Name: Abhinay Kumar

Roll No: 2201CS04

Course Code: CS502 Assignment 1.

1. Introduction

This report describes an analysis using the **California Housing** dataset with the goal of predicting median house values. We apply Exploratory Data Analysis (EDA), Linear Regression, demonstrate the Central Limit Theorem (CLT), analyze bias-variance trade-off.

---

2. Dataset Description

- **Source / Loading Method:** fetch_california_housing(as_frame=True) from scikit-learn.
- **Samples:** 20,640
- **Features (8):** MedInc, HouseAge, AveRooms, AveBedrms, Population, AveOccup, Latitude, Longitude.
- **Target:** MedHouseVal — median house value of districts (in units of $100,000).
- **Data Quality:** No missing values. All features are continuous. Some features (e.g. Population, AveRooms, AveOccup) show high variance/skew.

---

3. Exploratory Data Analysis (EDA)

3.1 Summary Statistics

[Insert here your printed output of df.describe(), df.info()]

3.2 Distributions of Features

Insert histograms of several features (e.g. MedInc, Population, etc.)

Screenshot / Plot:

3.3 Correlation and Feature-Target Relationships

Correlation Matrix:

Plot:

Scatter Plots:

- MedHouseVal vs MedInc
  Plot:
- Other features vs target: [HouseAge, AveRooms etc.]
  Insert respective plots.

3.4 Geographic Trends

Plot:

Interpretation: (e.g. districts with higher income and closer to coast tend to have higher house values.)

3.5 CLT Demonstration

Description: We sampled the target variable (MedHouseVal) many times (n=1000 samples of size 30 each), took sample means, and plotted the distribution of the sample means.

Plot:

Observation: The distribution of sample means approximates a normal distribution, even though the underlying target distribution is skewed.

---

4. Linear Regression Model

4.1 Preprocessing & Model Setup

- Split data: 80% train, 20% test (random_state=42).
- Pipeline: StandardScaler + LinearRegression.

4.2 Performance on Test Set

- **Test R^2:** ~ 0.576
- **Test RMSE:** ~ 0.746

4.3 Cross-Validation

- 5-fold CV R^2 scores: [0.5487, 0.4682, 0.5508, 0.5370, 0.6605]
- Mean CV R^2: ~ 0.553

---

5. Residuals & Error Analysis

Plot:

Interpretation:

- The residuals are scattered around zero, but there may be some pattern: e.g. error increases / bias in some ranges.
- Possible heteroscedasticity (spread of residuals increases for certain predicted ranges) etc.

---

6. Bias-Variance Trade-Off: Learning Curve

Plot:

Discussion:

- Training R^2 increases as data size grows, but validation R^2 (cross-validation) increases more slowly.
- There remains a gap between training and validation R^2, implying some variance. But overall, both curves flatten out suggesting adding more data beyond a point provides diminishing returns.
- Also suggests model has some bias (constraint by linear assumption).

---

7. Results Summary

Metric	**Value**
Test R^2	~ 0.576
Test RMSE	~ 0.746
Mean CV R^2	~ 0.553

These show moderate predictive ability of the linear model — it explains over half the variance in house values, but errors are still large and many regions are not well predicted.

---

8. Conclusion

This baseline model using simple linear regression achieves a reasonable level of performance for predicting median house values in California: explaining about **57%** of variance. However, significant error remains, especially in certain regions / values. With further feature engineering, model complexity, and additional spatial or external data, the predictive performance could likely be improved.

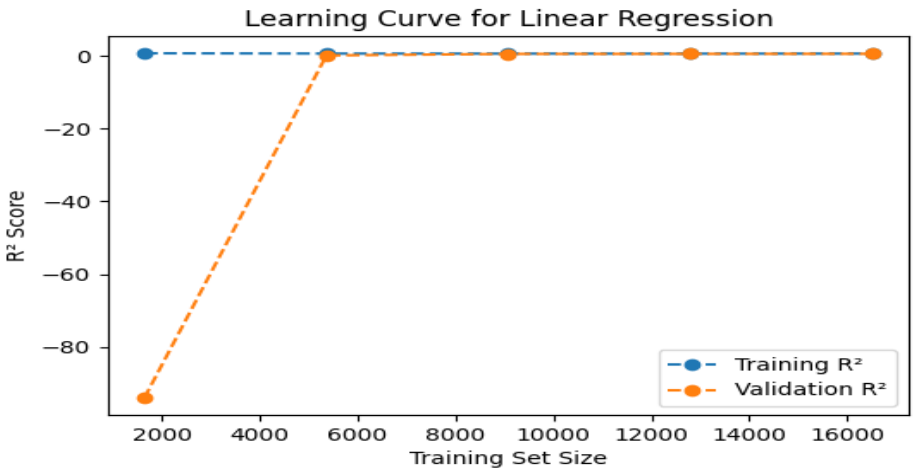
9. Code Appendix

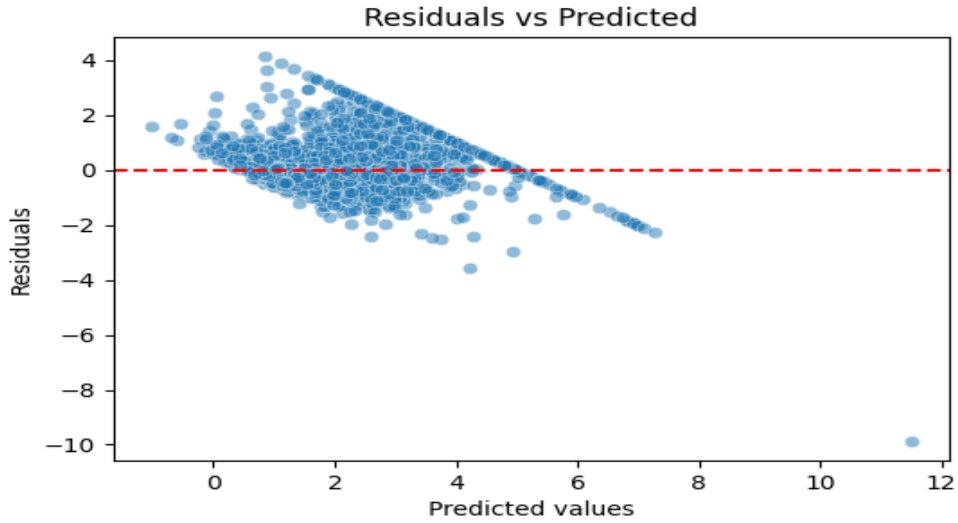
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split, cross_val_score,
learning_curve
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline
from sklearn.metrics import mean_squared_error, r2_score

# 1. Load data
housing = fetch_california_housing(as_frame=True)
df = housing.frame  # has data + target

# Optional: save
df.to_csv('data/california_housing.csv', index=False)

# 2. EDA
print(df.head())
print(df.info())
print(df.describe())

# Check missing values
print("Missing values per column:\n", df.isnull().sum())

# Distributions
df.hist(bins=30, figsize=(15,10), edgecolor='black')
plt.tight_layout()
plt.savefig('plots/hist_all_features.png')

# Correlation matrix
corr_mat = df.corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr_mat, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Matrix')
plt.savefig('plots/corr_matrix.png')
```

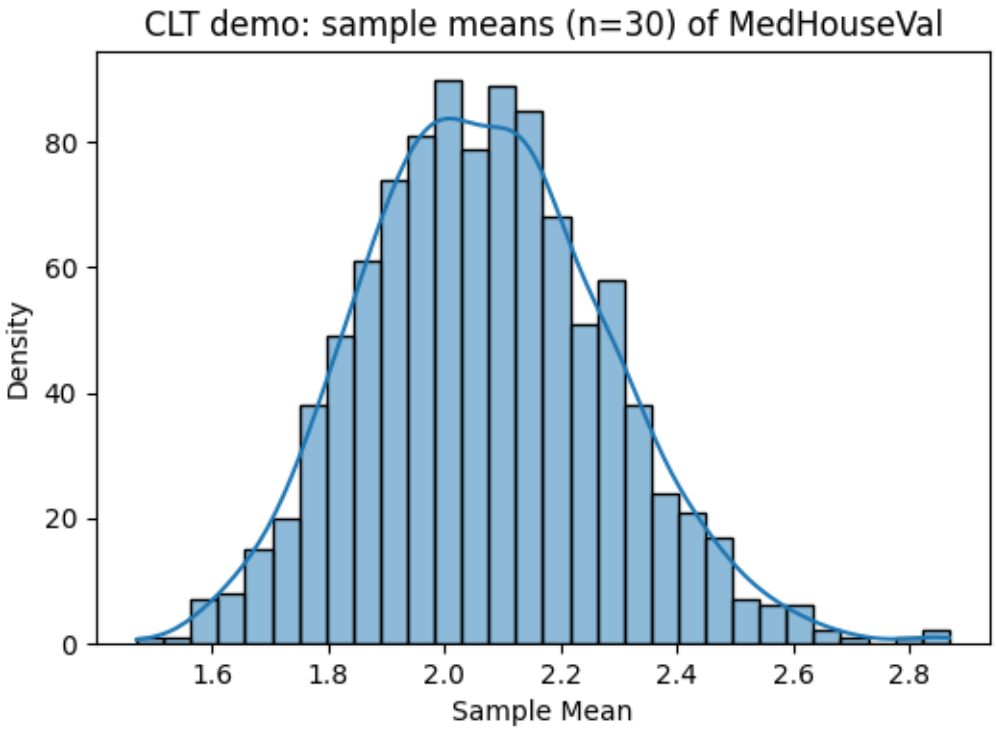
```
# Scatter plots target vs important features
important = ['MedInc', 'HouseAge', 'AveRooms', 'Latitude', 'Longitude']
for feat in important:
  plt.figure(figsize=(6,4))
  sns.scatterplot(x=df[feat], y=df['MedHouseVal'], alpha=0.5)
  plt.xlabel(feat)
  plt.ylabel('MedHouseVal')
  plt.title(f'MedHouseVal vs {feat}')
  plt.savefig(f'plots/scatter_{feat}_vs_target.png')

# Geographical scatter
plt.figure(figsize=(8,6))
sns.scatterplot(data=df.sample(5000, random_state=42),
        x='Longitude', y='Latitude',
        hue='MedHouseVal', size='Population',
        palette='viridis', alpha=0.6, sizes=(20,200))
plt.title('House Value by Location (coast vs inland)')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.savefig('plots/geo_scatter.png')

# 3. CLT demo (sample means of target)
target = df['MedHouseVal'].values
n_samples = 1000
sample_size = 30
means = [target[np.random.randint(0, len(target), sample_size)].mean() for _ in
range(n_samples)]
plt.figure(figsize=(6,4))
sns.histplot(means, bins=30, kde=True)
plt.title(f'CLT demo: sample means (n={sample_size}) of MedHouseVal')
plt.xlabel('Sample Mean')
plt.ylabel('Density')
plt.savefig('plots/clt_means.png')

# 4. Preprocessing, Train/Test split
X = df.drop(columns=['MedHouseVal'])
y = df['MedHouseVal']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 5. Model training: Linear Regression with scaling
```

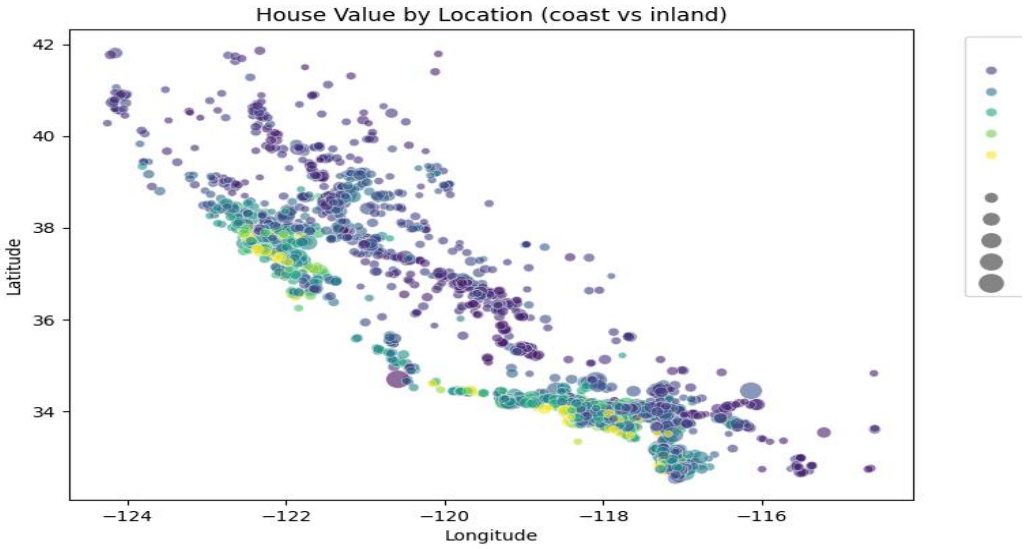
```
pipe = make_pipeline(StandardScaler(), LinearRegression())
pipe.fit(X_train, y_train)
y_pred = pipe.predict(X_test)

# Metrics on test set
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)
print(f"Test RMSE: {rmse:.4f}")
print(f"Test R²: {r2:.4f}")

# Residual plot
residuals = y_test - y_pred
plt.figure(figsize=(6,4))
sns.scatterplot(x=y_pred, y=residuals, alpha=0.5)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Predicted values')
plt.ylabel('Residuals')
plt.title('Residuals vs Predicted')
plt.savefig('plots/residuals_vs_predicted.png')

# 6. Cross-validation
cv_scores = cross_val_score(pipe, X, y, cv=5, scoring='r2')
print("5-fold CV R² scores:", cv_scores)
print("Mean CV R²:", cv_scores.mean())

# 7. Learning curve (bias-variance)
train_sizes, train_scores, test_scores = learning_curve(pipe, X, y, cv=5,
                                        train_sizes=np.linspace(0.1,1.0,5),
                                        scoring='r2', random_state=42)
train_mean = np.mean(train_scores, axis=1)
test_mean = np.mean(test_scores, axis=1)
plt.plot(train_sizes, train_mean, 'o--', label='Training R²')
plt.plot(train_sizes, test_mean, 'o--', label='Validation R²')
plt.xlabel('Training Set Size')
plt.ylabel('R² Score')
plt.title('Learning Curve for Linear Regression')
plt.legend()
plt.savefig('plots/learning_curve.png')
```

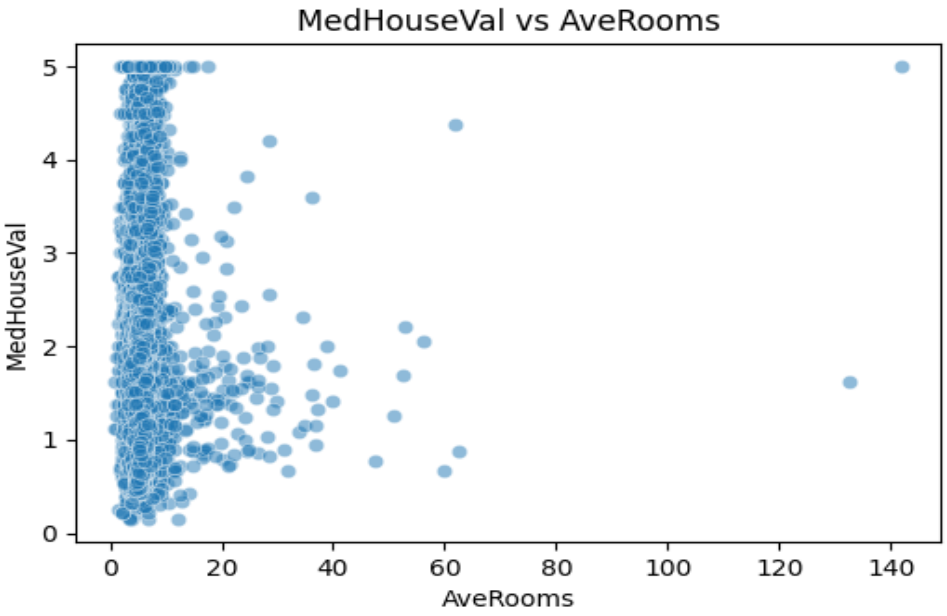
Plots:

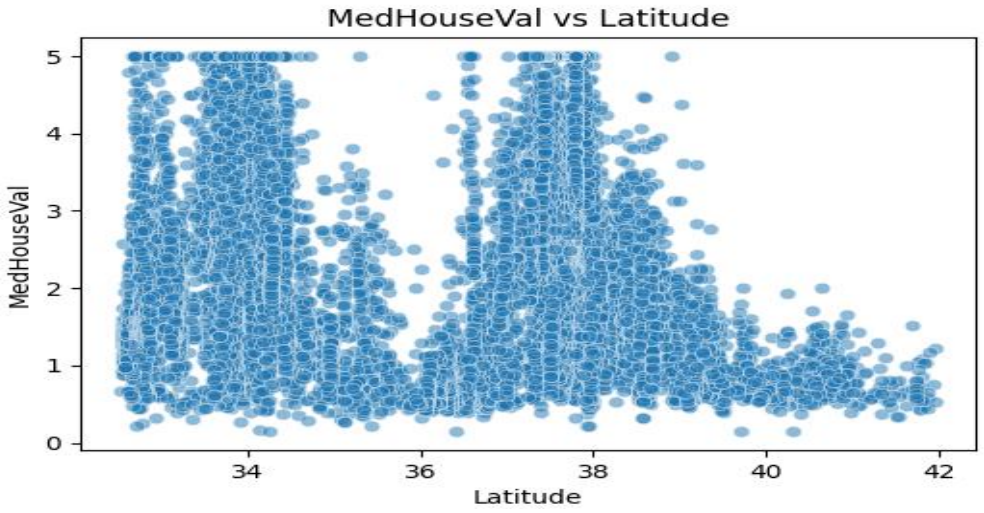

Learning Curve for Linear Regression
R² Score
0
−20
−40
−60
−80
2000
4000
6000
8000
10000
12000
14000
16000
Training Set Size
Training R²
Validation R²

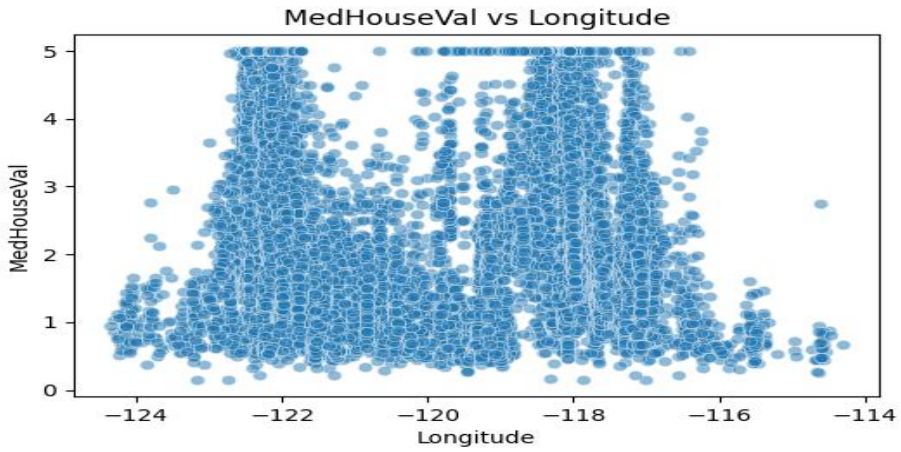

Residuals vs Predicted
Residuals
4
2
0
−2
−4
−6
−8
−10
0
2
4
6
8
10
12
Predicted values


CLT demo: sample means (n=30) of MedHouseVal
Density
Sample Mean
0
20
40
60
80
1.6
1.8
2.0
2.2
2.4
2.6
2.8


House Value by Location (coast vs inland)
Latitude
Longitude
42
40
38
36
34
−124
−122
−120
−118
−116

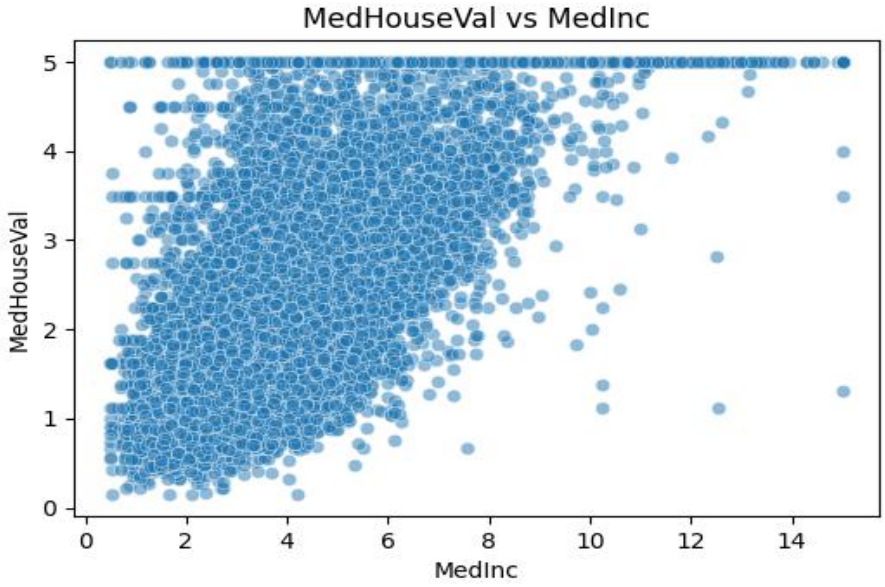

MedHouseVal vs AveRooms
MedHouseVal
AveRooms
0
1
2
3
4
5
0
20
40
60
80
100
120
140

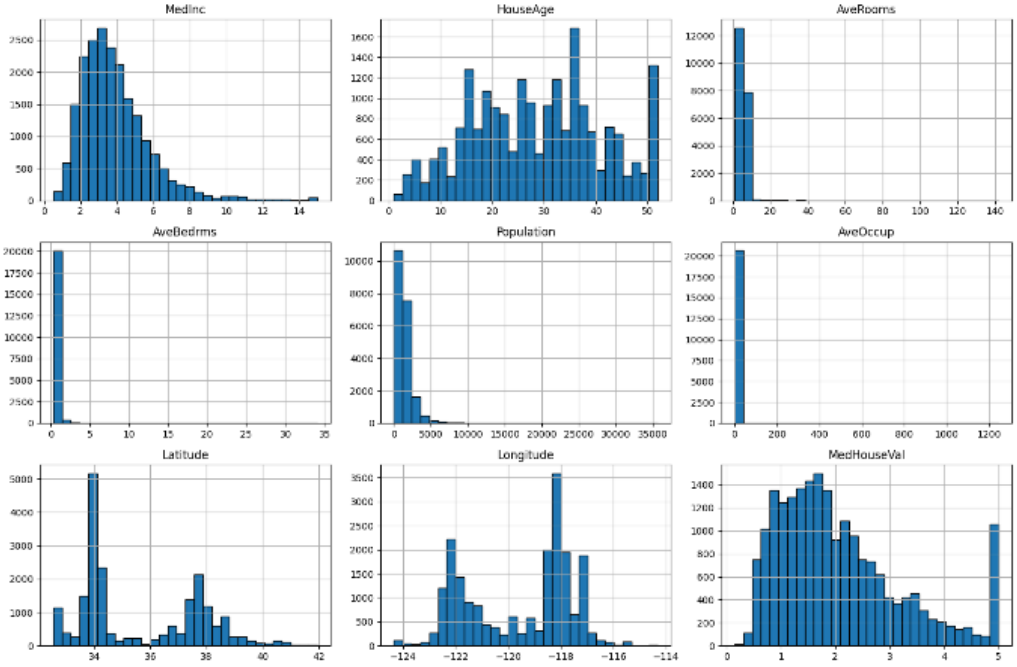

MedHouseVal vs Latitude
MedHouseVal
Latitude
0
1
2
3
4
5
34
36
38
40
42


MedHouseVal vs Longitude
MedHouseVal
Longitude
0
1
2
3
4
5
−124
−122
−120
−118
−116
−114

Correlation Matrix
MedInc
HouseAge
AveRooms
AveBedrms
Population
AveOccup
Latitude
Longitude
MedHouseVal
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

MedHouseVal vs HouseAge
MedHouseVal
HouseAge
0
1
2
3
4
5
0
10
20
30
40
50


MedHouseVal vs MedInc
MedHouseVal
MedInc


MedInc
HouseAge
AveRooms
AveBedrms
Population
AveOccup
Latitude
Longitude
MedHouseVal