
How do factors such as Location, Age of the House and Average Occupancy steer consumer's decision to adjust one's budget into purchasing a house

---

Project 1

Introduction

This project aims to scrutinize the discourse inside the world of data analysis using Python based tools to understand and possibly answer questions related to the housing market of California. At the end of our analysis, we should be able to answer questions that can help a prospective buyer to provide his preferences and constraints and get answers to help him decide on buying his/her dream house.

Project Overview:

Housing Prices is an intriguing yet familiar topic which is touched upon time and again in an attempt to grasp and derive the relationship between the mechanism of pricing houses and factors affecting these very prices. Often, the results drawn from this analysis turn out to be skewed and non-representative of the original population. With the aid of the dataset provided below named 'California Housing Prices in 1990', we wish to re-analyse the data provided within the Dataframe with a macro level view of perspective and depict the resulting correlation.

Our objective is to ascertain how do various independent variables such as the age of the house, ocean proximity, and Average Occupancy(Rooms Per Person) steer a consumer’s decision to adjust his budget into purchasing a house. We wish to do the same by visualizing each of our handpicked independent as well as dependent variables and plotting them against each other using meaningful visualization techniques and graphical tools to derive relationships that can help us in achieving our objective. We will also try to recreate them on a 2D map using various Python Libraries in order to understand the variables on a broader geographic scale, widening our scope of analysis.

Describing the Variables:

The rationale behind picking the independent variables we have chosen is associated with the research question we wish to answer at the end of our analysis. Since we would like to establish the dynamics of consumer budget with respect to the price of a house, we need to study how prices vary given the age of the house, rooms per person, ocean_proximity and average income of the population living in a particular region

Each of the variables stated above perform the following roles:

- **Ocean Proximity** describes how far the house is from the nearest water body and can take upto 5 values. This variable gains importance due to the fact that the general notion associated with reality market is that houses near to the ocean are relatively more expensive than the ones further away. We want to understand this and verify what kind of a relationship does the proximity to the ocean have with the price of a house. A person having budget constraints may not want to consider houses near the ocean in case of a positive correlation.
- **Age of the House** refers to the median age of houses within that location that are bought by consumers. Age of the house can be a deciding factor for a house buyer in case his/her preferences for younger houses play a major role in his final decision. We want to understand if older houses demand a higher price compared to the newer houses or is it vice versa. Alternatively, we also would like to know if they are not related at all. If there is a budget constraint then the knowledge that switching to houses based on age can help find a more suitable property, it would help the user a lot. .
- **Average Occupancy(Rooms Per Person)** is a derived variable which is calculated by dividing the Total Rooms with the Households and displays the average occupancy. It indicates how many houses are available in a particular location given its longitude and latitude. It is a crucial variable because in the absence of the Total Houses data, there is no way to calculate the actual average rooms per person since we don't have a clue as to the number of rooms shown in the dataset belong to how many houses. A higher value suggests more number of houses are available in that location.
- **Average Income** of a region was chosen as a variable in order to ascertain whether systemmic differences in average household income between different counties made any difference to house prices. An intuitive guess would be that a richer neighbourhood will boast of higher property prices and vice versa. Hence it becomes a very important factor to decide which counties fit into the user budget and which do not provided there is a positive correlation between them.

Literature Review:

We went through a series of research papers that delved in similar topics of either predicting the house prices or finding the relationship of dependent factors with the price of a house. We could not find a specific literature that has touched upon the relationship of the factors provided in our given dataset and evaluated via a linear regression, a model analysing relationship between these factors and the price of a house. In this context, hence my analysis takes importance as it clearly demonstrates the importance of the factors (independent variables) such as the age of the house, proximity to the ocean front etc. and helps a prospective to consider these factors as part of his/her budget calculations

In *Truong, Q. D., Nguyen, M. T., Dang, H., & Mei, B. (2020). Housing Price Prediction via Improved Machine Learning Techniques*, The authors reiterate our hypothesis that the very commonly used factor House Price Index (HPI) to estimate the changes in housing price is incorrect since housing price is strongly correlated to other factors such as location, area, population, it requires information apart from HPI to predict individual housing price. But their work is based on data for Beijing and not California and hence our research becomes pertinent in this context

In *Gallatin, N., Hill, D. (2022, October 9). California Housing Markets: A Tale Twice Told*, the author aim to examine the current health of the California residential real estate market through an analysis that is threefold. In the first portion of our analysis, they try to observe the relationship between home price and real estate specific measures that are widely considered predictive of home price appreciation. These five variables are: The Traditional Housing Affordability Index (HAI), Unsold Inventory Index (UII), Median Time on Market (TOM), Building Permits, and Ten-Year Treasury Note yield. They employ the traditional HAI from the California Association of Realtors (CAR), which is calculated as the proportion of people that can afford a median priced home. This paper does not consider the 5 variables that my work has build upon hence the importance of the factors I have considered are ignored in this research paper

In *Zixu Wu. Prediction of California House Price Based on Multiple Linear Regression. Academic Journal of Engineering and Technology Science (2020) Vol. 3*, the author although affirms that the multivariate linear regression deployed in my research and analysis of the California is the correct approach to attach the given research question, the data the paper uses seems to be of Boston not withstanding the title of the paper that says California. Also it uses only a few of the factors considered by me namely the number of rooms. Hence the value of my research and findings is different and probably more valuable than that presented by the author

Summary Findings:

Our main goal is to analyse and deduce notable facts and figures which were not so evident to the public eye by viewing the numerous rows of data presented in the California Housing Market dataset in order to understand their effects on consumers purchasing behaviour. Below are some of the most noteworthy that will help a buyer to use our analysis in order to narrow down on a prospective house.

- When we considered the effect of **Ocean Proximity** on the pricing of the house, the frequency of houses found under the categories of 'INLAND' and 'NEAR THE OCEAN' far exceeded those of the other proximities provided, thus indicating a higher demand for houses along the coast among consumers.
- Upon a careful analysis of the **Income Distribution** across counties when considering consumers individually as well as clubbing them into income groups led us to believe that income is positively skewed against the backdrop of Price. Similiarly, we notice a negatively skewed graph when comparing the frequency of consumers against each income group.
- **Age of the House** may have a considerably weaker impact on the consumer's decision to purchase a home but it does have a significant relationship with ocean proximity. A majority of the consumers prefer to live in older houses either along the coastline or inland.
- The **Average Occupancy** values provided across different coordinates suggests that there is a mix of a surplus and shortage of houses depending upon the location due to an unstable and inconsistent demand and supply of houses.

Scraping New Data:

Additionaly, we have also looked at some of the factors that can influence a buyer's buying mindset but are not available in the given dataset. Many of such factors are available on different web portals and need to examined first to analyse if they are meaningful in our context and then figure out how can we get the relevant data off of the web portal. One of the ways to do this is web scraping. But in order to do so, the data should be in a format that can easily be scraped off. Its possible that the data is present, is relevant but its impossible to scrape it off and use in adjacency to the given dataset. In our analysis below, we have successfully scraped off data from a Wikipedia webpage and aligned it to our existing dataset which has produced significant results such as being able to map out how crime rates are fundamentally different across different counties while also being able to set up a relationship between the Income of a given neighborhood and its corresponding crime level index. Similiarly, we have been able to analyse how the demand of the number of houses vary with the different levels of crime that take place in that county on average.

Data Cleaning

Step 1: The process of Data Cleaning was a 4 step procedure beginning with loading the Original Dataset onto the Jupyter Notebook as shown below. Upon reading the DataFrame, it consisted of 10 columns and about 20640 observations in total.

In [2]:
```
import pandas as pd
import numpy as np
from pathlib import Path
from matplotlib import pyplot as plt
import math
import seaborn as sns
import geopandas as gpd
import geopy
from geopy.geocoders import Nominatim
from shapely.geometry import Point
import statistics
import regex
import requests
import urllib.request
import time
from bs4 import BeautifulSoup
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col
from linearmodels.iv import IV2SLS
from sklearn import tree
from sklearn import (
    linear_model, metrics, pipeline, model_selection)

df1 = pd.read_csv('housing_CSV2.csv') # Loading original Dataset
df1
```

Out[2]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.0	1.5603	78100.0	INLAND
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.0	2.5568	77100.0	INLAND
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.0	1.7000	92300.0	INLAND
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.0	1.8672	84700.0	INLAND
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.0	2.3886	89400.0	INLAND

20640 rows × 10 columns

Step 2: The second step involved transforming all string values into their numeric forms for one of the Independent Variables known as 'Ocean Proximity'. We assigned random integers between 1 and 5 to each of the proximities stated in the DataFrame such as Inland, Island, Near Ocean and <1H Ocean.

In [3]:
```
# Creating dummy variables for the 'ocean_proximity' column and updating the dtype from str to int
def changing_ocean_proximity_variables(df):
    dict1 = {'INLAND': 1, 'ISLAND': 2, 'NEAR BAY': 3, 'NEAR OCEAN': 4, '<1H OCEAN': 5}
    for row in df.iterrows():
```

```
        index, column_values = row
        new_ocean_status = dict1[column_values['ocean_proximity']]
        df.at[index, 'ocean_proximity'] = new_ocean_status
    df[['ocean_proximity']] = df[['ocean_proximity']].apply(pd.to_numeric)
    return df
df2 = changing_ocean_proximity_variables(df1)
df2
```

Out[3]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	3
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	3
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	3
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	3
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	3
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.0	1.5603	78100.0	1
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.0	2.5568	77100.0	1
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.0	1.7000	92300.0	1
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.0	1.8672	84700.0	1
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.0	2.3886	89400.0	1

20640 rows × 10 columns

Step 3: The next step involved finding all missing values present within the raw Dataset and replacing the empty cells with 'NaN' instead. Since the resulting column with all missing values was 'Total Bedrooms', we leave the cells empty and do not fill in any values as this column fails to enable us in supporting our analysis to answer the main objective stated in the Overview.

In [4]:
```
# Finding any missing values in the Dataset provided to us
def finding_null_values(df):
    return df.isnull().sum()

print(finding_null_values(df2))
```

```
longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
```

Step 4: The final step of Data Cleaning involved creating additional an variable which is derived from two columns Total rooms and households as stated in the description. The new Independent Variable is called 'average_rooms_per_person'.

In [5]:
```
# Creating a variable Rooms Per Person
def creating_new_variables(df):
    for row in df.iterrows():
        index, column_values = row
        total_rooms = column_values['total_rooms']
        households = column_values['households']
        rooms_person = round(total_rooms / households)
```

```
        df.at[index, 'avg_rooms_per_person'] = rooms_person
    return df

df3 = creating_new_variables(df2)
df3
```

Out[5]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity	avg_rooms_per_person
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	3	7.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	3	6.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	3	8.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	3	6.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	3	6.0
...	...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.0	1.5603	78100.0	1	5.0
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.0	2.5568	77100.0	1	6.0
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.0	1.7000	92300.0	1	5.0
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.0	1.8672	84700.0	1	5.0
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.0	2.3886	89400.0	1	5.0

20640 rows × 11 columns

Summary Statistics

Here is a description of the different statistical values that the California Housing dataset provides us with:

- **count**: It tells is the total number of rows that the dataset has. Hence we have 20640 observations in the dataset. This does not mean there are 20640 unique observations since I noticed some discrepancies in the dataset. It was found that for a given pair of longitude and latitude, there were multiple set of observations resulting in different values of the columns and hence the variables. I was unable to find a way to remove the redundancy.

- **mean**: The average value of the column. The average house age is between 28-29 years which tells us the houses are quite old. The average rooms per person is almost 2. It indicates big spacious houses. The average price for any given house is almost 200,000 USD which indicates that housing is expensive in California on an average.

- **std**: This value tells us about the standard deviation of each of the variables. In a normal distribution, we know that 68% of the values have a standard deviation of 1, 95% have a standard deviation of 2 and 99% have a standard deviation of 3. Using this information, we can see that 68% of the houses whose data is available, have an age of 28.6 years +/- 12.6 years. So 68% of the houses have a fairly wide age ranging from 16 years to 40 years. This improves our understanding of the houses in California from an age perspective. Houses as new as just as 15-16 years old to almost 40 years gives a choice to the housing market buyer in case his preferences of a newer vs a older house are dominant in his bid to purchase one. Similarly for the average house price of 200,000 USD, 68% of the houses actually fall between the price of 200,000 - 115,000 to 200,000 + 115,000 giving a fairly wide range of buying choices ranging from almost 75,000 to 315,000. This wide band of pricing should spur the home buyer to consider buying a house more seriously.

- **min/max**: The min and max represent the smallest and the largest values for a particular column/variable. The housing age variable varies from 1 to 52 years showing that California provides a prospective home buyer a wide variety of options to choose from ranging from almost brand new houses to some that are almost a half a century old. This allows buyers to prioritize their buying preferences. The min/max values for the average rooms per person are ranging from 1 to 142. As described earlier, the average number of rooms per person in any typical family house should not be more than 3-4 rooms on the higher side and 1-2 rooms on the lower. But as we can see the value goes as high as 142. This does not mean that there some houses are so large that a person has 142 rooms on average to himself/herself but when this value is read with an eye on the number of households in that area, we can see that given the total number of rooms in that area, the area is inhabited by a very few households thus indicating that there should be much more availability of houses in that area.

- **25%/50%/75%**: These are percentile markers. 50% is nothing but the median, 25% being the lower percentile and 75% being the higher percentile. The lower percentile for housing age indicates that 25% or less of the total houses available are less than 18 years old but 75% of the houses are 37 years old or below with half the total houses available being 29 years older or less. For avarage rooms per person, the lower and higher percentile are close which when rounded off lie between 1 - 2 rooms per person. This I believe is fine and is line with a house requirements. The average house price though presents interesting numbers. Half the total number of the houses are available with 179,000 USD or below with only 25% houses being more expensive than 264,000 USD. Infact, 25% or less of the total houses lie below 119,000 USD. Affordable housing is certainly available in California, or at least this is what the dataset tells us.

In [5]:
```
# Creating the Summary Statistics Table using Three Independent and One Dependent Variables

sub_df3 = df3.loc[:, ['housing_median_age', 'avg_rooms_per_person', 'median_house_value', 'ocean_proximity']]
sub_df3
summary_stats1 = sub_df3.describe()
summary_stats1
```

Out[5]:

	housing_median_age	avg_rooms_per_person	median_house_value	ocean_proximity
count	20640.000000	20640.000000	20640.000000	20640.000000
mean	28.639486	5.424806	206855.816909	3.379021
std	12.585558	2.491940	115395.615874	1.739433
min	1.000000	1.000000	14999.000000	1.000000
25%	18.000000	4.000000	119600.000000	1.000000
50%	29.000000	5.000000	179700.000000	4.000000
75%	37.000000	6.000000	264725.000000	5.000000
max	52.000000	142.000000	500001.000000	5.000000

Graphical Analysis

Discrete Variable - Ocean Proximity

Using Ocean Proximity as our first Independent Variable we can create two graphs, a Bar Chart as well as a Boxplot as shown below. Since Ocean Proximity is a Discrete Variable and takes only 5 distinct values, both these graphical tools are fitting to conduct the analysis. The Bar Chart, depicted in the first figure, represents the frequency of houses found within different proximities to a Water Body such as an Ocean or Sea.

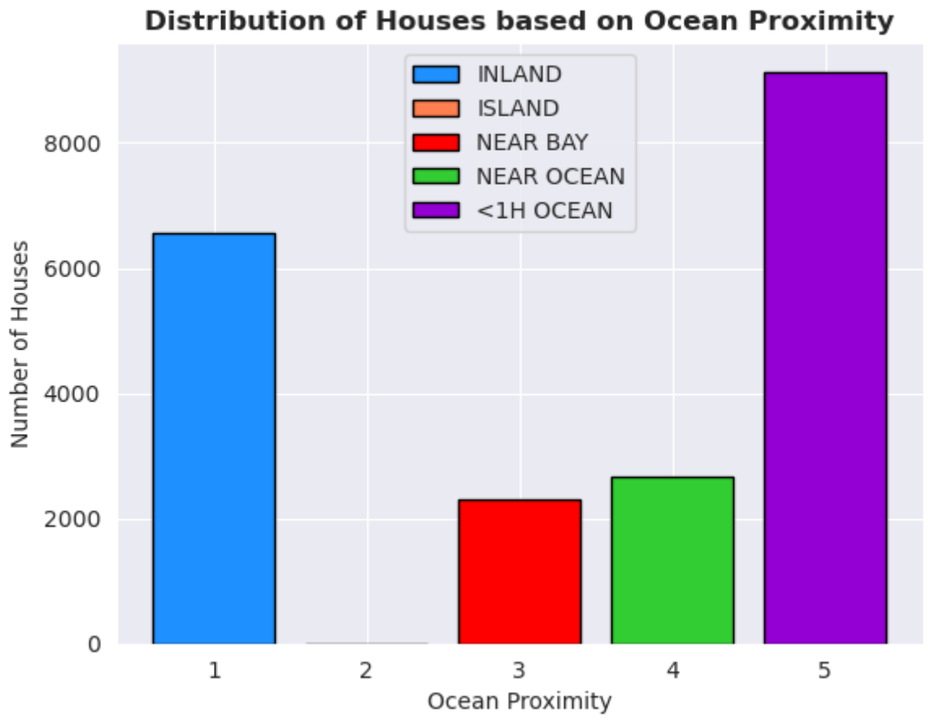
It is quite evident from the chart that most number of houses are found quite close to the Ocean, less than an hour's journey. This is not suprising since California lies along the Pacific water bed and most consumers would prefer to enjoy the view of the coastal region. Another observation indicates that consumers also prefer buying houses Inland or in urban and metropolitan settings as an alternative to buying them near the Ocean. We notice that the graph shows no bar for the category of Island. Now, since the scale of frequency was higher for other categories and quite low for the latter, the Bar does not show up. We can however make out the small category from the Boxplot.

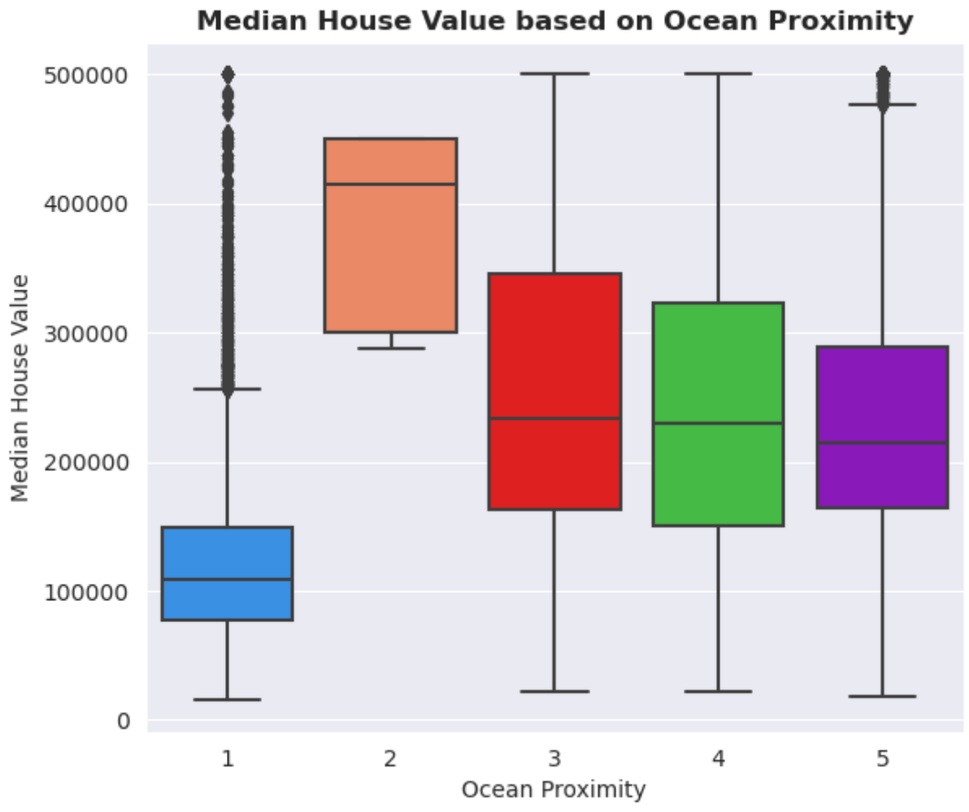
The Boxplot represents the relationship between the proximity of the house and the median house value. Each box is subdivided into two smaller boxes by a line in the center. That is the median and depicts the 50 percentile of the data. In other words, it shows that half of the houses found Inland are priced less than $100,000 whereas half of them are priced above $100,000. Similiarly, we can see for other proximities as well. It is easy to notice that houses found away from Oceans are priced relatively cheaper as compared to those which are found near a water body by comparing their Median Prices.

Another key fact is that data for the categories of 'Inland' and 'Island' are positively and negatively skewed, since the length of the whiskers fail to match on either side. This can be verified since for the category of Inland the area of the box above the median is slightly greater than that of the box beneath it. Similiarly, for the category of Island, we notice it is negaatively skewed since the area of the box below the median is greater than that of the one above it.

In [53]:
```
# Creating a Bar Chart with Ocean Proximity as Independent Variable

dict1 = {'INLAND': 1, 'ISLAND': 2, 'NEAR BAY': 3, 'NEAR OCEAN': 4, '<1H OCEAN': 5}
freq_dict = {}
```

```
location_df = df3.loc[:, ['ocean_proximity', 'median_house_value']]
gb_location = df3.groupby('ocean_proximity')
set1 = set(df3['ocean_proximity'])
for x in set1:
    freq = gb_location.get_group(x).count()['ocean_proximity']
    freq_dict[x] = freq

# sorting dictionary
Keys = list(freq_dict.keys())
Keys.sort()
sorted_freq_dict = {i: freq_dict[i] for i in Keys}
sorted_freq_dict
sorted_freq_dict

plt.bar(list(sorted_freq_dict.keys())[0], list(sorted_freq_dict.values())[0], label='INLAND', ec='black', color='dodgerblue')
plt.bar(list(sorted_freq_dict.keys())[1], list(sorted_freq_dict.values())[1], label='ISLAND', ec='black', color='coral')
plt.bar(list(sorted_freq_dict.keys())[2], list(sorted_freq_dict.values())[2], label='NEAR BAY', ec='black', color='red')
plt.bar(list(sorted_freq_dict.keys())[3], list(sorted_freq_dict.values())[3], label='NEAR OCEAN', ec='black', color='limegreen')
plt.bar(list(sorted_freq_dict.keys())[4], list(sorted_freq_dict.values())[4], label='<1H OCEAN', ec='black', color='darkviolet')

colors = ['dodgerblue', 'coral', 'red', 'limegreen', 'darkviolet']
plt.legend(loc='upper center')
plt.xlabel('Ocean Proximity')
plt.ylabel('Number of Houses')
plt.title('Distribution of Houses based on Ocean Proximity', fontweight='bold')
sns.set_palette(colors)
plt.show()
fig, ax = plt.subplots(figsize=(6.5,5.5))
sns.boxplot(x=location_df['ocean_proximity'], y=location_df['median_house_value'], ax=ax).set_title(
    'Median House Value based on Ocean Proximity', weight='bold')
ax.set_xlabel('Ocean Proximity')
ax.set_ylabel('Median House Value')

plt.show()
```

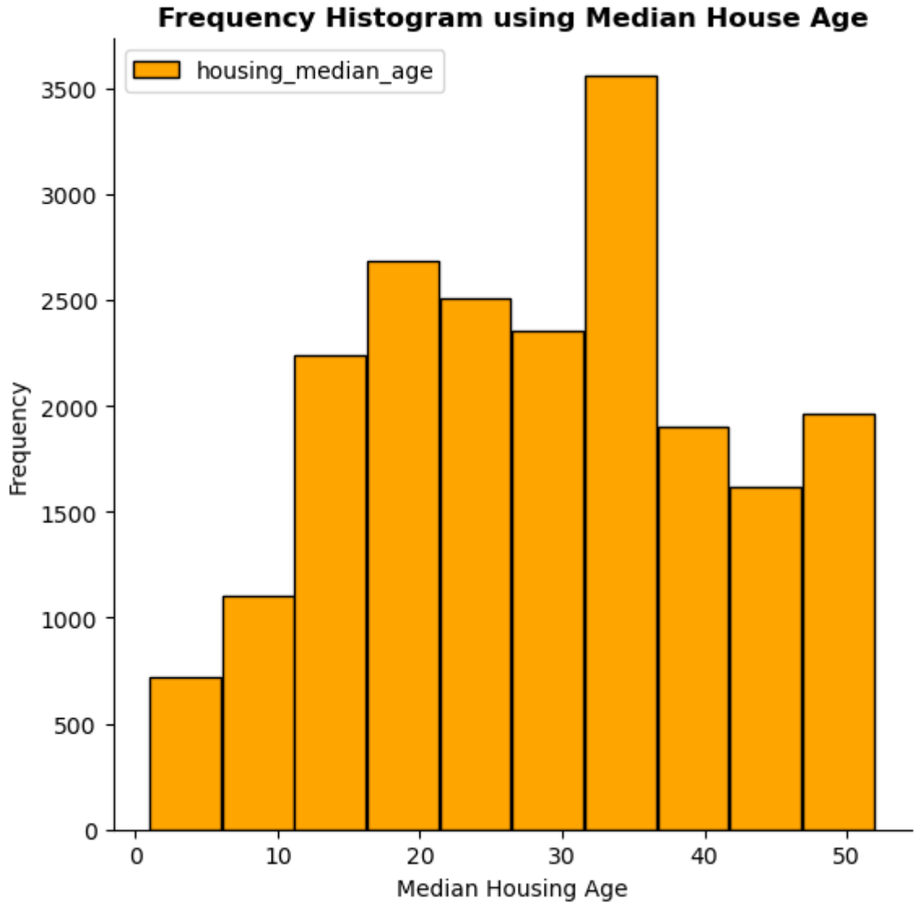


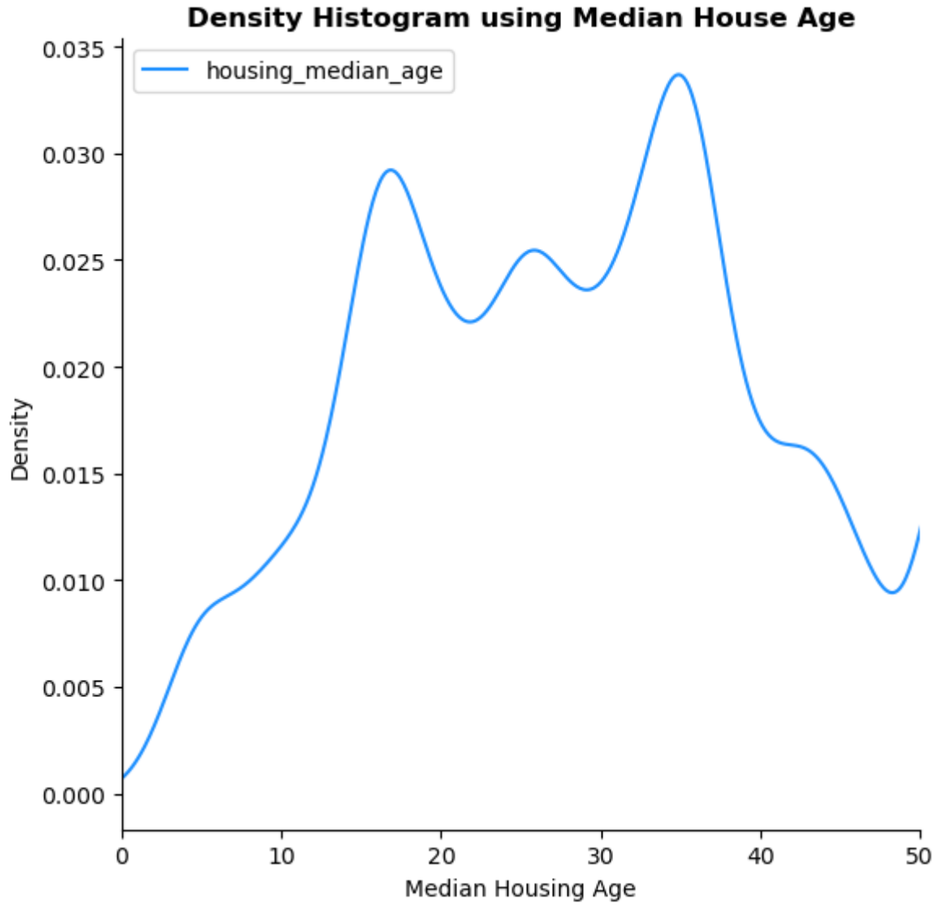


Continous Variables - Median Housing Age

Our second graphical representation includes a histogram for the Independent Variable of Housing Median Age. The graph for the same depicts the distribution of houses with different ages across different locations in the state of California. It presents itself as a non-symmetric unimodal representation with 10 equal sized bins with a width of 5 units. Since it is a unimodal bin, we can say that the mode or the most frequently appearing value is in the bin of 30 - 40 years. The density histogram below aids our analysis by describing the shape of the frequency distribution. It clearly indicates that the Housing Age is not normally distributed since we fail to notice a bell shaped curve. Moreover, a density histogram is useful in telling the probability of a certain value occurring in the distribution. So, for the bins of 10-20 years and 30-40 years, we see the highest probabilities of 0.028 and 0.034 respectively.

In [7]:
```
# Creating a Frequency Histogram with House Age as Independent Variable

fig, ax = plt.subplots(1, 2, figsize=(15, 8))
house_age_df = df3.loc[:, ['housing_median_age']]
house_age_df.plot(ax=ax[0], kind='hist', color='orange', ec='black', bins=10, width=5)
ax[0].set_title('Frequency Histogram using Median House Age', fontweight='bold')
house_age_df.plot(ax=ax[1], kind='kde')
ax[0].set_xlabel('Median Housing Age')
ax[1].set_xlabel('Median Housing Age')

plt.ylabel('Density')
plt.xlim(0,50)
ax[1].set_title('Density Histogram using Median House Age', fontweight='bold')
plt.tight_layout(pad=9)
ax[0].spines['top'].set_visible(False)
ax[0].spines['right'].set_visible(False)
ax[1].spines['top'].set_visible(False)
ax[1].spines['right'].set_visible(False)

plt.show()
```

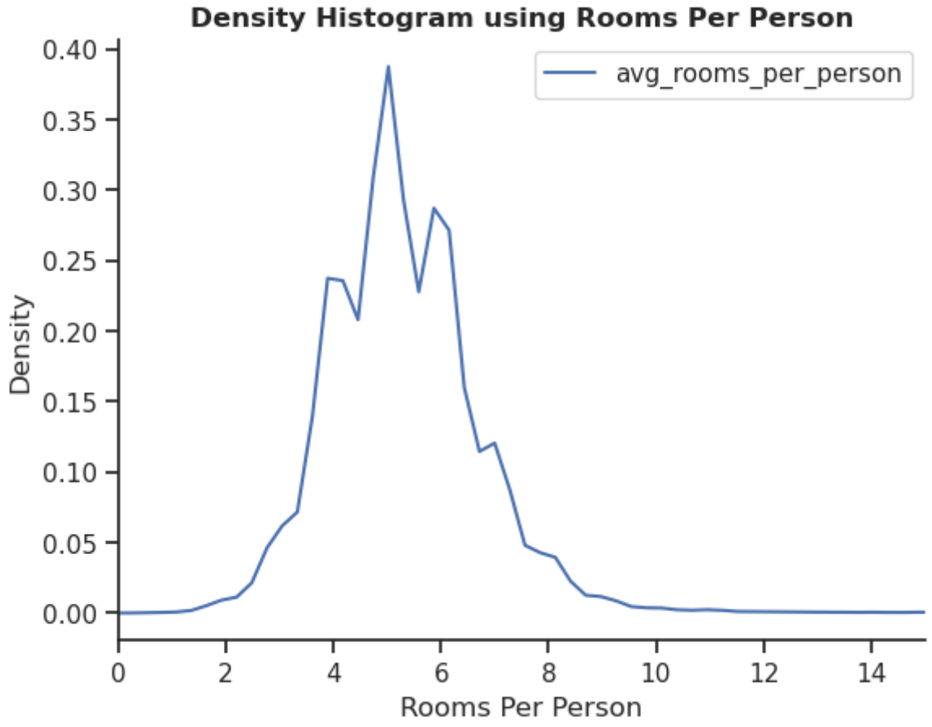




Average Rooms Per Person

The figure below consists of the Density distribution histogram of the Variable 'avg_rooms_per_person'. The Histogram can be described as a non-symmetric and unimodal distribution. As we stated above, this variable is important since it enables the consumers in knowing the number of houses available within that location. Using the plot below, we can notice that nearly 40% of all observations consist of values between 4 - 6. This indicates that most locations provided within the Dataset in the state of California consist of houses already occupied by consumers. A value as low as 4 implies that demand of houses is higher than its supply in most parts. Since we know that in equilibrium prices will move toward the market clearing point and there exists a shortage, we would see prices of houses rise gradually.

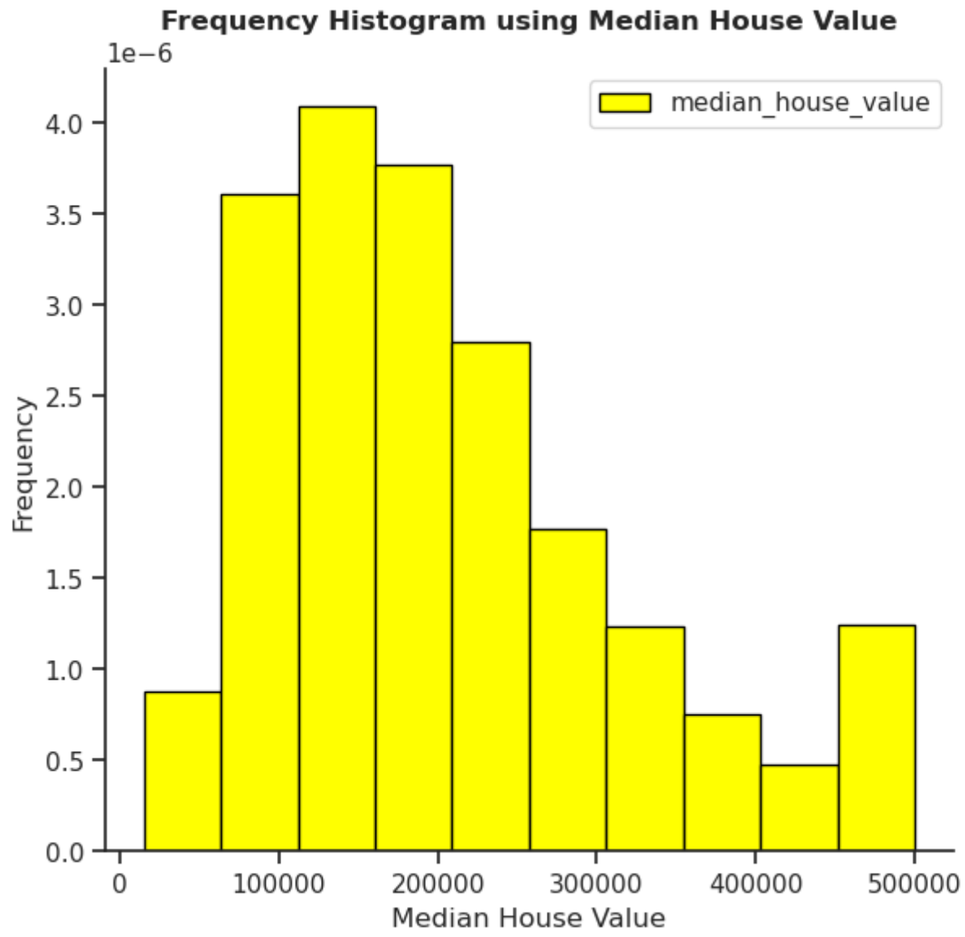
In [24]:
```
# Creating a Density Histogram with Avg Rooms Per Person as Independent Variable
fig, ax = plt.subplots()
rooms_person_df = df3.loc[:, ['avg_rooms_per_person']]
rooms_person_df.plot(kind='kde', ax=ax)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.xlabel('Rooms Per Person')
plt.ylabel('Density')
plt.title('Density Histogram using Rooms Per Person', fontweight='bold')
plt.xlim(0,15)
plt.show()
```

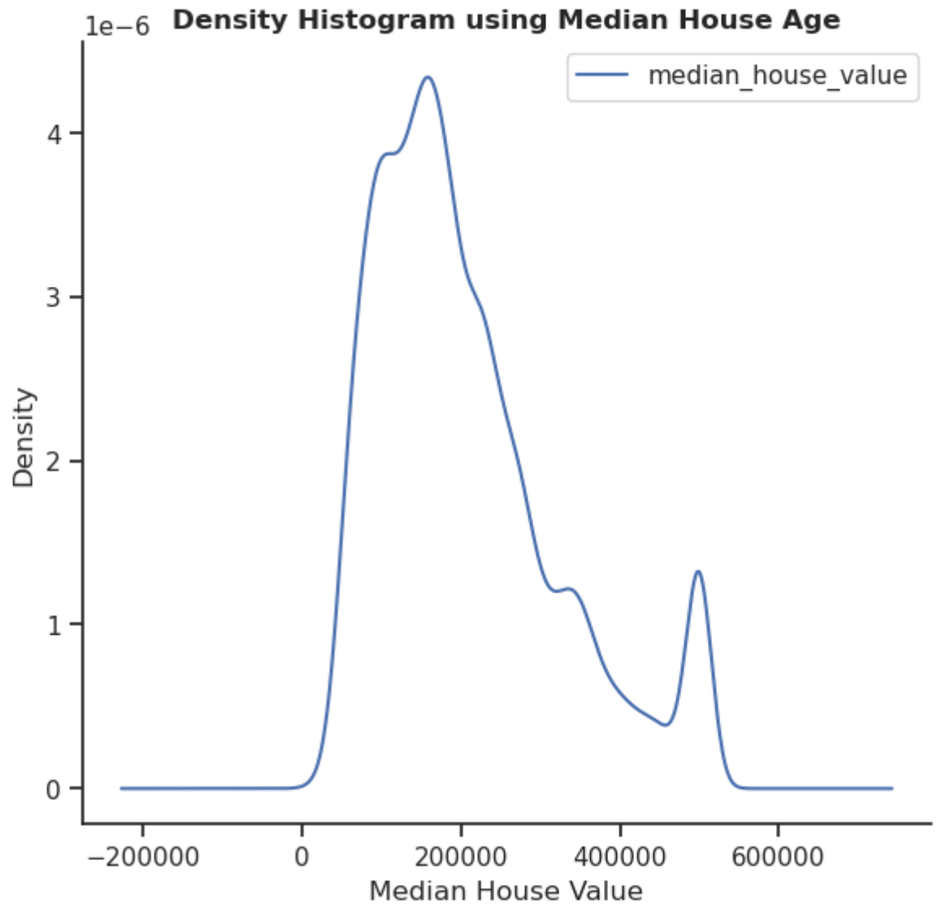


Median House Value

The figure below consists of the frequency distribution histogram of the Variable 'median_house_value' as well as the density histogram beneath it. The Histogram can be described as a positively skewed unimodal distribution with a long tail extending towards its right. This shape can also be verified by the Density histogram right below it, beginning at the value of 20000 and extending all the way through. This shape is quite intuitive since price and demand of a house always have an inverse relationship, as price gradually rises, the frequency of houses in that pirce range will drop thus creating a tail. We can also notice that the most frequently seen houses lie in the price range of $100,000 - 200,000, making up nearly 4\% of all observations.

In [20]:
```
# Creating a Frequency Histogram with House Value as Dependent Variable

fig, ax = plt.subplots(1, 2, figsize=(15, 8))
rooms_person_df = df3.loc[:, ['median_house_value']]
rooms_person_df.plot(ax=ax[0], kind='hist', color='yellow', ec='black', rwidth=3, density=True)
ax[0].set_title('Frequency Histogram using Median House Value', fontweight='bold')
rooms_person_df.plot(ax=ax[1], kind='kde')
ax[0].set_xlabel('Median House Value')
ax[1].set_xlabel('Median House Value')

plt.ylabel('Density')
ax[1].set_title('Density Histogram using Median House Age', fontweight='bold')
ax[0].spines['top'].set_visible(False)
ax[0].spines['right'].set_visible(False)
ax[1].spines['top'].set_visible(False)
ax[1].spines['right'].set_visible(False)
plt.tight_layout(pad=7)
plt.show()
```





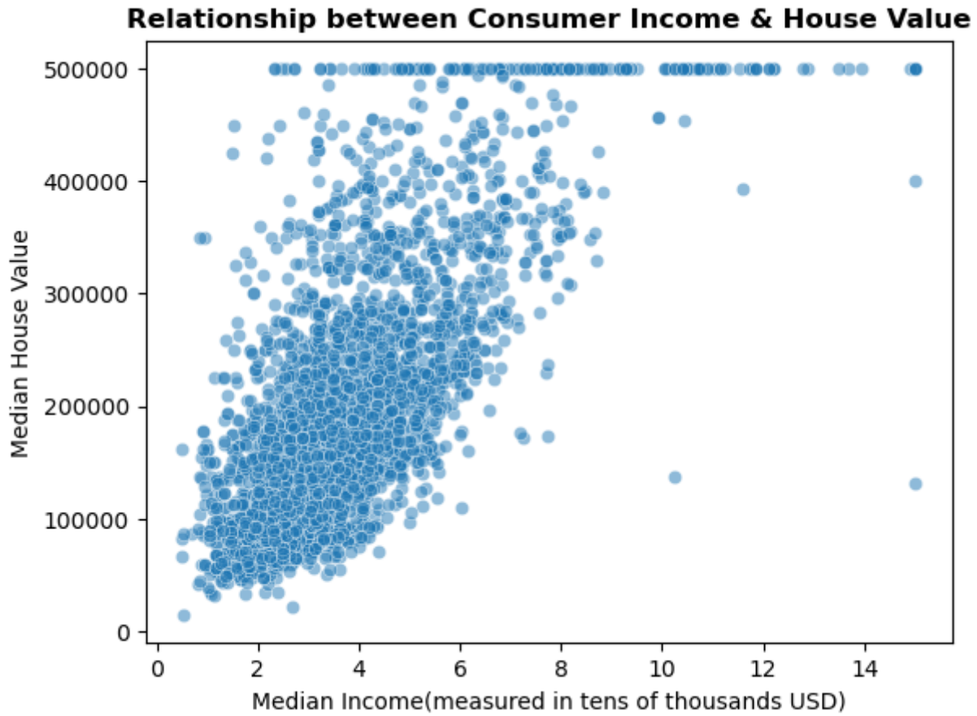
Scatterplots

Median Income and Median House Value

The figure below depicts a scatterplot showcasing the relationship between two variables, 'median_income' and 'median_house_value'. The plot takes a sample of about 3000 observations from the total Population of 20640 observations to prevent overplotting. Each data point in the figure represents a house in the state of California. The graph suggests that there seems to be in fact quite a strong relationship between the variables, since the scatter points are strongly clustered together in one group.

We can also notice that the scatter transitions upwards, rising from the bottom left to the upper right indicating a positive and linear relationship as well. Another way to confirm the above result is to draw a straight line through the scatter, if the resulting points lie around the line we can conclude a positive and linear relationship. So, from the plot below, the resulting conclusion that can be drawn is that there exists a direct correlation between median house value and median income of consumers, in other words as income increases the housing preference value increases subsequently.

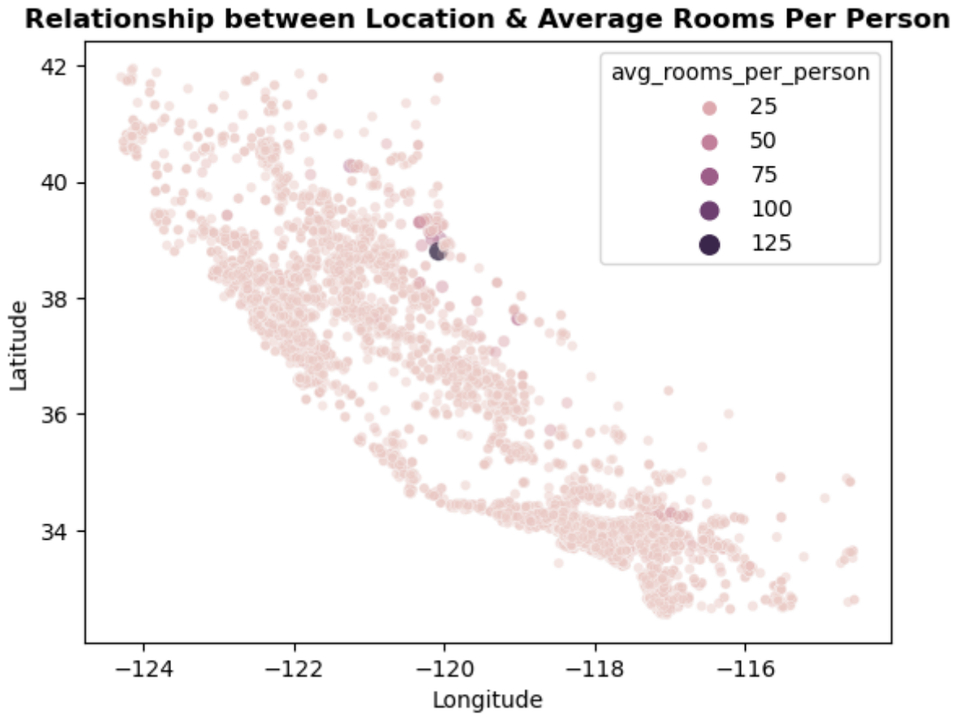
In [6]:
```
# Creating a Scatterplot with Variables Median Income and House Value

random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=3000)
sns.scatterplot(data=df3.iloc[sample], x="median_income", y="median_house_value", alpha=0.5)
plt.xlabel('Median Income(measured in tens of thousands USD)')
plt.ylabel('Median House Value')
plt.title('Relationship between Consumer Income & House Value', fontweight='bold')
plt.show()
```



Average Rooms Per Person and Location

The scatterplot depicts a strong linear and negative relationship between Location and Average Rooms Per Person. The intent of this scatter plot was to identify areas where the Average Rooms Per Person statistic was high. This would have highlighted the areas which had more available houses in it. But since there are not too many such areas with real high numbers, the scatter plot is not clearly able to show these. On the assumption that a normal family may require a maximum of 2-3 rooms per person (we are considering the whole house which does just include bedrooms only), all numbers higher than 3 are indicative of the fact that there are houses available in that region.

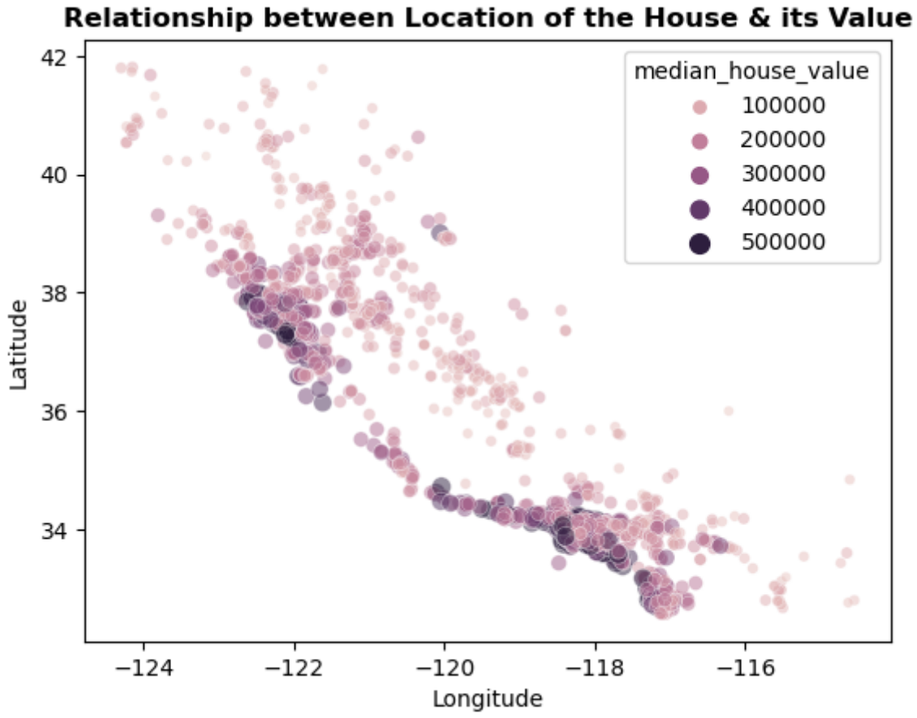
In [7]:
```
# Creating Scatterplot with Location and Average Rooms Per Person

random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=20000)
sns.scatterplot(data=df3.iloc[sample], x="longitude", y="latitude", size='avg_rooms_per_person',
        hue='avg_rooms_per_person', alpha=0.5)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Relationship between Location & Average Rooms Per Person', fontweight='bold')
plt.show()
```



Location and Median House Value

The scatterplot below shows how prices of houses change from locality to locality measured by the longitude and latitude of a specific block in the state of California. We are able to deduce a negative linear relationship from the graph below as the scatter extends from the top right of the figure all the way down to the bottom right. We also happen to notice that the data points are clustered into groups of two indicating a strong link within those coordinates. We can see that using the coordinates if we map the place, we get places within California where the houses are possibly located. Places such as San Diego, Rockland, Los Angeles, San Francisco will usually have houses priced at the higher end while those found in small towns like El Monte, Imperial County would be cheaper.

In [8]:
```
# Creating Scatterplot with Latitude and Longitude on each axis

random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=3000)
sns.scatterplot(data=df3.iloc[sample], x="longitude", y="latitude", size='median_house_value',
                hue='median_house_value', alpha=0.5)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Relationship between Location of the House & its Value', fontweight='bold')
plt.show()
```



Project 2

Message

The given problem on hand is based on the California Housing market data. The dataset contains various data points collected for the entire length of the California state categorised on the basis of latitudes and longitudes. These data points touch different aspects of what a prospective buyer would have in mind while deciding on a particular property. These range from the age of the house to its distance from the ocean, an aspect that is specific to the state of California having a large coast line. Our aim of the project is to help the buyer to get a perspective into what kind of a house can he get given one, his/her budget and two, his/her preferences.

For this very purpose, we make use of Bar Charts and Scatterplots to visualize these variables and aid us in understanding them. Quite similiar to an automated self generative software program, our analysis ideally should take these inputs from a house buyer and come up with a list of options best suited for him/her. For example, let's assume a consumer starts off with a requirement of a 3 bedroom house with a desired ocean proximity. Based on our analysis, he/she is able to find those counties which are budget friendly and buying a house is a distinct possibility. Moreover, we wish our analysis to be dynamic to the effect that if they decide to change their preferences, we are able to accomodate for the same.

Using this message and visualization, we can attempt to answer the question by subplotting our dependent variable House Value against different independent variables, for this analysis we will pick two of our most important ones: Income, Housing Age and Population Density.

Subplotting Income against House Value

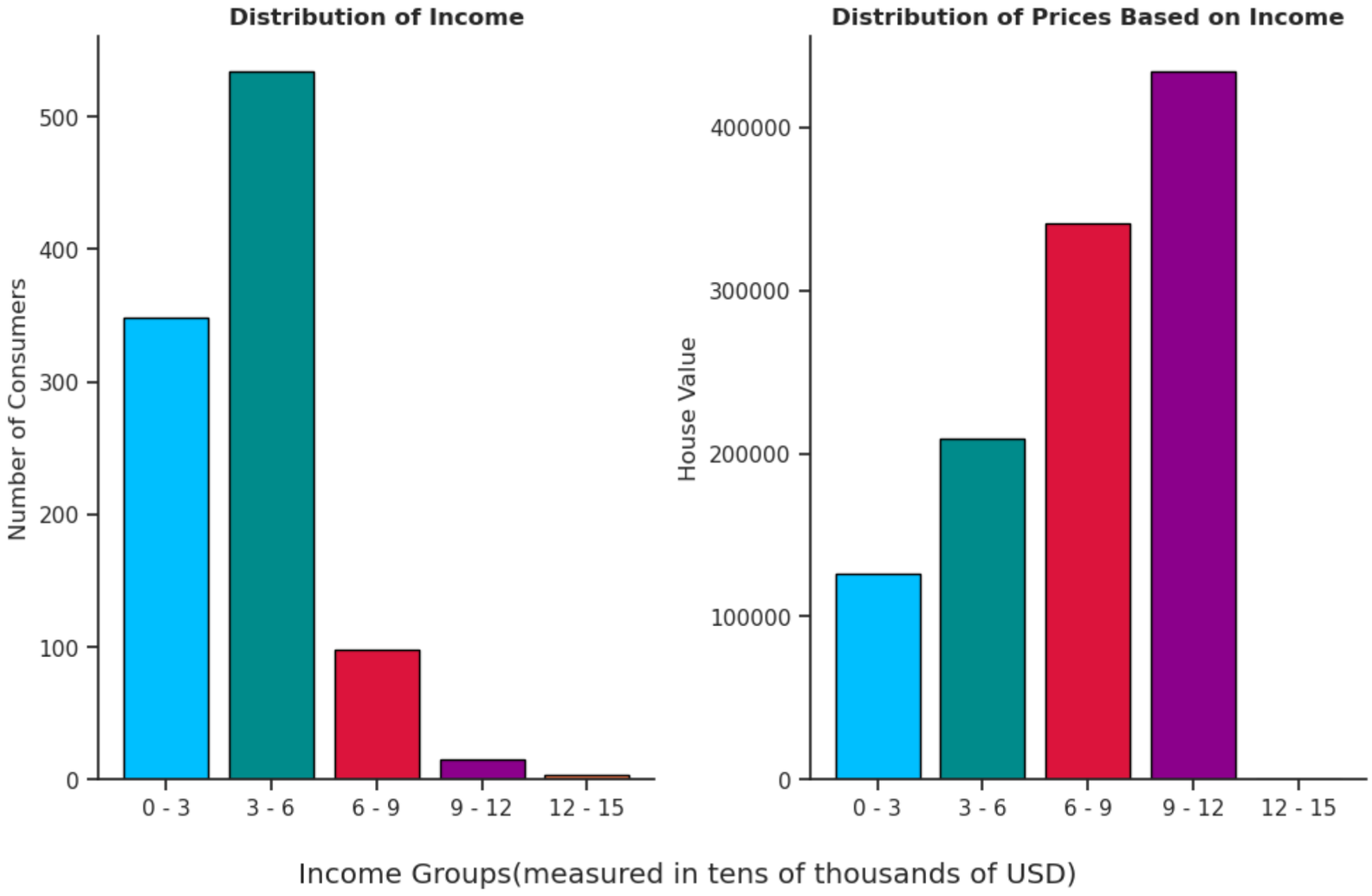
When we plotted the scatterplot depicting the positive correlation between Price and Income, we limited our results by providing a more wide frame of view since we were taking up income in general. We can perhaps create income groups and examine how consumers in different groups react differently to prices.

Our first graph on the left shows the income groups that can be created by splitting the population of California into each of these based on the data provided in the original Dataframe. We can clearly see a right tail in this graph which is quite intuitive since there would be a samller proportion of the consumers in the high income category as compared to the lower ones. We also notice that the income bloc of '3 - 6' has the most number of consumers, so we can say that this analysis is dealing with a large middle class population.

Our graph on the right subplots the Average Value of the House against these different income groups, i.e., it indicates the range of prices for houses that consumers in the particular bloc are comfortable with. Just as before, we get a positive relationship between the two variables. As the we move into higher income categories, the amount of money households are willing to pay increases as well due to rising purchasing power.

In [12]:
```
# Plotting the Distribution of Median Income based on Income Groups
random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=1000)
income_df = df3.loc[:, ['median_income']]
sample_income_df = income_df.iloc[sample]
unique_incomes = list(pd.unique(sample_income_df['median_income']))
max_income = int(max(list(pd.unique(df3['median_income']))))
min_income = min(list(pd.unique(df3['median_income'])))
num_groups = 5
interval_group = max_income // num_groups
income_groups = {}
lower_limit = 0
for x in range(interval_group, max_income + 1, interval_group):
    income_groups[f"{lower_limit} - {x}"] = 0
    lower_limit = x

for y in unique_incomes:
    gb_income = sample_income_df.groupby('median_income')
    freq = gb_income.get_group(y).count()['median_income']
    if 0 <= y <= interval_group:
        income_groups[f"{0} - {interval_group}"] += freq
    elif interval_group <= y <= (interval_group * 2):
        income_groups[f"{interval_group} - {interval_group * 2}"] += freq
    elif (interval_group * 2) <= y <= (interval_group * 3):
        income_groups[f"{(interval_group * 2)} - {interval_group * 3}"] += freq
    elif (interval_group * 3) <= y <= (interval_group * 4):
        income_groups[f"{(interval_group * 3)} - {interval_group * 4}"] += freq
    elif (interval_group * 4) <= y <= (interval_group * 5):
        income_groups[f"{(interval_group * 4)} - {interval_group * 5}"] += freq

income_price_df = df3.loc[:, ['median_income', 'median_house_value']]
sample_df = income_price_df.iloc[sample]
unique_prices = list(pd.unique(sample_df['median_house_value']))
max_price = int(max(list(pd.unique(sample_df['median_house_value']))))
prices = {'0 - 3': [], '3 - 6': [], '6 - 9': [], '9 - 12': [], '12 - 15': []}

q1 = sample_df['median_house_value'].quantile(0.25)
q3 = sample_df['median_house_value'].quantile(0.75)
IQR = q3 - q1
upp_bound = q3 + IQR*1.5
low_bound = q1 - IQR*1.5

for row in sample_df.iterrows():
    index, column_values = row
    if low_bound <= column_values['median_house_value'] <= upp_bound:
        continue
    else:
        sample_df = sample_df.drop(index)

for row in sample_df.iterrows():
    index, column_values = row
    income = column_values['median_income']
    price = column_values['median_house_value']
    if 0 <= income <= 3:
        prices['0 - 3'].append(price)
    elif 3 <= income <= 6:
        prices['3 - 6'].append(price)
    elif 6 <= income <= 9:
```

```
        prices['6 - 9'].append(price)
    elif 9 <= income <= 12:
        prices['9 - 12'].append(price)
    elif 12 <= income <= 15:
        prices['12 - 15'].append(price)
for x in prices:
    if prices[x] != []:
        prices[x] = statistics.mean(prices[x])
    else:
        prices[x] = 0

fig, ax = plt.subplots(1, 2, figsize=(10, 6.5), sharex=True)
colors = ['deepskyblue', 'darkcyan', 'crimson', 'darkmagenta', 'coral']
ax[0].bar(list(income_groups.keys()), list(income_groups.values()), color=colors, ec='black')
fig.supxlabel('Income Groups(measured in tens of thousands of USD)')
ax[0].set_ylabel('Number of Consumers')
ax[0].spines['top'].set_visible(False)
ax[0].spines['right'].set_visible(False)

ax[0].set_title('Distribution of Income', fontweight='bold')
ax[1].set_title('Distribution of Prices Based on Income', fontweight='bold')

ax[1].bar(list(prices.keys()), list(prices.values()), color=colors, ec='black')
ax[1].set_ylabel('House Value')
ax[1].spines['top'].set_visible(False)
ax[1].spines['right'].set_visible(False)
fig.tight_layout()
plt.show()
```



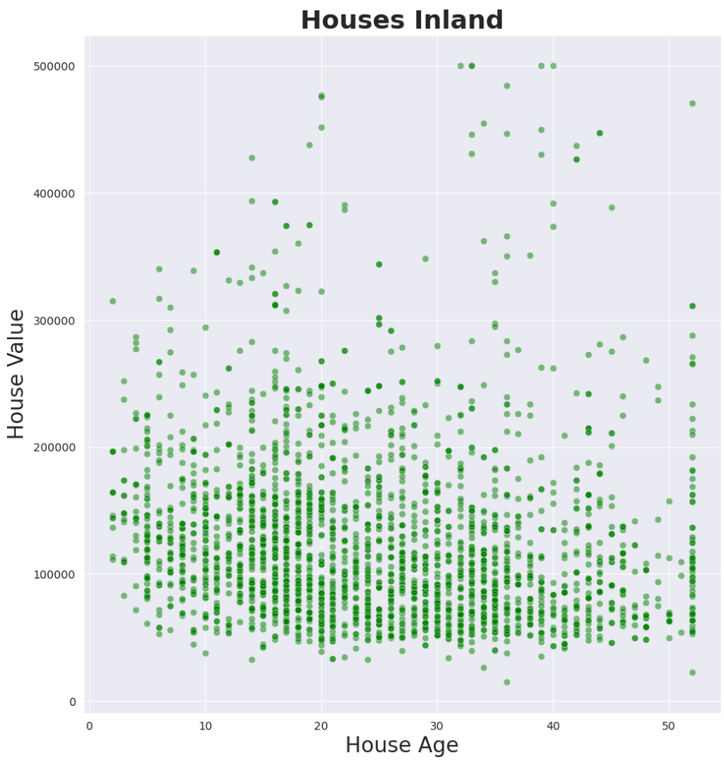
Subplotting Housing Age against House Value

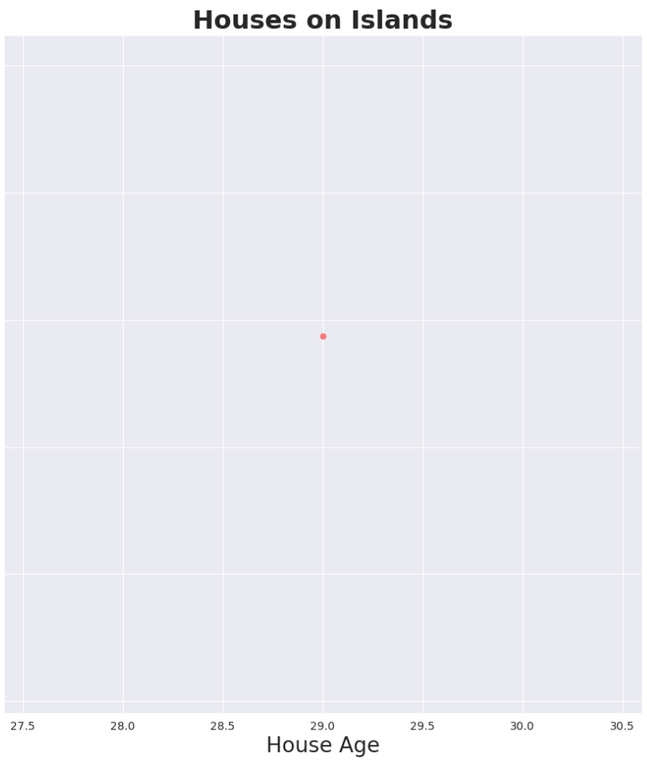
The 5 scatterplots below depict how the Age of Houses relate with the corresponding Median House Value based on different ocean proximities. It is quite evident that the graphs below do not immediately indicate a linear association between the two variables since we are unable to see the scatter moving in a linear pattern from bottom left to the top right corner. However, we are able to notice that the scatter is strongly clustered across all age values for houses found 'INLAND' and 'NEAR THE OCEAN'. We can notice that for the graph of 'INLAND', prices of houses vary between the range of 100,000 and 350,000 dollars for all ages, however the range for the same increases under the category of 'NEAR THE OCEAN' with it now being 200,000 to 500,000 dollars. One intuitive explanation for this can be that since houses near the coast offer a more splendid view of the Pacific, the value of the house is bound to increase across all age gaps.

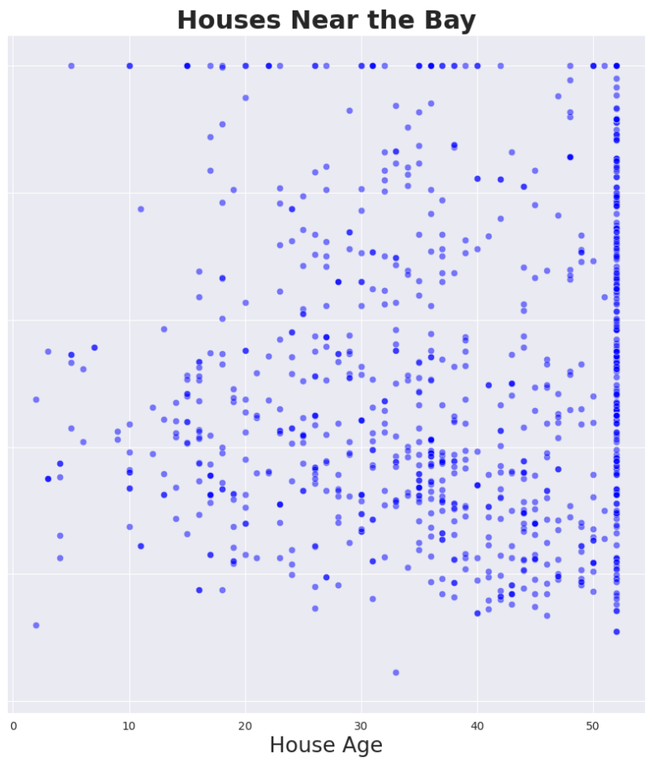
In [34]:
```
random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=9000)
age_price_df = df3.loc[:, ['median_house_value', 'housing_median_age', 'ocean_proximity']]
sample_df = age_price_df.iloc[sample]
gb_ocean_prox = sample_df.groupby('ocean_proximity')

inland_group = gb_ocean_prox.get_group(1)
inland_group = inland_group.drop(columns='ocean_proximity')
inland_group = inland_group.set_index('housing_median_age')

island_group = gb_ocean_prox.get_group(2)
```

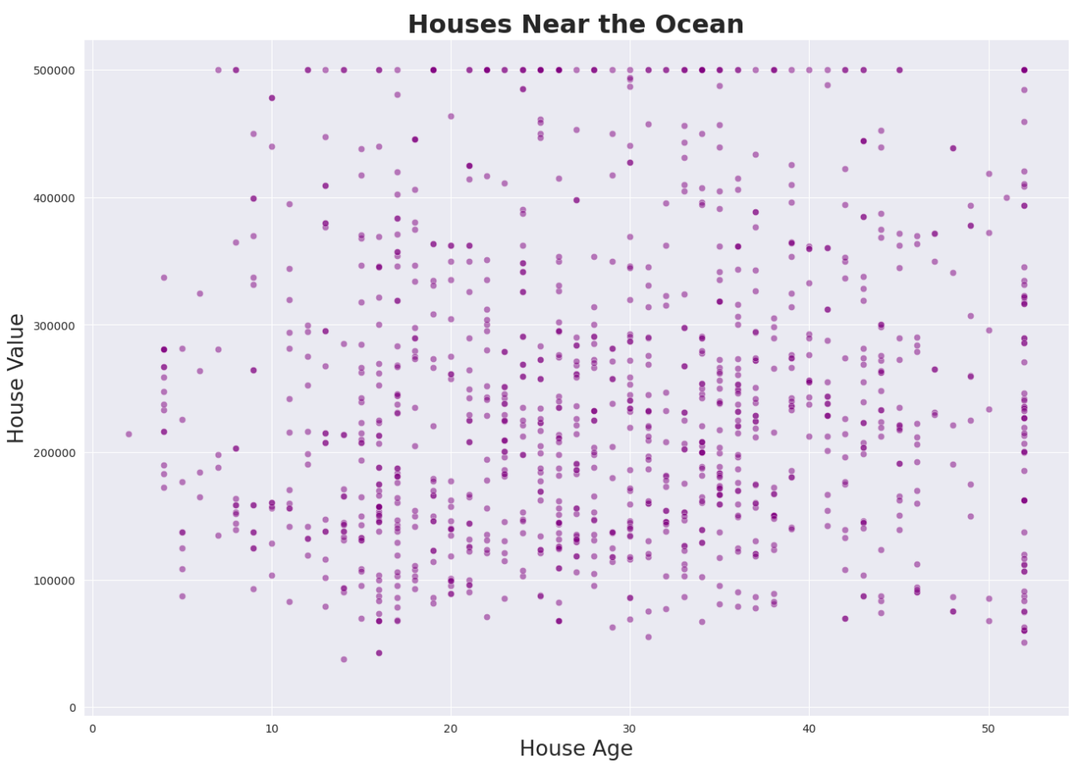
```
island_group = island_group.drop(columns='ocean_proximity')
island_group = island_group.set_index('housing_median_age')

near_bay_group = gb_ocean_prox.get_group(3)
near_bay_group = near_bay_group.drop(columns='ocean_proximity')
near_bay_group = near_bay_group.set_index('housing_median_age')

near_ocean_group = gb_ocean_prox.get_group(4)
near_ocean_group = near_ocean_group.drop(columns='ocean_proximity')
near_ocean_group = near_ocean_group.set_index('housing_median_age')

ocean_group = gb_ocean_prox.get_group(5)
ocean_group = ocean_group.drop(columns='ocean_proximity')
ocean_group = ocean_group.set_index('housing_median_age')

fig1, ax = plt.subplots(1, 3, figsize=(35, 11), sharey=True)
fig2, ax2 = plt.subplots(1, 2, figsize=(35, 11), sharey=True)
sns.scatterplot(data=inland_group, x="housing_median_age", y="median_house_value", alpha=0.5, color='green', ax=ax[0])
sns.scatterplot(data=island_group, x="housing_median_age", y="median_house_value", alpha=0.5, color='red', ax=ax[1])
sns.scatterplot(data=near_bay_group, x="housing_median_age", y="median_house_value", alpha=0.5, color='blue', ax=ax[2])
sns.scatterplot(data=near_ocean_group, x="housing_median_age", y="median_house_value", alpha=0.5, color='purple', ax=ax2[0])
sns.scatterplot(data=ocean_group, x="housing_median_age", y="median_house_value", alpha=0.5, color='maroon', ax=ax2[1])
ax[0].set_title('Houses Inland', fontsize=23, fontweight='bold')
ax[1].set_title('Houses on Islands', fontsize=23, fontweight='bold')
ax[2].set_title('Houses Near the Bay', fontsize=23, fontweight='bold')

ax[0].set_xlabel('House Age', fontsize=19)
ax[1].set_xlabel('House Age', fontsize=19)
ax[2].set_xlabel('House Age', fontsize=19)
ax[0].set_ylabel('House Value', fontsize=19)
ax[1].set_ylabel('House Value', fontsize=19)
ax[2].set_ylabel('House Value', fontsize=19)

ax2[0].set_xlabel('House Age', fontsize=19)
ax2[1].set_xlabel('House Age', fontsize=19)
ax2[0].set_ylabel('House Value', fontsize=19)
ax2[1].set_ylabel('House Value', fontsize=19)

ax2[0].set_title('Houses Near the Ocean', fontsize=23, fontweight='bold')
ax2[1].set_title('Houses on the Coast', fontsize=23, fontweight='bold')
sns.set_style("darkgrid")
fig1.suptitle('Comparing Housing Age with Prices based on Ocean Proximity', fontsize=30, fontweight='bold')
plt.show()
```

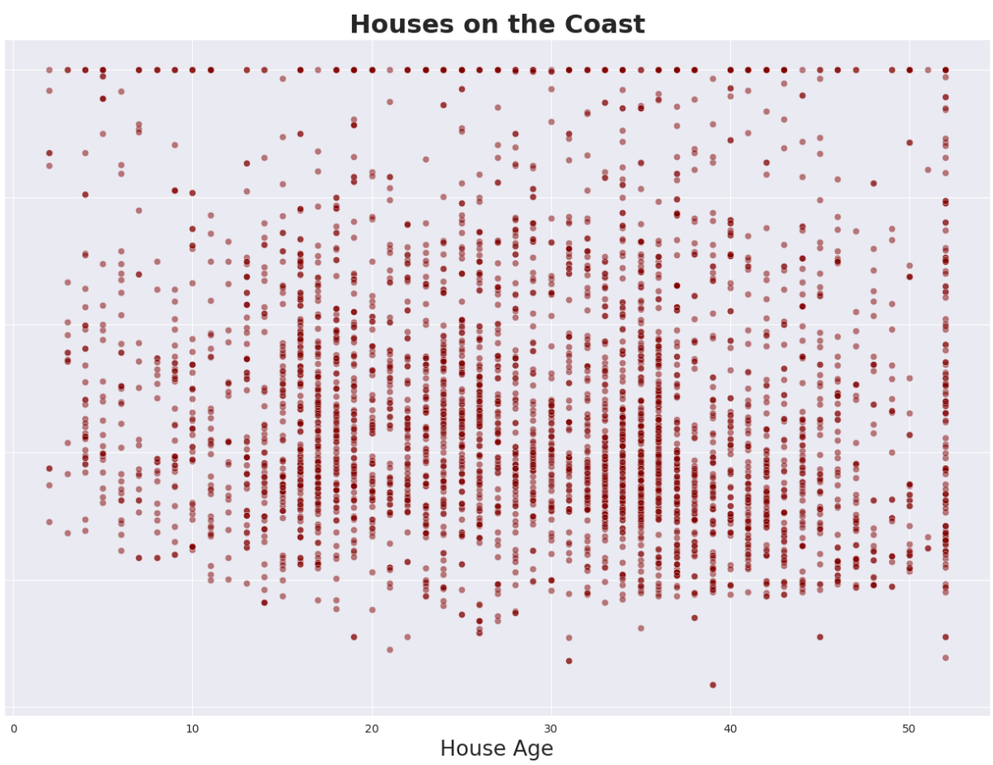
Comparing Housing Age with Prices based on Ocean Proximity











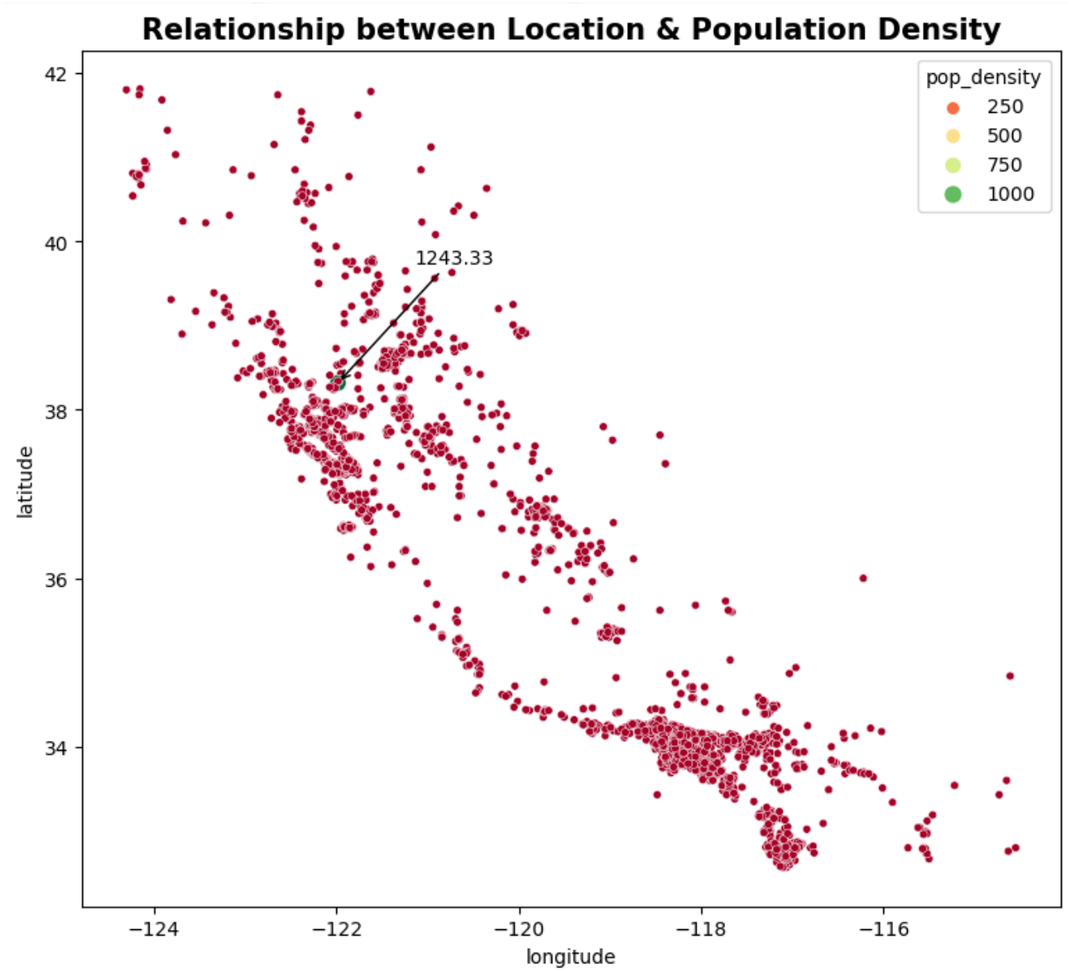
Subplotting Population Density against House Value

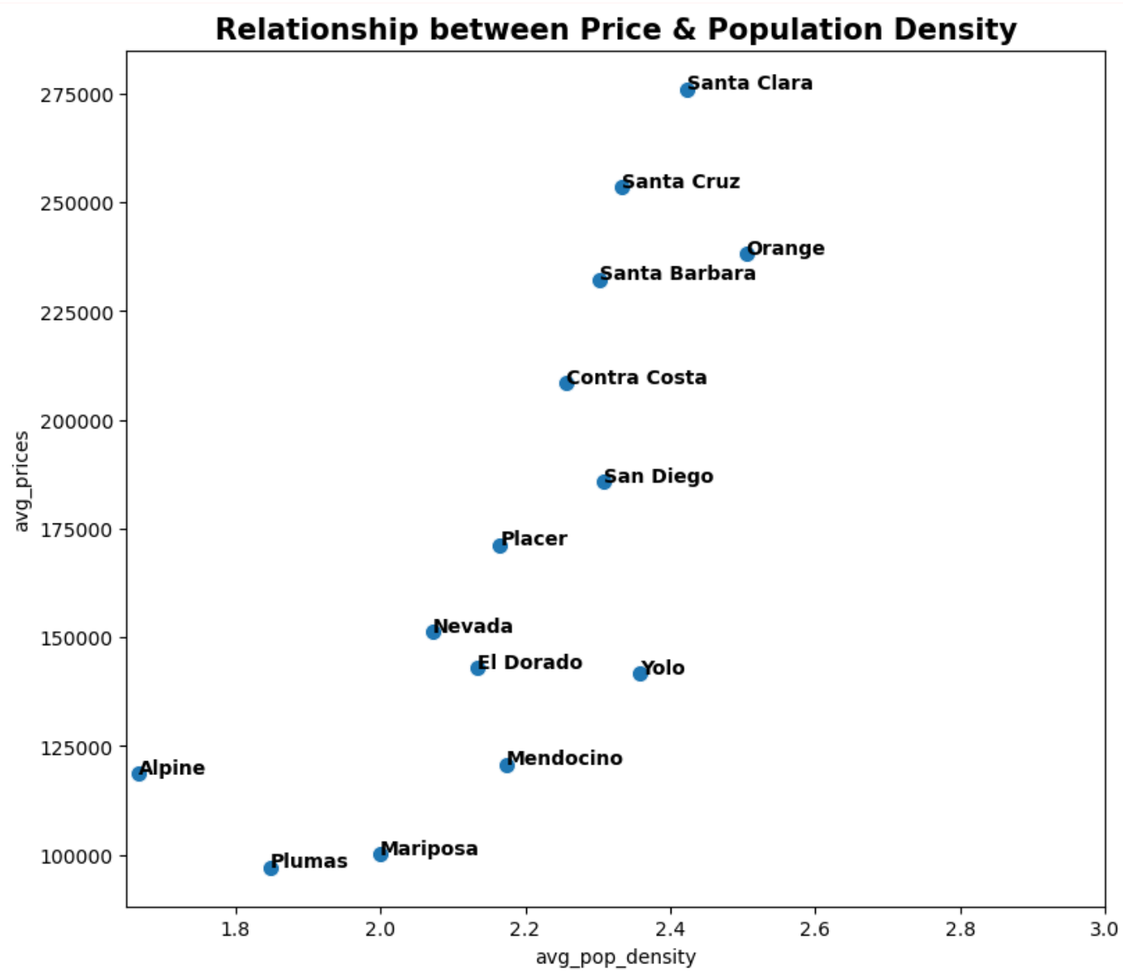
Population Density is usually considered as the number of people living within a square mile. As one might notice, our original DataFrame provided no data regarding the area of each county or even for the state of California. However, we can derive this using the existing variables. We know that 'Population' measures the number of people living within each pair of coordinates. Similiarly, 'Households' provides the number of people living in a block, thus if we use the block as an alternative for the area, we can divide both variables to get density.

The graphs help in displaying how the population density changes with location. We can see that there is a negative trend between the two from the first graph. The second one plots the relationship between price and population density measured across different counties. It depicts a slight positive trend between the two as we see that densely populated regions such as Santa Cruz and Santa Barbara also happen to expensive places to purchase homes while the same in sparse populated regions like Alpine or Plumas is cheaper.

In [89]:
```
random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=3000)
population_df = df3.loc[:, ['longitude', 'latitude', 'population', 'households', 'median_house_value', 'ocean_proximity']]
population_df['pop_density'] = population_df['population']/population_df['households']
population_df['coordinates'] = list(zip(population_df['longitude'], population_df['latitude']))
population_df['coordinates'] = population_df['coordinates'].apply(Point)


county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
CA_county = county_df.query("STATEFP == '06'")
CA_county.set_index('COUNTYFP')

# Converting into GeoDataFrame
city_gdf = gpd.GeoDataFrame(population_df, crs=4269, geometry='coordinates')
merged_gdf = gpd.sjoin(city_gdf, CA_county, op="within")
avg_pop_density = {}
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    pop_density = int(column_values['pop_density'])
    if county in avg_pop_density:
        avg_pop_density[county].append(pop_density)
    else:
        avg_pop_density[county] = [pop_density]
for x in avg_pop_density:
    avg_pop_density[x] = statistics.mean(avg_pop_density[x])

max_density = max(list(avg_pop_density.values()))
min_density = min(list(avg_pop_density.values()))

count = []
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    if county not in count:
        merged_gdf.at[index, 'avg_pop_density'] = avg_pop_density[county]
        count.append(county)
    else:
        merged_gdf = merged_gdf.drop(index)

merged_gdf = merged_gdf.set_index('COUNTYFP')
final_gdf = merged_gdf.join(gdf['avg_prices'])
final_gdf = final_gdf.sort_values(ascending=True, by=['avg_pop_density'])
final_gdf1 = final_gdf.loc[['061', '057', '003', '063', '043', '017', '045', '083', '087', '085', '013', '073', '113', '059'], :]

fig, ax = plt.subplots(1, 2, figsize=(20, 8))
sns.scatterplot(data=population_df.iloc[sample], x="longitude", y="latitude", size='pop_density',
                hue='pop_density', palette='RdYlGn', ax=ax[0], s=200)
sns.scatterplot(data=final_gdf1, x="avg_pop_density", y="avg_prices", ax=ax[1], s=77)

data = population_df.iloc[sample]
max_pop_density = max(list(data['pop_density']))
for row in data.iterrows():
```

```
    index, column_values = row
    pop_density = column_values['pop_density']
    longitude = column_values['longitude']
    latitude = column_values['latitude']
    if pop_density == max_pop_density:
        ax[0].annotate(round(pop_density, 2), xy=(longitude,latitude), xytext=(60,60), textcoords='offset points', ha='center', va='bottom',
                       arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))
for row in final_gdf1.iterrows():
    index, column_values = row
    county = column_values['NAME']
    pop_density = column_values['avg_pop_density']
    prices = column_values['avg_prices']
    ax[1].annotate(county, xy=(pop_density, prices), weight='bold')

ax[1].set_xlim(1.65, 3)
ax[0].set_title('Relationship between Location & Population Density', fontsize=15, fontweight='bold')
ax[1].set_title('Relationship between Price & Population Density', fontsize=15, fontweight='bold')
plt.show()
```



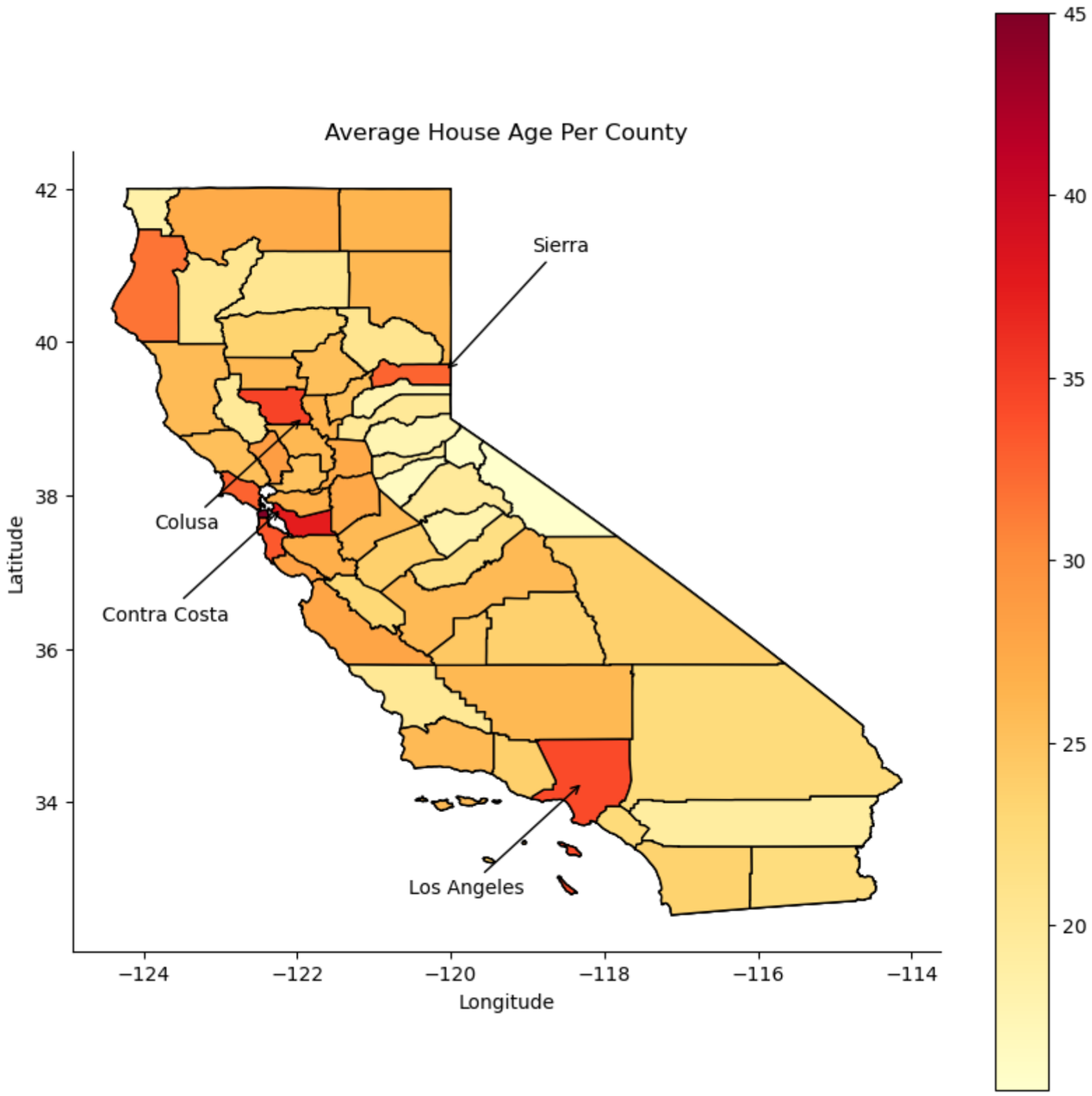


Maps and Interpretation

Mapping and Analysing Housing Age

In [85]:
```
# creating new dataframe with just the coordinates

df_calif_age = df3.loc[:, ['housing_median_age', 'latitude', 'longitude']]
df_calif_age['coordinates'] = list(zip(df_calif_age['longitude'], df_calif_age['latitude']))
df_calif_age['coordinates'] = df_calif_age['coordinates'].apply(Point)

state_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_state_5m.zip")
county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
CA_county = county_df.query("STATEFP == '06'")
CA_county.crs

# Converting into GeoDataFrame
city_gdf = gpd.GeoDataFrame(df_calif_age, crs=4269, geometry='coordinates')
merged_gdf = gpd.sjoin(city_gdf, CA_county, op="within")
avg_age = {}
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    age = int(column_values['housing_median_age'])
    if county in avg_age:
        avg_age[county].append(age)
    else:
        avg_age[county] = [age]
for x in avg_age:
    avg_age[x] = statistics.mean(avg_age[x])

max_age = max(list(avg_age.values()))
min_age = min(list(avg_age.values()))

count = []
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    if county not in count:
        merged_gdf.at[index, 'avg_house_age'] = avg_age[county]
        count.append(county)
    else:
        merged_gdf = merged_gdf.drop(index)
CA_county = CA_county.set_index('COUNTYFP')
for row in merged_gdf.iterrows():
    index, column_values = row
    countyfp = column_values['COUNTYFP']
    geometry = CA_county.loc[countyfp, 'geometry']
    merged_gdf.at[index, 'geometry'] = geometry
merged_gdf1 = gpd.GeoDataFrame(merged_gdf, geometry='geometry')

# Plot the graph
state_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_state_5m.zip")
fig, gax = plt.subplots(figsize=(10, 10))
state_df.query("NAME == 'California'").plot(ax=gax, edgecolor="black", color="white")
CA_county.plot(ax=gax, edgecolor="black", color="white")
merged_gdf1.plot(ax=gax, edgecolor='black', legend=True, column='avg_house_age', cmap='YlOrRd', vmin=min_age, vmax=45)

counties = ['Los Angeles', 'Contra Costa', 'Colusa']

for x, y, label in zip(merged_gdf1['longitude'], merged_gdf1['latitude'], merged_gdf1['NAME']):
    if label in counties:
        gax.annotate(label, xy=(x,y), xytext=(-60,-60), textcoords='offset points', ha='center', va='bottom',
                     arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))
    elif label == 'Sierra':
```

```
        gax.annotate(label, xy=(x,y), xytext=(60,60), textcoords='offset points', ha='center', va='bottom',
                     arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))
plt.xlabel('Longitude')
plt.ylabel('Latitude')
gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)
plt.title('Average House Age Per County')
plt.show()
```



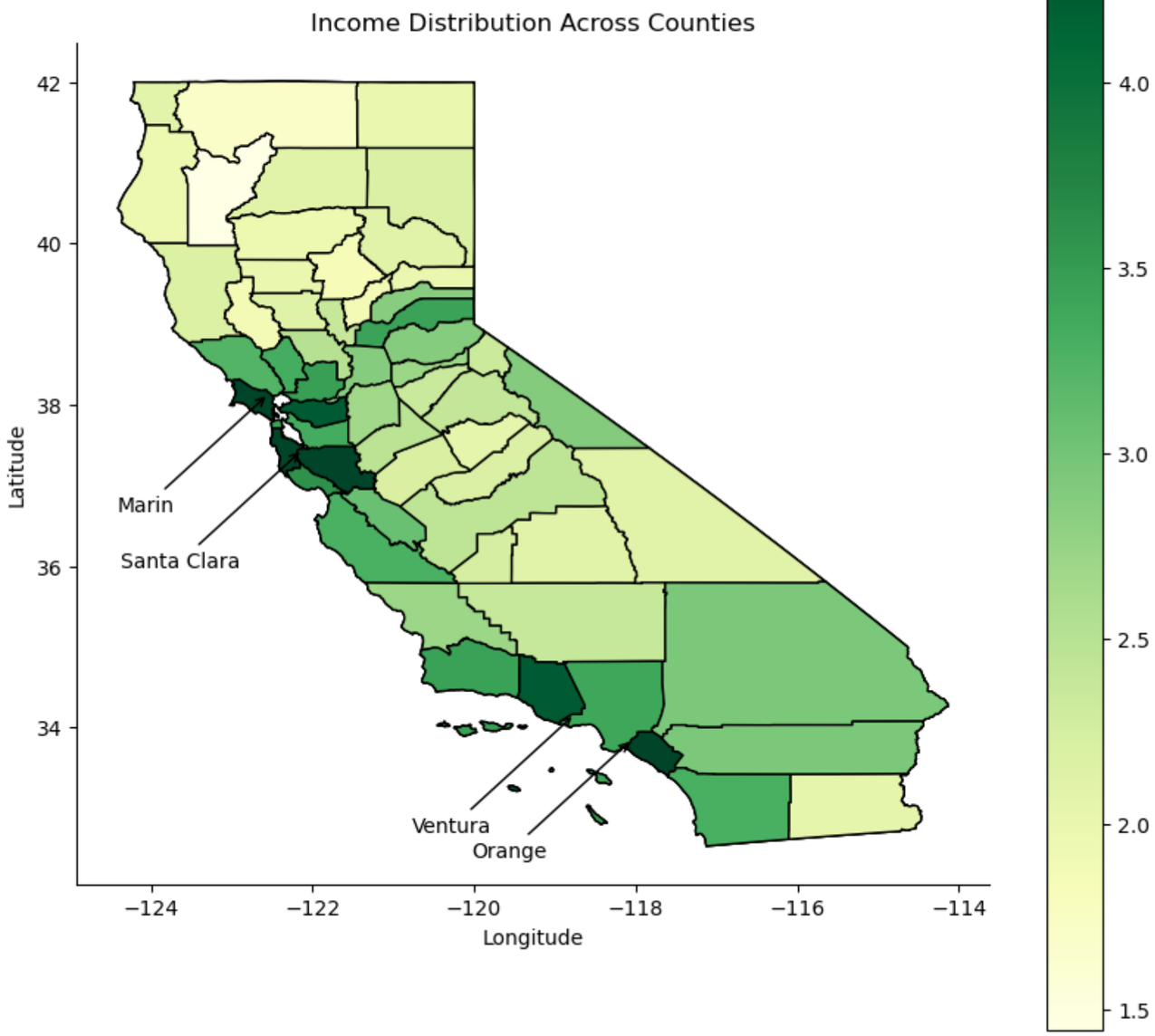
The map below showcases the state of California with its respective counties. This map mainly depicts the average age of all houses that are built in the respective county with the help of color differences. We use this color coordination to mark all counties with old houses such as the ones above the age of 25 years with a darker color as shown below while those that are relatively new with a slightly lighter color.

Housing Age is an important independent variable which is often used by consumers in deciding to purchase houses irrespective of the location. Using the graphical tool of Scatter Plots earlier, we attempted to showcase a relationship between the age of the house and its ocean proximity, ultimately highlighting two main categories of ocean proximites: 'INLAND' and 'NEAR OCEAN'. We can say the same using the map below, we see that most of the old houses are either found near the western coastline of California or deep inside the state in popular counties such as Sacramento, Nevada, Santa Bernardino etc. This variable is crucial in our analysis since we will club it along with the Average House Value in each County to derive the relationship between the two geographically.

Mapping and Analysing Income Distribution

In [8]:
```
df_calif_income = df3.loc[:, ['median_income', 'latitude', 'longitude']]
df_calif_income['coordinates'] = list(zip(df_calif_income['longitude'], df_calif_income['latitude']))
df_calif_income['coordinates'] = df_calif_income['coordinates'].apply(Point)

state_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_state_5m.zip")
county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
CA_county = county_df.query("STATEFP == '06'")
CA_county.crs

# Converting into GeoDataFrame
city_gdf = gpd.GeoDataFrame(df_calif_income, crs=4269, geometry='coordinates')
merged_gdf = gpd.sjoin(city_gdf, CA_county, op="within")
avg_income = {}
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    income = int(column_values['median_income'])
    if county in avg_income:
        avg_income[county].append(income)
    else:
        avg_income[county] = [income]
for x in avg_income:
    avg_income[x] = statistics.mean(avg_income[x])

max_income = max(list(avg_income.values()))
min_income = min(list(avg_income.values()))

count = []
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    if county not in count:
        merged_gdf.at[index, 'avg_income'] = avg_income[county]
        count.append(county)
    else:
        merged_gdf = merged_gdf.drop(index)
CA_county = CA_county.set_index('COUNTYFP')
for row in merged_gdf.iterrows():
    index, column_values = row
    countyfp = column_values['COUNTYFP']
    geometry = CA_county.loc[countyfp, 'geometry']
    merged_gdf.at[index, 'geometry'] = geometry
merged_gdf2 = gpd.GeoDataFrame(merged_gdf, geometry='geometry')

fig, gax = plt.subplots(figsize=(10,10))
state_df.query("NAME == 'California'").plot(ax=gax, edgecolor="black", color="white")
CA_county.plot(ax=gax, edgecolor="black", color="white")
merged_gdf2.plot(ax=gax, edgecolor='black', legend=True, column='avg_income', cmap='YlGn', vmin=min_income, vmax=4.5)

counties = ['Marin', 'Santa Clara', 'Orange', 'Ventura']
```

```
for x, y, label in zip(merged_gdf2['longitude'], merged_gdf2['latitude'], merged_gdf2['NAME']):
    if label in counties:
        gax.annotate(label, xy=(x,y), xytext=(-60,-60), textcoords='offset points', ha='center', va='bottom',
                     arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))

gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')
gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)
plt.title('Income Distribution Across Counties')
plt.show()
```

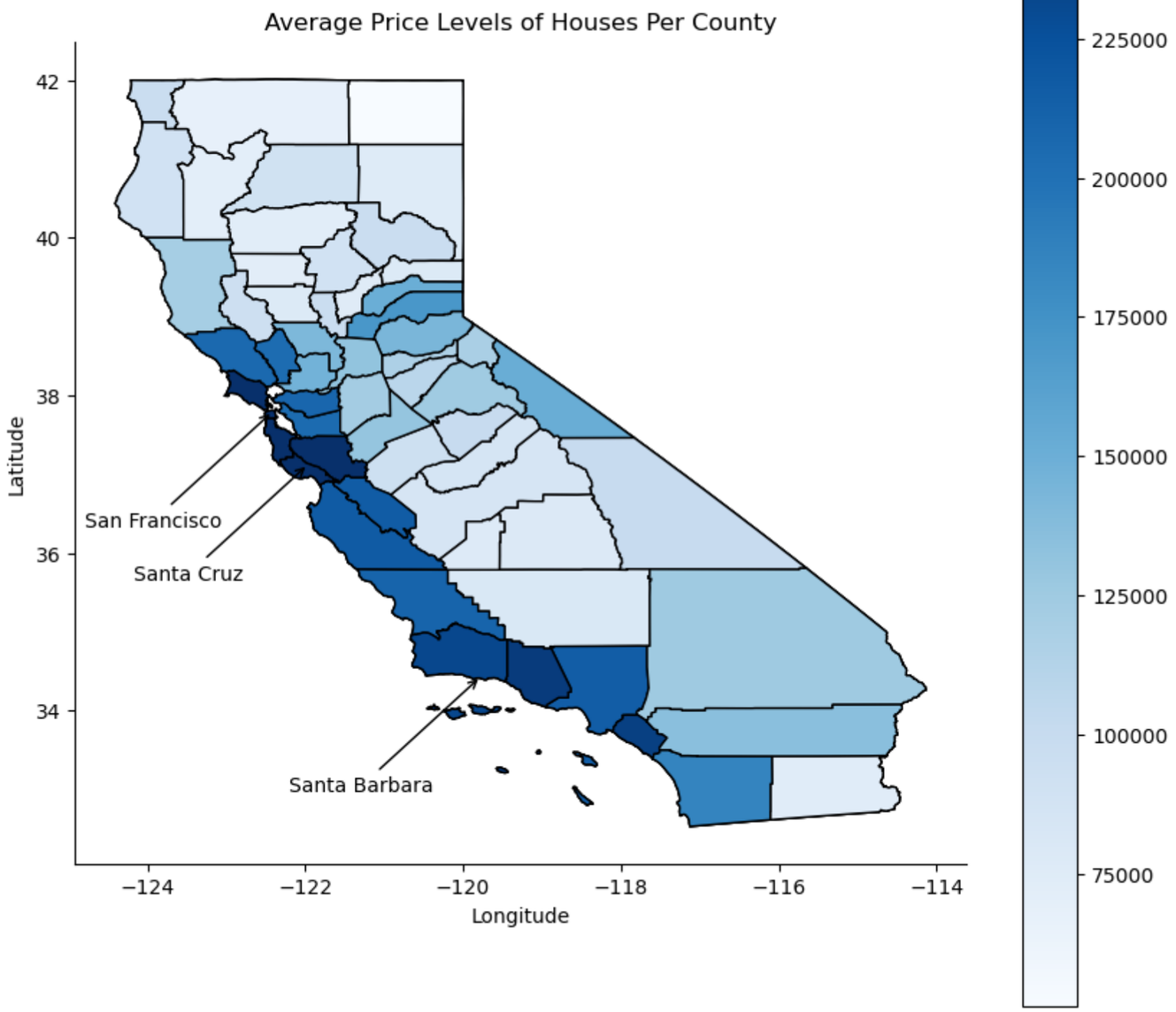


The map below displays how the Income of consumers provided to us in the original Dataframe have been distributed across different counties. We have used the same technique of color differences to depict the varying degrees of Incomes across the state, with a slightly darker color representing high income counties that are well off and the lighter shade for low income counties. Income was another crucial independent variable that was picked for this analysis since it will enable us to understand the purchasing behaviour of consumers among different income stratas. In order to achieve this, we will geographically plot the relationship of Income along with their repective House Values. The following map was a result of the acummulation of income across different coordinates given in the Dataset which were later traced back to their counties and averaged out. A key observation that we can make using this map is that all counties on the coastline are well off with higher average incomes as compared to those inland. This can be for a multitude of factors such as the accrual of economic benefits from ocean navigation, coastal fisheries, tourism and recreation.

Mapping and Analysing House Value

In [79]:
```
df_calif_price = df3.loc[:, ['median_house_value', 'latitude', 'longitude']]
df_calif_price['coordinates'] = list(zip(df_calif_price['longitude'], df_calif_price['latitude']))
df_calif_price['coordinates'] = df_calif_price['coordinates'].apply(Point)

state_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_state_5m.zip")
county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
CA_county = county_df.query("STATEFP == '06'")
CA_county.crs

# Converting into GeoDataFrame
city_gdf = gpd.GeoDataFrame(df_calif_price, crs=4269, geometry='coordinates')
merged_gdf = gpd.sjoin(city_gdf, CA_county, op="within")

# Removing Outliers
q1 = merged_gdf['median_house_value'].quantile(0.25)
q3 = merged_gdf['median_house_value'].quantile(0.75)
IQR = q3 - q1
upp_bound = q3 + IQR*1.5
low_bound = q1 - IQR*1.5

for row in merged_gdf.iterrows():
    index, column_values = row
    if low_bound <= column_values['median_house_value'] <= upp_bound:
        continue
    else:
        merged_gdf = merged_gdf.drop(index)

avg_prices = {}
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    price = int(column_values['median_house_value'])
    if county in avg_prices:
        avg_prices[county].append(price)
    else:
        avg_prices[county] = [price]
for x in avg_prices:
    avg_prices[x] = statistics.mean(avg_prices[x])

max_price = max(list(avg_prices.values()))
min_price = min(list(avg_prices.values()))

count = []
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    if county not in count:
        merged_gdf.at[index, 'avg_prices'] = avg_prices[county]
        count.append(county)
    else:
        merged_gdf = merged_gdf.drop(index)

CA_county = CA_county.set_index('COUNTYFP')

for row in merged_gdf.iterrows():
    index, column_values = row
    countyfp = column_values['COUNTYFP']
    geometry = CA_county.loc[countyfp, 'geometry']
    merged_gdf.at[index, 'geometry'] = geometry
merged_gdf3 = gpd.GeoDataFrame(merged_gdf, geometry='geometry')

fig, gax = plt.subplots(figsize=(10,10))
state_df.query("NAME == 'California'").plot(ax=gax, edgecolor="black", color="white")
```

```
CA_county.plot(ax=gax, edgecolor="black", color="white")
merged_gdf3.plot(ax=gax, edgecolor='black', legend=True, column='avg_prices', cmap='Blues', vmin=min_price, vmax=250000)

counties = ['Santa Cruz', 'San Francisco', 'Santa Barbara']

for x, y, label in zip(merged_gdf3['longitude'], merged_gdf3['latitude'], merged_gdf3['NAME']):
    if label in counties:
        gax.annotate(label, xy=(x,y), xytext=(-60,-60), textcoords='offset points', ha='center', va='bottom',
                     arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))

gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')
gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)
plt.title('Average Price Levels of Houses Per County')
plt.show()
```

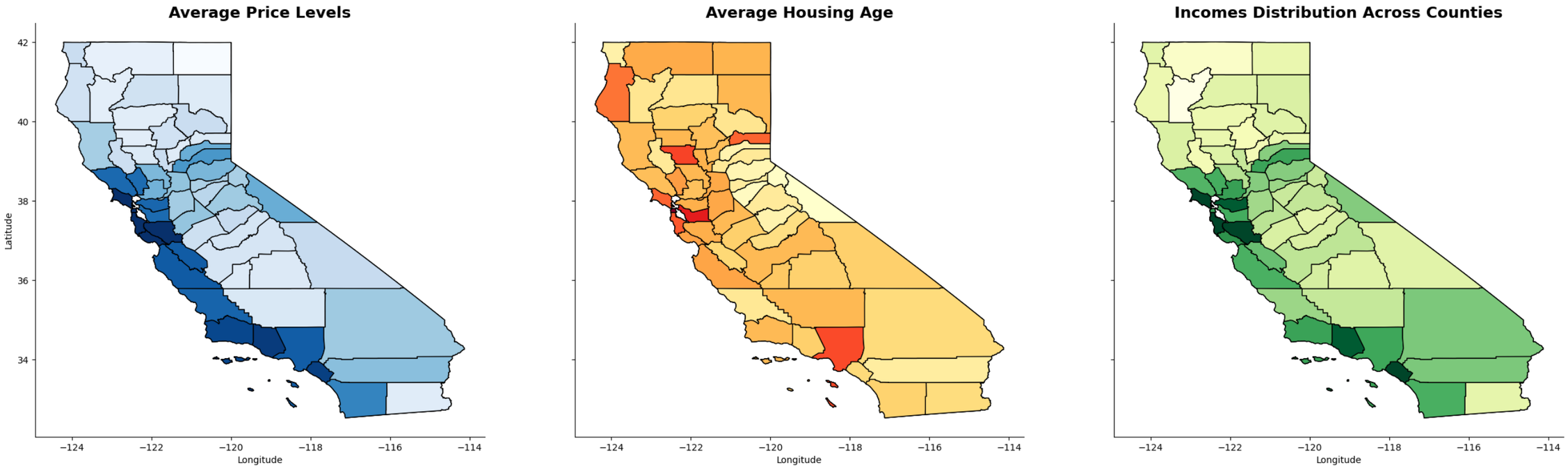


The map created below enables us to visualize the average price levels in each county across California based on the data provided in the raw DataFrame. Quite similiar to the map we created for the Average Income Distribution above, we use the same method of color coding each county based on the level of prices for houses found. What we can observe from the following map is that all counties on the coastline are extremely expensive relative to those found inland. This relationship can be verified from the Boxplot that we created for the discrete variable of Ocean Proximity, as we stated there too that houses found near the ocean or coast occupy the largest share of the Price Distribution Chart. The following map was created using the house values provided across each pair of coordinates, these were mapped to their respective counties which were then finally averaged out. Since there is the possibility of outliers being present, we used the Interquartile Range to find the Upper and Lower Bounds for this data and eliminate any outliers

Comparison of Housing Age, Income and House Value

In [92]:
```
fig, gax = plt.subplots(1, 3, figsize=(30,40), sharey = True)
state_df.query("NAME == 'California'").plot(ax=gax[0], edgecolor="black", color="white")
CA_county.plot(ax=gax[0], edgecolor="black", color="white")
```

```
merged_gdf3.plot(ax=gax[0], edgecolor='black', legend=False, column='avg_prices', cmap='Blues', vmin=min_price, vmax=250000)

state_df.query("NAME == 'California'").plot(ax=gax[1], edgecolor="black", color="white")
CA_county.plot(ax=gax[1], edgecolor="black", color="white")
merged_gdf1.plot(ax=gax[1], edgecolor='black', legend=False, column='avg_house_age', cmap='YlOrRd', vmin=min_age, vmax=45)

state_df.query("NAME == 'California'").plot(ax=gax[2], edgecolor="black", color="white")
CA_county.plot(ax=gax[2], edgecolor="black", color="white")
merged_gdf2.plot(ax=gax[2], edgecolor='black', legend=False, column='avg_income', cmap='YlGn', vmin=min_income, vmax=4.5)

gax[0].set_xlabel('Longitude')
gax[0].set_ylabel('Latitude')
gax[0].set_title('Average Price Levels', fontsize=17, fontweight='bold')
gax[0].spines['top'].set_visible(False)
gax[0].spines['right'].set_visible(False)

gax[1].set_title('Average Housing Age', fontsize=17, fontweight='bold')
gax[1].set_xlabel('Longitude')
gax[1].spines['top'].set_visible(False)
gax[1].spines['right'].set_visible(False)

gax[2].set_title('Incomes Distribution Across Counties', fontsize=17, fontweight='bold')
gax[2].set_xlabel('Longitude')
gax[2].spines['top'].set_visible(False)
gax[2].spines['right'].set_visible(False)
plt.show()
```



Since we have now successfully created 3 maps, 2 for our independent variables and 1 for the dependent variable, we can combine all three of these to derive various relationships and approach our research question with new insights. The question we wished to answer using this analysis was how do different independent variables such as income and housing age affect the consumer's decision to buy a house. We can now gradually observe how these variables relate to each other using the maps below.

We can see that all counties along the coast are darker in color in both maps 1 as well as 3, implying that these counties are not only well off but expensive as well. Intuitively, if we think about it, more income of a consumer raises his/her purchasing power, which enables him to buy higher priced houses. Thus, counties with higher incomes should have more pricey houses as well as shown below. The housing age map indicates that a similiar relationship with price exists as well, where houses found or built near the ocean as well as some places inland are much more old and aged increasing their worth and value, thereby prices being higher for those counties.

Project Three

Potential Data To Scrape

One of the many factors that a buyer considers before purchasing a house in any location is the various school districts that the prospective area (county) has to offer to him assuming he has a family. He will be quite keen on understanding which public and private schools are present in that locality and are are conveniently distanced from the region where he has potentially identified a house.

At times, the buyer actually reverses this line of thought and instead looks initially at the best schools affordable in the county/state and then starts shortlisting potential candidates for buying a house around these areas. Hence, this becomes an important factor to consider when a consumer begins to look at buying a house. The original dataset provided alongside our research unfortunately does not include any information related to this. I believe that when this additional information is also available to the buyer, it may influence his/her decision to zero-down on specific properties. The website used for this purpose is the California's state government webpage, http://cde.data.gov, which lists down all the K-12 educational institutions available.

The information that I hope to find on the website mentioned above is about the number of public and private schools available per county for majority of the regions. Addititionally, I am hoping to get the precise coordinates in terms of the latitude and longitude of each school which can then be added to the original dataset as an additional column and allow us to map the school locations geographically.

Another data table that we were successful at scraping was the Criminal Activity data for the state of California using the Wikipedia webpage https://en.m.wikipedia.org/wiki/California_locations_by_crime_rate. This table depicts the different kinds of Crime that were dominant in the state and their respective frequency across each county. Crime is another crucial factor that most families take into consideration before purchasing a home in a particular region. They would usually want to avoid these areas as much as possible which means that crime as a variable does affect or influence the decision of a consumer into buying a house.

Potential Challenges

The challenges that I faced while scraping the data explained above were of the following nature:

- Traversing numerous links usually brought me to a certain page where the relevant data was sorted and fit into various excel sheets instead of being displayed on the webpage. Moreover, access to these Excel files was prohibited which indicated that they were not available to the Public.
- If the data was available, most of the excel sheets on the browser would either end up timing out or refuse to open due to various errors .
- Additionally, most of the data preceding the year of 2000 is either missing or unavailable making it difficult to make accurate analysis.

We don't need to run the program and functions multiple times as the data is static for a particular year and our research also requires us to make analysis for a certain period only. However, if we move into a new year then we would need to update this data each year. Moreover, the original dataset would have to be modified as well since the number of houses would keep varying each year.

The biggest obstacle I have faced so far is to not be able to scrape the data from the California State Government's website to retrieve information regarding all schools. I believe the data has not been made public as of yet and its access is restricted to us. I have been able to access the site, browse its contents, examine the relevant portions of data but have been unsuccessful at scraping it. There may have been alternative API's to access the site but I have not been able to identify them yet. This will continue to be a constant challenge in our process of retrieving data since not every dataset that we wish to scrape is available in a simple and structured format on webpages.

Scraping Data from a Website

We will be scraping the Wikipedia page regarding the crime rates available for each county. In order to begin, we first need the URL or the Uniform Resource Locator which is the address of the webpage from where we wish to scrape the data *(Line 1)*.

We use the python library of **requests** to push a request to the web server to allow us to access the data on the Wikipedia page and retrieve it *(Line 2)*. The data provided to us by the server is in a clustered, unstructured and unorganised format which makes it very difficult to perform any kind of analysis. This is why we use another useful library provided by Python known as **BeautifulSoup** which simply reshapes the original data provided to us in a much more meaningful and comprehensible form *(Line 3)*.

Our last step in scraping the data would be filter it for only the information provided by the data table on the Wikipedia website. In order to do that, we would need to find the name and class of the respective table and pass it through the method **find_all** which is part of the BeautifulSoup library. This helps us in cleaning the extensive data available on the webpage for only the observations created in the table *(Line 4)*.

In [9]:
```
web_url = 'https://en.m.wikipedia.org/wiki/California_locations_by_crime_rate' # URL for the webpage to be scraped
response = requests.get(web_url) # Sending a get request to the web server in order to access the content of the page
soup_object = BeautifulSoup(response.content) # Converting the response to a meaningful data object
data_table = soup_object.find_all('table', 'wikitable sortable')[0] # Filtering the data for only table values
all_values = data_table.find_all('tr')
```

Before we move on to merge this data with other datasets, the most important task is to convert this new data into a fresh dataframe using the **DataFrame** tool provided within the pandas module. So, below we have created a new dataframe called **crime_rates** with all the relevant columns *(Line 1)*.

In order to parse through all the values in the table, we simply use the **find_all** method to collect all the values which are written within the HTML tag *tr* meaning a table row. This gathers all the observations in the table row-wise and enables us to view them the same way. Next, we set a for loop to go through each and every observation and create a variable for each value corresponding to the respective column they are found in. This is shown in *Lines 6 - 14*.

Finally, at the end of each loop, after collecting all the values in a particular row for all columns, we insert them into the new data using our **loc** slicing tool. Once done, we move on to the next row or observation and repeat the process for k^{th} rows.

In [10]:
```
# Creating a new dataframe
crime_rates = pd.DataFrame(columns = ['County', 'Population', 'Population Density',
                                      'Violent Crimes', 'Violent Crimes Per 1000 People',
                                      'Property Crimes', 'Property Crimes Per 1000 People'])
ix = 0

# Setting a loop to parse through all observations
for row in all_values[1:]:
    values = row.find_all('td')
    county = values[0].text
    pop = values[1].text
    pop_density = values[2].text
    violent_crimes = values[3].text
    violent_1000 = values[4].text
    property_crimes = values[5].text
    property_1000 = values[6].text.strip('\n')

    crime_rates.loc[ix] = [county, pop, pop_density, violent_crimes, violent_1000, property_crimes, property_1000]
    ix += 1

crime_rates.head(10)
```

Out[10]:

	County	Population	Population Density	Violent Crimes	Violent Crimes Per 1000 People	Property Crimes	Property Crimes Per 1000 People
0	Alameda	1,559,308	2,109.8	10,356	6.6	57,620	37.0
1	Alpine	1,202	1.6	4	3.3	24	20.0
2	Amador	37,159	62.5	81	2.2	629	16.9
3	Butte	221,578	135.4	678	3.1	6,631	29.9
4	Calaveras	44,921	44.0	113	2.5	989	22.0
5	Colusa	21,424	18.6	40	1.9	350	16.3
6	Contra Costa	1,081,232	1,496.0	3,650	3.4	32,232	29.8
7	Del Norte	28,066	27.9	165	5.9	649	23.1
8	El Dorado	181,465	106.3	409	2.3	3,138	17.3
9	Fresno	948,844	159.2	4,547	4.8	32,535	34.3

Merging the Scraped Dataset

Up until now, we have successfully scraped the data for crime rates in the state of California for each county using the Wikipedia webpage as a source. We have also reshaped the raw data retrieved from the webpage into a structured and organised dataframe which we have called as **crime_rates**. Our next step would involve merging this new dataset with our original modified version which was used to analyse and build relationships between various variables. We will perform an outer merge so as to not leave behind any observation. Since we want to merge the two datasets with a common key of *County*, but the variable name for it is different between both dataframes, we use the **on** paramater and mention both left and right on. The table below shows the first 10 out of 58 observations for the merged dataset. We have 58 observations since our entire analysis is based off each county and the number of counties in California stand at 58.

In [11]:
```
population_df = df3.loc[:, ['longitude', 'latitude', 'population', 'households']]
population_df['pop_density'] = population_df['population']/population_df['households']
population_df['coordinates'] = list(zip(population_df['longitude'], population_df['latitude']))
population_df['coordinates'] = population_df['coordinates'].apply(Point)


county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
CA_county = county_df.query("STATEFP == '06'")
CA_county.set_index('COUNTYFP')

# Converting into GeoDataFrame
city_gdf = gpd.GeoDataFrame(population_df, crs=4269, geometry='coordinates')
merged_gdf = gpd.sjoin(city_gdf, CA_county, op="within")
avg_pop_density = {}
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    pop_density = int(column_values['pop_density'])
    if county in avg_pop_density:
        avg_pop_density[county].append(pop_density)
    else:
        avg_pop_density[county] = [pop_density]
for x in avg_pop_density:
    avg_pop_density[x] = statistics.mean(avg_pop_density[x])

max_density = max(list(avg_pop_density.values()))
min_density = min(list(avg_pop_density.values()))

count = []
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
```

```
    if county not in count:
        merged_gdf.at[index, 'avg_pop_density'] = avg_pop_density[county]
        count.append(county)
    else:
        merged_gdf = merged_gdf.drop(index)
merged_gdf1 = merged_gdf.set_index('COUNTYFP')
final_gdf1 = merged_gdf1.join(gdf['avg_prices'])
final_gdf1 = final_gdf1.reset_index()
final_merged_gdf1 = pd.merge(final_gdf1, crime_rates, left_on='NAME', right_on='County', how='outer')
final_merged_gdf2 = final_merged_gdf1.drop(['population', 'pop_density', 'households', 'ALAND', 'AWATER', 'AFFGEOID',
                                            'latitude', 'longitude', 'STATEFP', 'NAME', 'LSAD', 'COUNTYNS',
                                            'GEOID', 'index_right'], axis=1)
CA_county = CA_county.set_index('COUNTYFP')
final_merged_gdf2.head(10)
```

Out[11]:

	COUNTYFP	coordinates	avg_pop_density	avg_prices	County	Population	Population Density	Violent Crimes	Violent Crimes Per 1000 People	Property Crimes	Property Crimes Per 1000 People
0	001	POINT (-122.23000 37.88000)	2.210109	204395.175879	Alameda	1,559,308	2,109.8	10,356	6.6	57,620	37.0
1	013	POINT (-122.19000 37.84000)	2.257191	208413.263525	Contra Costa	1,081,232	1,496.0	3,650	3.4	32,232	29.8
2	003	POINT (-119.78000 38.69000)	1.666667	118700.000000	Alpine	1,202	1.6	4	3.3	24	20.0
3	005	POINT (-120.56000 38.48000)	2.535714	117146.428571	Amador	37,159	62.5	81	2.2	629	16.9
4	007	POINT (-121.83000 39.76000)	2.115385	89611.538462	Butte	221,578	135.4	678	3.1	6,631	29.9
5	009	POINT (-120.46000 38.15000)	2.031250	107893.750000	Calaveras	44,921	44.0	113	2.5	989	22.0
6	011	POINT (-121.91000 39.03000)	2.312500	77731.250000	Colusa	21,424	18.6	40	1.9	350	16.3
7	095	POINT (-122.21000 38.06000)	8.587940	147259.798995	Solano	421,624	513.1	2,109	5.0	13,453	31.9
8	015	POINT (-124.17000 41.80000)	2.454545	97163.636364	Del Norte	28,066	27.9	165	5.9	649	23.1
9	017	POINT (-119.95000 38.95000)	2.133333	142900.840336	El Dorado	181,465	106.3	409	2.3	3,138	17.3

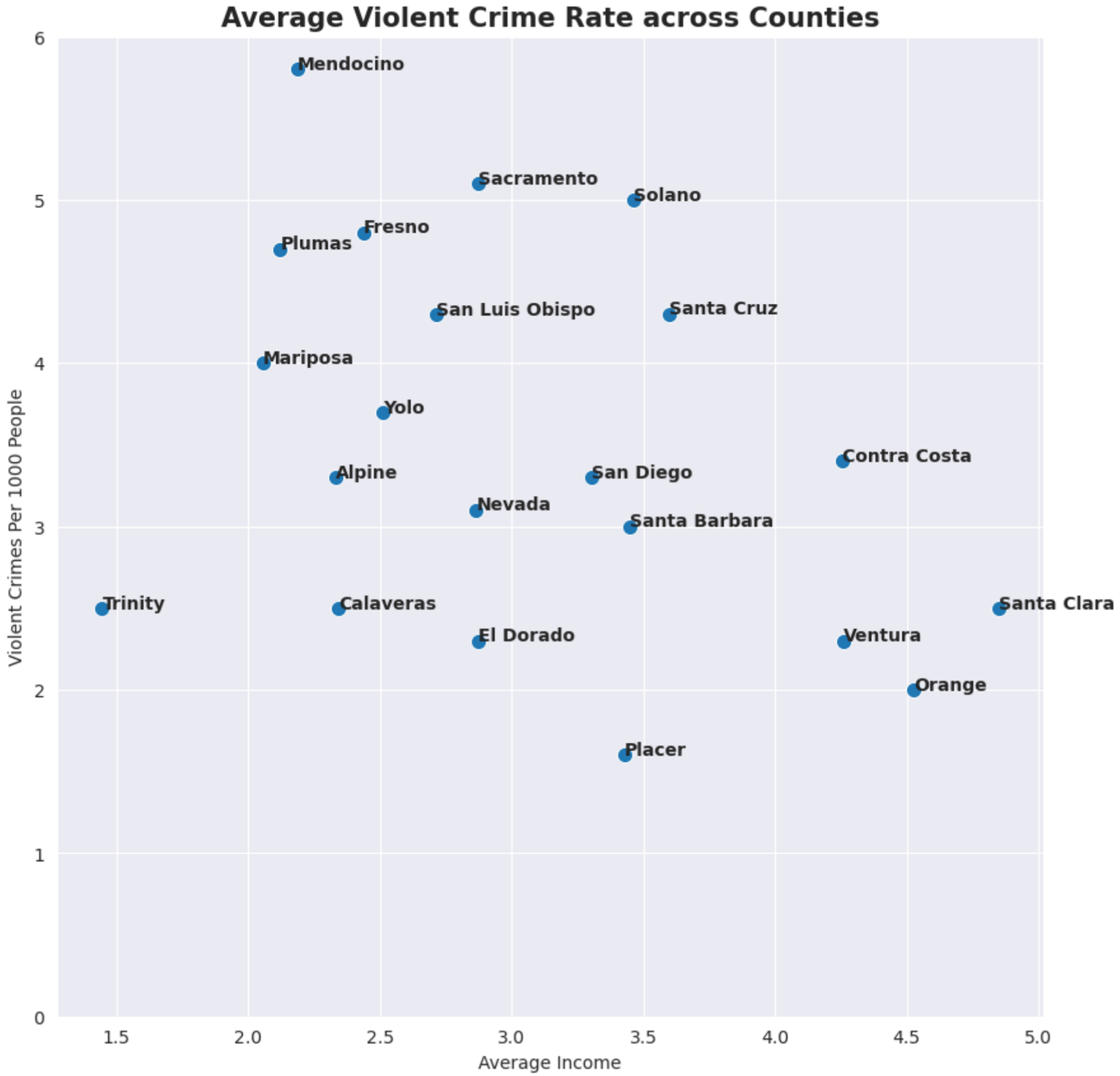
Visualization

Examination Into Violent Crimes

In [109…

```
# Creating a new geometry column

fig, gax = plt.subplots(figsize=(10,10))
final_merged_gdf3 = pd.merge(final_merged_gdf2, merged_gdf2, left_on='County', right_on='NAME')
final_merged_gdf3 = final_merged_gdf3.drop(['Population', 'Population Density', 'ALAND',
                                            'AWATER', 'AFFGEOID', 'coordinates_x', 'coordinates_y',
                                           'latitude', 'longitude', 'STATEFP', 'NAME', 'LSAD', 'COUNTYNS',
                                            'GEOID', 'index_right', 'COUNTYFP_x', 'median_income'], axis=1)
final_merged_gdf3 = final_merged_gdf3.astype({'Violent Crimes Per 1000 People': 'float64',
```

```
                                            'Property Crimes Per 1000 People': 'float64'})
final_merged_gdf3 = final_merged_gdf3.sort_values(ascending=True, by=['Violent Crimes Per 1000 People'])
final_merged_gdf3 = final_merged_gdf3.set_index('COUNTYFP_y')
final_merged_gdf3 = final_merged_gdf3.loc[['061', '057', '003', '063', '043', '017', '045', '083',
                                           '087', '085', '013', '073', '113', '059', '067', '095', '105',
                                           '111', '009', '079', '019'], :]

sns.scatterplot(data=final_merged_gdf3, x="avg_income", y="Violent Crimes Per 1000 People", ax=gax, s=77)

for row in final_merged_gdf3.iterrows():
    index, column_values = row
    county = column_values['County']
    crimes = column_values['Violent Crimes Per 1000 People']
    income = column_values['avg_income']
    gax.annotate(county, xy=(income, crimes), weight='bold')

gax.set_ylim(0, 6)
gax.set_xlabel('Average Income')
plt.title('Average Violent Crime Rate across Counties', fontweight='bold', fontsize=15)
plt.show()
```



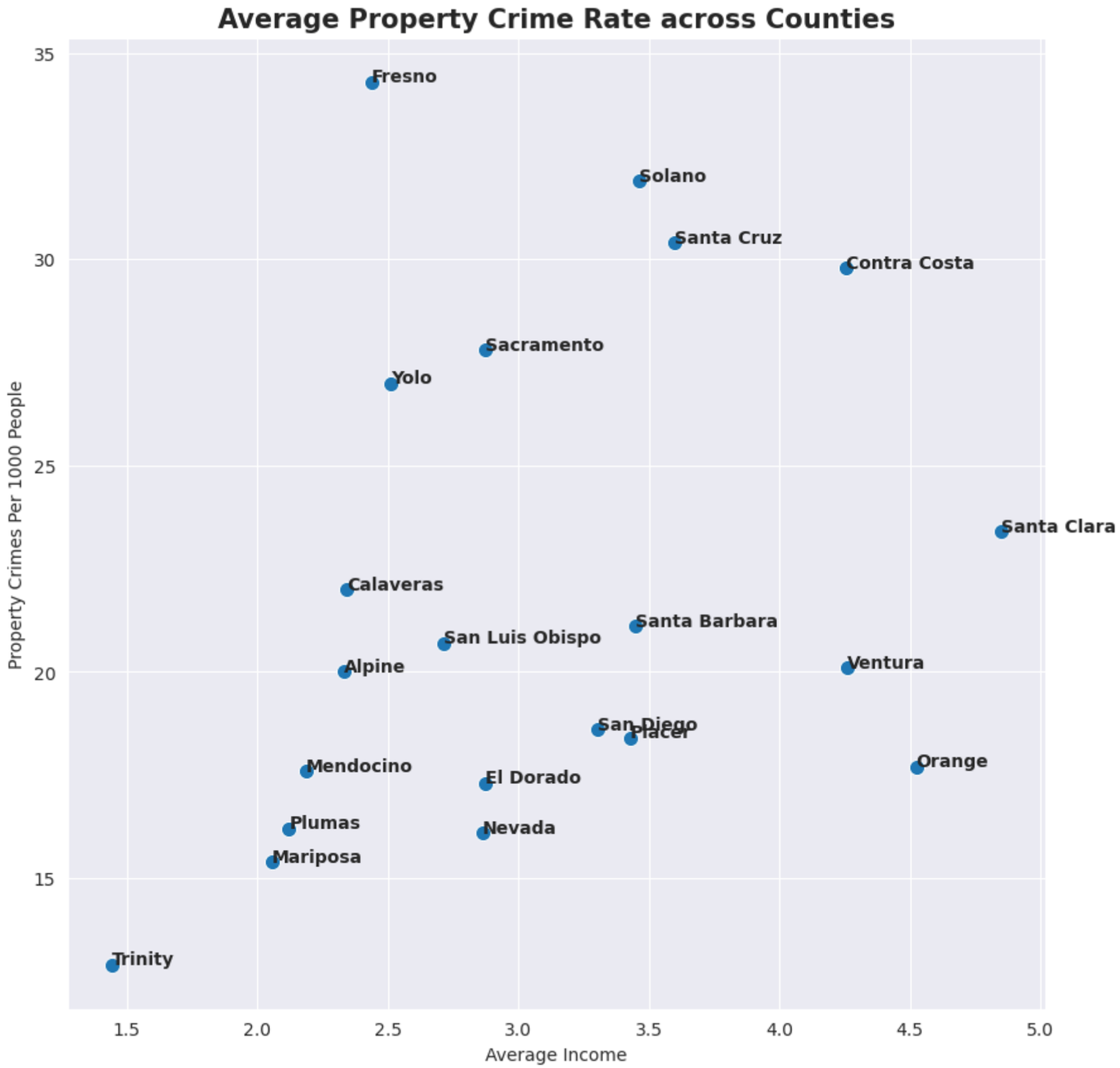
The scatterplot above displays the relationship between the level of Income Distribution and the number of violent crimes in the state of California across different counties. In order to carry out this analysis, we have used the variable **Violent Crimes Per 1000 People** which calculates the amount of crimes taking place for every 1000 people in each county. This data was retrieved from the dataframe shown above by merging the final dataset with our newly scraped information regarding crime. In the above graph, each scatterpoint represents the level of Violent Crimes prevailing for a given level of Average Income within each county in California.

Crime can be considered a fundamental factor which influences the decision of consumers while purchasing homes in certain areas. A clear analysis of the scatterplot indicates that there seems to be a negative linear relationship between Income and the Violent Crime Rate. This is quite intuitive as geographical areas with historically higher than average income distributions will consist of consumers and households who are relatively well off and so should witness lesser crime. On the contrary, localities with bigger disparities in Income distribution will experience higher crime rates since not all people are well off.

Analysis of Property Crimes

In [112…

```
fig, gax = plt.subplots(figsize=(10,10))
sns.scatterplot(data=final_merged_gdf3, x="avg_income", y="Property Crimes Per 1000 People", ax=gax, s=77)

for row in final_merged_gdf3.iterrows():
    index, column_values = row
    county = column_values['County']
    crimes = column_values['Property Crimes Per 1000 People']
    income = column_values['avg_income']
    gax.annotate(county, xy=(income, crimes), weight='bold')

gax.set_xlabel('Average Income')
plt.title('Average Property Crime Rate across Counties', fontweight='bold', fontsize=15)
plt.show()
```



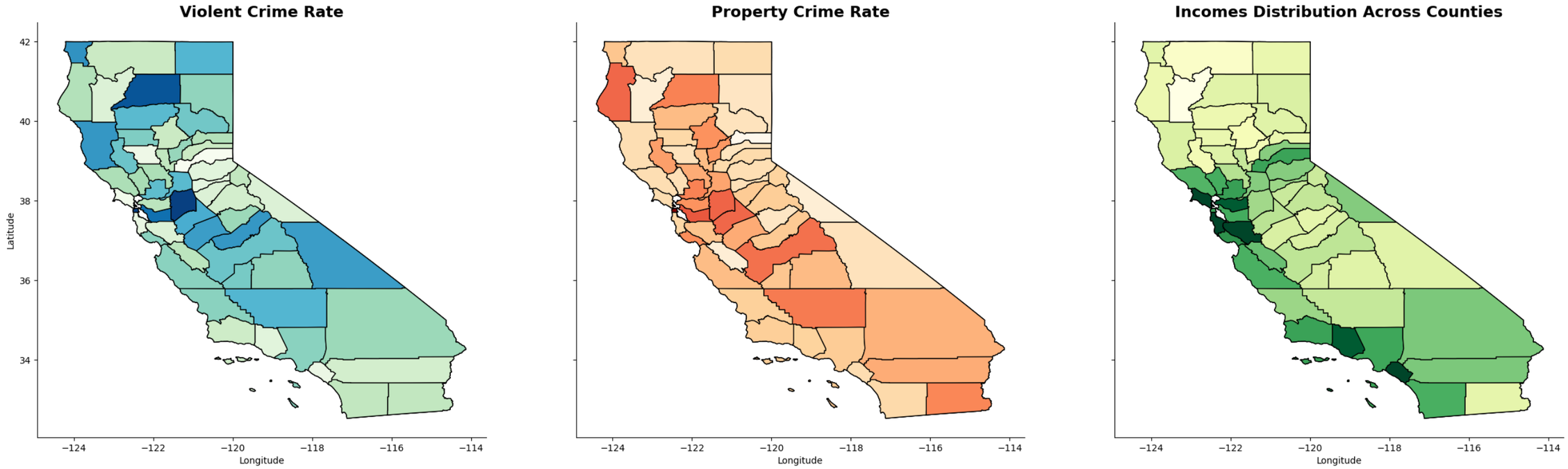
The scatterplot above displays the relationship between the level of Income Distribution and the number of Property crimes in the state of California across different counties. In order to carry out this analysis, we have used the variable **Property Crimes Per 1000 People** which calculates the amount of property crimes taking place for every 1000 people in each county. This data was also retrieved from the dataframe shown above by merging the final dataset with our newly scraped information regarding different sorts of crime. In the above graph, each scatterpoint represents the level of Property Crimes prevailing for a given level of Average Income within each county in California.

Property Crime is the second component to this analysis that we are considering in order to build an effective relationship between Crime and Demand for a House. While examining the graph above, we can notice a somewhat positive linear relationship between the two variables. This is again quite intuitive since Property Crimes are generally witnessed among expensive and high income neighborhoods as homes

located within such areas are much more valuable to break into and steal. Thus, all counties with a higher than average Income Distribution will attract more of such crime towards itself. On the other hand, counties with a lower Income Distribution won't be that attractive for carrying out theft and so we observe relatively a lower than usual crime rate.

Comparison of Violent Crimes, Property Crimes and Income Distribution

In [12]:
```
fig, gax = plt.subplots(1, 3, figsize=(30,40), sharey = True)

# Violent Crime Rate Mapping
for row in final_merged_gdf1.iterrows():
    index, column_values = row
    countyfp = column_values['COUNTYFP']
    geometry = CA_county.loc[countyfp, 'geometry']
    final_merged_gdf1.at[index, 'geometry'] = geometry

final_merged_gdf1 = gpd.GeoDataFrame(final_merged_gdf1, geometry='geometry')
final_merged_gdf1 = final_merged_gdf1.astype({'Violent Crimes Per 1000 People': 'float64',
                                              'Property Crimes Per 1000 People': 'float64'})

max_violent_crimes = max(list(final_merged_gdf1['Violent Crimes Per 1000 People']))
min_violent_crimes = min(list(final_merged_gdf1['Violent Crimes Per 1000 People']))

max_property_crimes = max(list(final_merged_gdf1['Property Crimes Per 1000 People']))
min_property_crimes = min(list(final_merged_gdf1['Property Crimes Per 1000 People']))

state_df.query("NAME == 'California'").plot(ax=gax[0], edgecolor="black", color="white")
CA_county.plot(ax=gax[0], edgecolor="black", color="white")
final_merged_gdf1.plot(ax=gax[0], edgecolor='black', legend=False, column='Violent Crimes Per 1000 People',
                       cmap='GnBu', vmin=min_violent_crimes, vmax=7.5)

# Property Crime Rate Mapping
max_property_crimes = max(list(final_merged_gdf1['Property Crimes Per 1000 People']))
min_property_crimes = min(list(final_merged_gdf1['Property Crimes Per 1000 People']))

state_df.query("NAME == 'California'").plot(ax=gax[1], edgecolor="black", color="white")
CA_county.plot(ax=gax[1], edgecolor="black", color="white")
final_merged_gdf1.plot(ax=gax[1], edgecolor='black', legend=False, column='Property Crimes Per 1000 People',
                       cmap='OrRd', vmin=min_property_crimes, vmax=51.5)

# Income Distribution Mapping
state_df.query("NAME == 'California'").plot(ax=gax[2], edgecolor="black", color="white")
CA_county.plot(ax=gax[2], edgecolor="black", color="white")
merged_gdf2.plot(ax=gax[2], edgecolor='black', legend=False, column='avg_income', cmap='YlGn', vmin=min_income, vmax=4.5)

gax[0].set_xlabel('Longitude')
gax[0].set_ylabel('Latitude')
gax[0].set_title('Violent Crime Rate', fontsize=17, fontweight='bold')
gax[0].spines['top'].set_visible(False)
gax[0].spines['right'].set_visible(False)

gax[1].set_title('Property Crime Rate', fontsize=17, fontweight='bold')
gax[1].set_xlabel('Longitude')
gax[1].spines['top'].set_visible(False)
gax[1].spines['right'].set_visible(False)

gax[2].set_title('Incomes Distribution Across Counties', fontsize=17, fontweight='bold')
gax[2].set_xlabel('Longitude')
gax[2].spines['top'].set_visible(False)
gax[2].spines['right'].set_visible(False)
plt.show()
```



In order to wrap up our Crime analysis, we can now create detailed maps for the different types of Crime Rates in each county and come up with a viable analysis of each by converging our existing knowledge regarding the income distribution of consumers spread across the state with the newly acquired results of the analysis described above using the scatterplots. Our conclusion from the first two maps which represent Violent and Property Crimes respectively was that in both scenarios, counties along the coast or near the ocean were facing crime on a much lower level relative to those counties that were located Inland.

Intuitively, we know that Crimes can have a number of societal reasons and factors leading to its creation but the main component that gives rise to such activities is the loss of income, wealth or well-being. Lower income forces people into committing various acts of crime such as burglary, murders, car thefts etc. So, we can set a direct relation between income and crime in a certain county, in other words, all counties will lower income distribution will exhibit a higher crime rate, property or violence alike. In the 3^{rd} map, we deduced earlier that all coastal areas have higher income. We also just concluded from maps 1 and 2 that coastal areas exhibit on average lower crime rates than their counterparts. Thus, we can see that since coastal areas had higher income, they showed lower criminal activities whereas other counties faced higher crimes.

We can infer then that consumers with higher income and wealth are usually concentrated along the coastal areas with lower crime rates.

Adding a New Dataset

We were able to retrieve two new datasets, one being the population dataset for the state of California for the years 1970 all the way to 2018 and the second dataset procured is the number of houses sold to various types of families in California for various years as well. Both these datasets will help us in constructing a more precise analysis of our research question. From our previous observations, particularly where we defined the relationship between average price levels and population density, we were faced with a caveat, that being we had to infer the values of population density using the variables **population** and **households**. Both these variables not only provided data at a block level for the state but also had limited observations. Using these two new datasets, we can now construct the same variables and relationship between them, but with a more enhanced and improved lens.

Step 1: Our very first step into making this new analysis is to clean the new datasets. Since both of these dataframes provided observations for a wide range of years, we can simply filter out the data consisting of the year 1990. The table below does exactly that and depicts a county wise distribution of population in absolute terms.

In [13]:
```
pop_df = pd.read_csv('Population_dtst_final.csv') # Loading the new population dataset
for row in pop_df.iterrows():
    index, column_values = row
    year = column_values['Year']
    if year != 1990:
```

```
        pop_df = pop_df.drop(index)
pop_df = pop_df.drop(['Year'], axis=1)
#pop_df = pop_df.set_index('County')
pop_df['County'] = pop_df['County'].str.strip()
pop_df.head(10)
```

Out[13]:

	County	Population
1160	Alameda	1274800.0
1161	Alpine	1100.0
1162	Amador	29610.0
1163	Butte	180420.0
1164	Calaveras	31540.0
1165	Colusa	16155.0
1166	Contra Costa	797600.0
1167	Del Norte	21660.0
1168	El Dorado	123900.0
1169	Fresno	661435.0

Step 2: Similiar to what we did for the first dataset, our next dataframe provides us with the number of houses sold to different types of families over a period of 10 years beginning 1991. The three main categories under which families are divided are: **Single**, **Multi** and **Mobile Homes**. So, we will need to filter out the data for the relevant year and for each county. Now, since we have not been provided data for the year 1990, we are simply going to assume for now that the observations provided for 1991 are pretty similiar to 1990 with some additional changes. Thus, for this dataset, we are going to pick the data for the year 1991 as shown below.

In [14]:
```
house_df = pd.read_csv('housing_family.csv') # Loading the new housing per family dataset
house_df = house_df.rename(columns={'Units': 'Houses'})
house_df
```

Out[14]:

	County	Year	Type	Houses
0	Alameda	1991	Single Family	301805.0
1	Alpine	1991	Single Family	881.0
2	Amador	1991	Single Family	10632.0
3	Butte	1991	Single Family	47276.0
4	Calaveras	1991	Single Family	16432.0
...	...	...	...	...
4867	Tulare	2018	Mobile Homes	10633.0
4868	Tuolumne	2018	Mobile Homes	3468.0
4869	Ventura	2018	Mobile Homes	11349.0
4870	Yolo	2018	Mobile Homes	3548.0
4871	Yuba	2018	Mobile Homes	2911.0

4872 rows × 4 columns

Step 3: For our next step, we have reshaped the above table in such a way that we have 4 main variables: **Single Family**, **Multi Family**, **County** and **Total Houses**. We have skipped the third category of family type **Mobile Homes** based on our assumption that thes families are constantly moving from one location to another. So, it is impossible to determine whether or not to count them in the population for that

particular year. For now, we are only going to be using the data provided for two types of families across various counties for the year 1991. Our Total Houses variable is the sum of the number of Single Family and Multi Family houses sold in each respective county for the year 1991.

In [15]:
```
housing_dict = {}
for row in house_df.iterrows():
    index, column_values = row
    county = column_values['County']
    year = column_values['Year']
    family_type = column_values['Type']
    houses = column_values['Houses']
    if year != 1991:
        house_df = house_df.drop(index)
    else:
        if county not in housing_dict and family_type == 'Single Family':
            housing_dict[county] = {'Single': houses, 'Multi': 0}
        elif county not in housing_dict and family_type == 'Multi-Family':
            housing_dict[county] = {'Single': 0, 'Multi': houses}
        elif county in housing_dict and family_type == 'Single Family':
            housing_dict[county]['Single'] += houses
        elif county in housing_dict and family_type == 'Multi-Family':
            housing_dict[county]['Multi'] += houses

house_final_df = pd.DataFrame(columns=['County', 'Single Family', 'Multi Family'])
index = 0
for x in housing_dict:
    county = x
    single = housing_dict[x]['Single']
    multi = housing_dict[x]['Multi']
    house_final_df.loc[index] = [county, single, multi]
    index += 1
#house_final_df = house_final_df.set_index('County')
house_final_df['County'] = house_final_df['County'].str.strip()
house_final_df['Total Houses'] = house_final_df['Single Family'] + house_final_df['Multi Family']
house_final_df.head(10)
```

Out[15]:

	County	Single Family	Multi Family	Total Houses
0	Alameda	301805.0	198944.0	500749.0
1	Alpine	881.0	393.0	1274.0
2	Amador	10632.0	1052.0	11684.0
3	Butte	47276.0	16498.0	63774.0
4	Calaveras	16432.0	1022.0	17454.0
5	Colusa	4825.0	795.0	5620.0
6	Contra Costa	231849.0	81070.0	312919.0
7	Del Norte	5442.0	1179.0	6621.0
8	El Dorado	49566.0	8342.0	57908.0
9	Fresno	157952.0	69976.0	227928.0

Step 4: Finally, after filtering, cleaning and reorganising our data, we recalled our old dataframe which we used in our previous analysis to build relationship between location and population density as well as average house value in each county vs the population density in that county. We merged that dataset with this new dataset as shown above to get the below table. Moreover, we have now calculated a revised population density given our new information which we have labelled as **new_pop_density**, which again has been calculated by dividing our new **population** variable with our new **Total Houses** amount.

In [16]:
```
# Creating a new dataset by merging the above two
pop_housing_df = pd.merge(pop_df, house_final_df, left_on='County', right_on='County')
pop_housing_df['new_pop_density'] = pop_housing_df['Population'] / pop_housing_df['Total Houses']
```

```
# Adding this new dataset into the older one

population_df = df3.loc[:, ['longitude', 'latitude', 'population', 'households']]
population_df['pop_density'] = population_df['population']/population_df['households']
population_df['coordinates'] = list(zip(population_df['longitude'], population_df['latitude']))
population_df['coordinates'] = population_df['coordinates'].apply(Point)


county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
CA_county = county_df.query("STATEFP == '06'")
CA_county.set_index('COUNTYFP')

# Converting into GeoDataFrame
city_gdf = gpd.GeoDataFrame(population_df, crs=4269, geometry='coordinates')
merged_gdf = gpd.sjoin(city_gdf, CA_county, op="within")
avg_pop_density = {}
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    pop_density = int(column_values['pop_density'])
    if county in avg_pop_density:
        avg_pop_density[county].append(pop_density)
    else:
        avg_pop_density[county] = [pop_density]
for x in avg_pop_density:
    avg_pop_density[x] = statistics.mean(avg_pop_density[x])

max_density = max(list(avg_pop_density.values()))
min_density = min(list(avg_pop_density.values()))

count = []
for row in merged_gdf.iterrows():
    index, column_values = row
    county = column_values['NAME']
    if county not in count:
        merged_gdf.at[index, 'avg_pop_density'] = avg_pop_density[county]
        count.append(county)
    else:
        merged_gdf = merged_gdf.drop(index)
merged_gdf = merged_gdf.set_index('COUNTYFP')
final_gdf = merged_gdf.join(gdf['avg_prices'])
final_gdf = final_gdf.reset_index()
final_merged_gdf4 = pd.merge(final_gdf, pop_housing_df, left_on='NAME', right_on='County')
final_merged_gdf5 = final_merged_gdf4.sort_values(ascending=True, by=['new_pop_density'])
final_merged_gdf6 = final_merged_gdf5.drop(['population', 'pop_density', 'households', 'ALAND', 'AWATER', 'AFFGEOID',
                                            'latitude', 'longitude', 'STATEFP', 'NAME', 'LSAD', 'COUNTYNS'], axis=1)
CA_county = CA_county.set_index('COUNTYFP')
final_merged_gdf6.head(10)
```

Out[16]:

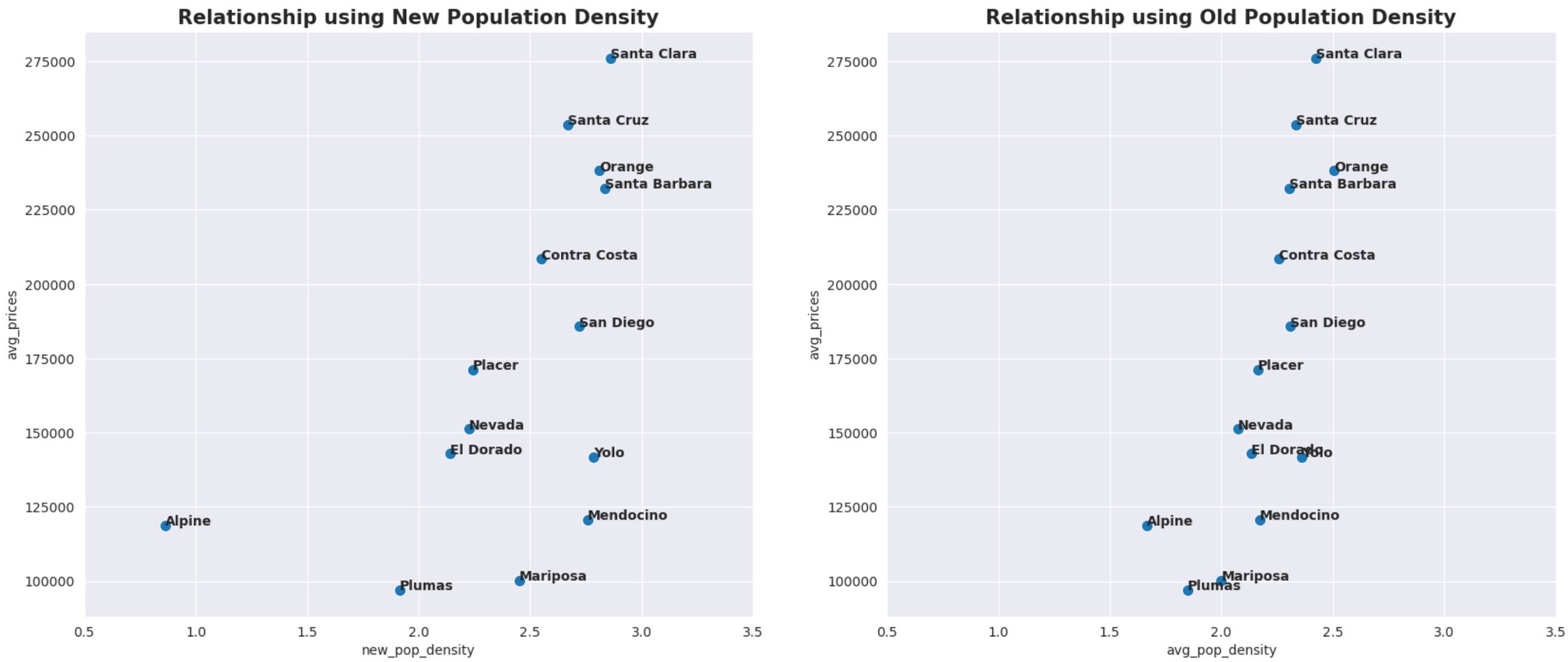
	COUNTYFP	coordinates	index_right	GEOID	avg_pop_density	avg_prices	County	Population	Single Family	Multi Family	Total Houses	new_pop_density
2	003	POINT (-119.78000 38.69000)	3181	06003	1.666667	118700.000000	Alpine	1100.0	881.0	393.0	1274.0	0.863422
30	051	POINT (-119.54000 38.51000)	1524	06051	2.058824	152129.411765	Mono	9750.0	4783.0	5206.0	9989.0	0.976074
48	091	POINT (-120.08000 39.61000)	84	06091	2.000000	77887.500000	Sierra	3280.0	1754.0	159.0	1913.0	1.714584
5	009	POINT (-120.46000 38.15000)	1021	06009	2.031250	107893.750000	Calaveras	31540.0	16432.0	1022.0	17454.0	1.807036
36	063	POINT (-120.98000 39.93000)	953	06063	1.848485	97109.090909	Plumas	19620.0	9183.0	1061.0	10244.0	1.915267
9	017	POINT (-119.95000 38.95000)	643	06017	2.133333	142900.840336	El Dorado	123900.0	49566.0	8342.0	57908.0	2.139601
40	075	POINT (-122.41000 37.81000)	2710	06075	2.042857	302908.913043	San Francisco	724100.0	105568.0	224901.0	330469.0	2.191128
55	109	POINT (-120.40000 38.00000)	2663	06109	2.210526	124328.070175	Tuolumne	47950.0	19668.0	2036.0	21704.0	2.209270
33	057	POINT (-121.07000 39.15000)	653	06057	2.072917	151272.916667	Nevada	77410.0	31469.0	3312.0	34781.0	2.225640
35	061	POINT (-120.10000 39.17000)	170	06061	2.165414	171257.142857	Placer	170110.0	63507.0	12303.0	75810.0	2.243899

Visualization

Visualizing the New Population Density & Average Price Level

Once we formulate and arrange all the data in a structured table, our last step is to visualize the data. The two graphs below represent the same relationship between the average house value and the population density in a select few counties, the only difference being the graph on the right uses value from our older dataframe whereas the graph on the left uses values from our new dataset. We can see quite a few difference between the two graphs, the first being its scatter. We notice that the scatter in our left graph is more spread out whereas the scatter on the right is more tightly bound or clustered. We also observe that the scatter on the left is rising with a gradual curve whereas in the same relationship on the left, we see the rise in scatter is quite steep. With this additional analysis, we can say now that there exists a weak yet positive relationship between average price of a house and the population density in a given county. However, it is inappropriate to infer a casual relationship from this data, in other words, just because there exists a positive relationship we cannot also say that higher population densities will cause price levels in counties to be higher or vice versa.

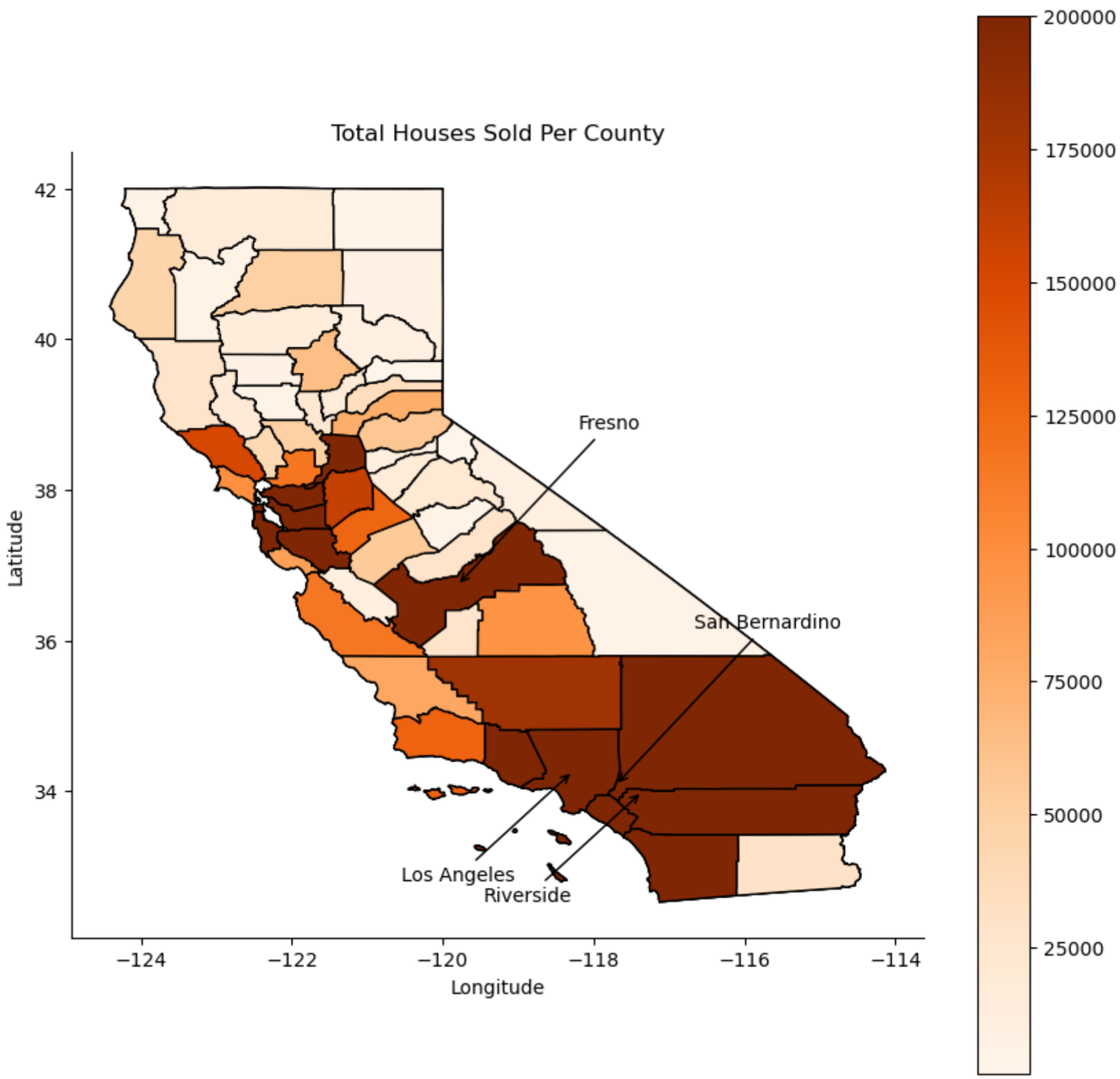
In [86]:
```
#final_merged_gdf = final_merged_gdf.set_index('COUNTYFP')
final_merged_gdf = final_merged_gdf.loc[['061', '057', '003', '063', '043', '017', '045', '083', '087', '085', '013', '073', '113', '059'], :]

fig, ax = plt.subplots(1, 2, figsize=(20, 8))
sns.scatterplot(data=final_merged_gdf, x="new_pop_density", y="avg_prices", ax=ax[0], s=77)
sns.scatterplot(data=final_gdf1, x="avg_pop_density", y="avg_prices", ax=ax[1], s=77)

for row in final_gdf1.iterrows():
    index, column_values = row
    county = column_values['NAME']
    old_pop_density = column_values['avg_pop_density']
    prices = column_values['avg_prices']
    ax[1].annotate(county, xy=(old_pop_density, prices), weight='bold')

for row in final_merged_gdf.iterrows():
    index, column_values = row
    county = column_values['County']
    new_pop_density = column_values['new_pop_density']
    prices = column_values['avg_prices']
    ax[0].annotate(county, xy=(new_pop_density, prices), weight='bold')

ax[0].set_xlim(0.5,3.5)
ax[1].set_xlim(0.5, 3.5)
ax[0].set_title('Relationship using New Population Density', fontsize=15, fontweight='bold')
ax[1].set_title('Relationship using Old Population Density', fontsize=15, fontweight='bold')
fig.suptitle('Relationship between Price & Population Density', fontsize=20, fontweight='bold')
plt.show()
```



Mapping Total Houses Bought

In [17]:
```
for row in final_merged_gdf4.iterrows():
    index, column_values = row
    countyfp = column_values['COUNTYFP']
    geometry = CA_county.loc[countyfp, 'geometry']
    final_merged_gdf4.at[index, 'geometry'] = geometry

final_merged_gdf4 = gpd.GeoDataFrame(final_merged_gdf4, geometry='geometry')

max_houses = max(list(final_merged_gdf4['Total Houses']))
min_houses = min(list(final_merged_gdf4['Total Houses']))

fig, gax = plt.subplots(figsize=(10,10))
state_df.query("NAME == 'California'").plot(ax=gax, edgecolor="black", color="white")
CA_county.plot(ax=gax, edgecolor="black", color="white")
final_merged_gdf4.plot(ax=gax, edgecolor='black', legend=True, column='Total Houses',
                       cmap='Oranges', vmin=1294, vmax=200000)

counties = ['San Bernardino', 'Riverside', 'San Diego', 'Los Angeles']

for x, y, label in zip(final_merged_gdf4['longitude'], final_merged_gdf4['latitude'], final_merged_gdf4['NAME']):
    if label == 'San Bernardino' or label == 'Fresno':
```

```
        gax.annotate(label, xy=(x,y), xytext=(80,80), textcoords='offset points', ha='center', va='bottom',
                     arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))
    elif label == 'Riverside' or label == 'Los Angeles':
        gax.annotate(label, xy=(x,y), xytext=(-60,-60), textcoords='offset points', ha='center', va='bottom',
                     arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0'))

gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')
gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)
plt.title('Total Houses Sold Per County')
plt.show()
```

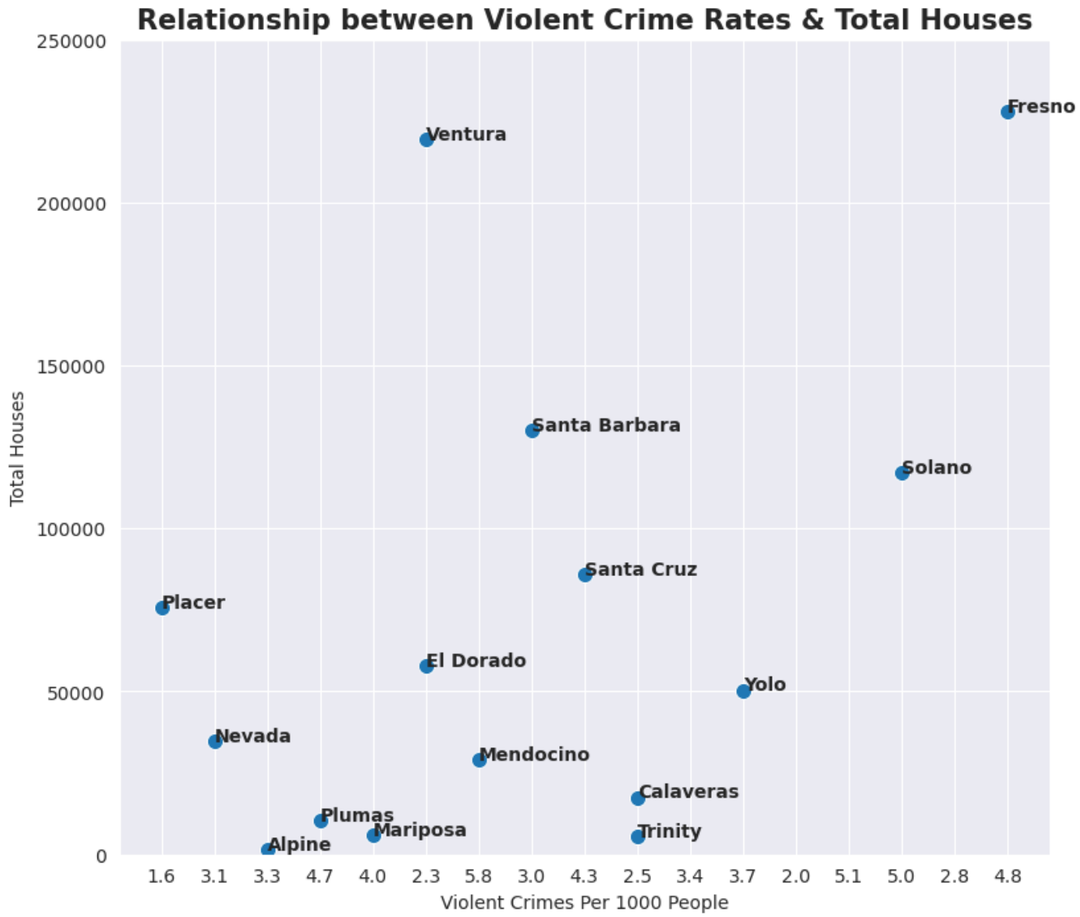


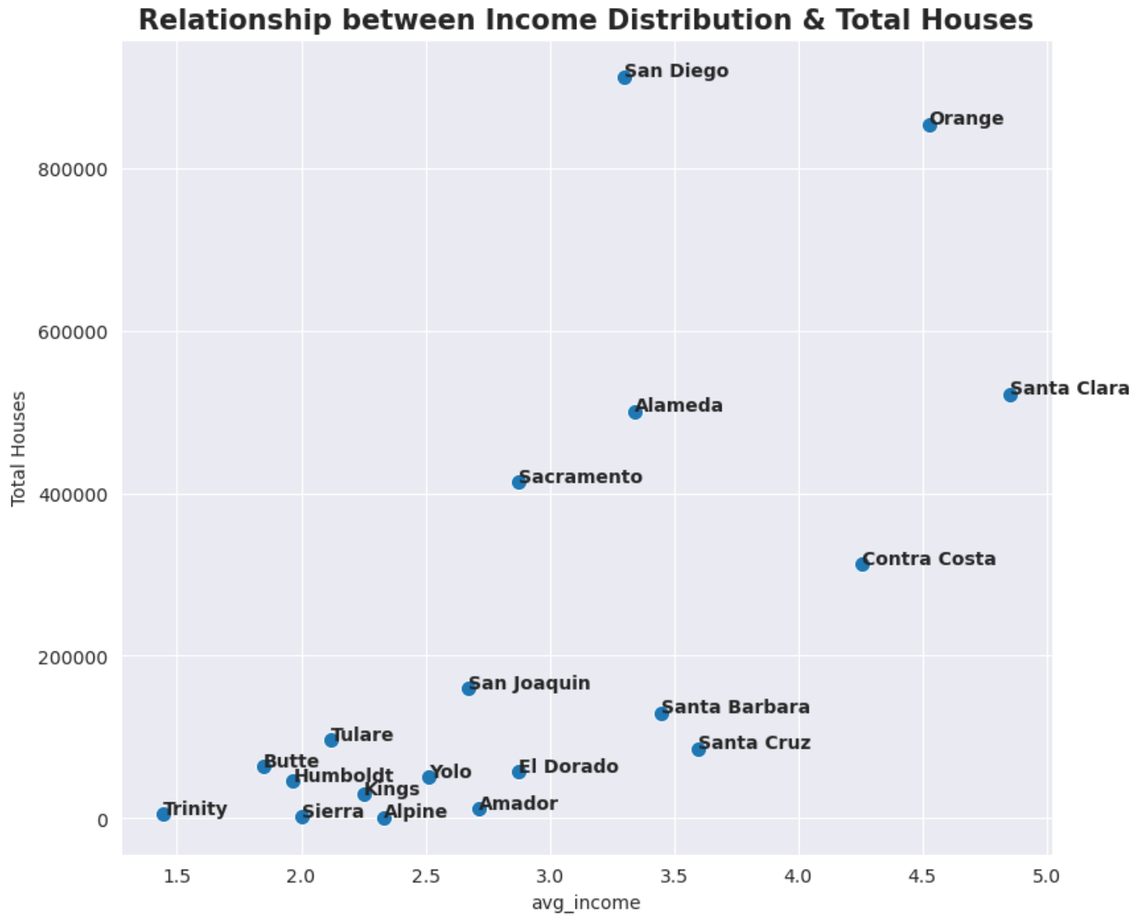
The map above displays the distribution of the total number of houses that were sold to families across the state of California in different counties. The variable we use in order to produce the above map is **Total Houses** which has been calculated by simply summing up the houses bought by the different kinds of families provided in the new dataset. For this analysis, we recognised two main categories: **Single** and **Multi Families**, adding them up gave us the final amount. We have used a color differentiating scheme to mark counties with higher number of houses purchased with a darker shade while those with lower purchases are shown in a lighter shade.

This analysis is crucial since it enables us to pinpoint the Demand of Houses across various counties and also allows us to use it to set up relationships with other variables. Using the map above, we can clearly see that there is a higher demand for houses in the Southern region of the state as well as along the coastal regions. There seems to be a relatively lower demand for houses Inland or in counties located up in the North as well.

Relationship between Total Houses Bought & Crime Rates

In [46]:
```
fig, gax = plt.subplots(1, 2, figsize=(20,8))

income_df = merged_gdf2.loc[:, ['NAME', 'avg_income']]
crimes_df = crime_rates.loc[:, ['County', 'Violent Crimes Per 1000 People', 'Property Crimes Per 1000 People']]
final_merged_gdf6 = pd.merge(final_merged_gdf6, crimes_df, left_on='County', right_on='County')
final_merged_gdf7 = pd.merge(final_merged_gdf6, income_df, left_on='County', right_on='NAME')
final_merged_gdf6 = final_merged_gdf6.sort_values(ascending=True, by=['Total Houses'])
final_merged_gdf7 = final_merged_gdf7.sort_values(ascending=True, by=['Total Houses'])

final_merged_gdf6 = final_merged_gdf6.set_index('COUNTYFP')
final_merged_gdf7 = final_merged_gdf7.set_index('COUNTYFP')

final_merged_gdf6 = final_merged_gdf6.loc[['061', '057', '003', '063', '043', '017', '045', '083',
                                           '087', '085', '013', '073', '113', '059', '067', '095', '105',
                                           '111', '009', '065', '019'], :]
final_merged_gdf7 = final_merged_gdf7.loc[['107', '003', '017', '083',
                                           '087', '085', '013', '073', '113', '059', '067', '105',
                                           '001', '005', '007', '091', '023', '077', '031'], :]

sns.scatterplot(data=final_merged_gdf6, x="Violent Crimes Per 1000 People", y="Total Houses", ax=gax[0], s=77)
sns.scatterplot(data=final_merged_gdf7, x="avg_income", y="Total Houses", ax=gax[1], s=77)

for row in final_merged_gdf6.iterrows():
    index, column_values = row
    county = column_values['County']
    houses = column_values['Total Houses']
    crimes = column_values['Violent Crimes Per 1000 People']
    gax[0].annotate(county, xy=(crimes, houses), weight='bold')

for row in final_merged_gdf7.iterrows():
    index, column_values = row
    county = column_values['County']
    houses = column_values['Total Houses']
    income = column_values['avg_income']
    gax[1].annotate(county, xy=(income, houses), weight='bold')

gax[0].set_ylim(0, 250000)
gax[0].set_title('Relationship between Violent Crime Rates & Total Houses', fontsize=15, fontweight='bold')
gax[1].set_title('Relationship between Income Distribution & Total Houses', fontsize=15, fontweight='bold')
sns.set_style('darkgrid')
plt.show()
```





If we compare the total number of houses bought by consumers in California with the amount of Violent crime present within each county, we cannot really see a significant relation between these variables in our first graph. It depicts a non-linear relationship with a dispersed scatter suggesting that we would require advanced logarithmic tools to derive a linear association between these two variables to carry out an extenive analysis. However, we can observe a slight trend that counties with a lower crime rate on average have a relatively higher demand for houses as compared to those with higher crime rates, after disregarding a few outliers. This indeed makes sense since consumers would prefer not to buy homes in crime-ridden areas and would rather choose the contrary.

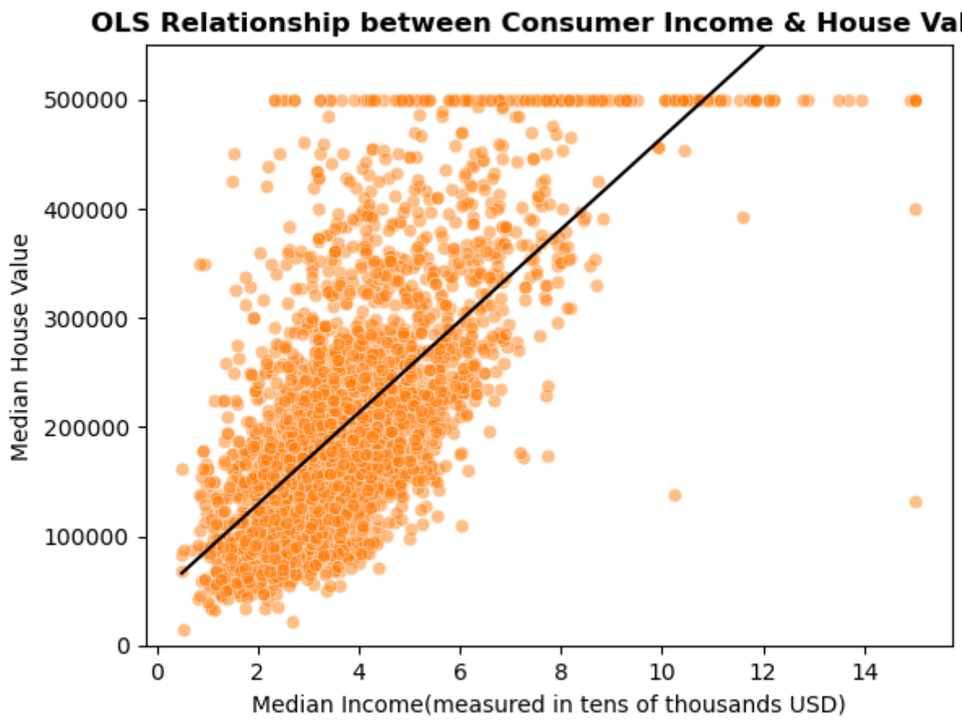
Similiarly, our second graph depicts a positive relationship between the Average Income Distribution and the Total Houses bought across counties. Intuitively, we can verify this, we all know that a house is a normal good for most consumers and more specifically a luxury good. Hence, as the income of a consumer rises, so does their demand for such goods. In other words, their demand for a normal good is positively correlated with their personal income. For this same reason, we are ultimately able to see a positive realtionship in the graph above as well because as income of consumers increases, the number of purchases made also rises.

Regression

At the very beginning, our main research analysis begged the question what are the various factors that might affect a consumer's decision into buying a house in the state of California. We initially defined many explanatory variables that can affect the prices of the house and thus influence the decision of consumers such as **Income**, **Age of the House**, **Ocean Proximity**, and **Population Density**. One of the crucial factors that consumers often integrate into their purchasing decisions is their **Disposable Income**, or the amount of money left after paying all their taxes. We have done a comprehensive visual and theoretical analysis above regarding the relationship between income and the price of a house using the help of a scatterplot. With this graphical analysis, we concluded that there indeed exists a strong linear as well as positive relationship between the two since we can also observe the scatter transitioning upwards, rising from the bottom left to the upper right with the scatter points clustered tightly. We can also think about it intuitively from the perspective of a consumer, homes are normal goods for such consumers, that is, as the income of people increase their demand for houses will increase proportionally. Since the supply of houses is limited and not flexible, there will be excessive demand causing a possible shortage of houses thereby increasing the average prices.

In order to carry out regressions, the variables we will be using are **Income**, **Age of the House**, **Ocean Proximity**, and **Population Density**. As explained above, each of these play an important role in affecting the house value. Just like we examined **Income** as a potential regression variable above, we can take **Age of the House** as an another regressor which fits within our model. Empirically, we verified earlier that age of the house do not have some sort of correlation with its market value but simply because there is no correlation doesn't mean we reject the variable. We did learn that consumers pay attention to the age while buying a home since a majority of them bought houses over a range of ages. Thus, regressing house value on age might be useful. Similiarly, with **Population Density** we saw that there was a strong positive correlation between population densities of different counties along prices of houses. Since the number of family members residing in a house dictates how big a house the family would purchase, it would significantly raise the price of houses they look for. Thus, it would be a good predictor of house value.

In [6]:
```
random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=3000)
new_df = df3.iloc[sample]
X = new_df['median_income']
y = new_df['median_house_value']

fig, ax = plt.subplots()
ax.scatter(X, y, marker='')

sns.scatterplot(data=new_df, x="median_income", y="median_house_value", alpha=0.5, ax=ax)
ax.plot(np.unique(X),
         np.poly1d(np.polyfit(X, y, 1))(np.unique(X)),
         color='black')

ax.set_ylim([0,550000])
plt.xlabel('Median Income(measured in tens of thousands USD)')
plt.ylabel('Median House Value')
plt.title('OLS Relationship between Consumer Income & House Value', fontweight='bold')
plt.show()
```



Regression on Income and Ocean Proximity

In our first regression, we are going to be regressing 2 explanatory variables **Income** and **Ocean Proximity** on the dependent variables **Average House Value**. The table below shows the regression results after fitting all these variables into our model. Our multivariate regression equation or model can be written down the following way:

$$HouseValue_i = \beta_0 + \beta_1 MedianIncome_i + \beta_2 OceanProx_i + u_i$$

where:

- β_0 is the intercept of the linear trend line on the y-axis
- β_1 is the first slope representing the *marginal effect* of a 10,000 USD increase in Median Income on the Average House Value after controlling for Ocean Proximity
- β_2 is the second slope of the dummy variable OceanProx which can take values from 1-5. It represents the marginal effect of a change in Ocean Proximity on the Average House Value after controlling for Median Income
- u_i is a random error term (deviations of observations from the trend line due to factors not included in the model)

As we explained earlier, Income is one of the many crucial factors that consumers consider before making a decision to purchase a house. Thus, the Income or Budget of a consumer directly affects the average price that he is ready to pay for a home. Since we are trying to establish a casual relationship between these two variables, it is important to run the regression above. The next variable Ocean Proximity also plays a significant role since Income isn't uniform throughout the state, depending upon the ocean proximity the income of consumers will vary as we have already established earlier that consumers situated nearer the coast reportedly have higher income. So, higher incomes can result in consumers purchasing expensive houses which is why it is important to include this variable as well.

We can now interpret the results of the regression table provided below. Each of the numerical values provided alongside the explanatory variables Income and Ocean Proximity in the first half of the table represent the slope coefficients β_1 and β_2. So, as described above, β_1 below shows that the effect of a 10,000 USD increase in income of a consumer will be a 38,000 USD rise in the price of the house after controlling for Ocean Proximity.

Now, when we try to interpret β_2, we need to remember that it is the slope coefficient of the dummy variable Ocean Proximity. So, it can be interpreted as we move towards the coast or ocean by increasing the value of this variable, the average house value increases by approximately 18,000 USD after controlling for Median Income. This is quite intuitive since we discussed above the relationship between house value and ocean proximity. We concluded that houses situated near the coast are more expensive compared to the houses located Inland. Lastly, the slope intercept represented by the value given alongside **const** cannot be interpreted since to interpret it would mean to assume all **x variables** as 0. We possibly cannot have a family or consumer that has 0 Median Income thus it would be impractical to set them as 0.

Also, we can observe that all the above values are significant, that is, they are both economically and statistically significant at a conventional 99% confidence level. The R^2 signifies that 55% of the proportion of variation in the price of houses are explained by the variation in Median Income and Ocean Proximity. To test whether the overall model is significant, R^2 is not the sole tool to rely on. Even though it is 55%, we need to consider the F test statistic which represents the overall statistical significance of the model. We can see that the F test which is a big value is highly statistically significant at a 99% confidence level since the P-value lies in the interval 0-0.01. This means that this model does help in providing an effective prediction of the Price of house given a certain Income of a consumer and the distance of the house from the ocean.

In [11]:
```
# Add constant term to our original dataset
new_df['const'] = 1

# Create lists of variables to be used in each regression
X1 = ['const', 'median_income', 'ocean_proximity']

# Estimate an OLS regression for each set of variables
reg1 = sm.OLS(new_df['median_house_value'], new_df[X1], missing='drop').fit()

stargazer = Stargazer([reg1])
stargazer.custom_columns(["House Value"],[1])
HTML(stargazer.render_html())
```

Out[11]:

	Dependent variable:median_house_value
	House Value
	(1)
const	-756.330
	(3822.802)
median_income	38084.756***
	(759.582)
ocean_proximity	17986.704***
	(830.845)
Observations	3,000
R^2	0.553
Adjusted R^2	0.553
Residual Std. Error	76575.778 (df=2997)
F Statistic	1853.217*** (df=2; 2997)
Note:	*p<0.1; **p<0.05; ***p<0.01

Regression on Income & Population Density

In our second regression, we are going to be regressing 2 explanatory variables **Median Income** and **Population Density** on the dependent variables **Average House Value**. The table below shows the regression results after fitting all these variables into our model. Our multivariate regression equation or model can be written down the following way:

$$HouseValue_i = \beta_0 + \beta_1 MedianIncome_i + \beta_2 PopDensity_i + u_i$$

where:

- β_0 is the intercept of the linear trend line on the y-axis
- β_1 is the first slope representing the *marginal effect* of a 10,000 USD increase in Median Income on the Average House Value after controlling for Population Density
- β_2 is the second slope representing the *marginal effect* of an additional family member living in the house on the Average House Value after controlling for Median Income
- u_i is a random error term (deviations of observations from the trend line due to factors not included in the model)

Population Density and Income for a given family are two variables which go hand in hand while making a decision about purchasing a house. We said earlier that Population Density was simply the number of family members that would live in a single home. Depending upon the density, the size of the house would vary a lot suggesting that there could be a change in prices. So, we should expect the prices to rise as the family gets bigger however we observed the contrary while making the same analysis above. There was a negative correlation between prices and population density, seemingly because there might be some families that despite being big in size are income constrained or in other words do not have the ability to afford bigger and expensive houses. Thus, a change in population density woud not suffice in understanding the change on house prices which is why it we need to regress income and population density together on house value.

In the regression table below, the values written alongside the explanatory variables Income and Population Density represent their slope coefficients β_1 and β_2. β_1 can be interpreted below the same way we did earlier, which is the average effect of a 10,000 USD increase in income of a consumer will be a 42,000 USD rise in the price of the house after controlling for population density. Similiarly, when we interpret β_2 we can say that on average the price of a house decreases by approximately 310 USD after an additional family member is added to the household by controlling for Median Income. The intercept cannot be interpreted since to do that would mean substituting all **x variables** with 0. We cannot possibly assume that any household or consumers will have a 0 income or that any house will have no family members residing within it.

Also, we can observe that all the above variables are statistically significant at a conventional 99% confidence level. However, only Median Income is economically significant of the two since its point estimate is significantly higher than population density. The R^2 signifies that about 49% of the proportion of variation in the price of houses are explained by the variation in Median Income and Population Density. To test whether the overall model is significant, we need to consult the F test statistic as well. We can see that the F test which is even greater than a 1000 is highly statistically significant at a 99% confidence level implying that the P-value lies in the interval 0-0.01. This means that this model does help in providing an effective prediction of the Price of house given a certain Income and population density of a household.

In [13]:
```
# Add constant term to our original dataset
new_df['const'] = 1
new_df['pop_density'] = new_df['population']/new_df['households']


# Create lists of variables to be used in each regression
X2 = ['const', 'median_income', 'pop_density']

# Estimate an OLS regression for each set of variables
reg2 = sm.OLS(new_df['median_house_value'], new_df[X2], missing='drop').fit()

stargazer = Stargazer([reg2])
stargazer.custom_columns(["House Value"],[1])
HTML(stargazer.render_html())
```

Out[13]:

	Dependent variable:median_house_value
	House Value
	(1)
const	45353.168***
	(3407.901)
median_income	42204.274***
	(791.963)
pop_density	-310.095***
	(66.220)
Observations	3,000
R^2	0.487
Adjusted R^2	0.486
Residual Std. Error	82046.164 (df=2997)
F Statistic	1421.169*** (df=2; 2997)
Note:	*p<0.1; **p<0.05; ***p<0.01

Regression on Median Income and Age of the House

In our third regression, we are going to be regressing 2 explanatory variables **Age of the House** and **Median Income** on the dependent variables **Average House Value**. The table below shows the regression results after fitting all these variables into our model. Our multivariate regression equation or model can be written down the following way:

$$HouseValue_i = \beta_0 + \beta_1 Age_i + \beta_2 MedianIncome_i + u_i$$

where:

- β_0 is the intercept of the trend line on the y-axis
- β_1 is the first slope representing the *marginal effect* of an increase in the age of the house by 1 year on the Average House Value after controlling for Median Income
- β_2 is the second slope representing the *marginal effect* of a 10,000 USD increase in Median Income on the Average House Value after controlling for Age of the House
- u_i is a random error term (deviations of observations from the trend line due to factors not included in the model)

This regression model is another useful tool in understanding the behaviour of price level of houses using age and Median Income as explanatory variables. Even though we determined earlier that age not necessarily correlates with the average house value at a particular location, it does not give us much evidence to simply reject this variable. Also, we observed earlier in our graphical analysis that a wide range of houses with different ages are priced differently. For instance, a house older than 20 years might be priced a bit on the higher side since its classical medieval looking architecture will be valued more by consumers.

In the regression table below, as usual, the values written alongside the explanatory variables Age and Median Income represent their slope coefficients β_1 and β_2. β_1 can interpreted as that on average as the age of the house becomes older by each year, the price of the house increases by about 1600 USD after controlling for Median Income. This does make sense since older houses which are available for sale have intact structures, strong integrity along with deep historical values making it attractive to various consumers. Thus, older houses can be considered to have a higher demand making them more expensive. Similiarly, we can interpret β_2 as the marginal effect of a 10,000 USD rise in the income of a consumer is the price of the house increasing by approximately 43,000 USD.

The slope intercept stated alongside **const** cannot be interpreted as in order do so would mean substitute all **x variables** with the value 0. It is impossible to imagine any consumer not earning any income and wanting to purchase a house which is why it is would improbable to make an interpretation. All of the variables stated above are statistically significant at a conventional 99% confidence level implying that the T test statistic must be greater than 2. However, only Median Income is economically significant and not Age of the house. For economic significance, we usually notice the point estimate or the slope coefficient, if we observe the same for Age, an increase of 1600 USD relative to the slope of Median Income is miniscule and so has no economic importance. Thus, Age of the house is only statistically significant whereas Median Income is significant as it is both economically and statistically important.

R^2 stands at an impressive 51% describing the fact that about 51% of the proportion of variation in the price of houses is explained by the variation in Age of the house and Median Income. The F test on the same hand produces an overall statistical significance at a 99% confidence level with a value of 1583. A high F test statistic suggests that the Regression value is also statistically significant at a 0.1% significance level and further indicates that the following model fits our data accurately and will enable us in predicting the average house value effectively.

In [14]:
```
# Add constant term to our original dataset
new_df['const'] = 1

# Create lists of variables to be used in each regression
X3 = ['const', 'housing_median_age', 'median_income']

# Estimate an OLS regression for each set of variables
reg3 = sm.OLS(new_df['median_house_value'], new_df[X3], missing='drop').fit()

stargazer = Stargazer([reg3])
stargazer.custom_columns(["House Value"], [1])
HTML(stargazer.render_html())
```

Out[14]:

	Dependent variable:median_house_value
	House Value
	(1)
const	-5081.303
	(4930.661)
housing_median_age	1594.765***
	(115.806)
median_income	43187.269***
	(774.436)
Observations	3,000
R^2	0.514
Adjusted R^2	0.513
Residual Std. Error	79857.957 (df=2997)
F Statistic	1583.366*** (df=2; 2997)
Note:	*p<0.1; **p<0.05; ***p<0.01

Regression on Population Density, Income and Age of the House

In our next regression, we are going to be regressing 3 explanatory variables **Population Density**, **Median Income** and **Age of the House** on the dependent variables **Average House Value**. The table below shows the regression results after fitting all these variables into our model. Our multivariate regression equation or model can be written down the following way:

$$HouseValue_i = \beta_0 + \beta_1 PopDensity_i + \beta_2 MedianIncome_i + \beta_3 Age_i + u_i$$

where:

- β_0 is the intercept of the linear trend line on the y-axis
- β_1 is the first slope representing the *marginal effect* of an additional family member living in the house on the Average House Value after controlling for Income and Age of the House
- β_2 is the second slope representing the *marginal effect* of a 10,000 USD increase in Median Income on the Average House Value after controlling for Age of the House and Population Density
- β_3 is the third slope representing the *marginal effect* of an increase in the age of the house by 1 year on the Average House Value after controlling for Income and Population Density
- u_i is a random error term (deviations of observations from the trend line due to factors not included in the model)

We have talked extensively about each of the variables listed above in the 3 regression models that we have already completed. Using this regression analysis, we will try our level best to combine these variables together and examine whether they do have any meaningful impact in effectively determining the average house value across the state of California. We use all three of these variables simply because consumers within a certain income bracket with a given family size determined by the Population Density use Age of the House while making purchasing decisions which influences the demand of these houses as well as there prices.

From the Regression results below, the values written alongside the explanatory variables Age, Median Income and Population Density represent their slope coefficients β_1, β_2 and β_3. β_1 as we can notice has a negative value next to Population Density which might be interpreted as with each additional family member introduced in the house, the price of the house will decrease by approximately 337 USD after controlling for Income and Age of the House. Now, intuitively one might think that more family members in a house translate to more rooms which means that the family would require a bigger house to accomodate more people causing the price to increase. However, we must remember that since we are in a multiple regression model, each of these variables do not independently influence the average house value. So, even though the table below states that there exists a negative relationship between price and population density, we cannot take this result to word and would require a correlation matrix to further enhance our analysis.

We can also assume that there might be other factors affecting the relationship between these two variables such as Income. It might be possible that we are provided an income constrained family who has a strict budget to follow in which case having a bigger family doesn't translate to more rooms in the house. Similiarly, another factor that can have a significant effect is the room size, it is entirely possible that the average size of a room is bigger such that it is able to accomodate all family members without necessarily increasing the size of the house thus resulting in lower price levels.

The value of β_2 can be interpreted the same way as we did before, i.e, with a 10,000 USD increase in income of a given consumer or family, the price of the house will increase on average by roughly 43000 USD after controlling for Age of the House and Population Density which is quite intuitive again since house is a luxury good with a higher income elasticity, which means that as income rise the ability or demand of a consumer to buy more expensive goods increase as well. Similiarly, β_3 can be interpreted as on average as the age of the house becomes older by each year, the price of the house increases by about 1600 USD after controlling for Median Income and Population Density.

The slope intercept stated alongside **const** cannot be interpreted as in order do so would mean substitute all **x variables** with the value 0. It is impossible to imagine any consumer not earning any income and wanting to purchase a house or a family having a 0 Population Density which is why it would be improbable to make an interpretation. All of the variables stated above are statistically significant at a conventional 99% confidence level implying that the T test statistic must be greater than 2. However, the economic signifance test only stands true for Median Income and not for Population Density or Age of the House. For a variable to be economically significant, we usually notice the point estimate or the slope coefficient, if we observe the same for Age or population density, an increase of 1600 USD relative to the slope of Median Income is miniscule and so has no economic importance. Thus, Age of the house and Population Density are only statistically significant whereas Median Income is significant as it is both economically and statistically important.

R^2 stands at an impressive 52% describing the fact that about 52% of the proportion of variation in the price of houses is explained by the variation in Age of the house, Population Density and Median Income. The F test on the same hand produces an overall statistical significance at a 99% confidence level with a value of 1074. A high F test statistic suggests that the Regression value is also statistically significant at a 0.1% significance level and further indicates that the following model fits our data accurately and will enable us in predicting the average house value effectively.

In [8]:
```
# Add constant term to our original dataset
new_df['const'] = 1
new_df['pop_density'] = new_df['population']/new_df['households']
```

```
# Create lists of variables to be used in each regression
X4 = ['const', 'pop_density', 'median_income', 'housing_median_age']

# Estimate an OLS regression for each set of variables
reg4 = sm.OLS(new_df['median_house_value'], new_df[X4], missing='drop').fit()

stargazer = Stargazer([reg4])
stargazer.custom_columns(["House Value"],[1])
HTML(stargazer.render_html())
```

Out[8]:

	Dependent variable:median_house_value
	House Value
	(1)
const	-5465.203
	(4909.370)
housing_median_age	1613.630***
	(115.349)
median_income	43438.532***
	(772.483)
pop_density	-337.994***
	(64.199)
Observations	3,000
R^2	0.518
Adjusted R^2	0.518
Residual Std. Error	79504.360 (df=2996)
F Statistic	1074.227*** (df=3; 2996)
Note:	*p<0.1; **p<0.05; ***p<0.01

Regression on Population Density, Income and Ocean Proximity

In our final regression, we are going to be regressing 3 explanatory variables **Population Density**, **Median Income** and **Ocean Proximity** on the dependent variables **Average House Value**. The table below shows the regression results after fitting all these variables into our model. Our multivariate regression equation or model can be written down the following way:

$$HouseValue_i = \beta_0 + \beta_1 PopDensity_i + \beta_2 MedianIncome_i + \beta_3 OceanProx_i + u_i$$

where:

- β_0 is the intercept of the linear trend line on the y-axis
- β_1 is the first slope representing the *marginal effect* of an additional family member living in the house on the Average House Value after controlling for Income and Age of the House
- β_2 is the second slope representing the *marginal effect* of a 10,000 USD increase in Median Income on the Average House Value after controlling for Age of the House and Population Density
- β_3 is the third slope of the dummy variable *OceanProx* which can take value from 1-5. It represents the *marginal effect* of a change in Ocean Proximity on the Average House Value after controlling for Income and Population Density
- u_i is a random error term (deviations of observations from the trend line due to factors not included in the model)

As we stated when we began our analysis, ocean proximity is also another crucial variable that distinguishes houses based on price level. As we move towards the ocean or the coast, we are most likely to observe the house value rise steeply. Thus, regressing ocean proximity along with Median Income and Population Density on the average price level of houses seems like an effective model to incorporate in our analsysis as it might help in providing an effective estimate of prices.

In the regression table below, the values written alongside the explanatory variables Population Density, Median Income and Ocean Proximity represent their slope coefficients β_1, β_2 and β_3. β_1 as we can notice has a negative value next to Population Density which might be interpreted as with each additional family member introduced in the house, the price of the house will decrease by approximately 258 USD after controlling for Income and Ocean Proximity. As we explained earlier, intuitively one might think that more family members in a house translate to more rooms which means that the family would require a bigger house to accomodate more people causing the price to increase. However, we know that since we are in a multiple regression model, each of these variables do not independently influence the average house value. So, even though the table below states that there exists a negative relationship between price and population density, we cannot take this result to word and would require a correlation matrix to further enhance our analysis.

The value of β_2 can be interpreted the same way as we did before, i.e, with a 10,000 USD increase in income of a given consumer or family, the price of the house will increase on average by roughly 38000 USD after controlling for Ocean Proximity and Population Density which is quite intuitive again since house is a luxury good with a higher income elasticity, which means that as income rise the ability or demand of a consumer to buy more expensive goods increase as well. β_3 which is the coefficient of the dummy variable Ocean Proximity can be interpreted as we move closer to ocean/coast by increasing the value of the dummy variable, the price of the house increases by approximately 18,000 USD after controlling for Income and Population Density. This makes intuitive sense since houses on the coast are more expensive to the ones located inland.

The slope intercept stated alongside **const** cannot be interpreted as in order do so would mean substitute all **x variables** with the value 0. However, it is impossible to imagine any consumer not earning any income and wanting to purchase a house or a family having a 0 Population Density which is why it would be improbable to make an interpretation. All of the variables stated above are statistically significant at a conventional 99% confidence level implying that the T test statistic must be greater than 2. However, Population Density fails to be economically significant yet again with a tiny point estimate. Thus, Population Density is only statistically significant whereas Median Income and Ocean Proximity are significant in value as they are both economically and statistically important.

R^2 stands at an impressive 57% describing the fact that about 57% of the proportion of variation in the price of houses is explained by the variation in Ocean Proximity, Population Density and Median Income. The F test on the same hand produces an overall statistical significance at a 99% confidence level with a value of 1248. A high F test statistic suggests that the Regression value is also statistically significant at a 0.1% significance level and further indicates that the following model fits our data accurately and will enable us in predicting the average house value effectively.

In [9]:
```
# Add constant term to our original dataset
new_df['const'] = 1
new_df['pop_density'] = new_df['population']/new_df['households']

# Create lists of variables to be used in each regression
X5 = ['const', 'ocean_proximity', 'median_income', 'pop_density']

# Estimate an OLS regression for each set of variables
reg5 = sm.OLS(new_df['median_house_value'], new_df[X5], missing='drop').fit()

stargazer = Stargazer([reg5])
stargazer.custom_columns(["House Value"],[1])
HTML(stargazer.render_html())
```

Out[9]:

	Dependent variable:median_house_value
	House Value
	(1)
const	-249.959
	(3814.212)
median_income	38295.314***
	(759.161)
ocean_proximity	17851.345***
	(829.192)
pop_density	-258.338***
	(61.682)
Observations	3,000
R^2	0.556
Adjusted R^2	0.555
Residual Std. Error	76365.325 (df=2996)
F Statistic	1248.144*** (df=3; 2996)
Note:	*p<0.1; **p<0.05; ***p<0.01

Combined Table for Regression Results

In [15]:
```
stargazer = Stargazer([reg1, reg2, reg3, reg4, reg5])
HTML(stargazer.render_html())
```

Out[15]:

					Dependent variable:median_house_value
	(1)	(2)	(3)	(4)	(5)
const	-756.330	45353.168***	-5081.303	-5465.203	-249.959
	(3822.802)	(3407.901)	(4930.661)	(4909.370)	(3814.212)
housing_median_age			1594.765***	1613.630***	
			(115.806)	(115.349)	
median_income	38084.756***	42204.274***	43187.269***	43438.532***	38295.314***
	(759.582)	(791.963)	(774.436)	(772.483)	(759.161)
ocean_proximity	17986.704***				17851.345***
	(830.845)				(829.192)
pop_density		-310.095***		-337.994***	-258.338***
		(66.220)		(64.199)	(61.682)
Observations	3,000	3,000	3,000	3,000	3,000
R^2	0.553	0.487	0.514	0.518	0.556
Adjusted R^2	0.553	0.486	0.513	0.518	0.555
Residual Std. Error	76575.778 (df=2997)	82046.164 (df=2997)	79857.957 (df=2997)	79504.360 (df=2996)	76365.325 (df=2996)
F Statistic	1853.217*** (df=2; 2997)	1421.169*** (df=2; 2997)	1583.366*** (df=2; 2997)	1074.227*** (df=3; 2996)	1248.144*** (df=3; 2996)
Note:					*p<0.1; **p<0.05; ***p<0.01

Machine Learning

Now that we have created various Regression Models using different variables and factors to enhance our analysis, a more effective tool used sometimes to verify our regression results is a Regression Tree. The goal of a regression analysis is to provide an accurate description of how a change in one of the exogenous variables will affect the endogenous variables. Exogenous variables simply mean factors that are readily available and are extracted from outside the regression model. Endogenous variables on the other hand are factors which are not easily available and so as a model maker we would like to analyse these components by using other factors at hand such as the exogenous variables. Till now, the endogenous variable that we have been trying to analyse and examine is the Average House Value in the state of California using various exogenous variables such as Median Income, Ocean Proximity, Age of the House and Population Density. Before we begin creating our regression tree, let's construct an objective function by using our last regression model which stated:

$$HouseValue_i = \beta_0 + \beta_1 PopDensity_i + \beta_2 MedianIncome_i + \beta_3 OceanProx_i + u_i$$

Thus, the objective function can be written as:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\text{HouseValue}_i - (\beta_0 + \beta_1 \text{PopDensity}_i + \beta_2 \text{MedianIncome}_i + \beta_3 \text{OceanProx}_i)\right)^2 + \alpha|\text{T}|$$

So, in the above equation β_1, β_2, and β_3 are parameters or weights whose values our Machine Learning algorithm is tasked with finding such that it would best describe the data given in our original dataset. In our previous analysis, we usually noticed that when we graphed out any these variables, the scatter would be highly dispersed and not be very close to the trend line. In other words, the residual or difference between the expected value and actual value was quite high. In order to minimise this difference across all output values which we call as the *Mean Square Error*, we use the algorithm above to get as linear or straight a line as possible such that it passes through the majority of the scatter points.

Another important component of this objective function is α or the **tree tuning paramater**. This is responsible for controlling the trade-off between the complexity of the regression tree and the quality it produces. Often times, as we try to predict a certain component of our analysis, we make complex assumptions and create difficult structures which make it impossible to examine or conclude any result from the model. In other words, we hamper with the quality of the analysis rendering it ineffectual. The paramater α is responsible for ensuring that the model is complex enough to not destroy the quality of the analysis or make it incomprehensible for the audience. A higher value of α suggests a stricter control and penalty over the complexity of the regression tree.

Now, the first step into creating a regression tree is to input our X and y variables. Our X variable will be nothing but the dataset including only the factors that we used in our algorithm, i.e, Median Income, Ocean Proximity and the Population Density. The table below shows the same:

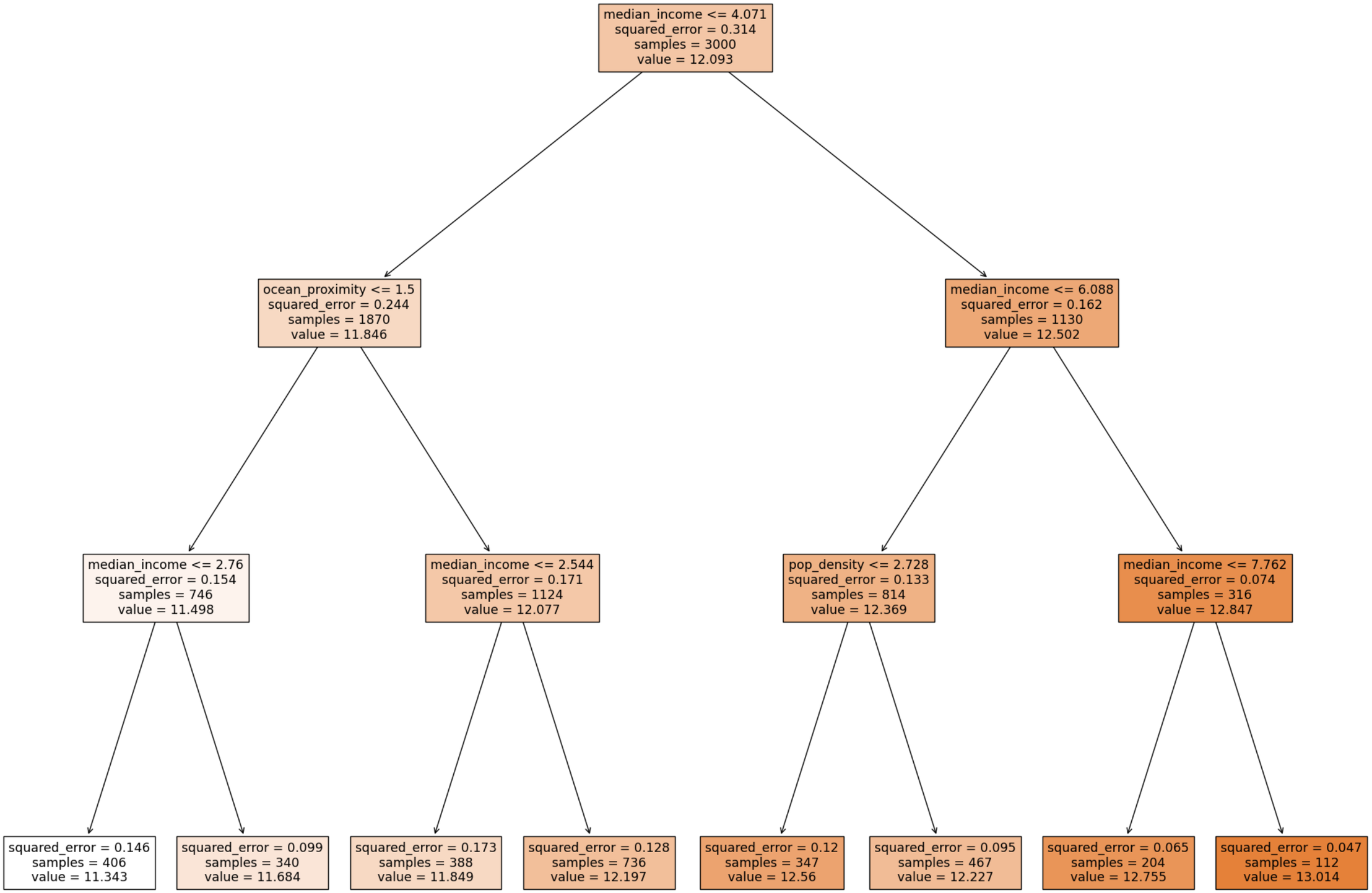
In [22]:
```
random_num = np.random.RandomState(0)
sample = random_num.choice(np.arange(df3.shape[0]), size=3000)
tree_df = df3.iloc[sample]

X = tree_df.loc[:, ['median_income', 'ocean_proximity', 'population', 'households']]
X['pop_density'] = X['population']/X['households']
X = X.drop(['population', 'households'], axis=1)
X.head()
```

Out[22]:

	median_income	ocean_proximity	pop_density
2732	1.3882	1	3.436242
10799	5.6315	5	2.247863
9845	3.1977	4	1.791349
19648	2.9135	1	3.095694
13123	4.4125	1	3.251142

The next step involves formulating our y variable which is the endogenous variable that we are trying to model from one of our regression analysis, the Average House Value. Once we compute both our X and y components, we simply fit both these variables into our tree function and the figure produced below is the Regression Tree.

In [23]:
```
y = np.log(tree_df["median_house_value"])
sqft_tree = tree.DecisionTreeRegressor(max_depth=3).fit(X,y)
sqrf_fig = plt.figure(figsize=(25,20))
sqrf_fig = tree.plot_tree(sqft_tree, feature_names=X.columns, filled=True)
```


median_income <= 4.071
squared_error = 0.314
samples = 3000
value = 12.093
ocean_proximity <= 1.5
squared_error = 0.244
samples = 1870
value = 11.846
median_income <= 6.088
squared_error = 0.162
samples = 1130
value = 12.502
median_income <= 2.76
squared_error = 0.154
samples = 746
value = 11.498
median_income <= 2.544
squared_error = 0.171
samples = 1124
value = 12.077
pop_density <= 2.728
squared_error = 0.133
samples = 814
value = 12.369
median_income <= 7.762
squared_error = 0.074
samples = 316
value = 12.847
squared_error = 0.146
samples = 406
value = 11.343
squared_error = 0.099
samples = 340
value = 11.684
squared_error = 0.173
samples = 388
value = 11.849
squared_error = 0.128
samples = 736
value = 12.197
squared_error = 0.12
samples = 347
value = 12.56
squared_error = 0.095
samples = 467
value = 12.227
squared_error = 0.065
samples = 204
value = 12.755
squared_error = 0.047
samples = 112
value = 13.014

We can observe from our regression tree above that we have created an 8 branch tree with a height or depth of 3. Our root is the Median Income stated at the top of the tree, with several internal and terminal nodes. So, we have used the California Housing Data and created a regression tree to predict the average house value of homes across the entire state, based on a given consumer's Income, the proximity of the house from the ocean/coast as well as the Population Density, i.e, number of family members living within the household. At a given internal node, the label $\mathrm{X}_i < \mathrm{t}_k$ indicates the left branch emanating from the split and the label $\mathrm{X}_i > \mathrm{t}_k$ corresponds to the right hand split. So, if we look at the root, it splits the tree into two main branches. The left hand branch corresponds to the MedianIncome <= 4.071, or all observations of consumers with Income less than 4,071 USD while the right hand branch corresponds to MedianIncome > 4.071, i.e, all observations of consumers with Income greater than 4,071 USD.

Also, in order to predict each our observations and outcomes we used the **Top Down Greedy** approach. Under this method, we started at the root node and tried finding the best split that minimized our Regression RSS. Simply put, we selected an X_i and a cut-off **s** that reduced the value of RSS. This way we would have two things on our hands:

- We will have two boxes R_1 and R_2 such that $\mathrm{R}_1 = \mathrm{X} \mid \mathrm{X}_i$ <= s and $\mathrm{R}_2 = \mathrm{X} \mid \mathrm{X}_i$ > s
- **s** is selected to minimize the difference between the expected and actual values of every observation for each of the two boxes:

$$\sum_{i:x_i \in R1} (\mathrm{y}_i - \hat{y}_{R1})^2 + \sum_{i:x_i \in R2} (\mathrm{y}_i - \hat{y}_{R2})^2$$

Using this, we keep producing new branches and we repeat this process at every internal node until our criterion to stop arrives.

Lastly, we can also calculate our *Mean Square of Error* for the algorithm we stated intially and compare it to the *MSE* of our Linear Regression Model to see if the tree was effective in reducing our residual term across all observations. From below, we can make out that our MSE for the Linear Regression was roughly about 0.14 rounding to the nearest second decimal while that of the regression tree was about 0.12. Since the MSE produced from the tree is lower than that created from the Model, we can say that the Regression Tree does a good job of accurately predicting the House Value given the exogenous factors.

In [24]:
```
# Calculating MSE using our older Linear Regression

sqft_above_lr_model = linear_model.LinearRegression()
sqft_above_lr_model.fit(X, y)
new_mse = metrics.mean_squared_error(y, sqft_above_lr_model.predict(X))
new_mse
```

Out[24]:
```
0.13648643291076704
```

In [26]:
```
# Calculating MSE using our New Regression Tree

y_pred_tree = sqft_tree.predict(X)
print('Mean Squared Error:', metrics.mean_squared_error(y, y_pred_tree))
```

```
Mean Squared Error: 0.11958092574871351
```

Now, we can briefly state how the Regression Tree compares to the Linear Regression Analysis after observing the results from both methods. The first major merit of using a tree over a linear model is how effectively the machine learning algorithm enabled us to reduce our *MSE*, thus enabling us to accurately predict our endogenous variable without any major deviations from the actual value. If we plot it out, we will also be able to observe that the OLS line is a straight linear curve passing through almost all scatter points. Another advantage of regression trees is that they adapt automatically to feature scales and units. For instance, the variable Ocean Proximity is a dummy variable which can take values from 1-5. However, we know that it also happens to be a categorical variable since it classifies a certain distance from the house under a unique category for each observation. Regression trees do not impose linearity or monotonicity, so having the variable Ocean Proximity is less harmful under a regression tree. Lastly, the Linear Regression tries to sum up our entire prediction model using one value which is the R^2 whereas the Regression Tree can be interpreted as a graphical breakdown of the process that each observation follows in order to come up with an effective prediction.

Conclusion

The aim of our research cum analysis was twofold viz:

- Find out which factors affect the price of a house in the state of California the most. Our study was spread across all the counties in the state. These factors were selected from the various possible variables available in the sample dataset provided. We will call them the Physical Factors.
- In addition, the study also tried to find other important factors that a prospective house owner could consider that could influence his/her decision to buy a house in a certain area. These factors were not part of the sample dataset and were found externally by analysing literature, journals etc. We will call these Social Factors.

Physical Factors: We found 4 factors that could contribute to the house prices that I would like to group them under Physical Factors. These are proximity to the ocean, age of the house, average occupancy in an area and average income of a particular region. It was observed that more than 50% of the houses are less than 1 hour away from the ocean with their median price range being > 180,000 USD. It was evident that there are only a very few Island houses but probably being very few they demand a high price of in excess of 300,000 USD. Over half the houses in the state of California were built almost 30 - 35 years back. It tells us that construction of new houses is not something that is happening and people are contend to live in older houses. They may renovate, undertake repairs to suit their living standard or personal preferences but not many new constructions have been taking place. Another factor that plays an important role is the Average Income of the community where the property is located. The study realized the hypothesis we started with was in fact True and that the average income has a direct positive correlation with the average price of a house in a certain area. The higher the average income of the neighbourhood was, the higher the price of the house vis-a-vis its counterpart in slightly less richer neighbourhood. This also makes a lot of commercial sense as rich neighbourhoods typically have higher per square foot price, have bigger and more glamorous properties and therefore demand a much higher premium compared to some of their lesser cousins in other communities.

Social Factors: Additionally, we have analysed and scrutinized two other factors that could enable a buyer to examine prospective houses and neighbourhoods in a more objective manner. These are the Crime Rates in a county and School Districts. We find that these two factors have direct correlation with the house pricing. People prefer to choose their house in a relatively less crime infested area with good school districts close by showing a positive correlation.

The study engaged in understanding the relationship between the narrowed down impact factors and the house price by using different mathematical and statistical tools such as Histograms, Bar Charts, Scatter Plots and Overlay of data onto physical state map of California, Ordinary Least Squares Linear Regression methodology and finally using Machine Learning Regression Tree algorithms to re-iterate the relationships. Not withstanding the method of analysis, each method kept building on top of the previous technique thereby reimposing the faith on our conclusion.

Everything though was not perfect in our study and there is a lot of scope for improvement in future. To start with, the dataset is well short of data that can more objectively help to find out important factors to consider when looking to buy a house in California. Factors such Crime rate, Schooling data, Public Transport facilities, Distance from Office Downtown district, Ethinicity are some of the ones that immediately come to one’s mind. Moreover, some of the data even though present could not be interpreted since it lacked the depth and was dependent on certain other data which was blatantly missing from the dataset