Census "hard to count" analysis

By Ben Welsh (https://palewi.re/who-is-ben-welsh/)

This data preparation routine was developed for the April 29, 2019, Los Angeles Times story "A census undercount could cost California billions — and L.A. is famously hard to track" (https://www.latimes.com/local/lanow/la-me-la-county-census-hard-to-count-20190429-htmlstory.html).

It combines the California Department of Finance's "hard to count" estimates with the tract maps published by the U.S. Census Bureau. Together they were used to make a graphic to accompany the story.

How we did it

Download the state's "hard to count" esimates.

In [22]:
```
!python download.py htc
```

Download the Census Bureau's tract maps.

In [24]:
```
!python download.py tracts
```

Import Python tools

In [1]:
```
import pandas as pd
import geopandas as gpd
```

Read in the hard-to-count data

In [2]:
```
df = pd.read_excel("./data/htc/tracts.xlsx", dtype={"GEOID": str})
```

In [3]:
```
df.head()
```

Out[3]:

	GEOID	CA HTC Index
0	06001400100	20
1	06001400200	16
2	06001400300	31
3	06001400400	35
4	06001400500	47

Clean it up.

In [4]:
```
df_trimmed = df.rename(columns={
    "GEOID": "geoid",
    "CA HTC Index": "htc_index"
})
```

In [5]:
```
df_trimmed.head()
```

Out[5]:

	geoid	htc_index
0	06001400100	20
1	06001400200	16
2	06001400300	31
3	06001400400	35
4	06001400500	47

Read in the tract maps.

In [6]:
```
gdf = gpd.read_file("data/tracts/tl_2010_06_tract10.shp")
```

In [25]:
```
gdf.head()
```

Out[25]:

	STATEFP10	COUNTYFP10	TRACTCE10	GEOID10	NAME10	NAMELSAD10	MTFCC10	FU
0	06	083	002103	06083002103	21.03	Census Tract 21.03	G5020	
1	06	083	002402	06083002402	24.02	Census Tract 24.02	G5020	
2	06	083	002102	06083002102	21.02	Census Tract 21.02	G5020	
3	06	083	002010	06083002010	20.10	Census Tract 20.10	G5020	
4	06	083	002009	06083002009	20.09	Census Tract 20.09	G5020	

Clean it up.

In [7]:
```
gdf_trimmed = gdf[[
    'GEOID10',
    'geometry'
]].rename(columns={
    "GEOID10": "geoid",
})
```

In [8]:
```
gdf_trimmed.head()
```

Out[8]:

	geoid	geometry
0	06083002103	POLYGON ((-120.417938 34.938341, -120.417658 3...
1	06083002402	POLYGON ((-120.473893 34.920814, -120.474285 3...
2	06083002102	POLYGON ((-120.417658 34.938345, -120.417938 3...
3	06083002010	POLYGON ((-120.411468 34.879619, -120.411413 3...
4	06083002009	POLYGON ((-120.423524 34.879283, -120.422856 3...

Merge the data and the map

In [9]:
```
merged_gdf = gdf_trimmed.merge(df_trimmed, on="geoid", how="inner")
```

Output the merged file for a graphic

In [11]:
```
merged_gdf.to_file("data/processed/tracts.shp")
```

How many of the hardest to count are here in LA County?

In [12]:
```
merged_gdf['county_fips'] = merged_gdf.geoid.str.slice(2, 5)
```

In [17]:
```
merged_gdf.county_fips.value_counts().head()
```

Out[17]:
```
037    2345
073     628
059     583
065     453
085     372
Name: county_fips, dtype: int64
```

In [14]:
```
top_100 = merged_gdf.sort_values("htc_index", ascending=False).head(100)
```

In [18]:
```
top_100.head()
```

Out[18]:

	geoid	geometry	htc_index	county_fips
3473	06077000100	POLYGON ((-121.292051 37.95407, -121.293315 37...	136	077
7102	06037212305	POLYGON ((-118.2998 34.057707, -118.29871 34.0...	128	037
6966	06037209300	POLYGON ((-118.271663 34.053097, -118.2714 34....	127	037
7270	06037231710	POLYGON ((-118.287222 34.010102, -118.28722 34...	123	037
3472	06077000300	POLYGON ((-121.292051 37.95407, -121.291901 37...	123	077

In [19]:
```
top_100.county_fips.value_counts()
```

Out[19]:
```
037    57
077     7
019     5
075     5
071     5
073     4
025     3
067     3
029     3
053     2
001     2
059     1
095     1
047     1
099     1
Name: county_fips, dtype: int64
```