BBER Data User's Conference - November 18, 2015 - Karl Benedict (UNM University Libraries, Director of Research Data Services)

Demonstrating Interaction with the US Census Bureau's Application Programming Interfaces

This demonstration will illustrate some strategies for interacting programmatically with the US Census Bureau's *Application Programming Interface* (API) accessible from http://www.census.gov/developers/. The specific APIs that will be demonstrated include:

- Census Geocoding Services - http://www.census.gov/data/developers/data-sets/Geocoding-services.html
- Decennial Census (2010, 2000, 1990) - http://www.census.gov/data/developers/data-sets/decennial-census-data.html
- Business Dynamics Statistics (1976-2013) - http://www.census.gov/data/developers/data-sets/business-dynamics.html

In order to use many of these APIs you must request and receive an API key from the Census Bureau - http://api.census.gov/data/key_signup.html. For this demonstration I will be using my key, but you must request your own to use these services.

While the examples demonstrated here use the Python programming language, interaction with the Census APIs can be automated using any modern programming language that supports interaction with remote web services (pretty much all of them).

Preliminary settings and loading of general programming libraries

```
%matplotlib inline

import requests # for interacting with remote web services published over HTTP

import matplotlib
import numpy as np
import matplotlib.pyplot as plt

myKey = '[redacted]' # replace this with your own
```

Census Geocoding Services

Census Geocoding Services allow for the submission of a street address for which an approximate Latitude-Longitude coordinate will be calculated and used to return the corresponding Census geography for that location.

The *Census Geocoding Services* API documentation PDF is found here - http://geocoding.geo.census.gov/geocoder/Geocoding_Services_API.pdf

Let's now build an address and obtain the Census geographies that correspond with that address

```
# Build the request
requestBase = "http://geocoding.geo.census.gov/geocoder/geographies/address"
params = {
    'street':'5600 Eagle Rock Ave',
    'city':'Albuquerque',
    'state':'NM',
    'zip':'87113',
    'benchmark':'Public_AR_Census2010', # Public Address Ranges - Census 2010 Benchmark
    'format':'json',
    'layers':'all',
    'vintage':'Census2010_Census2010' # Census 2010 Vintage - Census 2010 Benchmark
    }

# Some other addresses to try ...

# The Palace of the Governors in Santa Fe: 105 W Palace Ave, Santa Fe, NM 87501
params2 = {
    'street':'105 W Palace Ave',
    'city':'Santa Fe',
    'state':'NM',
    'zip':'87501',
    'benchmark':'Public_AR_Census2010',
    'format':'json',
    'layers':'all',
    'vintage':'Census2010_Census2010'
    }

# The Transamerica Pyramid in San Francisco, CA: 600 Montgomery St, San Francisco, CA 94111
params3 = {
    'street':'600 Montgomery St',
    'city':'San Francisco',
    'state':'CA',
    'zip':'94111',
    'benchmark':'Public_AR_Census2010',
    'format':'json',
```

```
    'layers':'all',
    'vintage':'Census2010_Census2010'
    }



# Submit the request to the US Census service
response = requests.get(requestBase,params3)
print(response.url)
print('')
print("The glossary for the attribute names for the output from the request can be found her
response.json()
```

```
http://geocoding.geo.census.gov/geocoder/geographies/address?layers=all&city=San+Francisco&z

The glossary for the attribute names for the output from the request can be found here - htt
```


```
{u'result': {u'addressMatches': [{u'addressComponents': {u'city': u'SAN FRANCISCO',
     u'fromAddress': u'600',
     u'preDirection': u'',
     u'preQualifier': u'',
     u'preType': u'',
     u'state': u'CA',
     u'streetName': u'Montgomery',
     u'suffixDirection': u'',
     u'suffixQualifier': u'',
     u'suffixType': u'St',
     u'toAddress': u'648',
     u'zip': u'94111'},
    u'coordinates': {u'x': -122.40312, u'y': 37.79469},
    u'geographies': {u'111th Congressional Districts': [{u'AREALAND': 92171832,
       u'AREAWATER': 202083306,
       u'BASENAME': u'8',
       u'CD111': u'08',
       u'CDSESSN': u'111',
       u'CENTLAT': u'+37.7871917',
       u'CENTLON': u'-122.4295366',
       u'FUNCSTAT': u'N',
       u'GEOID': u'0608',
       u'HU100': 320817,
       u'INTPTLAT': u'+37.7870773',
```

```
  u'INTPTLON': u'-122.4292519',
  u'LSADC': u'C2',
  u'MTFCC': u'G5200',
  u'NAME': u'Congressional District 8',
  u'OBJECTID': 6,
  u'OID': 21190355681088,
  u'POP100': 666827,
  u'STATE': u'06'}],
u'Alaska Native Regional Corporations': [],
u'Alaska Native Village Statistical Areas': [],
u'American Indian Joint-Use Areas': [],
u'Census Block Groups': [{u'AREALAND': 157738,
  u'AREAWATER': 0,
  u'BASENAME': u'1',
  u'BLKGRP': u'1',
  u'CENTLAT': u'+37.7956131',
  u'CENTLON': u'-122.4027375',
  u'COUNTY': u'075',
  u'FUNCSTAT': u'S',
  u'GEOID': u'060750611001',
  u'HU100': 668,
  u'INTPTLAT': u'+37.7956131',
  u'INTPTLON': u'-122.4027375',
  u'LSADC': u'BG',
  u'MTFCC': u'G5030',
  u'NAME': u'Block Group 1',
  u'OBJECTID': 32992,
  u'OID': 208903717106715,
  u'POP100': 993,
  u'STATE': u'06',
  u'TRACT': u'061100',
  u'UR': u''}],
u'Census Blocks': [{u'AREALAND': 14197,
  u'AREAWATER': 0,
  u'BASENAME': u'1013',
  u'BLKGRP': u'1',
  u'BLOCK': u'1013',
  u'CENTLAT': u'+37.7951953',
  u'CENTLON': u'-122.4024022',
  u'COUNTY': u'075',
  u'FUNCSTAT': u'S',
  u'GEOID': u'060750611001013',
  u'HU100': 0,
  u'INTPTLAT': u'+37.7951953',
  u'INTPTLON': u'-122.4024022',
  u'LSADC': u'BK',
```

```
  u'LWBLKTYP': u'L',
  u'MTFCC': u'G5040',
  u'NAME': u'Block 1013',
  u'OBJECTID': 1257572,
  u'OID': 210403968139441,
  u'POP100': 0,
  u'STATE': u'06',
  u'SUFFIX': u'',
  u'TRACT': u'061100',
  u'UR': u''}],
u'Census Designated Places': [],
u'Census Divisions': [{u'AREALAND': 2318780855476,
  u'AREAWATER': 296296354328,
  u'BASENAME': u'Pacific',
  u'CENTLAT': u'+56.3808919',
  u'CENTLON': u'-138.8776710',
  u'DIVISION': u'9',
  u'FUNCSTAT': u'S',
  u'GEOID': u'9',
  u'HU100': 19067795,
  u'INTPTLAT': u'+43.5008617',
  u'INTPTLON': u'-123.1351948',
  u'LSADC': u'69',
  u'MTFCC': u'G1200',
  u'NAME': u'Pacific Division',
  u'OBJECTID': 1,
  u'OID': 2739020719804,
  u'POP100': 49880102,
  u'REGION': u'4'}],
u'Census Regions': [{u'AREALAND': 4535207060164,
  u'AREAWATER': 316492173433,
  u'BASENAME': u'West',
  u'CENTLAT': u'+49.9388803',
  u'CENTLON': u'-124.1795441',
  u'FUNCSTAT': u'S',
  u'GEOID': u'4',
  u'HU100': 28592619,
  u'INTPTLAT': u'+35.2931746',
  u'INTPTLON': u'-111.9010442',
  u'LSADC': u'68',
  u'MTFCC': u'G1100',
  u'NAME': u'West Region',
  u'OBJECTID': 1,
  u'OID': 2729020719823,
  u'POP100': 71945553,
  u'REGION': u'4'}],
```

```
u'Census Tracts': [{u'AREALAND': 239862,
  u'AREAWATER': 0,
  u'BASENAME': u'611',
  u'CENTLAT': u'+37.7955998',
  u'CENTLON': u'-122.4040647',
  u'COUNTY': u'075',
  u'FUNCSTAT': u'S',
  u'GEOID': u'06075061100',
  u'HU100': 2351,
  u'INTPTLAT': u'+37.7955998',
  u'INTPTLON': u'-122.4040647',
  u'LSADC': u'CT',
  u'MTFCC': u'G5020',
  u'NAME': u'Census Tract 611',
  u'OBJECTID': 8886,
  u'OID': 207903717106348,
  u'POP100': 4307,
  u'STATE': u'06',
  u'TRACT': u'061100',
  u'UR': u''}],
u'Combined New England City and Town Areas': [],
u'Combined Statistical Areas': [{u'AREALAND': 22637782719,
  u'AREAWATER': 3755649611,
  u'BASENAME': u'San Jose-San Francisco-Oakland, CA',
  u'CENTLAT': u'+37.7324973',
  u'CENTLON': u'-122.1031953',
  u'CSA': u'488',
  u'FUNCSTAT': u'S',
  u'GEOID': u'488',
  u'HU100': 2908294,
  u'INTPTLAT': u'+37.7449044',
  u'INTPTLON': u'-122.1036069',
  u'LSADC': u'M0',
  u'MTFCC': u'G3100',
  u'NAME': u'San Jose-San Francisco-Oakland, CA CSA',
  u'OBJECTID': 70,
  u'OID': 26190141293320,
  u'POP100': 7468390}],
u'Consolidated Cities': [],
u'Counties': [{u'AREALAND': 121399963,
  u'AREAWATER': 479190317,
  u'BASENAME': u'San Francisco',
  u'CENTLAT': u'+37.7600830',
  u'CENTLON': u'-122.6941399',
  u'COUNTY': u'075',
  u'COUNTYCC': u'H6',
```

```
  u'COUNTYNS': u'00277302',
  u'FUNCSTAT': u'C',
  u'GEOID': u'06075',
  u'HU100': 376942,
  u'INTPTLAT': u'+37.7272391',
  u'INTPTLON': u'-123.0322294',
  u'LSADC': u'06',
  u'MTFCC': u'G4020',
  u'NAME': u'San Francisco County',
  u'OBJECTID': 3074,
  u'OID': 27590355701186,
  u'POP100': 805235,
  u'STATE': u'06',
  u'UR': u''}],
u'County Subdivisions': [{u'AREALAND': 121399963,
  u'AREAWATER': 479190317,
  u'BASENAME': u'San Francisco',
  u'CENTLAT': u'+37.7600830',
  u'CENTLON': u'-122.6941399',
  u'COUNTY': u'075',
  u'COUSUB': u'92790',
  u'COUSUBCC': u'Z5',
  u'COUSUBNS': u'01935284',
  u'FUNCSTAT': u'S',
  u'GEOID': u'0607592790',
  u'HU100': 376942,
  u'INTPTLAT': u'+37.7272391',
  u'INTPTLON': u'-123.0322294',
  u'LSADC': u'22',
  u'MTFCC': u'G4040',
  u'NAME': u'San Francisco CCD',
  u'OBJECTID': 33643,
  u'OID': 27690355723351,
  u'POP100': 805235,
  u'STATE': u'06',
  u'UR': u''}],
u'Elementary School Districts': [],
u'Estates': [],
u'Federal American Indian Reservations': [],
u'Hawaiian Home Lands': [],
u'Incorporated Places': [{u'AREALAND': 121399963,
  u'AREAWATER': 479190317,
  u'BASENAME': u'San Francisco',
  u'CBSAPCI': u'Y',
  u'CENTLAT': u'+37.7600830',
  u'CENTLON': u'-122.6941399',
```

```
  u'FUNCSTAT': u'A',
  u'GEOID': u'0667000',
  u'HU100': 376942,
  u'INTPTLAT': u'+37.7272391',
  u'INTPTLON': u'-123.0322294',
  u'LSADC': u'25',
  u'MTFCC': u'G4110',
  u'NAME': u'San Francisco city',
  u'NECTAPCI': u'N',
  u'OBJECTID': 18435,
  u'OID': 27890355730719,
  u'PLACE': u'67000',
  u'PLACECC': u'C1',
  u'PLACENS': u'02411786',
  u'POP100': 805235,
  u'STATE': u'06',
  u'UR': u''}],
u'Metropolitan Divisions': [{u'AREALAND': 2630357309,
  u'AREAWATER': 2034315130,
  u'BASENAME': u'San Francisco-San Mateo-Redwood City, CA',
  u'CBSA': u'41860',
  u'CENTLAT': u'+37.7625518',
  u'CENTLON': u'-122.5791736',
  u'CSA': u'488',
  u'FUNCSTAT': u'S',
  u'GEOID': u'4186041884',
  u'HU100': 759187,
  u'INTPTLAT': u'+38.0518169',
  u'INTPTLON': u'-122.7459738',
  u'LSADC': u'M3',
  u'METDIV': u'41884',
  u'MTFCC': u'G3120',
  u'NAME': u'San Francisco-San Mateo-Redwood City, CA Metro Division',
  u'OBJECTID': 12,
  u'OID': 26390351387792,
  u'POP100': 1776095}],
u'Metropolitan New England City and Town Areas': [],
u'Metropolitan Statistical Areas': [{u'AREALAND': 6398672255,
  u'AREAWATER': 2474985079,
  u'BASENAME': u'San Francisco-Oakland-Fremont, CA',
  u'CBSA': u'41860',
  u'CENTLAT': u'+37.7739326',
  u'CENTLON': u'-122.2732820',
  u'CSA': u'488',
  u'FUNCSTAT': u'S',
  u'GEOID': u'41860',
```

```
  u'HU100': 1741999,
  u'INTPTLAT': u'+37.7737185',
  u'INTPTLON': u'-122.2744317',
  u'LSADC': u'M1',
  u'MTFCC': u'G3110',
  u'NAME': u'San Francisco-Oakland-Fremont, CA Metro Area',
  u'OBJECTID': 497,
  u'OID': 26290141293319,
  u'POP100': 4335391}],
u'Micropolitan New England City and Town Areas': [],
u'Micropolitan Statistical Areas': [],
u'New England City and Town Area Divisions': [],
u'Off-Reservation Trust Lands': [],
u'Oklahoma Tribal Statistical Areas': [],
u'Public Use Microdata Areas': [{u'AREALAND': 11543902,
  u'AREAWATER': 126041601,
  u'BASENAME': u'San Francisco County (North & East)--North Beach & Chinatown',
  u'CENTLAT': u'+37.8046252',
  u'CENTLON': u'-122.3766929',
  u'FUNCSTAT': u'S',
  u'GEOID': u'0607502',
  u'HU100': 67182,
  u'INTPTLAT': u'+37.8505976',
  u'INTPTLON': u'-122.4013963',
  u'LSADC': u'P5',
  u'MTFCC': u'G6120',
  u'NAME': u'PUMA San Francisco County (North & East)--North Beach & Chinatown',
  u'OBJECTID': 2079,
  u'OID': 217404486282511,
  u'POP100': 107027,
  u'PUMA': u'07502',
  u'STATE': u'06'}],
u'Secondary School Districts': [],
u'State American Indian Reservations': [],
u'State Designated Tribal Statistical Areas': [],
u'State Legislative Districts - Lower': [{u'AREALAND': 62485205,
  u'AREAWATER': 152254425,
  u'BASENAME': u'13',
  u'CENTLAT': u'+37.7933468',
  u'CENTLON': u'-122.3971250',
  u'FUNCSTAT': u'N',
  u'GEOID': u'06013',
  u'HU100': 240272,
  u'INTPTLAT': u'+37.7931941',
  u'INTPTLON': u'-122.3970173',
  u'LSADC': u'L3',
```

```
  u'LSY': u'2010',
  u'MTFCC': u'G5220',
  u'NAME': u'Assembly District 13',
  u'OBJECTID': 566,
  u'OID': 21390355732044,
  u'POP100': 444835,
  u'SLDL': u'013',
  u'STATE': u'06'}],
u'State Legislative Districts - Upper': [{u'AREALAND': 2363247675,
  u'AREAWATER': 993984896,
  u'BASENAME': u'3',
  u'CENTLAT': u'+38.1072847',
  u'CENTLON': u'-122.6933467',
  u'FUNCSTAT': u'N',
  u'GEOID': u'06003',
  u'HU100': 421395,
  u'INTPTLAT': u'+38.1081106',
  u'INTPTLON': u'-122.6947251',
  u'LSADC': u'LU',
  u'LSY': u'2010',
  u'MTFCC': u'G5210',
  u'NAME': u'State Senate District 3',
  u'OBJECTID': 126,
  u'OID': 21290355732359,
  u'POP100': 880421,
  u'SLDU': u'003',
  u'STATE': u'06'}],
u'States': [{u'AREALAND': 403466310059,
  u'AREAWATER': 20501110720,
  u'BASENAME': u'California',
  u'CENTLAT': u'+37.1547249',
  u'CENTLON': u'-119.5277378',
  u'DIVISION': u'9',
  u'FUNCSTAT': u'A',
  u'GEOID': u'06',
  u'HU100': 13680081,
  u'INTPTLAT': u'+37.1485730',
  u'INTPTLON': u'-119.5406515',
  u'LSADC': u'00',
  u'MTFCC': u'G4000',
  u'NAME': u'California',
  u'OBJECTID': 29,
  u'OID': 2749018475066,
  u'POP100': 37253956,
  u'REGION': u'4',
  u'STATE': u'06',
```

```
  u'STATENS': u'01779778',
  u'STUSAB': u'CA',
  u'UR': u''}],
u'Subbarrios': [],
u'Tribal Block Groups': [],
u'Tribal Census Tracts': [],
u'Tribal Designated Statistical Areas': [],
u'Tribal Subdivisions': [],
u'Unified School Districts': [{u'AREALAND': 121399963,
  u'AREAWATER': 479190317,
  u'BASENAME': u'San Francisco Unified School District',
  u'CENTLAT': u'+37.7600830',
  u'CENTLON': u'-122.6941399',
  u'FUNCSTAT': u'E',
  u'GEOID': u'0634410',
  u'HIGRADE': u'12',
  u'HU100': 376942,
  u'INTPTLAT': u'+37.7272391',
  u'INTPTLON': u'-123.0322294',
  u'LOGRADE': u'KG',
  u'LSADC': u'00',
  u'MTFCC': u'G5420',
  u'NAME': u'San Francisco Unified School District',
  u'OBJECTID': 10517,
  u'OID': 28490355767183,
  u'POP100': 805235,
  u'SDTYP': u'',
  u'SDUNI': u'34410',
  u'STATE': u'06',
  u'UR': u''}],
u'Urban Clusters': [],
u'Urban Growth Areas': [],
u'Urbanized Areas': [{u'AREALAND': 1356171095,
  u'AREAWATER': 24162411,
  u'BASENAME': u'San Francisco--Oakland, CA',
  u'CENTLAT': u'+37.7351180',
  u'CENTLON': u'-122.2888818',
  u'FUNCSTAT': u'S',
  u'GEOID': u'78904',
  u'HU100': 1338437,
  u'INTPTLAT': u'+37.6901908',
  u'INTPTLON': u'-122.1285424',
  u'LSADC': u'75',
  u'MTFCC': u'G3500',
  u'NAME': u'San Francisco--Oakland, CA Urbanized Area',
  u'OBJECTID': 1174,
```

```
      u'OID': 27040141325051,
      u'POP100': 3281212,
      u'UA': u'78904'}],
    u'Voting Districts': [{u'AREALAND': 887924,
      u'AREAWATER': 0,
      u'BASENAME': u'03110',
      u'CENTLAT': u'+37.7930624',
      u'CENTLON': u'-122.4029980',
      u'COUNTY': u'075',
      u'FUNCSTAT': u'S',
      u'GEOID': u'0607503110',
      u'HU100': 5307,
      u'INTPTLAT': u'+37.7930624',
      u'INTPTLON': u'-122.4029980',
      u'LSADC': u'V1',
      u'MTFCC': u'G5240',
      u'NAME': u'Voting District 03110',
      u'OBJECTID': 24076,
      u'OID': 215903699030205,
      u'POP100': 7794,
      u'STATE': u'06',
      u'VTD': u'03110',
      u'VTDI': u'P'}],
    u'Zip Code Tabulation Areas': [{u'AREALAND': 891185,
      u'AREAWATER': 494331,
      u'BASENAME': u'94111',
      u'CENTLAT': u'+37.7991868',
      u'CENTLON': u'-122.3979027',
      u'FUNCSTAT': u'S',
      u'GEOID': u'94111',
      u'HU100': 2811,
      u'INTPTLAT': u'+37.7993672',
      u'INTPTLON': u'-122.3984074',
      u'LSADC': u'Z5',
      u'MTFCC': u'G6350',
      u'NAME': u'ZCTA5 94111',
      u'OBJECTID': 30414,
      u'OID': 221404258476618,
      u'POP100': 3713,
      u'ZCTA5': u'94111',
      u'ZCTA5CC': u'B5'}]},
   u'matchedAddress': u'600 Montgomery St, SAN FRANCISCO, CA, 94111',
   u'tigerLine': {u'side': u'R', u'tigerLineId': u'192281262'}}],
 u'input': {u'address': {u'city': u'San Francisco',
   u'state': u'CA',
   u'street': u'600 Montgomery St',
```

```
    u'zip': u'94111'},
   u'benchmark': {u'benchmarkDescription': u'Public Address Ranges - Census 2010 Benchmark',
    u'benchmarkName': u'Public_AR_Census2010',
    u'id': u'9',
    u'isDefault': False},
   u'vintage': {u'id': u'910',
    u'isDefault': True,
    u'vintageDescription': u'Census2010 Vintage - Census2010 Benchmark',
    u'vintageName': u'Census2010_Census2010'}}}}
```

```
myFIPS = []

for item in response.json()['result']['addressMatches']:
    for geography in item['geographies']['Census Blocks']:
        block = geography['BLOCK']
        blockGroup = geography['BLKGRP']
        tract = geography['TRACT']
        county = geography['COUNTY']
        state = geography['STATE']
        FIPScode = state+county+tract+blockGroup
        print 'Block: '+block
        print 'Block Group: '+blockGroup
        print 'Tract: '+tract
        print 'County: '+county
        print 'State: '+state
        print 'FIPS Code: '+FIPScode
        myFIPS.append({'FIPScode':FIPScode,'state':state,'county':county,'tract':tract,'bloc
        print myFIPS
```

```
Block: 1013
Block Group: 1
Tract: 061100
County: 075
State: 06
FIPS Code: 060750611001
[{'blockGroup': u'1', 'county': u'075', 'state': u'06', 'tract': u'061100', 'block': u'1013'
```

Decennial Census (2010, 2000, 1990)

Now let's get some demographic data for the identified geographies

```
#2010 Decennial Census
```

```
requestBase = "http://api.census.gov/data/2010/sf1"
params = {
    'key':myKey,
    'get':'H0030002,H0030003,H0040002,H0040003,H0040004', # Occupied,Vacant,Owned-with mortg
    'for':'county:'+county,
    'in':'state:'+state
    }

# Submit the request to the US Census service
response = requests.get(requestBase,params)
print(response.url)

print
print 'The variable names are defined here - http://api.census.gov/data/2010/sf1/variables.h
print
print(response.json())
print

d2010occupied = float(response.json()[1][0])
d2010vacant = float(response.json()[1][1])
d2010ownerOccupied = float(response.json()[1][2])+float(response.json()[1][3])
d2010renterOccupied = float(response.json()[1][4])

print "2010 Occupied: "+str(d2010occupied)
print "2010 Vacant: "+str(d2010vacant)
d2010occupancyRatio = d2010occupied/(d2010vacant+d2010occupied)
d2010ownerOccupancyRatio = d2010ownerOccupied/(d2010renterOccupied+d2010ownerOccupied)
print "2010 Occupancy ratio: "+str(d2010occupancyRatio)
print "2010 Owner Occupancy ratio: "+str(d2010ownerOccupancyRatio)

dResults = np.array([int('2010'),d2010occupied,d2010vacant,d2010occupancyRatio,d2010ownerOcc
print
print dResults

http://api.census.gov/data/2010/sf1?get=H0030002%2CH0030003%2CH0040002%2CH0040003%2CH004000

The variable names are defined here - http://api.census.gov/data/2010/sf1/variables.html

[[u'H0030002', u'H0030003', u'H0040002', u'H0040003', u'H0040004', u'state', u'county'], [u'

2010 Occupied: 345811.0
2010 Vacant: 31131.0
2010 Occupancy ratio: 0.91741169729
2010 Owner Occupancy ratio: 0.357553692624

[  2.01000000e+03   3.45811000e+05   3.11310000e+04   9.17411697e-01
```

```
   3.57553693e-01]



#2000 Decennial Census
requestBase = "http://api.census.gov/data/2000/sf1"
params = {
    'key':myKey,
    'get':'H003002,H003003,H004002,H004003', # Occupied,Vacant,Owner occupied,Renter Occupie
    'for':'county:'+county,
    'in':'state:'+state
    }

# Submit the request to the US Census service
response = requests.get(requestBase,params)
print(response.url)
print
print "The variable names are defined here - http://api.census.gov/data/2000/sf1/variables.h
print('')
print(response.json())
print

d2000occupied = float(response.json()[1][0])
d2000vacant = float(response.json()[1][1])
d2000ownerOccupied = float(response.json()[1][2])
d2000renterOccupied = float(response.json()[1][3])

print "2000 Occupied: "+str(d2000occupied)
print "2000 Vacant: "+str(d2000vacant)
d2000occupancyRatio = d2000occupied/(d2000vacant+d2000occupied)
d2000ownerOccupancyRatio = d2000ownerOccupied/(d2000renterOccupied+d2000ownerOccupied)
print "2000 Occupancy ratio: "+str(d2000occupancyRatio)
print "2000 Owner Occupancy ratio: "+str(d2000ownerOccupancyRatio)

dResults = np.vstack([dResults,[int('2000'),d2000occupied,d2000vacant,d2000occupancyRatio,d2
print
print dResults

http://api.census.gov/data/2000/sf1?get=H003002%2CH003003%2CH004002%2CH004003&key=001204b740

The variable names are defined here - http://api.census.gov/data/2000/sf1/variables.html

[[u'H003002', u'H003003', u'H004002', u'H004003', u'state', u'county'], [u'329700', u'16827'

2000 Occupied: 329700.0
2000 Vacant: 16827.0
```

```
2000 Occupancy ratio: 0.951441013254
2000 Owner Occupancy ratio: 0.349987867759

[[  2.01000000e+03   3.45811000e+05   3.11310000e+04   9.17411697e-01
    3.57553693e-01]
 [  2.00000000e+03   3.29700000e+05   1.68270000e+04   9.51441013e-01
    3.49987868e-01]]



#1990 Decennial Census - this isn't working ...
requestBase = "http://api.census.gov/data/1990/sf1"
params = {
    'key':myKey,
    'get':'H0020001,H0020002,H0030001,H0030002', # Occupied,Vacant,Owner occupied,Renter Occ
    'for':'county:'+county,
    'in':'state:'+state
    }

# Submit the request to the US Census service
response = requests.get(requestBase,params)
print(response.url)
print
print 'The variable names are defined here - http://api.census.gov/data/1990/sf1/variables.h
print
print(response.json())

d1990occupied = float(response.json()[1][0])
d1990vacant = float(response.json()[1][1])
d1990ownerOccupied = float(response.json()[1][2])
d1990renterOccupied = float(response.json()[1][3])

print "1990 Occupied: "+str(d1990occupied)
print "1990 Vacant: "+str(d1990vacant)
d1990occupancyRatio = d1990occupied/(d1990vacant+d1990occupied)
d1990ownerOccupancyRatio = d1990ownerOccupied/(d1990renterOccupied+d1990ownerOccupied)
print "1990 Occupancy ratio: "+str(d1990occupancyRatio)
print "1990 Owner Occupancy ratio: "+str(d1990ownerOccupancyRatio)

dResults = np.vstack([dResults,[int('1990'),d1990occupied,d1990vacant,d1990occupancyRatio,d1
print
print dResults


http://api.census.gov/data/1990/sf1?get=H0020001%2CH0020002%2CH0030001%2CH0030002&key=001204
```

```
The variable names are defined here - http://api.census.gov/data/1990/sf1/variables.html

[[u'H0020001', u'H0020002', u'H0030001', u'H0030002', u'state', u'county'], [u'305584', u'22
1990 Occupied: 305584.0
1990 Vacant: 22887.0
1990 Occupancy ratio: 0.930322616
1990 Owner Occupancy ratio: 0.345230771245

[[  2.01000000e+03   3.45811000e+05   3.11310000e+04   9.17411697e-01
    3.57553693e-01]
 [  2.00000000e+03   3.29700000e+05   1.68270000e+04   9.51441013e-01
    3.49987868e-01]
 [  1.99000000e+03   3.05584000e+05   2.28870000e+04   9.30322616e-01
    3.45230771e-01]]
```

```
dfDecennial = pd.DataFrame(dResults, index=['2010','2000','1990'], columns=['Year','Occupied
print dfDecennial
dfDecennial.plot(x='Year',y=['Occupied','Vacant'])
dfDecennial.plot(x='Year',y=['OccupancyRatio','OwnerOccupancyRatio'])
```

	Year	Occupied	Vacant	OccupancyRatio	OwnerOccupancyRatio
1990	1990	305584	22887	0.930323	0.345231
2000	2000	329700	16827	0.951441	0.349988
2010	2010	345811	31131	0.917412	0.357554

```
<matplotlib.axes._subplots.AxesSubplot at 0x10dad27d0>
```

Business Dynamics Statistics (1976-2013)

Now, let's get some business data at the state level for the selected geography

```
import pandas as pd
import numpy as np
years = [1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,200
requestBase = "http://api.census.gov/data/bds/firms"
results = []

print 'The variable names are defined here - http://api.census.gov/data/bds/firms/variables.
```

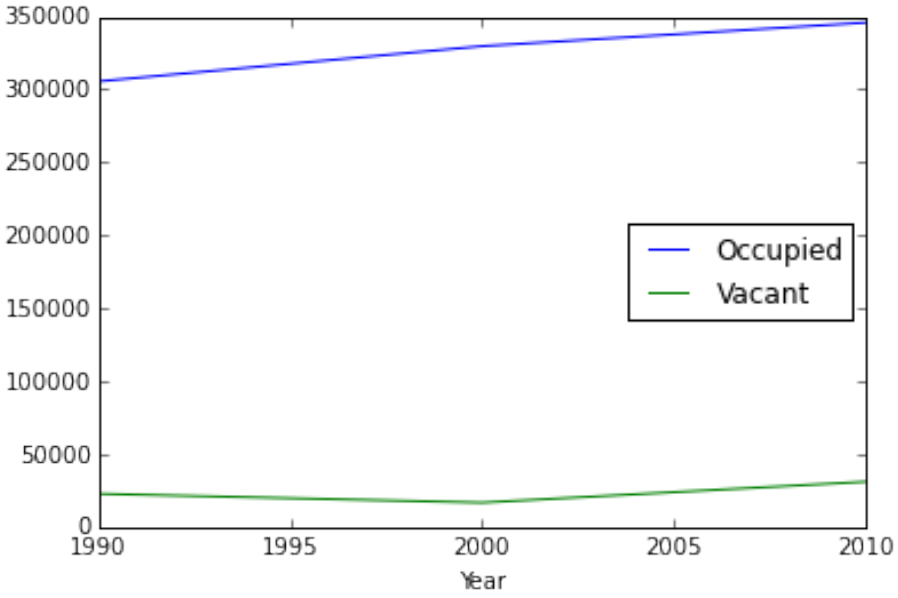

Figure 1: png

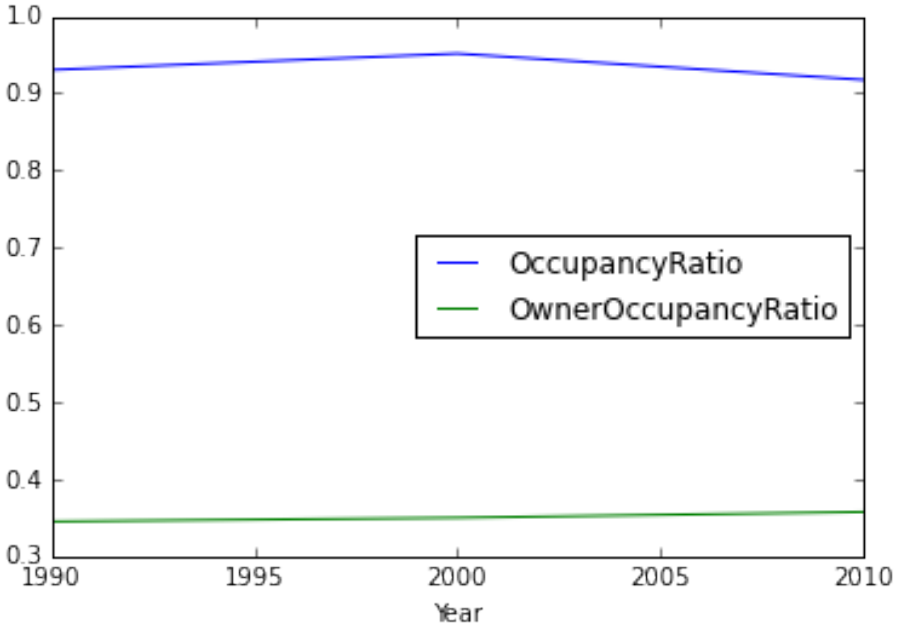

Figure 2: png

```
print

i=0
for year in years:
    params = {
        'key':myKey,
        'get':'emp,net_job_creation', # total employment, net job creation
        'for':'state:'+state,
        'time':year,
        'sic1':'0' # Economy wide
    }

    # Submit the request to the US Census service
    response = requests.get(requestBase,params)
    if i==0:
        print(response.url)
        print
        print "Sample output:"
        print(response.json())
        print
        npArray = np.array(response.json()[0])
    npArray = np.vstack([npArray,response.json()[1]])
    i = i+1

#print results
#print npArray[0]
print "Retrieved Data:"
print npArray[1:]

df = pd.DataFrame(npArray[1:], index=years, columns=npArray[0],dtype=int)

The variable names are defined here - http://api.census.gov/data/bds/firms/variables.html

http://api.census.gov/data/bds/firms?get=emp%2Cnet_job_creation&sic1=0&key=001204b7403884e24

Sample output:
[[u'emp', u'net_job_creation', u'sic1', u'time', u'state'], [u'11171471', u'520743', u'0', u

Retrieved Data:
[[u'11171471' u'520743' u'0' u'1990' u'06']
 [u'10922525' u'-246847' u'0' u'1991' u'06']
 [u'10598897' u'-274006' u'0' u'1992' u'06']
 [u'10523744' u'-41151' u'0' u'1993' u'06']
 [u'10460782' u'-43508' u'0' u'1994' u'06']
 [u'10772608' u'319825' u'0' u'1995' u'06']
 [u'10961827' u'178164' u'0' u'1996' u'06']
```

```
 [u'11338461' u'364191' u'0' u'1997' u'06']
 [u'11814940' u'524136' u'0' u'1998' u'06']
 [u'12129887' u'285142' u'0' u'1999' u'06']
 [u'12629377' u'529144' u'0' u'2000' u'06']
 [u'13142215' u'343792' u'0' u'2001' u'06']
 [u'12803211' u'-284934' u'0' u'2002' u'06']
 [u'12942795' u'331090' u'0' u'2003' u'06']
 [u'13121706' u'310852' u'0' u'2004' u'06']
 [u'13345295' u'311010' u'0' u'2005' u'06']
 [u'13776263' u'508504' u'0' u'2006' u'06']
 [u'13825745' u'107927' u'0' u'2007' u'06']
 [u'13786342' u'53886' u'0' u'2008' u'06']
 [u'12912329' u'-770861' u'0' u'2009' u'06']
 [u'12403206' u'-389443' u'0' u'2010' u'06']
 [u'12557394' u'213364' u'0' u'2011' u'06']
 [u'12852113' u'310980' u'0' u'2012' u'06']
 [u'13311466' u'527789' u'0' u'2013' u'06']]
```

```
# print df
df = df.astype(int)
df.plot(x='time',y=['emp','net_job_creation'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x10d220cd0>
```

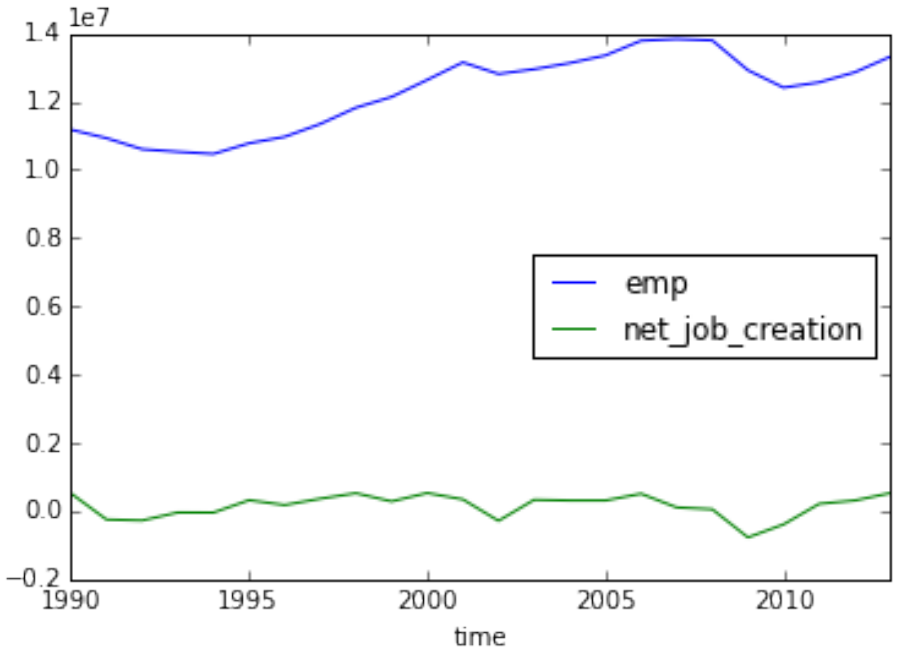
1e7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
1990
1995
2000
2005
2010
time
emp
net_job_creation

Figure 3: png