# California Housing Prices Data Analysis

Abdallah chidjou

December 11, 2024

```
# Abdallah Chidjou
# Citation: (source of help: Lecture note, googling in general, stackoverflow, and chatgpt)
```

In this analysis, we explore a dataset containing information from California Housing Prices. The dataset includes the following features:

longitude: Geographic coordinate. latitude: Geographic coordinate. housing_median_age: Median age of the houses. total_rooms: Total number of rooms. total_bedrooms: Total number of bedrooms. population: Population in the block. households: Number of households. median_income: Median income of residents. median_house_value: Median house value. ocean_proximity: Proximity to the ocean (categorical).

I aim to derive insights and patterns characterizing this dataset.

- Describe the dataset and its variables.
- Perform statistical analyses and visualizations to understand key characteristics.
- Explore relationships between variables.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
```

```
# Load the dataset
file_path = 'California Housing Prices.csv'
housing_data = pd.read_csv(file_path)

# Inspect the dataset
initial_summary = {
    "Shape": housing_data.shape,
    "Columns": housing_data.columns.tolist(),
    "Data Types": housing_data.dtypes,
    "Missing Values": housing_data.isnull().sum()
}


for key, value in initial_summary.items():
    print(f"{key}:\n{value}\n")
```

```
## Shape:
```

```
## (20640, 10)
##
## Columns:
## ['longitude', 'latitude', 'housing_median_age', 'total_rooms', 'total_bedrooms', 'population', 'hous
##
## Data Types:
## longitude             float64
## latitude              float64
## housing_median_age    float64
## total_rooms           float64
## total_bedrooms        float64
## population            float64
## households            float64
## median_income         float64
## median_house_value    float64
## ocean_proximity        object
## dtype: object
##
## Missing Values:
## longitude               0
## latitude                0
## housing_median_age      0
## total_rooms             0
## total_bedrooms        207
## population              0
## households              0
## median_income           0
## median_house_value      0
## ocean_proximity         0
## dtype: int64
```

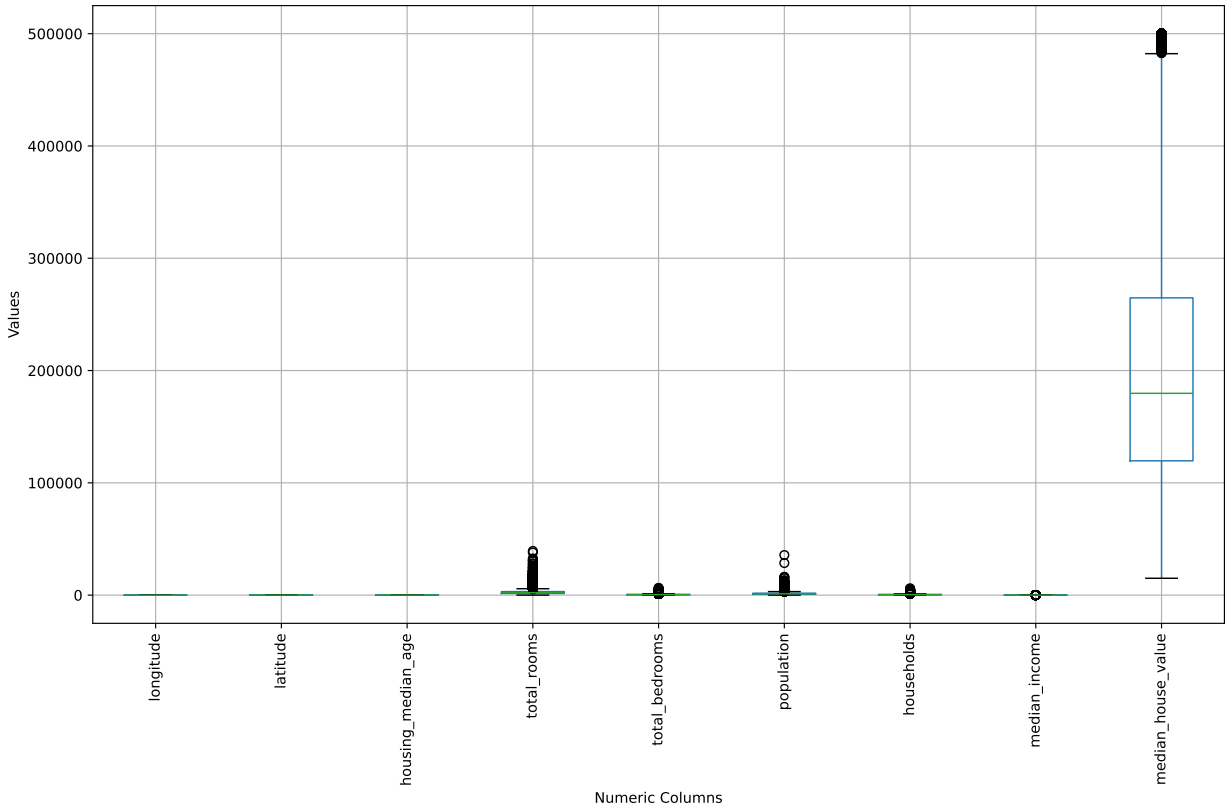
```python
# Address Missing Values
# Fill missing values in the 'total_bedrooms' column with the median value
median_value = housing_data['total_bedrooms'].median()
housing_data['total_bedrooms'] = housing_data['total_bedrooms'].fillna(median_value)

# Generate Descriptive Statistics
numeric_summary = housing_data.describe()
numeric_summary
```

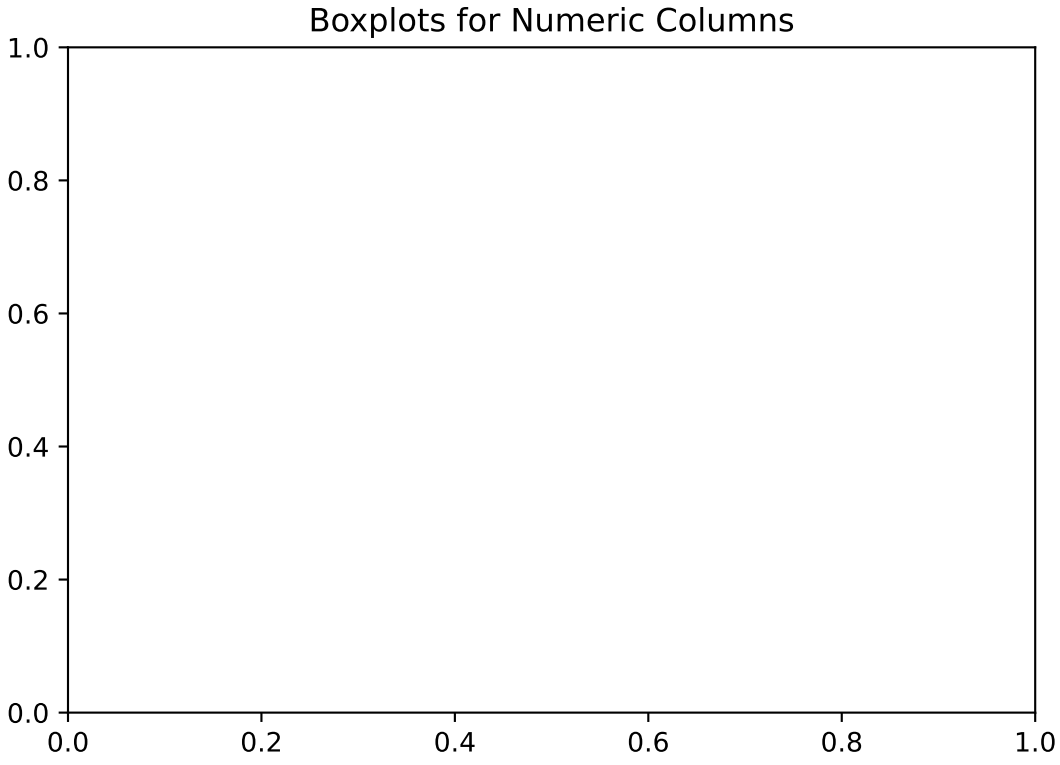
```
##           longitude      latitude  ...  median_income  median_house_value
## count  20640.000000  20640.000000  ...   20640.000000        20640.000000
## mean    -119.569704     35.631861  ...       3.870671       206855.816909
## std        2.003532      2.135952  ...       1.899822       115395.615874
## min     -124.350000     32.540000  ...       0.499900        14999.000000
## 25%     -121.800000     33.930000  ...       2.563400       119600.000000
## 50%     -118.490000     34.260000  ...       3.534800       179700.000000
## 75%     -118.010000     37.710000  ...       4.743250       264725.000000
## max     -114.310000     41.950000  ...      15.000100       500001.000000
##
## [8 rows x 9 columns]
```

```
# Display updated missing values count to confirm resolution
updated_missing_values = housing_data.isnull().sum()

# Printing the updated missing values
updated_missing_values
```

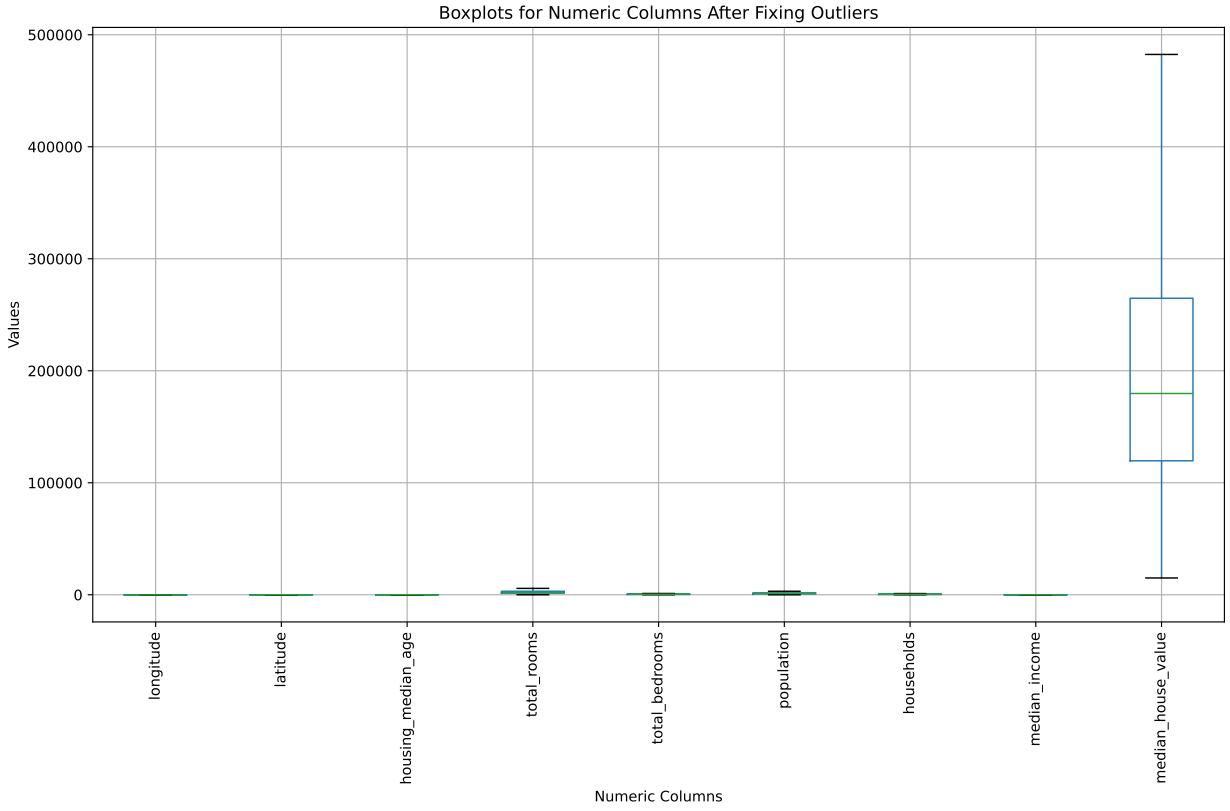
```
## longitude             0
## latitude              0
## housing_median_age    0
## total_rooms           0
## total_bedrooms        0
## population            0
## households            0
## median_income         0
## median_house_value    0
## ocean_proximity       0
## dtype: int64
```

```
# Create boxplots for numeric columns
numeric_columns = housing_data.select_dtypes(include=['float64', 'int64']).columns
plt.title("Boxplots for Numeric Columns")

# Generate boxplots for each numeric column
plt.figure(figsize=(12, 8))
housing_data[numeric_columns].boxplot(rot=90)
plt.xlabel("Numeric Columns")
plt.ylabel("Values")
plt.tight_layout()
plt.show()
```


500000
400000
300000
200000
100000
0
Values
longitude
latitude
housing_median_age
total_rooms
total_bedrooms
population
households
median_income
median_house_value
Numeric Columns


Boxplots for Numeric Columns
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

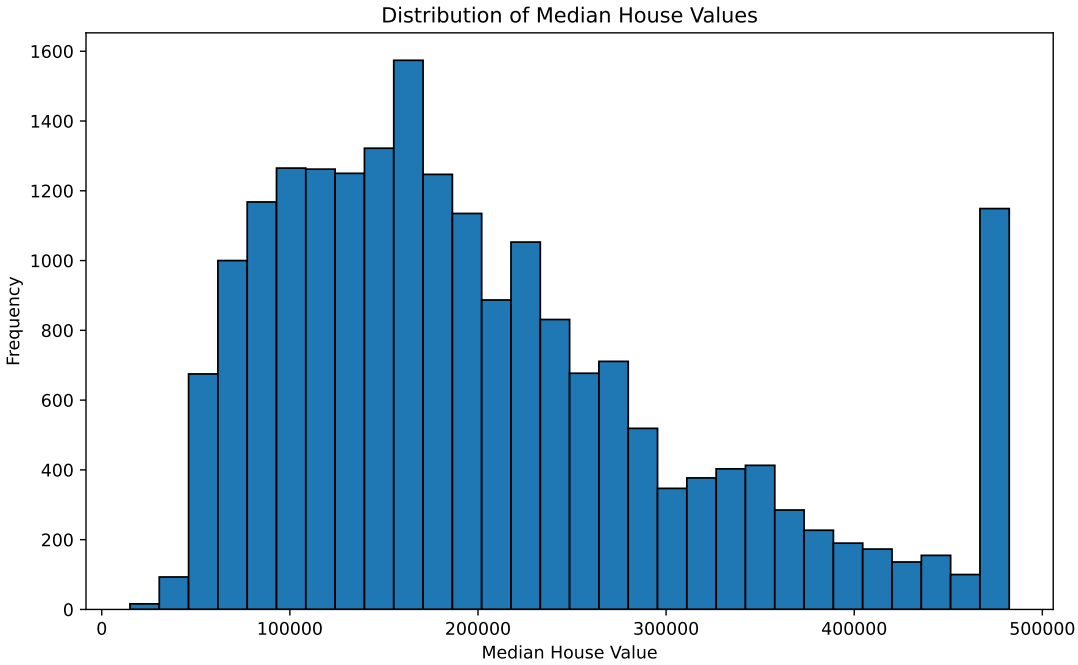
```python
# Fix Outliers
# Use the IQR method to handle outliers
for column in numeric_columns:
    Q1 = housing_data[column].quantile(0.25)
    Q3 = housing_data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    # Capping outliers
    housing_data[column] = housing_data[column].apply(lambda x: upper_bound if x > upper_bound else (lo

# Recheck boxplots after fixing outliers
plt.figure(figsize=(12, 8))
plt.title("Boxplots for Numeric Columns After Fixing Outliers")

# Generate updated boxplots for each numeric column
housing_data[numeric_columns].boxplot(rot=90)
plt.xlabel("Numeric Columns")
plt.ylabel("Values")
plt.tight_layout()
plt.show()
```



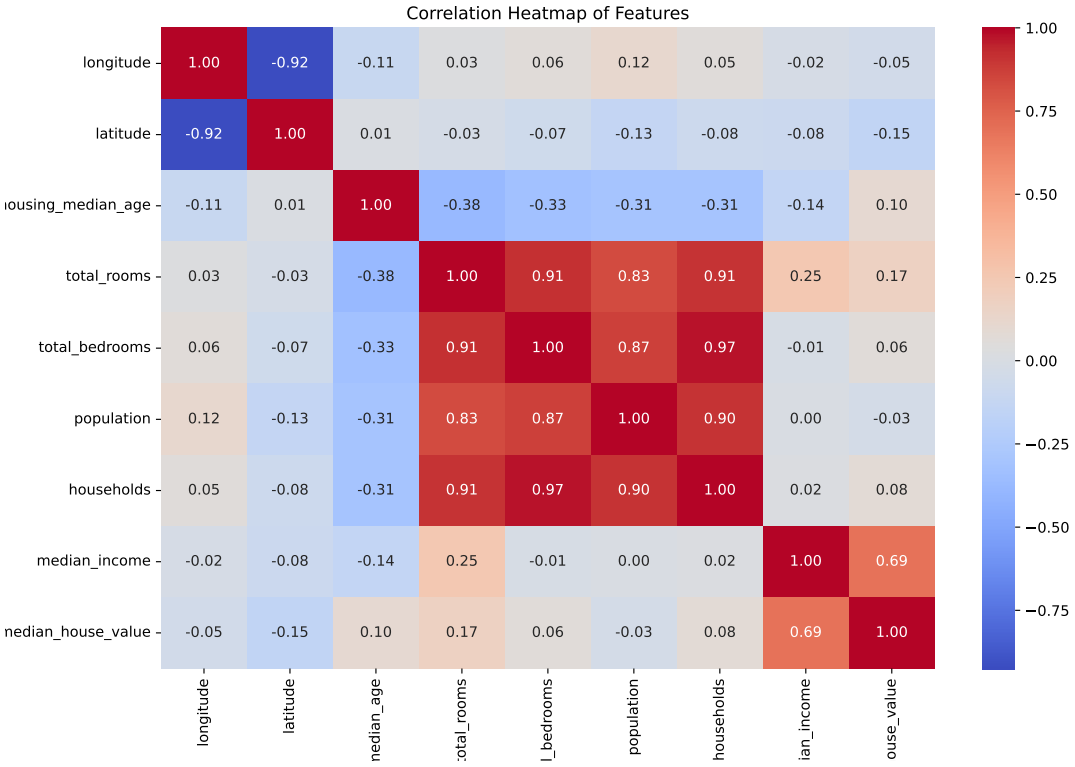

```python
# Scatter Plot: Median House Value vs. Median Income
plt.figure(figsize=(10, 6))
plt.scatter(housing_data['median_income'], housing_data['median_house_value'], alpha=0.5)
plt.title("Median House Value vs. Median Income")
```

```
plt.xlabel("Median Income")
plt.ylabel("Median House Value")
plt.grid(True)
plt.show()
```


```
# Histogram: Distribution of Median House Value
plt.figure(figsize=(10, 6))
plt.hist(housing_data['median_house_value'], bins=30, edgecolor='k')
plt.title("Distribution of Median House Values")
plt.xlabel("Median House Value")
plt.ylabel("Frequency")
plt.show()
```



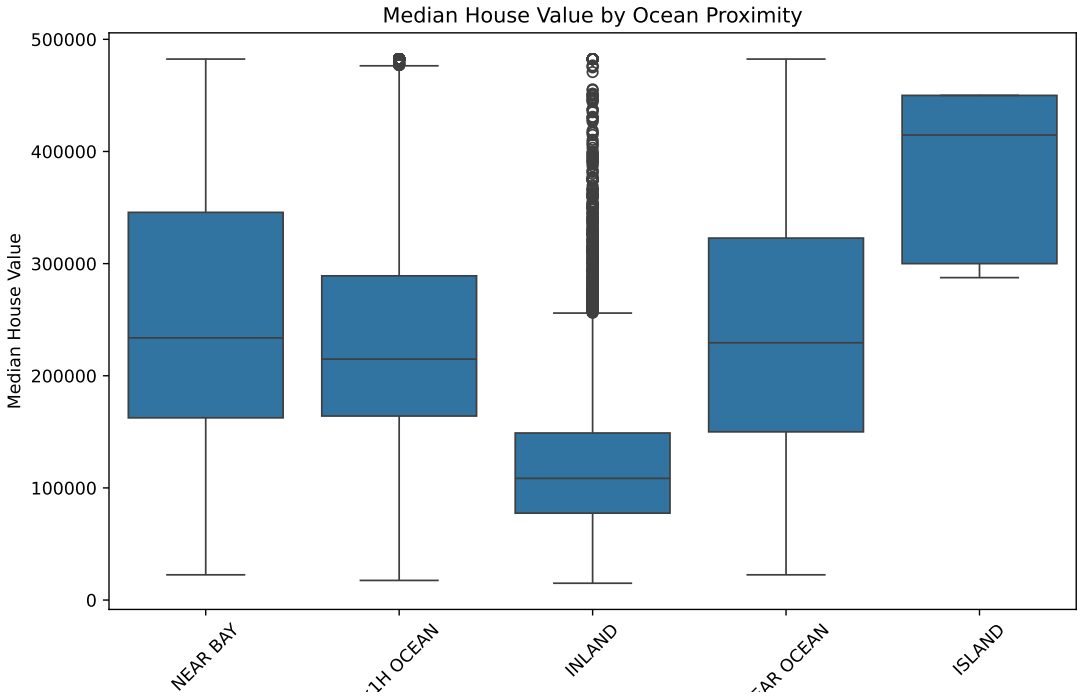

```
plt.figure(figsize=(12, 8))
numeric_only_data = housing_data.select_dtypes(include=['float64', 'int64'])
correlation_matrix = numeric_only_data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap of Features")
plt.show()
```




```
# Boxplot: Median House Value by Ocean Proximity
plt.figure(figsize=(10, 6))
sns.boxplot(x='ocean_proximity', y='median_house_value', data=housing_data)
plt.title("Median House Value by Ocean Proximity")
plt.xlabel("Ocean Proximity")
plt.ylabel("Median House Value")
plt.xticks(rotation=45)
```

```
## ([0, 1, 2, 3, 4], [Text(0, 0, 'NEAR BAY'), Text(1, 0, '<1H OCEAN'), Text(2, 0, 'INLAND'), Text(3, 0,
```

```
plt.show()
```


Median House Value by Ocean Proximity
Median House Value
500000
400000
300000
200000
100000
0
NEAR BAY
INLAND
ISLAND