CS176: Housing Prices vs. Unemployment Trends

Ziming Jerry Wang
wang7353@purdue.edu

Jaylin Eduardo Hernandez
hernje02@purdue.edu

Teeshan Darshan Mevada
tmevada@purdue.edu

November 2025

0.1 Introduction

In our study of housing prices and unemployment in California, we analyze how the housing market and labor market interact. With our carefully chosen datasets, we can get a good overview of the overall situation and examine whether rising unemployment is related to falling house prices and whether these trends differ across counties and prices. Our research seeks to answer questions such as:

- Do house prices decline during high unemployment?
- Does low unemployment correlate with faster home sales?
- Are expensive houses more resistant to labor market issues?

We will examine the extent of these interactions through the filtration of data by different unemployment thresholds, prices, seasons, and/or demographcis. We will be using multiple different visualizations to show how the California housing market works.

0.2 Data Selection

0.2.1 Description

The purpose of our project is to from multiple perspectives analyze the Housing Market and its economical context.

0.2.2 Datasets

1. **FHFA House Price Indexes** [1]
2. **Home Mortgage Disclosure Act 2007 - 2017** [2]
3. **Local Area Unemployment Statistics** [3]
4. **US Housing Trends** [4]

FHFA House Price Indexes

Public collection of house price indexes from the Federal Housing Finance Agency covering all **50** states and over **400** American cities. The data provide monthly and quarterly indexes dating back to the 1970s that measure changes in single-family home values.
Key Fields (Updated Monthly):

- Region
- Date
- Index Value

Table size: 131541

Local Area Unemployment Statistics

This dataset contains monthly estimates of employment, unemployment, and labor force participation for around 7,600 countries, cities, and metropolitian areas. It uses data from surveys and state unemployment insurance.
Key Fields:

- Date
- Location

This dataset is valuable after merging to examine the correlation between unemployment and housing prices and/or mortgage. It can be valuable information to graph.
Alternatively, we can merge this set with the Bank-Marketing dataset to examine consumer behavior vs employment.
It is easy to pivot first because of its .csv format, also because it has time valued columns. It is easy to deal with missing values and is easy to use with mathplotlib.

US Housing Trends

Derived from Zillow's research data. Shows housing market trends from 2018 to May of 2024. Some data includes median home values, average days on market, and price reduction. Field Columns also include date.

This dataset naturally matches with the FHFA HPI. When merged on location or time, they can be compared and analyzed easily. It can also be combined with the HMDA easily as all sets have common rows (mostly of time and location).

0.2.3 Correlation Between Data

All 6 datasets share common fields (Time and Location), which are key fields for analyzing the Housing Market, Economic Activity, and their correlations. Mortgage and Unemployment information can be joined to understand how the labor market influences loans. FHFA and Zillow can be easily merged as Zillow complements FHFA. Then the data can be used to analyze multiple areas. For example, merging with Bank Marketing can help analyze the impact of bank loans during different highs and lows of the housing market. Merging also allows for tracking the different long-term price trends and market cycles.

0.2.4 Choice of Data

These datasets were carefully chosen and made sure that they are able to integrate. Each dataset is provided in a tabular format(either .csv, .xslx, .xsl, etc.), which is easily read when using the pandas library for Python. After reading the files, we can also merge and clean by key values easily. Then, all the information can be sorted and/or subgrouped to provide a complete analysis from multiple directions. We can pivot the data to summarize trends, and then we can use matplotlib to visualize all of this data.
As for the choice of topic, we spent time searching in for datasets in multiple areas (9 total). These areas ranged from our interests (Ex. the video game sales and popularity) to recent or current issues (Covid 19 deaths, enviromental issues, housing, etc.). We finally decided on a topic of working on the Housing Market on press approval (basically everyone wanted to do this). We started with 3 datasets and each person added a dataset for a total of six in the case that some of the data overlaps too much or is unreliable.

0.3 Data Wrangling and Explanations

0.3.1 Code and Information

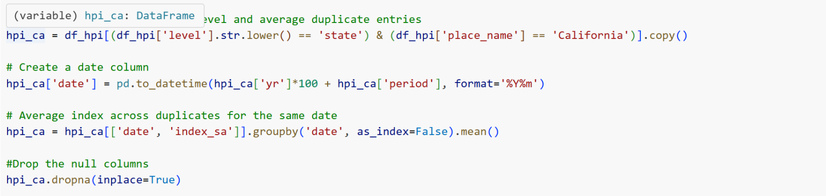
Our Code [5]

0.3.2 Cleaning the Data

Before starting to subset/merge/visualize the data, we must first tidy it up for easier usage. The following section gives a short description on how we processed our data.

FHFA HPI

Overall, this dataset was pretty clean to work with. However, there were still a few things that we had to work on.



```
(variable) hpi_ca: DataFrame evel and average duplicate entries
hpi_ca = df_hpi[(df_hpi['level'].str.lower() == 'state') & (df_hpi['place_name'] == 'California')].copy()

# Create a date column
hpi_ca['date'] = pd.to_datetime(hpi_ca['yr']*100 + hpi_ca['period'], format='%Y%m')

# Average index across duplicates for the same date
hpi_ca = hpi_ca[['date', 'index_sa']].groupby('date', as_index=False).mean()

#Drop the null columns
hpi_ca.dropna(inplace=True)
```

Figure 1: Cleaning process for FHFA HPI

- **Cleaning Column Names**

Column names such as **place_name** were renamed to **region** for better understanding. Columns such as **yr** or **period** were renamed so that they are consistent between all datasets

- **Missing Values**

  This included a seasonally adjusted index. However, not every row had a seasonally adjusted index. In fact, more than 10000 rows had no seasonally adjusted values. As such, we had to either drop the entire row or fill in the values. We chose to drop the rows with no seasonally adjusted values using dropna()

- **Multiple values per month**

  This dataset had multiple values per month, which when sorting on date proves to be a tremendous issue. We grouped by date and then averaged the values for each month, which solved this problem.

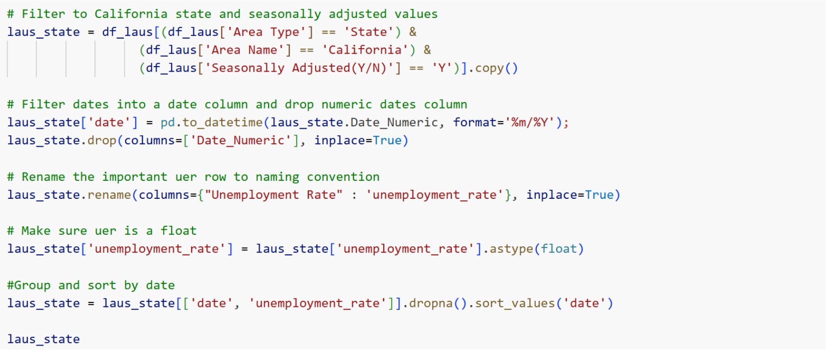
- **Date**

  The **year** and **month** columns were merged together into a new date column and then removed from the dataset. We used a to_datetime function then a to_period function as there is no actual date. Only the month and year values existed.

Here is the final result after cleaning the data and column names:

LAUS

This dataset's naming convention is different from the one used in our project so we started with renaming the columns. Here is a detailed breakdown of what was done for this dataset.



```
# Filter to California state and seasonally adjusted values
laus_state = df_laus[(df_laus['Area Type'] == 'State') &
                  (df_laus['Area Name'] == 'California') &
                  (df_laus['Seasonally Adjusted(Y/N)'] == 'Y')].copy()

# Filter dates into a date column and drop numeric dates column
laus_state['date'] = pd.to_datetime(laus_state.Date_Numeric, format='%m/%Y');
laus_state.drop(columns=['Date_Numeric'], inplace=True)

# Rename the important uer row to naming convention
laus_state.rename(columns={"Unemployment Rate" : 'unemployment_rate'}, inplace=True)

# Make sure uer is a float
laus_state['unemployment_rate'] = laus_state['unemployment_rate'].astype(float)

#Group and sort by date
laus_state = laus_state[['date', 'unemployment_rate']].dropna().sort_values('date')

laus_state
```

Figure 2: Cleaning for the LAUS dataset

- **Cleaning Column Names**

  We are using a naming protocol of all lowercase and an underscore between words. We adjusted each column name to be in this structure.

- **Removing Columns**

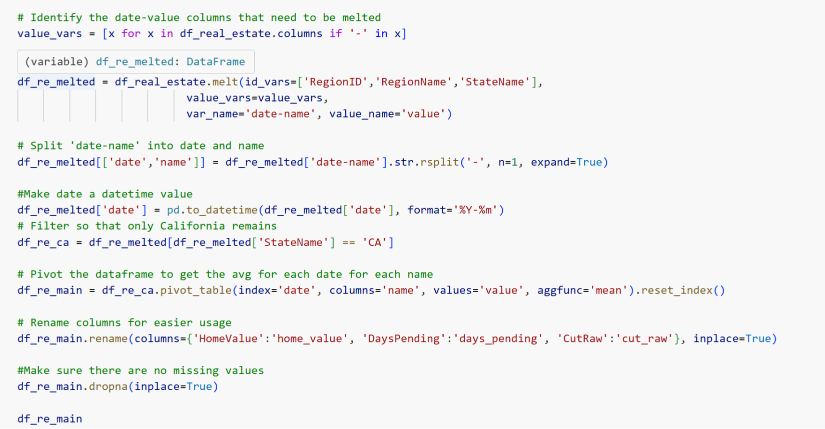
  We removed columns that had no impact on our analysis such as the **Benchmark** time, **Status** (We filtered by final status to remove preliminary data and removed the status column entirely), and year and month as we merged them into a date column.

- **Date**

  We parsed a date from the year and month then removed the year and month columns

Housing Trends

This dataset was harder to clean. We had to first unpivot parts of the dataframe (as it had pivoted information) using multiple melts and then merge, as well as rename the columns to our naming convention. Below is a detailed breakdown of the processing.



```
# Identify the date-value columns that need to be melted
value_vars = [x for x in df_real_estate.columns if '-' in x]

df_re_melted = df_real_estate.melt(id_vars=['RegionID','RegionName','StateName'],
                          value_vars=value_vars,
                          var_name='date-name', value_name='value')

# Split 'date-name' into date and name
df_re_melted[['date','name']] = df_re_melted['date-name'].str.rsplit('-', n=1, expand=True)

#Make date a datetime value
df_re_melted['date'] = pd.to_datetime(df_re_melted['date'], format='%Y-%m')
# Filter so that only California remains
df_re_ca = df_re_melted[df_re_melted['StateName'] == 'CA']

# Pivot the dataframe to get the avg for each date for each name
df_re_main = df_re_ca.pivot_table(index='date', columns='name', values='value', aggfunc='mean').reset_index()

# Rename columns for easier usage
df_re_main.rename(columns={'HomeValue':'home_value', 'DaysPending':'days_pending', 'CutRaw':'cut_raw'}, inplace=True)

#Make sure there are no missing values
df_re_main.dropna(inplace=True)

df_re_main
```

Figure 3: Cleaning done for the real estate data

- **Renaming Columns**

  We renamed each column that do not have to be unpivoted to a lowercase and underscore format using .rename()

- **Unpivoting Data**

  Our data had column names such as **2018-02-HomeValue** and **2018-02-DaysPending**. We used rename to rename these columns to only the date string (2018-02) separately for each set of info (home value, days pending, etc.). We melted these columns then used a right split to format them into the correct format.

- **Dates**

After unpivoting the data, we formatted the date column (actually nothing changed) using datetime and period.

- **Data Cleanup**

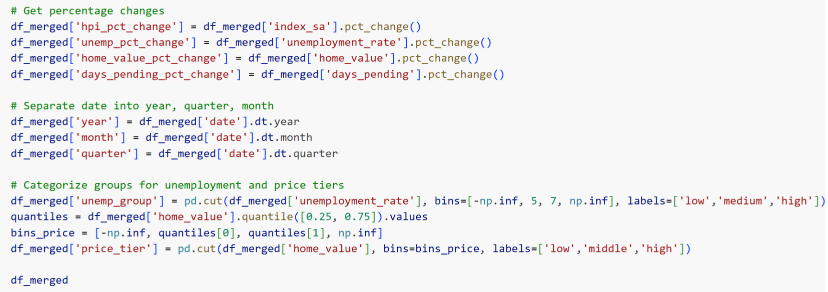
  We then rounded data with long decimals to 2 after the decimal. Finally, we dealt with the missing states with fillna and then dropped the columns with remaining null values.

0.3.3 Data Merging

The datasets all have a date field in common, so all three were joined on that field. The fhfa hpi originally had multiple values for each month. These were dealt with by averaging the seasonally adjusted values when grouping by date. This left us with a single value per month. The real estate data was melted when cleaning the data, and the date was processed so that we can merge on it. Then we joined the three on date.

0.3.4 Filtering Subsets

We made multiple theories on how the data could be subsetted and filtered, but we only visualized a few of them.

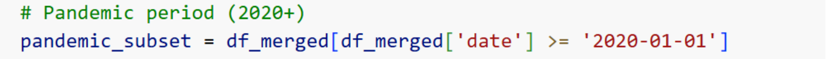


```python
# Get percentage changes
df_merged['hpi_pct_change'] = df_merged['index_sa'].pct_change()
df_merged['unemp_pct_change'] = df_merged['unemployment_rate'].pct_change()
df_merged['home_value_pct_change'] = df_merged['home_value'].pct_change()
df_merged['days_pending_pct_change'] = df_merged['days_pending'].pct_change()

# Separate date into year, quarter, month
df_merged['year'] = df_merged['date'].dt.year
df_merged['month'] = df_merged['date'].dt.month
df_merged['quarter'] = df_merged['date'].dt.quarter

# Categorize groups for unemployment and price tiers
df_merged['unemp_group'] = pd.cut(df_merged['unemployment_rate'], bins=[-np.inf, 5, 7, np.inf], labels=['low','medium','high'])
quantiles = df_merged['home_value'].quantile([0.25, 0.75]).values
bins_price = [-np.inf, quantiles[0], quantiles[1], np.inf]
df_merged['price_tier'] = pd.cut(df_merged['home_value'], bins=bins_price, labels=['low','middle','high'])

df_merged
```

Figure 4: Preprocessing to Filter Data

- **Pandemic Period (2020+)**

  We made a subset focused on the time from Jan 2020 and onward to explore the impact of the pandemic. During this period, California's mean unemployment rate was 6.08% while HPI averaged around 1.94% and home values grew 1.63%per month. Despite high unemployment, home prices still grew quickly.

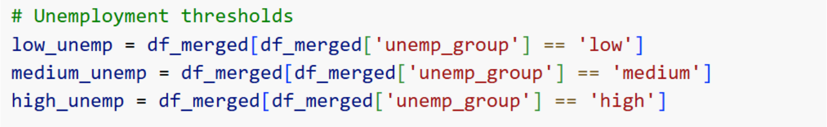


```python
# Pandemic period (2020+)
pandemic_subset = df_merged[df_merged['date'] >= '2020-01-01']
```

Figure 5: Pandemic Filter

- **Unemployment Thresholds**

  We partitioned the data into low, medium, and high unemployment (¡5, 5-7, 7+) to explore how different sets of unemployment rates differed. The lower unemployment rates had the lowest home value growth, while the highest unemployment rates corresponded to the highest hpi growths and high home value growths. The highest home value growth was actually during the middle period, which we infer were due to factors outside of what we cover. The five highest months were also actually those during the pandemic, which is interesting as given the pandemic, the rate of home values should be increasing drastically.

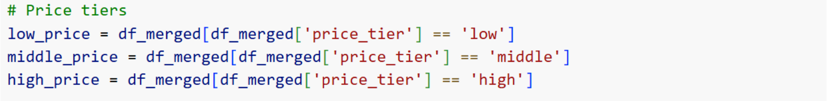


```
# Unemployment thresholds
low_unemp = df_merged[df_merged['unemp_group'] == 'low']
medium_unemp = df_merged[df_merged['unemp_group'] == 'medium']
high_unemp = df_merged[df_merged['unemp_group'] == 'high']
```

Figure 6: Unemployment Threshold Filter

- **Price tiers**

  We grouped the data into low, medium, and high price tiers (grouped by percentile). Mid-tier showed the strongest price although the average unemployment rate was 6.77%. The low and high price months grew slower at around 0.9% and 0.7% respectively.

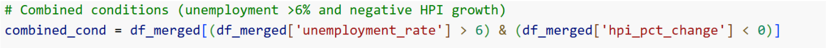


```
# Price tiers
low_price = df_merged[df_merged['price_tier'] == 'low']
middle_price = df_merged[df_merged['price_tier'] == 'middle']
high_price = df_merged[df_merged['price_tier'] == 'high']
```

Figure 7: Price Tiers Filter

- **Combined Conditions**

  We tried to make a set where the monthly unemployment was greater than 6% and the HPI change was negative. However, in the data we have, there was no month that met both criteria. HPI growth always remained positive, even during the pandemic.

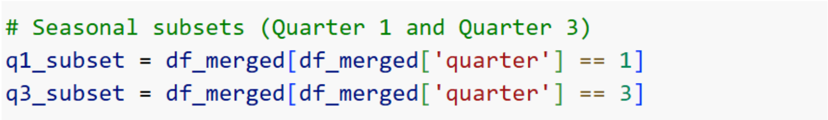


```
# Combined conditions (unemployment >6% and negative HPI growth)
combined_cond = df_merged[(df_merged['unemployment_rate'] > 6) & (df_merged['hpi_pct_change'] < 0)]
```

Figure 8: Combined Conditions Filter

- **Seasonal Subsetting**

  We made a subset by seasons(quarters of year) for the data. Due to missing data, the Q3 observations were incomplete.

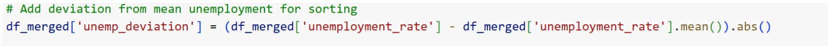


```
# Seasonal subsets (Quarter 1 and Quarter 3)
q1_subset = df_merged[df_merged['quarter'] == 1]
q3_subset = df_merged[df_merged['quarter'] == 3]
```

Figure 9: Seasonal Subsets Filter

0.3.5 Sorting data

After filtering a few possible subsets, we went on to sort the data



```
# Add deviation from mean unemployment for sorting
df_merged['unemp_deviation'] = (df_merged['unemployment_rate'] - df_merged['unemployment_rate'].mean()).abs()
```

Figure 10: Preprocessing for sorting data

- **Date descending**

  We sorted the data by a descending date to explore more recent months. By spring of 2024 the HPI reached around 427 and median home values were around 616k while unemployment was around 5.2%. This shows a snapshot of market conditions at the time.

- **Home value descending**

  We sorted the data by home value descending to find when the largest home values occurred. It was seen that it was between April of 2022 and 2024, showing when the market peaked in price. With a unemployment of around 5%, this shows how there was strong demand even though the labor market was cooling down.

- **UE descending**

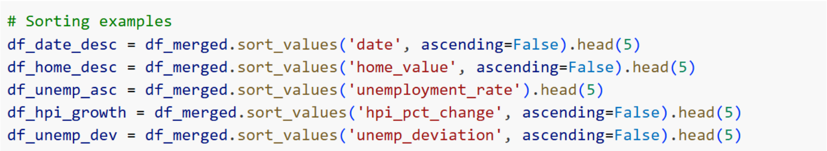
  This was interesting as the lowest unemployment rate was in 2018-2019, which was before the pandemic. These values corresponded to the lower home values and lower hpi, showing how prices were lower and steady before the recent after pandemic prices.

- **HPI growth descending**

  The largest hpi growth was seen 2020 and 2021 when the unemployment was the highest and during the pandemic. The pandemic seemed to escalate the market. We will give some detailed explanations below.

- **Unemployment deviation from mean**

  This set was sort by the abs difference between unemployment rate per month and the mean. April 2020 had the largest deviation, but hpi and house prices steadily grew, which is very unusual.



```
# Sorting examples
df_date_desc = df_merged.sort_values('date', ascending=False).head(5)
df_home_desc = df_merged.sort_values('home_value', ascending=False).head(5)
df_unemp_asc = df_merged.sort_values('unemployment_rate').head(5)
df_hpi_growth = df_merged.sort_values('hpi_pct_change', ascending=False).head(5)
df_unemp_dev = df_merged.sort_values('unemp_deviation', ascending=False).head(5)
```

Figure 11: All sorts

0.3.6 Visualizations

- **HPI and unemployment over time**

  Dual line plot that shows the HPI index climbing from 274 in early 2018 to 426 in 2024 while unemployment spiked in mid 2020 and gradually receded. Plotting these two together was actually a hard decision to make. The hpi's scale is very different from the unemployment scale by a significant amount. However, we thought it would be beneficial to be able to see the spike in unemployment and its correlation with the hpi.

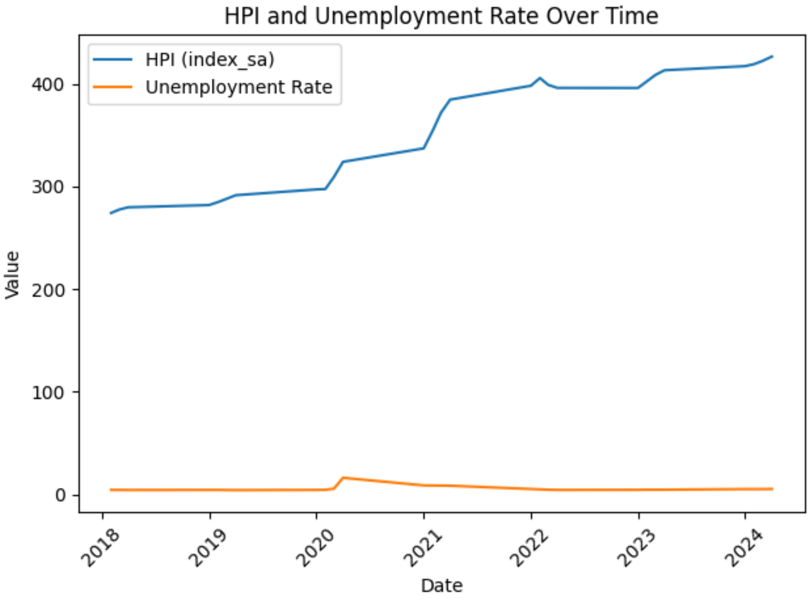


Figure 12: HPI and Unemployment rate over time

- **Unemployment vs. home value**

  Scatter plot that plots median home values against unemployment rate.

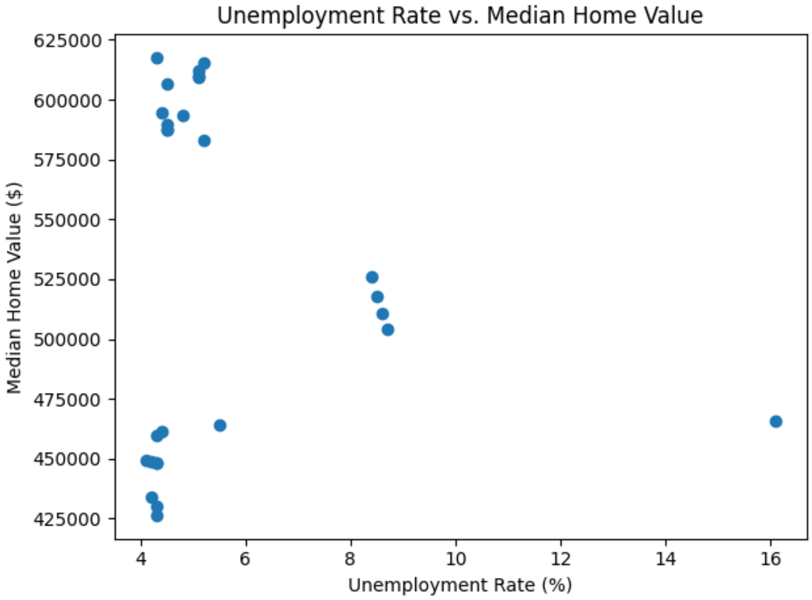


Figure 13: Scatter of Unemployment rate and home value

- **Distribution of Monthly HPI growth**

  A histogram of hpi percent change that shows that most price changes monthly fall between 0 and 2%.

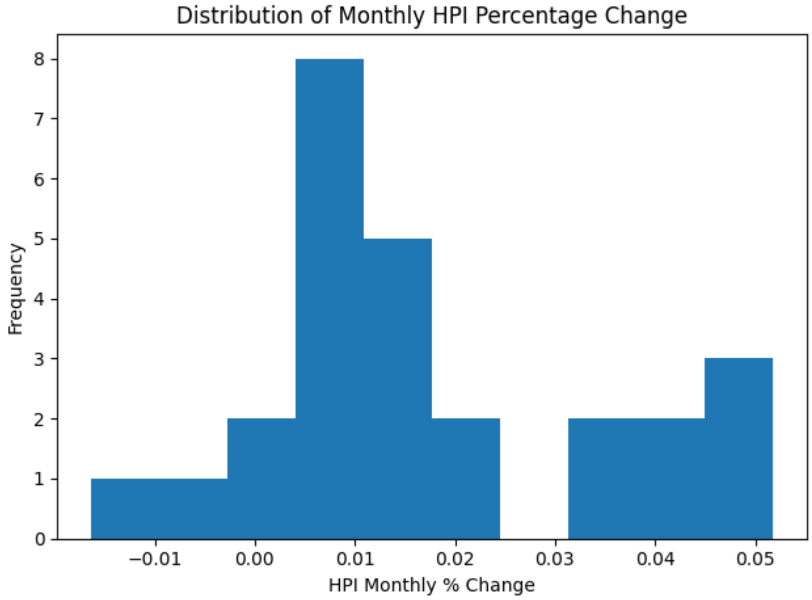


Figure 14: Histogram of HPI % change

- **Home value by unemployment group**

  A box plot that compares median home values across low medium and high unemployment groups.

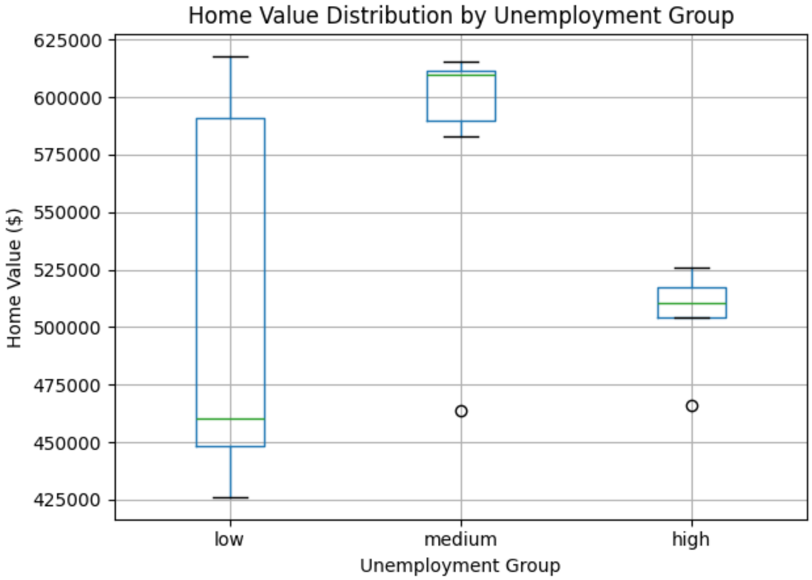


Figure 15: Home value by unemployment group

- **Heatmap of average home values**

  Heatmap by year and quarter, We can see that data has gaps in this heat map, but it shows the increase in home values quite clearly from 428k in 2018 Q1 to 610k 2024 Q2.

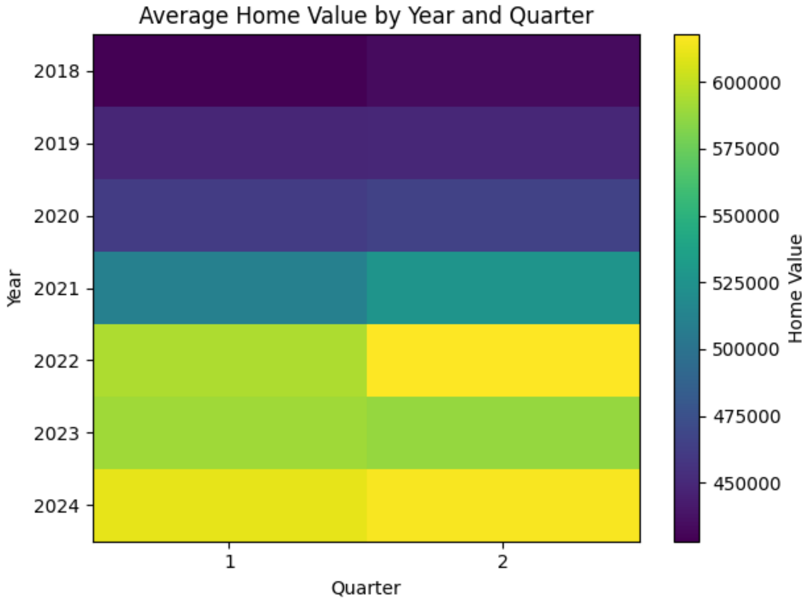


Figure 16: Heatmap of average home values

These visualizations show different complementary perspectives such as trends over time, relationships between variables, etc.

0.3.7 Pivots of the data

We did a few different pivots on the data as such:

- **Average home value by year/quarter**

  A pivot that displays the mean home value for each Y-Q combination. Average California median home value rose from 428k to 610k from 2018 Q1 to 2024 Q2. (Seen also in the heatmap above)

- **Avg. Unemployment Rate by y/q**

  Summary of unemployment rates by year and quarter. only full rows were 2019 q1-2 and 2024 q1-2

- **Avg days pending by unemployment group and price tier** A pivot that crosses days pending with unemployment group and price tiers. Low unemployment combined with high prices are the lowest days on market and medium unemployment with mid tier prices shows longer times. This suggests that high price homes sell faster when unemployment is low.

- **Multi index pivot stacking/unstacking**

Organized home values by year, unemployment group and price tier, creating a multi-index table. Stacking and unstacking the unemployment-group level changes the table to view price tiers across years for each labor-market category. (P.S. this is actually only here to show our group is comfortable using these manipulation processes)

- **Count of records by unemployment groups and price tier**

  Counts the number of months that fall into each unemployment and price tier combination. There are seven low(ue)/low(price), 6 low/mid, 3 low/high. High unemployment rate only occurred in mid price.

0.4 Analysis

First of all, a few patterns can be seen in this research:

1. **House Prices grew steadily** Even during pandemic spike during the pandemic, house prices grew steadily. The almost 0 (0.02) correlation between hpi and unemployment rate was quite interesting, showing that during this time the unemployment rate did not have an impact on housing prices. However, there was a strong correlation between HPI change and unemployment rate, showing that the hpi grew the quickest in the time of high unemployment (0.71).

2. **Weak link between unemployment and price levels** The scatter plot shows a correlation of (-0.11) which shows that prices do not fall when unemployment rises. Other factors may have more influence on the market.

3. **Difference across price tiers** We can see a distinct difference of behavior between price tiers when filter by them. High priced homes seemed to be sold quickest in low unemployment times.

These findings all indicate that the relationship between unemployment and housing prices are actually quite mild. Factors such as interest rate and the government passing out money during the pandemic may be more important factors that affect housing prices. However, it was interesting how hpi actually grew quicker in times of high unemployment.

In our graph of hpi against unemployment rate, we actually wanted to show this comparison. As we can see in 12, the unemployment spiked around the beginning of 2020. What was interesting was that housing prices continued to increase with a steeper slope than before. A spike in unemployment actually made the growth of the hpi spike as well. However, though there was a spike in hpi growth, it eventually leveled. In combination with 14 and 16, the housing prices tend to grow a little every month, with little tendency to actually drop. This shows a relatively stable increase in price, so although prices grew for a while with a steeper slope, the overall growth was not explosive.

Our final conclusion is that the extent to which unemployment rate can affect the housing market is limited. There are a few patterns that we can find within the data and correlations. For example, as in 15 and 13 we can see that in times of lower unemployment, there tends to be a wider range of housing available with both higher and lower prices. As unemployment rises, housing prices tend to contract and become more in the middle to low range. We can also see a correlation with unemployment and growth of hpi, where higher unemployment contributed to a higher hpi change.

To return to our questions:

1. House Prices do not decline during high unemployment. In fact, they do the exact opposite.
2. Low unemployment do in fact have a correlation with faster home sales.
3. Expensive houses are actually less resistant when the unemployment rate is high.

Bibliography

[1] Federal Housing Finance Agency. *FHFA House Price Indexes*. Data.gov. Feb. 12, 2025. URL: https://catalog.data.gov/dataset/fhfa-house-price-indexes-hpis-948c6.

[2] Consumer Financial Protection Bureau. *Home Mortgage Disclosure Act (HMDA) Public Data from 2007-2017*. Data.gov. Aug. 16, 2024. URL: https://catalog.data.gov/dataset/home-mortgage-disclosure-act-hmda-public-data-from-2007-2017#:~:text=Metadata%20Updated%3A%20August%2016%2C%202024.

[3] California Employment Development Department. *Local Area Unemployment Statistics (LAUS)*. Data.gov. Oct. 23, 2025. URL: https://catalog.data.gov/dataset/local-area-unemployment-statistics-laus#:~:text=Local%20Area%20Unemployment%20Statistics%20.

[4] Clovis Vieira. *US Housing Trends: Values, Time & Price Cuts*. Kaggle. July 1, 2024. URL: https://www.kaggle.com/datasets/clovisdalmolinvieira/us-housing-trends-values-time-and-price-cuts/data.

[5] Ziming Jerry Wang, Jaylin Eduardo Hernandez, and Teeshan Darshan Mevada. *CS176 Materials and Code*. Github. Dec. 7, 2025. URL: https://github.com/MouHuoZheDeRen/CS176Project.