**USE BELOW TO KAE ALL THE PLOTS ETC THEN EXPLORE WHAT YOU SEE**

**1. Descriptive Statistics**

* Calculate summary statistics for all numerical columns (mean, median, standard deviation, min, max, quartiles).

python

Copy code

df.describe()

**2. Distribution of Variables**

* Plot histograms and box plots for numerical variables to understand their distributions and identify outliers.

python

Copy code

import seaborn as sns

import matplotlib.pyplot as plt

numeric\_cols = ['price', 'bedrooms', 'bathrooms', 'sqft\_living', 'sqft\_lot', 'floors',

'sqft\_above', 'sqft\_basement', 'yr\_built', 'yr\_renovated', 'lat', 'long',

'sqft\_living15', 'sqft\_lot15']

for col in numeric\_cols:

plt.figure(figsize=(10, 6))

sns.histplot(df[col], kde=True)

plt.title(f'Distribution of {col}')

plt.show()

**3. Correlation Analysis**

* Calculate the correlation matrix to identify relationships between numerical variables.
* Visualize the correlation matrix using a heatmap.

python

Copy code

plt.figure(figsize=(12, 10))

sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

**4. Time Series Analysis**

* Analyze how sales prices change over time.
* Plot time series graphs to observe trends and seasonality.

python

Copy code

df.set\_index('date', inplace=True)

monthly\_sales = df['price'].resample('M').sum()

plt.figure(figsize=(12, 6))

monthly\_sales.plot()

plt.title('Monthly Total Sales')

plt.xlabel('Date')

plt.ylabel('Total Sales')

plt.show()

**5. Categorical Variables Analysis**

* Analyze the distribution and impact of categorical variables such as waterfront, view, condition, grade, and zipcode.
* Use bar plots and box plots to compare prices across these categories.

python

Copy code

categorical\_cols = ['waterfront', 'view', 'condition', 'grade', 'zipcode']

for col in categorical\_cols:

plt.figure(figsize=(12, 6))

sns.boxplot(x=col, y='price', data=df)

plt.title(f'{col} vs Price')

plt.show()

**6. Geographical Analysis**

* Visualize the geographical distribution of house prices using latitude and longitude.

python

Copy code

plt.figure(figsize=(10, 6))

sns.scatterplot(x='long', y='lat', hue='price', palette='coolwarm', data=df)

plt.title('Geographical Distribution of House Prices')

plt.xlabel('Longitude')

plt.ylabel('Latitude')

plt.show()

**7. Feature Engineering**

* Create new features that might be useful for predictive modeling.
  + For example, create a new feature for the age of the house.

python

Copy code

df['house\_age'] = 2024 - df['yr\_built']

**8. Interaction Effects**

* Explore interaction effects between features.
* For example, the effect of sqft\_living and grade on price.

python

Copy code

plt.figure(figsize=(12, 6))

sns.scatterplot(x='sqft\_living', y='price', hue='grade', palette='viridis', data=df)

plt.title('Sqft Living vs Price by Grade')

plt.show()

**9. Missing Values Analysis**

* Check for missing values and decide how to handle them.

python

Copy code

df.isnull().sum()

**10. Outliers Detection**

* Identify and handle outliers, especially in price and square footage columns.

python

Copy code

plt.figure(figsize=(12, 6))

sns.boxplot(x=df['price'])

plt.title('Boxplot of Price')

plt.show()

**11. Pair Plot**

* Use pair plots to visualize pairwise relationships between variables.

python

Copy code

sns.pairplot(df[['price', 'bedrooms', 'bathrooms', 'sqft\_living', 'grade']])

plt.show()

**12. Advanced Visualizations**

* Consider advanced visualizations like joint plots, violin plots, and swarm plots for deeper insights.

python

Copy code

sns.jointplot(x='sqft\_living', y='price', data=df, kind='reg')

plt.show()

These steps will help you gain a deeper understanding of your dataset and uncover insights that can guide your predictive modeling efforts. Be sure to document your findings and visualizations clearly, as this will be valuable for your GitHub project and potential employers.