Schools

November 13, 2019

[1]:
```
import pandas as pd
from sqlalchemy import create_engine
```

1 PROJECT PLAN

Project: merge CPS School Data of 660 schools (from egov.cityofchicago.org) with student demographic composition with a file of 440 affordable housing construction projects from www.johnsnowlabs.com. We plan to join these on zip code to see if there's a correlation of neighborhoods with lower income students and affordable housing construction.

ETL: We will join these two CSV's on zip code. We will need to aggregate the number of housing units built in a specific zip code. We will need to aggregate schools & student demographic composition per zip code.

[2]:
```
schools_csv = "Schools.csv"
schools_df = pd.read_csv(schools_csv)
schools_df.head()
```

[2]:
```
   School_ID  Legacy_Unit_ID  Finance_ID           Short_Name  \
0     610163            5770       30081                STOCK
1     610558            9598       46611             GOODE HS
2     609750            1750       49051           SIMPSON HS
3     610571            9636       65015  OMBUDSMAN - WEST HS
4     610123            5370       24911                 PENN

                            Long_Name      School_Type Primary_Category  \
0     Frederick Stock Elementary School     Neighborhood               ES
1           Sarah E. Goode STEM Academy  Citywide-Option               HS
2  Simpson Academy HS for Young Women  Citywide-Option               HS
3               Ombudsman Chicago- West  Citywide-Option               HS
4      William Penn Elementary School     Neighborhood               ES

  Is_High_School Is_Middle_School Is_Elementary_School  ...  \
0              N                N                    N  ...
1              Y                N                    N  ...
2              Y                Y                    Y  ...
3              Y                N                    N  ...
4              N                Y                    Y  ...
```

```
  Third_Contact_Name Fourth_Contact_Title Fourth_Contact_Name  \
0                NaN                  NaN                 NaN
1                NaN                  NaN                 NaN
2         Rita Somen                  NaN                 NaN
3                NaN                  NaN                 NaN
4                NaN                  NaN                 NaN

  Fifth_Contact_Title Fifth_Contact_Name Sixth_Contact_Title  \
0                 NaN                NaN                 NaN
1                 NaN                NaN                 NaN
2                 NaN                NaN                 NaN
3                 NaN                NaN                 NaN
4                 NaN                NaN                 NaN

  Sixth_Contact_Name Seventh_Contact_Title Seventh_Contact_Name  \
0                NaN                   NaN                  NaN
1                NaN                   NaN                  NaN
2                NaN                   NaN                  NaN
3                NaN                   NaN                  NaN
4                NaN                   NaN                  NaN

                                            Location
0  7507 W BIRCHWOOD AVE\nChicago, Illinois 60631\...
1  7651 S HOMAN AVE\nChicago, Illinois 60652\n(41...
2  1321 S PAULINA ST\nChicago, Illinois 60608\n(4...
3  2401 W CONGRESS PKWY\nChicago, Illinois 60612\...
4  1616 S AVERS AVE\nChicago, Illinois 60623\n(41...

[5 rows x 91 columns]
```

2 We only need "Long Name", "Student_Count_Total", "Student_Count_Low_Income", and "Zip", but we'll select a few more fields

for future analysis.

[3]:
```
new_schools_df = schools_df[['Long_Name', 'School_Type',␣
↪'Primary_Category','Zip','Student_Count_Total',␣
↪'Student_Count_Low_Income','Student_Count_Special_Ed','Student_Count_English_Learners','Stu
↪'Student_Count_Multi','Overall_Rating']].copy()
new_schools_df.head()
```

[3]:
```
                             Long_Name      School_Type Primary_Category  \
0   Frederick Stock Elementary School     Neighborhood               ES
1         Sarah E. Goode STEM Academy  Citywide-Option               HS
2  Simpson Academy HS for Young Women  Citywide-Option               HS
```

```
3            Ombudsman Chicago- West  Citywide-Option               HS
4      William Penn Elementary School      Neighborhood               ES

     Zip  Student_Count_Total  Student_Count_Low_Income  \
0  60631                  232                        37
1  60652                  900                       788
2  60608                   38                        37
3  60612                  341                       320
4  60623                  311                       279

   Student_Count_Special_Ed  Student_Count_English_Learners  \
0                        90                              27
1                       153                              57
2                         6                               2
3                        57                              31
4                        78                              13

   Student_Count_Black  Student_Count_Hispanic  Student_Count_White  \
0                    1                      39                  175
1                  459                     420                    7
2                   28                       8                    2
3                  187                     148                    4
4                  283                      26                    1

   Student_Count_Asian  Student_Count_Native_American  Student_Count_Multi  \
0                   16                              0                    0
1                    2                              5                    6
2                    0                              0                    0
3                    0                              1                    1
4                    0                              1                    0

      Overall_Rating
0  Inability to Rate
1           Level 1+
2            Level 2
3            Level 2
4           Level 1+
```

[4]:
```
housing_csv = "Housing.csv"
housing_df = pd.read_csv(housing_csv)
housing_df.head()
```

[4]:
```
  Community_Area_Name  Community_Area_Number Property_Description  \
0        Portage Park                     15                  ARO
1      West Englewood                     67          Multifamily
2           Englewood                     68          Multifamily
3     Washington Park                     40       Senior HUD 202
4       Humboldt Park                     23          Multifamily
```

```
              Property_Name                     Address  ZIP_Code  \
0  4812-15 W. Montrose Apts.  4812-15 W. Montrose Ave.     60641
1   New West Englewood Homes           2109 W. 63rd St.     60636
2            Antioch Homes II      301 W. Marquette Road     60621
3        St. Edmund's Corners      5556 S. Michigan Ave.     60637
4        Nelson Mandela Apts.            526 N. Troy St.     60624

   Phone_Number                   Management_Company  Units   Latitude  \
0  630-694-6968                          @properties      2        NaN
1  773-434-4929             Interfaith Housing Corp.     12        NaN
2  773-994-4546  Universal Management Service, Inc.     69  41.772564
3  773-667-7583    St. Edmund's Redevelopment Corp.     53  41.792975
4  773-227-6332                     Bickerdike Apts.      6  41.891173

   Longitude
0        NaN
1        NaN
2 -87.632419
3 -87.622569
4 -87.705338
```

3 The housing.csv is clean already so we'll go straight to aggregating the number of housing units per zip code and

converting the tuple to a dataframe.

[13]:
```
housing_zip = housing_df.groupby(['ZIP_Code'])['Units']
housing_zip_count = housing_zip.sum()
housing_zip_df = pd.DataFrame(housing_zip_count).reset_index()
housing_zip_df = housing_zip_df.rename(columns={"ZIP_Code":"zip",'Units':
 ↪'units'})
housing_zip_df.head()
# housing_zip_count.dtypes
```

[13]:

	zip	units
0	60601	16
1	60605	276
2	60607	233
3	60608	1022
4	60609	1207

4 We'll now aggregate the total students and total low-income students by zip code.

[6]:
```
students_zip = new_schools_df.
 ↪groupby(['Zip'])['Student_Count_Total','Student_Count_Low_Income']
students_zip_count = students_zip.sum()
students_zip_count.head()
```

[6]:

Zip	Student_Count_Total	Student_Count_Low_Income
60602	1326	1142
60605	2645	975
60607	5477	2358
60608	11009	9798
60609	12972	11467

5 We calculate the low-income-percentage-composition of the zip codes and add that percentage to the dataframe

[7]:
```
students_zip_count['low_inc_percent'] =␣
 ↪100*students_zip_count['Student_Count_Low_Income']/
 ↪students_zip_count['Student_Count_Total']
students_zip_count.head()
```

[7]:

Zip	Student_Count_Total	Student_Count_Low_Income	low_inc_percent
60602	1326	1142	86.123680
60605	2645	975	36.862004
60607	5477	2358	43.052766
60608	11009	9798	88.999909
60609	12972	11467	88.398088

6 Connect to PostGres

[8]:
```
rds_connection_string = "postgres:postgres@localhost:5432/chicago"
engine = create_engine(f'postgresql://{rds_connection_string}')
```

[9]:
```
engine.table_names()
```

[9]: ['housing', 'schools']

7 We reset_index of the students_zip dataframe so we can join to the no-index housing_zip dataframe in SQL.

We also renamed all our fields to lower case since PostGres would automatically make them lower case and this allows our dataframe to match our tables in Postgres.

[10]:
```
students_zip_count.reset_index(level=0, inplace=True)
students_zip_count = students_zip_count.rename(columns={'Zip':
↪'zip','Student_Count_Total':'student_count_total','Student_Count_Low_Income':
↪'student_count_low_income', 'low_inc_percent':'low_inc_perc'})
students_zip_count.head()
```

[10]:

	zip	student_count_total	student_count_low_income	low_inc_perc
0	60602	1326	1142	86.123680
1	60605	2645	975	36.862004
2	60607	5477	2358	43.052766
3	60608	11009	9798	88.999909
4	60609	12972	11467	88.398088

[11]:
```
students_zip_count.to_sql(name='schools', con=engine, if_exists='append',␣
↪index=False)
```

[14]:
```
housing_zip_df.to_sql(name='housing', con=engine, if_exists='append',␣
↪index=False)
```

[ ]:
```
students_zip_count.to_sql(name='schools', con=engine, if_exists='append',␣
↪index=False)
```

8 Here, we query our PostGres tables to confirm that our dataframe exports to sql worked.

[15]:
```
pd.read_sql_query('select * from schools', con=engine).head()
```

[15]:

	zip	student_count_total	student_count_low_income	low_inc_perc
0	60602	1326	1142	86.123680
1	60605	2645	975	36.862004
2	60607	5477	2358	43.052766
3	60608	11009	9798	88.999909
4	60609	12972	11467	88.398088

[16]:
```
pd.read_sql_query('select * from housing', con=engine).head()
```

[16]:

	zip	units
0	60601	16
1	60605	276
2	60607	233
3	60608	1022
4	60609	1207

9 Our final step is to join the tables on 'zip' and query all the zip codes.

[19]:
```
pd.read_sql_query('select * from schools inner join housing on schools.zip = housing.zip order by housing.units desc', con=engine)
```

[19]:

	zip	student_count_total	student_count_low_income	low_inc_perc	zip \
0	60653	5271	4456	84.538038	60653
1	60624	8044	7121	88.525609	60624
2	60612	9016	7676	85.137533	60612
3	60609	12972	11467	88.398088	60609
4	60637	7291	6053	83.020162	60637
5	60608	11009	9798	88.999909	60608
6	60616	7116	5752	80.831928	60616
7	60647	8937	7298	81.660512	60647
8	60640	2626	2152	81.949733	60640
9	60628	9006	7696	85.454142	60628
10	60621	5416	4888	90.251108	60621
11	60660	5134	3865	75.282431	60660
12	60622	7218	4986	69.077307	60622
13	60644	5627	5040	89.568154	60644
14	60626	3739	3209	85.825087	60626
15	60613	6108	3597	58.889980	60613
16	60623	17587	15930	90.578268	60623
17	60629	18386	16172	87.958229	60629
18	60620	9361	8029	85.770751	60620
19	60615	4475	3249	72.603352	60615
20	60617	12947	11198	86.491079	60617
21	60646	2180	625	28.669725	60646
22	60649	3341	2959	88.566298	60649
23	60610	4442	2214	49.842413	60610
24	60619	7338	6134	83.592259	60619
25	60605	2645	975	36.862004	60605
26	60639	13141	11647	88.631002	60639
27	60607	5477	2358	43.052766	60607
28	60618	16160	10024	62.029703	60618
29	60659	5644	4215	74.681077	60659
30	60657	3139	819	26.091112	60657
31	60638	6838	4661	68.163206	60638
32	60630	3236	1930	59.641533	60630
33	60641	9683	7866	81.235154	60641
34	60643	8947	7083	79.166201	60643
35	60634	11457	8432	73.596928	60634
36	60633	960	713	74.270833	60633
37	60642	4482	3564	79.518072	60642
38	60632	24088	21579	89.584025	60632
39	60614	5942	2190	36.856277	60614
40	60636	5994	5028	83.883884	60636

41	60652	7347	6164	83.898190	60652
42	60631	6278	2555	40.697674	60631
43	60651	6921	6165	89.076723	60651
44	60627	479	443	92.484342	60627
45	60661	148	131	88.513514	60661
46	60625	12042	8818	73.227039	60625
47	60707	995	787	79.095477	60707
48	60645	4789	3720	77.678012	60645

	units
0	3071
1	1544
2	1243
3	1207
4	1081
5	1022
6	1007
7	909
8	827
9	758
10	708
11	596
12	575
13	564
14	517
15	502
16	492
17	489
18	470
19	450
20	393
21	380
22	380
23	349
24	296
25	276
26	248
27	233
28	223
29	218
30	216
31	172
32	170
33	170
34	152
35	136
36	116

37	112
38	108
39	107
40	96
41	85
42	84
43	80
44	75
45	61
46	60
47	17
48	3