INVESTMENTS LLC       14
Name: PIN, dtype: int64
```