**ETL Report**

Harry Xing / Jaime Carrillo

The team performed two **ETL**s to correlate COVID-19 cases with socioeconomic data in Santa Clara County and a third ETL to extract ethnicity distribution in the same county.

**ETL-1**

**E\_xtract**

The sources of data:

* Census data was extracted from a JSON API at <https://www.census.gov/data/developers/updates/new-discovery-tool.html> (<https://api.census.gov/data/2018/acs/acs5/cprofile>)

[{'B19013\_001E': 13092.0,

'B01003\_001E': 17242.0,

'B01002\_001E': 40.5,

'B19301\_001E': 6999.0,

'B17001\_002E': 10772.0,

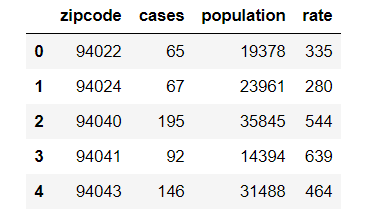
'B23025\_004E': 3495.0,

'B23025\_002E': 5811.0,

'B23025\_005E': 2316.0,

'zip code tabulation area': '00601'},

* COVID-19 data was extracted from the Santa Clara County in JSON format- <https://data.sccgov.org/resource/j2gj-bg6c.json>



*Requests* functions were performed to extract census and COVID-19 data.

**T\_ransform**

Data was transformed by cleaning, filtering and joining primarily

* The “*census\_zip”* dataframe was transformed by renaming columns

*census\_zip = census\_zip.rename(columns={*

*"B01003\_001E": "Population",*

*"B01002\_001E": "Median Age",*

*"B19013\_001E": "Household Income",*

*"B19301\_001E": "Per Capita Income",*

*"B17001\_002E": "Poverty Count",*

*"B23025\_004E": "employment\_employed",*

*"B23025\_002E": "Labor\_force",*

*"B23025\_005E": "employment\_unemployed",*

*"zip code tabulation area": "zipcode"})*

* and by dropping NaN (dropna). - *census\_zip = census\_zip.dropna()*
* In the “*santa\_clara\_covid\_zipcode”* table, the “cases” column was renamed to “infected count” –

*santa\_clara\_covid\_zipcode = santa\_clara\_covid\_zipcode.rename(\*

*columns={"cases": "infected count"*

*})*

* empty rows were filled with ‘0’ (fillna),

*santa\_clara\_covid\_zipcode['infected count'] = \*

*santa\_clara\_covid\_zipcode['infected count'].fillna(0)*

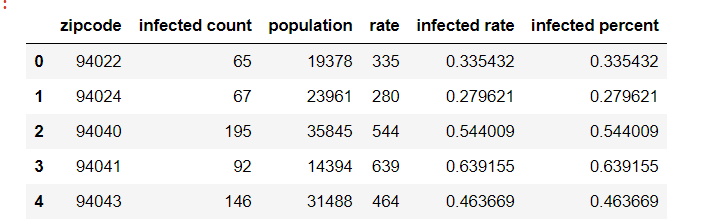
* and a new column; “infected percent,” was inserted and percentages calculated.

*santa\_clara\_covid\_zipcode["infected percent"] =\*

*100 \* santa\_clara\_covid\_zipcode["infected count"].astype(int)/santa\_clara\_covid\_zipcode["population"].astype(int)*

* *“santa\_clara\_covid\_zipcode”* and “*census\_zip”* were merged with a ‘left’ join into the “*santa\_clara\_covid\_demographics”*

*santa\_clara\_covid\_demographics = pd.merge(santa\_clara\_covid\_zipcode, census\_zip, how="left", on = ["zipcode","zipcode"])*



**L\_oad**

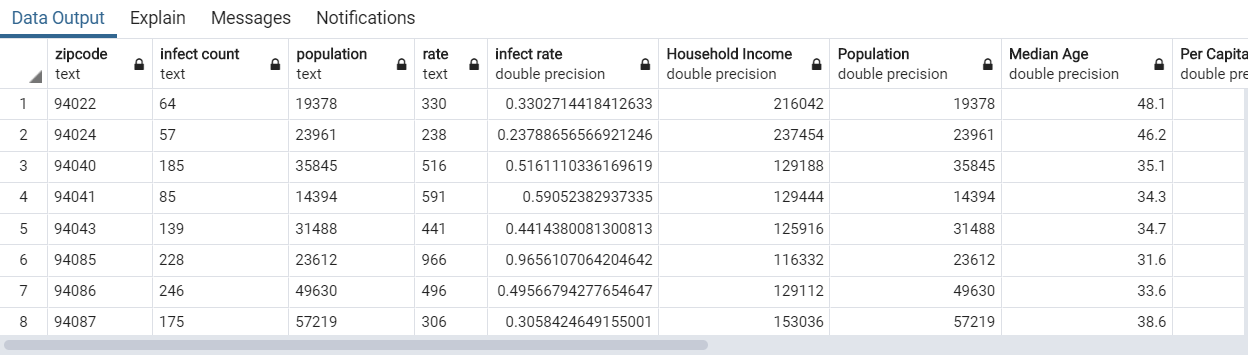
Data was loaded to postgresql database

* local connection (create\_engine) to postgresql was used to create *covid19\_santa\_claraDB*

*engine = create\_engine(f'postgresql://postgres:@localhost/covid19\_santa\_claraDB')*

* *santa\_clara\_covid\_demographics* was loaded to the covid19\_santa\_claraDB as the “*covid19”* table (*relational)*

*santa\_clara\_covid\_demographics.to\_sql(name='covid19', con=engine, if\_exists='replace', index=False)*



**ETL-2**

**E\_xtract**

Source of data:

* Zip code, population, and density data for Santa Clara County was extracted by scraping: http://www.mapszipcode.com/california/ladera%20ranch/ + zipcode

*response = requests.get(base\_url + zipcode)*

*soup = bs(response.text, 'html.parser')*

*results = soup.find\_all('div', class\_="dat")*

* family income was also scraping from http://www.mapszipcode.com/california/ladera%20ranch/' + zipcode + '/demographics/'

*response = requests.get('http://www.mapszipcode.com/california/ladera%20ranch/' + zipcodeDict['zipcode'] + '/demographics/')*

*#BeautifulSoup object*

*soup = bs(response.text, 'html.parser')*

*#Retrieve the latest subject and content from the Mars website*

*results = soup.find\_all('div', class\_="col span\_12\_of\_12")*

**T\_ransform**

* A list of dictionaries (“*zipcode\_list*”). One dictionary was created by scraping zip codes, population and density for Santa Clara County.

*zipcodes = santa\_clara\_covid\_zipcode['zipcode'].to\_list()*

* Text was split (text.split) to remove unnecessary text from results and using a *replace* function

*zipcodeDict["population"] = results[0].text.split(' ', 1 )[0]*

*zipcodeDict["density"] = results[1].text.split(' ', 1 )[0].replace('\t', '').replace('/', '')*

* Results were appended to the *zipcode\_list*

*zipcode\_list.append(zipcodeDict)*

[{'zipcode': '94022', 'population': '19,310', 'density': '1,104.17'},

{'zipcode': '94024', 'population': '22,536', 'density': '3,086.45'},

* Mapszipcode.com was also scraped to create the ‘family income’ dictionary.
* Results were cleaned by replacing ‘$’ with ‘USD’

*income[dictResult[0].contents[0].replace('$', 'USD')] = dictResult[1].contents[0]*

[{'zipcode': '94022',

'population': '19,310',

'density': '1,104.17',

'income': {'USD30,000 or less': '4.51%',

'USD30,000 and USD50,000': '5.67%',

'USD50,000 and USD100,000': '11.6%',

'USD100,000 and USD200,000': '28.45%',

'USD200,000 or more': '49.77%'}},

**L\_oad**

* *zipcode\_list* including results from the two scrapings were uploaded to MongoDB into a *demographics\_zip\_DB* database and populated a *demographics\_zip* collection. (non relational).

*{"\_id":{"$oid":"5f4f003b34ba69f8fdb5197e"},"zipcode":"94022","population":"19,310","density":"1,104.17","income":{"USD30,000 or less":"4.51%","USD30,000 and USD50,000":"5.67%","USD50,000 and USD100,000":"11.6%","USD100,000 and USD200,000":"28.45%","USD200,000 or more":"49.77%"}}*

**ETL-3**

**E\_xtract**

* Use a dataframe to request dataset from mapszipcode.com

*tables = pd.read\_html('http://www.mapszipcode.com/california/ladera%20ranch/'+ zipcode + '/demographics/')*

**T\_ransform**

**ETL-3**

* Rename column “Racial makeup” to “population

*df2.columns = ['Racial makeup', 'Population']*

* dataframe to extract the data and reverse from rows to columns

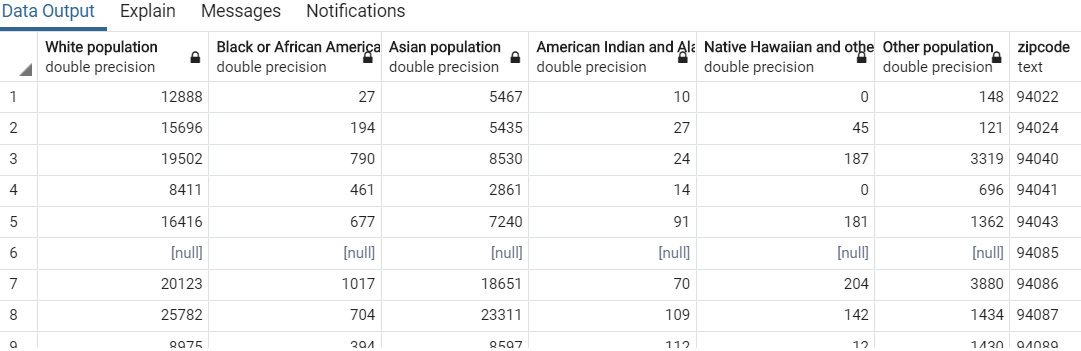
*dict1= (df2.to\_dict())['Population']*

* Data was cleaned by dropping NaN

*zipcode\_data.dropna()*

**L\_oad**

* Data was uploaded to a the covid19\_santa\_claraDB (postgresql database) in the *demographic\_zipcode* table.



We wanted to use PostgreSQL and mongoDB to practice what we learned in class.