HousePrice Dataset – 20 Problem Statements with NumPy and Pandas Solutions

## 1. Problem:

Find the average house price.

### Solution:

***avg\_price = df['SalePrice'].mean()***

## 2. Problem:

Determine the number of houses with prices above the average.

### Solution:

***count\_above\_avg = (df['SalePrice'] > df['SalePrice'].mean()).sum()***

## 3. Problem:

Calculate the median lot area.

### Solution:

***median\_lot = df['LotArea'].median()***

## 4. Problem:

Identify the maximum and minimum sale prices.

### Solution:

***max\_price = df['SalePrice'].max()  
min\_price = df['SalePrice'].min()***

## 5. Problem:

Count houses built before 1950.

### Solution:

***old\_houses = (df['YearBuilt'] < 1950).sum()***

## 6. Problem:

Compute the correlation between LotArea and SalePrice.

### Solution:

***correlation = df['LotArea'].corr(df['SalePrice'])***

## 7. Problem:

Find the most common number of garage spaces.

### Solution:

***most\_common\_garage = df['GarageCars'].mode()[0]***

## 8. Problem:

Calculate the standard deviation of SalePrice.

### Solution:

***std\_price = df['SalePrice'].std()***

## 9. Problem:

Determine the number of missing values in each column.

### Solution:

***missing\_values = df.isnull().sum()***

## 10. Problem:

List the top 5 most expensive houses.

### Solution:

***top5 = df.nlargest(5, 'SalePrice')***

## 11. Problem:

Group average house price by neighborhood.

### Solution:

***avg\_by\_neighborhood = df.groupby('Neighborhood')['SalePrice'].mean()***

## 12. Problem:

Normalize the SalePrice column using Min-Max scaling.

### Solution:

***df['SalePrice\_norm'] = (df['SalePrice'] - df['SalePrice'].min()) / (df['SalePrice'].max() - df['SalePrice'].min())***

## 13. Problem:

Determine the percentage of houses with OverallQual ≥ 8.

### Solution:

***high\_quality = (df['OverallQual'] >= 8).mean() \* 100***

## 14. Problem:

Find the average age of houses as of 2025.

### Solution:

***df['HouseAge'] = 2025 - df['YearBuilt']  
avg\_age = df['HouseAge'].mean()***

## 15. Problem:

Identify the neighborhood with the highest median house price.

### Solution:

***top\_neighborhood = df.groupby('Neighborhood')['SalePrice'].median().idxmax()***

## 16. Problem:

Create a new column representing price per square foot.

### Solution:

***df['PricePerSqFt'] = df['SalePrice'] / df['LotArea']***

## 17. Problem:

Determine how many unique house styles exist in the dataset.

### Solution:

***unique\_styles = df['HouseStyle'].nunique()***

## 18. Problem:

Filter houses with 2 or more garages and price above $300,000.

### Solution:

***filtered = df[(df['GarageCars'] >= 2) & (df['SalePrice'] > 300000)]***

## 19. Problem:

Replace missing values in LotFrontage with the median.

### Solution:

***df['LotFrontage'].fillna(df['LotFrontage'].median(), inplace=True)***

## 20. Problem:

Count the number of houses with outlier prices using IQR.

### Solution:

***Q1 = df['SalePrice'].quantile(0.25)  
Q3 = df['SalePrice'].quantile(0.75)  
IQR = Q3 - Q1  
outliers = df[(df['SalePrice'] < Q1 - 1.5\*IQR) | (df['SalePrice'] > Q3 + 1.5\*IQR)].shape[0]***