Nicole Gallo
D600 – Statistical Data Mining
July 15, 2025
UGN1 Task 1: Linear Regression Analysis

A. Create your subgroup and project in GitLab using the provided web link by doing the following:

- Clone the project to the IDE.
  ```
  git clone https://gitlab.com/wgu-gitlab-environment/students/d600-statistical-data-mining.git
  ```
- Commit with a message and push when you complete each requirement listed in parts C2 through D4.
- Submit a copy of the GitLab repository URL in the "Comments to Evaluator" section when you submit this assessment.
- Submit a copy of the repository branch history retrieved from your repository, which must include the commit messages and dates.

B. Describe the purpose of this data analysis by doing the following:

1. Propose one research question that is relevant to a real-world organizational situation captured in the provided dataset that you will answer using multiple linear regression in the initial model.

   How do housing features (square footage, number of bedrooms, and backyard space) influence housing prices?

2. Define one goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.

   The goal of the data analysis for this research question is to identify which housing characteristics most significantly affect the sale prices. This can support more accurate pricing and investment decisions.

C. Summarize the data preparation process for multiple linear regression analysis by doing the following:

1. Identify the dependent and all independent variables that are required to answer the research question and justify your selection of variables.

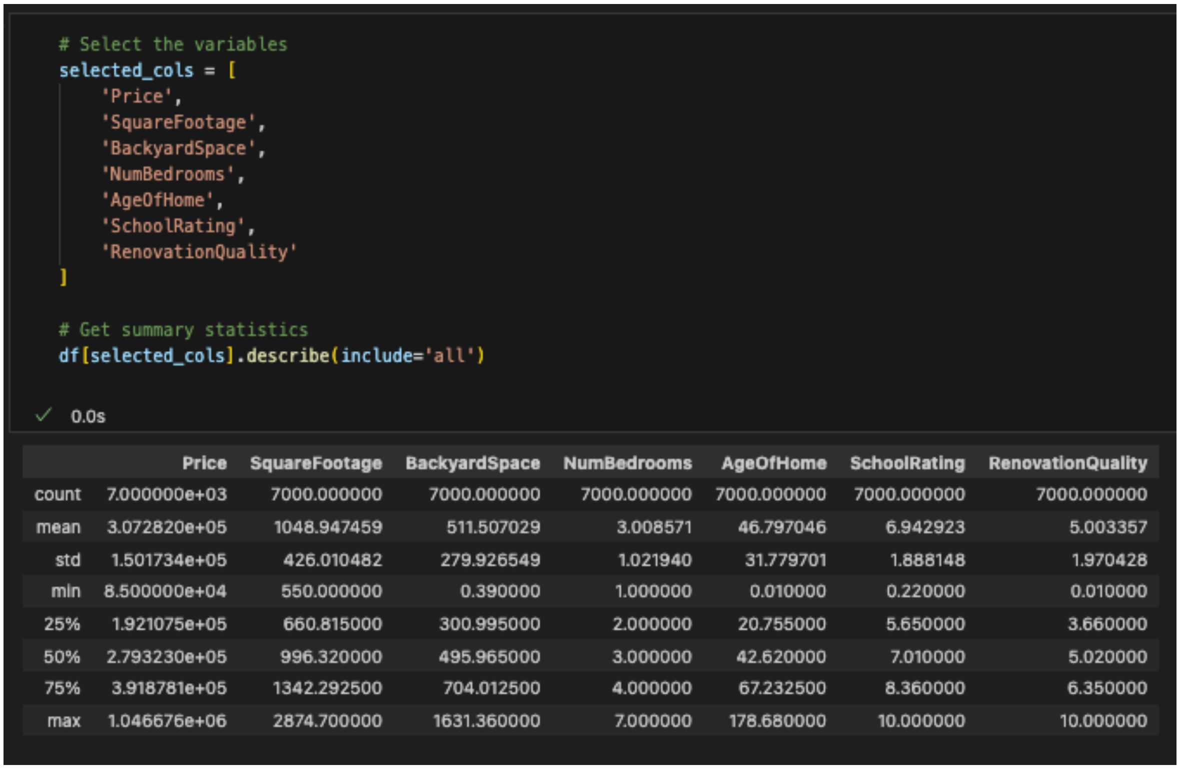
   Dependent Variable:
   The dependent variable is `Price`, as it represents our outcome of wanting to predict the price a house sells for.

   Independent Variables:
   - `SquareFootage`: larger homes generally command higher prices.
   - `BackyardSpace`: more land tends to add value.
   - `NumBedrooms`: more bedrooms may increase appeal to buyers.
   - `AgeOfHome`: newer homes may sell for more due to updated features.

- `SchoolRating:` higher-rated schools often boost neighborhood property values
- `RenovationQuality:` higher renovation quality can increase the sale price

These variables are quantitative, continuous, and directly related to property value.

2. Describe the dependent variable and all independent variables from part C1 using descriptive statistics (counts, means, modes, ranges, min/max), including a screenshot of the descriptive statistics output for each of these variables.



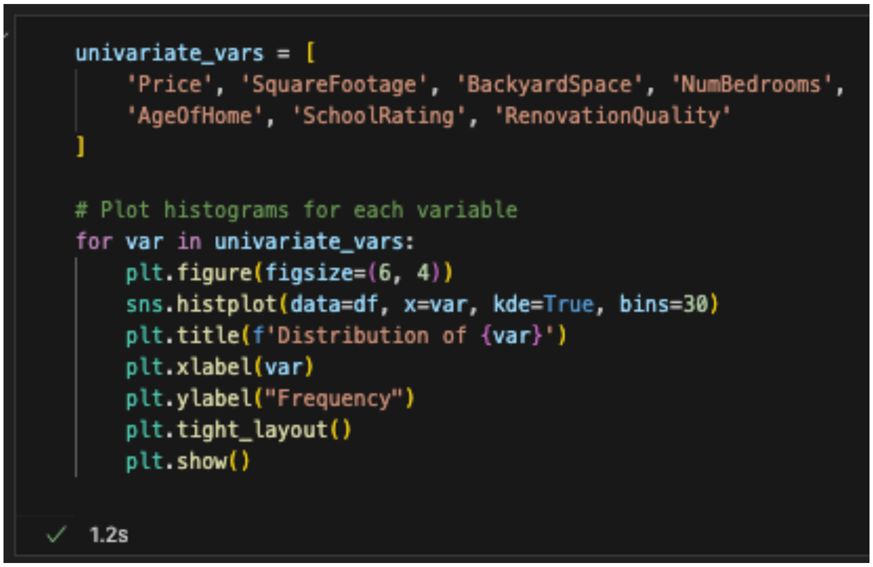
```python
# Select the variables
selected_cols = [
    'Price',
    'SquareFootage',
    'BackyardSpace',
    'NumBedrooms',
    'AgeOfHome',
    'SchoolRating',
    'RenovationQuality'
]

# Get summary statistics
df[selected_cols].describe(include='all')
```

✓ 0.0s

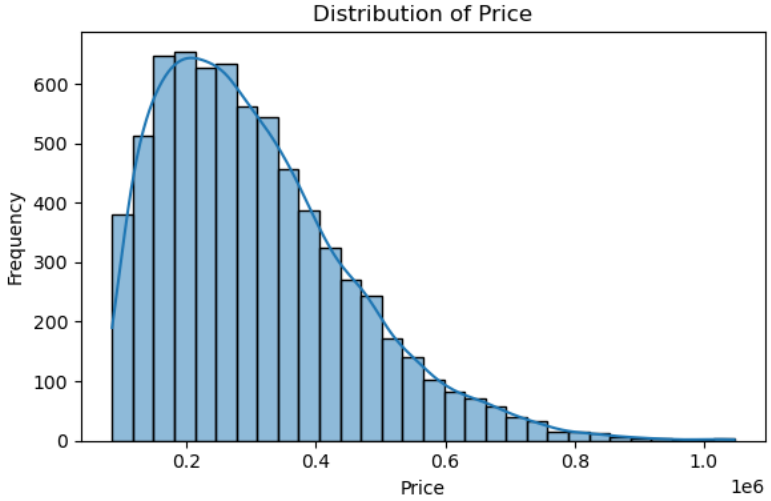
```
              Price  SquareFootage  BackyardSpace  NumBedrooms    AgeOfHome  SchoolRating  RenovationQuality
count  7.000000e+03    7000.000000    7000.000000  7000.000000  7000.000000   7000.000000        7000.000000
mean   3.072820e+05    1048.947459     511.507029     3.008571    46.797046      6.942923           5.003357
std    1.501734e+05     426.010482     279.926549     1.021940    31.779701      1.888148           1.970428
min    8.500000e+04     550.000000       0.390000     1.000000     0.010000      0.220000           0.010000
25%    1.921075e+05     660.815000     300.995000     2.000000    20.755000      5.650000           3.660000
50%    2.793230e+05     996.320000     495.965000     3.000000    42.620000      7.010000           5.020000
75%    3.918781e+05    1342.292500     704.012500     4.000000    67.232500      8.360000           6.350000
max    1.046676e+06    2874.700000    1631.360000     7.000000   178.680000     10.000000          10.000000
```

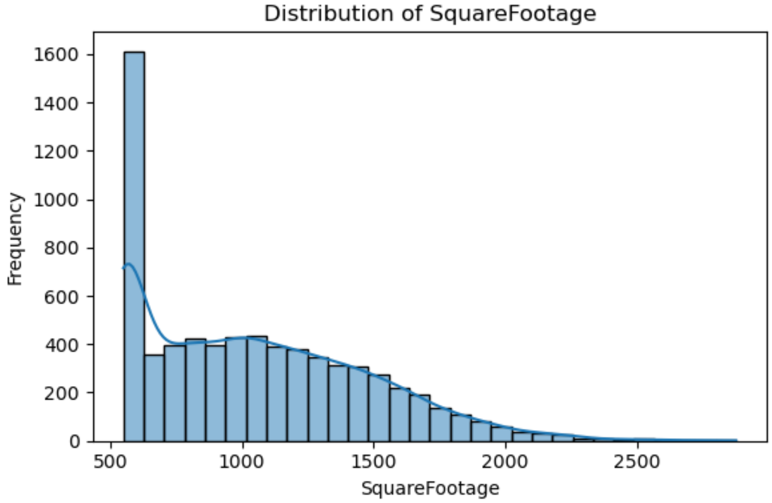
3. Generate univariate and bivariate visualizations of the distributions of the dependent and independent variables from part C1, including the dependent variable in the bivariate visualizations.

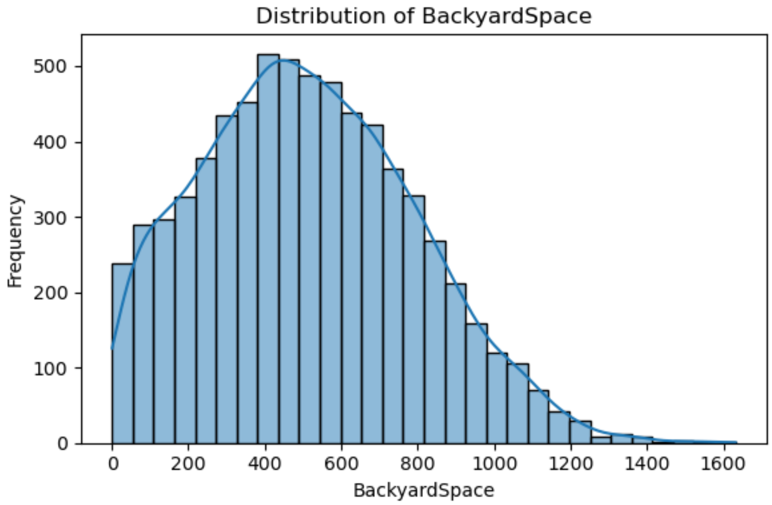
Univariate –



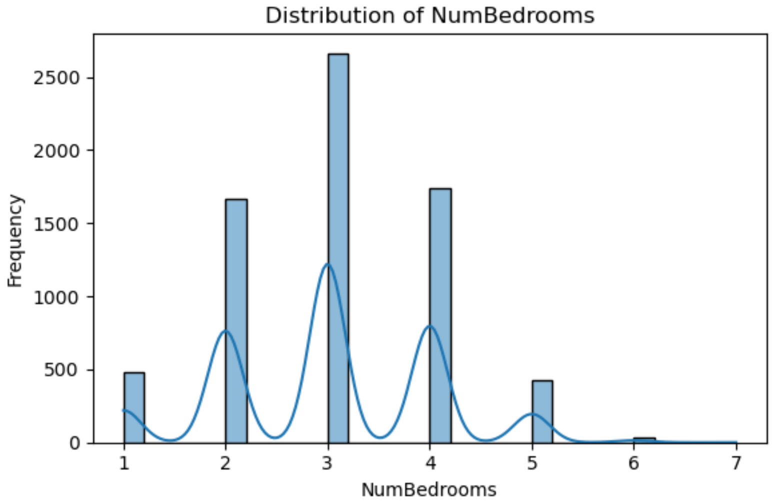
```python
univariate_vars = [
    'Price', 'SquareFootage', 'BackyardSpace', 'NumBedrooms',
    'AgeOfHome', 'SchoolRating', 'RenovationQuality'
]

# Plot histograms for each variable
for var in univariate_vars:
    plt.figure(figsize=(6, 4))
    sns.histplot(data=df, x=var, kde=True, bins=30)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel("Frequency")
    plt.tight_layout()
    plt.show()
```



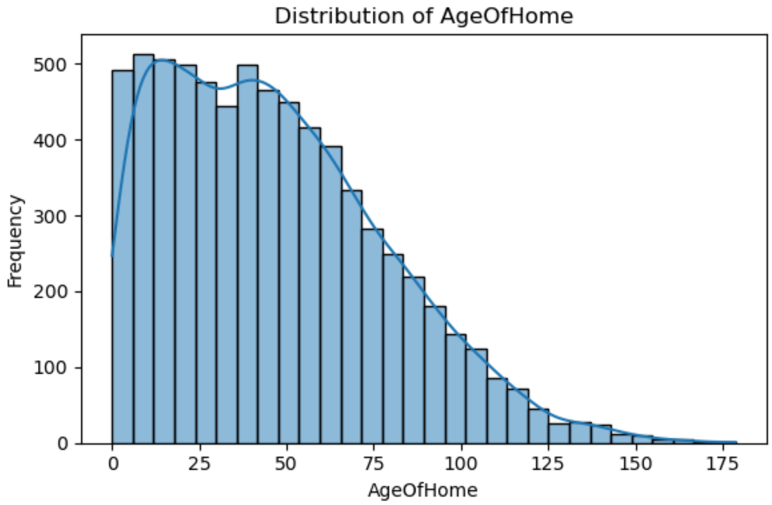


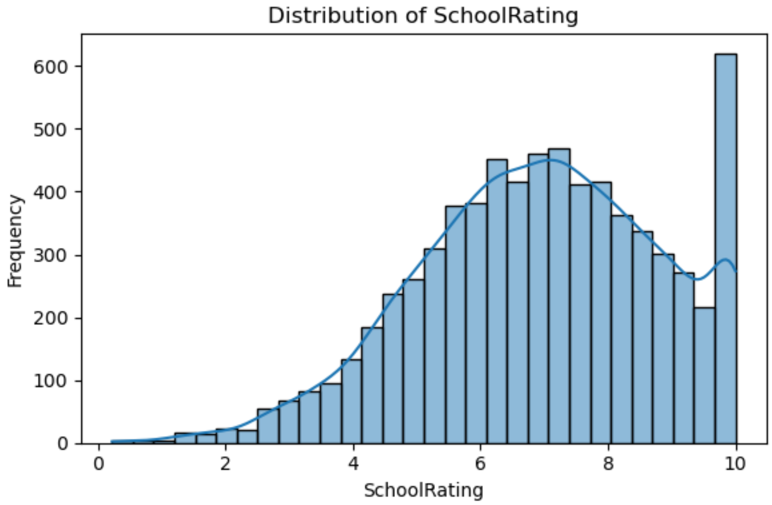


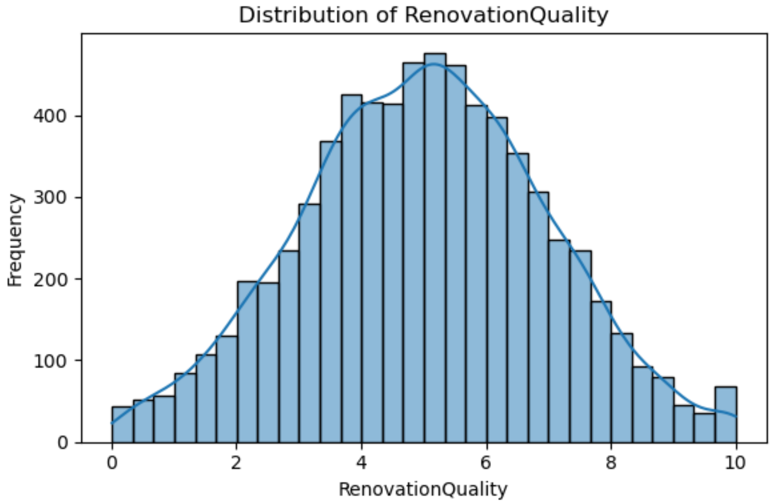
Distribution of BackyardSpace



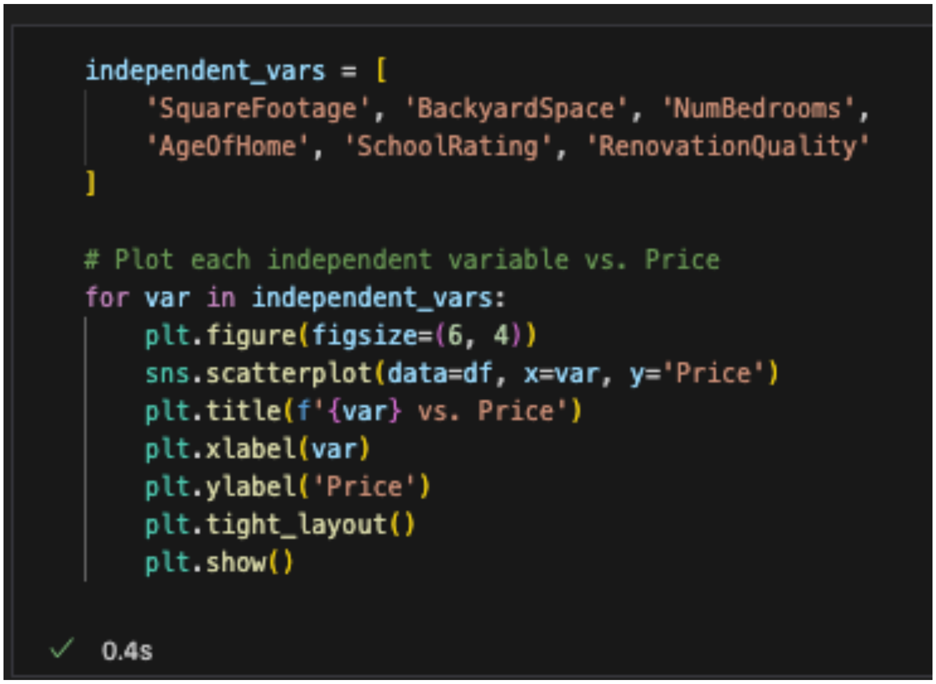
Distribution of NumBedrooms


Distribution of AgeOfHome
Frequency
0
100
200
300
400
500
0
25
50
75
100
125
150
175
AgeOfHome


Distribution of SchoolRating
Frequency
0
100
200
300
400
500
600
0
2
4
6
8
10
SchoolRating

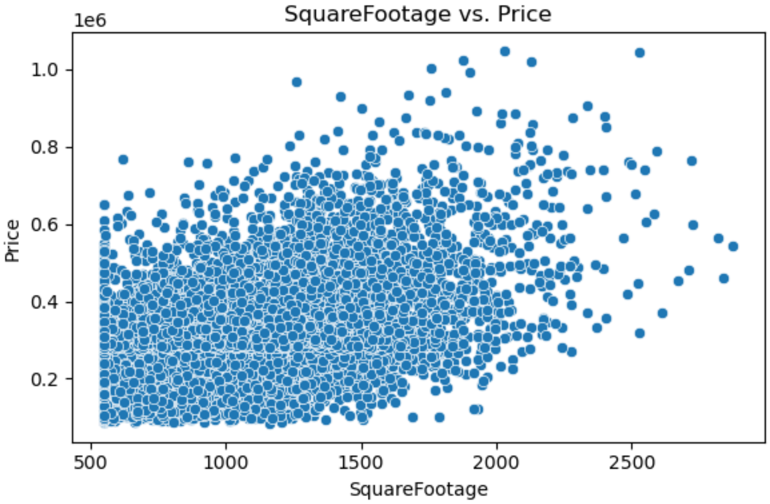

Distribution of RenovationQuality
Frequency
0
100
200
300
400
0
2
4
6
8
10
RenovationQuality

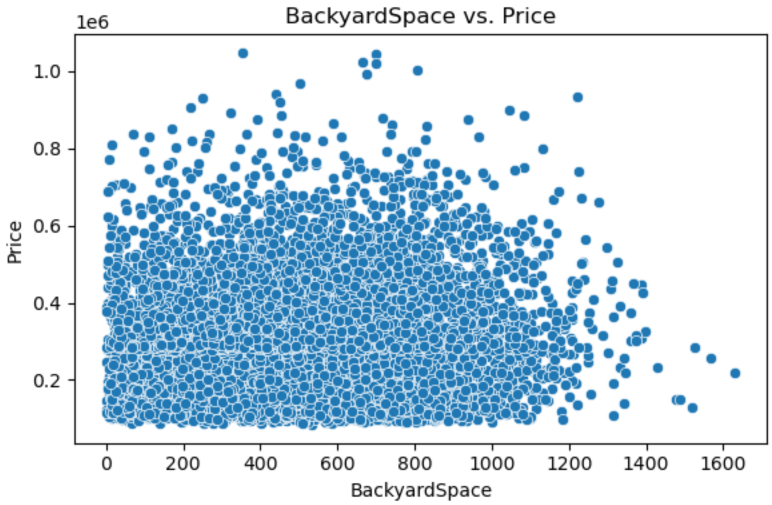
Bivariate –



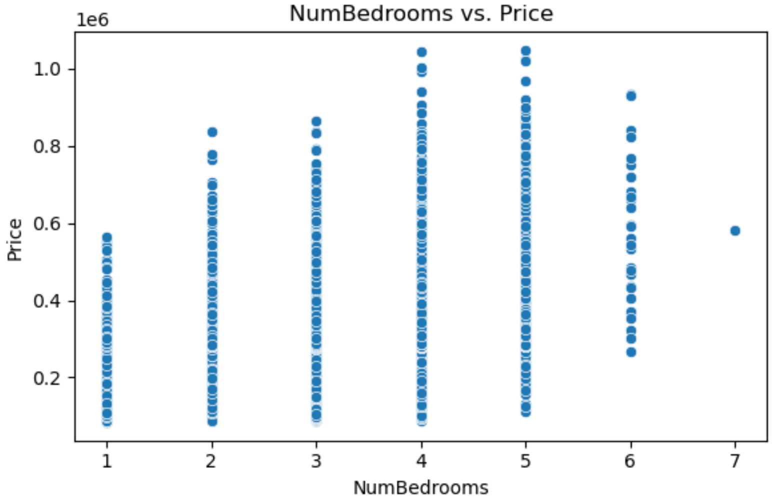
```python
independent_vars = [
    'SquareFootage', 'BackyardSpace', 'NumBedrooms',
    'AgeOfHome', 'SchoolRating', 'RenovationQuality'
]

# Plot each independent variable vs. Price
for var in independent_vars:
    plt.figure(figsize=(6, 4))
    sns.scatterplot(data=df, x=var, y='Price')
    plt.title(f'{var} vs. Price')
    plt.xlabel(var)
    plt.ylabel('Price')
    plt.tight_layout()
    plt.show()
```

✓ 0.4s

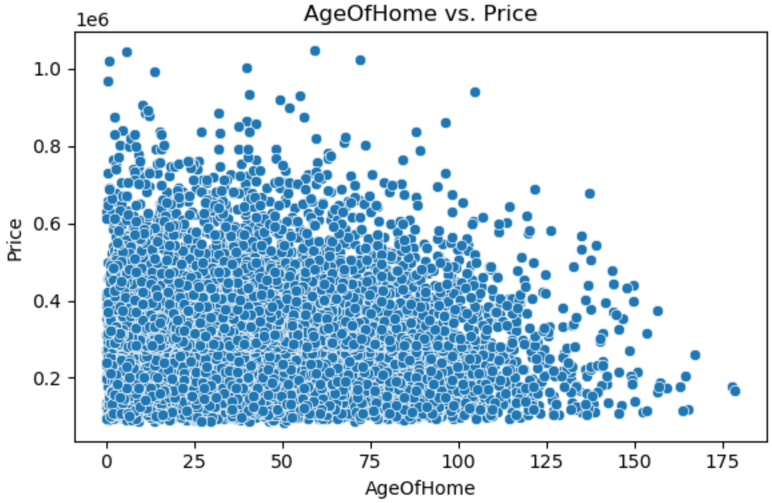




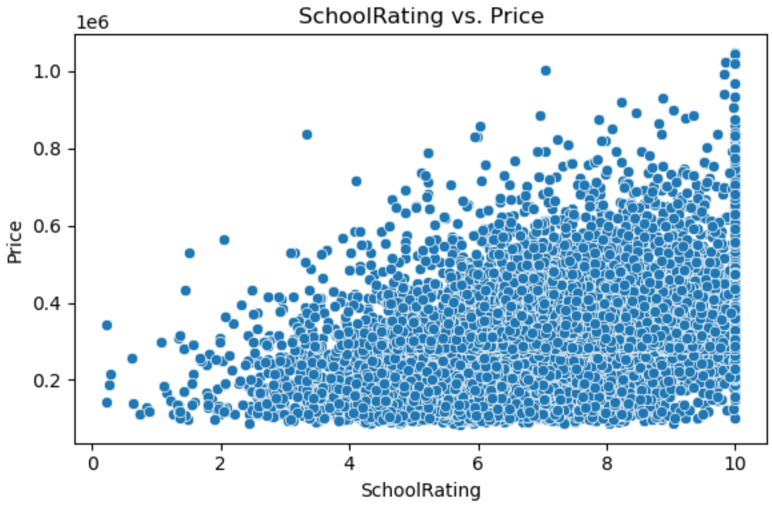
BackyardSpace vs. Price

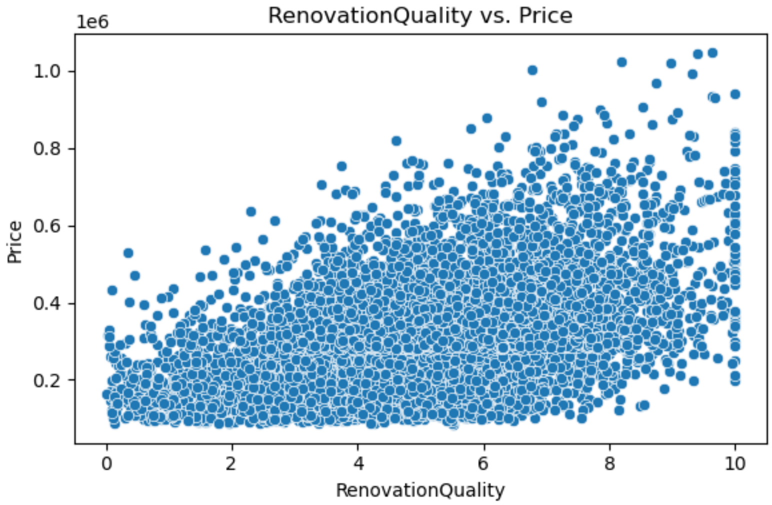


NumBedrooms vs. Price



AgeOfHome vs. Price





D. Perform the data analysis and report on the results by doing the following:

1. Split the data into two datasets, with a larger percentage assigned to the training dataset and a smaller percentage assigned to the test dataset. Provide the files.

   training_dataset.csv *(attached in submission)*

   test_dataset.csv *(attached in submission)*

2. Use the training dataset to create and perform a regression model using regression as a statistical method. Optimize the regression model using a process of your selection, including but not limited to, forward stepwise selection, backward stepwise elimination, and recursive selection. Provide a screenshot of the summary of the optimized model or the following extracted model parameters:
   Before optimization of the model:
   - Adjusted R2: **0.513**
   - R2: **0.514**

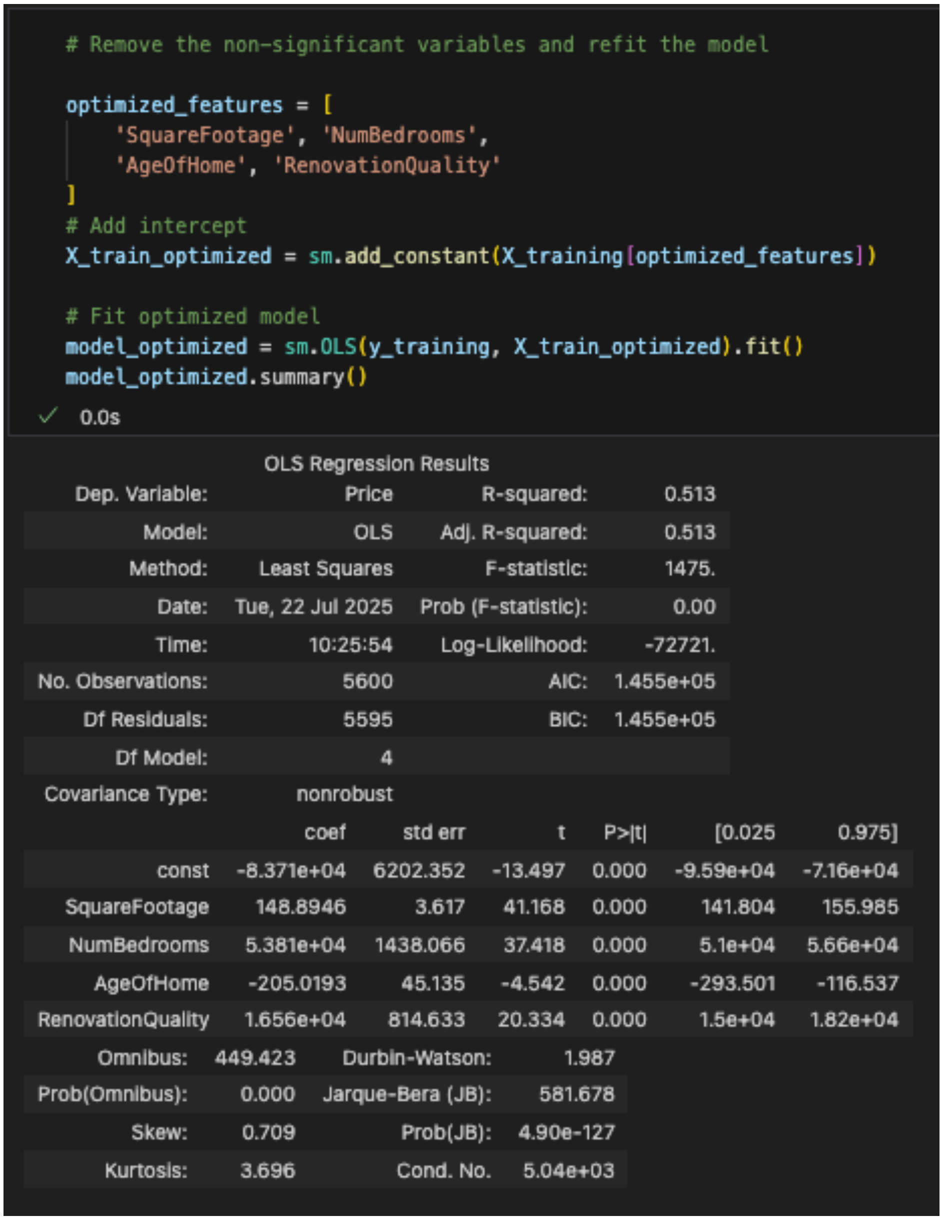
- F statistics: **984.1**
- Probability F statistics: **0.00**
- Coefficient estimates:

	coef
const	-8.742e+04
SquareFootage	148.3292
BackyardSpace	-2.2685
NumBedrooms	5.314e+04
AgeOfHome	-201.5456
SchoolRating	1409.3323
RenovationQuality	1.608e+04

- P-value of each independent variable:

	P>|t|
const	0.000
SquareFootage	0.000
BackyardSpace	**0.655**
NumBedrooms	0.000
AgeOfHome	0.000
SchoolRating	**0.122**
RenovationQuality	0.000

Due to `BackyardSpace` and `SchoolRating` being non-significant variables (p-value > 0.05), we will use **backward stepwise elimination** to **remove those variables and optimize the model**.



```
# Remove the non-significant variables and refit the model

optimized_features = [
    'SquareFootage', 'NumBedrooms',
    'AgeOfHome', 'RenovationQuality'
]
# Add intercept
X_train_optimized = sm.add_constant(X_training[optimized_features])

# Fit optimized model
model_optimized = sm.OLS(y_training, X_train_optimized).fit()
model_optimized.summary()
```

✓ 0.0s

```
                    OLS Regression Results
     Dep. Variable:           Price        R-squared:      0.513
            Model:              OLS    Adj. R-squared:      0.513
           Method:    Least Squares        F-statistic:      1475.
             Date: Tue, 22 Jul 2025  Prob (F-statistic):      0.00
             Time:         10:25:54     Log-Likelihood:    -72721.
 No. Observations:             5600                AIC:  1.455e+05
     Df Residuals:             5595                BIC:  1.455e+05
         Df Model:                4
  Covariance Type:        nonrobust
                        coef   std err        t   P>|t|    [0.025     0.975]
            const  -8.371e+04  6202.352  -13.497   0.000  -9.59e+04  -7.16e+04
    SquareFootage   148.8946     3.617   41.168   0.000    141.804    155.985
      NumBedrooms  5.381e+04  1438.066   37.418   0.000     5.1e+04   5.66e+04
        AgeOfHome  -205.0193    45.135   -4.542   0.000   -293.501   -116.537
RenovationQuality  1.656e+04   814.633   20.334   0.000     1.5e+04   1.82e+04
      Omnibus:  449.423    Durbin-Watson:      1.987
Prob(Omnibus):    0.000  Jarque-Bera (JB):   581.678
         Skew:    0.709         Prob(JB):  4.90e-127
     Kurtosis:    3.696         Cond. No.   5.04e+03
```

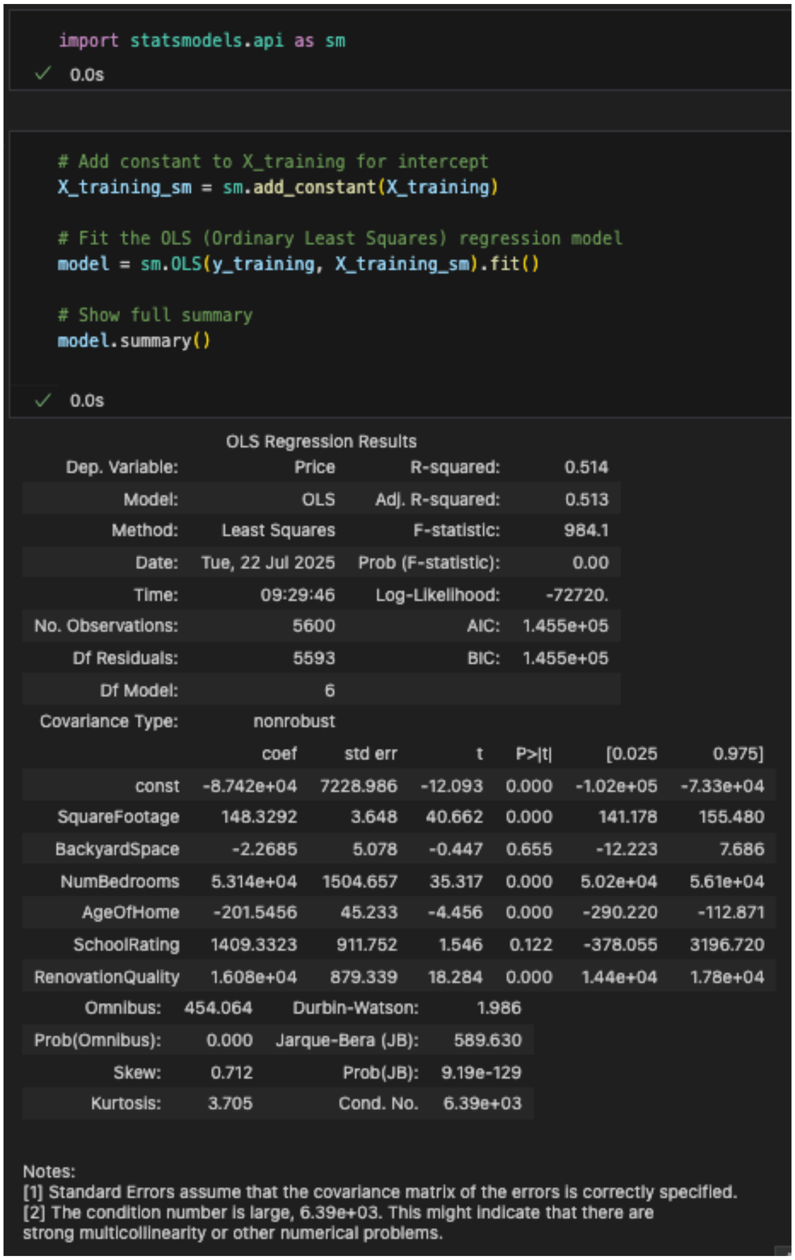
After optimization of the model:

- Adjusted R2: **0.513**
- R2: **0.513**
- F statistics: **1475**
- Probability F statistics: **0.00**
- Coefficient estimates:

	coef
const	-8.371e+04
SquareFootage	148.8946
NumBedrooms	5.381e+04
AgeOfHome	-205.0193
RenovationQuality	1.656e+04

- P-value of each independent variable:

	P>|t|
const	0.000
SquareFootage	0.000
NumBedrooms	0.000
AgeOfHome	0.000
RenovationQuality	0.000



```
import statsmodels.api as sm
```

✓ 0.0s

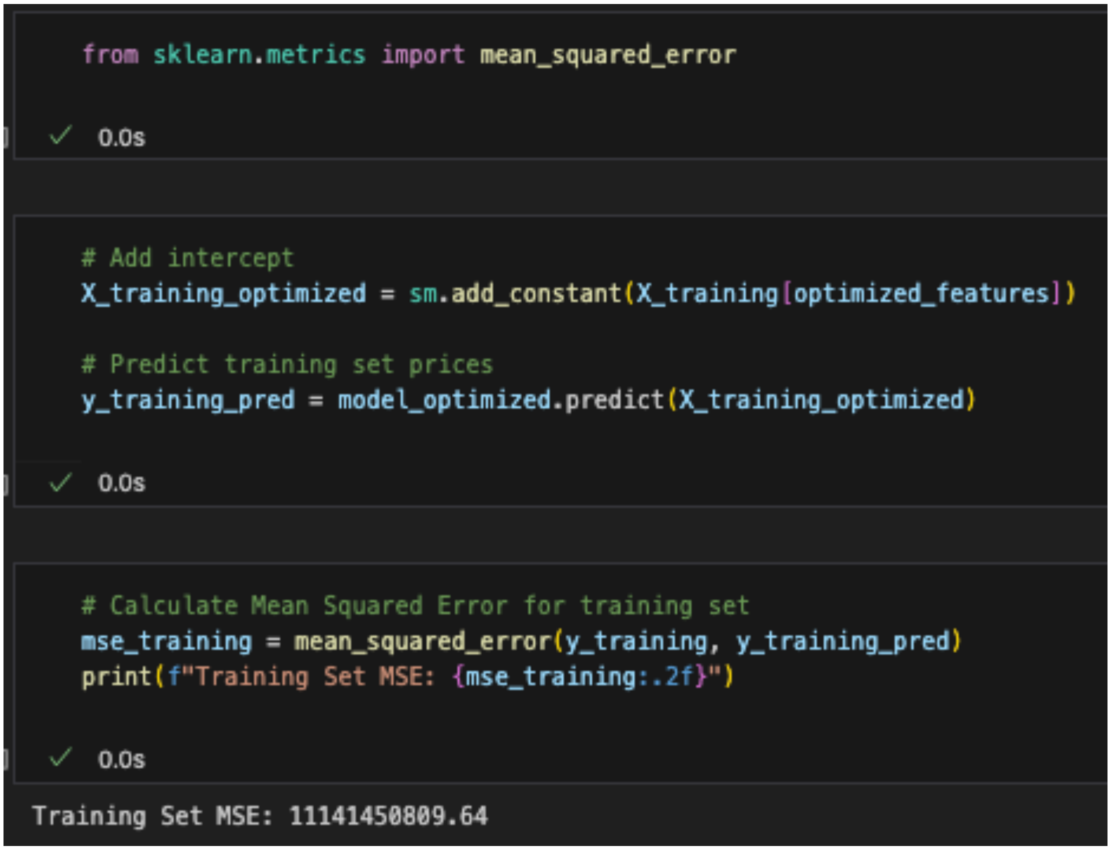
```
# Add constant to X_training for intercept
X_training_sm = sm.add_constant(X_training)

# Fit the OLS (Ordinary Least Squares) regression model
model = sm.OLS(y_training, X_training_sm).fit()

# Show full summary
model.summary()
```

✓ 0.0s

```
                        OLS Regression Results
      Dep. Variable:            Price          R-squared:      0.514
             Model:               OLS      Adj. R-squared:     0.513
            Method:     Least Squares         F-statistic:     984.1
              Date:  Tue, 22 Jul 2025   Prob (F-statistic):     0.00
              Time:          09:29:46      Log-Likelihood:   -72720.
 No. Observations:              5600                 AIC:  1.455e+05
      Df Residuals:              5593                 BIC:  1.455e+05
          Df Model:                 6
   Covariance Type:         nonrobust
                         coef    std err        t   P>|t|      [0.025     0.975]
             const  -8.742e+04   7228.986  -12.093   0.000   -1.02e+05  -7.33e+04
     SquareFootage   148.3292      3.648   40.662   0.000    141.178    155.480
     BackyardSpace    -2.2685      5.078   -0.447   0.655    -12.223      7.686
       NumBedrooms  5.314e+04   1504.657   35.317   0.000   5.02e+04   5.61e+04
         AgeOfHome  -201.5456     45.233   -4.456   0.000   -290.220   -112.871
      SchoolRating  1409.3323    911.752    1.546   0.122   -378.055   3196.720
 RenovationQuality  1.608e+04    879.339   18.284   0.000   1.44e+04   1.78e+04
          Omnibus:  454.064    Durbin-Watson:      1.986
    Prob(Omnibus):    0.000  Jarque-Bera (JB):   589.630
             Skew:    0.712         Prob(JB):  9.19e-129
         Kurtosis:    3.705        Cond. No.   6.39e+03


Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.39e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

3. Give the mean squared error (MSE) of the optimized model used on the training set.

   Training Set MSE: 11141450809.64



```
from sklearn.metrics import mean_squared_error
```

✓ 0.0s

```
# Add intercept
X_training_optimized = sm.add_constant(X_training[optimized_features])

# Predict training set prices
y_training_pred = model_optimized.predict(X_training_optimized)
```

✓ 0.0s

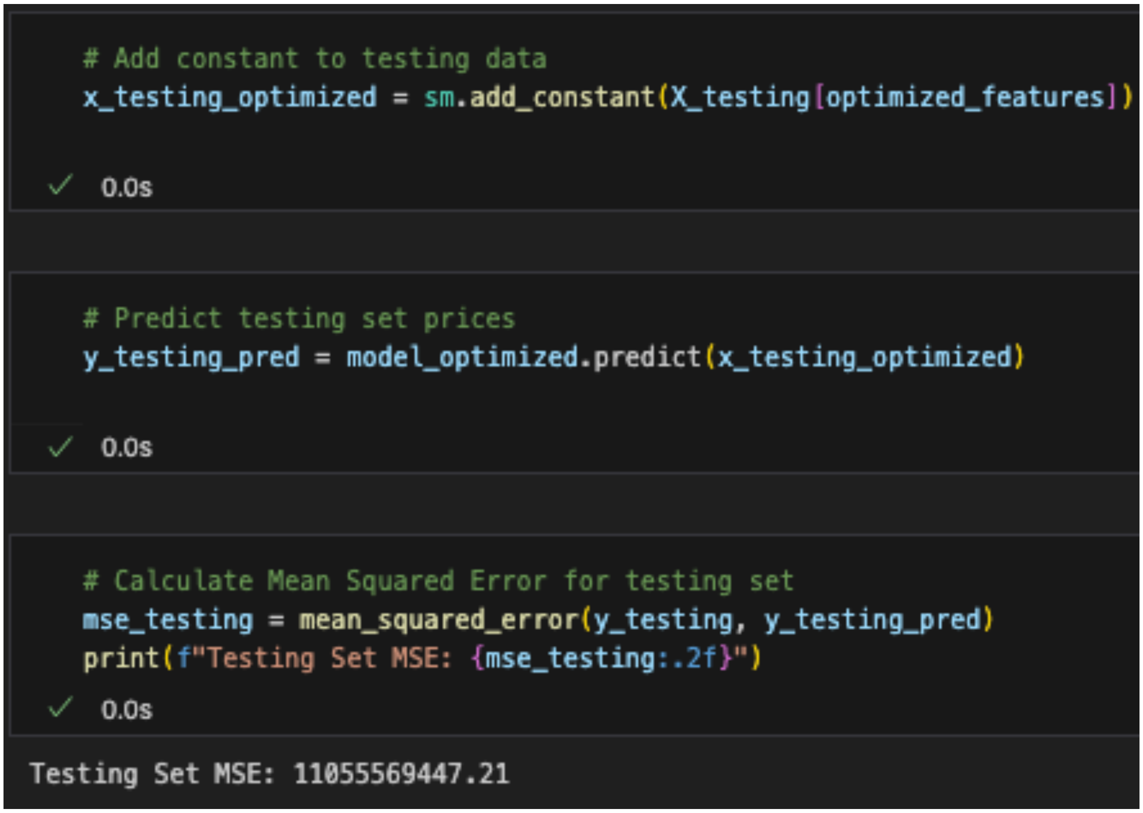
```
# Calculate Mean Squared Error for training set
mse_training = mean_squared_error(y_training, y_training_pred)
print(f"Training Set MSE: {mse_training:.2f}")
```

✓ 0.0s

```
Training Set MSE: 11141450809.64
```

4. Run the prediction on the test dataset using the optimized regression model from part D2 to give the accuracy of the prediction model based on the mean squared error (MSE).

   Testing Set MSE: 11055569447.21



```
# Add constant to testing data
x_testing_optimized = sm.add_constant(X_testing[optimized_features])
```

✓ 0.0s

```
# Predict testing set prices
y_testing_pred = model_optimized.predict(x_testing_optimized)
```

✓ 0.0s

```
# Calculate Mean Squared Error for testing set
mse_testing = mean_squared_error(y_testing, y_testing_pred)
print(f"Testing Set MSE: {mse_testing:.2f}")
```

✓ 0.0s

```
Testing Set MSE: 11055569447.21
```

E. Summarize your data analysis by doing the following:

1. List the packages or libraries you have chosen for Python or R and justify how each item on the list supports the analysis.
    a. `import pandas as pd` – used to load, clean, and explore the dataset as a dataframe (df), making it easier to manipulate the data
    b. `import numpy as np` – used for numerical operations and arithmetic calculations
    c. `import matplotlib.pyplot as plt`
    d. `import seaborn as sns`
        i. matplotlib and seaborn libraries are used to create univariate and bivariate visualizations to understand the distribution and relationships between variables
    e. `import statsmodels.api as sm` – used to build the linear regression model and find stat model diagnostics (e.g., p-values)
    f. `from sklearn.model_selection import train_test_split` – used to split the dataset into a training set and testing set
    g. `from sklearn.metrics import mean_squared_error` – used to calculate MSE and evaluate the performance of the regression model

2. Discuss the method used to optimize the model and justification for the approach.

    To optimize the model, I used **backward stepwise elimination**. I started with a model that included all 6 of the independent variables I chose: `SquareFootage`,

`BackyardSpace`, `NumBedrooms`, `AgeOfHome`, `SchoolRating`, and `RenovationQuality`. After fitting the model using `statsmodel.OLS,` the p-values for `BackyardSpace` and `SchoolRating` were greater than 0.05, indicating they were not statistically significant.
I removed these 2 variables and re-ran the model with the remaining 4 independent variables.
This method improved the model accuracy by only focusing on the variables with meaningful contributions to the prediction of house prices. This new optimized model also helped reduce potential overfitting (*when the model learns patterns in the training data that aren't useful for predicting future data)* by excluding irrelevant features.

3. Discuss the verification of assumptions used to create the optimized model.

   Linearity – The relationship between each independent variable and the dependent variable should be linear. **Scatterplots** were created between variables like `SquareFootage` and `Price`, and most showed a clear upward or downward trend – indicating a linear relationship.

   No Multicollinearity – The independent variables shouldn't be too highly correlated with each other.
   The variables were carefully selected to ensure different types of features were included (e.g., size, quality, age, etc.)

   Normality of Residuals – A histogram of residuals displayed fairly normal distributions.

4. Provide the regression equation and discuss the coefficient estimates.

   Price = β_0 + β_1*SquareFootage + β_2*NumBedrooms - β_3*AgeOfHome + β_4*RenovationQuality

   Price = -83.710.44 + 148.89*SquareFootage + 5,381.04*NumBedrooms - 205.02*AgeOfHome + 16,650.44*RenovationQuality

   Each coefficient estimate represents the expected change in predicted price for a one-unit increase in the variable, holding all other variables constant. As an example, for a one-unit increase in square footage, the expected price increases by $148.89, while there's a reduction in price by $205.05 for a home age going up by one year.

5. Discuss the model metrics by addressing each of the following:
   - The R2 and adjusted R2 of the training set
     - R2 = 0.513
     - Adjusted R2 = 0.513

    - This means the model explains 51.3% of the variance in housing prices in the training dataset.
- The comparison of the MSE for the training set to the MSE of the test set
    - MSE for the training set: 11,141,450,809.64
    - MSE for the testing set: 11,055,569,447.21
    - The close similarity between the training and test RMSE values suggests that the model is good for handling new data and is not overfitting.

6. Discuss the results and implications of your prediction analysis.

   The results of the prediction analysis show that the most influential factors affecting housing prices are square footage, number of bedrooms, age of the home, and renovation quality. Each of these variables had a statistically significant impact on home price. Independent variables like backyard space and school rating were removed from our optimized model as they showed a lack of statistically significant evidence.

   The R2 of 0.513 shows moderate predictive power. This implies that real estate professionals, appraisers, or property investors can focus on a few key features – especially square footage and renovation quality – when estimating home and property value. Renovation quality showed a large positive influence on price, suggesting that quality property improvements could lead to a high market value price.

7. Recommend a course of action for the real-world organizational situation from part B1 based on your results and implications discussed in part E6.

   I recommend that the organization prioritize square footage, renovation quality, and number of bedrooms when evaluating or pricing properties. These features were statistically significant predictors of house prices and offer the most value in forecasting outcomes.

   As it relates to our question, *How do housing features (square footage, number of bedrooms, and backyard space) influence housing prices?*, investment in renovations can lead to significant increases in sale price. Also, accurately assessing square footage and number of bedrooms in the home can improve the price of the home and guide buyers towards high-value properties.

   Lastly, I recommend the organization to use this model as a baseline pricing tool while continuously collecting more data to improve model prediction accuracy and analysis.

F. Panopto Presentation

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e972f1c5-d0a8-4919-8a3f-b322010b3eee

G. Sources

a. No other external sources were used beyond the sources provided by WGU.