```
 1 import requests
 2
 3 import numpy as np
 4 import pandas as pd
 5
 6 from requests.auth import HTTPBasicAuth
 7 from uszipcode import SearchEngine
 8
 9
10 def find_zip(latlong_list
   ):                                                  # Takes a list of
   tuples, form [(lat, long)]
11     try:
12         df = pd.read_hdf('.data_cache/US_long_lat_to_zip.h5', 'df'
   )         # Try reading hdf5 lat longs for USA
13     except FileNotFoundError:
14         df = pd.read_csv('../ref/US_long_lat_to_zip.csv', dtype=str
   )        # Read from CSV
15         df.to_hdf('./.data_cache/US_long_lat_to_zip.h5', 'df'
   )              # Store as hdf5 for future
16         print('caching...')
17     best_zips = []
18     for coord in latlong_list
   :                                                   # Loop through input
   list
19         lat = coord[0]
20         lon = coord[1]
21         best_zip = None
22         best_zip_distance = None
23         for index, row in df.iterrows
   ():                                       # Loop through USA latlongs
24             x1 = float(row['LAT'])
25             y1 = float(row['LNG'])
26             x_dif = (lat - x1)**2
27             y_diff = (lon - y1)**2
28             distance = np.sqrt(x_dif + y_diff)
29             if not best_zip_distance or distance < best_zip_distance:
30                 best_zip_distance = distance
31                 best_zip = row['ZIP']
32         best_zips.append(best_zip)
33     best_zips = list(set(best_zips))
34     return best_zips
35
36
37 def get_population(lat, lon):
38     request_url = "https://service.zipapi.us/population/zipcode/{zip
   }/" \
39                   "?X-API-KEY={key}&fields=male_population,
   female_population"
40     zip_ = list((lat, lon))
41     local_zip = find_zip(zip_)
42     key = '426151a147801b8aa34933bbd2c75abc'
43     submit_url = request_url.format(zip=local_zip, key=key)
```

```
44      usr = 'zacharyobrien2@my.unt.edu'
45      pas = 'ft2x8A!XmuY@XA6hk9xD*nsw'
46      r = requests.get(submit_url, auth=HTTPBasicAuth(usr, pas))
47      return local_zip, r.json()['data']['population']
48
49
50  def search_by_zip(zip_):
51      search = SearchEngine(simple_zipcode=True)  # set simple_zipcode=
    False to use rich info database
52      zipcode = search.by_zipcode(zip_)
53      return zipcode
54
55
56  def zips_unique_create():
57      try:
58          latlong_df = pd.read_hdf('.data_cache/data_latlong_table.h5',
    'df')
59      except FileNotFoundError:
60          traffic_df = pd.read_csv('../../bigquery-geotab-intersection-
    congestion/train.csv')
61          latlong_df = traffic_df[['IntersectionId', 'Latitude', '
    Longitude']].drop_duplicates('IntersectionId')
62          latlong_df.to_hdf('./.data_cache/data_latlong_table.h5', 'df')
63          print('caching...')
64
65      latlong_list = []
66      for _, row in latlong_df.iterrows():
67          latlong_list.append((row['Latitude'], row['Longitude']))
68
69      zips = pd.Series(find_zip(latlong_list))
70      zips.to_hdf('./.data_cache/dataset_zips.h5', 'df')
71      print('caching...')
72      return zips
73
74
75  def load_dataset_zips():
76      try:
77          zips = pd.read_hdf('.data_cache/dataset_zips.h5', 'df')
78      except FileNotFoundError:
79          zips = zips_unique_create()
80      return zips
81
```