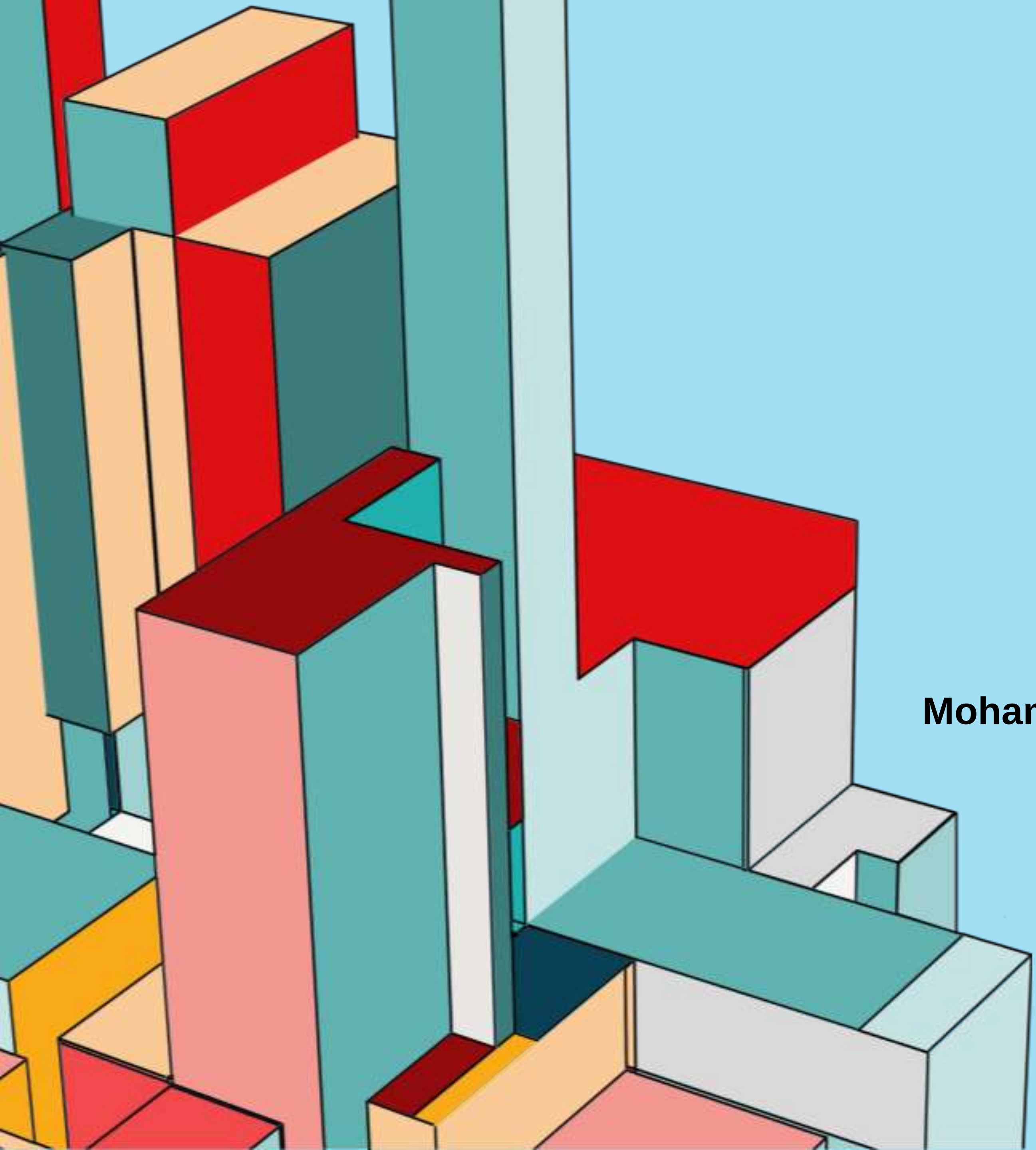


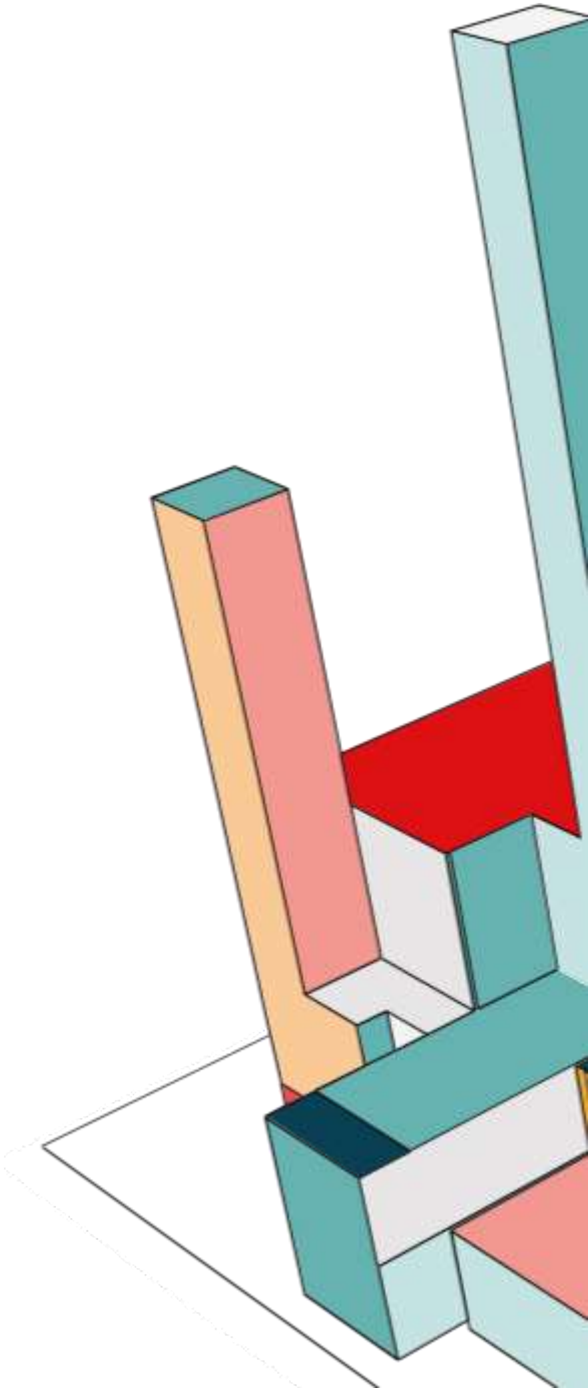
ZILLOW DATA ANALYSIS: RECENTLY SOLD HOMES IN LOS ANGELES, CA (LAST 90 DAYS)

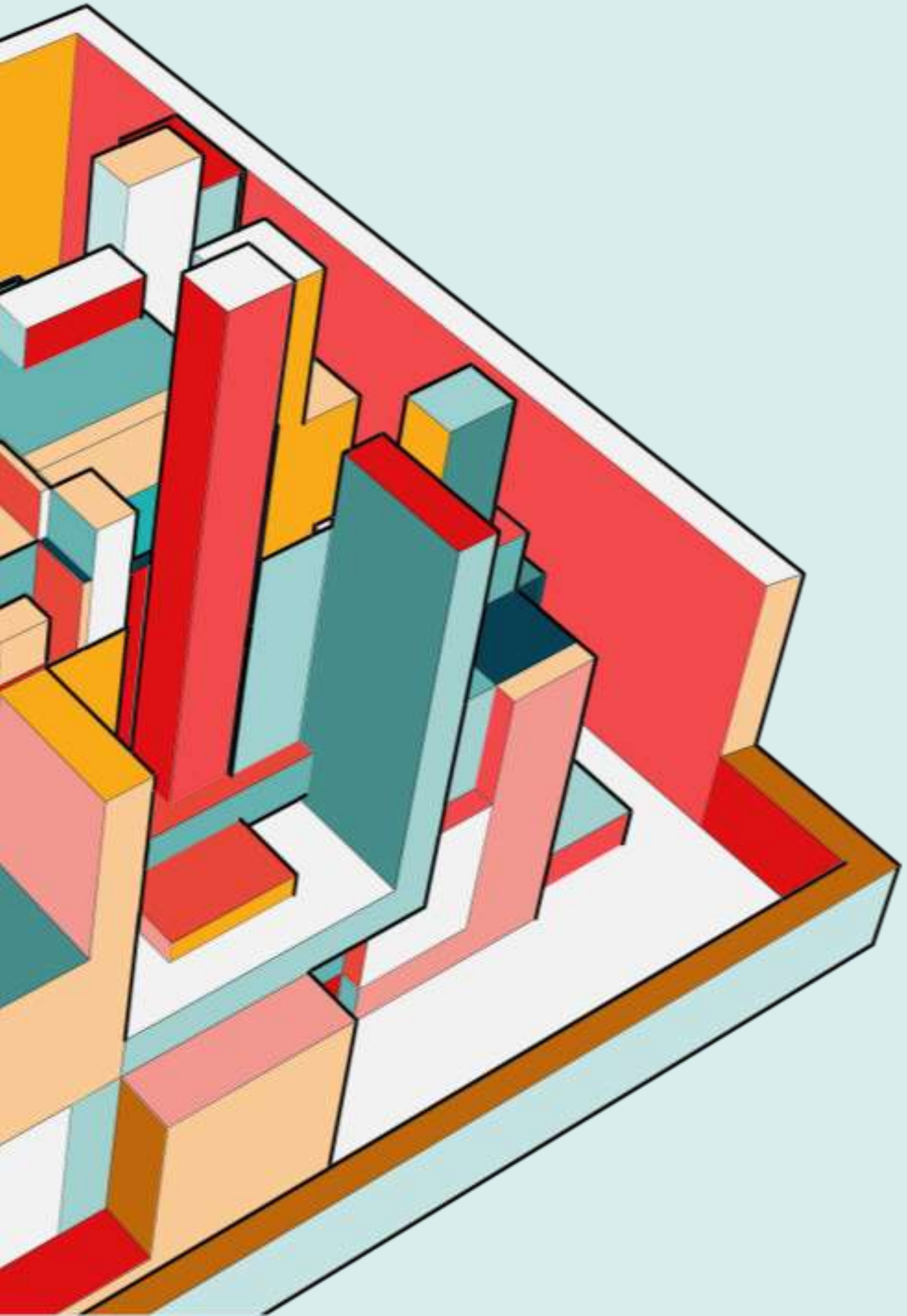
Mohammad Amin Yousefi - Ali Hamzehpour - Mina Shirazi

-Spring 2024-

AGENDA

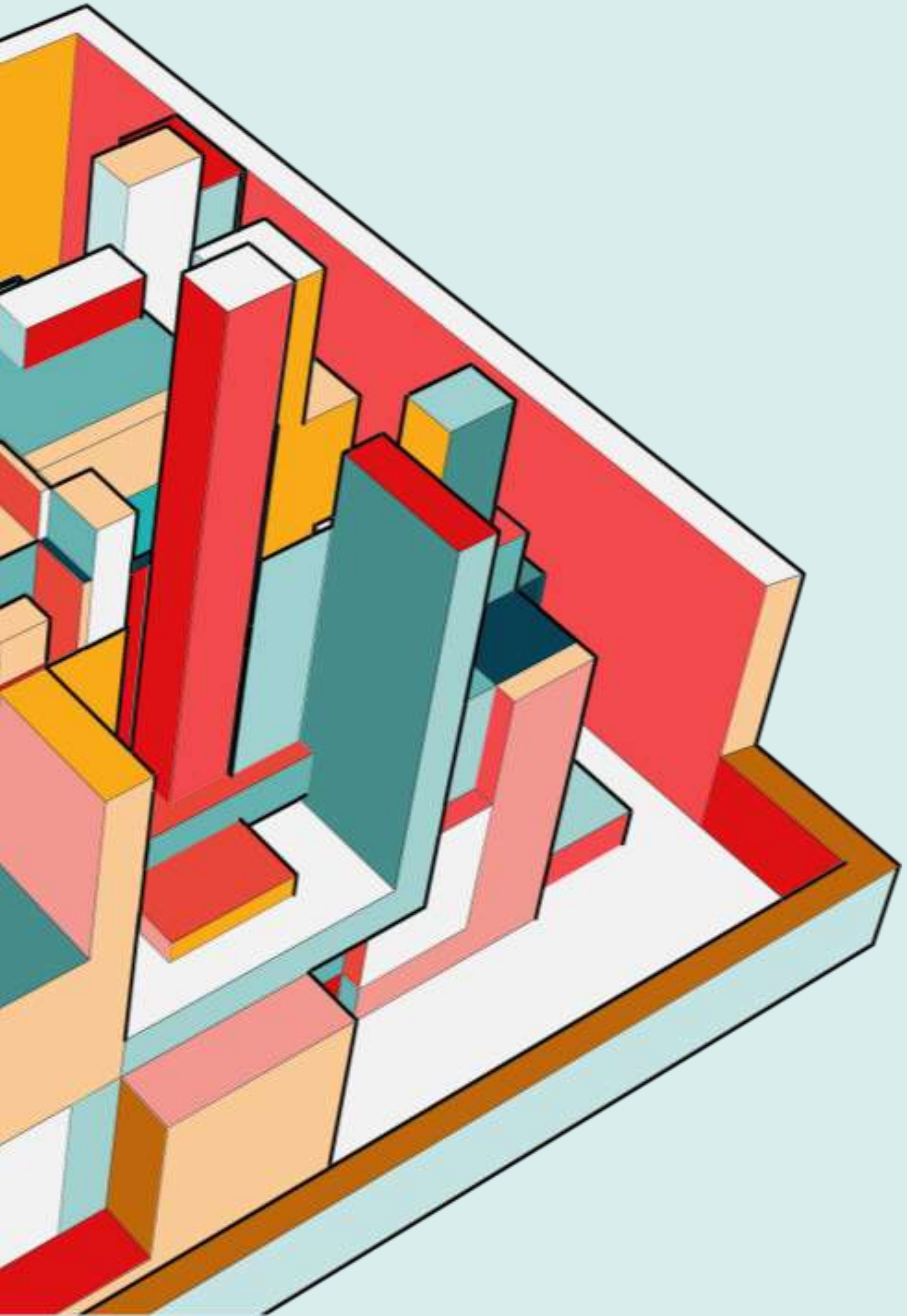
- Introducing our dataset:
  - What’s Zillow?
  - Why Zillow?
  - Why Los Angeles?
- Our Goal
- Scraping data
- Preprocess Steps
- Analyzing Distribution Outliers
- TOP 10s
- Findings from Visualizing Feature Changes Based on Year Built
- Correlation
- Group Bar Plots
- Growth Rate
- Statistical Test
- Next Step…





WHAT’S ZILLOW?

Zillow is a popular online real estate marketplace that provides information about homes, real estate listings, mortgages, and home improvement. It allows users to search for properties for sale or rent, as well as estimate the value of homes through its "Zestimate" feature, which uses an algorithm to provide an estimated market value based on various factors. Zillow also offers tools for homeowners, renters, real estate agents, and property managers to list properties, advertise, and manage their real estate transactions.



WHY ZILLOW?

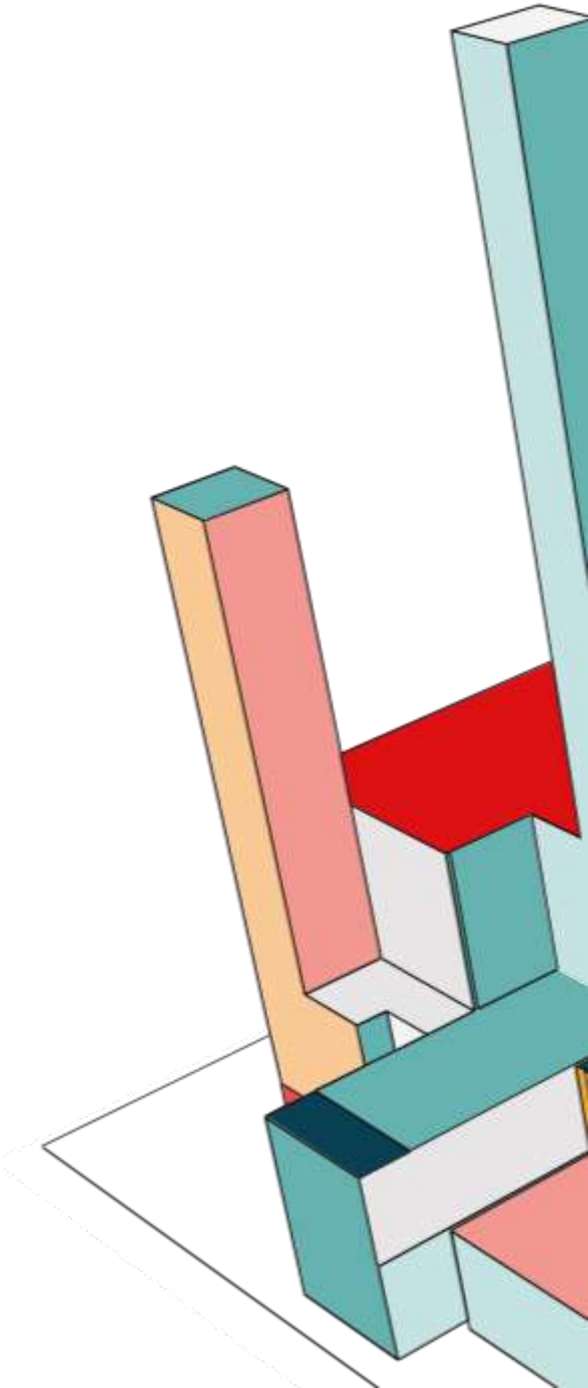
1. **Rich Dataset:** Zillow provides extensive real estate data including listings, property values, and demographic information.

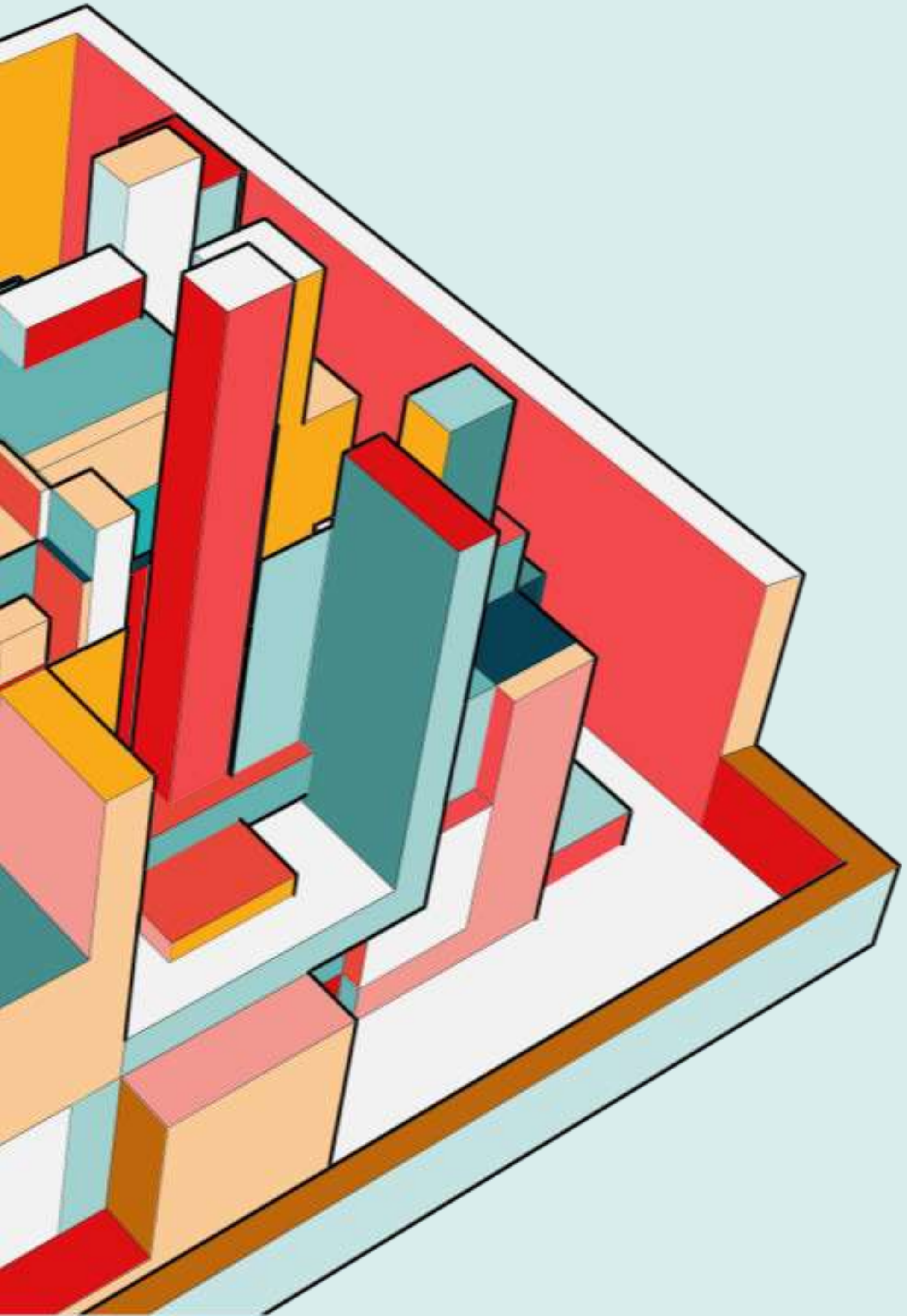
2. **High-Quality Data:** Zillow's data is well-structured and reliable, simplifying analysis and modeling.

3. **Popularity:** Widely used by stakeholders, Zillow data is familiar and relevant, encouraging collaboration.

4. **Real-World Applications:** Real estate data from Zillow enables practical analysis like price prediction and trend identification.

5. **Business Insights**: Analyzing Zillow data uncovers valuable insights for investors, developers, and policymakers, creating business opportunities.





WHY WE CHOSE LOS ANGELES?

- Los Angeles has a wide variety of property types and price ranges, from affordable homes to luxury estates. This diversity provides a rich dataset for analyzing different factors affecting housing prices and understanding various market segments.Tone inflection

- Scraping data for the entire United States can be resource-intensive in terms of computing power, time, and cost. Limiting the scope to Los Angeles may be more manageable within available resources.

- Focusing on a specific region like Los Angeles could be a pilot study to test methodologies or algorithms before scaling up to include data from other states.

OUR MAIN GOAL

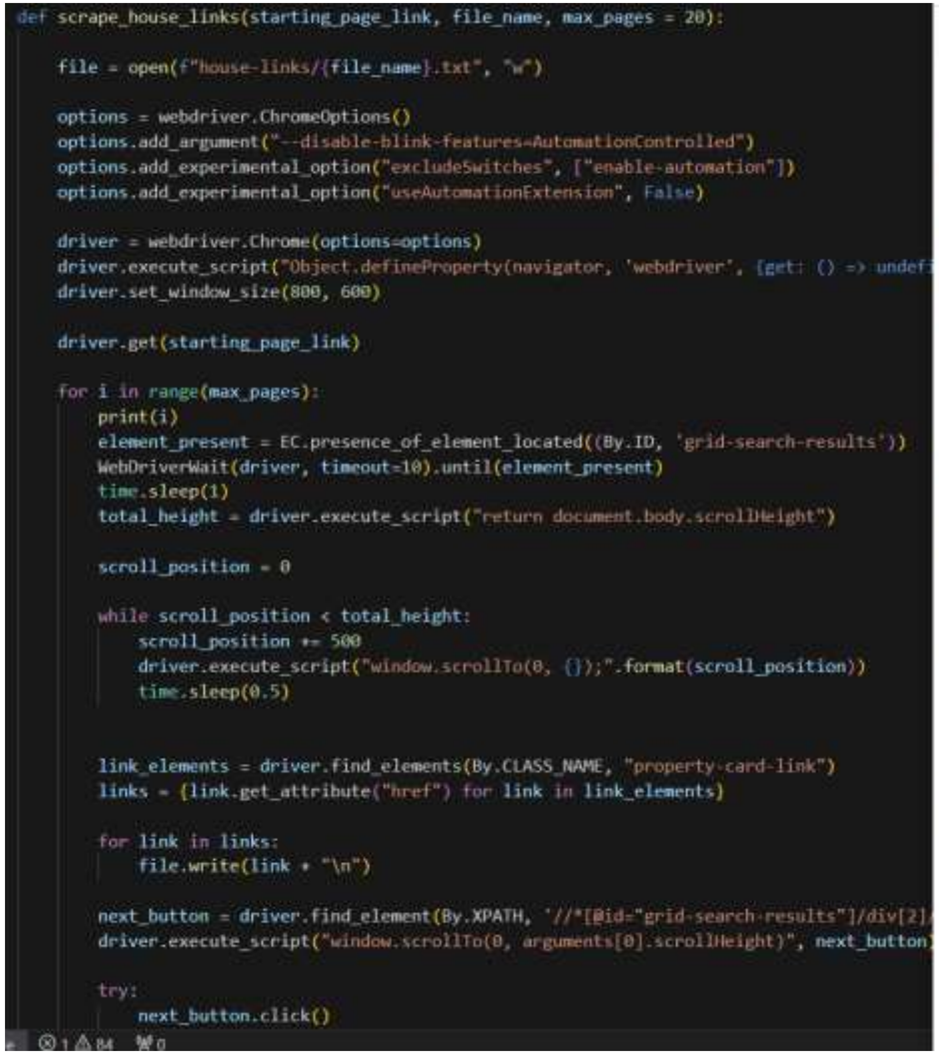
1. **Data Analysis**: Perform analysis on the collected real estate data to explore trends, patterns, and correlations between house features and prices. This phase involves examining the data to derive insights that can inform the modeling process.

2. **Model Development**: Build a predictive model that estimates house prices based on various home features. This model aims to accurately predict house prices using machine learning algorithms.

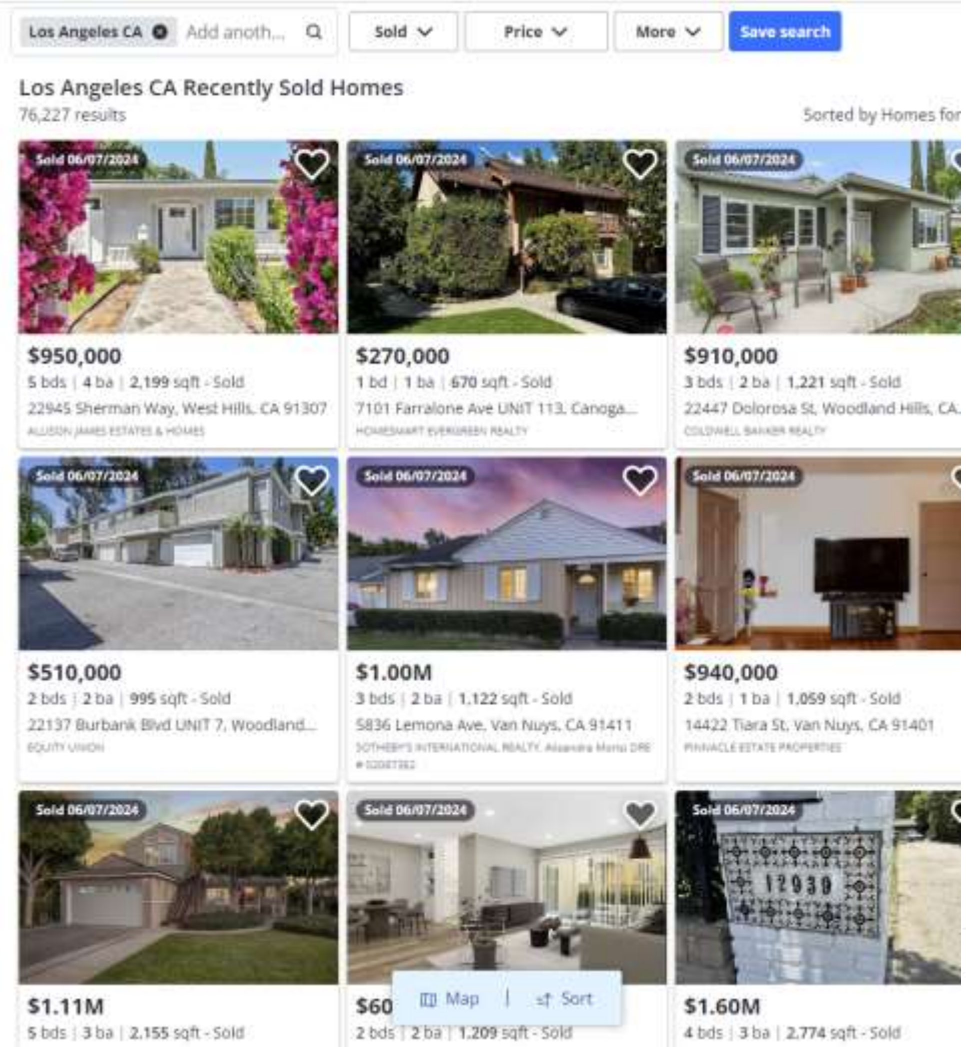
3. **Model Evaluation and Fine-Tuning**: Evaluate the performance of the predictive model using appropriate metrics and fine-tune it to improve accuracy and robustness. This iterative process involves adjusting model parameters and features based on performance results.

SCRAPING DATA

STEP 1: Scraping the links of the houses



```
def scrape_house_links(starting_page_link, file_name, max_pages = 20):

    file = open(f"house-links/{file_name}.txt", "w")

    options = webdriver.ChromeOptions()
    options.add_argument("--disable-blink-features=AutomationControlled")
    options.add_experimental_option("excludeSwitches", ["enable-automation"])
    options.add_experimental_option("useAutomationExtension", False)

    driver = webdriver.Chrome(options=options)
    driver.execute_script("Object.defineProperty(navigator, 'webdriver', {get: () => undefi
    driver.set_window_size(800, 600)

    driver.get(starting_page_link)

    for i in range(max_pages):
        print(i)
        element_present = EC.presence_of_element_located((By.ID, 'grid-search-results'))
        WebDriverWait(driver, timeout=10).until(element_present)
        time.sleep(1)
        total_height = driver.execute_script("return document.body.scrollHeight")

        scroll_position = 0

        while scroll_position < total_height:
            scroll_position += 500
            driver.execute_script("window.scrollTo(0, {});".format(scroll_position))
            time.sleep(0.5)


        link_elements = driver.find_elements(By.CLASS_NAME, "property-card-link")
        links = {link.get_attribute("href") for link in link_elements}

        for link in links:
            file.write(link + "\n")

        next_button = driver.find_element(By.XPATH, '//*[@id="grid-search-results"]/div[2]/
        driver.execute_script("window.scrollTo(0, arguments[0].scrollHeight)", next_button)

        try:
            next_button.click()
```



Los Angeles CA Recently Sold Homes

76,227 results

Sorted by Homes for

Sold 06/07/2024
$950,000
5 bds | 4 ba | 2,199 sqft - Sold
22945 Sherman Way, West Hills, CA 91307
ALLISON JAMES ESTATES & HOMES

Sold 06/07/2024
$270,000
1 bd | 1 ba | 670 sqft - Sold
7101 Farralone Ave UNIT 113, Canoga...
HOMESMART EVERGREEN REALTY

Sold 06/07/2024
$910,000
3 bds | 2 ba | 1,221 sqft - Sold
22447 Dolorosa St, Woodland Hills, CA...
COLDWELL BANKER REALTY

Sold 06/07/2024
$510,000
2 bds | 2 ba | 995 sqft - Sold
22137 Burbank Blvd UNIT 7, Woodland...
EQUITY UNION

Sold 06/07/2024
$1.00M
3 bds | 2 ba | 1,122 sqft - Sold
5836 Lemona Ave, Van Nuys, CA 91411
SOTHEBY'S INTERNATIONAL REALTY, Alexandra Mansi DRE # [illegible]

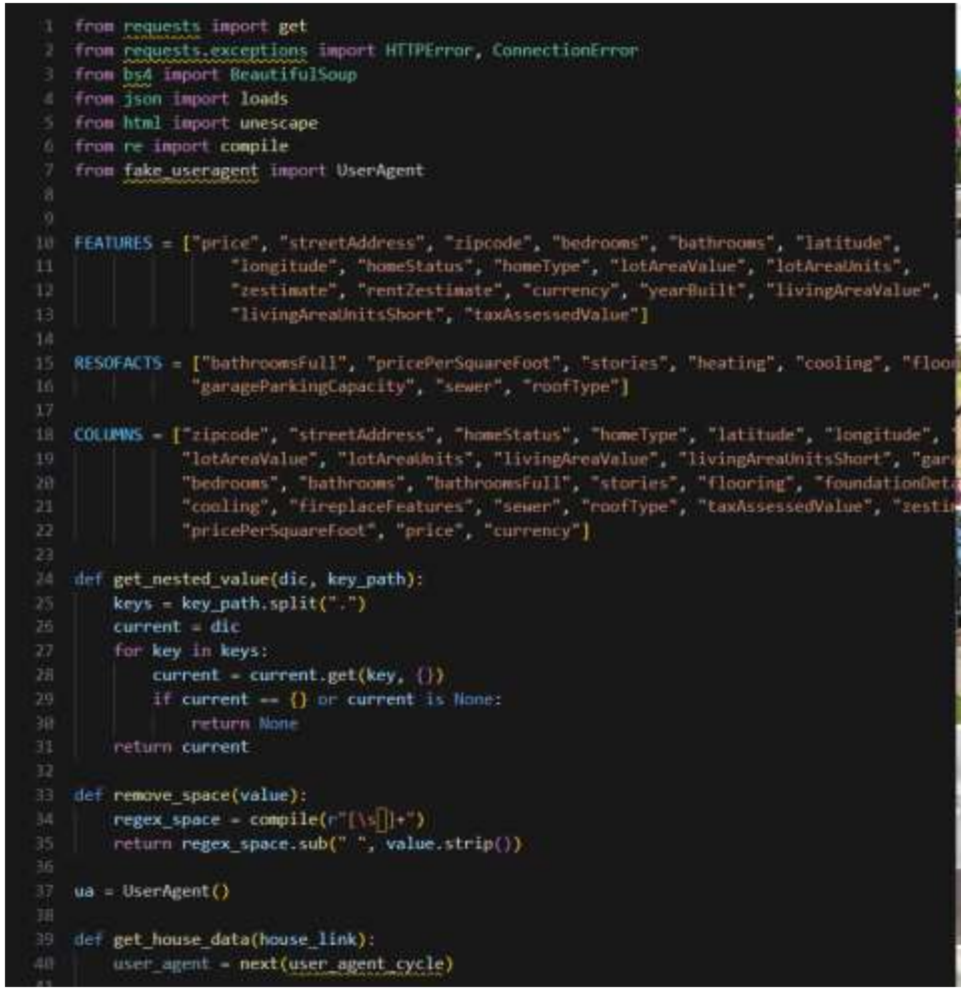
Sold 06/07/2024
$940,000
2 bds | 1 ba | 1,059 sqft - Sold
14422 Tiara St, Van Nuys, CA 91401
PINNACLE ESTATE PROPERTIES

Sold 06/07/2024
$1.11M
5 bds | 3 ba | 2,155 sqft - Sold

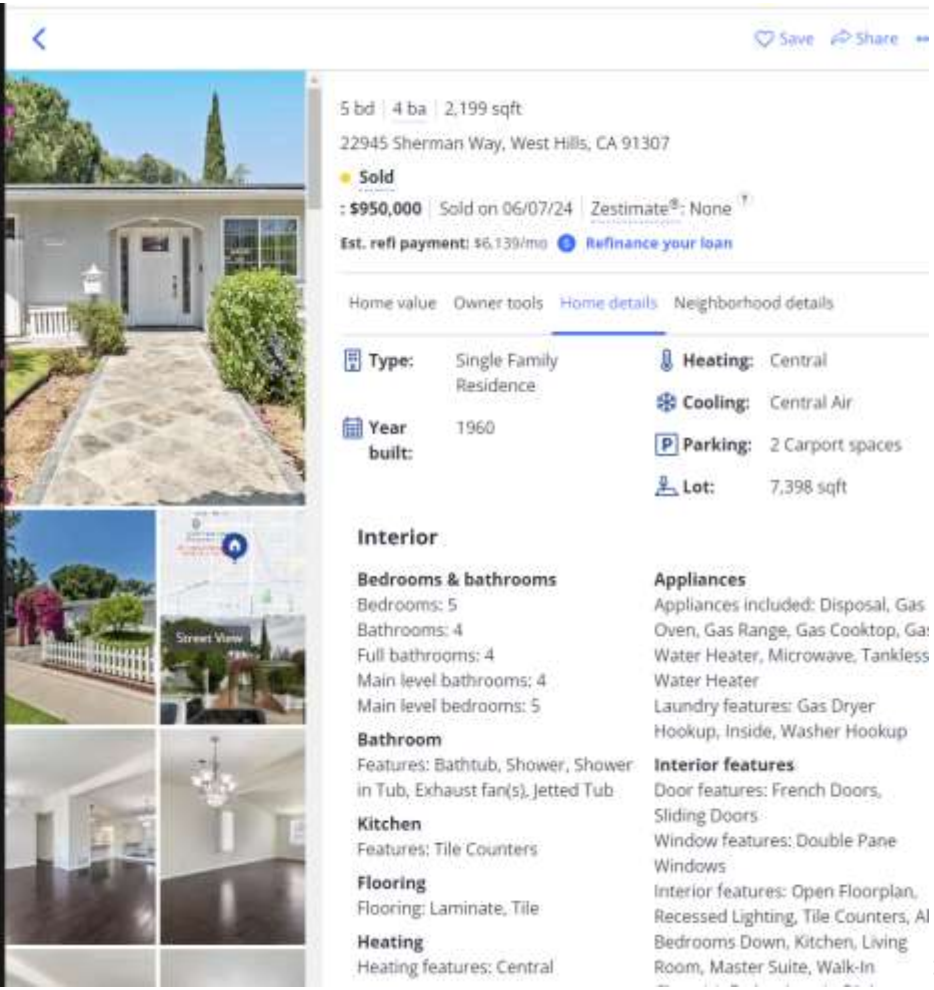
Sold 06/07/2024
$60
2 bds | 2 ba | 1,209 sqft - Sold

Sold 06/07/2024
$1.60M
4 bds | 3 ba | 2,774 sqft - Sold

STEP 2: Scraping the details of each house



```python
from requests import get
from requests.exceptions import HTTPError, ConnectionError
from bs4 import BeautifulSoup
from json import loads
from html import unescape
from re import compile
from fake_useragent import UserAgent


FEATURES = ["price", "streetAddress", "zipcode", "bedrooms", "bathrooms", "latitude",
            "longitude", "homeStatus", "homeType", "lotAreaValue", "lotAreaUnits",
            "zestimate", "rentZestimate", "currency", "yearBuilt", "livingAreaValue",
            "livingAreaUnitsShort", "taxAssessedValue"]

RESOFACTS = ["bathroomsFull", "pricePerSquareFoot", "stories", "heating", "cooling", "floo
          "garageParkingCapacity", "sewer", "roofType"]

COLUMNS = ["zipcode", "streetAddress", "homeStatus", "homeType", "latitude", "longitude",
           "lotAreaValue", "lotAreaUnits", "livingAreaValue", "livingAreaUnitsShort", "gar
           "bedrooms", "bathrooms", "bathroomsFull", "stories", "flooring", "foundationDet
           "cooling", "fireplaceFeatures", "sewer", "roofType", "taxAssessedValue", "zesti
           "pricePerSquareFoot", "price", "currency"]

def get_nested_value(dic, key_path):
    keys = key_path.split(".")
    current = dic
    for key in keys:
        current = current.get(key, {})
        if current == {} or current is None:
            return None
    return current

def remove_space(value):
    regex_space = compile(r"[\s ]+")
    return regex_space.sub(" ", value.strip())

ua = UserAgent()

def get_house_data(house_link):
    user_agent = next(user_agent_cycle)
```



5 bd | 4 ba | 2,199 sqft

22945 Sherman Way, West Hills, CA 91307

● Sold

: $950,000 | Sold on 06/07/24 | Zestimate®: None

Est. refi payment: $6,139/mo Refinance your loan

Type:	Single Family Residence	**Heating:**	Central
		Cooling:	Central Air
Year built:	1960	**Parking:**	2 Carport spaces
		Lot:	7,398 sqft

Interior

Bedrooms & bathrooms
Bedrooms: 5
Bathrooms: 4
Full bathrooms: 4
Main level bathrooms: 4
Main level bedrooms: 5

Bathroom
Features: Bathtub, Shower, Shower in Tub, Exhaust fan(s), Jetted Tub

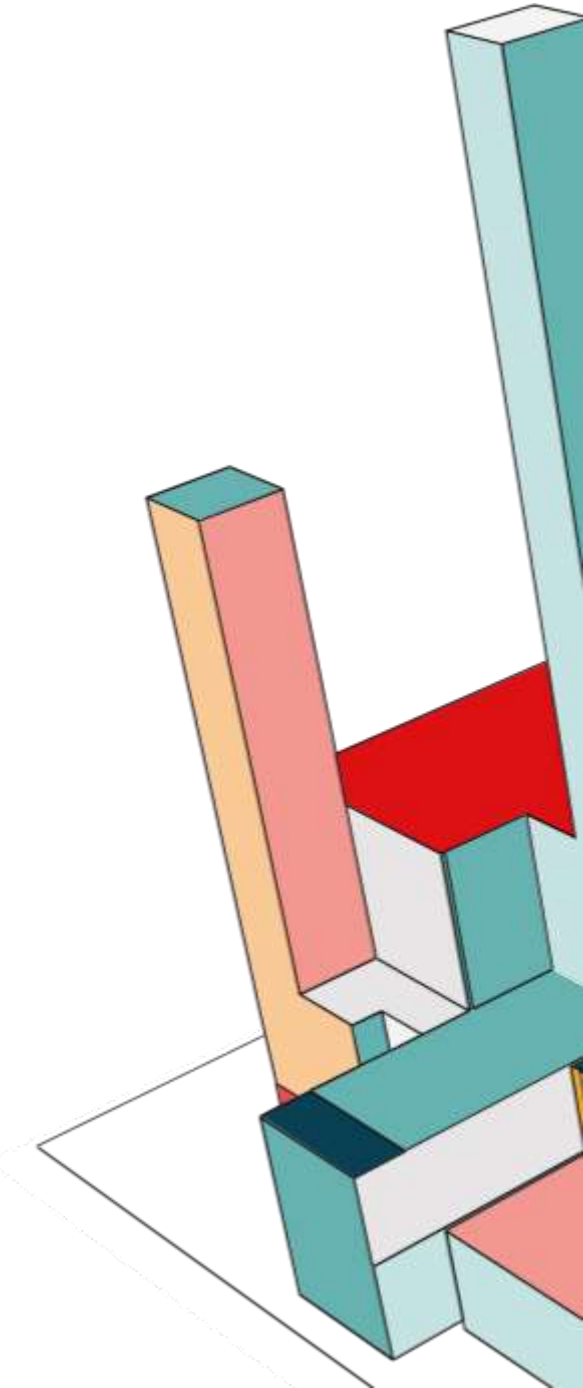
Kitchen
Features: Tile Counters

Flooring
Flooring: Laminate, Tile

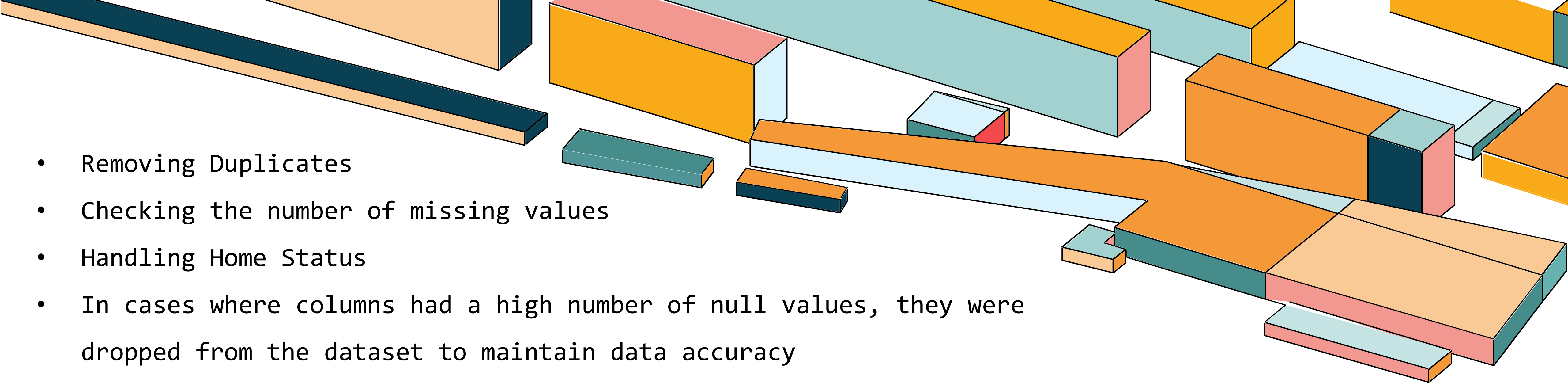
Heating
Heating features: Central

Appliances
Appliances included: Disposal, Gas Oven, Gas Range, Gas Cooktop, Ga: Water Heater, Microwave, Tankless Water Heater
Laundry features: Gas Dryer Hookup, Inside, Washer Hookup

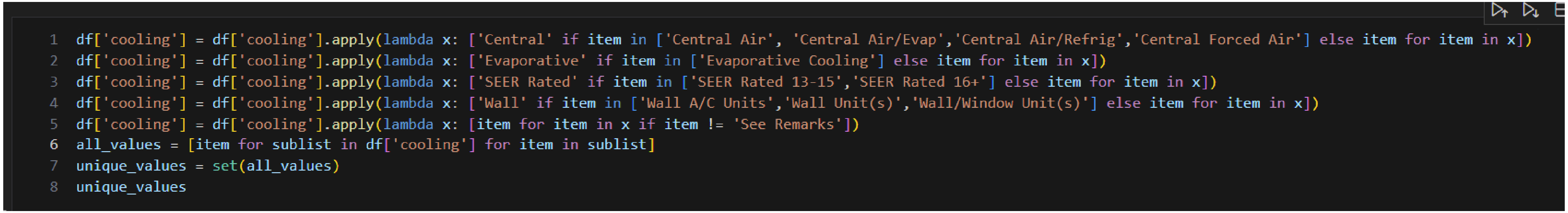
Interior features
Door features: French Doors, Sliding Doors
Window features: Double Pane Windows
Interior features: Open Floorplan, Recessed Lighting, Tile Counters, Al Bedrooms Down, Kitchen, Living Room, Master Suite, Walk-In



PREPROCESS STEPS



- Removing Duplicates
- Checking the number of missing values
- Handling Home Status
- In cases where columns had a high number of null values, they were dropped from the dataset to maintain data accuracy
- We convert string columns to real arrays and merge specific values for streamlined processing.
- We merged similar types of some features like cooling together and consolidated them into broader categories



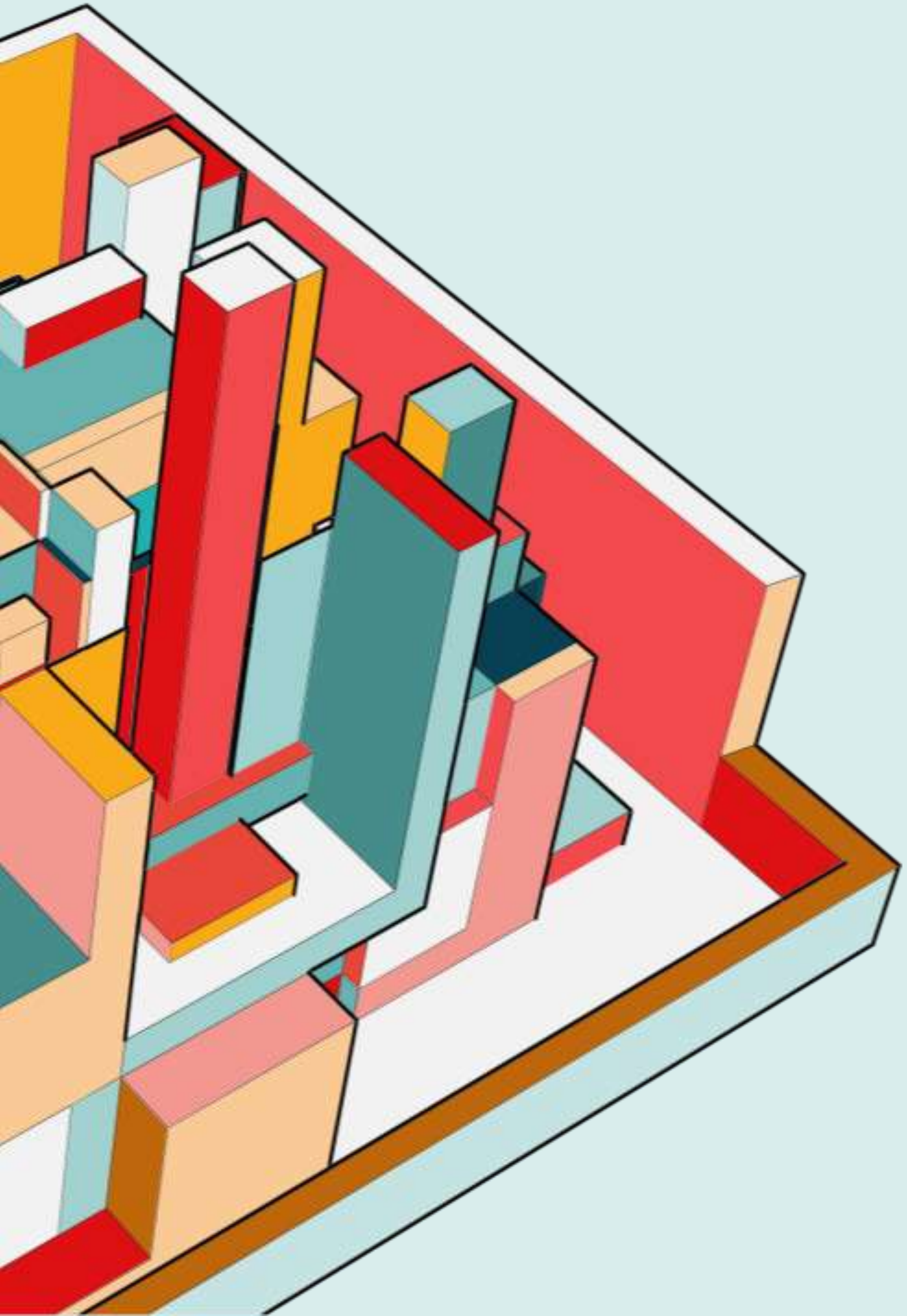
```
df['cooling'] = df['cooling'].apply(lambda x: ['Central' if item in ['Central Air', 'Central Air/Evap','Central Air/Refrig','Central Forced Air'] else item for item in x])
df['cooling'] = df['cooling'].apply(lambda x: ['Evaporative' if item in ['Evaporative Cooling'] else item for item in x])
df['cooling'] = df['cooling'].apply(lambda x: ['SEER Rated' if item in ['SEER Rated 13-15','SEER Rated 16+'] else item for item in x])
df['cooling'] = df['cooling'].apply(lambda x: ['Wall' if item in ['Wall A/C Units','Wall Unit(s)','Wall/Window Unit(s)'] else item for item in x])
df['cooling'] = df['cooling'].apply(lambda x: [item for item in x if item != 'See Remarks'])
all_values = [item for sublist in df['cooling'] for item in sublist]
unique_values = set(all_values)
unique_values
```

- Preprocess Home Features(some examples of how we handle different features)
  - Lot area and living:
    - convert units to the square
  - Tax Assesed Value
    - Missing values in these columns were filled using a KNN imputer technique.
  - Has… Features
    - We have some columns that indicate if house has a feature or not. We will convert them to 1 or 0 and fill the rows with null values with mode and if they are too much we drop the column.

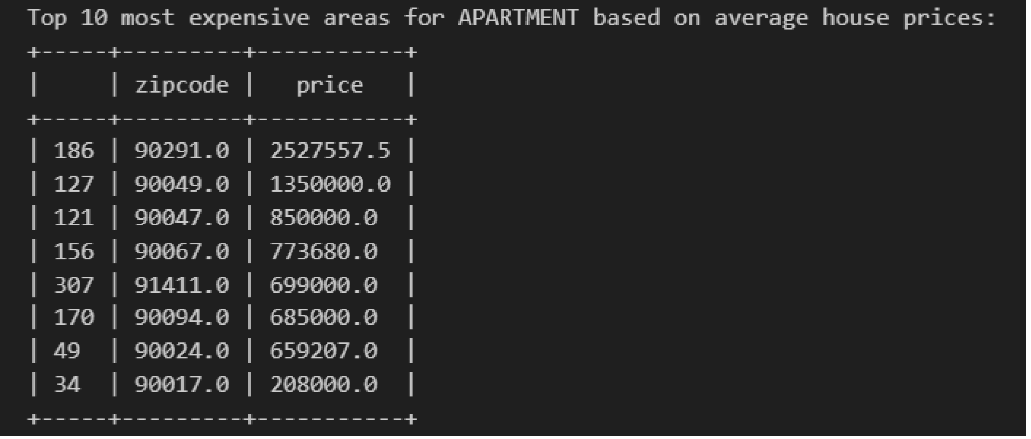
ANALYZING DISTRIBUTIONS AND OUTLIERS

We used box plots and histograms to analyze feature distributions, identify outliers, and clean the dataset. Selecting the number of bins for histograms is crucial, and we implemented three methods:

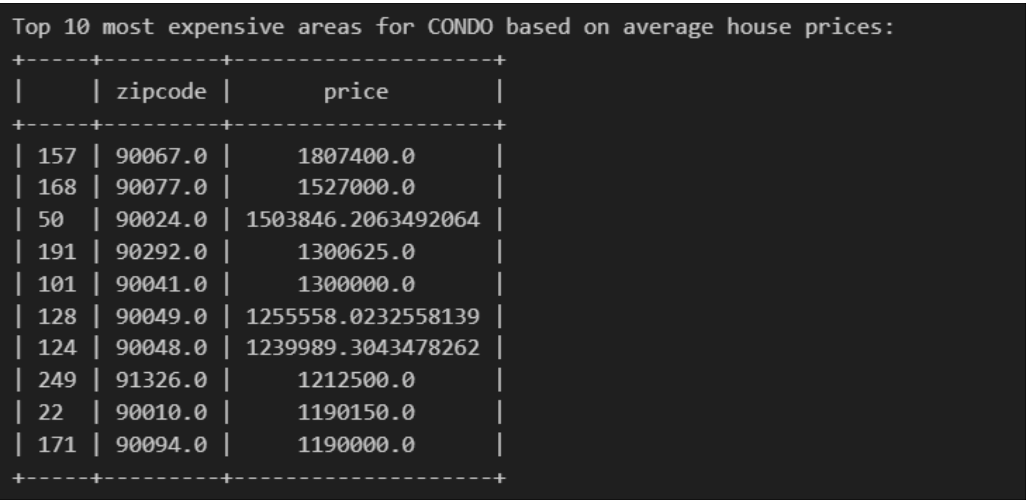
- Sturges' Rule
- Scott's Rule
- Freedman-Diaconis Rule



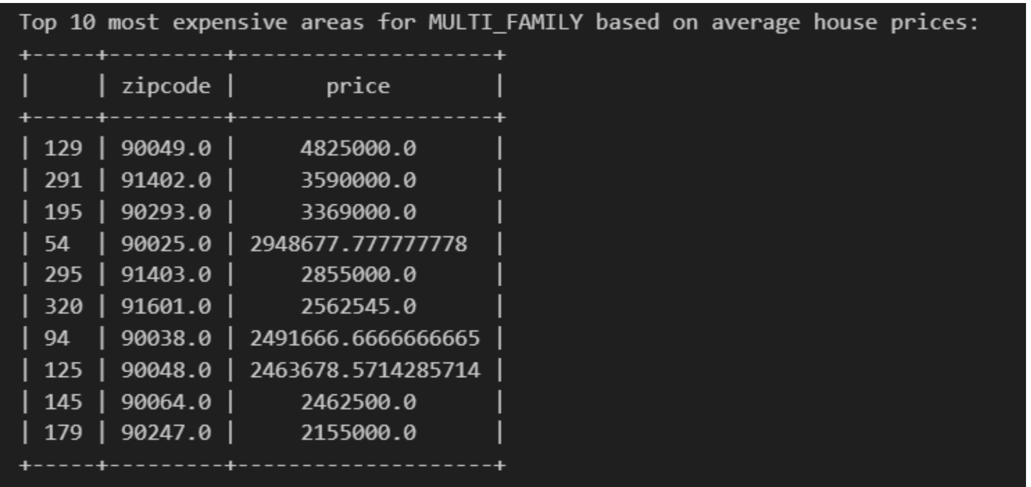
TOP 10S



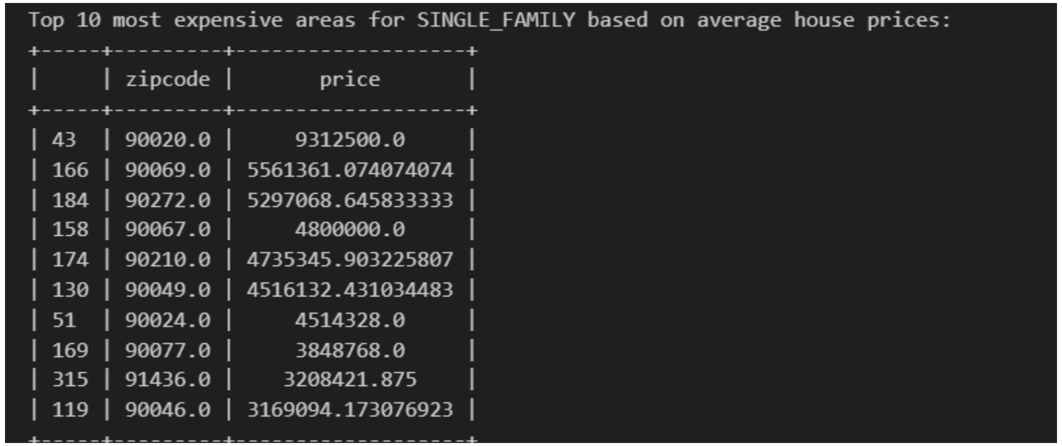
```
Top 10 most expensive areas for APARTMENT based on average house prices:
+-----+---------+-----------+
|     | zipcode |   price   |
+-----+---------+-----------+
| 186 | 90291.0 | 2527557.5 |
| 127 | 90049.0 | 1350000.0 |
| 121 | 90047.0 | 850000.0  |
| 156 | 90067.0 | 773680.0  |
| 307 | 91411.0 | 699000.0  |
| 170 | 90094.0 | 685000.0  |
| 49  | 90024.0 | 659207.0  |
| 34  | 90017.0 | 208000.0  |
+-----+---------+-----------+
```



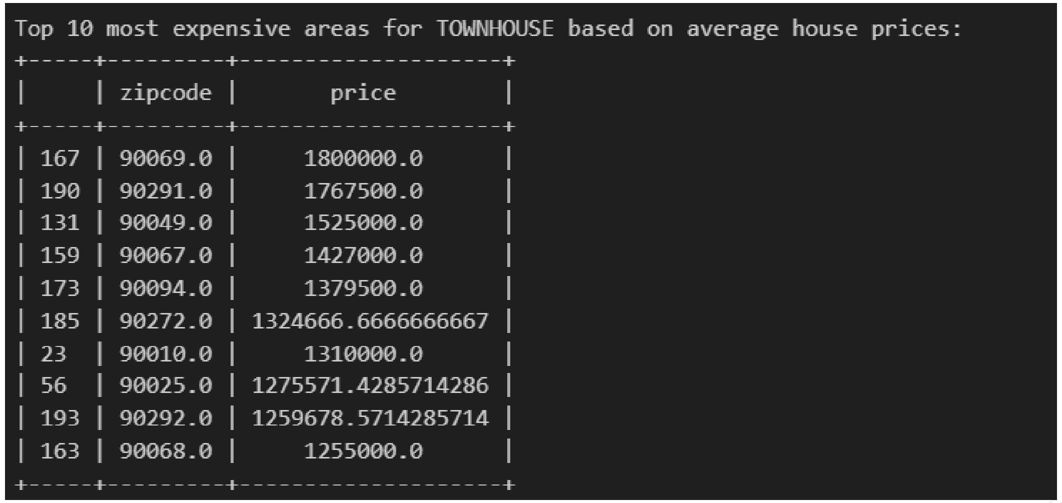
```
Top 10 most expensive areas for CONDO based on average house prices:
+-----+---------+--------------------+
|     | zipcode |       price        |
+-----+---------+--------------------+
| 157 | 90067.0 |     1807400.0      |
| 168 | 90077.0 |     1527000.0      |
| 50  | 90024.0 | 1503846.2063492064 |
| 191 | 90292.0 |     1300625.0      |
| 101 | 90041.0 |     1300000.0      |
| 128 | 90049.0 | 1255558.0232558139 |
| 124 | 90048.0 | 1239989.3043478262 |
| 249 | 91326.0 |     1212500.0      |
| 22  | 90010.0 |     1190150.0      |
| 171 | 90094.0 |     1190000.0      |
+-----+---------+--------------------+
```



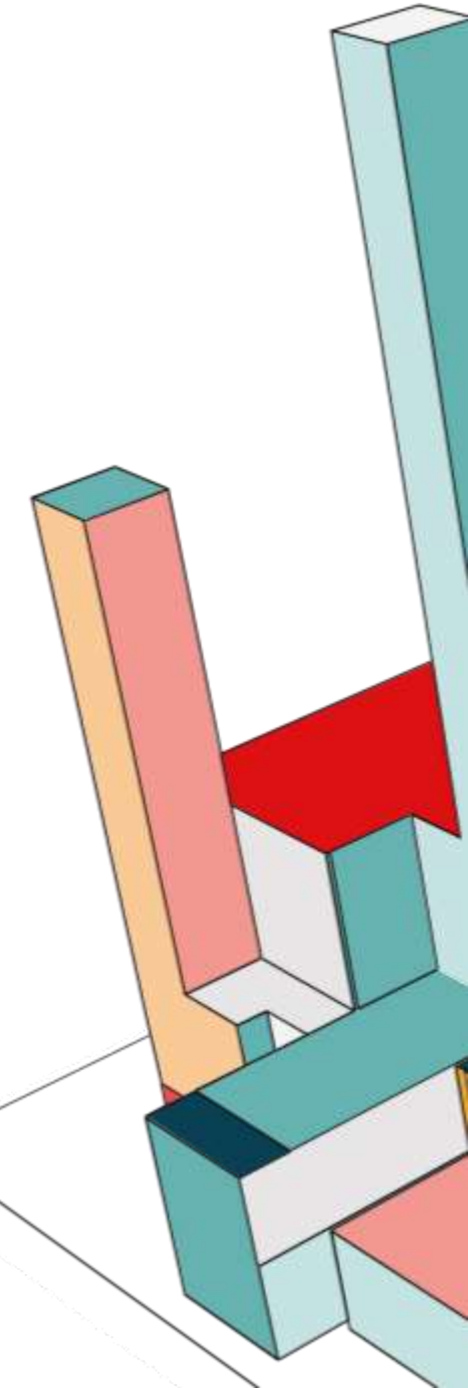
```
Top 10 most expensive areas for MULTI_FAMILY based on average house prices:
+-----+---------+--------------------+
|     | zipcode |       price        |
+-----+---------+--------------------+
| 129 | 90049.0 |     4825000.0      |
| 291 | 91402.0 |     3590000.0      |
| 195 | 90293.0 |     3369000.0      |
| 54  | 90025.0 | 2948677.777777778  |
| 295 | 91403.0 |     2855000.0      |
| 320 | 91601.0 |     2562545.0      |
| 94  | 90038.0 | 2491666.6666666665 |
| 125 | 90048.0 | 2463678.5714285714 |
| 145 | 90064.0 |     2462500.0      |
| 179 | 90247.0 |     2155000.0      |
+-----+---------+--------------------+
```



```
Top 10 most expensive areas for SINGLE_FAMILY based on average house prices:
+-----+---------+-------------------+
|     | zipcode |       price       |
+-----+---------+-------------------+
| 43  | 90020.0 |     9312500.0     |
| 166 | 90069.0 | 5561361.074074074 |
| 184 | 90272.0 | 5297068.645833333 |
| 158 | 90067.0 |     4800000.0     |
| 174 | 90210.0 | 4735345.903225807 |
| 130 | 90049.0 | 4516132.431034483 |
| 51  | 90024.0 |     4514328.0     |
| 169 | 90077.0 |     3848768.0     |
| 315 | 91436.0 |    3208421.875    |
| 119 | 90046.0 | 3169094.173076923 |
```



```
Top 10 most expensive areas for TOWNHOUSE based on average house prices:
+-----+---------+--------------------+
|     | zipcode |       price        |
+-----+---------+--------------------+
| 167 | 90069.0 |     1800000.0      |
| 190 | 90291.0 |     1767500.0      |
| 131 | 90049.0 |     1525000.0      |
| 159 | 90067.0 |     1427000.0      |
| 173 | 90094.0 |     1379500.0      |
| 185 | 90272.0 | 1324666.6666666667 |
| 23  | 90010.0 |     1310000.0      |
| 56  | 90025.0 | 1275571.4285714286 |
| 193 | 90292.0 | 1259678.5714285714 |
| 163 | 90068.0 |     1255000.0      |
+-----+---------+--------------------+
```



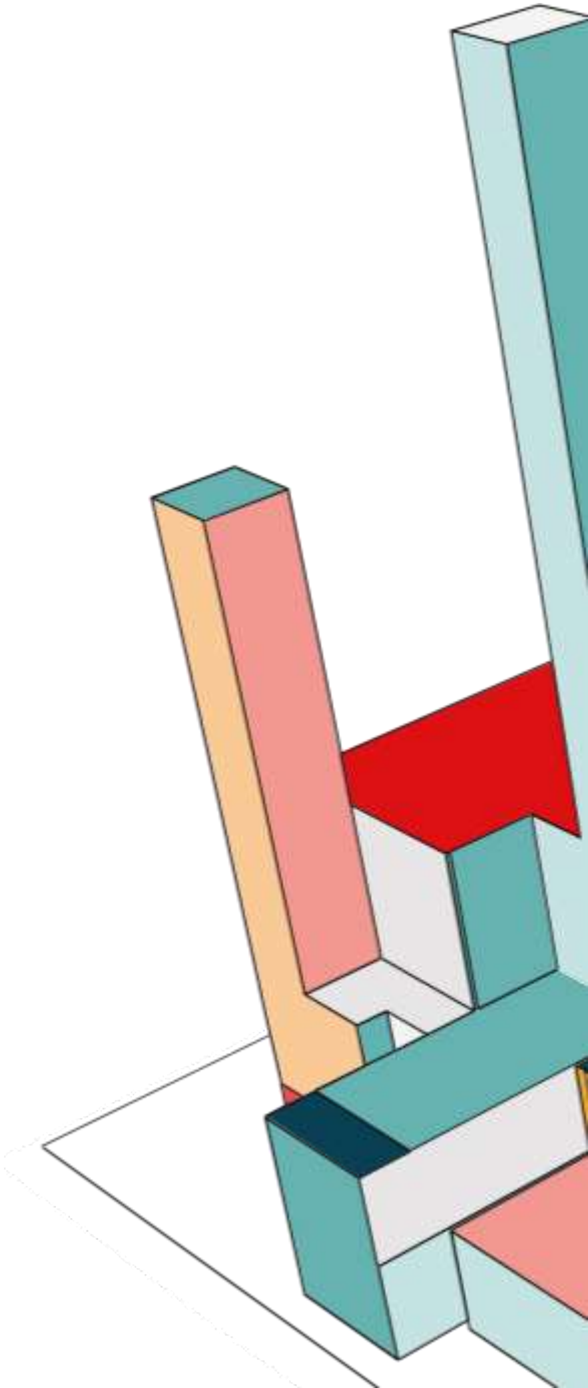
FINDINGS FROM VISUALIZING FEATURE CHANGES BASED ON YEAR BUILT

Here are our findings summarized:

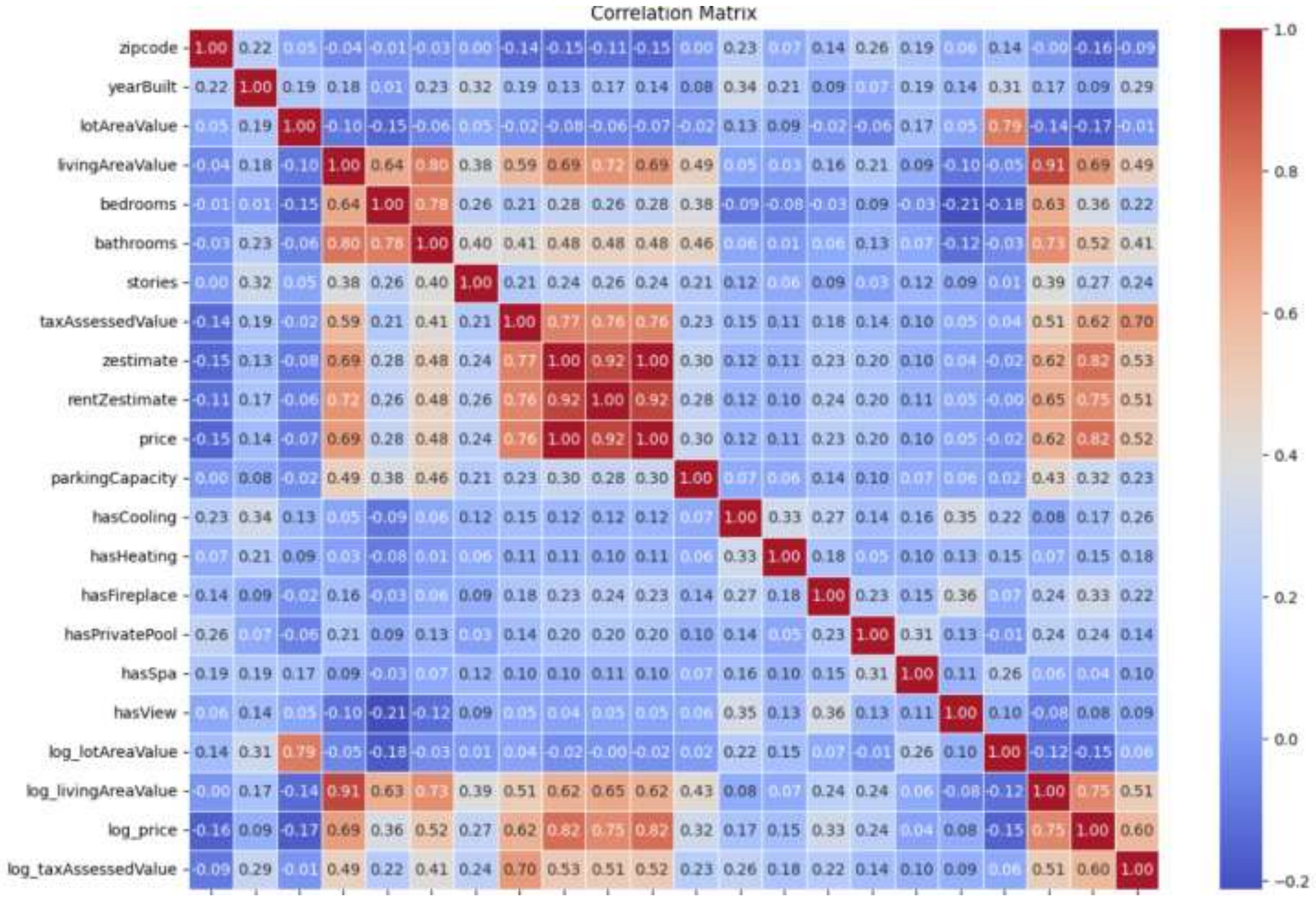
- **Average Bedrooms:**
  - **1900-1940: Stabilized around 3-5 bedrooms.**
  - **1940-1980: Increased fluctuations, peaks around 1955 and 1975.**
  - **1980-2000: Slight downward trend, dipping close to 3 around 1995.**
  - **2000-2020: Upward trend, reaching nearly 6 by 2020.**

- **Number of Stories:**
  - **Initially low variability and stability.**
  - **Gradual increase mid-20th century.**
  - **Significant upward trend from 1980 onwards.**

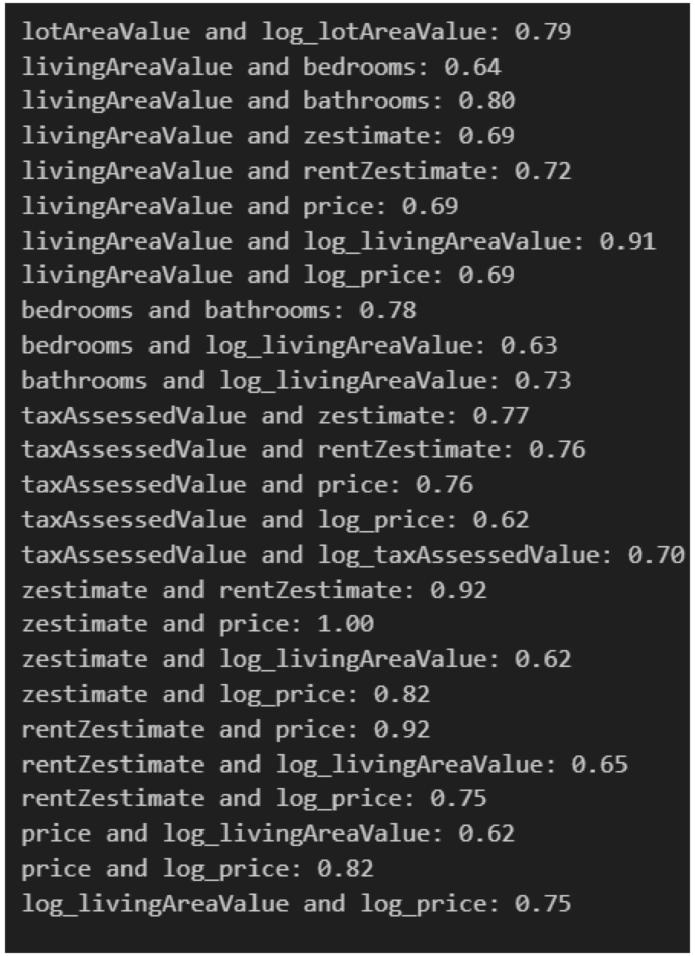
- **Cooling Systems:**
  - **Pre-1925: Almost no adoption.**
  - **1925-1965: Significant increase, peaks around 1940 and 1955.**
  - **1965-2000: Considerable fluctuations but overall increase.**
  - **2000-2020: Lower counts early on, sharp increase around 2020.**



CORRELATION


Correlation Matrix

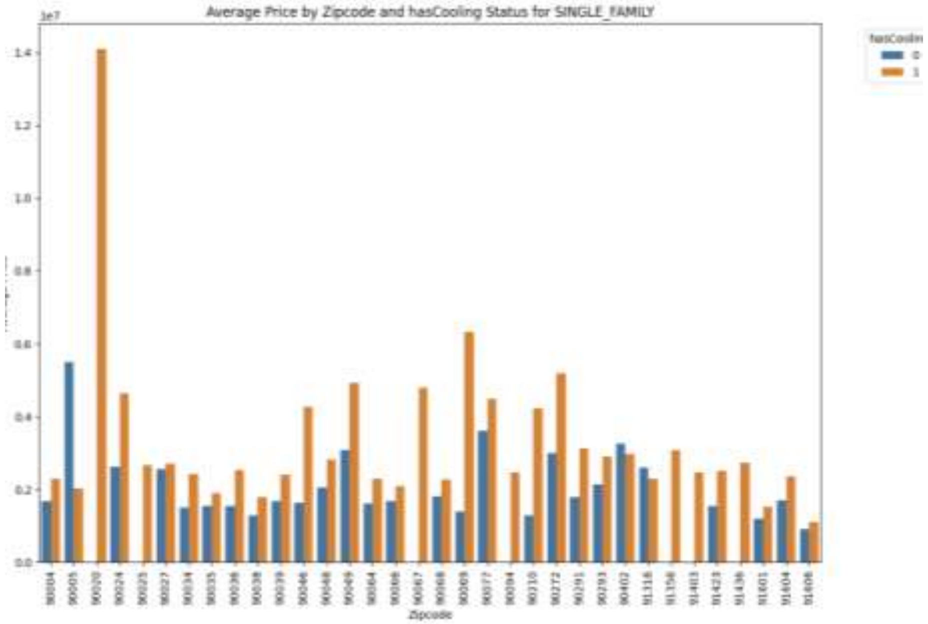
zipcode	1.00	0.22	0.05	-0.04	-0.01	-0.03	0.00	-0.14	-0.15	-0.11	-0.15	0.00	0.23	0.07	0.14	0.26	0.19	0.06	0.14	-0.00	-0.16	-0.09
yearBuilt	0.22	1.00	0.19	0.18	0.01	0.23	0.32	0.19	0.13	0.17	0.14	0.08	0.34	0.21	0.09	0.07	0.19	0.14	0.31	0.17	0.09	0.29
lotAreaValue	0.05	0.19	1.00	-0.10	-0.15	-0.06	0.05	-0.02	-0.08	-0.06	-0.07	-0.02	0.13	0.09	-0.02	-0.06	0.17	0.05	0.79	-0.14	-0.17	-0.01
livingAreaValue	-0.04	0.18	-0.10	1.00	0.64	0.80	0.38	0.59	0.69	0.72	0.69	0.49	0.05	0.03	0.16	0.21	0.09	-0.10	-0.05	0.91	0.69	0.49
bedrooms	-0.01	0.01	-0.15	0.64	1.00	0.78	0.26	0.21	0.28	0.26	0.28	0.38	-0.09	-0.08	-0.03	0.09	-0.03	-0.21	-0.18	0.63	0.36	0.22
bathrooms	-0.03	0.23	-0.06	0.80	0.78	1.00	0.40	0.41	0.48	0.48	0.48	0.46	0.06	0.01	0.06	0.13	0.07	-0.12	-0.03	0.73	0.52	0.41
stories	0.00	0.32	0.05	0.38	0.26	0.40	1.00	0.21	0.24	0.26	0.24	0.21	0.12	0.06	0.09	0.03	0.12	0.09	0.01	0.39	0.27	0.24
taxAssessedValue	-0.14	0.19	-0.02	0.59	0.21	0.41	0.21	1.00	0.77	0.76	0.76	0.23	0.15	0.11	0.18	0.14	0.10	0.05	0.04	0.51	0.62	0.70
zestimate	-0.15	0.13	-0.08	0.69	0.28	0.48	0.24	0.77	1.00	0.92	1.00	0.30	0.12	0.11	0.23	0.20	0.10	0.04	-0.02	0.62	0.82	0.53
rentZestimate	-0.11	0.17	-0.06	0.72	0.26	0.48	0.26	0.76	0.92	1.00	0.92	0.28	0.12	0.10	0.24	0.20	0.11	0.05	-0.00	0.65	0.75	0.51
price	-0.15	0.14	-0.07	0.69	0.28	0.48	0.24	0.76	1.00	0.92	1.00	0.30	0.12	0.11	0.23	0.20	0.10	0.05	-0.02	0.62	0.82	0.52
parkingCapacity	0.00	0.08	-0.02	0.49	0.38	0.46	0.21	0.23	0.30	0.28	0.30	1.00	0.07	0.06	0.14	0.10	0.07	0.06	0.02	0.43	0.32	0.23
hasCooling	0.23	0.34	0.13	0.05	-0.09	0.06	0.12	0.15	0.12	0.12	0.12	0.07	1.00	0.33	0.27	0.14	0.16	0.35	0.22	0.08	0.17	0.26
hasHeating	0.07	0.21	0.09	0.03	-0.08	0.01	0.06	0.11	0.11	0.10	0.11	0.06	0.33	1.00	0.18	0.05	0.10	0.13	0.15	0.07	0.15	0.18
hasFireplace	0.14	0.09	-0.02	0.16	-0.03	0.06	0.09	0.18	0.23	0.24	0.23	0.14	0.27	0.18	1.00	0.23	0.15	0.36	0.07	0.24	0.33	0.22
hasPrivatePool	0.26	0.07	-0.06	0.21	0.09	0.13	0.03	0.14	0.20	0.20	0.20	0.10	0.14	0.05	0.23	1.00	0.31	0.13	-0.01	0.24	0.24	0.14
hasSpa	0.19	0.19	0.17	0.09	-0.03	0.07	0.12	0.10	0.10	0.11	0.10	0.07	0.16	0.10	0.15	0.31	1.00	0.11	0.26	0.06	0.04	0.10
hasView	0.06	0.14	0.05	-0.10	-0.21	-0.12	0.09	0.05	0.04	0.05	0.05	0.06	0.35	0.13	0.36	0.13	0.11	1.00	0.10	-0.08	0.08	0.09
log_lotAreaValue	0.14	0.31	0.79	-0.05	-0.18	-0.03	0.01	0.04	-0.02	-0.00	-0.02	0.02	0.22	0.15	0.07	-0.01	0.26	0.10	1.00	-0.12	-0.15	0.06
log_livingAreaValue	-0.00	0.17	-0.14	0.91	0.63	0.73	0.39	0.51	0.62	0.65	0.62	0.43	0.08	0.07	0.24	0.24	0.06	-0.08	-0.12	1.00	0.75	0.51
log_price	-0.16	0.09	-0.17	0.69	0.36	0.52	0.27	0.62	0.82	0.75	0.82	0.32	0.17	0.15	0.33	0.24	0.04	0.08	-0.15	0.75	1.00	0.60
log_taxAssessedValue	-0.09	0.29	-0.01	0.49	0.22	0.41	0.24	0.70	0.53	0.51	0.52	0.23	0.26	0.18	0.22	0.14	0.10	0.09	0.06	0.51	0.60	1.00

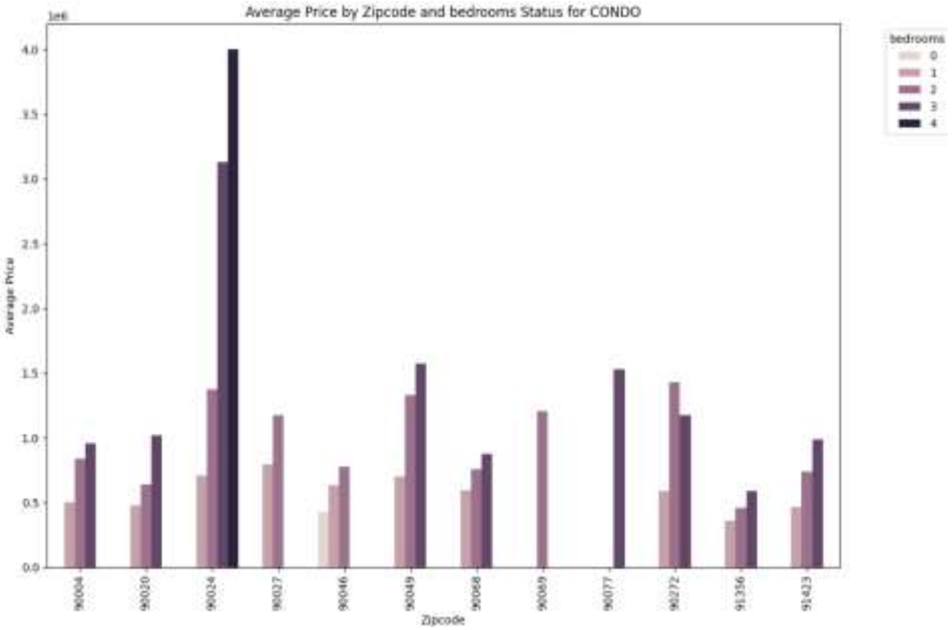


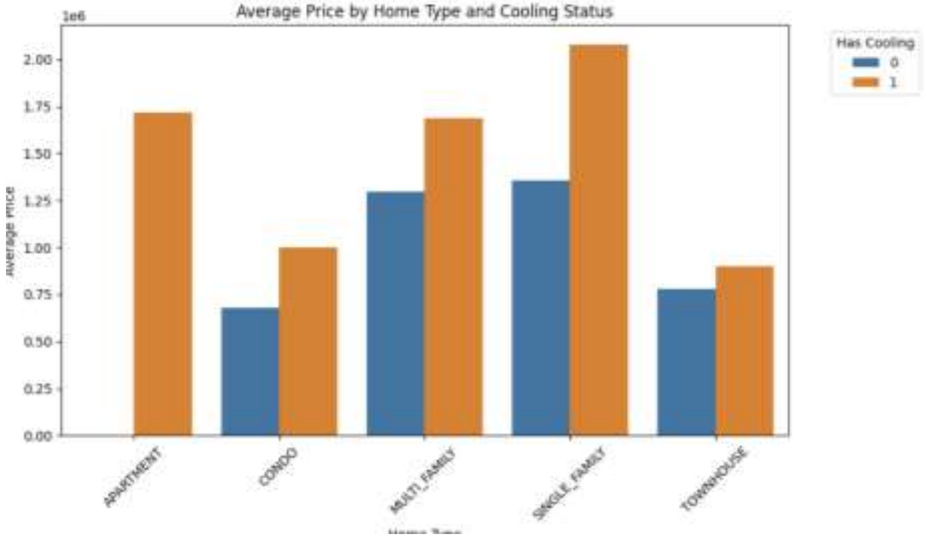
```
lotAreaValue and log_lotAreaValue: 0.79
livingAreaValue and bedrooms: 0.64
livingAreaValue and bathrooms: 0.80
livingAreaValue and zestimate: 0.69
livingAreaValue and rentZestimate: 0.72
livingAreaValue and price: 0.69
livingAreaValue and log_livingAreaValue: 0.91
livingAreaValue and log_price: 0.69
bedrooms and bathrooms: 0.78
bedrooms and log_livingAreaValue: 0.63
bathrooms and log_livingAreaValue: 0.73
taxAssessedValue and zestimate: 0.77
taxAssessedValue and rentZestimate: 0.76
taxAssessedValue and price: 0.76
taxAssessedValue and log_price: 0.62
taxAssessedValue and log_taxAssessedValue: 0.70
zestimate and rentZestimate: 0.92
zestimate and price: 1.00
zestimate and log_livingAreaValue: 0.62
zestimate and log_price: 0.82
rentZestimate and price: 0.92
rentZestimate and log_livingAreaValue: 0.65
rentZestimate and log_price: 0.75
price and log_livingAreaValue: 0.62
price and log_price: 0.82
log_livingAreaValue and log_price: 0.75
```

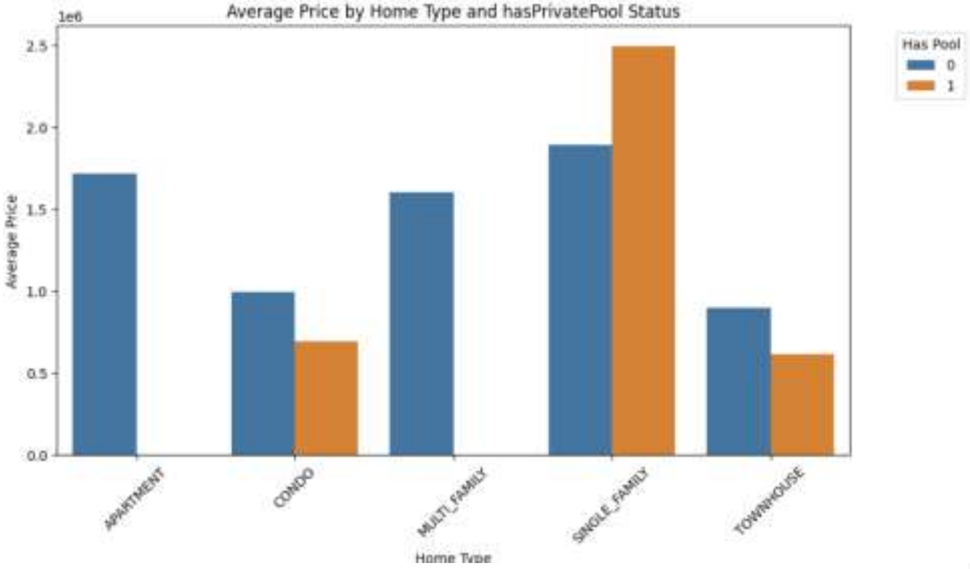
GROUP BAR PLOTS

We also visualized group bar plots, which provided interesting









Examples of Insights from Group Bar Plot Analysis

Single Family Homes:

1. **Desirable Views**: Highly desirable, premium pricing.
2. **Large Properties**: Better utilization of views.
3. **Luxury Status**: Associated with prestige and luxury.

Condos:

1. **Minimal View Impact**: Shared amenities, urban settings.
2. **Urban Proximity**: Convenience trumps views.
3. **Cost Focus**: Buyers prioritize affordability and location.

Townhouses/Apartments:

1. **Moderate View Value**: Higher floors, less exclusivity.
2. **Urban/Suburban**: Less impact in urban, some value in suburban.
3. **Convenience First**: Buyers prefer amenities and maintenance over views.

Insights from High-Priced Zip Codes:

•**Single Family Homes**: Views significantly increase prices due to desirability and exclusivity.

•**Townhouses**: Higher prices linked to scenic locations.

•**Condos and Multifamily Homes**: Views have minimal impact; shared amenities and property features are more important.

This highlights the need to consider property type and key value factors when assessing the impact of views on home prices in high-demand areas.

Parking Impact Insights:

Single Family Homes:

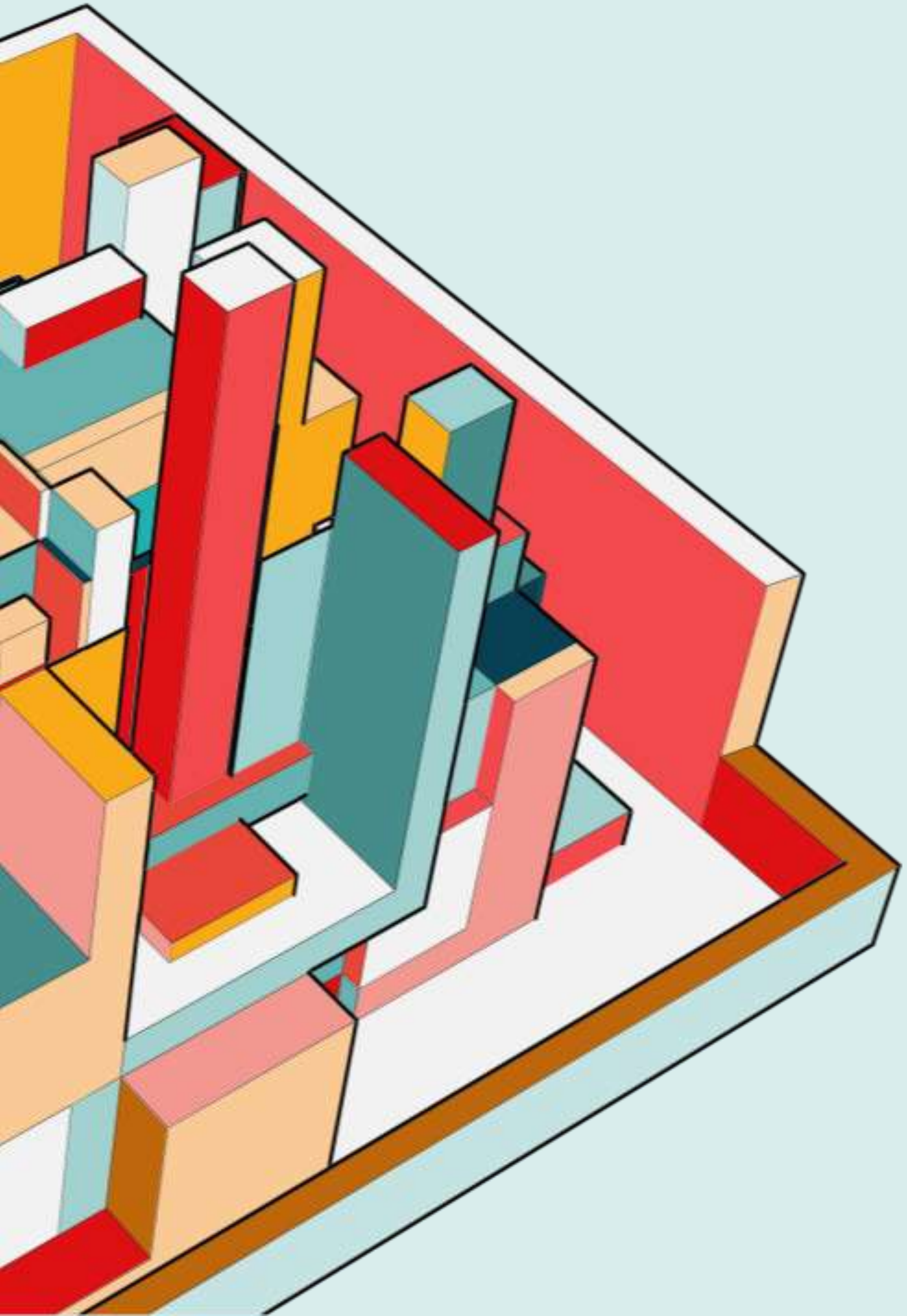
1. **Convenience**: Ample parking boosts demand and prices.
2. **Value**: Adds significant property value.

Condos and Multifamily Homes:

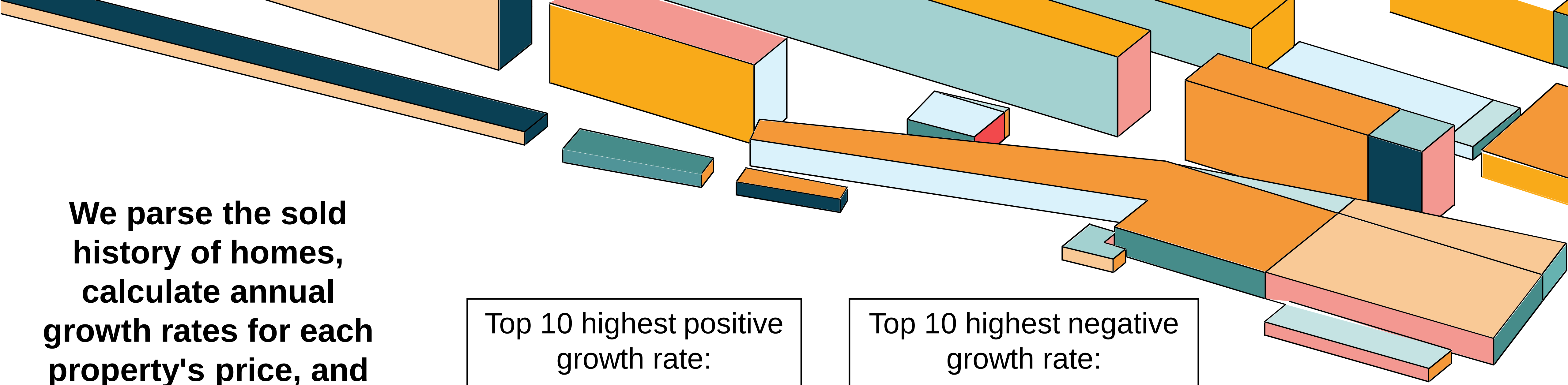
1. **Urban Necessity**: Parking crucial in urban areas, affecting prices.
2. **Demand**: High parking demand impacts design and pricing.

Townhouses and Apartments:

1. **Transport Access**: Less reliant on parking due to public transit.
2. **Shared Facilities**: Shared parking reduces focus on individual parking spaces.



GROWTH RATE BASED ON SOLD-HISTORY



We parse the sold history of homes, calculate annual growth rates for each property's price, and then compute the average growth rate by zipcode.

Top 10 highest positive growth rate:

```
90047 0.172034
90065 0.158968
90062 0.126168
90037 0.096886
90064 0.093413
90230 0.093221
91605 0.089674
90008 0.085783
90061 0.080442
90011 0.076536
```

Top 10 highest negative growth rate:

```
 90019 -1.020973
 90501 -1.255977
 90067 -1.265953
 91604 -1.282510
 90272 -1.730388
 90017 -3.617703
 91601 -4.139532
 90028 -5.974681
 90013 -7.372272
90015 -13.803312
```

A brief exploration into these neighborhoods🚀:

POSOTIVE GROWTH RATE:

- **Urban Revitalization**:
  - **90047, 90062**: South LA sees revitalization efforts and new housing developments.
  - **90037**: Near USC, demand from students and faculty boosts property values.
- **Improved Accessibility**:
  - **90064**: Metro Expo Line expansion enhances accessibility in Rancho Park.
  - **90230**: Culver City's redevelopment attracts residential and commercial interest.
- **Employment and Amenities**:
  - **91605**: NoHo Arts District's cultural attractions appeal to young professionals.
  - **90011**: Improving amenities draw residents to South LA near Central-Alameda.
- **Community and Culture**:
  - **90008**: Baldwin Hills' cultural significance attracts residents.
  - **90065**: Scenic views and community vibe in Glassell Park appeal to families.
- **Affordability**:
  - **90061**: Watts area offers relatively affordable housing options.
  - **90047**: South LA provides affordable living compared to other parts of LA.

NEGATIVE GROWTH RATE:

1.90019 (Mid-City): Limited development space and higher crime rates.

2.90501 (Torrance): Stable area with limited inventory and slower growth.

3.90067 (Century City): High property prices and limited residential options.

4.91604 (Studio City): High prices and limited inventory, leading to competition.

5.90272 (Pacific Palisades): Strict zoning and limited land availability.

6.90017 (Downtown LA - Historic Core): Homelessness and urban blight issues.

7.91601 (North Hollywood - Arts District): Industrial past and crime concerns.

8.90028 (Hollywood): High crime rates, traffic congestion, and homelessness.

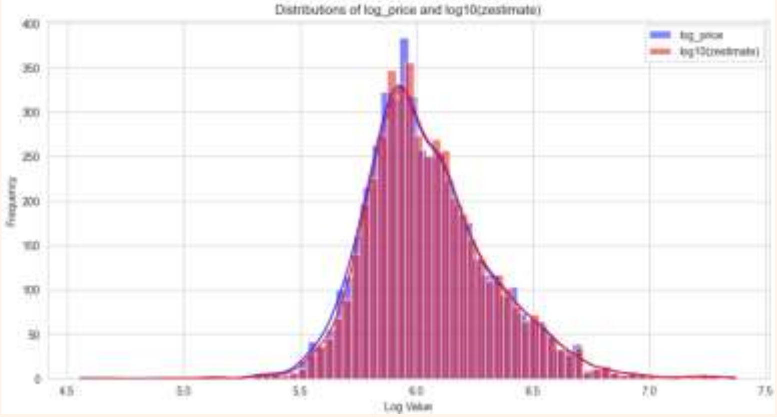
9.90013 (Downtown LA - Fashion District): Industrial area with limited amenities.

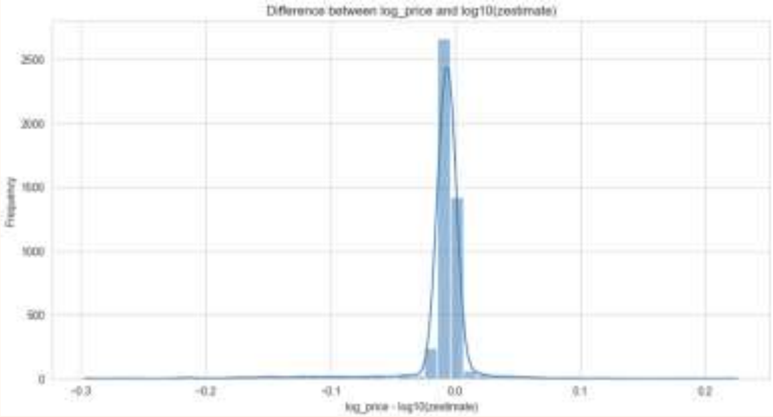
10.90015 (Downtown LA - South Park): Dense urban area with few residential amenities.

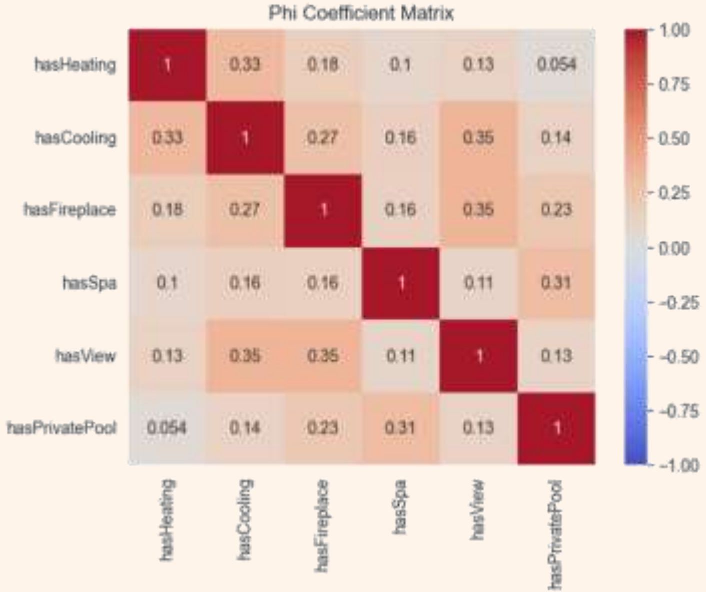
STATISTICAL TEST

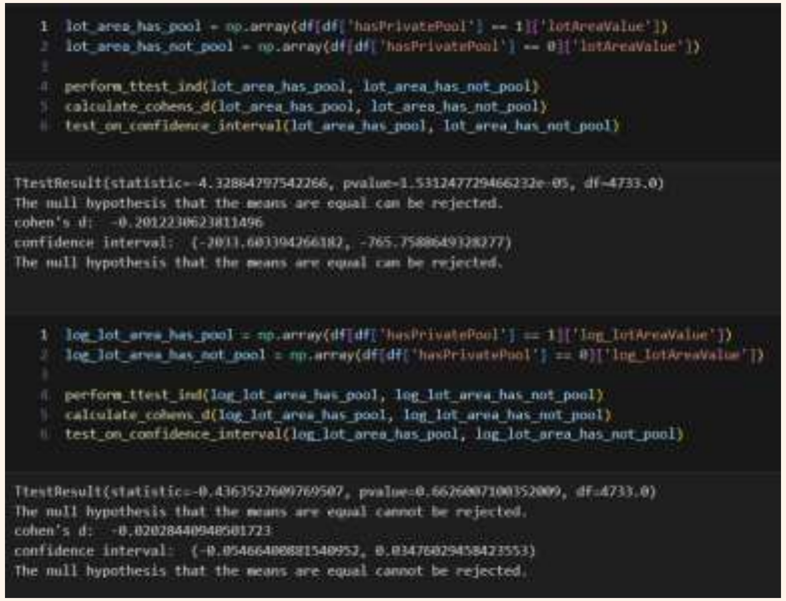
We will use the following statistical tests to analyze the features and their relationships:

- **Paired T-Test**
- **Independent T-Test**
- **Confidence Interval**
- **Chi-Square Test**
- **Cramér's V and Phi Coefficient**







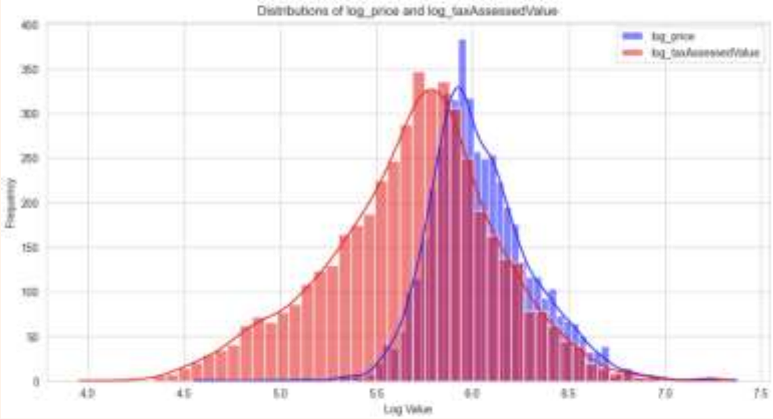


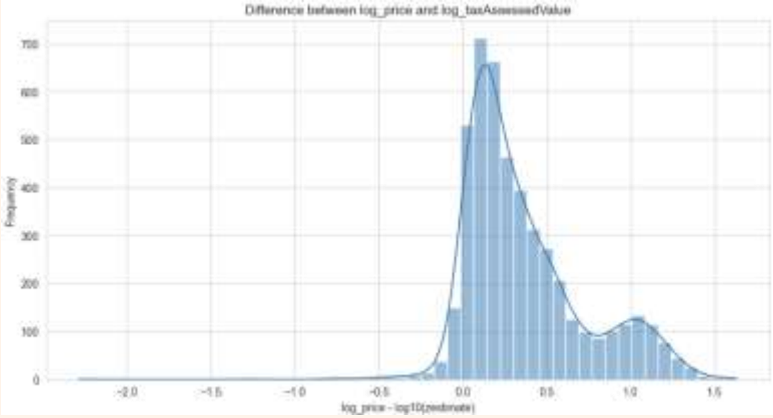
```
lot_area_has_pool = np.array(df[df['hasPrivatePool'] == 1]['lotAreaValue'])
lot_area_has_not_pool = np.array(df[df['hasPrivatePool'] == 0]['lotAreaValue'])

perform_ttest_ind(lot_area_has_pool, lot_area_has_not_pool)
calculate_cohens_d(lot_area_has_pool, lot_area_has_not_pool)
test_on_confidence_interval(lot_area_has_pool, lot_area_has_not_pool)
```

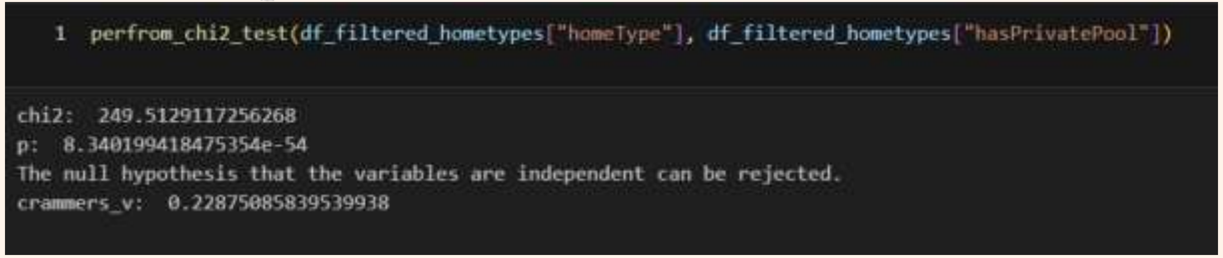
```
TtestResult(statistic=-4.32864797542266, pvalue=1.531247729466232e-05, df=4733.0)
The null hypothesis that the means are equal can be rejected.
cohen's d:  -0.2012230623811496
confidence interval:  (-2033.603394266182, -765.7588649328277)
The null hypothesis that the means are equal can be rejected.
```

```
log_lot_area_has_pool = np.array(df[df['hasPrivatePool'] == 1]['log_lotAreaValue'])
log_lot_area_has_not_pool = np.array(df[df['hasPrivatePool'] == 0]['log_lotAreaValue'])

perform_ttest_ind(log_lot_area_has_pool, log_lot_area_has_not_pool)
calculate_cohens_d(log_lot_area_has_pool, log_lot_area_has_not_pool)
test_on_confidence_interval(log_lot_area_has_pool, log_lot_area_has_not_pool)
```

```
TtestResult(statistic=-0.4363527609769507, pvalue=0.6626007100352009, df=4733.0)
The null hypothesis that the means are equal cannot be rejected.
cohen's d:  -0.02028440948501723
confidence interval:  (-0.05466400881540952, 0.03476029458423553)
The null hypothesis that the means are equal cannot be rejected.
```

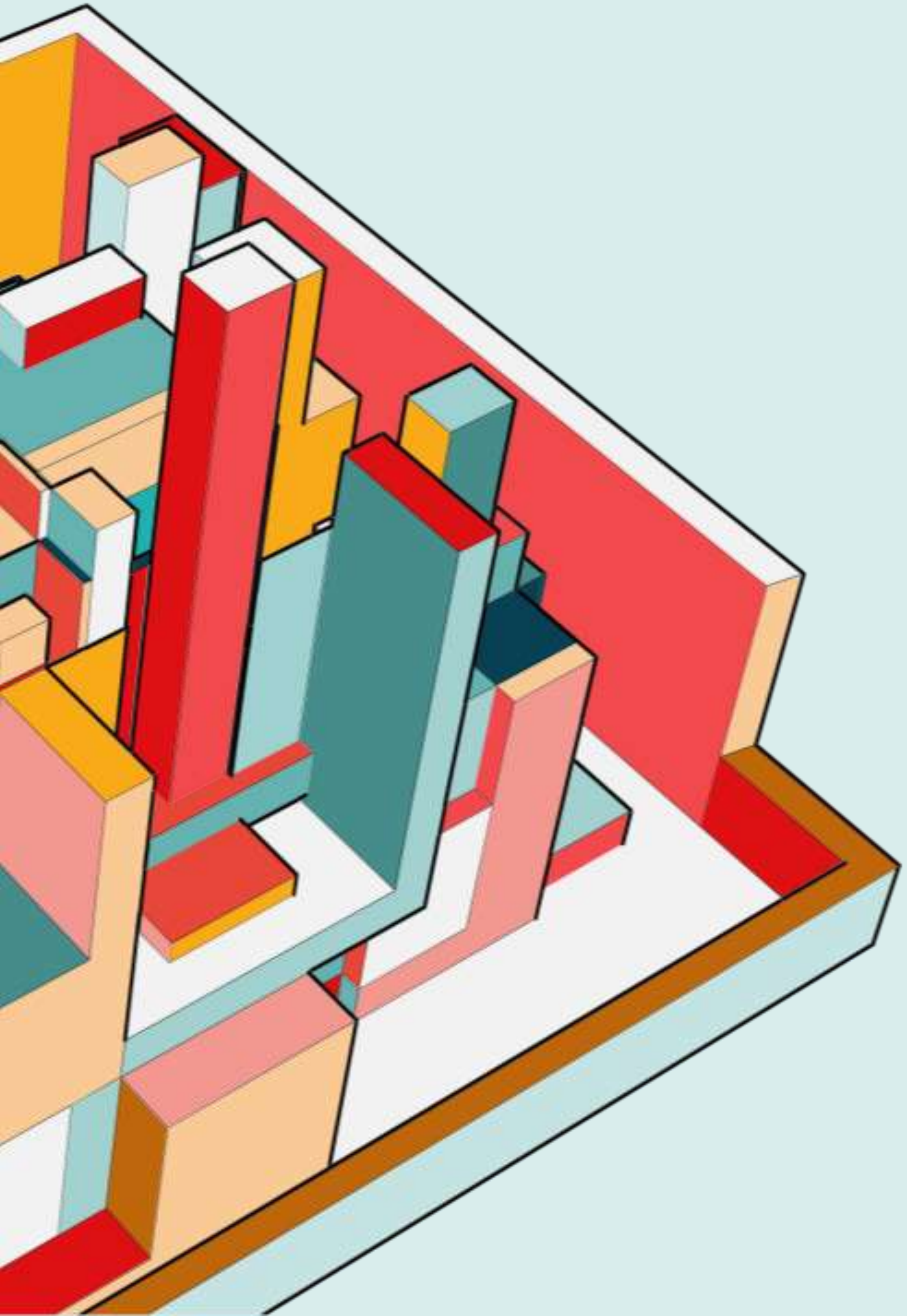






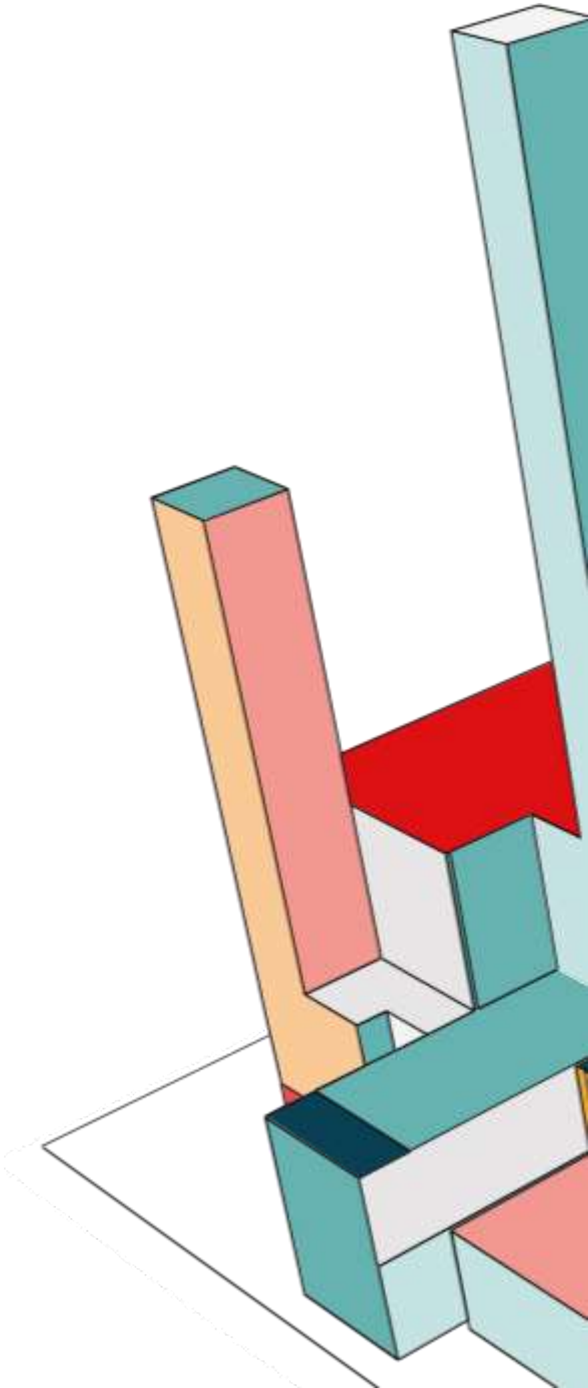
```
perfrom_chi2_test(df_filtered_hometypes["homeType"], df_filtered_hometypes["hasPrivatePool"])
```

```
chi2:  249.5129117256268
p:  8.340199418475354e-54
The null hypothesis that the variables are independent can be rejected.
crammers_v:  0.22875085839539938
```



NEXT STEP….

Our future work involves developing a predictive model to estimate house prices using various features, such as location, size, number of bedrooms, and other relevant attributes. This model aims to improve accuracy in price predictions by leveraging the detailed data we've gathered and analyzed.



THANK YOU !