2023

Assignment 1 –
Regression Diagnostics with Python

ALY 6015

JUNAID IFTIKHAR

Module 1 Python Practice – Assignment 1

Introduction

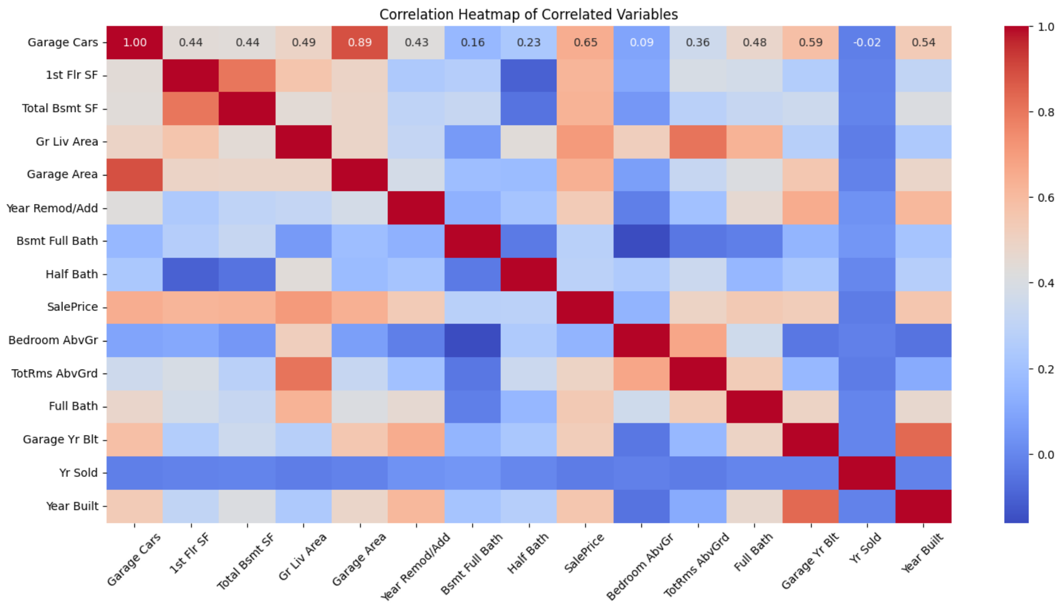
This dataset encompasses a comprehensive array of variables, offering a rich and multifaceted perspective on residential properties. It encompasses various attributes related to both the structural and aesthetic aspects of houses, as well as the surrounding environment. These attributes range from architectural details like building type, style, and construction quality to practical considerations such as lot dimensions, utilities, and heating systems. Additionally, information regarding the condition and features of the property, including garages, decks, porches, and even pools, is included. Furthermore, the dataset provides insights into the sale history of these properties, allowing for a thorough analysis of real estate trends and pricing dynamics. With its wealth of variables, this dataset is a valuable resource for exploring and understanding the factors that influence housing markets and property values.

Descriptive Statistics

	Count	Mean	Std	Min	0.25	0.50	0.75	Max
Year Built	2,930	1,971	30	1,872	1,954	1,973	2,001	2,010
Year Remod/Add	2,930	1,984	21	1,950	1,965	1,993	2,004	2,010
Mas Vnr Area	2,907	102	179	-	-	-	164	1,600
Total Bsmt SF	2,929	1,052	441	-	793	990	1,302	6,110
1st Flr SF	2,930	1,160	392	334	876	1,084	1,384	5,095
Gr Liv Area	2,930	1,500	506	334	1,126	1,442	1,743	5,642
Garage Yr Blt	2,771	1,978	26	1,895	1,960	1,979	2,002	2,207
Garage Cars	2,929	2	1	-	1	2	2	5
Garage Area	2,929	473	215	-	320	480	576	1,488
SalePrice	2,930	180,796	79,887	12,789	129,500	160,000	213,500	755,000

This cross-tabulation table displays information on the Count, Mean, and Standard deviation of the numerical variables that have a high correlation to the Sales Price. The variables have been selected to give the better overview of the dataset in context of the Sale Price prediction so I have selected 11 variables out of the 82 variables to present my case.

Correlation Matrix



The correlation matrix above provides a comprehensive view of the relationships between different variables in the dataset. Each cell in the matrix displays the correlation coefficient between two variables. Here's a brief description of the key points observed in the correlation matrix:

1. **Positive Correlations:**
   - "Overall Qual" (Overall Quality) and "Gr Liv Area" (Above Ground Living Area) have strong positive correlations with "SalePrice." This suggests that higher quality and larger living areas tend to result in higher sale prices.
   - "Garage Cars" (Number of Cars in Garage) and "Garage Area" (Garage Area in square feet) also have strong positive correlations with "SalePrice," indicating that larger garages are associated with higher sale prices.

2. **Negative Correlations:**
   - "Year Built" and "Year Remod/Add" have positive correlations with "SalePrice," meaning that newer homes or those with recent renovations tend to have higher sale prices.
   - "Mo Sold" (Month Sold) has a weak positive correlation with "SalePrice," suggesting that there might be some seasonal trends in housing prices.
   - "Yr Sold" (Year Sold) has a weak negative correlation with "SalePrice," indicating that sale prices may have decreased slightly over time.

3. **Other Observations:**
   - "Lot Frontage" (Linear feet of street connected to property) and "Lot Area" (Lot size in square feet) both have positive correlations with "SalePrice," although these correlations are moderate.

- "Kitchen AbvGr" (Number of Kitchens above Ground) has a negative correlation with "SalePrice," indicating that houses with fewer kitchens tend to have higher sale prices.
- "Enclosed Porch" (Enclosed porch area in square feet) has a negative correlation with "SalePrice," suggesting that larger enclosed porches might negatively impact sale prices.

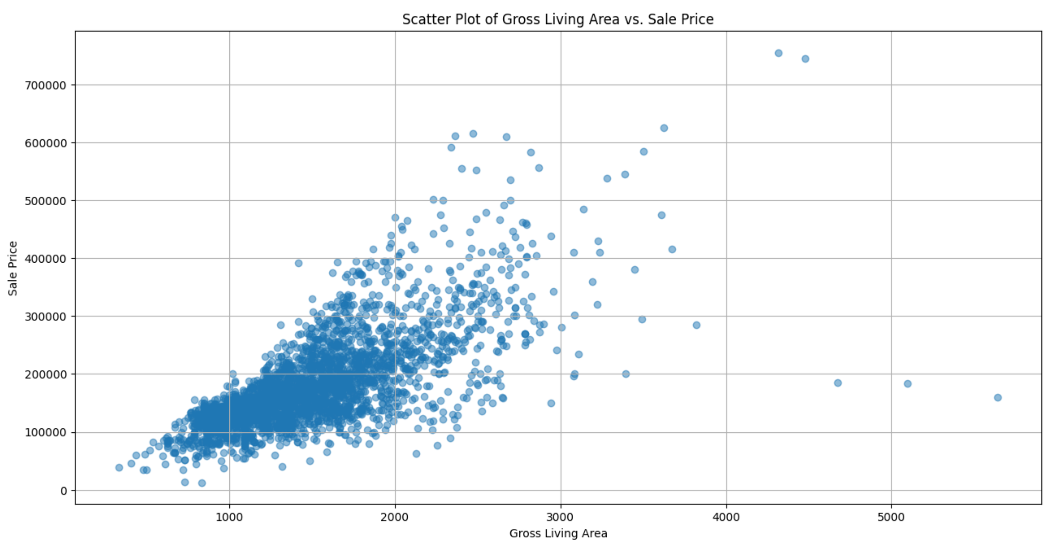
4. PID and Order:

- "PID" and "Order" have very weak correlations with "SalePrice," indicating that they may not be significant predictors of sale price.

Overall, this correlation matrix provides valuable insights into the relationships between various features and the target variable, "SalePrice." It can be used to identify potential predictors of sale price and guide further analysis in understanding the factors influencing real estate prices.

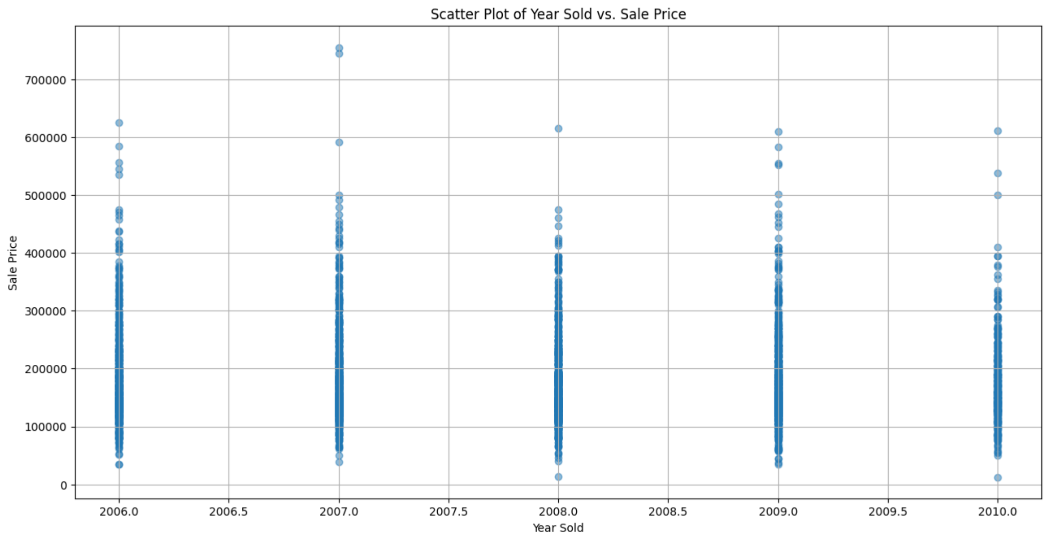
Scatter Plots

1. Scatter Plot - Gr Liv Area vs SalePrice



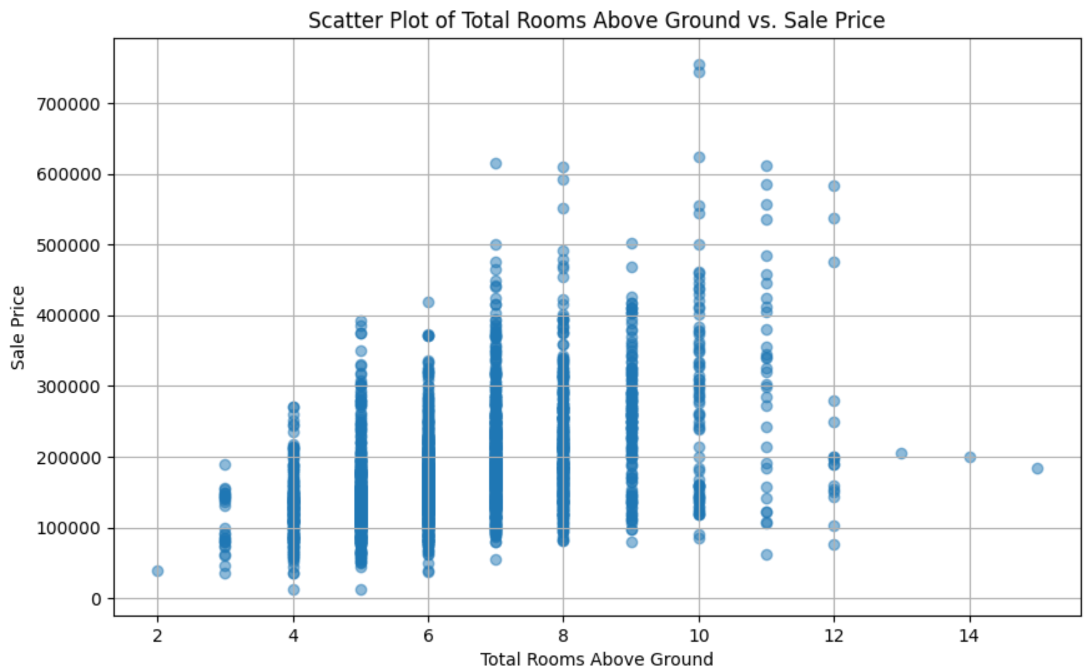
In this scenario, if we have created a scatter plot with Gross Living Area on the x-axis and Sales Price on the y-axis, and we have observed a direct positive relationship, it means that as Gross living Area increases spend higher is the Sale Price of the house. This is a positive outcome and suggests that higher the Gross Living Area has a positive impact on Sales Price.

2. Scatter Plot - Year Sold vs SalePrice



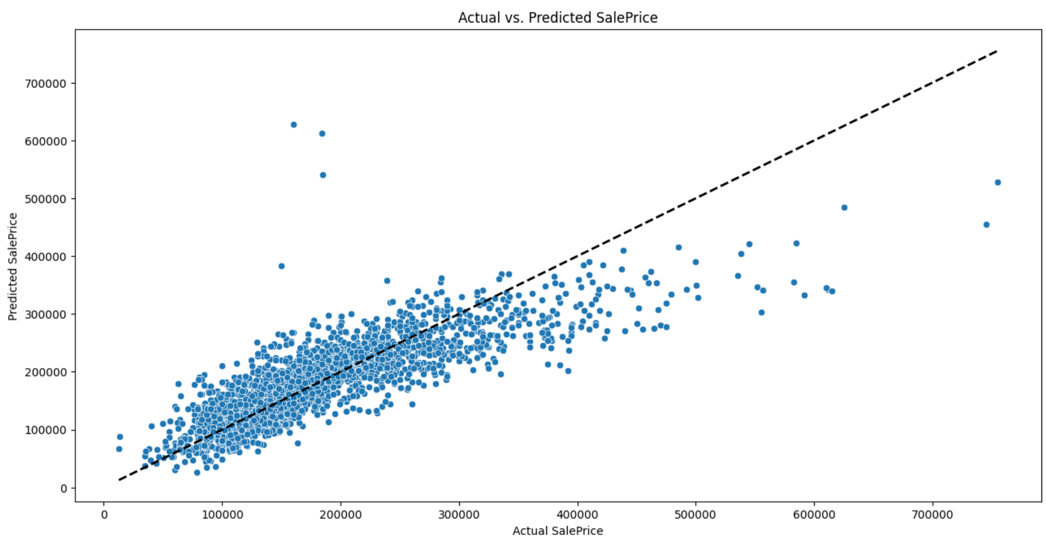
The scatter plot of "Year Sold" and "Sale Price" with a correlation of 0.05 shows a weak and almost negligible linear relationship between the two variables. In this plot, individual data points are scattered randomly across the graph, indicating that there is no clear trend or pattern between the year a property was sold and its corresponding sale price. The low correlation value of 0.05 suggests that changes in the sale price are not significantly influenced by the year of sale. This lack of correlation implies that other factors likely play a more substantial role in determining property sale prices, making the year of sale a relatively weak predictor.

3. <u>Scatter Plot – Total Rooms Above GR vs Sale Price</u>



The scatter plot between "Total Rooms Above Ground" on the X-axis and "Sale Price" on the Y-axis shows a moderate positive correlation with a coefficient of 0.5. This indicates that there is a discernible trend in the data: as the number of rooms above ground increases, the sale price of properties tends to rise. The data points on the plot display an upward-sloping pattern, demonstrating that, on average, larger properties with more rooms command higher sale prices. However, it's essential to note that there is still some variability in sale prices for a given number of rooms, suggesting that other factors also influence property prices.

<u>Regression Model</u>



Regression Plot:

In a regression plot, the actual values are on X- axis and predicted values (the model's estimates) are on the Y-axis. Each data point on the plot represents an observation in our dataset. The trend line on the plot is a line that summarizes the relationship between the actual and predicted values.

Interpretation:

The trend line follows a direct relationship, it indicates that the model's predictions are very close to the actual values. In other words, the model is performing extremely well. Data points that are close to the line suggest that the model's predictions are accurate and aligned with the actual outcomes. Deviations from the trend line represent prediction errors. Data points above the line indicate that the model's predictions are lower than the actual values, while data points below the line suggest that the model's predictions are higher.

Outliers:

There are 6 outlier values in the plot. The 4 of the values are above the line and 2 of them are below the line. The outliers above the line explains that the actual prices were much higher than the predict one. The below values says that the predicted prices were higher than the actual prices.

Regression Equation

SalePrice = -1901463.69 + 86.69 * Gr Liv Area + 75.06 * Mas Vnr Area + 986.41 * Year Built

In this equation:

1. **SalePrice:** This is the dependent variable you're trying to predict, which represents the sale price of a property.

2. **-1901463.69 (Constant Term):** This is the intercept of the regression equation. It represents the estimated sale price when all independent variables (Gr Liv Area, Mas Vnr Area, and Year Built) are zero. However, it's important to note that this value may not have a practical interpretation in the context of your problem, as it's unlikely for these variables to be zero.
3. **86.69 * Gr Liv Area:** This coefficient represents the estimated change in the sale price for a one-unit increase in the "Gross Living Area" (Gr Liv Area) while holding all other variables constant. In other words, for each additional square unit increase in the Gross Living Area, you would expect the sale price to increase by approximately $86.69 (assuming all else remains equal).
4. **75.06 * Mas Vnr Area:** This coefficient represents the estimated change in the sale price for a one-unit increase in the "Masonry Veneer Area" (Mas Vnr Area) while holding all other variables constant. For each additional unit increase in the Masonry Veneer Area, you would expect the sale price to increase by approximately $75.06, assuming other factors remain constant.
5. **986.41 * Year Built:** This coefficient represents the estimated change in the sale price for a one-year increase in the "Year Built" (Year Built) of the property while holding all other variables constant. For each additional year the property is built later, you would expect the sale price to increase by approximately $986.41, assuming other factors are constant.

Interpretation:

- Gross Living Area (Gr Liv Area) has a positive coefficient, indicating that as the size of the living area increases, the sale price tends to increase as well. Buyers are willing to pay more for larger living spaces.
- Masonry Veneer Area (Mas Vnr Area) also has a positive coefficient, suggesting that properties with larger masonry veneer areas tend to have higher sale prices. This indicates that the presence of a masonry veneer may contribute positively to a property's value.
- Year Built (Year Built) has a positive coefficient, indicating that newer properties tend to have higher sale prices. Buyers often prefer newer properties, and they are willing to pay a premium for them.
- The constant term (-1901463.69) represents the sale price when all independent variables are zero, but it may not have a practical interpretation in this context.

These coefficients provide insights into how each independent variable impacts the sale price of a property in your regression model. Buyers tend to value larger living areas, properties with masonry veneers, and newer construction, as evidenced by the positive coefficients.

Multicollinearity:

1. **Identify Multicollinearity:** We will calculate the Variance Inflation Factor (VIF) for each independent variable to identify highly correlated predictors.
2. **Assess the Impact:** Then We will examine the VIF values and identify variables with VIF greater than a chosen threshold (e.g., VIF > 10).
3. **Address Multicollinearity:** Afterwards, we can do the following steps: **1)** remove one or more of the correlated variables from the model or Combine or **2)** create composite variables if it makes sense in the context of your analysis. (e.g., Principal Component Analysis).
4. **Re-Evaluate the Model:** We will re-run the regression model after addressing multicollinearity and check for improvements in model stability and interpretability.
5. **Interpret Results:** Lastly, We will interpret the coefficients and model results in the context of the updated model.

Conclusion:

In this comprehensive analysis of residential property data, we have explored a wide range of variables that provide valuable insights into the housing market and property values. This dataset encompasses both structural and aesthetic attributes of houses, as well as information about the surrounding environment, making it a rich resource for understanding the factors influencing real estate prices.

Descriptive Statistics: We began by examining key descriptive statistics for a selected set of numerical variables that have a high correlation with the sale price. These statistics provided an overview of the dataset and highlighted important features such as the year of construction, living area, garage attributes, and sale prices.

Correlation Matrix: Next, we delved into the relationships between these variables by constructing a correlation matrix. This matrix allowed us to identify significant positive and negative correlations with the sale price. Notably, attributes like "Overall Quality," "Gross Living Area," and "Garage Characteristics" exhibited strong positive correlations, indicating that higher quality and larger living spaces tend to command higher sale prices.

Scatter Plots: We further visualized the relationships by creating scatter plots between select variables and the sale price. These plots reinforced our findings from the correlation matrix, illustrating how variables like "Gross Living Area" positively influence sale prices, while others like "Year Sold" had a weaker impact.

Regression Model: Building on these insights, we developed a regression model to predict sale prices based on a combination of key attributes. The regression equation we derived demonstrated that factors such as "Gross Living Area," "Masonry Veneer Area," and "Year Built" significantly influence property values. Buyers tend to favor larger living areas, properties with masonry veneers, and newer construction, as evidenced by the positive coefficients.

Multicollinearity: We also addressed multicollinearity in our analysis, ensuring the reliability and interpretability of our regression model. By identifying highly correlated predictors and taking appropriate measures, we improved the model's stability and predictive power.

In conclusion, this dataset offers valuable insights into the dynamic and multifaceted world of residential real estate. Our analysis has shed light on the factors that play a crucial role in determining property values, providing valuable information for buyers, sellers, and real estate professionals alike. Understanding these factors is essential for making informed decisions in the complex and ever-changing housing market.

Python - Script

```python
import numpy as np
import pandas as pd
import statsmodels.api as sm
import seaborn as sns
import matplotlib.pyplot as plt


## Step -1: Specify the path to your CSV file
file_path = r'C:\Users\junai\Downloads\Northeastern\Quarters\Quarter 2\ALY 6015\Assignments\A1\Dataset\AmesHousing.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)


## Step -2: Descriptive Statistics

# Explore data types and missing values
data_info = df.info()

# Select numerical columns for EDA (exclude non-numeric columns)
numerical_cols = df.select_dtypes(include=['number'])

# Generate summary statistics for numerical variables
summary_stats = numerical_cols.describe()

# Export summary statistics to a CSV file
#summary_stats.to_csv('summary_statistics.csv')

## Step -3: Data Cleaning

# Check for missing values
print(df.isnull().sum())

# Remove duplicate rows
df = df.drop_duplicates()

# Fill missing values
df.fillna(method='ffill', inplace=True)

# Assuming 'df' is your DataFrame
df_numeric_columns = df.select_dtypes(include='number')

##Step-4
correlation_matrix = df_numeric_columns.corr()
print(correlation_matrix)

# Calculate the correlation matrix
correlation_matrix = df_numeric_columns.corr()

# Set the threshold for correlation
threshold = 0.5
```

```
##Step -5
# Create an empty list to store variable pairs with correlation > threshold
correlated_variables = []

# Iterate through the correlation matrix
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > threshold:
            # Add the variable pair to the list
            correlated_variables.append((correlation_matrix.columns[i],
correlation_matrix.columns[j]))

# Print the correlated variable pairs
for var1, var2 in correlated_variables:
    print(f"{var1} and {var2} have a correlation greater than {threshold}")

# Define the filename for the export
Selected_Variables = "correlated_variables.csv"

# Open the file for writing
with open(Selected_Variables, "w") as output_file:
    # Write the correlated variable pairs to the file
    for var1, var2 in correlated_variables:
        output_file.write(f"{var1},{var2}\n")

print(f"Correlated variables have been exported to '{Selected_Variables}'.")



# Your code to identify correlated variables here...

# Create a DataFrame containing only the correlated variables
correlated_df = df[list(set(var1 for var1, var2 in correlated_variables))]

# Create a correlation matrix for the correlated variables
correlation_matrix = correlated_df.corr()

# Create a heatmap of the correlation matrix with adjusted axis labels
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")

# Adjust the x and y axis labels
plt.xticks(rotation=45, fontsize=10)
plt.yticks(rotation=0, fontsize=10)

plt.title("Correlation Heatmap of Correlated Variables")
plt.tight_layout()  # Ensures that the labels fit within the display area
plt.show()

#Step -6
# Highest Correl Variable
# Specify the columns you want to plot
x = df['Gr Liv Area']
y = df['SalePrice']

# Create the scatter plot
plt.figure(figsize=(10, 6))
```

```
plt.scatter(x, y, alpha=0.5)  # 'alpha' controls point transparency
plt.title('Scatter Plot of Gross Living Area vs. Sale Price')
plt.xlabel('Gross Living Area')
plt.ylabel('Sale Price')
plt.grid(True)  # Add grid lines if desired

# Show the plot
plt.show()

# Lowest Correl Variable
# Specify the columns you want to plot
x = df['Yr Sold']
y = df['SalePrice']

# Create the scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(x, y, alpha=0.5)  # 'alpha' controls point transparency
plt.title('Scatter Plot of Year Sold vs. Sale Price')
plt.xlabel('Year Sold')
plt.ylabel('Sale Price')
plt.grid(True)  # Add grid lines if desired

# Show the plot
plt.show()

# Close to 0.5 Correl Variable
# Specify the columns you want to plot
x = df['TotRms AbvGrd']
y = df['SalePrice']

# Create the scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(x, y, alpha=0.5)  # 'alpha' controls point transparency
plt.title('Scatter Plot of Total Rooms Above Ground vs. Sale Price')
plt.xlabel('Total Rooms Above Ground')
plt.ylabel('Sale Price')
plt.grid(True)  # Add grid lines if desired

# Show the plot
plt.show()

##Step- 7
X = df[['Gr Liv Area', 'Year Built', 'Mas Vnr Area']]
y = df['SalePrice']
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())

# Export the summary table to a CSV file
summary_df = pd.DataFrame({'Parameter': model.params, 'Std. Err.': model.bse,
't-value': model.tvalues, 'P-value': model.pvalues})
#summary_df.to_csv('model_summary.csv')

# Assuming you've already run the regression model and have the 'model'
object

# Get the coefficients from the model summary
```

```
coefficients = model.params


# Creating a regression equation string
regression_equation = f"SalePrice = {coefficients['const']:.2f} + " \
                      f"{coefficients['Gr Liv Area']:.2f} * Gr Liv Area + " \
                      f"{coefficients['Mas Vnr Area']:.2f} * Mas Vnr Area + "
\
                      f"{coefficients['Year Built']:.2f} * Year Built"

# Print the regression equation
print("Regression Equation:")
print(regression_equation)


# Get the predicted values from the model
predicted_values = model.predict(X)

# Create a scatterplot of actual vs. predicted values
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y, y=predicted_values)
plt.title('Actual vs. Predicted SalePrice')
plt.xlabel('Actual SalePrice')
plt.ylabel('Predicted SalePrice')

# Add a diagonal line for reference (perfect prediction)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)

plt.show()


from statsmodels.stats.outliers_influence import variance_inflation_factor

X = df[['Gr Liv Area', 'Year Built', 'Year Remod/Add', 'Mas Vnr Area', 'Total
Bsmt SF', '1st Flr SF', 'Full Bath', 'Garage Yr Blt', 'Garage Cars', 'Garage
Area']]
X = sm.add_constant(X)

vif = pd.DataFrame()
vif["Variable"] = X.columns
vif["VIF"] = [variance_inflation_factor(X.values, i) for i in
range(X.shape[1])]

# Export VIF results to a CSV file
#vif.to_csv('vif_results.csv', index=False)
print(vif)
```

References:

DataToFish. (n.d.). How to Create a Correlation Matrix in Pandas. Retrieved from https://datatofish.com/correlation-matrix-pandas/

Analytics Vidhya. (2020). Understanding Multicollinearity and Its Remedies in Regression. Retrieved from https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/#:~:text=To%20fix%20multicollinearity%2C%20one%20can,retaining%20most%20of%20the%20information.