Getting Into Business

Cole Moore

February 26, 2025

Dataset Overview

When was the data collected?

The data was collected in **2014**, with a time range from **May 2, 2014, to July 10, 2014**. It was uploaded to Kaggle and last updated **seven months ago**.

Where was the data acquired?

The data was acquired from **Kaggle (https://www.kaggle.com)**. The original source was **Zillow (https://www.zillow.com)**, a real estate website known for its housing market data.

How was the data acquired?

Zillow's **Economic Research Team** gathers, refines, and publishes housing and economic data from both **public and proprietary sources**. The core of Zillow's data comes from:

- **Public property records** (deeds, parcel information, transaction history).
- **Internal Zillow market analyses** using proprietary algorithms.
- **Government and private housing reports** for contextual insights.

These sources are used to calculate various housing metrics, explained in the next section.

Dataset Attributes

What are the attributes of this dataset?

The dataset consists of **18 attributes** that describe various characteristics of properties:

- **Date:** The date the property was sold.
- **Price:** The sale price of the property in USD, serving as the target variable in housing market analyses.
- **Bedroom:** The total number of bedrooms in the property, indicating the home's capacity.
- **Bathroom:** The total number of bathrooms in the property, including full and half-baths.
- **Sqft_living:** The total interior square footage of the home, representing the livable space.
- **Sqft_lot:** The total land area of the property, including the house and yard.
- **Floors:** The number of floors in the home, influencing the layout and design.
- **Waterfront:** A binary variable indicating whether the property is located on the waterfront (1 = Yes, 0 = No).
- **View:** An index ranging from **0 to 4**, where higher values indicate a better quality view from the property.
- **Condition:** An index ranging from **1 to 5**, where **1** represents poor condition and **5** represents excellent condition.
- **Sqft Above:** The total square footage of the home **excluding** the basement, reflecting the main living area.
- **Sqft Basement:** The total square footage of the basement area, which may or may not be finished living space.
- **Yr Built:** The year in which the property was originally constructed.
- **Yr Renovated:** The most recent year in which the property underwent major renovations or updates.
- **Street:** The street address of the property.
- **City:** The city where the property is located.
- **Statezip:** A combined variable containing both the **state** and **zip code** of the property.
- **Country:** The country where the property is located.

Data Types

What type of data do these attributes contain?

Data Type	Attributes
Nominal	Street, City, Statezip, Country, Waterfront
Ordinal	View, Condition
Interval	Yr Built, Yr Renovated
Ratio	Price, Bedroom, Bathroom, Sqft_living, Sqft_lot, Floors, Sqft Above, Sqft Basement

- **Nominal:** Categorical variables without a meaningful order, such as property location attributes.
- **Ordinal:** Ranked variables with a meaningful order but uneven intervals, such as **View** and **Condition** ratings.
- **Interval:** Numeric variables with meaningful differences but no true zero, such as **Yr Built** and **Yr Renovated**.
- **Ratio:** Continuous numerical attributes with a true zero, such as **Price**, **Sqft_living**, and **Sqft_lot**.

```r
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.3.3
```

```r
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.3.3
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
# Create a table for dataset attributes
attributes_table <- data.frame(
  Attribute = c("Date", "Price", "Bedroom", "Bathroom", "Sqft_living", "Sqft_lot", "Floors", "Waterfront",
                "View", "Condition", "Sqft Above", "Sqft Basement", "Yr Built", "Yr Renovated", "Street", "Cit
y",
                "Statezip", "Country"),
  Description = c("Date the property was sold",
                  "Sale price of the property in USD (target variable)",
                  "Number of bedrooms in the property",
                  "Number of bathrooms in the property (full and half-baths)",
                  "Total interior square footage of the home",
                  "Total land area of the property, including the house and yard",
                  "Number of floors in the home",
                  "Binary indicator for waterfront location (1 = Yes, 0 = No)",
                  "Index (0-4) rating the quality of the property's view",
                  "Index (1-5) rating the condition of the property",
                  "Total square footage of the home excluding the basement",
                  "Total square footage of the basement area",
                  "Year the property was originally constructed",
                  "Year of the most recent renovation",
                  "Street address of the property",
                  "City where the property is located",
                  "State and ZIP code of the property",
                  "Country where the property is located"),
  DataType = c("Date", "Ratio", "Ratio", "Ratio", "Ratio", "Ratio", "Ratio", "Nominal",
               "Ordinal", "Ordinal", "Ratio", "Ratio", "Interval", "Interval", "Nominal", "Nominal",
               "Nominal", "Nominal")
)

# Format and display the table
kable(attributes_table, caption = "Dataset Attributes and Descriptions") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"), full_width = TRUE) %>%
  column_spec(1, bold = TRUE, width = "15%") %>%
  column_spec(2, italic = TRUE, width = "65%") %>%
  column_spec(3, width = "20%") %>%
  row_spec(0, background = "lightgray")
```

Dataset Attributes and Descriptions

Attribute	Description	DataType
Date	*Date the property was sold*	Date
Price	*Sale price of the property in USD (target variable)*	Ratio
Bedroom	*Number of bedrooms in the property*	Ratio
Bathroom	*Number of bathrooms in the property (full and half-baths)*	Ratio
Sqft_living	*Total interior square footage of the home*	Ratio
Sqft_lot	*Total land area of the property, including the house and yard*	Ratio
Floors	*Number of floors in the home*	Ratio
Waterfront	*Binary indicator for waterfront location (1 = Yes, 0 = No)*	Nominal
View	*Index (0-4) rating the quality of the property’s view*	Ordinal
Condition	*Index (1-5) rating the condition of the property*	Ordinal
Sqft Above	*Total square footage of the home excluding the basement*	Ratio

Attribute	Description	DataType
Sqft Basement	*Total square footage of the basement area*	Ratio
Yr Built	*Year the property was originally constructed*	Interval
Yr Renovated	*Year of the most recent renovation*	Interval
Street	*Street address of the property*	Nominal
City	*City where the property is located*	Nominal
Statezip	*State and ZIP code of the property*	Nominal
Country	*Country where the property is located*	Nominal

```
# Load necessary libraries
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:kableExtra':
##
##     group_rows
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
library(knitr)
library(kableExtra)
library(ggplot2)


# Load the dataset (make sure the CSV is in your working directory or adjust the path)
df <- read_csv("USA Housing Dataset.csv")
```

```
## Rows: 4140 Columns: 18
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr   (4): street, city, statezip, country
## dbl  (13): price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterf...
## dttm  (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Select numeric columns
numeric_df <- df %>%
  select(where(is.numeric))

# Generate summary statistics
numeric_summary <- data.frame(
  Variable = names(numeric_df),
  Count = sapply(numeric_df, function(x) sum(!is.na(x))),
  Mean = sapply(numeric_df, mean, na.rm = TRUE),
  SD = sapply(numeric_df, sd, na.rm = TRUE),
  Min = sapply(numeric_df, min, na.rm = TRUE),
  Q1 = sapply(numeric_df, quantile, probs = 0.25, na.rm = TRUE),
  Median = sapply(numeric_df, median, na.rm = TRUE),
  Q3 = sapply(numeric_df, quantile, probs = 0.75, na.rm = TRUE),
  Max = sapply(numeric_df, max, na.rm = TRUE)
)

# Display as a formatted table using kable
kable(numeric_summary, digits = 2, caption = "Summary Statistics - Numeric Variables") %>%
  kable_styling(full_width = TRUE, bootstrap_options = c("striped", "hover")) %>%
  column_spec(1, bold = TRUE)
```

Summary Statistics - Numeric Variables

	Variable	Count	Mean	SD	Min	Q1	Median	Q3	Max
price	price	4140	553062.88	583686.45	0	320000.00	460000.00	659125.0	26590000.00
bedrooms	bedrooms	4140	3.40	0.90	0	3.00	3.00	4.0	8.00
bathrooms	bathrooms	4140	2.16	0.78	0	1.75	2.25	2.5	6.75
sqft_living	sqft_living	4140	2143.64	957.48	370	1470.00	1980.00	2620.0	10040.00
sqft_lot	sqft_lot	4140	14697.64	35876.84	638	5000.00	7676.00	11000.0	1074218.00
floors	floors	4140	1.51	0.53	1	1.00	1.50	2.0	3.50
waterfront	waterfront	4140	0.01	0.09	0	0.00	0.00	0.0	1.00
view	view	4140	0.25	0.79	0	0.00	0.00	0.0	4.00
condition	condition	4140	3.45	0.68	1	3.00	3.00	4.0	5.00
sqft_above	sqft_above	4140	1831.35	861.38	370	1190.00	1600.00	2310.0	8020.00
sqft_basement	sqft_basement	4140	312.29	464.35	0	0.00	0.00	602.5	4820.00

	Variable	Count	Mean	SD	Min	Q1	Median	Q3	Max
yr_built	yr_built	4140	1970.81	29.81	1900	1951.00	1976.00	1997.0	2014.00
yr_renovated	yr_renovated	4140	808.37	979.38	0	0.00	0.00	1999.0	2014.00

Missing and Empty Values

In this section, we examine whether the dataset contains any missing (`NA`) or empty (`""`) values and discuss how to handle them appropriately.

```
# Count missing or empty values in each column
# Identify missing (NA) values for all columns
missing_na <- sapply(df, function(x) sum(is.na(x)))

# Identify empty strings only for character columns
missing_empty <- sapply(df, function(x) {
  if (is.character(x)) sum(x == "")
  else 0
})

# Combine results
na_summary <- data.frame(
  Variable = names(df),
  Missing_Values = missing_na + missing_empty
)

# Display formatted table
kable(na_summary, caption = "Missing or Empty Values by Variable") %>%
  kable_styling(full_width = TRUE, bootstrap_options = c("striped", "hover")) %>%
  column_spec(1, bold = TRUE)
```

Missing or Empty Values by Variable

	Variable	Missing_Values
date	date	0
price	price	0
bedrooms	bedrooms	0
bathrooms	bathrooms	0
sqft_living	sqft_living	0
sqft_lot	sqft_lot	0
floors	floors	0
waterfront	waterfront	0
view	view	0
condition	condition	0

	Variable	Missing_Values
sqft_above	sqft_above	0
sqft_basement	sqft_basement	0
yr_built	yr_built	0
yr_renovated	yr_renovated	0
street	street	0
city	city	0
statezip	statezip	0
country	country	0

```
na_summary <- data.frame(Variable = names(na_summary), Missing_Values = na_summary)

# Display as formatted table
kable(na_summary, caption = "Missing or Empty Values by Variable") %>%
  kable_styling(full_width = TRUE, bootstrap_options = c("striped", "hover")) %>%
  column_spec(1, bold = TRUE)
```

Missing or Empty Values by Variable

	Variable	Missing_Values.Variable	Missing_Values.Missing_Values
date	Variable	date	0
price	Missing_Values	price	0
bedrooms	Variable	bedrooms	0
bathrooms	Missing_Values	bathrooms	0
sqft_living	Variable	sqft_living	0
sqft_lot	Missing_Values	sqft_lot	0
floors	Variable	floors	0
waterfront	Missing_Values	waterfront	0
view	Variable	view	0
condition	Missing_Values	condition	0
sqft_above	Variable	sqft_above	0
sqft_basement	Missing_Values	sqft_basement	0
yr_built	Variable	yr_built	0
yr_renovated	Missing_Values	yr_renovated	0

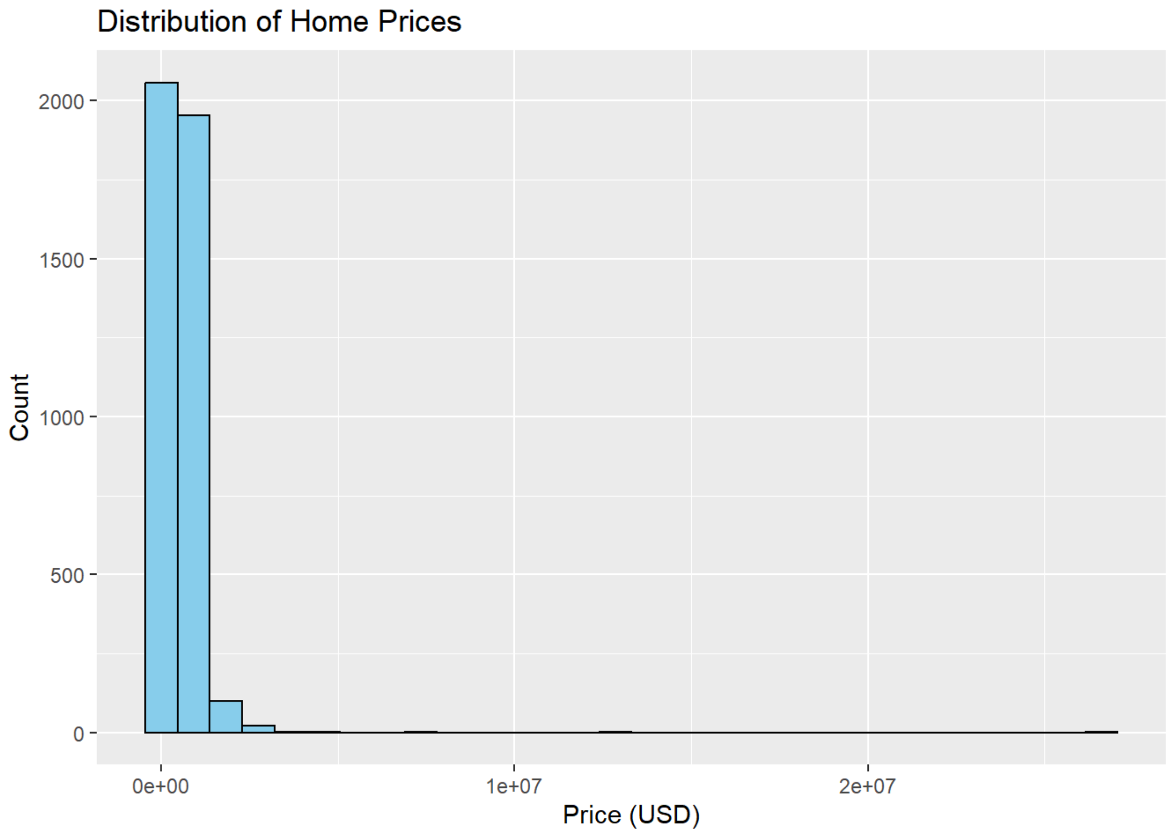
	Variable	Missing_Values.Variable	Missing_Values.Missing_Values
street	Variable	street	0
city	Missing_Values	city	0
statezip	Variable	statezip	0
country	Missing_Values	country	0

Exploratory Data Analysis

What is the distribution of home prices in the dataset?

Understanding the distribution of home prices helps identify typical price points and outliers — essential for evaluating affordability and investment potential.

```
ggplot(df, aes(x = price)) +
  geom_histogram(fill = "skyblue", bins = 30, color = "black") +
  labs(title = "Distribution of Home Prices", x = "Price (USD)", y = "Count")
```

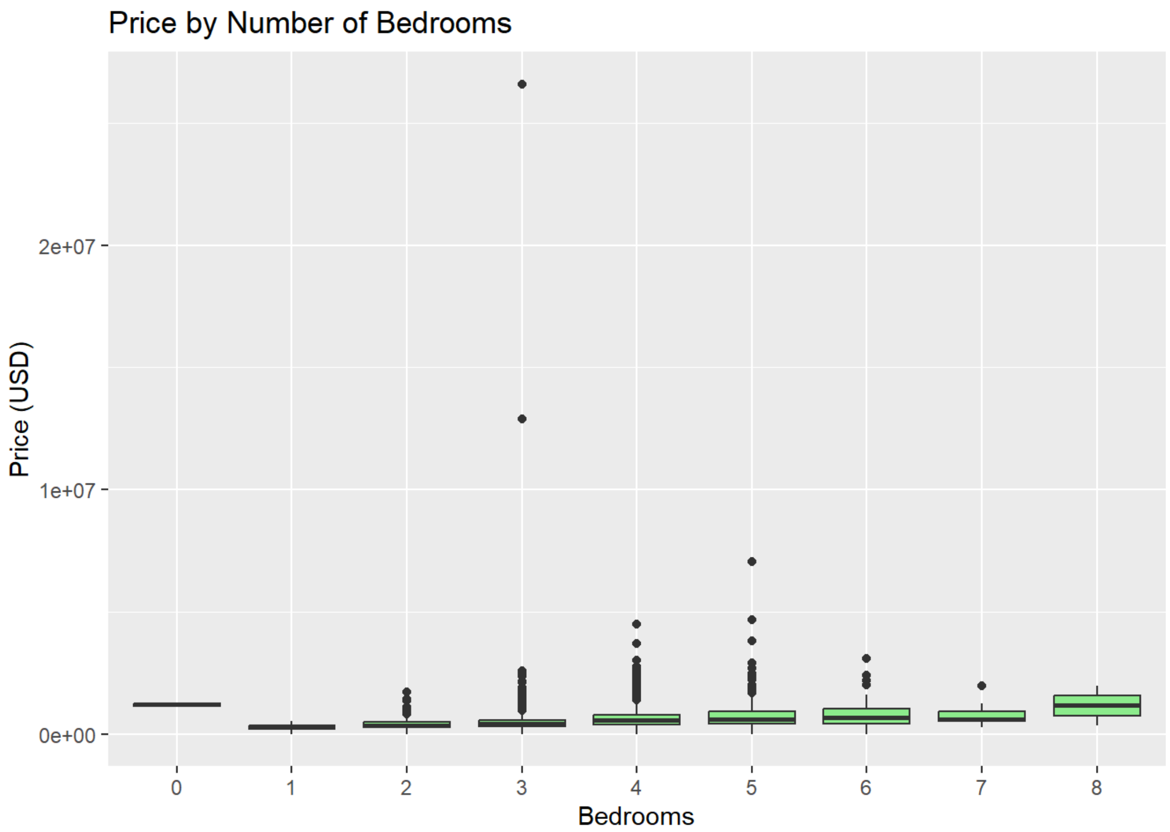


** Insight:** Most homes are priced under $600,000, with a clear right-skewed distribution. This suggests a small subset of high-end homes push the average upward.

How does the number of bedrooms affect home prices?

Bedrooms are a primary factor buyers consider. This plot explores whether more bedrooms tend to increase a property's market value.

```
ggplot(df, aes(x = factor(bedrooms), y = price)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Price by Number of Bedrooms", x = "Bedrooms", y = "Price (USD)")
```



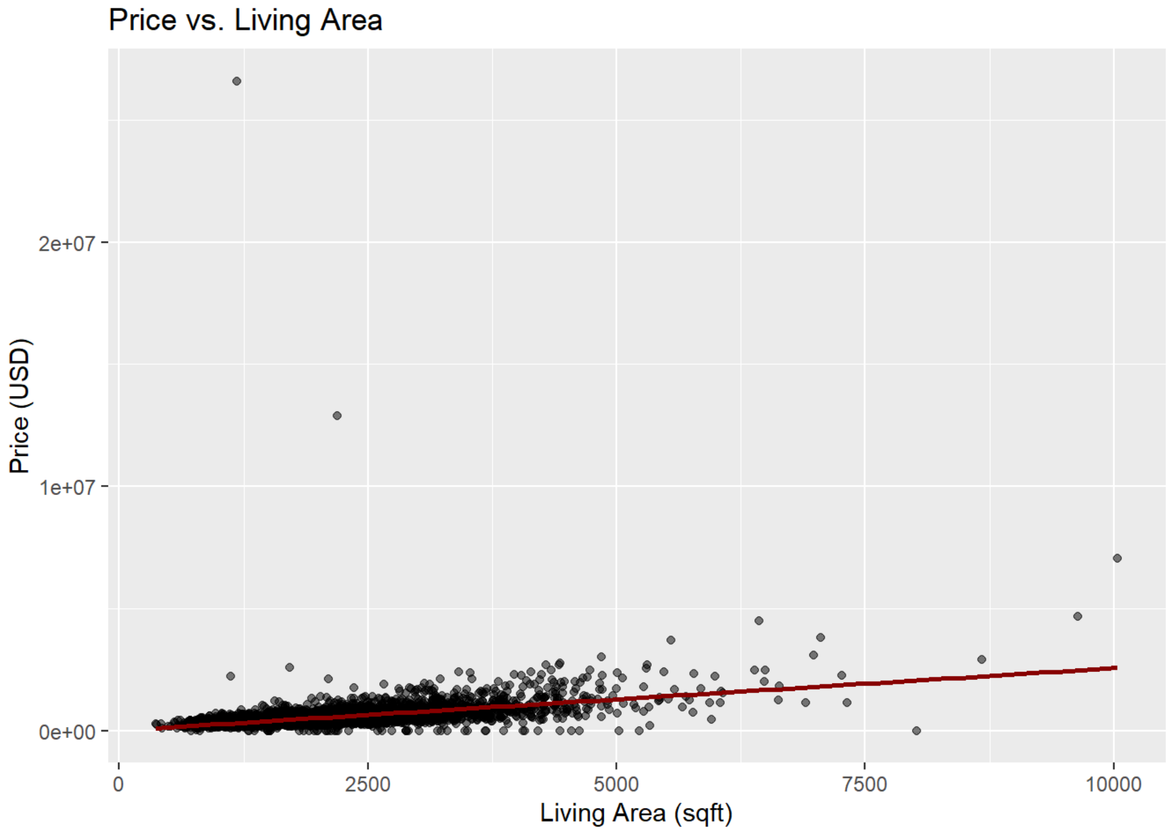
** Insight:** Homes with more bedrooms generally have higher prices, but there's substantial overlap. This suggests that other features (like square footage or location) also significantly affect price.

Is there a relationship between square footage and price?

We hypothesize that larger homes command higher prices. Let's visualize the relationship between interior living space and price.

```
ggplot(df, aes(x = sqft_living, y = price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "darkred") +
  labs(title = "Price vs. Living Area", x = "Living Area (sqft)", y = "Price (USD)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



** Insight:** There is a strong positive relationship between living area and price. Larger homes tend to be worth more, though with increasing variability at higher square footage levels.

3. Expanding Your Investment Knowledge

While this dataset offers a great snapshot of property characteristics and prices, supplementing it with additional data sources can provide deeper insights and improve investment decisions.

Additional Dataset: Zillow Home Value Index (ZHVI)

- **Source:** Zillow Research – ZHVI Data (https://www.zillow.com/research/data/)
- **Description:** The Zillow Home Value Index (ZHVI) tracks monthly median home values across regions, including ZIP codes, cities, counties, and metropolitan areas.

Why is this dataset useful?

- It provides time series data, allowing investors to analyze historical price trends and forecast future appreciation.
- It includes geographic variation, letting investors compare how property values change over time in different markets.

How does it complement your current data?

- The current dataset is cross-sectional (a snapshot in time), while the ZHVI adds a temporal dimension.
- By combining both, you could identify properties in regions that not only have good current value but also show strong long-term growth trends.
- It can help refine location-based investment decisions, guiding you toward markets with the best growth potential.

You can explore or download the ZHVI dataset here:
Zillow Home Value Index (ZHVI) (https://www.zillow.com/research/data/)

4. Communicating Your Findings

This analysis helps make real estate investment more approachable by exploring key features that influence home prices, such as square footage, number of bedrooms, and location. We've also addressed data quality by checking for missing values and visualized trends that affect investment decisions.

Even without prior real estate or data experience, readers can now: - Understand what attributes impact home value. - Identify how trends like larger living space or better condition contribute to pricing. - See how public datasets can be used to guide real-world investment strategy.

Our dataset gave us a snapshot of housing conditions across U.S. states. By examining average prices and property features, and supplementing with growth trend data (like Zillow's ZHVI), investors can target areas with both good current value and long-term appreciation potential.

Reproducibility and Data Access

The dataset used in this project is available on Kaggle:

Data Link: https://www.kaggle.com/datasets/fratzcan/usa-house-prices (https://www.kaggle.com/datasets/fratzcan/usa-house-prices)

Since Kaggle requires login and sometimes API authentication, it's recommended to manually download the CSV and load it like this:

```
# Load CSV after manually downloading from Kaggle
df <- read_csv("USA Housing Dataset.csv")
```

```
## Rows: 4140 Columns: 18
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr   (4): street, city, statezip, country
## dbl  (13): price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterf...
## dttm  (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```