Why live in California?— The Relationship between the Price of House, It’s Location and Structure

Author: Aichen Liu

Date: Dec. 18 2020

Introduction

California has been experiencing an extended and increasing housing shortage. Some believe that California's lack of affordable housing has reached a breaking point (Outlier -- or Harbinger 2019). California is one of the states with the highest proportion of high-income workers. This phenomenon creates significant pressure on middle-class workers on both buying and renting houses in California. Economics equality and an even distribution of wealth are essential for the long-term development of a society. In order to address this problem, it is impotent to find the factors that affect the housing price in California.

As indicated by the Legislative Analyst’s Office, they believe that “(we are) building less housing than people demand.” California is a desirable place to live in. Yet not enough housing exists in the communities to accommodate all of the households that want to live there (California Legislative Analyst’s Office 2015, under “Executive Summary”). Why do people want to live in California? What features the house have that attract so many buyers that drive up the prices? This paper builds onto the contribution of the California Legislative Analyst’s Office’s article, California’s High Housing Costs: Causes and Consequences and explores the factors that affect the prices for existing houses. One of the reasons to explore the pattern is that it would help us better understand the direction of the real estate market in order to better predict future housing prices. This paper uses Gamma regression and spatial visualizations to compare the effect of the house’s structure, house’s location, surrounding environment, public utilities/facilities on California housing price. Then it further analyzes the relationship between the owner’s income levels and housing price to provides a detailed break-down of the demand of the house buyers.

According to Hans P. Johnson, he believes that during the 1990s, there was a disturbing and widely noted decline in the construction of houses. This decline has been attributed to a myriad of causes for the high housing price in California (Johnson 2004, 3). Base on his theory, I also use data from 1990 to compare the housing price in 1990 with the age of the house, the size of the house, the population density in the neighbourhood, the education level, the magnitude of earthquakes in the neighbourhood, the owner’s income and the distance from the house with respect to the sea. This paper finds that housing price in 1990 is significantly affected by the location of the house to the ocean, the age of the houses and the surrounding education institutions (high schools). The following sections will explain the data, detailed analysis and the process of finding the results.

The data California Housing Prices-Median house prices for California districts are used in this paper. This dataset contains information of individual houses using block as the sampling unit. The data contains 20640 observations and each of the observations indicates one house in California and provides the information on the following variables in 1990 California: longitude，latitude，housing_median_age，total_rooms，total_bedrooms，population of people residing within a block，households，median_income，median_house_value，ocean_proximity.

To further analyze the research question, set median_house_value to be the dependent variable Y, and house_median_age, total_rooms, population, ocean_proximity, longitude, latitude and median_income to be the independent variables Xi.

Preview

This paper includes nine parts:

- Data Cleaning
- Summary Statistics and Plots
- Map
- Web-Scraping
- Model Selection
- Regression Result
- Machine Learning

- Conclusion
- Future Work
- Reference

Data Cleaning

The data used is pretty tidy. As indicated below, total_bedrooms is the only variable with missing values. Since it is reasonable that there may exist houses without bedrooms, such as commercial real estate. I decided to replace missing values with 0. The detailed code is shown below.

In [780…

```
data = pd.read_csv("housing.csv")
data.head()
```

Out[780…

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_prox
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR

In [781…

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

In [782…

```
data.isnull().sum()
```

Out[782…

```
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
```

In [783…

```
data.fillna(value=0, axis=1, inplace=True)
data.isnull().sum()
```

Out[783…

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

On the other hand, Ocean_proximity is a categorical variable that indicates the location of the house with respect to ocean. It contains five levels: H_ocean (houses within one-hour drive to ocean), inland, near_ocean, near_bay and island. To make further analysis easier each level is turned into an individual binary variable.

In [826…

```
data.ocean_proximity.value_counts()
```

Out[826…

```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
Name: ocean_proximity, dtype: int64
```

In [827…

```
data = pd.read_csv("housing.csv")
data.fillna(value=0, axis=1, inplace=True)
data['H_Ocean'] = np.where(data['ocean_proximity']== '<1H OCEAN', 1, 0)
data['Inland'] = np.where(data['ocean_proximity']== 'INLAND', 1, 0)
```

```
data['NEAR_OCEAN'] = np.where(data['ocean_proximity']== 'NEAR OCEAN', 1, 0)
data['NEAR_BAY'] = np.where(data['ocean_proximity']== 'NEAR BAY', 1, 0)
data['Island'] = np.where(data['ocean_proximity']== 'ISLAND', 1, 0)
data['intercept'] = 1
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20640 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object 
 10  H_Ocean             20640 non-null  int64  
 11  Inland              20640 non-null  int64  
 12  NEAR_OCEAN          20640 non-null  int64  
 13  NEAR_BAY            20640 non-null  int64  
 14  Island              20640 non-null  int64  
 15  intercept           20640 non-null  int64  
dtypes: float64(9), int64(6), object(1)
memory usage: 2.5+ MB
```

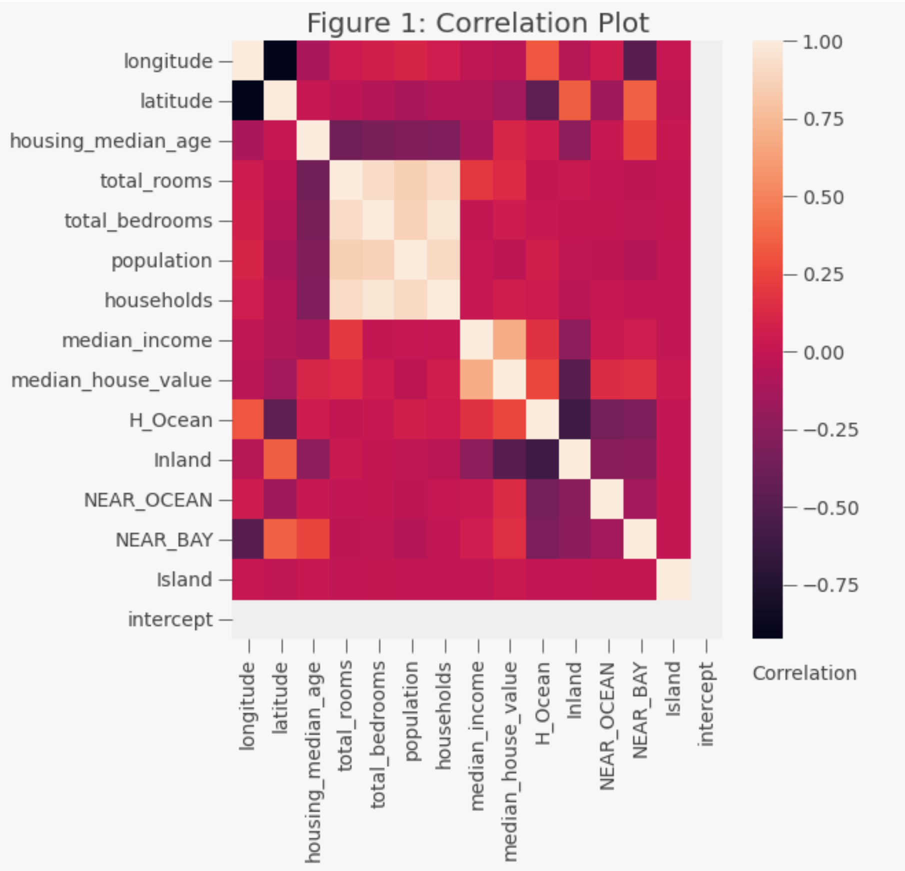
After creating the new variables, a heatmap of the correlation between each variable is shown below in figure 1. If the colour is close to black, it means that the correlation is close to negative one; If the colour is close to coral, it means that the correlation is close to one.

In [829…

```
fig, ax = plt.subplots(figsize=(8,8))
sns.heatmap(data.corr(),annot=False,annot_kws={"size":7.5})
plt.annotate('Correlation',xy=(0.83, 0.22),  xycoords='figure fraction')
plt.title('Figure 1: Correlation Plot')
```

Out[829…

```
Text(0.5, 1.0, 'Figure 1: Correlation Plot')
```



The table below summarizes the linear correlation between each variable and median_house_value. As indicated, there is a strong positive correlation between median_house_price and median_income, which is 0.68. On the other hand, median_house_price has a weakly negative correlation between latitude, longitude. Also the median_house_price has a strong correlation with latitude, longitude and ocean_proximity. Besides, from the heatmap we observe that the correlation between population, household, total rooms and total bedrooms are also very strong.

In [787…

```
corr_matrix=data.corr()
corr_matrix.median_house_value.sort_values(ascending=False)
```

Out[787…

```
median_house_value    1.000000
median_income         0.688075
H_Ocean               0.256617
NEAR_BAY              0.160284
NEAR_OCEAN            0.141862
```

```
total_rooms            0.134153
housing_median_age     0.105623
households             0.065843
total_bedrooms         0.049148
Island                 0.023416
population            -0.024650
longitude             -0.045967
latitude              -0.144160
Inland                -0.484859
intercept                   NaN
Name: median_house_value, dtype: float64
```

Summary Statistics and Plots

To further explore the structure of data, A summary table of statistics for all the numerical variables is shown below. It contains standard deviations, min and max values, and percentiles, which give us a better understanding of the range and the distribution of each variable.

In [788…

```
round(data[["housing_median_age",'households', "total_rooms",'total_bedrooms',
      "population","median_income","median_house_value", "ocean_proximity"]].describe(),3)
```

Out[788…

	housing_median_age	households	total_rooms	total_bedrooms	population	median_income	median_house_value
count	20640.000	20640.00	20640.000	20640.000	20640.000	20640.000	20640.000
mean	28.639	499.54	2635.763	532.476	1425.477	3.871	206855.817
std	12.586	382.33	2181.615	422.678	1132.462	1.900	115395.616
min	1.000	1.00	2.000	0.000	3.000	0.500	14999.000
25%	18.000	280.00	1447.750	292.000	787.000	2.563	119600.000
50%	29.000	409.00	2127.000	431.000	1166.000	3.535	179700.000
75%	37.000	605.00	3148.000	643.250	1725.000	4.743	264725.000
max	52.000	6082.00	39320.000	6445.000	35682.000	15.000	500001.000

Histograms provide better visualization for the distribution of variables. For each histogram below, the vertical dashed line indicates the mean and the solid curve illustrates the density line.

Figures 2, 3 and 4 each represent the density plot for Median_income, total_rooms and housing_median_age. Figure 5 represents the frequency plot of the median_house_value. Median_income(Figure 2), total_rooms(Figure 3) are right-skewed, whereas housing_median_age(Figure 4) is multimodal with some extreme values at the right tail. median_house_value(Figure 5) is also right-skewed with some extreme values at the right tail.

In [789…

```
fig, ax = plt.subplots()
data.plot(
    kind="hist", y="median_income", color=(244/255, 77/255, 24/255),
    bins=50, legend=False, density=True, ax=ax
)
plt.axvline(data["median_income"].mean(), color='k', linestyle='dashed', linewidth=1)
ax.set_facecolor((0.96, 0.96, 0.96))
fig.set_facecolor((0.96, 0.96, 0.96))
plt.xlabel('median_income (in tens of thousands of USD)')
data["median_income"].plot.density(xlim=(0, 15))

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.set_title("Figure 2: Histogram for Median Income")
```

Out[789…

```
Text(0.5, 1.0, 'Figure 2: Histogram for Median Income')
```

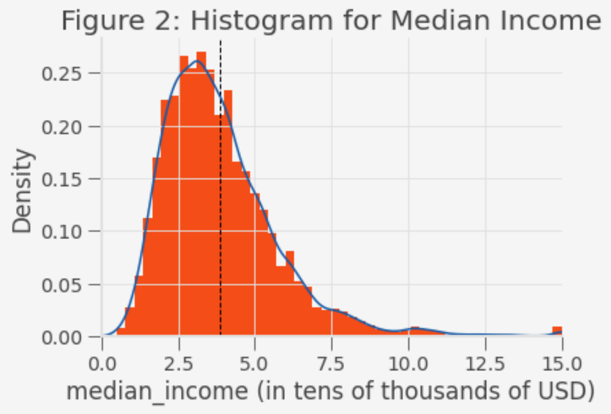


Figure 2: Histogram for Median Income

In [790…

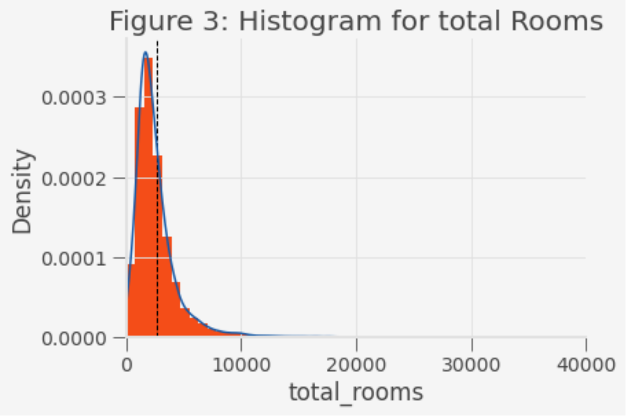
```
fig, ax = plt.subplots()
data.plot(
    kind="hist", y="total_rooms", bins=50,color=(244/255, 77/255, 24/255),
    legend=False, density=True, ax=ax
)
plt.axvline(data["total_rooms"].mean(), color='k', linestyle='dashed', linewidth=1)
ax.set_facecolor((0.96, 0.96, 0.96))
fig.set_facecolor((0.96, 0.96, 0.96))
```

```
plt.xlabel('total_rooms')
data["total_rooms"].plot.density(xlim=(0, 40000))

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.set_title("Figure 3: Histogram for total Rooms")
```

Out[790…

```
Text(0.5, 1.0, 'Figure 3: Histogram for total Rooms')
```



In [791…

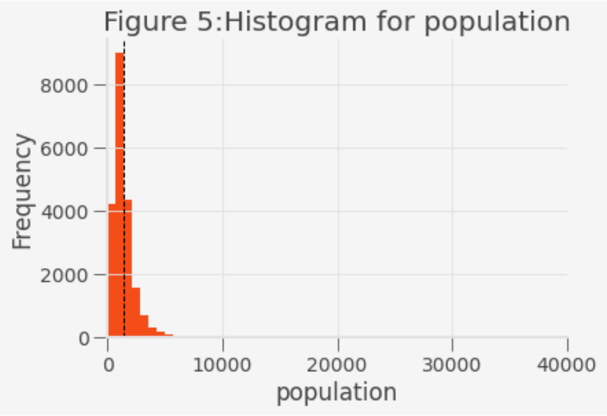
```
fig, ax = plt.subplots()
data.plot(
    kind="hist", y="population", bins=50,color=(244/255, 77/255, 24/255),xlim=(0, 40000),
    legend=False, density=False, ax=ax
)
plt.axvline(data["population"].mean(), color='k', linestyle='dashed', linewidth=1)
ax.set_facecolor((0.96, 0.96, 0.96))
fig.set_facecolor((0.96, 0.96, 0.96))

plt.xlabel('population')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.set_title("Figure 5:Histogram for population")
```

Out[791…

```
Text(0.5, 1.0, 'Figure 5:Histogram for population')
```



In [792…

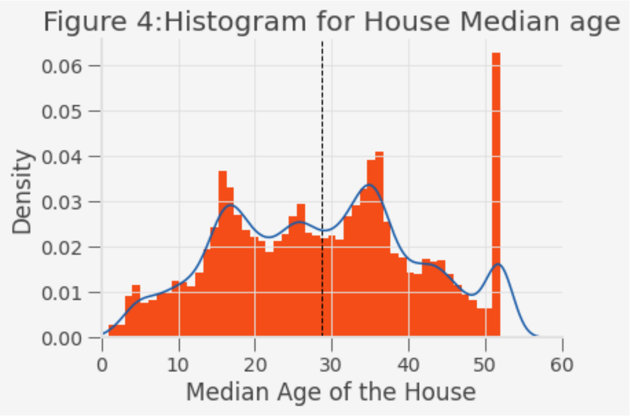
```
fig, ax = plt.subplots()
data.plot(
    kind="hist", y="housing_median_age", bins=50,color=(244/255, 77/255, 24/255),
    legend=False, density=True, ax=ax
)
plt.axvline(data["housing_median_age"].mean(), color='k', linestyle='dashed', linewidth=1)
ax.set_facecolor((0.96, 0.96, 0.96))
fig.set_facecolor((0.96, 0.96, 0.96))

plt.xlabel('Median Age of the House')
data["housing_median_age"].plot.density(xlim=(0, 60))

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.set_title("Figure 4:Histogram for House Median age")
```

Out[792…

```
Text(0.5, 1.0, 'Figure 4:Histogram for House Median age')
```



In [793…

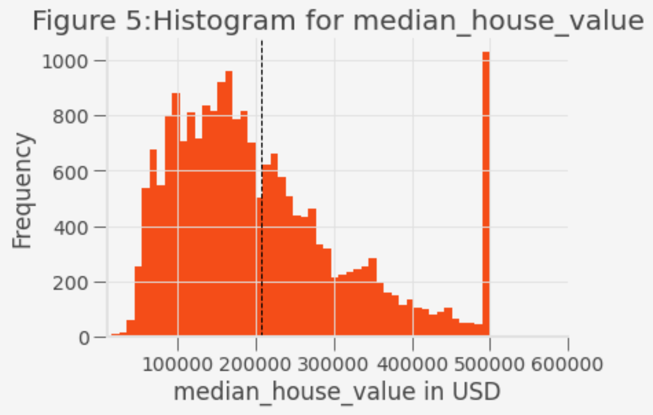
```
fig, ax = plt.subplots()
data.plot(
    kind="hist", y="median_house_value", bins=50,color=(244/255, 77/255, 24/255),xlim=(10000, 600000),
    legend=False, density=False, ax=ax
)
plt.axvline(data["median_house_value"].mean(), color='k', linestyle='dashed', linewidth=1)
ax.set_facecolor((0.96, 0.96, 0.96))
fig.set_facecolor((0.96, 0.96, 0.96))

plt.xlabel('median_house_value in USD')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.set_title("Figure 5:Histogram for median_house_value")
```

Out[793…

```
Text(0.5, 1.0, 'Figure 5:Histogram for median_house_value')
```



The following scatterplots are to indicate the correlation between variables. Figure 6 shows the correlation between median house value and median income, and the red line is the line of best fit. As indicated, these two variables are positively correlated, which means that people with higher incomes tend to buy a more expensive house. On the other hand, red dots indicate inland houses and blue dots represent other houses. Red dots are clutter below the line of best fit, which shows that inland houses are not as expensive as other types of houses.

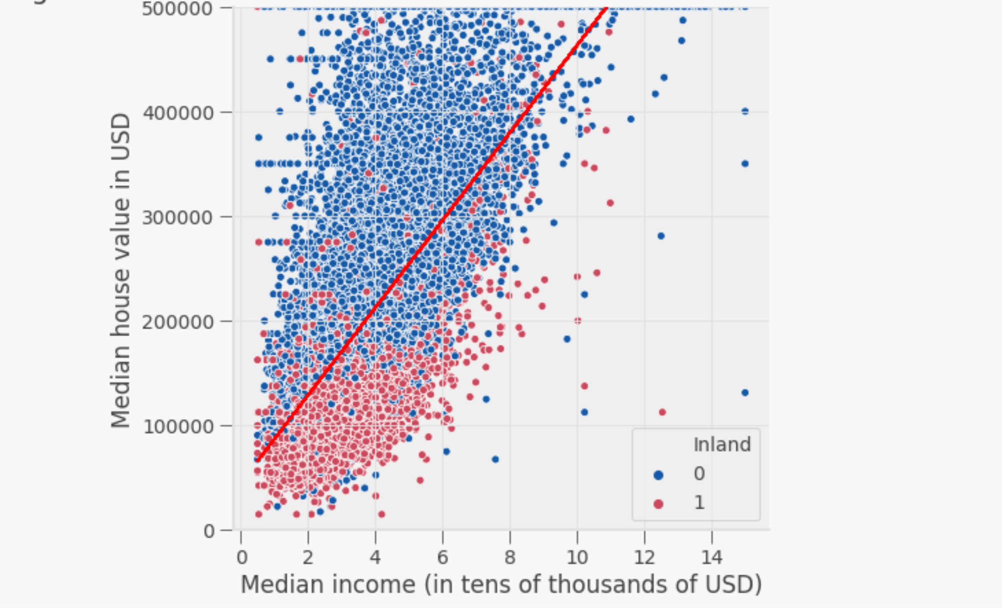
Figure 7 indicates the correlation between median house value and total rooms. Most blocks have total rooms within the range of 0-10000, but the house price scatters haphazardly. Thus, there is no significant relationship between housing prices and the number of rooms.

In [794…

```
plt.figure(figsize=(7,7))
sns.scatterplot(x="median_income", y="median_house_value",hue="Inland",
              data=data);

m, b = np.polyfit(data["median_income"], data["median_house_value"], 1)
plt.ylim(0, 500000)
plt.plot(data["median_income"], m*data["median_income"] + b,color = "r")
plt.xlabel('Median income (in tens of thousands of USD)')
plt.ylabel('Median house value in USD')
plt.title('Figure 6: Correlation between Median income and Median House value')
plt.show()
```

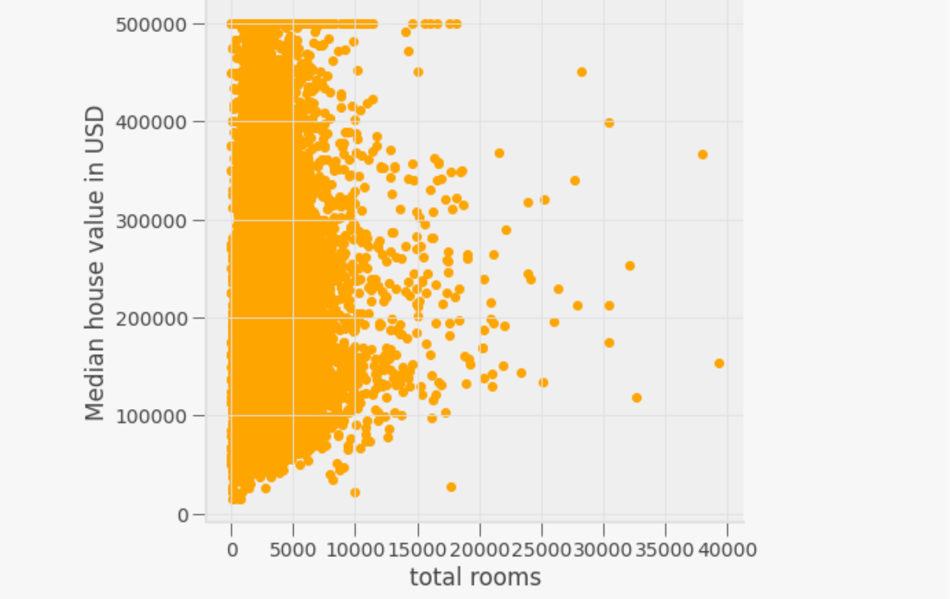
Figure 6: Correlation between Median income and Median House value



In [795…

```
plt.figure(figsize=(7,7))
plt.scatter(data["total_rooms"], data["median_house_value"],c='orange')
plt.xlabel('total rooms')
plt.ylabel('Median house value in USD')
plt.title('Figure 7: Correlation between total rooms and Median House value')
plt.show()
```

Figure 7: Correlation between total rooms and Median House value



The following boxplot explains the correlation between house price and ocean_proximity. The range of IQR for inland houses is the lowest with no overlap with others, but with some outliers at the top. Therefore we can conclude that the majority of inland houses have a low price. The range of IQR for houses on the island is the highest with no overlap with others. Thus, in general, the housing price on the island is more expensive. For <1H ocean, by bay and by ocean, the distribution and range of the house price are similar.

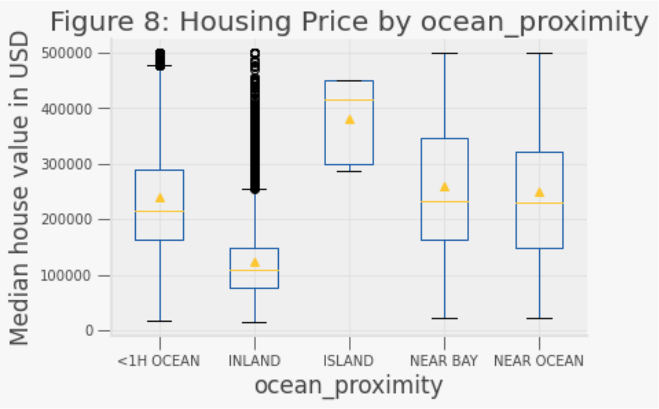
In [796…

```
plt.figure(figsize=(8,8))
data.boxplot(column="median_house_value", by="ocean_proximity",fontsize=10,showfliers=True, showmeans=True)

plt.ylabel('Median house value in USD')
plt.title("Figure 8: Housing Price by ocean_proximity")

plt.suptitle("")
plt.show()
```

```
<Figure size 576x576 with 0 Axes>
```



Map

Figure 9 is a map of California where each dot represents one observation in the dataset. The lower border is the shore, the upper border is the inland area. As indicated from the scale on the right, the colour of dots is corresponding to the price of houses. The more the colour is close to red and yellow, the more expensive the house is; The more the colour is close to blue and purple, the cheaper the house is. Besides, the size of the dots represents the population density of the corresponding neighbourhood. The larger the circle, the higher the population density around that area. We observed that the red dots and yellow dots gather around the shore, which means that houses by the ocean or by the bay are more expensive than houses inland. Also, the size of the circle is larger for those closer to the shore, hence we know that the population density is greater in areas close to the shore.

The majority of the large circles are in light blue, whereas the relatively small circles are in either dark blue or orange. This observation indicates that inland areas with low population density have low housing prices, whereas the near-shore areas with low population density tend to have higher housing prices. High population density doesn't imply high housing prices or low housing prices.

In [797…

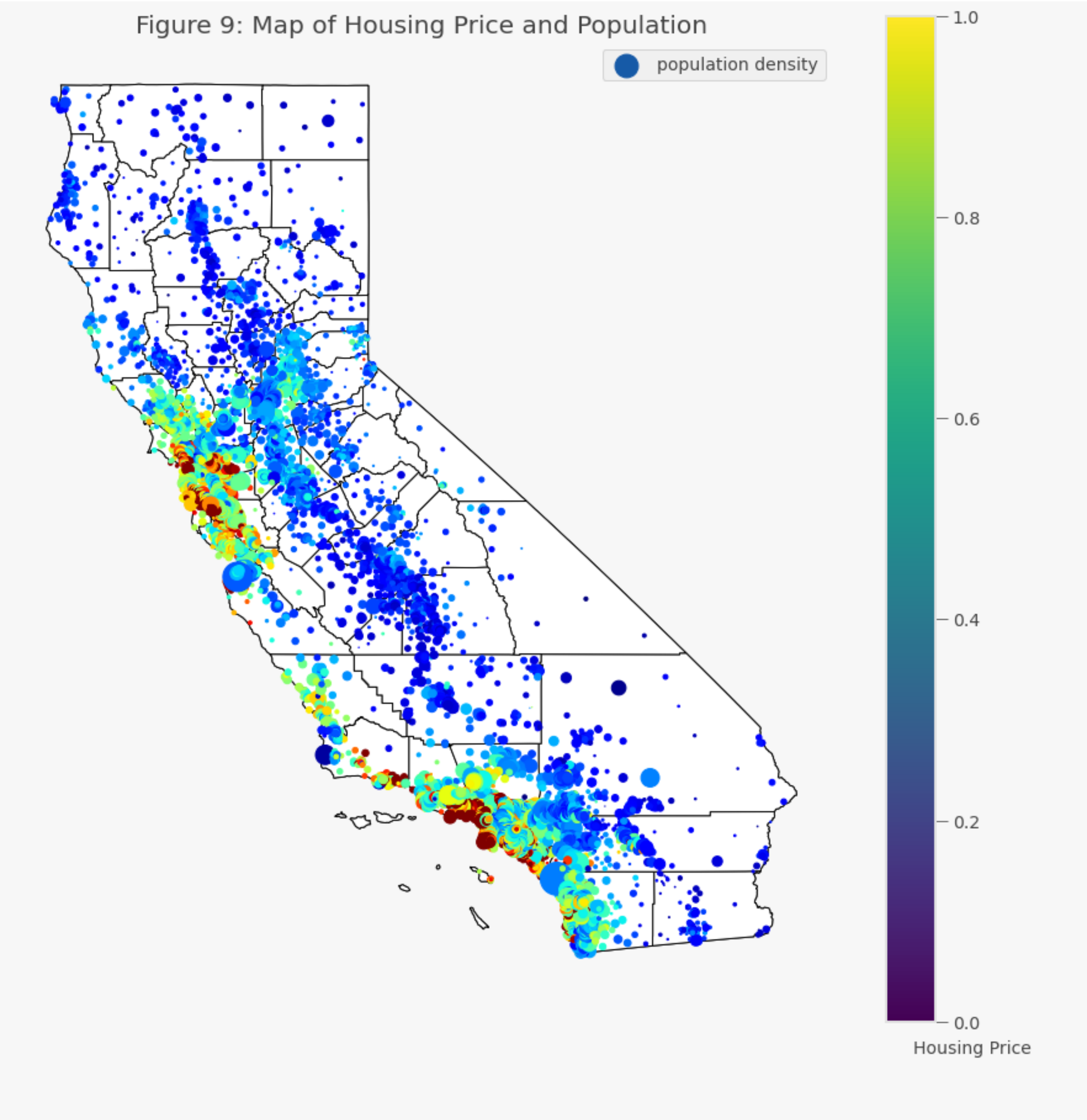
```
county_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_county_5m.zip")
state_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_state_5m.zip")
data["Coordinates"] = list(zip(data.longitude, data.latitude))
data["Coordinates"] = data["Coordinates"].apply(Point)
g_data = gpd.GeoDataFrame(data, geometry="Coordinates")

county_df_C = county_df.query("STATEFP == '06'")

fig, gax = plt.subplots(figsize=(15, 15))
state_df.query("NAME == 'California'").plot(ax=gax, edgecolor="black", color="white")
county_df_C.plot(ax=gax, edgecolor="black", color="white")

data.plot(ax=gax,kind='scatter',x='longitude', y='latitude',
         s=data['population']/50, label='population density',
         c=data['median_house_value'],
         cmap=plt.get_cmap('jet'),
         colorbar=True,
         figsize=(15,15))

gax.annotate('Housing Price',xy=(0.84, 0.06),  xycoords='figure fraction')
plt.axis('off')
plt.title("Figure 9: Map of Housing Price and Population")
plt.show()
```



Web-Scraping

Besides the structure of the houses, the surrounding environment, public utilities/facilities also have a significant effect on the price of the house. For instance, the surrounding school districts, surrounding secondary schools rating, and the frequency of natural disasters should also be taken into account. Since we don't have available data, we can conduct web-scarping for the two websites I found and collect the information from the scratch. Here are the addresses of the two websites. https://school-ratings.com/svg/index.html and http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_021708_Earthquakes

The first website includes an interactive map about the school-ratings of California for each of nearly 9000 public schools. The rating is determined by a school's API Score in comparison to all other schools in California where 1 is the worst and 10 is the best. I chose an image over a dataset because the datasets from the California Department of Education are not public. I think this map is very useful because I could scrape this image and compare it with the map I generated above to visualize the pattern and the correlation between the housing price and the goodness of the surrounding schools. Since the website I chose is an image, we don't need to merge or run it over time.

The scarped image is in the URL below. I have attached the image in the appendix. In this image, each dot represents one school, the greener the dots, the better the school. There are three clusters of green dots; One is on the lower shore, one is around the bay, and the other one is in the middle inland area. The corresponding areas in Figure 9 are mostly in red and green. This means that the price of the house is higher in the neighbourhood with high-rating high schools.

In [653…

```
html = urlopen('https://school-ratings.com/svg/index.html')
imge="https://www.school-ratings.com/svg/caState2.svgz"
bs = BeautifulSoup(html, 'html.parser')
images = bs.find_all('img')
for image in images:
    print("The images are:")
    print(image['src']+'\n')
    print(imge)
```

```
The images are:
../schoolRatingsSmall.gif
```

```
https://www.school-ratings.com/svg/caState2.svgz
The images are:
http://creativecommons.org/images/public/somerights20.gif

https://www.school-ratings.com/svg/caState2.svgz
```

The second website is the SOCR Data - California Earthquake Data from the Northern California Earthquake Data Center (NCEDC) provided by the professor. This dataset contains a table of the information of the earthquakes from 1969 October to 2007 November; It contains the longitude, latitude, magnitude, depth, and the date of the earthquakes. Since both my original data and the earthquake data have longitude and latitude, I plan to merge the new earthquake data onto my original data by both longitude and latitude. After merging the data, I will generate a scatterplot of the relationship between median_house_value versus the magnitude of the earthquakes.

Environment and surrounding school districts are very important factors to consider for both potential buyers and developers because these two factors significantly affect the appreciation of housing. The demand for a house located in a neighbourhood with nice facilities and great schools is often higher than others. Hence, houses are more likely to appreciate. In contrast, if a house is located in an area where major earthquakes took place, then both realtors and buyers would avoid the house which would drive down the prices. So I hypothesized that there might be a relationship between the magnitude of the earthquakes and the housing price. The following codes show the scraping process of the earthquake data.

In [798…

```python
web_url = 'http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_021708_Earthquakes'
response = requests.get(web_url)
soup_object = BeautifulSoup(response.content)
data_table = soup_object.find_all('table', 'wikitable')[1]
all_values = data_table.find_all('tr')

Earthquake = pd.DataFrame(columns = ['Date', 'Time', 'latitude','longitude','Depth',
                                     'Mag','Magt','Nst','Gap','Clo','RMS','SRC','EventID'])
ix = 0

for row in all_values[1:]:
    values = row.find_all('td')
    date = values[0].text
    time = values[1].text
    lat = float(values[2].text)
    long= float(values[3].text)
    depth = values[4].text
    mg  = float(values[5].text)
    mgt= values[6].text
    nst= values[7].text
    gap= values[8].text
    clo= values[9].text
    rms= values[10].text
    src= values[11].text
    eid= values[12].text.replace('\n','')
    Earthquake.loc[ix] = [date, time, lat,long,depth,mg,mgt,nst,gap,clo,rms,src,eid] # Store it in the dataframe as a ro
    ix += 1

Earthquake.head()
```

Out[798…

	Date	Time	latitude	longitude	Depth	Mag	Magt	Nst	Gap	Clo	RMS	SRC	EventID
0	1969/10/02	04:56:45.30	38.4978	-122.6640	0.22	5.6	ML	38	104	52	0.22	NCSN	-1003132
1	1969/10/02	06:19:56.39	38.4500	-122.7535	5.14	5.7	ML	53	139	58	0.22	NCSN	-1003135
2	1972/02/24	15:56:50.99	36.5903	-121.1905	4.18	5.1	ML	10	128	6	0.06	NCSN	-1009260
3	1974/11/28	23:01:24.59	36.9202	-121.4673	5.48	5.2	ML	51	61	4	0.13	NCSN	-1021953
4	1975/06/07	08:46:23.51	40.5415	-124.2763	23.48	5.3	ML	15	176	5	0.04	NCSN	-1024134

In [799…

```python
Earthquake.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 168 entries, 0 to 167
Data columns (total 13 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       168 non-null    object 
 1   Time       168 non-null    object 
 2   latitude   168 non-null    float64
 3   longitude  168 non-null    float64
 4   Depth      168 non-null    object 
 5   Mag        168 non-null    float64
 6   Magt       168 non-null    object 
 7   Nst        168 non-null    object 
 8   Gap        168 non-null    object 
 9   Clo        168 non-null    object 
 10  RMS        168 non-null    object 
 11  SRC        168 non-null    object 
 12  EventID    168 non-null    object 
dtypes: float64(3), object(10)
memory usage: 18.4+ KB
```

The three chunks of code above show the detailed process of web-scraping. Firstly, find the URL of the website and check the status code. If the status code is within the range of 200-299, it means that the request was successfully completed and it is OK the continue scraping. Then use the function BeautifulSoup() to extract the content of the website into the object called soup_object. By reading through the content of the website, we can locate the data table that we want to scrap is the second one on the website. The data table has an HTML tag called table and with class wikitable. Then we can further extract the data table information from the website using the code <soup_object.find_all('table', 'wikitable')[1]> and store it in the variable data_table. Then we find all the "tr" values from the data_table. Lastly, create an empty data frame called Earthquake. Then simply iterate over the rows of the dataset and extract all elements with tag td. Then pick only the text part from the td and insert the value from each data-contained cell to the empty data frame. The values in column longitude, latitude and magnitude are converted from string type to float type, in order to merge with the original data.

The scraped data has 168 observations. It is stored in variable Earthquake. The column information and the first six rows of the data are shown above.

In [800…

```
Earthquake=Earthquake.round({'latitude': 2, 'longitude': 2})
merged = pd.merge(data, Earthquake, how='left',on=["latitude", "longitude"])
merged =merged.drop(['EventID','SRC','RMS','Clo','Gap','Nst','Magt','Depth','Time','Date'],axis=1)
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20640 entries, 0 to 20639
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype   
---  ------              --------------  -----   
 0   longitude           20640 non-null  float64 
 1   latitude            20640 non-null  float64 
 2   housing_median_age  20640 non-null  float64 
 3   total_rooms         20640 non-null  float64 
 4   total_bedrooms      20640 non-null  float64 
 5   population          20640 non-null  float64 
 6   households          20640 non-null  float64 
 7   median_income       20640 non-null  float64 
 8   median_house_value  20640 non-null  float64 
 9   ocean_proximity     20640 non-null  object  
 10  H_Ocean             20640 non-null  int64   
 11  Inland              20640 non-null  int64   
 12  NEAR_OCEAN          20640 non-null  int64   
 13  NEAR_BAY            20640 non-null  int64   
 14  Island              20640 non-null  int64   
 15  intercept           20640 non-null  int64   
 16  Coordinates         20640 non-null  geometry
 17  Mag                 11 non-null     float64 
dtypes: float64(10), geometry(1), int64(6), object(1)
memory usage: 3.0+ MB
```

In [801…

```
merged.isnull().sum()
```

Out[801…

```
longitude                 0
latitude                  0
housing_median_age        0
total_rooms               0
total_bedrooms            0
population                0
households                0
median_income             0
median_house_value        0
ocean_proximity           0
H_Ocean                   0
Inland                    0
NEAR_OCEAN                0
NEAR_BAY                  0
Island                    0
intercept                 0
Coordinates               0
Mag                   20629
dtype: int64
```

The longitude and latitude in the new earthquake data are rounded to two decimal places, to merge it with the original data. Since not all the locations had earthquakes before, left join is used in this case to keep all the observations in the original data. The merged dataset is stored in the variable merged. The column names are shown above.

We are only interested in the magnitude of the earthquakes, therefore only variable Mag is merged to the original data.

The newly merged dataset contains a lot of missing values because we used left merge. Among the 20640 observations, only 11 observations have the earthquake values.

In figure 11, all 168 earthquake data are plotted on the map. The majority of the earthquakes happened at the border of California with a magnitude around 5 to 6. And some of the earthquakes happened outside of California. In figure 12, we used the merged data to plot the correlation between the magnitude of the earthquakes and the housing price. As indicated, the 11 data points randomly scatter in the plot. Therefore there is no significant relationship existed.

Since we only have 11 data points, we can't include the vairable Mag in the regression in the next section, because it would decrease the power of our model.

In [802…

```
Earthquake["Coordinates"] = list(zip(Earthquake.longitude,Earthquake.latitude))
Earthquake["Coordinates"] = Earthquake["Coordinates"].apply(Point)
g_Earthquake = gpd.GeoDataFrame(Earthquake, geometry="Coordinates")
#filter geometry information for California
county_df_C = county_df.query("STATEFP == '06'")
#Plot
fig, gax = plt.subplots(figsize=(10, 10))
state_df.query("NAME == 'California'").plot(ax=gax, edgecolor="black", color="white")
county_df_C.plot(ax=gax, edgecolor="black", color="white")

g_Earthquake.plot(
    ax=gax, edgecolor='black',column='Mag', legend=True, cmap='RdBu_r',
    vmin=5, vmax=8)

gax.annotate('Magnitude of Earthquakes',xy=(0.7, 0.06),  xycoords='figure fraction')
plt.axis('off')
plt.title("Figure 11: Map of California Earthquakes")
plt.show()
```

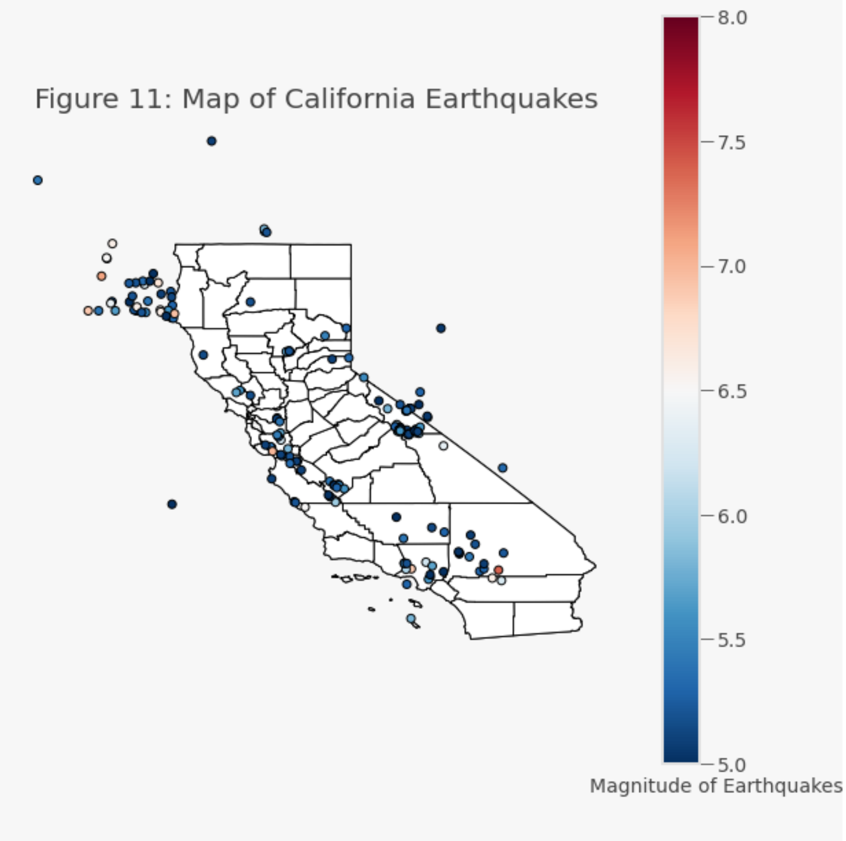


Figure 11: Map of California Earthquakes

In [803…

```
plt.figure(figsize=(5,5))
plt.scatter(merged["median_house_value"], merged["Mag"],c='orange')
plt.xlabel('Magnitude of Earthquakes')
plt.ylabel('Median house value')
plt.title('Figure 12:Correlation between Magnitude of Earthquakes and Median House value')
plt.show()
```

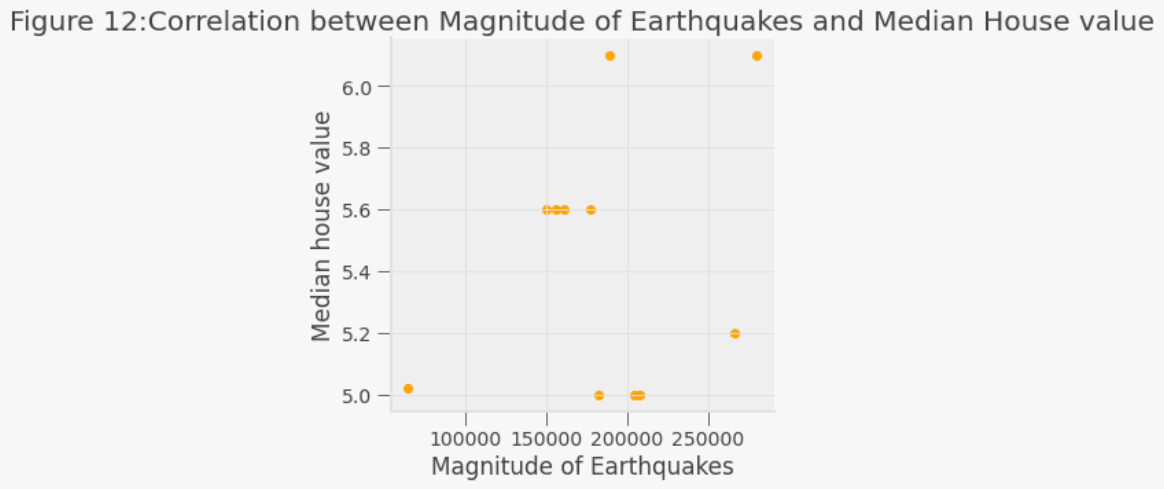


Figure 12:Correlation between Magnitude of Earthquakes and Median House value

Model Selection

In the introduction part of this paper, we set the dependent variable(Y) to be median_house_value. Variables such as total_rooms, population, ocean_proximity, longitude, latitude and median_income are set to be the covariates. Theory of Price states that the price of an object reflects the cumulative value of this object. Therefore, houses demand more construction would lead to higher costs, then lead to a higher price. On the other hand, since houses located in urban areas with high population density have a relatively large room for appreciation, the price for such a house would be higher as well. Hence, based on these economic intuitions, the relationship between the dependent variable and covariates is linear.

Based on the map in the previous section, the ambiguous relationship between the population density and housing price need to be tested in the model. Since the range of median_house_value is between 14999 to 500001, a natural log(Y) is considered to transform the dependent variable to reduce the variance when fitting the regression. All of the variables in the dataset should be included in the regression except longitude and latitude, because longitude and latitude represent locations, the magnitude of these two variables don't mean anything.

Four regressions are fitted below. The first one takes all the variables as covariates (except longitude and latitude) and regress on log(y). The reason I include all variables is that I want to conduct backward selection to ensure all significant variables are included in the model. The p-values for all variables are less than 0.05, therefore all variables in the model are significant. It is adequate to include all variables. The accuracy measurement coefficients are stated below.

In [840…

```
y1 = np.log(data['median_house_value'])
x1 = data[['intercept','housing_median_age','total_rooms','total_bedrooms',
         'population','households','median_income','H_Ocean','Inland','NEAR_OCEAN',"NEAR_BAY","Island"]]
reg1 = sm.OLS(y1, x1, missing='drop').fit()
print(round(reg1.pvalues,3))
prediction1 = reg1.predict()

rmse1 = mean_squared_error(y1, prediction1)
mae1 = sklearn.metrics.mean_absolute_error(y1, prediction1)
print('AIC:',round(reg1.aic,3))
print('BIC:',round(reg1.bic,3))
print('RMSE:',round(rmse1,3))
print("MAE:",round(mae1,3))
print("AdjR:",round(reg1.rsquared_adj,3))
```

```
intercept             0.0
housing_median_age    0.0
total_rooms           0.0
total_bedrooms        0.0
population            0.0
households            0.0
median_income         0.0
H_Ocean               0.0
Inland                0.0
NEAR_OCEAN            0.0
NEAR_BAY              0.0
Island                0.0
dtype: float64
AIC: 13839.041
BIC: 13926.325
RMSE: 0.114
MAE: 0.258
AdjR: 0.647
```

The second model uses the same covariates and dependent variable, the only difference is that interaction terms are added. Base on the results from the correlation matrix, we found a strong correlation between population, household, total_room, and total_bedroom. Adding interactions can control the dependency between covariates. The p values and accuracy coefficients are stated below. The p-values are all less than 0.05, therefore all variables and interactions are significant.

In [832…

```
data['logy'] = np.log(data['median_house_value'])
mod = smf.ols(formula='logy ~ intercept + housing_median_age+\
total_rooms*total_bedrooms + population*households +  median_income+Inland+ NEAR_OCEAN+NEAR_BAY+\
Island+H_Ocean', data=data)
reg2 = mod.fit()

prediction2 = reg2.predict()
rmse2 = mean_squared_error(y1, prediction2)
mae2 = sklearn.metrics.mean_absolute_error(y1, prediction2)
print('AIC:',round(reg2.aic,3))
print('BIC:',round(reg2.bic,3))
print('RMSE:',round(rmse2,3))
print("MAE:",round(mae2,3))
print("AdjR:",round(reg2.rsquared_adj,3))
print(round(reg2.pvalues,3))
```

```
AIC: 13620.342
BIC: 13723.497
RMSE: 0.113
MAE: 0.256
AdjR: 0.651
Intercept                     0.0
intercept                     0.0
housing_median_age            0.0
total_rooms                   0.0
total_bedrooms                0.0
total_rooms:total_bedrooms    0.0
population                    0.0
households                    0.0
population:households         0.0
median_income                 0.0
Inland                        0.0
NEAR_OCEAN                    0.0
NEAR_BAY                      0.0
Island                        0.0
H_Ocean                       0.0
dtype: float64
```

Model 3 is built base on model 2. In figure 5, we observe that median_house_value is right-skewed with some extreme values at the right tail. In model 3, we exclude the values greater than 500000 and construct an OLS model base on the new subset of the data.

The histogram for median_house_value without outliers are shown below. After removing the outliers, the distribution is closer to a Gamma or Normal distribution. In model 3, we assume model 3 follows a normal distribution.

The p values and accuracy coefficients for model 3 are stated below.

In [833…

```
data_p = data.loc[(data['median_house_value'] <= 500000)]
```

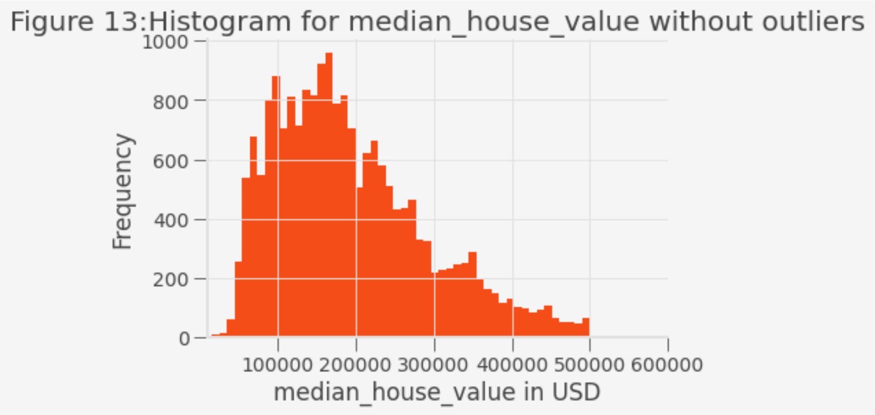
```
fig, ax = plt.subplots()
data_p.plot(
    kind="hist", y="median_house_value", bins=50,color=(244/255, 77/255, 24/255),xlim=(10000, 600000),
    legend=False, density=False, ax=ax
)
#plt.axvline(data["median_house_value"].mean(), color='k', linestyle='dashed', linewidth=1)
ax.set_facecolor((0.96, 0.96, 0.96))
fig.set_facecolor((0.96, 0.96, 0.96))

plt.xlabel('median_house_value in USD')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.set_title("Figure 13:Histogram for median_house_value without outliers")
```

Out[833…

```
Text(0.5, 1.0, 'Figure 13:Histogram for median_house_value without outliers')
```



In [834…

```
mod1 = smf.ols(formula='logy ~ intercept + housing_median_age+\
total_rooms*total_bedrooms + population*households +  median_income+Inland+ NEAR_OCEAN+NEAR_BAY+\
Island+H_Ocean', data=data_p)
reg3 = mod1.fit()
print('AIC:',round(reg3.aic,3))
print('BIC:',round(reg3.bic,3))
print('RMSE:',round(rmse3,3))
print("MAE:",round(mae3,3))
print("AdjR:",round(reg3.rsquared_adj,3))
print(round(reg3.pvalues,3))
```

```
AIC: 11078.391
BIC: 11180.923
RMSE: 0.103
MAE: 0.245
AdjR: 0.639
Intercept                     0.000
intercept                     0.000
housing_median_age            0.000
total_rooms                   0.337
total_bedrooms                0.000
total_rooms:total_bedrooms    0.000
population                    0.000
households                    0.000
population:households         0.000
median_income                 0.000
Inland                        0.000
NEAR_OCEAN                    0.000
NEAR_BAY                      0.000
Island                        0.000
H_Ocean                       0.000
dtype: float64
```

The histogram of median_house_value is a little right-skewed, therefore we can assume it follows a gamma distribution after removing the extreme values. Therefore I decided to fit a Gamma regression for model 4 with a log link function. The covariates and dependent variable are the same as model 2, and the subset of data without extreme values is used. The p values and accuracy coefficients for model 4 are stated below.

In [836…

```
Formula='logy ~ + housing_median_age+\
total_rooms+total_bedrooms + population+households +  median_income+Inland+ NEAR_OCEAN+NEAR_BAY+\
Island+H_Ocean'
d = data_p[['logy','housing_median_age','total_rooms',
         'total_bedrooms','population','households','median_income','Inland','NEAR_OCEAN',
            'NEAR_BAY','Island', 'H_Ocean']]
sm.families.family.Gamma.links
link_g = sm.genmod.families.links.log
link_g
reg4 = smf.glm(formula=Formula, data=d, family=sm.families.Gamma(link_g())).fit()
print('AIC:',round(reg4.aic,3))
print('BIC:',round(reg4.bic,3))
print('RMSE:',round(rmse4,3))
print("MAE:",round(mae4,3))
print(round(reg4.pvalues,3))
```

```
AIC: 11669.091
BIC: -194405.609
RMSE: 0.099
```

```
MAE: 0.24
Intercept             0.0
housing_median_age    0.0
total_rooms           0.0
total_bedrooms        0.0
population            0.0
households            0.0
median_income         0.0
Inland                0.0
NEAR_OCEAN            0.0
NEAR_BAY              0.0
Island                0.0
H_Ocean               0.0
dtype: float64
```

Criteria for selecting the best model is to compare AIC, BIC, RMSE, MAE and Adjusted R^2. AIC measures the information lost in a model, the less information is lost is better. Therefore, we should select the one with the smallest AIC. Similarly, BIC measures whether the model is overfitting. Thus, the model with the smallest BIC is preferred. Adjusted R^2 measures the goodness of fit of the data. So a large adjusted R^2 is preferred. Since GLM doesn't have adjusted R^2, we only compare AIC and BIC.

As stated below, model 3 has the smallest AIC, model 4 has the smallest BIC, model 1 and 2 has the highest adjusted R^2, therefore the goodness of fit for the four models are similar. However, model 4 has the lowest RMSE and MAE which indicates that model 4 has the smallest prediction errors. Therefore, the final model is GLM Gamma.

In [809…

```
print("M1 AIC",round(reg1.aic,3))
print('M2 AIC',round(reg2.aic,3))
print('M3 AIC',round(reg3.aic,3))
print('M4 AIC',round(reg4.aic,3))
print('M1 BIC',round(reg1.bic,3))
print('M2 BIC',round(reg2.bic,3))
print('M3 BIC',round(reg3.bic,3))
print('M4 BIC',round(reg4.bic,3))
```

```
M1 AIC 13839.041
M2 AIC 13620.342
M3 AIC 11078.391
M4 AIC 11669.091
M1 BIC 13926.325
M2 BIC 13723.497
M3 BIC 11180.923
M4 BIC -194405.609
```

In [810…

```
print("M1 AdjR:",round(reg1.rsquared_adj,3))
print("M2 AdjR:",round(reg1.rsquared_adj,3))
print("M3 AdjR:",round(reg3.rsquared_adj,3))
```

```
M1 AdjR: 0.647
M2 AdjR: 0.647
M3 AdjR: 0.639
```

In [811…

```
print('M1 RMSE:',round(rmse1,3))
print('M2 RMSE:',round(rmse2,3))
print('M3 RMSE:',round(rmse3,3))
print('M4 RMSE:',round(rmse4,3))
print("M1 MAE:",round(mae1,3))
print("M2 MAE:",round(mae2,3))
print("M3 MAE:",round(mae3,3))
print("M4 MAE:",round(mae4,3))
```

```
M1 RMSE: 0.114
M2 RMSE: 0.113
M3 RMSE: 0.103
M4 RMSE: 0.099
M1 MAE: 0.258
M2 MAE: 0.256
M3 MAE: 0.245
M4 MAE: 0.24
```

Regression Result

In [812…

```
print(reg4.summary())
```

```
                 Generalized Linear Model Regression Results                  
==============================================================================
Dep. Variable:                   logy   No. Observations:                19675
Model:                            GLM   Df Residuals:                    19664
Model Family:                   Gamma   Df Model:                           10
Link Function:                    log   Scale:                      0.00073387
Method:                          IRLS   Log-Likelihood:                -5823.5
Date:                Thu, 17 Dec 2020   Deviance:                       14.405
Time:                        22:01:08   Pearson chi2:                     14.4
No. Iterations:                     9                                         
Covariance Type:            nonrobust                                         
======================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------
Intercept              2.0249      0.002    926.902      0.000       2.021       2.029
housing_median_age     0.0003   1.79e-05     16.272      0.000       0.000       0.000
total_rooms        -3.172e-06   3.09e-07    -10.278      0.000   -3.78e-06   -2.57e-06
total_bedrooms      1.777e-05   1.99e-06      8.912      0.000    1.39e-05    2.17e-05
population         -1.191e-05   4.23e-07    -28.180      0.000   -1.27e-05   -1.11e-05
households          4.175e-05   2.39e-06     17.442      0.000    3.71e-05    4.64e-05
```

```
median_income          0.0168      0.000    104.676      0.000       0.016       0.017
Inland                 0.3614      0.002    176.704      0.000       0.357       0.365
NEAR_OCEAN             0.4022      0.002    193.342      0.000       0.398       0.406
NEAR_BAY               0.3991      0.002    189.942      0.000       0.395       0.403
Island                 0.4625      0.010     45.777      0.000       0.443       0.482
H_Ocean                0.3997      0.002    194.847      0.000       0.396       0.404
==============================================================================
```

The above table provides the estimated coefficients, p-values and 95% confidence interval of the Gamma regression in log scale. If coef is positive, then the corresponding variable has a positive effect on the housing price; Otherwise, they have a negative effect on the housing price. As indicated, coef for median_income is positive. It implies that people with higher incomes tend to buy more expensive houses. It makes sense because wealthy people have a high propensity to spend. On the other hand, the coef for Inland, NEAR_OCEAN, NEAR_BAY, Island and H_Ocean are 0.3614, 0.4022, 0.3991, 0.4625 and 0.3997 respectively. If we set the general house price to be one, then if a house is located inland, the price of this house would be 1.435 (e^0.3614). If the house is located near the ocean, then the house price would be 1.495 (e^0.4022). Similarly, the price for houses by bay would be 1.49 (e^1.3991), houses on an island would be 1.588 (e^0.4625) and houses within a one-hour drive to the ocean would be 1.4913 (e^0.3997). Hence, we have proven that houses on the island are the most expensive ones, houses near the ocean are the second expensive, houses within a one-hour drive to the ocean are the third expensive, houses near the bay are the fourth expensive, and inland houses are the least expensive. Since potential buyer's willingness to pay for houses that have an ocean view is higher, the high demand drives up the prices of the houses. Also, buildable land for houses on the island and at the sea is very limited, the supply of such houses is limited as well, which further leads to an increase in the house price.

Furthermore, the coef for housing_median_age is 0.0003, which means that older houses are more expensive. Also, we found that houses with more bedrooms are more expensive, but houses with more rooms in total are less expensive. Houses with a large number of households are more expensive but houses with high population density within the block are less expensive. However the coef for total_rooms, total_bedrooms, households and population are very small, they have a mere effect on housing price.

Base on the model, we know that factors that most significantly affect the California housing price are the location of the houses with respect to the ocean and the age of the houses. Houses on the island are the most expensive, houses closer to the ocean and bay are relatively expensive, and inland houses are the least expensive. Buildable land for houses on the island and at the sea is very limited, the supply of such houses is limited, which will lead to an increase in the house price. Besides the location of the houses, the age of the houses also has a positive effect on price. Older houses are more expensive.

Machine Learning

In [838…

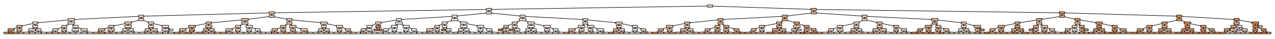
```python
y = data['median_house_value']
x2 = data.drop(['median_house_value','intercept','ocean_proximity','logy'], axis = 1)
x2 = sm.add_constant(x2)
x2 = np.array(x2, dtype=float)
y2 = np.array(y, dtype=float)
```

In [839…

```python
from sklearn import tree
fitted_tree = tree.DecisionTreeRegressor(max_depth=8).fit(x2,y2)
fitted_tree
import graphviz

tree_graph = tree.export_graphviz(fitted_tree, out_file=None,
                                  feature_names=["longitude", "latitude","housing_median_age",
                                                 "total_rooms","total_bedrooms","population",
                                                 "households","median_income","1H Ocean","Inland",
                                                 "NEAR OCEAN","Island",'NEAR_BAY',"median_house_value"],

                                  filled=True, rounded=True,
                                  special_characters=True)
display(graphviz.Source(tree_graph))
```



Conclusion

After a brief data cleaning and some visualizations of the data, we gained a more comprehensive understanding of the background and the structure of the data. The housing price in California is uniquely high even compare with other states like Los Angeles and San Diego. Many reasons drive up the price, such as real estate speculation, but the fundamental cause of this high price is that the demand for California houses excess supply. California has been experiencing an extended and increasing housing shortage. Why do people want to live in California? What unique features the houses in California have that attract people? This paper uses Gamma regression and found that the factors that most significantly affect the housing price in California are the distance of the houses with respect to the ocean and the rating of the surrounding school. As indicated in the analysis above, we found that houses located on the island have the highest price, and houses near the ocean or within a one-hour drive to the ocean have the second-highest price; Houses near the bay are not as expensive as houses near the ocean, and houses located inland have the lowest price. This research shows that people prefer houses with ocean-view. In general houses with ocean-view require higher construction costs, and further, due to the limit of buildable land around the shore, it is reasonable that houses on the island and near the ocean have a higher price. Besides locations, based on the maps, we found that the housing price is higher in the neighbourhood where the rating of high school is high. This implies that people prefer houses in a good school district. Houses in good school district have a large room for appreciation in the future, therefore the demand would be higher which also lead to an increase in price. Furthermore, based on the intuition of the concept of the propensity to spend, we found that people with high-income level owns more expensive housing. The model also proves that the housing price is not affected by the size of the house or the population density in the neighbourhood.

Future Work

To sum up, this paper concludes that the distance with respect to the ocean and quality of school district explains the uniquely high housing price in California. However, one of the limitations of this paper is that it didn't talk much about the effect of the natural environment on the price. Even though I used the earthquake data in the analysis, after merging the occurrence of earthquakes with housing data, only 11 data points are left. Eleven data points cannot include in the regression because a small sample size would reduce the power of the model and increase the prediction error. In the future study, I would like to include more earthquake data points and refit the model to include the natural environment component in the study. If it is possible, data of employments and companies can also be added to the model to further analyze the effect of social aspects on housing prices.

Reference

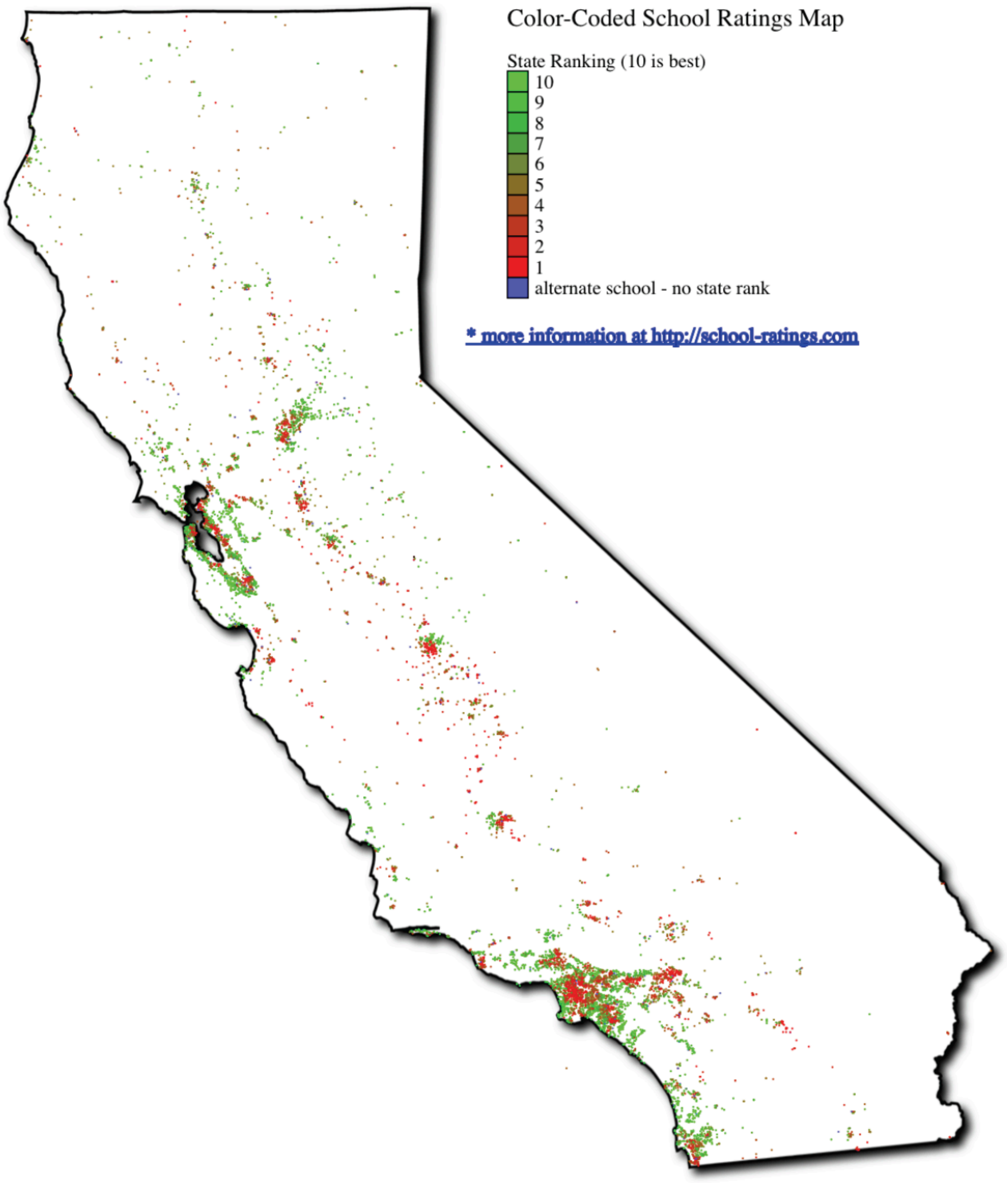
Advisory Board, Podcasts Real Estate Wharton Business Daily North America. "Outlier -- or Harbinger? California's Affordable Housing Crisis." Knowledge@Wharton. Wharton Business Daily, April 1, 2019. https://knowledge.wharton.upenn.edu/article/california-affordable-housing-crisis/.

Legislative Analyst's Office, Legislative Analyst's. "California's High Housing Costs: Causes and Consequences." The California Legislature's Nonpartisan Fiscal and Policy Advisor, March 17, 2015. https://lao.ca.gov/reports/2015/finance/housing-costs/housing-costs.aspx.

Johnson, Hans P., Michael Dardia, and Rosa Maria Moller. In Short Supply?: Cycles and Trends in California Housing. San Francisco, CA: Public Policy Inst. of California, 2004.

Pogol, Gina. "Should You Spend More for a House to Get Better Schools?" hsh.com. HSH ® Associates, Financial Publishers, May 1, 2019. https://www.hsh.com/homebuyer/should-you-spend-more-for-home-with-better-schools.html.

Appendix


Color-Coded School Ratings Map
State Ranking (10 is best)
10
9
8
7
6
5
4
3
2
1
alternate school - no state rank
* more information at http://school-ratings.com