# US Housing Market Analysis

**By: Rena Han & Cole Yang**

**Overview**

For our Data Bootcamp project, we decided to study how the domestic housing market has evolved over the course of the last 19 years. Specifically, we wanted to take a look at changes in the demographic and economic factors related to housing price, such as the change in population and income per capita. To best analyze these relationships, we decided to consolidate the data on a county-specific level, which we believe will offer sufficient detail.

Because this analysis was only conducted on the US housing market, its findings are not applicable to the housing markets in other international countries. Furthermore, because there are many other factors that influence housing price beyond just population and income per capita (quality of education, weather, etc.), this analysis is far from conclusive. Nonetheless, we hope that our project can bring to life some interesting insights.

**Data**

The key elements of this project are the use of the Bureau of Economic Analysis' API and Zillow's housing data (coming from Zillow Research). These two sources will provide access to the above mentioned demographic and economic factors at the correct geographic level between the years of 1998 and 2017.

From BEA

- regional accounts data on a county level (population and income per capita)

From Zillow

- housing price data for single-family homes (in excel format)

## Setup

In the following steps, we import the necessary modules

In [1114]:
```
import pandas as pd
import requests
import numpy as np
import matplotlib.pyplot as plt
import weightedcalcs as wc
import datetime as dt
import seaborn as sns

%matplotlib inline
```

## Grabbing the data

Here we use a BEA API key (from class) to grab income per capita and population for all years between 1998 and 2017

In [1115]:
```
BEA_ID = "6BF79D8C-8042-4196-88DC-0E0C55B0C3B6"

my_key = "https://APPS.BEA.gov/api/data?&UserID=" + BEA_ID

data_set =  "&method=GetData&" + "datasetname=RegionalIncome&"

table_and_line_income = "TableName=CA1&LineCode=3&"

table_and_line_population = "TableName=CA1&LineCode=2&"

year = "Year=" + "1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998,
 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
2011, 2012, 2013, 2014, 2015, 2016, 2017" + "&"

location = "GeoFips=COUNTY&"

form = "ResultFormat=JSON"
```

## Creating the income dataframe

In [1116]:
```
API_URL = my_key + data_set + table_and_line_income + year + location +
form

r = requests.get(API_URL)

df_income = pd.DataFrame(r.json()["BEAAPI"]["Results"]["Data"])
```

In [1117]:
```
df_income.drop(['CL_UNIT', 'Code',"NoteRef", "UNIT_MULT"], axis=1, inpla
ce = True)

df_income.rename(columns={"DataValue":"IncomePerCapita"}, inplace=True)
```

In [1118]:
```
df_income.head(10)
```

Out[1118]:

| | IncomePerCapita | GeoFips | GeoName | TimePeriod |
|---|---|---|---|---|
| **0** | 19,621 | 00000 | United States | 1990 |
| **1** | 20,030 | 00000 | United States | 1991 |
| **2** | 21,090 | 00000 | United States | 1992 |
| **3** | 21,733 | 00000 | United States | 1993 |
| **4** | 22,575 | 00000 | United States | 1994 |
| **5** | 23,607 | 00000 | United States | 1995 |
| **6** | 24,771 | 00000 | United States | 1996 |
| **7** | 25,993 | 00000 | United States | 1997 |
| **8** | 27,557 | 00000 | United States | 1998 |
| **9** | 28,675 | 00000 | United States | 1999 |

## Creating the population dataframe

In [1119]:
```
API_URL = my_key + data_set + table_and_line_population + year + locatio
n + form

r = requests.get(API_URL)

population = pd.DataFrame(r.json()["BEAAPI"]["Results"]["Data"])

population.drop(['CL_UNIT', 'Code',"NoteRef", "UNIT_MULT", "GeoName"], a
xis=1, inplace = True)

population.rename(columns={"DataValue":"Pop"}, inplace=True)
```

In [1120]:
```
population.head(10)
```

Out[1120]:

| | Pop | GeoFips | TimePeriod |
|---|---|---|---|
| 0 | 249,622,814 | 00000 | 1990 |
| 1 | 252,980,941 | 00000 | 1991 |
| 2 | 256,514,224 | 00000 | 1992 |
| 3 | 259,918,588 | 00000 | 1993 |
| 4 | 263,125,821 | 00000 | 1994 |
| 5 | 266,278,393 | 00000 | 1995 |
| 6 | 269,394,284 | 00000 | 1996 |
| 7 | 272,646,925 | 00000 | 1997 |
| 8 | 275,854,104 | 00000 | 1998 |
| 9 | 279,040,168 | 00000 | 1999 |

## Merging the two dataframes

In [1121]:
```
pop_income = pd.merge(population, df_income,
                      how='left',
                      on=['GeoFips',"TimePeriod"],
                      indicator=True)
```

We set the index to GeoFips so that the data will be easier to work with later on

In [1122]:
```
pop_income.set_index("GeoFips", inplace = True)
```

Some basic information about the merged pop_income df

In [1123]:
```
pop_income.shape
```

Out[1123]:
```
(89544, 5)
```

In [1124]:
```
pop_income.dtypes
```

Out[1124]:
```
Pop                 object
TimePeriod          object
IncomePerCapita     object
GeoName             object
_merge            category
dtype: object
```

In [1125]:
```
pop_income.head(10)
```

Out[1125]:

| GeoFips | Pop | TimePeriod | IncomePerCapita | GeoName | _merge |
|---|---|---|---|---|---|
| 00000 | 249,622,814 | 1990 | 19,621 | United States | both |
| 00000 | 252,980,941 | 1991 | 20,030 | United States | both |
| 00000 | 256,514,224 | 1992 | 21,090 | United States | both |
| 00000 | 259,918,588 | 1993 | 21,733 | United States | both |
| 00000 | 263,125,821 | 1994 | 22,575 | United States | both |
| 00000 | 266,278,393 | 1995 | 23,607 | United States | both |
| 00000 | 269,394,284 | 1996 | 24,771 | United States | both |
| 00000 | 272,646,925 | 1997 | 25,993 | United States | both |
| 00000 | 275,854,104 | 1998 | 27,557 | United States | both |
| 00000 | 279,040,168 | 1999 | 28,675 | United States | both |

In [1126]:
```
pop_income.tail(10)
```

Out[1126]:

| GeoFips | Pop | TimePeriod | IncomePerCapita | GeoName | _merge |
|---|---|---|---|---|---|
| 98000 | 51,608,614 | 2008 | 43,214 | Far West | both |
| 98000 | 52,167,532 | 2009 | 41,298 | Far West | both |
| 98000 | 52,686,778 | 2010 | 42,636 | Far West | both |
| 98000 | 53,176,406 | 2011 | 44,966 | Far West | both |
| 98000 | 53,679,832 | 2012 | 47,451 | Far West | both |
| 98000 | 54,161,802 | 2013 | 47,819 | Far West | both |
| 98000 | 54,695,081 | 2014 | 50,787 | Far West | both |
| 98000 | 55,249,155 | 2015 | 53,911 | Far West | both |
| 98000 | 55,772,858 | 2016 | 55,550 | Far West | both |
| 98000 | 56,250,544 | 2017 | 57,748 | Far West | both |

Converting to excel so we can doublecheck our changes

In [1127]:
```
pop_income.to_excel("Population Income.xlsx")
```

## Bringing in the Zillow dataset

In [1128]:
```
file = "https://github.com/zcy204/data_bootcamp_final_project/raw/master/zillowdata.xlsx" # from Zillow Home Value Index
zillow = pd.read_excel(file)
```

In [1129]:
```
zillow.drop(['RegionID',"State", "NEW STATE CODE", "MunicipalCodeFIPS", "NEW FIPS", "SizeRank"], axis=1, inplace = True)
```

In [1130]:
```
zillow.head(10)
```

Out[1130]:

| | RegionName | StateCodeFIPS | GEOFIPS | 1998-01 | 1998-02 | 1998-03 | 1998-04 | 1998-05 | 19 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Autauga County | 1 | 1001 | 109900.0 | 110300.0 | 110600.0 | 110800.0 | 110900.0 | 11 |
| **1** | Baldwin County | 1 | 1003 | 101500.0 | 102400.0 | 102800.0 | 102900.0 | 102600.0 | 10 |
| **2** | Blount County | 1 | 1009 | 79200.0 | 80600.0 | 81600.0 | 82600.0 | 83500.0 | 8 |
| **3** | Cherokee County | 1 | 1019 | 62100.0 | 62900.0 | 63600.0 | 64300.0 | 64800.0 | 6 |
| **4** | Chilton County | 1 | 1021 | 82600.0 | 83000.0 | 83300.0 | 83600.0 | 83900.0 | 8 |
| **5** | Coffee County | 1 | 1031 | 79600.0 | 78900.0 | 78400.0 | 77900.0 | 77700.0 | 7 |
| **6** | Colbert County | 1 | 1033 | 40100.0 | 40000.0 | 40000.0 | 40000.0 | 39800.0 | 3 |
| **7** | De Kalb County | 1 | 1049 | 56600.0 | 56300.0 | 56300.0 | 56700.0 | 57300.0 | 5 |
| **8** | Elmore County | 1 | 1051 | 112100.0 | 112900.0 | 113500.0 | 114100.0 | 114500.0 | 11 |
| **9** | Etowah County | 1 | 1055 | 65200.0 | 65700.0 | 65900.0 | 65800.0 | 65600.0 | 6 |

10 rows × 263 columns

## Cleaning the data

Because values in the GEOFIPS column are stored as integers instead of strings, many of them are missing 0s. For example, Autauga County should have a GEOFIPS of 01001, but because its an integer, the 0 dissapears and the GEOFIPS is incorrectly shown as 1001. Below, we wrote a for loop in order to fix that issue and update the GEOFIPS column with the correct values.

In [1131]:
```
values = zillow['GEOFIPS'].tolist()
```

In [1132]:
```
newval = []
for val in values:
    if len(str(val)) != 5:
        val = '0' + str(val)
        newval.append(val)
    else:
        val = val
        newval.append(val)
```

In [1133]:
```
zillow['GEOFIPS'] = newval
```

In [1134]:
```
zillow['GEOFIPS']
```

Out[1134]:

| | |
|---|---|
| 0 | 01001 |
| 1 | 01003 |
| 2 | 01009 |
| 3 | 01019 |
| 4 | 01021 |
| 5 | 01031 |
| 6 | 01033 |
| 7 | 01049 |
| 8 | 01051 |
| 9 | 01055 |
| 10 | 01063 |
| 11 | 01069 |
| 12 | 01073 |
| 13 | 01077 |
| 14 | 01081 |
| 15 | 01083 |
| 16 | 01089 |
| 17 | 01097 |
| 18 | 01101 |
| 19 | 01109 |
| 20 | 01113 |
| 21 | 01115 |
| 22 | 01117 |
| 23 | 01121 |
| 24 | 01123 |
| 25 | 01125 |
| 26 | 01133 |
| 27 | 02020 |
| 28 | 02090 |
| 29 | 02110 |
| ... | ... |
| 1881 | 55113 |
| 1882 | 55115 |
| 1883 | 55121 |
| 1884 | 55123 |
| 1885 | 55125 |
| 1886 | 55127 |
| 1887 | 55131 |
| 1888 | 55133 |
| 1889 | 55135 |
| 1890 | 55137 |
| 1891 | 55139 |
| 1892 | 55141 |
| 1893 | 56001 |
| 1894 | 56005 |
| 1895 | 56009 |
| 1896 | 56013 |
| 1897 | 56015 |
| 1898 | 56017 |
| 1899 | 56019 |
| 1900 | 56021 |
| 1901 | 56023 |
| 1902 | 56025 |
| 1903 | 56029 |
| 1904 | 56031 |
| 1905 | 56033 |
| 1906 | 56037 |

```
1907    56039
1908    56041
1909    56043
1910    56045
Name: GEOFIPS, Length: 1911, dtype: object
```

Sorting values by StateCodeFIPS and RegionName allows us to first order the data alphabetically by state and then again alphabetically by county name within each state. For example, Alabama has StateCodeFips 1 so after sorting, all of the Alabama counties show up first, with counties like Autauga, Baldwin, Cherokee, etc. also ranked accordingly.

In [1135]:
```
zillow.sort_values(by=['StateCodeFIPS', 'RegionName'], inplace = True) # trial and error
```

In [1136]:
```
zillow.drop(['StateCodeFIPS'], axis = 1, inplace = True )
```

In [1137]:
```
zillow.columns
```

Out[1137]:
```
Index(['RegionName',    'GEOFIPS',    '1998-01',    '1998-02',    '1998
-03',
          '1998-04',    '1998-05',    '1998-06',    '1998-07',    '1998
-08',
       ...
          '2017-04',    '2017-05',    '2017-06',    '2017-07',    '2017
-08',
          '2017-09',    '2017-10',    '2017-11',    '2017-12',
2017],
      dtype='object', length=262)
```

In [1138]:
```
newcol = ['RegionName','GEOFIPS',1998, 1999, 2000, 2001, 2002, 2003, 200
4, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 201
6, 2017]
```

In [1139]:
```
zillow = zillow[newcol]
```

In [1140]:
```
zillow.head(10)
```

Out[1140]:

| | RegionName | GEOFIPS | 1998 | 1999 | 2000 | 2001 | |
|---|---|---|---|---|---|---|---|
| **0** | Autauga County | 01001 | 110558.333333 | 111250.000000 | 115125.000000 | 109325.000000 | 118983.3 |
| **1** | Baldwin County | 01003 | 102433.333333 | 112983.333333 | 128941.666667 | 116558.333333 | 141233.3 |
| **2** | Blount County | 01009 | 84300.000000 | 92058.333333 | 94000.000000 | 91233.333333 | 102700.0 |
| **3** | Cherokee County | 01019 | 65075.000000 | 67283.333333 | 71141.666667 | 69208.333333 | 77825.0 |
| **4** | Chilton County | 01021 | 84183.333333 | 87950.000000 | 90741.666667 | 84783.333333 | 93691.6 |
| **5** | Coffee County | 01031 | 78391.666667 | 79166.666667 | 80216.666667 | 76516.666667 | 85458.3 |
| **6** | Colbert County | 01033 | 38958.333333 | 39550.000000 | 45375.000000 | 48808.333333 | 60750.0 |
| **7** | De Kalb County | 01049 | 59508.333333 | 64316.666667 | 67250.000000 | 64291.666667 | 71625.0 |
| **8** | Elmore County | 01051 | 114058.333333 | 116125.000000 | 121291.666667 | 116050.000000 | 148950.0 |
| **9** | Etowah County | 01055 | 64750.000000 | 61183.333333 | 65616.666667 | 67508.333333 | 75483.3 |

10 rows × 22 columns

## Preparing for the pop_income and Zillow merge

The Zillow data came in a format that was very wide, which was going to be an issue if we needed to merge it with the pop_income dataframe, which was long. Transposing didn't work, so instead we used the melt function. Our two identifier variables were Regionname and GEOFIPS, which we kept set (id_vars).

In [1141]:
```
zillowdf = zillow.melt(id_vars = ['RegionName','GEOFIPS'])
```

In [1142]:
```
zillowdf.columns = ['RegionName', 'GeoFips', 'TimePeriod','Median_Housing_Price']
```

In [1143]:
```
zillowdf.set_index('GeoFips', inplace = True)
```

In [1144]:
```
zillowdf = zillowdf.reset_index()
```

In [1145]:
```
zillowdf['Median_Housing_Price'] = np.round(zillowdf['Median_Housing_Price'], decimals=2)
```

In [1146]:
```
zillowdf.head(10)
```

Out[1146]:

| | GeoFips | RegionName | TimePeriod | Median_Housing_Price |
|---|---|---|---|---|
| **0** | 01001 | Autauga County | 1998 | 110558.33 |
| **1** | 01003 | Baldwin County | 1998 | 102433.33 |
| **2** | 01009 | Blount County | 1998 | 84300.00 |
| **3** | 01019 | Cherokee County | 1998 | 65075.00 |
| **4** | 01021 | Chilton County | 1998 | 84183.33 |
| **5** | 01031 | Coffee County | 1998 | 78391.67 |
| **6** | 01033 | Colbert County | 1998 | 38958.33 |
| **7** | 01049 | De Kalb County | 1998 | 59508.33 |
| **8** | 01051 | Elmore County | 1998 | 114058.33 |
| **9** | 01055 | Etowah County | 1998 | 64750.00 |

In [1147]:
```
zillowdf.tail(10)
```

Out[1147]:

| | GeoFips | RegionName | TimePeriod | Median_Housing_Price |
|---|---|---|---|---|
| **38210** | 56023 | Lincoln County | 2017 | 210408.33 |
| **38211** | 56025 | Natrona County | 2017 | 199000.00 |
| **38212** | 56029 | Park County | 2017 | 233491.67 |
| **38213** | 56031 | Platte County | 2017 | 178666.67 |
| **38214** | 56033 | Sheridan County | 2017 | 227908.33 |
| **38215** | 56037 | Sweetwater County | 2017 | 221233.33 |
| **38216** | 56039 | Teton County | 2017 | 762708.33 |
| **38217** | 56041 | Uinta County | 2017 | 203333.33 |
| **38218** | 56043 | Washakie County | 2017 | 169325.00 |
| **38219** | 56045 | Weston County | 2017 | 138875.00 |

## Preparing the pop_income dataframe for merging

In [1148]:
```
pop_income = pop_income.reset_index()
```

In [1149]:
```
pop_income = pop_income.drop(['_merge'], axis = 1)
```

In [1150]:
```
pop_income
```

Out[1150]:

| | GeoFips | Pop | TimePeriod | IncomePerCapita | GeoName |
|---|---|---|---|---|---|
| **0** | 00000 | 249,622,814 | 1990 | 19,621 | United States |
| **1** | 00000 | 252,980,941 | 1991 | 20,030 | United States |
| **2** | 00000 | 256,514,224 | 1992 | 21,090 | United States |
| **3** | 00000 | 259,918,588 | 1993 | 21,733 | United States |
| **4** | 00000 | 263,125,821 | 1994 | 22,575 | United States |
| **5** | 00000 | 266,278,393 | 1995 | 23,607 | United States |
| **6** | 00000 | 269,394,284 | 1996 | 24,771 | United States |
| **7** | 00000 | 272,646,925 | 1997 | 25,993 | United States |
| **8** | 00000 | 275,854,104 | 1998 | 27,557 | United States |
| **9** | 00000 | 279,040,168 | 1999 | 28,675 | United States |
| **10** | 00000 | 282,162,411 | 2000 | 30,657 | United States |
| **11** | 00000 | 284,968,955 | 2001 | 31,589 | United States |
| **12** | 00000 | 287,625,193 | 2002 | 31,832 | United States |
| **13** | 00000 | 290,107,933 | 2003 | 32,681 | United States |
| **14** | 00000 | 292,805,298 | 2004 | 34,251 | United States |
| **15** | 00000 | 295,516,599 | 2005 | 35,849 | United States |
| **16** | 00000 | 298,379,912 | 2006 | 38,114 | United States |
| **17** | 00000 | 301,231,207 | 2007 | 39,844 | United States |
| **18** | 00000 | 304,093,966 | 2008 | 40,904 | United States |
| **19** | 00000 | 306,771,529 | 2009 | 39,284 | United States |
| **20** | 00000 | 309,338,421 | 2010 | 40,545 | United States |
| **21** | 00000 | 311,644,280 | 2011 | 42,727 | United States |
| **22** | 00000 | 313,993,272 | 2012 | 44,582 | United States |
| **23** | 00000 | 316,234,505 | 2013 | 44,826 | United States |
| **24** | 00000 | 318,622,525 | 2014 | 47,025 | United States |
| **25** | 00000 | 321,039,839 | 2015 | 48,940 | United States |
| **26** | 00000 | 323,405,935 | 2016 | 49,831 | United States |
| **27** | 00000 | 325,719,178 | 2017 | 51,640 | United States |
| **28** | 01000 | 4,050,055 | 1990 | 15,861 | Alabama |
| **29** | 01000 | 4,099,156 | 1991 | 16,572 | Alabama |
| **...** | ... | ... | ... | ... | ... |
| **89514** | 97000 | 11,878,018 | 2016 | 47,505 | Rocky Mountain |
| **89515** | 97000 | 12,055,738 | 2017 | 49,265 | Rocky Mountain |
| **89516** | 98000 | 40,610,409 | 1990 | 21,119 | Far West |

| | GeoFips | Pop | TimePeriod | IncomePerCapita | GeoName |
|---|---|---|---|---|---|
| **89517** | 98000 | 41,427,985 | 1991 | 21,585 | Far West |
| **89518** | 98000 | 42,225,887 | 1992 | 22,471 | Far West |
| **89519** | 98000 | 42,797,622 | 1993 | 22,909 | Far West |
| **89520** | 98000 | 43,271,002 | 1994 | 23,537 | Far West |
| **89521** | 98000 | 43,744,822 | 1995 | 24,549 | Far West |
| **89522** | 98000 | 44,314,342 | 1996 | 25,804 | Far West |
| **89523** | 98000 | 45,053,779 | 1997 | 27,048 | Far West |
| **89524** | 98000 | 45,798,042 | 1998 | 28,921 | Far West |
| **89525** | 98000 | 46,505,506 | 1999 | 30,354 | Far West |
| **89526** | 98000 | 47,188,420 | 2000 | 32,799 | Far West |
| **89527** | 98000 | 47,891,178 | 2001 | 33,439 | Far West |
| **89528** | 98000 | 48,493,357 | 2002 | 33,607 | Far West |
| **89529** | 98000 | 49,053,068 | 2003 | 34,755 | Far West |
| **89530** | 98000 | 49,601,761 | 2004 | 36,653 | Far West |
| **89531** | 98000 | 50,090,268 | 2005 | 38,489 | Far West |
| **89532** | 98000 | 50,570,529 | 2006 | 41,153 | Far West |
| **89533** | 98000 | 51,031,362 | 2007 | 42,782 | Far West |
| **89534** | 98000 | 51,608,614 | 2008 | 43,214 | Far West |
| **89535** | 98000 | 52,167,532 | 2009 | 41,298 | Far West |
| **89536** | 98000 | 52,686,778 | 2010 | 42,636 | Far West |
| **89537** | 98000 | 53,176,406 | 2011 | 44,966 | Far West |
| **89538** | 98000 | 53,679,832 | 2012 | 47,451 | Far West |
| **89539** | 98000 | 54,161,802 | 2013 | 47,819 | Far West |
| **89540** | 98000 | 54,695,081 | 2014 | 50,787 | Far West |
| **89541** | 98000 | 55,249,155 | 2015 | 53,911 | Far West |
| **89542** | 98000 | 55,772,858 | 2016 | 55,550 | Far West |
| **89543** | 98000 | 56,250,544 | 2017 | 57,748 | Far West |

89544 rows × 5 columns

In [1151]:
```
US = pop_income.set_index('GeoFips').loc['00000']
```

In [1152]:
```
updateinc = []
for val in US['IncomePerCapita']:
    if ',' in val:
        val = val.replace(',', '')
        val = int(val)
        updateinc.append(val)
    else:
        updateinc.append(0)
```

In [1153]:
```
newpop = []
for val in US['Pop']:
    if ',' in val:
        val = val.replace(',', '')
        val = int(val)
        newpop.append(val)
    else:
        newpop.append(0)
```

In [1154]:
```
US['Pop'] = newpop
US['IncomePerCapita'] = updateinc
```

In [1155]:
```
US['Pop'] = US['Pop'].astype(int) / 1000000
```

In [1156]:
```
US['Pop'] = np.round(US['Pop'], decimals=2)
```

In [1157]:
```
US.set_index('TimePeriod', inplace = True)
```

In [1158]:
```
US # this gives us the country's population and IncomePC over the last 27 years
```

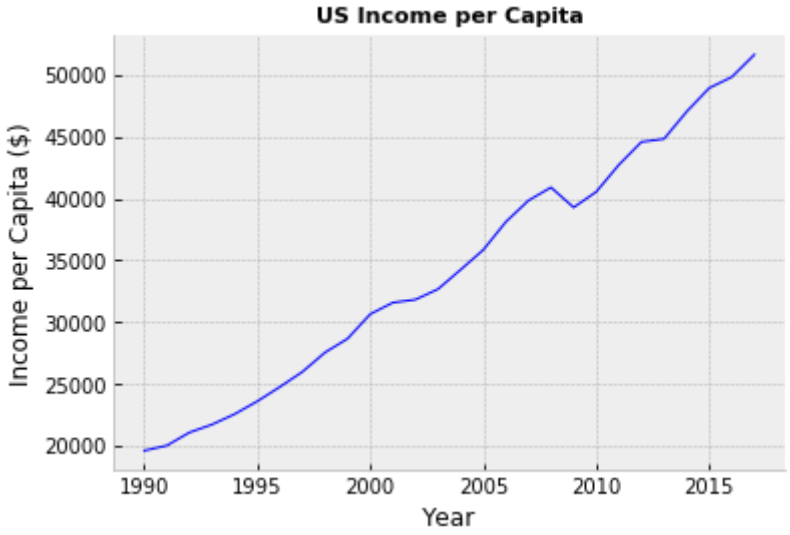
Out[1158]:

| TimePeriod | Pop | IncomePerCapita | GeoName |
|---|---|---|---|
| **1990** | 249.62 | 19621 | United States |
| **1991** | 252.98 | 20030 | United States |
| **1992** | 256.51 | 21090 | United States |
| **1993** | 259.92 | 21733 | United States |
| **1994** | 263.13 | 22575 | United States |
| **1995** | 266.28 | 23607 | United States |
| **1996** | 269.39 | 24771 | United States |
| **1997** | 272.65 | 25993 | United States |
| **1998** | 275.85 | 27557 | United States |
| **1999** | 279.04 | 28675 | United States |
| **2000** | 282.16 | 30657 | United States |
| **2001** | 284.97 | 31589 | United States |
| **2002** | 287.63 | 31832 | United States |
| **2003** | 290.11 | 32681 | United States |
| **2004** | 292.81 | 34251 | United States |
| **2005** | 295.52 | 35849 | United States |
| **2006** | 298.38 | 38114 | United States |
| **2007** | 301.23 | 39844 | United States |
| **2008** | 304.09 | 40904 | United States |
| **2009** | 306.77 | 39284 | United States |
| **2010** | 309.34 | 40545 | United States |
| **2011** | 311.64 | 42727 | United States |
| **2012** | 313.99 | 44582 | United States |
| **2013** | 316.23 | 44826 | United States |
| **2014** | 318.62 | 47025 | United States |
| **2015** | 321.04 | 48940 | United States |
| **2016** | 323.41 | 49831 | United States |
| **2017** | 325.72 | 51640 | United States |

## US Income

The plot below shows the nationwide trend in income per capita growth over the period 1990-2017. Besides a few hiccups, income per capita has been gradually increasing YoY, reaching approximately $55,000 by 2017.

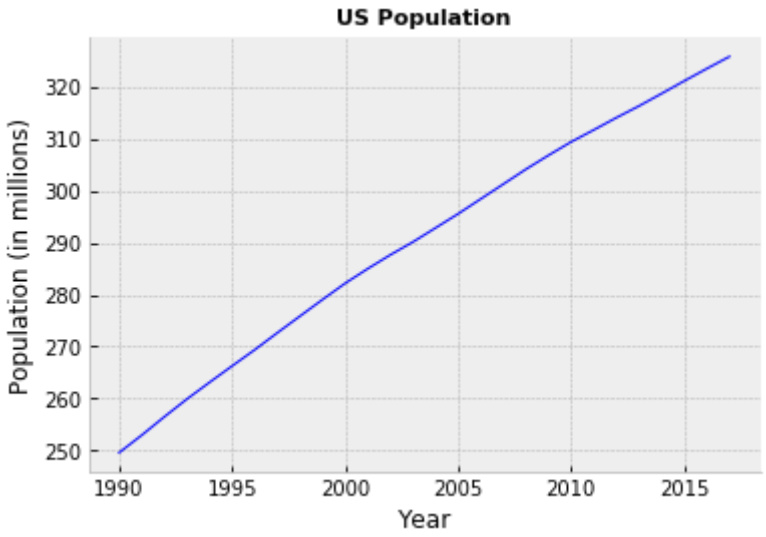
In [1159]:
```
fig, ax = plt.subplots()

plt.style.use('bmh')

ax.plot(US.index.astype(int), US.IncomePerCapita, color = 'b', linewidth
= '1')
ax.set_title("US Income per Capita", fontsize = 11, fontweight = "bold")
ax.set_ylabel("Income per Capita ($)")
ax.set_xlabel("Year")

ax.set_xticks(np.arange(1990,2017,5))

ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

plt.show()
```




## US Population

The time series line plot belows simply shows the United States population and its steady growth over the period 1990-2017. The US Population is now approximately 325 million people as of 2017/18.

In [1160]:
```
fig, ax = plt.subplots()

ax.plot(US.index.astype(int), US.Pop, color = 'b', linewidth = '1')
ax.set_title("US Population", fontsize = 11, fontweight = "bold")
ax.set_ylabel("Population (in millions)")
ax.set_xlabel("Year")

ax.set_xticks(np.arange(1990,2017,5))


ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

plt.show()
```




Below, we drop all the observations not directly related to a US county. This includes rows with data about the US as a whole, or regions like the Far West.

In [1161]:
```
pop_income = pop_income.set_index(['GeoName'])
```

In [1162]:
```
pop_income = pop_income.drop(['United States',
'New England',
'Mideast',
'Great Lakes',
'Plains',
'Southeast',
'Southwest',
'Rocky Mountain',
'Far West',
'Alabama',
'Alaska',
'Arizona',
'Arkansas',
'California',
'Colorado',
'Connecticut',
'Delaware',
'Florida',
'Georgia',
'Hawaii',
'Idaho',
'Illinois',
'Indiana',
'Iowa',
'Kansas',
'Kentucky',
'Louisiana',
'Maine',
'Maryland',
'Massachusetts',
'Michigan',
'Minnesota',
'Mississippi',
'Missouri',
'Montana',
'Nebraska',
'Nevada',
'New Hampshire',
'New Jersey',
'New Mexico',
'New York',
'North Carolina',
'North Dakota',
'Ohio',
'Oklahoma',
'Oregon',
'Pennsylvania',
'Rhode Island',
'South Carolina',
'South Dakota',
'Tennessee',
'Texas',
'Utah',
'Vermont',
'Virginia',
'Washington',
```

```
'West Virginia',
'Wisconsin',
'Wyoming'])
```

In [1163]:
```
pop_income.reset_index()
```

Out[1163]:

| | GeoName | GeoFips | Pop | TimePeriod | IncomePerCapita |
|---|---|---|---|---|---|
| **0** | Autauga, AL | 01001 | 34,353 | 1990 | 15,482 |
| **1** | Autauga, AL | 01001 | 35,010 | 1991 | 16,417 |
| **2** | Autauga, AL | 01001 | 35,985 | 1992 | 17,063 |
| **3** | Autauga, AL | 01001 | 36,953 | 1993 | 17,700 |
| **4** | Autauga, AL | 01001 | 38,186 | 1994 | 18,683 |
| **5** | Autauga, AL | 01001 | 39,112 | 1995 | 19,350 |
| **6** | Autauga, AL | 01001 | 40,207 | 1996 | 20,155 |
| **7** | Autauga, AL | 01001 | 41,238 | 1997 | 21,076 |
| **8** | Autauga, AL | 01001 | 42,106 | 1998 | 22,073 |
| **9** | Autauga, AL | 01001 | 42,963 | 1999 | 22,948 |
| **10** | Autauga, AL | 01001 | 44,021 | 2000 | 23,699 |
| **11** | Autauga, AL | 01001 | 44,889 | 2001 | 24,682 |
| **12** | Autauga, AL | 01001 | 45,909 | 2002 | 24,991 |
| **13** | Autauga, AL | 01001 | 46,800 | 2003 | 26,319 |
| **14** | Autauga, AL | 01001 | 48,366 | 2004 | 27,535 |
| **15** | Autauga, AL | 01001 | 49,676 | 2005 | 28,766 |
| **16** | Autauga, AL | 01001 | 51,328 | 2006 | 29,758 |
| **17** | Autauga, AL | 01001 | 52,405 | 2007 | 31,459 |
| **18** | Autauga, AL | 01001 | 53,277 | 2008 | 32,876 |
| **19** | Autauga, AL | 01001 | 54,135 | 2009 | 32,603 |
| **20** | Autauga, AL | 01001 | 54,750 | 2010 | 33,415 |
| **21** | Autauga, AL | 01001 | 55,199 | 2011 | 34,325 |
| **22** | Autauga, AL | 01001 | 54,927 | 2012 | 35,040 |
| **23** | Autauga, AL | 01001 | 54,695 | 2013 | 35,464 |
| **24** | Autauga, AL | 01001 | 54,864 | 2014 | 36,677 |
| **25** | Autauga, AL | 01001 | 54,838 | 2015 | 38,591 |
| **26** | Autauga, AL | 01001 | 55,278 | 2016 | 39,509 |
| **27** | Autauga, AL | 01001 | 55,504 | 2017 | 40,484 |
| **28** | Baldwin, AL | 01003 | 98,955 | 1990 | 15,831 |
| **29** | Baldwin, AL | 01003 | 102,420 | 1991 | 16,954 |
| **...** | ... | ... | ... | ... | ... |
| **87862** | Washakie, WY | 56043 | 8,188 | 2016 | 44,204 |
| **87863** | Washakie, WY | 56043 | 8,064 | 2017 | 45,933 |
| **87864** | Weston, WY | 56045 | 6,506 | 1990 | 17,867 |

| | GeoName | GeoFips | Pop | TimePeriod | IncomePerCapita |
|---|---|---|---|---|---|
| **87865** | Weston, WY | 56045 | 6,515 | 1991 | 18,420 |
| **87866** | Weston, WY | 56045 | 6,638 | 1992 | 18,574 |
| **87867** | Weston, WY | 56045 | 6,587 | 1993 | 19,658 |
| **87868** | Weston, WY | 56045 | 6,696 | 1994 | 19,804 |
| **87869** | Weston, WY | 56045 | 6,721 | 1995 | 20,561 |
| **87870** | Weston, WY | 56045 | 6,768 | 1996 | 20,945 |
| **87871** | Weston, WY | 56045 | 6,725 | 1997 | 22,772 |
| **87872** | Weston, WY | 56045 | 6,739 | 1998 | 22,851 |
| **87873** | Weston, WY | 56045 | 6,667 | 1999 | 24,962 |
| **87874** | Weston, WY | 56045 | 6,627 | 2000 | 27,305 |
| **87875** | Weston, WY | 56045 | 6,487 | 2001 | 28,091 |
| **87876** | Weston, WY | 56045 | 6,578 | 2002 | 26,586 |
| **87877** | Weston, WY | 56045 | 6,610 | 2003 | 27,167 |
| **87878** | Weston, WY | 56045 | 6,646 | 2004 | 27,387 |
| **87879** | Weston, WY | 56045 | 6,594 | 2005 | 30,606 |
| **87880** | Weston, WY | 56045 | 6,717 | 2006 | 35,169 |
| **87881** | Weston, WY | 56045 | 7,033 | 2007 | 35,845 |
| **87882** | Weston, WY | 56045 | 7,133 | 2008 | 38,942 |
| **87883** | Weston, WY | 56045 | 7,266 | 2009 | 37,202 |
| **87884** | Weston, WY | 56045 | 7,198 | 2010 | 38,045 |
| **87885** | Weston, WY | 56045 | 7,141 | 2011 | 40,359 |
| **87886** | Weston, WY | 56045 | 7,074 | 2012 | 42,337 |
| **87887** | Weston, WY | 56045 | 7,136 | 2013 | 43,760 |
| **87888** | Weston, WY | 56045 | 7,142 | 2014 | 46,850 |
| **87889** | Weston, WY | 56045 | 7,181 | 2015 | 47,600 |
| **87890** | Weston, WY | 56045 | 7,198 | 2016 | 42,727 |
| **87891** | Weston, WY | 56045 | 6,927 | 2017 | 43,340 |

87892 rows × 5 columns

Here we change the type of TimePeriod and GeoFips in both pop_income and zillowdf into integer and string, respectively.

In [1164]:
```
pop_income['TimePeriod'] = pop_income['TimePeriod'].astype(int)
pop_income['GeoFips'] = pop_income['GeoFips'].astype(str)
```

In [1165]:
```
zillowdf['TimePeriod'] = zillowdf['TimePeriod'].astype(int)
zillowdf['GeoFips'] = zillowdf['GeoFips'].astype(str)
```

## Merging the BEA pop_income and Zillow zillow_df dataframes

In [1166]:
```
combo = pd.merge(pop_income, zillowdf,
                 on=['TimePeriod', 'GeoFips'],
                 how = 'inner',
                 indicator=True)
```

In [1167]:
```
combo = combo.dropna(axis = "rows") # removing the rows with NaN (no dat
a available), some counties with 0 still remain
```

In [1168]:
```
combo = combo[(combo['Median_Housing_Price'] != 0)]
```

In [1169]:
```
combo.set_index(['GeoFips'], inplace = True)
```

In [1170]:
```
combo = combo.drop(['_merge'], axis = 1)
```

Now, we have to update the income and population data to remove commas. This will make it easier to manipulate.

In [1171]:
```
income = combo['IncomePerCapita'].tolist()
pop = combo['Pop'].tolist()
```

In [1172]:
```
updateinc = []
for val in income:
    if ',' in val:
        val = val.replace(',', '')
        val = int(val)
        updateinc.append(val)
    else:
        updateinc.append(0)

updateinc
```

Out[1172]:
```
[22073,
 22948,
 23699,
 24682,
 24991,
 26319,
 27535,
 28766,
 29758,
 31459,
 32876,
 32603,
 33415,
 34325,
 35040,
 35464,
 36677,
 38591,
 39509,
 40484,
 24643,
 25373,
 26757,
 27119,
 27402,
 28050,
 30338,
 32292,
 34784,
 36051,
 35751,
 34698,
 36282,
 37804,
 38166,
 38212,
 39561,
 41412,
 43004,
 44079,
 19265,
 19721,
 20291,
 21339,
 21304,
 22354,
 23543,
 24379,
 24787,
 26549,
 27661,
 27381,
 27754,
 28368,
 29600,
 30188,
 31359,
```

```
31987,
32261,
33707,
17722,
18085,
18584,
19885,
20241,
21153,
22545,
23277,
24116,
26213,
27437,
27198,
28021,
28873,
29564,
30702,
31457,
32906,
32758,
34371,
18551,
19494,
19690,
21098,
21428,
22161,
23420,
24559,
25967,
27198,
27448,
27102,
27513,
28707,
29458,
29716,
30580,
31830,
32268,
33350,
23044,
24013,
24285,
25477,
26169,
27385,
29253,
29763,
30456,
32409,
34052,
34006,
35513,
36009,
```

```
36137,
37346,
38111,
39558,
40319,
42076,
20473,
20852,
21561,
21811,
22242,
23785,
25176,
26691,
28209,
29621,
30395,
30523,
31718,
32688,
32910,
33424,
34855,
35684,
36452,
37602,
19786,
20204,
20762,
21285,
21451,
22334,
23882,
23693,
23872,
25376,
26194,
25934,
26814,
25711,
27063,
27908,
28848,
29764,
29405,
30982,
22587,
23198,
24094,
24056,
25052,
26310,
27694,
28744,
30276,
31593,
33057,
```

```
33166,
34007,
35178,
35788,
35739,
36661,
38445,
39575,
40601,
20356,
20862,
21791,
22378,
23084,
23869,
25149,
25977,
26803,
28406,
29238,
29162,
30420,
30889,
31520,
31571,
32856,
34129,
34783,
36069,
15908,
17009,
17996,
18838,
19285,
20606,
22254,
23645,
25485,
26169,
26493,
27044,
26741,
27584,
27581,
30548,
28769,
30428,
31084,
32156,
23310,
24114,
24960,
26935,
27643,
28110,
30175,
31561,
```

```
33354,
34673,
34650,
33876,
35649,
36614,
36825,
36654,
37540,
38609,
39511,
40878,
28004,
29029,
31351,
31511,
32562,
33752,
36194,
37827,
40209,
41371,
42368,
40283,
42270,
43517,
45622,
44998,
46779,
48902,
49678,
52003,
21244,
21709,
22434,
23514,
23673,
24430,
25866,
27187,
28398,
29758,
29977,
29667,
31125,
32034,
32533,
32666,
34068,
34939,
35412,
36448,
19755,
20414,
21049,
21400,
22113,
```

```
23284,
24936,
25703,
27016,
28261,
29578,
28594,
29418,
30741,
31858,
32001,
33218,
34198,
35758,
36941,
21434,
21637,
22628,
23050,
23436,
25320,
25961,
27395,
29115,
30702,
32705,
32651,
33716,
35229,
36183,
36251,
36823,
38046,
39080,
40381,
43947,
45239,
46895,
48076,
49650,
21310,
21392,
22082,
23173,
23613,
24212,
25285,
27230,
29772,
30595,
31493,
30565,
31979,
33675,
33494,
33911,
34647,
```

```
35924,
36136,
37089,
26136,
27087,
28403,
28360,
29543,
30811,
32736,
33927,
35156,
35646,
35808,
35147,
35798,
37451,
37394,
37899,
39055,
40550,
41484,
43074,
19968,
21341,
21245,
22324,
23241,
24284,
26157,
27383,
27899,
29127,
29630,
30039,
31798,
32368,
31711,
31003,
33505,
34618,
35168,
36225,
18647,
19168,
19941,
21745,
22339,
23469,
23875,
24512,
25157,
25544,
25911,
26445,
28362,
29641,
```

```
29236,
28883,
29070,
30330,
30773,
32818,
20390,
20926,
21580,
22683,
22907,
23510,
25557,
27176,
28132,
29107,
30459,
29353,
29591,
30796,
32242,
32337,
33822,
34903,
35638,
36941,
31022,
32256,
33002,
35001,
35076,
35143,
37788,
40813,
42613,
44925,
43936,
41031,
41705,
43857,
45929,
46033,
48071,
50115,
50640,
52406,
18300,
18551,
19216,
20424,
21583,
22599,
24359,
25348,
25662,
26532,
27028,
```

```
            26632,
            27795,
            28775,
            29344,
            29736,
            30899,
            31674,
            32613,
            33909,
            21130,
            21453,
            21860,
            22465,
            22430,
            23776,
            25584,
            27068,
            28816,
            29873,
            30948,
            31105,
            32710,
            32860,
            34113,
            34572,
            35200,
            37261,
            38217,
            39724,
            23149,
            24148,
            25020,
            25428,
            26049,
            26798,
            27857,
            29449,
            30571,
            31857,
            32311,
            31541,
            32313,
            33244,
            34182,
            34286,
            35358,
            35932,
            36211,
            37407,
            19508,
            20031,
            19270,
            21296,
            21496,
            21857,
            23858,
            25090,
```

```
25365,
26325,
27404,
26931,
27733,
27635,
28530,
28744,
30688,
31715,
32198,
34145,
32784,
33626,
35632,
37612,
39135,
40410,
41978,
44279,
46830,
50559,
55041,
53416,
56370,
58504,
59053,
57374,
61002,
63480,
61958,
63532,
26543,
27390,
29579,
29892,
31386,
33046,
33971,
36684,
40125,
41877,
45712,
45076,
46057,
50645,
50534,
48752,
51960,
53582,
52843,
54497,
35047,
35431,
38257,
39336,
39391,
```

```
40918,
41637,
43951,
46283,
49488,
52452,
51914,
54809,
57763,
60950,
60139,
63124,
63861,
64319,
66367,
25758,
26083,
29499,
30174,
30568,
30911,
32035,
33442,
35249,
37555,
41953,
41370,
43158,
45502,
46957,
46440,
49955,
50742,
48923,
49800,
33312,
33935,
37133,
38842,
39212,
40782,
41211,
43953,
44741,
49052,
51933,
52631,
54418,
56629,
59038,
59653,
62600,
64943,
63894,
65034,
26215,
27164,
```

```
29066,
30589,
30973,
34355,
35417,
37560,
39420,
41189,
45046,
44339,
49487,
51883,
54810,
53710,
57411,
60958,
58066,
60891,
24864,
25041,
26709,
29256,
29996,
30236,
30799,
32320,
33471,
36335,
40009,
39647,
40037,
42691,
43662,
43498,
45452,
46785,
44454,
44803,
27824,
28664,
30942,
33512,
33733,
35716,
36525,
38409,
39870,
43271,
47071,
47214,
52011,
58005,
60915,
64163,
65344,
67048,
61931,
```

```
65745,
13066,
14231,
14470,
16575,
17326,
18355,
19384,
20732,
21325,
22695,
24746,
26361,
26995,
27506,
27487,
27956,
29458,
30765,
31385,
32845,
18976,
19579,
20695,
22201,
23122,
25073,
26355,
28068,
29392,
31401,
33011,
34140,
34530,
35210,
34374,
34510,
35866,
36979,
37722,
39294,
21394,
22406,
23759,
24793,
25281,
26225,
27785,
29481,
32013,
33857,
35768,
33844,
34342,
37051,
36805,
38373,
```

```
40854,
42800,
44289,
46266,
26475,
28118,
30003,
30216,
30289,
31478,
31797,
32696,
34276,
35779,
36293,
38257,
14465,
14478,
15004,
15329,
16021,
17272,
18432,
19820,
21387,
22917,
25611,
23852,
24499,
25438,
26702,
27822,
29059,
29493,
29421,
30391,
26821,
27549,
29398,
29383,
29676,
30667,
32880,
35479,
38523,
39637,
38451,
35633,
36021,
37554,
39123,
39451,
41320,
42962,
43845,
45573,
17753,
```

```
18260,
19026,
20289,
20778,
21531,
22753,
23885,
24456,
24932,
25419,
25327,
25533,
25464,
25990,
26762,
28357,
29296,
30030,
30865,
14345,
14912,
15172,
16831,
17551,
19072,
19178,
20527,
21651,
22853,
24063,
24990,
24865,
25695,
25987,
26643,
28009,
29175,
30093,
31213,
22974,
23687,
24742,
26428,
26577,
27691,
28897,
30861,
33142,
34478,
36037,
33818,
33986,
35215,
36372,
36794,
38120,
39177,
```

```
39889,
41637,
16453,
17028,
17512,
18263,
18539,
19160,
21332,
23888,
24908,
24995,
26372,
24154,
23317,
24594,
24973,
26096,
27068,
27866,
28492,
29309,
23575,
22751,
24142,
25534,
27213,
27944,
28417,
29351,
30637,
31194,
31300,
33244,
33901,
34917,
36392,
20572,
20863,
21646,
22271,
22468,
23344,
25144,
27239,
28703,
30284,
30456,
28973,
28806,
29936,
30899,
31960,
34170,
35465,
36269,
37398,
```

```
18026,
17430,
17929,
18770,
20035,
21330,
23210,
23944,
24630,
25927,
26366,
26358,
27200,
27967,
27741,
29383,
29266,
31847,
33259,
34752,
21468,
21567,
22313,
22963,
23671,
23989,
24734,
25941,
27661,
29099,
29406,
28646,
28854,
29914,
31806,
31451,
33514,
34074,
34191,
35293,
28570,
29808,
31553,
33711,
34591,
35932,
38947,
42248,
45983,
49539,
52518,
47516,
48580,
56112,
66467,
63336,
73155,
```

```
77768,
79857,
81533,
20020,
20830,
21304,
21328,
22074,
22641,
23826,
24744,
25959,
27141,
27508,
27375,
28100,
29151,
30424,
30575,
32132,
33031,
33260,
34333,
20841,
20949,
21986,
22380,
22577,
22729,
23939,
24622,
...]
```

In [1173]:
```
newpop = []
for val in pop:
    if ',' in val:
        val = val.replace(',', '')
        val = int(val)
        newpop.append(val)
    else:
        newpop.append(0)

newpop
```

Out[1173]:
```
[42106,
 42963,
 44021,
 44889,
 45909,
 46800,
 48366,
 49676,
 51328,
 52405,
 53277,
 54135,
 54750,
 55199,
 54927,
 54695,
 54864,
 54838,
 55278,
 55504,
 134444,
 137555,
 141342,
 144875,
 147957,
 151509,
 156266,
 162183,
 168121,
 172404,
 175827,
 179406,
 183110,
 186534,
 190048,
 194736,
 199064,
 202863,
 207509,
 212628,
 48824,
 50237,
 51107,
 51845,
 52551,
 53457,
 54124,
 54624,
 55485,
 56240,
 57055,
 57341,
 57381,
 57562,
 57595,
 57623,
 57546,
```

```
57590,
57562,
58013,
23351,
23605,
24006,
24182,
24403,
24591,
24887,
25031,
25466,
25553,
25636,
25854,
25973,
25993,
25958,
26014,
25897,
25741,
25766,
25857,
38069,
38862,
39897,
40197,
40683,
41079,
41534,
41946,
42318,
42881,
43239,
43484,
43661,
43688,
43601,
43645,
43779,
43706,
43830,
44067,
43509,
43667,
43580,
43716,
43930,
44315,
45053,
45604,
46228,
47458,
48516,
49440,
50203,
50434,
```

```
51113,
50647,
50652,
50974,
51217,
51874,
54928,
54715,
54997,
54793,
54536,
54279,
54380,
54359,
54413,
54486,
54473,
54426,
54525,
54534,
54569,
54528,
54461,
54358,
54327,
54500,
62325,
63593,
64650,
65678,
65861,
66530,
67260,
67794,
68495,
69450,
70228,
70864,
71169,
71353,
70944,
70905,
71034,
71198,
71216,
71617,
62551,
64668,
66160,
67459,
68659,
69935,
71018,
72978,
75246,
77124,
77698,
```

```
78632,
79581,
80006,
80220,
80555,
80562,
80912,
81240,
81677,
104367,
104002,
103286,
102976,
102988,
103025,
103080,
103174,
103528,
103893,
104206,
104239,
104457,
104354,
104257,
103865,
103374,
102942,
102726,
102755,
10067,
10004,
9902,
9866,
9893,
9805,
9686,
9675,
9361,
9385,
9359,
9182,
8997,
8907,
8852,
8752,
8587,
8510,
8488,
8330,
87469,
88256,
88943,
89490,
89940,
91321,
92593,
93903,
```

```
95846,
97727,
99183,
100500,
101790,
102468,
103362,
103625,
104148,
104249,
104173,
104346,
664457,
662845,
662033,
660197,
657518,
657513,
656023,
654919,
655893,
655163,
656510,
658441,
658048,
657789,
657668,
658727,
659544,
660087,
659747,
659197,
87113,
87590,
88039,
87791,
87627,
88103,
88732,
89358,
90294,
91107,
92202,
92526,
92735,
92603,
92652,
92688,
92983,
92422,
92319,
92538,
109936,
112898,
115430,
116819,
118392,
```

```
            120071,
            122274,
            126133,
            129247,
            131934,
            134524,
            138566,
            140806,
            144117,
            148318,
            151445,
            154211,
            156740,
            158983,
            161604,
            63729,
            64880,
            65966,
            66936,
            67677,
            68578,
            69593,
            71303,
            73651,
            76063,
            78835,
            81326,
            83177,
            85530,
            87322,
            88902,
            90616,
            91633,
            92920,
            94402,
            346373,
            349596,
            352740,
            356312,
            361046,
            398479,
            399323,
            400073,
            400129,
            398549,
            398082,
            397959,
            398942,
            402916,
            405715,
            409196,
            411994,
            413320,
            413232,
            413737,
            413929,
            414314,
```

```
414588,
414852,
413955,
224224,
223548,
223428,
223048,
223185,
222680,
223002,
223608,
227291,
227888,
227461,
227607,
229515,
229162,
228702,
227927,
227307,
227002,
226716,
226646,
29570,
29469,
29748,
29844,
29695,
30050,
30593,
30772,
31309,
31761,
32149,
32560,
32967,
32967,
33093,
33563,
33016,
33372,
33215,
33267,
50287,
49982,
49805,
49622,
49489,
49065,
49498,
49731,
50459,
51074,
51434,
52162,
53309,
54833,
```

```
            57478,
            59123,
            59207,
            58847,
            58177,
            57045,
            61730,
            63650,
            65080,
            66093,
            67274,
            68629,
            70416,
            72596,
            75948,
            79348,
            81223,
            83009,
            83814,
            84211,
            84975,
            86029,
            86219,
            86758,
            87482,
            88199,
            135954,
            140502,
            144674,
            149358,
            153976,
            159704,
            165757,
            171691,
            178841,
            183491,
            188534,
            192708,
            196017,
            197885,
            200828,
            203897,
            206152,
            208877,
            211073,
            213605,
            79107,
            79971,
            80772,
            80939,
            81095,
            80900,
            81139,
            81546,
            81991,
            82226,
            82735,
```

```
82674,
82107,
81792,
82072,
81497,
81445,
81047,
80386,
80065,
41539,
41649,
41689,
41406,
41044,
40898,
40904,
40857,
41243,
41414,
41668,
41831,
41477,
41394,
41040,
41021,
40886,
40617,
40574,
40681,
163411,
164411,
165414,
166951,
168154,
169725,
171960,
175731,
181383,
184350,
188093,
192792,
194993,
196699,
198597,
200752,
202803,
204473,
206282,
207811,
24344,
24679,
24857,
24597,
24600,
24576,
24614,
24629,
```

```
24796,
24858,
24696,
24691,
24408,
24354,
24186,
24183,
24129,
23918,
23887,
23722,
257232,
259348,
260816,
264274,
268229,
270568,
274484,
276494,
280085,
278792,
281554,
287677,
293370,
296291,
298520,
301081,
300296,
298018,
297376,
294356,
83299,
83390,
83005,
84814,
86095,
86885,
89043,
90431,
90545,
93545,
94552,
95238,
98276,
98136,
100354,
100898,
99308,
99643,
100602,
99703,
30592,
30724,
30668,
30502,
30712,
```

```
30993,
31103,
31003,
30808,
30682,
31110,
30857,
31394,
32162,
32395,
32570,
32490,
32612,
32405,
32094,
49195,
49729,
49618,
49986,
50804,
50963,
51404,
51590,
52253,
53272,
53655,
54796,
55561,
56304,
56910,
56993,
57585,
58067,
58543,
58617,
14372,
14302,
13993,
13789,
13592,
13328,
13237,
13298,
13485,
13334,
13319,
13294,
13544,
13644,
13700,
13682,
13780,
13680,
13727,
13856,
13850,
13760,
```

```
13983,
13748,
13651,
13239,
13206,
13105,
13184,
13159,
13262,
13505,
13652,
13820,
14054,
14078,
13906,
13753,
13679,
13448,
55663,
57824,
59802,
61807,
64353,
67162,
70761,
74409,
78633,
81402,
84079,
86885,
89738,
91721,
93681,
95853,
98104,
101064,
104123,
106532,
8807,
8805,
8845,
8727,
8874,
8881,
8890,
8994,
9046,
8851,
8835,
8844,
8883,
8888,
9022,
8962,
8844,
8786,
8768,
```

```
8689,
69851,
69801,
69507,
67863,
67319,
68072,
68161,
68521,
69390,
69602,
69883,
71008,
71838,
72179,
72215,
72292,
71762,
70979,
71373,
71606,
116091,
116530,
118132,
118798,
119847,
120638,
123234,
125786,
127241,
128206,
129023,
130081,
131782,
132952,
131826,
129332,
127054,
126084,
125355,
124756,
114874,
115307,
116773,
118283,
121308,
122882,
125117,
127025,
128695,
130442,
131853,
133477,
134612,
134181,
136002,
136448,
```

```
137209,
138682,
140079,
140776,
52541,
53252,
53437,
53561,
53572,
53422,
52973,
52982,
53008,
52935,
53297,
53501,
32795,
33279,
33511,
33356,
33224,
32985,
32703,
32964,
33701,
35175,
36639,
37525,
37158,
37124,
37009,
37419,
38064,
37786,
37767,
37466,
2909040,
3004985,
3092197,
3175989,
3255388,
3328468,
3417860,
3538988,
3642884,
3711954,
3771061,
3803779,
3824644,
3869626,
3939776,
4006307,
4076708,
4154076,
4233383,
4307033,
145155,
```

```
150351,
156215,
160312,
166155,
172633,
180521,
188773,
196168,
199760,
200078,
199696,
200315,
202733,
203262,
203026,
203320,
204524,
205385,
207200,
94824,
96100,
97689,
97849,
100123,
101529,
102774,
104184,
105624,
106927,
107626,
107571,
107695,
107518,
107268,
107077,
107613,
107543,
108322,
108956,
813386,
828905,
848019,
859280,
874267,
885893,
901342,
920298,
940930,
955869,
967778,
975580,
981540,
987297,
991528,
994759,
1000916,
1005323,
```

```
1012519,
1022769,
165492,
173364,
181280,
187747,
197082,
207920,
219472,
235708,
271328,
306174,
335311,
349830,
379504,
378226,
381910,
384258,
394004,
404069,
415115,
430237,
40625,
41623,
42961,
44298,
45338,
46144,
47011,
47425,
47615,
47292,
46845,
46468,
46191,
46075,
46212,
157686,
162943,
168608,
172636,
177362,
182090,
187822,
195424,
204082,
208773,
211211,
211172,
210969,
210883,
211809,
214198,
217509,
220621,
224363,
228168,
```

```
149065,
155665,
160576,
162873,
165398,
168003,
172824,
178816,
183848,
187357,
191202,
193714,
197124,
202581,
202105,
201810,
203039,
203558,
205463,
207534,
37619,
38055,
38446,
38331,
38506,
38772,
39334,
39992,
40867,
41491,
41519,
41561,
41511,
41341,
41076,
40976,
40871,
41118,
41145,
41355,
142496,
148636,
154744,
159258,
164926,
171742,
179010,
186756,
195569,
203664,
210807,
216620,
222558,
228827,
234315,
238902,
244192,
```

```
251016,
258704,
266300,
33210,
33436,
34051,
34258,
34448,
34554,
34900,
35492,
36056,
36759,
36974,
36844,
36882,
37063,
37308,
37305,
37090,
37091,
37191,
37381,
23326,
23806,
24100,
24204,
24375,
24711,
25018,
25274,
...]
```

In [1174]:
```
combo['IncomePerCapita'] = updateinc
combo['Pop'] = newpop
```

## Housing Affordability Index

We then calculated our Housing Affordability Index, which was based off of a similar formula from the Federal Reserve Bank of San Francisco (https://www.frbsf.org/education/publications/doctor-econ/2003/december/housing-affordability-index/ (https://www.frbsf.org/education/publications/doctor-econ/2003/december/housing-affordability-index/)). It's calculated as income per capita / median housing price - the higher the index value, the more 'affordable' the typical home is in that specific county.

In [1175]:
```
#combo['IncomePerCapita'] = pop_income['IncomePerCapita'].astype(int)
#combo['Avg Housing Price'] = pop_income['Avg Housing Pric'].astype(floa
t)

combo['Housing_Affordability_Index'] = ((combo['IncomePerCapita'].astype
(int) / combo['Median_Housing_Price'].astype(float)) * 100)
```

In [1176]:
```
combo['Housing_Affordability_Index'] = np.round(combo['Housing_Affordability_Index'], decimals=2)
```

In [1177]:
```
combo = combo[['RegionName','TimePeriod', 'Pop', 'IncomePerCapita', 'Median_Housing_Price', 'Housing_Affordability_Index']]
```

## Combined (combo) dataframe

Combo presents a big picture overview for all counties in the US (for all years 1998-2017) and the corresponding statistics

In [1178]:
```
combo.head()
```

Out[1178]:

| GeoFips | RegionName | TimePeriod | Pop | IncomePerCapita | Median_Housing_Price | Housing_Affo |
|---|---|---|---|---|---|---|
| **01001** | Autauga County | 1998 | 42106 | 22073 | 110558.33 | |
| **01001** | Autauga County | 1999 | 42963 | 22948 | 111250.00 | |
| **01001** | Autauga County | 2000 | 44021 | 23699 | 115125.00 | |
| **01001** | Autauga County | 2001 | 44889 | 24682 | 109325.00 | |
| **01001** | Autauga County | 2002 | 45909 | 24991 | 118983.33 | |

In [1179]:
```
combo['Pop_Change'] = combo['Pop'].pct_change()
combo['Change_in_Housing_Price'] = combo['Median_Housing_Price'].pct_change()
combo['Change_in_IncomePerCapita'] = combo['IncomePerCapita'].pct_change()
```

In [1180]:
```
combo = combo[['RegionName','TimePeriod','Pop', 'Pop_Change', 'IncomePerCapita', 'Change_in_IncomePerCapita', 'Median_Housing_Price', 'Change_in_Housing_Price', 'Housing_Affordability_Index']]
```

In [1181]:
```
combo.head()
```

Out[1181]:

| GeoFips | RegionName | TimePeriod | Pop | Pop_Change | IncomePerCapita | Change_in_IncomePerC |
|---|---|---|---|---|---|---|
| **01001** | Autauga County | 1998 | 42106 | NaN | 22073 | |
| **01001** | Autauga County | 1999 | 42963 | 0.020353 | 22948 | 0.0 |
| **01001** | Autauga County | 2000 | 44021 | 0.024626 | 23699 | 0.0 |
| **01001** | Autauga County | 2001 | 44889 | 0.019718 | 24682 | 0.0 |
| **01001** | Autauga County | 2002 | 45909 | 0.022723 | 24991 | 0.0 |

Converting to excel so that we can doublecheck our changes.

In [1182]:
```
combo.to_excel("Combo.xlsx") # converting to excel so we can doublecheck our changes
```

## Case Studies

From combo, we can now pull in specific counties to further analyze and visualize. Instead of randomly selecting counties to work with, we decided to focus on three counties - the District of Columbia, El Paso County in Colorado, and Philadelphia County in Pennsylvania. We decided on these three cases by looking at the Housing Affordability Index in the combo dataframe and finding the counties behind the median (50th percentile), 10th percentile, and 90th percentile values (by cross referencing the HAI value with the corresponding GeoFips value and Population Income data).

In [1183]:
```
combo['Housing_Affordability_Index'].quantile(0.5) # District of Columbia
```

Out[1183]: 28.2

In [1184]:
```
combo['Housing_Affordability_Index'].quantile(0.10) # El Paso County (Colorado)
```

Out[1184]: 16.79

In [1185]:
```
combo['Housing_Affordability_Index'].quantile(0.90) # Philadelphia County (Pennsylvania)
```

Out[1185]: 45.28

But before we get into the county cases, let's start with some simple big-picture visualization of US housing.

# Data Visualization

## US Final Combination

USfincombo consolidates US-related data on Income per Capita, Median Housing Price, and the Housing Affordability Inde for the years 1998-2017. We grouped by Time Period and used the .mean() function to reach an average annual value for the above metrics for each year.

In [1186]:
```
fincombo = combo.groupby(['TimePeriod']).mean()
```

In [1187]:
```
fincombo
```

Out[1187]:

| TimePeriod | Pop | Pop_Change | IncomePerCapita | Change_in_IncomePerCapita | Median_I |
|---|---|---|---|---|---|
| **1998** | 151721.177207 | 3.135787 | 23460.142142 | -0.438591 | |
| **1999** | 153613.639950 | 0.011814 | 24299.634314 | 0.034764 | |
| **2000** | 155487.721979 | 0.011541 | 25701.674389 | 0.055603 | 1 |
| **2001** | 157204.631183 | 0.008374 | 26794.034440 | 0.044549 | 1 |
| **2002** | 158807.139011 | inf | 27143.541014 | inf | 1 |
| **2003** | 159586.186567 | 0.010597 | 27937.487562 | 0.027547 | 1 |
| **2004** | 160928.310174 | 0.010681 | 29301.782258 | 0.048238 | 1 |
| **2005** | 161391.179487 | 0.031931 | 30506.477411 | 0.034390 | 1 |
| **2006** | 161618.512063 | 0.019438 | 32167.052473 | 0.047141 | 1 |
| **2007** | 162929.939230 | 0.011088 | 33713.093863 | 0.048018 | 1 |
| **2008** | 164030.576139 | 0.007668 | 35086.889089 | 0.041805 | 1 |
| **2009** | 165543.913721 | 0.010454 | 34137.313361 | -0.022392 | 1 |
| **2010** | 152976.039804 | -0.006049 | 35208.995638 | 0.015767 | 1 |
| **2011** | 153294.287493 | 0.004280 | 37273.743909 | 0.056052 | 1 |
| **2012** | 154177.101457 | 0.002934 | 38728.202914 | 0.036577 | 1 |
| **2013** | 154952.056897 | 0.007144 | 39341.314010 | 0.017044 | 1 |
| **2014** | 155612.855235 | 0.014567 | 40766.502137 | 0.035378 | 1 |
| **2015** | 156549.704848 | 0.004438 | 41910.648908 | 0.029707 | 1 |
| **2016** | 157818.174747 | 0.003418 | 42364.574853 | 0.011673 | 1 |
| **2017** | 159049.982419 | 0.004294 | 43647.781034 | 0.030301 | 1 |

In [1188]:
```
fincombo = fincombo.drop(['Pop','Pop_Change','Change_in_IncomePerCapita'
,'Change_in_Housing_Price'], axis = 1)
```

In [1189]:
```
fincombo
```

Out[1189]:

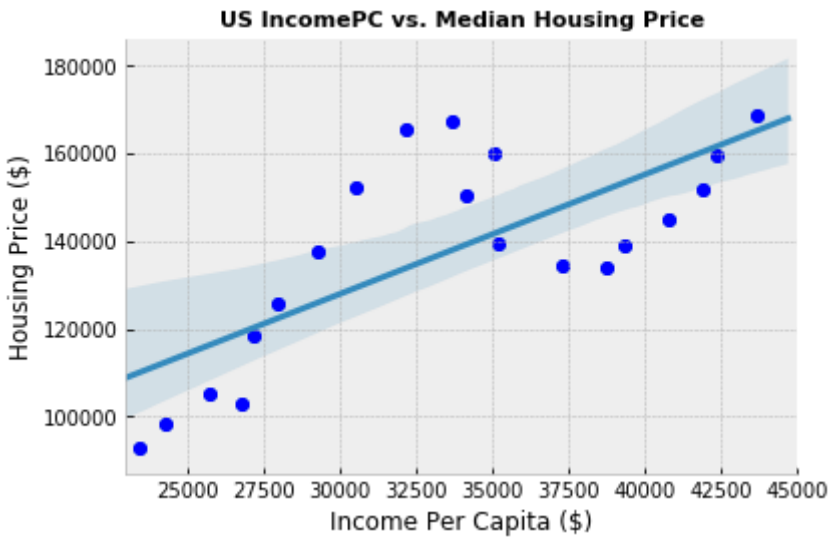
| TimePeriod | IncomePerCapita | Median_Housing_Price | Housing_Affordability_Index |
|---|---|---|---|
| **1998** | 23460.142142 | 92587.095711 | 28.775385 |
| **1999** | 24299.634314 | 98366.864903 | 27.961127 |
| **2000** | 25701.674389 | 105051.450683 | 27.955554 |
| **2001** | 26794.034440 | 102757.112254 | 30.122686 |
| **2002** | 27143.541014 | 118211.104214 | 26.807032 |
| **2003** | 27937.487562 | 125811.251082 | 26.394652 |
| **2004** | 29301.782258 | 137475.434349 | 25.888852 |
| **2005** | 30506.477411 | 152346.138645 | 25.071758 |
| **2006** | 32167.052473 | 165417.239644 | 25.026647 |
| **2007** | 33713.093863 | 167388.031450 | 25.465836 |
| **2008** | 35086.889089 | 159800.409646 | 27.486613 |
| **2009** | 34137.313361 | 150465.508209 | 27.909023 |
| **2010** | 35208.995638 | 139538.545120 | 31.833081 |
| **2011** | 37273.743909 | 134466.675734 | 34.438652 |
| **2012** | 38728.202914 | 134108.976395 | 35.673162 |
| **2013** | 39341.314010 | 138984.402292 | 35.667649 |
| **2014** | 40766.502137 | 144871.425321 | 35.662137 |
| **2015** | 41910.648908 | 151658.404379 | 35.075370 |
| **2016** | 42364.574853 | 159363.350192 | 33.760016 |
| **2017** | 43647.781034 | 168907.907214 | 32.968295 |

## US IncomePC vs. Median Housing Price

In the scatter plot below, you can see that across the entire US, there does seem to be a linear relationship between income per capita and the median housing price. When US income per capita rises, whether due to strong economic growth or other factors, the associated median housing price typically also rises. Since 1998, the median home price has also increased from under 100,000 dollars to almost 170,000 dollars as of 2017. This increase has been accompanied by an 86% jump in the US income per capita over the same time frame, representative of the proportional relationship between these two variables. This association makes sense, as when times are good and people are doing well, income per capita will increase. With higher incomes, they may look to invest or put that money towards real estate, contributing to rising housing prices.

In [1190]:
```
fig, ax = plt.subplots()

sns.regplot(fincombo.IncomePerCapita, fincombo.Median_Housing_Price)

ax.scatter(fincombo.IncomePerCapita, fincombo.Median_Housing_Price, colo
r = 'b', linewidth = '1')
ax.set_title("US IncomePC vs. Median Housing Price", fontsize = 11, font
weight = "bold")
ax.set_ylabel("Housing Price ($)")
ax.set_xlabel("Income Per Capita ($)")

ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

ax.set_xlim(23000,45000)
#ax.set_ylim(-0.8,0.8)

plt.show()
```

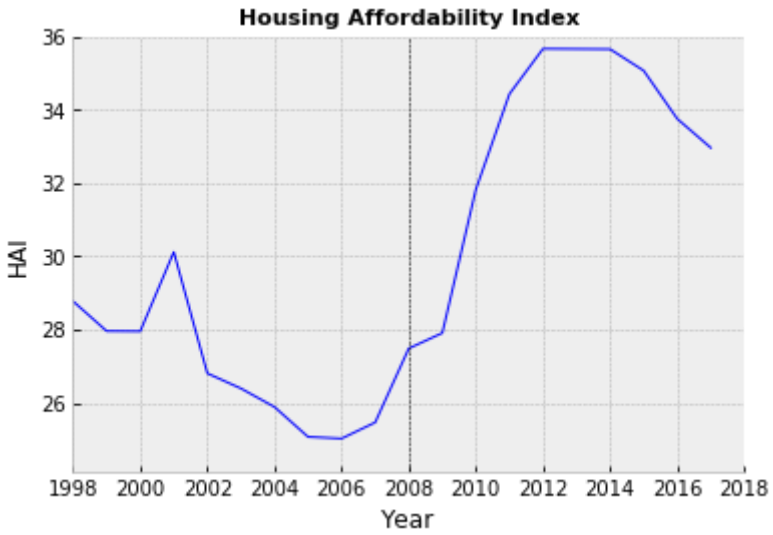
```
/Users/cole/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:
1713: FutureWarning: Using a non-tuple sequence for multidimensional in
dexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In t
he future this will be interpreted as an array index, `arr[np.array(se
q)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```




## Housing Affordability Index Time-Series

Below, we graphed the Housing Affordability Index for the entire US over the period 1998-2017. From the line plot, we can tell that the US HAI has been relatively positive, peaking around 2012-2014 at a value of about 35. In the years leading up to the subprime mortgage bubble and the financial crisis of 2008 (which is demarcated with a dotted black line), it's clear that the Affordability Index took a hit, with consecutive years of decreases. As the US economy has recovered since 08 however, the Affordability Index has moved higher, up until the last couple of years where once again, housing affordability has declined.

In [1191]:
```
fig, ax = plt.subplots()

ax.plot(fincombo.index, fincombo.Housing_Affordability_Index, color =
'b', linewidth = '1')
ax.set_title("Housing Affordability Index", fontsize = 11, fontweight =
"bold")
ax.set_ylabel("HAI")
ax.set_xlabel("Year")

ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

plt.xticks(np.arange(1998,2019,2))

plt.axvline(x=2008, linestyle = '--', color = 'black', linewidth = '0.5'
)


ax.set_xlim(1998,2018)
ax.set_ylim(24.1,36)

plt.show()
```




Now we'll begin looking at the same metrics for specific US counties:

## 50th percentile (median) of Housing Affordability Index

First, we tested the District of Columbia, which was the US county with the median Housing Affordability Index value (seen in cells above). Below, we plotted its Population Growth Rate against its Change in Median Housing Price, as well a 19-year time series of its Housing Affordability Index.

In [1192]:
```
dc = combo.loc['11001']
```

In [1193]:
```
dc.set_index('TimePeriod',inplace = True)
```

In [1194]:
```
dc.head()
```

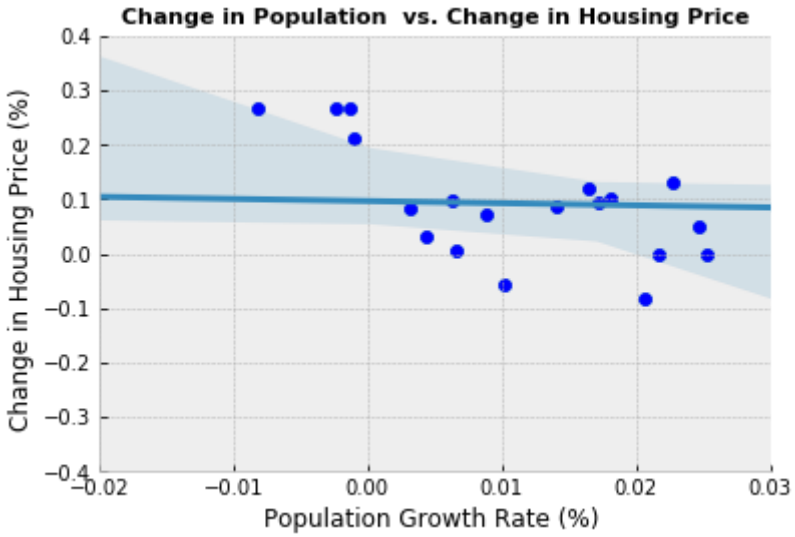
Out[1194]:

| TimePeriod | RegionName | Pop | Pop_Change | IncomePerCapita | Change_in_IncomePerCapita | M |
|---|---|---|---|---|---|---|
| **1998** | District of Columbia | 565230 | 1.508543 | 38083 | -0.195271 | |
| **1999** | District of Columbia | 570213 | 0.008816 | 39480 | 0.036683 | |
| **2000** | District of Columbia | 572046 | 0.003215 | 43329 | 0.097492 | |
| **2001** | District of Columbia | 574504 | 0.004297 | 44684 | 0.031272 | |
| **2002** | District of Columbia | 573158 | -0.002343 | 45080 | 0.008862 | |

## Population Growth vs. Housing Price Change

In the scatter plot below, we visualized the relationship between population change and housing price change for a specific county, the District of Columbia. This county had the median (50th percentile) tracked Housing Affordability Index value (from before). There is a very slight inverse relationship between the two variables, demonstrated by the line of best fit - as population decreases, housing prices inch upwards and vice versa. Although this is a trend that we didn't expect, there are many hidden variables that may be at play, including the incomes of those people that move into the DC area as well as those of people that leave. With more data points, the relationship might be easier to establish.

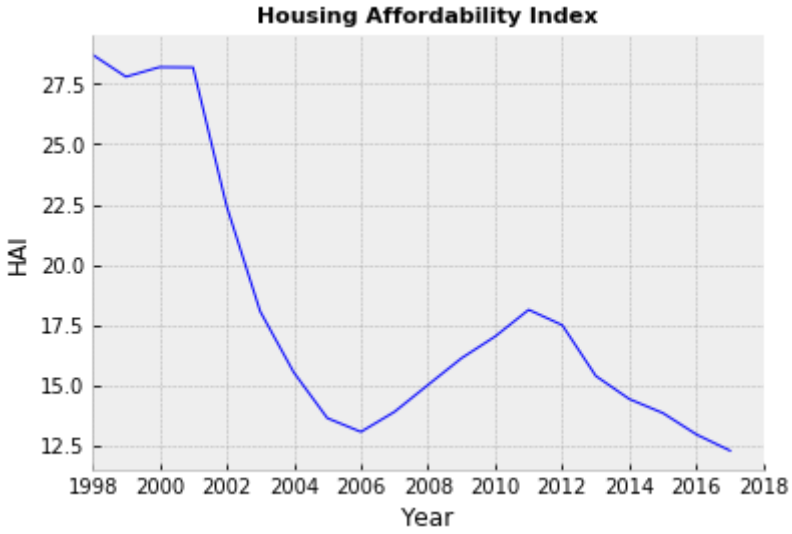
In [1195]:
```
fig, ax = plt.subplots()

sns.regplot(dc.Pop_Change, dc.Change_in_Housing_Price)

plt.style.use('bmh')

ax.scatter(dc.Pop_Change, dc.Change_in_Housing_Price, color = 'b', linew
idth = '0.5')
ax.set_title("Change in Population  vs. Change in Housing Price", fontsi
ze = 11, fontweight = "bold")
ax.set_ylabel("Change in Housing Price (%) ")
ax.set_xlabel("Population Growth Rate (%) ")

ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

ax.set_xlim(-0.02,0.03)
ax.set_ylim(-0.4,0.4)

plt.show() #VISUALIZING LINEAR RELATIONSHIPS
```




## Housing Affordability Time-Series

The line plot below displays the Housing Affordability Index over our project's 19 year period. In the couple of years preceding the mortgage bubble and crash, affordability is steadily declining in the District of Columbia. The index peaked around 2000, and has been in a slump recently, with HAI hitting its lowest level.

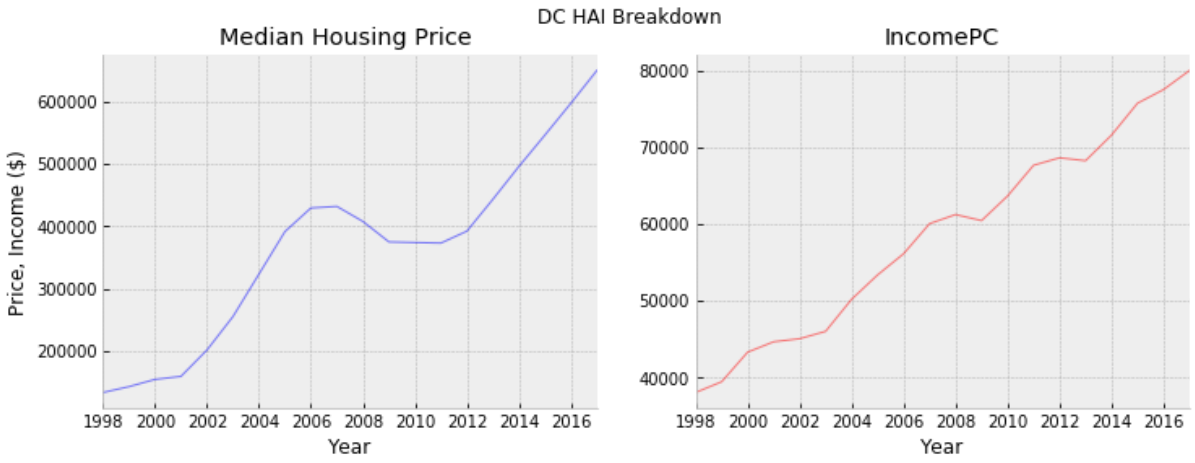
In [1196]:
```
fig, ax = plt.subplots()

ax.plot(dc.index, dc.Housing_Affordability_Index, color = 'b', linewidth
= '1')
ax.set_title("Housing Affordability Index", fontsize = 11, fontweight =
"bold")
ax.set_ylabel("HAI")
ax.set_xlabel("Year")

ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)

plt.xticks(np.arange(1998,2019,2))
ax.set_xlim(1998,2018)

plt.show()
```




## District of Columbia HAI Breakdown

Below, is a side-by-side breakdown of the HAI visual. It is split into the two components of the Housing Affordability Index, being income per capita and the median housing price. Incomes in the county have continued their steady rise, even through the chaos of the recession, but housing prices have had dips, especially around 2010-12 when the housing market bottomed out. This breakdown provides some more detailed insight into the HAI line plot above.

In [1197]:
```
fig, ax = plt.subplots(nrows = 1, ncols = 2, sharex = True, figsize = (1
2,4))

plt.xticks(np.arange(1998,2019,2))
ax[0].plot(dc.index, dc.Median_Housing_Price, color = "b", linewidth = 1
,
        alpha = 0.5)

ax[1].plot(dc.index, dc.IncomePerCapita, color = "red", linewidth = 1,
        alpha = 0.5)

ax[0].set_title("Median Housing Price ")
ax[1].set_title("IncomePC")

ax[0].set_ylabel("Price, Income ($)")



fig.suptitle("DC HAI Breakdown")

for var in ax:

    var.set_xlim(1998,2017)

    var.spines["right"].set_visible(False)
    var.spines["top"].set_visible(False)

    var.set_xlabel("Year")

    #var.legend(frameon = False)

plt.show()
```




## 10th and 90th percentile Housing Affordability Index

Side-by-side comparison

El Paso County in Colorado is the 10th percentile in our Housing Affordability Index distribution.

In [1198]:
```
elpaso = combo.loc['08041']
elpaso.set_index('TimePeriod',inplace = True)
```

In [1199]:
```
elpaso.head()
```

Out[1199]:

| TimePeriod | RegionName | Pop | Pop_Change | IncomePerCapita | Change_in_IncomePerCapita | M |
|---|---|---|---|---|---|---|
| **1998** | El Paso County | 498062 | 18.423680 | 27170 | -0.507674 | |
| **1999** | El Paso County | 509044 | 0.022049 | 28697 | 0.056202 | |
| **2000** | El Paso County | 519938 | 0.021401 | 30783 | 0.072691 | |
| **2001** | El Paso County | 535864 | 0.030631 | 31177 | 0.012799 | |
| **2002** | El Paso County | 544145 | 0.015454 | 31376 | 0.006383 | |

Philadelphia County in Pennsylvania is in the 90th percentile of our Housing Affordability Index distribution.

In [1200]:
```
philly = combo.loc['42101']
philly.set_index('TimePeriod', inplace = True)
```

In [1201]:
```
philly.head()
```

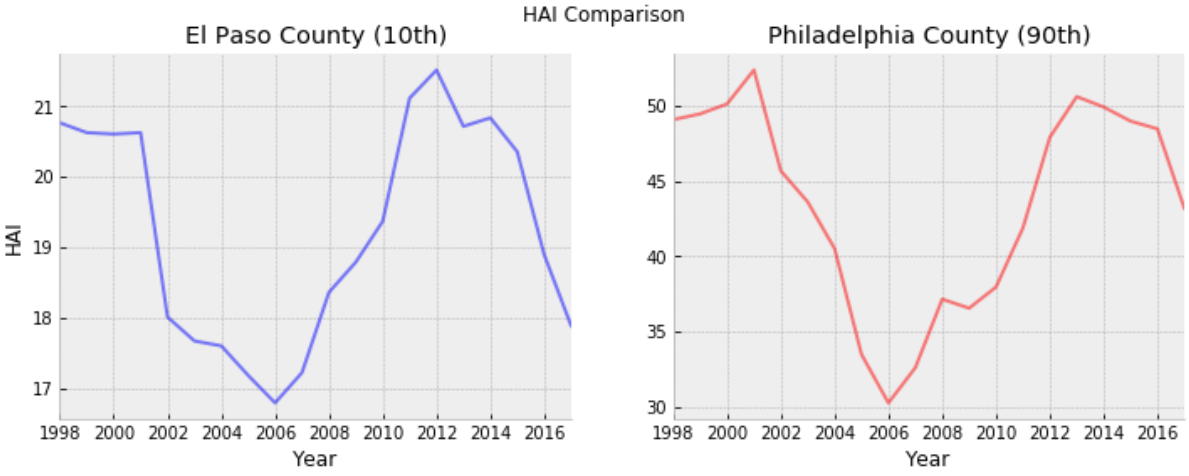
Out[1201]:

| TimePeriod | RegionName | Pop | Pop_Change | IncomePerCapita | Change_in_IncomePerCapita | I |
|---|---|---|---|---|---|---|
| **1998** | Philadelphia County | 1525955 | 32.081601 | 22957 | -0.457358 | |
| **1999** | Philadelphia County | 1520064 | -0.003861 | 23719 | 0.033192 | |
| **2000** | Philadelphia County | 1514563 | -0.003619 | 25424 | 0.071883 | |
| **2001** | Philadelphia County | 1505455 | -0.006014 | 25785 | 0.014199 | |
| **2002** | Philadelphia County | 1498493 | -0.004625 | 26513 | 0.028233 | |

## HAI Comparison between El Paso County and Philadelphia County

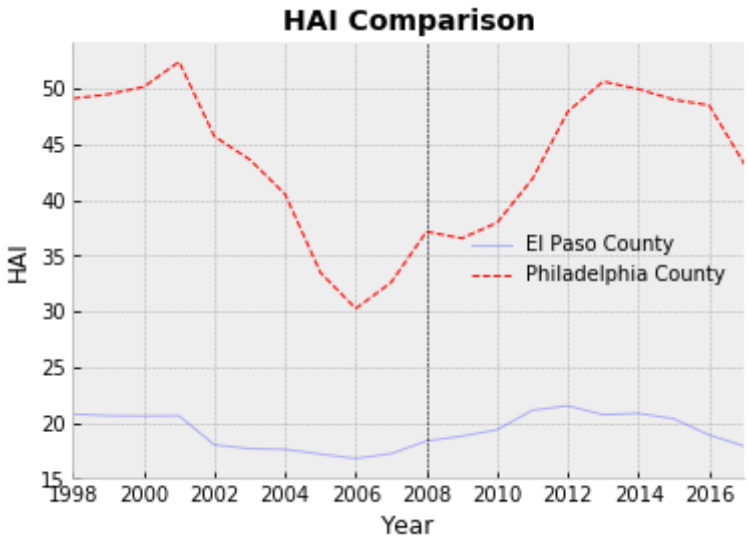
Below, we compared the housing affordability index values of both counties against each other in order to see the differences between a county that is in the 90th percentile of affordability, and one that is in the bottom 10th percentile. Both counties were negatively impacted by the recession, and they recovered equally as fast afterwards. Philly County's affordability index value has been consistently above that of El Paso County, with a peak of over 50 that it most recently hit in 2012. After that year, when the housing market turned a corner, both counties saw declines in their Housing Affordability Index, but Philly County was more vulnerable, seeing its HAI drop more dramatic. It's interesting to see the differences in the volatility between the two counties simply based off of their history of housing affordability.

In [1202]:
```
fig, ax = plt.subplots(nrows = 1, ncols = 2, sharex = True, figsize = (12,4))

plt.xticks(np.arange(1998,2019,2))
ax[0].plot(elpaso.index, elpaso.Housing_Affordability_Index, color = "b", linewidth = 2,
        alpha = 0.5)

ax[1].plot(philly.index, philly.Housing_Affordability_Index, color = "red", linewidth = 2,
        alpha = 0.5)

ax[0].set_title("El Paso County (10th)")
ax[1].set_title("Philadelphia County (90th)")

ax[0].set_ylabel("HAI")


fig.suptitle("HAI Comparison")

for var in ax:

    var.set_xlim(1998,2017)
    #var.set_ylim(16,55)

    var.spines["right"].set_visible(False)
    var.spines["top"].set_visible(False)

    var.set_xlabel("Year")

    #var.legend(frameon = False)

plt.show()
```




Another perspective on the data presented above:

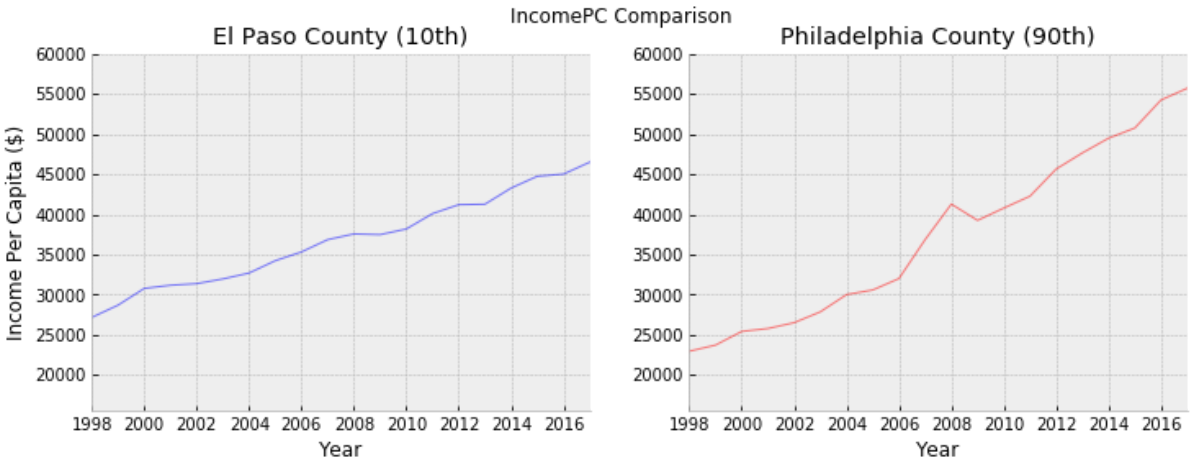
In [1207]:
```
fig, ax = plt.subplots()


plt.xticks(np.arange(1998,2018,2))

ax.plot(elpaso.index, elpaso.Housing_Affordability_Index, color = "b", l
inewidth = 1,
        alpha = 0.25, label = "El Paso County")

ax.plot(philly.index, philly.Housing_Affordability_Index, color = "r", l
inewidth = 1,
        linestyle = '--', label = "Philadelphia County")

ax.set_title("HAI Comparison", fontsize = 14, fontweight = "bold")
plt.axvline(x=2008, linestyle = '--', color = 'black', linewidth = '0.5'
)


ax.legend(frameon = False)

ax.set_ylabel("HAI")
ax.set_xlabel("Year")

ax.set_xlim(1998,2017)

ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)


plt.show()
```



## Income PC Comparison

The comparison graphs below look at any differences between IncomePC across the two counties. Although both counties start off with similar income per capita in 1998, the gap grows more apparent over time. Philadelphia County, which was in the 90th percentile for the Housing Affordability Index, sees income per capita growth that is much quicker than that experienced by El Paso County. After almost 20 years, income per capita is almost 10,000 dollars higher in Philly County compared to El Paso County, which is a sizable difference when looking at housing prices.

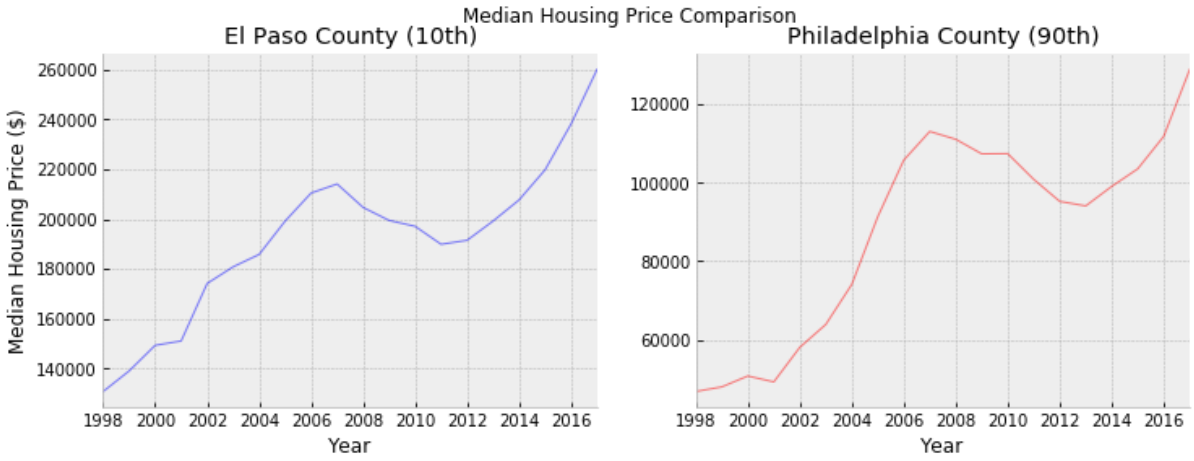
In [1204]:
```
fig, ax = plt.subplots(nrows = 1, ncols = 2, sharex = True, figsize = (1
2,4))

plt.xticks(np.arange(1998,2019,2))
ax[0].plot(elpaso.index, elpaso.IncomePerCapita, color = "b", linewidth
= 1,
        alpha = 0.5)

ax[1].plot(philly.index, philly.IncomePerCapita, color = "red", linewidt
h = 1,
        alpha = 0.5)

ax[0].set_title("El Paso County (10th)")
ax[1].set_title("Philadelphia County (90th)")

ax[0].set_ylabel("Income Per Capita ($)")


fig.suptitle("IncomePC Comparison")

for var in ax:

    var.set_xlim(1998,2017)
    var.set_ylim(15500,60000)

    var.spines["right"].set_visible(False)
    var.spines["top"].set_visible(False)

    var.set_xlabel("Year")

    #var.legend(frameon = False)

plt.show()
```



## Median Housing Price Comparison

The line plots below breakdown the HAI visual from above to focus solely on changes in median housing prices in both counties. From the graphs, we can see that the prices that homes started at are very different between El Paso and Philadelphia Counties. In El Paso, housing prices were already over 130,000 dollars in 1998, and have nearly doubled by 2017. While Philly County has seen three times that increase, its median housing prices in 2017 are still substantially lower than similar home prices in El Paso. This may speak to the desirability of both counties for people, which would cause prices to rise. Both counties saw median housing prices peak before crashing in '08, following roughly the same general trend. Philadelphia County's rapid housing price growth in recent years is worth further exploration.

In [1205]:
```
fig, ax = plt.subplots(nrows = 1, ncols = 2, sharex = True, figsize = (1
2,4))

plt.xticks(np.arange(1998,2019,2))
ax[0].plot(elpaso.index, elpaso.Median_Housing_Price, color = "b", linew
idth = 1,
        alpha = 0.5)

ax[1].plot(philly.index, philly.Median_Housing_Price, color = "red", lin
ewidth = 1,
        alpha = 0.5)

ax[0].set_title("El Paso County (10th)")
ax[1].set_title("Philadelphia County (90th)")

ax[0].set_ylabel("Median Housing Price ($)")


fig.suptitle("Median Housing Price Comparison")

for var in ax:

    var.set_xlim(1998,2017)

    var.spines["right"].set_visible(False)
    var.spines["top"].set_visible(False)

    var.set_xlabel("Year")

    #var.legend(frameon = False)

plt.show()
```




We also wrote a function by which any user can, by entering a valid GeoFips code, receive a time-series of the Population change, Median Housing Price change, Income Per Capita change, or Housing Affordability Index change. It is called geo_lookup().

## Lookup by GeoFips

In [1226]:
```
def geo_lookup():

    #user_input = input("Please enter a valid GeoFips code")

    try:
        user_input = input("Please enter a valid GeoFips code: ")
    except KeyError:
        user_input = input("Please enter a valid GeoFips code: ")


    new_combo = combo.loc[user_input]

    new_combo.set_index('TimePeriod', inplace = True)

    new_input = input("Please enter one of the following options: Pop, I
ncomePerCapita, Median_Housing_Price, Housing_Affordability_Index")

    if new_input == 'Pop':

        fig, ax = plt.subplots()

        ax.plot(new_combo.index.astype(int), new_combo.Pop, color = 'b',
linewidth = '1')
        ax.set_title("Population", fontsize = 11, fontweight = "bold")
        ax.set_ylabel("Population")
        ax.set_xlabel("Year")

        ax.spines["right"].set_visible(False)
        ax.spines["top"].set_visible(False)

        plt.xticks(np.arange(1998,2019,2))
        ax.set_xlim(1998,2018)

    elif new_input == 'IncomePerCapita':

        fig, ax = plt.subplots()

        ax.plot(new_combo.index.astype(int), new_combo.IncomePerCapita,
color = 'b', linewidth = '1')
        ax.set_title("IncomePC", fontsize = 11, fontweight = "bold")
        ax.set_ylabel("Income Per Capita ($)")
        ax.set_xlabel("Year")

        ax.spines["right"].set_visible(False)
        ax.spines["top"].set_visible(False)

        plt.xticks(np.arange(1998,2019,2))
        ax.set_xlim(1998,2018)

    elif new_input == 'Median_Housing_Price':

        fig, ax = plt.subplots()

        ax.plot(new_combo.index.astype(int), new_combo.Median_Housing_Pr
ice, color = 'b', linewidth = '1')
        ax.set_title("Median Housing Price", fontsize = 11, fontweight =
```

```
"bold")
        ax.set_ylabel("Median Housing Price ($)")
        ax.set_xlabel("Year")

        ax.spines["right"].set_visible(False)
        ax.spines["top"].set_visible(False)

        plt.xticks(np.arange(1998,2019,2))
        ax.set_xlim(1998,2018)

    elif new_input == 'Housing_Affordability_Index':

        fig, ax = plt.subplots()

        ax.plot(new_combo.index.astype(int), new_combo.Housing_Affordabi
lity_Index, color = 'b', linewidth = '1')
        ax.set_title("Housing Affordability Index", fontsize = 11, fontw
eight = "bold")
        ax.set_ylabel("Housing Affordability Index")
        ax.set_xlabel("Year")

        ax.spines["right"].set_visible(False)
        ax.spines["top"].set_visible(False)

        plt.xticks(np.arange(1998,2019,2))
        ax.set_xlim(1998,2018)

    else:
        new_input = input("Please enter one of the following options: Po
p, IncomePerCapita, Median_Housing_Price, Housing_Affordability_Index")
```

# Conclusion

## Final Thoughts

From this experiment, nothing particularly ground-breaking was found. It's not surprising that there is some level of relationship between the three variables we outlined in the beginning of our project, income per capita, population growth, and median housing prices. It was interesting to see in the US-wide analysis between income per capita and housing prices how a direct linear relationship seemed to exist. Given how the US has entered a rough few years where the Housing Affordability Index has dropped, but income per capita has continued to gradually improve, it's worth thinking about how housing prices (over 180,000 dollars now) have outpaced income per capita growth. This ties into many political and economic questions surrounding stagnating wage growth around the US and areas that are suffereing from lack of adequate economic opportunity. The case-by-case comparison of the 10th and 90th percentile (by HAI) of El Paso County and Philadelphia County was also very interesting to see visualized, although the insights were not extraordinary. The differences in home price volatility were especially telling as well as how the two counties absorbed the effects of the 2008 recession and rebounded afterwards.

For more meaningful insights, data can be drawn from a wider time horizon and across a larger range of variables, beyond just the three that we tested. Many other factors contribute to a given county/area's housing prices, and many of those variables are also quantifiable in some way. Expanding the time horizon of this project to maybe 50 years may also provide more data points and make clearer any relationships between variables. A rigorous statistical analysis could also be applied to any future data exploration so that assumed relationships can be verified as statistically significant or not. It's always important to keep in mind that correlation doesn't mean causation.

Thank you for spending the time to read this!