**Modeling single-family home prices in the Bay Area: (subtitle)**

Michael Boles

July 2019

****

*Photo credit: Unsplash (https://unsplash.com/photos/6b9rqGI\_w1s)*

**Abstract**

This article describes the collection, visualization, and modeling of single-family home prices currently listed across the San Francisco Bay Area. Complementing listing information (number of bedrooms and bathrooms, home size, and lot size) with location data (school quality and commute times) was found to significantly improve explanatory power of the model, and enabled the identification of undervalued listings and neighborhoods.

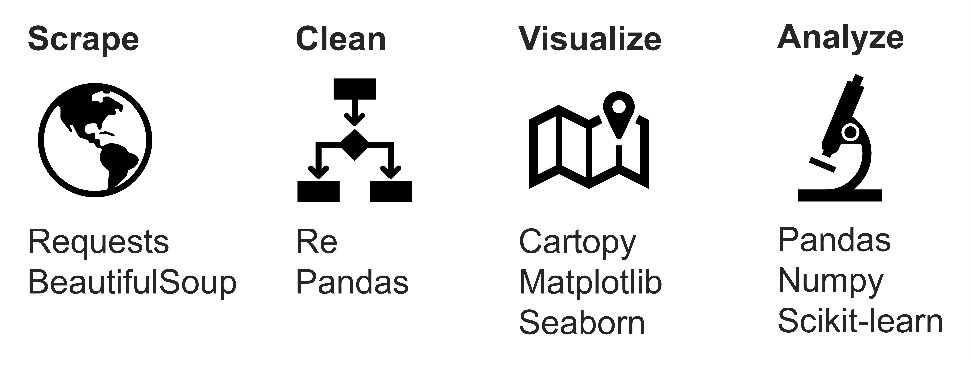
**Introduction**

While the performance of the S&P 500 and the overall US housing market has been nearly identical since 2000 (both up ~100%), home price indices in the San Francisco Bay Area have risen by approximately 167% (St. Louis Fed1). As such, Bay Area homeowners have enjoyed an opportunity to build wealth through real estate in a way that is not accessible to most of the rest of the country.

For those already bought into the market, this near-tripling of real estate values since 2000 has undoubtedly been a good thing. However, for those newly relocated to the region, saving towards a down payment and choosing where to buy can be a daunting task. Inspired by discussions I’ve had with friends and family, basic concepts in investing (i.e., buy undervalued assets), and a desire to hone my data science skillset, I set out to gather as much information about current prices of single-family homes in the Bay Area and apply machine learning techniques to tease out the most important factors driving home values.

**Methods**

Data from single family home listings (address, beds, baths, home size, lot size, latitude/longitude coordinates, and price) across the Bay Area was scraped in June 2019 from a real estate webpage (www.mlslistings.com) using the Requests and BeautifulSoup Python libraries, and cleaned and tabulated using Regex and Pandas. Complementing this listing data, commute times were obtained from Google Maps and school quality data pulled from the 2018 California Assessment of Student Performance and Progress (CAASPP). The resulting information was overlaid onto maps using Cartopy, Matplotlib, and shapefiles (town, zip code, and neighborhood borders) from Stanford Earthworks. Box/strip plots and pairwise relationships between variables were visualized using Seaborn. Ordinary least squares regression analysis was applied to the data using Statsmodels and scikit-learn libraries. The full source code for this project is available on my personal GitHub page (github.com/mboles01).

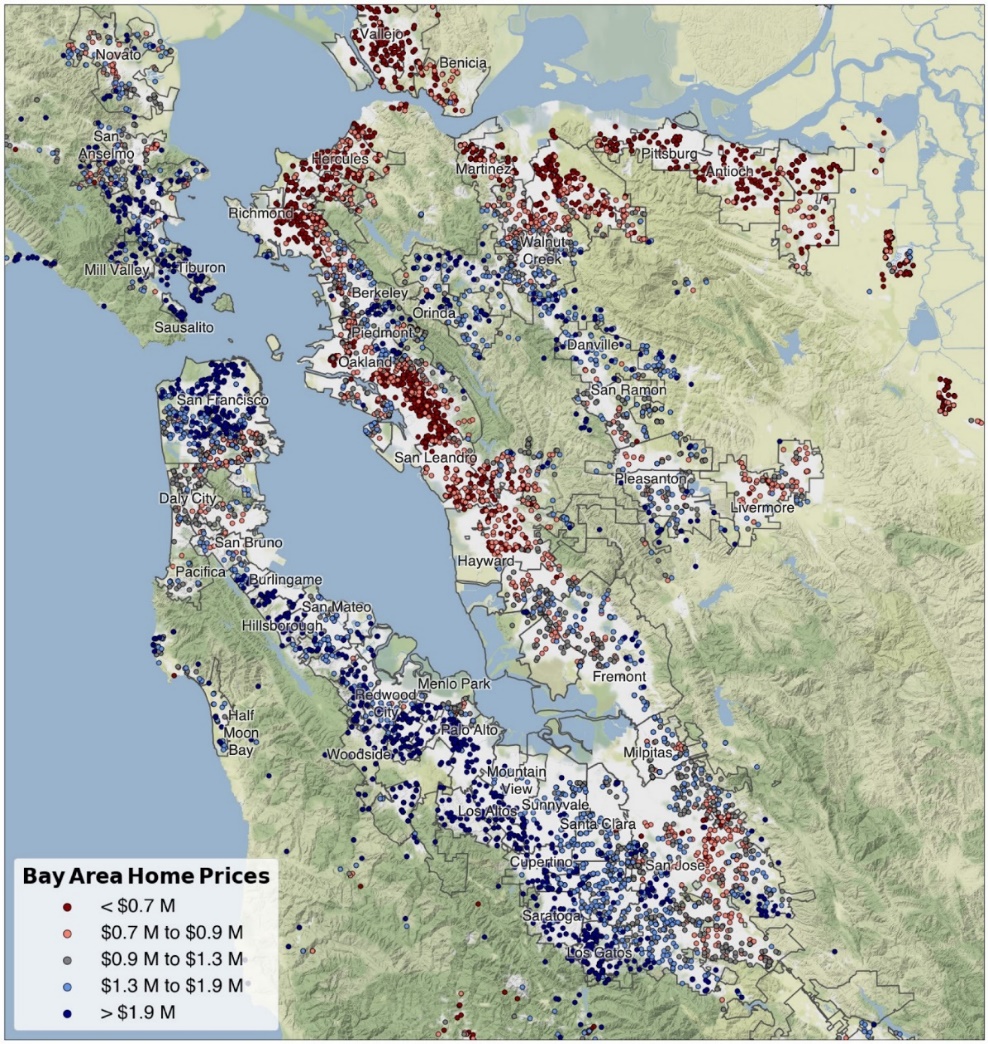


**Results**

This study began with the collection of listing data for all single family homes on a popular real estate webpage (www.mlslistings.com). Looping over a list of Bay Area zip codes, location details (address, latitude and longitude coordinates), property characteristics (number of bedrooms, number of bathrooms, home size, and lot size), and list price were scraped for each listing using the Python Requests and BeautifulSoup libraries (see source code at end of article). Across 214 zip codes, location, property, and price information was collected for 7151 listings in June 2019. The data was cleaned using regular expressions (re.findall(), re.sub() functions) to remove extra whitespace, unwanted characters, and entries with missing values.

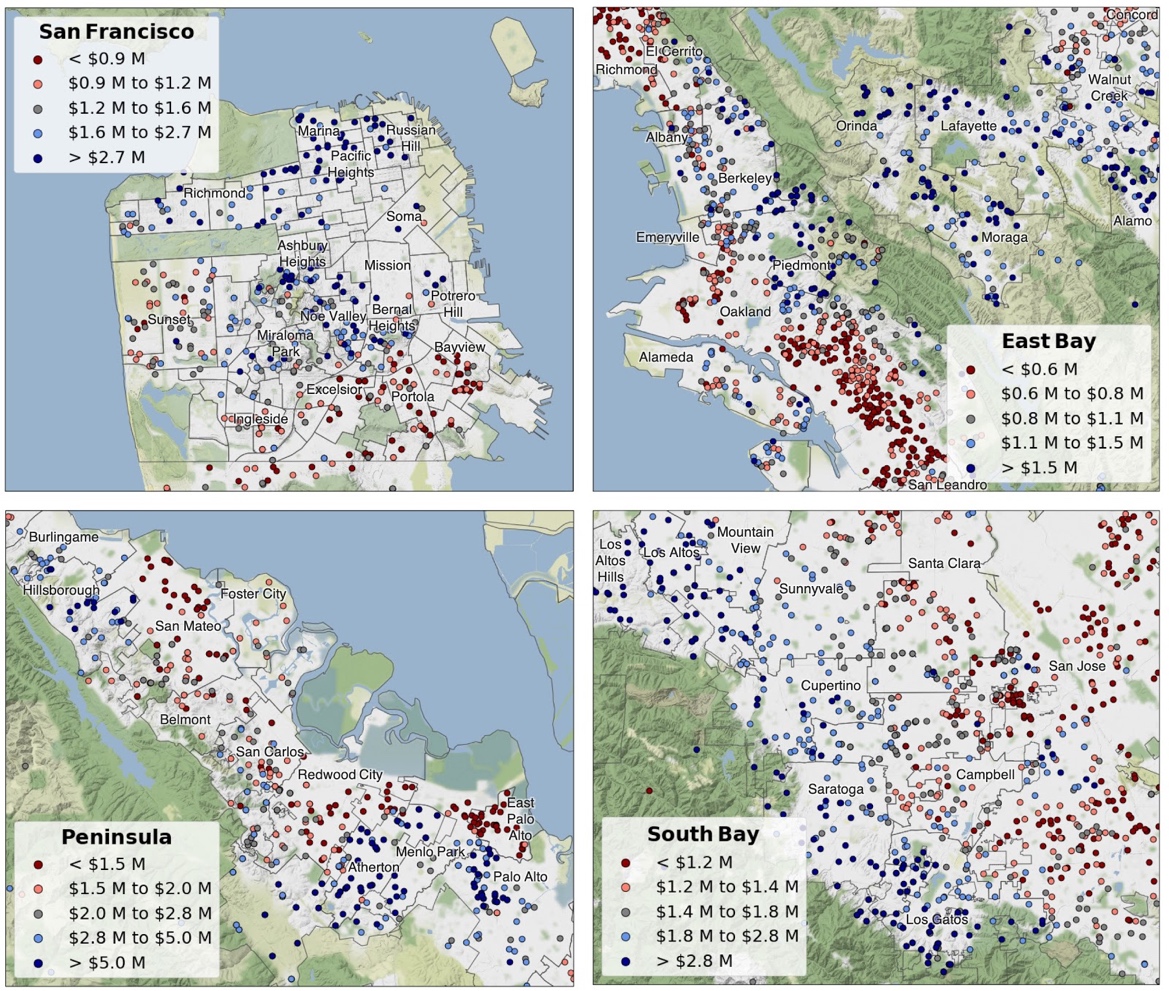
*I. Geographic trends*

With information for several thousand properties currently on the market across the Bay Area, the location of all listings was plotted on a terrain map and color-coded by price (Figure 1) using the Python Cartopy package together with city border information obtained from Stanford Earthworks. This map reveals clear geographic trends of the relative cost of homes across the region. For instance, San Francisco, Marin County, and the Peninsula typically contain the most expensive 20% of listings (dark blue data points), while those in Oakland, San Leandro, and Richmond are typically in the least expensive 20% (dark red data points). Similarly, in the South Bay, homes closer to the Santa Cruz Mountains are typically more expensive than those tucked into the Diablo Range.



**Figure 1.** Overview of single-family homes listed for sale in the Bay Area in June 2019. The 7153 entries are split into quintiles by price, with list prices falling within the bottom and top 20% colored red and blue, respectively.

A natural extension of this analysis involves splitting the listings into subregions. Zooming in on San Francisco, the East Bay, the Peninsula, and the South Bay, geographic price trends again emerge (Figure 2). The most expensive single-family home listings in San Francisco fall between downtown and the Presidio, while the least expensive are found in the southern portion of the city, from the Sunset to Bayview districts. In the East Bay, homes on either side of the Oakland Hills (including Piedmont, Berkeley, and Orinda) are the most expensive, while Richmond, South Oakland, and San Leandro are the least expensive. On the Peninsula, homes in Palo Alto, Atherton, and Hillsborough are the most expensive, while San Mateo and East Palo Alto are the least expensive. In the South Bay, homes closest to the Santa Cruz Mountains (Los Altos, Saratoga, and Los Gatos) are the most expensive, while nearly everything east of that is less expensive. Relative cost across regions is also interesting to note: while the top price quintile in the East Bay starts at $1.5 M, that same price point falls at the bottom of the scale on the Peninsula.

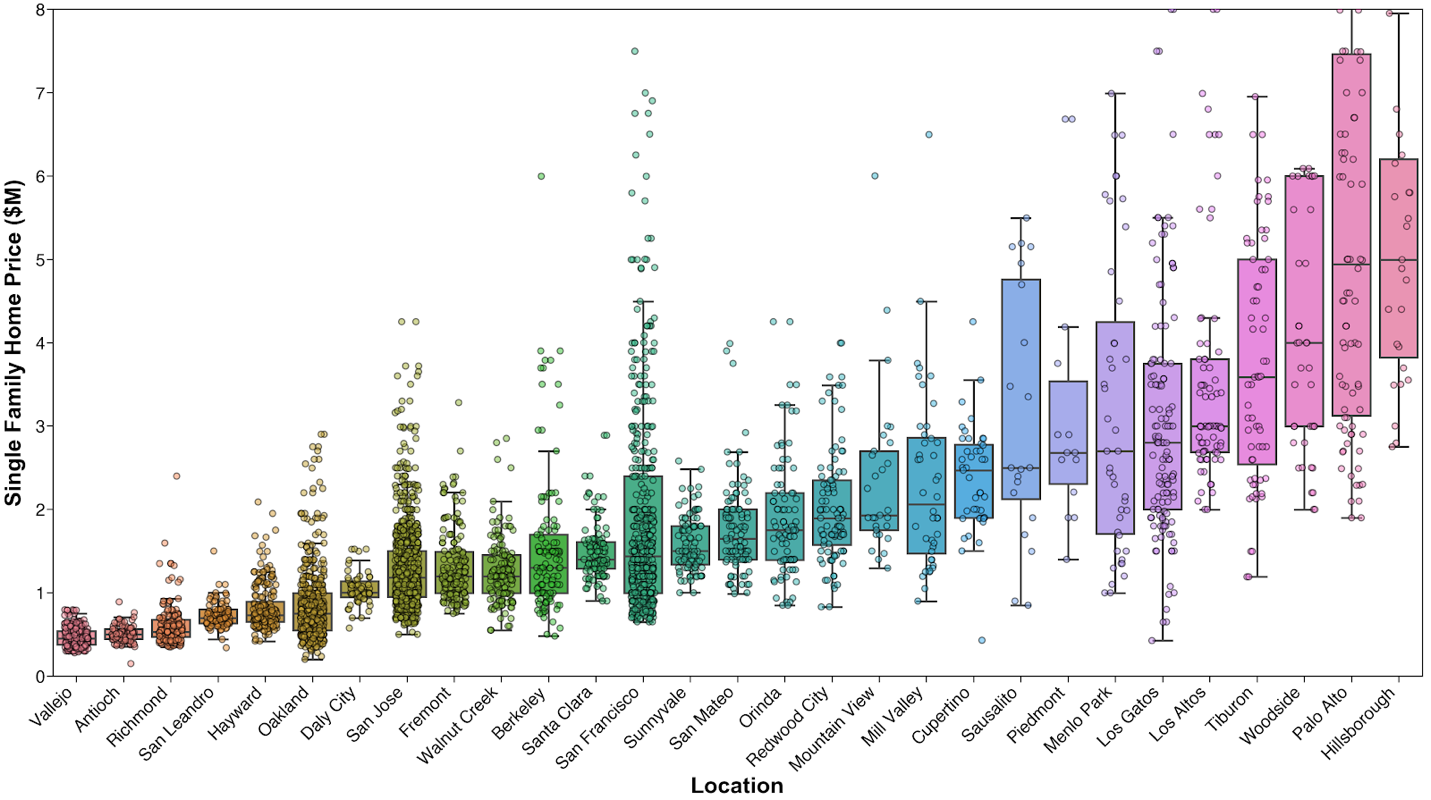


**Figure 2.** Zoom showing detail of single-family home list prices in the San Francisco, East Bay, Peninsula, and South Bay regions. In each case, price quintiles have been recalculated to reflect the distribution of prices within the highlighted region.

To better understand how prices vary across the Bay Area, box plots were constructed using the Python Seaborn library. In addition to the major cities (San Francisco, Oakland, and San Jose), several towns across the East Bay, Peninsula, and South Bay were included to illustrate the spread in single family home prices across the region.

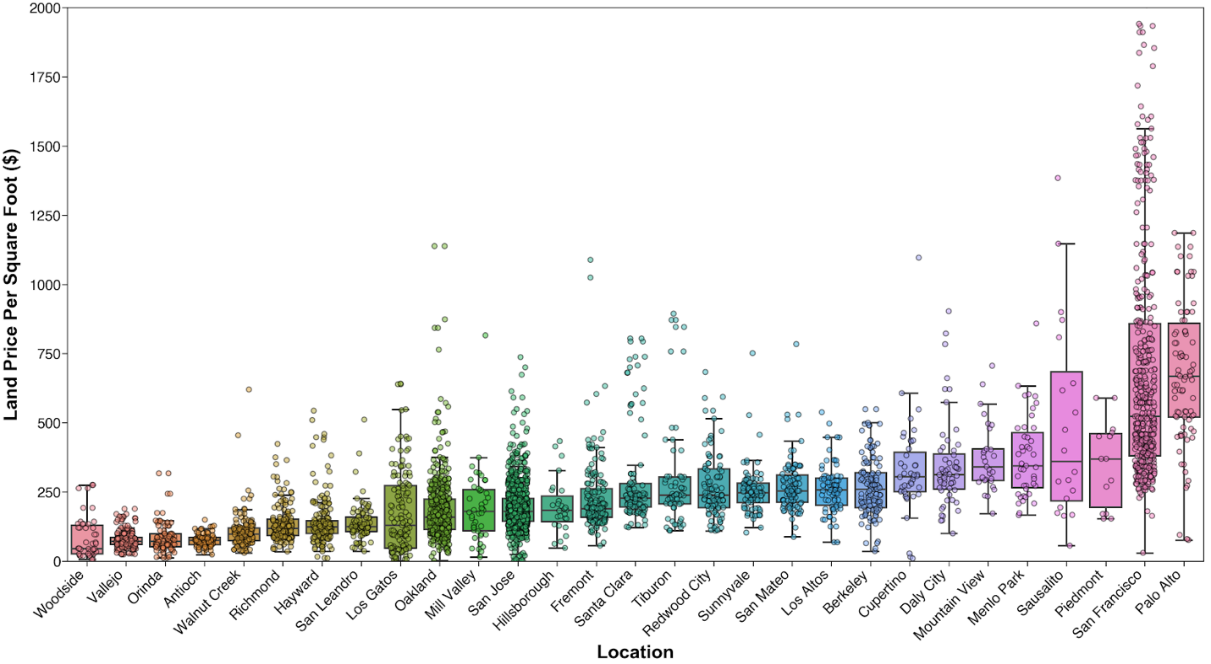
From this perspective, sorting localities from least- to most expensive median home price offers some interesting insights (Figure 3). For example, the lowest median home prices are largely found in the northeast Bay (Vallejo, Antioch, and Richmond), while the highest are found on the Peninsula (Hillsborough, Palo Alto, and Woodside). In addition, the overlay of scatter points on top of the boxplots speaks to where most people live:

Oakland, San Jose, and San Francisco have several hundred listings each, while the most expensive communities have just a handful of homes currently listed for sale.



**Figure 3.** Box plot displaying home price and for selected Bay Area cities, with individual observations superimposed to reveal sample size and distribution.

Interestingly, application of the same analysis to the price of the land the house sits on (Figure 4) leads to somewhat different conclusions. For instance, while the median home price in Woodside ranks third highest of the 29 cities and towns depicted in Figure 3, the median cost of land per unit area in this community (located mostly west of I-280 and known for horseback riding) is in fact the lowest of those 29. Similarly, other towns with high median home prices, such as Orinda and Los Gatos, appear to be more affordable from a land cost perspective. At the other end of the spectrum, Palo Alto and San Francisco are by far the most expensive places to buy land, reflecting their status as major hubs of economic activity in the region.



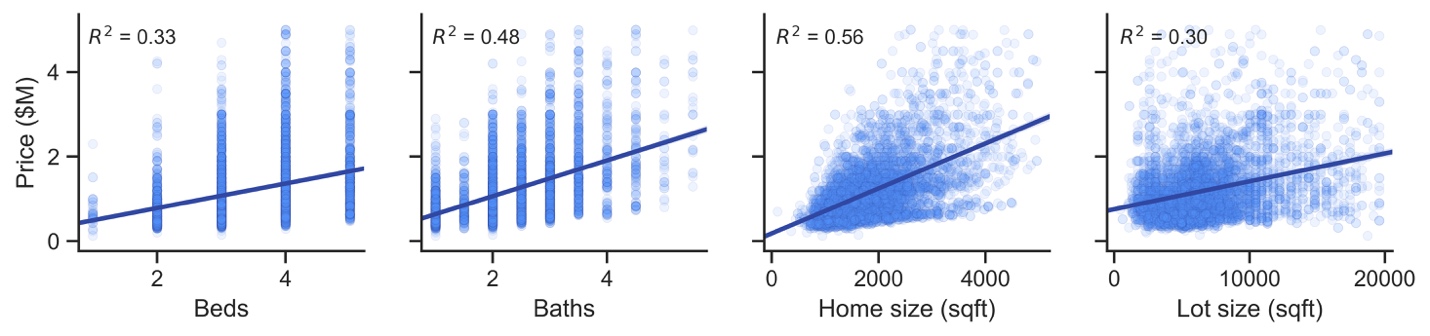
**Figure 4.** Box plot displaying land price for selected Bay Area cities.

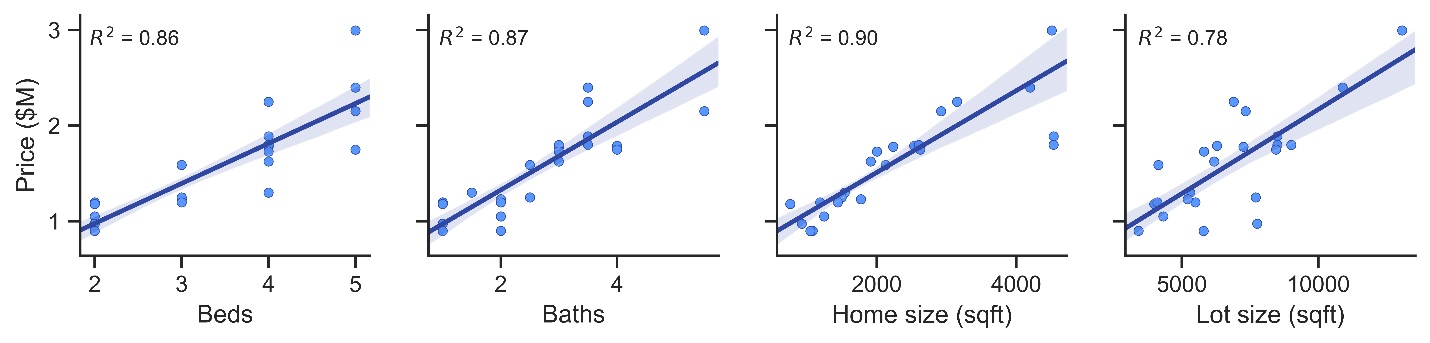
*II. Modeling home prices*

Beyond these observations concerning geographic distribution of Bay Area housing and land prices, the data set was used to model home prices using Python Statsmodels and Scikit-learn libraries.

To begin, prices of the 7151 homes in the data set were plotted individually against property data (number of bedrooms, number of bathrooms, home size, and lot size) and fitted using ordinary least squares (OLS) regression to assess pairwise correlations (Figure 5, top). Of these four features, the strongest individual predictor of list price is the home size (*R*2 = 0.56), while the weakest correlation was found to be lot size (*R*2 = 0.30).

On the other hand, when the full data set is narrowed to a single zip code (I chose my own: 95126), the correlation between these individual features and the home price is much stronger (Figure 5, bottom). Home size is again the strongest individual predictor of price, and explains 90% of the variation within a single zip code.

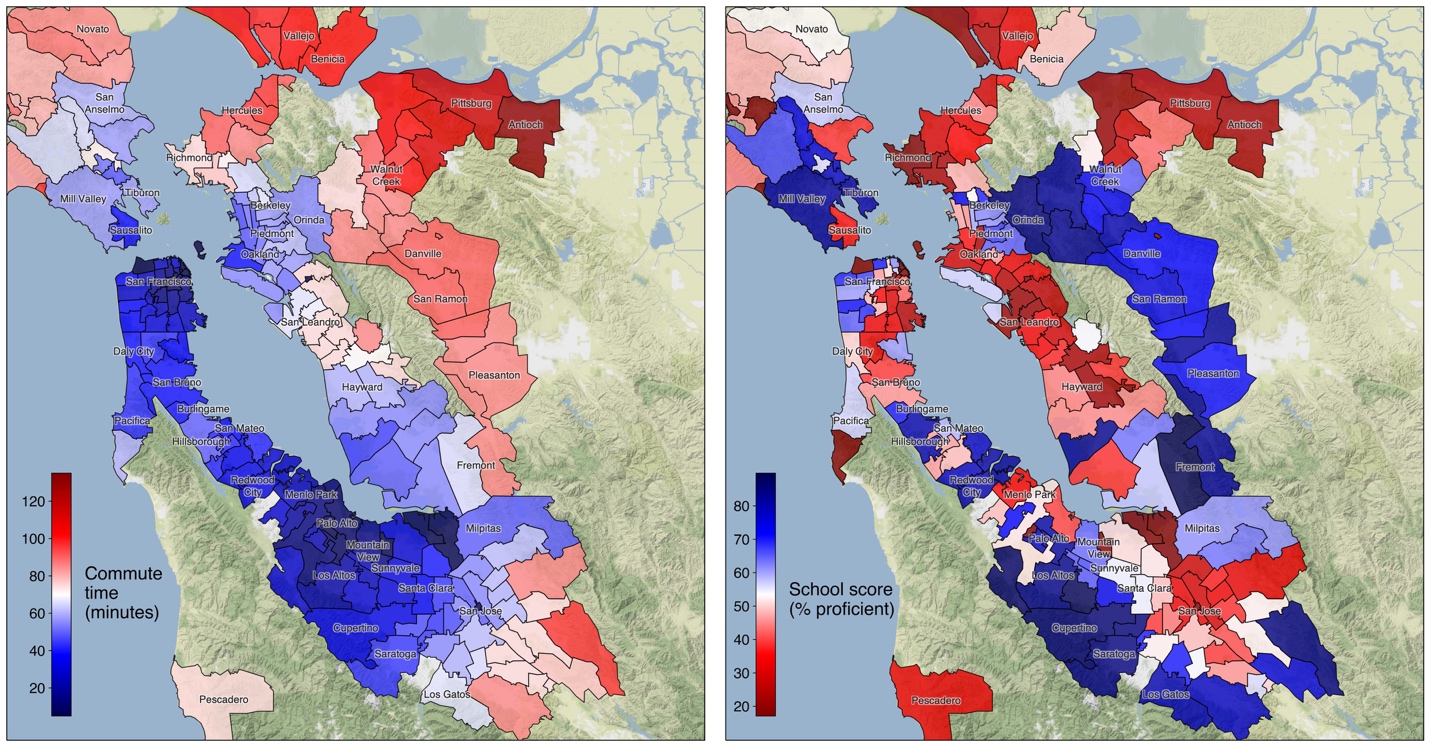




**Figure 5**. Correlation between price and listing factors is improved when full set of listings (top) is narrowed to a single zip code (bottom).

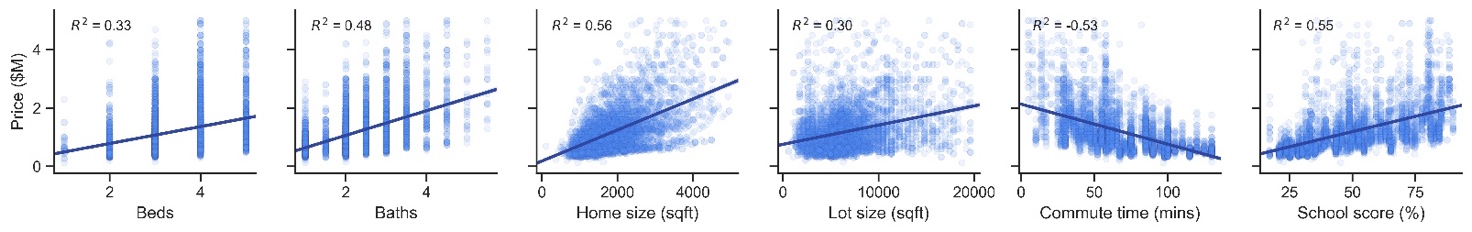
The modest correlation between home size and price over the entire region, and strong correlation within a single zip code, suggests that location-specific factors also contribute to price. To bring “place” into the equation, additional data reflecting convenience and privilege was introduced: commute time and public school quality.

Recognizing San Francisco and Palo Alto as the two primary economic centers of the region, the commute time was measured via Google Maps as travel time by car (at 8am on Wednesday) from each zip code to the closer of the two destinations. The result indicates that homes along the Peninsula, from San Francisco to San Jose, offer the possibility of commute times under one hour (Figure 6, left). Additionally, Marin and Oakland have similar access to San Francisco, as does Fremont to Palo Alto. On the other hand, commute times to the nearer hub are often more than 1.5 hours from homes in the outer East Bay (Richmond, Antioch, San Ramon).



**Figure 6**. Commute times (left) and school quality (right) for zip codes across the Bay Area.

New parameters commute time, school score are significantly related to home price



New parameters are not related to previous set – home, lot size. Three variables are however related – beds, baths, and home size (sqft). Because home size is most strongly correlated (R2 = 0.56) with price, beds and baths were discarded from the OLS model.

**Results**

a. Maps describe home price trends across the region (Figures 1,2)

b. Box/strip plots enable ranking cities by cost of house, cost of land (Figure 3)

c. Plots of list price vs. listing data (beds, baths, home size, lot size) show weak positive correlations that narrow upon zooming into one zip code (Figure 4)

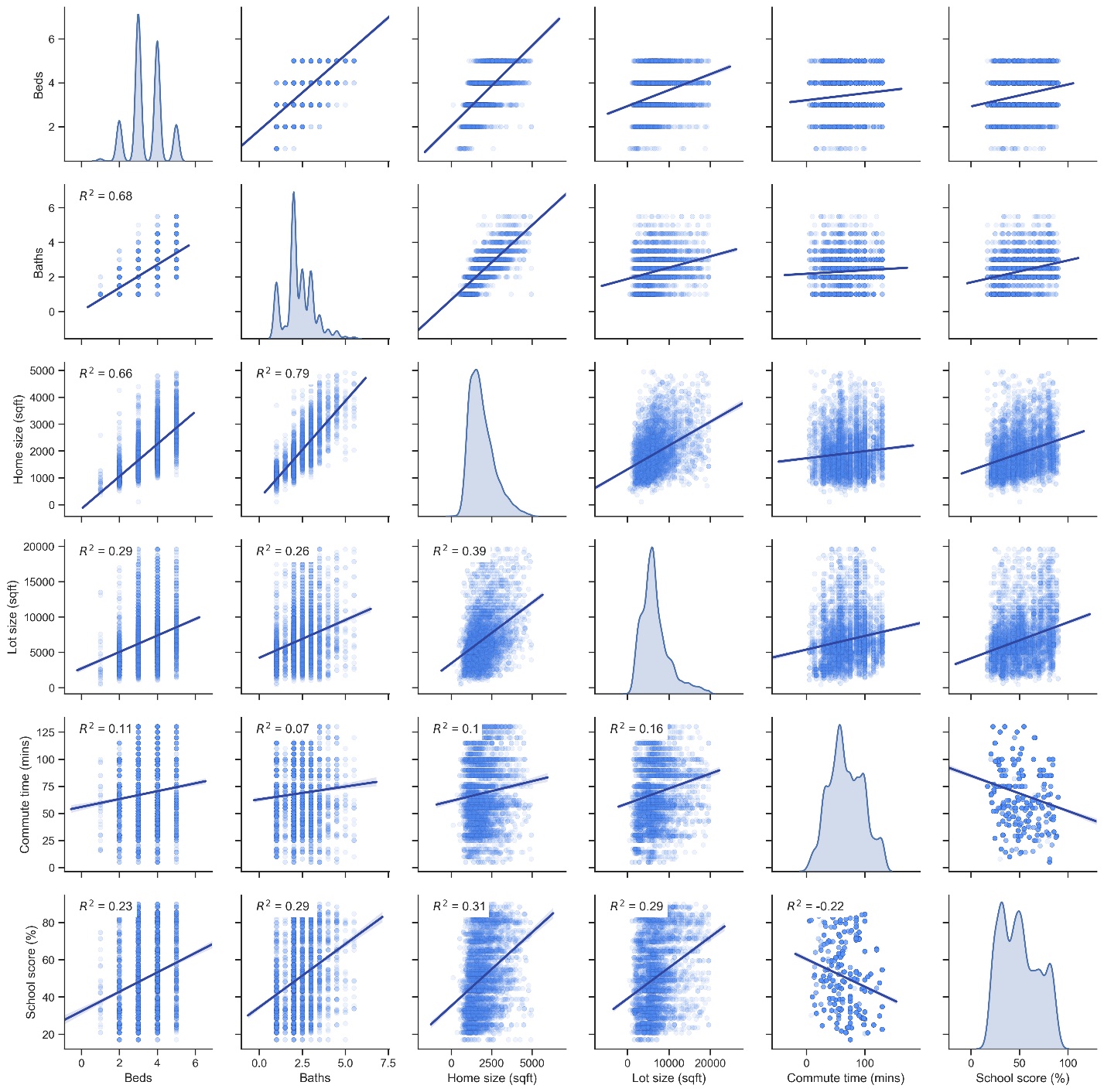
d. Aiming to capture other aspects of “place”, collected data on commute times, school quality, and crime rate (Figure 5?) – rationale for incorporating such metrics

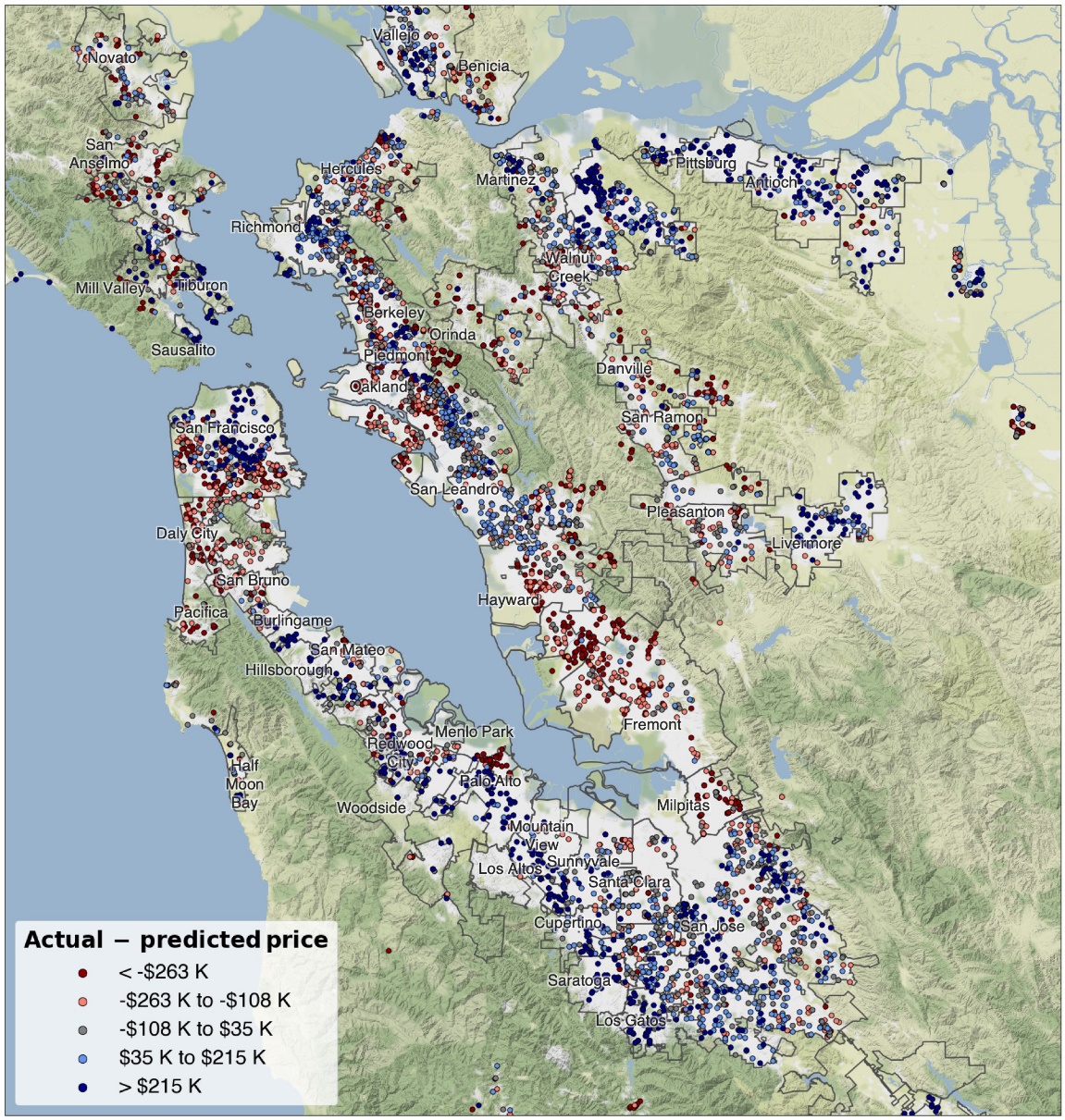
e. Ran linear regression fit on the full data set, evaluating the effect (coefficients) and statistical significance (P-values) of the inputs to devise a model for home prices (Figure 6? Equation 1?)

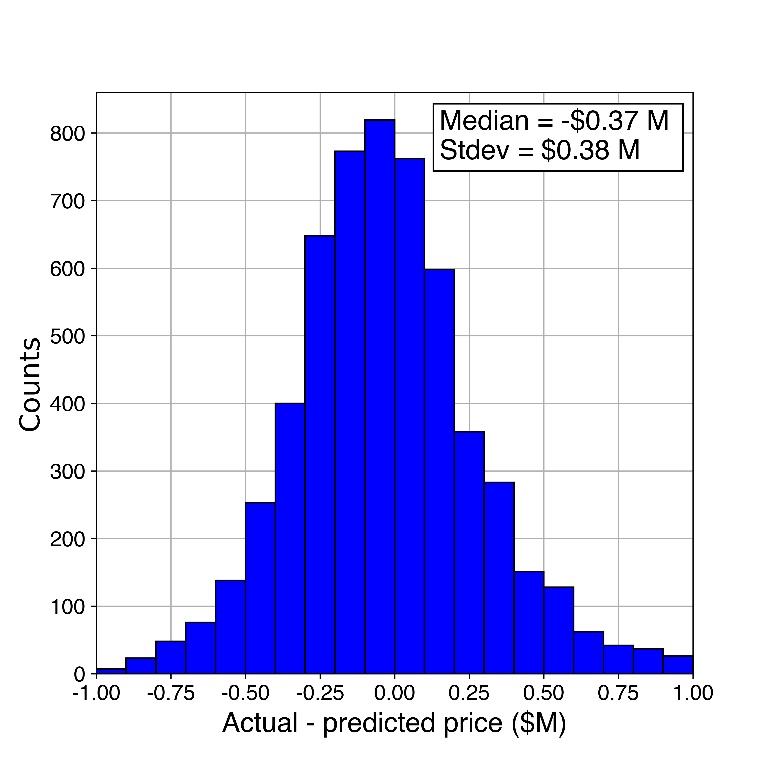
f. Calculated the difference between predicted and actual list prices, used this to identify potentially undervalued homes (Figure 7? – histogram of *P*pred – *P*list, Table 1 – 10 undervalued, 10 overvalued houses?)

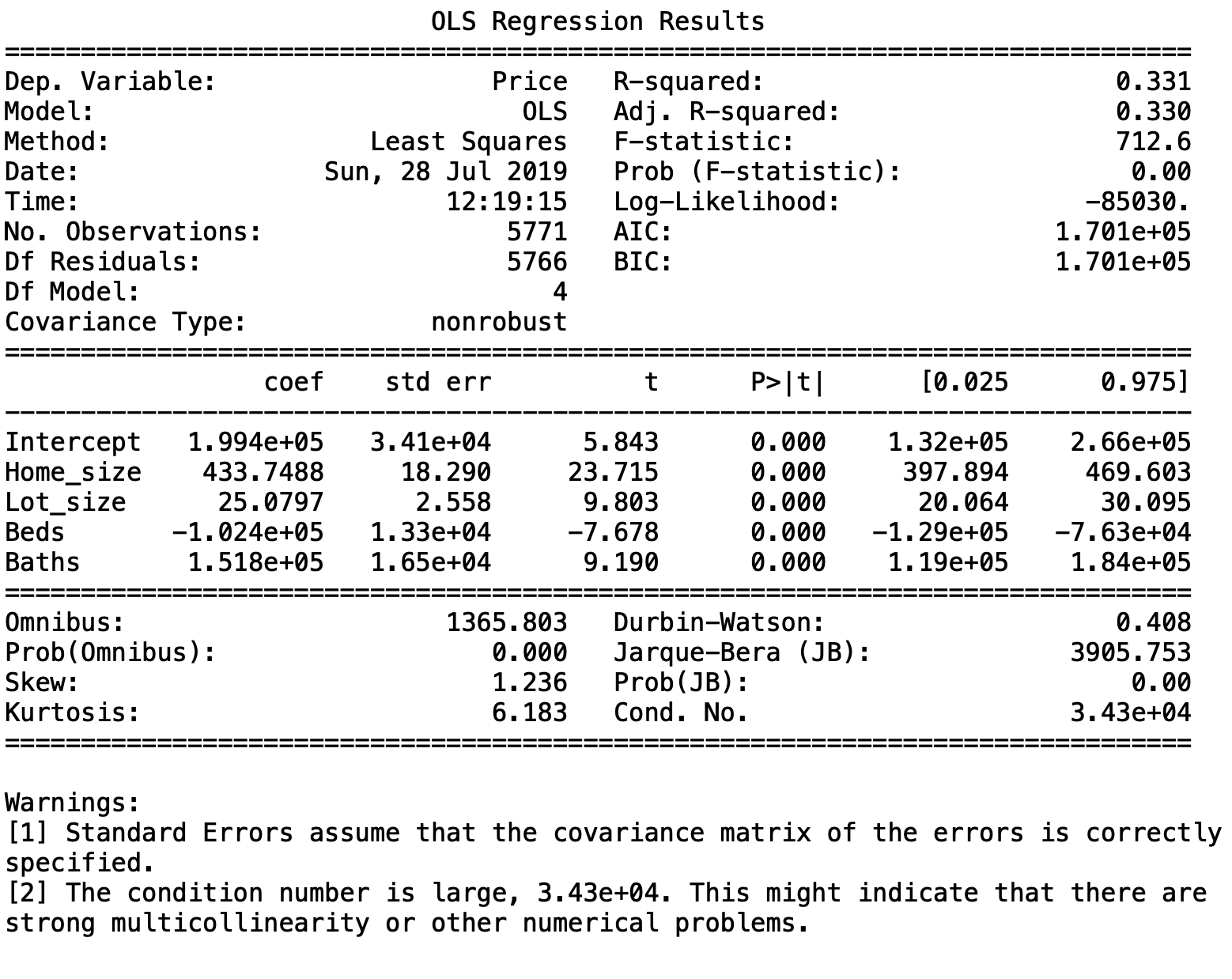
**Conclusions**

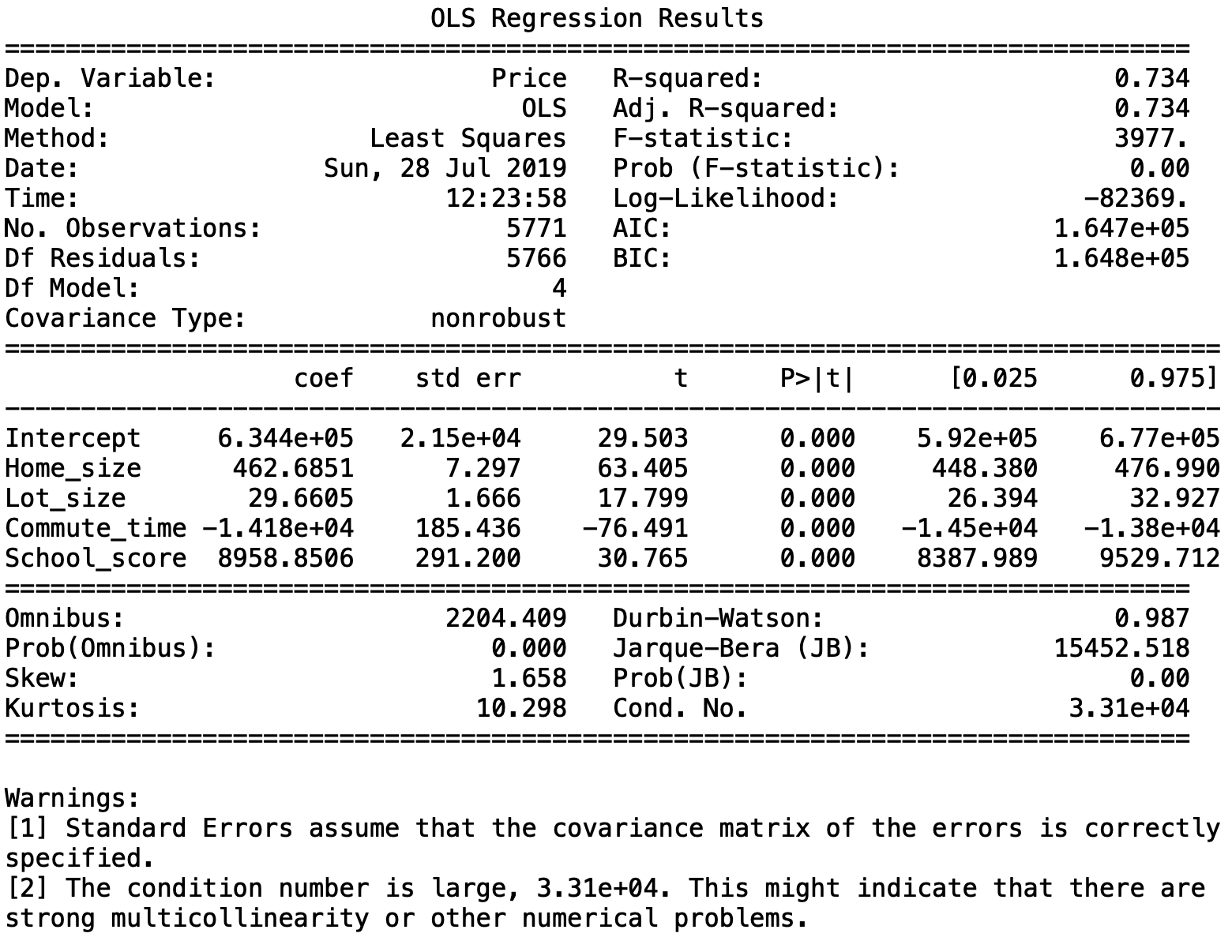
The application of data science techniques to inform real estate investment decisions need not be only pursued by corporate investors – free, open-source packages for use with Python empower the individual to scrape webpages, visualize data, apply machine learning to identify deals that may be overlooked by other market participants.











**References**

1Federal Reserve Bank of St. Louis (https://fred.stlouisfed.org/series/SFXRSA)

Shapefiles: Stanford Earthworks (https://earthworks.stanford.edu/catalog/stanford-vj593xs7263)

**Source code**

*Web scraping*

Mozilla Firefox developer tools was used to identify xpaths for items to be scraped and key-value pairs for data dictionary in the session.post() request

# import modules

from bs4 import BeautifulSoup

from lxml import html

import requests

import pandas as pd

import time

def webscrape(zipcodes):

# create empty data frame

data\_all = pd.DataFrame()

for counter, zipcode in enumerate(zipcodes,1):

# get homepage session

session = requests.Session()

homepage = session.get('https://www.mlslistings.com/')

soup = BeautifulSoup(homepage.content, "html.parser")

# get security token, post search data

token = soup.find("input", attrs={"name" : "\_\_RequestVerificationToken"})['value']

data = {'transactionType': 'buy', 'listing\_status': 'Active', 'searchTextType': '', 'searchText': zipcode,'\_\_RequestVerificationToken': token, 'property\_type': 'SingleFamilyResidence'}

search\_results = session.post("https://www.mlslistings.com/Search/ResultPost", data=data)

tree = html.fromstring(search\_results.content)

# update status

print('Scraping data for zipcode (%s/%s): ' % (counter,len(zipcodes)) + str(zipcode))

# scrape desired information

address = list(map(str, tree.xpath('//a[@class="search-nav-link"]//text()')))

price = list(map(str, tree.xpath('//span[@class="font-weight-bold listing-price d-block pull-left pr-25"]//text()')))

beds = list(map(str, tree.xpath('//span[@class="listing-info-item font-size-sm line-height-base d-block pull-left pr-50 listing-beds"]//text()')))

baths = list(map(str, tree.xpath('//span[@class="listing-info-item font-size-sm line-height-base d-block pull-left pr-50 listing-baths"]//text()')))

homesize = list(map(str, tree.xpath('//span[@class="font-weight-bold info-item-value d-block pull-left pr-25"]//text()')))

lot = list(map(str, tree.xpath('//span[@class="listing-info-item font-size-sm line-height-base d-block pull-left pr-50 listing-lot-size"]//text()')))

yearbuilt = list(map(str, tree.xpath('//span[@class="listing-info-item font-size-sm line-height-base d-block pull-left pr-50 listing-sqft last"]//text()')))

# create data frame from scraped, cleaned data

data\_temp = {'Address': address, 'City': city, 'Zip': zip\_code,

'Beds': beds, 'Baths': baths, 'Home size': homesize,

'Lot size': lot, 'Year built': yearbuilt, 'Garage': garage,

'Home type': hometype, 'Price': price}

dataframe\_temp = pd.DataFrame(data\_temp)

data\_all = data\_all.append(dataframe\_temp)

print('Zipcode %s was skipped' % zipcode)

# wait, then scrape next zipcode

time.sleep(1)

return data\_all