# Naming the data points for easier reading within the loop/API request

median\_income = 'B19013\_001E'

employed = 'B23025\_004E'

unemployed = 'B23025\_005E'

poverty = 'B17001\_002E'

poverty\_family = 'B17012\_002E'

age = 'B01002\_001E'

population = 'B01003\_001E'

pop\_white\_alone = 'B02001\_002E'

pop\_black\_alone = 'B02001\_003E'

pop\_american\_indian\_alone = 'B02001\_005E'

pop\_native\_hawaiian\_alone = 'B02001\_006E'

pop\_two\_or\_more\_races = 'B02001\_007E'

pop\_hispanic\_origin = 'B02001\_008E'

median\_home\_value = 'B25077\_001E'

median\_gross\_rent = 'B25064\_001E'

commute\_time\_pub\_transit = 'B08136\_007E'

commute\_time\_solo\_auto = 'B08136\_003E'

commute\_time\_walked = 'B08136\_011E'

transit\_solo\_auto = 'B08301\_003E'

transit\_pub\_transit = 'B08301\_010E'

transit\_walked = 'B08301\_019E'

transit\_other = 'B08101\_041E'

ed\_none = 'B15003\_002E'

# Starting the for loop to pull in the data fields. Having some trouble naming the four dataframes using the 'year' iterable...

# Any thoughts on how to do that?

# Run the for loop and it will give you the last dataframe in the series which is '2018', currently. I've formatted the df.

# This is almost certainly more information than we need, but I wanted to include data fields that might be relevent.

# We can always shrink down the dataframes to the fields we want to focus on.

years = [2012,2014,2016]

joint\_df = pd.DataFrame()

for year in years:

med\_income = c.acs1.state(('NAME',

median\_income,

employed,

unemployed,

poverty,

poverty\_family,

age,

population,

pop\_white\_alone,

pop\_black\_alone,

pop\_american\_indian\_alone,

pop\_native\_hawaiian\_alone,

pop\_two\_or\_more\_races,

pop\_hispanic\_origin,

median\_home\_value,

median\_gross\_rent,

commute\_time\_solo\_auto,

commute\_time\_pub\_transit,

commute\_time\_walked,

transit\_solo\_auto,

transit\_pub\_transit,

# 'year=year' is where the 'year in years' is iterated. You can replace this with whatever

# year you want, take it out of for loop and run.

transit\_walked),Census.ALL,year=year)

census\_df = pd.DataFrame(med\_income)

census\_df = census\_df.rename(columns={

'NAME':'State',

'B19013\_001E':'Median Income',

'B23025\_004E':'Number Employed',

'B23025\_005E':'Number Unemployed',

'B17001\_002E':'Number Poverty',

'B17012\_002E':'Number Families in Poverty',

'B01002\_001E':'Median Age',

'B01003\_001E':'Population',

'B02001\_002E':'Pop: White Only',

'B02001\_003E':'Pop: Black Only',

'B02001\_005E':'Pop: American Indian Only',

'B02001\_006E':'Pop: Native Hawaiian Only',

'B02001\_007E':'Pop: Two or More Races',

'B02001\_008E':'Pop: Hispanic Origin',

'B25077\_001E':'Median Home Value',

'B25064\_001E':'Median Gross Rent',

'B08136\_007E':'Commute Time: Solo Auto',

'B08136\_003E':'Commute Time: Public Transit',

'B08136\_011E':'Commute Time: Walking',

'B08301\_003E':'Transit: Solo Auto',

'B08301\_010E':'Transit: Public Transit',

'B08301\_019E':'Transit: Walking'

})

census\_df['Median Income'] = census\_df['Median Income'].astype(float).map("${:,.2f}".format)

census\_df['Median Home Value'] = census\_df['Median Home Value'].astype(float).map("${:,.2f}".format)

census\_df['Median Gross Rent'] = census\_df['Median Gross Rent'].astype(float).map("${:,.2f}".format)

census\_df['Umemployment Rate'] = (census\_df['Number Unemployed']/(census\_df['Number Employed']+census\_df['Number Unemployed'])).astype(float).map("{:.2%}".format)

census\_df['Poverty Rate'] = (census\_df['Number Poverty']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Commute Time: Public Transit'] = (census\_df['Commute Time: Public Transit']/census\_df['Population'])

census\_df['Commute Time: Solo Auto'] = (census\_df['Commute Time: Solo Auto']/census\_df['Population'])

census\_df['Commute Time: Walking'] = (census\_df['Commute Time: Walking']/census\_df['Population'])

census\_df['Transit: Solo Auto'] = (census\_df['Transit: Solo Auto']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Transit: Public Transit'] = (census\_df['Transit: Public Transit']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Transit: Walking'] = (census\_df['Transit: Walking']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Number Employed'] = census\_df['Number Employed'].astype(int)

census\_df['Number Unemployed'] = census\_df['Number Unemployed'].astype(int)

census\_df['Number Poverty'] = census\_df['Number Poverty'].astype(int)

census\_df['Number Families in Poverty'] = census\_df['Number Families in Poverty'].astype(int)

census\_df['Population'] = census\_df['Population'].astype(int)

census\_df['Pop: White Only'] = census\_df['Pop: White Only'].astype(int)

census\_df['Pop: Black Only'] = census\_df['Pop: Black Only'].astype(int)

census\_df['Pop: American Indian Only'] = census\_df['Pop: American Indian Only'].astype(int)

census\_df['Pop: Native Hawaiian Only'] = census\_df['Pop: Native Hawaiian Only'].astype(int)

census\_df['Pop: Two or More Races'] = census\_df['Pop: Two or More Races'].astype(int)

census\_df['Pop: Hispanic Origin'] = census\_df['Pop: Hispanic Origin'].astype(int)

census\_df['Pop Rate: White Only'] = (census\_df['Pop: White Only']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Pop Rate: Black Only'] = (census\_df['Pop: Black Only']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Pop Rate: American Indian Only'] = (census\_df['Pop: American Indian Only']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Pop Rate: Native Hawaiian Only'] = (census\_df['Pop: Native Hawaiian Only']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Pop Rate: Two or More Races'] = (census\_df['Pop: Two or More Races']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df['Pop Rate: Hispanic Origin'] = (census\_df['Pop: Hispanic Origin']/census\_df['Population']).astype(float).map("{:.2%}".format)

census\_df.insert(0, 'Year', year)

joint\_df = pd.concat([census\_df, joint\_df])

joint\_df.to\_csv("census\_data\_12\_to\_16.csv",encoding="utf-8",index=False)

joint\_df